

Ementa Nova:

<https://docs.google.com/document/d/1C3glpj23ZQ8dPkOlxgvgPkkSOMhe7e5i/edit?usp=sharing&oid=105171219935977454112&rtpof=true&sd=true>

ping está no protocolo: ICMP

Dentro da ferramenta administrativa do ping temos o protocolo ICMP, sendo ele o responsável por mandar uma requisição (Echo Request) para máquina remota e esperar um retorno dessa máquina remota (Echo Reply).

IaaS = você toma conta do sistema operacional, VM

PaaS = você não toma conta do sistema operacional, app service, heroku

FaaS = Uma única função que roda no serviço http

SaaS = um serviço pronto que você utiliza, dinamo db

Tipo de pagamento azure

<https://k21academy.com/microsoft-azure/az-900/az-900-azure-subscriptions/>

Tipos de suporte Azure

<https://azure.microsoft.com/en-us/support/plans/>

Tabela de preços Azure

<https://azure.microsoft.com/pt-br/pricing/calculator/>

Azure auto scale set

<https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-autoscale-overview>

Container instances azure

<https://azure.microsoft.com/pt-br/services/container-instances/>

Azure cdn = akamai

<https://docs.microsoft.com/pt-br/azure/cdn/cdn-overview>

Python conect sql server

<https://stackoverflow.com/questions/33725862/connecting-to-microsoft-sql-server-using-python/33787509>

Azure storage

- Blob = um container de arquivos (binário large objects) ou seja para armazenamento de arquivos grandes
- Disk = um disco a mais para sua vm
- File = serviço de file share, um serviço de compartilhamento de arquivos.

Exemplo S3

- Archive = uma classe de armazenamento, para preparar para ser utilizado

lsblk = lista os dispositivos de blocos no Linux (vc verifica os hds)

sudo fdisk /dev/meudisco para montar o disco

Instalar o sistema de arquivos no HD

sudo mkfs.xfs /dev/meudisco

Levantar fake VPS

python -m SimpleHTTPServer 80

Para fazer ddos

hey -z 2m -c 20 <http://site.com>

ssh -i ~/Desktop/devops/treinamentoltau

ubuntu@ec2-3-91-6-191.compute-1.amazonaws.com -o ServerAliveInterval=60

scp -r -i "tti-key.pem" ../treinamentos

[ubuntu@ec2-54-162-1-250.compute-1.amazonaws.com:/home/ubuntu](https://ec2-54-162-1-250.compute-1.amazonaws.com:/home/ubuntu)

Desafio:

https://docs.google.com/document/d/1kvTHM21AhvCi_px2MtkGSZDRNzE65irRfPngLsIX6nc/edit

kubectl run wes-nginx --image=nginx --port 80 --dry-run=client -o yaml >

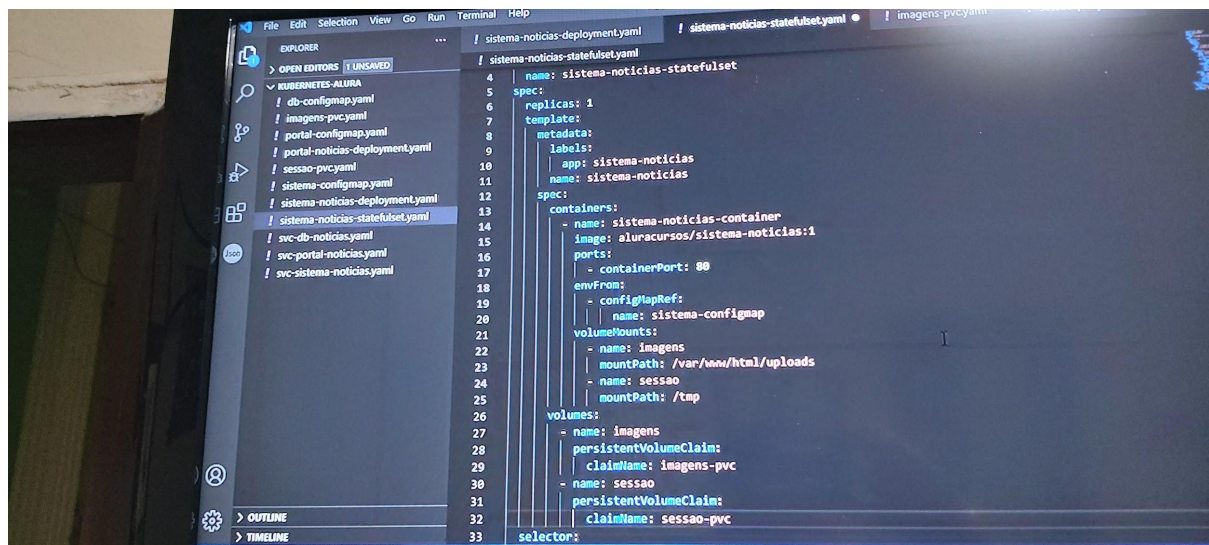
pod-nginx.yml

```
# sudo su -
# hostname k8s-master-0
# echo "k8s-master-0" > /etc/hostname
# bash
# hostnamectl set-hostname ec2-turma2
# bash
```

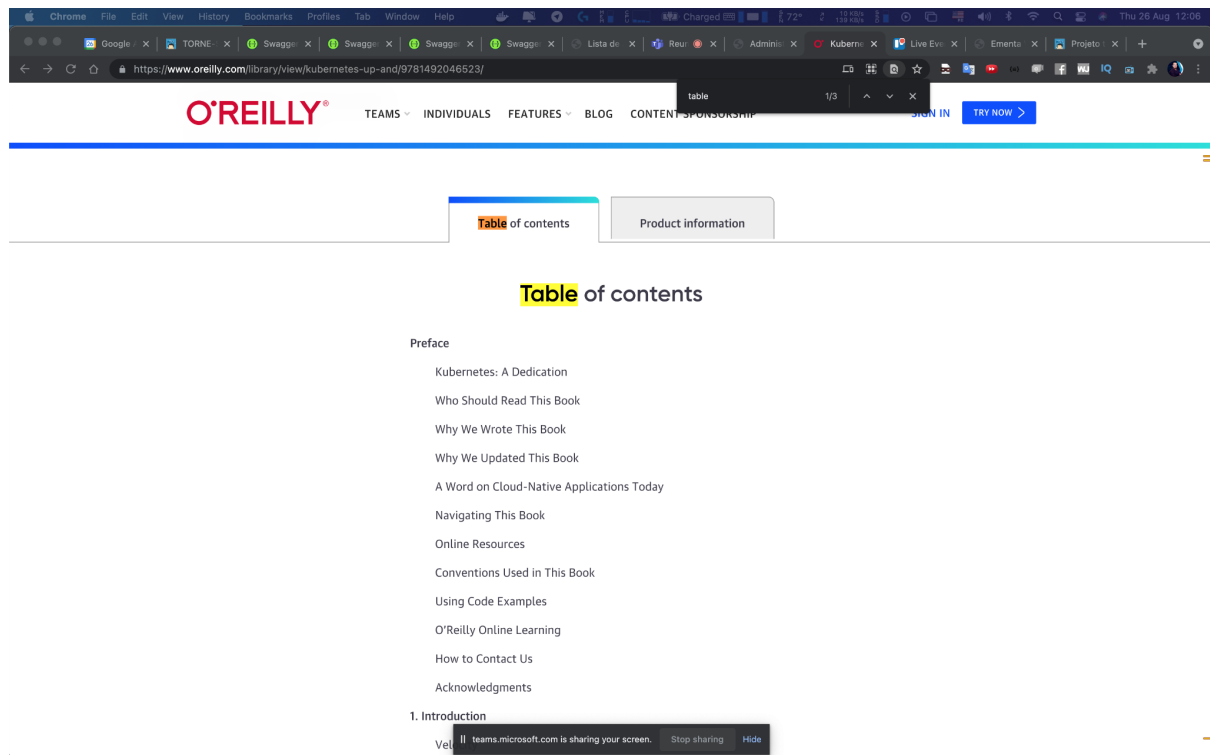
os cubos mostram posições não marcadas

Kubernetes

- Persistente volume - opção para criar um volume com a quantidade de GB
- Volume claim - forma de ligar o pod ao Persistente volume
- Storage class - opção para criar volume dinamicamente, uma espécie de classe para utilizar no persistente volume
- Statefulset - uma forma de criar pods com volume tudo no mesmo arquivo, parecido com do deployment para o replicaset, ou seja vc cria tudo por ele, volume e claim



<https://www.oreilly.com/library/view/kubernetes-up-and/9781492046523/>



ItauDOT - Ementa anotações -

<https://docs.google.com/document/d/1C3glpj23ZQ8dPkOlxgvgPkkSOMhe7e5i/edit>

Ementa final

<https://drive.google.com/file/d/1y0pflHs8bN-XG6GCdhq5-SPIFATVnmTX/view?usp=sharing>

Ementa final - Gama

https://drive.google.com/drive/folders/15p-6n_d39QD3OwTMweX6b661SvXQDvJJ

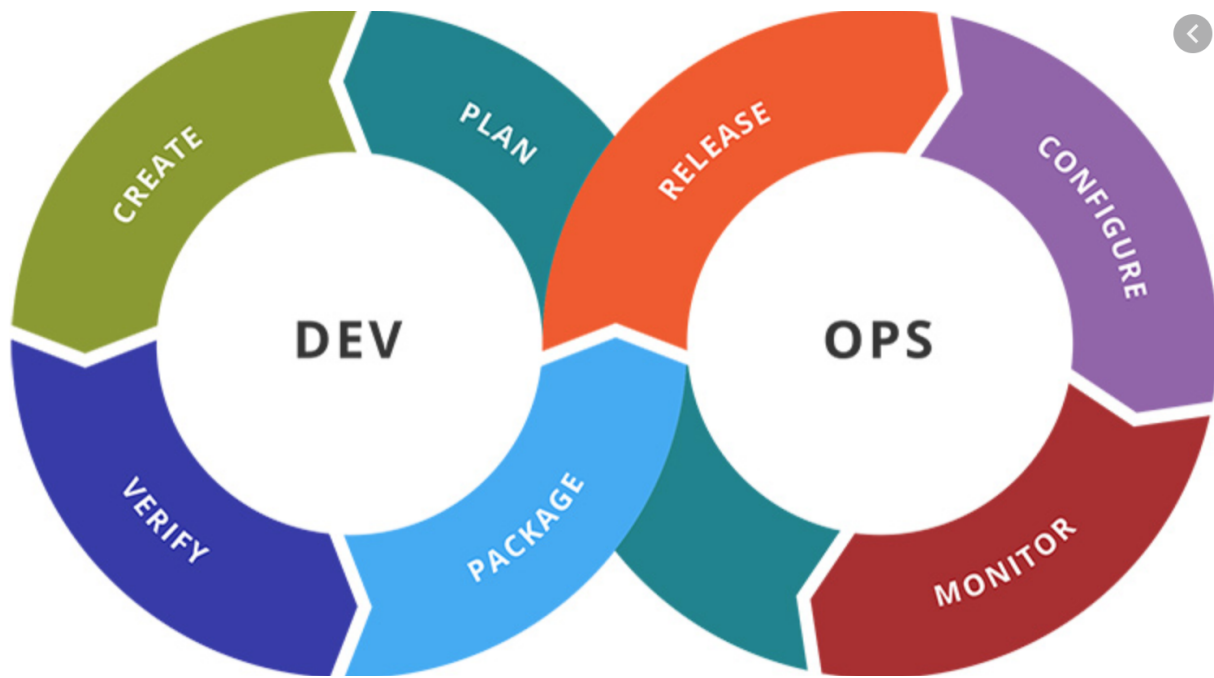
Daniel Gestor de infra telefone itaú - 11 97555-9193

Um pouco mais de material - apostila guia para treinamento que dei para concrete

<https://docs.google.com/document/d/19VBRm614oLk0fXceVz3kBXRVeDlCYWwPV6tNPKFH5Zc/edit>

Apresentação Terraform

https://docs.google.com/presentation/d/1h81actx7ReWqXyzGhYJRuH_byCxALVL6MbLUvx xmmU/edit#slide=id.gf113ee1e2f_0_335



C – Culture
A – Automation
L – Lean
M – Measurement
S – Sharing

<https://blog.estabil.is/calms-conceito-devops/>

<https://itrevolution.com/the-three-ways-principles-underpinning-devops/>

Three ways

1. Pensar na regra de negócio, não ser somente um digitador de códigos ser também uma pessoa que entende dos processos, pensar no negócio no geral, custos de infra, redução de valores e efetividade (esquerda para direita)
2. Dar mais feedbacks(pessoas e técnicos), errar pequeno ou antes, pegar os erros na automação antes de produção, não é errado errar, desde que peguemos antes.

3. Experimentação e aprendizagem contínua, saber que tecnologias mudam e que precisamos nos atualizar, aprender sempre. Assumir riscos. Errar faz parte, temos que aprender a trabalhar com erros e não jogar a culpa nos outros.

Livros

Projeto fênix

https://www.amazon.com.br/projeto-f%C3%AAnix-comemorativa-romance-neg%C3%B3cio/dp/8550814067/ref=asc_df_8550814067/?tag=googleshopp06-20&linkCode=df0&hvadid=379728732654&hvpos=&hvnetw=g&hvrnd=3072114101017601912&hvpone=&hvptwo=&hvqmt=&hvdev=m&hvdvcmdl=&hvlocint=&hvlocphy=1001773&hvtargid=pla-1168144096722&psc=1

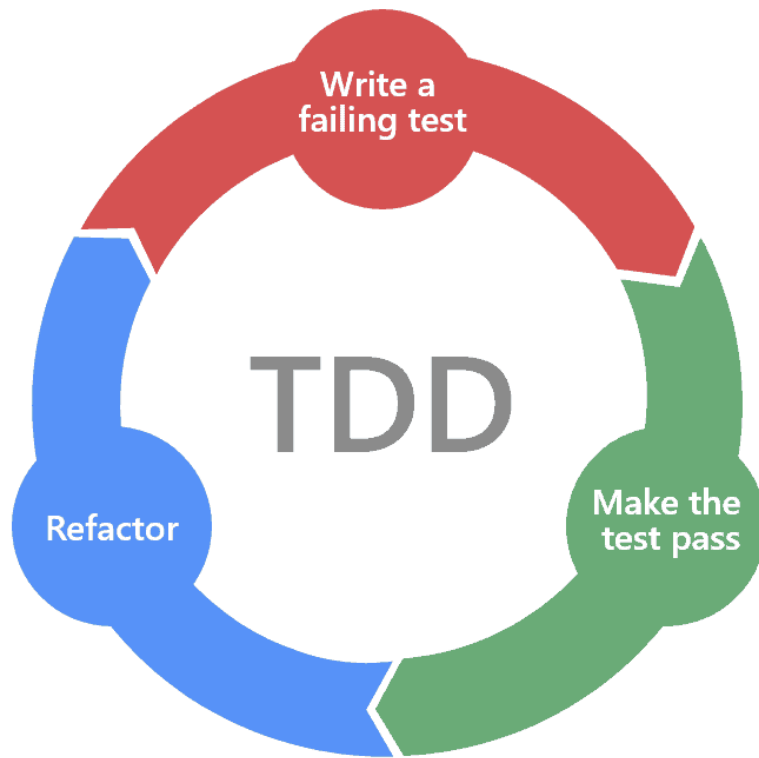
Manual devops

https://www.amazon.com.br/Agilidade-Confabilidade-Seguran%C3%A7a-Organiza%C3%A7%C3%B5es-Tecnol%C3%B3gicas/dp/8550802697/ref=asc_df_8550802697/?tag=googleshopp06-20&linkCode=df0&hvadid=379726090221&hvpos=&hvnetw=g&hvrnd=10351433782871869681&hvpone=&hvptwo=&hvqmt=&hvdev=m&hvdvcmdl=&hvlocint=&hvlocphy=1031606&hvtargid=pla-809606891693&psc=1

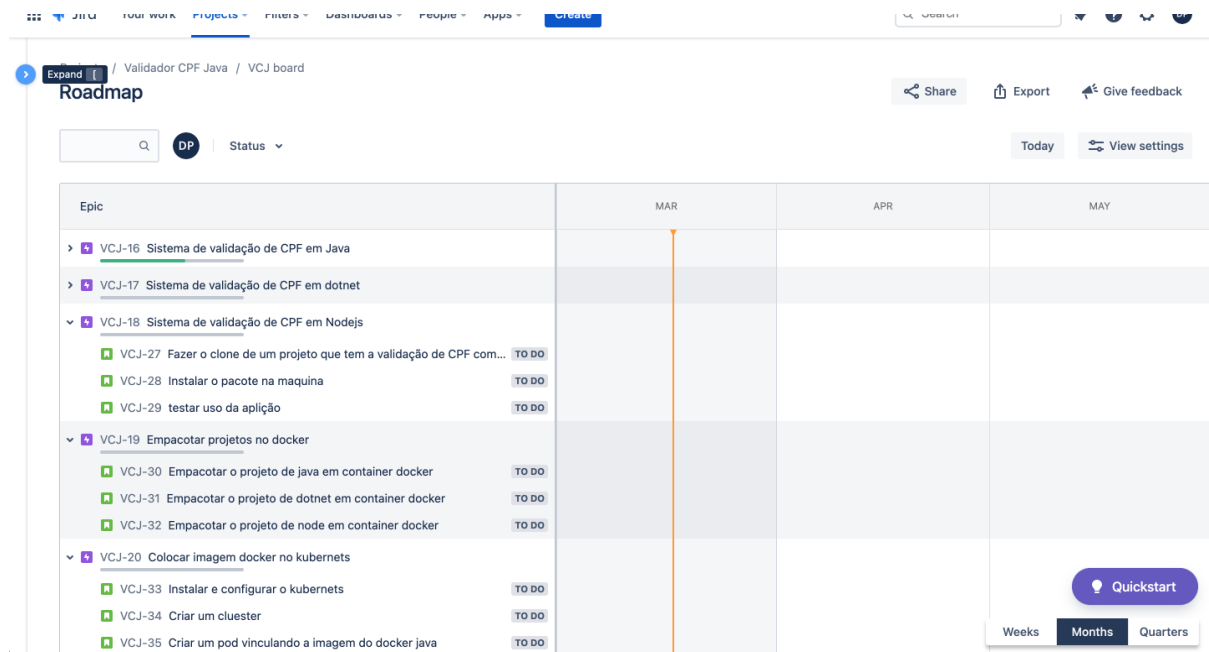
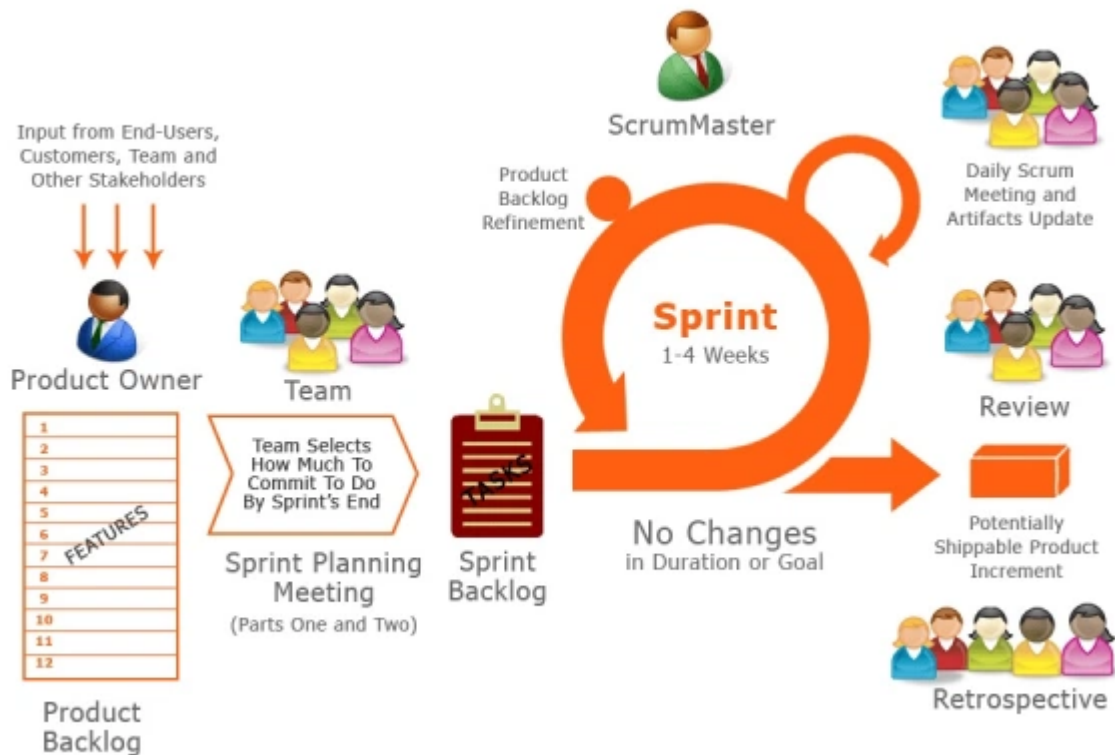
Valar sobre Versioning 0.0.1 (Major.Minor.Patch)

https://en.m.wikipedia.org/wiki/Software_versioning

TDD



https://pt.wikipedia.org/wiki/Test-driven_development



Épico - é uma tarefa macro, "Um desejo"

Histórias - tarefas sem detalhamento do que precisa ser feito

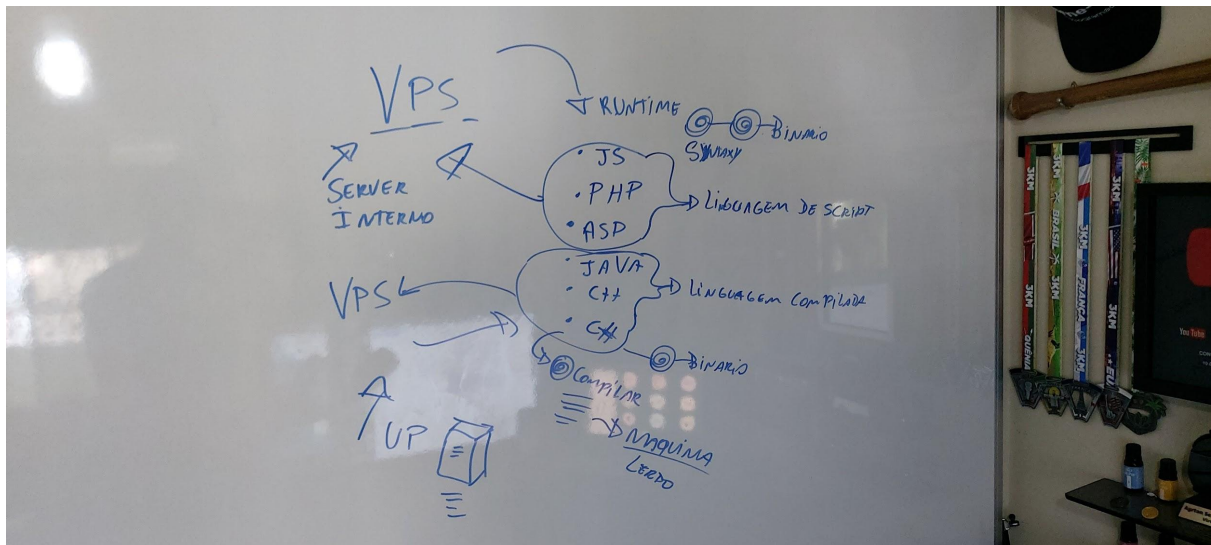
Bug - Problema no sistema que precisa ser corrigido

