Introdução ao Desenvolvimento Serverless na AWS

Cassiano Peres
Analista e desenvolvedor de sistemas

Mais sobre mim

- Graduado em TADS (UTFPR)
- Pós-graduando em Defesa Cibernética (UNICIV)
- CTO - Arabyka e Brexbit
- Apaixonado pela liberdade e descentralização
- GitHub: cassianobrexbit
- LinkedIn: peres-cassiano

Objetivo do curso

Neste curso vamos explorar o desenvolvimento de aplicações sem servidores utilizando a infraestrutura de nuvem da AWS, conhecer suas características e aplicações.

Percurso

Aula 1 O que é serverless?

Aula 2 Backend serverless com AWS Lambda

Aula 3 NoSQL na Amazon com DynamoDB

Percurso

Aula 4 Trabalhando com arquivos no Amazon S3

Aula 5 Criando API REST com Amazon API Gateway

Aula 6 Construindo nossa aplicação serverless

Percurso

Aula 7 Revisão

Aula 1

O que é serverless?

// Introdução ao Desenvolvimento Serverless na AWS

Objetivos

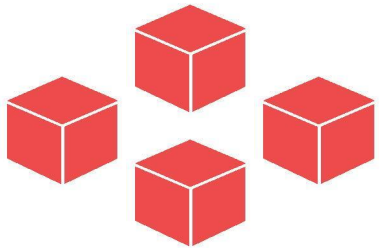
- Entender o que é desenvolvimento serverless
- Explorar as diferenças entre aplicações serverless e com servidor
- Explorar os recursos serverless da AWS

O que é serverless?

Serverless computing, ou apenas serverless (sem servidor) é um modelo de execução onde o provedor de nuvem é responsável pela provisão dos recursos para a execução do código, os alocando de forma dinâmica e cobrando apenas pelo seu uso.

O que é serverless?

O código é executado em **containers sem estado** acionados por eventos (requisições HTTP, eventos de banco de dados, serviços de filas, alertas, uploads de arquivos, etc).



O que é serverless?

O código é enviado para a nuvem na forma de **funções** - **Function-as-a-Service**, com os seu componentes principais sendo:

- **Eventos**: desencadeia a execução da função é considerado como um evento (upload de arquivo)
- **Funções**: são uma unidade independente de implantação (processamento de arquivos)
- **Recursos**: componentes usados pela função são definidos como recursos (banco de dados, repositório de arquivos).

Conceitos importantes

Microservices: a aplicação deve ser arquitetada na forma de funções desacopladas.
Stateless functions: os containers são destruídos após a execução da função, não armazenando seu estado.
Cold Starts: latência para responder a um evento quando ativado sob demanda.

