Contents

[Purpose 2](#_Toc55306443)

[General Diagram 2](#_Toc55306444)

[Prerequisites 2](#_Toc55306445)

[Lab 5A: Spot EC2 Instance with additional ENI, EIP, and bootstrap scripts 3](#_Toc55306446)

[Lab 5A using Web Management Console 3](#_Toc55306447)

[Create VPC, Subnet, IGW, Routing Table (Labs4c1) 3](#_Toc55306448)

[Create keys, Sec Group (Labs4c1) 3](#_Toc55306449)

[Check Price and make a Spot Request 3](#_Toc55306450)

[Create ENI, EIP and its associations 11](#_Toc55306451)

[Review using Web Console 15](#_Toc55306452)

[Lab 5A using Command Line (Windows) 17](#_Toc55306453)

[Create VPC, Subnet, IGW, Routing Table (Labs4c1) 17](#_Toc55306454)

[Create keys, Sec Group (Labs4c1) 18](#_Toc55306455)

[Determine AMI, bid price, make spot request, and see EC2 spot instance running 19](#_Toc55306456)

[Create ENI, EIP and associate it 20](#_Toc55306457)

[Review using CLI 21](#_Toc55306458)

[Review Configuration using Putty 21](#_Toc55306459)

[Clean Resources 24](#_Toc55306460)

[For Command Line (Windows) 24](#_Toc55306461)

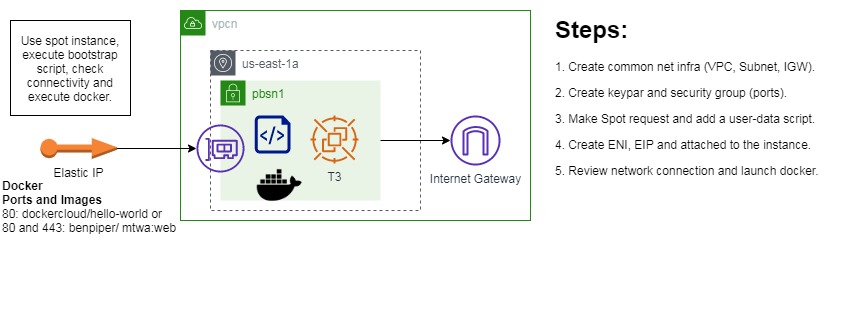
[Evidences to send 25](#_Toc55306462)

# Purpose

Using network infrastructure as base, create a spot instance with a public IP using EIP, ENI and show a HTTP Server using Docker as Container.

# General Diagram

Simple public network infrastructure with a Spot Instance using Docker and bootstrap script. Public instance using Elastic IP (EIP) on an Elastic Network Interface (ENI).



# Prerequisites

Labs1c1 have to be done and the context for Administrative user have to activated on Command Line Session.

Labs4c1 have to be done, because you learn how to: Create subnets, VPCs, IGW, and Routing Tables. For this case specifically, you have to create VPC, Public Subnet, IGW, Routing Table with the same names as that laboratory, therefore we only focus on the new things.

The bootstrap script using Base64 encode so you have to use one on Windows (certutil -encode <infile> <outfile>) or MacOs (openssl base64 -in <infile> -out <outfile>) or Web (<https://www.base64decode.org/>)

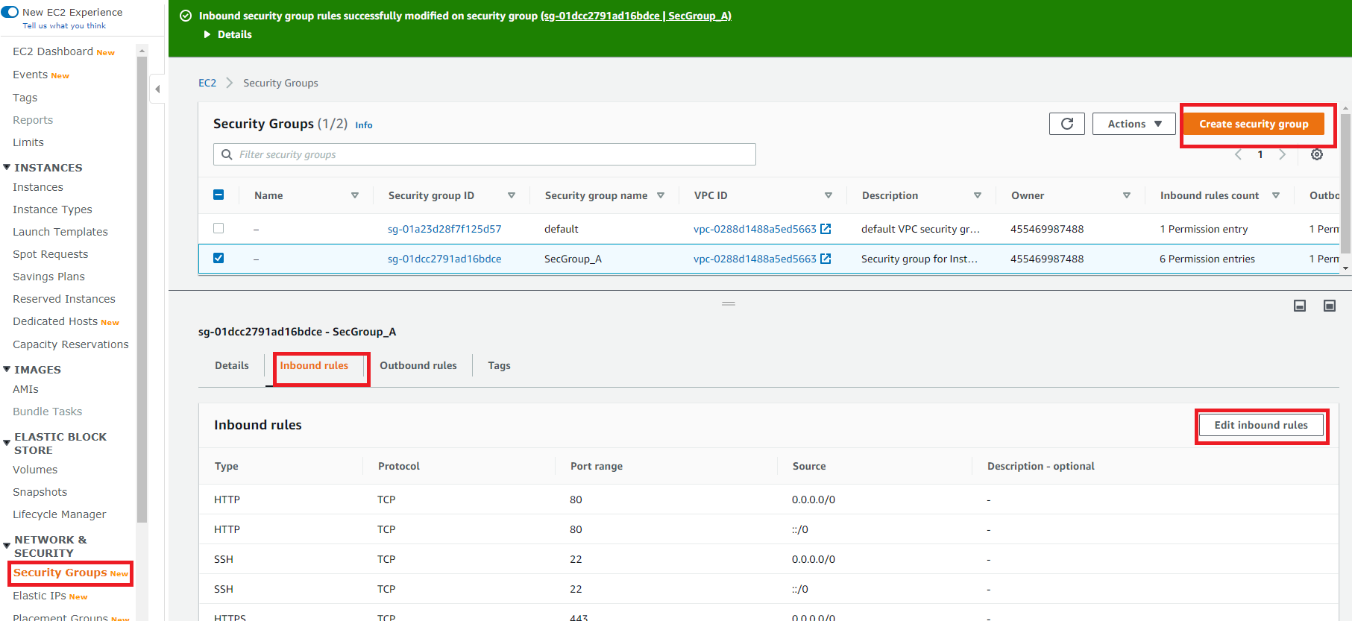
# Lab 5A: Spot EC2 Instance with additional ENI, EIP, and bootstrap scripts

## Lab 5A using Web Management Console

### Create VPC, Subnet, IGW, Routing Table (Labs4c1)

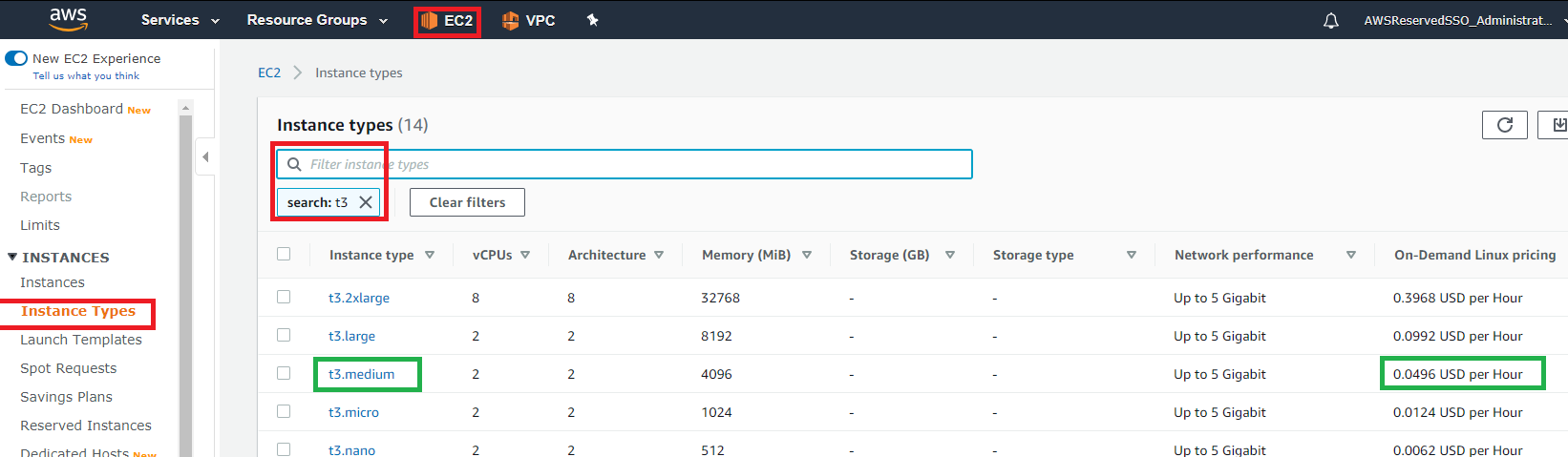
### Create keys, Sec Group (Labs4c1)

We use the same configuration as previous lab for one public VPC and subnet. In additional, we aggregate ports, 80 and 443 to the security group.

