Meus Prints

quinta-feira, 14 de agosto de 2025 10:30

Domínios abordados

Domínio	% do exame
Domínio 1: Conceitos de nuvem	26%
Domínio 2: Segurança e conformidade	25%
Domínio 3: Tecnologia	33%
Domínio 4: Faturamento e preços	16%
TOTAL	100%

Questões

- Tipos de Questão: Múltipla escolha, Resposta múltipla
- Quantidade: 65 questões, sendo 50 avaliadas
- Aprovação: 70% (Pontuação de 0 a 1000, mínimo 700)

Cloud Computing é

A computação em nuvem é a entrega de recursos de TI sob demanda por meio da Internet com definição de preço de pagamento conforme o uso.

aws

Benefícios

Troque despesas iniciais por despesas variáveis

Benefícios

Pare de tentar adivinhar a capacidade

Benefícios

Beneficie-se de enormes economias de escala

IaaS

Infraestrutura como serviço

- Componentes básicos de TI
- Disco, memória, CPU
- Usuário: Sysadmin
- Usuário gerencia infraestrutura

Amazon EC2

PaaS

Plataforma como serviço

- Plataforma para implantar (deploy)
- Não se preocupe com infraestrutura
- Usuário: Desenvolvedores

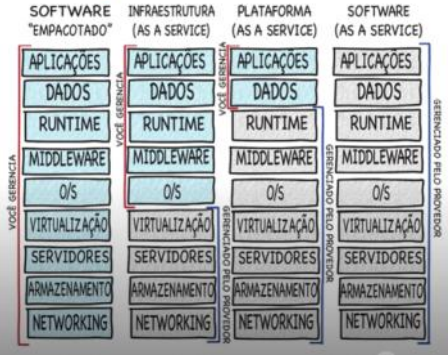
Elastic Beanstalk

SaaS

Software como serviço

- Produto completo, executado e gerenciado pelo provedor
- Não se preocupe com infraestrutura
- Pense apenas em utilizar

Comparações



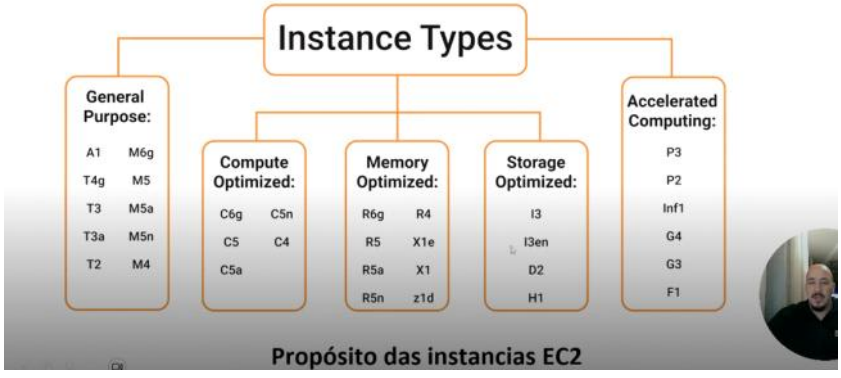
Infraestrutura Global AWS é

- Infraestrutura de datacenters em todo o mundo que fornecem os diversos serviços que você pode utilizar na AWS
- Composto por Regiões e Zonas de disponibilidade
- Vantagens: Alta disponibilidade, Tolerância a falhas

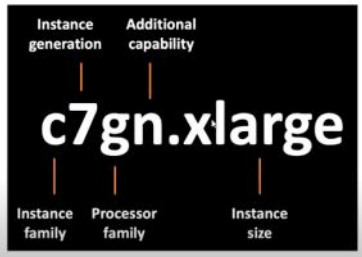
O que são instâncias EC2

General Purpose	Compute Optimised	Memory Optimised	Accelerated Computing	Storage Optimised
A1 — ARM based core and custom silicon	C4 — Compute - CPU intensive apps and DBs	R4 — RAM - Memory Intensive apps and DB's	P2 — Processing optimised - Machine Learning	H1 — High Disk Throughput - Big data clusters
T2 — Tiny - Web servers and small DBs		X1 — Xtreme RAM - For SAP/Spark	G3 — Graphics Intensive - Video and Streaming	I3 — IOPS - No SQL DBs
M4 — Main - App servers and general purpose		z1d — High Compute and High Memory - Gaming	F1 — Field Programmable - Hardware acceleration	D2 — Dense Storage - Data Warehousing

Os tipos de instâncias EC2



Os tipos de instâncias EC2

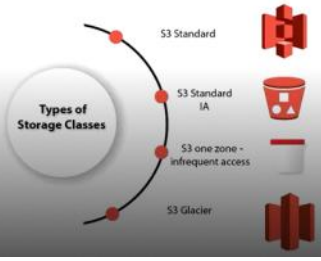


Convenção do nome dos tipos de instância
Entendendo mais sobre os nome

Amazon S3

O Amazon S3 possui algumas classes de storages onde conseguimos economizar nos custos.

Um exemplo muito bom para entendermos as diferentes classes do S3 é o exemplo do exames de Hospital.

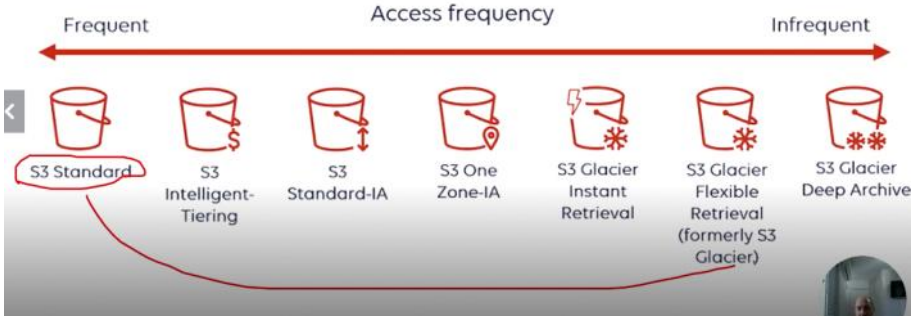


Amazon S3

	S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved

https://aws.amazon.com/pt/s3/storage-classes/

Amazon S3



Amazon S3

Podemos utilizar regra de ciclo de vida para definir a forma como o Amazon S3 gere os objetos durante o seu tempo de vida.
O Lifecycle permite fazer a transição de objetos e migrar automaticamente para a classe Glacier.

Criação e uso de imagens AMI

O que é a imagem de máquina da Amazon (AMI)?

No Amazon EC2 (Elastic Compute Cloud), uma AMI (Amazon Machine Image) é uma imagem de máquina virtual pré-configurada, que inclui as informações necessárias para iniciar uma instância, como o sistema operativo, o servidor de aplicações e as aplicações.

Criação e uso de imagens AMI

Veremos os principais pontos sobre as AMIs do EC2

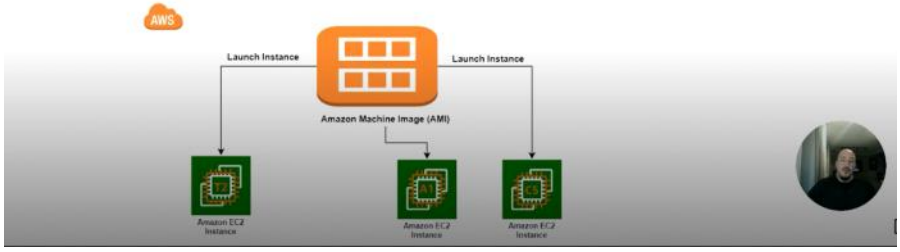
1. Criação: As AMIs podem ser criadas a partir de instâncias em execução ou paradas. Isto permite-lhe capturar um instantâneo do seu ambiente configurado.

2. AMIs públicas e privadas: A AWS fornece uma variedade de AMIs públicas que podem ser usadas, ou você pode criar e usar suas próprias AMIs privadas para segurança e personalização.

[5]

Criação e uso de imagens AMI

Entender as AMIs é crucial para gerenciar e implantar instâncias no EC2 de forma eficiente. Elas fornecem uma base para a criação de ambientes consistentes e reproduzíveis na nuvem.



Conteúdo Programático

- **Módulo 2: Computação na Nuvem com EC2**
  - Amazon EC2
  - Armazenamento na Nuvem
  - Gerenciamento de instâncias EC2

Snapshot com EBS

Para entendermos de fato o que são os snapshots EBS, para isto devemos primeiro compreender os componentes os serviços de nuvem da Amazon:

O Amazon Elastic Compute Cloud **(EC2)** é um serviço da Amazon que fornece capacidade de computação escalável na nuvem do Amazon Web Services (AWS) através de máquinas virtuais, conhecidas como instâncias.

Snapshot com EBS

O Amazon Elastic Block Store **(EBS)** é um serviço para fornecer armazenamento em bloco fiável (também conhecido como volumes ou discos rígidos). Foi concebido para ser utilizado com instâncias do Amazon Elastic Compute Cloud (EC2).

Snapshot com EBS

Quando falamos sobre infraestrutura, isto tem a ver com o disco de servidor virtual, vale lembrar que quando falamos de infraestrutura estamos falando do modelo de Cloud.

Neste caso estamos falando sobre o modelo IAAS.

Snapshot com EBS

Quanto custam os snapshots do AWS?

Os snapshot do EBS são oferecidos a diferentes custos em diferentes regiões.

- Na região leste dos EUA, por exemplo, os snapshots são cobrados a um custo fixo de US$ 0,05 por GB por mês.
- Na região da UE (Londres), os snapshots são cobrados a US$ 0,053 por GB por mês

Criação e uso de imagens AMI

O que é a imagem de máquina da Amazon (AMI)?

No Amazon EC2 (Elastic Compute Cloud), uma AMI (Amazon Machine Image) é uma imagem de máquina virtual pré-configurada, que inclui as informações necessárias para iniciar uma instância, como o sistema operativo, o servidor de aplicações e as aplicações.

Criação e uso de imagens AMI

Veremos os principais pontos sobre as AMIs do EC2

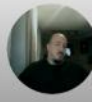
1. Criação: As AMIs podem ser criadas a partir de instâncias em execução ou paradas. Isto permite-lhe capturar um instantâneo do seu ambiente configurado.

2. AMIs públicas e privadas: A AWS fornece uma variedade de AMIs públicas que podem ser usadas, ou você pode criar e usar suas próprias AMIs privadas para segurança e personalização.

Criação e uso de imagens AMI

3. Personalização: É possível personalizar uma instância (instalar software, configurar definições) e, em seguida, criar uma AMI a partir dela. Isso facilita a replicação do seu ambiente.

4. Iniciar instâncias: Para executar instâncias no EC2, seleciona uma AMI. A AMI fornece as informações necessárias para iniciar a instância, como o volume do dispositivo raiz e as permissões de inicialização.

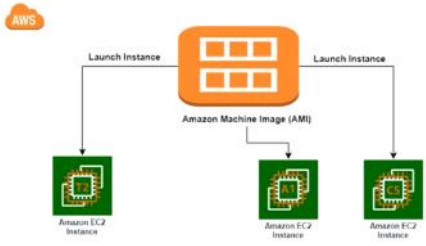


Criação e uso de imagens AMI

5. Tipos de AMI: Existem diferentes tipos de AMIs, incluindo Amazon Linux, Windows e outros. Escolhe-se uma AMI com base nos requisitos da aplicação e do sistema.

Criação e uso de imagens AMI

Entender as AMIs é crucial para gerenciar e implantar instâncias no EC2 de forma eficiente. Elas fornecem uma base para a criação de ambientes consistentes e reproduzíveis na nuvem.



Formação AWS Cloud Foundations

Alexsandro Lechner

Arquiteto de Soluções AWS

linkedin.com/in/alexsandrolechner

Snapshot com EBS

Para entendermos de fato o que são os snapshots EBS, para isto devemos primeiro compreender os componentes os serviços de nuvem da Amazon:
O Amazon Elastic Compute Cloud **(EC2)** é um serviço da Amazon que fornece capacidade de computação escalável na nuvem do Amazon Web Services (AWS) através de máquinas virtuais, conhecidas como instâncias.

Snapshot com EBS

Quando falamos sobre infraestrutura, isto tem a ver com o disco de servidor virtual, vale lembrar que quando falamos de infraestrutura estamos falando do modelo de Cloud.

Neste caso estamos falando sobre o modelo IAAS.

Snapshot com EBS

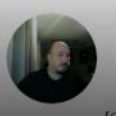
Dito isto, o snapshot do EBS é um serviço de backup nativo do AWS que faz backup dos volumes do EBS em um determinado momento.

E é possível configurar a frequência com que os snapshots são tirados.

Snapshot com EBS

Os instantâneos do EBS armazenam os volumes do EBS no AWS S3, em uma matriz de armazenamento diferente de onde estão os volumes do EBS.
Para fins de recuperação de desastres (DR), os instantâneos do EBS podem ser armazenados em uma região remota.

Podemos também otimizar o armazenamento em SSD para operações de leitura/gravação com uso intensivo de I/O.



Snapshot com EBS

Quanto custam os snapshots do AWS?

Os snapshot do EBS são oferecidos a diferentes custos em diferentes regiões.

- Na região leste dos EUA, por exemplo, os snapshots são cobrados a um custo fixo de US$ 0,05 por GB por mês.
- Na região da UE (Londres), os snapshots são cobrados a US$ 0,053 por GB por mês

Snapshot com EBS

Mas qual seria a diferença entre imagem e o snapshot?

No Amazon EC2, uma imagem de máquina da Amazon (AMI) faz o backup de um servidor inteiro, incluindo todos os volumes EBS anexados.

Um snapshot é uma cópia pontual de um determinado volume. Você pode tirar snapshot de seus volumes EBS e salvá-los no armazenamento S3.

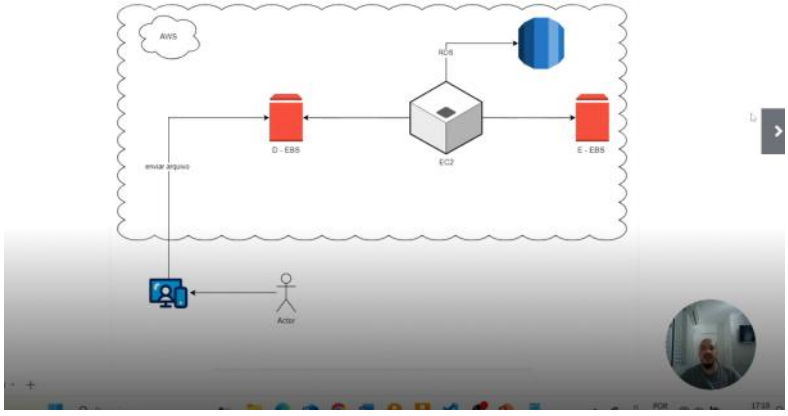
Snapshot com EBS

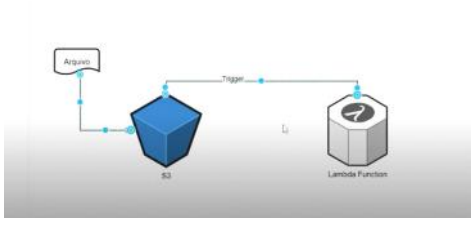
Snapshot com EBS

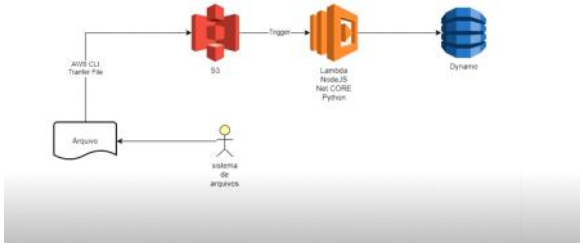
Conclusão:
Os snapshots do Amazon EBS são cópias em pontos no tempo de um volume do Amazon EBS, armazenadas no Amazon S3. Eles podem ser usados para criar um volume do Amazon EBS, aumentar a durabilidade de dados e fornecer um mecanismo de backup e restauração para volumes do EBS.

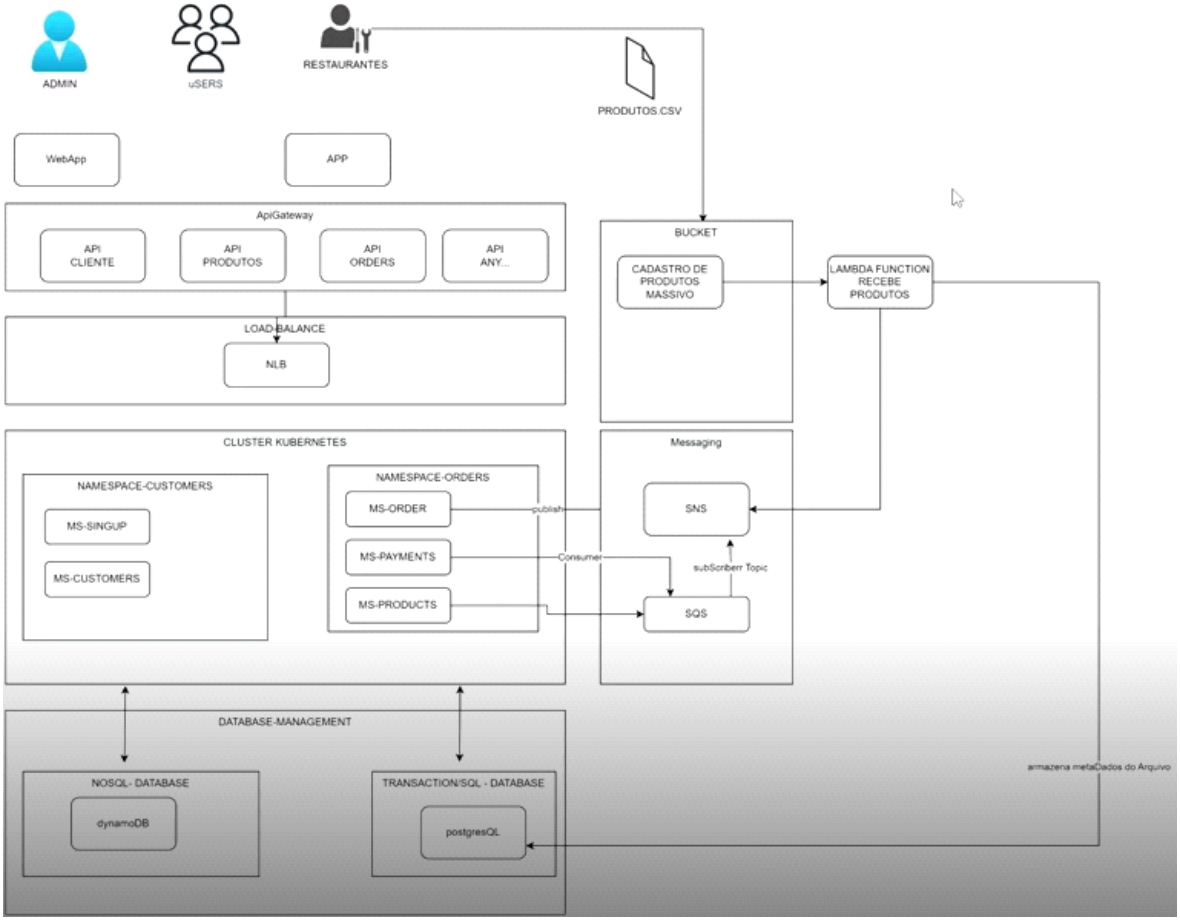
Desafio

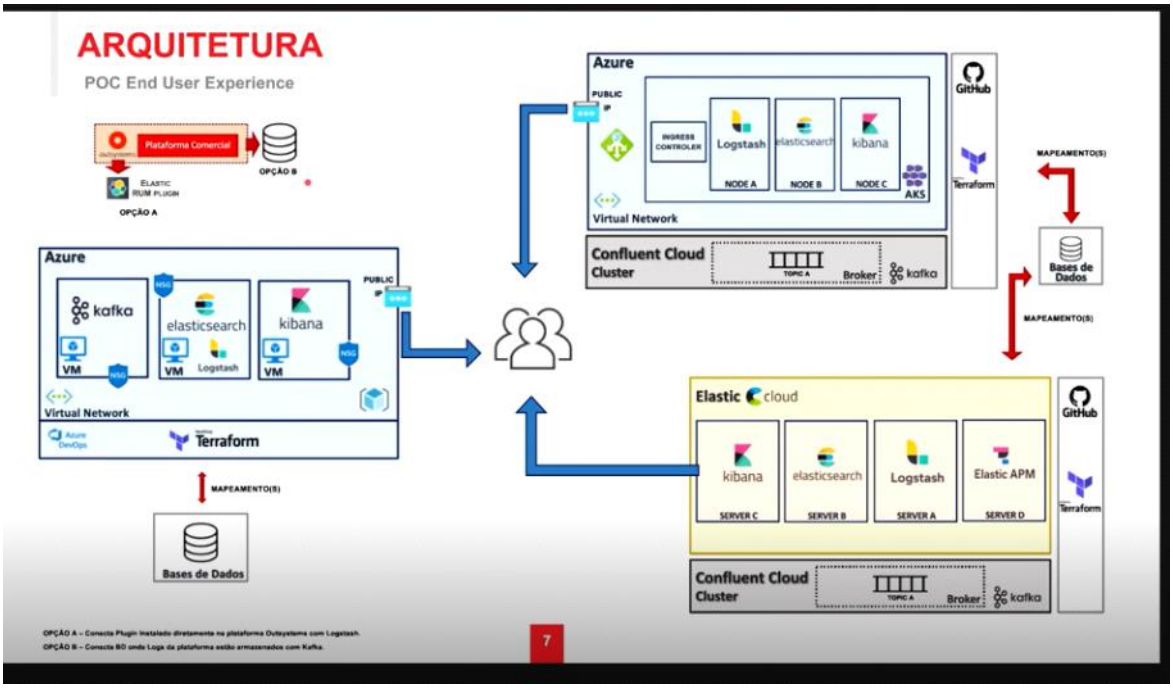
- Acessar o site (www.drawio.com), baixar o programa e instalar
- Criar o desenho de nossa arquitetura com S3 / EC2 / Lambda Function / EBS
- Disponibilizar na plataforma DIO

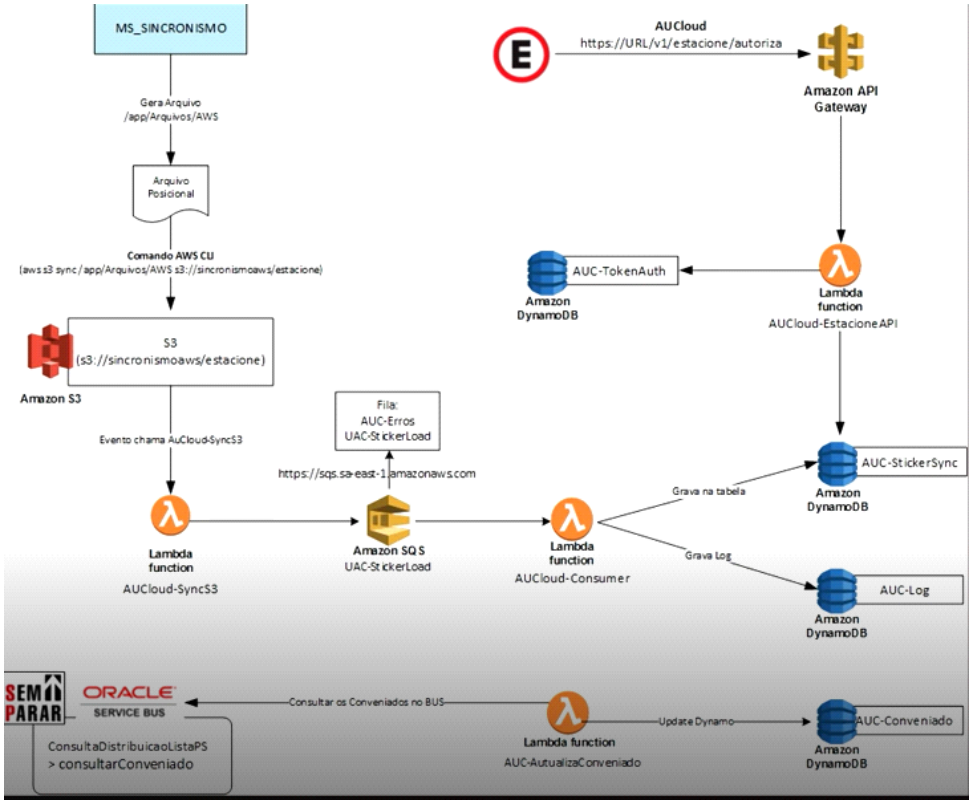


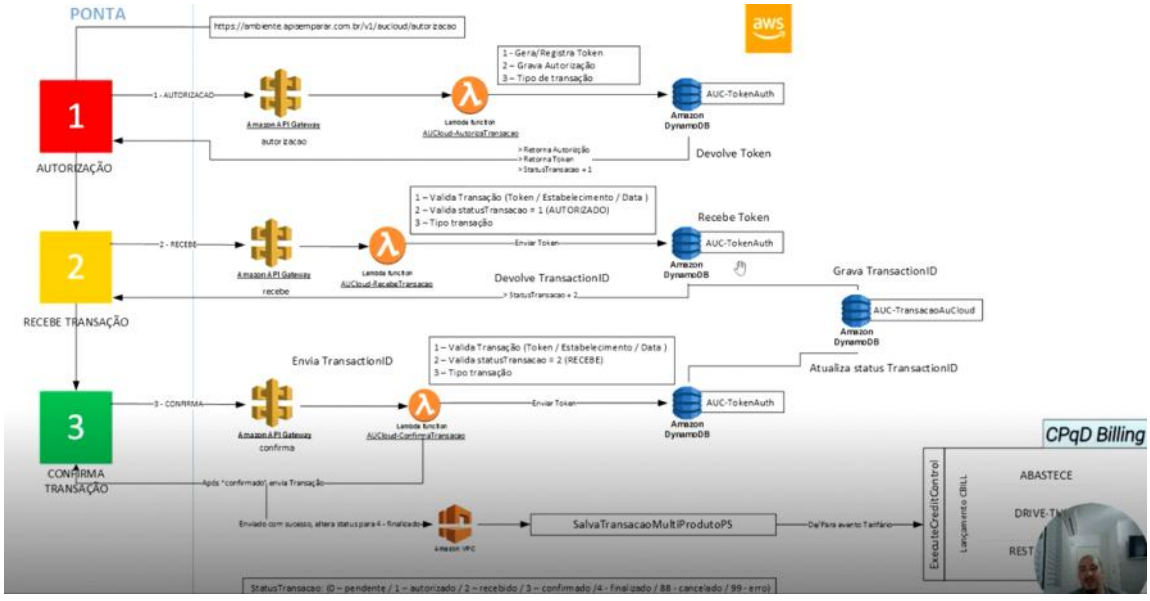


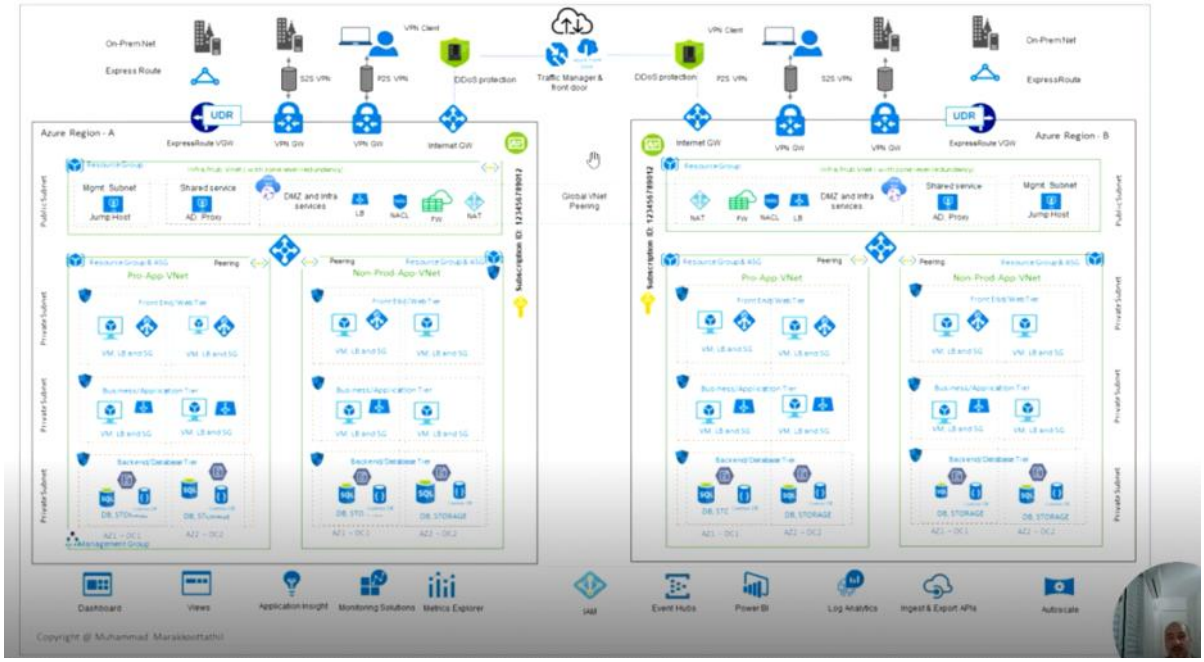


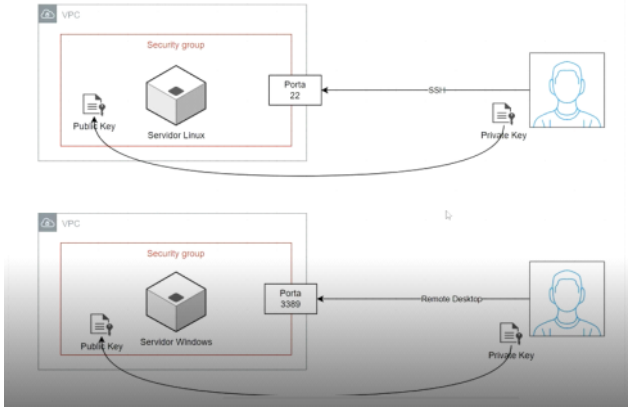

ADMIN
uSERS
RESTAURANTES
PRODUTOS.CSV
WebApp
APP
ApiGateway
API CLIENTE
API PRODUTOS
API ORDERS
API ANY...
LOAD BALANCE
NLB
BUCKET
CADASTRO DE PRODUTOS MASSIVO
LAMBDA FUNCTION RECEBE PRODUTOS
CLUSTER KUBERNETES
NAMESPACE-CUSTOMERS
MS-SINGUP
MS-CUSTOMERS
NAMESPACE-ORDERS
MS-ORDER
MS-PAYMENTS
MS-PRODUCTS
publish
Consumer
Messaging
SNS
subScriberr Topic
SQS
DATABASE-MANAGEMENT
NOSQL- DATABASE
dynamoDB
TRANSACTION/SQL - DATABASE
postgresQL
armazena metaDados do Arquivo

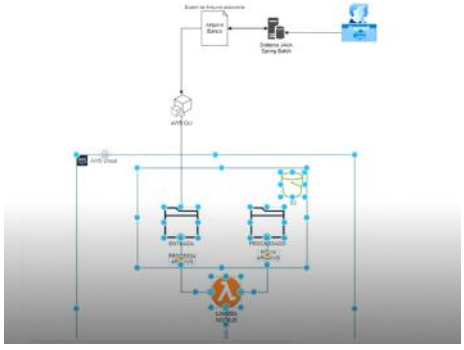

ARQUITETURA
POC End User Experience
Plataforma Comercial
OPÇÃO B
ELASTIC RUM PLUGIN
OPÇÃO A
Azure
PUBLIC IP
INGRESS CONTROLER
Logstash
elasticsearch
kibana
NODE A
NODE B
NODE C
AKS
Virtual Network
Confluent Cloud Cluster
TOPIC A
Broker
kafka
GitHub
Terraform
MAPEAMENTO(S)
Bases de Dados
kafka
VM
elasticsearch
Logstash
kibana
NSG
Azure DevOps
Elastic cloud
kibana
elasticsearch
Logstash
Elastic APM
SERVER C
SERVER B
SERVER A
SERVER D
OPÇÃO A – Conecta Plugin instalado diretamente na plataforma Outsystems com Logstash.
OPÇÃO B – Conecta BD onde Logs da plataforma estão armazenados com Kafka.
7

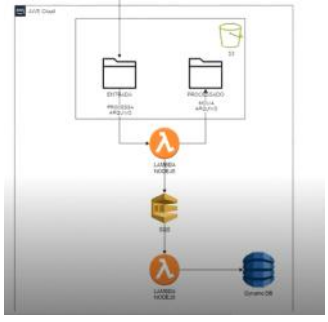

MS_SINCRONISMO
Gera Arquivo
/app/Arquivos/AWS
Arquivo
Posicional
Comando AWS CLI
(aws s3 sync /app/Arquivos/AWS s3://sincronismoaws/estacione)
S3
(s3://sincronismoaws/estacione)
Amazon S3
Evento chama AuCloud-SyncS3
Lambda
function
AUCloud-SyncS3
Fila:
AUC-Erros
UAC-StickerLoad
https://sqs.sa-east-1.amazonaws.com
Amazon SQS
UAC-StickerLoad
Lambda
function
AUCloud-Consumer
Grava na tabela
Grava Log
AUC-StickerSync
Amazon
DynamoDB
AUC-Log
Amazon
DynamoDB
AUCloud
https://URL/v1/estacione/autoriza
Amazon API
Gateway
AUC-TokenAuth
Amazon
DynamoDB
Lambda
function
AUCloud-EstacioneAPI
SEM PARAR
ORACLE
SERVICE BUS
ConsultaDistribuicaoListaPS
> consultarConveniado
Consultar os Conveniados no BUS
Lambda function
AUC-AutualizaConveniado
Update Dynamo
AUC-Conveniado
Amazon
DynamoDB


PONTA
https://ambiente.apisemparar.com.br/v1/aucloud/autorizacao
aws
1
AUTORIZAÇÃO
1 - AUTORIZACAO
Amazon API Gateway
autorizacao
Lambda function
AUCloud-AutorizaTransacao
1 - Gera/Registra Token
2 - Grava Autorização
3 - Tipo de transação
AUC-TokenAuth
Amazon
DynamoDB
Devolve Token
> Retorna Autorização
> RetornaToken
> StatusTransacao = 1
2
RECEBE TRANSAÇÃO
2 - RECEBE
Amazon API Gateway
recebe
Lambda function
AUCloud-RecebeTransacao
1 – Valida Transação (Token / Estabelecimento / Data)
2 – Valida statusTransacao = 1 (AUTORIZADO)
3 – Tipo transação
Enviar Token
Recebe Token
AUC-TokenAuth
Amazon
DynamoDB
Devolve TransactionID
> StatusTransacao = 2
Grava TransactionID
AUC-TransacaoAuCloud
Amazon
DynamoDB
Atualiza status TransactionID
3
CONFIRMA
TRANSAÇÃO
3 - CONFIRMA
Amazon API Gateway
confirma
Envia TransactionID
Lambda function
AUCloud-ConfirmaTransacao
1 – Valida Transação (Token / Estabelecimento / Data)
2 – Valida statusTransacao = 2 (RECEBE)
3 – Tipo transação
Enviar Token
AUC-TokenAuth
Amazon
DynamoDB
Após "confirmado", envia Transação
Enviado com sucesso, altera status para 4 - finalizado
Amazon SPC
SalvaTransacaoMultiProdutoPS
Dá/Para evento Tarifário
CPqD Billing
ExecuteCreditControl
Lançamento CBILL
ABASTECE
DRIVE-THRU
REST
StatusTransacao: 0 – pendente / 1 – autorizado / 2 – recebido / 3 – confirmado /4 - finalizado / 88 - cancelado / 99 - erro)


On-Prem Net
Express Route
VPN Client
S2S VPN
P2S VPN
DDoS protection
Traffic Manager & front door
UDR
ExpressRoute VGW
VPN GW
Internet GW
Azure Region - A
Azure Region - B
Mgmt. Subnet
Jump Host
Shared service
AD, Proxy
DMZ and infra services
LB
NACL
FW
NAT
Global Vnet Peering
Pro-App-VNet
Non-Prod-App-VNet
Peering
Public Subnet
Private Subnet
Dashboard
Views
Application Insight
Monitoring Solutions
Metrics Explorer
IAM
Event Hubs
Power BI
Log Analytics
Ingest & Export APIs
Autoscale
Copyright @ Muhammad Marakkoottathil


VPC
Security group
Porta 22
SSH
Public Key
Servidor Linux
Private Key
VPC
Security group
Porta 3389
Remote Desktop
Public Key
Servidor Windows
Private Key

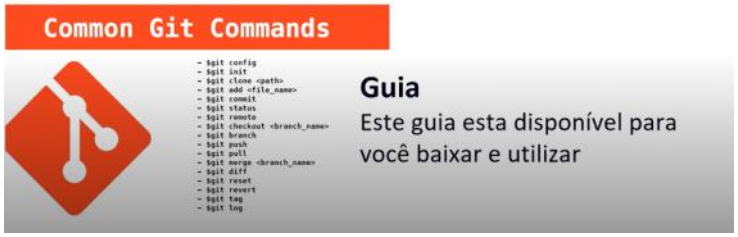




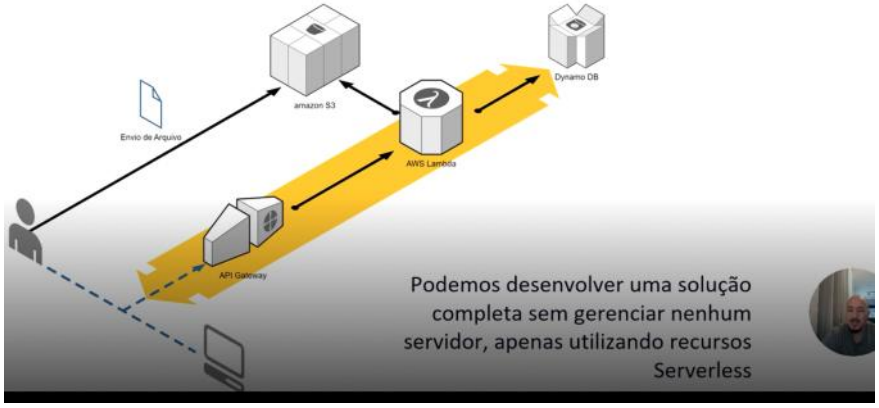
Desafio Amazon S3

Git Bash

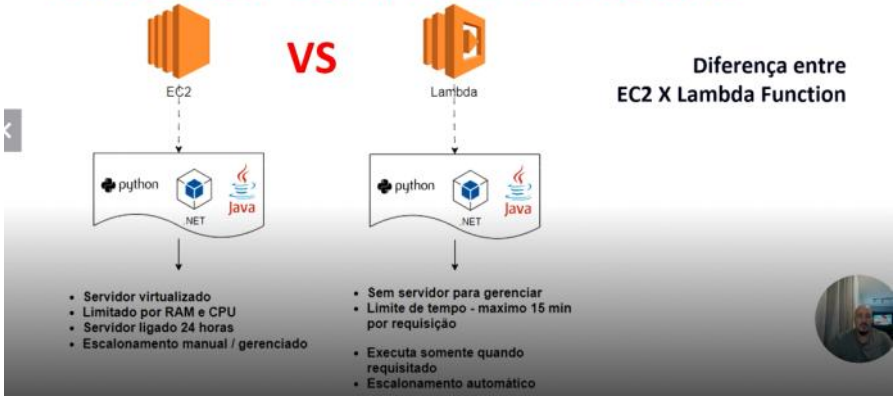
- Entender e estudar os comandos básicos do GIT
- Ver a imagem e procurar entender os comandos basicos



Amazon Lambda Function



Amazon Lambda Function




```
1 - {
2     "key1": "Desafio AWS 01",
3     "key2": "Desafio AWS 02",
4     "key3": "Desafio AWS 03"
5   }
```

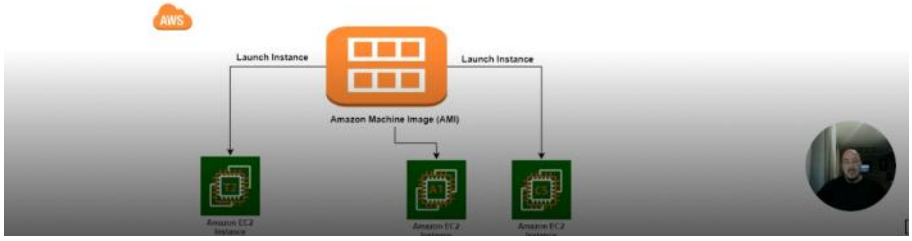
Conteúdo Programático

❑ Módulo 2: Computação na Nuvem com EC2

- ➢ Amazon EC2
- ➢ Armazenamento na Nuvem
- ➢ Gerenciamento de instâncias EC2

Criação e uso de imagens AMI

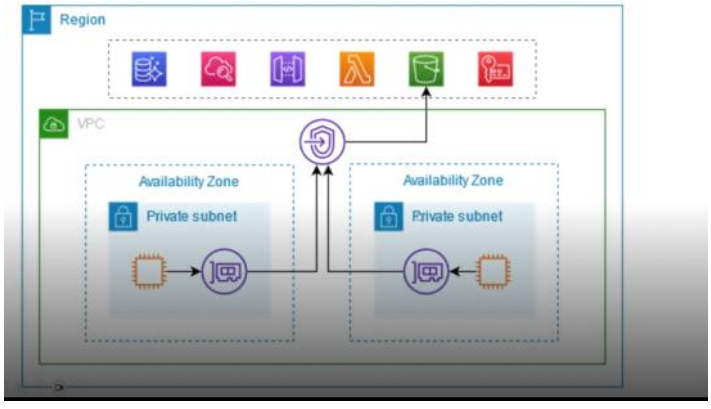
Entender as AMIs é crucial para gerenciar e implantar instâncias no EC2 de forma eficiente. Elas fornecem uma base para a criação de ambientes consistentes e reproduzíveis na nuvem.



