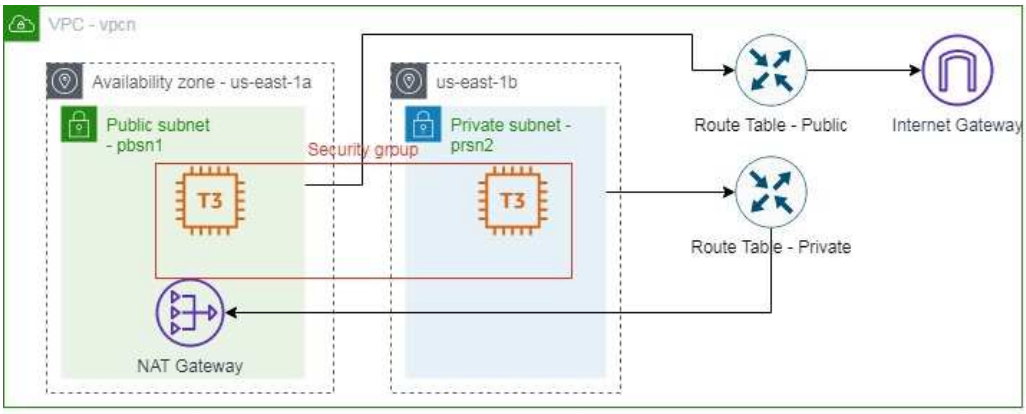
Contents

Purpose

To create a common computing infrastructure in a public and private subnet, so you have to configure and connect routing tables and internet or NAT gateway.

General Diagram

One VPC with two subnets, with instances connected on each subnet. Configuration to outbound connections has to be made on Routing Tables using Internet Gateway and NAT Gateway.



Simple Web Server using Python- SSH to Private Instance

Check Outbound Connection using NAT GW

Steps:

1. Create VPC and Subnets.
3. Create empty routing tables.
4. Create IGW and attached to VPC.
5. Create NAT GW.
6. Assign default route for Routing Tables (IGW and NAT GW).
7. Assign Routing Tables to Subnets.
8. Create Keypair.
9. Launch a instance on each subnet and create and assign Security Group.
10. Make the procedure using SSH and SFTP.

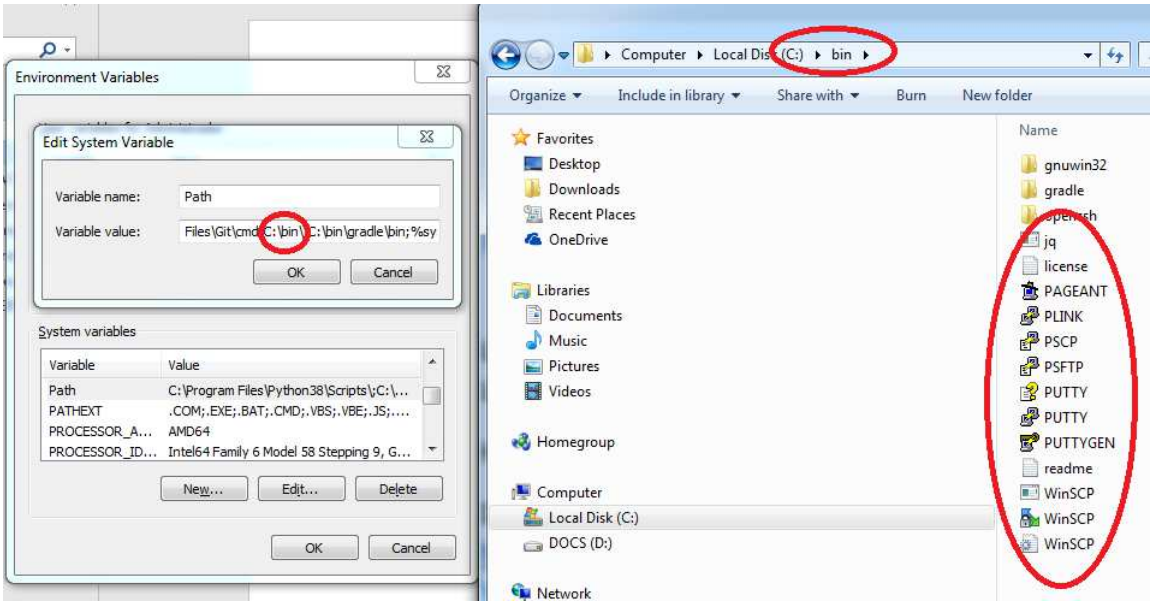
Prerequisites

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

Have installed putty and winscp on Windows; and those files on a folder in the PATH environment.

Download complete and portable putty and winscp using https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html and https://winscp.net/eng/downloads.php

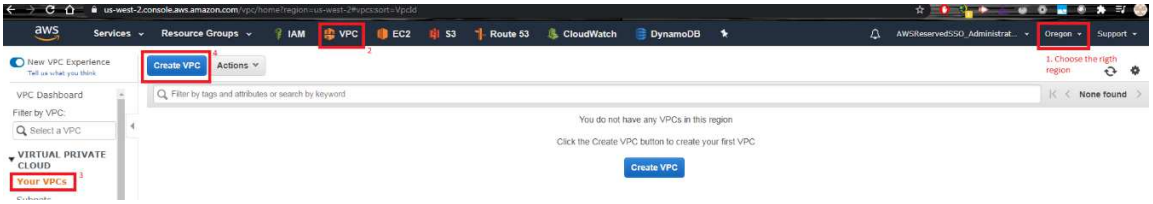
We use winscp as SFTP client (put Keypair on Public Instance) and for modifying PEM to PPK file on Windows. You can use Cyberduck as SFTP Client on MacOS.

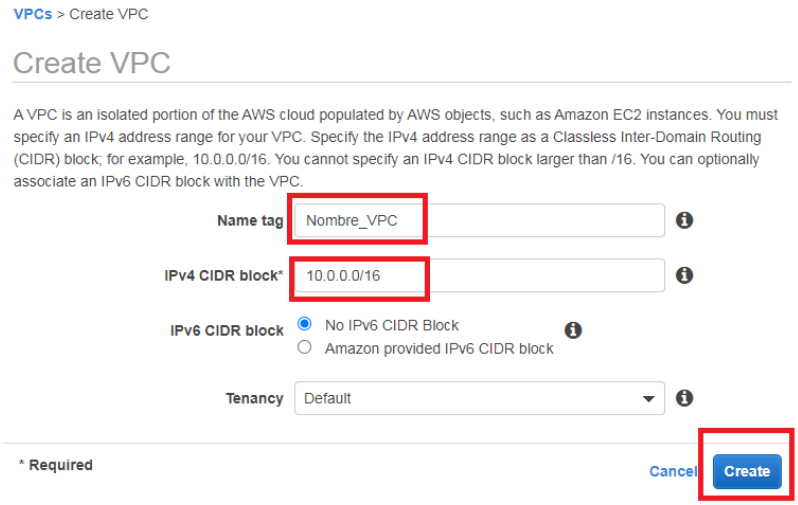


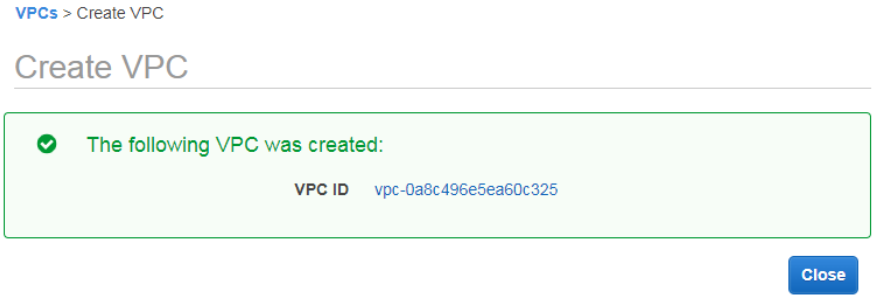
Lab 4A: VPC with IGW and NAT GW

Lab 4A using Web Management Console

Create VPC

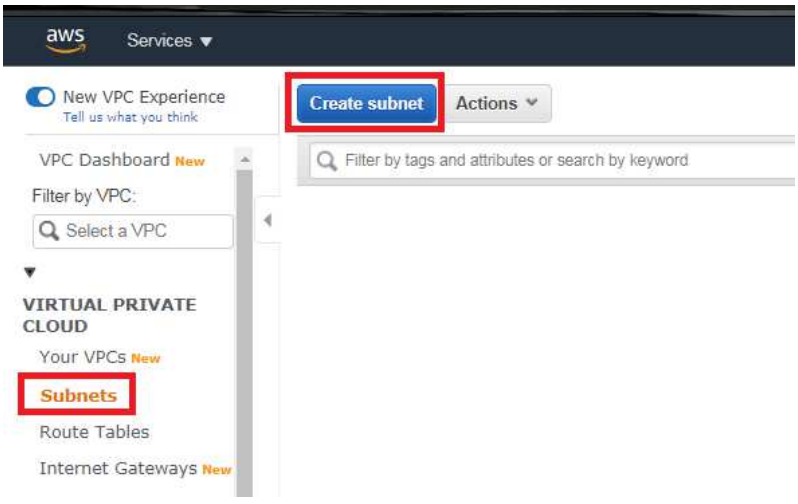


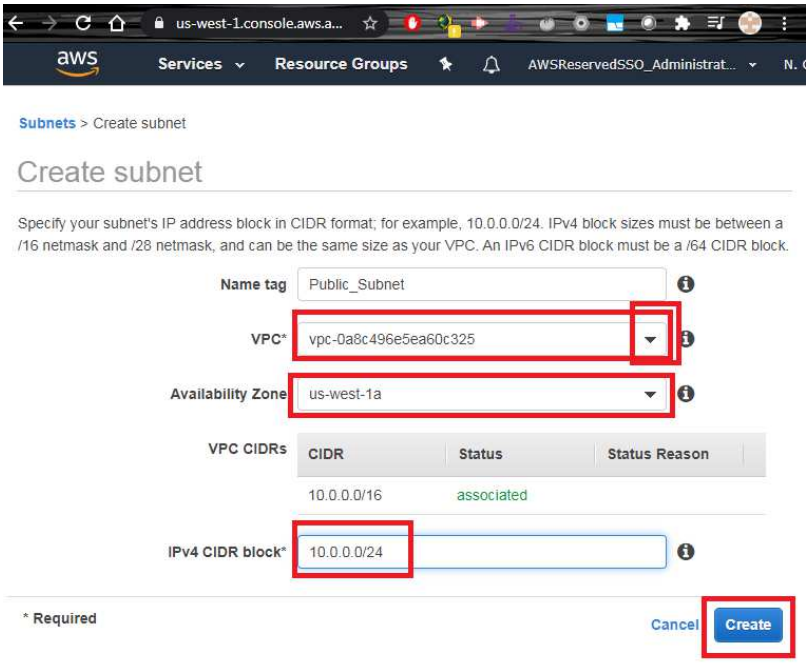




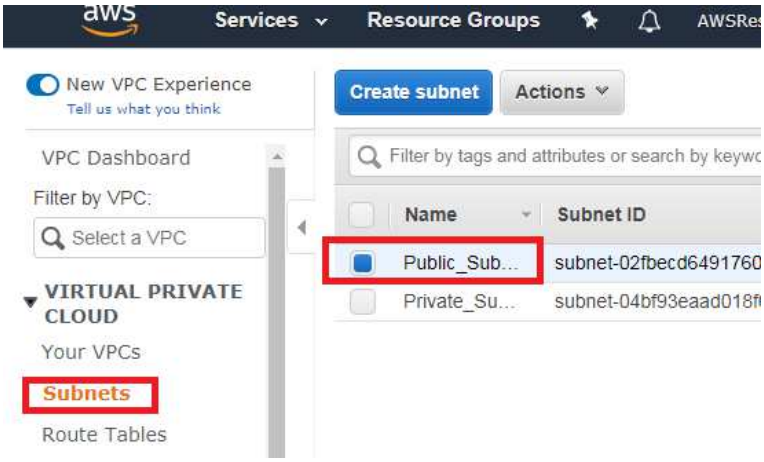
Create Subnets

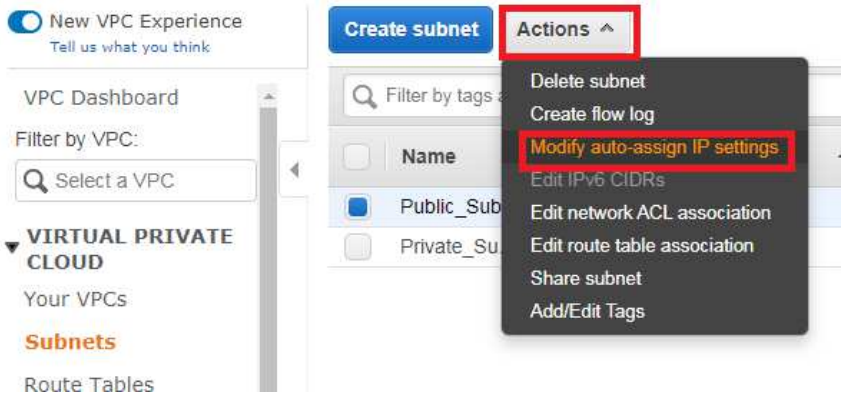
For Public Subnet,

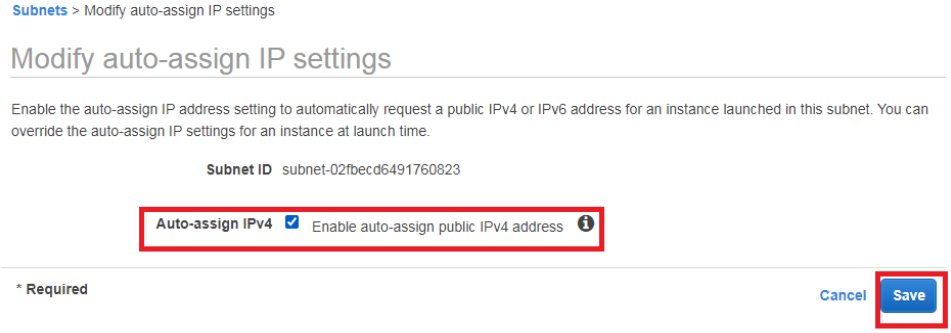




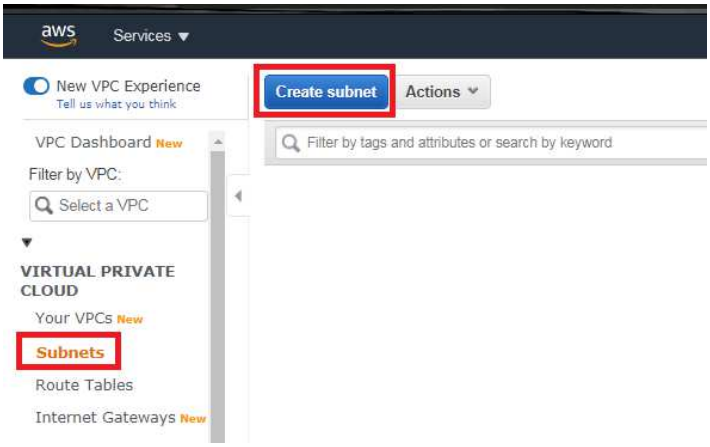
Modifying Public Subnet to assign Public IP to any instances on this subnet, you have to select the subnet and apply the feature.

