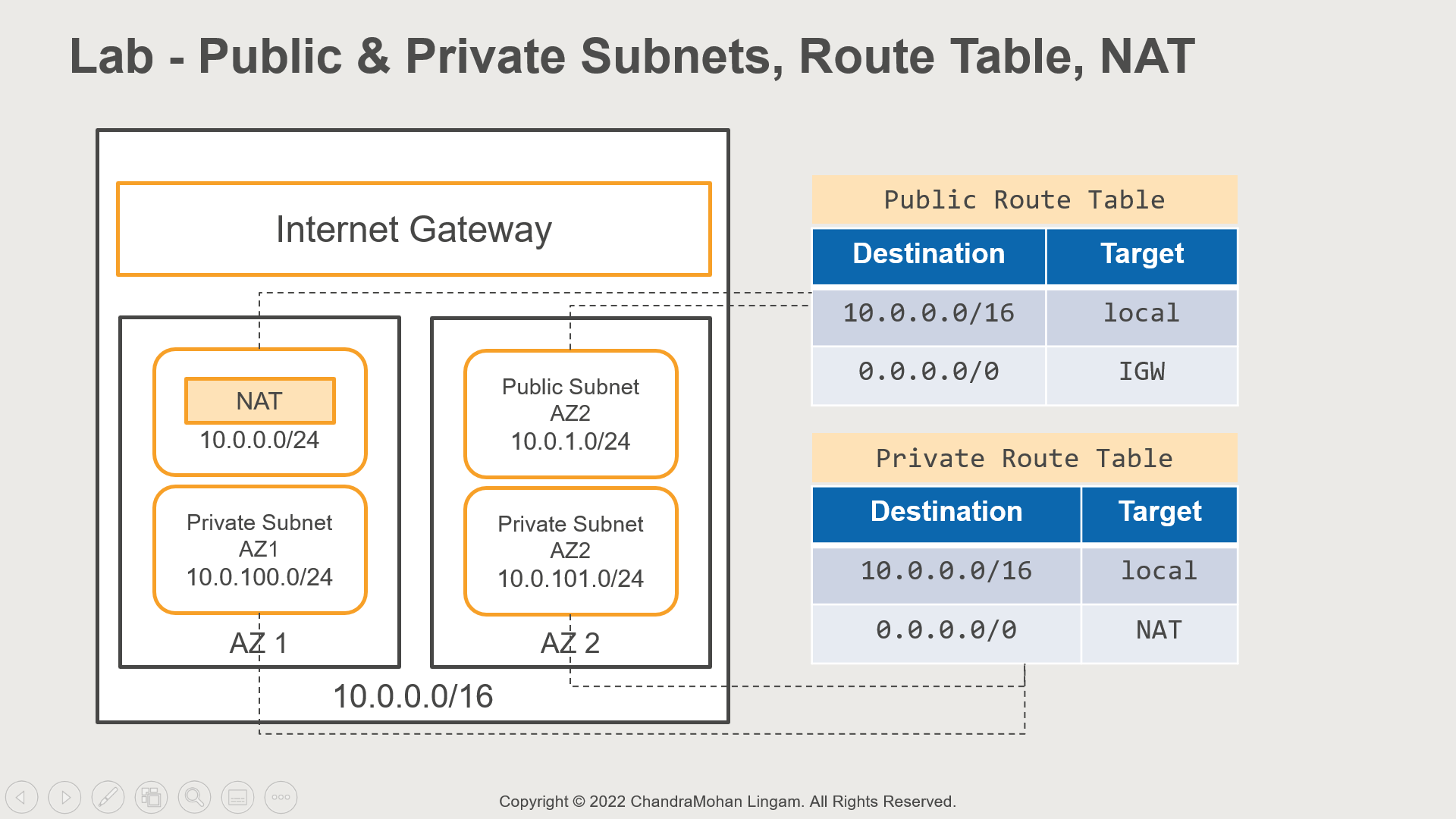
**Lab Instructions – Public & Private Subnets, Route Table, NAT, Bastion Host**

This "new VPC Console experience" instruction complements the next video **Lab – Public & Private Subnets, Route Table, NAT**. I recommend watching the video first and following these instructions to complete the lab.

This lab will require 1-hour to complete. Plan to complete this lab in a single sitting



For this lab, we will create a VPC with Public and Private Subnets.

