Contents

[Purpose 2](#_Toc54593143)

[General Diagram 2](#_Toc54593144)

[Prerequisites 2](#_Toc54593145)

[Lab 5B: ALB 3](#_Toc54593146)

[Lab 5B using Web Management Console 3](#_Toc54593147)

[Create Network Infrastructure, Routing Tables and IGW. 3](#_Toc54593148)

[Running Instance, Security Groups and Running the Code on SSH 3](#_Toc54593149)

[Create Security Group for ALB 3](#_Toc54593150)

[Create target groups 3](#_Toc54593151)

[Create ALB 7](#_Toc54593152)

[Create Additional Listeners for Ports 13](#_Toc54593153)

[Create Listener Rules for Routing Paths 15](#_Toc54593154)

[Lab 5B using Command Line (Windows) 17](#_Toc54593155)

[Create Network Infrastructure, Instances and its security groups 17](#_Toc54593156)

[Running the code on SSH 21](#_Toc54593157)

[Create target groups, ALB and listeners 23](#_Toc54593158)

[Able Routing Paths 26](#_Toc54593159)

[Clean Resources 27](#_Toc54593160)

[For Web Management Console 27](#_Toc54593161)

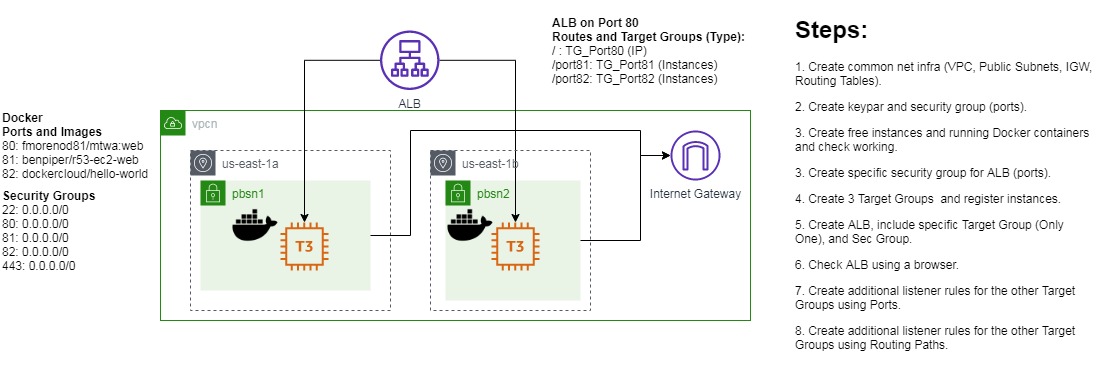
[Evidences to send 27](#_Toc54593162)

# Purpose

Using network infrastructure as base, create a common infrastructure using

# 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.

Labs5c1 have to be done, because you learn how to: Create instances, create and apply security groups, install and run docker.

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 5B: ALB

## Lab 5B using Web Management Console

### Create Network Infrastructure, Routing Tables and IGW.

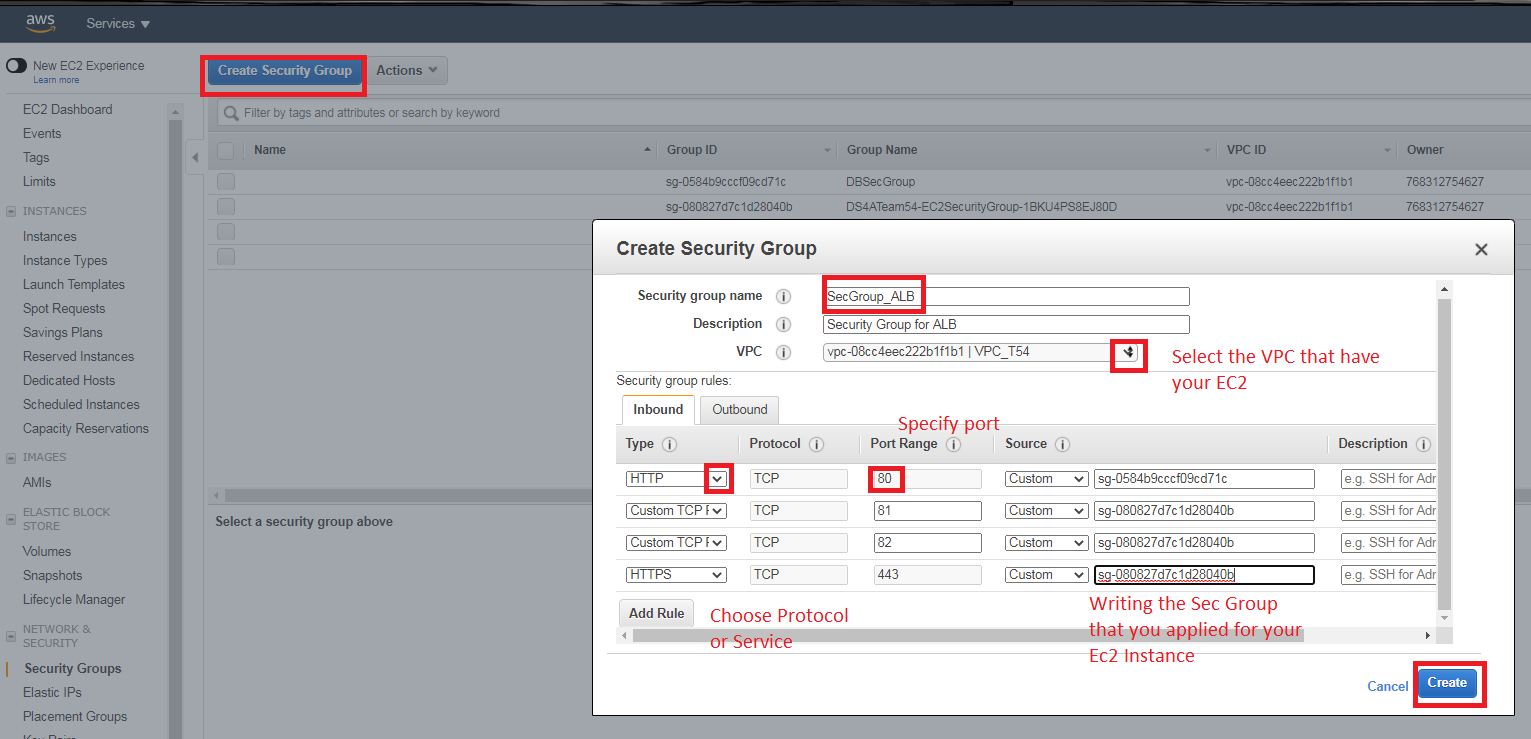
It’s done on Lab4c1.

### Running Instance, Security Groups and Running the Code on SSH

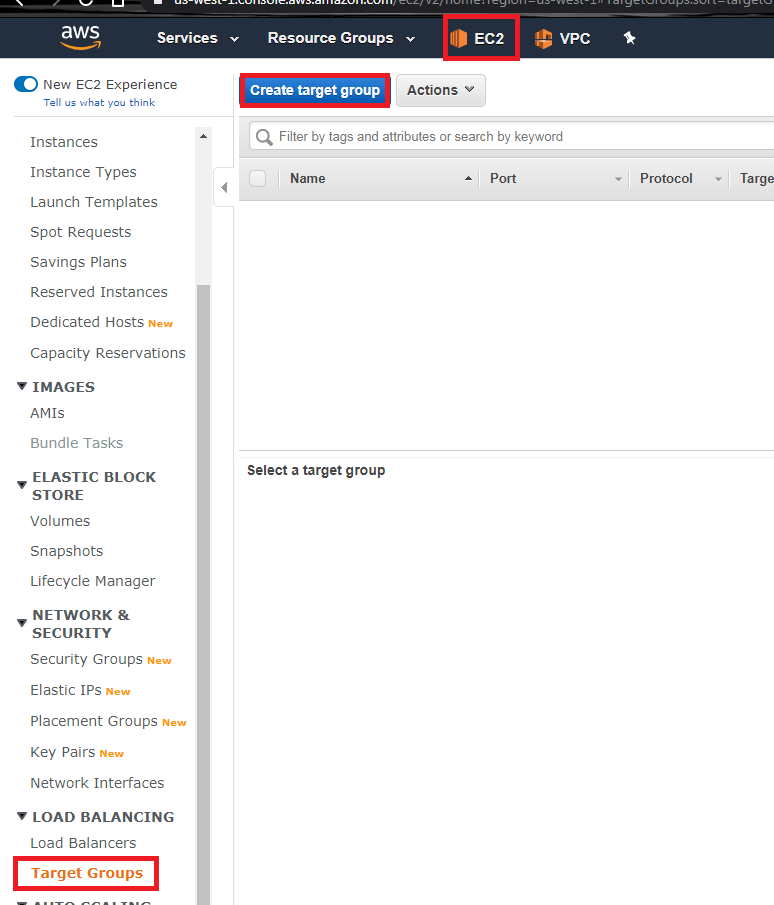
It’s done on Lab5c1.

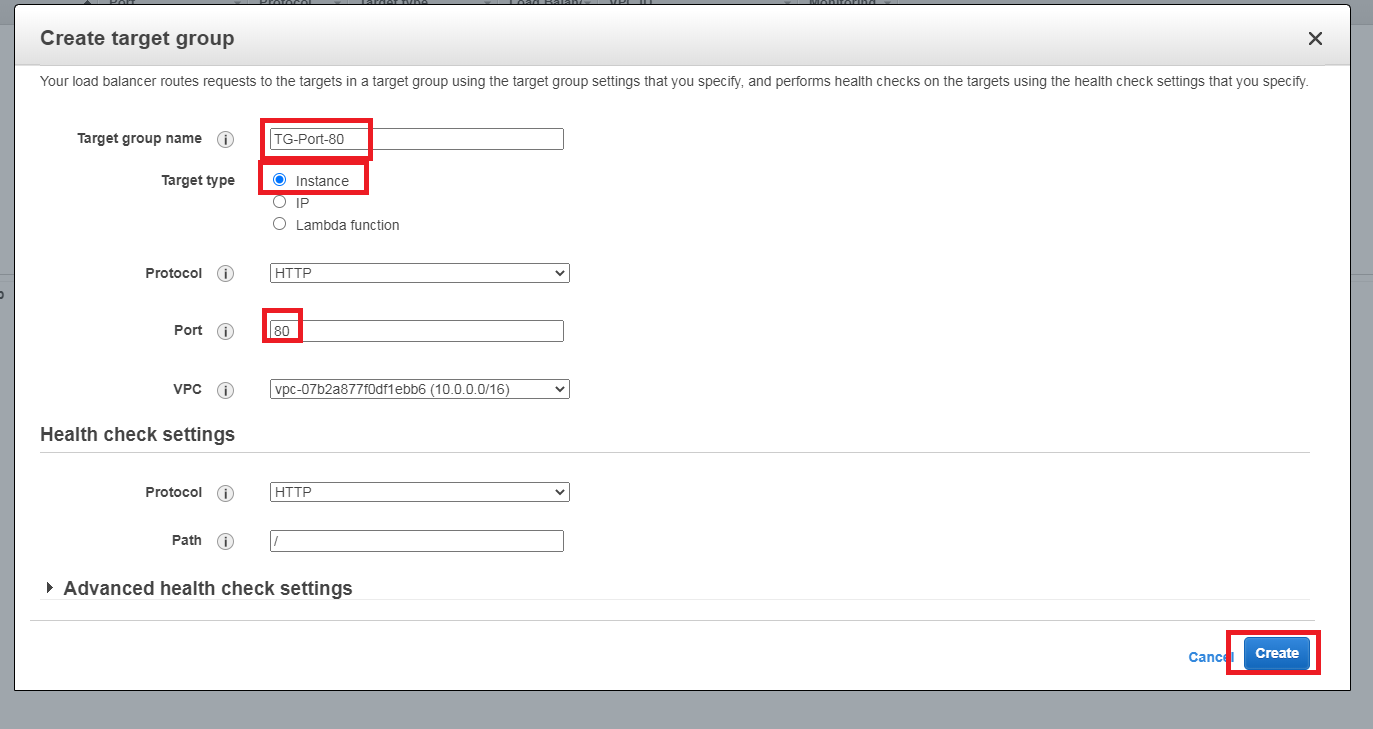
### Create Security Group for ALB

In spite of having a security group for EC2, we have to create a security group for ALB. The theory was done on the course; however, it is better to create this specific case for EC2.

