



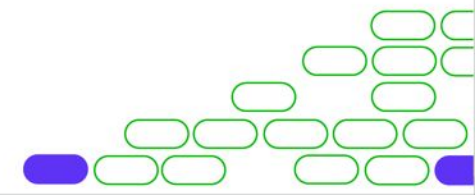
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Capítulo 1. Ferramentas para Gerenciamento e Configuração de Ambientes na AWS

Prof. Filipe Almeida





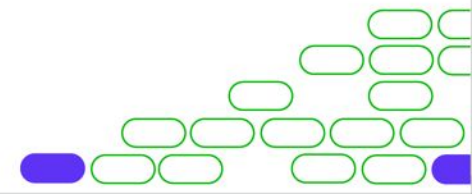
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 1.1. Management Console and Mobile Application

Prof. Filipe Almeida



# Nesta aula

- ❑ AWS Management Console.
- ❑ AWS Console Mobile Application.



# AWS Management Console

- Interface Web para acesso à sua conta da AWS;
  - Iniciar e Interromper suas instâncias EC2;
  - Criar Buckets S3;
  - Criar bancos de dados no RDS.



# AWS Management Console

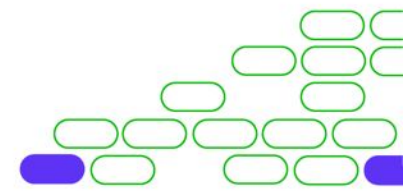
- O acesso pode ser feito via:
  - Usuário raiz da conta;
  - Usuário do IAM.



# AWS Management Console

Usuário raiz da conta da AWS:

- E-mail/Senha de criação da conta;
- Possui permissão total na conta, por padrão;
- É recomendado que não seja utilizado o usuário raiz para operações na conta;
- Habilitar o MFA.



# AWS Management Console

Usuário raiz da conta da AWS:

- Deve permitir explicitamente acesso ao billing via usuários do IAM.

# AWS Management Console

Usuário do IAM:

- Criado dentro da sua conta da AWS, no IAM;
- Recebe privilégios através de políticas de acesso;
- Os acessos aos recursos seguem as políticas definidas.





# AWS Console Mobile Application

- É um console mobile para gerenciamento limitado a alguns recursos e funcionalidades da AWS;
- É um complemento do console web.



# AWS Console Mobile Application

- Permite gerenciar algumas funcionalidades dos seguintes recursos:
  - EC2;
  - Load Balancers;
  - Route 53;
  - RDS;
  - Beanstalk;
  - DynamoDB;
  - CloudFormation;
  - CloudWatch.



# Conclusão

- ✓ Vimos duas formas de criar e gerenciar infraestrutura e ambientes na AWS:
  - Console de Gerenciamento Web;
  - Mobile Application.



# Próxima aula

- ❑ AWS Management Console HandsOn;
- ❑ AWS Mobile Application HandsOn.





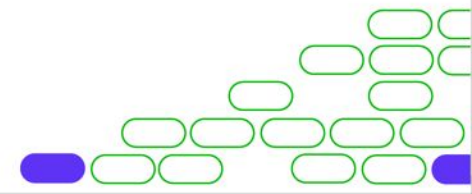
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 1.2. Management Console and Mobile Application - HandsOn

Prof. Filipe Almeida



# Nesta aula

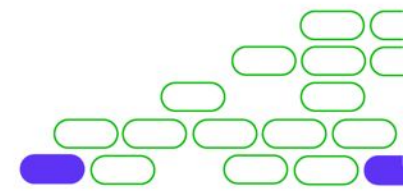
- ☐ Acesso básico via conseole de Gerenciamento da AWS.
- ☐ Habilitar acesso ao Billing via usuário do IAM.
- ☐ Acesso básico via Console Mobile Application.



# Próxima aula

## ☐ AWS CLI:

- Utilizando o CLI;
- Alguns comandos básicos;
- Permissões para o AWS CLI;
- Gerenciamento de múltiplas contas no AWS CLI.





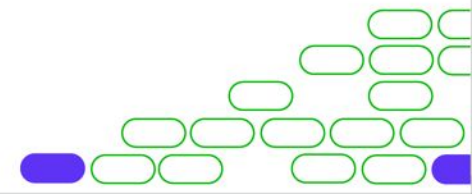
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 1.3. AWS CLI

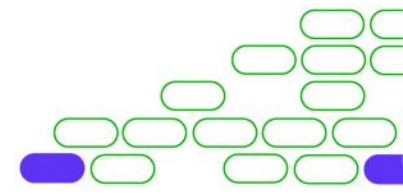
Prof. Filipe Almeida





# Nesta aula

- ❑ AWS Command Line Interface (AWS CLI).



# AWS CLI

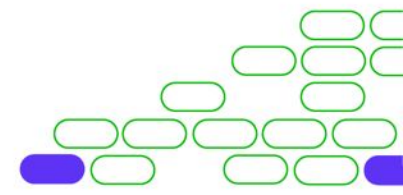
- É um pacote para linha de comando;
- Vários SO's;
- Permite criar, automatizar, gerenciar e destruir recursos e ambientes na AWS através de um terminal;



# AWS CLI

```
<aws ec2 start-instances --instance-ids  
i-1348636c>
```

- Inicia uma instância EC2 baseado no ID informado.

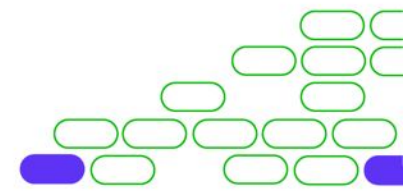


# AWS CLI

```
<aws s3 cp myfolder
```

```
s3://mybucket/myfolder --recursive>
```

- Copia os arquivos da pasta local “myfolder” para uma pasta “myfolder” dentro do bucket S3 criado.