Aula 1 . Etapa 2

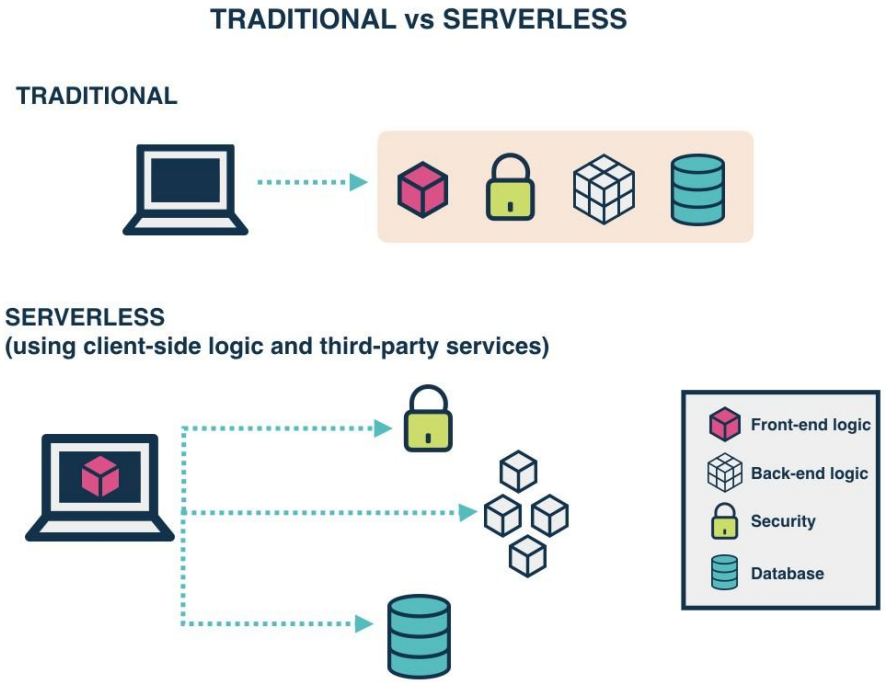
Serverless X Server Computing

// Introdução ao desenvolvimento serverless na AWS

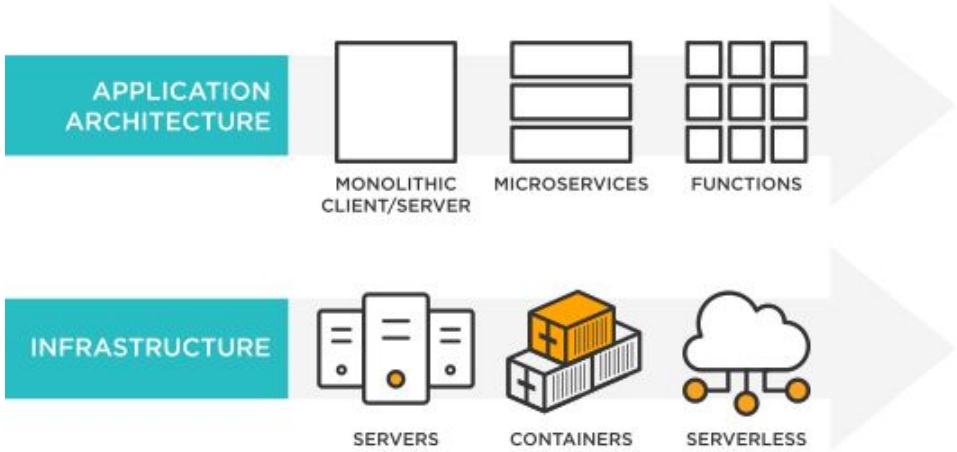
Diferenças entre server e serverless

A computação serverless permite que os desenvolvedores adquiram serviços de back-end em uma base flexível de "pagamento conforme o uso", pagando apenas pelos serviços utilizados.

Diferenças entre server e serverless



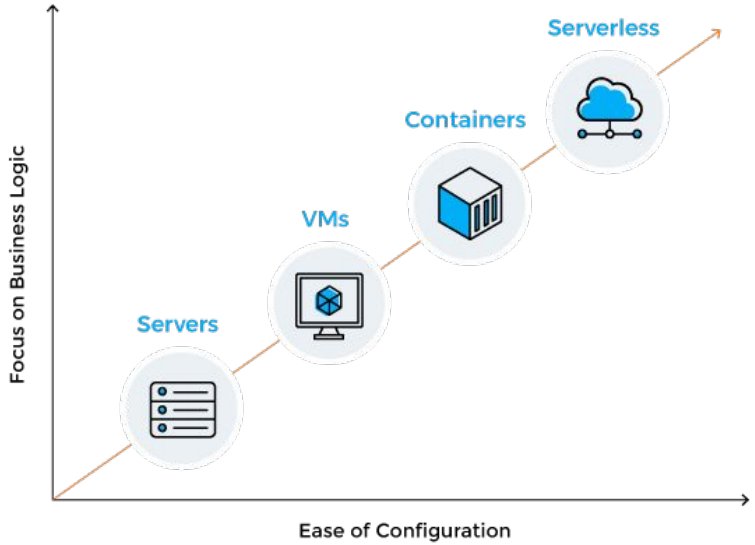
Diferenças entre server e serverless



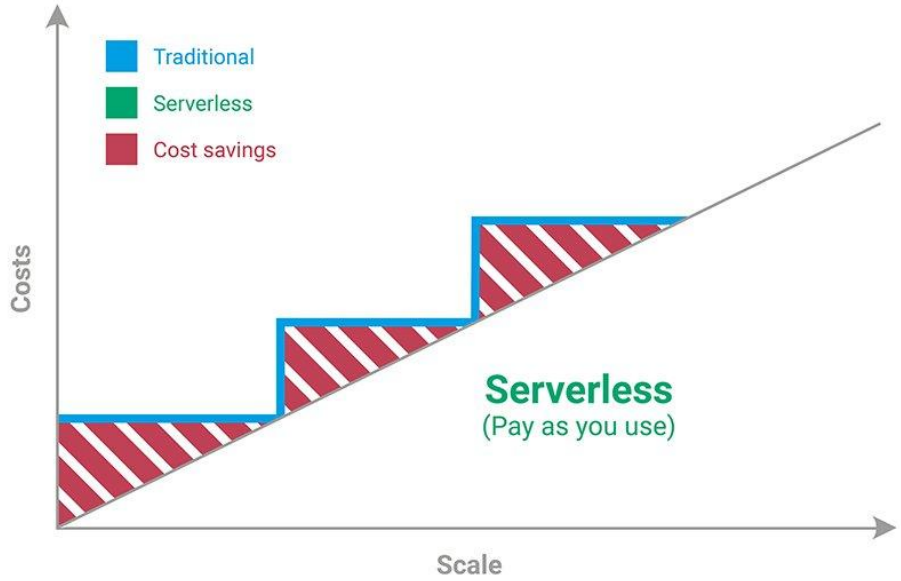
Benefícios da computação serverless

- Fácil configuração
- Foco na lógica
- Custos gerenciáveis
- Segurança
- Sem custos com tempo *idle* nos servidores
- Auto-escalável, evitando desperdícios e gargalos