Task - tarefa que faz o detalhe de uma história

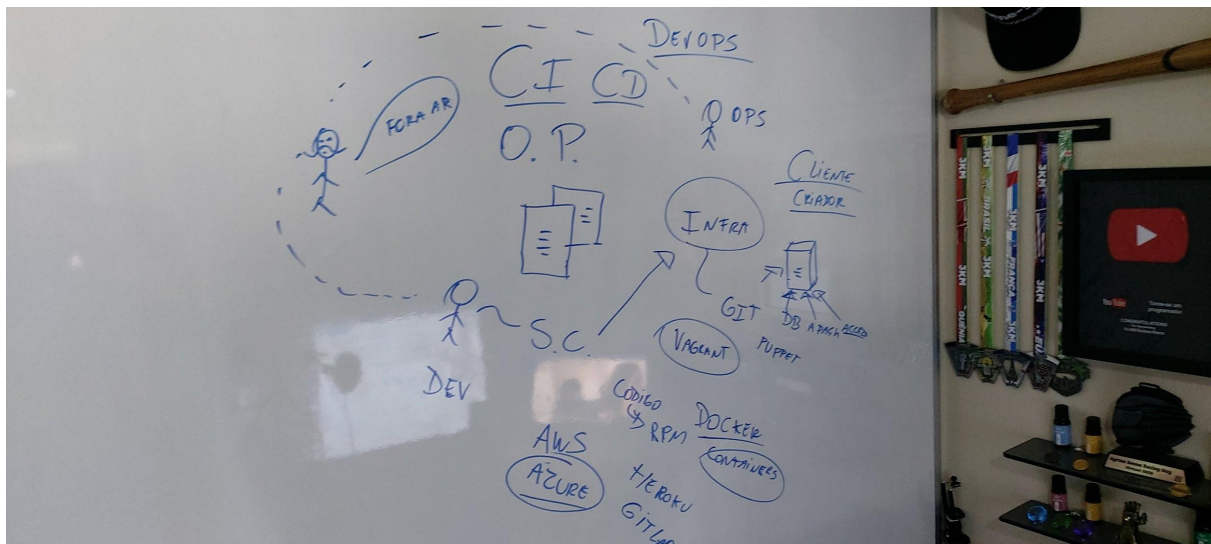
Backlog - todas as tarefas, bugs e histórias listadas em um banco de informações

Sprint - Ciclo com meta de entrega das tarefas

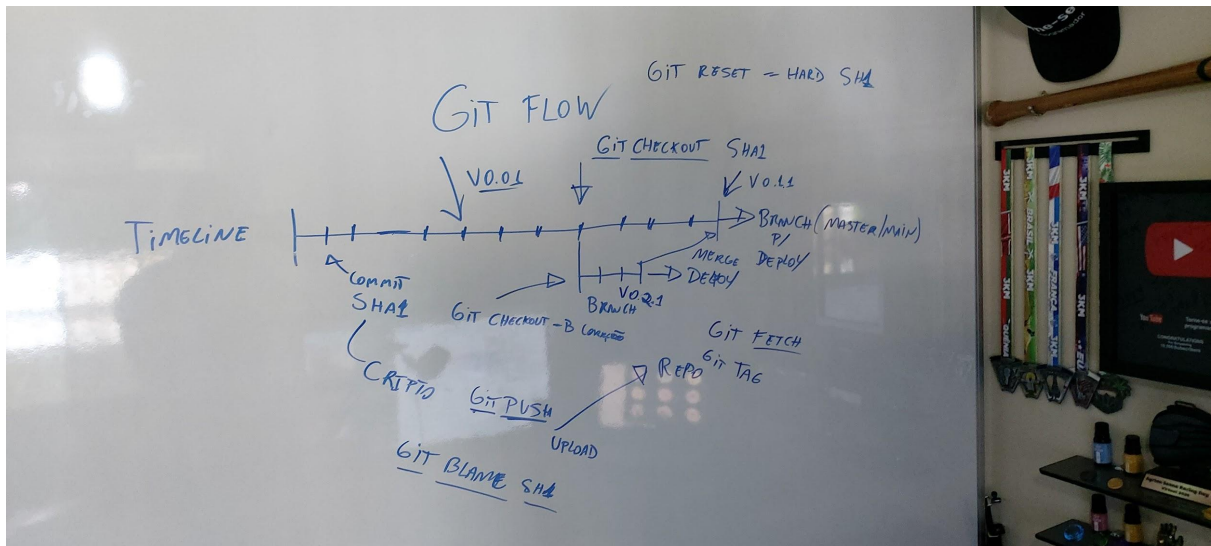
Estrutura de linguagens



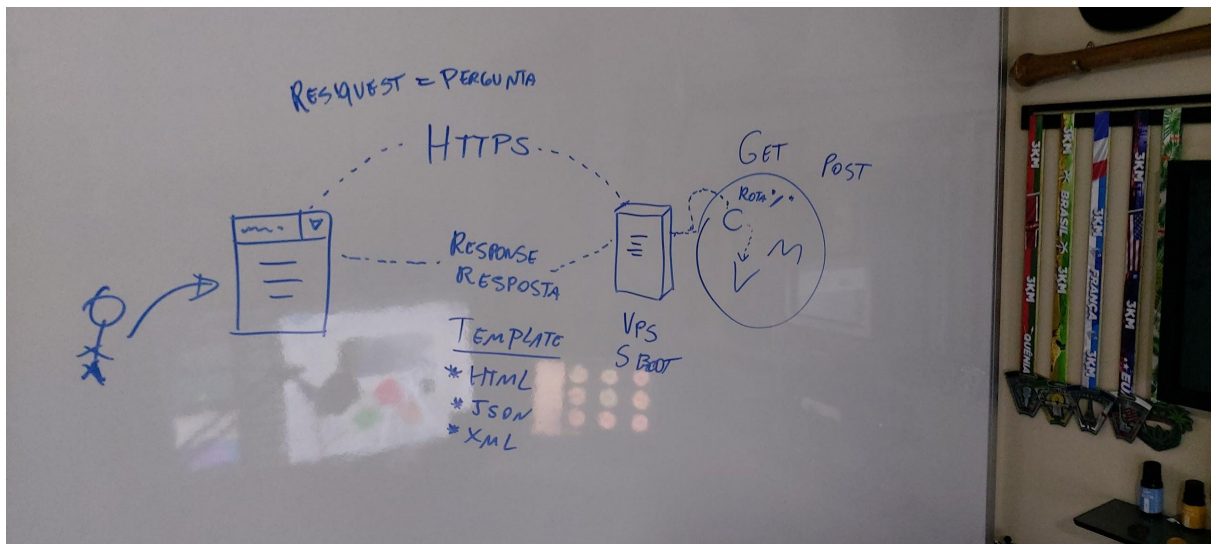
Cultura devops e como iniciou



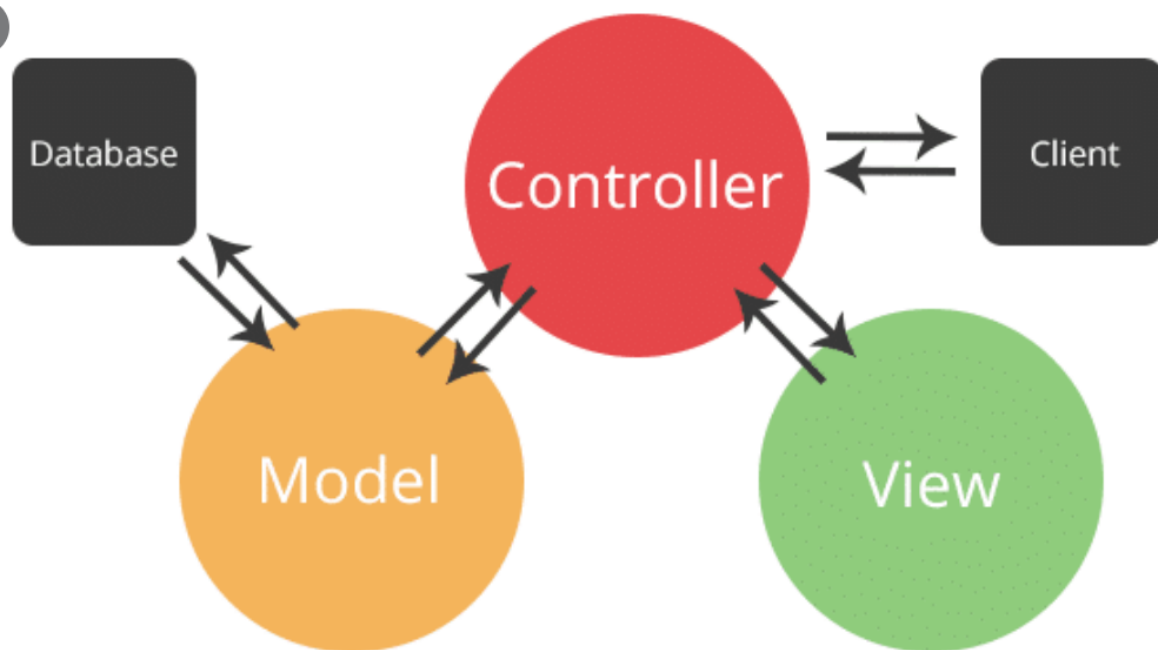
Git flow



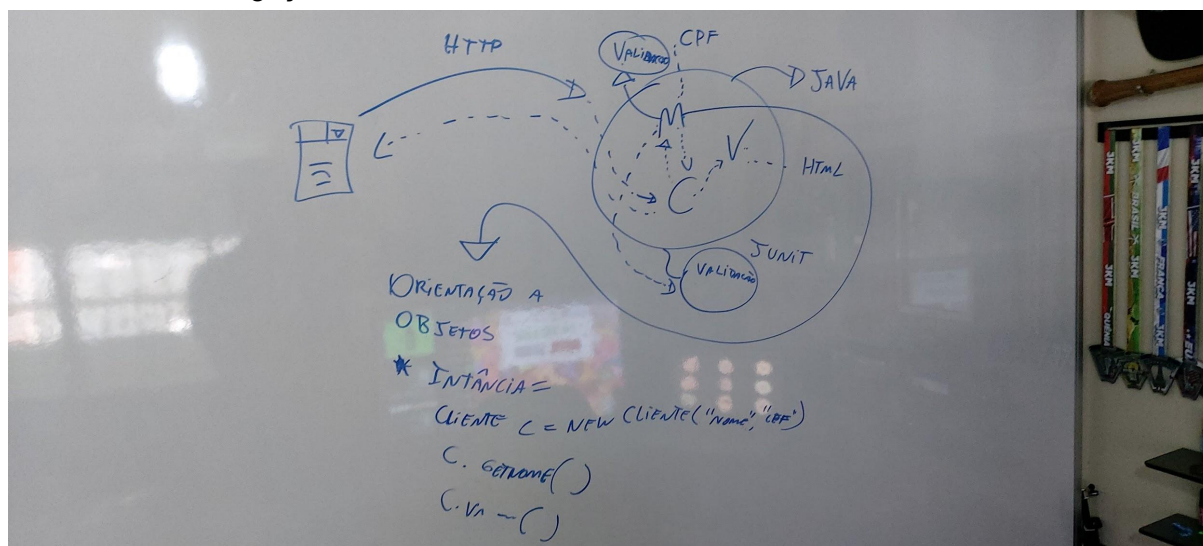
Http flow



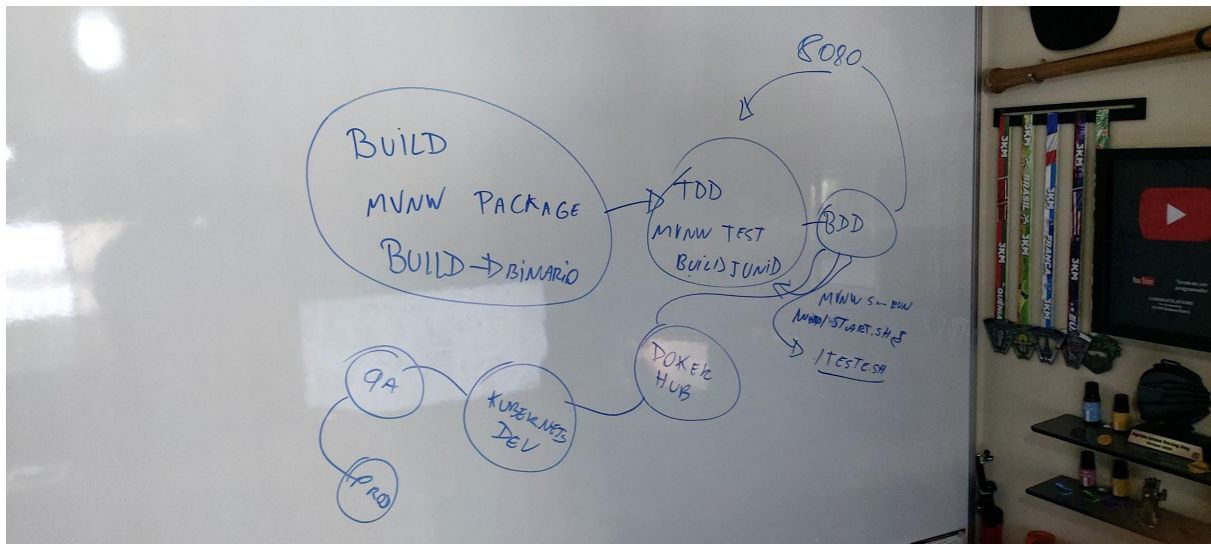
fluxo MVC



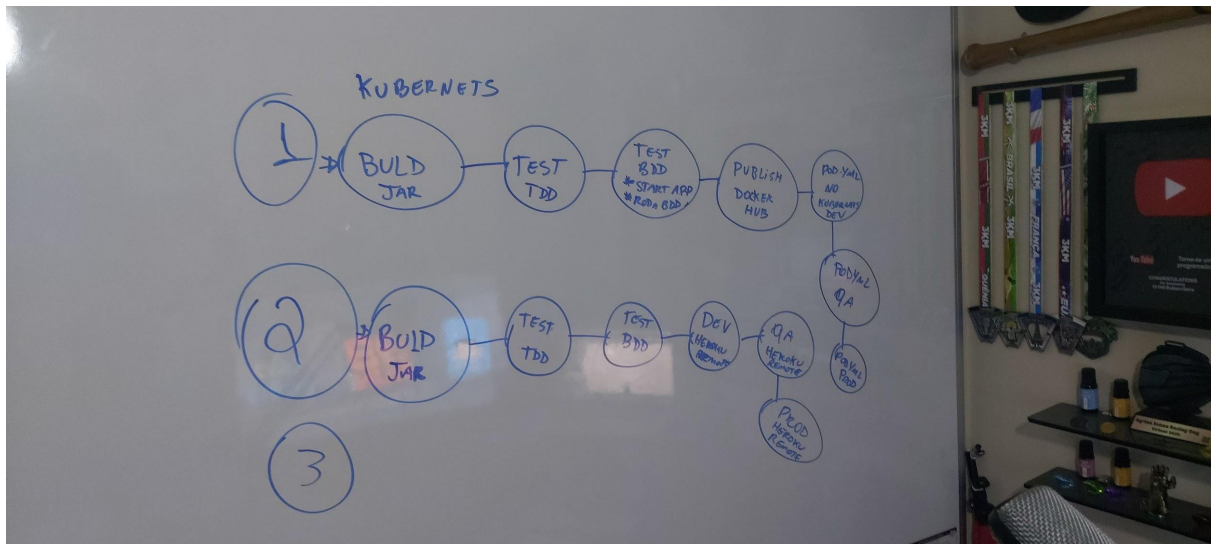
Fluxo controller e ligação com modelo



Estrutura de cache



Estratégia de pipeline kuberbets ou divers como heroku, azure, aws

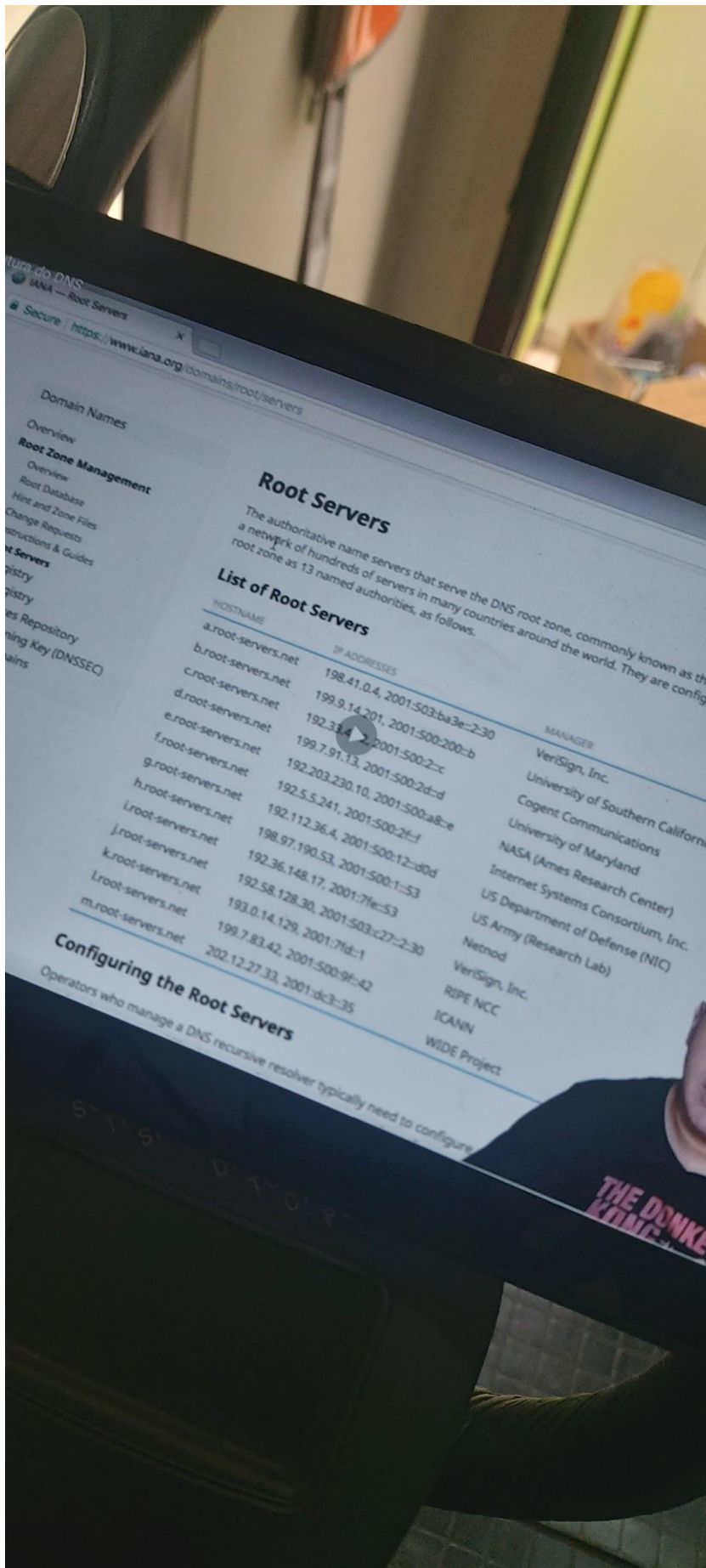


DNS

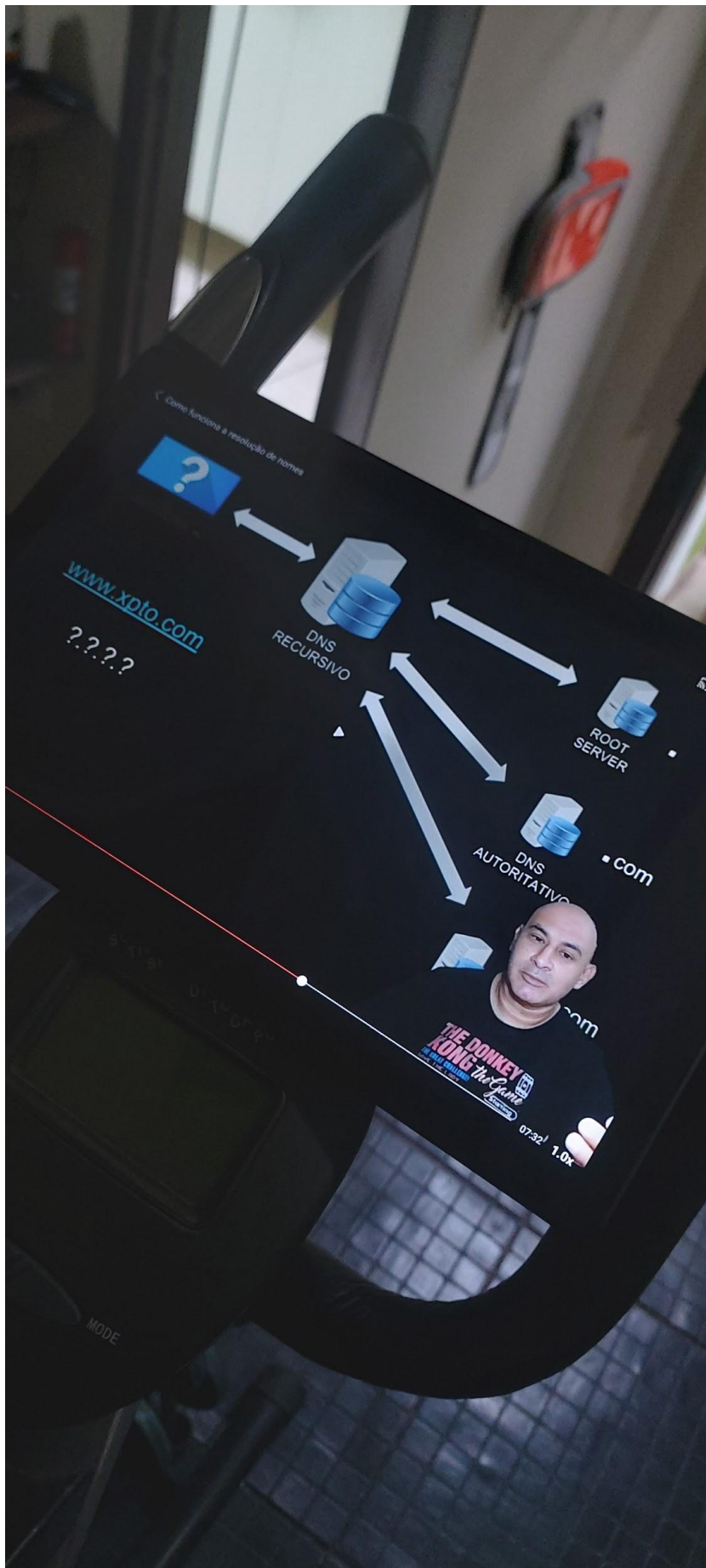
Responsáveis por abrir DNS

Digitar no Google "root servers"

