

# Dominando Pipelines de Dados na AWS

Este curso é uma jornada abrangente através dos conceitos fundamentais de engenharia de dados e serviços AWS.

Nele vamos aprender a construir pipelines de dados eficientes usando Python, Terraform e Github em duas arquiteturas comuns na AWS.

Domine os fundamentos da computação em nuvem e segurança AWS enquanto você mergulha no mundo do armazenamento S3, SQS, contêineres Fargate e EKS, além de bancos de dados Redshift e Athena. Este curso é perfeito para quem quer dar o próximo passo na engenharia de dados na nuvem.

# Quem é Anthony Rodrigues?

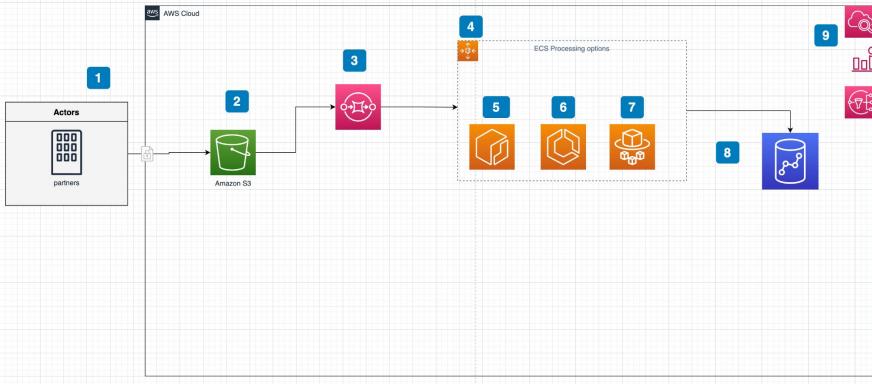
- 10+ anos de experiência, entre desenvolvimento e engenharia de dados.
- Bacharel em Sistemas de Informação pela UVA (Veiga de Almeida) - PROUNI
- Certificação em Arquitetura de Dados pela Udacity
- Carioca, morando em SP.
- Engenheiro de dados na Podchaser.



# Arquiteturas a serem criadas no curso.

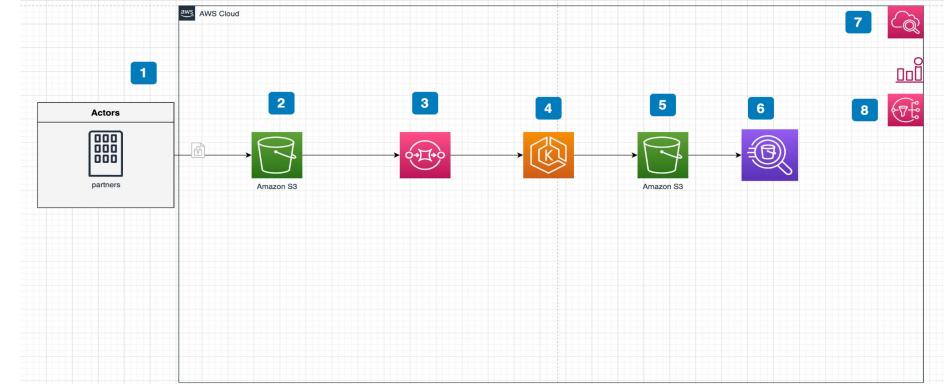
## Stack Academy ETL

AWS ETL for first course



## Stack Academy ETL

AWS ETL for second course



# O que precisaremos para o curso:

- Anaconda
- Conta AWS
- AWS cli
- GIT
- Terraform
- Python 3.6+
- Docker
- IDE (VSCode, PyCharm, Sublime, etc.)

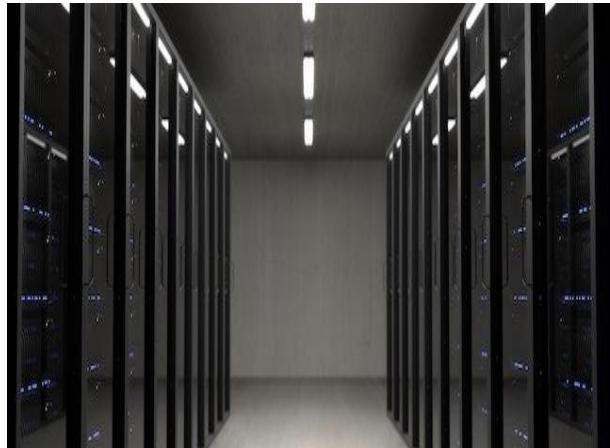
# Desmistificando a Engenharia de Dados.

## A Arte e a Ciência por trás dos Dados

**Engenharia de Dados é um campo de estudo que se concentra na coleta, validação, armazenamento, transformação e processamento de dados, para torná-los acessíveis e úteis para análise. Os engenheiros de dados criam a infraestrutura e as ferramentas necessárias para o manuseio de grandes volumes de dados, permitindo que analistas de dados e cientistas de dados realizem seu trabalho de forma mais eficaz.**



# On-premise vs Cloud



- Principais diferenças:**
- Escalabilidade
  - Flexibilidade
  - Segurança
  - Modelo de pagamento



# Infraestrutura como Código (IaC): Simplificando o Gerenciamento de Recursos na Nuvem

**A Infraestrutura como Código (IaC) mudou a maneira como provisionamos e gerenciamos os serviços na nuvem.**

**Com a IaC, podemos descrever e implantar toda a infraestrutura por meio de código, utilizando linguagens de programação e ferramentas especializadas.**

**Essa abordagem permite a automação dos processos de provisionamento, configuração e gerenciamento de recursos, proporcionando agilidade, consistência e rastreabilidade em todo o ciclo de vida da infraestrutura.**

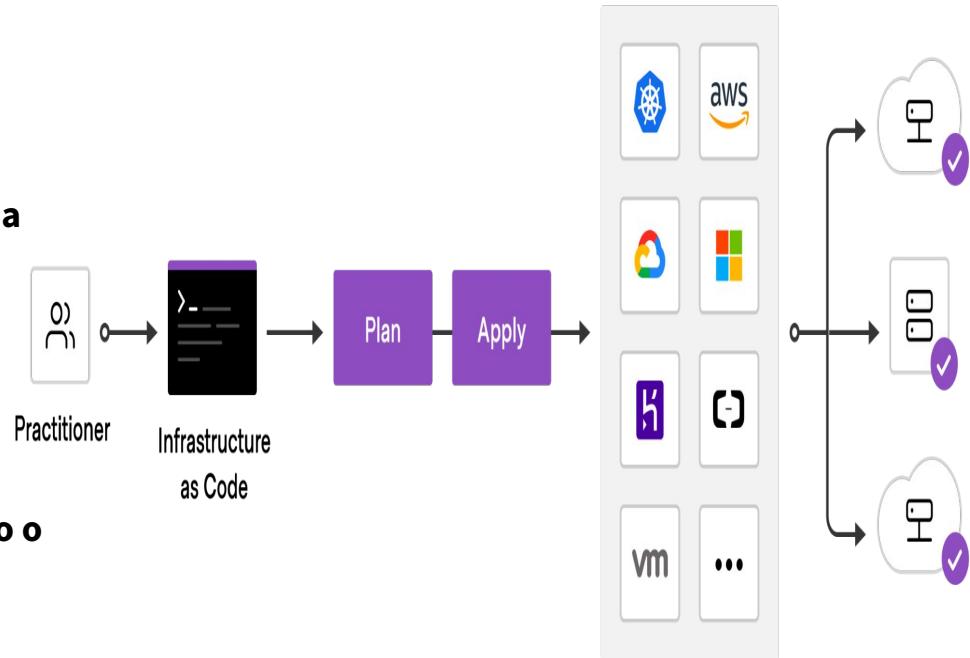
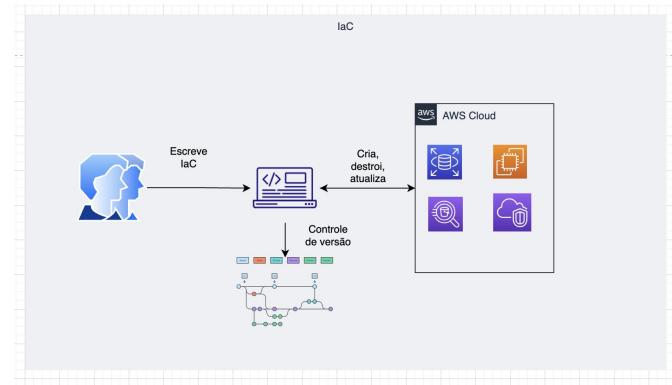


Image source:  
<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/infrastructure-as-code>

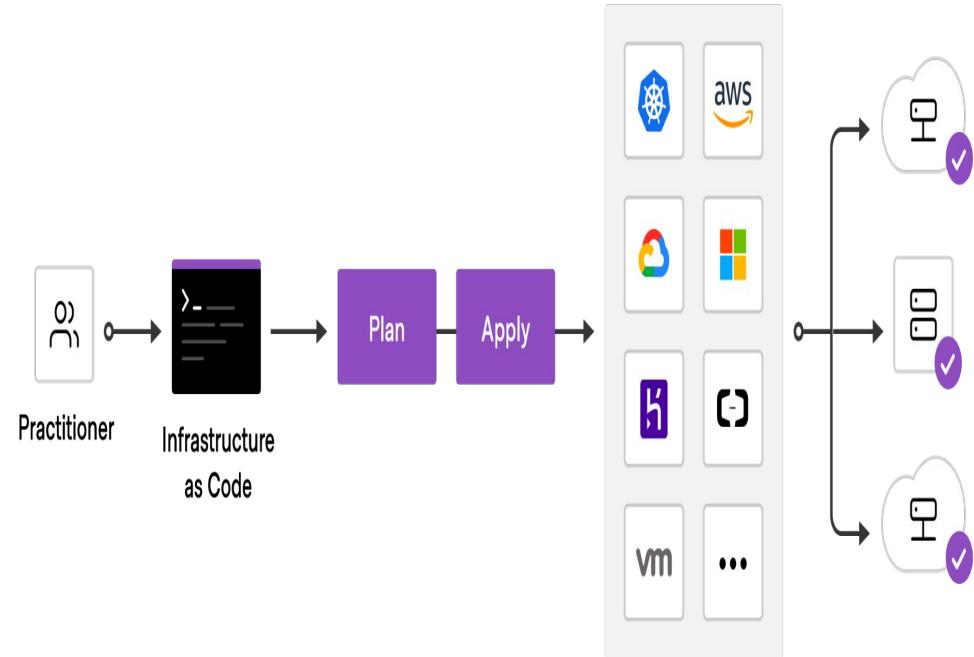
# Os Três Pilares do IaC: Consistência, Reprodutibilidade e Controle de Versão

O IaC proporciona consistência, minimizando a possibilidade de erros humanos e estabelecendo uma configuração de infraestrutura uniforme. A reproduzibilidade é otimizada, com a habilidade de erguer e derrubar rapidamente ambientes completos. E com controle de versão, as mudanças na infraestrutura podem ser rastreadas e gerenciadas como qualquer código de software.



