



# GE TAVARES



- Estudos para tirar **certificação AWS Certified Cloud Practitioner.CLF-C01.2**

O Certified Cloud Practitioner é a **certificação de nível básico da AWS** temos :

- Os fundamentos da nuvem, por exemplo, Conceitos de Nuvem, Arquitetura de Nuvem e Modelos de Implantação de Nuvem
- Uma análise mais detalhada dos principais serviços da AWS
- Uma rápida olhada na grande quantidade de serviços da AWS
- Identidade, segurança e governança da nuvem
- Faturamento, definição de preço e suporte dos serviços da AWS

# 1. Conceitos de Nuvem



## 1. Definição da Nuvem AWS

Computação em nuvem é o fornecimento de recursos de TI sob demanda, através da internet. Ou seja, em vez de despende \$ com data centers, infraestrutura física e manutenções, você só paga por aquilo que vai utiliza

### Tipos de Computação em Nuvem

<u>IaaS</u> <u>Infraestrutura c. Serviço</u>	<u>PaaS</u> <u>Plataforma c. Serviço</u>	<u>SaaS</u> <u>Software como Serviço</u>
Componentes básicos de TI na nuvem.  Exemplo: AWS (infraestrutura - AWS. Responsabilidade compartilhada)	Não gerencia a infraestrutura e foca na implantação e gerenciamento de apps.	Produto completo. Não se preocupa com a infra ou gerenciamento, apenas em como o software será utilizado.  Exemplo: Serviço Aurora da AWS

### Modelos de Implantação

<u>Pública</u>	<u>Privada</u>	<u>Híbrida</u>
Infraestrutura é compartilhada igualmente entre todos os usuários.  + Usada por pessoas físicas. \$ custo baixo. Segurança. - Flexibilidade - Flexibilidade.	Recursos reservados a um único usuário, ou seja, toda a infra de servidores, por exemplo, é para um determinado cliente.+ Segurança.  + Flexibilidade. \$ custo mais alto.	Junção dos dois modelos anteriores.  Permite que informações, plataformas e aplicativos sejam compartilhados tanto em um ambiente físico como à distância.  + muito utilizados por Big Corps.

## 2. Conceitos de Nuvem



### 1. Definição da Nuvem AWS

Computação em nuvem é o fornecimento de recursos de TI sob demanda, através da internet. Ou seja, em vez de despendar \$ com data centers, infraestrutura física e manutenções, você só paga por aquilo que vai utiliza

#### Tipos de Computação em Nuvem

#### Benefícios da Nuvem AWS

##### **ECONOMIA EM ESCALA**

###### **DESPESAS VARIÁVEIS - \$ CONFORME O USO**

Troque despesas iniciais (data centers, servidores físicos e outros recursos que você \$ antes de usar) por despesas variáveis (paga somente pelo que usa)

##### **AGILIDADE**

###### **IMPLANTANÇÃO EM - TEMPO**

A flexibilidade da nuvem facilita o desenvolvimento e a implantação de aplicativos.

##### **ALCANCE GLOBAL**

###### **BAIXA LATÊNCIA**

Implantação de aplicativos para clientes ao redor do mundo em poucos minutos, com BAIXA LATÊNCIA. Isso significa que mesmo estando distante fisicamente do cliente, ele poderá acessar seus apps com atrasos mínimos.

##### **CONFIABILIDADE**

###### **À PROVA DE FALHAS**

A AWS possui um alcance enorme e uma equipe de especialistas que permitiu a criação de uma rede sólida de servidores confiáveis e consistentes.

## Benefícios da Nuvem AWS

### **ALTA DISPONIBILIDADE**

#### **MULTI AZ / MULTI REGIÕES**

+ 99 zonas de disponibilidade em 31 regiões espalhadas pelo globo. Dentro de cada AZ podem existir milhares de datacenters.

#### **BAIXA LATÊNCIA / CACHE DE DADOS**

Além das AZ, existem as EDGE LOCATIONS, que são pequenos datacenters que ficam em locais estratégicos, usado para cache de dados.

### **ELASTICIDADE**

#### **REAJUSTE / RESILIÊNCIA / ASG**

Capacidade de reajuste automático (aumentar e diminuir) a infraestrutura sob demanda, na quantidade exata que precisar.

### **ESCALABILIDADE**

#### **FLEXIBILIDADE**

A quantidade de serviços contratados pode ser modificada de acordo com a necessidade.

### **SEGURANÇA**

#### **ESCALÁVEL / CONFIÁVEL / ABORDAGEM PONTA A PONTA**

O cliente se beneficiará de um datacenter e de uma arquitetura de rede com maior exigência de segurança e terá tudo o que precisa, de acordo com o uso.

## **Como a nuvem AWS permite que os usuários se concentrem no valor empresarial**

O valor da nuvem vai além da redução de custos total de propriedade (TCO). Os clientes também veem melhorias em outras áreas, como:

- ◆ Produtividade da equipe Resiliência
- ◆ operacional
- ◆ Agilidade nos negócios

→ **CLOUD VALUE FRAMEWORK** - usado para demonstrar o valor empresarial usando os 4 pilares. Ele ajuda os clientes a entender o valor comercial da mudança para a AWS e da criação nela.

- ♦ **Redução de custo (TCO)** - redução/prevenção de custos de infra.
  - ◻ A AWS ajuda seus clientes a reduzir os custos por meio do uso de:
    - modelo baseado em consumo: \$ pelo uso
    - modelo de definição de preço da AWS: modelo de uso + requisição de cargas de trabalho
    - frequentes reduções de preço. **Exceção** - instâncias reservadas \*\*
- ♦ **Produtividade em equipe** - melhoria na **eficiência por função**.
- ♦ **Resiliência operacional** - **maior disponibilidade**, segurança e conformidade.
- ♦ **Agilidade empresarial** - implantação de **novos apps + rápido**. Capacidade de experimentar mais e responder mais rápido.

## 1.2 Identificação de Aspectos da Economia da Nuvem AWS

### CAPEX - DESPESAS DE CAPITAL

→ Investimentos para obtenção de benefícios a longo prazo.

→ Custos e manutenção.

**Exemplo:** computadores e servidores.

### OPEX - DESPESAS OPERACIONAIS

→ Gasto contínuo que proporciona redução de custos e maximização de produtividade.

→ Gastos diários/mensais.

