|  |
| --- |
|  |
| Ver a imagem de origem  Certifield Developer Associate |

**IAM = Indentity Acess Management:**

E o local onde você consegue criar Usuários, grupos e permissões

* **Multifactor Autentication:** Logar em um site, normalmente você tem seu usuário e sua senha, Se os desenvolvedores tiverem apenas o user e o password, ficará um buraco de segurança bem grande, e aí onde habilitamos o MultiFactory Autentication

User, Pass, Token, Auth com isso depois de inserir o usuário e a senha ele vai pedir um código de autenticação que abrimos o app, e colocar o código de autenticação.

Eleva ao topo a segurança

* **Identity Federation:**  Se na sua empresa você já tem um AD (Active Directory) Você conseguira fazer um link e os usuários, vão utilizar esta base de dados para autenthicar na AWS.
* **Permissions: (Granular)** Você consegue aplicar permissões a usuários (Users) específicos, Exemplo tenho um usuário que queira criar maquinas EC2, mais somente do tamanho Small e que tenha no Max. 12 GB de Memória RAM, Max. 1 Terabyte de Disk.
* **Passwords:** Podemos criar uma política de rotatividade da senha, que ela expira em 3 meses, ou a cada 6 meses, e tenque ter um grau de dificuldade especifico. Consigo integrar todos os serviços ou **Services** (ou seja, o mesmo usuário poderá dar permissões ao S2 ao S3 DB e assim por diante.
* **Compartilhamento de Acesso:** consigo compartilhar o acesso da plataforma com outros usuários utilizando o IAM criando usuários pra cada um (recursos da plataforma com outros usuários) **Shared**
* **PCI DSS Complance:** Você tem um website, ele está hospedado na AWS, se você configurou ele inteiro com códigos dentro da AWS e agora eu preciso receber pagamentos via cartão de credito. (Visa e Mastercard) Você tem a garantia que os dados do seu cliente não vão ser compartilhados e estão seguros (boas práticas)
* **Criação de Grupos de usuários (Groups)** Dentro do IAM, podemos criar os Grupos de usuários (Users) que trabalham com: Databases, S3, EC2, e etc. Criando grupos eu consigo organizar melhor os meus usuários dentro da organização (empresas)
* **ROLES:** Regras que você associa, tanto pra usuários (Users), quanto para Groups, Regras de Acesso, e regras de serviço.

**Por exemplo:** Imagine que você tem uma web Server e este servidor hospeda uma página, e essa página tem um link com um Banco de Dados (Data-base), onde o Usuário que está acessando a página (Client) ele consegue acessar a página dele, e acessar as informações!

Só que o formato da página está no Web Service, e as Informações estão no Banco de Dados.

Conseguintemente, eu tenho 02 serviços aqui: um chamado EC2, e o outro DB e o serviço de EC2 tenque acessar o serviço de DB eles são 02 Instancias de serviços diferentes com uma frequência muito grande.

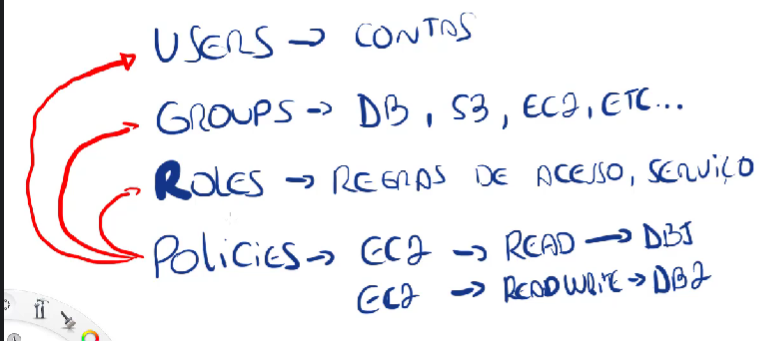
Conseguintemente ele tenque ter Regras (Roles) ou permissões, eu tenho que aplicar uma regra, olha o EC2 pode acessar o DB.

**Então:**

São regras de acesso a serviços que pode ser aplicada a usuários, quanto pra grupos e serviços também.

* **Polices:** O que esse usuário deve fazer, então eu crio uma police e falo: olha esse servidor EC2, ele pode acessar somente em Leitura (Read) Data-base I, esse EC2 tem Read-Write e pode acessar o Data-Base II e por ai vai.

**A Police você pode aplicar a:**



Resumindo:

Você cria os usuários, você divide eles em grupos de usuários, associa regras (Roles), e você define dentro das políticas o que eles podem fazer, consigo gerenciar 100%!

Laboratórios:

Hands On IAM Groups:

User Groups -> A melhor forma de criar e separar os departamentos ou funções e por grupos.

Posso segmentar os grupos, por funções ou Departamentos.

Interessante criar por funções, serão criados 02 Servidores, um para Administrador de Banco de Dados DBA, e outro para Servidores.

2 Grupos, um primeiro grupo para quem trabalha com server (EC2), e o Outro DataBases

Criando os grupos eu vou adicionar as Polices, e depois os Users com as minhas regras.

## EC2 INTRO

* Marquinas Virtuais ou (VMs)
* EBS-only (Elastic Block Storage)

Tipos de maquinas de acordo com sua categoria:

**General Purpose.** Medim, large, xlarge, 2xlarge, 4xlarge, metal

**Computer Optimized**

**Memory Optimized**

**Accelerated Computing GCPU**

**Storage Optimized**

**Instance Features**

**Mersuaring Instance Performace**

Bursty – colocar acima do que está especificado na máquina.

# **General Purpose:**

Me general purpose eu tenho maquinas virtuais para uso geral.

Types: A1, T3, T3a, T2, M5, M5a, M5n, M5zn, T4, Mac.

Observation: instance storage: NVMe: Now Volaty Memory express Solid State Drive

Get example:

A1 VirtualCPU Memory Storage networking Perfomance (Gbps)

2 4Gib EBS-Only up to 10 (Gbps)

# **Computer optimazed**

Se você tem Servidores, onde fazemos procuras, parte computacional um pouco maior, (marchine learning). Quantidade maior de cálculos.

**EBS – Bandwidth:** Quantidade de Banda para acessar o Disco.

Get example:

C5.large vCPU Memory Gib instance Storage networking Bandwidth(GB)

2 4 EBS-Only Up to 10

**MEMORY OPTIMAZED**

Dedicado para Servers para Armazenar Data-base com Hight Memory, focando bem a parte de memória nestes tipos de Instances.

**Name Processors Memory (GiB) Instance Storage (GB) networking Bandwidth**

**u-3tb1.56xlarge 224 3,072 EBS-Only 50 (Gbps)**

# **Accelerated Computing (Graphics Processor Unit)**

Instance GPUs vCPU Mem (Gib) GPU Memory (Gib) Networking performace

P2.xlarge 1 4 61 12 Hight

# **Storage Optimazed**

Tipos de instancia voltados especificamente para Espaço em disco e armazenamento dedicado.

Get exemple:

Instance size vCPU Memory (GiB) Instance Storage networking Bandwidth(GB)

Im4gn.xlarge 2 8 1 x 937 AWS nitro SSD Up to 25Gbps

**Instance Features**

Existem especificamente 02 tipos de maquinas, são maquinas que tem **Fixed Performace Instances**, se a máquina tem por exemplo: 5 processors, ela chega ao máximo de 5 processadores e para por lá.

Se você utiliza uma máquina Burstable Performace Instance, por exemplo uma máquina T3, a perfomacer dela pode ir um pouco acima do que está na especificação (Specific)

**Measuring Instance performance**

Como você mede a performance de cada máquina em CPus e em memória.

**EC2 Pricing:**

Valores das Maquinas Virtuais EC2 temos 02 métodos de cobrança, On-demand e Free-Tier

O método de cobrança free-tier você tem 12 meses (free), normalmente as maquinas com a instancia T2. micro por exemplo.

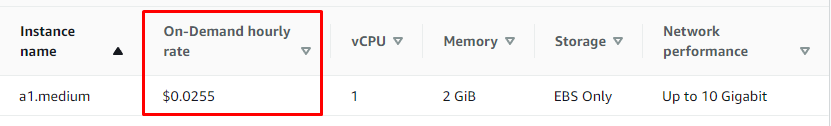
As maquinas ON-Demand são todas as outras maquinas que tem um valor:

Hoje em dia, a forma de cobrança, e através de per-second-billing feita em segundos.

* Pode ser por horas, se você escolher uma instancia reservada ou uma Instancia Spot (Dedicate Host) or Spot Instance.

Há AWS cobra por mês fechado, ou seja, o mês tem 750 horas então um exemplo de uma máquina ON-demand:

Get example:



Esse exemplo de instancia EC2 vai custar: 0.0255 X 750 Hours = US$ 19.12 Dollars.

Today date Dollars USD 21/03/2022 US$ 1.00 = R$ 5.04

US$ 19.12 X BRL5,04 = 96,36 Centavos em Real Brasileiro (R$)

Maquina EC2 no final do mês ela vai custar em BRL R$ 96.36

Free-tier.

On-demand influência nos valores $$ location (region), Operation System (S.O, Type de EC2 (small, large, xlarge)

[EC2 On-Demand Instance Pricing – Amazon Web Services](https://aws.amazon.com/ec2/pricing/on-demand/)

Custos de On-Demand Amazon AWS

# Entendendo Volumes Elastic Block Store ( EBS ):

Virtualização as Vms e você pegar uma máquina física, e dividir ela em cada parte.