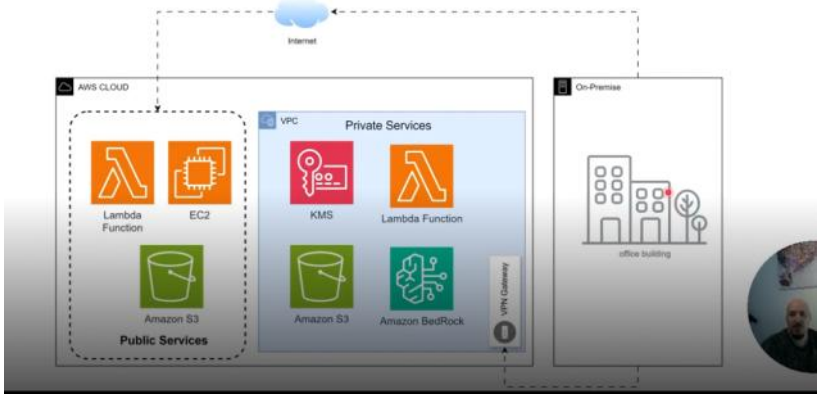
Conteúdo Programático

- **Módulo 4: Redes na AWS**
  - **Amazon VPC**
  - Amazon SubNet
  - Amazon Security Group
  - Amazon Route 53
  - Amazon Cloudfront
  - Amazon Elastic Load Balancer (EBL)

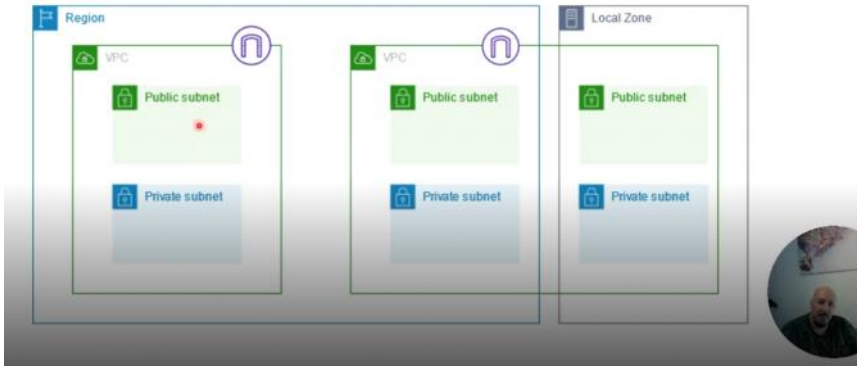
Amazon VPC



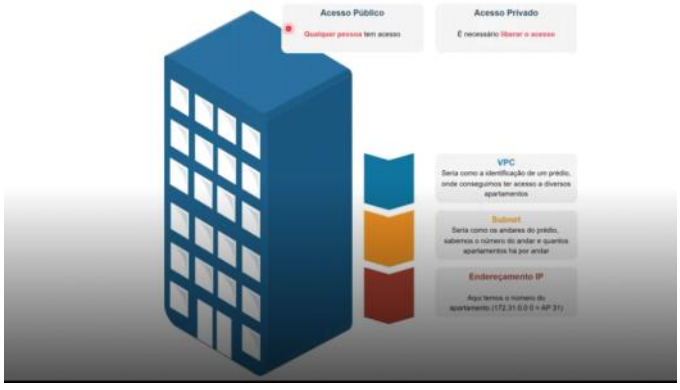
Amazon VPC



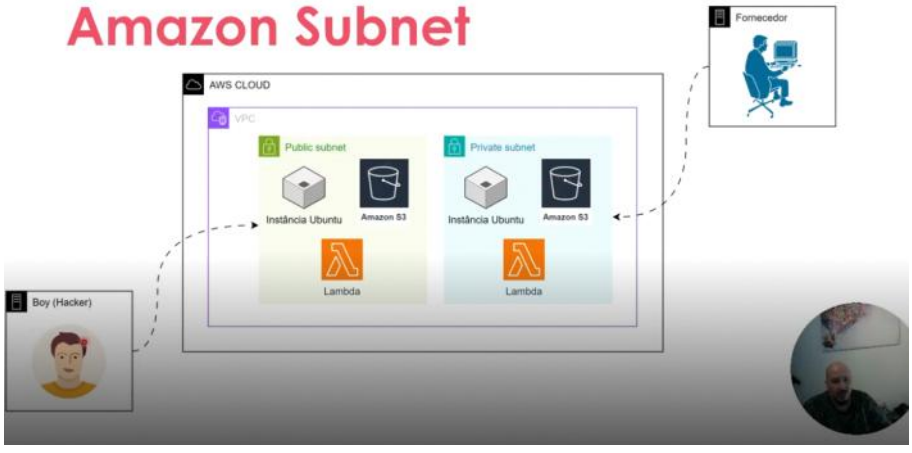
Amazon Subnet



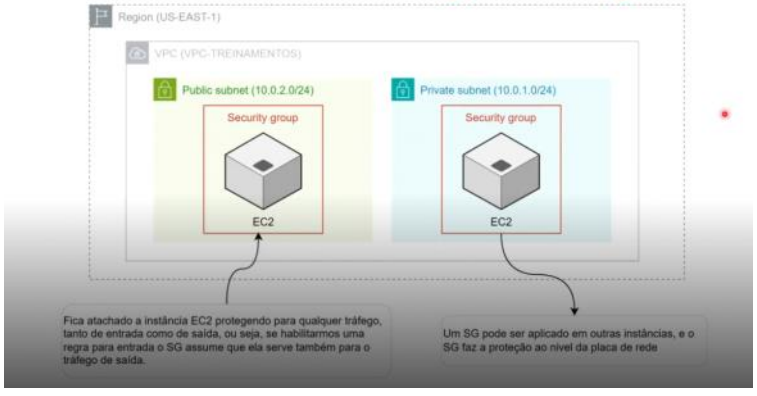
Amazon Subnet



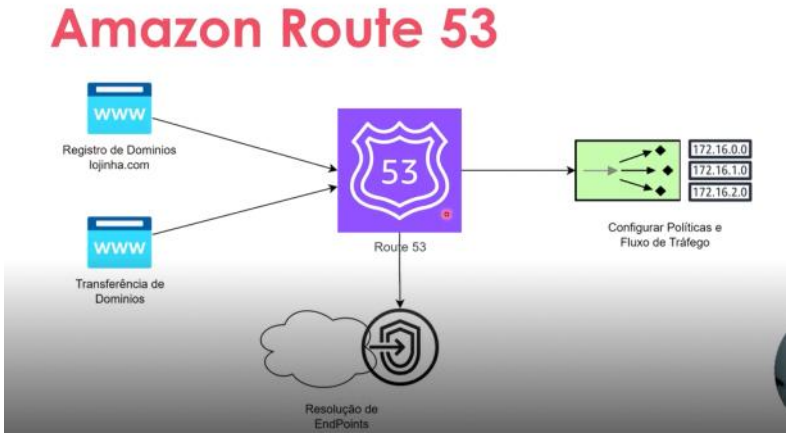
Amazon Subnet

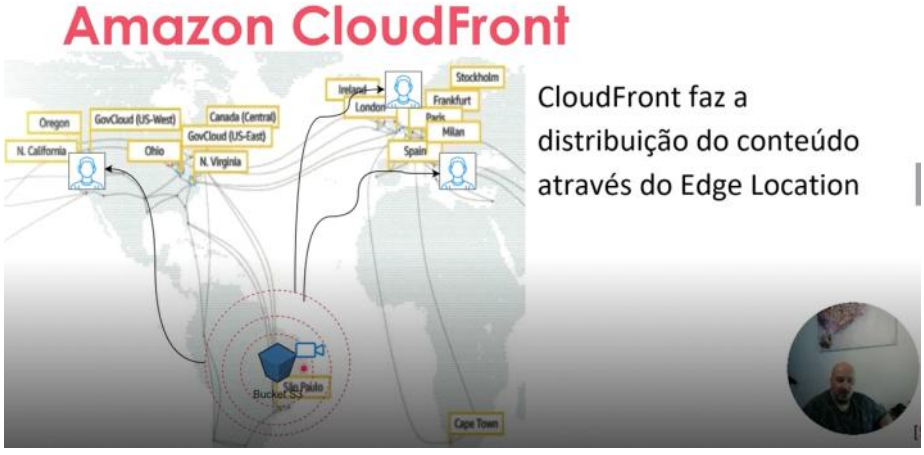


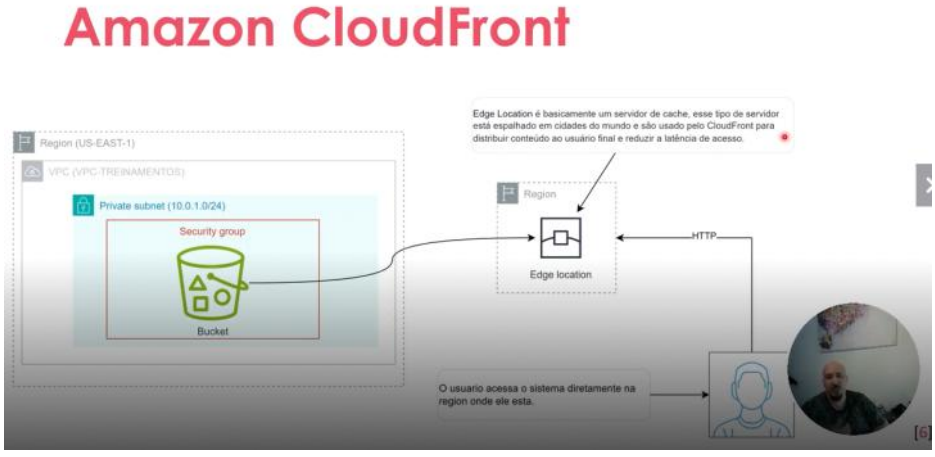
Amazon Security Group



Amazon Route 53
www
www.meuapp.com
53
Route 53
172.16.0.0
172.16.1.0
172.16.2.0
Endereço IP
A1

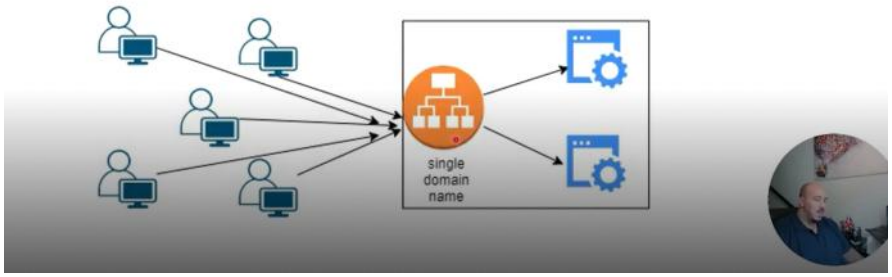

Amazon Route 53
www
Registro de Dominios
lojinha.com
www
Transferência de
Dominios
53
Route 53
172.16.0.0
172.16.1.0
172.16.2.0
Configurar Políticas e
Fluxo de Tráfego
Resolução de
EndPoints


Amazon CloudFront
Oregon
GovCloud (US-West)
Canada (Central)
GovCloud (US-East)
N. California
Ohio
N. Virginia
Ireland
London
Stockholm
Frankfurt
Paris
Milan
Spain
Bucket S3
São Paulo
Cape Town
CloudFront faz a
distribuição do conteúdo
através do Edge Location


Amazon CloudFront
Region (US-EAST-1)
VPC (VPC-TREINAMENTOS)
Private subnet (10.0.1.0/24)
Security group
Bucket
Edge Location é basicamente um servidor de cache, esse tipo de servidor está espalhado em cidades do mundo e são usado pelo CloudFront para distribuir conteúdo ao usuário final e reduzir a latência de acesso.
Region
Edge location
HTTP
O usuario acessa o sistema diretamente na region onde ele esta.

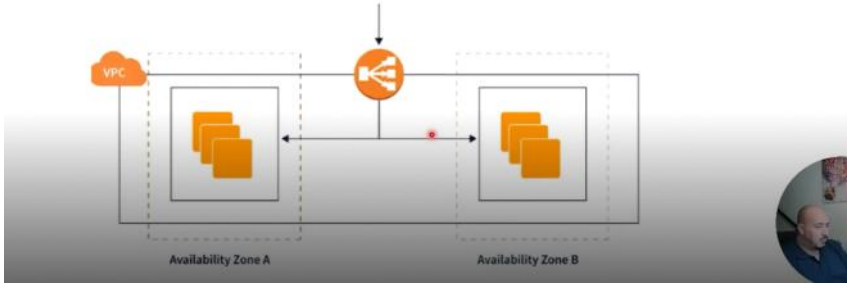
Amazon Elastic Load Balancer

Onde aplicar e em quais cenários podemos utilizar:



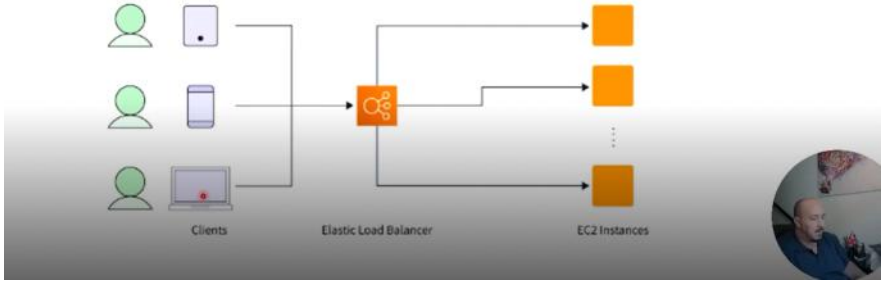
Amazon Elastic Load Balancer

Onde aplicar e em quais cenários podemos utilizar:

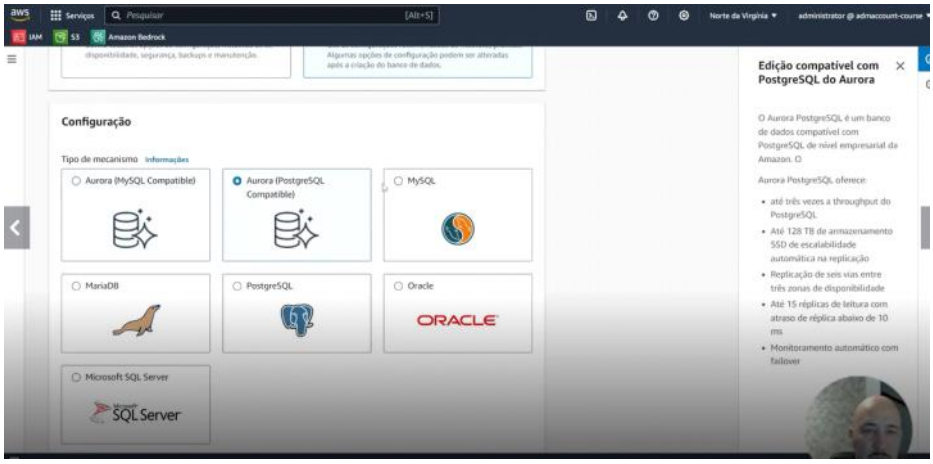


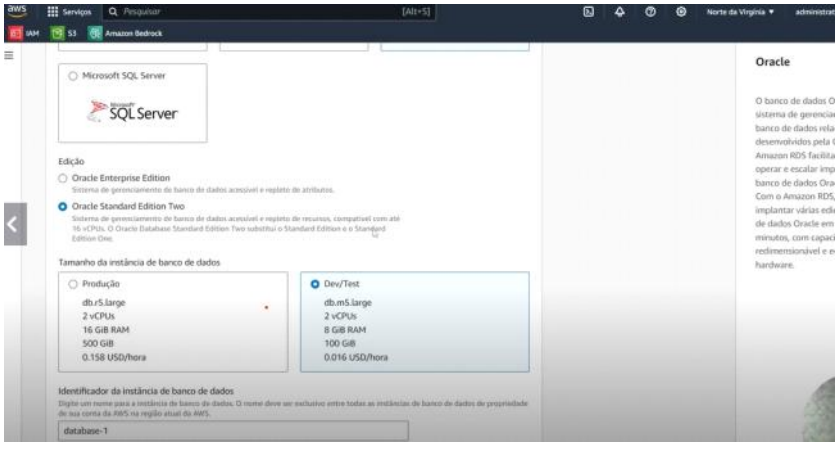
Amazon Elastic Load Balancer

Onde aplicar e em quais cenários podemos utilizar:



Amazon RDS





Amazon DynamoDB

1.Airbnb: utiliza para armazenar e processar dados de reservas e propriedades.

2.Lyft: usa para gerenciar dados de usuários e viagens em tempo real.

3.Netflix: usa para armazenar e dados de clientes e históricos de visualização rapidamente.

4.Samsung: Usa DynamoDB para diversas aplicações internas e externas.

5.Dropbox: Implementa DynamoDB para gerenciar metadados e operações de arquivos.

Amazon DynamoDB

Hands-On

1.Instalar AWS CLI
Instalar de acordo com a versão do seu SO

2.Criar o usuário programático
Na AWS ir em IAM e criar um user e baixar csv com suas chaves

3.Configurar o AWS CLI
Preencher com as chaves as informações (aws configure)

customer-items.json

```
{
    "Customers": [
        {
            "PutRequest": {
                "Item": {
                    "CustomerId": {"N": "110"},
                    "Name": {"S": "Aula de AWS"},
                    "City": {"S": "Brazil"},
                    "State": {"S": "SP"},
                    "Car": {"S": "RAM"}
                }
            }
        }
    ]
}
```

customers-table.json

```
{
    "TableName": "Customers",
    "KeySchema": [
        {
            "AttributeName": "CustomerId",
            "KeyType": "HASH"
        }, {
            "AttributeName": "Name",
            "KeyType": "RANGE"
        }
    ],
    "AttributeDefinitions": [
        {
            "AttributeName": "CustomerId",
            "AttributeType": "N"
        }, {
            "AttributeName": "Name",
            "AttributeType": "S"
        }
```

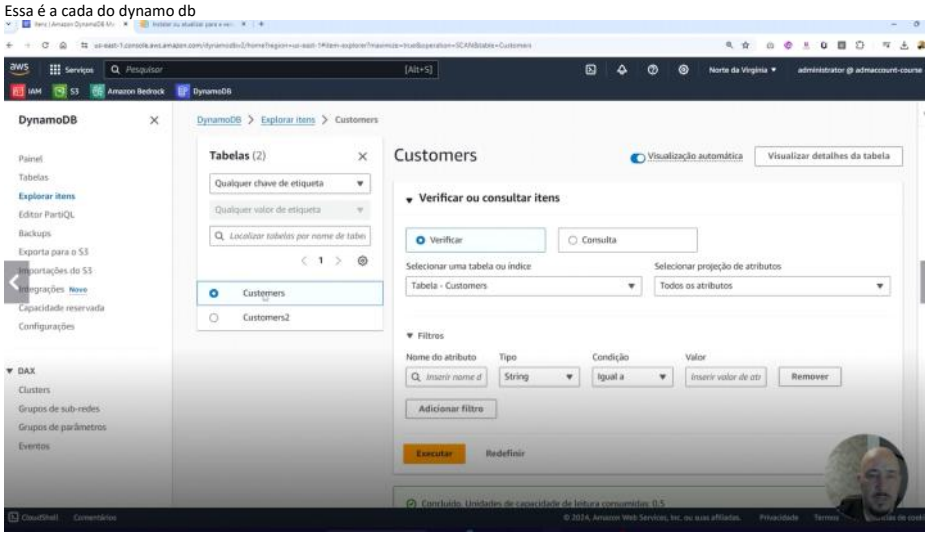
customers-table.json

```
{
    "TableName": "Customers2",
    "KeySchema": [
        {
            "AttributeName": "CustomerId",
            "KeyType": "HASH"
        }, {
            "AttributeName": "Name",
            "KeyType": "RANGE"
        }
    ],
    "AttributeDefinitions": [
        {
            "AttributeName": "CustomerId",
```

customers-table.json

```
    "KeySchema": [
        }, {
            "KeyType": "RANGE"
        }
    ],
    "AttributeDefinitions": [
        {
            "AttributeName": "CustomerId",
            "AttributeType": "N"
        }, {
            "AttributeName": "Name",
            "AttributeType": "S"
        }
    ],
    "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 5
    }
}
```

Essa é a cada do dynamo db



Estratégias de backup e recuperação de dados

Backup de Dados

Primeiramente temos que entender o que vem a ser Backup de Dados?
Um backup de dados é uma cópia dos dados do sistema, da configuração ou de aplicativos que é armazenada separadamente do original.

As vezes por conta de imprevistos como desastres naturais, erros humanos, eventos de segurança ou falhas no sistema.
O backup de dados é uma função essencial de proteção de dados para diminuir o risco de perda total ou parcial de dados no caso de eventos inesperados.

Estratégias de backup e recuperação de dados

Por que o backup de dados é importante?

O backup de dados é crucial porque sistemas podem falhar, e falhas no sistema, embora raras, são possíveis. O backup envolve copiar dados organizacionais para restauração em caso de falhas, incluindo um plano de recuperação de desastres. Concluindo, é essencial para prevenir perda de dados, garantir a continuidade dos negócios, reduzir custos, cumprir contratos, manter históricos de versões e atender a requisitos de conformidade e auditoria.

Estratégias de backup e recuperação de dados

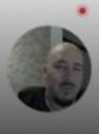
Definindo a estratégia

Uma estratégia de backup de dados envolve várias etapas para garantir a segurança, disponibilidade e recuperação dos dados.

1. Avaliação e Planejamento

Identificar dados críticos: determine quais dados são críticos e precisam ser protegidos.

Definição de RPO e RTO: Estabeleça os Objetivos de Ponto de Recuperação (RPO) e Objetivos de Tempo de Recuperação (RTO) para cada conjunto de dados.

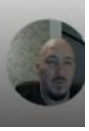


Estratégias de backup e recuperação de dados

O RPO tem a ver com dados e arquivos e a margem que **pode ser perdida** entre os ciclos de backup realizados diariamente, semanalmente, mensalmente.
O RTO é ligado ao **limite de tempo que o sistema** como um todo, de forma mais abrangente, **pode ficar parado** antes que tudo volte a funcionar adequadamente.

2. Seleção de Serviços AWS

Podemos utilizar o Amazon S3 para armazenar backups.
Temos também o AWS Backup para gerenciar e automatizar os backups dos nossos serviços na AWS



Estratégias de backup e recuperação de dados

3. Implementação da Estratégia de Backup

Backups Regulares: configurar backups automáticos diários para dados críticos e backups incrementais para otimizar o uso de armazenamento e tempo.

Cópias de Segurança em Múltiplas Regiões: Para maior resiliência, armazene cópias de segurança em várias regiões AWS usando a replicação entre regiões do S3.

Automação e Monitoramento: Use AWS Lambda e AWS CloudWatch para automatizar backups e monitorar a integridade e o status dos backups.

Estratégias de backup e recuperação de dados

Amazon RDS Automated Backups e Snapshots: Configure backups automatizados para bancos de dados RDS e utilize snapshots para backups manuais e restauração pontual.

Amazon DynamoDB On-Demand Backup: Utilize backups sob demanda e contínuos (PITR) para tabelas do DynamoDB.

Estratégias de backup e recuperação de dados

4. Recuperação e Teste

Documente e implemente planos detalhados para diferentes cenários de falha, faça testes regulares. Realize testes de recuperação de backups regularmente para garantir que os dados possam ser restaurados rapidamente e de forma eficaz.

Backup Drill: Conduza exercícios de simulação de desastres para validar a eficácia dos planos de recuperação.

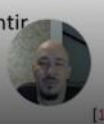
Estratégias de backup e recuperação de dados

5. Segurança e Conformidade

Criptografia de Dados: Use criptografia em trânsito (TLS) e em repouso (S3 Server-Side Encryption, RDS Encryption) para proteger dados sensíveis.

Controle de Acesso: Configure políticas do IAM para restringir o acesso aos backups apenas para usuários autorizados.

Registros e Auditorias: Utilize AWS CloudTrail para registrar atividades e garantir conformidade com requisitos regulatórios.

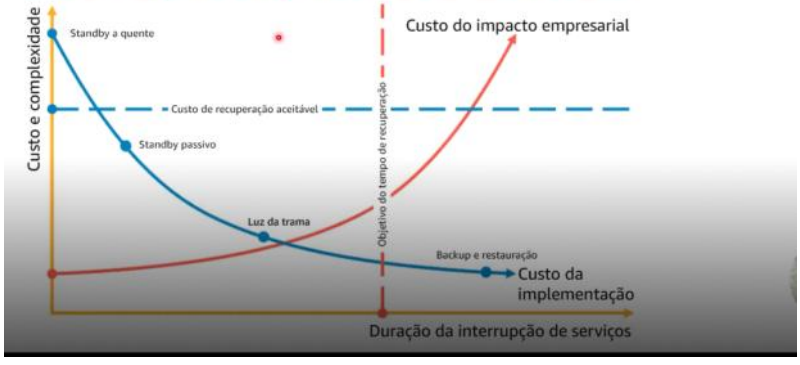


Estratégias de backup e recuperação de dados

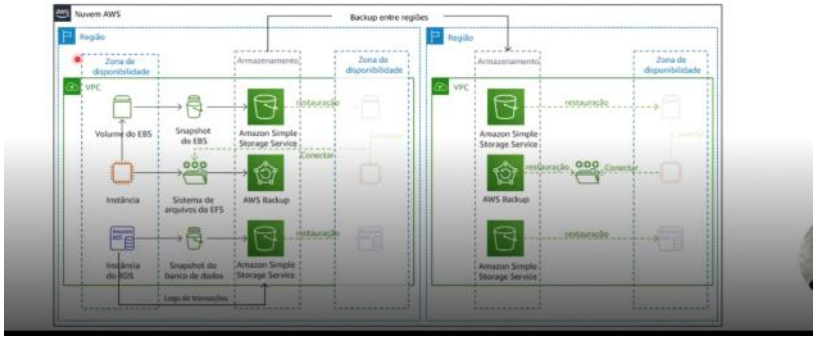
6. Custo e Otimização

Gestão de Custos: Utilize as ferramentas de análise de custo da AWS, como o AWS Cost Explorer, para monitorar e otimizar os gastos com backup.
Gerencie o Ciclo de vida através de políticas movendo os dados entre diferentes classes de armazenamento. Afine o seu custo até caber em seu bolso!

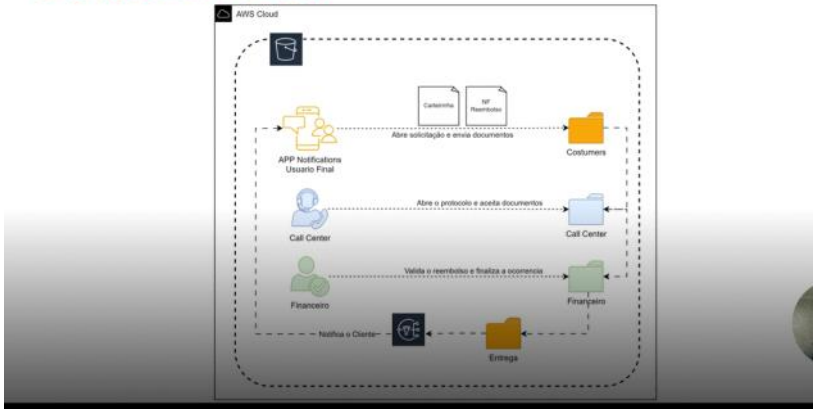
Estratégias de backup e recuperação de dados



Estratégias de backup e recuperação de dados



Amazon S3



Amazon S3

Política de Bucket para Tornar o Bucket Público:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::example-bucket/*"
        }
    ]
}
```

Amazon S3

Version: "2012-10-17": Esta é a versão da linguagem de políticas da AWS. O valor "2012-10-17" é a data de lançamento da versão

Statement: Uma política pode conter múltiplas declarações (statements). Cada declaração define um conjunto de permissões.

Sid: "PublicReadGetObject": O Sid (Statement ID) é um identificador opcional para a declaração. É útil para identificar e gerenciar políticas, especialmente quando há várias declarações.

Effect: "Allow": O efeito desta declaração é "Permitir" (Allow). Pode ser "Allow" para permitir acesso ou "Deny" para negar acesso.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::example-bucket/*"
        }
    ]
}
```

Principal: "*": O Principal especifica a entidade à qual a política se aplica. O "*" indica que esta política se aplica a todos os usuários, ou seja, é uma política pública.

Action: "s3:GetObject": A Action especifica a ação que está sendo permitida ou negada. Neste caso, s3:GetObject permite a ação de ler ou baixar objetos do bucket.

Resource: "arn:aws:s3:::example-bucket/*": O Resource especifica o recurso ao qual a política se aplica. O valor "arn:aws:s3:::example-bucket/*" indica que a política se aplica a todos os objetos dentro do bucket chamado example-bucket.

arn:aws:s3::: é o prefixo do ARN (Amazon Resource Name) para buckets S3.

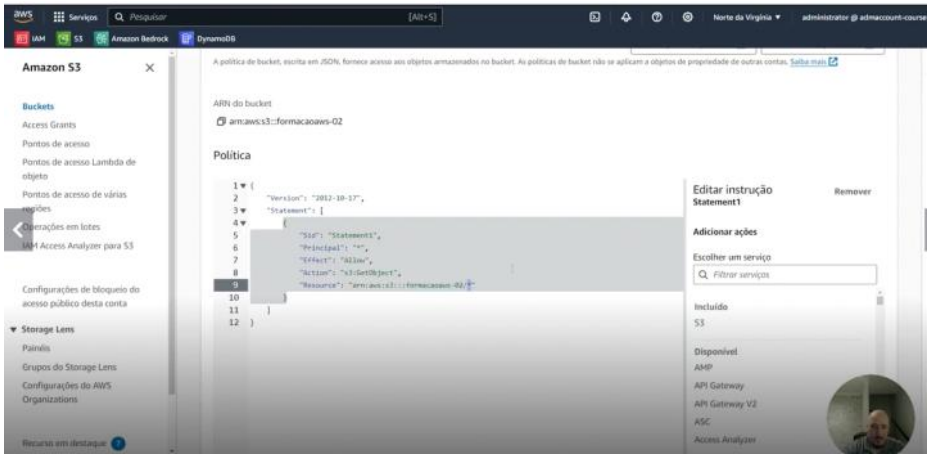
example-bucket é o nome do bucket.

O *** após o nome do bucket** indica que a política se aplica a todos os objetos dentro do bucket.

```
---------- Acesso total ao S3 e EC2
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "ec2:*"
            ],
            "Resource": "*"
        }
    ]
}

---------- Acesso de leitura ao S3 e EC2

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

```
---------- Acesso ao bucket 1* e 2* mas não acessa o 3*
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-1",
                "arn:aws:s3:::bucket-1/*",
                "arn:aws:s3:::bucket-2",
                "arn:aws:s3:::bucket-2/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-2",
                "arn:aws:s3:::bucket-3/*"
            ]
        }
    ]
}
```

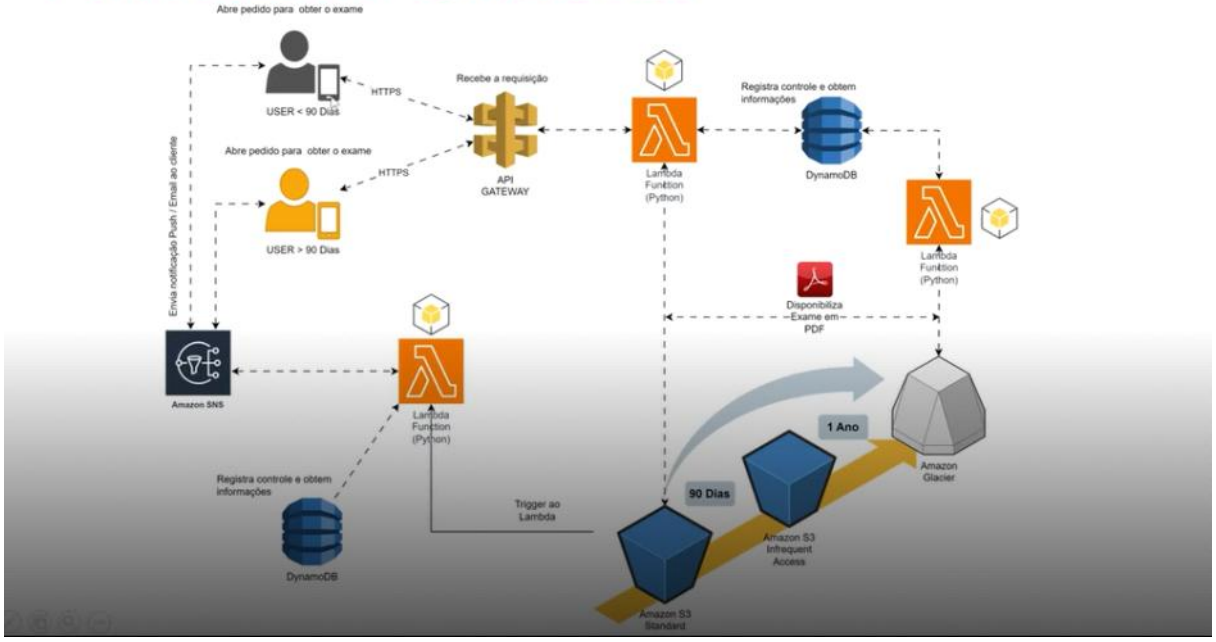
```
---------- Sobe arquivo mas não apaga no bucket

