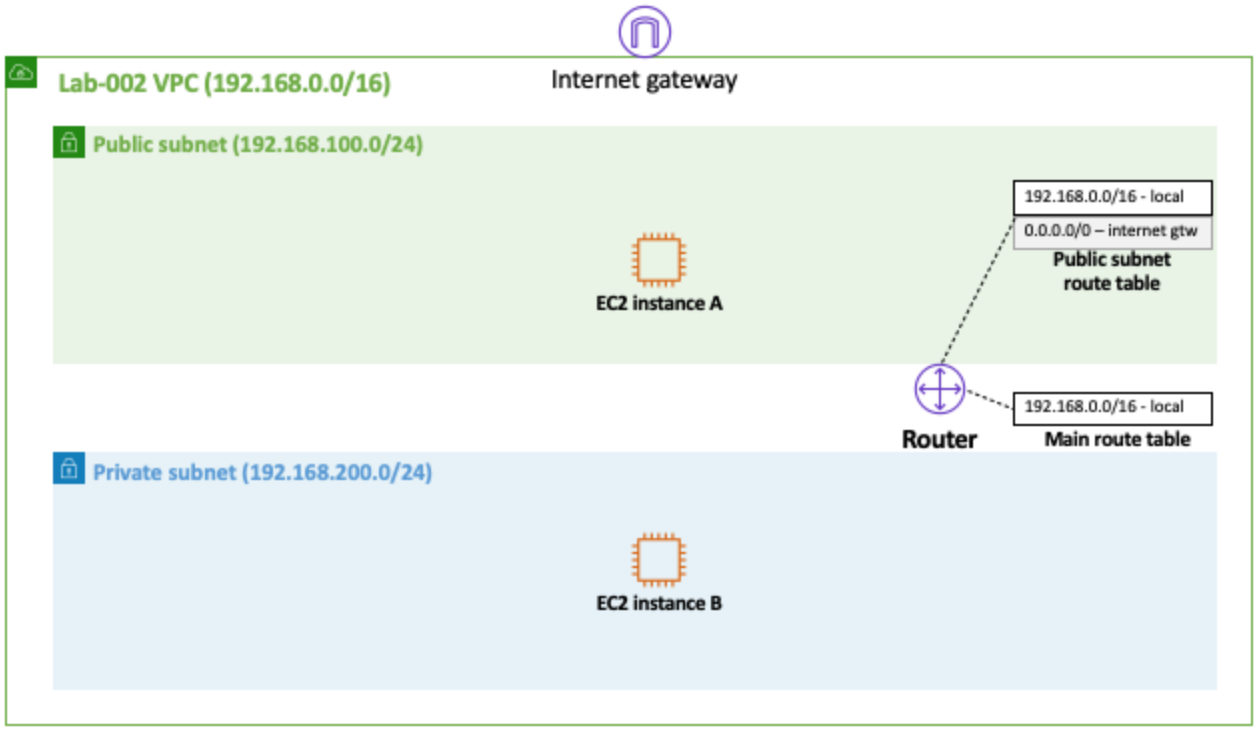
Lab : A Single EC2 Instance in a Private Subnet + Bastion Host +

Lab Overview

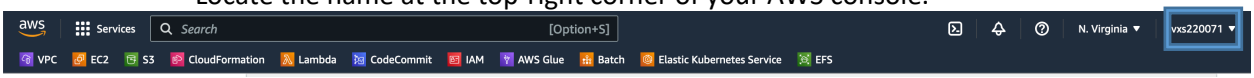
This lab has 2 part first part creating EC2 and using Baiston Host and then we are going to use same EC2 with S3 Gateway Endpoint.



NOTE: **You need to use your free tier account with full permission.**

Points to Remember before doing this Lab:

1. Please ensure that you attach all the screenshots labeled with "Note: This is a Deliverable" under each of them.
2. When capturing each screenshot, be certain that your AWS account name is visible. Locate the name at the top-right corner of your AWS console.

