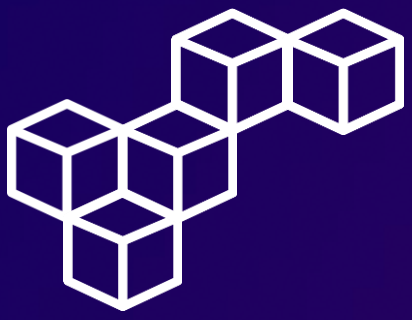
AWS re:Start LAB

Configuring an Amazon VPC

WEEK 10

aws re/start



Overview

Amazon Virtual Private Cloud (Amazon VPC) gives you the ability to provision a logically isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selecting your IP address ranges, creating subnets, and configuring route tables and network gateways.

Configuring an Amazon VPC involves setting up a secure and customizable network environment tailored to your specific requirements. This process includes defining subnets within your IP address range, associating route tables to manage traffic flow, and establishing network gateways to enable communication between your VPC and the internet or other networks. Additionally, you can implement security groups and network ACLs to control inbound and outbound traffic at the instance and subnet level, ensuring robust protection and efficient traffic management within your VPC.

Topics covered

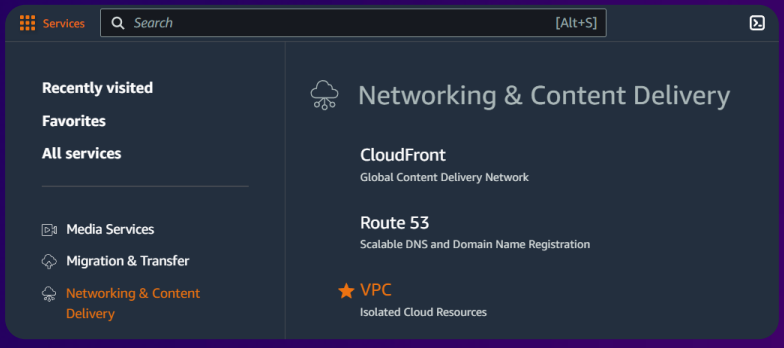
- Create a VPC with a private and public subnet, an internet gateway, and a NAT gateway.
- Configure route tables associated with subnets to local and internet-bound traffic by using an internet gateway and a NAT gateway.
- Launch a bastion server in a public subnet.
- Use a bastion server to log in to an instance in a private subnet.

Task 1

Creating a VPC

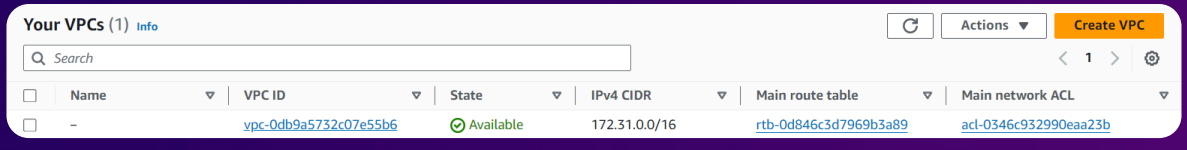
Step 1: Access the AWS Management Console

Open the AWS Management Console, and select VPC.



Step 2: Create VPC

Navigate to the **Your VPCs** section, and select Create VPC.