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::nome-do-bucket/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::nome-do-bucket/*"
            ]
        }
    ]
}
```



Amazon Glacier

Em conjunto com o gerenciamento do ciclo de vida de armazenamento, rotacionando dados raramente usados para o armazenamento frio para economizar dinheiro.

Amazon Glacier



Amazon Glacier

Quando usar:

Quando os clientes obtêm acesso aos dados em um período de 3 a 5 horas com a opção de recuperações padrão.

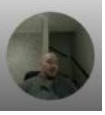
O preço é de US$ 0,01/GB para recuperações padrão e, para 1.000 solicitações, é de US$ 0,05.

Amazon Glacier

Quando usar:

90 dias é a duração mínima do armazenamento no caso do S3 Glacier.

180 dias é a duração mínima do armazenamento no caso do *S3 Glacier Deep Archive* e a estrutura de preços é semelhante à do S3 Glacier.



Amazon Glacier

Transferindo grandes volumes com família AWS Snow

AWS SnowBall

AWS SnowBall Edge

AWS Snowmobile

Quando necessitamos processar grande volume dados (petabytes) usamos os serviços de Snow Family.
https://aws.amazon.com/pt/snow/

Amazon Glacier

Transferindo grandes volumes com família AWS Snow

AWS SnowBall

Ele é um dispositivo de computação que tem como principal função a migração de dados e também o armazenamento de borda que oferece dois tipos de funções.
O AWS Snowball é muito utilizado em casos de armazenamento local e também quando é necessário transferir dados em larga escala.

Amazon Glacier

Transferindo grandes volumes com família AWS Snow

AWS SnowBall Edge

O Snowball Edge pode transportar dados em velocidades mais rápidas do que a internet. Esse transporte é feito enviando os dados nos dispositivos por meio de uma operadora regional. Os dispositivos são resistentes, complementados com etiquetas de envio E Ink.

Amazon Glacier

Transferindo grandes volumes com família AWS Snow

AWS Snowmobile

O Snowmobile é um “upgrade” de outro serviço de transporte de dados da Amazon chamado Snowball, que permite o envio de até 80 TB de dados por meios convencionais. Os clientes preenchem o espaço disponível, enviam para a empresa, que então sobe para a nuvem.
Capaz de levar até 100 petabytes em sua carroceria.

Amazon CloudFront

O AWS CloudFront é uma solução de CDN que faz cache dos assets armazenando cópias dos recursos estáticos (como imagens, vídeos, arquivos CSS, JavaScript, etc.) em locais estratégicos (pontos de presença, ou POPs) próximos aos usuários finais.

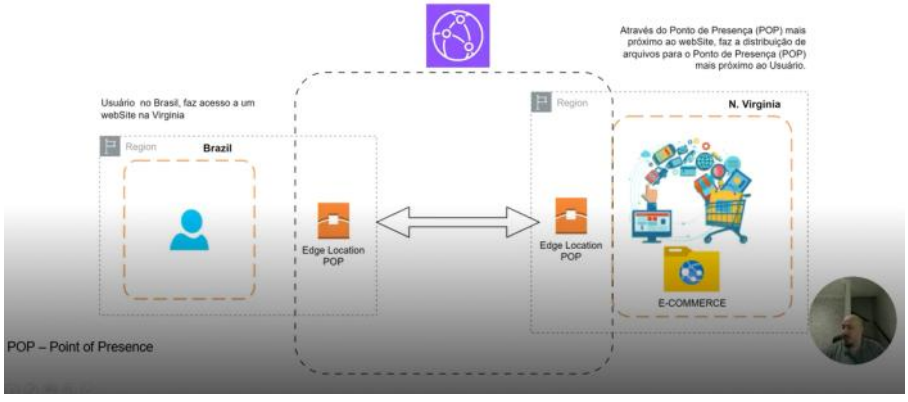
Amazon CloudFront

Isso é feito para que, em vez de buscar esses recursos diretamente do servidor de origem toda vez que são solicitados, eles possam ser entregues rapidamente a partir do cache mais próximo. Isso reduz a latência e melhora o desempenho do carregamento da página ou entrega de conteúdo.

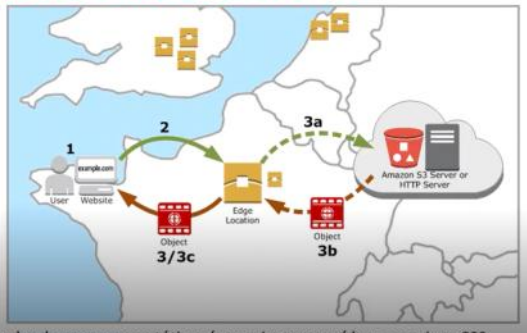
Amazon CloudFront

Exemplo 1: um site de comércio eletrônico que utiliza o CloudFront para distribuir imagens de produtos.
Quando um usuário acessa o site, as imagens são entregues dc POP mais próximo, garantindo carregamento rápido.
Se uma nova imagem é adicionada, ela é carregada diretamente do servidor de origem na primeira vez e, em seguida, cacheada para futuras solicitações.

Amazon CloudFront



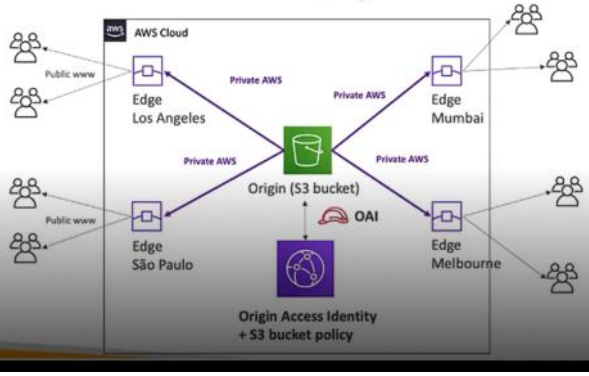
Amazon CloudFront



O AWS CloudFront faz cache dos recursos estáticos (como imagens, vídeos, arquivos CSS, JavaScript, etc.) em locais estratégicos (pontos de presença, ou POPs) próximos aos usuários finais.

Amazon CloudFront

CloudFront – S3 as an Origin



Conteúdo Programático

- ❑ **Módulo 7: Serviços Intermediários e Avançados**
  - ➢ **Amazon Lambda**
  - ➢ Amazon ECS e EKS
  - ➢ Amazon SNS e SQS
  - ➢ AWS Step Functions

AWS Lambda

O AWS Lambda é um serviço de computação que executa seu código **em resposta a eventos e gerencia automaticamente os recursos de computação**, tornando-se a maneira mais rápida de transformar uma ideia em aplicações de produção modernas e com tecnologia sem servidor.

AWS Lambda

O Lambda executa seu código em uma infraestrutura de

AWS Lambda

O Lambda executa seu código em uma infraestrutura de computação de alta disponibilidade e executa toda a administração dos recursos computacionais, incluindo manutenção do servidor e do sistema operacional, provisionamento e escalabilidade automática da capacidade e registro em log do código.

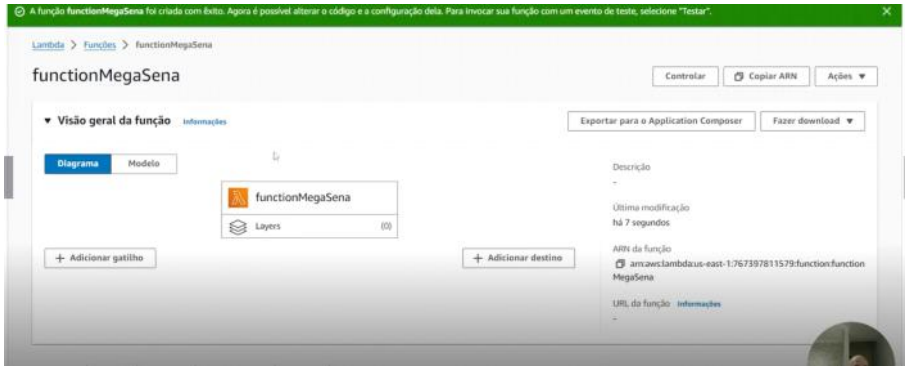
AWS Lambda

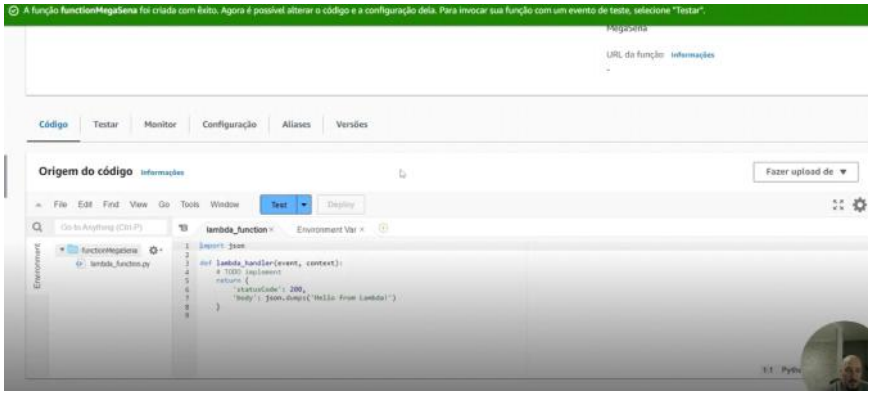
Serverless: sem servidor é a arquitetura nativa da nuvem que permite transferir mais das suas responsabilidades operacionais à AWS, aumentando a agilidade e a inovação.
Custo: 0,20 USD por 1 milhão de solicitações

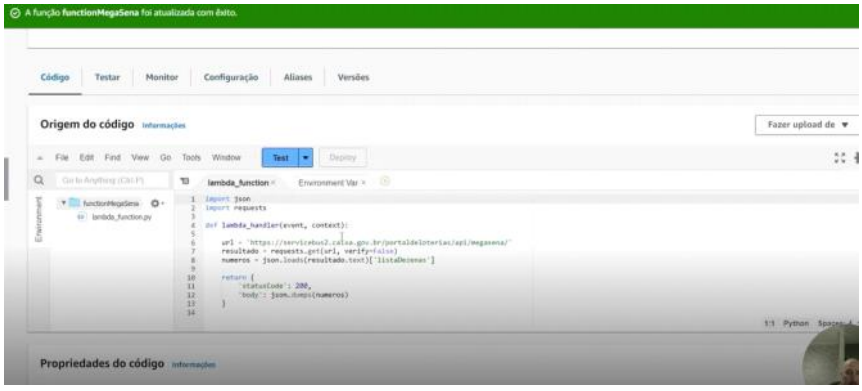
AWS Lambda

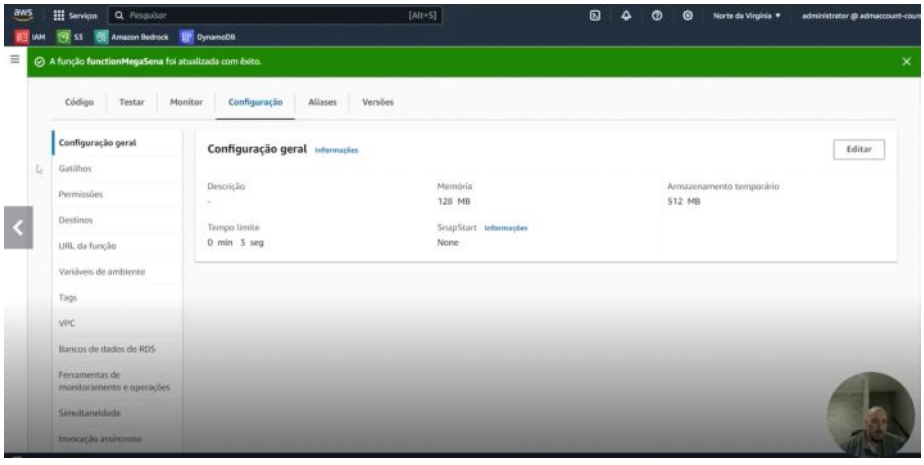
AWS Lambda oferece vários benefícios, incluindo:
Execução sob demanda: o Lambda executa seu código somente quando necessário e é cobrado apenas pelo tempo de computação usado.
Escalonamento automático: o Lambda é dimensionado automaticamente com base no número de eventos recebidos, eliminando a necessidade de gerenciamento manual do servidor.

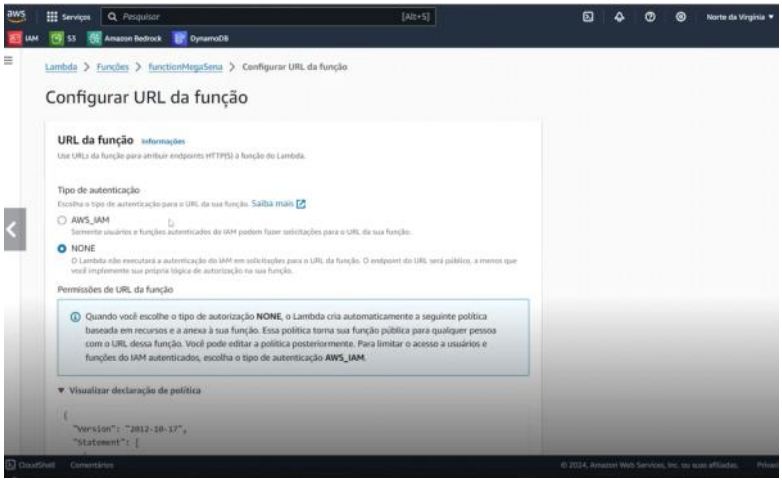
AWS Lambda

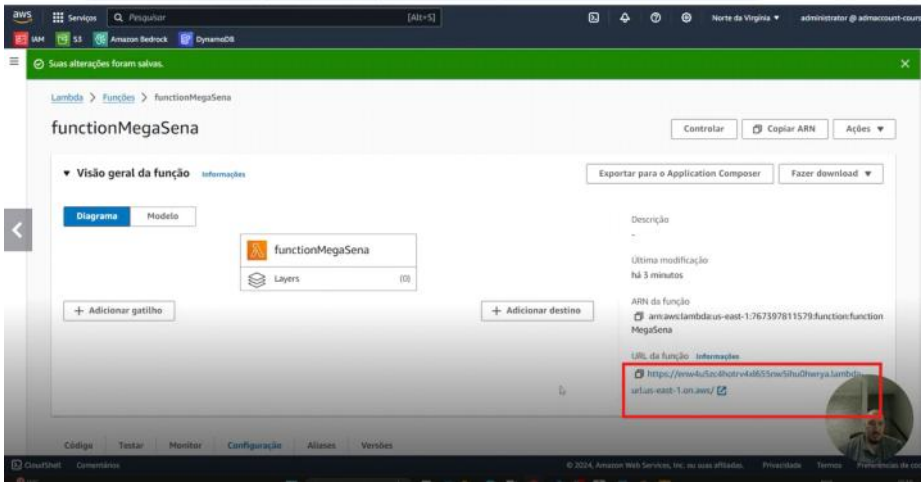



A função functionMegaSena foi criada com êxito. Agora é possível alterar o código e a configuração dela. Para invocar sua função com um evento de teste, selecione "Testar".
URL da função Informações
Código
Testar
Monitor
Configuração
Aliases
Versões
Origem do código Informações
Fazer upload de
File Edit Find View Go Tools Window
Test
Deploy
lambda_function
Environment Var
import json
def lambda_handler(event, context):
# TODO implement
return {
'statusCode': 200,
'body': json.dumps('Hello from Lambda!')
}


A função functionMegaSena foi atualizada com êxito.
Código
Testar
Monitor
Configuração
Aliases
Versões
Origem do código Informações
Fazer upload de
File Edit Find View Go Tools Window
Test
Deploy
lambda_function
Environment Var
import json
import requests
def lambda_handler(event, context):
url = 'https://servicebus2.caixa.gov.br/portaldeloterias/api/megasena/'
resultado = requests.get(url, verify=False)
numeros = json.loads(resultado.text)['listaDezenas']
return {
'statusCode': 200,
'body': json.dumps(numeros)
}
Propriedades do código Informações


Serviços
Pesquisar
[Alt+S]
Norte da Virgínia
IAM
S3
Amazon Bedrock
DynamoDB
A função functionMegaSena foi atualizada com êxito.
Código
Testar
Monitor
Configuração
Aliases
Versões
Configuração geral
Gatilhos
Permissões
Destinos
URL da função
Variáveis de ambiente
Tags
VPC
Bancos de dados do RDS
Ferramentas de monitoramento e operações
Simultaneidade
Invocação assíncrona
Configuração geral Informações
Editar
Descrição
-
Memória
128 MB
Armazenamento temporário
512 MB
Tempo limite
0 min 3 seg
SnapStart Informações
None


Serviços
Pesquisar
[Alt+S]
Norte da Virgínia
IAM
S3
Amazon Bedrock
DynamoDB
Lambda > Funções > functionMegaSena > Configurar URL da função
Configurar URL da função
URL da função Informações
Use URLs da função para atribuir endpoints HTTP(S) à função do Lambda.
Tipo de autenticação
Escolha o tipo de autenticação para o URL da sua função. Saiba mais
AWS_IAM
Somente usuários e funções autenticados do IAM podem fazer solicitações para o URL da sua função.
NONE
O Lambda não executará a autenticação do IAM em solicitações para o URL da função. O endpoint do URL será público, a menos que você implemente sua própria lógica de autorização na sua função.
Permissões de URL da função
Quando você escolhe o tipo de autorização NONE, o Lambda cria automaticamente a seguinte política baseada em recursos e a anexa à sua função. Essa política torna sua função pública para qualquer pessoa com o URL dessa função. Você pode editar a política posteriormente. Para limitar o acesso a usuários e funções do IAM autenticados, escolha o tipo de autenticação AWS_IAM.
Visualizar declaração de política
{
"Version": "2012-10-17",
"Statement": [
CloudShell
Comentários
© 2024, Amazon Web Services, Inc. ou suas afiliadas.

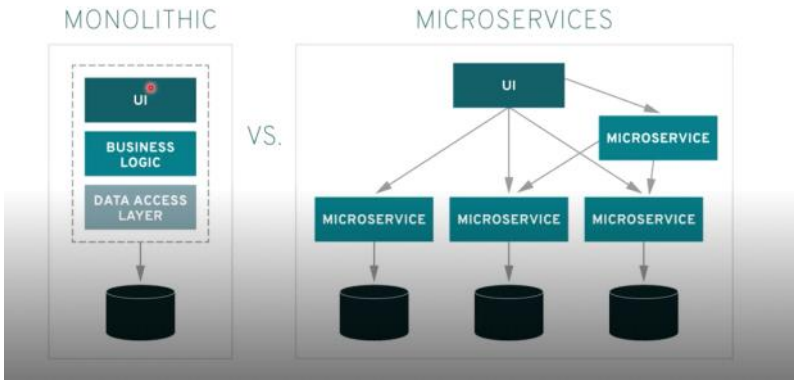


AWS ECS e EKS

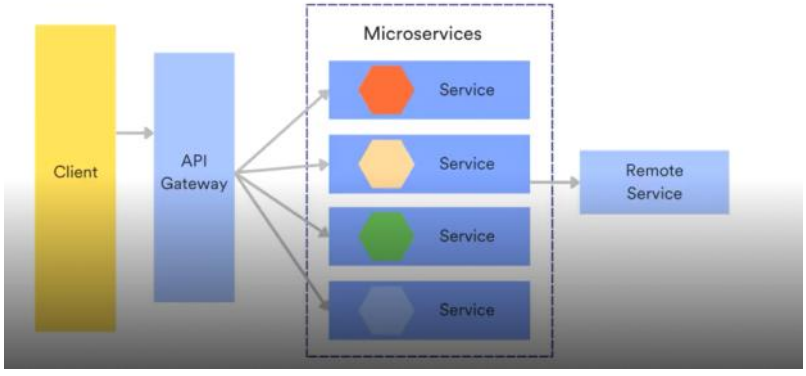
Microservices

Antes de tudo, falar em serviços como ECS e EKS que são serviços de containers e trabalham com microserviços, temos que entender o que vem a ser microserviços!

AWS ECS e EKS

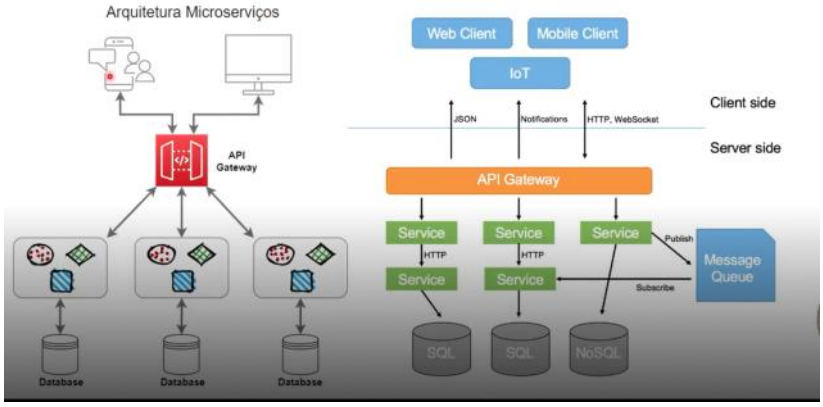


AWS ECS e EKS



AWS ECS e EKS

Arquitetura Microserviços

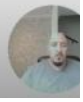


AWS ECS e EKS

ECS - Amazon Elastic Container Service

O ECS é um serviço gerenciado de orquestração de contêineres que permite executar, interromper e gerenciar facilmente contêineres em um cluster.

O ECS permite executar aplicações em contêineres em uma arquitetura de microservices, utilizando a escalabilidade, segurança e o desempenho da infraestrutura da AWS.



AWS ECS e EKS

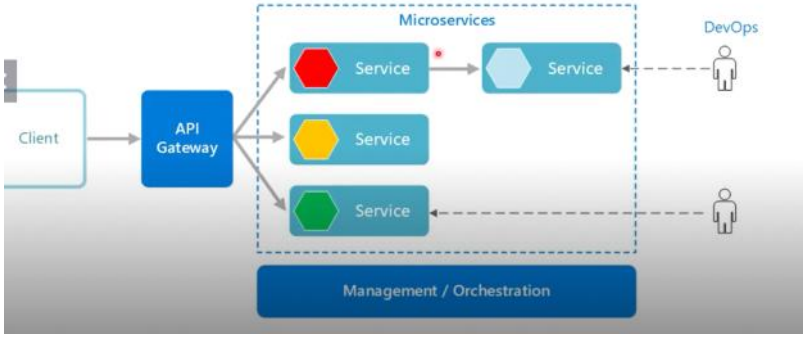
ECS - Amazon Elastic Container Service

As **principais características** do ECS são:

- Segurança: podemos aplicar políticas de segurança para controlar o acesso aos contêineres.

AWS ECS e EKS

ECS - Amazon Elastic Container Service



AWS ECS e EKS

ECS - Amazon Elastic Container Service

Microserviços em Contêineres: cada componente do aplicativo (catálogo, carrinho, pagamentos, etc.) é empacotado em um contêiner Docker.
Definição de Tarefas e Serviços no ECS: definimos uma tarefa para cada contêiner e configuramos os serviços no ECS para manter a disponibilidade desejada para cada contêiner.

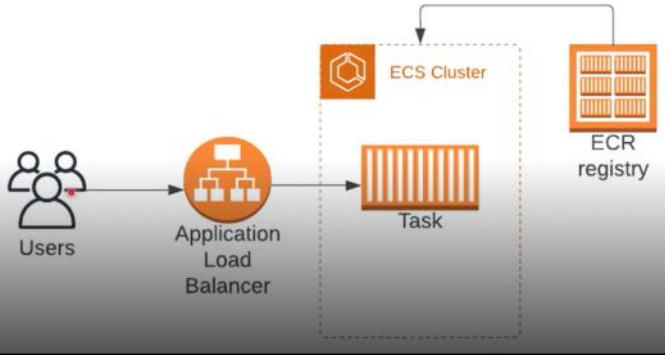
AWS ECS e EKS

ECS - Amazon Elastic Container Service

Escalabilidade Automática: configuramos as políticas de escalabilidade automática para aumentar ou diminuir o número de contêineres com base na carga de trabalho (por exemplo, aumentar contêineres durante promoções).
Monitoramento e Logging: utilizamos o Amazon CloudWatch para monitorar a performance e logs dos contêineres para troubleshooting e análise de desempenho.

AWS ECS e EKS

ECS - Amazon Elastic Container Service

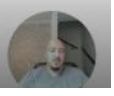


AWS ECS e EKS

ECS - Amazon Elastic Container Service

ECS: serviço de orquestração de contêineres que ajuda a gerenciar contêineres do Docker em um cluster.

Elastic Container Registry (ECR): serviço de registro de contêiner gerenciado para armazenar, gerenciar e implantar imagens do Docker.



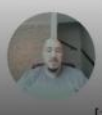
AWS ECS e EKS

ECS - Amazon Elastic Container Service

Porquê usar ECS?

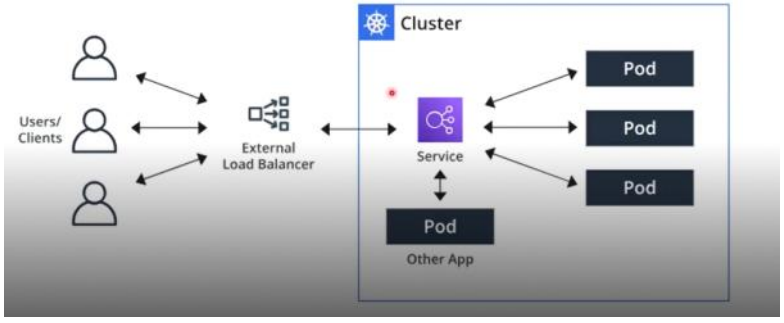
O Amazon ECS é ideal para tarefas em execução por mais de 15 minutos ou se precisar executar código fora das regiões da AWS.

O Amazon ECS oferece experiências opinativas sobre coisas como redes e observabilidade, mas pode ser personalizado de acordo com suas necessidades.



AWS ECS e EKS

EKS - Elastic Kubernetes Service (EKS)



AWS ECS e EKS

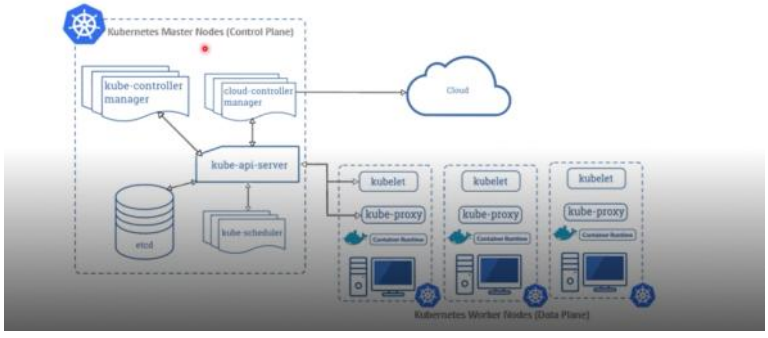
EKS - Elastic Kubernetes Service (EKS)

O AWS EKS é mais adequado para organizações que já utilizam o Kubernetes ou **estão considerando** implementar essa tecnologia para aproveitar a força da infraestrutura da AWS e, ao mesmo tempo, reduzir os esforços vinculados à manutenção do ambiente Kubernetes.

Pois a gestão de um cluster Kubernetes é muito complexa.

AWS ECS e EKS

EKS - Elastic Kubernetes Service (EKS)



AWS ECS e EKS

Amazon ECS VS Amazon EKS

Vamos agora ver a diferença entre os dois serviços.

À medida que as arquiteturas baseadas em micro serviços se tornaram mais populares, houve um aumento significativo nos workloads e serviços de contêiner em nuvem.
Por isto escolher entre um serviço e outro depende muito do tipo de serviço e arquitetura escolhida.

AWS ECS e EKS

Amazon ECS VS Amazon EKS

Elastic Container Service (ECS), serviço proprietário de orquestração de contêineres gerenciados da AWS para execução de aplicativos em contêineres em escala.

No back-end usa a tecnologia sem servidor (Fargate), que elimina a carga de gerenciamento.

AWS ECS e EKS

Amazon ECS VS Amazon EKS

Conclusão:

O EKS é voltado para **casos de uso corporativo**, atendendo a aplicativos complexos baseados em microsserviços distribuídos.
Embora ambos os serviços ofereçam flexibilidade em suas cargas de trabalho em contêineres, a melhor opção depende do caso de uso-alvo da sua organização, da maturidade em contêineres, do tamanho da equipe e da importância da facilidade de uso.

[27]

AWS SNS e SQS

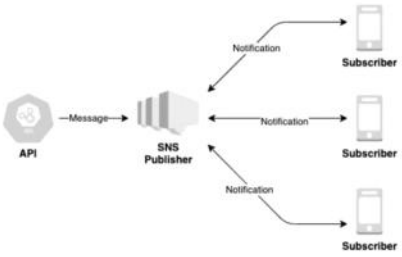
Amazon SNS (Simple Notification Service)

Serviço de mensagens assíncronas, disponível e seguro para notificações de mensagens entre aplicativos distribuídos e microserviços.
Ele permite o envio de notificações push para dispositivos móveis, mensagens de texto, e-mails ou integrações com outros serviços da AWS, como o Amazon SQS ou o AWS Lambda.

AWS SNS e SQS

Amazon SNS

O SNS é dividido em tópicos e assinaturas, na imagem temos uma visão macro com 3 subscribers no mesmo Pubisher (publicador)



AWS SNS e SQS

Amazon SNS

Tópicos: são pontos de acesso entre o Publisher (publicador) e o Subscriber (inscrito).

Um tópico pode ser dividido em 2 tipos.
Tipo Fifo ou tipo Padrão e isto é configurado no Subscriber que é onde recebe a mensagem.

AWS SNS e SQS

Amazon SNS

Tipo Fifo (first in/first out), possui o limite de 300 publicações por segundo. Garantindo a entrega da primeira mensagem que entra, primeira mensagem que sai.

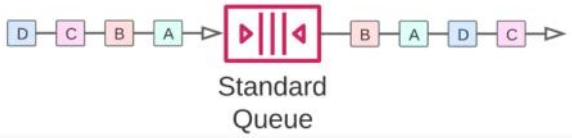
AWS SNS e SQS

Amazon SNS

Tipo Padrão, utilizado na maioria dos casos e é muito mais flexível e menos rigoroso como o Fifo.
Ele não visa ordenar por ordem de chegada, o throughput de publicações.
Desta forma ele não garante a ordem exata das mensagens podendo ser entregues mais de uma vez.

AWS SNS e SQS

Amazon SNS



AWS SNS e SQS

Amazon SQS (Simple Queue Service)

O SQS oferece suporte a várias filas, incluindo os tipos Padrão e FIFO (First-In-First-Out).

Trabalha com os consumidores e quando uma mensagem é recebida, ela se torna invisível para outros consumidores por um período de tempo limite especificado.

AWS SNS e SQS

Amazon SQS (Simple Queue Service)

O SQS suaviza o tráfego de rajadas e garante um processamento consistente.

AWS SNS e SQS

Amazon SQS X Amazon SNS

A escolha entre o Amazon SNS e o SQS exige que você tome uma decisão em relação ao gerenciamento das suas mensagens.

O Amazon SNS funciona como um sistema de transmissão, o que o torna ideal para alertar rapidamente os usuários de um determinado produto.

AWS SNS e SQS

Amazon SQS X Amazon SNS

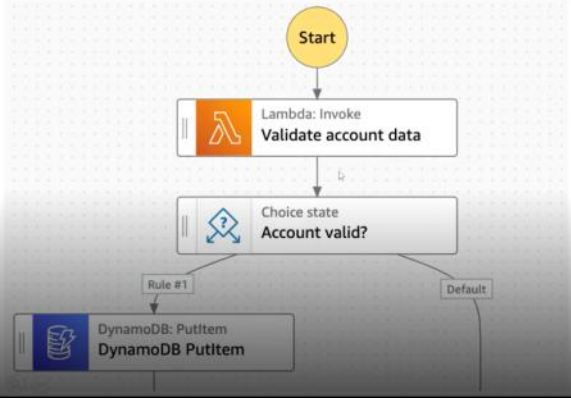
Para comunicação com clientes, de forma instantânea, o melhor é utilizar o Amazon SNS.

Para comunicação entre sistemas de diferentes áreas do seu aplicativo e de forma organizada, o melhor sera o Amazon SQS.

AWS Step Functions

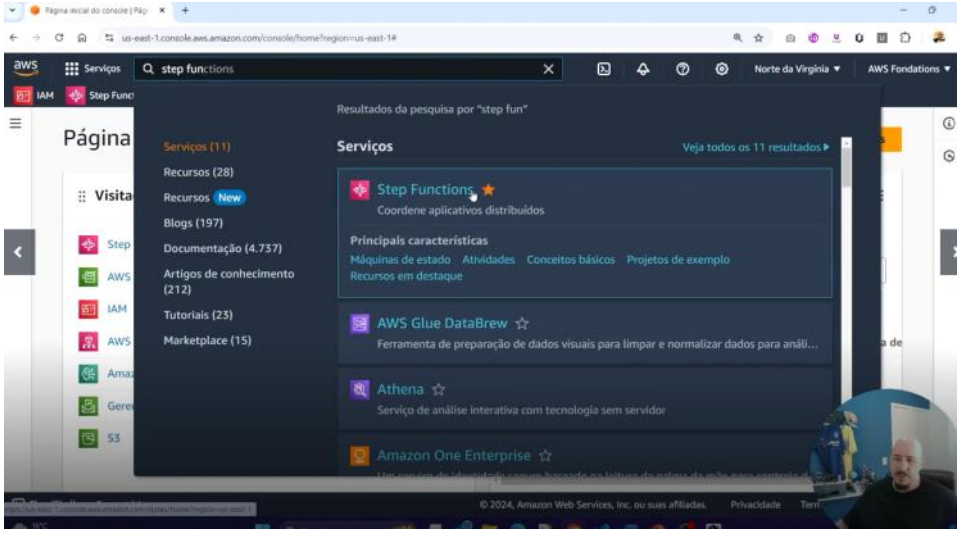
AWS Step Functions nada mais é do que um construtor visual para criar fluxos de trabalho.
Ele é um serviço que facilita a coordenação de aplicativos e microserviços, usando fluxos de trabalho visuais.

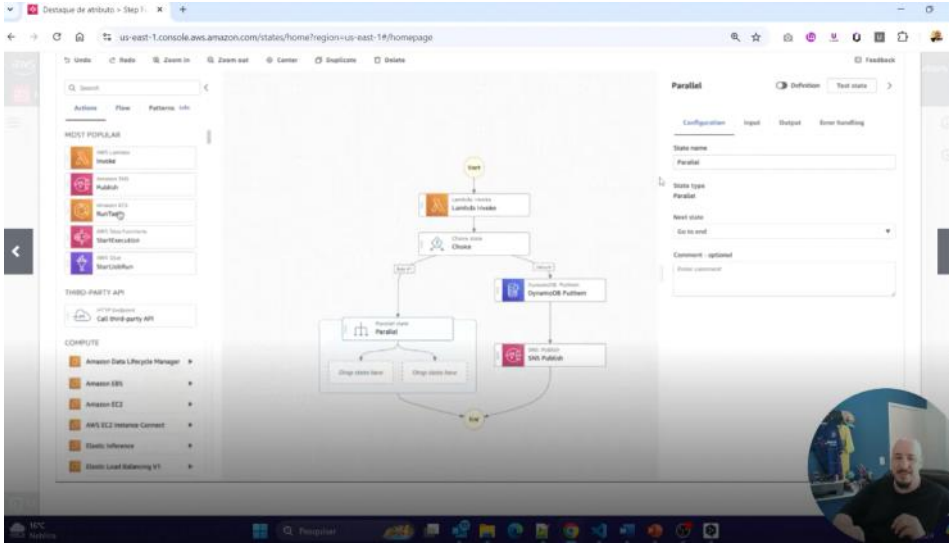
AWS Step Functions

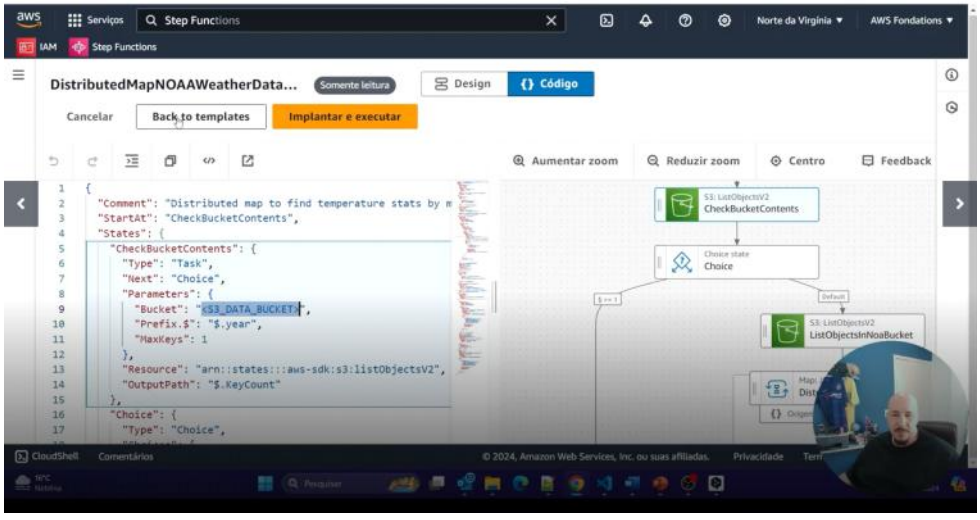


AWS Step Functions

Ao criar o fluxo de trabalho, não precisamos ter todos os recursos do AWS antecipadamente para começar.
Podemos criar o workflow e depois adicionar as definições aos recursos.


step functions
Resultados da pesquisa por "step fun"
Serviços (11)
Recursos (28)
Recursos New
Blogs (197)
Documentação (4.737)
Artigos de conhecimento (212)
Tutoriais (23)
Marketplace (15)
Serviços
Veja todos os 11 resultados
Step Functions
Coordene aplicativos distribuídos
Principais características
Máquinas de estado
Atividades
Conceitos básicos
Projetos de exemplo
Recursos em destaque
AWS Glue DataBrew
Ferramenta de preparação de dados visuais para limpar e normalizar dados para análi...
Athena
Serviço de análise interativa com tecnologia sem servidor
Amazon One Enterprise


Parallel


Step Functions
DistributedMapNOAAWeatherData...
Somente leitura
Design
Código
Cancelar
Back to templates
Implantar e executar
Aumentar zoom
Reduzir zoom
Centro
Feedback
"Comment": "Distributed map to find temperature stats by m
"StartAt": "CheckBucketContents",
"States": {
"CheckBucketContents": {
"Type": "Task",
"Next": "Choice",
"Parameters": {
"Bucket": "<S3_DATA_BUCKET>",
"Prefix.$": "$.year",
"MaxKeys": 1
},
"Resource": "arn::states:::aws-sdk:s3:listObjectsV2",
"OutputPath": "$.KeyCount"
},
"Choice": {
"Type": "Choice",
CheckBucketContents
Choice
ListObjectsInNoaBucket

AWS CloudWatch

Estas métricas são eventos que foram gerados e/ou variáveis que auxiliam na medição do serviço, possibilitando avaliar seus recursos e suas aplicações.

AWS CloudWatch

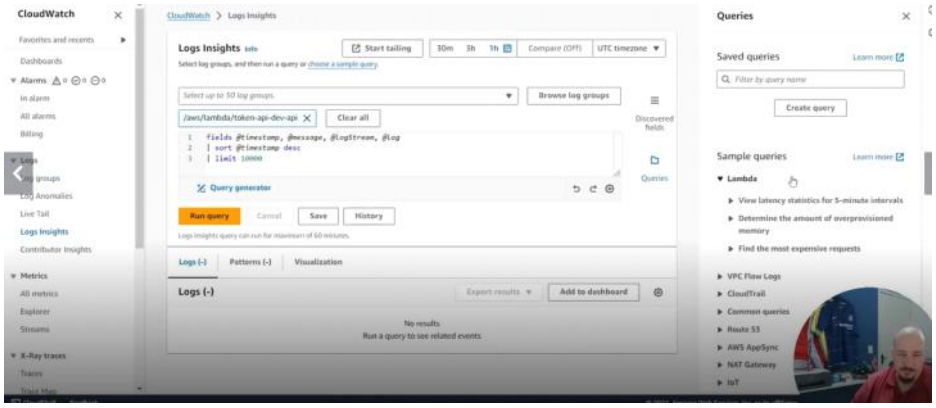
Caso de uso 1: é possível registrar as alterações no estado de uma instância do Amazon EC2 usando o CloudWatch Events com a ajuda da função AWS Lambda.

AWS CloudWatch

Caso de uso 2: você pode registrar as operações de API em nível de objeto nos seus buckets S3 usando o CloudWatch Events.
Mas antes devemos usar o AWS CloudTrail para definir uma trilha configurada para receber essas operações.

AWS CloudWatch

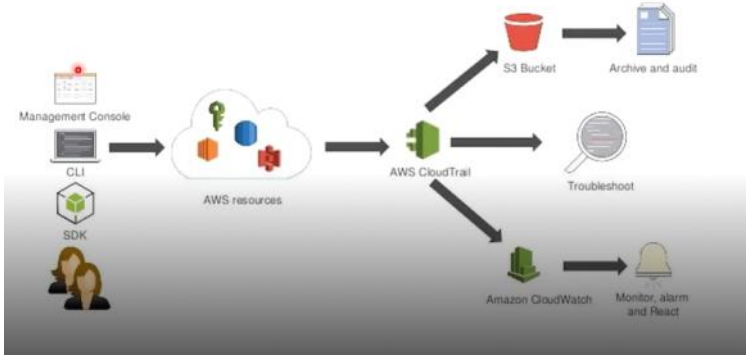
Com o CloudWatch, nós temos visibilidade no âmbito do sistema da utilização de recursos, da performance das aplicações e da integridade operacional.

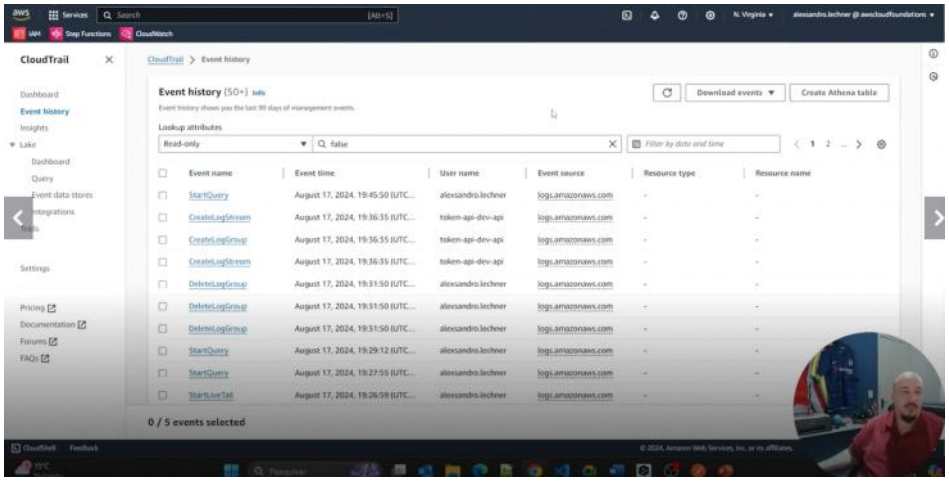


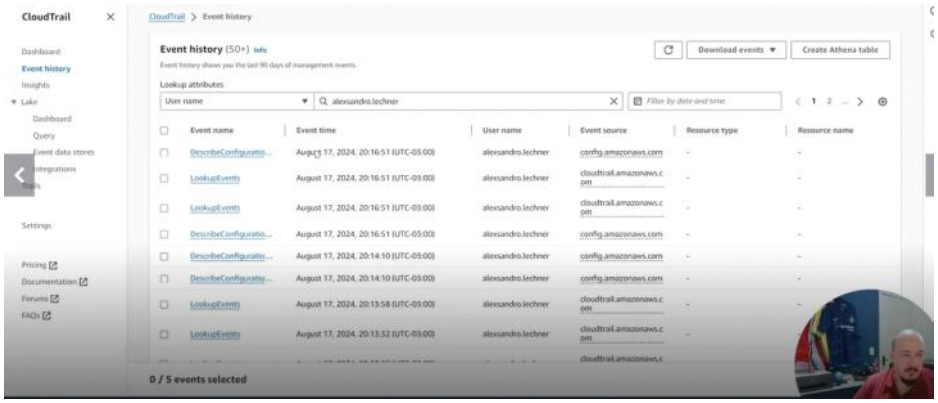
AWS CloudTrail

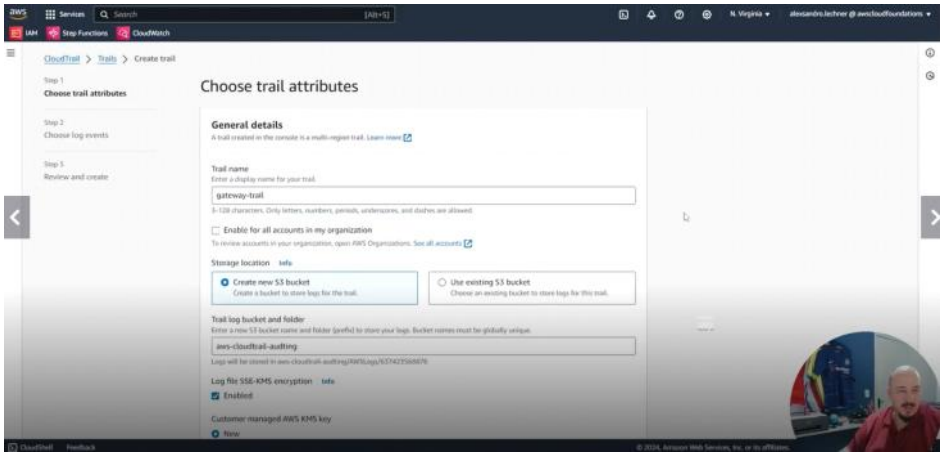
AWS CloudTrail é um Serviço da AWS que ajuda você na auditoria operacional e de risco, a governança e a conformidade da sua conta da AWS.
Toda ação realizada por um usuário, função ou serviço serão registradas como eventos em CloudTrail.
Os eventos incluem ações realizadas no Console AWS, Command Line Interface e APIs.

AWS CloudTrail









AWS CloudFormation

O AWS CloudFormation é um processo que auxilia na automação de criação de recurso na AWS por meio de templates JSON ou YAML.

Podemos utilizar os templates quantas vezes quisermos e pagamos apenas pelas stacks criadas (conjunto de recursos, ex: EC2, RDS, S3, etc).

AWS CloudFormation

Além de ser um processo automatizado, conseguimos versionar estes templates. Com ele podemos criar deste um recurso simples como um EC2 até um Arquitetura robusta com vários recursos.

AWS CloudFormation

```
"S3BackupBucket" : {
    "Type" : "AWS::S3::Bucket",
    "DeletionPolicy" : "Retain",
    "Properties" : {
      "AccessControl" : "Private",
      "BucketName" : "opentodo-backups",
      "LifecycleConfiguration" : {
        "Rules" : [ {
          "ExpirationInDays" : 15,
          "Status" : "Enabled"
        } ]
      }
    }
  },
  "S3LogBucket" : {
    "Type" : "AWS::S3::Bucket",
    "DeletionPolicy" : "Retain",
    "Properties" : {
      "AccessControl" : "Private",
      "BucketName" : "opentodo-logs",
      "LifecycleConfiguration" : {
        "Rules" : [ {
          "ExpirationInDays" : 30,
          "Status" : "Enabled"
        } ]
      }
    }
  }
```

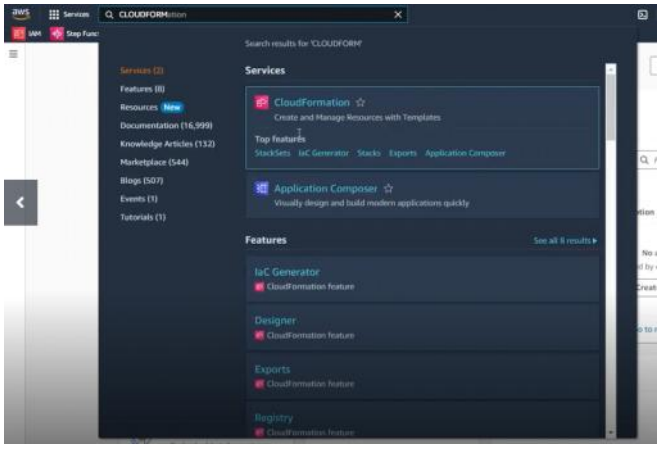
Template: arquivo JSON com informações sobre nossos recursos.

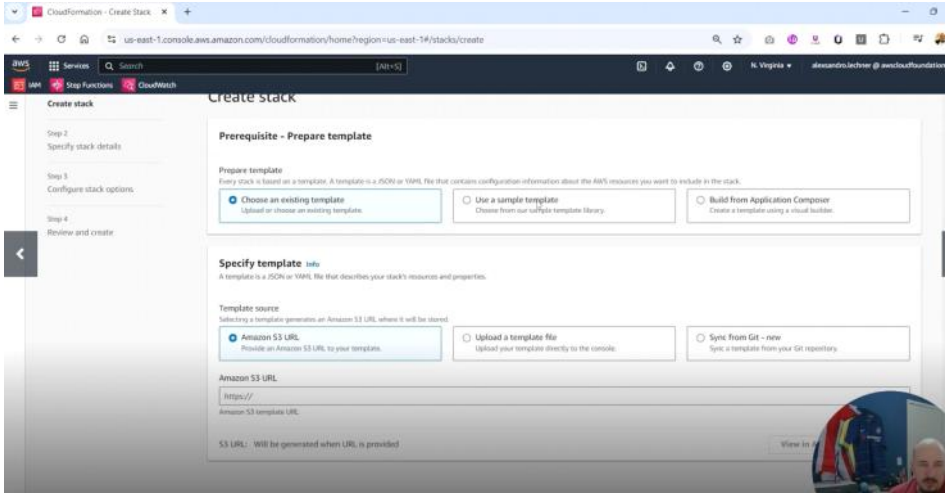
AWS::S3::Bucket: aqui definimos 2 buckets para registros e backups.

DeletionPolicy: aqui temos uma politica de que retem os dados do S3 buckets

AccessControl: definimos a modo de acesso, neste caso é private.

LifecycleConfiguration: permite criar uma politica de ciclo de vida, neste caso removerão os arquivos com mais de 15 ou 30 dias.





AWS Identity and Access Management (IAM)

O AWS Identity and Access Management (IAM) é um recurso importante na AWS e que nos ajuda na gestão de acessos a nossa plataforma.

Não somente isto, é um recurso fundamental para aplicarmos regras de acesso e também politicas de acesso.

