Datos Generales:

Nombre: Tomás Alfredo Villaseca Constantinescu

País: Chile

Fecha: 20/10/2023

Contacto: tomas.villaseca.c@gmail.com

Después de completar este laboratorio, usted podrá ser capaz de hacer lo siguiente:

- Lanzar una instancia EC2 utilizando la AWS Management Console.
- Conectarse a la instancia EC2 mediante EC2 Instance Connect.
- Lanzar una instancia EC2 mediante la AWS CLI.

Resumen Laboratorio:

AWS ofrece varias formas de lanzar una instancia de EC2.

En este laboratorio, utilizará la AWS Management Console para lanzar una instancia EC2.

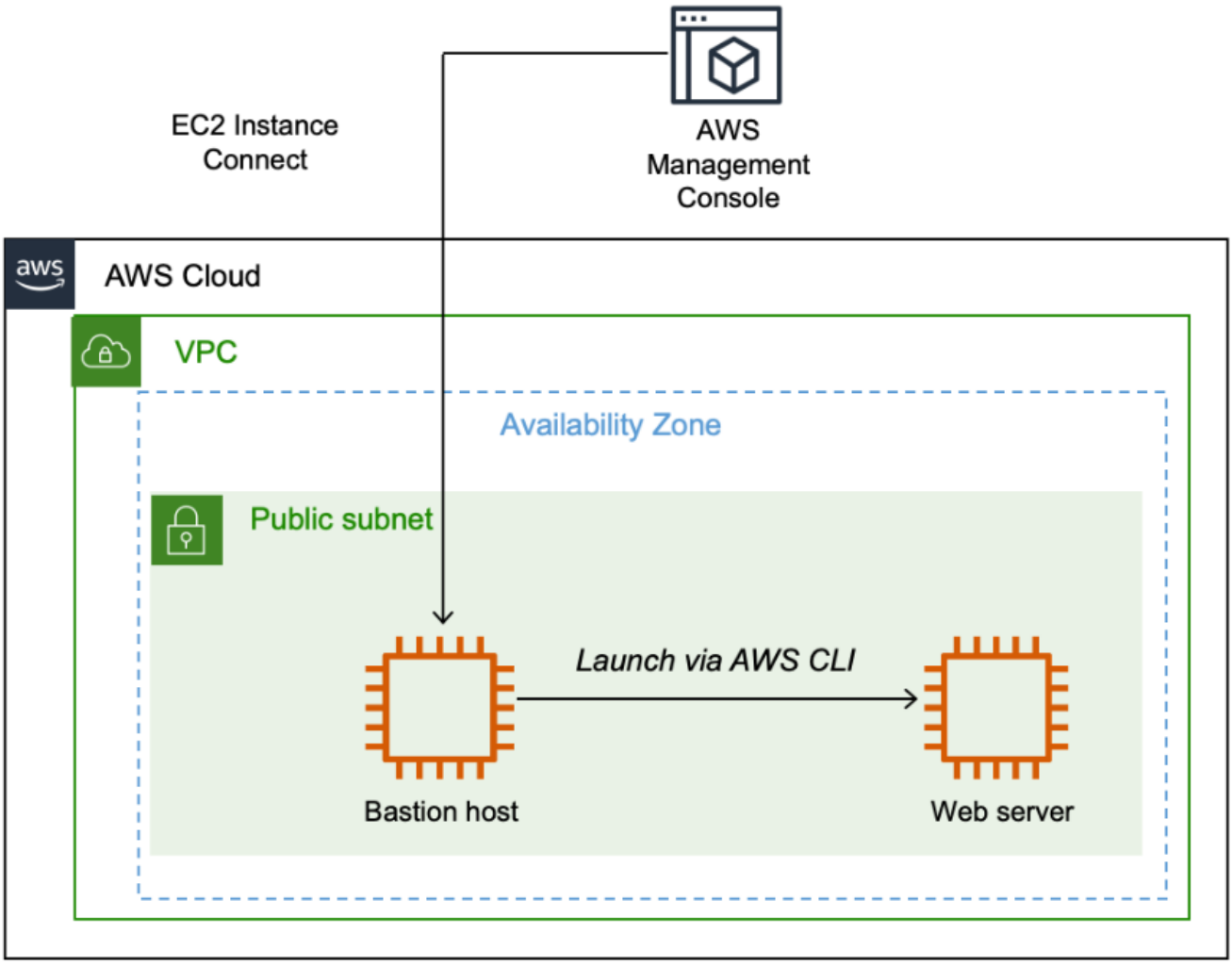
Utilizará la instancia EC2 como Bastion Host para lanzar otra instancia EC2, que será un servidor web.