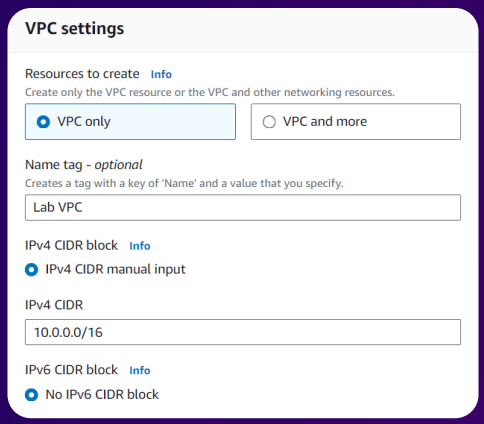
aws re/start

Task 1

Creating a VPC

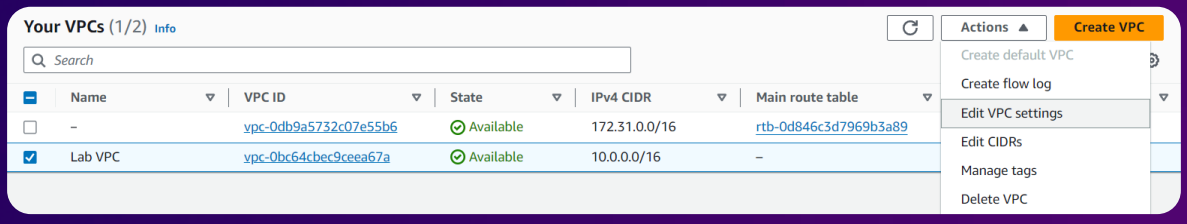
Step 3: VPC settings

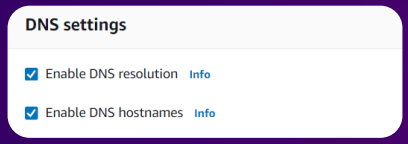
In the **VPC settings** section, configure the following settings.



Step 4: Edit DNS settings

Select the newly created **Lab VPC**, and choose Edit VPC settings. In the **DNS settings** section, select Enable DNS hostnames.





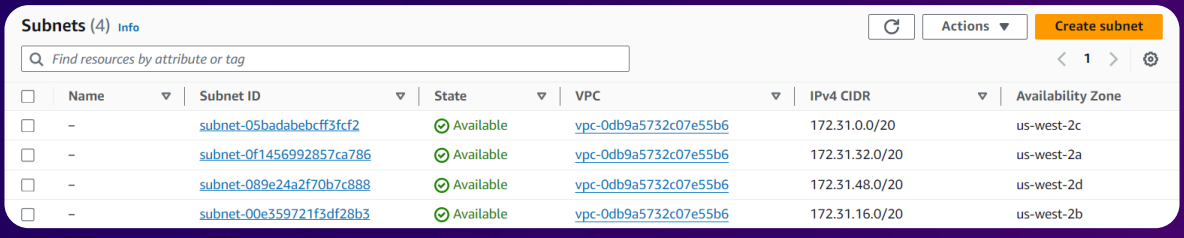
aws re/start

Task 2

Creating subnets

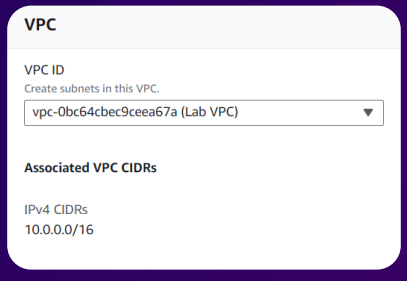
Step 1: Create Public subnet

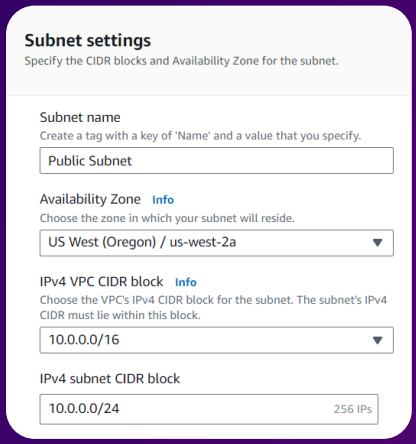
Navigate to the **Subnets** section, and select Create subnet.



Step 2: Public Subnet settings

In the **Create subnet** page, configure the following settings.





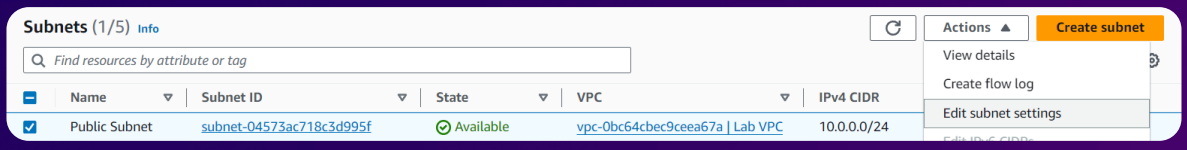
aws re/start

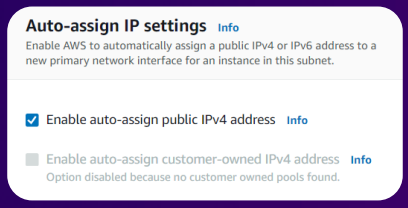
Task 2

Creating subnets

Step 3: Edit Auto-assign IP settings

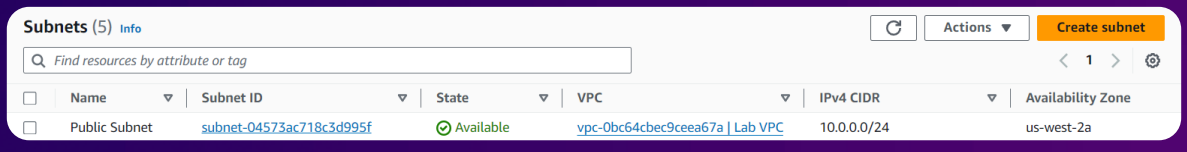
Select the newly created **Public Subnet**, and choose Edit subnet settings. In the **Auto-assign IP settings** section, select Enable auto-assign public IPv4 address.





