Whitepaper da AWS

Arquiteturas multicamada sem servidor da AWS com o Amazon API Gateway e o AWS Lambda

Arquiteturas multicamada sem servidor da AWS com o Amazon API Gateway e o AWS Lambda : Whitepaper da AWS

Table of Contents

Arquiteturas multicamada sem servidor da AWS com o Amazon API Gateway e o AWS Lambda

Data de publicação: 20 de outubro de 2021 (Revisões do documento)

Resumo

Esse whitepaper ilustra como as inovações da Amazon Web Services (AWS) podem ser usadas para mudar a maneira de projetar arquiteturas de vários níveis e implementar padrões populares, como microsserviços, backends móveis e aplicações de página única. Os arquitetos e desenvolvedores podem usar o Amazon API Gateway, o AWS Lambda e outros serviços para reduzir os ciclos de operações e desenvolvimento necessários para criar e gerenciar aplicações de várias camadas.

Introdução

A aplicação multicamadas (três camadas, n camadas e assim por diante) tem sido um padrão de arquitetura fundamental por décadas e continua sendo um padrão popular para aplicações voltadas para o usuário. Embora a linguagem usada para descrever uma arquitetura de várias camadas varie, uma aplicação de várias camadas geralmente consiste nos seguintes componentes:

- Camada de apresentação: componente com o qual o usuário interage diretamente (por exemplo, páginas da Web e interfaces de usuário de aplicativos móveis).
- Camada lógica: código necessário para converter ações do usuário para a funcionalidade da aplicação (por exemplo, operações de banco de dados CRUD e processamento de dados).
- Camada de dados: mídia de armazenamento (por exemplo, bancos de dados, armazenamentos de objetos, caches e sistemas de arquivos) que contêm os dados relevantes para a aplicação.

O padrão de arquitetura multicamadas fornece um framework geral para garantir que os componentes de aplicações desacoplados e escaláveis de forma independente possam ser desenvolvidos, gerenciados e mantidos separadamente (geralmente por equipes distintas).

Como consequência desse padrão em que a rede (uma camada deve fazer uma chamada de rede para interagir com outra camada) atua como o limite entre as camadas, o desenvolvimento de uma aplicação de várias camadas geralmente requer a criação de muitos componentes de aplicação semelhantes. Alguns desses componentes incluem:

- Um código que define uma fila de mensagens para a comunicação entre camadas
- Um código que define uma interface de programa da aplicação (API) e um modelo de dados
- Um código relacionado à segurança que garante o acesso adequado à aplicação

Todos esses exemplos podem ser considerados componentes “padronizados” que, embora necessários em aplicações de várias camadas, não variam muito em sua implementação de uma aplicação para outra.

A AWS oferece vários serviços que permitem a criação de aplicações multicamadas sem servidor, simplificando bastante o processo de implantação dessas aplicações na produção e removendo a sobrecarga associada ao gerenciamento tradicional de servidores. O Amazon API Gateway, um serviço para criar e gerenciar APIs, e o AWS Lambda, um serviço para executar funções de código

arbitrárias, podem ser usados juntos para simplificar a criação de aplicações robustas de várias camadas.

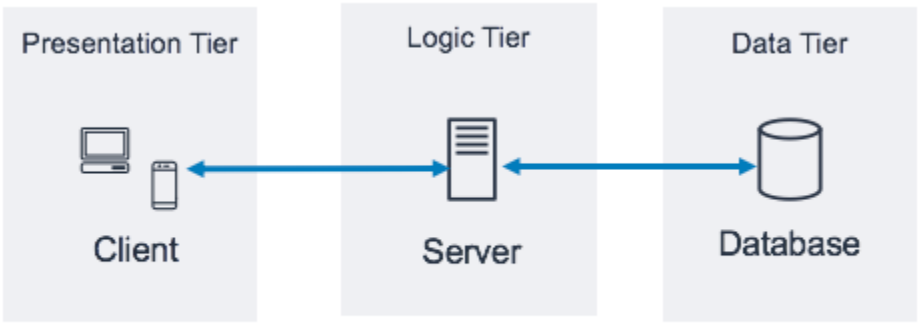
A integração do Amazon API Gateway com o AWS Lambda permite que funções de código definidas pelo usuário sejam iniciadas diretamente por meio de solicitações HTTPS. Independentemente do volume de solicitações, o API Gateway e o Lambda escalam automaticamente para atender às necessidades da aplicação (consulte Cotas do API Gateway do Amazon API Gateway e notas importantes para obter informações de escalabilidade). Ao combinar esses dois serviços, é possível criar uma camada que possibilita a você escrever apenas o código relevante para a aplicação, sem precisar se concentrar em vários outros aspectos indiferenciados da implementação de uma arquitetura de várias camadas, como arquitetura para alta disponibilidade, gravação de SDKs de cliente, gerenciamento do sistema operacional (SO) e do servidor, escalabilidade e implementação de um mecanismo de autorização do cliente.

O API Gateway e o Lambda permitem a criação de uma camada lógica sem servidor. Dependendo dos requisitos da aplicação, a AWS também oferece opções para criar uma camada de apresentação sem servidor (por exemplo, com o Amazon CloudFront e o Amazon Simple Storage Service) e uma camada de dados (por exemplo, com o Amazon Aurora e o Amazon DynamoDB).

Este whitepaper aborda o exemplo mais comum de uma arquitetura de várias camadas, a aplicação Web de três camadas. No entanto, esse padrão de várias camadas pode ser aplicado muito além de uma aplicação Web típica de três camadas.

Visão geral da arquitetura de três camadas

A arquitetura de três camadas é a implementação mais comum de uma arquitetura de várias camadas e consiste em uma única camada de apresentação, camada lógica e camada de dados. A ilustração a seguir mostra um exemplo de uma aplicação simples e genérico de três camadas.



Padrão de arquitetura de uma aplicação de três camadas

Existem muitos recursos online excelentes para aprender mais sobre o padrão geral da arquitetura de três camadas. Este whitepaper aborda um padrão de implementação específico dessa arquitetura usando o Amazon API Gateway e o AWS Lambda.

Camada lógica sem servidor

A camada lógica da arquitetura de três camadas representa o cérebro da aplicação. É aqui que usar o Amazon API Gateway e o AWS Lambda pode ter o maior impacto em comparação com uma implementação tradicional baseada em servidor. Os recursos desses dois serviços permitem que você crie uma aplicação sem servidor altamente disponível, escalável e segura. Em um modelo tradicional, sua aplicação pode exigir milhares de servidores; no entanto, ao usar o Amazon API Gateway e o AWS Lambda, você não é responsável pelo gerenciamento de servidores em qualquer capacidade. Além disso, ao usar esses serviços gerenciados juntos, você obtém os seguintes benefícios:

- AWS Lambda:
  - Não ter que escolher, proteger, corrigir ou gerenciar um sistema operacional
  - Não ter que dimensionar, monitorar ou escalar os servidores na medida
  - Redução do risco ao custo devido ao provisionamento excessivo
  - Redução do risco à performance devido ao subprovisionamento

- Amazon API Gateway:
  - Mecanismos simplificados para implantar, monitorar e proteger APIs
  - Melhor performance da API por meio de armazenamento em cache e entrega de conteúdo

AWS Lambda

AWS Lambda é um serviço computacional que permite executar funções de código arbitrárias em qualquer uma das linguagens compatíveis (Node.js, Python, Ruby, Java, Go, .NET, para obter mais informações, consulte Perguntas frequentes do Lambda) sem provisionar, gerenciar ou escalar servidores. As funções do Lambda são executadas em um contêiner gerenciado e isolado, são iniciadas em resposta a um evento que pode ser um dos vários acionadores programáticos disponibilizados pelo AWS, chamados de fonte de eventos. Consulte todas as fontes de eventos em Perguntas frequentes do Lambda.