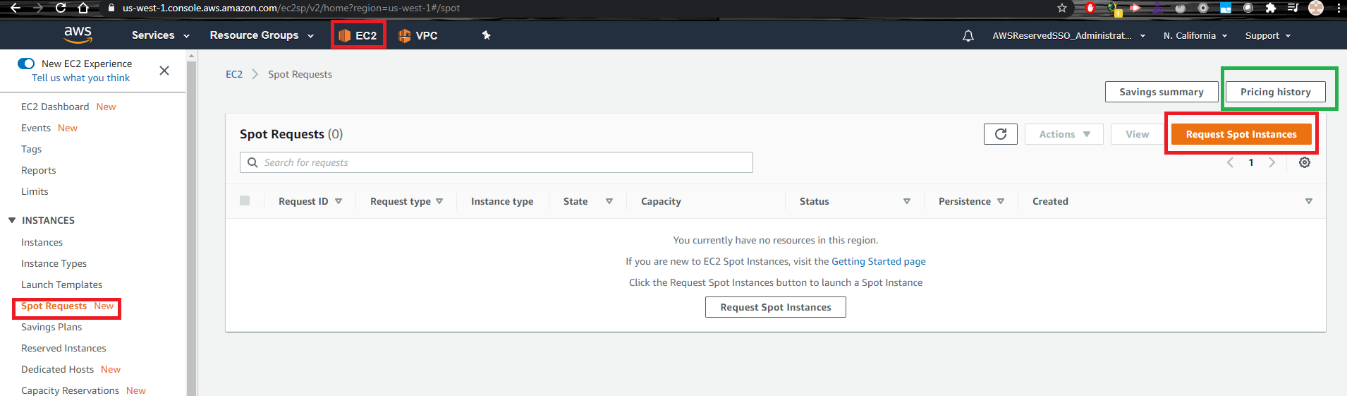


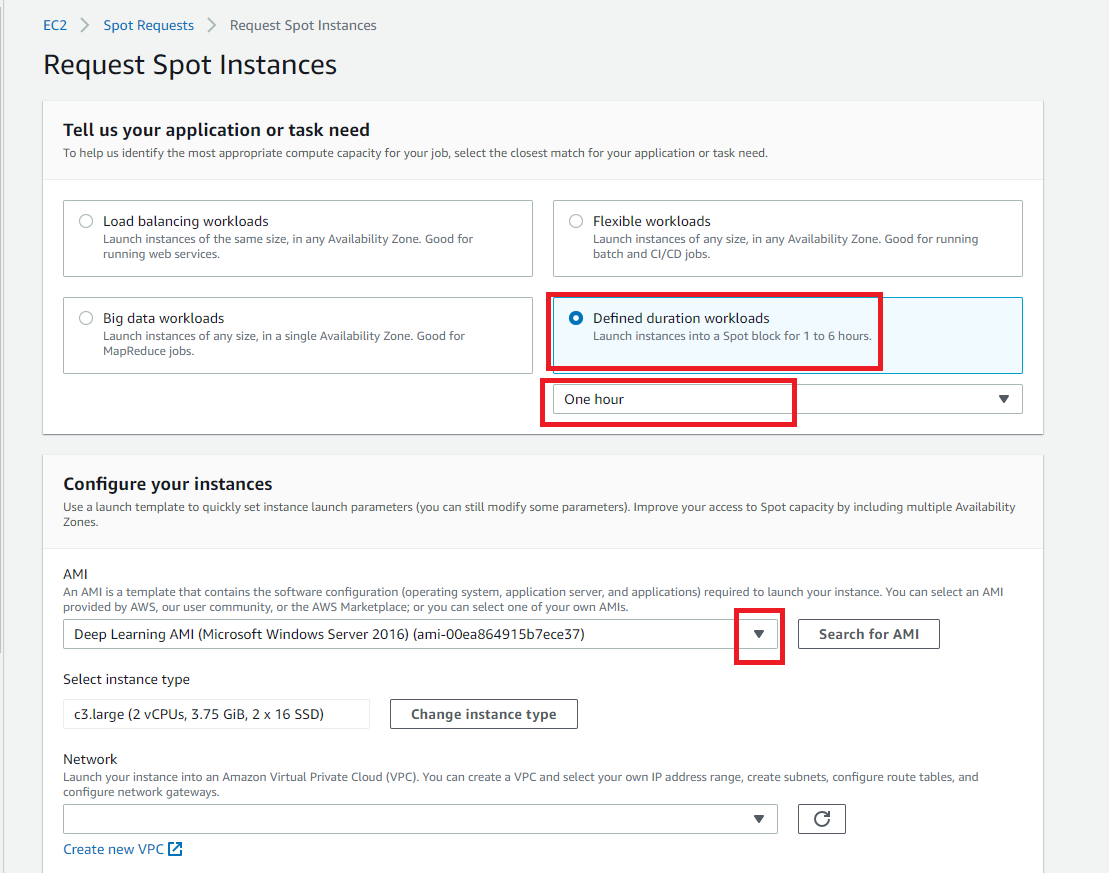
### Check Price and make a Spot Request

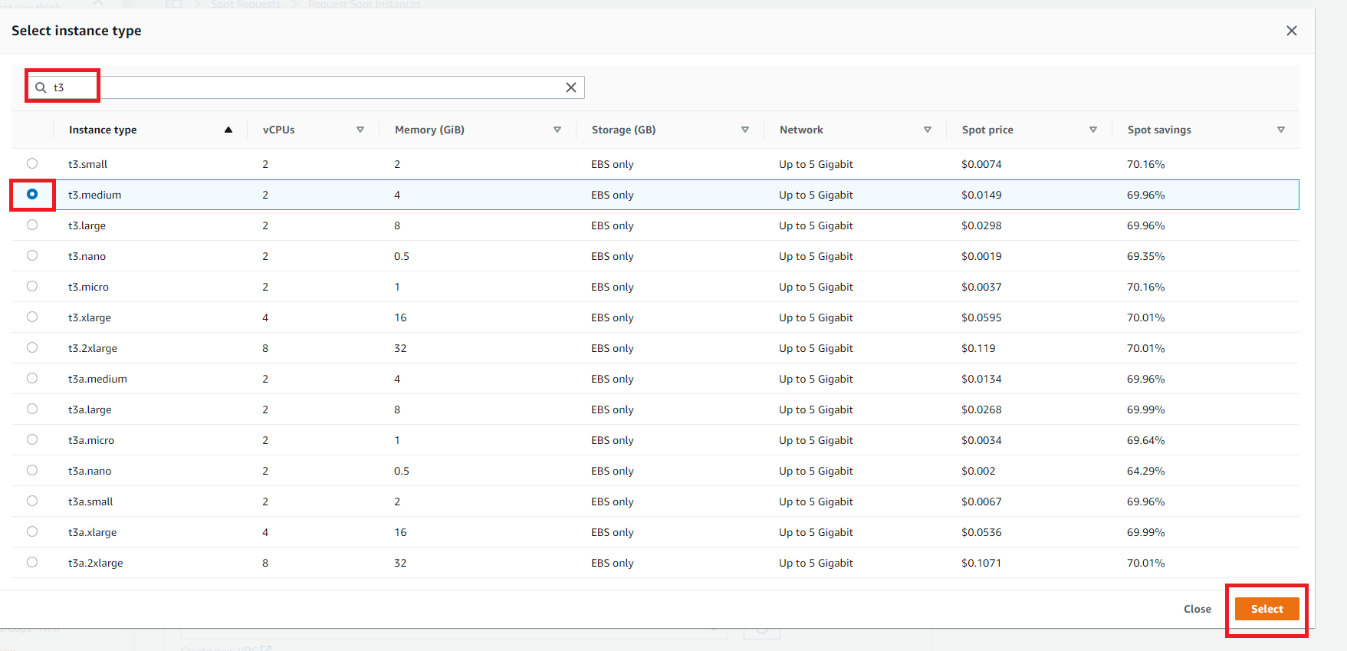
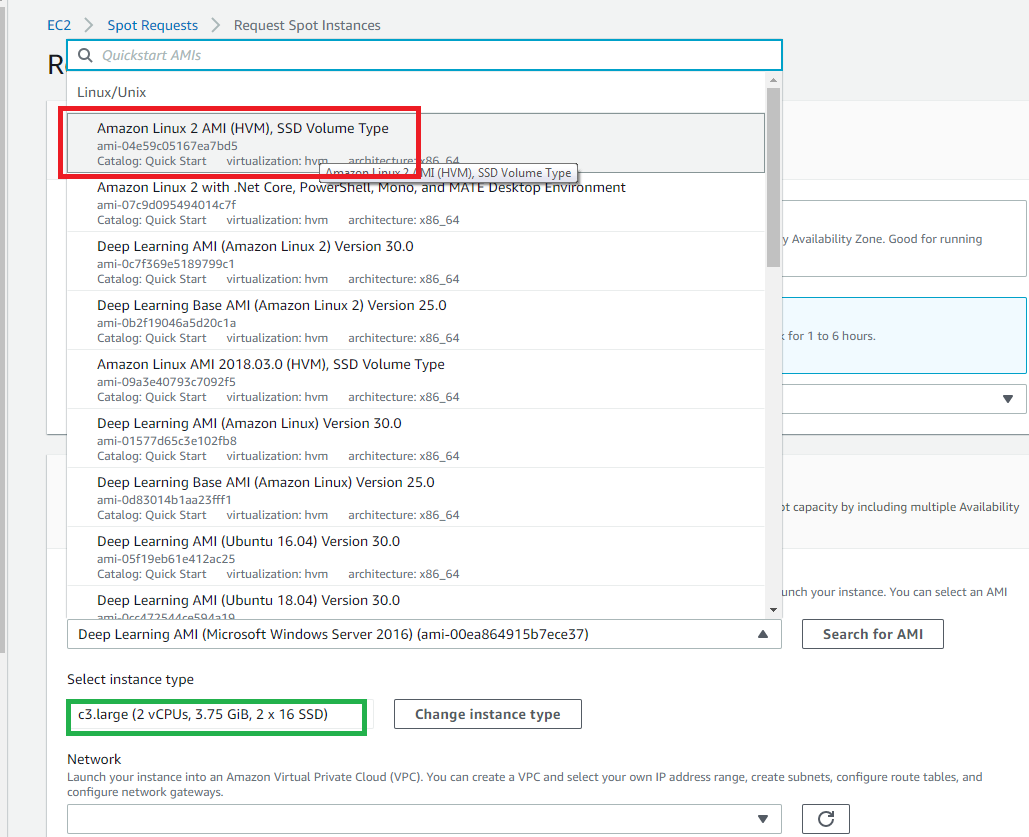
Before of making a Spot Request, you can review what is the price for specific on-demand instance, and then go to make a Spot Request.