Da mesma forma que virtualizo uma máquina física, consigo virtualizar um disco:

**EBS EBS**

**EBS EBS**

A AWS consegue pegar os discos físicos, e virtualizar esses discos.

Automaticamente, quando adquirimos um Volume EBS da Amazon, ela tem uma cópia desse volume idêntica. Em outra região.

# Variaveis do Sistema EBS:

HDD -> Dentro do tipo HD, temos 02 subcategorias: ST1 Thowput otimizado. SC1 Cold Disc

SSD -> Dentro do tipo SSD, temos IO1 (hyperFormace) e o GP2 (General Porpose Two)

Utilizamos um termo chamado IOPS que significa: Input Output Operation per Second

Significa o que: quantas operações ele consegue escrever no disco por segundo.

Temos também o **Throughput** que é a quantidade de Megabits por segundo que você consegue escrever no disco.

O Max IOPS / Instance:

# Entendendo Load Balances

Normalmente a AWS Prover 02 tipos de Load Balancers que são aplication networking e o clássico que e chamado de Elastic Load Balancer!

Porem vamos ver um conceito: o que é um Load Balancer:

Imagine que você está acessando de um computador, a sua requisição e traduzida através de um servidor DNS, ele passa qual endereçamento IP esta aquela requisição daquele site que vc está querendo.

O conteúdo que você quer está dentro de uma solução AWS, a empresa que detem o site, tem sua estrutura dentro da AWS, ela tem os servidores web:

Cliente -> DNS -> sua requisição traduzida Load Balancer -> Server1, Server2, Server3

A partir do momento que a sua requisição e apontada a um IP, esse IP de DNS não e o IP do servidor! Ele e o endereçamento IP do LOAD BALANCER

Temos o IP de Origem: 89.7.4.1 public e ele vai chegar ao endereçamento de IP 171.1.120

Quando ele chegar no LOAD BALANCER, o que ele vai fazer: ou dividir o trafego, ou ele vai enviar para o servidor que tem aquele recurso. (cada server tem seu endereçamento IP)

BALANCIAMENTO DE CARGA EM SERVIDOR COM CONTEUDO IGUAL.

A AWS prevê 03 tipos de Load Balancers:

* **Application Load Balancer:**

Ele sempre trabalha na camada de aplicação, então ele e responsável pelos protocolos de HTTP/HTTPS e trabalha na **Layer 7**. Ele consegue identificar a origem e o destino desse pacote. (consegue fazer o balanceamento porque ele sabe o tipo de trafego que está passando por lá)

* **Networking Load Balancer:**

Ele e mais pra parte de Networking Layer, parte de TCP e ele trabalha dentro da camada 4 (Layer 4)

Ele consegue filtrar portas, ele consegue filtrar origem e destino, mas ele para na camada 4 (Layer 4)

* Classic Load Balancer (Elastic Load Balancer)

Fazia o Http/Https -> Layer 4 e o Layer 7

Quando a sua requisição chega no Load Balancer, ele envia sua requisição para o server, como eles sabem que ele deve voltar para o servidor de origem.

Os Load Balancers utilizam um sistema chamado X-forward-for-header.

No cabeçalho da requisição Http/Https tem um campo aonde ele preenche esse campo com o endereçamento IP de Origem: 84.7.4.1 ele consegue identificar a origem e devolver

Https\_Header\_Field;

* Gateway Load Balancer

Seu cliente faz uma solicitação ao seu aplicativo. O load balancer recebe a solicitação com base nas configurações da tabela de rotas definidas em sua VPC, Internet Gateway ou Transit Gateway. O balanceador de carga roteia solicitações para um grupo de destino que consiste em uma frota escalável de dispositivos (por exemplo, firewalls, sistemas de inspeção profunda de pacotes, sistemas de filtragem de URL etc.) para processar fluxos de tráfego. O dispositivo virtual processa o tráfego e o encaminha de volta ao balanceador de carga ou descarta o tráfego com base em sua configuração. Esse tipo de balanceador de carga atua como uma ligação entre a origem e o destino. O balanceador de carga encaminha o tráfego para seu destino.

Leitura Complementar de Load Balancers:

O aplication load balancers opera no nível de camada 7 (Layer 7) roteando o trafego a destinos e instancias do EC2, contêineres endereços IP e funções de Lambda de acordo com o conteúdo da solicitação.

Ideal para Load Balancers avançados de trafego HTTP e HTTPS, o application load balancer oferece um roteamento avançado de solicitações e entrega de arquiteturas de aplicativos modernos, incluindo microserviços e aplicativos baseados em contêiners.

O Application Load Balancer simplifica e aprimora a segurança dos aplicativos, garantindo que as cifras e protocolos SSL/TLS mais recentes sejam sempre usados.

Principais Recursos

--Balanceamento de Carga da camada 7 (Layer 7)

Você pode usar o Load Balancer é compatível com a terminação de HTTPS entre os clientes e o Load Balancer. O Application Load Balancer também oferece gerenciamento de certificados SSL por meio da AWS Identity and Acess Management (IAM) e do AWS Certificate Manager para políticas de segurança definidas!

--Server Name Indicator (SNI)

Indicação de Nome de Servidor, e uma extensão do protocolo TLS quer permite que o cliente indique o nome de host a ser conectado no início do handshake do TLS.

O Load Balancer pode apresentar vários certificados pelo mesmo listerner seguro, o que permite oferecer suporte a vários sites seguros, usando um único listerner seguro. Os Application Load Balancers também oferecem suporte a um algoritimo inteligente de seleção de certificado SNI. Se o nome de host indicado por um cliente corresponder a vários certificados, o Load Balancer determinará o melhor certificado a usar considerando os fatores, incluindo os recursos do cliente.

--Endereço IP como destinos

É possível balancear a carga de qualquer aplicativo hospedado na AWS ou no local usando os endereços IP dos back-ends aplicativo como destinos. Isso permite fazer o balanceamento da carga para um back-end de aplicativo hospedado na mesma instancia.

Cada aplicativo hospedado na mesma instancia pode ter um grupo de segurança associado e usar a mesma porta.

Você também pode usar endereços IP como destinos para balancear a carga de aplicativos hospedados localmente (por meio do Direct Connect ou de uma conexão VPN) em VPCs emparelhadas e no EC2-Classic (usando o ClassicLink). A capacidade de balancear carga entre recursos na AWS local, ajuda a executar migrações, de bursting ou failover para a nuvem.

--Funções de Lambda como destinos

O Application Load Balancer é compatível com a invocação das funções do Lambda para atender as solicitações HTTP(S), permitindo que os usuários acessem o aplicativos sem servidor de qualquer cliente HTTP, incluindo navegadores da Web. Você pode registrar funções Lambda como destino para um load balancer e utilizar o suporte de regras de roteamento baseado em conteúdo para rotear solicitações para funções de Lambda diferentes.

Você pode usar um aplication Load Balancer com um endpoint HTTP comum para aplicativos que usam servidores e computação sem servidor.

Você pode criar um site inteiro usando funções de Lambda ou combinar instancias do EC2, contêineres, servidores locais e funções de Lambda para criar aplicativos.

--Alta Disponibilidade

Um aplication Load Balancer exige que você especifique mais de uma zona de disponibilidade (Availability zone). Você pode distribuir o trafego de entrada entre os seus destinos em várias zonas de disponibilidade.

Um aplication Load Balancer escala automaticamente sua capacidade de gerenciamento de solicitação de resposta ao trafego de entrada de aplicativos.

--Recursos de Segurança

Ao utilizar um Amazon Virtual Private Cloud (VPC), você pode criar e gerenciar grupos de segurança associados ao Elastic Load Balancing para disponibilizar opções adicionais de rede e segurança.

Você pode configurar um aplication Load Balancer voltado para internet ou criar um load balancer interno (não voltado para a internet)

--Roteamento Baseado em Conteúdo

Se o aplicativo for composto por vários serviços individuais, um aplication Load Balancer poderá rotear uma solicitação para um serviço com base no conteúdo da solicitação

--Roteamento Baseado em Host

Você pode rotear uma solicitação de cliente baseada no campo Host do cabeçalho de HTTP. Dessa forma, e possível rotear do mesmo load Balancer para vários domínios.

--Roteamento Baseado em caminho

Você pode rotear uma solicitação de cliente baseada no caminho de URL do cabeçalho HTTP.

--Roteamento baseado em método HTTP

Você pode rotear uma solicitação de cliente baseada no valor de qualquer cabeçalho HTTP padrão personalizado

--Roteamento baseado em parâmetros de sequência de caracteres de consulta

Você pode rotear uma solicitação de cliente baseada na sequência de caracteres de consulta ou de parâmetros de consulta.

--Roteamento baseado em CIDR de endereço IP de origem

Você pode rotear uma solicitação de cliente baseada em CIDR do endereço IP de origem onde a solicitação for proveniente

--Suporte a aplicativos Conteinerizados

O Application Load Balancer oferece suporte aprimorado a contêineres, executando o Load Balancing entre várias portas na mesma instancia do Amazon EC2.

A estreita integração com o Amazon EC2 Container Service, disponibiliza uma oferta de contêineres gerenciada. O ECS permite que você especifique uma porta dinâmica na definição de tarefas do ECS, disponibilizando ao contêiner uma porta não utilizada quando programado a instancia do EC2. O programador do ECS adiciona automaticamente a tarefa ao Load Balancer usando essa porta.

