



# Curso Programação Backend com Python

Computação e banco de dados na AWS

© Cleuton Sampaio 2021

## Sumário

Cloud business	3
Isso não é "Hosting"?	3
Serviços básicos de um Cloud Provider	4
VPS - Virtual Private Servers	7
Storage (Armazenamento)	7
Serviços de bancos de dados	8
Serviços de Rede e Segurança	8
Serviços avançados de Cloud Providers	9
Serviços de consultoria de Cloud Business	9
Serviços Serverless	10
Conta AWS	12
Cuidado com cobranças desnecessárias	13
Crie sua conta na AWS para usufruir do nível gratuito	14

## Cloud business

***Atenção: O uso de serviços em nuvem, especialmente da AWS (Amazon Web Services) representa custo e você poderá receber cobranças pelo uso deles, mesmo que esteja em modalidade gratuita! Se não se sentir à vontade com isso, não faça os exercícios, apenas acompanhe.***

Você precisa entender que estamos apenas mostrando opções para criar backend em nuvem com a AWS. O objetivo não é esgotar o assunto, pois este não é um curso de AWS, mas, com os conceitos demonstrados aqui (e um pouco mais de estudo sobre AWS) você será capaz de criar backends para esta plataforma com python.

Todo negócio de TI que envolva a contratação total ou parcial de serviços em modalidade nuvem.

### Isso não é "Hosting"?

*Hosting* ou hospedagem remota é um serviço que existe há muito tempo. Consiste em alocar uma instância de computador real ou virtual em um provedor de serviços de Internet, geralmente com o intuito de hospedar um Web Site.

*Hosting* é um tipo de *Cloud Business* sim, geralmente com menos funcionalidades e flexibilidade de custos. É o que chamamos de VPC - Virtual Private Server ou servidor virtual privado.

Geralmente, *Cloud Business* envolve mais de uma modalidade de *hosting*, por exemplo:

- **Infrastructure as a service (IAAS):** É uma infraestrutura de computação instantânea, provisionada e gerenciada pela Internet. A IaaS aumenta ou reduz rapidamente de acordo com demanda, permitindo que você pague somente pelo que usar. Ela ajuda a evitar gastos e a complexidade de comprar e gerenciar os próprios servidores físicos e outras infraestruturas do datacenter. Cada recurso é oferecido como um componente de serviço separado e você só pode alugar um específico pelo tempo que precisar. Exemplos: Subir um servidor, um firewall, um banco de dados e um balanceador de carga;
- **Platform as a service (PAAS):** Inclui tudo o que é necessário para você hospedar uma aplicação, sem ter que gerenciar a infraestrutura básica. Inclui servidores, firewalls, bancos de dados, e servidores de aplicação para você. Você tem acesso

aos servidores, mas não precisa se preocupar com tudo. Por exemplo, você contrata uma plataforma para servir seu website;

- **Function as a service (FAAS):** É um avanço do PAAS, que oferece aos clientes um ambiente Serverless (sem servidores), altamente elástico, para que implantem seu código-fonte sem se preocuparem com a infraestrutura básica. No PAAS ainda existem os servidores, no FAAS, você sequer sabe onde seu código é executado;

## Serviços básicos de um Cloud Provider

Contratamos *Cloud Services* de um *Cloud Provider*, através da Internet. Há vários *Cloud Providers*, entre eles:

- Amazon Web Services;
- Google Cloud;
- Microsoft Azure;
- Oracle Cloud;
- Huawei Cloud;
- Alibaba Cloud;

E vários outros. Todos oferecem serviços bem semelhantes, sendo diferenciados por modalidade de cobrança e regiões de disponibilidade.

Uma **Região** é um local geográfico onde o *Cloud Provider* possui Datacenters instalados. Para aumentar a disponibilidade e evitar desastres, os *Cloud Providers* possuem várias regiões, espalhadas por vários países e continentes.

Geralmente, você pode lançar serviços em qualquer região oferecida por um Cloud Provider (há poucas exceções). O que diferencia lançar serviços em cada região são:

- Disponibilidade: Alguns serviços podem não estarem disponíveis em algumas regiões;
- Custo: Pode variar de região para região pelo mesmo serviço;
- Latência: Lançar um serviço em uma região geograficamente distante dos seus clientes, pode aumentar a latência de rede;
- Governança dos dados: Pode haver regras estabelecendo que os dados devem residir na região de origem (no país de origem);

Cada região de um Cloud Provider tem dois ou mais datacenters separados, chamados de "**Zonas de disponibilidade**". Eles são separados o suficiente para que um problema, como: Falta de energia ou tempestades, não afete mais de uma Zona simultaneamente.