**Exemplo:** assinatura de serviços cloud.

## Custos Reduzidos com a Migração

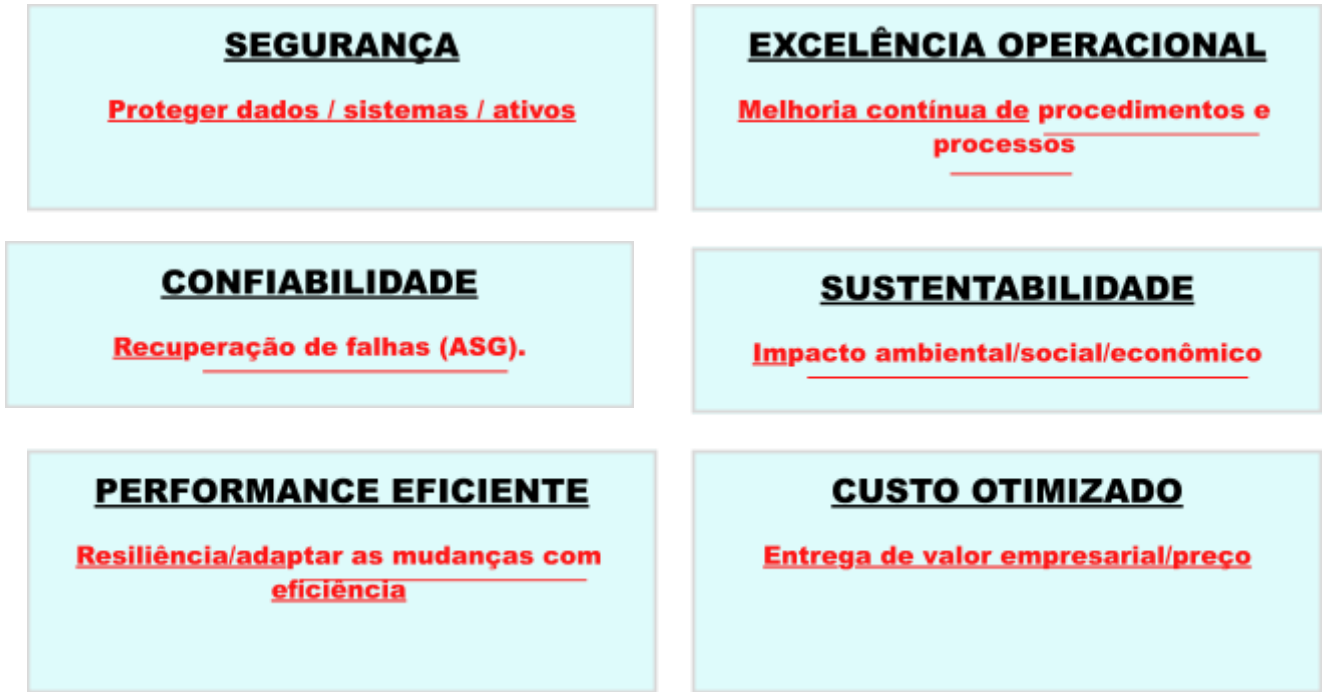
- ♦
- ♦ Redução de custo com a infraestrutura - right sizing

### ROI - Retorno sobre o investimento

CAPEX → é obtido muito tempo após a compra.

OPEX → obtido imediatamente, já que a infra é gerenciada pelo provedor.

# OS 6 PILARES DO WELL-ARCHITECTED FRAMEWORK



## DESIGN À PROVA DE FALHAS

Redundância, infraestrutura em diferentes Zonas de Disponibilidade.  
Serviços gerenciados oferecem soluções automáticas de recuperação de falhas, pois os serviços já possuem

# 2.Segurança e Conformidade

## 2.1. Responsabilidade Compartilhada da AWS

A **AWS criptografa** automaticamente os dados na **CAMADA FÍSICA** antes de sair de suas instalações protegidas.



fonte: <https://blog.estabil.is/aws-na-pratica-modelo-de-responsabilidade-compartilhada/>

## 2.2. Segurança e Conformidade da Nuvem AWS

**CONFORMIDADE = COMPLIANCE** → estar de acordo com os padrões de segurança do mercado.

Padrões reconhecidos:

♦

### Como atingir a conformidade - cliente?

→ Criptografia - variam de acordo com cada serviço.

**AT Rest (em repouso)** - no momento que está no servidor, está criptografado.

**In Transit (em trânsito)** - tráfego de dados, então você consegue criptografar os dados enquanto estão trafegando.

## **CLOUD TRAIL**

### **RASTREABILIDADE / AUDITORIA / LOCALIZAÇÃO NA CONTA**

Serviço de **logs relacionado a conta**, ou seja, consegue verificar quem alterou alguma permissão, ou excluiu uma EC2, por exemplo.

Ele habilita esse logs, acompanho-os e rastreando o que o cliente está fazendo na conta.

**DATA EVENT X MANAGEMENT EVENTS** - O primeiro vai + afundo em eventos de dados mais específico (Ex.: put de um objeto no S3). O segundo, foca mais na questão da gestão (Ex.: quem criou um banco de dados ou uma EC2).

## **CLOUD WATCH**

### **MONITORAMENTO/MÉTRICAS/ LOGS DA APP - CLOUD WATCH LOGS**

Realização de coleta de logs e métricas.

- Criação de dashboard.
- Métricas incorporadas e personalizadas.
- Criação de alarmes.
- Log da aplicação - centraliza os logs de todos os sistemas e apps.

**Amazon KMS** - serviço que gerencia chaves criptografadas - cliente pode gerenciar as próprias chaves (trazendo-as) ou a AWS se responsabiliza por elas



## **CONDIÇÕES ESPECÍFICAS/ PERMISSÃO/ PRIVILÉGIO MÍNIMO**

### **AWS CONFIG**

#### **CONFIGURAÇÕES/CONTA/PADRÃO CONFIG/HISTÓRICO**

→ Ele mostra o histórico de configurações realizadas na conta.

→ Detecta a mudança das modificações fora de compliance.

Ao definir as permissões do IAM, a ideia é conceder as permissões de privilégio mínimo para executar a tarefa, ou seja, você define as ações que podem ser executadas em recursos específicos com condições específicas.

## **2.3. Recursos de Gerenciamento de Acesso**

### **FINALIDADE DO GERENCIAMENTO DE USUÁRIOS E IDENTIDADES**

#### **→ Funcionamento do IAM:**

- ◆ Credenciais com senha para fazer login no console.
- ◆ Chaves de gerenciamento para fazer chamadas por meio da API AWS (AWS CLI, AWS SDK).
- ◆ Gerenciamento da complexidade/alternância de senha.
- ◆ Controle de autenticação MFA (multifactor authentication).

#### **→ IAM - Identify and Access Management**

### **USERS**

Usuário da conta (com login e senha), com suas permissões específicas

### **GROUPS**

Grupo de usuários para dar permissão x para todos os usuários do grupo

### **PERMISSIONS**

Regra de acesso para cada usuário, ou seja, lista de permissões para executar tarefas

### **ROLE**

Permissão dada a outro recurso da AWS ou a terceiros que não são usuários

**Ex.:** EC2 que quer acessar o S3

### **POLICY**

Documento com o conjunto de uma ou mais permissões

#### → **Conta raiz da AWS:**

- ◆ É a conta privilegiada, com mais permissão.
- ◆ **NÃO** deve ser utilizada nas tarefas diárias ou administrativa.

⚠ Colocar MFA para acessar conta raiz.

⚠ Criar conta/usuário IAM como adm. e fazer as tarefas a partir dele.

#### **Quando usar a conta raiz?**

- ◆ Alterar configurações da conta.
  - informação contato, preferência de moeda e regiões **NÃO** precisam da conta raiz/root.
- ◆ Restaurar as permissões de usuários do IAM.
- ◆ Ativar o acesso do IAM ao console do billing and cost management.
- ◆ Exibir determinadas faturas de imposto, como o da AWS Inc. ou Amazon Internet Services Private Limited - AISPL.
  - para o View Billing pode baixar e visualizar as faturas de IVA da AWS Europa.
- ◆ Feche sua conta na AWS.
- ◆ Registrado como vendedor no marketplace da instância reservada. Configure um bucket
- ◆ S3 para configurar o MFA.

## 2.4. Recursos para Suporte de Segurança SERVIÇOS NATIVO AWS

### AMAZON WAF

Serviço de firewall das aplicações web.

### NACL - NETWORK ACESS LIST

Firewall de grupo de recursos dentro da AWS.  
Nível de subnet.

### OTIMIZAÇÃO DE CUSTO

Veja como economizar ao eliminar recursos ociosos ou não usados, ou fazendo compromisso de capacidade reservada.

### AWS MARKETPLACE

Aprimore a performance do seu service limit  
monitoramento de instância com uso excessivo.

Para soluções de terceiros/parceiros.

\*\* Migração com licença específica

### TOLERÂNCIA A FALHAS

Aumenta a disponibilidade e a redundância, usando ASG, verificações de integridade, multi AZ e backup.

### LIMITES DE SERVIÇO

Procura por utilização de serviço acima de 80% do service limit.