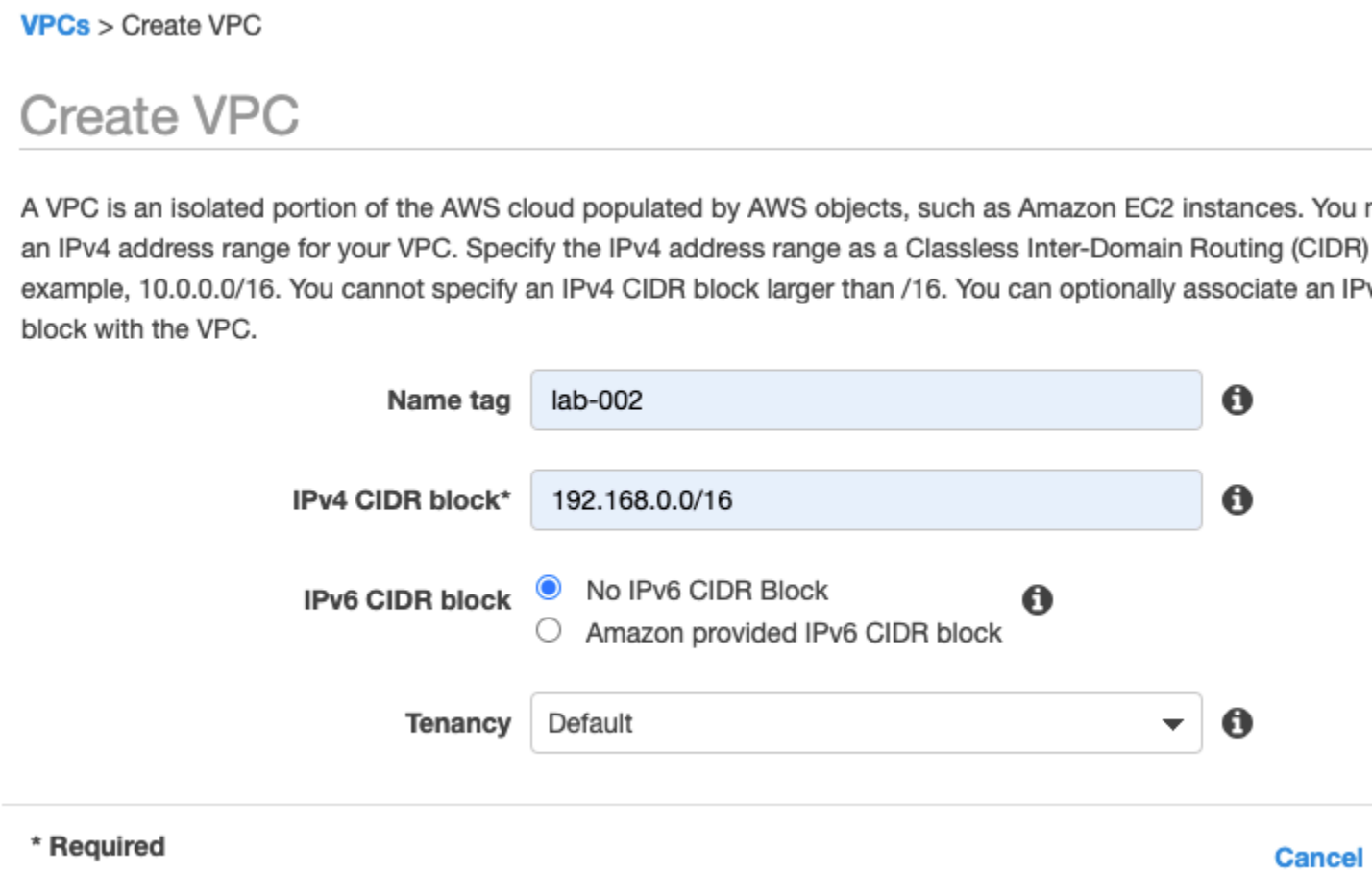


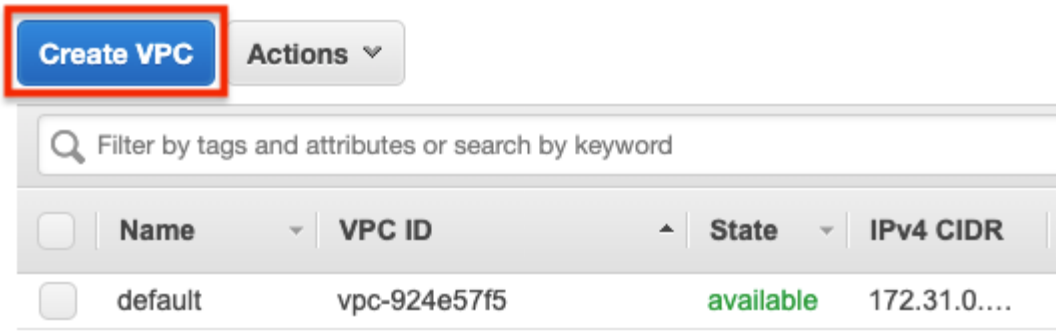
Pre-requisite for this lab:

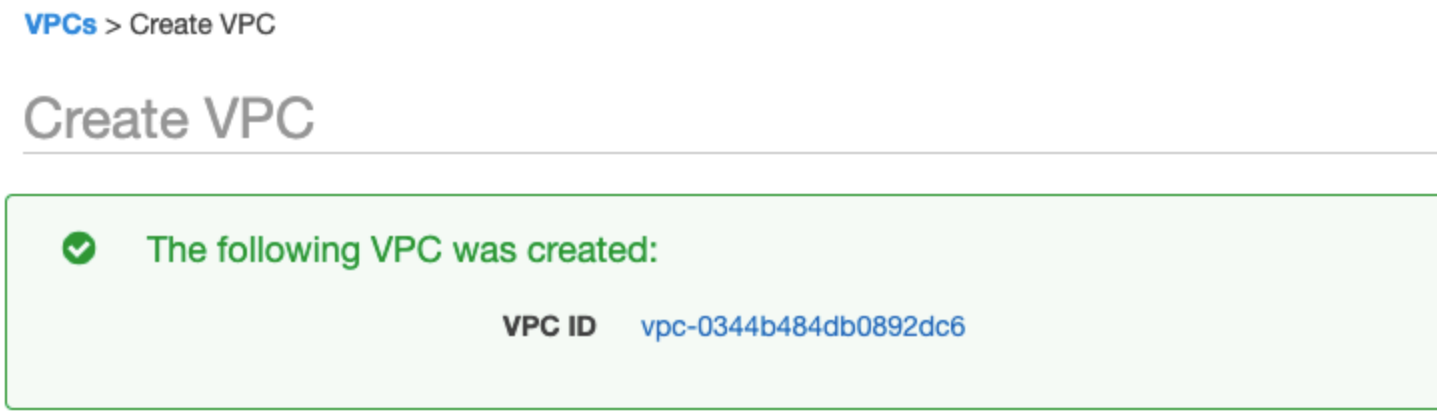
a.) Your file and folder names should start with <name>_<resource_name>.
b.) All labs must be performed in US East (N.Virginia) us-east-1 region

Step 1 - Create VPC

Create a new VPC with the *Name tag* lab-002 and the *IPv4 CIDR block* 192.168.0.0/16.







Step 2 - Create an Internet Gateway

Create an Internet gateway with the *Name tag* lab-002 and attach it to lab-002 VPC.

VPC > Internet gateways

Internet gateways (1/1) Info

Actions

Create internet ga

Filter internet gateways

1

Name	Internet gateway ID	State
–	igw-0bb0f6fd2bc46d356	Attached

VPC > Internet gateways > Create internet gateway

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway sp
for the gateway below.

Internet gateway settings

Name tag

Creates a tag with a key of 'Name' and a value that you specify.

lab-002

Cancel

Create intern

VPC > Internet gateways > igw-0c3b9e827c57c370e

igw-0c3b9e827c57c370e / lab-002

Attach
Detach
Manag
Delete

Details Info

Internet gateway ID	State	VPC ID	Owi
igw-0c3b9e827c57c370e	Detached	-	93909694

VPC > Internet gateways > Attach to VPC (igw-0c3b9e827c57c370e)

Attach to VPC (igw-0c3b9e827c57c370e) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Make sure to select the VPC with the name tag "lab-

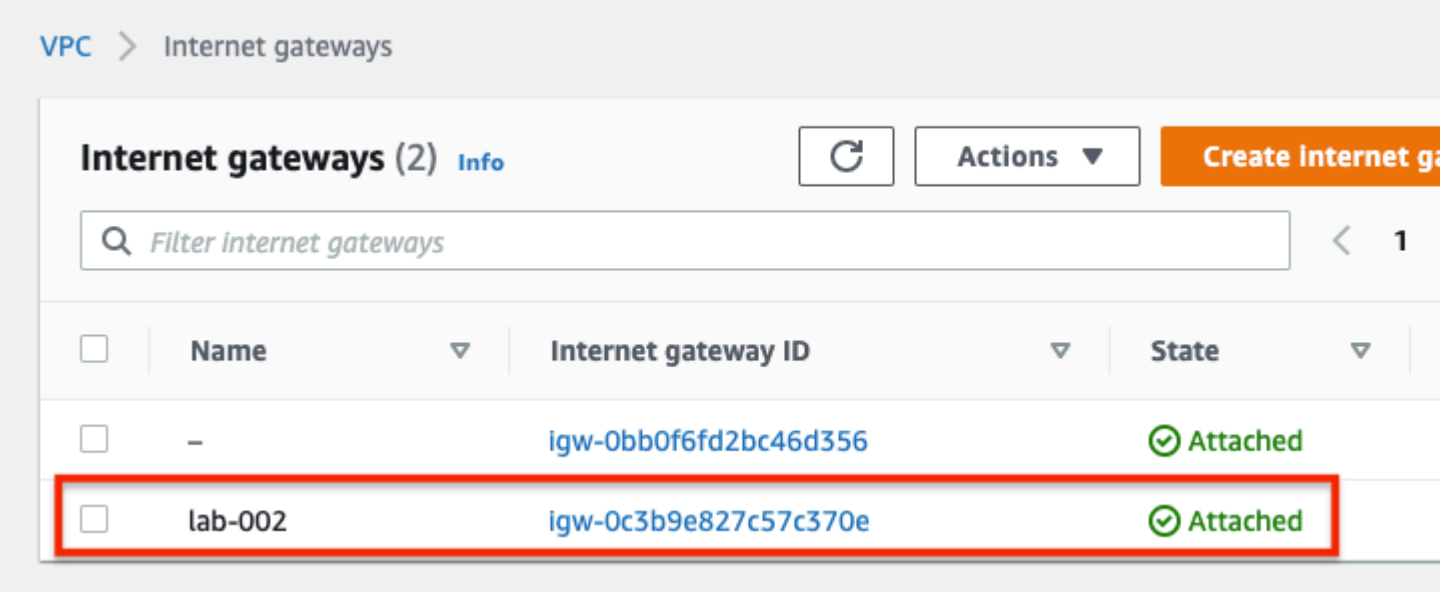
Attach the internet gateway to this VPC.