--Suporte ao HTTP/2

O HTTP/2 e uma nova versão do Hypertext Transport Protocol (HTTP), que usa uma única conexão multiplexada para permitir varias solicitações sejam enviadas na mesma conexão. Ela também compacta os dados de cabeçalho antes de envia-los em formato binário e oferece compatibilidade com conexões SSL para os clientes

--Suporte ao WebSockets

As sticky session são um mecanismo usado para rotear solicitações do mesmo cliente para o mesmo destino. O Application Load Balancer oferece suporte ao sticky session usando cookies gerados por Load Balancers.

Se você habilitar as sticky sessions, o mesmo destino recebera a solicitação e você poderá usar o cookie para recuperar o contexto da session ou da sessão. Esse recurso de sticky sessions, ou stickiness, e definido por grupos de destinos

--Monitoramento operacional

O Amazon CloudWatch relata métricas do Application Load Balancer como número de solicitações enviadas para o Load Balancer e armazenar os Logs no Amazon S3 para análise posterior.

Os logs são compactados e têm uma extensão do arquivo gzip. Os logs compactados economizam espaços de armazenamento de largura de banda usada para transferências, além de serem uteis para o diagnósticos de falhas de aplicativos e a análise de trafego web.

--Registro de Log

Você pode usar o registro de logs de acesso para registrar todas as solicitações enviadas para o load balancer e armazenar os logs no Amazon S3 para análise posterior. Os logs são compactados e tem uma extensão de arquivo gzip. Os logs compactados economizam espaço de armazenamento e largura de banda usada em transferências, além de serem uteis para o diagnostico de falhas de aplicativos de analises de trafego web.

Você também pode usar o AWS CloudTrail para registrar as chamadas de API Application Load Balancer na sua conta e entregar arquivos de log. O histórico de chamadas API permite executar analises de segurança, rastrear alterações de recursos e realizar auditorias conformidade.

--Proteção contra Exclusão

Você pode habilitar o recurso de proteção contra exclusão em um Application Load Balancer para impedir que ele seja excluído acidentalmente.

--Solicite o Monitoramento

O Application Load Balancer injeta o “X-Amzn-Trace-Id” um novo cabeçalho HTTP identificador personalizado, a todas as solicitações enviadas para o Load Balancer. O monitoramento de solicitações enviadas ao Load Balancer.

O monitoramento de solicitações permite rastrear uma solicitação por meio seu ID único conforme a solicitação passa pelos vários serviços que compõem sites e aplicativos distribuídos.

É possível usar o identificador de monitoramento exclusivo para descobrir qualquer problema de performance ou atraso na pilha de aplicativos com a mesma granularidade de uma aplicação individual.

--Firewall de aplicativos web

Você já pode usar o AWS WAF para proteger aplicativos web em Application Load Balancers. O AWS WAF é um firewall de aplicativos web que ajuda a proteger esses aplicativos contra exploits comuns na web, que podem afetar a disponibilidade do aplicativo, comprometer a segurança ou consumir recursos me excesso.

--Modo de início lento com algoritmo de balanceamento de carga

O Application Load Balancer aceita um algoritmo de balanceamento de carga round-robin. Além disso, o Application Load Balancer aceita o modo de início lento com algoritmo Round Robin que permite que você adicione novos destinos sem sobrecarrega-los com uma enxurrada de solicitações.

Com o modo de início lento, os destinos são ativados antes de aceitarem as partes de solicitações que são cabíveis a eles, com base em um período de aumento na quantidade especificado por você.

O início muito lento e muito útil para aplicativos que dependem do cache e precisam de um período de ativação antes de poderem responder a solicitações com performance ideal.

--Autenticação de Usuários

Você pode descarregar a funcionalidade de autenticação dos seus aplicativos no Application Load Balancer. O Application Load Balancer autenticará com segurança os usuários, conforme eles acessarem os aplicativos na nuvem.

O Application Load Balancer e totalmente integrado ao Amazon Cognito, o que permite que usuários finais façam autenticação por meio de provedores de identidade social como Google, Facebook e Amazon, e por meio de provedores de identidade empresarial, como Microsoft Active Directory via SAML ou qualquer provedor de identidade (IdP) em conformidade com o OpenID connect. Se você já tiver uma solução personalizada de IdP compatível com o OpenID connect, o Application Load Balancer também poderá autenticar usuários empresariais ao estabelecer conexão diretamente com seu provedor de identidade.

--Redirecionamentos

O Application Load Balancer pode redirecionar uma solicitação recebida de um URL para outro. Isso inclui a capacidade de redirecionar solicitações HTTP para solicitações HTTPS, o que permite cumprir metas de conformidade para navegação segura, alcançar uma melhor posição nas pesquisas e obter uma alta pontuação de SSL/TLS para o site.

Também é possível usar redirecionamentos para enviar os usuários a um site diferente, como de uma versão antiga de um aplicativo para a nova.

--Resposta Fixa

O Application Load Balancer pode controlar quais solicitações de cliente são atendidas pelos aplicativos. Isso permite responder a solicitações recebidas com códigos de erro HTTP e mensagens de erro personalizadas do próprio load balancer, sem enviar a solicitação para aplicativo.

--O que é Networking Load Balancer?

O Networking Load Balancer opera no nível de conexão (camada 4) ou Layer 4, roteando conexões a destinos e instancias do Amazon EC2, microserviços e contêineres em uma Amazon Virtual Private Cloud (VPC) de acordo com os dados do protocolo IP. Ideal para o balanceamento de carga de trafego TCP e UDP, Networking Load Balancer e capaz de processar milhões de solicitações por segundo, mantendo latências ultrabaixas.

O Networking Load Balancer e otimizado para gerenciar padrões de trafego súbitos e voláteis usando um único endereço IP estático por zona de disponibilidade (Availability zone).

Ele e integrado a outros serviços populares da AWS, como Auto Scaling, Amazon EC2 Containers Service (ECS), Amazon Cloud Formation e AWS Certificate Manager (ACM)

--Principais Recursos

Load Balancing baseado em conexões

Você pode fazer o balanceamento de carga de trafego TCP e UDP, roteando conexões de destinos como instancias do Amazon EC2, microserviços e contêineres

--Alta disponibilidade

O networking Load Balancer e altamente disponível. Ele recebe o trafego de clientes e distribui entre destinos na mesma zona de disponibilidade.

Além disso, o Load Balancer monitora a integridade dos destinos registrados e garante que o roteamento seja feito apenas para destinos íntegros.

Quando o Load Balancer detecta um destino com problemas de integridade, cessa o roteamento de trafego para esse destino e roteia o trafego para os demais destinos íntegros.

Se todos os destinos de uma zona de disponibilidade tiverem problemas de integridade e você tiver configurado destinos em outra zona de disponibilidade, O Networking Load balancer fara automaticamente um failover para rotear o trafego para destinos íntegros nas outras zonas de disponibilidade

--Alto Throughput

O Networking Load Balancer foi criado para assimilar o crescimento de trafego e pode fazer Load Balancing de milhões de solicitações por segundo. Além disso, também absorve padrões de trafego súbito voláteis

--Baixa Latência

O Networking Load Balancer oferece latências extremamente baixas para aplicativos que dependem da latência

--Preservação de endereço IP de Origem

O Networking Load Balancer preserva o IP de origem do lado do cliente, o que permite que o back-end veja o endereço IP desse cliente. Esse endereço pode ser usado em procedimento adicional pelos aplicativos

--Suporte a IP Estático

O Networking Load Balancer fornece automaticamente um IP estático por zona de disponibilidade (sub-rede), fornecendo assim o seu próprio IP Fixo.

--Transferência de Carga TLS

O Networking Load Balancer suporta o termino da sessão TLS do cliente.

Isso permite que você descarregue tarefas de termino de TLS para o Load Balancer, preservando o endereço IP de origem para os seus aplicativos back-end. Você pode escolher entre politicas de segurança predefinidas para seus listerner TLS, a fim de atender aos padrões de conformidade e segurança. Pode ser usado o AWS Certificate Manager (ACM) ou AWS Identity and Acess Management (IAM) para gerenciar os certificados do servidor.

Você pode usar o SNI (Indicador de Nome do Servidor) para atender a vários sites seguros por meio de um único listerner TLS. Se o nome do host cliente corresponder a vários certificados, O Load Balancer selecionara o melhor deles, que será usado com base em um algoritmo de seleção inteligente

--Verificações de Integridade

O Networking Load Balancer oferece suporte a verificações de integridade de destinos de rede e aplicativos. A integridade da rede e baseada na resposta geral do destino de trafego normal. Se o destino responder lentamente ou não responder novas conexões, O Load Balancer marcará esse destino como não disponível.

As verificações de integridade dos aplicativos também podem ser usadas para obter mais detalhes. A sondagem periódica de uma URL especifico em um determinado destino, permite integrar a integridade do aplicativo real. Uma visibilidade completa das verificações de integridade e possível motivos de falha também e disponibilizada por meio de “códigos de Motivo” na API do Networking Load Balancer e nas métricas da Amazon CloudWatch anexadas as verificações de integridade de destinos. Esse recurso proporciona diagnósticos rápidos e deputação avançada.

--Failover de DNS

Se não houver destinos íntegros registrados no Networking Load Balancer ou se os nós do Networking Load Balancer em uma determinada zona apresentarem problemas de integridade, o Amazon Route53 direcionara o trafego para os nós de Load Balancer em outras zonas de disponibilidade.

--Integração com serviços da AWS

O Networking Load Balancer e integrado a outros serviços da AWS, como Auto Scaling, Elastic Containers Service (ECS), Cloud Formation, Elastic BeanStalk, CloudWatch, Config, Cloudtrail, CodeDeploy e AWS Certificate Manager

--Conexão TCP de Longa Vida

O Networking Load Balancer oferece suporte a conexões TCP de longa vida, ideais para aplicativos do tipo WebSocket.