www.iana.org



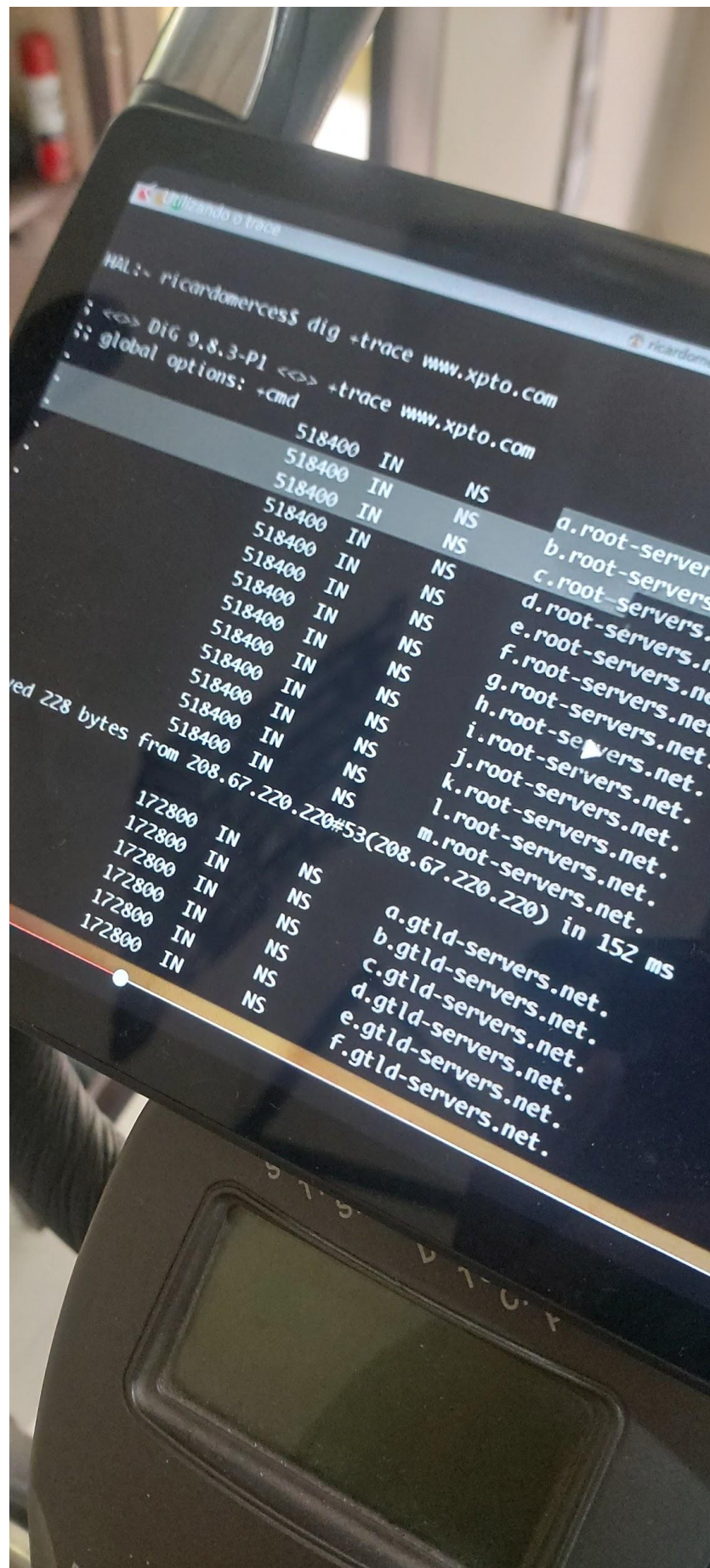
Estrutura DNS



host www.google.com

dig www.google.com

dig + trace www.google.com



DNS reverso = serve para traçar um IP através de sua rede

tracert 198.976.9.00 -- mostra os dados do DNS reverso ou seja do IP ao nome

```
DNSReverso
:: QUESTION SECTION:
www.google.com.
:: ANSWER SECTION:
www.google.com.
:: Query time: 4 msec
:: SERVER: 192.168.1.4#53(192.168.1.4)
:: WHEN: Wed Feb 21 09:15:29 2018
:: MSG SIZE rcvd: 48

241      IN      A      172.217.
ricardomercers traceroute 172.217.29.68, 64 hops max, 52 byte packet
aceroute to 172.217.29.68 (172.217.29.68)
  0112102.virtua.com.br (201.17.33.2) 12.256 ms 13.510 ms 11.024 ms
  110046.virtua.com.br (201.17.0.70) 18.391 ms 14.252 ms 14.381 ms
  292.virtua.com.br (201.17.34.146) 12.941 ms 16.106 ms 10.498 ms
  10046.virtua.com.br (201.17.31.202) 11.774 ms 11.166 ms 15.256 ms
  251.65 (108.170.251.65) 9.533 ms 11.772 ms
  1.81 (108.170.251.81) 11.502 ms
  5 (108.170.228.5) 13.346 ms
  5 (108.170.228.7) 11.716 ms
```

dig -x 186.228.33.22
Pega reverso também

```

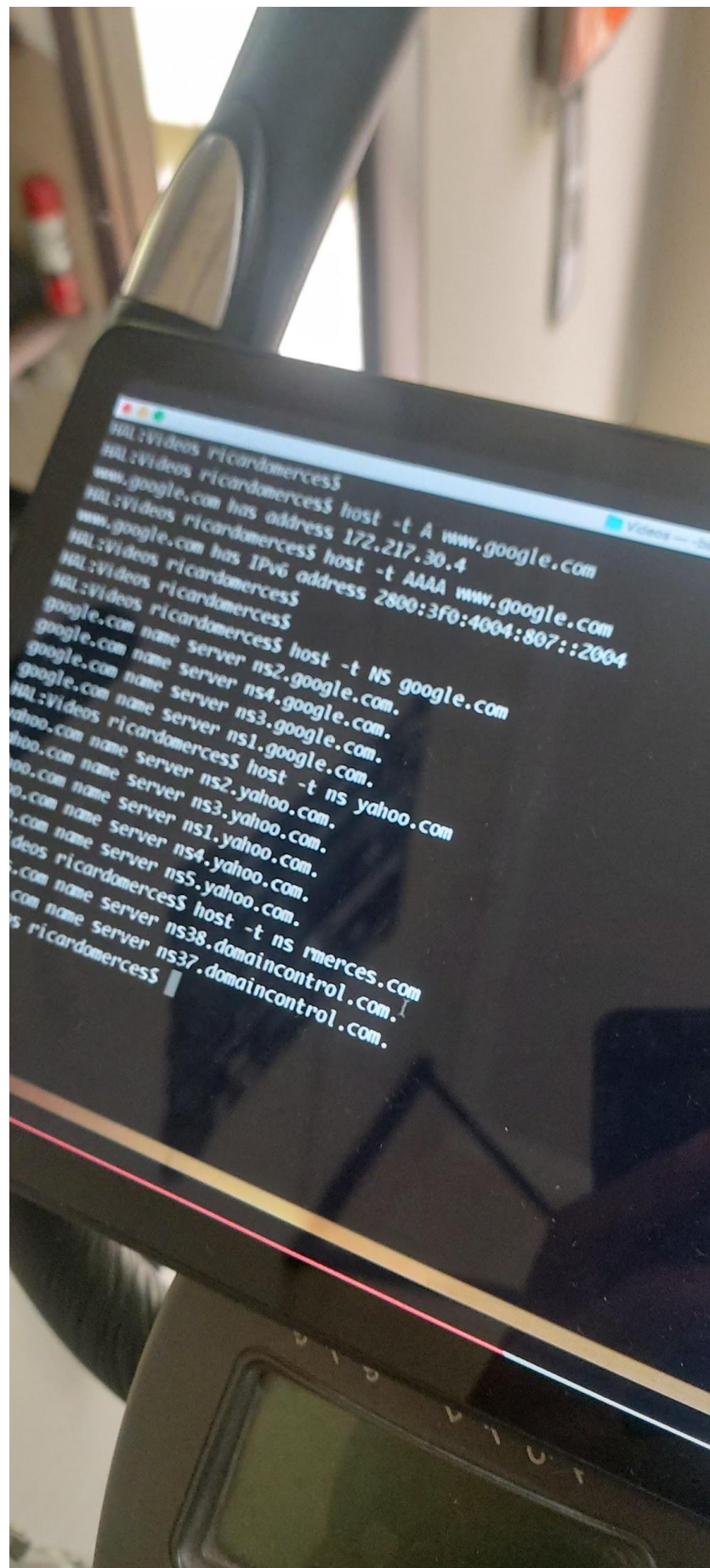
root@kali:~# dig -x 177.142.6.236
;; global options: +cmd
;; HEADER: opcode: QUERY, status: NOERROR, id: 44874
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
; 177.142.6.236.in-addr.arpa. IN PTR

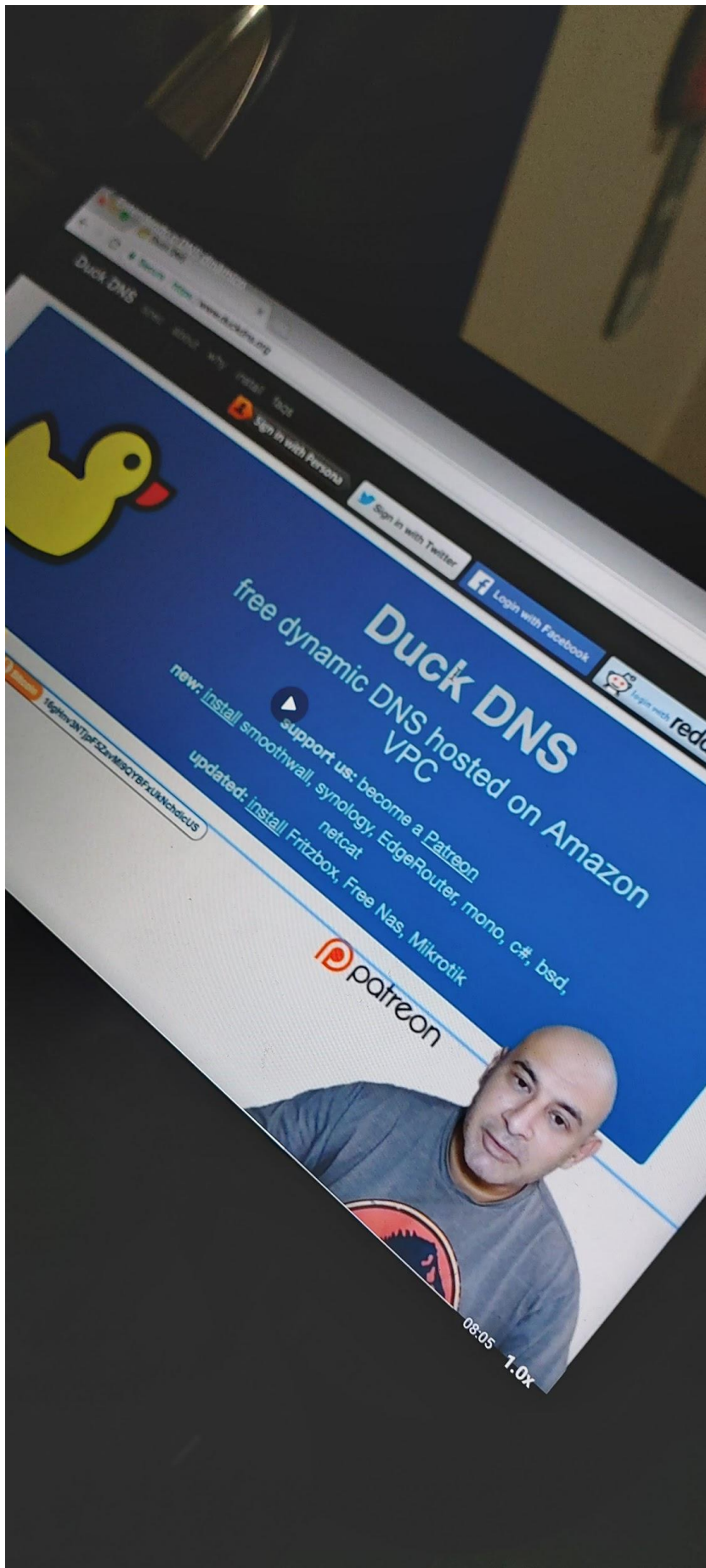
;; ANSWER SECTION:
177.142.6.236.in-addr.arpa. 3600 IN PTR
b18e06ec.virtua.com.br.

```

Filtrando tipos de registros



Definir DNS dinâmico tipo A por IP usar o Duck DNS do Google

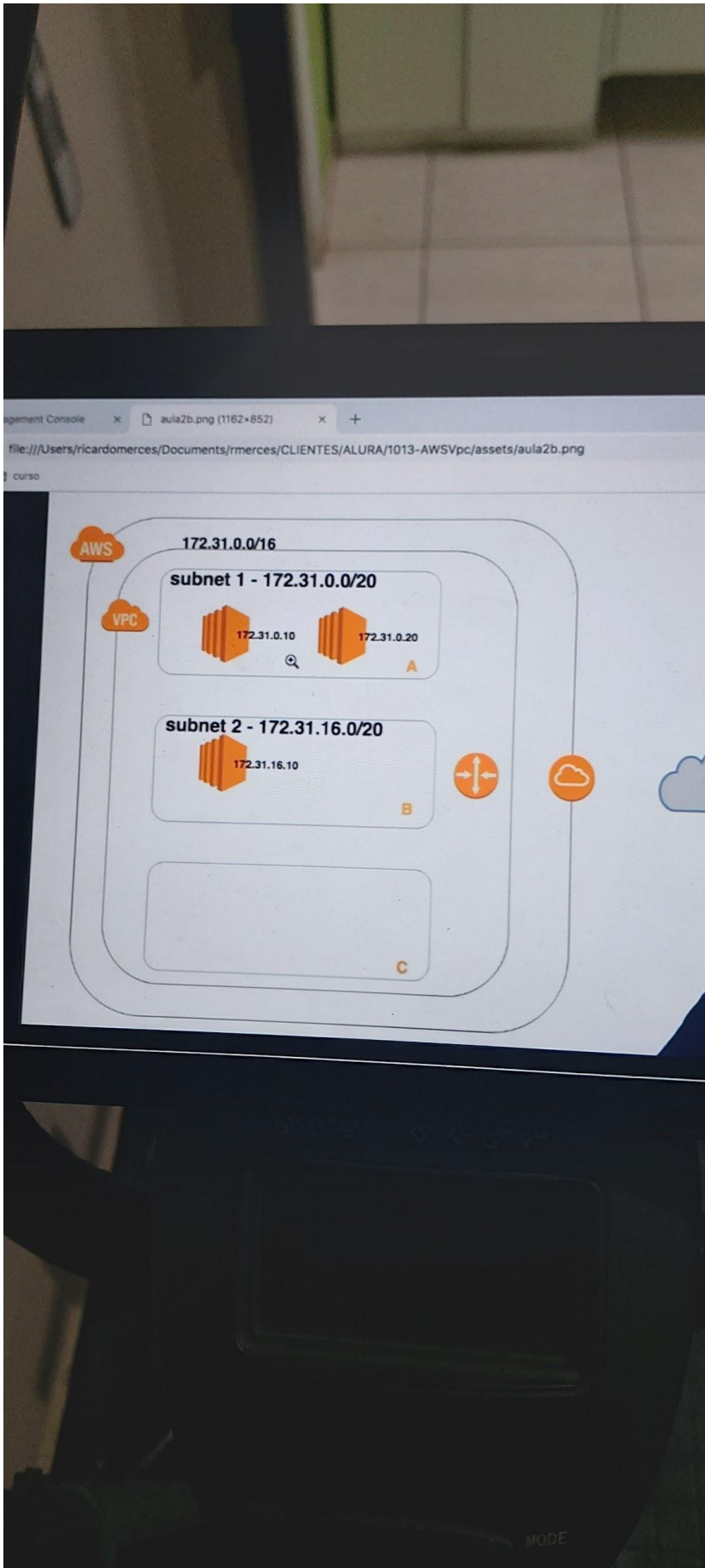


Instalar dentro do servidor uma cron para rodar e atualizar com o IP o duck dns
Ele cria um cname que atualiza o IP automático pela crontab

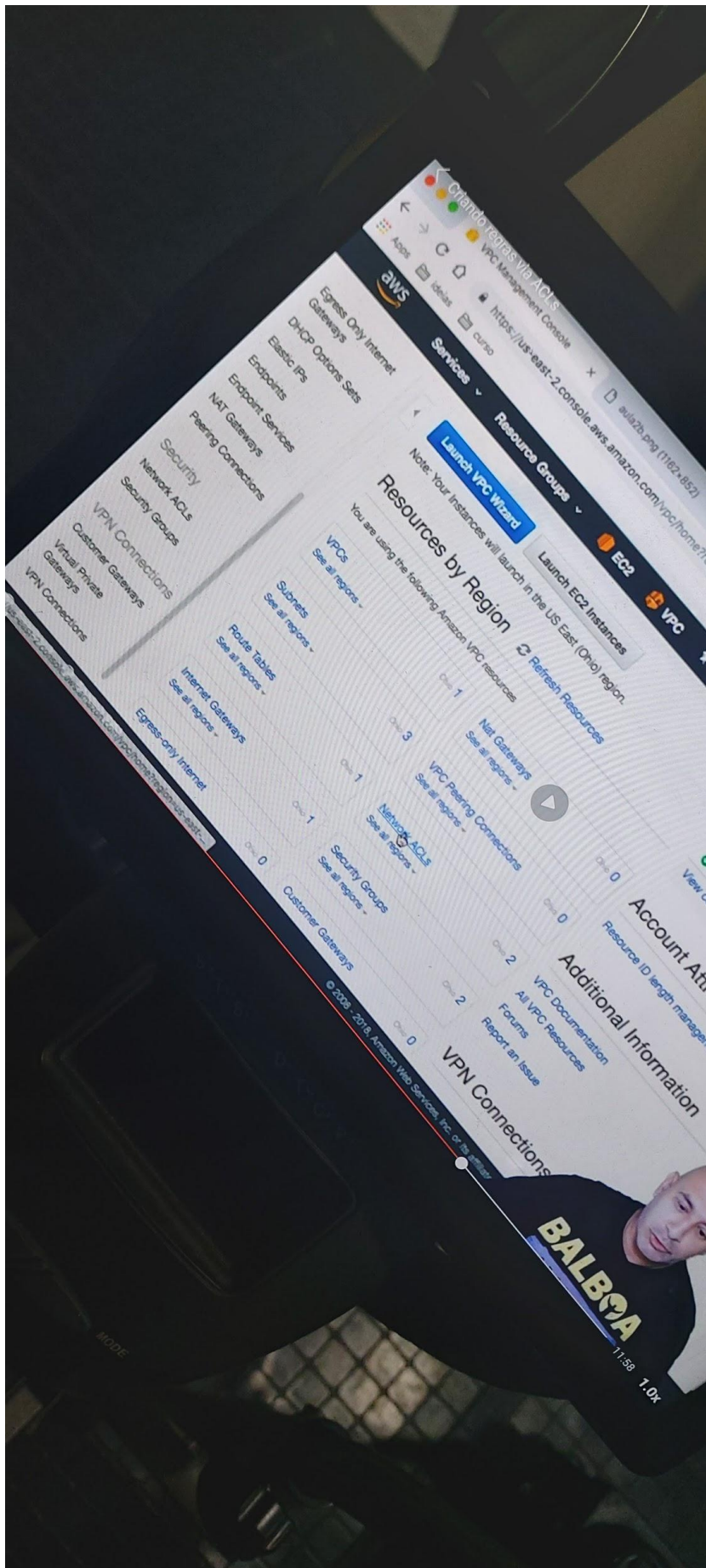
VPC

Subnet = range de IPs cada vpc vem por padrão 3 Subnets, cada subnet fica disponível em regiões diferentes para uma maior escalabilidade

Ao criar um EC2 você pode escolher uma região ou deixar automática



ACL - uma forma de limitar os acessos das subnets



Como Criar regra ACL

- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints
- Endpoint Services
- NAT Gateways
- Peering Connections
- Security
- Network ACLs
- Security Groups
- VPN Connections
- Customer Gateways
- Virtual Private Gateways
- VPN Connections

Create Network ACL

Delete

Search Network ACLs and the X

Name	Network ACL ID	Associated With	Default	VPC
db-accesso	acl-017ba06cafa1f8dd	0 Subnets	No	vpc-f9f13191

acl-017ba06cafa1f8dd | db-accesso

Summary

Inbound Rules

Outbound Rules

Subnet Associations

Tags

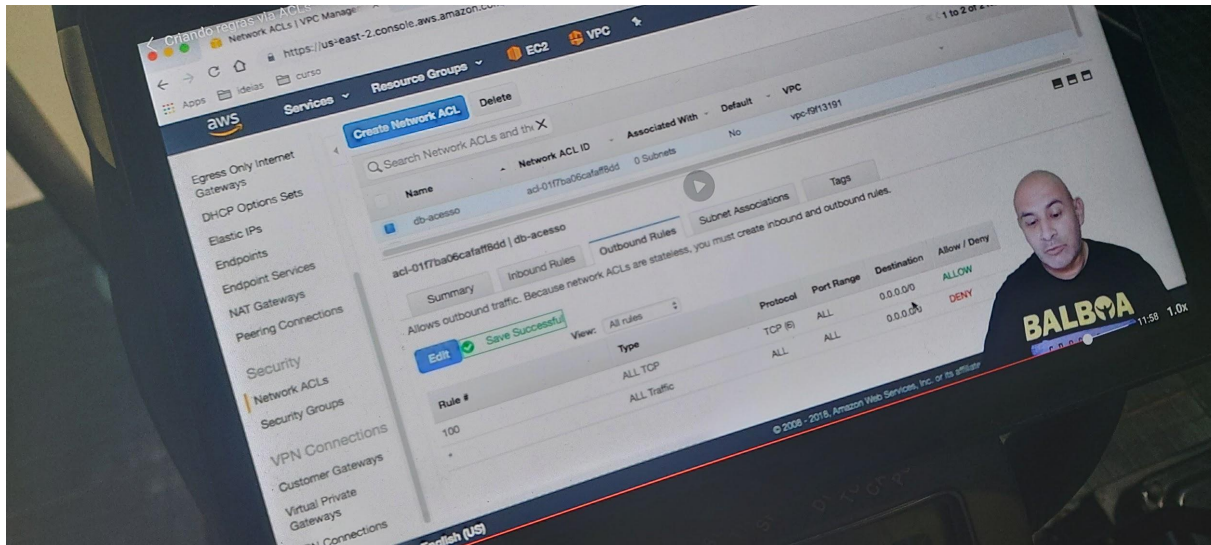
Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Edit

View: All rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	MySQL/Aurora (3306)	TCP (6)	3306	172.31.0.0/20	ALLOW
	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

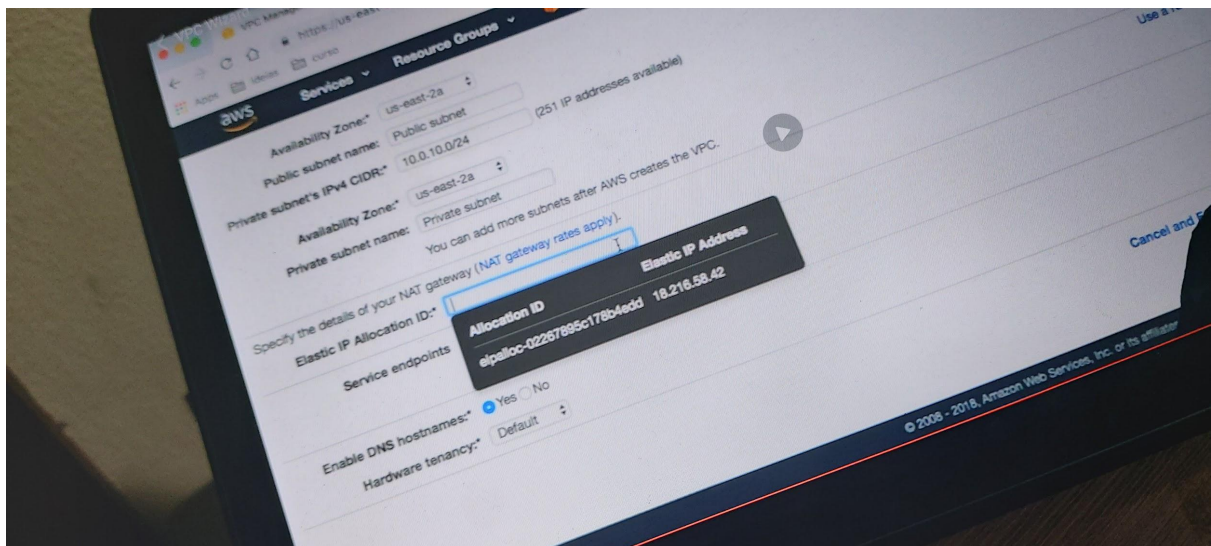
Para a ACL precisa ter o caminho de ida e volta