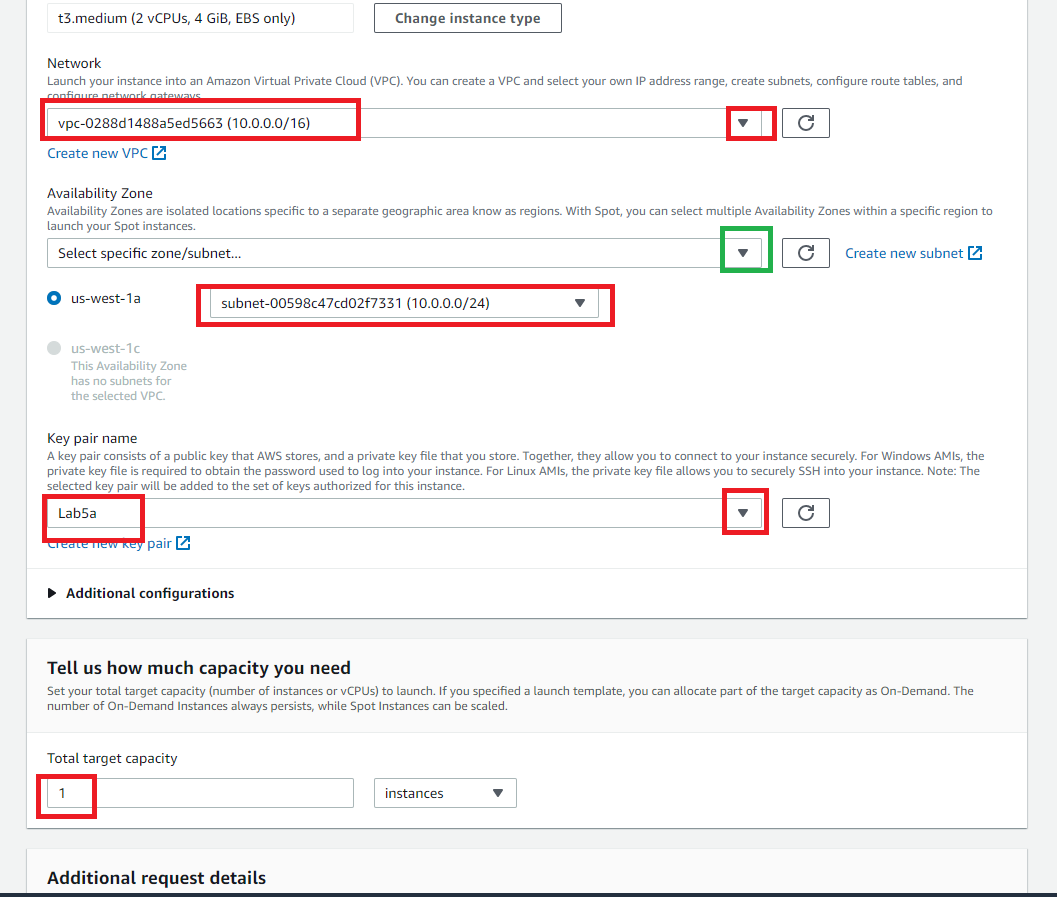


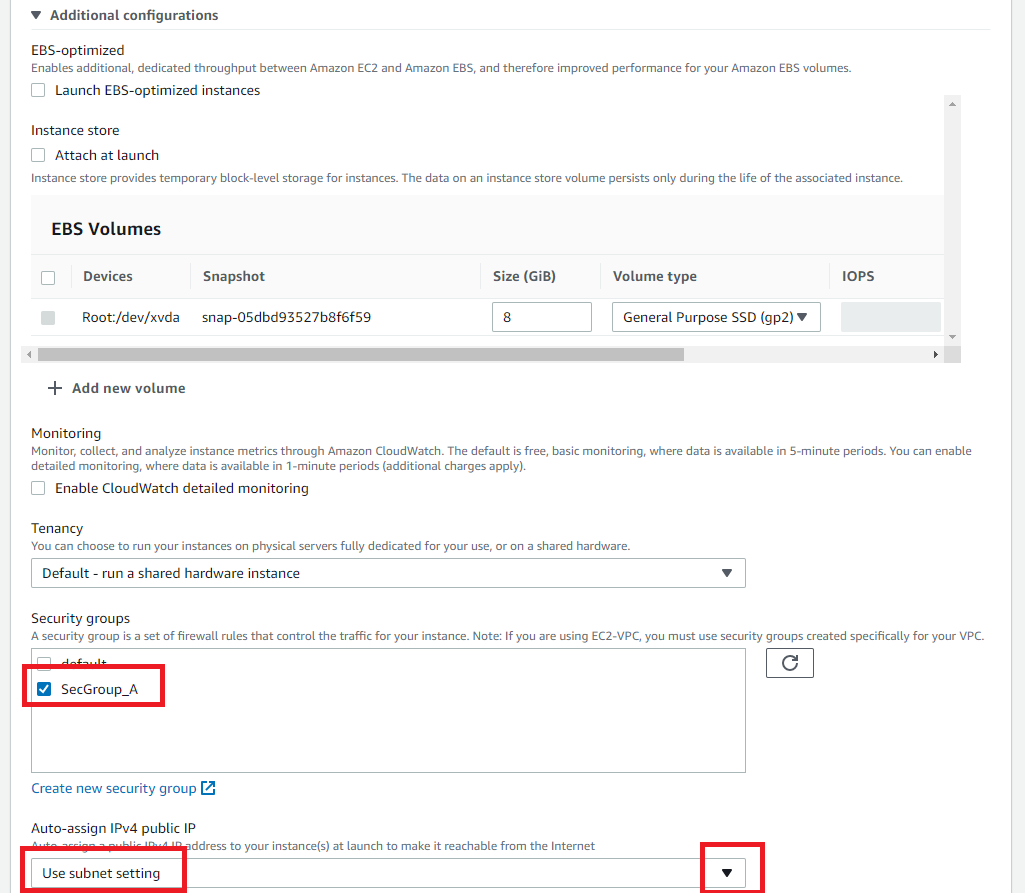
Make the spot request

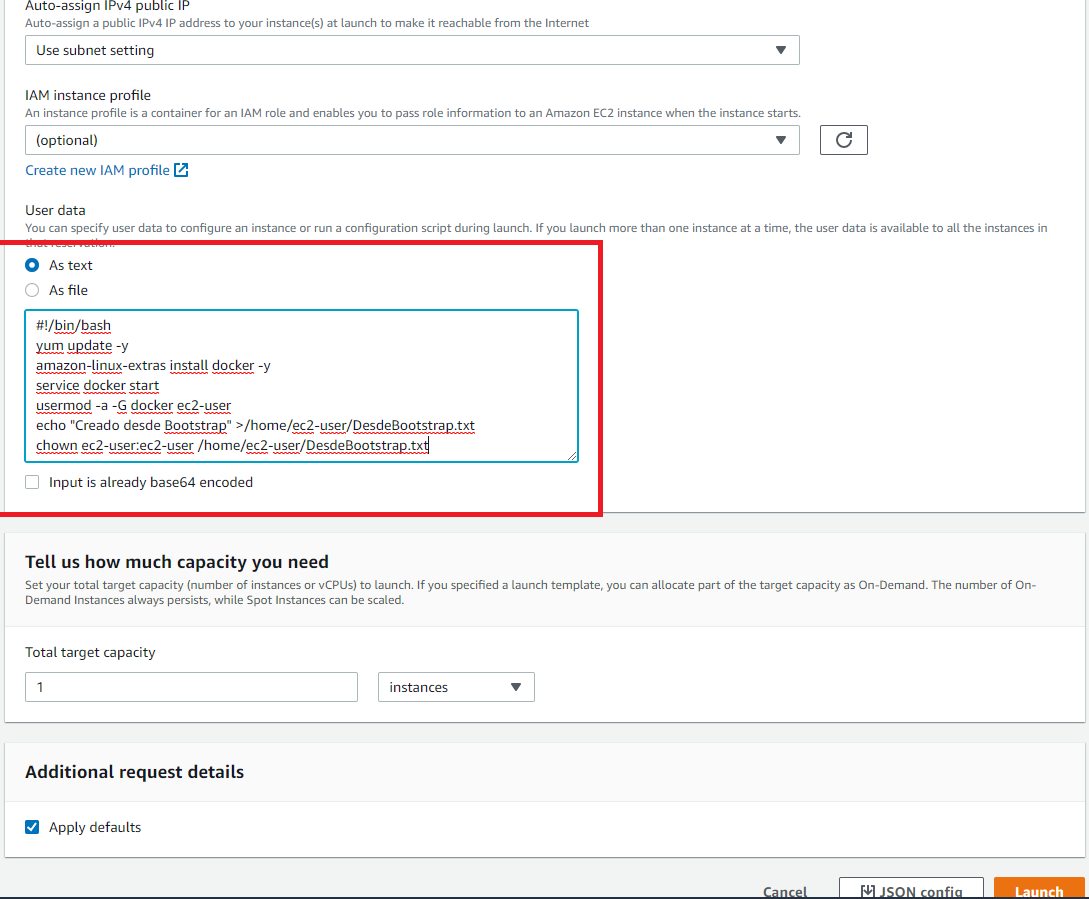


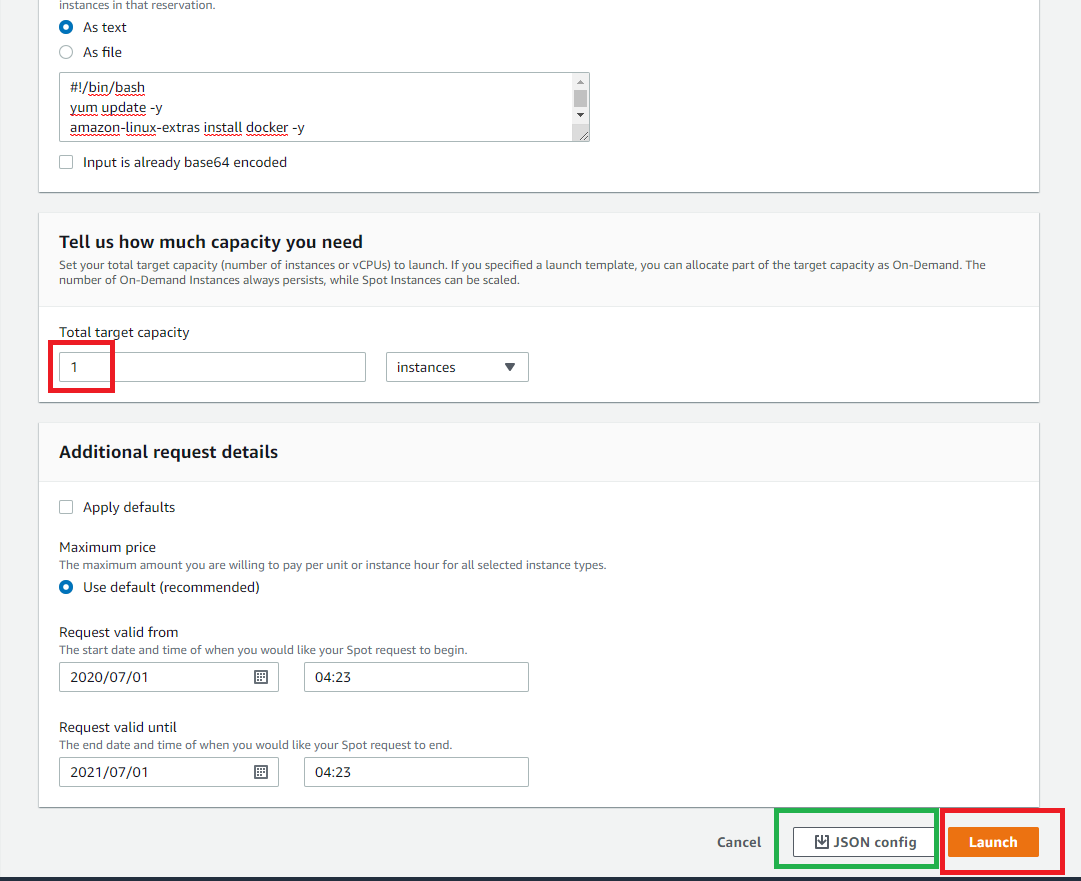




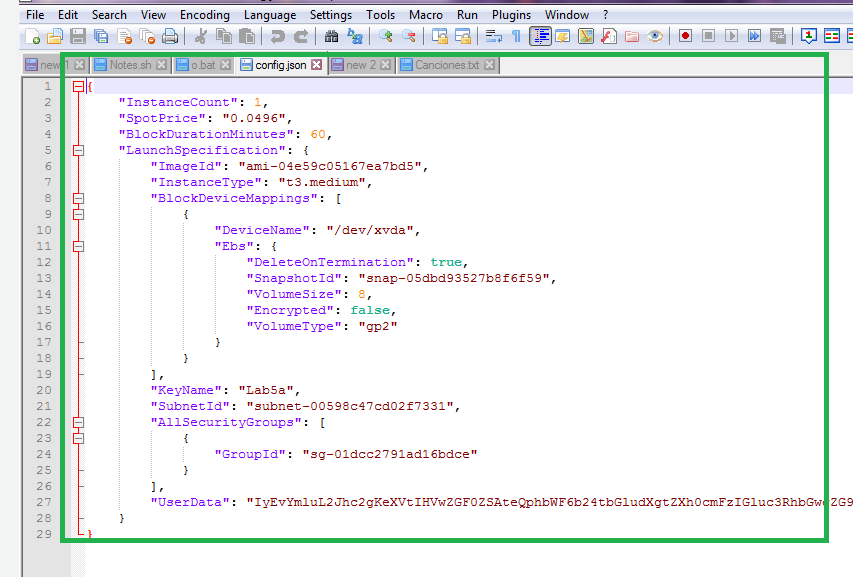




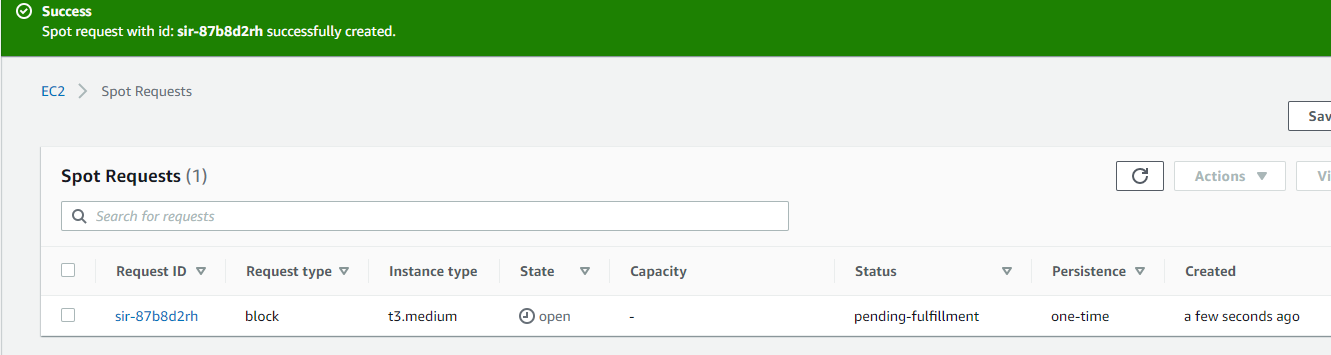




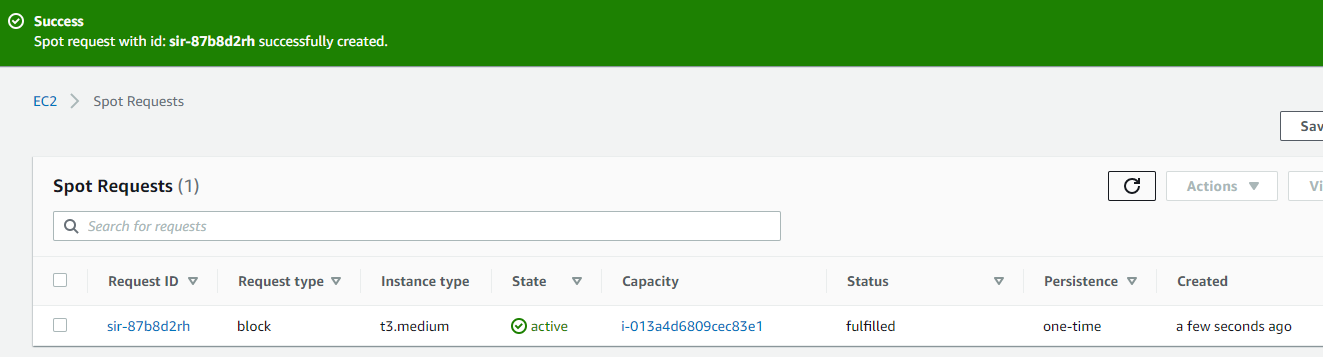
Check JSON File