```
alexs@ALECHNER MINGW64 /d/DIO/AWS/Módulo 8 Gerenciamento e Governança na AWS/Curso 04 - A
WS Identity and Access Management (IAM)
$ ls -l
total 203168
-rw-r--r-- 1 alexs 197609    176705 Aug 18 00:16 '04 - AWS_IAM.pptx'
-rw-r--r-- 1 alexs 197609 207824022 Mar 17 22:39 'IAM Automacao com usuarios e grupos.mkv'

-rw-r--r-- 1 alexs 197609     80095 Aug 17 11:54  Questionario.xlsx
-rw-r--r-- 1 alexs 197609        23 Aug 31 13:00  paradas.txt
-rwxr-xr-x 1 alexs 197609      1438 Mar 17 20:29  scriptIAM.ps1*
-rw-r--r-- 1 alexs 197609       281 Aug 31 13:45  usuarios.csv
-rw-r--r-- 1 alexs 197609       165 May 28 17:55 '~$01 - AmazonS3.pptx'
-rw-r--r-- 1 alexs 197609       162 Feb  1  2024 '~$01.docx'
-rw-r--r-- 1 alexs 197609       165 Mar 26 13:54 '~$02 - S3.pptx'
-rw-r--r-- 1 alexs 197609       165 Apr  1 22:48 '~$02 - Snapshots EBS.pptx'
-rw-r--r-- 1 alexs 197609       165 Aug 17 11:53 '~$Questionario.xlsx'
-rw-r--r-- 1 alexs 197609       165 Feb  3  2024 '~$conteudo01.pptx'

alexs@ALECHNER MINGW64 /d/DIO/AWS/Módulo 8 Gerenciamento e Governança na AWS/Curso 04 - A
WS Identity and Access Management (IAM)
$ pwd
/d/DIO/AWS/Módulo 8 Gerenciamento e Governança na AWS/Curso 04 - AWS Identity and Access Management (IAM)

alexs@ALECHNER MINGW64 /d/DIO/AWS/Módulo 8 Gerenciamento e Governança na AWS/Curso 04 - AWS Identity and Access Management (IAM)
$ bash scriptIAM.ps1 usuarios.csv
```

AWS Policies e Roles

O que vem a ser AWS Policies e onde aplicar?
Primeiramente nós temos que entender que Policies são POLÍTICAS e estas políticas são documentos que definem permissões para usuários, grupos e roles na AWS.

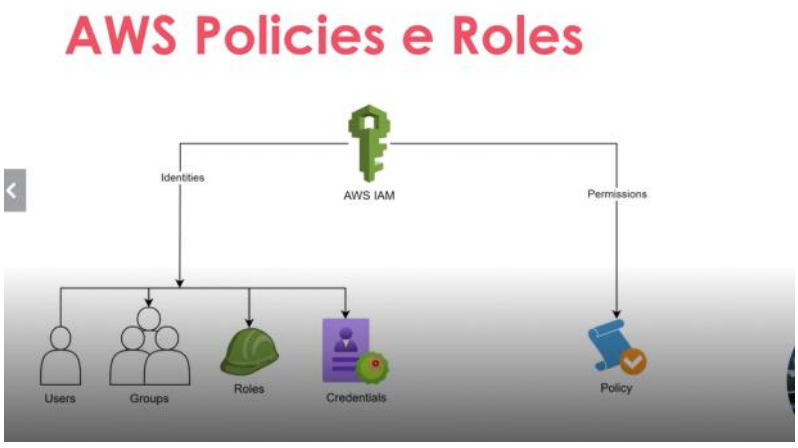
O formato, a especificação das politicas são escritas em JSON.

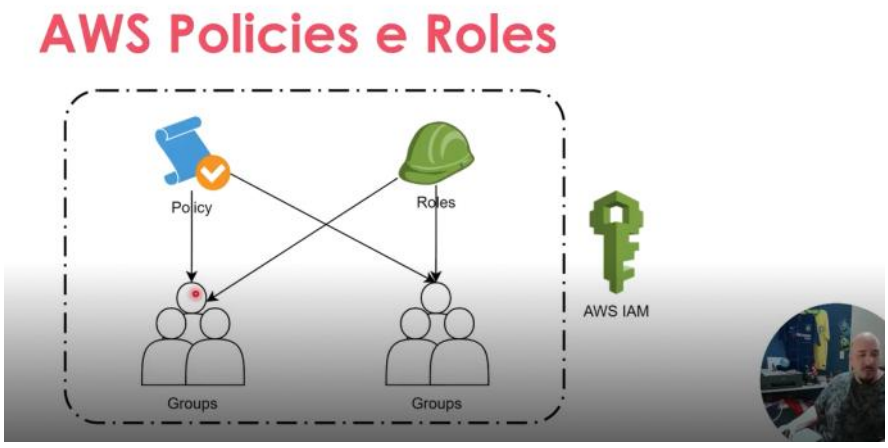
AWS Policies e Roles

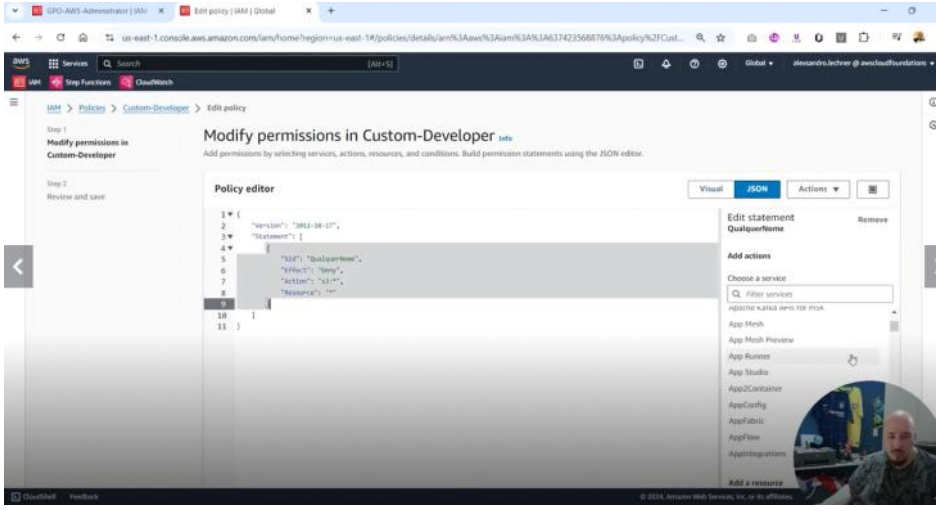
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": ["s3:*"],
            "Resource": ["*"]
        },
        {
            "Sid": "DenyCustomerBucket",
            "Action": ["s3:*"],
            "Effect": "Deny",
            "Resource": ["arn:aws:s3:::customer", "arn:aws:s3:::customer/*" ]
        }
    ]
}
```

AWS Policies e Roles

JSON é o documento que definimos as politicas de permissões
Identy Policy: é a politica de identidade anexada a um grupo, função ou usuário.


AWS Policies e Roles
Identities
AWS IAM
Permissions
Users
Groups
Roles
Credentials
Policy


AWS Policies e Roles
Policy
Roles
AWS IAM
Groups
Groups


Modify permissions in Custom-Developer
Policy editor
Visual
JSON
Actions
Edit statement
Add actions
Choose a service

Conteúdo Programático

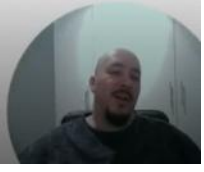
- **Módulo 9: Segurança na AWS**
  - **Práticas recomendadas de segurança na nuvem**
  - Criptografia de dados na AWS
  - AWS WAF: firewall de aplicativos da web

Práticas recomendadas

Não temos como falar de segurança na nuvem sem primeiro entender as responsabilidades dos envolvidos, neste caso:

VOCÊ vs AWS

Para isto teremos que analisar o "Modelo de Responsabilidade Compartilhada"

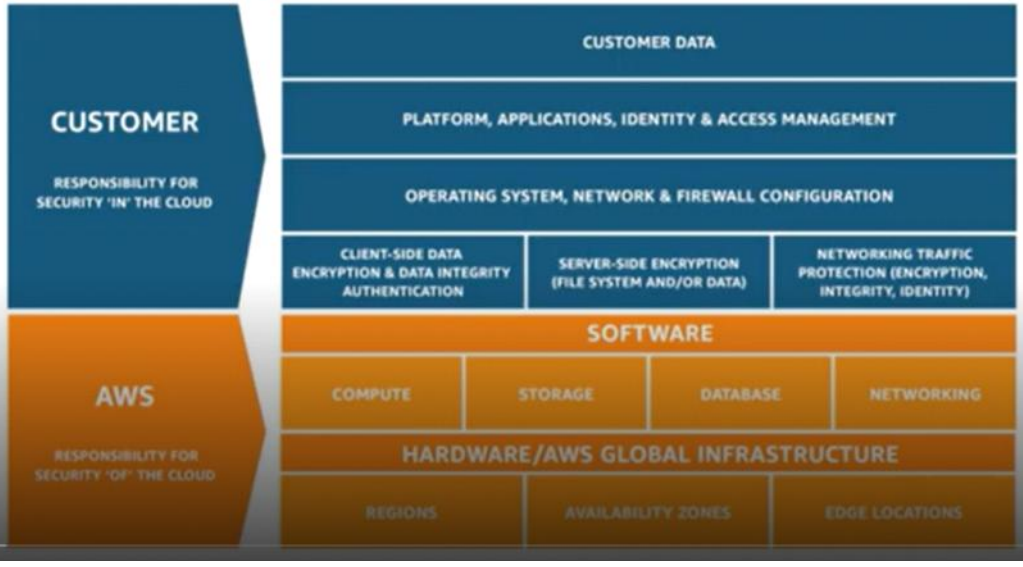


Práticas recomendadas

A AWS é responsável pela segurança da nuvem, o que inclui a proteção da infraestrutura global que dá suporte a todos os serviços em nuvem oferecidos. Isso abrange:

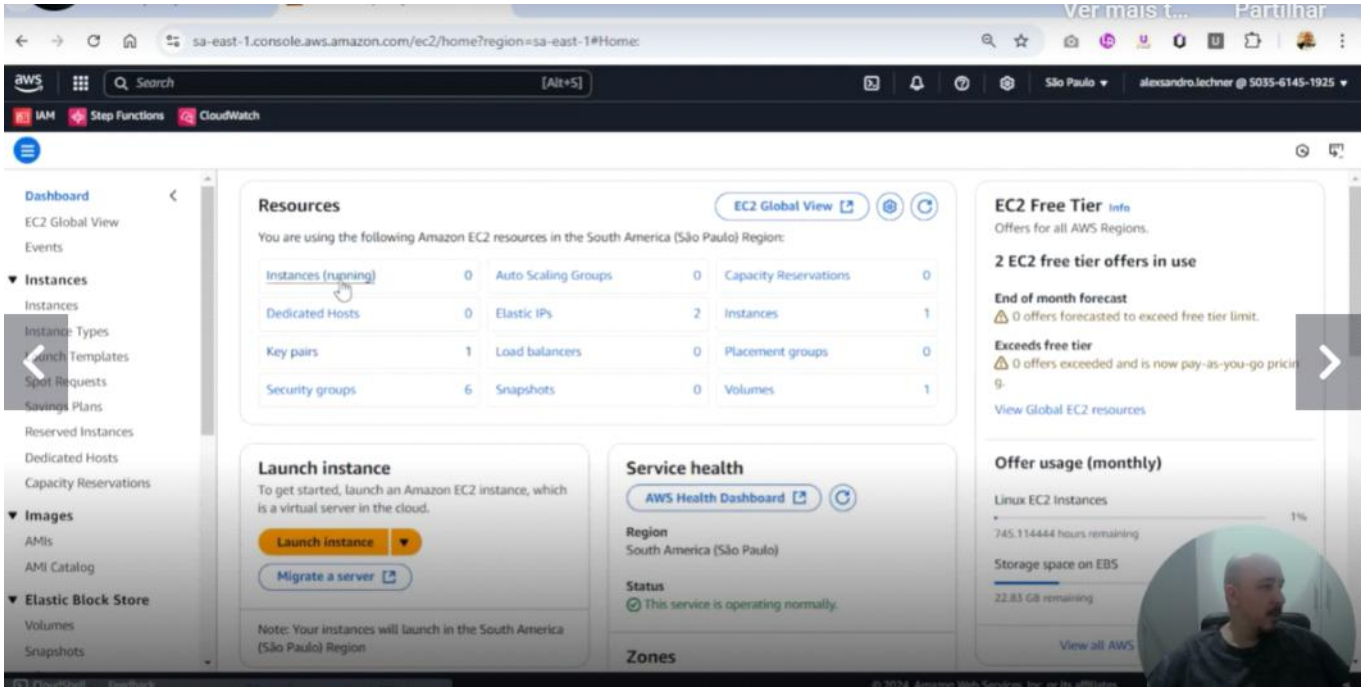
Hardware físico, Infraestrutura dos data centers, Softwares e algumas coisas mais.

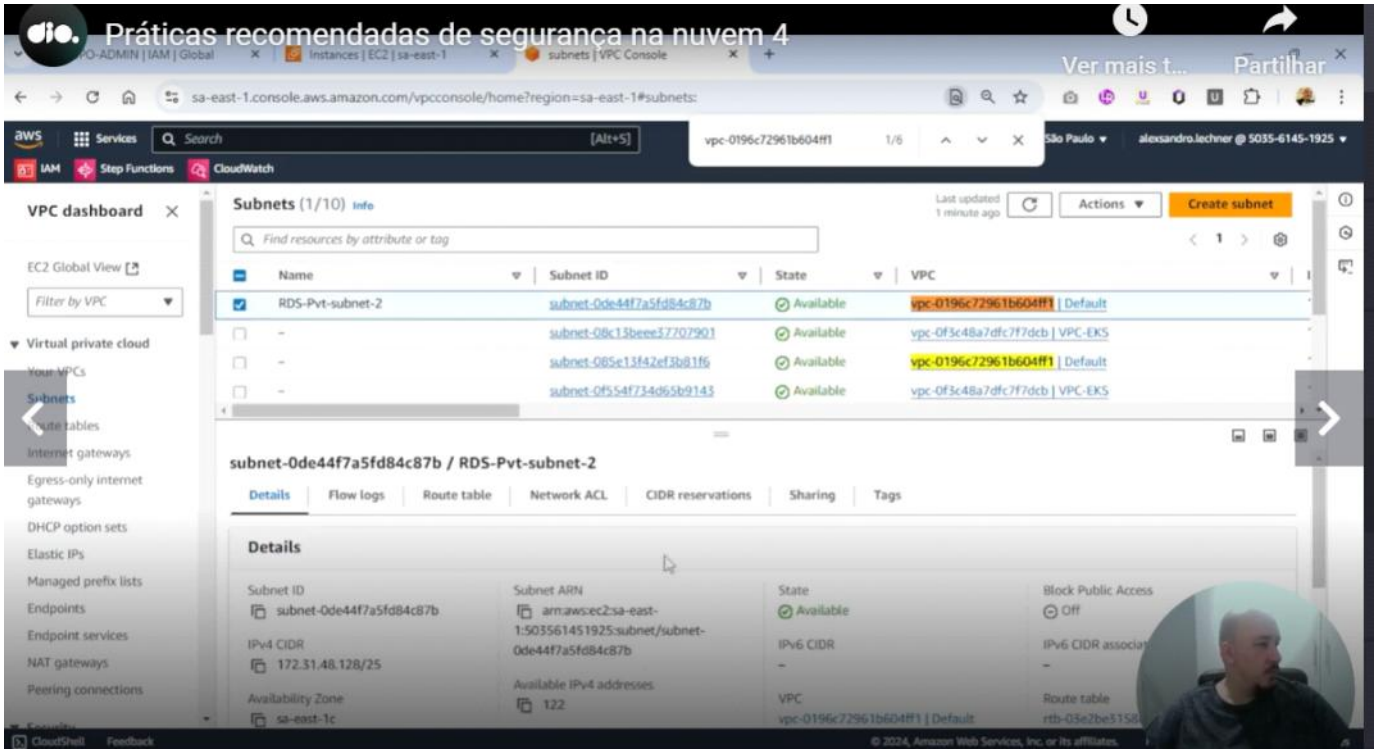
Práticas recomendadas

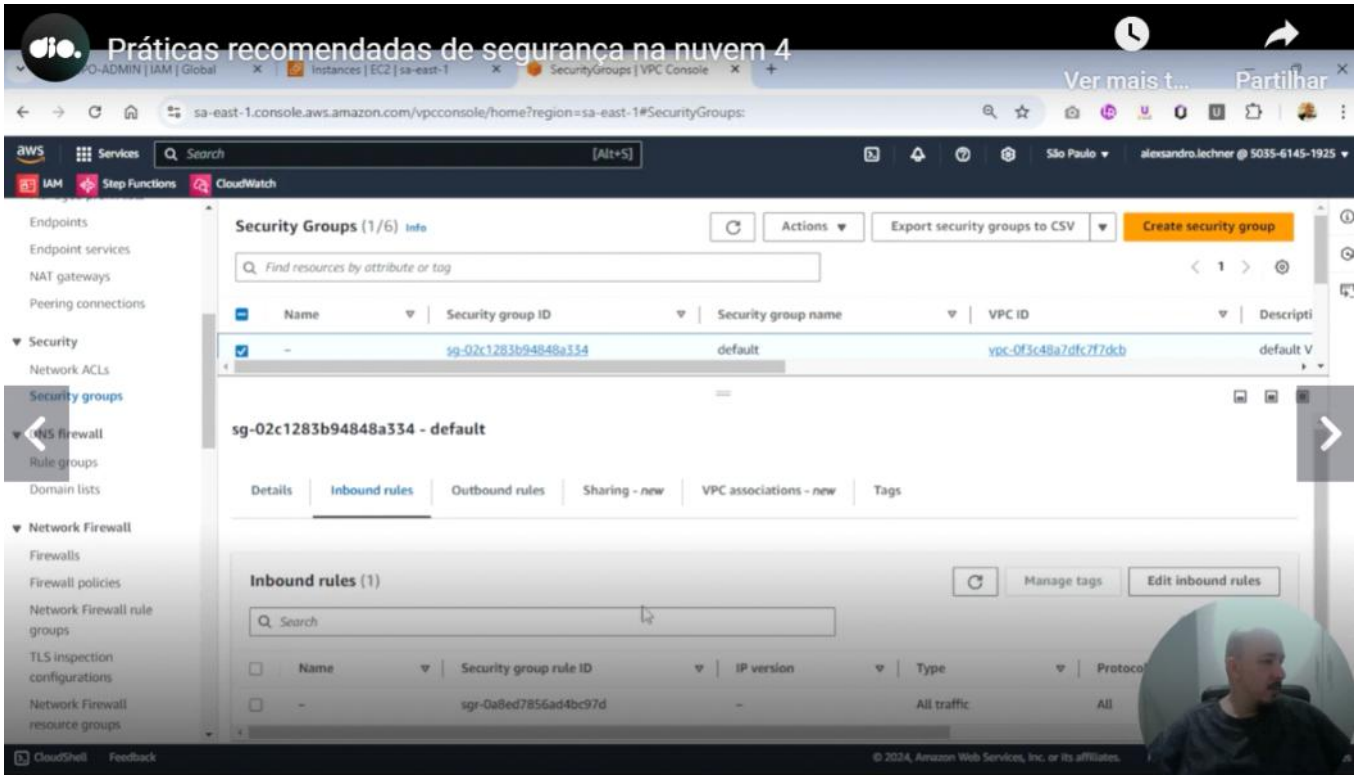


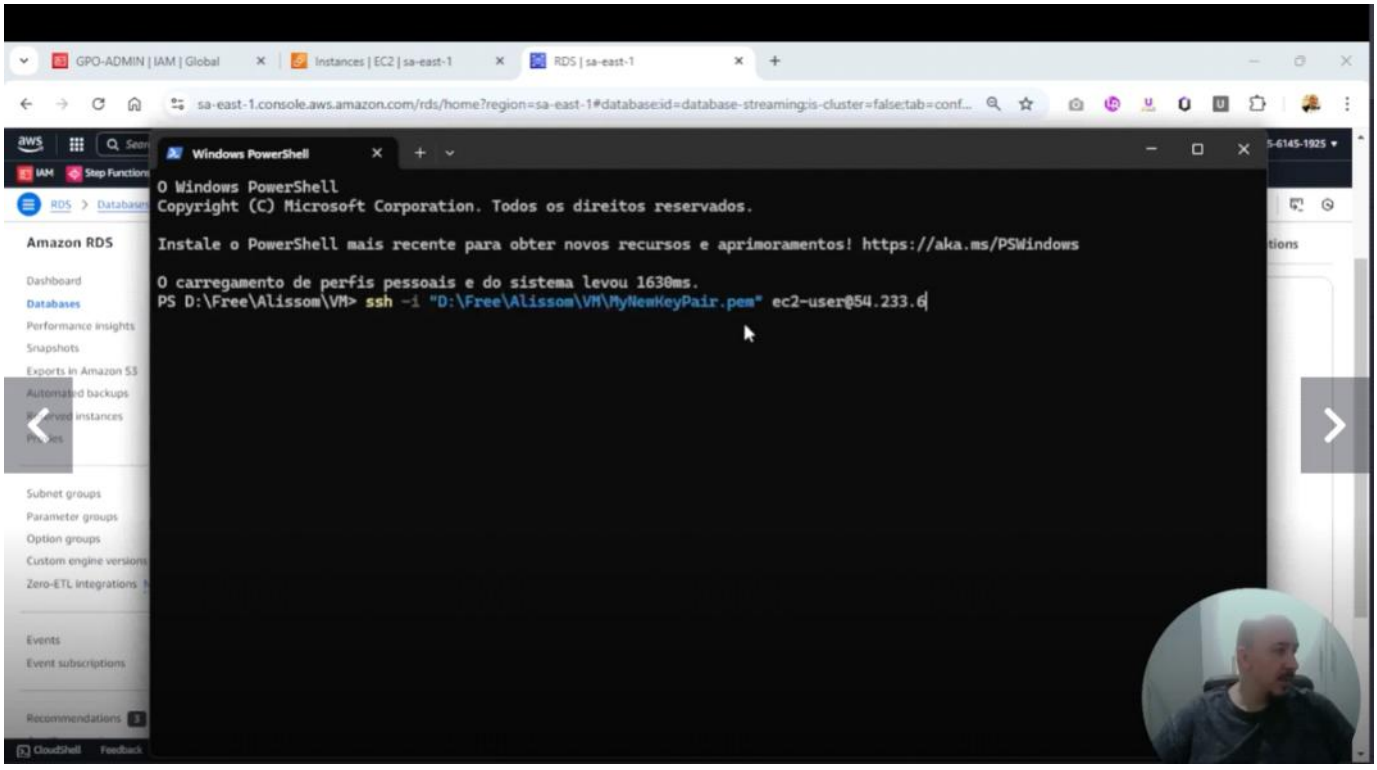
Práticas recomendadas

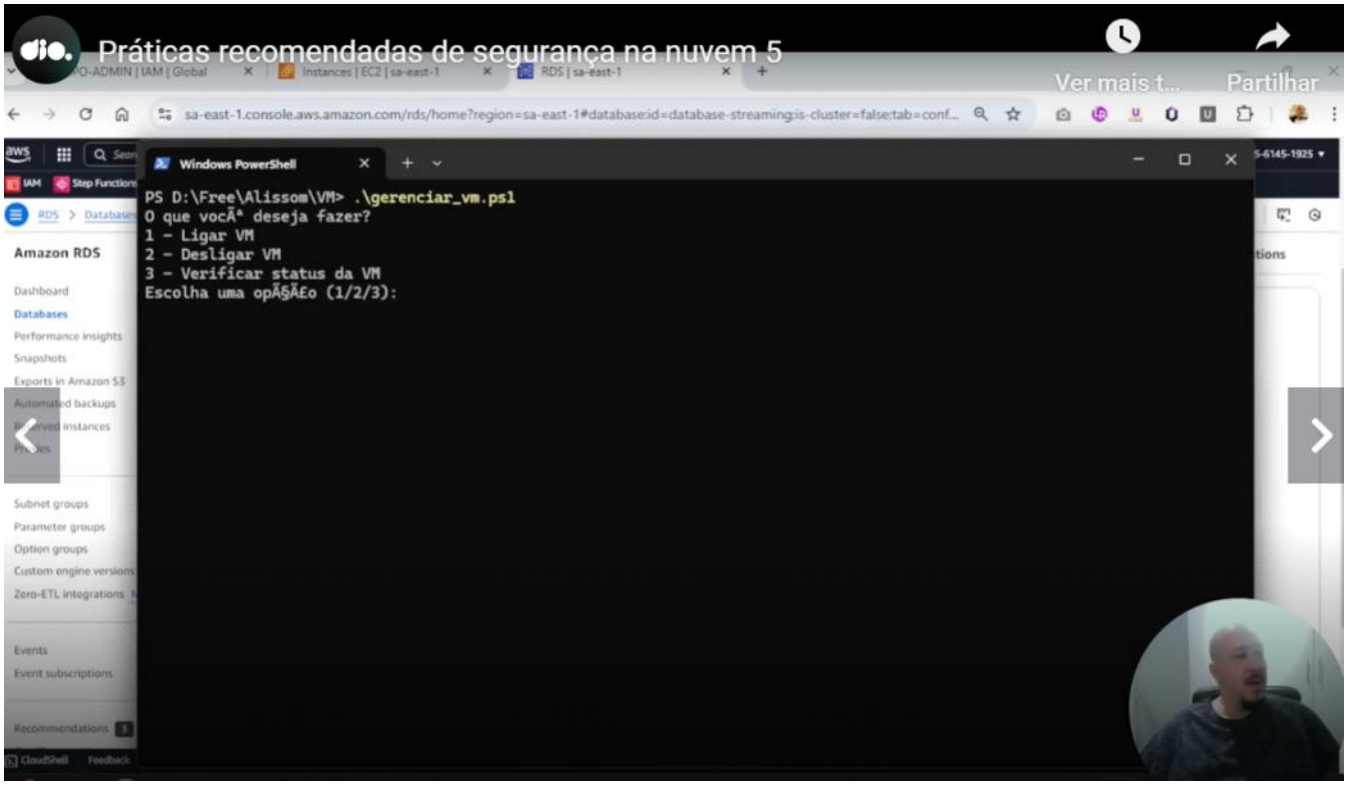
Em casa On Premise	Clube IaaS	Casa do amigo PaaS	Buffet SaaS
Quintal	Quintal	Quintal	Quintal
Música	Música	Música	Música
Churrasqueira	Churrasqueira	Churrasqueira	Churrasqueira
Carne	Carne	Carne	Carne
Comida	Comida	Comida	Comida
Farofa	Farofa	Farofa	Farofa
Bebidas	Bebidas	Bebidas	Bebidas
Água	Água	Água	Água




Práticas recomendadas de segurança na nuvem 4
Ver mais t...
Partilhar
sa-east-1.console.aws.amazon.com/vpcconsole/home?region=sa-east-1#subnets:
VPC dashboard
Subnets (1/10) Info
Actions
Create subnet
Find resources by attribute or tag
Name
Subnet ID
State
VPC
RDS-Pvt-subnet-2
subnet-0de44f7a5fd84c87b
Available
vpc-0196c72961b604ff1 | Default
subnet-08c13beee37707901
vpc-0f3c48a7dfc7f7dcb | VPC-EKS
subnet-085e13f42ef3b81f6
subnet-0f554f734d65b9143
subnet-0de44f7a5fd84c87b / RDS-Pvt-subnet-2
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Subnet ID
Subnet ARN
State
Block Public Access
IPv4 CIDR
172.31.48.128/25
IPv6 CIDR
Availability Zone
sa-east-1c
Available IPv4 addresses
122
VPC
Route table


Práticas recomendadas de segurança na nuvem 4
Ver mais t...
Partilhar
sa-east-1.console.aws.amazon.com/vpcconsole/home?region=sa-east-1#SecurityGroups:
Security Groups (1/6) Info
Actions
Export security groups to CSV
Create security group
Find resources by attribute or tag
Name
Security group ID
Security group name
VPC ID
sg-02c1283b94848a334
default
vpc-0f3c48a7dfc7f7dcb
sg-02c1283b94848a334 - default
Details
Inbound rules
Outbound rules
Sharing - new
VPC associations - new
Tags
Inbound rules (1)
Manage tags
Edit inbound rules
Security group rule ID
IP version
Type
sgr-0a8ed7856ad4bc97d
All traffic


GPO-ADMIN | IAM | Global
Instances | EC2 | sa-east-1
RDS | sa-east-1
sa-east-1.console.aws.amazon.com/rds/home?region=sa-east-1#database:id=database-streaming;is-cluster=false;tab=conf...
Windows PowerShell
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.
Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows
O carregamento de perfis pessoais e do sistema levou 1630ms.
PS D:\Free\Alissom\VM> ssh -i "D:\Free\Alissom\VM\MyNewKeyPair.pem" ec2-user@54.233.6
Amazon RDS
Dashboard
Databases
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations
Events
Event subscriptions
Recommendations
CloudShell
Feedback


Práticas recomendadas de segurança na nuvem 5
Ver mais t...
Partilhar
sa-east-1.console.aws.amazon.com/rds/home?region=sa-east-1#database:id=database-streaming;is-cluster=false;tab=conf...
Windows PowerShell
PS D:\Free\Alissom\VM> .\gerenciar_vm.ps1
O que vocÃª deseja fazer?
1 - Ligar VM
2 - Desligar VM
3 - Verificar status da VM
Escolha uma opÃ§Ã£o (1/2/3):
Amazon RDS
Dashboard
Databases
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations
Events
Event subscriptions
Recommendations
CloudShell
Feedback

Práticas recomendadas de segurança na nuvem 5

Ver mais t... Partilhar

sa-east-1.console.aws.amazon.com/ec2/home?region=sa-east-1#InstanceDetails:instanceId=i-070f47ae1bf113be0

Windows PowerShell

```
2 - Desligar VM
3 - Verificar status da VM
Escolha uma opÃ§Ã£o (1/2/3): 3
Verificando o status da VM...
Status da VM: stopped
PS D:\Free\Alissom\VM> .\gerenciar_vm.ps1
O que vocÃª deseja fazer?
1 - Ligar VM
2 - Desligar VM
3 - Verificar status da VM
Escolha uma opÃ§Ã£o (1/2/3): 1
Ligando a VM...
{
    "StartingInstances": [
        {
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "InstanceId": "i-070f47ae1bf113be0",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}

A VM foi ligada com sucesso.
PS D:\Free\Alissom\VM>
```

Práticas recomendadas de segurança na nuvem 5

Ver mais t... Partilhar

sa-east-1.console.aws.amazon.com/ec2/home?region=sa-east-1#InstanceDetails:instanceId=i-070f47ae1bf113be0

ec2-user@ip-172-31-14-131:~

```
PS D:\Free\Alissom\VM> .\gerenciar_vm.ps1
O que vocÃª deseja fazer?
1 - Ligar VM
2 - Desligar VM
3 - Verificar status da VM
Escolha uma opÃ§Ã£o (1/2/3): 3
Verificando o status da VM...
Status da VM: running
PS D:\Free\Alissom\VM> ssh -i "D:\Free\Alissom\VM\MyNewKeyPair.pem" ec2-user@54.232.56.22
Last login: Fri Nov 15 17:05:35 2024 from 191.163.52.110
       #_
       ~\_  ####_        Amazon Linux 2
      ~~  \_#####\
      ~~     \###|       AL2 End of Life is 2025-06-30.
      ~~       \#/ ___
       ~~       V~' '->
        ~~~         /    A newer version of Amazon Linux is available!
          ~~._.   _/
             _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
           _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

5 package(s) needed for security, out of 6 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-14-131 ~]$
```

Práticas recomendadas de segurança na nuvem 5

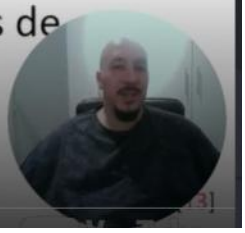
Ver mais t...

Partilhar

Práticas recomendadas

Exemplos práticos:

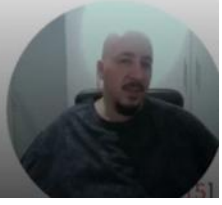
EC2 (Elastic Compute Cloud):
O usuário precisa garantir que as instâncias estejam configuradas de maneira segura, incluindo a escolha de sistemas operacionais atualizados, firewalls, e medidas de autenticação, como chaves SSH seguras.



Práticas recomendadas

Em resumo:
O modelo de responsabilidade da AWS nos ajuda a entender onde terminam as obrigações da AWS e onde começam as nossas.

A AWS se dedica em manter uma infraestrutura mundial com altos níveis de segurança e o usuário adota práticas de segurança robustas para proteger seus dados e aplicações.



Práticas recomendadas

AWS Well-Architected Framework
O pilar de segurança do framework, cobre áreas essenciais para uma arquitetura segura, incluindo:

Gerenciamento de Identidade e Acesso: Uso de práticas como o princípio do menor privilégio, gerenciamento de usuários e grupos com o AWS IAM (Identity and Access Management).

Criptografia de Dados na AWS

Introdução à Criptografia de Dados
- Conceitos básicos: Criptografia em repouso e em trânsito
- Importância de proteger dados sensíveis na nuvem

Criptografia de Dados na AWS

Criptografia em repouso concentra-se na proteção dos dados quando estão armazenados em discos ou bancos de dados, garantindo sua segurança mesmo fora de movimento.

Criptografia de Dados na AWS

Encripte os dados em trânsito ou em repouso

Criptografia de Dados na AWS

Criptografia em repouso:

Serviços como **Amazon S3** (armazenamento de objetos), **Amazon RDS** (banco de dados) e **EBS** (armazenamento de blocos) possui criptografia em repouso. Isso significa que os dados são automaticamente criptografados antes de serem gravados e descriptografados ao serem acessados.

Encripte os dados em trânsito ou em repouso

Ver mais t...

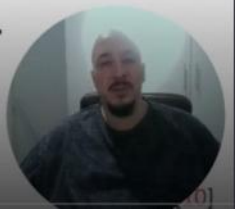
Partilhar

Criptografia de Dados na AWS

Criptografia em trânsito:

Os dados são protegidos enquanto são transmitidos entre clientes e serviços da AWS.

Isso garante que as informações não possam ser interceptadas ou manipuladas durante a transferência.



Encripte os dados em trânsito ou em repouso

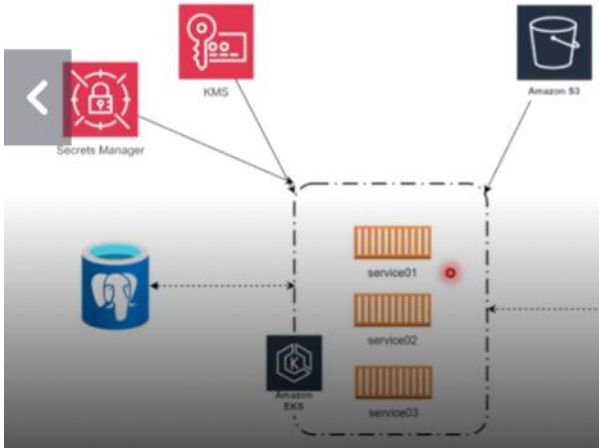
Ver mais t...

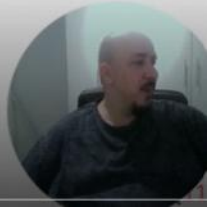
Partilhar

Criptografia de Dados na AWS

AWS KMS (Key Management Service):
Gerencia chaves de criptografia usadas para proteger dados.
Focado em criptografia de dados (em repouso ou em trânsito) e no controle de acesso às chaves.

- Quando precisamos criptografar dados em serviços como: S3, EBS (discos), DynamoDB (tabelas), RDS (bancos de dados).
- E quando desenvolvemos aplicações que exigem criptografia personalizada (usando SDK de KMS).