vpc-0344b484db0892dc6

▶ AWS Command Line Interface command

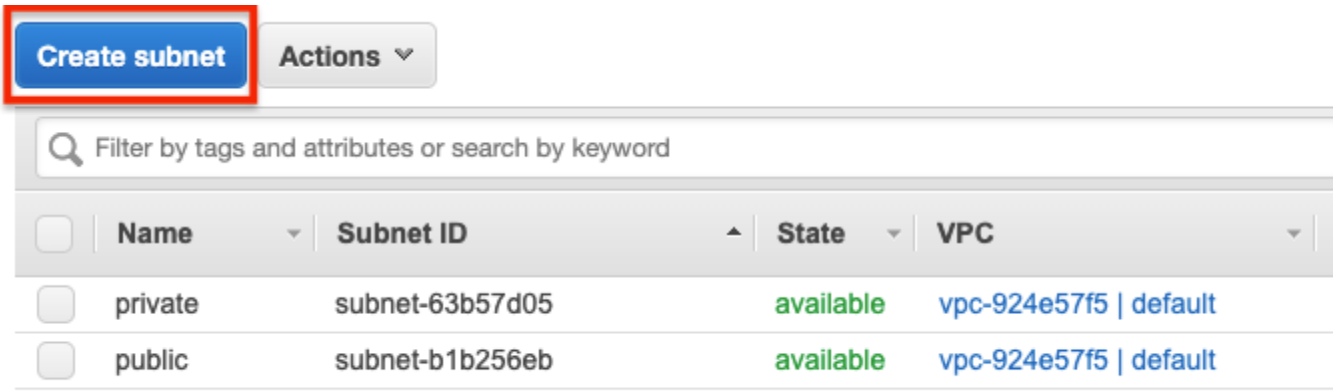
Cancel

Attach intern



Step 3 - Create the Public Subnet

Create a public subnet on lab-002 VPC with the *Name tag* public and *IPv4 CIDR block* 192.168.100.0/24.



Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be betwe
your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag public

VPC* vpc-0344b484db0892dc6

Filter by attributes

vpc-0344b484db0892dc6 lab-002

vpc-924e57f5 default

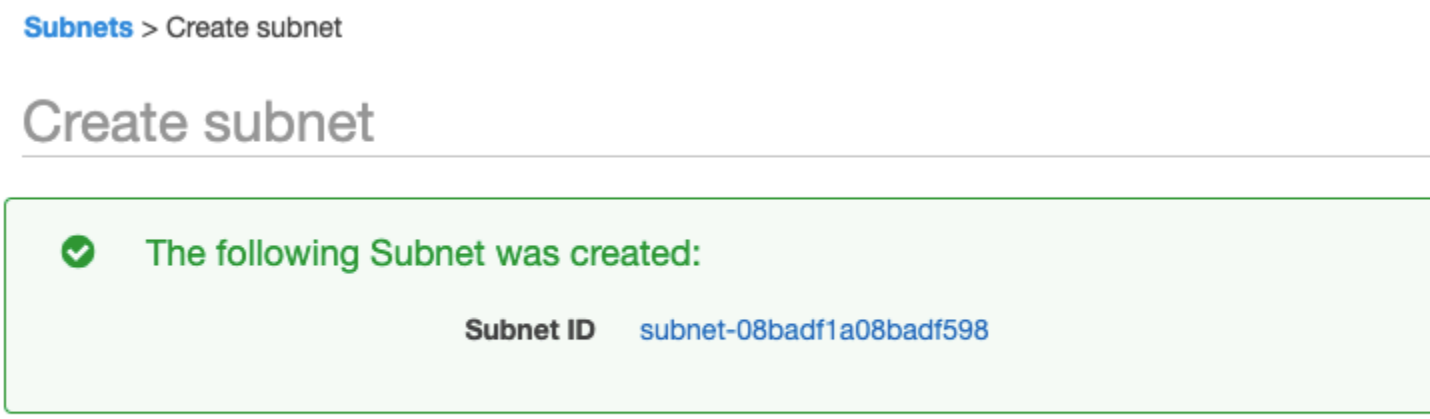
Availability Zone

VPC CIDRs

192.168.0.0/16 associated

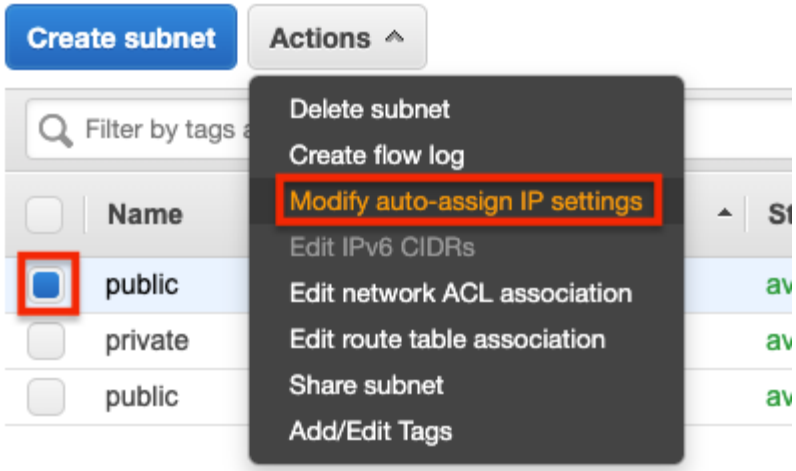
IPv4 CIDR block* 192.168.100.0/24

*** Required**



Step 4 - Enable Auto-assign IPv4

Enable *Auto-assign IPv4* in the newly created public subnet.