Se utiliza EC2 Instance Connect para conectarse de forma segura al Bastion Host para luego usar AWS CLI para lanzar una instancia de servidor web.

Bastion Host = Servidor dedicado que está reforzado y configurado para resistir ataques.

- Suele situarse entre la red interna de una organización y la Internet pública.
- Sirve como punto único de entrada para usuarios y administradores remotos.

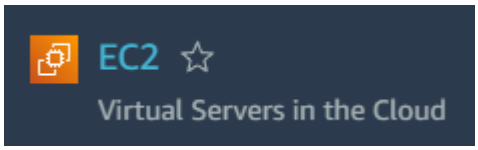
El siguiente diagrama ilustra la arquitectura final que construirá:



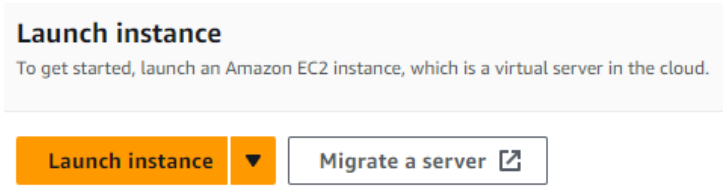
Tarea 1: lanzar una instancia de Amazon EC2 mediante la Management Console

En esta tarea, lanzará una instancia EC2 utilizando la consola de administración de AWS. La instancia será un Bastion Host desde el que podrá utilizar la AWS CLI.

Paso 1: AWS Management Console → Search → EC2

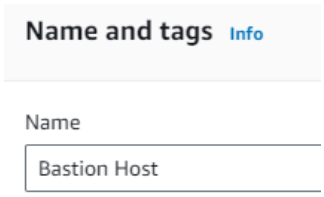


Paso 2: EC2 → Launch Instance



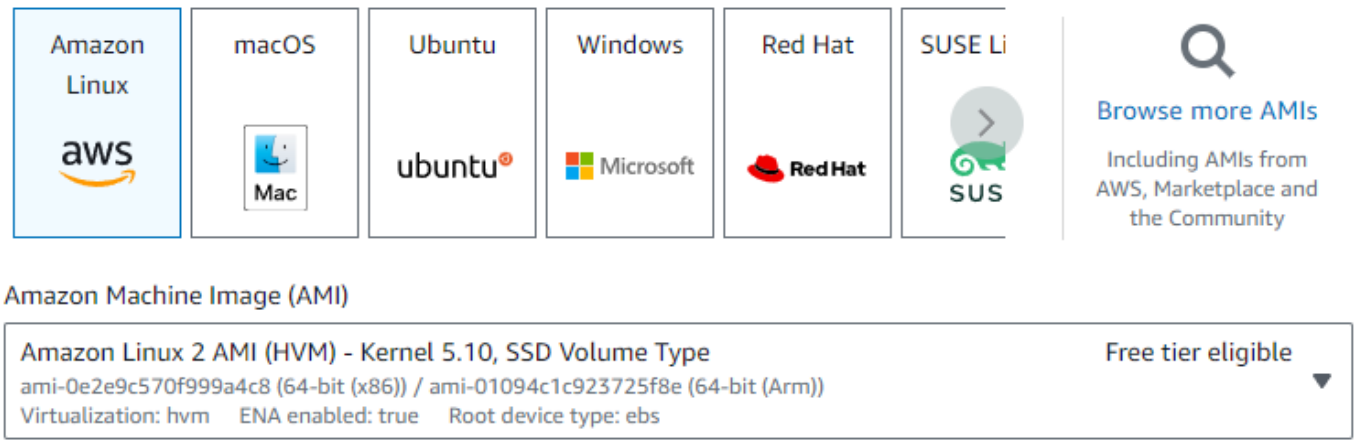
Paso 3: Launch Instance → Name and Tags

- Name = Bastion Host



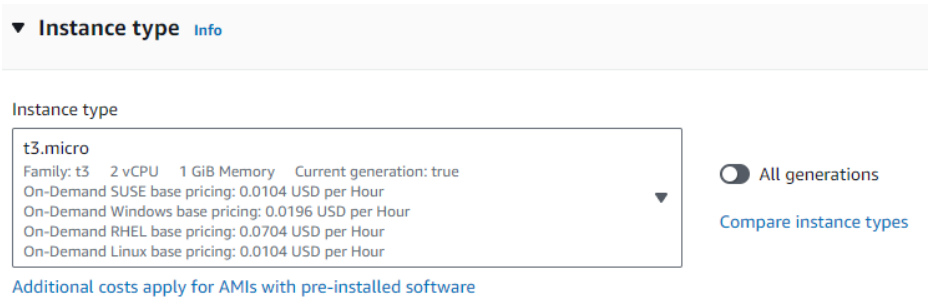
Paso 4: Launch Instance → Choose an AMI

- Seleccionar Amazon Linux 2



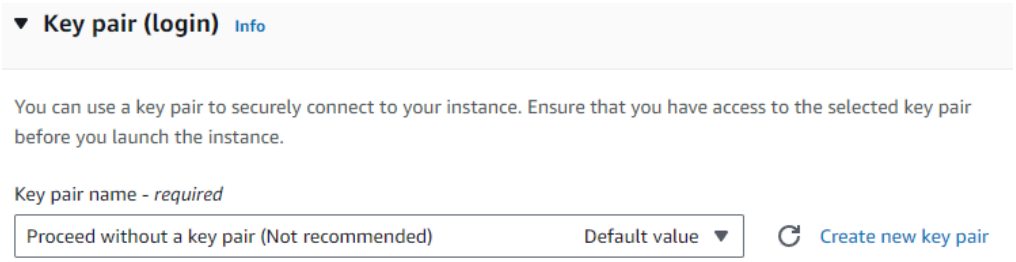
Paso 5: Launch Instance → Instance Type

- Instance type → t3.micro



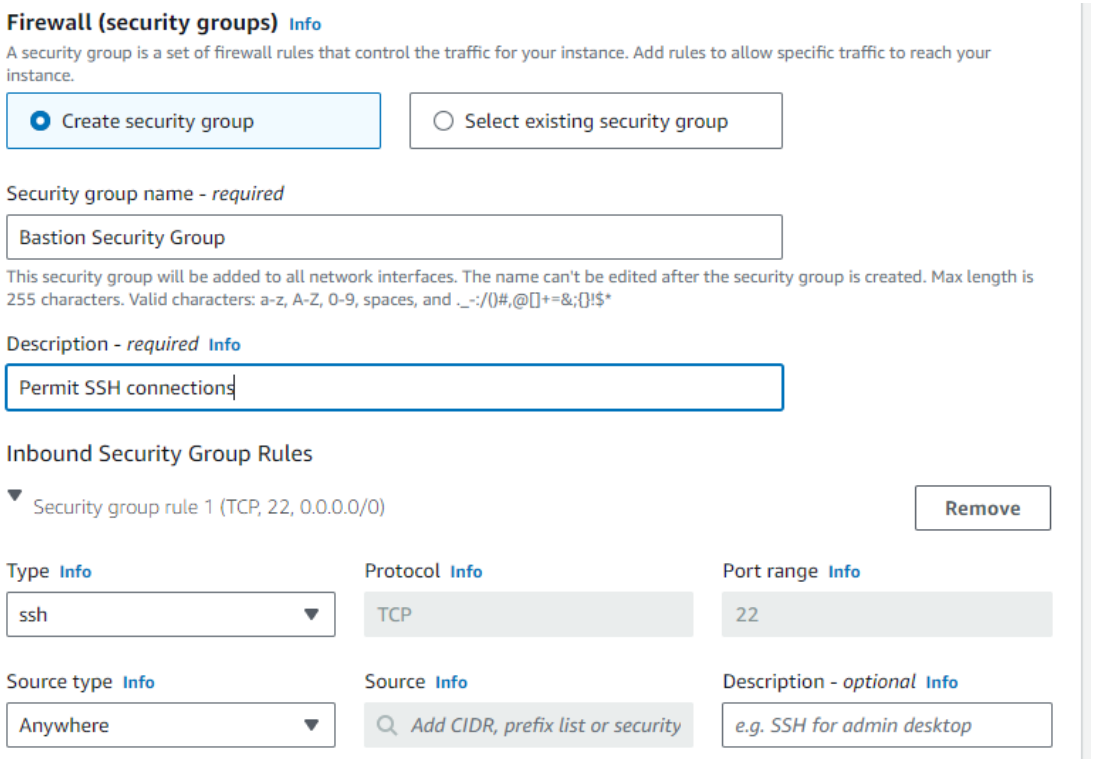
Paso 6: Launch Instance → Key Pair

- Proceed without key pair (not recommended)



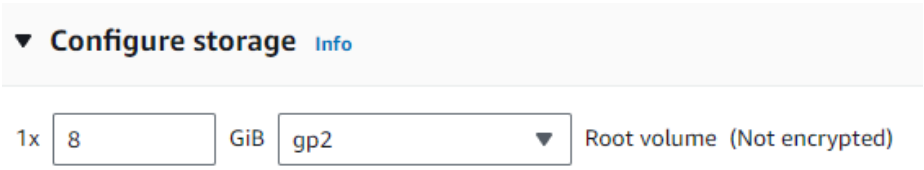
Paso 7: Launch Instance → Configure Network Settings

- VPC → Lab VPC
- Subnet → Default
- Seleccionar “Enable” para “Auto-assign Public IP”
- Security Groups → Create Security Group
- Security Group Name = Bastion Security Group
- Description = Permit SSH connections.



Paso 8: Launch Instance → Add storage

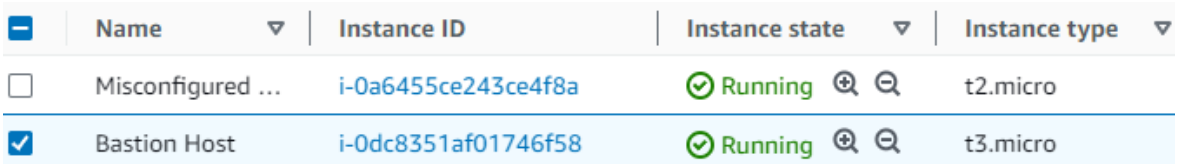
- Configure Storage → Default



Paso 9: Launch Instance → Configure Advanced Details

- IAM Instance Profile → Bastion-Role
- Bastion-Role Profile = Concede permiso a las aplicaciones que se ejecutan en la instancia para realizar solicitudes al servicio de Amazon EC2.

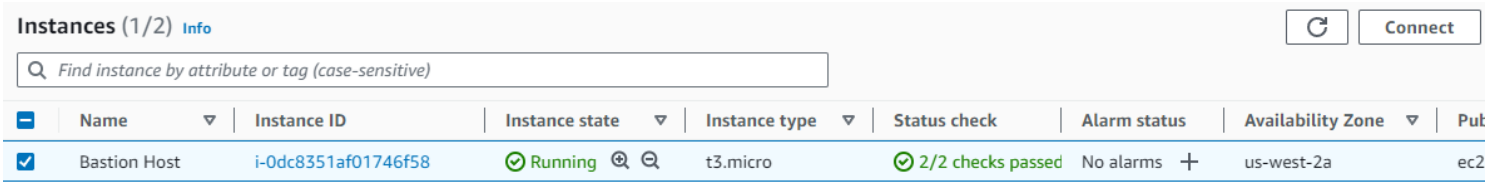
Paso 10: Launch Instance → Launch EC2 Instance

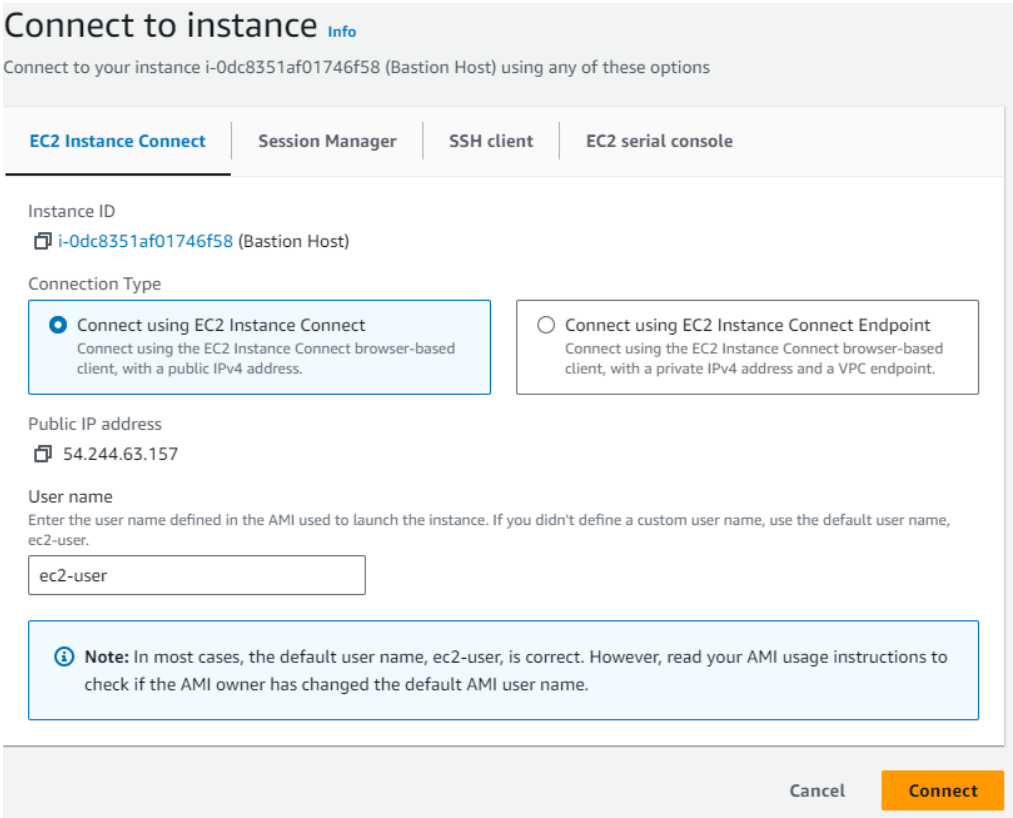


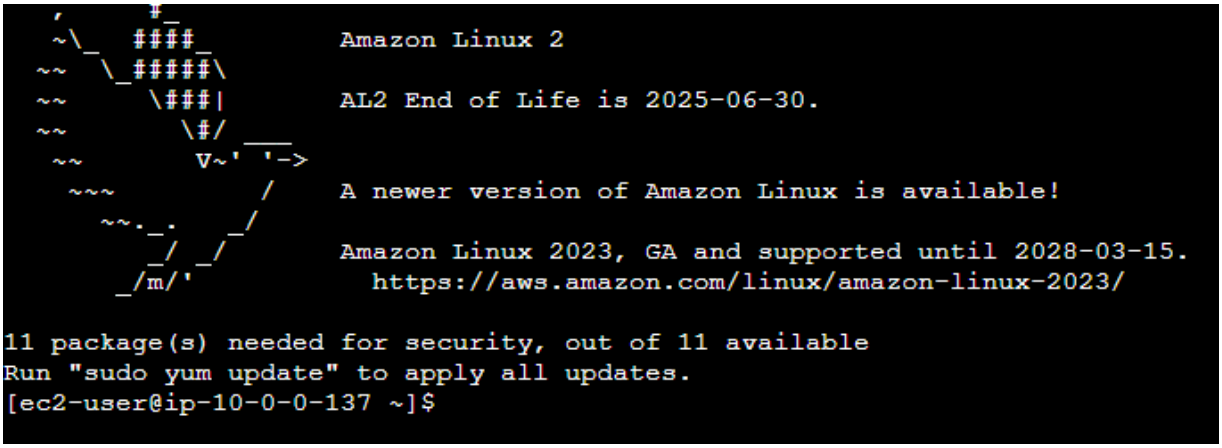
Tarea 2: Iniciar sesión en el servidor bastión

En esta tarea, utilizará EC2 Instance Connect para iniciar sesión en el Bastion Host que acaba de crear.

Paso 1: EC2 → Instances → Bastion Host → Connect







Tarea 3: Lanzar una instancia con la AWS CLI

En esta tarea, se lanza una instancia EC2 utilizando la AWS CLI.

AWS CLI permite automatizar el aprovisionamiento y la configuración de los recursos de AWS.

Cuando se utiliza un comando de la CLI, es necesario suministrar todos los parámetros del comando para ejecutarlo y lanzarlo correctamente.

Paso 1: Utilizar Parameter Store de AWS Systems Manager para recuperar el ID de la AMI Amazon Linux 2 más reciente.

Ejecutar el siguiente script para obtener el ID de la AMI:

```
#Set the Region
AZ=`curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone`
export AWS_DEFAULT_REGION=${AZ::-1}
#Retrieve latest Linux AMI
AMI=$(aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 --query 'Parameters[0].
[Value]' --output text)
echo $AMI
```

- El script recupera la AZ para la instancia en ejecución utilizando metadatos de la instancia.
- El script recupera la región de la AZ y la exporta al entorno para uso posterior.
- El script llama a AWS Systems Manager (ssm) y utiliza el comando **get-parameter** para recuperar la ID de AMI de parameter store.
- La ID de AMI se almacena en una variable llamada AMI.

```
[ec2-user@ip-10-0-0-137 ~]$ #Set the Region
[ec2-user@ip-10-0-0-137 ~]$ AZ=`curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone`
[ec2-user@ip-10-0-0-137 ~]$ export AWS_DEFAULT_REGION=${AZ::-1}
[ec2-user@ip-10-0-0-137 ~]$ #Retrieve latest Linux AMI
[ec2-user@ip-10-0-0-137 ~]$ AMI=$(aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/amzn2-
[ec2-user@ip-10-0-0-137 ~]$ echo $AMI
ami-0025f0db847eb6254
[ec2-user@ip-10-0-0-137 ~]$
```

Paso 2: Ejecutar el siguiente script para recuperar la ID de Subred Pública de la Subred Pública:

```
SUBNET=$(aws ec2 describe-subnets --filters 'Name=tag:Name,Values=Public Subnet' --query Subnets[].SubnetId --output text)
echo $SUBNET
```

```
[ec2-user@ip-10-0-0-137 ~]$ SUBNET=$(aws ec2 describe-subnets --filters 'N
[ec2-user@ip-10-0-0-137 ~]$ echo $SUBNET
subnet-0c6cd623e778f5b2a
[ec2-user@ip-10-0-0-137 ~]$
```

Paso 3: Ejecutar el siguiente script para recuperar el ID del Security Group del Web Security Group.

```
SG=$(aws ec2 describe-security-groups --filters Name=group-name,Values=WebSecurityGroup --query SecurityGroups[].GroupId --
output text)
echo $SG
```

```
[ec2-user@ip-10-0-0-137 ~]$ SG=$(aws ec2 describe-security-groups --filters Name=group-name,Val
[ec2-user@ip-10-0-0-137 ~]$ echo $SG
sg-011ae1cab8d786f4e
[ec2-user@ip-10-0-0-137 ~]$
```

Paso 4: Ejecutar el siguiente comando para descargar el User Data Script para instalar y configurar la instancia como un servidor web.

```
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-RESTRT-1-23732/171-lab-JAWS-create-ec2/s3/UserData.txt
```

```
[ec2-user@ip-10-0-0-137 ~]$ wget https://aws-tc-largeobjects.s3.us-
--2023-10-21 01:17:57--  https://aws-tc-largeobjects.s3.us-west-2.a
Resolving aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-la
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-t
HTTP request sent, awaiting response... 200 OK
Length: 327 [text/plain]
Saving to: 'UserData.txt'

100%[=====================================================================

2023-10-21 01:17:57 (15.0 MB/s) - 'UserData.txt' saved [327/327]

[ec2-user@ip-10-0-0-137 ~]$ ls
UserData.txt
[ec2-user@ip-10-0-0-137 ~]$
```

Paso 5: Ejecutar el siguiente comando para lanzar la instancia servidor web con la AWS CLI.

```
INSTANCE=$(\
aws ec2 run-instances \
--image-id $AMI \
--subnet-id $SUBNET \
--security-group-ids $SG \
--user-data file:///home/ec2-user/UserData.txt \
--instance-type t3.micro \
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Web Server}]' \
--query 'Instances[*].InstanceId' \
--output text \
)
echo $INSTANCE
```

```
[ec2-user@ip-10-0-0-137 ~]$ INSTANCE=$(\
> aws ec2 run-instances \
> --image-id $AMI \
> --subnet-id $SUBNET \
> --security-group-ids $SG \
> --user-data file:///home/ec2-user/UserData.txt \
> --instance-type t3.micro \
> --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Web Server}]' \
> --query 'Instances[*].InstanceId' \
> --output text \
> )
[ec2-user@ip-10-0-0-137 ~]$ echo $INSTANCE
i-0a9aed9d1d8081083
[ec2-user@ip-10-0-0-137 ~]$
```

Paso 6: Puede monitorear el estado de la instancia consultando el estado mediante la AWS CLI.

```
aws ec2 describe-instances --instance-ids $INSTANCE
```

```
[ec2-user@ip-10-0-0-137 ~]$ aws ec2 describe-instances --instance-ids $INSTANCE
{
    "Reservations": [
        {
            "Instances": [
                {
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "PublicDnsName": "ec2-54-218-192-208.us-west-2.compute.amazonaws.com",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "EbsOptimized": false,
```

```
aws ec2 describe-instances --instance-ids $INSTANCE --query 'Reservations[].Instances[].State.Name' --output text
```

El segundo comando es el mismo que el primero, pero entrega solamente el nombre del estado de la instancia.

```
[ec2-user@ip-10-0-0-137 ~]$ aws ec2 describe-instances --instar
running
[ec2-user@ip-10-0-0-137 ~]$
```

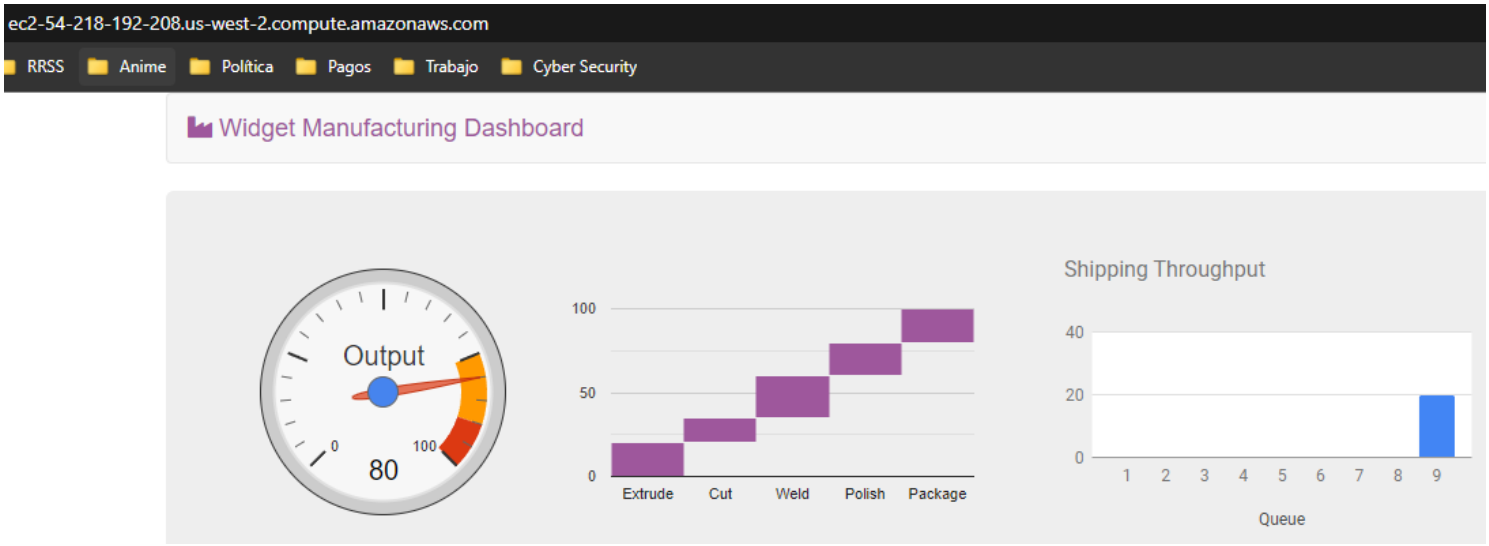
Paso 7: Ejecutar el siguiente comando para probar que el Servidor Web lanzado funciona correctamente:

```
aws ec2 describe-instances --instance-ids $INSTANCE --query Reservations[].Instances[].PublicDnsName --output text
```

- El comando entrega el Public IPv4 DNS de la instancia.
- Copiar el DNS y pegar en un navegador Web.
- Si se despliega una página web significa que el servidor web está funcionando correctamente.

```
[ec2-user@ip-10-0-0-137 ~]$ aws ec2 describe-instances --instance-ids $INSTANCE
ec2-54-218-192-208.us-west-2.compute.amazonaws.com
[ec2-user@ip-10-0-0-137 ~]$
```

ec2-54-218-192-208.us-west-2.compute.amazonaws.com

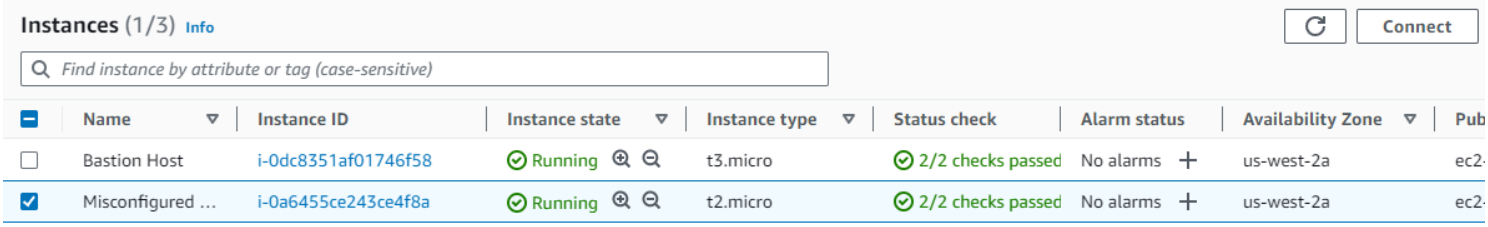


Paso 8: AWS Management Console → EC2 → Instances → Web Server

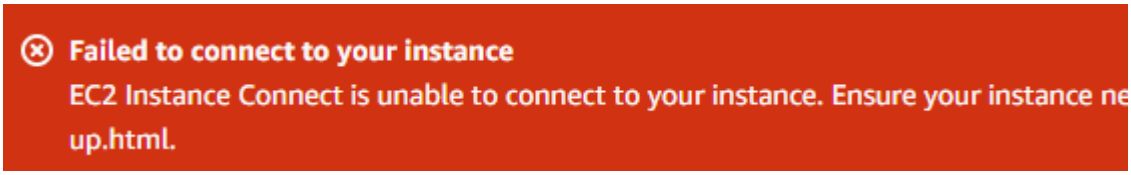
Name	Instance ID	Instance state	Instance type
Bastion Host	i-0dc8351af01746f58	Running	t3.micro
Misconfigured ...	i-0a6455ce243ce4f8a	Running	t2.micro
Web Server	i-0a9aed9d1d8081083	Running	t3.micro

Desafío 1: Conéctese a una instancia de Amazon EC2

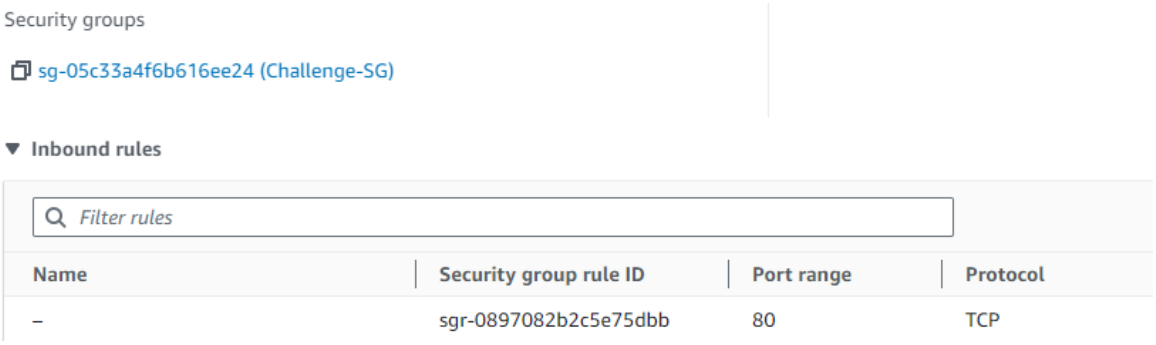
En este desafío, debe solucionar los problemas de configuración de seguridad de una instancia denominada Misconfigured Web Server.



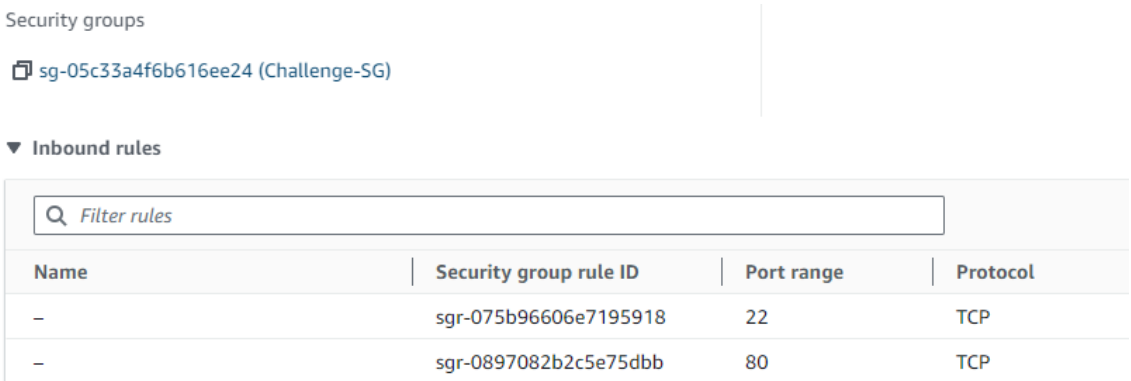
Paso 1: Intente conectarse a la instancia Misconfigures Web Server mediante EC2 Instance Connect.



Paso 2: Diagnostique por qué no funciona y corrija la configuración incorrecta.



El Security Group asociado a la instancia no tiene habilitada la conexión vía SSH.



Paso 3: Reintentar conectarse mediante EC2 Instance Connect nuevamente luego de permitir la conexión vía SSH en el Security group asociado.