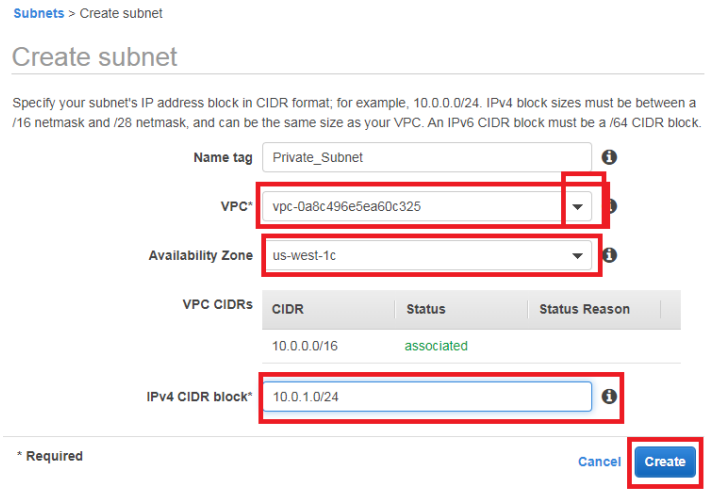


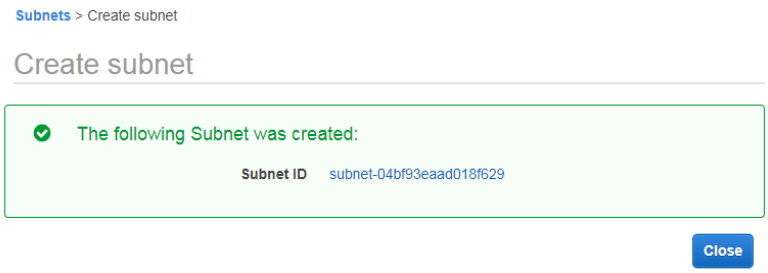




For Private Subnet,

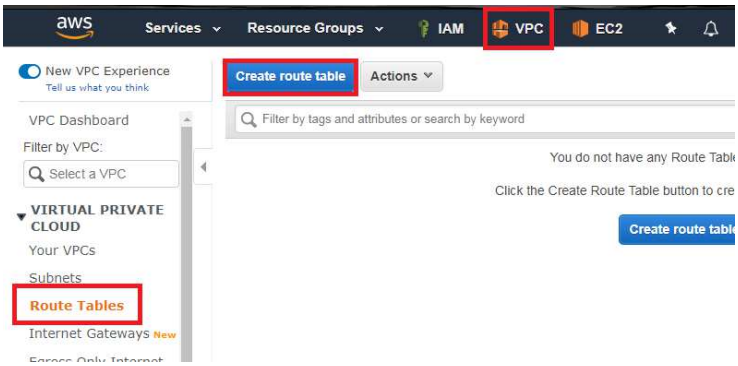


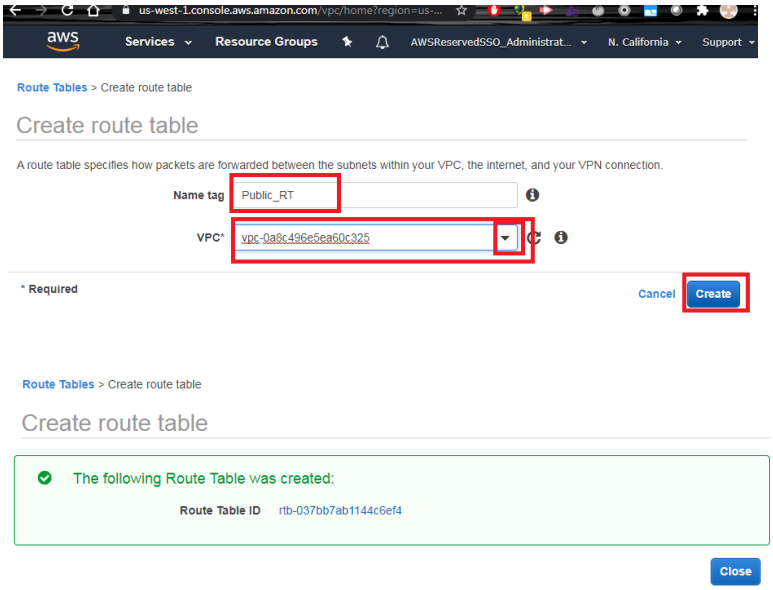




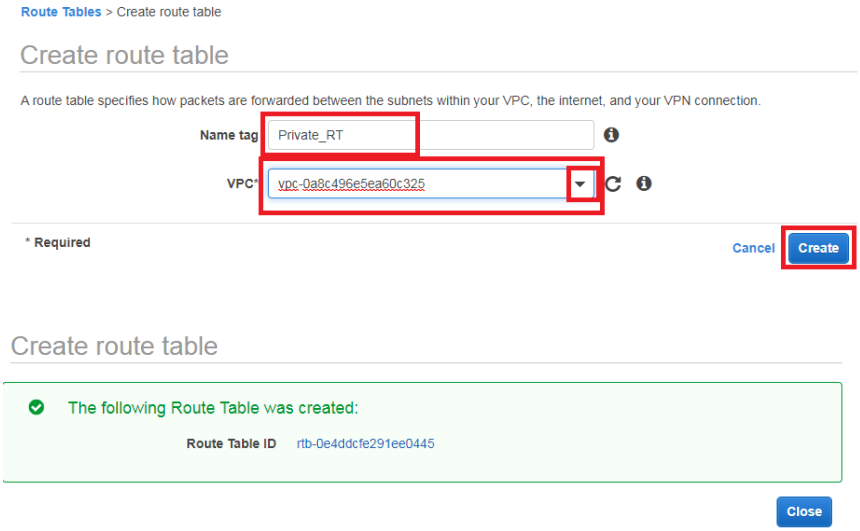
Create Routing Tables

For this step, we create empty routing tables without attached or assign anything. First, we create the public routing table,



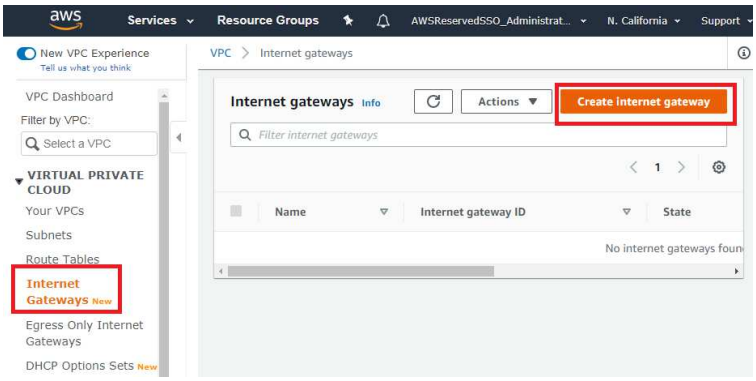


Then, we create the private routing table.

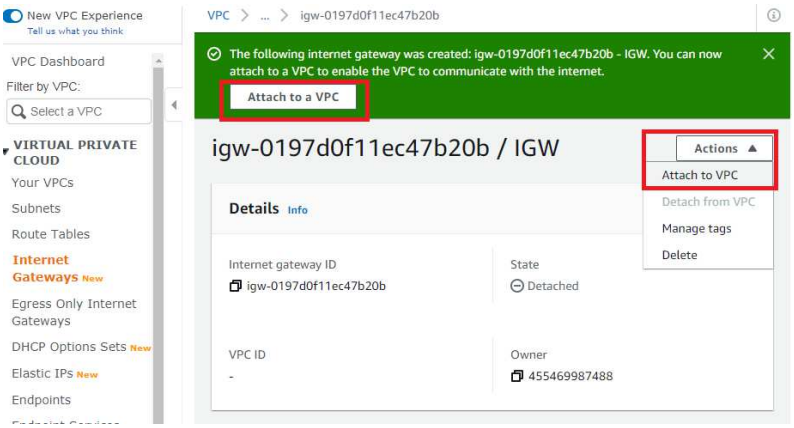


Create Internet Gateway (IGW)

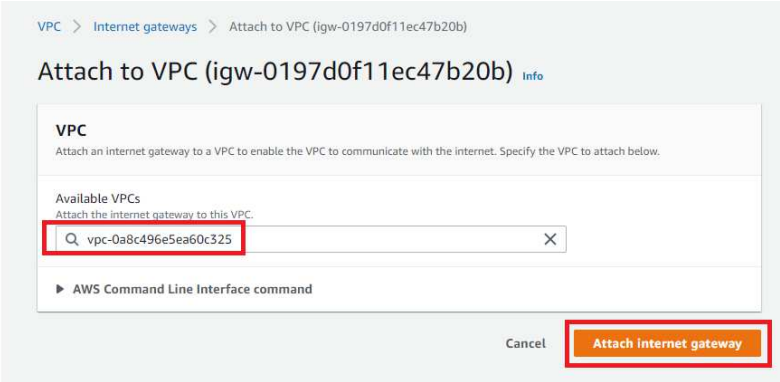
Create a IGW and then, attach to the VPC.



After it confirmation message, we attach to the VPC.

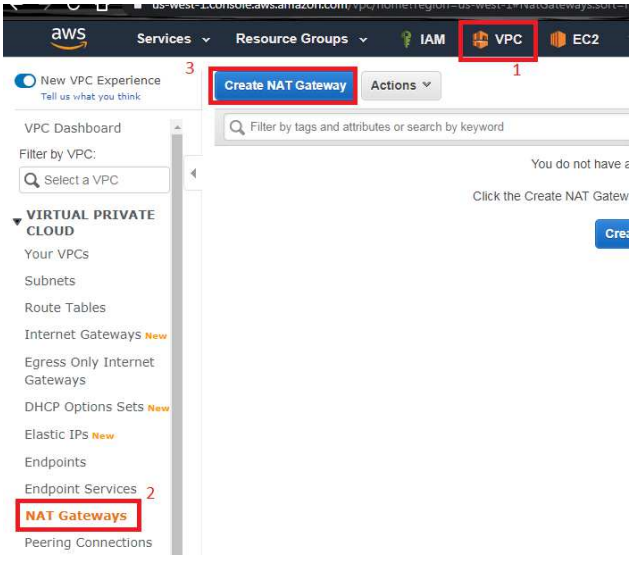


Select the VPC to attach the IGW,

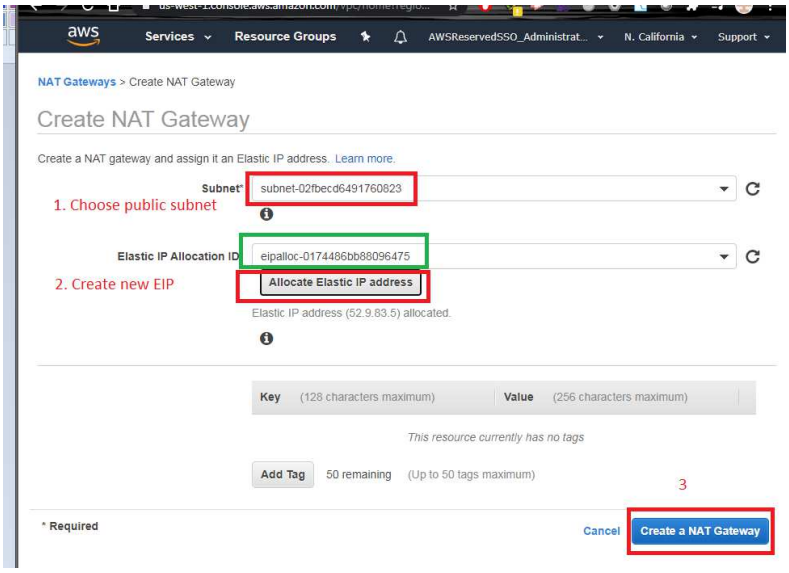


Confirmation message

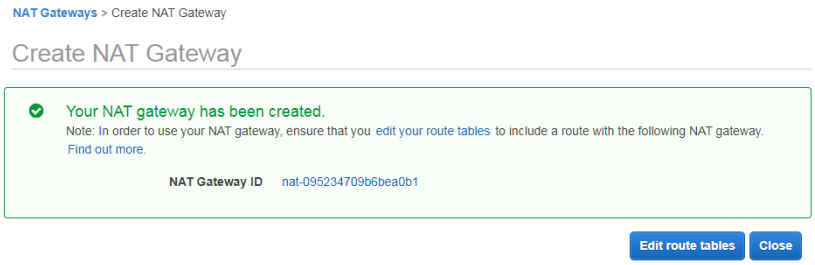
Create NAT Gateway (NAT GW)



In this info window, you have to create an Elastic IP for assign it to the NAT Gateway, therefore you click on “Allocate Elastic IP Address” and AWS Console create and assign an EIP for this NAT GW automatically.