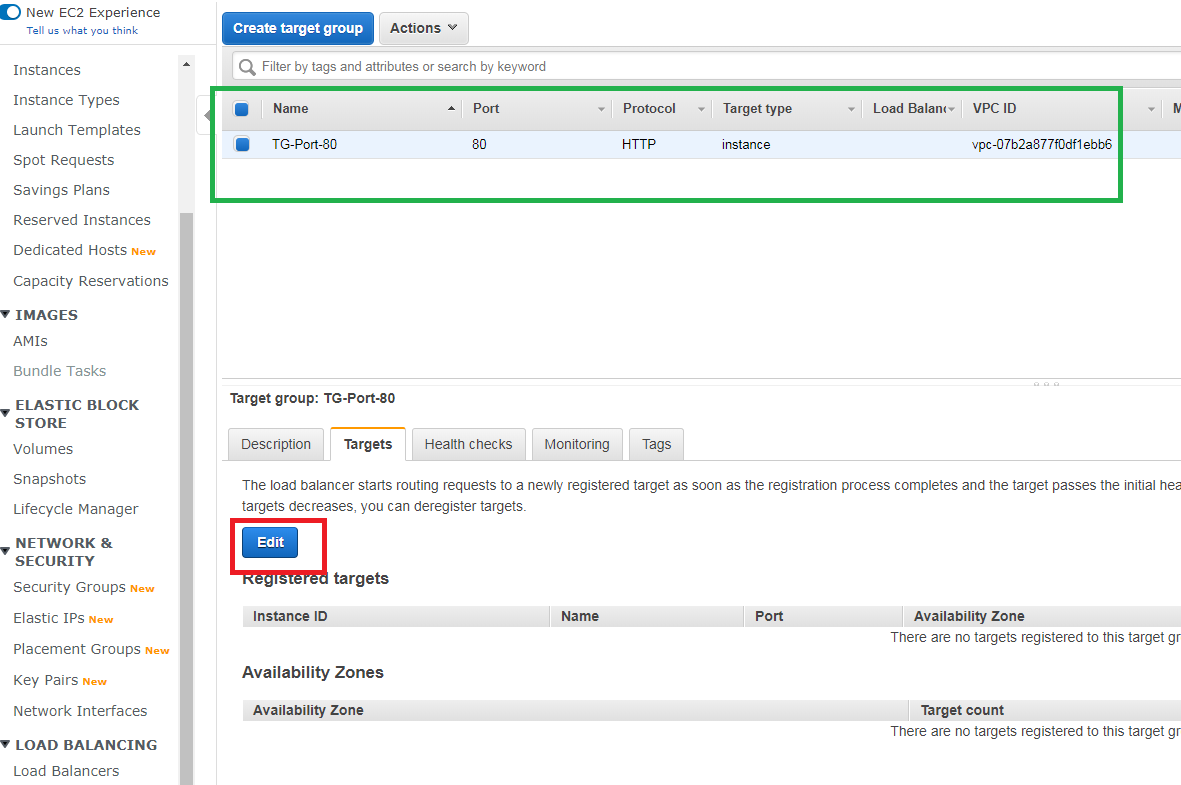
****

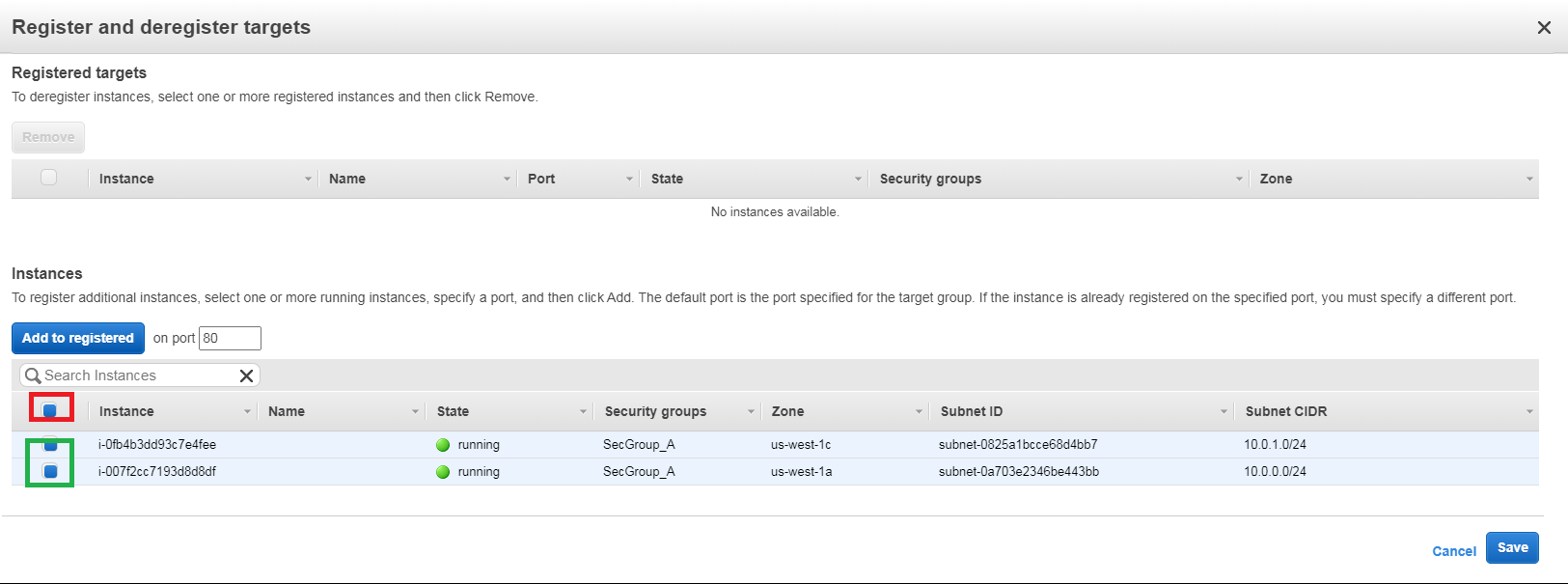
### Create target groups

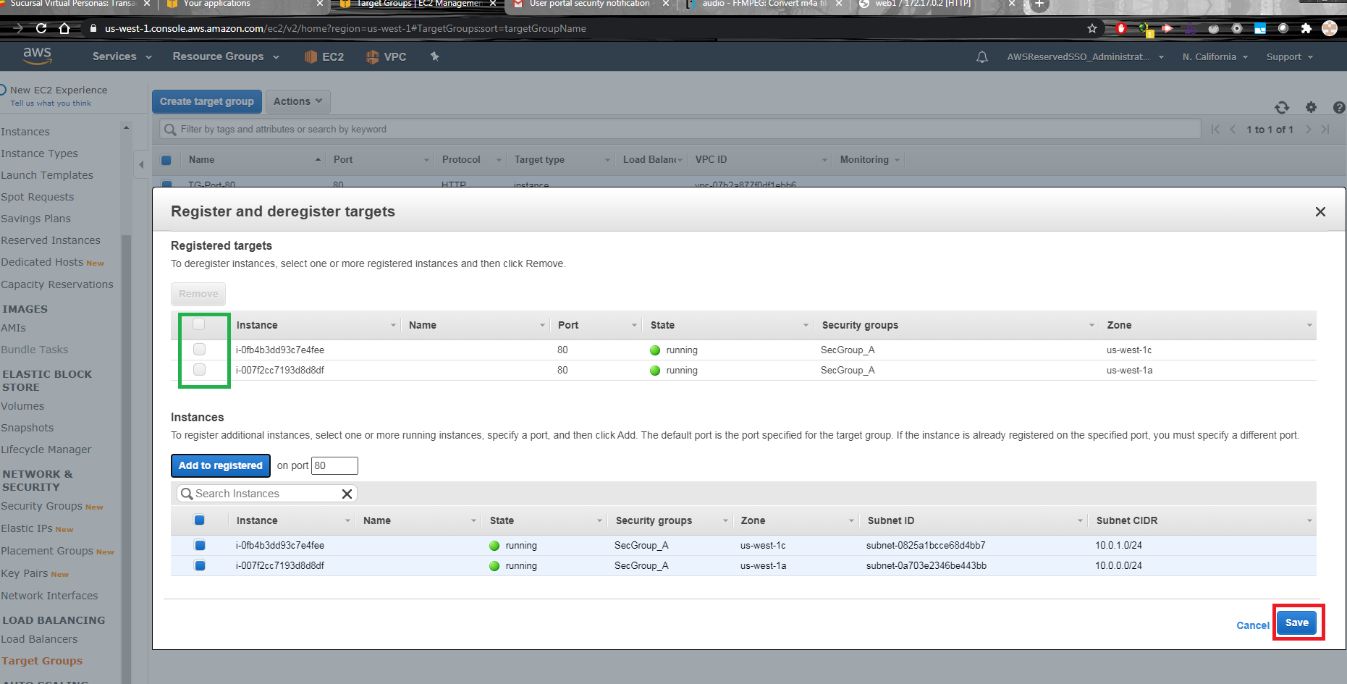




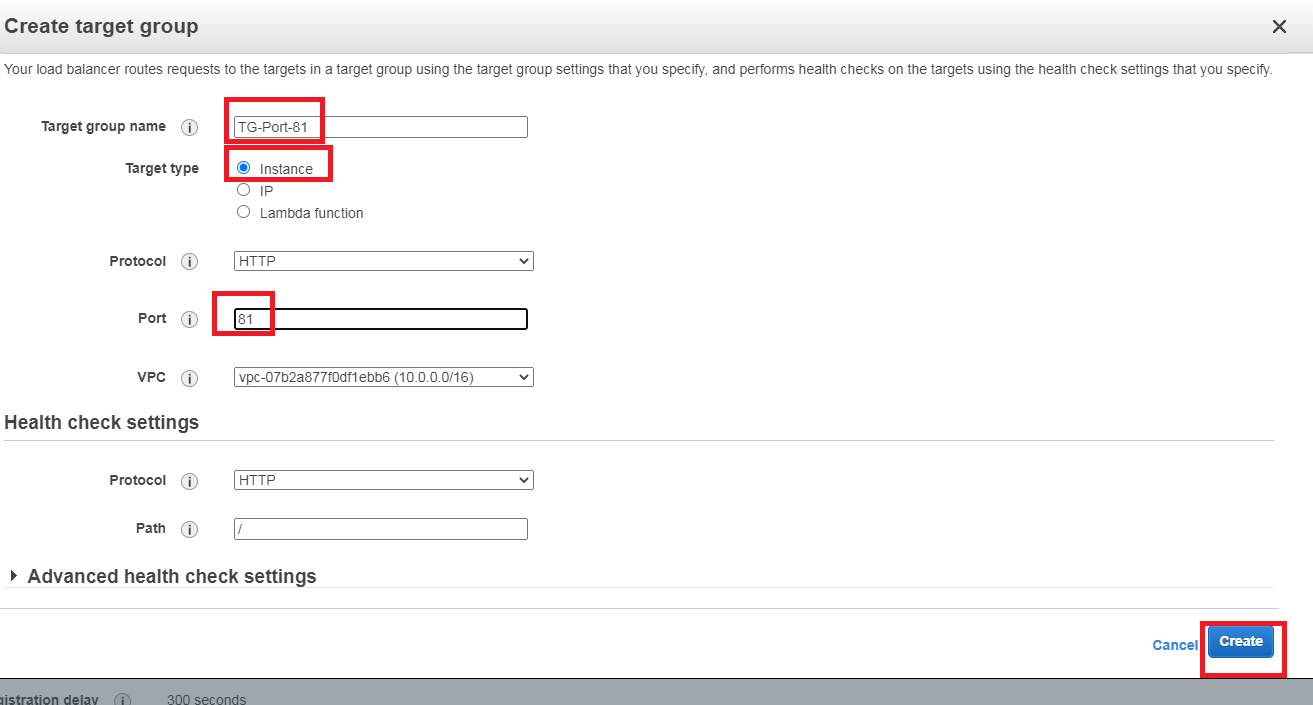
Now its time to register targets, it the same procedure for the remaining Target groups

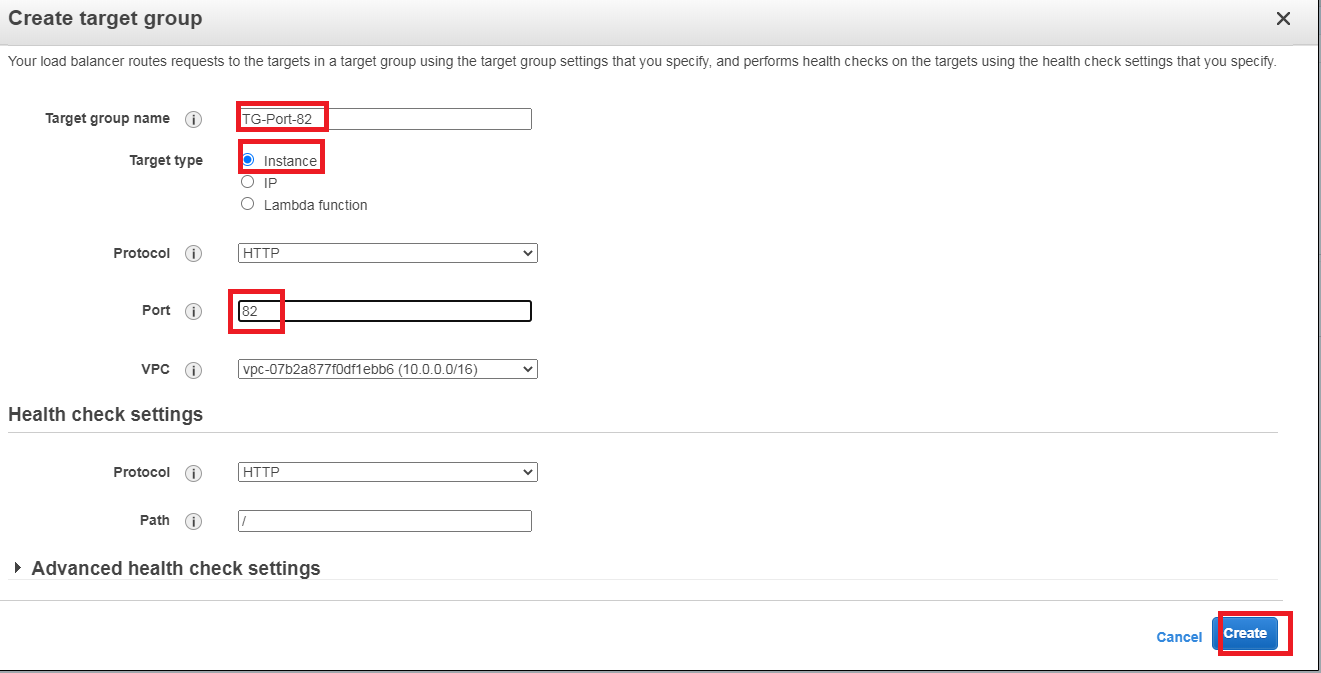




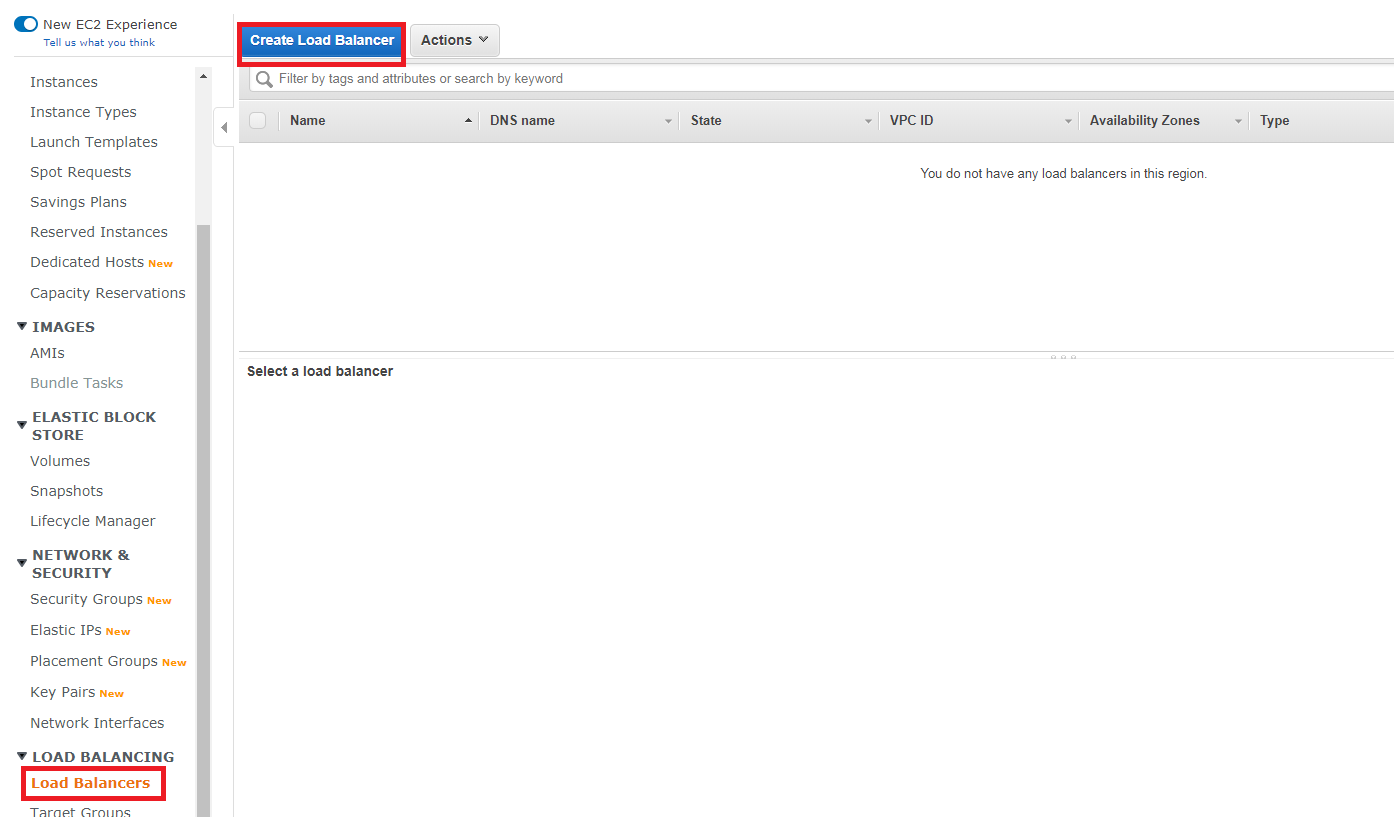


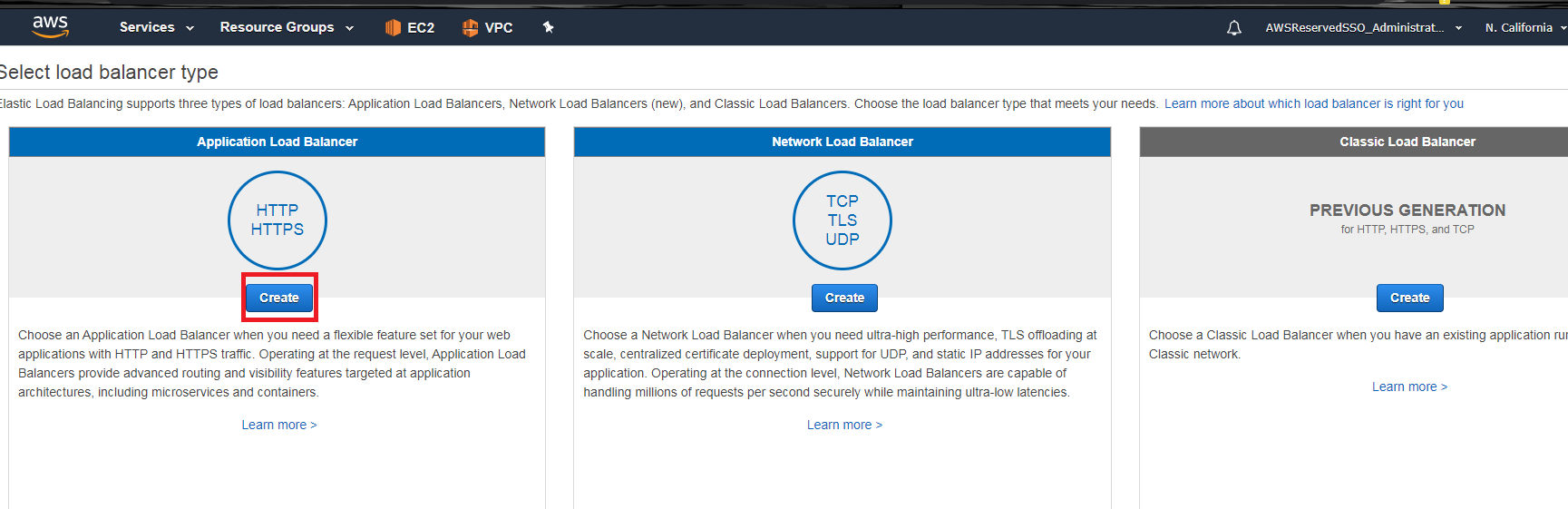
For the remaining TG, review the port and name,