Step 5 - Create a Route Table

On creating the VPC, the main route table is created by default, which says **Main**=**yes.** Edit the route table and name it the **main,** as shown in the screenshots below.

Create a new route table with the *Name tag* **public** and with a default route to the internet gateway created in step 2.

Create route table | Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations
	rtb-03cb78a03f95efc35	-	-
	rtb-a31287c5	-	-

Route Table: rtb-03cb78a03f95efc35

Summary | Routes | Subnet Associations | Edge Associations | Route Propagation

Route Table ID	rtb-03cb78a03f95efc35
Explicitly Associated with	-
Owner	939096940193

Create route table | Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations
main	rtb-03cb78a03f95efc35	-	-
	rtb-a31287c5	-	-

Route Table: rtb-03cb78a03f95efc35

Summary | Routes | Subnet Associations | Edge Associations | Route Propagation

Route Table ID	rtb-03cb78a03f95efc35
Explicitly Associated with	-
Owner	939096940193

Route Tables > Create route table

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your V

Name tag public

VPC* vpc-0d8181c524284e897

Filter by attributes

vpc-924e57f5	default
vpc-0d8181c524284e897	lab-002

*** Required**

Route Tables > Create route table

Create route table

The following Route Table was created:

Route Table ID rtb-0d297ed554058cbb5

Create route table | Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations
main	rtb-03cb78a03f95efc35	-	-
public	rtb-0d297ed554058cbb5	-	-
	rtb-a31287c5	-	-