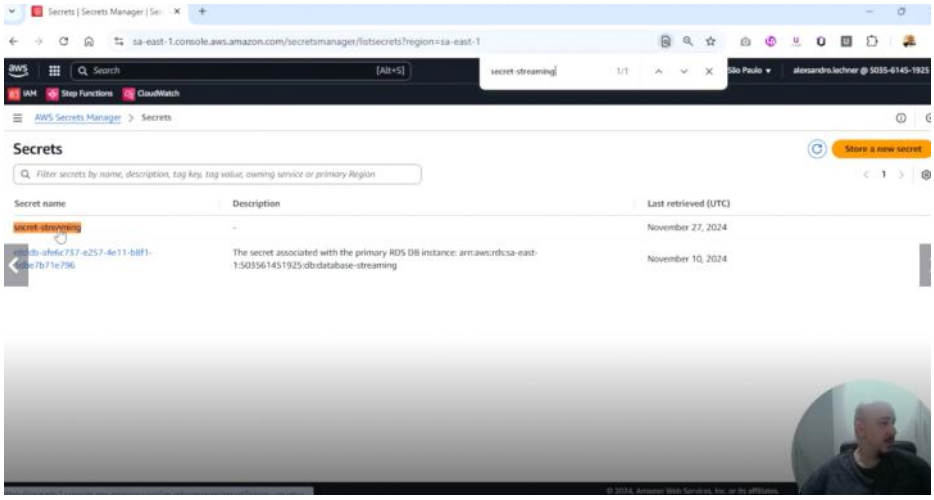
```
});

// Inicialização da aplicação e carregamento das credenciais
(async () => {
  try {
    const secretId = 'secret-streaming';
    const dbCredentials = await getCachedSecret(secretId);

    console.log(
      '>>>>>>>>>>>>> Credenciais obtidas do Secrets Manager:',
      dbCredentials
    );

    // Configura as variáveis de ambiente com as credenciais obtidas
    process.env.DB_USER = dbCredentials.DB_USER;
    process.env.DB_PASSWORD = dbCredentials.DB_PASSWORD;
    process.env.DB_HOST = dbCredentials.DB_HOST;
    process.env.DB_NAME = dbCredentials.DB_NAME;
    process.env.AWS_ACCESS_KEY_ID = dbCredentials.AWS_ACCESS_KEY_ID;
    process.env.AWS_SECRET_ACCESS_KEY = dbCredentials.AWS_SECRET_ACCESS_KEY;
```

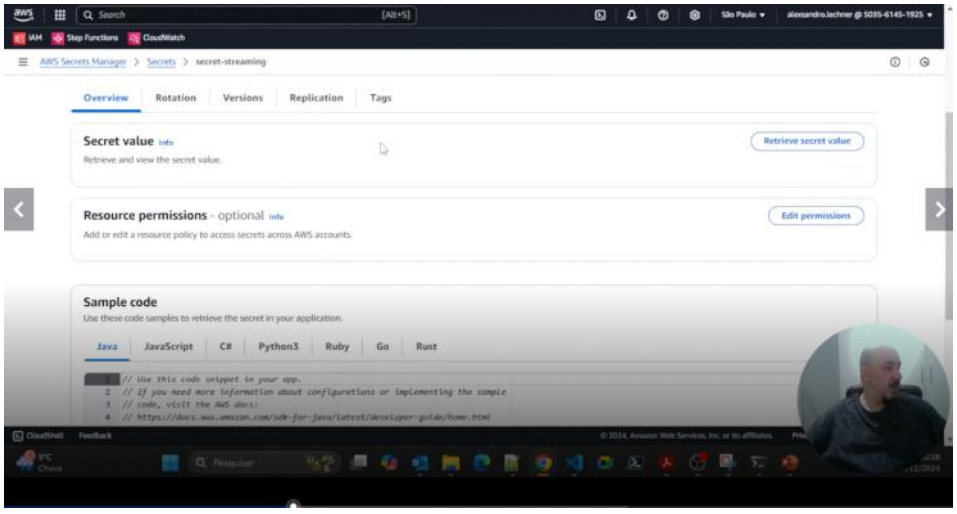
```
View build details: docker-desktop://dashboard/build/upbeat_carson/upbeat_carson0/ut2m8w45uaeo9m2pv2pnubqpb
PS D:\Free\Alissom\#COD\music-service>
History restored

PS D:\Free\Alissom\#COD\music-service>
History restored

PS D:\Free\Alissom\#COD\music-service>
```

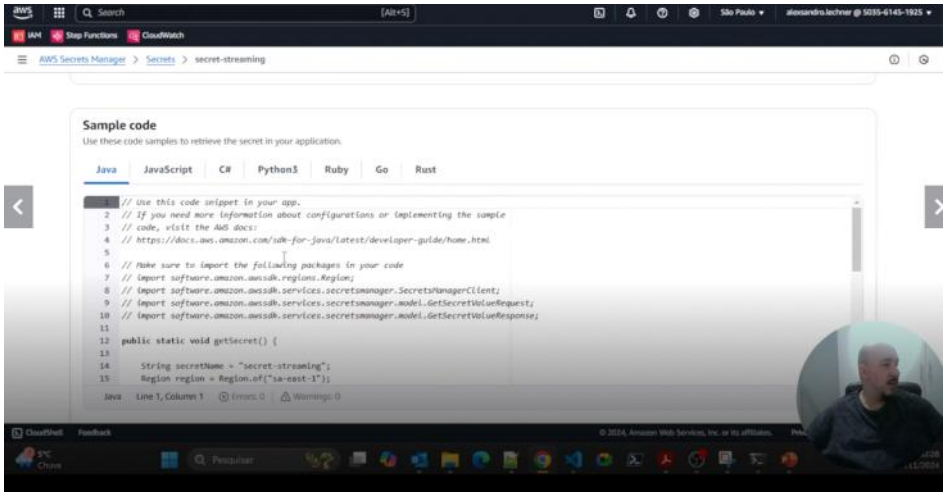
sa-east-1.console.aws.amazon.com/secretsmanager/home?region=sa-east-1

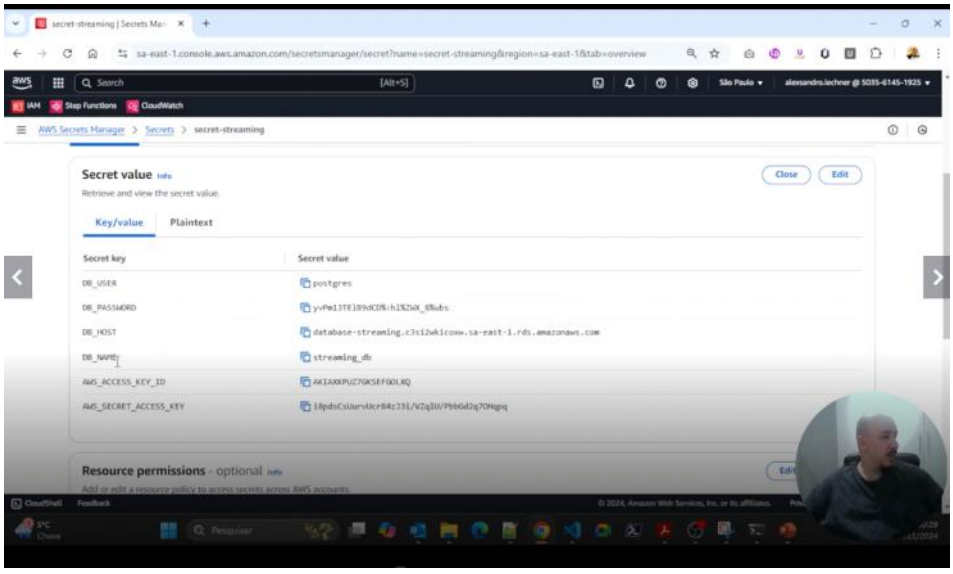
Services
- Secrets Manager — Easily rotate, manage, and retrieve secrets throughout their lifecycle
- Amazon CodeCatalyst — Integrated DevOps Service
- IAM — Manage access to AWS resources

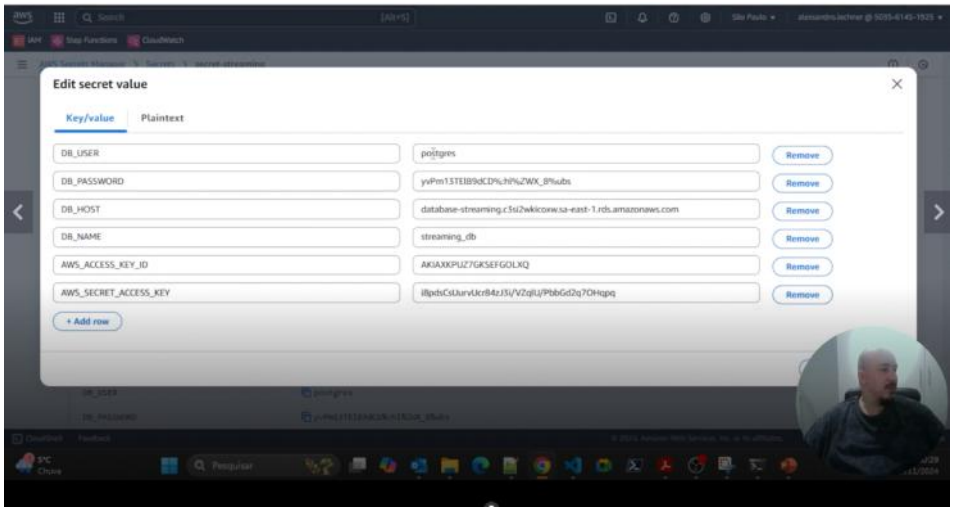
Features
- Secrets — Secrets Manager feature
- API Destinations — Amazon EventBridge feature
- Parameter Store — Systems Manager feature

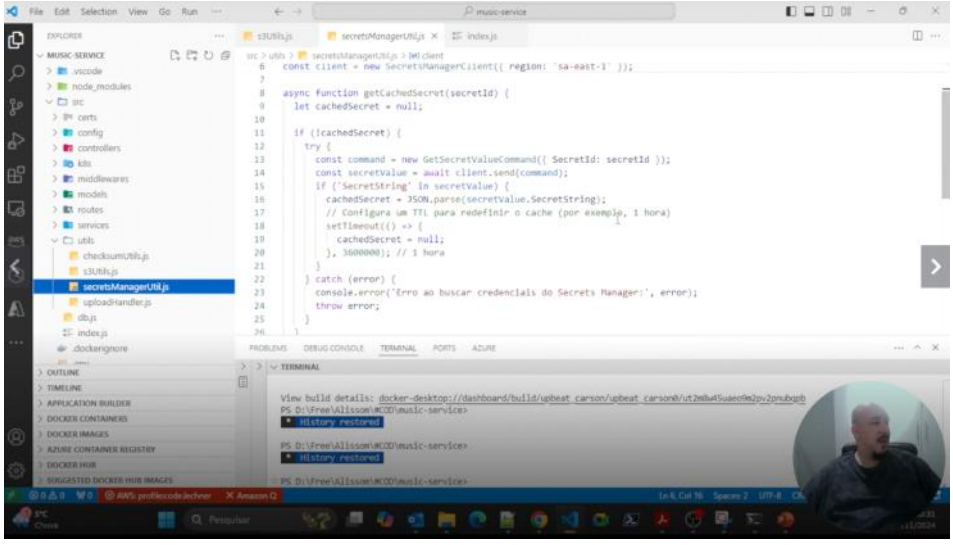

Secrets | Secrets Manager | Se
sa-east-1.console.aws.amazon.com/secretsmanager/listsecrets?region=sa-east-1
Search
[Alt+S]
secret-streaming
1/1
São Paulo
alexsandro.lechner @ 5035-6145-1925
IAM
Step Functions
CloudWatch
AWS Secrets Manager > Secrets
Secrets
Store a new secret
Filter secrets by name, description, tag key, tag value, owning service or primary Region
Secret name
Description
Last retrieved (UTC)
secret-streaming
-
November 27, 2024
rds!db-afe6c757-e257-4e11-b8f1-
d8e7b71e796
The secret associated with the primary RDS DB instance: arn:aws:rds:sa-east-1:503561451925:db:database-streaming
November 10, 2024


Search
[Alt+S]
São Paulo
alexsandro.lechner @ 5035-6145-1925
IAM
Step Functions
CloudWatch
AWS Secrets Manager > Secrets > secret-streaming
Overview
Rotation
Versions
Replication
Tags
Secret value Info
Retrieve and view the secret value.
Retrieve secret value
Resource permissions - optional Info
Add or edit a resource policy to access secrets across AWS accounts.
Edit permissions
Sample code
Use these code samples to retrieve the secret in your application.
Java
JavaScript
C#
Python3
Ruby
Go
Rust
// Use this code snippet in your app.
// If you need more information about configurations or implementing the sample
// code, visit the AWS docs:
// https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/home.html
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.


Search
[Alt+S]
São Paulo
alexsandro.lechner @ 5035-6145-1925
IAM
Step Functions
CloudWatch
AWS Secrets Manager > Secrets > secret-streaming
Sample code
Use these code samples to retrieve the secret in your application.
Java
JavaScript
C#
Python3
Ruby
Go
Rust
// Use this code snippet in your app.
// If you need more information about configurations or implementing the sample
// code, visit the AWS docs:
// https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/home.html
// Make sure to import the following packages in your code
// import software.amazon.awssdk.regions.Region;
// import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
// import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
// import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
public static void getSecret() {
String secretName = "secret-streaming";
Region region = Region.of("sa-east-1");
Java
Line 1, Column 1
Errors: 0
Warnings: 0
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.


secret-streaming | Secrets Ma
sa-east-1.console.aws.amazon.com/secretsmanager/secret?name=secret-streaming&region=sa-east-1&tab=overview
Search
[Alt+S]
São Paulo
alessandro.lechner @ 5035-6145-1925
IAM
Step Functions
CloudWatch
AWS Secrets Manager > Secrets > secret-streaming
Secret value Info
Retrieve and view the secret value.
Close
Edit
Key/value
Plaintext
Secret key
Secret value
DB_USER
postgres
DB_PASSWORD
DB_HOST
database-streaming.c5si2wkicoxw.sa-east-1.rds.amazonaws.com
DB_NAME
streaming_db
AWS_ACCESS_KEY_ID
AKIAXXPUZ7GKSEFGOLXQ
AWS_SECRET_ACCESS_KEY
Resource permissions - optional Info
Add or edit a resource policy to access secrets across AWS accounts.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.

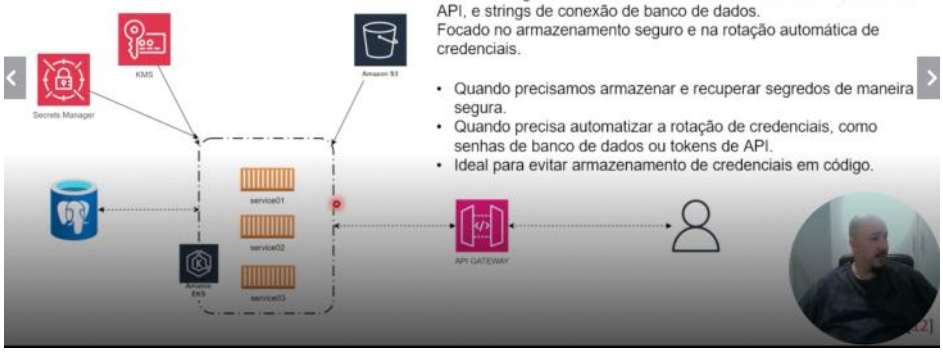

Edit secret value
Key/value
Plaintext
DB_USER
postgres
Remove
DB_PASSWORD
yvPm13TEIB9dCD%:hI%ZWX_8%ubs
DB_HOST
database-streaming.c5si2wkicoxw.sa-east-1.rds.amazonaws.com
DB_NAME
streaming_db
AWS_ACCESS_KEY_ID
AKIAXXPUZ7GKSEFGOLXQ
AWS_SECRET_ACCESS_KEY
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
+ Add row


File Edit Selection View Go Run
music-service
EXPLORER
MUSIC-SERVICE
.vscode
node_modules
src
certs
config
controllers
k8s
middlewares
models
routes
services
utils
checksumUtils.js
s3Utils.js
secretsManagerUtil.js
uploadHandler.js
db.js
index.js
.dockerignore
OUTLINE
TIMELINE
APPLICATION BUILDER
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
s3Utils.js
secretsManagerUtil.js
index.js
src > utils > secretsManagerUtil.js > (#) client
const client = new SecretsManagerClient({ region: 'sa-east-1' });
async function getCachedSecret(secretId) {
let cachedSecret = null;
if (!cachedSecret) {
try {
const command = new GetSecretValueCommand({ SecretId: secretId });
const secretValue = await client.send(command);
if ('SecretString' in secretValue) {
cachedSecret = JSON.parse(secretValue.SecretString);
// Configura um TTL para redefinir o cache (por exemplo, 1 hora)
setTimeout(() => {
cachedSecret = null;
}, 3600000); // 1 hora
}
} catch (error) {
console.error('Erro ao buscar credenciais do Secrets Manager:', error);
throw error;
}
PROBLEMS
DEBUG CONSOLE
TERMINAL
PORTS
AZURE
TERMINAL
View build details: docker-desktop://dashboard/build/upbeat_carson/upbeat_carson0/ut2m8u45uaec9m2ov2pnubgpb
PS D:\Free\Alisson\MCCD\music-service>
History restored
PS D:\Free\Alisson\MCCD\music-service>
History restored
PS D:\Free\Alisson\MCCD\music-service>
AWS: profilecodelechner
Amazon Q
Ln 6, Col 76
Spaces: 2
UTF-8

Criptografia de Dados na AWS

AWS Secrets Manager:
Gerencia segredos e credenciais sensíveis, como senhas, tokens de API, e strings de conexão de banco de dados.
Focado no armazenamento seguro e na rotação automática de credenciais.

- Quando precisamos armazenar e recuperar segredos de maneira segura.
- Quando precisa automatizar a rotação de credenciais, como senhas de banco de dados ou tokens de API.
- Ideal para evitar armazenamento de credenciais em código.



Criptografia de Dados na AWS

Arquitetura com AWS CloudHSM para suportar assinatura digital e transmissão segura de mensagens ao PIX

https://aws.amazon.com/pt/blogs/aws-brasil/arquitetura-com-aws-cloudhsm-para-suportar-assinatura-digital-e-transmissao-segura-de-mensagens-ao-pix/

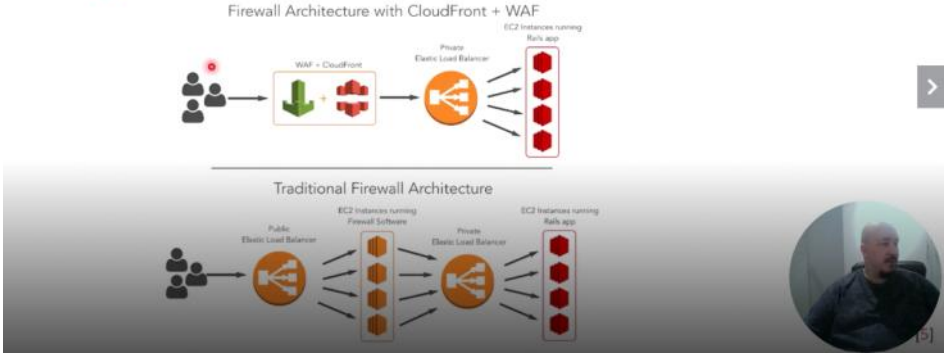
AWS WAF: firewall de aplicativos da web

O AWS WAF (Web Application Firewall) é uma ferramenta que protege seus sites e aplicações na internet contra ataques maliciosos.

amazon WAF

E funciona como uma barreira de segurança, analisando o tráfego que chega ao seu site e bloqueando atividades suspeitas, como tentativas de invasão, ataques de força bruta ou exploração de vulnerabilidades.

AWS WAF: firewall de aplicativos da web

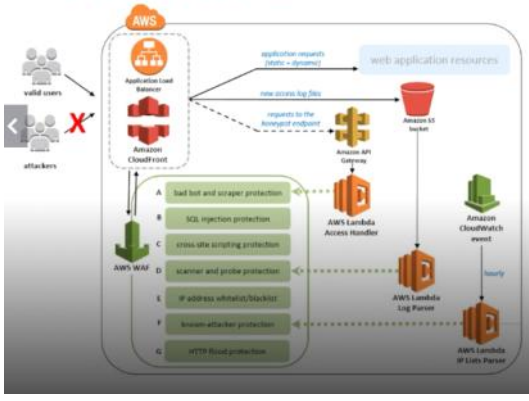


AWS WAF: firewall de aplicativos da web

Com ele podemos criar regras personalizadas no WAF para decidir o que permitir ou bloquear.

Por exemplo, podemos impedir que hackers explorem falhas conhecidas, bloqueiem endereços IP suspeitos ou impeçam ataques comuns, como DDoS e injeções de SQL.

AWS WAF



1. Proteção contra bots maus e scrapers
2. Proteção contra injeção de SQL
3. Proteção contra scripts
4. Proteção do scanner
5. Lista branca e lista negra de Ips
6. Proteção contra atacantes conhecidos
7. Proteção contra inundações HTTP

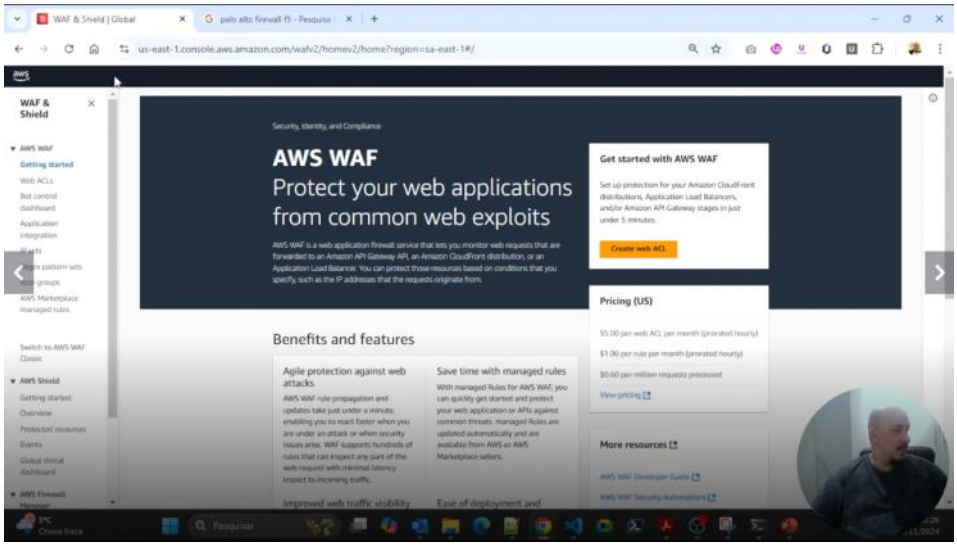
AWS WAF: firewall de aplicativos da web

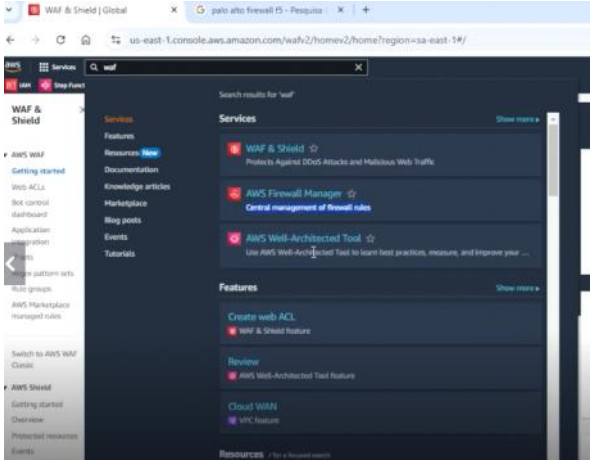
1. Proteção contra bots maus e scrapers
Quando o script AWS CloudFormation é iniciado, fornece ao utilizador um URL de honeypot na saída que é inserido em qualquer aplicação Web como uma ligação HTML oculta. Assim, se alguém tentar aceder a esse URL de forma anónima, esse IP específico será bloqueado e o utilizador não poderá continuar a aceder à aplicação Web.

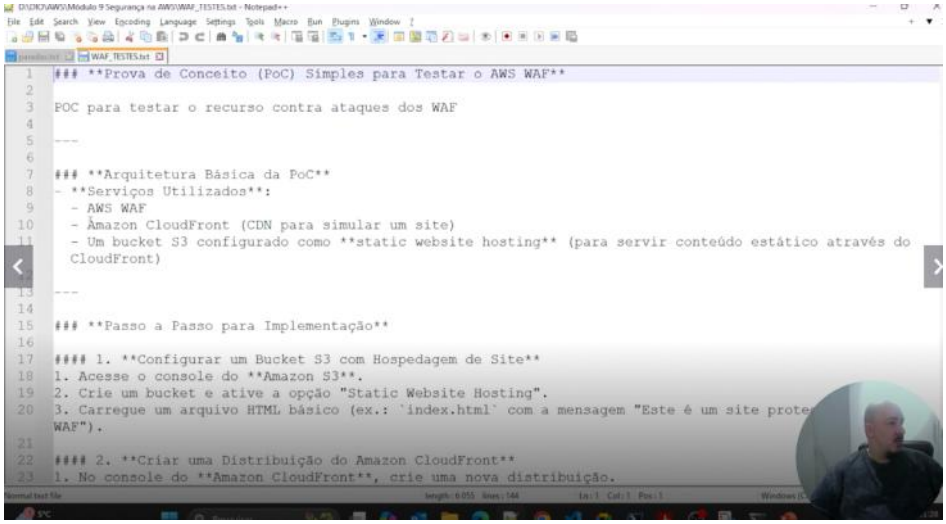
2. Proteção contra injeção de SQL
A regra de injeção de SQL protege a aplicação web contra ataques de injeção de SQL. Analisa os URL, a cadeia de consulta, os cabeçalhos e o corpo do HTML.

3. Proteção contra scripts entre sítios
A proteção contra scripts entre sítios cria uma regra que protege a sua aplicação Web contra scripts XSS. Analisa os URLs, a cadeia de consulta, os cabeçalhos e o corpo HTML. É definido um conjunto de regras semelhante para os ataques XSS, tal como foi definido para a injeção de SQL

4. Proteção do scanner e da sonda
Uma função AWS Lambda personalizada analisa naturalmente os registos de acesso e, consequentemente, analisa a conduta duvidosa e adiciona esse IP a uma lista de endereços IP bloqueados.

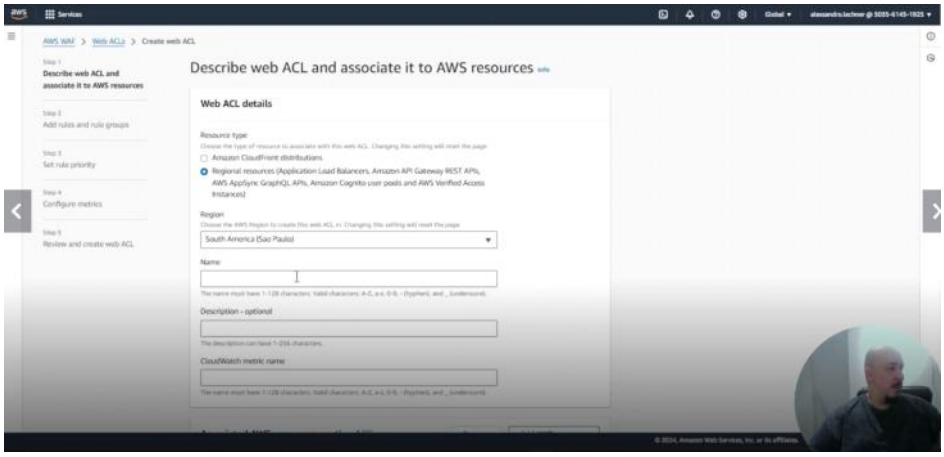




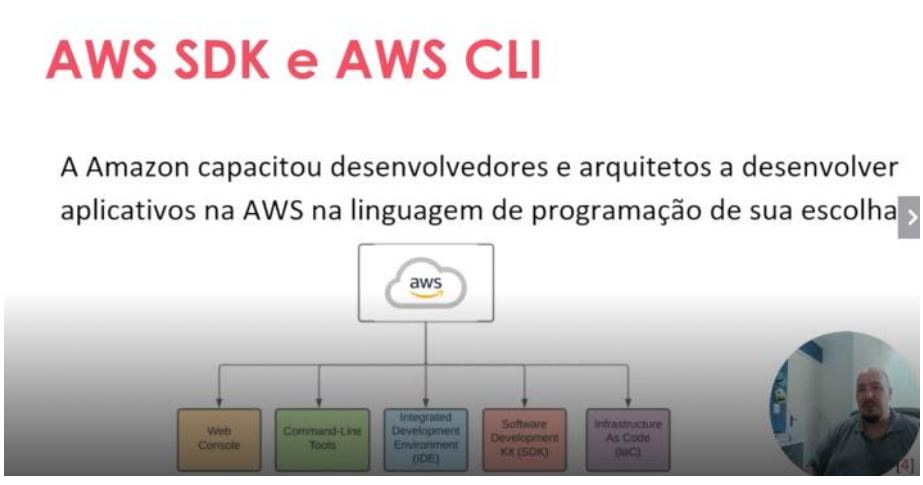

Prova de Conceito (PoC) Simples para Testar o AWS WAF
POC para testar o recurso contra ataques dos WAF

Arquitetura Básica da PoC
- **Serviços Utilizados**:
- AWS WAF
- Amazon CloudFront (CDN para simular um site)
- Um bucket S3 configurado como **static website hosting** (para servir conteúdo estático através do CloudFront)

Passo a Passo para Implementação
1. **Configurar um Bucket S3 com Hospedagem de Site**
1. Acesse o console do **Amazon S3**.
2. Crie um bucket e ative a opção "Static Website Hosting".
3. Carregue um arquivo HTML básico (ex.: `index.html` com a mensagem "Este é um site prote
WAF").
2. **Criar uma Distribuição do Amazon CloudFront**
1. No console do **Amazon CloudFront**, crie uma nova distribuição.


Describe web ACL and associate it to AWS resources
Web ACL details
Resource type
Amazon CloudFront distributions
Regional resources (Application Load Balancers, Amazon API Gateway REST APIs, AWS AppSync GraphQL APIs, Amazon Cognito user pools and AWS Verified Access instances)
Region
South America (Sao Paulo)
Name
Description - optional
CloudWatch metric name

Conteúdo Programático
Módulo 10 Desenvolvimento e Ferramentas
AWS SDKs e AWS CLI: ferramentas de desenvolvimento e linha de comando
AWS CloudFormation: automação de infraestrutura como código
AWS CodeDeploy: implantação automatizada de aplicações

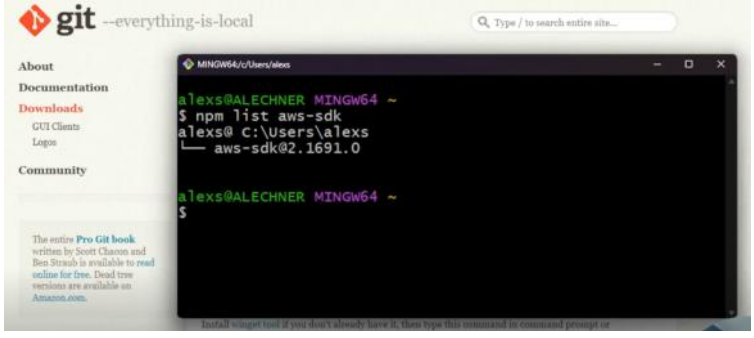

AWS SDK e AWS CLI
A Amazon capacitou desenvolvedores e arquitetos a desenvolver aplicativos na AWS na linguagem de programação de sua escolha
aws
Web Console
Command-Line Tools
Integrated Development Environment (IDE)
Software Development Kit (SDK)
Infrastructure As Code (IaC)

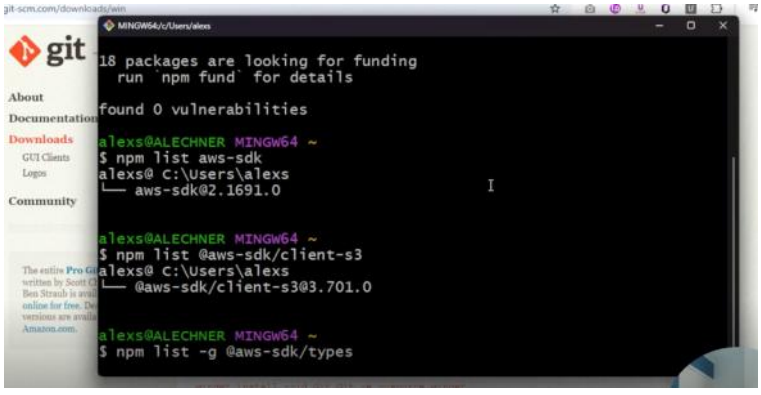
AWS SDK e AWS CLI

O AWS SDK nos permite interagir com os serviços da AWS diretamente em sua linguagem de programação de nossa preferencia, podendo ser:

AWS SDK e AWS CLI

Linguagem	Comando de instalação
Node.js	npm install @aws-sdk/client-s3 –save
Python	pip install boto3
Java	Maven: Adicione <dependency> ao pom.xml Gradle: implementation 'software.amazon.awssdk:s3:2.20.0'
.NET	dotnet add package AWSSDK.S3
Ruby	gem install aws-sdk-s3
PHP	composer require aws/aws-sdk-php
GO	go get github.com/aws/aws-sdk-go-v2





AWS SDK e AWS CLI

src > utils > secretsManagerUtil.js > ...

```
1  const {
2    SecretsManagerClient,
3    GetSecretValueCommand,
4  } = require('@aws-sdk/client-secrets-manager');
5
6  const client = new SecretsManagerClient({ region: 'sa-east-1' });
7
8  async function getCachedSecret(secretId) {
9    let cachedSecret = null;
10
11   if (!cachedSecret) {
12     try {
13       const command = new GetSecretValueCommand({ SecretId: secretId });
14       const secretValue = await client.send(command);
15       if ('SecretString' in secretValue) {
```

AWS SDK e AWS CLI

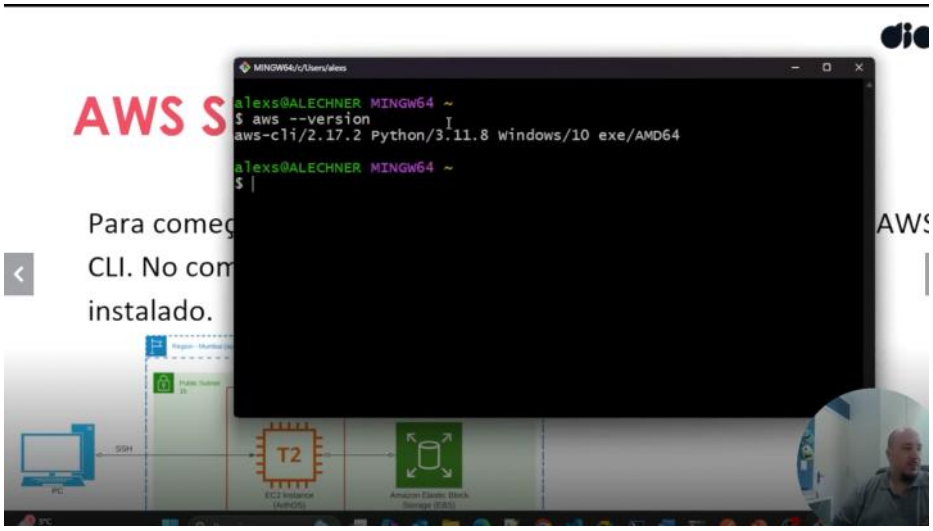
O AWS CLI (Command Line Interface) é uma ferramenta baseada em linha de comando para gerenciar a AWS.
É uma ferramenta ideal para administradores que precisam gerenciar infraestrutura.

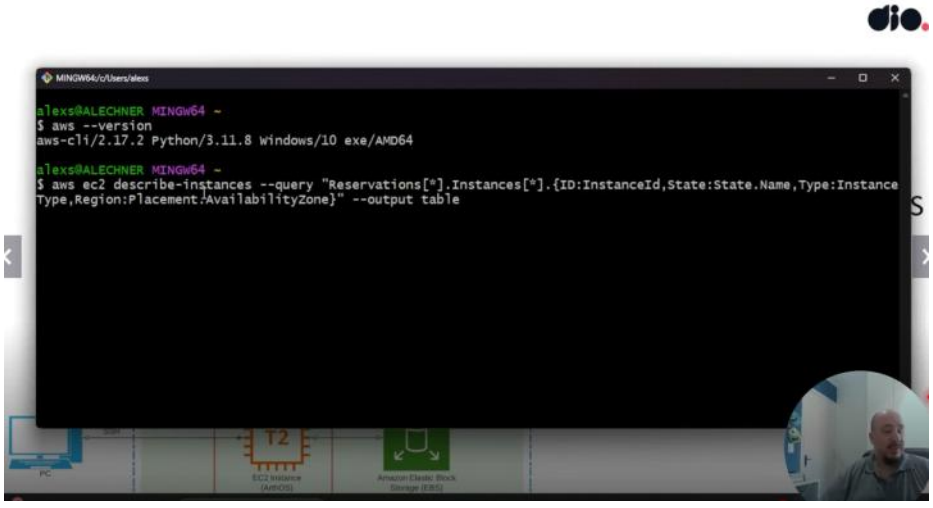
AWS CLI

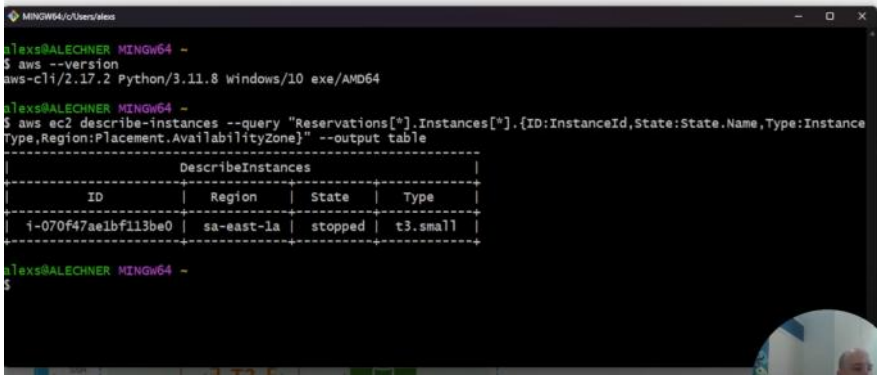
AWS SDK e AWS CLI

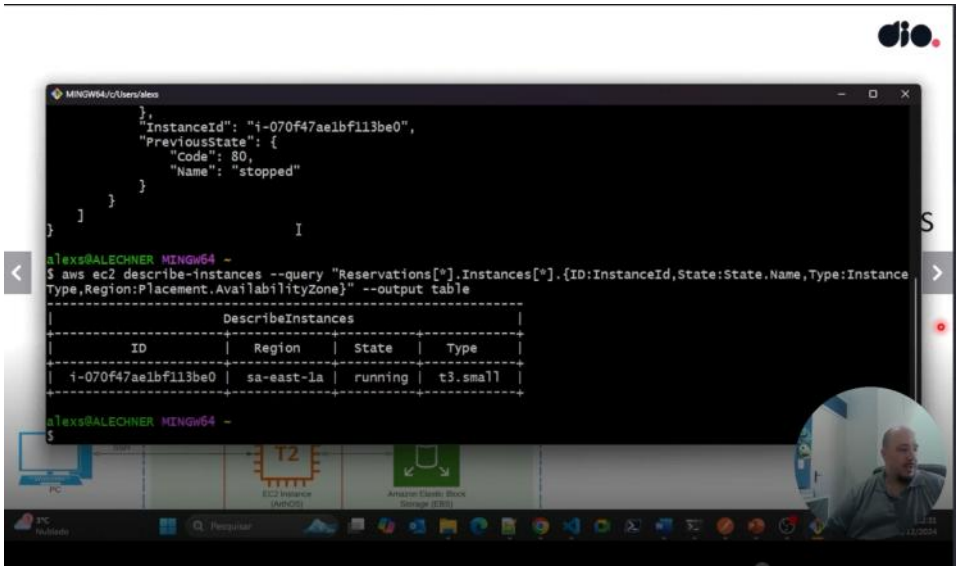
Para começarmos a entender, temos que validar a versão do AWS CLI. No comando aws –version nos exibe a versão do CLI instalado.