Benefícios da computação serverless



Benefícios da computação serverless



Observações sobre computação serverless

- Dependência das regras do provedor de nuvem
- Tarefas longas são muito caras
- Latência (Cold starts)
- Aumento da complexidade

Aula 1 . Etapa 3

Recursos serverless na AWS

// Introdução ao desenvolvimento serverless na AWS

Serverless na AWS

A AWS oferece tecnologias para executar código, gerenciar dados e integrar aplicações, sem a necessidade de gerenciar servidores, com escalabilidade automática, alta disponibilidade integrada e faturamento pago **por utilização** agilizando e otimizando custos.

Serviços serverless na AWS

- **Computação**

 AWS Lambda

 AWS Fargate

- **Integração de aplicações**

 Amazon EventBridge

 AWS Step Functions

 Amazon SQS

 Amazon SNS

 Amazon API Gateway

 AWS AppSync

Serviços serverless na AWS

- **Armazenamento de dados**

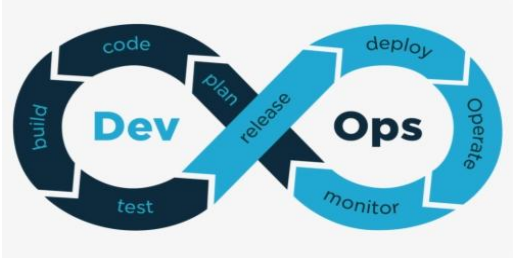
 Amazon S3

 Amazon DynamoDB

 Amazon Aurora Serverless

Serverless na AWS

Possui ferramentas contendo estruturas de trabalho, CI/CD, monitoramento, registros e diagnósticos, autoria e desenvolvimento, facilmente integrável com ferramentas e recursos focados em gerenciamento de código e DevOps.



Aula 2

Backend serverless com AWS Lambda

// Introdução ao desenvolvimento serverless na AWS

Objetivos

- Entender o que é o AWS Lambda
- Criar nossa primeira função Lambda

O que é AWS Lambda?

Serviço gerenciado pela AWS que permite a execução de código sem a necessidade da provisão e gerenciamento de servidores.

O que é AWS Lambda?

- Possui suporte a diversas linguagens de programação (Python, Java, C#, Go, Ruby)
- Métricas e logs rastreáveis pelo AWS Cloudwatch
- É pago apenas pelo tempo de processamento
- Escalabilidade contínua.

Como funciona o AWS Lambda?

Integração com serviços AWS

- REST APIs 
- Processamento de dados 
- Mensageria 
- Orquestração 
- Detecção de alterações 

Outros recursos do AWS Lambda

- Lambda Edge: deploy em múltiplas regiões AZ
- Destinations: integração com outros serviços AWS
- Layers: importar bibliotecas externas
- Simultaneidade provisionada: menor latência e alta disponibilidade.

Custos do AWS Lambda

- É cobrado a cada 100 ms de execução de código e pelo número de vezes que o código é acionado.
- A cobrança começa a partir da execução até retornar ou encerrar.
- O preço depende da quantidade de memória alocada.
- Nível gratuito: 400.000 segundos de execução com 1GB de memória alocado por mês.

Aula 2 . Etapa 2

Criando a primeira função Lambda

// Introdução ao desenvolvimento serverless na AWS

Criando uma função Lambda pelo AWS Console

Nesta atividade prática vamos:

- Acessar o console do AWS Lambda
- Criar uma função Lambda
- Testar a função criada
- Verificar logs e resultados

Aula 2 . Etapa 3

Adicionando layers a uma função Lambda

// Introdução ao desenvolvimento serverless na AWS

Layers em uma função Lambda

Nesta atividade prática vamos:

- Criar um layer para uma função lambda
- Importar para o nosso código
- Testar a função criada
- Verificar logs e resultados

Aula 3

NoSQL na Amazon com DynamoDB

// Introdução ao desenvolvimento serverless na AWS

Objetivos

- O que é NoSQL?
- Conhecendo o Amazon DynamoDB
- Criando nossa primeira tabela no Amazon DynamoDB

