# Guia do projeto

## VPC

Bom meu primeiro passo foi criar uma vpc para ter um ambiente seguro e configurável com total controle sobre ip, sub redes tabelas e afins, loguei na conta aws fui na sessão de vpc através da barra de pesquisa, cliquei em vpc, criar vpc selecionei a opção vpc e muito mais, dei o nome de “projeto” determinei o ip inicial e selecionei 2 zonas 1a/1b e selecionei 2 sub redes publicas e duas privadas configurei o ip e criei a vpc.

## Security groups

Logo em seguida comecei a criar os grupos de segurança indo na área ec2 e buscando a opção rede e segurança, grupos de segurança nomeei como rds-sg, dei uma descrição, selecionei a vpc projeto (a que criei) e coloquei como regra de entrada o mysql/aurora com o ip da vpc para somente quem esteja na minha vpc possa acessar a minha rede, logo em seguida comecei a criar o grupo do efs selecionando minha vpc e aplicando como regra de entrada o nfs selecionando o ip da minha vpc, logo em seguida o grupo de segurança da rede pública, nomeando como ec2 publica. Selecionando minha vpc e adicionando 5 regras de entrada, a do ssh, http, https, mysql, nfs porém as duas últimas só dão acesso a quem estiver na minha rede, criei meu grupo de segurança da rede privada nomeando como ec2-privada, selecionando minha vpc e adicionando 5 regras de entrada ssh, http, https, mysql e nfs.

## EC2 Pública

Agora que tenho a base para criação do projeto irei criar uma instância pública em ec2, executar instância, adicionei as tags do projeto em nome e tags, na imagem selecionei o ubuntu, selecionei a instância t2.micro e criei um par de chaves, selecionei a vpc projeto a qual criei, sub rede pública 1a e adicionei o grupo de segurança que criei para esse modelo de instância assim clicando em executar.

## Movendo a chave .PEM

Entrei no meu terminal ubuntu e movi a chave do sistema Windows para Linux com o código:

```bash

/mnt/c/users/user/download

cp wordkeynew1.pem ~/wordkeynew1.pem

E dei um cd para voltar mais ls para verificar a chave e um ls –l para verificar as permissões. Tive que usar o código sudo chmod 400 para habilitar a chave para o sistema, pois o ssh não trabalha com chaves duvidosas. Para acessar a instância dentro do meu terminal ubuntu utilizei o código:

bash

Copiar código

ssh -i wordkeynew1.pem ubuntu@54.174.136.10

Que contém a chave de segurança e o IP da ec2 pública. Assim que acessei a ec2, dei o comando:

bash

Copiar código

sudo apt update + sudo apt upgrade

**EFS**

Logo em seguida fui para a criação do efs na aba efs da aws e cliquei em criar sistema de arquivos, nomeei com efs, adicionei minha vpc como opção e fui para customizar, adicionando o grupo de segurança que criei para efs e selecionando a sub rede privada para que somente eu tenha acesso a esses arquivos. Finalizei e fui em criar. Para conseguir ter acesso ao efs no terminal tive que instalar um pacote:

bash

Copiar código

sudo apt install nfs-common

Agora iremos anexar o efs no ec2 através da opção anexar. Copiei o código, dei o comando:

bash

Copiar código

sudo mkdir /efs

E logo em seguida adicionei o código:

bash

Copiar código

sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0b20b5769e43c6546.efs.us-east-1.amazonaws.com:/ /efs

Utilizei o comando df –h para verificar se foi montado.

**RDS**

Logo em seguida parti para a criação do rds. Selecionei o mysql e escolhi a db.t3, alterei o nome e escolhi a minha vpc como opção. Selecionei o grupo de segurança que criei para o rds e dei nome ao usuário assim finalizando o rdb. Logo em seguida fui para o terminal instalar o docker pelo comando:

bash

Copiar código

sudo apt install docker.io –y

Utilizei o comando:

bash

Copiar código

sudo systemctl start docker

sudo systemctl enable docker

Para garantir que o docker seja iniciado. Utilizei o comando:

bash

Copiar código

sudo vim docker-compose.yaml

Para criar um arquivo yaml e utilizei o código:

yaml

Copiar código

services:

wordpress:

image: wordpress

restart: always

ports:

- "8080:80"

environment:

WORDPRESS\_DB\_USER: xxxxxxxx

WORDPRESS\_DB\_PASSWORD: xxxxxx

WORDPRESS\_DB\_NAME: wordpress

WORDPRESS\_DB\_HOST: projeto.ctk4oumeyvuf.us-east-1.rds.amazonaws.com

volumes:

- /efs/site:/var/www/html

Dei um :wq pra fechar e verifiquei com o comando cat. Logo em seguida utilizei o comando:

bash

Copiar código

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

Para instalar o docker compose. Logo em seguida entrei na pasta:

bash

Copiar código

cd /efs

E verifiquei se o arquivo estava ali com o comando ls e dei um:

bash

Copiar código

sudo docker-compose up –d

Para subir. Logo em seguida verifiquei com:

bash

Copiar código

docker ps –a

## EC2 Privada

Logo em seguida acessei a área \*\*NAT Gateway\*\* e fui para a criação da instância privada. Utilizei o comando:

```bash

scp -i nomeDaChave.pem nomeDaChave.pem ubuntu@x.x.x.x:/home/ubuntu

Para transferir a chave da minha máquina para a pública. Assim que acessei a máquina pública, verifiquei se meu script havia funcionado. Então, logo em seguida, criei o arquivo:

bash

Copiar código

vim docker-compose.yaml

Dei um ls para ver se estava criado e utilizei essa imagem:

bash

Copiar código

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

Para baixar o Docker Compose. Logo em seguida utilizei o comando:

bash

Copiar código

sudo chmod +x /usr/local/bin/docker-compose

Para dar permissões para o binário. Verifiquei se o Docker Compose foi instalado corretamente com o comando:

bash

Copiar código

docker-compose –version

Utilizei o comando:

bash

Copiar código

sudo docker-compose up –d

Para rodar o programa e um:

bash

Copiar código

docker ps

Para conferir.

Adiante, criei um script para usar como userdata. Reconfigurei uma nova instância privada e, em opções avançadas, no final, coloquei o seguinte código:

bash

Copiar código

#!/bin/bash

# Atualiza os pacotes do sistema

sudo apt update -y && sudo apt upgrade -y

# Instala pacotes necessários

sudo apt install -y apt-transport-https ca-certificates curl software-properties-common

# Adiciona a chave GPG oficial do Docker

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

# Adiciona o repositório do Docker informando para o apt onde buscar os pacotes

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Atualiza novamente os pacotes para incluir o repositório do Docker

sudo apt update -y

# Instala o Docker CE (Community Edition)

sudo apt install -y docker-ce docker-ce-cli containerd.io

# Inicia e habilita o serviço Docker para iniciar no boot

sudo systemctl start docker

sudo systemctl enable docker

# Adiciona o usuário ao grupo docker para evitar uso do sudo

sudo usermod -aG docker ubuntu

sudo mkdir -p /efs

sudo apt update -y && sudo apt upgrade -y

sudo apt-get install -y nfs-common

sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0b20b5769e43c6546.efs.us-east-1.amazonaws.com:/ /efs

sudo mkdir docker-arquivo

cd docker-arquivo

cat << EOF > docker-compose.yaml

services:

wordpress:

image: wordpress

restart: always

ports:

- 80:80

environment:

WORDPRESS\_DB\_USER: admin

WORDPRESS\_DB\_PASSWORD: xxxxxxx

WORDPRESS\_DB\_NAME: xxxxxxxxx

WORDPRESS\_DB\_HOST: mysql.ctk4oumeyvuf.us-east-1.rds.amazonaws.com

volumes:

- /efs/site:/var/www/html

EOF

docker compose up –p

Ativei o banco de dados na instância privada através dos comandos:

bash

Copiar código

sudo apt update

sudo apt install mysql-client

mysql –h endpoint –u

## Load Balancer

Logo em seguida, fui para a criação do **Load Balancer** para fazer essa conexão do bastion host com a EC2 privada. Nas configurações, atribuí minha VPC, DNS e adicionei 2 sub-redes privadas (1a/1b). Escolhi o grupo de segurança criado para o LB e finalizei.

Aguardei as instâncias de destino entrarem em serviço e, logo após, copiei o DNS do Load Balancer e joguei no navegador adicionando ao final do link "/wp-admin". Assim que apareceu a tela inicial do WordPress, acessei com o login criado pelo banco de dados.

## Auto Scaling

Para a criação do Auto Scaling, deixei pronto um modelo de execução da minha EC2 privada. Logo após:

1. Selecionei grupos de Auto Scaling e cliquei em **Criar Grupo**.
2. Nomeei, escolhi o modelo de execução e a versão latest.
3. Adicionei minha VPC e as sub-redes privadas.
4. Anexei ao balanceador de carga existente.
5. Configurei:
   * **Capacidade desejada**: 2.
   * **Mínima**: 2.
   * **Máxima**: 4.
6. Finalizei a criação.

Aguardei alguns minutos, acessei as EC2 privadas geradas pelo Auto Scaling e realizei alguns testes. O script estava funcionando normalmente.