--Suporte a APis centralizadas

O Networking Load Balancer usa a mesma API que o Application Load Balancer. Dessa forma, você pode trabalhar com grupos de destinos, verificações de integridade e usar o Load Balancer em varias portas na mesma instancia EC2 para oferecer suporte a aplicativos Conteinerizados.

--Monitoramento e auditoria Robustos

O Amazon CloudWatch relata métricas do Networking Load Balancer. O CloudWatch fornece métricas com Active Flow Count, Heath host Count, New Flow Count e Porcess Byte, entre outras. O Networking Load Balancer também e integrado ao AWS Cloudtrail rastreia as chamadas da API ao Application Load Balancer

--Registros em Logs aprimorados

Você pode usar o recurso de logs de fluxo para registrar todas as solicitações enviadas ao Load Balancer. Os logs de fluxo, capturam informações sobre o trafego IP e percorre as interfaces de rede na VPC, Os dados de Logs de fluxo são armazenados usando o Amazon CloudWatch Logs.

Quando os listerner TLS são adicionados ao Networking Load Balancer, e possível optar por ativar os Logs de acesso, que serão enviados para o bucket do S3 de sua preferencia. Os Logs de acesso capturam informações sobre todas as conexões TLS recebidas por seu listerner, proporcionando visibilidade dos Hand shakes TLS bem-sucedidos e com falha.

--Isolamento Zonal

O Networking Load Balancer foi criado com arquiteturas de aplicativos em uma única zona. Se algo na zona de disponibilidade falhar, executaremos automaticamente o failover para outras zonas de disponibilidade integras.

Recomendamos que clientes configurem os Load balancers e os destinos em várias Avaibility Zones para obter alta disponibilidade. No entanto, O Networking Load Balancer pode ser habilitado em uma única zona de disponibilidade para oferecer suporte a Arquiteturas que exigem isolamento Zonal.

**Load balancers: Heath Check**

**Ping Protocol** [HTTP]

**Ping Port** 80

**Ping Path** [Index.html]

*Advanced Details*

**Response Timeout** O tempo de espera de resposta para o Heath Check!

Significa, que ele vai enviar uma requisição e se ele não receber uma resposta em 2 segundos ele vai contar aquela requisição com a falha!

**Interval** O tempo entre os Heath Checks de quanto em quanto segundos e pra ele ir lá e verificar no servidor se ele está disponível ou não.

**Unheathy threshold** Quantos TimeOuts são necessários, pra falar que aquele servidor caiu, se eu receber 2 TimeOuts de HeathCheck, significa que caiu.

**heathy threshold** quando ele voltar, quanto tempo eu preciso aguardar? 10 vezes

Route 53

Serviço AWS para DNS, se você precisa trabalhar com Domain Name Server, você pode utilizar o DNS.

O que o ROTA 53 pode fazer, primeiro ele e responsável pelas resoluções de endereçamento IP e necessário um endereçamento IP pra fazer essa tradução, dentro do Route53, ele não trabalha de forma isolada.

Por exemplo: se você registra o seu domínio e hospeda o seu site, em uma região nos EUA por exemplo, a replicação dessas informações ela e por toda a AWS Cloud.

Você terá uma redundância, de todas as localizações, e você tem garantido por eles 100% SLA (Availability) quando os serviços de DNS, estiverem dentro da **Rota53**, essas informações são passadas para todos os outros servidores de todas as regiões, de todas as **Availability zones**, se você tem um usuário no Brasil, e o registro foi feito nos EUA, essa replicação e passada ao Brasil também.

E quem faz essa requisição, ele não precisa ir em todos os servidores de DNS até chegar naquele registro. Todos os registros estão em todos os servidores.

Voce tem uma disponibilidade de SLA de 100% que a própria **AWS** garante.

**Ponto Importante:**

Redundância de todas as localizações:

E que dentro do Rota 53 você pode habilitar que nós chamamos de **Geo\_location,** então você pode filtrar, quem acessa o seu site ou quem não acessa o seu site através de Geo\_Location

**Failover**

Temos esse sistema, se você tem 02 servidores web: o servidor 1.1.1.1 e o servidor: 2.2.2.2, através de regras de DNS, você pode apontar para um servidor, e se ele se tornar indisponível, ele automaticamente passa para os outros.

E claro que tem todo um sistema de cache, que as vezes pode demorar algumas horas, mais o que a rota 53 faz isso automaticamente, e você não precisa ficar adicionando

**Domain Name Servers:**

DNS –

Domain (Dominio) é nada mais nada menos que o nome do seu site.

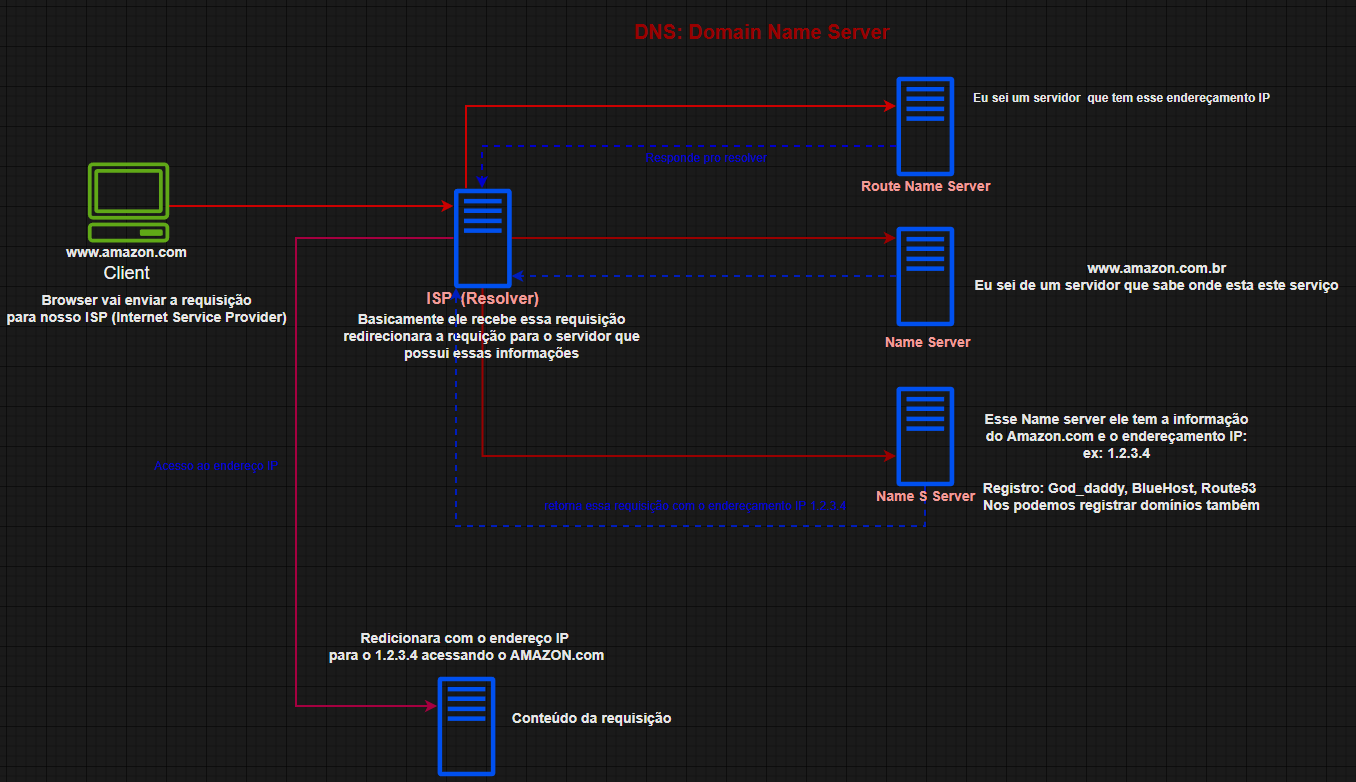
Porém existe uma diferença muito grande de alguém querer acessar o seu site, e chegar nos arquivos onde estão hospedados o seu site!