# Entendendo o Terraform

Terraform é uma ferramenta IaC que facilita a construção, alteração e versionamento de infraestrutura de maneira segura. Ele se destaca por ser agnóstico em relação ao provedor, permitindo trabalhar com uma variedade de serviços de cloud, e pela sua sintaxe declarativa que descreve de forma concisa e clara o estado desejado da infraestrutura

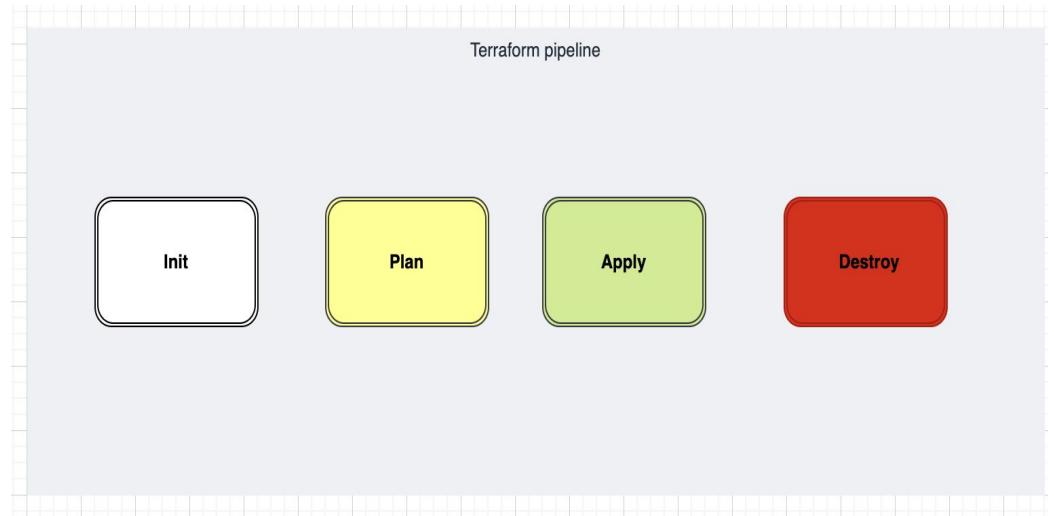


# O Fluxo de Trabalho do Terraform

Iniciar, Planejar, Aplicar e Destruir.

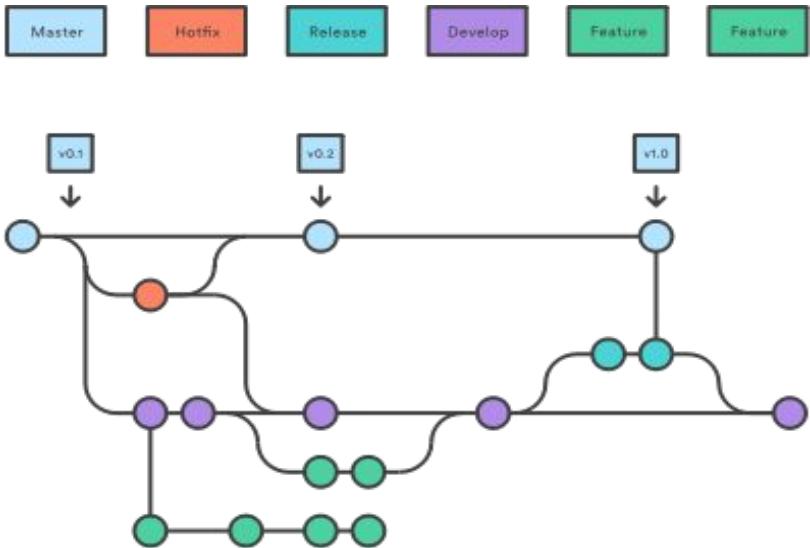
**O fluxo de trabalho do Terraform se baseia em quatro etapas principais:**

- 1. Init:** Essa fase o terraform inicia o projeto e baixa configurações iniciais.
- 2. Plan:** Terraform gera um plano de execução descrevendo o que será feito para atingir o estado desejado.
- 3. Apply:** Terraform executa as etapas planejadas para atingir o estado desejado.
- 4. Destroy:** Os recursos criados são removidos quando não são mais necessários.



# Git

**Git é um sistema de controle de versão distribuído que permite rastrear as mudanças em arquivos de computador e facilita a colaboração entre várias pessoas. É uma ferramenta indispensável no mundo do desenvolvimento de software, permitindo gerenciar e rastrear efetivamente as diferentes versões do seu código.**

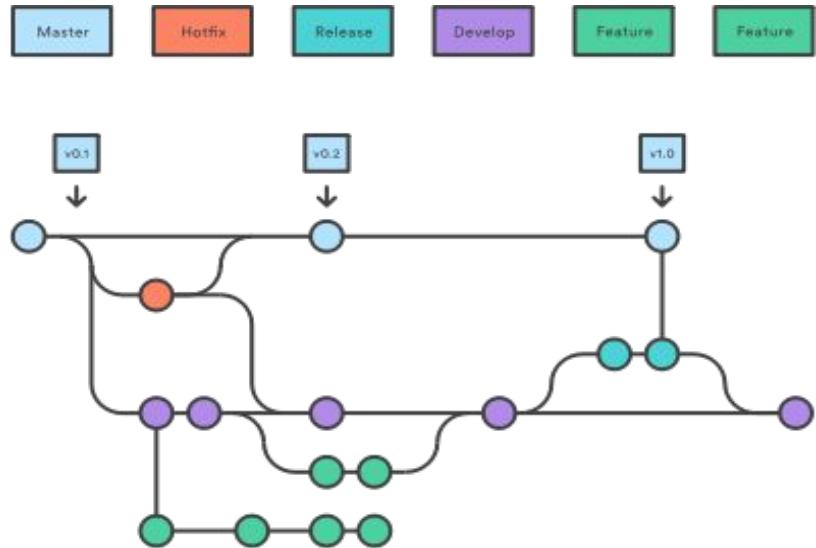


# Compreendendo os Fundamentos do Git.

O fluxo de trabalho do Git se inicia nas branches, que são "ramificações" independentes do código-fonte principal, permitindo que diferentes versões e alterações sejam desenvolvidas separadamente do branch principal.

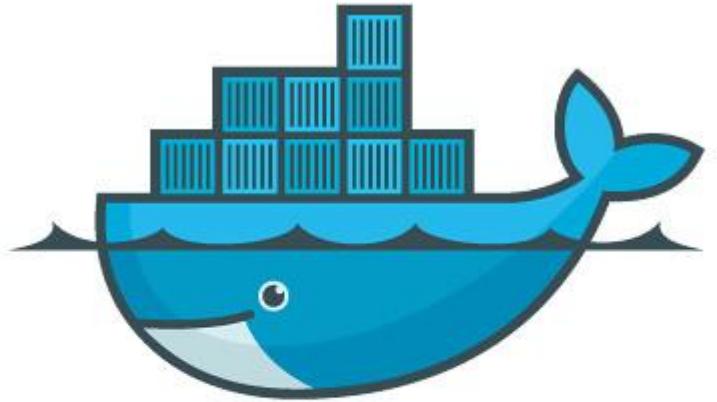
**Quando terminamos de fazer as mudanças executamos o "commit" que registra nossas alterações no repositório.**

**Por último fazemos o "push" onde  
compartilhamos nossas interações uns com os  
outros.**



# Docker: Descobrindo a plataforma de conteinerização

Docker é uma plataforma que facilita o desenvolvimento, a distribuição e a execução de aplicações por meio da conteinerização. Os containers Docker permitem que você empacote sua aplicação e suas dependências juntas, garantindo a sua execução em diversos ambientes computacionais.



**docker**

# Desvendando Containers: A Base do Docker

**Containers são a base do Docker. Eles são estruturas que empacotam o código e as dependências de uma aplicação, garantindo que ela seja executada de maneira confiável em diversos ambientes computacionais. Em outras palavras, um container garante a consistência da aplicação, independente do ambiente de execução.**



# O gigante dos serviços em nuvem: AWS

**A Amazon Web Services (AWS) é a plataforma de serviços em nuvem mais abrangente e amplamente adotada no mundo. Lançada em 2006, a AWS oferece mais de 200 serviços completos de data centers em todo o mundo.**



# Por que tantas empresas escolhem a AWS?

A AWS oferece diversas vantagens que a tornam uma escolha popular para empresas de todos os tamanhos. A escalabilidade e flexibilidade, a ampla gama de serviços, e a segurança robusta são algumas das principais razões para a popularidade da AWS.



# Entendendo a infraestrutura global e os principais serviços da AWS

A AWS possui uma infraestrutura global robusta, que é dividida em Regiões, Zonas de Disponibilidade e Pontos de Presença. Além disso, oferece uma variedade de serviços de computação e armazenamento, como EC2, Lambda, S3 e EBS.



# AWS Infraestrutura Global

O que são as Regiões, zonas de disponibilidade e Pontos de Presença?

**Zona de disponibilidade é um data center altamente redundante e isolado, com infraestrutura de energia, rede e resfriamento própria.**

**Região é um conjunto de AZs em uma região geográfica isolada.**

**Pontos de Presença (Edge Locations) são "pequenos" data-centers estrategicamente distribuídos e atuam como caches.**



# Principais serviços AWS

- **Computação**
- **Armazenamento**
- **Banco de dados**
- **Redes e entrega de conteúdo**
- **Análises**
- **Machine Learning**
- **Segurança, identidade e conformidade**



# Princípios da Cobrança da AWS

Compreendendo os Modelos de Preço e a Estratégia Pay-as-you-go

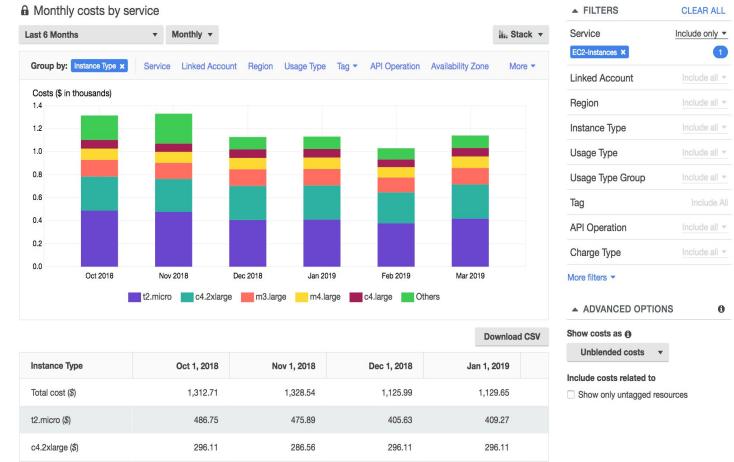
**A AWS segue um modelo de precificação pay-as-you-go, onde os usuários pagam apenas pelos recursos que utilizam, sem compromissos de longo prazo ou taxas iniciais. Este modelo é aplicado para a maioria dos serviços da AWS. Além disso, a AWS cobra pelo tráfego de dados, especialmente os dados que saem da AWS para a internet ou entre regiões da AWS.**



# Ferramentas de Gerenciamento de Custo

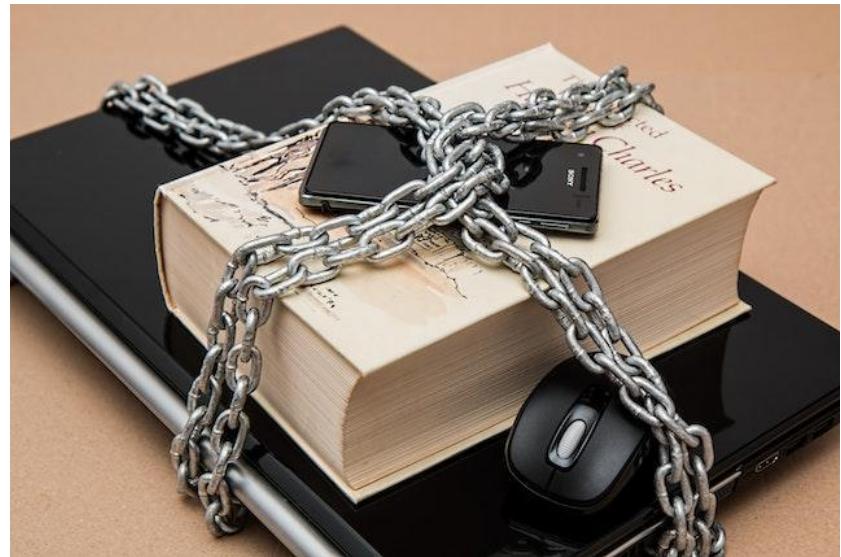
Monitorando e Controlando seus Custos na AWS

A AWS fornece várias ferramentas para ajudar os usuários a monitorar e controlar seus custos, incluindo o AWS Cost Explorer, AWS Budgets e AWS Cost and Usage Report. Cada uma dessas ferramentas tem um propósito específico e pode ser usada para identificar tendências, detectar deficiências e otimizar custos.