Muitos casos de uso comuns do Lambda giram em torno de fluxos de trabalho de processamento de dados orientados por eventos, como o processamento de arquivos armazenados no Amazon S3 ou a transmissão de registros de dados do Amazon Kinesis. Quando usada em conjunto com o Amazon API Gateway, uma função do Lambda executa a funcionalidade de um serviço da Web típico: inicia

o código em resposta a uma solicitação HTTPS do cliente; o API Gateway atua como a porta de entrada da camada lógica e o AWS Lambda chama o código da aplicação.

Sua lógica de negócios está aqui, sem necessidade de servidores

O Lambda exige que você escreva funções de código, chamadas de manipuladores, que serão executadas quando iniciadas por um evento. Para usar o Lambda com o API Gateway, você pode configurar o API Gateway para iniciar funções do manipulador quando ocorrer uma solicitação HTTPS para sua API. Em uma arquitetura multicamada sem servidor, cada uma das APIs criadas no API Gateway se integrará a uma função do Lambda (e ao manipulador interno) que chama a lógica de negócios necessária.

O uso de funções do AWS Lambda para compor a camada lógica permite definir um nível desejado de granularidade para expor a funcionalidade da aplicação (uma função do Lambda por API ou uma função do Lambda por método de API). Dentro da função do Lambda, o manipulador pode alcançar quaisquer outras dependências (por exemplo, outros métodos que você carregou com seu código, bibliotecas, binários nativos e serviços da Web externos) ou até mesmo outras funções do Lambda.

A criação ou atualização de uma função do Lambda requer o carregamento do código como um pacote de implantação do Lambda em um arquivo zip para um bucket do Amazon S3 ou o código de empacotamento como uma imagem de contêiner junto com todas as dependências. As funções podem usar diferentes métodos de implantação, como o Console de Gerenciamento da AWS, executar a AWS Command Line Interface (AWS CLI) ou executar modelos ou frameworks de infraestrutura como código como o AWS CloudFormation, o AWS Serverless Application Model (AWS SAM) ou o AWS Cloud Development Kit (AWS CDK). Ao criar sua função usando qualquer um desses métodos, você especifica qual método dentro do pacote de implantação atuará como o manipulador de solicitações. Você pode reutilizar o mesmo pacote de implantação para várias definições de função do Lambda, em que cada função do Lambda pode ter um manipulador exclusivo dentro do mesmo pacote de implantação.

Segurança do Lambda

Para executar uma função do Lambda, ela deve ser chamada por um evento ou serviço autorizado por uma política do AWS Identity and Access Management (IAM). Com as políticas do IAM, você pode criar uma função do Lambda que não pode ser iniciada, a menos que seja chamada por um recurso do API Gateway definido por você. Essa política pode ser definida com políticas baseadas em recursos em vários serviços da AWS.

Cada função do Lambda assume uma função do IAM atribuída quando a função do Lambda é implantada. Essa função do IAM define os outros recursos e serviços da AWS com os quais sua função do Lambda pode interagir (por exemplo, Amazon DynamoDB, Amazon S3). No contexto da função Lambda, isso é chamado de função de execução.

Não armazene informações confidenciais dentro de uma função do Lambda. O IAM lida com o acesso a serviços da AWS por meio da função de execução do Lambda; se você precisar acessar outras credenciais (por exemplo, credenciais de banco de dados e chaves de API) de dentro de sua função do Lambda, poderá usar o AWS Key Management Service (AWS KMS) com variáveis de ambiente ou usar um serviço como o AWS Secrets Manager para manter essas informações seguras quando não estiverem em uso.

Performance em grande escala

O código extraído como uma imagem de contêiner do Amazon Elastic Container Registry (Amazon ECR) ou de um arquivo zip carregado no Amazon S3 é executado em um ambiente isolado gerenciado pela AWS. Você não precisa escalar suas funções do Lambda. Cada vez que uma notificação de evento for recebida por sua função, o AWS Lambda localiza a capacidade disponível em sua frota de computação e executa seu código com configurações de tempo de execução, memória, disco e tempo limite definidas por você. Com esse padrão, a AWS pode iniciar quantas cópias de sua função forem necessárias.

Uma camada lógica baseada em Lambda é sempre do tamanho certo para as necessidades do cliente. A capacidade de absorver rapidamente picos de tráfego por meio de escalabilidade gerenciada e iniciação simultânea de código, combinada com o preço de pagamento por uso do Lambda, permite que você sempre atenda às solicitações dos clientes e, ao mesmo tempo, não pague pela capacidade computacional ociosa.

Implantação e gerenciamento sem servidor

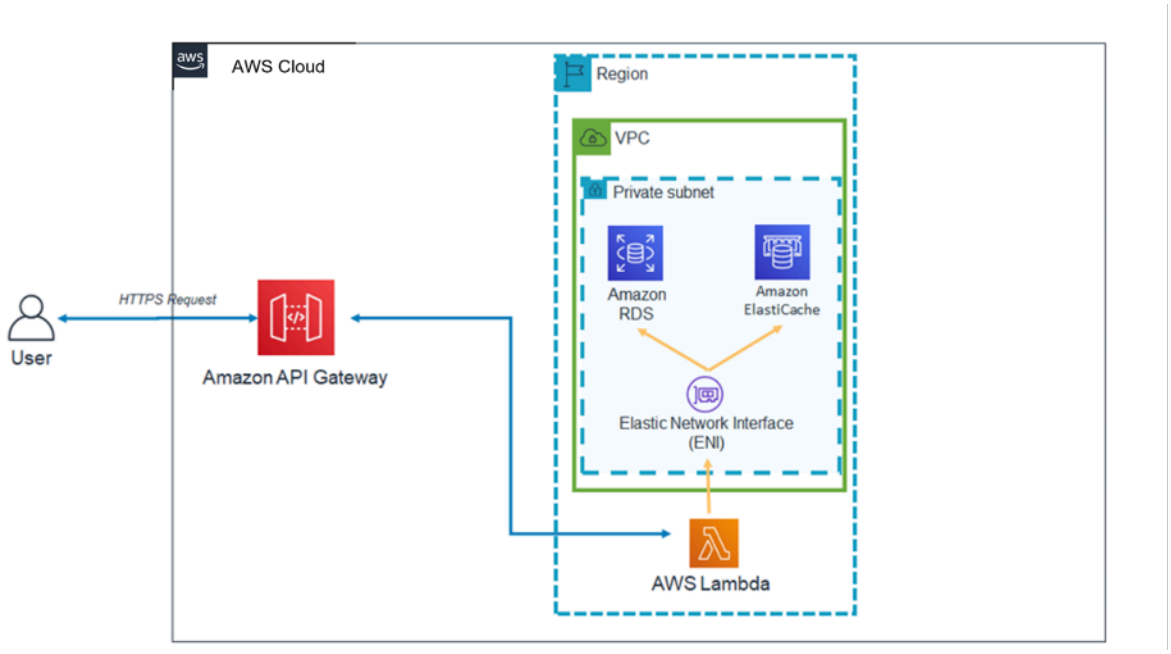
Para ajudá-lo a implantar e gerenciar suas funções do Lambda, use o AWS SAM AWS Serverless Application Model (AWS SAM), um framework de código aberto que inclui:

