

O guia do desenvolvedor do Azure

Segunda Edição



PUBLICADO POR
Microsoft Press
Uma divisão da Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2018 pela Microsoft Corporation

Todos os direitos reservados. Nenhuma parte do conteúdo deste livro pode ser reproduzida nem transmitida de qualquer forma ou por qualquer meio sem a permissão por escrito da editora.

Os livros da Microsoft Press estão disponíveis por meio de livreiros e distribuidores em todo o mundo. Se precisar de suporte referente a este livro, envie um email ao Suporte da Microsoft Press em mspinput@microsoft.com. Dê sua opinião sobre este livro em <http://aka.ms/tellpress>.

Este livro é fornecido “no estado” e expressa os pontos de vista e as opiniões do autor. Os pontos de vista, as opiniões e as informações expressados neste livro, inclusive URLs e outras referências a sites, podem ser alterados sem aviso prévio.

Alguns exemplos descritos aqui são fornecidos apenas para ilustração e são fictícios. Nenhuma associação ou conexão real é intencional nem deve ser inferida.

A Microsoft e as marcas listadas em <http://www.microsoft.com> na página da Web “Marcas” são marcas do grupo de empresas Microsoft. Todas as outras marcas pertencem a seus respectivos proprietários.

Sumário

Capítulo 1: O guia do desenvolvedor do Azure	1
Introdução	2
A tecnologia da informação como um utilitário, não um fardo.....	2
A computação como um utilitário	2
Tipos de implantação de computação na nuvem	4
A nuvem pública.....	4
A nuvem privada	4
A nuvem híbrida	4
Desmistificar os tipos de computação na nuvem.....	5
Infraestrutura como serviço	5
Plataforma como serviço	6
Funcionamento como serviço.....	6
Software como serviço.....	6
Por que Azure?.....	7
O Azure tem alcance global.....	8
O Azure é extremamente resiliente	9
O Azure está em conformidade com quase todas as indústrias.....	9
O Azure se concentra em recursos para desenvolvedores	10
O Azure é aberto e dá suporte às estruturas de sua escolha.....	10
Você pode monitorar seus serviços do Azure em qualquer lugar	10
Estamos aqui se você precisar de ajuda	10
Capítulo 2: Introdução ao Azure	11
Seus IDEs e editores são bem-vindos.....	11
Use a interface de linha de comando do Azure para scripts	11
Execute qualquer coisa no Azure	12
E quanto aos custos?	12
Selecionar os serviços certos do Azure	12

Onde hospedar seu aplicativo	12
O que usar quando?	18
Tornar seu aplicativo mais rápido	19
Rede de Distribuição de Conteúdo do Azure	19
Cache Redis do Azure	19
Gerenciador de Tráfego do Azure	20
Onde armazenar seus dados	21
Capítulo 3: Adicionando inteligência ao aplicativo	26
Azure Search	26
Serviços Cognitivos do Azure	27
Serviço de Bot do Azure	27
Usar eventos e mensagens em seu aplicativo	28
Barramento de Serviço do Azure	28
Hubs de Eventos do Azure	29
Hub IoT do Azure	29
Grade de Eventos do Azure	30
O que usar quando?	31
Capítulo 4: Proteger seu aplicativo	32
Azure Active Directory	32
Gerenciamento de API do Azure	32
Azure Key Vault	33
Proteção contra DDoS do Azure	33
Firewall do Aplicativo Web do Azure	33
Criptografia padrão de dados	34
Capítulo 5: Onde e como implantar seus serviços do Azure	35
A mentalidade de DevOps	35
Modelos do Azure Resource Manager	36
Azure Service Fabric	36
Contêineres no Azure	37
Azure Stack	38
Onde implantar e quando?	38
Capítulo 6: Um passo a passo do Azure	39
Passo a passo nº1: o Portal do Azure	39
Blocos no Portal do Azure	39
Criar uma nova máquina virtual	41
Passo a passo nº 2: desenvolver um aplicativo Web Node.js com o MongoDB no Azure	45
Criar o aplicativo Web e o banco de dados usando o Portal do Azure	45
Passo a passo nº3: criar um back-end para seu aplicativo móvel com o Azure	53

Criar um aplicativo para dispositivos móveis por meio do Portal do Azure	54
Recursos adicionais e alteração para a produção	58
Capítulo 7: Usar o Azure Marketplace.....	59
Como o Azure Marketplace pode me ajudar como desenvolvedor?	59
Soluções do Azure.....	60
Resumo e o que fazer depois	60
Continue aprendendo com uma conta gratuita do Azure.....	60
Sobre os autores	61

O guia do desenvolvedor do Azure

Hoje em dia, a demanda por software e serviços correlatos é maior do que nunca. Essa tendência continuará a crescer exponencialmente, à medida que aplicativos de todos os tipos se tornarem cada vez mais presentes nos vários aspectos da vida diária. Como desenvolvedor, você é o “mágico” que leva essa revolução digital aos usuários, e isso não é fácil. Além de criar os recursos tangíveis exclusivos do software, há muito a fazer nos bastidores. Os aplicativos precisam ter alta disponibilidade, oferecer ótima performance em todo o mundo, funcionar perfeitamente em uma vasta gama de dispositivos, ser seguros e permitir a execução em grande escala a um preço razoável. Esses desafios são comuns à maioria dos softwares e das indústrias. No entanto, as soluções são efêmeras e constantemente reinventadas, e sua criação e manutenção podem ser difíceis e caras.

Criamos o *Guia do Desenvolvedor do Azure* para ajudá-lo a migrar para a nuvem, quer você ainda esteja na fase de planejamento ou já tenha iniciado o processo. Este e-book foi escrito por e para desenvolvedores. Sua finalidade específica é oferecer a você, como desenvolvedor, conhecimentos essenciais sobre o Azure, o que ele oferece a você e à sua organização e como aproveitá-lo plenamente.

A segunda parte do guia abrange cenários como um tour pelo Portal do Azure e a criação de uma máquina virtual. Também abordamos o desenvolvimento e a implantação de um aplicativo Web que usa o Node.js e o MongoDB. Examinamos tarefas típicas, como CI/CD (integração e implantação

contínuas), ambientes de preparo, escala, registro em log e monitoramento. Para terminar, apresentamos a criação de um back-end para um aplicativo móvel que inclui autenticação e sincronização offline.

Introdução

Há inúmeros livros, artigos, papers, posts de blog e outras referências que divulgam as vantagens da computação na nuvem. Você pode encontrar zettabytes de informações que explicam a computação na nuvem em termos técnicos. Porém, antes de se aprofundar, você precisa fazer algumas perguntas simples: por que mudar para a nuvem? Que benefícios ela trará para mim, a equipe e a empresa?

A tecnologia da informação como um utilitário, não um fardo

A computação na nuvem é equivalente a um utilitário. O que isso significa? Para ajudá-lo a entender, pense em algo que está sempre presente em sua vida: a eletricidade.

Hoje em dia, ninguém acha que uma usina geradora para cada casa ou empresa seria uma forma eficiente de distribuição de eletricidade. Estamos acostumados a ligar o interruptor para acender a luz e realizar nossas tarefas. Você nem sonharia em comprar, instalar e manter todo o equipamento necessário para iluminar a casa ou o escritório. Em vez disso, você paga à companhia elétrica local pela energia que consome por mês. Nunca precisa se preocupar com nada: produção de eletricidade, despesa de capital, manutenção do equipamento ou reparos após tempestades. Basta ligar o interruptor e se concentrar em outras atividades.

Assim como você não precisa nem quer saber como a eletricidade é fornecida à sua casa, também não precisa saber como os recursos e serviços de computação consumidos na nuvem são disponibilizados. Outra pessoa se encarrega disso para você. Antes de mais nada, a migração dos recursos de computação, armazenamento e rede para a nuvem possibilita a facilidade de uso, como acender uma lâmpada. Você só paga pelo que usa, assim como só paga pela eletricidade que usa em casa.

Além dessa analogia, a nuvem oferece muitos benefícios mais específicos para as operações de TI. Por exemplo: escalabilidade e capacidade de usar serviços inteligentes como pesquisa reconhecimento facial nos aplicativos, para citar apenas duas opções.

A computação como um utilitário

Em última análise, serviços de nuvem como o Azure equivalem à *computação como um utilitário*. Em termos de produtos e serviços, o Azure oferece muito mais do que a companhia elétrica local. Portanto, às vezes é difícil definir claramente a computação na nuvem. Dizem até que a nuvem é como usar o *computador de outra pessoa*. Tecnicamente, é verdade, mas a nuvem é bem mais do que isso:



A nuvem é um sistema que fornece **serviços robustos, resilientes e inteligentes** e recursos de **computação** em **escala infinita, elástica e global**.

Vamos explicar tudo em detalhes:

A nuvem e o Azure oferecem serviços que ajudam a realizar muitas tarefas. Isso inclui desde as mais triviais, como adicionar pesquisa a um aplicativo, até as mais exóticas, como implementar fluxos de trabalho de CI (Integração Contínua) e CD (Implantação Contínua). Ajuste automaticamente um banco de dados ou configure notificações por push para dispositivos móveis, com rapidez e facilidade. Esses são apenas alguns exemplos de itens comuns que os desenvolvedores criavam repetidamente para si mesmos, mas que agora estão disponíveis como um serviço. Assim, é possível usar esses serviços com bem pouco esforço, quase como ligar o interruptor de luz! Você pode se concentrar no que torna o aplicativo único: os recursos que agregam valor real para os usuários.

Além de serviços, a nuvem oferece recursos de computação na forma de VMs (Máquinas Virtuais), contêineres, bancos de dados e assim por diante. Você pode usá-los para hospedar aplicativos ou fornecer uma infraestrutura completa aos usuários.

Como mencionamos, a grande vantagem da nuvem é que os serviços e recursos são incrivelmente robustos e resilientes. É pouco provável que eles tenham problemas de execução. Isso ocorre porque a nuvem é inteligente. Ela corrige automaticamente os problemas. Além disso, assim como ocorre com o Azure, há datacenters em todo o mundo, com dezenas de milhares de servidores. Se um servidor falha, outro assume o controle. Se um datacenter inteiro falhasse (algo extremamente improvável), outro funcionaria em seu lugar. Tudo isso é possível graças à enorme escala da nuvem.

Um dos argumentos mais convincentes para a adoção da nuvem é que você pode expandir os serviços e recursos de forma quase infinita. Isso certamente seria impossível com os recursos na infraestrutura local, a menos que você esteja disposto a gastar enormes somas em equipamento e pessoal para administrar tudo. Além disso, é possível escalar de forma global. Você pode disponibilizar seus serviços em qualquer lugar do mundo. Assim, pode fornecer uma experiência de alta performance aos usuários, onde quer que estejam. Isso também significa que você pode manter os dados onde for necessário.

O mais importante: ao usar recursos da nuvem, você pode reduzir os serviços e recursos quando a demanda diminui. Voltando à analogia com a eletricidade: se você convidar vários amigos e parentes para uma festa em casa, acenderá muitas luzes, usará o micro-ondas com frequência, tocará música e consumirá mais eletricidade. Quando a festa acabar e os convidados forem embora, o consumo de energia voltará ao normal, assim como sua fatura. Quando você usa a nuvem e o Azure, ocorre o mesmo: você paga apenas pelo que usa, não pelo que poderá ser necessário quando o uso for maior.



Mais informações Para saber mais sobre o Portal do Azure e criar sua primeira VM, acesse <https://docs.microsoft.com/azure/virtual-machines/windows/quick-create-portal>.

Além da grande escalabilidade, dos serviços inteligentes prontos para o uso e da eficiência do pagamento pelo uso, a nuvem oferece maior segurança.

A nuvem é usada por milhões de pessoas em todo o mundo, 24x7. Claro, também é atacada por muitas pessoas. Provedores de nuvem conceituados e experientes como a Microsoft conhecem os padrões de uso dos usuários normais, diferenciando-os de invasores mal-intencionados. Isso significa que ela pode garantir a proteção contra os ataques mais comuns e os mais avançados. Ferramentas de monitoramento inteligentes, algoritmos de aprendizado de máquina e inteligência artificial permitem aos provedores de nuvem detectar os ataques em tempo real e detê-los imediatamente.

[Décadas de experiência em segurança](#) e tráfego de grande escala, aliados à maior experiência na indústria de segurança, tornam a nuvem um ambiente muito mais seguro do que qualquer datacenter na infraestrutura local.



Mais informações Para ler mais sobre como o Azure protege aplicativos e dados, acesse [Como a Central de Segurança do Azure detecta ataques de DDoS usando inteligência de ameaças cibernéticas, Visão geral da segurança do Azure e Como a Microsoft lida com os crimes cibernéticos](#).

Analizamos resumidamente os motivos para iniciar a migração para a nuvem e o Azure. Agora, vamos examinar os principais componentes da nuvem, o que eles fazem e como se encaixam no panorama geral.

Tipos de implantação de computação na nuvem

A nuvem oferece vários níveis de produtos e serviços, como IaaS (infraestrutura como serviço) e PaaS (plataforma como serviço). Antes de abordá-los, vamos falar dos três tipos básicos de plataformas de nuvem: *pública*, *privada* e *híbrida*. Elas logo foram incorporadas ao jargão da tecnologia moderna. Vamos desmistificá-las.

A nuvem pública

No modelo de nuvem pública, você não possui hardware. O provedor de nuvem é responsável pela compra e manutenção de todo o hardware. A nuvem pública consiste em todos os serviços e recursos de computação que você não possui, mas que usa por meio do provedor de nuvem. O provedor é responsável por manter tudo funcionando e em conformidade com os SLAs (Contratos de Nível de Serviço). *Você paga pelo que usa, não pelo que possui.*

A nuvem privada

Com a nuvem privada, você possui todo o hardware ou, pelo menos, tem controle total sobre ele. O hardware que executa seus serviços e armazena os dados está em algum lugar no datacenter na infraestrutura local. Claro, o controle total tem preço: você deve comprar e manter tudo. *Você paga pelo que possui, não pelo que usa.*

Porém, como configurar uma plataforma de nuvem privada em um datacenter tradicional na infraestrutura local? Felizmente, o Azure fornece uma solução exclusiva, projetada especificamente para esse ambiente: o Azure Stack. Em resumo, o Azure Stack é sua própria instância privada de Azure em uma caixa que você pode executar no datacenter na infraestrutura local.

Executar serviços em uma nuvem privada não é o mesmo que executá-los no datacenter na infraestrutura local. O [Azure Stack](#) oferece todas as vantagens do Azure, inclusive serviços inteligentes que você pode usar em aplicativos, tudo isso no datacenter na infraestrutura local. Se, por razões especiais de segurança ou governança, você precisar manter determinadas operações no datacenter na infraestrutura local, ainda poderá desfrutar de todos os benefícios oferecidos pela Azure e pelo paradigma de nuvem.

A nuvem híbrida

Como o nome indica, o modelo de nuvem híbrida é uma mistura entre a nuvem pública e a privada. Você pode usar serviços de nuvem pública que utilizam os recursos da nuvem privada e vice-versa. No Azure, você pode executar um aplicativo em um Aplicativo Web do Azure que se conecta a um banco de dados na infraestrutura local usando Conexões Híbridas do Azure. Isso abre muitas possibilidades. Você controla onde os aplicativos e os dados estão, mas conta com os benefícios do uso de serviços de nuvem inteligentes.

O Azure fornece soluções para todos os tipos de implantação de computação na nuvem. Isso dá às empresas mais liberdade para usar o Azure onde e como quiserem. É possível determinar o nível de controle que desejam ter sobre aplicativos e dados.

Desmistificar os tipos de computação na nuvem

Na computação na nuvem, categorizamos serviços e recursos como tipos. Há muitas definições diferentes de tipos de computação na nuvem. As diferenças entre os tipos são o nível de controle e as responsabilidades que você versus o tempo e o esforço que pode gastar para agregar valor de negócios para a empresa ou os clientes. A Figura 1-1 e a Figura 1-2 ilustram como definimos os tipos de computação na nuvem para o Azure.

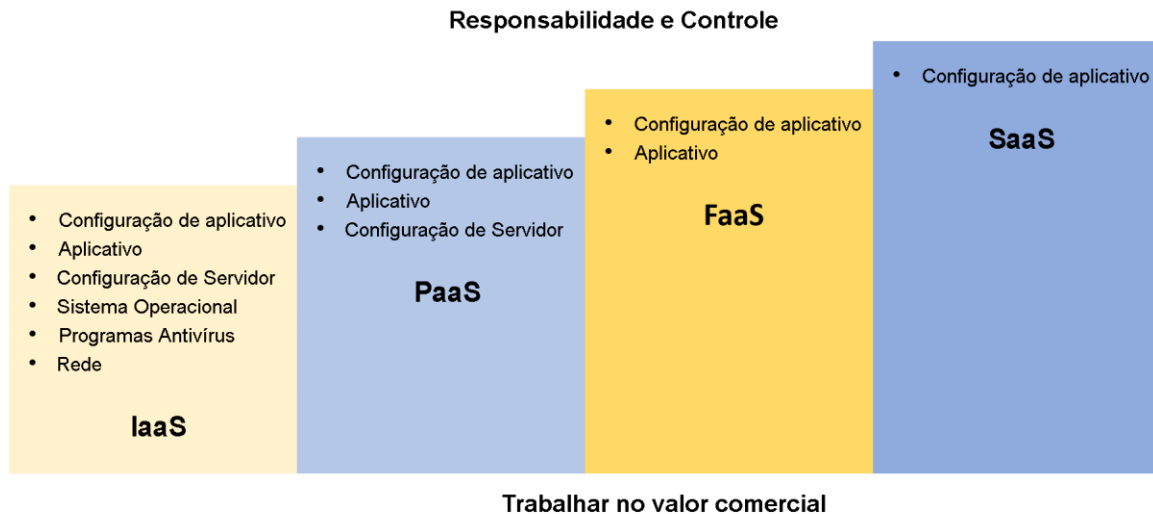


Figura 1-1: explicação dos tipos de computação na nuvem

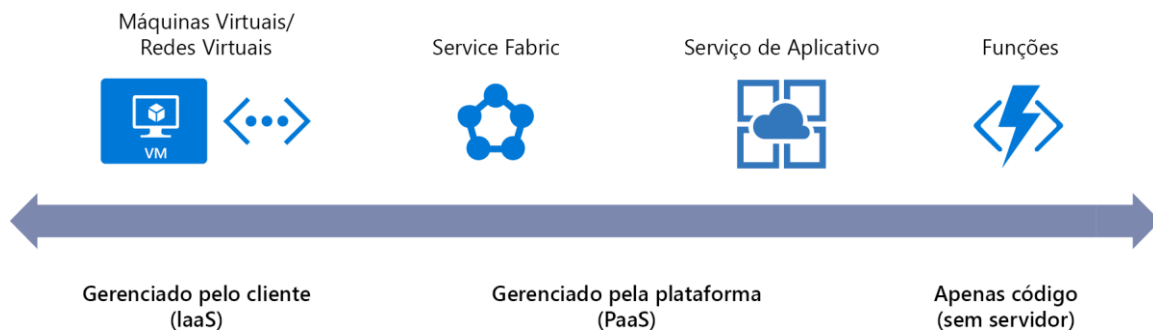


Figura 1-2: exemplos de ofertas baseadas na nuvem

Infraestrutura como serviço

Com a IaaS (infraestrutura como serviço), você é responsável por tudo, *exceto* o hardware. O Azure cuida de todo o hardware, como SSDs (unidades de estado sólido), cabeamento de rede, roteadores, alimentação, geradores de reserva, sistemas de refrigeração e assim por diante. Os serviços que se enquadram nessa categoria incluem [Máquinas Virtuais do Azure](#), [Redes Virtuais do Azure](#) e [Contêineres do Azure](#).

Você cuida de todo o resto. O aplicativo, patches do SO (sistema operacional), configuração de rede lógica e até mesmo atualização de programas antivírus nos computadores. Isso significa que você mantém controle total sobre esses elementos. No entanto, também significa que você gasta muito tempo e esforço para manter os serviços e recursos em execução e menos tempo trabalhando para agregar valor de negócios aos aplicativos principais.

Plataforma como serviço

Usando o modelo PaaS (plataforma como serviço), você é responsável pelo aplicativo e sua configuração. O Azure fornece o sistema operacional e o hardware. Como demonstra a Figura 1-1, o PaaS é um nível de abstração mais elevado do que o IaaS. Você tem bem menos responsabilidades, mas menos controle direto sobre os recursos de hardware do que com o IaaS. Os serviços que se enquadram na categoria PaaS incluem o [Serviço de Aplicativo do Azure](#), os [Bancos de Dados SQL do Azure](#), o [Cache Redis do Azure](#) e o [Lote do Azure](#).

Com PaaS, você pode hospedar seu aplicativo, mas esse nível também oferece recursos adicionais que estão prontos para o uso, como notificações por push para dispositivos móveis via Aplicativo Móvel do Azure e ajuste automático de seu banco de dados no Banco de Dados SQL do Azure.