# Introdução à Segurança na AWS

- Importância da segurança na nuvem
- Visão geral das ofertas de segurança da AWS
- IAM (Identity and Access Management): Users, Groups, Roles, Policies
- Práticas de segurança



# Entendendo a segurança compartilhada na AWS

**A AWS segue um modelo de responsabilidade compartilhada para a segurança. A AWS é responsável pela segurança "da" nuvem - infraestrutura física, software, redes e instalações - enquanto os usuários são responsáveis pela segurança "na" nuvem - proteção de dados, contas, acessos e informações pessoais.**



# O que é IAM e para que serve?

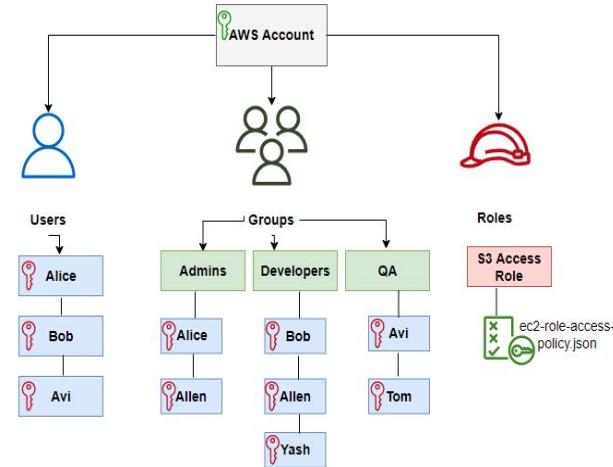
Gerencie Identidades, Grupos e Políticas para uma Governança Eficiente de Acessos

O IAM (Identity and Access Management) permite gerenciar de forma centralizada identidades e acessos aos recursos da plataforma. Ele fornece recursos como usuários, grupos, funções (roles) e políticas para controlar quem pode fazer o quê em uma conta da AWS.

Com o IAM, você pode criar usuários e atribuir a eles permissões específicas para acessar e gerenciar recursos. Os usuários podem ser agrupados em grupos, o que simplifica a atribuição e a gestão de permissões em grande escala.

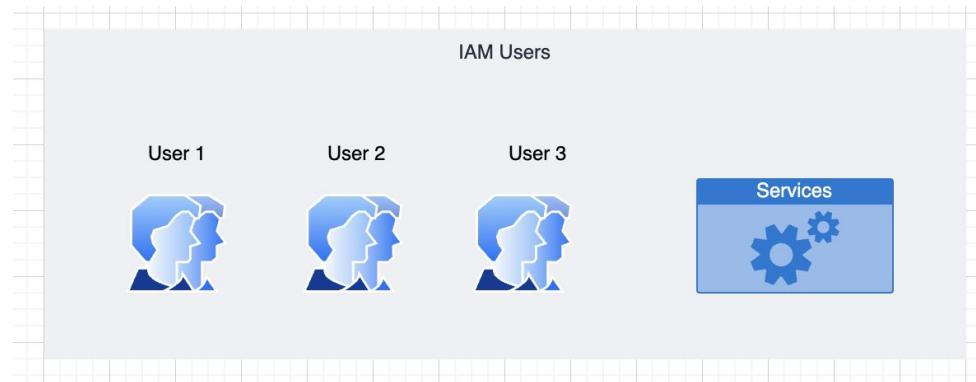
Além disso, as funções (roles) permitem definir permissões temporárias para serviços ou aplicativos, permitindo acesso seguro e granular a recursos específicos por um determinado período de tempo.

As políticas de acesso, por sua vez, são documentos que definem as permissões que podem ser associadas a usuários, grupos ou funções.



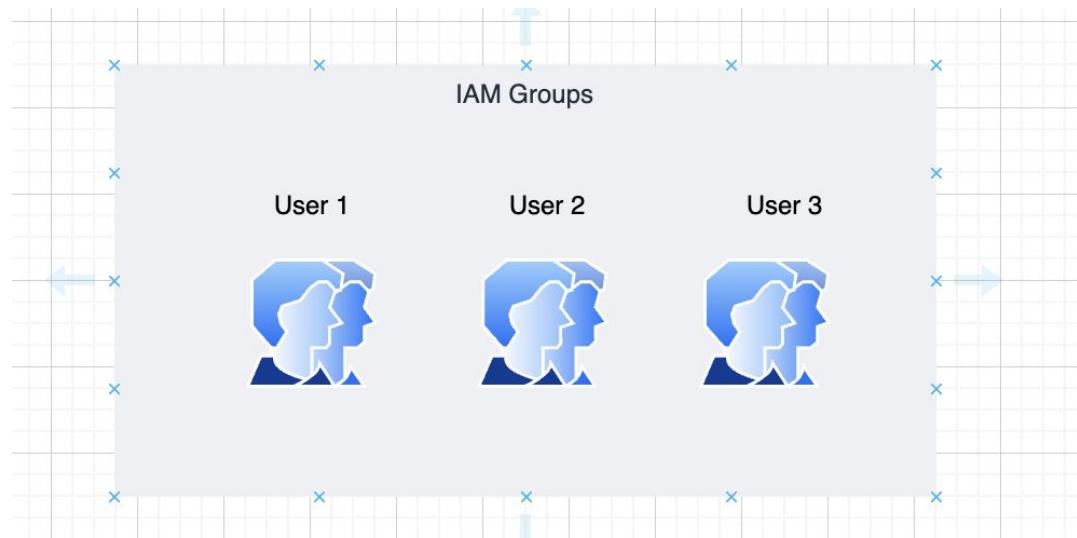
# Compreendendo Usuários IAM e Seu Uso na AWS

**Usuários IAM são representações de indivíduos ou aplicações no ecossistema da AWS. Cada usuário IAM possui suas próprias credenciais de segurança, permitindo que eles acessem os serviços e recursos da AWS de forma segura e controlada.**



# Grupos IAM: Simplificando a Gestão de Permissões na AWS

Grupos IAM são usados na AWS para gerenciar conjuntos de usuários com as mesmas permissões. Eles oferecem uma maneira simples e segura de administrar permissões para múltiplos usuários, facilitando a gestão do controle de acesso. A melhor prática sugere a organização de usuários em grupos, baseada em suas necessidades de acesso.



# Roles: Permitindo a Interação Segura de Serviços na AWS

Roles são uma ferramenta poderosa que permite aos serviços AWS assumirem permissões para acessar recursos que precisam. Eles proporcionam uma maneira segura de delegar permissões, sem a necessidade de compartilhar credenciais permanentes, sendo amplamente usados para permitir que serviços AWS interajam entre si de forma segura.

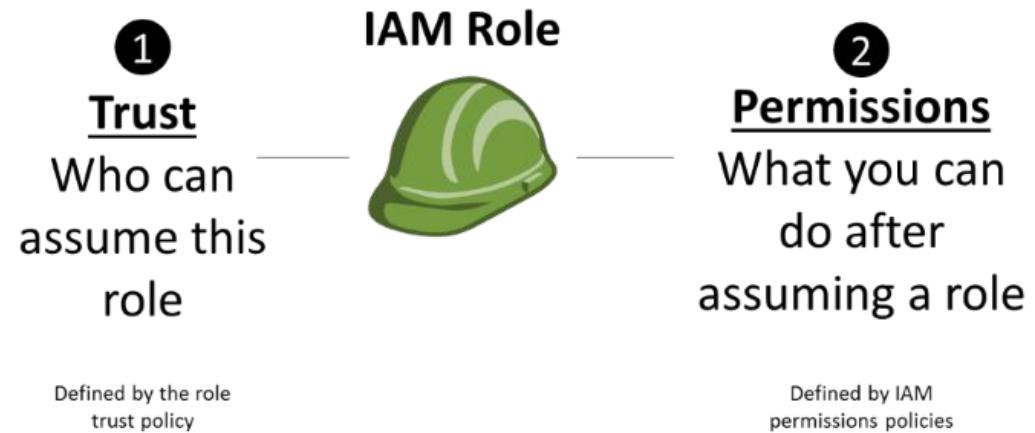


Image source: AWS

# IAM Policies: Definindo Permissões na AWS

**As IAM Policies são usadas para definir permissões para usuários, grupos e papéis na AWS.**

**Elas são documentos JSON que detalham o que é permitido ou negado em relação a recursos específicos.**

**Ao definir políticas, é crucial garantir que apenas as permissões necessárias sejam concedidas, em linha com o princípio do mínimo privilégio.**