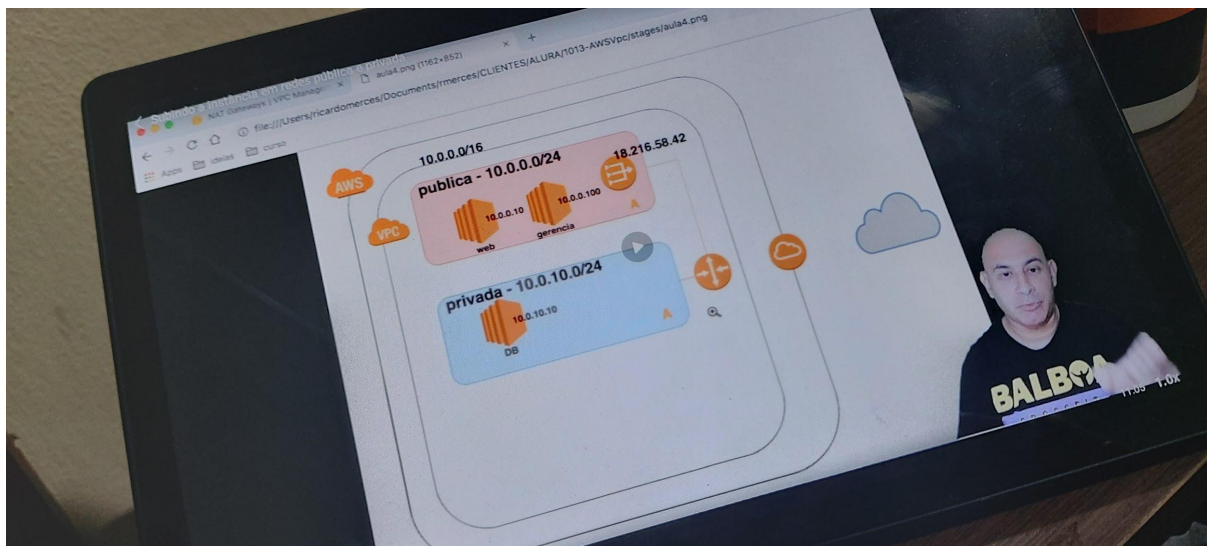


Usamos ACL quando queremos limitar a rede da Internet e deixar somente acessar das redes internas, ou seja somente das subnets

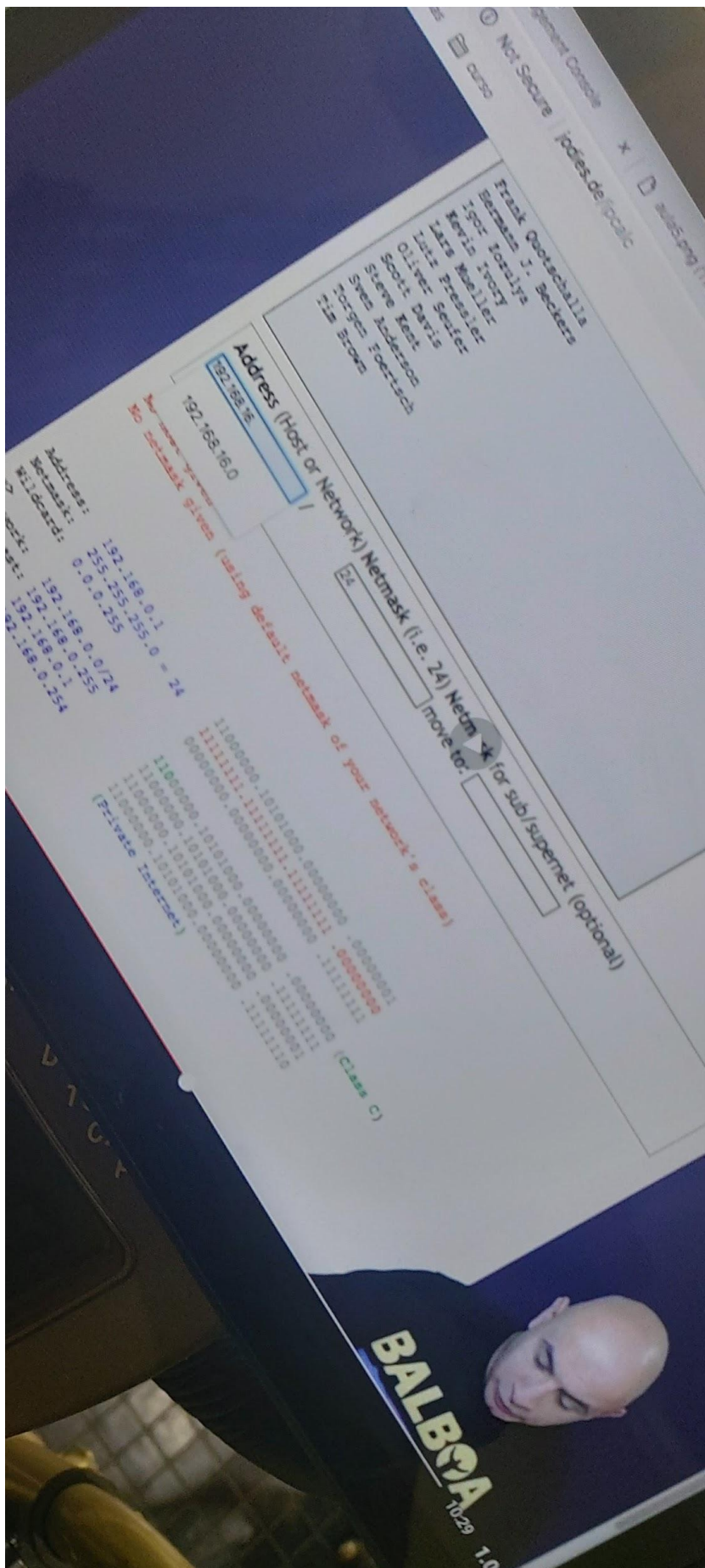
NAT Gateway = um roteador que só vai permitir o acesso privado de sua rede para Internet. Pois uma rede privada sem nat gateway não tem acesso a Internet

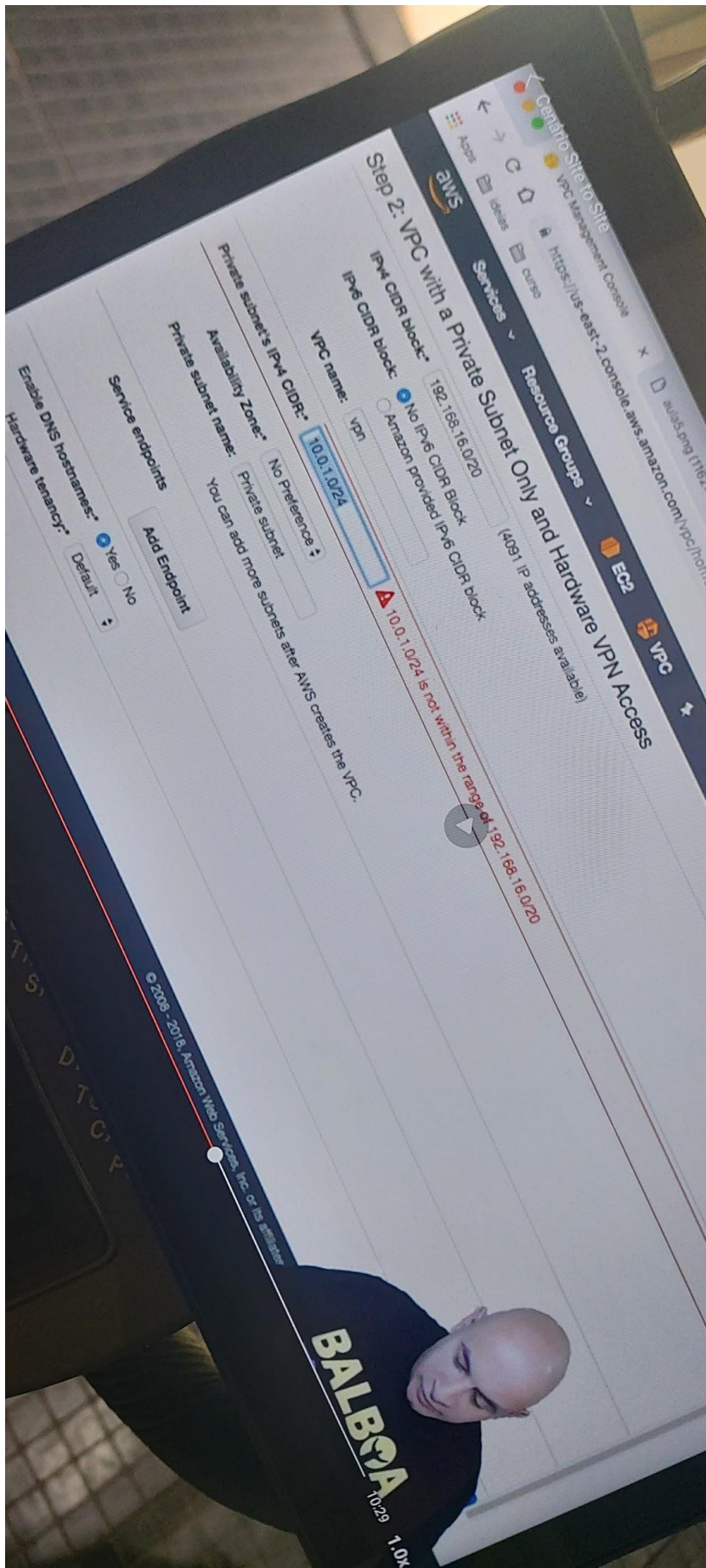
Ao criar uma VPC só consigo selecionar o nat gateway através de um elastic IP





Ipcalc um site que calcula as redes e IP

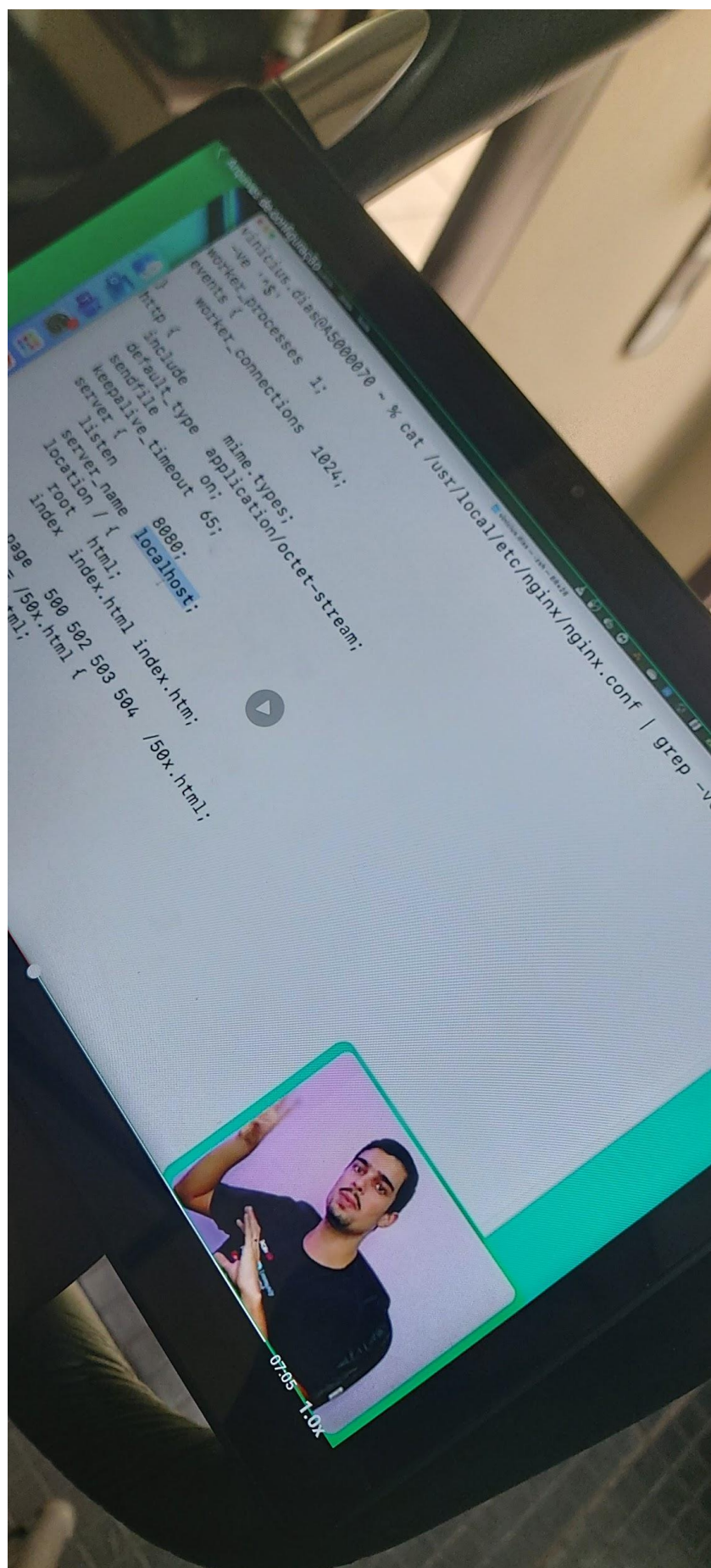


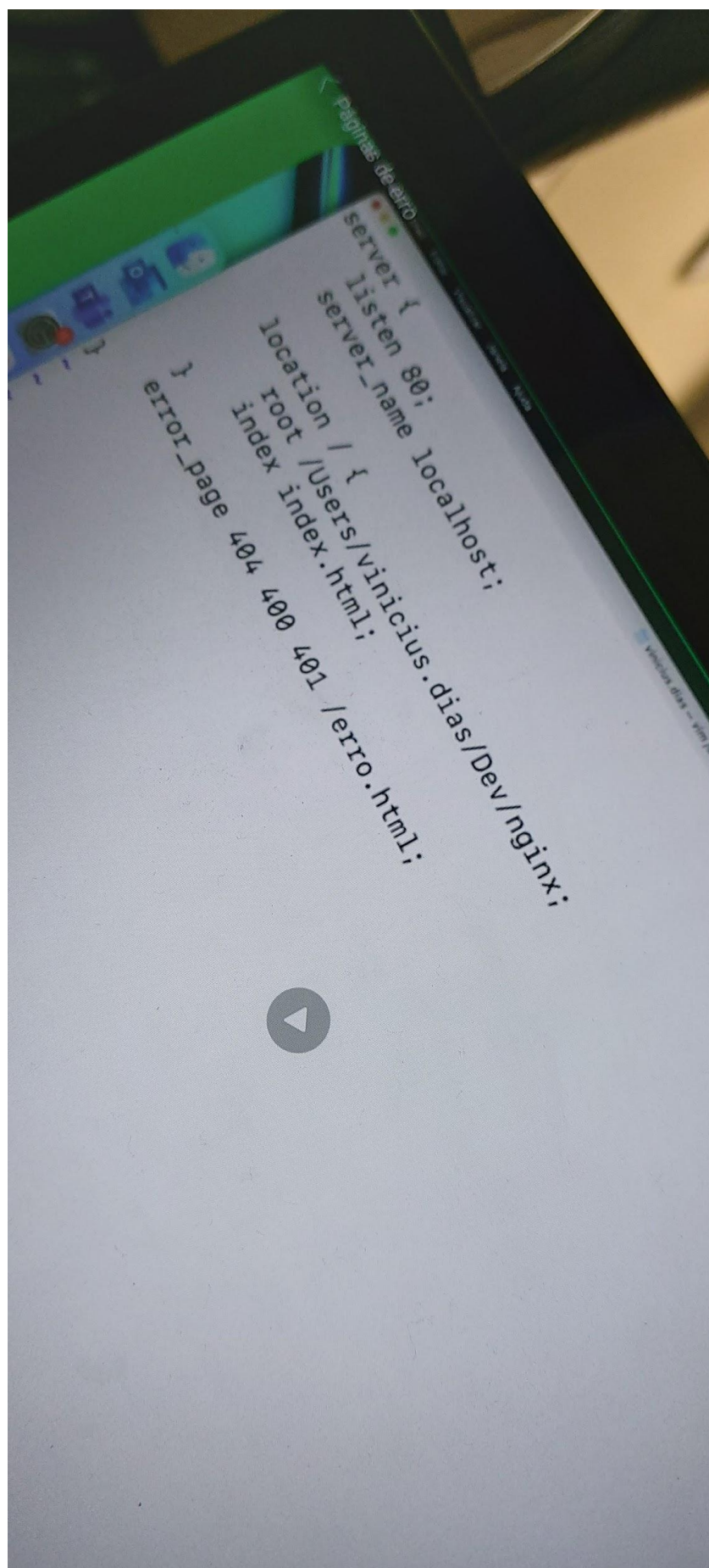


Simular levantando um servidor
sudo python -m SimpleHTTPServer 80

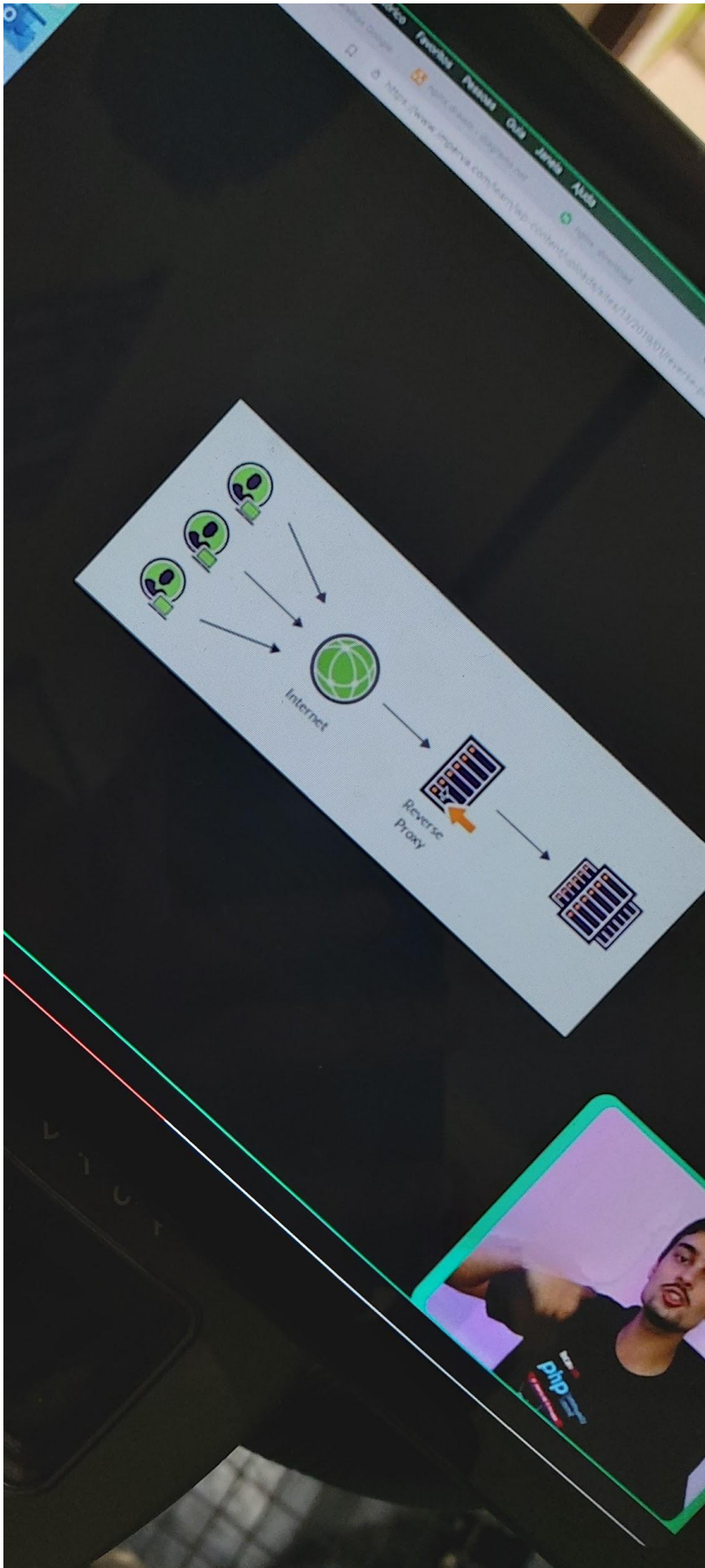
Nginx

Caminho padrão:





Proxy reverso



Recebe requisição e envia para VPS
Exemplo

```
server {
    listen 8080;
    server_name localhost;

    #charset koi8-r;
    #access_log logs/host.access.log main;

    location / {
        proxy_pass http://localhost;
    }

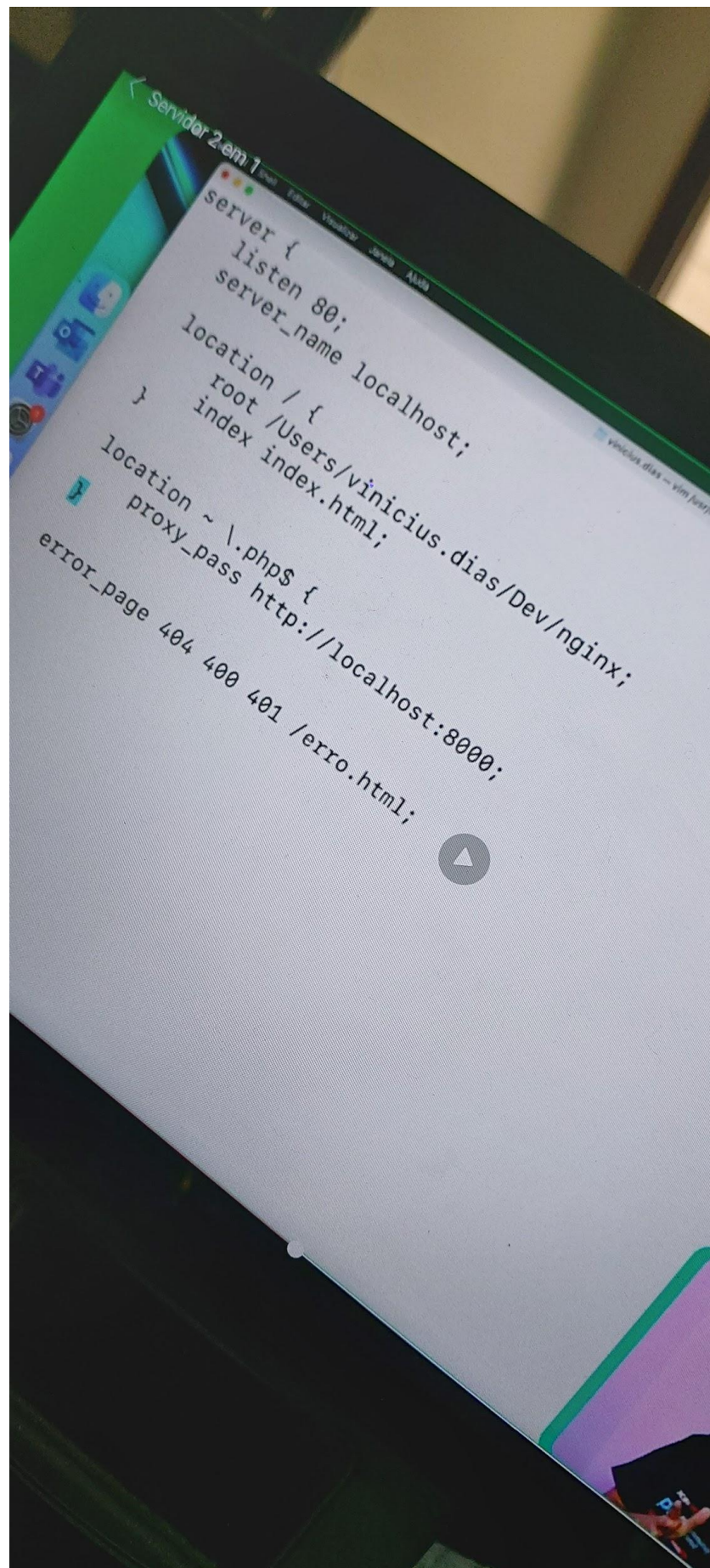
    #error_page 404
    # redirect server error pages to the static page /50x.html
    # redirect 500 502 503 504 /50x.html;
    # redirect 500 502 503 504 /404.html;
    # redirect 500 502 503 504 /50x.html;

    # to Apache listening on 127.0.0.1
```