Você implanta seu aplicativo e o configura. Você também faz a configuração básica de servidor, como escalonamento da camada de preços ou o número de instâncias em que o aplicativo é executado. O aplicativo ainda é executado em um servidor em algum lugar, mas você não é responsável por esse servidor (nem tem controle sobre ele). Isso significa que você pode gastar muito mais tempo concentrando-se em agregar valor de negócios aos aplicativos e menos tempo mantendo um servidor em execução.



Experimente Crie um aplicativo do Node.js com MongoDB no Azure neste passo a passo em <https://docs.microsoft.com/azure/app-service/app-service-web-tutorial-nodejs-mongodb-app>.

Funcionamento como serviço

Na função como serviço (FaaS), você é responsável apenas por seu aplicativo e sua lógica de negócios. Você não precisa se preocupar com escala, sistema operacional nem qualquer hardware. O FaaS às vezes é chamado de *computação sem servidor*. Os serviços na categoria FaaS são [Funções do Azure](#), [Aplicativos Lógicos do Azure](#) e [Grade de Eventos do Azure](#).

No FaaS, basta criar seu aplicativo ou reunir a lógica de negócios e executá-lo. A escala acontece automaticamente. Um dos benefícios mais notáveis do FaaS é que você paga pelos recursos que usa somente quando a lógica é executada, em vez de pagar por um serviço que está sempre ativado, esperando que alguém o utilize.

Porém, o FaaS é mais do que apenas hospedar o aplicativo. Ele conecta a lógica a gatilhos e fontes. Fontes externas, como Filas de Armazenamento do Azure ou WebHooks, podem iniciar Funções do Azure ou Aplicativos Lógicos e fornecer-lhes dados para processar. Você não precisa escrever a lógica de ativação nem configurar a estrutura para se conectar a repositórios de dados ou serviços externos. Ou seja, você pode se concentrar na lógica e nos recursos voltados para o cliente.

Software como serviço

Com o software como serviço (SaaS), você só precisa se preocupar em configurar o software, nada mais. De fato, o SaaS é o mais alto nível de abstração dos tipos de computação na nuvem. Os serviços na categoria SaaS incluem [Serviços Cognitivos do Azure](#) e [Azure IoT Suite](#). O [Office 365](#) é outro exemplo de um pacote de aplicativos SaaS do Azure, oferecendo aplicativos de negócios confiáveis, como Microsoft Word, Excel, PowerPoint e muito mais. Tudo isso está disponível sem ter de instalar ou manter nada.

Com o SaaS, o software está pronto para usar. Basta configurá-lo conforme suas preferências e você está pronto para começar. Você não escreve o software e não precisa pensar em implantação, escala, sistemas operacionais e hardware.

Por que Azure?

Você está pensando que a nuvem pode ser aquilo de que precisa para sua empresa criar produtos de software incríveis e agregar o valor e a experiência do usuário que seus clientes esperam. Certamente, a plataforma de nuvem do Azure pode ajudá-lo a conseguir isso. Com seus vastos recursos e poder de computação, além de um amplo leque de serviços inteligentes, você pode criar aplicativos altamente escalonáveis e dinâmicos e só pagar pelo que usar. Porém, existem muitos fornecedores de nuvem hoje em dia, cada um oferecendo produtos e recursos exclusivos. Então, novamente, por que o Azure?

Com o Azure, você obtém o profundo conhecimento e a experiência da Microsoft, uma empresa que se mantém na vanguarda da computação pessoal e corporativa e das soluções de TI há mais de 40 anos. A Microsoft projeta, cria e executa operações de datacenter e redes de grande escala para seus próprios fins há décadas. Essa experiência e conhecimento estão incorporados ao Azure. São sua própria essência.

Quem usa o Azure?

[Adobe](#), [Jet.com](#), [Geico](#), [GeekWire](#) e [DocuSign](#) são apenas alguns exemplos de clientes no mundo inteiro que recorrem à plataforma Azure para ajudá-los a atingir seus objetivos de negócios e agregar valor para os clientes. Alguns desses objetivos incluem a solicitação de serviços de aplicativo para opções sem servidor, inteligência artificial, Azure IoT e até mesmo streaming de mídia. Vejamos algumas maneiras interessantes pelas quais os clientes usaram a plataforma Azure para atender às suas necessidades de negócios.

A gigante industrial [Samsung](#), da Coreia do Sul, cria tudo, desde telefones, televisores, máquinas de lavar roupa e grandes eletrodomésticos até condicionadores de ar. A empresa decidiu usar o Azure para aumentar a eficiência e a robustez de seus condicionadores de ar, realizando manutenção e monitoramento remotos. A empresa usa Hubs IoT do Azure para coletar enormes quantidades de dados enviados de sensores integrados a seus condicionadores de ar, que monitoram o status das máquinas, bem como o ambiente em que elas funcionam. Os dados são capturados no SQL do Azure para análise em tempo real. Os engenheiros podem então usar as informações que extraem para ajustar o comportamento dos condicionadores de ar, resultando em uma economia de energia de 15 a 18%, em média, para seus clientes.

A [Coca-Cola](#) Company, uma das marcas mais reconhecidas do mundo, introduziu pela primeira vez o refrigerante de mesmo nome em 1886 e é uma empresa líder mundial desde então. Em 2012, a empresa lançou o Coco-Cola Journey, um projeto e site altamente ambiciosos, cujo objetivo é ajudar a Coca-Cola a se conectar melhor com uma nova geração de clientes que, cada vez mais, obtêm sugestões de compras de canais de marketing não tradicionais. A empresa quis aproveitar a mídia social e um serviço da Microsoft chamado How-Old.net para criar uma experiência interativa divertida e envolvente para os usuários, além de obter insights sobre seus clientes ao mesmo tempo. Com base na tecnologia de Serviços Cognitivos, o How-Old.net analisa dados extraídos de imagens faciais para estabelecer a idade e o sexo dos indivíduos. Porém, a Coca-Cola convidou a equipe da Microsoft para levar o projeto um passo além do reconhecimento facial, usando modelos de aprendizado de máquina para detectar objetos individuais em fotos. Por um mês, o site convidou os usuários a enviar uma foto de si mesmos e uma garrafa de Coca-Cola com um logotipo visível para ver o que aconteceria. Se o aplicativo Web detectasse uma garrafa, a idade do frasco apareceria, e a página ficaria vermelha. Então, os fãs da Coca-Cola foram incentivados a compartilhar sua experiência na mídia social e continuar a explorar o site Coca-Cola Journey para saber mais sobre a famosa garrafa.

A [CarMax](#), maior varejista de carros usados nos Estados Unidos, reformulou seu site usando os serviços da plataforma Azure para fornecer uma resposta mais rápida aos 16 milhões de pessoas que acessam seu site todos os meses. Em vez de apenas mudar o visual do site para um ambiente do Azure IaaS, a CarMax optou por reconstruir o site usando a oferta do Azure PaaS. Assim, a CarMax pôde modernizar o site com uma nova funcionalidade de nuvem, torná-lo 100% compatível com celulares e, o mais importante, mudar para um paradigma de desenvolvimento de DevOps dinâmico. A CarMax está mudando rapidamente para uma arquitetura de microsserviços, que dará à empresa a capacidade de escalonar e atualizar partes de aplicativos independentemente de outras partes. Isso levará anos para ser concluído, mas a escolha do Azure como a base de desenvolvimento vai ajudar a varejista de automóveis a adotar os microsserviços mais rápido.

A [NBC Sports](#) trabalha com a Microsoft para apresentar megaeventos de transmissão. Em 2016, as duas empresas colaboraram para transmitir os Jogos Olímpicos do Rio de Janeiro. A NBC usou os Serviços de Mídia do Azure para transmitir o conteúdo de mais de 50 eventos Olímpicos simultaneamente, fornecendo um perfil de 1080p e abrangendo, de forma impressionante, 100 milhões de usuários únicos que consumiram 2,71 bilhões de minutos de conteúdo em uma ampla gama de dispositivos e fatores forma. Os Serviços de Mídia ofereceram a escalabilidade e a robustez para fornecer esse conteúdo com perfeição aos telespectadores.

O Azure dá a grandes e pequenas empresas, bem como a organizações governamentais, instituições educacionais e qualquer usuário de serviços de TI, a capacidade de gerenciar de forma mais fácil e econômica seus mundos digitais, para si mesmos e para os clientes. Em resumo, veja como:

- **Proporcionar experiências de software entre plataformas, como experiências para celular, desktop, Web e híbridas:** O Azure fornece serviços com os quais você pode implementar esses cenários de imediato, como o Serviço de Aplicativo do Azure, incluindo o Aplicativo Web e o Aplicativo Móvel e o Mobile Center e o Visual Studio Team Services para entregar e gerenciar todos os tipos de aplicativo.
- **Fornecer serviços onde quer que sejam necessários:** O Azure tem uma gama única de opções de implantação: a nuvem pública, soluções híbridas e o Azure na infraestrutura local, no seu próprio datacenter (via Azure Stack). Todas essas opções podem fornecer grande escala, a um preço acessível, mantendo seus dados onde eles precisam estar.
- **Oferecer uma plataforma de dados inteligente em grande escala:** O Azure tem muitos serviços que você pode usar para capturar, armazenar, analisar e apresentar dados. Trata-se de serviços como Bancos de Dados SQL do Azure, Data Lake Store e Data Lake Analytics, HDInsight, Hubs de Eventos, Serviços Cognitivos e muitos mais. Todos esses serviços podem criar uma plataforma de dados extremamente escalonável e inteligente a um preço acessível, dando às empresas as ferramentas para extrair insights valiosos da vasta quantidade de dados produzidos hoje em dia.
- **Fornecer software de alta qualidade, rapidamente:** Você precisa testar sempre e corrigir as falhas rápido para fornecer software de qualidade. Com o Azure, você pode fazer isso com mecanismos de CI e CD integrados diretamente a serviços como Aplicativo Web ou por meio do Visual Studio Team Services. Recursos como Slots de Implantação de Serviços de Aplicativo permitem que você forneça rápido, sem tempo de inatividade. Para saber se seu aplicativo está funcionando como esperado na produção, o Azure fornece serviços de monitoramento, como o Application Insights, que permitem que você saiba exatamente como o aplicativo está se saindo e onde é preciso melhorar.

O Azure tem alcance global

Com datacenters em mais de 42 países e regiões ao redor do globo, a cobertura incrível do Azure oferece muitos benefícios. Primeiro, pode reduzir qualquer tempo de inatividade que seus aplicativos

apresentem se um datacenter inteiro falhar (uma probabilidade extremamente baixa, por si só), pois seu aplicativo pode executar o failover para outro datacenter e continuar a ser executado quase como se nada tivesse acontecido. Isso também pode reduzir os custos de recuperação de desastres.

Se você usar CDNs (Redes de Entrega de Conteúdo) para reduzir a latência, é provável que haja um datacenter do Azure nas proximidades de onde são seus clientes e onde você precisa entregar conteúdo. Outro benefício que o Azure pode oferecer, com tantas instalações em tantos locais, é que você conta com uma gama maior de opções para o armazenamento de dados. Isso pode ajudá-lo a garantir que sua empresa permaneça em conformidade com as regulamentações e as leis locais ou regionais no que diz respeito à soberania de dados.

A Microsoft continua a investir fortemente em inovação e infraestrutura de datacenter [por meio do desenvolvimento de hardware de software livre](#) e novas soluções de datacenter, como os *datacenters subaquáticos* do [projeto Natick](#).

O Azure é extremamente resiliente

O Azure é uma plataforma inteligente, com monitoramento e correção automáticos, com a qual você pode contar para manter seus aplicativos disponíveis e funcionando corretamente. Para confirmar isso, todos os serviços do Azure operam sob [SLAs abrangentes](#) que definem a performance em termos específicos. Os SLAs variam de 99,9% (três noves) até 99,99% (quatro noves). Isso significa que os serviços do Azure são extremamente resilientes e confiáveis. Caso seus serviços não sejam executados conforme as especificações do SLA que os rege, você pode ser beneficiar de um desconto de até 100% sobre os custos de serviço em sua fatura do Azure.

Para garantir que seus serviços permaneçam com performance máxima, a Microsoft monitora a integridade de cada serviço individual do Azure em cada datacenter em todo o mundo. Essas métricas também estão disponíveis para que você as monitore [na página de status pública do Azure](#).

Como mencionado anteriormente, a Microsoft opera datacenters e fornece serviços de TI em grande escala desde 1989. Ela acumulou muito conhecimento nesse período. Hoje, a Microsoft tem algumas das melhores mentes da indústria de TI trabalhando continuamente para manter o Azure em funcionamento, para que você possa recorrer a ele para hospedar seus serviços e armazenar seus dados. É por esse motivo que todos os serviços do Azure têm SLAs para confirmar isso.

O Azure está em conformidade com quase todas as indústrias

[O Azure oferece o conjunto mais abrangente de ofertas de conformidade de qualquer provedor de nuvem](#). A lista de conformidades é enorme e continua crescendo continuamente. Alguns dos padrões com os quais o Azure está em conformidade incluem [ISO 27001](#), [HIPAA](#) e [SOC 3](#). Conformidades específicas de regiões incluem [Proteção de Privacidade UE-EUA](#) e [China DJCP](#).

Por exemplo, mantendo a conformidade com a certificação ISO 27001, a Microsoft garante que implementa, monitora, mantém e melhora continuamente seus padrões de segurança para sua rede global de datacenters, bem como para os serviços individuais no Azure.



Mais informações Para obter uma lista completa de declarações de conformidade, acesse <https://www.microsoft.com/trustcenter/compliance/complianceofferings>.

O Azure tem até uma oferta específica para os clientes do governo dos EUA, chamada [Azure Government](#), que é basicamente uma “edição especial” da nuvem do Azure que aborda as necessidades exclusivas de entidades governamentais. O Azure Government é executado separadamente da nuvem pública do Azure, em sua própria *instância*. Isso significa que ele é executado em servidores dedicados separados em uma infraestrutura completamente separada e isolada.

O Azure se concentra em recursos para desenvolvedores

O Azure ajuda a você, como desenvolvedor, a ser mais produtivo, fornecendo grande parte da estrutura que, de outra forma, você mesmo precisaria criar. Com recursos como escala automática e autenticação/ferramentas de autorização, você pode adicionar recursos e funcionalidade avançados a seus aplicativos com pouco ou nenhum esforço. Você pode integrar serviços como Azure Search ou Serviços Cognitivos, que melhorarão seus aplicativos e agradarão aos usuários.

O Azure é aberto e dá suporte às estruturas de sua escolha

Escreva seus aplicativos em JavaScript e implante-os em um Aplicativo Web. Ou escreva-os em Ruby, Django, Java, PHP ou .NET, seja qual for seu ambiente preferido. O Azure tem amplo suporte para várias linguagens. Você pode migrar seu aplicativo existente ou escrever um do zero, e ele será executado no Azure.

Além disso, você pode escolher o ecossistema que preferir. O Azure dá suporte à maioria dos sistemas operacionais, como Linux e Windows, e você pode executar scripts para tudo usando o Bash, se quiser. Você também pode executar o aplicativo da maneira que quiser: utilizando contêineres, Azure Service Fabric, Serviços de Aplicativo, Azure Stack e assim por diante.

Depois de aprender a usar o Azure com um conjunto de ferramentas, você pode usá-lo com qualquer outro conjunto. Os serviços e o Portal do Azure funcionam da mesma forma para tudo.

Você pode monitorar seus serviços do Azure em qualquer lugar

Você é o tipo de pessoa que gosta de manter o controle das operações? Com o Azure, você pode monitorar a execução dos serviços e corrigir problemas que possam surgir usando o [aplicativo móvel do Azure](#). Esse aplicativo útil foi projetado para lhe dar visibilidade de seus recursos, mesmo quando você não estiver diante da mesa usando seu computador principal. Você pode lidar com tarefas como reiniciar um aplicativo Web e parar uma máquina virtual. Ele também fornece uma experiência completa de CLI (interface de linha de comando) via Shell de Nuvem, tornando possível para você fazer basicamente tudo o que pode fazer com o Portal do Azure.

Além disso, existem serviços no Azure que executam detecções inteligentes e recomendações para seus serviços e podem enviar alertas sobre eles. Trata-se de itens como detectar que um aplicativo Web está mais lento do que o habitual ou informar que um Serviço de Aplicativo foi ampliado automaticamente. A Central de Segurança do Azure e o Azure Monitor são alguns dos serviços que podem alertá-lo desses acontecimentos, para que você nunca perca nada importante.

Estamos aqui se você precisar de ajuda

Existem muitas maneiras de obter ajuda com o Azure, se você precisar.

Você pode comprar [um plano de suporte](#), que lhe dá acesso às equipes de suporte técnico do Azure e fornece outros serviços, como orientação baseada em práticas recomendadas para projetar para a nuvem ou assistência para planejar sua migração. Dependendo de suas necessidades, você pode comprar um plano de suporte que garante respostas das equipes de suporte técnico no prazo de 15 minutos.

Além disso, você pode obter ajuda gratuitamente, 24x7, por meio de muitos canais. Você pode enviar um tweet para [@AzureSupport](#) ou entrar em contato com as equipes de suporte em [fóruns do MSDN](#), no [StackOverflow](#), no [Reddit](#) ou na [Comunidade Microsoft Tech](#).

Introdução ao Azure

Após refletir por algum tempo, você se juntou à comunidade do Azure e, agora, quer começar a criar aplicativos. Do que você precisa? Não muito, na verdade. O mais importante é ter apenas uma conexão com o Azure para implantar seu aplicativo. Você pode usar as ferramentas, os aplicativos e as estruturas de sua escolha.

Seus IDEs e editores são bem-vindos

Tudo o que você usa para criar ou editar aplicativos em seu ambiente na infraestrutura local também pode usar com o Azure. Escolha qualquer coisa: desde o Bloco de Notas até o [Visual Studio Code](#), o [Sublime Text](#), o [Visual Studio](#) ou os editores de código no Portal do Azure. A decisão é sua!

Há plug-ins disponíveis para todos os IDEs populares, dando-lhe a capacidade de realizar tarefas como publicar diretamente no Azure. Mesmo que você prefira usar uma ferramenta simples, como o Bloco de Notas, ainda pode automatizar suas implantações no Azure usando práticas de IC (Integração Contínua)/CD (Entrega Contínua).

Use a interface de linha de comando do Azure para scripts

O Azure fornece uma poderosa [CLI \(interface de linha de comando\)](#) com a qual você pode fazer basicamente tudo no Azure. Você pode usar a CLI do Azure para iniciar e parar VMs (máquinas virtuais), implantar seu aplicativo em um aplicativo no Aplicativo Web do Azure, criar novos recursos e tudo o mais. Também é muito útil para automatização de tarefas e para executá-las em pipelines de CI/CD.

Você pode usar a CLI do Azure disponível no Portal do Azure no [Azure Cloud Shell](#) ou localmente, em sua máquina, por meio das ferramentas da CLI do Azure. Outra vantagem é que você pode aplicar seus conhecimentos existentes de Bash ou Windows PowerShell à CLI do Azure.

Execute qualquer coisa no Azure

O Azure é ótimo para aplicativos Web e APIs. Também é ótimo como um host para aplicativos móveis e de desktop. Por exemplo, você pode usá-lo para autenticar usuários em seu aplicativo de desktop ou enviar notificações por push ao aplicativo móvel. Não importa qual cenário você tenha em mente, o Azure pode agregar valor a ele.

E quanto aos custos?

Claramente, é importante saber os custos do Azure. Para ajudá-lo nisso, a Microsoft oferece a [Calculadora de Preços do Azure](#). Com essa ferramenta útil, você seleciona os serviços que vai usar e especifica o uso previsto. A calculadora informa qual será sua fatura mensal e anual. Juntamente com a quantidade de dados que você usará, isso determina seus custos.



Observação A Calculadora de Preços do Azure estima os custos para assinaturas pré-pagas. Existem outras opções de pagamento para empresas, parceiros e assinantes do MSDN, que podem afetar os custos dos serviços.

Você também pode ficar atento aos custos usando as [APIs de Cobrança do Azure](#) e os [Alertas de preços](#). Você pode tirar proveito deles para obter insights sobre os custos e as tendências de gastos, para que possa controlar seus custos.

Selecionar os serviços certos do Azure

O Azure fornece uma série de serviços para ajudá-lo a criar e executar aplicativos. Existem tantos serviços, aliás, que às vezes pode ser um pouco confuso escolher os serviços certos para seu cenário. Quais deles você deve escolher como desenvolvedor? Vamos explorar algumas de suas opções.

Onde hospedar seu aplicativo

A primeira escolha que você precisará fazer, logo de início, é onde hospedar seu aplicativo. O Azure oferece várias opções de hospedagem.

