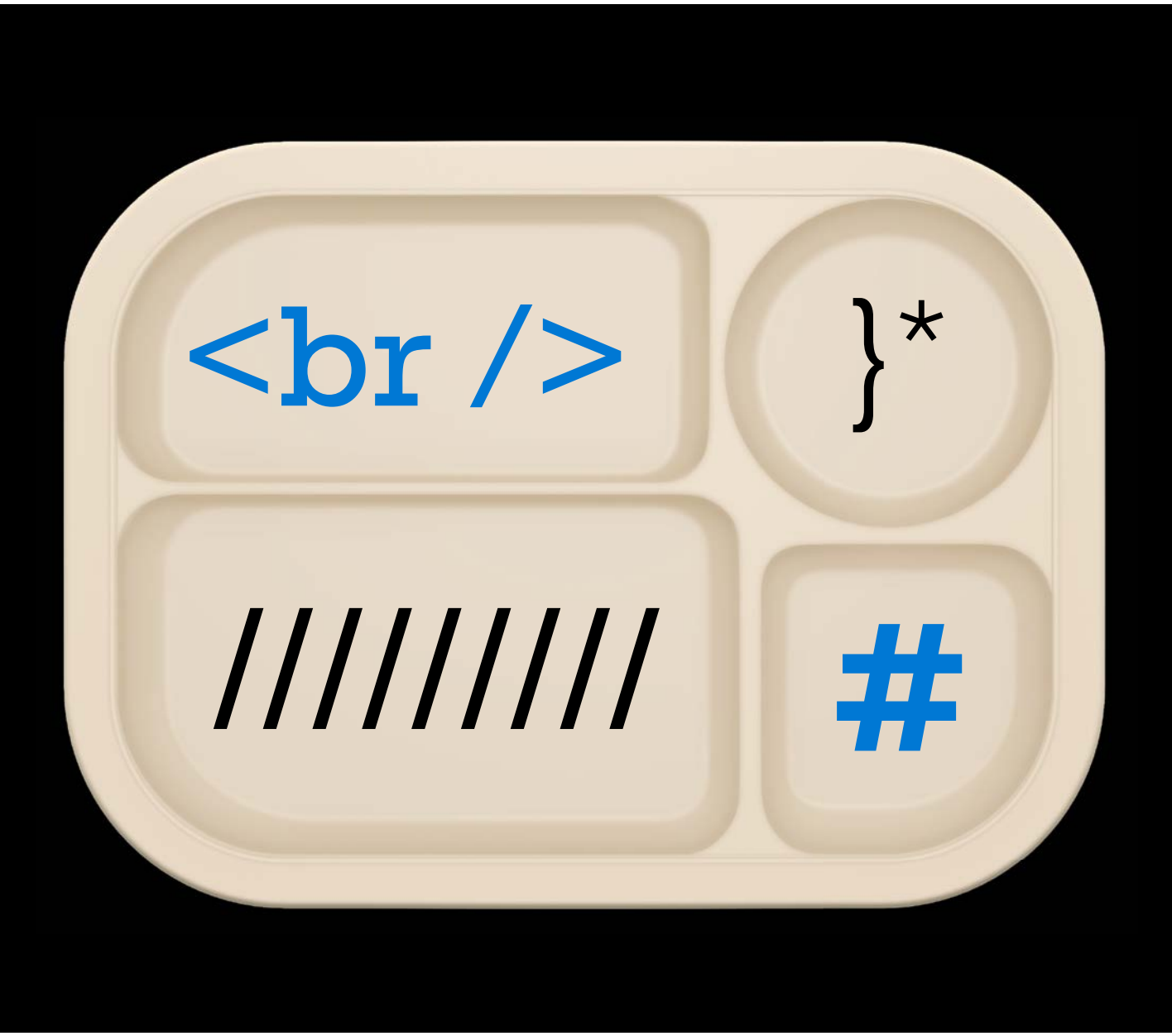
APRENDA A USAR O AZURE EM UM MÊS DE AULAS

Segunda Edição
Iain Foulds



startHere(Azure);

Explore 21 lições do Azure

Obtenha uma base sólida no Azure com as lições deste e-book. Inscreva-se para uma conta gratuita do Azure e use seu crédito de USD 200 para concluir os exercícios. Continue com sua conta e receba 12 meses de serviços populares gratuitos e mais de 25 serviços sempre gratuitos.

Elogios para a primeira edição

Desde a primeira edição do *Aprenda a usar o Azure em um mês de aulas* por Iain Foulds:

"Um livro incrível e repleto de informações para aprender conceitos básicos e avançados do Azure em um mês!"
— Sushil Sharma, Galvanize

"O Microsoft o Azure está se tornando rapidamente um líder no espaço de nuvem pública. Com os exercícios deste livro, você aprenderá rapidamente a usar essa tecnologia."
— Michael Bright, Consultor Freelancer de Desenvolvimento

"Excelente introdução ao Azure com muitos exemplos práticos. Abrange uma ampla gama de tópicos atuais."
— Sven Stumpf, ING-DiBa AG

"O Azure é como um oceano. Este livro é a melhor maneira de aprender diariamente com muitos exercícios práticos e exemplos."
— Roman Levchenko, Microsoft MVP

"Tudo que um desenvolvedor ocupado precisa começar a usar o Azure."
— Rob Loranger, Desenvolvedor Freelancer

"Uma ótima maneira de entender a amplitude das ofertas do Azure, seguindo uma abordagem concisa e focada em atividades."
— Dave Corun, Avanade

"O livro mais abrangente sobre o Azure que encontrei para começar a desenvolver meus projetos acadêmicos!"
— Marco Giuseppe Salafia, aluno de doutorado, Università degli Studi di Catania

"Este é o melhor livro sobre a plataforma do Azure. Ele é bem organizado, minucioso e abrangente. Começando com o básico, ele orienta o leitor por meio da criação de configurações cada vez mais complexas com a plataforma do Azure para fornecer escalabilidade, alta performance e redundância para aplicações e serviços hospedados. Este livro servirá como um tutorial para iniciantes e uma referência para usuários mais experientes."
— Robert Walsh, Excalibur Solutions

Aprenda a usar o Azure em um mês de aulas

SEGUNDA EDIÇÃO

IAIN FOULDS

MANNING
SHELTER ISLAND

Para obter informações online e fazer pedidos desse e de outros livros da Manning, acesse www manning.com. A editora oferece desconto neste livro quando solicitado em quantidade. Para obter mais informações, entre em contato com

Departamento de vendas especiais
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Editor de aquisições:	Mike Stephens
Editor de desenvolvimento:	Frances Lefkowitz
Editor de desenvolvimento técnico:	Karsten Strøbaek
Editor de revisão:	Aleksandar Dragosavljevic
Editor de produção:	Anthony Calcara
Editor gráfico:	Jennifer Houle
Editor de texto:	Kathy Simpson
Revisor:	Katie Tennant
Editor técnico:	Karsten Strøbaek
Tipógrafo:	Marija Tudor
Designer de capa:	Leslie Haimes

ISBN 9781617297625
Impresso nos Estados Unidos da América

Para as pessoas mais importantes da minha vida:
Abigail, Bethany e Charlotte

sumário

prefácio

Esta segunda edição do *Aprenda a usar o Azure em um mês de aulas* me lembra que as coisas mudam rapidamente e que você deve sempre continuar aprendendo. Foi-se o tempo em que você podia fazer um curso de uma semana sobre o Windows Server e executá-lo confortavelmente por anos sem muitas mudanças. Isso não significa que o mundo da TI é um lugar mais assustador, mas você precisa abordar a computação na nuvem com uma mente aberta e estar disposto a se ajustar constantemente.

Quando comecei a trabalhar com o Azure, o número de serviços disponíveis era quase espantoso. Eu sabia que deveria me atentar à segurança, à performance, à redundância e à escala, mas não sabia como adaptar mais de uma década de administração de servidores em larga escala ao mundo da computação em nuvem. Com o passar do tempo, comecei a aprender sobre os vários serviços do Azure que fornecem esses componentes-chave. Esses serviços raramente funcionam isoladamente, mas não sabia a melhor maneira de integrá-los ou como decidir qual serviço usar para cada tarefa. Este livro é uma maneira de explicar ao meu eu antigo, e a muitos outros que seguem um caminho semelhante, como entender rapidamente os principais serviços do Azure e fazê-los funcionar juntos.

Este livro tem mais de 350 páginas, mas apenas aborda o básico do que é possível fazer no Azure! Para ajudar a fornecer uma compreensão sólida dos conceitos necessários para obter êxito enquanto você cria soluções no Azure, tive que escolher quais tópicos escrever a respeito. O livro não aborda todos os 100 ou mais serviços do Azure e não detalha de forma abrangente os serviços incluídos. Em vez disso, ele se concentra nas áreas essenciais de alguns dos serviços principais, mostra exemplos de como conectar tudo com segurança e introduz as possibilidades do que você pode criar no Azure.

A computação na nuvem está mudando constantemente. Não há ciclos de lançamento de três ou quatro anos nem grandes implantações de atualização. Acho que hoje é um ótimo momento para criar soluções e escrever códigos; sempre há uma oportunidade de aprender algo novo e se aperfeiçoar. Espero que você aprenda a executar ótimas aplicações no Azure e aproveite para explorar todos os serviços disponíveis.

agradecimentos

Muitas pessoas nos bastidores da Manning Publications ajudaram a publicar este livro. Agradecimentos especiais a Mike Stephens por ter iniciado este projeto. Agradeço à minha editora, Marjan Bace, e a todos das equipes editoriais e de produção. Meus agradecimentos vão para os colegas revisores técnicos, liderados por Aleksandar Dragosavljević — Ariel Gamino, Charles Lam, Ernesto Cardenas Cangahuala, George Onofrei, Glen Thompson, Jose Apablaza, Juraj Borza, Michael Langdon, Michael Wall, Peter Kreyenhop, Rick Oller, Rob Ruetsch, Robert Walsh e Vishal Singh. E finalmente, na parte técnica, agradeço Karsten Strøbaek, que trabalhou como editor e revisor técnico do livro.

Para esta segunda edição, agradeço Phil Evans e Davanand Bahall pelo apoio e pela liberdade para atualizar este livro. Este foi um projeto fora do meu trabalho na Microsoft, mas muitas pessoas adoraram. Agradeço David Tolkov e Tim Teebken, que me deram oportunidades de me tornar alguém capaz de escrever este livro. E veja, Jean-Paul Connock, ganhamos uma Stanley Cup desde a última vez! Vai, time!

Agradeço Rick Claus por ajudar na necessidade de uma documentação técnica sólida no Azure e a Marsh Macy e Neil Peterson pelo apoio e orientação pessoais na criação da versão original deste livro. Ainda precisamos falar sobre o ônibus escolar.

Sobre este livro

Este livro foi criado para fornecer uma base sólida para ser bem-sucedido como desenvolvedor ou engenheiro de TI no Azure. Você aprenderá sobre as soluções de Infraestrutura como serviço (IaaS) e de Plataforma como serviço (PaaS) e também sobre o momento de usar cada abordagem. Ao analisar os capítulos, você aprenderá a planejar adequadamente a disponibilidade e a escala, manter a segurança em mente e considerar o custo e a performance. No final do livro, você deverá ser capaz de integrar as próximas tecnologias, como contêineres e Kubernetes, inteligência artificial e aprendizado de máquina (IA + ML) e a Internet das Coisas (IoT).

Quando se trata de criar e executar aplicações e serviços, o Azure permite escolher o sistema operacional, as ferramentas de aplicações e a plataforma com a qual você se sente mais à vontade. Este livro discute principalmente tecnologias que não são da Microsoft, como Linux, Python e Node.js. Os exemplos de comando usam a CLI do Azure, não o Azure PowerShell. Essas foram decisões conscientes para mostrar que usar o Azure não significa que você precisa usar o Windows Server, o IIS ou o ASP.NET.

Como você trabalha na nuvem, você geralmente trabalha em várias plataformas e precisa aprender novos tópicos, o que é outro motivo para mostrar tecnologias e plataformas que não são da Microsoft. Eu gostaria de apresentar algumas dessas novas áreas antes de se deparar com elas no mundo real. Por meio do livro, tentarei ensinar os conceitos e as etapas necessárias para integrar os serviços do Azure, para que você possa alternar as plataformas ou os idiomas que desejar e ter o mesmo conhecimento aplicado.

Roteiro

O livro está organizado em 4 partes e 21 capítulos:

- A parte 1 aborda alguns dos principais serviços de infraestrutura e plataforma do Azure: máquinas virtuais, aplicativos Web, armazenamento e redes.

- A parte 2 se aprofunda em como fornecer alta disponibilidade e redundância: modelos, conjuntos de disponibilidade e zonas, balanceadores de carga, escalonamento automático, bancos de dados distribuídos e roteamento de tráfego. No final do capítulo 12, você deverá ter um sólido conhecimento de como criar aplicações distribuídas de alta performance no Azure.
- A parte 3 aborda aspectos de segurança, como backup e recuperação, criptografia, gerenciamento de chaves digitais e atualizações. No momento em que você concluir o capítulo 16, estará preparado para criar aplicações seguras e estáveis no Azure.
- Para terminar o livro, a parte 4 apresenta um pouco de diversão, explorando novas áreas de computação, como computação sem servidor e aplicações baseadas em contêineres. Esses capítulos apresentam áreas do Azure que oferecem um resumo de como seria o futuro das aplicações de produção.

Além da parte 4, que é apropriadamente chamada de "As coisas legais", você deve tentar analisar os capítulos do livro em ordem. Você não trabalha no mesmo projeto em capítulos sucessivos, mas cada capítulo se baseia em exemplos anteriores de teoria e de laboratório prático.

O capítulo 1 orienta você na criação de uma conta de trial gratuita no Azure, o que é suficiente para concluir os exercícios de laboratório prático em cada capítulo. Eu também forneço um pouco mais de conhecimento sobre o Azure e como encontrar ajuda adicional ao longo do processo. Menciono esta página da Web algumas vezes no livro (talvez eu seja um pouco tendencioso!), mas http://docs.microsoft.com/azure é o melhor lugar para obter documentação e suporte adicionais em quaisquer áreas do Azure que interessem a você.

Sobre os exemplos e o código-fonte

Este livro contém muitos exemplos de código-fonte, tanto em listagens numeradas quanto em linha com o texto normal. Nos dois casos, o código-fonte é formatado em uma `fonte de largura fixa como essa` para separá-lo do texto comum.

Em muitos casos, o código-fonte original foi reformatado, com quebras de linha e recuo reformulado adicionados para acomodar o espaço de página disponível no livro. Em casos raros, até mesmo isso não era suficiente, e os anúncios incluem marcadores de continuação de linha (➥). Além disso, os comentários no código-fonte são removidos das listagens quando o código é descrito no texto. Anotações de código acompanham muitas das listagens, destacando conceitos importantes.

O código-fonte deste livro, junto com scripts, modelos e recursos de suporte que o acompanham, está disponível em https://www.manning.com/books/learn-azure-in-a-month-of-lunches-second-edition e no repositório GitHub do livro (https://github.com/fouldsy/azure-mol-samples-2nd-ed).

Todos os exercícios práticos podem ser concluídos no portal do Azure e com o Azure Cloud Shell, um shell interativo baseado em navegador para a CLI do Azure e o Azure PowerShell. Não há ferramentas a serem instaladas na sua máquina e você pode usar qualquer computador e sistema operacional que desejar, desde que suporte um navegador da Web moderno.

Geralmente, o portal do Azure implementa pequenas alterações. Parte do desafio de usar qualquer serviço de nuvem é que as coisas podem ser um pouco diferentes do que eram no dia anterior.

Esta segunda edição do livro tenta minimizar o número de capturas de tela do portal, mas não se preocupe se o conteúdo ainda for um pouco diferente do que é mostrado no livro. Os parâmetros necessários são geralmente os mesmos, o layout pode ser diferente. Se houver novas opções no portal que eu não informei especificamente em um exercício ou laboratório, geralmente é seguro aceitar os padrões fornecidos.

Se você trabalha fora do Azure Cloud Shell, tenha cuidado com os exemplos de comando. Os shells baseados no Windows, como o PowerShell e o CMD, tratam as quebras de linha e as continuações de maneira diferente dos shells baseados em *nix, como o Azure Cloud Shell. Muitos dos exemplos de comando são executados em várias linhas. Os comandos são mostrados com um caractere de barra invertida (\) para indicar que o comando continua na próxima linha, como no exemplo a seguir:

```
az resource group create \
--name azuremol \
--location eastus
```

Você não precisa digitar esses caracteres de barra invertida, mas isso pode tornar os comandos longos mais legíveis na tela. Se você optar por trabalhar localmente em seu computador com um shell do Windows, poderá usar um acento grave (`) em vez de uma barra invertida. Por exemplo, em um shell do PowerShell ou do CMD com o Python para Windows instalado, altere o comando anterior da seguinte maneira:

```
az resource group create `
--name azuremol `
--location eastus
```

Essa convenção pode parecer confusa no início, mas eu sigo essa convenção no livro porque a documentação oficial em https://docs.microsoft.com/azure usa esse formato. Os comandos da CLI do Azure, que são o que mais usamos neste livro, assumem um shell baseado em *nix e, portanto, usam um caractere de barra invertida. Os comandos do Azure PowerShell assumem um shell baseado no Windows e, portanto, usam um sinal de acento grave. Essa diferença no comportamento fará sentido rapidamente, e você perceberá que é fácil fazer a transição entre os shells. Se você nunca trabalhou em plataformas, essa diferença poderá ser uma pegadinha divertida!

Recomendo conferir o Windows Subsystem for Linux (WSL), se você usa o Windows 10 e quer se aprofundar nos sistemas Azure CLI e baseados em *nix em geral. Veja detalhes em https://docs.microsoft.com/windows/wsl. O WSL e as mais recentes melhorias no WSL2 proporcionam uma experiência nativa do kernel Linux enquanto executa o Windows. Não tente entender muito essa ideia. Apenas saiba que você pode executar comandos e aplicações nativos do Linux sem se preocupar com quebras de linha ou definições de variáveis diferentes. Para realmente impressionar você, o PowerShell está disponível para o .NET Core, que também é executado no Linux. Você pode executar o PowerShell no Linux enquanto estiver no Windows.

Fórum de discussão do LiveBook

A compra do livro Aprenda a usar o Azure em um mês de aulas inclui o acesso gratuito a um fórum da Web privado administrado pela Manning Publications, onde você pode fazer comentários sobre o livro, fazer perguntas técnicas e receber ajuda do autor e de outros usuários. Para acessar o fórum, visite https://livebook.manning.com/book/learn-azure-in-a-month-of-lunches-second-edition/discussion. Você também pode aprender mais sobre os fóruns da Manning e as regras de conduta em https://livebook.manning.com/discussion.

O compromisso da Manning com nossos leitores é fornecer um local onde possa ocorrer um diálogo significativo entre os diferentes leitores e entre os leitores e o autor. Não é um comprometimento com um grau de participação específico por parte do autor, cuja contribuição para o fórum permanece voluntária (e não remunerada). Sugerimos que você tente fazer algumas perguntas desafiadoras ao autor, para que ele não perca o interesse! O fórum e os arquivos das discussões anteriores estarão acessíveis no site da editora, contanto que o livro ainda esteja sendo comercializado.

Sobre o autor

IAIN FOULDS é desenvolvedor sênior de conteúdo na Microsoft, atualmente escrevendo documentações técnicas para o Azure Active Directory. Antes, ele era engenheiro de campo de primeira linha na Microsoft voltado para tecnologias de virtualização, como Azure, Hyper-V e System Center Virtual Machine Manager. Com mais de 15 anos de experiência em TI, a maioria em operações e serviços, Iain aderiu à virtualização inicialmente com a VMware e ajudou a desenvolver e ensinar outras pessoas sobre computação em nuvem por anos.

Nascido na Inglaterra, ele vive nos Estados Unidos há mais de uma década e atualmente mora fora de Seattle com sua esposa e duas filhas pequenas, às quais este livro é dedicado. Ele é fã de futebol e também gosta de hóquei no gelo e quase qualquer tipo de automobilismo. Além da computação, os interesses de Iain incluem carros de performance e clássicos, fotografia de aviação e também alega tocar violão. Ele também é um grande nerd em miniaturas de trens, participa e trabalha regularmente como voluntário em programas e eventos em todo o noroeste do Pacífico.

Parte 1

Principais serviços do Azure

Para criar a próxima grande aplicação, você precisa de uma compreensão sólida dos recursos básicos no Azure. Coisas como armazenamento e rede podem não ser os aspectos mais legais para analisar, mas elas são fundamentais para o que você executa no Azure. Antes de começar a entrar em máquinas virtuais redundantes de várias instâncias ou em aplicativos Web do Azure, ela ajuda a ver as opções e as tarefas de gerenciamento disponíveis para uma única instância. Essa abordagem permite que você aprenda sobre as diferenças e semelhanças entre a abordagem IaaS de VMs e a abordagem PaaS de aplicativos Web. Capítulos 1 a 5: exploraremos VMs e aplicativos Web, além de recursos de armazenamento principal e rede virtual.

1 Antes de começar

O Azure é um dos maiores provedores de computação na nuvem pública para serviços como máquinas virtuais (VMs), contêineres, computação sem servidor e machine learning. Nós não abordaremos os 100 ou mais serviços do Azure neste livro, mas você vai aprender sobre os principais serviços e recursos que abrangem a maior parte do que você precisa para iniciar a criação e execução de soluções no Azure. Vamos examinar um exemplo comum de como criar e executar um aplicativo Web, e você verá como usar alguns dos principais serviços de infraestrutura e plataforma que podem facilitar seu trabalho.

Com o Azure, você não precisa de uma varinha mágica para prever quantos servidores ou quanto de armazenamento é necessário nos próximos três anos. Chega de atrasos até obter aprovação do orçamento, aguardar o recebimento do novo hardware e, em seguida, colocar no rack, instalar e configurar tudo. Você não precisa se preocupar sobre quais versões de software ou bibliotecas são instaladas enquanto grava o código.

Em vez disso, selecione um botão e crie quaisquer recursos necessários. Você só paga por minuto pelos recursos que estão em execução ou pela quantidade de espaço de armazenamento ou largura de banda de rede usada. Quando você não precisar mais dos recursos, poderá desativá-los ou exclui-los. Se de repente você precisar aumentar a quantidade de potência de computação por um fator de 10, selecione um botão, aguarde alguns minutos e pronto. Tudo isso é gerenciado por outra pessoa, liberando você para se concentrar em seus clientes e aplicações.

1.1 Este livro é para você?

A indústria de TI está em um período de transição quando se trata de títulos de trabalho. Você pode se referir a si mesmo como um profissional de TI, um desenvolvedor de software, um administrador de sistema ou um engenheiro de DevOps. Independentemente de como você se vê, se quiser aprender as habilidades básicas

necessárias para criar e executar aplicações seguras e altamente disponíveis na nuvem, você está no lugar certo. Em termos genéricos, você provavelmente cairá nas operações de TI ou no lado de desenvolvimento das coisas. A verdade é que há muita sobreposição, especialmente em computação na nuvem. Esteja você na equipe de desenvolvimento ou operações, é importante compreender os principais serviços de infraestrutura e plataforma para criar e executar as aplicações mais adequadas para seus clientes.

A segunda edição deste livro introduz alguns desses conceitos fundamentais no Azure e ensina a você as habilidades necessárias para tomar decisões informadas. Antes de ler este livro, você deve ter alguma experiência prévia com VMs e saber as noções básicas de rede e armazenamento. Você também deve ser capaz de criar um site básico e entender o que é um certificado SSL e um banco de dados. Depois de cobrir os processos principais, vamos analisar rapidamente as tecnologias novas e futuras. Você quer ficar à frente de onde seu trabalho pode levá-lo para aprender sobre contêineres, a Internet das Coisas, aprendizado de máquina, inteligência artificial e computação sem servidor. Tanto os desenvolvedores quanto os profissionais de TI devem encontrar algumas áreas novas para aprender!

1.2 Como usar este livro

Eu gosto de sanduíches, então o almoço é um ótimo momento para eu explorar a nova tecnologia legal. Você pode ser uma coruja noturna que tem algum tempo extra à noite ou uma pessoa que gosta de acordar cedo (o que há de errado com você?) que pode ler um capítulo no café da manhã. Não há hora certa ou errada para aprender, mas se reservar cerca de 45 minutos, você conseguirá ler um capítulo e fazer os exercícios. Cada capítulo abrange algo novo, por isso dê-se tempo para absorver a lição de cada dia.

1.2.1 Os capítulos principais

O livro é dividido em quatro partes, o que é conveniente pensando que um mês tem quatro semanas:

- A Parte 1 (capítulos 1 a 5) aborda alguns dos principais recursos do Azure. Se nada mais funcionar, tente seguir esses capítulos para ter uma compreensão sólida. Então, você poderá se concentrar nos outros capítulos que mais lhe interessam.
- A Parte 2 (capítulos 6 a 12) abrange disponibilidade e escala. Você aprenderá a dimensionar automaticamente os recursos, o tráfego de balanceamento de carga e a manipular eventos de manutenção sem tempo de inatividade. Se quiser saber mais sobre a execução de aplicações altamente disponíveis em escala global, esta parte é para você.
- A Parte 3 (capítulos 13 a 16) é para os geeks de segurança. Ela abrange coisas como criptografar VMs, armazenar certificados SSL em um cofre seguro e fazer backup e restaurar seus dados.
- A Parte 4 (capítulos 17 a 21) abrange uma mistura de áreas interessantes para dar um gostinho do que o Azure pode fazer por você e seus clientes. Vamos falar sobre automação, contêineres, a Internet das Coisas e computação sem servidor. Escolha algo que lhe interessa, e divirta-se!

1.2.2 Experimente agora

Você quer apenas ler ou quer arregaçar as mangas e brincar com o Azure? Ao longo do livro, pequenas tarefas permitem que você tente algo novo rapidamente. Se tiver tempo, experimente. A maioria da prática vem em um exercício de laboratório no final do capítulo, mas é muito importante dividir a leitura, tentando novos conceitos ao longo do caminho. Alguns desses exercícios orientam você passo a passo, outros vão fazer você pensar um pouco mais e aprender a criar soluções sozinho, como faria no mundo real.

1.2.3 Laboratórios práticos

Cada capítulo termina com um exercício de laboratório prático. Alguns capítulos, como este, têm um exercício de laboratório no meio do capítulo. Nesses exercícios de laboratório, você aprende como todas as peças do Azure se encaixam e pode começar a construir alguma memória muscular mental. Pegue o teclado e o mouse e comece a criar algo incrível!

1.2.4 Código-fonte e materiais suplementares

O código-fonte deste livro, junto com scripts, modelos e recursos de suporte que o acompanham, podem ser encontrados em https://www.manning.com/books/learn-azure-in-a-month-of-lunches-second-edition e no repositório do GitHub do livro em https://github.com/fouldsy/azure-mol-samples-2nd-ed. Além disso, você pode participar do fórum do livro em https://livebook.manning.com/book/learn-azure-in-a-month-of-lunches-second-edition/discussion.

1.3 Criar seu ambiente de laboratório

Este livro não exagera em conceitos e arquitetura. Ele fala mais sobre o tempo prático com a plataforma do Azure. Para fazer isso, você precisa de uma conta do Azure.

1.3.1 Criar uma conta gratuita do Azure

O Azure oferece uma conta trial gratuita que pode ser usada por 30 dias e fornece até USD 200 de crédito gratuito. Esse crédito gratuito deve ser suficiente para ler todos os capítulos e exercícios, com espaço para explorar um pouco e se divertir ao longo do caminho. Há também muitos serviços e recursos do Azure que permanecem gratuitos, mesmo após o término do período de avaliação.

Experimente agora

Siga as etapas desta seção para criar sua conta gratuita do Azure:

1. Abra o navegador da Web https://azure.microsoft.com/free e escolha a opção para começar com uma conta gratuita do Azure.
2. Quando solicitado, entre em sua conta da Microsoft. Se você precisar de uma conta da Microsoft ou quiser criar uma nova, escolha o link Criar uma Nova Conta da Microsoft.

3. Uma vez conectado a uma conta da Microsoft, conclua os prompts para criar uma conta gratuita do Azure:
   - Insira seus dados pessoais conforme solicitado.
   - Para ajudar a minimizar o abuso e a fraude, forneça um número de telefone para confirmar sua identidade por mensagem de texto ou ligação.
   - Um cartão de crédito também é necessário para a verificação de identidade, mas não há cache aqui. Sua conta só começa a faturar depois de 30 dias ou quando você usar seu crédito de USD 200. A transição para uma assinatura pré-paga não será feita automaticamente no final do trial. Você pode ver um pequeno valor de verificação de USD 1 (ou o equivalente na moeda local), que será reembolsado em alguns dias.
4. Revise e aceite o contrato de assinatura do Azure e a política de privacidade e selecione Inscrever-se. Pode demorar alguns minutos para que sua assinatura do Azure seja preparada.
5. Depois que o processo de inscrição terminar e o portal do Azure for carregado, faça o tour rápido para se familiarizar.

O painel — a página inicial do portal — parece estar vazio neste momento. Entretanto, no capítulo 2, você mergulhará na criação de sua primeira VM, que começará a ficar como mostra a Figura 1.1.

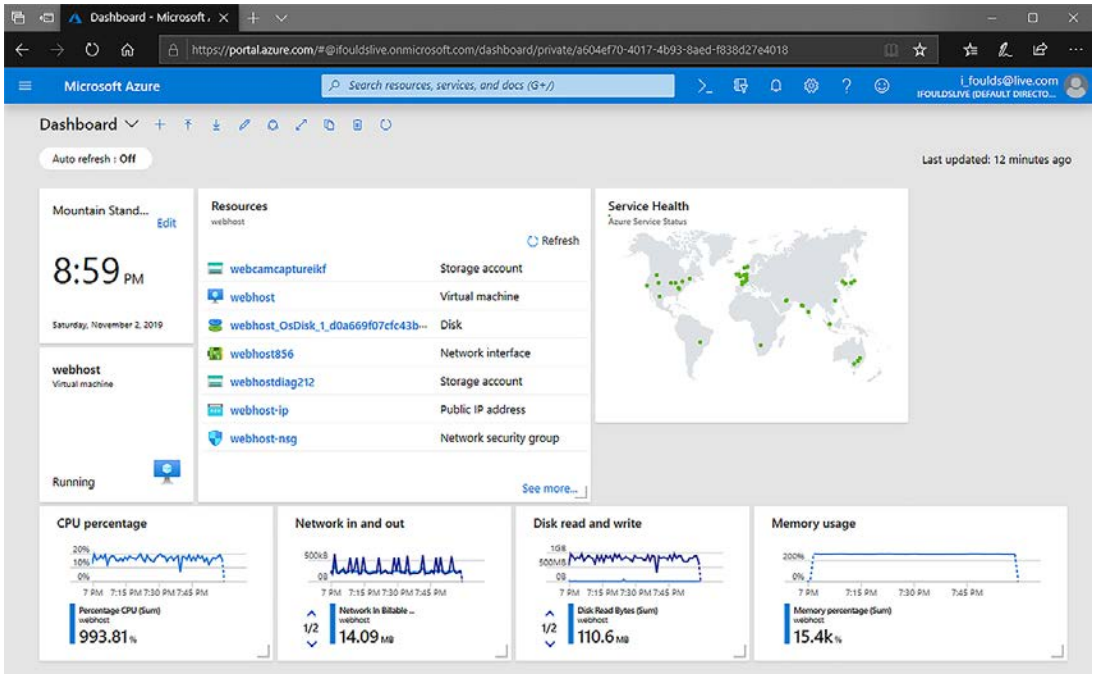


Figura 1.1 O portal do Azure, pronto para criar suas próprias aplicações e soluções

É realmente gratuito?

O Azure tem um Marketplace que contém centenas de imagens pré-construídas (a base de VMs) e soluções que você pode implantar. Usaremos algumas dessas ofertas do Marketplace em todo o livro. Elas são ótimas para implantar um pacote de aplicações inteiro.

Nem todas as ofertas do Azure Marketplace são gratuitas. Alguns editores de terceiros combinam os custos de licenciamento ou de suporte na solução implantada. Por exemplo, uma VM que você implanta da Red Hat pode incorrer em uma taxa adicional que cobre o contrato de suporte e a licença da Red Hat. Esses encargos não são cobertos pelo seu crédito de avaliação gratuito; somente o uso da VM base é coberto.

Os exercícios neste livro só usam recursos que permanecem dentro da avaliação gratuita. Mas, se você explorar outras ofertas interessantes do Marketplace no Azure, preste atenção no que criar. Qualquer solução que inclua taxas adicionais deve indicar claramente as taxas antes da implantação!

1.3.2 Exercício de laboratório de bônus: criar uma conta gratuita do GitHub

O GitHub é um serviço da Web gratuito que muitas organizações e indivíduos usam para gerenciar projetos como código, modelos e documentação. O Azure tem centenas de modelos gratuitos e exemplos de script que você pode usar e com os quais pode contribuir. Este é um dos pontos fortes da comunidade de open source: compartilhar e retribuir.

Alguns dos exercícios neste livro usam recursos do GitHub. Você não precisa de uma conta do GitHub para fazer nada disso. No entanto, se você não tiver uma conta, não poderá salvar nenhuma modificação e começar a criar sua própria coleção de modelos e scripts. Criar uma conta do GitHub é opcional, mas é uma parte altamente recomendada da criação de seu ambiente de laboratório:

1. Abra seu navegador da Web para https://github.com.
2. Para criar uma conta gratuita do GitHub, insira um nome de usuário, endereço de email e senha. Você receberá uma mensagem de validação do GitHub.
3. Selecione o link no email de validação para ativar sua conta.
4. Confira alguns dos repositórios do Azure que fornecem recursos de exemplo:
   - Modelos de início rápido do Azure: https://github.com/Azure/azure-quickstart-templates
   - Azure CLI: https://github.com/Azure/azure-cli
   - Utilitários do Azure DevOps: https://github.com/Azure/azure-devops-utils
   - *Aprenda a usar o Azure em um mês de aulas* recursos do livro: https://github.com/fouldsy/azure-mol-samples-2nd-ed

1.4 Uma pequena ajuda

Este livro não cobre tudo o que o Azure oferece. Mesmo se tentasse, no momento em que ler este capítulo, eu aposto que haverá algo novo no Azure. A computação na nuvem se move rapidamente, e novos serviços e recursos estão sempre sendo liberados. Eu posso ser um pouco

suspeito, mas quando você começar a explorar o Azure e quiser aprender sobre serviços adicionais, o site https://docs.microsoft.com/azure excelente é o melhor lugar para começar. Cada serviço do Azure está documentado com exemplos de início rápido, tutoriais, exemplos de código, referências de desenvolvedor e guias de arquitetura. Você também pode acessar opções de suporte gratuitas e pagas se precisar de ajuda ao longo do caminho.

1.5 Noções básicas sobre a plataforma do Azure

Antes de entrar no restante deste livro, vamos dar um passo atrás e entender o que é o Azure e os serviços que estão disponíveis. Como mencionei anteriormente, o Azure é um provedor de computação na nuvem em escala global. Até o momento da redação deste documento, havia 54 regiões do Azure. Cada região contém um ou mais data centers. Por comparação, os outros dois grandes provedores de nuvem operam em 23 regiões (Amazon Web Services [AWS]) e 20 regiões (Google Cloud).

A computação na nuvem fornece mais do que apenas recursos de computação. O Azure tem mais de 100 serviços, agrupados em famílias de serviços relacionados, como computação, Web + rede móvel, contêineres e identidade. Com todos esses serviços, o Azure abrange muitos modelos de serviço. Vamos pegar uma fatia de pizza para o almoço para entender o que isso significa: veja a (Figura 1.2).

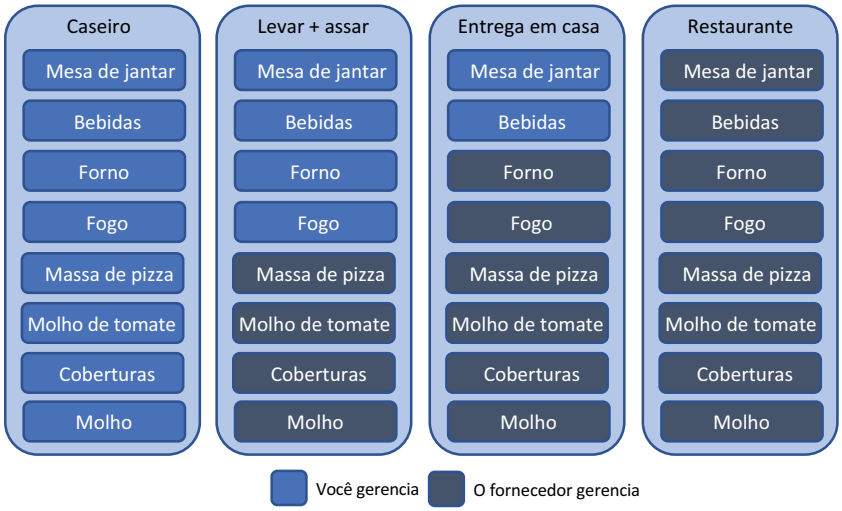


Figura 1.2 Modelo de Pizza como um serviço. À medida que passa do modelo de pizza caseira, onde você fornece tudo, para o modelo de restaurante, onde você acabou de aparecer, as responsabilidades e as demandas de gestão mudam conforme necessário.

No modelo de Pizza como um serviço, há quatro opções para escolher. Ao explorar os modelos, você se preocupa cada vez menos com o processo de comer uma fatia de pizza:

- *Caseira*: você faz a massa, acrescenta o molho, coberturas e queijo, assa a pizza em seu forno, compra as bebidas e senta para comer em sua mesa de jantar.

- *Comprar + assar*: você compra uma pizza pronta. Você só precisa assá-la no forno, *comprar as bebidas e sentar para comer em sua mesa de jantar.*
- *Entrega em casa*: você pede uma pizza para entregar em casa. Você só precisa comprar as bebidas e sentar para comer em sua mesa de jantar.
- *Restaurante*: você quer sair e comer pizza com o mínimo de esforço!

Agora que você está com fome, vamos olhar para o modelo mais tradicional que envolve alguns recursos de computação (Figura 1.3). Esse modelo é um pouco mais parecido com o que você vê no Azure.

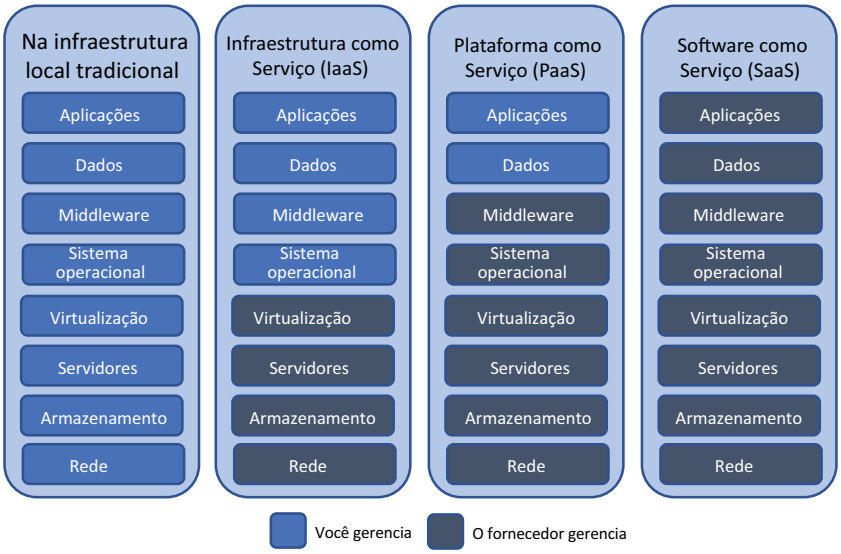


Figura 1.3 Modelo de serviço de computação na nuvem

À medida que explora os modelos, você gerencia menos os recursos subjacentes e pode concentrar mais tempo e energia nos seus clientes:

- *Na infraestrutura local*: você configura e gerencia todo o data center, como os cabos de rede, o armazenamento e os servidores. Você é responsável por todas as partes do ambiente de aplicação, suporte e redundância. Essa abordagem fornece o controle máximo, mas com muita sobrecarga de gerenciamento.
- *Infraestrutura como serviço (IaaS)*: adquire os recursos de computação básicos de um fornecedor que gerencia a infraestrutura principal. Você cria e gerencia as VMs, os dados e as aplicações. O provedor de nuvem é responsável pela infraestrutura física, pelo gerenciamento de host e pela resiliência. Você ainda pode ter uma equipe de infraestrutura para ajudar a oferecer suporte e implantar VMs, mas eles estão livres do tempo e do custo de gerenciar o equipamento físico.

Essa abordagem é boa quando você começa a mover aplicações para fora do seu próprio ambiente na infraestrutura local. O gerenciamento e as operações são geralmente semelhantes a um ambiente na infraestrutura local, de modo que a IaaS fornece uma progressão natural para os proprietários de negócios, TI e aplicações se sentirem confortáveis com a nuvem.

- *Plataforma como um serviço (PaaS)*: você adquire a pilha de plataforma subjacente de um fornecedor que gerencia o sistema operacional e os patches e traz seus dados e aplicações. Você não se preocupa com VMs ou com a rede virtual, e sua equipe de operações pode focar mais tempo na confiabilidade e na performance da aplicação.

  Com essa abordagem, geralmente a organização de TI e a empresa começam a se acostumar com a execução de aplicações na nuvem. Seu foco está nos aplicativos e em seus clientes, com menos preocupações sobre a infraestrutura necessária para executar esses aplicativos.
- *Software como serviço (SaaS)*: você só precisa de acesso ao software, com um fornecedor fornecendo todo o resto. Os desenvolvedores podem criar com uma plataforma existente para fornecer personalizações ou recursos exclusivos, sem ter que manter uma base de código grande.

  Essa abordagem geralmente é assustadora no início, mas você provavelmente já conhece e usa ofertas de SaaS bem-sucedidas, como Salesforce, Office 365 ou o pacote do Google de email ou documentos. Você usa email, cria documentos ou apresentações ou gerencia informações de contato do cliente e informações de vendas. Seu foco está no conteúdo que você cria e gerencia, não em como fazer a execução da aplicação.

A maior parte do que você cria no Azure se enquadra nas áreas de IaaS e PaaS. Os principais casos de uso incluem VMs e redes virtuais (IaaS) ou os serviços Aplicativos Web do Azure, Azure Functions e Cosmos DB (PaaS). Se você for desenvolvedor, as soluções de PaaS provavelmente são as áreas que mais lhe interessam, pois a Microsoft abrange as partes de infraestrutura para permitir que você se concentre no seu código. Os profissionais de TI podem inclinar-se mais para as soluções de IaaS para criar e controlar a infraestrutura do Azure.

Nunca pare de aprender

Não se esqueça que, mesmo conforme um negócio migra de IaaS para o modelo PaaS, o profissional de TI continua sendo relevante. É importante entender o que se passa por baixo da camada de PaaS quando você projeta ou resolve problemas de uma solução. Se você for profissional de TI, não ignore os capítulos sobre as soluções de PaaS no Azure. Há muita coisa que você pode adicionar à sua empresa e aos clientes se entender a transição para esse modelo de implantação.

1.5.1 Virtualização no Azure

A virtualização é a verdadeira magia por trás do Azure. Os modelos IaaS, PaaS e SaaS usam a virtualização para capacitar seus serviços. O conceito de virtualização não é novo, remontando aos dias de mainframe da década de 1960. Em meados da década de 2000, a virtualização de servidores no data center começou a ganhar impulso e, agora, apenas alguns workloads são implantados em servidores bare-metal em vez de serem virtualizados.

Livros inteiros são dedicados à virtualização, mas para dar a você uma visão geral rápida, a virtualização divide logicamente os recursos físicos em um servidor em recursos virtuais que podem ser acessados com segurança por workloads individuais. Uma VM é um dos recursos mais comuns de computação na nuvem. Uma VM contém uma CPU virtual (vCPU), memória (VRAM), armazenamento (vDisk) e conectividade de rede (vNIC), como mostra a Figura 1.4.

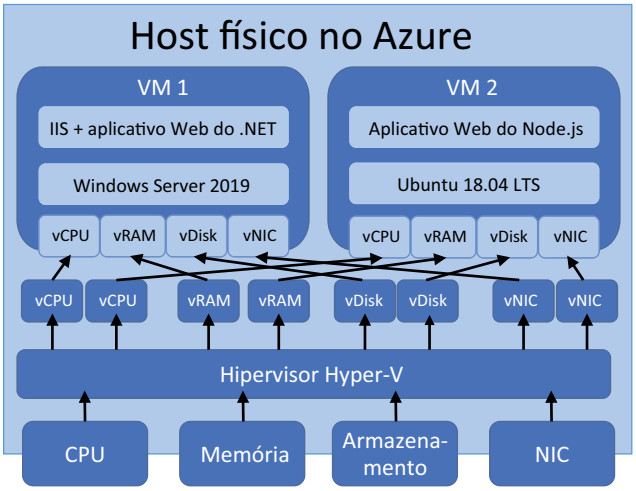


Figure 1.4 Virtualização em ação em um host físico no Azure

Além dos servidores físicos, o armazenamento e a rede são comumente virtualizados, o que permite que a plataforma do Azure defina rapidamente tudo que você precisa no software. Nenhuma interação física ou configuração manual de dispositivos é necessária. Não é necessário aguardar que outra equipe forneça um endereço IP, abra uma porta de rede ou adicione armazenamento para você.

Basicamente, o Azure é executado em um tipo de Windows. Uma versão modificada do hipervisor Hyper-V capacita os servidores de computação. O Hyper-V é um hipervisor tipo 1 (bare-metal) que está disponível no Windows Server há uma década. E não se preocupe, você ainda pode executar o Linux como um workload totalmente compatível de qualidade. A Microsoft contribui muito para o kernel e a comunidade Linux. Algumas das principais redes definidas por software no Azure são acionadas por uma solução personalizada baseada no Debian Linux — Software for Open Networking in the Cloud (SONiC) — que a Microsoft criou como open source. Você pode fazer um tour virtual dos data centers da Microsoft em https://azure.microsoft.com/global-infrastructure.

1.5.2 Ferramentas de gerenciamento

Com tantos serviços do Azure, como você os utiliza? Como você quiser! Se você quiser selecionar tudo em um navegador da Web, há um portal incrível baseado na Web. Confortável com o PowerShell? Como seria de esperar, há um módulo do Azure

PowerShell. Há também uma ferramenta de interface de linha de comando (CLI) entre plataformas que é ótima se você estiver no macOS ou Linux. E os desenvolvedores podem interagir com o Azure por meio de APIs REST usando uma variedade de linguagens comuns, como .NET, Python e Node.js.

PORTAL DO AZURE

O portal do Azure deve funcionar em qualquer navegador da Web moderno, e é uma maneira conveniente de usar o Azure sem instalar nada no seu computador. O portal também é uma ótima maneira de aprender a criar e gerenciar recursos vendo rapidamente uma representação visual de tudo.

Novos recursos e serviços estão sendo constantemente adicionados ao Azure, de modo que o portal pode mudar sempre ligeiramente do que você vê nas capturas de telas neste livro ou na documentação online e nos blogs. A nomenclatura de um botão pode mudar um pouco, ou uma nova opção pode ser adicionada, mas as operações principais permanecem todas iguais. Bem-vindo ao admirável mundo novo da computação na nuvem!

AZURE CLOUD SHELL

Se você deseja colocar as mãos no teclado e digitar comandos, o portal também inclui o Azure Cloud Shell, mostrado na Figura 1.5. Esse shell é um console interativo baseado na Web que fornece um shell bash, a CLI do Azure e algumas ferramentas de desenvolvimento de aplicações pré-instaladas, como Git e Maven. Há também uma versão do PowerShell do Cloud Shell que, como o nome indica, fornece acesso aos cmdlets mais recentes do Azure PowerShell.

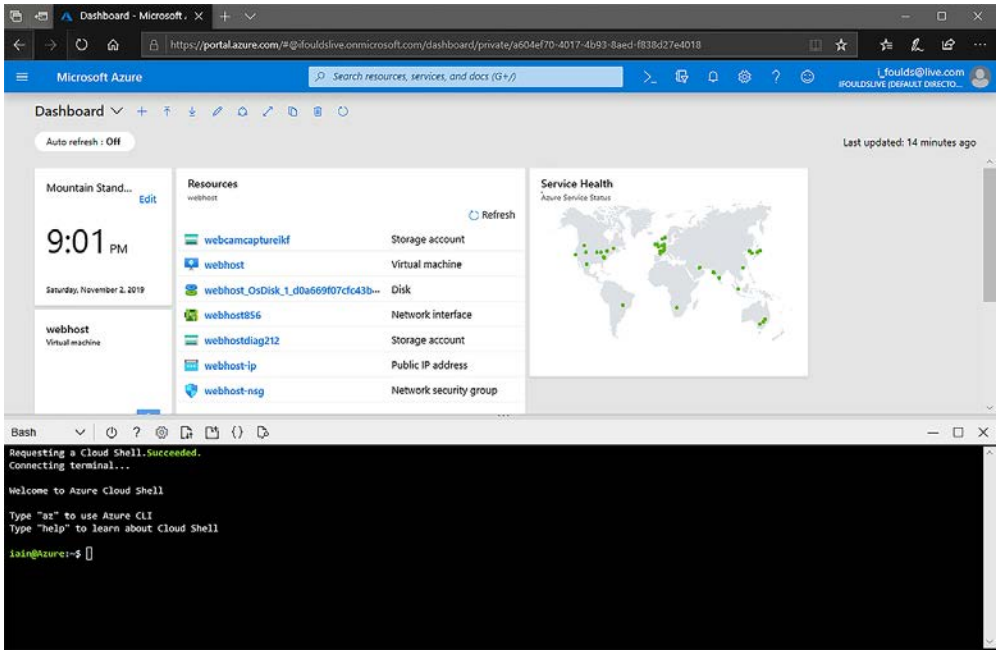


Figura 1.5 O Azure Cloud Shell no Portal baseado na Web

Você pode acessar o Azure Cloud Shell em um navegador da Web em qualquer computador sem precisar instalar nenhuma ferramenta em https://shell.azure.com. Editores como o Visual Studio Code (https://code.visualstudio.com) fornecem acesso ao Cloud Shell dentro da aplicação. Há até mesmo um aplicativo do Azure disponível para iOS e Android que permite que você use o Azure Cloud Shell diretamente do seu smartphone.

Com o Azure Cloud Shell, você sempre tem acesso à versão mais recente das ferramentas da CLI ou do PowerShell. O armazenamento persistente permite que você crie e salve scripts, modelos e arquivos de configuração.

Ferramentas locais do Azure CLI e PowerShell

Embora existam vantagens para o Azure Cloud Shell, você geralmente precisa de acesso ao seu sistema de arquivos e ferramentas locais. Você pode instalar a CLI do Azure ou o Azure PowerShell localmente para que possa trabalhar com recursos locais e recursos do Azure.

Neste livro, usamos principalmente a CLI do Azure (tecnicamente, a Azure CLI 2.0). Pode parecer estranho escolhê-la em vez do PowerShell nativo da Microsoft; a vantagem é que os exemplos e exercícios podem funcionar tanto no Azure Cloud Shell quanto localmente no seu computador, independentemente do sistema operacional que você usa. Embora isso não faça parte da configuração do ambiente de laboratório, os seguintes guias detalham como instalar as ferramentas de gerenciamento do Azure no seu computador:

- *Introdução ao Azure PowerShell*—https://docs.microsoft.com/powershell/azure/get-started-azureps
- *Instale a CLI do Azure*—https://docs.microsoft.com/cli/azure/install-azure-cli

2 Criar uma máquina virtual

Pronto para ver a rapidez com que você pode configurar um servidor Web no Azure? Neste capítulo, analisamos uma das solicitações mais comuns quando se trata de VMs: criar um servidor Web básico. Esse workload é um ótimo exemplo dos principais componentes de infraestrutura como um serviço (IaaS) no Azure.

Vamos supor que você trabalhe para uma pizzaria que quer expandir suas operações e aceitar pedidos online para entrega ou retirada de pizza. Para criar uma presença online, você precisa de um site. Nas primeiras partes deste livro, exploramos os diferentes recursos e serviços no Azure que permitem criar e executar aplicativos Web IaaS e PaaS. Você pode começar a tomar decisões informadas sobre quando criar e executar uma VM para alimentar um site, e quando você pode usar PaaS para fazer isso. Mas a primeira etapa é a criação de um servidor Web.

Neste capítulo, você cria uma VM do Ubuntu Linux e instala um servidor Web básico. Não se preocupe em usar o Linux; você criará uma VM do Windows no exercício de laboratório

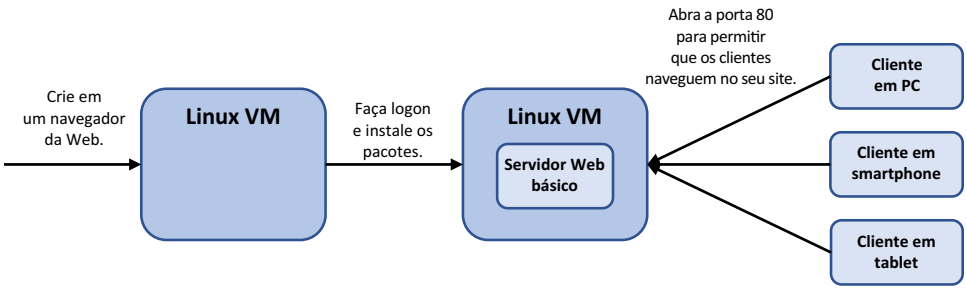


Figura 2.1 Neste capítulo, você cria uma VM básica, faz login para instalar um servidor Web e, em seguida, abre uma porta de rede para permitir que os clientes naveguem para o site de exemplo.

de fim do capítulo. Ubuntu é uma plataforma de servidor Web comum, e é uma ótima maneira de aprender sobre autenticação de chave pública SSH. Em seguida, você verá como abrir uma porta de rede para que os clientes acessem seu site na Internet. Uma visão geral de alto nível desse ambiente básico é mostrada na Figura 2.1.

2.1 Noções básicas sobre a configuração da máquina virtual

As VMs estão entre os elementos básicos mais comuns que você usará quando começar a executar aplicações na nuvem. Por quê? Elas geralmente são conhecidas. A maioria dos departamentos de TI executa muitos workloads usando o Hyper-V ou o VMware em um ambiente na infraestrutura local. Então, você provavelmente já tem alguma experiência na criação e execução de VMs. As organizações costumam dar os primeiros passos no Azure com VMs, pois os workloads de IaaS não exigem o grande esforço mental que você precisa fazer quando começa a executar workloads de PaaS.

Há soluções para migrar VMs de um ambiente na infraestrutura local, como Hyper-V ou VMware para o Azure, mas antes que você se empolgue demais com o que é possível no Azure (alguns dos quais exploraremos em capítulos posteriores), vamos examinar alguns fundamentos. As próximas páginas podem parecer considerações e opções familiares com VMs na infraestrutura local. Se sim, ótimo! Se isso é novo, não se preocupe; muito do gerenciamento é abstraído no Azure, e coisas como redes virtuais são normalmente criadas e configuradas apenas uma vez. Veremos cada área com mais detalhes nos próximos capítulos, então respire fundo e dê um passo de cada vez.

2.1.1 Regiões e opções de disponibilidade

O Azure é dividido em regiões ao redor do mundo, e cada região tem um ou mais data centers. Esses data centers fornecem os principais recursos de computação, armazenamento e rede para executar seus workloads e aplicações. O Azure é executado em mais de 50 regiões, e a lista está crescendo. Com tantas regiões, a ideia é que você possa implantar aplicações próximas a seus funcionários ou clientes. Essa disponibilidade regional reduz a latência e melhora a experiência do usuário final.

Uma região do Azure pode não oferecer todos os serviços disponíveis no Azure. Com centenas de serviços disponíveis, o conjunto mais comum de serviços principais geralmente está em todos os lugares, mas os serviços novos ou de nicho normalmente são lançados ao mesmo tempo. Ao planejar suas aplicações no Azure, confira a disponibilidade do produto por região em https://azure.microsoft.com/global-infrastructure/services.

No capítulo 8, analisaremos algumas das opções de alta disponibilidade, como conjuntos e zonas de disponibilidade. Essas opções de redundância permitem que o Azure distribua várias instâncias de suas VMs ou aplicações em um único data center ou em uma região inteira. Essa capacidade permite que você defina sua tolerância para atualizações de manutenção ou falha de hardware. Nos primeiros capítulos deste livro, você normalmente criará apenas uma ou duas VMs, portanto, não se preocupe com essas opções de disponibilidade ainda.

2.1.2 *imagens de VMs*

Para criar uma VM, você precisa de um ponto de partida. Normalmente, esse ponto de partida se resume à escolha do sistema operacional: Windows ou Linux. Em seguida, vem a escolha de qual versão do Windows a ser usada (como o Windows Server 2016 ou 2019) ou qual distribuição do Linux será usada (como Ubuntu, Red Hat Enterprise Linux ou SUSE).

Uma *imagem*, um pacote de sistema operacional pré-configurado com opções de configuração básica aplicadas, é esse ponto de partida. O Azure contém centenas dessas imagens pré-criadas no Azure Marketplace para usar quando você cria VMs. Muitas vezes, você pode aplicar licenças existentes do Windows, dependendo do seu modelo de licenciamento atual, ou optar pelo suporte adicional da Canonical para executar o Ubuntu Linux ou atualizações da Red Hat, por exemplo.

Para manter as coisas simples o suficiente para que você possa concluir essas lições em uma hora de almoço, você usará essas imagens pré-criadas no Azure ao longo do livro. No mundo real, você provavelmente quer fazer personalizações para atender às suas necessidades e requisitos de negócios. Para fazer isso, muitas vezes você criará suas próprias imagens de VM. O fluxo de trabalho para criar e gerenciar as VMs é o mesmo que com as imagens do Azure Marketplace. Porém, muitas vezes a criação de suas próprias imagens requer muito planejamento e, em seguida, horas de configuração, generalização e captura de suas próprias imagens antes do tempo.

Experimente agora

Veja algumas ideias para pensar enquanto você planeja aplicações no Azure. Eles parecem básicos e, muitas vezes, você pode tomar essas decisões automaticamente, sem pensar muito. Mas ainda é importante entender suas necessidades de aplicação antes de começar a criá-los e executá-los.

- Em quais regiões a aplicação deve ser executada? Você tem uma grande concentração de usuários em uma região específica? Como você fornecerá redundância?

  Se você estiver criando aplicações internas, execute-as na região do Azure mais próxima dos usuários. Se você tem um grande escritório em Houston, Texas, por exemplo (pode ser que goste de foguetes!), execute suas aplicações e VMs do Azure na região centro-sul dos Estados Unidos.

  Se está criando aplicações externas, você prevê que terá clientes de certas regiões? Essa configuração pode exigir várias instâncias implantadas em regiões diferentes (e também fornecer alta disponibilidade). Vamos falar sobre essa configuração no capítulo 12.
- Você precisa fornecer muitas personalizações de VM? Quanto tempo leva para testar e validar todas essas alterações? O que as empresas precisam impulsionar?

  Em um ambiente tradicional na infraestrutura local, muitas vezes é gasto muito tempo criando imagens pré-configuradas para implantações. Na nuvem, tente minimizar esse tempo. As imagens pré-criadas do Azure incluem as atualizações de segurança mais recentes. Eles são testados para você e, em seguida, replicados geograficamente para os tempos de implantação mais rápidos.

Se você criar suas próprias imagens, use recursos como a Galeria de imagens compartilhadas do Azure para distribuir e replicar essas imagens conforme necessário (https://docs.microsoft.com/azure/virtual-machines/windows/shared-image-galleries).

2.1.3 Tamanhos de VM

Há várias famílias de tamanhos de VM no Azure. Essas famílias contêm grupos de tipos de hardware virtuais semelhantes que são direcionados para determinados workloads. Os tamanhos às vezes são atualizados à medida que novas ofertas de hardware e workload se tornam disponíveis, mas as famílias principais permanecem constantes. Os tipos de família são os seguintes:

- *Objetivo geral*— Ótimo para desenvolvimento e teste, ou bancos de dados de produção ou servidores Web de baixo uso.
- *Computação otimizada* — CPUs de alta performance, como para servidores de aplicativos de produção.
- *Memória otimizada* — Opções de memória maior, como quando você precisa executar grandes bancos de dados ou tarefas que requerem muito processamento de informações in-memory.
- *Armazenamento otimizado* — performance de baixa latência e alto disco para aplicações com uso intensivo de disco.
- *GPU* — VMs especializadas em gráficos baseados em NVIDIA, se você precisar de renderização de gráficos ou processamento de vídeo.
- *Computação de alta performance*—muito de tudo: abundância de CPU, memória e throughput de rede para os workloads mais exigentes.

Qual tamanho de VM pode ser criado no Azure? As coisas estão melhorando constantemente, mas no momento de gravar a maior VM, você pode criar uma série Mv2 (parte da família de memória otimizada) com 208 CPUs virtuais e 5,7 TiB de memória. Isso deve fazer um servidor de Minecraft decente, você não acha?!

A principal coisa a aprender aqui é que o número de VMs e a quantidade de CPU e memória que você pode solicitar no Azure são limitados apenas pelo seu orçamento. Você provavelmente lutaria para criar VMs desse tamanho no mundo da infraestrutura local tradicional.

Ao criar uma VM no portal do Azure ou usando a CLI ou o PowerShell, você deve escolher o tamanho da VM a ser usada. Um tamanho de VM comum, como D2s_v3, é frequentemente usado como padrão para começar. Isso provavelmente é muita potência para um servidor Web básico neste capítulo, mas é rápido para criar a VM e instalar os pacotes necessários.

O portal do Azure permite filtrar com base em um tamanho aproximado (como pequeno, médio ou grande) ou uma família específica (como as VMs de uso geral ou com otimização de memória). Um custo mensal estimado também é mostrado para dar uma ideia do preço de cada VM. Preste atenção nos custos, pois eles podem subir rapidamente! Dentro do motivo, você pode alterar o tamanho da VM depois que a VM está em operação, embora a VM deva ser encerrada e reiniciada para concluir o processo.

Redução de custos de VM

As VMs criadas por padrão geralmente são superalimentadas para o que você precisa, mas são rapidamente implantadas e usadas, o que ajuda a reduzir a quantidade de tempo que você gasta Instalando pacotes na sua pausa para o almoço.

No mundo real, preste atenção às demandas de memória, CPU e armazenamento das suas VMs. Crie VMs de tamanho adequado. Essa abordagem é a mesma que no mundo da infraestrutura local, onde você pode acabar com VMs que têm muito mais memória ou muitas CPUs virtuais mais atribuídas do que o necessário.

Há também um tipo especial de VM no Azure: a *Série B*. Esses tamanhos de VM usam recursos de CPU e memória inicializáveis, e você pode usar créditos bancários para recursos de computação não utilizados. Se pretender economizar seus créditos do Azure, escolha essa série de VMs para os exercícios do livro. Eles vêm com um preço mais baixo e são ótimos para cenários onde você nem sempre precisa de muitos recursos de CPU e memória. Entretanto, tome cuidado: dependendo do tamanho da VM da série B que você criar, ela pode ter menos CPU e memória do que algo como a série D2s_v3. Então, ela será executada um pouco mais lentamente.

2.1.4 Armazenamento do Azure

O armazenamento para VMs no Azure é simples. Quantos discos você quer, de que tamanho e de que tipo? Os dois primeiros realmente não são específicos do Azure. Então, vamos ignorá-los. Esses tipos de armazenamento estão disponíveis:

- *Discos SSD Premium (unidade de estado sólido)*: usam SSDs de baixa latência e alta performance e são perfeitos para workloads de produção. Você deve usar principalmente esse tipo para ter melhor performance para suas aplicações.
- *Discos SSD padrão*: usam SSDs padrão e oferecem performance consistente em comparação com unidades de disco rígido (HDDs). Esse tipo é excelente para workloads de desenvolvimento e testes ou uso de produção de acordo com o orçamento e de baixa demanda, como servidores Web.
- *Discos HDD padrão*: usam discos giratórios normais e são ideais para acesso não frequente aos dados, como arquivos ou backups. Esse tipo não recomendado para executar workloads de aplicação.

Você não precisa ir muito mais fundo nas especificidades do armazenamento para criar um servidor Web rápido. Você aprenderá mais no capítulo 4, incluindo os discos ultra que são apenas para discos de dados anexados. Por enquanto, é suficiente saber que, quando você escolhe um tamanho de VM, isso também ajuda a definir o tipo de armazenamento usado.

Os discos virtuais que você usa, independentemente do tipo, são chamados de *discos gerenciados do Azure*. Esses discos gerenciados permitem que você crie uma VM e anexe discos de dados adicionais sem se preocupar com contas de armazenamento subjacentes, limites de recursos ou cotas de performance. Os discos gerenciados também são criptografados automaticamente em repouso: você não precisa configurar nada para proteger seus dados! Novamente, o capítulo 4 abrange tudo isso e muito mais. Por enquanto, você geralmente pode permitir que o Azure crie o disco mais apropriado com base no tamanho da VM selecionado.

Experimente agora

Para verificar seu conhecimento, trabalhe com as seguintes perguntas:

- Para a maioria dos workloads de produção, que tipo de disco fornece a melhor performance?

  Um disco SSD Premium geralmente é o que você deve executar para workloads de produção. Esse tipo geralmente é a opção padrão quando você cria uma VM. Os discos SSD padrão são uma segunda escolha razoável, e os SSDs ultra devem ser usados somente em aplicações muito intensivas em disco que exigem baixa latência. Embora haja um pouco de economia HDDs padrão, a performance geralmente é ótima, assim como ocorre com ambientes virtuais na infraestrutura local.
- Qual família de VMs é uma boa opção para um servidor de banco de dados?

  Uma VM otimizada para memória é adequada, pois os bancos de dados muitas vezes precisam de um maior número de memória do que os recursos de CPU. Sempre tente estimar as necessidades de recursos e, em seguida, monitorar a performance após a implantação. Não tenha medo de alternar para um tamanho de VM diferente para fornecer a performance desejada.

2.1.5 Redes virtuais

Parece óbvio, mas uma VM precisa de conectividade de rede se você quiser que alguém acesse suas aplicações. Para um servidor Web básico, você precisa de uma rede virtual e conectividade externa. O capítulo 5 aborda detalhadamente o Core Networking do Azure, e o capítulo 9 aborda como distribuir o tráfego para várias VMs usando balanceadores de carga. As coisas realmente ficam interessantes no capítulo 11, com o DNS do Azure e o roteamento global dos usuários finais com o Gerenciador de tráfego. Não vou fazer de você um engenheiro de rede, mas você vai aprender muito sobre as redes do Azure neste livro!

Para começar a usar os fundamentos necessários para este capítulo, uma rede virtual no Azure é constituída pelos mesmos recursos essenciais de uma rede física regular:

- Um espaço de endereçamento e uma máscara de sub-rede, como 10.0.0.0/16
- Uma ou mais sub-redes, que você pode usar para dividir o tráfego externo, de banco de dados ou de aplicação, por exemplo
- Placas de interface de rede virtual (NICs) que conectam VMs a uma determinada sub-rede
- Endereços IP virtuais que são atribuídos a recursos como uma NIC virtual ou balanceador de carga

Você pode criar uma VM que só esteja conectada a uma rede virtual sem fornecer conectividade externa, o que pode ser o caso para banco de dados de back-end ou servidores de aplicações. Para se conectar a essas VMs para administração e manutenção, você pode criar uma conexão de rede virtual privada (VPN) ou usar uma conexão privada e dedicada ao Azure a partir de seu equipamento de rede local. No Azure, essa conexão dedicada é chamada de *ExpressRoute*.

O servidor Web básico que você vai criar neste capítulo requer um tipo específico de endereço IP virtual: um endereço IP público. Esse endereço IP público é atribuído à NIC virtual e permite que o tráfego externo atinja sua VM. Então, você pode controlar o fluxo de tráfego para sua VM com NSGs (grupos de segurança de rede). Pense em um firewall normal que você usa para abrir ou fechar várias portas e protocolos. No Azure, os grupos de segurança de rede bloqueiam o tráfego por padrão e permitem apenas o tráfego específico que você define. O tráfego comum a ser permitido é HTTP ou HTTPS em portas TCP 80 e 443. O gerenciamento remoto usando o Remote Desktop Protocol (RDP) ou o Secure Shell (SSH) também pode ser aberto, com cuidado, o que você fará mais adiante neste capítulo para ver como se conectar e instalar pacotes.

2.2 Criar um par de chaves SSH para autenticação

No exercício de laboratório de fim de capítulo, você criará o que provavelmente já conhece: uma VM do Windows Server. Esse tipo de VM usa a autenticação baseada em senha. Muitas aplicações na nuvem são executados no Linux; na verdade, mais da metade das VMs no Azure executam o Linux. Normalmente, você não usa a autenticação baseada em senha com o Linux; em vez disso, você usa SSH e um par de chave pública. Para começar a expandir suas habilidades, o servidor Web básico neste capítulo executa o Linux, então você precisa ter confiança no modo como cria e usa SSH. Você não *precisa* da experiência do Linux para trabalhar na nuvem, mas eu recomendo que aprenda algumas das noções básicas!

Pares de chaves SSH

Secure Shell (SSH) é um protocolo usado para se comunicar com segurança com computadores remotos e é a maneira mais comum de fazer login em VMs Linux. É semelhante ao uso de uma conexão RDP com uma VM do Windows, exceto que, no Linux, toda a sessão SSH normalmente se baseia no console. Com criptografia de chave pública, você pode usar um par de chaves digital para autenticá-lo com uma VM Linux remota.

Um par de chaves SSH tem duas partes: uma chave pública e uma chave privada. A chave pública é armazenada na sua VM do Linux no Azure. Você mantém uma cópia da chave privada. Quando você precisa fazer login na sua VM do Linux, a chave pública na VM remota é associada à chave privada mantida localmente. Se os pares de chaves corresponderem, você está conectado à VM. Há um pouco mais do que isso, mas, basicamente, a criptografia de chave pública é um grande meio para verificar a identidade.

Eu gostaria de ter o hábito de usar chaves SSH para fazer login em VMs Linux. As chaves SSH são muito mais seguras do que senhas porque, entre outras coisas, elas não são suscetíveis a ataques de senha de força bruta. Você deve sempre focar na segurança como um conceito central, especialmente na nuvem.

Experimente agora

Crie um par de chaves públicas SSH usando o Azure Cloud Shell:

1 Abra um navegador da Web para https://portal.azure.com. Faça logon na conta do Azure que você criou no capítulo 1 e selecione o ícone do Cloud Shell na parte superior do painel, como mostrado na Figura 2.2. Você também pode abrir o Cloud shell diretamente em https://shell.azure.com.

Figura 2.2 Selecione e inicie o Cloud Shell no portal do Azure selecionando o ícone do Shell.

2 Na primeira vez que você abrir o Cloud Shell, levará alguns instantes para criar um armazenamento persistente que sempre será conectado às suas sessões. Esse armazenamento permite que você salve e recupere scripts, arquivos de configuração e pares de chaves SSH. Aceite todos os prompts para permitir que esse armazenamento seja criado.

3 Se necessário, escolha Bash no menu suspenso, no canto superior esquerdo do Cloud Shell. O suporte do PowerShell também está disponível, mas vamos nos concentrar principalmente no Bash e na CLI do Azure ao longo do livro.

4 Para criar um par de chaves, digite o seguinte comando:

```
ssh-keygen
```

5 Aceite os prompts padrão pressionando a tecla Enter. Em alguns segundos, você tem um par de chaves públicas SSH que pode usar com todas as suas VMs. O comando `ssh-keygen` assume como padrão uma chave de 2.048 bits de comprimento e usa o protocolo RSA versão 2. Este é um bom equilíbrio de segurança e é o tipo recomendado para a maioria dos casos de uso. A Figura 2.3 mostra um exemplo de um par de chaves SSH concluído no Cloud Shell.

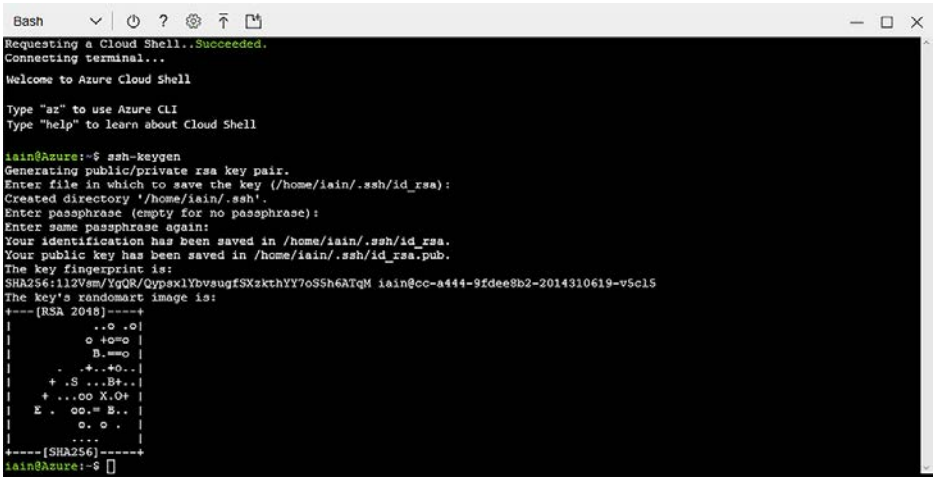


Figura 2.3 Um par de chaves SSH criado no Azure Cloud Shell com o comando `ssh-keygen`

6 Para exibir sua chave pública e usá-la com uma VM, digite o seguinte comando:

```
cat .ssh/id_rsa.pub
```

7 Selecione a saída e copie-a em um arquivo de texto simples no seu computador. Você usará essa chave pública para criar uma VM na seção 2.3. Essa VM é referenciada da CLI do Azure ao longo do restante do livro. Normalmente, você não precisa copiar e colar toda a chave a cada vez, mas é bom ver o que está acontecendo no início. Essas informações não são super secretas, então convém usar o Notepad ou o TextEdit para criar e salvar uma cópia da chave. Tenha cuidado ao copiar a saída da chave pública, pois ela é sensível a espaço em branco adicional ou falta um caractere. Um exemplo de uma chave pública SSH completa é o seguinte:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDPGaOBsfhJJOHAWAv+RLLR/vdUTzS9HOIj
➥JyzWWLsnu0ESH2M6R+YYPPNXv9X7dmVyM1zCXXEaLucpnyFjevbwPedxTgifyxgCFTgylr1
➥kg7o4EyCTGBGhTA+hSHuhXGXa12KPdKWehsPwHMa6Hs8fbt/in9Z1k2ZAwvbT+LWPcmJgNO
➥FuolIHOsOEeoQQqdXLrGa7NU/3fzSXdT9Y2BT1KLINc4KnwdOuONddLw3iANvK+Gkwax8iK
➥7IicKMoammwvJUCRf+MTEK9pZ84tfsc9qOIAdhrCCLbQhtoWjZpIwYnFk+SNBE8bZZtB8b2
➥vkDFNZlA5jcAd6pUR3tPuL0D iain@cc-a444-9fdee8b2-2014310619-v5cl5
```

DICA O Cloud Shell é baseado em navegador, portanto, os atalhos de teclado para copiar e colar podem funcionar um pouco diferente do que você está acostumado. Ctrl-Insert e Shift-Insert devem copiar e colar, em vez dos comandos Ctrl-C e Ctrl-V normais.

2.3 Criar uma VM a partir do seu navegador da Web

Agora que você conhece um pouco da teoria das VMs do Azure e criou um par de chaves SSH, já pode entrar e criar uma VM. Vou começar e, depois, vou deixar você configurar a VM com base no que acabou de aprender. Então, preste atenção!

As ferramentas do Azure CLI e do Azure PowerShell são incrivelmente poderosas, mas uma grande vantagem do Azure é quanto tempo levou para criar uma grande experiência de portal. O portal do Azure é uma ferramenta gráfica baseada na Web que permite que você veja como todos os diferentes componentes se unem e faça uma verificação visual rápida de que tudo está bem. O portal inclui algumas coisas únicas que as outras ferramentas não fornecem, e é rápido de usar, pois você não precisa instalar nada.

Experimente agora

A criação de uma VM no Azure oferece muitos padrões que você pode usar para reduzir o número de opções que você precisa criar. Para este exercício, veja os recursos que o Azure vai criar com base no que você aprendeu na seção 2.2 para itens como rede e armazenamento:

1 No portal do Azure, (https://portal.azure.com), selecione Create a Resource (Criar um Recurso) no canto superior esquerdo do painel. Os recursos populares devem ser listados, incluindo a versão mais recente de suporte a longo prazo (LTS) do Ubuntu (até o momento da redação deste livro, era o Ubuntu Server 18.04 LTS).

2 Selecione a versão LTS.

Você também pode pesquisar no Marketplace na parte superior da janela ou navegar pela lista de serviços de alto nível (como computação e rede) para ter uma ideia do que mais está disponível para suas necessidades futuras. Tente ficar com o Ubuntu Server 18.04 LTS para que você possa seguir um dos próximos exercícios para instalar os componentes do servidor Web.

3 Crie um grupo de recursos para seu servidor Web.

Quando você cria recursos no Azure, eles são logicamente contidos em um grupo de recursos definido. Esses grupos normalmente contêm recursos de mentalidade semelhante para suas aplicações. O capítulo 7 mostra como planejar e gerenciar aplicações usando grupos de recursos.

Por enquanto, sugiro nomear grupos de recursos por capítulo para ajudar a organizar as coisas. Dê um nome ao grupo de recursos neste exercício `azuremol-chapter2`, por exemplo.

4 Dê um nome à sua VM, como a webvm, e escolha uma região próxima a você. Não se preocupe com a redundância da infraestrutura por enquanto.

Veja as opções para a imagem da VM, apenas para ter uma ideia das outras opções, mas para este exercício, fique com o Ubuntu Server 18.04 LTS. O tamanho padrão da VM é adequado para este exercício, mas, novamente, procure ver o que está disponível e como você consulta os diversos tamanhos de aluguel e qual hardware eles executam. Veja como os tamanhos se alinham com as famílias de VM que você analisou anteriormente neste capítulo.

5 Verifique se você está usando a autenticação de chave pública SSH e, em seguida, forneça um nome de usuário, como azuremol. Você usará esse nome de usuário para fazer logon na VM no próximo exercício.

6 Copie e cole a chave pública SSH que você criou na seção anterior. Novamente, certifique-se de que não haja nenhum espaço em branco ou formatação adicional ao copiar e colar a chave pública. A chave SSH deve estar em uma linha. Até mesmo a quebra automática de linha no Bloco de Notas pode causar problemas! O portal do Azure valida a chave antes de você prosseguir.

7 Para se conectar à VM no próximo exercício e instalar os componentes do servidor Web,abra a SSH na porta 22.

Abrir a SSH em uma VM pública não é uma boa prática de segurança. O capítulo 16 analisa como abrir e restringir o acesso automaticamente usando o acesso just-in-time da VM.

Veja algumas das outras portas que você pode abrir aqui. HTTP e HTTPS são portas comuns para abrir, e você deve criar um servidor Web neste capítulo, certo? Não trapaceie e abra essas portas. Quero apresentá-lo à CLI do Azure no próximo exercício, onde você permitirá o tráfego HTTP.

Conecte-se com segurança usando um host bastion

Em cenários do mundo real, você não deve abrir portas de gerenciamento remoto para SSH ou RDP para a Internet pública. Sério, não faça isso! Siga as práticas recomendadas que você deve usar no mundo que não é na nuvem, na infraestrutura local, como se conectar somente quando necessário e limitar o acesso remoto a um conjunto específico de endereços de gerenciamento.

(continuação)

Uma maneira comum de fornecer acesso remoto é usar um host bastion ou uma caixa de salto. Nesse tipo de configuração, você não se conecta diretamente a servidores de aplicações do seu laptop ou PC. Em vez disso, você se conecta a um host bastion dedicado e ao servidor que precisa gerenciar. Essa abordagem bloqueia o acesso a um conjunto limitado de endereços e proporciona uma maneira segura de permitir o gerenciamento remoto.

O Azure Bastion (https://docs.microsoft.com/azure/bastion) fornece uma abordagem gerenciada para essa necessidade de conexão remota segura. Você cria um host do Azure Bastion em uma sub-rede dedicada e, em seguida, usa esse host para se conectar a VMs que executam suas aplicações. Essas VMs não precisam ser acessíveis ao público. Você pode fazer tudo por meio do portal do Azure sem abrir portas de rede para SSH ou RDP. O próprio host do bastion é gerenciado para você em termos de atualizações de segurança e regras de grupo de segurança.

8. Veja algumas das outras opções de configuração de VM para armazenamento e rede para se familiarizar com as opções, embora você possa deixar tudo como padrão por enquanto.
9. Há também algumas opções de gerenciamento interessantes, como habilitar o desligamento automático, backups e diagnósticos, que são abordados nos capítulos 12 e 13. Por enquanto, não se importe com diagnóstico de inicialização e de convidado do sistema operacional, pois você precisa criar e configurar uma conta de armazenamento para que eles funcionem.
10. Quando você estiver pronto, revise e crie sua VM básica.

2.4 Conectar-se à VM e instalar o servidor Web

Quando sua VM estiver em funcionamento, você poderá usar a chave SSH que criou anteriormente para fazer logon na VM. Você pode então começar a instalar e configurar o servidor Web, e você pode fazer isso com o Cloud Shell.

2.4.1 Conectando-se à VM com SSH

Esta seção aborda como você pode obter rapidamente os detalhes da conexão para sua VM.

Experimente agora

Se Linux é algo novo para você, não se preocupe! Siga as próximas etapas para fazer login em sua VM:

1. No portal do Azure, navegue até e selecione Máquinas Virtuais na barra de navegação no lado esquerdo da tela. Leva alguns minutos para criar a VM do exercício anterior, então selecione o botão Atualizar até que o status da VM seja *Em execução*. Quando estiver pronto, escolha sua VM e selecione Conectar, como mostrado na Figura 2.4.

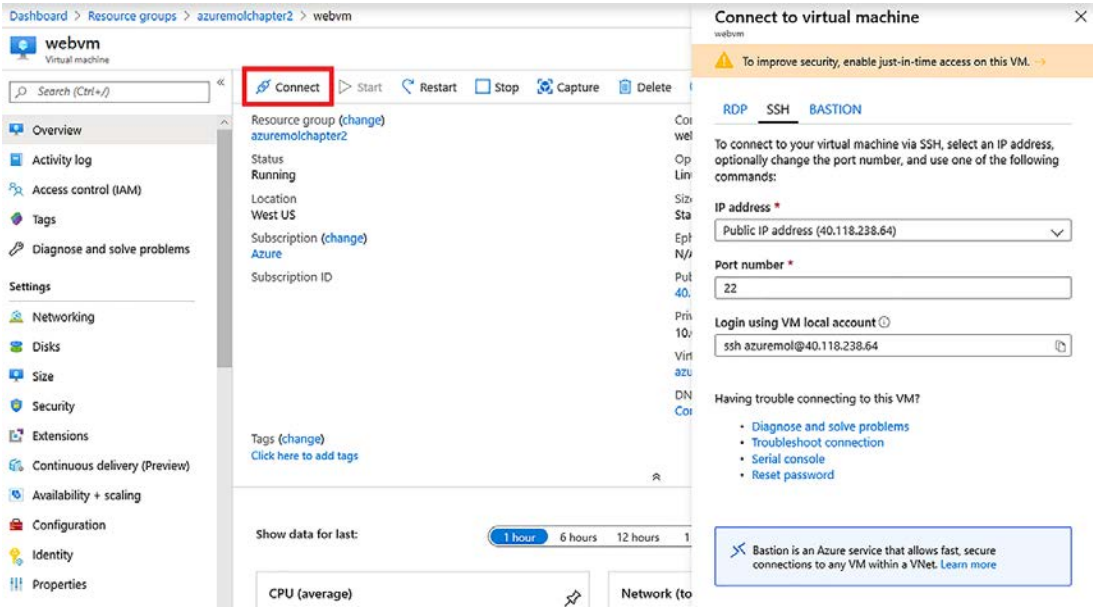


Figura 2.4 Selecione sua VM no portal do Azure e selecione Conectar para gerar as informações de conexão SSH.

Com uma VM do Linux, você vê o comando SSH que inclui seu nome de usuário e endereço IP público. Copie este comando de conexão SSH, como `ssh azuremol@104.209.208.158`.

Em uma VM do Windows, o botão Conectar baixa um arquivo de conexão RDP para o computador que está preenchido com o endereço IP público da VM.

2. Se necessário, abra o Cloud Shell novamente. Se você estiver alternando entre o Cloud Shell e o portal, poderá minimizar o Cloud Shell para mantê-lo disponível em segundo plano.
3. Cole o comando SSH no Cloud Shell e, em seguida, pressione Enter. A chave SSH que você criou anteriormente é usada automaticamente para autenticar.

   Na primeira vez que você se conecta a uma VM com SSH, ela solicita a adição de uma lista de hosts confiáveis. Essa é outra camada de segurança que a SSH fornece. Se alguém tentar interceptar o tráfego e direcioná-lo para uma VM remota diferente, o cliente SSH local saberá que algo mudou e avisará antes de se conectar.

   Aceite o prompt para armazenar a conexão à VM remota. A Figura 2.5 mostra o processo de conexão SSH no Azure Cloud Shell.

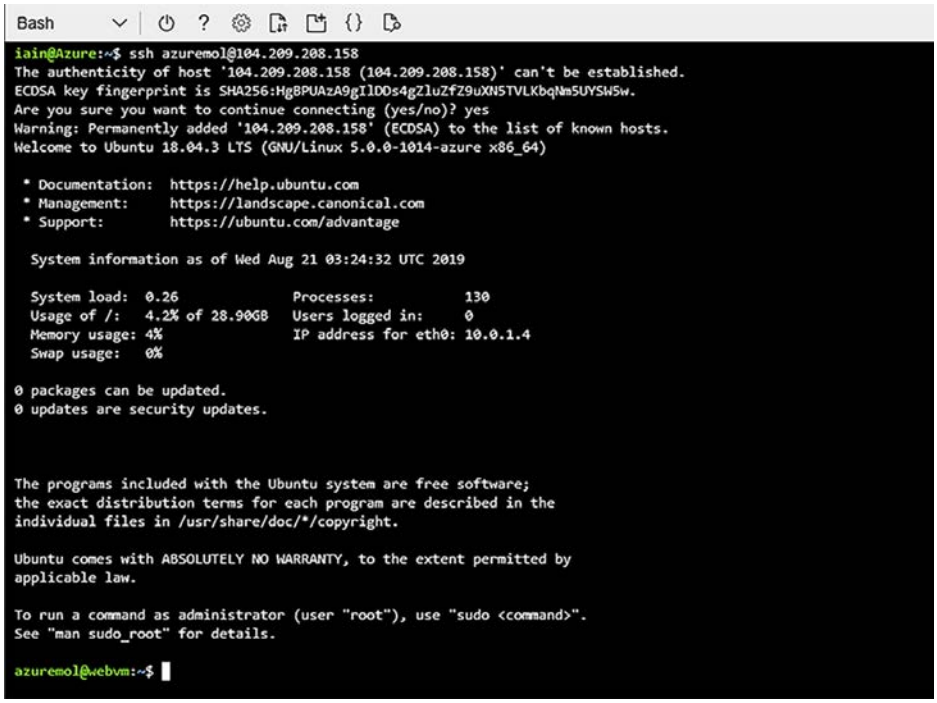


Figura 2.5 Use a string de conexão mostrada no portal do Azure para criar uma conexão SSH com a VM a partir do Cloud Shell.

Neste ponto, você está em casa, longe de casa ou o prompt do Linux é totalmente estranho. Não se preocupe. Você não precisa saber um grande número de comandos do Linux, e cada comando é explicado ao longo do livro. Dito isto, eu recomendo que você aprenda pelo menos algumas habilidades básicas de administração do Linux. Grande parte da nuvem é baseada em sistemas Linux, e há um grande movimento em direção a contêineres e microsserviços para desenvolvimento e gerenciamento de aplicações. Se você é um administrador do Windows das antigas, seja bem-vindo! Há algo especial para você no final do capítulo, então tenha paciência.

2.4.2 Instalar o servidor Web

Criar uma VM? Pronto. Conectar-se à VM com SSH? Pronto. Agora você pode instalar os pacotes para um servidor Web e se preparar para vê-lo em ação.

O Azure suporta muitas distribuições Linux diferentes (*distros*). As ferramentas de gerenciamento de pacotes e os locais de arquivos de configuração variam um pouco entre os distros. Vamos usar o Ubuntu neste livro porque é uma das distros Linux mais populares e bem documentadas para a computação na nuvem. Se você tiver dificuldades ao longo do caminho, há muita documentação para ajudá-lo, começando em https://help.ubuntu.com. Se você quiser usar uma distribuição diferente com a qual já está familiarizado, fique à vontade. Caso contrário, use o Ubuntu.

Experimente agora

Na sessão SSH para a VM, instale os pacotes do servidor Web com a APT:

1. No Ubuntu, você instala pacotes com uma ferramenta de embalagem avançada (APT) — uma ferramenta de gerenciamento de pacote superpoderosa que instala automaticamente os pacotes adicionais de que precisa. Tudo que você precisa fazer é dizer "Instalar um servidor Web", e a APT instala todos os componentes necessários.

   Neste exemplo, instale a pilha da Web LAMP. Este é provavelmente o conjunto mais comum de componentes da Web: Linux, Apache (um servidor Web), MySQL (um servidor de banco de dados) e PHP (uma linguagem de programação da Web):

   ```
   sudo apt update && sudo apt install -y lamp-server^
   ```

   O primeiro comando atualiza os pacotes disponíveis, o que é uma boa prática para garantir a instalação dos pacotes mais recentes e maiores. Quando esse comando terminar, execute o próximo comando com `&&`. Por que não começar uma nova linha para o próximo comando? O `&&` executará o próximo comando somente se o comando anterior tiver sido bem-sucedido. Se não houvesse conectividade de rede para apt obter os pacotes mais recentes, por exemplo, (pode rir, eu sei que você deve ter conectividade de rede para se conectar!), não haveria necessidade de executar o comando install.

   Se o comando update for bem-sucedido, apt determinará quais pacotes adicionais precisa e começará a instalar `lamp-server`. Por que o símbolo de acento circunflexo no final (`^`)? Ele informa a `apt` para instalar todo o conjunto de pacotes que compõem o servidor AMP, não apenas um único pacote chamado `lamp-server`.

2. O instalador pode pedir uma senha ou usar por padrão uma senha MySQL vazia. Isso não é muito seguro e, para uso de produção real, você precisa especificar uma senha forte. No capítulo 15, ficamos muito legais e armazenamos uma senha forte e segura no Azure Key Vault que é automaticamente injetada nesse assistente de instalação do MySQL.

   Leva um minuto ou mais para instalar todos os pacotes para a pilha da Web LAMP, e pronto.

3. Digite exit para sair da VM e retornar ao prompt do Cloud Shell.

Pronto! Seu servidor Web está funcionando, mas você não poderá acessá-lo em um navegador da Web ainda. Para fazer isso, é necessário permitir que o tráfego da Web atinja a VM.

2.5 Permitir que o tráfego da Web atinja a VM

Seu servidor Web está em execução, mas se você inserir o endereço IP público da VM em um navegador da Web, a página da Web não será carregada. Por quê? Lembre-se dos grupos de segurança de rede discutidos brevemente na seção 2.1.5? Quando você criou a VM, um grupo de segurança de rede foi criado para você. Uma regra foi adicionada para permitir o gerenciamento remoto — neste caso, SSH.

O restante da VM é bloqueado. Para permitir que os visitantes acessem seu servidor Web pela Internet, você precisa criar uma regra no grupo de segurança de rede que permita o tráfego da Web. Caso contrário, ninguém poderá pedir pizzas!

2.5.1 Criar uma regra para permitir o tráfego da Web

Esta seção mistura as coisas um pouco ao usar a CLI do Azure para criar uma regra para o tráfego da Web. Você poderia ter aberto essa porta HTTP no portal quando criou a VM, mas perderia a metade da diversão!

O Azure CLI está disponível no Cloud Shell. Você não precisa instalar nada. O capítulo 5 abrange redes virtuais e grupos de segurança de rede em mais detalhes. Por enquanto, podemos verificar como o Azure CLI é rápido e poderoso com apenas um comando.

Experimente agora

Abra o Azure Cloud Shell e siga estas etapas para ver o Azure CLI em ação:

1. Se você fechou a janela do Cloud Shell, abra-a novamente no portal do Azure. Certifique-se de que o shell Bash seja carregado, não o PowerShell. Se necessário, mude para a versão Bash.
2. Para ver a CLI do Azure e os módulos instalados, digite `az --version`. Uma lista de módulos e números de versão é mostrada. O que é ótimo sobre o Cloud Shell é que ele sempre tem a versão mais recente e melhor disponível.

OBSERVAÇÃO Se estiver atento, você pode ter notado que o comando gerou informações sobre a versão do Python. Por que essa informação é importante? Python é uma linguagem de programação eficiente e popular. O Azure CLI é gravado em Python, que faz parte do que a torna versátil e disponível para instalar localmente em qualquer computador se não quiser usar sempre o Cloud Shell. Para acompanhar a Microsoft contribuindo para a comunidade de código aberto, o Azure CLI é disponibilizado no GitHub para que qualquer pessoa faça contribuições, sugestões ou informe problemas (https://github.com/Azure/azure-cli).

3. Para abrir uma porta, especifique o nome da VM e seu grupo de recursos, além do número da porta. Para o tráfego da Web, você precisa abrir a porta 80. Insira o grupo de recursos (`-g`) e o nome da VM (`-n`) que você especificou ao criar sua VM:

```
az vm open-port -g azuremolchapter2 -n webvm --port 80
```

2.5.2 Ver o servidor Web em ação

Agora que você tem uma porta aberta para sua VM, vamos ver o que acontece quando você tenta acessá-lo em um navegador da Web:

1. No portal do Azure, selecione sua VM, caso tenha saído dela. O endereço IP público é listado no canto superior direito da página de visão geral da VM.

2 Selecione o endereço e copie-o.

3 No navegador da Web, abra uma nova guia ou janela e cole o endereço IP público. O site padrão do Apache é carregado, como mostrado na Figura 2.6. Ok, ele não se parece com uma pizzaria, mas você tem a base pronta para trazer seu código e começar a criar sua aplicação.

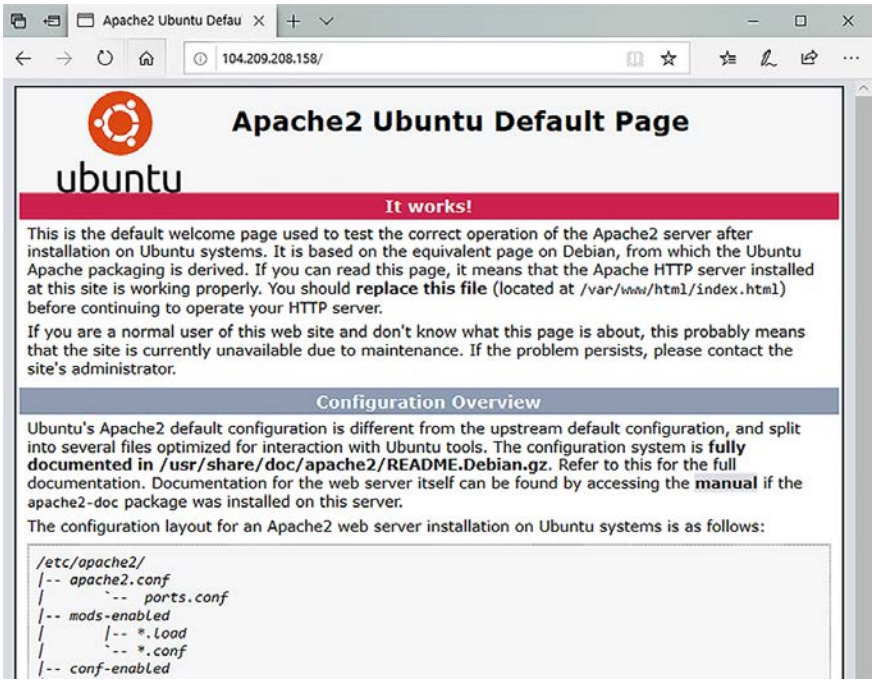


Figura 2.6 Para ver o servidor Web em ação e visualizar a página padrão do Apache 2, digite o endereço IP público em um navegador da Web.

2.6 Laboratório: criar uma VM do Windows

As seções anteriores mostraram a instalação da pilha LAMP em uma VM do Ubuntu Linux. Essa plataforma é comum para sites, mas você talvez queira um pouco de amor e atenção se usar Windows. Suas equipes de desenvolvimento ou tomadores de decisão de negócios podem querer usar .NET, por exemplo. Mesmo assim, você pode executar .NET Core em VMs Linux, portanto, não deixe que a linguagem conduza sua decisão.

Pelo que você aprendeu no exemplo passo a passo, tente criar uma VM que execute Serviços de Informações da Internet (IIS). Veja algumas dicas:

- Você precisa de uma VM que execute o Windows Server 2019.
- Como você usa RDP, não SSH, a experiência de conexão será um pouco diferente.
- No Gerenciador de Servidores, procure uma opção para Adicionar Funções e Recursos.

- Você precisa instalar o servidor Web (IIS).
- Não se esqueça de abrir uma porta de rede para o tráfego HTTP na porta TCP 80. Você pode usar o portal, se quiser.

2.7 Limpar os recursos

À medida que você cria recursos no Azure, o medidor de faturamento começa a girar. Você é cobrado por minuto, por isso é sábio criar bons hábitos e não deixar recursos como VMs em execução quando terminar de usá-los. Há duas maneiras de parar os encargos de cobrança para executar uma VM:

- *Desaloque uma VM.* Você pode selecionar o botão Parar no portal para interromper e desalocar uma VM, liberando todos os recursos de computação e de rede mantidos.
- *Excluir uma VM.* Essa opção é bastante óbvia. Se não restar nada no Azure, não haverá nada para pagar. Certifique-se de que você tenha encerrado a VM antes de excluí-la. Não há nenhum botão Desfazer no Azure.

Eu recomendo que você crie um grupo de recursos para cada implantação de aplicação ao começar a criar coisas no Azure. Ao percorrer os exercícios neste livro, você fará isso. Se você nomear seus grupos de recursos por capítulo, como `azuremolchapter2`, será mais fácil manter o controle de seus recursos e do que excluir. Isso torna a limpeza um pouco mais fácil, pois você pode excluir todo o grupo de recursos no final de cada capítulo. Escolha Grupos de Recursos no menu de navegação no lado esquerdo da tela. Abra cada grupo de recursos que você criou neste capítulo e selecione Excluir Grupo de Recursos, como mostrado na Figura 2.7. Para confirmar, você deverá inserir o nome do grupo de recursos.

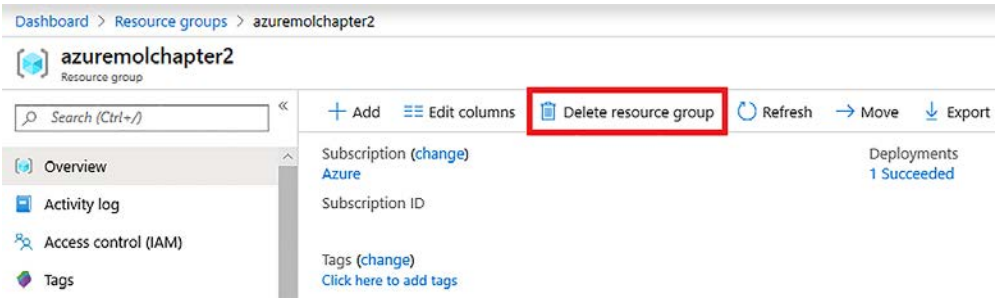


Figura 2.7 Para economizar custos, exclua grupos de recursos quando você não precisar mais deles.

Se você tiver o hábito de excluir recursos assim que terminar de usá-los, poderá fazer isso confortavelmente neste livro com os créditos gratuitos do Azure. No mínimo, desaloque sua VM no final de cada lição para que você possa retomar no dia seguinte e parar o relógio na cobrança.

2.8 Houston, temos um problema

Às vezes, você terá problemas no Azure. Pronto, já disse. Normalmente, a plataforma do Azure é boa com problemas que surgem à medida que você cria recursos:

- O Azure CLI ou o Azure PowerShell dá feedback à medida que você executa comandos, portanto, deve ser óbvio quando algo der errado. O Azure PowerShell normalmente usa texto vermelho agradável e calmo para chamar sua atenção.
- O Azure CLI pode ser um pouco mais enigmático porque geralmente inclui as respostas reais para as chamadas de API REST subjacentes do servidor. Se isso é tudo novo, pode levar alguns sucessos e falhas para entender o que está acontecendo de errado. A parte útil de obter as respostas REST é que você pode copiar e colar as mensagens de erro em seu mecanismo de pesquisa favorito e geralmente obter resultados sólidos para ajudá-lo a solucionar problemas.

Executar REST? Acabamos de começar!

Quando você abre uma página da Web no navegador, seu computador está se comunicando com um servidor Web usando HTTP. Eu posso quase garantir que você já viu uma mensagem de erro 404 em um site antes. Essa mensagem significa que a página da Web não foi encontrada. Outros erros comuns que você pode ter visto são 403, se não tiver permissões para exibir a página, e 500, se o servidor encontrar um erro.

Mesmo quando as coisas vão bem, em segundo plano, o navegador recebe mensagens de código 200 quando a página é carregada sem problemas ou mensagens de código 301 se uma página foi redirecionada para um novo local. Você não precisa entender e manter o controle de todos esses códigos; é apenas uma maneira padrão que HTTP usa para facilitar a comunicação entre computadores.

Anteriormente, este capítulo falou sobre como criar e gerenciar recursos do Azure por meio do portal da Web, CLI ou PowerShell. Todos os serviços do Azure são acessados por interfaces de programação de aplicação (APIs) Representational State Transfer (REST).

Se isso for novidade para você, as APIs REST são uma forma (um pouco) padronizada de expor serviços via TTP. Você usa solicitações HTTP padrão como `GET` e `POST` para solicitar informações ou fazer uma alteração e, assim que a plataforma aceita e processa a solicitação, você recebe uma mensagem de status. O Azure tem um conjunto bem definido de APIs REST.

Você não precisa entender o que isso significa. Basta estar ciente de que, quando você vê uma mensagem de erro, não é sempre no formato mais legível e útil. Às vezes, você começa a resposta HTTP bruta na API REST que deve decifrar por si mesmo. Novamente, cole esse erro em seu mecanismo de pesquisa favorito. É bem provável que alguém tenha encontrado o problema e fornecido uma razão mais legível por humanos para o que deu errado e o que você precisa corrigir.

Os problemas mais comuns com VMs ocorrem quando você se conecta à sua VM. Você pode estar se conectando para administração remota com SSH ou RDP, ou tentando acessar suas aplicações por meio de um navegador da Web ou cliente de desktop. Esses problemas estão frequentemente relacionados com a rede. Eu não culpo totalmente as pessoas da rede até o capítulo 5, então veja algumas coisas para verificar:

- Você pode se conectar a outras VMs ou aplicações do Azure em execução no Azure? Se não pode, algo local para a sua rede está provavelmente impedindo o acesso.

  Se você puder se conectar a outros recursos do Azure, verifique se abriu as regras de grupo de segurança de rede que falei na (seção 2.5). O capítulo 5 detalha essas regras.
- Para problemas de autenticação, tente o seguinte:
  - Confirme se você tem as chaves SSH corretas. O Azure deve informar quando você cria a VM se a chave pública é inválida. Se você tiver mais de uma chave privada, certifique-se de usar a correta.
  - Para problemas de RDP, tente se conectar a localhost\<*username*> e digite sua senha. Por padrão, a maioria dos clientes de RDP tenta apresentar credenciais locais ou credenciais de rede, que sua VM não entenderá.

3 Aplicativos Web do Azure

No capítulo 2, você criou uma VM e pacotes instalados manualmente para executar um servidor Web básico. Você poderia criar uma pizzaria online com essa VM se estivesse ansioso para começar. Um dos maiores casos de uso das VMs do Azure é executar Aplicativos Web, normalmente em escala. Os Aplicativos Web são um workload confortável para VMs. O conforto é bom, se você também gosta da manutenção atrelada ao gerenciamento de todas aquelas VMs, sabe, coisas divertidas, como atualizações de software, patches de segurança, log centralizado e relatórios de conformidade. E se você pudesse obter todo o poder de um servidor Web seguro para executar seus Aplicativos Web, incluindo a capacidade de dimensionar automaticamente para atender às demandas, mas sem a necessidade de criar e gerenciar todas essas VMs? Deixe-me apresentá-lo ao serviço Aplicativos Web do Azure.

Neste capítulo, comparamos a abordagem de infraestrutura como serviço (IaaS) de VMs e servidores Web à abordagem de plataforma como serviço (PaaS). Você aprende os benefícios dos Aplicativos Web do Azure ao criar um aplicativo Web e vê como trabalhar com suas versões de desenvolvimento e produção. Em seguida, você aprenderá a implantar seu aplicativo Web automaticamente a partir de um controle de origem, como o GitHub. Esse fluxo de trabalho é exibido na Figura 3.1. Os Aplicativos Web do Azure permitem implantar e executar

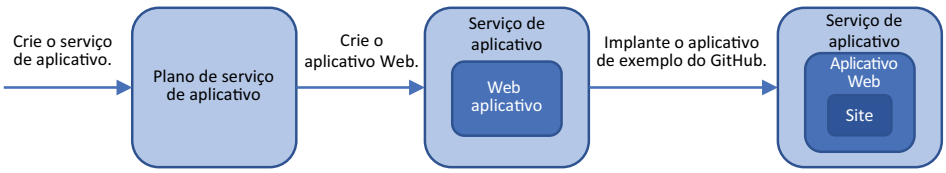


Figura 3.1 Neste capítulo, você cria um plano de serviço de aplicativo e um aplicativo Web básico e em seguida, implanta um site do GitHub.

pizzaria em questão de minutos, sem a necessidade de instalar e configurar uma VM e pacotes de servidores Web.

3.1 Visão geral e conceitos dos Aplicativos Web do Azure

Com os Aplicativos Web do Azure, você começa a mergulhar no maravilhoso mundo das soluções PaaS. Se você acha que a computação na nuvem envolve VMs, provavelmente deve redefinir essa ideia um pouco. No início deste livro, eu falei sobre a compra de energia do computador e focar em seus clientes e aplicações. À medida que você se move de soluções IaaS, como VMs, e deriva para soluções PaaS, como Aplicativos Web, seus clientes e aplicações se tornam o foco.

Executar Aplicativos Web em VMs IaaS requer o gerenciamento do sistema operacional, das atualizações de aplicações, das regras de segurança e tráfego e da configuração de todo o sistema. Com os Aplicativos Web, você carrega seu aplicativo Web e todas essas tarefas de gerenciamento são realizadas para você. Agora você pode focar em melhorar a experiência de aplicações para seus clientes, ou melhorar a disponibilidade com opções de dimensionamento e gerenciamento de tráfego.

Isso significa que você nunca deve executar VMs para hospedar um aplicativo Web? Provavelmente não. Há razões válidas para executar a pilha de aplicações inteira e configurá-la, como se você precisasse de suporte de aplicação muito específico ou tempo de execução de linguagem. Mas os Aplicativos Web podem fornecer muitos dos casos de uso para executar Aplicativos Web.

3.1.1 Linguagens e ambientes compatíveis

Que tipo de flexibilidade os Aplicativos Web oferecem em termos de linguagens de programação que você pode usar para criar seu aplicativo Web? Muita! Há duas plataformas primárias para a execução de Aplicativos Web: Windows e Linux. Você pode executar aplicativos Web .NET Core, Node.js, Python, Java, Ruby e PHP nativamente em instâncias de aplicativos Web do Windows e do Linux. No Windows, você também pode executar a estrutura .NET completa. Se você quiser ser muito legal e executar seu aplicativo Web em contêineres, também há Aplicativos Web para contêineres que permitem executar contêineres nativos do Docker para Linux. Falaremos mais sobre contêineres e Docker no capítulo 19. Por enquanto, entenda quais opções são cobertas com Aplicativos Web.

Quando os Aplicativos Web podem não fazer sentido? Nem todas as linguagens de aplicações têm o suporte de aplicativos Web. Digamos que você realmente quer se torturar com um aplicativo Web escrito em Perl. Nesse cenário, você provavelmente voltará a executar VMs IaaS gerenciadas por você, pois não há suporte para Perl em Aplicativos Web. Mas os Aplicativos Web, sem dúvida, suportam as linguagens de programação da Web mais comuns que você deseja usar. Você também provavelmente deve usar uma versão mais recente do seu aplicativo do que uma gravada em Perl.

Os Aplicativos Web oferecem suporte a várias linguagens e a várias versões dessas linguagens. Pegue o PHP como exemplo. Normalmente, você pode selecionar três ou quatro versões de PHP para melhor oferecer suporte à sua aplicação. E, o melhor de tudo, você não precisa se preocupar com as dependências do servidor Web subjacente para dar suporte a tudo, como faria se você mesmo tivesse gerenciado uma VM IaaS. Python é outro exemplo de diferenças entre as versões estáveis 2.7 e 3.6 (e posteriores), como mostrado na Figura 3.2.

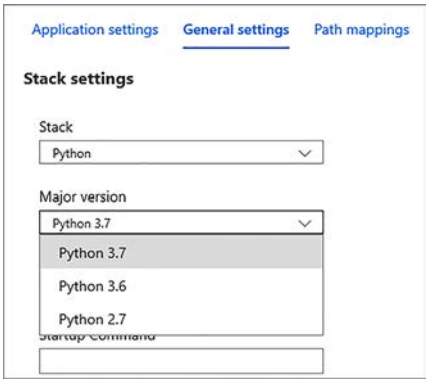


Figura 3.2 Selecione uma versão específica de uma linguagem nas configurações de aplicação dos Aplicativos Web.

Os Aplicativos Web também permanecem atualizados em correções de segurança. Mas não espere que uma versão mais antiga do PHP ou Python continue sendo suportada indefinidamente. Haverá um corte em versões mais antigas suportadas em um determinado ponto. Novamente, isso pode acontecer quando você voltar a executar VMs IaaS se seu aplicativo precisar de uma versão de linguagem mais antiga. Mas se você precisar executar uma versão mais antiga de uma determinada linguagem para oferecer suporte a uma aplicação herdada, não use uma abordagem de manutenção constante. Sempre procure mover esses aplicativos herdados para plataformas compatíveis mais modernas.

3.1.2 Preparação de diferentes versões com slots de implantação

Os slots de implantação fornecem um ambiente preparado para seu aplicativo Web. Você pode enviar novas versões do seu aplicativo para um slot de implantação e executá-las usando variáveis ambientais ou conexões de banco de dados, sem afetar o site ao vivo. Quando você estiver satisfeito com a aparência das coisas em um slot de implantação, poderá alternar essa versão para o site ao vivo em um instante. Então, o site anteriormente ao vivo alterna para um slot de implantação próprio, fornecendo uma versão arquivada ou, se necessário, você pode inverter o aplicativo de volta para a produção.

O número de slots de implantação disponíveis varia de acordo com a camada de aplicativo Web selecionada. Um número maior de slots de implantação permite que diversos desenvolvedores usem várias versões à medida que eles preparam e testam suas próprias atualizações.

3.1.3 Planos de Serviço de Aplicativo

Os Aplicativos Web fazem parte da família de serviços de aplicativo mais ampla no Azure. O Serviço de Aplicativo do Azure também inclui aplicativos móveis, aplicativos de API e aplicativos lógicos. Todos, exceto os aplicativos lógicos estão disponíveis em todas as regiões em que o Azure é executado. Veja um ótimo recurso para verificar a disponibilidade do serviço do Azure por região: https://azure.microsoft.com/regions/services. Muitos serviços estão disponíveis globalmente.

Quando precisa criar um recurso de serviço de aplicativo, como um aplicativo Web, você cria ou usa um plano de serviço existente. O plano de serviço define a quantidade de recursos disponíveis para você, quanta automação está disponível para escalar e fazer backup de seu aplicativo Web e como está

altamente disponível para criar seu site com slots de preparo e o Gerenciador de tráfego (uma maneira de encaminhar o tráfego geograficamente para a instância mais próxima de um usuário, o que veremos no capítulo 11). Como em qualquer coisa, você recebe pelo que paga. Suas necessidades de aplicações e negócios devem orientá-lo em relação à quantidade de recursos necessários e de quais recursos adicionais você precisa. Cada camada de serviço baseia-se nos recursos das camadas inferiores, geralmente adicionando mais armazenamento e recursos disponíveis.

Os quatro níveis principais do plano de serviço são os seguintes:

- *Free/Shared (Gratuito/Compartilhado)* — Usa uma infraestrutura compartilhada, oferece armazenamento mínimo e não tem opções para implantar diferentes versões em estágios, roteamento de tráfego ou backups. A camada compartilhada permite que você use um domínio personalizado e cobra por ele.
- *Basic (Básico)* — Fornece recursos de computação dedicados para seu aplicativo Web e permite que você use SSL e escale manualmente o número de instâncias de aplicativos Web que executa. As camadas gratuitas/compartilhadas e básicas proporcionam um bom ambiente para você testar o serviço de aplicativos Web, mas eu não recomendaria a realização de workloads de produção ou desenvolvimento reais. A performance não é um fator limitante, mas você perde alguns dos recursos automatizados, como backups e dimensionamento.
- *Standard (Padrão)* — Adiciona backups diários, escala automática de instâncias de Aplicativos Web e slots de implantação, e permite rotear usuários com o Gerenciador de Tráfego. Essa camada pode ser adequada para aplicações de baixa demanda ou ambientes de desenvolvimento em que você não precisa de um grande número de backups ou slots de implantação.
- *Premium* — Fornece backups mais frequentes, maior armazenamento e um maior número de slots de implantação e opções de dimensionamento de instâncias. Essa camada é ideal para a maioria dos workloads de produção.

O caso do isolamento

Com soluções de PaaS como Aplicativos Web, a infraestrutura é intencionalmente abstraída. Como os nomes de alguns níveis do plano de serviço sugerem, os Aplicativos Web são executados em uma plataforma compartilhada de recursos disponíveis. Isso não significa dizer que os Aplicativos Web são inseguros e que outros podem ver seus dados privados. Mas as razões de conformidade ou regulatórias podem exigir que você execute suas aplicações em um ambiente controlado e isolado. Acesse *ambientes de Serviço de Aplicativo: ambientes isolados que permitem executar instâncias do Serviço de Aplicativo como Aplicativos Web em uma parte segmentada de um data center do Azure. Você controla o tráfego de rede de entrada e saída e pode implementar firewalls e criar conexões de rede virtual privada (VPN) de volta nos recursos na infraestrutura local.*

Todos esses componentes de infraestrutura ainda são amplamente abstraídos com ambientes de Serviço de Aplicativo, mas essa abordagem fornece um grande equilíbrio quando você deseja a flexibilidade das soluções de PaaS, mas também quer reter alguns dos controles mais refinados sobre o fluxo de tráfego de conexões de rede.

Você pode fazer muito com as camadas gratuita e básica, embora para workloads de produção você provavelmente deve usar a camada Standard ou Premium. O exemplo deste capítulo usa a camada Standard para que você possa ver todos os recursos disponíveis. Ao usar Aplicativos Web com suas próprias aplicações, você pode decidir quantos desses recursos são necessários e selecionar a camada de plano de serviço mais apropriada de acordo.

3.2 Criar um aplicativo Web

Com um pouco de teoria por trás, vamos ver um aplicativo Web em ação. É preciso seguir algumas etapas para executar uma aplicação. Primeiro, você cria o aplicativo Web básico e vê o site padrão no seu navegador. Em seguida, use uma página da Web de exemplo do GitHub e envie-a para o Azure. Talvez seus desenvolvedores da Web tenham começado a criar uma vitrine para sua pizzaria online, então você tem um site básico pronto para upload.

> **OBSERVAÇÃO** Se você nunca usou Git antes, não se preocupe. Você não precisa entender o que o Git está fazendo nesse momento, e há espaço no final do capítulo para brincar e explorar um pouco. *Aprenda o Git em um mês de aulas,* por Rick Umali (https://www.manning.com/books/learn-git-in-a-month-of-lunches), é uma excelente introdução ao uso do Git se quiser aprender um pouco mais. Ele está disponível para leitura gratuita na plataforma Manning LiveBook.

3.2.1 Criar um aplicativo Web básico

Assim como eu fiz no capítulo 2, vou dar algumas orientações ao longo do caminho, mas veja se você consegue aplicar algumas das teorias sobre tempos de execução de aplicações e planos de serviço de aplicação para criar um aplicativo Web. Se você não tiver certeza sobre algumas opções, é seguro aceitar os padrões por enquanto.

PaaS, não IaaS

Este Aplicativo Web é um novo recurso. Ele é separado das VMs, como a que você criou no capítulo 2, que é uma abordagem IaaS da criação e da execução de Aplicativos Web. A abordagem PaaS é Aplicativos Web. Não há nenhuma relação real entre os dois tipos. Na verdade, se você seguiu o conselho no capítulo 2 e excluiu sua VM, esse aplicativo Web é executado sem uma VM em sua assinatura do Azure.

Experimente agora

Para criar um Aplicativo Web, conclua as etapas a seguir:

1. Abra um navegador da Web para https://portal.azure.com e faça logon na sua conta do Azure.
2. No portal, selecione Create a Resource (Criar um Recurso) no canto superior esquerdo do painel.
3. Escolha Web na lista de recursos que você pode criar e selecione Web App (Aplicativo Web).
4. Para ajudar a manter as coisas limpas e organizadas como você fez no capítulo 2, sugiro que você crie um grupo de recursos dedicado para seu Aplicativo Web, como azuremolchapter3.

5 Para o nome do Aplicativo Web, insira um nome globalmente exclusivo. Esse nome deve ser exclusivo, pois ele cria o URL para seu aplicativo Web na forma de http://<*name*>.azurewebsite.net. Se estiver se perguntando, sim — você pode aplicar um nome de domínio personalizado aqui. Por enquanto, use o endereço azurewebsites.net padrão.

6 Você vai usar códigos HTML básicos, não um contêiner do Docker, mas veja todas as diferentes pilhas de tempo de execução que estão disponíveis. Você pode alterar essa configuração depois de criar o Aplicativo Web, mas, por enquanto, escolha um tempo de execução ASP.NET que seja executado no Windows.

7 Permita que o Azure crie um plano de serviço de aplicativo automaticamente, mas altere o tamanho para o padrão S1. Essa camada fornece todos os recursos principais sem fornecer muitos recursos para seu site básico demonstrativo. Em implantações reais, é nessa etapa que você pode criar e configurar manualmente seus próprios planos de serviço de aplicativo ou selecionar um plano existente.

8 Quando estiver pronto, revise e crie seu primeiro aplicativo Web.

Leva alguns segundos para criar seu serviço de aplicativo. Quando estiver pronto, navegue e selecione os serviços de aplicativo na barra de navegação no lado esquerdo da tela. Em seguida, escolha seu aplicativo Web na lista. Na janela Visão geral do seu aplicativo Web, exiba e selecione o URL dele, como https://azuremol.azurewebsites.net.

Quando você seleciona o URL para seu aplicativo Web, uma nova janela ou guia do navegador é aberta. A página padrão do aplicativo Web é carregada, como mostrado na Figura 3.3. Ainda não parece uma pizza. . . .

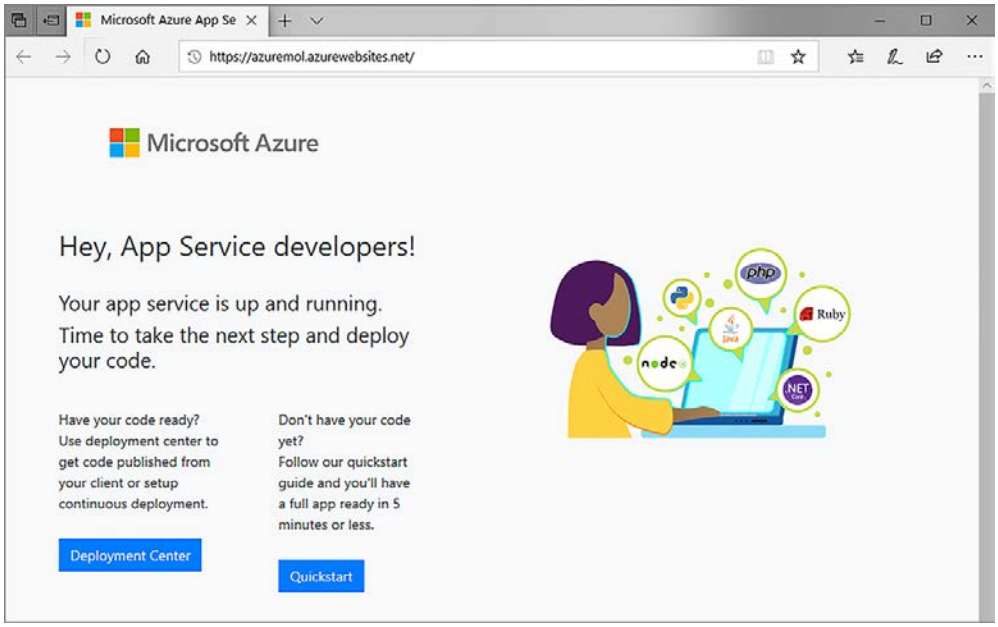


Figura 3.3 Para ver a página padrão do aplicativo Web em ação, abra um navegador da Web para o URL do seu site.

3.2.2 Implantar um site HTML de exemplo

Você tem um aplicativo Web no Azure, mas é um site padrão inútil. Como conseguir seu próprio site no Azure? Uma das formas mais comuns entre plataformas é usar o Git. A maioria dos desenvolvedores de aplicações e equipes usa um sistema de controle de origem. Em vez de armazenar arquivos em seu computador e salvar as alterações gradualmente, os sistemas de controle de origem controlam as alterações e permitem que você trabalhe com outras pessoas. Você pode criar versões de teste que não afetarão seu código de produção e reverter para versões anteriores se surgir problemas. O Git é um dos sistemas de controle de origem mais comuns. O GitHub é um serviço baseado em nuvem que permite compartilhar e contribuir com o código com o resto do mundo. A Microsoft adquiriu o GitHub em junho de 2018, mas não há nada que o obrigue a usar o GitHub com o Azure ou vice-versa. Todos os exemplos neste livro estão disponíveis no GitHub.

Neste exemplo, você cria uma cópia local do site HTML estático de exemplo e, em seguida, envia os arquivos para seu aplicativo Web do Azure. Esse fluxo de trabalho é exibido na Figura 3.4.

Figura 3.4 Você cria uma cópia local dos arquivos de exemplo do GitHub com o comando `git clone`. Para enviar esses arquivos locais para seu aplicativo Web do Azure, use `git push`.

> **Experimente agora**
>
> Para obter uma cópia da página HTML de exemplo do GitHub e enviá-la para seu aplicativo Web, conclua as etapas a seguir:

1 Abra o Cloud Shell no portal do Azure e aguarde alguns segundos para que sua sessão se conecte. Para começar, você precisa do site de exemplo HTML do GitHub.

Para *clonar* ou copiar o site HTML de exemplo do GitHub, digite o seguinte comando:

```
git clone https://github.com/foouldsy/azure-mol-samples-2nd-ed.git
```

Se essa for sua primeira vez com o Git no Cloud Shell, você precisará definir algumas configurações para que o Git entenda quem você é. Para a maioria dos exercícios neste livro, isso realmente não importa. Mas para uso com seus próprios projetos e aplicações, é uma ótima maneira de rastrear quem executa certas ações em um sistema de controle de origem. Você só precisa definir essas configurações uma vez. Digite seu endereço de email e nome completo em git config da seguinte maneira:

```
git config --global user.email "iain@azuremol.com"
git config --global user.name "Iain Foulds"
```

2 Mude para o diretório azure-mol-samples-2nd-ed que foi criado quando você clonou o repositório Git:

```
cd azure-mol-samples-2nd-ed/03/prod
```

3 Para se preparar para carregar a página HTTP de exemplo, você deve inicializar o Git e, em seguida, adicionar e confirmar seus arquivos. Não se preocupe muito com os comandos do Git. Você precisa dizer ao Git quais arquivos rastrear e adicionar e ter uma maneira de rastrear essas alterações mais tarde, se necessário:

```
git init && git add . && git commit -m "Pizza"
```

4 Agora você pode se preparar para enviar esse site HTML de exemplo para seu aplicativo Web. Primeiro, defina as credenciais de implantação. Para proteger aplicativos Web quando você usa um método de implantação como o Git ou o FTP, defina um nome de usuário e senha. Os aplicativos Web podem usar um conjunto de credenciais que são válidas em todos os planos de serviço de aplicativo no Azure ou credenciais de nível de aplicativo válidas apenas para um aplicativo.

No mundo real, recomendo que você use credenciais no nível do aplicativo para minimizar o escopo de um ataque caso uma das credenciais seja exposta. O Azure gera automaticamente as credenciais no nível do aplicativo, mas você precisa recuperar e atribuir essas credenciais sempre. Para manter as coisas simples, use uma credencial definida que você possa reutilizar nos próximos capítulos.

Crie as credenciais de implantação e especifique seu próprio nome de usuário e senha segura. O nome de usuário deve ser globalmente exclusivo. Se isso ajudar, adicione suas iniciais ao nome de usuário ou a uma convenção de nomenclatura que faça sentido para o seu ambiente.