# AWS CLI - Permissões

- Utilizam credenciais de usuários do IAM.
- Seguem as permissões aplicadas a esse usuário.
- Utilizam credenciais programáticas.

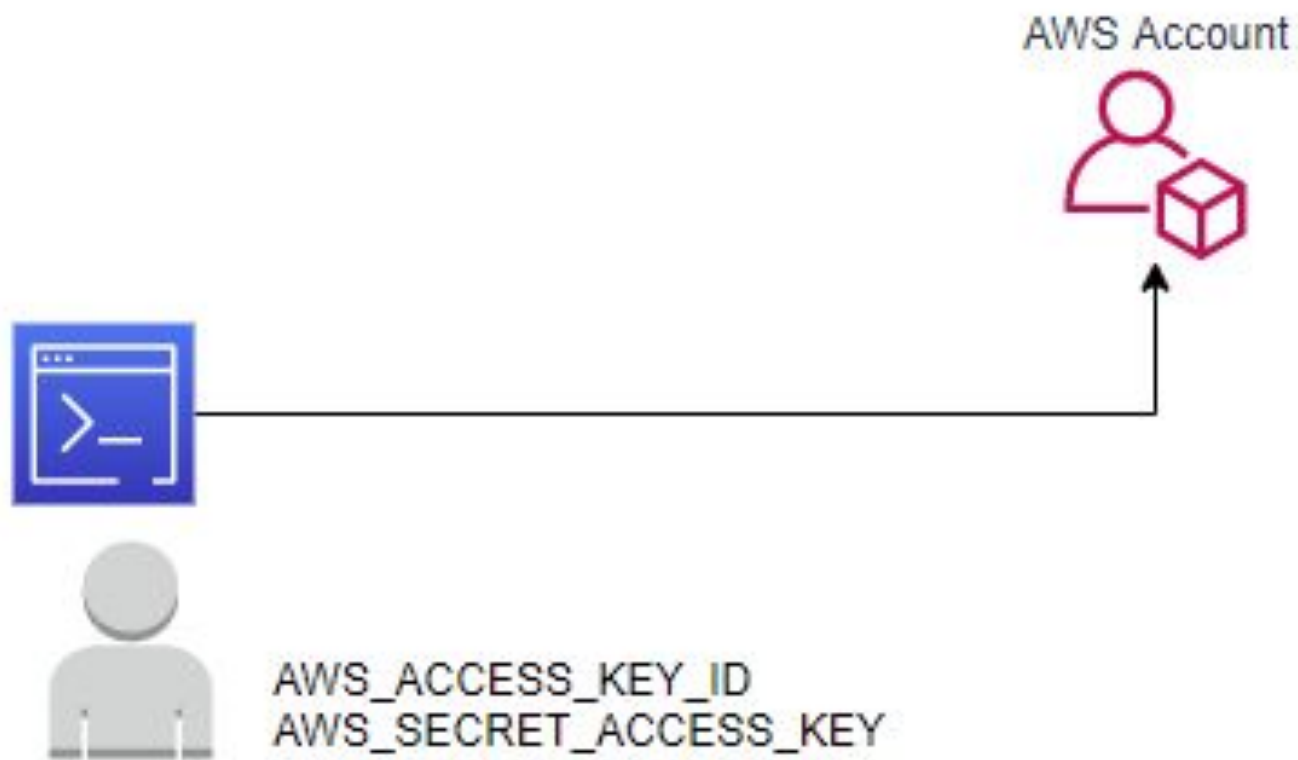


# AWS CLI - Permissões

- Requer um cuidado a mais com a segurança das chaves programáticas.



# AWS CLI



# AWS CLI – Gerenciamento de múltiplas contas

Surge um problema!

- Necessidade de gerenciamento de várias contas da AWS;
- Reconfiguração frequente de credenciais.