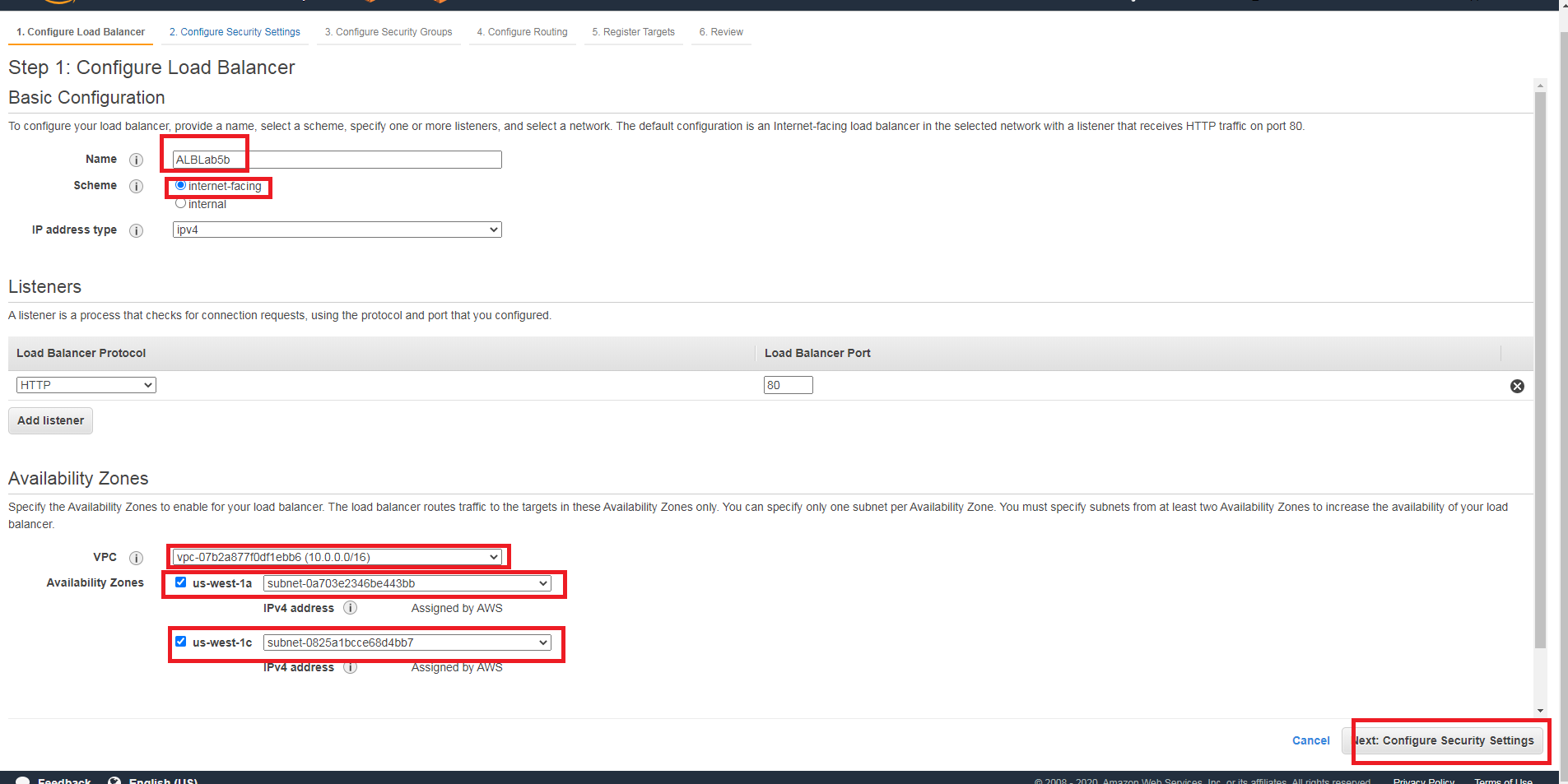


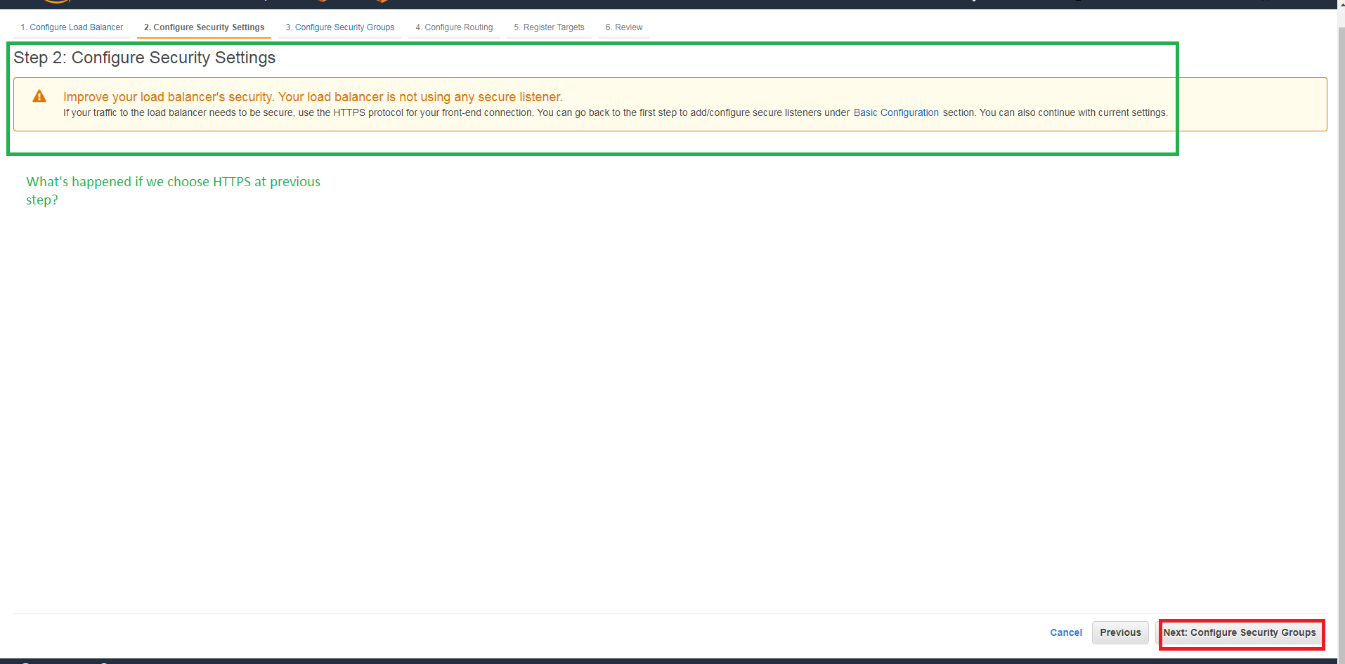


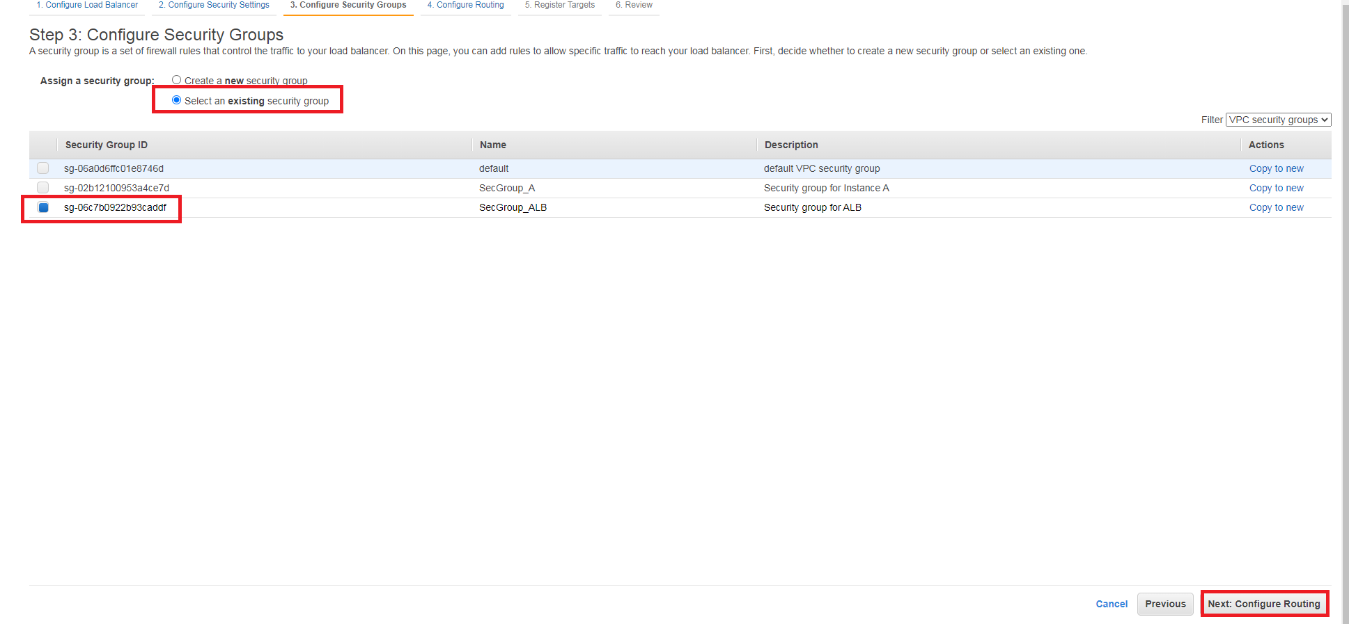
### Create ALB

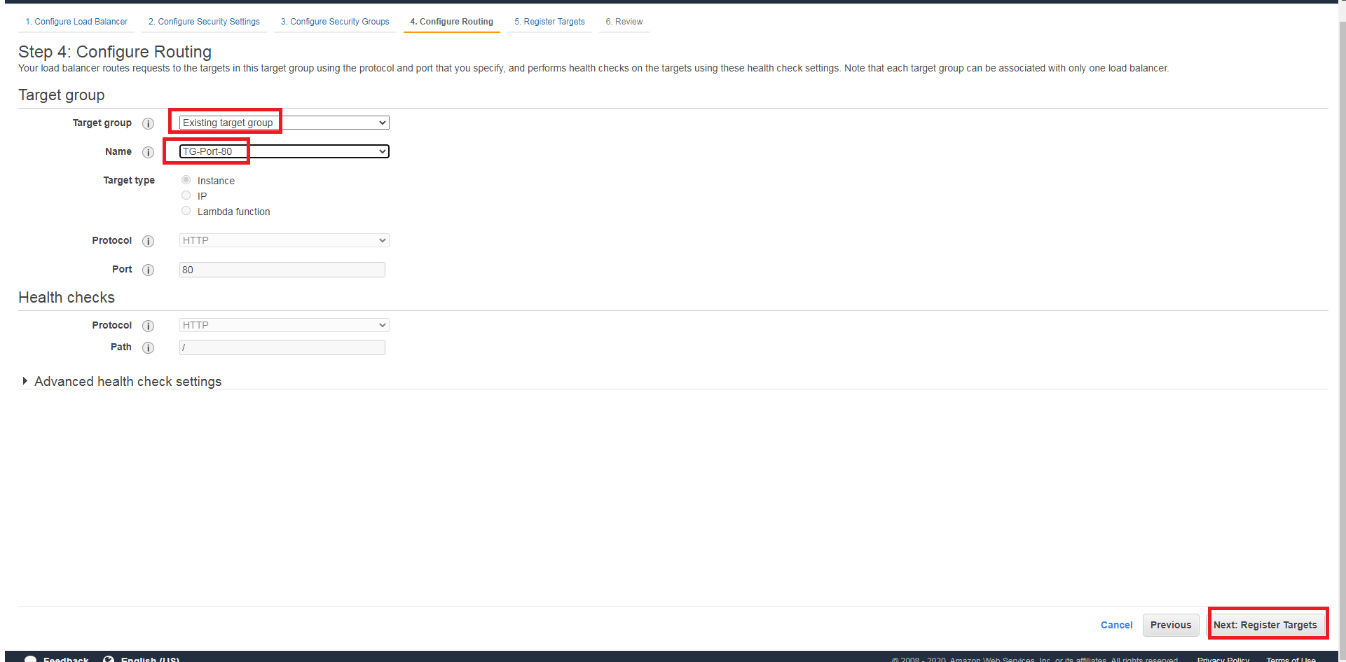


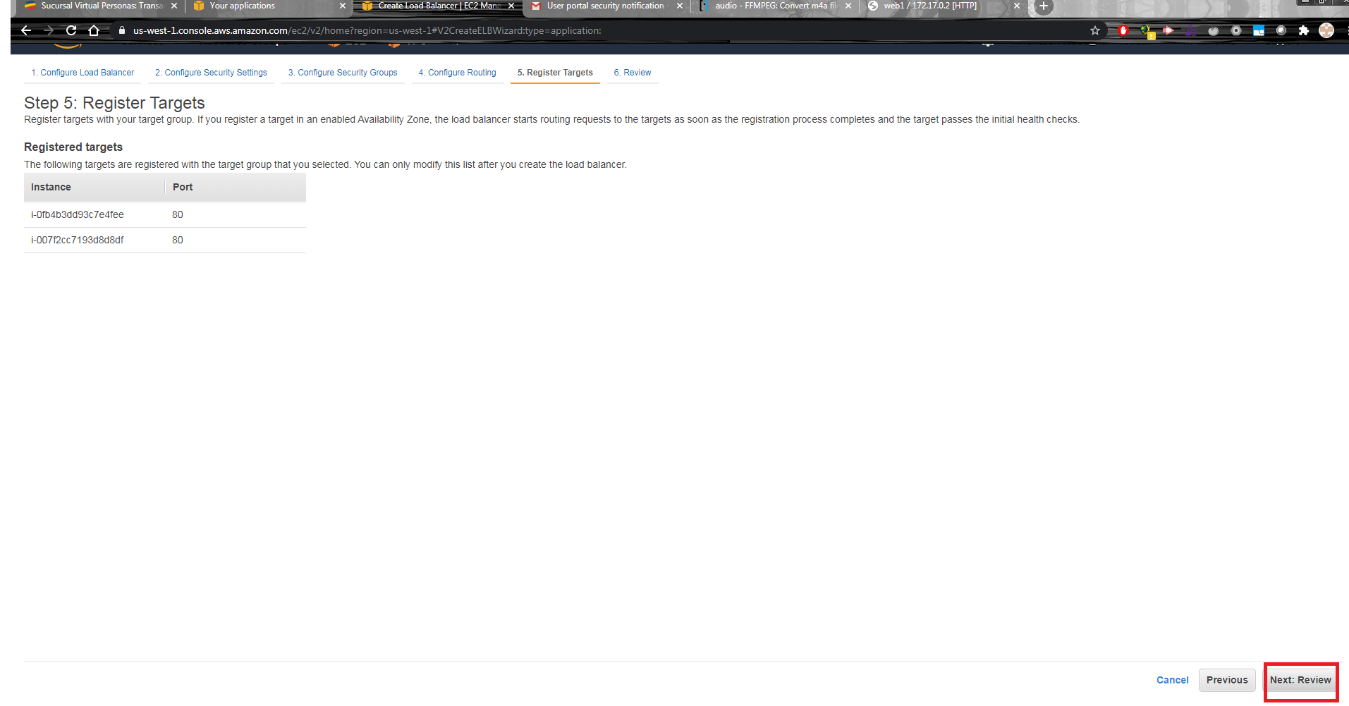


