

DOCUMENTAÇÃO DO PROJETO DE INFRAESTRUTURA1

TURMA 2 – SOÁRI CORRÊA DE CARVALHO

PASSO A PASSO:

Antes de começar a detalhar o projeto, tive que recriar tudo novamente, devido a vm_002-infptaluno41 ao ser encerrada e recriada deu problema por estar conectada ainda ao VPC padrão Default_CheckPoint_DH-vpc.

a) Entrar na conta criada pela Digital House na AWS e colocar a senha;

b) Primeiro, vamos criar as duas instâncias: vm_001-infpt2aluno41 e vm_003-infpt2aluno41 (criada uma nova instância, devido o problema apresentado acima)

- 1- Escolher a região de Ohio;
- 2 - Colocar em Pesquisa “EC2” e dá enter (ou clique no ícone laranja na barra;
- 3 - Clicar em “Executar instância” e mais uma vez;
- 4 - Escreva o nome da primeira instância a ser criada **vm_001-infpt2aluno41** na página “**Iniciar uma Instância**” e escolha depois o Sistema Operacional UBUNTU na imagem de máquina da Amazon (AMI);
- 5 - Verificar se está “Qualificado para o nível gratuito”;
- 6 - Em “**Tipo de Instância**”, deixar “Qualificado para o nível gratuito”;
- 7 - Em “**Par de Chaves**”, vamos clicar em “Criar um novo par de chaves” (somente criaremos esta vez, depois caso queira criar novas instâncias, use o par de chaves criado na primeira vez);
- 8 - Agora vamos criar um nome para o par de chaves que foi : **pc-infpt2aluno41** na página “**Criar par de chaves**”;
- 9 - Vamos deixar marcado em “**Tipo de par de chaves**”: “**RSA**”;
- 10 - Em “**Formato de arquivo de chave privada**”, vamos deixar marcado “**.pem**”;
- 11 - Agora clique em “**Criar par de chaves**” em laranja (neste momento ele vai pedir p/você armazenar seu par de chaves criado em um local que você possa depois manipulá-lo. Eu coloquei na Área de Trabalho. Detalhe o meu ficou em formato de Bloco de Notas)
- 12 - Agora em “**Configurações de rede**”, clique em “**Editar**”. Em “VPC”, escolha “a que tiver (Default_DH_CheckPoint-vpc) e a “Sub-rede” é automático sendo a escolhida a “Default_DH_CheckPoint-subnet-public1-us-west-1a”. Em “**Atribuir IP público automaticamente**”, clique na seta e mude para “**Habilitar**”. E deixe como está em “**Atribuir IP IPV6 automaticamente**”.
- 13 - Em “**Firewall**”, vamos criar o grupo de segurança (detalhe caso já tenha é só clicar em “Selecionar grupo de segurança já existente”), portanto coloque o nome de segurança - obrigatório: **gs-infpt2aluno41**;
- 14 - Em “**Regras do grupo de segurança de entrada**”, clique antes, lá embaixo em “Adicionar regra de grupo de segurança” e vai aparecer “Regra de grupo de segurança

2" depois em **"Tipo"** clique na seta e mude para **"HTTP"** e em **"Tipo de origem"** clique na seta e mude para **"Qualquer lugar"**;

15 - Após alterar a regra do grupo de segurança de entrada, vá para **"Executar instância"** em laranja e verá abrir uma outra janela carregando a execução da instância. Aguarde até o final, quando ficar uma mensagem em verde de **"Êxito"**;

16 - Agora, na mesma janela, clique em **"Instâncias"**, lá em cima, para ver como está sua instância ou instâncias criadas. Agora verá assim:

Instâncias (1/33) Informações

Conectar

Estado da instância ▼

Ações ▼

Executar instâncias

Q

Localizar instância por atributo ou tag (case-sensitive)

<

1

>

<div></div>	Name ▼	ID de instância	Estado da inst... ▼	Tipo de inst... ▼	Verificação de s...	Status do al... ▼	Zona de dispo
<div><div></div></div>	vm_001-infpt...	i-05b330043103c01c0	<div><div></div>Interrompido</div>	t2.micro	-	Sem alar...	us-west-1a
<div><div></div></div>	vm_001-infpt...	i-07520b24efcee5e9e	<div><div></div>Interrompido</div>	t2.micro	-	Sem alar...	us-west-1a
<div><div></div></div>	VM_001-profe...	i-07d14413d9719b8a8	<div><div></div>Executando</div>	t2.micro	2/2 verificações a	Sem alar...	us-west-1a
<div><div></div></div>	vm_001-infpt...	i-0970f7134b4b2c91e	<div><div></div>Executando</div>	t2.micro	2/2 verificações a	Sem alar...	us-west-1a
<div><div></div></div>	infpt2-larissar1	i-00cf0556556dd0584	<div><div></div>Interrompido</div>	t2.micro	-	Sem alar...	us-west-1b