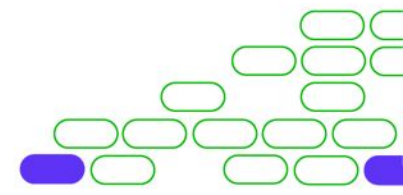




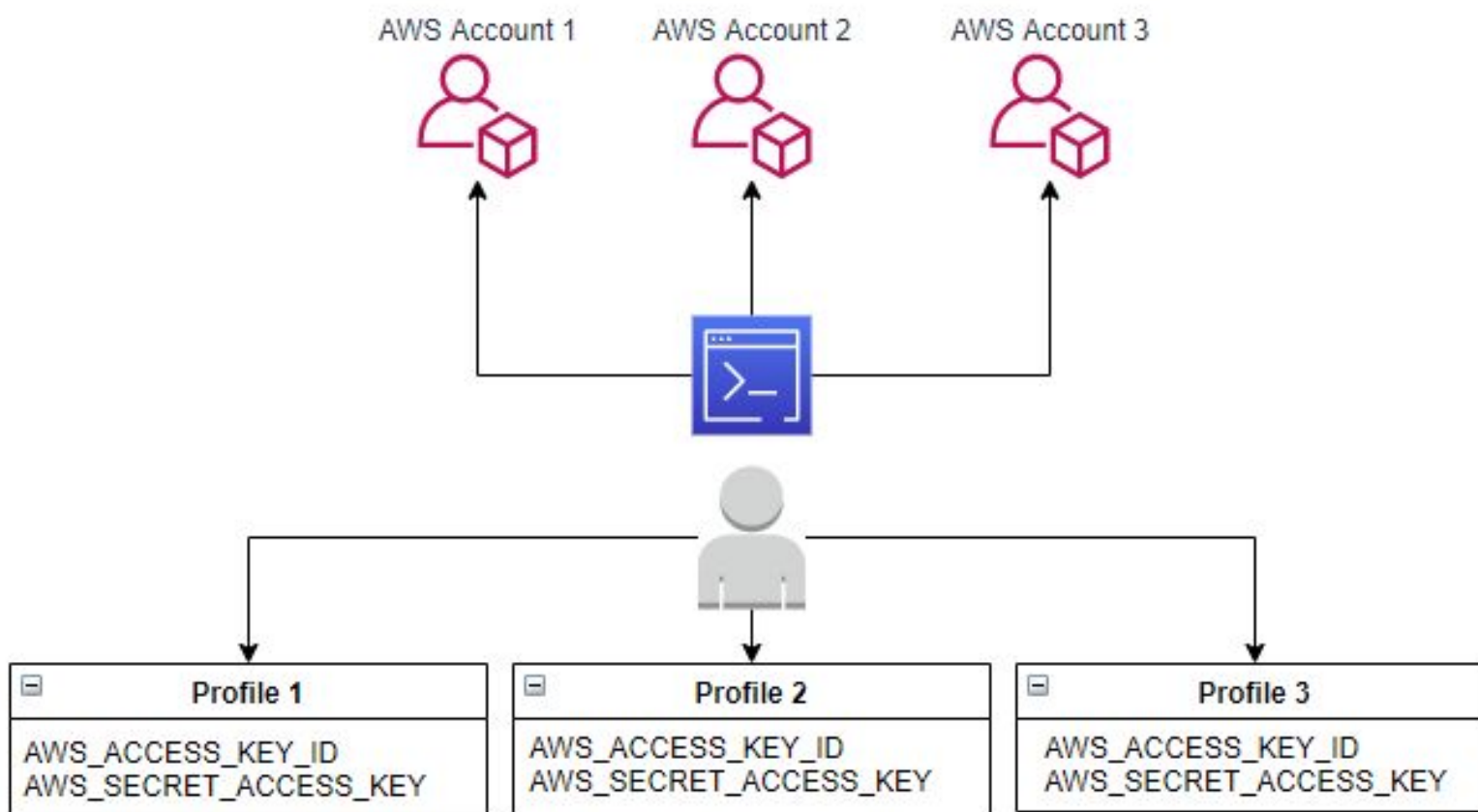
# AWS CLI – Gerenciamento de múltiplas contas

Qual a solução?

- Perfis de acesso no AWS CLI:
  - Facilita o acesso a múltiplas contas da AWS trocando apenas o perfil de acesso utilizado naquele momento.



# AWS CLI – Gerenciamento de múltiplas contas



# Conclusão

- ✓ Vimos mais uma forma de criar e gerenciar recursos na AWS:
  - AWS CLI;
    - Seguem as permissões do IAM;
    - Configuração de perfis de acesso.



# Próxima aula

- ❑ AWS CLI HandsOn.





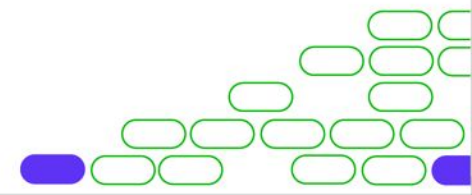
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 1.4. AWS CLI - HandsOn

Prof. Filipe Almeida



# Nesta aula

- ☐ Instalação do AWS CLI;
- ☐ Configurando credenciais no seu terminal;
- ☐ Alguns comandos básicos;
- ☐ Permissões;
- ☐ Configuração de perfis de acesso.



# Próxima aula

- ❑ AWS SDK.





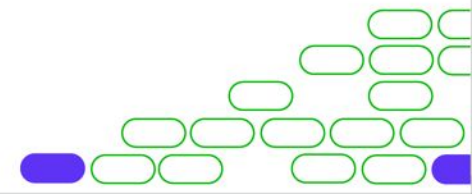
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 1.5. AWS SDK

Prof. Filipe Almeida





# Nesta aula

- ❑ Uma breve introdução aos SDKs da AWS.



# AWS SDK

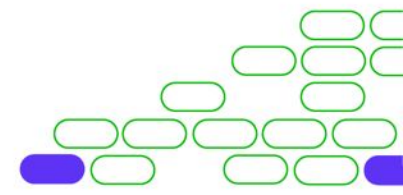
- Software Development Kit;
- Bibliotecas fornecidas pela AWS compatíveis com várias linguagens de programação.



# AWS CLI

Linguagens suportadas:

- Java;
- C++;
- Go;
- JavaScript;
- Node.js;
- .NET;



# AWS SDK

- Escrever códigos de provisionamento e gerenciamento de infra na AWS na sua linguagem de programação preferida.

# Conclusão

- ✓ Vimos mais uma forma de criar e gerenciar recursos na AWS:
  - AWS SDK.



# Próxima aula

- ❑ AWS SDK HandsOn.





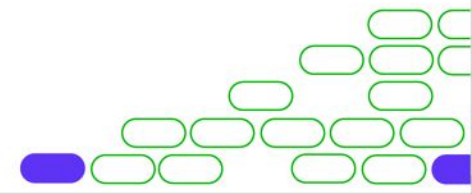
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 1.6. AWS SDK - HandsOn

Prof. Filipe Almeida



# Nesta aula

- ❑ Gerenciando uma infraestrutura na AWS usando Código e o SDK.





# Próxima aula

- ❑ Capítulo 2 – Soluções para IaC na AWS.





