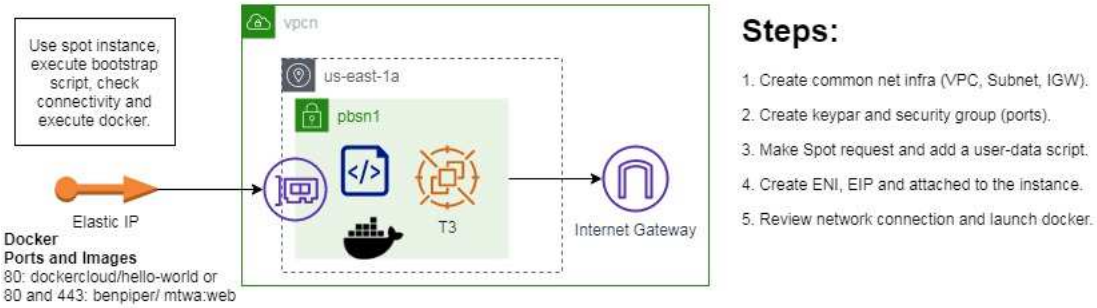
Contents

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 Netowork 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/)

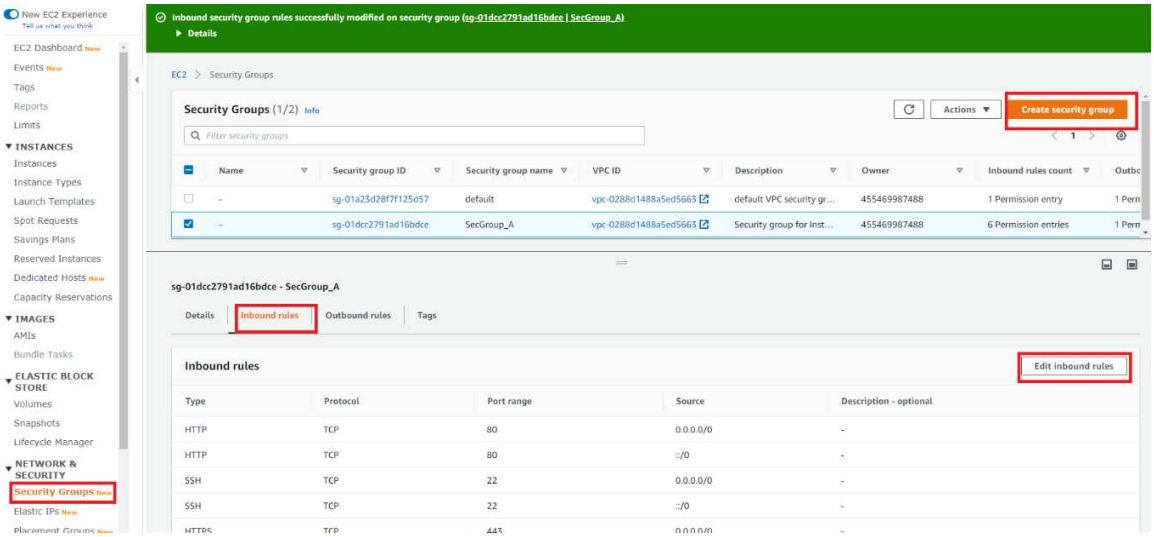
Lab 5A: Spot EC2 Instance with additional ENI, EIP, and bootstrap scripts

Lab 5A using Web Management Console

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

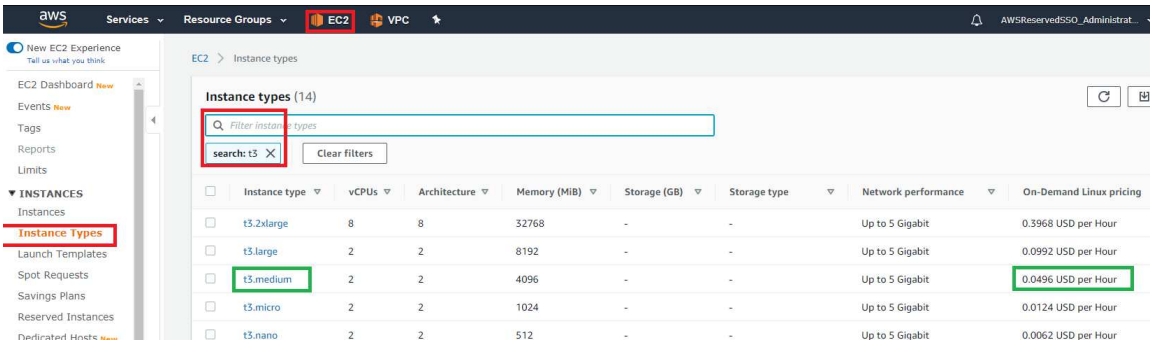
Create keys, Sec Group (Labs4c1)

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.

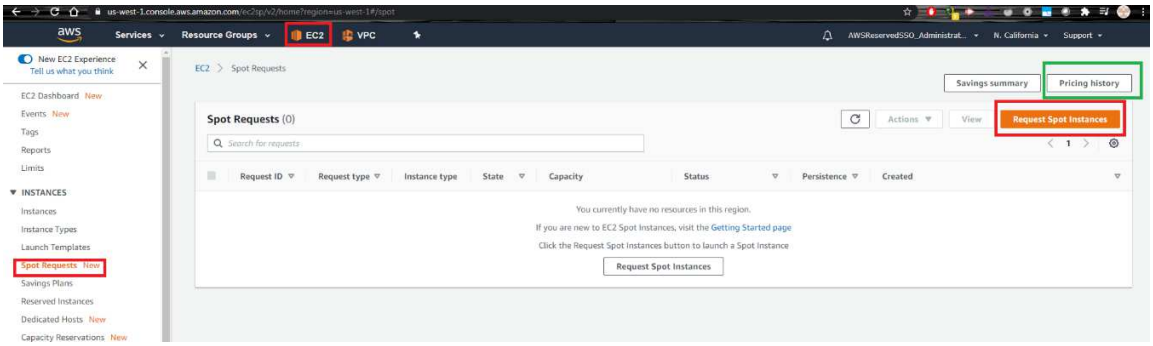


Check Price and make a Spot Request

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



EC2 > Spot Requests > Request Spot Instances

Request Spot Instances

Tell us your application or task need

To help us identify the most appropriate compute capacity for your job, select the closest match for your application or task need.

- ○ Load balancing workloads
  Launch instances of the same size, in any Availability Zone. Good for running web services.
- ○ Flexible workloads
  Launch instances of any size, in any Availability Zone. Good for running batch and CI/CD jobs.
- ○ Big data workloads
  Launch instances of any size, in a single Availability Zone. Good for MapReduce jobs.
- ◉ Defined duration workloads
  Launch instances into a Spot block for 1 to 6 hours.

One hour ▼

Configure your instances

Use a launch template to quickly set instance launch parameters (you can still modify some parameters). Improve your access to Spot capacity by including multiple Availability Zones.

AMI

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Deep Learning AMI (Microsoft Windows Server 2016) (ami-00ea864915b7ece37) ▼ Search for AMI

Select instance type

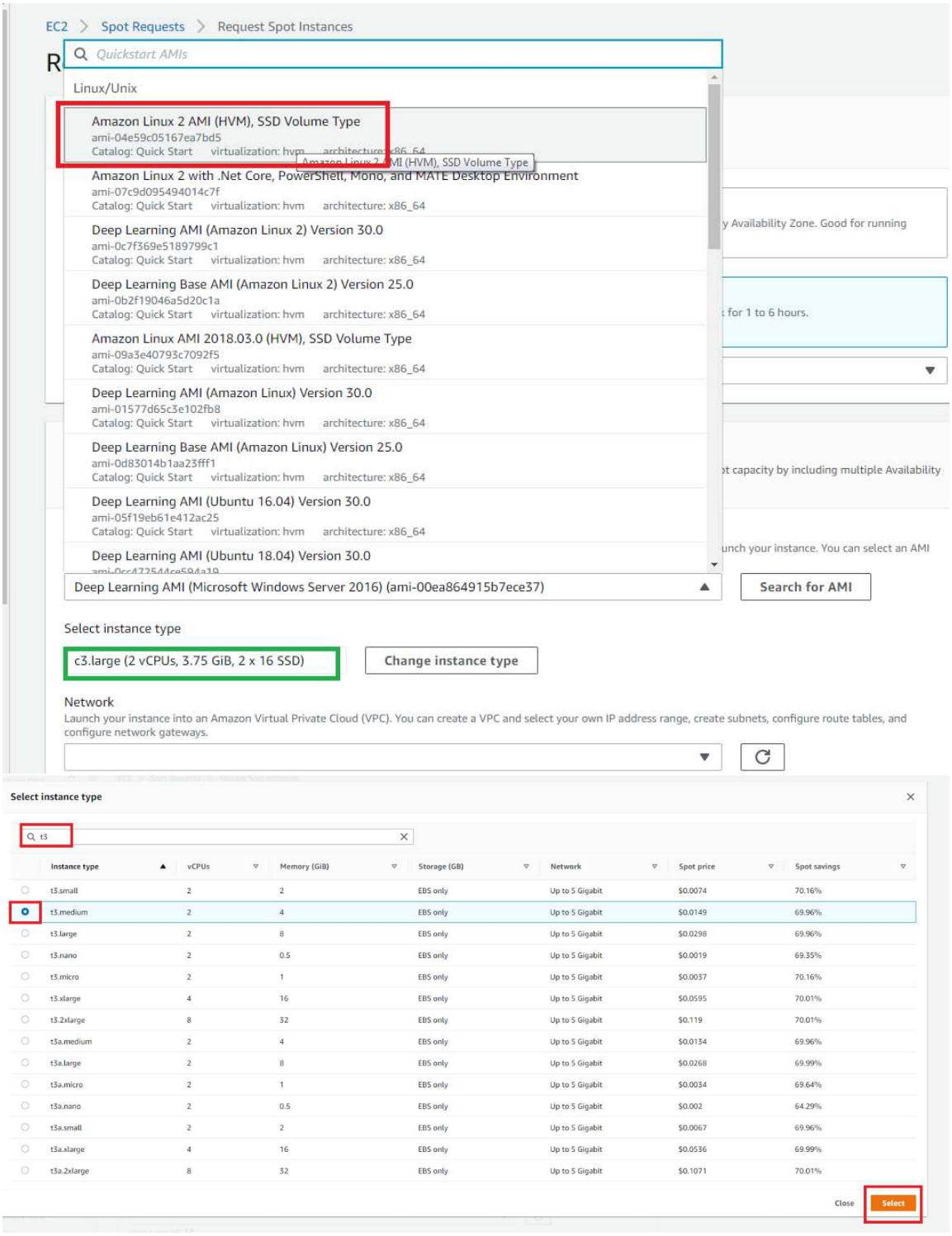
c3.large (2 vCPUs, 3.75 GiB, 2 x 16 SSD) Change instance type