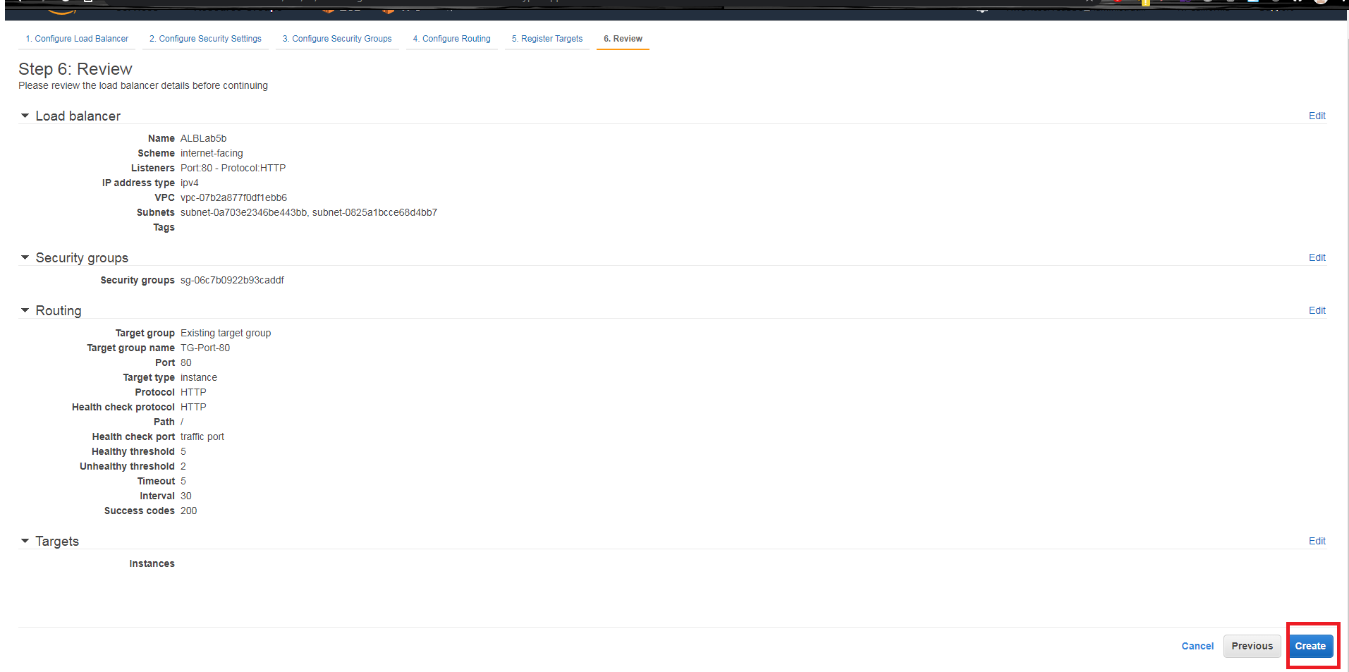


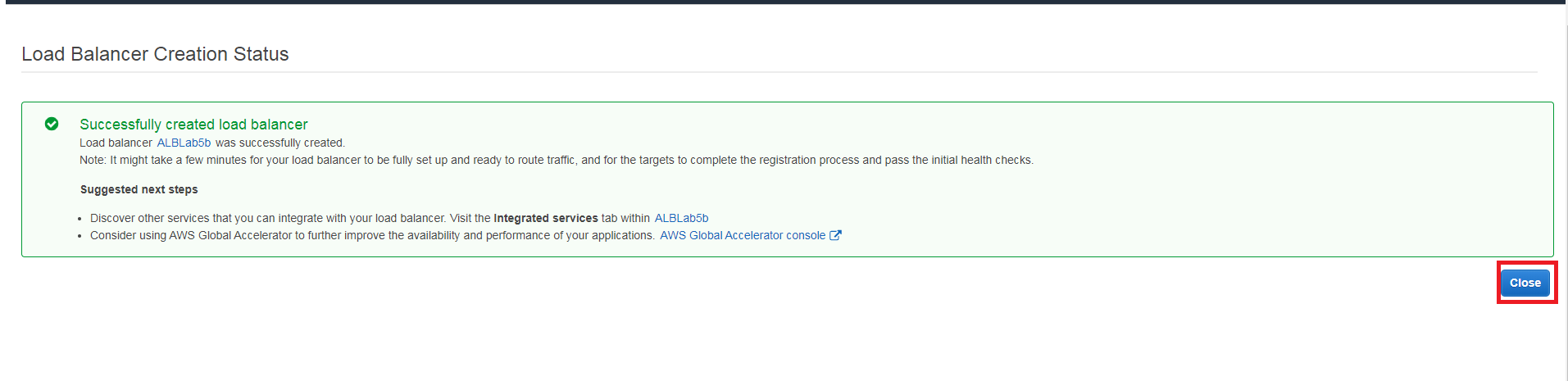


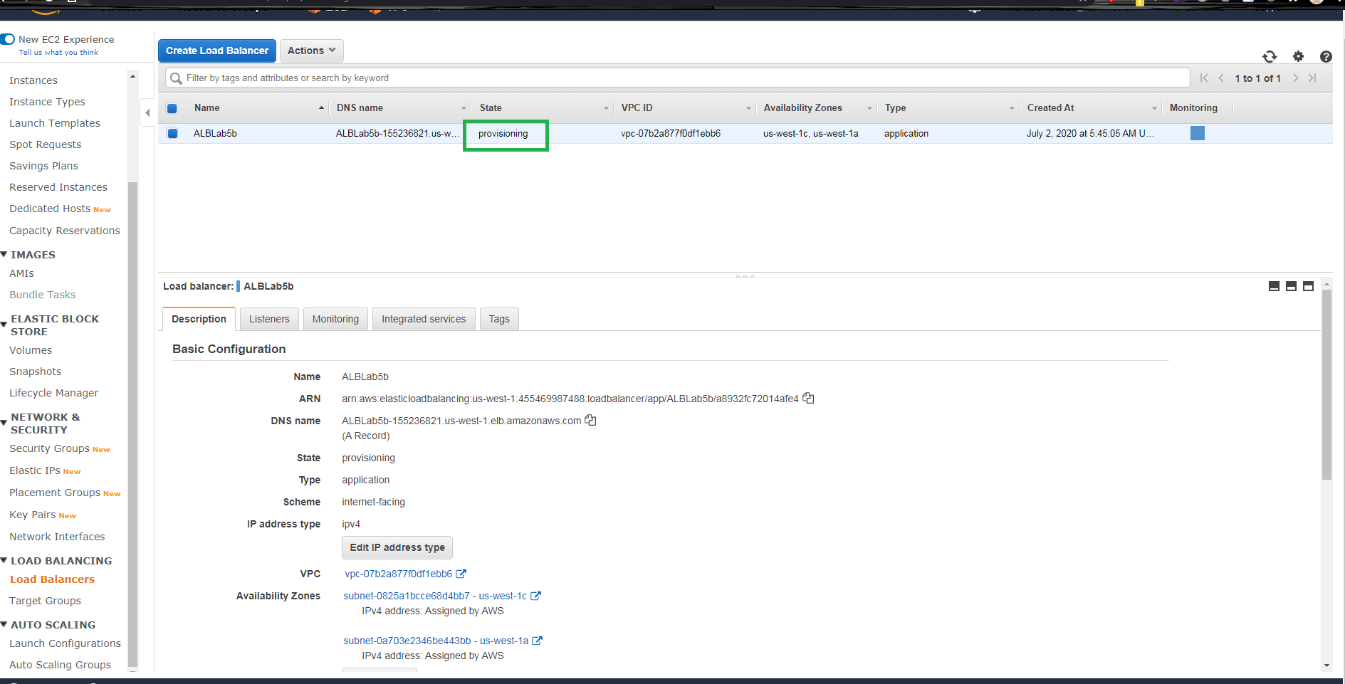


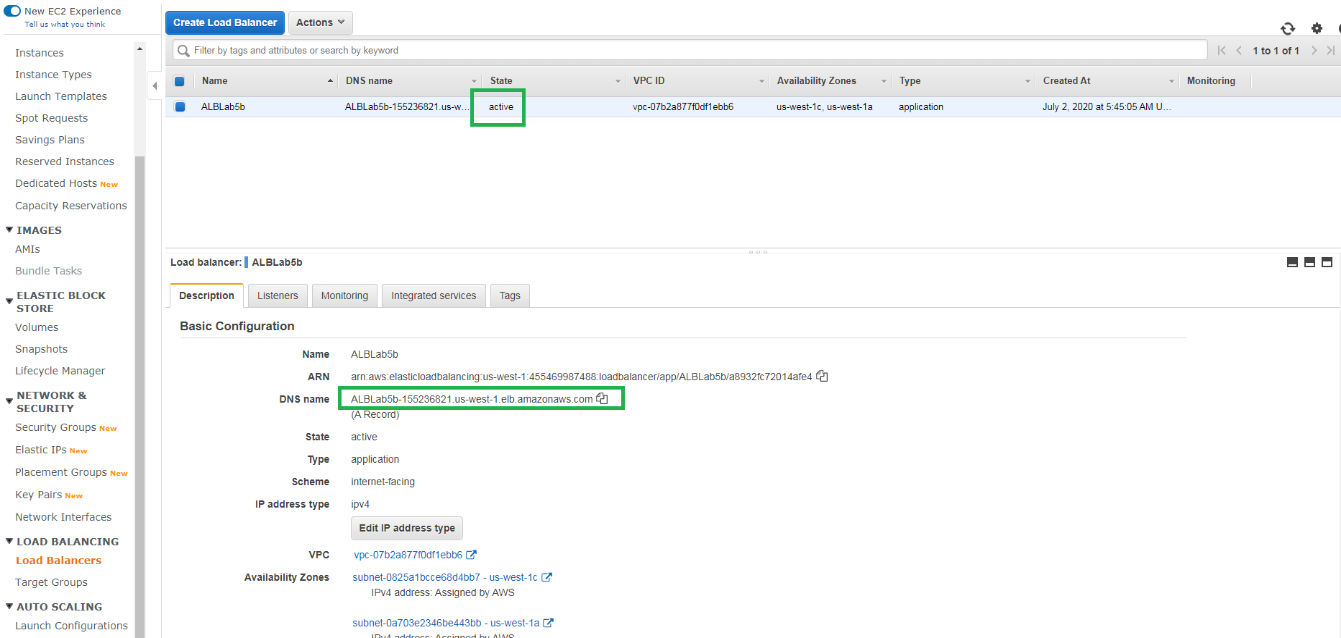




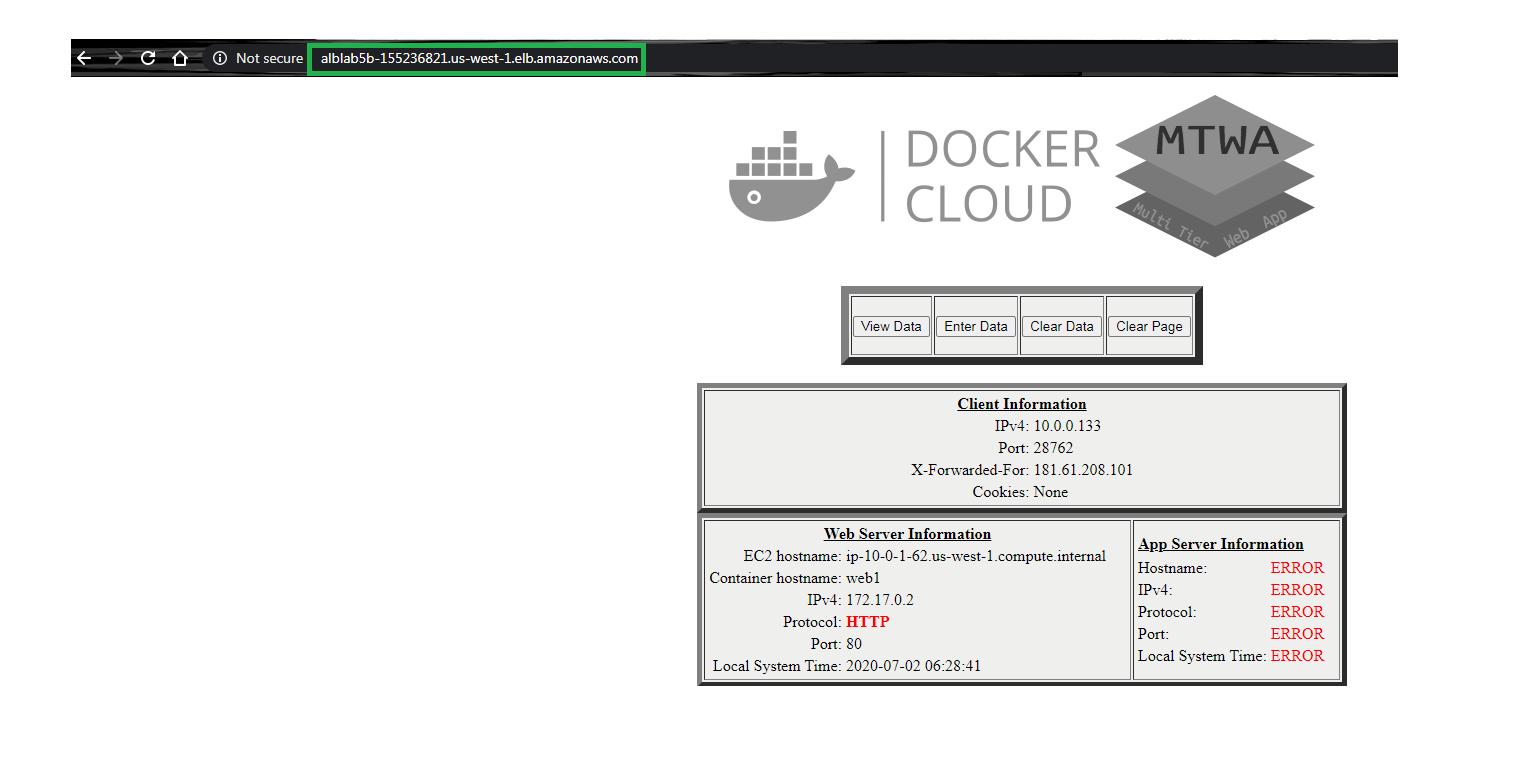




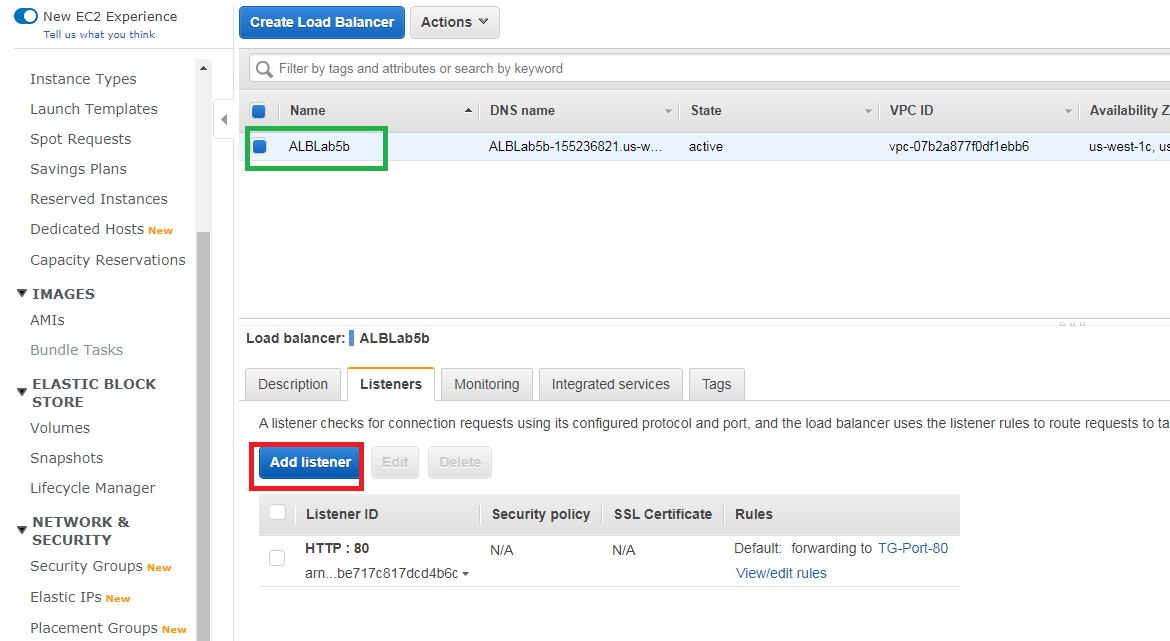