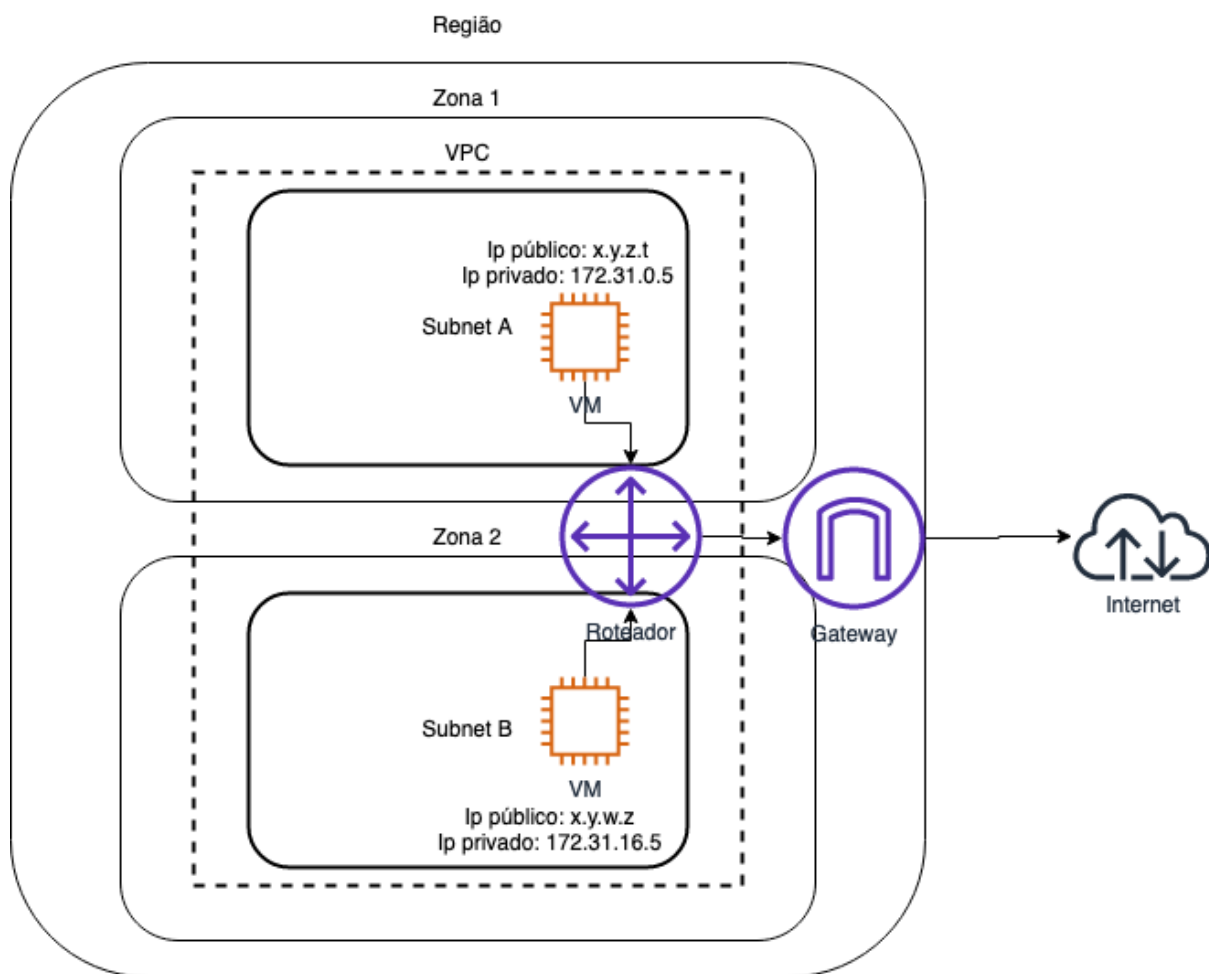


Você pode lançar serviços em uma ou mais regiões e, dentro de cada região, em uma ou mais zonas de disponibilidade. Lançar serviços em mais de uma zona de disponibilidade aumenta a disponibilidade (High Availability) da sua aplicação, pois se uma zona cair, a outra pode continuar a manter a aplicação no ar. É claro que lançar serviços em mais de uma Zona aumenta os custos e exige algum tipo de balanceamento de carga, para distribuir os acessos entre as várias instâncias do seu aplicativo.