We will use the VPC wizard to create the VPC – however, it will create subnets only in one Availability Zone

a. 10.0.0.0/24 Public AZ 1

b. 10.0.100.0/24 Private AZ 1

For high availability, it is recommended that we use at least two availability zones

So, we will create the public and private subnets in the second availability zone ourselves

a. 10.0.1.0 Public AZ 2

b. 10.0.101.0 Private AZ 2

In addition, we will have an internet gateway

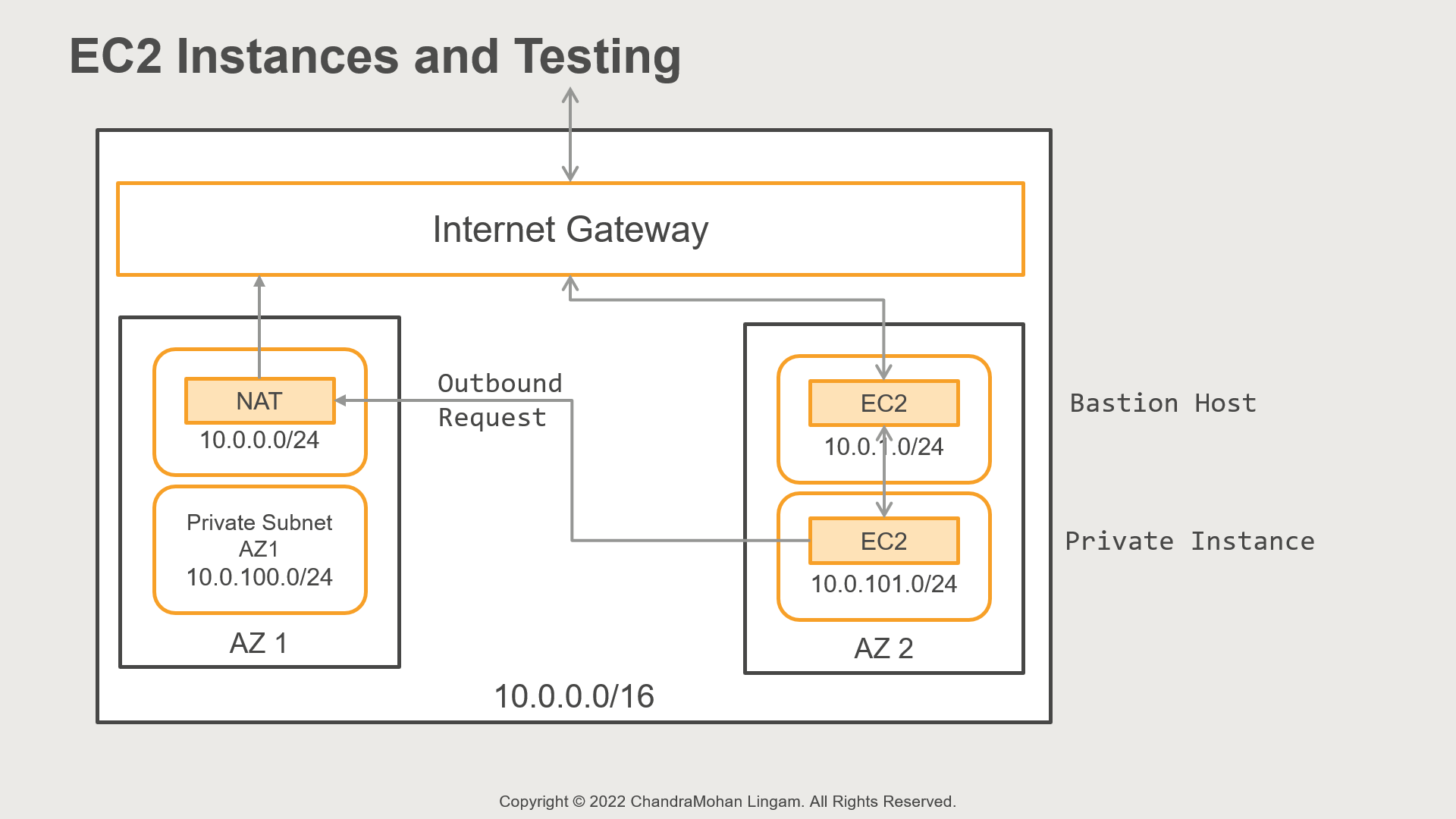
We will also deploy a NAT Gateway to allow instances in the private subnet to send the request to the internet

There are two route tables – one for public subnets and another for private subnets

The public subnet route table allows all local traffic, and the non-local traffic is routed to the internet gateway

The private subnet route table allows all local traffic, and the non-local traffic is routed to the NAT gateway