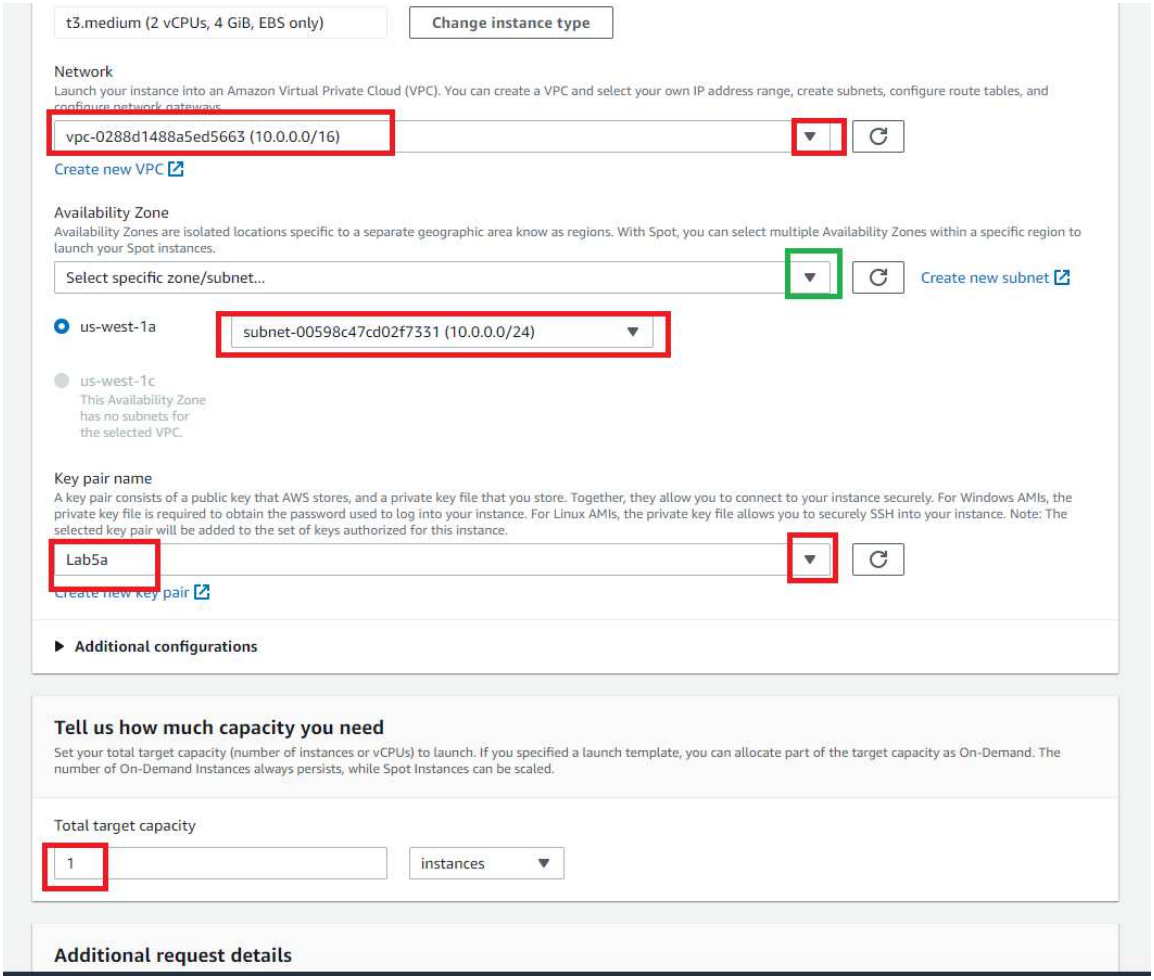
Network

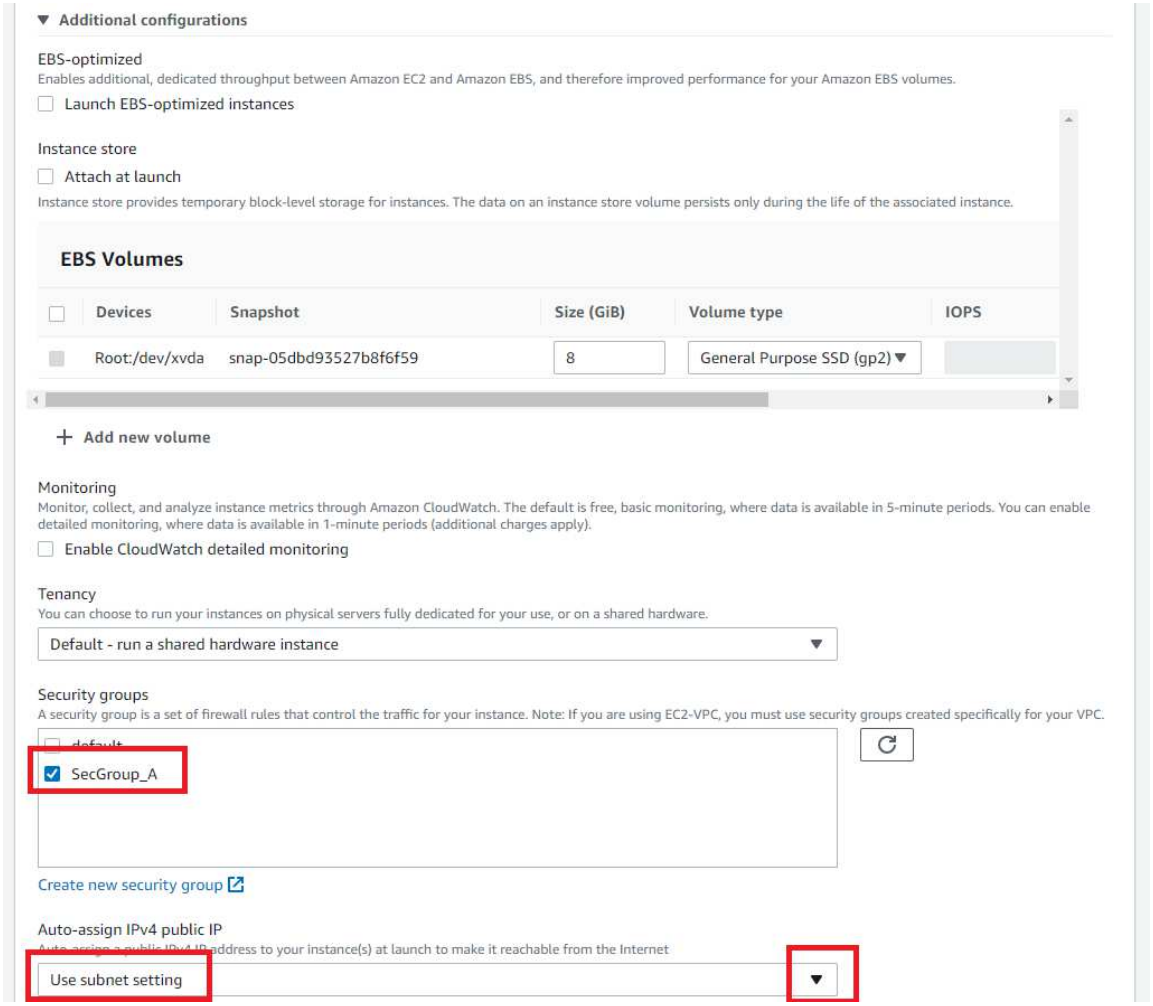
Launch your instance into an Amazon Virtual Private Cloud (VPC). You can create a VPC and select your own IP address range, create subnets, configure route tables, and configure network gateways.