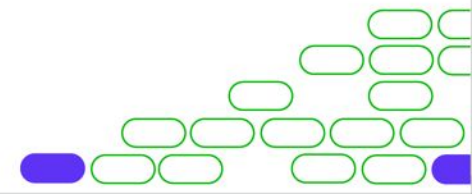
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Capítulo 2. Soluções para IaC na AWS

Prof. Filipe Almeida





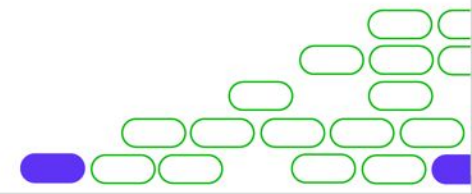
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 2.1. IaC – CloudForm. E Terraform

Prof. Filipe Almeida



# Nesta aula

- ❑ Soluções para IaC na AWS:
  - ❑ Por que IaC?
  - ❑ CloudFormation;
  - ❑ Terraform;



# O que é a IaC – Um outro caminho

- Existe um outro caminho...
- Derivada dos conceitos de DevOps;
- Utilizar Código para gerenciar Infraestrutura;
  - Sem acesso físico direto;
  - Sem acesso a consoles, portais e ferramentas de configuração.



# laC – Vantagens

- Reutilização do Código;
- Agilidade;
- Diminui a frequência de erros de provisionamento.



# laC – Vantagens

- Versionamento do ambiente;
- Automatização na criação de ambientes;
- Facilita a documentação do ambiente;
- Facilita o controle de custos.



## IaC – Desvantagens

- Curva de aprendizado média;
- Tempo maior para o primeiro provisionamento.

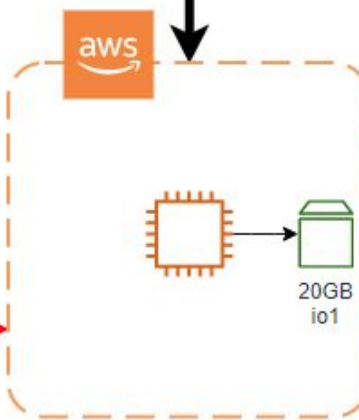




# IaC

YAML

```
MyEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: "ami-79fd7eee"
    KeyName: "testkey"
    BlockDeviceMappings:
      - DeviceName: "/dev/sdm"
        Ebs:
          VolumeType: "io1"
          Iops: "200"
          DeleteOnTermination: "false"
          VolumeSize: "20"
      - DeviceName: "/dev/sdk"
        NoDevice: {}
```



# AWS Cloud Formation

- Ferramenta nativa da AWS para IaC;
- Somente dentro da AWS e algumas soluções de terceiros que estejam no registro público do Cloud Formation.



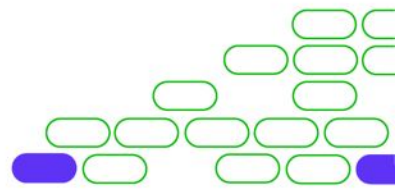
# AWS Cloud Formation

- Os códigos são escritos em:
  - YAML
  - JSON





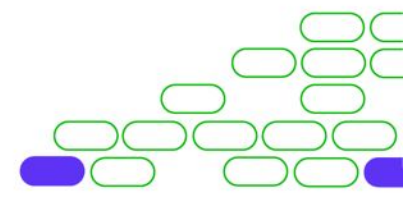
```
"MyEC2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "ImageId" : "ami-79fd7eee",
    "KeyName" : "testkey",
    "BlockDeviceMappings" : [
      {
        "DeviceName" : "/dev/sdm",
        "Ebs" : {
          "VolumeType" : "io1",
          "Iops" : "200",
          "DeleteOnTermination" : "false",
          "VolumeSize" : "20"
        }
      },
      {
        "DeviceName" : "/dev/sdk",
        "NoDevice" : {}
      }
    ]
  }
}
```



# YAML



```
MyEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: "ami-79fd7eee"
    KeyName: "testkey"
    BlockDeviceMappings:
      - DeviceName: "/dev/sdm"
        Ebs:
          VolumeType: "io1"
          Iops: "200"
          DeleteOnTermination: "false"
          VolumeSize: "20"
      - DeviceName: "/dev/sdk"
        NoDevice: {}
```



# AWS Cloud Formation

- O fluxo do provisionamento:
  - Código escrito;
  - Upload para o S3/Console do CF;
  - Criar uma Stack no CF.



# Terraform - Hashicorp

- Ferramenta de terceiro, amplamente usada na comunidade de DevOps;
- Se conecta com múltiplos provedores de nuvem.



# Terraform - Hashicorp

- Idempotência – Pode ser aplicado várias vezes e reconhece alterações no código;
- Linguagem de alto nível, fácil de reutilizar.





# Terraform - Requisitos

- Ambiente com o terraform instalado;
  - Vários SO.
- Permissões necessárias para o provisionamento de recursos.



# Conclusão

- ✓ Vimos mais uma forma de criar e gerenciar recursos na AWS:
  - CloudFormation;
  - Terraform.



# Próxima aula

- ❑ AWS CloudFormation HandsOn.





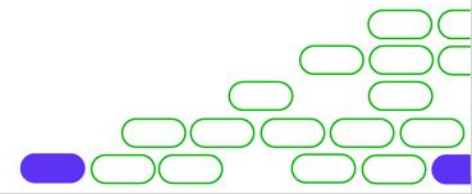
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 2.2. Cloud Formation HandsOn

Prof. Filipe Almeida



# Nesta aula

- ❑ CloudFormation HandsOn:
  - ❑ Provisionar um ambiente na AWS utilizando o CF.



# Conclusão

- ✓ Vimos como provisionar infraestrutura na AWS através da sua solução nativa de IaC, o CloudFormation.



# Próxima aula

- ❑ AWS Terraform HandsOn.





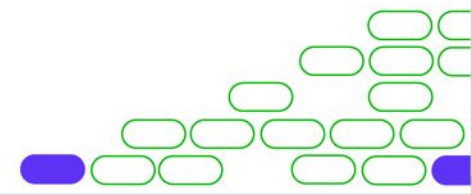
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 2.3. Terraform HandsOn

Prof. Filipe Almeida





# Nesta aula

- ❑ Terraform HandsOn:
  - ❑ Provisionar um ambiente na AWS utilizando o terraform como ferramenta de IaC.