17 - Repita os passos do item b.1) até b.16) para criar a outra instância :
"vm_002-infpt2aluno41" e use o mesmo par de chaves (**pc-infpt2aluno41**) e posso usar o mesmo grupo de segurança, porém criei outro (**gs1-infpt2aluno41**);

c) Segundo, vamos criar o grupo de destino (seria o terceiro a ser criado, logo após a criação das instâncias), já que a rede já foi criada (seria a primeira a ser criada caso não tivesse sido feita):

1 - No menu a esquerda da AWS, vamos até ao item **"Balanceamento de carga"** e vamos clicar embaixo em **"Grupos de destino"**;

2 - Dentro da página principal de grupo de destino, clique em **"Create target group"**(Criar grupo de destino).

3 - Em **"Escolha um tipo de destino"**, escolha **"Instâncias"**, e agora vamos colocar o nome do grupo de destino: **"gd-infpt2aluno41"**, deixo o protocolo **"HTTP"** e a porta **"80"** e venho embaixo e escolho a VPC que quero me conectar, ou seja aquela que criei e escolhi: (Default_DH_CheckPoint-vpc);

4 - Feito isto, não mexer nos outros itens e ir lá embaixo da página e clicar em **"Próximo"** (Next);

5 - Agora, ele entra na página de **"Registrar destinos"**, onde estão as minhas instâncias criadas com as suas respectivas vpcs. Vou então selecionar minhas instâncias e clicar lá embaixo em **"Incluir como pendente abaixo"** (Include as pending below);

6 - Agora elas já vão estar incluídas, como se vê na tela abaixo:

Review targets

Targets (2) Remove all pending

All < 1 > ⚙

Remove	Health status ▾	Instance ID ▾	Name ▾	Port ▾	State ▾	Security groups ▾	Zone ▾	Subnet ID
✕	Pending	i-0235f43516e87d294	vm_002-infpt2aluno41	80	🔄 running	gs1-infpt2aluno41	us-west-1b	subnet-03147a35d0
✕	Pending	i-0970f7134b4b2c91e	vm_001-infpt2aluno41	80	🔄 running	gs-infpt2aluno41	us-west-1a	subnet-0fd5d71a01f

2 pending Cancel Previous Create target group

E só falta clicar em “Criar grupo de destino” (Create target group);

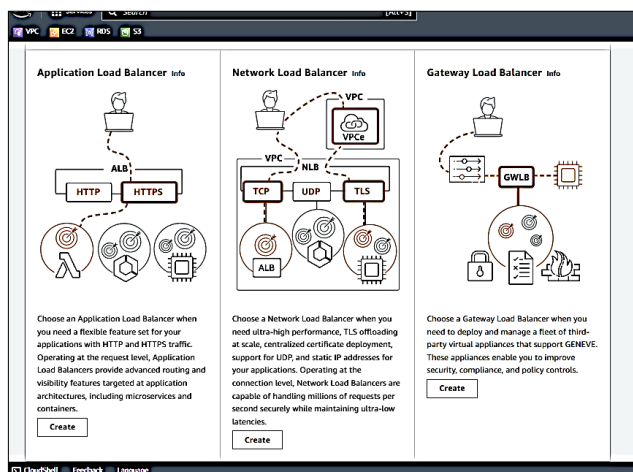
7 - Ao mudar a tela, irei clicar no meu grupo de destino, “ **gd-infpt2aluno41** ”, e lá poderei ir verificando a integridade, pois somente quando ambas as instâncias ficaram ativas (verdes) no “Status de integridade” é que poderei criar meu load balancer;

8 - Já criadas as instâncias, o vpc e os grupos de destino, vamos criar o **load balancer** (meu balanceador de carga);

d) Terceiro, vamos criar o Load Balancer (seria o quarto a ser criado, após o vpc, as instâncias e o grupo de destino).

1 - Vou no canto esquerdo da página, onde tem o menu da AWS, e escolho “ **Load balancers** ” (Balanceador de cargas).

2 - Dentro da página do Balanceador de cargas, cliço em “ **Criar load balancer** “, e irá aparecer nova janela mostrando os “**Tipos de load balancer**” com três aplicações diferentes, e usaremos a mais básica que é a primeira da esquerda para a direita.

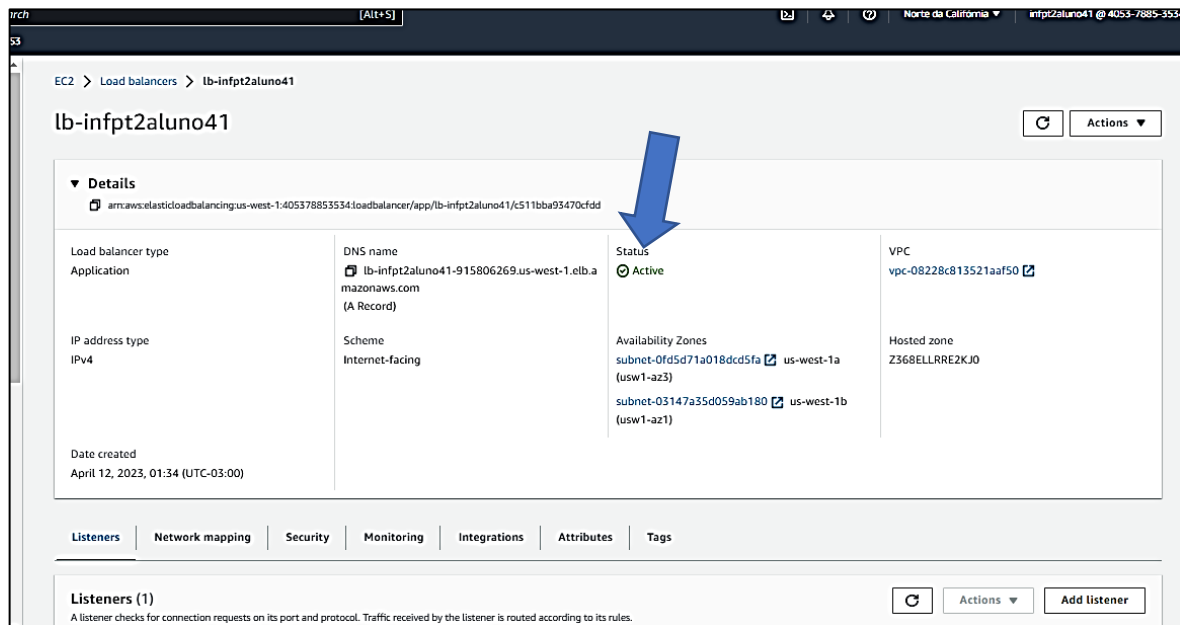


3 - E agora vamos clicar lá embaixo em “Criar”;

4 - Ao entrar na página “**Criar Aplicação Load Balancer**”, em “Configuração básica”, vamos nomear nosso load balancer: “ **lb-infpt2aluno41** ”

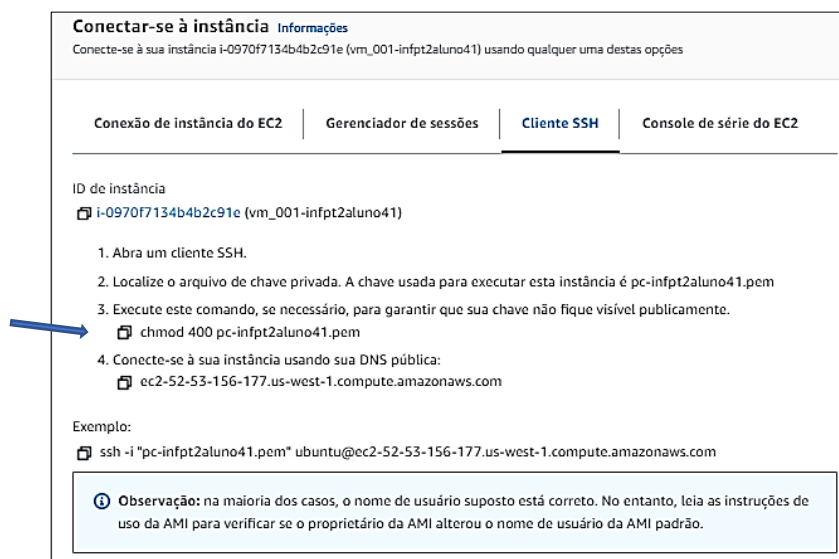
5 - Em “**Esquema**”, deixaremos marcado “ **Voltado para internet** “ e em “ **Tipo de Endereço IP** ”, deixamos marcado IPv4 (tipo de roteador que usamos, mais comum), e em “**Mapeamento de rede**”, vamos usar a nossa VPC (Default_DH_CheckPoint-vpc).

- 6 - Em “Mapeamentos”, vamos escolher as zonas a serem marcadas e em “Sub-rede” a nossa VPC, (Default_DH_CheckPoint-vpc).
- 7 - Em “Grupos de Segurança” os grupos de segurança criado: **gs-infpt2aluno41 (instância 1)** e **gs1-infpt2aluno41 (instância 2)**
- 8 - Em “Listeners e roteamento”, com protocolo “http” e porta “80” já marcadas, vamos em “Ação padrão” colocar nosso grupo de destino, **gd-infpt2aluno41** .
- 9 - Se tudo estiver certo, sem nenhum aviso ou alerta de erro, clicar “**Criar Load Balancer**”.
- 10 - Após, clique no load balancer criado que está em azul e clique em atualizar para que seja ativado. Ele ficará assim:

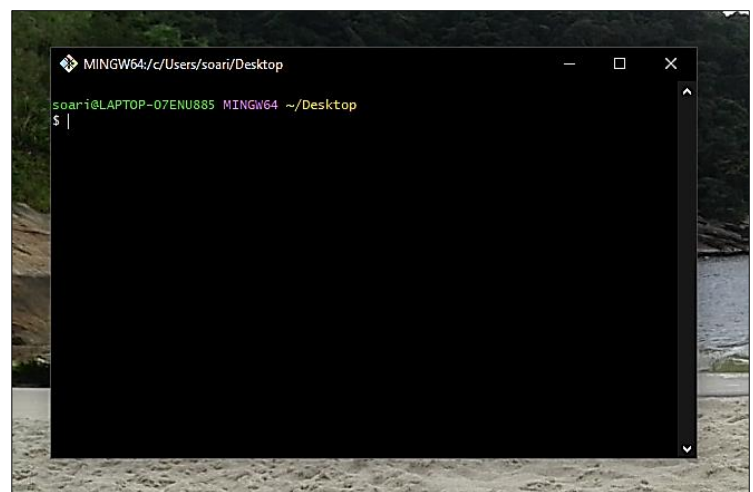
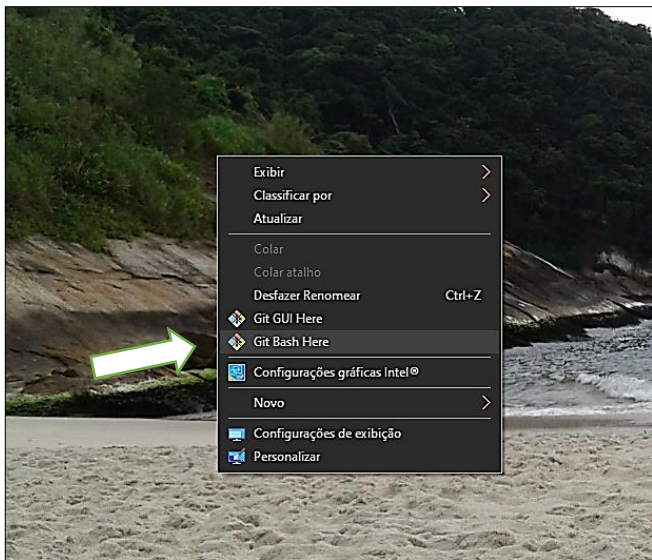


e) Vamos agora no painel de instâncias, clicar na nossa primeira instância, e conectá-la. Logo após abrir a janela “Conectar-se à instância”, clique em “**Cliente SSH**”, agora vamos copiar os comandos desta tela no Git Bash:

- 1 - Vamos primeiro copiar o primeiro comando:



2 - Clicando com o mouse direito na nossa Área de Trabalho, vamos colar o comando que copiamos para garantir que a nossa chave de segurança não fique pública, conforme tela abaixo:



3 - Colamos na linha de comando do Git Bash : “ chmod 400 pc-infpt2aluno41.pem ”

4 - Agora, copiamos o segundo comando no Git Bash: “ssh -i "pc-infpt2aluno41.pem" ubuntu@ec2-52-53-156-177.us-west-1.compute.amazonaws.com ”

Conectar-se à instância Informações
Conecte-se à sua instância i-0970f7134b4b2c91e (vm_001-infpt2aluno41) usando qualquer uma destas opções

Conexão de instância do EC2

Gerenciador de sessões

Cliente SSH

Console de série do EC2

ID de instância

i-0970f7134b4b2c91e (vm_001-infpt2aluno41)

1. Abra um cliente SSH.

2. Localize o arquivo de chave privada. A chave usada para executar esta instância é pc-infpt2aluno41.pem

3. Execute este comando, se necessário, para garantir que sua chave não fique visível publicamente.
❏ `chmod 400 pc-infpt2aluno41.pem`

4. Conecte-se à sua instância usando sua DNS pública:
❏ `ssh -i "pc-infpt2aluno41.pem" ubuntu@ec2-52-53-156-177.us-west-1.compute.amazonaws.com`

Comando copiado

`ssh -i "pc-infpt2aluno41.pem" ubuntu@ec2-52-53-156-177.us-west-1.compute.amazonaws.com`

Observação: na maioria dos casos, o nome de usuário suposto está correto. No entanto, leia as instruções de uso da AMI para verificar se o proprietário da AMI alterou o nome de usuário da AMI padrão.

Cancelar

5 - E agora segue os seguintes comandos:

- ⇒ yes
- ⇒ sudo su
- ⇒ apt-get update
- ⇒ apt-get install apache2
- ⇒ y
- ⇒ systemctl status apache2 (Aparecerá esta tela abaixo dizendo que o Apache2 está ativo.

```

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-10-0-8-160:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-04-12 05:19:23 UTC; 2min 34s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 3083 (apache2)
      Tasks: 55 (limit: 1141)
     Memory: 6.5M
        CPU: 44ms
    CGroup: /system.slice/apache2.service
            └─3083 /usr/sbin/apache2 -k start
              └─3085 /usr/sbin/apache2 -k start
                └─3086 /usr/sbin/apache2 -k start

Apr 12 05:19:23 ip-10-0-8-160 systemd[1]: Starting The Apache HTTP Server...
Apr 12 05:19:23 ip-10-0-8-160 systemd[1]: Started The Apache HTTP Server.
root@ip-10-0-8-160:/home/ubuntu#

```

- ⇒ systemctl stop apache2
- ⇒ systemctl status apache2
- ⇒ sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release
- ⇒ y
- ⇒ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/
- ⇒ ^C
- ⇒ sudo apt-get update
- ⇒ sudo apt-get update
- ⇒ sudo apt-get install \
- ⇒ ca-certificates \
- ⇒ curl \
- ⇒ gnupg \
- ⇒ lsb-release -y
- ⇒ sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release
- ⇒ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
- ⇒ echo "deb [arch=amd64 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
- ⇒ sudo apt-get install docker-ce docker-ce-cli containerd.io
- ⇒ sudo mkdir -p /etc/apt/keyrings curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
- ⇒ 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
- ⇒ sudo apt-get update
- ⇒ sudo apt-get install docker-ce docker-ce-cli containerd.io -y
- ⇒ sudo docker run -d -p 80:80 httpd
- ⇒ apt-get update
- ⇒ docker version
- ⇒ sudo apt-get install docker-ce docker-ce-cli containerd.io
- ⇒ y
- ⇒ git clone https://github.com/SoariCarvalho/INFRA2-.git
- ⇒ chmod 777 -R INFRA2-/
- ⇒ cd INFRA2-/
- ⇒ ls -lia
- ⇒ nano Dockerfile

- ⇒ ls
- ⇒ docker build -t intancia3 .
- ⇒ docker run -p 80:80 -d intancia3

- f) Agora fazemos o mesmo para outra instância, criando outro container de imagem do Docker, conforme passos e.1) até e.5)**
- g) Após colocarmos nossos dois sites nas portas de entrada 80, copiando de ambas as instâncias seu DNS público, podemos fazer o load balancer clicando em atualizar a página da porta 80, protocolo HTTP.**