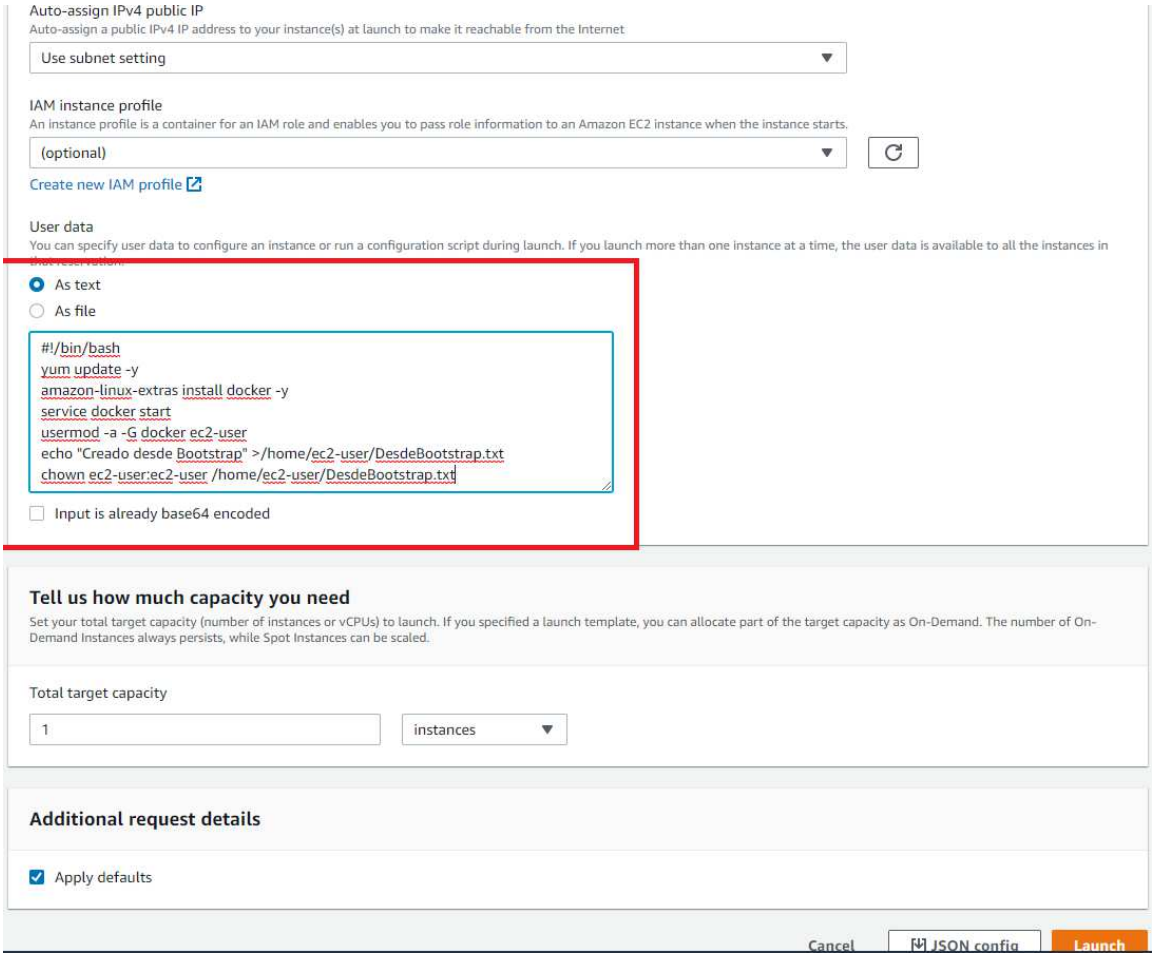
▼ ↻

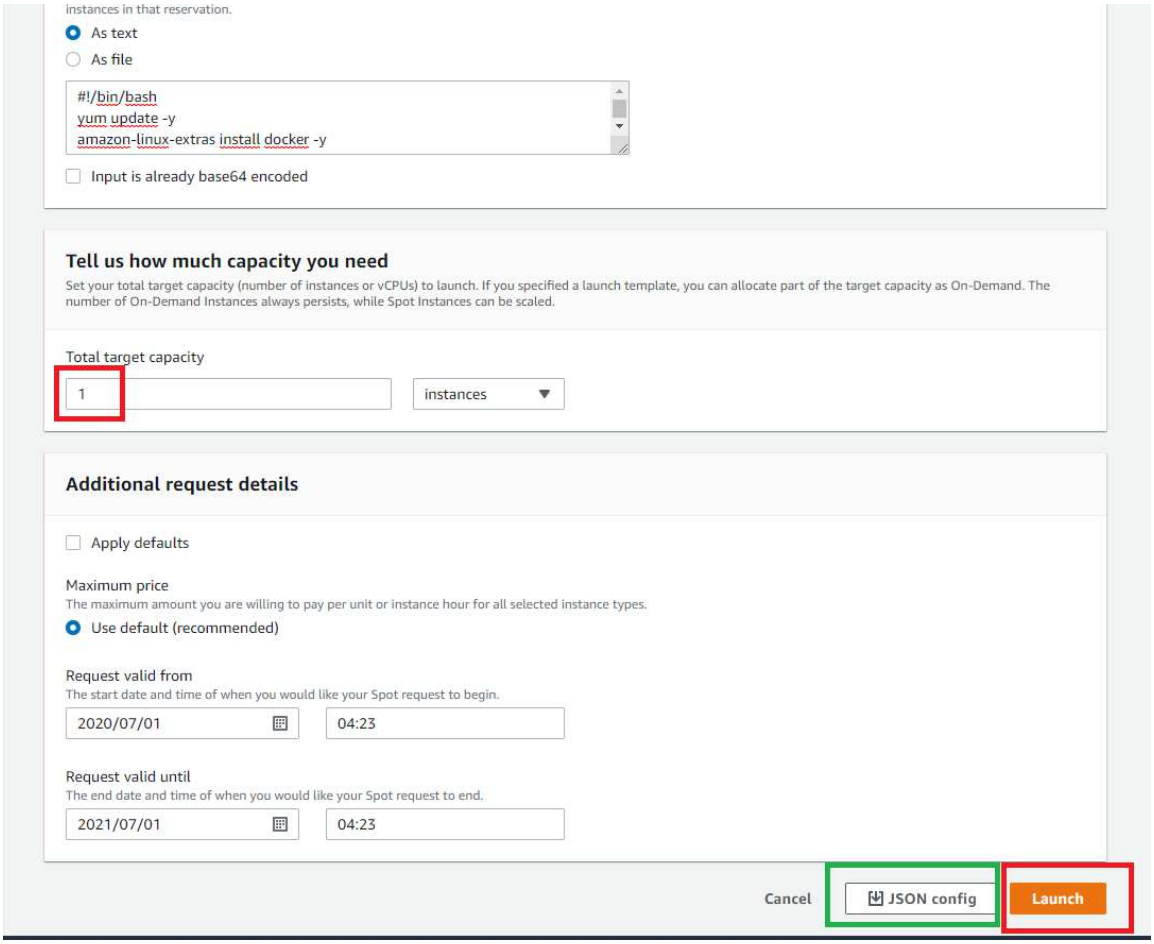
Create new VPC


EC2 > Spot Requests > Request Spot Instances
Quickstart AMIs
Linux/Unix
Amazon Linux 2 AMI (HVM), SSD Volume Type
ami-04e59c05167ea7bd5
Catalog: Quick Start virtualization: hvm architecture: x86_64
Amazon Linux 2 with .Net Core, PowerShell, Mono, and MATE Desktop Environment
ami-07c9d095494014c7f
Catalog: Quick Start virtualization: hvm architecture: x86_64
Deep Learning AMI (Amazon Linux 2) Version 30.0
ami-0c7f369e5189799c1
Catalog: Quick Start virtualization: hvm architecture: x86_64
Deep Learning Base AMI (Amazon Linux 2) Version 25.0
ami-0b2f19046a5d20c1a
Catalog: Quick Start virtualization: hvm architecture: x86_64
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type
ami-09a3e40793c7092f5
Catalog: Quick Start virtualization: hvm architecture: x86_64
Deep Learning AMI (Amazon Linux) Version 30.0
ami-01577d65c3e102fb8
Catalog: Quick Start virtualization: hvm architecture: x86_64
Deep Learning Base AMI (Amazon Linux) Version 25.0
ami-0d83014b1aa23fff1
Catalog: Quick Start virtualization: hvm architecture: x86_64
Deep Learning AMI (Ubuntu 16.04) Version 30.0
ami-05f19eb61e412ac25
Catalog: Quick Start virtualization: hvm architecture: x86_64
Deep Learning AMI (Ubuntu 18.04) Version 30.0
Deep Learning AMI (Microsoft Windows Server 2016) (ami-00ea864915b7ece37)
Search for AMI
Select instance type
c3.large (2 vCPUs, 3.75 GiB, 2 x 16 SSD)
Change instance type
Network
Launch your instance into an Amazon Virtual Private Cloud (VPC). You can create a VPC and select your own IP address range, create subnets, configure route tables, and configure network gateways.
Select instance type
t3
Instance type | vCPUs | Memory (GiB) | Storage (GB) | Network | Spot price | Spot savings
t3.small 2 2 EBS only Up to 5 Gigabit $0.0074 70.16%
t3.medium 2 4 EBS only Up to 5 Gigabit $0.0149 69.96%
t3.large 2 8 EBS only Up to 5 Gigabit $0.0298 69.96%
t3.nano 2 0.5 EBS only Up to 5 Gigabit $0.0019 69.35%
t3.micro 2 1 EBS only Up to 5 Gigabit $0.0037 70.16%
t3.xlarge 4 16 EBS only Up to 5 Gigabit $0.0595 70.01%
t3.2xlarge 8 32 EBS only Up to 5 Gigabit $0.119 70.01%
t3a.medium 2 4 EBS only Up to 5 Gigabit $0.0134 69.96%
t3a.large 2 8 EBS only Up to 5 Gigabit $0.0268 69.99%
t3a.micro 2 1 EBS only Up to 5 Gigabit $0.0034 69.64%
t3a.nano 2 0.5 EBS only Up to 5 Gigabit $0.002 64.29%
t3a.small 2 2 EBS only Up to 5 Gigabit $0.0067 69.96%
t3a.xlarge 4 16 EBS only Up to 5 Gigabit $0.0536 69.99%
t3a.2xlarge 8 32 EBS only Up to 5 Gigabit $0.1071 70.01%
Close
Select