Location ~ ... unsensitive case para regex do proxypass

Location ~* ... sensitive case para regex do proxypass

Exemplo qualquer coisa que termina com .php



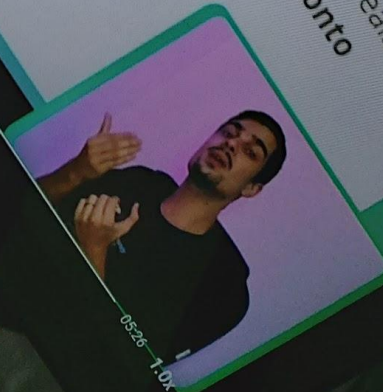
Configurando micro serviços VPS

```
Usando proxy_pass  
#gzip on;  
server {  
    listen  
    server_name 8080;  
    #charset koi8-r;  
    #access_log logs/host.access.log main;  
    location / {  
        #proxy_pass http://localhost;  
        location /servico1 {  
            proxy_pass http://localhost:8001/;  
        }  
        location /servico2 {  
            proxy_pass http://localhost;  
        }  
        #error_page 404 /404.html;  
        # redirect server error pages to the static page /50x.h  
        error_page 500 502 503 504 /50x.html;
```

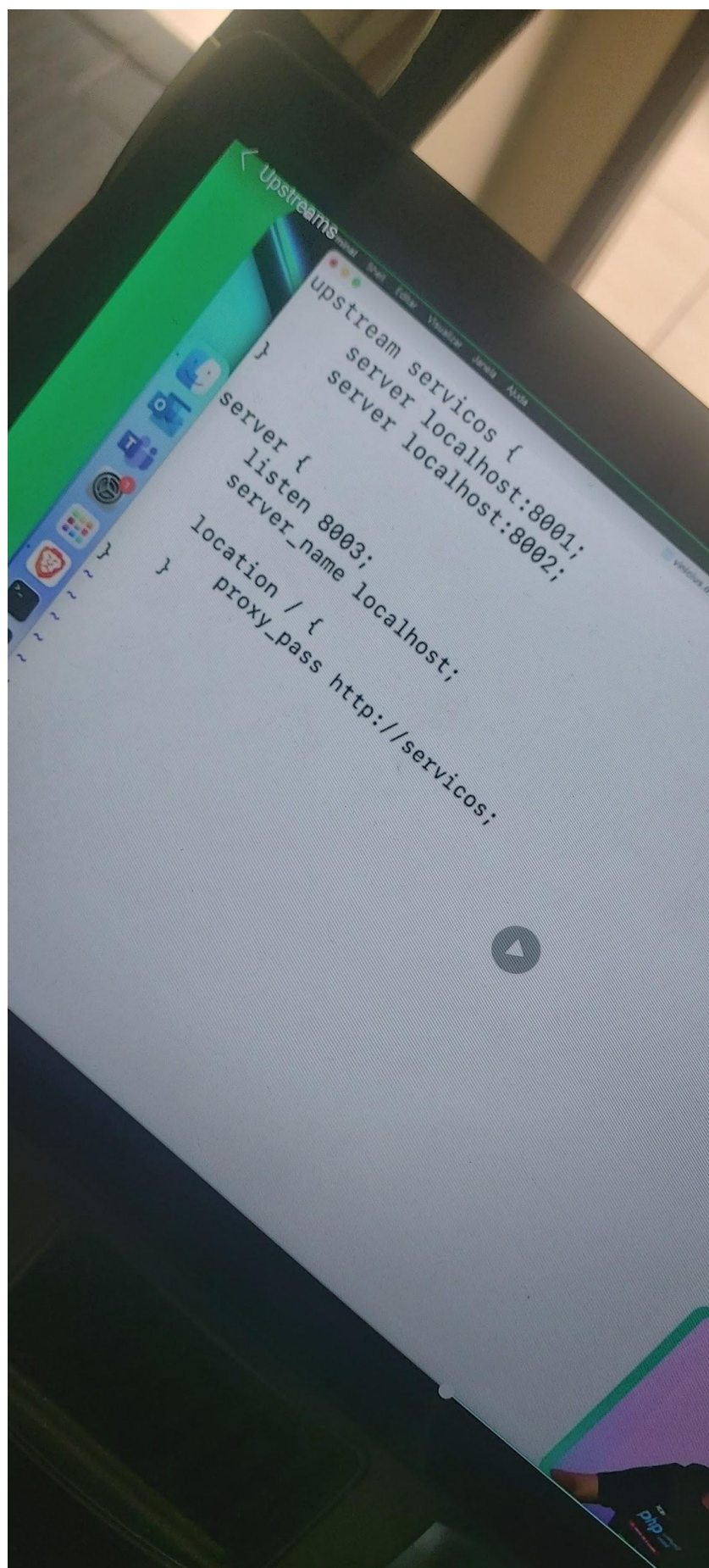
API gateway = quando vc tem vários proxys no nginx apontando para serviços diferentes

API Gateway

- Problema: Clientes acessando livremente os serviços geram caos
- Gateway fornece um proxy, uma fachada, para as necessidades reais
- Desvantagem: Esse portão de entrada pode se tornar um ponto central de falha



Exemplo de LoadBalancer



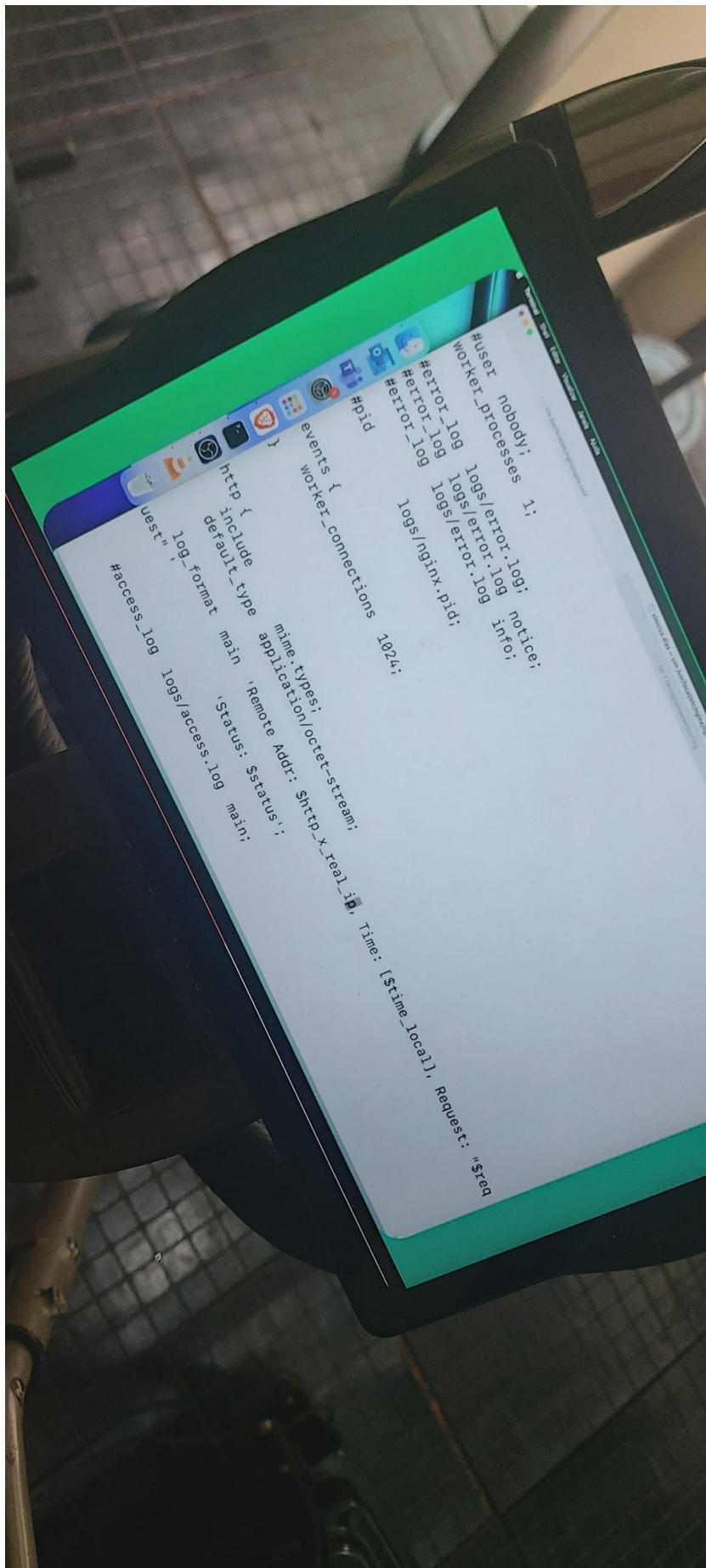
```
upstream services {  
    server localhost:8001;  
    server localhost:8002;  
}  
  
server {  
    listen 8003;  
    server_name localhost;  
    location / {  
        proxy_pass http://services;  
    }  
}
```

Sentando header no nginx

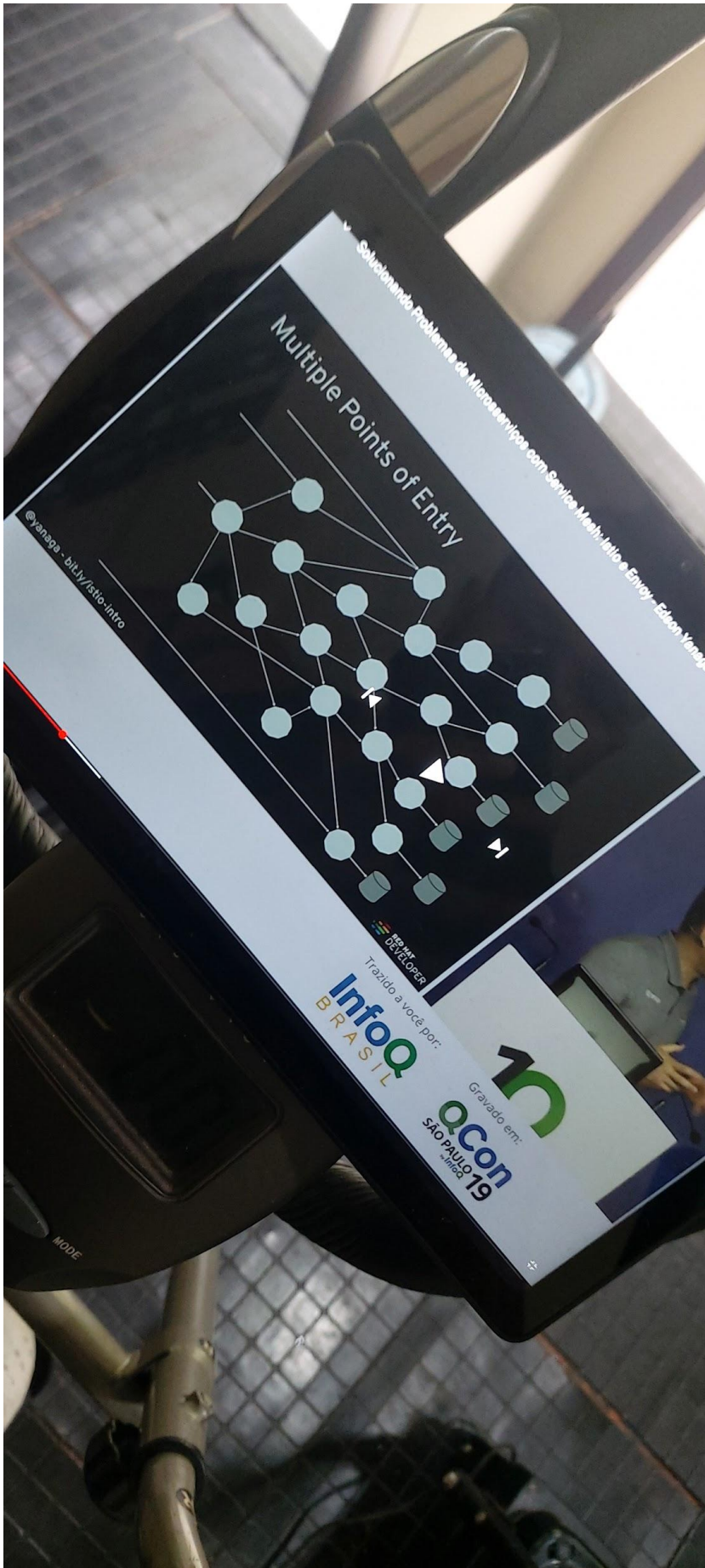
< Adicionando informações

```
upstream servicos {  
    server localhost:8001;  
    server localhost:8002;  
}  
  
server {  
    listen 8003;  
    server_name localhost;  
  
    location / {  
        proxy_pass http://servicos;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

Log que lê os dados do header o IP



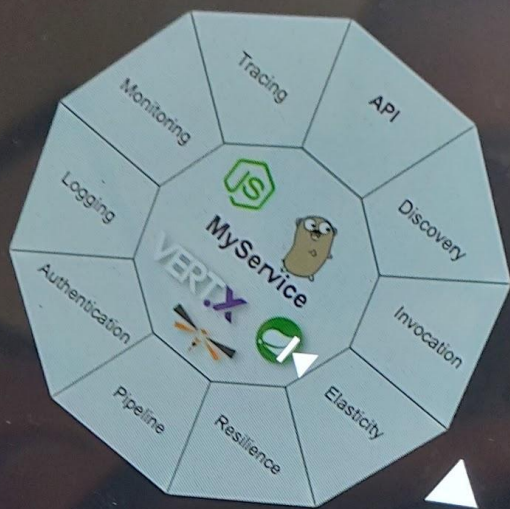
Service Mash - arquitetura para auxiliar softwares em microservicos
<https://youtu.be/AWpsmh47plk>



Ferramentas para microservicos

Resolvendo Problemas de Microserviços com Service

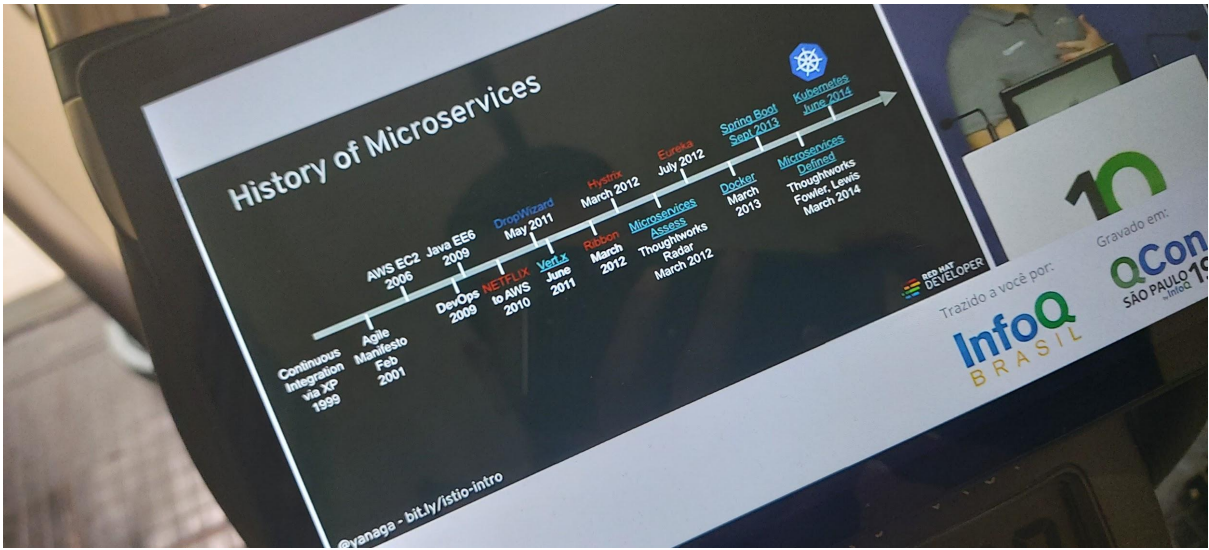
Microservices'ilities



Trazido a você por:

InfoQ
BRASIL

Gratuito
InfoQ
SÃO PAULO



Service Mesh Defined

A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as an array of lightweight network proxies that are deployed alongside application code, without the application needing to be aware

RENAULT
DEVELOPER

Trazido a você por:
InfoQ
BRASIL

Gravado em:
InfoQ
SÃO PAULO

Service mesh = serve para manter uns transações entre apis, parecido com commit/rollback do sql

Isto = ferramenta que orquestra o service mesh

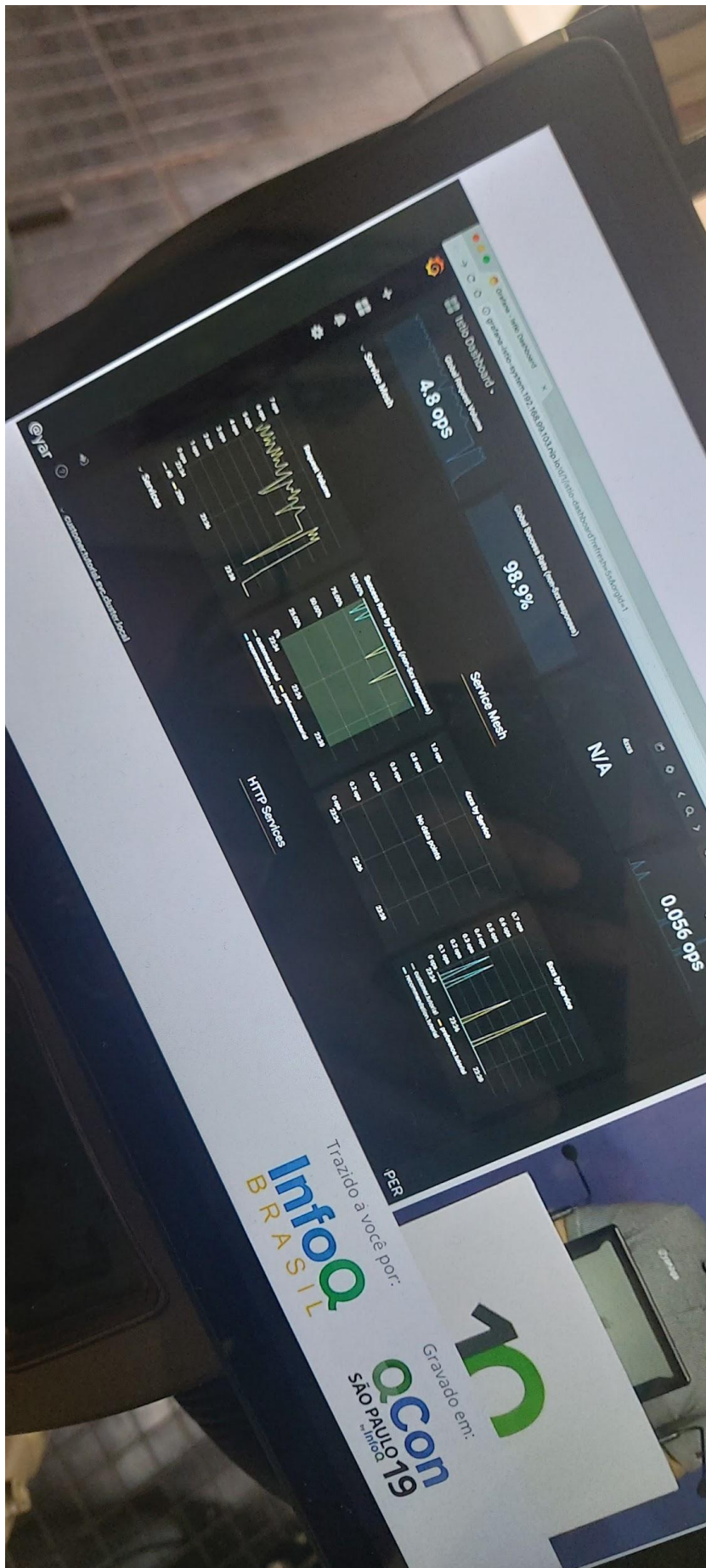
Microservices'ilities + Istio



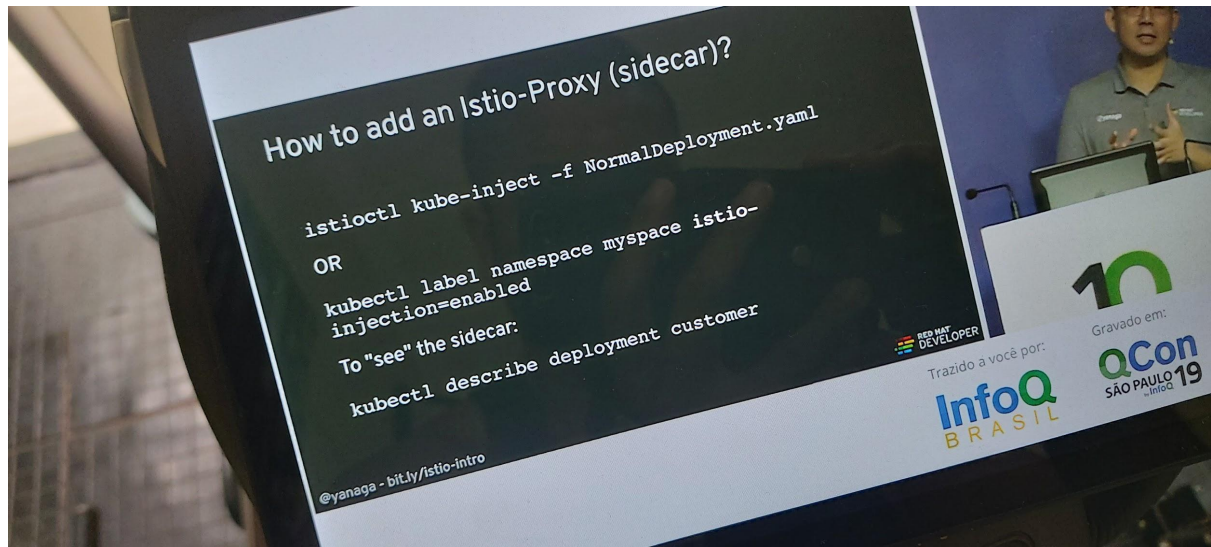
REACT
DEVELOPER

Trazido a você por
Info
BRAS

Exemplo do Istio no grafana

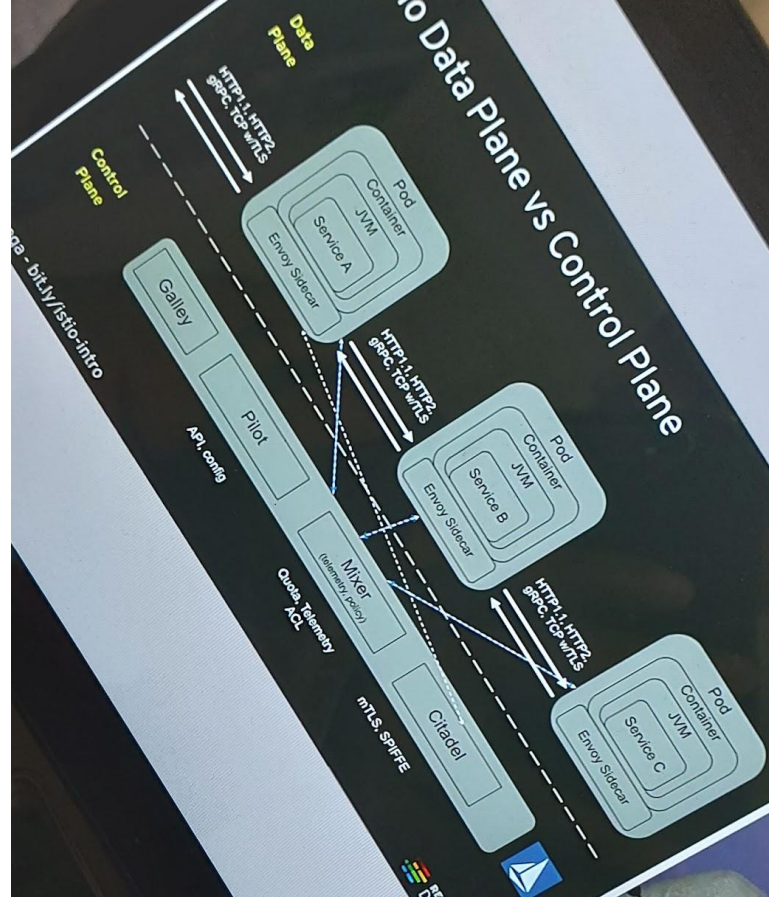


Como adicionar o sidcar(passageiro moto)no istio



Exemplo de um plano com Istio

Istio Data Plane vs Control Plane



Trazido a você por:

InfoQ
BRASIL

Gravado em:

19
São Paulo

Como instalar e confirmar Istio

@yanaga - bit.ly/istio-intro

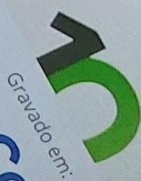
Exercises

bit.ly/istio-tutorial



Trazido a você por:

InfoQ
BRASIL



Gravado em:
qcon
SÃO PAULO 19

AWS Instance

Explicar sobre criação de imagens EC2 reaproveitamento de máquinas

Wordpad CC

<https://wordpad.cc/turma-devops>

Download Java

https://www.java.com/pt-BR/download/ie_manual.jsp?locale=pt_BR

Download vscode

<https://code.visualstudio.com/download>

Google: Start spring boot

Start spring command

<https://spring.io/guides/gs/spring-boot/>

Google: spring boot generator

<https://start.spring.io/>

Configurar Path Java Windows

<https://stackoverflow.com/questions/63043585/visual-studio-code-showing-java-11-or-more-recent-is-required-to-run-please-do/63287737>

Configurar java path windows

<https://www.java.com/en/download/help/path.html>

Start VPS spring boot

`./mvnw spring-boot:run`

Start test de unidade

`./mvnw test`

Maven install Windows

<https://mkyong.com/maven/how-to-install-maven-in-windows/>

Maven install mac

<https://www.appsdeveloperblog.com/how-to-install-maven-on-mac-os/>

::: Config git :::

git init

git add .

git commit -m "Mensagem"

git remote add origin git@bitbucket.org:didox_59/validadorcpf.git

git push -u origin master

Material de apoio dotnet

<https://www.torneseumprogramador.com.br/aulas/c-sharp>

Gerador de CPF

https://www.4devs.com.br/gerador_de_cpf

Gerador de .gitignore

<https://www.toptal.com/developers/gitignore>

Como rodar vscode terminal mac

<https://code.visualstudio.com/docs/setup/mac>

Aula de GIT

<https://www.torneseumprogramador.com.br/aula?id=JiyKMqVfP1M&aula=20&tipo=back-end&professor=Danilo>

Aula de HTTP

https://www.torneseumprogramador.com.br/aula?id=kRy_VVLWtf4&aula=41&tipo=back-end&professor=Danilo

Repo professor

<https://github.com/torneseumprogramador/spring-app-validar-cpf-treinamento-concrete>

Orientação a objetos

<https://www.torneseumprogramador.com.br/aula?id=NcZPGJfYEx8&aula=10&tipo=c-sharp&professor=Danilo>

<https://www.torneseumprogramador.com.br/aula?id=zvZE0aFaCSI&aula=11&tipo=c-sharp&professor=Danilo>

https://www.torneseumprogramador.com.br/aula?id=c1bvl_HqXKg&aula=12&tipo=c-sharp&professor=Danilo

Cache Aulas

<https://www.torneseumprogramador.com.br/aulas/cache>

Estrutura java cucumber

<https://github.com/Didox/estrutura-cucumber-vazia>

Sistema em dotnet para fazer clone

<https://github.com/Didox/turma-devops-dotnet>

Install dotnet core

<https://docs.microsoft.com/pt-br/dotnet/core/install/macos>

<https://dotnet.microsoft.com/download>

install home brew

<https://formulae.brew.sh/cask/dotnet>

Sistema em nodejs

<https://github.com/Didox/turma-devops-nodejs>

Instalar nodejs

<https://nodejs.org/en/download/>

dentro da pasta do projeto

npm install

npm run test

npm start

npm run test-comportamento

Instalador do Docker

<https://docs.docker.com/docker-for-windows/install/>

<https://docs.docker.com/docker-for-mac/install/>

Para gerar imagem docker da aplicação java

passo 1 gerar o build

`./mvnw package && java -jar target/validarCpf.jar`

Documentação jar Dockerfile java

<https://spring.io/guides/gs/spring-boot-docker/>

passo 2 - gerar o build do Dockerfile

`docker build -t didox/validador-cpf-turma-devops -f Dockerfile .`

Dockerfile Java

FROM openjdk:8-jdk-alpine

ARG JAR_FILE=target/*.jar

COPY \${JAR_FILE} app.jar