### SEGURANÇA

Verificações de segurança fazem parte.

Elimine lacunas, habilite recursos e analise permissões.

→ Permissões S3 bucket - essa verificação examina permissões associadas que podem substituir as permissões do bucket.

→ Uso do MFA na conta raiz.

### TRUSTED ADVISOR

Serviço que faz recomendações para seguir as melhores práticas.



# 3.1. Infraestrutura Global da AWS

<u>REGION</u>	<u>AVAILABILITY ZONE</u>	<u>EDGE LOCATIONS</u>
<p><u>ESPAÇO GEOGRÁFICO</u></p> <p>Espaço geográfico macro em que se agrupam zonas de disponibilidade. Os serviços da AWS são separados por região.</p>	<p><u>CONJUNTO DE DATA CENTERS</u></p> <p>Conjunto de data centers que se conectam em uma região. Uma mesma região possui mais de uma AZ.</p>	<p><u>CACHE DE DADOS/BAIXA LATÊNCIA/LOCAIS DE BORDA</u></p> <p>Locais com um pequeno setup para transmitir e cachear dados. Serve para aplicações em cache, com entrega de baixa latência.</p>

## BENEFÍCIOS DOS LOCAIS DE BORDA - EDGE LOCATIONS:

<u>AMAZON CLOUD FRONT</u>	<u>AWS GLOBAL ACCELERATOR</u>
<p><u>CACHE DE DADOS/ENTREGA DE DADOS + RÁPIDA/BAIXA LATÊNCIA/WEBSITE + RÁPIDO</u></p> <p>Serviço de CDN (Cloud Delivery Network) - entrega de dados em cache e mais rápido</p>	<p><u>BAIXA LATÊNCIA/APLICAÇÃO LOCAL/MULTI REGIÕES/ALTA DISPONIBILIDADE</u></p> <p>IP's estáticos como uma única porta de entrada para aplicações locais</p>

□	<p><u>ALTA DISPONIBILIDADE</u></p> <p><u>VÁRIAS ZONAS DE DISPONIBILIDADE</u></p> <p>Alta disponibilidade → serviços estarem presentes em mais de uma zona. AZ <b>NÃO</b> compartilham um único ponto de falha.</p>
---	--

SDK - permite que acesse e gerencie serviços da AWS com sua linguagem ou plataforma de desenvolvimento preferencial.

- ◆ Console gerenciamento AWS - via console.
- ◆ CLI - via terminal.
- ◆ Infra as code - AWS CloudFormation - template (semelhante ao terraform). AWS
- ◆ CDK - integração dos recursos via código.

### OPÇÕES DE CONECTIVIDADE

- ◆ VPN
- ◆ AWS Direct Connect
- ◆ Internet Pública

## 3.2. Infraestrutura Global da AWS

### REGION

#### ESPAÇO GEOGRÁFICO

Espaço geográfico macro em que se agrupam zonas de disponibilidade. Os serviços da AWS são separados por região.

### AVAILABILITY ZONE

#### CONJUNTO DE DATA CENTERS

Conjunto de data centers que se conectam em uma região. Uma mesma região possui mais de uma AZ.

### EDGE LOCATIONS

#### CACHE DE DADOS/BAIXA LATÊNCIA/LOCAIS DE BORDA

Locais com um pequeno setup para transmitir e cachear dados. Serve para aplicações em cache, com entrega de baixa latência.

### **BENEFÍCIOS DOS LOCAIS DE BORDA - EDGE LOCATIONS:**

#### AMAZON CLOUD FRONT

#### CACHE DE DADOS/ENTREGA DE DADOS + RÁPIDA/BAIXA LATÊNCIA/WEBSITE + RÁPIDO

Serviço de CDN (Cloud Delivery Network) - entrega de dados em cache e mais rápido

#### AWS GLOBAL ACCELERATOR

#### BAIXA LATÊNCIA/APLICAÇÃO LOCAL/MULTI REGIÕES/ALTA DISPONIBILIDADE

IP's estáticos como uma única porta de entrada para aplicações locais

### ALTA DISPONIBILIDADE

#### VÁRIAS ZONAS DE DISPONIBILIDADE

Alta disponibilidade → serviços estarem presentes em mais de uma zona. AZ **NÃO**

# QUANDO USAR MÚLTIPLAS REGIÕES?

## DESASTER RECOVERY

- ◆ Recuperação de desastres/ continuidade do negócio.
- ◆ **Baixa latência** - cenário global de acesso.
- ◆ **Soberania de dados** - dados do cliente residirão em uma determinada região pré-selecionada mesmo que a aplicação esteja em outra região.
- ◆ **RPO** → quantidade de **informação** tolerável a perder quando estou me recuperando de uma falha.
- ◆ **RTO** → quantidade de **tempo** tolerável a perder.

## 3.3. Identificação dos Serviços AWS

### CATEGORIAS DE SERVIÇOS AWS:

<u>COMPUTAÇÃO</u>	<u>ARMAZENAMENT</u> <u>O</u>	<u>REDE</u>	<u>BANCO DE DADOS</u>
EC2	S3	VPC	EC2
LIGHTSAIL	AWS SNOWBALL	SECURITY GROUPS	RDS
ECS/EKS	EBS	ROUTE 53	DYNAMO DB
BEANSTALK	EFS	CLOUDFRONT	REDSHIFT
LAMBDA	STORAGE GATEWAY	VPN	AURORA
		DIRECT CONNECT	ELASTIC CACHE
		IP ELÁSTICO	NEPTUNE

## EC2

Máquina virtual.

- ◆ **Placement Groups - como organizar as instâncias.**
  - **Cluster** → instâncias mais próximas uma das outras.
    - **BAIXA LATÊNCIA/HIGH WORKLOAD/ HPC.**
  - **Partition** → divide instâncias em partições de tal forma que cada partição NÃO divide o mesmo hardware. Usado em BIG DATA.
    - **APLICAÇÃO DE WORKLOADS/ WORKLOADS DISTRIBUÍDOS/ HADOOP/ KASSANDRA/ KAFKA.**
  - **Spread** → divide instâncias de tal forma que cada instância não divide o mesmo hardware.
    - **WORKLOAD CRÍTICO/ REDUÇÃO DE FALHAS/ MIX DE INSTANCE TYPES**
- ◆ **Aumento de Limite EC2** → via Support Center.
- ◆ **Diferença** → modos de pricing
  - **On-Demand** - paga conforme o uso - não tem muita ideia do tempo que irá usar.
  - **Reservada** - Paga adiantado ou não - por 1 ou 3 anos - pode ser conversível ou não. **Spot** - até 90% de desconto - instáveis/podem cair - com base em oferta e demanda da AWS.
  - **Host Dedicado** - não compartilha fisicamente a VM com outras empresas.

<u>DEDICATED HOST</u>	<u>DEDICATED INSTANCE</u>	<u>SHARED TENANCY</u>
Host dedicado. Instância dedicada.	Host compartilhado. Instância dedicada.	Localização compartilhada. Instância compartilhada.
\$ hora fixa - paga por hora e host.	\$ paga por instância + região.	\$ máquina rodando - paga por instância.

### EC2 - FAMILY INSTANCE

- ◆ **Uso Geral** → equilibram recursos - computação, memória e rede.
  - **Casos de uso:** jogos, banco de dados pequeno e médio e back end.
- ◆ **Computação otimizada** → computação com alto desempenho.
  - **Casos de uso:** para aplicações vinculadas a computação que se beneficiam de processamento com alto desempenho.
- ◆ **Memória otimizada** → grandes conjuntos de dados com desempenho rápido.
  - **Casos de uso:** grandes memórias.
- ◆ **Computação acelerada** → acelerador de hardware para agilizar o processamento de dados.
  - **Casos de uso:** uso matemático, machine learning e netflix\*\*.
- ◆ **Armazenamento otimizado** → alto acesso sequencial de leitura e gravação a grande conjunto de dados no armazenamento local. Muita escrita e leitura.