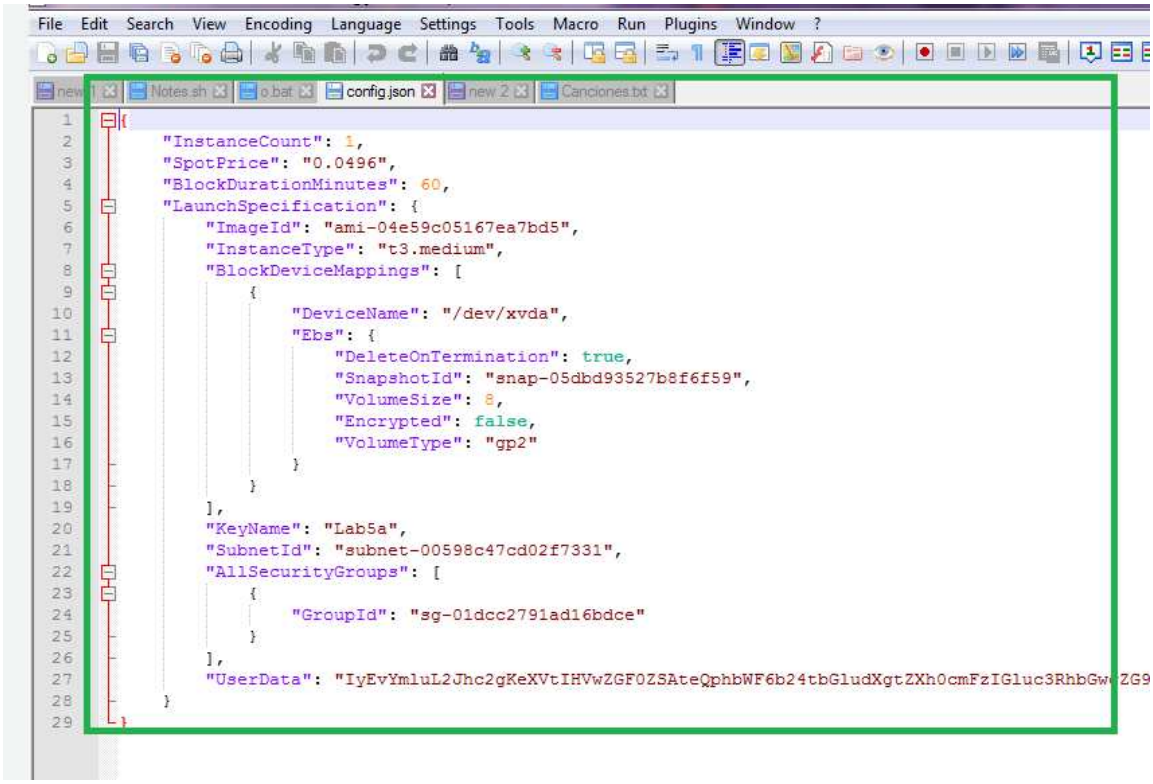

t3.medium (2 vCPUs, 4 GiB, EBS only)
Change instance type
Network
Launch your instance into an Amazon Virtual Private Cloud (VPC). You can create a VPC and select your own IP address range, create subnets, configure route tables, and configure network gateways.
vpc-0288d1488a5ed5663 (10.0.0.0/16)
Create new VPC
Availability Zone
Availability Zones are isolated locations specific to a separate geographic area know as regions. With Spot, you can select multiple Availability Zones within a specific region to launch your Spot instances.
Select specific zone/subnet...
Create new subnet
us-west-1a
subnet-00598c47cd02f7331 (10.0.0.0/24)
us-west-1c
This Availability Zone has no subnets for the selected VPC.
Key pair name
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Note: The selected key pair will be added to the set of keys authorized for this instance.
Lab5a
Create new key pair
Additional configurations
Tell us how much capacity you need
Set your total target capacity (number of instances or vCPUs) to launch. If you specified a launch template, you can allocate part of the target capacity as On-Demand. The number of On-Demand Instances always persists, while Spot Instances can be scaled.
Total target capacity
1
instances
Additional request details


Additional configurations
EBS-optimized
Enables additional, dedicated throughput between Amazon EC2 and Amazon EBS, and therefore improved performance for your Amazon EBS volumes.
Launch EBS-optimized instances
Instance store
Attach at launch
Instance store provides temporary block-level storage for instances. The data on an instance store volume persists only during the life of the associated instance.
EBS Volumes
Devices
Snapshot
Size (GiB)
Volume type
IOPS
Root:/dev/xvda
snap-05dbd93527b8f6f59
8
General Purpose SSD (gp2)
Add new volume
Monitoring
Monitor, collect, and analyze instance metrics through Amazon CloudWatch. The default is free, basic monitoring, where data is available in 5-minute periods. You can enable detailed monitoring, where data is available in 1-minute periods (additional charges apply).
Enable CloudWatch detailed monitoring
Tenancy
You can choose to run your instances on physical servers fully dedicated for your use, or on a shared hardware.
Default - run a shared hardware instance
Security groups
A security group is a set of firewall rules that control the traffic for your instance. Note: If you are using EC2-VPC, you must use security groups created specifically for your VPC.
SecGroup_A
Create new security group
Auto-assign IPv4 public IP
Use subnet setting

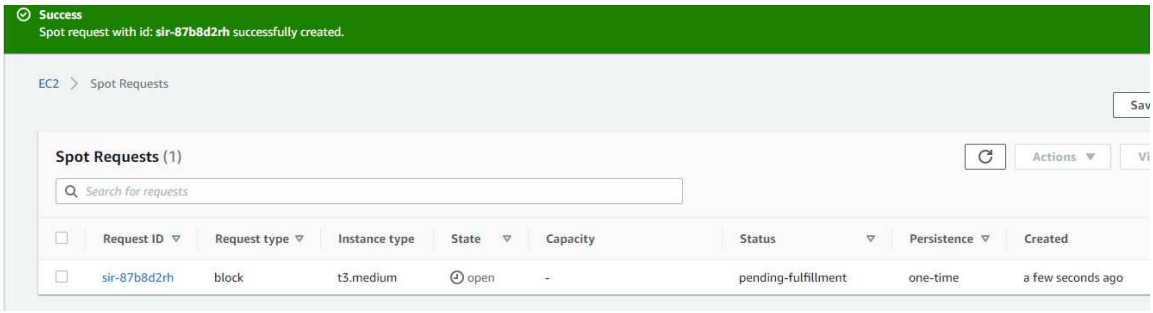

Auto-assign IPv4 public IP
Auto-assign a public IPv4 IP address to your instance(s) at launch to make it reachable from the Internet
Use subnet setting
IAM instance profile
An instance profile is a container for an IAM role and enables you to pass role information to an Amazon EC2 instance when the instance starts.
(optional)
Create new IAM profile
User data
You can specify user data to configure an instance or run a configuration script during launch. If you launch more than one instance at a time, the user data is available to all the instances in
As text
As file
#!/bin/bash
yum update -y
amazon-linux-extras install docker -y
service docker start
usermod -a -G docker ec2-user
echo "Creado desde Bootstrap" >/home/ec2-user/DesdeBootstrap.txt
chown ec2-user:ec2-user /home/ec2-user/DesdeBootstrap.txt
Input is already base64 encoded
Tell us how much capacity you need
Set your total target capacity (number of instances or vCPUs) to launch. If you specified a launch template, you can allocate part of the target capacity as On-Demand. The number of On-Demand Instances always persists, while Spot Instances can be scaled.
Total target capacity
1
instances
Additional request details
Apply defaults
Cancel
JSON config
Launch