We will then deploy two EC2 instances



One in the public subnet (Public AZ 2) and another in the private subnet (Private AZ 2)

The public instance has internet access – so we can log in to the server, and the server can directly make outbound calls to the internet

However, to access the private instance, we need to first login into the public instance and then log in to the private instance

So, in this case, the public instance will act as a bastion host or jump server to access private resources

And from the private instance, we will also make an outbound internet call by using NAT Gateway

After this hands-on exercise, you will have a clear understanding of how to configure a public and private subnet, route table, and the purpose of a NAT. You will also learn how to login to a private instance

Note that there is a 4.5 cent (USD) hourly charge for NAT Gateway. So, ensure you follow the cleanup steps at the end to remove the resources that were created

This lab will require 1-hour to complete. Plan to complete this lab in a single sitting

**Updated: April 2023 [Use Amazon Linux 2 AMI. Do not use Amazon Linux 2023]**

Steps – New VPC Console Experience

The Create VPC console has a new interface and layout. Follow these instructions if you see the new console

1. Select VPC Dashboard from the left navigation pane

2. Click Create VPC

3. Under VPC Settings, choose VPC and more

4. Under Name tag auto-generation

a. Enable Auto-generate

b. Enter: PublicPrivateVPC

5. IPv4 CIDR Block: 10.0.0.0/16

6. Number of Availability Zones: 1

7. Expand Customize AZs and choose the first availability zone from the drop-down

8. Number of Public Subnets: 1

9. Number of Private Subnets: 1

Note: we are going to ask the wizard to create subnets in one AZ and then manually add subnets in the second AZ

10. Expand Customize subnets CIDR blocks

a. Public subnet CIDR block: 10.0.0.0/24

b. Private subnet CIDR block: 10.0.100.0/24

11. NAT gateways - In 1 AZ [Elastic IP is automatically created and assigned]

12. VPC endpoints - None

13. Create VPC

14. Wizard will take 2-5 minutes to complete – most of the time is used for launching a NAT Gateway

Rename Subnets

1. Select Subnets from the left navigation pane

2. We can see subnets from both default and the new VPC. So, let’s sort by VPC column

3. For the subnet with IPv4 CIDR: 10.0.0.0/24, change the Name to PublicSubnetAZ1

4. For subnet with IPv4 CIDR: 10.0.100.0/24, change the Name to PrivateSubnetAZ1

5. We can now continue with rest of the steps

Steps – Elastic IP

1. Open VPC console

2. Make sure you are in the correct region

3. We need an Elastic IP for this lab to assign to NAT Gateway. So, let’s allocate an Elastic IP

4. Select Elastic IPs in the left navigation pane

5. Use the Allocate Elastic IP address option

6. And Allocate

7. We will use this Elastic IP for the NAT Gateway

Steps - Create a VPC with Public and Private Subnets

1. Select VPC Dashboard from the left navigation pane

2. Click Launch VPC Wizard

3. Wizard currently provides predefined VPC configurations

a. VPC with a single public subnet

b. VPC with public and private subnets

c. VPC with the public, private subnet, and VPN

d. VPC with private subnets only and VPN

4. We will use the VPC with Public and Private Subnets option

5. This option will create a public and private subnet and also deploy a NAT gateway in the public subnet

6. Select the configuration

7. Let’s use these configurations for our VPC

a. IPv4 CIDR Block: 10.0.0.0/16

b. VPC name: PublicPrivateVPC

c. Public subnet’s IPv4 CIDR: 10.0.0.0/24

d. Availability Zone: choose the first one [in my case, I use us-west-2a in Oregon]

e. Public subnet name: PublicSubnetAZ1

f. Private subnet’s IPv4 CIDR: 10.0.100.0/24 [companies follow a convention of adding 100 to the public subnet to indicate a private subnet]

g. Availability Zone: choose the first one [in my case, I use us-west-2a]

h. Private subnet name: PrivateSubnetAZ1

i. Elastic IP Allocation ID: choose the Elastic IP you created earlier

j. Create VPC

k. Wizard will take 2-5 minutes to complete – most of the time is used for launching a NAT Gateway

l. VPC is ready now

Steps – Review the Subnets, Route Table

1. Let’s review the VPC components

2. Select Subnets from the left navigation pane

3. We can see subnets from both default and the new VPC. So, let’s sort by VPC column

4. Our VPC has a public and private subnet

5. The public subnet is 10.0.0.0/24, and the Private subnet is 10.0.100.0/24

6. Available IPv4 addresses: public subnet has 250 addresses left, whereas the private subnet has 251 addresses left. Why this difference? Why is there one less address in the public subnet?