# Conclusão

- ✓ Vimos como provisionar infraestrutura na AWS através da uma solução de terceiro de mercado, o Terraform.



# Próxima aula

- ❑ Capítulo 3 – AWS Lambda.





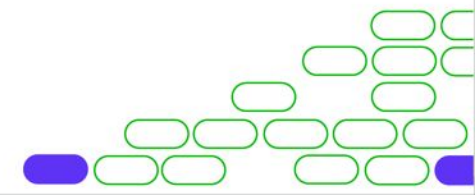
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Capítulo 3. AWS Lambda

Prof. Filipe Almeida





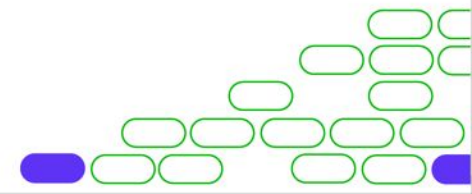
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 3.1. AWS Lambda

Prof. Filipe Almeida



# Nesta aula

- ❑ O dilema do desenvolvedor;
- ❑ Introdução ao conceito de serverless e suas vantagens;
- ❑ AWS Lambda.

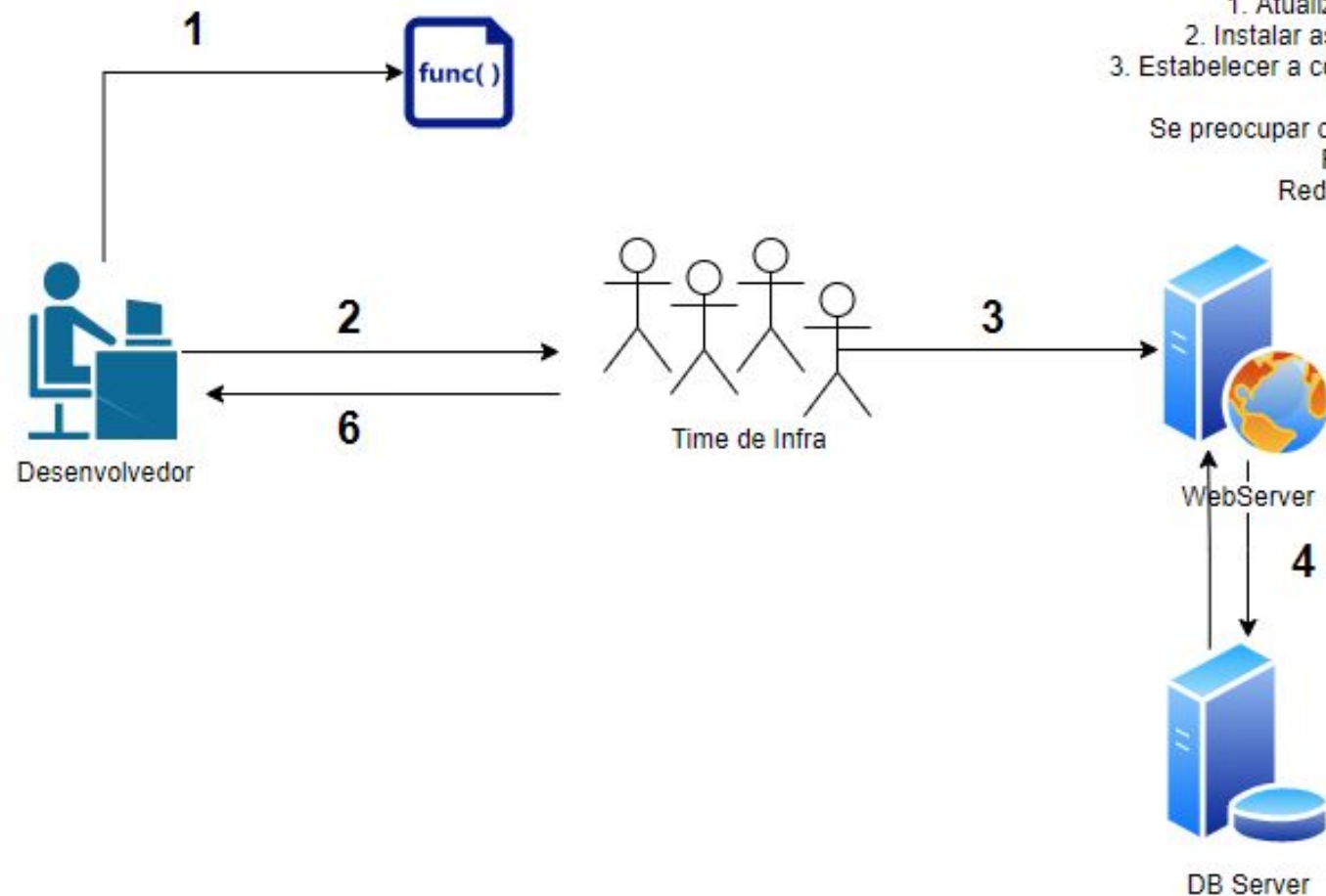


5

Preparar o ambiente

1. Atualizar o sistema operacional
2. Instalar as dependências da aplicação
3. Estabelecer a comunicação com o banco de dados

Se preocupar com a segurança desse servidor  
Rotina de Backup  
Redundância do servidor