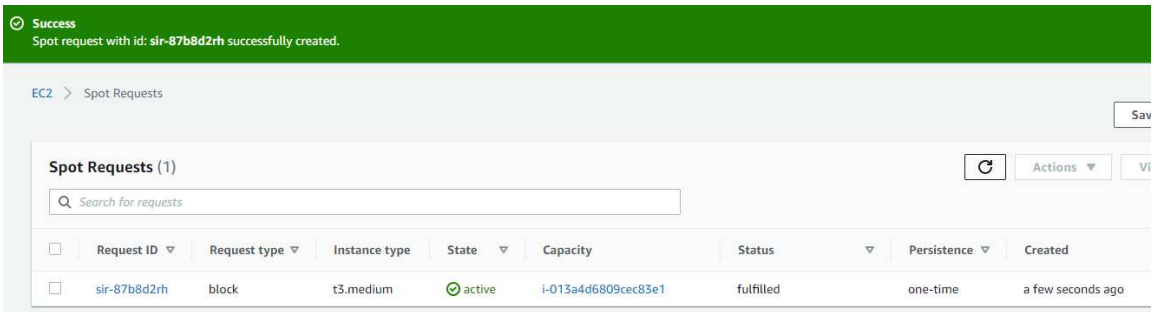
Check JSON File



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

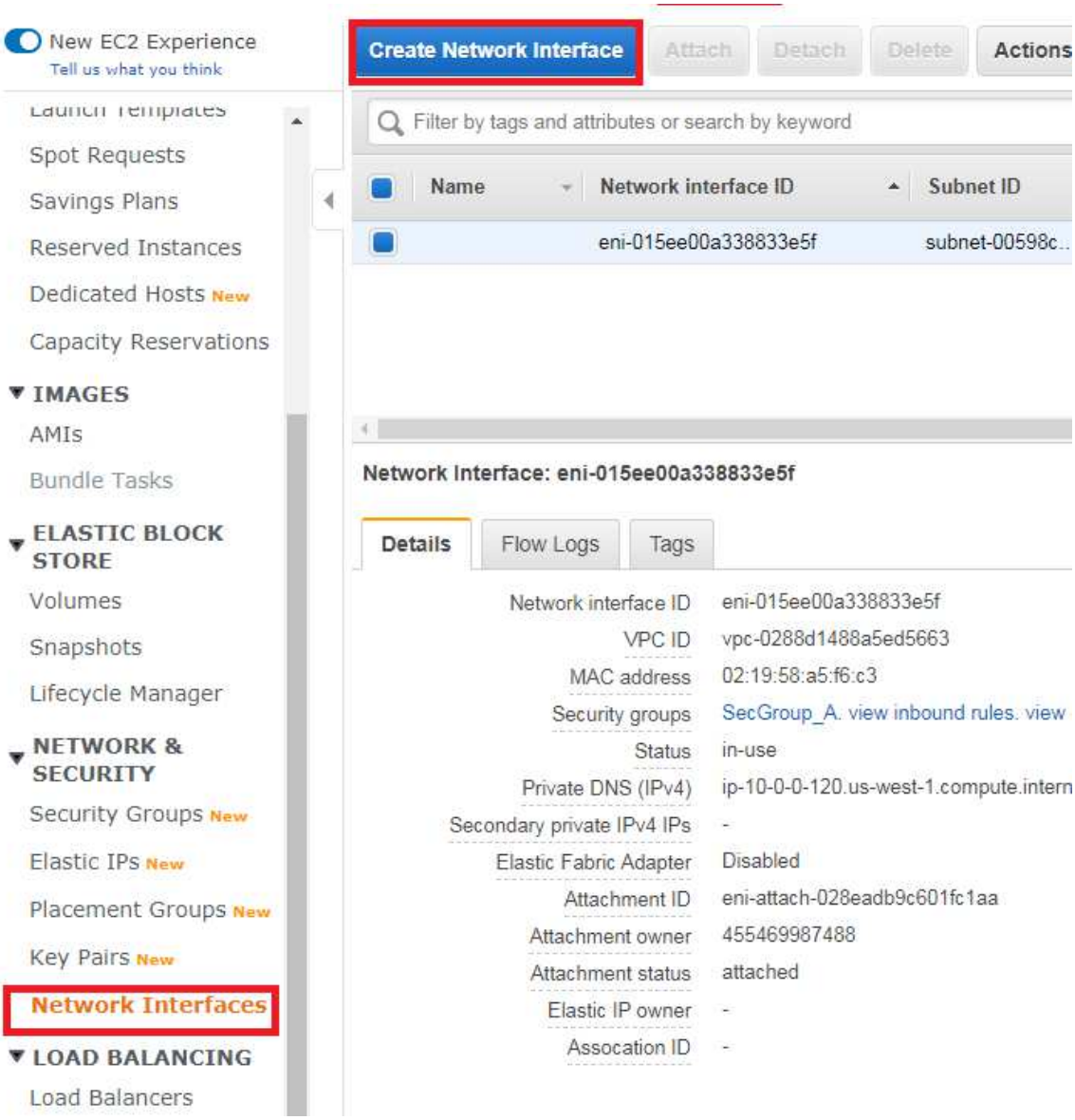


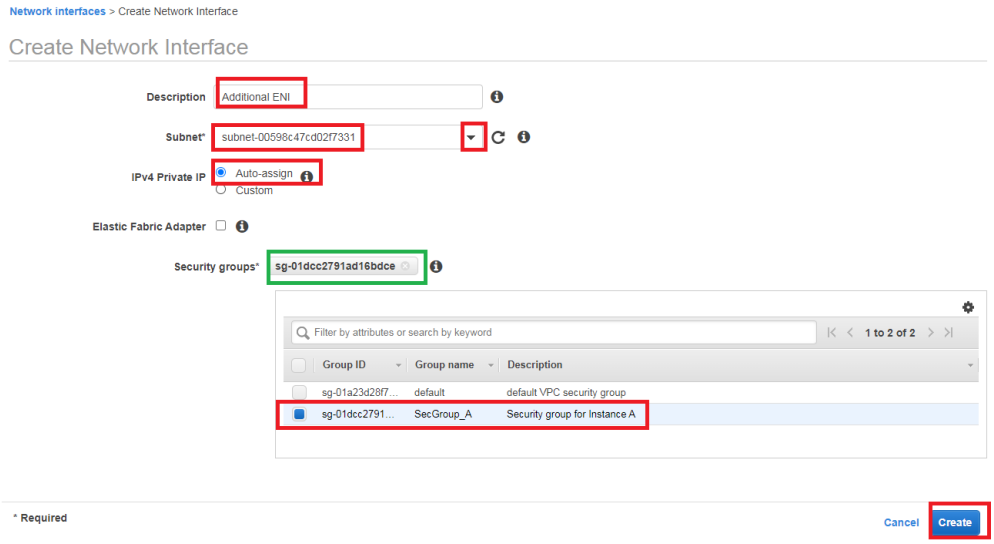
After some seconds, status changes and the instance is created.