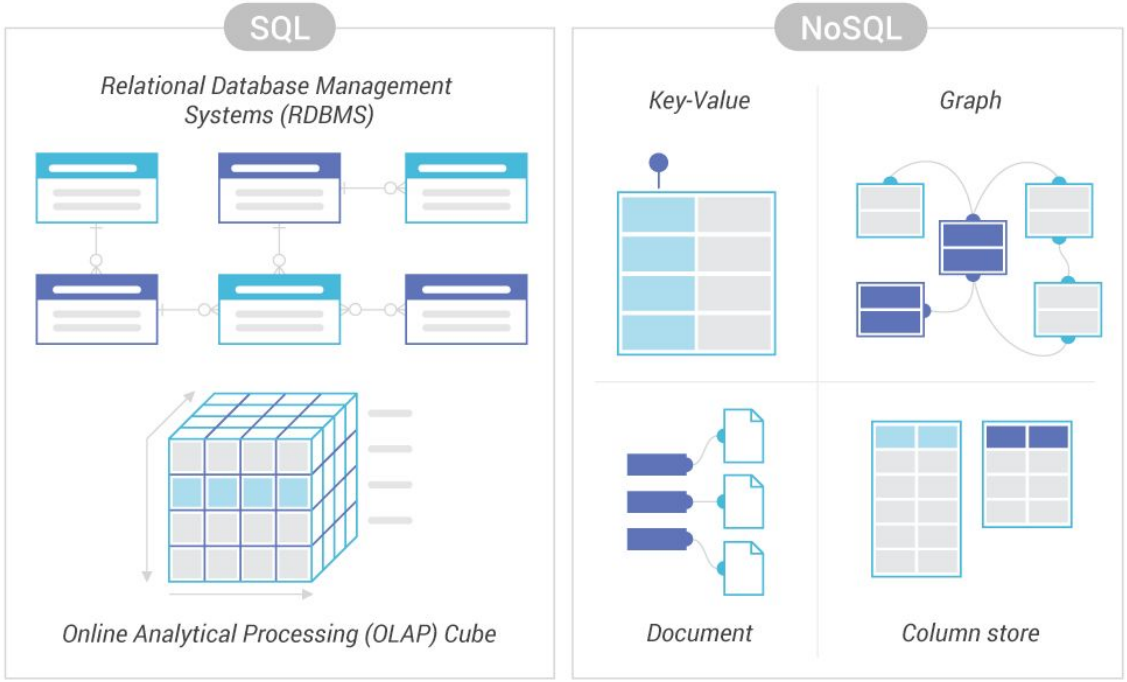
O que é NoSQL?

Bancos de dados NoSQL **(Not only SQL)** são criados para **modelos de dados específicos** e têm esquemas flexíveis para a criação de aplicativos modernos. Os bancos de dados NoSQL possuem facilidade de desenvolvimento, funcionalidade e performance em escala e podem ser estruturados ou não.

O que é NoSQL?

- Em um banco de dados relacional os registros são normalizados em tabelas separadas, com relacionamentos definidos por chaves primárias e estrangeiras
- Em um banco de dados não relacional utilizando normalmente esquemas de pares chave/valor, grafos, família de colunas e documentos.

NoSQL vs SQL databases



Por que utilizar NoSQL?

- **Flexibilidade**: os bancos de dados NoSQL fornecem esquemas flexíveis que permitem um desenvolvimento mais rápido e iterativo.
- **Escalabilidade**: escalados horizontalmente com clusters distribuídos de hardware, em vez de escalá-los verticalmente adicionando servidores caros e robustos.

Por que utilizar NoSQL?

- **Alta performance**: otimizado para modelos de dados específicos e padrões de acesso com maior performance do que bancos de dados relacionais**.**
- **Altamente funcional**: os bancos de dados NoSQL fornecem APIs e tipos de dados altamente funcionais criados especificamente para cada um de seus modelos de dados.

Observações sobre NoSQL

- **Falta de padrões:** tecnologia e comunidade novas, atualizações constantes e documentação não tão completa.
- **Falta de ferramentas de relatórios**: não possui a quantidade de ferramentas de relatórios, análises de desempenho, testes como os RDBs.