Route Table: rtb-0d297ed554058cbb5

Summary | Routes | Subnet Associations | Edge Associations | Route Propagation

Edit routes

View All routes

Destination	Target
192.168.0.0/16	local

Route Tables > Edit routes

Edit routes

Destination	Target
192.168.0.0/16	local

Add route

* Required

Route Tables > Edit routes

Edit routes

Destination	Target
192.168.0.0/16	local
0.0.0.0/0	igw-

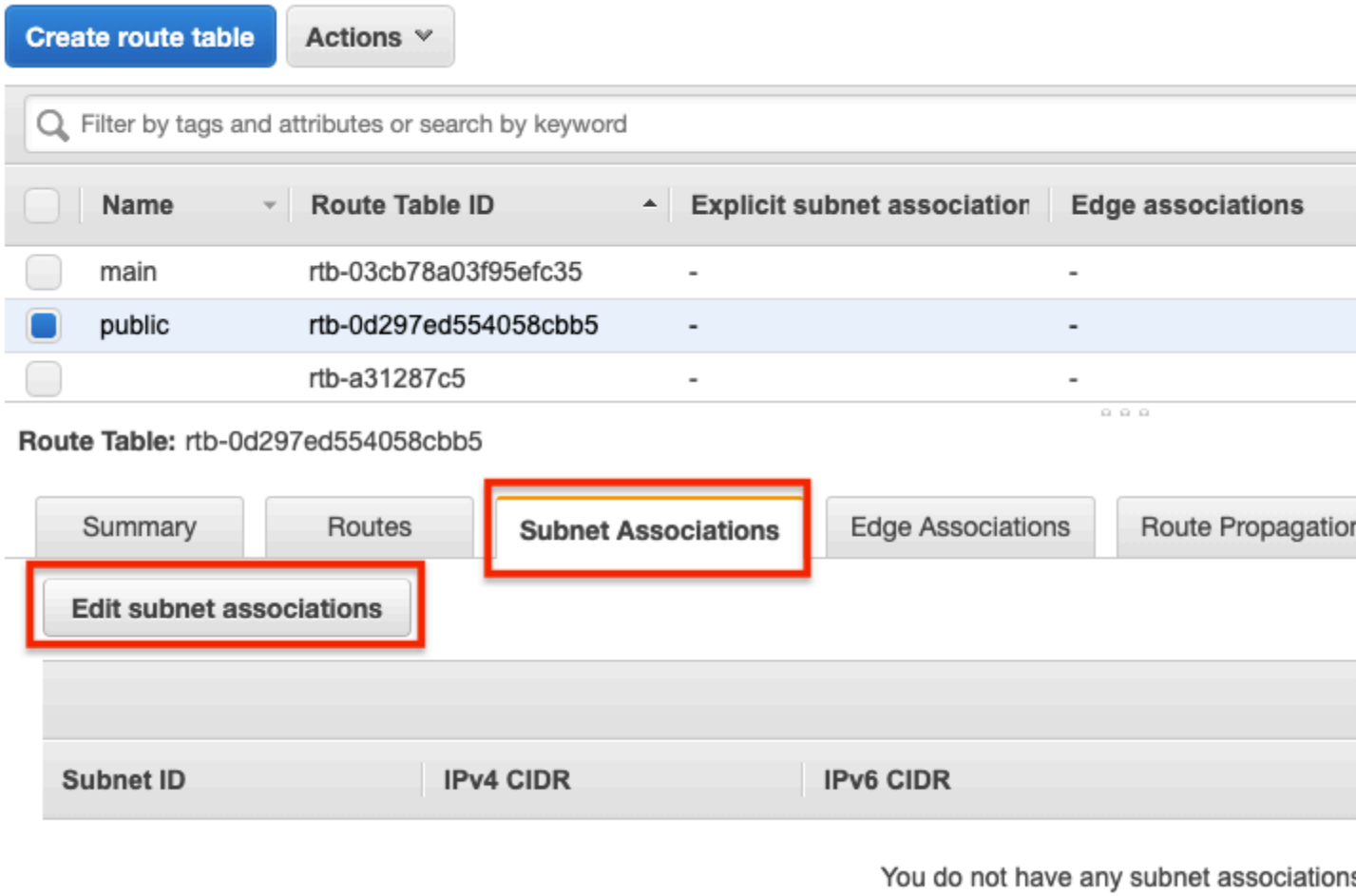
Add route

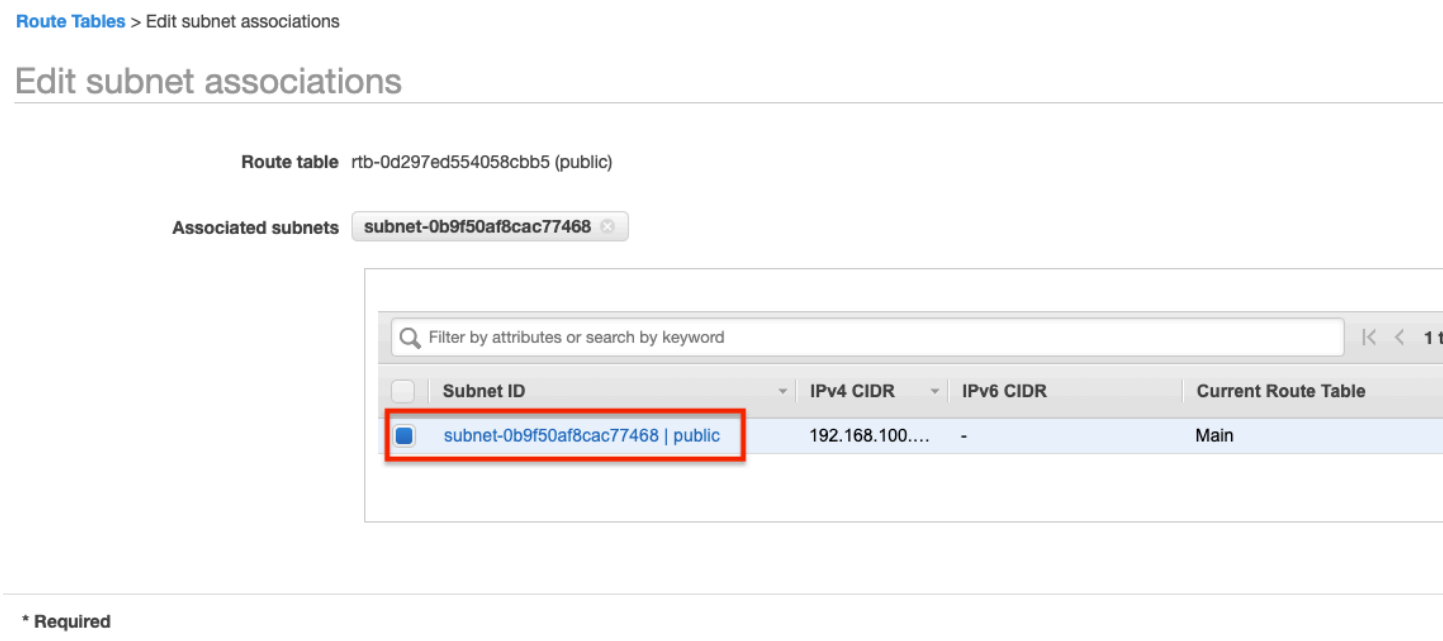
igw-0a35b250bdf127629 lab-002

* Required

Step 6 - Associate the Route Table to the Public Subnet

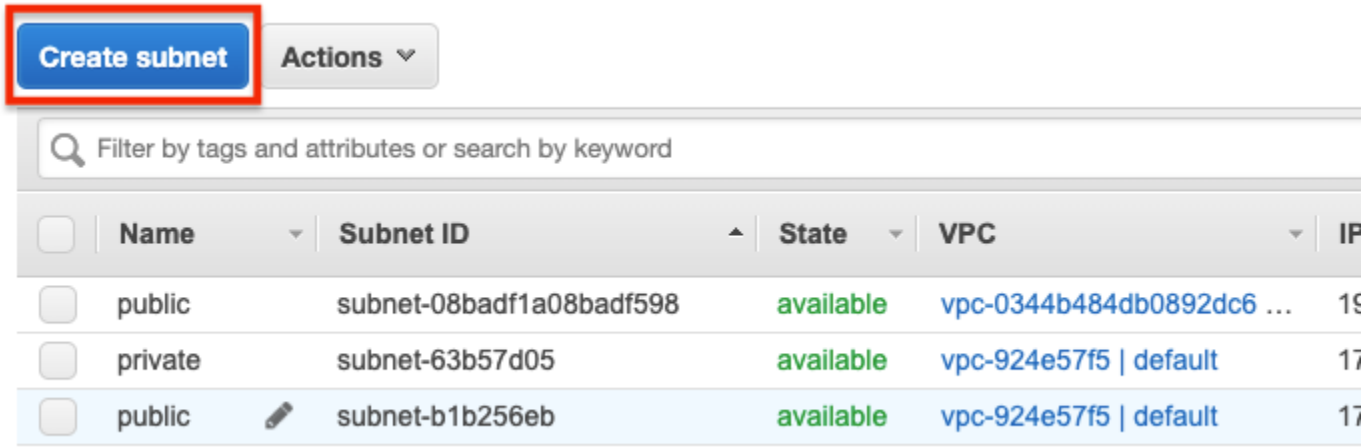
Associate the newly created route table to the public subnet.