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Stmt1684698239753",  
      "Action": [  
        "sns:DeleteMessage",  
        "sns:GetQueueAttributes",  
        "sns:ReceiveMessage"  
      ],  
      "Effect": "Allow",  
      "Resource": "arn:aws:sns:us-east-1:123:abc"  
    }  
  ]  
}
```



# Melhores práticas de segurança ao criar uma conta na AWS

**Uma segurança efetiva com IAM envolve várias práticas recomendadas. Algumas dessas práticas são:**

- **Não use a conta root.**
- **Ative a autenticação de dois fatores (MFA).**
- **Crie usuários IAM com o princípio de mínimo privilégio.**
- **Mantenha suas credenciais de segurança seguras.**
- **Garanta a mudança periódica das credenciais.**



# Princípio do Mínimo Privilégio: Maximizando a Segurança

O Princípio do Mínimo Privilégio é um conceito central na segurança de TI, que propõe que cada usuário, sistema ou processo deve ter apenas as permissões necessárias para realizar suas funções.

Adotar este princípio ajuda a minimizar a exposição a riscos e a mitigar possíveis falhas de segurança.



# Criação de Conta na AWS

- Email válido
- Telefone
- Cartão de crédito ativo para compras internacionais



### Cadastrar-se na AWS

Explore os produtos do nível gratuito com uma nova conta da AWS.

Para saber mais, acesse [aws.amazon.com/free](http://aws.amazon.com/free).



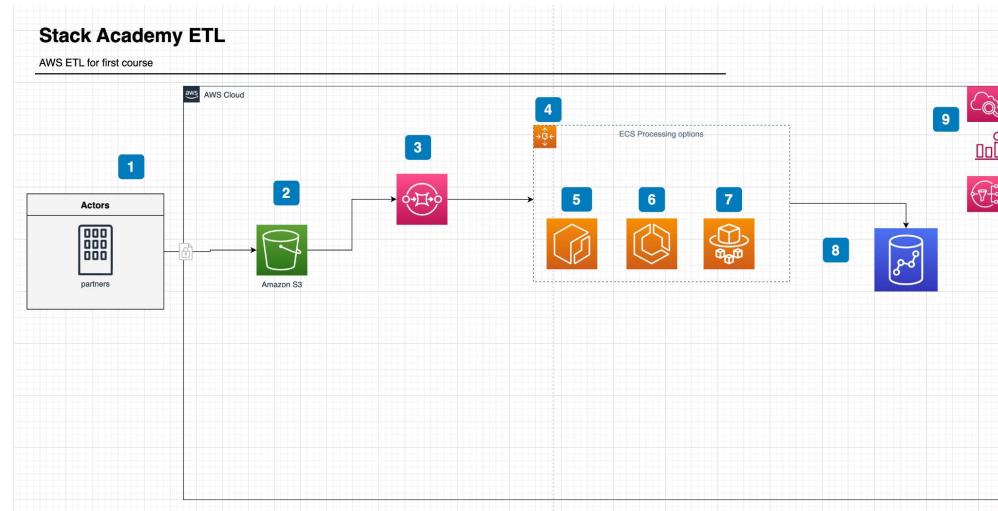
Endereço de e-mail do usuário raiz  
Usado para recuperação de contas e algumas funções administrativas

Nome da conta da AWS  
Escolha um nome para a sua conta. Você pode mudar o nome nas configurações de sua conta após o cadastro.

OU

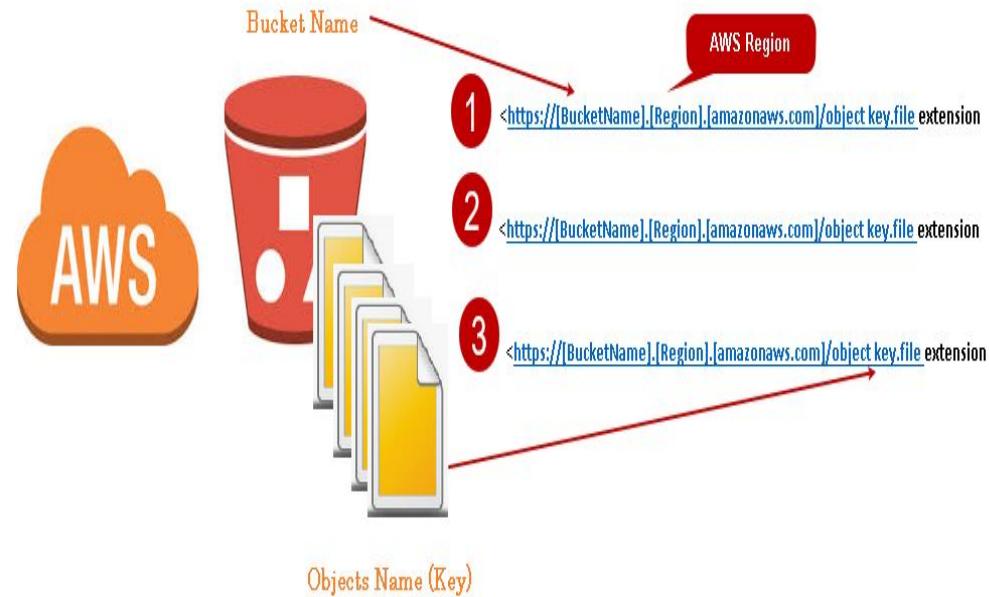
# S3, SQS, Fargate, e Redshift: Nossa Primeira Arquitetura

Vamos construir nossa primeira arquitetura de pipeline de dados usando o AWS S3 para armazenamento, SQS para filas de mensagens, Fargate para execução de contêineres e Redshift para análises de dados. Além disso, integramos o CloudWatch para garantir a coleta eficiente de logs e monitoramento do pipeline.



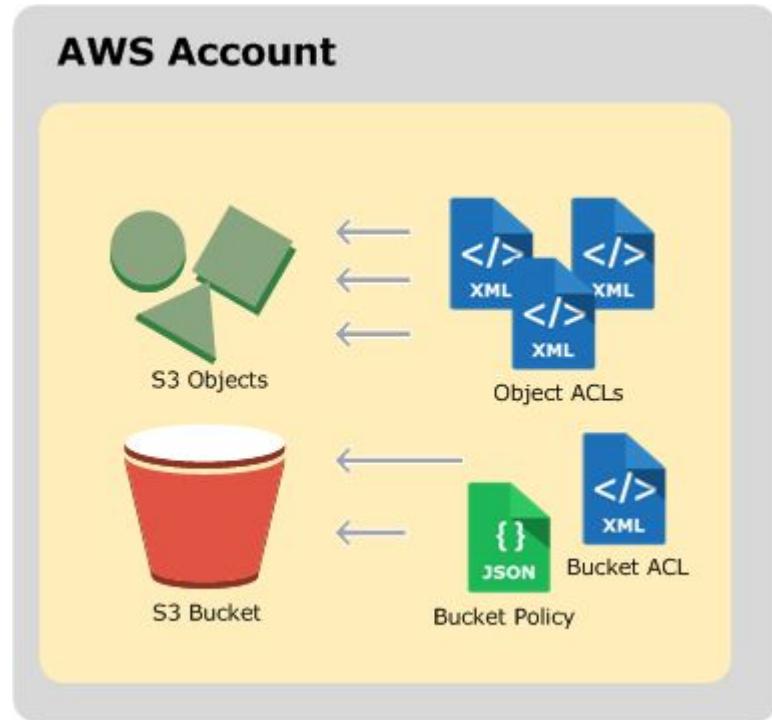
# S3: Armazenamento na Nuvem Simplificado

O Amazon S3, ou Simple Storage Service, é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade de dados, segurança e performance. Ele é usado para uma variedade de casos, como backup e restauração, arquivamento de dados, distribuição de conteúdo, data lake e mais recentemente, para armazenamento de sites estáticos.



# Buckets S3: Armazenamento e Organização de Dados

**Buckets são as unidades básicas de armazenamento no Amazon S3. Eles funcionam como contêineres onde você pode armazenar seus objetos de dados, organizar seus recursos e definir permissões de acesso. Cada bucket possui um nome único dentro do S3.**



# Classes de Armazenamento do Amazon S3

O S3 fornece várias classes de armazenamento, incluindo Standard, Intelligent-Tiering, One Zone-IA, Glacier e Glacier Deep Archive. A escolha da classe de armazenamento depende da frequência de acesso aos dados e do custo desejado. Cada classe tem um preço de armazenamento.

**Standard:** Esta é a classe de armazenamento mais cara, mas também a mais acessível e durável. É melhor para dados que precisam ser acessados com frequência.

**Intelligent-Tiering:** Esta classe move automaticamente os dados entre duas camadas (frequente e infrequente) com base nos padrões de acesso. Se os padrões de acesso mudarem, ele se ajusta automaticamente. A taxa de armazenamento é um pouco mais baixa que a classe Standard.

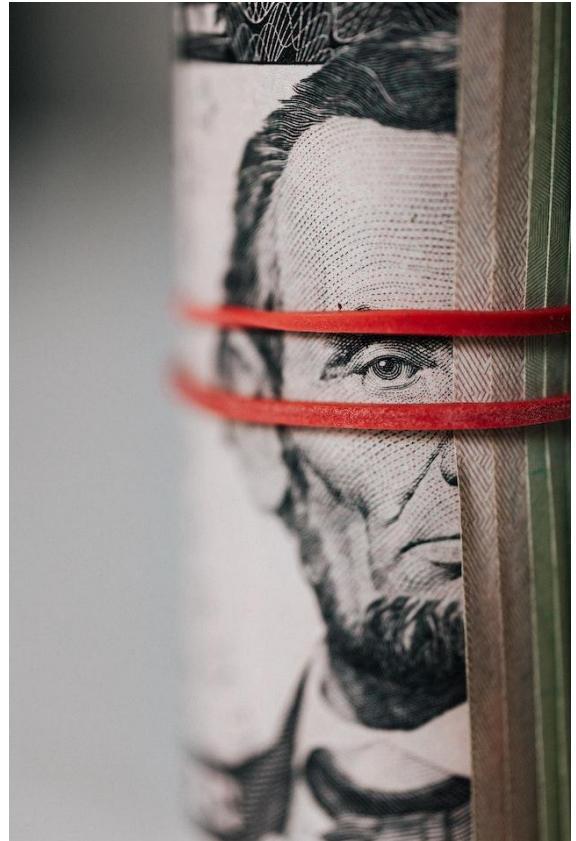
**One Zone-IA:** Esta classe é usada para dados que são acessados com menos frequência, mas que exigem rápida recuperação quando necessário. Ele armazena dados em uma única zona de disponibilidade, tornando-o menos durável, mas mais barato que as classes Standard e Intelligent-Tiering.

**Glacier e Glacier Deep Archive:** Essas classes são usadas para arquivamento de longo prazo de dados. Glacier é ideal para dados que os clientes podem precisar acessar em minutos ou horas, enquanto o Glacier Deep Archive é a opção mais econômica para a retenção de dados por 7-10 anos, com tempo de recuperação em horas.

# Cobrança S3

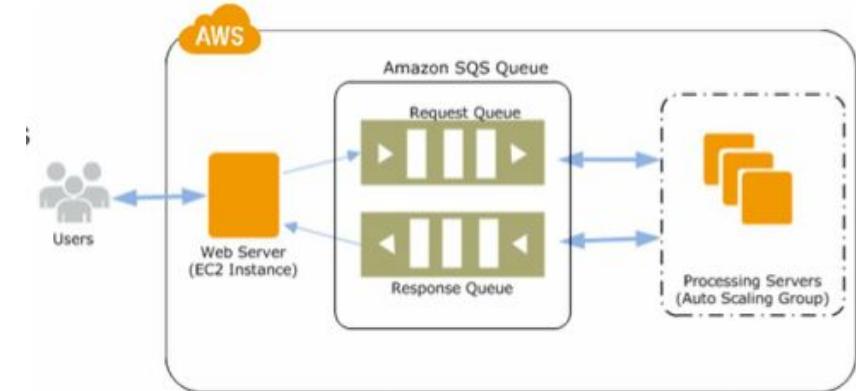
A precificação do Amazon S3 é determinada por vários fatores, incluindo a classe de armazenamento que você escolhe, a quantidade de dados armazenados, a quantidade de dados transferidos e recursos adicionais, como o Amazon S3 Glacier para arquivamento de longo prazo.

Além disso dados que saem da AWS para internet ou outra zona, tem tráfego pago.



# Compreendendo o Amazon SQS

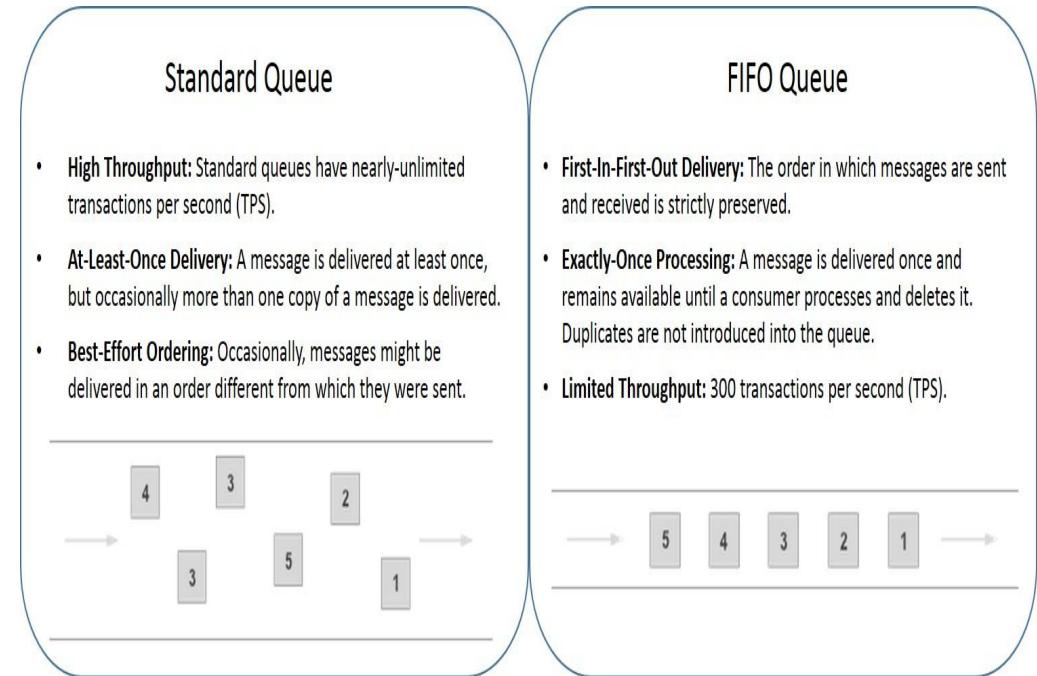
O Amazon Simple Queue Service (SQS) é um serviço de fila de mensagens totalmente gerenciado que permite que você integre e desacople componentes de aplicativos distribuídos. É amplamente utilizado para lidar com qualquer volume de dados, sem perder mensagens ou exigir que outros serviços estejam sempre disponíveis.



# Tipos de Filas no SQS da Amazon: Padrão vs FIFO

**Amazon SQS oferece dois tipos de filas: padrão e FIFO.**

- **Padrão:**
  - Garantem a entrega de mensagens pelo menos uma vez, com alta taxa de transferência
- **FIFO:**
  - Garantem que a ordem da mensagem seja preservada e que cada mensagem seja processada exatamente uma vez.



Source image:  
<https://aws.amazon.com/blogs/developer/using-python-and-amazon-sqs-fifo-queues-to-preserve-message-sequencing/>

# Amazon SQS: Entendendo as Mensagens

**As mensagens no Amazon SQS são os pacotes de informação enviados de um componente do sistema para outro. A vida de uma mensagem inclui etapas de envio, recebimento e exclusão. Importante também é o conceito de tempo de visibilidade, que define a janela de tempo durante a qual a mensagem fica invisível para outros consumidores após ter sido recebida.**