VMs

Uma das maneiras de hospedar seu aplicativo é em uma VM em uma [Máquina Virtual do Azure](#). Isso fornece muito controle sobre como você hospeda o aplicativo, mas você é responsável por manter o ambiente, como aplicar correções do SO (sistema operacional) e manter programas antivírus atualizados.

Por exemplo, você pode usar uma VM para testar a mais recente versão de visualização do Visual Studio, sem afetar sua máquina.

Lote do Azure

Se você precisar executar aplicativos de computação em grande escala ou de alta performance (HPC) em VMs, escolha o [Lote do Azure](#). O Lote cria e gerencia uma coleção de até milhares de VMs, instala os aplicativos que você deseja executar e agenda os trabalhos nas VMs. Não é necessário implantar e gerenciar VMs individuais ou clusters de servidor. O Lote programa, gerencia e escala automaticamente seus trabalhos, para que você use apenas as VMs necessárias. Lote é um serviço gratuito. Você só paga pelos recursos subjacentes consumidos, como VMs, armazenamento e rede.

O Lote é bem adequado para executar cargas de trabalho paralelas em escala, como modelos de risco financeiro, transcodificação de mídia, VFX e renderização de imagem 3D, simulações de engenharia

e muitos outros aplicativos de computação intensiva. Use o Lote para escalar horizontalmente um aplicativo ou script que você já executa em estações de trabalho ou um cluster na infraestrutura local ou desenvolver soluções SaaS que usam o Lote como uma plataforma de computação.



Experimente Aprenda a usar o Lote do Azure em cinco minutos com [estes tutoriais passo a passo](#).

Contêineres

Contêineres são muito mais leves do que VMs. Você pode iniciá-los e pará-los em poucos segundos. Os contêineres também oferecem enorme portabilidade, o que os torna ideais para desenvolver um aplicativo localmente, em sua máquina, e depois hospedá-lo na nuvem, em testes e, mais tarde, na produção. Você pode até mesmo executar contêineres na infraestrutura local ou em outras nuvens. O ambiente que você usa em sua máquina de desenvolvimento vai junto com o contêiner, assim, o aplicativo sempre é executado no mesmo ecossistema.

Assim como VMs, contêineres fornecem muito controle sobre o ambiente. Você pode instalar tudo de que precisa para executar os aplicativos. Porém, novamente, você é responsável por corrigir e manter o sistema operacional que é executado no contêiner e por produtos conexos, como programas antivírus.

Contêineres de host com Instâncias de Contêiner do Azure

Você pode hospedar o contêiner usando [Instâncias de Contêiner do Azure](#). Usando esse serviço, você pode rapidamente criar um contêiner, sem a necessidade de um orquestrador de contêiner, como o Kubernetes, e sem a necessidade de gerenciar os recursos que hospedam o contêiner.

O serviço de Instâncias de Contêiner é faturado por segundo, por CPU virtual, por gigabyte ou por memória.



Mais informações Para saber mais sobre Instâncias de Contêiner, acesse [Introdução a Instâncias de Contêiner do Azure](#).

Contêineres de host com o Serviço de Contêiner do Azure

Outra forma de hospedar contêineres é usar o [Serviço de Contêiner do Azure](#). Com esse serviço, você pode escalar e gerenciar os contêineres usando orquestradores como Mesosphere DC/OS, Docker Swarm ou Kubernetes. Esse serviço é uma ótima maneira de começar a migrar os contêineres para a nuvem.



Mais informações Para saber mais sobre o Serviço de Contêiner, acesse [Introdução ao Serviço de Contêiner do Azure](#).

Funções do Azure

Com Funções do Azure, você pode escrever apenas o código necessário para uma solução, sem se preocupar com a criação de um aplicativo completo ou a infraestrutura para executá-lo. Uma função é uma unidade lógica de código que é disparada por uma solicitação HTTP, um evento em outro serviço do Azure ou com base em uma agenda. As associações de entrada e saída conectam o código de função a outros serviços, como Armazenamento de Blobs do Azure, Azure Cosmos DB e Barramento de Serviço do Azure, com código mínimo. Usando Funções, você pode criar pequenos itens de funcionalidade rapidamente e hospedá-los em um ambiente elástico que gerencia automaticamente a escala.

Outra coisa que torna as Funções do Azure especiais é que você pode optar por pagar apenas pelas funções executadas, sem ter que manter as instâncias de computação em execução durante todo o mês. Isso também é chamado de *sem servidor* porque exige apenas que você crie seu aplicativo e não lide com nenhum servidor ou mesmo com escala de servidores.

Você pode escrever Funções do Azure em C#, F#, Node.js, Java, PHP e uma crescente lista de linguagens.

Um exemplo de um aplicativo que usa Funções é aquele que ativa uma função sempre que um novo arquivo de imagem é carregado no Armazenamento de Blobs do Azure. A função redimensionaria a imagem e a gravaria em outra conta do Armazenamento de Blobs. A assinatura da função para este exemplo seria assim (no script C#):

```
public static void Run(Stream myBlob, string name, TraceWriter log, BlockBlob outputBlob) { }
```

Os dados do Blob que disparou a função são passados para a função como o parâmetro myBlob, que inclui a URL do Blob. Você pode usar o parâmetro de associação de saída outputBlob para especificar o Blob no qual gravar o resultado. Não há necessidade de gravar a estrutura para se conectar ao Armazenamento de Blobs: você apenas a configura.



Experimente [Crie sua primeira Função do Azure usando o Portal do Azure.](#)

Aplicativos Lógicos do Azure

Você pode orquestrar a lógica de negócios com [Aplicativos Lógicos](#), automatizar um processo de negócios ou integrar aplicativos de SaaS (software como serviço). Assim como nas Funções do Azure, os Aplicativos Lógicos podem ser ativados por uma fonte externa, por exemplo, uma nova mensagem em uma Fila de Armazenamento do Azure. Você liga as chamadas da API a conectores para criar um fluxo de trabalho (possivelmente complexo) que pode envolver recursos na nuvem e na infraestrutura local. Os Aplicativos Lógicos têm [muitos conectores disponíveis para APIs](#), como um para conectar-se aos Bancos de Dados SQL do Azure, ao Salesforce, ao SAP e assim por diante. Você também pode expor suas próprias APIs ou Funções do Azure como conectores para usar em um Aplicativo Lógico, podendo realizar ações facilmente em relação a sistemas externos em seu fluxo de trabalho ou fazer com que o Aplicativo Lógico seja ativado por um deles.

Assim como as Funções do Azure, os Aplicativos Lógicos são sem servidor, são dimensionados automaticamente e você paga por eles somente quando são executados.

Aqui está um exemplo de um fluxo de trabalho nos Aplicativos Lógicos:

1. O Aplicativo Lógico é ativado por um email contendo uma ordem de envio que chega no Office 365.
2. Usando os dados no email, o Aplicativo Lógico verifica a disponibilidade do item encomendado no SQL Server.
3. O Aplicativo Lógico envia uma mensagem de texto para o telefone do cliente usando o Twilio (o número de telefone também estava no email), indicando que a solicitação foi recebida e o item foi enviado.



Experimente [Introdução aos Aplicativos Lógicos do Azure.](#)

Serviço de Aplicativo do Azure

Como alternativa, você pode hospedar seus aplicativos em uma das principais ofertas de serviços no Azure: o [Serviço de Aplicativo do Azure](#). O Serviço de Aplicativo do Azure é uma coleção de serviços de hospedagem e orquestração que compartilham recursos. Por exemplo, todos os Serviços de Aplicativos têm a capacidade de proteger um aplicativo usando o [Azure Active Directory](#) e podem usar domínios personalizados.

O Serviço de Aplicativo do Azure compreende o seguinte:

- **Aplicativo Web:** [Aplicativo Web](#) é um dos serviços do Azure mais usados. Você pode usar isso para hospedar seus aplicativos Web ou APIs. Um aplicativo Web é basicamente uma abstração de um servidor Web, como IIS (Serviço de Informações da Internet) ou Tomcat, que você usa para hospedar aplicativos com base em HTTP.

O aplicativo Web pode hospedar aplicativos escritos em .NET, Node.js, PHP, Java ou Python, e há extensões que você pode usar para executar ainda mais linguagens.



Experimente Explicamos um aplicativo de exemplo em Node.js e MongoDB em nossa seção de exemplo.

- **Aplicativo Web para Contêineres:** [Aplicativo Web para Contêineres](#) ajuda você a implantar e executar facilmente aplicativos Web em contêineres em escala. Basta extrair as imagens de contêiner do Docker Hub ou um Registro privado do Contêiner do Azure, e o Aplicativo Web para Contêineres vai implantar o aplicativo em contêiner com suas dependências preferenciais para a produção em segundos. A plataforma cuida automaticamente de correções do sistema operacional, provisionamento de capacidade e balanceamento de carga.



Experimente Explicamos a implantação de um [aplicativo Web](#) com um contêiner formatado do Docker em nossa seção de exemplo.

- **Aplicativo Móvel:** [Aplicativo Móvel](#) fornece um back-end para seus aplicativos móveis. Você hospeda uma API no Aplicativo Móvel à qual que seus aplicativos móveis se conectam por meio do SDK do cliente entre plataformas. Isso está disponível para iOS, Android, Windows e Xamarin para iOS e Android e Xamarin Forms. O Aplicativo Móvel oferece recursos exclusivos, como Sincronização Offline e Notificações por Push, que o ajudam a criar uma experiência móvel moderna, com boa performance e segura.

Você pode escrever seu back-end de Dispositivo Móvel no .NET ou Node.js.



Experimente Explicamos a criação de um back-end móvel em nossa seção de exemplo.

Recursos do Serviço de Aplicativo do Azure

O Serviço de Aplicativo do Azure é um dos principais serviços do Azure que você pode usar para hospedar seus aplicativos. Cada um desses serviços oferece recursos exclusivos, mas todos eles compartilham alguns recursos comuns:

Escala

O Serviço de Aplicativo do Azure é executado em [Planos de Serviços de Aplicativo](#), que são abstrações de VMs. Uma ou mais VMs executam seu Serviço de Aplicativo do Azure, mas você não precisa saber quais delas, pois o Azure cuida delas. No entanto, você pode dimensionar os recursos que executam seu Serviço de Aplicativo do Azure. Você pode escolher um nível mais alto de preços (variando de grátis a premium) ou aumentar o número de instâncias do aplicativo que estão sendo

executadas. Você pode até fazer com que o Serviço de Aplicativo do Azure dimensione automaticamente o número de instâncias para você, com base em uma agenda ou métricas como CPU, memória ou comprimento da fila HTTP.

Slots de implantação

Esse é um recurso muito útil do Serviço de Aplicativo do Azure. Você pode implantar uma nova versão de seu aplicativo em um slot de implantação, em que pode testar se ele funciona como esperado e movê-lo para seu slot de produção. Você pode até usar o *recurso* [Teste em Produção](#) do Azure para encaminhar um percentual do tráfego de seu aplicativo de produção para um slot de implantação. Por exemplo, você pode mover 10% dos usuários para a nova versão de seu aplicativo no slot de implantação para ver se os novos recursos estão funcionando como esperado e se os usuários estão realmente utilizando-os.

Depois que estiver satisfeito com a performance da nova versão de seu aplicativo no slot de implantação, você poderá realizar uma *troca*, que troca o aplicativo no slot de implantação por aquele no slot de produção. Você também pode trocar de um slot de desenvolvimento para um slot de teste e para o slot de produção, conforme ilustrado na Figura 2-1. Antes de fazer isso, a operação de troca verifica se a nova versão de seu site está pronta para ser usada. Quando isso é confirmado, a operação de troca muda os slots, e os usuários agora veem a nova versão do aplicativo, sem tempo de inatividade. Se você quiser, também poderá trocar de volta, retornando a implantação da nova versão.

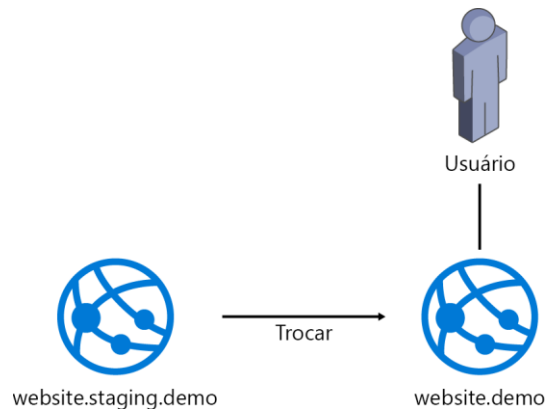


Figura 2-1: Trocando um slot de implantação

Você usa os slots de implantação *dentro* de um ambiente como Desenvolvimento, Teste ou Produção. Você não usa os slots de implantação *como* ambientes, pois todos eles residem no mesmo Plano de Serviço de Aplicativo, e você deseja que sejam separados para segurança, escala, cobrança e performance.

Você pode trocar os slots de implantação manualmente, por meio da CLI (interface de linha de comando) do Azure e por meio da API de Gerenciamento do Azure. Isso permite que ferramentas como o Visual Studio Team Services executem operações de troca durante um lançamento.

Um slot de implantação é outro elemento do Serviço de Aplicativo do Azure (como um aplicativo Web) que é executado no mesmo Plano de Serviço de Aplicativos do Azure, ao lado de seu Serviço de Aplicativo do Azure original. Como os slots de implantação são executados no mesmo Plano de Serviço de Aplicativo do Azure que seu Serviço de Aplicativo do Azure original, eles não têm custo extra para serem usados.

Entrega Contínua

Para publicar seu aplicativo em Serviços de Aplicativos, você pode usar serviços externos, como Visual Studio Team Services, Jenkins ou Octopus Deploy. Você também pode usar o *recurso de CD (entrega contínua)* nos Serviços de Aplicativos. Isso possibilita que você crie um pipeline de lançamento de teste de compilação diretamente no Serviço de Aplicativo. O processo faz o seguinte:

1. Recupera o código-fonte mais recente do repositório que você indica
2. Compila o código de acordo com um modelo que você escolhe (ASP.NET, Node.js e assim por diante)
3. Implanta o aplicativo em um ambiente de teste, carrega-o e testa-o
4. Implanta o aplicativo para a produção após a aprovação (você pode indicar se deseja usar um slot de implantação)

Conectar-se a recursos na infraestrutura local

Você pode conectar recursos externos, como repositórios de dados, aos Serviços de Aplicativos. Esses recursos não precisam estar localizados no Azure. Eles podem estar em qualquer lugar, como na infraestrutura local, em seu próprio datacenter. Você pode se conectar a serviços na infraestrutura local por meio de muitos mecanismos, dependendo de seus requisitos. Você pode usar [Conexões Híbridas do Azure](#), [Redes Virtuais do Azure](#) e o [Azure ExpressRoute](#) para se conectar a recursos na infraestrutura local.

Domínios personalizados e certificados do Serviço de Aplicativo do Azure

Quando você gera um aplicativo no Serviço de Aplicativo do Azure, ele expõe uma URL. Por exemplo: <https://myazurewebsite.azurewebsites.net>. Provavelmente, convém usar seu próprio domínio personalizado, o que você pode fazer mapeando esse nome de domínio para Serviços de Aplicativos. [Veja como fazer isso](#).

Além disso, você pode garantir que seu aplicativo seja fornecido por meio de HTTPS usando um certificado SSL (Secure Sockets Layer). Você pode trazer seu próprio certificado ou comprar um [diretamente do Portal do Azure](#). Ao comprar um certificado SSL do Portal do Azure, você compra um certificado de Serviço de Aplicativo do Azure, que pode configurar para ser usado por associações de domínio personalizadas.



Experimente [Veja como comprar e configurar um certificado neste passo a passo](#).

Ambiente de Serviço de Aplicativo

Em um aplicativo Web com várias camadas, você geralmente tem um banco de dados ou serviços que são usados pelo aplicativo no Aplicativo Web. De forma ideal, você deseja que esses serviços sejam expostos apenas ao aplicativo, não à Internet. Claro, o aplicativo em si é muitas vezes voltado para a Internet, pois fornece o ponto de entrada para seus usuários.

Para isolar esses serviços de suporte da Internet, você pode usar uma Rede Virtual do Azure. Esse serviço envolve seus serviços de suporte e os conecta ao aplicativo no Aplicativo Web, de maneira que os serviços de suporte sejam expostos apenas ao aplicativo, não à Internet. [Este artigo](#) descreve esse serviço com mais detalhes e mostra como usá-lo.

Às vezes, convém ter ainda mais controle. Talvez você queira que seu aplicativo seja envolvido em uma Rede Virtual para que você possa controlar o acesso a ele. Talvez você queira que ele seja chamado por outro aplicativo no aplicativo Web e que faça parte de seu back-end. Para esse cenário, você pode usar um [ambiente de Serviço de Aplicativo do Azure](#). Isso proporciona uma escala muito alta e lhe dá controle sobre isolamento e acesso à rede. Observe, no entanto, que o Ambiente do Serviço do Aplicativo está disponível apenas para Serviços de Aplicativos nas camadas de preços premium.



Observação No momento, o Ambiente do Serviço de Aplicativo não funciona para o aplicativo Web para Contêineres.

O que usar quando?

Mesmo que o Serviço de Aplicativo, as Funções e os Aplicativos Lógicos funcionem frequentemente lado a lado, cada um é projetado tendo em mente as necessidades de aplicativos específicos.

A Tabela 2-1 descreve resumidamente o propósito da categoria de necessidades que você pode ter.

Tabela 2-1: Qual serviço de aplicativo você deve usar quando

Categoria	Objetivo	Tipo de Serviço de Aplicativo do Azure
Aplicativos de hospedagem	Aplicativos Web host e APIs	Aplicativo Web e/ou Aplicativo Web para Contêineres do Serviço de Aplicativo
	Back-end de host para aplicativos móveis	Aplicativo Móvel do Serviço de Aplicativo
Orquestrar processos	Orquestrar uma etapa em um processo	Funções
	Orquestrar todo o processo	Aplicativos Lógicos

Tornar seu aplicativo mais rápido

Depois que seu aplicativo está funcionando no Azure, você quer que ele tenha a melhor performance possível. O Azure fornece uma gama de serviços que podem ajudá-lo a obter isso.

Rede de Distribuição de Conteúdo do Azure

Um dos serviços Azure que pode ajudá-lo a tornar seu aplicativo mais rápido é a [Rede de Distribuição de Conteúdo do Azure](#). Você carrega arquivos estáticos (vídeos, imagens, JavaScript, CSS e até mesmo arquivos HTML estáticos) para um repositório de dados, como o Armazenamento de Blobs do Azure, e depois conecta a Rede de Distribuição de Conteúdo do Azure a ele. A Rede de Distribuição de Conteúdo obtém então esses arquivos estáticos e os replica para centenas de PoP (Pontos de Presença) em todo o mundo. Tudo o que você precisa fazer no aplicativo é mudar a referência aos arquivos estáticos para uma URL diferente. Por exemplo, se anteriormente essa referência era ~/images/image.png, agora será https://example.azureedge.com/image.png. Isso é muito fácil de fazer e melhora a performance de seu aplicativo das seguintes maneiras:

- Descarrega o conteúdo do aplicativo. Agora ele é fornecido pela Rede de Distribuição de Conteúdo, liberando assim ciclos de processamento para o aplicativo
- Torna o conteúdo estático fisicamente mais próximo dos usuários, distribuindo-o para PoPs em todo o mundo

Você pode se beneficiar da Rede de distribuição de conteúdo em aplicativos Web, mas também em aplicativos móveis e da área de trabalho.

Um exemplo de uso da Rede de Distribuição de Conteúdo é para fornecer vídeos a um aplicativo móvel. Os vídeos podem ser grandes, e você não deseja armazená-los no dispositivo móvel (e os usuários também não!). Usando a Rede de Distribuição de Conteúdo, eles são fornecidos do PoP, o que também melhora a performance, pois está perto do usuário.



Experimente [Comece com a Rede de Distribuição de Conteúdo do Azure.](#)

Cache Redis do Azure

Todo aplicativo moderno trabalha com dados. Quando você recupera dados de um repositório de dados como um banco de dados, isso geralmente envolve a verificação de várias tabelas ou documentos em algum servidor distante, reunindo os resultados e enviando o resultado para o dispositivo solicitante. Isso, é claro, leva tempo e pode frustrar e irritar seus usuários.

Para eliminar algumas dessas “viagens de ida e volta”, você pode armazenar em cache dados que não mudam com frequência. Dessa forma, em vez de consultar o banco de dados a cada vez, você pode recuperar alguns dos dados de um cache, como o [Cache Redis do Azure](#). O benefício do cache é que ele armazena dados em um formato simples, como o valor-chave. Você não precisa executar uma consulta complexa para obter esses dados. Basta saber a chave para recuperar o valor. Isso pode melhorar muito a performance do aplicativo. Veja como funciona este fluxo de trabalho:

1. O aplicativo precisa de alguns dados e tenta recuperá-lo do cache.
2. Se os dados não estiverem lá, ele os obtém do banco de dados e também armazena os dados no cache.
3. Na próxima vez que o aplicativo estiver procurando por esses dados específicos, ele os achará no cache, economizando uma viagem ao banco de dados.