## EC2 - STORAGE

- ◆ **Instance Store** → armazenamento temporário
  - Se terminar a instância, os dados se perdem.
- ◆ **EBS - Elastic Block Store** → dados persistentes.
  - Se encerrar a instância, os dados permanecem.

## LIGHTSAIL

- ◆ Virtual Private Service.
- ◆ **Mais simples.**
- ◆ Para aplicações simples → como websites.
- ◆ **Casos de uso:** websites, business softwares, ambientes de teste/dev.

## ECS

- ◆ Ferramenta para orquestração de contêineres.
- ◆ Orquestração gerenciada pela AWS.
- ◆ Casos de uso: aplicação de microsserviços, batch workloads, docker contêineres. O
- ◆ EKS pode ser utilizado com o ECS para orquestrar os contêineres.
- ◆ **ECR** → registro de contêiner docker.

## FEATURES ECS

- ◆ **AWS COPILOT** → semelhante ao CLI, mas aqui é para ajudar com comandos na orquestração.
- ◆ **AWS OUTPOSTS** → on-premises - hack da AWS que ajuda no deploy. Estende a infraestrutura e os serviços AWS para o datacenter local.
- ◆ **FARGATE** → Funciona com ECS e EKS - gerenciador de contêiner sem servidor.

## BEANSTALK

- ◆ Orquestrador totalmente gerenciado.
- ◆ Fácil deploy de aplicações, trazendo junto o load balancer e o auto scaling group.
- ◆ **Compatível:** Java, .Net, PHP, Node, Python, Ruby, Gp, Docker.

## LAMBDA

- ◆ Function as a Service.
- ◆ Infraestrutura totalmente gerenciada pela AWS. Já inclui o LB e ASG.
- ◆ **Sem servidor (serverless).**
  - Mesmo sendo sem servidor, é possível trazer toda configuração de rede para as Lambdas.
- ◆ Você fica focado totalmente no código da aplicação.
- ◆ **Contras:**
  - Cold starts → como ela gerencia tudo, as primeiras vezes podem ser um pouco mais lentas.
  - Timeout → máximo de 15 minutos. Se sua aplicação demora muito, não compensa.
- ◆ **Pode ser acionado diretamente pelo SNS e S3.**
- ◆ Orientado por eventos.



- ◆ Faz provisionamento automático para executar as funções.

## AUTO SCALING

### ALTA DISPONIBILIDADE / MULTI AZ

#### Launch template + scaling group

- ◆ Possibilidade de aumentar e reduzir suas instâncias conforme necessário.
- ◆ **Scale Out / Scale In** → criar novas instâncias umas espelhos das outras para atender aumento da demanda e reduzir, quando não for mais necessário.
- ◆ Configuração: através do launch template, similar ao launch configuration, porém permite versionamento.
  - Launch config → não atualiza mais / legado. Template com especificações da instância a ser criada dentro de um ASG (AMI, key pair, EBS, etc.).
- ◆ **Auto scaling garante alta disponibilidade quando cria instâncias em mais de uma AZ.**
- ◆ Tipos de escalonamento:
  - **Manual**
  - **Dinâmico** → baseado em alarmes (cloud watch)
    - **Target tracking** → manter média estipulada. Ex.: quero que a VM fique com 40% CPU.
    - **Simple scaling** → 2 passos, mais simples que o step.
      - ◆ **Cooldown** → verifica a necessidade → ele cria a nova máquina, ai você estipula um tempo (5 min), ele vai e verifica a regra novamente e confirma se a máquina precisa ser mantida ou não, ou se precisa aumentar (é um ciclo).
    - **Step scaling** → muito mais configurável.
    - **Predictive scaling** → funciona com machine learning (mais complexo) para prever o ciclo do tráfego.
    - **Scheduled scaling** → define data/hora para subir ou reduzir as máquinas. Ex.: na data da black friday já iniciar com 6 VM's.

## LOAD BALANCER

- ◆ Distribuir tráfego na rede conforme necessário. Tipos
- ◆ de load balancer (divisão por protocolo):
  - **Application LB** → HTTP/HTTPS - tráfego vindo da internet, web app. - camada 7.
  - **Network LB** → TCP/TLS/UDP - tráfego interno: DB - camada 4.
  - **Gateway LB (novo)** → IP - conexão direta com os softwares de terceiros - protocolo GENEVA.
- ◆ **OBS.:** Classic LB - **NÃO EXISTE MAIS**

ESCALONAMENTO **VERTICAL** AUMENTA INSTÂNCIA E

ESCALONAMENTO **HORIZONTAL** ADICIONA INSTÂNCIA.

# FAMÍLIA ARMAZENAMENTO S3

## ESTÁTICO/DURÁVEL/USA POUCA CPU

- ◆ Armazenamento do tipo **object storage**.
- ◆ O arquivo sempre é substituído como um todo → **não** permite editar.
- ◆ Ideal para criação de websites estáticos.
- ◆ Características:
  - Batch operations → envio de grandes cargas de trabalho por lotes.
  - Multipart upload → recomendado para arquivos maiores que 100MB para agilizar.
  - Versionamento e MFA on **delete**.
  - Replication Cross Region → replicar objetos em outra região.
  - Gerenciamento de acesso → ACLs, access points. Políticas específicas de acesso.
    - S3 bucket policy → informação confidencial no S3 - acesso restrito.
- ◆ Limites de tamanho:
  - 5GB → única operação PUT usando SDK, API Rest e CLI.
  - 160GB → usando S3.
  - 5TB → upload fracionado - API, usando SDK, API Rest e CLI.
- ◆ Classes:
  - S3 standard
  - S3 intelligent-tiering
  - S3 standard-IA
  - S3 onde zone-IA
  - S3 glacier
  - S3 glacier deep archive



- ◆ **S3 Transfer Acceleration** → transferência de arquivos rápidas, fáceis e rápidas em longas distâncias entre cliente e bucket S3. Usa os pontos de presença (edge locations) distribuídos no cloud front (**DICA**: cloud front é rede).
- ◆ URL pré-assinada → disponibiliza acesso específico por um tempo.

## AWS SNOWBALL

### TRANSFERÊNCIA DE DADOS OFF - FÍSICO/ PETA SIZE

- ◆ Migração de dados em peta size. Transferir
- ◆ dados off.
- ◆ Como funciona: a AWS te dá uma caixona → você recebe e coloca todos os arquivos nela → devolve para a AWS → transferência para o S3.
  - **Snowball edge** - vem com algumas configurações a mais além de só transferir dados.
  - **Snowmobile - caminhão** → 100PB
  - **Snowcone** - 8TB

## EBS - ELASTIC BLOCK STORE

### PERSISTÊNCIA DE DADOS/ARQUIVOS EM BLOCO/TRABALHA JUNTO COM EC2

- ◆ Dados são persistentes
  - Ou seja, se a instância for excluída, os dados permanecem.
- ◆ Arquivos em bloco
- ◆ **EC2 + EBS** = garante elasticidade e economia da EC2.
- ◆ **OBS:** não é multi-AZ e apenas 1 instância.
- ◆ Snapshots do EBS → o primeiro, todos os arquivos são copiados, os seguintes, somente as alterações.