```
{  
  "MessageId": "a0e32be5-d91a-4062-8feb-2a8d7a44d5fa",  
  "ReceiptHandle": "AQEB7DN+JuqmYwSQAJSAdgizBzCPb3+s9e1zfjfQ5SuprfYvsF7A4CeihDhp6gdjqYTc8dLrUVGSHTQhC7mRIq3AwadXQu8axKaiFq0HxPmdJtgehylnWazqFdEGR2mZziX2x+/f840r00bVl0eva1JUqs1HHxcstr65U/  
UK3R7vD2Edstw5bDZgnjytU3lUqZNejqKnPkdW56tDujbpmc4XPWGSMVwXrPx0ySTh1lIFI7if195eHutXSdyRuE3mBXy3zHBIsIEG/4pGiSbMYEcKMPPKKWBmE+90DyXL0MfaqYK7znVRILiM72PuhsaCqKjwbKdlcJpMjm2JVFRvtWTTgda3kno/  
zu2BVbcx7i2UudrFrGrVEJNAtbzSkVPQah",  
  "MD5ofBody": "10f606dadd7a2b25Wa4c848b508eda8d",  
  "Body": "{\"id\": 1, \"data\": \"abc\"}",  
  "Attributes": {  
    "SenderId": "AIDAIDCSJPNZ6YTV6NUC",  
    "ApproximateFirstReceiveTimestamp": "168485637621",  
    "ApproximateReceiveCount": "1",  
    "SentTimestamp": "168357615280"  
  }  
}
```

# Cobrança SQS

O modelo de precificação do Amazon SQS é bastante simples e direto. Ele é baseado no número de solicitações feitas ao serviço, seja para enviar, receber ou excluir mensagens.

Os principais componentes da precificação do SQS incluem:

- **Solicitações de mensagens:** Você paga por cada solicitação de mensagem feita ao SQS. As solicitações podem incluir envio, recebimento e exclusão de mensagens.
- **Transferência de dados:** A transferência de dados para o SQS a partir da internet é gratuita. No entanto, a transferência de dados para fora do SQS para a internet é cobrada. A quantidade de dados transferidos determinará o custo.
- **Recursos adicionais:** Recursos como a Filtragem de Mensagens do SQS, o SQS Extended Client, entre outros, podem ter custos adicionais.

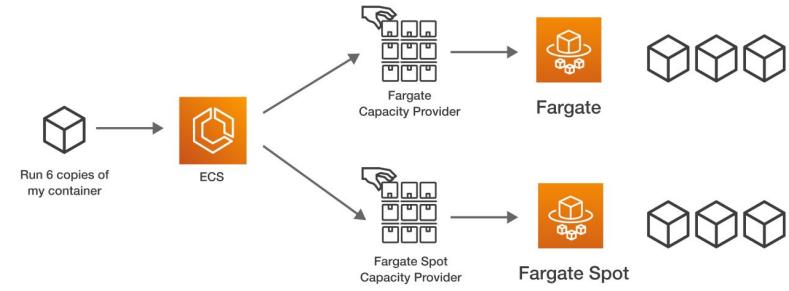


# AWS Fargate: A solução para contêineres na AWS

O AWS Fargate é um serviço que permite executar contêineres no Amazon ECS e EKS sem a necessidade de gerenciar a infraestrutura subjacente.

Ele simplifica o processo de implantação de aplicações em contêineres, permitindo que você se concentre no código e nos contêineres, e não na infraestrutura.

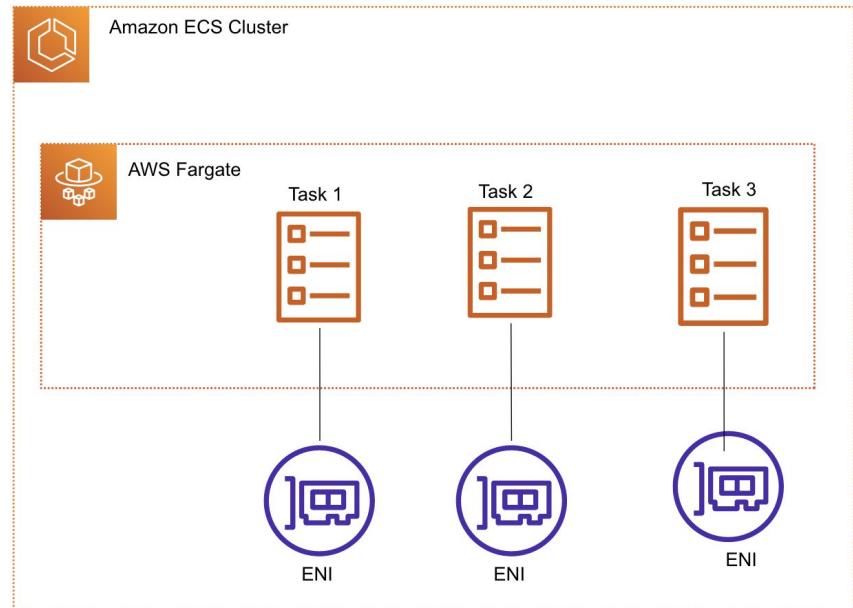
Com sua natureza sem servidor, escalonamento fácil, segurança aprimorada e integração perfeita com outros serviços AWS, o Fargate oferece uma solução poderosa para o gerenciamento de aplicações baseadas em contêineres.



 SIMFORM

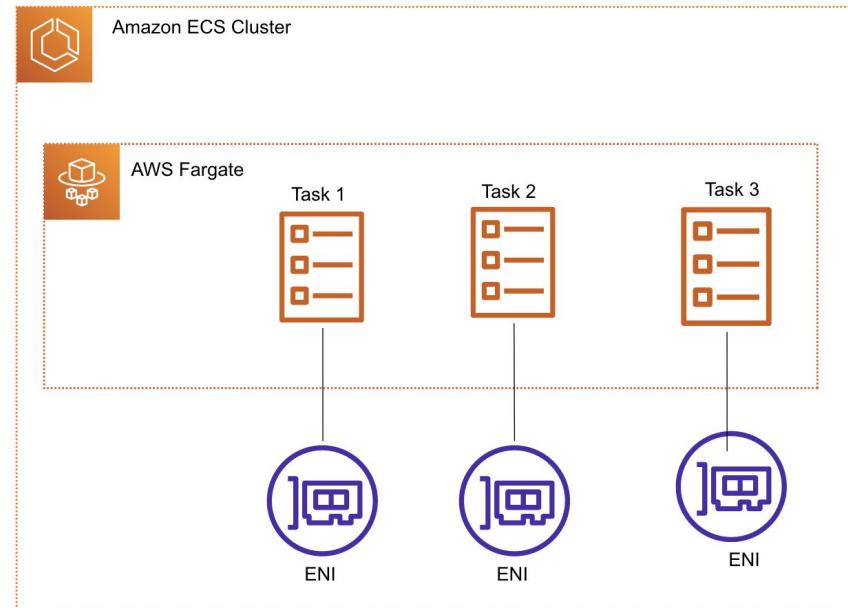
# Dominando Fargate Tasks

**As tasks no AWS Fargate representam uma ou mais instâncias de um contêiner que deve ser implantado juntas. As definições de tarefa especificam a configuração dos contêineres a serem executados, incluindo a imagem do Docker, a quantidade de CPU e memória, e outras configurações relevantes.**



# Gerenciamento Eficiente de tasks com Serviços no AWS Fargate

**Os serviços no AWS Fargate são projetados para garantir que um número definido de tarefas esteja sempre em execução. Se uma tarefa falhar, o serviço tem a capacidade de reiniciá-la automaticamente, assegurando a continuidade e a confiabilidade da execução das tarefas.**



Source image:  
<https://dev.classmethod.jp/articles/ecs-fargate-log-task-events-to-cloudwatch-logs-using-lambda/>

# Cobrança Fargate

O AWS Fargate cobra com base nas vCPUs e na memória utilizada pelos seus contêineres. Não há taxa mínima nem compromissos iniciais. Dessa forma, você paga apenas pelo que realmente usar, tornando o Fargate uma solução econômica e eficiente.



# AWS Redshift: Uma introdução ao data warehousing na nuvem

**AWS Redshift oferece uma solução de data warehousing rápida, simples e econômica. É uma plataforma poderosa para armazenamento de dados que suporta a análise de dados em grande escala.**



# Explorando a Arquitetura do AWS Redshift

**AWS Redshift é baseado em uma arquitetura de computação paralela massiva e orientada a colunas. Ele é composto por nós que podem ser organizados em clusters. Cada nó tem fatias, que são os recursos de computação que executam o trabalho real.**

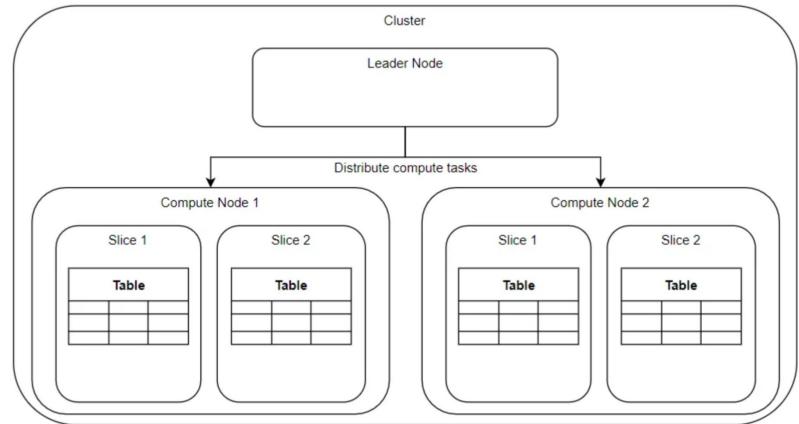