O Azure fornece Cache como Serviço com o Cache Redis. Isso é baseado no projeto de software livre do Redis e agora é apoiado por engenheiros e SLAs da Microsoft. Tem alta performance e tem opções avançadas como clustering e replicação geográfica.



Experimente [Introdução ao Cache Redis do Azure.](#)

Gerenciador de Tráfego do Azure

Muitos aplicativos modernos têm usuários em todo o mundo. Fornecer uma experiência de alta performance para todos é desafiador, no mínimo. O problema mais óbvio com o qual você precisa lidar é a latência. Latência é o tempo necessário para um sinal ou uma solicitação chegar a um usuário. Quanto mais longe os usuários estão do aplicativo, mais latência eles experimentam.

O [Gerenciador de Tráfego do Azure](#) é escalado entre regiões, o que ajuda a reduzir a latência e a fornecer aos usuários uma experiência de alta performance, independentemente de onde eles estão. O Gerenciador de Tráfego é um mecanismo de roteamento inteligente que você coloca na frente, por exemplo, de aplicativos do Aplicativo Web, em todo o mundo. O Aplicativo Web funciona como pontos de extremidade, que o Gerenciador de Tráfego do Azure monitora para verificar a integridade e a performance. Como mostra a Figura 2-2, quando uma usuária acessa seu aplicativo, o Gerenciador de Tráfego a encaminha para o aplicativo do Aplicativo Web que tem melhor performance nas proximidades.

Incluir o Gerenciador de Tráfego em sua arquitetura é uma ótima maneira de melhorar a performance de seu aplicativo.

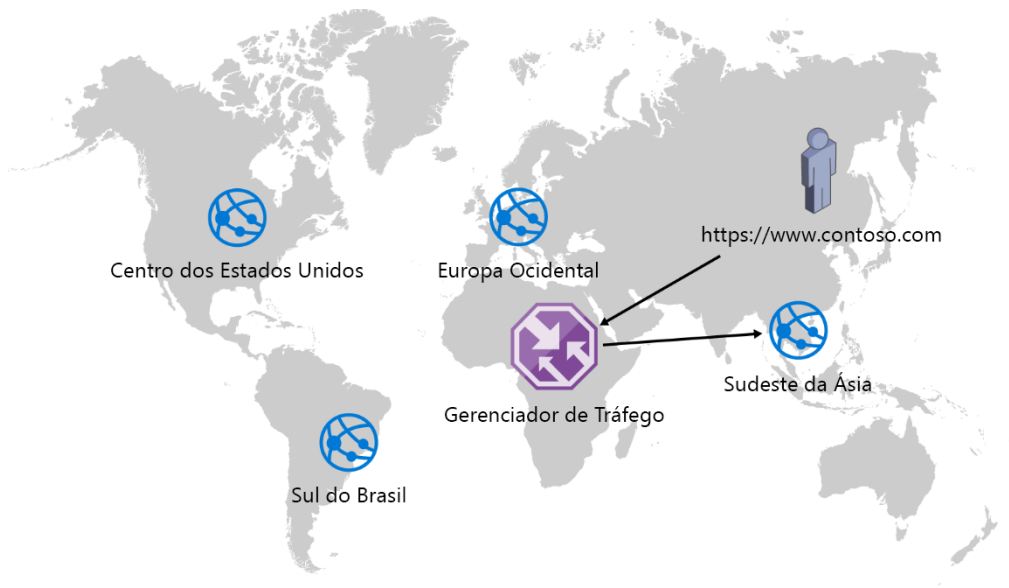


Figura 2-2: O Gerenciador de Tráfego Azure dirige o tráfego para o ponto de extremidade geográfico com melhor performance

Onde armazenar seus dados

Os dados são um aspecto muito importante de qualquer aplicativo moderno e têm todos os formatos e portes. O Azure fornece muitos tipos de repositórios de dados que podem ajudá-lo a manter e recuperar dados em qualquer cenário. A Tabela 2-2 apresenta as opções de armazenamento disponíveis no Azure.

Tabela 2-2: Opções de armazenamento no Azure

	Bancos de dados SQL	MySQL	PostgreSQL	Cosmos DB	Blob	Tabela	Fila	Arquivo	Disco	Data Lake Store	SQL Data Warehouse
Dados relacionais	X	X									
Dados relacionais-objeto			X								
Dados não estruturados				X	X						
Dados semiestruturados						X					
Mensagens de fila							X				
Arquivos em disco								X			
Arquivos de alta performance em disco									X		
Armazenar dados grandes					X			X	X	X	X
Armazenar dados pequenos	X	X	X	X		X	X	X	X		
Replicação de dados geográficos	X			X							



Observação Você pode usar quase todas as opções de armazenamento mencionadas nesta seção [como ativadores e associações para as Funções do Azure](#).

Vamos conferir cada opção de armazenamento.

Bancos de Dados SQL do Azure

Se você quer usar tabelas com colunas e linhas para armazenar dados, o [Banco de Dados SQL do Azure](#) é uma ótima escolha. É um sistema de banco de dados relacional semelhante ao Microsoft SQL Server na infraestrutura local. Como o Banco de Dados SQL é executado na nuvem, ele é totalmente gerenciado, com boa performance, escalonável, com backup automático e muitos recursos avançados.

Você pode usar o Banco de Dados SQL com suas ferramentas favoritas, inclusive o SQL Server Management Studio e o Entity Framework.

Aqui estão alguns dos recursos mais avançados:

- Replicação geográfica, que replica dados para outras regiões geográficas em tempo real ([Introdução à replicação geográfica](#))

- Mascaramento de dados dinâmicos, que mascara dinamicamente dados confidenciais para determinados usuários em tempo de execução ([Introdução ao mascaramento de dados dinâmicos](#))
- Auditoria, que fornece uma trilha de auditoria completa de todas as ações que acontecem com os dados ([Introdução à auditoria de Banco de Dados SQL do Azure](#))

Os bancos de dados no Banco de Dados SQL são extremamente confiáveis e robustos e oferecem um SLA (Contrato de Nível de Serviço) que garante 99,99% de tempo de atividade.

Banco de Dados do Azure para MySQL

O MySQL é um sistema de banco de dados relacional de software livre que é usado por milhões de aplicativos, como WordPress, Joomla e Drupal. Agora, o MySQL está disponível como um serviço de banco de dados gerenciado no serviço [Banco de Dados do Azure para MySQL](#). Isso significa que você pode começar a trabalhar rapidamente, sem a necessidade de configurar ou manter servidores. O Azure faz tudo para você. Você obtém segurança e escalabilidade desde o início. Além disso, o serviço Banco de Dados para MySQL tem a mesma confiabilidade que o Banco de Dados SQL, fornecendo um SLA de 99,99% de tempo de atividade líder na indústria.

Assim como com o Banco de Dados SQL e o Banco de Dados do Azure para PostgreSQL, os backups incrementais são feitos a cada cinco minutos e backups completos são feitos a cada hora, o que você pode usar para recuperar os dados em um estado anterior de até 35 dias no passado.



Experimente [Crie e conecte-se a um banco de dados MySQL no Azure.](#)

Banco de Dados do Azure para PostgreSQL

Com base na popular tecnologia de banco de dados PostgreSQL, o [Banco de Dados do Azure para PostgreSQL](#) fornece um serviço de banco de dados objeto-relacional. Isso é ligeiramente diferente dos serviços de banco de dados relacionais. Os bancos de dados objeto-relacionais podem armazenar tipos de dados mais complexos, como tipos de dados aninhados. Portanto, estão mais alinhados com as linguagens de programação orientadas a objetos que usamos em nossos aplicativos.

O banco de dados para PostgreSQL é um serviço gerenciado que tem as mesmas características que o Banco de Dados para MySQL. Você pode ampliá-lo e reduzi-lo, ele está altamente disponível (SLA de 99,99%) e o backup é feito automaticamente.



Experimente [Crie um Banco de Dados do Azure para PostgreSQL usando a CLI do Azure.](#)

Azure Cosmos DB

O [Azure Cosmos DB](#) é a nova versão e marca do DocumentDB, e muito mais. O Cosmos DB é um novo tipo de banco de dados verdadeiramente feito para a nuvem. Aqui estão alguns dos principais recursos:

- Um SLA de 99,99% que inclui latências baixas (menos de 10 ms em leituras e menos de 15 ms em gravações).
- Replicação geográfica, que replica dados para outras regiões geográficas em tempo real ([Como distribuir dados globalmente com o Azure Cosmos DB](#)).
- Gerenciamento de Tráfego, que envia os usuários para a réplica de dados da qual estão mais próximos.

- Escala global sem limites. Você paga apenas pela taxa de transferência e pelo armazenamento de que precisa.
- Indexação automática de dados. Não é mais necessário manter ou ajustar o banco de dados.

Além de todos esses recursos, o Cosmos DB oferece diferentes APIs com as quais você pode armazenar e recuperar dados, inclusive SQL, JavaScript, Gremlin, MongoDB e Armazenamento de Tabelas do Azure. APIs diferentes manipulam dados de diferentes maneiras. Você pode usar documentos como dados, bem como tabelas não estruturadas, gráficos, blobs e assim por diante. Você usa a API que se adapta às suas necessidades, e o Cosmos DB cuida do resto. Você se beneficia de performance, escalabilidade e confiabilidade de classe da nuvem e ainda usa o modelo de programação ao qual está acostumado.



Experimente [Introdução à API do DocumentDB do Azure Cosmos DB.](#)

Armazenamento do Azure

Outra opção para armazenar dados é o [Armazenamento do Azure](#). Esse é um dos serviços mais antigos, confiáveis e com boa performance no Azure. O Armazenamento do Azure oferece cinco tipos de armazenamento que se beneficiam dos seguintes recursos compartilhados:

- Redundância geográfica que reúne dados em diferentes datacenters para que você possa recuperá-los, caso um desastre faça um datacenter individual falhar
- Criptografia de dados em tempo de execução
- Domínios personalizados

Os quatro tipos de armazenamento Azure são *Blob*, *Fila*, *Arquivo* e *Disco* (Figura 2-3).

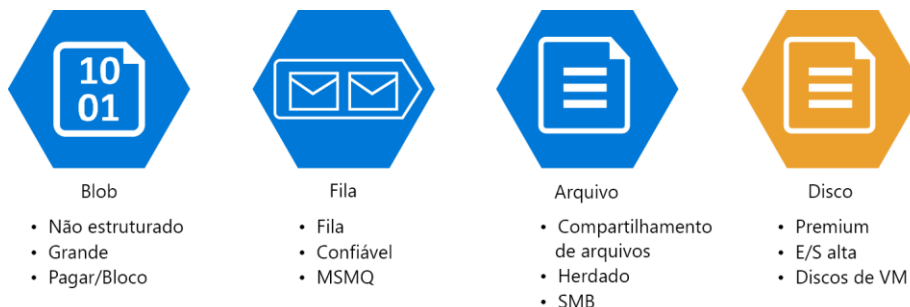


Figura 2-3: Visão geral dos serviços de Armazenamento do Azure

Armazenamento de Blobs

Você pode usar o [Armazenamento de Blobs do Azure](#) para armazenar grandes dados não estruturados, ou seja, blobs de dados. Isso pode consistir em vídeo, imagens, áudio ou texto ou até mesmo arquivos de VHD (disco rígido virtual) para VMs.

Existem dois tipos de Blobs: [Blobs de Página e de Bloco](#). Os Blobs de Página são otimizados para operações aleatórias de leitura e gravação. São perfeitos para armazenar um VHD. Os Blobs de Bloco são otimizados para carregar de forma eficiente grandes quantidades de dados. São perfeitos para armazenar arquivos de vídeo grandes que não mudam com frequência.



Experimente [Introdução ao Armazenamento de Blobs do Azure.](#)

Fila de Armazenamento

O [Armazenamento de Filas do Azure](#) é um tipo de armazenamento incomum, pois é usado para armazenar pequenas mensagens de dados, mas seu objetivo principal é servir como uma fila. Você coloca mensagens na fila, e outros processos as obtêm. [Esse padrão](#) separa o remetente da mensagem do processador de mensagens, o que resulta em benefícios de performance e confiabilidade. O Armazenamento de Filas do Azure é baseado no Enfileiramento de Mensagens da Microsoft que você pode encontrar em versões anteriores do Windows.



Experimente [Introdução ao Armazenamento de Filas do Azure.](#)

Armazenamento de Arquivos

É possível usar o [Armazenamento de Arquivos do Azure](#) como uma unidade da qual você pode compartilhar arquivos. Ele usa o protocolo SMB (Server Message Block), o que significa que você pode usá-lo com o Windows e o Linux e pode acessá-lo da nuvem ou de sistemas na infraestrutura local. Como os outros tipos de Armazenamento do Azure, o Armazenamento de Arquivos é escalonável e barato.



Experimente [Introdução ao Armazenamento de Arquivos do Azure.](#)

Armazenamento em Disco

O [Armazenamento em Disco do Azure](#) é semelhante ao Armazenamento de Arquivos, mas é especificamente destinado a alta performance de E/S. É perfeito para usar como uma unidade em uma VM que precisa de alta performance, como uma VM que executa o SQL Server. Observe, no entanto, que o Armazenamento em Disco está disponível apenas na camada de preços premium do Armazenamento do Azure.

Azure Data Lake Store

Os repositórios de dados anteriores se destinam ao uso regular de aplicativos ou ao uso com VMs. O [Azure Data Lake Store](#) é um armazenamento para aplicativos de Big Data. Você o utiliza para armazenar grandes quantidades de dados em seu formato nativo: estruturado, não estruturado ou qualquer item intermediário. O objetivo do Data Lake Store é conter os dados brutos para que você possa analisá-los ou transformá-los e movê-los. A seguir estão as principais características do Azure Data Lake:

- Capacidade de armazenamento ilimitada. Um único arquivo pode ter mais de um petabyte: 200 vezes maior do que o que os outros provedores de nuvem oferecem.
- Performance escalonável para acomodar análises massivamente paralelas.
- Você pode armazenar dados em qualquer formato, sem um esquema.

Essa é uma abordagem muito diferente do data warehouse tradicional, no qual você define os esquemas de dados antecipadamente.

Por exemplo, você pode usar um Data Lake Store para armazenar todos os dados que obtêm de seus dispositivos da IoT (Internet das Coisas) que estão coletando dados de temperatura. Você pode deixar os dados no repositório e, depois, filtrá-los e criar uma exibição dos dados por hora ou por semana. Armazenar os dados no Data Lake Store é muito barato. Assim, você pode manter anos de dados a um custo muito baixo.



Experimente [Introdução ao Azure Data Lake Store usando o Portal do Azure.](#)

SQL Data Warehouse do Azure

Você usa o [SQL Data Warehouse do Azure](#) quando precisa de uma solução tradicional de armazenamento de dados que é completamente gerenciada, escalável em tamanho, segura e tem boa performance. Você armazena dados em esquemas predefinidos e os consulta usando o dialeto SQL familiar do SQL Server.

Como o SQL Data Warehouse é executado no Azure, você pode usar recursos avançados, como a detecção automática de ameaças, que usa o aprendizado de máquina para entender os padrões de sua workload e funciona como um sistema de alarme para alertá-lo de uma violação potencial.

Um exemplo de uso do SQL Data Warehouse é quando você sabe quais relatórios deseja mostrar aos usuários e sabe qual é o esquema de dados para esses relatórios. Você pode então criar esquemas no SQL Data Warehouse e populá-los com dados para que os usuários possam navegar pelos dados.



Experimente [Crie um armazenamento de dados SQL Azure.](#)

Adicionando inteligência ao aplicativo

Hospedar seu aplicativo e os dados e mantê-lo escalonável, seguro e com alta performance é bom, mas não seria ótimo se você pudesse adicionar recursos inteligentes a ele?

Azure Search

A pesquisa é um recurso comum na maioria dos aplicativos. No entanto, tradicionalmente, tem sido uma função difícil de implementar. O [Azure Search](#) fornece uma grande parte da estrutura para realizar a pesquisa. Você gera uma instância do Azure Search, cria um índice que o ajuda a pesquisar e o preenche com dados, e pronto. Há muitas opções que você pode usar para ajustar o Azure Search e há muitos recursos excelentes que facilitarão a pesquisa para os usuários:

- Pesquisa geográfica que oferece aos usuários a capacidade de explorar dados com base na proximidade de um resultado de pesquisa a um local físico.
- Analisadores de linguagem da Lucene, bem como NLPs (processadores de linguagem natural) da Microsoft, disponíveis em 56 idiomas para lidar de forma inteligente com a linguística específica do idioma, inclusive tempos verbais, gênero, substantivos com plural irregular (por exemplo, “mouse” versus “mice”), decomposição de palavras, quebra de palavras (para idiomas sem espaços) e muito mais.
- Monitoramento e relatórios que informam sobre o que foi pesquisado e a velocidade e o sucesso da pesquisa.
- Recursos de experiência do usuário como classificação e paginação de resultados de pesquisa e filtragem inteligente e fornecimento de sugestões de pesquisa.

Você pode implementar o Azure Search para ajudar os usuários a pesquisar seu catálogo de produtos em um aplicativo de comércio eletrônico.



Experimente [Crie seu primeiro índice do Azure Search no portal.](#)

Serviços Cognitivos do Azure

Com os [Serviços Cognitivos do Azure](#), você pode adicionar incrível inteligência a seus aplicativos. Os Serviços Cognitivos são uma coleção de APIs que você pode usar para fazer o que às vezes parece ser mágica. Existem mais de 25 APIs nas categorias de Visão, Fala, Linguagem, Conhecimento e Pesquisa. Aqui está um resumo das APIs que você pode usar:

- Usando a [API Facial](#), você pode executar reconhecimento e identificação facial e até mesmo determinar a cor dos olhos de uma pessoa, mesmo que ela use óculos.
- A [API de Emoção](#) determina quais emoções um indivíduo está mostrando em uma imagem.
- A [API de Serviços Inteligentes de Reconhecimento Vocal](#) (ou LUIS) pode realmente entender o texto, com base no modelo que você usa para treiná-la.
- A [API de Tradução de Fala](#) transforma o discurso em texto e o traduz para você em tempo real

Os Serviços Cognitivos obtêm a inteligência por meio de aprendizado de máquina e inteligência artificial, que você pode treinar com seus próprios conjuntos de dados. Esses serviços são simplesmente incríveis e abrem possibilidades que sequer podíamos imaginar há apenas alguns anos.

Serviço de Bot do Azure

O [Serviço de Bot do Azure](#) facilita a criação de um *bot*, um software que pode interagir de forma automática e autônoma com os usuários.

Criar um bot não é uma tarefa trivial. Você precisa acompanhar o contexto de sua interação com o usuário e deve estar pronto para responder a uma infinidade de possíveis parâmetros de interação. O Serviço de Bot o ajuda das seguintes maneiras:

- Isso o ajuda a acompanhar o contexto de interação e fornece modelos para começar do Bot Framework.
- Tem estreita integração com os Serviços Cognitivos, tornando mais fácil tornar seu bot “inteligente”.
- Isso o ajuda a fazer a integração a serviços como Facebook, Slack, Microsoft Teams, Telegram e assim por diante.
- Oferece todos os benefícios de um serviço gerenciado no Azure: escala maciça, entrega contínua embutida (CD) e você paga apenas pelo que você usa.

Um exemplo de um bot que você pode criar com o Serviço de Bot é um que fornece aos usuários respostas às perguntas mais frequentes. A interface do bot pode ser uma caixa de chat em seu site.



Experimente [Introdução ao Chatbot usando o Serviço de Bot do Azure.](#)

Usar eventos e mensagens em seu aplicativo

Os aplicativos modernos e distribuídos globalmente muitas vezes devem lidar com grandes quantidades de mensagens que chegam. Eles precisam ser projetados tendo em mente a dissociação e a escala. O Azure fornece vários serviços para ajudar na ingestão de eventos e padrões de análise e mensagens.

Barramento de Serviço do Azure

O núcleo das mensagens no Azure é o [Barramento de Serviço do Azure](#). O serviço Barramento de Serviço engloba uma coleção de serviços que você pode usar para padrões de mensagens. Os mais importantes entre eles são Tópicos e Filas do Barramento de Serviço do Azure.



Experimente Introdução a [Filas](#) e [Tópicos](#) do Barramento de Serviço do Azure.

Filas do Barramento de Serviço do Azure

Você usa [Filas do Barramento de Serviço do Azure](#) para dissociar sistemas uns dos outros. Aqui está um exemplo: um aplicativo Web recebe pedidos dos usuários e precisa invocar um serviço Web para processar o pedido. O serviço Web levará muito tempo, talvez até cinco minutos, para processar o pedido completamente. Claro, é inaceitável que o aplicativo Web aguarde cinco minutos entre o envio do pedido pelo usuário e a exibição dos comentários. Uma boa maneira de resolver isso é usar uma fila para dissociar o aplicativo Web do serviço Web.