Then, the request is done is going to check the feasibility to deliver the instance.

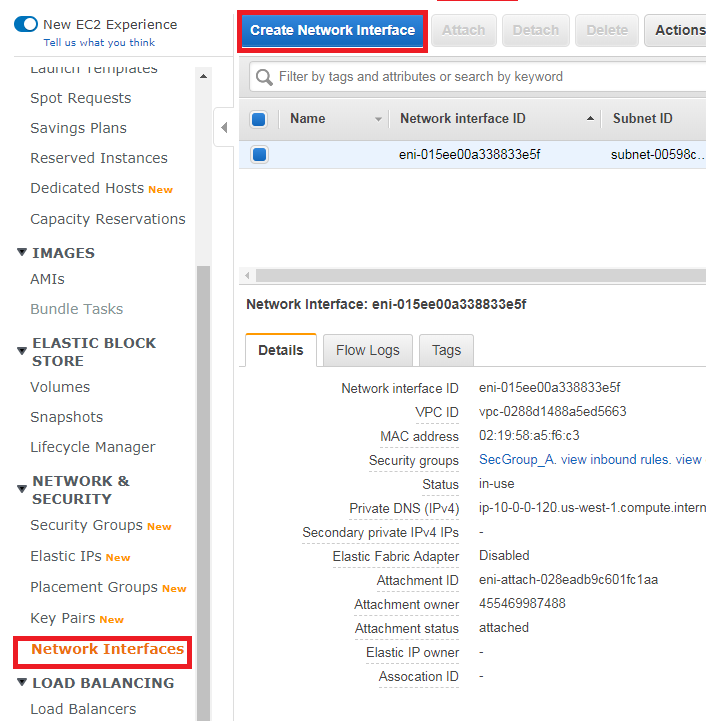


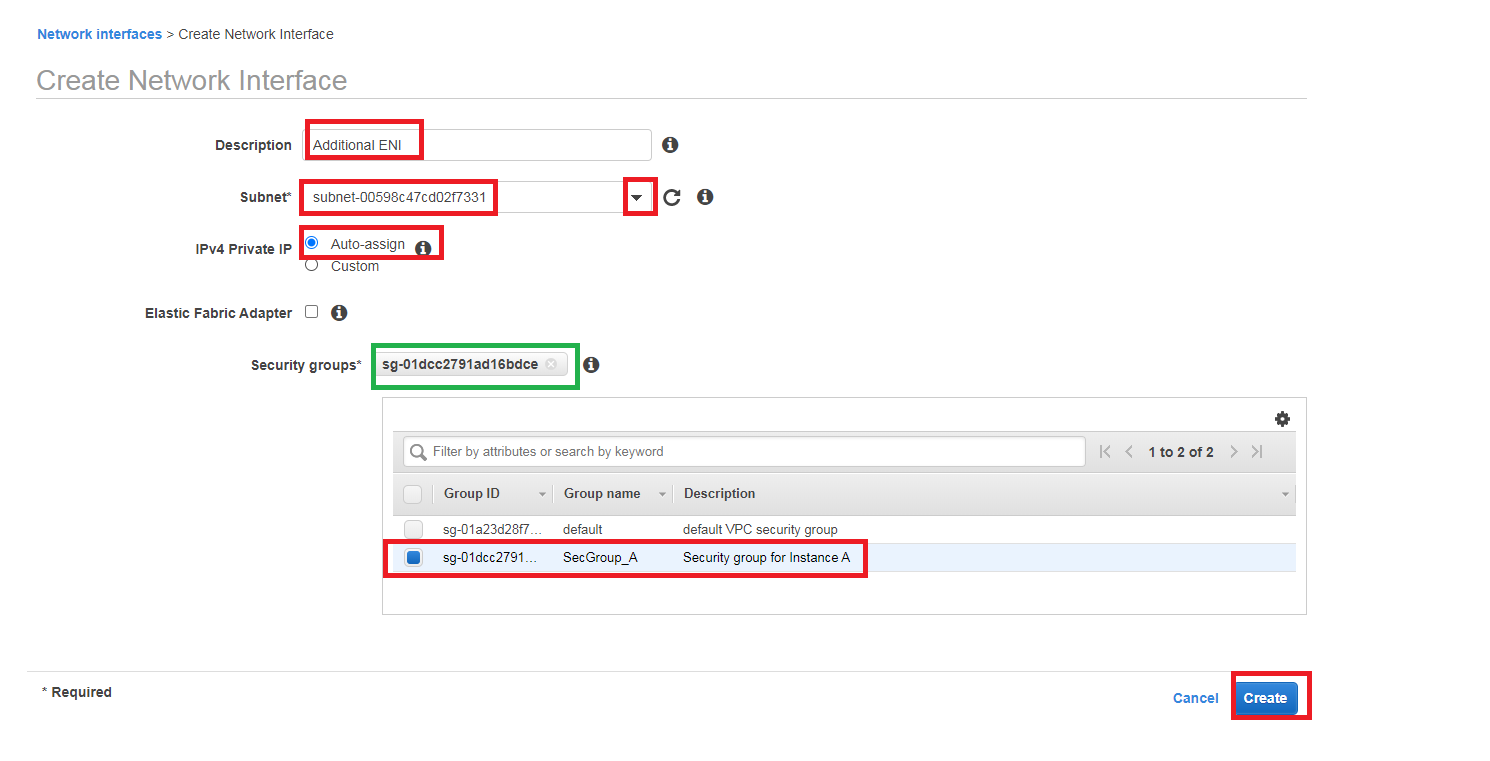
After some seconds, status changes and the instance is created.

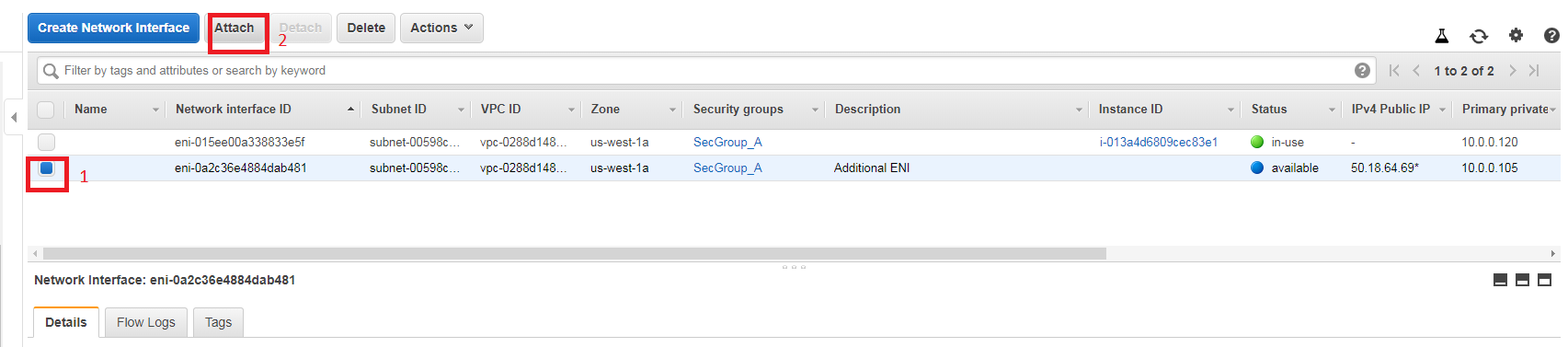


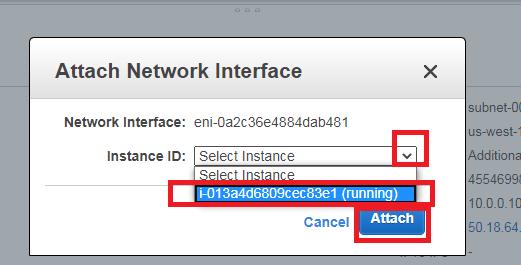
### Create ENI, EIP and its associations