- Especificação de modelo do AWS SAM: sintaxe usada para definir suas funções e descrever seus ambientes, permissões, configurações e eventos para simplificar o carregamento e a implantação.
- AWS SAM CLI: comandos que permitem verificar a sintaxe do modelo SAM, invocar funções localmente, depurar funções do Lambda e funções do pacote de implantação.

Você também pode usar o AWS CDK, que é um framework de desenvolvimento de software para definir a infraestrutura de nuvem usando linguagens de programação e provisioná-la por meio do CloudFormation. O CDK fornece uma maneira imperativa de definir os recursos da AWS, enquanto o AWS SAM fornece uma forma declarativa.

Normalmente, quando você implanta uma função do Lambda, ela é chamada com permissões definidas por sua função do IAM atribuída e é capaz de alcançar endpoints voltados para a Internet. Como o núcleo de sua camada lógica, o AWS Lambda é o componente que se integra diretamente com a camada de dados. Se a camada de dados contiver informações confidenciais de negócios ou de usuários, é importante garantir que essa camada de dados esteja adequadamente isolada (em uma sub-rede privada).

Você pode configurar uma função do Lambda para se conectar a sub-redes privadas em uma nuvem privada virtual (VPC) em sua conta da AWS se quiser que a função Lambda acesse recursos que você não pode expor publicamente, como uma instância de banco de dados privada. Quando você conecta uma função a uma VPC, Λ cria uma interface de rede elástica para cada sub-rede na configuração da VPC da sua função e a interface de rede elástica é usada para acessar seus recursos internos de forma privada.



Padrão de arquitetura do Lambda dentro de uma VPC

O uso do Lambda com VPC significa que bancos de dados e outras mídias de armazenamento das quais sua lógica de negócios depende podem ficar inacessíveis pela Internet. A VPC também

garante que a única maneira de interagir com seus dados da Internet seja por meio das APIs que você definiu e das funções de código do Lambda que você escreveu.

Amazon API Gateway

O Amazon API Gateway é um serviço totalmente gerenciado que permite que desenvolvedores criem, publiquem, mantenham, monitorem e protejam APIs em qualquer escala.

Os clientes (ou seja, camadas de apresentação) se integram às APIs expostas por meio do API Gateway usando solicitações HTTPS padrão. A aplicabilidade das APIs expostas por meio do API Gateway a uma arquitetura multicamada orientada a serviços é a capacidade de separar partes individuais da funcionalidade da aplicação e expor essa funcionalidade por meio de endpoints REST. O Amazon API Gateway tem recursos e qualidades específicos que podem adicionar recursos poderosos à sua camada lógica.

Integração com o AWS Lambda

O Amazon API Gateway é compatível com os tipos de APIs REST e HTTP. Uma API do API Gateway é composta por recursos e métodos. Um recurso é uma entidade lógica que uma aplicação pode acessar por meio de um caminho de recurso (por exemplo, `/tickets`). Um método corresponde a uma solicitação de API enviada a um recurso da API (por exemplo, `GET /tickets`). O API Gateway permite que você faça backup de cada método com uma função do Lambda, ou seja, quando você chama a API por meio do endpoint HTTPS exposto no API Gateway, o API Gateway chama a função Lambda.

Você pode conectar as funções do API Gateway e do Lambda usando integrações de proxy e integrações que não sejam de proxy.

Integrações de proxy

Em uma integração de proxy, toda a solicitação HTTPS do cliente é enviada como está para a função do Lambda. O API Gateway transmite toda a solicitação do cliente como o parâmetro de evento da função do manipulador do Lambda, e a saída da função do Lambda é retornada diretamente ao cliente (incluindo código de status, cabeçalhos e etc).

Integrações sem proxy

Em uma integração sem proxy, você configura como os parâmetros, os cabeçalhos e o corpo da solicitação do cliente são passados para o parâmetro de evento da função do manipulador do Lambda. Além disso, você configura como a saída do Lambda é convertida de volta para o usuário.

> **Note**
>
> O API Gateway também pode servir como proxy para recursos sem servidor adicionais fora do AWS Lambda, como integrações simuladas (úteis para o desenvolvimento inicial de aplicações) e proxy direto para objetos do S3.

Performance estável da API em todas as regiões

Cada implantação do Amazon API Gateway inclui uma distribuição do Amazon CloudFront por dentro do sistema. O CloudFront é um serviço de entrega de conteúdo que usa a rede global de local de borda da Amazon como pontos de conexão para clientes que usam sua API. Isso ajuda a diminuir a latência de resposta da sua API. Ao usar vários locais de borda em todo o mundo, o Amazon CloudFront também oferece recursos para combater cenários de ataques de negação de serviço distribuída (DDoS). Para obter mais informações, consulte o whitepaper Práticas recomendadas da AWS para resiliência contra DDoS.

Você pode melhorar a performance de solicitações de API específicas usando o API Gateway para armazenar respostas em um cache opcional na memória. Essa abordagem não só oferece benefícios de performance para solicitações repetidas de API, mas também reduz o número de vezes que suas funções do Lambda são chamadas, o que pode reduzir o custo geral.

Incentive a inovação e reduza a sobrecarga com recursos integrados

O custo de desenvolvimento para criar qualquer aplicação nova é um investimento. O uso do API Gateway pode reduzir o tempo necessário para determinadas tarefas de desenvolvimento e reduzir o custo total de desenvolvimento, permitindo que as organizações experimentem e inovem com mais liberdade.

Durante as fases iniciais de desenvolvimento de aplicações, a implementação do registro em log e a coleta de métricas geralmente são negligenciadas para entregar um nova aplicação mais rapidamente. Isso pode levar a dívidas técnicas e riscos operacionais ao implantar esses recursos na execução de uma aplicação em desenvolvimento. O Amazon API Gateway integra-se de forma transparente ao Amazon CloudWatch, que coleta e processa dados brutos do API Gateway em métricas legíveis, quase em tempo real, para monitorar a execução de APIs. O API Gateway também oferece suporte ao registro em log com relatórios configuráveis e rastreamento AWS X-Ray para depuração. Cada um desses recursos não exige que nenhum código seja escrito e pode ser ajustado na execução de uma aplicação em desenvolvimento sem risco para a lógica de negócios principal.