O aplicativo Web recebe o pedido e o grava em uma mensagem em uma Fila do Barramento de Serviço do Azure. Depois de fazer isso, ele pode informar ao usuário que o pedido está sendo processado. O serviço Web obtém as mensagens da fila, uma a uma, e as processa. Quando o serviço Web termina de processar um pedido, envia uma notificação por email ao usuário para avisar que o item foi solicitado.

Dissociando os sistemas, o aplicativo Web pode funcionar a uma velocidade diferente do serviço Web, e ambos podem ser dimensionados individualmente conforme as necessidades dos aplicativos.

A Fila do Barramento de Serviço é um mecanismo simples. Vários aplicativos podem colocar mensagens na fila, mas uma mensagem da fila só pode ser processada por um único aplicativo por vez. Ele tem alguns recursos inteligentes para trabalhar com mensagens na fila, como a detecção de duplicatas e uma subfila de mensagens mortas para a qual as mensagens são movidas quando não são processadas corretamente.

Tópicos do Barramento de Serviço do Azure

Assim como as Filas do Barramento de Serviço, os [Tópicos do Barramento de Serviço do Azure](#) são uma forma de dissociação de aplicativo. Aqui estão as diferenças entre a fila e os tópicos:

- Com uma fila, vários aplicativos gravam mensagens na fila, mas *apenas um aplicativo* de cada vez pode processar uma mensagem.
- Com um tópico, vários aplicativos gravam mensagens no tópico e *vários aplicativos* podem processar uma mensagem ao mesmo tempo.

O Tópico do Barramento de Serviço funciona como uma fila, e vários aplicativos podem processar a mesma mensagem. Os aplicativos podem criar uma assinatura do tópico, que indica o tipo de mensagens em que estão interessados. Assim como as filas, os tópicos têm recursos como detecção

de duplicatas e uma subfila de mensagens mortas para a qual as mensagens são movidas quando não são processadas corretamente.

Comparando filas de barramento de serviço e filas de armazenamento Azure

Existem Filas do Barramento de serviço, mas também há Filas do Armazenamento do Azure. Basicamente, fazem o mesmo, mas há diferenças sutis entre elas. A Tabela 3-1 examina cada uma delas.

Tabela 3-1: Uma comparação de recursos entre Filas do Barramento de Serviço e Filas de Armazenamento

Filas do Barramento de Serviço do Azure	Filas de Armazenamento do Azure
Vida útil da mensagem >7 dias	Vida útil da mensagem <7 dias
Ordenação garantida (primeira a entrar, primeira a sair)	Tamanho da fila >80 GB
Detecção de duplicatas	Logs de transações
Tamanho da mensagem ≤1 MB	Tamanho da mensagem ≤64 KB

Hubs de Eventos do Azure

Em nosso mundo de software onipresente, os dados são gerados de muitos aplicativos, dispositivos e locais o tempo todo. Hoje, é cada vez mais comum que as empresas capturem grandes quantidades de dados para analisá-los ou transformá-los e movê-los para uso posterior. Os [Hubs de Eventos do Azure](#) podem ajudar.

Os Hubs de Eventos são projetados para ingestão de grande volume de dados. Você pode incluir milhões de mensagens por segundo, e eles lidam com os dados sem esforço. Eles podem reter mensagens por até sete dias ou mantê-las indefinidamente, gravando-as em um repositório de dados usando o recurso Captura de Hubs de Eventos.

Você pode usar os Hubs de Eventos para filtrar os dados com consultas, à medida que chegam, e enviá-los para um repositório de dados como o Azure Cosmos DB. Você pode até mesmo reproduzir as mensagens, se precisar.



Experimente [Introdução ao envio de mensagens para os Hubs de Eventos do Azure no .NET Standard.](#)

Hub IoT do Azure

Assim como os Hubs de Eventos, o [Hub IoT do Azure](#) foi criado para a ingestão de grande volume de dados. É especificamente voltado para lidar com o enorme volume de mensagens de dados de dispositivos da Internet das Coisas (IoT; consulte a barra lateral no Capítulo 1, “Quem usa o Azure?”, por exemplo), como termostatos inteligentes ou sensores em carros. Tem muitas das mesmas propriedades que os Hubs de Eventos, como a capacidade de reter mensagens por até sete dias e reproduzi-las.

O que torna o Hub IoT único é que, além de receber mensagens de dispositivos, ele também pode enviar mensagens para eles. Tem a capacidade de gerenciar sua infraestrutura completa de IoT. Você pode usar o Hub IoT para registrar dispositivos e informar seu estado, gerenciá-los protegendo-os e reiniciando-os e enviando dados para dispositivos.



Experimente [Conecte seu dispositivo ao hub IoT usando o .NET.](#)

Grade de Eventos do Azure

A [Grade de Eventos do Azure](#) oferece um tipo diferente de mensagens. A Grade de Eventos oferece um serviço de publicação/inscrição totalmente gerenciado que se encaixa em quase todos os serviços do Azure e pode se conectar a editores e assinantes personalizados.

Isso é diferente de trabalhar com Filas e Tópicos do Barramento de Serviço, nos quais você precisa sondar a Fila ou Tópicos para verificar se há novas mensagens. As Grades de Eventos enviam mensagens automaticamente aos assinantes, sendo assim um serviço de eventos reativo em tempo real.

Os serviços dentro e fora do Azure publicam eventos (por exemplo, quando um novo Blob é adicionado ou quando um novo usuário é adicionado a uma assinatura do Azure). A Grade de Eventos do Azure pode detectar esses eventos e disponibilizá-los para manipuladores de eventos e serviços que assinaram os eventos (Figura 3-1). Os manipuladores de eventos podem ser itens como Funções do Azure ou Aplicativos Lógicos do Azure, que podem atuar sobre os dados no evento.

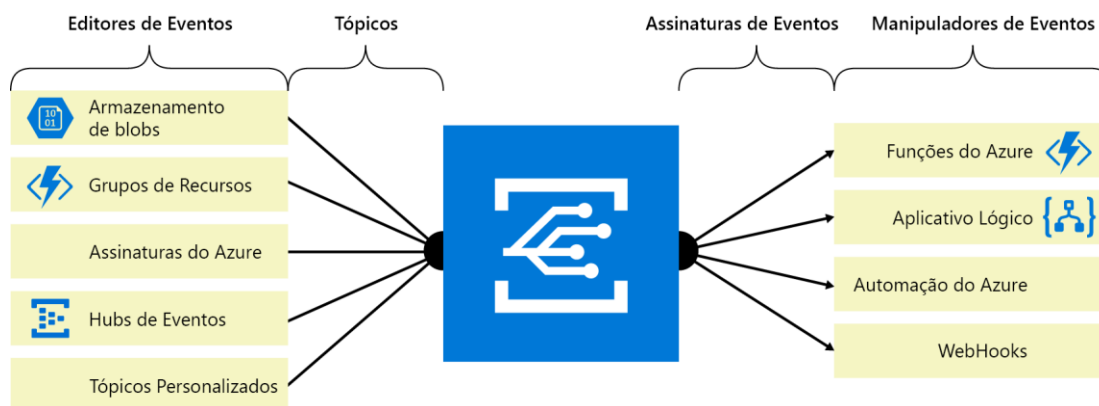


Figura 3-1: Visão geral da Grade de Eventos do Azure

Outro aspecto importante da Grade de Eventos é que ela não tem servidor. Isso significa que, como os Aplicativos Lógicos do Azure e as Funções do Azure, a Grade de Eventos é dimensionada automaticamente e não é preciso que uma instância dela seja implantada. Você apenas a configura e utiliza. E você paga apenas quando ela é usada, não apenas por ela estar presente.

Aqui está um exemplo de uso da Grade de Eventos do Azure:

Você quer ser notificado por email sempre que um usuário for adicionado ou excluído em sua lista de endereçamento no MailChimp.

A Grade de Eventos do Azure é usada para ativar um aplicativo nos Aplicativos Lógicos do Azure e é configurada para ouvir as alterações na lista de endereçamento do MailChimp. A próxima etapa é enviar um email contendo o nome do usuário que foi adicionado ou excluído e a ação que foi executada (adicionar ou excluir). Agora, quando um novo usuário é adicionado à lista de endereçamento, a Grade de Eventos sinaliza isso para os Aplicativos Lógicos, que enviam um email.



Experimente [Monitore alterações de máquinas virtuais com a Grade de Eventos do Azure e o Aplicativo Lógico.](#)

O que usar quando?

O Azure oferece inúmeras opções para enviar mensagens e desassociar aplicativos. Qual delas você deve usar e quando? A Tabela 3-2 resume as diferenças para ajudá-lo a decidir.

Tabela 3-2: Quando usar qual serviço do Azure para mensagens ou eventos

	Grade de Eventos	Hubs de Eventos	Hub IoT	Tópicos	Filas do Barramento de Serviço	Filas de Armazenamento
Ingestão de eventos	X	X	X			
Gerenciamento de dispositivos			X			
Mensagens	X	X	X	X	X	X
Vários consumidores	X	X	X	X		
Vários remetentes	X	X	X	X	X	X
Uso para dissociação		X	X	X	X	X
Uso para publicar/assinar	X					
Tamanho máximo da mensagem	64 KB	256 KB	256 KB	1 MB	1 MB	64 KB

Proteger seu aplicativo

Hoje a segurança é um dos aspectos mais importantes de qualquer aplicativo, e não é simples fazer tudo corretamente. Felizmente, o Azure oferece muitos serviços para ajudá-lo a proteger seu aplicativo. Examinamos alguns deles neste capítulo.

Azure Active Directory

Uma parte importante da segurança de seu aplicativo é autenticar os usuários antes que eles possam usá-lo. A autenticação não é fácil de implementar. Você precisa armazenar identidades e credenciais de usuários em algum lugar, implementar gerenciamento de senhas, criar um handshake de autenticação segura e assim por diante.

O [Azure Active Directory](#) oferece tudo isso e muito mais pronto para uso. Você armazena identidades de usuário no Azure Active Directory e faz com que os usuários se autentiquem em relação a elas, redirecionando-os para seu aplicativo somente depois de serem autenticados. O Azure Active Directory cuida do gerenciamento de senhas, inclusive cenários comuns como "Esqueci minha senha".

O Azure Active Directory é usado por milhões de aplicativos todos os dias, inclusive o [Portal do Azure](#), o [Outlook.com](#) e o [Office 365](#). Por isso, pode detectar mais facilmente o comportamento mal-intencionado e tomar medidas em relação a ele. Por exemplo, se um usuário entrar em um aplicativo de uma localização na Europa e, um minuto depois, entrar na Austrália, o Azure Active Directory indicará isso como um comportamento mal-intencionado e pedirá ao usuário credenciais adicionais por meio da autenticação multifatorial.

Gerenciamento de API do Azure

As APIs devem ser seguras. Isso se aplica às APIs que você mesmo cria, bem como às que você consome de fornecedores de terceiros. Para ajudar a tornar as APIs seguras, você pode usar o [Gerenciamento de API do Azure](#). Esse é basicamente um proxy que você coloca na frente de

suas APIs ou daquelas de fornecedores de terceiros, e que adiciona recursos como armazenamento em cache, limitação e autenticação/autorização.

Você pode proteger a API com o Gerenciamento de API, exigindo que os consumidores criem uma assinatura para a API. Dessa forma, os aplicativos de chamada precisam se autenticar para que possam usar a API. Você pode usar vários métodos de autenticação, como tokens de acesso, autenticação básica e certificados. Além disso, pode controlar quem está chamando a API e bloquear chamadores indesejados.



Experimente [Introdução ao Gerenciamento de API do Azure.](#)

Azure Key Vault

Como parte de sua arquitetura de segurança, você precisa de um lugar seguro para armazenar certificados, chaves e outros segredos. O [Azure Key Vault](#) fornece isso. Com o Key Vault, você tem uma localização central onde armazena os segredos que seus aplicativos usam.

Podem ser as credenciais de um connectionstring que o aplicativo usa. O aplicativo obtém o connectionstring do Key Vault em vez de obtê-lo do sistema de configuração. Dessa forma, os administradores podem controlar os segredos e os desenvolvedores nunca precisam lidar com eles.

Você também usa o Key Vault para armazenar certificados como os certificados SSL (Secure Sockets Layer) que usa para proteger o tráfego de e para seus aplicativos por meio de HTTPS.



Experimente [Introdução ao Azure Key Vault.](#)

Proteção contra DDoS do Azure

Você já ouviu falar a respeito muitas vezes no noticiário e, certamente, não quer que isso aconteça com sua empresa: seu aplicativo é atacado por um ataque de DDoS (Negação de Serviço Distribuído). Esses tipos de ataques estão se tornando mais comuns e podem sobrecarregar o aplicativo até o ponto em que ninguém mais possa usá-lo.

O Azure protege você dos ataques de DDoS com a camada de Proteção contra DDoS do Azure. Essa é uma camada na rede física do Azure que não é acessível aos clientes. Faz parte da plataforma Azure, e você a obtém gratuitamente com todos os seus serviços. Você não precisa fazer nem configurar nada para usá-la. Ela está sempre lá para protegê-lo.



Mais informações [Leia mais sobre a Proteção contra DDoS do Azure.](#)

Firewall do Aplicativo Web do Azure

Você precisa proteger seu aplicativo contra muitas ameaças possíveis, inclusive as definidas na especificação OWASP (Open Web Application Security Project), como injeção de SQL e XSS (Cross-Site Scripting). O [Firewall do Aplicativo Web do Azure](#) pode ajudar. O Firewall do Aplicativo Web, um recurso do serviço [Gateway de Aplicativo do Azure](#), fornece proteção em tempo real ao aplicativo.

Ele pode detectar um ataque mal-intencionado, conforme definido no [conjunto de regras principais do OWASP](#), e bloquear esse ataque para que não alcance o aplicativo. Ele também informa sobre os ataques que foram tentados ou estão em andamento para que você possa ver quais ameaças estão tentando chegar ao aplicativo. Isso fornece uma camada extra de segurança que protege seus aplicativos.

Criptografia padrão de dados

Você protege os repositórios de dados da melhor forma possível tendo um `connectionstrings` seguro e regras de firewall em vigor. Porém, se um invasor puder acessar seus dados de alguma forma, convém torná-los inúteis por meio da criptografia.

Por padrão, seus dados são criptografados quando você os armazena no [Banco de Dados SQL do Azure](#), no [SQL Data Warehouse do Azure](#), no [Banco de Dados do Azure para MySQL](#), no [Banco de Dados do Azure para PostgreSQL](#), no [Armazenamento do Azure](#), no [Azure Cosmos DB](#) ou no [Azure Data Lake Store](#). E isso tudo funciona automaticamente. Você não precisa fazer nada para configurar esses recursos ou criptografar ou descriptografar dados aos utilizar.

Onde e como implantar seus serviços do Azure

Os serviços do Azure com os quais você escolhe trabalhar determinam suas opções de implantação e vice-versa. Portanto, é importante compreender as opções de implantação que você tem no Azure e quais são suas ramificações.

A mentalidade de DevOps

Para criar um ótimo software, é crucial entregar rápido e reiterar depressa. Portanto, o novo código deve ser mesclado ao código de sua equipe. Deve ser implantado e testado tão rápido quanto possível para ver se tudo funciona e se o que você criou é aquilo de que o usuário precisa.

Muitos dos serviços do Azure que já examinamos neste livro podem ter o código automaticamente entregue a eles, muitas vezes por meio de um pipeline de [CD \(entrega contínua\)](#) que você configura no serviço.

Além dos recursos nativos dos serviços do Azure, você pode usar o Visual Studio Team Services para criar, testar e implantar seu aplicativo. Você [pode facilmente criar novos pipelines de compilação e implantação no Visual Studio Team Services](#), além de fazer tarefas como automatizar o teste de carga e trocar slots de implantação na produção.



Experimente [Introdução ao Visual Studio Team Services.](#)

Modelos do Azure Resource Manager

Todos os serviços do Azure introduzidos neste guia são baseados no Azure Resource Manager. Podemos usar isso para criar scripts de nosso ambiente em IaC (Infraestrutura como Código), usando [modelos do Resource Manager](#). Esses modelos são arquivos JSON que descrevem o que você deseja implantar e quais são seus parâmetros.

Você pode criar modelos do Resource Manager no Visual Studio e no Visual Studio Code usando o modelo de projeto do Grupo de Recursos do Azure. Você também pode gerar modelos do Resource Manager no Portal do Azure clicando no botão Script de Automação, que está disponível na barra de menu de todos os recursos no Portal do Azure. Isso gera o modelo do Resource Manager para o recurso fornecido e até gera código para criar o recurso usando a CLI (interface de linha de comando) do Azure, o Windows PowerShell, o .NET e o Ruby.

Depois de obter um Modelo do Resource Manager, você pode implantá-lo no Azure usando o Windows PowerShell, a CLI do Azure ou o Visual Studio. Ou, você pode automatizar sua implantação em um pipeline de CD usando o Visual Studio Team Services.

Você usa os modelos do Resource Manager para implantar aplicativos para serem executados na plataforma Azure, seja na nuvem pública ou na infraestrutura local, no Azure Stack.

Um excelente exemplo de implantação de recursos na nuvem usando o Resource Manager é o botão Implantar no Azure, que você pode encontrar em muitos repositórios do GitHub, conforme ilustrado na Figura 5-1.



Figura 5-1: Implantação de um botão de um Aplicativo Lógico do Azure usando um modelo do Resource Manager no GitHub

Azure Service Fabric

Outra maneira de executar seus aplicativos é executá-los no [Azure Service Fabric](#). O Service Fabric é o fundamento que a Microsoft usa para executar muitos recursos do Azure e para torná-los altamente disponíveis, com boa performance e correção automática. Agora você mesmo pode usar o Service Fabric para hospedar seus próprios serviços.

Você pode usar o SDK do Azure Service Fabric para criar aplicativos para Service Fabric. Também pode executar qualquer executável no Service Fabric e pode até usá-lo para hospedar contêineres.

O Service Fabric é ótimo para tornar seus aplicativos tão competitivos, confiáveis e seguros quanto muitos serviços do Azure, e você pode usá-lo em qualquer lugar. Você pode implantar o Service Fabric no Azure, na infraestrutura local, em seu próprio computador e até mesmo em VMs (Máquinas Virtuais) em outras nuvens.

Depois de implantar seu aplicativo no Azure Service Fabric, ele fornece benefícios como estes:

- Balanceamento de carga
- Escala automática
- Alta disponibilidade
- Correção automática

- Replicação e failover
- Atualizações sem interrupção
- E reversão automática

Contêineres no Azure

Os contêineres são uma das palavras-chave da tecnologia que estão sempre nas notícias. Porém, eles são mais do que apenas uma novidade: realmente são muito úteis para executar seus aplicativos. Um contêiner é, basicamente, uma VM leve que é iniciada e interrompida muito mais rápido do que uma VM. Portanto, é muito mais útil para desenvolvimento, teste e execução de aplicativos na produção.

O principal benefício dos contêineres é que um contêiner individual é sempre o mesmo. Você executa um contêiner localmente quando desenvolve o aplicativo e usa a mesma configuração de contêiner na nuvem ou em qualquer outro lugar. Toda a sua equipe usa exatamente a mesma configuração de contêiner. Assim, você sabe que a infraestrutura é a mesma para todos e na produção. Com contêineres, a antiga afirmativa dos desenvolvedores, “isso funciona na minha máquina”, agora significa que o item também funcionará na produção.

Existem muitas tecnologias para executar contêineres, e o [Docker](#) é uma delas. O Azure pode executar e gerenciar contêineres para você com as [Instâncias de Contêiner do Azure](#) e o [Serviço de Contêiner do Azure](#), até mesmo no Services Fabric e no [Serviço de Aplicativo do Azure executado no Linux](#).

A Tabela 5-1 mostra o serviço que você pode usar para determinado cenário ao usar contêineres.

Tabela 5-1: Escolher qual serviço do Azure usar para contêineres

	Serviços de Contêiner do Azure	Instâncias de Contêiner do Azure	Azure Service Fabric
Para implantações de produção de sistemas complexos (com um orquestrador de contêiner)	X		
Para executar configurações simples (possivelmente sem orquestrador)		X	
Para workloads de longa duração em contêineres	X		
Para workloads de curta duração em contêineres		X	
Para orquestrar um sistema baseado em contêineres	X		X
Orquestrar com orquestradores de software livre (DC/OS , Docker Swarm , Kubernetes)	X		
Orquestrar com orquestrador interno			X



Observação Tenha em mente que, ao usar contêineres, você usa um produto de IaaS (infraestrutura como serviço) e que é responsável pelo SO (sistema operacional), correções, balanceamento de carga e assim por diante.