Table I/O is reduced significantly as only the necessary data are accessed



Source image:  
<https://towardsdatascience.com/amazon-redshift-architecture-b674513eb996>

# Desempenho e Otimização de Consultas no AWS Redshift

**O AWS Redshift oferece várias estratégias para otimizar o desempenho das consultas, incluindo o uso eficaz de sortkey e distkey.**

**As sortkey e distkey são fundamentais para a otimização do desempenho.**

**As sortkey organizam os dados de uma maneira que acelera as consultas, enquanto as distkey ajudam a dividir os dados entre os nós do cluster para processamento paralelo.**

**Entender quando e como utilizar essas chaves pode significar um grande aumento na eficiência das consultas.**

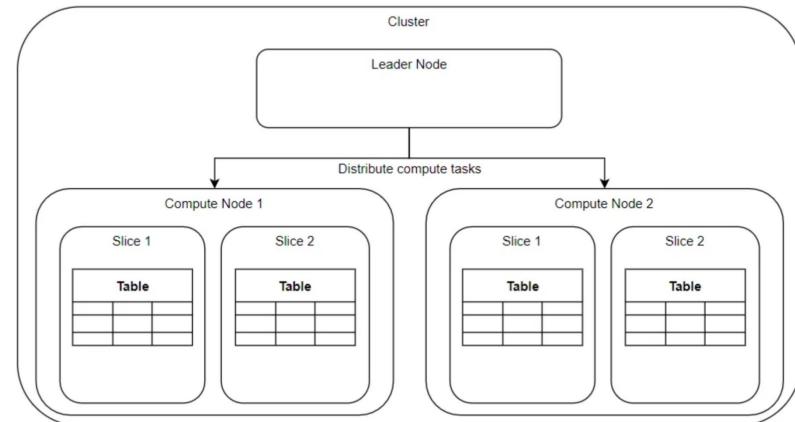


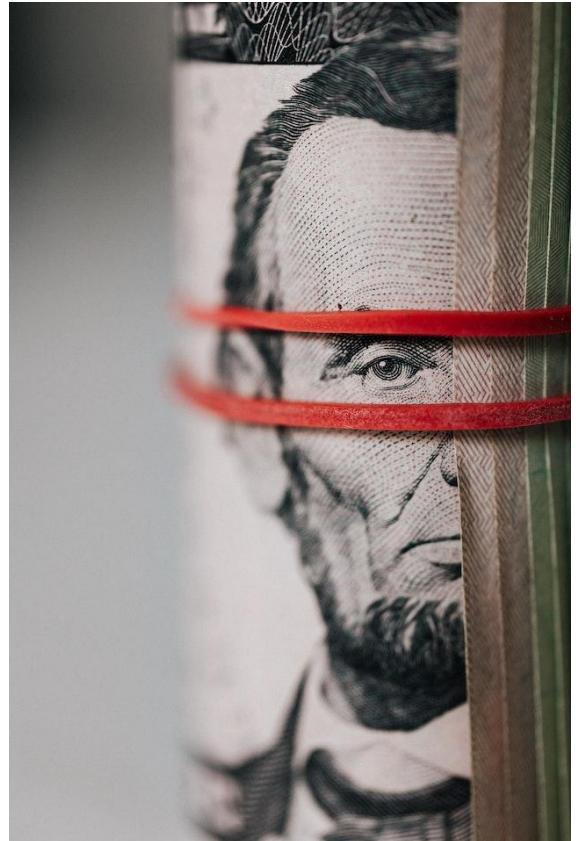
Table I/O is reduced significantly as only the necessary data are accessed.



# Cobrança Redshift

O AWS Redshift tem uma estrutura de preços que é baseada principalmente no tipo de instância que você escolhe e no tempo que você a utiliza.

- **Nodes:** O node é a unidade fundamental de processamento do Redshift. Cada node tem um determinado número de CPUs, quantidade de memória, armazenamento e um custo associado.
- **Custo por hora:** O Redshift é cobrado por hora, e o preço por hora depende do tipo de nó que você escolher. Você pode optar por pagar conforme o uso (on-demand) ou reservar uma capacidade por um ou três anos para economizar custos (reserved instances).
- **Armazenamento de backup:** O Redshift cobra pelo armazenamento de backup de acordo com a quantidade de dados que você tem no seu cluster. Você recebe um valor de armazenamento de backup igual ao tamanho do seu cluster gratuitamente.



# Mão na massa

**Agora chegou o momento de começar a criar os serviços da nossa primeira arquitetura.**

**Vamos criar do zero usando terraform.**



# Mão na massa

**Agora chegou o momento de começar a criar o código para ser usado na nossa arquitetura.**

**Vamos criar do zero usando Python e suas bibliotecas.**



# Um Olhar Retrospectivo: Terraform, Docker, Git

Terraform nos permite descrever e provisionar infraestrutura de maneira programática e abre a possibilidade de facilmente trabalhar com múltiplas clouds.

Docker nos ajuda a empacotar aplicações em contêineres, proporcionando consistência e portabilidade.

Git é o sistema de controle de versão que nos permite acompanhar e controlar as alterações nos nossos arquivos e códigos



# Revisão do Gerenciamento de Identidade e Acesso (IAM) da AWS

**IAM Usuários** são associados a pessoas ou aplicações, possuindo credenciais de segurança individuais.

**Grupos IAM** facilitam o gerenciamento de múltiplos usuários com permissões semelhantes.

**Roles IAM** permitem a interação segura entre serviços AWS e recursos.

**Policies IAM** delimitam as permissões dos Usuários, Grupos e Funções.

A segurança no IAM é otimizada através da autenticação multifatorial, rotação regular de credenciais e adoção do princípio de mínimo privilégio.

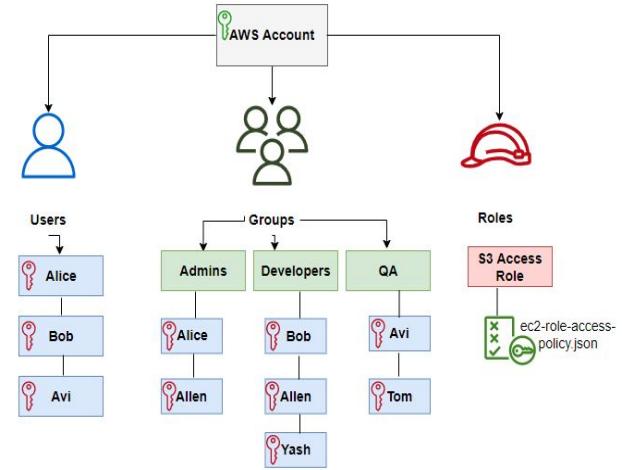
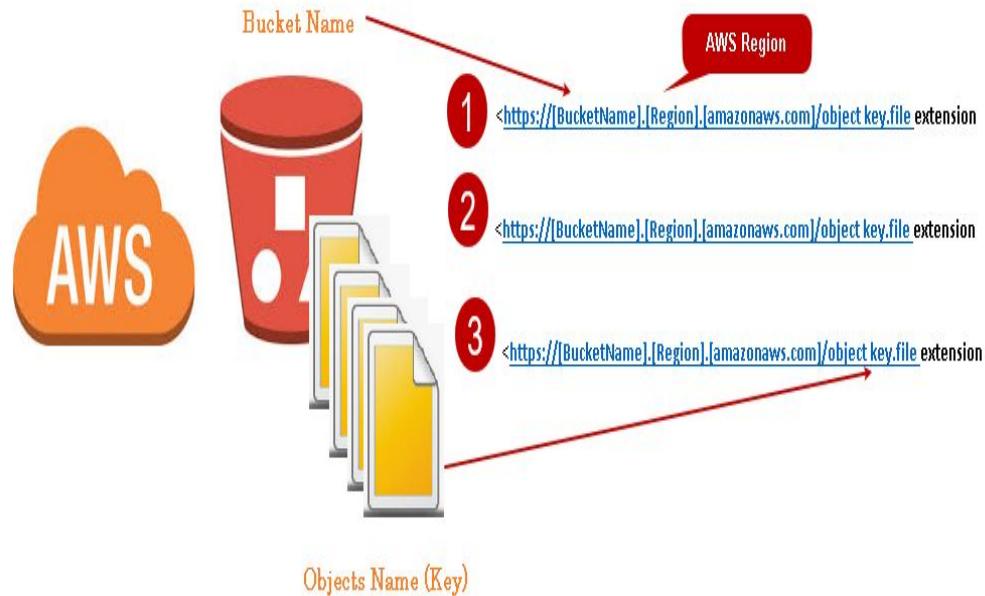


Image Source:  
<https://cloudiofy.com/aws-iam-permissions-and-policies/>

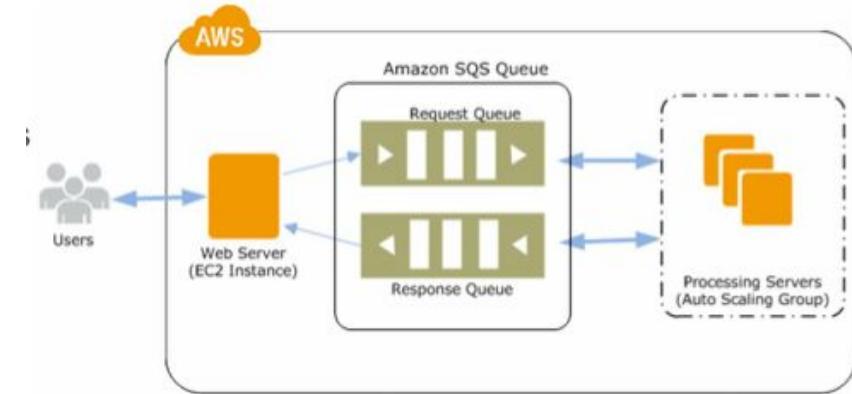
# Revisão Rápida: AWS Simple Storage Service (S3)

O AWS S3 é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade e segurança. Os Buckets no S3 são usados para armazenar e organizar objetos. As Classes de Armazenamento do S3 permitem otimizar custos e desempenho com base na frequência de acesso aos dados. O modelo de preços do S3 varia conforme a classe de armazenamento, a quantidade de dados armazenados e o nível de acesso aos dados.