A vida útil geral de uma aplicação pode ser desconhecida ou pode ser conhecida por ser de curta duração. Criar um caso de negócios para a criação de tais aplicações pode ser facilitado se o ponto de partida já incluir os recursos gerenciados fornecidos pelo API Gateway e se você só incorrer em custos de infraestrutura depois que suas APIs começarem a receber solicitações. Para obter mais informações, consulte a Definição de preço do Amazon API Gateway.

Itere rapidamente, mantenha-se ágil

O uso do Amazon API Gateway e do AWS Lambda para criar a camada lógica da sua API permite que você se adapte rapidamente às demandas em constante mudança da sua base de usuários, simplificando a implantação de APIs e o gerenciamento de versões.

Implantação de estágio

Ao implantar uma API no API Gateway, você deve associar a implantação a um estágio do API Gateway. Cada estágio é um snapshot da API e é disponibilizado para as aplicações do cliente chamarem. Usando essa convenção, você pode implantar facilmente aplicações em desenvolvimento, teste, preparação ou em estágios de produção e mover implantações entre os estágios. Cada vez que você implanta sua API em um estágio, você cria uma versão diferente da API que pode ser revertida, se necessário. Esses recursos permitem que a funcionalidade existente e as dependências do cliente continuem sem serem modificadas enquanto novas funcionalidades são lançadas como uma versão separada da API.

Integração desacoplada com o Lambda

A integração entre a API no API Gateway e a função Lambda pode ser desacoplada usando variáveis de estágio do API Gateway e um alias de função do Lambda. Isso simplifica e acelera a implantação da API. Em vez de configurar o nome ou o alias da função do Lambda diretamente na API, você pode configurar a variável de estágio na API, que pode apontar para um alias específico na função do Lambda. Durante a implantação, altere o valor da variável de estágio para apontar para um alias de função do Lambda e a API executará a versão da função do Lambda por trás do alias do Lambda para um estágio específico.

Implantação do lançamento Canary

O lançamento Canary é uma estratégia de desenvolvimento de software em que uma nova versão de uma API é implantada para fins de teste, e a versão base permanece implantada como uma versão de produção para operações normais no mesmo estágio. Em uma implantação de lançamento canary, o tráfego total da API é separado aleatoriamente em uma versão de produção e uma versão

canary com uma proporção pré-configurada. As APIs no API Gateway podem ser configuradas para a implantação do lançamento canary para testar novos recursos com um conjunto limitado de usuários.

Nomes de domínios personalizados

Você pode fornecer um nome de URL intuitivo para negócios para a API, em vez do URL fornecido pelo API Gateway. O API Gateway fornece recursos para configurar o domínio personalizado para as APIs. Com nomes de domínio personalizados, você pode configurar o nome de host da API e escolher um caminho base de vários níveis (por exemplo, `myservice`, `myservice/cat/v1` ou `myservice/dog/v2`) para mapear o URL alternativo para sua API.

Priorize a segurança da API

Todas as aplicações devem garantir que somente clientes autorizados tenham acesso aos seus recursos de API. Ao projetar uma aplicação de várias camadas, você pode aproveitar várias formas diferentes oferecidas pelo Amazon API Gateway para proteger sua camada lógica:

Segurança de trânsito

Todas as solicitações para suas APIs podem ser feitas por meio de HTTPS para habilitar a criptografia em trânsito.

O API Gateway fornece certificados SSL/TLS integrados, se estiver usando a opção de nome de domínio personalizado para APIs voltadas ao público, você poderá fornecer seu próprio certificado SSL/TLS por meio do AWS Certificate Manager. O API Gateway também oferece suporte à autenticação TLS mútua (MTLs). O TLS mútuo aumenta a segurança da sua API e ajuda a proteger seus dados contra ataques como falsificação de clientes ou ataques MITM.

Autorização de API

Cada combinação de recurso/método que você cria como parte da sua API recebe um nome do recurso da Amazon (ARN) exclusivo que pode ser referenciado em políticas AWS Identity and Access Management (IAM).

Existem três métodos gerais para adicionar autorização a uma API no API Gateway:

- Funções e políticas do IAM: os clientes usam a autorização do AWS Signature versão 4 (SIGv4) e as políticas do IAM para acesso à API. As mesmas credenciais podem restringir ou permitir o acesso a outros recursos e serviços da AWS conforme necessário (por exemplo, buckets do Amazon S3 ou tabelas do Amazon DynamoDB).

- Grupos de usuários do Amazon Cognito: os clientes fazem login por meio de um grupo de usuários do Amazon Cognito e obtêm tokens, que estão incluídos no cabeçalho de autorização de uma solicitação.
- Autorizador do Lambda: defina uma função do Lambda que implemente um esquema de autorização personalizado que usa uma estratégia de token de portador (por exemplo, OAuth e SAML) ou usa parâmetros de solicitação para identificar usuários.

Restrições de acesso

O API Gateway oferece suporte à geração de chaves de API e à associação dessas chaves a um plano de uso configurável. Você pode monitorar o uso da chave de API com o CloudWatch.

O API Gateway é compatível com controle de utilização, limites de taxa e de taxa de intermitência para cada método em sua API.

APIs privadas

Com o Amazon API Gateway, é possível criar APIs REST privadas que só podem ser acessadas de sua nuvem privada virtual no Amazon VPC usando um endpoint da VPC da interface. Essa é uma interface de rede de endpoint que você cria em sua VPC.

Com as políticas de recursos, é possível permitir ou negar o acesso à sua API de algumas VPCs e endpoints de VPC, incluindo nas contas da AWS. Cada endpoint pode ser usado para acessar várias APIs privadas. Você também pode usar o AWS Direct Connect para estabelecer uma conexão a partir de uma rede no local para o Amazon VPC e acessar sua API privada nessa conexão.

O tráfego para a API privada sempre usa conexões seguras e não deixa a rede da Amazon; ele é isolado da Internet pública.

Proteção de firewall com o AWS WAF

As APIs voltadas para a Internet são vulneráveis a ataques maliciosos. O AWS WAF é um firewall de aplicação Web que ajuda a proteger as APIs desses ataques. Ele protege as APIs de explorações comuns da Web, como injeção de SQL e ataques de desenvolvimento de scripts multiplataforma. Você pode usar o AWS WAF com o API Gateway para ajudar a proteger APIs.

Camada de dados

Usar AWS Lambda como sua camada lógica não limita as opções de armazenamento de dados disponíveis na camada de dados. As funções do Lambda se conectam a qualquer opção de armazenamento de dados, incluindo o driver de banco de dados apropriado no pacote de implantação do Lambda e usam acesso baseado em função do IAM ou credenciais criptografadas (por meio de AWS KMS ou AWS Secrets Manager).

A escolha de um armazenamento de dados para seu aplicação depende muito dos requisitos da aplicação. A AWS oferece vários armazenamentos de dados com e sem servidor que você pode usar para compor a camada de dados da sua aplicação.

Opções de armazenamento de dados sem servidor

O Amazon S3 é um serviço de armazenamento de objetos que oferece performance, segurança, disponibilidade de dados e escalabilidade líderes do setor.