Azure Stack

Se é preciso que os aplicativos e dados permaneçam na infraestrutura local, mas você ainda quer se beneficiar da capacidade que o Azure tem a oferecer, o [Azure Stack](#) é o produto para você. Único na indústria, o Azure Stack é uma extensão do Azure que você hospeda em seu próprio ambiente. Essencialmente, é o Azure em uma caixa.

Você usa o Azure Stack da mesma forma como faz com o Azure, com a mesma experiência do Portal do Azure e as mesmas APIs com as quais pode usar a CLI do Azure, o Windows PowerShell ou seu IDE favorito.

Você pode executar itens como o Serviço de Aplicativo do Azure e Máquinas Virtuais do Azure no Azure Stack. Tudo é exatamente como na nuvem pública, só que agora você executa na infraestrutura local. E se decidir mudar para a nuvem pública, você simplesmente enviará os serviços do Azure Stack para o Azure.

Onde implantar e quando?

A Tabela 5-2 resume as opções de implantação do Azure. Observe que o uso de todos os serviços de PaaS (plataforma como serviço) que discutimos neste guia é possível somente ao usar a Nuvem Pública com modelos do Resource Manager ou na infraestrutura local com o Azure Stack.

Tabela 5-2: Comparar as opções de implantação do Azure

	IaaS	PaaS
Na infraestrutura local	Azure Stack Contêineres Service Fabric	Azure Stack Service Fabric
Nuvem pública	Contêineres Service Fabric	Service Fabric Modelos do Resource Manager

Um passo a passo do Azure

Agora que você sabe o que é o Azure e aprendeu sobre os serviços que ele tem a oferecer, vamos começar a usá-lo.

Passo a passo nº1: o Portal do Azure

Uma das mais importantes ferramentas do Azure é o Portal do Azure, que você pode encontrar em <https://portal.azure.com/>. Esse é seu hub central do Azure, onde você faz tudo que desejar com o Azure. A maioria das tarefas que você pode realizar no Portal do Azure também podem ser feitas por meio da API Azure, da CLI (interface de linha de comando) do Azure e do Azure PowerShell.

Você pode usar uma visualização do Portal do Azure acessando <https://preview.portal.azure.com>. A visualização já contém recursos que ainda não estão com disponibilidade geral, mas que você pode experimentar e fornecer comentários. O restante deste guia se concentrará na versão com disponibilidade geral do portal, não na visualização.

O primeiro item que você verá no Portal do Azure é um painel com blocos. Você pode criar e personalizar painéis e compartilhá-los com os membros da equipe ou mantê-los apenas para você.

Blocos no Portal do Azure

Os painéis contêm blocos que exibem informações de um serviço ou funcionam como atalho para um serviço. Você pode encontrar essas blocos, mostrados na Figura 6-1, em todo o portal, nas páginas de todos os serviços. Eles podem ser muito úteis para obter uma rápida visão geral de como um serviço está se saindo.

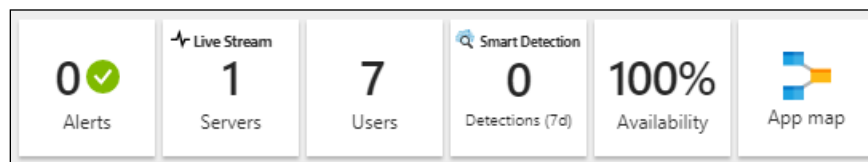


Figura 6-1: Blocos informativos no Portal do Azure

Por exemplo, o serviço Application Insights, por padrão, mostra blocos que informam se houver algum alerta ativo, se houver dados chegando, quantos usuários estiveram ativos nas últimas 24 horas, qual a disponibilidade e assim por diante. Essas são informações muito úteis em um formato muito fácil de consumir. Você pode personalizar o tamanho e as informações dos blocos. Também pode personalizar a aparência dos gráficos, ajustando suas linhas de tempo e fazendo com que eles mostrem dados em diferentes formatos, como linhas ou barras.

Você também pode anexar blocos (Figura 6-2) diretamente aos seus painéis para que sejam a primeira coisa que você vê ao inserir o portal. É possível, por exemplo, fixar blocos de todas as métricas de serviço que você considera importantes, a fim de criar um painel de monitoramento que você [compartilha com sua equipe](#) ou exibe em um monitor na sala da equipe.

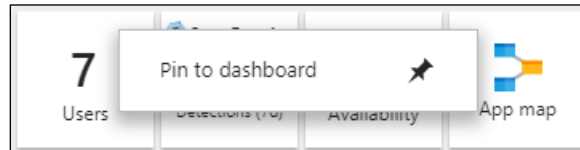


Figura 6-2: Você pode fixar blocos em um painel

Encontrar serviços

Os serviços do Azure são o tema central do Portal do Azure. Você pode adicioná-los e encontrá-los de várias maneiras.

Para criar novos serviços, no canto superior esquerdo da janela do portal, clique no sinal de mais ("+"). Isso abre a caixa de pesquisa do marketplace, em que você encontrará tudo, desde o Aplicativo Web até Servidores Linux, conforme mostrado na Figura 6-3.

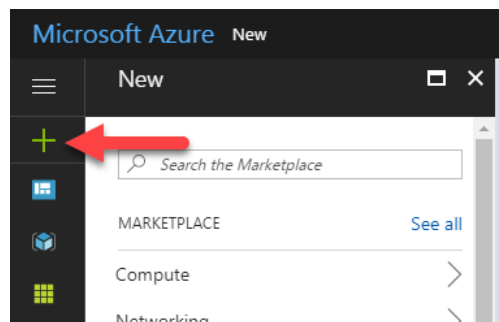


Figura 6-3: O botão Novo Recurso no Portal do Azure

Quando você encontra o serviço desejado (Figura 6-4), um assistente orienta você para configurá-lo e implantá-lo.

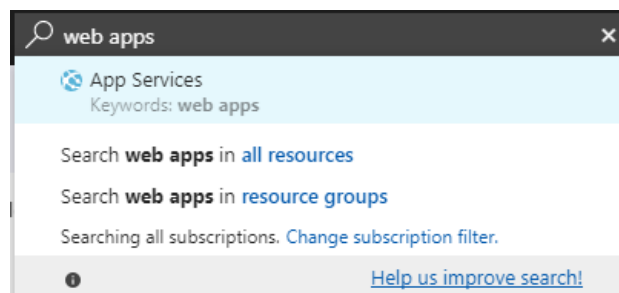


Figura 6-4: Resultados de pesquisa de recursos

Quando tem alguns recursos, você pode encontrá-los por meio da pesquisa. Você pode usar a caixa de pesquisa na parte superior do portal (Figura 6-5) para pesquisar todos os seus recursos e ser levado diretamente para eles.

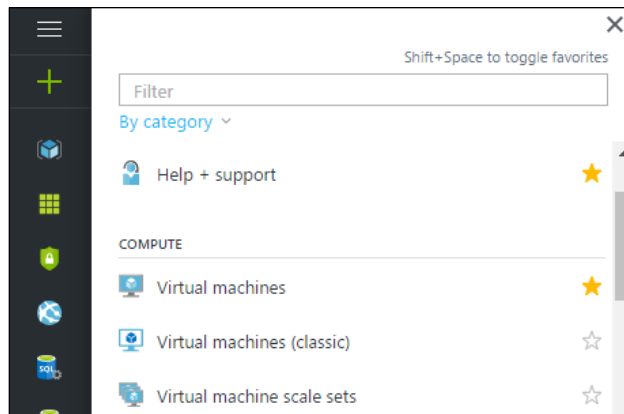


Figura 6-5: O menu de favoritos no Portal do Azure

No painel no lado esquerdo do portal está o menu de favoritos. Esse menu exibe as categorias de recursos (representadas por seus ícones), como o Serviço de Aplicativo do Azure. Você pode reorganizar os ícones arrastando-os para cima e para baixo. Também pode selecionar quais deseja ver expandindo o menu de favoritos e clicando nas estrelas dessas categorias.

Entender folhas

As páginas no Azure também são chamadas de *folhas*. As folhas estão em toda parte, e você pode até colocá-las em seus painéis. Quando você abre um aplicativo Web, primeiro vê a folha de visão geral. Essa folha específica fornece ferramentas para parar, iniciar e reiniciar o aplicativo Web e exibe telas mostrando suas métricas, como número de solicitações e erros. Quando você escolhe outro item de menu, uma nova folha é aberta. As folhas sempre são abertas em contexto. Portanto, se você abrir a folha de Slots de Implantação e clicar no botão Criar Novo Slot de Implantação, uma nova folha aparecerá à direita da folha de Slots de Implantação, preservando o contexto em que você está.

Criar uma nova máquina virtual

Vamos usar o Portal do Azure para criar uma nova VM (máquina virtual) que executa o Red Hat Linux. Depois de criá-la, mostraremos como desligá-la e removê-la para que você não pague mais por ela.

Uma explicação sobre grupos de recursos

A VM será implantada em um *grupo de recursos*. Todos os recursos do Azure residem dentro de grupos de recursos. Um grupo de recursos é um contêiner lógico que contém seus recursos. Você pode gerenciar a segurança de um grupo de recursos e pode ver quais são os recursos do grupo. É uma prática comum agrupar os serviços relacionados em um grupo de recursos para que eles sejam fáceis de proteger e você possa facilmente descobrir quanto eles custam.

1. No Portal do Azure, no canto superior esquerdo, clique no botão Criar um Novo Serviço.
2. Na caixa Pesquisar, digite **máquina virtual Linux**.

Você obterá muitos resultados de pesquisa, inclusive o Red Hat Enterprise Linux. Vamos experimentar sua versão mais recente.

3. Clique na última versão do Red Hat Enterprise Linux.

4. Clique em Criar.

O assistente Criar Máquina Virtual é aberto.

- a. Escolha um nome para a VM.
- b. Escolha o tipo de unidade. O SSD fornece uma VM mais rápida, mas também é mais caro. Para este passo a passo, escolha SSD.
- c. Digite um nome de usuário.
- d. Selecione Senha para o Tipo de autenticação.
- e. Digite uma senha e confirme.
- f. Na caixa Grupo de Recursos, digite um novo nome.
- g. Escolha a localização da VM e clique em OK.

5. Escolha o tamanho da VM.

A performance da VM determina o quanto você paga. Existem muitas opções de dimensionamento para VMs, algumas pequenas, outras incrivelmente grandes. Você pode usar o assistente para selecionar quantos núcleos e quanta memória deseja e escolher as opções com base nisso. Além disso, existem outros recursos que têm opções de tamanho:

- a. Tipo de disco rígido (SSD ou HDD normal).
- b. A quantidade máxima de IOPS (Operações de Entrada/Saída por Segundo). Isso determina a performance da VM de forma significativa, especialmente se os aplicativos que você executa leem e gravam muito de e para o disco rígido.
- c. A quantidade de unidades de dados que podem ser instaladas na VM.
- d. A capacidade de fazer balanceamento de carga.
- e. A placa gráfica instalada na VM. Isso é útil se você precisa executar uma grande quantidade de renderização de gráficos ou workloads computacionais pesados.

6. Depois de selecionar o tamanho, você pode definir configurações adicionais, como a rede virtual e o endereço IP, e instalar extensões no computador. Por enquanto, deixe tudo como está e clique em OK.

7. Examine o resumo, aceite os termos e clique em Comprar.

A VM será agora implantada. Isso geralmente leva apenas alguns minutos.

Conecte-se à VM usando o Secure Shell

Agora que você criou uma VM com o Red Hat Linux, pode se conectar a ela e começar a usá-la. Você pode usar várias ferramentas, como SSH (Secure Shell) ou até mesmo a [área de trabalho remota para a VM](#). Vamos usar o SSH para conectar-se à VM usando o Cloud Shell no Portal do Azure.

O Cloud Shell é a CLI do Azure no navegador. Não é mais necessário instalar nada em seu computador. O Cloud Shell usa a CLI do Azure versão 2.0. Você pode usar o Bash ou o PowerShell no Cloud Shell para interagir com seus recursos do Azure.

Para iniciar o Cloud Shell, no canto superior direito do Portal do Azure, clique no botão mostrado aqui:



Para se conectar à VM, você precisa de seu endereço IP. Encontre a VM no Portal do Azure (procurando por ela ou navegando até ela). Na folha Visão geral da VM, no canto superior esquerdo, clique no botão conectar (consulte a Figura 6-6). Quando você clica nele, a folha exibe o endereço IP a ser usado para se conectar à VM, conforme mostrado na Figura 6-6. Copie o endereço IP.

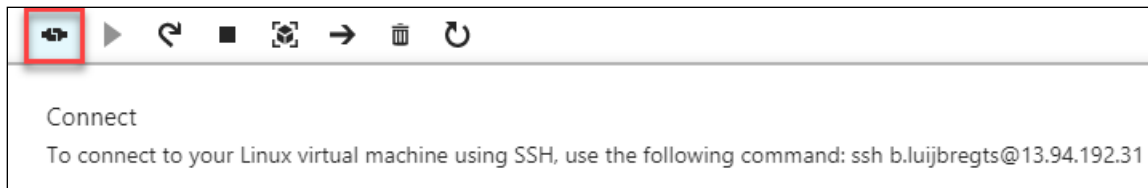


Figura 6-6: Conectar-se a uma VM no Portal do Azure

Agora, você pode usar esse comando diretamente no Cloud Shell: basta colá-lo e pressionar Enter. Você verá um aviso de que não há nenhuma chave de autenticação. Isso ocorre porque escolhemos trabalhar com uma senha, em vez disso. Escolha Sim, digite sua senha e pronto! Agora você está conectado ao Red Hat Linux enterprise e pode começar a usá-lo. A Figura 6-7 mostra que estamos conectados à Máquina Virtual por meio do Cloud Shell e atravessamos o sistema de arquivos:

```
Bash | [Power] [?] [Settings]
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

barry@Azure:~$ ssh b.luijbregts@13.95.29.180
Password:
Last login: Thu Sep 28 07:00:23 2017 from 52.170.90.72
[b.luijbregts@myredhatserver ~]$ ls -al /etc
total 1132
drwxr-xr-x. 84 root root    8192 Sep 28 06:54 .
dr-xr-xr-x. 17 root root    224 Aug  9 23:05 ..
drwxr-xr-x.  3 root root    101 Aug  9 23:01 abrt
-rw-r--r--.  1 root root     16 Aug  9 23:03 adjtime
-rw-r--r--.  1 root root   1518 Jun  7  2013 aliases
drwxr-xr-x.  2 root root     69 Aug  9 23:05 alternatives
-rw-----.  1 root root    541 Mar 29 16:34 anacrontab
-rw-r--r--.  1 root root     55 Mar  1  2017 asound.conf
-rw-r--r--.  1 root root      1 May 23  2016 at.deny
```

Figura 6-7: Percorrendo o sistema de arquivos Linux usando o Cloud Shell no Portal do Azure

Adicionar armazenamento à VM

Se está usando sua VM como um servidor de banco de dados, você geralmente quer ter espaço de armazenamento extra para lidar com os dados do banco de dados. Você pode criar espaço de armazenamento extra incluindo uma unidade adicional por meio do Portal do Azure.

1. Vá para a VM, selecione a guia Discos e clique em Adicionar Disco de Dados, conforme mostrado na Figura 6-8.

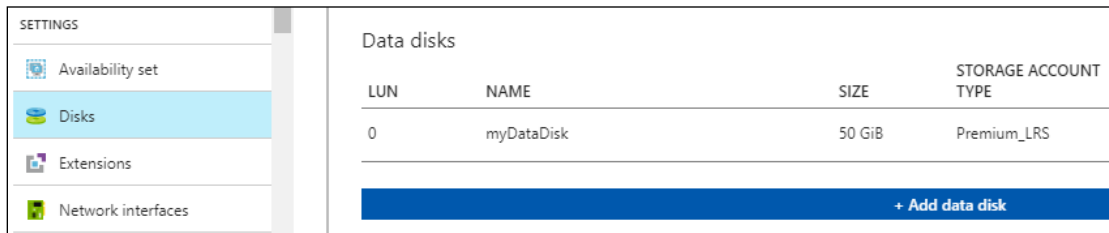


Figura 6-8: A guia Disco de uma VM no Portal do Azure

Como não temos discos existentes para anexar, selecione Criar Disco.

2. É aberto o Assistente para Criar Disco Gerenciado. Use esse assistente para criar uma unidade gerenciada (em vez de uma unidade não gerenciada).

Os discos gerenciados são completamente gerenciados pelo Azure. Você não precisa criar e manter uma conta de armazenamento subjacente para eles. Opcionalmente, você pode escolher o tamanho da unidade e o que deve estar nela, mas vamos deixar tudo como está. Digite um nome e clique em Criar. Agora você tem uma unidade adicional anexada à sua VM.

Você também pode criar e anexar uma nova unidade via Cloud Shell ou CLI do Azure usando este comando:

```
az vm disk attach -g myResourceGroup --vm-name myVM --disk myDataDisk --new --size-gb 50
```

Se precisar de ajuda com a CLI do Azure, você poderá digitar **az -h**. Todos os comandos da CLI do Azure seguem a mesma estrutura: **az** **noun(s)** (por exemplo, **vm**) **verb** (por exemplo, **attach**) **options**.

Depois de criar uma nova unidade para sua VM, você precisa inicializá-la no Linux. [Este documento](#) orienta você durante o processo.

A criação de unidades adicionais para o armazenamento extra de VM é simples. As etapas depois disso podem ser mais complicadas, dependendo do sistema operacional que você usa.

Limpar os recursos de passagem

Se terminou de usar a VM, você pode desligá-la e removê-la excluindo o grupo de recursos que criamos no início da criação da VM. Isso contém a VM e todos os outros recursos que são criados automaticamente. Depois que o grupo de recursos é excluído, você não paga mais por nenhum dos recursos que usou neste passo a passo.

Passo a passo nº 2: desenvolver um aplicativo Web Node.js com o MongoDB no Azure

Neste passo a passo, implementamos um aplicativo Node.js simples que se conecta ao MongoDB e hospeda isso no Azure. Vamos hospedar o aplicativo no Aplicativo Web do Azure e usar o Azure Cosmos DB como nosso banco de dados, pois ele tem uma interface do MongoDB.

Para acompanhar, você precisará ter a [CLI do Azure \(v 2.0 ou superior\)](#), o [Git \(v2 ou superior\)](#), o [node.js e o NPM \(v6 ou superior\)](#) e o [MongoDB \(3.4 ou superior\)](#) instalados no computador.

Usaremos um aplicativo de exemplo que usa a estrutura Web [MEAN.js](#).

Criar o aplicativo Web e o banco de dados usando o Portal do Azure

Para hospedar o aplicativo Node.js, criaremos um novo aplicativo Web no Aplicativo Web do Azure.

1. No Portal do Azure, clique no botão Criar um Novo Serviço.
2. Procure por Aplicativo Web. A folha do Aplicativo Web é aberta. Clique em Criar para começar.
3. A folha Criar Aplicativo Web é aberta.
 - a. Digite um nome para o Aplicativo Web.
 - b. Crie um novo grupo de recursos, dando-lhe um nome.
 - c. Selecione o sistema operacional para a execução (Windows ou Linux).
 - d. Selecione ou crie um Plano de Serviço de Aplicativo e clique em Criar.

O Aplicativo Web é executado em Planos de Serviços de Aplicativos

Os Serviços de Aplicativos, assim como o Aplicativo Web, são executados em Planos de Serviços de Aplicativos do Azure. Um Plano de Serviço de Aplicativo é uma abstração de recursos, como CPU e memória, e é representado em camadas de preços. Os Planos de Serviços de Aplicativos também estão vinculados a uma região geográfica específica que você escolhe. Você pode, por exemplo, executar o aplicativo do Aplicativo Web em um Plano de Serviço de Aplicativo da camada de preços S1 (consulte a Figura 6-9), que tem um núcleo e 1,75 GB de RAM.

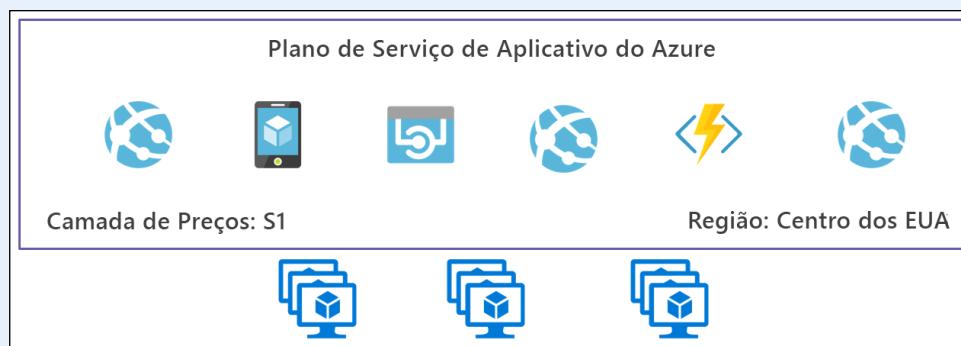


Figura 6-9: Plano de Serviço de Aplicativo do Azure explicado

Você pode executar tantos Serviços de Aplicativos em um Plano de Serviço de Aplicativo quanto desejar, contanto que perceba que precisa compartilhar os recursos entre todos os Serviços de Aplicativos.