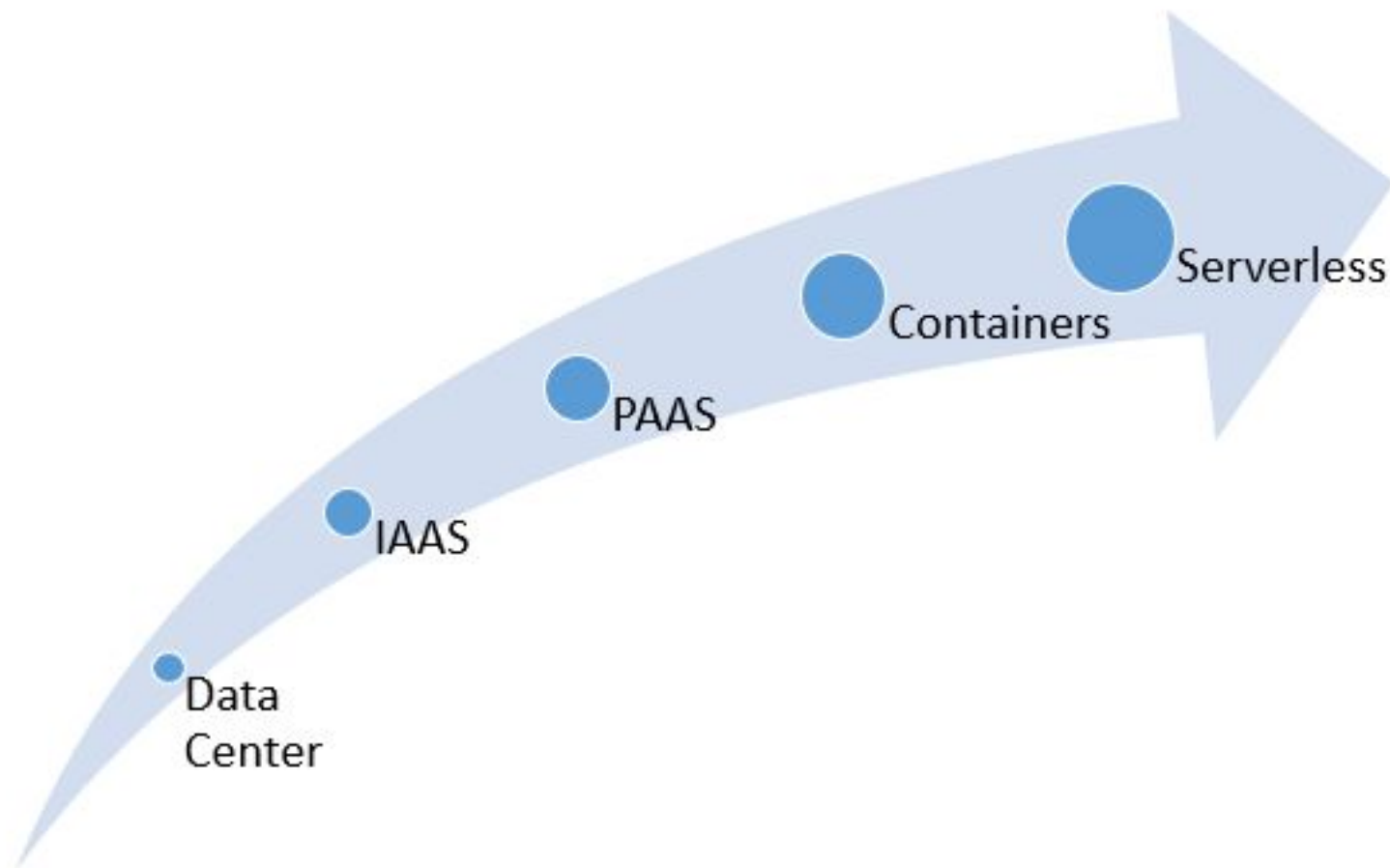
# Arquiteturas Serverless

- É um conceito fortemente relacionado ao surgimento da computação em nuvem;
- Precisamos entender o histórico de evolução tecnológica até o serverless.





# Arquiteturas Serverless



# Arquiteturas Serverless - Benefícios

- Foco no negócio;
- Não exige administração de infraestrutura;
- Escalabilidade;
- Alta Disponibilidade;
- Pague conforme o uso.



# Arquiteturas Serverless - Benefícios

- Diminuição dos riscos de ataques à sua infraestrutura;
- Inovação rápida – Time To Market.



# AWS Lambda

- Permite a execução de códigos em resposta a eventos;
  - Agendamentos (cron);
  - Mudanças no ambiente.



# AWS Lambda – Linguagens nativas

- Linguagens nativamente suportadas pelo Lambda:
  - Node.js;
  - Python;
  - Java;
  - Go;
  - PowerShell;
  - C#;
  - Ruby.



# AWS Lambda – Custos

- Sem custos para criar uma função;
- É cobrado por:
  - Tempo de execução;
  - Quantidade de memória utilizada;
  - Quantidade de requisições.
- 1M Free -> \$0.20/M



# AWS Lambda em ação

- Dispositivos Echo da AWS;
- Aplicações CloudNative, como Netflix, AirBnb etc.