7. The reason is NAT Gateway is deployed in the public subnet, and it used up one private IP address

8. Both subnets are in the same availability zone

9. Let’s now look at the route tables

10. Select Route Tables from the left navigation pane

11. Sort by VPC column

12. There is one Main route table as well as another route table – so, our VPC has two route tables

13. Let’s select the Main route table

14. Go to the Routes tab on the detail page

15. This has two entries: one for local traffic, and all other traffic is sent to the NAT Gateway. So, this route table is for private subnets

16. Let’s tag this route as PrivateRoute

a. Click on the Pen Icon in the Name column

b. In the Edit Name text box: type PrivateRoute

c. Save

17. Let’s see the subnet associations

18. Select the Subnet associations tab in the detail pane

19. Our private subnet is not explicitly associated

20. Let’s Edit subnet associations

21. Choose PrivateSubnetAZ1

22. Save associations

23. Let’s now select the second route table for PublicPrivateVPC

24. Go to the Routes tab on the detail page

25. This has two entries: one for local traffic, and all other traffic is sent to the Internet Gateway. So, this route table is for public subnets

26. Let’s tag this route as PublicRoute

a. Click on the Pen Icon in the Name column

b. In the Edit Name text box: type PublicRoute

c. Save

27. Let’s see the subnet associations

28. Select the Subnet associations tab in the detail pane

29. Our public subnet is explicitly associated. So, we are good here!

30. There is also an internet gateway that is attached to our VPC [select Internet Gateways from the left navigation pane]

31. And we also have a NAT Gateway [select NAT Gateways in the left navigation pane]

Steps – Add subnets in the second availability zone

1. Let’s add public and private subnets in the second availability zone

2. For high availability, you need an identical setup in at least two availability zones so that the application can be deployed across multiple AZs to handle AZ failure

3. Let’s start with the public subnet

4. Choose Subnets in the left navigation pane

5. Create subnet

a. VPC ID: Select PublicPrivateVPC

b. Subnet name: PublicSubnetAZ2

c. Availability Zone: choose the second one [in my case, I use us-west-2b in Oregon]

d. IPv4 CIDR block: 10.0.1.0/24

e. And Create subnet

6. We need to ensure public route table is attached to this subnet

7. Select the PublicSubnetAZ2

8. Go to the Route table tab in the detail pane

9. By default, the PrivateRoute is automatically attached [as it is the main route table]

10. Let’s Edit route table association

a. Route table ID: choose PublicRoute

b. Save

11. PublicSubnetAZ2 has the correct route table attached

12. Let’s now create the private subnet

13. Create subnet

a. VPC ID: Select PublicPrivateVPC

b. Subnet name: PrivateSubnetAZ2

c. Availability Zone: choose the second one [in my case, I use us-west-2b in Oregon]

d. IPv4 CIDR block: 10.0.101.0/24

e. And Create subnet

14. We need to ensure private route table is attached to this subnet

15. Select the PrivateSubnetAZ2

16. Go to the Route table tab in the detail pane

17. By default, the PrivateRoute is automatically attached

18. Let’s Edit route table association

a. Route table ID: choose PrivateRoute

b. Save

c. This will explicitly attach the PrivateRoute to the subnet

19. All our subnets have the correct route tables attached

20. We have four subnets – the first availability zone has a public and private subnet, and we have a similar setup in the second availability zone

Steps – Launch Bastion Host EC2 Instances

1. We need to now deploy an EC2 instance in the public subnet and another instance in the private subnet

2. Open EC2 console

3. Launch an Instance

4. Name the instance: BastionHost

5. We will use the Amazon Linux AMI

a. Choose Amazon Linux AMI

b. From AMI drop down: Select **Amazon Linux 2**

***NOTE: Do NOT use Amazon Linux 2023 AMI. This version uses a different setting for SSH and may not work correctly***

6. For instance type, use t2.micro or t3.micro – these are free-tier eligible instances. If you don’t see t2.micro, simply choose t3.micro

7. Key pair: Specify the Keypair to Login to the machine

8. Network settings - Specify VPC information

a. Edit

b. Network: Select PublicPrivateVPC

c. Subnet: PublicSubnetAZ2

d. Auto-assign Public IP: Enable

e. Create a new security group

f. Security group name: SSHSG

g. Allow SSH traffic from anywhere

h. Type: SSH, Protocol: TCP, Port: 22, Source: Select Anywhere [this rule should already be there]