Basicamente, o Domain que pode ser [www.amazon.com.br](http://www.amazon.com.br) para o numero IPv6: 192.168.1.1 seja ele IPv4 e IPv6

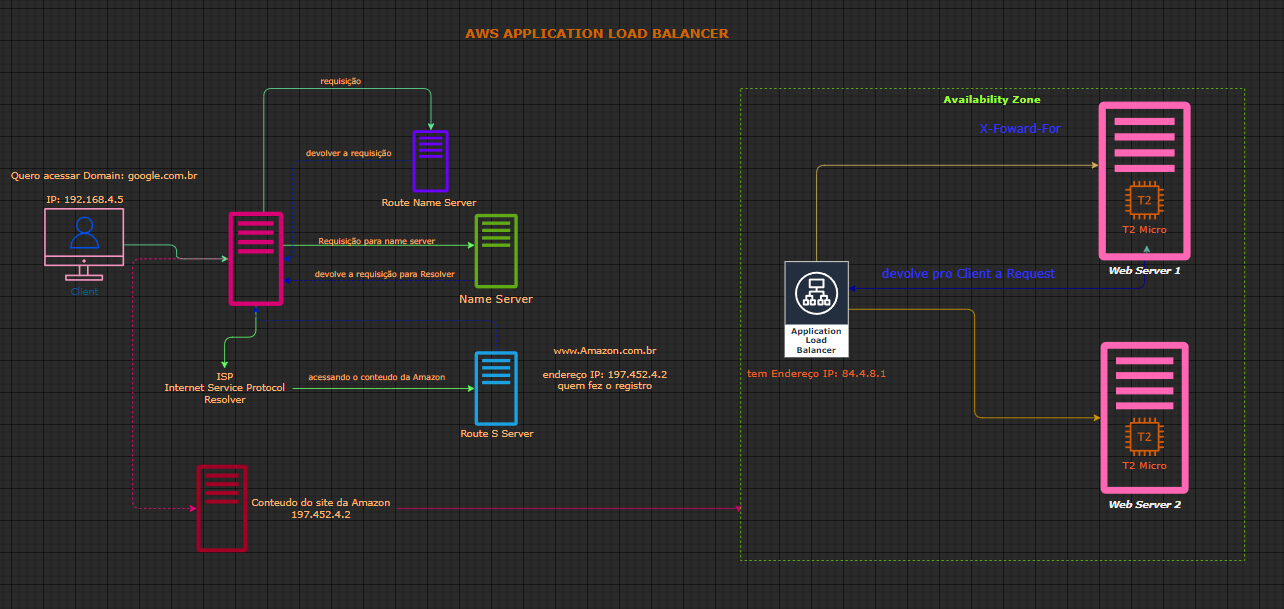
Como isso tudo funciona:

O computador: O browser vai enviar a requisição, para os registros de DNS, normalmente um PC ele está com 2 DNS apontados. Normalmente ele aponta para um grupo de servidores, vamos dar o nome de Resolver, normalmente este Resolver ele está dentro do ISP (Internet Service Provider) que é seu provedor.

Exemplificando no Grafico o DNS:



Com Load Balancers:



Registros

Visualizar um registro chamado Host A (pode ser definido como) e um tipo de registro dentro da arvore de DNS

Tipos de Registros:

* Host A (A ou AAAA) o que e o Host A, e um tipo de registro onde dentro da arvore de DNS, O site completo ele esta apontado ao endereçamento IP: 1.1.1.1 (aonde esta os arquivos desse site em especifico)

O A e para IPv4(A) e p IPv6(AAAA) esse registro pode ser usado para internos como por exemplo: você tem o servidor para AD server 1, esse endereçamento IP192.168.10.1 (supondo que seja o servidor de Active Directory)

* Alias (Cname) OU Canonical Name, ele e muito utilizado em conjunto com o Host A:

Digamos que eu queira um Subdomínio (subdomain) ele está apontado para 1.1.1.1

* Mail Exchange: Registros de Email, onde esta o servidor de e-mail, então colocamos através do MX.amazon.com, você também consegue colocar um numero aqui como prioridade 5 por exemplo.

Eu posso ter vários servidores de e-mail, com prioridades diferentes:

Mx.amazon.com 5

Mx.amazon.com 10

Então toda vez que eu visualizar um registro MX, significa que os Email serão enviados e encaminhados para aquele servidor em Especifico

* SRV (Service Record) Se você tem algum serviço, e precise que as aplicações aprendam sobre este serviço

Quando você tem um SRV apontado para o endereçamento IP (SRV) ele indica qual o serviço esta rodando: LDAP, ele informa também o que e o Target: Endereçamento IP, informa a Port: 8080, e a Priority (prioridade) qual e a Menor (onde e o primeiro a ser utilizado)

-Start of Authority (SOA) por Zona, normalmente ele e o Name Server (Primary Name Server)

Dentro do Soa: nos temos informações por exemplo como o Name Server, Qual e o administrador que registro aquele domínio (se a informação não estiver como privado) SN (Serial Numer) que toda vez que você altera um registro de DNS esse numero aumenta por 1, e lá dentro do SOA, tem as informações de Refresh Time, atualiza as informações e tabelas com os seus Records!

* Name Server: NS

Nós temos nada mais nada menos que as informações dos Records, e a melhor origem de todas as informações que eles podem estar, basicamente ele armazena o Start of Authority o (SOA)

* Pointer (PTR)

Ele basicamente faz o oposto que o Host A faz, para um registro de nome, ele dá o endereçamento IP, o Pointer pega o endereçamento IP (1.1.1.1) e converte para um domínio (name)

**Lab: Route 53**

DNS management: Através dela eu consigo gerenciar as minhas regras de DNS, que são os DNS contidos no DNS Records: Host A, Cname, MX, SRV, NS, Pointer

Traffic management: Consegue gerenciar o trafego, qual trafego vai pra que lado, conseguimos aplicar as politicas (policys) de peso, simple Router, eiteh Router, Failover (pra encontrar o servidor que está disponível)

Availability Monitoring: Onde Monitora onde os servidor estão ativos e qual não estão!

Register Domain: Onde registramos nossos domínios.

Configurando Topologias com Servidores Web.

100% baseado em como aumentar sua disponibilidade do seu site, ou da aplicação web.

Vamos montar 03 servidores:

Em localizações diferentes a que eu estou:



Com as 03 maquinas Virtuais Criadas, vamos coletar o endereçamento IP das Regions de London, Paris, Sidney

London Ipv4 Public: **18.130.106.37**

Paris Ipv4 Public: **35.180.42.206**

Sidney Ipv4 Public: **13.55.212.177**

Vamos tentar acessar através do endereçamento IP public.

Com os 03 Sites Funcionando.

Trabalhando com métricas no Route53

Simple Routing: vou enviar uma requisição de DNS para cada servidor (sendo que cada usuário vai acessar um servidor diferente de acordo com o DNS)

Failover Routing: Quando um se torna indisponível ele faz o giro pro outro.

GeoLocation Routing: Rotear usuários baseados na sua localização ´pra poder encontrar o servidor mais próximo, e esse servidor mais próximo se não estiver disponível, ele vai pro outro como segundo caminho.

GeoProximity Routing (Traffic Flow only) Ele vai pelo recurso, pelo servidor mais próximo.

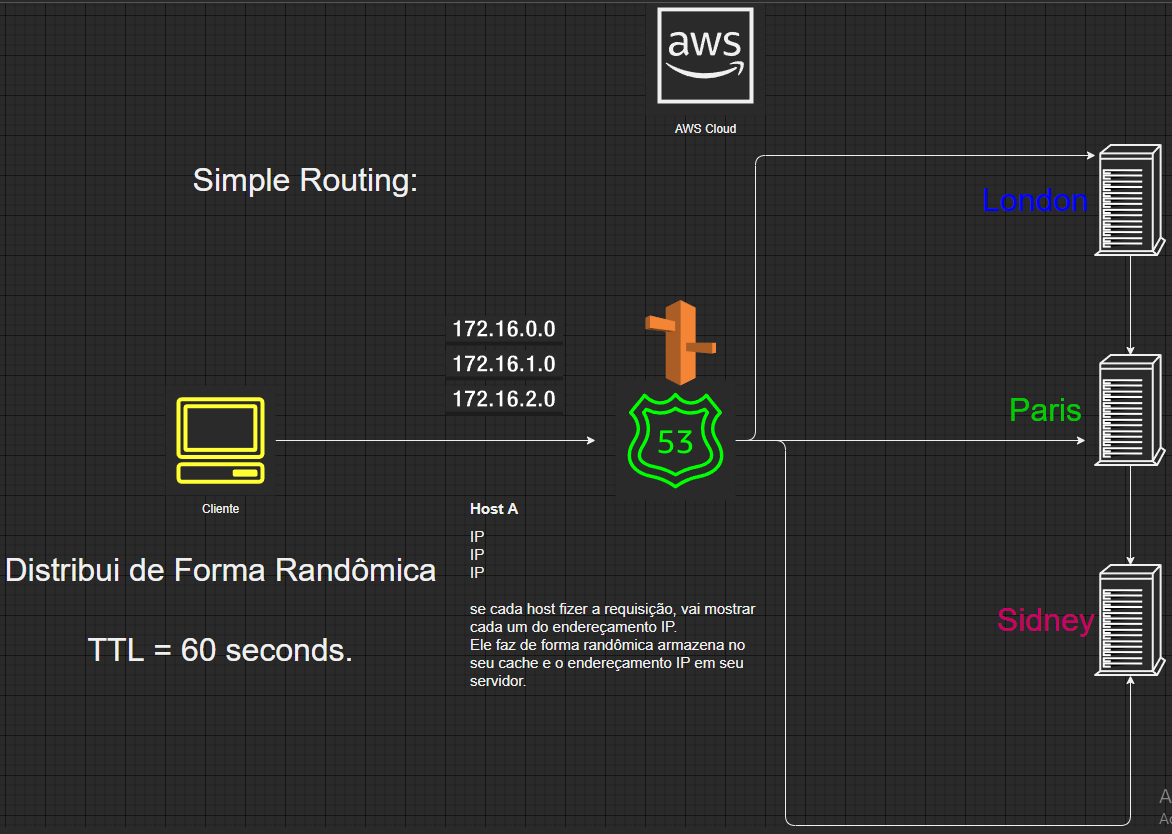
Latency-Based Routing: Ele vai pela Latencia, de acordo com a latecia, se esse servidor está próximo, ou tem uma latência menor ele vai pela menor

Multivalue Awser Routing: Se pode passar vários endereçamentos IP para mesma URL

Weighted Routing: Um peso, pra esse servidor vou enviar 80% trafego, e o outro 20% do trafego.

O TTL (Time to Live) Quando o cliente enviar pro DNS e o ROUTE53 vai falar assim, pra você chegar neste endereçamento IP 18.10.106.37, quando ele passar essa informação, o Time to Live quanto tempo em segundos, essa informação passada para o cliente, vai ser armazenada no cache do cliente.