Para hospedar o banco de dados, criaremos um banco de dados do Cosmos DB que usa a API do MongoDB. Isso funcionará exatamente do mesmo modo que o MongoDB faria e tem todos os benefícios que o Cosmos DB oferece.

1. No Portal do Azure, clique no botão Criar um Novo Serviço.
2. Procure o Cosmos DB. A folha do Cosmos DB é aberta. Clique em Criar para começar.
 - a. Digite um nome (ID) para a conta do banco de dados.
 - b. Escolha a API do MongoDB.
 - c. Selecione o grupo de recursos que você criou para o aplicativo Web.
 - d. Escolha o local. De preferência, deve ser um local que seja o mesmo ou esteja próximo de onde o aplicativo Web está.
 - e. Clique em Criar.

Execute o aplicativo Node.js e o banco de dados do MongoDB localmente

Vamos executar o aplicativo localmente, antes de executá-lo no Azure.

Se você ainda não o fez, [instale o MongoDB Community edition](#):

1. Abra um prompt de comando no computador e navegue até o diretório bin da instalação do MongoDB. Isso pode ser algo como C:\Program Files\MongoDB\Server\3.4\bin.
2. Digite **mongo** para se conectar ao servidor do MongoDB local.
3. Se ele falhar, verifique se [seu banco de dados do MongoDB foi iniciado](#). Muitas vezes, você precisa iniciá-lo executando mongod.

Agora, faça o seguinte para clonar o aplicativo inicial do Node.js e executá-lo:

4. Na janela de comando, navegue até um diretório para o qual deseja que a origem do aplicativo seja baixada.
5. Execute `git clone https://github.com/Azure-Samples/meanjs.git` para clonar o aplicativo.
6. Inicie o aplicativo executando os seguintes comandos:

```
cd meanjs
npm install
npm start
```

7. Quando ele for totalmente carregado, você verá uma mensagem que contém a URL do aplicativo. Abra um navegador e navegue até o endereço do servidor (algo como `http://localhost:3000`). Isso deve ser semelhante à figura 6-10.

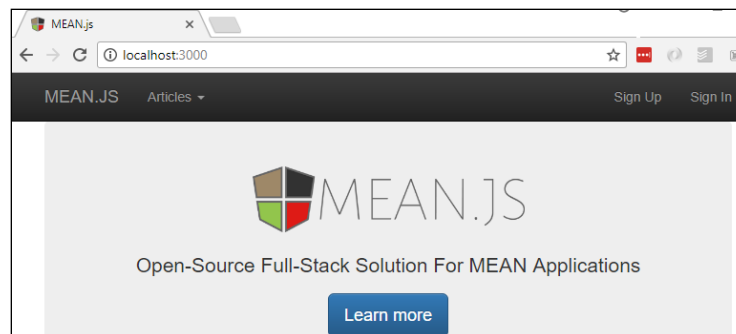


Figura 6-10: aplicativo clichê do Mean.JS

8. Na barra de menus, clique em Inscrever-se e tente criar um novo usuário. Se isso funcionar, o aplicativo tem acesso ao banco de dados local do MongoDB e ele o utiliza para gravar usuários.

Implante o aplicativo do Node.js para o Azure

Agora você tem um aplicativo funcional em execução localmente. Vamos implantá-lo no Azure. Começaremos pela implantação do aplicativo do Node.js no aplicativo Web do Aplicativo Web que criamos anteriormente.

Vamos usar a CLI do Azure para enviar o aplicativo do Git para o Azure. Verifique se você tem a CLI do Azure 2.0 ou posterior instalada no computador. Em seguida, você precisa executar o comando `az login` para fazer login em sua assinatura do Azure. Isso abre um site em que você pode se autenticar e autorizar o uso de sua assinatura do Azure pela CLI.

Para FTP e Git local, você deve ter um usuário de implementação configurado no servidor para autenticar sua implantação. Esse usuário de implantação está no nível de conta e é diferente de sua conta de assinatura do Azure. Você precisa configurar esse usuário de implantação apenas uma vez. Na CLI do Azure, execute o seguinte comando, em que `username` e `password` são valores que você cria e preenche (não inclua o caractere `<` ou `>`):

```
az webapp deployment user set --user-name <username> --password <password>
```

Depois, associamos o repositório Git local ao aplicativo Web executando o seguinte:

```
az webapp deployment source config-local-git --name <app_name> --resource-group <resource_group_name>
```

Isso retornará uma URL na forma de:

```
https://<username>@<app_name>.scm.azurewebsites.net:443/<app_name>.git
```

Agora usaremos essa URL para enviar o aplicativo para o Azure. Primeiro, adicionamos um Azure remoto ao repositório Git local. Execute `cd` para ir até o diretório `meanjs` do código-fonte do aplicativo. Agora execute:

```
git remote add azure <url_for_deployment>
```

Para implantar o aplicativo, execute `git push azure master`. Isso solicita as credenciais que você criou anteriormente. A implantação pode demorar um pouco. Ela vai notificá-lo quando estiver pronta.

Depois de concluir, você pode procurar o aplicativo usando `http://<app_name>.azurewebsites.net`.

Agora, o aplicativo está implantado no aplicativo Web, e temos um banco de dados do Cosmos DB em execução com a API do MongoDB, mas os dois ainda não estão conectados. Vamos fazer isso depois.

Conectar o aplicativo e o banco de dados

Para conectar o aplicativo ao banco de dados, ele precisa saber onde está o banco de dados. Podemos fazer isso definindo a variável de ambiente `MONGODB_URI` no aplicativo Web. Podemos fazer isso usando a CLI do Azure com a execução do seguinte comando:

```
az webapp config appsettings update \
--name <app_name> \
--resource-group myResourceGroup \
--settings
MONGODB_URI="mongodb://<cosmosdb_name>:<primary_master_key>@<cosmosdb_name>.documents.azure.com:10250/mean?ssl=true"
```

Você pode encontrar a cadeia de conexão do Cosmos DB que precisa usar para o valor `MONGODB_URI` no Portal do Azure. Lá, você pode encontrá-la no menu Cadeia de Conexão da conta do Cosmos DB.

Depois de executar o comando com a CLI do Azure, as Configurações de Aplicativo do aplicativo Web serão semelhantes à Figura 6-11.

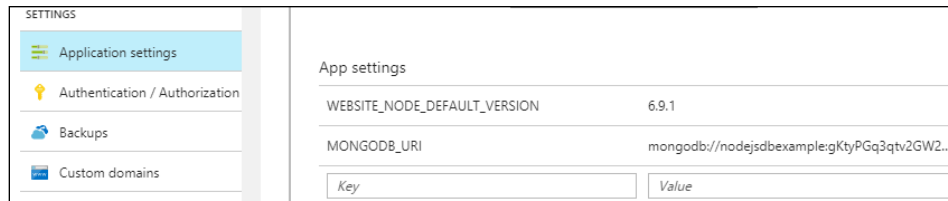


Figura 6-11: Configurações de Aplicativo do Aplicativo Web do Azure

Em vez de executar o comando da CLI do Azure, você também poderia ter adicionado a configuração usando o Portal do Azure.

Agora, navegue até o aplicativo Web novamente usando `http://<app_name>.azurewebsites.net`. Tente inscrever um novo usuário. Quando isso for bem-sucedido, o aplicativo está se comunicando com o banco de dados. Agora você tem um aplicativo do Node.js com um banco de dados do Azure Cosmos usando a API do MongoDB.

Configurando entrega contínua com o GitHub

Até agora, estamos enviando o código de nosso repositório do Git local para o Azure. Isso é adequado se você trabalha sozinho, mas se trabalhar em equipe, você precisará de outro tipo de controle de código-fonte, como o Visual Studio Team Services ou o GitHub.

Nós vamos usar o GitHub para empurrar nosso código e ligar isso ao nosso aplicativo Web, de modo que as alterações sejam implantadas automaticamente em um pipeline de entrega contínua (CD).

1. Vá para <https://github.com/> e crie um novo repositório.

Você precisa entrar ou criar uma nova conta para fazer isso.

2. Crie um novo repositório indo para <https://github.com/new>.
 - a. Digite um nome para o repositório.
 - b. Deixe as outras configurações como estão (repositório público; não crie um README)
 - c. Crie o repositório. Isso resulta em uma tela que deve ser semelhante à Figura 6-12

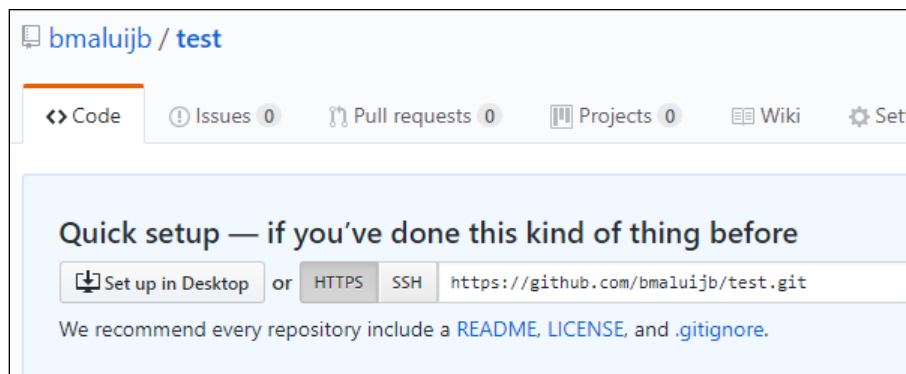


Figura 6-12: Link do repositório no GitHub

3. Use a URL para definir um destino remoto para o repositório do Git local. Você pode fazer isso na janela de comando.

a. Execute `cd` para ir até o diretório meanjs do código-fonte do aplicativo.

b. Execute o seguinte comando:

```
git remote add github https://github.com/bmaluijb/test.git
```

c. Execute o seguinte comando para enviar o código ao GitHub:

```
git push github
```

Agora, nosso código está no GitHub e você pode compartilhá-lo com sua equipe. Em seguida, vamos configurar a opção CD. Usaremos o recurso Opções de Implantação do Aplicativo Web por meio do Portal do Azure. Também poderíamos usar o recurso de Entrega Contínua no Aplicativo Web, mas isso requer uma conta do Visual Studio Team Services, e queremos que tudo seja simples.

1. No Portal do Azure, acesse o aplicativo Web que hospeda o aplicativo do Node.js.
2. Na barra de menus, clique em Opções de implantação.
3. É possível que isso já esteja configurado para o Repositório Git local. Se for esse o caso, você precisará clicar em Desconectar.
4. Em Escolher Origem, selecione GitHub.
5. Na seção Autorização, autorize o Azure a usar o GitHub clicando em Autorizar e concedendo permissão.
6. Na seção Escolher Projeto, escolha o repositório do GitHub que você acabou de criar.
7. Deixe a Ramificação configurada como master
8. Clique em OK e aguarde alguns segundos.

Volte para o menu Opções de Implantação. Agora você pode ver que está conectado. Depois disso, sempre que você enviar uma nova versão do código-fonte para o GitHub, ele será compilado e implantado automaticamente no aplicativo Web, conforme demonstrado na Figura 6-13. Você também pode forçar esse processo clicando no botão Sincronizar.

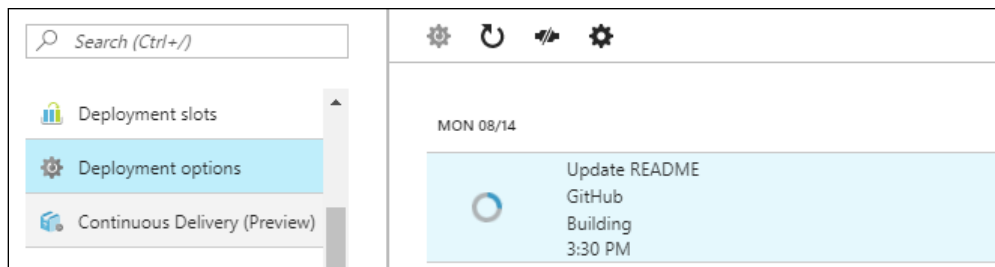


Figura 6-13: A folha Opções de Implantação do Aplicativo Web

Configurar ambientes de preparo

Usando o aplicativo Web, você pode configurar um slot de preparo para testar a nova versão de seu aplicativo. Você pode usar [slots de implantação](#) para isso. Os slots de implantação são Serviços de Aplicativos completos por si mesmos, que você pode usar para testar o código antes de promovê-lo para o próximo slot.

Você pode ter slots de implantação para preparo, teste de carga e produção (que é sempre o Serviço de Aplicativo original, sendo o aplicativo Web do Node.js em nosso caso). De fato, você pode ter tantos intervalos de implantação quantos quiser, sem incorrer em custos adicionais. Os slots de implantação são executados no mesmo Plano de Serviço de Aplicativo, e é isso que você paga.

Você deve se lembrar de que ter slots de implantação adicionais em um Plano de Serviço de Aplicativo consumirá recursos como CPU e memória.

Você pode criar novos slots de implantação por meio do item de menu Slots de Implantação no aplicativo Web. Verifique se está executando o aplicativo Web na camada de preços padrão ou premium, pois o plano gratuito não vem com Slots de Implantação.

Em cada slot de implantação que criou, você pode configurar as opções de implantação como fizemos anteriormente para implantar o código de forma automática. Você pode até trabalhar em diferentes *ramificações de código-fonte* para diferentes ambientes e implantar automaticamente ramificações específicas em slots de implantação específicos.

Além disso, você pode testar sua versão final em um slot de implantação e, quando estiver satisfeito, *trocá-la* com o slot de produção. Isso prepara o aplicativo antes da troca, o que resulta em uma implantação sem tempo de inatividade.

Vamos examinar a criação de um slot de implantação e trocar para ele.

1. No Portal do Azure, acesse o aplicativo Web que hospeda o aplicativo do Node.js.
2. Na barra de menus, clique em Slots de Implantação. A folha de slots de implantação é aberta.
3. Clique no sinal de mais (+) para criar um novo slot de implantação.
 - a. Digite um nome para o slot (por exemplo, encenação).
 - b. Escolha o aplicativo Web do Node.js como Origem da Configuração (isso copia as configurações do aplicativo para o novo slot).
 - c. Clique em OK para criar o slot.
4. Depois que o slot é criado, é semelhante ao aplicativo Web original.
 - a. Configure a opção CD para o slot exatamente como você fez no procedimento anterior para o aplicativo Web.
 - b. Desconecte a conexão de CD no aplicativo Web original do Node.js. Dessa forma, quando você enviar novo código, ele será entregue apenas no slot de preparo
5. Faça uma alteração no aplicativo do Node.js.
 - a. Altere texto no arquivo `home.client.view.html` (você pode encontrá-lo na pasta `meanjs\modules\core\client\views`).
 - b. Confirme-o no Git e envie-o para o GitHub, assim como fez ao implantar o aplicativo do Node.js.

A nova versão está agora no slot de preparo e não no aplicativo Web original (que chamamos de slot de produção). Você pode verificar isso navegando para a URL do aplicativo Web Node.js e para a URL do slot de preparo (que você pode encontrar na folha de visão geral do slot, assim como na folha de visão geral do Aplicativo Web).

Vamos colocar a nova versão em produção.

1. No Portal do Azure, vá para o aplicativo do Web Node.js.
2. Na barra de menus, clique em Slots de Implantação para abrir a folha de slots de implantação.
3. Clique no botão de troca para abrir a folha Troca.
 - a. Deixe todas as configurações como estão.
 - b. Clique em OK para iniciar a troca.

Depois que a troca for concluída, a nova versão do aplicativo do Web Node.js estará em produção (teste-a navegando para a URL do aplicativo do Web Node.js). O uso de slots de implantação dessa maneira é muito benéfico porque você pode testar a nova versão antes de entrar em produção e depois implantá-la na produção sem tempo de inatividade.

Diagnóstico adicional com logs de diagnóstico

Quando o aplicativo está sendo executado, é vital saber como ele está sendo se saindo. Uma ótima maneira de monitorar o aplicativo é usar logs de diagnóstico para ver logs de diagnóstico ativos do aplicativo Web. Você pode até mesmo colocá-los na janela do console. Para fazer isso, execute o seguinte comando na CLI do Azure:

```
az webapp log tail --name <app_name> --resource-group myResourceGroup
```

Você verá alguns logs quando usar o aplicativo no aplicativo Web para gerar tráfego.

Como configurar monitoramento e alertas sofisticados

Outra ótima maneira de acompanhar a performance de seu aplicativo é usar o [Application Insights](#). Essa é uma ferramenta de monitoramento no Azure que pode mantê-lo informado sobre como tudo está sendo feito no aplicativo, desde quantos visitantes usaram o aplicativo até quantas exceções ocorreram e onde elas ocorreram no código. Diferentemente dos logs de diagnóstico, o Application Insights é fornecido mediante uma taxa nominal.

Você pode configurar o Application Insights no Portal do Azure do aplicativo Web.

1. Vá para o Portal do Azure e para o aplicativo Web que hospeda o aplicativo do Node.js.
2. Na barra de menus, clique em Application Insights.
3. Selecione Criar Novo Recurso.
 - a. Digite um nome e selecione um local para a instância do Application Insights.
 - b. Clique em OK. Agora, o Application Insights será implantado e começará a coletar dados para o aplicativo.

Você precisa configurar seu aplicativo para começar a enviar dados ao Application Insights. Para nosso aplicativo de exemplo do Node.js, você precisa fazer o seguinte:

1. No prompt de comando, execute `cd` para acessar a pasta `meanjs` do aplicativo do Node.js.
2. Execute `npm install applicationinsights --save`.
3. Vá para o Portal do Azure e obtenha a chave de instrumentação Application Insights.

Você a encontrará na instância do Application Insights, em Propriedades.

4. Abra o arquivo `server.js` do aplicativo do Web Node.js.
5. Adicione as três seguintes linhas de código JavaScript *antes* da linha:

```
var app = require('./config/lib/app');
const appInsights = require("applicationinsights");
appInsights.setup("<instrumentation_key>");
appInsights.start();
```

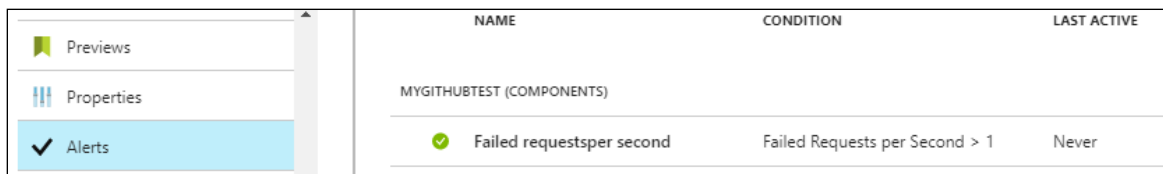
6. Envie as alterações ao GitHub e inicie a implantação automática. No prompt de comando, execute o seguinte:

```
git commit -a -m "added Application Insights support"
```


7. Execute `git push github`, e as alterações serão enviadas para o GitHub e serão implantadas automaticamente no aplicativo Web.

Quando a implantação for concluída, o aplicativo enviará dados para o Application Insights.

Por padrão, o Application Insights executa a *deteção inteligente*. Esse é um recurso muito inteligente que detecta quando há problemas e alerta você quando eles ocorrem. Ele pode detectar itens como um aumento súbito nas solicitações com falha e quando o aplicativo está se comportando de forma incomum. Você também pode criar seus próprios eventos personalizados para todos os tipos de métricas e condições no menu Alertas do Application Insights, conforme demonstrado na Figura 6-14.



NAME	CONDITION	LAST ACTIVE
MYGITHUBTEST (COMPONENTS)		
Failed requests per second	Failed Requests per Second > 1	Never

Figura 6-14: Alertas do Application Insights

Agora, vá para o Portal do Azure, encontre o recurso Application Insights e clique nele. Quando estiver lá, você verá a visão geral, que mostra as métricas básicas, como o tempo de resposta do servidor, o tempo de carregamento de exibição de página e a quantidade de solicitações do servidor e solicitações com falha. Se você vir alguns dados, o Application Insights está funcionando.