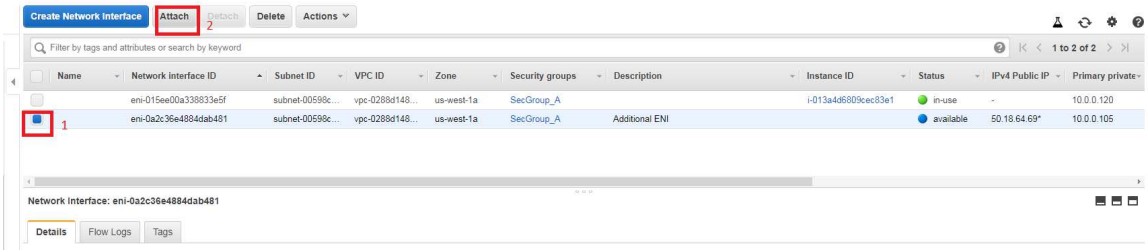


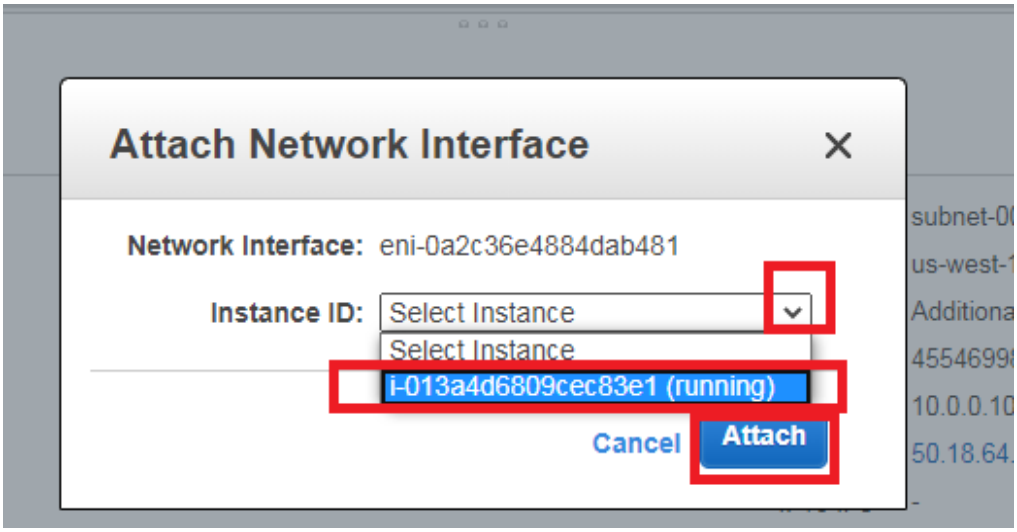
Create ENI, EIP and its associations

Create the ENI and attach to the running instances

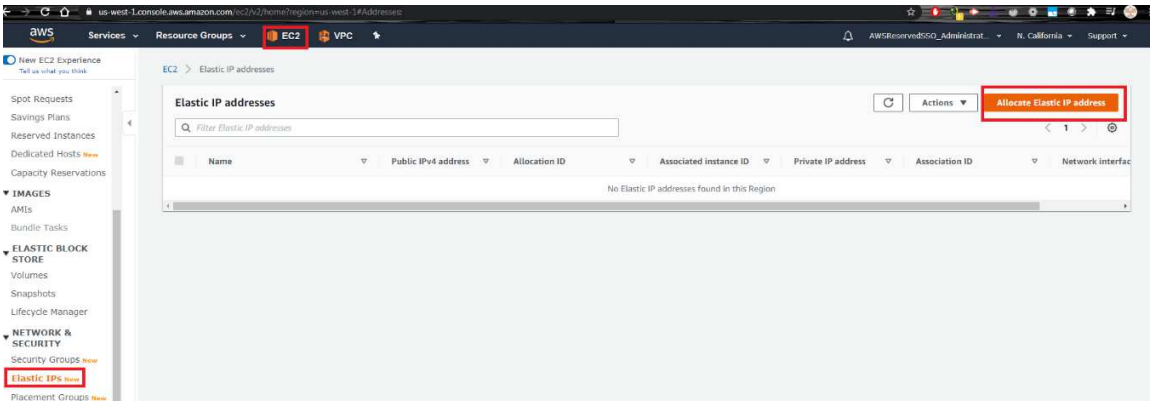


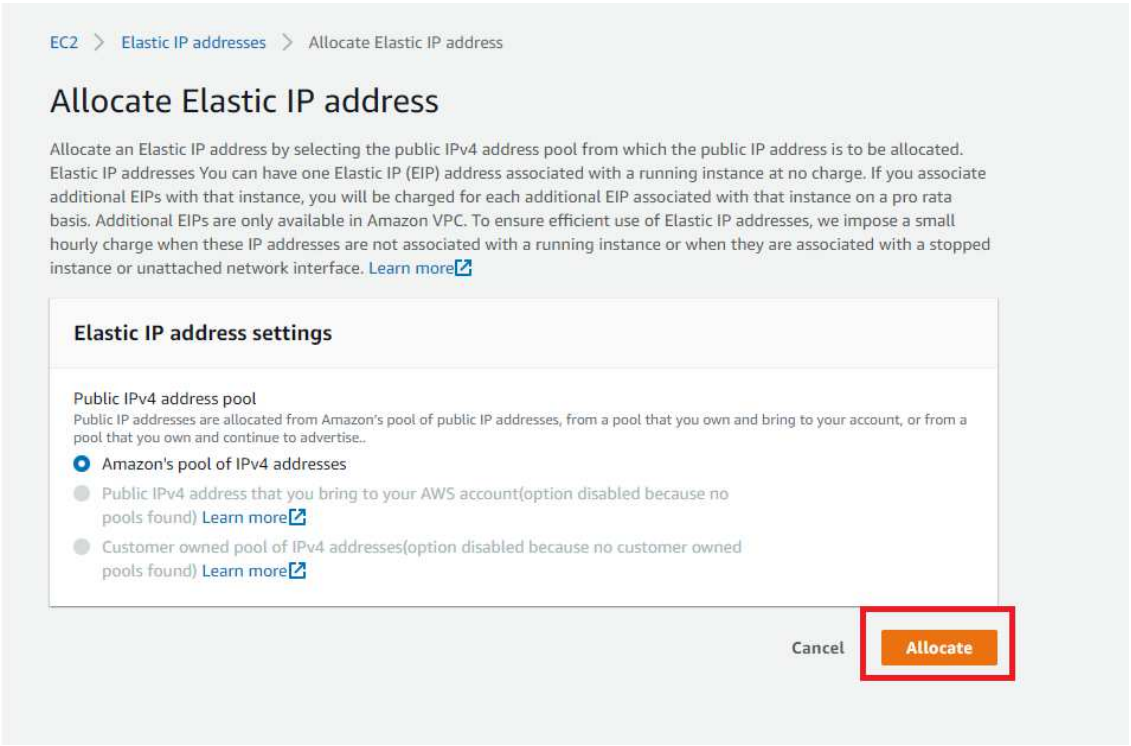


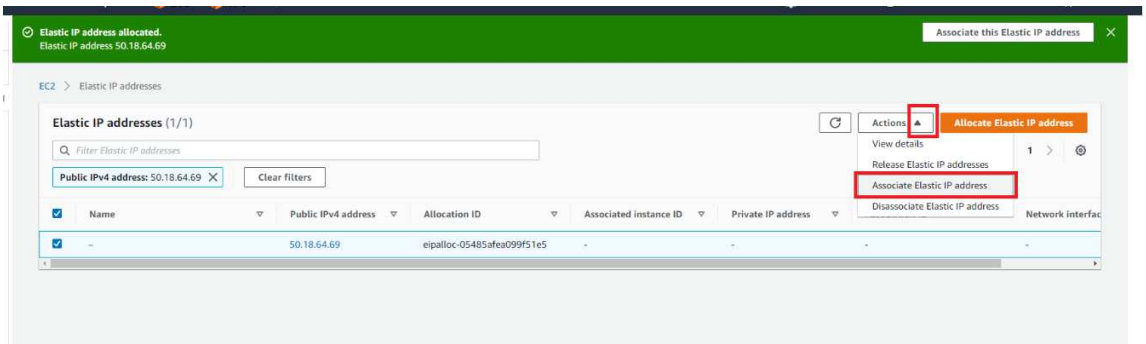


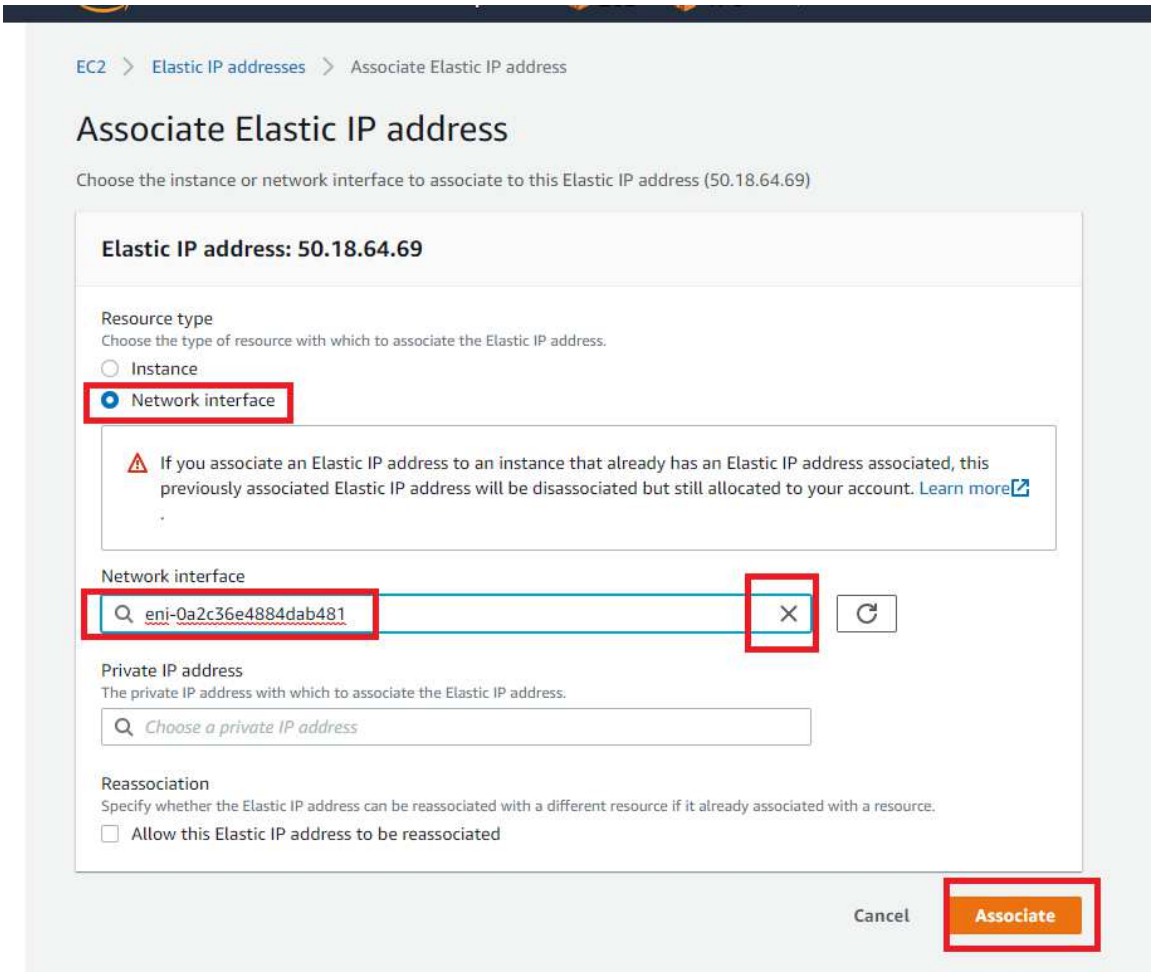


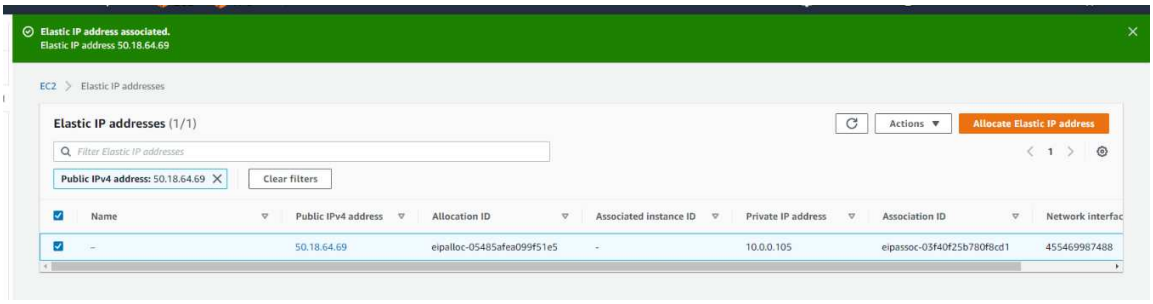
Create (Allocate) EIP and associated it to the ENI


Services
Resource Groups
EC2
VPC
Elastic IP addresses
Actions
Allocate Elastic IP address
No Elastic IP addresses found in this Region
Elastic IPs

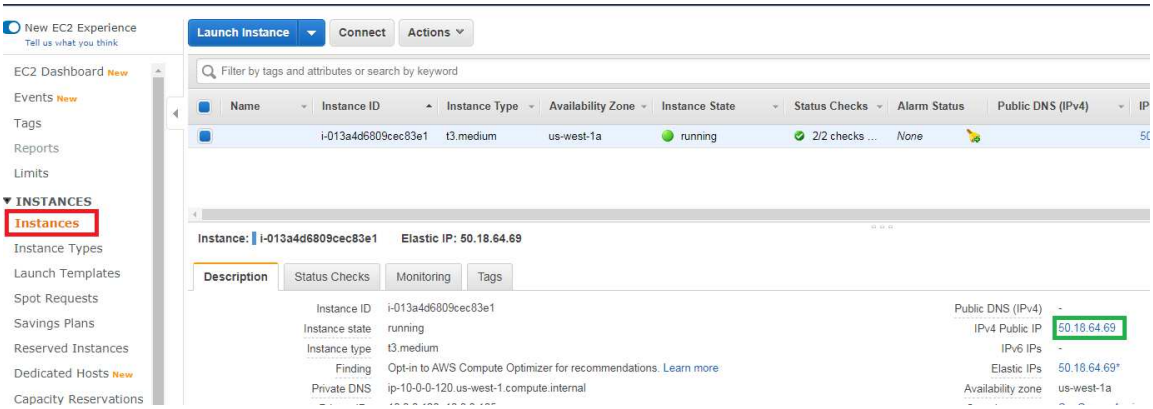

EC2 > Elastic IP addresses > Allocate Elastic IP address
Allocate Elastic IP address
Allocate an Elastic IP address by selecting the public IPv4 address pool from which the public IP address is to be allocated. Elastic IP addresses You can have one Elastic IP (EIP) address associated with a running instance at no charge. If you associate additional EIPs with that instance, you will be charged for each additional EIP associated with that instance on a pro rata basis. Additional EIPs are only available in Amazon VPC. To ensure efficient use of Elastic IP addresses, we impose a small hourly charge when these IP addresses are not associated with a running instance or when they are associated with a stopped instance or unattached network interface. Learn more
Elastic IP address settings
Public IPv4 address pool
Public IP addresses are allocated from Amazon's pool of public IP addresses, from a pool that you own and bring to your account, or from a pool that you own and continue to advertise..
Amazon's pool of IPv4 addresses
Public IPv4 address that you bring to your AWS account(option disabled because no pools found) Learn more
Customer owned pool of IPv4 addresses(option disabled because no customer owned pools found) Learn more
Cancel
Allocate


Elastic IP address allocated.
Elastic IP address 50.18.64.69
Associate this Elastic IP address
Elastic IP addresses (1/1)
Actions
Allocate Elastic IP address
View details
Release Elastic IP addresses
Associate Elastic IP address
Disassociate Elastic IP address
Public IPv4 address: 50.18.64.69
Clear filters
50.18.64.69
eipalloc-05485afea099f51e5





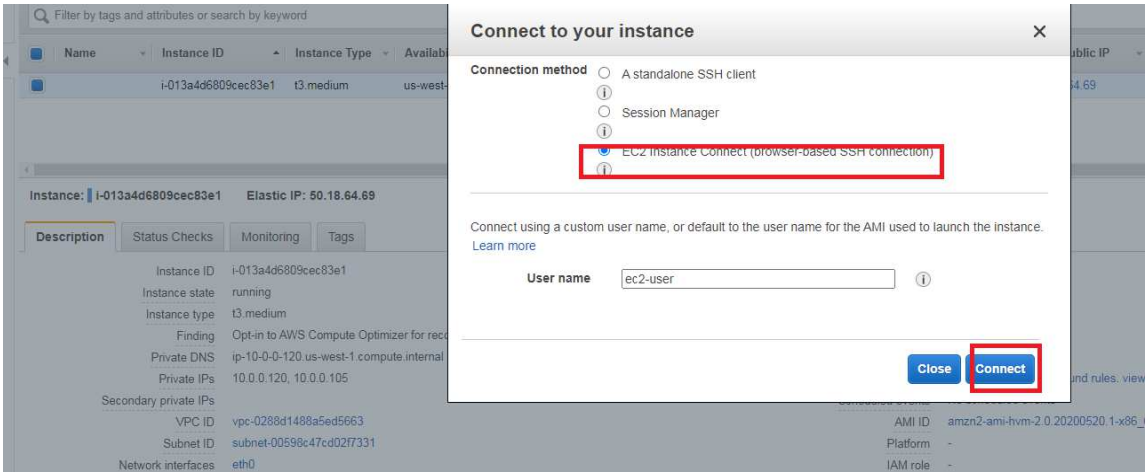
When you go to EC2 Instance section, you can see the assigned IP to the instance