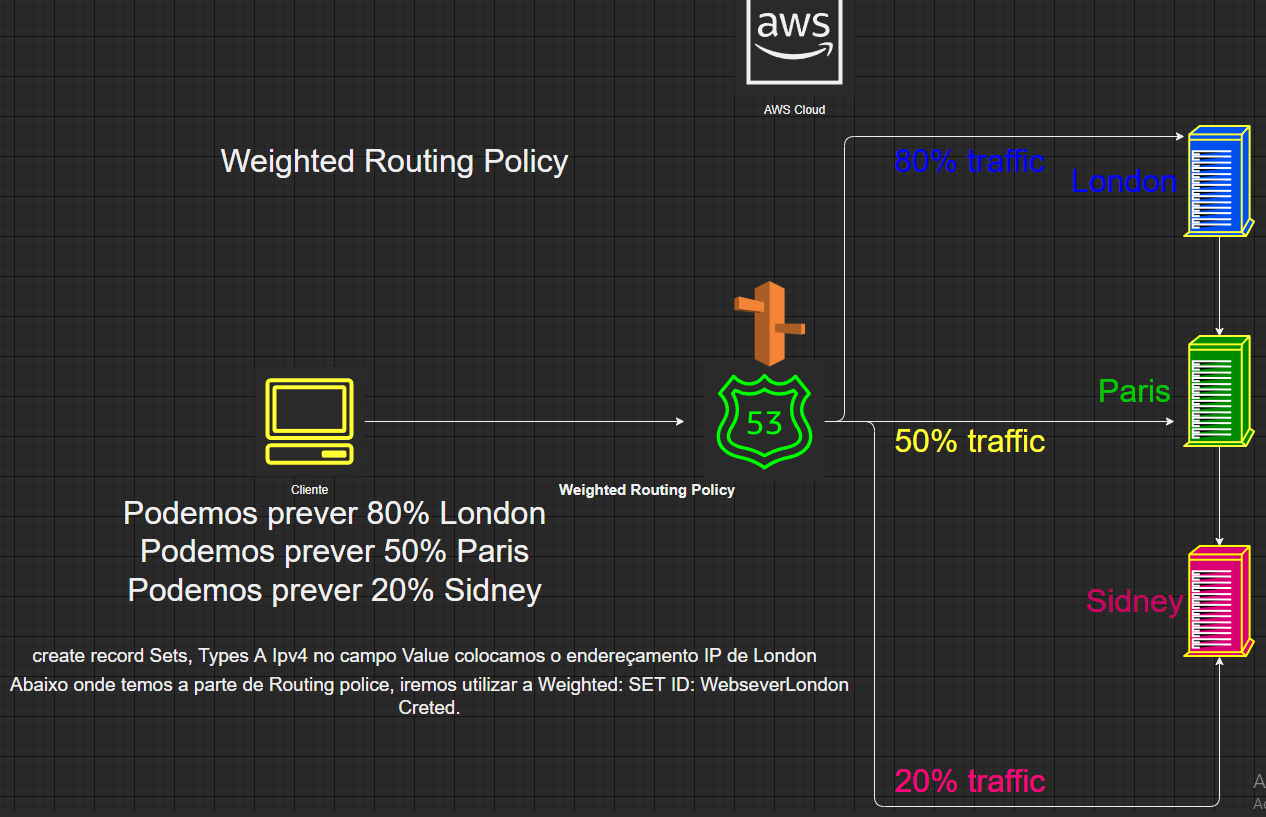
Create the Roles.

Simple Routing Policy

WEIGHTED ROUTING POLICY:

Vamos tirar a Policy anterior, indo em Delete Records Set

Criaremos uma Nova Policy Create Records Set

Não esqueça de alterar as TTL (Time to Line de cada Server) nas regras de Políticas. d

**Weigthted Routing Policy com Health Check!**

Temos a opção de criar um health Check, caso algum dos servidores se tornem indisponíveis, suponhamos que o server de London se torne indisponível, tem a funcionalidade de Setarmos um Health Check para verificar isto.

**Associate with Health Check: Yes.**

Configure to Health Check:

**Name:** HC-London

**What Monitor:** Endpoint

**Monitoring and Endpoint:**

**Specify:** IP Adress

**Protocol:** HTTP

**IP Adress: 18.130.106.37**

**Host Name: WWW.CLUBECLOUDBRASIL.COM**

Port.: 80

Path: INDEX.HTML

**Create Heath Check**

Depois que criarmos os 03 Health Check dos 03 Servidores, faremos a associação deles dentro do DNS Records

Derrubando o Server Sidney na EC2 (Stop), e verificando o nosso Health Check, definindo as configurações de Interval e as demais. Ele vai estar Unheath

**Latency Routing Policy**

**Se temos, 03 servidores, e queremos que os usuários sempre acessem os servidores mais próximos: através do Latency Routing Policy.**

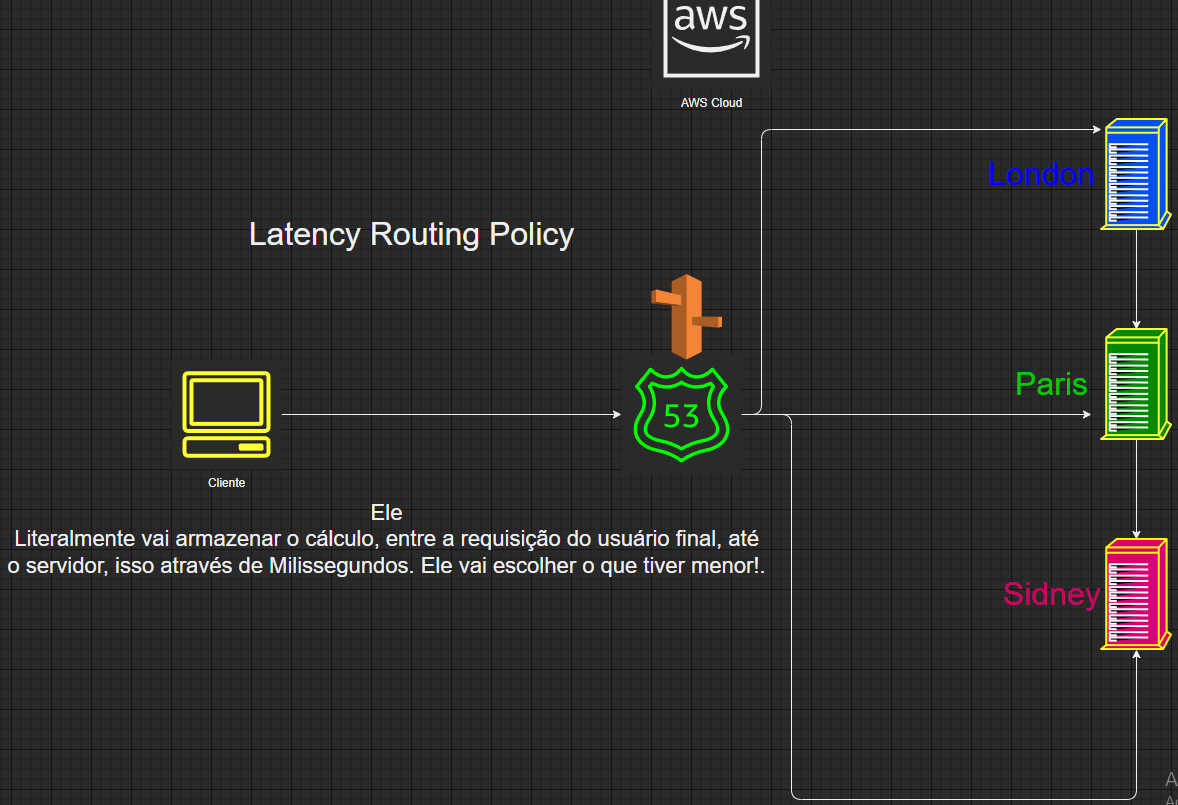
Ele Literalmente vai armazenar o cálculo, entre a requisição do usuário final, até o servidor, isso através de Milissegundos. Ele vai escolher o que tiver menor!