Step 4: Create Private subnet

Navigate to the **Subnets** section, and select Create subnet.



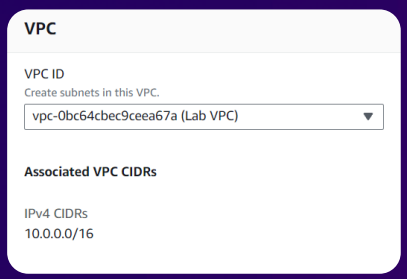
aws re/start

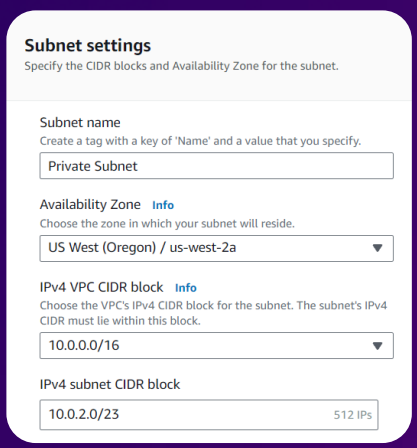
Task 2

Creating subnets

Step 5: Private Subnet settings

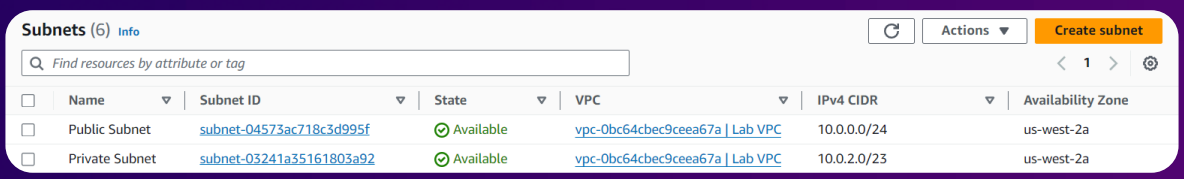
In the **Create subnet** page, configure the following settings.





Step 6: Review Subnets creation

In the **Subnets** section, review the Available State of the newly created subnets **Public Subnet** and **Private Subnet**.



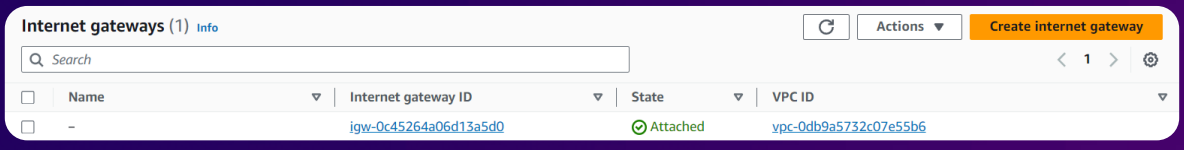
aws re/start

Task 3

Creating an internet gateway

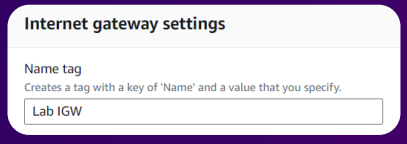
Step 1: Create internet gateway

Navigate to the **Internet gateways** section, and select Create internet gateway.



Step 2: Internet gateway settings

In the **Internet gateway settings** section, for the Name tag, enter Lab IGW.



