

# LAB 175: Usando Auto Scaling na AWS (Linux)

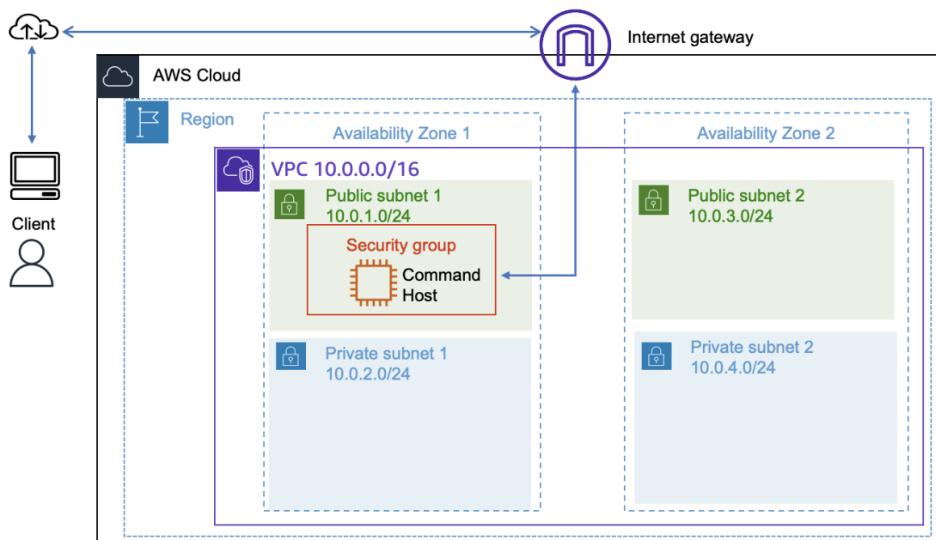
## Visão geral do laboratório

Neste laboratório, você utilizará a **AWS Command Line Interface (AWS CLI)** para:

1. Criar uma instância **Amazon EC2** que hospeda um servidor web.
2. Criar uma **Amazon Machine Image (AMI)** a partir dessa instância.
3. Utilizar essa AMI como base para lançar uma estrutura de **Auto Scaling**, que se ajusta automaticamente à carga variável.
4. Criar um **Elastic Load Balancer (ELB)** para distribuir o tráfego entre as instâncias EC2 criadas em múltiplas Zonas de Disponibilidade (Availability Zones).

## Arquitetura inicial:

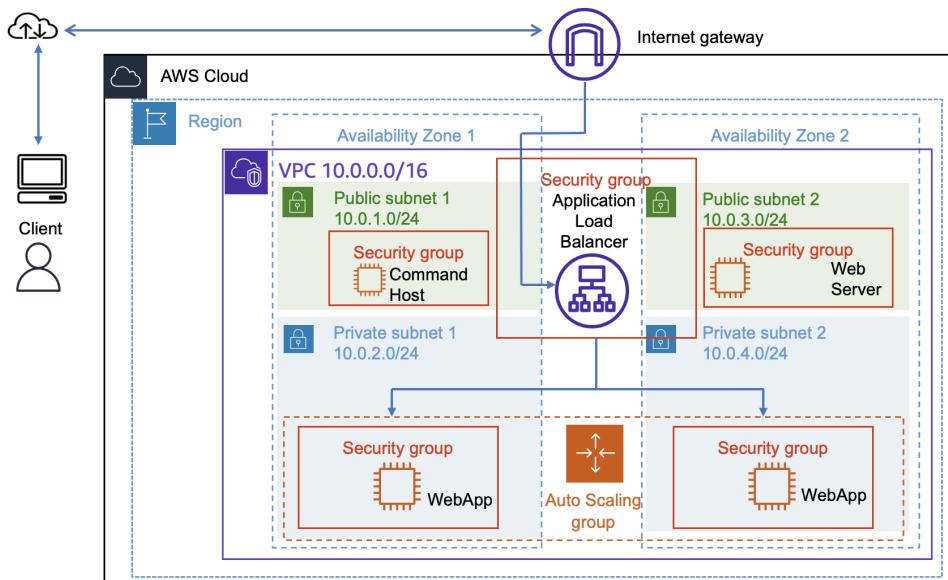
A arquitetura inicial mostra uma infraestrutura na AWS com uma instância **Command Host** em uma **sub-rede pública**.



## Arquitetura final:

A arquitetura final mostra:

- Um **Elastic Load Balancer**
- Instâncias EC2 em um **grupo de Auto Scaling**
- Sub-redes **privadas**
- Zonas de Disponibilidade diferentes



## Objetivos

Após concluir este laboratório, você será capaz de:

- Criar uma instância EC2 usando a AWS CLI
- Criar uma AMI personalizada
- Criar um **Launch Template**
- Configurar um grupo de **Auto Scaling**
- Definir políticas de escala automática com base na utilização de CPU

# Duração

Este laboratório leva aproximadamente **45 minutos** para ser concluído.

## Acessando o Console de Gerenciamento da AWS

1. No topo da página, clique em **Start Lab** para iniciar o laboratório.
2. Aguarde até aparecer a mensagem "**Lab status: ready**", depois feche o painel com X.

```
Start Lab X
Region: us-west-2
Lab ID: arn:aws:cloudformation:us-west-
2:708894503294:stack/c163842a4208284l11685597t1w708894503294/2115b610-9562-11f0-
b59b-0acdcfd2236d
Creation Time: 2025-09-19T07:08:34-0700

Start session at: 2025-09-19 07:08:34
Remaining session time: 03:00:00(180 minutes)

Lab status: ready
```

3. No topo, clique em **AWS** para abrir o Console da AWS em uma nova aba (login feito automaticamente).
4. Caso o navegador bloqueie o pop-up, permita a abertura.
5. Organize as janelas do navegador para visualizar as instruções e o Console simultaneamente.
6. **Importante:** Não altere a região padrão do laboratório, a menos que seja instruído.

## Tarefa 1: Criar uma nova AMI para Auto Scaling

### 1.1 Conectando à instância Command Host

1. No Console da AWS, pesquise e abra o serviço **EC2**.
2. No painel esquerdo, clique em **Instâncias**.
3. Selecione a instância  **Command Host**.
4. Clique em **Conectar > EC2 Instance Connect > Conectar**.

### 1.2 Configurar a AWS CLI

A AWS CLI já está pré-configurada.

Para verificar a região atual da instância execute o comando a seguir:

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
```

Exemplo:

```
[ec2-user@ip-10-0-1-214 ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total   Spent    Left  Speed
100  475  100  475    0      0  350k      0 --:--:-- --:--:-- 463k
  "region" : "us-west-2",
[ec2-user@ip-10-0-1-214 ~]$
```

Para configurar as credenciais da CLI execute o comando a seguir:

```
aws configure
```

Nos prompts:

- **AWS Access Key ID:** pressione **[Enter]**
- **AWS Secret Access Key:** pressione **[Enter]**
- **Default region name:** insira a região exibida anteriormente (ex: **us-west-2**)
- **Default output format:** **json**

Exemplo

```
[ec2-user@ip-10-0-1-214 ~]$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [us-west-2]: us-west-2
Default output format [None]: json
[ec2-user@ip-10-0-1-214 ~]$
```

Depois, acesse o diretório dos scripts, executando o seguinte comando:

```
cd /home/ec2-user/
```

### 1.3 Criar uma nova instância EC2

Visualize o script de inicialização, executando o seguinte comando:

```
less UserData.txt
```

Exemplo

```

#!/bin/bash
yum update -y --security
amazon-linux-extras install epel -y
yum -y install httpd php stress
systemctl enable httpd.service
systemctl start httpd
cd /var/www/html
wget http://aws-tc-largeobjects.s3.amazonaws.com/CUR-TF-100-TULABS-1/10-lab-autoscaling-linux/s3/ec2-stress.zip
unzip ec2-stress.zip

echo 'UserData has been successfully executed.' >> /home/ec2-user/result
find -wholename /root/.history -wholename /home/*/.history -exec rm -f {} \;
find / -name 'authorized_keys' -exec rm -f {} \;
rm -rf /var/lib/cloud/data/scripts/*
UserData.txt (END)

```

**Dica:** Utilize as teclas para navegar no texto e a tecla **[ q ]** para sair do comando **less**

Esse script instalará atualizações e uma aplicação PHP que simula uso intenso de CPU.

No topo da página do Vacareum, clique em **Details > Show**.

Copie os valores: **KEYNAME**, **AMIID**, **HTTPACCESS** e **SUBNETID** para um bloco de notas.

AMIID	ami-02 [REDACTED] 4a
HTTPACCESS	sg-06 [REDACTED] 67
COMMANDHOSTIP	35. [REDACTED] 141
KEYNAME	vockey
SUBNETID	subnet-0f [REDACTED] 21

Atualize e execute os seguintes comandos substituindo os valores somente nos campos de criação das variáveis, **removendo os colchetes [ ]**:

# 4 Comandos para criação das 4 variáveis

AMIID=[ coloque aqui o valor do AMIID ]

HTTPACCESS=[ coloque aqui o valor do HTTPACCESS ]

KEYNAME=[ coloque aqui o valor do KEYNAME ]

SUBNETID=[ coloque aqui o valor do SUBNETID ]

**Dica:** após a entrada de cada variável, não é exibido nada no terminal

Execute o seguinte comando para verificar a criação das variáveis:

# 1 Comando para verificação das 4 variáveis

```
echo "KEYNAME = $KEYNAME"; echo "AMIID = $AMIID"; echo "HTTPACCESS = $HTTPACCESS"; echo
"SUBNETID = $SUBNETID"
```

Execute o comando a seguir para criar a nova instância, obter o ID da nova instância, o endereço da URL do DNS público da nova instância e o endereço URL da página rodando o aplicativo WEB:

```
# Apenas 1 Comando para criação da instância, utilizando as 4 variáveis criadas \
# A saída deste comando exibirá o NEW_INSTANCEID, o PUBLIC_DNS_NAME e o PUBLIC-APP-ADDRESS
```

```
NEW_INSTANCEID=$(aws ec2 run-instances \
--key-name $KEYNAME \
--instance-type t3.micro \
--image-id $AMIID \
--user-data file:///home/ec2-user/UserData.txt \
--security-group-ids $HTTPACCESS \
--subnet-id $SUBNETID \
--associate-public-ip-address \
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=WebServer}]' \
--output text \
--query 'Instances[*].InstanceId'); \
aws ec2 wait instance-running --instance-ids $NEW_INSTANCEID; \
echo "NEW_INSTANCEID = $NEW_INSTANCEID"; \
aws ec2 wait instance-status-ok --instance-ids $NEW_INSTANCEID; \
PUBLIC_DNS_NAME=$(aws ec2 describe-instances --instance-id $NEW_INSTANCEID \
--query 'Reservations[0].Instances[0].NetworkInterfaces[0].Association.PublicDnsName' --output text | tr -d ""); \
echo "PUBLIC_DNS_NAME = $PUBLIC_DNS_NAME"; \
echo "PUBLIC-APP-ADDRESS = http://$PUBLIC_DNS_NAME/index.php"
```

Exemplo

```
[ec2-user@ip-10-0-1-214 ~]$ NEW_INSTANCEID=$(aws ec2 run-instances \
>   --key-name $KEYNAME \
>   --instance-type t3.micro \
>   --image-id $AMIID \
>   --user-data file:///home/ec2-user/UserData.txt \
>   --security-group-ids $HTTPACCESS \
>   --subnet-id $SUBNETID \
>   --associate-public-ip-address \
>   --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=WebServer}]' \
>   --output text \
>   --query 'Instances[*].InstanceId'); \
> aws ec2 wait instance-running --instance-ids $NEW_INSTANCEID; \
> echo "NEW_INSTANCEID = $NEW_INSTANCEID"; \
> aws ec2 wait instance-status-ok --instance-ids $NEW_INSTANCEID; \
> PUBLIC_DNS_NAME=$(aws ec2 describe-instances --instance-id $NEW_INSTANCEID \
>   --query 'Reservations[0].Instances[0].NetworkInterfaces[0].Association.PublicDnsName' --output text | tr -d '''); \
> echo "PUBLIC_DNS_NAME = $PUBLIC_DNS_NAME"; \
> echo "PUBLIC_APP-ADDRESS = http://$PUBLIC_DNS_NAME/index.php"
NEW_INSTANCEID = i-06b017980167f78e2
PUBLIC_DNS_NAME = ec2-54-245-165-65.us-west-2.compute.amazonaws.com
PUBLIC_APP-ADDRESS = http://ec2-54-245-165-65.us-west-2.compute.amazonaws.com/index.php
[ec2-user@ip-10-0-1-214 ~]$
```

Copie os valores exibidos e guarde-os no bloco de notas

Abra uma nova guia no navegador e **acesse a URL do PUBLIC-APP-ADDRESS**.

**ATENÇÃO:** Não clique ainda em "Start Stress".

## 1.4 Criar uma AMI personalizada

Execute o comando utilizando a variável **\$NEW\_INSTANCEID** para criar uma nova imagem AMI:

# 1 Comando para criação de imagem AMI

```
aws ec2 create-image --name WebServerAMI --instance-id $NEW_INSTANCEID
```

A instância pode ser reiniciada durante o processo de criação da AMI.

## Tarefa 2: Criar o ambiente de Auto Scaling

### 2.1 Criar um Application Load Balancer

1. No painel lateral do EC2, vá até **Load Balancing > Load Balancers**
2. Clique em **Create load balancer**
3. Escolha **Application Load Balancer > Create**

Configurações:

- **Load balancer name:** **WebServerELB**
- **Scheme:** **Internet-facing (Voltado para a Internet)**
- **VPC:** **Lab VPC**
- **Availability Zones:** selecione ambas
  - **Sub-rede pública 1**
  - **Sub-rede pública 2**
- **Security groups:** remova o padrão(*default*), selecione **HTTPAccess**

Criar target group:

1. Clique em **Create target group(Criar grupo de destino)**, uma nova aba será aberta.
2. Escolha:
  - Target type: **Instances(Instâncias)**
  - Target group name: **webserver-app**
  - Health check path: **/index.php**
3. Clique em **Next > Create target group**
4. Feche a aba e retorne à aba do Load Balancer
5. Em **Listeners and routing**, atualize e selecione **webserver-app** como target
6. Clique em **Create load balancer**

Copie o **DNS name** do load balancer para uso posterior.

### 2.2 Criar Launch Template

1. No menu do EC2, vá em **Launch Templates > Create launch template**

Configurações:

- **Launch template name:** **web-app-launch-template**
- **Template version description:** **A web server for the load test app**
- Selecione  **Provide guidance to help me...**
- Em **My AMIs**, selecione **WebServerAMI**
- **Instance type:** **t3.micro**
- **Key pair:** **Don't include in launch template**
- **Security groups:** selecione **HTTPAccess**

Clique em **Create launch template**

## 2.3 Criar Auto Scaling Group

1. Selecione [web-app-launch-template](#) > [Actions](#) > [Create Auto Scaling group](#)

Configurações:

- Auto Scaling group name: **Web App Auto Scaling Group**
- **Next**
- VPC: **Lab VPC**
- Sub-redes: selecione as **sub-redes PRIVADAS 1 e 2**
- **Next**

Load balancing:

- Selecione **Attach to an existing load balancer** (*Anexar a um balanceador de carga existente*)
- Escolha **webserver-app | HTTP** como target group
- Ative **Elastic Load Balancing health checks** (*Ative as verificações de integridade do Elastic Load Balancing*)
- **Next**

Group size:

- Desired capacity: **2**
- Minimum: **2**
- Maximum: **4**

Scaling policies:

- Tipo: **Target tracking**
- Métrica: **Average CPU utilization**
- Valor-alvo: **50**
- **Next**

Adicionar notificações - opcional

- **Next**

Adicionar etiquetas

Clique em [Add Tags](#) (*Adicionar etiquetas*)

- Key: **Name**
- Value: **WebApp**
- **Next**

Revise e clique em **Create Auto Scaling group**

## Tarefa 3: Verificar configuração de Auto Scaling

1. Acesse **EC2 > Instâncias**
2. Aguarde até que as instâncias **WebApp** estejam com o status **3/3 checks passed**
3. Vá em **Target Groups**, selecione **webserver-app**, e verifique se os targets estão **healthy**

## Tarefa 4: Testar Auto Scaling

1. Abra uma nova aba do navegador
2. Acesse o **DNS do Load Balancer** copiado anteriormente
3. Na página web, clique em **Start Stress**  
Isso causará um pico de CPU em uma instância, acionando o **scale out**.
4. No Console EC2, vá para **Auto Scaling Groups**
5. Selecione **Web App Auto Scaling Group** > aba **Activity**

Após alguns minutos, você verá uma nova instância sendo criada devido ao aumento no uso de CPU.

## Conclusão

Parabéns! Você realizou com

sucesso:

- ✓ Criação de uma instância EC2 via AWS CLI
- ✓ Criação de uma AMI personalizada
- ✓ Criação de um Launch Template
- ✓ Configuração de um grupo de Auto Scaling
- ✓ Definição de políticas de escala com base na CPU

## Finalizar o laboratório

1. No topo da página, clique em **Submit** e aguarde até a submissão ser completada
2. Clique em **End Lab**
3. Confirme clicando em **Yes**
4. Uma mensagem indicará que os recursos estão sendo encerrados
5. Clique em **X** para fechar o painel