O Amazon Aurora é um banco de dados relacional compatível com MySQL e PostgreSQL criado para a nuvem e que combina a performance e a disponibilidade de bancos de dados empresariais tradicionais com a simplicidade e a economia de bancos de dados de código aberto. O Aurora oferece modelos de uso tradicional e sem servidor.

O Amazon DynamoDB é um banco de dados de chave-valor e documentos que oferece performance abaixo de 10 milissegundos em qualquer escala. É um banco de dados totalmente gerenciado, sem servidor, multirregião, multimestre e durável com recursos incorporados de segurança, backup e restauração, além de armazenamento em cache na memória para aplicações na escala da Internet.

O Amazon Timestream é um serviço totalmente gerenciado de banco de dados de séries temporais rápido e escalável para aplicações operacionais e de IoT. Esse serviço simplifica o armazenamento e a análise de trilhões de eventos por dia a um décimo do custo dos bancos de dados relacionais. Impulsionados pelo aumento de dispositivos IoT, sistemas de TI e máquinas industriais inteligentes, os dados de séries temporais, dados que medem como as situações mudam ao longo do tempo, são um dos tipos de dados que mais crescem.

O Amazon Quantum Ledger Database (Amazon QLDB) é um banco de dados ledger totalmente gerenciado que fornece um log de transação transparente, imutável e verificável criptograficamente pertencente a uma autoridade central confiável. O Amazon QLDB monitora todas as alterações de

dados da aplicação e mantém um histórico completo e verificável dessas alterações no decorrer do tempo.

O Amazon Keyspaces (for Apache Cassandra) é um serviço de banco de dados escalável, altamente disponível e gerenciado que é compatível com o Apache Cassandra. Com o Amazon Keyspaces, é possível executar suas workloads do Cassandra na AWS usando o mesmo código de aplicação e as mesmas ferramentas do desenvolvedor do Cassandra que você usa hoje. Não é necessário provisionar, aplicar patches ou gerenciar servidores nem instalar, manter ou operar software. Como o Amazon Keyspaces é um produto sem servidor, você paga apenas pelos recursos que usa, e o serviço aumenta e reduz automaticamente a escala das tabelas na vertical de acordo com o tráfego da aplicação.

O Amazon Elastic File System (Amazon EFS) fornece um sistema de arquivos elástico simples e sem servidor para definição única que permite compartilhar dados de arquivos sem provisionar nem gerenciar o armazenamento. Ele pode ser usado com serviços da Nuvem AWS e recursos on-premises e foi criado para escalar sob demanda até petabytes sem interromper as aplicações. Com o Amazon EFS, você pode expandir e reduzir seus sistemas de arquivos automaticamente à medida que adiciona e remove arquivos, dispensando a necessidade de provisionar e gerenciar capacidade para se adaptar ao crescimento. O Amazon EFS pode ser montado com a função Lambda, o que o torna uma opção viável de armazenamento de arquivos para APIs.

Opções de armazenamento de dados com servidor

O Amazon Relational Database Service (Amazon RDS) é um serviço da Web gerenciado que facilita a configuração, a operação e a escalabilidade de um banco de dados relacional usando qualquer um dos mecanismos disponíveis (Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle e Microsoft SQL Server) e executado em diversos tipos de instância de banco de dados que são otimizados para memória, performance ou E/S.

O Amazon Redshift é um serviço de data warehouse na nuvem, totalmente gerenciado, na escala dos petabytes.

O Amazon ElastiCache é uma implantação totalmente gerenciada do Redis ou Memcached. Implante, opere e escale de forma transparente armazenamentos de dados na memória comuns e compatíveis de código aberto.

O Amazon Neptune é um serviço de banco de dados de grafos rápido, confiável e totalmente gerenciado que facilita a criação e a execução de aplicações que trabalham com conjuntos de dados

altamente conectados. O Neptune oferece suporte aos modelos de gráficos comuns, gráficos de propriedades e W3C Resource Description Framework (RDF), e suas respectivas linguagens de consulta, permitindo que você crie facilmente consultas que navegam com eficiência em conjuntos de dados altamente conectados.

O Amazon DocumentDB (compatível com MongoDB) é um serviço totalmente gerenciado de banco de dados de documentos rápido, escalável e altamente disponível que oferece suporte a workloads do MongoDB.

Por fim, você também pode usar armazenamentos de dados executados de forma independente no Amazon EC2 como a camada de dados de uma aplicação de multicamadas

Camada de apresentação

A camada de apresentação é responsável por interagir com a camada lógica por meio dos endpoint REST do API Gateway expostos pela Internet. Qualquer dispositivo ou cliente compatível com HTTPS pode se comunicar com esses endpoints, dando à sua camada de apresentação a flexibilidade de assumir várias formas (aplicações de desktop, aplicativos móveis, páginas da Web, dispositivos IoT e etc). Dependendo dos requisitos, sua camada de apresentação pode usar as seguintes ofertas sem servidor da AWS: qualquer dispositivo ou cliente compatível com HTTPS pode se comunicar com esses endpoints, dando à camada de apresentação a flexibilidade de assumir várias formas (aplicações de desktop, aplicativos móveis, páginas da Web, dispositivos de IoT e etc). Dependendo dos seus requisitos, a camada de apresentação pode usar as seguintes ofertas sem servidor da AWS:

- Amazon Cognito: um serviço de sincronização de dados e identidade de usuário sem servidor que permite adicionar cadastro, login e controle de acesso de usuários aos seus aplicativos móveis e da Web de forma rápida e eficiente. O Amazon Cognito escala para milhões de usuários e oferece suporte ao login com provedores de identidade social, como Facebook, Google e Amazon, e provedores de identidade empresariais via SAML 2.0.
- Amazon S3 com CloudFront: permite a hospedagem de sites estáticos, como aplicações de página única, diretamente de um bucket do S3 sem exigir o fornecimento de um servidor Web. Você pode usar o CloudFront como uma rede de entrega de conteúdo (CDN) gerenciada para melhorar a performance e habilitar o SSL/TL usando certificados gerenciados ou personalizados.

O AWS Amplify é um conjunto de ferramentas e serviços que podem ser usados em conjunto ou individualmente para ajudar desenvolvedores de front-end de plataformas móveis e da Web a criar aplicações escaláveis e de pilha completa, desenvolvidas pela AWS. O Amplify oferece um serviço totalmente gerenciado para implantação e hospedagem de aplicações Web estáticas globalmente, oferecido por meio da rede confiável de entrega de conteúdo da Amazon com centenas de pontos de presença em todo o mundo e com fluxos de trabalho de CI/CD incorporados que aceleram o ciclo de lançamento da aplicação. O Amplify é compatível com frameworks comuns da Web, incluindo JavaScript, React, Angular, Vue, Next.js, e plataformas móveis, incluindo Android, iOS, React Native, Ionic e Flutter. Dependendo das configurações de rede e dos requisitos da aplicação, talvez seja necessário habilitar as APIs do API Gateway para serem compatíveis com o compartilhamento de recursos de origem cruzada (CORS). A conformidade com CORS permite que os navegadores da Web chamem diretamente suas APIs de dentro de páginas Web estáticas.