## EFS - ELASTIC FILE SYSTEM

- ◆ Armazenamento em forma de arquivos compartilhados entre serviços de computação. Arquivos
- ◆ compartilhados
- ◆ Multi-AZ
- ◆ Várias instâncias Ec2 simultaneamente
- ◆ Sistema de arquivos escalável

## STORAGE GATEWAY

- ◆ Solução para **cloud híbrida**
- ◆ **Conectar on-premises para acessar dados na nuvem**
- ◆ Tipos:
  - File gateway - acessar backups, media content, arquivos para times → protocolo SMB/NFS
    - Tipos de migração: S3 e FSX
  - Volume gateway - armazenamento híbrido, acesso com cache de dados, migração de dados, armazenamento em bloco → protocolo ISCSI
  - Tape gateway - criação de backups, arquivos raramente acessados, substituir fitas fís→ protocolo ISCSI.

# FAMÍLIA DA REDE

## VPC - VIRTUAL PRIVATE CLOUD

- ◆ Criar recursos em uma rede isolada. A princípio, essa rede não é acessível da internet. Subnets →
- ◆ sub-porções da rede virtual. Cada subnet se situa em uma AZ diferente.
  - **NACL - Network Access Control Lists** → lista que controla (allow ou deny) do acesso. São stateless (precisa especificar o IP). As regras são ordenadas por prioridade.
- ◆ Route tables → estão associadas a subnet e dizem para onde deve ser roteado o IP que chega a subnet.
- ◆ Internet gateway → é uma ferramenta que possibilita que a VPC se comunique com a internet. VPC
- ◆ peering → conexão entre VPCs

## SECURITY GROUPS

- ◆ Associados às instâncias
- ◆ Definem quais IPs são permitidos naquele grupo de segurança É
- ◆ possível especificar IPs ou outro grupo de segurança
- ◆ Só permite regras de allow (deny é padrão)
- ◆ Stateful → uma vez que defina que ele pode acessar a máquina, subentende que a máquina pode acessar o IP (diferente do stateless).

## ROUTE 53

- ◆ Serviço de DNS (Domain Name System)
- ◆ Mapeia IPs para domínios
- ◆ Routing Policies:
  - Simple routing → roteamento padrão sem especialidade
  - Failover routing → muda o roteamento conforme um IP para de responder (active/passive)
  - Geolocation routing → faz o roteamento de acordo com a localização geográfica dos usuários - baixa latência
    - Exemplo: brasileiros acessam o servidor de SP.
  - Geo proximity routing → faz o roteamento de acordo com a localização geográfica dos usuários e dos recursos. Possui regras mais complexas para definir roteamento Latency →
  - based routing - roteamento para a região de menor latência.
  - Multivalue answer routing → define um conjunto de IPs e o Route 53 escolhe um para rotear
  - Weighted routing → associa pesos a diferentes recursos para rotear.
    - Bom para criação de site novo, para realizar migração Exemplo:
    - 60% da rota vai para x e 40% vai para y
- ◆ Tipos de record:
  - Alias → aponta para o nome de um recurso AWS (específico do route 53) A
  - → um IP
  - CNAME → nome (www.example.com)
  - MX - nome do servidor de e-mail

# CLOUD FRONT

## BAIXA LATÊNCIA/EDGE LOCATION

- ◆ Serviço de CDN - Content Delivery Network
- ◆ Distribui conteúdos com baixa latência - conteúdo de cache (edge location)
- ◆ Distribui vídeo on demand (VOD) - live streaming
- ◆ Definiu uma origem e aponta para S3, EC2 ou outro web server
- ◆ Possui proteção contra DDoS (ataque de negação de serviço)
  - **Dica:** o recurso relacionado à segurança que possui esse nível de proteção, é o AWS shield.

# VPN - VIRTUAL PRIVATE NETWORK

- ◆ Gerenciado pela AWS
- ◆ Estabelece conexão segura entre redes locais
- ◆ Integração de rede
- ◆ Client VPN → totalmente gerenciado, TLS VPN, acesso aos recursos de qualquer lugar
- ◆ Site to site VPN → conecta toda a VPC com uma rede privada, IPsex VPN
  - Conexão através do route 53 + VPG - virtual private gateway VPN
- ◆ cloud hub → configura inúmeras VPNs site to site

# DIRECT CONNECT

## CONEXÃO DEDICADA/SEM INTERNET/BAIXA LATÊNCIA

- ◆ Conexão dedicada on-premises na nuvem
- ◆ Não passa pela internet
- ◆ Como funciona?
  - Pedido de conexão (precisa de um motivo) → AWS aprova o download da carta de autorização → vai ao parceiro da AWS que possui conexão dedicada → configura e usa

# IP ELÁSTICO

- ◆ Permite manter um IP ativo como porta de entrada e também que sejam feitas associações de IPs dinâmicos a cada provisionamento de uma nova máquina, uma vez que nesses cenários os IPV4 mudam
- ◆ **Ele é associado à sua conta e não a instância**

## ● FAMÍLIA DO BANCO DE DADOS **EC2**

- ◆ É possível criar uma EC2 e configurar para usá-la como banco de dados
- ◆ Totalmente gerenciado pelo cliente

## BANCO RELACIONAL/GERENCIADO/ESCALAVEL/MULTI-AZ/READ REPLICAS

- ◆ Banco relacional
- ◆ Gerenciado
- ◆ Multi-AZ
  - aumenta a disponibilidade e tolerância a falhas (MAIS CARO).
- ◆ Read replicas → direciona o somente leitura para as cópias - reduzindo a latência aumenta a
  - escalabilidade e melhora o tráfego dos dados, direcionando os processos somente leitura para as cópias das base de dados (MAIS BARATO).
- ◆ Pague somente pelo que usar
- ◆ Snapshot: máximo de 100 manuais por região.
- ◆ Em relação a EC2, é mais benéfico, pois mantém o sistema operacional e executa patches de software.
- ◆ Compatibilidade: postgre SQL, MySQL, MariaDB, Oracle e SQL Server

## DYNAMO DB

### **BANCO NÃO RELACIONAL/NoSQL/CHAVE VALOR/\$ POR KB/GERENCIADO/SERVERLESS/BAIXA LATÊNCIA**

- ◆ Banco não relacional
- ◆ Banco chave e valor
- ◆ Tabelas e índices
- ◆ Método de prancing → RCUs e WCUs - quantidade de escrita e leitura
- ◆ Pague "on-demand" ou "provisioned"
- ◆ Tipos de busca:
  - Scan → lê tudo
    - OBS.: se usar muito, o \$ aumenta, pois a cobrança é por leitura Query
  - → lê uma parte por chave/index
- ◆ **COBRANÇA FEITA POR KB**

## REDSHIFT

### **BANCO COLUNAR/DATA WAREHOUSE/PETA BYTES/ BIG DATA/BI**

- ◆ Banco colunar
- ◆ Gerenciado
- ◆ Ideal para data warehouse/data lake
- ◆ Escalável a peta bytes
- ◆ Integração com sistemas de BI.

## AURORA

- ◆ Banco relacional
- ◆ Gerenciado
- ◆ \$ caro
- ◆ x mais rápido
- ◆ Compatibilidade: Engine MySQL e PostGre SQL

- ◆ Aurora serverless → pausa o banco após 5 minutos de inatividade (could start)

## ELASTIC CACHE

### ARMAZENAMENTO DE INF POR SESSÃO/CACHE

- ◆ Armazenamento em memória para ser utilizado por aplicações escaláveis para guardar informações de sessão, consultas de bancos
- ◆ Casos de uso flexível e em tempo real
- ◆ Armazenamento em cache
- ◆ Quando usar: casos que não exigem durabilidade, como armazenamento em sessão, placares de jogos, streaming e análises
- ◆ **Compatibilidade:** Redis e Memcached
  - Redis → persiste dados, escalável → transações em tempo real, game leaderboard e session store.
  - Memcached → memória chave valor, cache de dados de banco, banco em stand-by e não persiste os dados.