# Revisão Rápida: AWS Simple Queue Service (SQS)

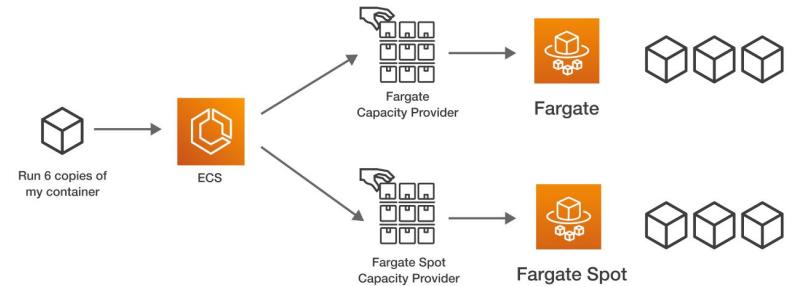
O AWS SQS é um serviço de fila de mensagens que permite o desacoplamento dos componentes de um sistema na nuvem. Ele oferece filas padrão e FIFO para lidar com diferentes necessidades de throughput. As mensagens no SQS passam por um ciclo de vida de envio, recebimento e exclusão, com um tempo de visibilidade configurável. O modelo de preços do SQS é baseado na quantidade de solicitações e no uso de recursos adicionais.



Source image:  
<https://aws.amazon.com/blogs/developer/using-python-and-amazon-sqs-fifo-queues-to-preserve-message-sequencing/>

# Revisão Rápida: AWS Fargate

O AWS Fargate é um serviço de computação sem servidor para containers. As tarefas do Fargate são a unidade de execução e podem conter um ou mais containers. Serviços no Fargate garantem que o número especificado de tarefas esteja sempre em execução e substituem tarefas se elas falharem. O modelo de preços do Fargate é baseado nos recursos provisionados e no tempo de execução.



 SIMFORM



Source image:  
<https://www.simform.com/blog/aws-fargate-pricing/>

# Revisão Rápida: AWS Redshift

O AWS Redshift é um serviço de armazenamento de dados em nuvem, escalável e totalmente gerenciado. Sua arquitetura é baseada em clusters, que são compostos por um ou mais nós. As bases de dados no Redshift podem ser criadas e gerenciadas facilmente, suportando consultas SQL padrão.



**amazon  
REDSHIFT**

# Um Olhar Retrospectivo: Terraform, Docker, Git

Terraform nos permite descrever e provisionar infraestrutura de maneira programática e abre a possibilidade de facilmente trabalhar com múltiplas clouds.

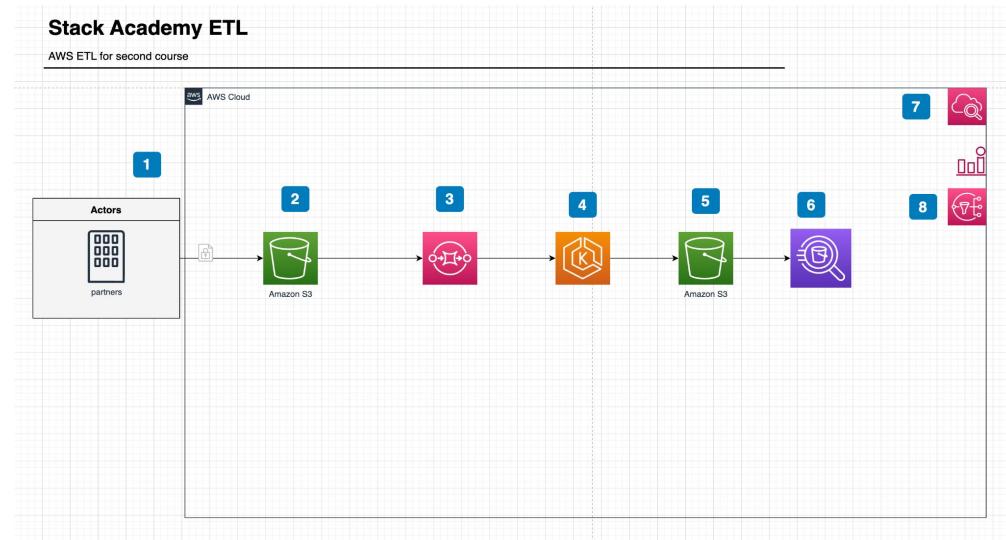
Docker nos ajuda a empacotar aplicações em contêineres, proporcionando consistência e portabilidade.

Git é o sistema de controle de versão que nos permite acompanhar e controlar as alterações nos nossos arquivos e códigos



# S3, SQS, EKS Fargate, e Athena: Nossa Segunda Arquitetura

A segunda arquitetura ilustra como os serviços AWS podem trabalhar em conjunto para lidar com dados. Os dados são carregados no S3 e as mensagens de eventos são enviadas para a SQS. O EKS, usando Fargate, responde a essas mensagens e processa os dados. A Athena é então usada para análise. O CloudWatch fornece monitoramento e logs durante todo o processo.



# Entendendo o Amazon EKS: O que é e por que é importante

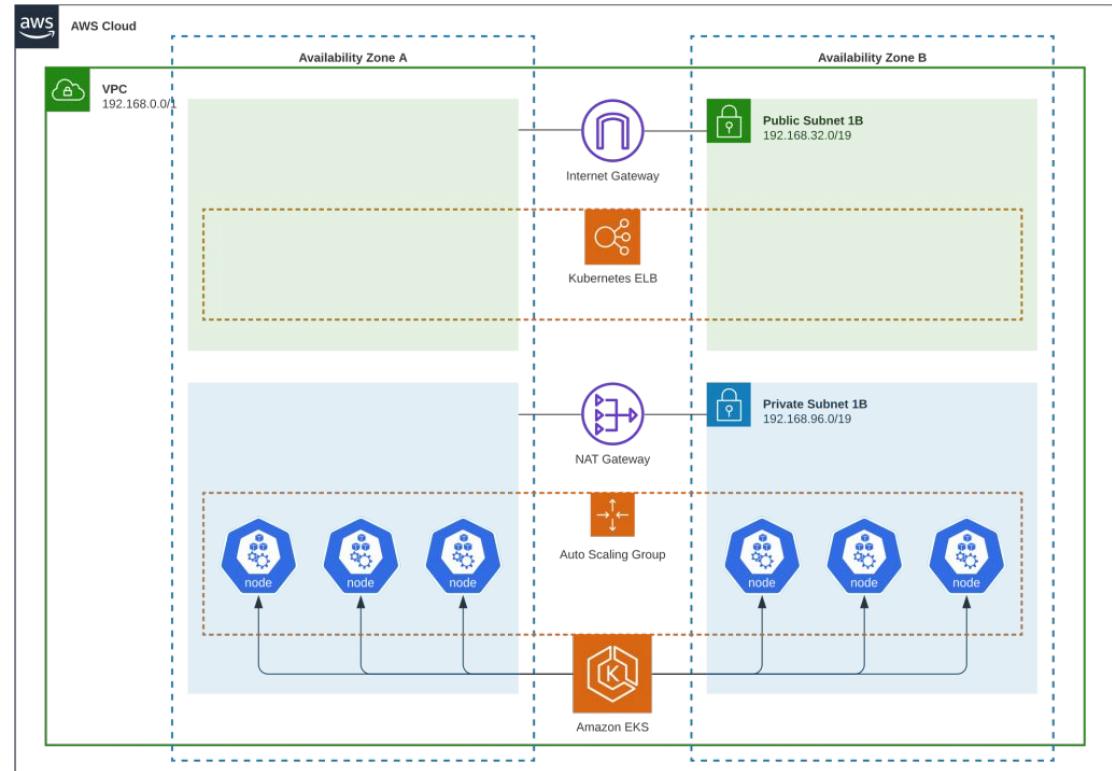
O Amazon Elastic Kubernetes Service (EKS) é um serviço gerenciado que facilita a execução do Kubernetes na AWS sem a necessidade de instalar, operar e manter a própria infraestrutura do Kubernetes. EKS oferece alta disponibilidade, segurança e escalabilidade, permitindo que as equipes de desenvolvimento se concentrem em desenvolver aplicativos, não em gerenciar infraestrutura.



## Amazon EKS

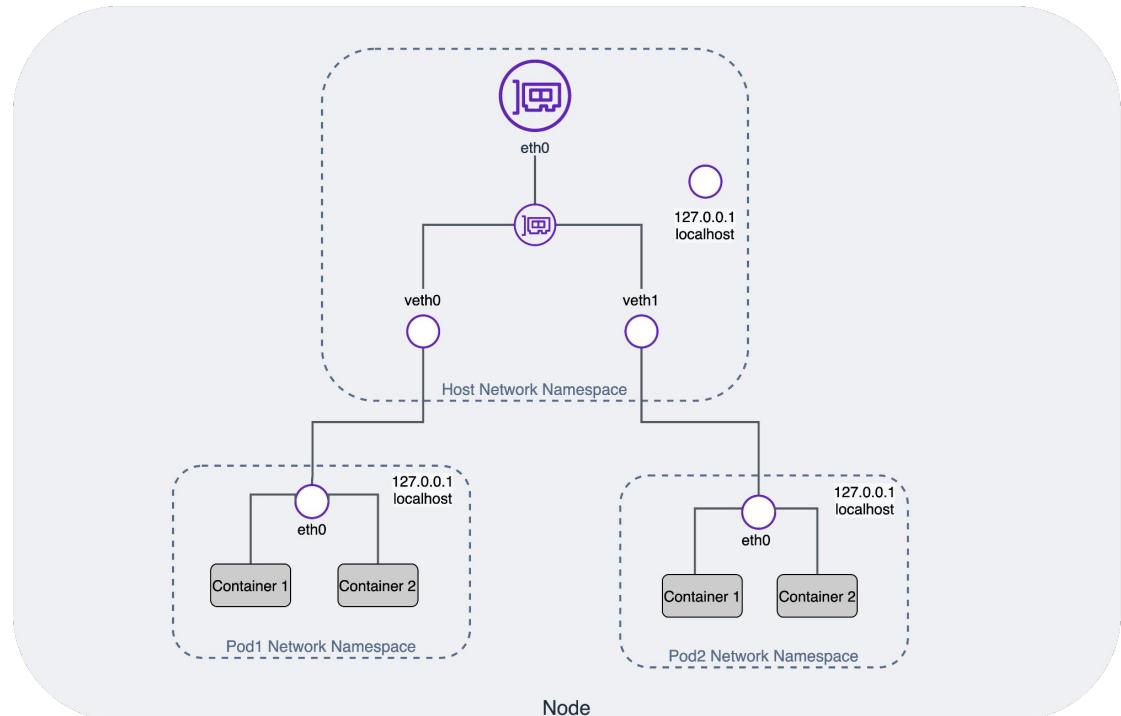
# Arquitetura do Amazon EKS: Um mergulho profundo

A arquitetura do Amazon EKS é dividida em dois principais componentes: o plano de controle e o plano de dados. O plano de controle gerencia o estado desejado do cluster, enquanto o plano de dados executa as tarefas e hospeda os pods. Nesta apresentação, vamos explorar cada componente e entender como eles trabalham juntos para gerenciar aplicações em contêineres



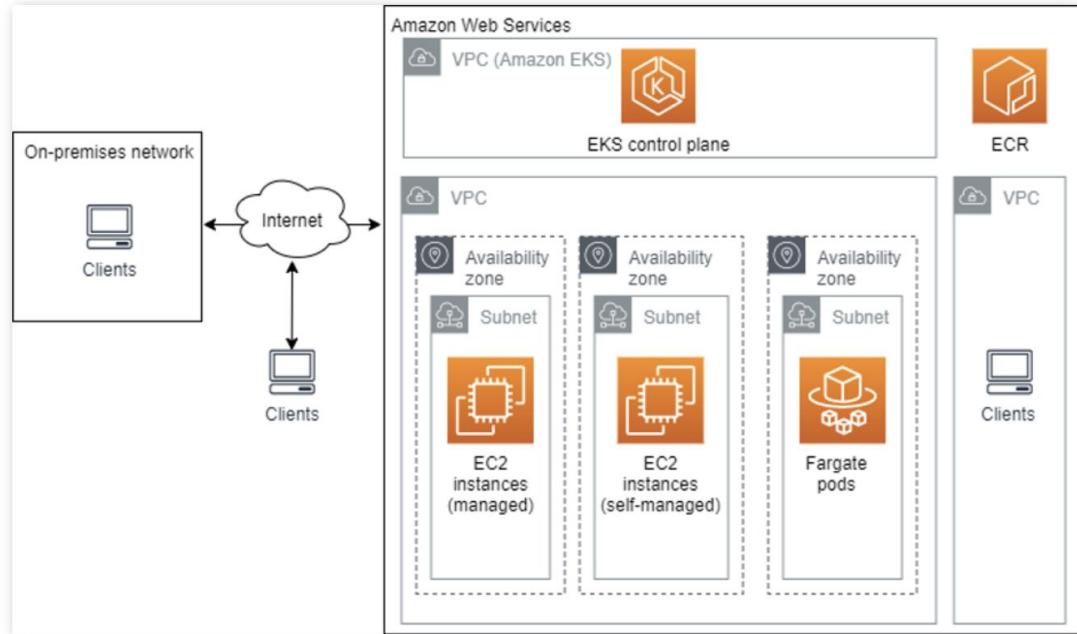
# Estrutura de Rede no Amazon EKS: VPCs, Sub-redes e Grupos de Segurança Explicados

A rede é uma parte integrante do Amazon EKS, envolvendo o uso de VPCs, sub-redes e grupos de segurança. Esta apresentação oferece um vislumbre aprofundado de como esses componentes interagem para garantir um ambiente seguro e altamente disponível para executar suas cargas de trabalho no EKS.