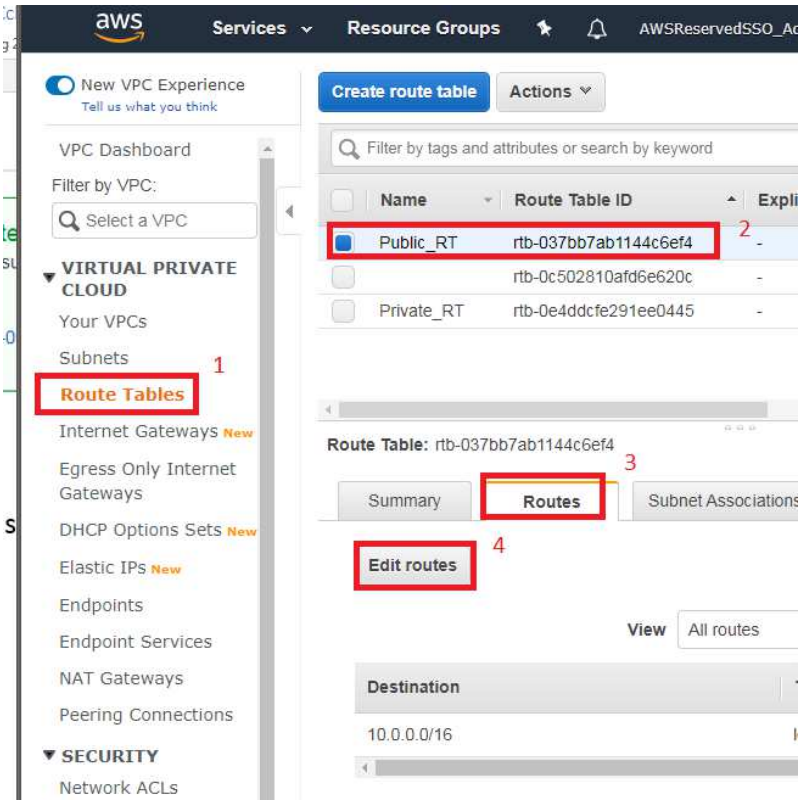


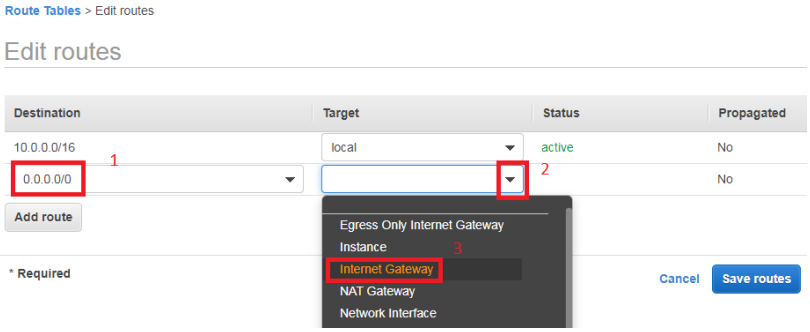
Confirmation message,

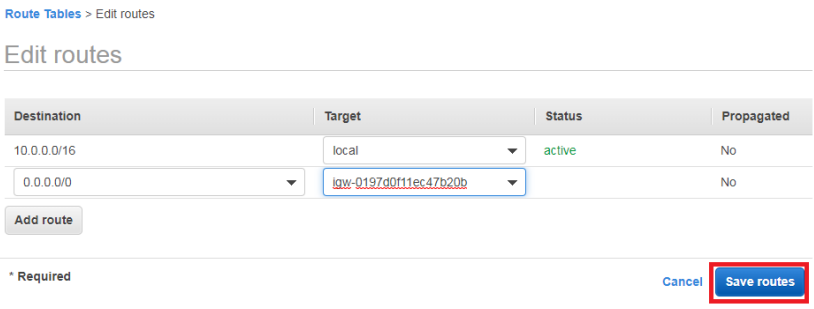


Assign IGW and NAT GW to Routing Tables

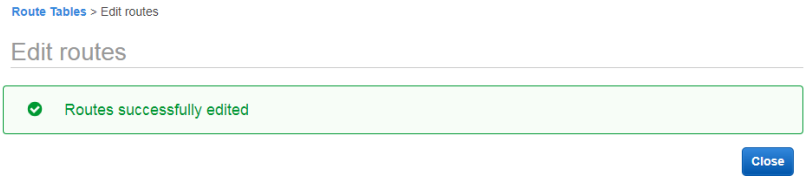
For Public Table, you modify routing tables to assign default route (0.0.0.0/0) to IGW.



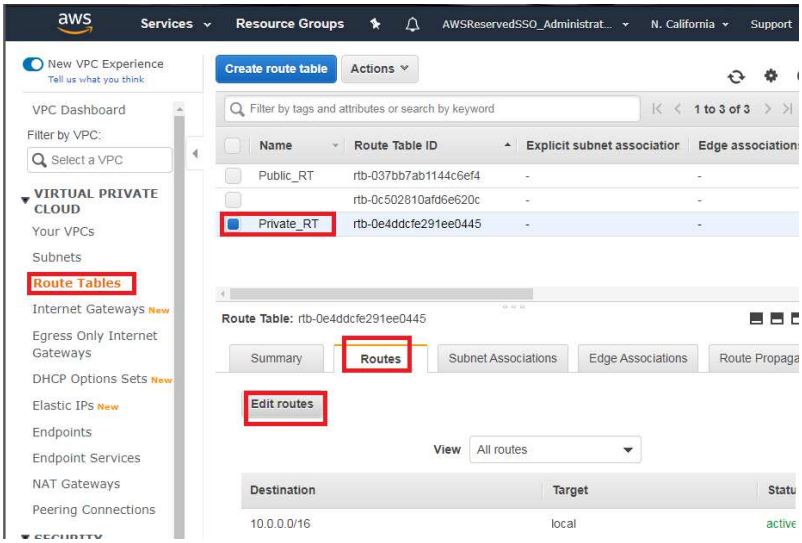


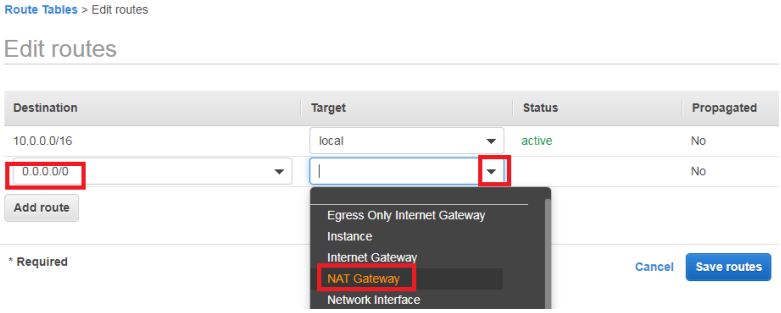


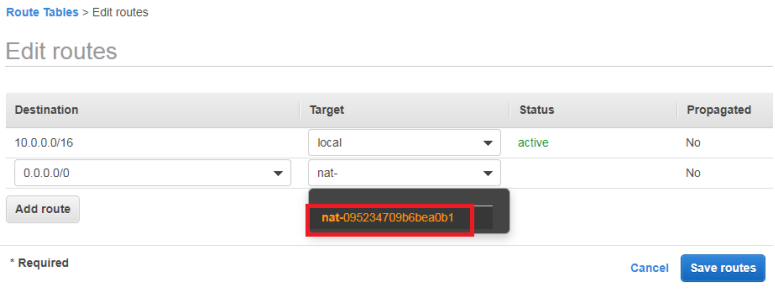
Confirmation message



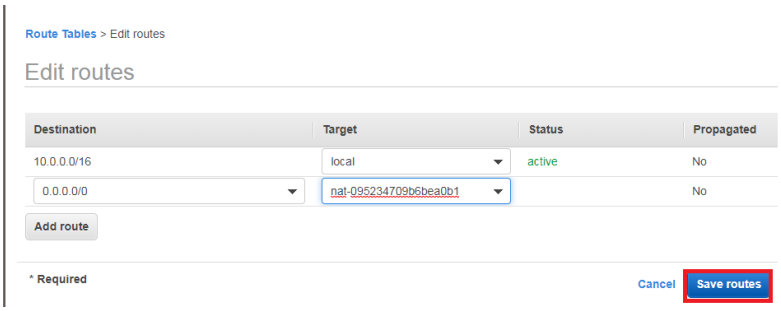
For Private Routing Table, you modify routing tables to assign default route (0.0.0.0/0) to NAT GW.

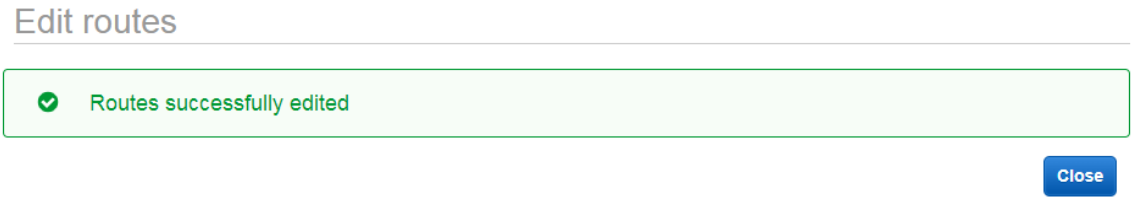






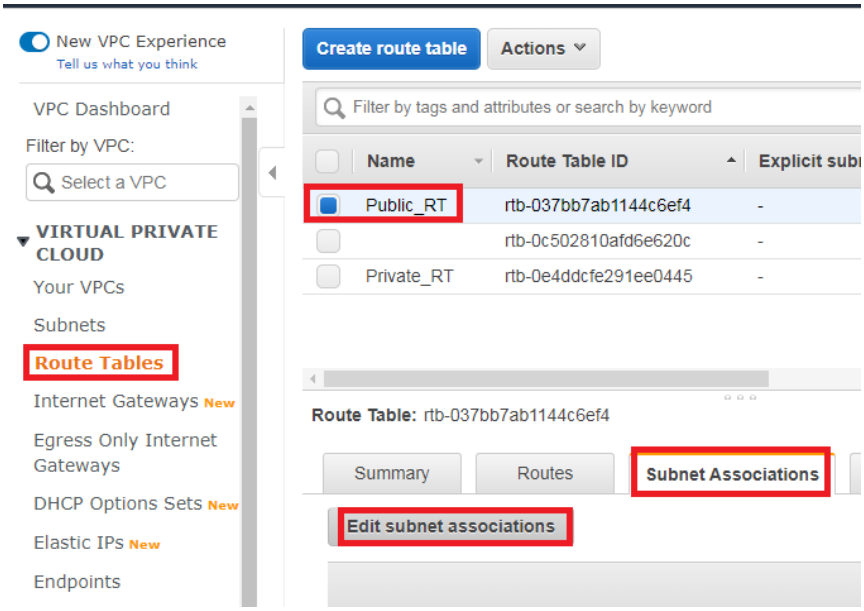
Confirmation message,

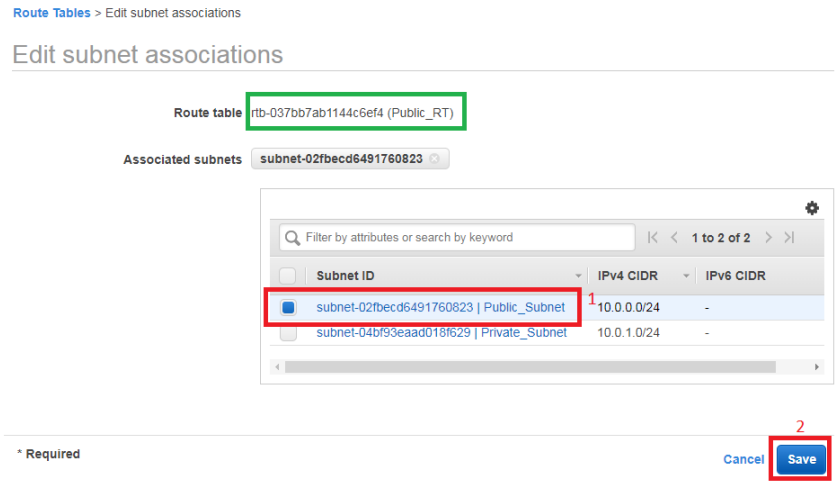




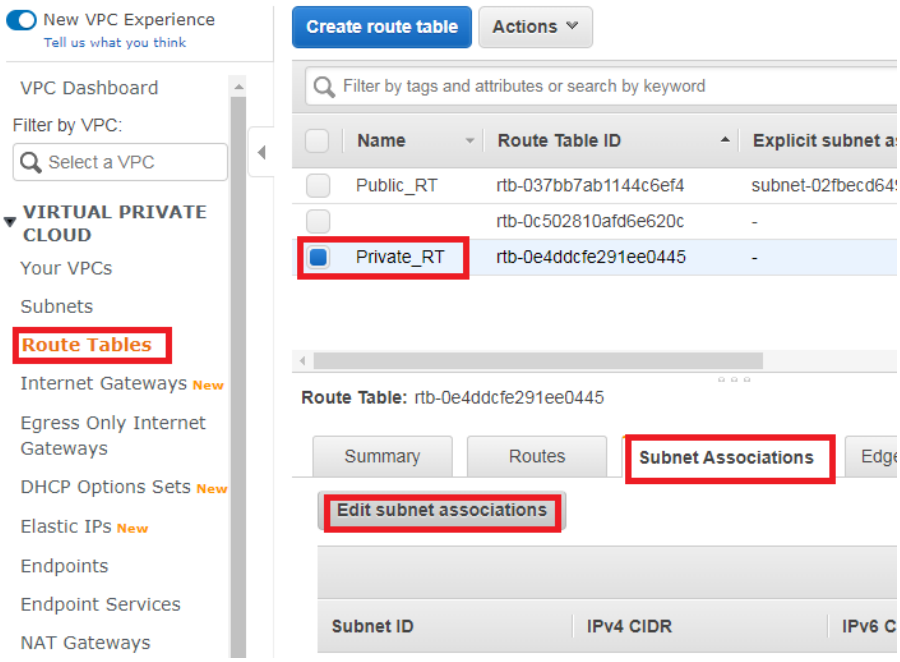
Assign Routing Tables to Subnets

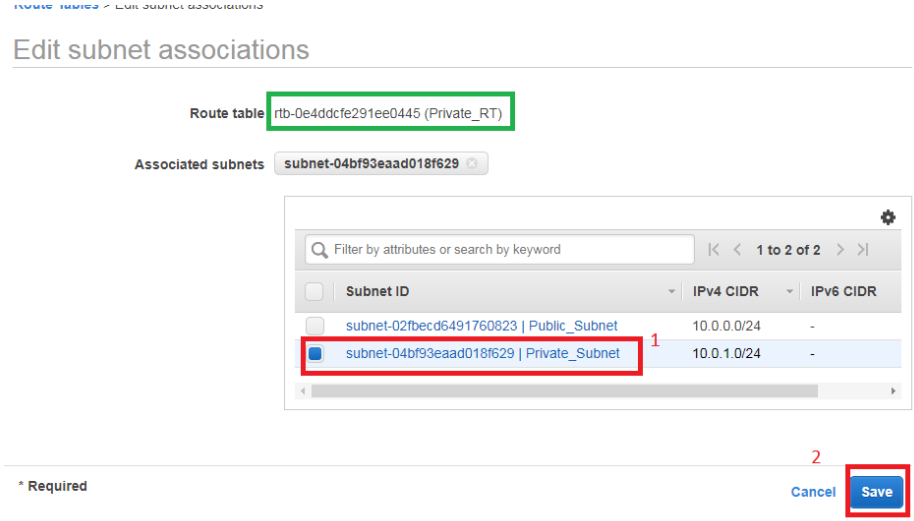
For Public Routing Table, you select the public subnet.