```
az webapp deployment user set --user-name azuremol --password @azurem0l!
```

5 Em seguida, você precisa obter a URL do repositório Git do seu aplicativo Web. Insira o nome do aplicativo Web (não o nome de usuário que você criou na etapa 4) e o grupo de recursos que você especificou quando o aplicativo Web foi criado para exibir a URL do repositório do Git.

Como se cortar

No exemplo a seguir e capítulos posteriores, a barra invertida (\) significa que o comando continua na próxima linha. É uma maneira de encapsular linhas longas, e essa abordagem é usada em muitos exemplos online onde você pode copiar e colar os comandos. Você não precisa digitar as barras invertidas nos exemplos deste livro se não quiser. Basta continuar digitando os parâmetros adicionais como parte de uma grande linha.

Se você estiver usando o prompt de comando do Windows em vez de um shell Bash, não inclua as barras invertidas. Se você fizer isso, realmente não terá o resultado desejado.

```
az webapp deployment source config-local-git \
--name azuremol \
--resource-group azuremolchapter3 -o tsv
```

6 Seu aplicativo Web está configurado para funcionar com os repositórios do Git, então você precisa dizer ao Cloud Shell o que é esse repositório. No Git, você define esses locais como remotos.

Copie o URL do clone do Git da etapa 5 e, em seguida, defina esse URL como um destino para o site HTML de exemplo no Cloud Shell com o seguinte comando:

```
git remote add azure your-git-clone-url
```

7 Para carregar ou copiar arquivos com o Git, envie-os. Para onde o Git os envia? Um local remoto como você configurou na etapa anterior, como *azure*. A parte final do comando é uma ramificação, normalmente *master*. Uma ramificação no Git é o meio para controlar os diferentes modelos de trabalho em andamento. Uma prática recomendada em ambientes de produção é enviar para liberar ramificações que você pode nomear como desejar, como *dev ou staging*. Essas ramificações adicionais permitem que seu código de produção seja executado normalmente. Você pode trabalhar em novos recursos ou atualizações com segurança e sem afetar os workloads reais que seus clientes usam.

Envie o site HTML de exemplo para seu aplicativo Web:

```
git push azure master
```

8 Quando solicitado, digite a senha que você criou para as credenciais de implantação. Você pode copiar e colar a senha para minimizar os erros aqui.

Você pode ver na saída que a página do site do aplicativo Web padrão existente é removida e o site HTML de exemplo é carregado e configurado para ser executado. Veja um exemplo de saída:

```
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 510 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote: Updating branch 'master'. remote: Updating submodules.
remote: Preparing deployment for commit id 'dda01e9d86'.
remote: Generating deployment script.
remote: Generating deployment script for Web Site
remote: Running deployment command...
remote: Handling Basic Web Site deployment.
remote: Creating app_offline.htm
remote: KuduSync.NET from: 'D:\home\site\repository' to:'D:\home\site\wwwroot'
remote: Deleting file: 'hostingstart.html'
remote: Copying file: 'index.html'
remote: Deleting app_offline.htm
remote: Finished successfully.
remote: Running post deployment command(s)...
remote: Deployment successful.
To https://azuremolikf@azuremol.scm.azurewebsites.net/azuremol.git
  * [new branch]       master -> master
```

Para ver seu aplicativo Web atualizado, atualize seu site em um navegador da Web ou abra-o novamente na janela Overview (Visão geral) no portal do Azure. Deve parecer como o ótimo exemplo na Figura 3.5. Sim, o site é básico, mas o fluxo de trabalho para implantar o site HTML estático mais básico em um aplicativo Web .NET ou Node.js complexo é o mesmo.

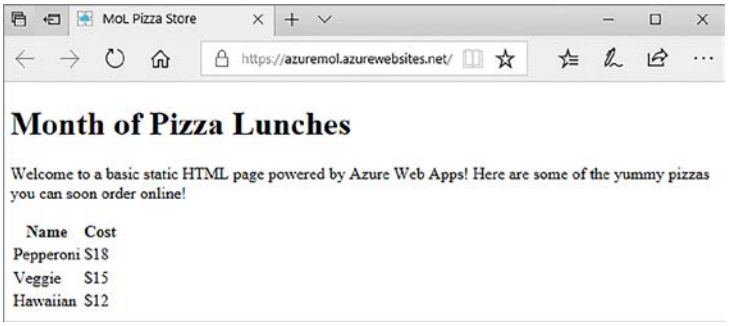


Figura 3.5 Atualize seu navegador da Web para ver a página padrão do aplicativo Web substituída pelo site HTML estático básico do GitHub.

3.3 Exibir os logs de diagnóstico

Agora que você já viu como criar um aplicativo Web básico e implantar um site HTML simples, que dizer sobre gerenciamento geral? Se você tiver problemas, seria útil ver os logs de servidor Web ou aplicação. Para ajudar a solucionar problemas de seus aplicativos, você pode gravar a saída do seu aplicativo nesses arquivos de log. Os arquivos de log podem ser visualizados em tempo real ou gravados em arquivos de log e analisados posteriormente.

Seu aplicativo Web em grande parte é executado por si só. Não há muito que você pode fazer do ponto de vista de manutenção no host da Web subjacente. Se sua aplicação tiver problemas, convém examinar os logs para ver o que está acontecendo e solucionar o problema. Com os Aplicativos Web do Azure, você configura coisas como o nível de mensagens de log para revisar, onde armazenar os logs e quanto tempo manter os logs. A Figura 3.6 descreve como gerar e exibir arquivos de log com Aplicativos Web.

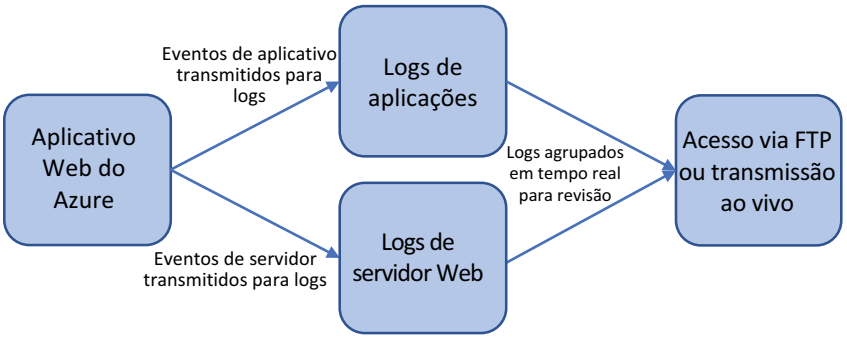


Figura 3.6 Sua aplicação pode gerar logs de aplicação e logs do servidor. Para analisar ou solucionar problemas, você pode fazer download desses logs com FTP ou visualizá-los em tempo real.

Experimente agora

Para configurar seu aplicativo Web para logs de diagnóstico, conclua as etapas a seguir:

1. No portal do Azure, selecione o aplicativo Web que você criou no exercício anterior.
2. Na janela Overview (Visão geral), role para baixo até a seção Monitoring (Monitoramento) e selecione App Service Logs (Logs de Serviço de Aplicativo).
3. Analise as opções de log disponíveis, como a verbosidade e se você deseja habilitar o rastreamento de solicitação com falha. Se você lida com o lado da infraestrutura do Azure, talvez seja necessário trabalhar com os desenvolvedores de aplicações para determinar quais logs são necessários para ajudar a solucionar problemas. Então, você pode ativar o registo relevante aqui. Os logs podem ser armazenados no sistema de arquivos local do aplicativo Web ou enviados ao Armazenamento do Azure para processamento com outra aplicação.
4. Por enquanto, ative os logs de aplicação (sistema de arquivos). Ative também os logs de servidor Web para o sistema de arquivos com um período de retenção de sete dias. O nível de erro padrão pode não mostrar nada se tudo funcionar bem, mas tome cuidado com a mudança para Depuração ou Rastreamento, pois seus logs podem ser preenchidos rapidamente e dificultar a visualização do que está acontecendo. Se você tiver um problema, aumente gradualmente o nível de log até que capturar informações suficientes para solucionar problemas sem ser sobrecarregado pelos dados de log.

Se você realmente quiser se aprofundar nos dados, poderá acessar os logs armazenados no sistema de arquivos usando o FTP. Os endereços FTP são mostrados na seção de logs de download ou na janela de visão geral do aplicativo Web. Você pode estar pensando: "FTP é uma maneira complicada de obter logs de diagnóstico. Não há uma maneira mais fácil?" Sim, há! No portal do Azure, bem onde você configurou seus logs de diagnóstico, há uma opção Log Stream (Fluxo de log). Adivinha o que ela faz? Vou dar uma dica: tem algo a ver com o streaming de seus arquivos de log.

Se você selecionar esse botão no portal do Azure, poderá escolher entre logs de aplicação e logs de servidor Web. Esses logs são lidos dos mesmos logs de diagnóstico que são gravados no arquivo. Pode levar alguns minutos para que os dados de log sejam exibidos no fluxo, e o que é exibido depende dos níveis de log que você especificar e se seu aplicativo Web gera eventos de aplicação. Para o site HTML básico, o fluxo é bastante chato, mas é um ótimo recurso para ter no navegador da Web. A Figura 3.7 mostra exemplo de streaming de logs do servidor Web no portal do Azure.

Experimente agora

Exiba o streaming de arquivos de log no portal do Azure. Talvez seja necessário atualizar a página em seu navegador da Web algumas vezes para gerar atividade nos logs.

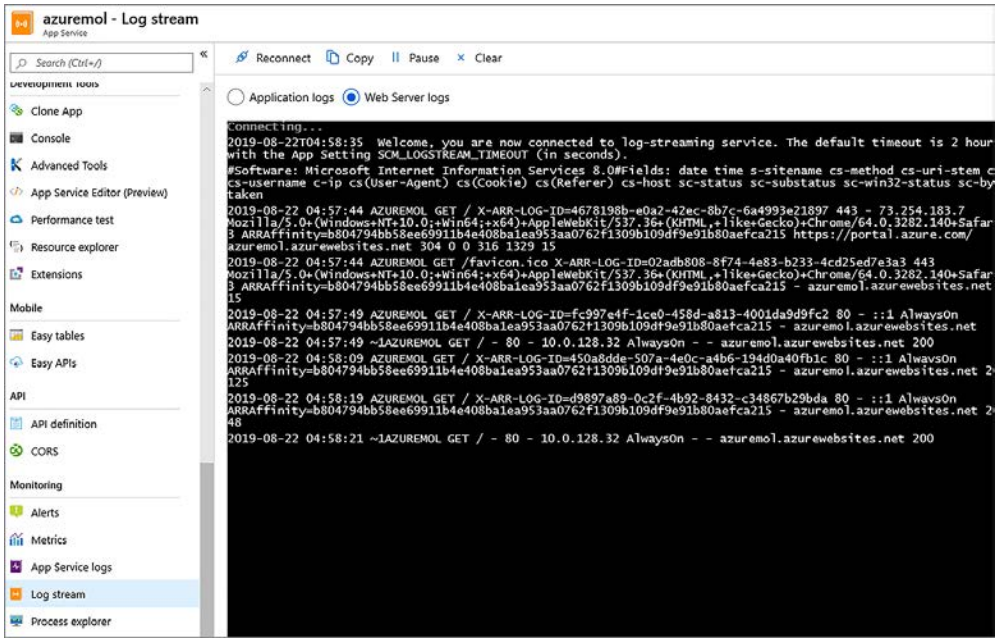


Figura 3.7 Você pode exibir os fluxos de log do servidor Web de Aplicativos Web de logs ao vivo do sua aplicação para ajudar a verificar e depurar a performance da aplicação. A caixa do console no lado direito da tela mostra os logs de streaming em tempo real do seu aplicativo Web.

À medida que você ficar mais familiarizado com o Azure e usar a CLI do Azure ou o módulo do Azure PowerShell, poderá transmitir logs com essas ferramentas. Os desenvolvedores também podem habilitar a depuração remota com o Visual Studio ou configurar o Application Insights para permitir que um aplicativo Web forneça telemetria a serviços adicionais para monitoramento e diagnóstico. A solução aqui é que, à medida que migra para soluções de PaaS como Aplicativos Web, você ainda pode obter logs de diagnóstico cruciais e dados de aplicação para solucionar problemas e monitorar a performance do seu aplicativo Web.

3.4 Laboratório: criar e usar um slot de implantação

Você viu como criar um site HTML simples e enviar a página para Aplicativos Web do Azure com o Git. E se você agora quiser adicionar alguns novos estilos de pizza e vê-los antes de publicar o site para os clientes pedirem? Veja como usar um slot de implantação para fornecer algum lugar para carregar suas alterações, analisá-las e, em seguida, trocá-las para produção:

1. Em seu aplicativo Web, escolha Slots de Implantação. ADICIONE um slot de implantação chamado Dev, mas não clone nenhuma configuração do slot de implantação existente.
2. Quando estiver pronto, selecione o slot de preparo na lista. O portal mostra as mesmas opções de configuração e de log que o slot de produção, que mostra como você pode alterar as configurações neste slot de implantação sem afetar o site ativo.

3 Desta vez, explore as opções no portal do Azure para o Centro de Implantação. Você deseja usar o Git Local para controle de origem que utiliza o Serviço de build do Serviço de Aplicativo para o slot de preparo. Isso aconteceu nos bastidores, quando você usou a CLI do Azure no exercício anterior, mas você pode implantar seu código de outro local e tem outras opções de serviço para criar essa implantação.

4 Quando terminar, copie o URI de clone Git, como https://azuremol-dev.scm.azurewebsites.net:443/azuremol.git. Observe como o repositório Git inclui o `-dev` para o slot de preparo.

Um site de desenvolvimento de exemplo está incluído nos exemplos do GitHub que você clonou anteriormente.

5 No Azure Cloud Shell, mude para o diretório de desenvolvimento da seguinte maneira:

```
cd ~/azure-mol-samples-2nd-ed/03/dev
```

6 Como antes, inicialize, adicione e confirme suas alterações no Git com os seguintes comandos:

```
git init && git add . && git commit -m "Pizza"
```

7 Crie um link novamente para o novo repositório Git no slot de preparo com `git remote add dev`, seguido pelo URL de implantação do Git de slot de preparo.

8 Use `git push dev master` para enviar suas alterações para o slot de implantação.

9 Selecione o URL para o slot de preparo na janela Overview (Visão geral) do portal do Azure. Não é uma grande alteração, mas o título da página mostra que você está vendo a versão de desenvolvimento.

10 No portal do Azure do aplicativo Web o que você acha que acontecerá se selecionar o botão Swap (Trocar), como mostrado na Figura 3.8? Faça um teste e atualizeapáginaprincipal,comohttps://azuremol.azurewebsites.net, no navegador da Web.

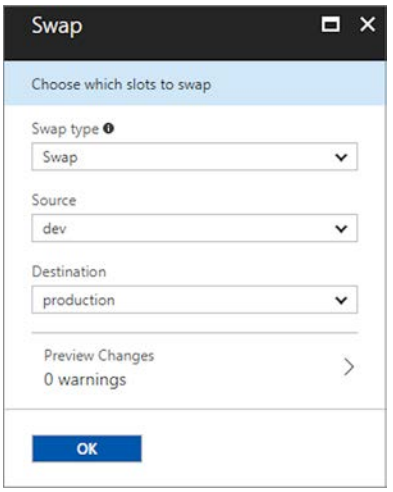


Figura 3.8 Ao trocar slots de um aplicativo Web, você escolhe as instâncias de origem e destino para alterar. Você também pode visualizar o novo visual antes de fazer as alterações em tempo real.

Slots de implantação, por trás dos bastidores

Quando você troca os slots, o que estava em tempo real no *slot de produção* agora está no slot *dev*, e o que estava no *dev* agora está em tempo real na *produção*. Nem todas as configurações podem ser trocadas, como configurações de SSL e domínios personalizados. Mas, na maior parte, os slots de implantação são uma ótima maneira de preparar e validar o conteúdo antes que ele seja publicado para os clientes. Você também pode executar uma troca com visualização, o que lhe dá a oportunidade de ter certeza de que o conteúdo trocado funcione corretamente antes de ser colocado em produção.

Para uso de produção em fluxos de trabalho de DevOps, você também pode configurar a troca automática. Aqui, quando uma confirmação de código é observada no controle de origem, como o GitHub, ela pode acionar uma compilação em um slot de implantação do Aplicativos Web do Azure. Assim que essa compilação é concluída e o aplicativo está pronto para veicular conteúdo, os slots de implantação são trocados automaticamente para ativar o código em produção. Esse fluxo de trabalho é normalmente usado com um ambiente de teste para revisar as alterações de código primeiro, não para publicar direto na produção.

4 Introdução ao Armazenamento do Azure

Temos certeza de uma coisa no mundo da TI: quando as coisas vão mal, armazenamento e rede são inevitavelmente os culpados. Eu digo isso como alguém que já foi administrador de SAN em uma das minhas vidas passadas. Eu era o melhor amigo da equipe de rede. Estou brincando (sobre ser melhor amigo), mas não importa o quão bem uma aplicação é criada e gravada: as peças de infraestrutura fundacional devem estar no lugar para apoiá-la. Nos próximos capítulos, vou explorar o Armazenamento do Azure e a Rede do Azure. Você pode querer explorar mais esses serviços para chegar ao material interessante nos capítulos posteriores, mas é válido passar algum tempo explorando e aprendendo esses serviços principais. Isso não deixará o gosto do alimento melhor, mas pode ajudar seus clientes ao pedirem uma deliciosa pizza para entrega.

Este capítulo aborda os diferentes tipos de armazenamento no Azure e quando usá-los. Também vou falar sobre as opções de redundância e replicação para o serviço de Armazenamento do Azure e como obter a melhor performance para suas aplicações.

4.1 Discos gerenciados

Anos atrás, o armazenamento de servidores era caro, lento e excessivamente complicado. Não era incomum para um fornecedor de armazenamento vender um hardware que custa centenas de milhares de dólares e levava dias, ou mesmo semanas, para um exército de seus consultores e engenheiros configurar. À medida que a virtualização começou a se enraizar no data center e a VMware e o Hyper-V tornaram-se mais aceitos, o armazenamento tornou-se frequentemente o gargalo. E isso sem falar nas incompatibilidades de firmware entre adaptadores de armazenamento no servidor e na matriz de armazenamento, caminhos de rede redundantes falhando e discos de estado sólido (SSDs) sendo considerados a única maneira de ganhar performance.

O Azure corrigiu magicamente todos esses problemas de armazenamento? Claro que não! Mas ele afasta 95% dessas preocupações e deixa você se concentrar

no desenvolvimento e na criação de experiências fantásticas para seus clientes. Este capítulo abrange os últimos conceitos que você precisa aprender.

O serviço de discos gerenciados do Azure simplifica a abordagem do armazenamento de VMs. Os discos gerenciados eliminam grande parte do trabalho dos bastidores para dar a você . . ... um disco. É só isso que você precisa saber sobre as VMs: o tamanho e a velocidade delas e a que elas se conectam. Em todo o livro e em todas as suas implantações reais, você deve sempre usar discos gerenciados para VMs. Os discos gerenciados são a opção padrão, e não há muitas boas razões para alterar esse comportamento.

Antes dos discos gerenciados, você precisava criar uma conta de armazenamento com um nome exclusivo, limitar o número de discos virtuais criados em cada uma e mover manualmente as imagens de disco personalizadas para criar VMs em diferentes regiões. Esses tipos de discos são conhecidos como *discos não gerenciados* ou *discos clássicos*. O serviço de Discos Gerenciados elimina a necessidade de uma conta de armazenamento, limita a "apenas" 50.000 discos por assinatura e permite criar VMs a partir de uma imagem personalizada entre regiões. Você também ganha a capacidade de criar e usar instantâneos de discos, criptografar dados automaticamente em repouso e usar discos de até 64 TiB.

Por que isso é importante? Se você consultar documentação ou postagens de blog antigas, talvez precise criar uma conta de armazenamento para suas VMs. Pare aí mesmo! Sim, você pode converter VMs de discos não gerenciados em discos gerenciados, mas se possível, procure começar cada projeto com discos gerenciados desde o início. O caso de uso para discos não gerenciados é mais para manter a compatibilidade com versões anteriores com implantações existentes, embora eu recomendaria converter esses workloads em discos gerenciados.

4.1.1 Discos do SO

Lembra que, se você quisesse a melhor performance, tinha que comprar SSDs? Não há nenhuma correção mágica para contornar esse requisito no Azure. Sinto muito. A verdade é que os SSDs superam muito os discos giratórios normais. Há limites físicos para a rapidez com que os discos giratórios podem . . girar. Os engenheiros da Microsoft ainda não conseguiram burlar as leis da física. Ainda há casos de uso para discos giratórios normais, como armazenamento de arquivos de baixo custo, e assim como em matrizes de armazenamento normais, as tecnologias mais recentes podem fornecer uma boa performance em um grupo de discos giratórios.

A primeira e principal escolha que você precisa fazer para uma VM do Azure é o tipo de armazenamento a ser usado:

- *Discos SSD Premium*: use discos SSD de alta performance para ter um desempenho ideal, maior IOPS e baixa latência; tipo de armazenamento recomendado para a maioria dos workloads.
- *Discos SSD padrão*: usam SSDs padrão e oferecem performance consistente em comparação com unidades de disco rígido (HDDs). Esses discos são excelentes para workloads de desenvolvimento e testes ou uso de produção de acordo com o orçamento e de baixa demanda.
- *Discos HDD padrão*: use discos giratórios normais para obter acesso a dados mais raros, como arquivos ou backups.

O tamanho da VM que você escolher ajuda a determinar o tipo de armazenamento que pode ser selecionado. De volta ao capítulo 2, quando criou uma VM, você escolheu

um tamanho que forneceu uma VM rapidamente. O padrão era provavelmente algo como uma VM série D2S_v3, que dava acesso a discos SSD Premium. Como saber quais VMs podem acessar discos SSD Premium? Procure um *s* de *S*SD no tamanho da VM. Há algumas exceções à regra, mas esse é um bom padrão a ser seguido. Os exemplos a seguir ajudam você a identificar quais VMs podem acessar discos Premium e quais VMs podem acessar SSDs ou HDDs padrão:

- As VMs série D2S_v3, F*s*, GS e L*s* podem acessar discos SSD Premium.
- As VMs série D, A, F e M só podem acessar discos SSD ou HDD padrão.

Se você selecionar um tamanho de VM que possa usar discos SSD Premium, não há nenhuma obrigação de fazer isso. Você pode criar e usar discos SSD ou HDD padrão. Ao escolher discos SSD Premium, você prova a aplicação e tem a opção de usar SSDs de alta performance, pois precisa deles sem a necessidade de redimensionar suas VMs e incorrer em um pouco de tempo de inatividade no processo. Todos os tamanhos de VM podem usar discos SSD padrão.

Disco do SO efêmero

Há um tipo especial de disco do sistema operacional chamado *disco efêmero*. De certo modo, ele ainda é um disco gerenciado, mas é local para o host do Azure subjacente. Esse fato torna o disco efêmero muito rápido, com baixa latência.

Como os dados não são gravados em uma matriz de armazenamento remota, os dados podem não persistir durante inicializações de VM, se você migrar para outro host subjacente. Discos efêmeros são ótimos para workloads sem estado que podem gerenciar a inicialização sempre com uma imagem limpa a cada vez e não precisam armazenar dados localmente para acessos entre reinicializações.

Apenas alguns tamanhos de VM suportam discos efêmeros, mas não há nenhum custo adicional para usá-los, e eles estão disponíveis em todas as regiões. Você perde algumas funcionalidades para coisas como o Azure Site Recovery e Azure Disk Encryption (capítulos 13 e 14, respectivamente), mas se você quiser armazenamento em alta velocidade e baixa latência, teste os discos efêmeros.

Experimente agora

Como você pode dizer quais tamanhos de VM estão disponíveis para você? No portal do Azure, abra o Cloud Shell. Digite o seguinte comando (sinta-se livre para usar sua própria região):

```
az vm list-sizes --location eastus --output table
```

Lembre-se de que qualquer tamanho com um s fornece acesso a discos SSD Premium.

4.1.2 Discos temporários e discos de dados

Agora que você já descobriu o nível de performance necessário para suas aplicações, vou falar sobre mais algumas peças do quebra-cabeça. Os discos são conectados de duas maneiras:

- *Discos temporários*: cada VM tem automaticamente o armazenamento SSD local anexado do host subjacente que oferece uma pequena quantidade de armazenamento de alta performance. Tome muito cuidado ao usar esse disco

temporário. Como o nome indica, esse disco pode não persistir com a VM. Se a VM mudar para um novo host em um evento de manutenção, um novo disco temporário será conectado, e todos os dados armazenados nela serão perdidos. O disco temporário foi concebido para ser um espaço de rascunho ou cache de aplicações.

- *Discos de dados*: todos os discos criados especificamente e anexados à VM atuam como esperado em termos de partições, sistemas de arquivos e pontos de montagem persistentes. Os discos de dados são reanexados à medida que a VM se move pelo data center do Azure, e eles estão onde a maioria dos seus dados e aplicações deve ser armazenada. Você ainda pode usar espaços de armazenamento ou RAID de software para agrupar discos de dados no nível da VM para ter uma performance ainda melhor.

Há um tipo específico de disco de dados que você pode conectar a uma VM se precisar de performance máxima e baixa latência: Discos Ultra. Esses discos são uma etapa acima dos discos SSD Premium e estão disponíveis apenas para discos de dados. Os Discos Ultra foram criados para grandes bancos de dados e workloads com uso intenso de dados, como SAP HANA. Qual é a rapidez deles? No momento da gravação, os Discos Ultra podem ter tamanho de até 64 TiB e fornecer até 160.000 IOPS por disco com uma taxa de transferência máxima de 2.000 MBps.

4.1.3 Opções de cache de disco

Também é importante considerar o disco do sistema operacional que vem com a VM. Ao criar uma VM, você sempre obtém pelo menos um disco: aquele em que o sistema operacional em si está instalado. É tentador usar esse disco para instalar suas aplicações ou gravar arquivos de log nele. A menos que você execute uma pequena implementação de prova de conceito, não execute suas aplicações no disco do sistema operacional. Há uma boa chance de você não obter a performance desejada.

Os discos no Azure podem ter uma política de cache definida neles. Por padrão, o disco do sistema operacional tem cache de *leitura/gravação* aplicado. Esse tipo de cache normalmente não é ideal para workloads de aplicação que gravam arquivos de log ou bancos de dados, por exemplo. Os discos de dados, por outro lado, não têm *nenhuma* política de cache padrão. Essa é uma boa política para workloads que executam muitas gravações. Você também pode aplicar uma política de cache *somente leitura*, que é mais adequada para workloads de aplicações que leem principalmente os dados dos discos.

Em geral, sempre anexe e use discos de dados para instalar e executar suas aplicações. Mesmo a ausência de política de cache padrão provavelmente oferece melhor performance do que a política de cache de leitura/gravação do disco do sistema operacional.

4.2 Adicionar discos a uma VM

Nesta seção, você verá como adicionar discos a uma VM ao criá-la. No capítulo 2, você criou uma VM com o portal do Azure. Desta vez, use a CLI do Azure para criar uma VM. A CLI do Azure fornece uma maneira rápida de criar uma VM e anexar um disco de dados ao mesmo tempo.

Experimente agora

Para criar uma VM e ver os discos de dados em ação, conclua as etapas a seguir:

1. No Azure Cloud Shell, crie um grupo de recursos com `az group create`, atribuindo um nome ao grupo junto com um local:

   ```
   az group create --name azuremolchapter4 --location eastus
   ```

2. Crie uma VM com o comando `az vm create`. O parâmetro final, `--data-disk-sizes-gb`, permite criar um disco de dados junto com a VM. No laboratório de fim de capítulo, você pode se conectar a essa VM e inicializar os discos.
   - Você pode criar uma VM do Linux ou do Windows para este exercício. Se você estiver familiarizado com Linux ou quiser aprender a inicializar e preparar um disco para Linux, use o seguinte comando para criar uma VM Ubuntu LTS:

     ```
     az vm create \
       --resource-group azuremolchapter4 \
       --name storagevm \
       --image UbuntuLTS \
       --size Standard_B1ms \
       --admin-username azuremol \
       --generate-ssh-keys \
       --data-disk-sizes-gb 64
     ```

   - Se você estiver mais familiarizado com o Windows, use o seguinte comando para criar uma VM do Windows Server 2019. Você pode usar RDP para se conectar à VM para configurar os discos mais tarde:

     ```
     az vm create \
       --resource-group azuremolchapter4 \
       --name storagevm \
       --image Win2019Datacenter \
       --size Standard_B1ms \
       --admin-username azuremol \
       --admin-password P@ssw0rd! \
       --data-disk-sizes-gb 64
     ```

   - Demora alguns minutos para criar a VM. A VM já tem um disco de dados anexado e pronto para uso.

E se você quiser adicionar outro disco de dados após algumas semanas ou meses? Use a CLI do Azure novamente para ver como adicionar um disco rapidamente. O processo é o mesmo para uma VM do Linux ou do Windows. Tudo o que você faz é dizer ao Azure para criar um novo disco e anexá-lo à sua VM.

Experimente agora

Adicione um disco de dados adicional à sua VM, conforme mostrado a seguir.

Crie um novo disco com o comando `az vm disk attach`. Atribua um nome e um tamanho ao disco. Lembre-se da discussão anterior sobre discos padrão e Premium? No exemplo a seguir, você cria um disco SSD Premium:

```
az vm disk attach \
    --resource-group azuremolchapter4 \
    --vm-name storagevm \
    --name datadisk \
    --size-gb 64 \
    --sku Premium_LRS \
    --new
```

Reconhece a última parte desse tipo de armazenamento? *LRS* significa *armazenamento com redundância local*. Vamos analisar as opções de redundância na seção 4.3.3.

Em dois comandos, você criou uma VM com a CLI do Azure que incluía um disco de dados e, em seguida, simulou como anexar um disco de dados adicional mais tarde. Entretanto, você não pode gravar dados nesses discos imediatamente só porque os anexou. Como em qualquer disco, seja um disco físico anexado a um servidor na infraestrutura local ou um disco virtual anexado a uma VM, você precisa inicializar o disco e criar uma partição e um sistema de arquivos. Você pode fazer isso no exercício opcional, no laboratório de fim de capítulo.

4.3 Armazenamento do Azure

Armazenamento pode não parecer o tópico óbvio a ser examinado para criar e executar aplicações, mas é um serviço amplo que abrange muito mais do que você pode esperar. O serviço de Armazenamento do Azure oferece muito mais do que apenas algum lugar para armazenar arquivos ou discos virtuais para suas VMs.

Veja o que sua pizzaria fictícia precisa para criar um aplicativo que processa pedidos de clientes para retirada ou entrega. O aplicativo precisa de um armazenamento de dados que contém as pizzas disponíveis, lista de coberturas e preços. Conforme os pedidos são recebidos e processados, a aplicação precisa de uma maneira de enviar mensagens entre os componentes da aplicação. O site principal precisa de imagens que dão água na boca para mostrar aos clientes como são as pizzas. Como você pode ver na Figura 4.1, o Armazenamento do Azure tem diversos recursos de armazenamento e pode cobrir todas essas três necessidades.

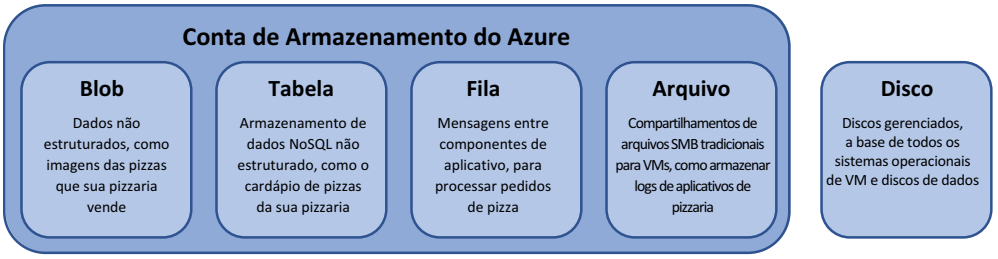


Figura 4.1 Uma conta de Armazenamento do Azure permite que você crie e use uma grande variedade de recursos de armazenamento, muito além de apenas um lugar para armazenar arquivos.

- *Armazenamento de blobs*: para dados não estruturados, como arquivos de mídia e documentos. As aplicações podem armazenar dados no armazenamento de blobs, como imagens, e renderizá-los. Você pode armazenar imagens de suas pizzas no armazenamento de blobs.
- *Armazenamento de tabelas*: para dados não estruturados em um armazenamento de dados NoSQL. Como acontece com qualquer debate sobre armazenamentos de dados SQL versus NoSQL, planeje sua aplicação e estime os requisitos de performance quando se trata de processamento de grandes quantidades de dados. Você pode armazenar a lista de pizzas em seu menu no armazenamento de tabelas. A seção 4.3.1 explora o NoSQL em mais detalhes.
- *Armazenamento de filas*: para aplicações de nuvem se comunicarem entre várias camadas e componentes de forma confiável e consistente. Você pode criar, ler e excluir mensagens que passam entre os componentes da aplicação. Você pode usar o armazenamento de filas para passar mensagens entre o frontend da Web quando um cliente faz um pedido e o back-end para processar e assar as pizzas.

- *Armazenamento de arquivos*: para um bom compartilhamento de arquivos de bloco de mensagens de servidor (SMB) antigo, acessível por ambas as plataformas Windows e Linux/macOS, muitas vezes usado para centralizar a coleção de logs de VMs.

O Armazenamento do Azure para VMs é simples. Você cria e usa discos gerenciados do Azure, um tipo de disco rígido virtual (VHD) que abstrai muitas considerações de design em torno da performance e distribui os discos virtuais pela plataforma. Você cria uma VM, anexa todos os discos de dados gerenciados e permite que a plataforma do Azure descubra redundância e disponibilidade.

4.3.1 Armazenamento de tabelas

Vamos discutir alguns tipos de armazenamento de dados. O primeiro é o *armazenamento de tabelas*. A maioria das pessoas provavelmente está familiarizada com um banco de dados SQL tradicional, como o Microsoft SQL Server, MySQL ou PostgreSQL. Esses são *bancos de dados relacionais*, compostos de uma ou mais tabelas que contêm uma ou mais linhas de dados. Os bancos de dados relacionais são comuns no desenvolvimento de aplicações e podem ser projetados, visualizados e consultados de maneira estruturada — o *S* em *SQL* (para linguagem de consulta estruturada).

Os bancos de dados NoSQL são um pouco diferentes. Eles não seguem a mesma abordagem estruturada, e os dados não são armazenados em tabelas onde cada linha contém os mesmos campos. Existem diferentes implementações de bancos de dados NoSQL; os exemplos incluem MongoDB e CouchDB. As vantagens elogiadas dos bancos de dados NoSQL são que eles são dimensionados horizontalmente (o que significa que você pode adicionar mais servidores em vez de adicionar mais memória ou CPU), podem lidar com grandes quantidades de dados e são mais eficientes no processamento desses grandes conjuntos de dados.

A forma como os dados são armazenados em um banco de dados NoSQL pode ser definida em algumas categorias:

- *Chave-valor*, como Redis
- *Coluna*, como Cassandra
- *Documento*, como MongoDB

Cada abordagem tem prós e contras do ponto de vista de desempenho, flexibilidade ou complexidade. Uma tabela de armazenamento do Azure usa um armazenamento de chave-valor e é uma boa introdução aos bancos de dados NoSQL quando você está acostumado a um banco de dados SQL, como o Microsoft SQL ou o MySQL.

Você pode baixar e instalar o Gerenciador de Armazenamento do Microsoft Azure em https://azure.microsoft.com/features/storage-explorer, se quiser visualizar os dados. Não precisa fazer isso agora. O Gerenciador de Armazenamento é uma ótima ferramenta para ver tabelas e filas em ação. Neste capítulo, eu não quero levá-lo muito longe dos bancos de dados NoSQL. No capítulo 10, exploramos alguns bancos de dados NoSQL interessantes em detalhes com o Azure Cosmos DB. Na verdade, no exercício a seguir, você usa a API do Cosmos DB para se conectar ao Armazenamento do Azure e

criar uma tabela. O uso de tabelas do Azure é mais uma introdução aos bancos de dados NoSQL do que um exemplo sólido de uso de produção.

Por enquanto, vamos executar uma aplicação de exemplo rápido para ver como você pode adicionar e consultar dados, exatamente como faria com uma aplicação real. Esses exemplos são básicos, mas mostram como você pode armazenar os tipos de pizza que vende e quanto custa cada pizza. Em vez de usar algo grande, como o Microsoft SQL Server ou o MySQL, vamos usar um banco de dados NoSQL com o armazenamento de tabelas do Azure.

Experimente agora

Para ver as tabelas do Azure em ação, conclua as etapas a seguir:

1. Abra o portal do Azure em um navegador da Web e abra o Cloud Shell.
2. No capítulo 3, você obteve uma cópia dos exemplos do Azure no GitHub. Se você não fez isso, pegue uma cópia da seguinte forma:

   ```
   git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
   ```

3. Mude para o diretório que contém os exemplos de Armazenamento do Azure:

   ```
   cd ~/azure-mol-samples-2nd-ed/04
   ```

4. Instale algumas dependências do Python, se elas ainda não estiverem instaladas. Aqui você instala o pacote `azurerm`, que manipula a comunicação que permite criar e gerenciar recursos do Azure, e dois pacotes `azure`, que são os SDKs Python subjacentes para o Azure CosmosDB e o armazenamento:

   ```
   pip install --user azurerm azure-cosmosdb-table azure-storage-
   ➥queue==2.1.0
   ```

   O que significa `--user` quando você instala os pacotes? Se você usar o Azure Cloud Shell, não poderá instalar pacotes no sistema principal. Você não tem permissões. Em vez disso, os pacotes são instalados no ambiente do usuário. Essas instalações de pacotes persistem em sessões e permitem que você use todos os SDKs puros do Azure nesses exemplos.

5. Execute a aplicação Python de exemplo para tabelas. Siga as instruções para comer uma pizza:

   ```
   python storage_table_demo.py
   ```

Serpentes em um plano

Python é uma linguagem de programação amplamente utilizada que é muito usada em classes de "Introdução à Ciência da Computação". Se você trabalha principalmente nas operações ou na administração de TI, pense em Python como uma poderosa linguagem de script que funciona em SOs. Python não é apenas para scripts; ele também pode ser

usado para criar aplicações complexas. Por exemplo, a CLI do Azure que você está usando é gravada em Python.

Eu uso Python para alguns dos exemplos neste livro porque eles devem funcionar fora do Cloud Shell sem modificação. As distribuições macOS e Linux incluem Python nativamente. Os usuários do Windows podem baixar e instalar rapidamente Python e executar esses scripts localmente. Python é ótimo para aqueles com pouca experiência de programação, bem como desenvolvedores mais experientes em outras linguagens. A documentação do Azure para o Armazenamento do Azure e muitos outros serviços fornece suporte para diversas linguagens, incluindo .NET, Java e Node.js. Você não está limitado a usar Python ao criar seus próprias aplicações que usam tabelas.

O Quick Python Book, 3ª edição, de Naomi Ceder (http://mng.bz/6QZA), pode ajudar você a se atualizar, se quiser saber mais. Há também um curso em vídeo sobre programação de *Get com Python em movimento*, de Ana Bell (http://mng.bz/oPap).

4.3.2 Armazenamento de filas

As tabelas do Azure são legais quando você começa a mergulhar no mundo do desenvolvimento de aplicações em nuvem. À medida que você começa a criar e gerenciar aplicações nativamente na nuvem, normalmente quebra uma aplicação em componentes menores que podem dimensionar e processar dados por conta própria. Para permitir que esses componentes se comuniquem e transmitam dados, alguma forma de fila de mensagens normalmente é necessária. Insira filas do Azure.

O serviço de filas do Azure permite criar, ler e excluir mensagens que carregam pequenos blocos de dados. Essas mensagens são criadas e recuperadas por diferentes componentes de aplicação à medida que transmitem os dados. As filas do Azure só excluirão a mensagem depois que uma aplicação acabar de processar os dados dela.

Experimente agora

Para ver as filas do Azure em ação, execute o seguinte script Python a partir do mesmo diretório azure-samples/4. Siga os prompts para ver as mensagens gravadas, lidas e excluídas da fila:

```
python storage_queue_demo.py
```

Continue a aplicação de exemplo que lida com pedidos de pizza. Você pode ter um componente de aplicação frontend com o qual os clientes interagem para encomendar a pizza e uma fila de mensagem que transmite mensagens para um componente de aplicação de back-end para processar esses pedidos. Conforme os pedidos são recebidos, as mensagens na fila podem ser visualizadas como mostrado na Figura 4.2.

ID	Message Text	Insertion Time (UTC)	Expiration Time (UTC)	Dequeue Count	Size
ca57a12c-21b8-4640-9e07-4fc3a81c8dd5	Veggie pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	21 B
d68f90a9-1d5a-4a0e-af79-f285efa2aca2	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
7f3c6f4a-9d47-488f-9344-1cb5bbec0fa4	Hawiian pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	22 B
63f07e06-ab0d-48c0-81c9-019d4255f335	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
66b48f73-d136-4d82-9b41-f32f93f3d725	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B

Figura 4.2 As mensagens são recebidas do componente de aplicação frontend que detalha a pizza que cada cliente pediu na propriedade `Message Text`.

Conforme o componente de aplicação de back-end processa cada pedido de pizza deliciosa, as mensagens são removidas da fila. A Figura 4.3 mostra como fica a fila quando você tem uma pizza vegetariana no forno e essa primeira mensagem é removida.

ID	Message Text	Insertion Time (UTC)	Expiration Time (UTC)	Dequeue Count	Size
d68f90a9-1d5a-4a0e-af79-f285efa2aca2	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
7f3c6f4a-9d47-488f-9344-1cb5bbec0fa4	Hawiian pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	22 B
63f07e06-ab0d-48c0-81c9-019d4255f335	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
66b48f73-d136-4d82-9b41-f32f93f3d725	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B

Figura 4.3 À medida que cada mensagem é processada, ela é removida da fila. A primeira mensagem mostrada na Figura 4.2 foi removida depois de ser processada pelo componente de aplicação de back-end.

4.3.3 Disponibilidade e redundância de armazenamento

Os data centers do Azure são projetados para serem tolerantes a falhas com conexões de Internet redundantes, geradores de energia, vários caminhos de rede, matrizes de armazenamento e assim por diante. Você ainda precisa fazer sua parte ao projetar e executar aplicações. Com o Armazenamento do Azure, você escolhe o nível de redundância de armazenamento necessário. Esse nível varia para cada aplicação e com a importância dos dados. Veja as opções de redundância de armazenamento disponíveis:

- *LRS (armazenamento com redundância local)*: seus dados são replicados três vezes dentro do data center único no qual a conta de armazenamento foi criada. Essa opção fornece redundância no caso de uma única falha de hardware, mas se o data center inteiro for desligado (raro, mas possível), seus dados serão perdidos também.
- *Armazenamento com redundância de zona (ZRS)*: o próximo nível acima de LRS é replicar seus dados três vezes em dois ou três data centers em uma região (onde existem vários data centers em uma região) ou entre regiões. O ZRS também está disponível nas zonas de disponibilidade, que exploramos em mais detalhes no capítulo 7.

- *GRS (armazenamento com redundância geográfica)*: com GRS, seus dados são replicados três vezes na região primária em que o armazenamento é criado e, em seguida, replicados três vezes em uma região emparelhada. A região emparelhada geralmente está a centenas ou milhares de quilômetros de distância. Por exemplo, o oeste dos EUA está emparelhado com o leste dos EUA, o norte da Europa está emparelhado com a Europa ocidental e o sudeste asiático está emparelhado com o leste asiático. O GRS oferece uma ótima opção de redundância para aplicações de produção.
- *RA-GRS (armazenamento com redundância geográfica de acesso de leitura)*: é a opção Premium de redundância de dados. Seus dados são replicados em regiões emparelhadas como GRS, mas você também pode ter acesso de leitura aos dados nessa zona secundária.

4.4 Laboratório: explorar o armazenamento do Azure

Aqui está a chance de testar suas habilidades. Escolha uma das tarefas a seguir para concluir seu exercício de laboratório.

4.4.1 Focado na VM

Se você quiser entrar em uma VM e ver que o processo para inicializar um disco e criar um sistema de arquivos é o mesmo que qualquer outra VM com a qual tenha trabalhado, experimente um destes exercícios:

1. Faça login na VM criada na seção 4.2. Dependendo de sua escolha, você vai se conectar com SSH ou RDP.
2. Inicialize o disco e crie uma partição.
   - No Linux, o fluxo é `fdisk`, `mkfs` e `mount`.
   - No Windows, use qualquer sequência com a qual você esteja familiarizado — provavelmente Disk Management (Gerenciamento de Discos) > Initialize (Inicializar) > Create Volume (Criar Volume) > Format (Formato).

4.4.2 Focado no desenvolvedor

Se você for mais do que um desenvolvedor e não quiser descobrir como inicializar discos de dados em uma VM, volte para o Cloud Shell e explore as duas demonstrações Python que usam tabelas e filas. Mesmo que você seja novo no Python, deve ser capaz de acompanhar o que está acontecendo:

- Pense em alguns cenários em que você poderia implementar tabelas ou filas em suas próprias aplicações. O que seria preciso para criar aplicações nativos da nuvem com componentes de aplicações individuais que pudessem usar filas, por exemplo?
- Modifique um dos exemplos que lhe interessam, para criar um item de menu de pizza adicional (se for uma tabela) ou uma nova mensagem de pedido de pizza (se for uma fila).

5 Noções básicas de rede do Azure

No capítulo 4, você explorou o serviço de Armazenamento do Azure. Um dos outros serviços principais para aplicações de nuvem é a Rede do Azure. Há muitos recursos de rede poderosos no Azure para proteger e rotear o tráfego em uma escala verdadeiramente global. Esses recursos são projetados para ajudar a focar em como criar e manter seus aplicativos, para que você não tenha que se preocupar com detalhes como endereços IP e tabelas de rotas. Se você criar e executar uma loja online para lidar com pedidos de pizza, ela deverá transmitir com segurança os dados do cliente e processar transações de pagamento.

Neste capítulo, vamos analisar as redes virtuais e as sub-redes do Azure e ver como criar interfaces de rede virtual. Para proteger e controlar o fluxo de tráfego, você cria grupos e regras de segurança de rede. Se a rede for nova para você ou se faz tempo que você teve que trabalhar com endereços IP e cartões de rede, este capítulo pode demorar um pouco mais. Ele tem muitos exercícios do tipo Faça um teste. Convém ler capítulo, no entanto, como rede é um assunto fundamental para muitos serviços do Azure.

5.1 Componentes de rede virtual

Pense em quantos cabos estão atrás da sua mesa do computador ou em seu Home Theater. Agora pense em todos os cabos necessários para conectar os computadores em determinado andar de um prédio de escritórios. E em todo o prédio de escritórios? Você já esteve em um data center ou viu fotos de um? Tente imaginar o tamanho dos data centers do Azure. Agora tente imaginar dezenas de data center do Azure em todo o mundo. Matemática não é o meu ponto forte, então eu não posso calcular quantos quilômetros de cabos de rede são usados para transportar todo o tráfego no Azure.

A conectividade de rede é uma parte crucial da vida moderna. No Azure, a rede é fundamental para como tudo se comunica. Para todos os milhares de dispositivos

de rede física e quilômetros de cabos de rede que conectam tudo em um data center do Azure, você trabalha com recursos de rede *virtual*. Como? Redes definidas por software. Quando você cria uma VM ou um aplicativo Web, um técnico não precisa ir ao data center do Azure para conectar cabos fisicamente para você e atribuir endereços IP (embora isso seja engraçado de assistir!). Em vez disso, todos os recursos de rede que definem todo o ambiente de rede são tratados logicamente pela plataforma do Azure. A Figura 5.1 mostra os componentes de rede virtual que você criará durante este capítulo.

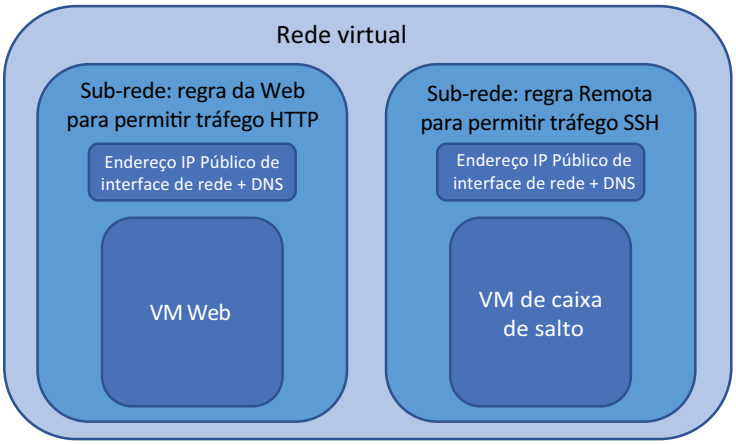


Figura 5.1 Conexões de rede definidas por software no Azure

Alguns dos componentes de rede serão abstraídos se você usar recursos de PaaS. Os componentes principais que você usa para VMs são os seguintes:

- Redes virtuais e sub-redes (incluindo pools de endereços IP)
- Placas de interface de rede virtual
- Um ou mais endereços IP públicos
- Nome DNS interno e nomes DNS públicos opcionais para resolução de nomes externos
- Grupos de segurança de rede e regras, que protegem e controlam o fluxo de tráfego de rede como um firewall normal faz

5.1.1 Redes e sub-redes virtuais

Quando você criou uma VM no capítulo 2, não foi necessário ajustar nenhum parâmetro de rede. A plataforma do Azure pode criar esses recursos para você com nomes padrão e escopos de endereço IP. Nesta seção, você criará os recursos de rede antes do tempo e ver como eles se reúnem para uma VM.

Experimente agora

A rede é muitas vezes mais fácil de visualizar quando você a vê em ação. Você vai usar o portal do Azure para começar (são necessárias algumas etapas separadas), mas você verá a eficiência da CLI do Azure mais adiante no capítulo.

Não se preocupe muito sobre como usar seus próprios espaços de endereço ou nomes de DNS personalizado. Para criar sua rede e sub-rede virtual, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione Criar um recurso no canto superior esquerdo do painel.
2. Selecione Networking (Rede) na lista de serviços do Marketplace e escolha Virtual Network (Rede Virtual).
3. Insira um nome para a rede virtual, como `vnetmol`.
4. Para brincar um pouco mais, mude o espaço de endereçamento para 10.0.0.0/16.

Intervalos de endereços IP

As redes virtuais abrangem um determinado intervalo de IPs — um espaço de endereçamento. Se você já viu um endereço IP, pode ter notado a máscara de sub-rede: muitas vezes algo como `255.255.255.0`. Essa máscara de sub-rede é frequentemente usada em uma forma abreviada que especifica o tamanho do intervalo, como `/24`.

O portal do Azure assume um espaço de endereçamento padrão `/24`. Você deseja aumentar o número de intervalos IP adicionais aqui sem muito conhecimento de rede e, portanto, aumenta o espaço de endereçamento para `/16`. Você não dá esse tipo de endereço IP diretamente a uma VM. Na próxima etapa, você cria uma sub-rede que abrange uma seção menor desse espaço de endereçamento.

Se os espaços de endereço de rede forem totalmente estranhos para você, não se preocupe. Na maior parte, você não terá que lidar com eles diariamente. A governança sensata do Azure pode funcionar da mesma forma que no seu mundo de TI existente na infraestrutura local: um grupo de pessoas pode gerenciar as redes virtuais do Azure e você descarta suas aplicações em um espaço pré-criado.

5. Crie um grupo de recursos, como `azuremolchapter5`, e selecione uma região do Azure perto de você.
6. Informe um nome de sub-rede, como `websubnet`, e insira o intervalo de endereços de sub-rede `10.0.1.0/24`. Esse intervalo de endereços faz parte da rede virtual mais ampla especificada anteriormente. Posteriormente, você adicionará outra sub-rede.
7. Veja algumas das outras opções, como proteção de negação de serviço (DDoS) distribuída, pontos de extremidade de serviço e firewall do Azure. Deixe os padrões por enquanto, mas espero que este exemplo dê algumas dicas sobre o que é possível além de uma rede virtual básica.
8. Quando estiver pronto, crie a rede virtual e configure a sub-rede.

5.1.2 Placas de interface de rede virtual

Agora que você criou uma rede virtual e uma sub-rede, precisa conectar uma VM. Assim como faz com um desktop, laptop ou tablet normal, você usa uma placa de interface de rede (NIC) para se conectar à rede virtual. E não, não há Wi-Fi gratuito! Mas há tamanhos de VM no Azure que atualmente fornecem até oito NICs com velocidades de até 32 Gbps. Mesmo se eu fosse bom em matemática, eu poderia dizer que esses números representam uma boa largura de banda.

Você pode se perguntar por que criaria cada um desses recursos antes do tempo. Você pode fazer tudo isso quando cria uma VM. Isso é verdade, mas dê um passo para trás e pense sobre os recursos de rede como recursos de longa duração.

Os recursos de rede existem separadamente dos recursos de VM e podem persistir além do ciclo de vida de uma determinada VM. Esse conceito permite criar os recursos de rede fixos e criar, excluir e criar novamente uma VM que mantém os mesmos recursos de rede, como endereços IP e nomes DNS. Pense em uma VM de laboratório ou em um ambiente de desenvolvimento e teste. Você pode reproduzir rapidamente o mesmo ambiente exato, pois somente a VM é alterada.

Experimente agora

Para criar uma NIC, conclua as etapas a seguir:

1. No portal do Azure, (), selecione Create a Resource (Criar um Recurso) no canto superior esquerdo do painel.
2. Procure e selecione Network Interface (Interface de Rede), e então selecione Create (Criar).
3. Dê um nome para sua interface de rede, como `webvnic`. Selecione a rede e a sub-rede virtual que você criou no exercício anterior.
4. Falei sobre os recursos de longa duração anteriormente. Agora você pode ver como eles funcionam. Crie uma atribuição de endereço IP estático que use o endereço `10.0.1.4`.

DICA Por que .4? E os três primeiros endereços no espaço de endereçamento? O Azure reserva os três primeiros endereços IP em cada intervalo para seu próprio gerenciamento e roteamento. O primeiro endereço utilizável que você pode usar em cada intervalo é .4.

5. Não crie um grupo de segurança de rede por enquanto. Voltaremos a isso em alguns minutos. Se você é inteligente e sabe tudo sobre IPv6, marque a caixa Private IP Address (IPv6) (Endereço IP Privado) e forneça um nome. Caso contrário, use IPv4.
6. Selecione o grupo de recursos existente do exercício anterior e escolha criar a NIC na mesma região que a rede virtual.
7. Quando estiver pronto, crie a NIC.

Separação de funções no Azure

Você não precisa criar outros recursos de computação dentro do mesmo grupo de recursos que sua rede virtual. Pense novamente no conceito de governança do Azure que discutimos anteriormente. Você pode ter um grupo de engenheiros de rede que gerenciam todos os recursos de rede virtual no Azure. Quando cria recursos para suas aplicações, como VMs, você os cria e gerencia em seus próprios grupos de recursos.

Discutimos em capítulos posteriores alguns dos recursos de segurança e política no Azure que permitem que você defina quem pode acessar e editar determinados recursos. A ideia é que, se você não sabe, ou não quer saber, usar uma grande quantidade de recursos da rede, pode se conectar ao que é fornecido, e pronto. O mesmo se aplica a outros engenheiros ou desenvolvedores; eles podem ser capazes de ver os recursos da aplicação, mas não editá-los ou excluí-los.

Esse tipo de modelo de governança no Azure é bom, mas tome cuidado para evitar a armadilha de trabalhar em silos. Em grandes empresas, seu acesso nas linhas de departamento pode ser restrito. Mas uma das grandes vantagens dos provedores de computação na nuvem como o Azure é acelerar o tempo de implantação de aplicações, pois você não precisa esperar que os recursos de rede física sejam cabeados e configurados. Planeje ter os recursos de rede do Azure criados e configurados, e você deve ser capaz de criar e gerenciar seus recursos de aplicação sem problemas.

5.1.3 *Endereço IP público e resolução de DNS*

Ninguém pode acessar seus recursos ainda, porque nenhum endereço IP público ou nome DNS está associado a eles. Novamente, siga o princípio dos recursos de longa duração para criar um endereço IP público e um nome DNS público e, em seguida, atribua-os à sua interface de rede.

Experimente agora

Para criar um endereço IP público e uma entrada DNS para sua interface de rede, conclua as etapas a seguir:

1. No portal do Azure, (), selecione Create a Resource (Criar um Recurso) no canto superior esquerdo do painel.
2. Procure e selecione Public IP Address (Endereço IP Público) e, em seguida, selecione Create (Criar).
3. Crie um endereço SKU e IPv4 básico. SKUs padrão e endereços IPv6 devem ser usados com balanceadores de carga (capítulo 8). Não se preocupe muito com as diferenças agora.
4. Insira um nome, como webpublicip, que usa uma atribuição dinâmica.

Tipos de atribuição de endereço IP

Uma atribuição dinâmica aloca um endereço IP público quando a VM é iniciada. Quando a VM é interrompida, o endereço IP público é desalocado. Há alguns pontos importantes aqui:

- Você não terá um endereço IP público até atribuí-lo a uma VM e iniciá-lo.
- O endereço IP público poderá ser alterado se você parar, desalocar e iniciar a VM.

Uma atribuição *estática* é um endereço IP público alocado sem uma VM associada, e que esse endereço não mudará: Essa atribuição é útil para cenários em que você está usando um certificado SSL mapeado para um endereço IP, ou um nome DNS personalizado e um registro que aponta para o endereço IP.

Agora, você está usando uma única VM. Para uso de produção, você idealmente executará sua aplicação em várias VMs com um balanceador de carga na frente delas. Nesse cenário, o endereço IP público é atribuído ao balanceador de carga e normalmente cria uma atribuição estática nesse ponto.

5 Insira um nome DNS exclusivo. Esse nome forma o nome de domínio totalmente qualificado (FQDN) do recurso baseado na região do Azure em que você o cria. Se você criar um rótulo de nome DNS chamado azuremol na região leste dos EUA, por exemplo, o FQDN se tornou azuremol.eastus.cloudapp.azure.com.

Entradas de DNS

Que tal um nome DNS personalizado? O FQDN padrão não é exatamente fácil de usar. Use um endereço IP público estático e, em seguida, crie um registro CNAME na zona do seu DNS registrado. Você mantém o controle do registro DNS e pode criar quantas entradas quiser para suas aplicações.

Como exemplo, na zona DNS manning.com, você pode criar um CNAME para azuremol que aponta para um endereço IP público estático no Azure. Um usuário acessaria azuremol.manning.com para obter sua aplicação. Esse endereço é muito mais fácil do que webmol.eastus.cloudapp.azure.com.

6 Selecione o grupo de recursos existente do exercício anterior e escolha criar o endereço IP público na mesma região que a rede virtual.

7 Quando estiver pronto, crie o endereço IP público.

8 Associe o endereço IP público e o rótulo de nome DNS à interface de rede que você criou na seção 5.1.2. Navegue até e selecione Resource Group (Grupo de Recursos) na barra de navegação no lado esquerdo do portal do Azure. Então, escolha o grupo de recursos no qual você criou os recursos de rede, como azuremolchapter5.

9 Selecione seu endereço IP público na lista de recursos e escolha Associate (Associar).

10 Escolha associar-se a uma interface de rede (mas preste atenção no que mais você pode associar ao endereço IP público). Em seguida, escolha a interface de rede que você criou, como webvnic.

Depois de alguns segundos, a janela de endereço IP público é atualizada para mostrar que o endereço IP agora está associado à sua interface de rede. Se você selecionou dinâmico como o tipo de atribuição, o endereço IP ainda está em branco, como mostrado na Figura 5.2. Lembre-se de que um endereço IP público é alocado quando uma VM associada está ligada.

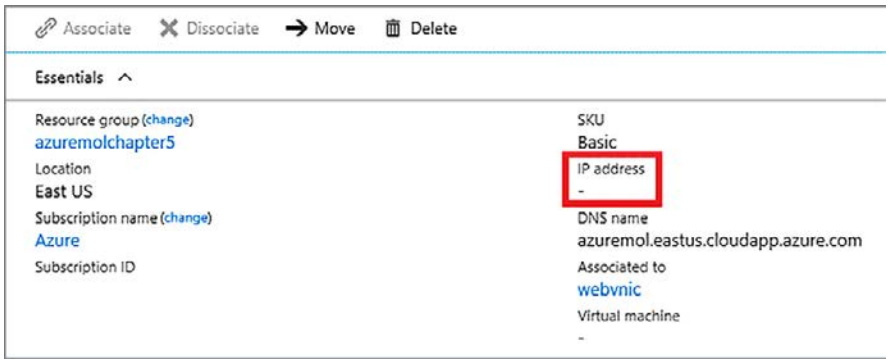


Figura 5.2 O endereço IP público agora está associado a uma interface de rede. Com uma atribuição dinâmica, nenhum endereço IP público é mostrado até que uma VM seja criada e ligada.

5.2 Proteger e controlar o tráfego com grupos de segurança de rede

Hora do quiz: você deve conectar uma VM à Internet sem um firewall para controlar e restringir o fluxo de tráfego? Se você respondeu "Claro, por que não?", talvez deva passar o resto do almoço lendo um pouco sobre a segurança da rede na Internet.

Espero que sua resposta tenha sido um retumbante "*Não!*". Infelizmente, existe uma grande chance de sua VM sofrer um ciberataque automatizado logo após ser ligada. Você deve sempre seguir as práticas recomendadas para manter o sistema operacional e a aplicação atualizados, mas você não quer mesmo que o tráfego de rede atinja sua VM se não for necessário. Um computador macOS ou Windows normal tem um firewall de software incorporado e cada rede (competente) na infraestrutura local que eu vi tem um firewall de rede entre a Internet e a rede interna. No Azure, as regras de firewall e tráfego são fornecidas por grupos de segurança de rede.

5.2.1 Criar um grupo de segurança de rede

No Azure, um NSG aplica logicamente um conjunto de regras a recursos de rede. Essas regras definem o tráfego que pode entrar e sair da VM. Você define quais portas, protocolos e endereços IP são permitidos e em que direção. Esses grupos de regras podem ser aplicados a uma única interface de rede ou a uma sub-rede de rede inteira. Essa flexibilidade permite que você controle finamente como e quando as regras são aplicadas para atender às necessidades de segurança da sua aplicação.

A Figura 5.3 mostra o fluxo lógico de um pacote de rede de entrada à medida que passa por um NSG. O mesmo processo se aplicaria a pacotes de saída. O host do Azure não diferencia o tráfego da Internet e o tráfego de outro lugar dentro do ambiente do Azure, como outra sub-rede ou rede virtual. Qualquer pacote de rede de entrada tem as regras de NSG de entrada aplicadas e qualquer pacote de rede de saída tem as regras de NSG de saída aplicadas.

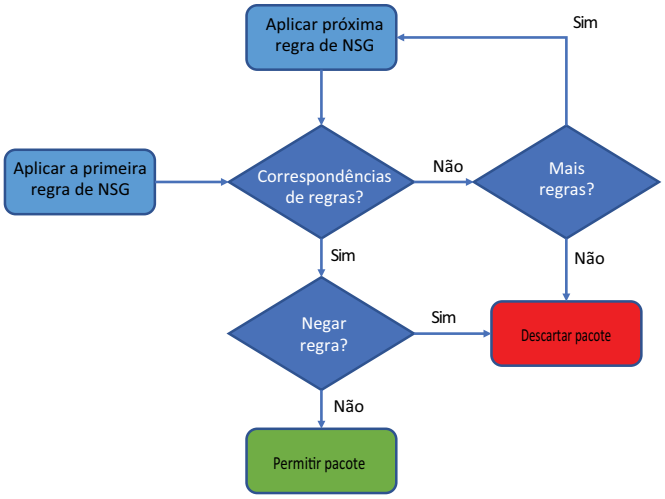


Figura 5.3 Os pacotes de entrada são examinados e cada regra de NSG é aplicada em ordem de prioridade. Se uma correspondência de regra Allow (Permitir) ou Deny (Negar) for feita, o pacote será encaminhado para a VM ou descartado.

Veja o que acontece com cada pacote de rede:

1. A primeira regra de NSG é aplicada.
2. Se a regra não corresponder ao pacote, a próxima regra será carregada até que não haja mais regras. Então, a regra padrão para descartar o pacote é aplicada.
3. Se uma regra corresponder, verifique se a ação é negar o pacote. Em caso afirmativo, o pacote é descartado.
4. Caso contrário, se a regra for permitir o pacote, o pacote será passado para a VM.

Em seguida, você criará um NSG para que esses conceitos comecem a fazer sentido.

Experimente agora

Para criar um grupo de segurança de rede, conclua as etapas a seguir:

1. No portal do Azure, selecione Create a Resource (Criar um Recurso) no canto superior esquerdo do painel.
2. Procure e selecione Network Security Group (Grupo de Segurança de Rede), e então selecione Create (Criar).
3. Insira um nome, como `webnsg`, e escolha usar o grupo de recursos existente.

Pronto! A maior parte da configuração de um NSG acontece quando você cria as regras de filtragem. A seção 5.2.2 mostra como você faz isso e coloca seu NSG para funcionar.

5.2.2 Associar um grupo de segurança de rede a uma sub-rede

O NSG não faz muito para proteger suas VMs sem regras. Você também precisa associá-lo a uma sub-rede, da mesma forma que associou seu endereço IP público a uma interface de rede anteriormente. Associe seu NSG a uma sub-rede primeiro.

Experimente agora

Para associar sua sub-rede de rede virtual ao grupo de segurança de rede, conclua as etapas a seguir:

1. Navegue e selecione Resource Group (Grupo de Recursos) na barra de navegação no lado esquerdo do portal do Azure. Depois, escolha o grupo de recursos em que você criou os recursos de rede, como azuremolchapter5.
2. Selecione seu NSG, como webnsg.
3. No lado esquerdo, em Settings options (Opções de configuração), você encontra as opções Network Interfaces (Interfaces de Rede) e Subnets (Sub-redes).Escolha Subnets (Sub-redes).
4. Selecione o botão Associar. Selecione a rede e a sub-rede virtual de rede que você criou anteriormente. Depois, selecione OK para associar seu NSG à sub-rede.

   A flexibilidade dos NSGs significa que você pode associar várias sub-redes, em várias redes virtuais, a um único NSG. O mapeamento é de um para muitos, o que permite definir regras de segurança de rede principais que se aplicam a uma ampla variedade de recursos e aplicações.

   Agora você pode ver como é seu NSG e quais regras padrão são aplicadas.
5. No lado esquerdo do NSG, selecione Inbound Security Rules (Regras de Segurança de Entrada). Nenhuma regra de segurança é listada, pelo menos nenhuma que você criou.
6. Selecione Default Rules (Regras Padrão) para ver o que a plataforma do Azure cria para você, como mostrado na Figura .5.4

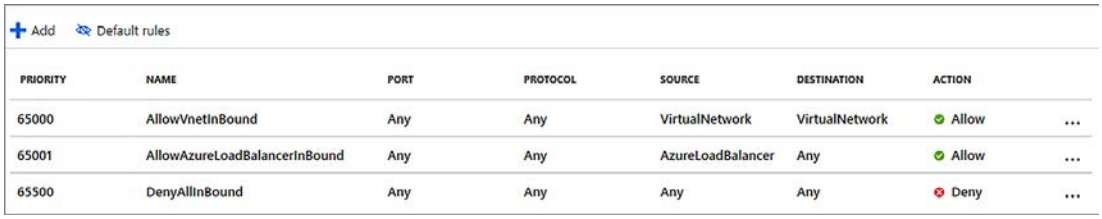


Figura 5.4 Regras de segurança padrão são criadas que permitem o tráfego de rede virtual interna ou balanceador de carga, mas negam todo o outro tráfego.

Três regras padrão foram criadas para você. É importante entender essas regras:

- *AllowVnetInBound*: permite qualquer tráfego que seja interno à rede virtual. Se você tiver várias sub-redes em sua rede virtual, o tráfego não será filtrado por padrão e será permitido.
- *AllowAzureLoadBalancerInBound*: permite que qualquer tráfego de um balanceador de carga do Azure atinja sua VM. Se você colocar um balanceador de carga entre suas VMs e a Internet, essa regra garantirá que o tráfego do balanceador de carga atinja suas VMs, como para monitorar uma pulsação.
- *DenyAllInBound*: a regra final que é aplicada. Descarta os pacotes de entrada que determinam essa distância. Se não existirem regras Allow (Permitir) anteriores, por padrão, essa regra descartará todo o tráfego. Você só precisa permitir qualquer tráfego específico desejado. O restante é descartado.

A prioridade de uma regra de NSG é importante. Se uma regra Allow (Permitir) ou Deny (Negar) for aplicada, não serão aplicadas regras adicionais. As regras são aplicadas em ordem de prioridade numérica ascendente; uma regra com prioridade 100 é aplicada antes de uma regra com prioridade 200, por exemplo.

Como com discussões anteriores sobre a governança de recursos do Azure, essas regras de NSG podem já ser criadas e aplicadas a uma determinada sub-rede. Você cria suas VMs e executa suas aplicações, e outra pessoa gerencia os NSGs.

É importante entender como o tráfego flui caso ocorra algum problema. Algumas ferramentas no Azure podem ajudá-lo a determinar por que o tráfego pode não chegar à sua aplicação quando deveria.

5.2.3 Criar regras de filtragem de grupo de segurança de rede

Agora que você tem o NSG associado à sub-rede da rede e analisamos as regras padrão, vamos criar uma regra de NSG básica que permita o tráfego HTTP.

Experimente agora

Para criar suas próprias regras com o grupo de segurança de rede, conclua as etapas a seguir:

1. Para criar uma regra de NSG na janela anterior do portal do Azure, selecione Adicionar na seção Regras de segurança de entrada.
2. Você tem duas opções para criar regras: Básico e Avançado. Para criar rapidamente regras pré-criadas, selecione Basic (Básico) na parte superior da janela.
3. Escolha HTTP no menu suspenso Service (Serviço). Muitos serviços padrão são fornecidos, como SSH, RDP e MySQL. Quando você seleciona um serviço, o intervalo de porta apropriado é aplicado, neste caso, porta 80. A ação padrão em regras básicas permite o tráfego.

4. Um valor Priority (Prioridade) é atribuído a cada regra. Quanto menor o número, maior a prioridade. Aceite a prioridade baixa padrão, como 100.
5. Aceite o nome padrão ou atribua seu próprio nome e selecione OK.

5.3 Criar um aplicativo Web de exemplo com tráfego seguro

Até agora, você criou uma rede virtual e uma sub-rede. Em seguida, criou uma interface de rede e associou um endereço IP público e um rótulo de nome DNS. Você criou um NSG e o aplicou à sub-rede inteira, além de uma regra de NSG para permitir o tráfego HTTP. Você está perdendo uma coisa: a VM.

5.3.1 Criar conexões de rede de acesso remoto

Em produção, você não deve abrir o acesso remoto, como SSH ou RDP, a VMs que executam suas aplicações. Você normalmente tem uma VM de jump-box separada à qual se conecta a partir da Internet e, em seguida, acessa VMs adicionais pela conexão interna. Até agora, você criou todos os recursos de rede virtual no portal do Azure. Vamos alternar para a CLI do Azure para ver a rapidez com que você pode criar esses recursos a partir da linha de comando.

Experimente agora

Você criou o primeiro NSG foi criado no portal do Azure. Para criar outro NSG com o Azure CLI, conclua as etapas a seguir:

1. Selecione o ícone do Cloud Shell na parte superior do painel do portal do Azure. Certifique-se de que o shell Bash seja aberto, não o PowerShell.
2. Crie um NSG adicional no grupo de recursos existente. Como nos capítulos anteriores, as barras invertidas (\) nos exemplos de comando a seguir são para ajudar com quebras de linha. Você não precisa digitá-las se não quiser. Atribua um nome, como remotensg:

```
az network nsg create \
  --resource-group azuremolchapter5 \
  --name remotensg
```

3. Crie uma regra de NSG no novo NSG que permita a porta 22. Forneça o grupo de recursos e o NSG que você criou na etapa anterior, juntamente com um nome, como allowssh:

```
az network nsg rule create \
  --resource-group azuremolchapter5 \
  --nsg-name remotensg \
  --name allowssh \
  --protocol tcp \
  --priority 100 \
  --destination-port-range 22 \
  --access allow
```

4. Crie uma sub-rede de rede para sua VM remota. Forneça um nome de sub-rede, como remotesubnet, juntamente com um prefixo de endereço dentro do intervalo da rede virtual, como 10.0.2.0/24. Você também anexa o NSG que criou na etapa anterior à sub-rede, como remotensg:

```
az network vnet subnet create \
  --resource-group azuremolchapter5 \
  --vnet-name vnetmol \
  --name remotesubnet \
  --address-prefix 10.0.2.0/24 \
  --network-security-group remotensg
```

Três comandos: isso é tudo para criar uma sub-rede, criar um NSG e criar uma regra. Você pode começar a ver o poder da CLI do Azure? O Azure PowerShell é igualmente poderoso, portanto, não pense que você deve criar todos os recursos no portal do Azure. Ao avançar no livro, use a CLI do Azure em vez do portal na maioria dos casos.

5.3.2 Criar VMs

Com todos os componentes de rede preparados, crie duas VMS. Uma VM é criada na sub-rede que permite o tráfego HTTP para que você possa instalar um servidor Web. A outra VM é criada na sub-rede que permite SSH para que você tenha uma jump box para proteger ainda mais seu ambiente de aplicação e começar a replicar uma implantação de produção. A Figura 5.5 lembra o que você está criando.

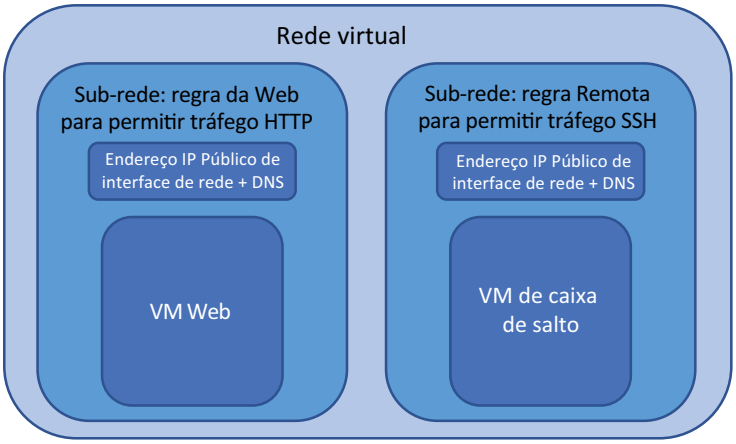


Figura 5.5 Você está reunindo duas sub-redes, NSGs, regras, interfaces de rede e VMs. Esse exemplo é semelhante a uma implantação pronta para produção em que uma VM executa o servidor Web e está aberta ao tráfego público, e outra VM em uma sub-rede separada é usada para conexões remotas com o restante do ambiente de aplicação.

Quando você cria uma VM, pode fornecer a interface de rede virtual que criou anteriormente. Se você não especificou esse recurso de rede, a CLI do Azure criará uma rede virtual, uma sub-rede e uma NIC usando padrões integrados. Isso é ótimo para criar uma VM rapidamente, mas você deseja seguir o princípio dos recursos de rede de longa duração que outra equipe pode gerenciar e em que você criará suas VMs.

Experimente agora

Para usar a CLI do Azure para criar sua VM de jump box e servidor Web, conclua as etapas a seguir:

1. Crie a primeira VM para seu servidor Web e atribua um nome, como webvm. Anexe a interface de rede, como webvnic, e insira uma imagem, como UbuntuLTS. Forneça um nome de usuário, como azuremol. A etapa final, `--generate-ssh-keys`, adiciona à VM as chaves SSH que você criou no capítulo 2:

```
az vm create \
  --resource-group azuremolchapter5 \
  --name webvm \
  --nics webvnic \
  --image UbuntuLTS \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys
```

2. Crie a segunda VM para a jump box. Esse exemplo mostra como você pode usar uma sub-rede existente e o NSG e permitir que a CLI do Azure crie a interface de rede e faça as conexões apropriadas. Crie um endereço IP público, como remotepublicip, como parte desse comando:

```
az vm create \
  --resource-group azuremolchapter5 \
  --name remotevm \
  --vnet-name vnetmol \
  --subnet remotesubnet \
  --nsg remotensg \
  --public-ip-address remotepublicip \
  --image UbuntuLTS \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys
```

A saída de ambos os comandos mostra um endereço IP público. Anote esses endereços IP. No próximo exercício, se você tentar usar o SSH em sua primeira VM para o servidor Web, haverá falha. Por quê? Você pode usar o SSH para a VM remota porque criou uma regra de NSG para permitir apenas o tráfego HTTP para a VM da Web.

5.3.3 Usar o agente SSH para se conectar às suas VMs

Eu preciso introduzir um pouco de magia com SSH que permite usar sua jump box corretamente e se conectar à VM da Web pela rede virtual do Azure: é o chamado *agente SSH*. Esse agente só se aplica a VMs Linux, portanto, se você trabalha principalmente com VMs do Windows e conexões Remote Desktop Protocol, não se preocupe se a conversa sobre SSH é nova. Você pode criar conexões RDP para VMs do Windows a partir de sua jump box com as credenciais remotas locais, ou com credenciais de domínio se você configurar o servidor apropriadamente.

Um agente SSH pode armazenar suas chaves SSH e encaminhá-las conforme necessário. De volta ao capítulo 2, quando você criou um par de chaves públicas SSH, falei sobre a chave pública e privada. A chave privada é algo que permanece no seu computador. Somente a chave pública é copiada para as VMs remotas. Embora a chave pública tenha sido adicionada às duas VMs que você criou, não é possível simplesmente usar SSH para sua jump box e, em seguida, para a VM da Web. Por quê? Essa jump box não tem uma cópia da sua chave privada. Quando você tenta fazer a conexão SSH da jump box, ela não tem nenhuma chave privada para emparelhar com a chave pública na VM da Web para autenticação.

A chave privada é algo para proteger, então você não deve seguir o caminho mais fácil e copiar a chave privada para a jump box. Quaisquer outros usuários que acessarem a jump box poderão obter uma cópia de sua chave privada e, em seguida, representá-la em qualquer lugar em que a chave seja usada. É aqui que o agente SSH entra em cena.

Se você executar o agente SSH na sua sessão do Cloud Shell, poderá adicionar suas chaves SSH a ele. Para criar sua conexão SSH com a jump box, especifique o uso desse agente para fazer o túnel da sessão. Essa técnica permite que você passe pela chave privada para uso a partir da jump box, sem nunca copiar a chave privada. Quando você usa SSH da jump box para a VM da Web, o agente SSH encapsula sua chave privada pela jump box e permite que você seja autenticado.

Experimente agora

Para usar SSH com sua VM de jump box, conclua as etapas a seguir:

1. No Cloud Shell, inicie o agente SSH da seguinte forma:

   ```
   eval $(ssh-agent)
   ```

2. Adicione a chave SSH que você criou no capítulo 2 para o agente da seguinte maneira:

   ```
   ssh-add
   ```

3. Use SSH para a VM de jump box. Especifique o uso do agente SSH com o parâmetro `-A`. Insira seu próprio endereço IP público que foi mostrado na saída quando você criou a VM de jump box:

   ```
   ssh -A azuremol@<publicIpAddress>
   ```

4. Esta é a primeira vez que você criou uma conexão SSH para a VM de jump box, portanto, aceite o prompt para se conectar com as chaves SSH.
5. Você se lembra de como criou uma atribuição de endereço IP privado estático para a VM da Web na seção 5.1.2? Esse endereço estático facilita muito o uso de SSH. Use SSH para a VM da Web da seguinte forma:

   ```
   ssh 10.0.1.4
   ```

6 Aceite o prompt para continuar a conexão SSH. O agente SSH encapsula a chave SSH privada pela jump box e permite que você se conecte com sucesso à VM da Web. E agora? Bem, você tem um laboratório para ver esse trabalho!

5.4 Laboratório: instalar e testar o servidor Web LAMP

Você já fez o trabalho duro durante todo o capítulo. Este laboratório rápido reforça como instalar um servidor Web e permite que você veja a regra de NSG em sua VM em ação:

1 *Instale um servidor Web Linux básico.* Pense de volta ao capítulo 2 quando você criou uma conexão SSH à VM e, em seguida, instalou o pacote de servidor Web LAMP com `apt`. From conexão SSH à VM da Web criada na seção 5.3.2, instale e configure a pilha da Web LAMP padrão.

2 *Navegue até o site padrão.* Quando a pilha da Web LAMP estiver instalada, abra o navegador da Web para o rótulo de nome DNS digitado quando você criou um endereço IP público na seção 5.1.3. No exemplo, foi azuremol.eastus.cloud-app.azure.com. Você também pode usar o endereço IP público que foi emitido quando criou a VM da Web. Lembre-se: esse endereço IP público é diferente da VM de jump-box VM em que usou SSH.

Parte 2

Alta disponibilidade e escala

OK, vamos começar a nos divertir! Agora que você compreende os principais recursos do Azure, pode mergulhar em áreas como redundância, balanceamento de carga e distribuição geográfica de aplicações. É aqui que as coisas ficam empolgantes, e os tópicos que você aprende devem começar a mostrar soluções e práticas recomendadas que podem ser usadas em implantações do mundo real. O Azure tem alguns recursos incríveis para replicar dados globalmente, distribuir o tráfego do cliente para a instância mais próxima da sua aplicação e escalar automaticamente com base na demanda. Esses recursos são o poder da computação na nuvem e onde você realmente agrega valor para o seu trabalho.

6 Azure Resource Manager

Na maioria dos dias, você deseja gastar o menor tempo pensando sobre como implantar um ambiente de aplicação e continuar com a implantação real. Em muitos ambientes de TI, há um movimento para equipes de desenvolvimento e operações que colaboram e trabalham em estreita colaboração, com o chavão de *DevOps* presente em muitas conferências e em blogs.

Não há nada inerentemente novo ou inovador sobre a cultura de DevOps, mas equipes diferentes geralmente não trabalharam juntas como deveriam. Ferramentas modernas têm estimulado o movimento de DevOps, com soluções de integração contínua/entrega contínua (CI/CD) que podem automatizar toda a implantação de ambientes de aplicações com base em uma única verificação de código por um desenvolvedor. A equipe de operações geralmente é aquela que cria e mantém esses pipelines de CI/CD, o que permite testes e implantações muito mais rápidos de atualizações de aplicações para desenvolvedores.

O modelo de implantação do Azure Resource Manager é fundamental para criar e executar recursos, mesmo que você provavelmente não tenha percebido ainda. O Resource Manager é uma abordagem para criação e implantação de recursos, assim como os processos de automação e modelos que impulsionam essas implantações. Neste capítulo, você aprenderá como usar recursos do Resource Manager, como controles de acesso e bloqueios, implantações de modelo consistentes e implementações de várias camadas automatizadas.

6.1 A abordagem do Azure Resource Manager

Ao criar uma VM ou um aplicativo Web em capítulos anteriores, você criou um grupo de recursos pela primeira vez como a construção principal para conter todos os seus recursos. Um grupo de recursos é essencial para todos os recursos: uma VM, um aplicativo Web, uma rede virtual ou uma tabela de armazenamento não pode existir fora de um grupo de recursos. Porém, o grupo de recursos é mais do que apenas um lugar para organizar muito mais seus recursos. Esta seção examina o que é o modelo subjacente do Azure Resource Manager e mostra por que ele é importante ao criar e executar aplicações.

6.1.1 Projetar em torno do ciclo de vida da aplicação

Preferencialmente, você não vai criar uma aplicação, nem nunca mantê-la. Você geralmente tem atualizações para desenvolver e implantar, novos pacotes a serem instalados, novas VMs para adicionar e slots de implantação de aplicativos Web adicionais para criar. Talvez seja necessário fazer alterações nas configurações de rede virtual e endereços IP. Mencionei em capítulos anteriores sobre as redes virtuais no Azure poderem ser gerenciadas por uma equipe diferente. Você precisa começar a pensar sobre como executa em uma escala grande e global, e em termos de ciclo de vida e gerenciamento de aplicações.

Você tem algumas abordagens principais para agrupar recursos no Azure:

- *Todos os recursos para uma determinada aplicação no mesmo grupo de recursos* — Como mostrado na Figura 6.1, essa abordagem funciona bem para aplicações menores ou ambientes de desenvolvimento e teste. Se você não precisar compartilhar grandes espaços de rede e puder gerenciar individualmente o armazenamento, poderá criar todos os recursos em um lugar e, em seguida, gerenciar atualizações e alterações de configuração em uma operação.

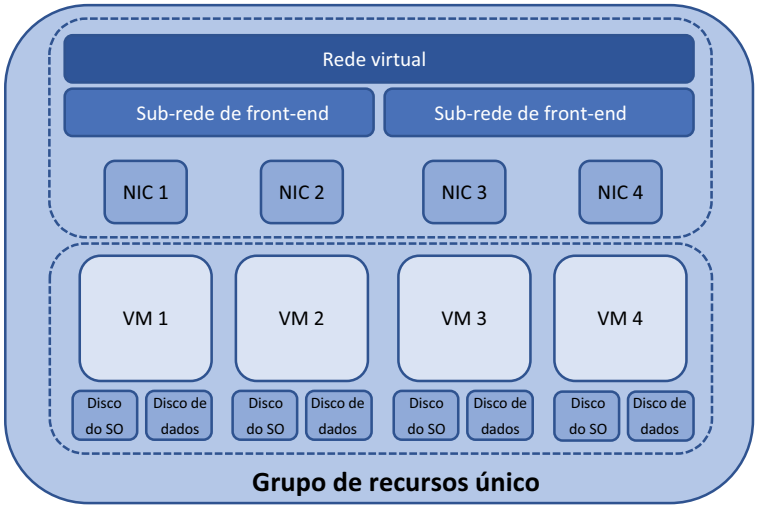


Figura 6.1 Uma maneira de criar uma aplicação no Azure é criar todos os recursos relacionados a essa implantação de aplicação no mesmo grupo de recursos e gerenciá-los como uma entidade.

- *Recursos semelhantes agrupados por função no mesmo grupo de recursos* — Como mostrado na Figura 6.2, essa abordagem geralmente é mais comum em aplicações e ambientes maiores. Sua aplicação pode existir em um grupo de recursos com apenas as VMs e componentes de aplicação de suporte. Recursos de rede virtual e endereços IP podem existir em um grupo de recursos diferentes, protegidos e gerenciados por um grupo diferente de engenheiros.

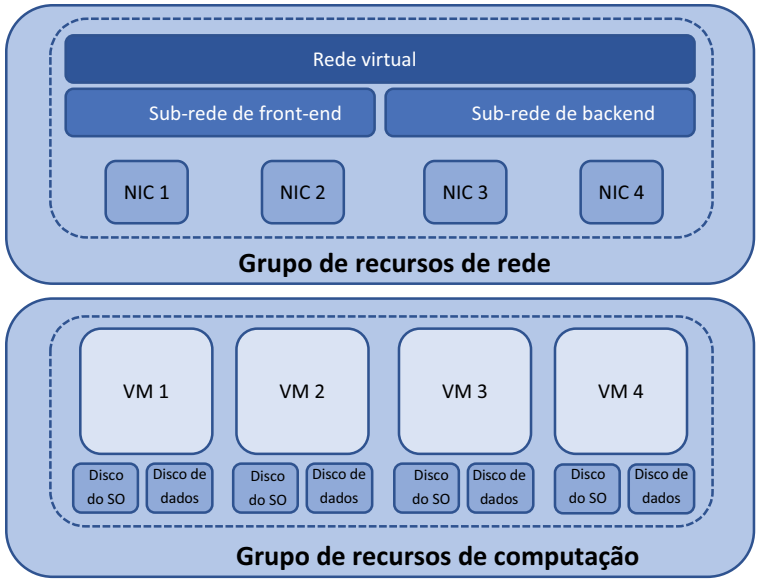


Figura 6.2 Uma abordagem alternativa é criar e agrupar recursos com base em sua função. Um exemplo comum é que todos os recursos de rede principais estão em um grupo de recursos separado dos recursos de computação da aplicação principal. As VMs no grupo de recursos de computação podem acessar os recursos de rede no grupo separado, mas os dois conjuntos de recursos podem ser gerenciados e protegidos de forma independente.

Por que há diferentes abordagens? A resposta não é simplesmente por segurança do trabalho e pelos silos adoráveis em que algumas equipes gostam de trabalhar. É sobre como você precisa gerenciar os recursos subjacentes. Em ambientes menores e aplicações onde todos os recursos existem no mesmo grupo de recursos, você é responsável por tudo nesse ambiente. Essa abordagem também é adequada para ambientes de desenvolvimento e teste em que tudo é empacotado junto. As alterações feitas na rede virtual só afetam sua aplicação e grupo de recursos.

A realidade é que as redes não mudam com frequência. Os intervalos de endereços costumam ser bem definidos e planejados para que possam coexistir no Azure e em locais de escritório ao redor do mundo. Logicamente, em geral, faz sentido colocar os componentes de rede em seu próprio grupo de recursos. A rede é gerenciada separadamente da aplicação. O armazenamento pode ser gerenciado e atualizado separadamente da mesma forma. Não há nada inerentemente errado com dividir recursos dessa maneira, contanto que a equipe de TI não fique presa em uma mentalidade de silo, resultando em uma falta de cooperação.

Para suas aplicações, a divisão de recursos também pode ser um benefício porque você fica livre para fazer as alterações e atualizações desejadas. Precisamente, como não tem os componentes de rede em seu grupo de recursos, você não precisa se preocupar com elas quando faz atualizações.

6.1.2 Proteger e controlar recursos

Cada recurso pode ter diferentes permissões de segurança aplicadas a ele. Essas políticas definem quem pode fazer o que. Pense nisso: você quer que um estagiário reinicie seu aplicativo Web ou exclua os discos de dados da VM? E você acha que seus bons amigos na equipe de rede querem que você crie uma nova sub-rede de rede virtual? Provavelmente não. No Azure, há quatro funções principais que você pode atribuir a recursos, bem como permissões de arquivo:

- *Proprietário* — Controle total, basicamente um administrador
- *Colaborador* — Gerenciamento completo do recurso, exceto fazer alterações nas atribuições de segurança e função
- *Leitor* — Capacidade de exibir todas as informações sobre o recurso, mas não fazer alterações
- *Administrador de acesso do usuário* — Capacidade de atribuir ou remover acesso a recursos

O RBAC (controle de acesso baseado em função) é um dos recursos principais do Azure que integra-se automaticamente às contas de usuário em suas assinaturas. Pense nas permissões de arquivo em seu computador normal. As permissões de arquivo básicas são leitura, gravação e execução. Quando essas permissões são combinadas, você pode criar diferentes conjuntos de permissões para cada usuário ou grupo no seu computador. À medida que você trabalha com compartilhamentos de arquivos de rede, as permissões são ferramentas comuns para controlar o acesso. O RBAC no Azure funciona da mesma maneira para controlar o acesso a recursos, assim como permissões de arquivo no seu computador local ou compartilhamento de rede (Figura 6.3).

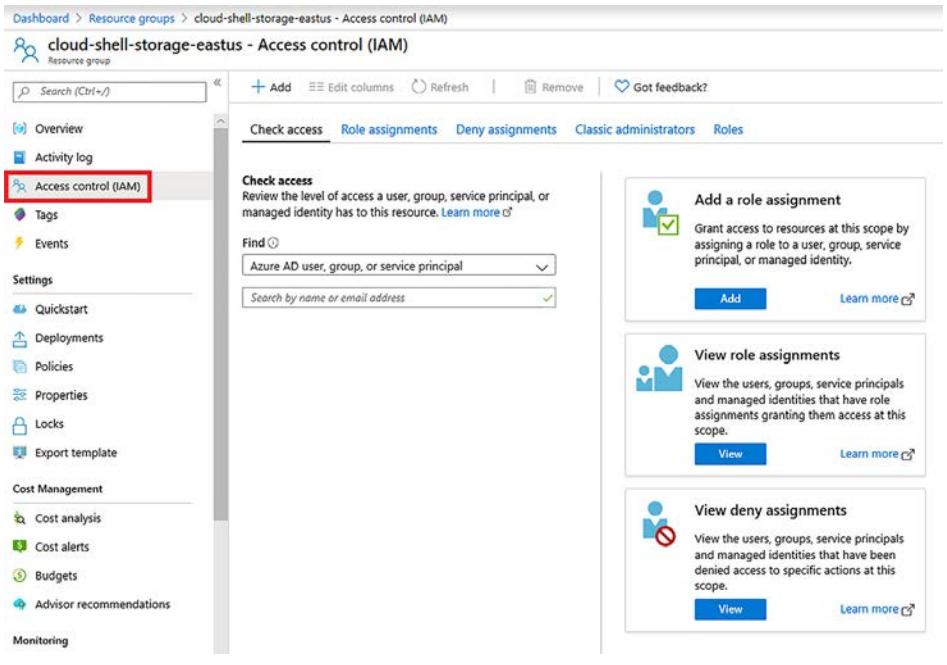


Figura 6.3 O controle de acesso para cada recurso do Azure lista as atribuições atuais. Você pode adicionar atribuições ou selecionar Roles (Funções) para ver informações sobre quais conjuntos de permissões estão disponíveis.

Experimente agora

Abra o portal do Azure em um navegador da Web e, em seguida, selecione qualquer recurso que você tenha, como cloud-shell-storage. Escolha o botão Access Control (Controle de Acesso) (IAM), como mostrado na Figura 6.3. Revise as atribuições de função atuais. Veja como adicionar uma atribuição de função e explore todas as atribuições de função disponíveis. O ícone de informações para cada função mostra quais permissões são atribuídas.

Ao explorar as funções disponíveis, você pode observar várias funções específicas de recursos, incluindo as seguintes:

- Colaborador de máquina virtual
- Colaborador de site
- Colaborador de rede

Adivinhe o que essas funções significam? Elas assumem a função de colaborador de plataforma principal e a aplicam a um tipo de recurso específico. O caso de uso aqui remonta a esse conceito de como você gerencia recursos semelhantes. Você pode ser atribuído à função de colaborador de máquina virtual ou colaborador de site. Em seguida, as VMs ou os Aplicativos Web criados nesse grupo de recursos estariam disponíveis para você gerenciar. No entanto, você não pode gerenciar recursos de rede, que podem estar em um grupo de recursos totalmente diferente.

6.1.3 Proteger recursos com bloqueios

A abordagem baseada em permissões do RBAC é ótima para limitar quem pode acessar o que. Porém, erros ainda podem acontecer. Há uma razão pela qual você normalmente não faz login em um servidor como um usuário com permissões administrativas ou raiz. Um pressionamento de tecla ou clique do mouse errado, e você pode excluir recursos por engano. Mesmo se você tiver backups (você tem backups, certo? E você os testa regularmente?), é um processo demorado que pode significar perda de produtividade ou receita para o negócio. No capítulo 13, você aprenderá mais sobre as maneiras como os serviços de backup, recuperação e replicação do Azure protegem seus dados.

Outro recurso incorporado ao modelo do Resource Manager é bloqueios de recursos. Cada recurso pode ter um bloqueio aplicado que limita o acesso somente leitura ou evita operações de exclusão. O bloqueio de exclusão é particularmente útil, pois pode ser muito fácil excluir o grupo de recursos errado. Quando você iniciar uma operação de exclusão, não há como voltar atrás ou cancelar a operação depois que a plataforma do Azure aceitou sua solicitação.

Para workloads de produção, sugiro que você implemente bloqueios em seus recursos principais para evitar exclusões. Esses bloqueios são somente nos níveis de recurso e plataforma do Azure, não para os dados dentro de seus recursos. Por exemplo, você pode excluir arquivos dentro de uma VM ou descartar uma tabela. Os bloqueios de recursos do Azure seriam aplicados somente se você tentasse excluir toda a VM ou o banco de dados SQL do Azure. Na primeira vez que um bloqueio for ativado e impedir que o grupo de recursos incorreto seja excluído, você vai me agradecer.

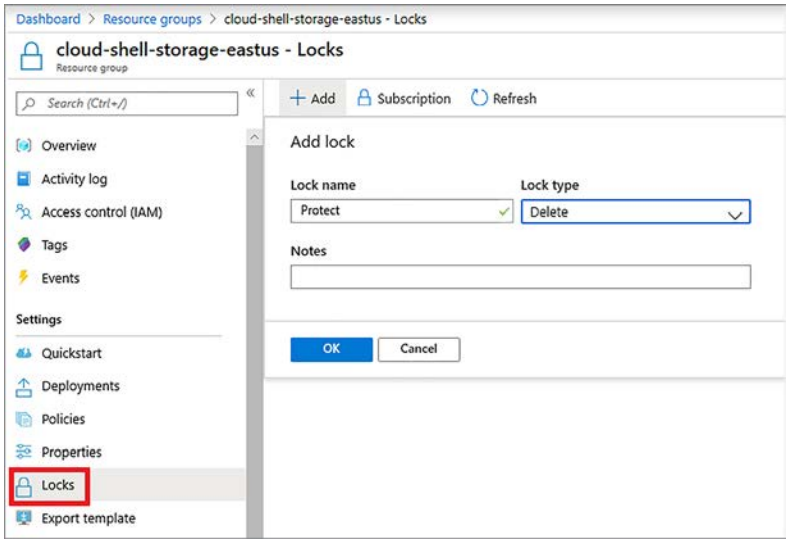


Figura 6.4 Crie um bloqueio de recurso no portal do Azure.

Experimente agora

Para ver os bloqueios de recursos do Azure em ação, como mostrado na Figura 6.4, conclua as etapas a seguir:

1. Abra o portal do Azure em um navegador da Web e selecione qualquer grupo de recursos que você tenha, como cloud-shell-storage.
2. Escolha Bloqueios no lado esquerdo do portal.
3. Insira um nome de bloqueio, como Proteger, selecione Excluir no menu suspenso Tipo de bloqueio e escolha OK. Seu novo bloqueio aparece na lista.
4. Selecione Visão geral para o grupo de recursos e, em seguida, tente excluir o grupo de recursos. Você precisa digitar o nome do grupo de recursos para confirmar que deseja excluí-lo (o que também é um bom prompt mental para se certificar de que tenha o recurso certo para excluir!).
5. Ao escolher o botão Excluir, revise a mensagem de erro exibida para ver como o bloqueio impediu o Azure de excluir o recurso.

6.1.4 *Gerenciar e agrupar recursos com tags*

Um recurso final no modelo do Azure Resource Manager que desejo apresentar são as *tags*. Não há nada de novo ou especial sobre como você marca recursos no Azure, mas esse conceito de gerenciamento geralmente é negligenciado. Você pode aplicar tags a um recurso no Azure que descreva propriedades como a aplicação da qual ele faz parte, o departamento responsável por ele ou se é um recurso de desenvolvimento ou produção.

Você pode direcionar recursos com base em tags para aplicar bloqueios ou funções de RBAC ou relatar os custos e o consumo de recursos. As tags não são exclusivas de um grupo de recursos e podem ser reutilizadas em sua assinatura. Até 50 tags podem ser aplicadas a um único recurso ou grupo de recursos, portanto, você tem muita flexibilidade na forma como você marca e filtra os recursos marcados.

Experimente agora

Para ver as tags de recursos do Azure em ação, conclua as etapas a seguir:

1. Abra o portal do Azure em um navegador da Web e, em seguida, selecione qualquer recurso, como cloud-shell-storage. Embora você possa marcar um grupo de recursos em si, não escolha um grupo de recursos para este exercício.
2. Com seu recurso selecionado, escolha o botão Tags, como mostrado na Figura 6.5.
3. Insira um nome, como `workload`, e um valor, como `desenvolvimento`.
4. Selecione Salvar.
5. Abra o Cloud Shell.
6. Para filtrar recursos com base em tags, use `az resource list` com o parâmetro `--tag`. Use seu próprio nome e valor da seguinte maneira:

```
az resource list --tag workload=development
```

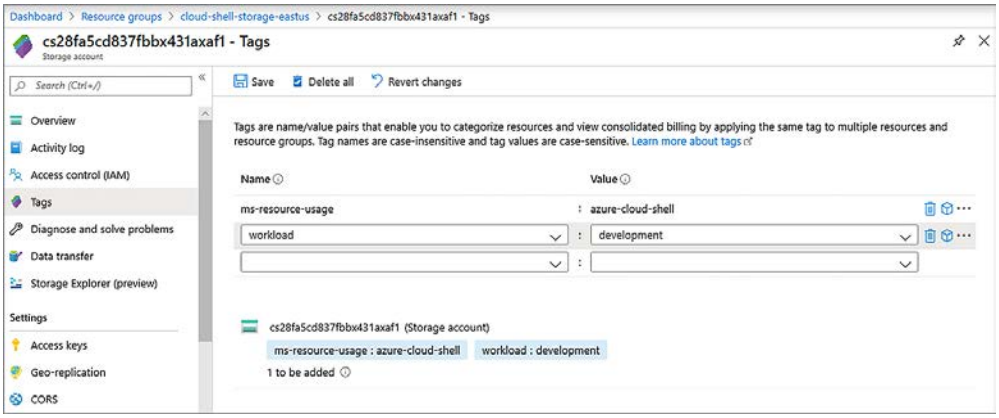


Figura 6.5 Você pode criar até 50 tags name:value para cada recurso ou grupo de recursos do Azure.

6.2 Modelos do Azure Resource Manager

Até agora, você criou um pequeno número de recursos do Azure por vez. Para fazer isso, você usou o portal do Azure ou a CLI do Azure. Embora eu não tenha mostrado a você o Azure PowerShell, mencionei-o no capítulo 1, e ele está disponível no Cloud Shell. Talvez você tenha experimentado sem mim. Tudo bem, eu não ligo. Como mencionei no capítulo 1, o Azure tem ferramentas que permitem escolher o que é mais adequado para você e para o ambiente em que trabalha.

A desvantagem de usar o portal ou os comandos da CLI ou do PowerShell é que você deve clicar em vários botões no navegador da Web ou digitar linhas de comandos para criar seu ambiente de aplicação. Você pode criar scripts para fazer tudo isso, mas deve criar lógica para lidar com a criação de vários recursos ao mesmo tempo ou a ordem em que os recursos devem ser criados.

Um script que encapsula os comandos da CLI do Azure ou do PowerShell começa a se mover na direção certa em termos de como você deve criar e implantar ambientes de aplicações, não apenas no Azure, mas em qualquer plataforma. Há um movimento em direção à infraestrutura como código (IaC), que não é nenhuma novidade se você já trabalha com TI há algum tempo. Tudo isso significa que você não confia em um ser humano para digitar comandos e seguir um conjunto de passos; em vez disso, você cria programaticamente sua infraestrutura a partir de um conjunto de instruções. As implantações manuais introduzem um elemento humano que muitas vezes pode levar a pequenas configurações incorretas e diferenças nas VMs finais, como mostrado na Figura 6.6.

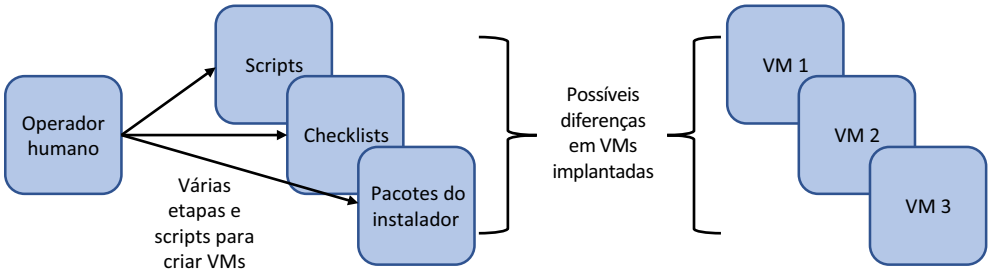


Figura 6.6 Os seres humanos cometem erros, como o erro de digitação de um comando ou pular uma etapa em uma implantação. Você pode acabar com VMs ligeiramente diferentes no final da saída. A automação é frequentemente usada para remover o operador humano da equação e, em vez disso, criar implantações consistentes e idênticas todas as vezes.

Mesmo quando você tem scripts, ainda precisa de alguém para gravá-los, fazer manutenção e mantê-los atualizados à medida que novas versões dos módulos da CLI do Azure ou do PowerShell forem lançadas. Sim, às vezes, são feitas alterações nas ferramentas para acomodar novos recursos, embora sejam raras.

6.2.1 Criar e usar modelos

Os modelos do Resource Manager podem ajudar a reduzir o erro humano e a dependência de scripts gravados manualmente. Os modelos são gravados em JavaScript Object Notation (JSON), uma abordagem padrão aberta entre plataformas que permite editá-los em um editor de texto básico. Com modelos, você pode criar implantações consistentes e reproduzíveis que minimizem erros. Outro recurso interno dos modelos é que a plataforma compreende dependências e pode criar recursos em paralelo, sempre que possível, para acelerar o tempo de implantação. Se você criar três VMs, por exemplo, não é necessário esperar que a primeira VM termine a implantação antes de criar a segunda. O Resource Manager pode criar todas as três VMs ao mesmo tempo.

Como um exemplo de dependências, se você criar uma NIC virtual, precisará conectá-la a uma sub-rede. Logicamente, a sub-rede deve existir antes de criar a NIC virtual e a sub-rede deve fazer parte de uma rede virtual, de modo que a rede deve ser criada antes da sub-rede. A Figura 6.7 mostra a cadeia de dependências em ação. Se você tentar gravar um script por conta própria, deverá planejar cuidadosamente a ordem em que os recursos são criados e, mesmo assim, deverá criar a lógica para saber quando os recursos principais estão prontos e você pode passar para os recursos dependentes.

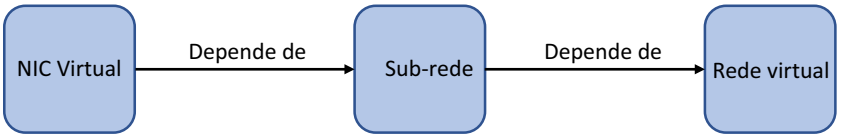


Figura 6.7 O Azure Resource Manager manipula dependências para você. A plataforma sabe a ordem em que deve criar recursos e tem consciência do estado de cada recurso sem o uso de lógica manuscrita e loops como aqueles que você deve usar em seus próprios scripts.

Quer saber algo legal? Você já usou modelos do Resource Manager, no capítulo 2 e na primeira VM que criou. À medida que você cria uma VM no portal ou na CLI do Azure, em segundo plano, um modelo é criado e implantado de forma programática. Por quê? Bem, por que reinventar a roda e passar pelo processo de construção de toda essa lógica para as implantações? Deixe o Azure Resource Manager fazer isso por você.

Veja a seguir como é uma seção de um modelo do Resource Manager. A listagem a seguir mostra a seção que cria um endereço IP público, exatamente como em exemplos anteriores quando você criou uma VM.

Listagem 6.1 Criar um endereço IP público em um modelo do Resource Manager