## ELASTIC CACHE/OPEN SEARCH

- ◆ Alta capacidade de armazenamento de dados de logs de serviços par garantir a análise de problemas e falhas
  - Mecanismo de pesquisa
  - Análises de logs Pesquisa
  - de texto
- ◆ Open search → para escalabilidade de cluster

## DOCUMENT DB

- ◆ Armazenamento de dados baseado em JSON
- ◆ Gerenciado
- ◆ Opera workloads MongoDB de missão crítica Banco
- ◆ NoSQL

## 3.4. Recursos para Suporte Tecnológico

### DOCUMENTAÇÃO - CONHECIMENTO

- ◆ Documentação oficial
- ◆ Knowledge center
- ◆ Whitepapers
- ◆ AWS fórum
- ◆ AWS blogs

### SUPORTE - AWS Support

#### Alguns pontos relevantes:

### **BASIC**

7 verificações do consultor, dashboard com a saúde das máquinas e documentação / formulário de suporte

### **DEVELOPER**

Para quem está conhecendo ou testando Suporte

básico + e-mail

1 contato

Tempo de resposta:

24h geral

2h sistema prejudicado/falhas

### **BUSINESS**

Desenvolvimento + verificação completa de consultor (trusted advisor)

Telefone de suporte

Contatos ilimitados

Tempo de resposta:

1h para queda de produção

API do AWS Support

### **ENTERPRISE**

Business + Engenheiro de suporte

Tempo de resposta:

15min para negócios críticos de sistema

Inclui revisão Well-Architected pelo Arquiteto de soluções da AWS

Laboratórios

Concierge do time de suporte - ajuda com a parte de \$\$

TAM - Gerente técnico da conta

## **PERSONAL HEALTH DASHBOARD**

- ◆ Alertas quando um serviço da AWS está fora do ar
- ◆ Criação de alertas que podem afetar na conta
- ◆ Recomendação do que deve ser feito

## **SUPPORT FORMS FOR - AWS Support Case**

- ◆ Encontro de abuso (enviado à equipe de abuso)
- ◆ Aumentar os limites
- ◆ Teste de penetração



# 3. Faturamento e Preços

## 4.1. e 4.2. Modelos de Preços da AWS e Estruturas de Contas

**PAGUE PELO QUE USAR, MAS SE QUISER PAGAR ANTES, PAGUE MAIS BARATO**

<u>ON-DEMAND</u>	<u>SPOT</u>	<u>RESERVED</u>
0% de desconto	até 90% OFF	até 72% OFF
Subo a máquina e uso. Pago pelo tempo que ela estiver rodando.	AWS não garante que ela vai estar sempre rodando, pode ser que caia.	Se compromete a pagar pela máquina por <b>1 ou 3 anos</b> , por isso ganha desconto.
<b>SEM PREVISÃO DE USO</b>	<b><u>Baseado na oferta e demanda da AWS</u></b>	Pode pagar antecipado ou não
	Indicado para procedimentos pequenos e que <b>possam falhar</b>	Pode ser reversível ou não. A reversível é mais cara
	<b>É INSTAVÉL/TOLERANTE A FALHA</b>	<b>É ESTÁVEL</b>

### INSTÂNCIA RESERVADA

- ◆ Pague adiantado ou não
- ◆ Tenha a opção de modificar a sua instância ou não
  - Quanto mais adiantado o pagamento, maior o desconto
  - Se escolher a opção de máquina convertível, fica mais caro

## AWS ORGANIZATIONS

### **FATURAMENTO CONSOLIDADO/BENEFICIOS COMPARTILHADOS**

- ◆ Agrupa contas AWS
- ◆ Billing unificado → vantagens e redução de custos (consolidated billing)
- ◆ **Benefícios:**
  - **Uma fatura** → você recebe uma fatura para várias contas
  - **Fácil rastreamento** → você pode rastrear as despesas em várias contas e fazer download dos dados combinados de custos e uso
  - **Uso combinado** → é possível combinar o uso em todas as contas da organização para compartilhar os descontos de preços por volume, os descontos de instância reservada e os savings plans
    - **Resultado** → custo mais baixo para o projeto, departamento ou empresa

- ◻ **Sem taxa extra** → não possui custo adicional
- ◆ **INSTÂNCIA RESERVADA + AWS ORGANIZATIONS**
  - ◻ Todas as contas da organização podem receber o custo benefício por hora das instâncias reservadas que são compradas por qualquer outra conta.
  - ◻ O compartilhamento do desconto pode ser desativado.

## FREE TIER

- ◆ 12 meses gratuitos EC2
  - ◻ - 750h por mês
- ◆ Sempre gratuito
- ◆ Teste/experimentação
  - ◻ Exemplo: 2 meses

## 4.3. Recursos Disponíveis para Suporte de Faturamento FORMAS DE OBTER SUPORTE E INFORMAÇÕES DE FATURAMENTO **COST EXPLORER**

### VISÃO DE CUSTO/FORECAST/DETALHAMENTO/HISTÓRICO

- ◆ Visualizar e compreender gastos na AWS.
- ◆ Visão do detalhe do custo
- ◆ Consegue ver o histórico de 12 meses Gera
- ◆ relatórios personalizados

## AWS COST AND USAGE REPORT

- ◆ Conjunto mais abrangente de dados de custo e uso. Possível ver a granularidade de data, horário e etc.
- ◆ **Publica os relatórios em um bucket S3**

## AWS QUICK SIGHT

### FERRAMENTA DE BI/ANÁLISE

- ◆ Pode se conectar com o cost and usage Ferramenta
- ◆ de BI que ingere dados e gera painéis

## BILLING SUPPORT CASE

Tire dúvidas sobre sua fatura [AMAZON CONCIERGE](#)

Suporte a nível empresarial para codenação de gastos

## AWS BUDGETS

### ALERTAS DE PREÇO/LIMITE DE USO

Configurar gastos na conta com **alertas de preço**

- ◆ TAGS: habilitar cost allocation tags
  - cost allocation tags → usadas para alocação de custos
- ◆ Consolidating billing → consolidar faturamento
- ◆ Pode enviar e-mail para **até 10 destinatários**

## INFORMAÇÃO DE PREÇOS DA

## AWS CALCULADORA MENSAL

### DA AWS

- ◆ Ferramenta de planejamento para criar **estimativas** para os casos de uso
- ◆ Para quem nunca usou a AWS
- ◆ Para quem quer reorganizar ou expandir
- ◆ SERVIÇO GRATUITO

## API DE PREÇOS DA AWS

- ◆ **API Price List Bulk** → consulta de preços de serviços da AWS em massa. Retorna em arquivo JSON ou CSV. Possui histórico de \$ .
- ◆ **API Price List Query** → consulta de informações específicas dos serviços, produtos e preços, em vez de recuperar preços em massa.
  - Obter informações de preços em ambientes que talvez não consigam processar uma lista de preços em massa
  - Não possui histórico de preço, só o atual

# 4. RECURSOS EXTRAS

## ATHENA

### ANÁLISE/BIG DATA/SEM SERVIDOR/ S3

Lê dados em S3, consulta por queries SQL.

## KINESIS

### STREAMING DE DADOS/REAL TIME

Streaming de dados real time e near real time.

- ◆ Fire hose → near real time, dado final gerenciado pelo firehose, totalmente gerenciado Data
- ◆ streams → real time, suporta múltiplos consumidores, configuração de shards

## AMAZON APPSTREAM

Streaming de aplicação que fornece aos usuários acesso instantâneo a suas aplicações de desktop de qualquer lugar.

## QUICK SIGHT

### FERRAMENTA DE BI

Ferramenta de BI para a criação de dashboards.

## INTEGRAÇÃO DE APLICAÇÕES:

## SQS

### FILA/DESACOPLAR/SONDAGEM

Serviço de Mensageria, poll de mensagens, standard ou FIFO.