Todo *Cloud Provider* tem o conceito de **VPC**: *Virtual Private Cloud* ou nuvem privada virtual. Uma VPC é um trecho da rede do Cloud Provider totalmente dedicada à sua conta. Ou seja, só você pode acessar os serviços lançados dentro da sua VPC.

Uma VPC é dividida em redes virtuais, chamadas de *Subnets*, cada uma com sua faixa de endereços IP privados únicos.

Uma VPC abrange toda as **Zonas de disponibilidade** de uma **Região** e você pode criar *Subnets* em cada uma dessas Zonas.



Na figura temos uma VPC, criada em uma determinada região. Ela abrange as duas Zonas de disponibilidade (A e B), com Subnets em cada uma dessas Zonas. Ela possui uma instância de servidor (máquina virtual - EC2 instance) em cada Subnet. Note que existem endereços IP públicos (disponíveis na Internet) e privados (disponíveis apenas em cada Subnet).

Você pode separar sua aplicação em várias Subnets, por exemplo: Web servers, Databases ou "Produção" e "Desenvolvimento". É possível criar firewalls em cada Servidor ou em cada rede.

Todo Cloud Provider tem algum tipo de mecanismo de autenticação e autorização, que permite controlar o acesso aos serviços que você lança em suas VPCs. Com este mecanismo, você tem uma conta raiz (Root User), que geralmente é associada a quem paga pelos serviços, e pode ter uma ou mais contas associadas (que usam o serviço).

Você também pode ter credenciais de acesso temporárias, que podem ser associadas aos seus servidores, para que possam consumir serviços da Nuvem.

## VPS - Virtual Private Servers

Todo *Cloud Provider* tem o serviço de VPS, que é lançar instâncias de máquinas virtuais na sua VPC. Este serviço pode ter nomes diferentes em cada *Cloud Provider*. Na AWS - Amazon Web Services, é chamado de **EC2 - Elastic Compute Cloud**.

Um VPS é geralmente cobrado de acordo com os recursos que utiliza. A maioria dos Cloud Providers cobra por: Quantidade de CPUs virtuais (vCPUs), quantidade de memória RAM e licença de Sistema Operacional (ou software pré instalado). O espaço em disco é cobrado à parte.

Há várias modalidades de pagamento por VPS:

- **Sob demanda (On Demand):** Você sobe o Servidor e será cobrado por hora em que ele estiver no ar. Alguns provedores quebram em segundos;
- **Reservada:** Você contrata o VPS por um período (1 ano, 3 anos etc) e paga mensalmente por 730 horas de uso (1 mês comercial), obtendo descontos significativos no custo por hora;
- **Spot:** Uso de capacidade ociosa do *Cloud Provider*. Você faz uma oferta (bid) de quanto quer pagar e se sua oferta for boa o suficiente, o Provedor aloca a máquina para você por um preço muito mais baixo. Só que, a qualquer momento, ele pode reclamar a máquina de volta.

Cada VPS tem dois tipos de Storage (armazenamento): Transiente e Permanente. Discos transientes servem para "dar boot" na instância e contém a imagem do Sistema Operacional. Quando o Servidor (a instância) é terminada, o disco e seu conteúdo são perdidos. Mas você pode alocar Storage permanente, pagando por espaço reservado (GB por mês). A Storage permanente é um serviço à parte do VPS e pode continuar mesmo que você termine o Servidor. Só que é cobrado à parte.

## Storage (Armazenamento)

Os *Cloud Providers* oferecem serviços de armazenamento para você guardar seus dados e eles variam de acordo com a finalidade e modalidade. Exemplos:

- **File Storage:** São sistemas de arquivos que podem ser montados em mais de uma instância de VPS simultaneamente, servindo como armazenamento compartilhado. São o equivalente aos "discos de rede" de uma rede corporativa;

- **Block Storage:** São volumes virtuais de disco, que podem ser atachados (instalados) em uma instância de VPS de cada vez, servindo como Storage permanente. Você pode desinstalar de uma instância e instalar em outra;
- **Object Storage:** São web storages, como o Google Drive. Você usa uma API via Web para armazenar ou recuperar objetos dela. Servem para armazenar grandes volumes de documentos, de vários tipos;
- **Cold Storage:** Armazenamento de dados infrequentemente acessados, como: Backups e "arquivo morto". Geralmente há uma espera para obter os dados e um custo de recuperação por GB.

## Serviços de bancos de dados

É claro que você pode subir uma instância de VPS e instalar um banco de dados nela. Nada impede isso. É a modalidade IAAS (Infrastructure as a service), na qual você tem que operar, suportar e gerenciar seus servidores diretamente.