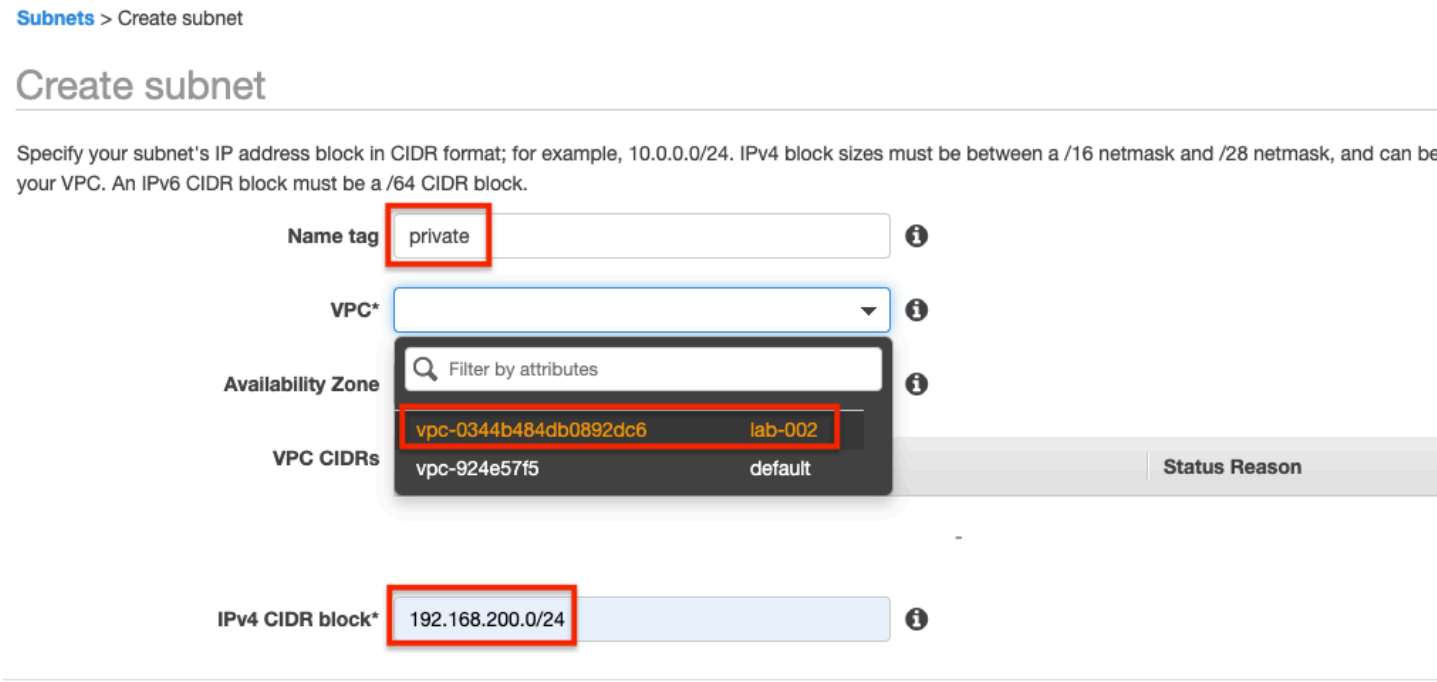


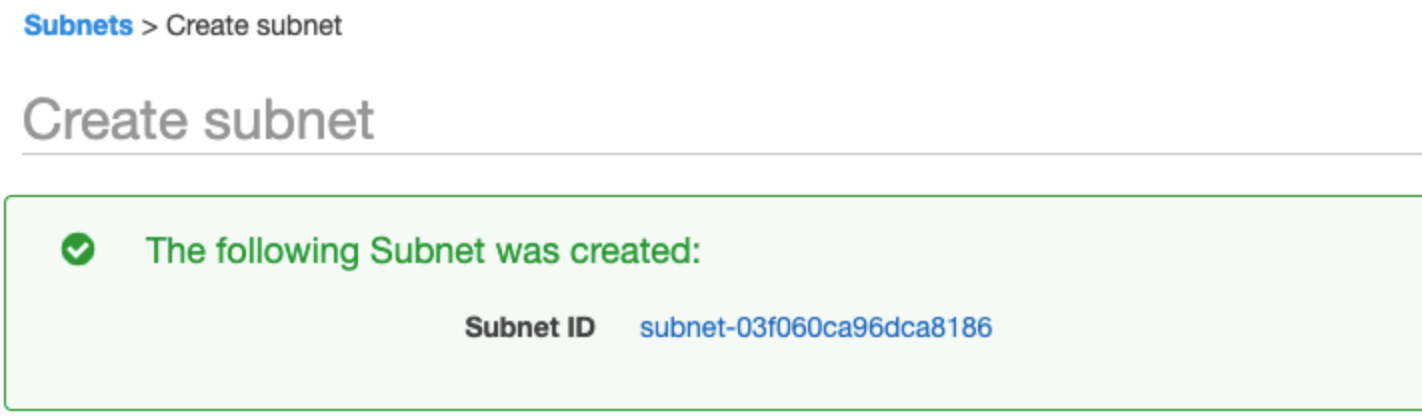


Step 7 - Create the Private Subnet

Create a private subnet on lab-002 VPC with the *Name tag* **private** and the *IPv4 CIDR block* **192.168.200.0/24**.







Step 8 - Launch the EC2 Instances

Now, Launch an EC2 instance using the public subnet. Create another EC2 in the private subnet. Note that the EC2 instance that you will create in the public subnet (labeled as A

in the diagram) is necessary so you can later connect to the EC2 instance in the private subnet (labeled as B in the diagram). EC2 A is normally called *bastion host* or *jump host*.

Lab Delivery

First add the EC2 key pair into your local ssh authentication agent using:

```
ssh-add -K lab-002.pem (macos)
ssh-add -c lab-002.pem (linux)
```

1. Then connect to the EC2 instance A but with ssh agent forwarding enabled:

```
 Provide your ->   ssh -A ec2-user@<public-IP of A>
```

Screenshot of Successful login to bastion server.

```
venkatgirisasanapuri@Venkatgiris-Air ~ % ssh -A ec2-user@54.86.232.198
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Oct 18 20:20:29 2023 from 35.146.98.70
[ec2-user@ip-192-168-100-156 ~]$
```

2. Finally, from EC2 instance A ssh to the EC2 instance B using its private IP address:

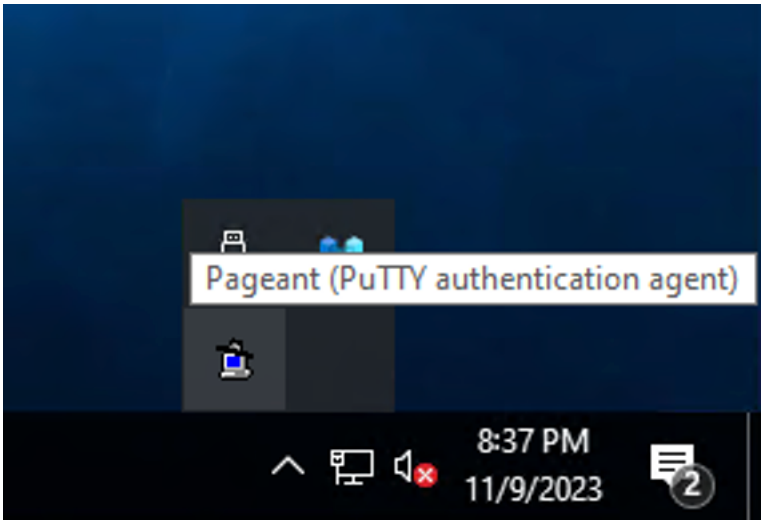
Provide -> ssh <private-IP of B> login page of your linux instance.

Below is the screenshot of Successful login to private server from bastion server. observe that you logged into the private server without using the private key(ppk or pem file). This is called as Agent Forwarding.

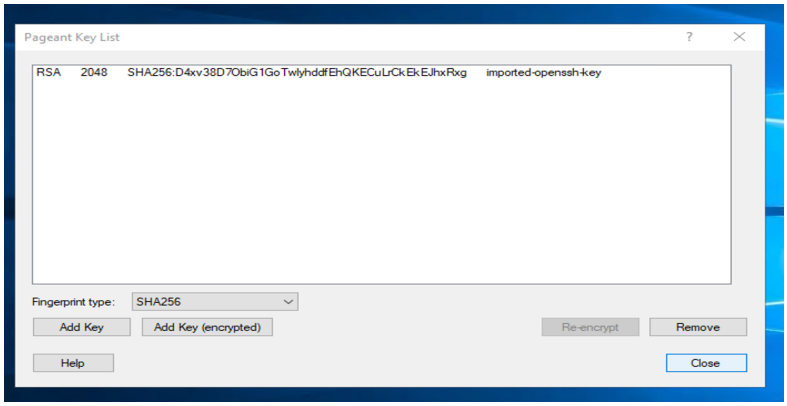
```
[ec2-user@ip-192-168-100-156 ~]$ ssh 192.168.200.79
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Oct 17 21:55:45 2023 from 192.168.100.156
[ec2-user@ip-192-168-200-79 ~]$
```