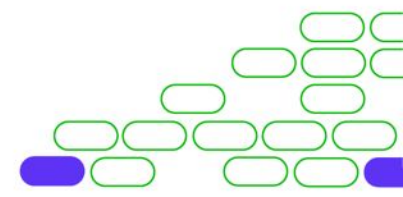
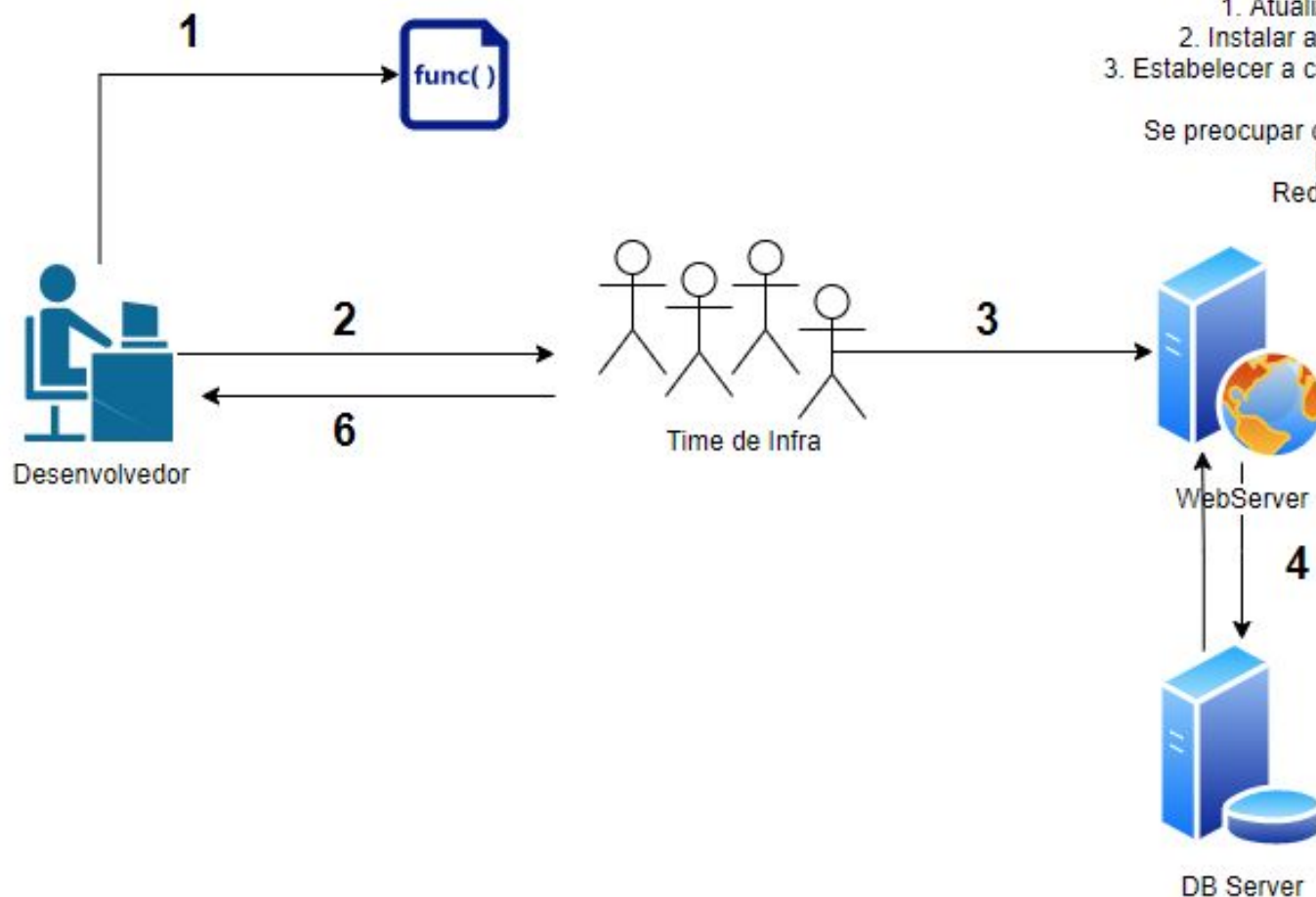


5

Preparar o ambiente

1. Atualizar o sistema operacional
2. Instalar as dependências da aplicação
3. Estabelecer a comunicação com o banco de dados

Se preocupar com a segurança desse servidor  
Rotina de Backup  
Redundância do servidor





# Conclusão

- ✓ Definição do conceito de serverless.
- ✓ AWS Lambda.



# Próxima aula

- ❑ AWS Lambda HandsOn.





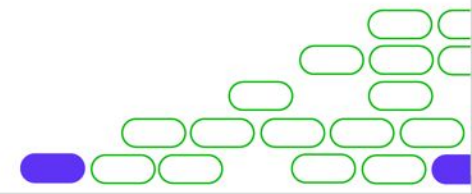
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 3.2. AWS Lambda HandsOn

Prof. Filipe Almeida



# Nesta aula

- ❑ Como automatizar algumas tarefas da AWS usando lambda.
- ❑ Uma aplicação web simples utilizando lambda.



# Conclusão

- ✓ Função lambda para gerenciar recursos na AWS de maneira automatizada.
- ✓ Aplicação simples em lambda.



# Próxima aula

- ❑ Cap. 4 – AWS API Gateway e Soluções AWS para IoT.





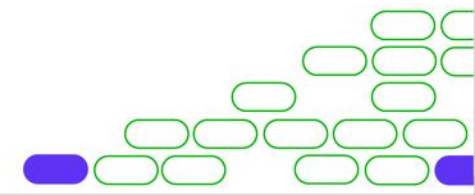
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Capítulo 4. Amazon API Gateway

Prof. Filipe Almeida





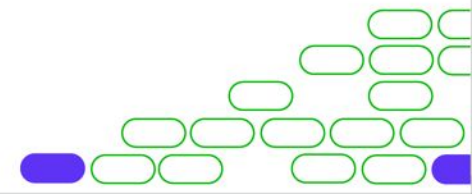
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 4.1. Amazon API Gateway

Prof. Filipe Almeida





# Nesta aula

- ❑ O que é uma API?
- ❑ Introdução à API Gateway.



# API – Application Programming Interface

- Conjunto de normas que possibilita comunicação entre duas aplicações;
- Série de padrões e protocolos.



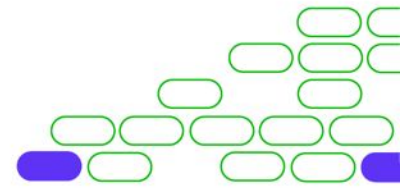
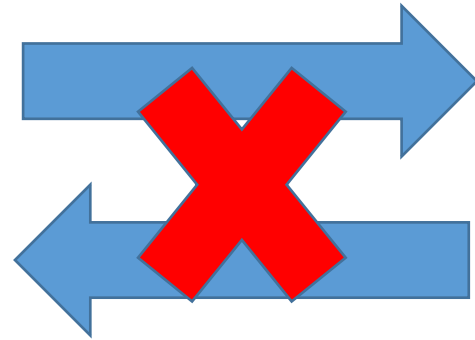
# API – Application Programming Interface

- Possibilita criação de softwares que se comunicam com outras plataformas;
- App que utiliza geolocalização via Google Maps;
- App que integra com Spotify e Instagram.