Mas os *Cloud Providers* oferecem um tipo de PAAS para Banco de dados, chamado de "Bancos de dados gerenciados", no qual o *Cloud Provider* administra os servidores e você os dados.

Neste caso é possível ter benefícios como: Backup automático, redundância (armazenamento em múltiplas Zonas de disponibilidade), aplicação automática de "patches" etc.

Você pode escolher o tamanho da instância do Servidor (vCPUs, memória) e o espaço de storage para o Banco de dados e para o seu backup. É claro que pagará mais caro quanto mais selecionar e também pagará pelo tipo de SGBD que deseja usar (se forem pagos, como Oracle ou MS SQL Server).

## Serviços de Rede e Segurança

Subir um VPS é apenas parte do serviço. Você precisará alocar um IP fixo para ele, de modo que possa registrar um nome de domínio. E ambos são pagos por você. Quando você sobe uma instância de VPS em uma Subnet pública, recebe um endereço IP público transiente. Quando sua instância terminar, o endereço será perdido. Quando você quer associar um nome de domínio ao seu servidor, precisará de um endereço IP fixo e pagará por isso. Também pagará pelo nome de domínio.



Outro serviço muito importante é o Balanceamento de carga. Se você subir mais de uma instância da sua aplicação, precisará distribuir a carga de requests entre eles. Geralmente os Cloud Providers oferecem algum tipo de balanceador que executa este serviço.

E há os serviços de segurança, como: Firewalls de aplicação, detectores de ataques DDOS (distributed denial of service) que são cobrados à parte. Os firewalls simples, oferecidos de graça, não protegem seu site de ataques de aplicação, como: SQL injection, por exemplo.

## Serviços avançados de Cloud Providers

Os mais modernos provedores oferecem soluções avançadas, baseadas em vários conceitos:

- **IaC:** Infrastructure as Code uma maneira de provisionar sua infraestrutura virtual baseada em codificação;
- **Contêineres:** Uso de contêineres (LXC) em vez de máquinas virtuais, oferecendo menor custo por hora;
- **Servidores dedicados:** Uso de servidores particulares para as suas aplicações;
- **Hospedagem Serverless:** Oferecimento de serviços de hosting, storage e bancos de dados sem servidores (FAAS);

A tendência atual é usar serviços Serverless, que podem ser apropriados dependendo do cenário do Cliente. Por exemplo, se você hospeda uma aplicação que roda 24 x 7, então um VPS (ou Contêiner) pode ser uma boa solução, pois você pagará pelo tempo em que o seu servidor estiver no ar. Porém, e se o fluxo de transações que sua aplicação recebe é irregular? Se usar uma solução Serverless, pagará pela quantidade de transações processadas e não mais por hora de Servidor.

## Serviços de consultoria de Cloud Business

Os serviços oferecidos pelos Cloud Providers são complexos, cuja cobrança exige estudo e prática para otimizar. Uma história recente de uma startup conta que eles subiram uma aplicação distribuída, que usava uma fila de mensagens do Cloud Provider, e em apenas 1 dia gastaram quase 10.000 Reais por uma falha de configuração.

Hoje em dia, classificamos as aplicações e recursos de TI em:

- **On premises:** Hospedadas em no datacenter do Cliente;
- **Cloud based:** Hospedadas em um (ou mais) *Cloud Provider*;
- **Híbrid cloud:** Hospedados no datacenter do Cliente com parte Cloud based;

Serviços de migração de *on premises* para *cloud based* ou mesmo *híbrid cloud* são muito populares, pois ajudam as empresas a transformarem investimento de capital ou **CAPEX** (Capital Expenditures) em despesas operacionais ou **OPEX** (Operational Expenditures). Diminuindo a imobilização de capital e aumentando a flexibilidade e agilidade da Empresa para novos negócios.

Imagine o cenário de uma empresa que deseja lançar um novo produto ou serviço e que precise aumentar sua infraestrutura de TI para acomodar a nova aplicação. Talvez ela tenha que: aumentar a carga elétrica do edifício, investir em "no breaks", talvez até alugar mais espaço. Certamente terá que contratar mais pessoas, investir em hardware e licenças de software. Com um cenário *cloud based* ou *híbrid cloud*, ela só pagará por aquilo que utilizar, sem ter que imobilizar capital.

E tudo isso pode ser feito utilizando-se múltiplos Cloud Providers, aproveitando aqueles que oferecem o menor custo e maior benefício para cada necessidade. É o que se chama de "**Multicloud**".