Sistema de fila

**Só** ele armazena mensagens

## SNS

### A2A/A2P/GERENCIADO/TÓPICO

Serviço de mensageria, notificações, pub/sub e fan out.

Integração de sistemas.

## SES

E-mail. Envia de qualquer aplicação.

## COMPUTAÇÃO, CONTAINERS E

## ARMAZENAMENTO AWS GLUE

Serviço de integração de dados com tecnologia sem servidor para: **extrair, transformar e carregar dados de várias fontes** para análise, machine learning e desenvolvimento de aplicações.

## AWS BATCH

### EXECUÇÃO DE TAREFAS EM LOTE

Executa milhares de tarefas de computação em lote e em machine learning, enquanto otimizam recursos computacionais.

## **AMAZON WORKSPACES**Sirtualização de desktop.

## **AMAZON EKS**

Orquestração de contêineres kubernetes gerenciado.

## **EMR**

### **Solução de big data.**

Processamento de dados, **análise interativa e machine learning** que usa estrutura de código aberto, como: Apache, presto e outros workloads de big data e execução de map reduce.

## **FERRAMENTAS DO DEV**

### **CODE PIPELINE**

**Cria a esteira** de desenvolvimento CI/CD.

### **CODE COMMIT**

Gerenciamento de **código fonte**, estilo git.

### **CODE BUILD**

Compila e realiza os **testes**.

### **CODE DEPLOY**

Automatiza a **implementação** de código em qualquer instância.

### **CODE STAR**

**Compilação** - configura toda a cadeia de ferramentas de entrega contínua. Acompanha um painel unificado de projetos e uma integração com o software do JIRA da Atlassian.

Gerenciamento para criação e implementação de aplicações.

### **COMMIT + BUILD + DEPLOY**

### **CODE SUITE**

### **COMMIT + BUILD + DEPLOY + PIPELINE**

### **CODE ARTIFACT**

**Gerencia pacote de software** e suas dependências.

## AWS AMPLIFY

**Front-end** → conjunto de ferramentas para criação de plataforma web e móveis completas de front.

**Back-end** → configura e conecta uma aplicação em minutos.

## AWS STEP FUNCTIONS

Fluxo de trabalho visual usando pouco código para **orquestrar os serviços da AWS.**

**Automatiza fluxo de trabalho.**

## AWS MQ

Atende as tecnologias MQ (Apache e Rabbit).

Para sistemas **legados.**

## INTERAÇÃO COM CLIENTE

### AMAZON CONNECT

**CHAT ITERATIVO/SUORTE AO CLIENTE**

**Call center** na nuvem.

Central de contato em minutos. Pode ser dimensionada para oferecer suporte a milhões de clientes.

## ENTREGA DE CONTEÚDO

### API GATEWAY

Serviço gerenciado para criação de API. Única  
porta de entrada.

## SEGURANÇA, IDENTIDADE E CONFORMIDADE

### ARTIFACT

**RELATÓRIOS DE CONFORMIDADE**

Portal de relatórios de conformidade da AWS.

Portal de autoatendimento para recuperação de artefatos de auditoria que oferece aos clientes acesso sob demanda à documentação de conformidade e aos acordos da AWS.

Gratuito.

### CERTIFICATE MANAGER

Gerenciamento de certificados (SSL).

Criação e gerenciamento de certificados digitais.

Utilizado para **criptografar os dados entre usuários e os servidores** com base HTTPS.

## AWS KMS

Criação de chaves criptografadas **para uso em recursos da AWS**, geralmente para dados e arquivos no bucket S3.

## AWS CLOUD HSM

Gerencia chaves de criptografia.

## COGNITO

### CADASTRO E AUTENTICAÇÃO/ ARMAZENAMENTO DE SINCRONIZAÇÃO

Permite que você adicione facilmente o **cadastro e autenticação** de usuário as aplicações móveis e web.

Também atua com provedores externos compatíveis com SAML ou OPENID Connect.

- **Armazenamento de sincronização** → armazenamento de dados num banco local de chave-valor para depois realizar a sincronização.

Autenticação e autorização na aplicação (sistema de login).

## AWS SERVICE CATALOG

Governança e conformidade

Administra, cria, gerencia e distribui catálogo de produtos aprovados para usuários finais, que podem, acessar um portal personalizado.

Controle de quais users têm acesso a cada produto. Para

usuários finais.

## DETECTIVE

### INVESTIGATIVO/CAUSA RAIZ

Análise de dados para encontrar problemas de segurança (causa raiz).

# AWS SHIELD

Proteção **contra ataque DDoS.**

- ◆ Standard - camada 3 e 4.
- ◆ Advanced - camada 7.

# WAF

Firewall para **aplicações web, contra bots.**

<u>MACIE</u>	<u>GUARD DUTY</u>	<u>INSPECTOR</u>
Dados (proteção)	<b>Olha para a conta e recursos internos (S3)</b>	<b>Olha para EC2/APP e workloads</b>
Machine Learning	Monitoramento contínuo	Ataque
Padrões *	Atividades mal intencionadas	<b>Vulnerabilidade de software</b>
<b>Dados sensíveis</b>	<b>Não usual/estranho</b>	Exposição não intencional
Inf. no S3	Dados por eventos	Monitoramento contínuo
Dados classificados		

# ARMAZENAMENO

# AWS BACKUP

Centraliza e facilita a gestão de rotina de backups dos recursos da AWS.

# GERENCIAMENTO, MONITORAMENTO, SEGURANÇA E OUTROS



## **AWS CLOUD FORMATION**

### **Infra as code.**

Recursos em template.

Arquivos contendo o IAC, podem ser versionados e atualizados de forma automatizada.

## **CLOUD WATCH**

Logs da aplicação, alertas e métricas.

Também monitora as cobranças estimadas em sua conta, através da métrica.

## **AWS CONFIG**

Auditar e avaliar a configuração dentro da AWS. Possui

histórico.

Informa se tiver alguma configuração fora do padrão.

## EVENT BRIDGE - CLOUD WATCH EVENTS

Acionar eventos a partir de padrões ou datas, serverless.

Acesso em tempo real a alterações de dados (sem precisar de código)em: Serviços AWS

- APP
- SaaS

Barramento de eventos sem servidor que permite receber, filtrar, transformar, rotear e entregar eventos.

## LICENSE MANAGER

Facilitar o acesso e gerenciamento de **licença**.

## MANAGED SERVICES

Gerenciar **migração** e operação na nuvem.

## DMS

**Migração de banco de dados** on-premise para a AWS.

## DATA SYNC

Automatiza e otimiza processos de sincronização de dados entre data centers.

**Movimentação de dados entre on-premises e AWS.**

## SECRETS MANAGER

Armazenar chaves de segurança de forma segura.

## SYSTEM MANAGER

Obtenha insights operacionais sobre a AWS e recursos on-premises. Para nuvem híbrida.

Hub de operações da AWS com: gerenciamento de operações, aplicações, alterações e nós.

## PARAMETER STORE

Armazenar parâmetros da aplicação.

## AMAZON SAGEMAKER

Várias **ferramentas para o uso de machine learning** na AWS.

## AWS X-RAY

Rastreia solicitações de usuário enquanto percorrem toda a aplicação.

## AMAZON COMPREHEND

Processamento de linguagem natural.

## AMAZON POLLY

Converte texto em fala realista.

## AMAZON REKOGNITION

Análise de imagem e vídeo de seu app (detecta pessoas e etc).