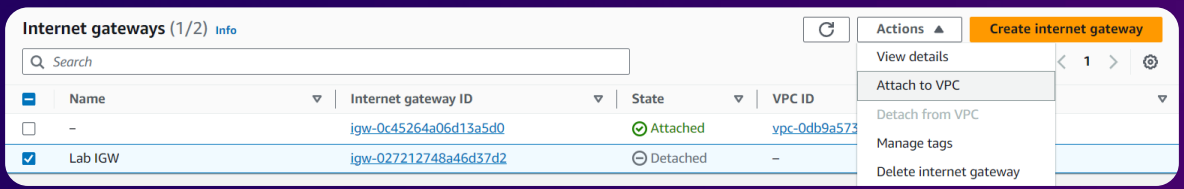
aws re/start

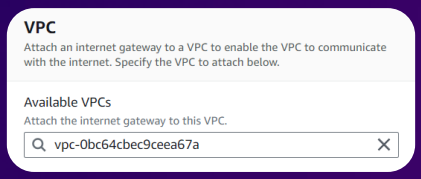
Task 3

Creating an internet gateway

Step 3: Attach to VPC

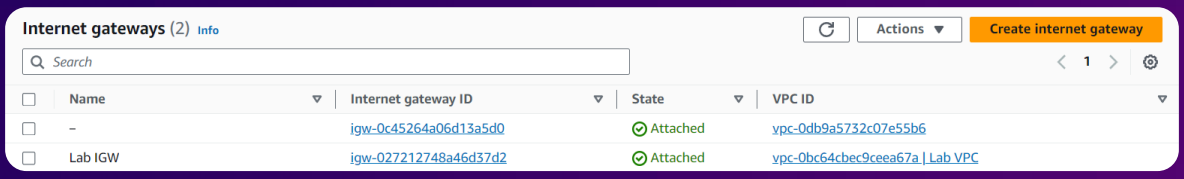
Select the newly created **Lab IGW**, and choose Attach to VPC.





Step 4: Review Attachment

In the **Internet gateways** section, review the **Lab IGW** attachment to the **Lab VPC**.



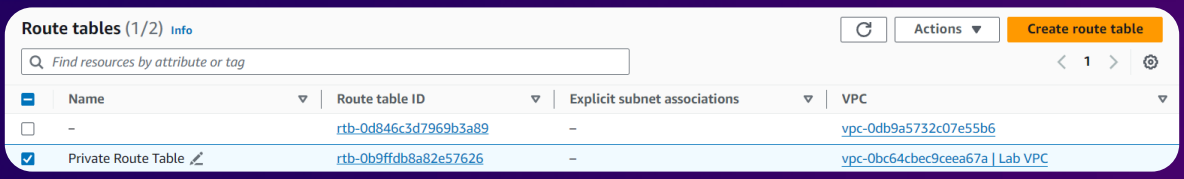
aws re/start

Task 4

Configuring route tables

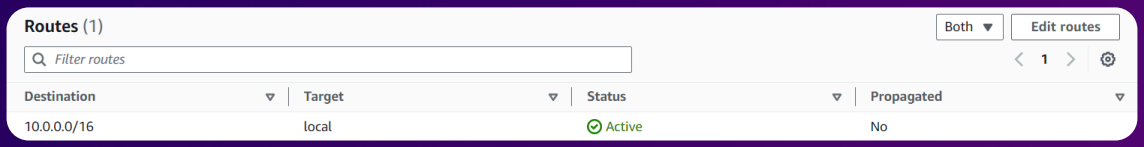
Step 1: Edit Route Table Name

Navigate to the **Route tables** section, select the route table that includes Lab VPC in the VPC column. In the Name column, choose the edit icon, and enter Private Route Table.



Step 2: Review routes

Select the **Private Route Table**, and choose the **Routes** tab. There is currently only one route. It shows that all traffic destined for 10.0.0.0/16 (which is the range of the Lab VPC) will be routed locally. This option allows all subnets within a VPC to communicate with each other.



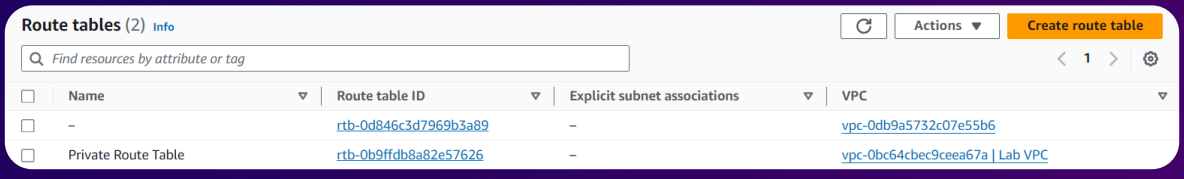
aws re/start

Task 4

Configuring route tables

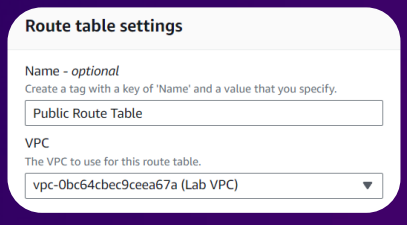
Step 3: Create Public Route Table

In the **Route tables** section, select Create route table.