## Serviços Serverless

Dentre as modalidades de serviço oferecidas pelos Cloud Providers, os denominados "serverless" oferecem grande flexibilidade e baixo investimento inicial, sendo voltados para quem está começando, como startups, ou para quem deseja usufruir dos serviços de nuvem sob demanda, sem necessidade de provisionar infraestrutura (IaaS).

Eles são chamados de "serverless" porque prescindem de provisionamento de instâncias de servidores, como Instâncias EC2 ou de Banco de Dados. Você usa e paga pelo uso apenas, ficando isento de cobranças por tempo.

A AWS (Amazon Web Services) possui alguns serviços serverless bastante práticos e que podem ser utilizados para criação de backends, como microsserviços em python. São eles:

- **DynamoDB:** Um banco de dados não relacional com desempenho fantástico e baixo custo de armazenamento e acesso;

- **AWS Lambda:** Um serviço de computação serverless sob demanda, que executa suas funções na modalidade Function as a Service, com baixo custo de utilização e boa performance;
- **API Gateway:** Um serviço de publicação e monitoramento de APIs de alta performance e baixo custo;

E é claro que estes serviços podem ser utilizados com a linguagem python, se tornando excelentes escolhas para criação de aplicações backend. Portanto, veremos uma aplicação exemplo utilizando estas tecnologias aqui.

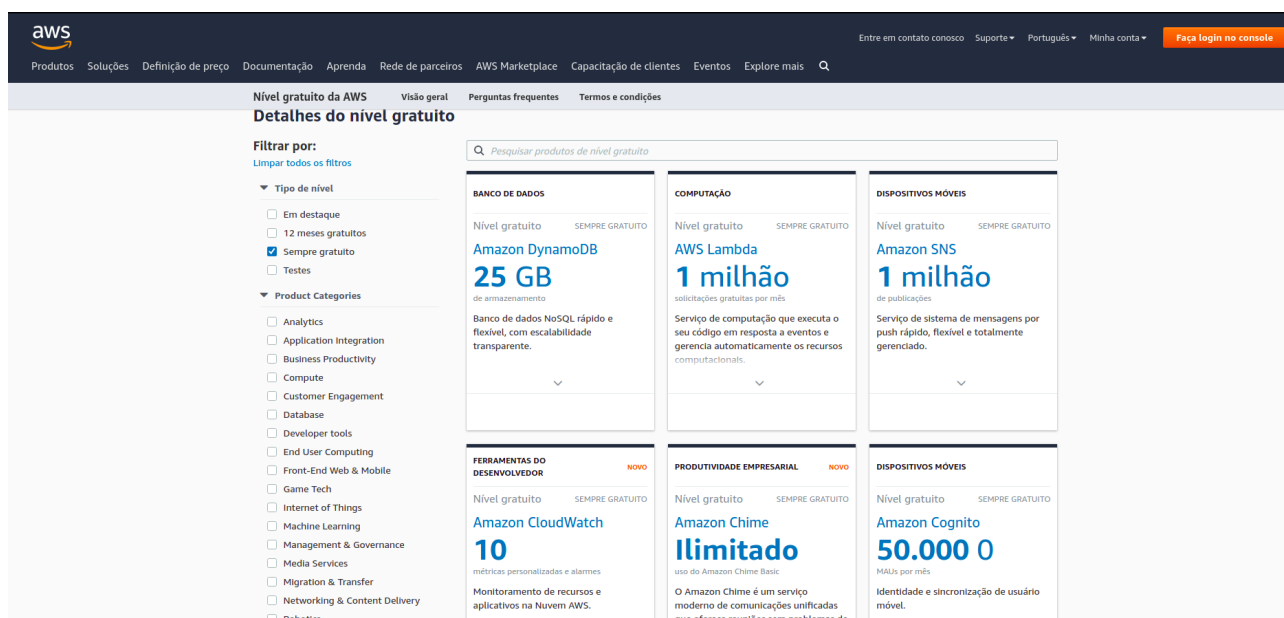
***Atenção: O uso de serviços em nuvem, especialmente da AWS (Amazon Web Services) representa custo e você poderá receber cobranças pelo uso deles, mesmo que esteja em modalidade gratuita! Se não se sentir à vontade com isso, não faça os exercícios, apenas acompanhe.***

Por favor, leia o parágrafo anterior, em vermelho mais umas duas vezes, ok? A responsabilidade pelos custos gerados ao tentar fazer os exercícios é sua e somente sua e acredite: Negligenciar isso pode gerar altíssimas cobranças em seu cartão de crédito!