9. Review and Launch

10. Wait until server state is Running

11. This will be our public instance, and we named it the BastionHost

Steps - Launch EC2 Private Instance

1. Let’s now launch the private instance

2. Name the instance: PrivateInstance

3. We will use the Amazon Linux AMI

a. Choose Amazon Linux AMI

b. From AMI drop down: Select **Amazon Linux 2**

***NOTE: Do NOT use Amazon Linux 2023 AMI. This version uses a different setting for SSH and may not work correctly***

4. For instance type, use t2.micro or t3.micro – these are free-tier eligible instances. If you don’t see t2.micro, simply choose t3.micro

5. Key pair: Specify the Keypair to Login to the machine

6. Network settings - Specify VPC information

a. Edit

b. Network: Select PublicPrivateVPC

c. Subnet: PrivateSubnetAZ2

d. Auto-assign Public IP: Disable [Instances in private subnet are not accessible from the internet. So, public IP is not required]

e. Select an existing security group

f. Security group name: SSHSG

7. Review and Launch

8. Wait until server state is Running

9. This will be our private instance

Steps – Login to Bastion Host and Private Instance

1. Since BastionHost has a public IP, we can directly connect to it from our laptop

2. However, PrivateInstance is reachable from only inside the VPC

3. So, we need to login to Bastion Host first and then login to the PrivateInstance

4. But, how do we present our keypair credentials to the private instance? You don’t want to store your keypair in Bastion Host. Instead, we will ask Putty to forward the credentials when needed.

5. I will show you how to perform credential forwarding using Putty. If you use SSH client on Mac or Linux laptop, please follow the steps in this article

<https://aws.amazon.com/blogs/security/securely-connect-to-linux-instances-running-in-a-private-amazon-vpc/>

6. When you installed Putty in the earlier labs, it also installed a tool called Pageant (Putty Authentication Agent)

7. The pageant app is used for credential forwarding, and we need to add our private key to pageant

8. Start Pageant from Windows search

9. Pageant runs in the background, and you need to double-click on the tiny computer icon in the Windows tray [bottom right corner in the task bar]

10. Pageant app will show available key list

11. Choose Add Key

12. And browse to your key pair file and open

13. Your key pair should be added now

14. Click on Close

15. Launch Putty

16. Load your existing saved sessions [in my case, it is OregonServer]

17. Rename the Saved Sessions

a. In the Saved Session text box, type BastionHostOregonServer

b. Save

18. Expand SSH tree in the left navigation pane

19. Select Auth

20. Enable Allow agent forwarding option

21. Clear the Private key file for authentication text box

22. Select Sessions

23. Paste the Bastion Host Public IP address in the Hostname: ec2-user@BastionHostPublicIP

24. Save

25. And click Open

26. Once you log in, you should see the message in the Linux terminal

Authenticating with public key “your key pair name” from agent

27. We are now logged in to the Bastion Host

28. Try pinging [www.google.com](http://www.google.com/)

ping www.google.com

29. You should see a response from the google server

30. Now, let’s login to the private instance

31. From the EC2 console, select your PrivateInstance

32. Copy the Private IP Address

33. Login using SSH

ssh ec2-user@PrivateInstanceIP

34. Type Yes for the prompt and connect

35. You are now connected to the private instance

36. Try pinging [www.google.com](http://www.google.com/) from your private instance

ping www.google.com

37. And you should see the response from google

38. The private instance does not have a direct connection to the internet. Instead, the outbound request is routed to the NAT Gateway by the private route table, and the NAT gateway is pinging the google server. NAT Gateway then forwards the response from google server to the private instance

39. This concludes the lab, and close the terminal

Steps – Cleanup

1. To clean up, from EC2 console, terminate BastionHost and PrivateInstance

2. Wait until both instances are Terminated

3. From VPC Console

4. Select NAT Gateways from the left navigation pane

5. Choose the NAT Gateway that is attached to PublicPrivateVPC

6. From Actions, select Delete NAT Gateway

7. Confirm and delete

8. Wait until state changes to Deleted [this may take a minute or two]

9. Select Elastic IPs from the left navigation pane

10. Choose the Elastic IP and from Actions, select Release Elastic IP addresses

11. And Release

12. Select Your VPCs from the left navigation pane

13. Choose PublicPrivateVPC

14. From the Actions menu, select Delete VPC

15. Confirm and delete

Summary

In this lab, you learned how to create a new VPC

Configure Public and Private Subnets

How to attach the correct route table to the subnet

How private instances use a NAT

And how to bastion host to login to the private instances