```
{
       "apiVersion": "2019-04-01",
       "type": "Microsoft.Network/publicIPAddresses",
       "name": "publicip",
       "location": "eastus",
       "properties": {
         "publicIPAllocationMethod": "dynamic",
         "dnsSettings": {
           "domainNameLabel": "azuremol"
    }
  }
},
```

Mesmo que JSON seja novo para você, ele é escrito em um formato (um pouco) legível por humanos. Você define um tipo de recurso, neste exemplo, `Microsoft.Network/publicIPAddresses`. Em seguida, você fornece um nome, como `publicip`, e um local, como `eastus`. Por fim, você define o método de alocação, `dinâmico` neste exemplo, e um rótulo de nome DNS, como `azuremol`. Esses parâmetros são os mesmos fornecidos quando você usou o portal do Azure ou a CLI. Se você usa o PowerShell, adivinhe? Você deverá usar os mesmos parâmetros.

A diferença com o modelo é que você não precisa inserir nenhuma informação. Está tudo incluído no código. "Ótimo", você pode pensar, "mas e se eu quiser usar nomes diferentes de cada vez?" Assim como acontece com um script, você pode atribuir nomes dinamicamente usando parâmetros e variáveis:

- *Parâmetros* são valores que são solicitados. Eles geralmente são usados para credenciais de usuário, o nome da VM e o rótulo de nome DNS.
- *Variáveis* podem ser pré-atribuídas valores, mas também são ajustadas sempre que você implanta o modelo, como o tamanho da VM ou o nome da rede virtual.

Experimente agora

Para ver um modelo completo do Resource Manager, abra um navegador da Web para o repositório GitHub em http://mng.bz/QyWv.

6.2.2 Criar múltiplos de um tipo de recurso

À medida que você criar modelos, tente pensar antes sobre como talvez seja necessário expandir suas aplicações no futuro. Você pode precisar apenas de uma única VM ao implantar sua aplicação pela primeira vez, mas conforme a demanda para a aplicação cresce, talvez seja necessário criar instâncias adicionais.

Em uma implantação de script tradicional, você cria um loop `for` ou `while` para criar vários tipos de recurso. O Resource Manager tem essa funcionalidade incorporada. Há mais de 50 tipos de funções no Resource Manager, assim como na maioria das linguagens de programação e scripts. Funções comuns do Resource Manager incluem `length`, `equals`, `or` e `trim`. Você controla o número de instâncias a serem criadas com a função `copy`.

Quando você usa a função `copy`, o Resource Manager cria o número de recursos especificado. Cada vez que o Resource Manager itera sobre a operação de criação, um valor numérico está disponível para você nomear recursos de forma sequencial. Você acessa esse valor com a função `copyIndex()`. O exemplo na Listagem 6.1 criou um único endereço IP público. O exemplo na Listagem 6.2 usa o mesmo tipo de provedor de recursos `Microsoft.Network/publicIPAddresses`, mas cria dois endereços IP públicos. Você pode usar copy para definir quantos endereços deseja criar e `copyIndex()` para nomear os endereços sequencialmente.

Listagem 6.2 Criar vários endereços IP públicos com `copy`

```
{
       "apiVersion": "2019-04-01",
       "type": "Microsoft.Network/publicIPAddresses",
       "name": "[concat('publicip', copyIndex())]",
       "copy": {
         "count": 2
       }
       "location": "eastus",
       "properties": {
         "publicIPAllocationMethod": "dynamic",
       }
},
```

Você também usa a função `concat` para combinar o nome do endereço IP público e o valor numérico de cada instância criada. Depois que esse modelo é implantado, seus dois endereços IP públicos são chamados de `publicip0` e `publicip1`. Esses nomes não são super descritivo, mas este exemplo básico mostra o conceito de como você pode usar uma convenção de numeração à medida que cria vários recursos com a função `copy`.

6.2.3 Ferramentas para criar seus próprios modelos

E confesso: embora os modelos do Resource Manager sejam organizados e entre as principais maneiras que eu sugiro para criar e implantar aplicações no Azure, você ainda precisará gravar os modelos. Algumas ferramentas diferentes simplificam essa tarefa para você, e centenas de modelos de exemplo estão disponíveis na Microsoft e em terceiros. Na verdade, uma das melhores maneiras de aprender a criar e usar modelos é examinar os modelos de início rápido que a Microsoft disponibiliza em seu repositório de exemplos em https://github.com/Azure/azure-quickstart-templates.

Se você quiser arregaçar as mangas e começar a gravar seus próprios modelos, recomendo duas ferramentas principais que eu recomendo. A primeira é o Visual Studio Code, um editor de open source gratuito para várias plataformas (https://code.visualstudio.com). Junto com algumas funcionalidades incorporadas, como o controle de origem e a integração do GitHub, extensões estão disponíveis que podem criar automaticamente as diferentes seções ou provedores, para que os recursos criem um modelo, como mostrado na Figura 6.8. Se você fizer o download e instalar o VS Code, escolha Exibir > Extensões e, em seguida, pesquise *Azure*.

Figura 6.8 Muitas extensões estão disponíveis no Visual Studio Code para aprimorar e agilizar a forma como você cria e usa modelos do Azure Resource Manager.

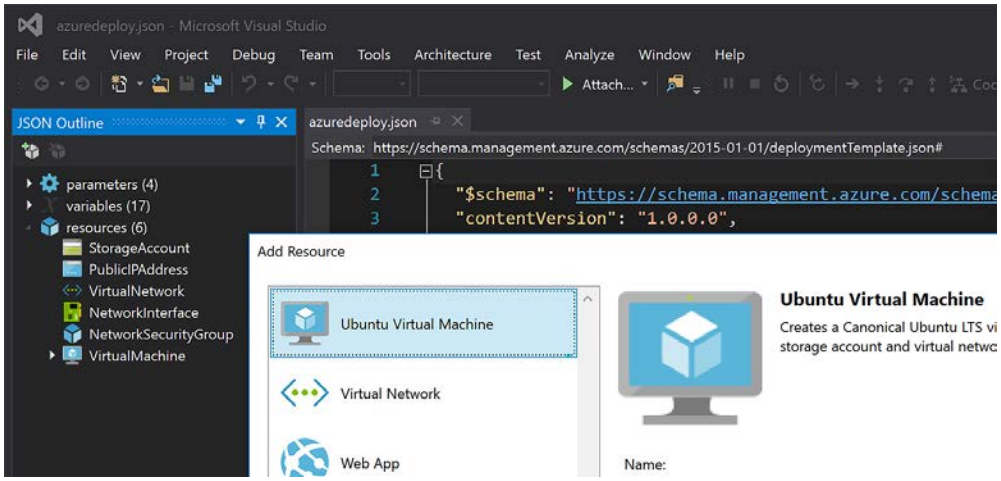


Figura 6.9 Com o Visual Studio, você pode criar graficamente modelos e explorar recursos JSON.

Uma maneira mais gráfica de criar modelos do Azure Resource Manager é usar o editor completo do Visual Studio, como mostrado na Figura 6.9. Há versões para Windows e macOS, mas você precisa de uma licença separada para usar o editor. Uma Community Edition está disponível, mas tome cuidado se você criar modelos dentro de sua empresa: normalmente, você precisa de uma versão licenciada. Consulte seus especialistas em licenças, porque o Visual Studio é destinado a desenvolvedores de aplicações.

Você pode, naturalmente, usar um editor de texto básico. Parte do motivo pelo qual os modelos do Azure Resource Manager são gravados em JSON é que ele remove a necessidade de ferramentas especiais. Há uma curva de aprendizado para trabalhar com JSON, e é por isso que eu recomendo que você explore os modelos de início rápido no repositório de exemplos do Azure. Tome cuidado com o recuo, vírgulas à direita e o uso de parênteses, colchetes e chaves.

Vida em Marte

Existem ferramentas de outros fabricantes e outras formas de usar modelos no Azure. A Hashicorp fornece muitas ferramentas de Open Source e soluções para computação na nuvem; uma delas é Terraform. Com o Terraform, você define todos os recursos que deseja criar da mesma forma que faz em um modelo nativo do Azure Resource Manager. Você também pode definir dependências e usar variáveis. A diferença é que o Terraform é tecnicamente voltado para vários provedores. As mesmas construções e abordagem de modelo podem ser usadas em Azure, Google Cloud, AWS e vSphere, por exemplo. A diferença são os provisionadores que você usa para cada recurso.

É realmente uma abordagem de "um modelo para qualquer provedor"? Não, não é. O Terraform também é uma aplicação que analisa seu modelo e, em seguida, se comunica com o provedor de nuvem relevante, como o Azure. Você não tem recursos de edição, muito menos ferramentas gráficas, para criar seu modelo. Você escolhe um editor e grava o modelo manualmente. Novamente, a melhor maneira de aprender a usar o Terraform é explorar sua documentação e modelos de exemplo.

A razão pela qual menciono este tópico está relacionada ao conceito de escolha no Azure. Se você achar que os modelos do Azure Resource Manager gravados em JSON são um pouco complicados, explore um produto como o Terraform. Porém, não desista de implantações do Resource Manager. Para alcançar essas implantações consistentes e reproduzíveis em escala, os modelos são a melhor abordagem, portanto, encontre uma boa abordagem orientada por modelo que funcione para você.

6.2.4 Armazenar e usar modelos

Então você adora a ideia de modelos do Azure Resource Manager e instalou o Visual Studio ou o código para gravar seus próprios modelos. Como armazená-los e implantá-los? No laboratório de fim de capítulo, você implanta um modelo do repositório de exemplos do Azure no GitHub. Este repositório é público, e talvez você não queira disponibilizar seus modelos de aplicação para o mundo inteiro.

Há alguns métodos comuns para armazenar os modelos do Resource Manager em particular:

- Use um repositório privado ou compartilhamento de arquivos de rede em sua organização.
- Use o Armazenamento do Azure para armazenar e proteger centralmente modelos para implantação.

Não há maneira certa ou errada de armazenar e implantar modelos. Você tem a flexibilidade para usar quaisquer processos e ferramentas já em vigor. A vantagem de usar um repositório é que você normalmente tem alguma forma de controle de versão para garantir implantações consistentes e revisar o histórico de seus modelos, se necessário. A única limitação é que, ao implantar o modelo, você precisa fornecer as credenciais apropriadas para acessar o local compartilhado. Esse processo de autenticação pode variar, como fornecer um nome de usuário ou token de acesso como parte da URL para um modelo em um repositório ou fornecer um token de assinatura de acesso compartilhado (SAS) se você usar o Armazenamento do Azure.

Repositórios públicos como o GitHub também podem ser ótimas maneiras de aprender e compartilhar. Eu sugiro que você mantenha seus modelos de produção armazenados em particular, mas se criar um modelo organizado para um ambiente de laboratório ou para experimentar alguns novos recursos, compartilhá-lo no GitHub dá um retorno para a comunidade de TI e pode ajudar outros que querem fazer as mesmas implantações que você faz. E, ao começar seus próprios modelos, verifique quais modelos já existem para você não começar do zero e reinventar a roda sempre.

6.3 Laboratório: Implantar recursos do Azure de um modelo

Toda essa teoria sobre modelos de implantação e abordagens é excelente, mas você (preferencialmente) começa a ver os benefícios e a eficiência quando usa modelos para algo real:

1. Acesse os exemplos de início rápido do Azure no GitHub (https://github.com/Azure/azure-quickstart-templates) e encontre um que lhe interesse. Um bom lugar para começar é uma simples VM Linux ou Windows.

2 Os exemplos do GitHub têm botões integrados que implantam diretamente no Azure. Quando encontrar um modelo de que goste, selecione Implantar no Azure, como mostrado na Figura 6.10 e seguir as etapas no portal. O processo é muito parecido com a criação de uma VM anteriormente no livro, mas alguns prompts são necessários para concluir os parâmetros necessários. Todos os outros recursos são criados para você e abstraídos.

3 A etapa final para implantar seu modelo é aceitar o contrato de licença mostrado e, em seguida, escolher Comprar. Você cria recursos do Azure ao implantar um modelo, comprar significa que você concorda em pagar pelos custos desses recursos do Azure.

Um dos modelos básicos, como uma simples VM do Linux ou do Windows, custa o mesmo que qualquer outra VM criada até agora. Exclua o grupo de recursos quando a implantação for concluída, assim como limparia após qualquer outro exercício.

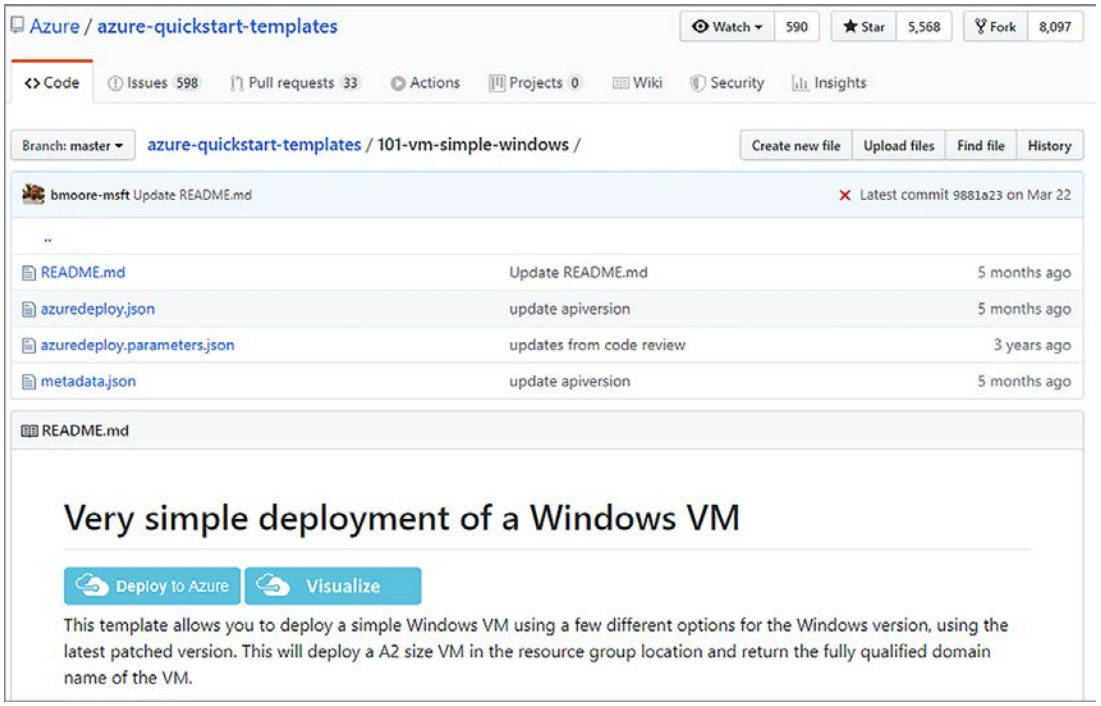


Figura 6.10 Para cada modelo do Resource Manager no repositório de exemplos do GitHub, há um botão Implantar no Azure. Se você selecionar esse botão, o portal do Azure será carregado, e o modelo também. Você deverá inserir alguns parâmetros básicos e o restante da implantação é manipulado pelo modelo.

Parâmetros em modelos

Conforme discutido na seção 6.2.1, você pode usar parâmetros e variáveis em seus modelos. Lembre-se que os parâmetros são valores solicitados e variáveis são valores dinâmicos que podem ser aplicados em todo um modelo. Os valores solicitados (parâmetros) variam de modelo para modelo. Portanto, dependendo do modelo de início rápido selecionado, você deverá inserir um ou dois valores, ou fornecer sete ou oito itens.

Ao projetar seus modelos, tente antecipar como você e outros usuários podem querer reutilizar o modelo ao implantar aplicações. Você pode fornecer um valor padrão e limitar quais valores são permitidos. No entanto, tome cuidado com esses valores padrão e permitidos, ou você pode restringir muito os usuários e forçá-los a criar seus próprios modelos. Sempre que possível, tente criar modelos principais reutilizáveis que tenham flexibilidade suficiente.

4 Quando o modelo for implantado, volte para o GitHub e examine o arquivo azure-deploy.json. Esse arquivo é o modelo do Azure Resource Manager que você usou para criar e implantar o exemplo. Veja se você consegue entender os diferentes tipos de recursos e configurações que foram aplicados. À medida que você trabalha com mais tipos de recursos e modelos do Azure, o formato JSON fica mais fácil de entender. De verdade!

7 Alta disponibilidade e redundância

Eu não posso contar o número de vezes que algo em TI falhou comigo. Eu tive um acidente com o disco rígido do laptop no dia anterior a uma conferência, uma fonte de alimentação soltando fumaça em um servidor de email e interfaces de rede falharam em um roteador central. Isso sem falar em sistema operacional, driver e atualizações de firmware. Tenho certeza de que qualquer pessoa que trabalha em TI adoraria compartilhar histórias de horror de situações com as quais tiveram de lidar, geralmente os problemas aconteceram tarde da noite ou em um momento crítico para o negócio. Mas existe fracasso bom e momento agradável?

Se você antecipar falhas em TI, aprenderá a planejar e projetar suas aplicações para acomodar problemas. Neste capítulo, você aprenderá a usar os recursos de alta disponibilidade e redundância do Azure para minimizar as interrupções causadas por atualizações e interrupções de manutenção. Este capítulo constrói a base para os próximos dois ou três capítulos à medida que você começa a passar de uma aplicação que é executada em uma única VM ou aplicativo Web para um que pode ser escalado e distribuído globalmente.

7.1 A necessidade da redundância

Para que os clientes confiem em você para sua importante pizzaria, você deve fornecer aplicações que são acessíveis sempre que necessário. A maioria dos clientes não vai procurar "horário de funcionamento" em um site, principalmente se você trabalha em um ambiente global e os clientes podem ser de todo o mundo. Quando estão com fome, eles querem comer.

A Figura 7.1 mostra um exemplo básico de uma aplicação que é executada em uma única VM. Infelizmente, essa aplicação cria um único ponto de falha. Se essa VM não estiver disponível, a aplicação não estará disponível, o que leva à insatisfação e à fome do cliente.

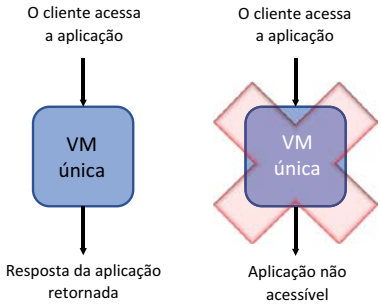


Figura 7.1 Se sua aplicação é executada em uma única VM, qualquer interrupção nessa VM faz com que a aplicação fique inacessível. Isso pode fazer com que os clientes levem seus negócios para outro lugar ou, pelo menos, não fiquem satisfeitos com o serviço fornecido.

Se você dirige, provavelmente tem um pneu de reserva no carro caso o pneu fure. Se você usar um laptop ou tablet, provavelmente ligará o dispositivo em um carregador caso a bateria acabe no meio do trabalho. Em casa ou em seu apartamento, você tem lâmpadas de reserva para quando a luz acaba? Que tal uma lanterna ou velas para quando houver queda de energia?

A maioria das pessoas gosta de ter alguma forma de redundância ou plano de backup, no dia a dia e, principalmente, em TI. Se você estiver pronto para usar um pneu ou uma lâmpada de reserva, saberá lidar com interrupções e falhas com interrupção mínima. Ao projetar e criar suas aplicações para redundância, você fornece um alto nível de disponibilidade para seus clientes que minimiza ou até mesmo oculta quaisquer interrupções que a aplicação venha a ter. Todos os datacenters do Azure são criados para alta disponibilidade. Fontes de alimentação de backup, várias conexões de rede e matrizes de armazenamento com discos sobressalentes são apenas alguns dos principais conceitos de redundância que o Azure fornece e gerencia para você. Toda a redundância que o Azure fornece talvez não seja útil se você executar sua aplicação em uma única VM. Para dar a você flexibilidade e controle sobre como tornar sua aplicação altamente disponível, dois recursos principais para workloads IaaS estão disponíveis:

- *Zona de disponibilidade* — Permite que você distribua VMs em segmentos fisicamente isolados de uma região do Azure para maximizar ainda mais a redundância da aplicação. As zonas também podem fornecer alta disponibilidade para recursos de rede, como endereços IP públicos e balanceadores de carga.
- *Conjuntos de disponibilidade* — Permite que você agrupe logicamente as VMs para distribuí-las em um único datacenter do Azure e minimize interrupções causadas por paralizações ou atualizações de manutenção.

Para a maioria das novas implantações de aplicações no Azure, sugiro que você planeje usar zonas de disponibilidade. Essa abordagem oferece flexibilidade na forma de distribuir sua aplicação e fornece redundância aos recursos de rede que são geralmente essenciais para os clientes finalmente acessarem as VMs subjacentes. Para ver como cada uma dessas abordagens funciona, vamos discuti-las em mais detalhes.

7.2 Redundância de infraestrutura com zonas de disponibilidade

As zonas de disponibilidade são datacenters fisicamente separados que operam em utilitários de núcleo independentes, como energia e conectividade de rede. Cada região do Azure que oferece suporte a zonas de disponibilidade fornece três zonas. Você cria seus recursos nessas zonas. A Figura 7.2 mostra como os recursos do Azure podem ser distribuídos entre zonas de disponibilidade.

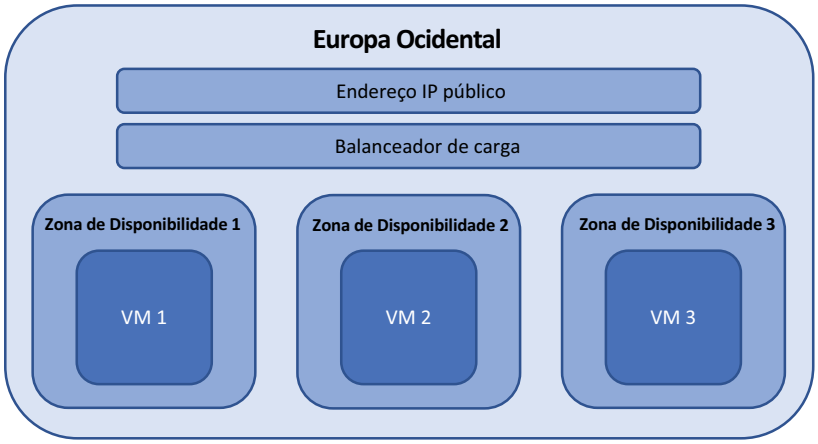


Figura 7.2 Uma região do Azure pode conter várias zonas de disponibilidade: datacenters fisicamente isolados que usam energia, rede e resfriamento independentes. Os recursos de rede virtual do Azure, como endereços IP públicos e balanceadores de carga, podem abranger todas as zonas de uma região para fornecer redundância para mais do que apenas as VMs.

Com zonas de disponibilidade, suas aplicações podem tolerar todo o datacenter do Azure ficando offline. Claro, seria preciso um grande evento para que essa situação ocorra, mas ainda é possível.

Em implantações de aplicações grandes, você pode criar mais de uma VM em cada zona de disponibilidade. Várias VMs em uma zona de disponibilidade são distribuídas automaticamente entre o hardware disponível na zona. Não há nada que você precisa configurar ou pode controlar. Mesmo que uma atualização de manutenção ou falha de equipamento dentro de uma zona afete todas as VMs executadas na zona, lembre-se de que as zonas estão fisicamente isoladas umas das outras. As VMs em outra zona continuariam sendo executadas.

Agora, se você acha que é particularmente azarado, todas as VMs em diferentes zonas poderiam passar por atualizações de manutenção ao mesmo tempo? Sim, mas isso é improvável. As zonas dentro de uma região têm ciclos de atualização preparados. As atualizações são executadas em uma zona. Quando forem concluídas, as atualizações serão executadas na próxima zona. As zonas de disponibilidade fornecem um nível alto de abstração e redundância, e você deve examinar sua aplicação em toda a implantação, não apenas onde residem as VMs em uma zona.

A inclusão dos recursos de rede virtual em zonas de disponibilidade é muito mais importante do que pode parecer a princípio. A Figura 7.3 mostra o que aconteceria se o datacenter

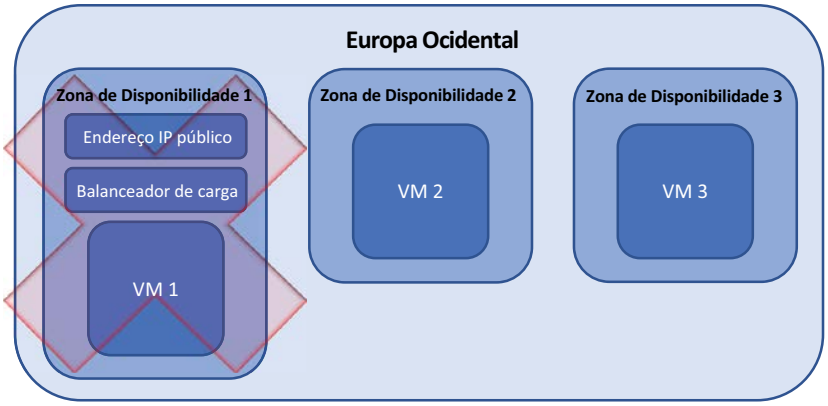


Figura 7.3 Quando os recursos de rede são anexados a um único datacenter do Azure ou zona, uma paralisação nessa instalação faz com que toda a aplicação seja inacessível para o cliente. Não importa que as outras VMs continuem a ser executadas em outras zonas. Sem a conectividade de rede para distribuir o tráfego de seus clientes, a aplicação inteira não está disponível.

ficasse indisponível para recursos de rede, como um endereço IP público e um balanceador de carga que são executados em zonas de disponibilidade.

Vou explicar mais sobre balanceadores de carga no capítulo 8, mas por enquanto, tudo o que você precisa entender é que o balanceador de carga distribui o tráfego em todas as VMs disponíveis que estão anexadas a ele. As VMs relatam seu status de integridade em intervalos definidos, e o balanceador de carga não distribui mais o tráfego para uma VM que relata que não está disponível. Com um balanceador de carga que funciona em zonas de disponibilidade, uma paralisação em um datacenter do Azure faz com que essas VMs fiquem indisponíveis e sejam retiradas da rotação do balanceador de carga.

Um endereço IP público que abrange zonas de disponibilidade fornece um único ponto de entrada para os clientes atingirem o balanceador de carga e, em seguida, serem distribuídos para uma VM disponível. Em uma implantação de aplicação em que esse endereço IP público reside em um único datacenter do Azure, se esse datacenter tiver um problema, nenhum cliente poderá acessar o endereço IP público. O cliente não pode usar sua aplicação, mesmo se houver VMs disponíveis para atender às solicitações do cliente.

Os recursos que podem usar zonas de disponibilidade incluem serviços de zonas e serviços de redundância de zona:

- *Os serviços de zonas* são fornecidos para itens como VMs, um endereço IP público ou um balanceador de carga. O recurso inteiro será executado dentro de uma determinada zona e poderá ser operado por conta própria se outra zona não estiver disponível.
- *Os serviços de redundância de zona* são fornecidos para recursos que podem ser replicados automaticamente entre zonas, como armazenamento com redundância de zona e bancos de dados SQL. O recurso inteiro não será executado dentro de uma determinada zona. Em vez disso, seus dados são distribuídos entre zonas para que continue disponível se uma zona tiver um problema.

O suporte à zona de disponibilidade está disponível para mais de 20 serviços do Azure em mais de dez regiões. O número de serviços e regiões que se integram às zonas de disponibilidade continua aumentando. No entanto, devido às limitações da região, pode haver momentos em que o suporte à zona de disponibilidade não está disponível para recursos essenciais, como VMs. Nesses casos, há outro tipo de redundância de VM que você pode usar em qualquer região que examinarmos na seção 7.2.1: Conjuntos de disponibilidade.

7.2.1 Criar recursos de rede em uma zona de disponibilidade

Para começar a ver parte dessa disponibilidade e redundância em ação, vamos criar alguns recursos comuns, como um endereço IP público e balanceador de carga e, em seguida, VMs. O objetivo aqui é ver que você não precisa definir muitas configurações para aproveitar as zonas de disponibilidade no Azure. Estes são exemplos simples, mas formam o núcleo da maioria dos ambientes de aplicações que você implantaria.

Os endereços IP públicos e os balanceadores de carga podem ser criados em um dos dois níveis disponíveis: básico e padrão. A principal diferença é que o nível padrão permite que o recurso de rede use zonas de disponibilidade. Por padrão, um endereço IP público padrão ou balanceador de carga obtém automaticamente redundância de zona. Não há nenhuma configuração adicional para você concluir. A plataforma do Azure armazena centralmente os metadados para o recurso dentro da região especificada e certifica-se de que o recurso continue sendo executado se uma zona ficar indisponível.

Não se preocupe muito com o que acontece com o balanceador de carga e recursos de rede agora. Lembre-se do que eu disse no início: esses próximos dois ou três capítulos estão relacionados. No capítulo 8, nos aprofundaremos em balanceadores de carga, e tudo isso deve começar a fazer mais sentido.

Experimente agora

Para criar recursos de rede redundantes em zonas de disponibilidade, conclua as etapas a seguir:

1. Selecione o ícone do Cloud Shell na parte superior do painel do portal do Azure.
2. Crie um grupo de recursos, como `azuremolchapter7az`:

```
az group create --name azuremolchapter7az --location westeurope
```

3. Crie um endereço IP público padrão em seu grupo de recursos. Por padrão, um endereço IP público *básico* seria criado e atribuído a apenas uma única zona. O parâmetro `--sku standard` instrui o Azure a criar um recurso redundante de zona cruzada:

```
az network public-ip create \
  --resource-group azuremolchapter7az \
  --name azpublicip \
  --sku standard
```

4 Crie um balanceador de carga que abrange zonas de disponibilidade. Novamente, um balanceador de carga básico seria criado por padrão e atribuído a uma única zona, que não é o design de alta disponibilidade que você deseja para suas aplicações. Especifique um SKU *padrão* para criar um balanceador de carga com redundância de zona, da seguinte maneira:

```
az network lb create \
  --resource-group azuremolchapter7az \
  --name azloadbalancer \
  --public-ip-address azpublicip \
  --sku standard
```

7.2.2 Criar VMs em uma zona de disponibilidade

Para criar uma VM em uma zona de disponibilidade, especifique qual zona deve executar a VM. Para implantar muitas VMs, preferencialmente, você cria e usa um modelo. O modelo define e distribui as zonas para cada uma das VMs. Conforme a demanda do cliente para sua pizzaria online cresce, você também pode atualizar o modelo com o número de VMs que deseja agora e, em seguida, reimplantar o modelo. As novas VMs são distribuídas automaticamente em zonas para você, e não há necessidade de rastrear manualmente em quais zonas as VMs são executadas. No laboratório de fim de capítulo, você usará um modelo para criar e distribuir várias VMs automaticamente. Para ver o processo lógico para especificar uma zona para uma VM, vamos criar uma VM e especificar manualmente a zona.

Experimente agora

Para criar uma VM em uma zona de disponibilidade, conclua as etapas a seguir:

1 No portal do Azure, selecione o ícone do Cloud Shell na parte superior do painel.

2 Crie uma VM com o comando `az vm create` que você usou em capítulos anteriores. Use o parâmetro `--zone` para especificar a zona 1, 2 ou 3 para que a VM seja executada. O exemplo a seguir cria uma VM chamada `zonedvm` na zona 3:

```
az vm create \
  --resource-group azuremolchapter7az \
  --name zonedvm \
  --image ubuntults \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys \
  --zone 3
```

Demora alguns minutos para criar a VM. Quando o processo for concluído, a saída do comando indica a zona em que a VM é executada. Você também pode exibir essas informações com o comando `az vm show`:

```
az vm show \
--resource-group azuremolchapter7az \
--name zonedvm \
--query zones
```

OBSERVAÇÃO Os exemplos nestes exercícios "Experimente agora" são simples, mas são projetados para mostrar que as zonas requerem pouca configuração para serem usadas. Você não integrou o balanceador de carga com redundância de zona e a VM, mas no capítulo 8, criará um ambiente de aplicação mais utilizável que é distribuído entre zonas de disponibilidade. O objetivo aqui é mostrar que a plataforma do Azure manipula a redundância e a distribuição de seus recursos, para que você possa se concentrar na aplicação em si.

7.3 Redundância da VM com conjuntos de disponibilidade

As zonas de disponibilidade são ótimas ao projetar para redundância em um conjunto mais amplo de recursos que compõem suas aplicações e workloads. Recomendo, quando possível, usá-las para novos workloads. No entanto, há momentos em que você não necessariamente deve para adicionar redundância de zona a todos os recursos. Ou talvez você queira criar VMs em uma região do Azure que atualmente não tem suporte à zona de disponibilidade.

Se você quer fornecer redundância somente para VMs, os conjuntos de disponibilidade são ideais. Eles são comprovados, confiáveis e estão disponíveis em todas as regiões. Os conjuntos de disponibilidade contêm um grupo lógico de VMs que indicam à plataforma do Azure que o hardware subjacente em que as VMs são executadas precisa ser escolhido com cuidado. Se você criar duas VMs executadas no mesmo servidor físico e um servidor falhar, as duas VMs terão problemas. Com potencialmente dezenas de milhares ou mais servidores físicos em um datacenter do Azure, é altamente improvável ter essas duas VMs no mesmo servidor, mas é possível. Pode não ser uma falha, mas uma atualização de manutenção que faz com que o servidor físico fique rapidamente indisponível.

E se suas VMs forem executadas no mesmo rack, anexadas ao mesmo equipamento de armazenamento ou de rede? Você está de volta ao único ponto de falha discutido no início do capítulo.

Os conjuntos de disponibilidade permitem que a plataforma do Azure crie suas VMs entre grupos lógicos chamados *domínios de falha* e *domínios de atualização*. Esses domínios lógicos permitem que a plataforma do Azure compreenda os limites físicos dos grupos de hardware para garantir que suas VMs sejam distribuídas uniformemente entre eles. Se uma parte do hardware tiver um problema, apenas algumas VMs no seu conjunto de disponibilidade serão afetadas. Ou se houver atualizações de manutenção a serem aplicadas ao hardware físico, a manutenção afetará apenas algumas de suas VMs. A relação de hardware físico com domínios lógicos de falha e domínios de atualização dentro de um conjunto de disponibilidade é mostrada na Figura 7.4.

As zonas de disponibilidade fazem o mesmo tipo de distribuição em segundo plano, mas é abstraída e não exposta. Mesmo com os conjuntos de disponibilidade, não há muito que você possa configurar. Porém, é útil saber o que está acontecendo nos bastidores.

7.3.1 Domínios de falha

Um *domínio de falha* é um grupo lógico de hardware em um datacenter do Azure. Ele contém hardware que compartilha um equipamento de energia ou rede. Você não controla o que esses domínios de falha são, e não há nada para configurar no nível da VM. A plataforma do Azure rastreia em quais domínios de falha suas VMs são colocadas e distribui novas VMs entre esses domínios de falha para que você sempre tenha VMs disponíveis se a energia ou um comutador de rede falhar.

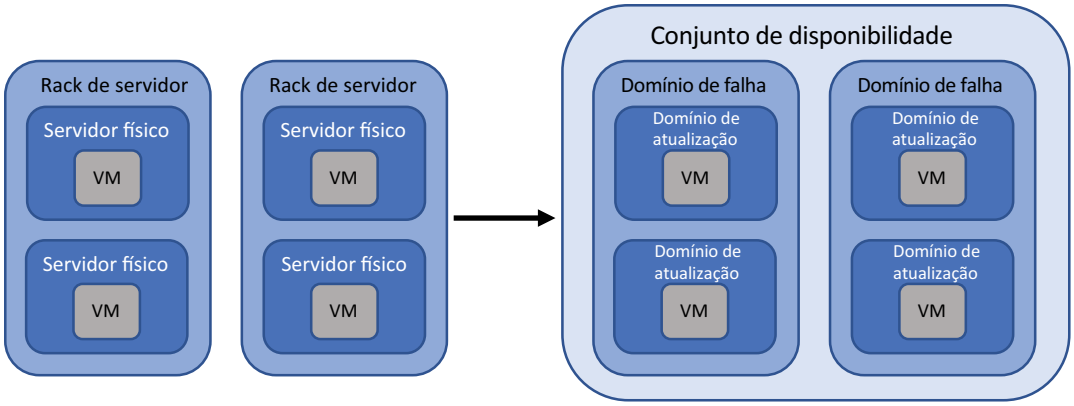


Figura 7.4 O hardware em um datacenter do Azure é dividido logicamente em domínios de atualização e domínios de falha. Esses domínios lógicos permitem que a plataforma do Azure entenda como distribuir suas VMs no hardware subjacente para atender aos seus requisitos de redundância. Este exemplo é básico, pois um domínio de atualização provavelmente contém mais de um servidor físico.

As VMs que usam discos gerenciados (lembre-se de que *todas* as VMs devem usar discos gerenciados) também respeitam limites lógicos de domínio de falha e distribuição. A plataforma do Azure atribui logicamente clusters de armazenamento a domínios de falha para garantir que, à medida que as VMs são distribuídas em grupos de hardware, os discos gerenciados também sejam distribuídos no hardware de armazenamento. Não haveria nenhum ponto na redundância de VM em hardware de servidor se houvesse a possibilidade de que todos os discos gerenciados acabassem em um cluster de armazenamento. E sim, discos gerenciados podem ser usados com zonas de disponibilidade também.

7.3.2 Domínios de atualização

Considerando que os domínios de falha criam um grupo lógico de hardware para proteger contra falhas de hardware, os domínios de atualização protegem contra manutenção de rotina. Para fornecer essa proteção, um domínio de falha é dividido logicamente em domínios de atualização. Novamente, não há nada para configurar aqui. Os domínios de atualização são uma forma de a plataforma do Azure entender como deve distribuir VMs em seu conjunto de disponibilidade.

Os engenheiros do Azure executam a manutenção (principalmente automatizada) e aplicam atualizações em todo o hardware físico em um domínio de atualização e executam a mesma manutenção em todo o hardware no próximo domínio de atualização. Esse trabalho de manutenção é preparado em domínios de atualização para garantir que as VMs em um conjunto de disponibilidade não estejam em execução no hardware que passa pela manutenção ao mesmo tempo. É o mesmo tipo de processo que analisamos com zonas de disponibilidade. A distribuição de seus recursos significa que você não pode ter um cenário no qual todo o hardware subjacente para seus recursos está sendo atualizado ao mesmo tempo.

Não há nenhuma relação entre domínios em vários conjuntos de disponibilidade. Os recursos físicos que compõem os domínios de falha e atualização em um conjunto de disponibilidade podem não ser os mesmos para um segundo conjunto de disponibilidade. Essa conscientização significa que, se você criar vários conjuntos de disponibilidade e distribuir suas VMs entre eles, o domínio de falha 1, por exemplo, nem sempre conterá o mesmo hardware físico.

7.3.3 Distribuir VMs em um conjunto de disponibilidade

Vamos explicar passo a passo e ver como as VMs são distribuídas entre os domínios lógicos de falha e atualização que compõem um conjunto de disponibilidade. Dessa forma, você tem várias VMs que podem administrar sua pizzaria, e os clientes não vão ficar com fome.

Experimente agora

Para ver os conjuntos de disponibilidade em ação, conclua as etapas a seguir para implantar um modelo do Resource Manager:

1. Abra um navegador da Web para um modelo do Resource Manager no repositório de exemplos do GitHub em https://github.com/fouldsy/azure-mol-samples-2nd-ed/tree/master/ 07/availability-set e selecione o botão Implantar no Azure. Você usará um modelo neste exercício para que possa implantar rapidamente VMs e explorará como essas VMs são distribuídas pelo conjunto de disponibilidade.

   O portal do Azure é aberto e solicita alguns parâmetros.

2. Escolha criar um novo grupo de recursos e forneça um nome como `azuremol-chapter7`. Selecione uma região e forneça seus dados de chave SSH (você pode obter no Cloud Shell com cat ~/.ssh/id_rsa.pub).

   O modelo cria um conjunto de disponibilidade que contém três VMs. Essas VMs são distribuídas entre os domínios lógicos de falha e atualização. Com base no que você aprendeu sobre o Resource Manager no capítulo 6, este modelo usa a função `copyindex()` para criar várias VMs e NICs.

3. Para confirmar que você deseja criar os recursos detalhados no modelo, marque a caixa "Aceito os termos e condições acima" e selecione Comprar.

Leva alguns minutos para criar as três VMs no conjunto de disponibilidade. Deixe a implantação continuar no portal enquanto você lê o restante desta seção.

Quando o modelo começa a ser implantado, um conjunto de disponibilidade é criado e o número solicitado de domínios de atualização e de falha é atribuído. As seguintes propriedades foram definidas no modelo de exemplo:

```
"properties": {
    "platformFaultDomainCount": "2",
    "platformUpdateDomainCount": "5",
    "managed": "true"
}
```

Essas propriedades criam um conjunto de disponibilidade com dois domínios de falha e cinco domínios de atualização, como mostrado na Figura 7.5, e indicam que as VMs devem usar discos gerenciados; portanto, faça a distribuição de discos conforme necessário. A região selecionada para o conjunto de disponibilidade determina o número máximo de domínios de falha e atualização. As regiões oferecem suporte a 2 ou 3 domínios de falha e até 20 domínios de atualização.

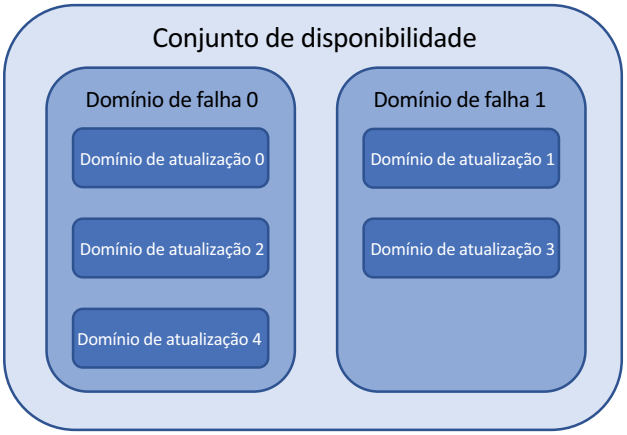


Figura 7.5 O conjunto de disponibilidade que o modelo de exemplo implanta contém dois domínios de falha e cinco domínios de atualização. O sistema de numeração é baseado em zero. Os domínios de atualização são criados em sequência entre os domínios de falha.

À medida que você cria mais VMs em um conjunto de disponibilidade, precisa considerar quantos domínios de atualização usar. Por exemplo, cinco domínios de atualização significam que até 20% das VMs podem não estar disponíveis devido à manutenção:

- Digamos que você tenha 10 VMs em seu conjunto de disponibilidade. Duas dessas VMs podem ser submetidas à manutenção ao mesmo tempo. Se você quisesse permitir que apenas uma VM de cada vez fosse submetida à manutenção, precisaria criar 10 domínios de atualização. Quanto mais domínios de atualização você criar, mais tempo sua aplicação poderá ficar em um estado de manutenção.
- Vamos continuar o exemplo anterior de 10 VMs em 10 domínios de atualização. Agora, há um potencial de interrupção para suas aplicações até que todos os 10 domínios de atualização concluam seu ciclo de manutenção. Se você tiver apenas 5 domínios de atualização, esse período de manutenção será reduzido. Não é necessariamente ruim ter um período de manutenção longo. É mais sobre qual é a sua tolerância possivelmente executar em menos de capacidade total.

É importante lembrar que esses domínios de atualização e os ciclos de manutenção são o que a plataforma do Azure executa em si. Você também precisa levar em consideração suas próprias necessidades de atualização e janelas de manutenção.

Quando a primeira VM é criada, a plataforma do Azure procura ver onde seria a primeira posição de implantação disponível. Este é o domínio de falha 0 e o domínio de atualização 0, como mostrado na Figura 7.6.

Quando a segunda VM é criada, a plataforma do Azure procura ver onde seria a próxima posição de implantação disponível. Agora, este é o domínio de falha 1 e o domínio de atualização 1, como mostrado na Figura 7.7.

Figura 7.6 A primeira VM é criada no domínio de falha 0 e no domínio de atualização 0.

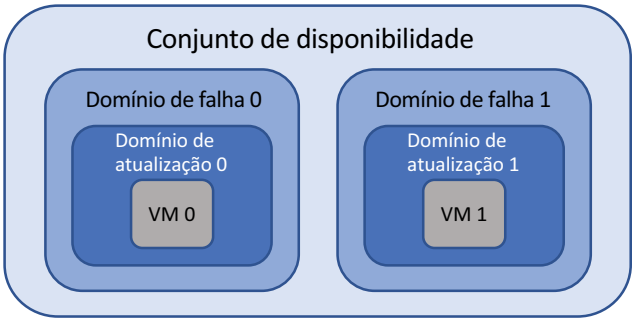


Figura 7.7 Com uma segunda VM criada, as VMs são distribuídas uniformemente entre domínios de falha e atualização. Isso é muitas vezes considerado a quantidade mínima de redundância necessária para proteger suas aplicações.

Seu modelo cria três VMs. O que você acha que acontece a seguir? A plataforma do Azure novamente procura ver onde seria a próxima posição de implantação disponível. Você criou apenas dois domínios de falha, portanto, a VM é criada novamente no domínio de falha 0. Porém, a VM é criada em um domínio de atualização diferente da primeira VM. A terceira VM é criada no domínio de atualização 2, como mostrado na Figura 7.8.

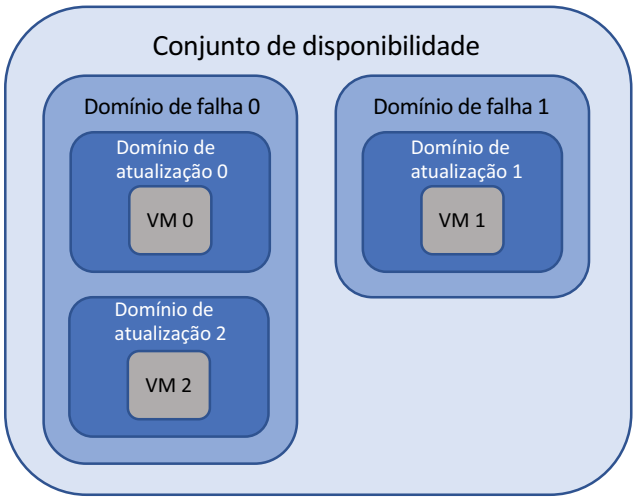


Figura 7.8 A terceira VM é criada novamente no domínio de falha 0, mas no domínio de atualização 2. Embora as VMs 0 e 2 possam compartilhar o risco de falha de hardware, elas estão em domínios de atualização diferentes e, portanto, não serão submetidas à manutenção regular ao mesmo tempo.

As VMs 0 e 2 estão no mesmo domínio de falha e, portanto, uma falha de hardware pode afetar ambas as VMs. Porém, a manutenção de rotina afeta apenas uma dessas VMs de cada vez, pois elas são distribuídas em domínios de atualização. Se você continuar e criar mais VMs, a plataforma do Azure continuará a distribuí-las em domínios de falha e atualização diferentes. Quando os cinco domínios de atualização são usados, a sexta VM é criada novamente no domínio de atualização 0 e o ciclo continua.

7.3.4 Ver a distribuição de VMs em um conjunto de disponibilidade

Agora que você compreende a teoria de como as VMs são distribuídas entre domínios de falha e atualização em um conjunto de disponibilidade, vamos verificar o que aconteceu com sua implantação de modelo do Resource Manager.

Experimente agora

Para ver como suas VMs são distribuídas em um conjunto de disponibilidade, conclua as etapas a seguir:

1. Navegue e selecione Grupo de recursos na barra de navegação à esquerda no portal do Azure.
2. Escolha o grupo de recursos que você criou para sua implantação de modelo, como `azuremolchapter7`.
3. Selecione o conjunto de disponibilidade na lista de recursos, como `azuremolavailabilityset`.

   A janela Visão geral exibe uma lista de VMs e dos domínios associados de falha e atualização, como mostrado na Figura 7.9.

NAME	STATUS	FAULT DOMAIN	UPDATE DOMAIN
vm0	Running	1	1
vm1	Running	0	2
vm2	Running	0	0

Figura 7.9 O conjunto de disponibilidade lista as VMs que ele contém e mostra o domínio de falha e o domínio de atualização para cada VM. Essa tabela permite que você visualize como as VMs são distribuídas entre os domínios lógicos.

Se você estiver particularmente atento, poderá perceber que as VMs não estão perfeitamente alinhadas com a ordem esperada dos domínios de falha e atualização. Algum bug? Provavelmente não. Se você examinar o exemplo na Figura 7.9 e compará-lo com o que os conceitos anteriores lhe indicaram, esperará que as VMs sejam distribuídas como mostrado na tabela 7.1.

Tabela 7.1 Conjunto de disponibilidade de VMs criadas sequencialmente e distribuídas entre domínios

Nome	Domínio de falha	Domínio de atualização
vm0	0	0
vm1	1	1
vm2	0	2

Então, o que deu errado? Nada. Pense novamente como o Resource Manager cria recursos de um modelo. A plataforma do Azure não espera que a primeira VM seja criada antes que a segunda possa ser criada. As três VMs são criadas ao mesmo tempo. Assim, pode haver frações de uma segunda diferença em que a VM é associada a um conjunto de disponibilidade primeiro. Não importa qual seja essa ordem, pois você não controla o que os domínios subjacentes da falha e atualização representam. Isso fica a cargo da plataforma do Azure. Você só precisa garantir que suas VMs *sejam* distribuídas, não *onde*.

Não, eu devo ter números bonitos

Se o comportamento de criação serial de VMs falhar e você *precisar* distribuir as VMs de maneira organizada, poderá instruir o Resource Manager para criar VMs em *série*, não em *paralelo*. Nesse modo, as VMs são criadas uma após a outra e, portanto, o tempo de implantação é maior. Para habilitar esse comportamento em série, use `"mode": "serial"` em seus modelos como parte da função `copyIndex()`. Isso deve distribuir as VMs de uma forma agradável e sequencial para você.

7.4 Laboratório: Implantar VMs altamente disponíveis de um modelo

Este laboratório combina e reforça o que você aprendeu no capítulo 6 sobre o Azure Resource Manager e modelos, com zonas de disponibilidade. Tire algum tempo para examinar o modelo de início rápido de exemplo neste exercício e ver como você pode usar a lógica e as funções para distribuir várias VMs entre as zonas. Não basta implantar o modelo e seguir adiante. Veja como o modelo se baseia nos recursos introduzidos no capítulo 6.

O que é uma cota?

No Azure, as cotas padrão em sua assinatura impedem a implantação acidental de vários recursos e o esquecimento deles, o que lhe custará muito dinheiro. Essas cotas geralmente variam por tipo de recurso e tipo de assinatura e são impostas no nível da região. Você pode ver a lista completa de cotas em http://mng.bz/ddcx.

Quando você começar a criar várias VMs nesses próximos capítulos, poderá ter problemas de cota. Você também pode ter problemas de cota se não excluiu recursos de capítulos e exercícios anteriores. As cotas são um bom sistema que mantém você ciente de seu uso de recursos. As mensagens de erro podem não ser claras, mas se você vir o texto de erro ao longo das linhas de

```
A operação resulta em exceder os limites de cota do Core.
Máximo permitido: 4, Atualmente em uso: 4, Pedido adicional: 2.
```

é uma boa indicação de que você precisa solicitar um aumento em suas cotas. Não há nada complicado ou exclusivo para o Azure. Você pode exibir sua cota atual para uma determinada região da seguinte maneira:

```
az vm list-usage --location eastus
```

Se você tiver problemas com esse laboratório, exclua os dois primeiros grupos de recursos criados neste capítulo, como `azuremolchapter7` e `azuremolchapter7az`. Se você tiver um conjunto de cota padrão baixo, as quatro VMs entre esses grupos de recursos poderão impedir que você termine esse exercício.

Para solicitar um aumento em suas cotas para uma região, siga as etapas descritas em http://mng.bz/Xq2f.

Vamos revisar e implantar um modelo de exemplo que inclui várias VMs em zonas de disponibilidade.

1. Em um navegador da Web, abra o arquivo JSON em https://github.com/Azure/azure-quick-start-templates/blob/master/201-multi-vm-lb-zones/azuredeploy.json e procure o seguinte texto:

   ```
   Microsoft.Compute/virtualMachines
   ```

   A seção VMs é semelhante ao que você usou no capítulo 6, mas observe o valor da propriedade para zonas. Esta seção combina algumas funções disponíveis em modelos para escolher uma das zonas 1, 2 ou 3 à medida que a VM é criada. Dessa forma, você não precisa rastrear manualmente qual VM é executada em qual zona e como implantar VMs adicionais.
2. No navegador da Web, procure cada um dos seguintes itens para ver as seções do endereço IP público e do balanceador de carga:

   ```
   Microsoft.Network/publicIPAddresses
   Microsoft.Network/loadBalancers
   ```

   Os dois recursos usam SKU padrão, que fornece redundância de zona por padrão. Não há nenhuma configuração adicional para que isso funcione. Vamos ver tudo isso em ação.
3. No navegador da Web, abra o modelo de início rápido em http://mng.bz/O69a e selecione o botão Implantar no Azure.
4. Crie ou selecione um grupo de recursos e, em seguida, forneça um nome de usuário e senha para as VMs.
5. Insira um nome DNS exclusivo, como azuremol.

6 Decida se deseja criar VMs Linux ou Windows. As VMs do Windows levam um pouco mais de tempo para criar.

7 Especifique quantas VMs deseja criar, como 3.

8 Marque a caixa para concordar com os termos e condições da implantação do modelo e selecione Comprar, como mostrado na Figura 7.10.

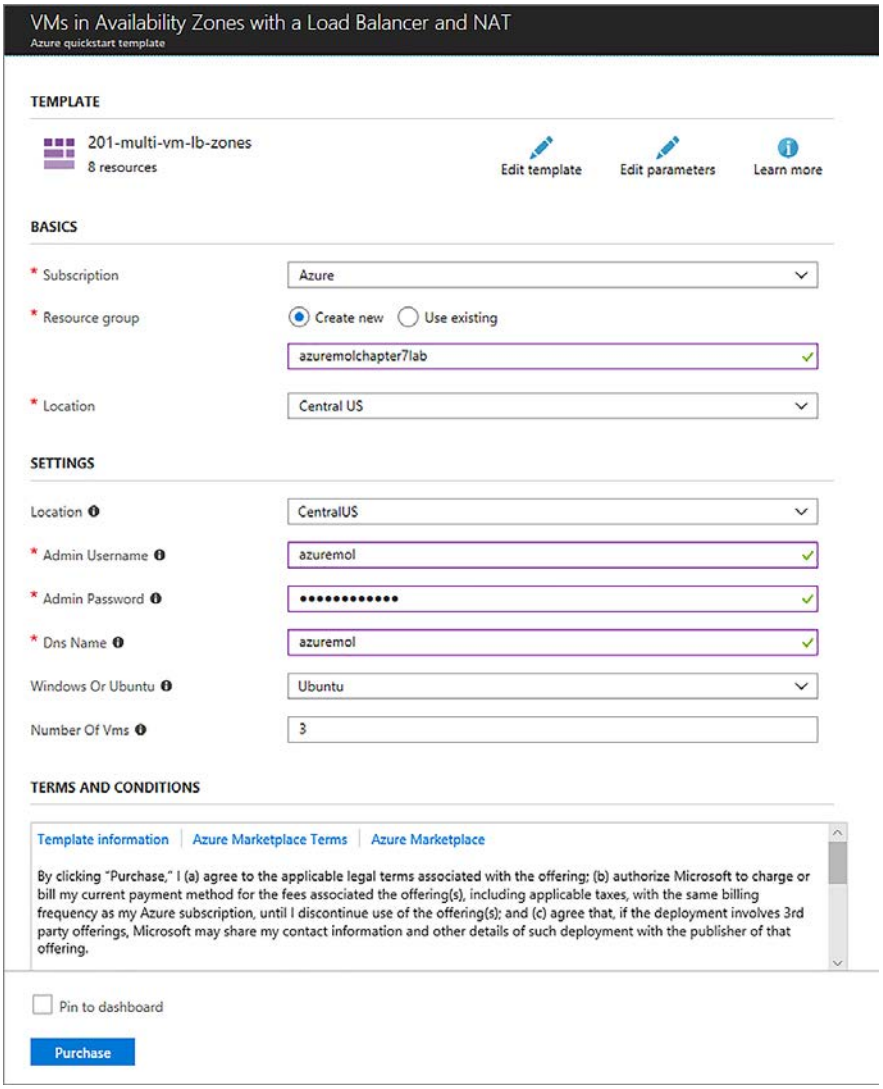


Figura 7.10 Para implantar o modelo de zona de disponibilidade no portal do Azure, especifique um grupo de recursos, nome de usuário e senha e, em seguida, o tipo de sistema operacional e o número de VMs que você deseja criar. O modelo usa loops, `copyIndex()`, `dependsOn`, variáveis e parâmetros, conforme abordado no capítulo 6.

Quando as VMs forem criadas, use o portal do Azure ou o comando `az vm show` para ver como as VMs foram distribuídas entre as zonas. Se você estiver curioso sobre o que o resto do modelo faz com os recursos de rede, o capítulo 8 se aprofunda em balanceadores de carga.

Limpeza no corredor 3

Lembre-se de que, no início do livro, eu disse para você limpar depois para minimizar o custo dos seus créditos gratuitos do Azure. Recomendo que você exclua os grupos de recursos que criou neste capítulo. Os próximos dois capítulos continuam criando várias VMs e instâncias do aplicativo Web, portanto, mantenha os custos e as cotas sob controle.

Sempre que fizer login no portal do Azure, você deve receber uma notificação pop-up com o status dos seus créditos do Azure. Se notar que seu crédito disponível diminuiu muito de um dia para o outro, examine quais grupos de recursos você pode ter esquecido de excluir.

8 Aplicações de balanceamento de carga

Um componente importante de aplicações altamente disponíveis é como distribuir o tráfego em todas as suas VMs. No capítulo 7, você aprendeu a diferença entre os conjuntos de disponibilidade e as zonas de disponibilidade e como é possível criar várias VMs em datacenters ou regiões do Azure para fornecer redundância de aplicação. Mesmo se você tiver todas essas VMs altamente disponíveis e distribuídas, isso não ajudará se apenas uma VM receber todo o tráfego do cliente.

Balanceadores de carga são recursos de rede que recebem o tráfego de entrada da aplicação de seus clientes, examinam o tráfego para aplicar filtros e regras de balanceamento de carga e, em seguida, distribuem as solicitações em um pool de VMs que executam sua aplicação. No Azure, há algumas maneiras diferentes de fazer balanceamento de carga de tráfego, como se você precisar executar SSL sem carga em aplicações grandes que usam o tráfego de rede criptografado. Neste capítulo, você aprenderá sobre os vários componentes do balanceador de carga e como configurar regras e filtros de tráfego e distribuir tráfego para VMs. Você vai desenvolver os componentes de alta disponibilidade do capítulo 8 e se preparar para o capítulo 9, que aborda como escalar recursos.

8.1 Componentes do balanceador de carga do Azure

Os balanceadores de carga no Azure podem trabalhar em dois níveis diferentes: camada 4, onde apenas o tráfego de rede é examinado e distribuído (a camada de transporte, na verdade) e camada 7, onde há uma conscientização dos dados da aplicação dentro do tráfego de rede para ajudar a determinar a distribuição de dados. Ambos os níveis do balanceador de carga funcionam da mesma forma, como mostrado na Figura 8.1.

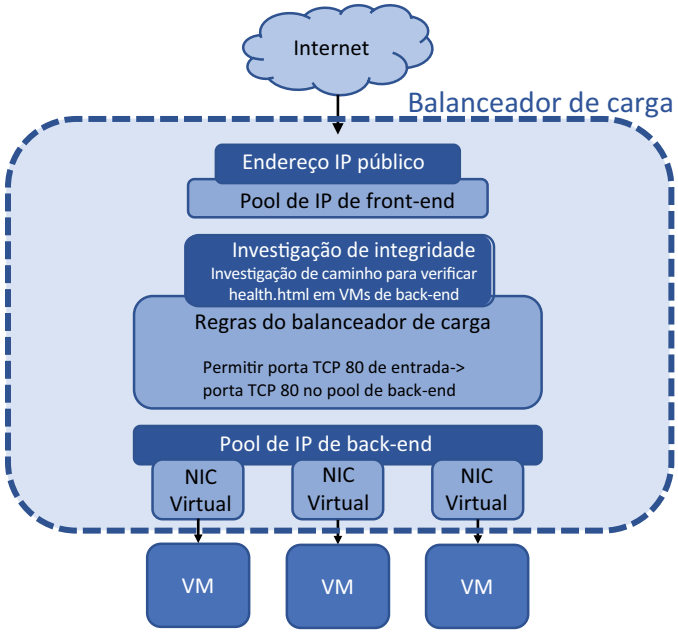


Figura 8.1 O tráfego da Internet insere o balanceador de carga por meio de um endereço IP público anexado a um pool de IPs de front-end. O tráfego é processado por regras de balanceador de carga que determinam como e onde o tráfego deve ser encaminhado. As sondas de integridade anexadas às regras garantem que o tráfego seja distribuído somente para nós íntegros. Em seguida, um pool de back-end de NICs virtuais conectadas às VMs recebe o tráfego distribuído pelas regras do balanceador de carga.

Um balanceador de carga consiste em alguns componentes principais:

- *Pool de IPs de front-end* — Ponto de entrada para o balanceador de carga. Para permitir o acesso a partir da Internet, um endereço IP público pode ser anexado ao pool de IPs de frontend. Endereços IP privados podem ser anexados para balanceadores de carga interna.
- *Sondas de integridade* — Monitore o status das VMs anexadas. Para garantir que o tráfego seja distribuído somente para VMs íntegras e responsivas, as verificações são executadas regularmente para confirmar se uma VM responde corretamente ao tráfego.
- *Regras de balanceador de carga* — Distribuem o tráfego para VMs. Cada pacote de entrada é comparado com as regras, que definem os protocolos de entrada e as portas e, em seguida, são distribuídas em um conjunto de VMs associadas. Se nenhuma regra coincidir com o tráfego de entrada, o tráfego será descartado.
- *Regras de conversão de endereço de rede (NAT)* — rotear o tráfego diretamente para VMs específicas. Por exemplo, se você quiser fornecer acesso remoto via SSH ou RDP, poderá definir regras NAT para encaminhar o tráfego de uma porta externa para uma única VM.
- *Pool de IPs de back-end* — Onde as VMs que executam sua aplicação estão anexadas. As regras do balanceador de carga estão associadas a pools de back-end. Você pode criar pools de back-end diferentes para diferentes partes de suas aplicações.

Gateway de Aplicação do Azure: balanceamento de carga avançado

Os balanceadores de carga do Azure podem trabalhar na camada de rede ou na camada de aplicação. Este capítulo se concentra no balanceador de carga do Azure regular, que funciona na camada de rede (camada 4 ou protocolo de transporte). Nesta camada, o tráfego é examinado e distribuído, mas o balanceador de carga não tem nenhum contexto do que significa o tráfego ou as aplicações que você executar.

Gateway de aplicação do Azure é um balanceador de carga que trabalha na camada de aplicação (camada 7). O Gateway de aplicação obtém insights sobre a aplicação que é executado na VM e pode gerenciar os fluxos de tráfego de maneiras mais avançadas. Um grande benefício do Gateway de aplicação é a capacidade de lidar com o tráfego HTTPS criptografado da Web.

Ao fazer balanceamento de carga em sites com certificados SSL, você pode descarregar o processo que verifica e descriptografa o tráfego dos servidores Web. Em sites com muito tráfego SSL, o processo para verificar e descriptografar o tráfego pode consumir uma grande parte do tempo de computação nas VMs ou aplicativos Web. O Gateway de Aplicação pode verificar e descriptografar o tráfego, passar a solicitação da Web pura para os servidores Web e, em seguida, criptografar novamente o tráfego recebido dos servidores Web e devolvê-lo ao cliente.

O Gateway de Aplicação oferece alguns outros recursos de balanceador de carga mais avançados, como a capacidade de distribuir o tráfego em qualquer ponto de extremidade IP em vez de apenas uma VM do Azure. À medida que cria aplicações que usam mais do que VMs, essas regras de distribuição avançadas podem ser úteis para você. Os mesmos conceitos básicos se aplicam assim como com um balanceador de carga normal, que é o que nos concentraremos neste capítulo para que você entenda como tudo funciona no Azure.

8.1.1 Criar um pool de IPs de front-end

Em capítulos anteriores, você criou VMs que tinham um endereço IP público atribuído diretamente a elas. Em seguida, você usou esse endereço IP público para acessar a VM com uma conexão remota, como SSH ou RDP, ou usou um navegador da Web para acessar um site que foi executado na VM. Quando você usa um balanceador de carga, não se conecta mais diretamente às VMs. Em vez disso, para permitir que o tráfego atinja seu balanceador de carga e seja distribuído para suas VMs, um ou mais endereços IP devem ser atribuídos à interface externa de um balanceador de carga.

Os balanceadores de carga podem operar em um dos dois modos:

- *Balanceador de carga da Internet* — Tem um ou mais *endereços* IP públicos conectados ao pool de IPs de front-end. Um balanceador de carga de Internet recebe diretamente o tráfego da Internet e o distribui para VMs de back-end. Um exemplo comum envolve servidores Web de front-end que os clientes acessam diretamente pela Internet.
- *Balanceador de carga interno* — Tem um ou mais *endereços* IP privados conectados ao pool de IPs de front-end. Um balanceador de carga interno funciona dentro de uma rede virtual do Azure, como para VMs de banco de dados de backend. Você normalmente não expõe bancos de dados de back-end ou camadas de aplicação para o mundo externo. Em vez disso, um conjunto de servidores Web de front-end se conecta a um balanceador de carga interno que distribui o tráfego sem nenhum acesso público direto. A Figura 8.2 mostra como um balanceador de carga interno pode distribuir tráfego para VMs de back-end que estão atrás de um balanceador de carga voltado para o público e VMs da Web de front-end.

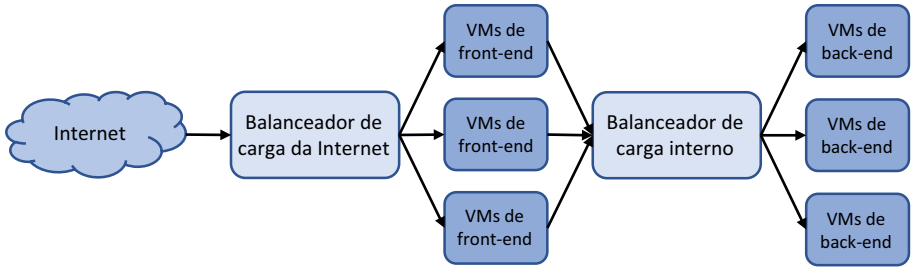


Figura 8.2 Um balanceador de carga da Internet pode ser usado para distribuir o tráfego para VMs de front-end que executam seu site, que então se conectam a um balanceador de carga interno para distribuir o tráfego para uma camada de banco de dados de VMs. O balanceador de carga interno não é acessível publicamente e só pode ser acessado pelas VMs de front-end na rede virtual do Azure.

O modo do balanceador de carga não altera o comportamento do pool de IPs de front-end. Você atribui um ou mais endereços IP que são usados quando o acesso ao balanceador de carga é solicitado. Os endereços IPv4 e IPv6 podem ser configurados para o pool de IPs de front-end, permitindo que você configure as comunicações IPv6 de ponta a ponta entre clientes e suas VMs à medida que o tráfego entra e sai do balanceador de carga.

Experimente agora

Para entender como os componentes do balanceador de carga funcionam juntos, conclua as etapas a seguir para criar um balanceador de carga e um pool de IPs de front-end:

1. Abra o portal do Azure e selecione o ícone do Cloud Shell na parte superior do painel.
2. Crie um grupo de recursos com `az group create`.

   Especifique um nome de grupo de recursos, como `azuremolchapter8`, e um local:

   ```
   az group create --name azuremolchapter8 --location westeurope
   ```

   À medida que você continuar criando no capítulo 7 e desejar usar zonas de disponibilidade, tome cuidado com a região selecionada, para garantir que o suporte à zona de disponibilidade esteja disponível.
3. Crie um endereço IP público com `az network public-ip create`.

   No capítulo 7, você aprendeu que as zonas de disponibilidade fornecem redundância aos recursos de rede, portanto, crie um endereço IP público e especifique um nome, como `publicip`:

   ```
   az network public-ip create \
     --resource-group azuremolchapter8 \
     --name publicip \
     --sku standard
   ```

Para criar um endereço IP público IPv6, você pode adicionar `--version IPv6` ao comando anterior. Para esses exercícios, você pode usar endereços IPv4.

4 Crie o balanceador de carga e atribua o endereço IP público ao pool de IPs de front-end. Para adicionar o endereço IP público, especifique o parâmetro `--public-ip-address`. Se você quiser criar um balanceador de carga interno, use o parâmetro `--private-ip-address`.

Como com o endereço IP público, crie um balanceador de carga padrão com redundância de zona que funcione nas zonas de disponibilidade:

```
az network lb create \
  --resource-group azuremolchapter8 \
  --name loadbalancer \
  --public-ip-address publicip \
  --frontend-ip-name frontendpool \
  --backend-pool-name backendpool \
  --sku standard
```

Nos aprofundaremos no que é o pool de back-end em poucas páginas.

8.1.2 Criar e configurar sondas de integridade

Se uma das VMs que executam sua aplicação tiver um problema, você acha que o balanceador de carga deve continuar a distribuir o tráfego para essa VM? Um cliente que tenta acessar sua pizzaria pode ser direcionado para essa VM e ser incapaz de encomendar qualquer alimento. Um balanceador de carga monitora o status das VMs e pode remover as VMs que têm problemas. O balanceador de carga continua a monitorar a integridade e adiciona a VM de volta ao pool para distribuição de tráfego quando a VM é mostrada para responder corretamente novamente.

Uma sonda de integridade pode funcionar de alguns modos:

- *Baseado em porta* — O balanceador de carga verifica uma resposta da VM em uma porta e protocolo específicos, como a porta TCP 80. Contanto que a VM responda à sonda de integridade na porta TCP 80, a VM permanecerá na distribuição de tráfego do balanceador de carga. Caso contrário, a VM será removida da distribuição de tráfego do balanceador de carga, como mostrado na Figura 8.3. Esse modo não garante que a VM atenda o tráfego conforme o esperado, apenas que a conectividade de rede e o serviço de destino retornem uma resposta.
- *Baseado em caminho HTTP* — Uma página personalizada, como health.html, é gravada e colocada em cada VM. Essa verificação de integridade personalizada pode ser usada para verificar o acesso a um armazenamento de imagem ou conexão de banco de dados. Nesse modo, a VM permanece na distribuição de tráfego do balanceador de carga somente quando a página de verificação de integridade retorna uma resposta 200 de código HTTP, como mostrado na Figura 8.4. Com uma sonda de integridade baseada em porta, o servidor Web real pode ser executado, mas não tem nenhuma conexão de banco de dados. Com uma página de verificação de integridade personalizada, o balanceador de carga pode confirmar que a VM é capaz de veicular o tráfego real para o cliente.

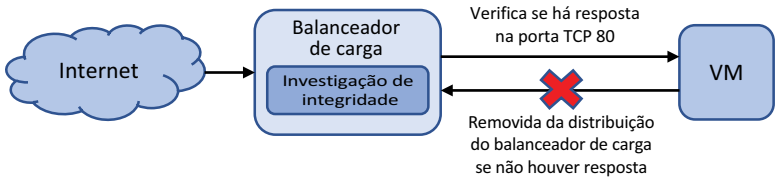


Figura 8.3 Uma sonda de integridade de balanceador de carga baseada em porta verifica uma resposta da VM em uma porta e protocolo definidos. Se a VM não responder dentro do limite determinado, a VM será removida da distribuição de tráfego do balanceador de carga. Quando a VM começa a responder corretamente novamente, a sonda de integridade detecta a alteração e adiciona a VM de volta à distribuição de tráfego do balanceador de carga.

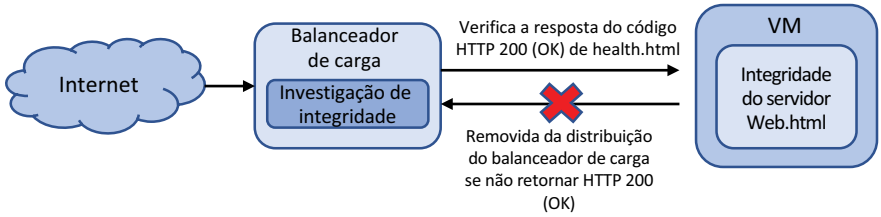


Figura 8.4 Uma VM que executa um servidor Web e tem uma página health.html personalizada permanece na distribuição de tráfego do balanceador de carga, desde que a sonda de integridade receba uma resposta de código HTTP 200 (OK). Se o processo do servidor Web encontrar um problema e não puder retornar as páginas solicitadas, elas serão removidas da distribuição de tráfego do balanceador de carga. Esse processo fornece uma verificação mais completa do estado do servidor Web que os testes de integridade baseados em porta.

É necessário trabalho adicional para criar a página de verificação de integridade personalizada, mas a melhora da experiência do cliente vale a pena. A página de verificação de integridade não precisa ser complicada. Pode ser uma página HTML básica que é usada para confirmar se o servidor Web pode veicular páginas. Sem a página de verificação de integridade, se o processo do servidor Web tiver um problema, a VM ainda estaria disponível na porta TCP 80, portanto, a sonda de integridade baseada em porta acreditaria que a VM está íntegra. Uma sonda de integridade baseada em caminho HTTP requer que o servidor Web retorne corretamente uma resposta HTTP. Se o processo do servidor Web trava ou falha, uma resposta HTTP não é enviada, portanto, a VM é removida da distribuição de tráfego do balanceador de carga.

Com que frequência a sonda de integridade verifica a VM e qual é a resposta também podem ser configuradas com dois parâmetros:

- *Intervalo* — Define a frequência com que a sonda de integridade verifica o status da VM. Por padrão, a sonda de integridade verifica o status a cada 15 segundos.
- *Limite* — Define quantas falhas de resposta consecutivas a sonda de integridade recebe antes da VM ser removida da distribuição de tráfego do balanceador de carga. Por padrão, a sonda de integridade tolera duas falhas consecutivas antes da VM ser removida da distribuição de tráfego do balanceador de carga.

Experimente agora

Para criar uma sonda de integridade para o balanceador de carga como na Figura 8.4, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione o ícone do Cloud Shell na parte superior do painel.
2. Especifique um nome para a sonda de integridade, como `healthprobe`. Para configurar a sonda de integridade para um servidor Web, especifique a porta HTTP 80 e, em seguida, defina uma página de verificação de integridade personalizada em health.html. Na seção 8.2, você criará esta página de verificação de integridade em suas VMs. Para mostrar como o intervalo e o limite para a resposta da sonda de integridade podem ser configurados, defina um intervalo de 10 segundos e um limite de três falhas consecutivas:

```
az network lb probe create \
  --resource-group azuremolchapter8 \
  --lb-name loadbalancer \
  --name healthprobe \
  --protocol http \
  --port 80 \
  --path health.html \
  --interval 10 \
  --threshold 3
```

Depois que a sonda de integridade é criada, como você a faz verificar o status de suas VMs? As sondas de integridade estão associadas às regras do balanceador de carga. A mesma sonda de integridade pode ser usada com várias regras de balanceador de carga. Lembre-se do capítulo 5, em que você criou grupos de segurança de rede (NSGs) e regras. Esses NSGs podem ser associados a várias VMs ou sub-redes de rede virtual. Uma relação um para muitos semelhante se aplica às sondas de integridade.

Vamos ver como colocar sua sonda de integridade em funcionamento e criar regras de balanceador de carga.

8.1.3 Definir distribuição de tráfego com regras de balanceador de carga

Quando o tráfego é direcionado pelo balanceador de carga para as VMs de back-end, você pode definir quais condições fazem com que o usuário seja direcionado para a mesma VM. Convém que o usuário mantenha uma conexão com a mesma VM durante uma única sessão ou permita que ele retorne e mantenha sua afinidade de VM com base no endereço IP de origem. A Figura 8.5 mostra um exemplo do modo de afinidade de sessão padrão.

No modo de afinidade de sessão, o fluxo de tráfego é manipulado por um hash de 5 tuplas que usa o endereço IP de origem, a porta de origem, o endereço IP de destino, a porta de destino e o tipo de protocolo. Basicamente, para cada solicitação que um usuário faz ao seu servidor Web na porta TCP 80, ele é direcionado para a mesma VM de back-end durante essa sessão.

O que acontece se o cliente fechar a sessão do navegador? Na próxima vez que ele se conectar, uma nova sessão será iniciada. Como o balanceador de carga distribui o tráfego em todas as VMs íntegras no pool de IPs de back-end, é possível que o usuário se conecte novamente à mesma VM; mas quanto mais VMs você tiver no pool de IPs de back-end, maior será a chance de o usuário se conectar a uma VM diferente.

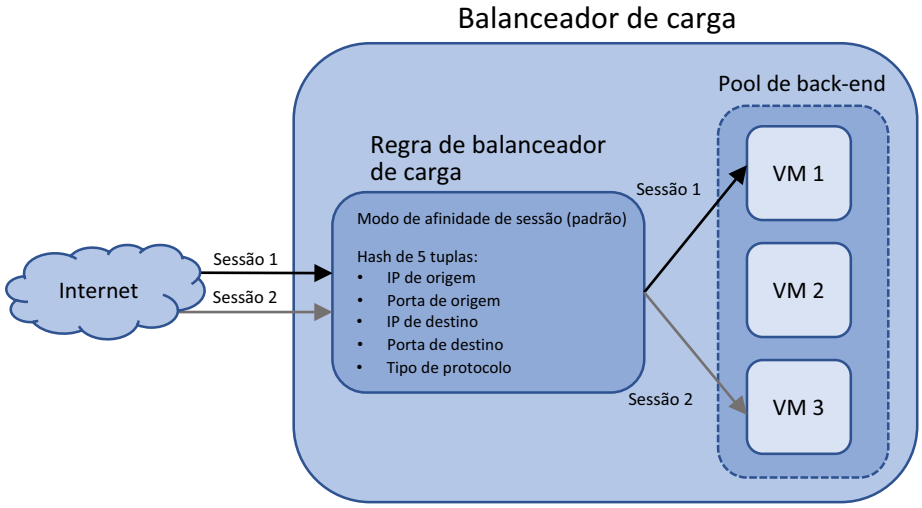


Figura 8.5 No modo de afinidade de sessão, o usuário se conecta à mesma VM de back-end somente durante a sessão.

Como proprietário e desenvolvedor da aplicação, você talvez queira que o usuário se conecte à mesma VM de antes quando iniciar outra sessão. Por exemplo, se sua aplicação manipula transferências de arquivos ou usa UDP em vez de TCP, você provavelmente deseja que a mesma VM continue a processar as solicitações do usuário. Nesses cenários, você pode configurar as regras do balanceador de carga para a afinidade de IP de origem. A Figura 8.6 mostra um exemplo de modo de afinidade de IP de origem.

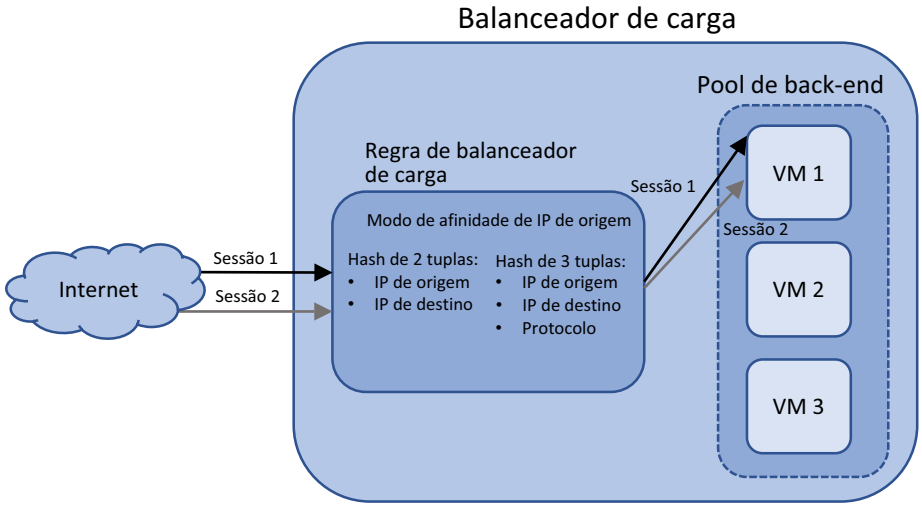


Figura 8.6 Quando você configura as regras do balanceador de carga para usar o modo de afinidade de IP de origem, o usuário pode fechar e, em seguida, iniciar uma nova sessão, mas continuar a se conectar à mesma VM de back-end. O modo de afinidade de IP de origem pode usar um hash de 2 tuplas que usa o endereço IP de origem e destino ou um hash de 3 tuplas que também usa o protocolo.

Experimente agora

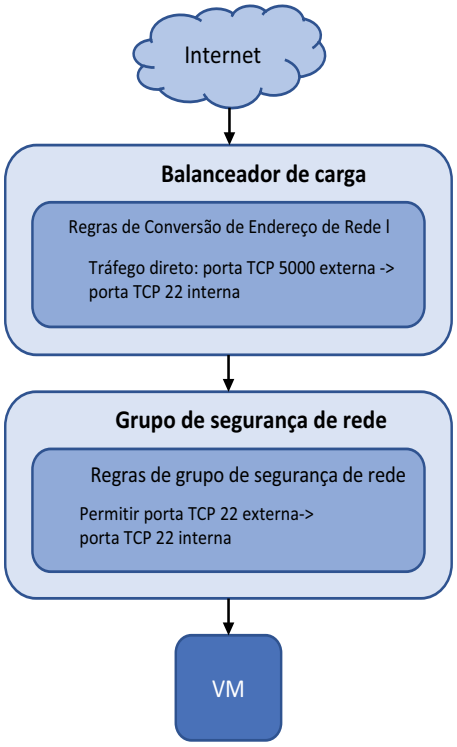
Para criar uma regra de balanceador de carga que usa uma sonda de integridade, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione o ícone do Cloud Shell na parte superior do painel.
2. Para criar uma regra de balanceador de carga, especifique um nome para a regra, como `httprule`.
3. Forneça a porta externa na qual o tráfego é recebido e a porta interna para a qual distribuir o tráfego. Neste exemplo básico, o tráfego é recebido na porta 80 e, em seguida, distribuído na porta 80:

```
az network lb rule create \
  --resource-group azuremolchapter8 \
  --lb-name loadbalancer \
  --name httprule \
  --protocol tcp \
  --frontend-port 80 \
  --backend-port 80 \
  --frontend-ip-name frontendpool \
  --backend-pool-name backendpool \
  --probe-name healthprobe
```

Se você executar vários sites em uma VM que responda em diferentes portas, uma determinada regra poderá direcionar o tráfego para um site específico na VM.

8.1.4 Roteamento direto de tráfego com regras de conversão de endereço de rede



As regras do balanceador de carga distribuem o tráfego entre os pools de back-end das VMs, portanto, não há nenhuma garantia de que você possa se conectar a uma determinada VM para fins de manutenção ou gerenciamento. Como você pode se conectar a uma VM específica que está por trás de um balanceador de carga? Uma parte final da configuração do balanceador de carga para examinar são regras de conversão de endereço de rede (NAT), que permitem que você controle o fluxo de tráfego específico para direcioná-lo para uma única VM. A Figura 8.7 mostra como as regras de NAT encaminham tráfego específico para VMs individuais.

Figura 8.7 O tráfego no balanceador de carga é processado por regras de NAT. Se um protocolo e uma porta corresponderem a uma regra, o tráfego será encaminhado para a VM de back-end definida. Nenhuma sonda de integridade está conectada, portanto, o balanceador de carga não verifica se a VM pode responder antes de encaminhar o tráfego. O tráfego deixa o balanceador de carga e, em seguida, é processado por regras de NSG. Se o tráfego for permitido, ele será passado para a VM.

As regras de NAT funcionam junto com as regras do NSG. A VM pode receber o tráfego somente se houver uma regra de NSG que permita o mesmo tráfego que a regra de NAT do balanceador de carga.

Por que você pode criar regras de NAT? E se você quiser usar SSH ou RDP para se conectar a uma VM específica (e não estiver usando o Azure Bastion, que mencionei no capítulo 2) ou use ferramentas de gerenciamento para se conectar a um servidor de banco de dados de back-end? Se o balanceador de carga distribuir o tráfego entre as VMs de back-end, você terá que tentar se conectar diversas vezes e ainda talvez não consiga se conectar à VM desejada.

Manter as coisas seguras

Analisaremos alguns tópicos de segurança na parte 3 do livro, mas a segurança deve ser uma consideração sempre presente à medida que você cria e executa aplicações no Azure. Segurança não deve ser algo deixado para mais tarde. Com o surgimento da computação na nuvem e de VMs descartáveis e aplicativos Web, é fácil ignorar algumas práticas recomendadas básicas de segurança. Principalmente se você trabalha no Azure como parte de uma assinatura corporativa mais ampla, certifique-se de que os recursos criados não forneçam acidentalmente uma maneira de os invasores obterem acesso à sua infraestrutura.

Que tipo de coisas são ruins? Bem, algumas das coisas que você já viu neste livro. As portas de gerenciamento remoto para SSH e RDP não devem ser abertas à Internet pública como você fez. Ou pelo menos você deve restringir o acesso de um intervalo de endereços IP específico.

A prática recomendada é usar um serviço gerenciado, como o Azure Bastion, ou criar manualmente uma VM segura com gerenciamento remoto disponível. Conforme necessário, você se conecta ao host do Azure Bastion sua VM protegida e, em seguida, se conecta pela rede virtual do Azure interna a VMs adicionais. Você usou essa abordagem de VM de jump-box básica no capítulo 5. Essa abordagem minimiza a pegada de ataque e reduz a necessidade de regras de NSG e regras de NAT do balanceador de carga. O capítulo 16 discute a Central de Segurança do Azure e mostra como você pode solicitar e abrir dinamicamente portas de gerenciamento remoto para um período específico, que é o melhor de ambos os mundos.

Mesmo que você trabalhe em uma assinatura privada do Azure que não tenha conectividade com outras assinaturas do Azure na escola ou no trabalho, tente minimizar a quantidade de conectividade remota que você fornece.

Experimente agora

Para criar uma regra de NAT do balanceador de carga, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione o ícone do Cloud Shell na parte superior do painel.
2. Para criar uma regra de NAT do balanceador de carga, defina um nome, como `natrulessh`, e o pool de IPs de front-end a ser usado. A regra de NAT examina o tráfego em um determinado protocolo e porta, como a porta TCP 50001. Quando há uma correspondência de regra, o tráfego é encaminhado para a porta de back-end 22:

```
az network lb inbound-nat-rule create \
  --resource-group azuremolchapter8 \
  --lb-name loadbalancer \
  --name natrulessh \
  --protocol tcp \
  --frontend-port 50001 \
  --backend-port 22 \
  --frontend-ip-name frontendpool
```

Neste ponto, você criou um balanceador de carga básico. Examine como os componentes do balanceador de carga se uniram:

```
az network lb show \
  --resource-group azuremolchapter8 \
  --name loadbalancer
```

Um endereço IP público foi atribuído ao pool de IPs de front-end e você criou uma sonda de integridade para verificar o status em uma página de integridade personalizada para um servidor Web. Uma regra de balanceador de carga foi criada para distribuir o tráfego da Web de seus clientes para um pool de back-end. A regra usa a sonda de integridade. Você também tem uma regra de NAT de balanceador de carga que permite o tráfego de SSH, mas ainda não há nenhuma VM para receber esse tráfego. Os clientes da sua pizzaria estão com fome, então vamos criar algumas VMs que podem executar seu aplicativo Web e para as quais o balanceador de carga pode distribuir tráfego.

8.1.5 Atribuir grupos de VMs a pools de back-end

A seção final do balanceador de carga define pools de back-end que incluem uma ou mais VMS. Esses pools de back-end contêm VMs que executam os mesmos componentes de aplicação, o que permite que o balanceador de carga distribua o tráfego para um determinado pool de back-end e confie que qualquer VM nesse pool pode responder corretamente à solicitação do cliente. A Figura 8.8 detalha como os pools de back-end agrupam logicamente as VMs que executam as mesmas aplicações.

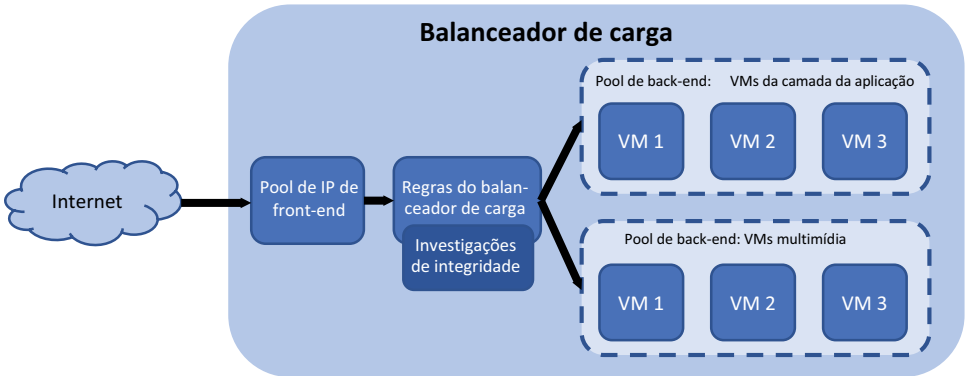


Figura 8.8 Um ou mais pools de back-end podem ser criados em um balanceador de carga. Cada pool de back-end contém uma ou mais VMs que executam o mesmo componente de aplicação. Neste exemplo, um pool de back-end contém VMs que executam a camada de aplicativo Web e outro pool de back-end contém as VMs que veiculam multimídia, como imagens e vídeo.

Você cria e usa um balanceador de carga com VMs, mas tudo funciona no nível da rede virtual. O pool de IPs de frontend usa endereços IP públicos ou privados. A sonda de

integridade examina as respostas em uma determinada porta ou protocolo. Mesmo quando uma sonda HTTP é usada, o balanceador de carga procura uma resposta de rede positiva. As regras do balanceador de carga concentram-se em como distribuir o tráfego de uma porta externa no pool de frontend para uma porta no pool de back-end.

Quando você atribui VMs ao pool de back-end que recebe o tráfego distribuído pelo balanceador de carga, é a NIC virtual que se conecta ao balanceador de carga. A VM é anexada à NIC virtual. Pense no capítulo 5, e essa separação de VMs e NIC virtual faz sentido em termos de como os recursos são gerenciados. As regras de NSG controlam qual tráfego pode entrar na VM, mas elas são aplicadas a uma sub-rede de rede virtual ou NIC virtual, não à VM.

O que isso significa para configurar pools de IP de back-end? Você deve criar o restante de seus recursos de rede virtual antes de conectar uma VM ao balanceador de carga. As etapas para criar os recursos de rede devem ser uma recapitulação do que você aprendeu alguns capítulos atrás, então vamos ver o quanto você se lembra.

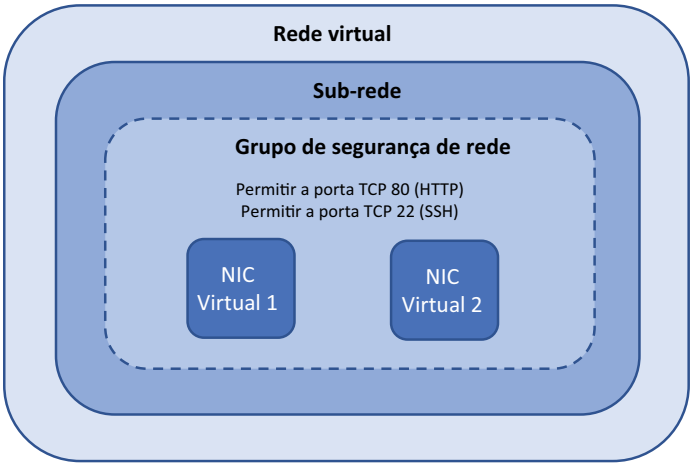


Figura 8.9 Para preparar a rede virtual, neste exercício, você criará uma rede, uma sub-rede e NICs virtuais protegidas por um NSG. As regras anexadas ao NSG permitem o tráfego HTTP e SSH.

Experimente agora

Para criar os recursos de rede adicionais, como mostrado na Figura 8.9, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione o ícone do Cloud Shell na parte superior do painel.
2. Criar uma rede virtual e uma sub-rede:

```
az network vnet create \
  --resource-group azuremolchapter8 \
  --name vnetmol \
  --address-prefixes 10.0.0.0/16 \
  --subnet-name subnetmol \
  --subnet-prefix 10.0.1.0/24
```

Na prática, há uma boa chance de que esses recursos de rede já existam. Além disso, esses nomes e intervalos de endereços IP são os mesmos usados no capítulo 5. Você deve apagar os recursos do Azure no final de cada capítulo, de modo que essa reutilização de intervalos de IP não deve ser um problema. Apenas esteja ciente de que você normalmente não criará uma rede virtual e uma sub-rede toda vez que criar um balanceador de carga. Em vez disso, você pode usar os recursos de rede virtual existentes que já estão em vigor.

3 Criar um NSG:

```
az network nsg create \
  --resource-group azuremolchapter8 \
  --name webnsg
```

4 Crie uma regra de NSG que permita que o tráfego da porta TCP 80 atinja suas VMs. Essa regra é necessária para que as VMs do servidor Web recebam e respondam ao tráfego do cliente:

```
az network nsg rule create \
  --resource-group azuremolchapter8 \
  --nsg-name webnsg \
  --name allowhttp \
  --priority 100 \
  --protocol tcp \
  --destination-port-range 80 \
  --access allow
```

5 Adicione outra regra para permitir o tráfego SSH para gerenciamento remoto. Essa regra de NSG funciona com a regra de NAT do balanceador de carga criada na seção 8.1.4 para uma de suas VMs:

```
az network nsg rule create \
  --resource-group azuremolchapter8 \
  --nsg-name webnsg \
  --name allowssh \
  --priority 101 \
  --protocol tcp \
  --destination-port-range 22 \
  --access allow
```

6 Associe o NSG à sub-rede criada na etapa 2. As regras de NSG são aplicadas a todas as VMs que se conectam a essa sub-rede:

```
az network vnet subnet update \
  --resource-group azuremolchapter8 \
  --vnet-name vnetmol \
  --name subnetmol \
  --network-security-group webnsg
```

7 O balanceador de carga funciona com NICs virtuais, portanto, crie duas NICs virtuais e conecte-as à sub-rede da rede virtual. Também especifique o nome do balanceador de carga e o pool de endereços de back-end aos quais as NICs virtuais se conectam. A regra de NAT do balanceador de carga só é anexada a essa primeira NIC virtual criada:

```
az network nic create \
  --resource-group azuremolchapter8 \
  --name webnic1 \
```

```
  --vnet-name vnetmol \
  --subnet subnetmol \
  --lb-name loadbalancer \
  --lb-address-pools backendpool \
  --lb-inbound-nat-rules natrulessh
```

8 Crie a segunda NIC da mesma forma, menos a regra de NAT do balanceador de carga:

```
az network nic create \
  --resource-group azuremolchapter8 \
  --name webnic2 \
  --vnet-name vnetmol \
  --subnet subnetmol \
  --lb-name loadbalancer \
  --lb-address-pools backendpool
```

8.2 Criar e configurar VMs com o balanceador de carga

Vamos pausar para explorar o que você criou. A Figura 8.10 mostra o cenário global dos seus recursos de rede e balanceador de carga. Observe como esses recursos são integrados. O balanceador de carga não pode existir por si só. As NICs virtuais devem estar conectadas

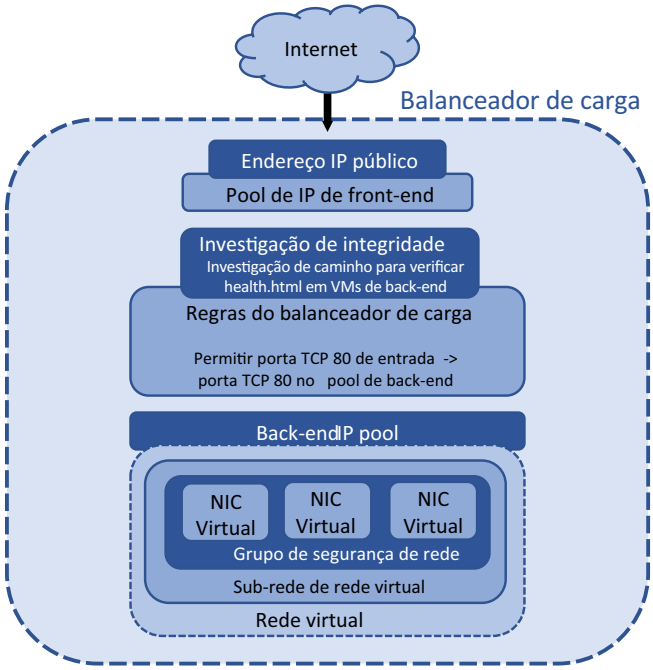


Figura 8.10 Nenhuma VM foi criada aqui. A configuração do balanceador de carga lida com recursos de rede virtual. Há uma relação estreita entre o balanceador de carga e os recursos de rede virtual.

ao balanceador de carga para que qualquer tráfego seja distribuído. Essas NICs virtuais requerem uma rede virtual e sub-rede e, preferencialmente, são protegidas por um NSG. As VMs que executam sua aplicação não têm quase nada a ver com as etapas para criar e configurar o balanceador de carga.

Você criou muitos recursos de rede e configurou várias partes do balanceador de carga. O endereço IP público e o balanceador de carga foram criados em uma zona de disponibilidade como recursos com redundância de zona, portanto, vamos criar duas VMs em diferentes zonas para reforçar como as zonas de disponibilidade melhoram a alta disponibilidade de suas aplicações.

Se você usar conjuntos de disponibilidade em vez de zonas de disponibilidade, é aqui que criará primeiro um conjunto de disponibilidade e adicionará suas VMs a ele. Em seguida, a plataforma do Azure distribui as VMs entre os domínios de falha e atualização. Você deseja maximizar o uso da alta disponibilidade do Azure para sua pizzaria e, portanto, usa zonas de disponibilidade.

Experimente agora

Para criar as VMs e anexá-las ao balanceador de carga, conclua as etapas a seguir:

1 Crie a primeira VM e atribua-a a uma zona de disponibilidade com `--zone 1`:

```
az vm create \
  --resource-group azuremolchapter8 \
  --name webvm1 \
  --image ubuntults \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys \
  --zone 1 \
  --nics webnic1
```

2 Crie a segunda VM, atribua-a à zona de disponibilidade 2 e anexe a segunda NIC virtual que você criou anteriormente, usando `--nics webnic2`:

```
az vm create \
  --resource-group azuremolchapter8 \
  --name webvm2 \
  --image ubuntults \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys \
  --zone 2 \
  --nics webnic2
```

Para ver o balanceador de carga em ação, você precisa instalar um servidor Web básico, como fez no capítulo 2. Você também pode testar a regra de NAT do balanceador de carga. Consegue começar a ver como todos esses componentes do Azure estão relacionados e desenvolvem uns sobre os outros?

Experimente agora

No capítulo 5, abordamos o agente SSH. O agente SSH permite que você passe uma chave SSH de uma VM para a próxima. Somente VM1 tem uma regra de NAT do balanceador de carga, portanto, você precisa usar o agente para se conectar a VM2. Para instalar um servidor Web em suas VMs, conclua as etapas a seguir:

1. Inicie o agente SSH e adicione sua chave SSH para se conectar a ambas as VMs:

```
eval $(ssh-agent) && ssh-add
```

2. Obtenha o endereço IP público anexado ao pool de IPs de front-end do balanceador de carga. Essa é a única maneira de rotear o tráfego pelas VMs:

```
az network public-ip show \
  --resource-group azuremolchapter8 \
  --name publicip \
  --query ipAddress \
  --output tsv
```

3. Agora você está pronto para usar o SSH para a VM 1. Especifique o endereço IP público do balanceador de carga (substitua `<your-ip-address>` no comando a seguir) e a porta que foi usada com a regra de NAT do balanceador de carga, como 50001. O parâmetro `-A` usa o agente SSH para passar pelas chaves SSH:

```
ssh -A azuremol@<your-ip-address> -p 50001
```

   No capítulo 2, você usou `apt-get` para instalar a pilha LAMP inteira, incluindo o servidor Web Apache. Vamos ver algo um pouco diferente do servidor Web Apache com o servidor Web independente, mas poderoso NGINX. Em uma VM do Windows, normalmente é aqui onde você instalaria o IIS. Execute o seguinte comando para instalar o servidor Web NGINX.