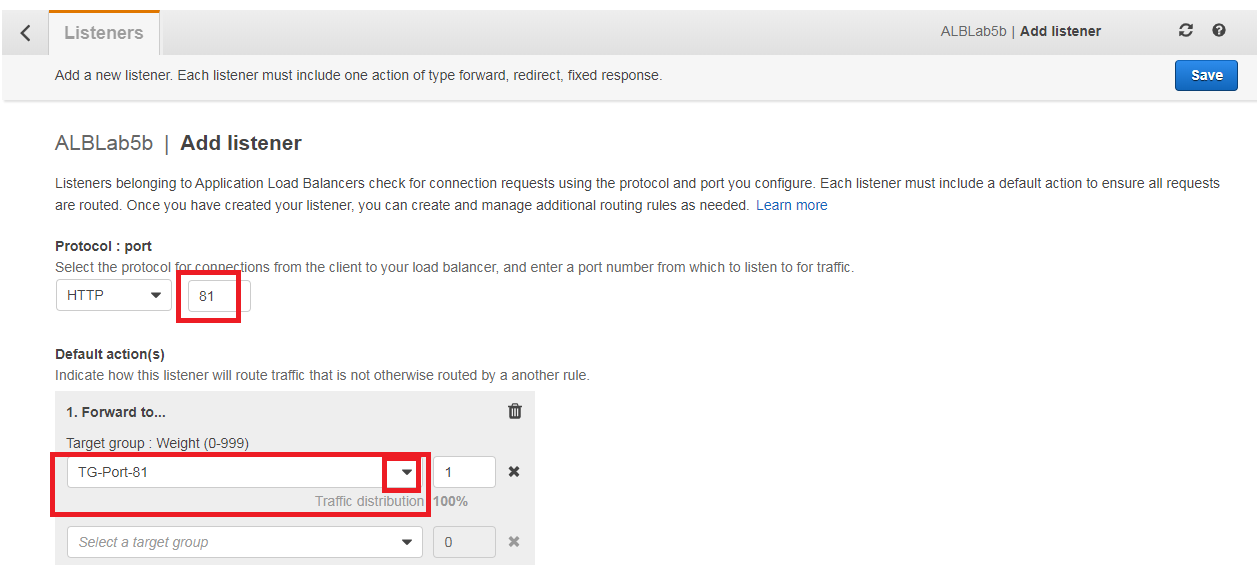


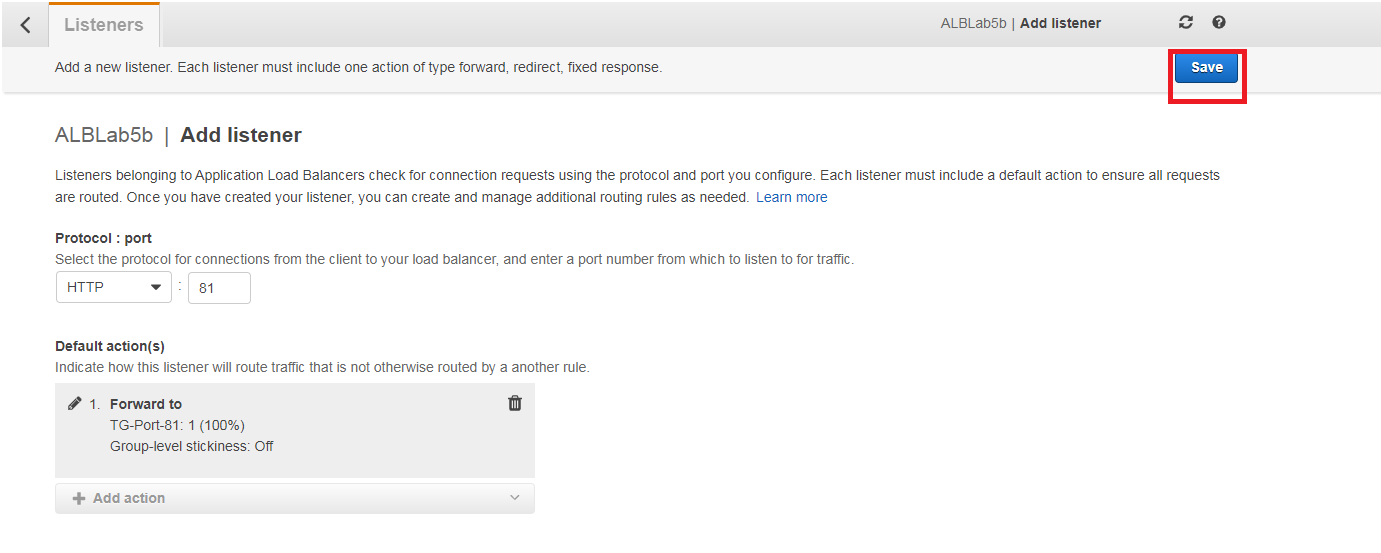
Test it with the DNS name of the balancer,

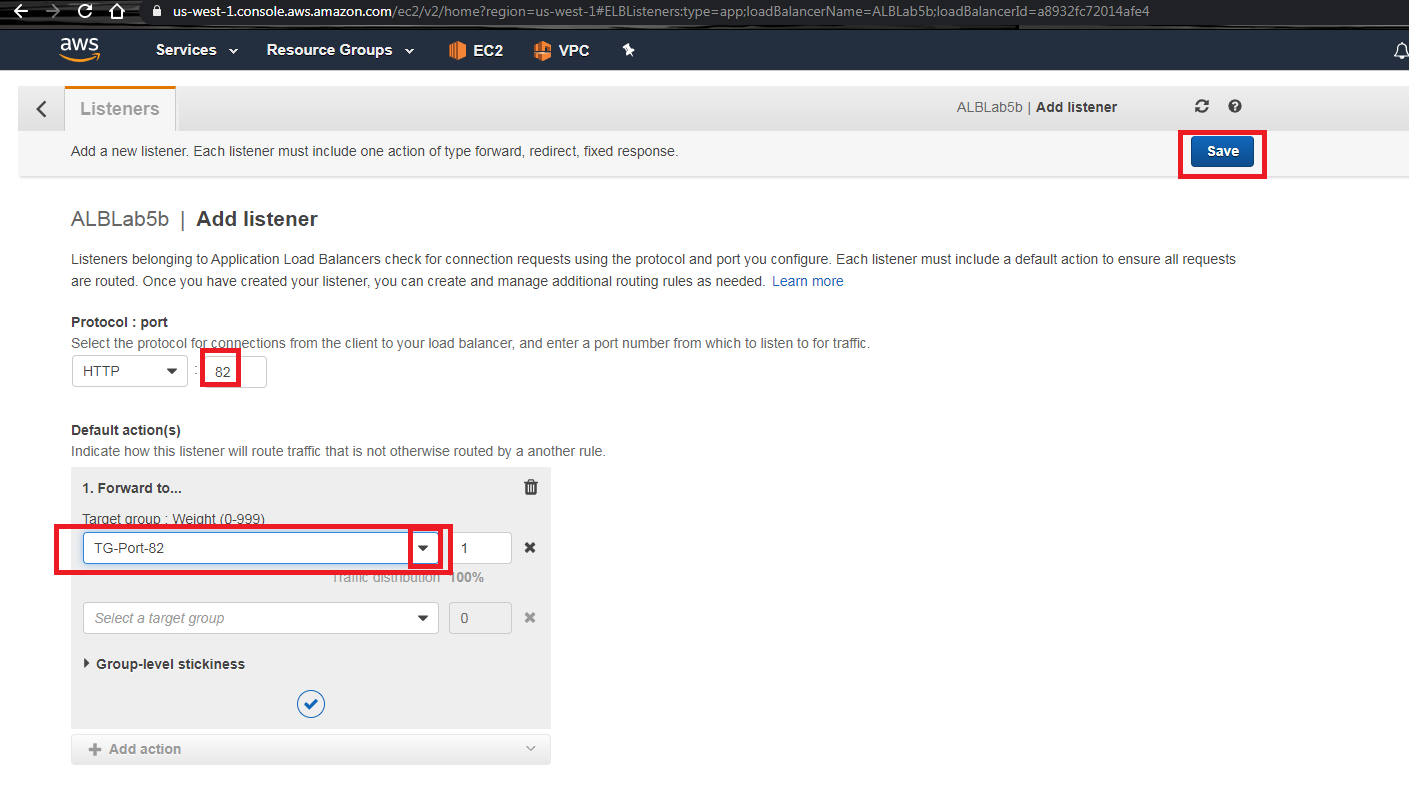


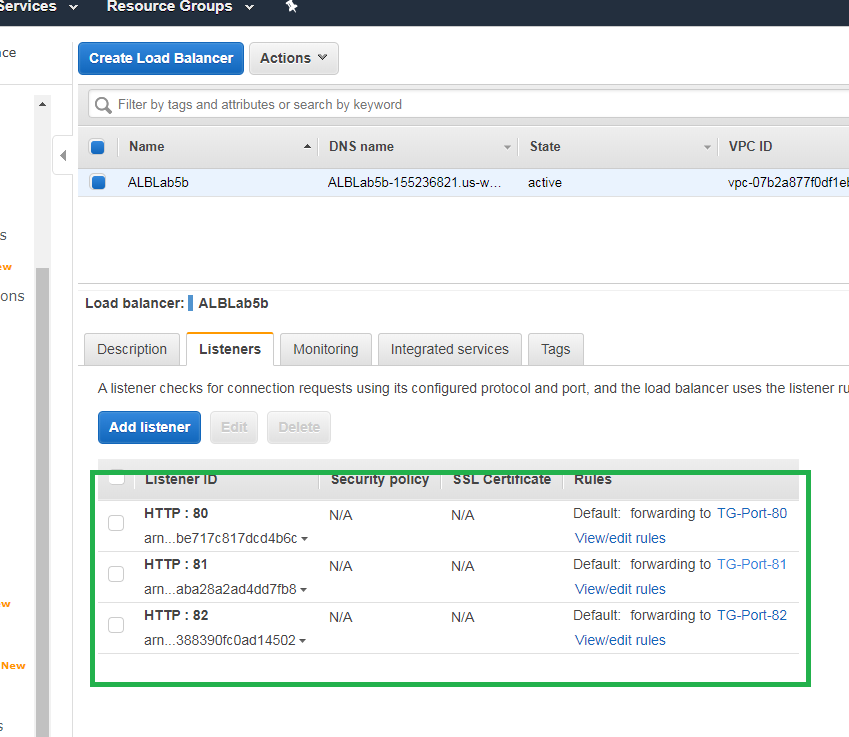
### Create Additional Listeners for Ports