For Windows users,

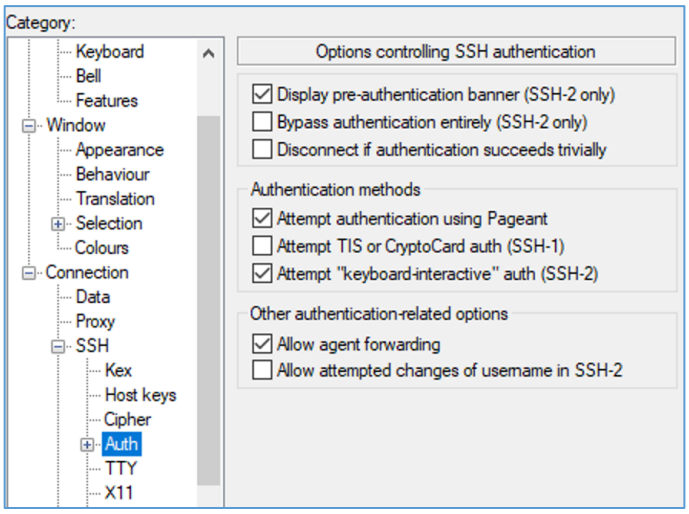
1. Open Pageant resided in the putty folder. Once you opened it, you couldn’t see the application on your taskbar. You can find it running, as shown in the below screenshot. Double-click on it.

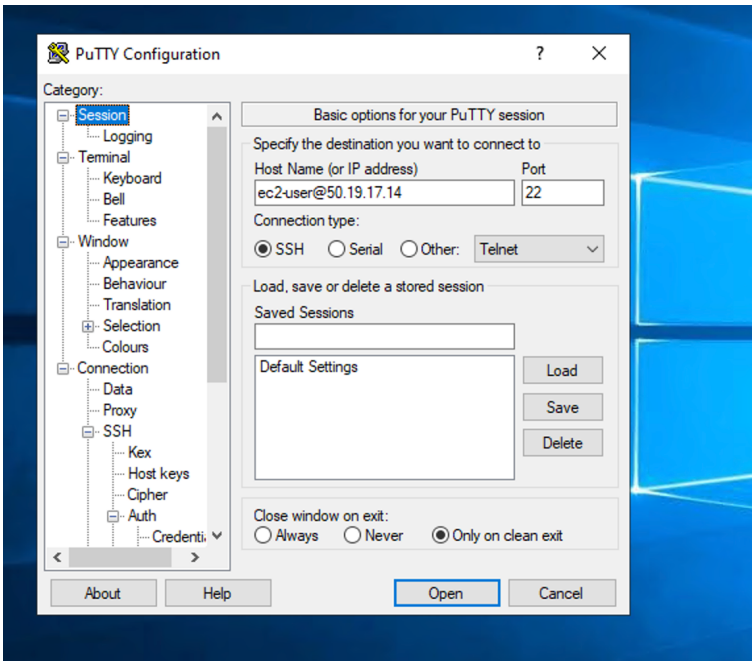


2. Once Pageant is opened, click on **Add key** and add the ppk file stored in the computer.



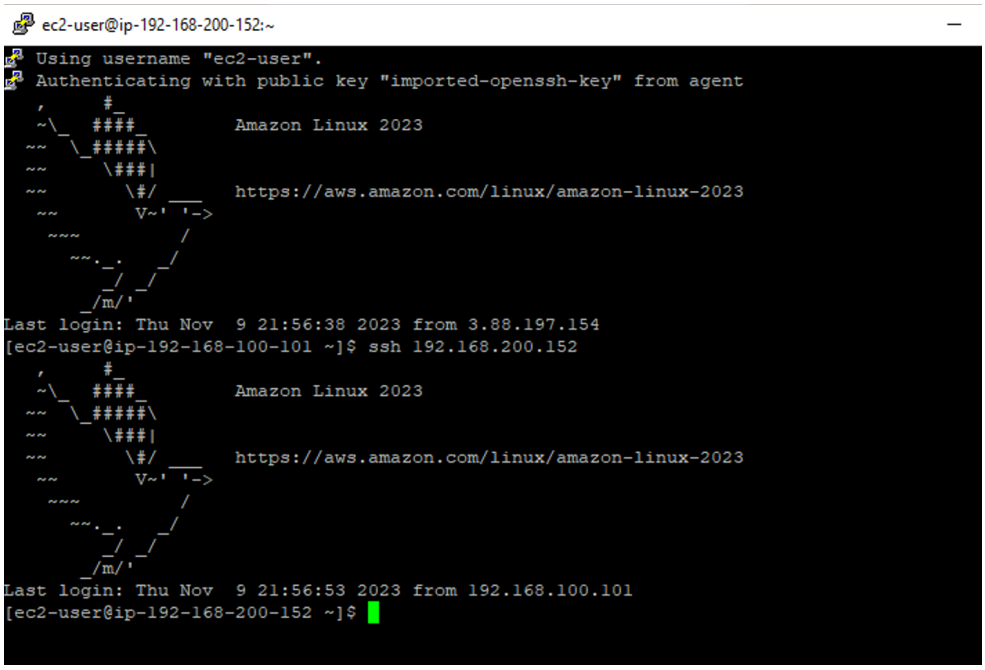
3. Once the ppk is added to the pageant, open putty and login to ec2 server A as usual but, note two points.
   a) Once the putty is opened, allow the agent-forwarding
   b) Don’t add the private file(ppk file) as you already added the key in the pageant.