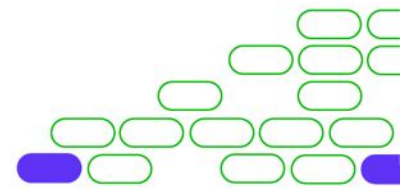


# API – Application Programming Interface

- Analogia do restaurante;
  - Casal;
  - Cozinheiros;
  - Garçom;
  - Menu.
- Sites de pacotes de viagem.

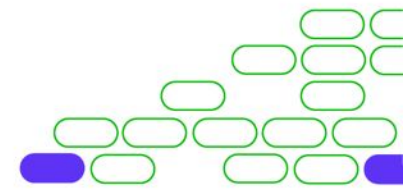
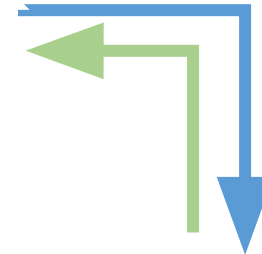
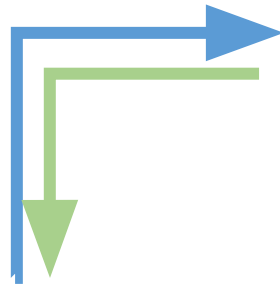


# API – Application Programming Interface





# API – Application Programming Interface

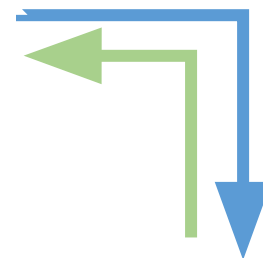
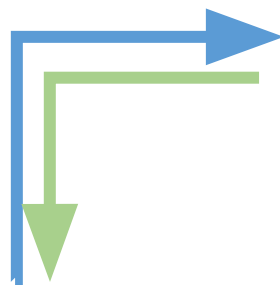


# API – Application Programming

Interface

API

Origem da  
Requisição



Aplicação





# Amazon API Gateway

- APIs REST e WebSocket escaláveis;
- Serverless;
- Facilita a disponibilização de aplicações construídas na AWS para terceiros.



# Amazon API Gateway

- As APIs RESTful são:
  - Baseadas em HTTP;
  - Implementam os métodos HTTP padrão:
    - GET
    - POST
    - PUT
    - PATCH
    - DELETE

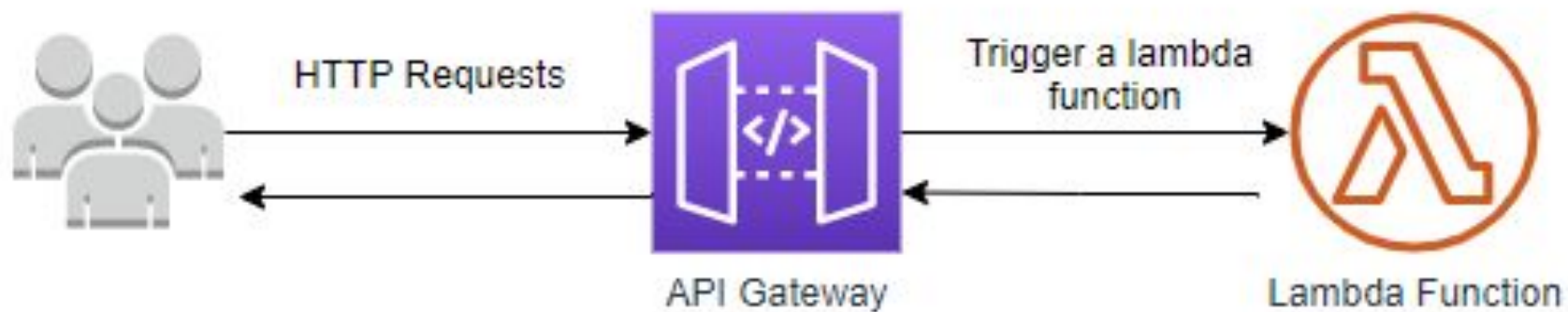


# Amazon API Gateway

- As APIs WebSockets:
  - Seguem o protocol WebSocket, full-duplex entre cliente e servidor com estado;
  - Roteia mensagem recebidas baseado no conteúdo da mensagem.



# Amazon API Gateway na prática



# Conclusão

- ✓ Vimos um serviço da AWS para criação de APIs;
- ✓ Serverless:
  - Escalável;
  - Completamente gerenciado pela AWS;
  - Seguro;
  - Resiliente.



# Próxima aula

- ☐ Amazon API Gateway HandsOn;
- ☐ Construindo uma API 100% Serverless na AWS.





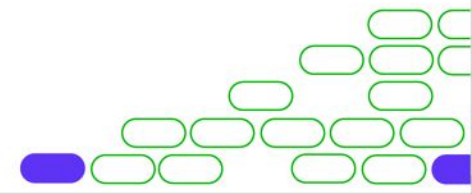
Faculdade



# Soluções para Desenvolvimento e Ferramentas de Gerenciamento

Aula 4.2. Amazon API Gateway - HandsOn

Prof. Filipe Almeida



# Nesta aula

- ❑ Amazon API Gateway HandsOn;
- ❑ Construir uma API 100% Serverless na AWS.





# Conclusão

- ✓ Construimos uma API que publica um serviço na Web via Lambda.