Aula 3 . Etapa 2

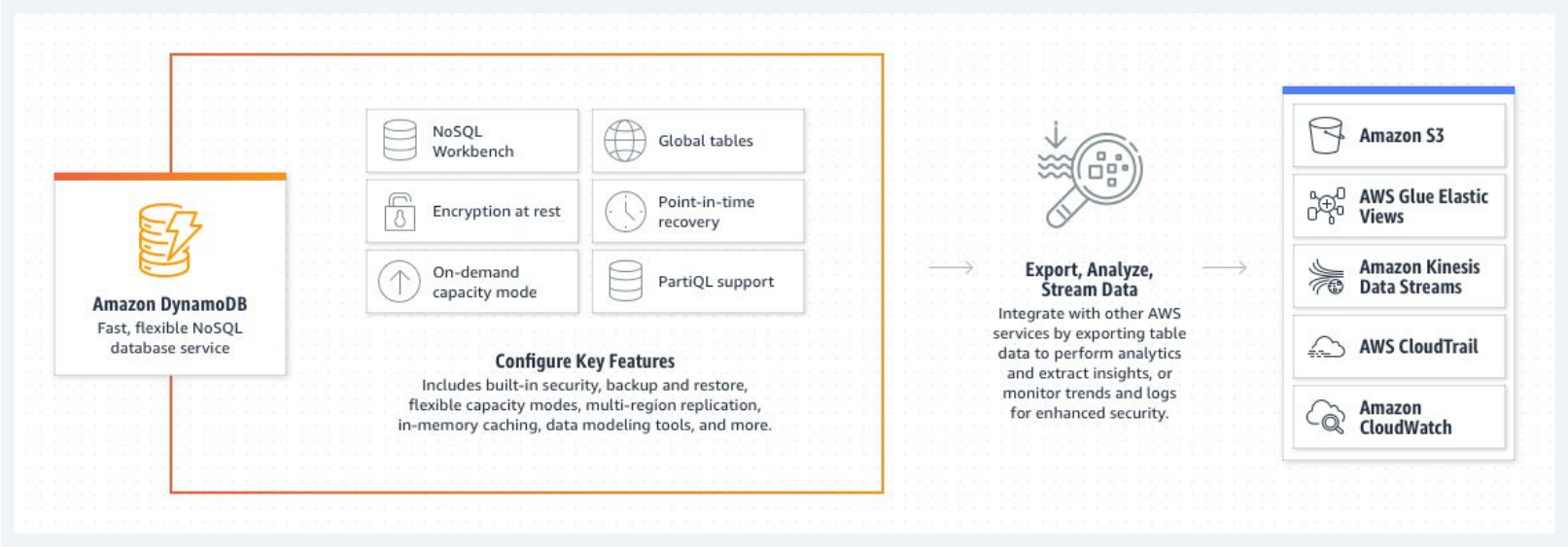
O que é o Amazon DynamoDB?

// Introdução ao desenvolvimento serverless na AWS

Conhecendo o Amazon DynamoDB

- Banco de dados de chave-valor NoSQL
- Sem servidor e totalmente gerenciado
- Executar aplicações de alta performance em qualquer escala
- Replicação multiregional automatizada
- Armazenamento em cache na memória e ferramentas de exportação de dados.

Conhecendo o Amazon DynamoDB



Arquitetura do Amazon DynamoDB

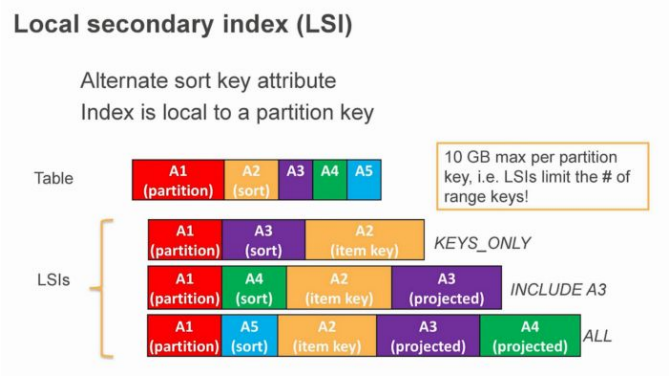
- Possui dois modos de capacidade para cada tabela: sob **demanda e provisionada**.
- Para tabelas no modo sob demanda, se adapta de forma instantânea às cargas de trabalho.
- Auto escalável com dimensionamento automático da taxa de transferência.
- Rastreamento de alterações com triggers

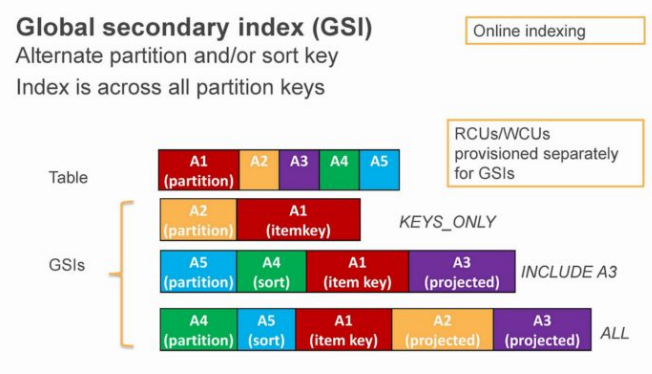
Estrutura do Amazon DynamoDB

- **Tabelas**: único atributo obrigatório é a **Partition Key**, semelhante a chave primária de um banco relacional, com a opção de utilização de uma **Sort Key**, que permite o retorno maior e mais rápido dos dados ordenados e queries.