Create the ENI and attach to the running instances

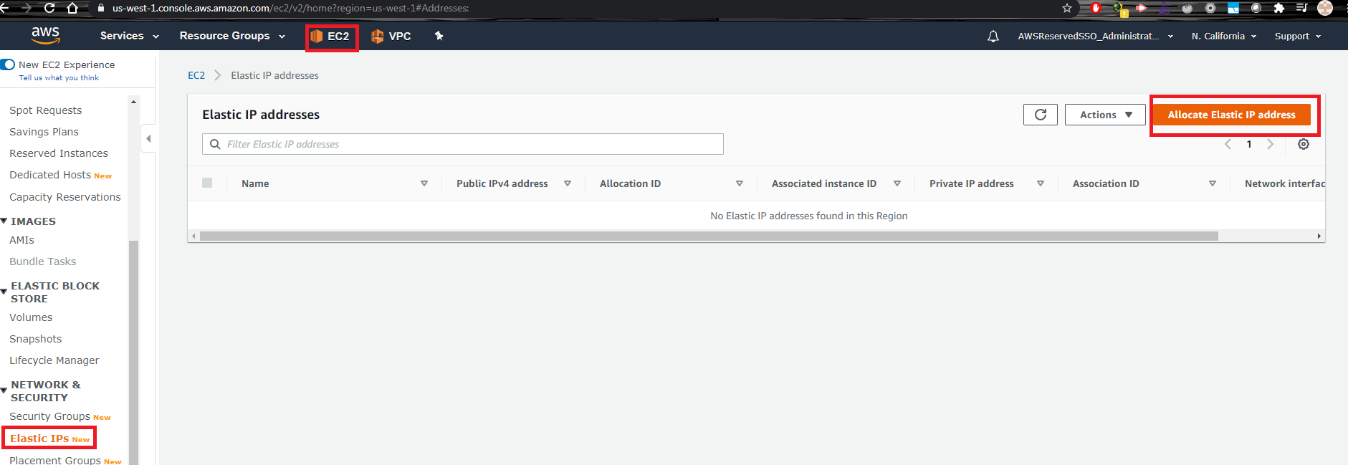


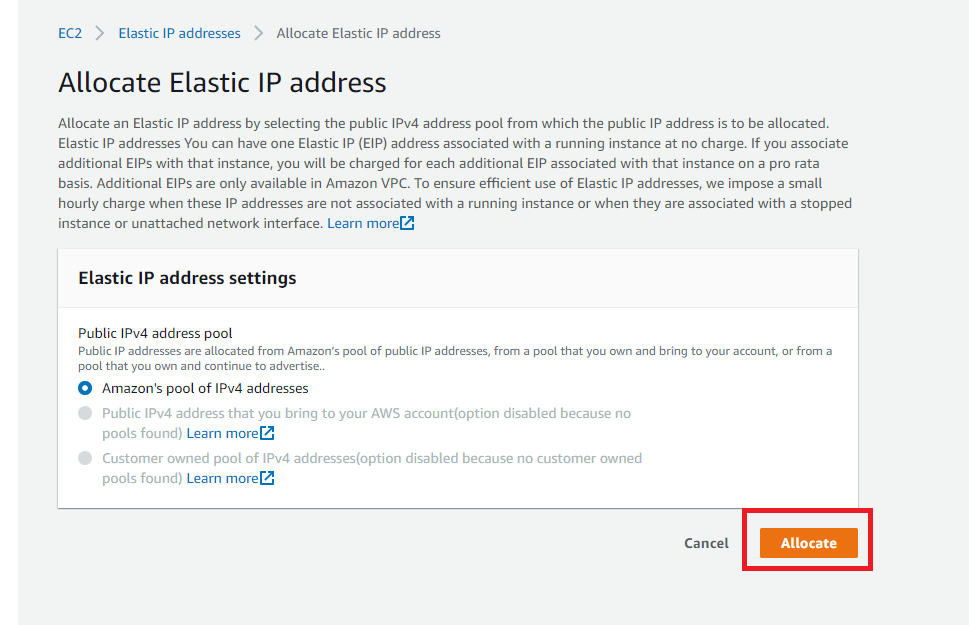


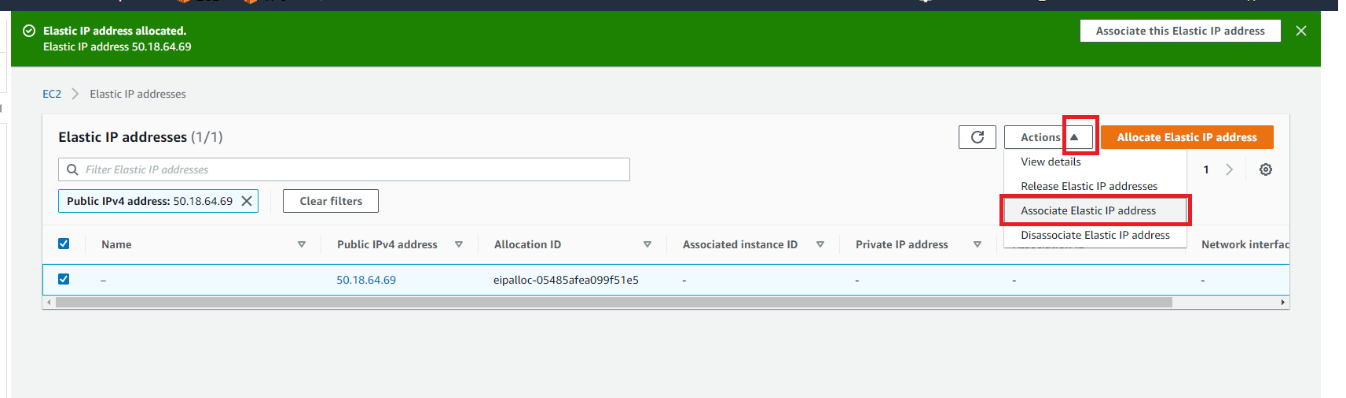


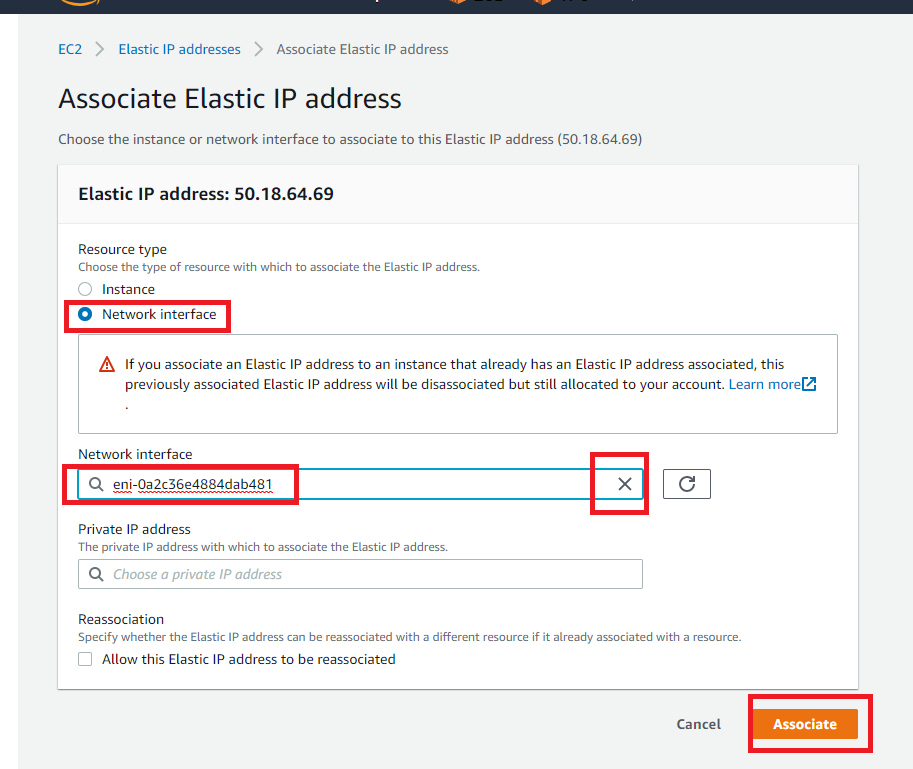


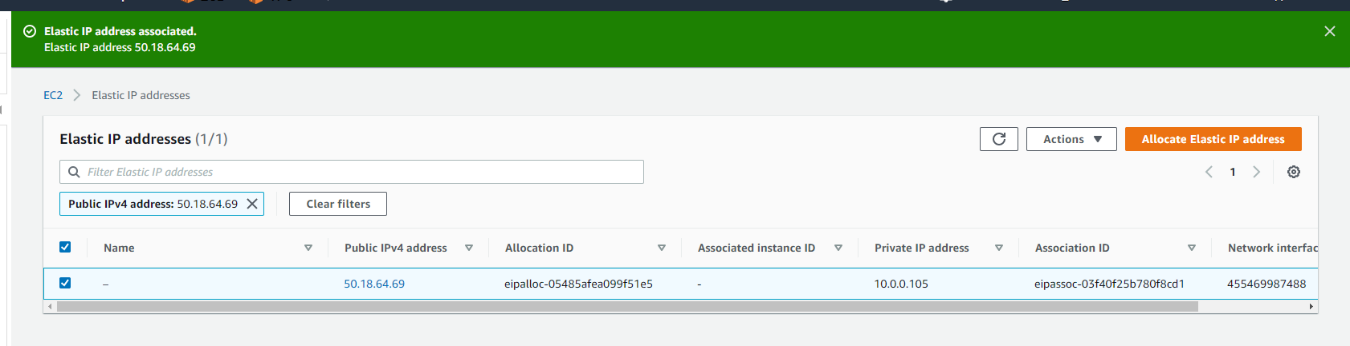
Create (Allocate) EIP and associated it to the ENI