Utilizando o CMD, iremos bater o ping 18.130.106. 37 -t

Security Group Edit inbound Rules, criar um ICMP Custom ICMP 4 echo Request

**Security Group Edit inbound Rules, criar um ICMP Custom ICMP 4 echo Request**

**Fazer a mesma coisa com os Security Groups e adicionar o ICMP pra todos**.



Criaremos o Primeiro Records:

Routing policy: Latency

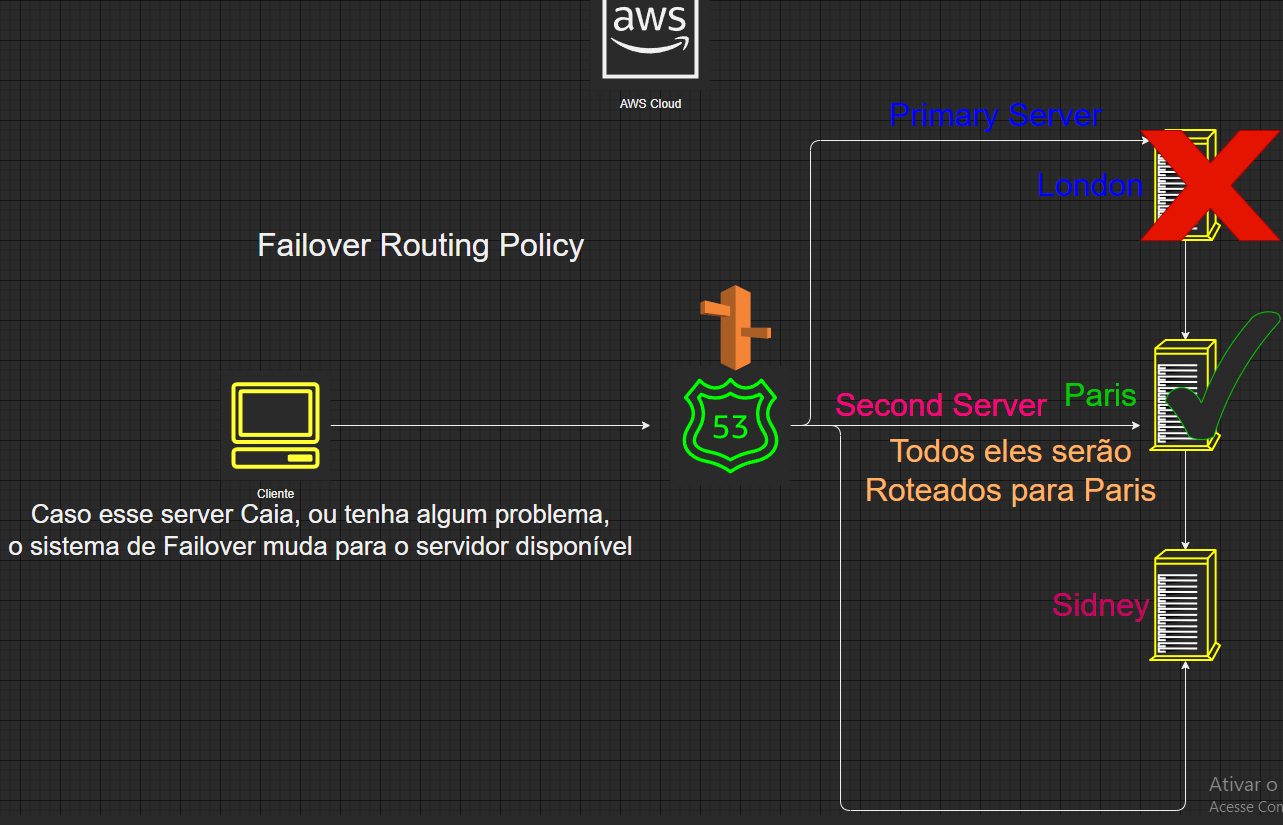
Region: eu-West-2

Set ID: London

Habilitando o Health Check em Londres que já estava criado anteriormente no Laboratório. Ele ira verificar a Menor latência assim como fizemos com o ping, utilizando o endereçamento IP (Para isso precisaremos mudar no Security Group e criar um ICMP 4 e deixar com echo Request

**Failover Routing Policy**

Os usuários que fizerem a requisição: Request para a Route53 para o domínio clubecloudbrazil.com



Create Record Sets no AvaibilityLondon com um TTL de 1 minuto.

**Value: 18.130.106.37** cole o endereçamento IP de London.

**Routing policy: Failover**

**Failover Types:** Primary or Secondary?

**Set ID:** Primary

Create Record Sets no AvaibilityLondon com um TTL de 1 minuto

**Value:** **35.180.42.206**

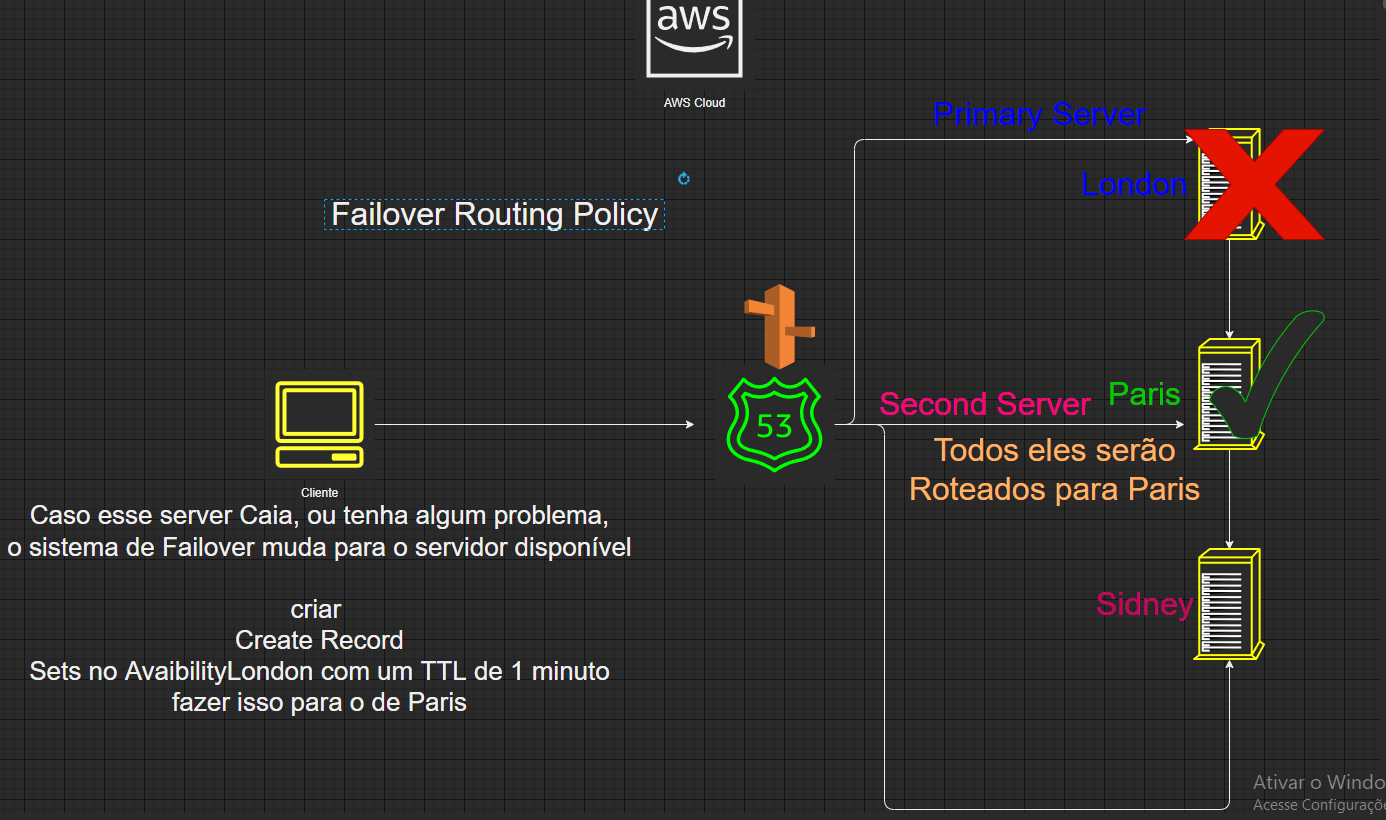
Associate with Heal Check: Yes

Health Check Associate: HC-London

Create novo Records Sets, para Paris, com TTL 60 seconds e Failover Types: Primary or Secondary?

Set ID: Secondary

Heath Check Associate: HC-Paris

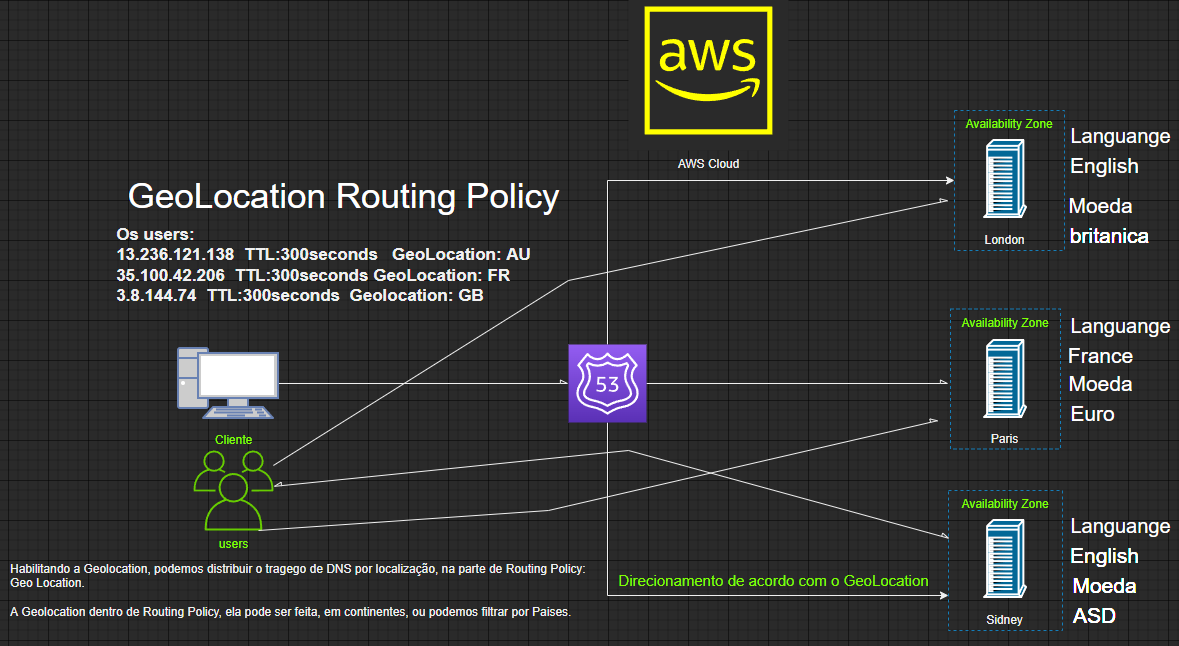


**GeoLocation Routing Policy**

Habilitando o Geolocation, podemos distribuir o trafego de DNS por localização, na parte de Routing policy: GeoLocation

A geoLocation dentro de Routing Policy, ela pode ser feita, em continentes ou podemos filtrar por países.