## OBSERVAÇÕES

**IPV4** IP público possui limite de 5 por região.

**TRÁFEGO DE DE DADOS:**

**NÃO TEM CUSTO:**

- ♦ **ENTRADA** Em todas as regiões.
- ♦ **SAÍDA** **Dentro da mesma região**

Material de estudos de Rogério Tavares

[https://github.com/rogtavares/Estudos-CLF-C01\\_Cloud-Practitioner-2023-GT.git](https://github.com/rogtavares/Estudos-CLF-C01_Cloud-Practitioner-2023-GT.git)

## MATERIAIS UTILIZADOS simulados UDEMY

SIM PORTUGUES 4X 65

SIM ANDRE IACONO UDEMY **4X60**

SIM STEPHANI UDEMY **4X65**

SIM NEAL UDEMY **4X65**

SIM RANGA UDEMY **4X65**

<https://www.udemy.com/>

# Simulados FREE

Perguntas práticas gratuitas com respostas

## Lista de exames

- [Exame prático - 1](#)
- [Exame prático - 2](#)
- [Exame prático - 3](#)
- [Exame prático - 4](#)
- [Exame prático - 5](#)
- [Exame prático - 6](#)
- [Exame prático - 7](#)
- [Exame prático - 8](#)
- [Exame prático - 9](#)
- [Exame prático - 10](#)
- [Exame prático - 11](#)
- [Exame prático - 12](#)
- [Exame prático - 13](#)
- [Exame prático - 14](#)
- [Exame prático - 15](#)
- [Exame prático - 16](#)
- [Exame prático - 17](#)
- [Exame prático - 18](#)
- [Exame prático - 19](#)
- [Exame prático - 20](#)
- [Exame prático - 21](#)
- [Exame prático - 22](#)
- [Exame prático - 23](#)

## Arquitetando na AWS

- [Serviços de nuvem AWS \(catálogo\)](#)

- [Mapa de infraestrutura de nuvem global da AWS](#)
- [Regiões de infraestrutura global da AWS](#)
- [Saiba como protegemos os datacenters da AWS](#)
- [Abstrações de computação na AWS: uma história visual](#)
- [AWS Well-Architected - Aprenda, meça e crie usando as melhores práticas de arquitetura](#)
- [Estrutura bem arquitetada da AWS](#)
- [Lente do setor de serviços financeiros - AWS Well-Architected Framework](#)
- [Ferramenta bem arquitetada da AWS – analise as cargas de trabalho em relação às práticas recomendadas](#)
- [Ferramenta bem arquitetada da AWS – DEMO](#)
- [Nove maneiras de reduzir sua conta da AWS](#)
- [Desmistificando as 9 formas de reduzir custos na AWS](#)
- [Conhecendo um pouco mais sobre o Graviton2](#)
- [Amazon S3 + Amazon CloudFront: uma combinação feita na nuvem](#)
- [Como configurar uma distribuição do CloudFront para Amazon S3](#)
- [Visão geral do gerenciamento de acesso no S3](#)
- [Mergulhando fundo na consistência do S3](#)
- [Plano de depreciação do Amazon S3 Path – o resto da história](#)
- [Tipos de instância do Amazon EC2](#)
- [Consultor de Instância Spot](#)
- [Execução de aplicativos da web de grande escala em instâncias spot do Amazon EC2](#)
- [Ganchos de ciclo de vida do Amazon EC2 Auto Scaling](#)
- [Elastic Load Balancing e Amazon EC2 Auto Scaling](#)
- [Tipos de volume do Amazon EBS](#)
- [Recuperar arquivos de um backup de volume do Amazon EBS](#)
- [Novo – Multi-Attach para IOPS provisionadas \(io1\) Amazon EBS Volumes](#)
- [Posso usar volumes EBS Multi-Attach para permitir que várias instâncias do EC2 acessem simultaneamente um sistema de arquivos padrão?](#)
- [Trabalhando com clusters multimestres do Aurora](#)
- [Amazon Aurora Backtrack – Volte no tempo](#)
- [Amazon DynamoDB - Escolhendo configurações iniciais de taxa de transferência](#)
- [Práticas recomendadas para projetar e arquitetar com o DynamoDB](#)
- [Práticas recomendadas para projetar e usar chaves de partição com eficácia](#)
- [Construindo resiliência em escala no Tinder com o Amazon ElastiCache](#)
- [Como trabalhar com o Modo Cluster no Amazon ElastiCache para Redis](#)
- [Comparando Redis e Memcached](#)
- [Casos de uso comuns do ElastiCache e como o ElastiCache pode ajudar](#)
- [Apresentando o Amazon Elasticsearch Service como destino no AWS Database](#)

## Migration Service

- Migração concluída – o negócio de consumo da Amazon acaba de desativar seu banco de dados Oracle final
- Dimensionar sua instância do Amazon RDS vertical e horizontalmente
- Instâncias do Amazon RDS do tamanho certo em escala com base nas métricas de desempenho da Oracle
- Usando réplicas de leitura na região no Amazon RDS for SQL Server
- Novo – Dispositivo de hardware do AWS Storage Gateway
- Disponibilidade expandida e pedido simplificado do AWS Storage Gateway Hardware Appliance
- Usando endereços IP estáticos para Application Load Balancers
- Novo – Application Load Balancer simplifica a implantação com grupos-alvo ponderados
- Como publicar com segurança aplicativos da Internet em escala usando o Application Load Balancer e o AWS PrivateLink
- Ferramenta de depuração para conectividade de rede do Amazon VPC
- Conectividade multi-VPC de várias regiões
- Trabalhando com Direct Connect Gateways
- Parceiros do AWS Direct Connect
- Como provisionar uma conexão do AWS Direct Connect?
- Construindo um sistema DNS dinâmico sem servidor com a AWS
- Como crio uma política do IAM para controlar o acesso aos recursos do Amazon EC2 usando tags?
- Simplifique a concessão de acesso aos seus recursos da AWS usando tags em usuários e funções do AWS IAM
- Escrevendo políticas do IAM: conceda acesso a pastas específicas do usuário em um bucket do Amazon S3
- Políticas de IAM e políticas de bucket e ACLs! Oh meu Deus! (Controle de acesso aos recursos do S3)
- Agora crie e gerencie funções IAM da AWS com mais facilidade com o console IAM atualizado
- Diretrizes para proteger sua conta da AWS ao usar o acesso programático
- Criando um alarme de cobrança para monitorar suas cobranças estimadas da AWS
- Como monitorar melhor suas métricas de aplicativos personalizados usando o Amazon CloudWatch Agent
- Exemplos de expressões de pesquisa do CloudWatch
- Atualizar para o Amazon EventBridge a partir do Amazon CloudWatch Events
- Consumidor de assinatura do CloudWatch Logs + Elasticsearch + Kibana Dashboards
- Execução de scripts do GitHub e Amazon S3

- Validar modelos do AWS CloudFormation
  - Amazon CloudFront – suporte para conteúdo dinâmico
  - Entrega de conteúdo dinâmico do Amazon CloudFront
  - Posso usar uma única distribuição na Web do CloudFront para fornecer conteúdo de várias origens usando vários comportamentos?
  - Considerações de implementação - Aprenda padrões de design de cache
  - Cache de conteúdo com base em parâmetros de string de consulta
  - Amazon API Gateway - Habilitando o cache da API para melhorar a capacidade de resposta
  - Capacitando aplicativos de jogos com o Amazon DynamoDB
  - Amazon DynamoDB: casos de uso de jogos e padrões de design
  - Modelo de aplicativo sem servidor da AWS
  - Apresentando os pipelines do AWS SAM: gere automaticamente pipelines de implantação para aplicativos sem servidor
  - Firecracker – Virtualização leve para computação sem servidor
  - Vídeo sob demanda na AWS
  - Backups entre regiões
  - macaco do caos
  - Como encerro minha conta da AWS?
- 

E-mail para contato [rogerio@almeidatavares.com.br](mailto:rogerio@almeidatavares.com.br)