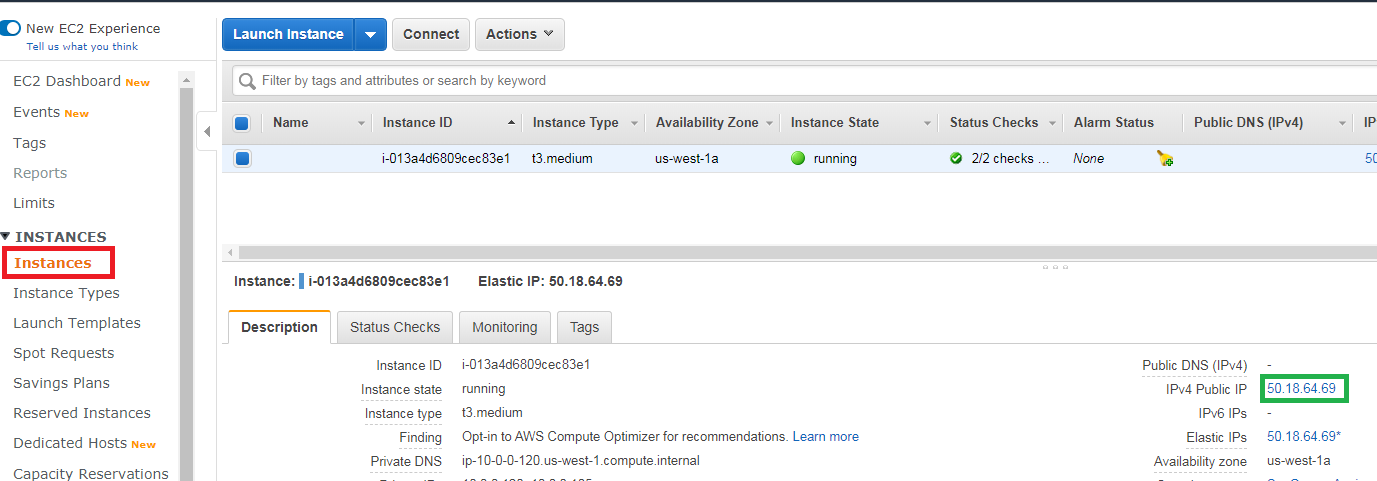








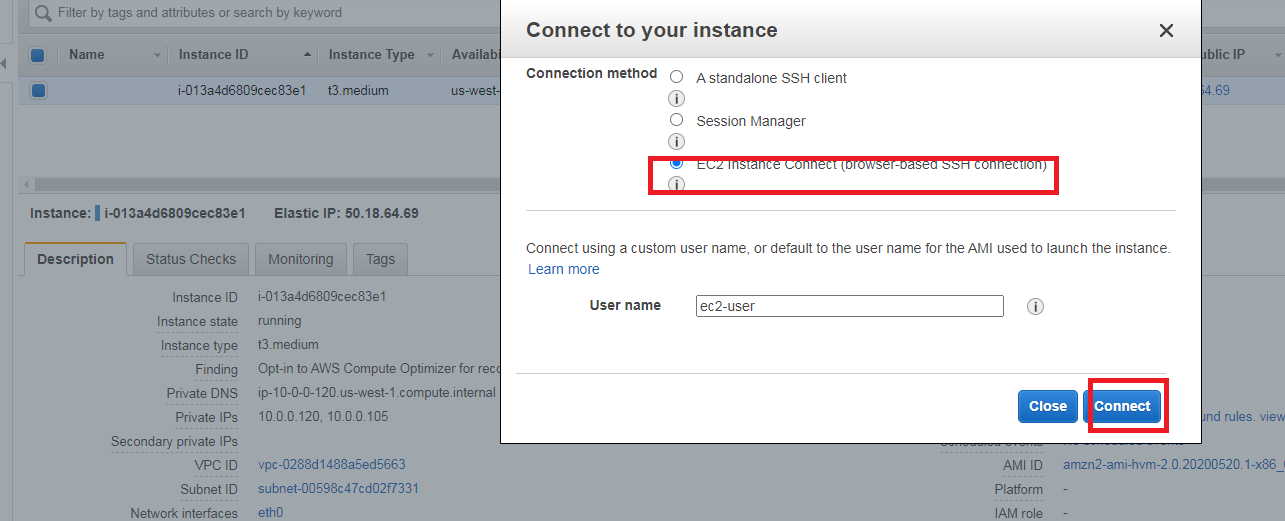
When you go to EC2 Instance section, you can see the assigned IP to the instance



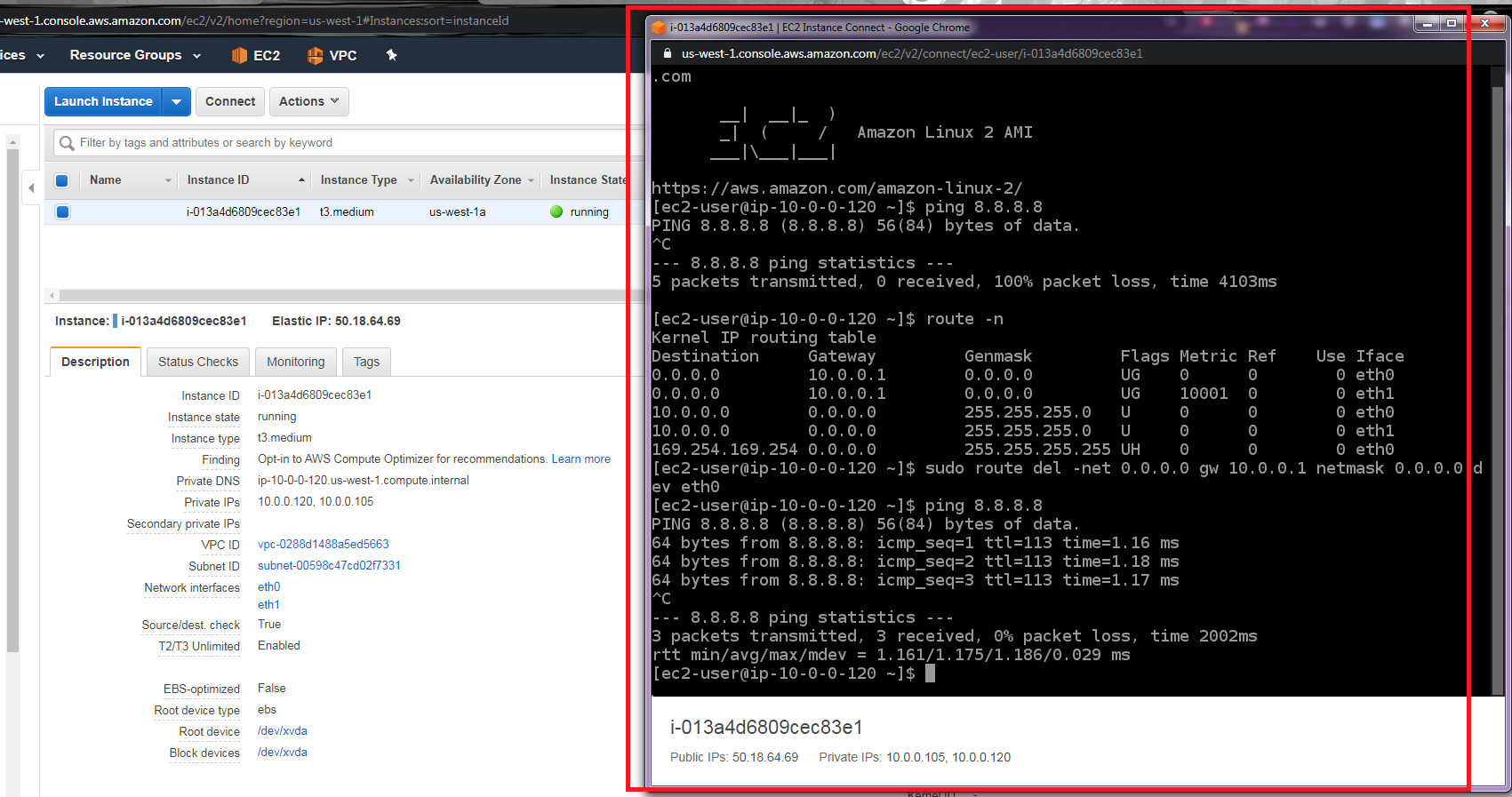
### Review using Web Console

Go to section Review Configuration using Putty.

You can compare the connection of Putty and connection using EC2 Instance Connect, right click on instance and click Connect.



You have to review the reason of those commands on [Review Configuration using Putty](#_Review_Configuration_using) section. You have to jump to that section to complete the Lab.



## Lab 5A using Command Line (Windows)

### Create VPC, Subnet, IGW, Routing Table (Labs4c1)

rem Crear la VPC y habilitar resolucion DNS

aws ec2 create-vpc --cidr-block %vpcn\_Mask%|jq ".Vpc.VpcId" >tmpFile

set /p vpcn\_Id= < tmpFile

aws ec2 modify-vpc-attribute --vpc-id %vpcn\_Id% --enable-dns-hostnames "{\"Value\":true}"

rem Crear subred Publica

aws ec2 create-subnet --vpc-id %vpcn\_Id% --cidr-block %pbsn1\_Mask% --availability-zone %first\_az%|jq ".Subnet.SubnetId" >tmpFile

set /p pbsn1\_Id= < tmpFile

rem Crear el Internet Gateway IGW y asignarlo a la VPC

aws ec2 create-internet-gateway|jq ".InternetGateway.InternetGatewayId"  >tmpFile

set /p IGW\_Id= < tmpFile

aws ec2 attach-internet-gateway --vpc-id %vpcn\_Id% --internet-gateway-id %IGW\_Id%

rem Crear tabla de ruteo publica y asignarle IGW como ruta por defecto

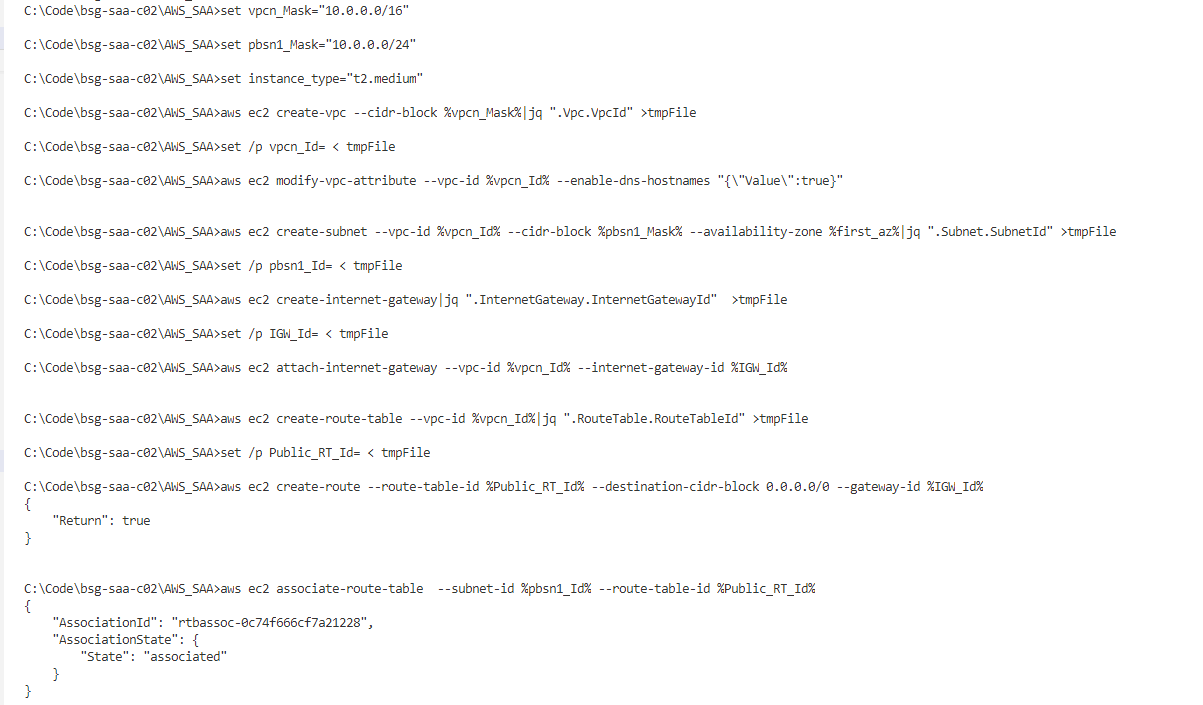
aws ec2 create-route-table --vpc-id %vpcn\_Id%|jq ".RouteTable.RouteTableId" >tmpFile

set /p Public\_RT\_Id= < tmpFile

aws ec2 create-route --route-table-id %Public\_RT\_Id% --destination-cidr-block 0.0.0.0/0 --gateway-id %IGW\_Id%

rem Asociar la tabla de ruta a la subred

aws ec2 associate-route-table  --subnet-id %pbsn1\_Id% --route-table-id %Public\_RT\_Id%