Step 4: Public Route table settings

In the **Route table settings** section, configure the following settings.



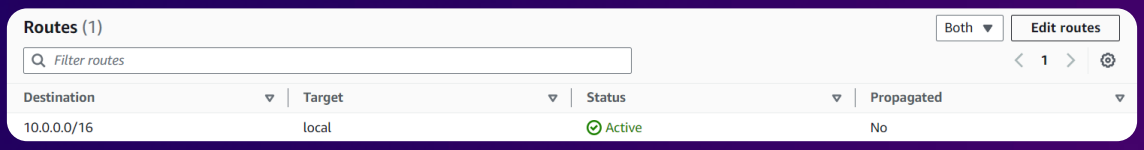
aws re/start

Task 4

Configuring route tables

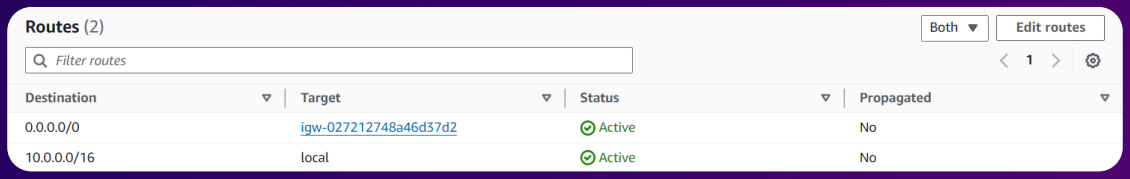
Step 5: Edit routes

Select the **Public Route Table**, choose the **Routes** tab, and select Edit routes.



Step 6: Add route

Add a route to direct internet-bound traffic (0.0.0.0/0) to the **Lab IGW** internet gateway.



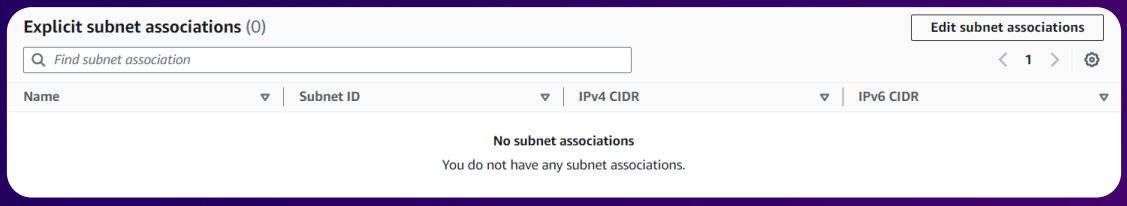
aws re/start

Task 4

Configuring route tables

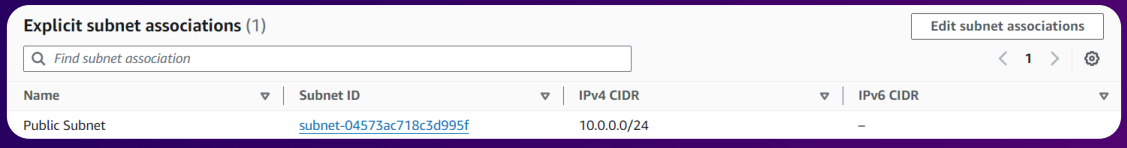
Step 7: Edit subnet associations

Choose the **Subnet associations** tab, and select Edit subnet associations, and select the **Public Subnet**.



Step 8: Review Explicit subnet associations

Review the **Public Route Table Explicit subnet associations**. The **Public Subnet** is now public because it has a route table entry that sends traffic to the internet through the internet gateway.