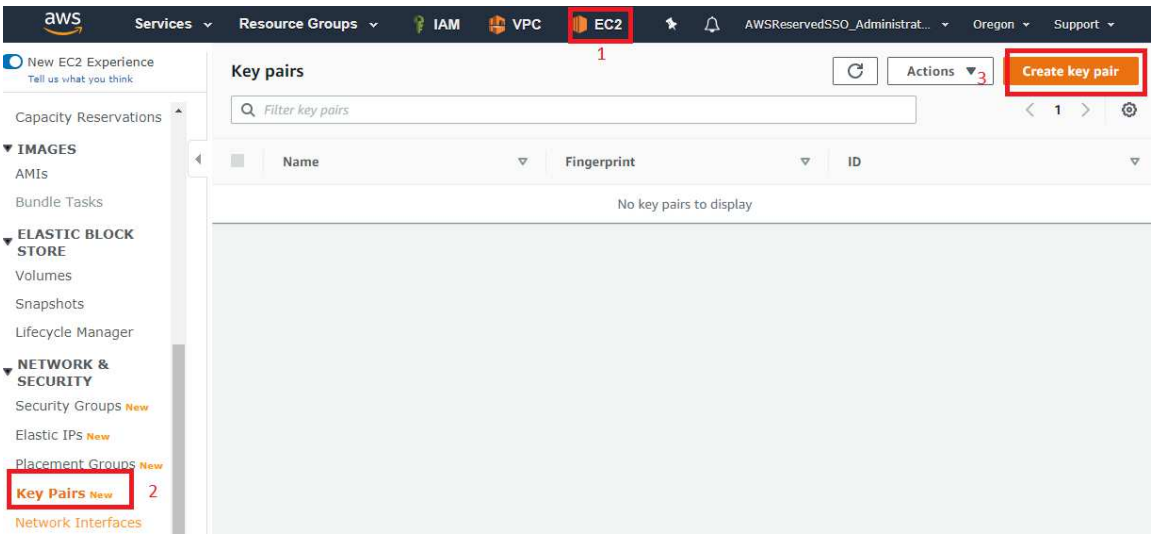
For Private Routing Table, you select private subnet.

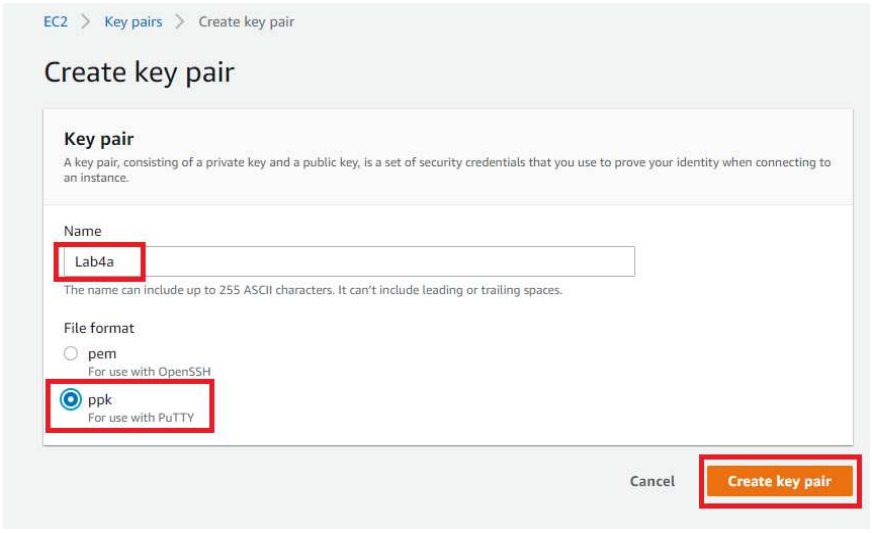




Create Key Pair to connect to Instances

A Keypair, it is a file to authenticate ec2-user (default user for Amazon Linux AMI). For this step, we create that file and use to connect it.





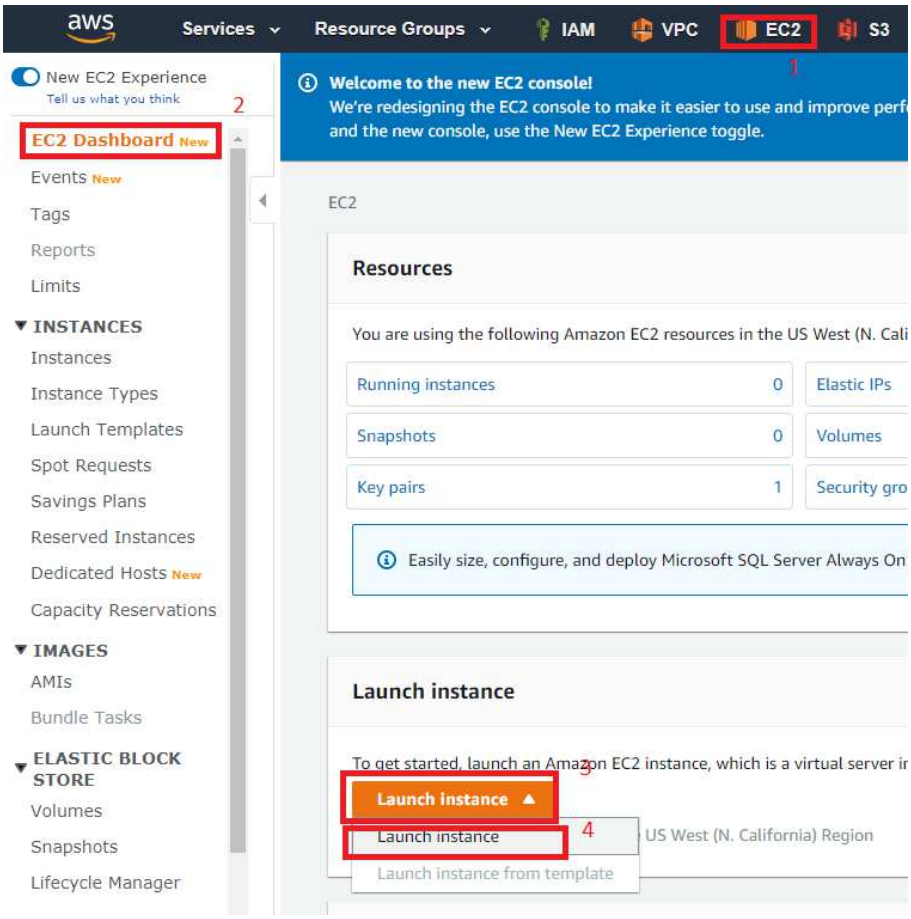
Store on the folder with the code of Command Line.

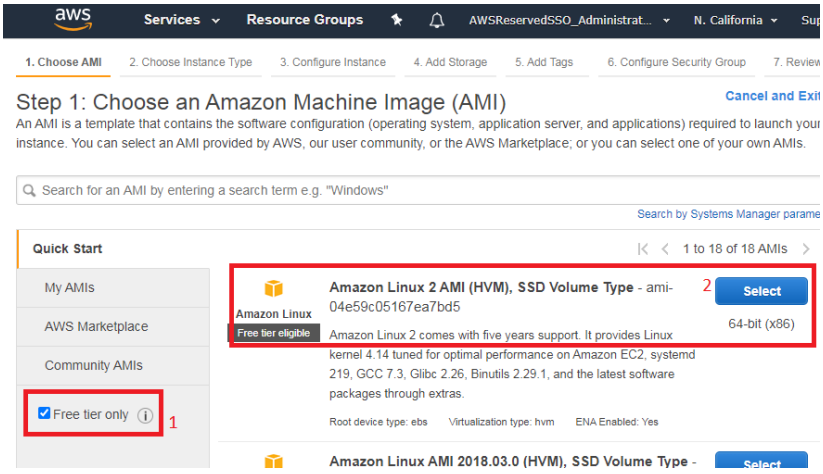
If you have Windows, we don't need to use winscp to make the transformation to ppk. You download ppk file to use putty.

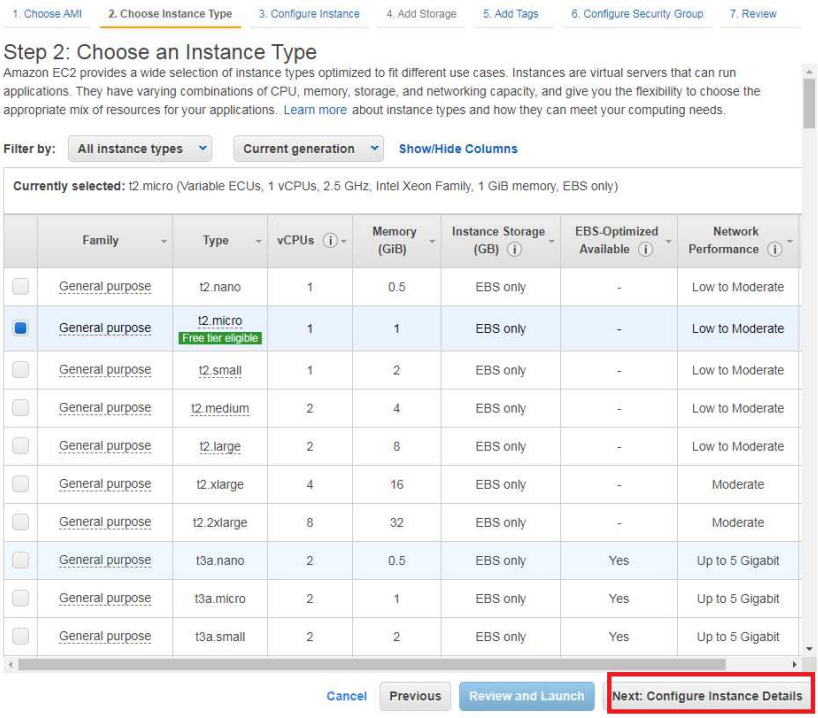
If you have MacOs, you don't need to download the ppk format, you have to use pem file to use SSH in command line.

Create EC2 instances

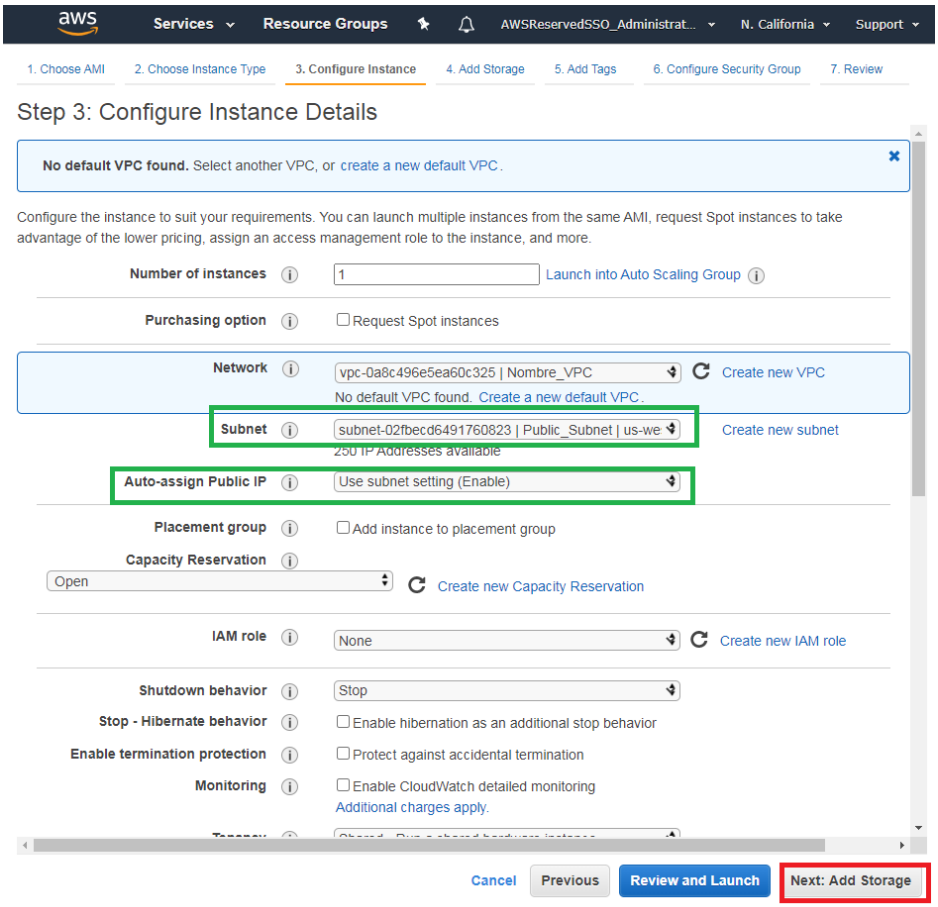
An instance is a Virtual Machine. Those steps are straight forward because we have to make it on detail on next session.


aws
Services
Resource Groups
IAM
VPC
EC2
S3
1
New EC2 Experience
Tell us what you think
2
EC2 Dashboard New
Events New
Tags
Reports
Limits
INSTANCES
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
Welcome to the new EC2 console!
We're redesigning the EC2 console to make it easier to use and improve perf
and the new console, use the New EC2 Experience toggle.
EC2
Resources
You are using the following Amazon EC2 resources in the US West (N. Cali
Running instances 0
Elastic IPs
Snapshots 0
Volumes
Key pairs 1
Security gro
Easily size, configure, and deploy Microsoft SQL Server Always On
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server ir
3
Launch instance
Launch instance
4
US West (N. California) Region
Launch instance from template

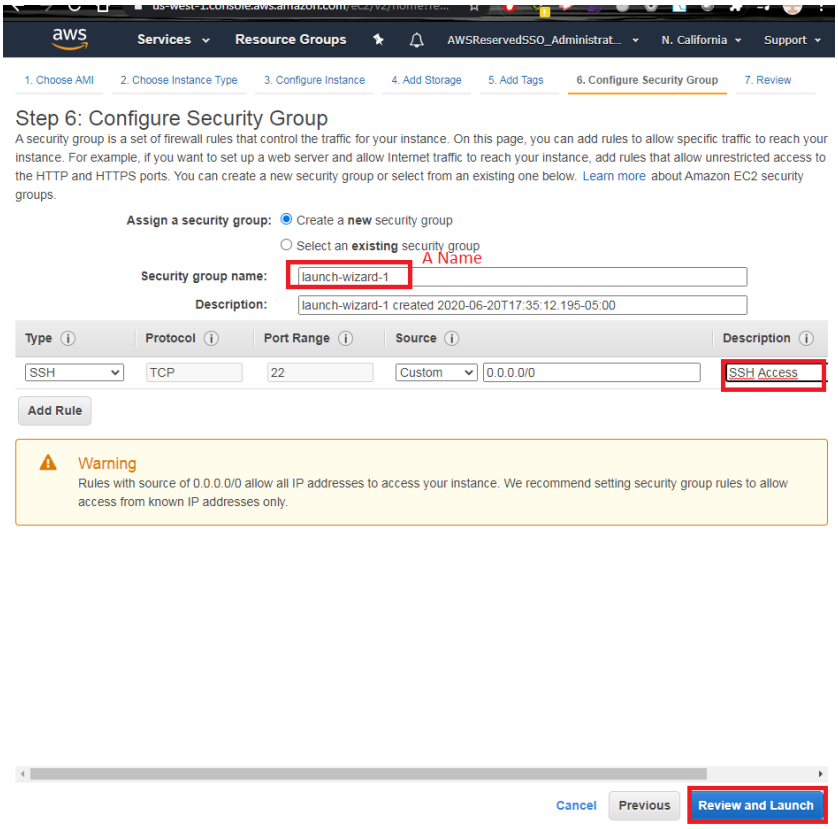

aws
Services
Resource Groups
AWSReservedSSO_Administrat...
N. California
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
An AMI is a template that contains the software 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.
Search for an AMI by entering a search term e.g. "Windows"
Search by Systems Manager parame
Quick Start
1 to 18 of 18 AMIs
My AMIs
AWS Marketplace
Community AMIs
Free tier only
1
Amazon Linux
Free tier eligible
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-04e59c05167ea7bd5
2
Select
64-bit (x86)
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type -
Select