```
ENTRYPOINT ["java","-jar","/app.jar"]
```

README.md

Gerar Build do java

```
./mvnw package && java -jar target/validarCpf.jar
```

Gerar o build do docker

```
docker build -t didox/validador-cpf-turma-devops -f Dockerfile .
```

Rodar imagem docker e gravar localmente

```
docker run -d -p 8081:8080 --name validador-cpf-turma-devops  
didox/validador-cpf-turma-devops
```

Rodar imagem docker em modo iterativo localmente

```
docker run -it -p 8081:8080 --name validador-cpf-turma-devops  
didox/validador-cpf-turma-devops
```

Para parar o serviço rodar

```
docker stop validador-cpf-turma-devops
```

Para startar o serviço rodar

```
docker start validador-cpf-turma-devops
```

Para remover o serviço rodar

```
docker rm validador-cpf-turma-devops
```

Para fazer login no dockerhub

```
docker login
```

Criar a tag apontando para o repositório do docker hub

```
docker tag didox/validador-cpf-turma-devops  
hub.docker.com/r/didox/validador-cpf-turma-devops
```

Fazer o push da imagem para a docker hub

```
docker push didox/validador-cpf-turma-devops
```

passo 3 - gerar e startar a imagem

```
docker run -d -p 8081:8080 --name validador-cpf-turma-devops  
didox/validador-cpf-turma-devops
```

o primeiro é porta de saída que acessamos o docker

o segundo é a porta que roda a aplicação dentro do docker

8081:8080

Para parar o serviço rodar

```
docker stop validador-cpf-turma-devops
```

Para startar o serviço rodar

```
docker start validador-cpf-turma-devops
```

Para remover o serviço rodar

```
docker rm validador-cpf-turma-devops
```

Docker hub, local onde versionamos nossas imagens

<https://hub.docker.com/>

Gerar dockerfile C#

<https://docs.microsoft.com/pt-br/dotnet/core/docker/build-container?tabs=windows>

Dockerfile C#

```
-----  
FROM mcr.microsoft.com/dotnet/core/sdk:3.1 AS build  
WORKDIR /app  
# copy csproj and restore as distinct layers  
COPY *.csproj ./  
RUN dotnet restore  
# copy everything else and build  
COPY . ./  
RUN dotnet publish -c Release -o release  
# build runtime image  
FROM mcr.microsoft.com/dotnet/core/aspnet:3.1  
WORKDIR /app  
COPY --from=build /app/release .  
ENTRYPOINT ["dotnet", "WebTddBdd.dll"]  
-----
```

README.md

Gerar o build do docker

```
docker build -t didox/validador-cpf-csharp-turma-devops -f Dockerfile .
```

Rodar imagem docker e gravar localmente

```
docker run -d -p 5001:80 --name validador-cpf-csharp-turma-devops  
didox/validador-cpf-csharp-turma-devops
```

Rodar imagem docker em modo iterativo localmente

```
docker run -it -p 5001:80 --name validador-cpf-csharp-turma-devops  
didox/validador-cpf-csharp-turma-devops
```

Para parar o serviço rodar ###
docker stop validador-cpf-csharp-turma-devops

Para startar o serviço rodar ###
docker start validador-cpf-csharp-turma-devops

Para remover o serviço rodar ###
docker rm validador-cpf-csharp-turma-devops

Para fazer login no dockerhub ###
docker login

Criar a tag apontando para o repositório do docker hub ###
docker tag didox/validador-cpf-csharp-turma-devops
hub.docker.com/r/didox/validador-cpf-csharp-turma-devops

Fazer o push da imagem para a docker hub ###
docker push didox/validador-cpf-csharp-turma-devops

Dockerfile Nodejs

```
FROM node:10
WORKDIR /usr/src/app
COPY package-docker.json ./package.json
RUN npm install
COPY . .
CMD ["node", "./bin/www"]
```

README.md

start test ###
npm run test
npm run test-comportamento

start app ###
npm start

build ###
npm install

Gerar o build do docker

docker build -t didox/validador-cpf-nodejs-turma-devops -f Dockerfile .

Rodar imagem docker e gravar localmente

docker run -d -p 3000:3000 --name validador-cpf-nodejs-turma-devops
didox/validador-cpf-nodejs-turma-devops

Rodar imagem docker em modo iterativo localmente

docker run -it -p 3000:3000 --name validador-cpf-nodejs-turma-devops
didox/validador-cpf-nodejs-turma-devops

Para parar o serviço rodar

docker stop validador-cpf-nodejs-turma-devops

Para startar o serviço rodar

docker start validador-cpf-nodejs-turma-devops

Para remover o serviço rodar

docker rm validador-cpf-nodejs-turma-devops

Para fazer login no dockerhub

docker login

Criar a tag apontando para o repositório do docker hub

docker tag didox/validador-cpf-nodejs-turma-devops
hub.docker.com/r/didox/validador-cpf-nodejs-turma-devops

Fazer o push da imagem para a docker hub

docker push didox/validador-cpf-nodejs-turma-devops

Docker compose

docker-compose.yml

version: '3.9'

services:

sistema_concrete_java:

container_name: sistema_concrete_java_container

image: didox/validador-cpf-java-turma-devops

ports:

- "3001:8080"

sistema_concrete_csharp:

container_name: sistema_concrete_csharp_container

image: didox/validador-cpf-csharp-turma-devops

ports:

- "3002:80"

para buildar imagem

\$ docker-compose build

para levantar

\$ docker-compose up

\$ docker-compose up --detach # roda em background

para derrubar

\$ docker-compose stop

para mostrar log

\$ docker-compose logs --follow

para ver os serviços disponíveis

\$ docker-compose ps

para derrubar os serviços

\$ docker-compose down

para verificar network containers

\$ docker network ls

para entrar na maquina e verificar rede

\$ docker exec -it sistema_csharp bash

\$ docker exec -it sistema_java bash

\$ docker exec -it sistema_nodejs bash

```
# ver o IP interno do container
# $ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
sistema_csharp
# $ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
sistema_java
# $ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
sistema_nodejs
```

```
# mostra info network
# $ docker network inspect bridge
```

```
# verificar acesso interno containers mesma rede
# $ curl 172.17.0.1:3001 -- sistema java
```

Curso de docker - Linux tips

https://www.youtube.com/watch?v=0xxHiOSJVe8&list=PLf-O3X2-mxDkiUH0r_BadgtELJ_qyrFJ_

Aula sobre crontab

<https://www.torneseumprogramador.com.br/aula?id=SbdswGbDDko&aula=4&tipo=arquitetura&professor=Danilo>

Explicação load balancer nginx

<https://towardsdatascience.com/sample-load-balancing-solution-with-docker-and-nginx-cf1ffc60e644>

Repo docker compose professor:

<https://github.com/Didox/treinamento-docker-concrete>

● Kubernetes

Kubernetes

Master = (control plane)

- Api = responsável por receber e executar os comandos
- Control manage = manter e atualizar o estado desejado
- Schedule = defini onde determinado pode será executado em nossos cluster
- Etcd = responsável por armazenar as informações, chave e valor

Nodes

- kubelet = responsável pela execução dos pods dentro dos nodes
- K-proxy = responsável pela comunicação entre os nodes

Kubernbetes

Curso:

<https://youtu.be/X48VuDVv0do>

https://youtu.be/uFgcCL3t_JM

Infra

<https://youtu.be/pV0nkr61XP8>

<https://youtu.be/CNZmKjklfeg>

<https://youtu.be/mk8mKVymU6E>

<https://youtu.be/04xdSqcEEyE>

<https://youtu.be/L0o1pyRTO2c>

4 horas

<https://youtu.be/zz1p3gjyHgc>

Multumaster

<https://youtu.be/-wbtj11MqvK>

<https://youtu.be/-W8b4d8ITNY>

<https://youtu.be/rYQKjEF5thw>

Com istio

<https://youtu.be/Qk7FFBby43U> -

set kubectl manual

<https://www.youtube.com/watch?v=UCync6iHCpo>

Ingress controller

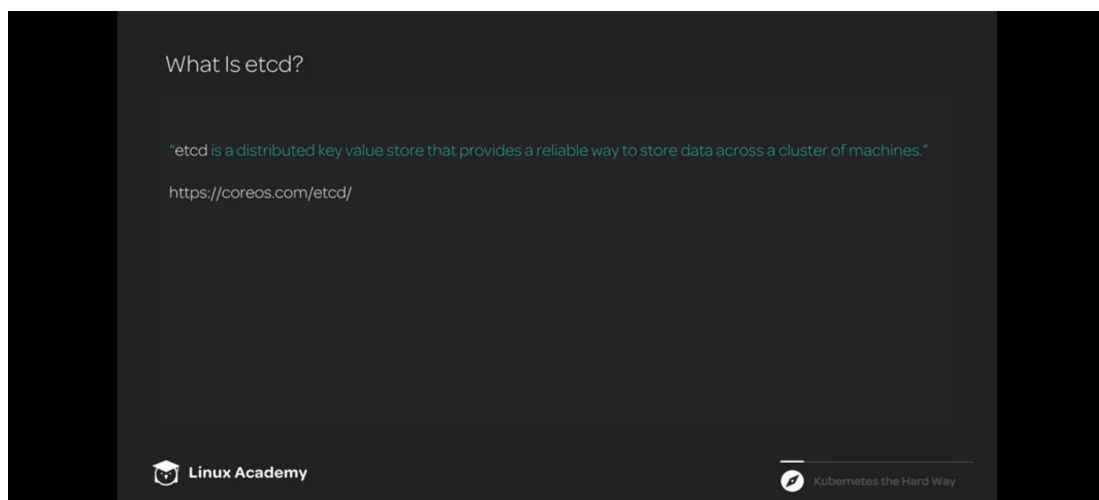
https://youtu.be/_MsdFRI12jY

[Kubernetes - Configurando NGINX Ingress Controller](#)

<https://www.youtube.com/watch?v=AGSGcUzkqrE>

Configurar ConfigMap com Nginx interno kubernetes

https://youtu.be/uFgcCL3t_JM?t=3055



<https://youtu.be/a0y3-YDH-70>

Helm



<https://youtu.be/-ykwb1d0DXU>

Instalar o VM

<https://www.virtualbox.org/wiki/Downloads>

Client kubectl

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

Instalar o minikube

<https://minikube.sigs.k8s.io/docs/start/>

minikube start - inicia o cluster

minikube start --driver=virtualbox - inicia o minikube mandando para o driver do virtualbox

Decobrindo o IP do cluster

kubectl get nodes -o wide

kubectl run didox-nginx --image=nginx:latest # cria um pod com uma imagem docker

kubectl get pods -o wide # Lista o pod e mostra o IP local

kubectl get pods --watch # Acompanha em modo watch a criação do pod

kubectl delete pods didox-nginx # Apaga o pod criado

kubectl describe pod didox-nginx # Descreve o pod

kubectl exec -it didox-nginx bash # Acessa o pod via ssh methodo deprecated

kubectl exec --stdin --tty didox-nginx -- /bin/bash # Acessa o pod via ssh

kubectl edit pod didox-nginx

kubectl delete pods --all # apaga todos os pods

kubectl delete services --all # apaga todos os serviços

Criando pod via script

-----pod-nginx.yml-----

apiVersion: v1

kind: Pod

```
metadata:
  name: didox-nginx
spec:
  containers:
    - name: container-pod-nginx
      image: nginx:latest
      ports:
        - containerPort: 80
```

```
kubectl apply -f pod-nginx.yml
kubectl describe -f pod-nginx.yml
```

Criando pod via script

```
-----pod-nodejs.yml-----
apiVersion: v1
kind: Pod
metadata:
  name: didox-nodejs
  labels:
    app: label-pod-didox-nodejs
spec:
  containers:
    - name: container-pod-nodejs
      image: didox/validador-cpf-nodejs-turma-devops
      ports:
        - containerPort: 3000
```

```
kubectl apply -f pod-nginx.yml
kubectl describe -f pod-nginx.yml
```

NodePort exporta porta do pod para o cluster

```
-----configurando nodePort --- svc-nginx.yml-----
apiVersion: v1
kind: Service
metadata:
  name: nodeport-svc-pod-didox-nginx
spec:
  type: NodePort
  selector:
    app: label-pod-didox-nginx
  ports:
    - port: 80
      nodePort: 30000 #até 32767
```

```
kubectl apply -f svc-nginx.yml
```

kubectl describe -f svc-nginx.yml

NodePort exporta porta do pod para o cluster

-----configurando nodePort --- svc-nodejs.yml-----

apiVersion: v1

kind: Service

metadata:

name: nodeport-svc-pod-didox-nodejs

spec:

type: NodePort

selector:

app: label-pod-didox-nodejs

ports:

- port: 3000

nodePort: 30001 #até 32767

kubectl apply -f svc-nodejs.yml

kubectl describe -f svc-nodejs.yml

ClusterIP cria IP fixo para ser acessado dentro da rede do cluster

-----configurando ClusterIP --- svc-clusterip.yml-----

apiVersion: v1

kind: Service

metadata:

name: clusterip-didox

spec:

type: ClusterIP

selector:

app: label-pod-didox-nodejs

ports:

- port: 3000

targetPort: 3000

kubectl apply -f svc-clusterip.yml

kubectl describe -f svc-clusterip.yml

Repo professor:

<https://github.com/Didox/kubernetes-aula-concrete>

Desafio final:

<https://github.com/Didox/desafio-final-concrete/blob/main/README.md>

Download heroku cli

<https://devcenter.heroku.com/articles/heroku-cli>

Testar se heroku cli está instalado

heroku --version

heroku login -i

heroku git:remote -a dev-meu-projeto

git push heroku master

ou se estiver no main rodar

git push heroku main

gitlab

image: openjdk

stages:

- build
- tdd
- build-docker
- dev
- qa
- prod

build:

script:

- ./mvnw package

stage: build

tdd:

script:

- ./test.sh

stage: tdd

dependencies:

- build

build-docker:

image: docker

services:

- docker:dind

before_script:

- docker login -u \$CI_REGISTRY_USER -p \$CI_REGISTRY_PASSWORD

stage: build-docker
dependencies:
- tdd
script:
- tag=\$(git describe --tags --abbrev=0)
- docker build -t didox/validador-cpf-java-turma-devops:\$tag -f Dockerfile .
- docker push didox/validador-cpf-java-turma-devops:\$tag

dev:
image: ruby
stage: dev
dependencies:
- build-docker
script:
- gem install dpl
- dpl --provider=heroku --app=dev-accenture
--api-key=\$HEROKU_PRODUCTION_API_KEY
only:
- master

qa:
image: ruby
stage: qa
dependencies:
- dev
script:
- gem install dpl
- dpl --provider=heroku --app=qa-accenture
--api-key=\$HEROKU_PRODUCTION_API_KEY
only:
- master

prod:
image: ruby
stage: prod
dependencies:
- qa
script:
- gem install dpl
- dpl --provider=heroku --app=prod-accenture
--api-key=\$HEROKU_PRODUCTION_API_KEY
only:
- master
<https://docs.gitlab.com/ee/ci/yaml/#whenmanual>
TODO - testar
when: manual

Gitlab runner

<https://docs.gitlab.com/runner/install/docker.html>

Windows, Linux

```
docker run -d --name gitlab-runner --restart always -v  
/srv/gitlab-runner/config:/etc/gitlab-runner -v /var/run/docker.sock:/var/run/docker.sock  
gitlab/gitlab-runner:latest
```

MacOS

```
docker run -d --name gitlab-runner --restart always -v  
/Users/Shared/gitlab-runner/config:/etc/gitlab-runner -v  
/var/run/docker.sock:/var/run/docker.sock gitlab/gitlab-runner:latest
```

Definir volume do gitlab runner

```
docker volume create gitlab-runner-config
```

entrar dentro do container

```
docker exec -it gitlab-runner bash
```

Rodar o comando "gitlab-runner register" dentro do container

```
gitlab-runner register
```

Gitlab com runner

image: openjdk

stages:

- build
- tdd
- build-docker
- dev
- qa
- prod

build:

script:

- ./mvnw package

stage: build

tags:

- runner-didox

tdd:

script:

- ./test.sh

stage: tdd

dependencies:

- build

tags:

- runner-didox

build-docker:

image: docker

services:

- docker:dind

before_script:

- docker login -u \$CI_REGISTRY_USER -p \$CI_REGISTRY_PASSWORD

stage: build-docker

dependencies:

- tdd

script:

#- tag=\$(git describe --tags --abbrev=0)

- docker build -t didox/validador-cpf-java-turma-devops:\$tag -f Dockerfile .

- docker push didox/validador-cpf-java-turma-devops:\$tag

- docker build -t didox/validador-cpf-java-turma-devops -f Dockerfile .

- docker push didox/validador-cpf-java-turma-devops

dev:

image: ruby

stage: dev

dependencies:

- build-docker

script:

- gem install dpl

- dpl --provider=heroku --app=dev-accenture

--api-key=\$HEROKU_PRODUCTION_API_KEY

only:

- master

qa:

image: ruby

stage: qa

dependencies:

- dev

script:

- gem install dpl

- dpl --provider=heroku --app=qa-accenture

--api-key=\$HEROKU_PRODUCTION_API_KEY

only:

- master

prod:

image: ruby

stage: prod

dependencies:

- qa

script:

- gem install dpl

- dpl --provider=heroku --app=prod-accenture

--api-key=\$HEROKU_PRODUCTION_API_KEY

only:

- master

<https://docs.gitlab.com/ee/ci/yaml/#whenmanual>

TODO - testar

when: manual

Como fazer deploy via GIT no azure

https://www.torneseumprogramador.com.br/aula?id=y8Y4ln_Hgqo&aula=104&tipo=c-sharp&professor=Danilo

Como fazer deploy via visual studio no azure

<https://www.torneseumprogramador.com.br/aula?id=PNScSt58ua0&aula=85&tipo=c-sharp&professor=Danilo>

Install az cli

<https://docs.microsoft.com/pt-br/cli/azure/install-azure-cli-macos>

<https://docs.microsoft.com/pt-br/cli/azure/install-azure-cli-windows?tabs=azure-cli>

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-linux?pivots=apt>

simples deploy AZURE

<https://docs.microsoft.com/en-us/azure/app-service/quickstart-java?tabs=javase&pivots=platform-linux>

rodar e configurar serviço de deploy

mvn com.microsoft.azure:azure-webapp-maven-plugin:1.12.0:config

fazer deploy

mvn package azure-webapp:deploy

apagar deploy

az group delete --name grupo-treinamento --yes

AWS CLI

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

Azure ZKS

az aks get-credentials --name treinamento-concrete --resource-group Kubernetes-Test-Danilo

-----Criando IP Publico com LoadLancer do Kubernetes

apiVersion: v1

kind: Service

metadata:

name: loadbalancer-svc-javaapp

spec:

type: LoadBalancer

ports:

- port: 80

targetPort: 8080

selector:

app: label-pod-javapp

-----Criando IP Publico com LoadLancer do Kubernetes

Instalar o AWS CLI

https://docs.aws.amazon.com/pt_br/cli/latest/userguide/install-cliv2.html

gerar um usuário no IAM

baixar o csv e colocar as informações da conta no configure do aws cli

comando:

aws configure

Meu PPT

https://docs.google.com/presentation/d/1h81actx7ReWqXyzGhYJRuH_byCxALVL6MbLUvxxmmU/edit#slide=id.gf113ee1e2f_0_135

Terraform

Baixar e configurar o terraform

<https://learn.hashicorp.com/tutorials/terraform/install-cli?in=terraform/aws-get-started>

depois de instalar o terraform usar:

Dentro do diretório que tem um arquivo main.tf

Exemplo:

```
provider "aws" {  
  region = "us-west-1"  
}
```

terraform init

```

$
[Danilo@MacBook-Pro-de-Danilo:~/projetos/terraform/itau_terraform]
$ cd terraform/provider/
[Danilo@MacBook-Pro-de-Danilo:~/projetos/terraform/itau_terraform/terraform/provider]
$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.59.0...
- Installed hashicorp/aws v3.59.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[Danilo@MacBook-Pro-de-Danilo:~/projetos/terraform/itau_terraform/terraform/provider]
$

```

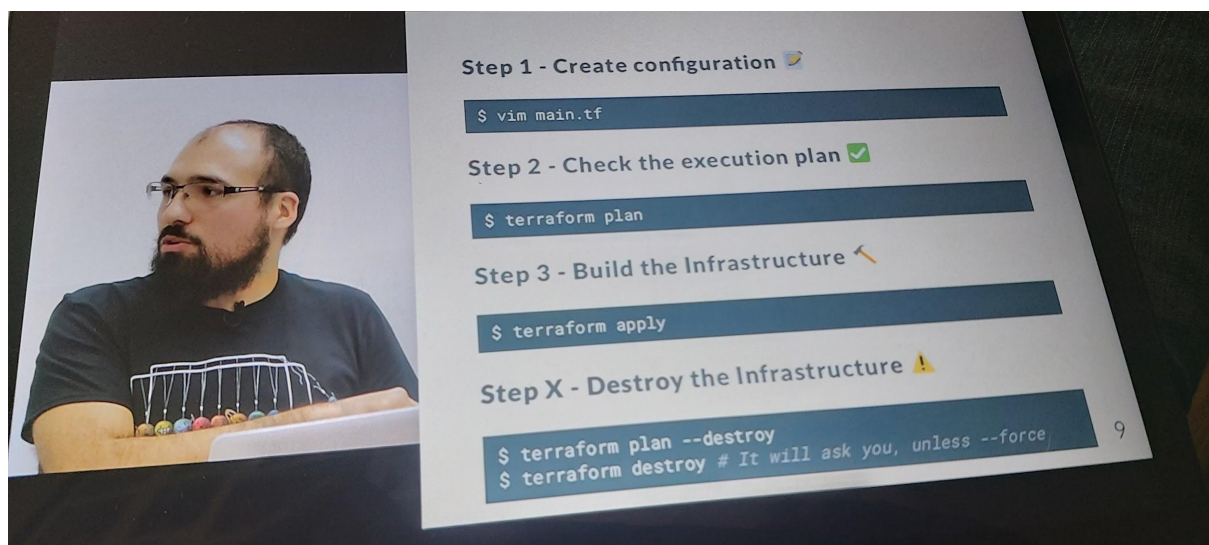
Vídeo legal sobre a língua HCL e seus blocos

<https://youtu.be/RLwvMDgVU80>

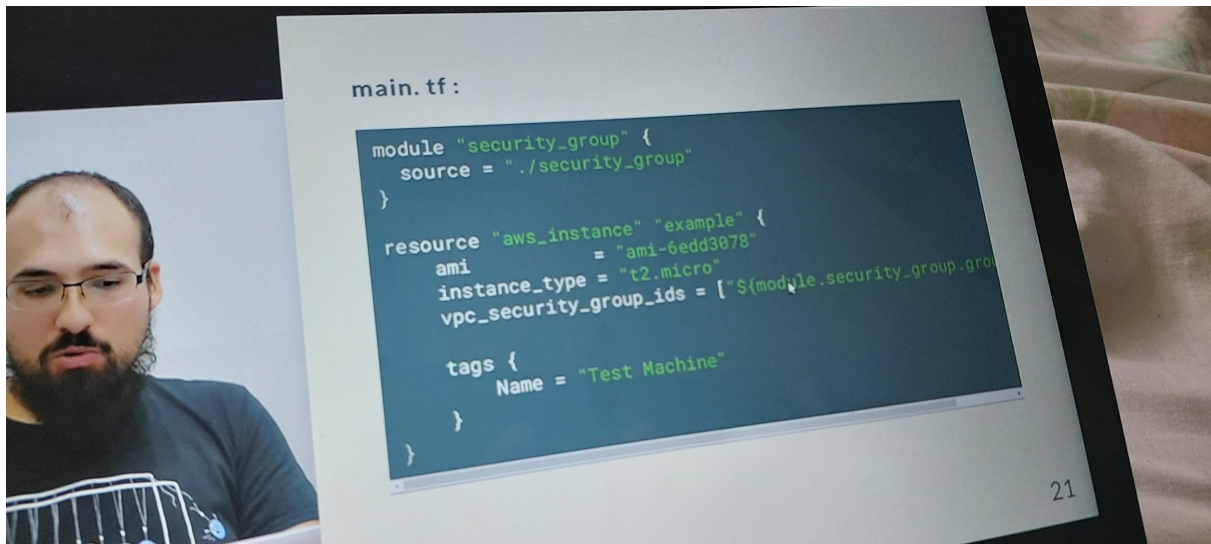
HCL linguagem do terraform

O Terraform também aceita o formato Json, mas o uso do HCL é fortemente recomendado, visto que você escreve bem menos código usando HCL e a chance de você esquecer uma chave na hora de criar o Json é grande.

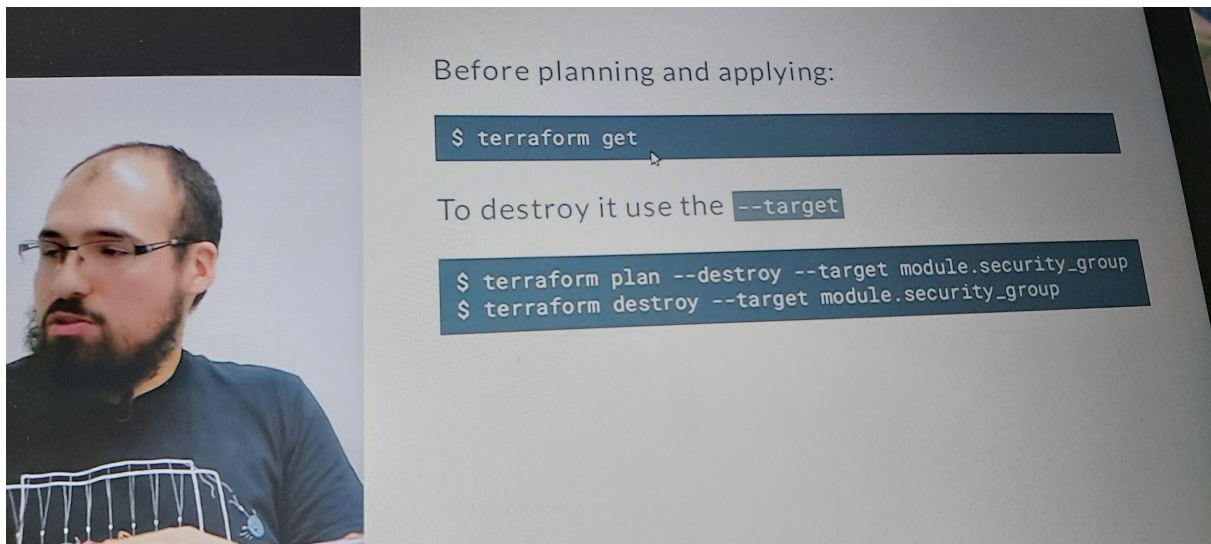
Exemplo de arquivo .tf, extensão usada pelo Terraform para provisionamento de uma instância EC2.



Module terraforms



No source de um modulo terraform, vc pode colocar um repo git. Usar um terraform get para fazer download



A linguagem HCL

Explorando o HCL

Blocos

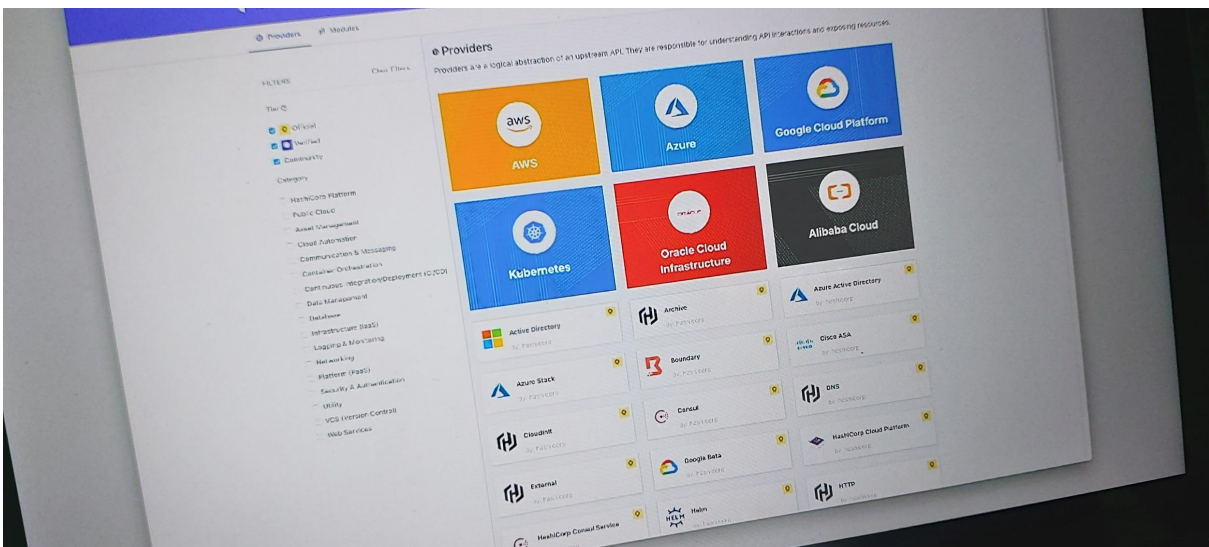
Provider block

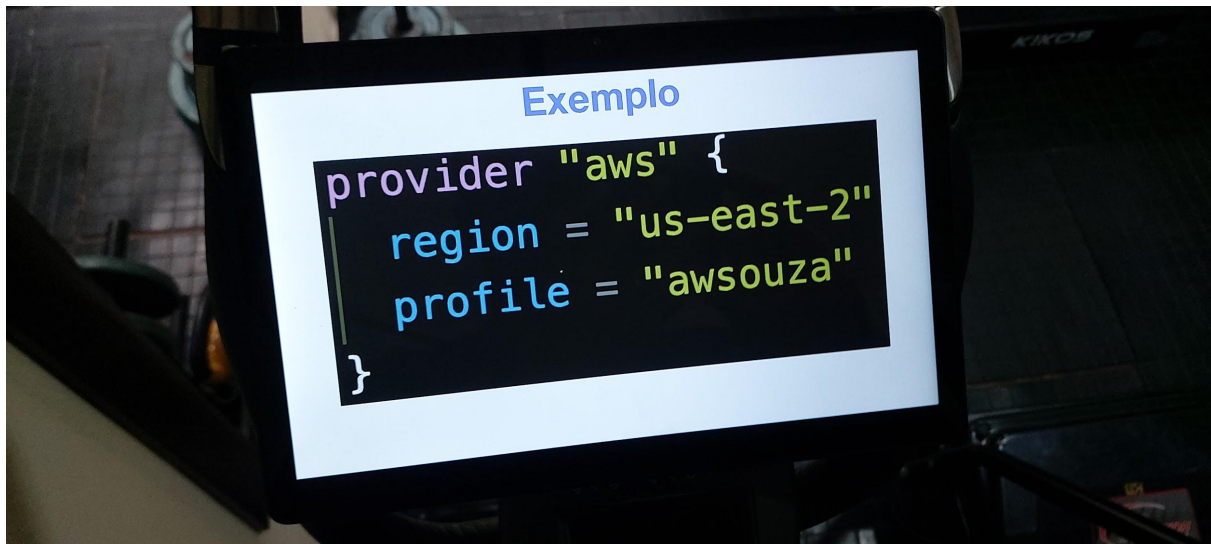
```
provider "algum_provider" {  
  // Inputs para o provider  
}
```

Provider block

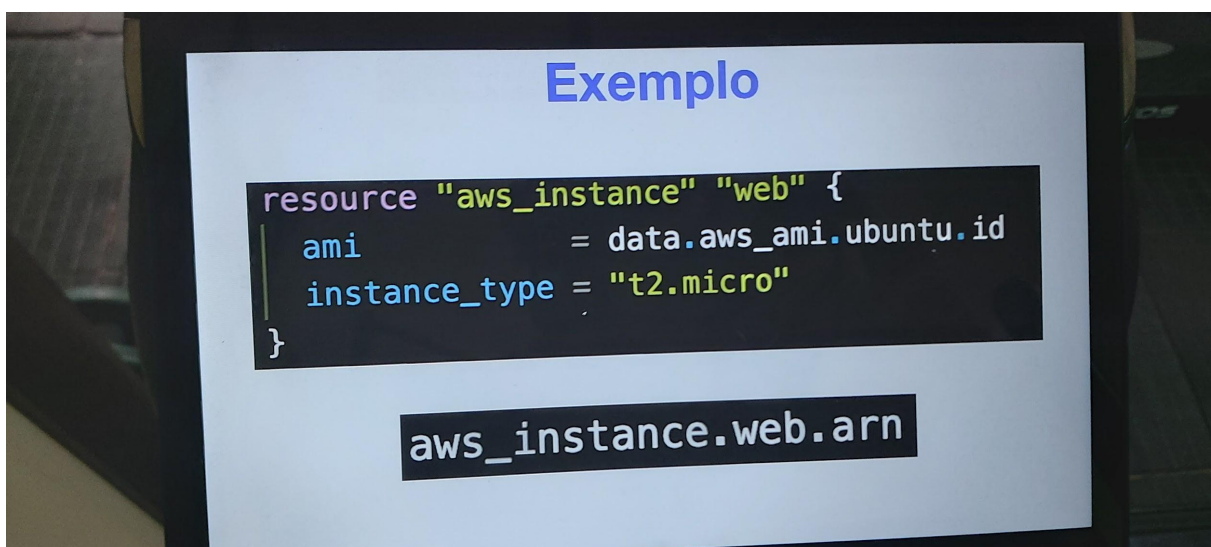
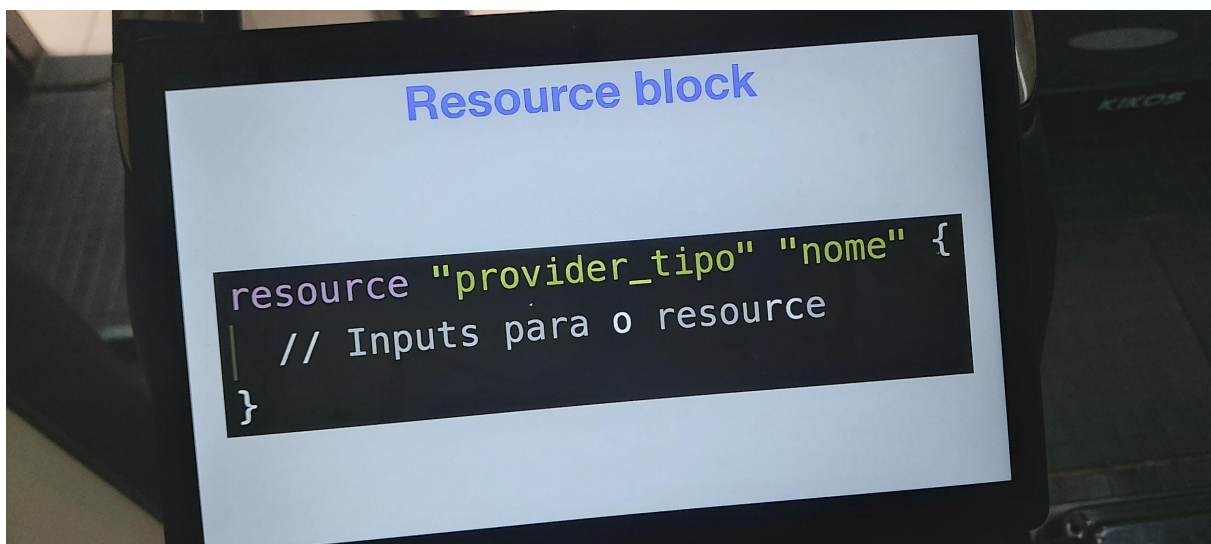
```
provider "algun_provider" {  
  // Inputs para o provider  
}
```

Lista de providers - <https://registry.terraform.io/browse/providers>





Resources



Variable e output

Variable e Output block

```
variable "nome_de_uma_variável" {  
  type = "tipo da variável. string, number, bool, etc."  
  default = "Essa variável tem algum valor padrão, ou sempre deve ser passada?"  
}  
  
output "nome_do_seu_output" {  
  value = "Algum output de um resource, módulo ou variável"  
}
```

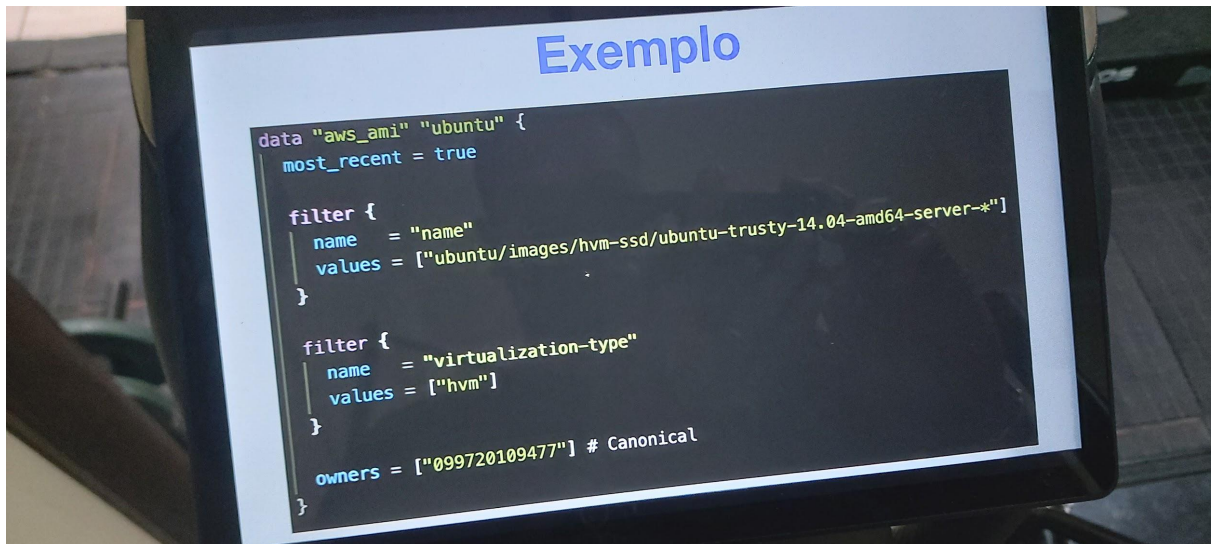
Exemplo

```
variable "idade" {  
  type      = number  
  default   = 28  
  description = "Qual é a sua idade?"  
}  
  