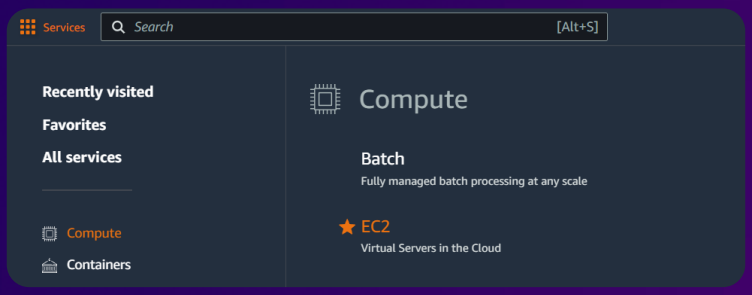
aws re/start

Task 5

Launching a bastion server in the public subnet

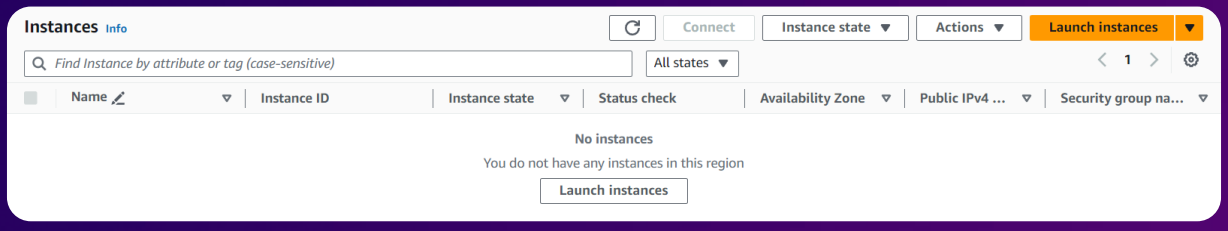
Step 1: Access the EC2 Management Console

In the AWS Management Console, select EC2.



Step 2: Launch the Bastion Server

Navigate to the **Instances** section, and select Launch instances.



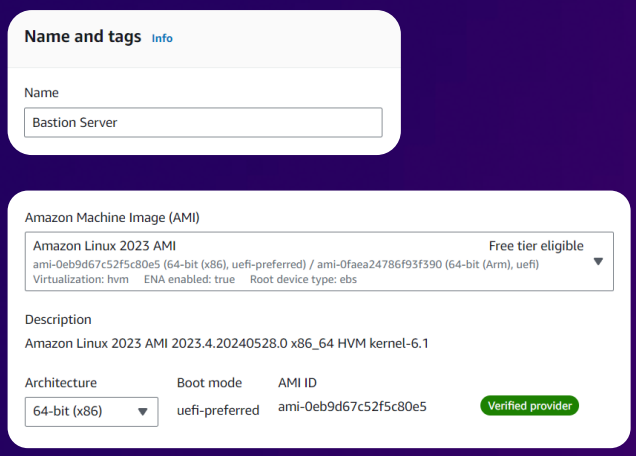
aws re/start

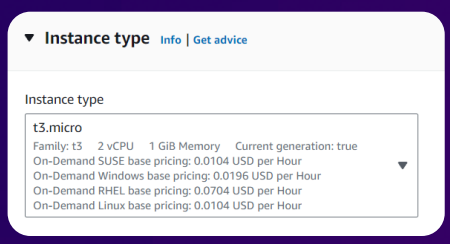
Task 5

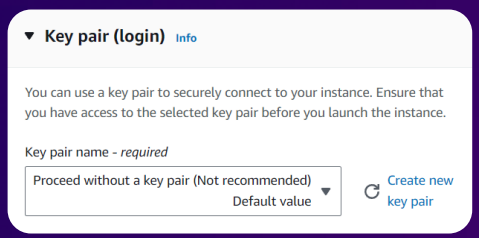
Launching a bastion server in the public subnet

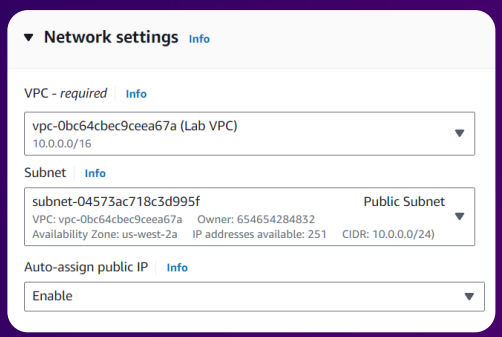
Step 3: Set up the instance

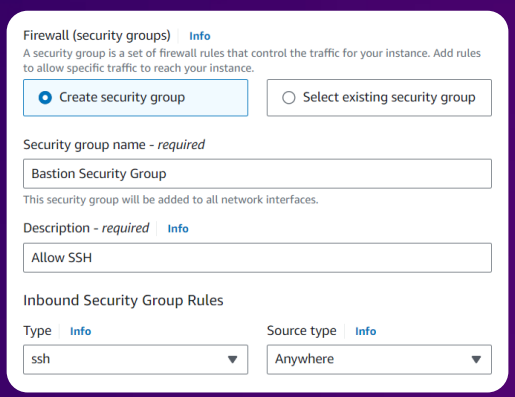
Use the following parameters to configure the instance settings.