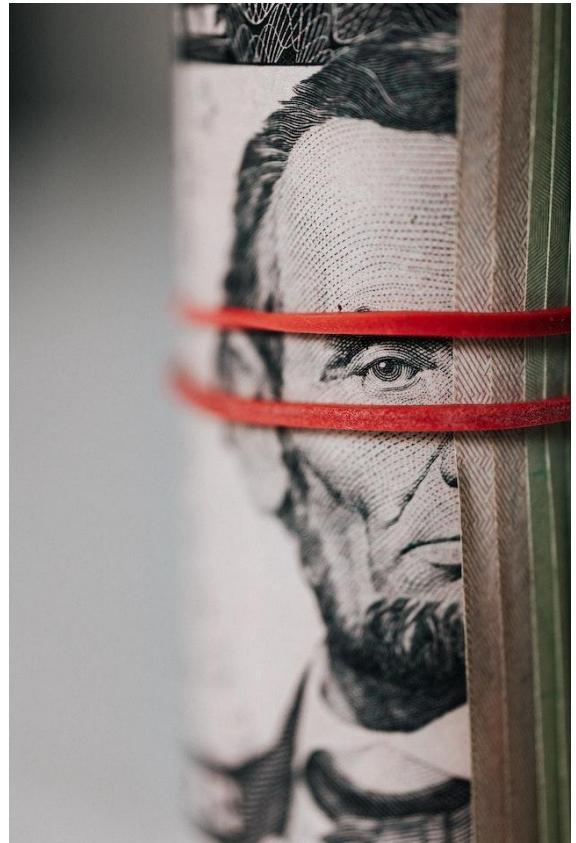
# Amazon EKS e Fargate: Uma Combinação de Sucesso

A integração do Amazon EKS com o Fargate proporciona uma experiência poderosa e sem complicações para a orquestração de containers. Vamos explorar os benefícios dessa combinação e compartilhar casos de uso que demonstram o poder do EKS com Fargate.



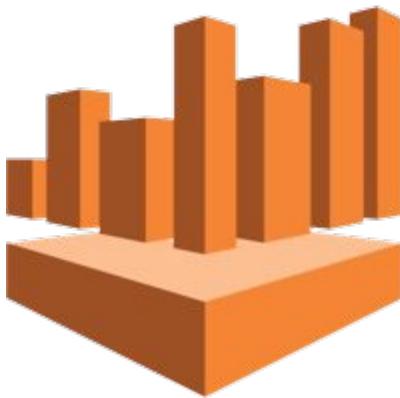
# Amazon EKS Pricing

A precificação do Amazon EKS é baseada em duas principais componentes: taxas de controle mestre e taxas de instância de nó. As taxas de controle mestre são cobradas por hora para gerenciar o cluster, enquanto as taxas de instância de nó dependem do tipo de instância EC2 escolhida. Além disso, há também custos adicionais relacionados a serviços complementares, como armazenamento EBS e tráfego de dados. É importante consultar a documentação da AWS para obter detalhes específicos sobre os preços e opções de dimensionamento.



# Amazon Athena: Uma Visão Geral do Serviço de Consulta Interativa

O Amazon Athena é uma ferramenta poderosa que permite analisar dados no Amazon S3 usando consultas SQL familiares. Nesta sessão, vamos abordar o que é o Athena, como ele funciona e como você pode utilizá-lo para obter insights valiosos dos seus dados armazenados no S3.



amazon  
ATHENA

# Explorando os Diferenciais do Athena

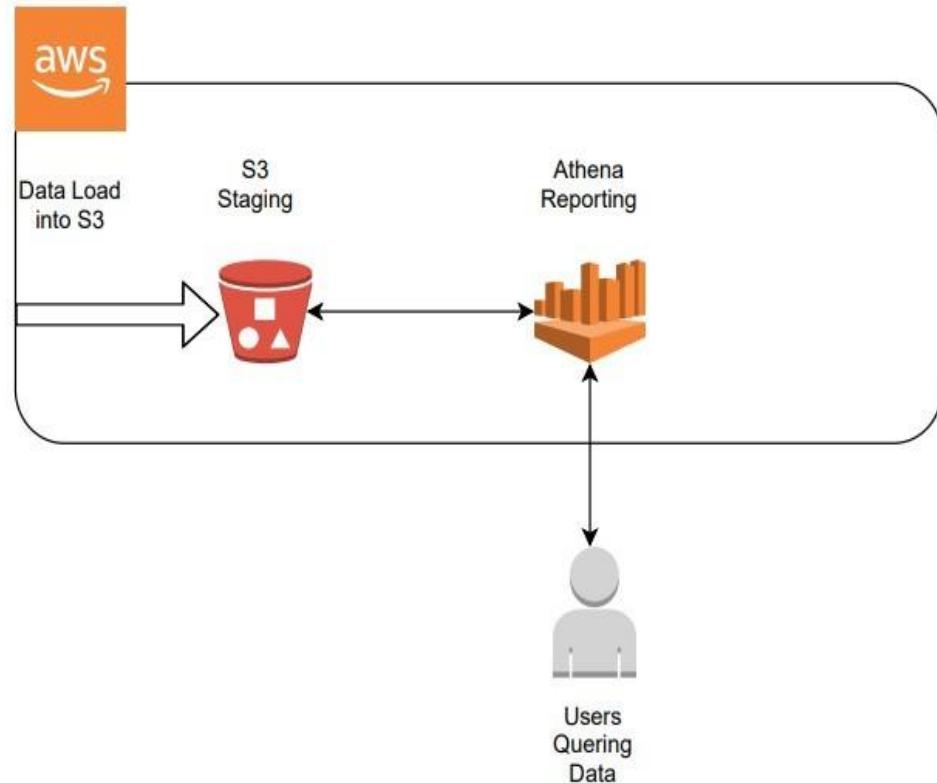
O Amazon Athena oferece uma série de recursos-chave que o tornam uma poderosa ferramenta de análise de dados. Nesta apresentação, vamos explorar os principais recursos do Athena, incluindo sua arquitetura sem servidor, escalabilidade sob demanda e suporte a consultas SQL padrão.



amazon  
ATHENA

# Amazon Athena: Uma Visão Geral do Serviço de Consulta Interativa

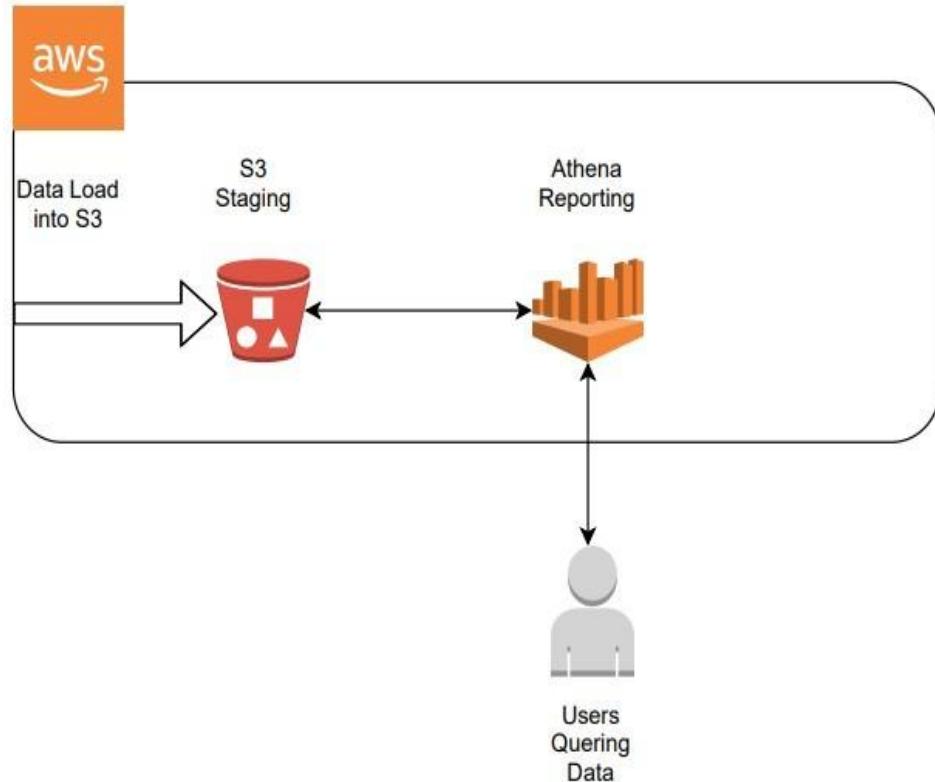
O Amazon Athena oferece uma ampla gama de recursos-chave para aprimorar a análise de dados. Sua arquitetura sem servidor permite que você se concentre na análise, em vez de gerenciar a infraestrutura. Com a escalabilidade sob demanda, o Athena se ajusta automaticamente às suas necessidades de processamento, garantindo tempos de resposta rápidos, mesmo para grandes volumes de dados. Além disso, o suporte nativo a consultas SQL permite que você use suas habilidades SQL existentes para explorar e analisar os dados no Amazon S3 de maneira eficiente.



# Integração de Dados sem Complicações: Amazon S3 e Amazon Athena

O Amazon S3 oferece um armazenamento confiável e escalável para uma variedade de dados. Com o Amazon Athena, você pode acessar e analisar esses dados de forma fácil e rápida.

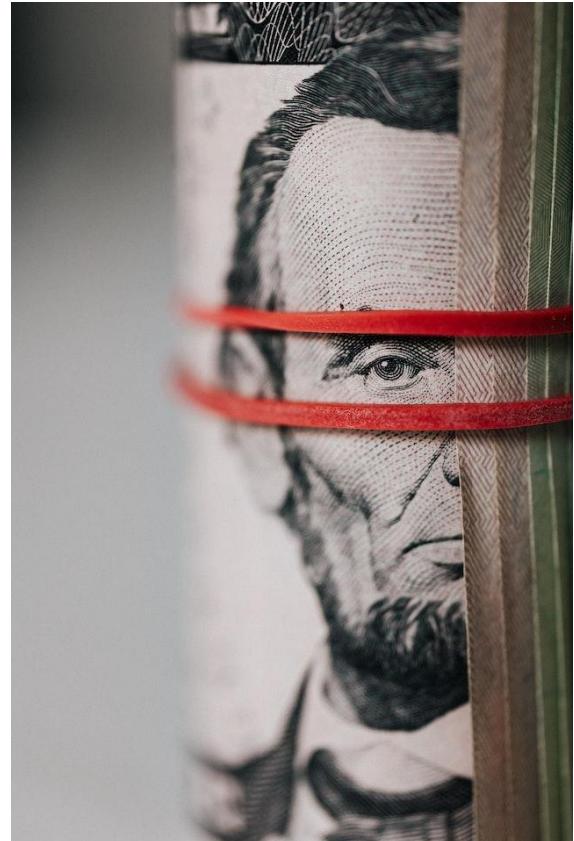
Amazon S3 e o Amazon Athena se integram perfeitamente, permitindo que você consulte diretamente os dados armazenados nos buckets do S3 usando consultas SQL padrão.



# Amazon Athena Pricing

**A quantidade de dados escaneados se refere à quantidade de dados no Amazon S3 que são acessados durante a execução das consultas. O Athena cobra uma taxa por gigabyte (GB) de dados escaneados. É importante destacar que o Athena lê somente os dados necessários para responder às consultas, o que significa que você paga somente pelos dados realmente acessados durante a execução das consultas.**

**O tempo de execução das consultas é medido em unidades de tempo, denominadas unidades de consulta. Cada unidade de consulta equivale a 1 TB de dados escaneados. O Athena cobra uma taxa fixa por unidade de consulta utilizada.**



# Mão na massa

**Agora chegou o momento de começar a criar os serviços da nossa primeira arquitetura.**

**Vamos criar do zero usando terraform.**



# Mão na massa

**Agora chegou o momento de começar a criar o código para ser usado na nossa arquitetura.**

**Vamos criar do zero usando Python e suas bibliotecas.**



# OBRIGADO