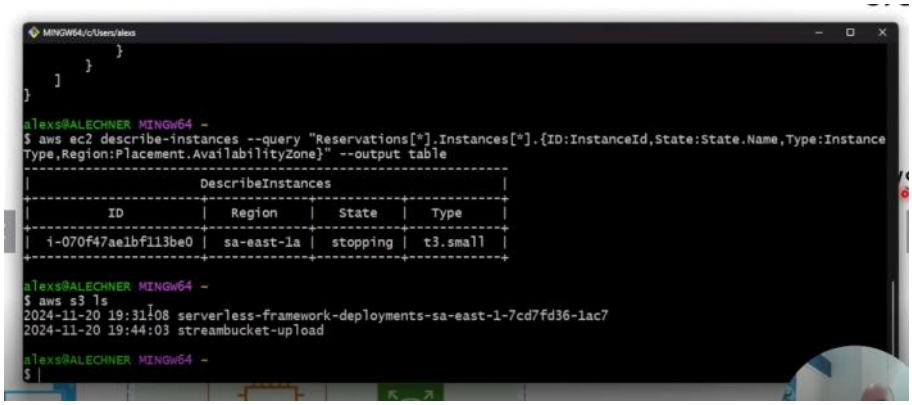
```
PowerShell 7 (x64)
PowerShell 7.4.6
PS C:\Users\alexs> aws --version
aws-cli/2.17.2 Python/3.11.8 Windows/10 exe/AMD64
```

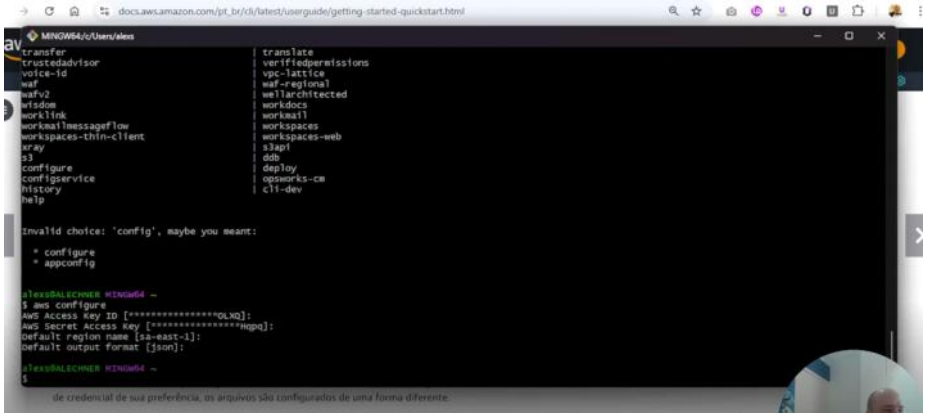

AWS S
Para começ
AWS
CLI. No com
instalado.
MINGW64:/c/Users/alexs
alexs@ALECHNER MINGW64 ~
$ aws --version
aws-cli/2.17.2 Python/3.11.8 Windows/10 exe/AMD64
alexs@ALECHNER MINGW64 ~
$

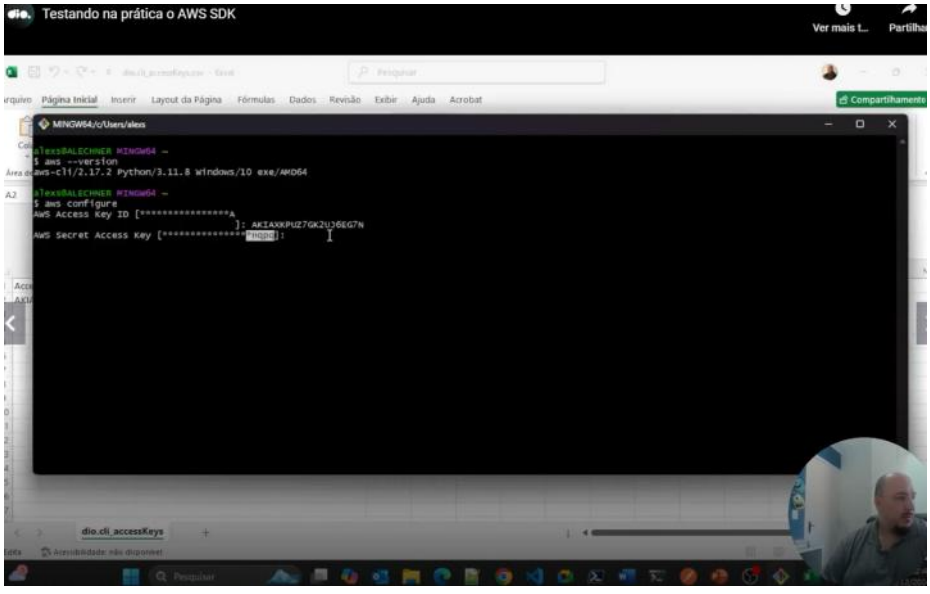

MINGW64:/c/Users/alexs
alexs@ALECHNER MINGW64 ~
$ aws --version
aws-cli/2.17.2 Python/3.11.8 Windows/10 exe/AMD64
alexs@ALECHNER MINGW64 ~
$ aws ec2 describe-instances --query "Reservations[*].Instances[*].{ID:InstanceId,State:State.Name,Type:Instance
Type,Region:Placement.AvailabilityZone}" --output table


MINGW64:/c/Users/alexs
alexs@ALECHNER MINGW64 ~
$ aws --version
aws-cli/2.17.2 Python/3.11.8 Windows/10 exe/AMD64
alexs@ALECHNER MINGW64 ~
$ aws ec2 describe-instances --query "Reservations[*].Instances[*].{ID:InstanceId,State:State.Name,Type:Instance
Type,Region:Placement.AvailabilityZone}" --output table
DescribeInstances
ID
Region
State
Type
i-070f47ae1bf113be0
sa-east-1a
stopped
t3.small
alexs@ALECHNER MINGW64 ~
$

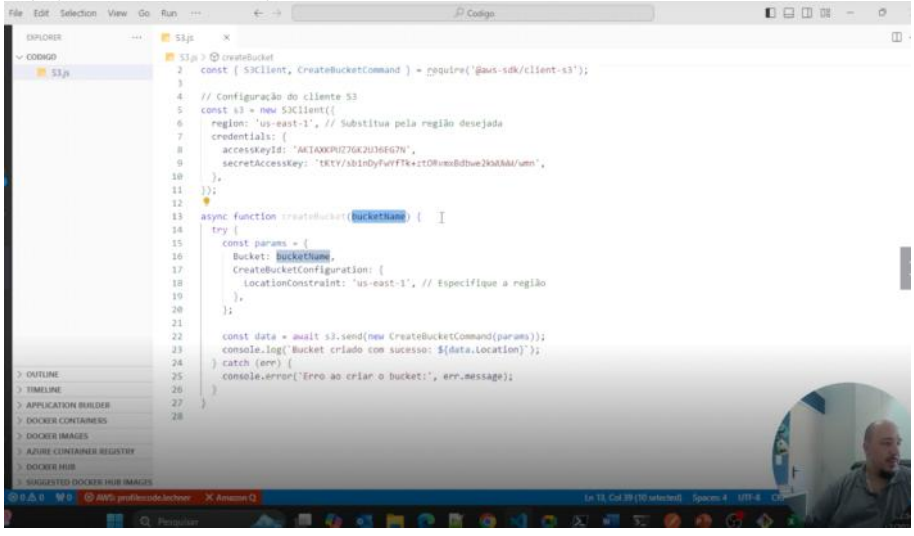

MINGW64:/c/Users/alexs
"InstanceId": "i-070f47ae1bf113be0",
"PreviousState": {
"Code": 80,
"Name": "stopped"
alexs@ALECHNER MINGW64 ~
$ aws ec2 describe-instances --query "Reservations[*].Instances[*].{ID:InstanceId,State:State.Name,Type:Instance
Type,Region:Placement.AvailabilityZone}" --output table
DescribeInstances
ID
Region
State
Type
i-070f47ae1bf113be0
sa-east-1a
running
t3.small
alexs@ALECHNER MINGW64 ~
$







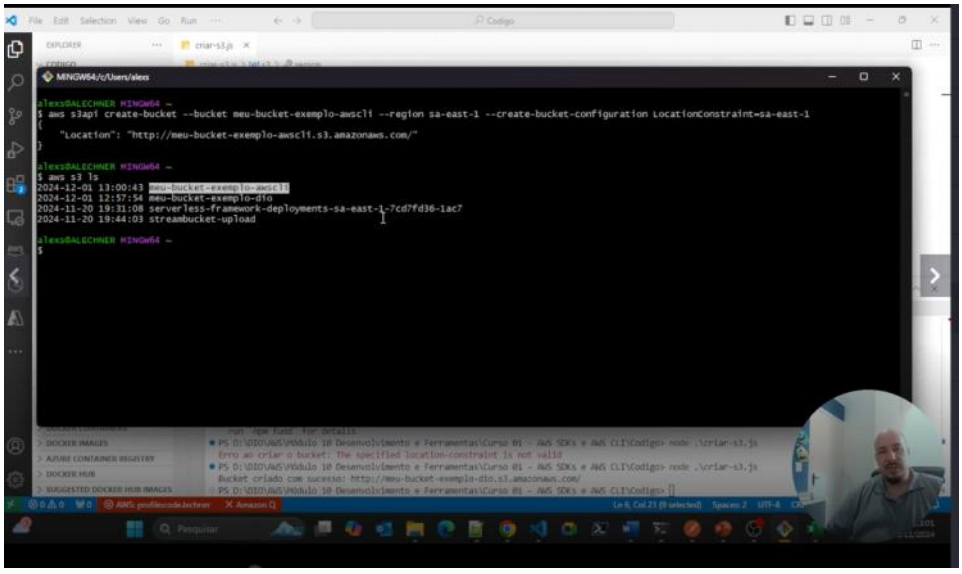
Função para criar um bucket em node js



```
const s3 = new S3Client({
  credentials: {
    accessKeyId: 'AKIAKKPUZ/GK2UJ6G6/N ,
    secretAccessKey: 'tKtY/sb1nDyFwYfTk+zt0RvmxBdbue2kWAWd/wmn',
  },
});

async function createBucket(bucketName) {
  try {
    const params = {
      Bucket: bucketName,
      CreateBucketConfiguration: {
        LocationConstraint: 'us-east-1',
      },
    };

    const data = await s3.send(new CreateBucketCommand(params));
    console.log(`Bucket criado com sucesso: ${data.Location}`);
  } catch (err) {
    console.error('Erro ao criar o bucket:', err.message);
  }
}

createBucket('meu-bucket-exemplo-dio');
```

```
Node.js v20.16.0
PS D:\DIO\AWS\Módulo 10 Desenvolvimento e Ferramentas\Curso 01 - AWS SDKs e AWS CLI\Codigo> npm install @aws-sdk/client-s3

added 103 packages in 20s

2 packages are looking for funding
  run `npm fund` for details
PS D:\DIO\AWS\Módulo 10 Desenvolvimento e Ferramentas\Curso 01 - AWS SDKs e AWS CLI\Codigo> node .\criar-s3.js
Erro ao criar o bucket: The specified location-constraint is not valid
PS D:\DIO\AWS\Módulo 10 Desenvolvimento e Ferramentas\Curso 01 - AWS SDKs e AWS CLI\Codigo> node .\criar-s3.js
Bucket criado com sucesso: http://meu-bucket-exemplo-dio.s3.amazonaws.com/
PS D:\DIO\AWS\Módulo 10 Desenvolvimento e Ferramentas\Curso 01 - AWS SDKs e AWS CLI\Codigo>
```

```
alexs@ALECHNER MINGW64 ~
$ aws s3api create-bucket --bucket meu-bucket-exemplo-awscli --region sa-east-1 --create-bucket-configuration LocationConstraint=sa-east-1
{
    "Location": "http://meu-bucket-exemplo-awscli.s3.amazonaws.com/"
}

alexs@ALECHNER MINGW64 ~
$
```



AWS CloudFormation

Benefícios do AWS Cloudformation

Automação: O AWS CloudFormation ajuda a automatizar o processo de criação, configuração e gerenciamento de recursos da AWS. Isso permite que a infraestrutura seja implantada de forma rápida, confiável e repetida.

AWS CloudFormation

Benefícios do AWS Cloudformation

Consistência e padronização: Com o AWS CloudFormation, é possível criar modelos padrão de pilhas de infraestrutura que podem ser usados para criar cópias idênticas da mesma infraestrutura. Isso garante consistência na implantação da infraestrutura e facilita a manutenção.

AWS CloudFormation

Benefícios do AWS Cloudformation

Economia de custos: o AWS CloudFormation ajuda a reduzir custos permitindo que os clientes usem modelos de infraestrutura existentes e os reutilizem em vários ambientes. Isso reduz o custo de projetar e implementar uma nova infraestrutura.

AWS CloudFormation

Benefícios do AWS Cloudformation

Segurança: O AWS CloudFormation ajuda a garantir que todos os recursos da AWS sejam configurados com segurança usando políticas e regras de segurança. Isso ajuda a proteger a infraestrutura de potenciais ameaças à segurança.

AWS CloudFormation

AWS CloudFormation

JSON: formato tradicional e originalmente suportado pelo CloudFormation e a estrutura baseada em pares chave-valor.

```
{
  "Resources":{
    "MyInstance":{
      "Type":"AWS::EC2::Instance",
      "Properties":{
        "InstanceType":"t2.micro",
        "ImageId":"ami-12345678"
      }
    }
  }
}
```

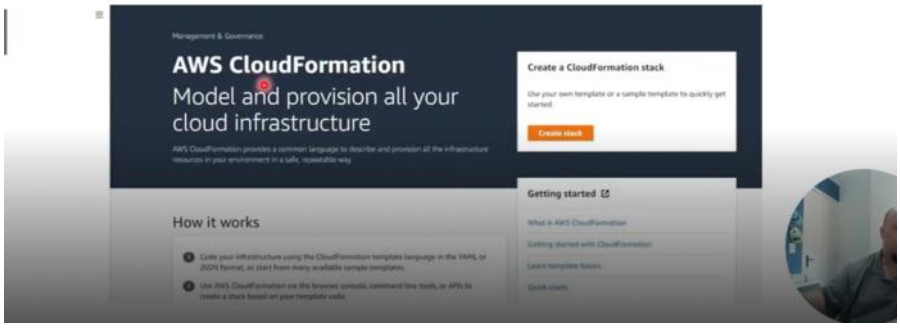
AWS CloudFormation

YAML: formato preferido por muitos, devido à simplicidade.
Exemplo equivalente ao JSON anteior.

```
Resources:
  MyInstance:
    Type: "AWS::EC2::Instance"
    Properties:
      InstanceType: "t2.micro"
      ImageId: "ami-12345678"
```

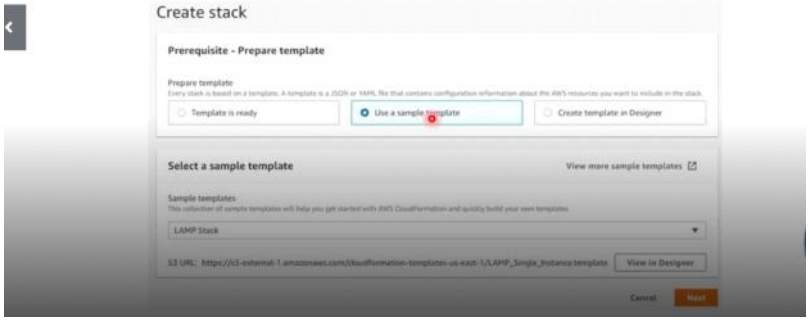
AWS CloudFormation

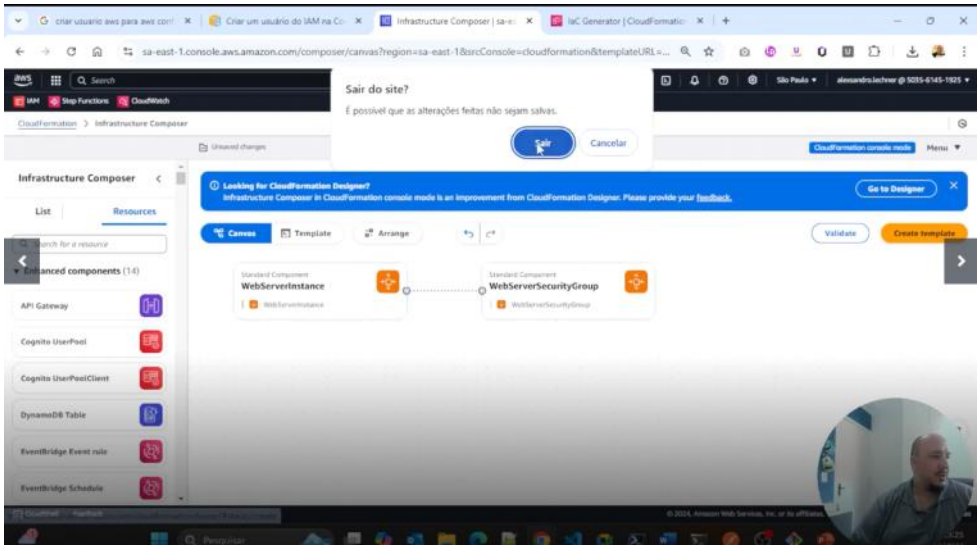
No console da AWS selecione o serviço CloudFormation e clique em Create Stack.



AWS CloudFormation

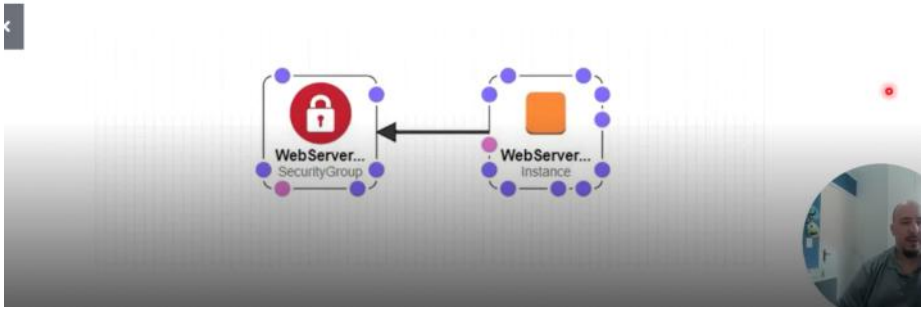
Selecione o template e selecione o LAMP Stack





AWS CloudFormation

Aqui você pode projetar sua infraestrutura adequadamente.

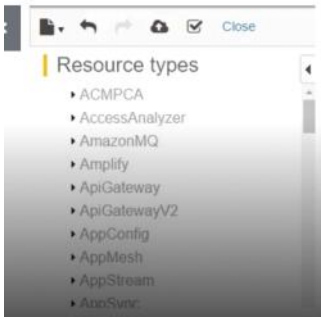


AWS CloudFormation

Este é o código escrito no formato JSON que contém todas as especificações e dependências sobre a infraestrutura a ser criada.

AWS CloudFormation

Agora clique no botão de upload em forma de nuvem para sair do designer.



Nas demais telas iremos finalizar as configurações para a criação do nosso recurso, por exemplo se for um EC2, escolheremos tipo do EC2 e outros detalhes.

AWS CloudFormation

Qual é a diferença entre AWS CloudFormation e Terraform?

AWS CloudFormation	Terraform
Específico da AWS	Suporta vários provedores cloud, o que o torna versátil para ambientes multi-cloud.

AWS CodeDeploy

AWS CodeDeploy é uma ferramenta da AWS que ajuda a automatizar o processo de implantação de aplicativos.

AWS CodeDeploy

Imagine que você está trabalhando em um projeto, por exemplo um site ou aplicativo. E precisa colocá-lo no ar ou atualizar o que já está rodando.

Fazer isso manualmente pode ser demorado, propenso a erros e complicado, especialmente quando há muitos servidores envolvidos.

AWS CodeDeploy

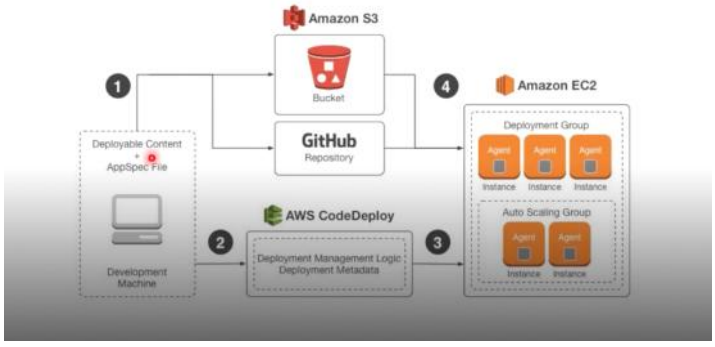
O AWS CodeDeploy é como um "robô" que faz esse trabalho para você:

- Ele pega o seu aplicativo ou código.

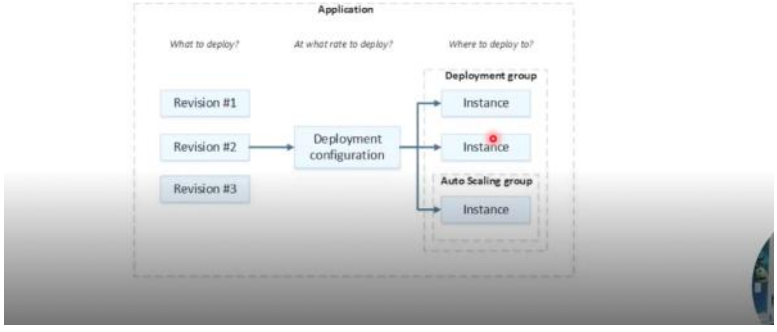
- Distribui automaticamente nos servidores, seja na nuvem (como instâncias EC2) ou no seu próprio datacenter.

- Faz isso de forma segura e eficiente, sem interromper os usuários que estão acessando o sistema.

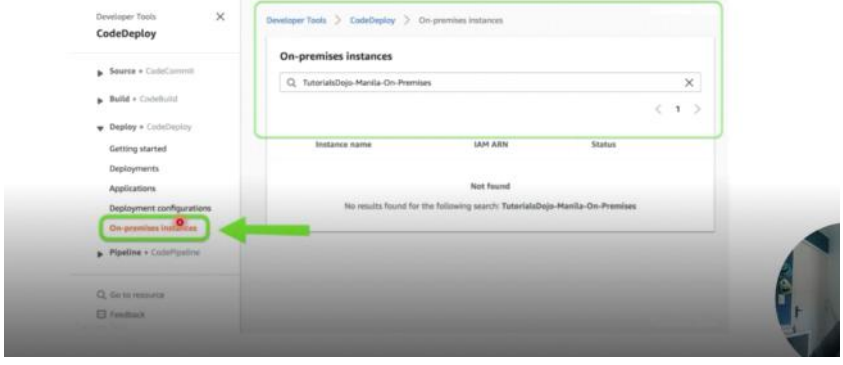
AWS CodeDeploy



AWS CodeDeploy



AWS CodeDeploy



AWS CodeDeploy

Benefícios de usar o AWS CodeDeploy

Veremos o que você pode fazer utilizndo o CodeDeploy:

1. Automatizar implantações de software

O AWS CodeDeploy permite que você automatize suas implantações de software, eliminando a necessidade de intervenções manuais e reduzindo o risco de erro humano. Você pode implantar seus aplicativos simultaneamente em várias instâncias, garantindo um processo de lançament consistente e confiável.

AWS CodeDeploy

Benefícios de usar o AWS CodeDeploy

2. Aumente a velocidade e a eficiência da implantação

Ao automatizar suas implantações, o AWS CodeDeploy acelera significativamente o processo de lançamento. Em vez de implantar alterações manualmente, você pode aproveitar o poder do CodeDeploy para lançar atualizações automaticamente, reduzi tempo necessário para implantar novos recursos ou correçõ bugs.

AWS CodeDeploy

Benefícios de usar o AWS CodeDeploy

3. Garanta a disponibilidade e a escalabilidade do aplicativo

O AWS CodeDeploy fornece recursos integrados para balanceamento de carga e dimensionamento de seus aplicativos. Com o dimensionamento automático, você pode lidar facilmente com picos de tráfego e garantir que seus aplicativos estejam sempre disp ais para seus usuários, mesmo durante períodos de pico.

AWS CodeDeploy

Benefícios de usar o AWS CodeDeploy

5. Integração perfeita com serviços da AWS

O AWS CodeDeploy integra-se com outros serviços da AWS, como Amazon EC2, Elastic Load Balancing e Amazon S3. Essa integração permite que você aproveite todo o poder do ecossistema da AWS e crie aplicativos escaláveis e resilientes.

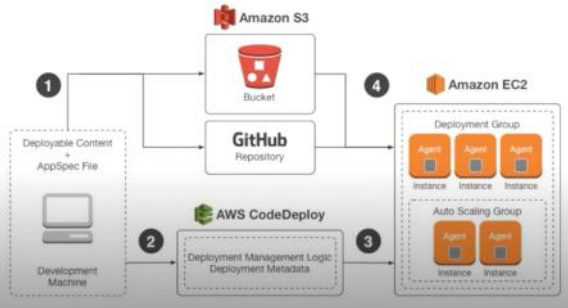
AWS CodeDeploy

Benefícios de usar o AWS CodeDeploy

5. Integração perfeita com serviços da AWS

O AWS CodeDeploy integra-se com outros serviços da AWS, como Amazon EC2, Elastic Load Balancing e Amazon S3. Essa integração permite que você aproveite todo o poder do ecossistema da AWS e crie aplicativos escaláveis e resilientes.

AWS CodeDeploy



Introdução Automatização de Tarefas na AWS

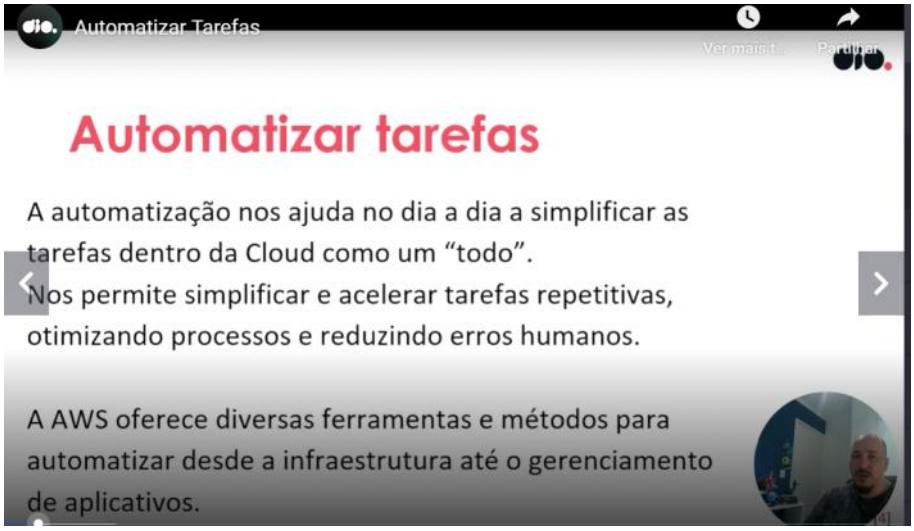
Ver mais t...

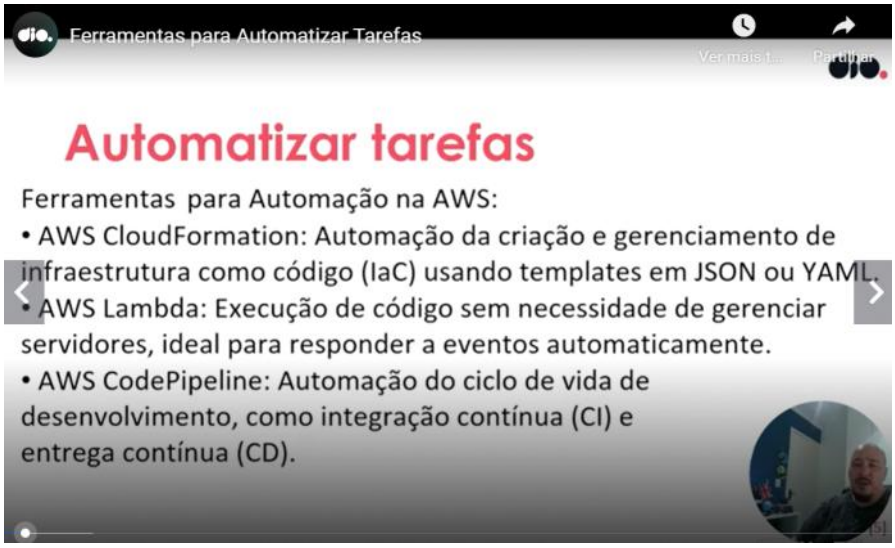
Partilh

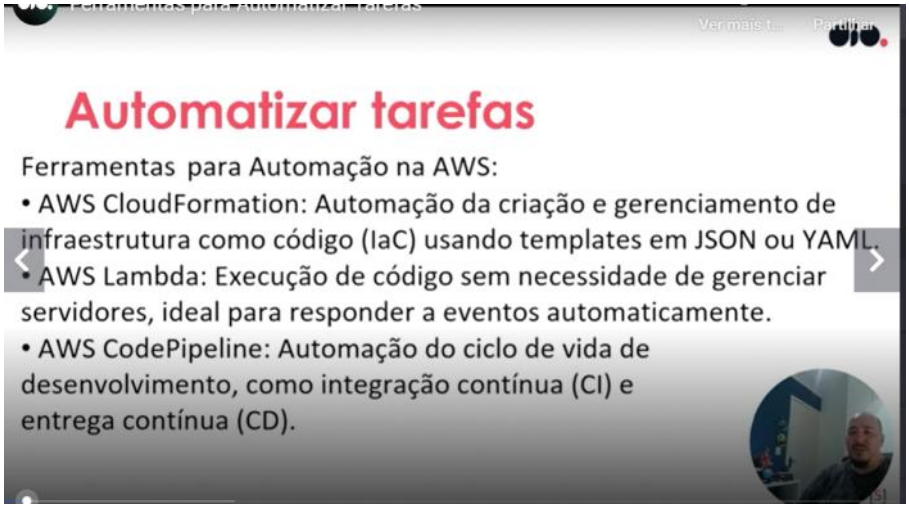
Conteúdo Programático

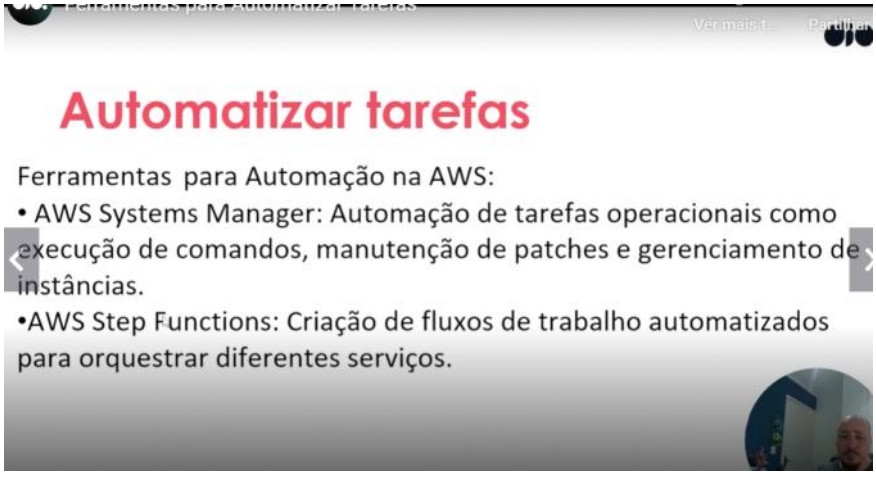
❑ **Módulo 11 Automação e DevOps na AWS**

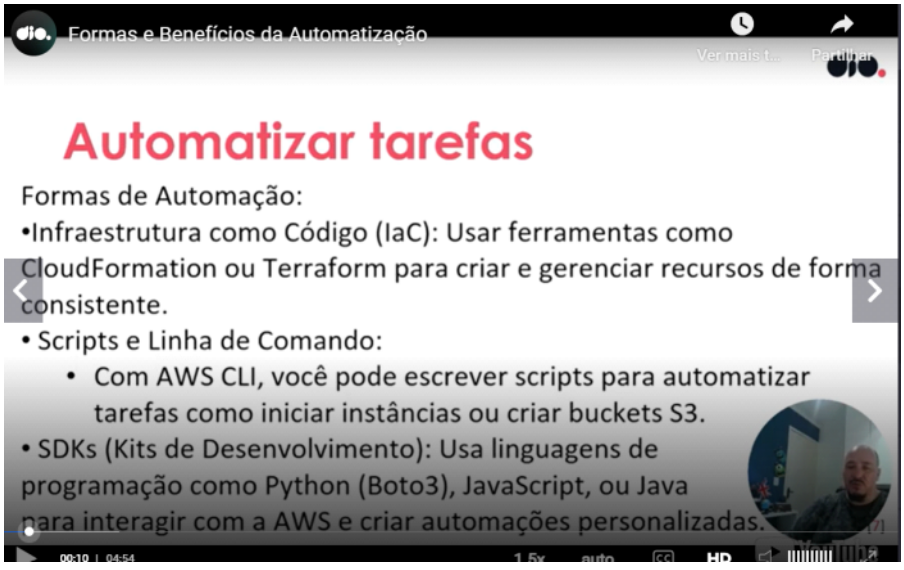
- **Como automatizar tarefas**
- Executando tarefas automatizadas com Lambda Function
- Criação de recursos com Terraform
- O que é DevOps
- Aplicando conceito de DevOps na AWS
- Ferramentas da AWS para DevOps

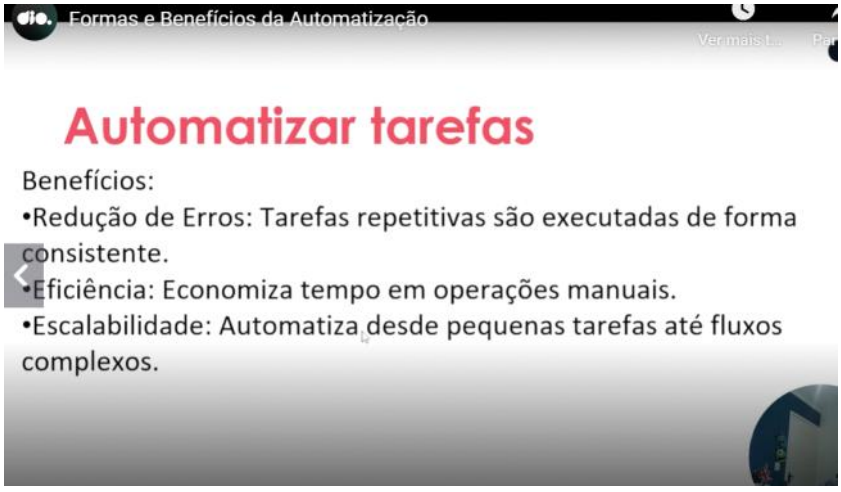

Automatizar Tarefas
Ver mais t...
Partilhar
Automatizar tarefas
A automatização nos ajuda no dia a dia a simplificar as tarefas dentro da Cloud como um "todo".
Nos permite simplificar e acelerar tarefas repetitivas, otimizando processos e reduzindo erros humanos.
A AWS oferece diversas ferramentas e métodos para automatizar desde a infraestrutura até o gerenciamento de aplicativos.

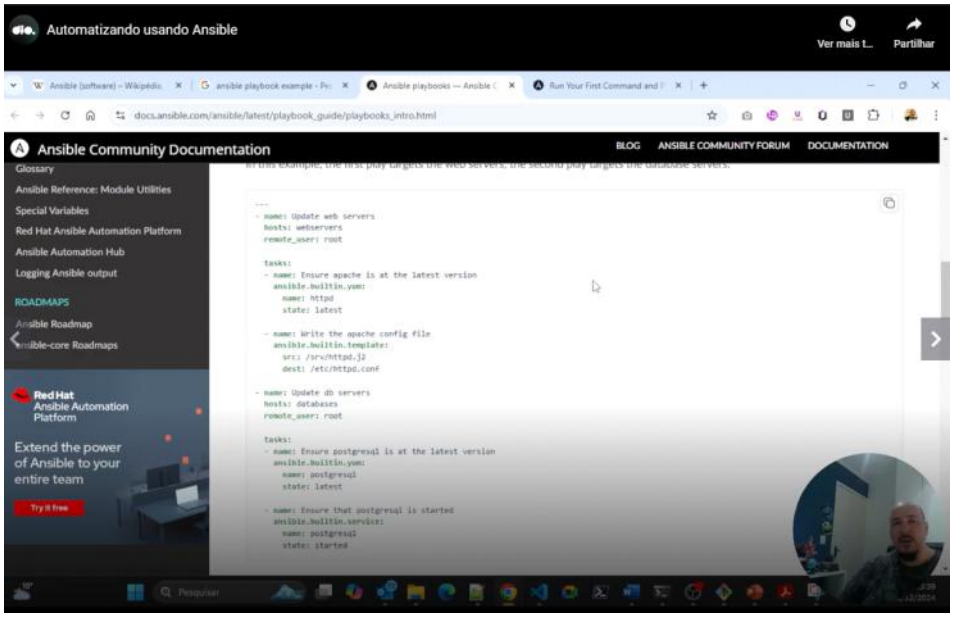

Ferramentas para Automatizar Tarefas
Ver mais t...
Partilhar
Automatizar tarefas
Ferramentas para Automação na AWS:
• AWS CloudFormation: Automação da criação e gerenciamento de infraestrutura como código (IaC) usando templates em JSON ou YAML.
• AWS Lambda: Execução de código sem necessidade de gerenciar servidores, ideal para responder a eventos automaticamente.
• AWS CodePipeline: Automação do ciclo de vida de desenvolvimento, como integração contínua (CI) e entrega contínua (CD).

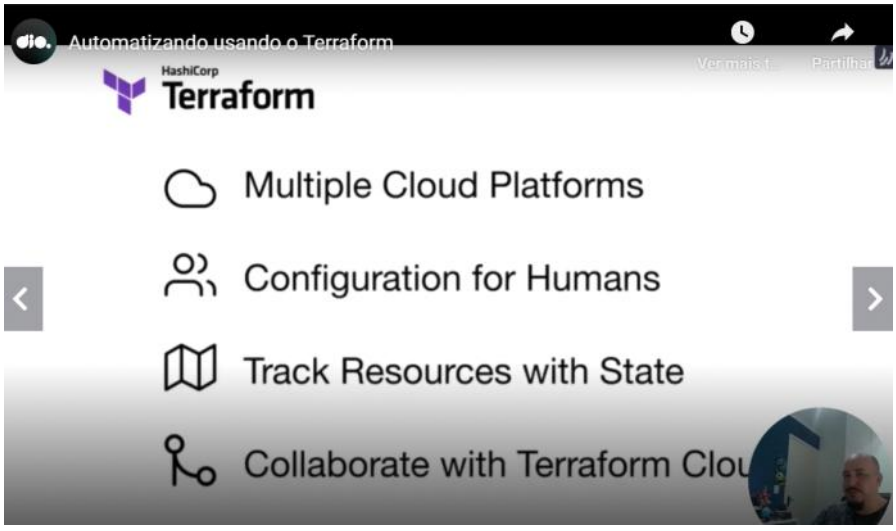

Ferramentas para Automatizar Tarefas
Ver mais t...
Partilhar
Automatizar tarefas
Ferramentas para Automação na AWS:
• AWS CloudFormation: Automação da criação e gerenciamento de infraestrutura como código (IaC) usando templates em JSON ou YAML.
• AWS Lambda: Execução de código sem necessidade de gerenciar servidores, ideal para responder a eventos automaticamente.
• AWS CodePipeline: Automação do ciclo de vida de desenvolvimento, como integração contínua (CI) e entrega contínua (CD).

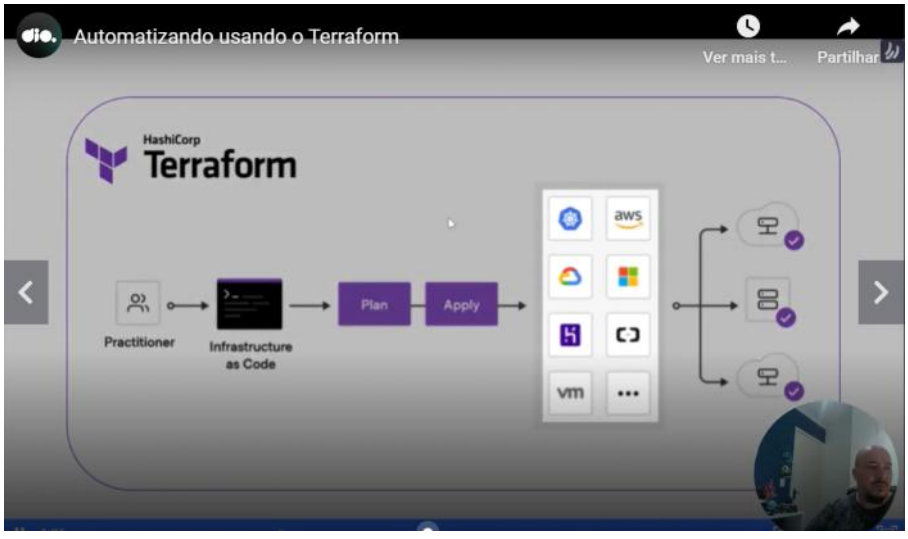

Ferramentas para Automatizar Tarefas
Ver mais t...
Partilhar
Automatizar tarefas
Ferramentas para Automação na AWS:
• AWS Systems Manager: Automação de tarefas operacionais como execução de comandos, manutenção de patches e gerenciamento de instâncias.
•AWS Step Functions: Criação de fluxos de trabalho automatizados para orquestrar diferentes serviços.


Formas e Benefícios da Automatização
Ver mais t...
Partilhar
Automatizar tarefas
Formas de Automação:
•Infraestrutura como Código (IaC): Usar ferramentas como CloudFormation ou Terraform para criar e gerenciar recursos de forma consistente.
• Scripts e Linha de Comando:
• Com AWS CLI, você pode escrever scripts para automatizar tarefas como iniciar instâncias ou criar buckets S3.
• SDKs (Kits de Desenvolvimento): Usa linguagens de programação como Python (Boto3), JavaScript, ou Java para interagir com a AWS e criar automações personalizadas.
00:10 | 04:54
1.5x
auto
HD


Formas e Benefícios da Automatização
Ver mais t...
Automatizar tarefas
Benefícios:
•Redução de Erros: Tarefas repetitivas são executadas de forma consistente.
•Eficiência: Economiza tempo em operações manuais.
•Escalabilidade: Automatiza desde pequenas tarefas até fluxos complexos.


Automatizando usando Ansible
Ver mais t...
Partilhar
Ansible Community Documentation
BLOG
ANSIBLE COMMUNITY FORUM
DOCUMENTATION
Glossary
Ansible Reference: Module Utilities
Special Variables
Red Hat Ansible Automation Platform
Ansible Automation Hub
Logging Ansible output
ROADMAPS
Ansible Roadmap
Ansible-core Roadmaps
Red Hat
Ansible Automation Platform
Extend the power of Ansible to your entire team
Try it free


Automatizando usando o Terraform
HashiCorp
Terraform
Multiple Cloud Platforms
Configuration for Humans
Track Resources with State
Collaborate with Terraform Clou


Automatizando usando o Terraform
Ver mais t...
Partilhar
HashiCorp
Terraform
Practitioner
Infrastructure as Code
Plan
Apply
aws
vm

Tarefas com Lambda e S3

HandsOn:

Projeto: Upload de Arquivos com Processamento e Registro no DynamoDB

Fluxo do Projeto:

1.O usuário faz upload de um arquivo (ex: CSV ou JSON) em um bucket S3.
2.Um evento no S3 dispara uma Lambda Function escrita em Python.
3.A Lambda processa o conteúdo do arquivo (ex: extrai informações) e gravar esses dados em uma tabela no DynamoDB.
4.Outra função Lambda irá consultar a tabela e expor os dados por meio de uma API Gateway.

Tarefas com Lambda e S3

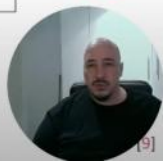
Nosso Case

Case real: Sistema de Processamento de Notas Fiscais

- O usuário envia um arquivo JSON contendo informações de uma nota fiscal (número, cliente, valor, data, etc.).
- A Lambda lê e valida o arquivo, depois grava no DynamoDB os campos relevantes.

Componentes AWS Utilizados:

- **Amazon S3:** Armazenamento dos arquivos.
- **AWS Lambda:** Processamento dos arquivos.
- **Amazon DynamoDB:** Armazenamento dos dados extraídos.
- **IAM:** Gerenciamento de permissões.
- (Opcional) **API Gateway:** Exposição de dados por API RESTful.