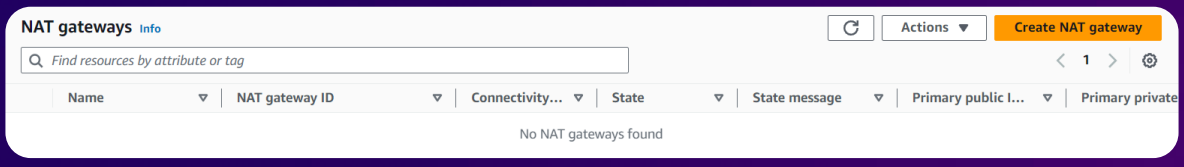
aws re/start

Task 6

Creating a NAT gateway

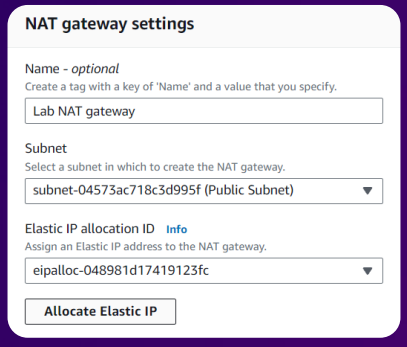
Step 1: Create NAT gateway

Navigate to the **NAT gateways** section, and select Create NAT gateway.



Step 2: NAT gateway settings

In the **NAT gateway settings** section, configure the following settings.



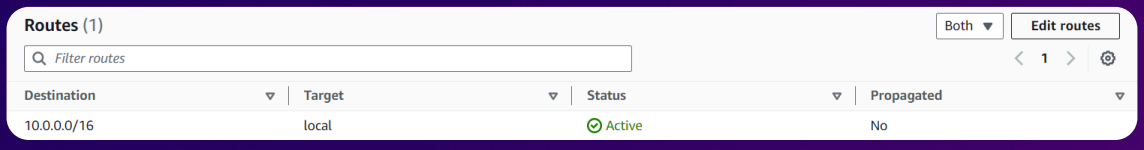
aws re/start

Task 6

Creating a NAT gateway

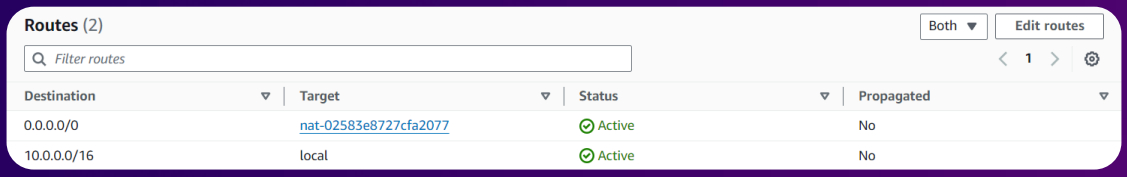
Step 3: Edit routes

Select the **Private Route Table**, choose the **Routes** tab, and select Edit routes.



Step 4: Add route

Add a route to direct internet-bound traffic (0.0.0.0/0) to the **Lab NAT gateway**. Resources in the private subnet that wish to communicate with the internet now have their network traffic directed to the NAT gateway, which forwards the request to the internet. Responses flow through the NAT gateway back to the private subnet.



aws re/start

Task 7

Optional challenge: Testing the private subnet

Step 1: Launch the Private Instance

Launch a new EC2 instance using the following settings.

Name and tags Info

Name

Private Instance

Instance type Info | Get advice

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour
On-Demand RHEL base pricing: 0.0704 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Proceed without a key pair (Not recommended)
Default value

Create new key pair

Advanced details Info

User data - *optional* Info

Upload a file with your user data or enter it in the field.

Choose file

```
#!/bin/bash
# Turn on password authentication for lab challenge
echo 'lab-password' | passwd ec2-user --stdin
sed -i 's|[#]*PasswordAuthentication no|PasswordAuthentication yes|g'
/etc/ssh/sshd_config
systemctl restart sshd.service
```

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible
ami-0eb9d67c52f5c80e5 (64-bit (x86), uefi-preferred) / ami-0faea24786f93f390 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.4.20240528.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	
64-bit (x86)	uefi-preferred	ami-0eb9d67c52f5c80e5	Verified provider

Network settings Info

VPC - *required* Info

vpc-0bc64cbec9ceea67a (Lab VPC)
10.0.0.0/16

Subnet Info

subnet-03241a35161803a92 Private Subnet
VPC: vpc-0bc64cbec9ceea67a Owner: 654654284832
Availability Zone: us-west-2a IP addresses available: 507 CIDR: 10.0.2.0/23)

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

(•) Create security group ( ) Select existing security group

Security group name - *required*

Private Instance SG

This security group will be added to all network interfaces.