```
sudo apt update && sudo apt install -y nginx
```

4. No repositório de exemplos do GitHub que você usou em capítulos anteriores, há uma página da Web HTML básica e uma página de verificação de integridade para a sonda de integridade do balanceador de carga. Clone esses exemplos para a VM:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

5. Copie a página HTML de exemplo e a verificação de integridade para o diretório do servidor Web:

```
sudo cp azure-mol-samples-2nd-ed/08/webvm1/* /var/www/html/
```

6. Agora você precisa se conectar à segunda VM e instalar o servidor Web NGINX e o código de exemplo. Lembre-se do agente SSH. Você deve ser capaz de usar SSH da VM 1 para a VM 2 no endereço IP interno privado:

```
ssh 10.0.1.5
```

7 Instale o servidor Web NGINX:

```
sudo apt update && sudo apt install -y nginx
```

8 Clone os exemplos do GitHub para a VM:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

9 Copie a página HTML de exemplo e a verificação de integridade para o diretório do servidor Web:

```
sudo cp azure-mol-samples-2nd-ed/08/webvm2/* /var/www/html/
```

Abra um navegador da Web e conecte-se ao endereço IP público do seu balanceador de carga. A página da Web básica carrega e mostra que sua pizzaria agora tem VMs redundantes em zonas de disponibilidade que são executadas atrás de um balanceador de carga, como mostrado na Figura 8.11. Talvez seja necessário forçar a atualização do navegador da Web para ver se a VM 1 e a VM2 respondem à medida que o balanceador de carga distribui o tráfego entre elas.

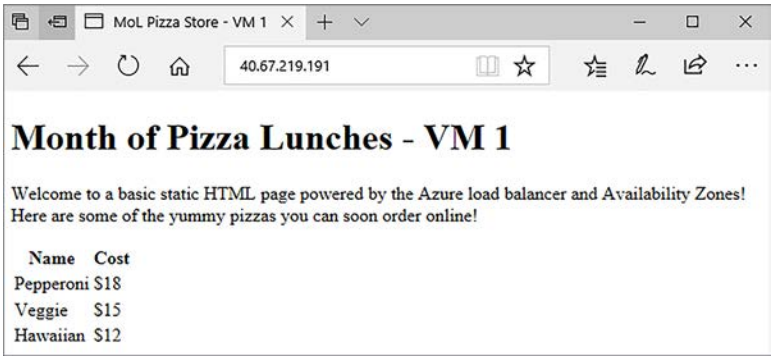


Figura 8.11 Quando você abre o endereço IP público do balanceador de carga em um navegador da Web, o tráfego é distribuído para uma das VMs que executam seu site básico. A sonda de integridade do balanceador de carga usa a página health.html para confirmar que o servidor Web responda com um código HTTP 200 (OK). Em caso afirmativo, a VM está disponível como parte da distribuição de tráfego do balanceador de carga.

8.3 Laboratório: Exibir modelos de implantações existentes

Este capítulo une o que você aprendeu em vários capítulos anteriores. Você criou recursos de rede, como no capítulo 5. Você fez o balanceador de carga e as VMs altamente disponíveis com zonas de disponibilidade, como no capítulo 7. E você instalou e implantou um servidor Web e arquivos de exemplo, como no capítulo 2. Sua pizzaria percorreu um longo caminho a partir da página da Web básica em uma única VM no início do livro.

Para associar mais um tema de um capítulo anterior, neste laboratório, eu quero que você explore todos os recursos que compõem o balanceador de carga. Para fazer isso, você examina o modelo do Resource Manager, como aprendeu no capítulo 6. O objetivo deste laboratório é ver como um único modelo pode criar e configurar o que está envolvido em muitas páginas e vários comandos da CLI. E, acredite, isso envolveria ainda mais comandos do PowerShell. Siga estas etapas:

1. Abra o portal do Azure.
2. Navegue e selecione Grupo de recursos na barra de navegação à esquerda no portal.
3. Escolha seu grupo de recursos, como azuremolchapter 8.
4. Escolha Exportar modelo na barra no lado esquerdo, como mostrado na Figura 8.12.
5. Para ver a parte relevante do modelo, selecione cada um dos recursos mostrados na lista. Reserve alguns minutos para explorar esse modelo e veja como todos os recursos e componentes configurados na CLI do Azure estão presentes.

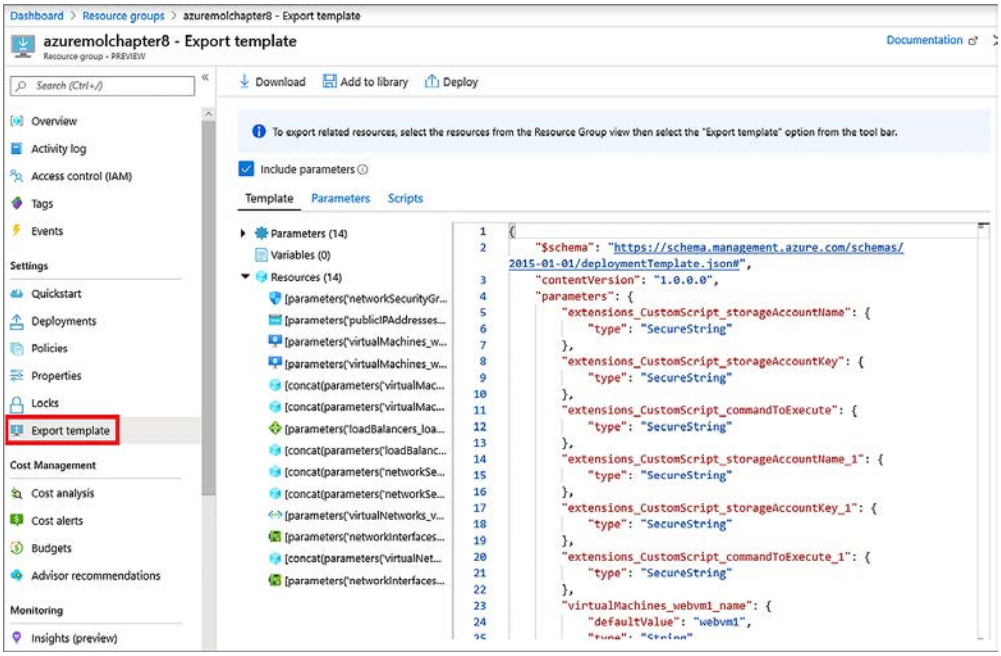


Figura 8.12 No portal do Azure, selecione o grupo de recursos do balanceador de carga e visualize o modelo do Resource Manager.

Um modelo facilita muito a implantação de um ambiente de aplicação altamente disponível, com redundância e com balanceamento de carga. Você pode alterar o nome, as regras e o modo de distribuição do balanceador de carga e permitir que o modelo implante e configure todo o ambiente de aplicação para você.

Não se esqueça de excluir esse grupo de recursos para aproveitar ao máximo seus créditos gratuitos do Azure.

9 Aplicações escaláveis

Nos dois capítulos anteriores, examinamos como criar aplicações altamente disponíveis e usar balanceadores de carga para distribuir o tráfego para várias VMs que executam sua aplicação. Mas como executar e gerenciar várias VMs com eficiência e executar o número certo de instâncias de VM quando seus clientes mais precisam deles? Quando a demanda do cliente aumenta, você deseja aumentar automaticamente a escala da sua aplicação para lidar com essa demanda. E quando a demanda diminui, como no meio da noite, quando a maioria das pessoas sem crianças estão dormindo, você quer que a aplicação diminua em escala e economize algum dinheiro.

No Azure, você pode escalar automaticamente os recursos de IaaS com conjuntos de escala de máquina virtual. Esses conjuntos de escala executam VMs idênticas, normalmente distribuídas por trás de um balanceador de carga ou gateway de aplicação. Você define as regras de dimensionamento automático que aumentam ou diminuem o número de instâncias de VM à medida em que a demanda do cliente é alterada. O balanceador de carga ou o gateway de aplicativo distribui automaticamente o tráfego para as novas instâncias de VM, o que permite que você se concentre em como criar e executar melhor seus aplicativos. Os conjuntos de escala oferecem controle dos recursos de IaaS com alguns dos benefícios elásticos de PaaS. Os aplicativos Web, que não abordamos muito nos capítulos anteriores, agora aparecem neste capítulo, fornecendo sua própria capacidade de escala com a demanda da aplicação.

Neste capítulo, examinaremos como projetar e criar aplicações que podem ser escaladas automaticamente. Veremos por que essa capacidade de escalar com a demanda ajuda você a executar aplicações eficientes e exploraremos diferentes maneiras de escalar com base em métricas diferentes.

9.1 Por que criar aplicações escalonáveis e confiáveis?

O que significa criar aplicações escaláveis? Ele permite que você cresça e acompanhe a demanda do cliente na medida em que o workload aumenta, mesmo quando você está no cinema em um fim de semana. Isso significa que você não ficar preso a uma fatura para muitos recursos extras que você não usa ou, talvez pior ainda, ter a

sua aplicação inativa devido à falta de recursos disponíveis. O ponto ideal para aplicações e os recursos de que necessitam raramente é estático. Normalmente, a aplicação exige refluxo e fluxo durante todo o dia e noite, ou entre dias úteis e fins de semana.

Há duas maneiras principais que você pode escalar recursos, como mostrado na Figura 9.1: verticalmente e horizontalmente. Os conjuntos de escala de máquina virtual e aplicativos Web podem escalar verticalmente ou horizontalmente.

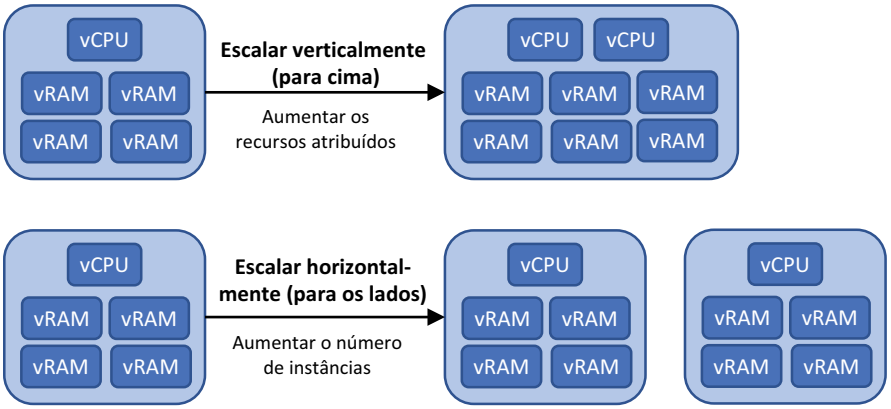


Figura 9.1 Você pode expandir e reduzir suas aplicações, verticalmente ou horizontalmente. O método que você usa depende de como sua aplicação é criada para lidar com a escala. A escala vertical ajusta os recursos atribuídos a uma VM ou a um aplicativo Web, como o número de núcleos de CPU ou a quantidade de memória. Esse método para dimensionar uma aplicação funciona bem se a aplicação executa apenas uma instância. A escala horizontal altera o número de instâncias que executam sua aplicação e ajuda a aumentar a disponibilidade e a resiliência.

Aplicações escaláveis têm um forte relacionamento com aplicações altamente disponíveis. Nos capítulos 7 e 8, passamos bastante tempo com conjuntos de disponibilidade e zonas de disponibilidade, e como configurar balanceadores de carga. Ambos os capítulos se concentram na necessidade de executar várias VMs. Quando suas aplicações podem ser escaladas automaticamente, a disponibilidade dessa aplicação também é expandida à medida que essas VMs são distribuídas entre conjuntos de disponibilidade ou zonas de disponibilidade. Tudo isso é uma coisa boa. O poder do Azure é que você não precisa se preocupar sobre como adicionar mais instâncias de aplicações, disseminá-las em todo o hardware do datacenter ou até mesmo datacenters e, em seguida, atualizar os recursos de rede para distribuir o tráfego para as novas instâncias da aplicação.

9.1.1 Escalar VMs verticalmente

A primeira maneira de escalar recursos é muitas vezes a melhor maneira que você pode ter usado no passado. Se sua aplicação começar a ser executada lentamente como mais clientes usando a aplicação, o que você faria normalmente? Aumentar a quantidade de vCPU ou memória, certo? Você *expande* o recurso em resposta à demanda.

Um dos usos mais comuns da escala vertical é para servidores de banco de dados. Bancos de dados são notoriamente famintos quando se trata de recursos de computação, ainda mais famintos do que seus clientes da pizzaria! Os servidores de banco de dados geralmente consomem todos os recursos fornecidos a uma VM, mesmo que não os usem imediatamente. Isso pode tornar difícil monitorar as demandas reais no sistema e saber quando você precisa escalar verticalmente e fornecer mais recursos. A Figura 9.2 mostra a resposta de escala vertical típica a um servidor de banco de dados que precisa de mais recursos.

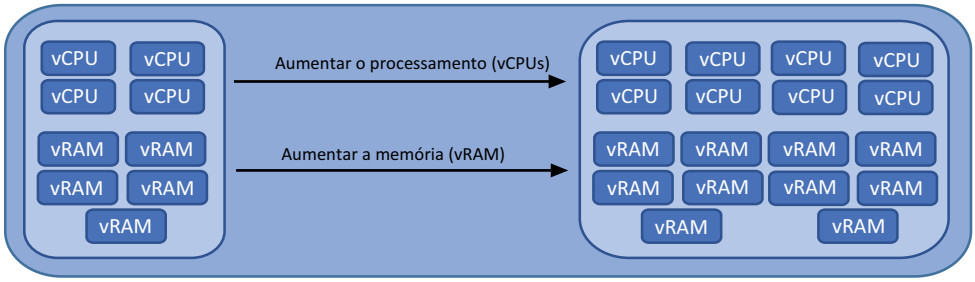


Figura 9.2 À medida em que um banco de dados aumenta, ele precisa de mais recursos para armazenar e processar os dados in-memory. Para dimensionar verticalmente nesse cenário, você adiciona mais CPU e memória.

Talvez você precise escalar além da demanda por CPU ou memória. E se você executar um site que serve um monte de imagens ou vídeo? Pode não haver muitos requisitos de processamento, mas as demandas de largura de banda podem ser altas. Para aumentar a largura de banda disponível, você pode aumentar o número de NICs em sua VM. E se você precisa armazenar mais imagens e vídeo, você adiciona mais armazenamento. Você pode adicionar ou remover recursos como NICs virtuais e armazenamento à medida que a VM continua a ser executada.

REDIMENSIONAR MÁQUINAS VIRTUAIS

No Azure, você pode aumentar o tamanho da VM (expandir) se precisar de mais recursos de computação para sua aplicação. No capítulo 2, você criou uma VM básica. Seu tamanho provavelmente era algo como Standard_D2s_v3. Esse nome não diz muito sobre os recursos de computação atribuídos a uma VM para determinar se você precisa aumentar a CPU ou a memória. Se você quiser escalar verticalmente, você precisa saber quais são suas opções.

Experimente agora

Acompanhe para ver os tamanhos de VM disponíveis e os recursos de computação:

1. Abra o portal do Azure em um navegador da Web e abra o Cloud Shell.
2. Insira o seguinte comando da CLI do Azure para listar os tamanhos de VM disponíveis e os recursos de computação que eles fornecem:

```
az vm list-sizes --location eastus --output table
```

A saída de `az vm list-sizes` varia de região para região e muda ao longo do tempo como o Azure ajusta suas famílias de VM. Aqui está um exemplo resumido da saída, exibindo `MemoryInMb` e `NumberOfCores` que cada tamanho de VM fornece:

```
MaxDataDiskCount    MemoryInMb    Nome                    NumberOfCores
------------------  ------------  ----------------------  ---------------
4                   8192          Standard_D2s_v3         2
8                   16384         Standard_D4s_v3         4
16                  32768         Standard_D8s_v3         8
32                  65536         Standard_D16s_v3        16
8                   4096          Standard_F2s_v2         2
16                  8192          Standard_F4s_v2         4
32                  16384         Standard_F8s_v2         8
2                   2048          Standard_B1ms           1
2                   1024          Standard_B1s            1
4                   8192          Standard_B2ms           2
4                   4096          Standard_B2s            2
```

Portanto, sua VM Standard_D2s_v3 fornece dois núcleos de CPU e 8 GB de memória, que é mais do que suficiente para uma VM básica que executa um servidor Web. Vamos supor que sua pizzaria online começa a receber alguns pedidos, e você quer escalar verticalmente. Você pode usar `az vm resize` para escolher outro tamanho. Você especifica o tamanho da VM que tem o número de núcleos de CPU e a memória que sua aplicação precisa.

A CPU e a memória adicionais não aparecem magicamente na VM. Esse comportamento pode ser um pouco diferente do que você encontra com o Hyper-V ou VMware em um mundo na infraestrutura local. Com razão, você pode adicionar ou remover recursos de computação de núcleo em um ambiente na infraestrutura local à medida que a VM continua a ser executada. No Azure, uma reinicialização geralmente é necessária quando você faz o redimensionamento de uma VM para registrar os novos recursos de computação e acionar as regras de cobrança apropriadas. Quando você deseja escalar verticalmente, planeje algum tempo de inatividade à medida que a VM é reinicializada.

REDUZIR

E se você tiver uma VM com mais recursos do que ela precisa? Esse cenário geralmente é mais comum do que uma VM que tem menos recursos do que o necessário. Os proprietários de aplicações podem escolher um tamanho de VM maior do que o necessário para ter certeza de que sua aplicação seja executada sem problemas. Todos os recursos desperdiçados custam dinheiro, e é fácil os custos passarem despercebidos até a fatura chegar no final do mês.

A capacidade de escalar recursos funciona em ambas as direções. Nós nos concentramos em como *expandir* recursos, mas todos os mesmos conceitos funcionam para *reduzir* recursos. É importante identificar os tamanhos de VM em uso e a quantidade de uma demanda que as aplicações fazem desses recursos. Você pode usar `az vm resize` para escolher um tamanho de VM com menos núcleos de CPU e memória. Novamente, uma reinicialização VM é necessária no momento para qualquer operação de redimensionamento.

9.1.2 Escalar aplicativos Web verticalmente

Aplicativos Web podem ser expandidos ou reduzidos com base nas necessidades de recursos, da mesma forma que as VMs fazem. Quando você criou um aplicativo Web no capítulo 3, o tamanho padrão S1 Standard forneceu um núcleo de CPU e 1,75 GB de

RAM. Cada camada e tamanho do aplicativo Web fornece uma quantidade definida de recursos, como núcleos de CPU, memória e slots de preparação. Mesmo se o tamanho padrão ou a alocação de recursos for alterado ou se você escolher um tamanho de aplicativo Web diferente, o conceito continua o mesmo.

Se criar seu aplicativo Web e descobrir que a aplicação exige mais recursos do que o plano de serviço fornece, você poderá alterar para uma camada diferente, como mostrado na Figura 9.3. O mesmo processo funciona se você tiver mais recursos do que precisa. Seu aplicativo Web pode ser expandido ou reduzido manualmente dessa maneira, conforme necessário.

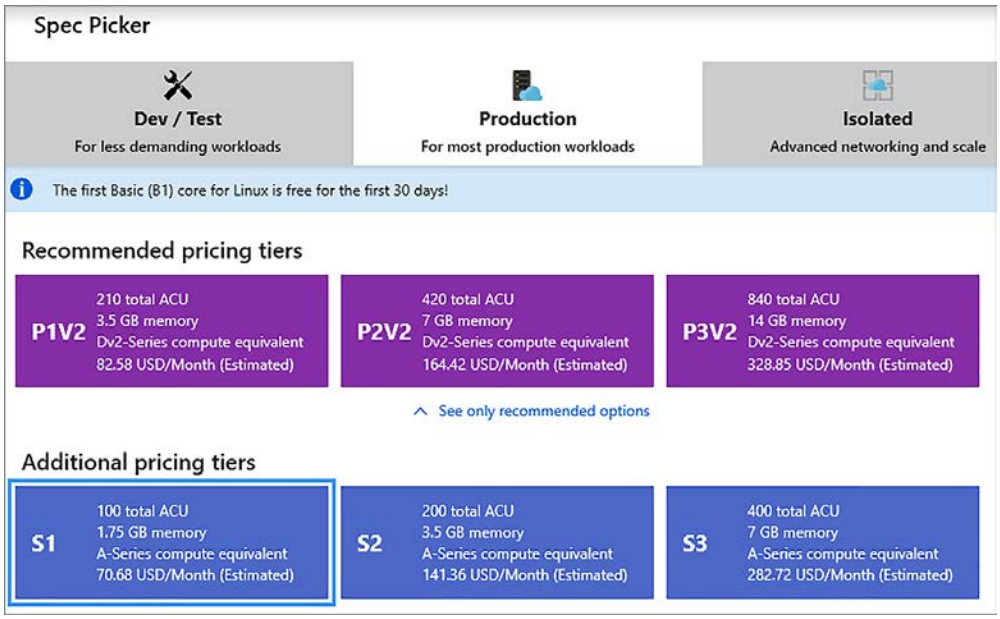


Figura 9.3 Para escalar um aplicativo Web verticalmente de forma manual, altere o tipo de preço (tamanho) do plano de serviço de aplicativo subjacente. O plano de serviço de aplicativo define a quantidade de recursos atribuídos ao seu aplicativo Web. Se sua aplicação exigir uma quantidade diferente de armazenamento, número de CPUs ou slots de implantação, você poderá alterar para um tipo diferente para escalar corretamente os recursos atribuídos à demanda da aplicação.

9.1.3 Escalar recursos horizontalmente

Uma abordagem diferente para acompanhar a demanda é escalar horizontalmente. Para escalar verticalmente, você aumenta a quantidade de CPU e memória atribuída a um único recurso, como uma VM. Para escalar horizontalmente, aumente o número de VMs, como mostrado na Figura 9.4.

Para escalar horizontalmente, sua aplicação precisa estar ciente dessa capacidade e ser capaz de processar dados sem conflitos. Um aplicativo Web é uma ótima candidata para escalar horizontalmente, porque a aplicação normalmente pode processar dados por si só.

À medida que você constrói aplicações mais complexas, você pode quebrar uma aplicação em componentes individuais menores. Se você pensar nas filas de armazenamento do Azure no capítulo 4, talvez

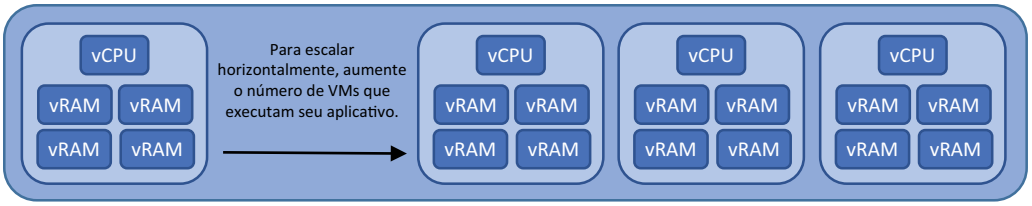


Figura 9.4 Para lidar com um aumento na demanda em sua aplicação, você pode aumentar o número deVMs que executam a aplicação, distribuindo a carga em várias VMs, em vez de VMs de instância única cada vez maiores.

tenha um componente de aplicação que receba os pedidos do front-end da Web e outro componente de aplicação que processe esses pedidos e os transmita para a pizzaria. O uso de filas de mensagens é uma abordagem para a criação e a gravação de aplicações que podem ser operadas em um ambiente que pode ser escalado horizontalmente. Essa abordagem também permite dimensionar cada componente de aplicação separadamente e usar diferentes tamanhos de VM ou planos de aplicativo Web para maximizar a eficiência e reduzir sua fatura mensal.

Historicamente, você escalaria verticalmente porque seria mais fácil colocar mais recursos de computação em uma aplicação e esperar que ficasse satisfeito. Configurar um cluster de recursos e escalar uma aplicação horizontalmente era muitas vezes complexo no mundo físico. Com a computação em nuvem e a virtualização, os desafios da escalabilidade horizontal são minimizados até o ponto em que muitas vezes você pode escalar horizontalmente mais rapidamente do que verticalmente e sem tempo de inatividade.

Lembre-se do comando `az vm resize` mencionado anteriormente neste capítulo. O que acontece quando a operação de redimensionamento da VM é concluída? A VM é reiniciada. Se essa é a única instância da sua aplicação, ninguém pode acessá-lo até que ele volte a ficar online. Quando você escala horizontalmente, não há tempo de inatividade quando você adiciona instâncias de VM. Quando as novas VMs estão prontas, elas começam a processar algumas das solicitações da aplicação. Os testes de integridade do balanceador de carga (capítulo 8) detectam automaticamente quando uma nova VM no pool de back-end está pronta para processar solicitações de clientes e o tráfego começa a ser distribuído a ele.

O Azure foi projetado para oferecer flexibilidade e escolha quando se trata de como você escala. Se você estiver projetando um novo ambiente de aplicação, sugiro implementar uma abordagem de escala horizontal. As VMs têm um primo legal no Azure que pode ajudá-lo aqui: conjuntos de escala de máquinas virtuais.

9.2 Conjuntos de escala de máquinas virtuais

VMs são alguns dos workloads mais comuns no Azure, por um bom motivo. A curva de aprendizado para criar e executar uma VM é superficial, porque a maior parte do que você já sabe, o que pode ser transferido diretamente para o Azure. Os servidores Web estão entre os workloads mais comuns para uma VM, o que é conveniente para que você não tenha que aprender novas habilidades para transferir seu conhecimento de execução do Apache, do IIS ou do NGINX em uma VM do Azure.

Que tal um cluster de VMs que executa um servidor Web? Como lidar com isso em seu ambiente regular na infraestrutura local? Há muitas soluções de cluster possíveis, para começar. E quanto a atualizações para seus servidores físicos ou VMs? Como você

lidaria com isso? E se você quiser aumentar ou diminuir automaticamente o número de instâncias no cluster? Você precisa usar outra ferramenta? A Figura 9.5 mostra uma descrição de um conjunto de escala de máquinas virtuais.

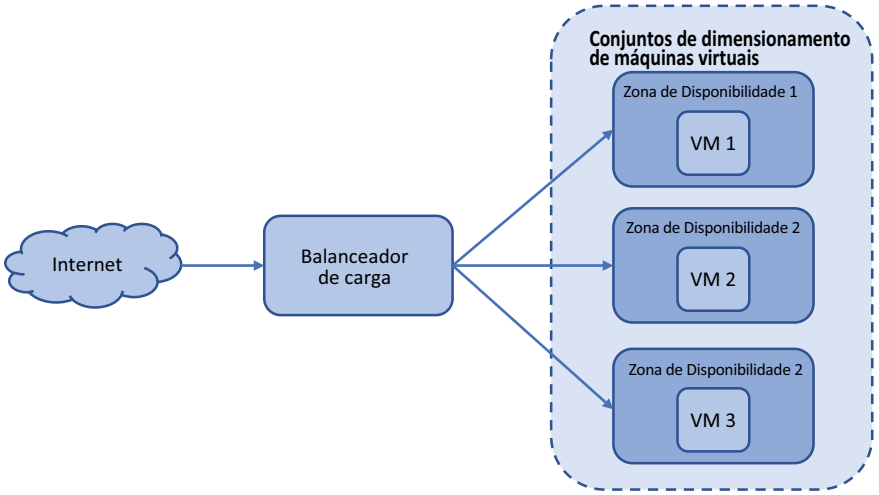


Figura 9.5 Um conjunto de escala de máquinas virtuais definida agrupa logicamente um conjunto de VMs. As VMs são idênticas e podem ser gerenciadas, atualizadas e escaladas de forma central. É possível definir métricas que aumentem ou diminuam automaticamente o número de VMs no conjunto de escalas com base na carga da aplicação.

Um conjunto de escalas simplifica a forma como você executa e gerencia várias VMs para fornecer uma aplicação altamente disponível e com balanceamento de carga. Você indica ao Azure qual tamanho VM deve ser usado, uma imagem base para a VM e quantas instâncias você deseja. Em seguida, você pode definir métricas de CPU ou memória para aumentar ou diminuir automaticamente o número de instâncias em resposta à carga da aplicação ou em uma agenda em horários de pico do cliente. Os conjuntos de escala combinam o modelo de IaaS de VMs com o poder dos recursos de PaaS, como escala, redundância, automação e gerenciamento centralizado de recursos.

Um único conjunto de escala de VM?

Se você criar aplicações em VMs, planeje iniciar com um conjunto de escala, mesmo que você precise apenas de uma VM. Por quê? Um conjunto de escala pode expandir a qualquer momento e cria automaticamente as conexões com um balanceador de carga ou um gateway de aplicação. Se a demanda para a aplicação aumenta repentinamente em dois meses, você pode informar a escala definida para criar uma ou duas instâncias de VM adicionais.

Para expandir uma VM normal e autônoma, você precisa adicionar essa VM a um balanceador de carga. E se você não começou com a VM em um conjunto de disponibilidade ou zona de disponibilidade, deve planejar como fazer essas VMs altamente disponíveis. Ao criar uma escala definida para começar, mesmo para uma VM, você prepara sua aplicação para o futuro com o mínimo de trabalho adicional necessário.

9.2.1 Criar um conjunto de escala de máquinas virtuais

Embora um conjunto de escala simplifique a criação e execução de aplicações altamente disponíveis, é necessário criar e configurar alguns novos componentes. Dito isso, você pode reduzir o processo para dois comandos para implantar uma escala definida com a CLI do Azure.

Experimente agora

Para criar uma escala definida com a CLI do Azure, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione o ícone do Cloud Shell na parte superior do painel.
2. Crie um grupo de recursos com `az group create`; especifique o nome de um grupo de recursos, como `azuremolchapter9`, e um local:

   ```
   az group create --name azuremolchapter9 --location westeurope
   ```

   Os conjuntos de escala podem usar zonas de disponibilidade, portanto, selecione uma região onde o suporte está disponível.
3. Para criar um conjunto de escala, especifique o número de instâncias de VM desejadas e como as instâncias de VM devem lidar com as atualizações de sua configuração. Quando você faz uma alteração nas VMs, como instalar uma aplicação ou aplicar atualizações do sistema operacional convidado, as VMs podem atualizar automaticamente assim que detectarem a alteração. Ou você pode definir a política de atualização como manual e aplicar as atualizações em um momento adequado de sua escolha. O restante dos parâmetros deve ser conhecido de quando você criou uma única VM:

   ```
   az vmss create \
     --resource-group azuremolchapter9 \
     --name scalesetmol \
     --image UbuntuLTS \
     --admin-username azuremol \
     --generate-ssh-keys \
     --instance-count 2 \
     --vm-sku Standard_B1ms \
     --upgrade-policy-mode automatic \
     --lb-sku standard \
     --zones 1 2 3
   ```

Pronto! Você criou várias VMs em uma zona de disponibilidade que pode ser escalada. Prepare-se para o que é realmente legal sobre o conjunto de escala que você acabou de criar com a CLI do Azure. Lembre-se de todo o capítulo inteiro sobre balanceadores de carga (capítulo 8), e todos os comandos da CLI que você teve que usar e como modelos podem simplificar como você cria um balanceador de carga. Aquele comando `az vmss create` criou e configurou um balanceador de carga para você.

Lembre-se de seus limites de cota

Eu mencionei esse problema de cota no capítulo 7, mas vale a pena repetir caso você encontre problemas. No Azure, as cotas padrão em sua assinatura impedem a implantação acidental de recursos e o esquecimento deles, o que lhe custará dinheiro. Você pode ver a lista de cotas em http://mng.bz/ddcx.

Ao criar várias VMs, você pode encontrar problemas de cota. Você também pode ter problemas se não excluir recursos de capítulos e exercícios anteriores. Se você vir texto de erro ao longo das linhas de

```
A operação resulta em exceder os limites de cota do núcleo.
Máximo permitido: 4, Atualmente em uso: 4, Pedido adicional: 2.
```

é uma boa indicação de que você precisa solicitar um aumento em suas cotas. Você pode exibir sua cota atual para uma determinada região da seguinte maneira:

```
az vm list-usage --location westeurope
```

Para solicitar um aumento em suas cotas para uma região, siga as etapas descritas em http://mng.bz/Xq2f.

O Azure CLI ajuda a criar um conjunto de escalas com prompts mínimos. Um balanceador de carga foi criado e configurado, um endereço IP público atribuído e as instâncias de VM do conjunto de escala adicionadas ao pool IP de back-end.

Experimente agora

Confira os recursos criados com o seu conjunto de escala, conforme descrito nos comandos a seguir.

Para ver quais recursos foram criados com seu conjunto de escala, execute o seguinte comando:

```
az resource list \
  --resource-group azuremolchapter9 \
  --output table
```

A saída é semelhante ao seguinte exemplo. Veja a coluna Tipo para comprovar que uma rede virtual, um endereço IP público e um balanceador de carga foram criados:

```
Nome     ResourceGroup     Tipo
-------  ----------------  ------------------------------------
mol      azuremolchapter9  Microsoft.Compute/virtualMachineScaleSets
molLB    azuremolchapter9  Microsoft.Network/loadBalancers
molLBIP  azuremolchapter9  Microsoft.Network/publicIPAddresses
molVNET  azuremolchapter9  Microsoft.Network/virtualNetworks
```

O que significa toda esta magia? Quando você cria uma escala definida com o Azure CLI, um balanceador de carga com redundância de zona e um endereço IP público são criados para você. As VMs são criadas e adicionadas a um pool de IPS de back-end no balanceador de carga. As regras NAT são criadas que permitem que você se conecte às instâncias de VM. A única coisa que falta são as regras do balanceador de carga, pois elas variam de acordo com as aplicações que você deseja executar. À medida que você adiciona ou remove VMs do conjunto de escala, a configuração do balanceador de

carga é atualizada automaticamente para permitir que o tráfego seja distribuído às novas instâncias. Essa mágica não está limitada à CLI do Azure: se você usar o Azure PowerShell ou o portal do Azure, esses recursos de rede de suporte são criados e conectados para funcionar juntos.

Experimente agora

Sua escala foi criada com duas instâncias. Você pode escalar manualmente o número de instâncias de VM em seu conjunto de escala. Quando você fizer isso, o balanceador de carga atualiza automaticamente a configuração de pool de IP de back-end. Defina `--new-capacity` do conjunto de escala para quatro instâncias da seguinte maneira:

```
az vmss scale \
    --resource-group azuremolchapter9 \
    --name scalesetmol \
    --new-capacity 4
```

9.2.2 Criar regras de dimensionamento automático

Ao criar seu conjunto de escala, você implantou um número fixo de instâncias. Um dos mais importantes recursos dos conjuntos de escala é a capacidade de expandir e reduzir automaticamente ou não o número de instâncias de VM que o conjunto de escala executa.

Como mostrado na Figura 9.6, o número de instâncias em um conjunto de escala pode aumentar automaticamente à medida que a carga da aplicação aumenta. Considere uma aplicação de negócios típica em seu ambiente. No início do dia de trabalho, os usuários começam a acessar a aplicação, o que faz com que a carga de recurso nessas instâncias de VM aumente. Para garantir o melhor performance da aplicação, o conjunto de escala adiciona automaticamente mais instâncias de VM. O balanceador de carga começa a distribuir o tráfego às novas instâncias automaticamente. Mais tarde, à medida em que os usuários vão para casa, a demanda da aplicação diminui. As instâncias de VM usam menos recursos e, portanto, o conjunto de escala remove automaticamente algumas instâncias de VM para reduzir recursos desnecessários e custo mais baixo.

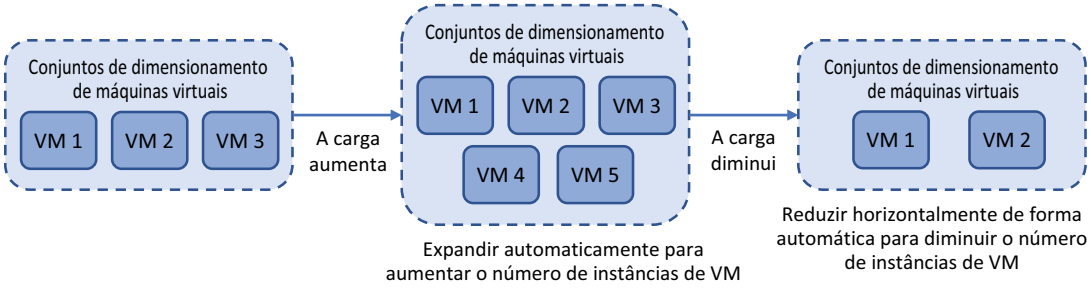


Figura 9.6 Os conjuntos de escala podem ser automaticamente expandidos ou reduzidos. Você define regras para monitorar determinadas métricas que acionam as regras para aumentar ou diminuir o número de instâncias de VM executadas. À medida que a demanda da aplicação for alterada, o número de instâncias de VM também será. Essa abordagem maximiza a performance e a disponibilidade da sua aplicação, minimizando também o custo desnecessário quando a carga da aplicação diminui.

Você pode basear suas regras de conjunto de escalas em várias métricas. Você pode examinar as métricas de host para o consumo de recursos básicos, configurar a coleção de métricas de VM no convidado para análise de contadores específicos de performance da aplicação ou usar o Azure Application Insights para monitoramento dentro do código da aplicação.

Você também pode usar agendas para definir um determinado número de instâncias de VM em um conjunto de escala para uma janela de tempo. No exemplo de uma aplicação de negócios comum para a qual a demanda é maior durante o horário comercial do que à noite, talvez você queira definir um número fixo maior de instâncias a serem executadas durante o horário comercial e definir um menor número de instâncias a serem executadas à noite.

As regras de dimensionamento automático com base em métricas monitoram o performance em um intervalo de tempo definido, como cinco minutos, e podem levar mais alguns minutos para gerar as novas instâncias de VM e configurá-las para uso da aplicação. Se você usar agendas fixas para escalar automaticamente o número de instâncias de VM em seu conjunto de escala, esses recursos adicionais já estarão em uso e o balanceador de carga distribuirá tráfego para eles durante o dia.

O uso de agendas requer uma linha de base para a demanda de aplicação típica e não conta para maior ou menor demanda em determinadas partes da conta comercial ou ciclo de vendas. Às vezes, você pode terminar acima com mais recursos do que é necessário, assim que você paga mais do que necessário. E você pode ter situações em que a carga da aplicação é maior do que o número de instâncias de VM no conjunto de escala pode fornecer.

Experimente agora

Para criar regras de dimensionamento automático para um conjunto de escala, conclua as etapas a seguir.

1. Navegue e selecione Grupo de recursos na barra de navegação à esquerda no portal do Azure.
2. Escolha o grupo de recursos que você criou para sua implantação de modelo, como `azuremolchapter9`.
3. Selecione seu conjunto de escala na lista de recursos, como scalesetmol.
4. Em Configurações à esquerda na janela Conjunto de escala, escolha Escala. Você pode escalar manualmente ou criar suas próprias regras personalizadas de dimensionamento automático.
5. Escolha criar regras de dimensionamento automático personalizadas.
6. Insira um nome, como `autoscale`, e, em seguida, defina uma contagem de instância mínima, máxima e padrão. Para este exercício, defina o mínimo para 2, máximo para 10 e padrão para 2.
7. Escolha adicionar uma regra e, em seguida, examine as configurações de regra disponíveis, como mostrado na Figura 9.7.

   Os parâmetros padrão olham para o consumo médio da CPU. A regra é acionada quando a carga é maior que 70% em um intervalo de 10 minutos. O conjunto de escala é expandido por uma instância de VM e as regras, em seguida, aguarde 5 minutos antes de começar a monitorar e pode acionar a próxima regra.

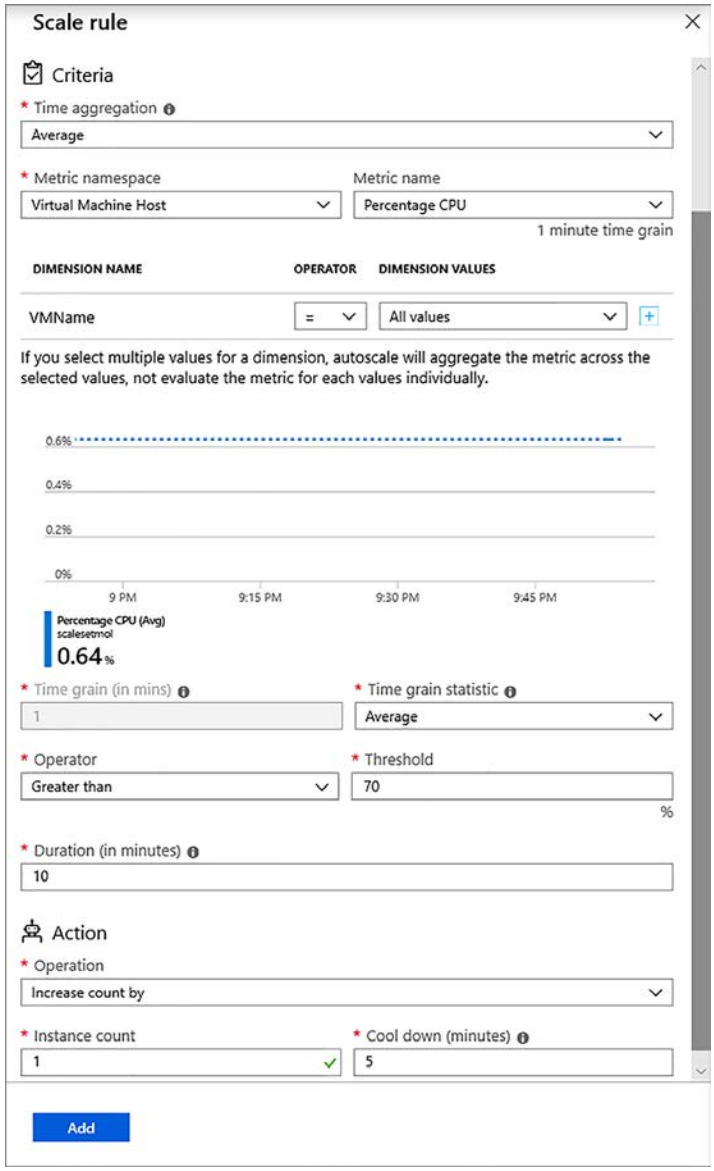


Figura 9.7 Ao adicionar uma regra de dimensionamento automático, você define o comportamento exato necessário para que a regra seja acionada.

Esse período de esfriamento oferece tempo às novas instâncias de VM para implantar e começar a receber o tráfego do balanceador de carga, que deve diminuir a carga geral da aplicação no conjunto de escala. Sem esse período de esfriamento, as regras podem acionar outra instância de VM a ser adicionada antes que a carga comece a ser distribuída entre a instância de VM criada anteriormente.

8 Para criar a regra, selecione Adicionar.

9 Escolha adicionar outra regra. Dessa vez, configure a regra para diminuir contagem por um quando a carga média da CPU for menor que 30% em uma duração de 5 minutos.

10 Revise suas regras, como mostrado na Figura 9.8, e selecione Salvar.

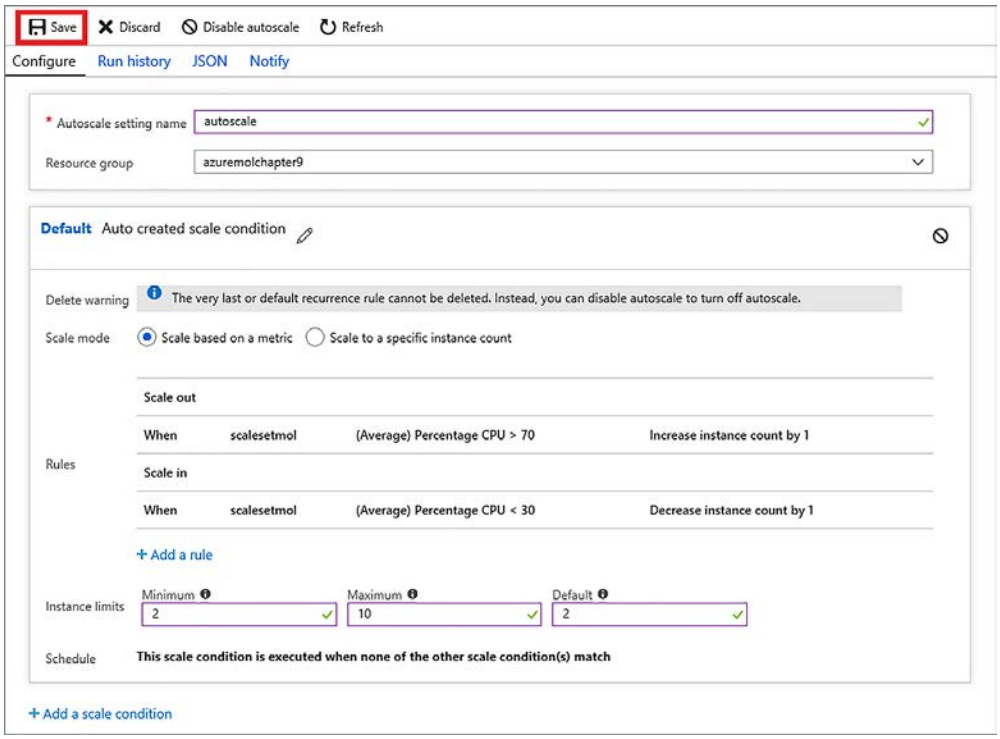


Figura 9.8 Você deve ter uma regra que aumenta a contagem de instância por 1 quando a carga média da CPU é maior que 70% e outra regra que diminui a contagem de instâncias por um quando a carga média da CPU é menor que 30%.

Você também pode configurar regras de dimensionamento automático como CLI do Azure, o Azure PowerShell ou modelos. O portal fornece uma forma visual agradável para rever as regras e ver as opções disponíveis para cada parâmetro. À medida que você cria regras mais complexas, os modelos fornecem uma maneira de criar conjuntos de escala com o mesmo conjunto de regras de forma reproduzível.

9.3 Escalar um aplicativo Web

Se você estava super interessado em aplicativos Web no capítulo 3 ou em tabelas e filas do Azure no capítulo 4, esses três últimos capítulos, que se concentraram muito em VMs de IaaS, podem ter deixado você com algumas dúvidas. A nuvem não devia ser mais fácil do que isso? Para componentes PaaS como aplicativos Web, sim.

Eu não quero que você pense que vamos nos apressar nas próximas páginas sobre como fornecer a mesma alta disponibilidade e capacidades de dimensionamento automático para aplicativos Web. A verdade é que é muito mais fácil de fazer. Como a maioria das coisas, a escolha entre IaaS e PaaS é um equilíbrio entre flexibilidade e facilidade de gerenciamento. Grande parte da redundância subjacente é abstraída em serviços de PaaS como aplicativos Web e, portanto, você não precisa de um capítulo inteiro sobre alta disponibilidade e outro capítulo sobre balanceadores de carga.

O caminho IaaS para criar e executar suas próprias VMs ou conjuntos de escalas com balanceadores de carga e zonas de disponibilidade pode vir de uma necessidade ou restrição de negócios. Desenvolvedores, engenheiros de operações, ou ferramentas e fluxos de trabalho podem não estar prontos para prover tudo para aplicativos Web. Dito isto, recomendo muito que você pense em aplicativos Web para novas implantações de aplicações. O uso de componentes de PaaS como aplicativos Web oferece a você mais tempo para se concentrar nas aplicações e nos seus clientes, em vez de infraestrutura e administração.

Experimente agora

Para criar um aplicativo Web com a CLI do Azure, conclua as etapas a seguir:

1. No capítulo 3, você criou um aplicativo Web no portal do Azure. Como na maioria dos recursos, geralmente é mais rápido e fácil usar a CLI do Azure. Abra o Cloud Shell no portal do Azure.
2. Crie um plano de serviço de aplicativo que é um tamanho Standard S1. Esse tamanho permite escalar automaticamente até 10 instâncias do seu aplicativo Web:

```
az appservice plan create \
  --name appservicemol \
  --resource-group azuremolchapter9 \
  --sku s1
```

3. Crie um aplicativo Web que use um repositório Git local para implantação, como você fez no capítulo 3:

```
az webapp create \
  --name webappmol \
  --resource-group azuremolchapter9 \
  --plan appservicemol \
  --deployment-local-git
```

Todos os conceitos e cenários para regras de dimensionamento automático e agendas para conjuntos de escala discutidos na seção 9.2.2 também aplicam-se a aplicativos Web. Como uma recapitulação rápida, aqui estão alguns cenários comuns para aplicativos Web de dimensionamento automático:

- Aumente ou diminua automaticamente o número de instâncias do aplicativo Web com base nas métricas de performance, para oferecer suporte à demanda da aplicação durante todo o dia de trabalho.
- Agendar um aplicativo Web para aumentar automaticamente o número de instâncias no início do dia de trabalho e, em seguida, diminuir o número de instâncias no final do dia de trabalho.

No caso da pizzaria, o aplicativo Web pode receber mais tráfego mais tarde no dia e durante a noite, por isso, não há um conjunto de regras de dimensionamento automático que se aplica a todas as situações. Novamente, você precisa de linha de base do performance da aplicação para entender como ela é executada em uso normal e a métrica de performance em que a aplicação precisa ser expandido ou reduzido. Mesmo quando usar agendas de dimensionamento automático, você deve continuar monitorando e rastreando quando suas demandas de pico da aplicação são para criar regras que oferecem suporte a esse padrão de uso.

Experimente agora

Para criar regras de dimensionamento automático para um aplicativo Web, conclua as etapas a seguir:

1. Navegue e selecione Grupo de recursos na barra de navegação à esquerda no portal do Azure.
2. Escolha o grupo de recursos que você criou para seu aplicativo Web, como azuremolchapter9.
3. Selecione seu aplicativo Web na lista de recursos, como webappmol.
4. Em Configurações à esquerda na janela do aplicativo Web, escolha Expandir (Plano de Serviço de Aplicação).
5. Novamente, opte por configurar regras de dimensionamento automático personalizadas, e não apenas escalar manualmente o aplicativo Web.
6. Insira um nome, como `autoscalewebapp`, e, em seguida, defina uma contagem de instância mínima, máxima e padrão. Para este exercício, defina o mínimo para 2, máximo para 5 e padrão para 2.
7. Escolha adicionar uma regra e, em seguida, examine as configurações de regra disponíveis. Esta janela tem a mesma aparência que as regras de dimensionamento automático para conjuntos de escala. Os parâmetros padrão examinam o consumo médio da CPU e são acionados quando a carga é maior que 70% em um intervalo de 10 minutos. O aplicativo Web é expandido por uma instância de VM e as regras, aguarde 5 minutos antes de começar a monitorar e pode acionar a próxima regra.
8. Escolha adicionar outra regra. Dessa vez, configure a regra para diminuir contagem por um quando a carga média da CPU for menor que 30% em uma duração de 5 minutos.
9. Revise e salve suas regras.

Quando as regras de dimensionamento automático acionam o aplicativo Web para expandir ou reduzir, a plataforma do Azure atualiza a distribuição de tráfego às instâncias de aplicativo Web disponíveis. Não há um balanceador de carga exposto a você, pois você tem com os conjuntos de escala, mas o tráfego ainda é distribuído automaticamente entre as instâncias do aplicativo Web à medida que seu ambiente é expandido ou reduzido. O conceito é semelhante, apenas abstraído para que você possa aproveitar a abordagem PaaS e não se preocupe tanto.

Os conjuntos de escala e os Aplicativos Web fornecem uma maneira de criar regras que escalam automaticamente o número de instâncias que executam suas aplicações. Com várias instâncias para executar sua aplicação, você também aumenta a disponibilidade do seu aplicativo. Os conjuntos de escala são um bom meio-termo entre desenvolvedores e tomadores de decisão de negócios que desejam ou precisam criar aplicações em VMS, enquanto usam os recursos do tipo PaaS para dimensionamento automático e reconfigurar o fluxo de tráfego do cliente.

No capítulo 11, analisaremos o Gerenciador de Tráfego do Azure, que completa essas implantações de alta disponibilidade. Agora, você ainda não está totalmente pronto para a produção em termos de ser capaz de oferecer vários conjuntos de escala com redundância ou instâncias de aplicativos Web com o tráfego distribuído automaticamente entre elas. Chegaremos nesse assunto mais tarde.

9.4 Laboratório: Instalar aplicações no seu conjunto de escala ou aplicativo Web

Cobrimos muito conteúdo neste capítulo, então agora você pode escolher um laboratório final rápido para qualquer conjunto de escala ou aplicativo Web. Ou se você quiser prolongar a sua pausa para o almoço, faça os dois.

9.4.1 Conjuntos de escala de máquinas virtuais

Você tem várias instâncias de VM em seus conjuntos de escala, mas eles não fazem muita coisa agora. Para obter uma visão geral das diferentes maneiras de instalar aplicações em instâncias de VM em um conjunto de escala, consulte http://mng.bz/9Ocx. Na prática, você usaria um desses métodos de implantação automatizados, mas, por enquanto, instale manualmente um servidor Web nas instâncias de VM, como você fez no capítulo 8:

1. Lembre-se das regras NAT do balanceador de carga. Por padrão, cada instância de VM em um conjunto de escala tem uma regra NAT que permite que você use SSH diretamente nele. As portas não estão na porta TCP padrão 22. Visualize a lista de instâncias de VM em um conjunto de escala e seus números de porta da seguinte maneira:

```
az vmss list-instance-connection-info \
  --resource-group azuremolchapter9 \
  --name scalesetmol
```

2. Para se conectar a uma porta específica via SSH, use o parâmetro `-p` da seguinte maneira (forneça seu próprio endereço IP público e números de porta):

```
ssh azuremol@40.114.3.147 -p 50003
```

3. Instale um servidor Web NGINX básico em cada instância de VM com `apt install`. Pense de novo em como você fez isso no capítulo 8.
4. Para ver a escala definida em ação, abra o endereço IP público do balanceador de carga do conjunto de escala em um navegador da Web.
5. Se você encontrar problemas, verifique se o balanceador de carga criou corretamente uma regra de balanceador de carga para porta 80 TCP e tem uma sonda de integridade associada para a porta 80 TCP ou sua própria sonda de integridade HTTP personalizada que busca /health.html na VM.

9.4.2 Aplicativos Web

Para implantar sua aplicação em um aplicativo Web que executa várias instâncias, o processo é o mesmo que o aplicativo Web único do capítulo 3. Você pressiona a aplicação para o repositório Git local para o aplicativo Web e, graças ao poder de PaaS, a plataforma do Azure implanta essa única base de código em várias instâncias do aplicativo Web:

1. Inicialize um repositório Git em azure-mol-samples-2nd-ed/09 e, em seguida, adicione e confirme os arquivos de exemplo, como você fez no capítulo 3:

```
cd azure-mol-samples-2nd-ed/09
git init && git add . && git commit -m "Pizza"
```

2. Seu aplicativo Web tem um repositório Git local. Adicione um remoto para seu aplicativo Web da mesma maneira que você fez no capítulo 3:

```
git remote add webappmolscale <your-git-clone-url>
```

3. Envie este exemplo para seu aplicativo Web. Isso faz com que um único código seja enviado, mas seu aplicativo é distribuído a várias instâncias do aplicativo Web:

```
git push webappmolscale master
```

10 Bancos de dados globais com Cosmos DB

Dados. Você não pode ficar sem eles. Quase todas as aplicações que você constrói e executa cria, processa ou recupera dados. Tradicionalmente, esses dados ficavam armazenados em um banco de dados estruturado, como MySQL, Microsoft SQL ou PostgreSQL. Esses grandes bancos de dados estruturados são estabelecidos e bem conhecidos, têm ampla documentação e tutoriais, e podem ser acessados da maioria das principais linguagens de programação.

Com grande poder vem grande responsabilidade, e grande parte da sobrecarga de infraestrutura e gerenciamento normalmente acompanha esses bancos de dados estruturados tradicionais. Isso não quer dizer que você não deve usá-los, longe disso. Porém, quando se trata de aplicações que são executadas em uma escala global, não é tarefa fácil também desenvolver clusters de servidores de banco de dados que replicam seus dados e encaminham de forma inteligente os clientes para a sua instância mais próxima.

É aí que o Azure Cosmos DB torna-se seu melhor amigo. Você não precisa se preocupar sobre como replicar seus dados, garantir a consistência e distribuir solicitações de clientes. Em vez disso, você adiciona dados em um dos muitos modelos disponíveis e, em seguida, escolhe onde deseja que seus dados estejam disponíveis. Neste capítulo, você aprenderá sobre modelos de banco de dados não estruturados no cosmos DB, como criar e configurar seu banco de dados para distribuição global e como criar aplicativos Web que usam sua instância do Cosmos DB altamente escalável e com redundância.

10.1 O que é o Cosmos DB?

O capítulo 4 começou a explorar bancos de dados não estruturados com tabelas de armazenamento do Azure. O exemplo era básico, mas os conceitos são os fundamentos do Cosmos DB. Primeiro, vamos voltar um pouco e examinar o que queremos dizer com bancos de dados *estruturados* e *não estruturados*.

10.1.1 Bancos de dados estruturados (SQL)

Banco de dados estruturados são a abordagem mais tradicional para armazenamento. Uma *estrutura*, ou *esquema*, para o banco de dados define como os dados são representados. Os dados são armazenados em tabelas, com cada linha representando um item e um conjunto fixo de valores atribuídos a ele. Se tomarmos o modelo da pizzaria, cada fileira em uma tabela que armazene os tipos de pizza pode indicar o nome da pizza, seu tamanho, e o preço. Um banco de dados SQL básico é mostrado na Figura 10.1.

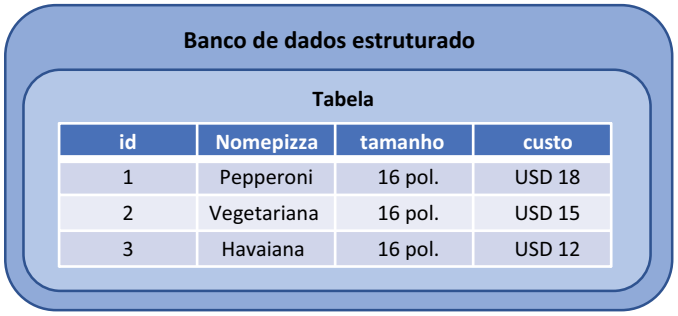
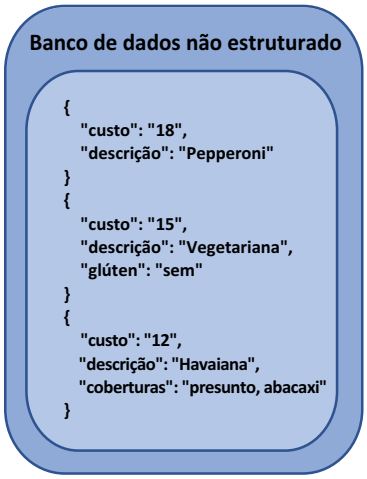


Figura 10.1 Em um banco de dados estruturado, os dados são armazenados em linhas e colunas dentro de uma tabela. Cada linha contém um conjunto fixo de colunas que representam o esquema para o banco de dados.

Em bancos de dados estruturados, cada servidor normalmente deve conter todo o banco de dados para que as consultas e a recuperação de dados sejam bem-sucedidas. Os dados são unidos em consultas para extrair de tabelas diferentes com base em critérios que o desenvolvedor cria como parte da consulta estruturada. Esta é origem do nome *Structured Query Language* (SQL). À medida que os bancos de dados crescem em tamanho e complexidade, os servidores que executam o banco de dados devem ser suficientemente escalados para manipular esses dados in-memory. Isso se torna difícil, e dispendioso, com conjuntos de dados muito grandes. Considerando que eles precisam de uma estrutura, também dificulta a adição de propriedades e a alteração da estrutura mais tarde.

10.1.2 Bancos de dados não estruturados (NoSQL)



Os dados não estruturados em bancos NoSQL não são armazenados em tabelas de linhas e colunas. Em vez disso, eles são armazenados em matrizes dinâmicas que permitem que você adicione novas propriedades para um item, conforme necessário. Uma grande vantagem dessa abordagem é que você pode adicionar rapidamente um novo tipo de pizza ou cobertura sem alterar a estrutura subjacente do banco de dados. Em um banco de dados estruturado, você precisaria adicionar uma nova coluna a uma tabela e, em seguida, atualizar a aplicação para manipular a coluna adicional. Em bancos de dados NoSQL, você adiciona outra propriedade a uma determinada entrada do seu código. Consulte a Figura 10.2.

Figura 10.2 Em um banco de dados não estruturado, os dados são armazenados sem mapeamentos fixos de colunas para uma linha em uma tabela. Você pode adicionar coberturas a uma única pizza, por exemplo, sem atualizar todo o esquema e outros registros.

Os bancos de dados NoSQL também oferecem diferentes modelos de banco. Esses modelos fornecem uma indicação de como os dados são armazenados e recuperados no banco de dados. O modelo usado varia de acordo com o tamanho e o formato dos dados com os quais você trabalha e como você precisa representar os dados em sua aplicação. Esses modelos incluem documento, gráfico e tabela. Não se prenda muito aos detalhes dos modelos agora. Diferentes modelos funcionam melhor para diferentes conjuntos de dados não estruturados, dependendo de como você precisa relacionar e consultar os dados. O importante é que os bancos de dados NoSQL não estruturados têm um conceito subjacente diferente de como eles armazenam e recuperam dados, algo que você pode usar para sua vantagem ao criar e executar aplicações de nuvem no Azure.

10.1.3 Escalar bancos de dados

Lembre-se de que eu disse que para um banco de dados estruturado, o banco de dados inteiro normalmente precisa existir em cada servidor. À medida que você usa bancos de dados muito grandes, você precisa de servidores cada vez maiores para executá-los. Você pode nunca precisar trabalhar com bancos de dados que aumentam a centenas de gigabytes ou até mesmo terabytes, mas os bancos de dados NoSQL abordam como os bancos de dados aumentam e são escalados de forma diferente dos bancos de dados SQL. A diferença é que os bancos de dados NoSQL em geral são escalados horizontalmente, e não verticalmente.

Há um limite para o quanto você pode escalar verticalmente uma VM, ou seja, dar a ela mais memória e CPU. Você começa a encontrar problemas de performance em outras partes da camada de computação à medida em que aproveita ao máximo a taxa de transferência de armazenamento e a largura de banda de rede. E isso sem recorrer à sua carteira (ou carteira do seu chefe) quando você vê o tamanho da fatura para essas grandes VMs. Como uma recapitulação do capítulo 9, a escala vertical é ilustrada na Figura 10.3. Agora imagine um cluster dessas VMs de banco de dados grandes, porque você quer redundância e resiliência para sua aplicação, certo?

Por outro lado, escalar horizontalmente permite que você execute VMs de banco de dados com menos recursos e um preço mais baixo. Para fazer isso, os bancos de dados do NoSQL dividem o banco em nós de dados e roteiam solicitações da sua aplicação para o nó apropriado. Os outros nós no cluster não precisam estar cientes de onde todos os dados estão armazenados.

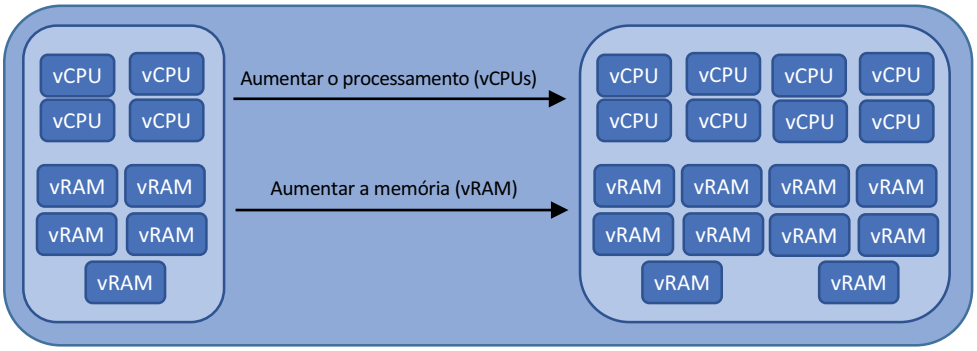


Figura 10.3 Os bancos de dados estruturados tradicionais são escalados verticalmente. À medida que o banco de dados aumenta, você aumenta a quantidade de armazenamento, memória e capacidade da CPU no servidor.

Eles só precisam responder às suas próprias solicitações. Você pode adicionar rapidamente nós a um cluster em resposta à demanda do cliente, conforme necessário.

Como resultado, em um banco de dados NoSQL, o banco de dados inteiro não precisa caber na memória de um host. Somente parte do banco de dados, um *fragmento*, precisa ser armazenado e processado. Se sua aplicação funcionar com grandes quantidades de dados *estruturados*, um banco de dados NoSQL poderá prejudicar o performance porque os hosts diferentes são consultados sobre suas informações para retornar ao cliente. Se você tiver uma grande quantidade de dados *não estruturados* para processar, os bancos NoSQL podem oferecer uma melhoria de performance, sem mencionar um benefício de gerenciamento e eficiência. Um exemplo de escala de bancos de dados não estruturados horizontalmente entre hosts é mostrado na Figura 10.4.

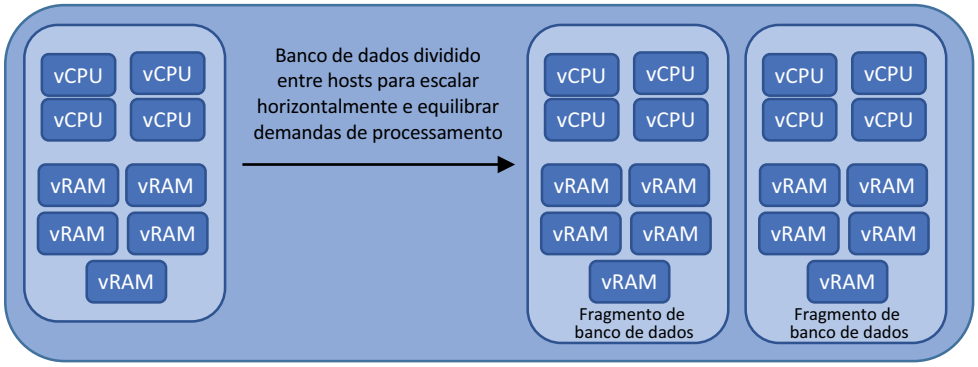


Figura 10.4 Os bancos de dados NoSQL não estruturados são escalados horizontalmente. À medida que um banco de dados cresce, ele é fragmentado em segmentos de dados que são distribuídos entre servidores de banco de dados.

10.1.4 Combinar tudo com o Cosmos DB

Então, o que é o Cosmos DB? É uma plataforma de banco de dados globalmente distribuída, com dimensionamento automático, que permite que você use várias formas de bancos NoSQL. Como com serviços como o Web Apps, o Cosmos DB abstrai bastante a camada de gestão. Ao criar um aplicativo Web, você não precisa configurar o balanceamento de carga ou o clustering. Você escolhe suas regiões e pode configurar o dimensionamento automático e, em seguida, carregar o código da aplicação. A plataforma do Azure manipula como replicar e distribuir o tráfego do aplicativo Web de uma maneira altamente disponível. Com o Cosmos DB, você não precisa se preocupar com o tamanho de um banco de dados necessário, a quantidade de memória a atribuir ou a forma como replicar dados para redundância. Você escolhe a quantidade de taxa de transferência que pode ser necessárias e quais regiões para armazenar seus dados e, em seguida, começa a adicionar dados.

Este capítulo usa um modelo SQL para o Cosmos DB, mas os dados são armazenados em um formato JSON NoSQL. Estes podem ser conceitos novos, mas me acompanhe. Você pode usar outros modelos, incluindo Mongo, Cassandra, Gremlin e Table. A funcionalidade é a mesma para todas eles: escolha seu modelo, escolha suas regiões e adicione seus dados. Esse é o poder do Cosmos DB.

10.2 Criar uma conta e banco de dados do Cosmos DB

Vamos ver o Cosmos DB e bancos de dados não estruturados em ação, o que podemos fazer de algumas maneiras. A primeira é usar o portal do Azure para criar uma conta, selecionar e criar um modelo de banco de dados e inserir dados no banco de dados para que seu aplicativo possa consultá-los. Ou você pode usar a CLI do Azure, o Azure PowerShell ou kits de desenvolvimento de software (SDKs) específicos da linguagem para criar tudo em código. Vamos usar o portal do Azure para que também possamos criar e consultar os dados visualmente.

10.2.1 Criar e preencher um banco de dados do Cosmos DB

No capítulo 4, você criou seu primeiro banco de dados NoSQL com uma tabela de armazenamento do Azure. Vamos usar o Cosmos DB para criar um banco de dados semelhante, desta vez um que oferece todas as opções de redundância geográfica e replicação para garantir que sua loja online permita que os clientes peçam pizzas sem qualquer tempo de inatividade. Vamos criar uma conta do Cosmos DB e um banco de dados de documentos e, em seguida, adicionar algumas entradas para três tipos de pizza, como mostrado na Figura 10.5.

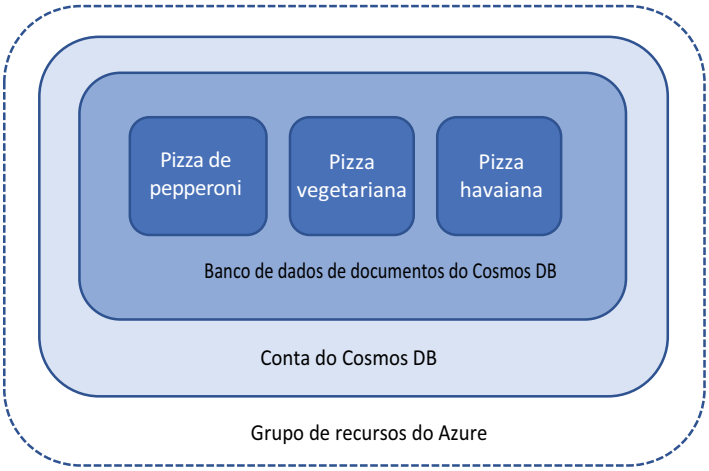


Figura 10.5 Nesta seção, você criará um grupo de recursos e uma conta do Cosmos DB. Um banco de dados de documentos é criado nessa conta, e você adicionará três entradas para representar um menu básico para sua pizzaria.

Experimente agora

Para ver o Cosmos DB em ação, crie uma conta usando o portal do Azure:

1. Abra o portal do Azure e selecione Criar um recurso no canto superior esquerdo do painel.
2. Pesquise e selecione Azure Cosmos DB e escolha Criar.
3. Escolha criar um grupo de recursos, como `azuremolchapter10` e insira um nome exclusivo para sua conta do Cosmos DB, como `azuremol`.

4. O tipo de modelo que você pode usar para seu banco de dados é chamado de API. Neste exemplo, escolha Core (SQL) no menu suspenso.
5. Para local, selecione Leste dos EUA. O Cosmos DB está disponível em todas as regiões do Azure, mas, neste exemplo, o aplicativo Web implantadas no laboratório de fim de capítulo espera que você use Leste dos EUA.
6. Deixe a opção para redundância geográfica desabilitada, juntamente com quaisquer outros recursos, como regiões de várias gravações. A seção 10.2.2 fornece mais detalhes sobre como replicar seu banco de dados globalmente.

Tráfego seguro com pontos de extremidade de serviço

Você tem a opção de conectar seu Cosmos DB a uma rede virtual do Azure com algo chamado *ponto de extremidade de serviço*. Não discutiremos essa opção agora, mas é um recurso interessante que ajuda a proteger sua instância, permitindo o acesso ao banco de dados somente de uma rede virtual definida.

Se criar aplicações de middleware que usam o Cosmos DB ou aplicações somente internas, você poderá usar um ponto de extremidade de serviço de rede virtual para reduzir o acesso de uma rede virtual específica, não pela Internet e com um ponto de extremidade público. Um número crescente de serviços do Azure oferece suporte a esses tipos de pontos de extremidade, e é outro exemplo de oferecer opções a você para proteger seu ambiente a fim de atender aos seus requisitos de negócios.

7. Quando estiver pronto, analise e crie sua conta do Cosmos DB. Leva alguns minutos para criar a conta.

Seu banco de dados está vazio agora, então vamos explorar como você pode armazenar alguns dados básicos para o seu menu de pizzaria. O Cosmos DB agrupa os dados em um banco de dados em algo chamado *contêiner*. Não, esse não é o mesmo tipo de contêiner que é a força motriz por trás do Docker, do Kubernetes e de aplicações nativas de nuvem que você pode ter ouvido falar. Essa confusão de nomes não é tão importante, mas me acompanhe por enquanto.

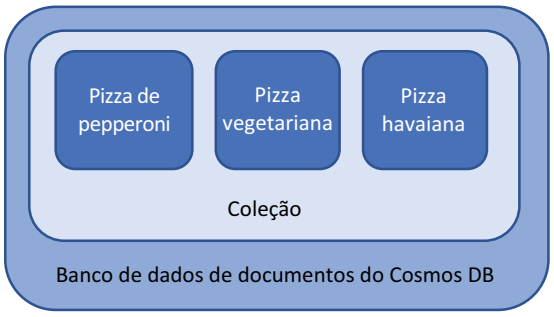


Figura 10.6 Um banco de dados do Cosmos DB que usa o modelo de documento armazena dados em coleções. Essas coleções permitem que você agrupe dados para indexação e consulta mais rápidas.

Em bancos de dados do Cosmos DB que usam o modelo de documento, os dados são logicamente agrupados em contêineres chamados *coleções*. Outros modelos de API têm um nome um pouco diferente para a entidade de contêiner, como *gráfico* para a API Gremlin. Para nossa API SQL, coleções armazenam partes de dados relacionadas que podem ser indexadas e consultadas rapidamente, como mostrado na Figura 10.6. As coleções não são totalmente diferentes de como você organiza um banco de dados SQL tradicional em tabelas, mas as coleções oferecem muito mais flexibilidade quando se trata de distribuir os dados para performance ou redundância.

Como o Cosmos DB foi projetado para lidar com grandes quantidades de dados e taxa de transferência, você pode escolher como dimensionar e controlar o fluxo e o custo desses dados. A taxa de transferência é calculada em unidades de solicitação por segundo (RU/s) e uma unidade de solicitação é o equivalente a 1 KB de dados do documento. Basicamente, você define quanta largura de banda você deseja que seu banco de dados tenha. Caso você não tenha adivinhado, quanto mais largura de banda (RU/s) você quiser, mais você paga. O Cosmos DB mostra a quantidade de dados que você está usando e a quantidade de taxa de transferência que sua aplicação usa, e você normalmente não precisa se preocupar muito com as coisas de dimensionamento correto. Porém, para a sua pizzaria, não vamos começar com algo muito diferente.

Experimente agora

Para criar uma coleção e preencher algumas entradas no banco de dados, conclua as etapas a seguir:

1. Navegue e selecione Grupo de recursos na barra de navegação à esquerda no portal do Azure.
2. Escolha o grupo de recursos no qual você criou seu banco de dados do Cosmos DB, como azuremolchapter10.
3. Selecione sua conta do Cosmos DB na lista de recursos e escolha a página Visão geral.
4. Escolha adicionar um contêiner.
5. Este é o seu primeiro banco de dados, por isso, insira um nome, como `pizzadb`.
6. Deixe a taxa de transferência definida com o valor padrão.
7. Para a ID do contêiner, insira `pizzas`. Esta etapa cria um contêiner lógico que você pode usar para armazenar os itens em seu menu de pizzaria.
8. Insira uma chave de partição de `/description` para garantir que os tipos de pizza sejam distribuídos de maneira uniforme.

   A chave de partição identifica como os dados podem ser separados no banco de dados. Não é realmente necessário em um banco de dados de exemplo pequeno como este, mas usá-lo é uma prática recomendada à medida que o aplicativo é escalado.
9. Não escolha adicionar uma chave exclusiva. As chaves definem mais logicamente o recipiente, como para subseções de clientes de alimentos podem encomendar. A coleção mais ampla é para o seu menu, mas, em bancos de dados muito maiores, você pode querer chaves de partição como pizzas, bebidas e sobremesas.
10. Para criar o banco de dados e a coleção, selecione OK.

Agora você tem uma Conta do cosmos DB, um banco de dados e uma coleção, mas o Cosmos DB ainda não contém suas pizzas. Você pode importar alguns dados ou escrever código que insere muitos dados. Vamos criar manualmente três pizzas para explorar algumas das ferramentas gráficas incorporadas ao portal do Azure para navegar, consultar e manipular os dados no banco de dados do Cosmos DB.

Experimente agora

Para criar e adicionar algumas entradas ao banco de dados, conclua as etapas a seguir, como mostrado na Figura 10.7:

1. Na sua conta do Cosmos DB, escolha o Data Explorer no menu à esquerda na janela Visão geral.

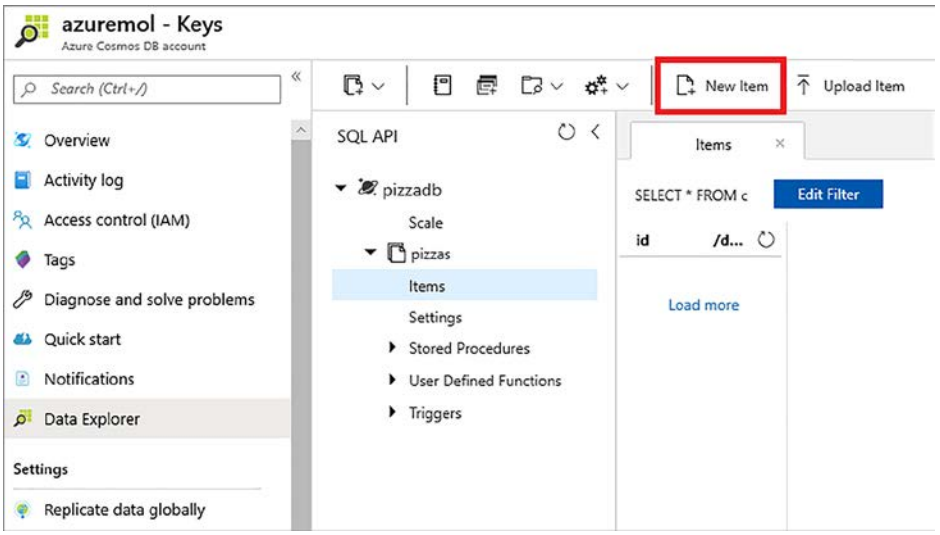


Figura 10.7 Com o Data Explorer no portal do Azure, você pode navegar em suas coleções para consultar ou criar novos documentos. Esta ferramenta gráfica permite gerenciar rapidamente a sua base de dados em um navegador da Web.

2. Expanda o primeiro banco de dados `pizzadb` e a coleção `pizzas`.
3. Adicione um novo item para colocar algumas pizzas no banco de dados. Os dados são adicionados no formato JSON.
4. Na caixa de texto, substitua textos existentes pelos seguintes dados para criar um novo item de menu para uma pizza de pepperoni básica:

```
{
    "description": "Pepperoni",
    "cost": "18"
}
```

5. Para adicionar os dados ao banco de dados, selecione Salvar.
6. Adicione outra pizza ao seu menu. Desta vez, adicione uma propriedade para indicar que esta pizza tem uma crosta sem glúten. Você não precisa fazer nada de especial para o banco de dados subjacente. Basta adicionar outra propriedade aos seus dados. Para adicionar outro novo item, insira os seguintes dados e selecione Salvar:

```
{
    "description": "Veggie",
    "cost": "15",
    "gluten": "free"
}
```

7 Adicione um tipo final de pizza. Desta vez, adicione uma propriedade que inclui quais coberturas estão na pizza. Para adicionar mas um novo item, insira os seguintes dados e selecione Salvar:

```
{
    "description": "Hawaiian",
    "cost": "12",
    "toppings": "ham, pineapple"
}
```

Essas três entradas mostram o poder de um banco de dados NoSQL. Você adicionou propriedades às entradas sem precisar atualizar o esquema de banco de dados. Duas propriedades diferentes mostraram que a pizza vegana tem uma crosta sem glúten e quais coberturas estão na pizza havaiana. O Cosmos DB aceita essas propriedades adicionais e esses dados agora estão disponíveis para suas aplicações.

Algumas propriedades extras JSON são adicionadas para itens como `id`, `_rid` e `_self`. Essas não são propriedades com que você precisa se preocupar muito por enquanto. O Cosmos DB usa essas propriedades para rastrear e identificar os dados. Você não deve editá-las ou excluí-las manualmente.

10.2.2 Adicionar redundância global a um banco de dados do Cosmos DB

Você tem um banco de dados do Cosmos DB que armazena um menu básico de pizza na região Leste dos EUA. Mas, sua pizzaria está pronta para abrir franquias no mundo todo! Você deseja replicar os dados sobre suas pizzas para regiões do Azure em diferentes locais, perto de seus novos clientes.

Por que quer fazer isso? Se todos os seus clientes lerem e gravarem dados do banco de dados em uma região, que é muito tráfego potencial cruzamento de cabos o oceano e roteamento em todo o mundo. Para fornecer a melhor experiência de baixa latência aos clientes, você pode replicar seus dados para regiões do Azure em todo o mundo e os clientes podem se conectar à réplica mais próxima a eles, como mostrado na Figura 10.8.

Os modelos de consistência e as garantias são incorporados à plataforma do Cosmos DB para lidar com a consistência e replicação de dados para você. Você designa uma ou mais regiões como o local de gravação principal. Os exemplos deste livro usam um único ponto de gravação, mas você pode usar o suporte a vários mestres para gravar dados no ponto de extremidade mais próximo que é propagado

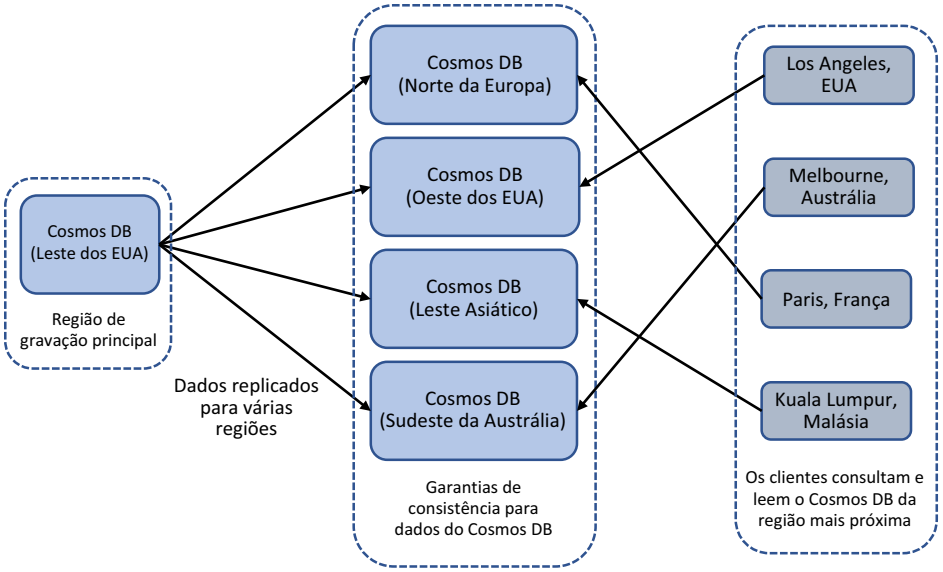


Figura 10.8 Os dados são replicados de uma instância principal do Cosmos DB para várias regiões do Azure ao redor do mundo. Os aplicativos Web podem ser direcionadas para leitura de sua região mais próxima, e os clientes podem ser roteados dinamicamente para o local mais próximo para minimizar a latência e melhorar os tempos de resposta.

de forma assíncrona para outras regiões. Os dados também são replicados rapidamente para as regiões de leitura que você designar. Você pode controlar a ordem de failover, designar regiões de leitura e, com sua aplicação, especificar automaticamente ou manualmente as regiões de leitura.

Você pode definir um modelo de consistência (que é mais uma consideração de design do que em um operacional (que define a rapidez com que as gravações em várias regiões são replicadas. Os modelos de consistência variam de *strong*, que aguarda gravações replicadas a serem confirmadas por réplicas e, assim, garantem que as leituras sejam consistentes, a *eventual*, que é mais relaxada. O modelo eventual garante que todos os dados sejam replicados, mas pode haver um pouco de atraso quando leituras de réplicas retornam valores diferentes até que estejam todos sincronizados.

Há um equilíbrio entre uma distribuição geográfica mais limitada, como com o modelo de consistência forte, e uma replicação geográfica mais ampla, como modelo de consistência eventual, mas com o entendimento de que há um pouco de atraso à medida que os dados são replicados. Há também largura de banda e custos de processamento, dependendo de como consistentemente e oportuna você deseja que os dados sejam replicados. A plataforma do Azure manipula a replicação subjacente de dados do seu ponto de gravação; você não precisa criar suas aplicações para replicar os dados ou determinar a melhor forma de ler dados de pontos de extremidade replicados.

Em uma escala global, que você pode ter várias VMs ou aplicativos Web como você criou em capítulos anteriores, mas em diferentes regiões ao redor do mundo. Esses aplicativos se conectam a uma instância local do Cosmos DB para consultar e ler todos os seus dados. Por meio de alguns recursos de tráfego de rede do Azure legais que discutiremos no capítulo 11, os usuários podem ser roteados automaticamente para uma dessas instâncias locais do aplicativo Web, que também usam uma instância local do Cosmos DB. Em caso de paralisações ou manutenção regionais, toda a plataforma roteia o cliente para a instância mais próxima.

No mundo do banco de dados estruturado tradicional, em que você gerencia as VMs, a instalação do banco de dados e a configuração de cluster, essa configuração leva um planejamento de design sério e é complicada de implementar. Com o Cosmos DB, o processo precisa de três cliques do mouse. Sério.

Experimente agora

Para replicar os dados do Cosmos DB globalmente, conclua as etapas a seguir:

1. Navegue e selecione Grupo de recursos na barra de navegação à esquerda no portal do Azure.
2. Escolha o grupo de recursos no qual você criou seu banco de dados do Cosmos DB, como `azuremolchapter10`.
3. Selecione sua conta do Cosmos DB na lista de recursos. Esses dois cliques de mouse foram gratuitos, mas a contagem começa a partir daqui.
4. Selecione a opção de menu à esquerda para replicar dados globalmente. O mapa, que mostra todos as regiões do Azure disponíveis, mostra que seu banco de dados está atualmente disponível na região Leste dos EUA (Figura 10.9).

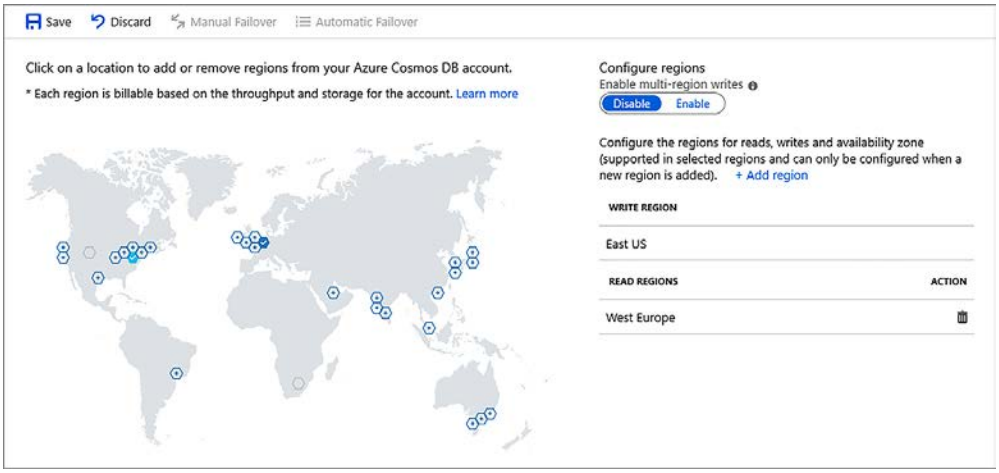


Figura 10.9 Selecione uma região do Azure para replicar seu banco de dados do Cosmos DB e escolha Salvar. Essas são todas as etapas necessárias para distribuir globalmente seus dados.

5 Escolha Oeste da Europa e, em seguida, selecione Salvar. Você pode escolher qualquer região do Azure que desejar, mas o laboratório de fim de capítulo espera que seus dados sejam replicados para Oeste da Europa. Leva alguns instantes para replicar os dados para a região selecionada e colocar os dados online para que suas aplicações possam usá-los.

Certo, conte os cliques do mouse. Três cliques, certo? Sejamos generosos e consideremos os dois primeiros cliques do mouse para selecionar o grupo de recursos e a conta do Cosmos DB. Assim, em não mais de cinco cliques do mouse e uma questão de segundos, você criou uma instância de réplica do seu banco de dados que permite que suas aplicações acessem dados da região mais próxima a eles. Você pode fazer isso com um cluster MySQL tradicional? Me mande um tweet @fouldsy se você pode fazer isso rapidamente fora do Cosmos DB!

Com o seu banco de dados agora distribuído globalmente, é preciso um monte de alterações em seu código para determinar qual região do Cosmos DB para se conectar? Como você pode manter todas essas versões diferentes de suas aplicações com base em qual região do Azure são executadas? É fácil: deixe a plataforma do Azure determinar tudo para você.

10.3 Acessar dados distribuídos globalmente

Na maioria das vezes, a plataforma do Azure determina a melhor localização para a interação com sua aplicação. Uma aplicação normalmente precisa ler e gravar dados. Você pode definir as políticas de failover para seu banco de dados do Cosmos DB, que controla a localização de gravação principal. Essa localização de gravação serve como o hub central para garantir que os dados sejam replicados de forma consistente entre regiões. Porém, o seu aplicativo Web pode normalmente ler de várias regiões disponíveis para acelerar as consultas e retornar dados para o cliente. Tudo isso é tratado por chamadas REST.

Vejamos o que acontece na CLI Azure quando você pede informações sobre um banco de dados do Cosmos DB. Esse processo é parecido como uma aplicação que faz uma conexão com um banco de dados, mas impede que você se aprofunde muito no código.

Experimente agora

Use `az cosmosdb show` para encontrar informações sobre sua localização de leitura e gravação:

1 Abra o portal do Azure em um navegador da Web e abra o Cloud Shell.

2 Use `az cosmosdb show` para visualizar os locais de leitura e gravação para o banco de dados do banco de dados Cosmos DB.

Digite o nome do grupo de recursos e o nome do banco de dados que você criou nos exercícios "Experimente agora" anteriores. No exemplo a seguir, o grupo de recursos é `azuremolchapter10` e o nome do banco de dados do Cosmos DB é `azuremol`:

```
az cosmosdb show \
  --resource-group azuremolchapter10 \
  --name azuremol
```

Uma grande quantidade de saída é retornada deste comando, por isso, vamos examinar as duas partes principais: localizações de leitura e de gravação. Aqui está um exemplo de saída da seção `readLocations`:

```
"readLocations": [
    {
      "documentEndpoint":"https://azuremol-eastus.documents.azure.com:443/",
      "failoverPriority": 0,
      "id": "azuremol-eastus",
      "isZoneRedundant": "false",
      "locationName": "East US",
      "provisioningState": "Succeeded"
    },
    {
      "documentEndpoint":
      ➥"https://azuremol-westeurope.documents.azure.com:443/",
      "failoverPriority": 1,
      "id": "azuremol-westeurope",
      "isZoneRedundant": "false",
      "locationName": "West Europe",
      "provisioningState": "Succeeded"
      }
],
```

Quando sua aplicação faz uma conexão com um banco de dados do Cosmos DB, você pode especificar uma política de conexão. Se os bancos de dados não são normalmente sua praia, pense em uma conexão ODBC básica que você pode criar em uma máquina Windows. A cadeia de conexão normalmente define um nome do host, um nome de banco de dados, uma porta e credenciais. No Cosmos DB não é diferente. Você pode se conectar ao Cosmos DB de várias linguagens, incluindo .NET, Python, Node. js e Java. As linguagens podem ser diferentes, mas todos os SDKs têm uma configuração semelhante: descoberta de ponto de extremidade. Duas propriedades principais da política de conexão são importantes:

- *Descoberta de ponto de extremidade* — O SDK lê todos os pontos de extremidade disponíveis do Cosmos DB e usa a ordem de failover especificada. Essa abordagem garante que sua aplicação sempre segue a ordem que você especificar no nível de banco de dados. Por exemplo, você pode querer todas as leituras para passar por Leste dos EUA e usar apenas a Oeste da Europa quando há manutenção no local principal.
- *Localizações de ponto de extremidade preferenciais* — Você especifica as localizações que deseja usar. Um exemplo é se você implantar seu aplicativo na Europa Ocidental e quiser garantir que você use o ponto de extremidade da Europa Ocidental. Você perde um pouco de flexibilidade como pontos de extremidade são adicionados ou removidos, mas verifique se o ponto de extremidade padrão está próximo ao seu aplicativo sem precisar de roteamento de rede mais avançado para ajudar a determinar isso para você.

Normalmente, sua aplicação permite que o SDK do Cosmos DB lide com essa tarefa. Sua aplicação não altera como ele lida com a conexão com o banco de dados: ele só sabe que *pode* se conectar a localizações diferentes. Porém, o SDK é o que *faz* a conexão e usa esse reconhecimento de localização.

A figura 10.10 mostra uma abordagem simplificada de como essa conscientização de local é usada entre sua aplicação e o SDK. Novamente, a linguagem não importa, e a abordagem é a mesma. A figura usa o SDK do Python porque essa é a linguagem em que alguns exemplos foram escritos. Este exemplo também pressupõe que você esteja usando localizações de ponto de extremidade automáticas.

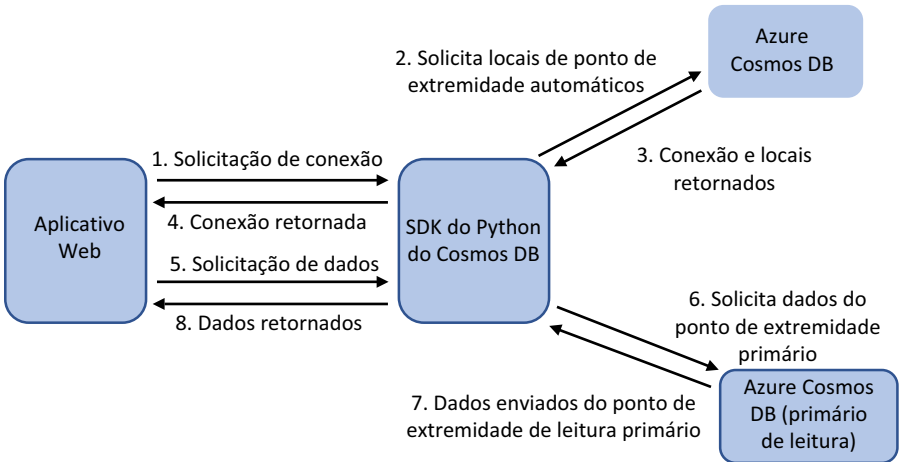


Figura 10.10 O fluxo de solicitações por meio de um SDK do Cosmos DB quando uma aplicação usa o reconhecimento de localização para consultar o Cosmos DB

As etapas ilustradas na Figura 10.10 são as seguintes:

1. Sua aplicação precisa fazer uma conexão com um banco de dados do Cosmos DB. Na diretiva de conexão, você habilita a descoberta de ponto de extremidade automática. A aplicação usa o SDK do Cosmos DB para fazer uma conexão de banco de dados.
2. O SDK do Cosmos DB faz uma solicitação de conexão e indica que deseja usar localizações de ponto de extremidade automáticas.
3. Uma conexão é retornada com base nas credenciais e no banco de dados solicitado.
4. O SDK retorna um objeto de conexão para a aplicação usar. As informações de local são abstraídas da aplicação.
5. A aplicação solicita alguns dados do banco de dados do Cosmos DB. O SDK é usado novamente para consultar e obter os dados.
6. O SDK usa a lista de pontos de extremidade disponíveis e faz a solicitação para o primeiro ponto de extremidade disponível. Em seguida, o SDK usa o ponto de extremidade de conexão para consultar os dados. Se o ponto de extremidade primário não estiver disponível, como durante um evento de manutenção, a próxima localização do ponto de extremidade será usada automaticamente.
7. O Cosmos DB retorna os dados da localização do ponto de extremidade.
8. O SDK transmite os dados do Cosmos DB de volta para a aplicação para analisar e exibir conforme necessário.

As últimas coisas a examinar no Cosmos DB são as chaves de acesso, que permitem que você controle quem pode acessar os dados e quais permissões cada pessoa tem. As chaves podem ser regeneradas, e, como você faz com senhas, você pode querer implementar uma política para executar regularmente este processo de regeneração de chaves. Para acessar os dados distribuídos no Cosmos DB, você precisa obter suas chaves. O portal do Azure fornece uma maneira de exibir todas as chaves e cadeias de conexão para seu banco de dados.

Experimente agora

Para exibir as chaves da sua conta do Cosmos DB, conclua as etapas a seguir:

1. Navegue e selecione Grupo de recursos na barra de navegação à esquerda no portal do Azure.
2. Escolha o grupo de recursos no qual você criou seu banco de dados do Cosmos DB, como `azurremolchapter10`.
3. Selecione sua conta do Cosmos DB na lista de recursos.
4. No lado esquerdo, escolha Chaves.
5. Anote o URI e a chave primária (Figura 10.11). Você usará esses valores no laboratório de fim de capítulo.

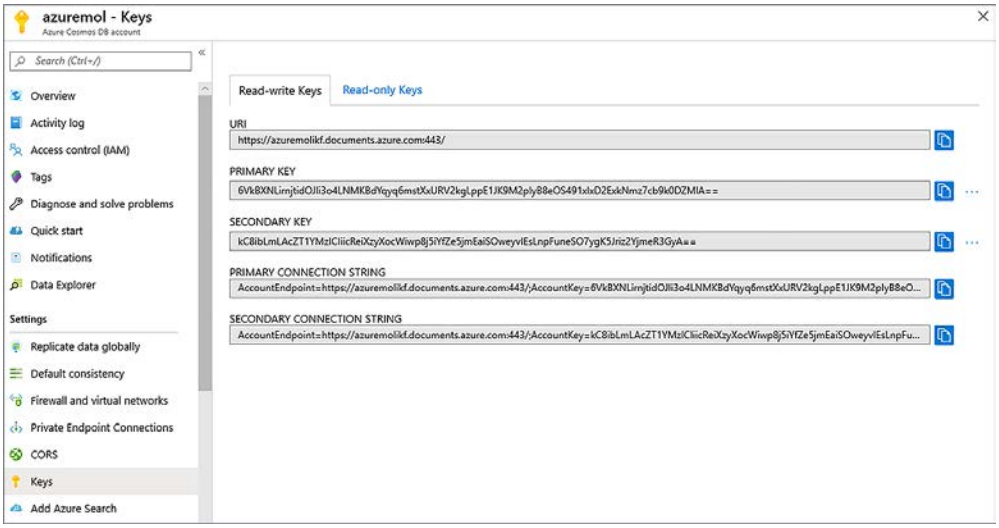


Figura 10.11 A seção Chaves da sua conta do Cosmos DB lista as informações de conexão e as chaves de acesso. Você precisa dessas informações quando você constrói e executa aplicações, como no laboratório de fim de capítulo.

Muito no Cosmos DB acontece nos bastidores para distribuir seus dados e permitir que suas aplicações leiam e gravem a partir dos locais mais adequados. Porém, isso é tudo o que é necessário. Um reconhecimento do que o serviço Cosmos DB faz ajuda a projetar e planejar sua aplicação ou solucionar problemas se as aplicações não permitem que o SDK execute operações de leitura e gravação conforme necessário. Porém, você não precisa se preocupar como e quando. Concentre-se em suas aplicações e use os serviços do Azure como o Cosmos DB para fornecer a funcionalidade e os benefícios da nuvem que permitem que você opere em uma escala global.

10.4 Laboratório: Implantar um aplicativo Web que usa o Cosmos DB

Na seção 10.2.2, você distribuiu o banco de dados do Cosmos DB globalmente. Em seguida, passamos por muita teoria sobre como os aplicativos Web passam ler de localizações ao redor do mundo. Agora você provavelmente só quer ver o cosmos DB em ação, então aqui está sua chance. Neste laboratório, o aplicativo Web básico de capítulos anteriores é usado, mas, desta vez, o menu de pizza vem dos itens adicionados ao banco de dados do Cosmos DB em um exercício anterior, "Experimente agora":

1. No portal do Azure, crie um aplicativo Web.
2. Como a pizzaria não é mais uma página HTML básica, escolha `Nó LTS` para o tempo de execução executado no Linux.
3. Quando o aplicativo Web estiver pronto, crie uma origem de implantação (repositório Git local). As etapas são as mesmas que quando você criou um em capítulos anteriores, como o capítulo 3. Por isso, confira esses exercícios, se você precisar de um lembrete.
4. Abra o Cloud Shell. Nos capítulos anteriores, você obteve uma cópia dos exemplos do Azure no GitHub. Se você não fez isso, pegue uma cópia da seguinte forma:

   ```
   git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
   ```

5. Mude para o diretório que contém o exemplo do aplicativo Web do Cosmos DB:

   ```
   cd ~/azure-mol-samples-2nd-ed/10/cosmosdbwebapp
   ```

6. Edite o arquivo de configuração com o URI do banco de dados e a chave de acesso que você copiou no exercício anterior ,"Experimente agora", para visualizar suas chaves do Cosmos DB:

   ```
   nano config.js
   ```

7. Grave o arquivo pressionando o CTRL + O e, em seguida, saia pressionando CTRL + X.
8. Adicione e confirme suas alterações no Git com o seguinte comando:

   ```
   git init && git add . && git commit -m "Pizza"
   ```

9. Crie um link para o novo repositório Git no slot de preparação com `git remote add azure`, seguido pela URL de implantação do Git de slot de preparação.
10. Use `git push azure master` para enviar suas alterações ao seu aplicativo Web.
11. Selecione a URL para seu aplicativo Web na janela Visão geral do portal do Azure.

12 Abra esta URL em um navegador da Web para ver a sua pizzaria, que agora é alimentado pelo Cosmos DB, como mostrado na Figura 10.12.

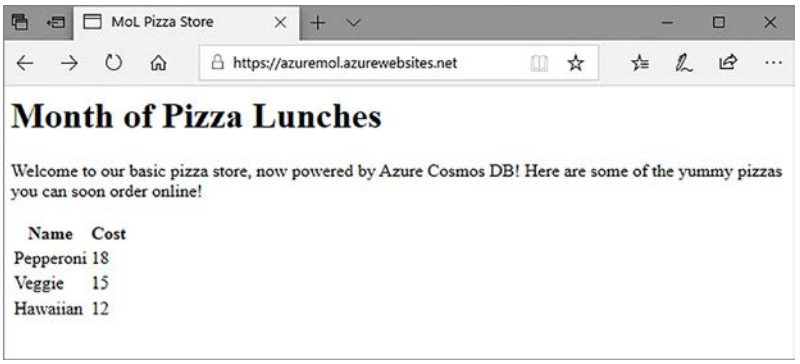


Figure 10.12 O aplicativo Web básico do Azure mostra seu breve menu de pizza com base em dados no banco de dados do Cosmos DB. A pizzaria dos capítulos anteriores é mostrada, mas agora a lista de pizzas e seus preços é alimentada pelo Cosmos DB. O site ainda é básico, pois o objetivo é que você veja o serviço em ação e entenda como você pode começar a criar suas próprias aplicações.

11 Gerenciar tráfego e roteamento de rede

A resolução Domain Name System (DNS) está no centro de quase todas as conexões digitais que você faz. É como você navega na Web, recebe emails, assiste Netflix e faz chamadas do Skype. DNS é o mecanismo que converte um nome, como manning.com, em um endereço IP. Quando eu quero aprender um tópico novo, eu não preciso lembrar de 35.166.24.88. Eu apenas insiro `manning.com` em um navegador da Web e navego por alguns livros. Dispositivos de rede fazem o roteamento do tráfego com base em endereços IP. Por isso, você precisa de uma abordagem que ajude aqueles de nós com memória ruim para fazer coisas como comprar livros ou pizza online.

Nos últimos capítulos, você passou muito tempo aprendendo sobre como criar aplicações que podem ser escaladas, sejam altamente disponíveis e distribuídas globalmente. Uma das últimas peças que faltam é como direcionar clientes de todo o mundo para a instância de aplicação mais apropriada, normalmente a instância mais próxima deles. O Gerenciador de Tráfego do Azure facilita a rota automática dos clientes para suas instâncias de aplicação com base no performance ou na localização geográfica. Neste capítulo, vamos discutir como você pode criar e gerenciar zonas DNS no Azure e, em seguida, como usar o Gerenciador de Tráfego para rotear clientes com consultas DNS, como mostrado na Figura 11.1.

11.1 O que é o Azure DNS?

Você não precisa de uma compreensão profunda de como o DNS funciona para concluir este capítulo e usar o DNS do Azure. A Figura 11.2 mostra uma visão geral de alto nível de como um usuário consulta um serviço DNS para obter o endereço IP de um aplicativo Web. Várias subetapas podem ser usadas em torno das etapas 1 e 2 e, por isso, se você ainda tiver um pouco de tempo em seu intervalo do almoço no final deste capítulo, sinta-se à vontade para ler sobre como as consultas DNS e trabalho de recursão.

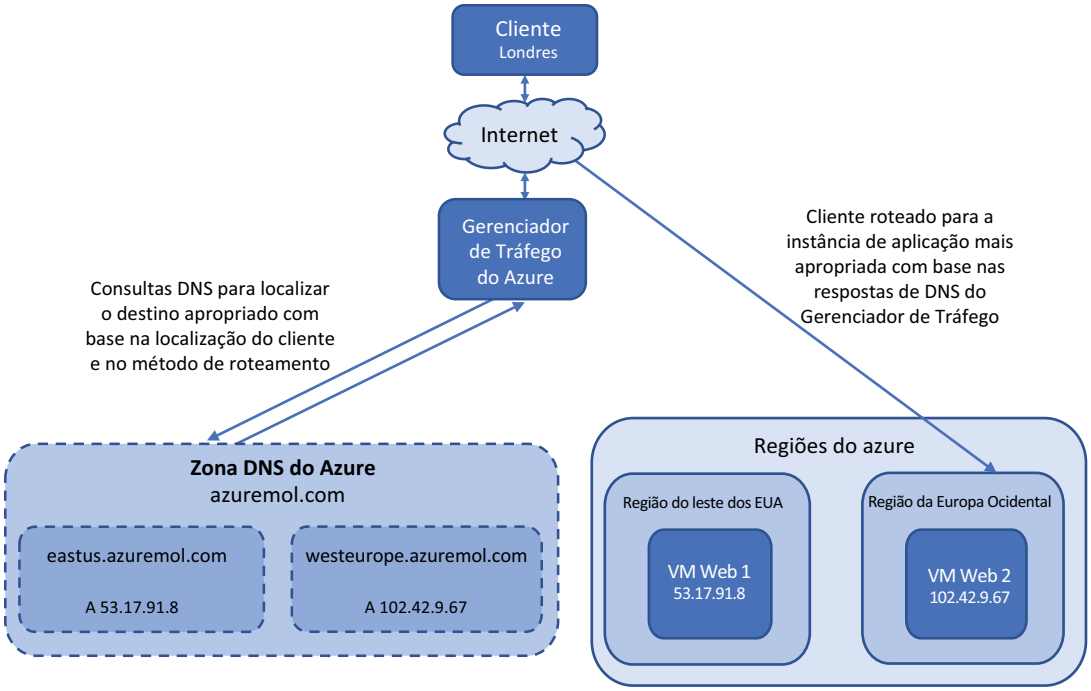


Figura 11.1 Neste capítulo, examinamos como você pode criar zonas DNS no DNS do Azure. Para minimizar a latência e melhorar os tempos de resposta, o Gerenciador de Tráfego pode ser usado para consultar o DNS e direcionar os clientes para sua instância de aplicação mais próxima.

O DNS do Azure funciona da mesma forma que qualquer solução DNS existente que você possa usar ou esteja familiarizado. A zona e os registros são armazenados no Azure e os servidores de nomes que respondem às consultas DNS são distribuídos globalmente nos datacenters do Azure.

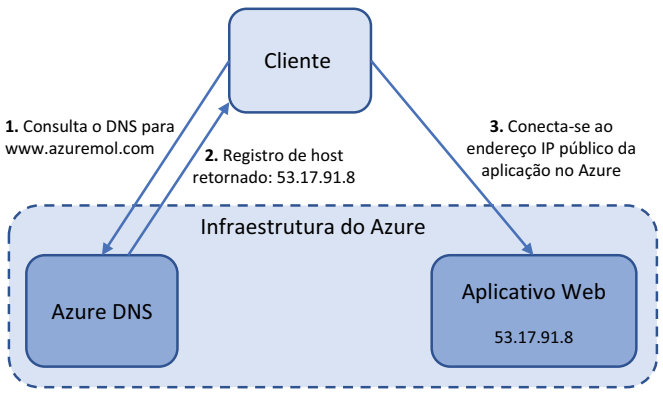


Figura 11.2 Esse fluxo simplificado de tráfego DNS mostra como um usuário envia uma solicitação DNS para www.azuremol.com para um servidor DNS, recebe uma resposta que contém o endereço IP associado e, em seguida, se conecta ao aplicativo Web.

O DNS do Azure oferece suporte a todos os tipos de registro esperados em uma oferta de serviço DNS regular. Os registros IPv4 e IPv6 podem ser criados. Os tipos de registro são os seguintes:

- *A* — Registros de host IPv4, para direcionar clientes para suas aplicações e serviços
- *AAAA* — Registros de host IPv6, para o pessoal legal que usa o IPv6 para direcionar os clientes para suas aplicações e serviços
- *CNAME* — Nome canônico ou alias, registros, como fornecer um nome abreviado que é mais fácil de usar do que o nome do host completo de um servidor
- *MX* — Registros de troca de emails para rotear o tráfego de email para seus servidores de email ou pro vedor
- *NS* — Registros de servidor de nomes, que incluem registros gerados automaticamente para os servidores de nomes do Azure
- *PTR* — Registros de ponteiro, para consultas de DNS reverso para mapear endereços IP para nomes do host
- *SOA* — Registros de início de autoridade, que incluem registros gerados automaticamente para os servidores de nomes do Azure
- *SRV* — Registros de serviço, para fornecer a descoberta de serviços de rede, como para identidade
- *TXT* — Registros de texto, como para Sender Protection Framework (SPF) ou DomainKeys Identified Mail (DKIM)

Em uma configuração típica de DNS, você configura vários servidores DNS. Mesmo com a distribuição geográfica desses servidores para redundância, os clientes podem consultar um servidor de nomes do outro lado do mundo. Esses milissegundos necessários para consultar, resolver e, em seguida, solicitar uma resposta para o aplicativo Web pode somar quando você tem muitos clientes que desejam pedir pizza.

Uma zona DNS do Azure é replicada globalmente nos datacenters do Azure. A rede *anycast* garante que, quando um cliente faz uma consulta DNS ao seu domínio, o servidor de nomes disponível mais próximo responda à sua solicitação. Como o roteamento anycast faz isso? Normalmente, um único endereço IP é propagado em várias regiões. Em vez de usar uma consulta DNS simples que resolve de volta para um único endereço IP que só existe em uma localização, roteamento anycast permite que a infraestrutura determine de forma inteligente de onde uma solicitação é proveniente e encaminhe o cliente para a região propagada mais próxima. Este roteamento permite que seus clientes conectem-se ao seu aplicativo Web mais rapidamente e oferece uma melhor experiência geral do cliente.

Você não precisa ser um especialista em rede para entender totalmente como isso funciona. O Azure faz isso para você. Quando você combina o DNS do Azure com o Gerenciador de Tráfego do Azure (seção 11.2), você não só retorna consultas DNS dos servidores de nomes mais próximos, mas também conecta os clientes à instância de aplicação mais próxima deles. Aproveite cada milissegundo.

11.2 Delegar um domínio real ao DNS do Azure

Quando você registra um domínio real, seu provedor oferece uma interface de gerenciamento e ferramentas para gerenciar esse domínio. Para permitir que os clientes acessem seus serviços e usem a zona e os registros do Azure DNS, você delega a autoridade do seu domínio para os servidores de nomes do Azure. Essa delegação faz com

que todas as consultas DNS sejam direcionadas imediatamente a esses servidores de nomes do Azure, como mostrado na Figura 11.3. Atualmente, o Azure não permite que você adquira e registre domínios na plataforma. Portanto, você precisa comprar o nome de domínio por meio de um registrador externo e, em seguida, apontar os registros NS para os servidores de nomes do Azure.

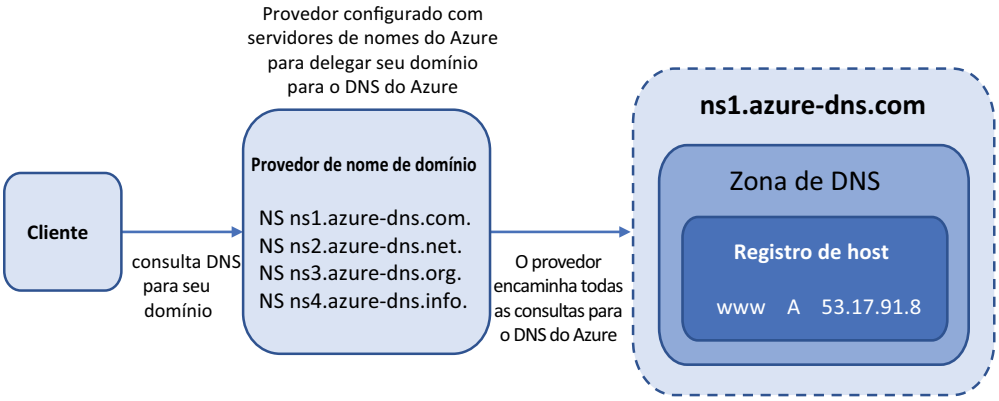


Figura 11.3 Para delegar seu domínio no Azure, configure o provedor de domínio atual com os endereços do servidor de nomes do Azure. Quando um cliente faz uma consulta DNS para seu domínio, as solicitações são enviadas diretamente para os servidores de nomes do Azure para sua zona.

Por que delegar seu DNS ao Azure? Para gerenciar o gerenciamento e as operações. Se você criar serviços adicionais, ajuste a configuração do balanceador de carga, ou se quiser aprimorar os tempos de resposta com o DNS globalmente replicado, o Azure fornecerá essa única interface de gerenciamento para concluir essas tarefas. Quando suas zonas DNS são hospedadas no Azure, você também pode implementar alguns dos recursos de segurança do Resource Manager discutidos no capítulo 6: recursos como o RBAC (controle de acesso baseado em função) para limitar e auditar o acesso às zonas DNS e aos bloqueios de recursos para impedir a exclusão acidental, ou mesmo maliciosa, da zona.

A maioria dos registradores de domínio fornecem interfaces e controles bastante básicos para gerenciar zonas e registros DNS. Para reduzir a sobrecarga de gerenciamento e melhorar a segurança, o DNS do Azure permite que você use a CLI do Azure, o Azure PowerShell ou as APIs REST para adicionar ou editar registros. As equipes de operações podem usar as mesmas ferramentas e fluxos de trabalho para novos serviços integrados; e se ocorrerem problemas, geralmente é mais fácil solucionar problemas quando você pode verificar se o DNS funciona como esperado sem introduzir a variável de um provedor de DNS de terceiros.

Portanto, se você estiver convencido de que há lógica para delegar seu domínio ao DNS do Azure, a quais servidores de nomes do Azure apontar seu domínio? Se você criar uma zona DNS do Azure, os servidores de nomes serão listados no portal, como mostrado na Figura 11.4. Você também pode acessar esses endereços de servidor de nomes com a CLI do Azure ou o Azure PowerShell.

Não há exercícios "Experimente agora" nas páginas anteriores, porque a menos que você compre e configure um domínio real, não poder testar como rotear o tráfego

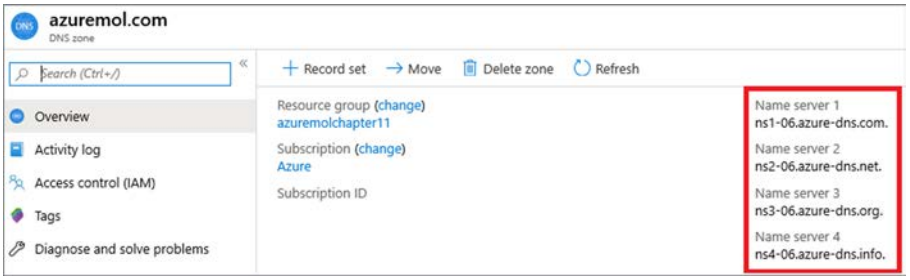


Figura 11.4 Você pode exibir os servidores de nomes do Azure para sua zona DNS no portal do Azure, na CLI do Azure ou no Azure PowerShell.

real. Você pode criar uma zona Azure DNS sem um domínio real, mas nenhum tráfego pode ser roteado para ele. Na vida real, você atualiza os registros NS com seu provedor atual para apontar quaisquer consultas para o seu domínio para os servidores de nomes do Azure. Pode levar de 24 a 48 horas (embora geralmente muito menos tempo) para a delegação do seu domínio se propagar por toda a hierarquia de DNS global. Portanto, planeje adequadamente. Esse comportamento pode causar breves interrupções para clientes que acessam sua aplicação.

11.3 Roteamento e resolução globais com o Gerenciador de Tráfego

Nos capítulos anteriores, você aprendeu sobre aplicações altamente disponíveis que são distribuídas globalmente. A meta é várias instâncias de aplicativo Web ou de VM, em diferentes regiões ou continentes, que se conectam a uma instância de banco de dados do Cosmos DB próxima a elas. Porém, como você faz com que seus clientes conectem-se à VM ou ao aplicativo Web mais próximo que executa sua aplicação?

O Gerenciador de Tráfego do Azure é um serviço de rede que atua como um destino central para seus clientes. Vamos usar o exemplo de um aplicativo Web no endereço www.azure-mol.com. A Figura 11.5 fornece uma visão geral de como o Gerenciador de Tráfego roteia os usuários para a aplicação mais próxima disponível.

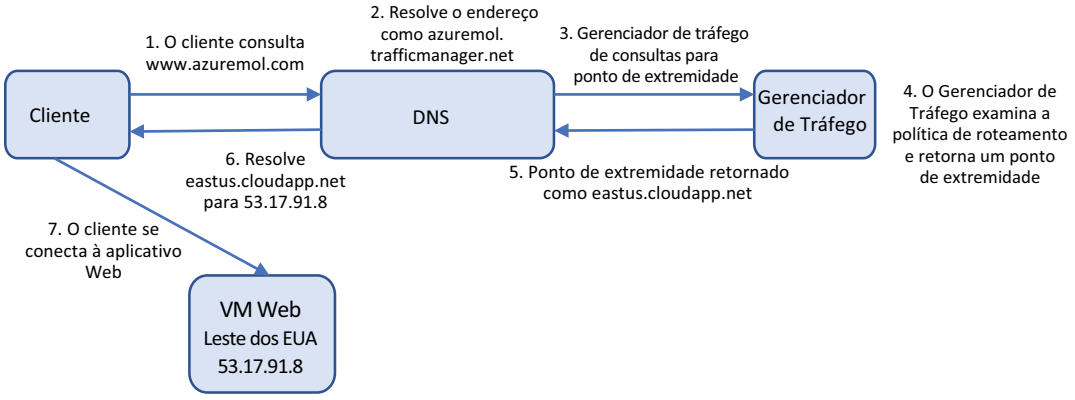


Figura 11.5 Um cliente envia uma consulta DNS para um serviço DNS para www.azuremol.com. O serviço DNS encaminha a consulta para o Gerenciador de Tráfego, que retorna um ponto de extremidade com base no método de roteamento em uso. O ponto de extremidade é resolvido para um endereço IP, que o cliente usa para se conectar ao aplicativo Web.

O Gerenciador de Tráfego não executa a função de um balanceador de carga que você aprendeu no capítulo 8. Como mostrado na Figura 11.5, o Gerenciador de Tráfego roteia o tráfego para um IP público. Vamos examinar o fluxo de tráfego um pouco mais de perto:

1. O usuário faz uma consulta DNS para www.azuremol.com. Seu servidor DNS contata os servidores de nome para azuremol.com (que podem ser servidores de nomes do Azure se você usar o DNS do Azure) e solicita o registro para www.
2. O host www é resolvido para um registro CNAME que aponta para azuremol.trafficmanager.net.
3. O serviço DNS encaminha a solicitação DNS aos servidores de nomes do Azure para trafficmanager.net.
4. O Gerenciador de Tráfego, em seguida, examina a solicitação e determina um ponto de extremidade para encaminhar o usuário. A integridade e status do ponto de extremidade são examinados, como com balanceadores de carga do Azure. O método de roteamento do Gerenciador de Tráfego também é revisado. Os métodos de roteamento que o Gerenciador de Tráfego pode usar são os seguintes:
   - *Prioridade* — Controla a ordem na qual os pontos de extremidade são acessados
   - *Ponderada* — Distribui o tráfego entre pontos de extremidade com base em uma métrica de peso atribuída
   - *Performance* — Roteamento de usuários com base em latência para um ponto de extremidade para que o usuário receba o tempo de resposta mais rápido possível
   - *Geographic* — Associa pontos de extremidade a uma região geográfica e direciona os usuários com base em sua localização
5. O eastus.cloudapp.net de ponto de extremidade é retornado para o serviço DNS pelo Gerenciador de Tráfego.
6. O serviço DNS procura o registro DNS para eastus.cloudapp.net e retorna o resultado da consulta para o cliente.
7. Com o endereço IP de seu ponto de extremidade solicitado, o cliente contata o aplicativo Web diretamente. Neste ponto, o tráfego pode atingir o endereço IP público de um balanceador de carga do Azure em vez de uma VM diretamente.

Como você pode ver, a função do Gerenciador de Tráfego é determinar um ponto de extremidade de determinada aplicação para direcionar os clientes. Algumas verificações de integridade monitoram o status dos pontos de extremidade, semelhantes às sondas de integridade do balanceador de carga que você aprendeu no capítulo 8. E você pode definir uma prioridade ou um mecanismo de roteamento de tráfego ponderado para distribuir os usuários em um conjunto de pontos de extremidade disponíveis, novamente, semelhante a um balanceador de carga. O Gerenciador de Tráfego normalmente direciona o tráfego para um balanceador de carga ou um gateway de aplicação do Azure ou para uma implantação de aplicativo Web.

Azure Front Door

O Gerenciador de tráfego, que examinamos nesta seção, é ótimo para a distribuição global e o roteamento de tráfego. Ele funciona com qualquer tipo de ponto de extremidade da Internet, não apenas recursos no Azure. O roteamento de tráfego é baseado no DNS e não considera a aplicação em si.

Se você precisar de distribuição de tráfego no nível da aplicação e a capacidade de fazer o descarregamento ou por roteamento de solicitações HTTP/HTTPS, o Azure Front Door pode ajudar. Gerenciador de Tráfego

(continuação)

e o Front Door oferecem o mesmo tipo de opções de configuração e serviço, mas o Front Door é especificamente criado para trabalhar com a camada da aplicação. A Front Door também tem alguns truques de performance interessante, como o TCP dividido para colocar conexões em partes menores e reduzir a latência.

De volta ao capítulo 8, analisamos balanceadores de carga e mencionamos o Gateway de Aplicação, que funciona na camada da aplicação e faz coisas como o descarregamento de TLS. O foco nesse capítulo era os balanceadores de carga para ajudá-lo a aprender os conceitos principais, que o Gateway de Aplicação desenvolveria. O mesmo ocorre aqui. Nos concentramos no Gerenciador de Tráfego neste capítulo, embora muitos dos mesmos conceitos e opções de configuração, como opções de roteamento, também estejam disponíveis para o Azure Front Door. Como na maioria das coisas no Azure, o que usar em cada serviço é orientado pelas aplicações executadas e pelas suas necessidades.

11.3.1 Criar perfis do Gerenciador de Tráfego

O Gerenciador de Tráfego usa perfis para determinar qual método de roteamento usar e quais são os pontos de extremidade associados para uma determinada solicitação. Para continuar o tema dos capítulos anteriores sobre uma aplicação distribuída globalmente, você deseja que os usuários usem o aplicativo Web mais próxima a eles. Se você examinar os métodos de roteamento novamente, há duas maneiras de fazer isso:

- *Roteamento de performance* — O cliente é roteado para o ponto de extremidade com a latência mais baixa, em relação à origem da solicitação. Esse método de roteamento fornece alguma inteligência e sempre permite que o Gerenciador de Tráfego encaminhe o cliente para um ponto de extremidade disponível.
- *Roteamento geográfico* — O cliente sempre é roteado para um determinado ponto de extremidade, com base na origem de sua solicitação. Se o cliente está nos Estados Unidos, eles é sempre direcionado para Leste dos EUA, por exemplo. Esse método de roteamento exige que você defina regiões geográficas a serem associadas a cada ponto de extremidade.

Ao usar o roteamento geográfico, você tem um pouco mais de controle sobre os pontos de extremidade que os clientes usam. Pode haver razões regulatórias que exigem que os clientes em uma determinada região sempre usem pontos de extremidade na mesma região. Os exercícios usam pontos de extremidade geográficos para mostrar um exemplo mais real, porque há um truque para roteamento geográfico: você deve especificar um *perfil filho*, não um ponto de extremidade diretamente.

O céu não vai cair se você usar o método de roteamento geográfico com pontos de extremidade, mas a prática recomendada é usar outro perfil do Gerenciador de Tráfego para passar o tráfego para o ponto de extremidade final. Por quê? As regiões só podem ser associadas a um perfil do Gerenciador de Tráfego. Nos capítulos anteriores sobre alta disponibilidade, você sempre quis ter certeza de que tinha redundância. Se você associar uma região a um determinado ponto de extremidade e usar o roteamento geográfico, não terá opção de failover se esse ponto de extremidade encontrar um problema ou se executar a manutenção.

Em vez disso, perfis filho aninhados permitem que você defina uma prioridade que sempre direciona o tráfego para um ponto de extremidade íntegro. Se o ponto de extremidade não estiver íntegro, o tráfego vai para um ponto de extremidade alternativo. A Figura 11.6 mostra o tráfego fazendo failover em uma região diferente, embora você também possa criar várias instâncias do aplicativo Web no Oeste dos EUA e usar um método de roteamento ponderado no perfil filho. À medida que você começa a expandir seu ambiente de aplicação, reserve um tempo para pensar na melhor forma de fornecer alta disponibilidade para pontos de extremidade por trás do Gerenciador de Tráfego. Para esses exemplos, você criará failover entre regiões para ver claramente as diferenças de comportamento.

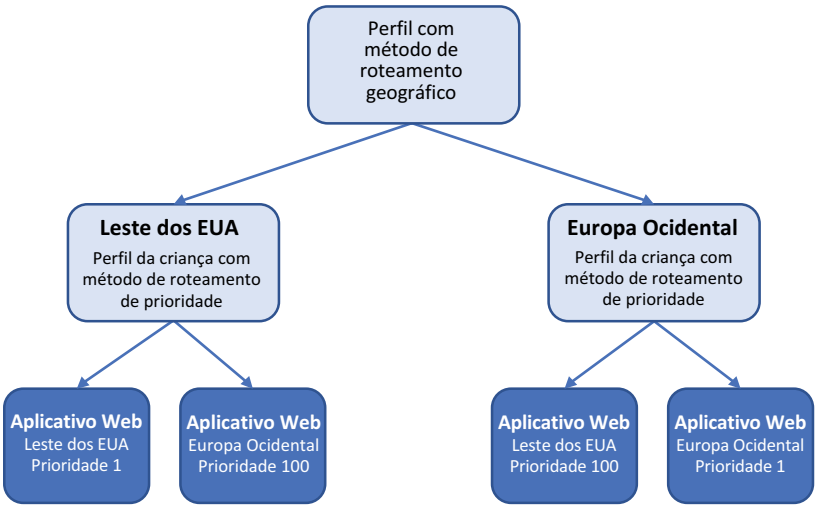


Figura 11.6 Um perfil do Gerenciador de Tráfego pai com o método de roteamento geográfico deve usar perfis filho que contenham vários pontos de extremidade. Esses pontos de extremidade filho podem usar roteamento de prioridade para direcionar sempre o tráfego para o ponto de extremidade preferencial. Por exemplo, o perfil de filho do Leste dos EUA sempre envia tráfego para o ponto de extremidade no Leste dos EUA, contanto que o ponto de extremidade seja íntegro. Se o ponto final não estiver íntegro, o tráfego será direcionado para Oeste da Europa. Sem esse perfil de filho, os clientes em Leste dos EUA não puderam fazer failover para um ponto de extremidade alternativo e não puderam acessar seu aplicativo Web.

Experimente agora

Para criar os perfis do Gerenciador de Tráfego para sua aplicação distribuída, conclua as etapas a seguir.

O restante dos exercícios usam o leste dos EUA e Europa Ocidental. Se você não vive em uma daquelas regiões, escolha uma região diferente que seja mais apropriada. Apenas lembre-se de ser consistente em todos os exercícios! O laboratório de fim de capítulo mostra como tudo isso se combina e funciona, mas você não será direcionado corretamente para seus aplicativos Web se você viver fora da América do Norte ou da Europa e não alterar as regiões de acordo.

1. Abra o portal do Azure e selecione o ícone do Cloud Shell na parte superior do painel.
2. Crie um grupo de recursos, especificando um nome de grupo de recursos, como `azuremolchapter11` e uma localização, como `eastus`:

```
az group create --name azuremolchapter11 --location eastus
```

3 Crie o perfil do Gerenciador de Tráfego pai. Você quer usar o método de roteamento geográfico e, em seguida, especifique um nome, como `azuremol`. O parâmetro para o nome DNS informa que ele deve ser exclusivo, portanto, forneça um nome exclusivo. O domínio a seguir cria o nome de host azuremol.trafficmanager.net, usado para configurar os aplicativos Web no laboratório no final do capítulo:

```
az network traffic-manager profile create \
  --resource-group azuremolchapter11 \
  --name azuremol \
  --routing-method geographic \
  --unique-dns-name azuremol
```

4 Crie um dos perfis filho do Gerenciador de Tráfego. Desta vez, use o método de roteamento prioritário e o nome eastus e especifique outro nome DNS exclusivo, como `azuremoleastus`:

```
az network traffic-manager profile create \
  --resource-group azuremolchapter11 \
  --name eastus \
  --routing-method priority \
  --unique-dns-name azuremoleastus
```

5 Crie mais um perfil filho do Gerenciador de Tráfego com o nome westeurope e outro nome DNS exclusivo, como `azuremolwesteurope`:

```
az network traffic-manager profile create \
  --resource-group azuremolchapter11 \
  --name westeurope \
  --routing-method priority \
  --unique-dns-name azuremolwesteurope
```

6 Você criou um aplicativo Web algumas vezes agora, por isso, vamos usar a CLI para criar rapidamente dois planos de serviço de aplicativo e um aplicativo Web em cada plano. Um desses aplicativos Web é em Leste dos EUA, o outro em Oeste da Europa. No laboratório de fim de capítulo, você carregará páginas da Web de exemplo para esses aplicativos Web, então, por enquanto, basta criar o site vazio e deixá-lo pronto para usar um repositório Git local.

Crie o aplicativo Web no Leste dos EUA da seguinte maneira:

```
az appservice plan create \
  --resource-group azuremolchapter11 \
  --name appserviceeastus \
  --location eastus \
  --sku S1
az webapp create \
  --resource-group azuremolchapter11 \
  --name azuremoleastus \
  --plan appserviceeastus \
  --deployment-local-git
```

7 Crie um segundo aplicativo Web em Oeste da Europa:

```
az appservice plan create \
  --resource-group azuremolchapter11 \
  --name appservicewesteurope \
  --location westeurope \
  --sku S1
```