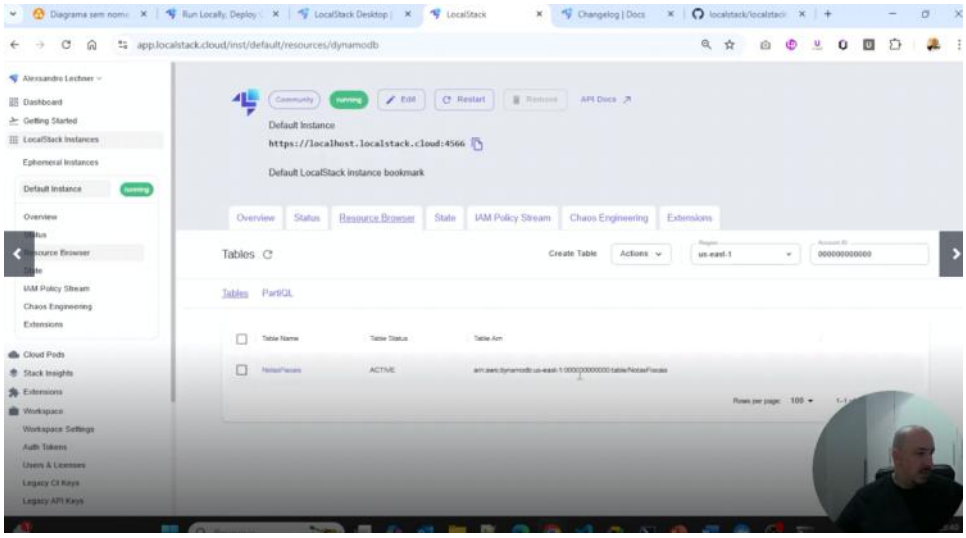
D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\HELP.txt - Notepad++

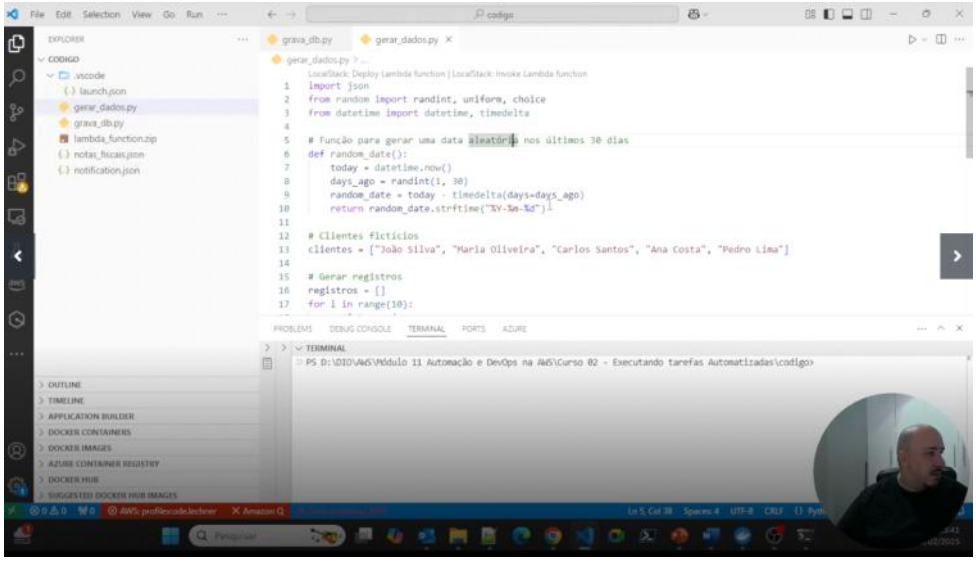
```
> aws configure
environment:
 - AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
 - AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
 - AWS_DEFAULT_REGION=us-east-1
 - AWS_DEFAULT_OUTPUT=json

# Atenção: as credenciais NÃO PRECISAM SER VÁLIDAS, mas devem ser definidas!

Tarefas para Configuração:
    1) Criar o bucket S3: Configure um bucket chamado notas-fiscais-upload
    2) Criar a tabela no DynamoDB: Nome NotasFiscais, com chave primária id.
    3) Criar uma Lambda Function: Configure as permissões para acesso ao S3 e DynamoDB.
    4) Criar o trigger do S3: Configure o bucket para disparar a Lambda ao fazer upload de arquivos

1) S3 - Criar o  notas-fiscais-upload:
    awslocal s3api create-bucket --bucket notas-fiscais-upload
    awslocal s3api list-buckets

2) DynamoDB - criar a tabela NotasFiscais:
        aws dynamodb create-table --endpoint-url=http://localhost:4566 --table-name NotasFiscais
        --attribute-definitions AttributeName=id,AttributeType=S --key-schema AttributeName=id,KeyT
        --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5

    Verificar tabela:
        aws dynamodb list-tables --endpoint-url=http://localhost:4566
```

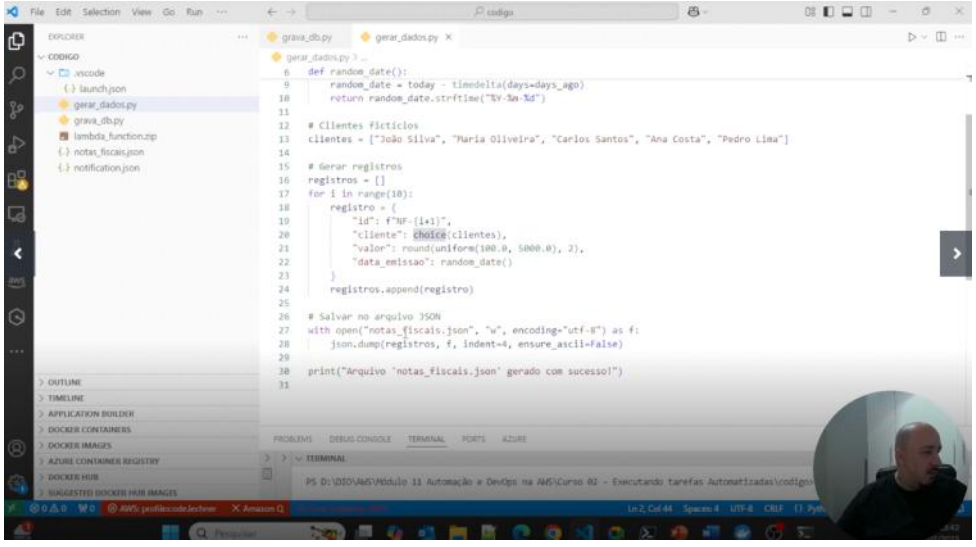
```
Tarefas para Configuração:
    1) Criar o bucket S3: Configure um bucket chamado notas-fiscais-upload
    2) Criar a tabela no DynamoDB: Nome NotasFiscais, com chave primária id.
    3) Criar uma Lambda Function: Configure as permissões para acesso ao S3 e DynamoDB.
    4) Criar o trigger do S3: Configure o bucket para disparar a Lambda ao fazer upload de arquivos

1) S3 - Criar o  notas-fiscais-upload:
    awslocal s3api create-bucket --bucket notas-fiscais-upload
    awslocal s3api list-buckets

2) DynamoDB - criar a tabela NotasFiscais:
        aws dynamodb create-table --endpoint-url=http://localhost:4566 --table-name NotasFiscais
        --attribute-definitions AttributeName=id,AttributeType=S --key-schema AttributeName=id,KeyType=HASH
        --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5

    Verificar tabela:
        aws dynamodb list-tables --endpoint-url=http://localhost:4566


> Download NoSQL Workbench for DynamoDB, para fazer querys ou outro a sua escolha

    https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.settingup.html


Script para gerar o arquivo notas_fiscais.json, na pasta codigo
    Para executar: python gerar_dados.py - ira gerar um arquivo fake
```

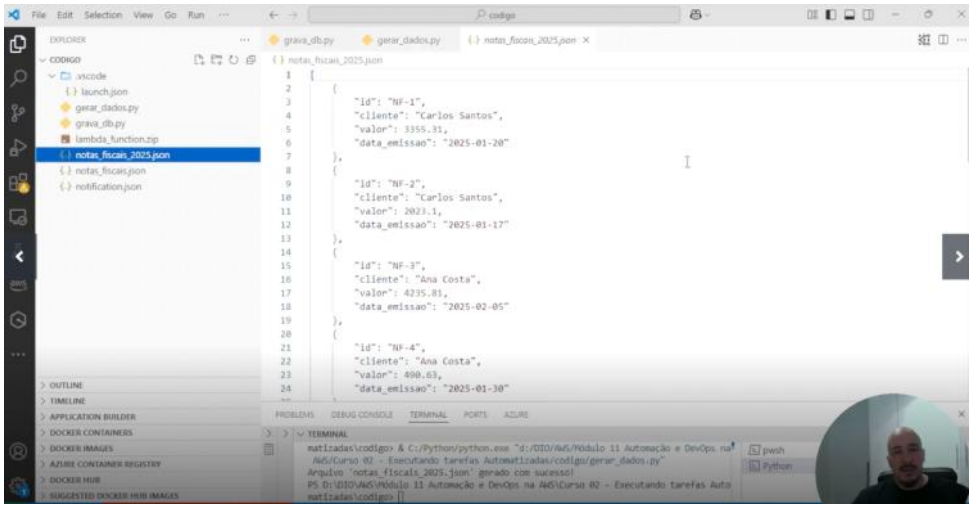
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

O carregamento de perfis pessoais e do sistema levou 3102ms.
PS C:\Users\alexs> aws configure
AWS Access Key ID [****************MPLE]: 000000000000
AWS Secret Access Key [****************EKEY]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [us-east-1]:
Default output format [json]:
PS C:\Users\alexs> aws configure
AWS Access Key ID [****************0000]:
AWS Secret Access Key [****************EKEY]:
Default region name [us-east-1]:
Default output format [json]:
PS C:\Users\alexs> awslocal
```

```
PS C:\Users\alexs> awslocal s3api create-bucket --bucket notas-fiscais-upload
{
    "Location": "/notas-fiscais-upload"
}
PS C:\Users\alexs> awslocal s3api list-buckets
{
    "Buckets": [
        {
            "Name": "notas-fiscais-upload",
            "CreationDate": "2025-02-07T02:38:15.000Z"
        }
    ],
    "Owner": {
        "DisplayName": "webfile",
        "ID": "75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a"
    },
    "Prefix": null

PS C:\Users\alexs> aws dynamodb create-table --endpoint-url=http://localhost:4566 --table-name NotasFiscais --attribute-definitions AttributeName=i
d,AttributeType=S --key-schema AttributeName=id,KeyType=HASH --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
```


app.localstack.cloud/inst/default/resources/dynamodb
Default Instance
https://localhost.localstack.cloud:4566
Default LocalStack instance bookmark
Overview
Status
Resource Browser
State
IAM Policy Stream
Chaos Engineering
Extensions
Tables
Create Table
Actions
us-east-1
000000000000
Table Name
Table Status
Table Arn
NotasFiscais
ACTIVE


gerar_dados.py
import json
from random import randint, uniform, choice
from datetime import datetime, timedelta
# Função para gerar uma data aleatória nos últimos 30 dias
def random_date():
today = datetime.now()
days_ago = randint(1, 30)
random_date = today - timedelta(days=days_ago)
return random_date.strftime("%Y-%m-%d")
# Clientes fictícios
clientes = ["João Silva", "Maria Oliveira", "Carlos Santos", "Ana Costa", "Pedro Lima"]
# Gerar registros
registros = []
for i in range(10):
TERMINAL
PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo>


gerar_dados.py
def random_date():
random_date = today - timedelta(days=days_ago)
return random_date.strftime("%Y-%m-%d")
# Clientes fictícios
clientes = ["João Silva", "Maria Oliveira", "Carlos Santos", "Ana Costa", "Pedro Lima"]
# Gerar registros
registros = []
for i in range(10):
registro = {
"id": f"NF-{i+1}",
"cliente": choice(clientes),
"valor": round(uniform(100.0, 5000.0), 2),
"data_emissao": random_date()
}
registros.append(registro)
# Salvar no arquivo JSON
with open("notas_fiscais.json", "w", encoding="utf-8") as f:
json.dump(registros, f, indent=4, ensure_ascii=False)
print("Arquivo 'notas_fiscais.json' gerado com sucesso!")


File Edit Selection View Go Run
EXPLORER
CODIGO
.vscode
launch.json
gerar_dados.py
grava_db.py
lambda_function.zip
notas_fiscais_2025.json
notas_fiscais.json
notification.json
grava_db.py
gerar_dados.py
notas_fiscais_2025.json
{ "id": "NF-1", "cliente": "Carlos Santos", "valor": 3355.31, "data_emissao": "2025-01-20" },
{ "id": "NF-2", "cliente": "Carlos Santos", "valor": 2023.1, "data_emissao": "2025-01-17" },
{ "id": "NF-3", "cliente": "Ana Costa", "valor": 4235.81, "data_emissao": "2025-02-05" },
{ "id": "NF-4", "cliente": "Ana Costa", "valor": 490.63, "data_emissao": "2025-01-30"
OUTLINE
TIMELINE
APPLICATION BUILDER
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
PROBLEMS DEBUG CONSOLE TERMINAL PORTS AZURE
TERMINAL
Arquivo 'notas_fiscais_2025.json' gerado com sucesso!
PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Auto
pwsh
Python

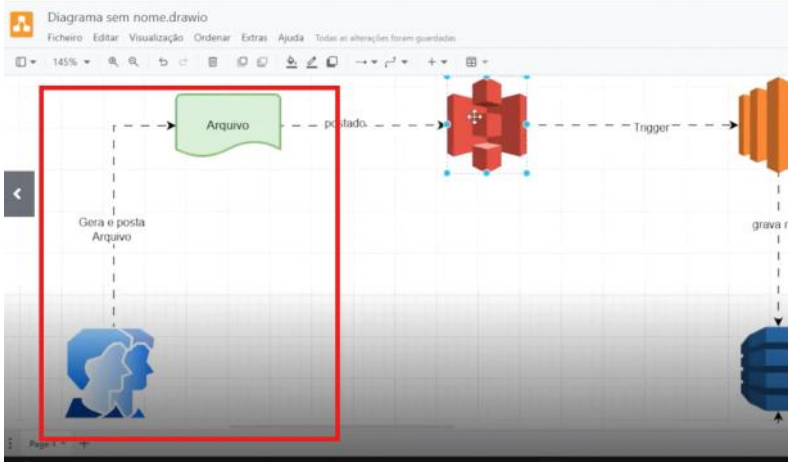
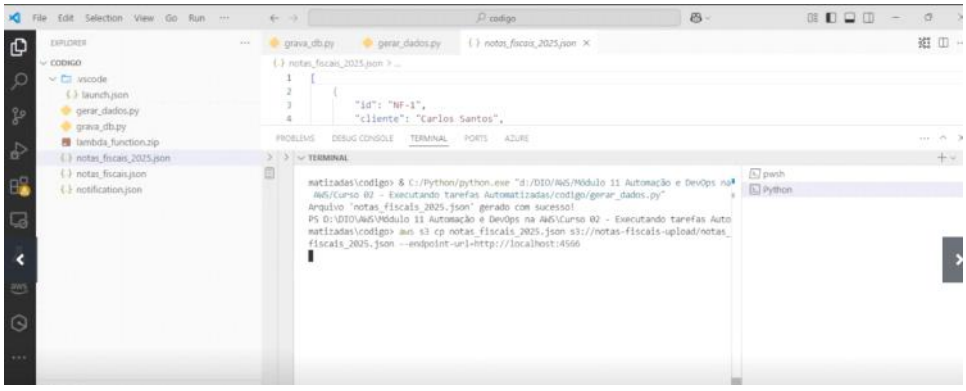


Diagrama sem nome.drawio
Ficheiro Editar Visualização Ordenar Extras Ajuda Todas as alterações foram guardadas
145%
Arquivo
postado
Trigger
Gera e posta
Arquivo
grava r
Page-1


File Edit Selection View Go Run
EXPLORER
CODIGO
.vscode
launch.json
gerar_dados.py
grava_db.py
lambda_function.zip
notas_fiscais_2025.json
notas_fiscais.json
notification.json
grava_db.py
gerar_dados.py
notas_fiscais_2025.json
{ "id": "NF-1", "cliente": "Carlos Santos",
PROBLEMS DEBUG CONSOLE TERMINAL PORTS AZURE
TERMINAL
Arquivo 'notas_fiscais_2025.json' gerado com sucesso!
PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Auto
matizadas\codigo> aws s3 cp notas_fiscais_2025.json s3://notas-fiscais-upload/notas_fiscais_2025.json --endpoint-url=http://localhost:4566
pwsh
Python

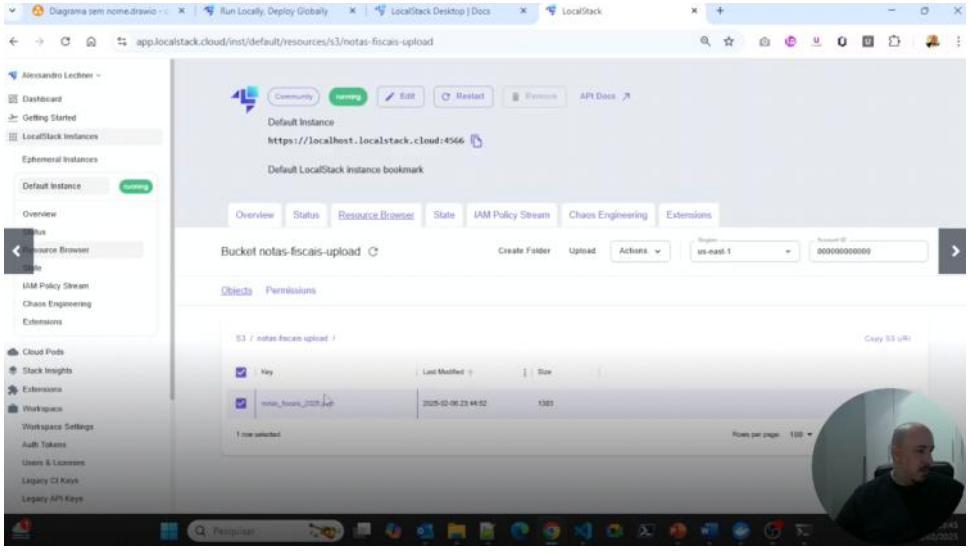
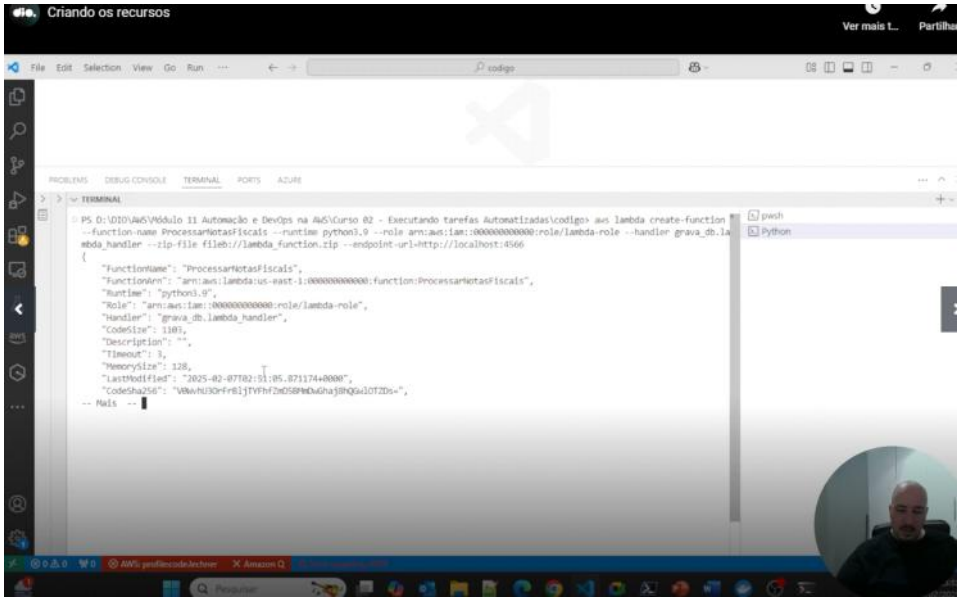
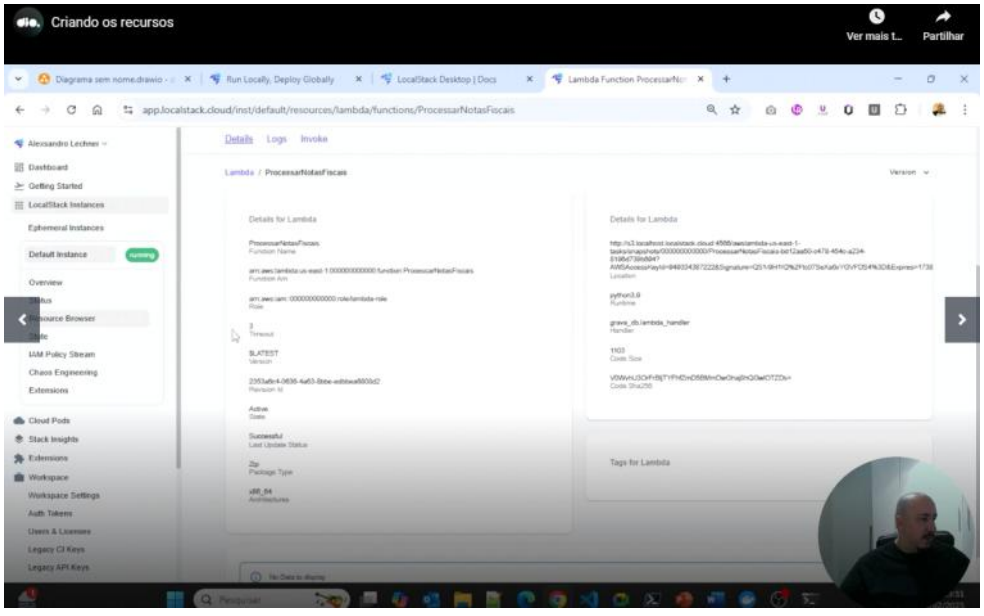


Diagrama sem nome.drawio
Run Locally, Deploy Globally
LocalStack Desktop | Docs
LocalStack
app.localstack.cloud/inst/default/resources/s3/notas-fiscais-upload
Alessandro Lechner
Dashboard
Getting Started
LocalStack Instances
Ephemeral Instances
Default Instance
Overview
Resource Browser
IAM Policy Stream
Chaos Engineering
Extensions
Cloud Pods
Stack Insights
Workspace
Workspace Settings
Auth Tokens
Users & Licenses
Legacy CI Keys
Legacy API Keys
Community
Edit
Restart
Remove
API Docs
Default Instance
https://localhost.localstack.cloud:4566
Default LocalStack instance bookmark
Overview
Status
Resource Browser
State
IAM Policy Stream
Chaos Engineering
Extensions
Bucket notas-fiscais-upload
Create Folder
Upload
Actions
us-east-1
000000000000
Objects
Permissions
S3 / notas-fiscais-upload /
Copy S3 URI
Key
Last Modified
Size
notas_fiscais_2025.json
2025-02-06 23:44:52
1383
1 row selected
Rows per page: 100

```
3) Lambda Function:
aws lambda create-function \
    --function-name ProcessarNotasFiscais \
    --runtime python3.9 \
    --role arn:aws:iam::000000000000:role/lambda-role \
    --handler grava_db.lambda_handler \
    --zip-file fileb://lambda_function.zip \
    --endpoint-url=http://localhost:4566

Configurar o trigger S3
    aws lambda add-permission \
    --function-name ProcessarNotasFiscais \
    --statement-id s3-trigger \
    --action "lambda:InvokeFunction" \
    --principal s3.amazonaws.com \
    --source-arn arn:aws:s3:::notas-fiscais-upload \
```

```
3) Lambda Function:
aws lambda create-function --function-name ProcessarNotasFiscais --runtime python3.9 --role
arn:aws:iam::000000000000:role/lambda-role --handler grava_db.lambda_handler --zip-file
fileb://lambda_function.zip --endpoint-url=http://localhost:4566
```





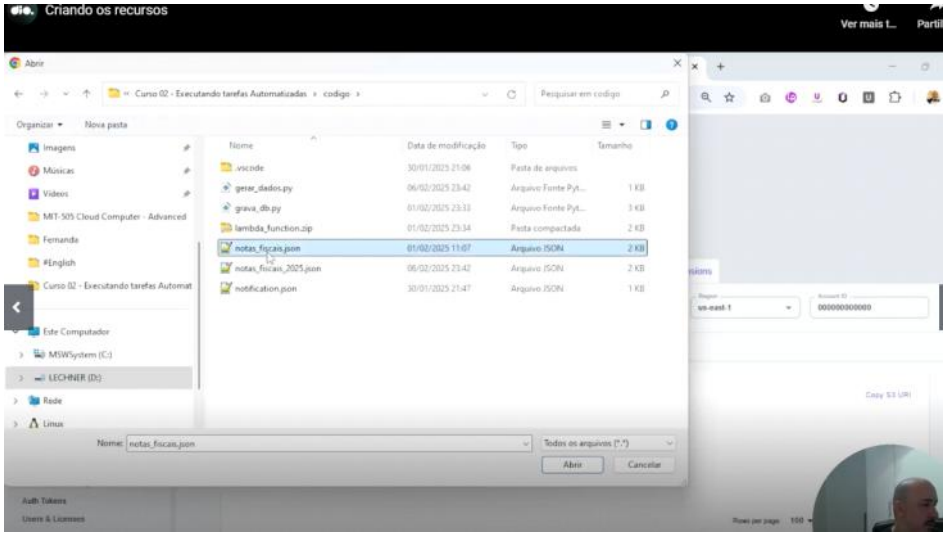

```
Configurar o trigger S3
    aws lambda add-permission --function-name ProcessarNotasFiscais --statement-id s3-trigger --action
    "lambda:InvokeFunction" --principal s3.amazonaws.com --source-arn arn:aws:s3:::notas-fiscais-upload
    --endpoint-url=http://localhost:4566
```

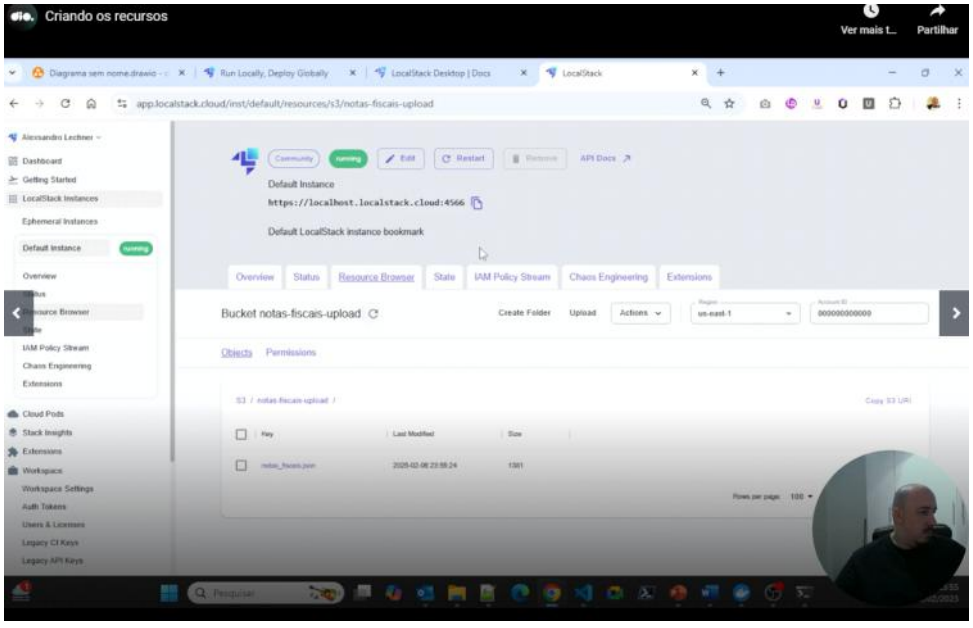
```
Envie um arquivo JSON para o bucket S3:
aws s3 cp notas_fiscais.json s3://notas-fiscais-upload --endpoint-url=http://localhost:4566

Resumo:
Empacote o código em .zip.
Crie a função Lambda no Localstack.
Configure o bucket S3 para disparar a Lambda.
Envie um arquivo para o S3.
Verifique os logs e os dados inseridos no DynamoDB.

Role ficticia: arn:aws:iam::000000000000:role/lambda-role
```

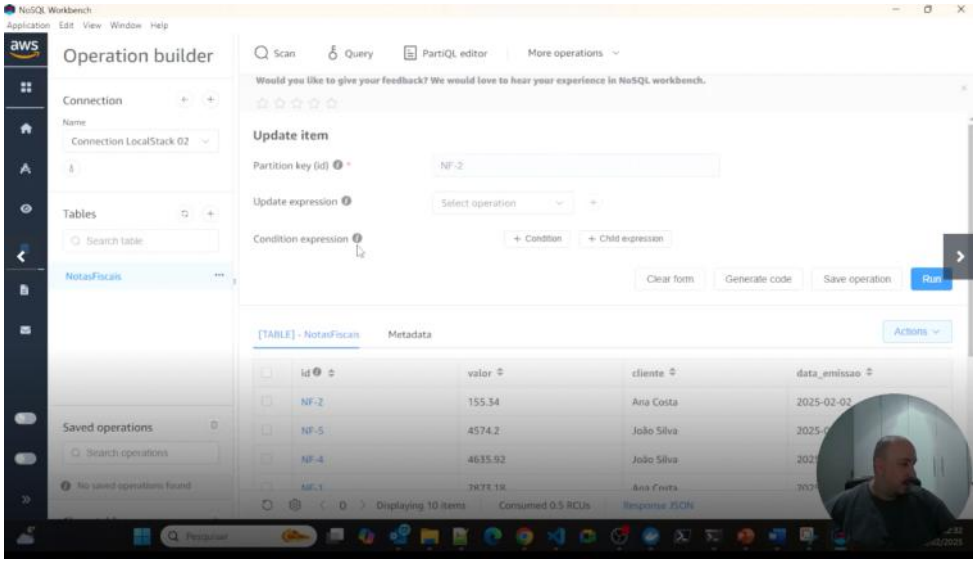
Corrija essa parte do código pois precisa criar o evento que "se cair qualquer arquivo .json no bucket acione o evento" - o código abaixo não tá correto. Ta lendo o nome especifico de um arquivo ao entrar no bucket

```
Configurar o evento de trigger no bucket S3:
    aws s3api put-bucket-notification-configuration --bucket notas-fiscais-upload --notification-configuration
    file://notas_fiscais_2025.json --endpoint-url=http://localhost:4566
```

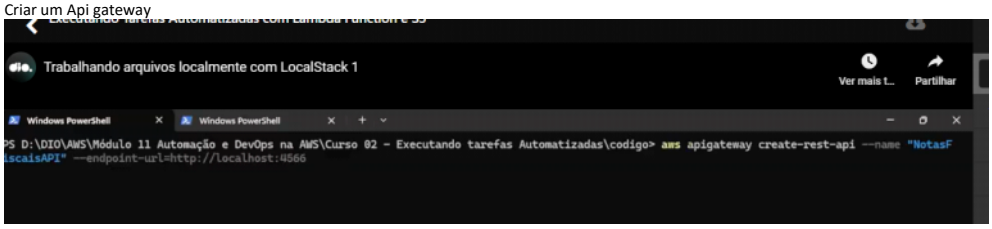
```
D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo\notification.json - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
notification.json
{
    "LambdaFunctionConfigurations": [
        {
            "LambdaFunctionArn": "arn:aws:lambda:us-east-1:000000000000:function:ProcessarNotasFiscais",
            "Events": ["s3:ObjectCreated:*"],
            "Filter": {
                "Key": {
                    "FilterRules": [
                        {
                            "Name": "suffix",
                            "Value": ".json"
                        }
                    ]
                }
            }
        }
    ]
}
```

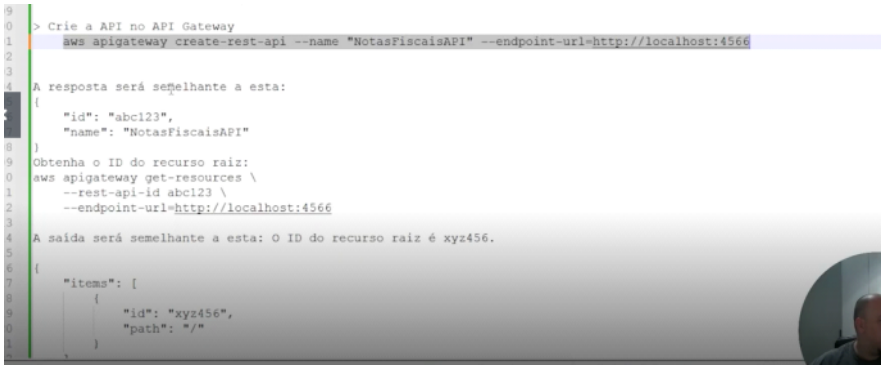


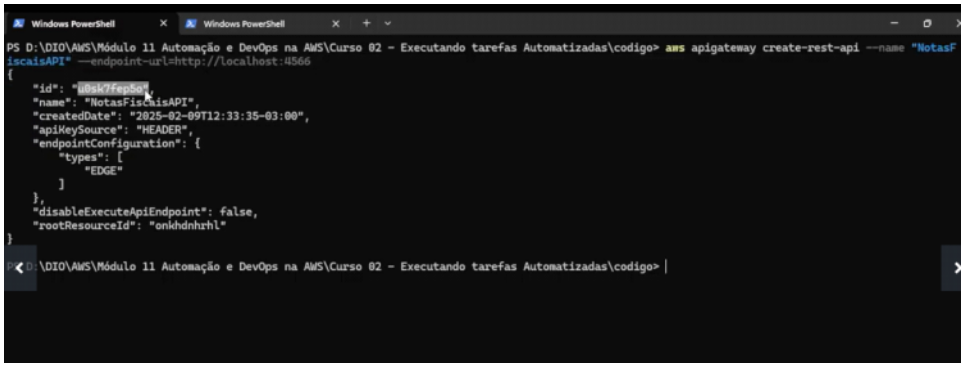


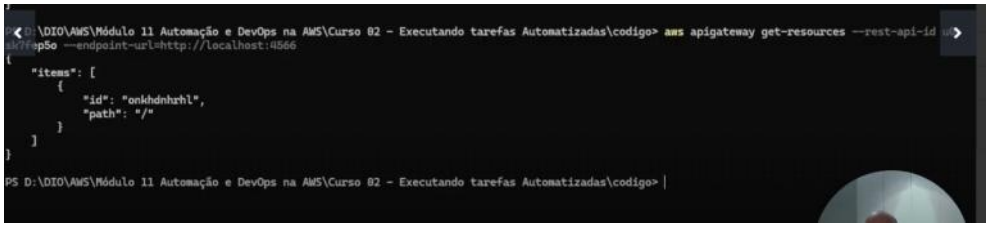


Criar um Api gateway









```
Obtenha o ID do recurso raiz:
    aws apigateway get-resources --rest-api-id u0sk7fep5o --endpoint-url=http://localhost:4566

A saída será semelhante a esta: O ID do recurso raiz é xyz456.