**Criação de 03 Geo\_location: uma para cada pais, mas funcionando e esse e o cenário que posso fazer de acordo com a nossa área**

**.**

**GeoProximity Routing Policy**

Ele funciona basicamente por Traffic Flow: significa saber exatamente onde o trafego está passando, e podemos adicionar politicas no meio do caminho para esse trafego!

Ele e baseado em localização de GeoLocation por GeoProximity, podemos fazer que usuários de determinada região, sejam roteados para aquela região, e podemos fazer também aplicar um failover, dentro da mesma regra: Então fazemos 02 tipos de Policy, dentro da Mesma regra. Podemos fazer várias Policy.

Criando um Traffic Policy:

Policy Name: Europe

Version:

Version Description:

Ele abre uma janela, possibilitando-me trabalhar com o Flow do meu Trafego, por onde meu tráfego chega, e como eu posso tratar o trafego.

Escolhendo a GeoProximity Role, posso trabalhar inicialmente com duas regiões, sendo que a primeira região eu falo pra ele: eu quero rotear esse trafego para um destino, a partir da regra de proximidade, eu posso passar as coordenadas de latitude e longitude, ou eu posso escolher as coordenadas:

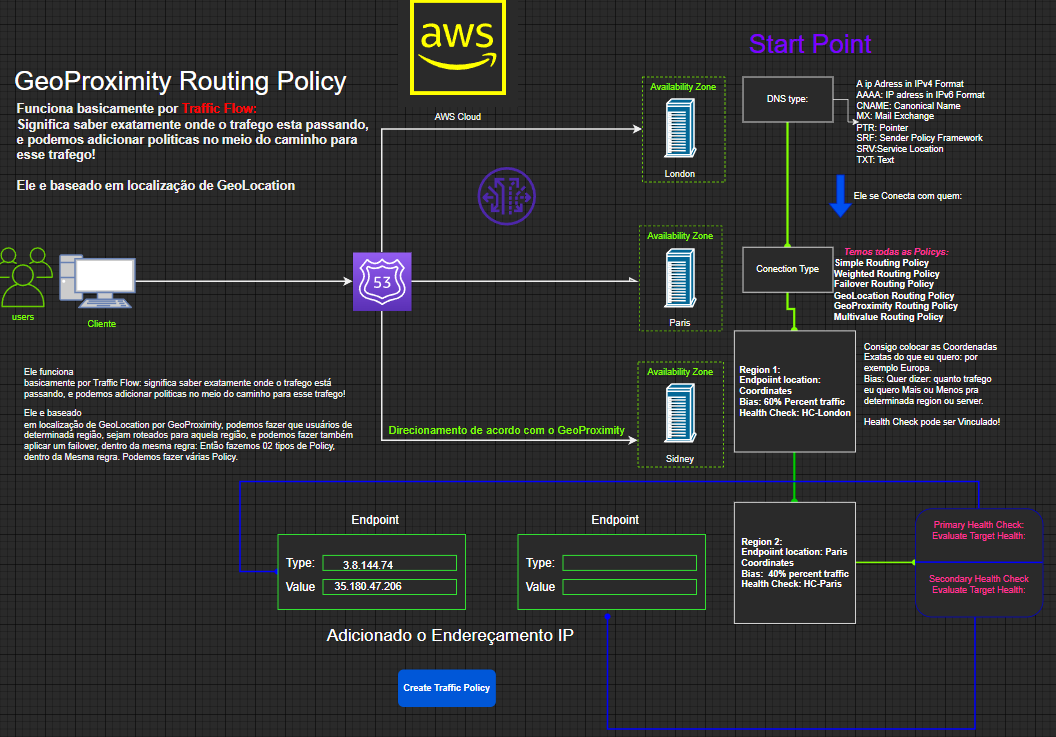
Europe London.

Bias (significa o que: quem bater nesta de London, eu quero mais trafego, saindo, ou menos trafego saindo.

Se o trafego for ligado direto, ao servidor de Londres, ele pode ser ligado diretamente a este servidor:

Pra ficar ainda mais sofisticado: posso ainda adicionar um Failover Roles: dizendo que de acordo (Baseado) com o Health Check eu vou apontar para o endereçamento IP.

Ou baseado no Health Check de Paris eu quero que ele saia a um EndPoint de Paris.



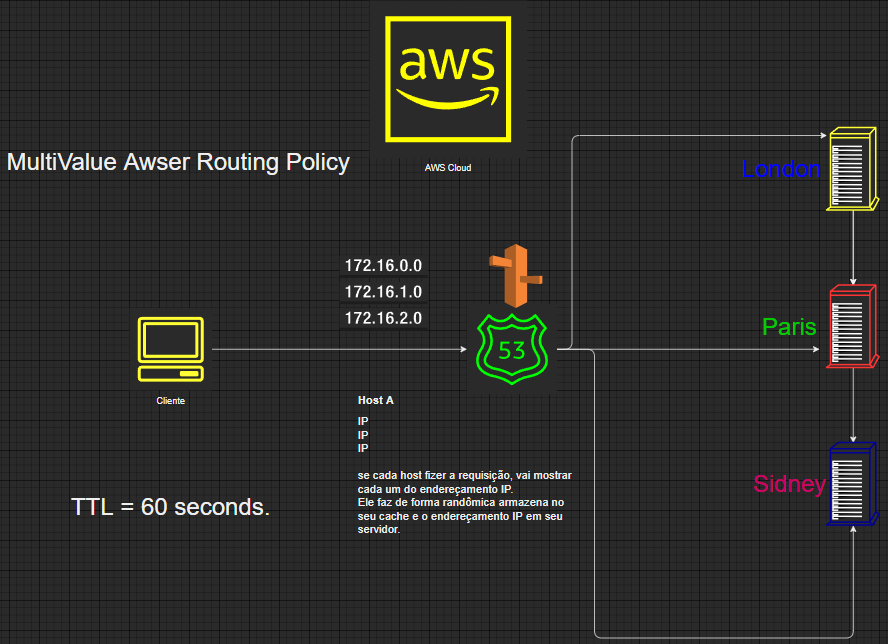
Esta policy tem um Custo de 50$ Dólares por Mês, esse tipo de Policy ela e cobrada.

Remova o tipo de Policy.

**E a habilidade que você tem de visualizar o Traffic Flow e setar regras com Politicas**

**Multivalue Awser Routing Policy**

Entendendo bem o Simple Routing Policy, eu criando os Heath Check consigo setar o Multivalue Awser Routing Policy



**Amazon S3**

**Simple Storage Service**

O Amazon S3 foi criado para armazenar arquivos, tipos de arquivos: MP4, Docs., Jpeg, etc... qualquer tipo de arquivo você pode guardar dentro do Storage S3.

* Existe um Limite máximo de armazenamento de 5 TB. (por arquivo)
* Storage ele é Unlimited: Não tem Limite, você pode armazenar a quantidade de arquivos que você tiver dentro da S3, você pagará por Gigabytes.

Discos dentro da Amazon S3 se chamam **Buckets. (discos, ou cestas) poderá criar suas folders.**

Importante!

Quando você cria uma Bucket, você precisa dar um nome pra essa Bucket, e o nome precisa ser exclusivo mundialmente, quando você cria a sua Bucket isto gera um DNS, e conseguintemente uma URL exclusiva para sua Bucket.

Exemplo de uma Bucket: https://s3-us-west-2.amazonaws.com/nome\_da\_bucket/folder/nome\_do\_arquivo.Doc

Disponibilidade da Amazon:

* **Avalability**: Ela garante 99.9% de Avalability.
* (Integridade) **Durability:** 99.9999999% - Significa, a saúde ou a integridade do seu arquivo.
* Ele funciona com diferentes **Tiers**, significa diferentes preços $$
* **Versioning** (você pode fazer o upload de um arquivo hoje, mudar o arquivo amanha no dia seguinte, no terceiro dia se faz outra alteração do arquivo. (Você consegue armazenar em pastas diferentes, ou em folders diferentes) ele vai colocando uma **TAG**, ou uma **Label** que é uma identificação e vai falando qual arquivo e de qual versão!
* **Encryption:** Todos os seus arquivos você deve habilitar os arquivos, que se tornem 100% seguros, outros deveram decriptar seus arquivos para acessa-los
* **ACL**: Acess Listes, você pode aplicar a um arquivo, se você tem um arquivo de vendas.xls quem pode ter acesso a esse arquivo: somente o grupo ou departamento de vendas, ou somente 04 vendedores do departamento de vendas, você consegue fazer isso através e utilizando as ACLs
* **Bucket Policie:** Por exemplo, você cria um Bucket do depto de vendas: Eu posso aplicar uma policy nessa Bucket, permitindo somente o grupo de vendas, todos os arquivos que estiver lá dentro, eles podem ser acessados pelo pessoal que esta dentro do grupo de vendas V.

**Mais não pode ser associado ao pessoal do Financeiro X**

Nota: Se você quiser criar camadas de segurança, ou camadas de complexidade, mais ela acaba ainda de armazenar arquivos na sua conta AWS!