```
az webapp create \
  --resource-group azuremolchapter11 \
  --name azuremolwesteurope \
  --plan appservicewesteurope \
  --deployment-local-git
```

11.3.2 Distribuição global de tráfego para a instância mais próxima

Você criou os perfis e os pontos de extremidade do Gerenciador de Tráfego, mas nenhum tráfego que possa fluir. Se os clientes fossem direcionados para os perfis, não haveria nenhuma associação com seus pontos de extremidade. O diagrama na Figura 11.7 mostra como você precisa associar pontos de extremidade a perfis.

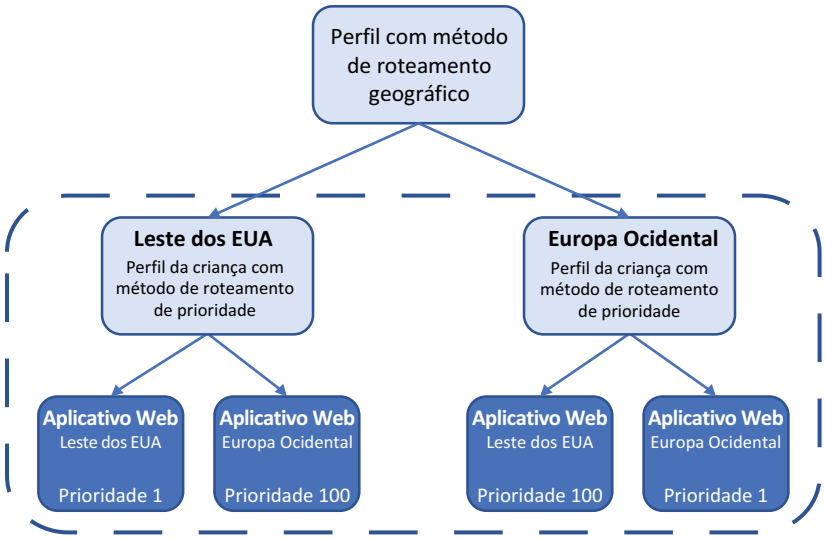


Figura 11.7 Nesta seção, você associará os pontos de extremidade aos perfis do Gerenciador de Tráfego e define a prioridade para o tráfego a ser distribuído.

As primeiras associações que você faz são para os pontos de extremidade do aplicativo Web. Lembre-se de que, para alta disponibilidade, você quer que ambos os Aplicativos Web estejam disponíveis para cada perfil do Gerenciador de Tráfego. Você usa um método de roteamento prioritário para direcionar todo o tráfego para o aplicativo Web principal para cada perfil. Se esse aplicativo Web não estiver disponível, o tráfego poderá fazer failover para o ponto de extremidade do aplicativo Web secundário.

Quando você criou os perfis do Gerenciador de Tráfego na seção 11.3.1, alguns padrões foram usados para as opções de verificação de integridade e o monitoramento de ponto de extremidade. Vamos explorar o que são essas opções:

- *Vida útil (TTL) de DNS: 30 segundos* — Define quanto tempo as respostas do Gerenciador de Tráfego podem ser armazenadas em cache. Um TTL curto garante que o tráfego do cliente seja roteado apropriadamente quando as atualizações são feitas na configuração do Gerenciador de Tráfego.

- *Protocolo de monitor de ponto de extremidade HTTP* —Você também pode escolher HTTPS ou uma verificação TCP básica. Como com balanceadores de carga, HTTP ou HTTPS garante que uma resposta HTTP 200 OK seja retornada de cada ponto de extremidade.
- *Porta: 80* — A porta a ser verificada em cada ponto de extremidade.
- *Caminho: /* — Por padrão, verifica a raiz do ponto de extremidade, embora você também possa configurar uma página personalizada, como a página de verificação de integridade usada por balanceadores de carga.
- *Intervalo de sondagem do ponto de extremidade: 30 segundos* — Com que frequência deve-se verificar a integridade do ponto de extremidade. O valor pode ser 10 segundos ou 30 segundos. Para executar a sondagem rápida a cada 10 segundos, há um custo adicional por ponto de extremidade.
- *Tolerar número de falhas: 3* — Quantas vezes um ponto de extremidade pode falhar em uma verificação de integridade antes que o ponto de extremidade seja marcado como indisponível.
- *Limite te tempo de sondagem: 10 segundos* — O período de tempo antes de uma sonda ser marcada como falha e o ponto de extremidade ser sondado novamente.

Você não precisa alterar nenhuma dessas opções padrão. Para workloads críticos quando você cria seus próprios ambientes de aplicações no mundo real, é possível reduzir o número de falhas a tolerar ou o intervalo de sondagem. Essas alterações garantirão que os problemas de integridade sejam detectados rapidamente, e o tráfego será roteado para um ponto de extremidade diferente mais cedo.

Experimente agora

Para associar pontos de extremidade a perfis e concluir o roteamento geográfico, conclua as etapas a seguir:

1. No portal do Azure, navegue e selecione seu grupo de recursos. Para este exercício, selecione o perfil do Gerenciador de Tráfego criado para Leste dos EUA.
2. Escolha Pontos de extremidade na barra de navegação à esquerda no perfil e, em seguida, selecione Adicionar.
3. Crie um ponto de extremidade do Azure e insira um nome, como `eastus`.
4. Existem diferentes tipos de recursos de destino. Você deseja usar o Serviço de Aplicativo. Para o recurso de destino, selecione seu aplicativo Web em Leste dos EUA, como azuremoleastus.
5. Deixe Prioridade definido como 1, aceite quaisquer outros padrões que possam ser definidos e selecione OK.
6. Repita o processo para adicionar outro ponto de extremidade. Dessa vez, nomeie o ponto de extremidade como `westeurope`, selecione seu aplicativo Web em Leste da Europa como o recurso de destino e defina uma prioridade de 100.

   Agora, seu perfil do Gerenciador de Tráfego agora lista dois pontos de extremidade: um para o aplicativo Web em Leste dos EUA e outro para o aplicativo Web em Oeste da Europa, como mostrado na Figura 11.8. Esse roteamento baseado em prioridade dos pontos de extremidade sempre direciona o tráfego para o aplicativo Web em Leste dos EUA quando esse recurso está íntegro. Se esse recurso não estiver disponível, há redundância para failover para o aplicativo Web em Oeste da Europa.

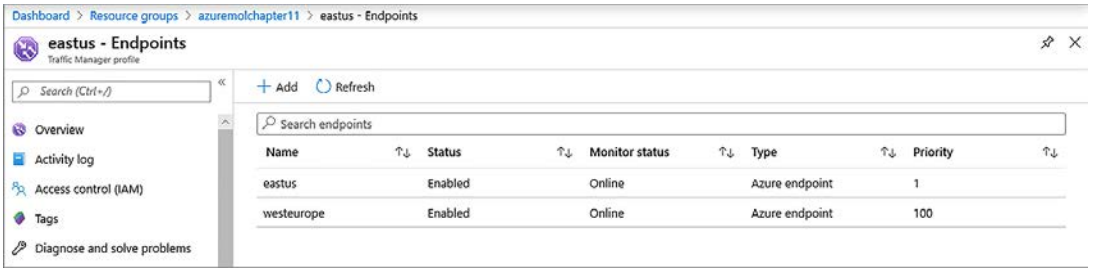


Figura 11.8 Dois pontos de extremidade são listados no perfil do Gerenciador de Tráfego. O ponto de extremidade para Leste dos EUA tem a prioridade mais baixa, portanto, ele sempre recebe o tráfego quando o ponto de extremidade está íntegro. A redundância é fornecida com o ponto de extremidade de Oeste da Europa, que é usado somente quando o ponto de extremidade do Leste dos EUA não está disponível.

7 Volte ao seu grupo de recursos e selecione o perfil do Gerenciador de Tráfego para Oeste da Europa.
8 Escolha adicionar pontos de extremidade.
9 Repita as etapas para adicionar dois pontos de extremidade e configure-os da seguinte maneira:
   - Nome: westeurope
     Recurso de destino: aplicativo Web em Oeste da Europa
     Prioridade: 1
   - Nome: eastus
     Recurso de destino: aplicativo Web em Leste dos EUA
     Prioridade: 100

Agora, seu perfil do Gerenciador de Tráfego agora lista dois pontos de extremidade: um para o aplicativo Web em Oeste dos EUA e outro para o aplicativo Web em Leste da Europa, como mostrado na Figura 11.9. Você forneceu a mesma redundância que o perfil anterior do Gerenciador de Tráfego, desta vez com todo o tráfego indo para Oeste da Europa quando íntegro, e, caso contrário, para Leste dos EUA.

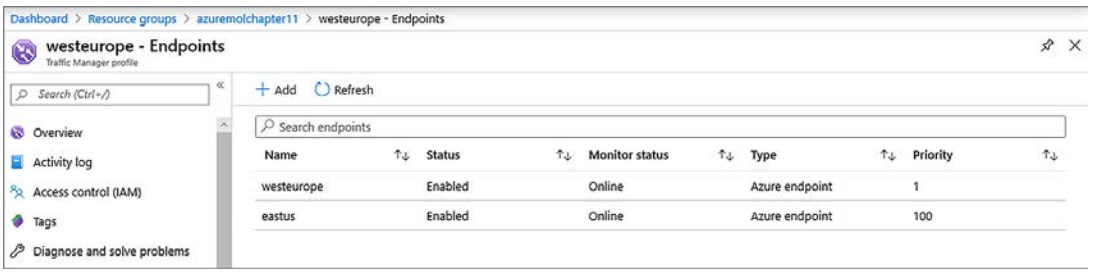


Figura 11.9 A mesma configuração de pontos de extremidade como o perfil anterior do Gerenciador de Tráfego, desta vez com o local dos aplicativos Web revertidos. Esses perfis filho podem ser usados para rotear clientes para o aplicativo Web em Leste dos EUA ou Oeste da Europa, mas agora você tem redundância para fazer failover para outro ponto de extremidade se o ponto de extremidade primário na região não estiver disponível.

Só mais uma parte para este processo, eu prometo. Lembre-se de que esta é uma prática recomendada para alta disponibilidade se você usar o Gerenciador de Tráfego para distribuição global de aplicações. No mundo real, seu ambiente pode não ser tão complexo. Veja o diagrama novamente para ver os perfis filho e as associações com os aplicativos Web regionais que você precisa criar, como mostrado na Figura 11.10.

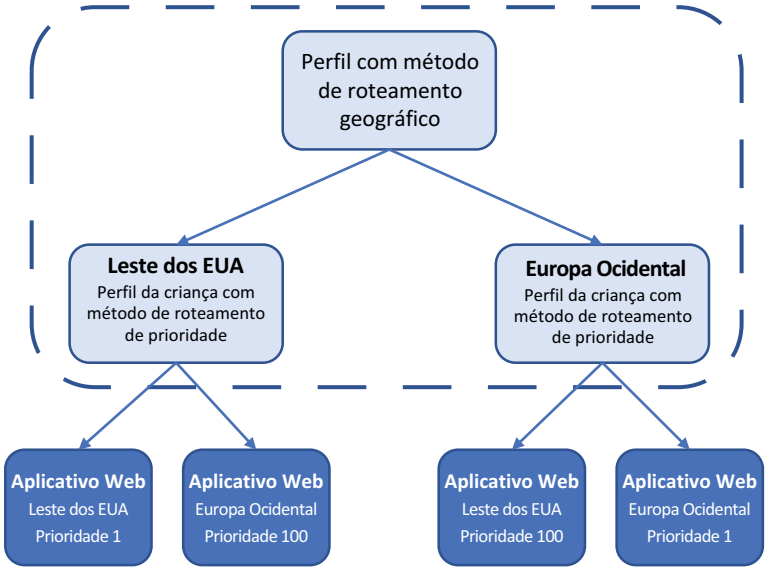


Figura 11.10 Os perfis filho do Gerenciador de Tráfego para Leste dos EUA e Oeste da Europa foram criados, com os aplicativos Web regionais e as prioridades configurados conforme necessário. Agora, você precisa associar os perfis filho com o perfil pai.

Para direcionar o tráfego com base na geografia, você define uma região, como a América do Norte, e um perfil aninhado, tal como eastus. Todos os clientes da região da América do Norte são direcionados a esse perfil filho. Você configurou as prioridades desse filho para que o aplicativo Web em Leste dos EUA sempre atenda o tráfego. Porém, você forneceu uma opção redundante para fazer o failover para o aplicativo Web em Oeste da Europa, conforme necessário.

O inverso acontece para os clientes na Europa Ocidental. Outro ponto de extremidade para o perfil pai do Gerenciador de Tráfego pode ser adicionado, desta vez com a Europa como a região a ser associada com o ponto de extremidade e, em seguida, o perfil aninhado `westeurope`. Todo o tráfego europeu é encaminhado para este perfil, e o aplicativo Web na Europa Ocidental sempre atende ao aplicativo Web. No caso de um problema, o tráfego pode fazer o failover para o leste dos EUA.

Se você tiver mandatos de política ou soberania de dados de modo que o tráfego não possa fazer failover para uma região diferente como essa, talvez seja necessário ajustar a forma como os pontos de extremidade e os perfis do Gerenciador de Tráfego são configurados. Por exemplo, você pode criar vários aplicativos Web em Oeste da Europa, como você viu no capítulo 9. Dessa forma, você tem várias instâncias do aplicativo Web que podem atender clientes. Ou, se sua aplicação é executada em VMs, use uma escala definida por trás de um balanceador de carga para perfil redundância semelhante.

Experimente agora

Este é o lugar onde a sua própria localização regional importa. Se você mora fora de um dos agrupamentos regionais exibidos nos perfis do Gerenciador de Tráfego, certifique-se de selecionar sua própria região. Caso contrário, não conseguirá acessar o aplicativo Web no laboratório de fim de capítulo.

Para associar os perfis filho ao perfil pai, conclua as etapas a seguir:

1. No portal do Azure, navegue e selecione seu grupo de recursos.
2. Selecione o perfil pai do Gerenciador de Tráfego. Nos exemplos anteriores, ele foi chamado de azuremol.
3. Escolha Pontos de extremidade na barra de navegação à esquerda no perfil e, em seguida, selecione Adicionar.
4. Crie um ponto de extremidade que usa o primeiro perfil filho. Defina o tipo como um ponto de extremidade aninhado e forneça um nome, como `eastus`. Como recurso de destino, selecione o perfil do Gerenciador de Tráfego criado para Leste dos EUA.
5. Em Agrupamento regional, escolha América do Norte/América Central/Caribe no menu suspenso e selecione OK.
6. Repita as etapas para adicionar outro ponto de extremidade. Desta vez, nomeie o ponto de extremidade como westeurope, defina recurso de destino para o perfil filho do Gerenciador de Tráfego para Oeste da Europa e escolha Europa no menu suspenso para o agrupamento regional.

   Agora, seus pontos de extremidade para o perfil pai listam os dois perfis filho, com cada um com um ponto de extremidade associado à região geográfica apropriada, conforme mostrado na Figura 11.11.

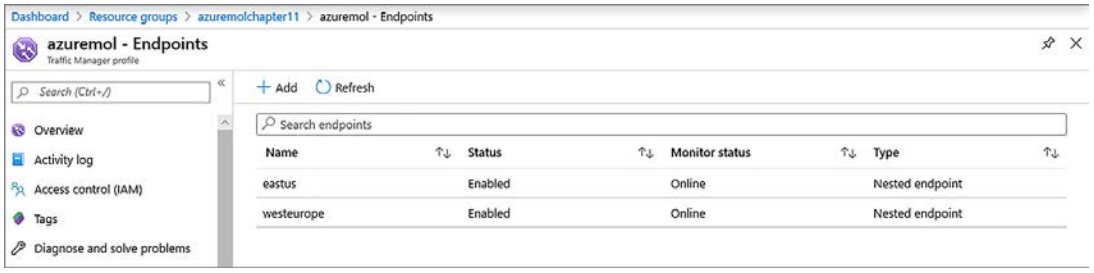


Figura 11.11 Perfis filho aninhados com regiões geográficas associadas. Este perfil pai do Gerenciador de Tráfego direciona todo o tráfego da Europa para o aplicativo Web em Oeste da Europa, com redundância para usar Leste dos EUA se houver um problema. O oposto é verdadeiro para clientes na América do Norte/América Central/Caribe.

Os aplicativos Web estão definidos para aceitar o tráfego apenas no seu domínio padrão, que está no formulário *webappname*.azurewebsites.net. Quando o Gerenciador de Tráfego direciona os clientes para essas instâncias do aplicativo Web, o tráfego parece vir do domínio do perfil pai, como azuremol.trafficmanager.net. Os aplicativos Web não reconhecem esse domínio, portanto, o aplicativo Web não será carregada.

7 Adicione o domínio do perfil pai do Gerenciador de Tráfego a ambas as instâncias criadas nas etapas 4 a 6. Se necessário, você pode encontrar o nome de domínio na página Visão geral do perfil pai do Gerenciador de Tráfego:

```
az webapp config hostname add \
  --resource-group azuremolchapter11 \
  --webapp-name azuremoleastus \
  --hostname azuremol.trafficmanager.net
az webapp config hostname add \
  --resource-group azuremolchapter11 \
  --webapp-name azuremolwesteurope \
  --hostname azuremol.trafficmanager.net
```

Agora, quando você abre o endereço do seu perfil pai do Gerenciador de Tráfego em um navegador da Web, como https://azuremol.trafficmanager.net, você não pode dizer qual endpoint você acessa, pois os dois aplicativos Web executam a mesma página da web padrão. No laboratório de fim de capítulo, você carregará uma página da Web básica para cada aplicativo Web para fazer diferenciação entre eles.

Vamos pausar para examinar o que você criou por meio destes exercícios. Isto é importante porque agora os clientes podem usar todos os recursos de alta disponibilidade e redundância de capítulos anteriores, com roteamento de tráfego automático que os direciona para a instância mais próxima da seu aplicativo Web. Neste capítulo, você criou o seguinte:

- Um aplicativo Web em Leste dos EUA e outro em Oeste da Europa.
- Perfis do Gerenciador de Tráfego que usam o roteamento geográfico para direcionar todos os clientes na América do Norte e Central para o aplicativo Web do leste em Leste dos EUA e todos os clientes na Europa para o aplicativo Web em Oeste da Europa
- Politicas do Gerenciador de Tráfego filho com roteamento prioritário para fornecer o uso de failover da região alternativa se o aplicativo Web principal para a região estiver indisponível.

Em termos de alta disponibilidade:

- Se você combinar essa configuração com aplicativos Web que têm dimensionamento automático, você tem muita redundância agora.
- Se combinar esses aplicativos Web com o Cosmos DB, você terá toda a aplicação automaticamente escalado e distribuído globalmente, com os clientes sempre acessando recursos próximos a eles para a latência mais baixa em tempos de resposta e melhor performance.
- Mesmo que você esteja preso a VMs, você pode usar conjuntos de escala com balanceadores de carga para fornecer o mesmo ambiente altamente disponível e globalmente distribuído.

E sim, você pode substituir o Gerenciador de Tráfego pelo Front Door se precisar usar recursos avançados de gerenciamento de tráfego no nível da aplicação.

Eu sei que os capítulos anteriores tiveram muitas coisas novas, e cada capítulo tem tomado praticamente todo o seu intervalo de almoço todos os dias. Porém, veja o quanto você progrediu na última semana. Agora você pode criar um aplicativo Web com VMs IaaS ou aplicativos Web de PaaS, torná-los altamente disponíveis e balanceados e deixá-los automaticamente escalado (Figura 11.12). Você pode usar um back-end do Cosmos DB globalmente distribuído para suas necessidades de banco de dados e pode rotear automaticamente os clientes para a instância regional mais próxima da sua aplicação, tudo com o DNS hospedado no Azure.

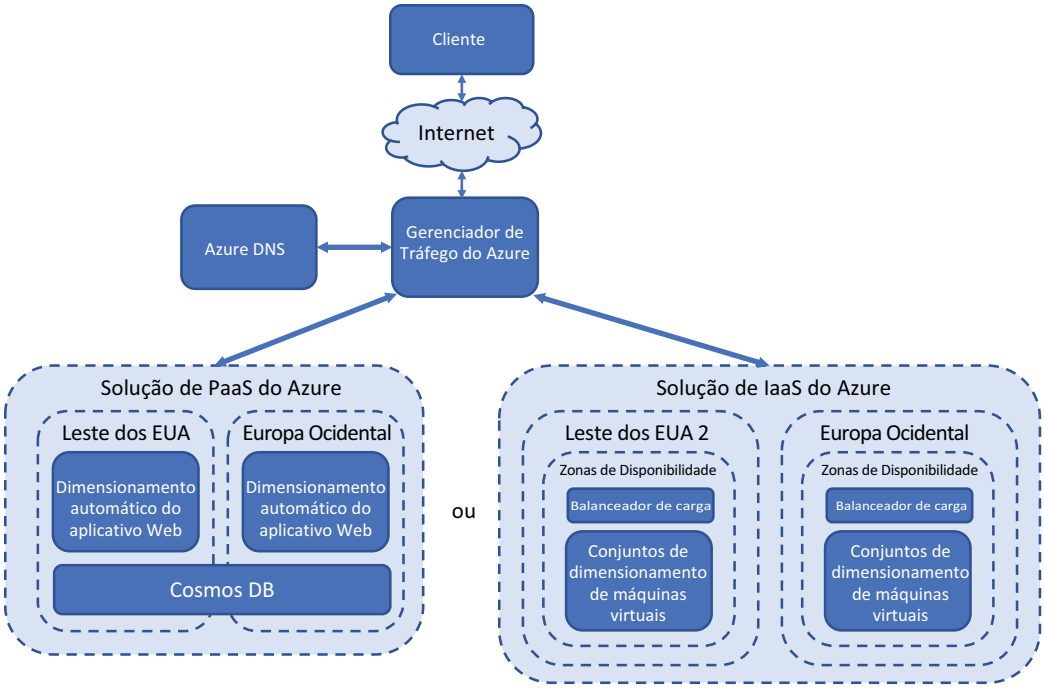


Figura 11.12 Após os capítulos anteriores, você deve entender como criar aplicações de IaaS ou PaaS altamente disponíveis no Azure. As soluções de IaaS podem usar zonas de disponibilidade, balanceadores de carga e conjuntos de escala. As soluções de PaaS podem usar aplicativos Web de dimensionamento automático e o Cosmos DB. O Gerenciador de Tráfego e o Azure DNS podem rotear os clientes para a instância de aplicação mais apropriada automaticamente, com base em sua localização geográfica.

O laboratório de fim de capítulo carrega alguns sites básicos para seus aplicativos Web, apenas para comprovar que o Gerenciador de Tráfego funciona e o ponto de extremidade apropriado atende ao seu tráfego. Se você tiver tempo, sinta-se à vontade para completar o exercício. Caso contrário, dê a si mesmo parabéns e tire uma soneca. Não vou contar para o seu chefe.

Temos mais um capítulo nesta segunda seção do livro, e ele aborda como certificar-se de que suas aplicações permaneçam íntegras: como monitorar e solucionar problemas de suas aplicações e infraestrutura.

11.4 Laboratório: Implantar aplicativos Web para ver o Gerenciador de Tráfego em ação

Este capítulo cobriu muito conteúdo, de modo que este exercício deve ser aquele que mantém o fortalecimento de suas habilidades de Azure com aplicativos Web. No repositório GitHub de exemplo do Azure, há duas páginas da Web básicas para a aplicação de pizzaria online. O título de cada página da Web mostra o local do aplicativo Web. Carregue essas páginas da Web para a instância relevante do aplicativo Web para ver os fluxos do Gerenciador de Tráfego na prática:

1. Se necessário, clone o repositório de exemplos do GitHub em seu Cloud Shell da seguinte maneira:

   ```
   git clone https://github.com/foouldsy/azure-mol-samples-2nd-ed.git
   ```

2. Comece com a página da Web eastus e, em seguida, repita as seguintes etapas no diretório westeurope:

   ```
   cd ~/azure-mol-samples-2nd-ed/11/eastus
   ```

3. Inicialize o repositório Git e adicione a página da Web básica:

   ```
   git init && git add . && git commit -m "Pizza"
   ```

4. No portal do Azure, a Visão geral do seu aplicativo Web lista a URL do clone do Git. Copie essa URL do clone do Git como um destino para seu site HTML de exemplo no Cloud Shell com o seguinte comando:

   ```
   git remote add eastus <your-git-clone-url>
   ```

5. Envie o site HTML de exemplo para seu aplicativo Web:

   ```
   git push eastus master
   ```

6. Repita essas etapas para o diretório azure-mol-samples-2nd-ed/11/westeurope.
7. Quando terminar, abra o navegador da Web para o nome de domínio do seu perfil pai do Gerenciador de Tráfego, como https://azuremol.trafficmanager.net, para ver como tráfego flui.

12 Monitoramento e solução de problemas

Nos capítulos anteriores, você aprendeu como tornar suas aplicações altamente disponíveis e rotear clientes de todo o mundo para instâncias globalmente distribuídas da sua aplicação. Um objetivo era minimizar a quantidade de interação com sua infraestrutura de aplicação e deixar a plataforma do Azure gerenciar para você a integridade e a performance. Às vezes, você ainda precisa arregaçar as mangas e rever diagnósticos ou métricas de performance. Neste capítulo, você aprenderá a revisar os diagnósticos de inicialização para uma VM, monitorar métricas de performance e solucionar problemas de conectividade com o Observador de Rede.

12.1 Diagnósticos de inicialização de VM

Com aplicativos Web, você implanta seu código e deixa que a plataforma do Azure cuide do resto. No capítulo 3, examinamos as noções básicas de como solucionar problemas e diagnosticar dificuldades com implantações de aplicativos Web. Você aprendeu como ver eventos de aplicação em tempo real para monitorar a performance. Ao trabalhar com VMs na nuvem, muitas vezes é difícil solucionar um problema quando você não pode ver fisicamente a tela do computador para que você possa obter diagnósticos do aplicativo Web.

Um dos problemas mais comuns com VMs é a falta de conectividade. Se você não pode fazer SSH ou RDP em uma VM, como pode solucionar o problema? Uma das primeiras coisas que você pode querer verificar é se a VM está sendo executada corretamente. Para ajudar, o Azure fornece diagnósticos de inicialização da VM que incluem logs de inicialização e uma captura de tela do console.

Acesso interativo ao console de inicialização

Para cenários de resolução de problemas específicos, você também pode acessar um console serial ao vivo para VMs no Azure. Este console serial permite logins interativos e resolução de problemas em caso de problema de inicialização. Você pode reconfigurar sua VM para corrigir cenários de inicialização com falha ou configurações incorretas de serviços e aplicações que impedem que sua VM inicialize corretamente.

Este capítulo não entra em cenários específicos para uso do console serial, mas é um ótimo recurso que permite que você praticamente se sente na frente da tela de uma VM na medida em que ela é inicializada. Você também precisa de diagnósticos de inicialização habilitados e, portanto, esses exercícios são pré-requisitos para o console serial.

Experimente agora

Para criar uma VM e habilitar o diagnóstico de inicialização, conclua as etapas a seguir:

1. No portal do Azure, selecione Criar um recurso, no canto superior esquerdo.
2. Pesquise e selecione uma imagem de VM do Windows Server 2019 Datacenter.
3. Crie um grupo de recursos, como `azuremolchapter12` e selecione a região do Azure mais apropriada, mais próxima a você.
4. Selecione um tamanho de VM, como DS1_v2.
5. Insira um nome de usuário para a VM, como `azuremol`, e uma senha. A senha deve ter um mínimo de 12 caracteres e conter três dos seguintes: um caractere minúsculo, um caractere maiúsculo, um número e um caractere especial.
6. Aceite as opções de redundância ou regras de porta de entrada.
7. Aceite os padrões para discos e rede. Não há nada que você precise mudar. Essas configurações já devem ser conhecidas para você.

   Uma coisa que você pode ter ignorado anteriormente foi a seção Gerenciamento. Como mostrado na Figura 12.1, a opção Diagnósticos de inicialização é habilitada por padrão e uma conta de armazenamento é criada.
8. Por enquanto, deixe a opção de diagnóstico de sistema operacional convidado desabilitada.
9. Revise suas definições de configuração de VM e selecione Criar.

Leva alguns minutos para criar e configurar a VM, por isso, vamos continuar explorando o diagnóstico de inicialização.

Se você não tiver o diagnóstico de inicialização habilitado, mas tiver um problema, provavelmente não poderá inicializar a VM para habilitar o diagnóstico. É um cenário divertido de quem veio primeiro, a galinha ou o ovo, certo? Como resultado, os diagnósticos de inicialização são ativados automaticamente para VMs criadas no portal do Azure. Para o Azure PowerShell, a CLI do Azure e os SDKs específicos de linguagem, você precisa habilitar diagnósticos de inicialização.

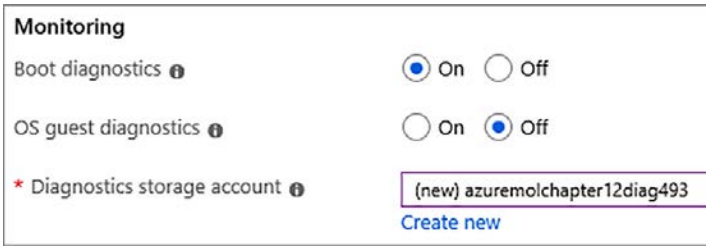


Figure 12.1 Por padrão, os diagnósticos de inicialização são habilitados quando você cria uma VM no portal do Azure. Uma conta de armazenamento é criada. O diagnóstico de inicialização é armazenado nesta conta. Em um exercício posterior, você revisará e habilitará o diagnóstico de sistema operacional convidado; portanto, não ative-os agora. Para uso de produção, recomendo que você habilite os diagnósticos de inicialização e os diagnósticos do sistema operacional convidado para cada VM criada.

Recomendo muito que você habilite o diagnóstico de inicialização em suas VMs quando criá-las. Crie o hábito de usar os modelos do Azure Resource Manager (capítulo 6) ou seus próprios scripts da CLI do Azure ou do PowerShell que permitem diagnósticos de inicialização durante a implantação.

Você precisa criar uma conta de armazenamento para os logs de inicialização e capturas de tela do console, mas o custo para armazenar esses dados provavelmente é menor do que US$ 0,01 por mês a menos que tenha uma VM ocupada que gera muitos dados. A primeira vez que você tiver um problema em uma VM e precisar de acesso ao diagnóstico de inicialização, este centavo por mês vai valer a pena. Essa conta de armazenamento também pode ser usada para armazenar métricas e logs de performance de nível de VM adicionais, que examinaremos na seção 12.2. Novamente, os custos de armazenamento devem ser mínimos. Mesmo que o seu ambiente VM cresça, vale a pena o pequeno custo adicional para você conseguir resolver rapidamente um problema quando as coisas dão errado.

Experimente agora

Para visualizar o diagnóstico de inicialização para sua VM, conclua as etapas a seguir:

1. No portal do Azure, selecione Virtual Machines, no menu à esquerda.
2. Escolha a VM que você criou no exercício anterior.
3. Na seção Suporte + solução de problemas do menu VM, escolha Diagnósticos de inicialização. O diagnóstico de inicialização e o estado da VM são exibidos, como mostrado na Figura 12.2. O relatório de integridade indica os problemas de inicialização com a VM e permite que você faça o diagnóstico da causa raiz do problema.

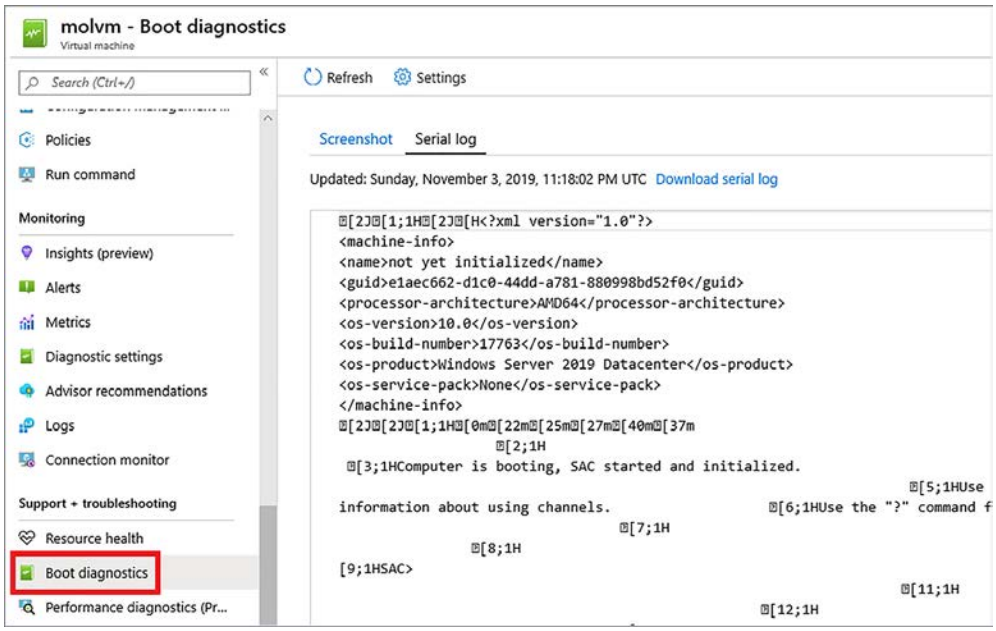


Figura 12.2 O diagnóstico de inicialização de uma VM relata a integridade e o estado da inicialização. Se forem exibidos erros, você deve ser capaz de solucionar problemas e diagnosticar a causa raiz. Você também pode fazer o download dos logs do portal para análise em seu computador local.

12.2 Métricas e alertas de performance

Uma das primeiras etapas na solução de um problema é a análise da performance. Quanta memória está disponível, quanta CPU é consumida, e quanta atividade de disco existe?

À medida em que você criar e testa suas aplicações no Azure, recomendo que você registre linhas de base de performance em vários pontos. Essas linhas de base dão a você uma ideia de como sua aplicação deve executar diferentes quantidades de carga. Por que isso é importante? Em três meses, como você pode determinar se você encontra problemas de performance sem alguns dados para comparar a performance atual?

Quando você aprendeu a escalar automaticamente as aplicações no capítulo 9, você usou métricas de performance básicas, como o uso da CPU, para informar a plataforma do Azure quando aumentar ou diminuir o número de instâncias da sua aplicação. No entanto, essas métricas básicas dão a você apenas pequenos insights de como a VM é operada. Para obter métricas mais detalhadas, você precisa examinar a performance da VM e, para fazer isso, você precisa instalar a extensão de diagnóstico do Azure.

12.2.1 Exibir métricas de performance com a extensão de diagnóstico de VM

Para adicionar funcionalidade às suas VMs, o Azure tem dezenas de extensões que você pode instalar facilmente. Essas extensões instalam um pequeno agente ou tempo de execução da aplicação na VM que, com frequência, relata as informações para a plataforma do Azure ou para soluções de terceiros. As extensões de VM podem automaticamente configurar e instalar componentes ou executar scripts em suas VMs.

A extensão de diagnóstico de VM é uma comumente usada para transmitir métricas de performance da VM para uma conta de armazenamento. Essas métricas de performance podem ser analisadas no portal do Azure ou baixadas e usadas em uma solução de monitoramento existente. Você pode usar a extensão de diagnóstico para obter uma compreensão mais profunda da performance de CPU e do consumo de memória da VM, que normalmente pode fornecer uma imagem mais detalhada e precisa do que o host.

Extensões de automação e VM

No capítulo 18, abordaremos a Automação do Azure, que permite que você execute tarefas em suas VMs de forma automatizada e agendada. Um recurso poderoso da Automação do Azure é servir como um servidor de pull de Desired State Configuration (DSC) do PowerShell. O PowerShell DSC define um determinado estado de como um sistema deve ser configurado, quais pacotes devem ser instalados, arquivos e permissões, etc. Você cria definições para a configuração desejada e as aplica a VMs ou servidores físicos. Em seguida, você pode relatar e aplicar a conformidade com essas políticas. A extensão Azure PowerShell DSC é usada para aplicar configurações de DSC, tais como de um servidor de pull da Automação do Azure.

Outras extensões que podem aplicar configurações e executar scripts em VMs incluem a Azure Custom Script Extension. Com a Custom Script Extension, você define um conjunto simples de comandos ou aponta para um ou mais scripts externos, como aqueles hospedados no Armazenamento do Azure ou no GitHub. Esses scripts podem executar tarefas complexas de configuração e instalação e garantir que todas as VMs implantadas sejam consistentemente configuradas.

A extensão Azure PowerShell DSC e a extensão Custom Script Extension são geralmente usadas com conjuntos de escala de máquinas virtuais. Você aplica uma dessas extensões ao conjunto de escala e, como as instâncias de VM são criadas dentro do conjunto de escala, elas são configuradas automaticamente para executar sua aplicação. O objetivo dessas extensões é minimizar a configuração manual necessária de VMs, que é um processo propenso a erros que requer interação humana.

Outras formas de automatizar as configurações de VM incluem Puppet e Chef, ambos com extensões de VM do Azure disponíveis. Se você já tiver uma ferramenta de gerenciamento de configuração em uso, verifique com o fornecedor da sua abordagem com suporte para uso no Azure. Há uma boa chance de uma extensão VM estar disponível para facilitar sua vida.

Experimente agora

Para habilitar a extensão de diagnóstico de VM, conclua as etapas a seguir:

1. No portal do Azure, escolha Máquinas virtuais, no menu à esquerda.
2. Escolha a VM que você criou em um exercício anterior.
3. Na seção Monitoramento, do menu VM, escolha Configurações de diagnóstico.
4. Selecione o botão para habilitar o monitoramento em nível de convidado.

Leva alguns minutos para habilitar o monitoramento em nível de convidado. Nos bastidores, veja o que o Azure faz:

- Instala a extensão de diagnóstico de VM
- Configura a extensão para transmitir métricas de nível de convidado para as seguintes áreas:
  - Disco lógico
  - Memória
  - Interface de rede
  - Processo
  - Processador
  - Sistema
  - Permite que os logs de aplicação, segurança e sistema sejam transmitidos para o Armazenamento do Azure

Quando a extensão de diagnóstico é instalada, você pode limitar quais dados são coletados, selecionando apenas determinados contadores de performance para relatar. Por exemplo, você pode desejar coletar somente o uso de memória ou habilitar a coleção de métricas do Microsoft SQL Server. Por padrão, as métricas são coletadas a cada 60 segundos. Você pode ajustar essa taxa de amostragem conforme desejado para suas aplicações e infraestrutura.

A extensão de diagnóstico da VM também pode transmitir arquivos de log de sua VM, permitindo que você centralize os logs de aplicação, segurança e sistema para análise ou alertas, como mostrado na Figura 12.3. Por padrão, os logs de aplicação e sistema que geram alertas críticos, de erros ou de avisos são registrados, juntamente com eventos de segurança de falha de auditoria. Você pode alterar os níveis de log para registro para habilitar a coleção de log do IIS e logs de aplicação e para eventos de Event Tracing for Windows (ETW). Como parte do planejamento e implantação da aplicação, determine quais logs você deseja coletar.

Não há nada exclusivo para VMs do Windows aqui. Você pode usar da mesma forma a extensão de diagnóstico em VMs Linux para obter métricas de performance e transmitir vários logs.

Se sua VM encontra um problema, muitas vezes a única maneira de analisar o que aconteceu é revisar os *crash dumps*. Canais de suporte muitas vezes solicitam esses crash dumps, se você quiser chegar à causa raiz de um problema. Como com o diagnóstico de inicialização, não há nenhuma maneira de habilitar retroativamente crash dumps para ver por que algo teve falha, para determinar se você precisa monitorar determinados processos e ser proativo sobre como configurar crash dumps. Por exemplo, você pode monitorar o processo do IIS e registrar um crash dump completo para o Armazenamento do Azure, se o processo falhar.

Aqui estão algumas outras áreas que você pode configurar para métricas de convidado:

- *Coletores* permitem que você configure a extensão de diagnóstico de VM para enviar determinados eventos para o Azure Application Insights. Com o Application Insights, você pode obter visibilidade diretamente em como seu código é executado.
- *Agente* permite que você especifique uma cota de armazenamento para todas as suas métricas. (O padrão é 5 GB.) Você também pode habilitar a coleção de logs para o próprio agente ou desinstalá-lo.

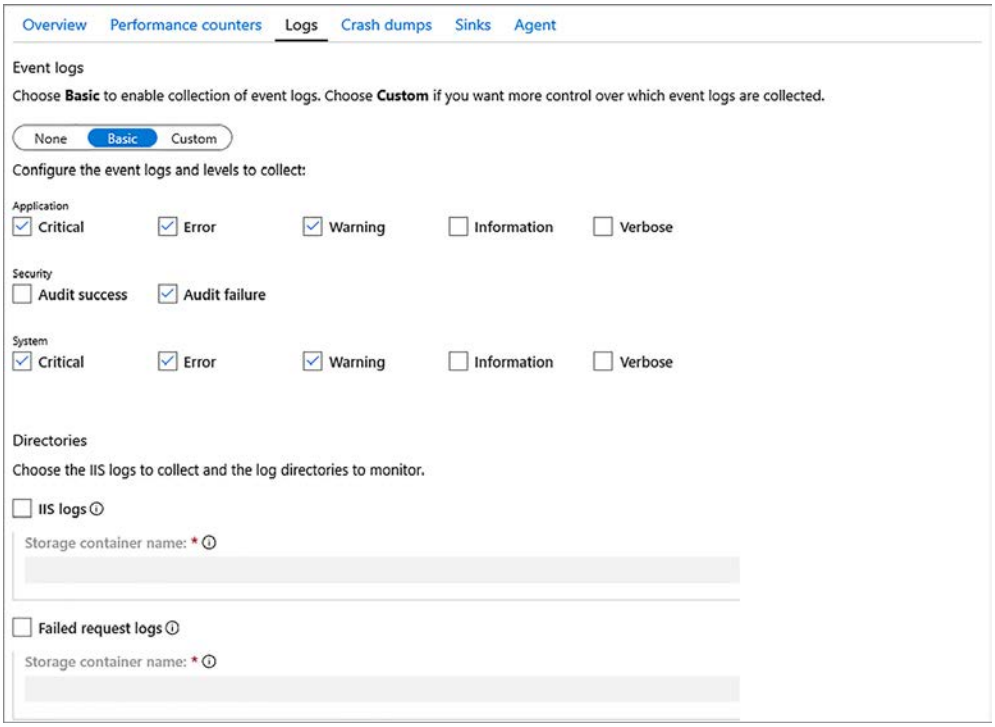


Figura 12.3 Você pode configurar eventos e níveis de log para vários componentes na VM. Esse recurso permite centralizar seus logs de VM para análise e gerar alertas. Sem a necessidade de instalar sistemas de monitoramento complexos e, muitas vezes, dispendiosos, você pode revisar e receber notificações quando surgem problemas em suas VMs do Azure.

Experimente agora

Para exibir as métricas de nível de convidado, conclua as etapas a seguir:

1. No portal do Azure, escolha Máquinas virtuais, no menu à esquerda.
2. Escolha a VM que você criou em um exercício anterior.
3. Na seção Monitoramento, do menu VM, escolha Métricas.

Muitas outras métricas já estão disponíveis, em comparação com as métricas básicas baseadas em host do capítulo 9. Explore algumas das métricas de host e convidado da VM disponíveis e pense em algumas aplicações nas quais você pode querer monitorar métricas específicas.

12.2.2 Criar alertas para condições de performance

Com sua VM configurada para expor métricas de performance de nível de convidado, como você sabe quando há um problema? Você não quer ter que ficar sentado e olhando gráficos do performance em tempo real e esperar até que um problema ocorra. Eu não sou seu chefe, se é com isso que você está preocupado. Porém, há uma maneira muito melhor: alertas de métrica.

Alertas de métricas permitem selecionar um recurso, métrica e limite e, em seguida, definir quem e como você deseja notificar quando esse limite for atingido. Alertas funcionam em mais do que apenas para VMs. Por exemplo, você pode definir alertas em endereços IP públicos que procuram a entrada distribuída de negação de serviço (DDoS) pacotes e avisá-lo quando um determinado limite é atingido que poderia constituir um ataque.

Quando os alertas são gerados, você pode optar por enviar uma notificação por email aos proprietários, colaboradores e leitores. Esses usuários e endereços de email são obtidos com base nas políticas RBAC aplicadas. Em organizações maiores, os alertas podem enviar notificações por email para um grande grupo de pessoas, então, use com cuidado. Outra opção é especificar endereços de email, que podem ser os proprietários de aplicações ou engenheiros de infraestrutura específicos ou uma lista de distribuição ou grupo direcionado para as partes diretamente envolvidas.

Existem algumas outras opções úteis para ações a serem tomadas quando um alerta é acionado:

- *Execute um runbook*. No capítulo 18, examinaremos a Automação do Azure. O serviço de Automação permite que você crie e use runbooks que executam scripts. Esses scripts podem executar uma ação de correção básica na VM, como reiniciar um processo ou até mesmo reinicializar a VM. Eles também podem executar cmdlets do Azure PowerShell para habilitar os recursos do Observador de Rede do Azure, como pacotes de captura, que exploraremos no restante deste capítulo.
- *Execute um aplicativo lógico*. Aplicativos lógicos do Azure permitem que você crie fluxos de trabalho que executam código sem servidor. Você pode escrever informações para um sistema de chamado de suporte ou iniciar uma chamada telefônica automatizada para um engenheiro de plantão. No capítulo 21, exploraremos o maravilhoso mundo da computação sem servidor com aplicativos lógicos do Azure e funções do Azure.

No laboratório de fim de capítulo, você configurará alguns alertas para sua VM. No entanto, o Azure pode fazer mais do que ajudar a solucionar problemas e monitorar suas VMs. Vamos discutir outra causa comum de erros: a rede.

12.3 Observador de Rede do Azure

As métricas de performance da VM e os diagnósticos de inicialização são ótimas maneiras de monitorar suas aplicações de IaaS do Azure. Os logs de aplicativos Web e o App Insights fornecem conscientização sobre o desempenho de suas aplicações PaaS. O tráfego de rede é geralmente menos glamouroso, mas é mais provável que seja a causa de problemas de conectividade de aplicações que você ou seus clientes encontram.

De volta ao capítulo 5, eu brinquei que a equipe de rede sempre recebe a culpa por problemas que a equipe de operações não pode explicar. Aqui é onde podemos tentar fazer amigos novamente, ou pelo menos obter alguma prova sólida de a rede ser a culpada! O Observador de Rede do Azure é um desses recursos que ajuda a reunir equipes para um agradável abraço em grupo. Com o Network Watcher, você pode monitorar e solucionar problemas usando recursos como estes:

- Capturar pacotes de rede
- Validar o fluxo de IP para NSGs
- Gerar topologia de rede

O que é ótimo sobre estas características é que colocam equipes diferentes no controle para resolver problemas. Se você criar algumas VMs e não conseguir se conectar a elas, poderá verificar se há conectividade de rede. Para desenvolvedores, se sua aplicação não pode se conectar a uma camada de banco de dados back-end, você pode examinar as regras de NSG para ver se há um problema. E os engenheiros de rede podem capturar pacotes para examinar o fluxo de comunicação completo entre hosts para uma análise mais aprofundada.

Solução de problemas adicionais de rede

O Observador de Rede funciona em conjunto com os logs e métricas de diagnóstico discutidos anteriormente no capítulo. Recursos de rede, como balanceadores de carga e gateways de aplicação também podem gerar logs de diagnóstico. Esses logs funcionam da mesma forma que os logs de aplicação e do sistema de uma VM ou um aplicativo Web. Os logs são agrupados no portal do Azure para que você determine se há erros na configuração ou nas comunicações entre hosts e aplicações.

O DNS e o Gerenciador de Tráfego também têm uma área Solução de problemas no portal do Azure. O portal orienta você com alguns erros comuns que você pode encontrar, oferece conselhos de configuração e fornece links para documentação adicional. Se todo o restante falhar, você poderá abrir uma solicitação de suporte com o Suporte do Azure.

Embora possa ser mais fácil criar implantações de aplicações grandes com modelos do Azure Resource Manager ou com os scripts da CLI do Azure ou do PowerShell, o portal do Azure tem muitas ferramentas e recursos excelentes que você pode usar quando as coisas dão errado. Principalmente com complicadas configurações de rede e políticas de segurança, alguns segundos do seu tempo para revisar a saída das ferramentas do Observador de Rede podem identificar um problema e deixá-lo resolvê-lo rapidamente. Todas essas ferramentas ajudam a melhorar a integridade geral e a experiência de suas aplicações para seus clientes.

Quais são alguns cenários em que convém usar o Observador de Rede e os recursos de resolução de problemas que ele oferece? Vamos conferir alguns problemas comuns e ver como o Observador de Rede pode ajudar.

12.3.1 Verificar os fluxos de IP

Aqui está um problema comum: os clientes não podem se conectar à sua aplicação. A aplicação funciona bem quando você se conecta do escritório, mas os clientes não podem acessar a aplicação através da internet pública. Por quê?

VPNs e ExpressRoute

As VPNs (redes virtuais privadas) do Azure fornecem comunicações seguras entre escritórios na infraestrutura local e datacenters do Azure. O Azure ExpressRoute fornece conexões privadas de alta velocidade e dedicadas de escritórios na infraestrutura local para os datacenters do Azure e é frequentemente usado em grandes organizações.

Ambas as conexões são um pouco mais complicadas para configurar do que podemos cobrir em um único intervalo para o almoço, e, muitas vezes, são coisas que você configurar uma única vez. A equipe de rede é geralmente responsável por configurá-los, e você pode nem mesmo perceber que você acessa o Azure por uma conexão privada.

Todos os testes da sua aplicação funcionam muito bem. Você pode acessar a aplicação por meio de um navegador da Web, fazer pedidos e receber notificações por email. Porém, quando seus clientes tentam fazer um pedido, a aplicação não carrega.

Como o Observador de Rede pode ajudar? Por verificar os fluxos de IP. O Observador de Rede simula o fluxo de tráfego para o seu destino e relata se o tráfego pode alcançar sua VM.

Experimente agora

Para habilitar o Observador de Rede e verificar os fluxos de IP, conclua as etapas a seguir:

1. No portal do Azure, escolha Todos os Recursos, na parte superior do menu de navegação à esquerda.
2. Filtre e selecione o Observador de Rede na lista de serviços disponíveis. Você habilita o Network Watcher na(s) região(ões) que pretende monitorizar. Quando você habilita o Observador de Rede em uma região, o Azure usa controles de acesso baseados em função para os vários recursos e o tráfego de rede.
3. Expanda a lista de regiões para sua conta. Algumas regiões já podem estar habilitadas. Se a região em que sua VM foi implantada não estiver habilitada, selecione a região e, em seguida, habilite o Observador de Rede.
4. Quando o Observador de Rede estiver habilitado em uma região (leva um ou dois minutos), selecione Fluxo de IP Verifique em Ferramentas de diagnóstico de rede no lado esquerdo da janela Observador de Rede.
5. Selecione o grupo de recursos, como azuremolchapter12, e a VM, como molvm. Por padrão, o protocolo é definido como TCP, e Direção como Entrada. O endereço IP local da NIC virtual também é preenchido.
6. Em Porta local, insira a porta 80. Se você aceitou os padrões quando criou a VM no exercício anterior, você não abriu a porta 80 e, portanto, este é um bom teste do que acontece quando o tráfego é negado.
7. Em Endereço IP remoto, insira `8.8.8.8`. Esse endereço pode parecer familiar: é um servidor DNS aberto fornecido pelo Google. Você não está fazendo nada com este servidor; você só precisa fornecer ao Observador de Rede um endereço IP externo para simular o fluxo de tráfego. Você também pode acessar https://whatsmyip.com e digitar seu endereço IP público real.
8. Defina Porta remota para a porta 80 e, em seguida, selecione Verificar.

O resultado da sua verificação de fluxo de IP deve ser `Acesso negado`. O Observador de Rede deve informar qual regra causou a falha do fluxo de tráfego: a regra DenyAllInBound. Você sabe que há uma regra de segurança de rede que bloqueia o tráfego, mas onde esta regra é aplicada? Na sub-rede, no NIC virtual ou no grupo de segurança da aplicação? Outro recurso do Observador de Rede que pode informar isso a você.

12.3.2 Exibir regras de NSG efetivas

As regras de NSG podem ser aplicadas a uma única NIC virtual, no nível de sub-rede ou em um grupo de VMs em um grupo de segurança de aplicações. As regras são combinadas, o que permite especificar um conjunto comum de regras em uma

sub-rede inteira e, em seguida, ser mais granular para grupos de segurança de aplicações (como "Permitir a porta TCP 80 em todos os servidores Web") ou uma VM individual.

Aqui estão alguns exemplos comuns de como as regras de NSG podem ser aplicadas:

- *Nível da sub-rede* — Permitir a porta TCP 5986 para gerenciamento remoto seguro da sub-rede de gerenciamento 10.1.10.20/24.
- *Nível do grupo de segurança de aplicações* — Permitir a porta TCP 80 para o tráfego HTTP para aplicativos Web e aplicar o grupo de segurança de aplicações a todas as VMs de aplicativos Web.
- *Nível de NIC virtual* — Permitir a porta TCP 3389 para acesso de área de trabalho remota da sub-rede de gerenciamento 10.1.10.20/24.

Estas regras são básicas, e elas permitem tráfego determinado de forma explícita. Se nenhuma regra de *permissão* correspondem a um pacote de rede, regras padrão *DenyAll* serão aplicadas para descartar o tráfego.

Durante o teste da aplicação discutido no exemplo, você pode ter configurado essa regra HTTP para permitir apenas o tráfego de uma de suas sub-redes na infraestrutura local. Agora, os clientes que acessam pela internet pública não podem se conectar.

Experimente agora

Para determinar onde uma regra de NSG é aplicada, conclua as etapas a seguir:

1. No Observador de Rede, selecione Regras de segurança efetivas à esquerda.
2. Selecione o grupo de recursos, como azuremolchapter12, e sua VM, como molvm. Leva alguns segundos para que as regras efetivas sejam exibidas, como mostrado na Figura 12.4.

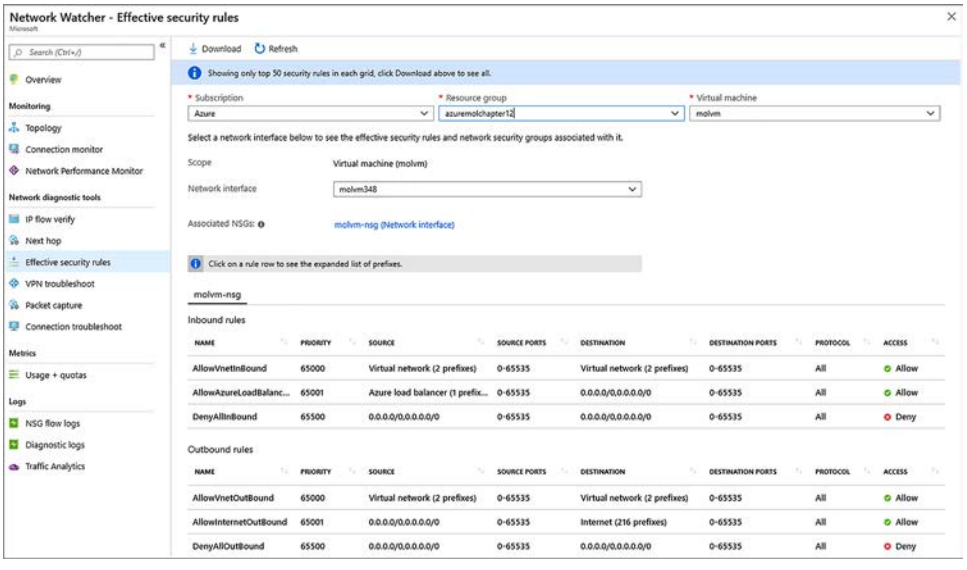


Figura 12.4 Quando você seleciona uma VM, o Observador de Rede examina como todas as regras de NSG são aplicadas e a ordem de precedência e mostra quais regras efetivas estão aplicadas no momento. Em seguida, você pode rapidamente fazer uma busca detalhada na sub-rede, regras de NIC virtual e padrão para localizar e editar onde uma determinada regra é aplicada.

As regras padrão da VM que você criou anteriormente não são empolgantes, mas você pode percorrer a sub-rede, interface de rede e regras padrão para ter uma noção de como as regras efetivas são combinadas e como você pode identificar onde as regras são aplicadas se você precisa fazer alterações.

12.3.3 Capturar pacotes de rede

Vamos supor que você atualizou suas regras de segurança de rede para permitir o acesso à sua aplicação para clientes de internet pública, mas um cliente relata que tem um comportamento incomum. Às vezes, o aplicativo Web não carrega ou apresenta imagens quebradas. Sua conexão muitas vezes parece estar fora do ar.

Problemas intermitentes são muitas vezes os mais difíceis de solucionar, principalmente se você limitou, ou não, o acesso ao computador que encontra um problema. Uma abordagem comum de resolução de problemas é capturar os pacotes de rede e analisá-los para obter sinais de problemas, como erros de transmissão de rede, pacotes malformados ou problemas de protocolo e comunicação.

Com capturas de pacotes de rede, você recebe o fluxo bruto de dados entre dois ou mais hosts. É uma arte analisar capturas de rede, e não é para os fracos de coração. Ferramentas especiais de terceiros, como o Wireshark, da Riverbed, o Fiddler, da Telerik, e o Message Analyzer

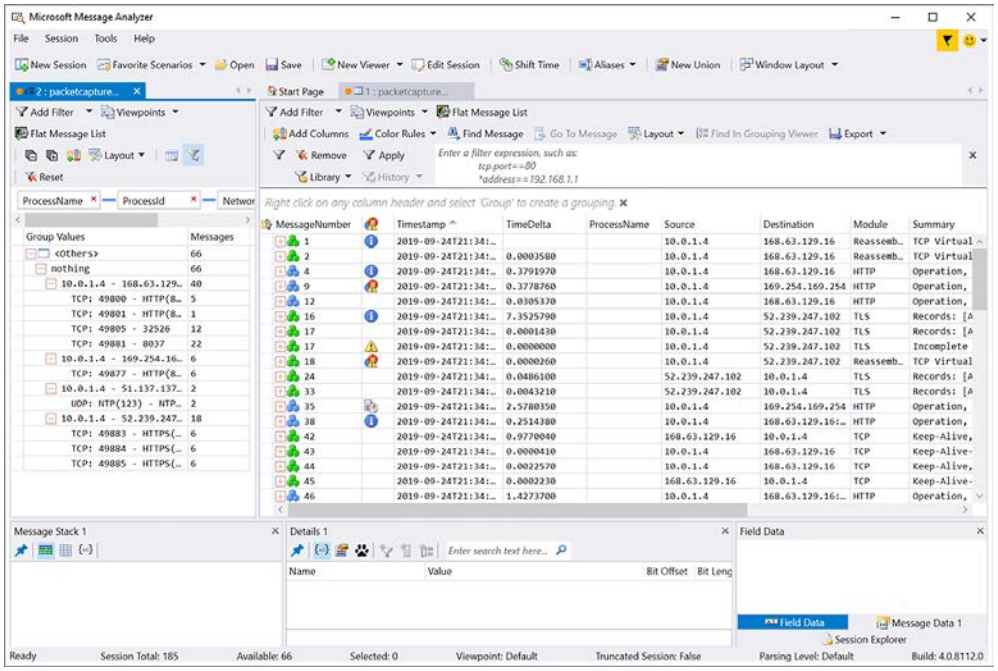


Figura 12.5 Uma captura de pacotes de rede visualizada no Message Analyzer da Microsoft. Cada pacote individual está disponível para inspeção. Você pode agrupar e filtrar por protocolo de comunicação ou cliente-host. Essa profundidade de dados de rede permite que você examine os pacotes reais que fluem entre os nós para solucionar problemas onde ocorre um erro. Um ex-colega me disse uma vez: "Os pacotes nunca mentem." O enigma é descobrir o que os pacotes dizem.

da Microsoft, fornecem uma maneira gráfica para você exibir e filtrar os pacotes de rede, normalmente agrupando-os por comunicações ou protocolos relacionados. A Figura 12.5 mostra um exemplo de captura de pacotes de rede.

Para habilitar o Observador de Rede para capturar pacotes de e para suas VMs, primeiro instale a extensão da VM do Observador de Rede. Como você viu na seção 12.3.2, as extensões de VM fornecem uma maneira para a plataforma do Azure alcançar o interior de uma VM para executar várias tarefas de gerenciamento. No caso da extensão da VM do Observador de Rede, ele examina o tráfego de rede de e para a VM.

Experimente agora

Para instalar a extensão da VM do Observador de Rede e capturar pacotes de rede, conclua as etapas a seguir.

1. No portal do Azure, escolha Máquinas virtuais no menu à esquerda e selecione sua VM, como molvm.
2. Na categoria Configurações à esquerda na janela VM, selecione Extensões.
3. Escolha Adicionar uma extensão.
4. Na lista de extensões disponíveis, escolha Agente do Observador de Reed para Windows e, em seguida, selecione Criar.
5. Para confirmar a instalação da extensão, selecione OK. Pode levar alguns minutos para que o Agente do Observador de Rede seja instalado em sua VM.
6. Para voltar ao menu do Observador de Rede no portal do Azure, escolha Todos os recursos na parte superior do menu de navegação Serviços à esquerda no portal e escolha Observador de Rede.
7. Na seção Ferramentas de diagnósticos de rede à esquerda na janela do Observador de Rede, selecione Captura de pacotes e escolha Adicionar uma nova captura.
8. Selecione o grupo de recursos, como azuremolchapter12, e a VM, como molvm. Em seguida, insira um nome para sua captura de pacotes, como `molcapture`.

   Por padrão, as capturas de pacotes são salvas no Armazenamento do Azure. Você também pode escolher salvar em arquivo e especificar um diretório local na VM de origem. A extensão do Agente do Observador de Rede, em seguida, grava o arquivo de captura de pacotes no disco na VM.
9. Se ele ainda não estiver selecionado, escolha o nome da conta de armazenamento que começa com o nome do seu grupo de recursos, como azuremolchapter12diag493. Essa é a conta de armazenamento criada e usada pela extensão de diagnóstico da VM que você habilitou anteriormente.
10. Você pode especificar um tamanho máximo para cada pacote (o padrão é 0 para o pacote inteiro), o tamanho máximo do arquivo para a sessão de captura de pacotes (o padrão é 1 GB) e o limite de tempo para a captura de pacotes (o padrão é 5 horas). Para capturar somente o tráfego de fontes ou portas

específicas, você também pode adicionar um filtro para restringir o escopo de suas capturas de pacotes.

11 Defina um limite de tempo de 60 segundos.

12 Para iniciar a captura de pacotes, selecione OK.

Leva um minuto ou dois para iniciar a captura. Quando a captura estiver em andamento, os dados são transmitidos para a conta do Armazenamento do Azure ou o arquivo local na VM. A lista de capturas é mostrada na página do portal do Observador de Rede. Se você transmitir os logs para o Armazenamento do Azure, poderá fazer com que a captura vá direto para a conta de Armazenamento e fazer o download do arquivo de captura .cap. Em seguida, você pode abrir a captura de pacote em um programa de análise, como discutido na seção 12.3.3. Na verdade, a captura de rede de exemplo mostrado na Figura 12.5 anteriormente neste capítulo era de uma captura de um pacotes do Observador de Rede do Azure.

12.4 Laboratório: Criar alertas de performance

Espero que os recursos do Observador de Rede, diagnóstico e métricas abordados neste capítulo tenham dado a você alguns insights sobre o que está disponível no Azure para solucionar problemas de aplicações. Algumas coisas, como diagnósticos de inicialização e a extensão de diagnóstico de VM, fazem mais sentido quando você os habilita e configura ao implantar VMs.

Neste laboratório, você configura alguns alertas de métrica para ver sobre o que você pode ser notificado e como são os alertas ao recebê-los:

1 No portal do Azure, navegue até a VM criada nos exercícios anteriores.

2 Na seção Monitoramento da VM, selecione Alertas.

3 Escolha criar uma regra de alerta e, em seguida, adicione uma condição para quando a porcentagem da CPU for maior do que uma média de 10% nos últimos 5 minutos. Um gráfico deve mostrar quais são as métricas mais recentes, por isso, ajuste o limite se 10% não disparar um alerta.

4 Adicione um grupo de ação e dê a ele um nome e um nome curto. Para este laboratório, defina os dois nomes como azuremol. Os grupos de ação permitem definir conjuntos reutilizáveis de etapas a realizar quando um alerta é gerado, como enviar emails para um conjunto de usuários ou executar um script do PowerShell automatizado ou um aplicativo lógico do Azure.

5 Explore os tipos de ação disponíveis e selecione Email/SMS/Push/Voz.

6 Escolha como deseja ser notificado, como por meio de email ou mensagem de texto. Algumas cobranças de operadora podem ser aplicadas para notificações por SMS ou voz.

7 Quando o grupo de ação for criado, forneça um nome ao alerta e, em seguida, especifique uma gravidade. Essa gravidade é útil quando você tem muitos alertas definidos para ajudá-lo a fazer a triagem e priorizar o que resolver primeiro.

8 Crie a regra quando estiver pronto. Leva de 10 a 15 minutos para que a regra fique ativa e gere as notificações que você definiu.

Esse exemplo é básico, então pense em alertas e notificações existentes que você tenha para aplicações e serviços e como você pode usar esse recurso ao executar workloads no Azure.

Parte 3

Seguro por padrão

Em um mundo online em que as aplicações costumam ser conectadas à Internet dia e noite, a ameaça de um ataque digital é muito real. Esses ataques custam tempo, dinheiro e a confiança do cliente. Uma parte central da criação de aplicações altamente redundantes e distribuídas inclui protegê-los e proteger seus dados. O Azure tem vários recursos incorporados para proteger seus dados, incluindo criptografia, monitoramento, cofre de chaves digitais e backups. Nesta parte do livro, você aprende a proteger e manter seguros suas aplicações desde o início.

13 Backup, recuperação e replicação

Os próximos capítulos introduzem alguns dos principais recursos e serviços do Azure que permitem que você forneça segurança para suas aplicações. Isso é provavelmente muito subjetivo: a segurança não deve ser um recurso ou uma consideração complementar. Em vez disso, a segurança deve ser incorporada de forma inerente no cerne da aplicação desde o início. Neste capítulo, você iniciará sua jornada na segurança do Azure com o backup e a recuperação de seus dados. Backups podem não parecer um tópico de segurança comum, mas pense sobre segurança como mais do que criptografia de dados ou certificados SSL do site. E quanto à proteção de seus dados contra interrupções, perda de dados e hackers? Uma discussão de backups e replicação também é um bom tópico para conectar o capítulo sobre alta disponibilidade e este capítulo.

Backups podem parecer algo trivial e, como ex-administrador de backup, posso dizer que não há nada de muito emocionante sobre trabalhos de backup e rotações. Mas backups oportunos que funcionam são cruciais para proteger suas aplicações e garantir que, na pior das hipóteses, você possa restaurar seus dados de forma rápida e confiável. Você também pode replicar suas VMs de uma região do Azure para outra. Essa capacidade baseia-se nos conceitos de alta disponibilidade que vimos anteriormente no capítulo 7.

Neste capítulo, você aprenderá como fazer backup e restaurar VMs e, em seguida, replicar VMs automaticamente no Azure. Todos esses backups e pontos de restauração são criptografados para proteger seus dados.

13.1 Backup do Azure

Uma das coisas interessantes sobre o Backup do Azure é que ele é um serviço e um grande bucket de armazenamento para os backups reais. O Backup do Azure pode proteger VMs no Azure, VMS na infraestrutura local ou servidores físicos e até mesmo VMs em outros provedores como a Amazon Web Services (AWS). Os backups de dados podem ser armazenados em suas próprias matrizes de armazenamento na infraestrutura local ou em um cofre de recuperação do Azure. A Figura 13.1 mostra como o serviço do Backup do Azure pode proteger e orquestrar todas as suas necessidades de backup.

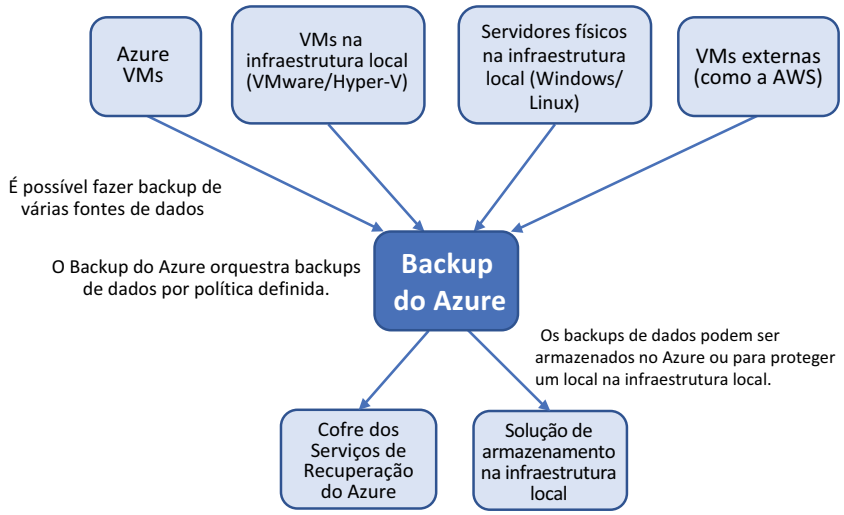


Figura 13.1 Várias VMs ou servidores físicos, de vários provedores e locais, podem ser copiados pelo serviço de orquestração central. O Backup do Azure usa políticas definidas para fazer backup de dados com uma determinada frequência ou agenda. Esses backups podem ser armazenados no Azure ou em uma solução de armazenamento na infraestrutura local. Os dados são inteiramente criptografados para maior segurança.

Basicamente, o Backup do Azure gerencia agendas de backup e retenção de dados e organiza os trabalhos de backup ou restauração. Para fazer backup de VMs do Azure, não há nenhum componente de servidor a ser instalado e nenhum agente para se instalar manualmente. Todas as operações de backup e restauração são incorporadas à plataforma do Azure.

Para fazer backup de VMs na infraestrutura local ou de servidores físicos ou VMs em outros provedores, como a AWS, você instala um pequeno agente que habilita a comunicação segura com o Azure. Essa comunicação segura garante que seus dados sejam criptografados durante a transferência.

Para dados armazenados no Azure, os backups são criptografados com uma chave de criptografia que você cria. Só você tem acesso a esses backups criptografados. Você também pode fazer backup de VMs criptografadas (que veremos no capítulo 14) para garantir que seus backups de dados realmente estejam seguros.

Não há cobrança de fluxo de tráfego de rede para fazer backup ou restaurar dados. Você paga somente por cada instância protegida e pelo armazenamento consumido no Azure. Se você usar um local de armazenamento na infraestrutura local, o custo para usar o Backup do Azure é mínimo, porque não há nenhum custo de tráfego de rede ou armazenamento do Azure.

13.1.1 Políticas e retenção

O Backup do Azure usa um modelo de backup incremental. Quando você protege uma instância, a primeira operação de backup executa um backup completo dos dados. Depois disso, cada operação de backup executa um backup incremental dos dados. Cada um desses backups é chamado *ponto de recuperação*. Backups incrementais são uma abordagem com eficiência de tempo que otimiza o uso de armazenamento e largura de banda de rede. Somente os dados que foram alterados desde o backup anterior são transferidos com segurança para o local de backup de destino. A Figura 13.2 detalha como os backups incrementais funcionam.

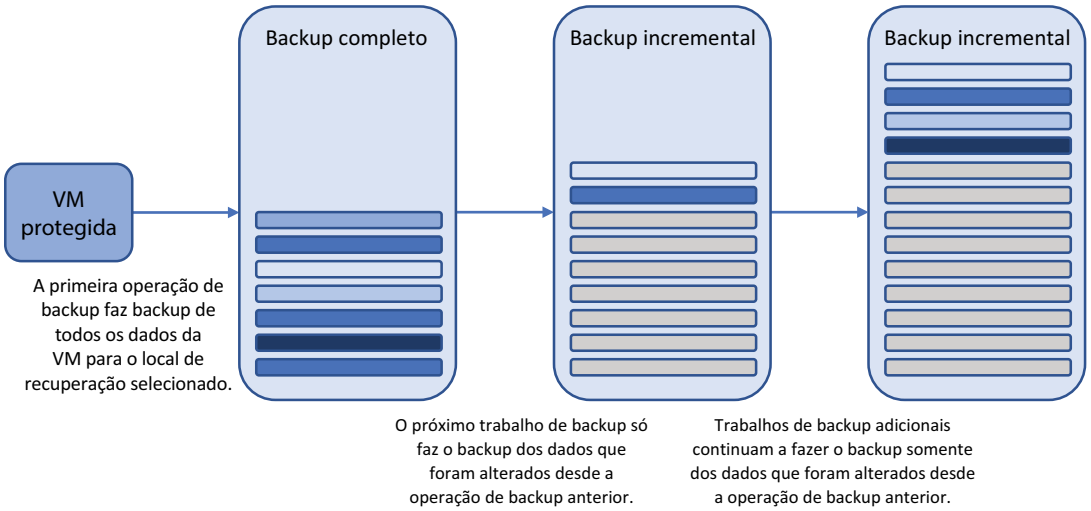


Figura 13.2 Backups incrementais fazem backup somente dos dados que foram alterados desde a operação anterior. O primeiro backup é sempre um backup completo. Cada tarefa de backup subsequente só faz backup dos dados alterados desde a tarefa anterior. Você controla a frequência de backups completos com políticas. Essa abordagem minimiza a quantidade de dados que precisam trafegar com segurança pela rede e ser alojados no local de armazenamento de destino. O Backup do Azure mantém as relações de backups incrementais entre si para garantir que, ao restaurar dados, eles sejam consistentes e completos.

Com o Backup do Azure, você pode armazenar até 9.999 pontos de recuperação para cada instância protegida. Para oferecer algum contexto, se você fizer um backup diário regular, poderá fazer isso por mais de 27 anos. E você poderia manter backups semanais por quase 200 anos. Acho que isso cobriria a maioria das situações de auditoria. Você pode optar por reter backups diária, semanal, mensal ou anualmente, o que está de acordo com a maioria das políticas de backup existentes.

Para implementar a estratégia de backup ideal para seu workload, você precisa entender e determinar seu objetivo de *ponto de recuperação* (RPO) e *objetivo de tempo de recuperação* (RTO) aceitáveis.

OBJETIVO DE PONTO DE RECUPERAÇÃO

O RPO define o ponto que seu backup mais recente permite que você restaure. Por padrão, o Backup do Azure faz um backup diário. Em seguida, você define políticas de retenção quanto a quantos dias, semanas, meses ou anos deseja manter esses pontos de recuperação. Embora o RPO seja tipicamente usado para definir a quantidade máxima de perda de dados aceitável, você também deve considerar o quão longe de volta no tempo deseja ir. A Figura 13.3 mostra como o RPO define a quantidade de perda aceitável de dados.

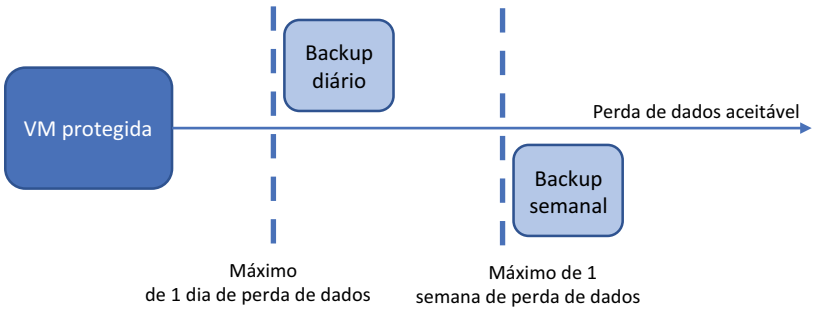


Figura 13.3 O RPO define a quantidade de perda de dados que você pode sustentar para uma instância protegida. Quanto mais tempo o RPO, maior a perda aceitável de dados. Um RPO de um dia significa que até 24 horas de dados podem ser perdidos, dependendo de quando a perda de dados ocorreu em relação ao último backup. Um RPO de uma semana significa que até sete dias de dados podem ser perdidos.

Grandes interrupções e grandes quantidades de perda de dados são ocorrências raras. Mais comuns são pequenos incidentes de perda ou substituições de dados. Esses incidentes muitas vezes não são observados ou reportados até algum tempo depois que a perda de dados tenha ocorrido. É aqui que a política de retenção das instâncias protegidas torna-se importante. Se você tiver uma política de retenção curta, talvez não seja possível restaurar dados do ponto exigido. Você precisa determinar um equilíbrio entre reter vários pontos de recuperação e os custos de armazenamento para reter todos os pontos de recuperação.

O armazenamento do Azure é relativamente barato: normalmente menos de US$ 0,02 por gigabyte de armazenamento. Esse custo equivale a aproximadamente US$ 2 por mês para um backup de dados de VM de 100 GB (além de uma cobrança para o serviço do Backup do Azure). Dependendo do quanto seus dados mudam, o tamanho dos pontos de recuperação incremental pode aumentar rapidamente. Reter pontos de recuperação por semanas ou meses pode custar dezenas de dólares por mês por instância protegida. Isso não é para desencorajá-lo, mas é importante planejar suas necessidades e fazer um uso inteligente dos seus custos. O Armazenamento parece barato com menos de US$ 0,02 por gigabyte até que você tenha centenas de gigabytes por instância protegida e dezenas ou até mesmo centenas de instâncias para proteger.

Eu sou um ex-administrador de backup, e capacidade de armazenamento era muitas vezes um fator central quando eu determinava a quantidade de pontos de recuperação para retenção. Essa capacidade de armazenamento frequentemente criava concessões com esses RPOs. Se você usar o Armazenamento do Azure em vez de uma solução de armazenamento na infraestrutura local, não precisará se preocupar com a capacidade de armazenamento disponível. Eu posso tudo, menos garantir que haja mais armazenamento do que o limite do seu cartão de crédito.

OBJETIVO DO TEMPO DE RECUPERAÇÃO

O RTO dita a rapidez com que você pode restaurar seus dados. Se você optar por fazer backup de VMs do Azure e armazenar os pontos de recuperação em uma solução de armazenamento na infraestrutura local, levará muito mais tempo para restaurar esses backups do que se estivessem alojados diretamente no Armazenamento do Azure. O inverso seria verdadeiro se você fizesse backup de VMs ou servidores físicos na infraestrutura local para o Armazenamento do Azure. A Figura 13.4 descreve o RTO.

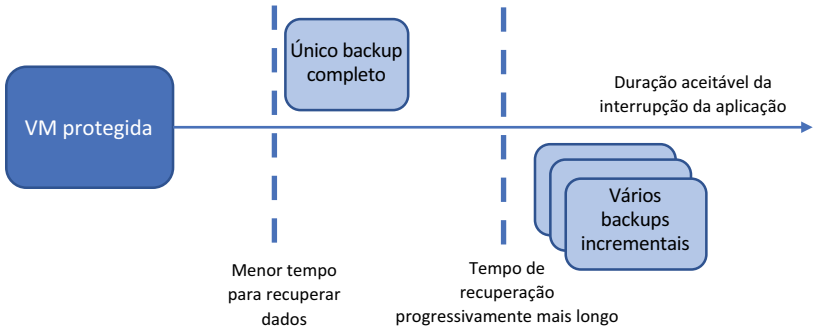


Figura 13.4 O RTO define quanto tempo é aceitável para o processo de restauração de dados ser executado e a aplicação estar indisponível. Quanto mais pontos de recuperação estiverem envolvidos no processo de restauração, mais longo será o RTO. De maneira semelhante, quanto mais próximo o armazenamento de backup estiver no ponto de restauração, menor será o RTO.

Em qualquer cenário, os dados de ponto de recuperação precisariam ser transferidos do local de armazenamento do ponto de recuperação para o local de restauração. Para grandes operações de restauração, em que você pode precisar transferir centenas de gigabytes, sua largura de banda de rede torna-se um gargalo real que controla a rapidez com que você pode tornar as aplicações disponíveis novamente.

O mesmo é verdadeiro para políticas de retenção longa com muitos pontos sucessivos de recuperação incremental. A restauração dos dados pode exigir que vários pontos de recuperação sejam montados e restaurados. Seu trabalho é determinar o quão longe de volta no tempo você precisa ser capaz de ir, e quanto tempo você pode demorar para restaurar os dados.

O RPO e o RTO variam de acordo com suas aplicações e uso comercial. Uma aplicação que processa pedidos em tempo real não pode tolerar uma paralisação ou tempo de inatividade muito longos e, portanto, o RPO e o RTO provavelmente serão muito baixos. Normalmente, você usa um banco de dados para armazenar seus dados, e normalmente cria tolerâncias na aplicação em vez de depender de pontos de recuperação.

Considerando o Cosmos DB, não há nada para se fazer backup: a plataforma do Azure executa a replicação e a proteção de dados para você. Se você criou uma solução personalizada no MySQL ou Microsoft SQL Server, normalmente usará um tipo semelhante de cluster e replicação para garantir que existam várias cópias do banco de dados e, portanto, a perda de uma instância não exigirá a restauração de um backup. Os backups principalmente protegem contra uma falha grave ou corrupção de dados.

13.1.2 Agendas de backup

Como você controla a frequência de seus backups e a retenção dos pontos de recuperação? No Backup do Azure, essas configurações são definidas em políticas. Você cria essas políticas para abranger os vários cenários nos quais quer se proteger e pode reutilizar as políticas para várias instâncias protegidas.

Por exemplo, uma política de backup pode definir que você deseja fazer um backup às 18h30 todo dia. Você deseja manter backups diários por seis meses e variá-los para reter backups semanais por dois anos. Para fins de conformidade, você retém backups mensais por cinco anos. Um backup anual é retido por 10 anos. Esses valores de retenção podem parecer excessivos, mas para uma aplicação que envolve comunicação e mensagens, você geralmente precisa reter backups para fins regulatórios e de conformidade para esses prazos longos. O Backup do Azure fornece a flexibilidade para definir políticas para atender a diferentes workloads da aplicação e aplicar rapidamente a conformidade.

Experimente agora

Todos os seus backups do Azure são armazenados em um cofre de Recovery Services. Para criar um cofre e uma política de backup, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione Criar um Recurso no menu no canto superior esquerdo.
2. Pesquise e selecione Backup e Site Recovery,, em seguida, escolha Criar.
3. Crie um grupo de recursos, como `azuremolchapter13` e, em seguida, insira um nome para o cofre, como `azuremol`.
4. Selecione um local e, em seguida, revise e crie o cofre.
5. Quando o cofre for criado, escolha Grupo de recursos no menu à esquerda no portal e, em seguida, escolha o grupo de recursos criado.
6. Selecione seu cofre dos Serviços de Recuperação na lista de recursos disponíveis, escolha Políticas de backup no menu à esquerda e escolha adicionar uma política.
7. Selecione o tipo de política de Máquina Virtual do Azure e forneça um nome para sua nova política, como `molpolicy`. Por padrão, um backup é criado diariamente.
8. Escolha o fuso horário mais apropriado no menu suspenso. Por padrão, o Azure usa o Horário Coordenado Universal (UTC).

   Se desejar, revise e ajuste as políticas de retenção para diário, semanal, mensal e anual. A seção sobre os conceitos de agendas de backup e retenção detalhou como você selecionaria esses valores. Esses valores normalmente variam à medida que você cria e aplica políticas de backup para proteger suas instâncias de VM.
9. Quando estiver pronto, selecione Criar.

A vida simples

Você também pode configurar backups de VM ao criar uma VM no portal do Azure. Na página Settings (Configurações) em que define configurações de rede virtual ou diagnósticos e opções de solução de problemas, você pode habilitar o Backup do Azure. Você pode escolher um cofre existente dos Serviços de Recuperação ou criar um e, em seguida, criar ou usar uma política de backup. Atualmente, você não pode habilitar backups como parte da implantação da VM na CLI do Azure ou no Azure PowerShell, mas geralmente basta um único comando pós-implantação para fazer isso.

Eu gosto de planejar uma estratégia de backup, políticas de retenção e agendas, razão pela qual esses exercícios criaram primeiro o cofre e as políticas dos Serviços de Recuperação. Porém, se você quiser criar uma VM e habilitar backups rapidamente, poderá fazer isso no portal do Azure em uma única etapa.

Agora você tem uma política de backup, que também define políticas de retenção para vários períodos, mas não tem nada para fazer backup ainda. Vamos criar uma VM com o Cloud Shell para que você possa criar um backup e, em um exercício posterior, replicar os dados.

Experimente agora

Para criar uma VM de teste para backup e replicação, conclua as etapas a seguir:

1. Selecione o ícone do Cloud Shell na parte superior do portal do Azure.
2. Crie uma VM com `az vm create`; forneça o nome do grupo de recursos criado no laboratório anterior, como `azuremolchapter13`; e insira o nome de uma VM, como as `molvm`:

```
az vm create \
  --resource-group azuremolchapter13 \
  --name molvm \
  --image win2019datacenter \
  --admin-username azuremol \
  --admin-password P@ssw0rdMoL123
```

Uma política de backup é definida e uma VM de teste está pronta. Para ver o Backup do Azure em ação, vamos aplicar a política de backup à VM.

Experimente agora

Para fazer backup de uma VM com sua política definida, conclua as etapas a seguir:

1. Selecione Grupos de recursos no menu à esquerda no portal.
2. Escolha o grupo de recursos e, em seguida, a VM criada.
3. Em Operações, selecione Backup.
4. Verifique se o cofre dos Serviços de Recuperação está selecionado e escolha a política de backup no menu suspenso.

5 Revise as opções de agenda e retenção e, em seguida, habilite o backup. Leva alguns segundos para que a política de backup seja aplicada.

6 Quando a política estiver habilitada, retorne às configurações de backup. O status da VM relata `Aviso (backup inicial pendente)`.

7 Para criar o primeiro backup, escolha o botão Fazer backup agora, como mostrado na Figura 13.7.

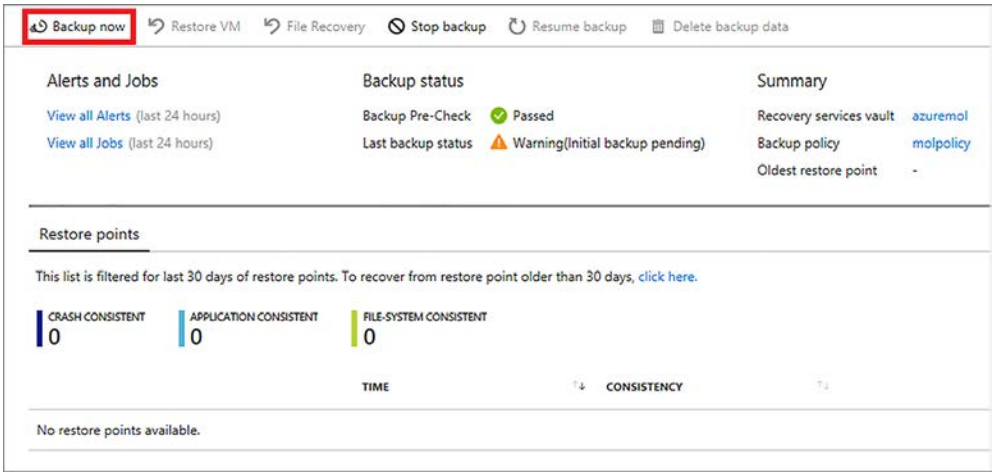


Figura 13.5 Para criar o primeiro backup, selecione o botão Fazer backup agora. O status é atualizado quando concluído e exibe a hora de backup mais recente, o ponto de restauração mais recente e o ponto de restauração mais antigo.

Pode levar de 15 a 20 minutos para concluir a primeira operação de backup. Para ver o andamento do trabalho de backup, você pode selecionar a opção Exibir todos os trabalhos. Não há nenhuma barra de progresso ou indicador de porcentagem, mas você pode verificar se o trabalho ainda está em execução.

Isso é tudo o que é necessário para fazer backup de VMs e proteger seus dados no Azure. Continue lendo para ver como você pode restaurar os dados, se algo der errado.

13.1.3 Restaurar uma VM

O Backup do Azure permite restaurar uma VM completa ou executar uma restauração no nível do arquivo. Em todos os meus anos de atuação no campo, operações de restauração no nível do arquivo eram as mais comuns das duas. Esse tipo de trabalho de restauração geralmente é executado quando os arquivos são excluídos ou substituídos acidentalmente. As restaurações no nível do arquivo geralmente determinam as políticas de retenção para seus backups. Quanto mais importantes forem os dados, mais provável que você queira reter backups por mais tempo, no caso de você receber uma chamada tarde da noite para restaurar um arquivo de seis meses atrás.

Uma restauração completa da VM, como você pode esperar, restaura toda a VM. Raramente eu executei uma restauração completa de VM para colocar uma VM excluída novamente online. Um ótimo caso de uso para uma restauração completa da VM é fornecer uma VM de teste, que é funcionalmente equivalente à original. Você pode restaurar uma VM e, em seguida, testar uma atualização de software ou outro procedimento de manutenção, que pode ajudá-lo a identificar possíveis problemas e criar um plano para lidar com a VM de produção real.

Também é importante testar os backups regularmente. Não espere até que precise restaurar dados em um cenário real. Confie no Backup do Azure, mas verifique se você sabe como e onde restaurar os dados quando necessário.

Restauração no nível do arquivo

Uma restauração no nível do arquivo é um processo muito legal no Backup do Azure. Para oferecer flexibilidade em como e onde você restaura arquivos, o Azure cria um script de recuperação que você faz o download e executa. Esse script de recuperação é protegido por uma senha para que somente você possa executar o processo de recuperação. Quando executar o script de recuperação, insira a senha antes de continuar. A janela para fazer o download do script de recuperação é mostrada na Figura 13.6.

Quando você executa o script de recuperação, seu ponto de recuperação é conectado como um sistema de arquivos local em seu computador. Para VMs do Windows, um script do PowerShell é gerado e um volume local é conectado, como F:. Para VMs Linux, o ponto de recuperação é montado como um disco de dados, como /dev/sdc1 em seu volume inicial. Em ambos os casos, o script de recuperação indica claramente onde você pode encontrar seus arquivos.

Quando concluir a restauração de arquivos do cofre de recuperação, você retornará ao portal do Azure e selecionará a opção Desmontar discos. Esse processo desanexa os discos do computador local e os retorna para uso no cofre de recuperação. Não se preocupe se você esquecer de executar esse processo de desmontagem no calor do momento em que precisa restaurar rapidamente arquivos para uma VM de produção. O Azure desanexa automaticamente quaisquer pontos de recuperação anexados depois de 12 horas.

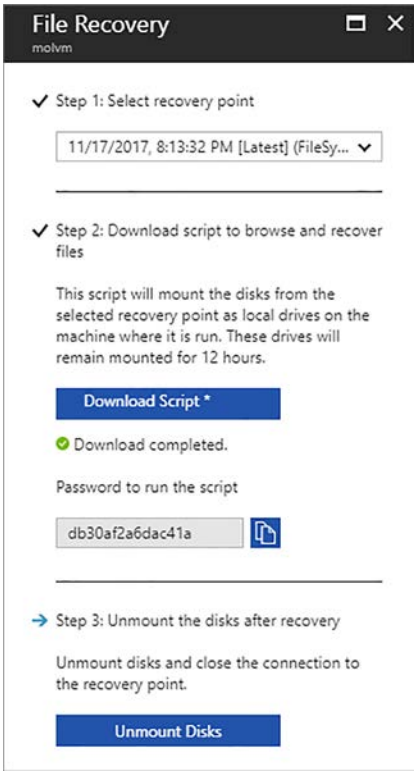


Figura 13.6 Ao executar uma restauração no nível do arquivo, você escolhe um ponto de recuperação para restaurar. Em seguida, um script de recuperação é baixado no seu computador. Você pode executar esse script. Basta inserir a senha gerada. O script de recuperação monta o ponto de recuperação como um volume local no computador. Quando você restaurar os arquivos necessários, desmonte os discos do computador, o que os retorna para uso no cofre de recuperação.

RESTAURAÇÃO COMPLETA DA VM

Uma restauração completa da VM cria uma VM, conecta a VM à rede virtual e anexa todos os discos rígidos virtuais. Vamos experimentar o processo para uma restauração completa da VM. Como é sempre melhor testar atualizações de manutenção antes de executá-las na vida real, este exercício de restauração é uma boa prática.

Experimente agora

Para restaurar uma VM completa, conclua as etapas a seguir:

1. No grupo de recursos, selecione a VM que você fez backup no exercício anterior.
2. Selecione a opção Backup no menu à esquerda na VM. A visão geral de backup deve relatar que um ponto de recuperação foi criado, como mostrado na Figura 13.7. Caso contrário, aguarde alguns minutos e, em seguida, volte a este exercício. Ou basta ler o que o processo implica.

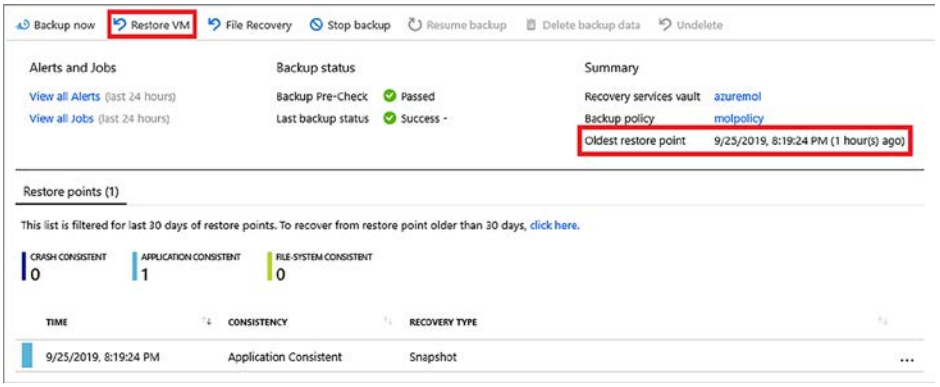


Figura 13.7 Quando o backup da VM for concluído, a página de visão geral mostrará os dados do último backup e dos pontos de restauração disponíveis. Para iniciar o processo de restauração, selecione Restaurar VM.

3. Selecione o botão Restaurar VM, escolha um ponto de restauração da lista e selecione OK.
4. Escolha um ponto de restauração e escolha como restaurar a VM. Você pode escolher criar uma nova VM ou substituir uma VM existente.

   A opção padrão é criar uma nova VM. Nesta configuração, uma nova VM é criada e associada à rede virtual especificada, e os discos são restaurados e conectados.

   Você também pode escolher substituir uma VM existente. Nesse cenário, os discos são restaurados do backup e anexados à VM existente. Qualquer rede virtual ou outras opções de configuração aplicadas à VM são retidas.
5. Para este exercício, escolha restaurar para uma nova VM. Forneça um nome para a VM restaurada, como `restoredvm`, e revise as configurações de rede virtual e armazenamento. Em produção, você normalmente conecta a VM restaurada a uma rede virtual isolada para que não afete o tráfego de produção.
6. Selecione OK e, em seguida, Restaurar.

Leva alguns minutos para conectar o ponto de recuperação e criar uma VM restaurada com os discos anteriores anexados. Neste ponto, você pode se conectar à VM restaurada para testar atualizações de software ou restaurar grandes quantidades de dados, conforme necessário.

Você também pode fazer backup de um aplicativo Web, portanto, essa abordagem não é apenas para VMs. O processo é um pouco diferente, mas os conceitos são os mesmos. Migrar seu modelo de aplicação para uma solução de PaaS como um aplicativo Web não significa que você pode esquecer os conceitos básicos de backups e retenção de dados.

13.2 Azure Site Recovery

Lembre-se de quando discutimos o Cosmos DB, em que você aprendeu que, com o clique de um botão, seus dados são replicados para uma região do Azure completamente diferente para redundância e tolerância a falhas. Você também pode fazer isso com VMs inteiras. O Azure Site Recovery é um serviço poderoso que pode fazer muito mais do que apenas replicar VMs para uma região diferente. A Figura 13.8 descreve como o Azure Site Recovery atua para orquestrar workloads entre locais.

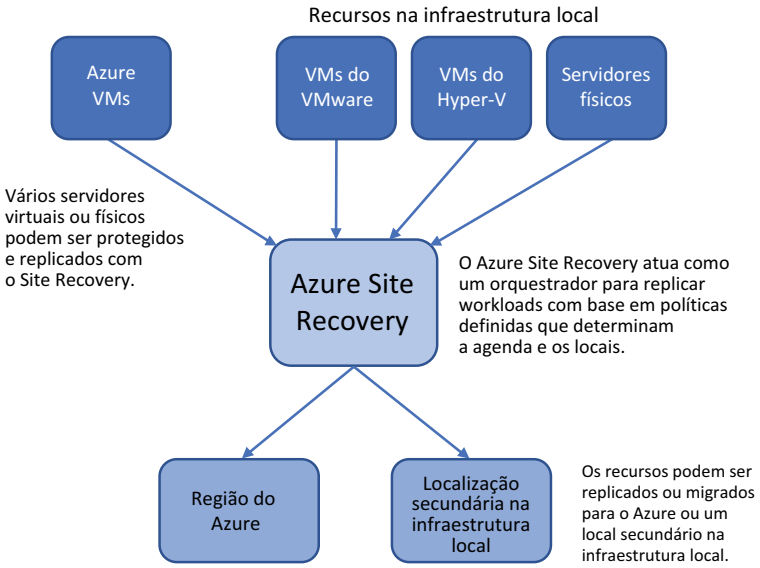


Figura 13.8 O Azure Site Recovery organiza a replicação e a migração de recursos físicos ou virtuais para outro local. Os locais na infraestrutura local e o Azure podem servir como pontos de origem e de destino para proteção, replicação ou migração.

Um aspecto importante é que o Azure Site Recovery serve para mais do que apenas VMs do Azure. O Site Recovery pode ser usado para replicar VMs do VMware ou Hyper-V na infraestrutura local para o Azure para recuperação de desastre (DR) ou como parte de uma migração para o Azure. Você também pode usar o Azure Site Recovery puramente como o orquestrador para replicar VMs e servidores físicos da infraestrutura local para o Azure, ou para replicá-los em uma infraestrutura local secundária.

Da mesma forma que o backup do Azure não significa "só funciona com o Azure", o Azure Site Recovery não significa "apenas replica VMs do Azure". O Azure Backup e o Azure Site Recovery podem ser usados como soluções híbridas para backup e recuperação de desastre. Esses serviços do Azure podem ser usados para proteger todos os workloads, na infraestrutura local e no Azure. Em seguida, uma única estrutura de relatórios para conformidade e validação pode ser gerada para garantir que todos os workloads que você acha que estão protegidos estejam realmente seguros contra perda de dados.

Por que você usaria o Azure Site Recovery? Duas razões principais são mais comuns: replicação e migração.

A replicação protege você de uma paralisação completa da região do Azure. Seria preciso um evento catastrófico para uma região inteira ficar offline, mas quando você trabalha em TI, você sabe que tudo é possível. Mesmos conjuntos de disponibilidade e zonas de disponibilidade, que vimos no capítulo 7, normalmente só protegem você de uma interrupção menor dentro de uma região do Azure. Se a região inteira cair, seu aplicativo ficará indisponível. Com o Site Recovery, todo o seu ambiente de aplicação, incluindo recursos de rede virtual, é replicado para uma região secundária do Azure. Ao clicar em um botão, esse local secundário pode ser colocado online e ativo. Em seguida, o tráfego pode então encaminhar para este local secundário e começar a atender os seus clientes. A Figura 13.9 apresenta uma visão geral de alto nível sobre como o Azure Site Recovery protege seu ambiente.

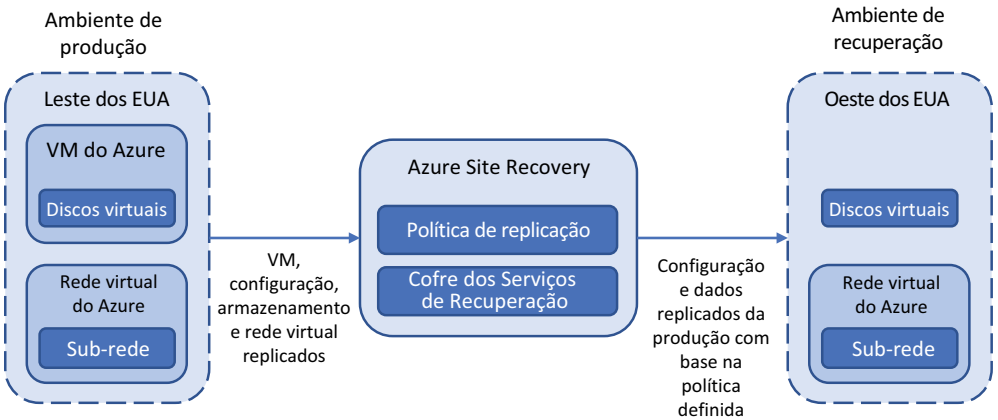


Figura 13.9 O Azure Site Recovery replica a configuração, os dados e as redes virtuais do ambiente de produção para um ambiente de recuperação. As VMs não são criadas no ambiente de recuperação até que um failover seja iniciado. Somente os dados são replicados.

A VM é apenas metadados que definem o tamanho da VM, quais discos estão conectados e quais recursos de rede a VM se conecta. Esses metadados são replicados, o que permite que as VMs sejam criadas rapidamente quando um failover é iniciado. Os discos virtuais são replicados para o ambiente de recuperação e são anexados quando uma VM de recuperação é criada durante um evento de failover.

Para a replicação do Azure para o Azure, não há nenhuma agenda de replicação definida. Os discos são replicados em tempo quase real. Quando os dados nos discos virtuais de origem forem alterados, eles serão replicados para o ambiente de recuperação. Para workloads híbridos, onde você protege VMs VMware ou Hyper-V da infraestrutura local, você define políticas que controlam a agenda de replicação.

Se nos concentrarmos na replicação do Azure para o Azure, como os dados são replicados em tempo quase real? Um cache de conta de armazenamento é criado no local do ambiente de produção, como exibido na figura 13.10. Alterações gravadas nos discos virtuais de produção são imediatamente replicados no cache da conta de armazenamento. Em seguida, o cache da conta de armazenamento é replicado para o ambiente de recuperação. Esse cache de conta de armazenamento atua como um buffer para que qualquer atraso de replicação no local de recuperação distante não afete a performance no workload de produção.

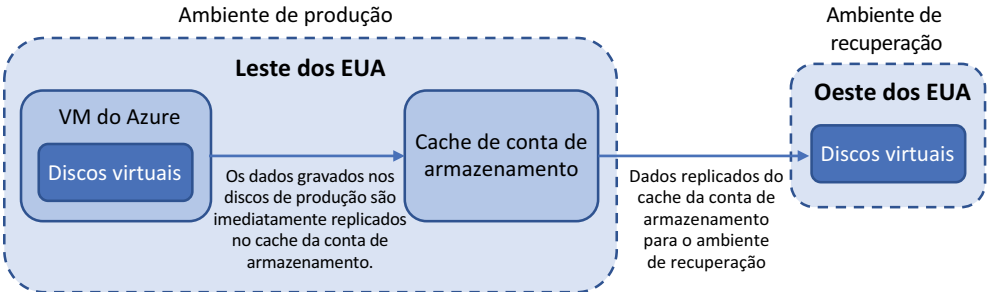


Figura 13.10 Alterações nos discos de produção são imediatamente replicados para um cache da conta de armazenamento. Esse cache de conta de armazenamento evita impactos de performance nos workloads de produção à medida que aguardam para replicar as alterações no local de recuperação remota. Em seguida, as alterações do cache da conta de armazenamento são replicadas para o ponto de recuperação remoto para manter a consistência dos dados.

O processo para configurar o Site Recovery para a replicação do Azure para o Azure é simples, mas leva algum tempo para criar todos os recursos replicados necessários e concluir a replicação de dados inicial. No laboratório de fim de capítulo, você configurará essa replicação do Azure para o Azure.

O que você pode fazer com VMs replicadas para um local secundário com o Azure Site Recovery? Para a maior parte, cruze os dedos e espero que você não precise deles. Porém, existem alguns cenários em que você precisaria deles.

A primeira deve ser óbvia: uma grande paralisação. Se uma região do Azure ficar totalmente indisponível, como por causa de um desastre natural na área, você pode iniciar um failover de seus recursos. Esse failover informa ao Azure Site Recovery para criar VMs no local de recuperação com base nos metadados de VM replicados e, em seguida, anexar os discos rígidos virtuais e as conexões de rede apropriados. Você também pode ser proativo aqui: se está previsto que um desastre natural vá atingir uma região do Azure, você poderá iniciar um failover *antes* de o evento ocorrer. Esta abordagem permite que você decida quando incorrer em algum tempo de inatividade potencial à medida que os recursos fazem failover para o local secundário, normalmente fora do horário comercial primário. Quando o evento previsto passar na região principal do Azure, você poderá voltar seus recursos e deixar que continuem a ser executados normalmente.

O segundo cenário em que você pode fazer failover é testar se o processo funciona. Da mesma forma que os backups devem ser testados regularmente, você deve testar um plano de replicação e failover. Seria muito embaraçoso e estressante descobrir que quando você precisa trazer um local secundário on-line, há alguma configuração incorreta nas redes virtuais, ou uma das aplicações não volta a funcionar. O Azure deve fornecer uma opção especificamente para teste de failover. Uma rede virtual do Azure isolada é normalmente usada no local secundário e os workloads de produção continuam a ser executadas normalmente no local principal. Se você usar o Azure Site Recovery, certifique-se de testar o processo de failover regularmente.

13.3 Laboratório: Configurar uma VM para o Site Recovery

Há vários pré-requisitos para se configurar a replicação do VMware ou do Hyper-V na infraestrutura local com o Azure Site Recovery. É uma excelente funcionalidade, tanto para fins de recuperação de desastres e para migrar VMs para o Azure, mas leva muito mais tempo do que seu horário de almoço. Então, se você quiser saber mais sobre esses cenários, acesse http://mng.bz/x71V.

Vamos configurar a replicação do Azure para o Azure com a VM de teste que você criou e fez backup anteriormente:

1. No portal do Azure, escolha Grupos de recursos no menu à esquerda.
2. Selecione o grupo de recursos usado nos exercícios anteriores, como azuremolchapter13.
3. Selecione a VM que você criou nos exercícios anteriores, como o molvm.
4. Escolha Recuperação de desastres no menu à esquerda na janela VM.
5. Nas configurações avançadas, observe as configurações padrão usadas pelo Azure site Recovery para criar um grupo de recursos e uma rede virtual no local de destino. Um cache de conta de armazenamento é criado para replicar dos discos virtuais de origem e um cofre e uma política dos Serviços de Recuperação são criados para controlar o processo de replicação.
6. Você não precisa alterar aqui, embora se você usar o Site Recovery em produção e tiver várias VMs para proteger, você precisará revisar como as VMs mapeiam para redes virtuais replicadas e sub-redes existentes. Para este laboratório, revise e habilite a replicação usando os valores padrão.

Agora, volte ao trabalho. Sério! Leva um tempo para configurar todos os recursos replicados e concluir a sincronização de dados inicial. Não espere, a menos que seu chefe concorde que você faça um intervalo para almoço mais longo hoje.

Manter os backups protegidos contra exclusão

Espero que, como prática recomendada, você exclua grupos de recursos e seus recursos no final de cada capítulo para manter seus créditos gratuitos do Azure disponíveis para uso no restante do livro.

Se você tiver VMs protegidas com Azure Backup or Site Recovery, você não poderá excluir o cofre dos Serviços de Recuperação ou o grupo de recursos da VM. A plataforma do Azure sabe que você tem dados ativos que são copiados ou replicados e impede que esses recursos sejam excluídos.

Para excluir VMs protegidas, desative primeiro qualquer trabalho de backup ativo ou VMS replicadas. Quando fizer isso, você pode optar por reter os dados protegidos ou removê-los. Para os exercícios de laboratório neste capítulo, escolha excluir os pontos de restauração. Como recurso de segurança, o Azure automaticamente exclui de forma reversível esses pontos de restauração e permite que você cancele a exclusão deles por 14 dias. Não há nada para configurar aqui, e você não pode remover de forma forçada esses pontos de restauração excluídos de forma reversível. Eu não recomendo, mas você também pode desabilitar a função de exclusão reversível de um cofre dos Serviços de Recuperação selecionando Propriedades do cofre no portal do Azure.

A boa notícia é que o restante do grupo de recursos pode ser excluído, e você não paga por estes pontos de restauração excluídos de forma reversível. Quando o período de exclusão reversível de 14 dias terminar, o cofre dos Serviços de Recuperação pode ser excluído normalmente. O objetivo aqui é proteger você contra exclusão acidental ou maliciosa de pontos de restauração e oferecer tempo para perceber que eles são realmente necessários e recuperá-los.

14 Criptografia de dados

A segurança dos seus dados é importante. Mais especificamente, a segurança dos dados dos seus clientes é fundamental. Dificilmente ficamos uma semana sem ler a notícia de uma grande empresa que encontrou uma violação de dados. Muitas vezes, esses incidentes são causados por falta de segurança, configuração incorreta ou simples descuido. Nesta era digital, é muito fácil para os invasores automatizarem as tentativas de obter acesso aos seus dados. O tempo para se recuperar de um incidente de segurança no nível da aplicação pode não ser nada comparado a quanto tempo a empresa leva para recuperar a confiança dos clientes caso os dados *deles* sejam expostos.

O Azure inclui recursos de criptografia que tornam difícil afirmar que você não tem tempo ou experiência para proteger seus dados. Neste capítulo, examinamos como criptografar dados armazenados no Armazenamento do Azure, em discos gerenciados ou na VM inteira. Livros inteiros foram escritos sobre criptografia de dados, e este capítulo não se profunda em considerações e métodos de criptografia. Em vez disso, você verá como habilitar alguns dos principais recursos e serviços do Azure para proteger seus dados em todo o ciclo de vida da aplicação.

14.1 O que é criptografia de dados?

Ao comprar algo online, você verifica se há um pequeno ícone de cadeado na barra de endereços para indicar que o site usa HTTPS? Por que é ruim enviar seus dados de crédito em uma conexão HTTP normal e não segura? Cada bit de dados em um pacote de rede que flui entre dispositivos poderia ser monitorado e examinado. A Figura 14.1 mostra como fazer compras online sem uma conexão HTTPS pode ser ruim para o extrato do seu cartão de crédito.

Não há desculpa para os servidores Web usarem conexões não seguras. Cada aplicativo Web que você cria no Azure tem automaticamente um certificado SSL curinga aplicado a ele.

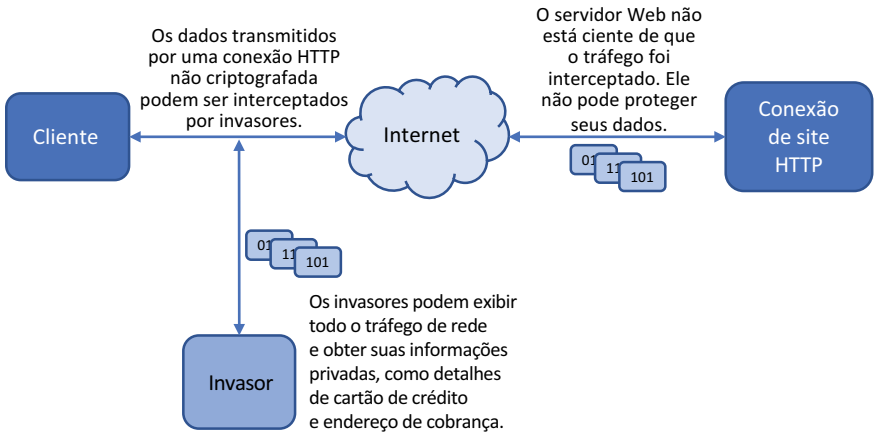


Figura 14.1 Neste exemplo básico, um invasor pode interceptar o tráfego de rede enviado por uma conexão HTTP não criptografada. Como seus dados não são criptografados, o invasor pode juntar os pacotes de rede e obter suas informações pessoais e financeiras. Se você se conectar ao servidor Web em uma conexão HTTPS criptografada, um invasor não poderá ler o conteúdo dos pacotes de rede e exibir os dados.

Um *certificado SSL* é um componente digital usado para proteger o servidor Web e permitir que um navegador da Web valide a conexão. Um certificado SSL curinga pode ser usado em um domínio inteiro, como *.azurewebsites.net, o domínio padrão para aplicativos Web. Ao criar um aplicativo Web no capítulo 3, você poderia ter adicionado https:// tao endereço da Web e começado a usar comunicações criptografadas com seus Aplicativos Web. E é só isso.

Certificados SSL personalizados são relativamente baratos e fáceis de implementar. Em projetos como Let's Encrypt (https://letsencrypt.org), você pode obter um certificado gratuitamente e configurar automaticamente seu servidor Web em minutos. Você também pode comprar e usar um certificado do Serviço de Aplicativo que é integrado diretamente a aplicativos Web. Os certificados do Serviço de Aplicativo são armazenados no Azure Key Vault, que examinaremos mais no capítulo 15.

Ao projetar e criar aplicações no Azure, você deve implementar comunicações seguras sempre que possível. Essa abordagem ajuda a proteger os dados enquanto em trânsito, mas e quando esses dados são gravados em disco? Existe um processo semelhante para discos e VMs que assegura e protege seus dados em repouso. A Figura 14.2 mostra como funciona a criptografia de disco e VM.

Espero que esses exemplos simplificados de criptografia de dados no Azure motivem você a implementar a criptografia ao projetar e criar aplicações no Azure. A maioria dos clientes espera que seus dados sejam protegidos, e muitas empresas têm mandatos regulatórios e de conformidade que requerem criptografia. Não considere apenas as multas em potencial para a empresa por uma violação de dados ou a perda de confiança do cliente. Considere o risco ao qual os dados pessoais e financeiros dos clientes serão expostos e como essa exposição pode

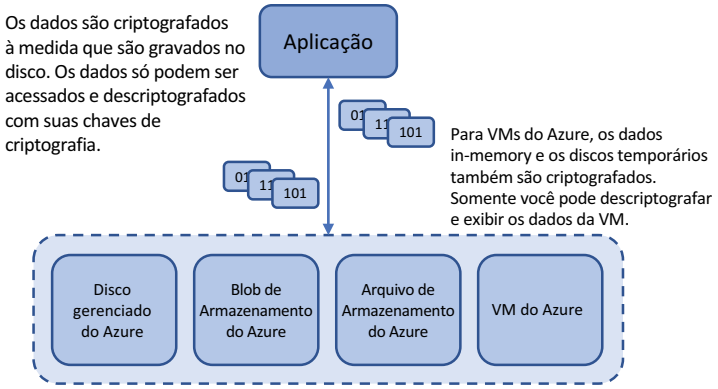


Figura 14.2 Ao criptografar seus dados, somente você pode descriptografar e exibir o conteúdo. Se um invasor obtivesse acesso a um disco virtual ou a arquivos individuais, ele não seria capaz de descriptografar o conteúdo. Os métodos de criptografia podem ser combinados: os clientes podem se conectar à sua Web por HTTPS, você pode forçar o tráfego para armazenar as contas por HTTPS e criptografar os dados gravados no disco.

afetar sua vida diária. Você provavelmente não gosta da ideia de seus próprios dados serem expostos. Por isso, faça tudo o que puder para proteger os dados de seus clientes.

14.2 Criptografia em repouso

Se a criptografia de dados é tão importante, como usá-la no Azure? Basta continuar fazendo o que você já aprendeu neste livro! Logo no início, eu mencionei que todas as suas VMs devem usar discos gerenciados, certo? Há muitas boas razões para isso, e uma delas é a segurança. Um disco gerenciado é criptografado automaticamente. Você não precisa configurar nada, e não há nenhum impacto de performance quando ele está habilitado. Isso não é opcional. Seus dados são automaticamente criptografados em repouso com discos gerenciados.

O que significa criptografar *dados em repouso*? Quando você usa discos gerenciados, seus dados são criptografados quando gravados no armazenamento subjacente do Azure. Os dados que residem nos discos temporários ou dados que existem na memória na VM não são criptografados. Os dados são criptografados somente quando os dados do SO ou disco de dados *estão* no disco físico subjacente. A Figura 14.3 mostra como os dados são criptografados à medida que são gravados em um disco gerenciado.

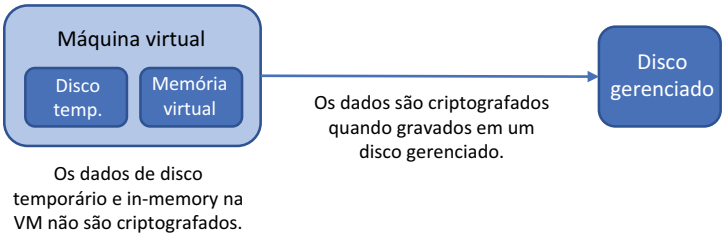


Figura 14.3 À medida que os dados são gravados em um disco gerenciado, eles são criptografados. Dados in-memory na VM ou dados em discos temporários locais para a VM não são criptografados, a menos que a VM inteira esteja habilitada para criptografia, algo que veremos na seção 14.4.2. A criptografia automática de dados gravados em discos gerenciados não causa sobrecarga para a VM. A plataforma do Azure executa a operação de criptografia no armazenamento subjacente. A VM não precisa manipular nenhum processo de criptografia/descriptografia.

Essa criptografia em repouso para discos gerenciados significa que não há nenhum impacto de performance nas VMs. Não há nenhum processamento adicional para a VM executar para criptografar e descriptografar os dados e, portanto, toda a capacidade de CPU disponível pode ser usado para executar aplicações. Em cenários típicos de criptografia de VM, a VM usa uma certa quantidade de poder de computação para processar e gerenciar a criptografia de dados. A compensação para a criptografia automática de disco gerenciado é que apenas os discos de SO e dados são protegidos. Outros dados de disco temporário ou na memória na VM podem ser expostos.

A Microsoft gerencia as chaves de criptografia digital na plataforma do Azure com a criptografia automática de discos gerenciados. Isso cria outra compensação, pois você pode criptografar automaticamente seus dados sem a necessidade de criar, gerenciar, alternar ou revogar chaves, mas tem que confiar na Microsoft para proteger essas chaves.

14.3 Criptografia do Serviço de Armazenamento

A criptografia automática de disco gerenciado é ótima, mas e se você usar o Armazenamento do Azure para armazenamento de blobs ou arquivos? A Criptografia do Serviço de Armazenamento (SSE) do Azure permite criptografar dados no nível da conta de armazenamento. Os dados são criptografados à medida que são gravados na conta. Mais uma vez, a Microsoft lida com as chaves de criptografia e, portanto, nenhuma sobrecarga ou configuração de gerenciamento é necessária. A plataforma do Azure abstrai a geração e o gerenciamento de chaves para você. Se preferir, você pode criar e usar suas próprias chaves de criptografia, com um pouco de sobrecarga de gerenciamento adicional. Assim como a criptografia de disco gerenciada automática em repouso, a criptografia de armazenamento do Azure é automaticamente habilitada quando você cria uma conta.

O objetivo da criptografia automática de disco gerenciado e SSE é tornar o mais fácil possível para você criptografar seus dados e gastar mais tempo em projetar, criar e executar suas aplicações. A Figura 14.4 mostra como a SSE protege seus dados e também pode forçar comunicações seguras quando os dados estão em trânsito.

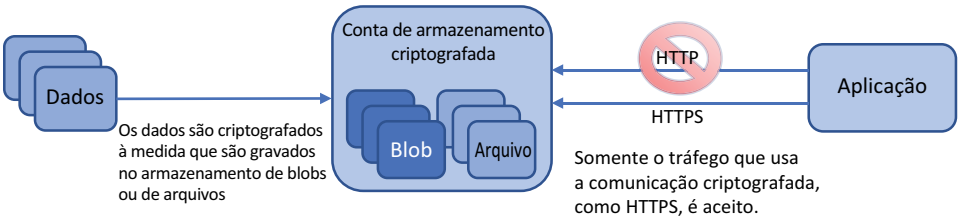


Figura 14.4 Quando você habilita a SSE, os blobs e arquivos do Azure são criptografados à medida que os dados são gravados no disco. As tabelas e filas do Azure não são criptografadas. Para obter segurança adicional de dados, você pode forçar todas as comunicações com uma conta de Armazenamento a usar protocolos de comunicação segura, como HTTPS. Isso protege os dados em trânsito até o momento em que eles são criptografados no disco.

Forçar o tráfego de armazenamento a usar transferências seguras

Junto com a habilitação da SSE, você pode forçar todas as solicitações de armazenamento e transferências a usar um método de comunicação segura. Essa configuração força todas as chamadas de API REST a usar HTTPS, e todas as conexões de arquivo do Azure que não habilitam a criptografia, como versões mais antigas do protocolo SMtB, a serem descartadas.

(continuação)

Os SDKs do Azure, como os exemplos do Python que examinamos no capítulo 4, podem usar conexões criptografadas. Os documentos de referência para cada SDK específico da linguagem fornecem orientações sobre como implementar comunicações seguras.

O uso de comunicações seguras deve ser incorporado em aplicações desde o início. Isso pode causar problemas para habilitar comunicações seguras em uma aplicação existente se alguns componentes não foram originalmente configurados apropriadamente. No mínimo, primeiro teste comunicações seguras para uma aplicação existente em um ambiente de desenvolvimento.

Experimente agora

Para criar uma conta de armazenamento e habilitar a criptografia e as comunicações seguras, conclua as etapas a seguir:

1. Abra o portal do Azure e escolha o ícone do Cloud Shell no menu superior.
2. Crie um grupo de recursos, forneça um nome de grupo de recursos, como `azuremolchapter14` e forneça um local, como `eastus`:

   ```
   az group create --name azuremolchapter14 --location eastus
   ```

3. Crie uma conta de armazenamento com `az storage account create`. Forneça um nome exclusivo, como `azuremolstorage` e insira o grupo de recursos criado na etapa 2. Insira um tipo de conta de armazenamento, como `Standard_LRS` para armazenamento com redundância local. Para forçar comunicações seguras, defina `--https-only`.

   ```
   az storage account create \
     --name azuremolstorage \
     --resource-group azuremolchapter14 \
     --sku standard_lrs \
     --https-only true
   ```

4. Verifique se a conta de armazenamento é criptografada e habilitada para comunicações seguras consultando `enableHttpsTrafficOnly` e os parâmetros de criptografia:

   ```
   az storage account show \
     --name azuremolstorage \
     --resource-group azuremolchapter14 \
     --query [enableHttpsTrafficOnly,encryption]
   ```

   A saída é semelhante à seguinte:

   ```
   [
     true,
     {
       "keySource": "Microsoft.Storage",
   ```