## Conta AWS

**Atenção: O uso de serviços em nuvem, especialmente da AWS (Amazon Web Services) representa custo e você poderá receber cobranças pelo uso deles, mesmo que esteja em modalidade gratuita! Se não se sentir à vontade com isso, não faça os exercícios, apenas acompanhe.**

Vou demonstrar uma pequena solução de backend em nuvem utilizando os serviços da AWS (Amazon Web Services). Você não precisa fazer estes exercícios. Se quiser fazer, terá que criar uma conta na AWS. Há um nível de uso gratuito, que pode ser consultado nesta página [AQUI](#). Sugiro que leia com cuidado e atenção.



The screenshot displays the AWS Free Tier page. On the left, there's a sidebar with filters for 'Tipo de nível' (Em destaque, 12 meses gratuitos, Sempre gratuito, Testes) and 'Product Categories' (Analytics, Application Integration, Business Productivity, Compute, Customer Engagement, Database, Developer tools, End User Computing, Front-End Web & Mobile, Game Tech, Internet of Things, Machine Learning, Management & Governance, Media Services, Migration & Transfer, Networking & Content Delivery, Robotics). The main content area is titled 'Nível gratuito da AWS' and 'Detalhes do nível gratuito'. It features a search bar and a grid of service cards. The cards are organized into categories: BANCO DE DADOS (Amazon DynamoDB, 25 GB), COMPUTAÇÃO (AWS Lambda, 1 milhão), DISPOSITIVOS MÓVEIS (Amazon SNS, 1 milhão), FERRAMENTAS DO DESENVOLVEDOR (Amazon CloudWatch, 10), PRODUTIVIDADE EMPRESARIAL (Amazon Chime, Ilimitado), and DISPOSITIVOS MÓVEIS (Amazon Cognito, 50.000 O).

Há serviços que, em determinados limites são gratuitos para sempre, outros, são gratuitos por 12 meses e ainda outros que não são gratuitos.

Mesmo os serviços gratuitos “para sempre” podem gerar cobranças, caso os limites sejam excedidos.

Se você quiser seguir e fazer este exercício na AWS, será por sua responsabilidade, conta e risco. A probabilidade de gerar cobranças é baixa, mas as regras podem mudar, portanto, confira sempre os limites gratuitos da AWS [NESTE SITE](#) e calcule seus gastos antes de qualquer coisa.

## Cuidado com cobranças desnecessárias

***Atenção: O uso de serviços em nuvem, especialmente da AWS (Amazon Web Services) representa custo e você poderá receber cobranças pelo uso deles, mesmo que esteja em modalidade gratuita! Se não se sentir à vontade com isso, não faça os exercícios, apenas acompanhe.***

Sim, estou repetindo o aviso.

Serviços em nuvem, seja da AWS ou de outros provedores, geram cobranças pelo uso ou pela reserva. Se você instanciar uma máquina virtual (um VPS) pagará pelo tempo em que ele ficou ativo, se você criar um disco virtual na nuvem (Storage), pagará pela quantidade de bytes, e também pagará pelo tráfego de rede de qualquer serviço.

Serviços **serverless** são cobrados sob demanda. Teoricamente, você não paga pela reserva do serviço, mas isto pode mudar. Cada serviço serverless tem uma definição de preço que eu recomendo **FORTEMENTE** que você leia e leia mais uma vez as ofertas de nível gratuito da AWS [NESTA PÁGINA](#), evitando incorrer em cobranças, que, mais uma vez, são de sua responsabilidade, conta e risco.

Os níveis gratuitos geralmente servem apenas para desenvolvimento de soluções, não prestando para a produção (operação comercial) das mesmas. Consulte sempre os limites de uso gratuito e as políticas de preço e modalidades de cobrança antes de fazer qualquer coisa.