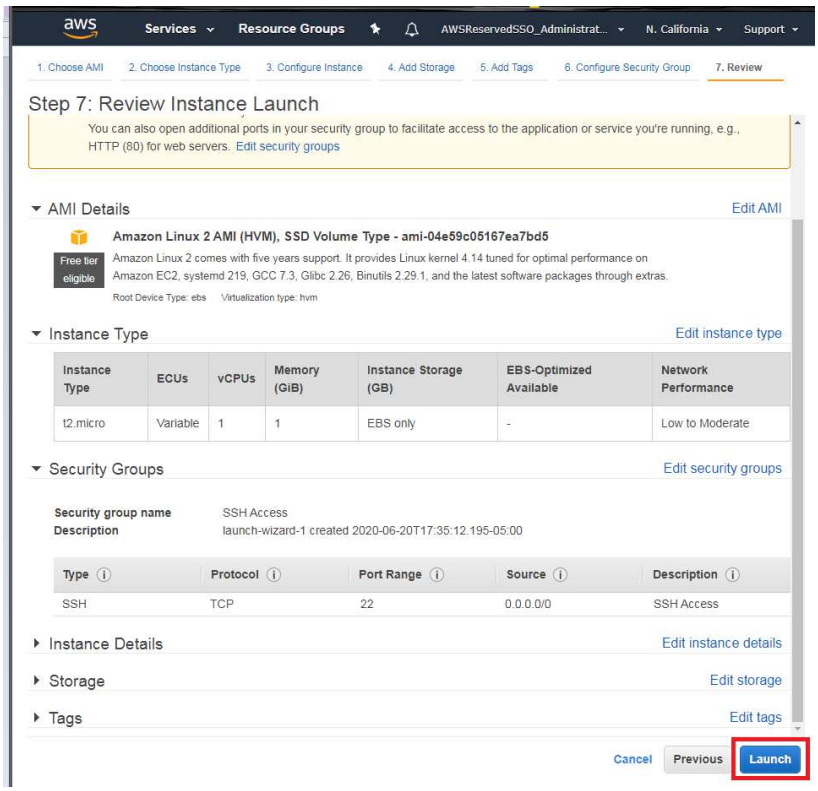
For this case, you create the public instance:

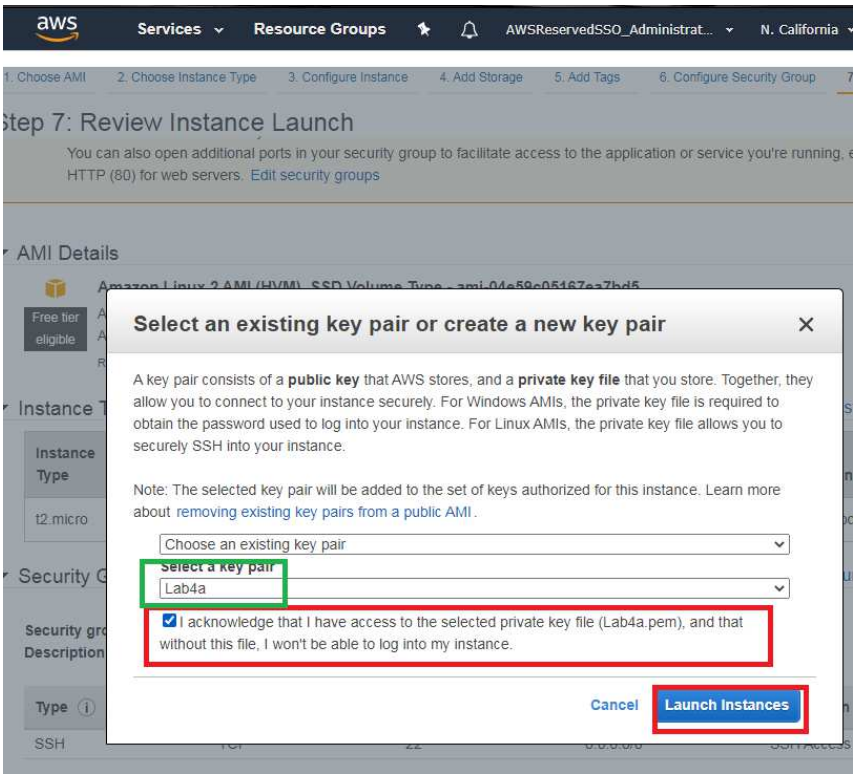


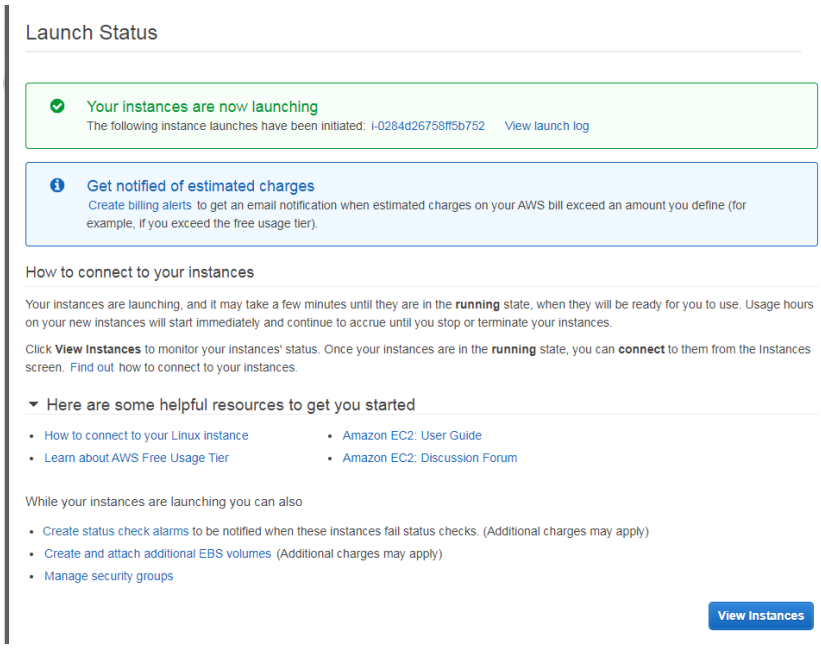
You have to choose next steps until you reach, Configuring Security Group



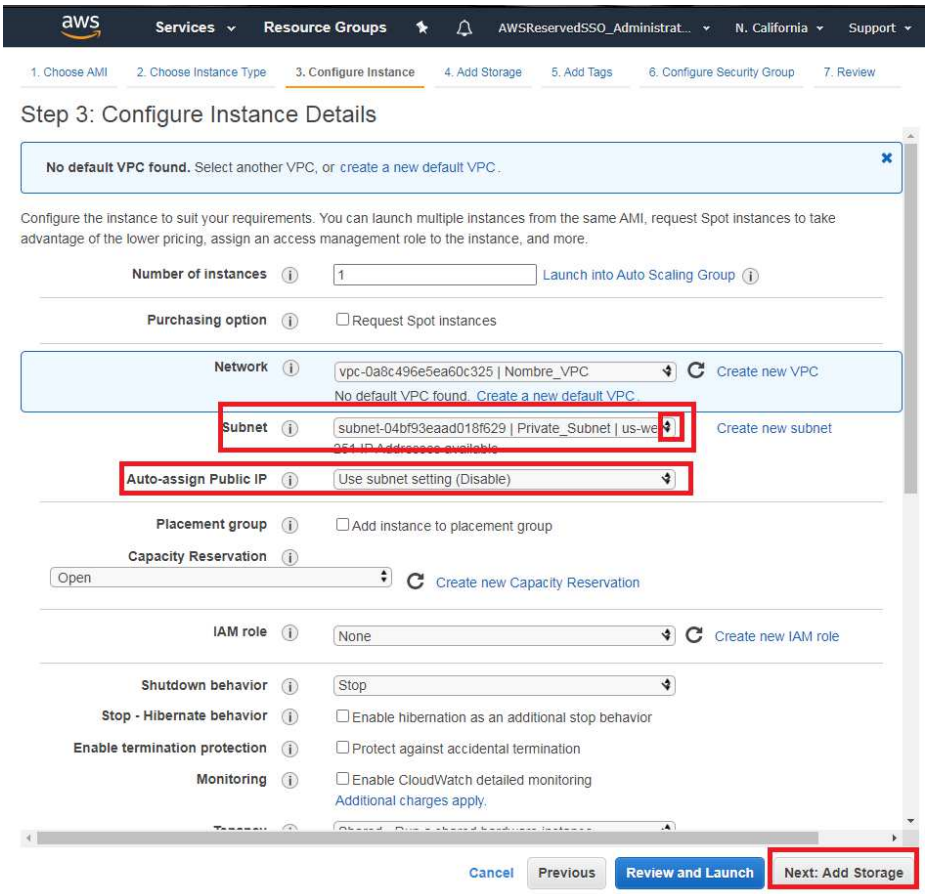
Review page, before of launching EC2 Instance.

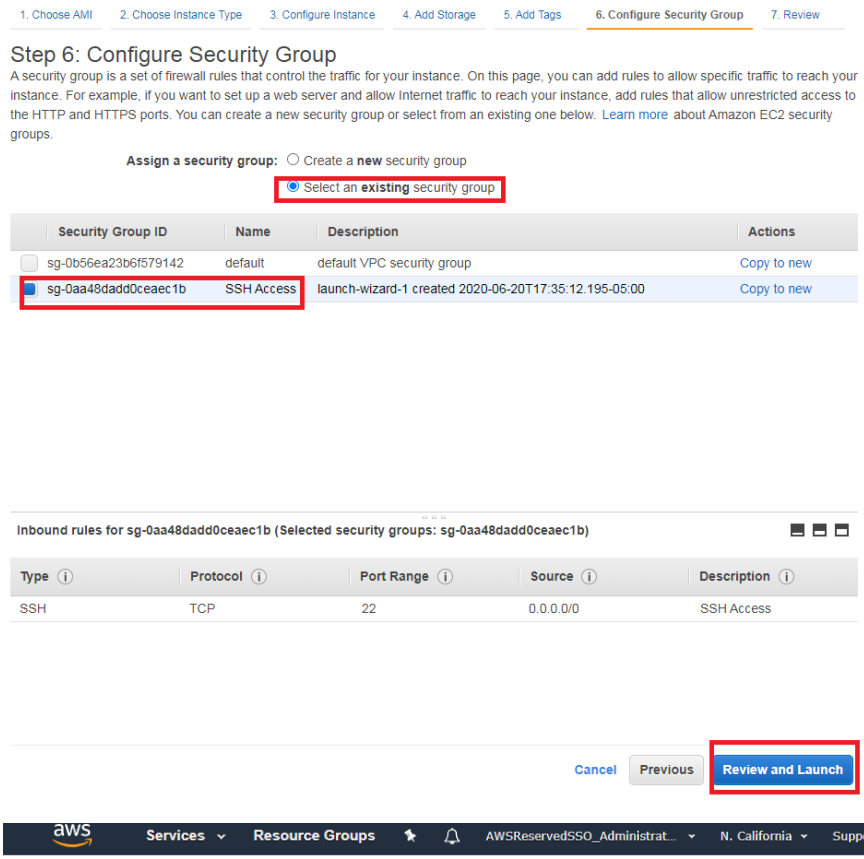


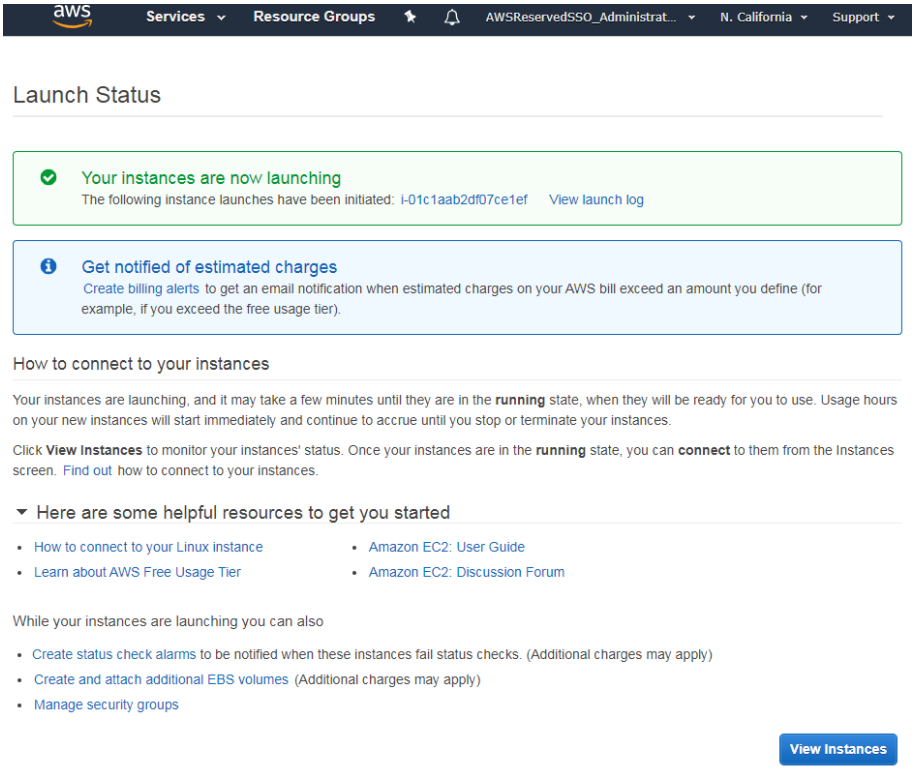




For Private Instances, the procedure is similar however, it changes on the subnet to create the instances and the security group is chosen from the previous step.


aws
Services
Resource Groups
AWSReservedSSO_Administrat...
N. California
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
No default VPC found. Select another VPC, or create a new default VPC.
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.
Number of instances
1
Launch into Auto Scaling Group
Purchasing option
Request Spot instances
Network
vpc-0a8c496e5ea60c325 | Nombre_VPC
Create new VPC
No default VPC found. Create a new default VPC
Subnet
subnet-04bf93eaad018f629 | Private_Subnet | us-we
Create new subnet
Auto-assign Public IP
Use subnet setting (Disable)
Placement group
Add instance to placement group
Capacity Reservation
Open
Create new Capacity Reservation
IAM role
None
Create new IAM role
Shutdown behavior
Stop
Stop - Hibernate behavior
Enable hibernation as an additional stop behavior
Enable termination protection
Protect against accidental termination
Monitoring
Enable CloudWatch detailed monitoring
Additional charges apply.
Cancel
Previous
Review and Launch
Next: Add Storage