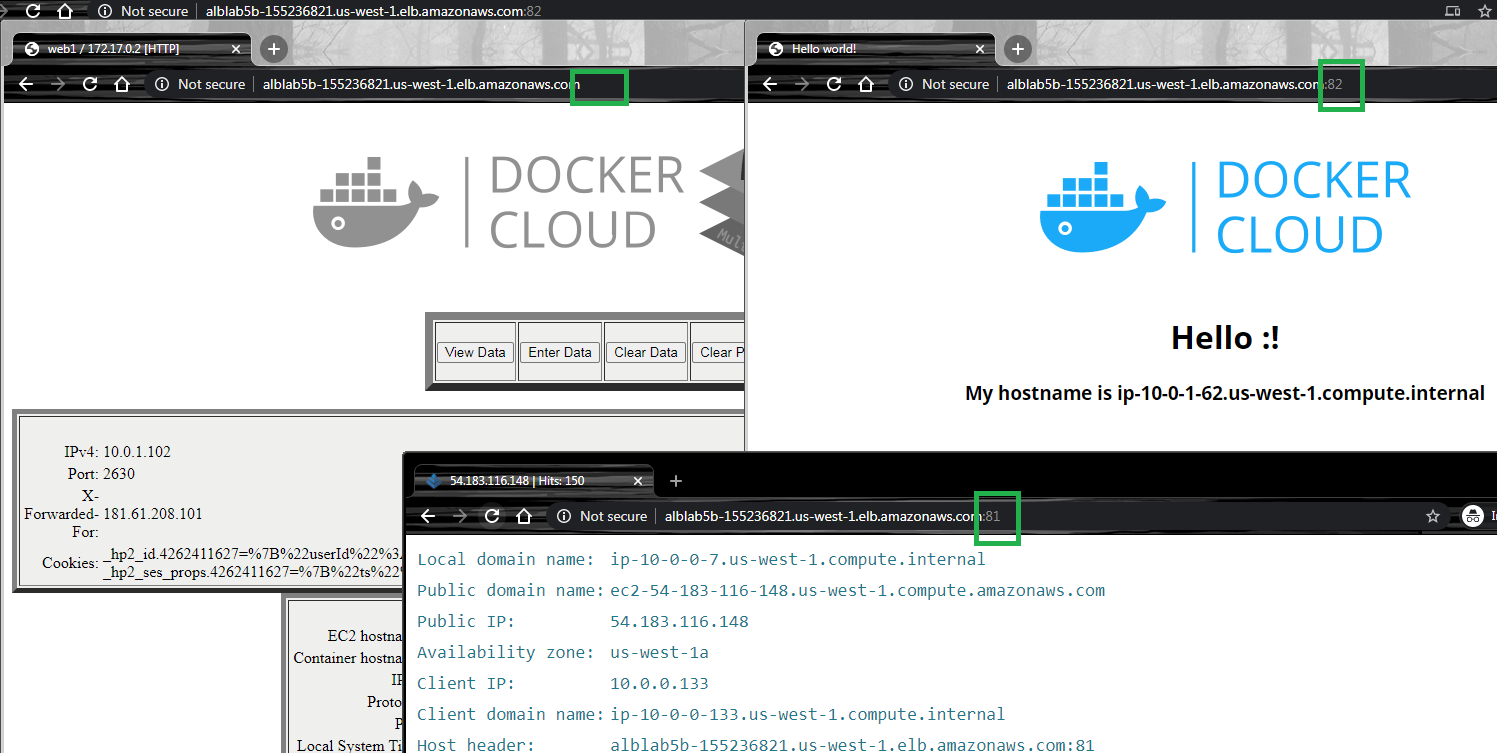




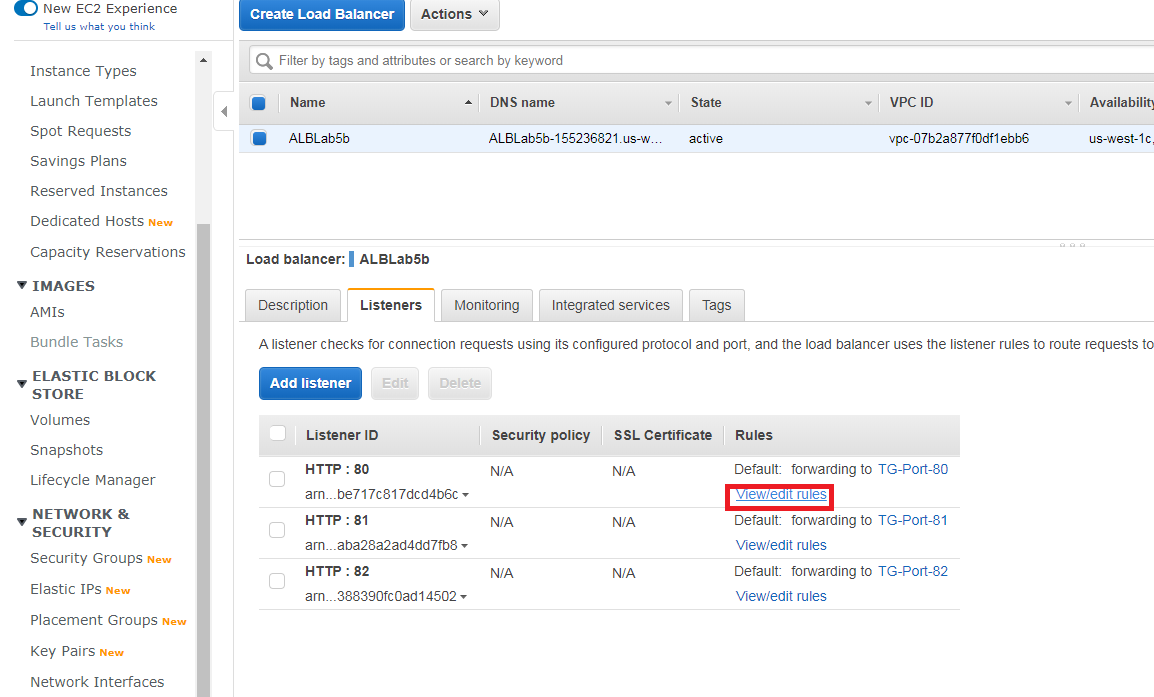


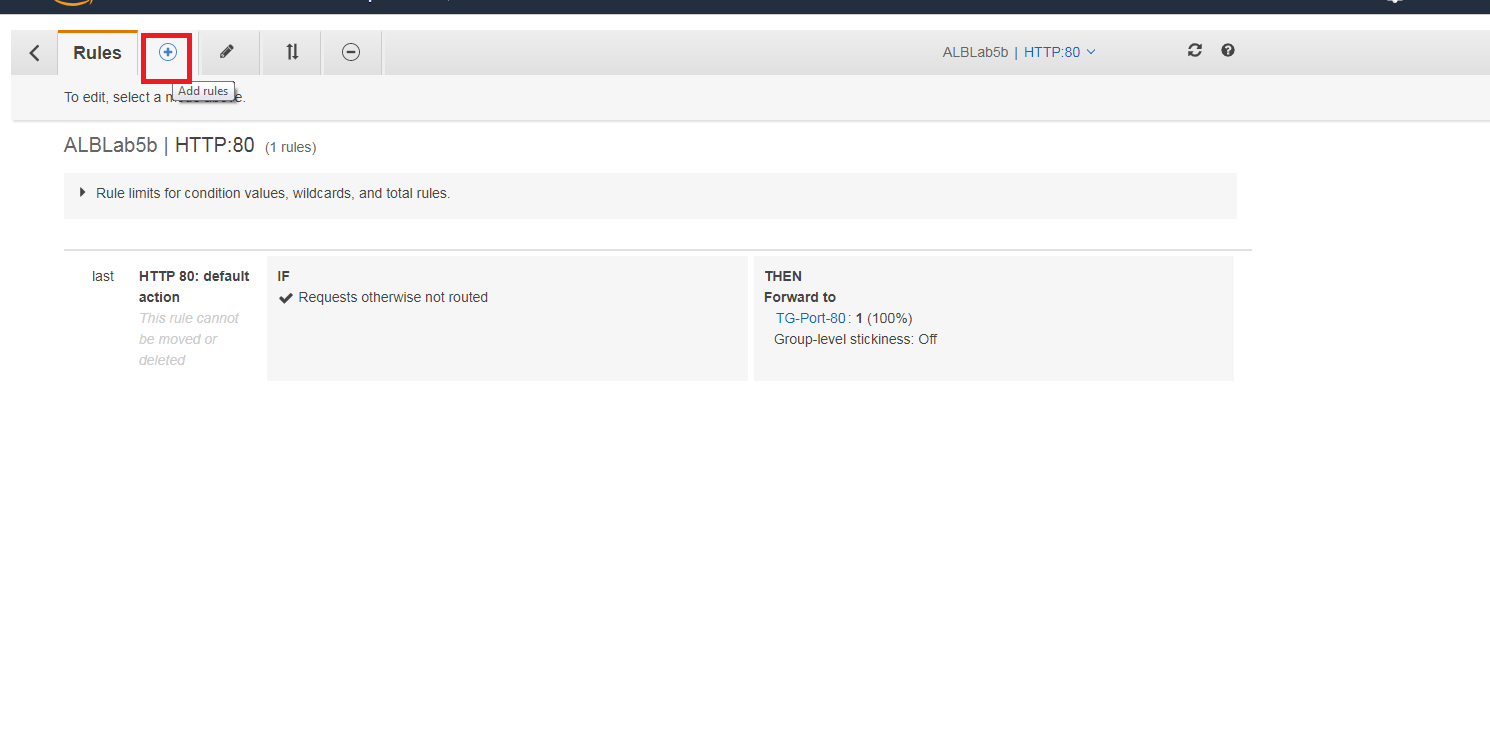


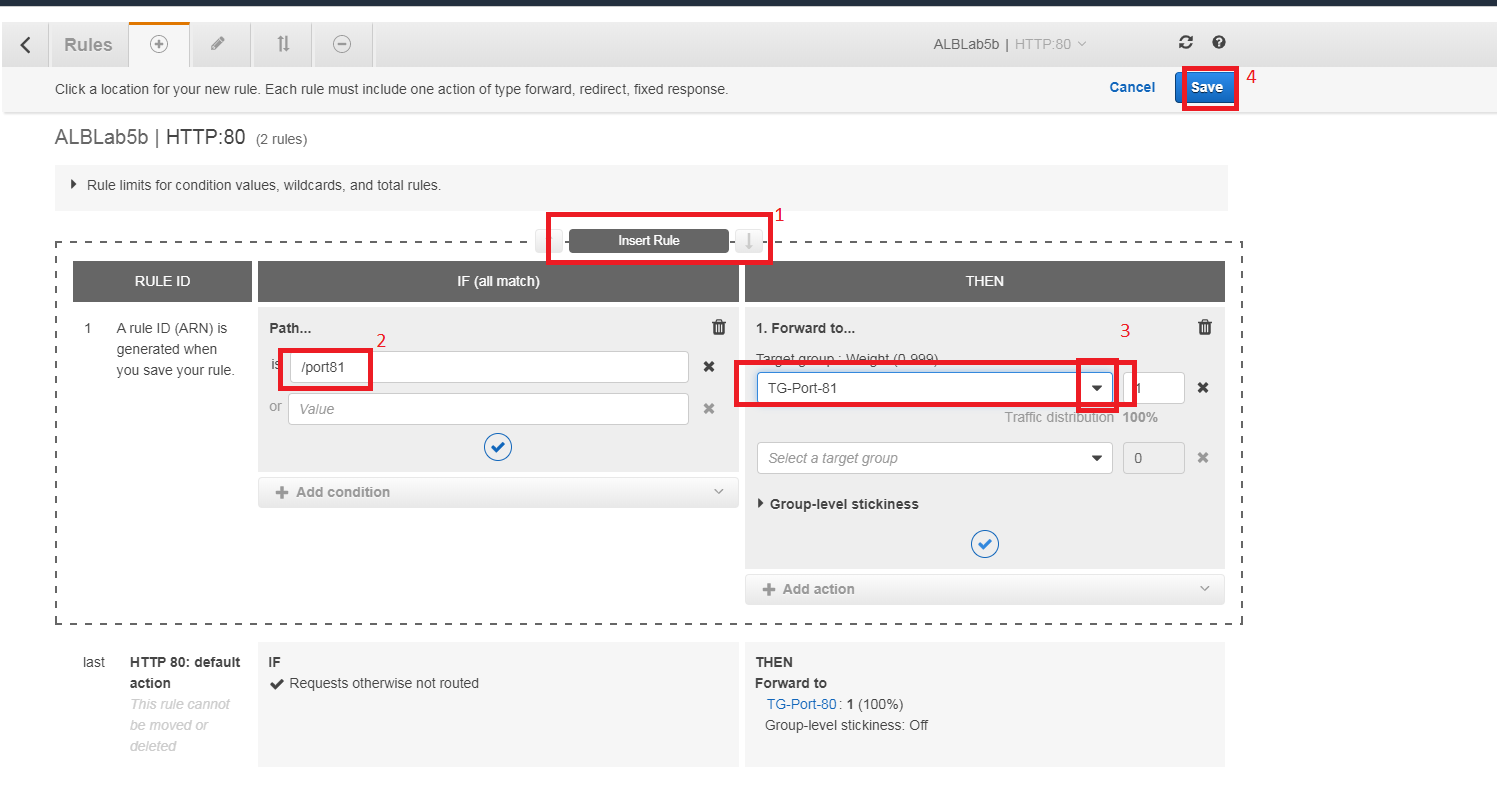


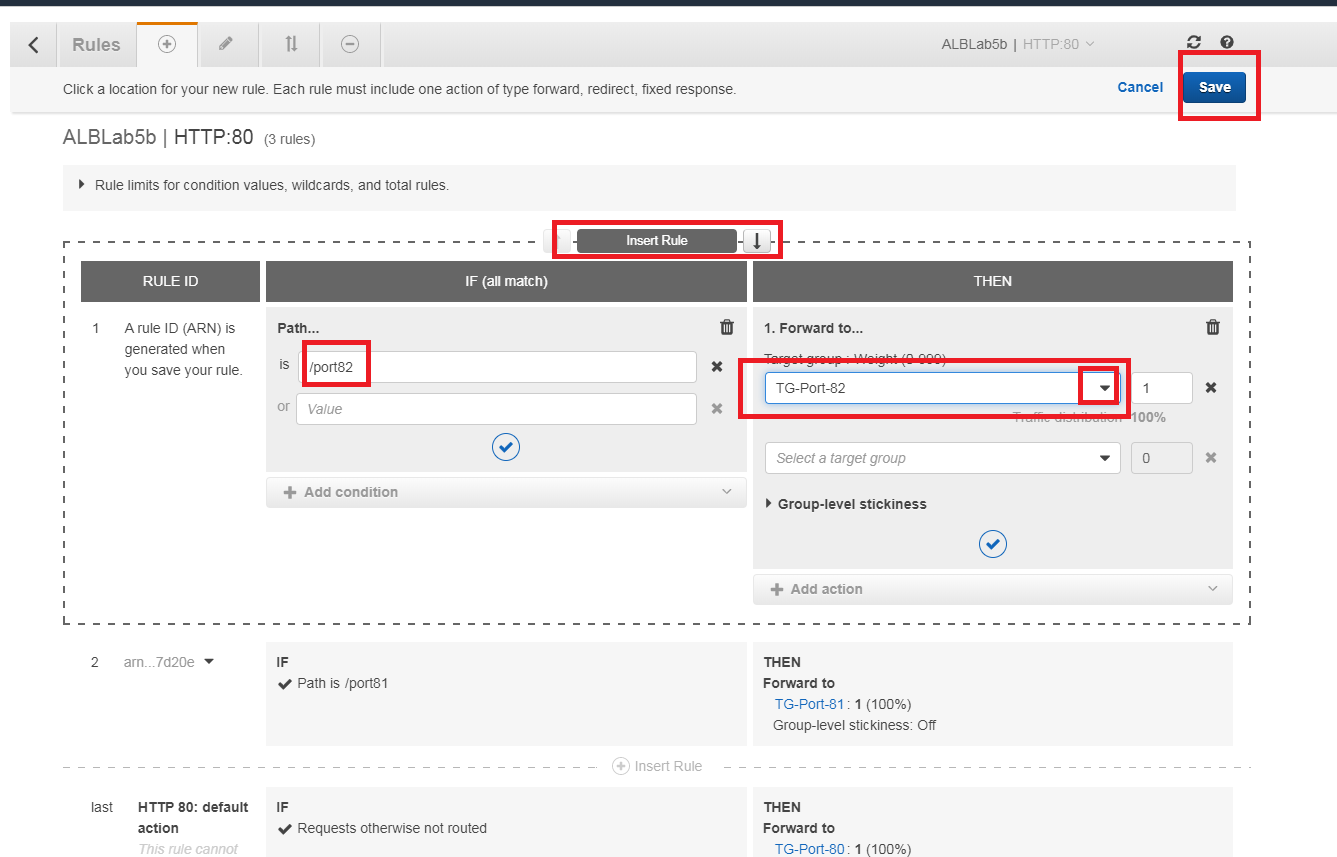


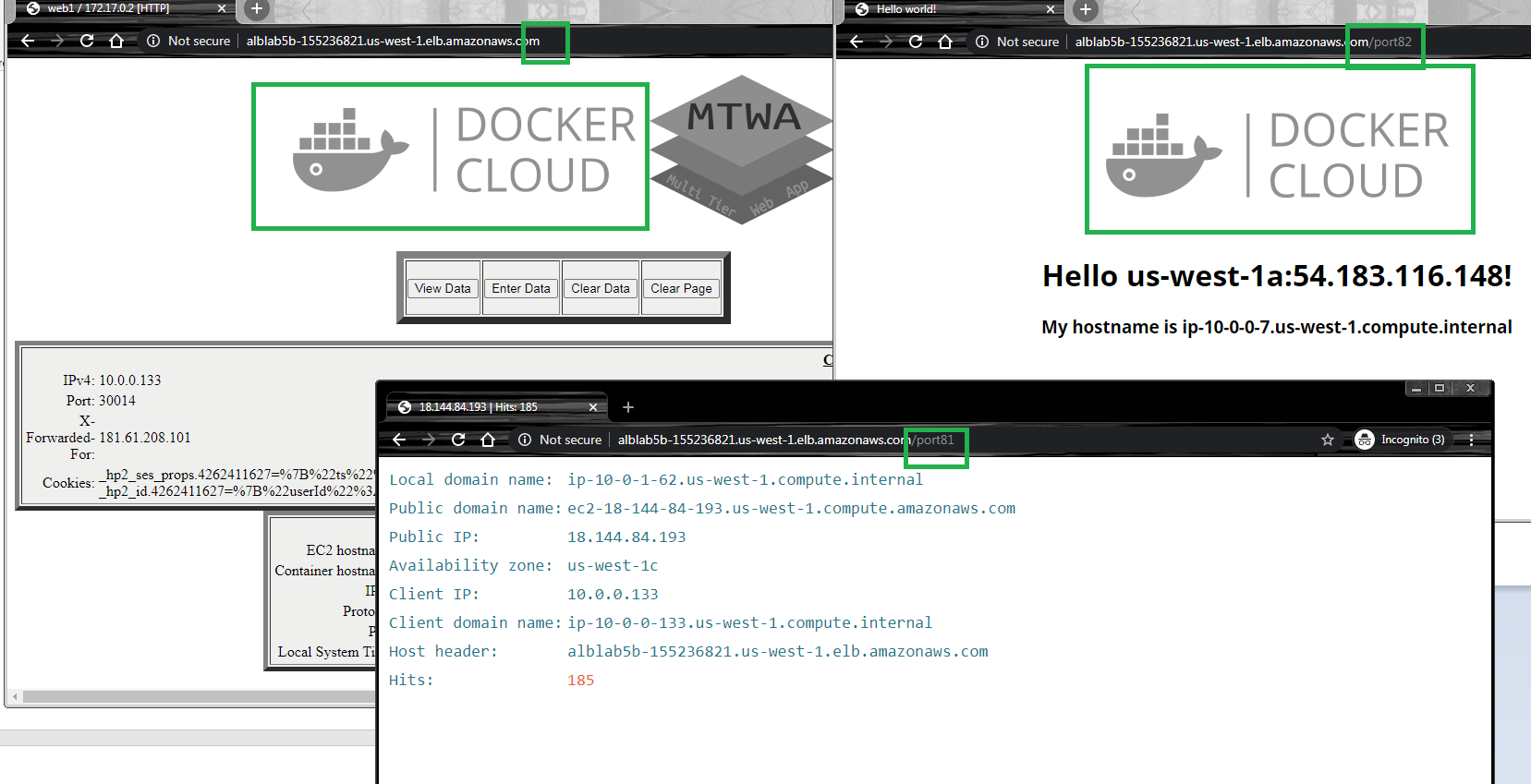
### Create Listener Rules for Routing Paths











## Lab 5B using Command Line (Windows)

### Create Network Infrastructure, Instances and its security groups

rem Setear las variables de su grupo. Clase A: 10.x.x.x/8 Clase B: 172.16.x.x a 172.31.x.x

set vpcn\_Mask="10.0.0.0/16"

set pbsn1\_Mask="10.0.0.0/24"

set pbsn2\_Mask="10.0.1.0/24"

set first\_az="us-west-1a"

set second\_az="us-west-1c"

set instance\_type="t3.small"

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 1

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 Permitir que las instancias que se ejecutan en la subred se hagan publicas

aws ec2 modify-subnet-attribute --subnet-id %pbsn1\_Id% --map-public-ip-on-launch

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%

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 Lab5b --query "KeyMaterial" --output text > Lab5b.pem

winscp.com /keygen "Lab5b.pem" /output="Lab5b.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 81 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress --group-id %SecGroup\_A\_Id% --protocol tcp --port 82 --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

rem Crear subred Publica 2, ponerla public

aws ec2 create-subnet --vpc-id %vpcn\_Id% --cidr-block %pbsn2\_Mask% --availability-zone %second\_az%|jq ".Subnet.SubnetId" >tmpFile

set /p pbsn2\_Id= < tmpFile

aws ec2 modify-subnet-attribute --subnet-id %pbsn2\_Id% --map-public-ip-on-launch

aws ec2 associate-route-table  --subnet-id %pbsn2\_Id% --route-table-id %Public\_RT\_Id%

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 Se solicitan instancias y se adiciona un bootstrap para comprobar que el docker fue instalado

aws ec2 run-instances --image-id %AMI% --count 1 --instance-type %instance\_type% --key-name Lab5b --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|.]"|jq ".[0]" >tmpFile

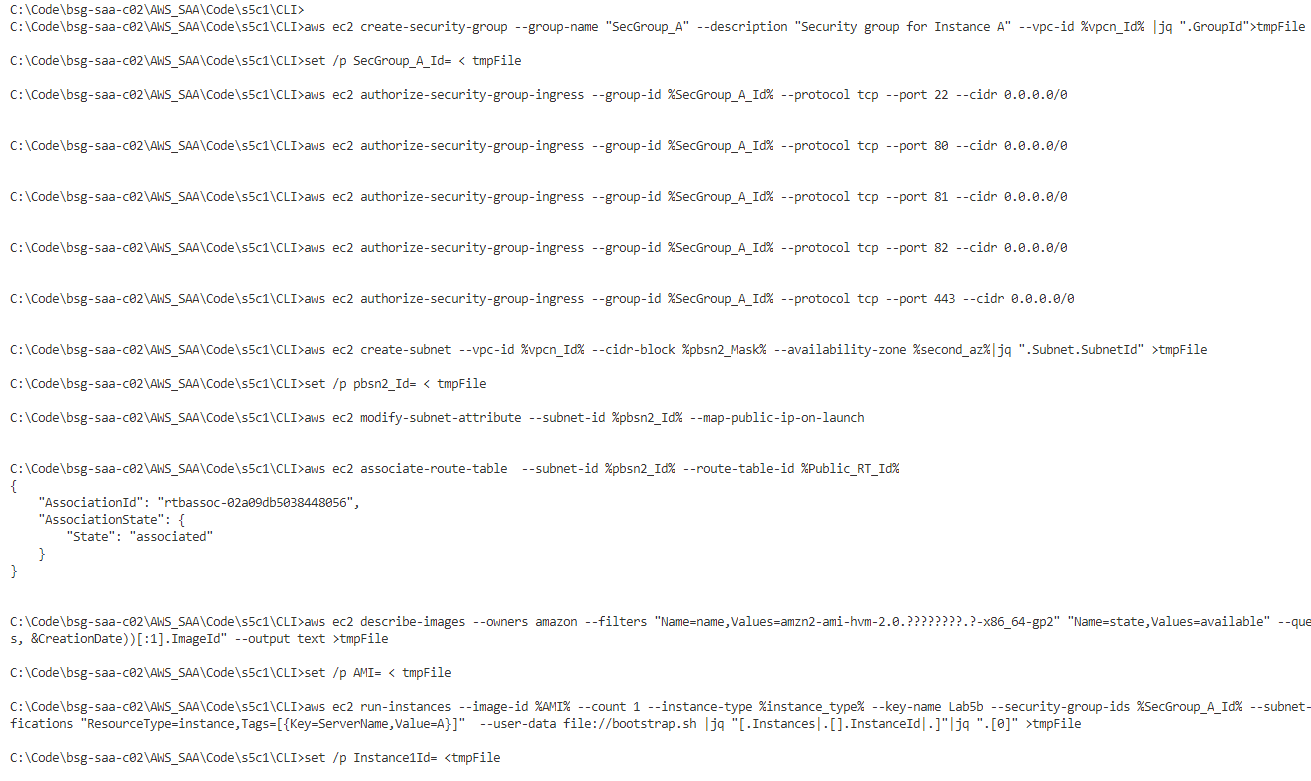
set /p Instance1Id= <tmpFile

aws ec2 run-instances --image-id %AMI% --count 1 --instance-type %instance\_type% --key-name Lab5b --security-group-ids %SecGroup\_A\_Id% --subnet-id %pbsn2\_Id% --tag-specifications "ResourceType=instance,Tags=[{Key=ServerName,Value=B}]"  --user-data file://bootstrap.sh |jq "[.Instances|.[].InstanceId|.]"|jq ".[0]" >tmpFile

set /p Instance2Id= <tmpFile

rem Traer Datos especificos de instancia A y B; y setearlos a las variables A\_IP y B\_IP





### Running the code on SSH

Use the first putty and run docker; finally check on browser that containers are running

rem Ingresar a ambas instancias publica por SSH. Ejecutar las mismas acciones y despues ir al navegador a ver que funcionan las IPs

putty.exe -i "Lab5b.ppk" ec2-user@%A\_IP%

rem Comprobar la instalacion de Docker y borramos cualquier contenedor anterior

docker ps -a

docker stop $(docker ps -aq)

docker rm $(docker ps -aq)

rem Comprobar las instancias de docker. Se explica el mapeo de puerto, Zonar horarias y el ejemplo anterior

sudo docker run -d -p 80:80 -p 443:443 -e TZ=America/Bogota -h web1 fmorenod81/mtwa:web

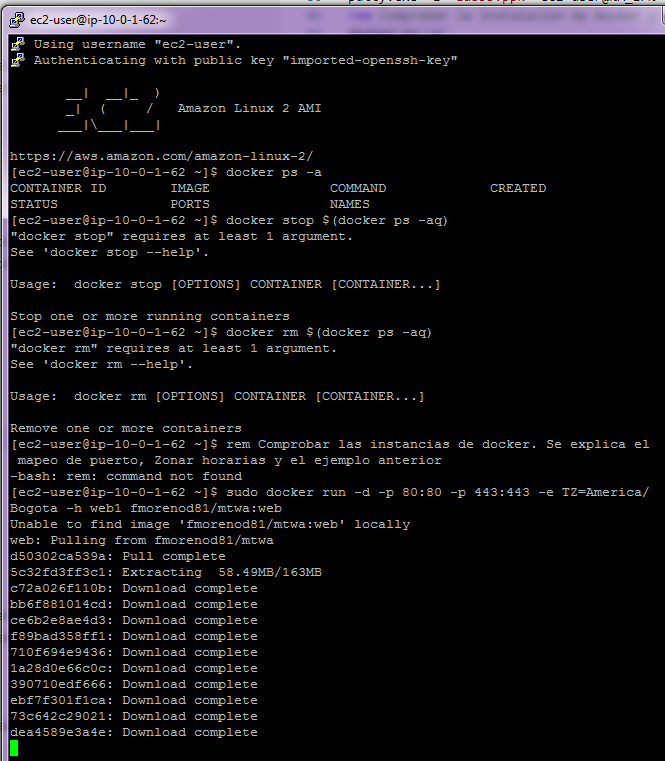
sudo docker run -d -p 81:80 -h web2 benpiper/r53-ec2-web

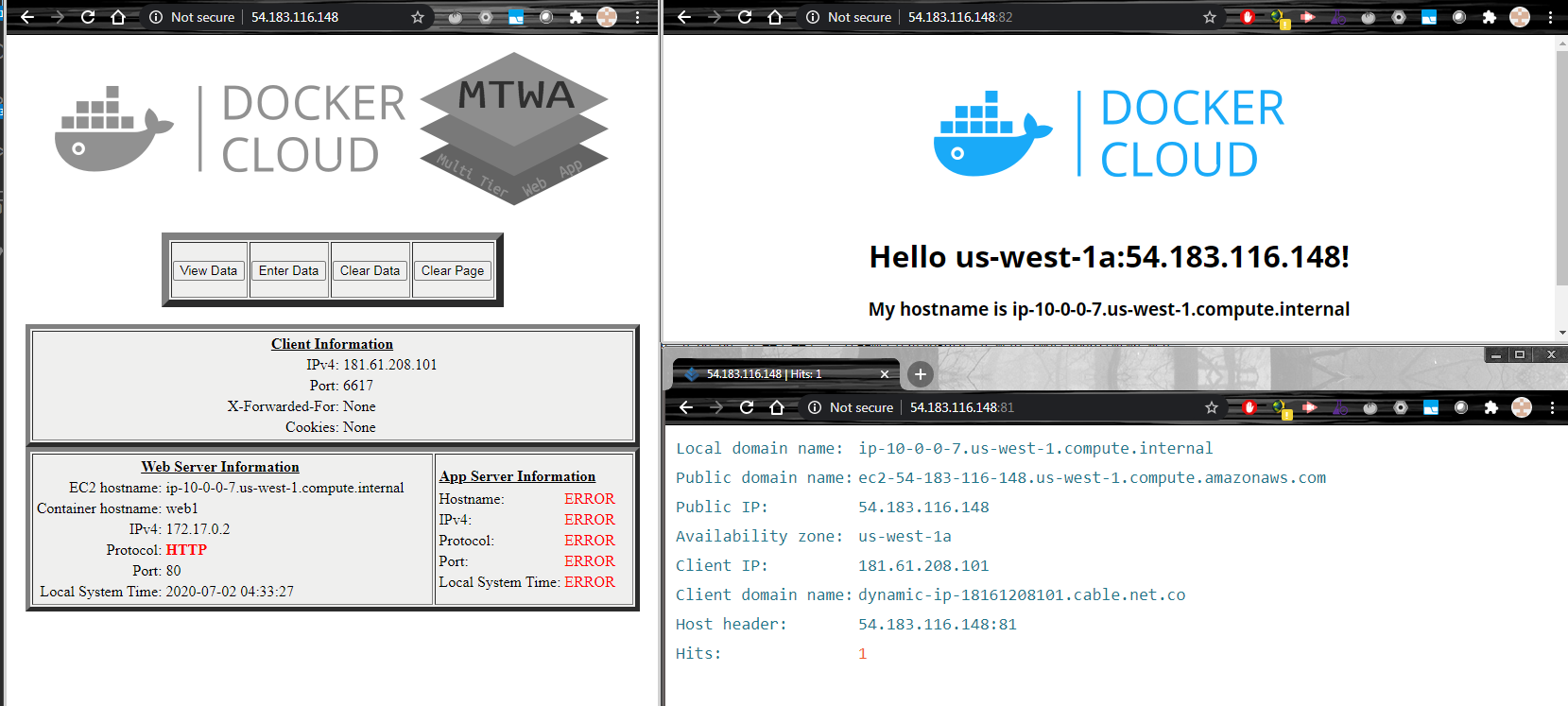
export AZ=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)

export PublicIP=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)

sudo docker run -d -p 82:80 -h $HOSTNAME --env NAME=$AZ:$PublicIP --env PORT=82  --env PROTO=TCP --env VALUE=$AZ dockercloud/hello-world

rem Se va al navegador y se visualizan con las IPs publicas los puertos 80, 443 y 82