Estrutura do Amazon DynamoDB

- **Indexes**:
  - Local Secondary Index (LSIs)
  - Global Secondary Index (GSI)





Aula 3 . Etapa 3

Criando a primeira tabela no DynamoDB

// Introdução ao desenvolvimento serverless na AWS

Criando uma tabela no DynamoDB pelo AWS Console

Nesta atividade prática vamos:

- Acessar o console do DynamoDB
- Criar uma tabela
- Inserir itens
- Consultar itens

Aula 4

Trabalhando com arquivos no Amazon S3

// Introdução ao desenvolvimento serverless na AWS

Objetivos

- Entender o Amazon S3
- Criar o primeiro bucket no S3
- Realizar upload e acesso de arquivos em um bucket no S3

O que é o Amazon S3?

O Amazon Simple Storage Service (Amazon S3) é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade de dados, segurança e performance.

O que é o Amazon S3?

Recursos do Amazon S3

- Criação de buckets
- Armazenamento de dados e arquivos
- Download de dados e arquivos
- Criação de permissões
- Interface simples e padrão

Classes de armazenamento no Amazon S3

- S3 Standard
- S3 Intelligent-Tiering
- S3 Standard-Infrequent Access (S3 Standard-IA)
- S3 One Zone-Infrequent Access (S3 One Zone-IA)
- Amazon S3 Glacier (S3 Glacier)
- Amazon S3 Glacier Deep Archive (S3 Glacier Deep Archive)
- S3 Outposts.

Preços Amazon S3

5 GB de armazenamento padrão do Amazon S3 na classe de armazenamento S3 Standard, 20.000 solicitações GET, 2.000 solicitações PUT, COPY, POST ou LIST e 15 GB de transferência de dados nos primeiros 12 meses.

Aula 4 . Etapa 2

Criando um bucket no Amazon S3

// Introdução ao desenvolvimento serverless na AWS

Criar um bucket e gerenciar arquivos no Amazon S3

Nesta atividade prática vamos:

- Criar um bucket
- Realizar o upload de arquivos
- Realizar tarefas de gerenciamento de arquivos em um bucket do S3

Aula 5

Criando API REST com o Amazon API Gateway

// Introdução ao desenvolvimento serverless na AWS

Objetivos

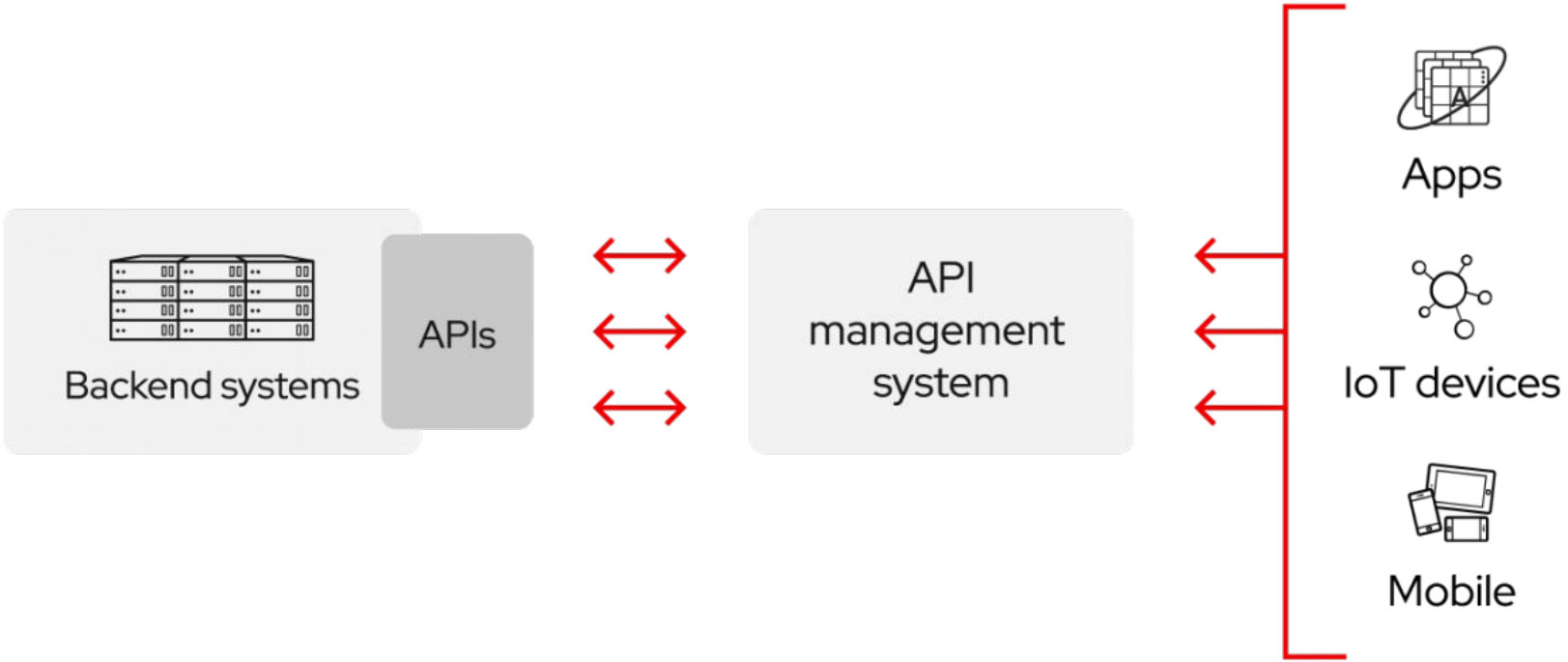
- Entender o que são APIs
- Conhecer o Amazon API Gateway
- Criar a nossa primeira API REST