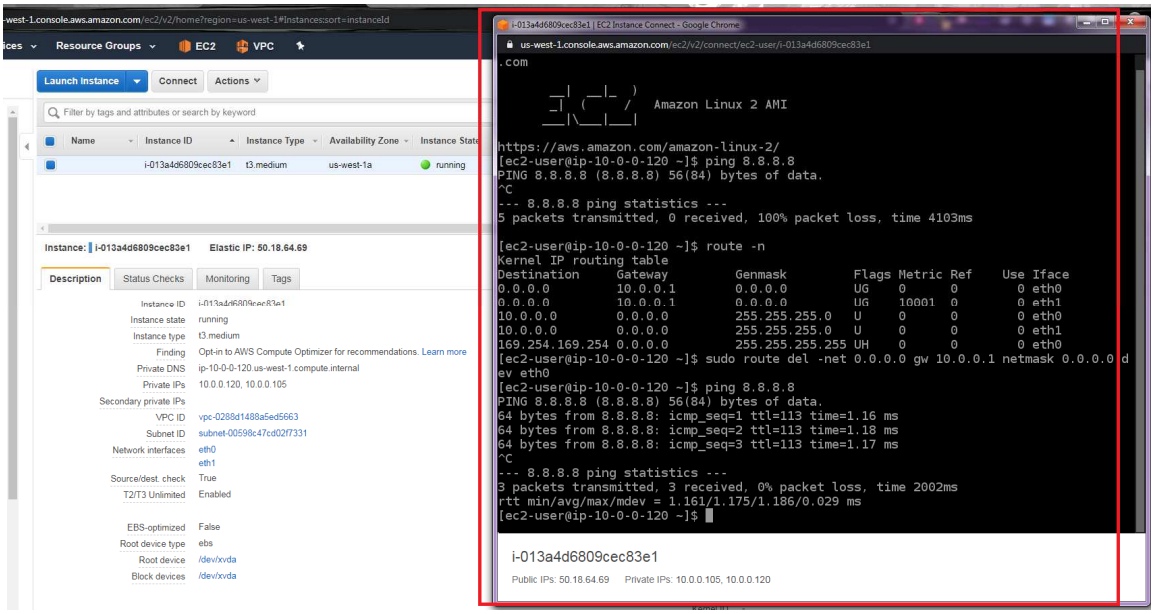
Review using Web Console

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 section. You have to jump to that section to complete the Lab.


Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
Name
Instance ID
Instance Type
Availability Zone
i-013a4d6809cec83e1
t3.medium
us-west-1a
running
Instance: i-013a4d6809cec83e1 Elastic IP: 50.18.64.69
Description
Status Checks
Monitoring
Tags
Instance ID i-013a4d6809cec83e1
Instance state running
Instance type t3.medium
Finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS ip-10-0-0-120.us-west-1.compute.internal
Private IPs 10.0.0.120, 10.0.0.105
Secondary private IPs
VPC ID vpc-0288d1488a5ed5663
Subnet ID subnet-00598c47cd02f7331
Network interfaces eth0
eth1
Source/dest. check True
T2/T3 Unlimited Enabled
EBS-optimized False
Root device type ebs
Root device /dev/xvda
Block devices /dev/xvda
i-013a4d6809cec83e1 | EC2 Instance Connect - Google Chrome
us-west-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-013a4d6809cec83e1
.com
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-120 ~]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4103ms
[ec2-user@ip-10-0-0-120 ~]$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 10.0.0.1 0.0.0.0 UG 0 0 0 eth0
0.0.0.0 10.0.0.1 0.0.0.0 UG 10001 0 0 eth1
10.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
10.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
169.254.169.254 0.0.0.0 255.255.255.255 UH 0 0 0 eth0
[ec2-user@ip-10-0-0-120 ~]$ sudo route del -net 0.0.0.0 gw 10.0.0.1 netmask 0.0.0.0 d
ev eth0
[ec2-user@ip-10-0-0-120 ~]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=1.16 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=1.18 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=1.17 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.161/1.175/1.186/0.029 ms
[ec2-user@ip-10-0-0-120 ~]$
i-013a4d6809cec83e1
Public IPs: 50.18.64.69 Private IPs: 10.0.0.105, 10.0.0.120

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%
```

```
C:\Code\bsg-saa-c02\AWS_SAA>set vpcn_Mask="10.0.0.0/16"

C:\Code\bsg-saa-c02\AWS_SAA>set pbsn1_Mask="10.0.0.0/24"

C:\Code\bsg-saa-c02\AWS_SAA>set instance_type="t2.medium"

C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 create-vpc --cidr-block %vpcn_Mask%|jq ".Vpc.VpcId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>set /p vpcn_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 modify-vpc-attribute --vpc-id %vpcn_Id% --enable-dns-hostnames "{\"Value\":true}"


C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 create-subnet --vpc-id %vpcn_Id% --cidr-block %pbsn1_Mask% --availability-zone %first_az%|jq ".Subnet.SubnetId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>set /p pbsn1_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 create-internet-gateway|jq ".InternetGateway.InternetGatewayId"  >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>set /p IGW_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 attach-internet-gateway --vpc-id %vpcn_Id% --internet-gateway-id %IGW_Id%


C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 create-route-table --vpc-id %vpcn_Id%|jq ".RouteTable.RouteTableId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>set /p Public_RT_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 create-route --route-table-id %Public_RT_Id% --destination-cidr-block 0.0.0.0/0 --gateway-id %IGW_Id%
{
    "Return": true
}


C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 associate-route-table  --subnet-id %pbsn1_Id% --route-table-id %Public_RT_Id%
{
    "AssociationId": "rtbassoc-0c74f666cf7a21228",
    "AssociationState": {
        "State": "associated"
    }
}
```

Create keys, Sec Group (Labs4c1)

```
rem Crear las llaves para el SSH a las nuevas instancias y convertirlas a PP
K 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
```

```
C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 create-key-pair --key-name Lab5a --query "KeyMaterial" --output text > Lab5a.pem

C:\Code\bsg-saa-c02\AWS_SAA>winscp.com /keygen "Lab5a.pem" /output="Lab5a.ppk"
Key saved to "Lab5a.ppk".

C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 create-security-group --group-name "SecGroup_A" --description "Security group for Instance A" --vpc-id %vpcn_Id% |jq ".GroupId">tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>set /p SecGroup_A_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 authorize-security-group-ingress --group-id %SecGroup_A_Id% --protocol tcp --port 22 --cidr 0.0.0.0/0


C:\Code\bsg-saa-c02\AWS_SAA>aws ec2 authorize-security-group-ingress --group-id %SecGroup_A_Id% --protocol tcp --port 80 --cidr 0.0.0.0/0


C:\Code\bsg-saa-c02\AWS_SAA>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/developergui
de/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 eje
cuta, 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 aq
ui 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
```

```
C:\Code\bsg-saa-c02\AWS_SAA>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, &CreationDa
te))[:1].ImageId" --output text >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>set /p AMI= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA>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
{
    "SpotPriceHistory": [
        {
            "AvailabilityZone": "us-west-1c",
            "InstanceType": "t2.medium",
            "ProductDescription": "Linux/UNIX",
            "SpotPrice": "0.016600",
            "Timestamp": "2020-06-23T22:50:55+00:00"
        },
        {
            "AvailabilityZone": "us-west-1a",
            "InstanceType": "t2.medium",
            "ProductDescription": "Linux/UNIX",
            "SpotPrice": "0.016600",
            "Timestamp": "2020-06-23T22:50:55+00:00"
        },
        {
            "AvailabilityZone": "us-west-1c",
            "InstanceType": "t2.medium",
            "ProductDescription": "Linux/UNIX",
            "SpotPrice": "0.016600",
            "Timestamp": "2020-06-22T22:50:02+00:00"
        },
        {

C:\Code\bsg-saa-c02\AWS_SAA>echo AMI es %AMI% Sec Group es %SecGroup_A_Id% subnet es %pbsn1_Id%
AMI es ami-04e59c05167ea7bd5 Sec Group es "sg-04e342ac4505c678a" subnet es "subnet-07d5898b3607d27db"
```

```
C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>aws ec2 request-spot-instances --spot-price "0.03" --block-duration-minutes 60 --instance-count 1 --type "one-time" --launch-specification file://config.json
{
    "SpotInstanceRequests": [
        {
            "BlockDurationMinutes": 60,
            "CreateTime": "2020-06-25T10:37:53+00:00",
            "LaunchSpecification": {
                "SecurityGroups": [
                    {
                        "GroupName": "SecGroup_A",
                        "GroupId": "sg-04e342ac4505c678a"
                    }
                ],
                "ImageId": "ami-04e59c05167ea7bd5",
                "InstanceType": "t3.medium",
                "KeyName": "LabSa",
                "Placement": {
                    "AvailabilityZone": "us-west-1a"
                },
                "SubnetId": "subnet-07d5898b3607d27db",
                "Monitoring": {
                    "Enabled": false
                }
            },
            "ProductDescription": "Linux/UNIX",

C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>aws ec2 describe-spot-instance-requests --query "SpotInstanceRequests[*].{ID:InstanceId}"|jq ".[].ID" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>set /p InstanceId= <tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>aws ec2 create-network-interface --subnet-id %pbsn1_Id% --description "Additional Network Interface Instance A" --groups %SecGroup_A_Id%|jq ".NetworkInterface.NetworkInt
erfaceId">tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>set /p ENI_Id= < 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
```

```
C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>aws ec2 create-network-interface --subnet-id %pbsn1_Id% --description "Additional Network Interface Instance A" --groups %SecGroup_A_Id%|jq ".NetworkInterface.NetworkInt
erfaceId">tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>set /p ENI_Id= < tmpFile


C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>aws ec2 attach-network-interface --network-interface-id %ENI_Id% --instance-id %InstanceId% --device-index 1
{
    "AttachmentId": "eni-attach-003f3d5eb0ccb2bd2"
}