### Create target groups, ALB and listeners

rem Crear los target groups y registrar las instancias a los mismos en cada puerto

aws elbv2 create-target-group --name TG-Port-80 --protocol HTTP --port 80 --target-type instance --vpc-id %vpcn\_Id% |jq ".TargetGroups[].TargetGroupArn" >tmpFile

set /p TG80\_ARN= < tmpFile

aws elbv2 register-targets --target-group-arn %TG80\_ARN% --targets Id=%Instance1Id% Id=%Instance2Id%

aws elbv2 create-target-group --name TG-Port-81 --protocol HTTP --port 81 --target-type instance --vpc-id %vpcn\_Id%|jq ".TargetGroups[].TargetGroupArn" >tmpFile

set /p TG81\_ARN= < tmpFile

aws elbv2 register-targets --target-group-arn %TG81\_ARN% --targets Id=%Instance1Id% Id=%Instance2Id%

aws elbv2 create-target-group --name TG-Port-82 --protocol HTTP --port 82 --target-type instance --vpc-id %vpcn\_Id%|jq ".TargetGroups[].TargetGroupArn" >tmpFile

set /p TG82\_ARN= < tmpFile

aws elbv2 register-targets --target-group-arn %TG82\_ARN% --targets Id=%Instance1Id% Id=%Instance2Id%

rem Crear el Balanceador

aws elbv2 create-load-balancer --name ALBLab5b --subnets %pbsn1\_Id% %pbsn2\_Id% --security-groups %SecGroup\_ALB\_Id% >tmpFile2

cat tmpFile2|jq ".LoadBalancers[].LoadBalancerArn" >tmpFile

set /p LB\_ARN= < tmpFile

cat tmpFile2|jq ".LoadBalancers[].DNSName" >tmpFile

set /p LB\_DNSName= < tmpFile

del tmpFile2

rem Se crea el Listener para Puerto 80

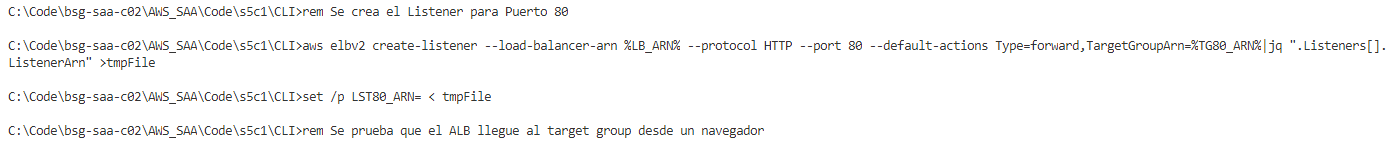
aws elbv2 create-listener --load-balancer-arn %LB\_ARN% --protocol HTTP --port 80 --default-actions Type=forward,TargetGroupArn=%TG80\_ARN%|jq ".Listeners[].ListenerArn" >tmpFile

set /p LST80\_ARN= < tmpFile

rem Se prueba que el ALB llegue al target group desde un navegador

echo Para navegar a %LB\_DNSName%





rem Se crea el Listener para Puerto 81, 82

aws elbv2 create-listener --load-balancer-arn %LB\_ARN% --protocol HTTP --port 81 --default-actions Type=forward,TargetGroupArn=%TG81\_ARN%|jq ".Listeners[].ListenerArn" >tmpFile

set /p LST81\_ARN= < tmpFile

aws elbv2 create-listener --load-balancer-arn %LB\_ARN% --protocol HTTP --port 82 --default-actions Type=forward,TargetGroupArn=%TG82\_ARN%|jq ".Listeners[].ListenerArn" >tmpFile

set /p LST82\_ARN= < tmpFile

rem Probar porque no funciona en los puertos 81, 82 y 443

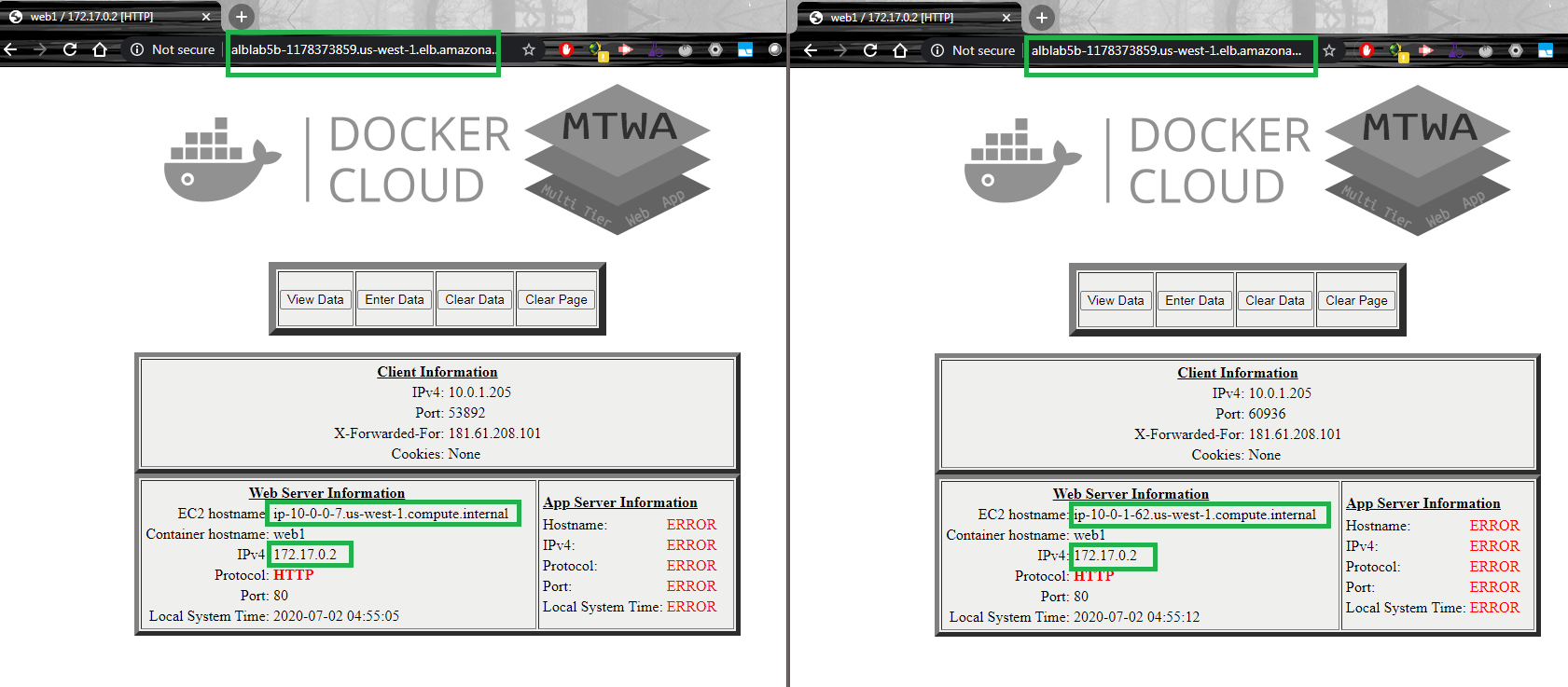
rem Habilitar los sec group al ALB

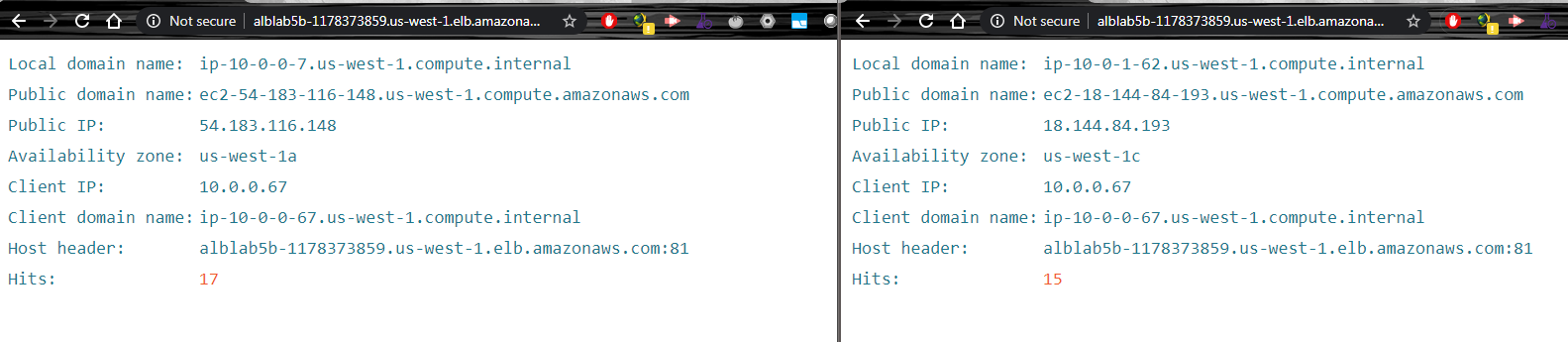
aws ec2 authorize-security-group-ingress --group-id %SecGroup\_ALB\_Id% --protocol tcp --port 81 --cidr 0.0.0.0/0

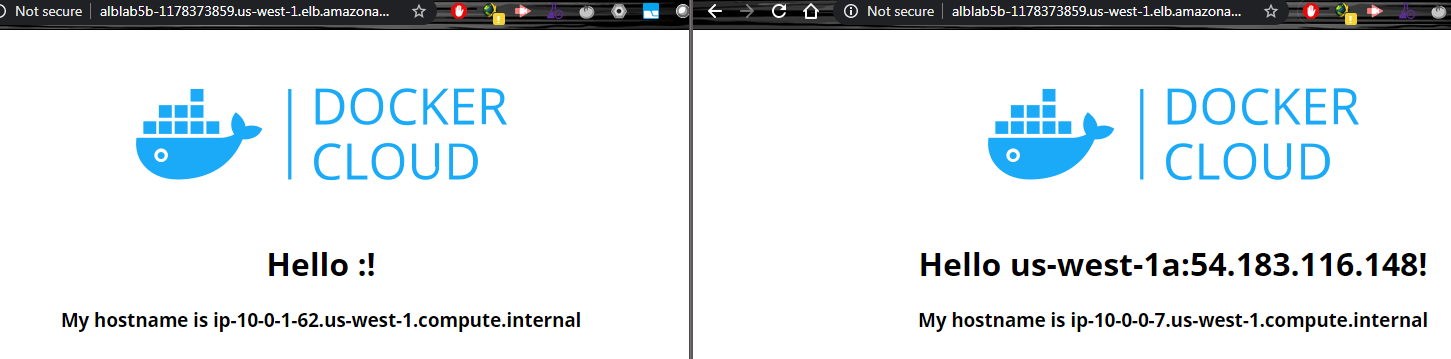
aws ec2 authorize-security-group-ingress --group-id %SecGroup\_ALB\_Id% --protocol tcp --port 82 --cidr 0.0.0.0/0

rem Probar porque funciona en los puertos 81, 82

From the LB\_DNSName variable review with browser, the balance of 2 instances on port 80, 82 without routing







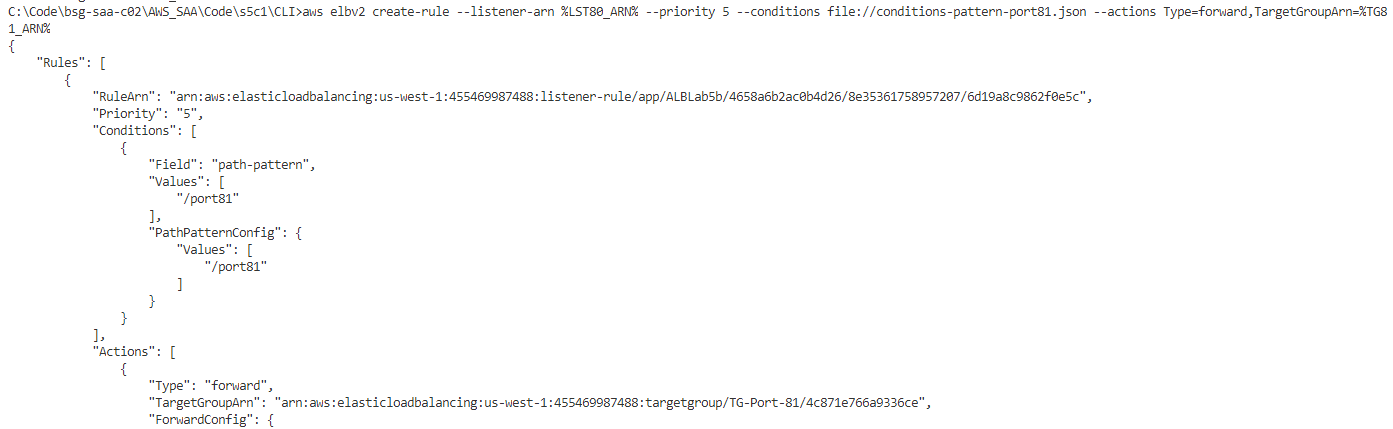
### Able Routing Paths

rem Crear la regla para el puerto 80 y que cumpla el path del archivo JSON

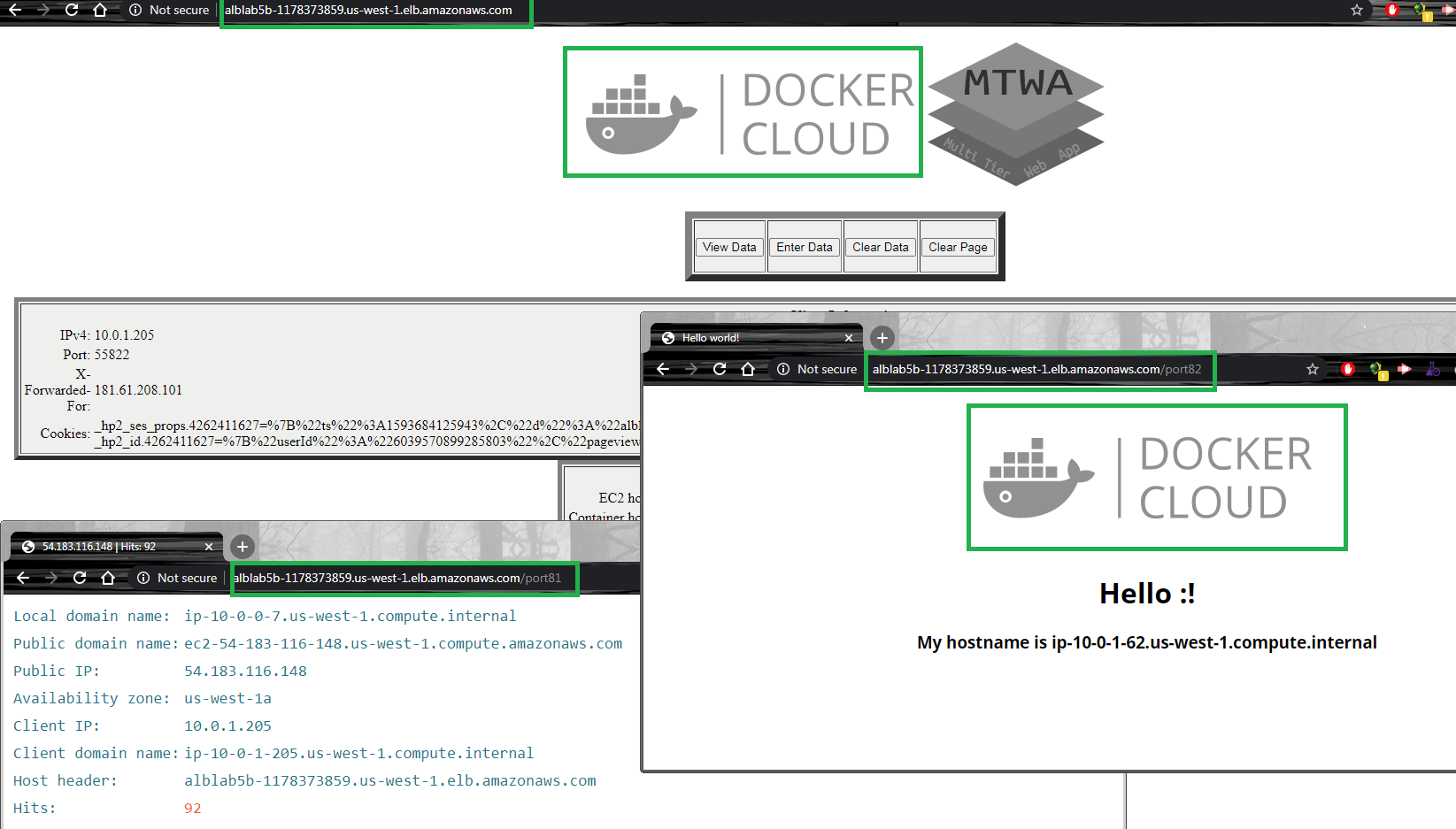
aws elbv2 create-rule --listener-arn %LST80\_ARN% --priority 5 --conditions file://conditions-pattern-port81.json --actions Type=forward,TargetGroupArn=%TG81\_ARN%

aws elbv2 create-rule --listener-arn %LST80\_ARN% --priority 4 --conditions file://conditions-pattern-port82.json --actions Type=forward,TargetGroupArn=%TG82\_ARN%

rem Revisar las rutas del balanceador







## Clean Resources

### For Web Management Console

Delete Instances

Delete Keypair

Delete Security Groups

Delete ALB

Delete Target Groups

Delete VPC

# Evidences to send

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

1. The last picture of [Create Additional Listeners for Ports](#_Create_Additional_Listeners), which show 3 navigation tabs (browsers) with 3 different ports for the same ALB.
2. The last picture of [Create Listener Rules for Routing Paths](#_Create_Listener_Rules), which show 3 navigation tabs (browsers) with 3 different paths for the same ALB.