Ao implantar um site com o CloudFront, você recebe um nome de domínio do CloudFront para acessar a aplicação (por exemplo, `d2d47p2vcczkh2.cloudfront.net`). O Amazon Route 53 pode ser usado para registrar nomes de domínio e direcioná-los para sua distribuição do CloudFront ou direcionar nomes de domínio já pertencentes à sua distribuição do CloudFront. Isso permite que os usuários acessem seu site usando um nome de domínio conhecido. Observe que você também pode atribuir um nome de domínio personalizado à sua distribuição do API Gateway usando o Route 53, o que permite que os usuários chamem APIs usando nomes de domínio conhecidos.

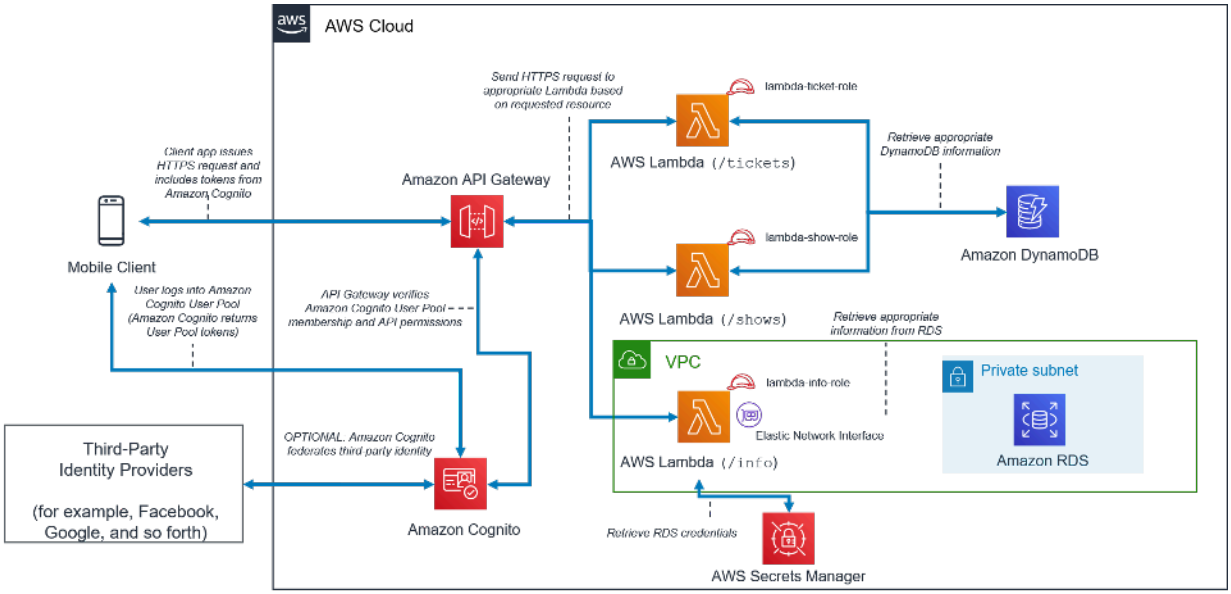
Exemplos de padrões de arquitetura

É possível implementar padrões de arquitetura comuns usando o API Gateway e o AWS Lambda como sua camada lógica. Este whitepaper inclui os padrões de arquitetura mais conhecidos que utilizam as camadas lógicas baseadas em AWS Lambda:

- Backend móvel: um aplicativo móvel se comunica com o API Gateway e o Lambda para acessar os dados do aplicativo. Esse padrão pode contemplar clientes HTTPS genéricos que não usam recursos da AWS sem servidor para hospedar recursos da camada de apresentação (como clientes de desktop, servidor Web em execução no EC2 e etc).
- Aplicação de página única: uma aplicação de página única hospedada no Amazon S3 e no CloudFront se comunica com o API Gateway e o AWS Lambda para acessar os dados da aplicação.
- Aplicação Web: a aplicação Web é um back-end de uso geral e orientado a eventos que usa o AWS Lambda com o API Gateway para lógica de negócios. Ela também usa o DynamoDB como banco de dados e o Amazon Cognito para gerenciamento de usuário. Todo o conteúdo estático é hospedado usando o Amplify.

Além desses dois padrões, este whitepaper discute a aplicabilidade do Lambda e do API Gateway a uma arquitetura geral de microsserviços. Uma arquitetura de microsserviço é um padrão comum que, embora não seja uma arquitetura padrão de três camadas, envolve o desacoplamento de componentes de aplicações e sua implantação como unidades individuais de funcionalidade sem estado que se comunicam entre si.

Backend móvel



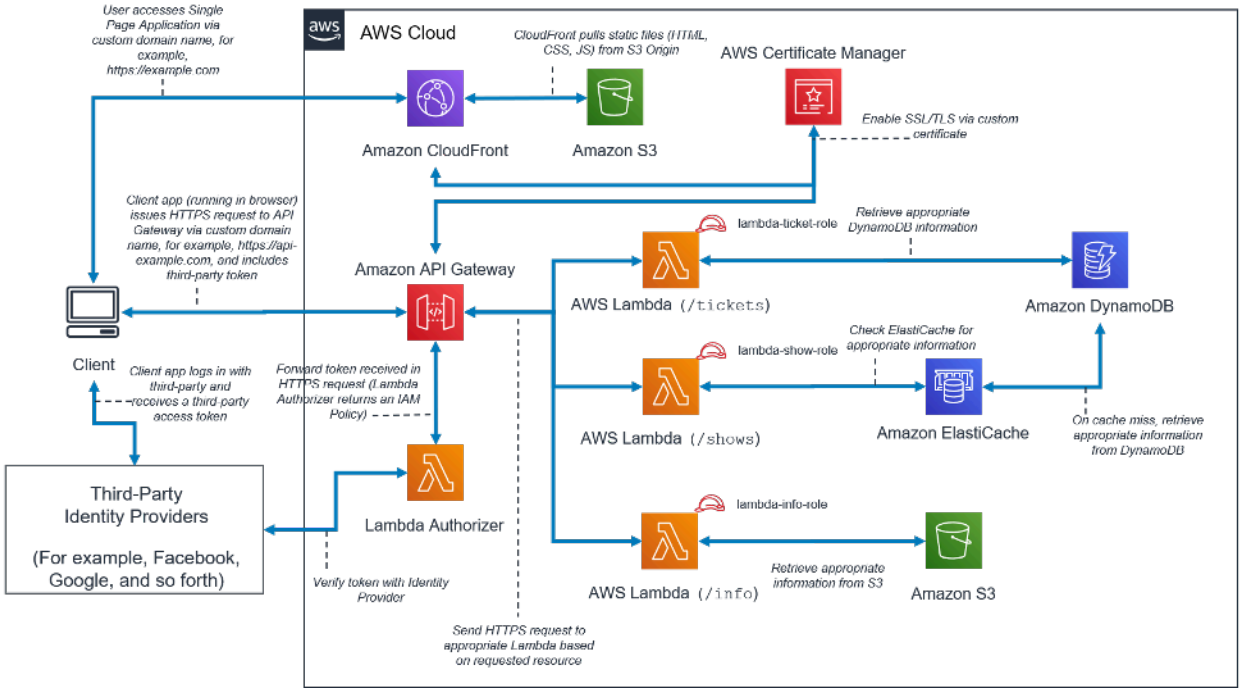
Padrão de arquitetura de backend móvel sem servidor