output "idade" {  
  value      = var.idade  
  description = "Sua idade"  
}
```

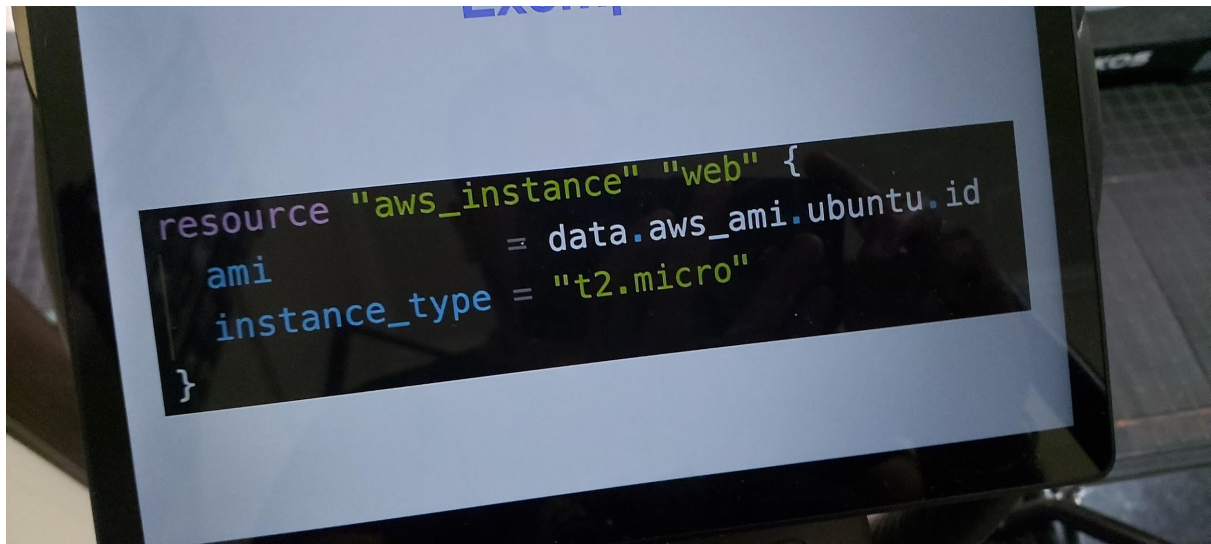
Bloco data: para buscas no provider, para trazer dados

Resource block

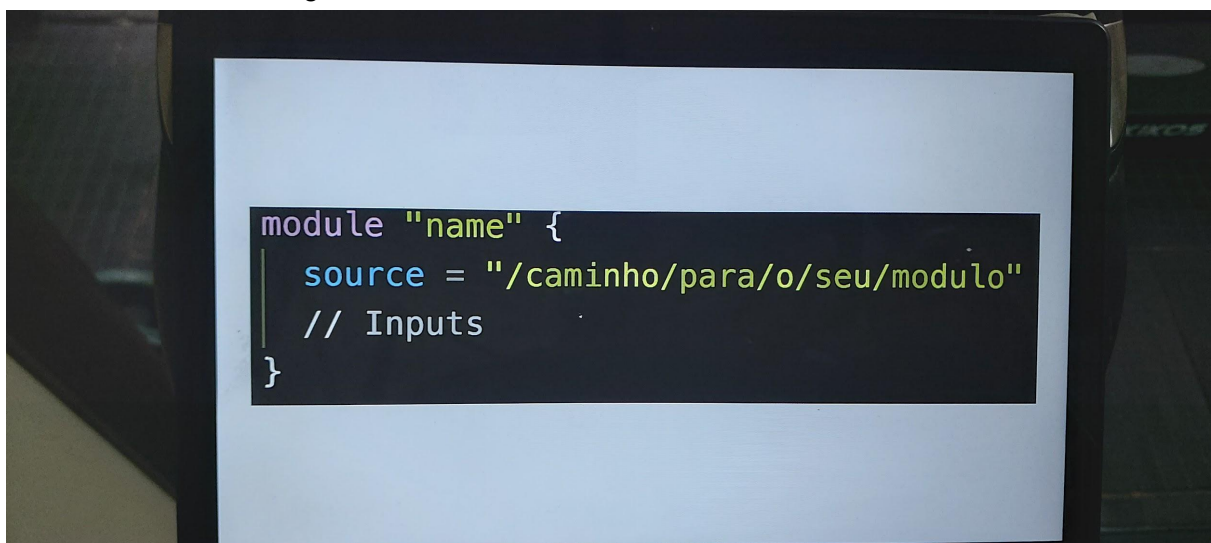
```
data "provider_tipo" "nome" {  
  // Inputs para o data  
}
```



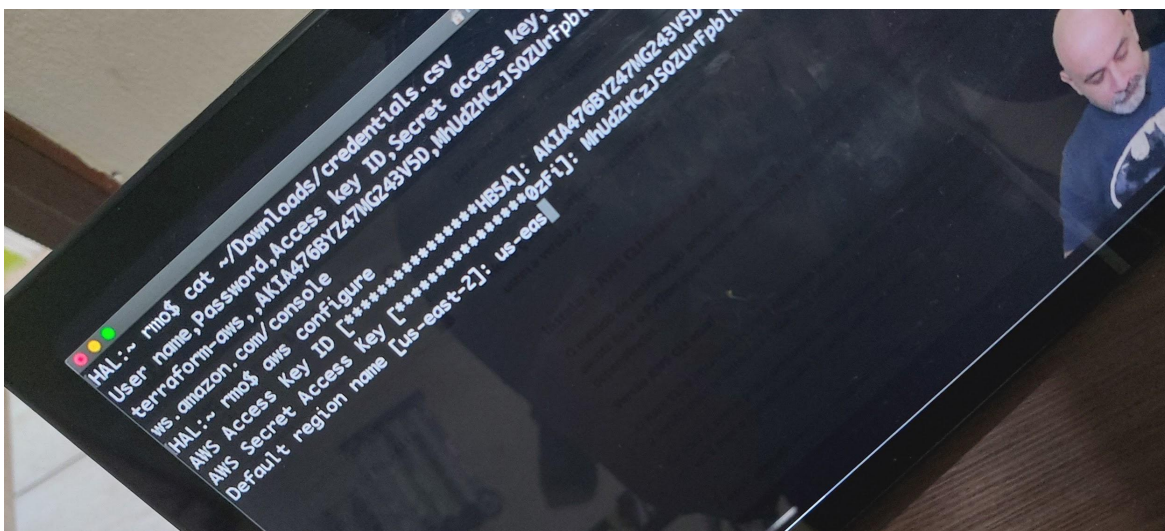
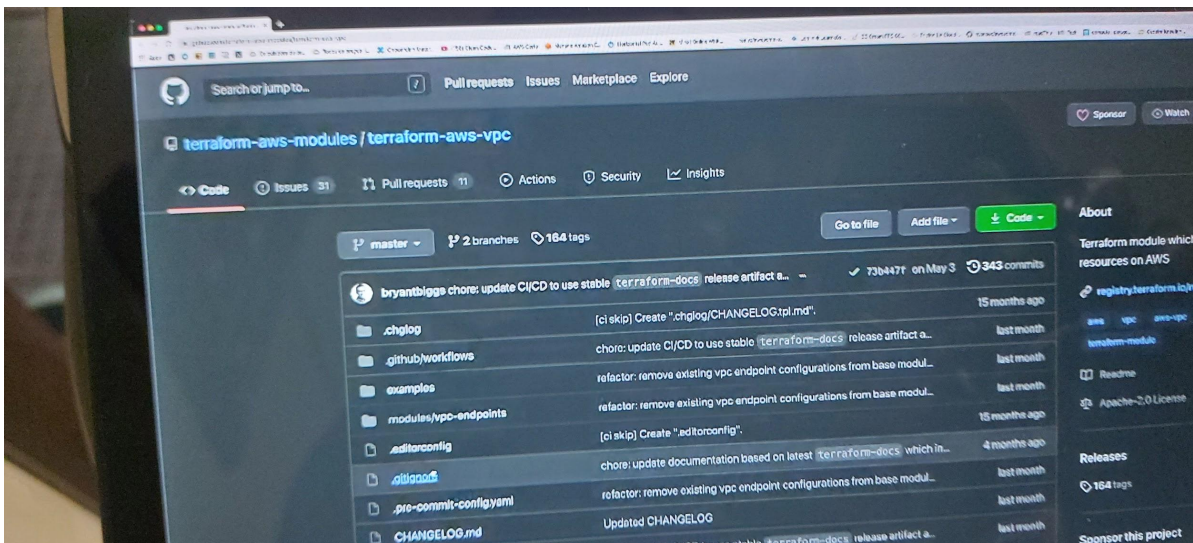
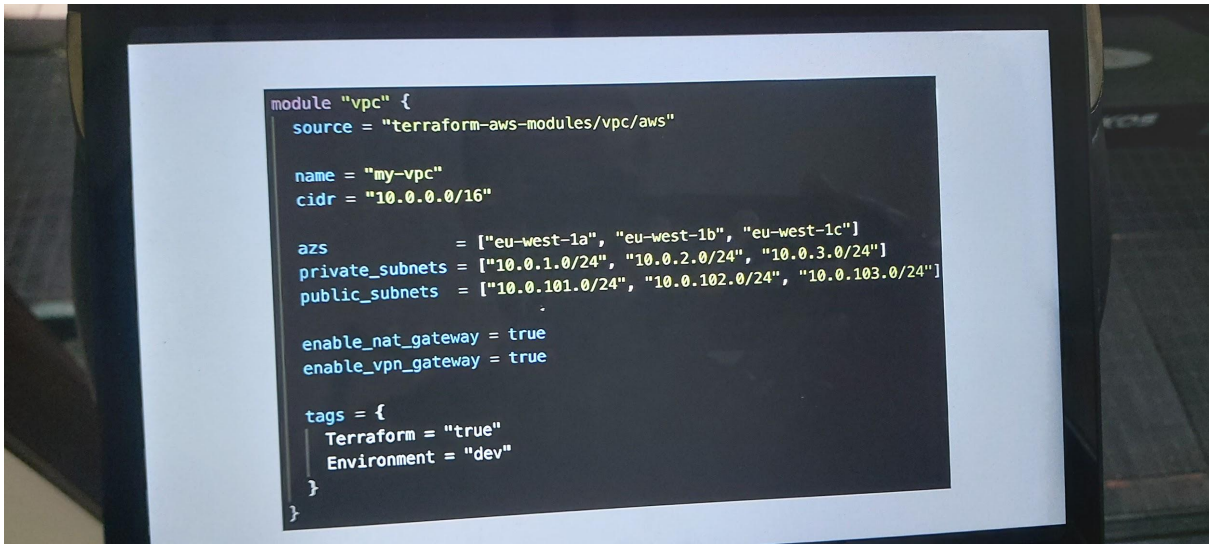
Uso do data



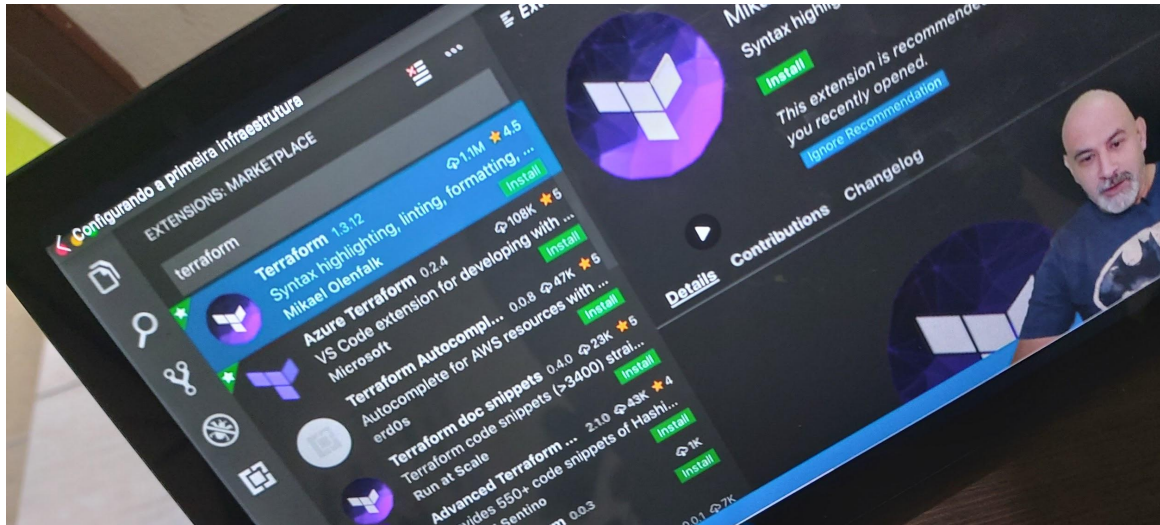
Módulo um dos mais legais



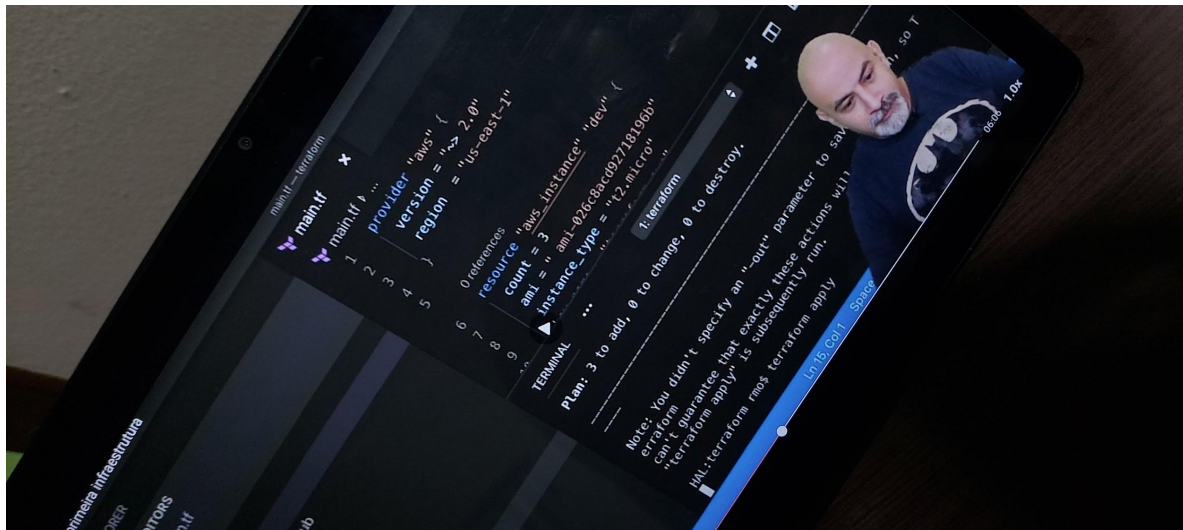
Cria funções por módulo



Plug-in terraforms vscode



Exemplo terraform criar 3 máquinas



```
provider "aws" {  
  region = "us-east-1"  
}
```

```
resource "aws_instance" "maqui" {  
  ami = "ami-013f17f36f8b1febf"  
  instance_type = "t2.micro"  
  key_name = "concrete-terraform"  
  tags = {  
    Name = "maqui${count.index}"  
  }  
  count = 2  
  vpc_security_group_ids = ["${aws_security_group.acesso_ssh.id}"]  
  depends_on = [aws_security_group.acesso_ssh, aws_s3_bucket.concete_s3]
```

```

}

resource "aws_security_group" "acesso_ssh" {
  name      = "acesso_ssh"
  description = "Treinamento concrete com acesso a SSH"
  ingress {
    description = "Para acessar a maquina via ssh"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["187.3.223.136/32"]
  }
  tags = {
    Name = "acesso_ssh_concrete"
  }
}

resource "aws_s3_bucket" "concrete_s3" {
  bucket = "concrete-s3"
  acl    = "public-read"

  tags = {
    Name = "Concrete S3"
  }
}

```

gerar um chave na sua máquina:
ssh-keygen -f concrete-terraform -t rsa

para verificar se o script de criação de maquina está ok
terraform plan

para aplicar a criação da maquina
terraform apply

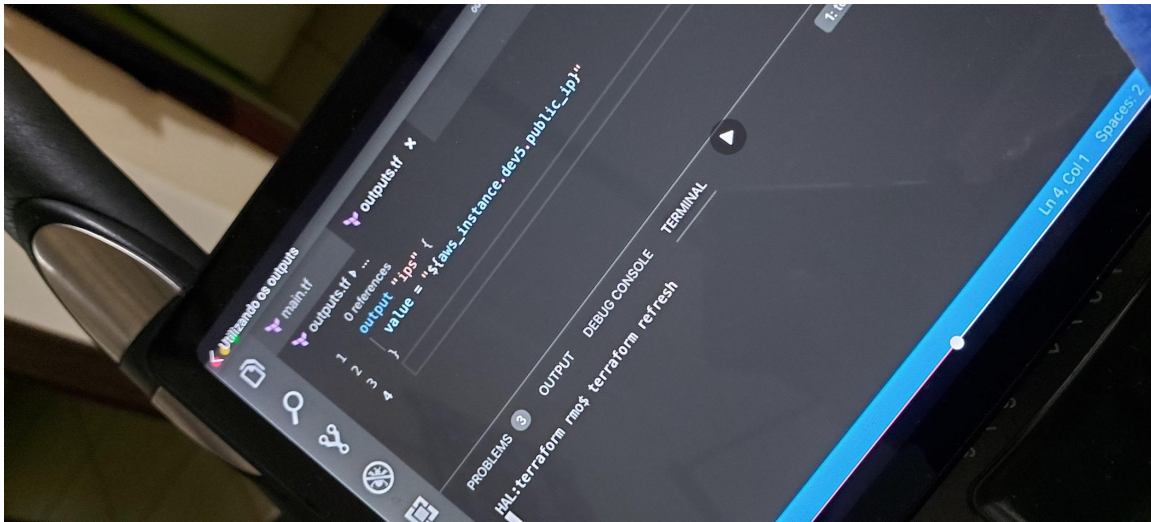
para acessar a maquina:
ssh -i "concrete-terraform" ubuntu@ec2-54-225-22-14.compute-1.amazonaws.com

para apagar as maquinas e os recursos do terraform rodar:
terraform destroy

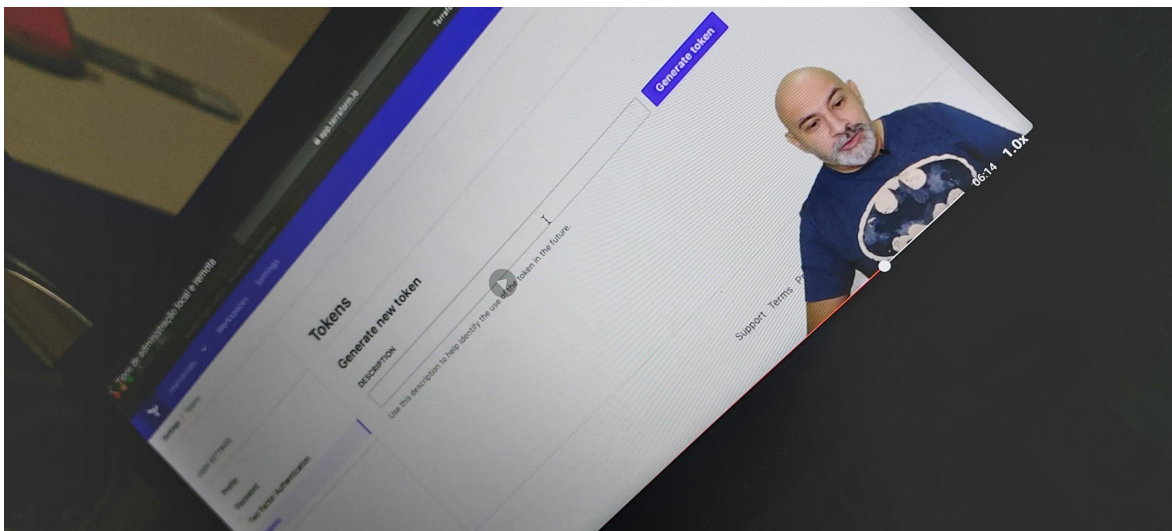
Repo terraform professor
https://github.com/Didox/terraform_concrete_lab

Terraform refresh

Para listar o ips das instâncias no output



É possível trabalhar com terraforms remoto, fazendo o cadastro no terraform e pegando o token



instalar ansible

https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-on-macos

<https://hvops.com/articles/ansible-mac-osx/>

Git actions - Curso

https://www.youtube.com/watch?v=R8_veQiYBjI

Pipeline Jenkins File

<https://github.com/Didox/validadorcpfjava/blob/main/jenkinsfile>

Scripts Terraform + Ansible + WordPress

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "maquina_wp" {
  ami = "${var.amis["us-east-1-ubuntu18"]}"
  instance_type = "${var.instance_type.micro}"
  key_name = "didox-terraform-key"
  tags = {
    Name = "maquina_ansible_com_worpress"
  }
  vpc_security_group_ids = ["${aws_security_group.acesso_ssh.id}"]
}

resource "aws_security_group" "acesso_ssh" {
  name      = "acesso_ssh"
  description = "Treinamento concrete com acesso a SSH"
  ingress {
    description = "Para acessar a maquina via ssh"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["187.3.223.136/32"]
  }

  tags = {
    Name = "acesso_ssh"
  }
}

variable "amis" {
  default = {
    "us-east-1-ubuntu18" = "ami-013f17f36f8b1fefb"
    "us-east-1-ubuntu20" = "ami-042e8287309f5df03"
    "us-east-2-ubuntu20" = "ami-08962a4068733a2b6"
  }
}

variable "instance_type" {
  default = {
    "micro" = "t2.micro"
  }
}
```

```
# arquivo hosts
[ec2-wordpress]
ec2-54-234-131-240.compute-1.amazonaws.com
```

```
# provisionar.yml
- hosts: all
  tasks:
    - shell: "echo 'escrevendo algo no servidor' >> teste.txt"
```

```
# rodar playbook no ansible
ansible-playbook -i hosts provisionar.yml -u ubuntu --private-key ~/.ssh/terraform-aws
```

```
# acesso ssh maquina
ssh -i ~/.ssh/terraform-aws ubuntu@ec2-54-234-131-240.compute-1.amazonaws.com
```

```
- hosts: all
  tasks:
    - name: "Criando um arquivo a toa no servidor"
      shell: "echo 'escrevendo algo no servidor' >> teste.txt"

    - name: "Criando um arquivo a toa no servidor"
      apt:
        name: "apache2"
        state: latest
        become: yes # permite rodar como sudo
```

```

resource "aws_security_group" "acesso_geral" {
  name      = "acesso_geral"
  description = "Treinamento concrete com acesso a geral"
  ingress {
    cidr_blocks    = [
      "0.0.0.0/0",
    ]
    description     = "Para acessar a maquina com todas as portas"
    from_port       = 0
    ipv6_cidr_blocks = [
      "::/0",
    ]
    prefix_list_ids = []
    protocol        = "-1"
    security_groups = []
    self            = false
    to_port         = 0
  }

  egress {
    cidr_blocks    = [
      "0.0.0.0/0",
    ]
    description     = ""
    from_port       = 0
    ipv6_cidr_blocks = [
      "::/0",
    ]
    prefix_list_ids = []
    protocol        = "-1"
    security_groups = []
    self            = false
    to_port         = 0
  }

  tags = {
    Name = "acesso_geral"
  }
}

```

```
- hosts: all
tasks:
  - name: "Criando um arquivo a toa no servidor"
    shell: "echo 'escrevendo algo no servidor' >> teste.txt"
```

```
- name: "Instala pacotes"
  apt:
    name: "{{ item }}"
    state: latest
  become: yes # roda com sudo
  with_items:
    - php7.2
    - apache2
    - libapache2-mod-php7.2
    - php-mysql
    - php-curl
    - mysql-server
    - python3-pycurl
    - python3-mysqldb
```

```
- hosts: all
handlers:
  - name: "restart apache"
    service:
      name: apache2
      state: restarted
    become: yes
tasks:
  # - name: "Criando um arquivo a toa no servidor"
  #   shell: "echo 'escrevendo algo no servidor' >> teste.txt"

- name: "Instala pacotes"
  apt:
    name: "{{ item }}"
    state: latest
  become: yes # roda com sudo
```

```

with_items:
- php7.2
- apache2
- libapache2-mod-php7.2
- php-mysql
- php-curl
- mysql-server
- python3-pycurl
- python3-mysqldb

- name: "Fazendo download do wordpress"
  get_url:
    url: "https://br.wordpress.org/latest-pt_BR.tar.gz"
    dest: "/tmp/wordpress.tar.gz"

- name: "Fazendo descompactação dos arquivos do WP"
  unarchive:
    src: "/tmp/wordpress.tar.gz"
    dest: "/var/www"
    remote_src: yes
    become: yes

- name: "Trocando pasta do render do apache para o wordpress"
  copy:
    src: "apache/000-default.conf"
    dest: "/etc/apache2/sites-available/000-default.conf"
  become: yes
  notify:
    - "restart apache"

```

apache/000-default.conf

<VirtualHost *:80>

```

# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.

```

```
#ServerName www.example.com
```

```
ServerAdmin webmaster@localhost
```

```
DocumentRoot /var/www/wordpress
```

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
```

```
# error, crit, alert, emerg.
```

```
# It is also possible to configure the loglevel for particular
```

```
# modules, e.g.
```

```
#LogLevel info ssl:warn
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
# For most configuration files from conf-available/, which are
```

```
# enabled or disabled at a global level, it is possible to
```

```
# include a line for only one particular virtual host. For example the
```

```
# following line enables the CGI configuration for this host only
```

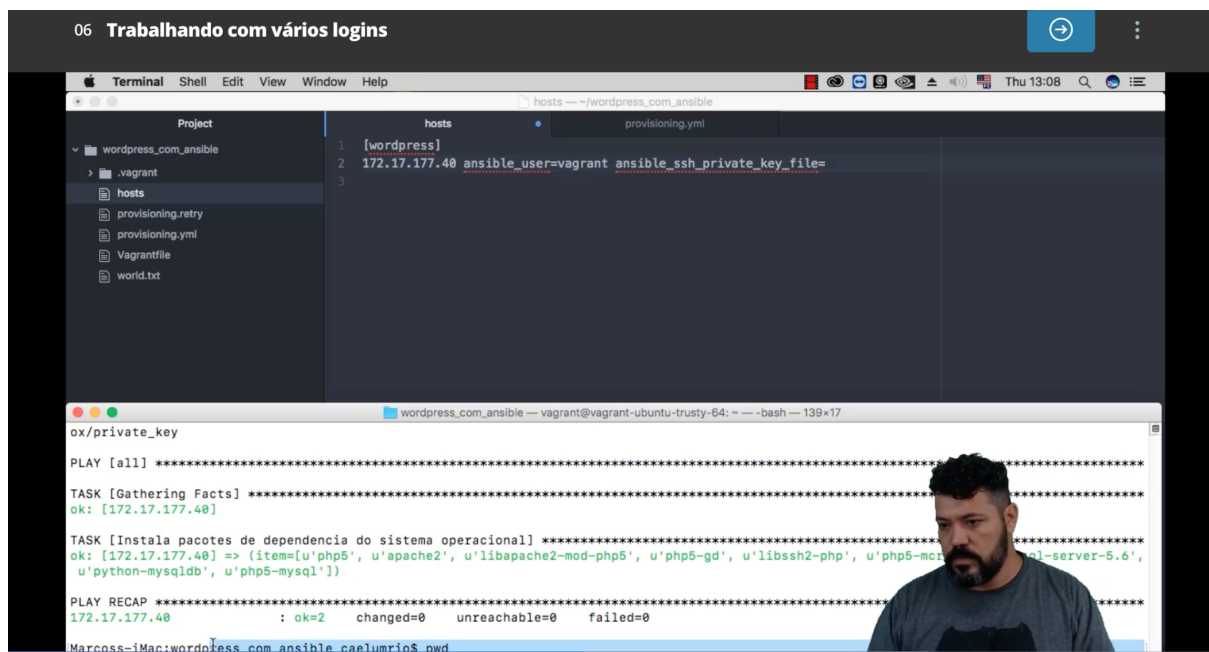
```
# after it has been globally disabled with "a2disconf".
```

```
#Include conf-available/serve-cgi-bin.conf
```

```
</VirtualHost>
```

https://github.com/Didox/terraform_ansible_wordpress_aws

Posso configurar o arquivo de host sem ter que passar na linha de comando o usuário



cada máquina com sua config