Você pode evitar cobranças seguindo estas regras:

1. **Você está dentro do limite de uso gratuito?** Sempre verifique se o que vai fazer se enquadra nos limites de uso gratuito da AWS ou do seu provedor de nuvem, que pode ser diferente;
2. **Antes de terminar, apague tudo o que criou?** Sempre apague arquivos, bancos de dados e remova funções e termine instâncias. Jamais deixe algo ativo ou registrado depois de terminar seu trabalho ou estudo;
3. **Suas aplicações estão com acesso público?** Jamais deixe suas aplicações com acesso público via internet. Verifique sempre isso;
4. **Você tem alertas de custo e cobrança ativados?** A AWS, assim como outros provedores de serviços de nuvem, oferecem alertas e monitoração de custos e cobranças. Sempre verifique isso;

Por fim, gostaria de recomendar que leia com bastante atenção esta página da AWS sobre como evitar cobranças inesperadas:

[https://docs.aws.amazon.com/pt\\_br/awsaccountbilling/latest/aboutv2/checklistforunwantedcharges.html](https://docs.aws.amazon.com/pt_br/awsaccountbilling/latest/aboutv2/checklistforunwantedcharges.html)

## **Crie sua conta na AWS para usufruir do nível gratuito**

***Atenção: O uso de serviços em nuvem, especialmente da AWS (Amazon Web Services) representa custo e você poderá receber cobranças pelo uso deles, mesmo que esteja em modalidade gratuita! Se não se sentir à vontade com isso, não faça os exercícios, apenas acompanhe.***

**Quaisquer custos e cobranças que lhe sejam imputados pela Amazon serão de sua inteira responsabilidade. Tenha certeza do que está fazendo. Sentindo desconforto com isso, evite fazer os exercícios. Só acompanhe!**

**Mais uma vez: Fazer estes exercícios não é parte do curso, sendo um bônus. É possível aprender apenas observando e acompanhando os vídeos.**

Se decidir levar isso adiante e fazer os exercícios que vou demonstrar, é provável que você não receba cobrança alguma, embora eu não possa garantir isso, pois não depende de mim. Consulte sempre [AQUI](#) os níveis gratuitos antes de se decidir a fazer ou não os exercícios.

Eu sei.. Estou sendo excessivamente chato e repetitivo, mas você precisa entender bem que pode haver cobrança e isso será por sua própria conta!

Após criar sua conta, siga estas recomendações:

1. Adote a MFA - Autenticação [Multi-fator com o Google Authenticator](#);
2. Crie usuários IAM e evite usar sua conta raiz;
3. Crie papéis (roles) IAM e atribua aos seus recursos e usuários;
4. Jamais guarde senhas e usuários dentro do seu código-fonte;
5. Leia sempre as [recomendações de segurança](#);

Dito isto, se realmente quiser criar uma conta AWS e experimentar, acesse a Console de gerenciamento AWS e crie sua conta.

<https://aws.amazon.com/pt/console/>

Há alguns vídeos (infelizmente em inglês) que explicam como criar sua conta e receber 12 meses de serviços (alguns e com limites) gratuitos:

- O que é o nível gratuito da AWS: <https://youtu.be/rkKvzCskpLE>
- Como saber se estou excedendo o nível gratuito: <https://youtu.be/CWKK0sypkE>
- Como eu crio e ativo uma conta no AWS: [https://youtu.be/v3WLJ\\_0hnOU](https://youtu.be/v3WLJ_0hnOU)

## Backends cloud native na AWS

A AWS oferece vários serviços de computação e storage (armazenamento) para soluções backend, entre eles:

- **EC2** - Elastic Compute Cloud: Uma solução de máquinas virtuais (Infrastructure as a Service - IaaS) com toda configuração de rede por software;
- **ECS** - Elastic Container Service: Uma solução de gerenciamento de containers;
- **Elastic Beanstalk**: Uma solução de plataforma com serviço (PaaS) para hospedagem de aplicações, preferencialmente Web;
- **Lambda**: Um serviço de hospedagem de Funções como Serviço (FaaS) flexível e com cobrança sob demanda, ou seja, você só paga quando sua função é invocada;

O **EC2**, o **ECS** e o **Elastic Beanstalk** também podem ser utilizados sob demanda e você paga o tempo em que o recurso (VM ou Contêiner) estiver ativo, independentemente da quantidade de transações que tenham executado. Há um custo por tráfego de saída de rede e armazenamento (Storage). É possível reservar os recursos, o que faz o preço cair muito.

O **AWS Lambda** é diferente, pois é um serviço serverless, no qual você não aloca e nem gerencia recurso algum. O ambiente do Lambda provisiona e serve a sua função. A cobrança é por transação ou invocação da sua função. Se não for invocada, não há cobrança.

Se utilizar o AWS Lambda como solução para hospedar seu backend, provavelmente precisará de um serviço para expor sua API, preferencialmente que controle o acesso e ofereça a possibilidade de gestão de cotas por usuário. Este é o **AWS API Gateway**.