Tabela 1: Componentes da camada de backend móvel

Camada	Componentes
Apresentação	Aplicativo móvel em execução em um dispositivo de usuário.
Lógica	Amazon API Gateway com AWS Lambda. Essa arquitetura mostra três serviços expostos (`/tickets`, `/shows` e `/info`). Os endpoints do API Gateway são protegidos por grupos de usuários do Amazon Cognito. Nesse método, os usuários fazem login nos grupos de usuários do Amazon Cognito (por meio de um terceiro federado, se necessário) e recebem tokens de ID e acesso usados para autorizar chamadas do API Gateway. Cada função do Lambda recebe sua própria função do Identity and Access Management

Camada	Componentes
	(IAM) para fornecer acesso à origem de dados apropriada.
Dados	O DynamoDB é usado para os serviços `/tickets` e `/shows`. O Amazon RDS é usado para o serviço `/info`. Essa função do Lambda recupera credencia is do Amazon RDS do AWS Secrets Manager e usa uma interface de rede elástica para acessar a sub-rede privada.

Aplicação de página única.

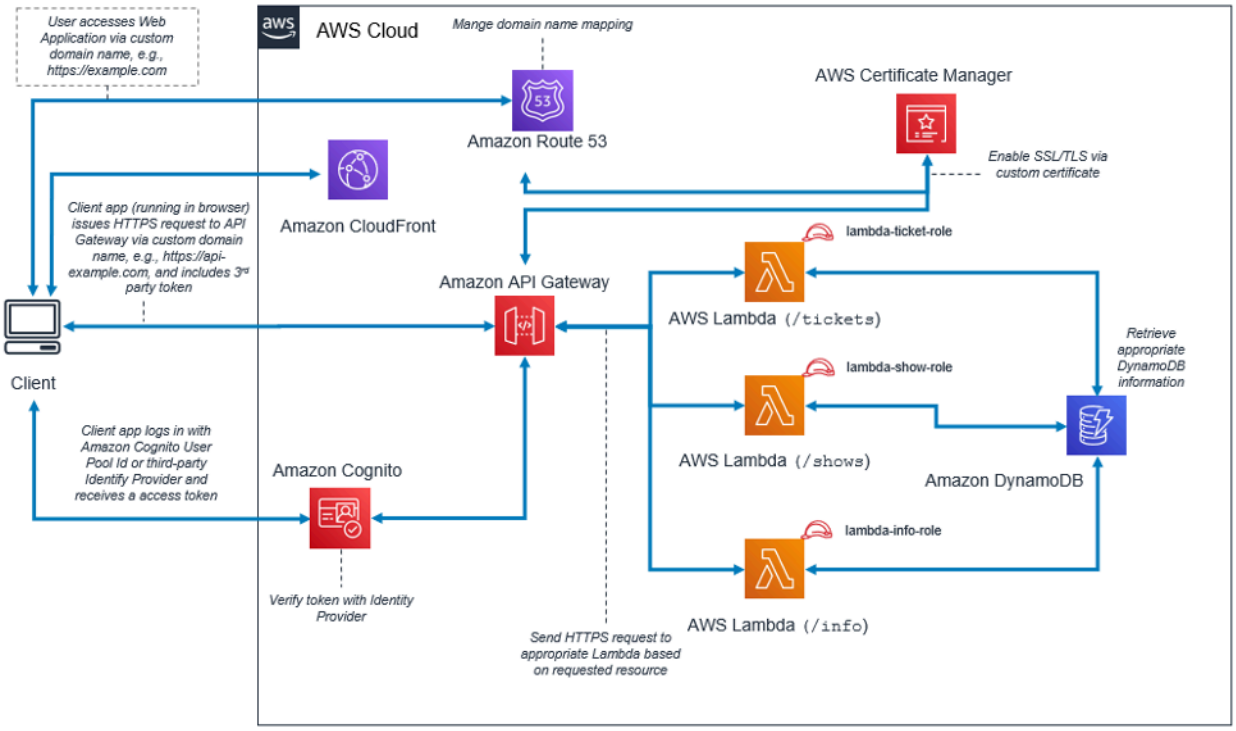


Padrão de arquitetura de aplicações de página única sem servidor

Tabela 2: Componentes da aplicação de página única

Camada	Componentes
Apresentação	Conteúdo estático do site hospedado no Amazon S3, distribuído pelo CloudFront. O AWS Certificate Manager permite a utilizaçã o de um certificado SSL/TLS personalizado.
Lógica	API Gateway com AWS Lambda. Essa arquitetura mostra três serviços expostos (`/tickets`, `/shows` e `/info`). Os endpoints do API Gateway são protegidos por um autorizador do Lambda. Nesse método, os usuários fazem login por meio de um provedor de identidade de terceiros e obtêm tokens de ID e acesso. Esses tokens são incluídos nas chamadas do API Gateway, o autorizad or do Lambda valida esses tokens e gera uma política do IAM contendo permissões de iniciação de API. Cada função do Lambda recebe sua própria função do IAM para fornecer acesso à origem de dados apropriada.
Dados	O Amazon DynamoDB é usado para os serviços `/tickets` e `/shows`. O Amazon ElastiCache é usado pelo serviço `/shows` para melhorar a performance do banco de dados. As falhas de cache são enviadas ao DynamoDB. O Amazon S3 é usado para hospedar conteúdo estático usado pelo `/info service`.