### Create keys, Sec Group (Labs4c1)

rem Crear las llaves para el SSH a las nuevas instancias y convertirlas a PPK para usar Putty ya sea con puttygen o winscp

aws ec2 create-key-pair --key-name Lab5a --query "KeyMaterial" --output text > Lab5a.pem

winscp.com /keygen "Lab5a.pem" /output="Lab5a.ppk"

rem Crear los Security Groups para esa instancia

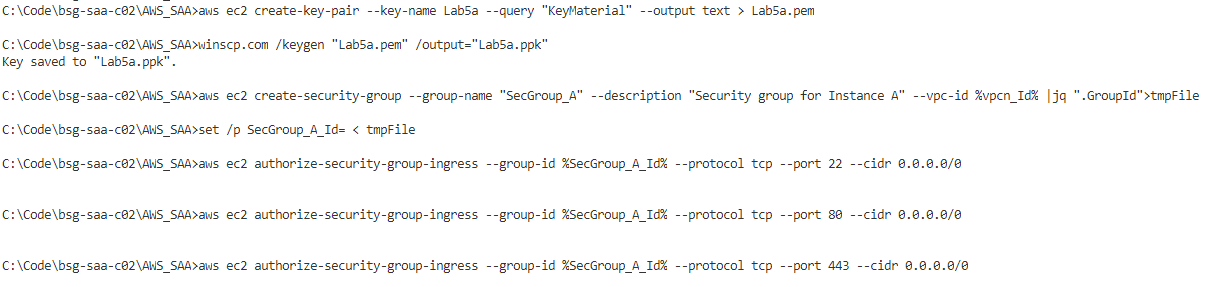
aws ec2 create-security-group --group-name "SecGroup\_A" --description "Security group for Instance A" --vpc-id %vpcn\_Id% |jq ".GroupId">tmpFile

set /p SecGroup\_A\_Id= < tmpFile

aws ec2 authorize-security-group-ingress --group-id %SecGroup\_A\_Id% --protocol tcp --port 22 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress --group-id %SecGroup\_A\_Id% --protocol tcp --port 80 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress --group-id %SecGroup\_A\_Id% --protocol tcp --port 443 --cidr 0.0.0.0/0



### Determine AMI, bid price, make spot request, and see EC2 spot instance running

rem En el laboratorio de EC2 Inicial se mostrar la importancia de buscar una AMI correcto.

rem AWS sugiere que se tome el AMI Amazon Linux 2 y se instale docker desde linea de comandos: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html#install\_docker

aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-2.0.????????.?-x86\_64-gp2" "Name=state,Values=available" --query "reverse(sort\_by(Images, &CreationDate))[:1].ImageId" --output text >tmpFile

set /p AMI= < tmpFile

rem Vamos a buscar un valor establecido para la subasta de la instancia

aws ec2 describe-spot-price-history --instance-types %instance\_type% --product-description "Linux/UNIX (Amazon VPC)" --start-time 2020-06-23T07:08:09 --end-time 2020-06-24T08:09:10

rem Despues de mirar valor se va a ser solicitud de una sola vez. Se podria hacerse persistente, y estado de solicitudes en https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-requests.html

rem Recuerde que antes de lanzar este comando se tiene que modificar con el AMI, el Security Group, subred en pbsn1\_Id y el bootstrap script (user data) en base 64 (usar certutil -encode bootstrap.sh bootstrapb64.sh en Windows)

rem Si el precio de la apuesta es muy bajo no alcanza a competir y no se ejecuta, por eso es importante revisar el estado del request

aws ec2 request-spot-instances --spot-price "0.03" --block-duration-minutes 60 --instance-count 1 --type "one-time" --launch-specification file://config.json

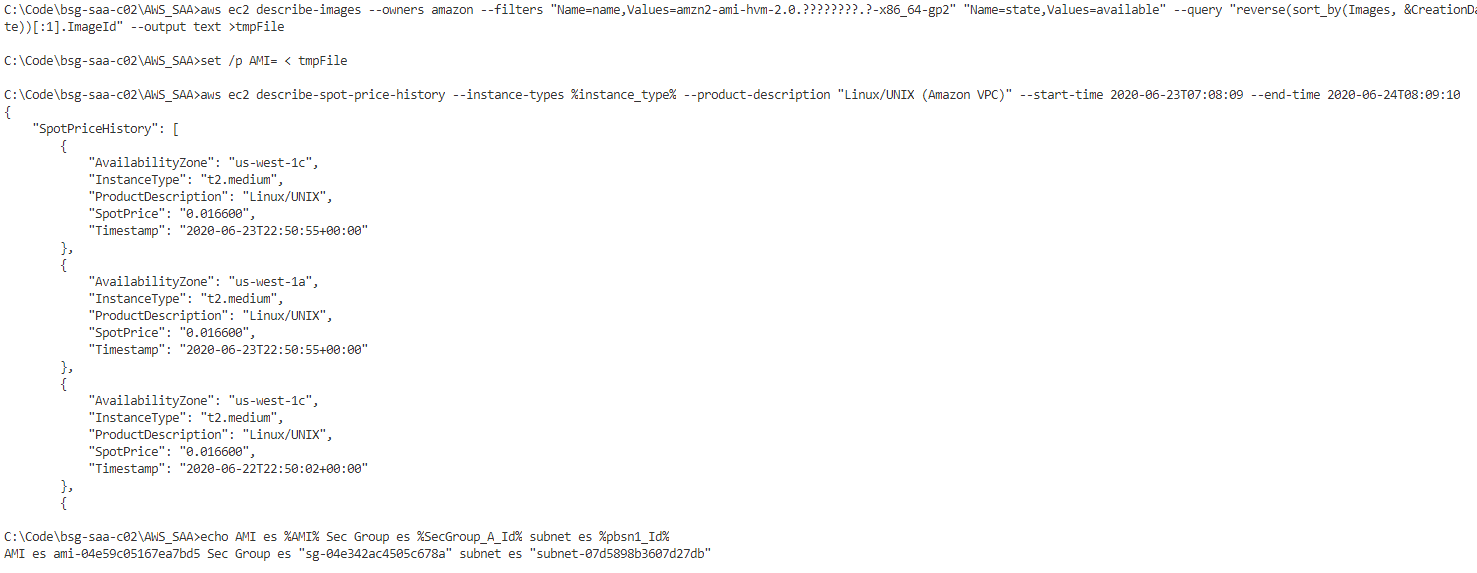
rem Revisar cuales son las instancias ejecutandose

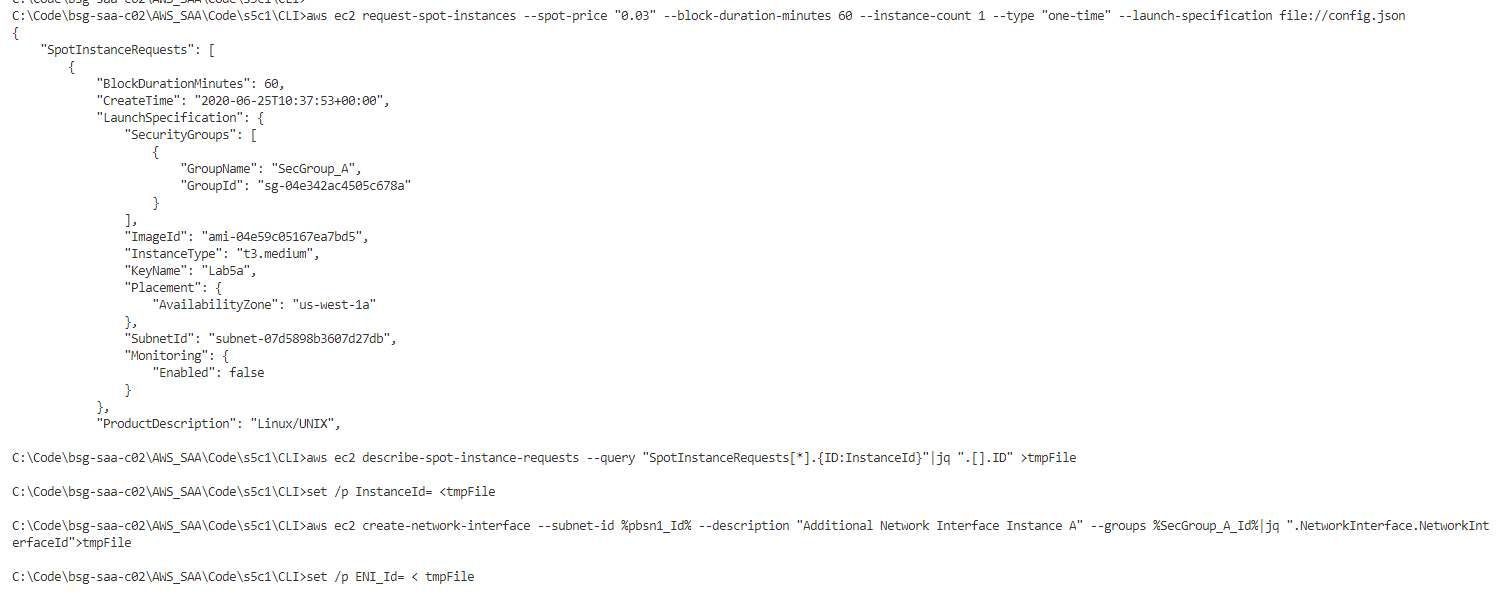
aws ec2 describe-spot-instance-requests --query "SpotInstanceRequests[\*].{ID:InstanceId}"|jq ".[].ID" >tmpFile

rem Comparar que dato se envian del JSON a la linea de comando como se ve aqui y mirar la comparativa de precios

rem aws ec2 run-instances --image-id %AMI% --count 1 --instance-type t2.medium --key-name Lab5a --security-group-ids %SecGroup\_A\_Id% --subnet-id %pbsn1\_Id% --tag-specifications "ResourceType=instance,Tags=[{Key=ServerName,Value=A}]"  --user-data file://bootstrap.sh |jq ".Instances.InstanceId" >tmpFile

set /p InstanceId= <tmpFile





### Create ENI, EIP and associate it

aws ec2 create-network-interface --subnet-id %pbsn1\_Id% --description "Additional Network Interface Instance A" --groups %SecGroup\_A\_Id%|jq ".NetworkInterface.NetworkInterfaceId">tmpFile

set /p ENI\_Id= < tmpFile

aws ec2 attach-network-interface --network-interface-id %ENI\_Id% --instance-id %InstanceId% --device-index 1

rem Obtener una EIP para asignarla a la ENI, se puede asociar directamente a la instancia

aws ec2 allocate-address --domain vpc |jq ".AllocationId" >tmpFile

set /p EIP\_for\_EC2= < tmpFile

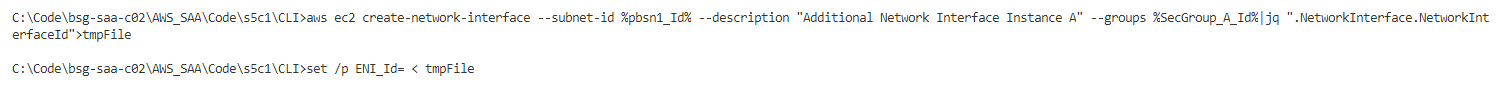
aws ec2 associate-address --allocation-id %EIP\_for\_EC2% --network-interface-id %ENI\_Id%