C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>aws ec2 allocate-address --domain vpc |jq ".AllocationId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>set /p EIP_for_EC2= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s5c1\CLI>aws ec2 associate-address --allocation-id %EIP_for_EC2% --network-interface-id %ENI_Id%
{
    "AssociationId": "eipassoc-0c32bc068f37594e9"
}
```

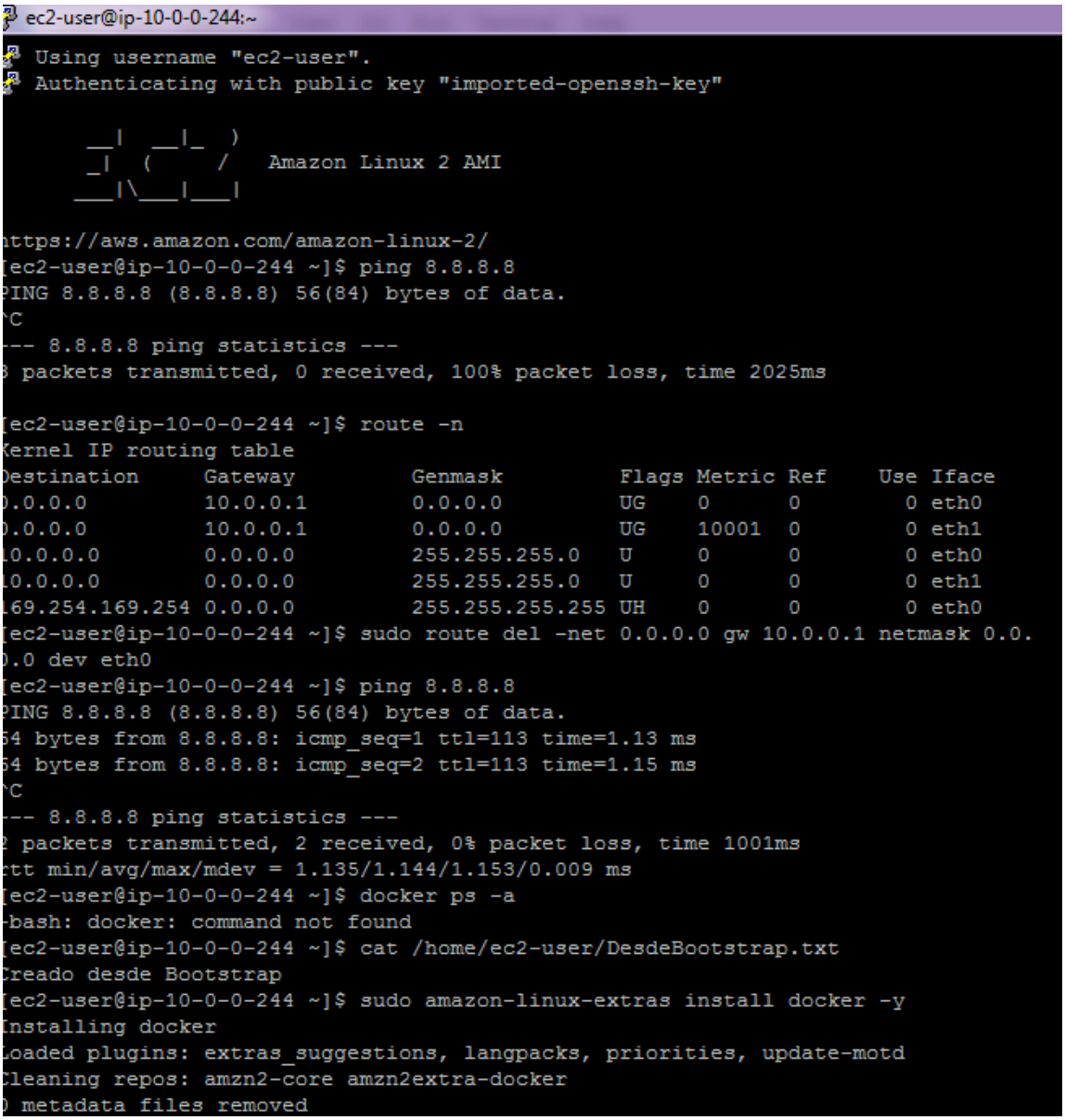
Review using CLI

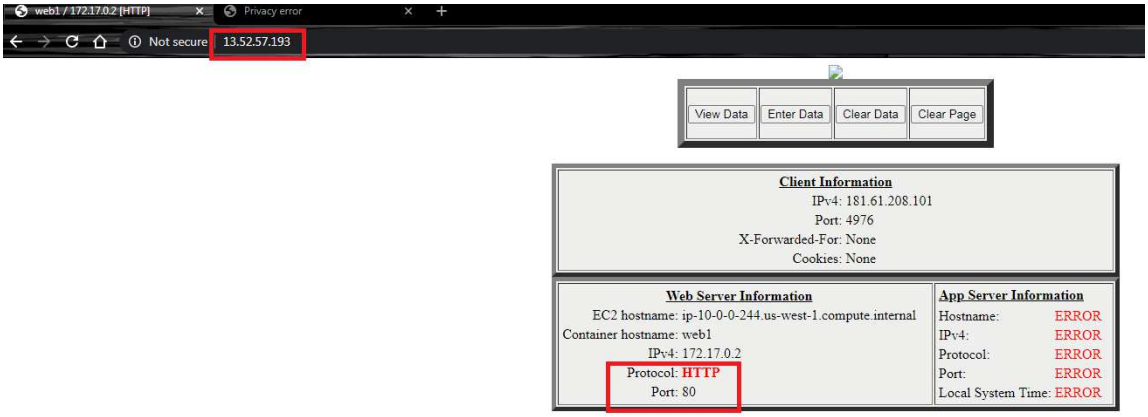
Go to section Review Configuration using Putty

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 pe
rmite 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
```

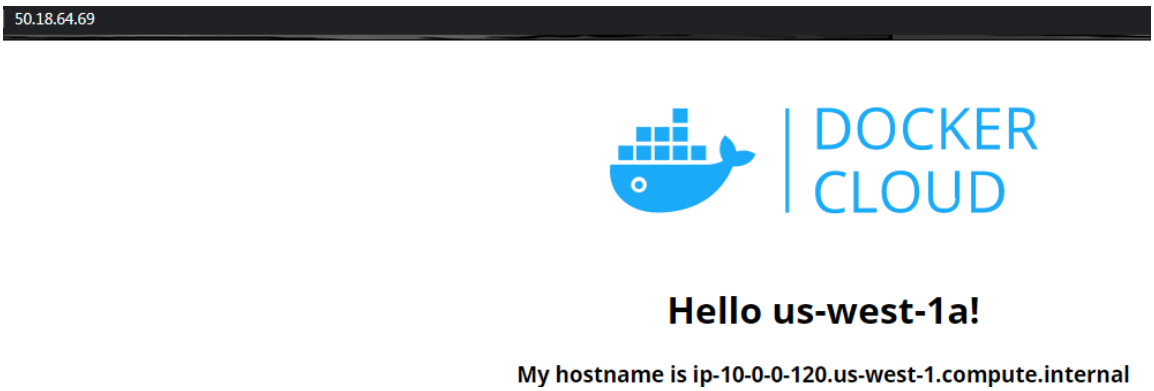
```
rem Check web with that IP on port 80 and 443


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
```



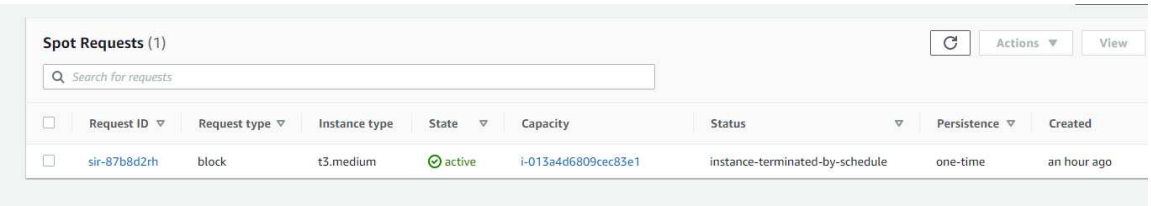

web1 / 172.17.0.2 [HTTP]
Privacy error
Not secure
13.52.57.193
View Data
Enter Data
Clear Data
Clear Page
Client Information
IPv4: 181.61.208.101
Port: 4976
X-Forwarded-For: None
Cookies: None
Web Server Information
EC2 hostname: ip-10-0-0-244.us-west-1.compute.internal
Container hostname: web1
IPv4: 172.17.0.2
Protocol: HTTP
Port: 80
App Server Information
Hostname: ERROR
IPv4: ERROR
Protocol: ERROR
Port: ERROR
Local System Time: ERROR

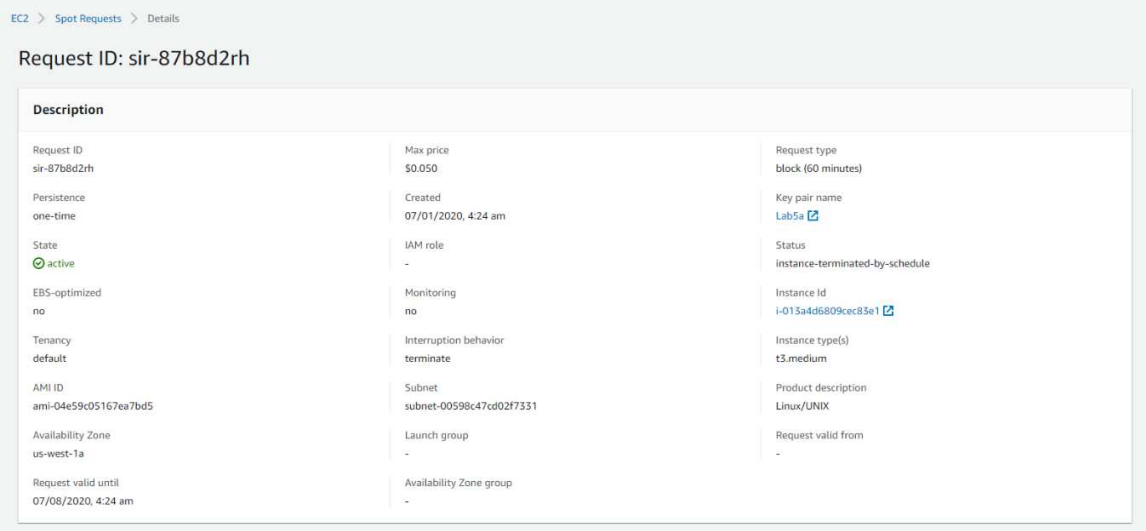

web1 / 172.17.0.2 [HTTP]
web1 / 172.17.0.2 [HTTPS]
Not secure
13.52.57.193
View Data
Enter Data
Clear Data
Clear Page
Client Information
IPv4: 181.61.208.101
Port: 4932
X-Forwarded-For: None
Cookies: None
Web Server Information
EC2 hostname: ip-10-0-0-244.us-west-1.compute.internal
Container hostname: web1
IPv4: 172.17.0.2
Protocol: HTTPS
Port: 443
App Server Information
Hostname: ERROR
IPv4: ERROR
Protocol: ERROR
Port: ERROR
Local System Time: ERROR

Or


50.18.64.69
DOCKER
CLOUD
Hello us-west-1a!
My hostname is ip-10-0-0-120.us-west-1.compute.internal

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 (Web Console) where you see that Spot Request is fulfilled.
2. The last screenshot of Review using Web Console (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.