1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.
Assign a security group: Create a new security group
Select an existing security group
Security Group ID
Name
Description
Actions
sg-0b56ea23b6f579142
default
default VPC security group
Copy to new
sg-0aa48dadd0ceaec1b
SSH Access
launch-wizard-1 created 2020-06-20T17:35:12.195-05:00
Copy to new
Inbound rules for sg-0aa48dadd0ceaec1b (Selected security groups: sg-0aa48dadd0ceaec1b)
Type
Protocol
Port Range
Source
Description
SSH
TCP
22
0.0.0.0/0
SSH Access
Cancel
Previous
Review and Launch


aws
Services
Resource Groups
AWSReservedSSO_Administrat...
N. California
Support
Launch Status
Your instances are now launching
The following instance launches have been initiated: i-01c1aab2df07ce1ef
View launch log
Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).
How to connect to your instances
Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.
Click View Instances to monitor your instances' status. Once your instances are in the running state, you can connect to them from the Instances screen. Find out how to connect to your instances.
Here are some helpful resources to get you started
How to connect to your Linux instance
Learn about AWS Free Usage Tier
Amazon EC2: User Guide
Amazon EC2: Discussion Forum
While your instances are launching you can also
Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
Create and attach additional EBS volumes (Additional charges may apply)
Manage security groups
View Instances

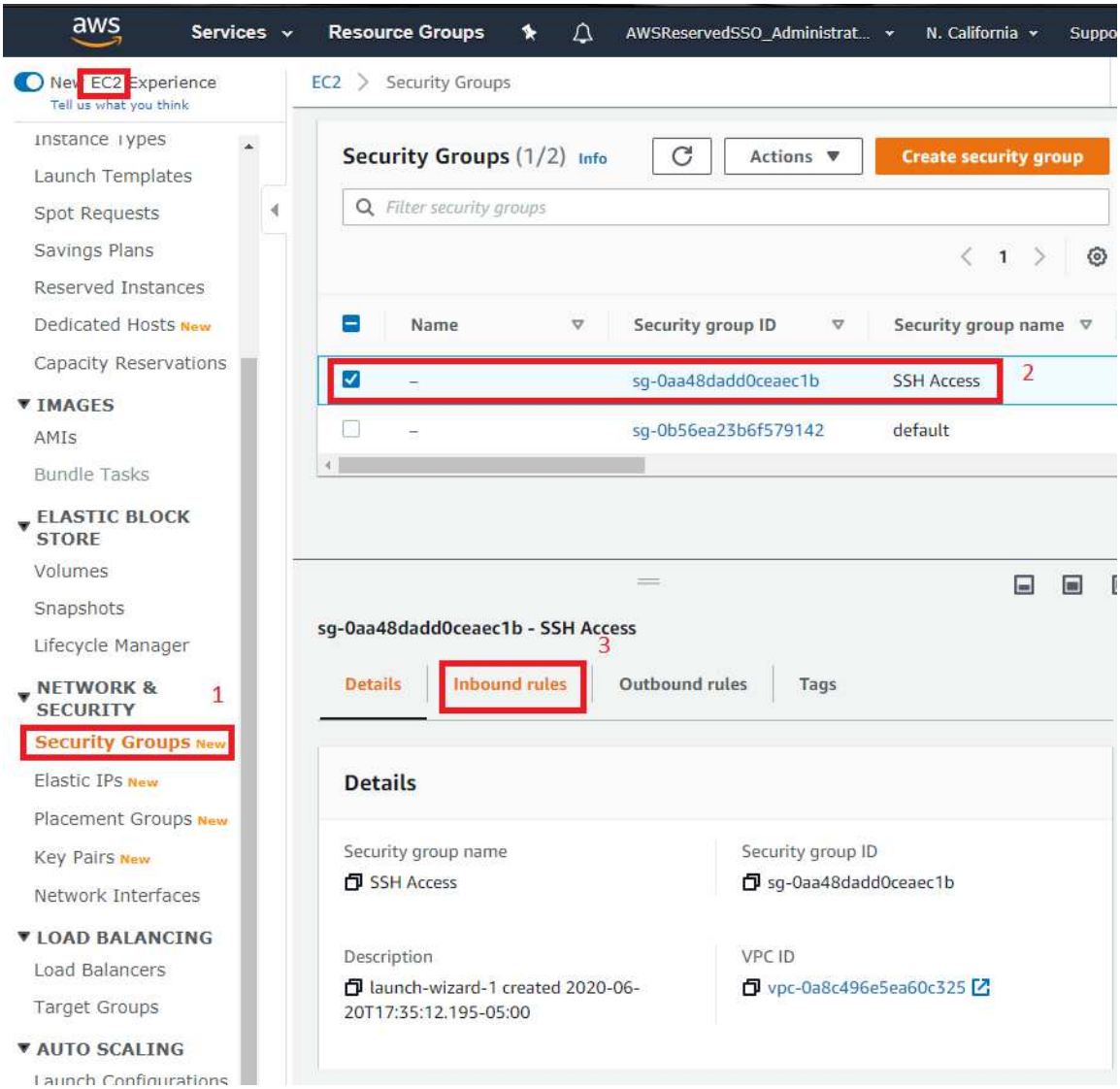
Make the review

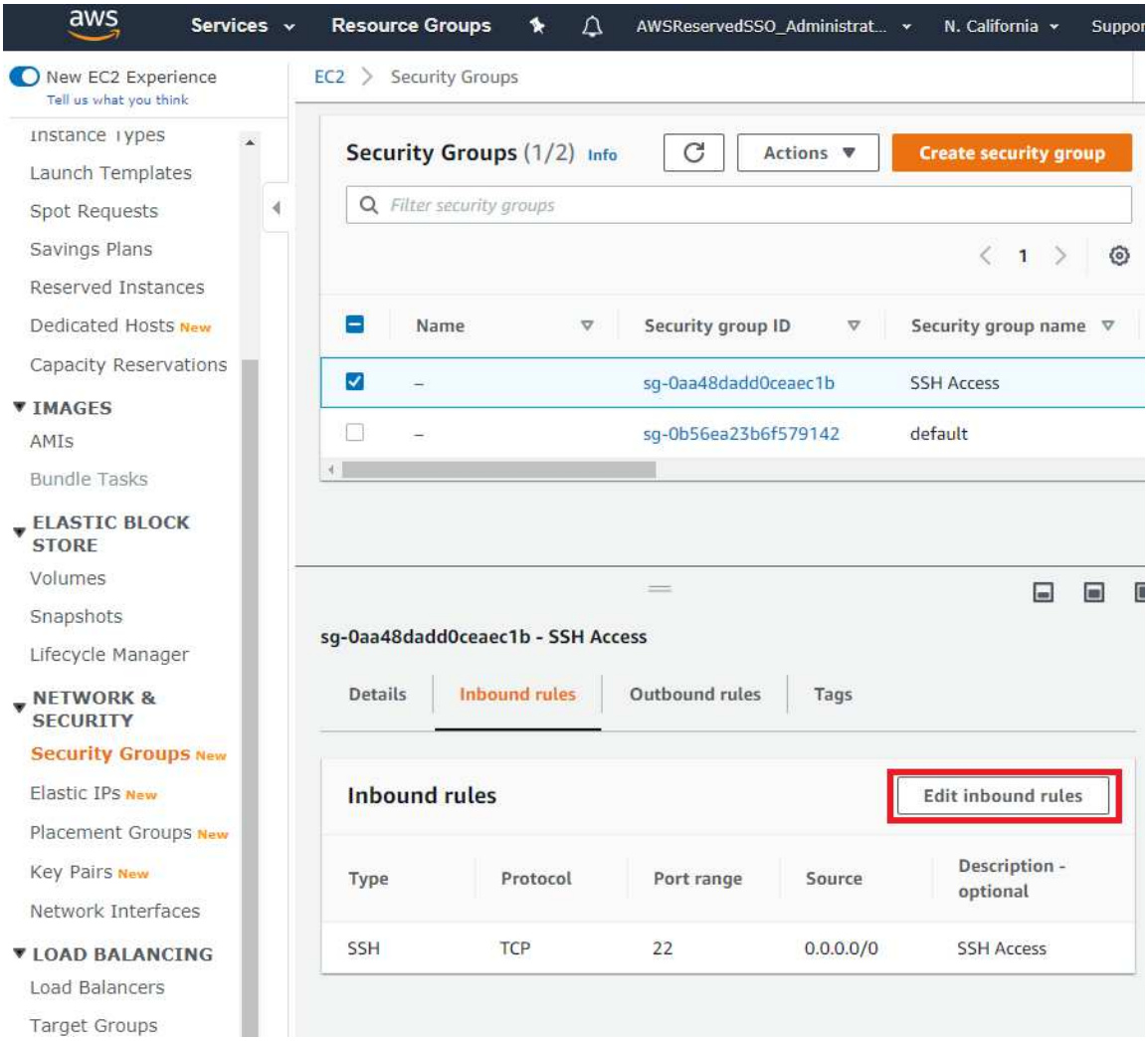
Use the same tools of Command Line (Windows): putty.

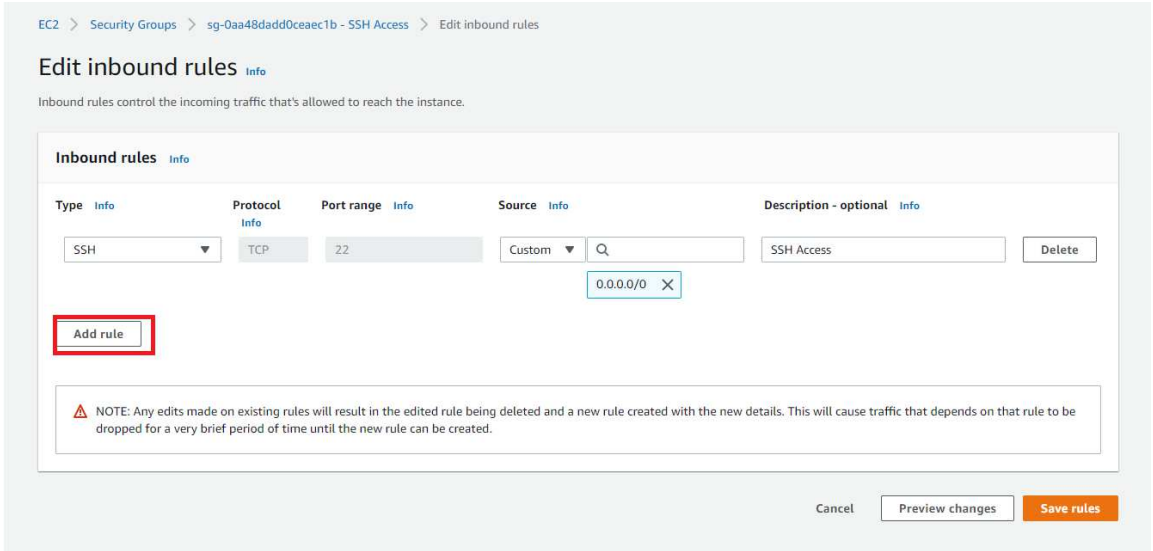
Check the “Review Configurations using Putty, SFTP and Browser” section, except the Add and Revoke Ports on Security Groups.

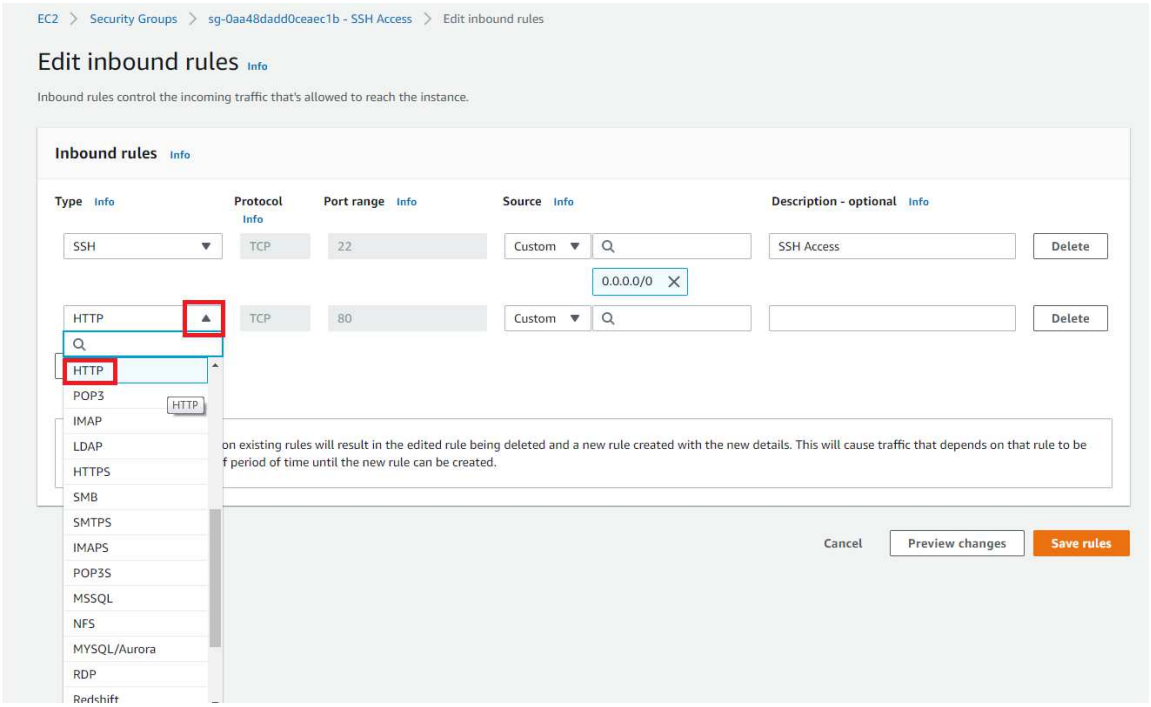
Add and Revoke Ports on Security Group

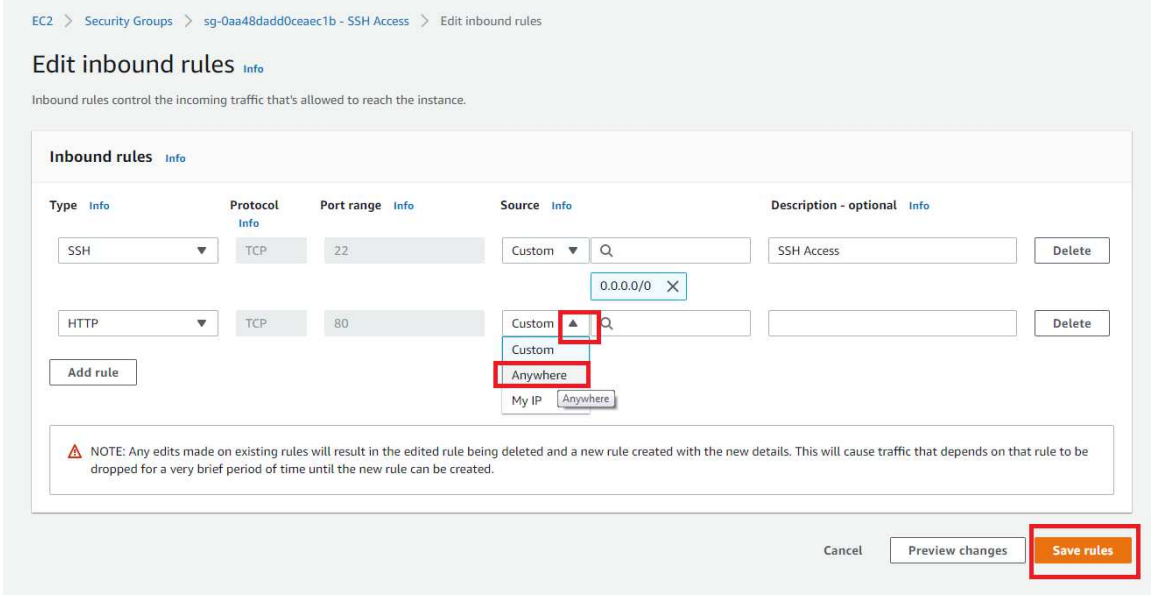
To add ports,

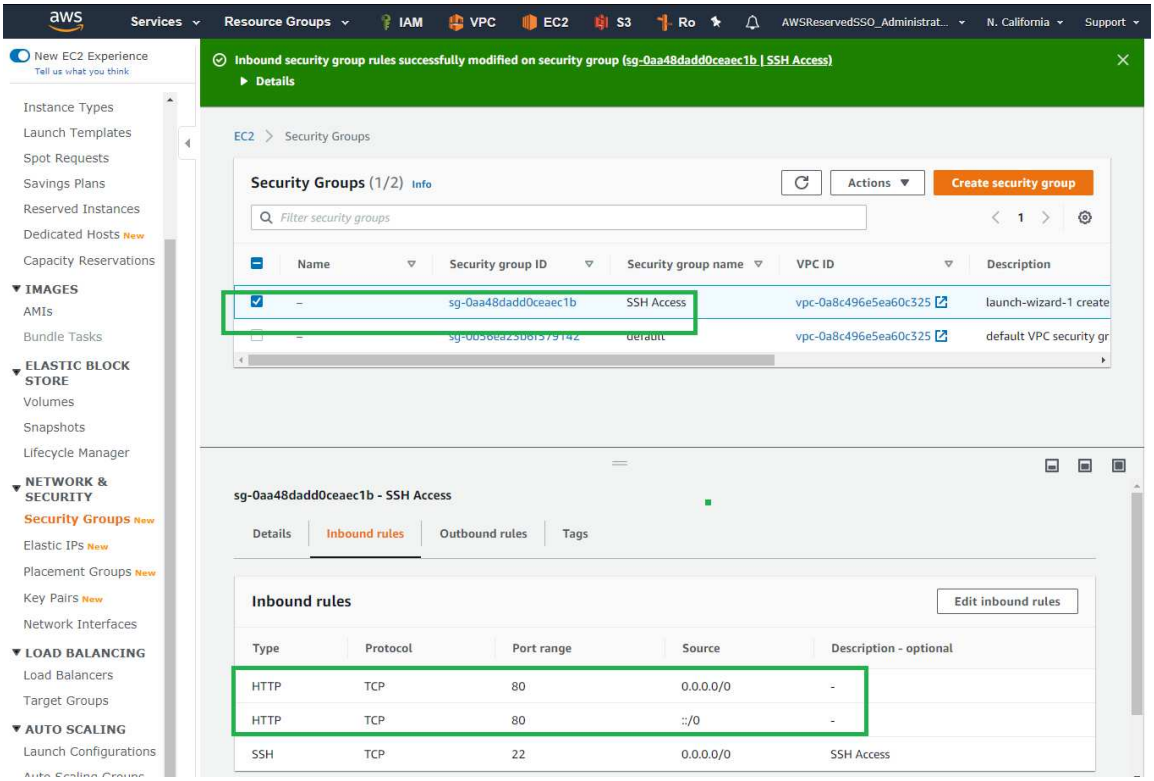



aws
Services
Resource Groups
AWSReservedSSO_Administrat...
N. California
New EC2 Experience
Tell us what you think
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups New
Elastic IPs New
Placement Groups New
Key Pairs New
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
EC2 > Security Groups
Security Groups (1/2) Info
Actions
Create security group
Filter security groups
Name
Security group ID
Security group name
sg-0aa48dadd0ceaec1b
SSH Access
sg-0b56ea23b6f579142
default
sg-0aa48dadd0ceaec1b - SSH Access
Details
Inbound rules
Outbound rules
Tags
Inbound rules
Edit inbound rules
Type
Protocol
Port range
Source
Description - optional
SSH
TCP
22
0.0.0.0/0
SSH Access

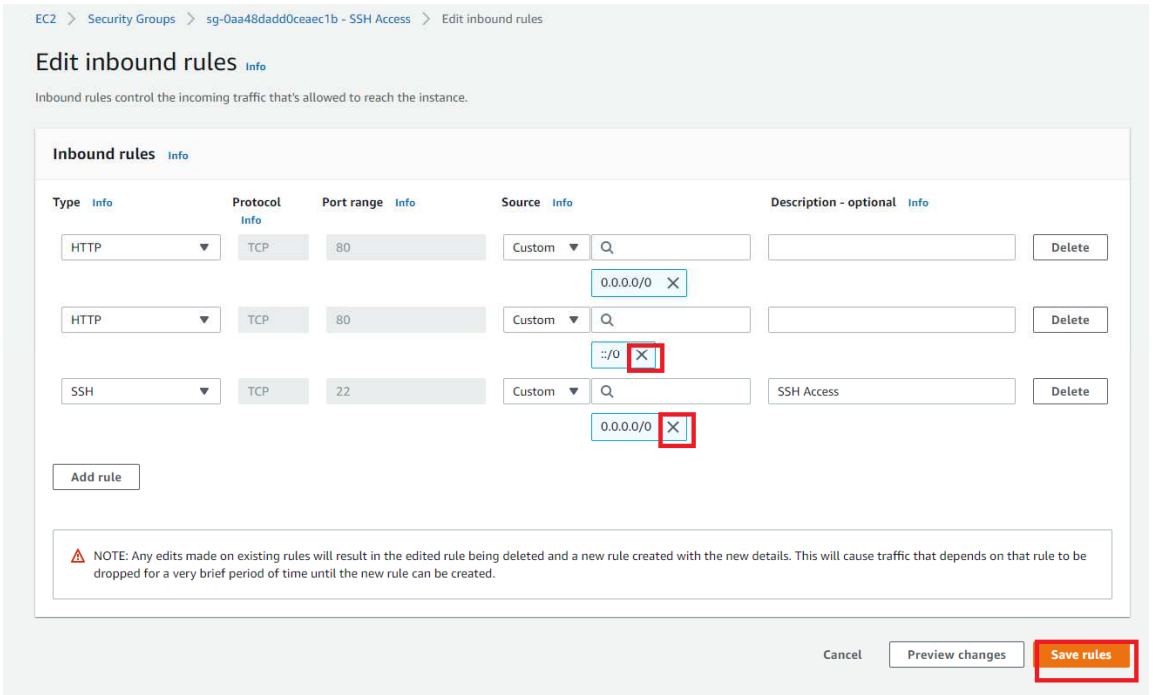

EC2 > Security Groups > sg-0aa48dadd0ceaec1b - SSH Access > Edit inbound rules
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
SSH
TCP
22
Custom
0.0.0.0/0
SSH Access
Delete
Add rule
NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.
Cancel
Preview changes
Save rules


EC2 > Security Groups > sg-0aa48dadd0ceaec1b - SSH Access > Edit inbound rules
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
SSH
TCP
22
Custom
SSH Access
Delete
0.0.0.0/0
HTTP
TCP
80
Custom
Delete
HTTP
POP3
IMAP
LDAP
HTTPS
SMB
SMTPS
IMAPS
POP3S
MSSQL
NFS
MYSQL/Aurora
RDP
Redshift
on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be
f period of time until the new rule can be created.
Cancel
Preview changes
Save rules


EC2 > Security Groups > sg-0aa48dadd0ceaec1b - SSH Access > Edit inbound rules
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
SSH
TCP
22
Custom
SSH Access
Delete
0.0.0.0/0
HTTP
TCP
80
Custom
Delete
Custom
Anywhere
My IP
Anywhere
Add rule
NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.
Cancel
Preview changes
Save rules



To revoke ports on Sec Groups, you have to edit the inbound rules:



Lab 4A using Command Line (Windows)

```
rem Setear las variables
set vpcn_Mask="10.0.0.0/16"
set pbsn1_Mask="10.0.0.0/24"
set prsn2_Mask="10.0.1.0/24"
set first_az="us-east-1a"
set second_az="us-east-1b"
```

Create VPC, Public Subnet, IGW and Route Table

```
rem Crear la VPC
aws ec2 create-vpc --cidr-block %vpcn_Mask%|jq ".Vpc.VpcId" >tmpFile
set /p vpcn_Id= < tmpFile

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 Revisar Rutas de la Tabla de Ruteo
aws ec2 describe-route-tables --route-table-id %Public_RT_Id%
```

```
C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set vpcn_Mask="10.0.0.0/16"

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set pbsn1_Mask="10.0.0.0/24"

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set prsn2_Mask="10.0.1.0/24"

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-vpc --cidr-block %vpcn_Mask%|jq ".Vpc.VpcId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p vpcn_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>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\Code\s4c1\CLI>set /p pbsn1_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-internet-gateway|jq ".InternetGateway.InternetGatewayId"  >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p IGW_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 attach-internet-gateway --vpc-id %vpcn_Id% --internet-gateway-id %IGW_Id%


C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-route-table --vpc-id %vpcn_Id%|jq ".RouteTable.RouteTableId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p Public_RT_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>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\Code\s4c1\CLI>aws ec2 describe-route-tables --route-table-id %Public_RT_Id%
{
    "RouteTables": [
        {
            "Associations": [],
            "PropagatingVgws": [],
            "RouteTableId": "rtb-00928f9319847ff6f",
            "Routes": [
                {
                    "DestinationCidrBlock": "10.0.0.0/16",
                    "GatewayId": "local",
                    "Origin": "CreateRouteTable",
                    "State": "active"
                },
                {
                    "DestinationCidrBlock": "0.0.0.0/0",
                    "GatewayId": "igw-02666aa3671e69214",
                    "Origin": "CreateRoute",
                    "State": "active"
                }
            ],
            "Tags": [],
            "VpcId": "vpc-0191cac28409315b9",
```

```
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 Permitir que las instancias que se ejecutan en la subred se hagan public
as
aws ec2 modify-subnet-attribute --subnet-id %pbsn1_Id% --map-public-ip-on-
launch
```

```
C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 associate-route-table  --subnet-id %pbsn1_Id% --route-table-id %Public_RT_Id%
{
    "AssociationId": "rtbassoc-0d333946210ba99b3",
    "AssociationState": {
        "State": "associated"
    }
}

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 modify-subnet-attribute --subnet-id %pbsn1_Id% --map-public-ip-on-launch
```

Create EC2 Keys, Sec Groups. Choose AMI and create EC2 Instances.

```
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 Lab4a --query "KeyMaterial" --
output text > Lab4a.pem
winscp.com /keygen "Lab4a.pem" /output="Lab4a.ppk"

rem Crear los Security Groups para esa instancia
aws ec2 create-security-group --group-name "SSHAccess" --
description "Security group for SSH access" --vpc-
id %vpcn_Id% |jq ".GroupId">tmpFile
set /p SSH_Sec_Group_Id= < tmpFile
aws ec2 authorize-security-group-ingress --group-id %SSH_Sec_Group_Id% --
protocol tcp --port 22 --cidr 0.0.0.0/0

rem En el laboratorio de EC2 Inicial se mostrar la importancia de buscar una
 AMI correcto.
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
aws ec2 run-instances --image-id %AMI% --count 1 --instance-type t2.micro --
key-name Lab4a --security-group-ids %SSH_Sec_Group_Id% --subnet-
id %pbsn1_Id% --tag-
specifications "ResourceType=instance,Tags=[{Key=ServerName,Value=A}]"
```

```
C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-key-pair --key-name Lab4a --query "KeyMaterial" --output text > Lab4a.pem

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>winscp.com /keygen "Lab4a.pem" /output="Lab4a.ppk"
Key saved to "Lab4a.ppk".

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-security-group --group-name "SSHAccess" --description "Security group for SSH access" --vpc-id %vpcn_Id% |jq ".GroupId">tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p SSH_Sec_Group_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 authorize-security-group-ingress --group-id %SSH_Sec_Group_Id% --protocol tcp --port 22 --cidr 0.0.0.0/0


C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>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(Image
s, &CreationDate))[:1].ImageId" --output text >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p AMI= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 run-instances --image-id %AMI% --count 1 --instance-type t2.micro --key-name Lab4a --security-group-ids %SSH_Sec_Group_Id% --subnet-id %pbsn1_Id% --tag-specifica
tions "ResourceType=instance,Tags=[{Key=ServerName,Value=A}]"
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-04e59c05167ea7bd5",
            "InstanceId": "i-02aad94a8fa32b097",
            "InstanceType": "t2.micro",
            "KeyName": "Lab4a",
            "LaunchTime": "2020-06-20T12:50:47+00:00",
```