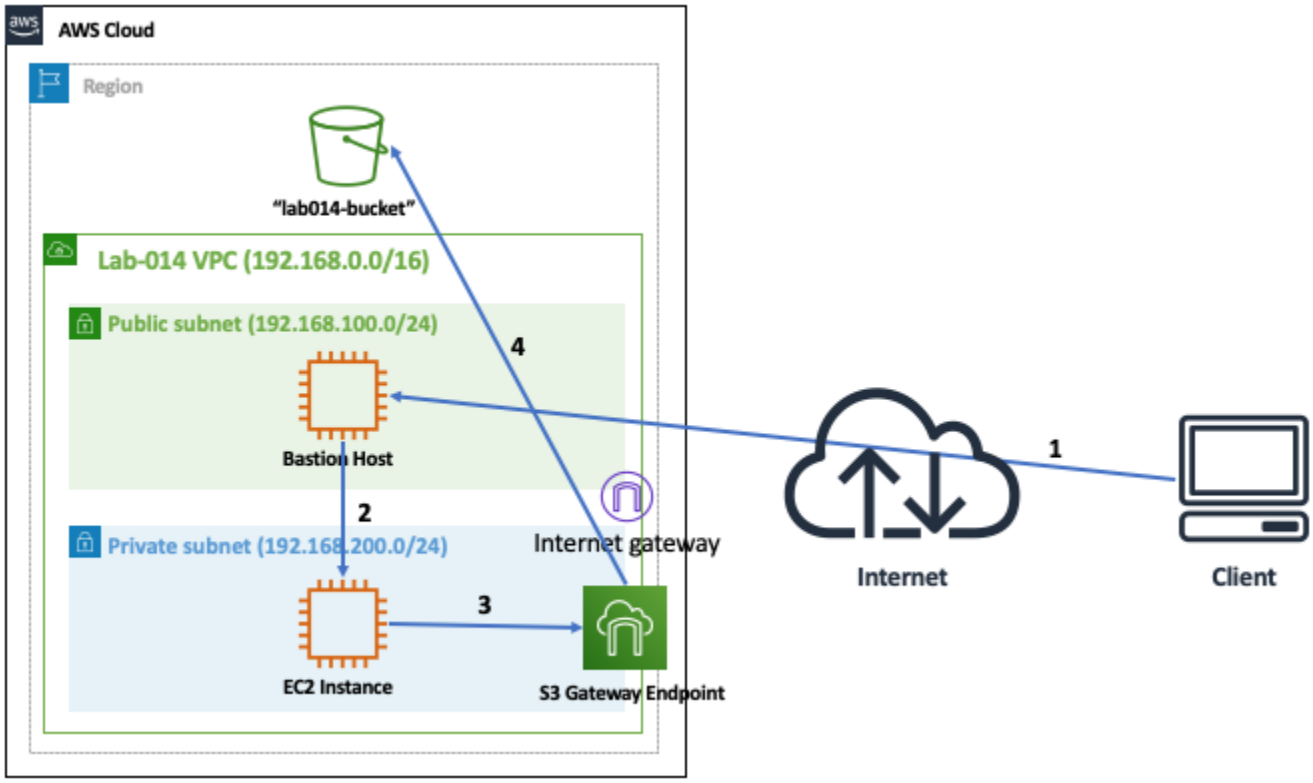


4. Once done, click on Open. You are successfully logged into the ec2-serverA.

5. Now login to the server B in the private subnet; observe that you logged into the private server without using the private key(ppk or pem file). This is called as Agent Forwarding.



PART 2: S3 Gateway Endpoint



The architecture diagram illustrates the steps we will have to take to demonstrate how the EC2 instance in the private subnet can still access an S3 bucket even if the instance does not have access to the internet.

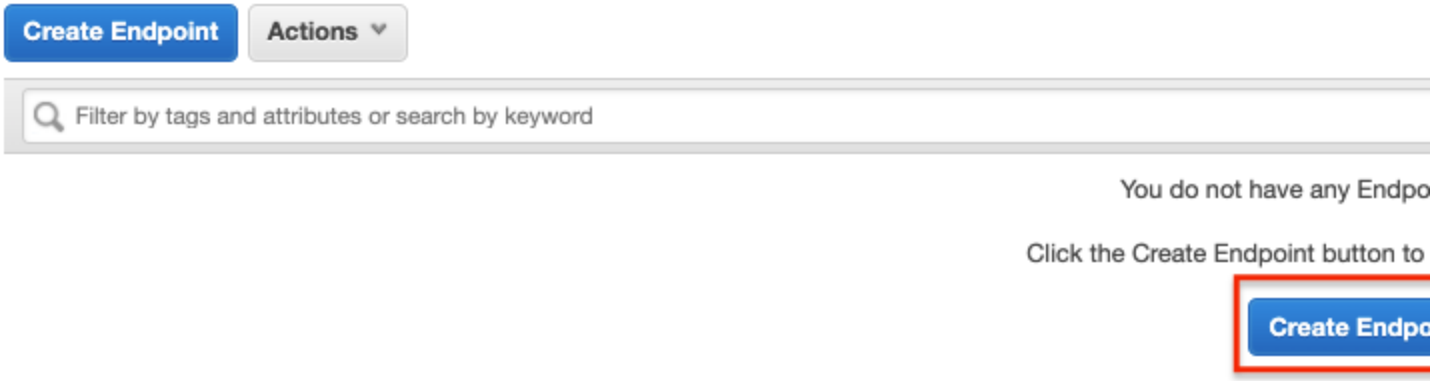
1. From the client you can connect to a bastion host configured in the public subnet
2. From the bastion host you can then connect to the EC2 instance in the private subnet,
3. From the EC2 instance you can use the S3 gateway endpoint, to ...
4. Access the S3 bucket.

Step 9 - Create an S3 Bucket

Name your S3 bucket *lab002-bucket*.

Step 10 - Create an S3 Gateway Endpoint

Go to VPC - Endpoints and click on *Create Endpoint*. **Make sure you are selecting the Main Route table (default one). Not the one you created.**



Endpoints > Create Endpoint

Create Endpoint

A VPC endpoint allows you to securely connect your VPC to another service.
An interface endpoint is powered by PrivateLink, and uses an elastic network interface (ENI) as an entry point for traffic destined to the service.
A gateway endpoint serves as a target for a route in your route table for traffic destined for the service.

Service category
- (●) AWS services
- ( ) Find service by name
- ( ) Your AWS Marketplace services

Service Name com.amazonaws.us-west-1.s3

search : s3 | Add filter

	Service Name	Owner	Type
(●)	com.amazonaws.us-west-1.s3	amazon	Gateway

VPC* vpc-080c2ae1ce3f75272

Configure route tables A rule with destination **pl-6ba54002 (com.amazonaws.us-west-1.s3)** and a target with this endpoints' ID (e.g. vpce-12
tables you select below.

Subnets associated with selected route tables will be able to access this endpoint.

rtb-03ea4195fc0e60b5d

	Route Table ID	Main	Associated With
[x]	rtb-03ea4195fc0e60b5d	Yes	2 subnets

Policy* Full Access - Allow access by any user or service within the VPC using credentials from any AWS accounts t
service. All policies — IAM user policies, VPC endpoint policies, and AWS service-specific policies (e.g. Ama
ACL policies) — must grant the necessary permissions for access to succeed.

Custom

Use the policy creation tool to generate a policy, then paste the generated policy below.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

Key (128 characters maximum) **Value** (256 characters maximum)

This resource currently has no tags

Add Tag 50 remaining (Up to 50 tags maximum)

*** Required**

Create Endpoint

The following VPC Endpoint was created:

VPC Endpoint ID vpce-06752d8599b389e04

Lab Deliveries

Total 6 screenshots

1. VPC Screenshot with Resource tab clearly visible once done with the creation of Network Infrastructure. (till endpoint)

2. Connect to the EC2 instance in the private subnet. Configure AWS CLI with an access key. (Refer AWS CLI lab for configuring the AWSCLI). Create some test files.

3. Provide screenshots of list of all your buckets using:

   ```
   aws s3 ls
   ```

4. If you want to list the objects (and folders) of your *lab014-bucket* bucket try:

   ```
   aws s3 ls s3://lab002-bucket
   ```

5. Try uploading some files using:

   ```
   aws s3 cp file1.txt s3://lab002-bucket
   ```

6. To illustrate downloading a file you can try:

   ```
   aws s3 cp s3://lab002-bucket/file1.txt file1-copied-back.txt
   ```

Once done with the lab, try to delete the ec2 instances. Remaining are optional.