```
    "keyVaultProperties": null,
    "services": {
      "blob": {
        "enabled": true,
        "lastEnabledTime": "2019-09-27T03:33:17.441971+00:00"
      },
      "file": {
        "enabled": true,
        "lastEnabledTime": "2019-09-27T03:33:17.441971+00:00"
      },
      "queue": null,
      "table": null
    }
  }
]
```

14.4 Criptografia de VM

A criptografia automática de discos gerenciados do Azure ajuda a fornecer um nível de segurança da VM. Para obter uma abordagem abrangente da segurança de dados da VM, você pode criptografar a própria VM. Esse processo envolve mais do que criptografar os discos rígidos virtuais subjacentes. O disco do SO e todos os discos de dados anexados, juntamente com o disco temporário, são criptografados. A memória da VM também é criptografada para reduzir ainda mais a superfície de ataque. Você usa chaves digitais para criptografar VMs.

Uma vantagem de criptografar a VM inteira é que você gerencia as chaves de criptografia. Essas chaves de criptografia são armazenadas com segurança no Azure Key Vault e você pode escolher entre usar chaves geradas por software ou hardware. Você controla essas chaves e, assim, pode definir o acesso a elas e usar controles de acesso baseado em função e auditoria para rastrear o uso. Você também pode alternar as chaves de criptografia em uma agenda definida, bem como alterar sua senha a cada 60 ou 90 dias. Esses controles adicionais e tarefas de gerenciamento para chaves de criptografia adicionam alguma sobrecarga de gerenciamento, mas fornecem a máxima flexibilidade para proteger seus dados e podem ser necessários para determinados fins regulatórios. Vamos analisar um pouco mais o Azure Key Vault.

14.4.1 Armazenar chaves de criptografia no Azure Key Vault

O capítulo 15 abordará o Azure Key Vault, mas eu quero primeiro mostrar a você o poder de criptografia de dados e criptografia de VM. Como uma visão geral rápida, o Azure Key Vault é um cofre digital que permite armazenar com segurança chaves de criptografia, certificados SSL e segredos, como senhas. Para redundância, os cofres de chave são replicados em regiões do Azure. Essa replicação protege suas chaves e segredos e assegura que estejam sempre disponíveis para uso.

Só você tem acesso aos seus cofres de chave. Você gera e armazena objetos em cofres de chave e, em seguida, define quem tem acesso aos cofres. A Microsoft gerencia o serviço subjacente do Key Vault, mas não tem acesso ao conteúdo dos cofres. Esse limite de segurança significa que, quando criptografa seus dados no Azure, você é o único que pode descriptografá-los e exibi-los.

Experimente agora

Para criar um cofre de chaves e chave de criptografia, conclua as etapas a seguir:

1. Abra o portal do Azure e escolha o ícone do Cloud Shell no menu superior.
2. Crie um cofre de chaves com o comando `az keyvault create`. Especifique o grupo de recursos que você criou no exercício anterior, como azuremolchapter14, e forneça um nome exclusivo para seu cofre de chaves, como `azuremolkeyvault`:

```
az keyvault create \
  --resource-group azuremolchapter14 \
  --name azuremolkeyvault \
  --enabled-for-disk-encryption
```

Vamos pausar e pensar sobre por que você adiciona um parâmetro em `--enabled-for-disk-encryption`. Quando você criptografa uma VM, a plataforma do Azure precisa ser capaz de iniciar e descriptografar a VM para que ela possa ser executada. A plataforma do Azure não tem permissões para acessar esses dados e a Microsoft não tem acesso para exibir e usar essas chaves de criptografia para nada além de iniciar uma VM. Ao habilitar um cofre de chaves para criptografia de disco, você concede permissões para que o Azure acesse o cofre de chaves e use a chave de criptografia associada a uma VM.

Novamente, a Microsoft não tem acesso a essas chaves ou seus dados, apenas a capacidade de iniciar sua VM criptografada. É difícil fazer muita coisa com uma VM criptografada quando esta não pode ser inicializada. A Figura 14.5 mostra como a plataforma do Azure usa a chave de criptografia para iniciar uma VM criptografada.

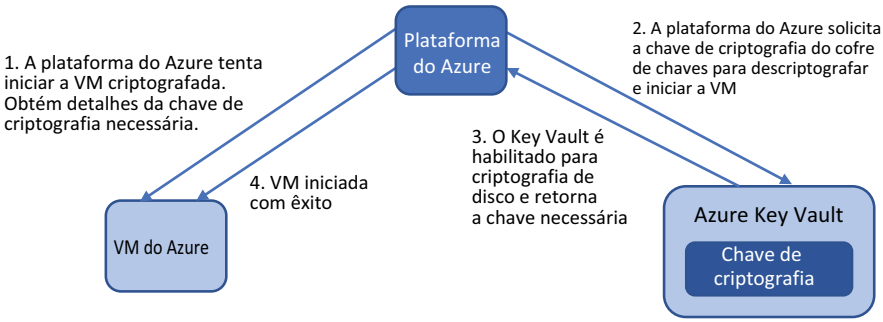


Figura 14.5 Quando um cofre de chaves é habilitado para criptografia de disco, ele concede permissão para a plataforma do Azure solicitar e usar a chave de criptografia para iniciar com êxito uma VM criptografada.

As chaves podem ser criadas e armazenadas em software ou podem ser armazenadas em módulos de segurança de hardware (HSMs) para segurança adicional. Para muitas finalidades, as chaves de software funcionam muito bem, embora você possa ter mandatos de segurança que requerem o uso de HSMs. Discutiremos mais sobre este tópico no capítulo 15.

3 Para criar uma chave, especifique o cofre criado na etapa 2, como azuremolkeyvault, e forneça um nome de chave, como `azuremolencryptionkey`:

```
az keyvault key create \
  --vault-name azuremolkeyvault \
  --name azuremolencryptionkey \
  --protection software
```

14.4.2 Criptografar uma VM do Azure

A chave de criptografia que você criou na seção 14.4.1 pode ser usada para criptografar muitas VMs, se quiser. Essa abordagem minimiza a sobrecarga de gerenciamento de chaves e, se você usar conjuntos de escala de máquinas virtuais, permite que escalar automaticamente o número de instâncias de VM sem a necessidade de gerar chaves de criptografia a cada vez. A alternativa é que cada VM tem a sua própria chave de criptografia, o que adiciona complexidade, mas fornece uma camada de segurança para suas VMs. Por exemplo, se você tiver a mesma chave de criptografia usada para VMs de aplicação de back-end e, em seguida, VMs de banco de dados, um suposto invasor com essa chave poderá obter acesso aos dados de ambos os conjuntos de VMs. Se forem usadas chaves diferentes, o número de VMs possivelmente comprometidas será menor. No laboratório de fim de capítulo, você criptografará uma única VM, embora o mesmo processo possa funcionar com um conjunto de escala que tem várias VMs, mas usa apenas uma única chave. Principalmente quando você trabalha com aplicações maiores de dimensionamento automático, certifique-se de projetar e criar recursos de segurança.

Quando você criptografa uma VM, uma extensão de VM do Azure é instalada. A extensão controla a criptografia do disco do sistema operacional, disco temporário, todos os discos de dados anexados e dados in-memory, como mostrado na Figura 14.6. Para VMs do Windows, o mecanismo de criptografia BitLocker é usado. Para VMs Linux, o dm-crypt é usado para processar a criptografia. A extensão de VM pode relatar o status de criptografia e descriptografar a VM conforme desejado.

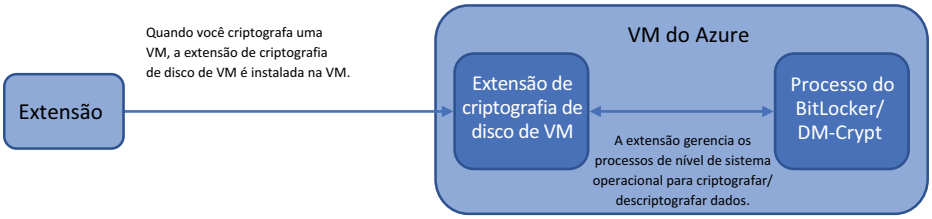


Figura 14.6 Quando você criptografa uma VM, a extensão de criptografia de disco do Azure é instalada. Essa extensão gerencia o uso do BitLocker em VMs do Windows ou dm-crypt em VMs do Linux, para executar a criptografia de dados em sua VM. A extensão também é usada quando você consulta o status de criptografia de uma VM.

Como a extensão de criptografia de disco de VM depende do BitLocker ou do dm-crypt, há algumas limitações sobre o uso da criptografia de VM. A maioria das imagens do Azure Marketplace oferece suporte a criptografia de disco, embora existam algumas restrições para tamanhos de VM que oferecem suporte a criptografia ou

criptografia de compartilhamentos de arquivos de rede conectados, como arquivos do Azure. Para obter as informações mais abrangentes sobre limitações e considerações com suporte para criptografia de VM, leia os documentos mais recentes do Azure em http://mng.bz/yyvd.

Este capítulo forneceu uma introdução rápida aos recursos de criptografia e segurança de dados no Azure. A criptografia automática de discos gerenciados e o SSE não exigem muita configuração, portanto, não há nenhum obstáculo real para impedi-lo de usá-los.

14.5 Laboratório: Criptografar uma VM

Vamos ver tudo isso em ação criptografando uma VM com a chave de criptografia que você armazenou no cofre de chaves:

1. Crie uma VM A maioria das imagens do Linux no Azure Marketplace oferece suporte à criptografia, assim como as imagens do Windows Server do Server 2008 R2 e posterior. Para agilizar e facilitar, crie uma VM do Ubuntu LTS, assim como você fez na maior parte deste livro. Como a VM exige memória suficiente para executar a operação de criptografia de disco, especifique um tamanho de Standard_D2s_v3:

```
az vm create \
  --resource-group azuremolchapter14 \
  --name molvm \
  --image ubuntults \
  --size Standard_D2s_v3 \
  --admin-username azuremol \
  --generate-ssh-keys
```

2. Habilite a criptografia na VM e forneça o nome do Azure Key Vault e uma chave digital usada em um exercício anterior:

```
az vm encryption enable \
  --resource-group azuremolchapter14 \
  --name molvm \
  --disk-encryption-keyvault azuremolkeyvault \
  --key-encryption-key azuremolencryptionkey
```

   Leva alguns minutos para instalar a extensão de criptografia de disco da VM do Azure e começar o processo de criptografar a VM.

3. Quando que a criptografia for iniciada, monitore o andamento e esteja pronto para reiniciar a VM para concluir o processo de criptografia. Visualize o status da seguinte maneira:

```
az vm encryption show \
--resource-group azuremolchapter14 \
--name molvm \
--query 'status'
```

   Veja um exemplo de saída de uma VM no processo de ser criptografada. No início, a mensagem de status relata como

```
[
  {
    "code": "ProvisioningState/succeeded",
    "displayStatus": "Provisioning succeeded",
    "level": "Info",
    "message": "OS disk encryption started",
    "time": null
  }
]
```

Pode demorar um pouco para concluir a criptografia de disco, por isso, este pode ser outro bom exercício de laboratório para voltar em cerca de uma hora, a menos que você queira um intervalo maior para o almoço. Ei, eu *ainda* não sou seu chefe, mas fica chato olhar para a mesma mensagem de status de criptografia.

4 Quando o status de criptografia for relatado como `Criptografia bem-sucedida para todo o volumes`, reinicie a VM:

```
az vm restart --resource-group azuremolchapter14 --name molvm
```

Em seguida, você pode verificar novamente o status de criptografia da VM com `az vm encryption show` para confirmar que a VM seja relatada como criptografada.

Lembre-se de suas tarefas de administração

Estes últimos dois laboratórios de fim de capítulo não demoraram muito para serem concluídos, mas eles podem ter levado um tempo para terminar. Não se esqueça de voltar e excluir os recursos quando você terminar.

Como abordado no capítulo 13, lembre-se de que você precisa desabilitar a proteção do Azure Backup ou do Site Recovery antes de excluir o cofre dos Serviços de Recuperação e o grupo de recursos (após aguardar 14 dias para que os pontos de recuperação gratuitos de exclusão reversível expirem). Volte e limpe os recursos de laboratório antes que eles comecem a usar muitos dos seus créditos gratuitos do Azure.

15 Proteger informações com o Azure Key Vault

Quase todas as semanas, surgem notícias de um incidente de segurança cibernética envolvendo uma grande empresa. Da mesma forma que você usou várias formas de automação para aumentar ou replicar suas aplicações e dados, os invasores automatizam suas próprias ações. É improvável que uma única pessoa vá tentar comprometer manualmente a segurança de seus sistemas. Este conceito torna difícil defender seus sistemas 24 horas por dia, 7 dias por semana, 365 dias por ano (está bem, ou 366 dias).

O capítulo 14 discutiu como criptografar seus dados e VMs. Esse processo é um ótimo primeiro passo e examinamos brevemente como criar e usar chaves de criptografia armazenadas com o serviço Azure Key Vault. Dados confidenciais, como chaves, segredos e certificados, são melhor armazenados em um cofre digital, como um cofre de chaves, que pode gerenciar, emitir e auditar centralizadamente o uso de suas credenciais e dados críticos. Como suas aplicações e serviços precisam de acesso a diferentes recursos, eles podem automaticamente solicitar, recuperar e usar essas chaves, segredos e credenciais. Neste capítulo, você aprenderá por que e como criar um cofre de chave seguro, controlar o acesso e, em seguida, armazenar e recuperar segredos e certificados.

15.1 Proteger informações na nuvem

À medida em que as aplicações se tornam mais complexos e o risco de ciberataques aumenta, a segurança torna-se uma parte crítica de como você cria e executa seus serviços. Principalmente à medida que você executa mais aplicações voltadas para a Internet, na infraestrutura local ou na nuvem, minimizar o risco de acesso a dados não autorizados deve ser uma das principais áreas de design em que você se concentra. Não faz sentido ter a maior pizzaria do mundo se os clientes não puderem confiar a você seus detalhes de pagamento ou informações pessoais.

Uma maneira comum de fornecer segurança para aplicações e serviços é por meio do uso de chaves digitais, segredos e certificados, como mostrado na Figura

15.1. Em vez de usar um nome de usuário e uma senha que devem ser inseridos toda vez manualmente (ou, talvez pior, gravados em um arquivo de configuração não criptografado), use um cofre digital para armazenar essas credenciais e dados seguros. Quando uma aplicação ou um serviço exige acesso, ele solicita a chave ou o segredo específico necessário, e uma trilha de auditoria também é criada para rastrear qualquer possível uso indevido ou violação de segurança.

Figura 15.1 O Azure Key Vault fornece uma maneira segura de armazenar informações digitais, como certificados, chaves e segredos. Em seguida, esses itens seguros podem ser acessados diretamente por suas aplicações e serviços, ou recursos do Azure, como VMs. Com a interação humana mínima, você pode distribuir centralizadamente credenciais e certificados seguros em seus ambientes de aplicações.

Quando projetados e implementados corretamente, esses cofres digitais são quase totalmente automatizados e seguros. Os serviços podem solicitar um novo certificado digital, ter um emitido que seja armazenado de forma segura no cofre, e usá-lo para outros componentes da aplicação. Os servidores podem configurar o software recuperando segredos como senhas do cofre digital e instalando componentes de aplicações, sem que as credenciais sejam armazenadas em um arquivo de configuração baseado em texto. Um administrador de aplicação pode gerenciar centralizadamente todos os segredos, chaves e certificados para um serviço e atualizá-los regularmente, conforme necessário.

O Azure Key Vault fornece todos esses recursos de segurança digital e permite que você controle de forma rígida quais usuários e recursos podem acessar os dados seguros. Os cofres de chaves podem ser replicados com segurança para redundância e performance aprimorada da aplicação e são integrados a recursos comuns do Azure, como VMs, aplicativos Web e contas do Armazenamento do Azure.

15.1.1 Cofres de software e módulos de segurança de hardware

Antes de ir para um exemplo prático de como criar e usar um cofre de chaves, é importante entender a maneira como suas informações seguras são armazenadas em um cofre. Como mostrado na Figura 15.2, todas as chaves, segredos e certificados em um cofre de chaves são armazenados em um módulo de segurança de hardware (HSM). Estes dispositivos não são exclusivos no Azure. Eles são dispositivos de hardware de todo o setor, que fornecem um alto nível de segurança para todos os dados armazenados neles.

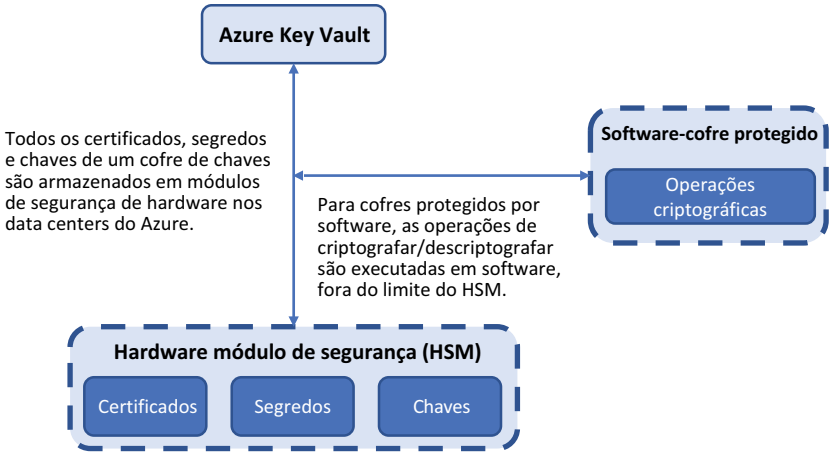


Figura 15.2 O Azure Key Vault é um recurso lógico no Azure, mas todos os certificados, segredos e chaves são armazenados em um HSM. Para cenários de desenvolvimento ou teste, um cofre protegido por software pode ser usado, que então executa qualquer operação criptográfica (como criptografar ou descriptografar dados) em software, não em hardware no HSM. Para produção, você deve usar um cofre protegido por HSM, em que todo o processamento é feito em hardware.

No momento, você pode usar dois tipos de cofre de chaves: protegido por software e protegido por HSM. A diferença entre eles pode ser confusa, e é por isso que quero esclarecer isso antes de começarmos:

- Um *cofre protegido por software* armazena chaves, segredos e certificados em um HSM, mas todas as operações criptográficas necessárias para criptografar ou descriptografar seu conteúdo são executadas pela plataforma do Azure no software. Cofres protegidos por software são ótimos para cenários de desenvolvimento e teste, embora você possa decidir que os workloads de produção exijam uma maneira um pouco mais segura para executar as operações criptográficas.
- Um *cofre protegido por HSM* armazena chaves, segredos e certificados em um HSM, e as operações criptográficas necessárias para criptografar ou descriptografar seu conteúdo são executadas diretamente no HSM. Você também pode gerar suas próprias chaves seguras em um HSM na infraestrutura local e, em seguida, importá-las para o Azure. Existem algumas ferramentas e processos adicionais a serem seguidos, mas desta forma você garante que controla totalmente as chaves e que elas nunca saiam dos limites do HSM.

  Para maximizar a segurança e a integridade dos seus dados, os cofres protegidos por hardware são a abordagem preferencial para workloads de produção.

Independentemente de qual tipo de cofre você usa, é importante lembrar que todos os seus dados são armazenados com segurança em um HSM validado (no mínimo) com o padrão federal de processamento de informações (FIPS) 140-2 nível 2 e que a Microsoft não pode acessar ou recuperar as chaves. Há um custo adicional para cofres protegidos por HSM, de modo que, como com qualquer coisa no Azure e na computação na nuvem, equilibre o custo versus o risco de que seus dados sejam comprometidos.

15.1.2 Criar um cofre de chaves e segredo

Um cofre digital parece ótimo, mas você pode estar um pouco inseguro sobre como aproveitar o poder que o Azure Key Vault oferece. Vamos criar um exemplo de um servidor básico que executa um banco de dados como o MySQL Server, como mostrado na Figura 15.3.

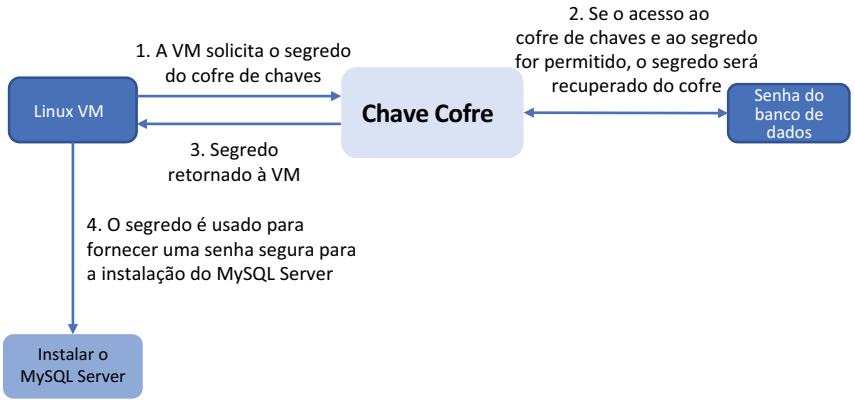


Figura 15.3 Nos próximos exercícios, você criará um exemplo de um segredo armazenado em um cofre de chaves que pode ser usado como a senha do banco de dados para uma instalação do MySQL Server. Uma VM é criada com permissões para solicitar o segredo do cofre de chaves. Em seguida, o segredo recuperado é usado para inserir automaticamente uma credencial segura durante o processo de instalação da aplicação.

Um dos primeiros exercícios deste livro foi criar uma VM e, em seguida, instalar a pilha do servidor Web LAMP. Provavelmente, você precisou fornecer uma senha do servidor MySQL, ou uma senha em branco foi usada automaticamente. Agora que você sabe tudo sobre cofres-chave, você pode recuperar uma senha do cofre de forma automática e usá-la de forma dinâmica para instalar e configurar o servidor.

Experimente agora

Para criar um cofre de chaves e adicionar um segredo, conclua as etapas a seguir:

1. Abra o portal do Azure. Inicie o Cloud Shell e crie um grupo de recursos, como as `azuremolchapter15`:

```
az group create --name azuremolchapter15 --location eastus
```

2. Crie um cofre de chaves com um nome exclusivo, como `azuremol`, habilite-o para implantação para que você possa usar o cofre para injetar chaves e certificados em uma VM:

```
az keyvault create \
  --resource-group azuremolchapter15 \
```

```
  --name azuremol \
  --enable-soft-delete \
  --enabled-for-deployment
```

Por padrão, sua conta de usuário do Azure recebe permissões completas para o cofre de chaves. Para esses exercícios, isso é suficiente, embora como uma prática recomendada de segurança, você deve considerar limitar quem pode acessar seu cofre de chaves. Você pode adicionar o parâmetro `--no-self-perms` para ignorar a atribuição de permissão à sua conta.

3. Crie um segredo, como `databasepassword` e atribua um valor de senha, como `SecureP@ssw0rd`. (Sim, bem seguro, certo?) Esse segredo pode ser usado como credenciais de um servidor de banco de dados, que você implantará nos seguintes exercícios:

```
az keyvault secret set \
  --name databasepassword \
  --vault-name azuremol \
  --description "Database password" \
  --value "SecureP@ssw0rd"
```

4. Você tem permissões totais para o cofre de chaves, para que você possa exibir o conteúdo do seu segredo:

```
az keyvault secret show \
--name databasepassword \
--vault-name azuremol
```

Sob uma perspectiva de gerenciamento, você também pode executar ações comuns, como backup e restauração, download, atualização e exclusão de itens armazenados em um cofre de chaves. Uma propriedade adicional que você definiu quando o cofre de chaves foi criado é a opção `enable-soft-delete`. Se seus serviços e aplicações não puderem recuperar os segredos que precisam do cofre de chaves, você pode ter que lidar com uma interrupção de aplicação bem longa. Um cofre de chaves pode armazenar metadados para segredos por até 90 dias depois de serem realmente excluídos, o que permite que você recupere dados que são excluídos de forma incorreta ou maliciosa.

5. Exclua a chave que você acabou de criar para simular um erro ou possivelmente alguém com intenção maliciosa:

```
az keyvault secret delete \
  --name databasepassword \
  --vault-name azuremol
```

6. Recupere o segredo para que você possa continuar a usar a senha do banco de dados com sua aplicação e serviços:

```
az keyvault secret recover \
  --name databasepassword \
  --vault-name azuremol
```

Se realmente deseja remover um segredo, você também tem a opção de limpar um segredo excluído. Essa opção remove permanentemente o segredo sem aguardar o período de recuperação padrão de 90 dias.

Fique à vontade para usar `az keyvault secret show` novamente para ver as informações sobre o seu segredo e confirmar que a senha que você armazenou esteja lá depois de exclui-los e restaurá-los. Agora, vamos avançar para ver como uma VM pode acessar um cofre de chaves e usar o segredo para instalar o MySQL Server.

15.2 Identidades gerenciadas para recursos do Azure

A capacidade de usar o Azure Key Vault para armazenar segredos ou chaves é ótima, mas como acessar esses segredos? A CLI do Azure ou o Azure PowerShell pode acessar as informações armazenadas em um cofre de chaves, mas geralmente é mais conveniente permitir que suas VMs ou aplicações recuperem segredos ou chaves diretamente quando necessárias. Uma maneira de fazer isso é com identidades gerenciadas para recursos do Azure, conforme mostrado na Figura 15.4.

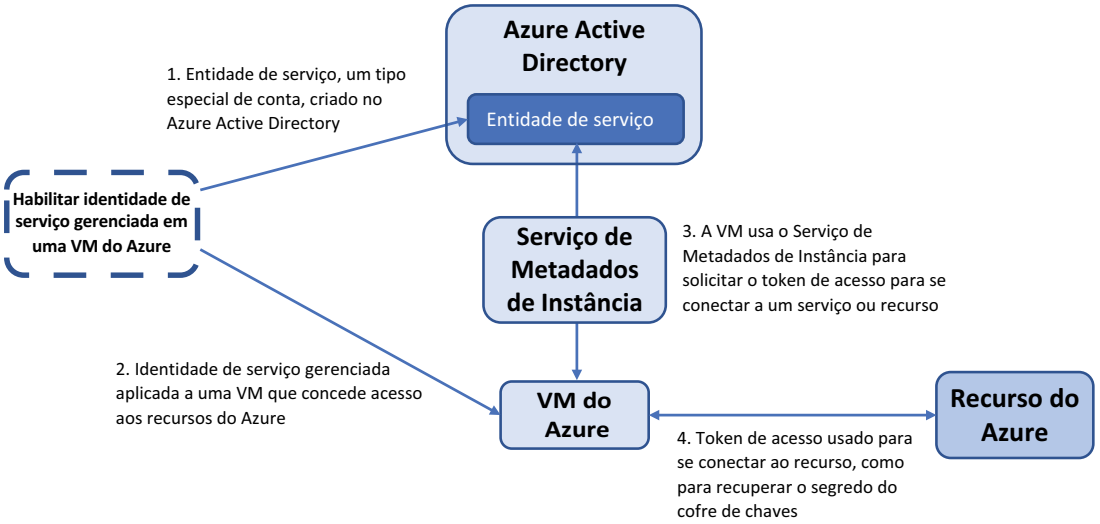


Figura 15.4 Quando você cria uma identidade de serviço gerenciada para uma VM, uma entidade de serviço é criada no Azure Active Directory. Esta entidade de serviço é um tipo especial de conta que pode ser usada para os recursos se autenticarem. Em seguida, esta VM usa o ponto de extremidade do Serviço de Metadados de Instância para fazer solicitações de acesso a recursos. O ponto de extremidade conecta-se ao Azure AD para solicitar tokens de acesso quando a VM precisa solicitar dados de outros serviços. Quando um token de acesso é retornado, ele pode ser usado para solicitar acesso aos recursos do Azure, como um cofre de chaves.

Uma *identidade gerenciada* permite que você crie um tipo especial de conta que pode ser usado por um recurso do Azure, como uma VM. Se você usou um serviço de diretório como o Active Directory, uma conta de computador muitas vezes é usada para identificar e conceder acesso a vários recursos de rede necessários para um computador. Você não cria e usa contas de usuário regulares para esse tipo de autenticação, o que melhora a segurança: você pode conceder um conjunto restritivo de permissões apenas para um computador em vez de também se preocupar com permissões de usuário e acesso a pastas compartilhadas, por exemplo.

Uma identidade gerenciada é como uma conta de computador, mas é armazenado no Azure Active Directory (Azure AD). A identidade, denominada *entidade de serviço*, é exclusiva para cada VM e pode ser usada para atribuir permissões a outros recursos do Azure, como uma conta do Armazenamento do Azure ou um cofre de chaves. A VM tem permissões para acessar esses recursos, para que você possa executar tarefas de script (como a Automação do Azure, que exploraremos no capítulo 18) que não exigem nenhuma intervenção do usuário ou que não exige nomes de usuários e senhas. As VMs autenticam-se, e a plataforma do Azure autoriza o acesso aos recursos atribuídos.

Você pode criar dois tipos de identidades gerenciadas:

- *Atribuído pelo sistema* — Esse tipo de identidade gerenciada é aplicado diretamente a um recurso, como uma VM, e é usado somente por esse recurso. Cada recurso tem sua própria identidade exclusiva quando se trata de auditoria ou solução de problemas de acesso. Quando o recurso é excluído, a identidade gerenciada é excluída automaticamente.
- *Atribuído pelo usuário* — Um recurso separado do Azure é criado e gerenciado para a identidade gerenciada especificada. Essa identidade gerenciada pode ser compartilhada entre outros recursos para definir o acesso. Quando todos os recursos que usam a identidade são excluídos, a identidade gerenciada permanece disponível para uso.

Vamos ver como você pode usar uma identidade gerenciada atribuída pelo sistema para solicitar o segredo `databasepassword` de um cofre de chaves. Uma vez que a VM possa recuperar o segredo, a senha pode ser usada para instalar um servidor de banco de dados MySQL automaticamente. Com um cofre de chaves e MSIs, você pode executar alguns comandos para recuperar o segredo do cofre de chaves, executar o instalador do MySQL Server e fornecer automaticamente a senha segura.

Serviço de Metadados de Instância do Azure

Uma VM habilitada com uma identidade gerenciada usa um ponto de extremidade REST por meio do Serviço de Metadados de Instância (IMDS) para solicitar um token de acesso do Azure AD que ele pode usar para solicitar dados do Azure Key Vault. Mas o que é o Serviço de Metadados de Instância?

IMDS é um ponto de extremidade REST que só é acessível internamente para VMs. O ponto de extremidade está disponível no endereço não roteável 169.254.169.254. Uma VM pode fazer uma solicitação para o ponto de extremidade do IMDS para recuperar informações sobre si mesmo, como a região do Azure ou o nome do grupo de recursos. Essa capacidade permite que a VM entenda como e onde na plataforma do Azure está em execução. O ponto de extremidade do IMDS pode ser acessado de várias linguagens, incluindo Python, C#, Go, Java e PowerShell.

Para eventos de manutenção, o ponto de extremidade do IMDS também pode ser consultado para que a VM fique ciente de um evento de atualização ou reinicialização pendente. Em seguida, todas as tarefas de pré-atualização ou reinicialização que são necessárias podem ser executadas. Como o IMDS é um ponto de extremidade REST em um endereço IP não roteável, não há nenhum agente ou extensão para a VM ser instalada e nenhuma preocupação com segurança de rede ou problemas de roteamento.

Para fins de identidade gerenciada, o ponto de extremidade do IMDS é usado para retransmitir a solicitação de um token de acesso ao Azure AD. Essa abordagem fornece uma maneira segura para que as VMs solicitem acesso sem a necessidade de se comunicar diretamente com o Azure AD.

Experimente agora

Para criar uma VM com um MSI, conclua as etapas a seguir:

1. Crie uma VM do Ubuntu e forneça seu grupo de recursos, como `azuremolchapter15`, e um nome para a VM, como `molvm`. Uma conta de usuário denominada azuremol é criada e as chaves SSH que você usou nos capítulos anteriores são adicionadas à VM:

```
az vm create \
  --resource-group azuremolchapter15 \
  --name molvm \
  --image ubuntults \
  --admin-username azuremol \
  --generate-ssh-keys
```

2. Como prática recomendada de segurança, você não deve permitir que as contas acessem todos os recursos em toda a sua assinatura do Azure. Principalmente para identidades gerenciadas, conceda somente a quantidade mínima necessária de permissões.

   Para este exercício, limite o acesso apenas ao seu grupo de recursos, como `azuremolchapter15`. Você define o escopo consultando a ID do grupo de recursos com `--query id`. Em seguida, essa ID é atribuída a uma variável chamada `scope`:

```
scope=$(az group show --resource-group azuremolchapter15
➥--query id --output tsv)
```

3. Crie uma identidade gerenciada atribuída pelo sistema para a VM com a função de leitura para que só possa ler recursos, e não fazer alterações neles. Limite a identidade ao grupo de recursos. A variável que você criou na etapa anterior que contém a ID do grupo de recursos é fornecida:

```
az vm identity assign \
  --resource-group azuremolchapter15 \
  --name molvm \
  --role reader \
  --scope $scope
```

4. Aplique permissões no Azure Key Vault que concede acesso à entidade de serviço para a identidade gerenciada. Você pode fazer isso por meio do portal em Políticas de acesso para o recurso do Key Vault ou pode usar a CLI do Azure. Vamos usar a CLI para ver como obter as informações de forma programática.

   Primeira, obtenha informações sobre a entidade de serviço do Azure AD para sua identidade gerenciada. Filtre o `display-name` da VM criada na etapa 3, como `molvm`:

```
az ad sp list \
  --display-name molvm \
  --query [].servicePrincipalNames
```

A saída é semelhante ao seguinte exemplo resumido. Não se preocupe muito com o que esses valores significam. Você não precisa trabalhar com eles além da atribuição das permissões iniciais aqui. Novamente, você poderá usar o portal do Azure para evitar a CLI se estiver desconfortável.

Anote o primeiro `servicePrincipalName`. Este valor é usado para atribuir permissões em recursos do Azure, como seu cofre de chaves, e é necessário na próxima etapa:

```
[
  "887e9665-3c7d-4142-b9a3-c3b3346cd2e2",
  "https://identity.azure.net//
  ➥ihxXtwZEiAeNXU8eED2Ki6FXRPkklthh84S60CiqA4="
]
```

5 Defina a política no cofre de chaves, de modo que a entidade de serviço da sua VM possa ler segredos e insira seu primeiro `servicePrincipalName` da etapa 4:

```
az keyvault set-policy \
  --name azuremol \
  --secret-permissions get \
  --spn 887e9665-3c7d-4142-b9a3-c3b3346cd2e2
```

Um ponto aqui é que quando a identidade gerenciada foi criada e limitada ao grupo de recursos, isso não significa que a VM pudesse fazer qualquer coisa que quisesse. Primeiro, a única função criada para a identidade era a leitura de permissões de recursos. Porém, você ainda tinha que atribuir permissões para o cofre de chaves em si. Essas camadas de segurança e permissões dão a você um controle refinado sobre os recursos exatos que cada identidade pode acessar.

Agora que você tem acesso a um cofre de chaves, provavelmente quer saber como recuperar o segredo, certo?

15.3 Obter um segredo de uma VM com identidade de serviço gerenciado

Você armazenou um segredo em um cofre de chaves para uma senha de banco de dados, e você tem uma VM com uma identidade gerenciada que fornece acesso para ler esse segredo no cofre de chaves. E agora? Como recuperar e usar o segredo? A figura 15.5 mostra como uma VM usa o IMDS para solicitar acesso a um recurso, tal como um cofre de chaves. Vamos conferir as etapas para ver como a VM recupera o segredo.

A maioria dos casos de uso para o Azure Key Vault não teria uma VM conectando e recuperando os segredos dessa forma. O Key Vault é realmente útil quando as aplicações em si, dentro do código, acessam para recuperar segredos. O código da aplicação usaria o SDK do Azure apropriado, como Python, .Net ou Java. Para evitar complexidades do código abstraindo o que está acontecendo, o exercício a seguir usa uma VM e um trabalho de linha de comando. Ao trabalhar neste exercício, lembre-se de que essa magia normalmente aconteceria no código da aplicação.

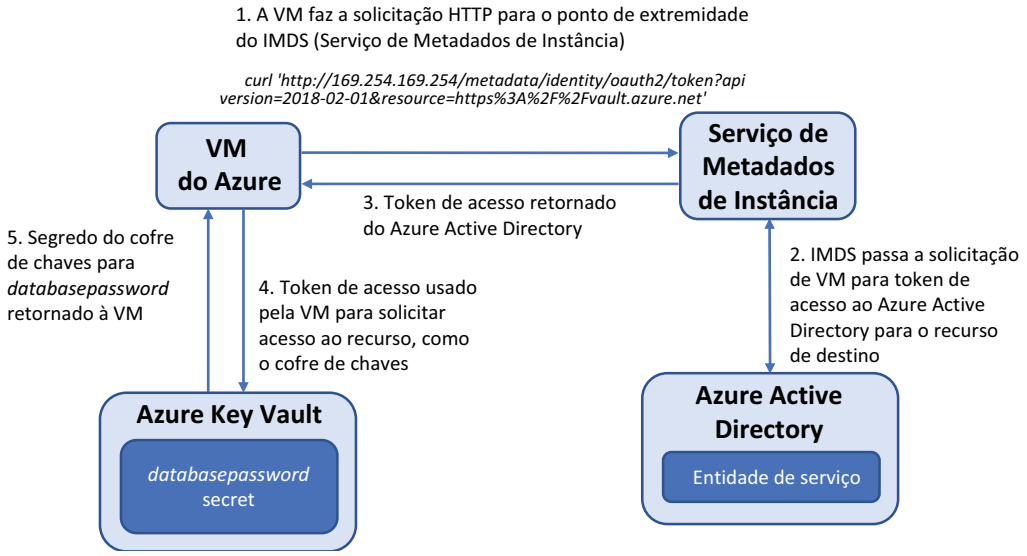


Figura 15.5 A VM usa o IMDS para solicitar acesso a um cofre de chaves. O ponto de extremidade comunica-se com o Azure AD para solicitar um token de acesso. O token de acesso é retornado para a VM, que é usado para solicitar acesso do cofre de chaves. Se o acesso for concedido pelo cofre de chaves, o segredo para `databasepassword` será retornado para a VM.

Experimente agora

Para recuperar e usar um segredo em uma VM com uma identidade gerenciada, conclua as etapas a seguir:

1. Obtenha o endereço IP público da VM criada no exercício anterior, como `molvm`:

```
az vm show \
  --resource-group azuremolchapter15 \
  --name molvm \
  --show-details \
  --query [publicIps] \
  --output tsv
```

2. Faça SSH para sua VM, como `ssh azuremol@publicIps`.
3. Para acessar um cofre de chaves, você precisa de um token de acesso. Esse token de acesso é solicitado do IMDS. É uma solicitação HTTP, e em uma VM do Linux você pode usar o programa `curl` para fazer a solicitação. O IMDS passa sua solicitação para o AAD:

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?
➥api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
➥-H Metadata:true
```

4 A saída é um pouco difícil de ler, porque ela se parece com um emaranhado de texto. Ela está no formato JSON Web Token (JWT). Para processar a saída JSON e tornar as coisas mais legíveis, instale um analisador JSON chamado `jq`:

```
sudo apt-get update && sudo apt-get -y install jq
```

5 Faça sua solicitação `curl` novamente, mas desta vez veja a saída com `jq`:

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?
➥api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
➥-H Metadata:true --silent | jq
```

Estas primeiras etapas mostram como as solicitações são feitas e como é a saída, como mostrado na Figura 15.6. Se você ainda fizer login na VM e solicitar manualmente um token de acesso, qual é o sentido de usar uma identidade gerenciada? Você poderia apenas fornecer suas próprias credenciais. Em produção, você provavelmente usaria um script executado na VM para fazer a solicitação de um token de acesso automaticamente e, em seguida, recuperar o segredo do cofre de chaves. Vamos continuar para ver como automatizar esse processo e recuperar o segredo.

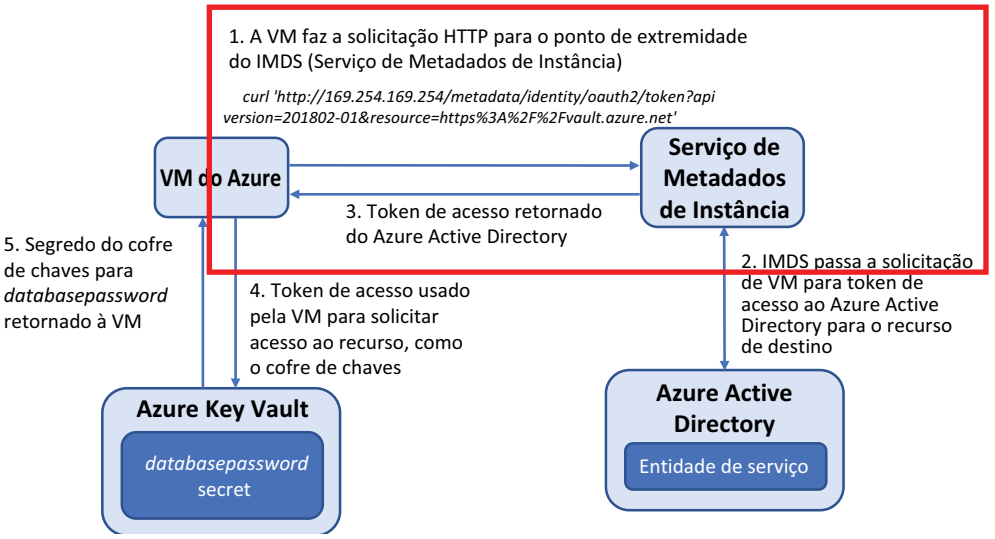


Figura 15.6 A solicitação `curl` cobre as três primeiras etapas deste diagrama. A solicitação `curl` é feita, o ponto de extremidade comunica-se com o Azure AD, e um token de acesso é emitido.

6 Para facilitar as coisas (e se você fosse fazer tudo isso em um script), você pode usar o `jq` para processar a resposta do curl, extrair somente o token de acesso e defini-la como uma variável chamada `access_token`:

```
access_token=$(curl
➥'http://169.254.169.254/metadata/identity/oauth2/token?
➥api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
➥-H Metadata:true --silent | jq -r '.access_token')
```

7 Como uma etapa manual para ajudá-lo a entender como é isso, veja a variável `access_token`:

```
echo $access_token
```

8 Agora, a parte divertida. Use o token de acesso para solicitar seu segredo no cofre de chaves. Vamos primeiro fazer isso manualmente para que você entenda o que acontece.

9 Recupere o segredo com outra solicitação `curl` formate a saída com `jq`. Insira seu próprio nome do cofre de chaves no início do https:// address:

```
curl https://azuremol.vault.azure.net/secrets/databasepassword?
➥api-version=2016-10-01 -H "Authorization: Bearer $access_token"
➥--silent | jq
```

A saída é semelhante à seguinte, que mostra o valor da senha armazenada no segredo, juntamente com alguns metadados adicionais (com os quais você não precisa se preocupar por enquanto) sobre o segredo:

```
{
  "value": "SecureP@ssw0rd!",
  "contentType": "Database password",
  "id":
➥"https://azuremol.vault.azure.net/secrets/databasepassword/
➥87e79e35f57b41fdb882c367b5c1ffb3",
}
```

Esta `curl` é a segunda parte do fluxo de trabalho, como mostrado na Figura 15.7.

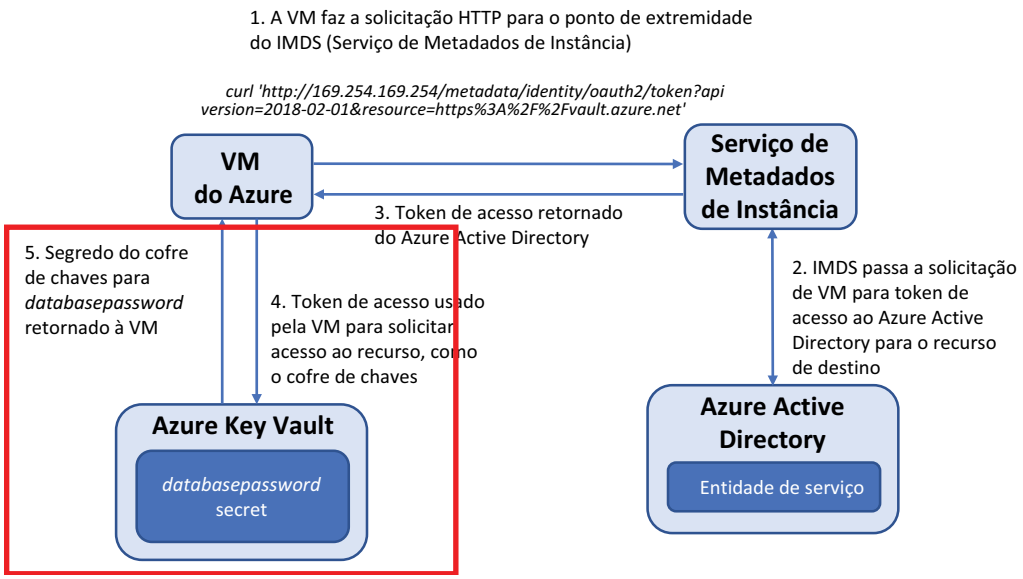


Figura 15.7 Esta segunda solicitação `curl` aborda as duas últimas etapas no diagrama. O token de acesso é usado para solicitar o segredo no cofre da chave. A resposta JSON é retornada, incluindo o valor do segredo.

10 Da mesma forma que você usou uma variável para armazenar o token de acesso, você pode, em um script, atribuir também o valor do segredo a uma variável. Desta vez, use `jq` para processar a resposta, extraia apenas o valor do segredo e defina-o como uma variável `database_password`:

```
database_password=$(curl
➥https://azuremol.vault.azure.net/secrets/databasepassword?
➥api-version=2016-10-01 -H "Authorization: Bearer $access_token"
➥--silent | jq -r '.value')
```

11 De novo, como uma etapa manual para ajudá-lo a entender o processo, visualize o conteúdo da variável `database_password`:

```
echo $database_password
```

Espero que você esteja acompanhando. Por exemplo, se você escrever uma aplicação em Python, ASP.NET ou Node. js, o processo será semelhante a fazer uma solicitação para o token de acesso e, em seguida, usar o token para solicitar um segredo de um cofre de chaves. Há provavelmente outras bibliotecas que você poderia usar em seu código em vez do utilitário `jq` na linha de comando.

Como uma recapitulação rápida, todas essas etapas podem ser resumidas em duas linhas, como mostrado na listagem a seguir.

Listagem 15.1 Solicitar um token de acesso e, em seguida, um segredo de um cofre de chaves

```
access_token=$(curl
➥'http://169.254.169.254/metadata/identity/oauth2/token?
➥api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
➥-H Metadata:true --silent | jq -r '.access_token')
database_password=$(curl
➥https://azuremol.vault.azure.net/secrets/databasepassword?
➥api-version=2016-10-01 -H "Authorization: Bearer $access_token"
➥-silent | jq -r '.value')
```

E agora? A identidade gerenciada para sua VM pode recuperar um segredo de um cofre de chaves. Vamos ver como você pode usar essa identidade gerenciada para instalar e configurar o MySQL Server.

No Ubuntu, você pode definir seleções de configuração para instaladores de pacotes, como o MySQL Server. Estas seleções de configuração permitem que você forneça valores como nomes de usuário e senhas e que eles sejam usados automaticamente na parte relevante do processo de instalação. Os prompts manuais para fornecer uma senha, como você pode ter visto de volta no capítulo 2, não aparecem mais.

12 Defina as seleções de configuração para as senhas do servidor MySQL com a variável `database_password` que você criou na etapa 10:

```
sudo debconf-set-selections <<< "mysql-server mysql-server/root_password
➥password $database_password"
sudo debconf-set-selections <<< "mysql-server mysql-
➥server/root_password_again password $database_password"
```

13 Instale o MySQL Server. Não há prompts, porque a senha é fornecida pelas seleções de configuração:

```
sudo apt-get -y install mysql-server
```

14 Vamos comprovar que tudo isso funcionou. Visualize a variável `database_password` para que possa ver claramente qual sua senha deve ser:

```
echo $database_password
```

15 Faça login no MySQL Server. Quando for solicitada uma senha, insira o valor de `database_password`, que é o valor do segredo do cofre de chaves:

```
mysql -u root -p
```

Você está conectado ao servidor MySQL, que confirma que o segredo do cofre de chaves foi usado para criar com êxito as credenciais do SQL Server.

16 Digite `exit` duas vezes para fechar o prompt de comando do servidor MySQL e, em seguida, feche a sessão SSH da VM.

Este exemplo foi básico. Você ainda precisa proteger o servidor MySQL e fornecer credenciais adicionais, por exemplo, para que aplicações acessem bancos de dados ou tabelas. A vantagem de usar um segredo de um cofre de chaves é que você garante que todas as senhas sejam as mesmas. Por exemplo, se você usar conjuntos de escala de máquinas virtuais, cada instância de VM poderá solicitar automaticamente o segredo e instalar o servidor MySQL para que ele esteja pronto para atender aos dados da sua aplicação. Essas senhas nunca são definidas em scripts, e ninguém precisa ver quais são as senhas. Você pode até mesmo gerar senhas aleatoriamente e variá-las como segredos em um cofre de chaves.

Armazenar senhas em um cofre de chaves é ótimo, mas você pode usar um cofre de chaves para armazenar certificados e recuperá-los automaticamente de suas aplicações ou VMs? É claro que sim.

15.4 Criar e injetar certificados

Certificados digitais são uma forma comum de segurança e autenticação em aplicações e serviços Web. Os certificados são emitidos por uma autoridade de certificação (CA), em que os usuários finais confiam (assim esperamos). O certificado permite aos usuários verificar se um site ou uma aplicação é realmente quem diz ser. Toda vez que você vê um site com um endereço de navegador da Web que começa com https:// e tem um símbolo de cadeado, o tráfego é criptografado e protegido por um certificado digital.

Gerenciar certificados digitais pode se tornar uma das principais tarefas de gerenciamento. Um problema comum é como armazenar e conceder acesso a certificados, pois serviços e aplicações precisam deles. Nos exercícios anteriores, examinamos como um cofre de chaves pode ser usado para compartilhar segredos e chaves seguras com serviços e aplicações, mas um cofre de chaves pode fazer o mesmo com certificados. Como mostrado na Figura 15.8, um cofre de chaves pode ser usado para solicitar, emitir e armazenar certificados.

Em produção, você sempre deve usar uma autoridade de certificação confiável para emitir seus certificados. Para uso interno, você pode emitir certificados autoassinados criados por você mesmo. Esses

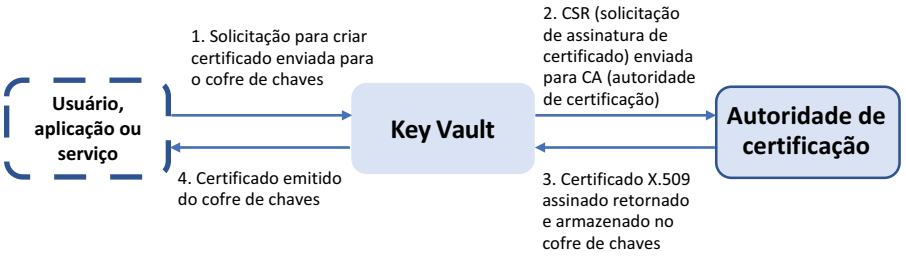


Figura 15.8 Um usuário, aplicação ou serviço pode solicitar um novo certificado de um cofre de chaves. Uma solicitação de assinatura de certificado (CSR) é enviada pelo cofre de chaves para uma CA externa de terceiros ou uma CA interna confiável. O Azure Key Vault também pode atuar como sua própria autoridade de certificação para gerar certificados autoassinados. Em seguida, a CA emite um certificado X. 509 assinado, que é armazenado no cofre de chaves. Por fim, o cofre de chaves retorna o certificado para o solicitante original.

certificados autoassinados não são confiáveis por outros serviços e aplicações e, portanto, eles normalmente geram um aviso, mas os certificados autoassinados permitem que você rapidamente execute e verifique se seu código funciona conforme o esperado com o tráfego criptografado.

O Azure Key Vault pode gerar certificados autoassinados para você. Nos bastidores, o Key Vault atua como sua própria CA para solicitar, emitir e, em seguida, armazenar certificados. Vamos usar essa habilidade para gerar um certificado autoassinado e ver como injetá-lo facilmente em uma VM. O certificado é então usado para um servidor Web básico para mostrar como habilitar rapidamente o SSL para proteger o tráfego da Web.

Experimente agora

Para criar e injetar um certificado em uma VM, conclua as etapas a seguir:

1. Crie um certificado autoassinado no Azure Key Vault e insira um nome, como `molcert`. As políticas são usadas para definir propriedades, como períodos de tempo de expiração, força de criptografia e formato do certificado. Você pode criar políticas diferentes para atender às necessidades de suas aplicações e serviços. Para este exercício, use a política predefinida que cria um certificado de 2.048 bits e é válida por um ano:

```
az keyvault certificate create \
  --vault-name azuremol \
  --name molcert \
  --policy "$(az keyvault certificate get-default-policy)"
```

2. Para ver o certificado em ação, crie outra VM, como `molwinvm`. Desta vez, crie uma VM do Windows que usa o Windows Server 2019, para que você espalhe o amor do sistema operacional e veja que esses recursos do Key Vault não dependem de um sistema operacional específico. Forneça seu próprio nome de usuário e senha de administrador:

```
az vm create \
  --resource-group azuremolchapter15 \
```

```
  --name molwinvm \
  --image win2019datacenter \
  --admin-username azuremol \
  --admin-password P@ssw0rd1234
```

3 Você pode adicionar automaticamente o certificado à VM diretamente da CLI do Azure. Esta abordagem não depende de uma identidade gerenciada. A plataforma do Azure injeta o certificado usando o agente da VM do Microsoft Azure.

Adicione seu certificado, como `molcert`, à VM criada na etapa 2, como `molwinvm`:

```
az vm secret add \
  --resource-group azuremolchapter15 \
  --name molwinvm \
  --keyvault azuremol \
  --certificate molcert
```

4 Conecte-se à VM e verifique se o certificado foi injetado corretamente. Para se conectar à sua VM, primeiro obtenha seu endereço IP público:

```
az vm show \
  --resource-group azuremolchapter15 \
  --name molwinvm \
  --show-details \
  --query [publicIps] \
  --output tsv
```

Use um cliente de conexão local do Microsoft Remote Desktop no seu computador para se conectar à sua VM. Use as credenciais para se conectar a localhost\azuremol, não as credenciais padrão do seu computador local que o cliente da Área de Trabalho Remota pode tentar usar, como mostrado na Figura 15.9.

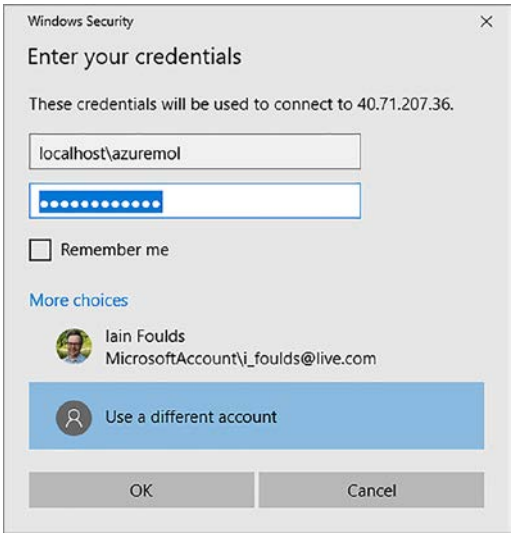


Figura 15.9 O cliente da Área de Trabalho Remota pode tentar usar as credenciais do seu computador local padrão. Em vez disso, selecione Usar uma conta diferente e forneça as credenciais localhost\azuremol especificadas quando você criou a VM.

5 Quando você estiver conectado, selecione o botão Iniciar do Windows e digite `mmc` e abra o Console de Gerenciamento Microsoft.

6 Escolha Arquivo > Adicionar / Remover snap-in e, em seguida, selecione a opção para adicionar os Certificates snap-in.

7 Escolha add certificados for the Computer account, selecione Next e, em seguida, selecione Concluir.

8 Escolha OK para fechar a janela Adicionar / Remover snap-in.

9 Expanda a pasta Certificados (computador local) > Pessoal > Certificados. O certificado do Azure Key Vault que você injetou na VM aparece listado, como CLIGetDefaultPolicy, como mostrado na Figura 15.10.

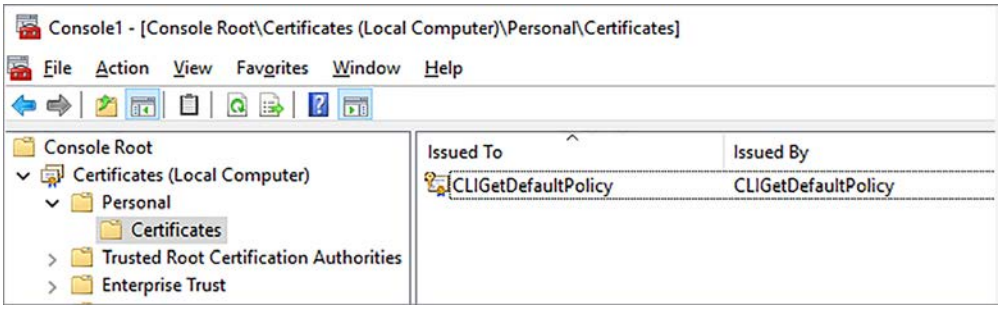


Figura 15.10 No Console de Gerenciamento Microsoft, adicione o Certificados snap-in no computador local. Expanda o armazenamento Pessoal > Certificados para exibir os certificados instalados. O certificado injetado do Key Vault está listado.

E é só isso. Crie o certificado no Key Vault e, em seguida, adicione o certificado à VM. O certificado é colocado no armazenamento de certificados local do computador, que permite que qualquer serviço ou aplicação possa acessá-lo. Em uma VM do Windows, os certificados são armazenados no cache de certificado local, como mostrado neste exercício. Em VMs do Linux, arquivos .prv e .crt para as partes pública e privada do certificado são armazenados em/var/lib/waagent/. Você pode mover os certificados para onde você precisa para sua aplicação ou serviço.

Os certificados podem ser usados para autenticação entre clientes e servidores ou entre componentes e serviços de aplicações. Um exemplo comum é que um servidor Web use um certificado SSL, que é o que você vai fazer no laboratório de fim de capítulo.

15.5 Laboratório: Configurar um servidor Web seguro

No último exercício, você injetou um certificado autoassinado do Azure Key Vault em uma VM do Windows. Para este laboratório, instale e configure o servidor Web do IIS para usar o certificado, seguindo estas orientações:

1 Abra o PowerShell na sua VM do Windows e instale o servidor Web do IIS:

```
Add-WindowsFeature Web-Server -IncludeManagementTools
```

2. Abra o Gerenciador do Servidor de Informações da Internet (IIS). Você pode fazer isso no menu Ferramentas no Gerenciador do Servidor.
3. Para o site padrão, escolha Editar vinculações.
4. Adicione uma vinculação HTTPS a todos os endereços IP não atribuídos na porta 443.
5. Selecione o certificado autoassinado criado e injetado do Key Vault, normalmente algo como CLIGetDefaultPolicy.
6. Abra um navegador da Web na VM e insira `https://localhost`. Você gerou um certificado autoassinado no Key Vault, para que o navegador da Web não confie nele.
7. Aceite o aviso para continuar e verifique se a vinculação HTTPS funciona.
8. De volta ao Azure Cloud Shell ou ao portal do Azure, crie uma regra de NSG para a VM na porta TCP 443. Insira `https://yourpublicipaddress` em um navegador da Web em seu computador local. Esta é a experiência que os usuários teriam, com um aviso sobre um certificado autoassinado não confiável. Para a maioria dos casos de uso, lembre-se de usar uma CA interna ou de terceiros confiável para gerar certificados confiáveis e armazená-los em um cofre de chaves.

16 Central de Segurança do Azure e atualizações

Não seria ótimo se o Azure fosse inteligente o suficiente para monitorar todos os recursos principais da sua aplicação e alertá-lo sobre as preocupações de segurança? Ou e se o seu negócio tivesse políticas de segurança já definidas? (Se você não tiver nenhuma política de segurança, pare agora e crie um lembrete para criar algumas.) Neste último caso, como garantir que suas implantações do Azure permaneçam em conformidade? Se já passou por uma auditoria de segurança de TI, você sabe como pode ser divertido olhar uma lista de configurações incorretas aplicadas ao seu ambiente, principalmente os lapsos de segurança básica que você conhece.

A Central de Segurança do Azure fornece um local centralizado para que alertas e recomendações de segurança sejam agrupados para sua revisão. Você pode definir suas próprias políticas de segurança e permitir que o Azure monitore o estado de seus recursos para conformidade.

Neste capítulo, discutiremos como a Central de Segurança pode alertá-lo de problemas e fornecer etapas para corrigi-los, como você pode usar o acesso just-in-time à VM para controlar e auditar conexões remotas, e como o Gerenciamento de atualizações mantém suas VMs atualizadas com os patches de segurança mais recentes.

16.1 A Central de Segurança do Azure

Ao longo deste livro, discutimos tópicos relacionados à segurança, por exemplo, como criar e configurar grupos de segurança de rede (NSGs) para restringir o acesso a VMs e como permitir apenas o tráfego criptografado em contas do Armazenamento do Azure. Para suas próprias implantações além dos exercícios neste livro, como você sabe por onde começar, e como você pode verificar que aplicou todas as práticas recomendadas de segurança? É aí que a Central de Segurança do Azure pode ajudar, verificando seu ambiente em áreas que você pode ter deixado passar.

A Central de Segurança do Azure examina seus recursos, recomenda correções e ajuda a tratar as preocupações de segurança, como mostrado na Figura 16.1. Quando você tiver apenas algumas VMs de teste e uma única rede virtual em sua assinatura do Azure, talvez não pareça difícil controlar quais restrições de segurança você precisa implementar. Porém, à medida em que você expande até dezenas, centenas ou até milhares de VMs, manter manualmente o controle de quais configurações de segurança precisam ser aplicadas a cada VM torna-se não gerenciável.

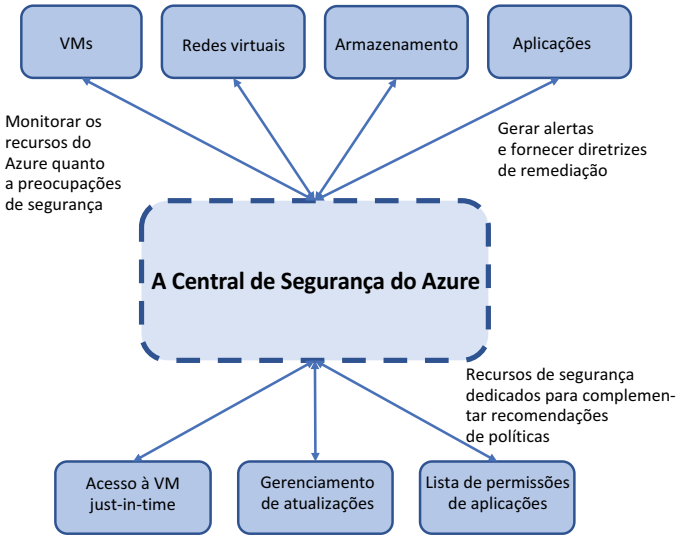


Figura 16.1 A Central de Segurança do Azure monitora seus recursos do Azure e usa políticas de segurança definidas para alertá-lo sobre possíveis ameaças e vulnerabilidades. São fornecidas recomendações e etapas para corrigir problemas. Você também pode usar o acesso just-in-time à VM, monitorar e aplicar atualizações de segurança e controlar aplicações permitidas que podem ser executadas nas VMs.

A Central de Segurança também pode alertá-lo sobre as práticas recomendadas gerais, como se uma VM não tiver o diagnóstico habilitado. Lembre-se de quando vimos no capítulo 12 sobre como monitorar e solucionar problemas de VMs? Você precisa instalar e configurar o agente de diagnóstico *antes* que tenha um problema. Se você suspeitar de uma violação de segurança, talvez não consiga acessar a VM e revisar os logs. Porém, se você configurou a extensão de diagnóstico para transmitir logs para o Armazenamento do Azure, poderá revisar o que ocorreu e (se tudo der certo) rastrear a origem e a extensão do problema.

Experimente agora

Para começar com a Central de Segurança do Azure, conclua as etapas a seguir:

1. Abra o portal do Azure e escolha o ícone do Cloud Shell no menu superior.
2. Crie um grupo de recursos, forneça um nome de grupo de recursos, como `azuremolchapter16` e forneça um local, como `eastus`:

```
az group create --name azuremolchapter16 --location eastus
```

3 Crie uma VM do Linux básica para que a Central de Segurança tenha algo para monitorar e fornecer recomendações:

```
az vm create \
  --resource-group azuremolchapter16 \
  --name azuremol \
  --image ubuntults \
  --admin-username azuremol \
  --generate-ssh-keys
```

4 Quando a VM for implantada, feche o Cloud Shell.

5 No portal do Azure, selecione Central de Segurança na lista de serviços à esquerda. A primeira vez que o painel é aberto, leva alguns segundos para preparar todos os componentes disponíveis. Consulte a Figura 16.2.

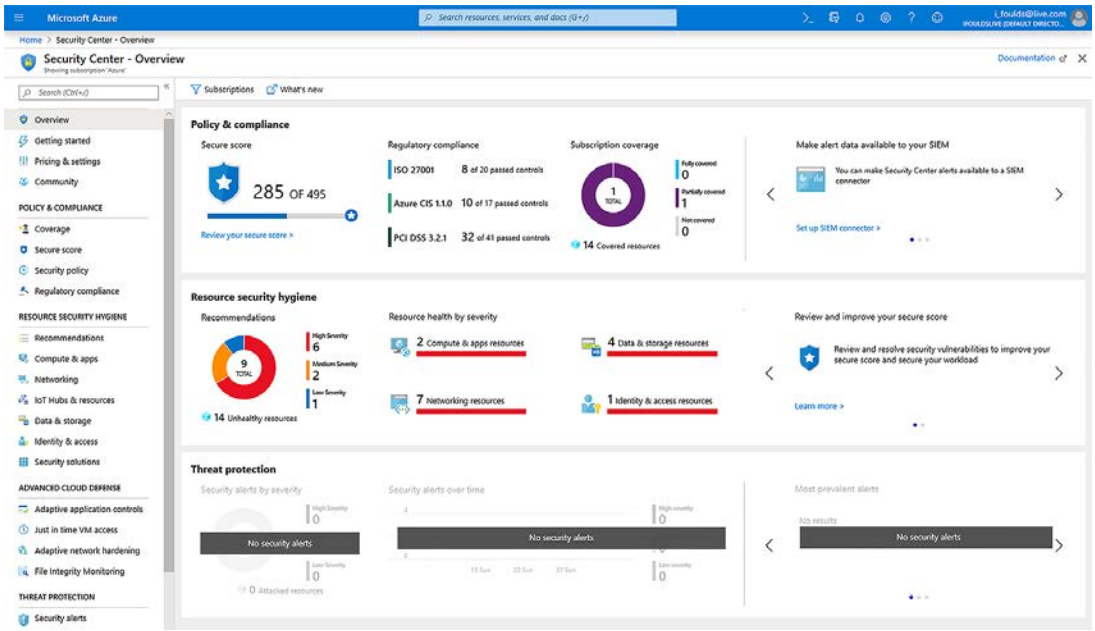


Figura 16.2 A janela Visão geral da Central de Segurança do Azure fornece uma lista de recomendações, alertas e eventos. Você pode selecionar um tipo de recurso principal, como Computação ou Rede, para exibir uma lista de itens de segurança específicos para esses recursos.

A Central de Segurança examina como recursos como VMs, regras de NSG e armazenamento são implantados. As linhas de base de segurança incorporadas são usadas para identificar problemas e fornecer recomendações. Por exemplo, a rede virtual implantada com sua VM gera avisos, como mostrado na Figura 16.3. Você pode e deve implementar suas próprias políticas de segurança que digam ao Azure como deseja restringir o acesso ou o que precisa ser feito para cumprir as obrigações de negócios. Em seguida, ao criar ou atualizar recursos, o Azure

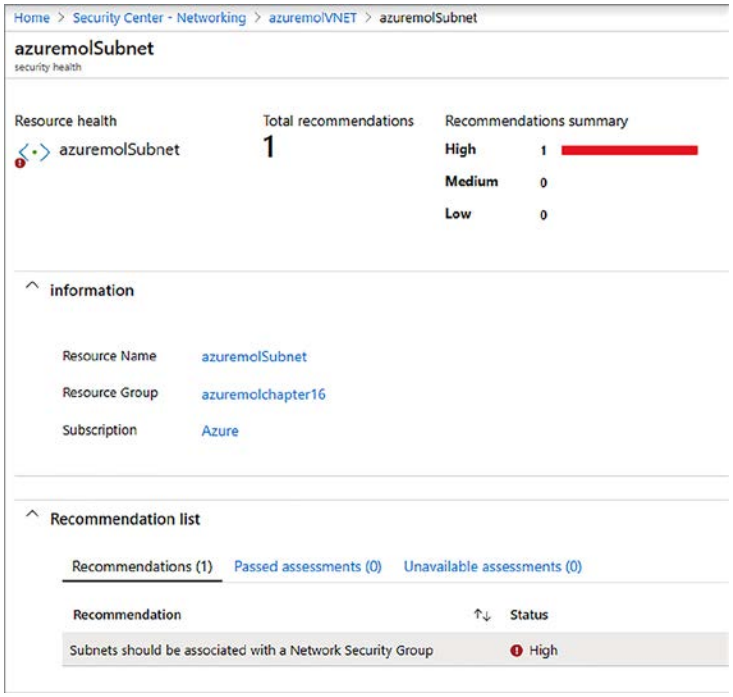


Figura 16.3 A rede virtual para sua VM já aciona avisos de segurança. Neste exemplo, avisa que um grupo de segurança de rede deve ser associado à sub-rede.

monitora constantemente os desvios dessas políticas e o alerta sobre quais etapas precisam ser tomadas para corrigir os problemas de segurança. Você usará as políticas de segurança padrão do Azure neste capítulo, mas considere qualquer configuração de segurança específica que você queira aplicar às suas VMs e como elas podem ser definidas em suas próprias políticas personalizadas.

6 Escolha Computação e aplicativo no menu esquerdo da janela da Central de Segurança. Em seguida, escolha VMs e Computadores

7 Selecione a VM que você criou na etapa 3. Mesmo que você tenha criado essa VM e usado valores padrão da CLI do Azure, alguns avisos de segurança são mostrados.

Explore algumas dessas recomendações. Ao selecionar cada recomendação, algumas apenas fornecem mais informações; outras pessoas orientam você sobre a remediação. Essas não são regras rígidas e rápidas; são recomendações e práticas recomendadas. Em seu próprio ambiente, algumas delas podem não fazer sentido. Porém, são um bom ponto de partida para saber quais coisas você deve fazer para proteger os recursos à medida que os cria no Azure.

16.2 Acesso just-in-time

Na seção 16.1, você aprendeu como a Central de Segurança sugere que você limite o escopo da conectividade remota de entrada. Você pode fornecer um intervalo de IP para limitar o tráfego, mas, preferencialmente, abra a conectividade de entrada somente quando necessário. Dessa forma, a VM é completamente fechada para conexões remotas e é acessível apenas por um curto período de tempo quando

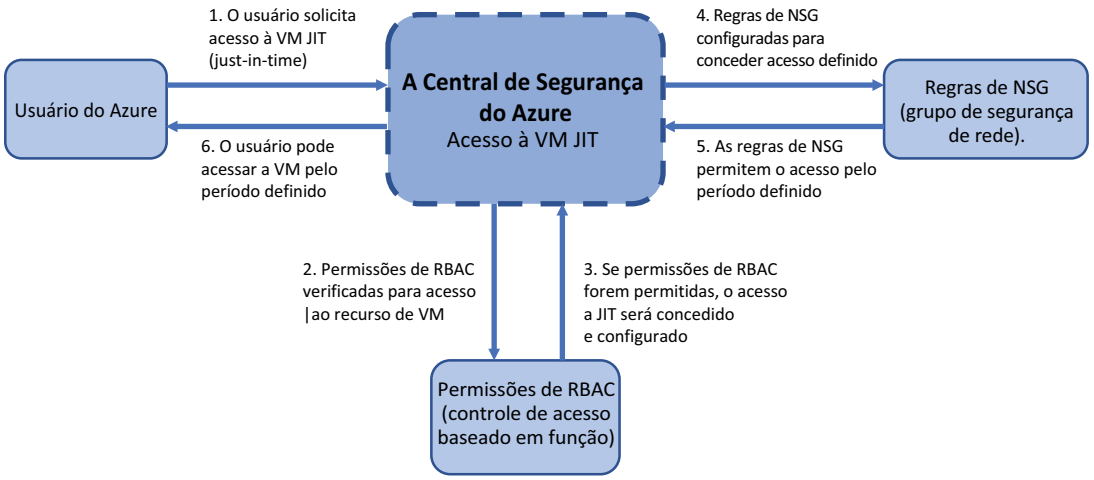


Figura 6.4 Com o acesso JIT VM, as regras de NSG são configuradas para negar conexões remotas a uma VM. As permissões RBAC são usadas para verificar permissões quando um usuário solicita acesso a uma VM. Essas solicitações são auditadas e, se a solicitação for concedida, as regras de NSG serão atualizadas para permitir o tráfego de um determinado intervalo de IP por um período definido. O usuário pode acessar a VM somente durante esse tempo. Quando o tempo expirar, as regras de NSG automaticamente revertem para um estado de negação.

necessário. E, sim, você ainda deve limitar essa breve janela de conectividade para um intervalo de IP específico. É aí que o acesso just-in-time (JIT) à VM é útil, como mostrado na Figura 16.4.

Com o acesso JIT, a Central de Segurança ajusta de forma dinâmica as restrições de acesso em uma VM. Quando habilitado, são criadas regras de NSG que negam todo o tráfego de conexão remota. Em seguida, um usuário pode solicitar acesso a uma VM somente quando necessário. Em combinação com o controle de acesso baseado em função (discutido no capítulo 6), a Central de Segurança determina se um usuário tem direitos para acessar uma VM quando eles solicitam uma conexão. Se o usuário tiver permissões, a Central de Segurança atualizará as regras de NSG relevantes para permitir o tráfego de entrada. Essas regras são aplicadas somente em uma janela de tempo específica. Quando esse tempo é esgotado, as regras são revertidas e a VM fica novamente fechada para conexões remotas. Se tiver uma conexão ativa com uma VM, você não será desconectado automaticamente quando o tempo expirar. Você pode concluir sua manutenção ou solução de problemas e desconectar quando estiver pronto, mas não poderá iniciar uma nova conexão, a menos que solicite acesso o JIT novamente.

Sobrecarregar de trabalho

Não analisamos o Firewall do Azure, mas é um recurso de rede virtual um pouco mais semelhante a um firewall físico na infraestrutura local do que NSGs por si só. Se você precisa de mais flexibilidade e controle de tráfego, o Firewall do Azure é uma ótima opção, embora um custo esteja associado a ele.

Sem se aprofundar muito no Firewall do Azure, quero observar que a Central de Segurança do Azure também pode ser integrado ao Firewall do Azure para abrir e fechar as regras necessárias. Se

usar o Firewall do Azure para proteger o tráfego de VM em redes virtuais, não apenas NSGs, você ainda pode usar o gerenciamento automatizado de regras de acesso JIT da VM.

Para saber mais sobre o Firewall do Azure, consulte o documentos em https://docs.microsoft.com/azure/firewall/overview.

Quando você usaria o JIT em sua pizzaria fictícia? Pense em qualquer VM que executaria seu aplicativo Web, sistema de pedidos ou aplicações de lógica de negócios. Você quer que estes estejam conectados à Internet e disponíveis para que as pessoas acessem o tempo todo? Espero que não. Há razões válidas para o acesso remoto com SSH ou RDP, mas sempre tente minimizar o tempo que o acesso está disponível. Mesmo que você tenha regras de NSG que restrinjam o acesso a determinados intervalos de IP, o JIT adiciona outra camada de proteção em termos do que os usuários do Azure podem acessar e, em seguida, cria uma trilha de auditoria mais fácil sobre a qual a Central de Segurança pode fornecer relatórios.

Experimente agora

Para habilitar o acesso JIT da VM, conclua as etapas a seguir:

1. Abra o portal do Azure e escolha Central de Segurança no menu à esquerda.
2. Em Defesas avançadas da nuvem, selecione Acesso just in time da VM.
3. Se solicitado, escolha a opção Tentar acesso just in time da VM ou Atualizar para camada padrão da Central de Segurança. Este trial gratuito dura 60 dias e não deve ser prologando automaticamente. Isso se sobrepõe à sua conta gratuita do Azure e não custará nada a mais. Selecione a opção Aplicar plano padrão e aguarde alguns instantes para que ela seja habilitada. Quando habilitada, talvez seja necessário fechar e reabrir o portal do Azure antes de concluir as etapas a seguir.
4. Selecione novamente Acesso just in time da VM na janela da Central de Segurança. Quando sua conta de camada padrão estiver habilitada, você poderá exibir uma lista de VMs a serem usadas.
5. Selecione sua VM e, em seguida, escolha Solicitar acesso, como mostrado na Figura 16.5.

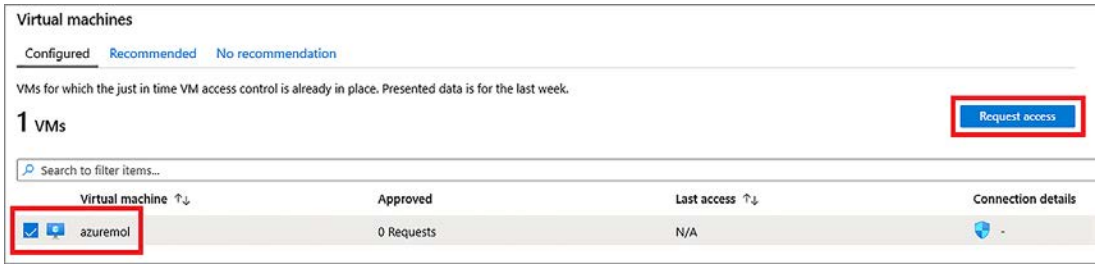


Figura 16.5 Selecione VM nas opções recomendadas e, em seguida, escolha Habilitar JIT em 1 VM. Estado agora mostra que essa VM está aberta para todo o acesso remoto, que sinaliza a severidade da preocupação de segurança como High (alta).

Por padrão, o JIT define regras que podem abrir portas para SSH (porta 22), RDP (porta 3389) e comunicação remota do PowerShell (portas 5985 e 5986) por um período de três horas.

6 Para este exercício, opte por habilitar o SSH de seu próprio IP. Como uma prática recomendada para uso de produção, insira uma justificativa para acompanhar por que o acesso está sendo solicitado. Deixe todos os padrões e escolha Abrir portas, como mostrado na Figura 16.6.

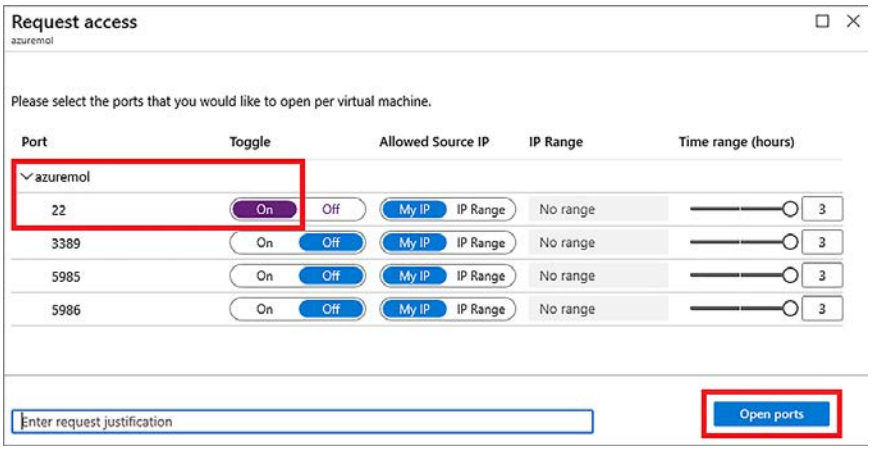


Figura 16.6 Ao habilitar o JIT, você pode alterar para que as regras padrão sejam permitidas, os IPs de origem permitidos e um tempo máximo, em horas, de solicitação. Essas regras JIT permitem o controle granular sobre o que é permitido, para permitir apenas o mínimo de conectividade.

7 Com o JIT habilitado, navegue até o grupo de recursos e selecione sua VM.

8 Escolha Rede para exibir a configuração de rede virtual atribuída para a VM. A lista de regras de NSG atribuídas é exibida, como na Figura 16.7.

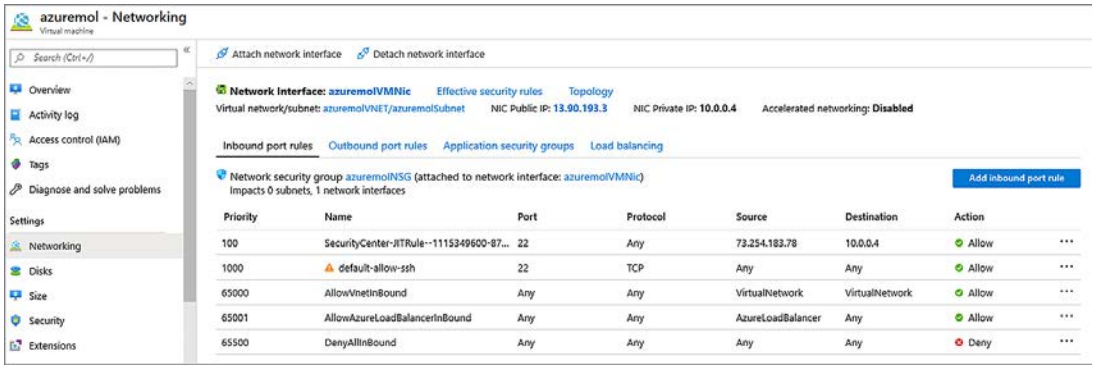


Figura 16.7 As regras JIT são criadas com a prioridade mais baixa. Essas prioridades certificam-se de que as regras JIT tenham precedência sobre as regras posteriores aplicadas no nível da sub-rede.

As regras JIT são exibidas na parte superior da lista, porque elas têm a prioridade mais baixa. O tráfego é permitido ao endereço IP da VM, mas apenas de seu próprio endereço IP. Isso é o que o JIT configurou. O que pode parecer estranho aqui é que uma regra default-allow-ssh ainda existe e permite todo o tráfego. Pense novamente no capítulo 5, quando discutimos NSGs. Consegue ver o que está acontecendo aqui?

O JIT aplica-se somente à VM. Na regra JIT, Destination mostra o endereço IP da VM. No exemplo mostrado na Figura 16.7, isso é 10.0.0.4. O tráfego é permitido. Porém, a regra de NSG real é aplicada à sub-rede inteira. A regra default-allow-ssh aplica-se no nível da sub-rede e permite o tráfego de qualquer origem e de qualquer destino.

As regras de NSG são processadas em ordem de prioridade, de baixa para alta. Como discutido no capítulo 5, uma ação de negação sempre terá efeito, independentemente de qualquer regra adicional. Mesmo que você tenha alterado essa regra default-allow-ssh para negar o tráfego, a regra JIT ainda permitirá o acesso à VM específica e ao endereço IP de origem definido.

Tome cuidado com essa camada de regras de NSG. Preferencialmente, você removeria a regra default-allow-ssh e, em seguida, permitiria o acesso somente quando necessário com JIT. Nessa abordagem, o SSH é negado pela regra final DenyAllInbound. Quando você precisa se conectar a uma VM, use o JIT para solicitar acesso, que cria automaticamente uma regra para permitir que o SSH com escopo para seu endereço IP por um período definido.

A regra de NSG é excluída automaticamente após o término do período especificado. Por padrão, regras JIT são aplicadas por três horas. Após esse tempo, a VM retornará para um estado mais seguro e você precisará novamente solicitar o acesso à VM.

Este processo JIT controla quem pode solicitar e ter concedido o acesso à VM. Porém, só porque uma pessoa pode com êxito solicitar acesso a uma VM com êxito não significa que ela tenha permissões para fazer login nesta VM. Tudo o que acontece no Azure é que as regras NSG definidas são atualizadas. O Security Center e o JIT não podem adicionar, remover ou atualizar credenciais de acesso na VM.

Todas as solicitações JIT também são registradas. Na Central de Segurança, selecione a opção Acesso just in time da VM e escolha sua regra. À direita, selecione a opção de menu ... e escolha Log de atividades. Esse log de atividades ajuda a auditar quem solicitou acesso a uma VM no caso de um problema.

O acesso JIT da VM é uma maneira como a Central de Segurança e o Azure ajudam a manter suas VMs seguras. O controle do acesso às VMs é uma grande parte da segurança. Mas e quanto às aplicações, bibliotecas e serviços executados nas VMs? É aí que você precisa garantir que todas as atualizações de segurança mais recentes sejam aplicadas às suas VMs de maneira oportuna.

16.3 Gerenciamento de Atualizações do Azure

Uma área na qual a Central de Segurança do Azure pode relatar é o status de as atualizações do sistema operacional exigidas pela VM. Em pizzaria, você deve tentar instalar os patches mais recentes de segurança e de aplicação. Você não quer executar qualquer sistema que tenha uma vulnerabilidade conhecida ou área de ataque;

portanto, uma maneira de automatizar as atualizações desses sistemas e controlar a conformidade melhora sua segurança. Quando você trabalha com aplicações que envolvem dados de cliente e informações de pagamento, não execute sistemas sem os patches mais recentes instalados.E lembre-se de planejar um ambiente de teste que permite aplicar, com segurança, patches de segurança e validar que eles não causam problemas antes de aplicá-los aos sistemas de produção.

Um recurso de Gerenciamento de atualizações é incorporado às VMs do Azure e pode examinar, relatar e corrigir as atualizações do sistema operacional. O que é ótimo sobre esta solução é que funciona em Windows e Linux, e mesmo dentro do Linux, por meio de diferentes distribuições, como Ubuntu, Red Hat, and SUSE. A Figura 16.8 mostra como o Gerenciamento de atualizações monitora e pode instalar as atualizações necessárias.

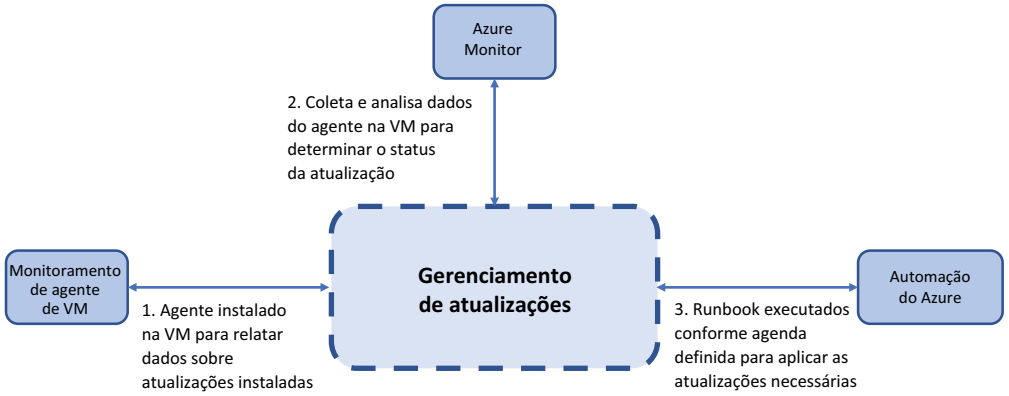


Figura 16.8 O Gerenciamento de atualizações instala um agente de VM que coleta informações sobre as atualizações instaladas em cada VM. Estes dados são analisados pelo Azure Monitor e relatados de volta à plataforma do Azure. A lista de atualizações necessárias pode ser agendada para instalação automática por meio de runbooks da Automação do Azure.

Leva alguns minutos para a VM ser preparada e relatar novamente seu status de atualização. Por isso, vamos configurar sua VM e, em seguida, ver o que acontece nos bastidores.

Experimente agora

Para configurar sua VM para o Gerenciamento de atualizações, conclua as etapas a seguir:

1. Abra o portal do Azure e escolha Grupos de recursos no menu à esquerda.
2. Selecione seu grupo de recursos, como `azuremolchapter16`, e selecione sua VM, como `azuremol`.
3. Em Operações, selecione Gerenciamento de atualizações.
4. Aceite a opção padrão para Localização e a opção para criar um espaço de trabalho do Log Analytics e conta da Automação. Examinaremos esses componentes no restante desta seção.
5. Para ativar o gerenciamento de atualizações para a VM, selecione Habilitar.

Você volta à janela Visão geral do Gerenciamento de atualizações, mas leva alguns minutos para configurar a VM e relatar de volta em seu estado. Continue lendo, e deixe o processo continuar.

Vamos ver um pouco mais o que acontece para fazer esta solução de Gerenciamento de atualizações funcionar.

16.3.1 Serviços de gerenciamento do Azure combinados

Se já trabalhou com alguma tecnologia Microsoft na infraestrutura local, você pode ter se deparado com o pacote do System Center. O System Center consiste em vários componentes, como Configuration Manager, Operations Manager, Orchestrator, e Data Protection Manager. Há alguns outros componentes, mas esses componentes principais fornecem uma forma de se fazer estas coisas:

- Definir as configurações e o estado desejado
- Instalar aplicações e atualizações
- Reportar a integridade e a segurança
- Automatizar implantações de grandes serviços e aplicações
- Fazer backup e replicar dados

À medida em que as empresas migraram para a computação na nuvem nos últimos anos, esses componentes na infraestrutura local mais tradicionais do System Center foram substituídos por serviços do Azure que podem funcionar em um ambiente híbrido. Vimos dois componentes em capítulos anteriores, mesmo que você não tenha percebido isso:

- *Azure Backup* fornece uma maneira de fazer backup de VMs ou arquivos individuais, definir políticas de retenção e restaurar dados.
- *Azure Site Recovery* permite replicar VMs para diferentes regiões geográficas em caso de desastre natural ou interrupção prolongada.

O Azure Backup e o Site Recovery ajudaram você a proteger seus dados no capítulo 13. Agora, você usará dois componentes adicionais com Gerenciamento de atualizações:

- *Os espaços de trabalho do Log Analytics* coleta informações de várias origens ou agentes e permite que você defina políticas e consultas para alertá-lo sobre as condições que podem ocorrer. Essas consultas e alertas podem ajudá-lo a rastrear o status de atualização de uma VM ou notificá-lo sobre problemas de configuração ou segurança.
- *Azure Monitor* detalhe e relata informações com base no processamento realizado em espaços de trabalho do Log Analytics. O Azure Monitor fornece uma maneira centralizada de exibir alertas, consultar dados de log e gerar notificações em todos os seus recursos do Azure.
- *Automação do Azure* permite que você crie runbooks que executam comandos ou scripts inteiros. Runbooks podem ser implantações grandes e complexas e podem chamar vários outros runbooks. Veremos a Automação do Azure em profundidade no capítulo 18.

A integração desses componentes é mostrada na Figura 16.9.

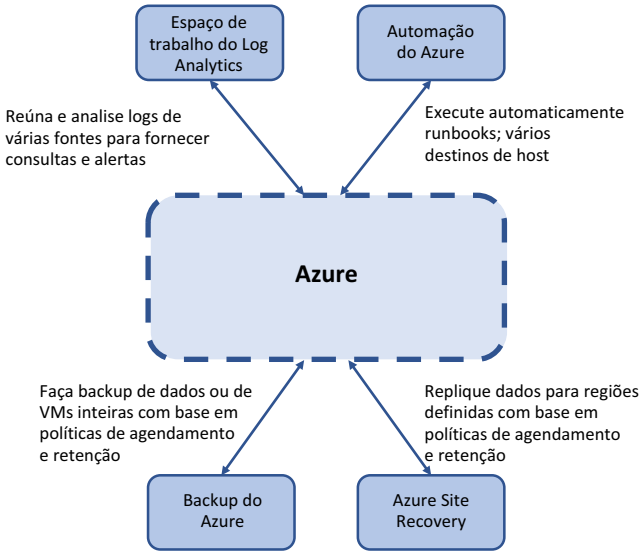


Figura 16.9 Vários serviços do Azure que funcionam juntos para fornecer recursos de gerenciamento e configuração em todo o ambiente da aplicação. Os serviços que usam esses componentes não estão limitados a VMs ou recursos do Azure e podem funcionar em outros provedores de nuvem ou sistemas na infraestrutura local quando configurados apropriadamente.

Os espaços de trabalho do Log Analytics e a Automação do Azure são componentes poderosos e podem facilmente ocupar sozinhos capítulos inteiros de um livro. Com apenas algumas de VMs para gerenciar, você pode achar que é fácil ignorar a necessidade de um repositório de logs centralizado para consultar e alertar ou ignorar uma maneira de automatizar configurações e implantações em VMs. Se você ainda não fez uma lista de componentes do Azure para acompanhar quando terminar com este livro, abra uma lista e adicione esses dois componentes a essa lista.

Uma coisa a se entender é que, no Azure, vários serviços e componentes podem interagir e complementar uns aos outros. Da mesma forma como as VMs do Azure e as redes virtuais do Azure são serviços individuais, ambos os serviços também se complementam ou até dependem um do outro. O Azure Backup e a extensão de diagnóstico do Azure são ótimos componentes individuais, mas eles realmente se destacam se os espaços de trabalho do Log Analytics e o Azure Monitor forem usados para monitorar seu status e agrupar os eventos ou avisos gerados. Espero que você tenha começado a identificar alguns desses componentes relacionados e a ver como os serviços do Azure geralmente funcionam entre si. Agora que estamos nesses capítulos finais e observando as opções de segurança e monitoramento, o objetivo é garantir que as aplicações executadas no Azure sejam íntegras e estáveis.

Esta coisinha chamada "Identity" (ou, identidade, em inglês)

Considerando serviços que se complementam, uma grande (e eu quero dizer *grande mesmo*) parte do Azure que nós tocamos apenas de leve é o Azure Active Directory (Azure AD). O Identity é central para tudo no Azure, e o Azure AD fornece alguns dos recursos de segurança examinados no capítulo 6 com o modelo de implantação do Azure Resource Manager. A capacidade de usar RBACs para limitar quais ações podem ser executadas em um recurso por determinados usuários ou grupos está vinculada a uma solução centralizada de identidade. Mesmo a capacidade de entrar no portal do Azure ou na CLI do Azure é acionada pelo Azure AD.

Este livro não abrange o Azure AD, pois o escopo do que ele fornece é amplo e bastante diferente dos serviços de IaaS e PaaS do Azure, como VMs, conjuntos de escala e aplicativos Web. Pode haver alguma sobreposição no público-alvo de tópicos, mas a maioria dos desenvolvedores teria uma meta diferente para o que eles queriam aprender sobre o Azure AD em comparação com um gerente de aplicação ou um profissional de TI que implanta a infraestrutura.

Dependendo da sua conta do Azure, você também pode ficar limitado no que você pode fazer com o Azure AD. Quando você se inscreve para uma conta de trial gratuito do Azure, uma instância padrão do Azure AD é criada para você. Você é a conta principal nesse diretório, e você tem direitos completos de administrador. Se você fizer login no Azure com uma conta de sua empresa ou instituição educacional, há uma boa chance de que você tenha pouco ou nenhum direito administrativo. Assim, mesmo se pudéssemos concordar em focar alguns tópicos, você pode não ser capaz de realizar diretamente nenhum dos exercícios. E eu realmente não recomendo que você se aprofunde em um ambiente do Azure AD sozinho para aprender como as coisas funcionam.

Porém, o Azure AD é outro desses serviços centrais no Azure que une muitos outros serviços e componentes. A computação na nuvem não torna as coisas magicamente mais fáceis nem eliminam os silos operacionais. Você ainda precisa das habilidades para trabalhar com diferentes equipes e stakeholders. Espero que, ao longo desses capítulos, você tenha adquirido as habilidades básicas para esses serviços do Azure, que vão ajudá-lo a entender como criar aplicações grandes e com redundância e conversar em um melhor nível e com mais conscientização sobre o que outras equipes podem ter que lidar.

16.3.2 Revisar e aplicar atualizações

Pode levar algum tempo para o agente VM executar a primeira verificação e relatar novamente o estado das atualizações aplicadas. A lista de componentes instalados também deve ser cruzada com a lista de atualizações disponíveis para um determinado sistema operacional e versão. Se sua VM terminou e relatou novamente em seu estado, continue lendo e verifique novamente em alguns minutos. Quando estiver pronto, a visão geral é como mostrado na Figura 16.10. Seja paciente. Pode levar de 10 a 15 minutos para que a preparação do agente seja exibida como pronta e permita que você programe as atualizações para instalação.

Uma lista de atualizações necessárias é ótima, mas e quanto a uma maneira de instalá-las? É aí onde entra a Automação do Azure. Quando você habilitou o Gerenciamento de atualizações, vários runbooks da Automação do Azure foram criados, que manipulam automaticamente o processo para aplicar as atualizações necessárias.

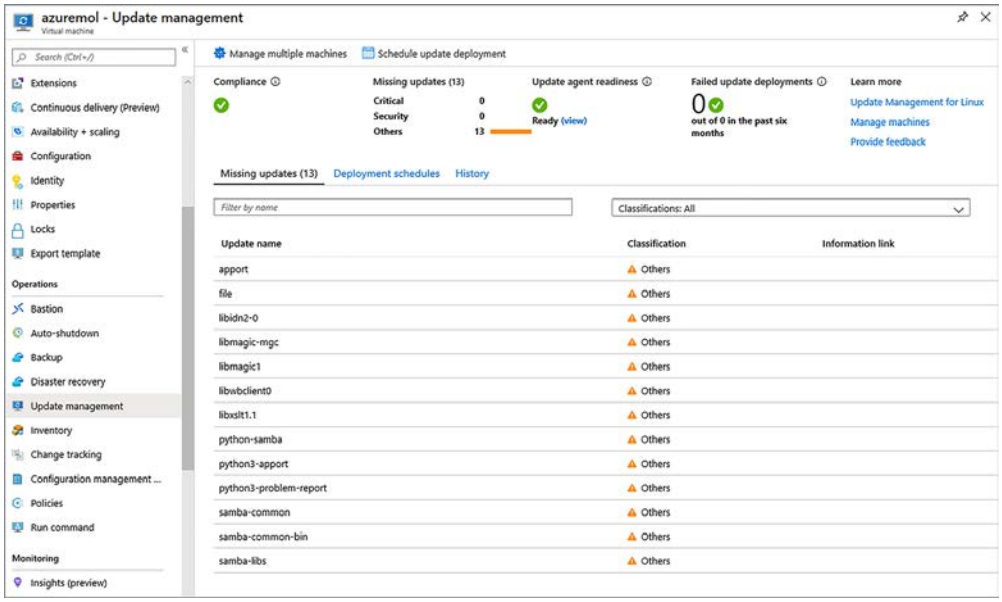


Figura 16.10 Quando agente de VM examinar a conformidade, uma lista de atualizações disponíveis será fornecida. Dependendo do sistema operacional e da versão, o Gerenciamento de atualizações pode ser capaz de trabalhar com o espaço de trabalho Log Analytics e o Azure Monitor para classificar as atualizações com base na severidade ou fornecer links para as páginas de correção de atualização relevantes.

Experimente agora

Se você tiver sorte (ou azar), sua VM pode relatar que nenhuma atualização é necessária. As imagens de VM são atualizadas com frequência no Azure, e se você implantar uma VM logo depois que a última imagem for criada, todas as atualizações necessárias já estarão instaladas. Em caso afirmativo, leia estas etapas para que você entenda o que é necessário quando suas VMs precisam ser atualizadas.

Para aplicar as atualizações necessárias para sua VM, conclua as etapas a seguir:

1. Na seção Gerenciamento de atualizações da sua VM, selecione Agendar implantação de atualização.
2. Insira um nome para a implantação de atualização, como `azuremolupdates` e examine as Classificações de atualizações. Você pode controlar quais conjuntos de atualizações são aplicados. Por enquanto, deixe todas as opções padrão definidas.
3. Atualizações a excluir permitem especificar atualizações que você não deseja instalar. Se souber que sua aplicação exige uma versão específica de um pacote ou biblioteca, você pode certificar-se de que um pacote atualizado que possa criar problemas não esteja instalado. Revise as opções disponíveis, mas não há nada para alterar neste exercício.

4. Selecione Configurações de agenda e, em seguida, escolha uma hora para que as atualizações sejam aplicadas nas opções de calendário e hora. A hora de início deve ser pelo menos cinco minutos antes da hora atual, para oferecer à plataforma do Azure alguns instantes para processar e agendar seu runbook na Automação do Azure.
5. Quando estiver pronto, selecione OK.
6. Se determinadas aplicações e serviços precisarem pausar ou encerrar antes que as atualizações sejam aplicadas e forem iniciados novamente quando as atualizações forem concluídas, escolha Pré-scripts + pós-scripts. Tarefas de automação separadas poderão ser configuradas para executar ações nas VMs antes e depois que as atualizações forem aplicadas.
7. Janela de manutenção (minutos) define quanto tempo, em minutos, o processo de atualização pode ser executado antes que a VM precise estar de volta em operação. Essa janela de manutenção evita processos de atualização de longa execução que podem fazer com que uma VM não esteja disponível por horas. Você pode querer tornar a janela de manutenção mais curta ou mais longa, dependendo de acordos de nível de serviço para as aplicações executadas nessas VMS ou o número e o tamanho das atualizações necessárias. Aceite o valor padrão e selecione Criar.
8. De volta à janela Gerenciamento de atualizações, selecione Agendas de implantação. As atualizações são listadas conforme agendadas para instalação na data e hora que você seleciona, como mostrado na Figura 10.86.11.

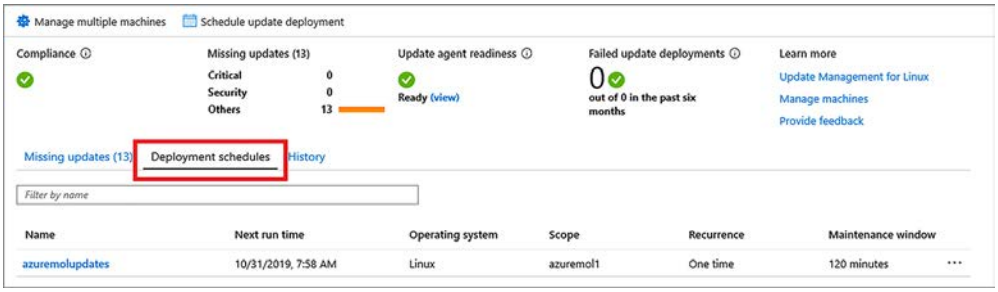


Figura 16.11 A lista de tarefas de implantação agendadas é mostrada. Se desejar, você pode excluir uma determinada tarefa. Caso contrário, as atualizações são aplicadas automaticamente na hora definida.

9. Na parte superior da janela Gerenciamento de atualizações, selecione Gerenciar várias máquinas. A janela alterna para a conta da Automação do Azure que foi criada quando Gerenciamento de atualizações foi habilitado para a VM. Não se preocupe muito agora com o que os runbooks fazem. Não há nada para você personalizar, e examinaremos a Automação do Azure no capítulo 18.

   Observe que você pode escolher Adicionar VM do Azure ou Adicionar máquinas que não são do Azure, como mostrado na Figura 16.12. Essa capacidade destaca uma abordagem única para gerenciar atualizações em todo o ambiente da aplicação, não apenas para VMs do Azure.

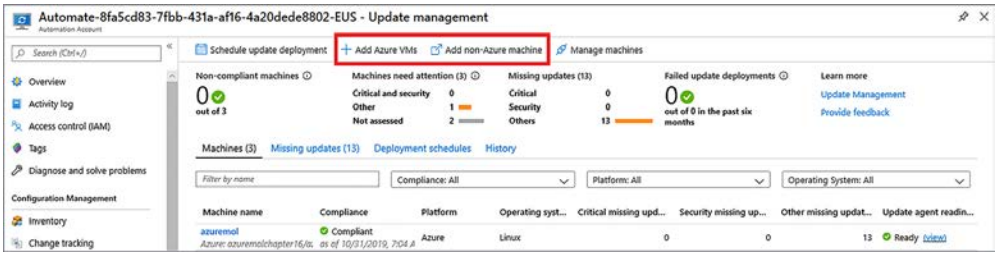


Figura 16.12 **Na conta da Automação do Azure, você pode gerenciar vários computadores e exibir o estado ou aplicar atualizações. As VMs do Azure e os computadores que não são do Azure podem ser monitorados e controlados pela mesma conta da Automação do Azure. Nos bastidores, o Azure pode ser integrado a outros provedores para instalar agentes em computadores em um ambiente híbrido. Essa integração permite que um único painel e plataforma de gerenciamento cuidem de suas necessidades de atualização.**

10 Volte para a janela Gerenciamento de atualizações para sua VM e selecione a guia Histórico. Quando a implantação da atualização é iniciada, seu estado é exibido. Lembre-se de que você programou o trabalho para ser executado em alguns minutos no futuro, por isso, ele não é exibido imediatamente.

11 Selecione a agenda para ver o status e a saída, como mostrado na Figura 16.13.

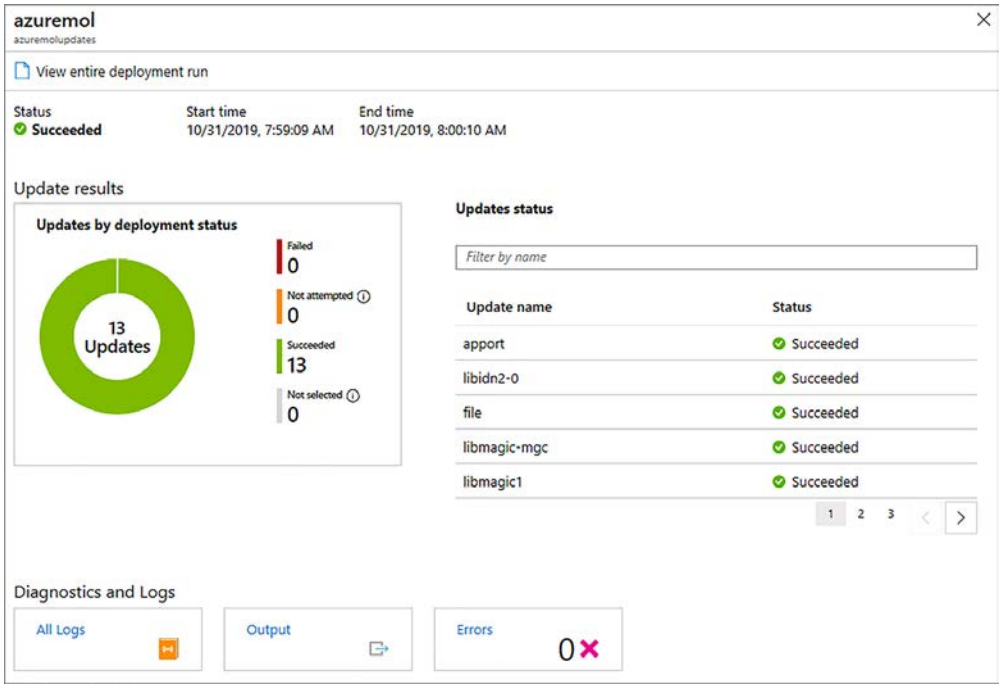


Figura 16.13 **Você pode monitorar o estado da execução de trabalhos da Automação do Azure no portal. Para revisar ou solucionar problemas de tarefas, você pode clicar em um trabalho para exibir qualquer saída e logs gerados.**

12 Quando a implantação da atualização for concluída, navegue de volta para o grupo de recursos, selecione sua VM e escolha Gerenciamento de atualizações. Pode levar alguns minutos para o agente ser atualizado e relatar novamente por meio do espaço de trabalho do Log Analytics que as atualizações foram aplicadas. Em seguida, ele deve mostrar no painel que a VM está atualizada, e nenhuma atualização adicional é necessária.

Este capítulo foi um mergulho na Central de Segurança e componentes associados, como acesso JIT da VM e Gerenciamento de atualizações. O objetivo é que você comece a pensar não apenas em implantar e executar uma VM ou um aplicativo Web mas, em vez disso, planejar o gerenciamento mais amplo de aplicações. A computação na nuvem não altera a necessidade de políticas de segurança. Sem dúvida, há uma necessidade maior de recursos serem protegidos. Deixe que os recursos do Azure, como a Central de Segurança, orientem sobre o que precisa ser feito e use as ferramentas internas, como o Gerenciamento de atualizações e a Automação do Azure, para manter as coisas seguras o tempo todo.

16.4 Laboratório: Habilitar o JIT para uma VM do Windows

Este capítulo cobriu alguns dos componentes que podem ter levado algum tempo para serem habilitados e relatar seu estado esperado. Este laboratório é opcional. Ele serve mais para mostrar que não há nada específico de um sistema operacional em qualquer um desses recursos. Se você não tiver tempo ou se você sentir que entende como aplicar esses recursos a uma VM do Windows, sinta-se à vontade para pular este laboratório. Caso contrário, tente as tarefas a seguir para obter alguma prática adicional com a Central de Segurança e o Gerenciamento de atualizações. A prática leva à perfeição, certo?

1 Crie uma VM do Windows Server de sua escolha no mesmo grupo de recursos usado para os exercícios anteriores, como `azuremolchapter16`.
2 Veja as regras de NSG para a VM/sub-rede e exclua todas as regras padrão que permitem RDP na porta TCP 3389.
3 Use seu cliente local de Remote Desktop Connection para verificar se as conexões RDP estão bloqueadas.
4 Solicite acesso JIT, revise as regras de NSG novamente e confirme que agora você pode fazer RDP para sua VM.
5 Habilite o Gerenciamento de atualizações na sua VM do Windows. Desta vez, você deve ser capaz de usar o espaço de trabalho do Log Analytics existente e as contas da Automação do Azure.
6 Deixe que o agente de monitoramento relate as atualizações necessárias e, em seguida, programe as atualizações a serem aplicadas pela Automação do Azure.

Parte 4

As coisas legais

Agora, as coisas realmente legais. Nestes últimos capítulos, você aprenderá sobre algumas das tecnologias futuras que pode usar no Azure, como inteligência artificial e aprendizado de máquina, contêineres, Kubernetes e a Internet das Coisas. Você pode não estar usando esses serviços agora, mas, com as tendências atuais na computação, provavelmente usará em breve. Esses serviços são algumas das tecnologias mais empolgantes para trabalhar. Embora o livro aborde muito rapidamente esses tópicos durante seu intervalo para o almoço, essa parte é uma ótima maneira de concluir as coisas e mostrar as possibilidades do que você pode criar no Azure.

17 Aprendizado de máquina e inteligência artificial

Esperamos que não terminemos em um mundo em que filmes como *O Exterminador do Futuro* e *Matrix* se tornem realidade. Nesses filmes, a ascensão da inteligência artificial (AI) quase provoca o extermínio da humanidade na medida em que as máquinas lutam para assumir o controle. Uma causa de preocupação na computação agora é como o desenvolvimento da IA é feito na maior parte por grandes corporações privadas, com pouco ou nenhuma regulamentação e supervisão centralizada. Isso não é de todo para dizer que a AI é uma coisa ruim! Os assistentes digitais em smartphones podem ajudar com muitas tarefas do dia a dia. O aprendizado de máquina (ML) em aplicativos de navegação pode monitorar a movimentação diária do usuário para sugerir rotas alternativas com base na estrada ou condições meteorológicas. Os controles de aquecimento residencial podem ser ajustados automaticamente com base na temperatura externa, hora do dia e época do ano (como verão ou inverno).

Ao iniciar esta parte final do livro, você aprenderá sobre os serviços do Azure para aprendizado de máquina e inteligência artificial. Em um capítulo. No seu intervalo para o almoço. Vamos definir algumas expectativas realistas: você não vai se tornar um especialista em ML ou IA nos próximos 45 minutos. Se você comer seu sanduíche rapidamente, poderá aprender o suficiente sobre os muitos serviços que o Azure oferece para entender como integrar alguns desses serviços de ML e IA em suas aplicações. Muitos dos serviços de ML e IA do Azure esperam pelo menos alguma experiência prévia em algoritmos de dados, linguagens de programação, processamento em lote ou compreensão de linguagem. Portanto, não espere se tornar um especialista na próxima hora.

Neste capítulo, vamos nos aprofundar em alguns dos serviços cognitivos do Azure que fornecem recursos de ML e IA. Você aprenderá como usar esses serviços para executar o aprendizado de máquina básico em modelos de dados. Em seguida, você usará um pouco do serviço de aplicativos Web do Azure e o Microsoft Bot Framework para aplicar alguns dos serviços de IA que podem executar um bot de pizzaria para que os clientes façam pedidos de pizza.

17.1 Visão geral e relação entre IA e ML

Segure firme, porque estamos prestes a ir de 0 a 900 km/h em apenas algumas páginas. IA e ML geralmente se sobrepõem à medida que você cria aplicações no Azure. Vamos explorar o que cada um é e depois ver sobre como eles funcionam juntos.

17.1.1 Inteligência artificial

IA permite que os computadores realizem tarefas com alguma flexibilidade e conscientização e ajustem suas decisões com base em fatores externos ou sem a necessidade de interação humana. Geralmente, o objetivo não é criar um sistema completamente autônomo que pode evoluir e desenvolver pensamentos por conta própria, mas sim para usar um conjunto de modelos de dados e algoritmos para ajudar a orientar o processo de tomada de decisão.

A IA comum em computadores e smartphones pessoais inclui Siri, Cortana e o assistente do Google. Como mostrado na Figura 17.1, esses recursos de IA permitem que você se comunique, muitas vezes por meio de comandos de voz, para perguntar o caminho, definir lembretes, pesquisar na Web e muito mais.

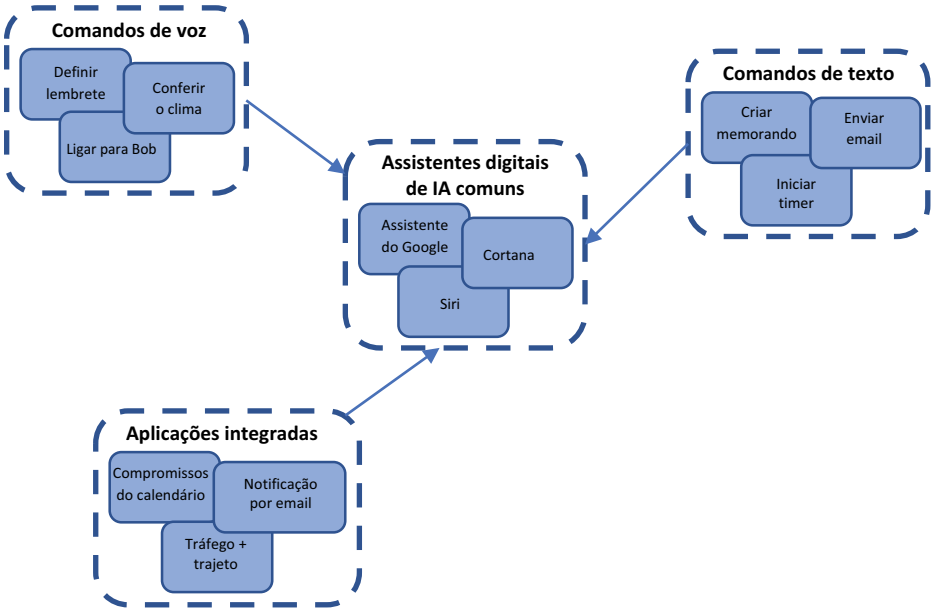


Figura 17.1 Um uso comum de IA no dia a dia são os assistentes digitais, como Cortana, Siri e o assistente do Google. Você pode usar comandos de voz ou texto para interagir com eles, e eles podem monitorar seu calendário diário e comutar condições para avisá-lo sobre problemas de trânsito.

Assistentes digitais como esses normalmente não envolvem uma grande quantidade do que você pode considerar uma *inteligência*. Eles escutam e respondem à entrada que você fornece. Mas, essas entradas podem variar e não podem nem sempre ser comandos específicos. Pense em como um assistente digital permite que você defina um lembrete. Você pode usar uma das seguintes frases:

- "Lembre-me às 5h de comprar leite."
- "Diga-me para comprar leite no caminho de casa."
- "Preciso de leite quando estiver no mercado."

Se você desenvolveu uma aplicação tradicional, você precisaria escrever um código que poderia lidar com todas as possíveis variações de como um usuário pode fornecer instruções. Você pode criar expressões regulares para ajudar a capturar algumas das variações, mas o que acontece quando o usuário vem com uma frase que você não programou? E se eles interagem via texto e houver um erro de digitação em seu pedido que você não antecipou? Estes tipos de interação são perfeitos para a AI. Como mostrado na Figura 17.2, a aplicação é programada para várias frases comuns e, em seguida, é capaz de fazer dar uma opinião abalizada com base no que ele "pensa" que o usuário está pedindo.

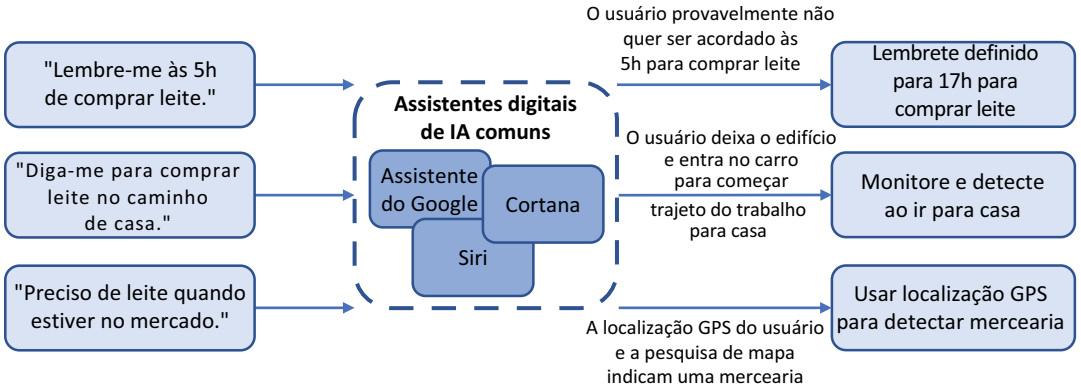


Figura 17.2 A IA pode pegar a entrada do usuário e tomar as decisões que melhor se adequam à ação antecipada. A AI não é pré-programada com todas essas possíveis respostas e árvores de decisão. Em vez disso, ela usa modelos de dados e algoritmos para aplicar o contexto para a entrada do usuário e interpretar o significado e o resultado apropriado.

Não é uma verdadeira inteligência (ainda), mesmo em formas complexas de IA. Em vez disso, é uma opinião abalizada com base em um modelo de dados com o qual a IA foi treinada. Este modelo de dados pode incluir muitas variações e frases e pode ser capaz de aprender novos significados ao longo do tempo. Como ela aprende, e de onde vêm esses modelos de dados? É aí que a ML torna-se importante.

17.1.2 Aprendizado de máquina

Uma expressão da moda na computação ao longo dos últimos anos é o *big data*. O conceito é que os sistemas de computador, principalmente na nuvem, são um ótimo recurso para processar grandes quantidades de dados. *Realmente,* grandes quantidades de dados. Esses trabalhos de processamento podem ser executados por alguns minutos ou horas, dependendo da quantidade dos dados e dos cálculos necessários, e permitem que você prepare e analise grandes volumes de dados para determinar padrões e correlações específicos. Esses aprendizados formam modelos de dados que outras aplicações ou

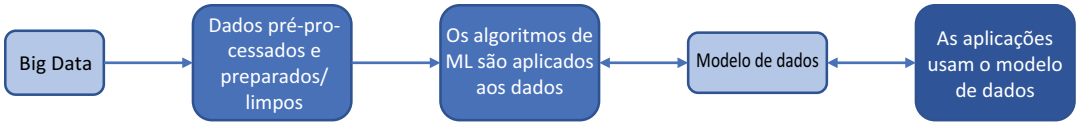


Figura 17.3 Grandes quantidades de dados brutos são processadas e deixadas prontas para uso. Diferentes técnicas de preparo e sanitização de dados podem ser aplicadas, dependendo das entradas brutas. Os algoritmos de ML são aplicados aos dados preparados para criar um modelo de dados apropriado que reflita a melhor correlação entre todos os pontos de dados. Diferentes modelos de dados podem ser produzidos e refinados ao longo do tempo. As aplicações podem usar os modelos de dados em suas próprias entradas de dados para ajudar a orientar sua tomada de decisão e compreender os padrões.

a IA pode usar para tomar decisões. Como mostrado na Figura 17.3, o ML envolve algumas etapas e inclui entradas e saídas.

Veja como a forma mais básica de ML funciona:

1. Para iniciar o processo, grandes quantidades de dados brutos são fornecidas como entrada.
2. Estes dados são processados e preparados em um formato utilizável para focar os pontos de dados específicos necessários para análise.
3. Os algoritmos de ML são aplicados aos dados. Este é o lugar onde o verdadeiro cálculo numérico acontece. Os algoritmos são criados para detectar e computar semelhanças ou diferenças em toda a grande quantidade de pontos de dados.
4. Com base na análise dos algoritmos, um modelo de dados é produzido, definindo padrões nos dados. Estes modelos de dados podem ser refinados ao longo do tempo se partes do modelo provarem estar incorretas ou incompletas quando dados adicionais do mundo real forem aplicados.
5. As aplicações usam os modelos de dados para processar seus próprios conjuntos de dados. Esses conjuntos de dados são normalmente muito menores do que os dados brutos fornecidos para os algoritmos de ML. Em seguida, se o modelo de dados é válido, mesmo com uma pequena entrada de dados da aplicação, o resultado ou a correlação correto pode ser determinado.

O ML geralmente envolve algoritmos complexos que são criados para processar todos os pontos de dados fornecidos. O Hadoop e o Apache Spark são duas pilhas de aplicações geralmente usadas para processar big data. O Azure HDInsight é um serviço gerenciado que permite analisar os grandes conjuntos de dados processados por essas pilhas de aplicações. Para se aprofundar um pouco mais na análise e nos algoritmos, os cientistas de dados geralmente usam a linguagem de programação R para desenvolver os modelos necessários. Não se preocupe muito com o que é Hadoop ou R. O ponto mais importante é que o Azure pode executar as ferramentas de ML comuns que são amplamente aceitas na indústria.

17.1.3 Colocar IA e ML juntos

Uma aplicação comum em um smartphone é a aplicação de navegação, como mostrado na Figura 17.4. Seu provedor, como o Google, pode rastrear a rota que você usa para ir trabalhar todo dia, a que horas você costuma sair de casa e quanto tempo leva para chegar lá.

Este exemplo do Google Maps mostra a IA e o ML funcionando juntos. A AI é aplicada para saber quando gerar uma notificação com base nos dados recebidos após o processamento do

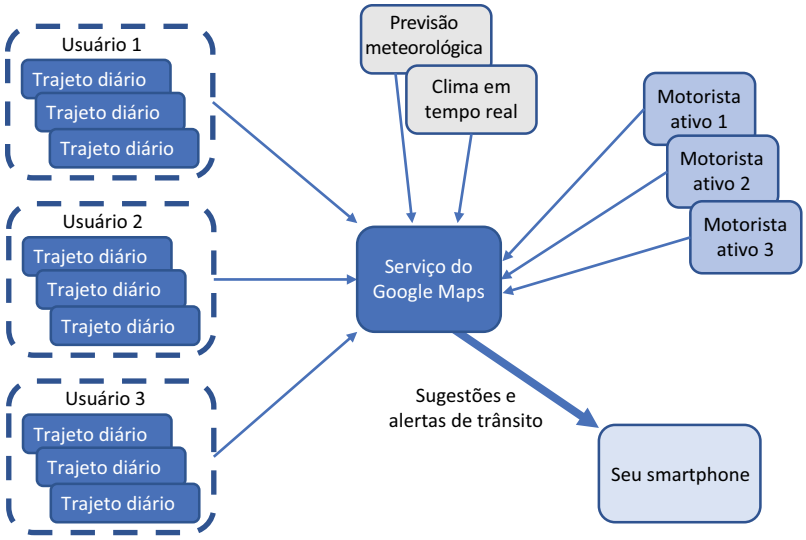


Figura 17.4 Todos os dias, o serviço do Google Maps recebe vários pontos de dados de usuários, que registram detalhes de seu trajeto. Estes dados podem ser preparados e processados, juntamente com a previsão do tempo e o clima em tempo real durante esses percursos. Os algoritmos de ML podem ser aplicados a esses grandes conjuntos de dados e um modelo de dados é produzido. À medida em que uma amostra menor de motoristas ativos alimentam as suas condições de seus percursos ou dados meteorológicos no serviço do Google Maps, o modelo de dados pode ser aplicado para prever o trajeto e gerar um alerta de tráfego para seus smartphones, que sugere um caminho alternativo para casa.

modelo de dados do ML. Outro exemplo de IA e ML funcionando juntos é a ideia de definir um lembrete para comprar leite. Se a IA foi treinada com modelos de dados de ML, o assistente saberá que você provavelmente compra leite no supermercado, e ele não vai lembrá-lo se você for para a loja de material de construção. O modelo de dados de ML também seria capaz de ajudar a IA entender que há uma probabilidade maior de que você quer ser lembrado de algo às 17:00, e não às 5:00, e por isso não deve acordá-lo às 5:00 para lembrá-lo de comprar leite. Se o seu smartphone rastreia você entrando no seu carro às 17:00 e começar a se afastar do trabalho, o ML vai gerar um modelo de dados que prevê que você está dirigindo para casa, então esse agora é um bom momento para a IA lembrá-lo da compra de leite.

Estes exemplos básicos, porém, poderosos, mostram como o ML é usado para melhorar a IA. Você treina a IA fornecendo um conjunto de pontos de dados que são processados por ML para melhorar a precisão ou a tomada de decisão.

17.1.4 Ferramentas de ML do Azure para cientistas de dados

Eu quero rapidamente abordar algumas maneiras como alguns cálculos numéricos do mundo real e o ML podem trabalhar juntos. Para tornar este capítulo acessível a todos, os exercícios usam o Microsoft Bot Framework para IA e ML com Language Understanding Intelligent Service (LUIS). Para colocar as mãos na massa com ML, precisamos focar um pouco mais no processamento de dados e algoritmos.

No Azure, alguns componentes interessantes ajudam você a se aprofundar em dados em uma grande escala. Primeiro, há o Azure Machine Learning, um serviço baseado na Web que permite criar visualmente experimentos por se adicionar conjuntos de dados e modelos de análise. Estes experimentos podem usar fontes de dados, como Hadoop e SQL, e o suporte de programação adicional é fornecido para linguagens como R e Python. Você pode arrastar e soltar fontes de dados, técnicas de preparação de dados e algoritmos de ML. Você pode ajustar esses algoritmos e, em seguida, revisar e ajustar os modelos de dados produzidos.

O Azure Machine Learning fornece uma barreira baixa para a entrada nos recursos de computação de grande escala disponíveis no Azure. Uma vantagem principal de executar o cálculo numérico de dados de ML no Azure é que você pode acessar uma grande quantidade de capacidade de computação e usá-la apenas pelo tempo necessário para fazer seus cálculos. Em ambientes tradicionais, estes recursos computacionais dispendiosos ficariam ociosos por longos períodos entre os trabalhos de processamento de dados.

Um outro recurso interessante que ajuda você a executar cálculos numéricos mais pesados de ML no Azure são as máquinas virtuais de ciência de dados (DSVMs). Estas VMs estão disponíveis para Linux e Windows. Eles vêm com muitas aplicações comuns pré-instaladas, incluindo Jupyter Notebooks, Anaconda Python e R Server ou SQL Server (Figura 17.5_.

Figura 17.5 DSVMs estão disponíveis para Windows e Linux. Esta DSVM do Window Server 2016 vem com várias aplicações de ciência de dados pré-instaladas, como R Server e Jupyter Notebooks. As DSVMs permitem que você rapidamente esteja pronto para o processamento de big data e criação de algoritmos de ML.

Não há necessidade de instalar todas as ferramentas e dependências em seu computador local. Você pode criar uma DSVM e recursos de CPU e memória que você precisa para processar rapidamente seus dados e, em seguida, excluir a VM quando seu trabalho de processamento for concluído e você obtiver os modelos de dados necessários.

17.2 Serviços Cognitivos do Azure

Muito bem, e quanto a serviços de IA para deixar suas aplicações mais inteligentes? No Azure, um conjunto de serviços relacionados compõem o pacote de Serviços Cognitivos. Os serviços abrangem algumas áreas comuns de IA que permitem que você integre rapidamente esses recursos inteligentes em suas aplicações, divididas nas seguintes áreas gerais:

- Visão
- Fala
- Linguagem
- Decisão
- Pesquisa

Mais de duas dúzias de serviços fazem parte da família dos Serviços Cognitivos. Alguns desses serviços são

- *Visão*, que inclui
  - *Visão computacional* para análise de imagem, legendagem e marcação.
  - *Rosto* para análise e detecção de rostos em imagens.
- *Fala*, que inclui
  - *Serviços de fala* para análise e conversão de fala em texto e vice-versa.
  - *Reconhecimento do locutor* para identificação e verificação do locutor.
- *Linguagem*, que inclui
  - *Reconhecimento de linguagem (LUIS)* para compreensão e processamento da interação com usuários. Vamos explorar o LUIS no laboratório no final deste capítulo.
  - *Texto do tradutor* para análise e correção de erros ortográficos ou execução de traduções.
- *Decisão*, que inclui
  - *Moderador de conteúdo* para análise e moderação de fotos, vídeos e textos.
  - *Personalizador* para análise de padrões e fornecimento de recomendações para clientes.
- *Pesquisa*, que inclui
  - *Pesquisa personalizada do Bing* para implementação de pesquisa em seus dados personalizados e em aplicações.
  - *Sugestão automática do Bing* para fornecimento de sugestões automáticas à medida que os usuários inserem frases e consultas de pesquisa.

Como você pode ver, muitos serviços do Azure combinam recursos de IA e ML. Este capítulo foca na linguagem, especificamente a LUIS. Este serviço é geralmente

usado para criar um bot inteligente que pode ajudar os clientes em seu site. Em seguida, você pode criar uma aplicação que usa serviços de IA no Azure que possa interpretar frases e perguntas, e fornecer a resposta apropriada para guiar um usuário por meio de um processo de pedido ou de uma solicitação de suporte.

17.3 Criar um bot inteligente para ajudar com pedidos de pizza

Um *bot* é uma aplicação que está programada para responder a tarefas e entrada de um usuário. Se isso soa muito como qualquer aplicação normal, bem, é porque é mesmo! A diferença é como a aplicação bot determina a resposta.

Um bot básico e comum frequentemente é uma aplicação que fornece alguma forma de automação. Quando um usuário envia uma mensagem, ele define uma marca em uma mensagem de email ou envia um termo de pesquisa, o bot executa tarefas pré-programadas que executam uma ação específica. Não há IA ou ML real aqui; a aplicação bot está apenas respondendo à entrada do usuário.

Com o enquadramento certo, um bot pode ser estendido e receber um pouco mais de liberdade e inteligência. No início de nossa visão geral da AI, discuti como um aplicativo típico deve ser pré-programado com todas as entradas de usuário antecipadas e qual seria a saída correspondente. Mas, não há flexibilidade se o usuário fornecer uma frase de entrada diferente ou cometer um erro ortográfico, por exemplo.

A Microsoft produz o Bot Framework, que permite que um bot do Azure integre facilmente os Bot Builder SDKs e conecte-se aos Serviços Cognitivos do Azure. Com experiência mínima codificação, você pode criar bots inteligentes que usem o poder do Azure para oferecer uma ótima experiência ao cliente. Só não tente construir a Skynet, a menos que você *saiba como Exterminador do Futuro* termina!

17.3.1 Criar um aplicativo Web no Azure

Vamos implementar um bot e integrar alguns serviços de AI e ML. O bot é executado em um aplicativo Web do Azure e usa o Microsoft Bot Framework para conectar-se à LUIS e permitir que um cliente faça um pedido de pizza. A Figura 17.6 descreve o que esses exercícios criarão e quais serviços são usados.

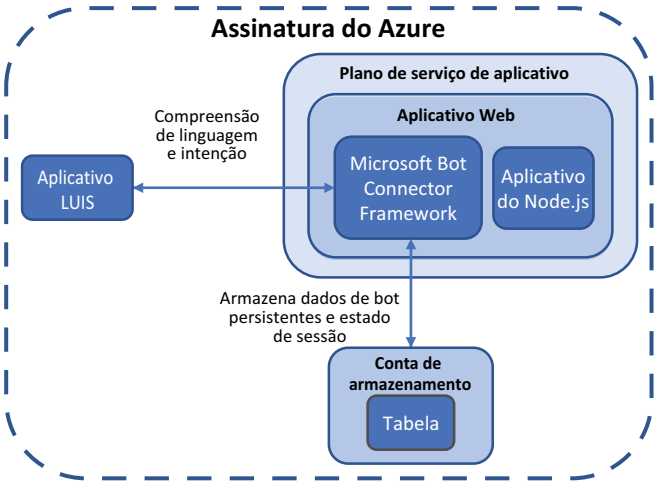


Figure 17.6 Nos próximos exercícios, você criará um bot de aplicativo Web que integra vários serviços de AI e ML do Azure para interagir com um cliente e ajudá-lo a fazer pedido de pizza.

Experimente agora

Para criar um bot do aplicativo Web do Azure, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione Criar um Recurso no canto superior esquerdo.
2. Pesquise e selecione Web App Bot, e então selecione Criar.
3. Digite um nome para o seu bot, como `azuremol`. Depois, crie um novo grupo de recursos e atribua um nome a ele, como `azuremolchapter17`.
4. Selecione a região mais apropriada para você e escolha a camada de preços F0. Seu bot não processará um monte de mensagens, de modo que a camada gratuita (F0) é suficiente.
5. Selecione um modelo de bot e escolha a linguagem SDK do Node.js.
6. Crie um bot básico, pois forneceremos nosso próprio exemplo de código de aplicação em um exercício posterior. Esta etapa cria um aplicativo LUIS que você pode usar para realizar o treinamento de linguagem e o ML.
7. Escolha a região mais apropriada para seu aplicativo LUIS e crie uma nova conta do LUIS.
8. Atribua um nome à conta do LUIS, como `azuremol`. Essa conta do LUIS gerencia o sentimento do usuário para o nosso bot.
9. Escolha App Service Plan e crie um novo plano. Atribua um nome, como `azuremol`, e novamente selecione a região mais apropriada para você.
10. Desative o App Insights, pois seu bot não vai usá-lo. Como nos capítulos anteriores sobre aplicativos Web, para uso de produção convém aproveitar a eficiência do App Insights para ter visibilidade da performance da aplicação, transmitindo dados e análises diretamente do código.
11. Aceite a opção de criar automaticamente o ID e a senha do aplicativo da Microsoft, aceite o acordo e escolha Criar.

Leva alguns minutos para criar o bot do aplicativo Web e os componentes associados. Muita coisa acontece por trás dos bastidores:

- Um plano Azure App Service é criado.
- Um aplicativo Web é implementado, juntamente com um aplicativo Web node.js de exemplo.
- Um aplicativo LUIS é criado e as chaves de conexão são configuradas com seu aplicativo Web.
- Um bot é criado com o Microsoft Bot Connector e as chaves de conexão são configuradas a partir do seu aplicativo Web.

17.3.2 Compreensão de linguagem e intenção com LUIS

Uma das áreas do Azure Cognitive Service que vimos anteriormente foi a linguagem. Isso faz sentido, porque alguma forma de linguagem é muitas vezes usada para interagir com uma AI. Você pode usar o LUIS para processar uma mensagem ou frase do usuário e determinar sua intenção. Essa intenção ajuda seu aplicativo a fornecer uma resposta apropriada. Vamos estender o seu bot com LUIS.

Experimente agora

Para criar um aplicativo LUIS e usar ML para treiná-lo, conclua as etapas a seguir

1. Abra um navegador da Web para www.luis.ai e faça logon com as mesmas credenciais da Microsoft da sua assinatura do Azure.
2. Selecione Meus aplicativos e escolha seu aplicativo, como `azuremol`. Seu nome de aplicativo LUIS provavelmente tem alguns caracteres numéricos adicionais anexados a ele a partir do nome do bot que você especificou no portal do Azure.

   Algumas intenções pré-construídas foram criadas, mas você deseja sobrescrever o aplicativo LUIS com uma amostra mais focada na pizzaria.
3. Faça download do arquivo azuremol.json do GitHub em https://github.com/fouldsy/azure-mol-samples-2nd-ed/blob/master/17/luisapp/azuremol.json para o seu computador local. Para facilitar a vida, selecione o botão Raw no GitHub para ver apenas o conteúdo do arquivo.
4. De volta ao aplicativo LUIS, escolha Gerenciar o aplicativo e, em seguida, selecione Versões.
5. Escolha importar uma versão, navegue e selecione o arquivo azuremol.json que você baixou, digite o nome da versão `1.0` e selecione Concluído.
6. Volte para Build no menu superior para ver as intenções importadas do aplicativo de exemplo. Escolha uma ou duas intenções, tais como `saudação` ou `orderFood`, e dê uma olhada em algumas das frases de exemplo que um cliente poderia usar para se comunicar com o bot.
7. Antes de poder ver o aplicativo em ação, você deve treiná-lo. Selecione Train e aguarde alguns segundos para que o processo seja concluído. A figura 17.7 mostra os processos de ML trabalhar para treinar seu aplicativo LUIS.

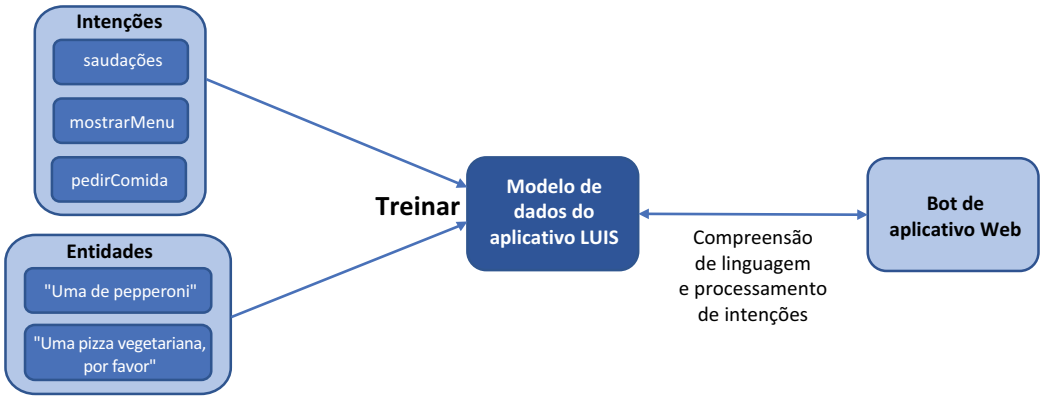


Figura 17.7 Quando você treina o aplicativo LUIS, as intenções e as entidades são inseridas e processadas para criar um modelo de dados. Então, o aplicativo Web usa esse modelo de dados para processar a compreensão e a intenção da linguagem. O número de intenções e entidades de entrada para processamento é pequeno e, portanto, o modelo de dados não é perfeito. No mundo real, muitas mais intenções e entidades seriam fornecidas, e você repetidamente treinaria, testaria e refinaria o modelo de dados para criar conjuntos de dados progressivamente maiores para criar um modelo preciso para processar a linguagem e a intenção.

Em uma aplicação mais complexa e real, pode levar mais tempo para concluir esse processo de treinamento, porque todas as suas intenções e entidades são processadas pelos algoritmos ML para criar o modelo de dados necessário para que sua aplicação responda adequadamente à comunicação do cliente.

8 Com o aplicativo LUIS treinado, selecione Teste e insira algumas saudações, como *o i* e *olá*. Abaixo de cada uma das suas mensagens está a intenção de pontuação superior, junto com a probabilidade de que a mensagem, ou enunciado, que você inseriu coincide com a intenção. Essas saudações básicas devem corresponder à intenção de `saudação`.

9 Tente inserir uma saudação diferente, como *(boa) tarde* ou *(boa) noite*. A saudação de palavra única com base na hora do dia pode retornar uma intenção incorreta de pontuação superior, como `orderStatus`. Experimente algumas outras frases até que algo não se alinhe com a intenção esperada, o que indica que o aplicativo LUIS não compreende totalmente o que você quer dizer. Selecione uma de suas mensagens incorretas, como *manhã*, e escolha Inspect.

10 No menu Inspect, escolha Edit para editar a intenção de pontuação superior incorreta. No menu suspenso, escolha `greetings` ou qualquer que seja a intenção mais apropriada para sua frase incorreta.

11 Você fez uma alteração em seu aplicativo, e então escolha Train para treinar o aplicativo LUIS novamente. A Figura 17.8 mostra como fornecer entradas adicionais para os algoritmos ML para processar o modelo de dados e refinar a compreensão da linguagem e intenção.

12 Na janela de mensagens de teste, digite a mensagem incorreta novamente, tal como *manhã*. Desta vez, a intenção de pontuação superior deve corretamente ser identificada como `saudação`.

13 Para disponibilizar o aplicativo LUIS atualizado para o bot do aplicativo Web, selecione a opção Publish no menu superior. Aceite todos os padrões e escolha Publish to Production Slot. Demora alguns segundos para concluir o processo de publicação.

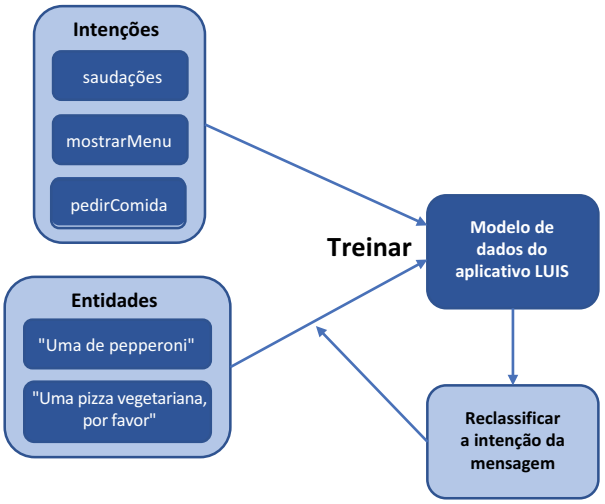


Figura 17.8 À medida que você reclassifica a intenção das mensagens e retreina o aplicativo LUIS, o modelo de dados é refinado à medida que as entradas de dados adicionais são fornecidas aos algoritmos ML. Quando você inserir saudações semelhantes no futuro, o modelo de dados deverá ser melhorado e responderá mais apropriadamente.

Lembre-se de que seu bot é executado em um aplicativo Web e por isso tem slots de preparação e produção como você aprendeu no capítulo 3. No mundo real, você deve publicar em um slot de preparação, verificar se tudo funciona conforme o esperado e, em seguida, publicar no slot de produção. Os mesmos recursos PaaS que permitiram que você testasse e movimentasse o código Web entre os ciclos de vida de desenvolvimento e produção também beneficiam o ciclo de vida do seu bot de aplicativo Web processado pelo LUIS.

Neste exemplo básico, o ML conseguiu pegar a sua entrada de dados *(bom) dia* como uma saudação e entender que saudações similares, tais como *(boa) noite*, também são saudações. O ML funciona melhor quando um grande conjunto de dados pode ser inserido no modelo de dados; portanto é importante testar e ajudar a treinar seu aplicativo. A IA, neste caso o aplicativo LUIS, é apenas tão bom quanto o tamanho e a qualidade dos dados fornecidos para os algoritmos de ML.

17.3.3 Construção e execução de um bot de aplicativo Web com LUIS

Você tem agora um bot de aplicativo Web básico no Azure e um aplicativo LUIS que lida com o processamento de linguagem e retorna a intenção do cliente. Para integrar os dois, você precisa modificar o código do seu bot para usar o LUIS. Os SDKs estão disponíveis para as linguagens de programação C# e Node. js. Eu acho que o Node.js torna um pouco mais rápido e fácil entender o que acontece no código, se isso tudo é novo para você. Se você estiver familiarizado com C#, você está convidado a explorar o C# SDK quando tiver terminado este capítulo. Por enquanto, vamos usar um aplicativo Node.js básico do repositório de exemplo do GitHub para ver seu bot em ação com LUIS.

Experimente agora

Para atualizar seu bot de aplicativo Web com seu bot LUIS treinado, conclua as etapas a seguir:

1. No portal do Azure, selecione Resource Groups no menu à esquerda e escolha seu grupo de recursos, como `azuremolchapter17`. Depois, selecione o bot do seu aplicativo Web, como `azuremol`.

   Vamos usar um exemplo de bot do nosso repositório de amostras do GitHub. O exemplo de bot é written in Node. js, mas como ocorre com aplicativos de exemplo anteriores, não se preocupe se não se aplicar a você.

2. Para implantar o exemplo de bot, abra o Cloud Shell. Se necessário, clone o repositório de exemplos do GitHub em seu Cloud Shell da seguinte maneira:

   ```
   git clone https://github.com/foouldsy/azure-mol-samples-2nd-ed.git
   ```

3. Mude para o diretório do capítulo 17:

   ```
   cd azure-mol-samples-2nd-ed/17/webappbot
   ```

4. Inicialize o repositório Git e adicione os arquivos de bot:

   ```
   git init && git add . && git commit -m "Pizza"
   ```

5. Para carregar o exemplo de bot, crie uma conexão com seu aplicativo Web. O comando a seguir obtém o repositório de aplicativos Web e configura seu repositório Git de exemplos locais para se conectar a ele. Nos capítulos anteriores, fiz você procurar esse endereço, mas agora espero que você tenha começado a

explorar o que mais a CLI do Azure pode fazer e percebido que muitas dessas informações podem ser obtidas rapidamente.