Description - *required* Info

Allow SSH from Bastion

Inbound Security Group Rules

Type Info	Port range Info
ssh	22

Source type Info	Source Info
Custom	Add CIDR, prefix list or security g

10.0.0.0/16 ×

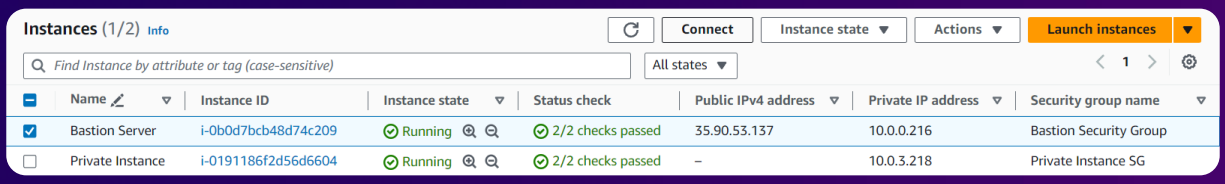
aws re/start

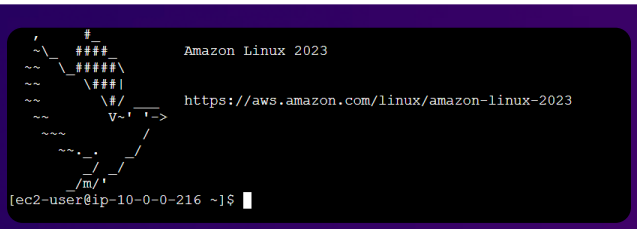
Task 7

Optional challenge: Testing the private subnet

Step 2: Connect to the Bastion Server

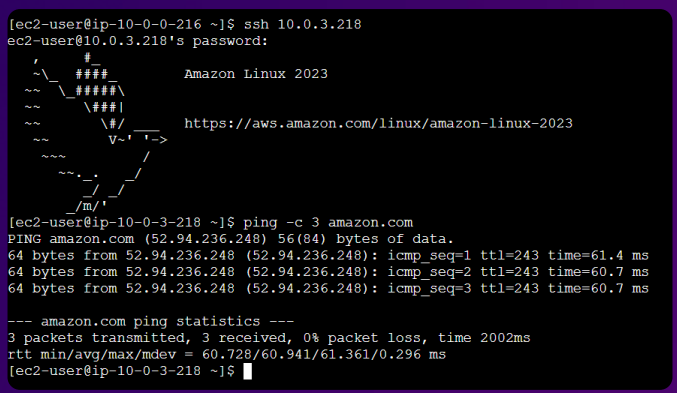
Connect to the **Bastion Server** using EC2 Instance Connect.



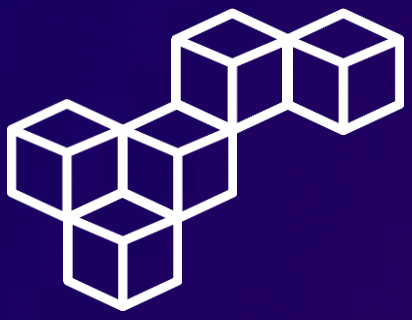


Step 3: Log in to the Private Instance

Log in to the **Private Instance** using the ssh command and the instance private IP address. You should now be connected to the **Private Instance**. Test the NAT gateway by running the ping command to confirm that the **Private Instance** can access the internet.



aws re/start



Conclusions

VPCs
VPCs provide isolated network environments within AWS, allowing full control over network configuration and security.

Subnets
Subnets partition your VPC into smaller network segments, enabling efficient resource organization and traffic management.

Internet Gateways
Internet Gateways facilitate internet access for resources within a VPC, allowing inbound and outbound traffic.

NAT Gateways
NAT Gateways enable instances in private subnets to access the internet while remaining inaccessible from the outside.

Routing Tables
Routing Tables direct traffic within your VPC, defining the paths for data to travel to its destination.

Bastion Hosts
Bastion Hosts serve as secure entry points for administrative access to instances in private subnets, enhancing overall security.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru