44°

Lab - AWS re/Start
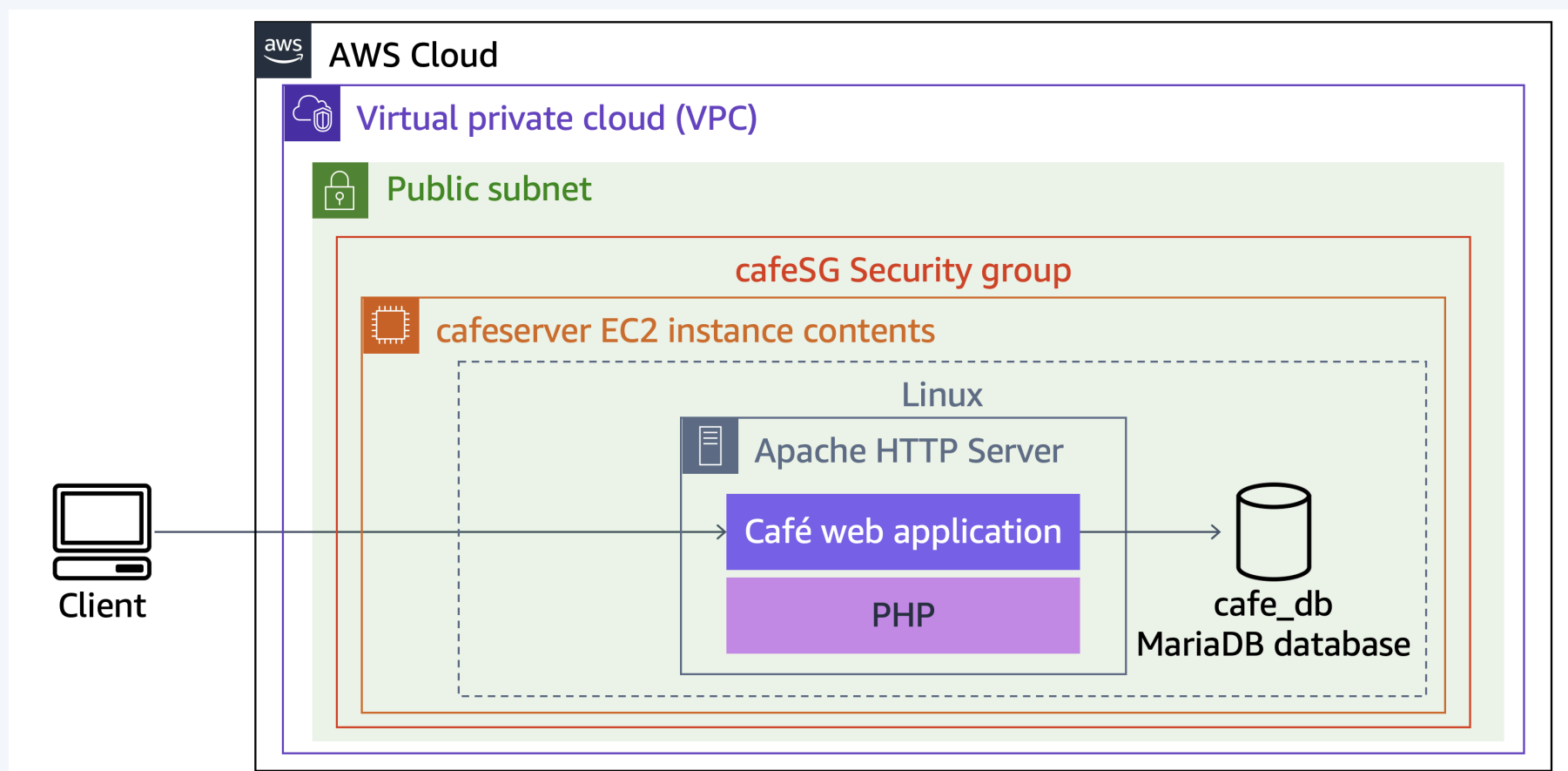
Solución de Problemas de Creación de una Instancia

EC2

Interactuando con Amazon EC2

A continuación, se muestra los objetivos del laboratorio:
- Lanzar una instancia EC2 mediante la CLI de AWS.
- Solucionar problemas con los comandos de la CLI de AWS y la configuración de los servicios de Amazon EC2

Tarea 01

Lo primero que haremos es conectarnos a la instancia mediante usando el *EC2 Instance Connect*. Donde configuramo la AWS CLI

Y procedemos a lanzar una nueva instancia. Primero, podemos analizar el script que utilizaremos en un rato
Como es un *Bash Script*, se necesita que la primera línea sea **#!/bin/bash**

```bash
#!/bin/bash
DATE=`date '+%Y-%m-%d %H:%M:%S'`
echo
echo "Running create-instance.sh on "$DATE
echo
```

Luego seteamos el tipo de instancia, y guardamos en una var

```bash
# Hard coded values
instanceType="t3.small"
echo "Instance Type: "$instanceType
profile="default"
echo "Profile: "$profile

echo
echo "Looking up account values..."
```

Luego, hacemos una búsqueda de la VPC llamada *Cafe VPC*, pues aquí es donda la nueva instancia debe ser desplegada.

```
# get vpcId
vpc=""
while [[ "$vpc" == "" ]] ; do
  for i in $(aws ec2 describe-regions | grep RegionName | cut -d '"' -f4) ; do
    region=$i;
    vpc=$(aws ec2 describe-vpcs --region $i --filters "Name=tag:Name,Values='Cafe VPC'" --profile $profile | grep VpcId | cut -d '"' -f4 | sed -n 1p );
    if [[ "$vpc" != "" ]]; then
      break;
    fi
  done
done
echo
echo "VPC: "$vpc
echo "Region: "$region

vpc=$(aws ec2 describe-vpcs \
--filters "Name=tag:Name,Values='Cafe VPC'" \
--region $region \
--profile $profile | grep VpcId | cut -d '"' -f4 | sed -n 1p)
echo "VPC: "$vpc
```

Asimismo, buscamos info de subnet, par de claves e IAM ID

```
# get subnetId
subnetId=$(aws ec2 describe-subnets \
--filters "Name=tag:Name,Values='Cafe Public Subnet 1'" \
"create-lamp-instance-v2.sh" [readonly][noeol] 205L, 6093B
:31
--region $region \

--profile $profile \
%
--query "Subnets[*]" | grep SubnetId | cut -d '"' -f4 | sed -n 1p)
%
echo "Subnet Id: "$subnetId
%

%
# Get keypair name
%
key=$(aws ec2 describe-key-pairs \
%
--profile $profile --region $region | grep KeyName | cut -d '"' -f4 )
%
echo "Key: "$key
%

%
# Get AMI ID
%
imageId=$(aws ssm get-parameters \
%
--names '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2' \
%
--profile $profile \
%
--region $region | grep ami- | cut -d '"' -f4 | sed -n 2p)
%
echo "AMI ID: "$imageId
```

Tarea 01

También verificamos que no haya instancia ni grupo de seguridad con el mismo nombre ya existente

```
#check for existing cafe instance

existingEc2Instance=$(aws ec2 describe-instances \

--region $region \

--profile $profile \

--filters "Name=tag:Name,Values=cafeserver" "Name=instance-state-name,Values=running" \

| grep InstanceId | cut -d '"' -f4)
if [[ "$existingEc2Instance" != "" ]]; then

  echo

  echo "WARNING: Found existing running EC2 instance with instance ID "$existingEc2Instance"."

  echo "This script will not succeed if it already exists. "

  echo "Would you like to delete it? [Y/N]"

  echo ">>"


  validResp=0

  while [ $validResp -eq 0 ];

  do

    read answer

    if [[ "$answer" == "Y" || "$answer" == "y" ]]; then

        echo
```

```
        echo "Deleting the existing instance..."

        aws ec2 terminate-instances --instance-ids $existingEc2Instance --region $region --profile $profile

        #wait for confirmation it was terminated

        aws ec2 wait instance-terminated --instance-ids $existingEc2Instance --region $region --profile $profile

        validResp="1"

    elif [[ "$answer" == "N" || "$answer" == "n" ]]; then

        echo "Ok, exiting."

        exit 1

    else

        echo "Please reply with Y or N."

    fi

  done


  sleep 10 #give it 10 seconds before trying to delete the SG this instance used.
fi

#check for existing cafeSG security Group

existingMpSg=$(aws ec2 describe-security-groups \

--region $region \
```

Se crea un grupo de seguridad con sus respectivas reglas de entrada y se definen los puertos de acceso

```
# CREATE a security group and capture the name of it

echo

echo "Creating a new security group..."

securityGroup=$(aws ec2 create-security-group --group-name "cafeSG" \

--description "cafeSG" \

--region $region \

--group-name "cafeSG" \

--vpc-id $vpc --profile $profile | grep GroupId | cut -d '"' -f4 )
echo "Security Group: "$securityGroup


# Open ports in the security group

echo

echo "Opening port 22 in the new security group"

aws ec2 authorize-security-group-ingress \

--group-id $securityGroup \

--protocol tcp \

--port 22 \

--cidr 0.0.0.0/0 \

--region $region \
```

```

echo "Opening port 80 in the new security group"

aws ec2 authorize-security-group-ingress \

--group-id $securityGroup \

--protocol tcp \

--port 8080 \

--cidr 0.0.0.0/0 \

--region $region \

--profile $profile
```

Después se crea la instancia, utilizando los parámetros que hemos recuperado previamente:

```
echo "Creating an EC2 instance in "$region

instanceDetails=$(aws ec2 run-instances \

--image-id $imageId \

--count 1 \

--instance-type $instanceType \

--region us-east-1 \

--subnet-id $subnetId \

--security-group-ids $securityGroup \

--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=cafeserver}]' \

--associate-public-ip-address \

--iam-instance-profile Name=LabInstanceProfile \

--profile $profile \

--user-data file://create-lamp-instance-userdata-v2.txt \

--key-name $key )


#if the create instance command failed, exit this script

if [[ "$?" -ne "0" ]]; then

  exit 1

fi
```

También podemos ver el contenido de los datos de Usuario de nuestra instancia

```
[ec2-user@cli-host starters]$ cat create-lamp-instance-userdata-v2.txt
#!/bin/bash
yum -y update
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum -y install httpd mariadb-server

systemctl enable httpd
systemctl start httpd

systemctl enable mariadb
systemctl start mariadb

echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php

usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www

#Check /var/log/cloud-init-output.log after this runs to see errors, if any.

#
# Download and unzip the Cafe application files.
#

# Database scripts
wget https://aws-tc-largeobjects.s3.amazonaws.com/CUR-TF-100-RESTRT-1/173-activity-JAWS-troubleshoot-instance/db-v2.tar.gz
tar -zxvf db-v2.tar.gz

# Web application files
wget https://aws-tc-largeobjects.s3.amazonaws.com/CUR-TF-100-RESTRT-1/173-activity-JAWS-troubleshoot-instance/cafe-v2.tar.gz
tar -zxvf cafe-v2.tar.gz -C /var/www/html/

#
# Run the scripts to set the database root password, and create and populate the application database.
# Check the following logs to make sure there are no errors:
#
#       /db/set-root-password.log
#       /db/create-db.log
#
cd db
./set-root-password.sh
./create-db.sh
hostnamectl set-hostname web-server
[ec2-user@cli-host starters]$
```

Tenemos un error al tratar de correr el script:

```
[ec2-user@cli-host starters]$ ./create-lamp-instance-v2.sh

Running create-instance.sh on 2024-03-04 05:20:39

Instance Type: t3.small
Profile: default

Looking up account values...

VPC: vpc-0a7194b027fca35e3
Region: us-west-2
VPC: vpc-0a7194b027fca35e3
Subnet Id: subnet-01f102be454266826
Key: vockey
AMI ID: ami-0dfd45428f2d4af0c

Creating a new security group...
Security Group: sg-0acc3f9a62eab356e

Opening port 22 in the new security group
Opening port 80 in the new security group

Creating an EC2 instance in us-west-2

An error occurred (InvalidAMIID.NotFound) when calling the RunInstances operation: The image id '[ami-0dfd45428f2d4af0c]' does not exist
[ec2-user@cli-host starters]$
```

SWIPE

Tarea 01

Como pista nos preguntan si el valor de la región es el correcto, entonces verifiquemoslo. Es en el lanzamiento de la instancia donde se define la región a desplegar, debemos colocar la correcta que es *us-west-2*:

Como resultado: Se logra lanzar la nueva instancia

Tarea 01

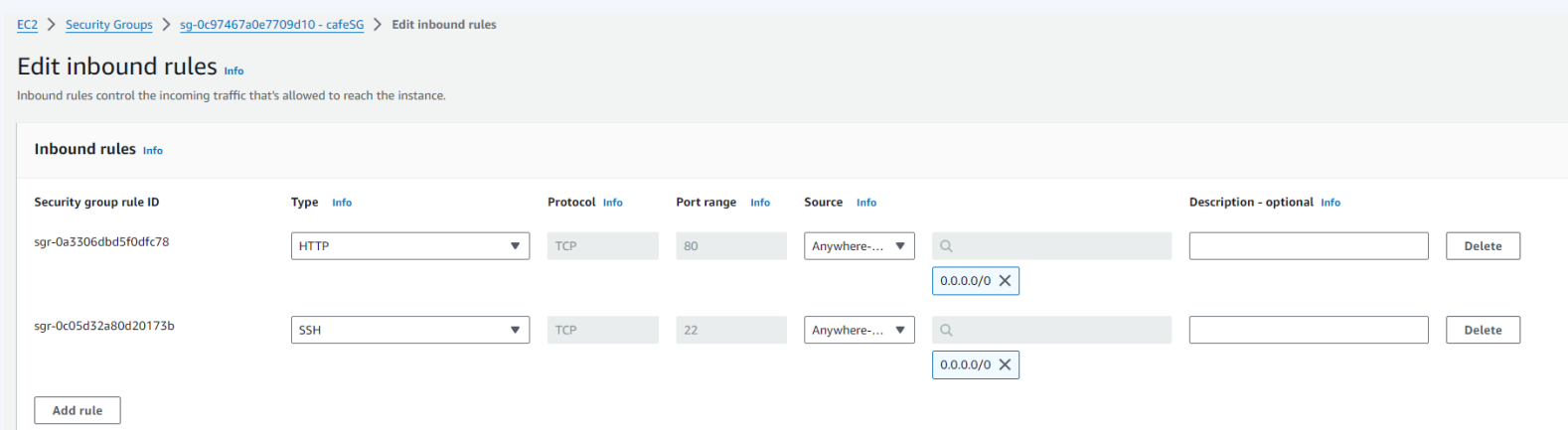Pero al intentar conectarnos a dicha instancia via HTTP con puerto 80, aún no podemos hacer dicha conexión
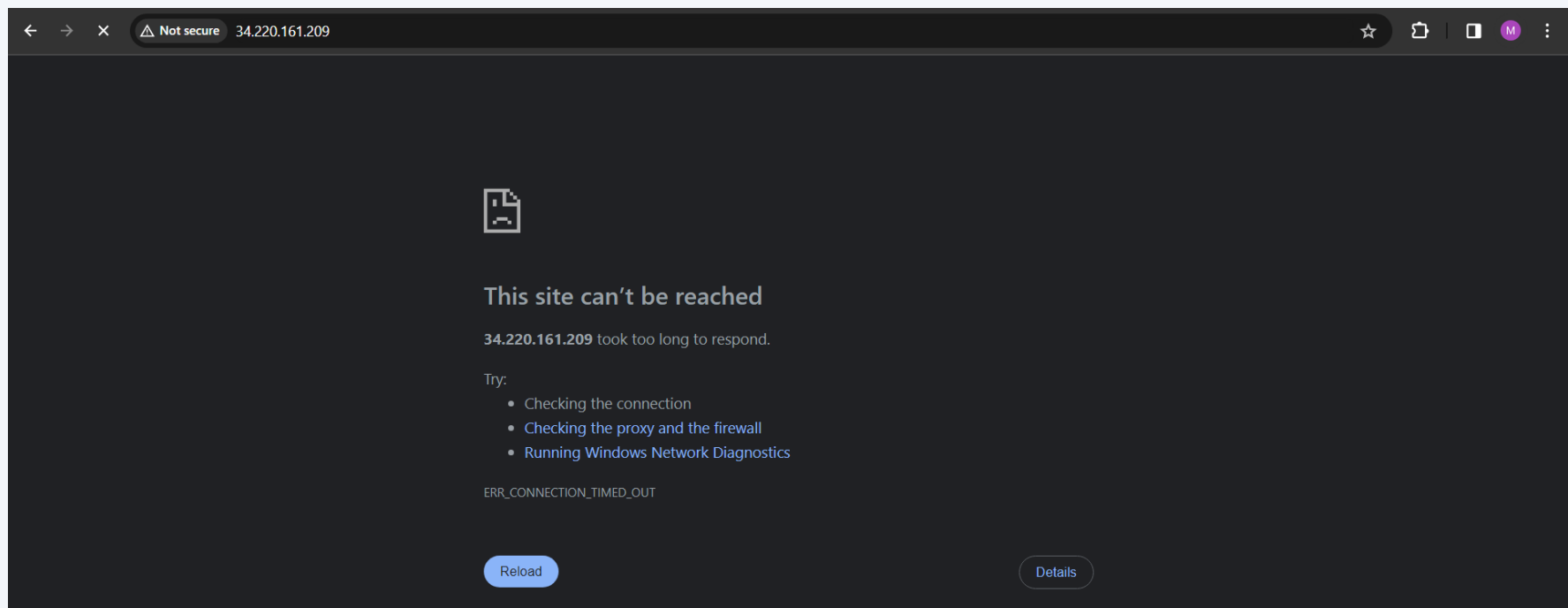
Vamos a verificar si este puerto está abierto. Y notamos que no es así, solo el puerto 22 está abierto

Vamos a abrir este puerto

Tarea 01

Veamos si ahora podemos acceder a dicha instancia. Lo logramos!!
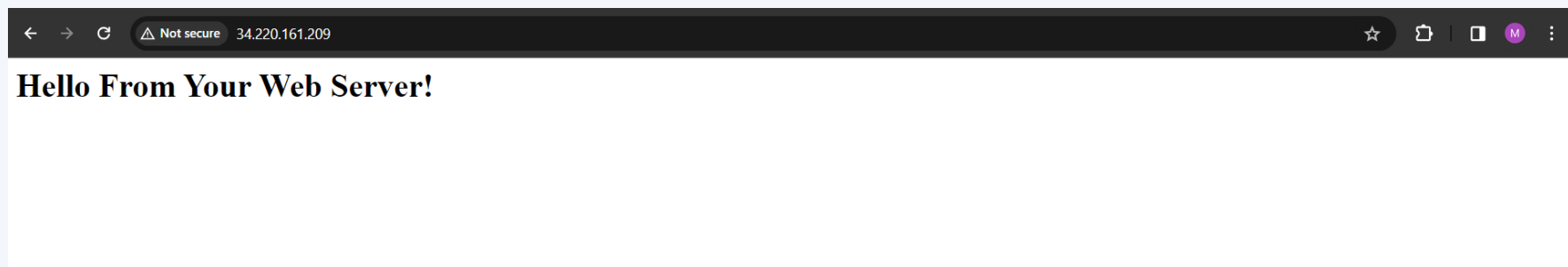
Podemos ver el archivo de registro, donde vemos la base de datos instalada en la instancia

Ahora ingresemos a la página del cafe, vemos que se registran correctamente las órdenes