Create Private Subnet, EIP, NAT Gateway, Private Route Table and EC2 Instance

```
rem Crear subred Privada
aws ec2 create-subnet --vpc-id %vpcn_Id% --cidr-block %prsn2_Mask% --
availability-zone %second_az%|jq ".Subnet.SubnetId" >tmpFile
set /p prsn2_Id= < tmpFile
```

```
rem Solicitar una IP Elastica para hacer el Nat Gateway
aws ec2 allocate-address --domain vpc |jq ".AllocationId" >tmpFile
set /p NAT_EIP= < tmpFile

rem Crear el NAT Gateway, asignarlo a una EIP Anterior.
aws ec2 create-nat-gateway --subnet-id  %pbsn1_Id% --allocation-
id %NAT_EIP%|jq ".NatGateway.NatGatewayId" >tmpFile
set /p NATGW_Id= < tmpFile

rem Crear tabla de ruteo para las redes privadas y asignar el NAT GW como ru
ta por defecto. Asociarla
aws ec2 create-route-table --vpc-
id %vpcn_Id%|jq ".RouteTable.RouteTableId" >tmpFile
set /p Private_RT_Id= < tmpFile
aws ec2 create-route --route-table-id %Private_RT_Id% --destination-cidr-
block 0.0.0.0/0 --nat-gateway-id %NATGW_Id%
aws ec2 associate-route-table  --subnet-id %prsn2_Id% --route-table-
id %Private_RT_Id%

rem Genera la segunda Instancia
aws ec2 run-instances --image-id %AMI% --count 1 --instance-type t2.micro --
key-name Lab4a --security-group-ids %SSH_Sec_Group_Id% --subnet-
id %prsn2_Id% --tag-
specifications "ResourceType=instance,Tags=[{Key=ServerName,Value=B}]"
```

```
C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-subnet --vpc-id %vpcn_Id% --cidr-block %prsn2_Mask% --availability-zone %second_az%|jq ".Subnet.SubnetId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p prsn2_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 allocate-address --domain vpc |jq ".AllocationId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p NAT_EIP= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 allocate-address --domain vpc |jq ".AllocationId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p NAT_EIP= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-nat-gateway --subnet-id  %pbsn1_Id% --allocation-id %NAT_EIP%|jq ".NatGateway.NatGatewayId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p NATGW_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-route-table --vpc-id %vpcn_Id%|jq ".RouteTable.RouteTableId" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>set /p Private_RT_Id= < tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 create-route --route-table-id %Private_RT_Id% --destination-cidr-block 0.0.0.0/0 --nat-gateway-id %NATGW_Id%
{
    "Return": true
}


C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 associate-route-table  --subnet-id %prsn2_Id% --route-table-id %Private_RT_Id%
{
    "AssociationId": "rtbassoc-014af5ad82f89a966",
    "AssociationState": {
        "State": "associated"
    }
}


C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 run-instances --image-id %AMI% --count 1 --instance-type t2.micro --key-name Lab4a --security-group-ids %SSH_Sec_Group_Id% --subnet-id %prsn2_Id% --tag-specifica
tions "ResourceType=instance,Tags=[{Key=ServerName,Value=B}]"
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-04e59c05167ea7bd5",
            "InstanceId": "i-01d086a884833e5d1",
            "InstanceType": "t2.micro",
            "KeyName": "Lab4a",
            "LaunchTime": "2020-06-20T12:52:21+00:00",
```

Get Information about Instances

```
rem Traer estados de la Instancias
aws ec2 describe-
instances | jq "[.Reservations | .[] | .Instances | .[] | {InstanceId: .Inst
anceId, State: .State.Name, SubnetId: .SubnetId, VpcId: .VpcId, Name: (.Tags
[]), PrivateIpAddress: .PrivateIpAddress, PublicIpAddress: .PublicIpAddress}
]"
```


```
rem Traer Datos especificos de instancia A. Revisar contenido de Read_A.jq
aws ec2 describe-instances | jq -f Read_A.jq
aws ec2 describe-instances | jq -
f Read_A.jq|jq ".[].PublicIpAddress" >tmpFile
set /p A_IP= < tmpFile
```

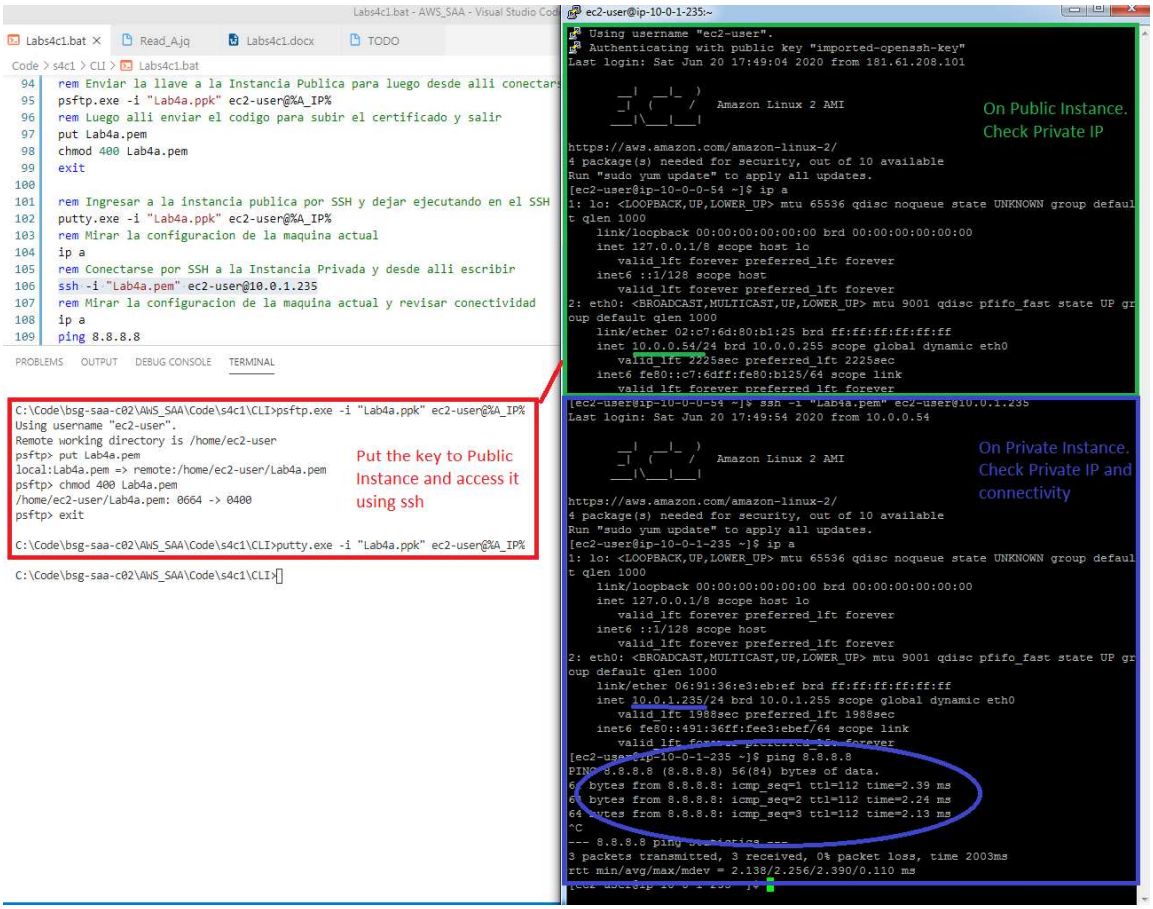
```
C:\Code\bsg-saa-c02\AWS_SAA\Code\s4c1\CLI>aws ec2 describe-instan
[]), PrivateIpAddress: .PrivateIpAddress, PublicIpAddress: .Publi
[
  {
    "InstanceId": "i-01d086a884833e5d1",
    "State": "running",
    "SubnetId": "subnet-0292b94f8f6653117",
    "VpcId": "vpc-0191cac28409315b9",
    "Name": {
      "Key": "ServerName",
      "Value": "B"
    },
    "PrivateIpAddress": "10.0.1.235",
    "PublicIpAddress": null
  },
  {
    "InstanceId": "i-02aad94a8fa32b097",
    "State": "running",
    "SubnetId": "subnet-0de359c860ccc3f11",
    "VpcId": "vpc-0191cac28409315b9",
    "Name": {
      "Key": "ServerName",
      "Value": "A"
    },
    "PrivateIpAddress": "10.0.0.54",
    "PublicIpAddress": "54.151.26.21"
  }
]
```

Review Configurations using Putty, SFTP and Browser

```
rem Enviar la llave a la Instancia Publica para luego desde alli conectarse
a la IP Privada
rem Aquí la IP A es la IP de la Instancia Publica
psftp.exe -i "Lab4a.ppk" ec2-user@%A_IP%
rem Luego alli enviar el codigo para subir el certificado y salir
put Lab4a.pem
chmod 400 Lab4a.pem
exit

rem Ingresar a la instancia publica por SSH y dejar ejecutando en el SSH  "s
udo python -m SimpleHTTPServer 80"
putty.exe -i "Lab4a.ppk" ec2-user@%A_IP%
rem Mirar la configuracion de la maquina actual
ip a
rem Conectarse por SSH a la Instancia Privada y desde alli escribir
rem Aquí la Ip mencionada es la IP de la instancia privada
```

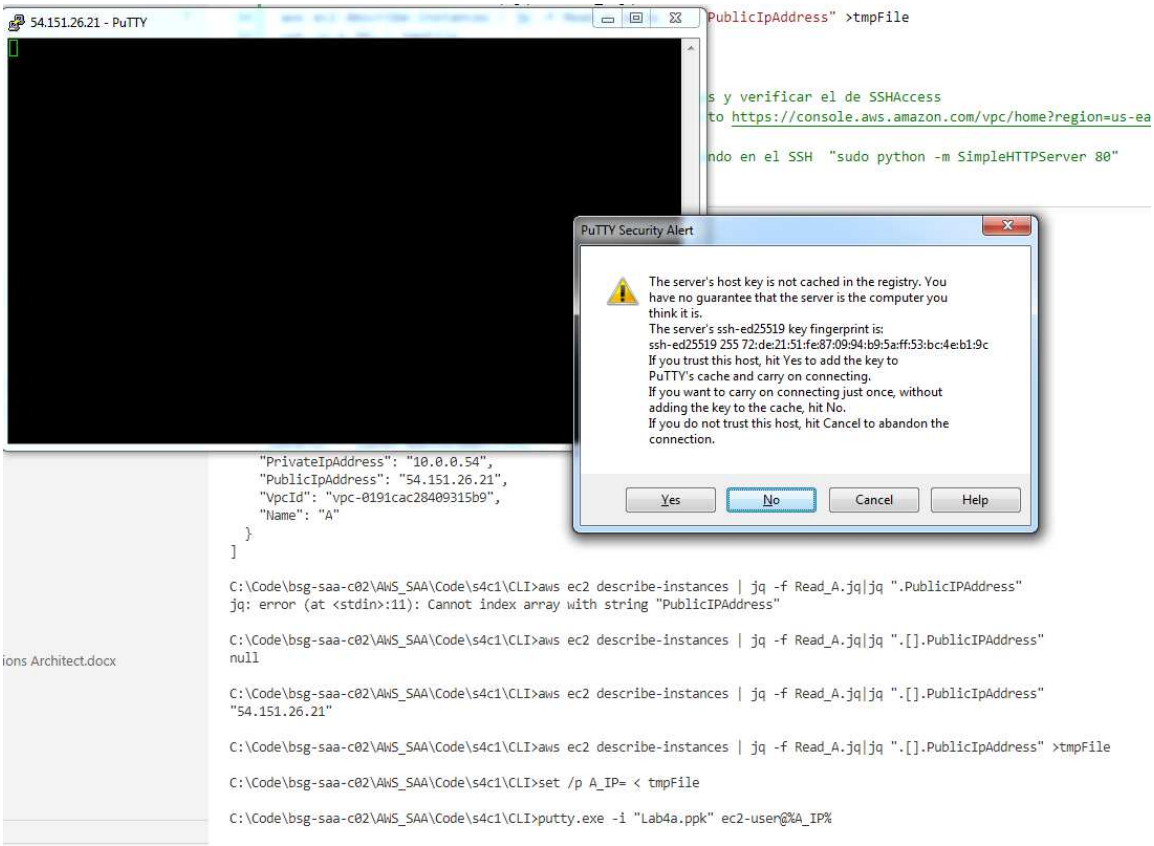
```
ssh -i "Lab4a.pem" ec2-user@10.0.1.235
rem Mirar la configuracion de la maquina actual y revisar conectividad
ip a
ping 8.8.8.8
exit
```



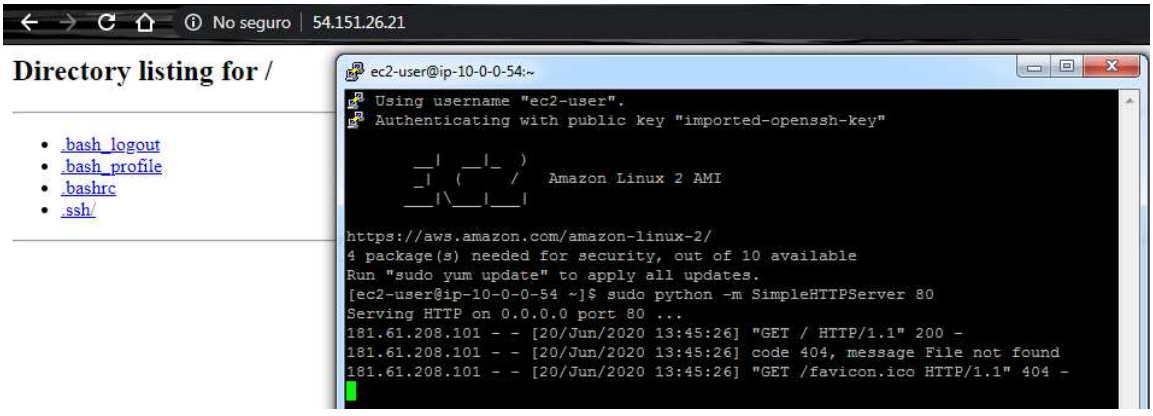

```
rem Dentro de la instancia ejecutar
sudo python -m SimpleHTTPServer 80
```

Add Port to Security Group

```
rem Intentar ingresar por un navegador a esa IP Publica
rem Modificar el Security Group para habilitar el puerto 80
aws ec2 authorize-security-group-ingress --group-id %SSH_Sec_Group_Id% --
protocol tcp --port 80 --cidr 0.0.0.0/0
```




```
rem Intentar ingresar por un navegador a esa IP Publica
```



Delete Port to Security Group

```
rem Eliminar el ingreso del Security Group anterior
```

```
aws ec2 revoke-security-group-ingress --group-id %SSH_Sec_Group_Id% --
protocol tcp --port 80 --cidr 0.0.0.0/0
rem Volver a intentar ingresar por un navegador a esa IP
```

Clean resources

For Web Management Console

EC2: Terminate Instances

EC2: Security Groups

EC2: KeyPairs

VPC: NAT Gateway

VPC: EIP (Release)

VPC: IGW (Detach and then Delete)

VPC: Subnets

VPC: RT

VPC: VPC

For Command Line (Windows)

```
rem ----- ELIMINAR RECURSOS ----
aws ec2 terminate-instances --instance-ids "i-01d086a884833e5d1" "i-
02aad94a8fa32b097"
aws ec2 delete-security-group --group-id %SSH_Sec_Group_Id%
aws ec2 delete-subnet --subnet-id %prsn2_Id%
aws ec2 delete-nat-gateway --nat-gateway-id %NATGW_Id%
aws ec2 delete-route-table --route-table-id %Private_RT_Id%
aws ec2 release-address --allocation-id %NAT_EIP%
aws ec2 delete-subnet --subnet-id %pbsn1_Id%
aws ec2 delete-route-table --route-table-id %Public_RT_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-vpc --vpc-id %vpcn_Id%
```

Evidences to send

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

1. The first screenshot of Review Configurations using Putty, SFTP and Browser. Showing SSH connection from Public Instance to Private Instances, and both different IPs.
2. The last screenshot of Review Configurations using Putty, SFTP and Browser. Showing the browser with list of users and pythons script running.