O que é uma API?

Uma API permite a interação entre serviços sem precisar saber como eles foram implementados, simplificando o desenvolvimento e a integração entre aplicações.

Podem ser vistas como **contratos**, com a documentação que representa o acordo entre as partes interessadas.

O que é uma API?



Tipos de APIs

- **API privada**: é utilizada internamente dentro da organização.
- **API de parceiros**: é compartilhada com parceiros de negócios específicos.
- **Api pública**: é disponibilizada para terceiros, que podem desenvolver aplicações que interagem enviando ou recebendo dados.

Aula 5 . Etapa 2

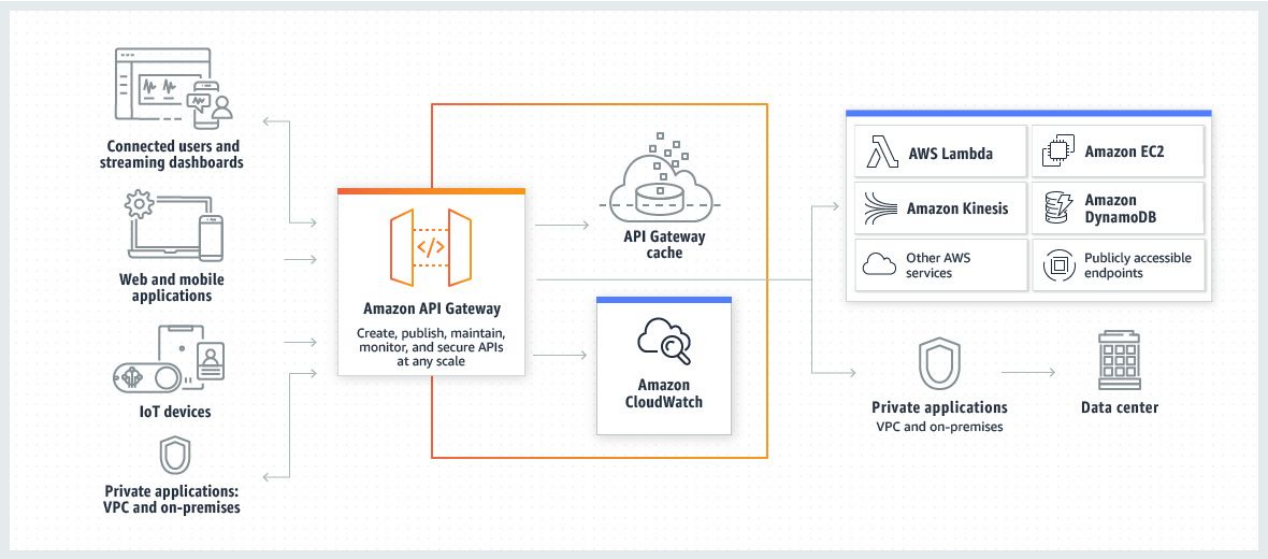
Conhecendo o Amazon API Gateway

// Introdução ao desenvolvimento serverless na AWS

O que é o Amazon API Gateway?