Todo backend precisa de armazenamento (Storage) e a Amazon oferece alguns serviços interessantes:

- **Simple Storage Service - S3:** Um File System de Objetos em nuvem, serverless. Você paga por Gigabyte armazenado por mês;
- **Elastic File System - EFS:** Um file system tipo NFS que pode ser montado em uma instância EC2 ou um contêiner ECS. A cobrança é por Gigabyte por mês;
- **Elastic Block Storage - EBS:** Um disco virtual que você pode montar em instância EC2 ou contêiner ECS. Você paga pelo tamanho do volume EBS que criou por mês;

Funções hospedadas no AWS Lambda podem usar S3 ou EFS como armazenamento de arquivos.

E temos os bancos de dados:

- **Relational Database Service - RDS:** Um SGBD relacional, com vários engines (PostgreSQL, MySQL, Oracle, MS SQL Server, MariaDB e Aurora), no qual você paga pelo tipo de instância de servidor por mês. Isso não inclui o armazenamento;
- **DynamoDB (NoSQL):** Um serviço serverless que oferece armazenamento simplificado e baixo custo, que pode ser cobrado sob demanda ou com capacidade provisionada (quantidade de leituras e gravações);

## Qual é a melhor solução?

Eu sempre digo que a solução depende do problema. É preciso saber seus requisitos funcionais e não funcionais, ou seja, as necessidades do seu negócio, antes de escolher uma solução.

Se já tem uma aplicação funcionando **on premises**, ou seja instalada em seu servidor e quer simplesmente mudar para nuvem, uma migração do tipo **lift-and-shift** pode ser a melhor opção inicial. Neste tipo de migração para nuvem, você simplesmente espelha os recursos que possui em seu datacenter na nuvem, e depois de um tempo, desliga sua versão on premises. Neste caso, você provavelmente utilizará EC2, RDS e EBS, para simular disco.

Agora, se for uma solução nova dependerá de alguns fatores:

- Você tem expertise em **devops**? Possui conhecimento ou equipe para desenvolver e gerenciar soluções de TI?
- Você conhece o fluxo de transações e o perfil de tráfego da sua aplicação?
- Você precisa de performance crítica ou de integrações especiais?



Se você não tem expertise em **devops** e não possui equipe que tenha esse conhecimento ou simplesmente não quer gerenciar isso, então é melhor pensar em soluções serverless, como **AWS Lambda** e **DynamoDB**. Embora o **ECS** e o **Elastic Beanstalk** possam abstrair parte da complexidade, ainda há necessidade de gerenciar os recursos.

Se o perfil de tráfego do seu sistema é conhecido e constante, apresentando poucas variações ao longo do dia e do mês, uma solução como **ECS** ou **EC2** podem ser a melhor opção, pois permitem reserva de recursos a longo prazo e contam com ferramentas de escalabilidade e alta disponibilidade, para picos de demanda. Se o tráfego é mais aleatório, não apresentando padrões, então é melhor uma solução AWS Lambda, na qual pagará por execução da função.

Finalmente, vamos tratar de performance crítica. Uma aplicação de “missão crítica” ou com elevados níveis de serviço, pode exigir a gestão da infraestrutura envolvida. Ou se você precisar gerenciar a infraestrutura por alguma razão, ou integrar diversas aplicações diferentes, o **EC2** com **RDS** pode ser a melhor solução.

## Startups e novos projetos

Neste caso, você não tem uma ideia clara de como será a demanda pelo seu serviço, portanto, não faz sentido investir muito em desenvolvimento e infraestrutura. Melhor utilizar serviços serverless como AWS Lambda e DynamoDB.

Em conjunção com o API Gateway, representam excelentes soluções para backend mobile, IoT ou de websites.

Neste curso vou mostrar exemplos com AWS Lambda e DynamoDB.

## Cloud native

Cloud native são aplicações criadas para funcionar em ambiente de nuvem. Aplicações baseadas em lift-and-shift ou EC2 podem ser aplicações tradicionais que, por acaso, estão rodando em ambiente de nuvem, possuindo todas as características de aplicações tradicionais **on premises**.

Uma solução baseada em API Gateway, Lambda e DynamoDB, serviços totalmente gerenciados e serverless, é uma solução cloud native e precisa de nova abordagem para desenvolvimento.

Em vez de criar um baita código servidor backend, que processe diversos tipos de transações, é melhor dividir estas transações em funções separadas e independentes, já que cada uma pode possuir perfil de tráfego e necessidade de segurança diferentes.

Eu não criaria um **CRUD** completo empacotado em um só programa. É melhor criar funções separadas e atribuir-lhes: Capacidade e permissões separadas.