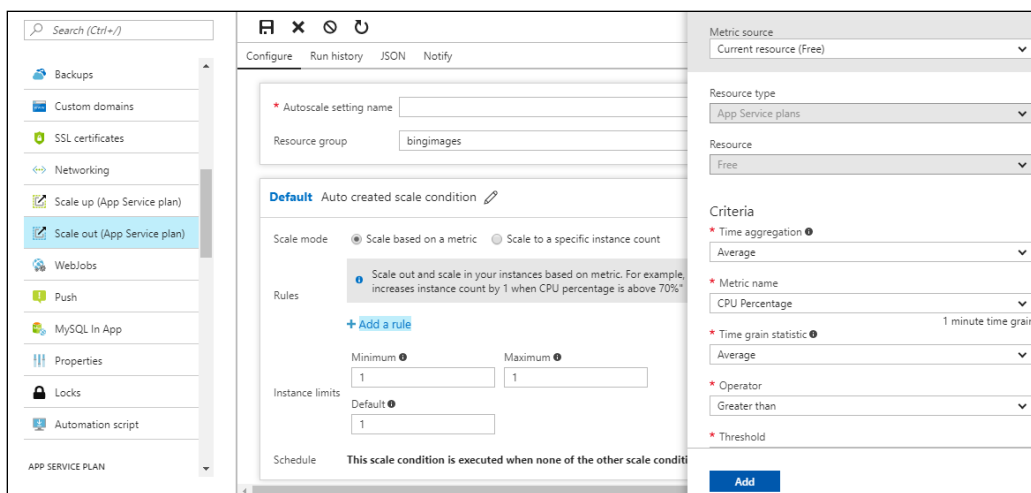


Observação Você pode coletar informações adicionais para enviar ao Application Insights seguindo [este guia](#).

Dimensionar o aplicativo Web

Quando há muitos usuários, o aplicativo Web precisa ser dimensionado para poder acomodar o tráfego. Quando ele não está ocupado, precisamos reduzi-lo para economizar custos. Você pode fazer isso usando o recurso de escala automática do Serviço de Aplicativo. Esteja ciente de que você precisa executar o Aplicativo Web na camada de preços Padrão ou Premium para usar esse recurso.

O aplicativo Web tem um item de menu chamado Expandir, como mostrado na Figura 6-15. Você pode usá-lo para dimensionar de forma manual ou automática.



Search (Ctrl+F)

Backups
Custom domains
SSL certificates
Networking
Scale up (App Service plan)
Scale out (App Service plan)
WebJobs
Push
MySQL In App
Properties
Locks
Automation script

APP SERVICE PLAN

Configure Run history JSON Notify

* Autoscale setting name

Resource group: bingimages

Default Auto created scale condition

Scale mode: ☒ Scale based on a metric ☐ Scale to a specific instance count

Rules: Scale out and scale in your instances based on metric. For example, increases instance count by 1 when CPU percentage is above 70%

+ Add a rule

Minimum: 1 Maximum: 1

Instance limits: Default: 1

Schedule: This scale condition is executed when none of the other scale conditions are active

Metric source: Current resource (Free)

Resource type: App Service plans

Resource: Free

Criteria

* Time aggregation: Average

* Metric name: CPU Percentage 1 minute time grain

* Time grain statistic: Average

* Operator: Greater than

* Threshold

Add

Figura 6-15: Dimensionar um aplicativo Web por meio do Portal do Azure

A redução significa que você adiciona mais instâncias de seu aplicativo para lidar com a carga. Ao reduzir ou ampliar automaticamente, você pode fazer isso com base em métricas, como percentual de uso de CPU ou memória, de acordo com uma agenda (todos os dias às 17h) ou uma combinação de ambos. Isso é muito fácil de configurar e monitorar.

Pronto para produção! Adicionar protocolo SSL

Agora que o aplicativo está pronto para a produção, você deve confirmar que é seguro. Além de autenticação e autorização, fornecer o aplicativo Web por meio de HTTPS é uma das coisas mais importantes que você pode fazer. Isso ocorre porque, sem HTTPS, os intrusos podem ver o tráfego entre seus recursos e usá-lo para realizar atividades mal-intencionadas (como entrar em seu aplicativo). Além disso, o HTTPS é um requisito para recursos de ponta, como os [trabalhos de serviços](#).

Fornecer o tráfego a seu aplicativo Web por meio do protocolo SSL (Secure Sockets Layer) é possível importando um certificado SSL para o aplicativo Web e vinculando-o a um de seus nomes de domínio (personalizados). Você pode importar seu próprio certificado SSL que você ou sua empresa comprou ou comprar um novo por meio dos [Certificados de Serviço de Aplicativo do Azure](#), o que facilita a compra e a validação do certificado. Depois de importar o certificado, você o vincula a uma das associações de nome de domínio do aplicativo Web. Você pode fazer tudo isso no menu Certificados SSL no aplicativo Web.

Passo a passo nº3: criar um back-end para seu aplicativo móvel com o Azure

Quase todos têm um dispositivo móvel hoje em dia. O tráfego para a maioria dos aplicativos provém principalmente de dispositivos móveis, e essa tendência continua a crescer. Criar um aplicativo móvel para sua empresa é uma coisa, mas fornecer uma experiência robusta e confiável é diferente. O Azure pode ajudar com isso, fornecendo um back-end avançado com o [Aplicativo Móvel do Serviço de Aplicativo do Azure](#).

Os aplicativos do Aplicativo Móvel do Azure consistem em duas partes:

- Um back-end que fornece APIs para se comunicar com um banco de dados e realizar tarefas como autenticação, sincronização offline e notificações por push
- SDKs para que todos os tipos de plataformas se conectem facilmente ao back-end

Você pode criar o back-end para o aplicativo do Aplicativo Móvel em Node.js e C#. O SDK do cliente está disponível basicamente para todas as plataformas móveis que estão em uso. Para aplicativos móveis nativos, híbridos e entre plataformas. Ele está disponível para Android, Cordova, iOS, Windows, Xamarin.Android, Xamarin.Forms e Xamarin.iOS.

Neste passo a passo, criamos um novo back-end de Aplicativo Móvel com o C# que usamos com um aplicativo para Android, que compilamos com o Xamarin.Android.

Para acompanhar, você precisa instalar o [Desenvolvimento Móvel com workload .NET](#) usando o instalador do [Visual Studio 2017](#). Isso instala as ferramentas do Xamarin para Android e os SDKs necessários.

Você pode [fazer o mesmo](#) quando estiver trabalhando em um Mac no Visual Studio para Mac.

Criar um aplicativo para dispositivos móveis por meio do Portal do Azure

É fácil começar. Vamos criar o back-end móvel e um aplicativo inicial para o cliente por meio do Portal do Azure.

1. No Portal do Azure, clique em Criar um Novo Serviço e, na caixa Pesquisar, digite **Início Rápido de Aplicativo Móvel**.
2. Selecione o resultado Mobile App Quickstart e clique em Criar.

O Assistente para Criar Aplicativo Móvel é aberto.

- a. Digite um nome para seu aplicativo móvel.
- b. Crie um novo grupo de recursos.
- c. Crie um novo Plano de Serviço de Aplicativo para a execução do Aplicativo Móvel. Você pode usar a Camada de Preços Gratuita para este passo a passo.
- d. Deixe a configuração do Application Insights desativada.
- e. Clique em Criar. Pronto! O aplicativo móvel agora será implantado.

Depois que o Aplicativo Móvel for implantado, você poderá começar a usá-lo. No Portal do Azure, no aplicativo móvel, clique no item do menu Início Rápido e selecione Xamarin.Android. O Assistente de Início Rápido é aberto.

Etapa 1: conectar-se a um banco de dados

Você precisa conectar um banco de dados para armazenar dados. Clique no ícone de informações para começar. Na guia Conexões de Dados que acabou de ser aberta, clique em Criar um Novo Serviço. No Assistente para Adicionar Conexão de Dados que será aberto, faça o seguinte:

1. Para Tipo, selecione Banco de Dados SQL.
2. Crie um novo banco de dados com um novo servidor de destino.
3. Deixe a Cadeia de Conexão como está.
4. Clique em OK.

A conexão de dados é criada.

Etapa 2: criar uma API de tabela

Para se comunicar com o banco de dados, você precisa de uma API. Você pode criar um back-end de API hospedado em seu aplicativo móvel seguindo estas etapas:

1. Na área da Etapa 2, selecione C# e clique em Download. Isso faz o download do projeto de back-end que você vai implantar no aplicativo móvel.
2. Descompacte o arquivo que você acabou de baixar e depois abra-o com o Visual Studio 2017 ou o Visual Studio para Mac.
3. No Visual Studio, clique com o botão direito do mouse no arquivo do projeto e selecione Publicar.
4. Selecione Serviço de Aplicativo, selecione Existente e clique em Publicar.

5. Encontre seu aplicativo móvel e clique em OK para começar a publicar. Pronto! O back-end móvel está funcionando.

Deixe a Etapa 3 para **Criar um novo aplicativo** e clique em **download**.

Descompacte o arquivo que você acabou de baixar e abra-o no Visual Studio 2017 ou no Visual Studio para Mac.

Execute o aplicativo pressionando F5. Isso inicia o Emulador do Android e implanta o aplicativo cliente. O aplicativo cliente é um aplicativo de tarefas pendentes. Você pode adicionar um item na caixa de texto e ele será gravado no back-end de seu aplicativo móvel.

Adicionar autenticação ao aplicativo

Agora que você tem um aplicativo móvel funcional com um back-end, deve se certificar de que ele é seguro. Vamos adicionar autenticação via Azure Active Directory. Para concluir esse processo, primeiro você precisa registrar o aplicativo móvel para autenticação com o Azure Active Directory. [Este tutorial](#) mostra como fazer isso.

Proteger o aplicativo

Nas API de back-end, você precisa definir quais as que deseja proteger.

1. Vá para seu projeto de back-end de aplicativo móvel no Visual Studio 2017 ou no Visual Studio para Mac.
2. Navegue até Controladores > TodoItemController.cs.
3. Adicione o atributo [Authorize] à classe TodoItemController.
Isso restringe qualquer uso não autenticado do controlador.
4. Clique com o botão direito do mouse no arquivo do projeto e depois publique-o em seu aplicativo móvel.

Adicionar autenticação ao aplicativo móvel

O back-end está protegido. Agora, você precisa tornar o aplicativo móvel ciente disso para que ele seja capaz de entrar. Precisamos exigir que os usuários se autenticuem para que possam usar o aplicativo.

1. No Visual Studio, vá para o projeto do Xamarin Mobile App.
2. Adicione o seguinte código à classe TodoActivity:

```
// Define a authenticated user.
private MobileServiceUser user;
private async Task<bool> Authenticate()
{
    var success = false;
    try
    {
        // Sign in with AAD login using a server-managed flow.
        user = await client.LoginAsync(this,
            MobileServiceAuthenticationProvider.WindowsAzureActiveDirectory);
        CreateAndShowDialog(string.Format("you are now logged in - {0}",
            user.UserId), "Logged in!");

        success = true;
    }
    catch (Exception ex)
    {
        CreateAndShowDialog(ex, "Authentication failed");
    }
    return success;
}

[Java.Interop.Export()]
public async void LoginUser(View view)
{
    // Load data only after authentication succeeds.
    if (await Authenticate())
    {
        //Hide the button after authentication succeeds.
        FindViewById<Button>(Resource.Id.buttonLoginUser).Visibility = ViewStates.Gone;

        // Load the data.
        OnRefreshItemsSelected();
    }
}
```

Isso cria um novo método para autenticar um usuário e um manipulador de método para um novo botão Entrar.

3. No método OnCreate, exclua ou comente a seguinte linha de código:
■ OnRefreshItemsSelected();

4. No arquivo `Activity_To_Do.xml`, adicione a seguinte definição de botão `LoginUser` antes do botão `AddItem` existente:

```
<Button
    android:id="@+id/buttonLoginUser"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="LoginUser"
    android:text="@string/login_button_text" />
```

5. Adicione o seguinte elemento ao arquivo de recursos `Strings.xml`:

```
<string name="login_button_text">Entrar</string>
```

6. Execute o aplicativo.

Um botão `Entrar` agora aparece ao lado do botão `Adicionar`. Você não poderá adicionar nenhum item sem entrar, pois todas as ações em `ToDoItemController` agora estão seguras. Ao clicar em `Entrar`, você é direcionado para a tela de autenticação do Azure Active Directory. Aqui, use a conta que você utiliza para entrar no Portal do Azure. Isso é adicionado ao Azure Active Directory por padrão. Entre e você verá que você foi conectado com êxito, conforme descrito na Figura 6-16, e pode começar a adicionar itens de tarefas pendentes.

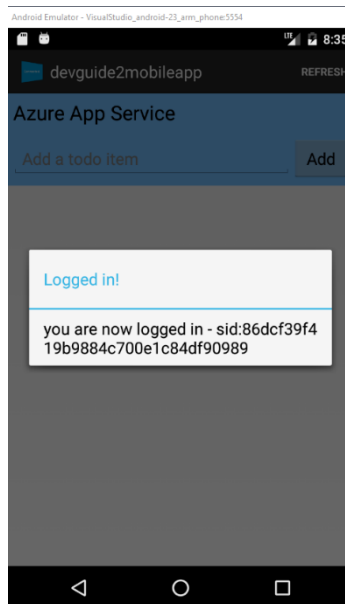


Figura 6-16: Logo após a autenticação no aplicativo móvel no emulador

Adicionar Sincronização Offline ao aplicativo

Em muitos cenários móveis, os usuários nem sempre terão acesso à Internet. Mesmo assim, os usuários devem poder continuar trabalhando. Para acomodar isso, eles podem usar o recurso Sincronização Offline no Aplicativo Móvel do Azure. Vamos ativá-lo para nosso aplicativo:

1. No Visual Studio 2017 ou no Visual Studio para Mac, acesse o aplicativo cliente do Xamarin Android.
2. Na classe `ToDoActivity`, remova o comentário da seguinte linha:

```
#define OFFLINE_SYNC_ENABLED
```
3. Pronto! Execute o aplicativo para implantá-lo no emulador do Android.

Agora, o aplicativo está funcionando exatamente como antes, mas com algumas alterações. Ele usa um objeto `IMobileServiceSyncTable` em vez de `IMobileServiceTable`. Ele também usa o pacote SQLite NuGet. Isso inicia um banco de dados SQLite no dispositivo Android que o aplicativo usa para salvar dados quando não está online. Quando o acesso à Internet é restaurado, as alterações do banco de dados SQLite local são enviadas para seu aplicativo móvel, e as possíveis alterações ocorridas no back-end são sincronizadas com o aplicativo no dispositivo Android. Vamos experimentar isso:

1. O aplicativo está sendo executado no emulador do Android.
2. Entre como fez antes.
3. Adicione um item.
4. Desconecte-se da Internet. No emulador do Android, você pode fazer isso nos Controles Estendidos, definindo a Força do Sinal da Rede como Nenhum.
5. Adicione outro item e conclua um item marcando-o. O aplicativo continua funcionando no banco de dados SQLite local.

Você pode verificar se os itens não estão gravados no banco de dados do back-end conectando-se ao banco de dados do Azure SQL que criamos anteriormente usando uma ferramenta como o [SQL Server Management Studio](#). Você pode encontrar o `connectionstring` do Banco de Dados SQL do Azure ao navegar até ele no Portal do Azure.

6. Conecte o dispositivo Android à internet novamente. No emulador do Android, vá para os controles estendidos e defina a Força do Sinal como Forte.
7. No aplicativo móvel, clique no botão Atualizar. As alterações locais agora serão sincronizadas com o Aplicativo Móvel do Azure.

Verifique o Banco de Dados SQL do Azure para ver as alterações que agora estão sincronizadas com o back-end.

Recursos adicionais e alteração para a produção

É fácil configurar um aplicativo usando o Aplicativo Móvel e incluir recursos avançados, como Autenticação e Sincronização Offline.

Como o aplicativo móvel é um Serviço de Aplicativo do Azure, ele tem todos os mesmos recursos que os outros Serviços de Aplicativos têm. Isso significa que o aplicativo móvel também pode ser dimensionado automaticamente, que você pode ter o mesmo monitoramento e alertas e que pode fazer diagnósticos adicionais com logs de transmissão e até mesmo incorporar depuração de aplicativos.

Quando você move o aplicativo para a produção, pode fornecer segurança adicional usando Certificados de Serviço de Aplicativo. Você também pode configurar a opção CD, assim como fizemos para o aplicativo do Node.js. E você pode usar slots de implantação para implantar sem tempo de inatividade.

Configurar tudo isso costumava ser difícil. Agora, os problemas estão resolvidos. Deixe o Azure fazer o trabalho pesado para você, e se concentre em criar os itens mais úteis!

Usar o Azure Marketplace

O ecossistema Azure não só o ajuda a criar aplicativos dinâmicos mais rápidas e eficazes, mas, com o Azure Marketplace, você pode encontrar soluções de todos os tipos para auxiliar em suas iniciativas de desenvolvimento. E então, ele pode ajudá-lo a conectar seu produto a usuários em todo o mundo.

Como o Azure Marketplace pode me ajudar como desenvolvedor?

O [Azure Marketplace](#) oferece configurações de todos os tipos de produtos, do Windows Server ao Octopus Deploy, aos bancos de dados Oracle, às instâncias do WordPress e aos [servidores Minecraft](#). Basta clicar em uma solução, configurá-la, implantá-la e você pode começar a usá-la. O Marketplace está repleto de produtos e soluções da Microsoft e Parceiros da Microsoft. Há também muitas soluções de software livre.

A vantagem do Marketplace é a facilidade de uso. As soluções são implantadas prontas para execução, juntamente com todos os itens necessários. Às vezes, é uma Máquina Virtual com software instalado, uma Rede Virtual configurada e uma licença integrada instalada ou uma licença que você traz. Às vezes, é um serviço que você adquire, como uma assinatura da API do Bing Mapas para Empresas. Você pode usar o Marketplace para testar facilmente novo software e usar soluções de nível empresarial. Não importa do que você precisa, provavelmente pode encontrar uma solução no Marketplace.

Soluções do Azure

Você pode aprender com as empresas que usam o Azure com êxito, por meio de [Soluções do Azure](#). Elas mostram a documentação do produto, casos de sucesso e práticas recomendadas de arquitetura para casos de uso comuns do Azure, como Marketing Digital, SharePoint no Azure, aplicativos de Linha de Negócios e aplicativos móveis.

O item mais valioso das Soluções do Azure para desenvolvedores são as [arquiteturas de soluções](#). Elas fornecem arquiteturas de referência para problemas comuns e explicam por que você deve usar determinados produtos em certos cenários.

Resumo e o que fazer depois

Neste guia, apresentamos a enorme capacidade que o Azure pode proporcionar a seus aplicativos. Usando o Azure, você pode fazer coisas incríveis com seus aplicativos: reconhecimento facial e de fala, gerenciamento de seus dispositivos na Internet das Coisas na nuvem, escala da maneira desejada e pagamento pelo que você usa.

Você viu que, independentemente da linguagem de programação que você use ou da plataforma para a qual escreve aplicativos, o Azure pode ajudá-lo, com serviços para quase todos os cenários.

Esperamos que você continue usando este e-book para se familiarizar melhor com a vasta gama de serviços do Azure e determinar quais os mais adequados para seu cenário.

Os dias de ter que escrever um estrutura complicado acabaram; agora você pode aproveitar uma grande quantidade de soluções pré-construídas. Libere seu tempo para trabalhar com o que é mais importante, e deixe que o Azure cuide dos problemas resolvidos.

Continue aprendendo com uma conta gratuita do Azure

Inscreva-se para obter uma [conta gratuita do Azure](#) e receba:

- Um crédito de R\$ 665 para usar em qualquer produto do Azure por 30 dias
- Acesso gratuito aos nossos produtos mais populares por 12 meses, inclusive computação, armazenamento, rede e banco de dados
- Mais de 25 produtos que são sempre gratuitos

Sobre os autores

Michael e Barry são entusiastas do Azure e encorajam você a contatá-los no Twitter para dúvidas sobre este livro.



Michael Crump trabalha na Microsoft na plataforma Azure e é codificador, blogueiro e palestrante internacional sobre vários tópicos de desenvolvimento na nuvem. Ele adora ajudar os desenvolvedores a entender os benefícios da nuvem de maneira prática.

Você pode contatá-lo no Twitter em [@mbcrump](https://twitter.com/mbcrump) ou seguindo o seu blog em <https://www.michaelcrump.net>.



Barry Luijbregts é um arquiteto de software independente e um desenvolvedor entusiasta da nuvem. Ele também é um autor da Pluralsight. Ele trabalhou para muitas empresas nos últimos 10 anos e está interessado em compartilhar seus conhecimentos com a comunidade. Ele tem um amplo e profundo conhecimento da pilha da Microsoft, com interesse especial pela tecnologia da Web e a nuvem. Barry é cooperador de um grupo de usuários voltado para a tecnologia e soft-skills chamado .NET Zuid (South) e, atualmente, ensina às pessoas os benefícios da nuvem. Ele mora nos Países Baixos com suas lindas esposa e filha e adora brincar com seus dois huskies siberianos.

Você pode entrar em contato com Barry no Twitter em [@AzureBarry](https://twitter.com/AzureBarry) e por meio de seu site em <https://www.azurebarry.com/>.