```
git remote add webappbot \
  $(az webapp deployment source config-local-git \
  --resource-group azuremolchapter17 \
  --name azuremol \
  --output tsv)
```

6 Envie por push o exemplo de bot Node.js para o aplicativo Web com o seguinte comando:

```
git push webappbot master
```

7 Quando solicitado, insira a senha para o usuário do Git que você criou e usou nos capítulos anteriores (a conta criada no capítulo 3).

Se você não escreveu sua senha do Git em um Post-It

Se você esqueceu a senha, pode redefini-la. Primeiro, obtenha o nome de usuário da sua conta de implantação do Git local:

```
az webapp deployment user show --query publishingUserName
```

Para redefinir a senha, insira o nome da sua conta do comando anterior e responda às instruções para definir uma nova senha. O exemplo a seguir redefine a senha da conta de usuário denominada azuremol:

```
az webapp deployment user set --user-name azuremol
```

Vamos dar uma olhada na figura 17.9 para ver o que você implementou. O aplicativo LUIS agora é treinado com algoritmos ML, e seu modelo de dados está pronto para o aplicativo Node. js para permitir que os clientes interajam e façam pedidos de pizza.

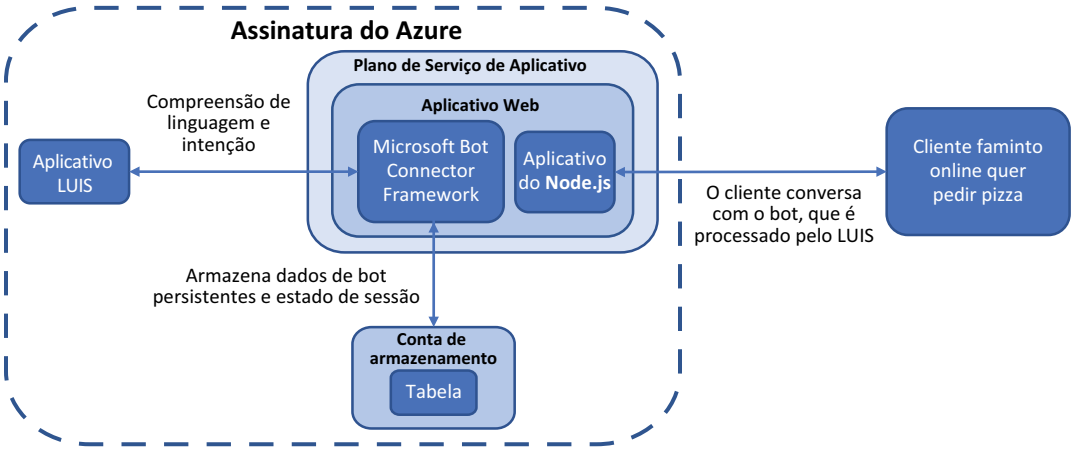


Figura 17.9 Um cliente pode agora acessar o seu bot on-line e pedir para ver o menu ou fazer pedidos de pizza. O LUIS fornece o entendimento da linguagem, que permite que o bot processe os pedidos e envie-os para o Azure Storage para processamento adicional.

De volta ao portal do Azure para seu bot de aplicativo Web, selecione Test in Web Chat. Leva alguns segundos a primeira vez que você se conecta ao bot, mas você deve então ser capaz de interagir, ver a lista de pizzas no menu, e criar um pedido, como exibido na figura 17.10. Experimente você mesmo!

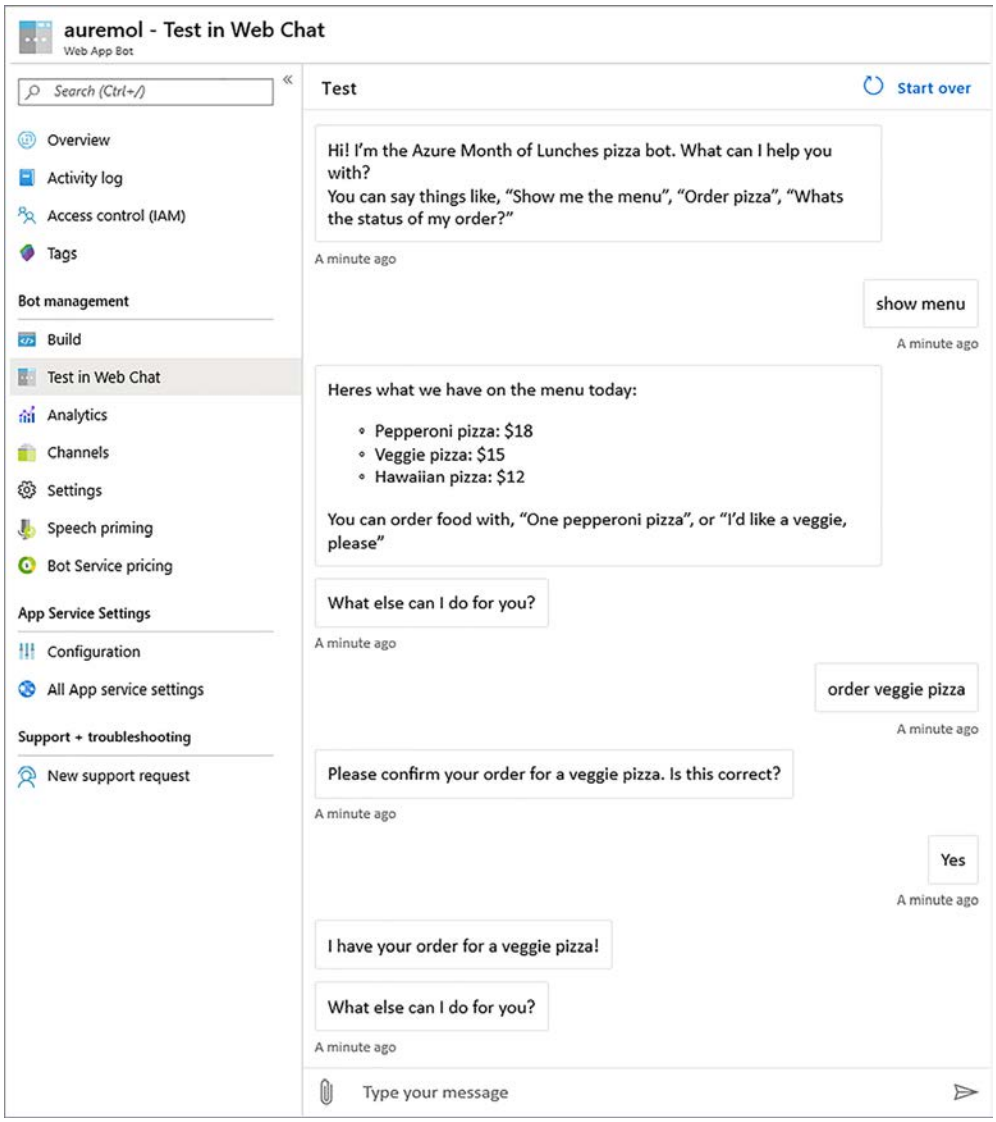


Figura 17.10 Com o seu bot de aplicativo Web em execução, inicie uma conversa e tentar fazer um pedido de pizza. Nesta caixa de diálogo de exemplo, você pode exibir o menu, fazer um pedido de pizza e verificar o status do pedido. O aplicativo é básico e não está realmente criando pedidos ou atualizando o status além da pizza pedida, mas espero que o exercício mostra como você pode rapidamente implantar um bot no Azure.

Espero que esses exercícios básicos tenham lhe dado uma ideia do que o Azure pode oferecer para IA e ML. O bot do aplicativo Web com LUIS pode ser expandido para incluir Serviços Cognitivos do Azure adicionais, como Spell Check e Translator. Esses serviços permitem que você interprete palavras e frases caso o usuário escreva-as incorretamente, ou deixar que seu bot converse em vários idiomas. Ou, você pode usar Face e Personalizer para detectar qual cliente estava fazendo um pedido, com base no reconhecimento facial de sua câmera e sugerir automaticamente pizzas que ele pode gostar.

O ML era parte do aplicativo LUIS, mas há muitos outros recursos e ferramentas de ML disponíveis no Azure. A capacidade de processar grandes conjuntos de dados e computar modelos de dados ML em recursos de computação do Azure de alta performance reduz a entrada para você criar aplicações suportadas por alguns conjuntos de dados maiores. As aplicações são mais precisas e eficientes, e não há hardware para comprar ou ferramentas especiais para instalar, porque as DSVMs incluem todos os componentes necessários. Nem todas as aplicações são adequadas para IA e ML, mas como os clientes começam a esperar mais recursos inteligentes da sua empresa, esses serviços do Azure geralmente podem ajudar a diferenciá-lo.

Processamento de workload em lote

Duas outras áreas do Azure que pode ser de interesse em termos de big data e computação para ML são os serviços Azure Batch e HPC. O Azure Batch permite executar tarefas de computação repetitivas e grandes sem a necessidade de gerenciar clusters de agendadores para o trabalho. O Batch executa tarefas em VMs com seu próprio gerenciamento e agendador para ajudá-lo, uma vez que os conjuntos de escalas incluem dimensionamento automático e balanceamento de carga para VMs. Embora o Batch não esteja diretamente relacionado ao ML, se você precisar de outras tarefas de processamento de computação grandes, o Batch é bastante adequado.

Há também componentes de computação de alta performance (HPC) no Azure para grandes tamanhos de VM ou acesso a VMs de unidade de processamento gráfico (GPU). Ferramentas e suites específicas, como DataSynapse e Microsoft HPC Pack também podem ser usadas para executar aplicações que exigem uma grande quantidade de poder de computação.

Áreas como ML, Azure Batch e HPC são ótimos exemplos de como usar provedores de computação em nuvem como o Azure para executar tarefas de computação grandes. Você só paga pelos recursos de computação que você usa, assim você não precisa comprar e manter um equipamento caro que seja pouco utilizado.

17.4 Laboratório: Adicionar canais para comunicação do bot

Nos exemplos anteriores, você se comunicou com o bot por meio de uma janela de teste no portal do Azure. Canais permitem que você expanda como você pode interagir com o seu bot. Você pode permitir que seu bot comunique-se com o Skype ou Facebook Messenger, ou com aplicativos como o Microsoft Teams e Slack. O Azure Bot Service simplifica as etapas necessárias para integrar um bot com esses serviços externos:

1. No portal do Azure, selecione o bot do aplicativo Web e escolha Channels.
2. Escolha um canal que você goste, como o Skype.

   Outros canais geralmente exigem que você crie uma conexão de desenvolvedor, como Facebook ou Slack. O Skype permite que você copie e cole algum código HTML para fazê-lo funcionar.
3. Forneça todas as informações necessárias, como Bot Application ID. Você pode encontrar esse ID em Settings for Bot Management.
4. Se necessário, use o editor de código online para criar uma página HTML básica, como default. htm, no diretório wwwroot e cole qualquer código incorporado para o seu canal. Você pode abrir seu aplicativo Web a partir do portal do Azure e, em seguida, selecionar seu URL para abrir a página default. htm que inclui seu código de canal, tal como http://azuremol.azurewebsites.net/default.htm.

18 Automação do Azure

Sempre que possível, você não deve fazer login manualmente em um servidor e fazer alterações. O software não precisa ser instalado clicando nos botões em uma interface, e as atualizações não precisam ser feitas em arquivos de configuração em um editor de texto. Essas ações manuais introduzem uma oportunidade para que ocorram erros, o que pode resultar em configurações incorretas e falhas de aplicações. Se você quiser replicar a configuração de um servidor, conseguirá se lembrar de todos os passos que foram necessários para colocar o servidor existente em funcionamento? E se você precisar fazer isso de novo em seis meses?

No capítulo 16, vimos como verificar e aplicar automaticamente atualizações aos servidores. Essa magia aconteceu com o uso da Automação do Azure. Neste capítulo, examinamos como você pode criar, executar e editar runbooks e usar a configuração de estado desejada do PowerShell para instalar aplicações e configurar servidores automaticamente.

18.1 O que é Automação do Azure?

Uma conta de Automação do Azure reúne muitos elementos, como exibido na Figura 18.1. Um recurso principal é criar e executar scripts sob demanda ou em um cronograma definido. Você pode criar scripts no PowerShell ou Python e permitir que a plataforma do Azure manipule o agendamento e a execução desses runbooks. Você pode compartilhar credenciais e objetos de conexão e aplicar e reportar automaticamente as configurações desejadas de servidores. O gerenciamento de atualizações, que examinamos no capítulo 16, mantém seus servidores seguros e atualizados com os patches e atualizações de host mais recentes ao longo do ciclo de vida do seu ambiente de aplicação.

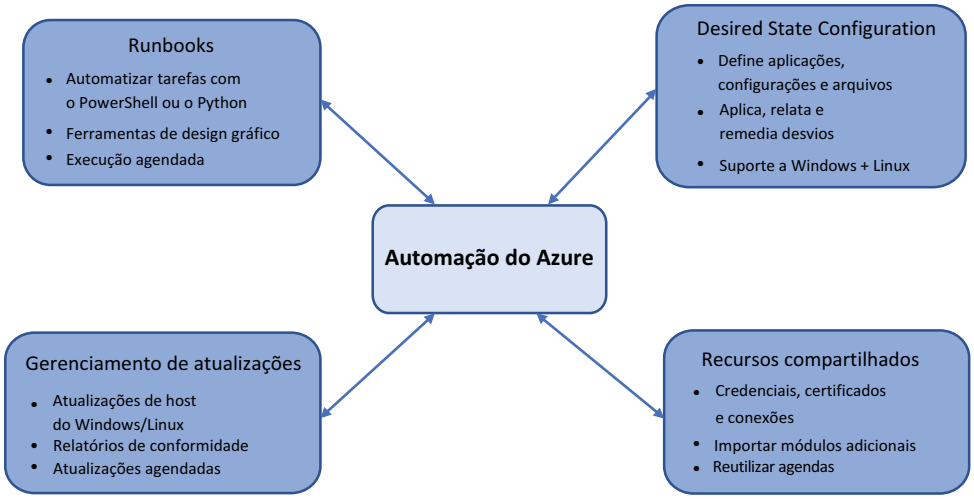


Figura 18.1 A Automação do Azure fornece muitos recursos relacionados. Um conjunto compartilhado de recursos, como credenciais, certificados, agendas e objetos de conexão, pode ser usado para executar automaticamente scripts do PowerShell ou Python em servidores de destino. Você pode definir o estado desejado de um servidor, e a Automação do Azure instala e configura o servidor apropriadamente. Atualizações de host e patches de segurança podem ser aplicados automaticamente. Todos esses recursos funcionam em servidores Windows e Linux, no Azure, em outros provedores de nuvem e na infraestrutura local.

Para simplificar o gerenciamento em vários runbooks ou configurações de estado desejadas em uma conta de Automação, você pode compartilhar os seguintes recursos:

- *Programações* permitem definir um conjunto de horários e recorrências que podem ser aplicados a cada tarefa de runbook ou gerenciamento de atualizações. Se desejar alterar posteriormente uma ocorrência regular, você poderá alterar um dos cronogramas compartilhados em vez de cada runbook individual ou tarefa de gerenciamento de atualizações que o usa.
- *Módulos* estendem a funcionalidade principal armazenando módulos adicionais do PowerShell. Os módulos básicos do Windows PowerShell e do Azure já estão disponíveis, mas módulos adicionais, como o gerenciamento do Linux, podem ser adicionados e usados em runbooks.
- *Credenciais* das diferentes contas que têm permissões para executar vários runbooks são armazenadas como ativos, não definidas em cada runbook. Essa abordagem permite atualizar e redefinir as credenciais conforme necessário, e cada runbook que faz uso delas é atualizado automaticamente. Assim, as credenciais não são armazenadas em texto sem formatação em runbooks, o que aumenta a segurança dos runbooks.
- *Conexões* definem propriedades de autenticação para entidades de serviço do Azure AD. Esse é um tipo especial de conta de usuário que permite que os runbooks acessem seus recursos do Azure. Essas conexões normalmente usam certificados digitais, não nomes de usuário e senhas, para fornecer uma camada adicional de segurança.
- *Certificados* geralmente são integrados aos ativos de conexão para fornecer uma maneira segura de verificar a identidade de uma entidade de serviço. Como com

credenciais básicas, você pode atualizar regularmente esses certificados em um local central, e cada runbook que faz uso deles pode acessar automaticamente os novos certificados. Você pode criar e armazenar seus próprios certificados para uso com runbooks ou definições de configuração de estado desejada.

- *Variáveis* fornecem um local central para valores de tempo de execução, como nomes, strings de local e inteiros a serem armazenados. Quando seus runbooks são executados, essas variáveis são injetadas. Essa abordagem limita a quantidade de recursos codificados dentro de cada runbook.

Trabalhe de forma mais inteligente, e não com mais trabalho

No capítulo 16, vimos como os serviços de gerenciamento do Azure funcionam juntos para monitorar e reportar servidores no Azure, na infraestrutura local ou em outros provedores de nuvem. Você instala e configura os agentes necessários em servidores remotos e, em seguida, fornece uma maneira para que eles se conectem de volta à infraestrutura do Azure.

A Automação do Azure também pode funcionar em plataformas e infraestruturas diferentes. O trabalhador de runbook híbrido pode executar runbooks de automação em servidores fora do Azure, por exemplo. Você continua usando os recursos de automação compartilhados que definem credenciais, conexões e certificados, somente desta vez, esses recursos podem ser usados para definir os componentes de autenticação para as diferentes plataformas. Você também pode usar configurações de estado desejadas em VMs que não são do Azure, tanto para Windows quanto para Linux.

Em todos os casos, um componente de gateway é instalado no ambiente remoto para atuar como um proxy para os comandos de Automação, pois eles são reenviados para os destinos designados. Essa abordagem de proxy de gateway fornece um único ponto de conexão para a automação em ambientes remotos e minimiza quaisquer preocupações de segurança, pois não há acesso direto a servidores remotos de outra forma.

Runbooks e definições de configuração de estado desejado talvez precisem ser editados um pouco para serem executados em servidores físicos na infraestrutura local em comparação com as VMs do Azure. Assim como o Azure Backup, Site Recovery, or Update Management, a vantagem da Automação do Azure é que ele fornece um único plano de gerenciamento e um conjunto de ferramentas para fornecer automação em todas as suas diferentes infraestruturas e servidores.

18.1.1 Criar uma conta de Automação do Azure

Vamos começar criando uma conta de Automação do Azure e examinar os runbooks padrão incluídos. Os runbooks de demonstração fornecem uma ótima estrutura para construir seus próprios runbooks, e há também um editor gráfico que você pode usar para arrastar e soltar blocos de construção para gerar scripts de automação.

Experimente agora

Para criar uma conta de Automação do Azure e runbooks de exemplo, conclua as etapas a seguir:

1. No portal do Azure, selecione Criar um recurso, no canto superior esquerdo.
2. Pesquise e selecione Automação e, depois, selecione Criar.

A opção Automação e Controle também cria um Operations Management Suite (OMS) e configura o Automation Hybrid Worker para gerenciar recursos fora do Azure. O OMS será substituído pelos serviços principais do Azure que examinamos nos capítulos anteriores. Por enquanto, opte por criar somente o recurso de automação.

3. Insira um nome, como azuremol, e crie um novo grupo de recursos, como `azuremolchapter18`.
4. Selecione a região mais apropriada do Azure mais próxima de você e aceite a opção Create Azure Run As Account (Criar Conta de Execução do Azure).

   A opção Create Run As Account (Criar Conta de Execução) cria contas adicionais no Azure AD. Certificados de segurança também são criados para permitir que as contas sejam autenticadas de forma automatizada, sem a necessidade de solicitação de usuário ou salvamento de uma senha. Você pode criar e especificar credenciais de conta regulares adicionais, definidas como um ativo de Automação, para fornecer um controle mais granular de quais contas são usadas para executar determinados runbooks.

Quando combinado com RBACs, que vimos no capítulo 6, você pode criar contas de execução específicas para runbooks que fornecem um conjunto limitado de permissões necessárias para realizar as tarefas que cada runbook, ou conjunto de runbooks, requer. Do ponto de vista de segurança, essa abordagem permite que você audite e controle como e quando essas contas são usadas. Evite a tentação de criar uma única conta de execução que forneça permissões de administrador, pois essa abordagem fornece pouca proteção contra abuso.

18.1.2 Ativos e runbooks de Automação do Azure

A conta de Automação do Azure criada na seção 18.1.1 inclui alguns runbooks de exemplo. Exemplos do PowerShell e do Python estão disponíveis. Ativos de conexão e certificados também são adicionados à conta de Automação para as contas de execução que foram criadas. Vamos explorar esses ativos de conexão compartilhados.

Experimente agora

Para ver os ativos configurados e os runbooks de exemplo, conclua as etapas a seguir:

1. No portal do Azure, selecione grupos de recursos à esquerda. Escolha seu grupo, como `azuremolchapter18`, e selecione sua conta da Automação do Azure, como `azuremol`.
2. Em Shared Resources (Recursos Compartilhados), no menu à esquerda, selecione Connections (Conexões).
3. Selecione AzureRunAsConnection, como mostrado na Figura 18.2.
4. Escolha Certificados no menu principal da conta da Automação em Recursos compartilhados e, em seguida, escolha o Certificado AzureRunAs. Como exibido na Figura 18.3, a impressão digital corresponde à RunAsConnection da etapa anterior.

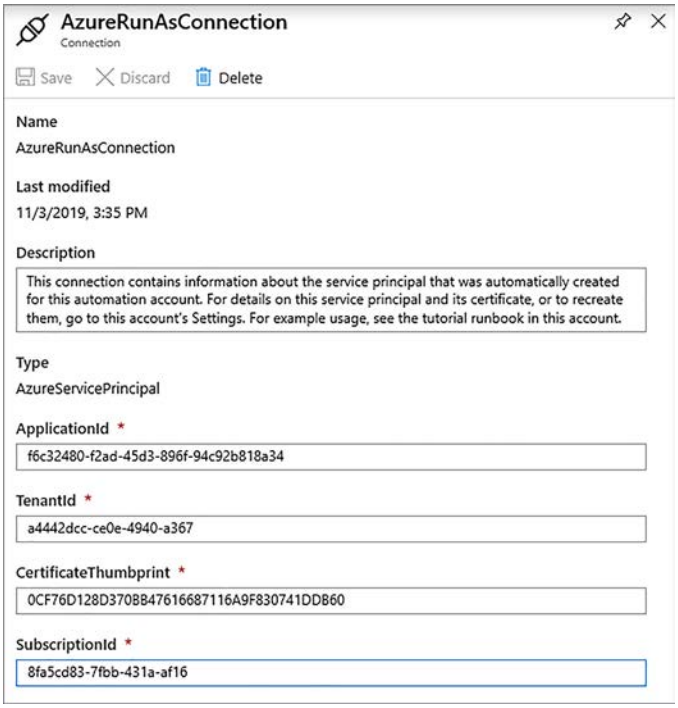


Figure 18.2 Informações na conta Executar como inclui um ApplicationId e TenantId — propriedades específicas do Azure AD que ajudam a identificar as credenciais dessa conta. Um CertificateThumbprint coincide com um certificado digital que veremos na próxima etapa.

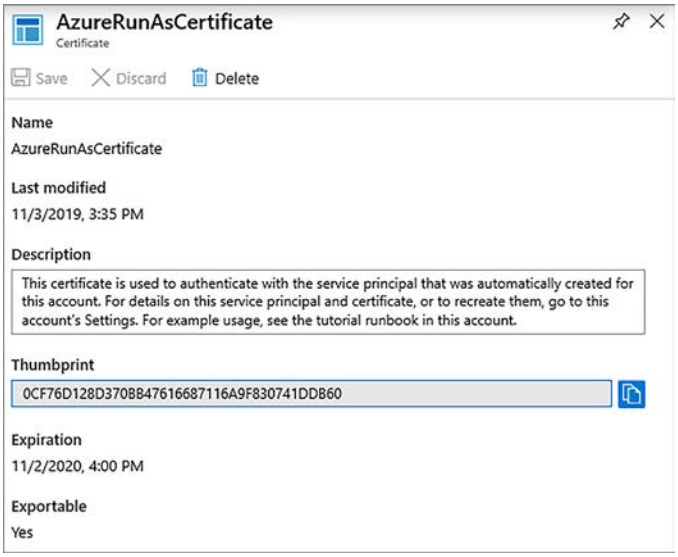


Figura 18.3 A impressão digital do RunAsCertificate corresponde à exibida em RunAsConnection. Em seus runbooks, você define qual ativo de conexão vai ser usado. O certificado apropriado é usado para entrar na conta do Azure.

5 Agora que você compreende os ativos para conexões e certificados, um dos exemplos de runbook. Escolha Runbooks no menu à esquerda na conta de Automação. Alguns runbooks de exemplo estão disponíveis.

6 Escolha o runbook do PowerShell chamado AzureAutomationTutorialScript.

7 Na parte superior do runbook de exemplo estão opções para iniciar, exibir e editar o runbook. Essas opções são autoexplicativas.

Você também pode fazer um agendamento, que permite criar ou selecionar um recurso compartilhado que define um cronograma para executar o runbook em um determinado momento, e Webhook, que permite criar um URL de webhook para executar o runbook de algum outro script ou ação. Escolha View (Exibir).

Automação do Azure e controle de origem com GitHub

Runbooks podem ser integrados com um sistema de controle de origem, como o GitHub. Um dos grandes benefícios de um sistema de controle de origem para seus runbooks é que ele fornece uma maneira para o gerenciamento de alterações ser documentado e reverter para versões anteriores dos runbooks caso ocorra algum problema.

Cada vez que você salva um runbook da Automação do Azure, uma nova versão é comprometida com o controle de origem. Você não precisa sair do editor do runbook, porque a plataforma do Azure e o sistema de controle de origem são configurados para funcionar de um lado para o outro. Se você tiver um problema com o novo runbook, poderá usar uma versão anterior do controle de origem que permite que os trabalhos continuem sendo executados sem demora e, em seguida, solucionar problemas da versão atualizada.

Usar o controle de origem também fornece um registro de quais alterações ocorreram e quando. Se você precisar auditar seus runbooks ou entender como eles se desenvolveram ao longo do tempo, os sistemas de controle de origem fornecem uma ótima maneira de ver as diferenças com cada revisão.

18.2 Runbook de exemplo de Automação do Azure

Vamos examinar como o exemplo de runbook do PowerShell, AzureAutomationTutorialScript, conecta-se ao Azure e coleta informações sobre seus recursos. Você pode acompanhar com o runbook de exemplo Python se preferir; o layout é semelhante. O PowerShell e o Python são as únicas linguagens atualmente suportadas nos runbooks de Automação do Azure. A listagem a seguir configura as credenciais de conexão no runbook.

Listagem 18.1 Configurar credenciais de conexão

```
$connectionName = "AzureRunAsConnection"
try
{
  # Get the connection "AzureRunAsConnection "
  $servicePrincipalConnection=Get-AutomationConnection -Name
➥$connectionName
```

Cria um objeto para $connectionName

Faz a solicitação de conexão

Cria um objeto de entidade de serviço

```
    "Logging in to Azure..."
    Add-AzureRmAccount `
        -ServicePrincipal `
        -TenantId $servicePrincipalConnection.TenantId `
        -ApplicationId $servicePrincipalConnection.ApplicationId `
        -CertificateThumbprint
➥$servicePrincipalConnection.CertificateThumbprint
}
```

Entra no Azure

O código começa criando um objeto para `$ConnectionName`. No exercício "Experimente agora", você viu que um ativo de conexão padrão para AzureRunAsConnection foi criado. À medida que você cria seus próprios runbooks, convém criar contas de execução e ativos de conexão adicionais para separar os runbooks e as credenciais que eles usam. As partes de conexão e o tratamento de exceções que vemos em seguida devem ser comuns em todos os runbooks. Conforme necessário, você pode alterar o ativo de conexão de execução a ser usado.

Em seguida, uma instrução `try` é usada para fazer a solicitação de conexão. Um objeto de entidade de serviço chamado `$servicePrincipalConnection` é criado com base em `$connectionName`. Em seguida, o runbook faz logon no Azure com `Add-AzureRmAccount` e usa o objeto `$servicePrincipalConnection` para obter `TenantId`, `ApplicationId` e `Certificate-Thumbprint`. Discutimos anteriormente esses parâmetros como parte do ativo de conexão. O ativo de certificado que coincide com a impressão digital de `$servicePrincipalConnection` é usado para concluir o login no Azure.

A próxima listagem mostra que, se a conexão falhar, o runbook capturará o erro e interromperá a execução.

Listagem 18.2 Capturar um erro e parar a execução do runbook

```
catch {
     if (!$servicePrincipalConnection)
     {
          $ErrorMessage = "Connection $connectionName not found."
          throw $ErrorMessage
     } else{
          Write-Error -Message $_.Exception
          throw $_.Exception
     }
}
```

A instrução `catch` manipula erros como parte da tentativa de login. Se uma conexão de entidade de serviço não é encontrada, um erro é gerado. Esse erro geralmente significa que o ativo de conexão especificado não pode ser encontrado. Verifique o nome e a ortografia da sua conexão.

Caso contrário, o objeto de conexão foi encontrado e a entidade de serviço foi usada para fazer login, mas esse processo de autenticação não teve êxito. Essa falha pode vir de um certificado que não é mais válido ou de uma conta Executar como que não está mais sendo habilitada. Essa funcionalidade mostra como você pode revogar uma conta no Azure AD e garantir que os runbooks que usam as credenciais não possam mais ser executados.

Agora, o runbook obtém uma lista de todos os recursos do Azure.

Listagem 18.3 Obter uma lista de recursos do Azure

```
$ResourceGroups = Get-AzureRmResourceGroup
foreach ($ResourceGroup in $ResourceGroups)
{
    Write-Output ("Showing resources in resource group "
➥+ $ResourceGroup.ResourceGroupName)
     $Resources = Find-AzureRmResource -ResourceGroupNameContains
➥$ResourceGroup.ResourceGroupName |
➥Select ResourceName, ResourceType
     ForEach ($Resource in $Resources)
     {
         Write-Output ($Resource.ResourceName + " of type "
➥+  $Resource.ResourceType)
     }
     Write-Output ("")
}
```

A parte final do runbook é para onde seu código de runbook iria. Um objeto é criado para `$ResourceGroups` que obtém uma lista de todos os grupos de recursos do Azure disponíveis. Em seguida, um loop `foreach` passa pelos grupos de recursos, localiza uma lista de recursos e grava uma lista de nomes e tipos de recursos.

Este exemplo básico mostra como você pode interagir com o Azure depois que o runbook for autenticado na assinatura. Se você implementar RBAC na conta de execução, somente os grupos de recursos em que a conta tem permissões de visualização serão retornados. Essa abordagem do RBAC destaca por que é um bom princípio de segurança criar e usar contas Executar como para limitar o acesso que os runbooks têm aos recursos em seu ambiente do Azure. Sempre tente fornecer a menor quantidade necessária de privilégios.

Se tudo isso no PowerShell ou Python for novo para você, não se preocupe. Ambos são ótimas linguagens básicas de script que também podem ser usados para desenvolver aplicações complexas e eficientes. Como desenvolvedor, qualquer linguagem deve ser relativamente fácil para você escolher e usar. Se você for um profissional de TI, automatizar tarefas libera seu tempo para executar todos os outros trabalhos, e o PowerShell ou Python são bons lugares para começar. A Manning Publications também tem alguns outros ótimos livros para ajudá-lo!

18.2.1 Executar e exibir a saída de um runbook de exemplo

Agora que você já viu o que o script de runbook de exemplo contém e como os ativos de conexão e de certificado são usados, vamos executar o runbook e examinar a saída.

Experimente agora

Para ver o runbook em ação, conclua as etapas a seguir:

1. Feche a janela que mostra o conteúdo do runbook e retorne à visão geral do AzureAutomationScriptTutorial.

2 Selecione Start (Iniciar) na parte superior da janela do runbook.

3 Confirme se deseja iniciar o runbook e aguarde alguns segundos para que o runbook comece a ser executado.

4 Selecione Output (Saída), como exibido na Figura 18.4, e veja a janela do console à medida que o runbook faz login no Azure, obtém uma lista de grupos de recursos e percorre e gera a lista de recursos em cada grupo.

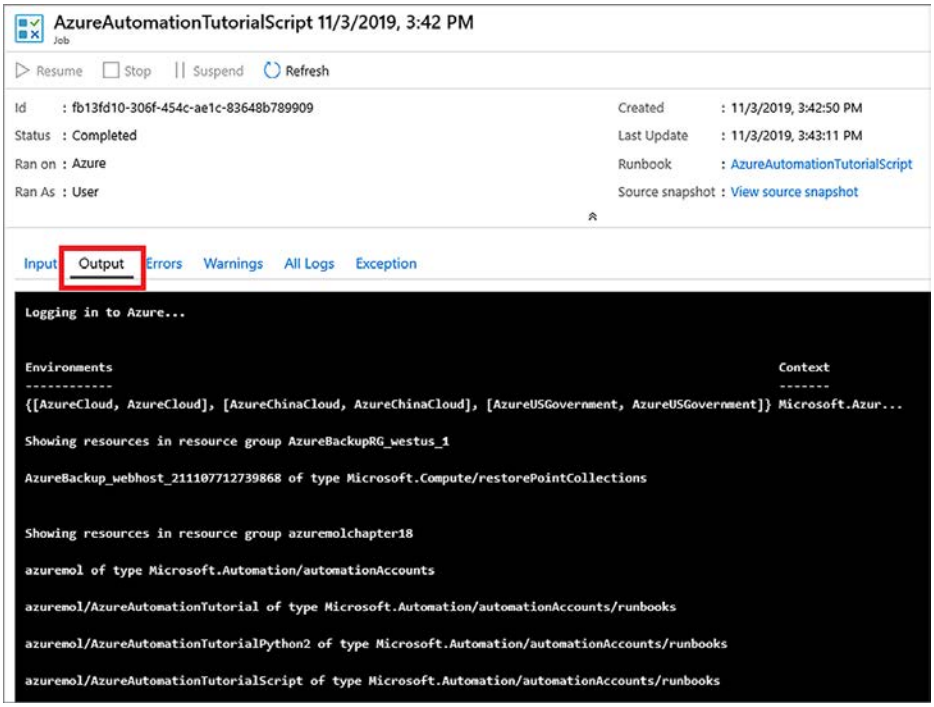


Figura 18.4 Você pode ver a saída do runbook, junto com os logs que são gerados ou erros e avisos. Esse exemplo básico é concluído em poucos segundos, mas runbooks mais complexos podem levar mais tempo. Você pode monitorar o status desses runbooks mais longos e parar ou pausar sua execução, conforme necessário.

Os runbooks de Automação não precisam existir isoladamente. Um runbook pode executar outro runbook. Essa capacidade permite que você crie uma automação complexa de várias etapas e minimize a duplicação de código. À medida que você projetar e criar runbooks, tente quebrá-los em blocos de código pequenos e separados. As funções comuns que você pode reutilizar, como fazer login no Azure e gerar uma lista de recursos ou uma lista de VMs, devem ser criadas como runbooks pequenos que podem ser incluídos em runbooks maiores. Conforme novos cmdlets do PowerShell são liberados ou os parâmetros são alterados, você pode atualizar rapidamente um único runbook compartilhado que inclua esses cmdlets, em vez de precisar atualizar vários runbooks diferentes. Em primeiro lugar, pode não parecer que runbooks

menores e reutilizáveis valem um pouco de trabalho extra, mas à medida que seu ambiente e uso da Automação aumentarem, você perceberá os benefícios! Muito do que você fez neste livro foi em implantações menores, mas comece a pensar sobre como implantar e gerenciar aplicações em grande escala.

18.3 PowerShell Desired State Configuration (DSC)

O capítulo 12 introduziu o conceito de extensões de VM. Uma *extensão* é um componente de software pequeno que é instalado em uma VM para executar uma determinada tarefa. A extensão de diagnóstico de VM foi instalada em uma VM para permitir que as métricas de performance e os logs de diagnóstico sejam reportados de volta à plataforma do Azure dentro da VM. Isso é ótimo, mas também conversamos um pouco sobre como você pode instalar automaticamente o software.

Uma forma de instalar o software e configurar um servidor é usar o PowerShell Desired State Configuration (DSC). Com o DSC, você define como deseja que um servidor seja configurado, o estado desejado. Você pode, por exemplo, definir pacotes a serem instalados, recursos a serem configurados ou arquivos a serem criados. O que é ótimo sobre o DSC é que ele vai além da primeira ação de instalação e configuração. Com o tempo, os servidores geralmente passam por eventos de manutenção ou resolução de problemas em que configurações e pacotes são alterados manualmente. Em seguida, o servidor se desvia do estado desejado que você definiu inicialmente. A Figura 18.5 mostra como a Automação do Azure pode atuar como um servidor central que armazena as definições de DSC, permitindo que os servidores de destino recebam suas configurações e reportem sua conformidade.

O Local Configuration Manager (LCM) em cada servidor de destino controla o processo de conexão com o servidor de extração de Automação do Azure, recebendo e analisando a definição de DSC e aplicando e reportando a conformidade. O mecanismo LCM pode operar sem um servidor de extração, onde você chama localmente o processo para ler e aplicar uma

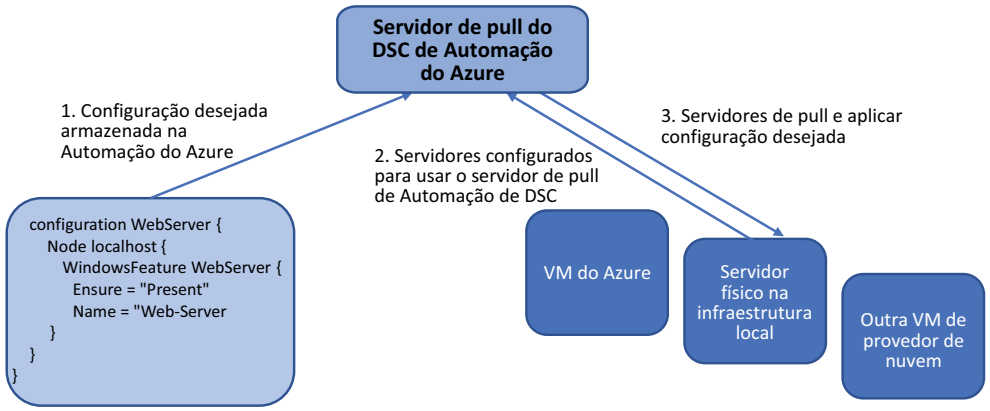


Figura 18.5 A configuração de estado desejado para um servidor é criada e armazenada em Automação do Azure. A conta de Automação atua como um servidor de extração, que permite que os servidores conectados extraiam a configuração necessária de um local central. Diferentes modos de configuração podem ser definidos para o comportamento de remediação do servidor se sua configuração se desviar do estado desejado.

definição de DSC. Nesse modo, em que você envia manualmente a configuração para o mecanismo LCM, perde-se um monte de controles centrais e relatórios que muitas vezes são necessários ao gerenciar muitos servidores.

Também há flexibilidade na forma como os servidores de destino processam as definições de DSC recebidas do servidor de extração de Automação do Azure. Você pode configurar o DSC para operar em um dos três modos de configuração:

- *Aplicar somente:* o estado desejado é enviado e aplicado ao servidor de destino; e isso é tudo. Isso é como o comportamento da extensão de script personalizado do Azure em que todas as configurações ou instalações são aplicadas quando implantadas pela primeira vez, mas não há nenhum processo implementado para interromper manualmente essas configurações alterando o ciclo de vida do servidor.
- *Aplicar e monitorar:* depois que o servidor tiver o estado desejado aplicado, o DSC continuará monitorando as alterações que fazem com que o servidor se desvie dessa configuração inicial. Um relatório central pode ser usado para exibir servidores que não são mais compatíveis com seu estado desejado. Essa configuração é um bom equilíbrio entre a necessidade de manter um servidor compatível com o estado desejado e fornecer um elemento de interação humana para decidir sobre as opções de remediação.
- *Aplicar e corrigir automaticamente*: a configuração mais automatizada e independente aplica o estado desejado, monitora os desvios e corrige automaticamente o servidor caso ocorram alterações para garantir que ela permaneça compatível. Há um risco de que as alterações manuais legítimas sejam substituídas e, em vez disso, retornadas para o estado desejado configurado, mas esse modo de configuração garante que as configurações que você atribui sempre tenham prioridade.

O PowerShell DSC pode ser usado em VMs executadas em outros provedores de nuvem, bem como em VMs na infraestrutura local e em servidores físicos. Graças ao .NET Core, o PowerShell DSC também pode ser usado em servidores Linux; portanto, não é uma solução somente para Windows. Esse suporte a vários provedores e sistemas operacionais torna o PowerShell uma opção poderosa para configurar e gerenciar servidores em escala.

Você pode criar e manter seu próprio servidor de extração de DSC, mas os recursos internos de Automação do Azure fornecem alguns benefícios adicionais:

- As credenciais são gerenciadas centralizadamente e os certificados são gerados automaticamente.
- A comunicação entre o servidor de extração de DSC e os servidores de destino é criptografada.
- Os relatórios incorporados são fornecidos para conformidade com DSC, e há integração com o Log Analytics para gerar relatórios e alertas mais detalhados.

Esta seção é muito mais que um curso intensivo do PowerShell DSC; é um componente poderoso por si só e já tem estado amplamente disponível por alguns anos. Quando combinado com Automação do Azure, o DSC é uma ótima opção para automatizar a instalação e a configuração do software. Recapitule, por exemplo, os capítulos anteriores sobre conjuntos de escala de máquina virtual. Você pode aplicar uma configuração de DSC ao conjunto de escalas com Automação do Azure e, como cada VM é criada no conjunto de escalas, ela será automaticamente configurada com os componentes e os arquivos de aplicação necessários.

18.3.1 Definir e usar o PowerShell DSC e um servidor de extração de Automação do Azure

Eu espero que esse mergulho no PowerShell DSC tenha dado a você uma ideia do que é possível. Vamos usar o PowerShell DSC para automatizar o exemplo de instalação de um servidor Web básico em uma VM.

Experimente agora

Para ver o PowerShell DSC em ação, conclua as etapas a seguir:

1. Crie uma VM do Windows Server 2019 data center e abra a porta TCP 80 para tráfego HTTP. Agora, no capítulo 18, não estamos mais levando você pela mão. Você pode criar a VM no Cloud Shell ou no portal do Azure: você decide. Use o grupo de recursos que você criou nos exercícios anteriores, como `azuremolchapter18`. Você pode continuar com as próximas etapas à medida que a VM é implantada.
2. No seu computador local, crie um arquivo chamado webserver.ps1, digite o código a seguir, salve e feche o arquivo quando terminar:

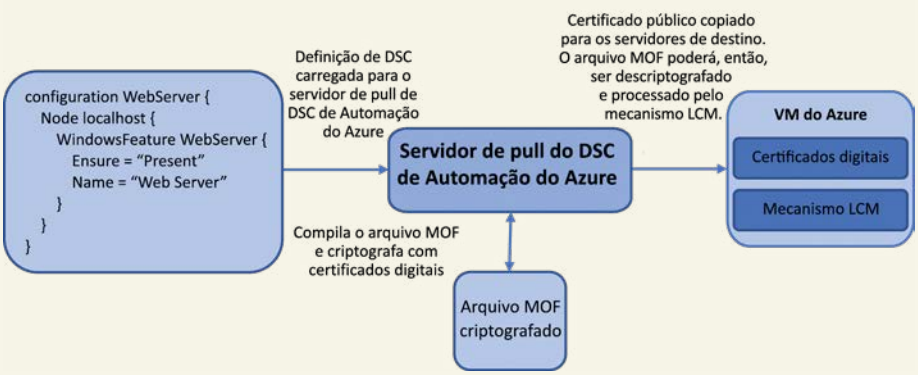
```
configuration WebServer {
   Node localhost {
      WindowsFeature WebServer {
         Ensure = "Present"
         Name = "Web-Server"
      }
   }
}
```

3. No portal do Azure, selecione seu grupo de recursos e escolha sua conta de Automação.
4. À esquerda, escolha a Configuração de Estado (DSC). Selecione a guia Configurações e, na parte superior da janela, escolha Adicionar uma configuração.
5. Navegue e selecione seu arquivo webserver.ps1. O nome da configuração deve coincidir com o nome do arquivo; portanto, deixe o nome padrão do servidor Web e, em seguida, escolha OK.

   Leva alguns instantes para carregar e criar a configuração.
6. Quando estiver pronto, selecione a configuração na lista e escolha Compile (Compilar).

Nos bastidores do DSC

Vamos fazer uma pausa para falar sobre o que acontece quando você compila a configuração, conforme exibido na figura a seguir. Para distribuir as definições de DSC, seus arquivos do PowerShell são convertidos em um arquivo Managed Object Format (MOF). Esse tipo de arquivo é usado para mais do que apenas o PowerShell DSC e permite alterações de configuração em componentes do Windows de uma maneira central e bem compreendida. Qualquer definição de DSC, não apenas em Automação do Azure, deve ser compilada antes de poder ser aplicada a um servidor de destino. O mecanismo LCM só aceita e processa arquivos MOF.



O servidor de extração de DSC de Automação do Azure compila automaticamente a definição de DSC que você fornece em um arquivo Managed Object Format (MOF). Os certificados digitais gerenciados pela Automação são usados para criptografar o arquivo MOF. Os servidores de destino de DSC recebem os certificados digitais públicos necessários e permitem que o mecanismo LCM descriptografe e processe o arquivo MOF. Então, o estado desejado pode ser aplicado ao servidor.

Como o arquivo MOF define o estado completo de seus servidores, você deve proteger seu conteúdo. Se um invasor conhece todos os componentes da aplicação instalados e o local de vários arquivos de configuração e código personalizado, a chance de seus servidores serem comprometidos aumenta. As versões recentes do PowerShell criptografam todo o arquivo MOF. A Automação do Azure gera automaticamente os certificados e chaves digitais necessários quando um servidor de destino está configurado para DSC, o que lhe permite utilizar facilmente arquivos MOF criptografados. Automação também criptografa o tráfego entre o servidor de extração de DSC e os nós de destino, não apenas o arquivo MOF.

O processo de compilação em Automação do Azure converte a definição de DSC que você fornece em um arquivo MOF e criptografa o arquivo MOF com as chaves e certificados digitais. O processo de compilação da definição de DSC leva alguns segundos, mas protege muito seu ambiente (esse é apenas outro exemplo do Azure protegendo seus recursos por padrão).

7 Para aplicar a configuração à sua VM, selecione a guia nós nas janelas de Configuração do Estado (DSC). Selecione Adicionar e escolha a VM que você criou em etapas anteriores.

8 Escolha Connect (Conectar).

9 No menu suspenso Node Configuration Name (Nome de Configuração do Nó), escolha webserver.localhost.

10 Defina o modo de configuração como ApplyAndMonitor e selecione OK.

Pode levar alguns minutos para permitir que a VM use o servidor de extração do Azure PowerShell DSC e aplique o estado desejado inicial.

11 Quando o portal do Azure informar que a configuração foi aplicada, selecione seu grupo de recursos. Em seguida, selecione a VM que você criou nas etapas anteriores.

12 Você abriu a porta TCP 80 para a VM quando a criou? Se não abriu, crie uma regra do grupo de segurança de rede para permitir o tráfego e, em seguida, abra o IP público da VM em um navegador da Web. O processo DSC instala o servidor Web do IIS, e a página da Web padrão é carregada, como exibido na Figura 18.6.

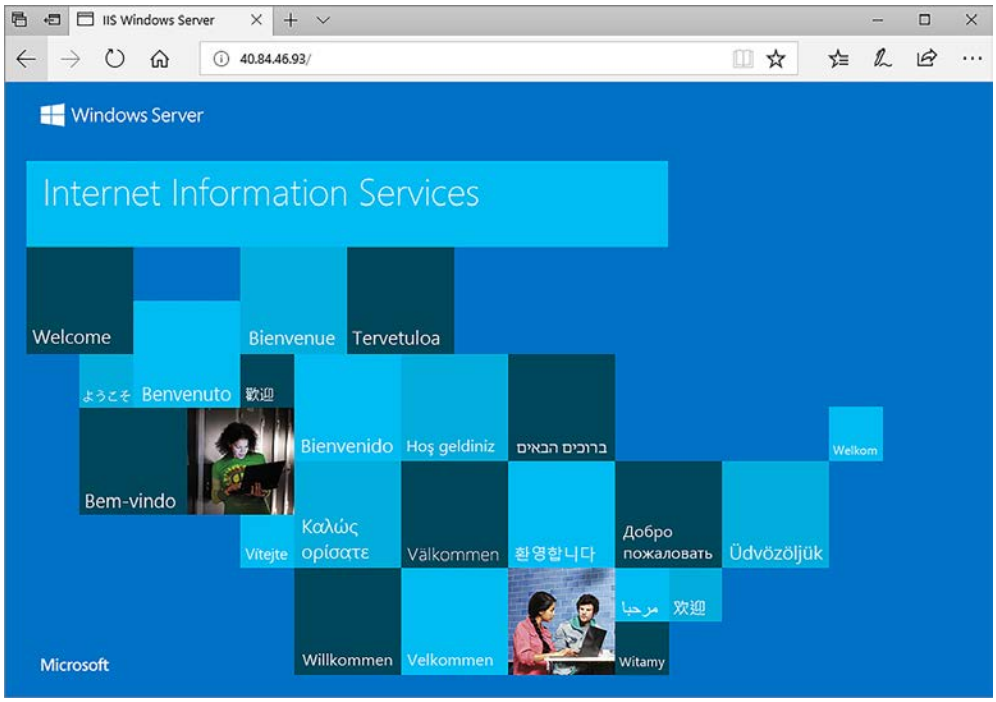


Figura 18.6 Depois que a VM tiver sido conectada ao Azure Automation DSC, o estado desejado será aplicado, e o servidor Web do IIS será instalado.

Este exemplo básico do PowerShell DSC só instala o recurso de servidor Web. Você pode usar o PowerShell DSC para configurar o servidor Web do IIS ou copiar o código da aplicação para a VM e executar o site. Definições complexas de DSC podem ser usadas para preparar a VM para atender ao tráfego dos clientes da pizzaria sem interação manual. Novamente, pense em como você deve criar suas aplicações para dimensionar automaticamente — a VM não pode esperar que alguém instale e configure tudo manualmente.

18.4 Laboratório: usar o DSC com Linux

Apenas para provar que o PowerShell DSC funciona em servidores Linux, vamos criar uma VM do Ubuntu, instalar os pré-requisitos necessários e, em seguida, instalar um servidor Web NGINX básico com DSC. Em produção, você pode usar uma imagem de VM personalizada que já tenha os componentes de gerenciamento instalados e, em seguida, aplicar definições do PowerShell DSC normalmente:

1. O PowerShell DSC para Linux tem algumas limitações nas distribuições do Linux a que oferece suporte sem configuração adicional. Portanto, para manter esse exercício de laboratório de fim de capítulo o mais simples possível, crie uma VM CentOS 7.7 ou posterior e abra a porta 80.
2. Na conta de Automação do Azure, selecione Módulos no menu à esquerda.
3. Selecione Browse Gallery e, em seguida, pesquise, selecione e importe o módulo nx para gerenciar os recursos do Linux DSC.
4. No seu computador local, crie um arquivo chamado httpd.ps1 e digite o seguinte código:

```
configuration httpd {
    Import-DSCResource -Module nx
    Node localhost {
        nxPackage httpd {
            Name = "httpd"
            Ensure = "Present"
            PackageManager = "yum"
        }
        nxService httpd {
            Name = "httpd"
            State = "running"
            Enabled = $true
            Controller = "systemd"
        }
    }
}
```

5. Adicione uma configuração de DSC à conta da Automação do Azure, faça upload do arquivo httpd.ps1 e compile a configuração.
6. Adicione um nó de DSC à sua conta da Automação do Azure, selecione sua VM CentOS e escolha seu nome de configuração de nó httpd.localhost.

   Novamente, leva alguns minutos para que a VM aplique a configuração desejada. Você pode exibir a lista de VMs conectadas e seu status de conformidade na janela de nós DSC. A VM é reportada como compatível quando o LCM aceita e aplica o arquivo MOF, mas os comandos para instalar e configurar os pacotes httpd necessários dentro da VM podem demorar mais um ou dois minutos.
7. Selecione a VM CentOS no portal do Azure, obtenha o endereço IP público e insira o endereço IP da sua VM em um navegador da Web para ver o servidor Web instalado pelo DSC. Se o site não for carregado, aguarde um ou dois minutos para que o processo de instalação seja concluído e, em seguida, atualize a página.

Se você quiser realmente experimentar o admirável mundo novo da Microsoft e Linux, instale o PowerShell em sua VM Linux. Conclua as etapas rápidas de configuração em http://mng.bz/VgyP para ver como podem ser os scripts do PowerShell entre plataformas.

19 Contêineres do Azure

Contêineres, Docker e Kubernetes ganharam um enorme número de adeptos em poucos anos. Da mesma forma que a virtualização de servidores começou a mudar a forma como os departamentos de TI executavam seus data centers em meados da década de 2000, as modernas ferramentas e orquestradores de contêiner agora estão reorganizando a maneira como criamos e executamos as aplicações. Não há nada que conecte de forma inerente o crescimento de contêineres com a computação na nuvem, mas, quando combinados, eles fornecem uma ótima maneira de desenvolver aplicações com uma abordagem nativa de nuvem. Livros inteiros foram escritos sobre Docker e Kubernetes, mas vamos a uma introdução rápida e ver como você pode executar contêineres no Azure rapidamente. Há um conjunto avançado de serviços do Azure dedicados a contêineres que se alinham mais à abordagem PaaS. Você pode se concentrar em como criar e executar suas aplicações, em vez de como gerenciar a infraestrutura de contêiner, a orquestração e os componentes de cluster.

Neste capítulo, examinaremos o que são contêineres, como o Docker foi envolvido e o que o Kubernetes pode fazer por você. Para ver como executar rapidamente uma única instância de contêiner ou várias instâncias de contêiner em um cluster, exploraremos Instância do Contêiner do Azure (ACI) e Serviço Azure Kubernetes (AKS).

19.1 O que são contêineres?

Houve uma enorme onda de interesse e adesão dos contêineres nos últimos anos, e eu ficaria impressionado se você não tivesse ouvido falar de uma empresa que liderou esse encargo: Docker. Mas o que exatamente é um contêiner e o que o Docker tem a ver com isso?

Primeiro, vamos analisar um host de virtualização tradicional que executa VMs. A Figura 19.1 é como o diagrama do capítulo 1, em que cada VM tem seu próprio hardware virtual e sistema operacional convidado.

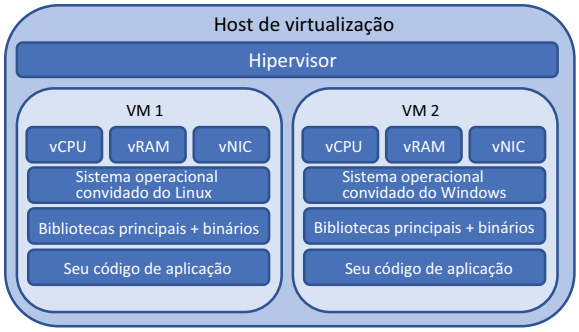


Figura 19.1 Com uma infraestrutura de VM tradicional, o hypervisor em cada host de virtualização fornece uma camada de isolamento, fornecendo a cada VM seu próprio conjunto de dispositivos de hardware virtual, como uma CPU virtual, RAM virtual e NICs virtuais. A VM instala um sistema operacional convidado, como o Ubuntu Linux ou o Windows Server, que pode usar esse hardware virtual. Finalmente, você instala sua aplicação e quaisquer bibliotecas necessárias. Esse nível de isolamento torna as VMs muito seguras, mas adiciona uma camada de sobrecarga em termos de recursos de computação, armazenamento e tempos de inicialização.

Um contêiner remove o hardware virtual e o sistema operacional convidado. Tudo o que está incluído em um contêiner são as principais aplicações e bibliotecas necessárias para executar sua aplicação, conforme mostrado na figura 19.2.

Muitas VMs podem ser executadas em um único hypervisor, cada VM com seu próprio sistema operacional virtual convidado, hardware virtual e pilha de aplicações. O hypervisor gerencia solicitações do hardware virtual de cada VM, agenda a alocação e o compartilhamento desses recursos físicos de hardware e reforça a segurança e o isolamento de cada VM. O trabalho do hipervisor é mostrado na figura 19.3.

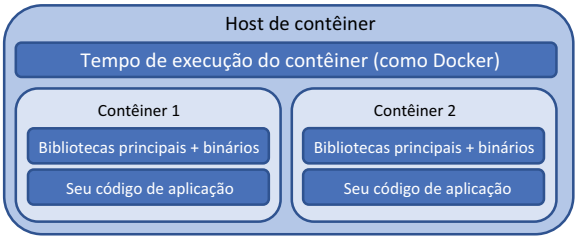


Figura 19.2 Um contêiner contém apenas as bibliotecas principais, os binários e o código da aplicação necessários para executar uma aplicação. O contêiner é leve e portátil, porque remove o sistema operacional convidado e a camada de hardware virtual, o que também reduz o tamanho do contêiner no disco e os tempos de inicialização.

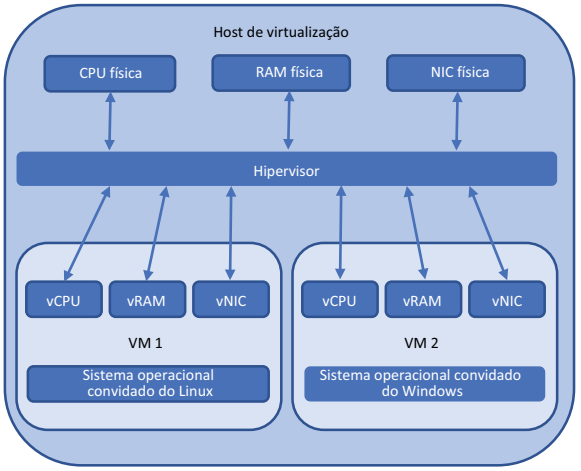


Figura 19.3 Em um host de VM tradicional, o hipervisor permite o agendamento de solicitações do hardware virtual em cada VM para o hardware físico subjacente e a infraestrutura. O hypervisor normalmente não tem consciência de quais instruções específicas o sistema operacional convidado está planejando no tempo de CPU físico, apenas o tempo de CPU é necessário.

Vários contêineres também podem ser executados em um único host. O host do contêiner recebe as várias chamadas do sistema de cada contêiner e agenda a alocação e a distribuição dessas solicitações em um kernel de base compartilhado, sistema operacional e recursos de hardware. Os contêineres fornecem um isolamento lógico dos processos de aplicações. O trabalho do tempo de execução do contêiner é mostrado na figura 19.4.

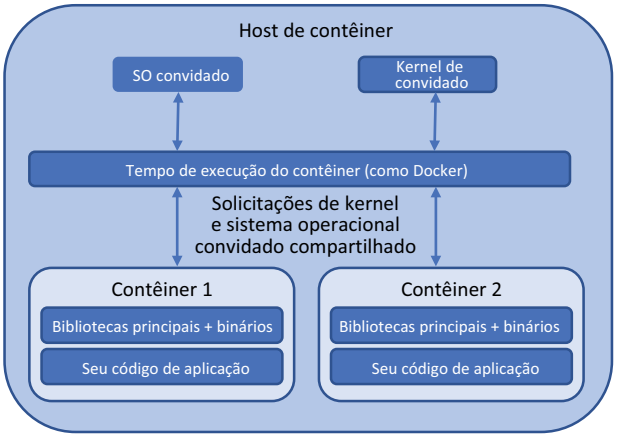


Figura 19.4 Os contêineres têm um sistema operacional convidado e um kernel comuns. O tempo de execução do contêiner manipula as solicitações dos contêineres para o kernel compartilhado. Cada contêiner é executado em um espaço de usuário isolado e alguns recursos de segurança adicionais protegem os contêineres uns dos outros.

Geralmente, os contêineres são muito mais leves que as VMs. Os contêineres podem ser inicializados mais rapidamente que as VMs, geralmente em segundos, em vez de minutos. O tamanho de uma imagem de contêiner geralmente é de apenas dezenas ou centenas de MBs, em comparação com muitas dezenas de GBs das VMs. Ainda existem limites e controles de segurança em vigor, mas é importante lembrar que cada contêiner compartilha tecnicamente o kernel de outros contêineres no mesmo host.

Experimente agora

Leva alguns minutos para criar um cluster dos Serviços Azure Kubernetes para uso nos próximos exercícios, portanto, conclua as etapas a seguir e continue lendo o capítulo:

1. Abra o portal do Azure e escolha o ícone do Cloud Shell no menu superior.
2. Crie um grupo de recursos. Atribua um nome, como `azuremolchapter19`, e um local, como `eastus`. A disponibilidade de região do Serviço Azure Kubernetes pode variar, por isso escolha uma região importante, como `eastus` ou `westeurope`. (Para obter uma lista atualizada da disponibilidade da região, veja https://azure.microsoft.com/regions/services.)

```
az group create --name azuremolchapter19 --location eastus
```

3 Para criar um cluster do Kubernetes, especifique `--node-count` como `2` e use conjuntos de escalas e zonas de disponibilidade (que você conheceu nos capítulos anteriores):

```
az aks create \
  --resource-group azuremolchapter19 \
  --name azuremol \
  --node-count 2 \
  --vm-set-type VirtualMachineScaleSets \
  --zones 1 2 3 \
  --no-wait
```

O parâmetro final `--no-wait` retorna o controle para o Cloud Shell enquanto o restante do cluster é criado. Continue a leitura enquanto o cluster é implantado.

O Docker juntou-se ao grupo de contêineres com um conjunto de ferramentas e formatos padrão que definiam como criar e executar um contêiner. O Docker se baseia em recursos existentes no nível do kernel do Linux e do Windows para fornecer uma experiência de contêiner consistente e portátil em todas as plataformas. Um desenvolvedor pode criar um contêiner do Docker no notebook que executa o macOS, validar e testar o aplicativo e, em seguida, executar exatamente o mesmo contêiner do Docker exato, sem modificação, em um cluster de servidor mais tradicional baseado em Linux ou Windows na infraestrutura local ou no Azure. Todos os binários, bibliotecas e arquivos de configuração necessários da aplicação são agrupados como parte do contêiner, portanto, o sistema operacional do host subjacente não se torna um fator ou restrição de design.

A importância do Docker não deve ser omitida aqui. Geralmente, os termos *contêiner* e *Docker* são usados de forma intercambiável, embora isso não seja tecnicamente exato. O Docker é um conjunto de ferramentas que ajuda os desenvolvedores a criar e executar contêineres de maneira consistente, confiável e portátil. A facilidade de usar essas ferramentas levou a uma adesão rápida e fez com que a tecnologia de contêineres subjacente, que já existia de uma forma ou de outra há mais de uma década, tornasse-se a tendência do momento. Os desenvolvedores aderiram aos contêineres e à plataforma do Docker, e os departamentos de TI tiveram que se esforçar para recuperar o atraso desde então.

O Docker participa da Open Container Initiative. O formato e as especificações definidos pelo Docker para estabelecer como um contêiner deve ser empacotado e executado eram alguns dos princípios básicos desse projeto. O trabalho do Docker continuou e foi desenvolvido por outras pessoas. Grandes colaboradores na área de contêineres incluem a IBM e a Red Hat, que contribuem com alguns dos principais projetos e códigos que alimentam as atuais plataformas de contêineres. A Open Container Initiative e o formato de design para embalagens e tempos de execução de contêineres são importantes porque permitem que cada fornecedor coloque suas próprias ferramentas sobre os formatos comuns, permitindo que você mova o contêiner subjacente entre as plataformas e tenha a mesma experiência básica.

19.2 A abordagem de microsserviços para aplicações

Se os contêineres oferecerem um conceito de isolamento semelhante às VMs, você poderá executar o mesmo tipo de workloads em uma VM? Bem, sim e não. Só porque você pode fazer algo não significa necessariamente que você deveria fazer! Os contêineres podem ser usados para executar qualquer workload com o qual você se sinta confortável, e há benefícios em termos de recursos de portabilidade e orquestração que

examinaremos na seção 19.4. Para maximizar os benefícios dos contêineres e se preparar para o sucesso, aproveite a oportunidade para adotar um modelo mental ligeiramente diferente ao começar a trabalhar com contêineres. A Figura 19.5 compara o modelo de aplicação tradicional com uma abordagem de microsserviços.

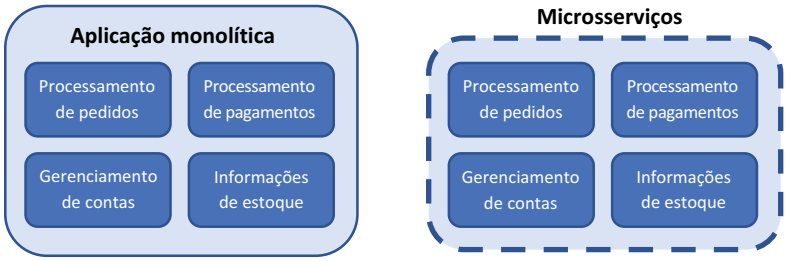


Figura 19.5 Em uma aplicação monolítica tradicional, toda a aplicação é executada como uma única aplicação. A aplicação pode ter vários componentes, mas é executada a partir de uma única instalação e é corrigida e atualizada como uma única instância. Com microsserviços, cada componente é dividido em seu próprio serviço de aplicação e unidade de execução. Cada componente pode ser atualizado, corrigido e escalado independentemente dos outros.

Uma VM padrão inclui uma instalação completa do sistema operacional convidado, como o Ubuntu ou o Windows Server. Essa instalação básica do sistema operacional inclui centenas de componentes, bibliotecas e ferramentas. Em seguida, você instala mais bibliotecas e aplicações, como o servidor Web NGINX ou o Microsoft SQL Server. Finalmente, você implanta o código da aplicação. Geralmente, essa VM executa uma grande parte, se não toda, da aplicação. É uma grande instalação de aplicação e instância em execução. Para melhorar a performance, você pode adicionar mais memória ou CPU à VM (dimensionamento vertical, discutido nos capítulos anteriores) ou aumentar o número de instâncias que executam a aplicação (escala horizontal, como em conjuntos de escalas). A criação de várias instâncias de aplicações só funciona se a aplicação fizer o reconhecimento de cluster, e geralmente isso envolve algum tipo de armazenamento compartilhado para permitir um estado consistente nas instâncias da aplicação. Essa forma tradicional de implantação é chamada de aplicação *monolítica*.

Uma abordagem diferente de como você projeta, desenvolve e executa aplicações é dividir as coisas em componentes menores e reduzidos. Essa é uma abordagem de *microsserviços* para desenvolvimento e implantação de aplicações. Cada microsserviço é responsável por uma pequena parte do ambiente de aplicação mais amplo. Os microsserviços podem crescer, escalar e ser atualizados independentemente do restante do ambiente da aplicação.

Embora esse modelo possa oferecer desafios no início, enquanto as equipes de desenvolvimento e de TI aprendem a adotar uma maneira diferente de criar e implantar aplicações, os contêineres são excelentes para a abordagem de microsserviço. Os desenvolvedores têm o poder de implantar atualizações menores e mais incrementais em um ritmo mais rápido do que a abordagem monolítica do desenvolvimento de aplicações. Os microsserviços e contêineres também são ótimos para fluxos de trabalho de integração contínua e entrega contínua (CI/CD), onde é possível criar, testar, preparar e implantar atualizações com mais facilidade. Seus clientes recebem novos recursos ou correções de bugs mais rapidamente do que de outra forma, e esperamos que sua empresa cresça como resultado disso.

Microsserviços com o Azure Service Fabric

Este capítulo se concentra em contêineres e orquestração do Docker com o Kubernetes, mas um serviço semelhante do Azure direciona o desenvolvimento de aplicações para um modelo de microsserviços. O Azure Service Fabric existe há vários anos e era, historicamente, uma abordagem centrada no Windows para criar aplicações em que cada componente era dividido em seu próprio microsserviço. O Service Fabric controla onde cada componente de microsserviço é executado em um cluster, permite que os serviços descubram e se comuniquem uns com os outros e lida com redundância e escala.

Muitos dos grandes serviços do Azure usam o Service Fabric em segundo plano, inclusive o Cosmos DB. Isso deve dar a você uma ideia de como o Service Fabric pode ser capaz e poderoso! O próprio Service Fabric é executado sobre conjuntos de escalas de máquinas virtuais. Você sabe uma coisa ou duas sobre conjuntos de escalas, certo?

A plataforma do Service Fabric está desenvolvida e agora pode gerenciar o Windows e o Linux como o sistema operacional convidado, para que você possa criar seu aplicativo com qualquer linguagem de programação com a qual esteja familiarizado. Veja outro exemplo de opção no Azure: você tem a flexibilidade de escolher como deseja gerenciar e orquestrar suas aplicações de contêiner. O Service Fabric e o Serviço Azure Kubernetes têm excelentes benefícios e casos de uso.

Como um bom ponto de partida, se atualmente você desenvolve ou gostaria de desenvolver microsserviços fora dos contêineres, o Service Fabric é uma ótima escolha. As aplicações criadas com o modelo de ator também são uma ótima opção, pois o Service Fabric foi originalmente criado com base nesse modelo de programação. O Service Fabric fornece uma abordagem unificada para lidar com aplicações de microsserviços mais tradicionais e aplicações baseadas em contêiner. Se você optar por adotar contêineres para outros workloads, poderá usar as mesmas ferramentas e interface de gerenciamento do Service Fabric para gerenciar todos os ambientes de aplicações.

Para uma abordagem de aplicação mais focada em contêineres desde o início, o Serviço Azure Kubernetes pode ser uma opção melhor, com o crescimento e a adesão do Kubernetes proporcionando uma experiência de contêiner de primeira classe. Você pode executar com contêineres Linux e Windows no AKS.

19.3 Instâncias de Contêiner do Azure

Agora que você entende um pouco mais sobre o que são os contêineres e como você pode usá-los, vamos nos aprofundar e criar uma instância básica da pizzaria. Esse é o mesmo exemplo dos capítulos anteriores, em que você criou uma VM básica que executava seu site ou implantou o aplicativo em aplicativos Web. Nos dois casos, você precisava criar a VM ou o aplicativo Web, conectar-se a ela e depois implantar uma página da Web básica nela. O poder dos contêineres pode tornar sua vida muito mais fácil? Com certeza!

Um serviço simples chamado Instâncias de Contêiner do Azure (ACI) permite criar e executar contêineres em questão de segundos. Não há recursos de rede iniciais para criar e configurar, e você paga por cada instância de contêiner por segundo. Se você nunca usou contêineres e não deseja instalar nada localmente em seu computador, a ACI é uma ótima maneira de experimentar a tecnologia.

Para ver como você pode operar sua pizzaria rapidamente, vamos criar uma instância de contêiner. É necessário apenas um comando para executar uma instância de contêiner, mas a figura 19.6 mostra como você reúne muitos componentes para que isso ocorra em segundo plano. Analisaremos os componentes de um Dockerfile e do Hub do Docker depois que a instância do contêiner estiver em funcionamento.

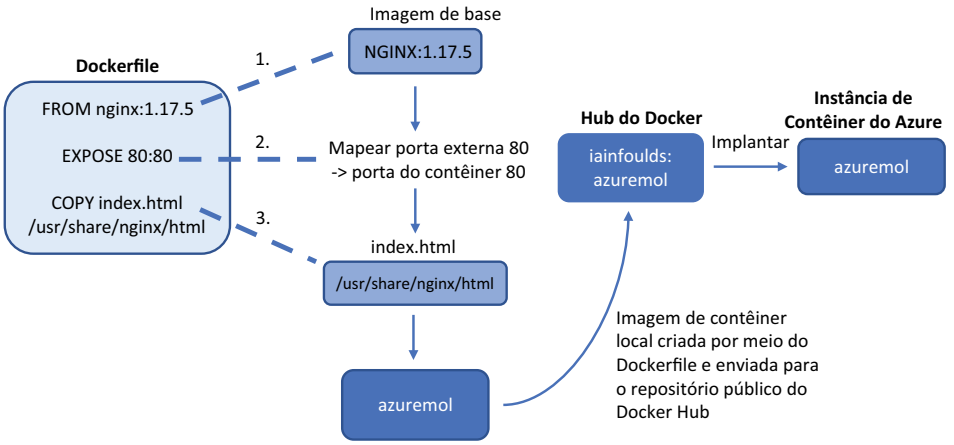


Figura 19.6 Um Dockerfile foi usado para criar uma imagem completa do contêiner, o azuremol. Essa imagem foi enviada por push para um registro público on-line chamado Hub do Docker. Agora você pode criar uma instância de contêiner usando essa imagem pública pré-criada do Hub do Docker, que fornece uma imagem de aplicação pronta para execução.

Experimente agora

Para criar uma instância de contêiner do Azure que execute um site básico, siga as etapas a seguir:

1. Abra o portal do Azure e escolha o ícone do Cloud Shell no menu superior.
2. Crie uma instância de contêiner e especifique que você deseja ter um endereço IP público e abrir a porta 80:

```
az container create \
  --resource-group azuremolchapter19 \
  --name azuremol \
  --image iainfoulds/azuremol \
  --ip-address public \
  --ports 80
```

Este exercício usa uma imagem de exemplo que criei e que examinaremos um pouco mais quando o contêiner estiver em execução.

3 Para ver o que foi criado, observe a saída do comando para criar o contêiner.

Na seção Eventos, você pode ver como a imagem é obtida (por download) do Hub do Docker, um contêiner será criado e após será iniciado.

Algumas reservas de CPU e memória também são atribuídas, que podem ser ajustadas, se necessário. Um endereço IP público é mostrado, junto com algumas informações no contêiner, como o estado de provisionamento, o tipo de sistema operacional e a política de reinicialização.

4 Para abrir o site básico que é executado no contêiner, você pode consultar apenas o endereço IP público atribuído:

```
az container show \
  --resource-group azuremolchapter19 \
  --name azuremol \
  --query ipAddress.ip \
  --output tsv
```

5 Abra o endereço IP público de sua instância de contêiner em um navegador da Web. A pizzaria básica deverá ser exibida, como mostra a figura 19.7.

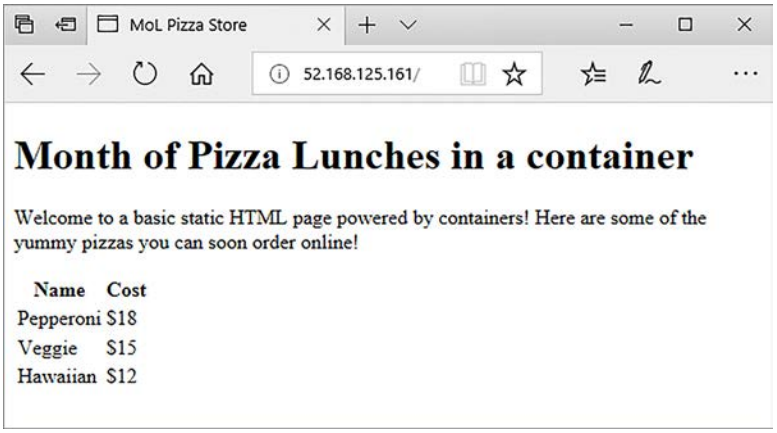


Figura 19.7 Quando você criar uma instância de contêiner, o site da pizzaria será executado sem qualquer configuração adicional. Toda a configuração e o conteúdo estão incluídos na imagem do contêiner. Esse exercício rápido destaca a portabilidade e o poder dos contêineres – quando a imagem do contêiner for preparada, seu aplicativo será instalado e executado assim que uma nova instância de contêiner for implantada.

Vamos examinar a imagem do contêiner. Não quero abordar os detalhes do Docker e de como criar imagens de contêiner, mas é importante entender de onde essa imagem veio e como ela é executada no site sem nenhuma configuração adicional.

A imagem é criada a partir de uma definição de configuração chamada *Dockerfile*. Em um Dockerfile, você define qual é a plataforma base, a configuração que deseja aplicar

e os comandos a serem executados ou arquivos a serem copiados. Os Dockerfiles podem ser, e geralmente são, mais complexos que o exemplo a seguir, que foi usado para criar o exemplo de contêiner azuremol:

```
FROM nginx:1.17.5

EXPOSE 80:80

COPY index.html /usr/share/nginx/html
```

Quando esse Dockerfile foi usado para criar uma imagem de contêiner do Docker, o NGINX foi usado como a imagem de origem e o exemplo de página da Web foi copiado nele. Esse contêiner foi enviado por push ao Hub do Docker, um repositório público on-line que o Docker fornece para compartilhar e implantar contêineres. Para implantar a instância do contêiner, você forneceu iainfoulds/azuremol como a imagem do contêiner a ser usada. O Azure procurou no Hub do Docker e encontrou um repositório chamado iainfoulds e, dentro dele, uma imagem chamada azuremol.

Vamos examinar cada linha do Dockerfile:

- `FROM nginx:1.17.5`: Nos capítulos anteriores, você criou uma VM básica, conectou-se a ela com SSH e, em seguida, instalou manualmente o servidor Web NGINX. No exemplo do Dockerfile, tudo isso é realizado em uma linha. Essa linha informa que o contêiner deve ser baseado em uma imagem de contêiner existente pré-instalada com o NGINX. A `1.17.5` é a versão da imagem de contêiner NGINX público a ser usada. Ela é a mais recente no momento da redação. É uma prática recomendada incluir o número de uma versão específica. Se você não incluir um número de versão, a mais recente será sempre usada. Isso soa bem na teoria, mas as aplicações de microsserviços podem ir para um número muito grande de contêineres ativos. Portanto, para garantir que você tenha um ambiente consistente, controle o número exato da versão de cada componente em uso.
- `EXPOSE 80:80` – Para permitir o acesso à VM nos capítulos anteriores, você criou uma regra NSG que permitia a porta 80. No Dockerfile, essa linha informa ao contêiner para abrir a porta 80 e mapeá-la para a porta interna 80. Quando você criou sua instância de contêiner com az container create, também especificou que a plataforma do Azure deve permitir tráfego com –porta 80. Essa é toda a rede virtual em que você precisa pensar!
- `COPY index.html /usr/share/nginx/html` – A parte final é obter sua aplicação no contêiner. Nos capítulos anteriores, você usou o Git para obter a página da Web da pizzaria de exemplo e enviá-la por push para o aplicativo Web. Com o Dockerfile, `COPY (copie)` o arquivo index.html no diretório local /usr/share/nginx/html no contêiner. Pronto!

Para seus próprios cenários, você pode definir um Dockerfile que usa uma imagem base diferente, como Node.js ou Python. Em seguida, instale quaisquer bibliotecas ou pacotes de suporte adicionais necessários, obtenha o código da aplicação no controle de origem, como o GitHub, e implante sua aplicação. Esse Dockerfile seria usado para criar imagens de contêiner que serão armazenadas em um registro de contêiner privado, não em um repositório público do Hub do Docker como aquele no exemplo.

Registro de Contêiner do Azure

Você pode pensar que o Docker Hub parece ótimo. O Azure tem algo tão incrível? Sim! Como você precisa criar um arquivo Docker e uma imagem de contêiner, infelizmente não é um exercício de rápido, e há muito a ser coberto neste capítulo. Você pode facilmente integrar o Registro de Contêiner do Azure (ACR) e AKs para que os dois serviços funcionem bem juntos. Você pode criar suas próprias imagens a partir de um Dockerfile no Cloud Shell, e eu o encorajo a explorar isso se tiver tempo. O Registro de Contêiner do Azure (ACR) é a opção que escolho para armazenar minhas imagens de contêiner, por alguns motivos:

- Ele é um registro privado para suas imagens de contêiner, portanto, você não precisa se preocupar com o acesso potencial indesejado a seus arquivos e configurações da aplicação. Você pode aplicar os mesmos mecanismos de RBAC que abordamos no capítulo 6. O RBAC ajuda a limitar e auditar quem tem acesso às suas imagens.
- Armazenar suas imagens de contêiner em um registro no Azure significa que suas imagens estão bem nos mesmos data centers da infraestrutura usada para executar suas instâncias de contêiner ou clusters (que veremos na seção 19.4.1). Embora as imagens do contêiner devam ser relativamente pequenas, geralmente com apenas dezenas de MB, isso pode aumentar se você continuar fazendo download dessas imagens em um registro remoto.

O ACR também oferece opções de replicação e redundância incorporadas que você pode usar para colocar seus contêineres perto de onde você os implanta e executá-los para que os usuários tenham acesso. Essa localidade de região é semelhante à maneira como você usou a replicação global do Cosmos DB no capítulo 10 para fazer com que esses milissegundos fossem contabilizados e fornecer aos seus clientes o tempo de acesso mais rápido possível às suas aplicações.

Se tudo isso parecer interessante, veja que o ACR começará a funcionar rapidamente com seu próprio repositório privado em alguns minutos: http://mng.bz/04rj.

19.4 Serviço de Kubernetes do Azure

A execução de uma única instância de contêiner é ótima, mas isso não significa muita redundância ou capacidade de escalar. Lembra-se de como usamos capítulos inteiros no início do livro para falar sobre como executar várias instâncias da aplicação, balancear a carga e escalá-las automaticamente? Não seria ótimo fazer o mesmo com os contêineres? É nesse caso que você precisa de um orquestrador de contêineres.

Como o nome indica, um *orquestrador de contêineres* gerencia suas instâncias de contêiner, monitora sua integridade e pode escalar conforme necessário. Os orquestradores podem lidar com muito mais do que isso (e geralmente conseguem), mas em um nível alto, um foco principal é manipular todas as partes móveis envolvidas na execução de uma aplicação baseada em contêiner altamente disponível e escalonável. Existem alguns orquestradores de contêineres, como o Docker Swarm e o Sistema Operacional de Nuvem Distribuída (DC/OS), mas um se destacou em relação aos demais para se tornar a escolha principal de orquestrador – o Kubernetes.

O Kubernetes começou como um projeto open source liderado e patrocinado pelo Google que nasceu das ferramentas internas de orquestração de contêineres da empresa. Amplamente aceito pela comunidade Open Source, o Kubernetes é um dos

maiores e mais crescentes projetos open source do GitHub. Muitas grandes empresas de tecnologia, inclusive Red Hat, IBM e Microsoft, contribuem para o projeto central do Kubernetes.

Nesta seção, usaremos o mesmo exemplo de aplicativo Web do exercício anterior com a ACI para executar uma implantação redundante e escalonável no Kubernetes. Você terminará com alguns componentes, como mostra a figura 19.8.

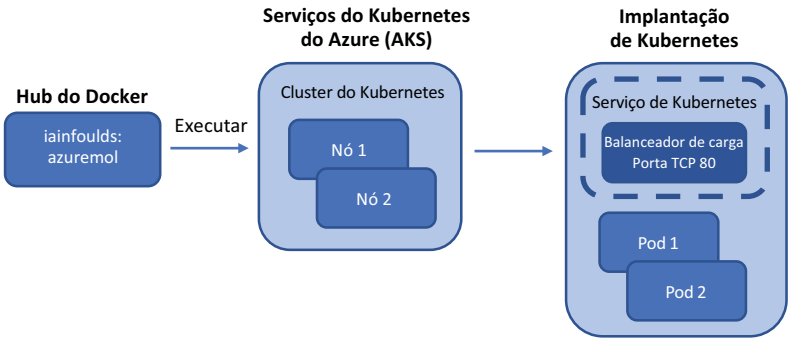


Figura 19.8 Seu exemplo de contêiner do Hub do Docker é executado em um cluster do Kubernetes de dois nós que você cria no Serviço Azure Kubernetes. A implantação do Kubernetes contém dois pods lógicos, um em cada nó do cluster, com uma instância de contêiner em execução dentro de cada pod. Em seguida, você expõe um balanceador de carga público para permitir que seu aplicativo Web seja visualizado on-line.

19.4.1 Criar um cluster com os Serviços Azure Kubernetes

No capítulo 9, examinaremos como os conjuntos de escalas de máquinas virtuais reduzem a complexidade de implantar e configurar a infraestrutura subjacente. Você informa quantas instâncias de VM deseja em um conjunto de escalas e o restante da rede, armazenamento e configuração será implantado para você. O AKS funciona da mesma maneira para oferecer um cluster do Kubernetes resiliente e escalonável, com gerenciamento controlado pela plataforma do Azure. Os conjuntos de escalas podem ser usados para as VMs subjacentes que são executadas no cluster do AKS, e essas VMs podem ser distribuídas entre zonas de disponibilidade. Os balanceadores de carga do Azure, também redundantes na zona, são usados. O AKS reúne vários dos componentes de infraestrutura e práticas recomendadas que você aprendeu até agora neste livro.

Experimente agora

Para visualizar as informações em seu cluster do AKS, conclua as etapas a seguir:

1. Abra o portal do Azure e escolha o ícone do Cloud Shell no menu superior.
2. No início do capítulo, você criou um cluster do Kubernetes. O processo levou alguns minutos, mas esperamos que esteja pronto agora! Observe o status do cluster da seguinte maneira:

```
az aks show \
  --resource-group azuremolchapter19 \
  --name azuremol
```

`provisioningState` perto do fim deve reportar `Succeeded`.

3 Se o cluster estiver pronto, obtenha um arquivo de credenciais que permita usar as ferramentas de linha de comando do Kubernetes para autenticar e gerenciar recursos:

```
az aks get-credentials \
  --resource-group azuremolchapter19 \
  --name azuremol
```

Isso é tudo o que é preciso para colocar o Kubernetes em funcionamento no Azure! Você pode estar se perguntando: "Não é possível criar meu próprio cluster com VMs ou conjuntos de escalas e instalar manualmente os mesmos componentes do Docker e do Kubernetes?" O paralelo é a abordagem de IaaS e PaaS de VMs versus aplicativos Web. A abordagem do aplicativo Web oferece muitos benefícios: você só se preocupa com as opções de configuração de alto nível e depois faz o upload do código da sua aplicação. Um cluster gerenciado do Kubernetes, oferecido pela AKS, reduz o nível de complexidade e gerenciamento; seu foco se torna suas aplicações e a experiência de seus clientes.

Da mesma forma que você pode escolher VMs em aplicativos Web, pode optar por implantar seu próprio cluster do Kubernetes em vez de usar o AKS. Isso é bom, as duas abordagens acabam usando os mesmos componentes de serviços do Azure. VMs, conjuntos de escalas, balanceadores de carga e NSGs são tópicos sobre os quais você aprendeu nos capítulos anteriores, e todos ainda fazem parte do tópico de clusters do AKS, embora estejam separados. Do ponto de vista de planejamento e solução de problemas, você deve ter as habilidades necessárias para entender o que está acontecendo em segundo plano para fazer as ofertas gerenciadas do Kubernetes funcionarem. Informações como seu nível de conforto e quanto tempo você deseja gastar gerenciando a infraestrutura ajudarão a guiar seu processo de tomada de decisão à medida que você cria um nova aplicação em torno de contêineres no Azure.

19.4.2 Executar um site básico no Kubernetes

Você criou um cluster do Kubernetes na seção 19.4.1, mas não há nenhuma aplicação em execução. Vamos mudar isso! Agora você precisa criar a implantação do Kubernetes que você viu anteriormente na figura 19.8. Veja a figura 19.9.

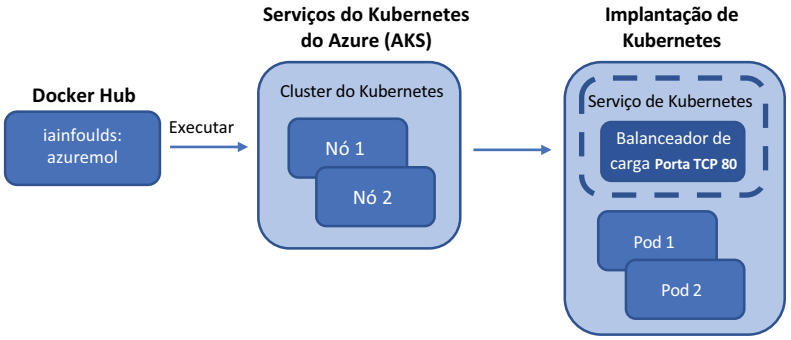


Figura 19.9 Com o cluster do Kubernetes criado no AKS, você pode criar uma implantação do Kubernetes e executar o aplicativo. Seu contêiner é executado nos dois nós, com um pod lógico em cada nó; você precisa criar um serviço do Kubernetes que exponha um balanceador de carga público para rotear o tráfego para seu aplicativo.

Experimente agora

Para implantar uma aplicação em seu cluster do Kubernetes, siga estas etapas:

1. Você interage com um cluster do Kubernetes usando um utilitário de linha de comando chamado `kubectl`. Use a mesma imagem do contêiner iainfoulds/azuremol do Hub do Docker que você executou como uma instância de contêiner:

   ```
   kubectl run azuremol \
     --image=docker.io/iainfoulds/azuremol:latest \
     --port=80
   ```

   Pode demorar um minuto para fazer download da imagem do contêiner do Hub do Docker e iniciar a aplicação no Kubernetes. A aplicação é executada em um *pod*: uma construção lógica no Kubernetes que abriga cada contêiner.

2. Os pods podem conter componentes auxiliares adicionais, mas, por enquanto, monitoram o status do seu contêiner observando o pod:

   ```
   kubectl get pods --watch
   ```

Mesmo quando o status dos relatórios do pod for `Em execução`, você não poderá acessar sua aplicação. A instância do contêiner que você criou anteriormente poderia rotear o tráfego sobre um endereço IP público diretamente para essa instância, mas o que você acha que é necessário para um cluster do Kubernetes rotear o tráfego para contêineres? Se disse: "um balanceador de carga". Parabéns! No momento, você tem apenas um pod: uma única instância de contêiner. Você expande o número de pods no laboratório no final do capítulo e, para que isso funcione, precisa de uma maneira de direcionar o tráfego para várias instâncias. Então, vamos dizer ao Kubernetes para usar um balanceador de carga.

Esse é o ponto onde a integração entre o Kubernetes e o Azure se torna interessante. Quando você informa ao Kubernetes que deseja criar um balanceador de carga para seus contêineres, em segundo plano, o Kubernetes entra novamente na plataforma do Azure e cria um balanceador de carga do Azure. Esse balanceador de carga do Azure é como o que você aprendeu no capítulo 8. Há pools de IP front-end e de back-end e regras de balanceamento de carga, e é possível configurar sondas de integridade. À medida que sua implantação do Kubernetes aumenta ou reduz, o balanceador de carga é atualizado automaticamente conforme necessário.

Experimente agora

Para expor sua aplicação à Internet, siga estas etapas:

1. Diga ao Kubernetes que você deseja usar um balanceador de carga e adicione uma regra para distribuir o tráfego na porta 80:

   ```
   kubectl expose deployment/azuremol \
     --type="LoadBalancer" \
     --port 80
   ```

2 Como antes, observe o status da sua implantação de serviço:

```
kubectl get service azuremol --watch
```

Quando o endereço IP público externo for atribuído, significa que o balanceador de carga do Azure terminou de implantar, e o cluster e os nós do Kubernetes estão conectados.

3 Abra o endereço IP público do seu serviço em um navegador da Web para ver seu aplicativo Web em execução.

Geralmente, as implantações de aplicações no Kubernetes são muito mais envolvidas do que esse exemplo básico. Você normalmente define um manifesto de serviço, semelhante a um modelo do Resource Manager, que define todas as características da sua aplicação. Essas propriedades podem incluir o número de instâncias da aplicação a ser executado, qualquer armazenamento a ser anexado, métodos de balanceamento de carga e portas de rede a serem usadas e assim por diante. No mundo real, você nem faz isso manualmente; um sistema CI/CD como o Azure DevOps ou o Jenkins automatiza as implantações de aplicações e serviços diretamente no cluster do AKS. O que é ótimo sobre o AKS é que você não precisa se preocupar com a instalação e a configuração do Kubernetes. Assim como com outros serviços de PaaS, como aplicativos Web e o Cosmos DB, você traz suas aplicações e permite que a plataforma do Azure cuide da infraestrutura e da redundância subjacentes.

Mantenha-o limpo e organizado

Lembre-se de limpar e excluir seus grupos de recursos para que você não acabe consumindo muitos dos seus créditos gratuitos do Azure. À medida que você começa a explorar contêineres, torna-se ainda mais importante prestar atenção aos recursos do Azure que você deixa ativados. Um único aplicativo Web não custa muito, mas um cluster do AKS de cinco nós e algumas instâncias do contêiner com imagens do Azure Container Registry georeplicadas certamente custa!

As instâncias da ACI são cobradas por segundo e o custo aumenta rapidamente se elas forem executadas por dias ou semanas. Um cluster do AKS executa uma VM para cada nó, portanto, se você aumentar e executar muitas VMs em seu cluster, estará pagando por uma VM para cada nó.

Não há cobrança pelo número de contêineres que cada um desses nós do AKS executa, mas, como ocorre com qualquer VM, um nó do AKS fica caro quando deixado em execução. O que é ótimo sobre o Kubernetes é que você pode exportar suas configurações de serviço (a definição para seus pods, balanceadores de carga, escalonamento automático e assim por diante) para implantá-las em outro lugar. À medida que você cria e testa suas aplicações, não precisa deixar um cluster do AKS em execução. Você pode implantar um cluster conforme necessário e implantar seu serviço a partir de uma configuração anterior.

Os clusters do AKS podem ser expandidos e reduzidos, como você verá no exercício de laboratório no fim do capítulo. Você também pode configurar o dimensionamento automático, que faz essa escala para você dependendo da carga. É o mesmo tipo de dimensionamento automático que analisamos no capítulo 9 para conjuntos de escalas e aplicativos Web. Você está começando a ver tudo se encaixando no Azure?

Este capítulo foi uma introdução rápida para contêineres e Kubernetes, então não se preocupe se você se sentir um pouco sobrecarregado agora! A Manning tem vários ótimos livros, como *Aprenda a usar o Docker em um mês de aulas*, por Elton Stoneman (https://livebook.manning.com/book/learn-docker-in-a-month-of-lunches) e *Kubernetes em*

(continuação)

ação, por Marko Luksa (https://livebook.manning.com/book/kubernetes-in-action), que podem ajudar você a se aprofundar ainda mais o Docker, no desenvolvimento de aplicações de microsserviços e no Kubernetes. Confira esses livros se esse capítulo parecer interessante e você quiser explorar mais!

Nos exemplos deste capítulo, usamos VMs Linux para os nós de cluster do AKS e, depois, executamos contêineres do Linux para NGINX. Os contêineres ficam um pouco complicados, pois você pode executar contêineres Linux somente em nós do Linux, por exemplo. Como você aprendeu no início do capítulo, os contêineres compartilham o sistema operacional convidado e o kernel. Então, não é possível executar contêineres do Windows em um nó do Linux. Em geral, também não é possível executar contêineres do Linux em um nó do Windows. Há alguns truques técnicos interessantes, mas, em geral, o contêiner e o sistema operacional de nó subjacente devem corresponder.

O que é ótimo no AKS é que você pode executar nós do Linux e do Windows para executar contêineres do Linux e do Windows. Você precisa prestar um pouco de atenção em como esses diferentes contêineres são programados nos diferentes sistemas operacionais de nó, mas essa abordagem expande muito as aplicações e os serviços que você pode executar no AKS.

19.5 Laboratório: escalar suas implantações do Kubernetes

O exemplo básico nesse capítulo criou um cluster do Kubernetes de dois nós e um único pod que executa o site. Neste laboratório, explore como você pode escalar o cluster e o número de instâncias de contêiner. Esse exemplo é básico, mas quanto mais nós você tiver, mais instâncias de contêiner poderá executar, o que será especialmente útil quanto mais aplicações você precisar executar no cluster.

1. Você pode ver quantos nós estão em seu cluster do Kubernetes com `kubectl get nodes`. Aumente seu cluster para três nós:

   ```
   az aks scale \
     --resource-group azuremolchapter19 \
     --name azuremol \
     --node-count 3
   ```

   Leva um ou dois minutos para aumentar e adicionar o novo nó.

2. Use `kubectl` novamente para ver o status dos nós. Ao aumentar um nó, o Kubernetes não cria instâncias de contêiner adicionais para suas aplicações automaticamente. Portanto, você não obtém imediatamente nenhum benefício com os recursos de computação adicionais que o novo nó fornece.

3. Veja sua implantação atual com `kubectl get deployment azuremol`. Apenas uma instância foi criada anteriormente. Este exemplo de aplicação não está aproveitando ao máximo o novo nó que você adicionou ao cluster na etapa 1. Aumente até cinco instâncias ou *réplicas*:

   ```
   kubectl scale deployment azuremol --replicas 5
   ```

4 Use `kubectl` novamente para examinar a implantação. Veja os pods, as instâncias de contêiner em execução, com `kubectl get pods`. Em questão de segundos, todas essas réplicas adicionais foram iniciadas e conectadas ao balanceador de carga.

5 Use `kubectl get pods -o wide` para ver em quais nós os pods são executados. Veja o último número no nome do nó, que indica qual nó no conjunto de escalas é usado. Os pods devem ser distribuídos entre todos os nós do cluster. Como outras aplicações aumentariam o número de contêineres de forma semelhante, você pode começar a maximizar o uso dos recursos de computação em todos os nós no cluster.

20 Azure e a Internet das Coisas

Para mim, uma das áreas mais interessantes da tecnologia nos últimos anos é a Internet das coisas (IoT). Eu não acredito que uma máquina de lavar louça ou uma geladeira precisa estar conectada à internet ainda, e há preocupações de privacidade válidas sobre uma TV ou dispositivo de áudio que está permanentemente conectado à internet sempre ouvindo o som da sua voz para emitir um comando. Existem, no entanto, muitas aplicações práticas para dispositivos IoT. Você pode ter relatórios de equipamentos de fabricação em seu status de integridade, gerar alertas de manutenção e permitir que os operadores entendam sua eficiência em várias fábricas em todo o mundo. Uma empresa de camionagem poderia transmitir a telemetria de seus veículos sobre as cargas transportadas e os tempos médios de condução, além de conseguir redirecionar de forma mais inteligente os motoristas conforme necessário. As transportadoras podem rastrear cada contêiner e ajudar seus clientes a gerenciar melhor sua cadeia de suprimentos, sabendo onde estão seus recursos.

No Azure, você pode integrar dispositivos IoT com uma variedade de serviços. Os Aplicativos Web do Azure podem fornecer um front-end para os dados serem visualizados, o armazenamento pode ser usado para registrar dados transmitidos de dispositivos, e os recursos sem servidor, como os Aplicativos Lógicos do Azure (abordados no próximo e último capítulo), podem processar os dados recebidos.

Neste capítulo, examinaremos o que é a IoT e como usar o Hub IoT do Azure para gerenciar e coletar dados de dispositivos centralmente. Em seguida, você verá como usar um aplicativo Web do Azure para exibir dados em tempo real de um dispositivo IoT.

20.1 O que é a Internet das Coisas?

O interesse na IoT cresceu consideravelmente nos últimos anos, mas é um termo vago que pode ser aplicado a muitos cenários diferentes. Em um nível básico, a IoT é uma abordagem em que muitos dispositivos interconectados – geralmente dispositivos

eletrônicos pequenos e de baixo custo – conectam-se aos sistemas e aplicações centrais. Geralmente, os dispositivos conectados relatam informações coletadas de sensores ou entradas conectados. Então, essas informações poderão ser processadas por um sistema central – talvez com IA ou ML, conforme discutido no capítulo 17 – e realizar ações apropriadas. A Figura 20.1 mostra uma abordagem de alto nível para a IoT.

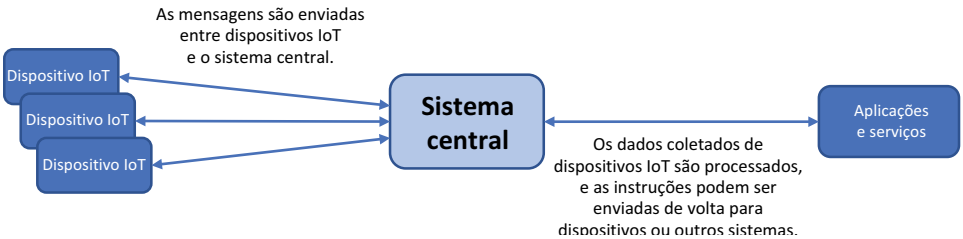


Figura 20.1 As mensagens são enviadas entre muitos dispositivos IoT conectados e um sistema central. Seus serviços e aplicações poderão processar os dados recebidos e enviar instruções do dispositivo para executar ações adicionais em resposta aos dados coletados.

Entre os exemplos de IoT em ação estão:

- *Garagem de estacionamento*: um pequeno sensor acima de cada baia de estacionamento *detecta se um veículo está* estacionado lá. Uma luz acima de cada baia poderá iluminar-se na cor verde se a baia de estacionamento estiver vazia ou na cor vermelha, se estiver ocupada. Os motoristas que entram na garagem de estacionamento podem ver quadros informativos em tempo real em cada andar, informando quantas vagas disponíveis existem. As luzes vermelha e verde acima de cada baia ajudam os motoristas a determinar rapidamente a localização dos pontos disponíveis enquanto dirigem por cada corredor.
- *Fábrica*: as máquinas na fábrica podem reportar informações sobre a saída operacional, os níveis de consumíveis e as necessidades de manutenção. Um sistema central poderá agendar um técnico de manutenção para reparar proativamente o equipamento ou reabastecer os consumíveis, o que reduz qualquer tempo de inatividade na linha de produção. Quando combinado com IA e ML, os cronogramas de manutenção podem ser previstos, e a quantidade correta de suprimentos ou matérias-primas pode ser entregue pouco antes de serem necessários na produção.
- *Transporte*: os ônibus ou trens de transporte público podem incluir sensores de GPS que relatam a localização e a velocidade. Informações sobre emissão de bilhetes podem ser coletadas para informar sobre quantas pessoas estão sendo transportadas. Placas de informação para passageiros em uma estação de trem ou terminal de ônibus podem fornecer informações em tempo real sobre quando cada veículo chegará. Quando essa tecnologia é combinada com IA e ML, os passageiros em espera podem receber sugestões de rotas alternativas com base nas condições de tráfego, atrasos ou grande quantidade de passageiros.

Geralmente, a IoT funciona junto com outros serviços e aplicações. Os cenários de fábrica e transporte poderiam usar IA e ML para melhor informar a decisão de produção ou dar sugestões aos passageiros. Os aplicativos Web podem usar informações recebidas de dispositivos IoT para fornecer acesso a partir de dispositivos móveis ou gerar alertas e notificações. Os dados recebidos de dispositivos IoT podem ser registrados em um sistema de banco de dados, como o Azure Cosmos DB, que é processado por aplicações de business intelligence e gera relatórios.

Ideias com maiores perspectivas de futuro em torno da IoT incluem coisas como a sua geladeira detectar os níveis de alimentos e gerar uma lista de compras ou até mesmo pedir comida de um supermercado local. Seu carro pode relatar os dados à concessionária, o que pode fazer com que todas as peças ou consumíveis necessários estejam prontos quando você levar o veículo para o serviço. E se, quando o despertador tocar para acordá-lo de manhã, a cafeteira ligar e se preparar para o café da manhã?

Uma grande área de preocupação com a IoT é a segurança do dispositivo. Com tantos dispositivos fora de sua infraestrutura de rede principal e muitas vezes conectados à Internet pública, poder provisionar, manter e atualizar esses dispositivos é um desafio. Muitos dispositivos IoT são de baixa potência, simples componentes eletrônicos que podem não ter os recursos de armazenamento ou processamento para atualizar-se com atualizações de segurança e de aplicações da mesma forma que um desktop ou notebook tradicional. Não é suficiente implantar vários dispositivos de IoT, especialmente dispositivos no nível do consumidor, sem um plano para protegê-los adequadamente e fornecer atualizações e manutenção.

Essas preocupações de segurança não devem impedir você de criar aplicações e serviços que usam dispositivos IoT. A IoT traz um novo conjunto de desafios para a manutenção de dispositivos tradicionais, mas existem soluções que permitem provisionar e manter dispositivos centralmente e proteger a comunicação do dispositivo.

A essa altura, tenho certeza de que você pode ter adivinhado que o Azure tem essa solução de IoT! Ele oferece um conjunto de serviços de IoT. Vamos ver como você pode começar a explorar a IoT com o Azure.

Acelerar suas implantações da IoT do Azure

Este capítulo se concentra no Hub IoT do Azure, um serviço que permite provisionar e conectar dispositivos IoT para criar suas próprias soluções. Você pode definir como esses dispositivos IoT se conectam, quais usuários ou aplicações podem acessar seus dados e proteger a conectividade. Como criar e implantar a infraestrutura de aplicações para conectar tudo junto é com você.

Os aceleradores de solução IoT do Azure são cenários principais pré-desenvolvidos, como o monitoramento remoto de dispositivos ou uma fábrica conectada. Os aceleradores implantam serviços comuns do Azure, como Hub IoT, Aplicativos Web, Cosmos DB e Armazenamento, e executam uma aplicação de exemplo que integra todos esses serviços diferentes.

Você ainda precisa personalizar a aplicação para seu próprio ambiente, os dispositivos de IoT em uso e os dados a serem coletados e monitorados, mas os aceleradores de solução de IoT oferecem uma excelente estrutura para começar. Enquanto o Hub IoT cria uma maneira de você conectar dispositivos IoT ao Azure e permite implantar serviços adicionais dos quais você precisa, os aceleradores de solução de IoT implantam soluções pré-criadas que usam os serviços mais comuns do Azure que você usaria.

Se você virar fã da IoT após esse capítulo e quiser saber mais, os aceleradores de solução IoT do Azure são uma ótima maneira de ver as possibilidades do que o Azure pode oferecer. Como falamos ao longo desse livro, o Azure é muito mais do que apenas um ou dois serviços independentes. Você pode implantar muitos serviços juntos para fornecer a melhor experiência de aplicativo possível para seus clientes.

20.2 Gerenciar centralmente os dispositivos com o Hub IoT do Azure

O Hub IoT do Azure permite gerenciar, atualizar e transmitir dados centralmente de dispositivos IoT. Com esse serviço, você pode executar ações como configurar rotas de aplicações para dados recebidos de dispositivos, provisionar e gerenciar certificados para proteger a comunicação e monitorar a integridade com diagnósticos e métricas do Azure. Você pode conectar seus dispositivos IoT a outros serviços e aplicações do Azure para permitir que eles enviem e recebam dados como parte de uma solução mais ampla. Como acontece com todas as coisas no Azure, o acesso pode ser controlado com RBAC, e os dados de diagnóstico podem ser coletados centralmente para solução de problemas e monitoramento ou alertas. A figura 20.2 descreve como um Hub IoT atua como o local central para dispositivos IoT se conectarem aos serviços e aplicações mais amplos do Azure.

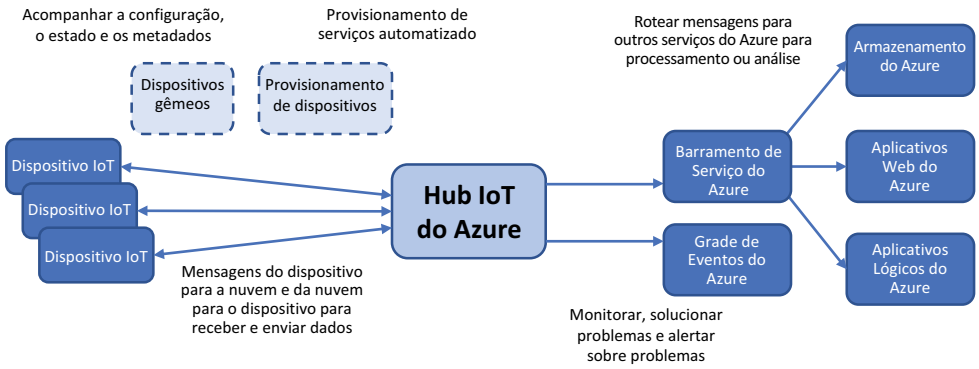


Figura 20.2 Com um hub IoT, você pode provisionar e gerenciar centralmente muitos dispositivos IoT em escala. A comunicação bidirecional existe entre dispositivos e o Azure para ler e gravar dados. Você pode processar dados recebidos de dispositivos e encaminhá-los para outros serviços do Azure, como Aplicativos Web e Armazenamento. Para monitorar e solucionar problemas, você pode rotear informações para a Grade de Eventos do Azure, que veremos no capítulo 21, e depois vincular a outras soluções de monitoramento.

Você controla o acesso a um hub IoT com políticas de acesso compartilhado. Essas políticas são como contas de usuário e permissões. Existem políticas padrão que permitem que dispositivos e serviços se conectem ao hub IoT ou que leiam e gravem informações do registro do dispositivo que rastreia dispositivos de IoT conectados e chaves de segurança. Cada política pode receber uma ou mais das seguintes permissões:

- Leitura de registro
- Gravação de registro

- Conexão de serviço
- Conexão do dispositivo

Chaves de acesso compartilhado são usadas por aplicações e serviços para se conectar a um hub IoT. Como no armazenamento, abordado no capítulo 4, as chaves de acesso compartilhado permitem que você defina strings de conexão para identificar o host, a política de acesso e a chave de acesso. Uma cadeia de conexão combina a chave de acesso, o tipo de política de acesso e o nome do host do hub IoT. Veja um exemplo de string de conexão de hub IoT:

```
HostName=azuremol.azure-devices.net;SharedAccessKeyName=registryRead;
➥SharedAccessKey=6be2mXBVN9B+UkoPUMuwVDtR+7NZVBq+C7A1xCmQGAb=
```

Existem chaves primárias e secundárias, que podem ser giradas e atualizadas para fins de segurança, assim como atualizar regularmente as senhas. Soluções como o Azure Key Vault, abordadas no capítulo 15, são ótimas maneiras de rastrear e armazenar essas chaves para as aplicações obterem quando necessário. Essa abordagem ao gerenciamento de chaves indica que você pode girar chaves de acesso com frequência sem precisar atualizar também todo o código da aplicação.

Os certificados digitais podem ser armazenados em um hub IoT e provisionados automaticamente para dispositivos IoT. Lembre-se de que os dispositivos IoT geralmente estão fora de sua infraestrutura principal e podem se conectar diretamente pela Internet sem qualquer forma de conexão de rede segura, como uma VPN. Verifique se todos os dados entre seus dispositivos e o Hub IoT estão criptografados usando conexões SSL/TLS. O Azure Key Vault pode gerar e armazenar certificados SSL que são adicionados ao hub IoT. Ou você pode usar uma autoridade de certificação existente para solicitar e emitir certificados. O importante é garantir que toda a comunicação entre seus dispositivos IoT e o Azure seja criptografada. Caso contrário, é provável que você receba um erro.

As rotas de hub IoT permitem enviar dados de dispositivos IoT para outros serviços do Azure. Você pode definir critérios, como o conteúdo da mensagem que contém uma determinada palavra-chave ou valor, e rotear as mensagens para serem armazenadas no Armazenamento do Azure ou processadas por um aplicativo Web. Em um dos exercícios a seguir, você simulará um sensor de temperatura básico conectado a um dispositivo IoT. Você pode definir uma rota no hub IoT para observar os dados recebidos e, se a temperatura registrada ultrapassar os 37,7° C, pode rotear os dados para um aplicativo lógico para enviar um alerta por email. Abordaremos o maravilhoso mundo da computação sem servidor e dos aplicativos lógicos no capítulo 21.

Trabalhando com o Azure IoT Edge

Neste capítulo, focaremos no Hub IoT do Azure. Outro serviço, o Azure IoT Edge, permite executar alguns serviços, como o Azure Functions e o Stream Analytics, em seu ambiente local. Em vez de todos os dispositivos IoT transmitirem dados para serem processados centralmente no Azure, você pode processar os dados em cada local.

O Azure IoT Edge executa aplicações e serviços em contêineres (isso foi abordado no capítulo 19). O uso de contêineres permite que o IoT Edge seja portátil e consistente na forma como funciona em diferentes dispositivos e ambientes. Serviços pré-criados do Azure podem ser implantados ou você pode criar suas próprias aplicações e distribuí-las para os locais de borda.

O principal benefício do IoT Edge é que você descarrega parte do processamento de dados e das transferências de dados da rede. Se você puder processar dados localmente no IoT Edge, poderá agrupar grandes blocos de dados e transmiti-los de volta ao Azure. Então, aplicações centrais podem agregar informações de outros locais do Edge para serem processadas por serviços como IA e ML.

Outro grande cenário para o Azure IoT Edge são os locais remotos, frequentemente encontrados nas indústrias de gás ou transporte de petróleo, em que a conectividade com a Internet pode não ser confiável o suficiente para que todos os dados do dispositivo IoT sejam transmitidos para o Azure para processamento. O IoT Edge permite que esses locais remotos continuem a operar com alguma autonomia, mesmo quando não há conexão com a Internet.

Ao planejar uma infraestrutura de aplicações que envolva dispositivos IoT, examine como você lida com interrupções de rede e conexões insatisfatórias com a Internet. Se o seu ambiente depende da Internet, planeje conexões de Internet redundantes e equipamentos para rotear os dados. Ou veja o IoT Edge para processar dados localmente quando não for possível fazer isso centralmente no Azure.

Experimente agora

Para começar a usar o IoT e criar um hub IoT, conclua as etapas a seguir:

1. Abra o portal do Azure. Lance o Cloud Shell e crie um grupo de recursos, como `azuremolchapter20`:

   ```
   az group create --name azuremolchapter20 --location eastus
   ```

2. Você trabalhou muito com a CLI do Azure neste livro porque os comandos do Cloud Shell e da CLI permitem a criação e o gerenciamento de recursos. Como mencionado nos capítulos anteriores, a CLI do Azure também pode usar módulos adicionais *chamados de extensões. Essas extensões adicionam mais funcionalidade e geralmente são atualizadas* fora do ciclo de lançamento regular da CLI principal do Azure. A IoT do Azure está se expandindo rapidamente e adicionando novos recursos, portanto, os principais comandos para interagir com o Hub IoT são provenientes de uma extensão da CLI do Azure.

   Para obter a funcionalidade completa de que você precisa para esses exercícios, instale a extensão IoT da CLI do Azure:

   ```
   az extension add --name azure-cli-iot-ext
   ```

3. Crie um Hub IoT e digite um nome, como `azuremol`. Para esses exercícios, `você pode usar um hub IoT de camada gratuita, f1`:

   ```
   az iot hub create \
     --resource-group azuremolchapter20 \
     --name azuremol \
     --sku f1 \
     --partition-count 2
   ```

OBSERVAÇÃO Você pode criar apenas um hub de camada gratuita por assinatura, mas esses hubs são ótimos para testar a comunicação entre dispositivos e se integrar a outros serviços do Azure. O hub de camada gratuita está atualmente

limitado a 8.000 mensagens por dia e oferece suporte a no máximo 500 dispositivos conectados. Isso pode parecer muito, mas dependendo do que você está fazendo, um único dispositivo que envia uma mensagem para o hub IoT a cada 12 segundos aproximadamente aumentaria o limite de 8.000 mensagens!

Seu hub IoT está bem vazio agora. Não há muito o que fazer sem um ou mais dispositivos IoT conectados. Um dispositivo comum usado para IoT é o Raspberry Pi, um minicomputador de baixo custo que pode se conectar a redes Wi-Fi e usar sensores comuns disponíveis para medir temperatura, umidade e pressão. Você também pode usá-lo para controlar pequenos motores, luzes e temporizadores. Você não precisa se apressar e comprar um Raspberry Pi para trabalhar com um hub IoT, pode simular um em seu navegador da Web!

20.3 Criar um dispositivo Raspberry Pi simulado

Dispositivos IoT são ótimos, mas há uma barreira para a entrada em que você precisa de um dispositivo real para usar, certo? Não! Existem algumas maneiras de simular um dispositivo IoT com software. Essa abordagem baseada em software permite que você se concentre na criação da aplicação rapidamente e, depois, faça a transição para um hardware real. Você ainda precisa prestar atenção em como seu código é executado em hardware IoT real, especialmente dispositivos de baixo consumo, porque eles podem não ter acesso a todas as bibliotecas necessárias, ou mesmo a recursos de memória, às quais sua aplicação simulada tem acesso.

A Microsoft fornece um simulador de Raspberry Pi gratuito por meio do GitHub em https://azure-samples.github.io/raspberry-pi-web-simulator. Um Raspberry Pi é ótimo para testes, mas tome cuidado ao usar hardware barato pronto para uso, como o Raspberry Pi em ambientes de produção. Planeje como atualizar e gerenciar esses dispositivos. Dispositivos IoT dedicados, como o Azure Sphere (https://azure.microsoft.com/services/azure-sphere), fornecem opções de segurança e gerenciamento adicionais. Para este livro e em seu próprio teste e aprendizagem, o Raspberry Pi é uma boa alternativa. Nesse simulador, um sensor BME comum que coleta leituras de temperatura e umidade é simulado em software, juntamente com um LED simulado para mostrar quando o dispositivo transmite dados para o hub IoT. Não é possível personalizar muito, mas você pode ver como uma aplicação Node.js básico pode ser executado no Raspberry Pi, pesquise dados de um sensor e envie-os de volta ao Azure.

> **OBSERVAÇÃO** Se coisas como o Raspberry Pi, os sensores eletrônicos e de temperatura e o Node.js parecerem intimidadores, não se preocupe. Assim como nos capítulos sobre IA e ML, contêineres e Kubernetes, não faremos uma análise profunda sobre dispositivos IoT e programação. Se você acha que até o final desse capítulo ficará fascinado com a eletrônica a ponto de querer ligar um ferro de solda, você é mais do que bem-vindo!

Antes de poder usar o simulador do Raspberry Pi, você precisa criar uma atribuição de dispositivo no Hub IoT do Azure. Esse processo cria um ID de dispositivo exclusivo para que o hub IoT compreenda com qual dispositivo ele está se comunicando e como processar os dados. Em cenários mais complexos, você pode provisionar configurações adicionais para o dispositivo e enviar certificados digitais por push. Para este exercício, basta criar uma identidade de dispositivo.

Experimente agora

Para criar um dispositivo IoT Raspberry Pi simulado, conclua as etapas a seguir:

1. No Azure Cloud Shell, crie uma identidade de dispositivo no seu Hub IoT, como `azuremol`, e forneça um nome para o dispositivo, como `raspberrypi`:

```
az iot hub device-identity create \
--hub-name azuremol \
--device-id raspberrypi
```

2. Lembra-se das políticas de acesso compartilhado da seção 20.2? Cada dispositivo IoT também possui sua própria chave de acesso e string de conexão que são usadas para identificá-lo quando ele se comunica com o Hub IoT. Esse recurso-chave do Azure IoT protege os dispositivos e minimiza o risco de exposição se um dispositivo for comprometido.

   Para usar seu dispositivo com o simulador Raspberry Pi, você precisa das informações da string de conexão do dispositivo. Esse identificador exclusivo inclui o nome do host do seu hub IoT, o ID do dispositivo e uma chave de acesso:

```
az iot hub device-identity show-connection-string \
  --hub-name azuremol \
  --device-id raspberrypi \
  --output tsv
```

3. Copie o conteúdo da sua string de conexão. Você precisará dele na etapa 4. O resultado é similar ao seguinte:

```
HostName=azuremol.azure-devices.net;DeviceId=raspberrypi;
➥SharedAccessKey=oXVvK40qYYI3M4u6ZLxoyR/PUKV7A7RF/JR9WcsRYSI=
```

4. Agora vem a parte divertida! Abra o simulador Raspberry Pi em seu navegador da Web: https://azure-samples.github.io/raspberry-pi-web-simulator. Veja a seção de código à direita no simulador. Em torno da linha 15, deve haver uma variável `connectionString`, que já solicita a *[String de conexão do dispositivo de Hub IoT]*. Copie e cole a string de conexão da etapa 3, conforme mostrado na figura 20.3.
5. Selecione o botão Executar logo abaixo da janela de código para iniciar o simulador.

   A cada 2 segundos, a janela do console exibe uma mensagem que mostra os dados enviados ao hub IoT. O LED vermelho no diagrama de circuito também pisca quando isso acontece, para simular como as saídas conectadas ao Raspberry Pi podem ser controladas. A mensagem de saída na janela do console é semelhante à seguinte:

```
Sending message: {"messageId":1,"deviceId":"Raspberry Pi Web
➥Client","temperature":24.207095037347923,
➥"humidity":69.12946775681091}
```

   De onde vieram as leituras de temperatura e umidade? Esse dispositivo é um Raspberry Pi simulado e não há um sensor BME280 real, portanto, a aplicação gera esses valores no software. Se você analisar o restante do código no

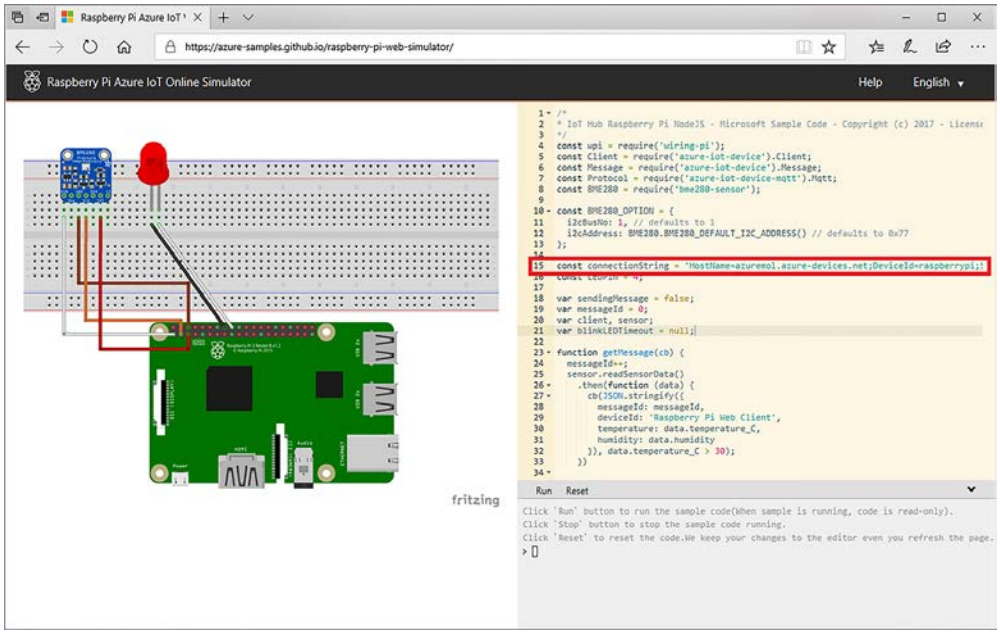


Figura 20.3 Copie e cole a string de conexão do seu dispositivo IoT do Azure no simulador do Raspberry Pi. A variável `connectionString` é usada para conectar-se para transmitir os dados do sensor simulado para o Azure.

janela do simulador, em torno da linha 99, a aplicação define o sensor. Depois, o simulador replica como o sensor real agiria e gera os dados retornados do sensor para a aplicação. É um exemplo básico, então pense no que mais você poderia ler aqui: revoluções por minuto (RPM) de um motor ou mecanismo ou coordenadas de GPS de um contêiner de transporte ou caminhão, e assim por diante. Há um equilíbrio entre simular um dispositivo no software e criar uma aplicação funcional com dados reais de hardware e sensor. Em algum momento, você precisará comprar ou pedir equipamentos emprestados se quiser analisar melhor a IoT do Azure.

6 Para confirmar que as mensagens do dispositivo simulado estão sendo recebidas pelo hub IoT, examine o status da cota. Forneça o nome do seu Hub IoT, como azuremol:

```
az iot hub show-quota-metrics --name azuremol
```

A saída é semelhante ao exemplo a seguir, que mostra que cinco mensagens do total máximo de 8.000 mensagens por dia foram recebidas e que há um dispositivo conectado de um máximo de 500 dispositivos no total. Pode levar alguns minutos para que essas métricas sejam preenchidas. Portanto, não se preocupe se você não vir dados imediatamente:

```
[
  {
    "currentValue": 5,
```

```
    "maxValue": 8000,
    "name": "TotalMessages"
  },
  {
    "currentValue": 1,
    "maxValue": 500,
    "name": "TotalDeviceCount"
  }
]
```

Você também pode procurar no portal do Azure: escolha seu grupo de recursos e selecione seu hub IoT. Na página Visão geral, o uso do Hub informa o número de mensagens recebidas e dispositivos conectados. Novamente, pode levar um ou dois minutos para que as mensagens apareçam e sejam registradas na cota. Todas as aplicações poderiam usar imediatamente as mensagens recebidas, como veremos na seção 20.4.

Problemas no paraíso

Se você não receber nenhuma mensagem no hub da IoT, verifique a janela de saída do seu dispositivo Raspberry Pi simulado. Uma das primeiras coisas que a aplicação faz é se conectar ao Hub IoT do Azure. Um erro de conexão será exibido se sua string de conexão estiver errada. Copie e cole corretamente a string de conexão inteira. A string de conexão começa com HostName e o último caractere em cada tecla de acesso é sempre um sinal de igual `(=)`.

Se a janela de saída reportar um erro, copie o texto do erro em seu mecanismo de busca preferencial e procure por um resultado correspondente. Certifique-se de que você não alterou nenhuma das outras linhas de código, o que causaria um problema. A única coisa que você precisa mudar na janela de código é a linha da sua string de conexão.

Como o dispositivo Raspberry Pi simulado é executado em um navegador da Web, você pode ter um problema de site genérico. Tente atualizar a página ou acesse o simulador em um navegador diferente (https://azure-samples.github.io/raspberry-pi-web-simulator).

20.4 Transmitir dados do hub IoT do Azure para aplicativos Web do Azure

Um dispositivo que se conecta a um Hub IoT não é útil se você não pode fazer nada com os dados. É aí que você pode começar a integrar muitos dos serviços e recursos que aprendeu neste livro. Deseja transmitir para tabelas ou filas do Armazenamento do Azure? Você pode fazer isso. Processar dados de dispositivos IoT em VMs ou contêineres do Azure? Vá em frente! Usar o Azure Cosmos DB para replicar seus dados e acessá-los com aplicativos Web do Azure globalmente redundantes e o Gerenciador de Tráfego? Claro!

No cenário de exemplo, o hub IoT é o mecanismo de conexão e o ponto de entrada de seus dispositivos IoT no Azure. O hub em si não faz nada com os dados diretamente. Existe um ponto de extremidade padrão para eventos, que é um grande intervalo para todas as mensagens recebidas do dispositivo IoT. Seu dispositivo Raspberry Pi simulado envia mensagens para o hub IoT, que atingiu esse ponto de extremidade de eventos. O fluxo de mensagens de dispositivos por meio do hub IoT para um ponto de extremidade é mostrado na figura 20.4.

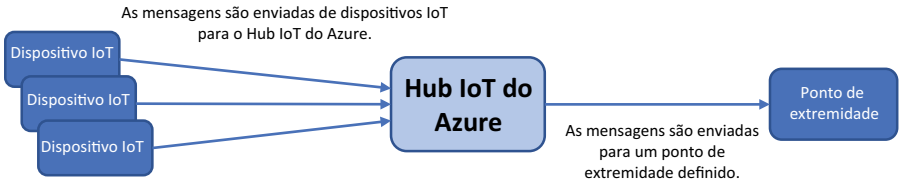


Figura 20.4 Um hub IoT recebe mensagens de dispositivos IoT conectados e envia as mensagens para um ponto de extremidade. Esses pontos de extremidade podem ser usados por outros serviços do Azure para consumir dados dos dispositivos IoT. Existe um ponto de extremidade padrão para eventos, do qual podem ser lidos serviços como aplicativos Web.

Você pode criar pontos de extremidade personalizados que roteiam mensagens diretamente para serviços do Azure, como Armazenamento e Barramento de Serviço. No capítulo 4, examinamos as filas do Armazenamento do Azure para obter uma maneira de retornar mensagens entre as aplicações. Uma plataforma de mensagens corporativa mais robusta e escalonável é o Barramento de Serviço do Azure. As mensagens podem ser adicionadas ao barramento de serviço, como dados recebidos de dispositivos IoT, e outras aplicações podem ouvir essas mensagens e responder adequadamente.

Se você não precisar da complexidade de ler mensagens de algo como um barramento de serviço, poderá usar grupos de consumidores com o ponto de extremidade de eventos padrão. Um grupo de consumidores permite que serviços como os Aplicativos Web do Azure leiam dados do ponto de extremidade, como mostra a figura 20.5. Cada leitura de serviço do Hub IoT do Azure deve ter seu próprio grupo de consumidores. Vários serviços, cada um com seu próprio grupo de consumidores, podem receber as mesmas mensagens e processá-las conforme necessário.

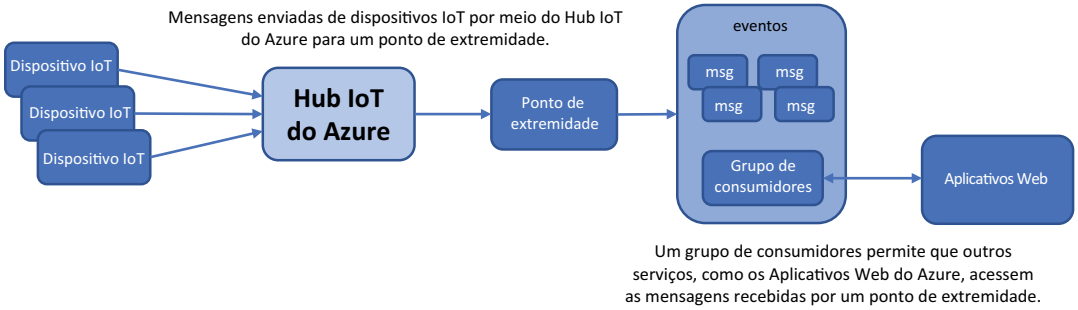


Figura 20.5 As mensagens são enviadas de dispositivos IoT para o hub IoT, que direciona as mensagens para um ponto de extremidade. Em cada ponto de extremidade, podem ser criados grupos de consumidores. Esses grupos de consumidores permitem que outros serviços do Azure acessem as mensagens do dispositivo, às quais não teriam acesso de outra forma. Com grupos de consumidores, você não precisa usar filas de mensagens para permitir que aplicações externas leiam dados do dispositivo IoT.

Vamos criar um aplicativo Web do Azure que use um grupo de consumidores para ler dados de mensagens em tempo real a partir do seu dispositivo Raspberry Pi simulado. Este exemplo básico mostra como você pode transmitir dados de dispositivos IoT e acessá-los a partir de aplicativos Web.

Experimente agora

Para criar um aplicativo Web do Azure que leia dados de dispositivos IoT, conclua as etapas a seguir:

1. Crie um plano do Serviço de Aplicativo do Azure para seu aplicativo Web no Azure Cloud Shell e atribua um nome, como `azuremol`. Para esses exercícios, a camada gratuita (f1) é boa o suficiente e mantém os custos baixos:

```
az appservice plan create \
  --resource-group azuremolchapter20 \
  --name azuremol \
  --sku f1
```

2. Crie o aplicativo Web. Forneça um nome, como molwebapp, e ative-o para uso com o Git para que você possa implantar a aplicação de exemplo. Assim como outros recursos do Azure acessíveis ao público, você precisa atribuir seu próprio nome globalmente exclusivo.

```
az webapp create \
  --resource-group azuremolchapter20 \
  --plan azuremol \
  --name molwebapp \
  --deployment-local-git
```

3. Defina o grupo de consumidores para o seu Hub IoT, junto com algumas configurações de aplicativos Web. Essas configurações permitem que seu aplicativo Web se conecte ao Hub IoT. A Figura 20.6 mostra o que você criará nas próximas etapas.

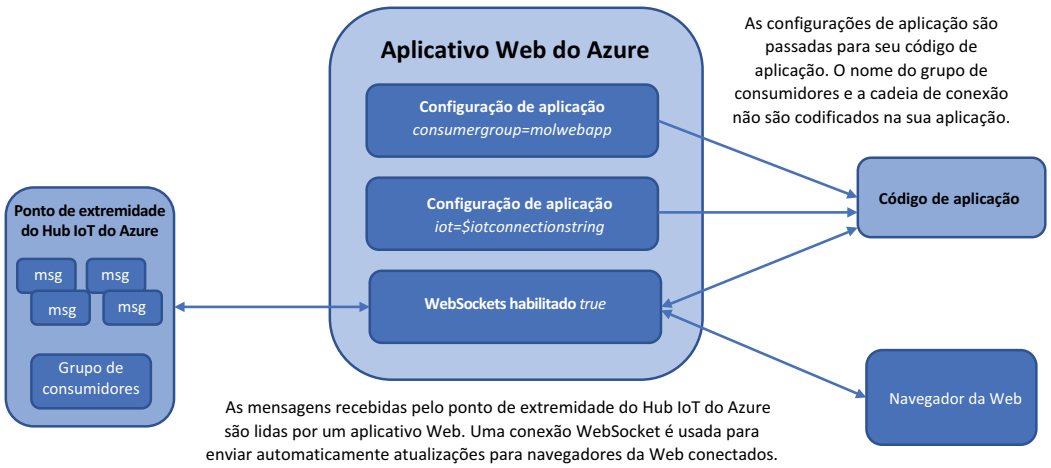


Figura 20.6 Para permitir que o aplicativo Web leia os dados do dispositivo IoT Raspberry Pi simulado, crie um grupo de consumidores no hub IoT. Depois, você definirá duas configurações de aplicação para seu aplicativo Web que permitem conectar-se ao grupo de consumidores. Para permitir que o navegador receba automaticamente o fluxo de dados do Raspberry Pi à medida que novos dados sejam recebidos, você também ativará uma configuração para WebSockets.

4 Crie um grupo de consumidores que permita que seu aplicativo Web acesse os dados do evento transmitidos pelo dispositivo IoT. Forneça seu Hub IoT, como `azuremol`, e insira um nome para o grupo de consumidores, como `molwebapp`. Use seu próprio nome ao longo das próximas etapas. Seu grupo de consumidores será criado no ponto de extremidade de eventos padrão:

```
az iot hub consumer-group create \
  --hub-name azuremol \
  --name molwebapp
```

5 Você precisa informar ao aplicativo Web o nome do grupo de consumidores. Crie uma configuração de aplicativo Web usada pela aplicação de exemplo que você implantará no final do exercício. As configurações de aplicação em aplicativos Web permitem que você defina configurações específicas, como o nome do grupo de consumidores e a string de conexão, sem que esses valores sejam codificados em sua aplicação.

Informe o nome do grupo de consumidores criado na etapa 4, como `mol-webapp`:

```
az webapp config appsettings set \
  --resource-group azuremolchapter20 \
  --name molwebapp \
  --settings consumergroup=molwebapp
```

6 Para se conectar ao hub IoT, seu aplicativo Web precisa saber a string de conexão do hub. Essa string de conexão é diferente daquela que você copiou para o seu dispositivo Raspberry Pi simulado no exercício anterior. Lembre-se de que há uma string de conexão para seu Hub IoT, que usa as políticas de acesso compartilhado para definir permissões de acesso. Além disso, há uma string de conexão para cada dispositivo IoT. O aplicativo Web precisa ler a partir do grupo de consumidores de ponto de extremidade do Hub IoT. Então, você deve definir uma string de conexão para o próprio Hub IoT.

7 Obtenha a string de conexão do hub IoT e a atribua a uma variável denominada `iotconnectionstring`, usada na etapa 8:

```
iotconnectionstring=$(az iot hub show-connection-string \
  --hub-name azuremol \
  --output tsv)
```

8 Crie outra configuração de aplicativo Web, desta vez para a string de conexão do Hub IoT. A variável definida na etapa 7 é usada para permitir que a aplicação de exemplo se conecte e leia dados do dispositivo IoT:

```
az webapp config appsettings set \
  --resource-group azuremolchapter20 \
  --name molwebapp \
  --settings iot=$iotconnectionstring
```

9 Habilite WebSockets. O *WebSocket* é um meio de comunicação bidirecional entre um navegador e um servidor. A aplicação de exemplo atualiza automaticamente o navegador da Web com os dados recebidos do dispositivo Raspberry Pi. Para

executar essa atualização automatizada, a aplicação usa o WebSockets. Então, o servidor poderá enviar dados por push ao navegador e fazer com que ele atualize automaticamente:

```
az webapp config set \
--resource-group azuremolchapter20 \
--name molwebapp \
--web-sockets-enabled
```

Vamos fazer uma pausa aqui e abordar o que você fez até agora. Você trabalhou com aplicativos Web em muitos dos capítulos anteriores, mas as configurações do aplicativo Web e os WebSockets são novos. A Figura 20.7 recapitula como seu aplicativo Web e o hub IoT estão conectados.

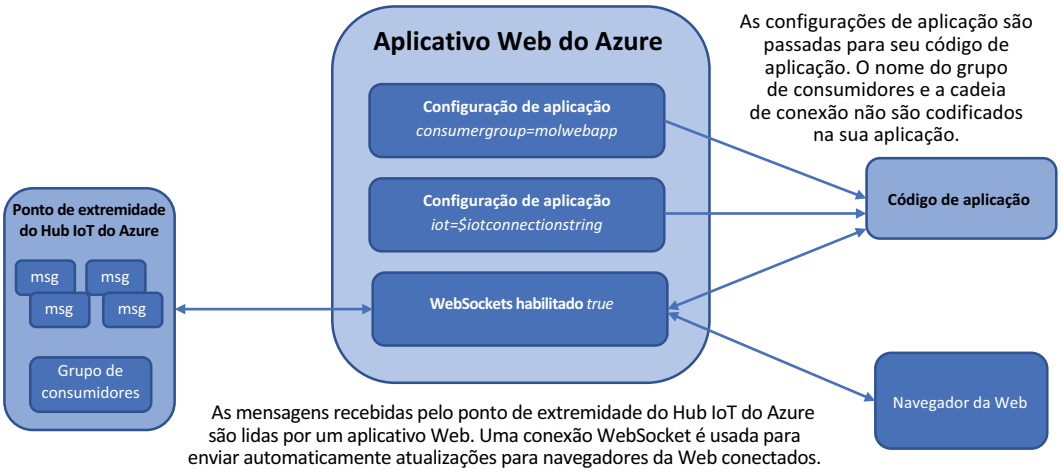


Figura 20.7 À medida que as mensagens são enviadas de dispositivos IoT, elas passam pelo hub IoT para um ponto de extremidade. O código da aplicação lê as configurações do aplicativo Web que definem a string de conexão do hub IoT e o grupo de consumidores a ser usado. Depois que a aplicação é conectada ao Hub IoT, o grupo de consumidores permite que os aplicativos Web leiam as mensagens do dispositivo IoT. Cada vez que uma nova mensagem for recebida de um dispositivo IoT, seu aplicativo Web usará uma conexão do WebSocket com navegadores da Web que acessam seu site para enviar atualizações por push automaticamente. Essa conexão permite exibir dados em tempo real transmitidos de dispositivos IoT, como informações de temperatura e umidade, a partir do seu dispositivo Raspberry Pi simulado.

Agora, vamos concluir o exercício e implantar a aplicação de exemplo do repositório do GitHub no aplicativo Web. Você poderá abrir o aplicativo Web no navegador e ver os dados em tempo real transmitidos do Raspberry Pi simulado!

10 Se necessário, clone o repositório de exemplos do GitHub em seu Cloud Shell da seguinte maneira:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

11 Mude para o diretório do capítulo 20:

```
cd azure-mol-samples-2nd-ed/20
```

12 Inicialize o repositório Git e adicione a página da Web básica:

```
git init && git add . && git commit -m "Pizza"
```

13 Para carregar a aplicação de exemplo, crie uma conexão com seu aplicativo Web. O comando a seguir obtém o repositório de aplicativos Web e configura seu repositório Git de exemplos locais para se conectar a ele:

```
git remote add molwebapp \
  $(az webapp deployment source config-local-git \
  --resource-group azuremolchapter20 \
  --name molwebapp \
  --output tsv)
```

Nos capítulos anteriores, fiz você procurar esse endereço, mas agora espero que você tenha começado a explorar o que mais a CLI do Azure pode fazer e percebido que muitas dessas informações podem ser obtidas rapidamente.

14 Envie por push o site de exemplo HTML para o aplicativo Web com o seguinte comando:

```
git push molwebapp master
```

15 Quando solicitado, insira a senha para o usuário do Git que você criou e usou nos capítulos anteriores (a conta criada no capítulo 3).

Se você não escreveu sua senha do Git em um Post-It

Se você esqueceu a senha, pode redefini-la. Primeiro, obtenha o nome de usuário da sua conta de implantação do Git local:

```
az webapp deployment user show --query publishingUserName
```

Para redefinir a senha, insira o nome da sua conta do comando anterior e responda às instruções para definir uma nova senha. O exemplo a seguir redefine a senha da conta de usuário denominada azuremol:

```
az webapp deployment user set --user-name azuremol
```

16 Visualize o nome do host do aplicativo Web e abra o endereço em um navegador da Web:

```
az webapp show \
  --resource-group azuremolchapter20 \
  --name molwebapp \
  --query defaultHostName \
  --output tsv
```

Pode levar alguns segundos na primeira vez que você abrir o site em seu navegador da Web, pois o aplicativo Web se conectará ao hub IoT, iniciará a conexão do WebSocket e aguardará a primeira mensagem do dispositivo a ser recebida. A cada 2 segundos, o navegador da Web deverá atualizar automaticamente com os dados simulados mais recentes do dispositivo Raspberry Pi, conforme mostrado na figura 20.8.

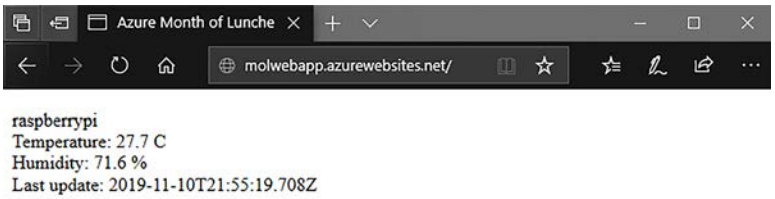


Figura 20.8 A aplicação de exemplo usa uma conexão do WebSocket entre o navegador da Web e o aplicativo Web para atualizar automaticamente a cada 2 segundos com os dados mais recentes do dispositivo Raspberry Pi simulado.

Se a instância do aplicativo Web não mostrar dados, verifique se o dispositivo Raspberry Pi simulado ainda está em execução. Se necessário, inicie o dispositivo simulado e verifique se ele se conecta à IoT do Azure e envia mensagens. Os dados devem começar a aparecer na instância do aplicativo Web.

20.5 Revisão do componente IoT do Azure

Espero que os exercícios desse capítulo tenham dado a você uma ideia de quais serviços estão disponíveis no Azure para soluções de IoT:

- *O* Hub IoT do Azure *fornece uma ótima maneira de provisionar, conectar e gerenciar muitos dispositivos IoT, além de integrar-se a outros serviços do Azure.*
- *Os* Aceleradores de solução de IoT do Azure *fornecem cenários pré-criados que integram automaticamente muitos serviços diferentes do Azure para fornecer um ambiente de aplicações completo.*
- O *Azure IoT Edge* permite implantar serviços do Azure em seu ambiente local para processar dados de dispositivos IoT sem a necessidade de transmitir todos os dados centralmente de volta ao Azure.

Para realmente explorar os dispositivos IoT e a IoT do Azure em geral, recomendo que você compre um Raspberry Pi básico ou um dispositivo semelhante. Eles são relativamente baratos, geralmente vêm com alguns sensores básicos ou componentes elétricos para testar ideias diferentes e oferecem uma ótima plataforma de aprendizado, pois você vê o que é capaz de fazer ao integrar o hardware e o software. Basta lembrar os avisos do capítulo 17 sobre IA e ML e a criação da Skynet! A Manning também tem alguns excelentes livros, como *Building the Web of Things*, de Dominique D. Guinard e Vlad M. Trifa (https://www.manning.com/books/building-the-web-of-things) e *JavaScript on Things* , de Lyza Danger Gardner (https://www.manning.com/books/javascript-on-things), que entram em mais detalhes no Raspberry Pi, nas práticas recomendadas de IoT e na programação de JavaScript e Node.js em dispositivos IoT.

Lembra-se de como eu disse para sempre excluir seus grupos de recursos?

A melhor prática em todo esse livro foi excluir seus grupos de recursos no final de cada capítulo. Essa abordagem garante que você não deixe serviços e aplicações que custam dinheiro em uso quando não precisa deles.

(continuação)

O Azure IoT oferece uma ótima plataforma para transmitir dados para o Azure. Você normalmente precisa processar esses dados, não apenas exibi-los em um aplicativo Web como nos exercícios. O capítulo 21 examina a computação sem servidor com os serviços de Aplicativos Lógicos e Funções.

Para mostrar como esses serviços do Azure funcionam bem juntos, não exclua o grupo de recursos e os serviços implantados neste capítulo. Você os usará logo no início do capítulo 21 para ver como pode executar ações com base nos dados recebidos de seus dispositivos IoT. Apenas certifique-se de voltar ao dispositivo Raspberry Pi simulado e selecionar o botão Parar – caso contrário, esse limite de 8.000 mensagens será usado muito rapidamente!

20.6 Laboratório: explorar casos de uso para IoT

Esse capítulo abordou muitas novidades e, sem um dispositivo IoT real, você está limitado ao que pode fazer. O capítulo 21 baseia-se no Hub IoT do Azure e no Raspberry Pi simulado, por isso não quero fazer mais configurações agora. Veja algumas coisas que você pode fazer para refletir sobre a IoT:

1. Você pode pensar em quais áreas os dispositivos IoT podem beneficiar sua empresa? Se você não trabalha em uma empresa agora, pense na pizzaria fictícia do mês de aulas do Azure:
2. O que você poderia fazer para melhorar as coisas para os clientes com IoT?
3. Você usaria o Azure IoT Edge? Por que ou por que não?
4. Quais outros serviços do Azure você provavelmente integraria para executar suas aplicações?
5. Se você tiver tempo sobrando no seu horário de almoço, experimente um dos aceleradores de solução IoT do Azure em www.azureiotsolutions.com/Accelerators. Há um cenário de Simulação de dispositivos que cria uma VM e sensores simulados, que é como o dispositivo Raspberry Pi simulado, mas muito maior! Leva alguns minutos para provisionar todos os recursos necessários, mas, depois procure no portal do Azure para ver o que foi criado e como todas as partes funcionam juntas:
6. Você pode ver como os serviços dos capítulos anteriores, como Armazenamento e Cosmos DB, são usados?
7. Quais outros aceleradores de solução de IoT estão disponíveis? Algum deles se alinha com as ideias que você tinha para suas próprias aplicações?

21 Computação sem servidor

Neste capítulo final, contemplaremos o futuro da computação sem servidor. Se você é desenvolvedor, a ideia de contêineres examinados no capítulo 19 pode ser interessante, pois há menos necessidade de configurar a infraestrutura subjacente para suas aplicações. Se a resposta for afirmativa, você vai adorar os componentes sem servidor do Azure! E se você é um administrador de TI que, de repente, se pergunta qual será seu trabalho se não houver servidores no futuro, não se preocupe! *Computação sem servidor* pode ser mais um termo de marketing e muitas das habilidades de servidor e infraestrutura que você continua aplicando!

No Azure, duas principais ofertas oferecem recursos de computação sem servidor: Aplicativos Lógicos do Azure e Aplicativos de Função do Azure. Neste capítulo, exploraremos o que cada serviço oferece e como eles podem trabalhar juntos. Para garantir que suas aplicações sem servidor possam se comunicar e transmitir dados, também abordamos os serviços de mensagens, como a Grade de Eventos, o Barramento de Serviço e os Hubs de Eventos do Azure.

21.1 O que é a computação sem servidor?

Dizer que essa computação sem servidor está sem um servidor é simplesmente errado: um servidor, em algum lugar, executa algum código para você. A diferença dos workloads de aplicações IaaS, como VMs do Azure e workloads de PaaS em aplicativos Web, é que as aplicações sem servidor geralmente são divididas em unidades discretas menores de uma aplicação. Você não executa um única aplicação grande. Em vez disso, executa componentes de aplicações pequenos. Se isso faz você lembrar dos contêineres e microsserviços que abordamos no capítulo 19, não se preocupe caso esteja enlouquecendo: a computação sem servidor tem muita coisa em comum com esses tópicos em termos de como você projeta suas aplicações. Você pode criar microsserviços usando as abordagens sem servidor que vamos examinar neste capítulo.

A figura 21.1 mostra como uma aplicação é dividida em pequenos componentes que são executados em um provedor de computação sem servidor e fornecem pequenas unidades de saída.

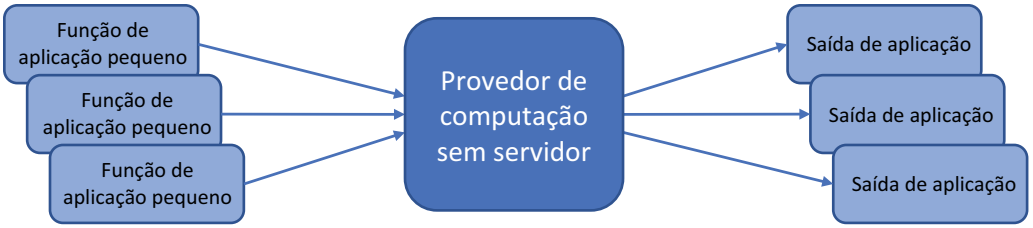


Figura 21.1 Em um ambiente de computação sem servidor, cada aplicação é dividido em unidades pequenas e discretas de componentes de aplicações. Cada componente é executado em um provedor de computação sem servidor, como os Aplicativos de Função do Azure, e é produzida uma saída que pode ser consumida por outros componentes da aplicação sem servidor ou outros serviços do Azure, como a IoT do Azure ou o Armazenamento do Azure.

No Azure, a computação sem servidor abrange dois serviços principais:

- *Aplicativos lógicos do Azure*: para responder a determinadas entradas e acionadores, os aplicativos lógicos permitem criar visualmente fluxos de trabalho que podem processar e gerar ações adicionais apenas apontando e clicando, sem necessidade de código. Os aplicativos lógicos podem ser criados por usuários sem experiência em programação ou infraestrutura de TI. Um esboço de aplicativo lógico simples é mostrado na figura 21.2.

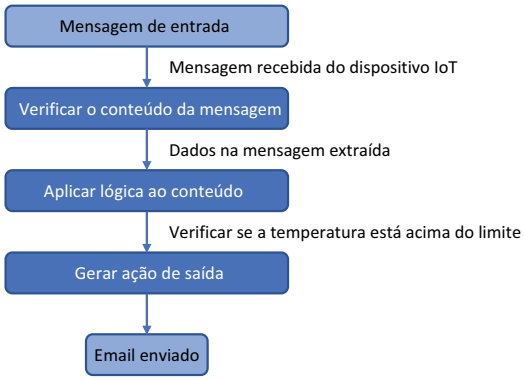


Figura 21.2 Em um aplicativo lógico, poderá ser uma entrada quando um tweet for publicado, um arquivo for carregado ou uma mensagem for recebida de um dispositivo IoT. O aplicativo lógico aplicará regras e filtros aos dados e determinará se a mensagem atende aos critérios definidos. Depois, as ações de saída, como gerar um email, serão concluídas. Toda essa lógica não envolve nenhuma infraestrutura de programação ou aplicação além de uma assinatura do Azure.

Não há atualizações de segurança a serem mantidas nem nenhum requisito de design em torno da alta disponibilidade ou da capacidade de escalar. A plataforma do Azure manipula isso automaticamente. Existem centenas de conectores pré-criados para os aplicativos lógicos se integrarem a serviços como Twitter, Office 365, SharePoint e Outlook. Você pode responder a tweets públicos sobre sua empresa ou produto, enviar um alerta por email quando um arquivo for carregado no SharePoint ou enviar uma notificação quando uma mensagem for recebida de um dispositivo IoT.

- *Aplicativos de Função do Azure*: para executar pequenos blocos de código, os aplicativos de função permitem usar linguagens de programação comuns, como C#, Node.js ou Python, sem qualquer gerenciamento de infraestrutura adicional.

Seu código será executado em um ambiente seguro e isolado, e você será cobrado com base no consumo de memória por segundo. A Figura 21.3 descreve o processo básico de um aplicativo de função.

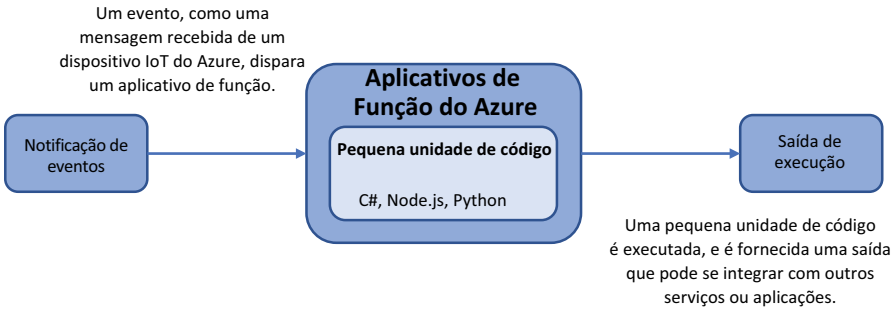


Figura 21.3 Como acontece com um aplicativo lógico, uma notificação ou acionador de evento geralmente inicia uma função do Azure. O aplicativo de função contém uma pequena unidade de código que executa uma tarefa específica. Não há nenhuma infraestrutura a ser configurada ou mantida. Apenas o pequeno bloco de código será necessário. Quando a execução do código estiver concluída, a saída poderá ser integrada a outro serviço ou aplicação do Azure.

Não há VMs a serem mantidas e nenhum aplicativo Web será necessário. Você não precisa se preocupar com alta disponibilidade ou escala, porque o serviço de Aplicativos de Função do Azure cuida disso para você. Tudo o que você fornece é seu código, a plataforma do Azure garante que sempre que você precisar executar esse código, os recursos estarão disponíveis para processar sua solicitação.

Aplicativos lógicos não requerem código, eles têm uma base de possíveis usuários mais ampla. Os proprietários de aplicações de negócios ou as equipes de finanças e contabilidade, por exemplo, podem criar suas próprias aplicações lógicas sem precisar escrever códigos. As aplicações de função fornecem mais controle e flexibilidade e permitem que você manipule eventos de maneira específica e se integre melhor a outros componentes da aplicação.

As aplicações lógicos e as aplicações de função fornecem uma maneira de executar ações com base em acionadores sem precisar manter nenhum ambiente de aplicação ou infraestrutura. Um servidor em algum lugar no Azure executa o aplicativo lógico ou a função, mas da sua perspectiva como o administrador ou desenvolvedor de TI, essas são tecnologias sem servidor.

21.2 Plataformas de mensagens do Azure

No capítulo 12, analisamos como monitorar e solucionar problemas dos recursos do Azure, e no capítulo 16 vimos como usar a Central de Segurança do Azure para detectar problemas e executar o gerenciamento de atualizações. Os dois recursos dependem de fluxos de dados, como a extensão de diagnóstico da VM do Azure, para informar à plataforma o que está acontecendo na VM. As plataformas de monitoramento e diagnóstico do Azure são excelentes, e outros serviços, como Aplicativos Web, Instâncias de Contêiner do Azure e Hub IoT do Azure, também podem transmitir diagnósticos de serviço para análise central.

Com aplicações sem servidor, você geralmente precisará trocar mensagens e transmitir dados reais da aplicação, não apenas solucionar diagnósticos ou atualizações de status. É nesse ponto que você precisa de uma plataforma de mensagens.

21.2.1 Grade de Eventos do Azure

E se você quiser apenas informar sobre certas ações ou atividades que estão sendo concluídas? Em fluxos de trabalho de automação e computação sem servidor, a capacidade de executar uma ação em resposta a um evento é útil, como mostra a figura 21.4.

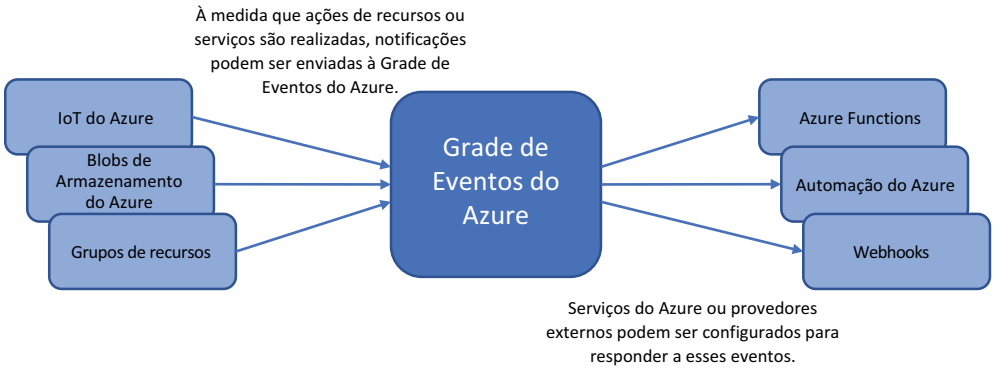


Figura 21.4 Os serviços do Azure, como a IoT do Azure e o Armazenamento do Azure, podem enviar notificações para a Grade de Eventos do Azure. Essas notificações podem ocorrer quando uma mensagem for recebida de um dispositivo IoT ou um arquivo for carregado no armazenamento. A Grade de Eventos do Azure permite que outros serviços e provedores assinem essas notificações para executar ações adicionais em resposta a eventos.

Vamos examinar alguns cenários que você pode usar em sua pizzaria:

- *Mensagem recebida em um Hub IoT: um dispositivo IoT conectado ao hub IoT pode relatar uma leitura de temperatura em um forno* ou a localização de um veículo de entrega. O Hub IoT é *configurado para encaminhar uma notificação para a* Grade de Eventos do Azure.

  Uma função do Azure é inscrita nas notificações da Grade de Eventos para o hub IoT e executa um pequeno componente de aplicação sem servidor para registrar as informações no Cosmos DB e enviar uma notificação por email. Você também pode usar aplicações lógicas em vez de aplicações de função do Azure, dependendo de qual complexidade a resposta da aplicação precisa ter.
- *Arquivo carregado no armazenamento do Azure: o departamento de marketing pode carregar no armazenamento um cupom promocional para economizar dinheiro* em um pedido de pizza. Quando um novo arquivo for criado, *uma notificação será enviada para a* Grade de Eventos.

  Um webhook será inscrito na Grade de Eventos e publicará uma cópia da imagem do armazenamento no Twitter. Esse tweet permitirá que os clientes saibam sobre a oferta da semana ou sobre o cupom para economizar dinheiro.

Esses cenários são para casos de computação sem servidor, mas a Grade de Eventos também poderá se integrar a recursos mais tradicionais, como VMs e aplicativos Web. Por exemplo, um grupo de recursos pode ser configurado para enviar notificações para a Grade de Eventos. Há muitas maneiras de criar uma VM, como no portal, com a CLI do Azure ou com um modelo do Gerenciador de Recursos, portanto, você deve certificar-se de que a VM esteja configurada corretamente para o Gerenciamento de Atualizações por meio da Central de Segurança. Um runbook de Automação do Azure poderia ser inscrito na Grade de Eventos para receber notificações sobre operações de criação de VMs, incorporar a VM ao serviço de Gerenciamento de Atualizações e instalar atualizações de segurança ou de aplicação necessárias.

21.2.2 Hubs de Eventos e Barramento de Serviço do Azure

A Grade de Eventos pode funcionar com muitos recursos do Azure e é adequada para computação sem servidor com aplicativos lógicos ou aplicativos de função. Mas aplicativos lógicos e aplicativos de função podem ser executados com base em outras entradas de dados, como hubs de eventos ou um barramento de serviço. Vejamos as diferenças entre esses vários serviços de mensagens para que você possa decidir melhor quando usá-los:

- *O Hub de Eventos do Azure* permite que você receba um fluxo de dados, como dispositivos IoT ou telemetria de aplicações. Os hubs de eventos fornecem uma plataforma de mensagens de baixa latência capaz de lidar com milhões de eventos por segundo de vários provedores simultâneos. Os Hubs de Eventos são um armazenamento de dados, não uma fila de mensagens, e o cliente ou a aplicação verifica eventos no Hub em qualquer frequência que você quiser. Em seguida, os dados recebidos no Hub de Eventos podem ser processados por outros serviços, como mostrado na Figura 21.5.
- *O Barramento de Serviço do Azure* permite que os componentes da aplicação troquem dados de mensagens, como as filas de armazenamento examinadas no capítulo 4. As filas de armazenamento são uma implantação anterior e mais básica de uma plataforma de mensagens no Azure. Um *Barramento de Serviço*

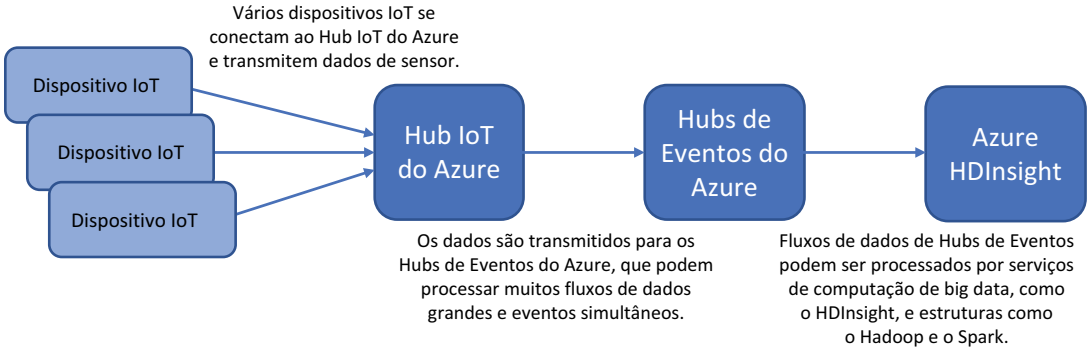


Figura 21.5 Os dispositivos IoT se conectam ao Hub IoT e podem transmitir todos os dados do sensor. Pode haver centenas ou milhares de dispositivos IoT conectados. Os Hubs de Eventos do Azure tratam de todos esses fluxos de dados separados e permitem que serviços como o Azure HDInsight processem os dados brutos nos clusters Hadoop ou Spark para analisar e gerar relatórios.

oferece recursos mais avançados, como garantia da ordem das mensagens, operações atômicas e envio de mensagens em lotes. A figura 21.6 descreve um cenário comum para um barramento de serviço.

Figura 21.6 As mensagens são colocadas em uma fila de barramento de serviço pelos componentes da aplicação – uma aplicação de front-end, neste exemplo. Outras aplicações de middleware ou de back-end poderão pegar essas mensagens e processá-las conforme necessário. Aqui, uma aplicação de back-end pega a mensagem e a processa. Os recursos avançados de mensagens incluem a garantia da ordem das mensagens na fila, bloqueio de mensagens, tempos limite e retransmissões.

Com três serviços que permitem transmitir, receber e processar dados entre aplicações e serviços no Azure, qual você usa e quando? A tabela 21.1 fornece uma recapitulação de alto nível dos serviços de Grade de Eventos, Hubs de Eventos e Barramento de Serviço.

Tabela 21.1 Cada serviço é projetado para abranger um cenário diferente. A Grade de Eventos permite reagir a eventos, os Hubs de Eventos permitem transmitir grandes quantidades de dados e o Barramento de Serviço permite transmitir mensagens entre serviços e componentes da aplicação.

Serviço do Azure	Fornece	Caso de uso
Grade de Eventos	Distribuição de eventos	Realizar uma ação adicional com base em uma ocorrência de evento.
Hubs de Eventos	Fluxos de dados	Receber e transmitir grandes volumes de dados simultâneos.
Barramento de Serviço	Transmissão de mensagens	Fornecer comunicação entre serviços e aplicativos.

Aplicativos lógicos e aplicativos de função do Azure podem ser acionados por todas as três plataformas de mensagens. Vamos criar um barramento de serviço que possa ser usado para acionar um aplicativo lógico.

21.2.3 Criar um barramento de serviço e integrá-lo a um hub IoT

Nesse cenário, vamos usar um barramento de serviço para transmitir mensagens recebidas de um hub IoT. Seu dispositivo Raspberry Pi simulado do capítulo 20 gera leituras de temperatura e as transmite para o Hub IoT. Se a temperatura for superior a 30° C, outro dado será incluído na mensagem do dispositivo IoT: temperature-Alert = true. A Figura 21.7 descreve como você pode integrar um hub IoT ao barramento de serviço para processar mensagens com esse alerta de temperatura.

Figura 21.7 Quando o dispositivo Raspberry Pi IoT simulado envia dados de mensagens, uma leitura de temperatura de 30° C ou mais gera um alerta. As mensagens marcadas com esse alerta são colocadas em um barramento de serviço. Essas mensagens podem ser usadas para acionar aplicativos lógicos.

Experimente agora

Para criar um barramento de serviço, conclua as etapas a seguir:

1. Abra o portal do Azure e selecione Create a Resource (Criar um Recurso) no canto superior esquerdo do menu.
2. Procure e selecione Barramento de Serviços e escolha Criar.
3. Informe um nome, como `azuremol` e selecione a camada de preços Básica.
4. Crie um novo grupo de recursos e informe um nome, como `azuremol-chapter21`. Verifique se o local é o mesmo dos recursos criados no capítulo 20, como o leste dos EUA. A interação entre uma fila de barramento de serviços, um aplicativo lógico e um aplicativo de função pode ter problemas se você não estiver consistente com seus locais.
5. Aceite os outros padrões e escolha criar o barramento de serviço.
6. Depois de criar o recurso, selecione seu grupo de recursos e escolha o barramento de serviço que foi criado na etapa 5.
7. Selecione Filas. Adicione uma nova fila e insira um nome, como `azuremol`.
8. Aceite todos os outros padrões e escolha Criar.

Com um barramento de serviço e uma fila criada, como você configura um hub IoT para usá-los? No Hub IoT, você define *pontos de extremidade* como os destinos de mensagens recebidas de dispositivos IoT. Existe um ponto de extremidade padrão no hub IoT para todas as mensagens que não atendem aos critérios definidos. Você pode configurar o barramento de serviço como um ponto de extremidade para receber mensagens. *Será definida uma rota* que inclui critérios para os quais as mensagens devem ser direcionadas para um ponto de extremidade. Neste exemplo, essa rota requer que qualquer mensagem que contenha temperatureAlert = true no corpo da mensagem seja encaminhada para o ponto de extremidade do barramento de serviço, conforme mostrado na figura 21.8.

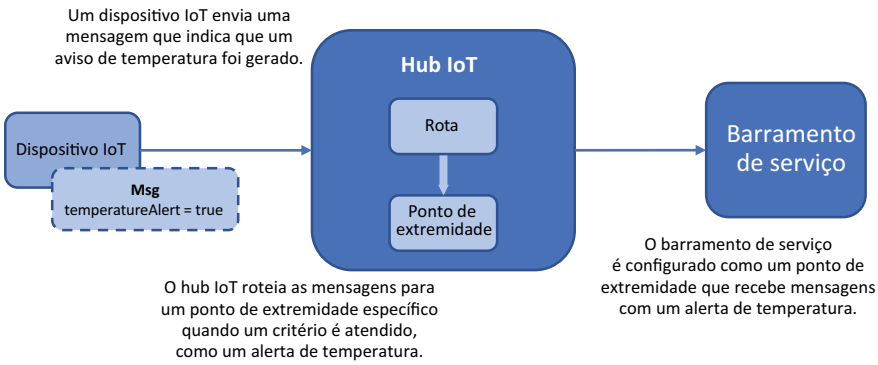


Figura 21.8 Como as mensagens são transmitidas de dispositivos IoT para um hub IoT, elas podem ser encaminhadas para pontos de extremidade específicos com base nos critérios definidos por você. Mensagens que contêm um alerta de temperatura no corpo da mensagem podem ser encaminhadas para um ponto de extremidade que usa a fila do barramento de serviço. As mensagens colocadas na fila do barramento de serviço que contêm um alerta de temperatura podem ser usadas para acionar coisas como aplicativos lógicos ou aplicativos de função do Azure.

Experimente agora

Para configurar um hub IoT para rotear mensagens de alerta de temperatura para o barramento de serviço, conclua as etapas a seguir:

1. Selecione seu grupo de recursos no capítulo 20, como `azuremolchapter20`, e escolha o IoT hub.
2. Em Mensagens na barra de navegação à esquerda, selecione Roteamento de mensagens e adicione um ponto de extremidade personalizado para uma fila de barramentos de serviço.
3. Informe o nome de um ponto de extremidade, como `azuremol`.
4. Selecione seu namespace de fila do barramento de serviços, como `azuremol`, e sua fila real.
5. Para direcionar mensagens para esse ponto de extremidade, crie uma rota. Na seção Roteamento de mensagens da barra de navegação à esquerda, selecione Rotas e escolha Adicionar uma nova rota.
6. Informe um nome, como `temperatureAlert`.
7. Escolha o ponto de extremidade do barramento de serviço que você criou na etapa anterior, como `azuremol`.
8. Para a consulta de roteamento, insira o seguinte:

```
temperatureAlert = "true"
```

9. Quando estiver pronto, salve a rota.

Agora você tem um dispositivo Raspberry Pi simulado que envia dados para o hub IoT e uma rota para colocar mensagens que contêm um alerta de temperatura em uma fila de mensagens do barramento de serviço. Você ainda não tem uma aplicação, não há nada que possa fazer com os dados na fila do barramento de serviços. O que você pode querer fazer com um alerta de temperatura? O envio de uma notificação por email é um exemplo comum, por isso veremos como você pode acionar um aplicativo lógico cada vez que uma mensagem for colocada na fila do barramento de serviço.

21.3 Criar um aplicativo lógico do Azure

Como você viu quando abordamos aplicativos lógicos na seção 21.1, uma mensagem recebida de uma fila de barramento de serviço pode ser usada como um gatilho para iniciar o processo de execução. Você usa o hub IoT para processar as mensagens recebidas de dispositivos IoT e encaminhar somente para as mensagens do ponto de extremidade da fila do barramento de serviço que contêm `temperatureAlert = true` no corpo da mensagem. Com essa abordagem, seu aplicativo lógico só será executado quando um alerta de temperatura for gerado.

A figura 21.9 descreve o que seu aplicativo lógico faz. Quando uma mensagem for colocada na fila do barramento de serviço, o aplicativo lógico será executado e enviará um alerta por email.

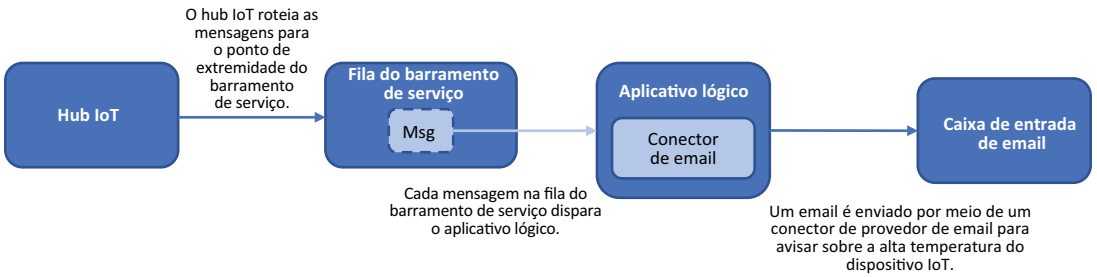


Figura 21.9 Cada mensagem recebida na fila de barramento de serviço do hub IoT aciona o aplicativo lógico. Quando o aplicativo lógico for executado, ele enviará uma notificação por e-mail por meio de um provedor de e-mail definido.

Experimente agora

Para criar um aplicativo lógico, conclua as etapas a seguir:

1. No portal do Azure, selecione Criar um recurso na parte superior esquerda do menu.
2. Procure e selecione Aplicativo Lógico e escolha Criar.
3. Informe um nome, como `azuremol`, e selecione seu grupo de recursos, como `azuremolchapter21`. Novamente, escolha o mesmo local que seus outros recursos de IoT do capítulo 20.
4. Aceite os outros padrões e escolha Criar.
5. Depois que o recurso for criado, selecione seu grupo de recursos e abra o aplicativo lógico. Para a opção "Adicionar gatilhos comuns", escolha "Quando uma mensagem é recebida em uma fila de barramento de serviço".

6 Informe um nome, como `azuremol`. Depois, selecione sua fila de barramento de serviço, como azuremol.

7 Escolha a política de barramento de serviço padrão listada, como RootManageSharedAccess-Key, e crie a conexão.

8 Selecione Continuar e escolha o nome da fila do barramento de serviços, como `azuremol`.

9 Aceite os padrões, como a frequência para verificar se há mensagens.

10 Escolha Adicionar uma nova etapa ao aplicativo lógico.

11 Para adicionar uma ação, procure o que deseja fazer. Neste exercício, procure *email*. Selecione seu provedor, como Gmail - Enviar um e-mail, Outlook.com - Enviar um e-mail ou SMTP - Enviar um e-mail, conforme mostrado na figura 21.10.

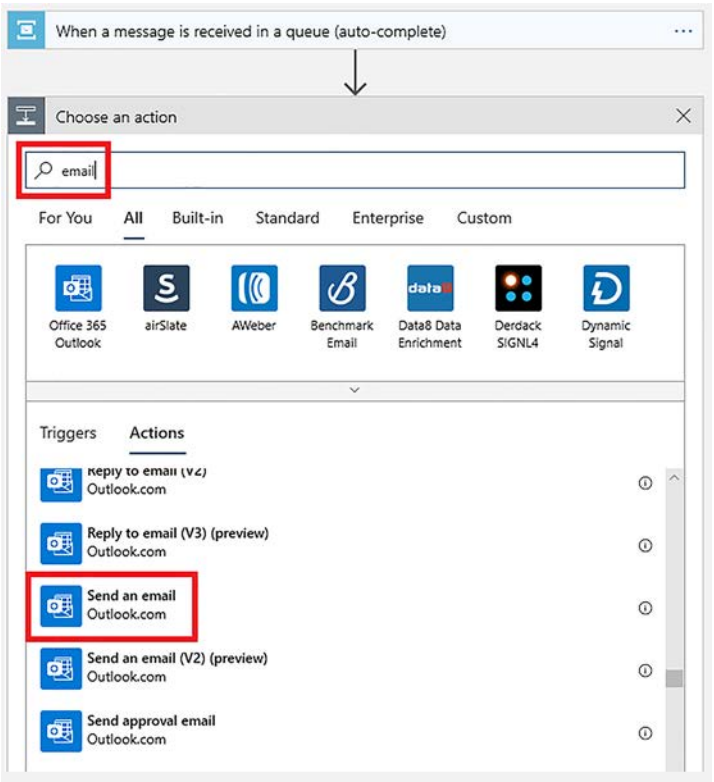


Figura 21.10 Pesquise e selecione seu provedor de e-mail atual, como o Gmail ou o Outlook.com. Você também pode escolher SMTP - Enviar um e-mail para configurar manualmente um provedor diferente.

12 Faça login no provedor de e-mail para autorizar o roteamento de e-mails e confirme que você deseja conceder permissões de aplicativos lógicos para enviar e-mails.

13 Informe um endereço do destinatário no qual você receba emails, um assunto, como `Alerta de temperatura`, e o corpo da mensagem, como `Alta temperatura detectada no dispositivo IoT`.

14 Salve o aplicativo lógico.

Vamos fazer uma pausa e revisar o que você criou nos últimos exercícios, como mostra a figura 21.11. Esse design básico de aplicação sem servidor não inclui nenhum controle que limite o número de mensagens a serem enviadas. Na aplicação lógico, você pode definir que deseja enviar no máximo cinco alertas por email e aguardar 30 minutos antes de enviar outros. Como parte de seu design de aplicação, você deve considerar como deseja receber notificações sobre situações como esta. Você também pode configurar o aplicativo lógico para ler os dados da mensagem da fila do barramento de serviço e incluir o registro de data e hora da mensagem do dispositivo IoT e a temperatura real registrada. Abordaremos como fazer isso no próximo exercício.

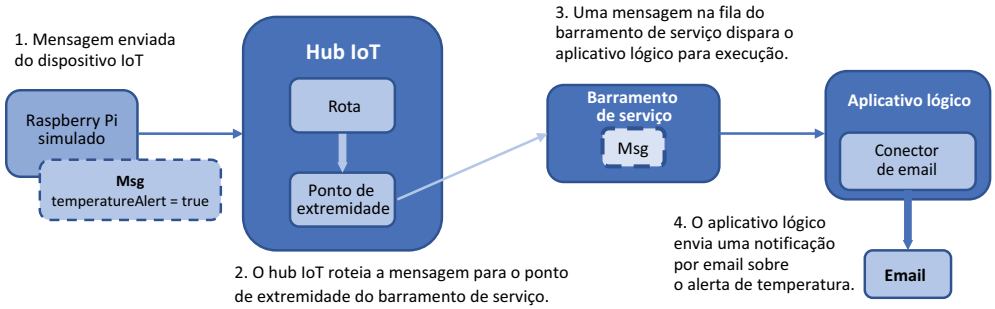


Figura 21.11 O dispositivo Raspberry Pi simulado envia uma mensagem ao hub IoT a cada dois segundos, que contém as leituras do sensor de temperatura. Se a temperatura estiver acima de 30° C, um alerta de temperatura será observado. O hub IoT encaminha qualquer mensagem que contenha um alerta de temperatura para uma fila de barramento de serviço. Mensagens nessa fila acionam um aplicativo lógico do Azure para ser executado. O aplicativo lógico é conectado a um provedor de email, como o Outlook ou o Gmail, e envia uma notificação por email sobre o aviso de temperatura do dispositivo IoT.

Vejamos esse aplicação básico sem servidor em ação.

Experimente agora

Para executar seu dispositivo Raspberry Pi simulado e testar seu aplicativo lógico, conclua as etapas a seguir:

1 Abra um navegador da Web para o dispositivo IoT Raspberry Pi simulado do capítulo 20 (https://azure-samples.github.io/raspberry-pi-web-simulator).

2 Verifique se a string de conexão do Hub IoT ainda está adicionada na janela de código que você configurou no capítulo 20.

3 Escolha Executar o aplicativo.

As leituras do sensor de temperatura e umidade simuladas são geradas a cada 2 segundos e uma mensagem é enviada para o hub IoT. Pode demorar algumas mensagens antes que uma leitura de temperatura simulada de 30° C seja gerada e exibida na janela de saída.

O hub IoT encaminha qualquer mensagem que contenha temperatureAlert: true para o ponto de extremidade do barramento de serviço. Como essas mensagens são colocadas na fila do barramento de serviço, o aplicativo lógico as coleta e envia um email por meio do provedor definido. Então, você recebe um email notificando sobre a leitura de uma temperatura alta. Esse processo deve levar apenas alguns segundos.

4 O dispositivo Raspberry Pi simulado gera mensagens a cada 2 segundos, então, a menos que você goste de receber muitos alertas de e-mail, pare o aplicativo!

Quando você recebe alertas por e-mail, a mensagem não contém muitas informações. Seu aplicativo lógico não extrai o conteúdo da mensagem do barramento de serviço nem formata as informações. Seria ótimo se o e-mail de alerta pudesse incluir o nome do dispositivo IoT ou a temperatura registrada. Como você pode processar cada mensagem e realizar algumas análises? E quanto ao outro serviço sem servidor do Azure que analisamos: aplicativos de função do Azure?

21.4 Criar um aplicativo de função do Azure para analisar dados do dispositivo IoT

Para estender sua aplicação sem servidor atual, você pode acionar uma aplicação de função do Azure em sua aplicação lógica. Os dados da mensagem do barramento de serviço podem ser enviados para um aplicativo de função para análise da temperatura registrada. A notificação por e-mail enviada pelo aplicativo lógico pode incluir informações sobre o nome do dispositivo IoT e a temperatura registrada. A interação entre o aplicativo lógico e o aplicativo de função é mostrada na figura 21.12.

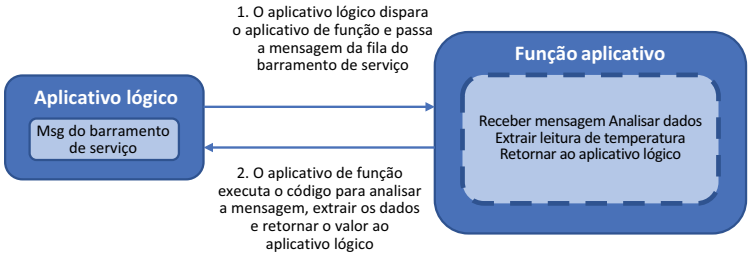


Figura 21.12 O aplicativo lógico acionará o aplicativo de função. A mensagem recebida na fila do barramento de serviço será passada para a função. O código no aplicativo de função analisa a mensagem, extrai a temperatura e retorna esse valor para o aplicativo lógico. Leva alguns milissegundos para o aplicativo de função executar esse código, portanto, o custo para executar essas tarefas de computação é equivalente a frações de um centavo.

Experimente agora

Para criar um aplicativo de função e ativá-lo a partir do aplicativo lógico, conclua as etapas a seguir:

1. No portal do Azure, selecione Criar um recurso na parte superior esquerda do menu.
2. Procure e selecione Aplicativo de Função e escolha Criar.
3. Selecione seu grupo de recursos, como `azuremolchapter21`, e atribua um nome, como `azuremol`. /convém ficar na mesma região que seus recursos anteriores.

   Você deseja publicar o código, também pode publicar uma imagem do contêiner do Docker (capítulo 19). Você nem precisaria criar uma instância de contêiner ou qualquer infraestrutura adicional; um contêiner de curta duração funcionaria conforme necessário e pararia.
4. Para esta aplicação básica, escolha o tempo de execução do Node.js, pois usamos alguns JavaScript simples.
5. Você tem três opções de plano de hospedagem. Um *Plano de consumo* permite que você pague por execução e os recursos necessários são dinamicamente atribuídos em tempo de execução. Para aplicações mais consistentes e prontos para produção, você pode usar um *plano Premium* ou *de hospedagem dedicada* que ofereça um custo mais fixo e previsível. Os planos Premium fornecem recursos adicionais, como proteger a conectividade para um conjunto definido de redes virtuais do Azure e sempre ter uma instância pronta para evitar alguns atrasos em um cenário de início frio para seu aplicativo. Para este exercício, escolha um plano de consumo.
6. Aceite os outros padrões para criar uma conta de armazenamento nomeada e os insights de aplicações. Depois, escolha Revisar + Criar.
7. Quando estiver pronto, crie o aplicativo de função. Leva um ou dois minutos para criar o aplicativo de função.
8. Quando o recurso for criado, selecione seu grupo de recursos, abra o aplicativo lógico do exercício anterior e selecione Editar.
9. No Designer de aplicativos lógicos, escolha Adicionar uma nova etapa.
10. Procure e selecione Funções do Azure e escolha a função criada nas etapas anteriores, como `azuremol`). Depois, escolha Criar uma nova função.
11. Atribua um nome à função, como `analyzeTemperature`.
12. Exclua qualquer código existente, substitua-o pelo código das listagens a seguir e escolha Criar.

    Esse código também está disponível no repositório do GitHub em w.

Listagem 21.1 Código JavaScript para um aplicativo de função `analyzeTemperature`

```
module.exports = function (context, req) {
   var buffer = new Buffer(req.body.ContentData, 'base64')
   var decodedString = buffer.toString();
   var objects = JSON.parse(decodedString);
   var temperature = objects["temperature"];
   context.res = {
      body: {
         analysis: "Recorded temperature was " + temperature + "!"
      }
   };
   context.log("Recorded temperature was " + temperature);
   context.done();
};
```

Cria a função. Cada aplicativo de função JavaScript começa com a exportação de uma função que contém um objeto de contexto. Esse objeto de contexto é usado para troca de dados.

Lê no conteúdo da mensagem do barramento de serviço

Cria um objeto JSON de mensagens de barramento de serviço decodificadas

Decodifica a partir de base64

Extrai a temperatura registrada do dispositivo IoT

Cria uma resposta para retornar ao Aplicativo Lógico

Emite a temperatura para o log do console

Encerra a função. Cada aplicativo de função JavaScript deve terminar com uma chamada para context.done, que informa ao aplicativo de função que seu código foi concluído.

13 De volta ao Designer de aplicativos lógicos para a etapa de função, selecione a caixa de texto Corpo da solicitação e escolha Mensagem do barramento de serviço na lista de conteúdo dinâmico no lado direito.

14 No Designer de aplicativos lógicos, arraste e solte para reorganizar as etapas para que a ação Enviar um e-mail fique abaixo da etapa do aplicativo de função `analyzeTemperature`, como mostrado na Figura 21.13.

15 Selecione a ação Enviar um e-mail e escolha a caixa de texto do corpo da mensagem de e-mail.

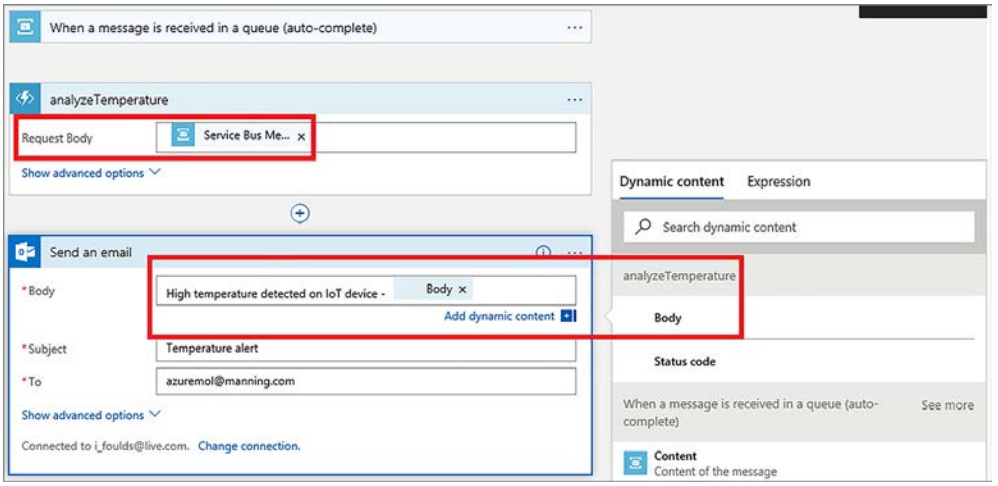


Figura 21.13 Arraste a ação Enviar um e-mail abaixo da função `analyzeTemperature`. Selecione o final da mensagem Corpo, assim, a caixa de diálogo Conteúdo dinâmico será exibida. Para inserir o valor de temperatura computado pelo aplicativo de função, selecione a mensagem Corpo na função `analyzeTemperature`.

16 `analyzeTemperature`, selecione a resposta Corpo, como mostrado na Figura 21.13.

17 No Designer de aplicativos lógicos, selecione Salvar.

Sua aplicação sem servidor tem muitas peças móveis. Vamos examinar o que você criou antes de executar o dispositivo IoT Raspberry Pi simulado para gerar alertas por e-mail que incluam a leitura da temperatura calculada pelo aplicativo de função. A Figura 21.14 fornece uma visão geral de todos os componentes agora em uso na aplicação sem servidor.

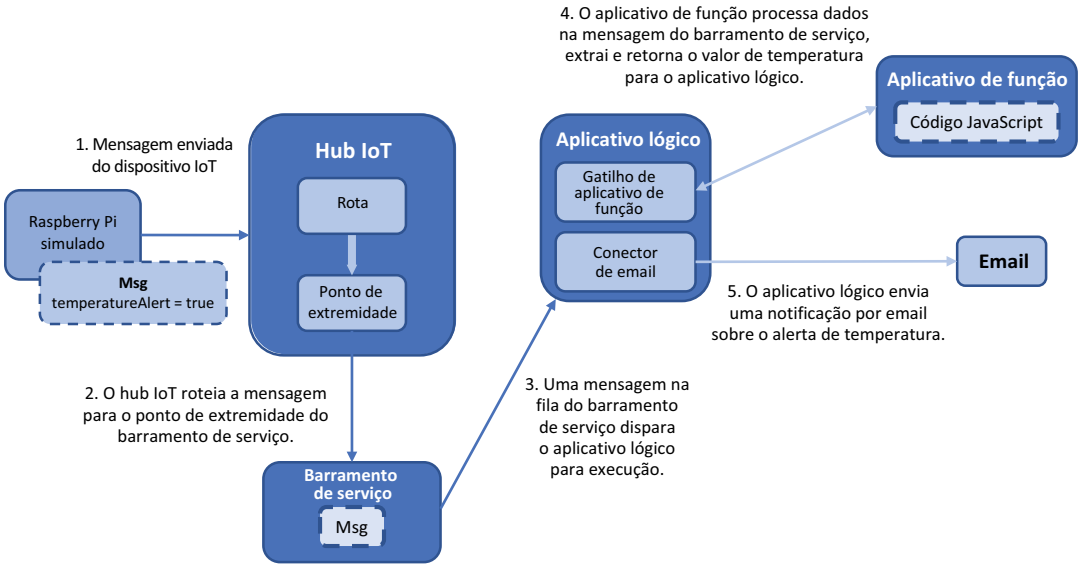


Figura 21.14 Conforme as mensagens são recebidas do dispositivo Raspberry Pi simulado, todas as mensagens que contêm um alerta de temperatura são roteadas para o ponto de extremidade da fila do barramento de serviços. As mensagens na fila do barramento de serviços acionam um aplicativo lógico, que passa a mensagem para um aplicativo de função. Uma função JavaScript analisa a leitura de temperatura e a retorna para o aplicativo lógico, que envia uma notificação por email que inclui a temperatura registrada por um sensor no dispositivo IoT.

18 Abra seu dispositivo Raspberry Pi simulado em um navegador da Web e execute a aplicação. Cada vez que o alerta de temperatura é gerado, o aplicativo lógico aciona o aplicativo de função para extrair os dados de temperatura do corpo da mensagem e incluí-lo na notificação por e-mail. Pode levar alguns instantes para uma leitura de temperatura ser superior a 30° C, que, após isso, sinaliza a mensagem com um alerta de temperatura. Quando esse alerta for enviado e a mensagem for processada, você receberá uma notificação por email que informa qual era a temperatura.

Muito bem! Agora respire fundo. Foi muita coisa para fazer no seu horário de almoço!

Erros de autenticação do seu aplicativo lógico para o aplicativo de função

Você pode ver o histórico de execução na janela Visão geral do seu aplicativo lógico no portal do Azure. Se você tiver vários erros repetidos, selecione um deles para ver mais sobre onde ele está ocorrendo.

Um problema comum é que o aplicativo lógico não está autorizado automaticamente a falar com o aplicativo de função. A reimplantação do aplicativo lógico geralmente corrige esse erro, mas a resolução real provavelmente é adicionar o que chamamos de chave de *função* para o cabeçalho do seu aplicativo lógico.

Para obter essa chave, selecione seu aplicativo de função e escolha a função que você criou, como `analyzeTemperature`. Na opção Gerenciar, a chave de função padrão pode ser exibida e copiada. Copie essa chave, volte para o seu aplicativo lógico e abra o designer.

Na função `analyzeTemperature`, escolha Adicionar um parâmetro e, em seguida, adicione um cabeçalho. Você deseja enviar informações no início da chamada para o aplicativo de função que envia a chave. O processo é um pouco atrasado à medida que você insere o par de chaves, mas insira `x-functions-key` para a chave e, em seguida, cole sua chave de função como o valor.

Leva alguns instantes para atualizar a integração do aplicativo lógico com o de função. Depois disso, o histórico de execuções do aplicativo lógico deve mostrar os eventos funcionando corretamente, e as notificações por email devem começar a ser entregues.

21.5 Não pare de aprender

Esse capítulo continha muitos conceitos novos. De fato, os últimos capítulos continham muitas novas ideias e tecnologias! Não se preocupe se está com dificuldades para entender como pode começar a implementar todos esses serviços do Azure, como contêineres, IA e ML e computação sem servidor. Esses capítulos mostraram o que é possível fazer no Azure e que você não precisa se limitar a realizar um lift-and-shift de aplicações herdadas. À medida que você começa a criar e executar aplicações no Azure, aproveite a oportunidade para modernizar aplicações e analisar fluxos de trabalho de gerenciamento ou implantação. Há muitos serviços do Azure que simplificam e aceleram o ciclo de vida da aplicação, por isso, não pense que você tem que continuar com as VMs em execução, por isso ser o que a empresa costuma usar.

Sim, o Azure oferece muitos serviços novos e brilhantes, mas todos eles são baseados em componentes de infraestrutura básica que foram abordados na parte 1 desse livro. Os desenvolvedores podem começar a usar as abordagens de design de aplicação mais recentes que envolvem Kubernetes ou computação sem servidor, e os administradores podem reutilizar seu conhecimento de data center na infraestrutura local com os fundamentos da computação na nuvem e as técnicas de solução de problemas. Conforme suas necessidades de negócios crescem, o Azure é capaz de oferecer suporte a elas.

No capítulo 1, fui sincero e honesto ao afirmar que não abordaria todos os serviços no Azure. Há muito mais serviços do Azure para aprender, e você pode se aprofundar muito mais do que fizemos neste livro. Espero que você tenha encontrado pelo menos algumas áreas que interessem a você e o motivem a explorar um pouco mais. Meus favoritos incluem conjuntos de escalas de máquinas virtuais, o Cosmos DB e o Serviço Azure Kubernetes.

21.5.1 Materiais de aprendizagem adicionais

Sou tendencioso, mas acho que um ótimo lugar para continuar aprendendo sobre o Azure é https://docs.microsoft.com/azure. Esta página da Web contém as principais documentações de serviço do Azure, guias de arquitetura, recursos de referência e SDK e exemplos. Cada serviço do Azure tem seu próprio conjunto de iniciações rápidas, tutoriais e exemplos, além de informações conceituais e guias de instruções individuais.

Se você levar o conteúdo a sério, existem opções de certificação para o Azure. Os exames individuais incluem *Microsoft Azure Administrator (AZ-104), Microsoft Azure Architect Technologies and Design (AZ-303 and AZ-304)* e *Microsoft Azure Security Technologies (AZ-500)*. Este livro e os exercícios de laboratório que você concluiu cobriram muitas das áreas em que os exames testam seu conhecimento, mas você precisa estudar algumas áreas adicionais do Azure AD e criar práticas recomendadas antes de fazer os exames. O site Microsoft Learn em https://docs.microsoft.com/learn tem mais alguns caminhos de aprendizagem para as diferentes opções de certificação do Azure para ajudá-lo a se preparar.

21.5.2 Recursos do GitHub

Ao longo deste livro, você usou exemplos de código, modelos e exemplos de aplicações de https://github.com/fouldsy/azure-mol-samples-2nd-edition. Esses exemplos devem permanecer atualizados à medida que novas versões da CLI do Azure são lançadas e o repositório do GitHub também inclui exemplos e modelos do PowerShell para todos os exercícios. Esse livro se concentra na CLI do Azure no Azure Cloud Shell, mas fique à vontade para explorar como é cada exercício no PowerShell ou em um modelo.

Se você notar algum problema com os exemplos, crie um problema no GitHub em https://github.com/fouldsy/azure-mol-samples-2nd-edition/issues. As coisas avançam rapidamente no Azure, e quero garantir que você sempre tenha os exemplos mais recentes e úteis para aprender. Sinta-se à vontade para fazer sugestões também! Todos os documentos do Azure em https://docs.microsoft.com/azure também aceitam comentários e edições. Assim, ao explorar o restante do que é oferecido no Azure, sinta-se à vontade para se envolver e ajudar os outros a aprender e crescer.

21.5.3 Uma consideração final

Respire fundo e lembre-se de que a mudança é normal. Novos recursos e serviços são lançados quase diariamente. O Azure, como todos os principais provedores de computação na nuvem, pode parecer e estar um pouco diferente da última vez que você usou (há uma hora). Se você tiver as principais habilidades fundamentais e a compreensão que acredito que tenha obtido nesse livro, poderá se adaptar e crescer com todas as novas oportunidades oferecidas pelo Azure. Você sempre terá algo novo para aprender, e eu adoraria ouvir o que você acabará criando e executando no Azure!

índice

Símbolos

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

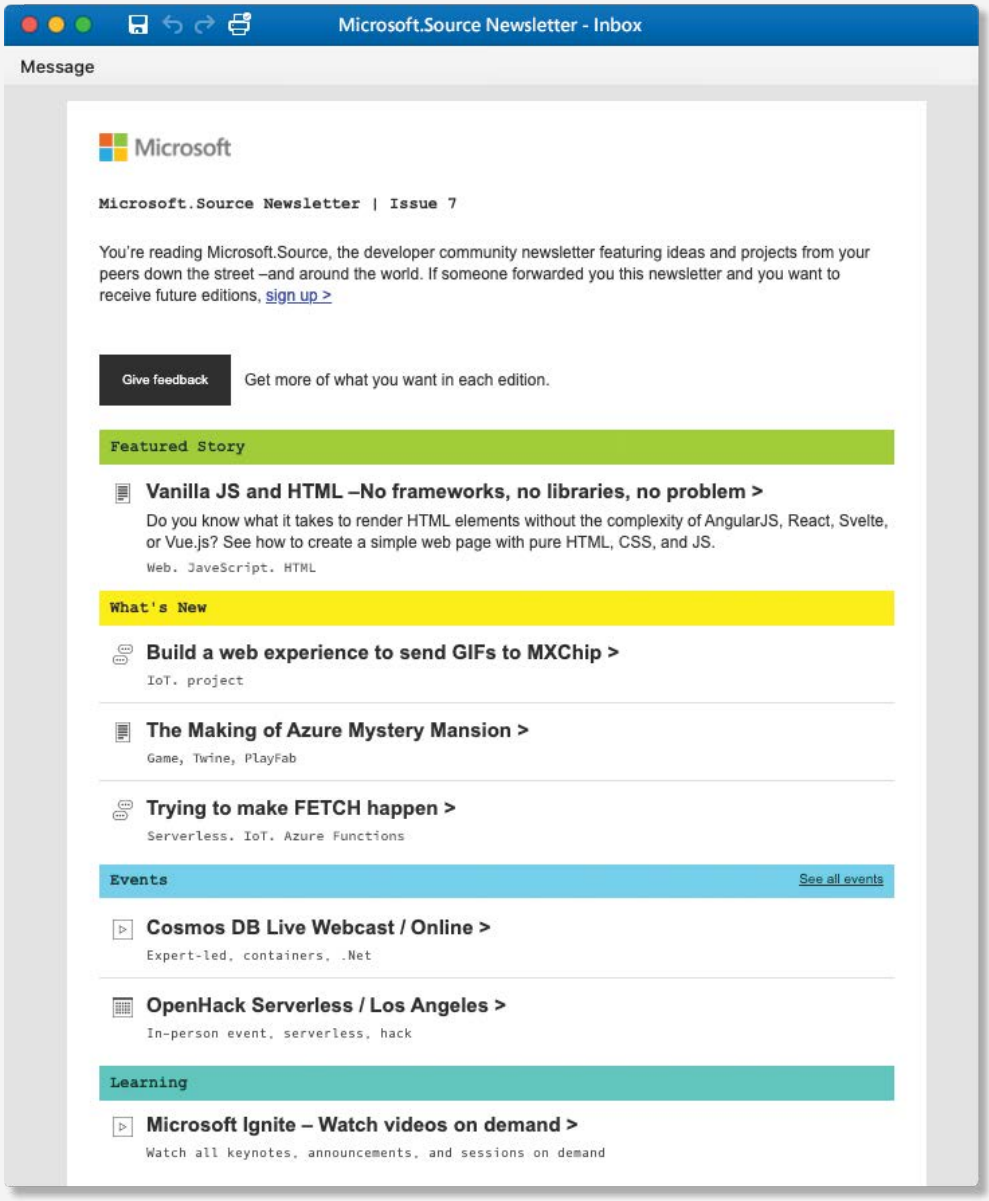
U

V

W

Z

Microsoft



Por desenvolvedores, para desenvolvedores

Newsletter Microsoft.Source

Receba artigos técnicos, código de exemplo e informações sobre os próximos eventos no Microsoft.Source, o boletim informativo mensal da comunidade de desenvolvedores.

- Acompanhe as tecnologias mais recentes
- Conecte-se com colegas em eventos da comunidade
- Aprenda com recursos práticos

Inscreva-se