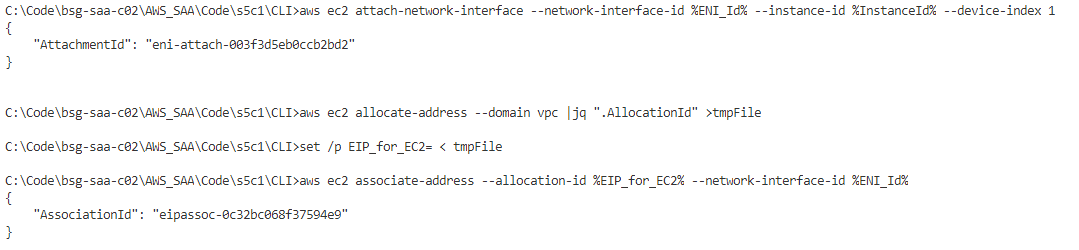
rem Traer Datos especificos de instancia A. Revisar contenido describe-instances y Read\_A.jq ya que es diferente a lo de anteriores laboratorios.

aws ec2 describe-instances | jq -f Read\_Spot.jq

aws ec2 describe-instances | jq -f Read\_Spot.jq|jq ".[0].ENIPublicIpAddress" >tmpFile

set /p A\_IP= < tmpFile





### Review using CLI

Go to section Review Configuration using Putty

## Review Configuration using Putty

rem Ingresar a la instancia publica por SSH, desde Web se toma la IP publica, pueden encontrarla desde la Web Management Console

putty.exe -i "Lab5a.ppk" ec2-user@%A\_IP%

ping 8.8.8.8

rem Debido a que no generamos la ruta por defecto en la otra interface no permite conectarse a Internet

route -n

sudo route del -net 0.0.0.0 gw 10.0.0.1 netmask 0.0.0.0 dev eth0

ping 8.8.8.8

rem Ejecutar la instalacion de Docker

docker ps -a

cat /home/ec2-user/DesdeBootstrap.txt

sudo amazon-linux-extras install docker -y

sudo service docker start

sudo usermod -a -G docker ec2-user

docker ps -a

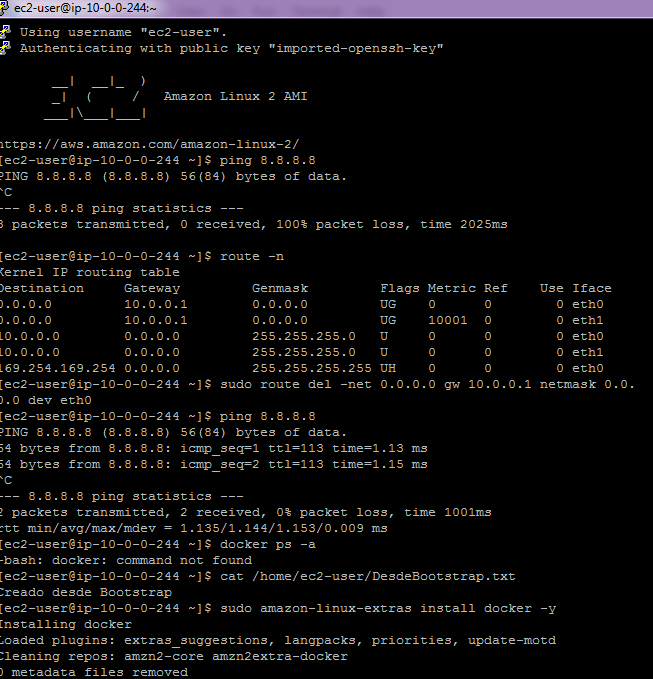
sudo docker run -d -p 80:80 -p 443:443 -h web1 benpiper/mtwa:web

rem Check web with that IP on port 80 and 443

rem Se puede usar tambien un hello world como imagen en vez del anterior

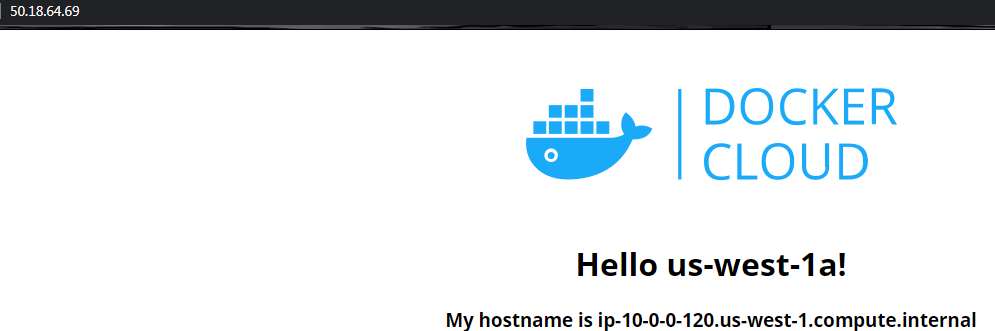
export AZ=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)

sudo docker run -d -p 80:80 -h $HOSTNAME -e NAME=$AZ dockercloud/hello-world

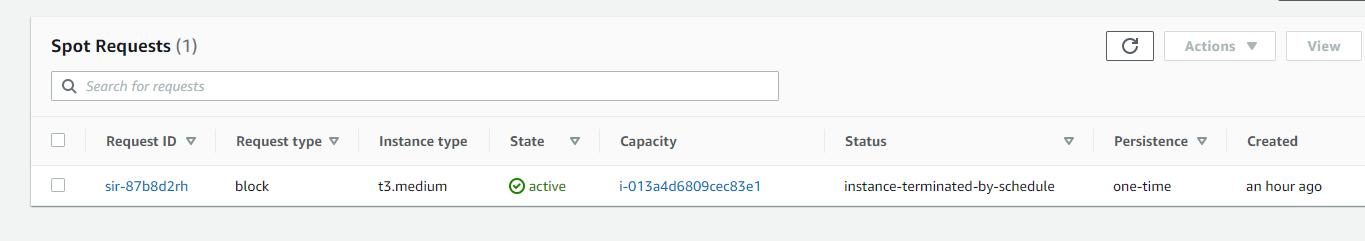


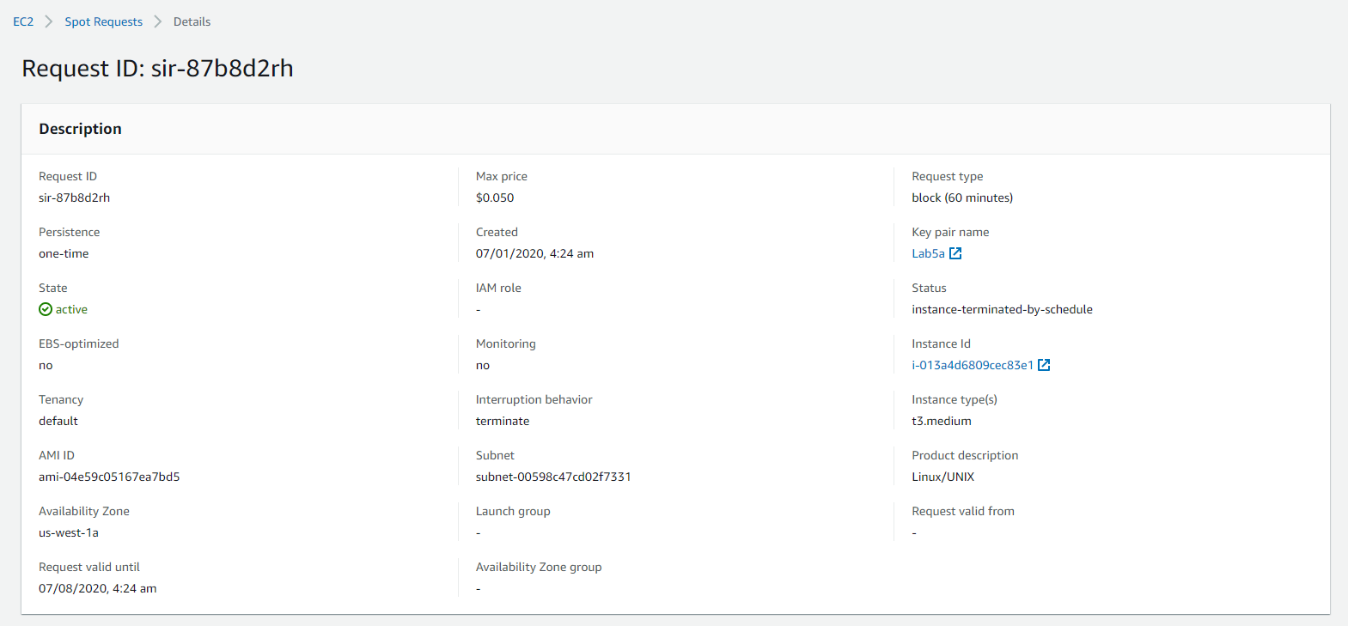


Or



When the schedule block of the Spot instances is fulfilled, then the instance is shutdown and the status of the spot request change its status.





## Clean Resources

### For Command Line (Windows)

rem ----- ELIMINAR RECURSOS ----

aws ec2 terminate-instances --instance-ids "i-0075c60461951e64a"

aws ec2 disassociate-address --public-ip %A\_IP%

aws ec2 release-address --allocation-id %EIP\_for\_EC2%

aws ec2 detach-network-interface --attachment-id "eni-attach-003f3d5eb0ccb2bd2"

aws ec2 delete-network-interface --network-interface-id %ENI\_Id%

aws ec2 delete-security-group --group-id %SecGroup\_A\_Id%

aws ec2 detach-internet-gateway --internet-gateway-id %IGW\_Id% --vpc-id %vpcn\_Id%

aws ec2 delete-internet-gateway --internet-gateway-id %IGW\_Id%

aws ec2 delete-subnet --subnet-id %pbsn1\_Id%

aws ec2 delete-route-table --route-table-id %Public\_RT\_Id%

aws ec2 delete-vpc --vpc-id %vpcn\_Id%

aws ec2 delete-key-pair --key-name Lab5a

# Evidences to send

To have a review, the student has to send some screenshots to instructor email:

1. The last screenshot of [Check Price and make a Spot Request](#_Check_Price_and) (Web Console) where you see that Spot Request is fulfilled.
2. The last screenshot of [Review using Web Console](#_Review) (Web Console) where you notice 2 Private IP and 1 Public IP using Web SSH Connection.
3. Image from Browser of you public Server using docker maybe using benpiper/mtwa:web or dockercloud/hello-world. Those images appear on [Review Configuration using Putty](#_Review_Configuration_using).