{
    "items": [
        {
            "id": "xyz456",
            "path": "/"
        }
    ]
}

Crie o recurso /notas na API:
    aws apigateway create-resource --rest-api-id u0sk7fep5o --parent-id onkhdnhrhl --path-part "notas" --endpoint-url=http

A saída será semelhante a:O ID do recurso é mno789.
{
    "id": "mno789",
    "path": "/notas"
}

Configure o método HTTP (POST ou GET): o recurso /notas, vinculando-o à sua função Lambda:
```

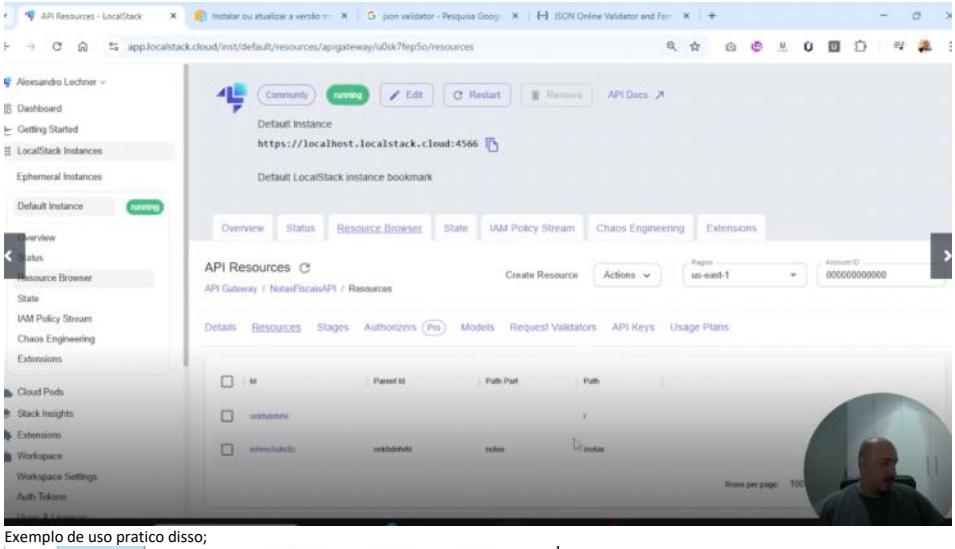
```
PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo> aws apigateway create-resource --rest-api-id
u0sk7fep5o --parent-id onkhdnhrhl --path-part "notas" --endpoint-url=http://localhost:4566
{
    "id": "mhmc5ukc8z",
    "parentId": "onkhdnhrhl",
    "pathPart": "notas",
    "path": "/notas"
}

PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo>
```

```
PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo> aws apigateway put-method --rest-api-id u0sk7
fep5o --resource-id mhmc5ukc8z --http-method POST --authorization-type "NONE" --endpoint-url=http://localhost:4566
{
    "httpMethod": "POST",
    "authorizationType": "NONE",
    "apiKeyRequired": false
}

PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo> aws apigateway put-integration --rest-api-id
u0sk7fep5o --resource-id mhmc5ukc8z --http-method POST --type AWS_PROXY --integration-http-method POST --uri "arn:aws:apigateway:us-east-1:lambda:p
ath/2015-03-31/functions/arn:aws:lambda:us-east-1:000000000000:function:ProcessarNotasFiscais/invocations" --endpoint-url=http://localhost:4566
{
    "type": "AWS_PROXY",
    "httpMethod": "POST",
    "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/arn:aws:lambda:us-east-1:000000000000:function:ProcessarNotasFiscais/invo
cations",
    "passthroughBehavior": "WHEN_NO_MATCH",
    "timeoutInMillis": 29000,
    "cacheNamespace": "mhmc5ukc8z",
    "cacheKeyParameters": []
}

PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo>
```

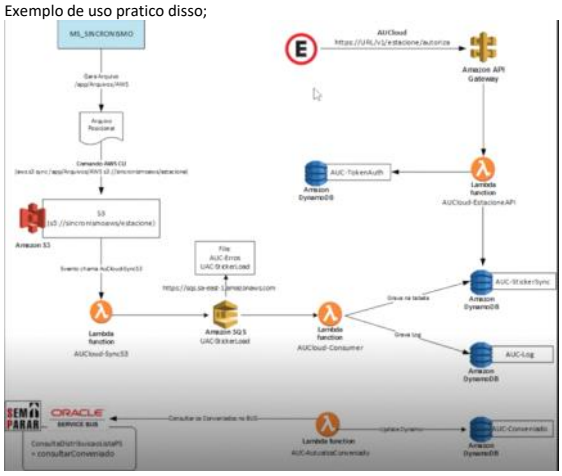
```
PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo> aws lambda add-permission --function-name Pro
cessarNotasFiscais --statement-id apigateway-access --action "lambda:InvokeFunction" --principal apigateway.amazonaws.com --source-arn "arn:aws:exe
cute-api:us-east-1:000000000000:u0sk7fep5o/*/POST/notas" --endpoint-url=http://localhost:4566
```

```
PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo> aws lambda add-permission --function-name Pro
cessarNotasFiscais --statement-id apigateway-access --action "lambda:InvokeFunction" --principal apigateway.amazonaws.com --source-arn "arn:aws:exe
cute-api:us-east-1:000000000000:u0sk7fep5o/*/POST/notas" --endpoint-url=http://localhost:4566
{
    "Statement": "{\"Sid\": \"apigateway-access\", \"Effect\": \"Allow\", \"Action\": \"lambda:InvokeFunction\", \"Resource\": \"arn:aws:lambda:us-
east-1:000000000000:function:ProcessarNotasFiscais\", \"Principal\": {\"Service\": \"apigateway.amazonaws.com\"}, \"Condition\": {\"
WS:SourceArn\": \"arn:aws:execute-api:us-east-1:000000000000:u0sk7fep5o/*/POST/notas\"}}"
}

PS D:\DIO\AWS\Módulo 11 Automação e DevOps na AWS\Curso 02 - Executando tarefas Automatizadas\codigo>
```

```
Configure o método HTTP (POST ou GET): o recurso /notas, vinculando-o à sua função Lambda:
    aws apigateway put-method --rest-api-id u0sk7fep5o --resource-id mhmc5ukc8z --http-method POST --authorization-type "NONE" --endp

Integre o método com a Lambda
    aws apigateway put-integration --rest-api-id u0sk7fep5o --resource-id mhmc5ukc8z --http-method POST --type AWS_PROXY --integrati


Conceder permissão à API para invocar a Lambda
    aws lambda add-permission --function-name ProcessarNotasFiscais --statement-id apigateway-access --action "lambda:InvokeFunction"

Implemente a API
    aws apigateway create-deployment --rest-api-id u0sk7fep5o --stage-name dev --endpoint-url=http://localhost:4566

Teste a API:
Invoke-RestMethod -Uri "http://localhost:4566/restapis/u0sk7fep5o/dev/_user_request_/notas"
```



Exemplo de uso pratico disso;



Recursos com Terraform

O que podemos fazer com Terraform na AWS:

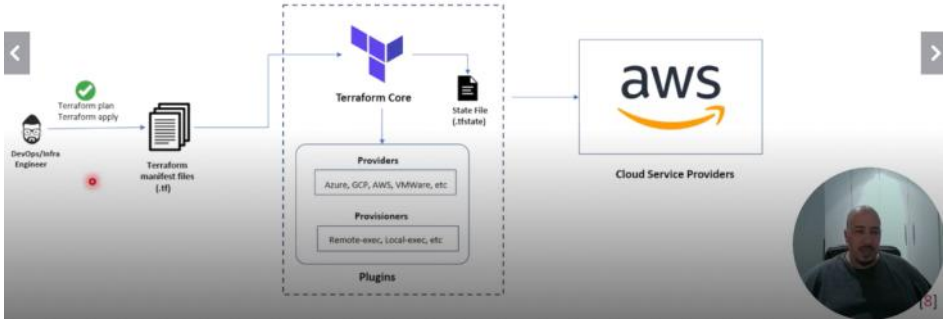
- Criar recursos: VPCs, instâncias EC2, buckets S3, funções Lambda, RDS, etc.
- Automatizar a infraestrutura: Executar mudanças em grande escala com scripts reutilizáveis.
- Gerenciar dependências: Configurar relações entre recursos (ex.: o bucket S3 depende de uma IAM Role).
- Prover versões: Aplicar, atualizar ou reverter mudanças com segurança.

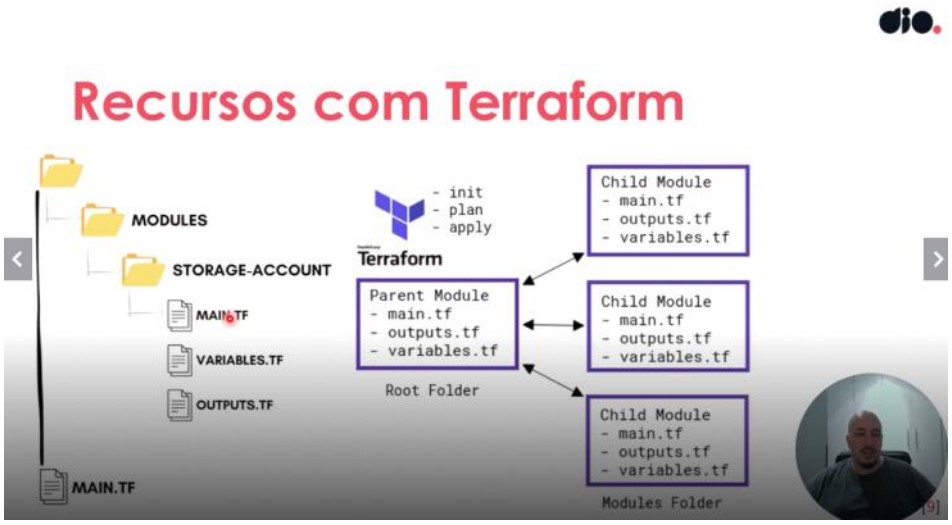
Recursos com Terraform

Imagine criar e atualizar recursos através de comandos, sem ter que acessar o painel da AWS.
Ou melhor, ele possibilita atualizarmos todos nossos recursos na cloud, como por exemplo
Atualizar todos os EC2, Database, Storages, conseguimos atualizar o size, capacidade, temos total poder sobre os recursos.

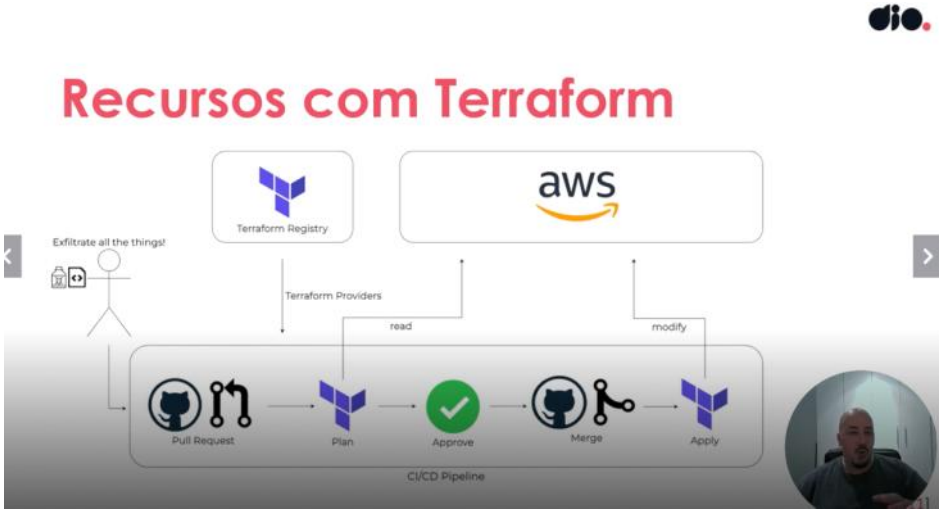
Recursos com Terraform

Terraform Architecture




Recursos com Terraform
MODULES
STORAGE-ACCOUNT
MAIN.TF
VARIABLES.TF
OUTPUTS.TF
MAIN.TF
- init
- plan
- apply
Terraform
Parent Module
- main.tf
- outputs.tf
- variables.tf
Root Folder
Child Module
- main.tf
- outputs.tf
- variables.tf
Child Module
- main.tf
- outputs.tf
- variables.tf
Child Module
- main.tf
- outputs.tf
- variables.tf
Modules Folder

Recursos com Terraform
terraform.state
terraform.tfvars
terraform-provider.tf
terraform-instances.tf
Terraform
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "3.33.0"
}
}
}
provider "aws" {
region = "us-west-2"
access_key = "ExampleKey"
secret_key = "SecretExampleKey"
}
resource "aws_instance" "my_ec2" {
ami = "ami-05b622b5fa0269787"
instance_type = "t2.micro"
}

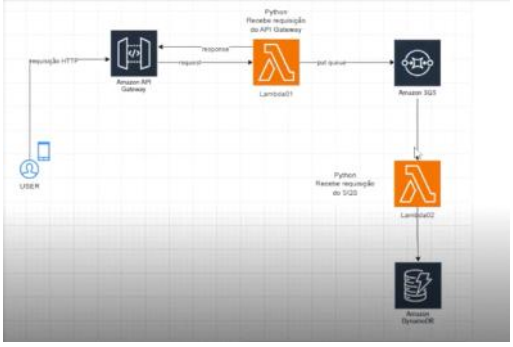

Recursos com Terraform
Terraform Registry
aws
Exfiltrate all the things!
Terraform Providers
read
modify
Pull Request
Plan
Approve
Merge
Apply
CI/CD Pipeline

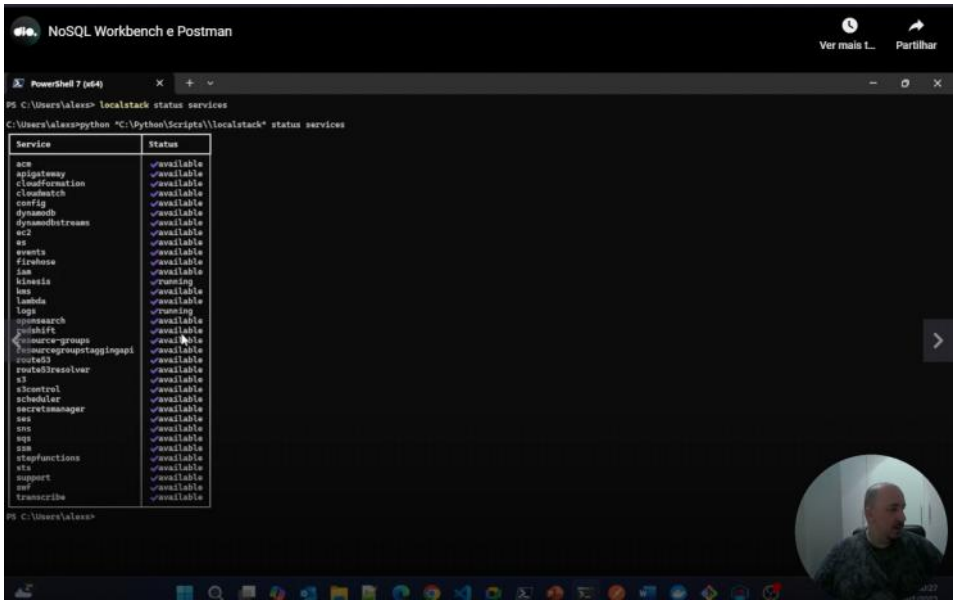
Recursos com Terraform

AWS Local com LocalStack:

Projeto OpenSource, que nos ajuda a simular localmente a AWS.
Criado em 2016 pela Atlantis Software, somente em 2017 o projeto se tornou publico.
A Localstack oferece uma versão open-source, gratuita e com suporte a vários serviços.

Também possui uma versão Pro e Enterprise, que oferece recursos avançados, como suporte a endpoints de rede personalizados.



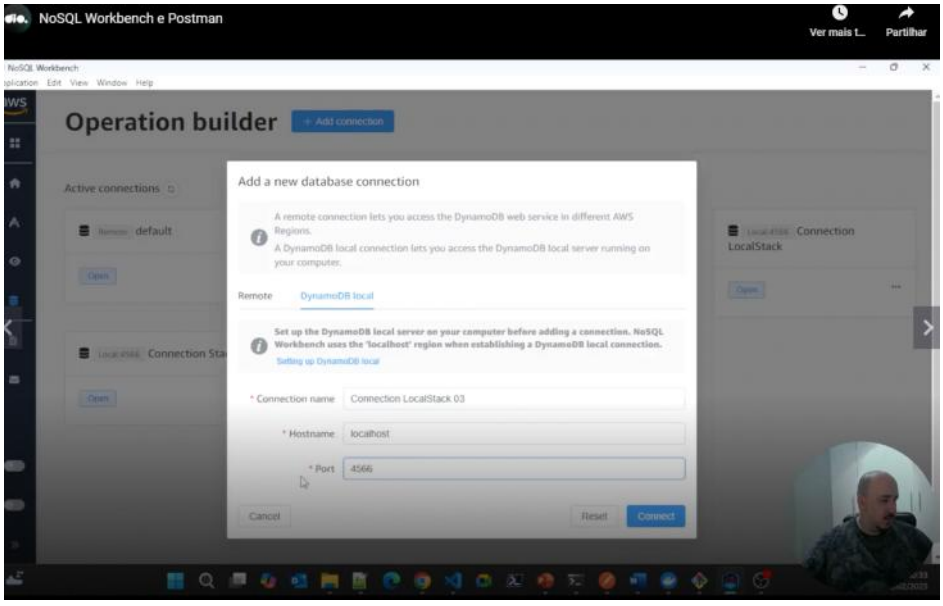


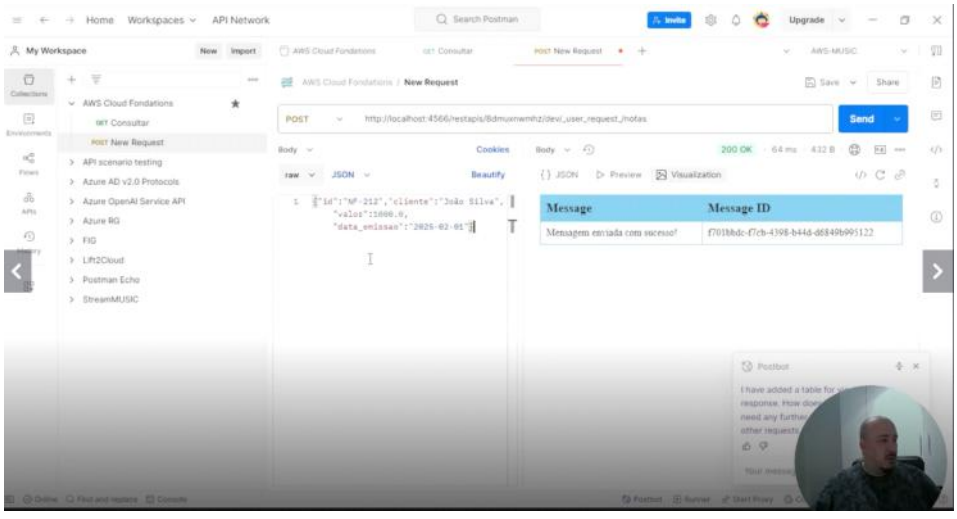
Recursos com Terraform

AWS Local com LocalStack:

Subir o nosso ambiente e criar os recursos em nossa LocalStack.

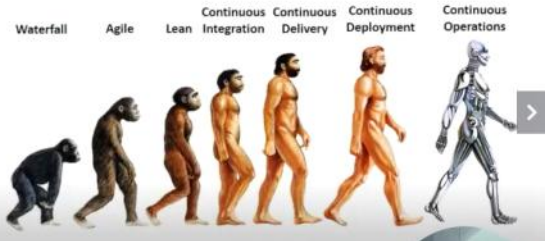
- Criar ambiente LocalStack
- Criar infraestrutura com código (IaC)
- Instalar banco NoSQL Workbench e acessar nosso BD
- Executar chamada com Postman
- Destruir e recriar no final com IaC



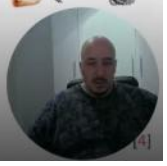


O que é DevOps?

Waterfall, Agile e Lean, que são metodologias antigas de desenvolvimento de software



Esta figura ilustra um pouco a evolução do processo de desenvolvimento de software.



O que é DevOps?

Mas como chegamos ao DevOps?
Depois de percorrer todos estes processos e amadurecer o nosso desenvolvimento de software, surgiu uma nova metodologia de desenvolvimento que tem como objetivo integrar as equipes de desenvolvimento (Dev) e operações (Ops) de uma organização para permitir a entrega **rápida e contínua** de software de alta qualidade.

O DevOps baseia-se numa equipa centrada na velocidade de "entrega e estabilidade" da Infra-estrutura, com os seguintes objectivos:

- ✓ Entrega rápida do Código Concluído à Produção
- ✓ Mínimo de falhas de produção
- ✓ Recuperação Imediata de Falhas

O que é DevOps?

50% Automação

50% Operação

DevOps é uma metodologia de desenvolvimento de software que tem como objetivo integrar as equipes de desenvolvimento (Dev) e operações (Ops) de uma organização para permitir a entrega rápida e contínua de software de alta qualidade.

O que é DevOps?

O DevOps é importante na implementação de Software e processos

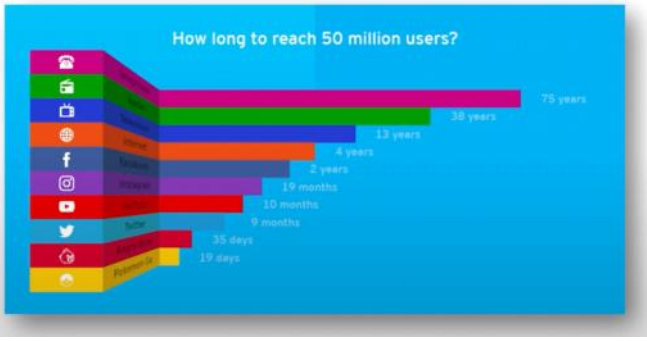
O DevOps trabalha com a entrega do software com a agilidade, de forma que o cliente não seja impactado.

O que é DevOps?

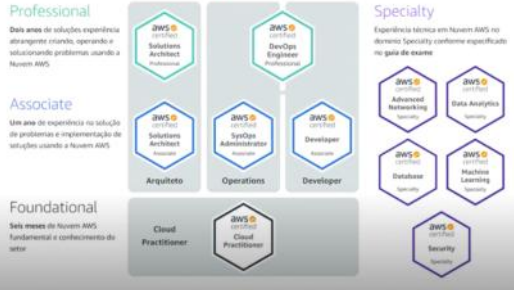
5 Princípios Chave do DevOps

- Colaboração
- Automação
- Melhoria continua
- Ação centrada no cliente
- Crie com o fim em mente

O que é DevOps?

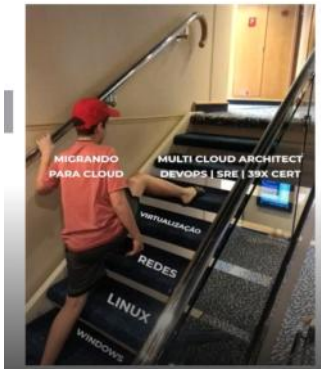


O que é DevOps?



Amazon Web Services, também conhecido como AWS, é uma plataforma de serviços de computação em nuvem, que formam uma plataforma de computação na nuvem

O que é DevOps?



É claro que a imagem ao lado é uma brincadeira, mas acontece muito 😊 !

É necessário ir por etapas, entender e reconhecer o caminho, mapear e procurar entender como funcionam as coisas.

O profissional DevOps é algo realmente novo e que tem tido grande demanda no mercado de trabalho.

O que é DevOps?

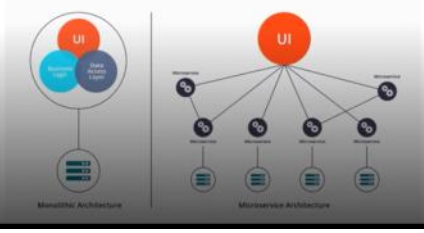
Quem pode se tornar um engenheiro de DevOps?

Se você é um desenvolvedor, uma pessoa de operações ou alguém de QA, ou até mesmo um novato. Você pode se tornar um engenheiro de DevOps.

Mas é bom saber que será uma longa jornada!

O que é DevOps?

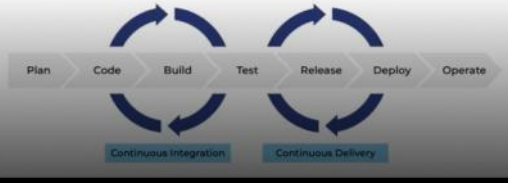
O profissional DevOps não é responsável por mudar toda a estrutura da empresa, a motivação para a empresa mudar e migrar vem das dores que ela tem. Ou seja, se aquela estrutura não lhe custa, conseguem produzir com uma arquitetura assim, não tem porque mudar.

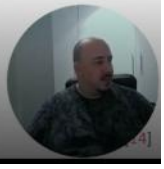


O que é DevOps?

CI (Continuous Integration): é usado durante a fase de construção e teste onde os desenvolvedores mesclam o seu código em um repositório várias vezes ao dia.

CD (Continuous Delivery): é usado assim que as alterações são confirmadas e que já passaram pela Integração Contínua (CI) são empacotados automaticamente e prontos para serem implementados no ambiente.





DevOps na AWS

O que é DevOps?

- DevOps é um conjunto de práticas que une desenvolvimento (Dev) e operações (Ops) para automatizar e melhorar processos.
- Objetivo: entregar software com mais rapidez, qualidade e confiabilidade.
- A AWS oferece ferramentas para cada etapa do ciclo de vida DevOps.

DevOps na AWS

Principais Ferramentas DevOps da AWS

Ferramentas AWS	Uso
AWS CodeCommit	Repositório Git gerenciado
AWS CodeBuild	Compilação e testes automatizados
AWS CodeDeploy	Automação da implantação
AWS CodePipeline	Orquestração do pipeline CI/CD
AWS CloudFormation	Infraestrutura como Código (IaC)
Amazon CloudWatch	Monitoramento e logs
AWS IAM	Gestão de permissões e segurança

https://aws.amazon.com/pt/devops/

DevOps na AWS

Pipeline CI/CD na AWS

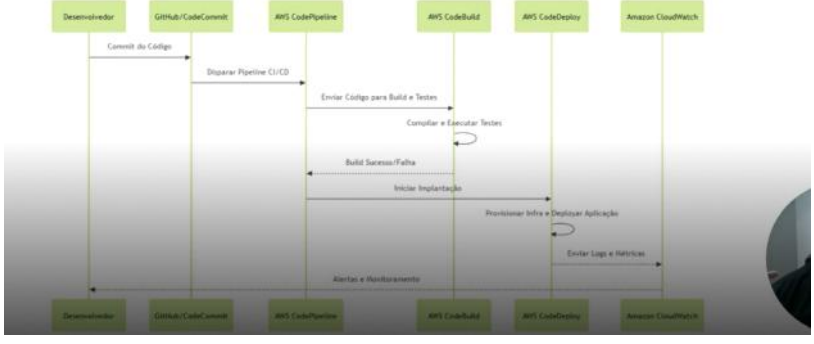
- CI/CD significa Continuous Integration (CI) e Continuous Deployment (CD)
- CI: Automatiza testes e builds sempre que há uma mudança no código
- CD: Automatiza a entrega do código em produção
- Benefícios: Rapidez, consistência e menos erros manuais.

Etapas típicas de um pipeline CI/CD:

> Commit do código → GitHub / CodeCommit
> Build e Testes → CodeBuild
> Implantação → CodeDeploy
> Monitoramento → CloudWatch

DevOps na AWS

Principais Ferramentas DevOps da AWS



DevOps na AWS

Desenvolvimento de Software com CI/CD na AWS

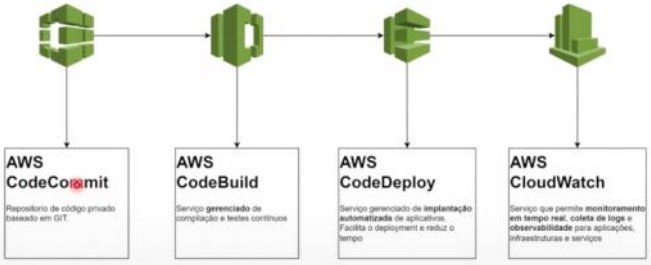
No desenvolvimento tradicional de software, um pipeline CI/CD permite que os times implementem mudanças rapidamente, garantindo qualidade e confiabilidade.

Etapas do pipeline CI/CD na AWS

1. **Commit do código** → O desenvolvedor faz um push do código para o **AWS CodeCommit**
2. **Build e Testes** → O **AWS CodeBuild** compila o código e executa testes automatizados.
3. **Implantação** → Se os testes passarem, o **AWS CodeDeploy** implanta o código em servidores EC2, containers ECS ou funções Lambda.
4. **Monitoramento** → O **Amazon CloudWatch** coleta logs e métricas da aplicação para análise de desempenho.

DevOps na AWS

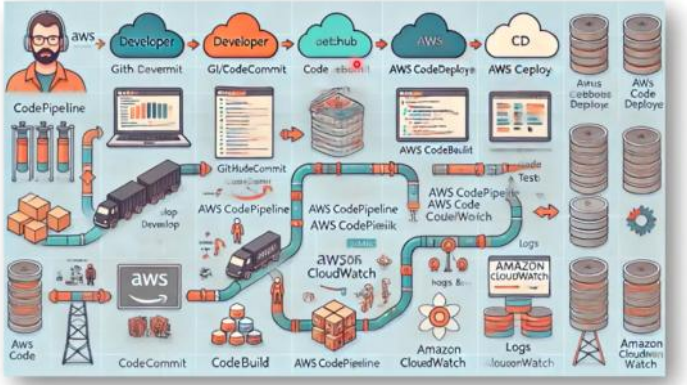
Principais Ferramentas DevOps da AWS

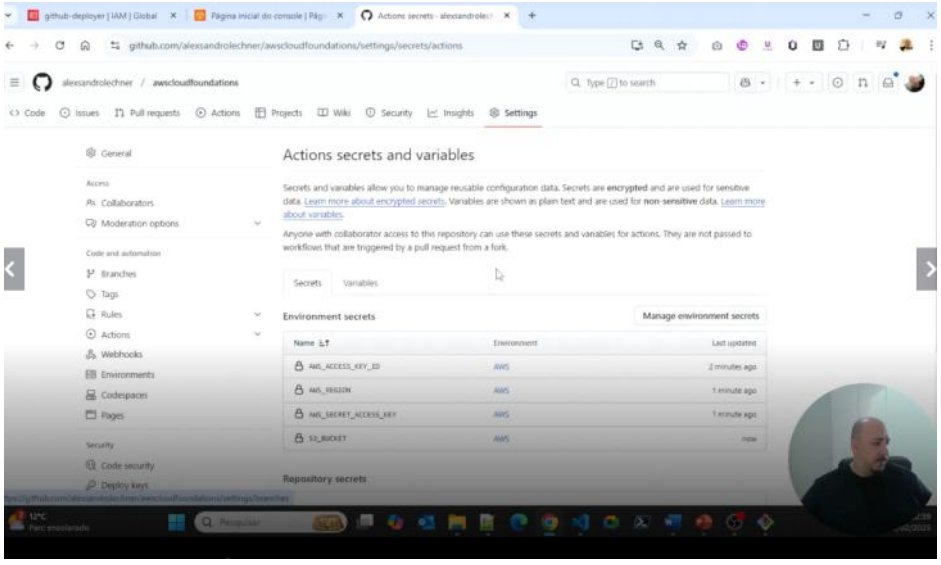


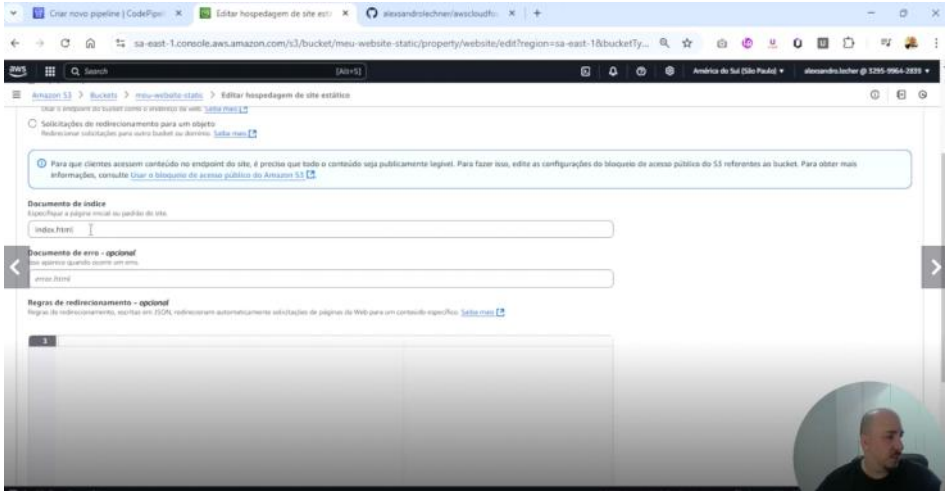
Benefícios:

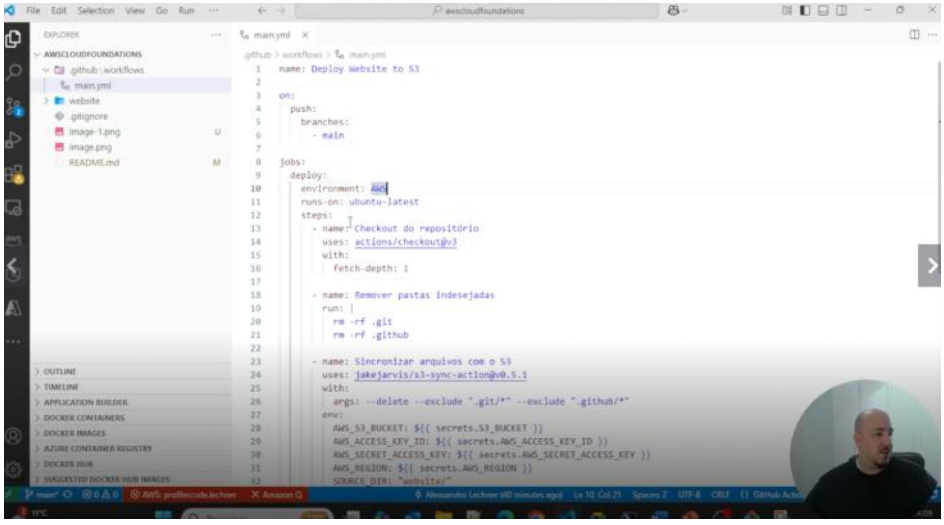
- Deploys automatizados e sem erros
- Testes contínuos garantem qualidade
- Redução do tempo entre desenvolvimento e produção

DevOps na AWS




github-deployer | IAM | Global
Página inicial do console | Pág
Actions secrets · alexandrolec
github.com/alexandrolechner/awscloudfoundations/settings/secrets/actions
alexandrolechner / awscloudfoundations
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Actions
Webhooks
Environments
Codespaces
Pages
Security
Code security
Deploy keys
Actions secrets and variables
Secrets and variables allow you to manage reusable configuration data. Secrets are encrypted and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for non-sensitive data. Learn more about variables.
Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.
Secrets
Variables
Environment secrets
Manage environment secrets
Name
Environment
Last updated
AWS_ACCESS_KEY_ID
AWS
2 minutes ago
AWS_REGION
AWS
1 minute ago
AWS_SECRET_ACCESS_KEY
AWS
1 minute ago
S3_BUCKET
AWS
now
Repository secrets


Criar novo pipeline | CodePipe
Editar hospedagem de site est
alexandrolechner/awscloudfo
sa-east-1.console.aws.amazon.com/s3/bucket/meu-website-static/property/website/edit?region=sa-east-1&bucketTy...
Search
[Alt+S]
América do Sul (São Paulo)
Amazon S3 > Buckets > meu-website-static > Editar hospedagem de site estático
Solicitações de redirecionamento para um objeto
Para que clientes acessem conteúdo no endpoint do site, é preciso que todo o conteúdo seja publicamente legível. Para fazer isso, edite as configurações do bloqueio de acesso público do S3 referentes ao bucket. Para obter mais informações, consulte Usar o bloqueio de acesso público do Amazon S3
Documento de índice
Especifique a página inicial ou padrão do site.
index.html
Documento de erro - opcional
Isso aparece quando ocorre um erro.
error.html
Regras de redirecionamento - opcional
Regras de redirecionamento, escritas em JSON, redirecionam automaticamente solicitações de páginas da Web para um conteúdo específico. Saiba mais


File Edit Selection View Go Run
awscloudfoundations
EXPLORER
AWSCLOUDFOUNDATIONS
.github > workflows
main.yml
website
.gitignore
image-1.png
image.png
README.md
OUTLINE
TIMELINE
APPLICATION BUILDER
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
main.yml
.github > workflows > main.yml
name: Deploy Website to S3
on:
push:
branches:
- main
jobs:
deploy:
environment: AWS
runs-on: ubuntu-latest
steps:
- name: Checkout do repositório
uses: actions/checkout@v3
with:
fetch-depth: 1
- name: Remover pastas indesejadas
run: |
rm -rf .git
rm -rf .github
- name: Sincronizar arquivos com o S3
uses: jakejarvis/s3-sync-action@v0.5.1
with:
args: --delete --exclude ".git/*" --exclude ".github/*"
env:
AWS_S3_BUCKET: ${{ secrets.S3_BUCKET }}
AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
AWS_REGION: ${{ secrets.AWS_REGION }}
SOURCE_DIR: "website/"

Ferramentas - DevOps na AWS

Dores e motivações para empresas migrarem para Cloud

As empresas que gerenciam seus sistemas de TI internamente lidam com uma variedade de desafios, incluindo custos excessivos, falta de escalabilidade e falta de flexibilidade.

Mudar para a nuvem pode ajudar a aliviar essas dores e fornecer às empresas uma variedade de incentivos.

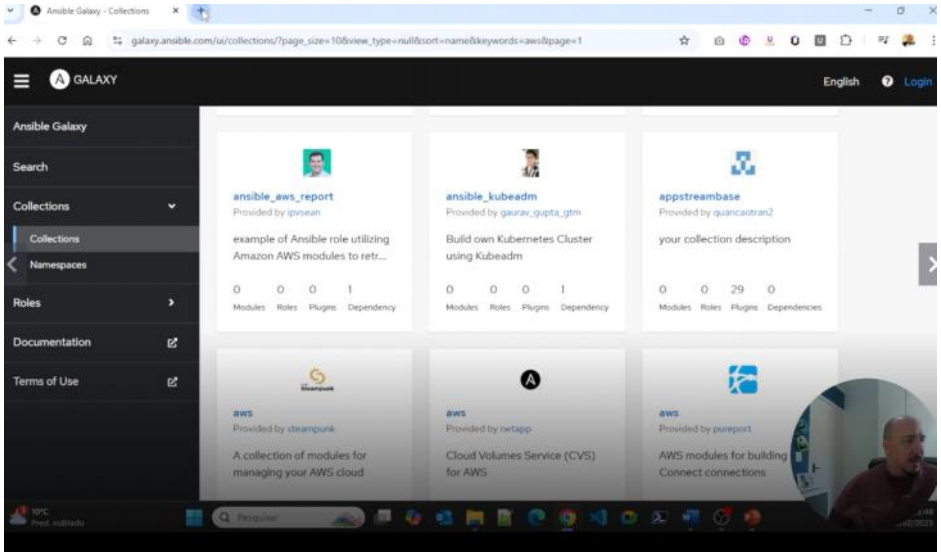
Ferramentas - DevOps na AWS

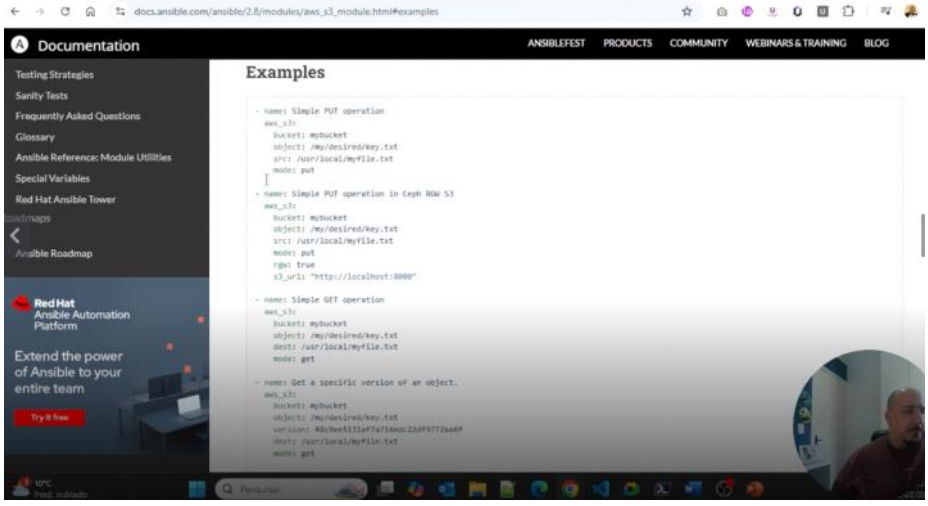
Modelos de Computação na Nuvem

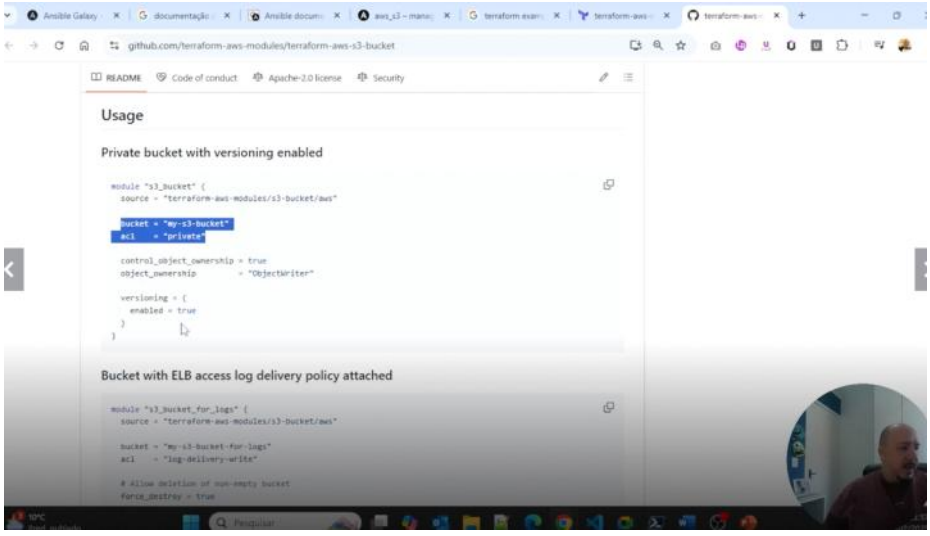
Ferramentas - DevOps na AWS

Infraestrutura como Código (IAC) é um processo de gerenciamento de infraestrutura de TI que aplica as melhores práticas do desenvolvimento de software Devops para o gerenciamento de recursos de infraestrutura em nuvem.

A ideia de Infraestrutura como Código (IAC) foi estimulada pelo sucesso do CI/CD.


Ansible Galaxy - Collections
galaxy.ansible.com/ui/collections/?page_size=10&view_type=null&sort=name&keywords=aws&page=1
GALAXY
English
Login
Ansible Galaxy
Search
Collections
Collections
Namespaces
Roles
Documentation
Terms of Use
ansible_aws_report
Provided by ipvsean
example of Ansible role utilizing Amazon AWS modules to retr...
0 Modules 0 Roles 0 Plugins 1 Dependency
ansible_kubeadm
Provided by gaurav_gupta_gtm
Build own Kubernetes Cluster using Kubeadm
0 Modules 0 Roles 0 Plugins 1 Dependency
appstreambase
Provided by quancaotran2
your collection description
0 Modules 0 Roles 29 Plugins 0 Dependencies
aws
Provided by steampunk
A collection of modules for managing your AWS cloud
aws
Provided by netapp
Cloud Volumes Service (CVS) for AWS
aws
Provided by pureport
AWS modules for building Connect connections

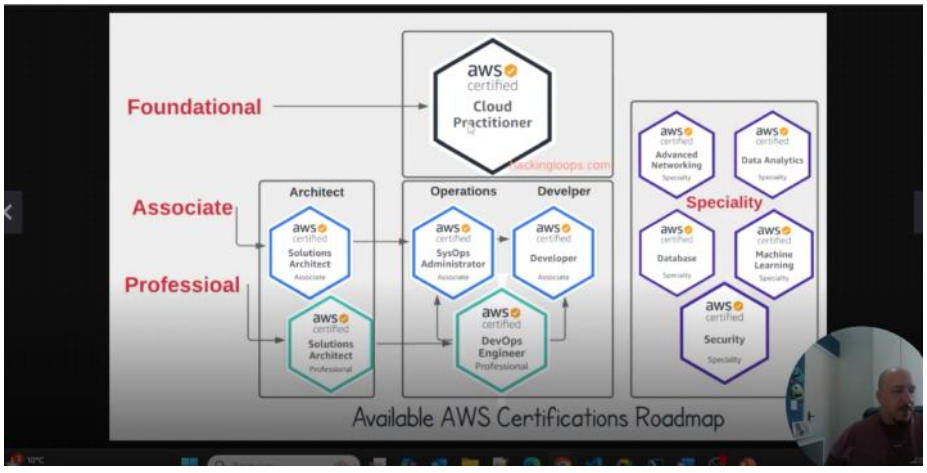

docs.ansible.com/ansible/2.8/modules/aws_s3_module.html#examples
Documentation
ANSIBLEFEST
PRODUCTS
COMMUNITY
WEBINARS & TRAINING
BLOG
Testing Strategies
Sanity Tests
Frequently Asked Questions
Glossary
Ansible Reference: Module Utilities
Special Variables
Red Hat Ansible Tower
Ansible Roadmap
Red Hat Ansible Automation Platform
Extend the power of Ansible to your entire team
Try it free
Examples


github.com/terraform-aws-modules/terraform-aws-s3-bucket
README
Code of conduct
Apache-2.0 license
Security
Usage
Private bucket with versioning enabled
module "s3_bucket" {
source = "terraform-aws-modules/s3-bucket/aws"
bucket = "my-s3-bucket"
acl = "private"
control_object_ownership = true
object_ownership = "ObjectWriter"
versioning = {
enabled = true
}
}
Bucket with ELB access log delivery policy attached

Ferramentas - DevOps na AWS

AWS CloudFormation, é um serviço fornecido pela Amazon Web Services que permite aos usuários modelar e gerenciar recursos de infraestrutura de maneira automatizada e segura.

- Evite erros humanos e automatize o provisionamento.
- Mantenha sua arquitetura armazenada em repositórios usando YAML ou JSON.
- Reutilize modelos para replicar ambientes.
- Total integração com quase todos os serviços da AWS.
- Provisione ambientes de forma mais rápida e simples comparado ao método manual.

Ferramentas - DevOps na AWS

AWS CloudFormation é essencial para atingir a excelência operacional, proporcionando ganhos imediatos e garantindo qualidade e reutilização no seu processo de provisionamento.