Aplicação Web



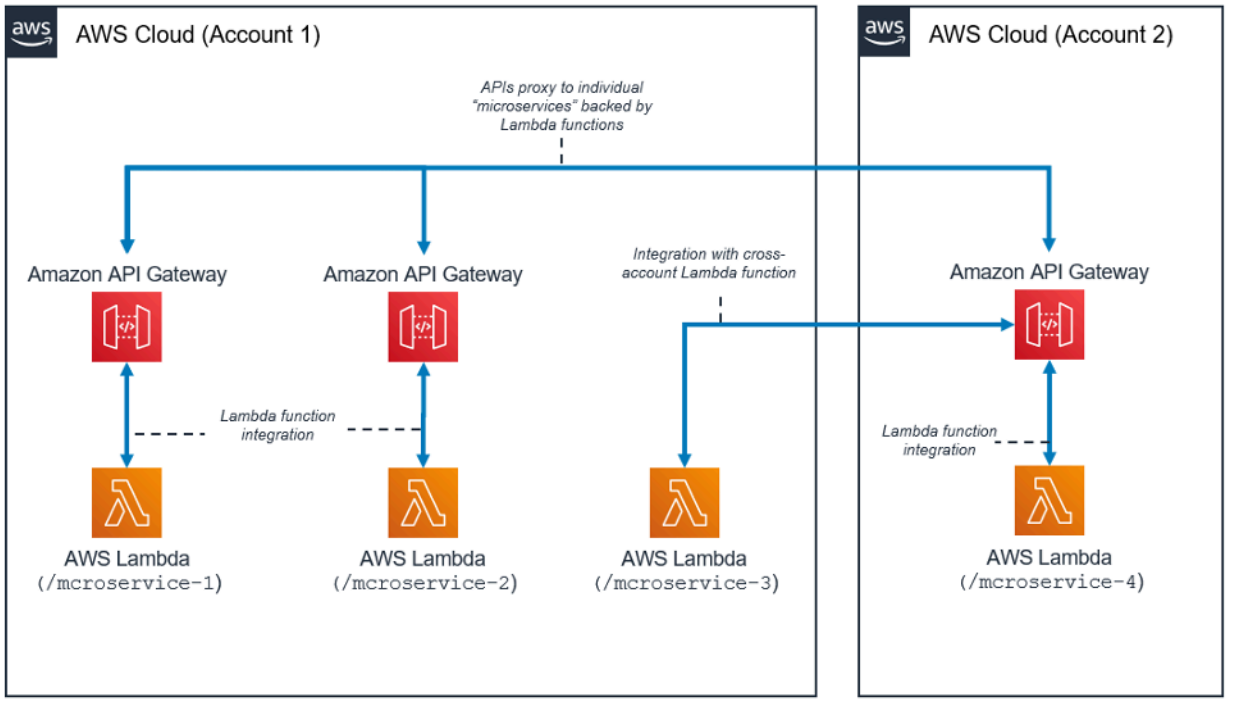
Padrão de arquitetura de aplicação Web

Tabela 3: Componentes de aplicação Web

Camada	Componentes
Apresentação	A aplicação front-end é o conteúdo estático (HTML, CSS, JavaScript e imagens) gerado por utilitários React como create-react-app. O Amazon CloudFront hospeda todos esses objetos. A aplicação Web, quando usada, baixa todos os recursos para o navegador e começa a ser executada. A aplicação Web se conecta ao backend chamando as APIs.
Lógica	A camada lógica é criada usando funções do Lambda lideradas por APIs REST do API Gateway.

Camada	Componentes
	Essa arquitetura mostra vários serviços expostos. Existem várias funções diferentes do Lambda. Cada uma lida com um aspecto diferente da aplicação. As funções do Lambda estão por trás do API Gateway e podem ser acessadas por caminhos de URL da API. A autenticação do usuário é processada usando grupos de usuários do Amazon Cognito ou provedores de usuários federados. O API Gateway usa integração imediata com o Amazon Cognito. Somente depois que um usuário for autenticado, o cliente receberá um token JSON Web Token (JWT) que deve ser usado ao fazer as chamadas de API. Cada função do Lambda recebe sua própria função do IAM para fornecer acesso à origem de dados apropriada.
Dados	Neste exemplo específico, o DynamoDB é usado para o armazenamento de dados, mas outros serviços de armazenamento ou banco de dados da Amazon de uso específico podem ser usados dependendo do cenário e caso de uso.

Microsserviços com Lambda



Padrão de arquitetura de microsserviços com Lambda

O padrão de arquitetura de microsserviços não está vinculado à arquitetura típica de três camadas; no entanto, esse padrão comum pode obter benefícios significativos com o uso de recursos sem servidor.

Nessa arquitetura, cada um dos componentes da aplicação é desacoplado, implantado e operado de forma independente. Uma API criada com o Amazon API Gateway, e funções posteriormente lançadas pelo AWS Lambda, é tudo o que você precisa para criar um microsserviço. Sua equipe pode usar esses serviços para desacoplar e fragmentar seu ambiente até o nível de granularidade desejado.

Em geral, um ambiente de microsserviços pode apresentar as seguintes dificuldades: sobrecarga repetida para a criação de cada novo microsserviço, problemas com a otimização da densidade e utilização do servidor, complexidade de executar várias versões de vários microsserviços simultaneamente e proliferação de código dos requisitos do lado do cliente para integração com diversos serviços distintos.

Quando você cria microsserviços usando recursos sem servidor, esses problemas se tornam menos difíceis de resolver e, em alguns casos, simplesmente desaparecem. O padrão de microsserviços sem servidor reduz a barreira da criação de cada microsserviço subsequente (o API Gateway

permite até mesmo a clonagem de APIs existentes e o uso de funções do Lambda em outras contas). A otimização da utilização do servidor não é mais relevante com esse padrão. Por fim, o Amazon API Gateway fornece SDKs de cliente gerados programaticamente em várias linguagens comuns para reduzir a sobrecarga de integração.

Conclusão

O padrão de arquitetura multicamadas incentiva a prática recomendada de criar componentes de aplicações que são simples de manter, desacoplar e escalar. Quando você cria uma camada lógica em que a integração ocorre pelo API Gateway e a computação ocorre pelo AWS Lambda, você atinge essas metas enquanto reduz a quantidade de esforço para alcançá-las. Juntos, esses serviços fornecem um front-end de API HTTPS para seus clientes e um ambiente seguro para aplicar sua lógica de negócios e, ao mesmo tempo, remover a sobrecarga envolvida no gerenciamento de uma infraestrutura típica baseada em servidor.

Colaboradores

Os colaboradores desse documento incluem:

- Andrew Baird, arquiteto de soluções da AWS
- Bryant Bost, consultor do AWS ProServe
- Stefano Buliani, gerente sênior de produtos, tecnologia, AWS Mobile
- Vyom Nagrani, gerente sênior de produtos, AWS Mobile
- Ajay Nair, gerente sênior de produtos, AWS Mobile
- Rahul Popat, arquiteto de soluções globais
- Brajendra Singh, arquiteto sênior de soluções

Leitura adicional

Para obter informações adicionais, consulte:

- Whitepapers e guias da AWS

Revisões do documento

Para ser notificado sobre atualizações deste whitepaper, inscreva-se no RSS feed.

update-history-change	update-history-description	update-history-date
Whitepaper atualizado	Atualizado quanto à novos recursos e padrões de serviço.	20 de outubro de 2021
Whitepaper atualizado	Atualizado quanto à novos recursos e padrões de serviço.	1.º de junho de 2021
Whitepaper atualizado	Atualizado quanto à novos recursos de serviço.	25 de setembro de 2019
Publicação inicial	Whitepaper publicado.	1.º de novembro de 2015

Avisos

Os clientes são responsáveis por fazer sua própria avaliação independente das informações neste documento. Este documento é: (a) fornecido apenas para fins informativos, (b) representa as ofertas e práticas de produtos atuais da AWS, que estão sujeitas a alterações sem aviso prévio e (c) não cria nenhum compromisso ou garantia da AWS e suas afiliadas, fornecedores ou licenciadores. Os produtos ou serviços da AWS são fornecidos no “estado em que se encontram”, sem garantias, declarações ou condições de qualquer tipo, explícitas ou implícitas. As responsabilidades e obrigações da AWS com seus clientes são regidas por contratos da AWS, e este documento não modifica nem faz parte de nenhum contrato entre a AWS e seus clientes.