Serviço para criação, publicação, manutenção, monitoramento e proteção de APIs REST e WebSocket em qualquer escala, permitindo que as APIs acessem outros recursos e dados da AWS

O que é o Amazon API Gateway?



Benefícios do Amazon API Gateway

- **Métricas**: permite definir planos para restrições de acesso, definições de cotas, chaves de API e mensurar o tráfego.
- **Segurança**: permite o controle de acesso por meio de autorizadores
- **Resiliência**: suporta picos de acesso com gerenciamento de tráfego.
- **Monitoramento**: gerenciamento de logs

Amazon API Gateway: Tipos

- **HTTP**: otimizadas para criar APIs que são proxy para funções do AWS Lambda ou back-ends HTTP, e ideais para cargas de trabalho sem servidor.
- **REST**: oferecem recursos de gerenciamento de API, como planos de uso, chaves de API e APIs de publicação e monetização.
- **WebSockets**: mantêm uma conexão persistente entre clientes conectados para permitir a comunicação de mensagens em tempo real

Aula 5 . Etapa 3

Criando uma API REST

// Introdução ao desenvolvimento serverless na AWS

Criando uma API REST

Nesta atividade prática vamos:

- Criar uma API Rest no console
- Configurar métodos
- Conectar com o backend Lambda
- Testar a execução

Aula 6

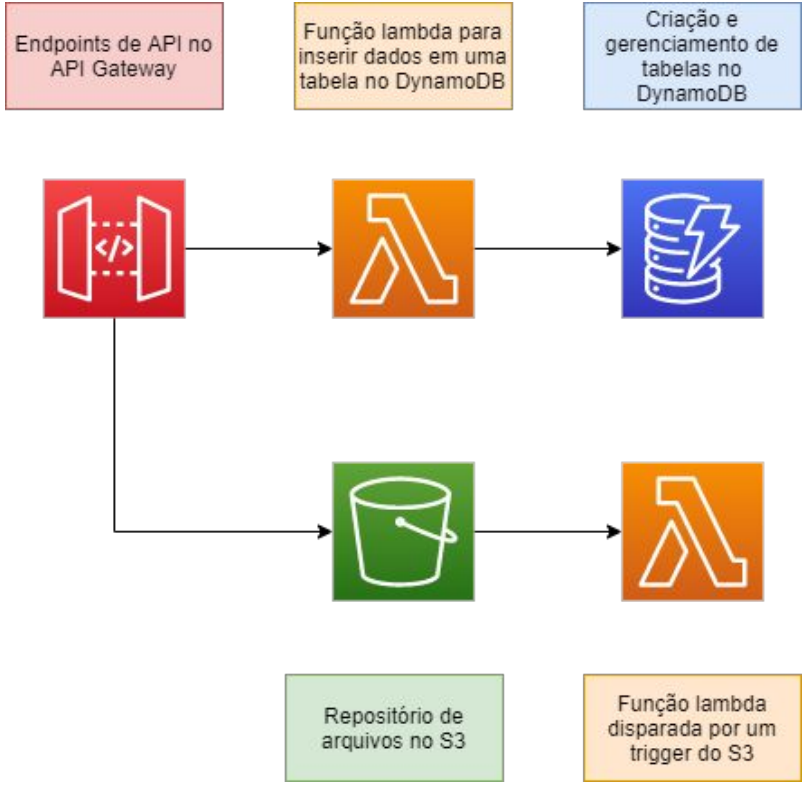
Construindo nossa aplicação serverless

// Introdução ao desenvolvimento serverless na AWS

Objetivos

- Criar uma aplicação com as tecnologias que estudamos
- Testar e verificar resultados

Arquitetura da aplicação



Aula 7

Revisão

// Introdução ao desenvolvimento serverless na AWS

Objetivos

- Relembrar o que estudamos neste curso
- Próximos passos

Aula 7. Etapa 2

O que estudamos

// Introdução ao desenvolvimento serverless na AWS

Temas abordados

Neste curso abordamos serviços que nos permitem criar aplicações web sem servidores de forma prática, escalável e facilmente gerenciável.

Aula 7. Etapa 3

Próximos passos

// Introdução ao desenvolvimento serverless na AWS

Como evoluir a partir deste curso?

Agora você poderá explorar de forma mais profunda os serviços estudados e desenvolver novas arquiteturas de aplicações.

Para saber mais

A AWS disponibiliza uma ampla gama de documentações que podem ser acessadas para maior conhecimento das ferramentas disponíveis.

Dúvidas durante o curso?

> Fórum do curso

> Comunidade online (Discord)

SCAN ME