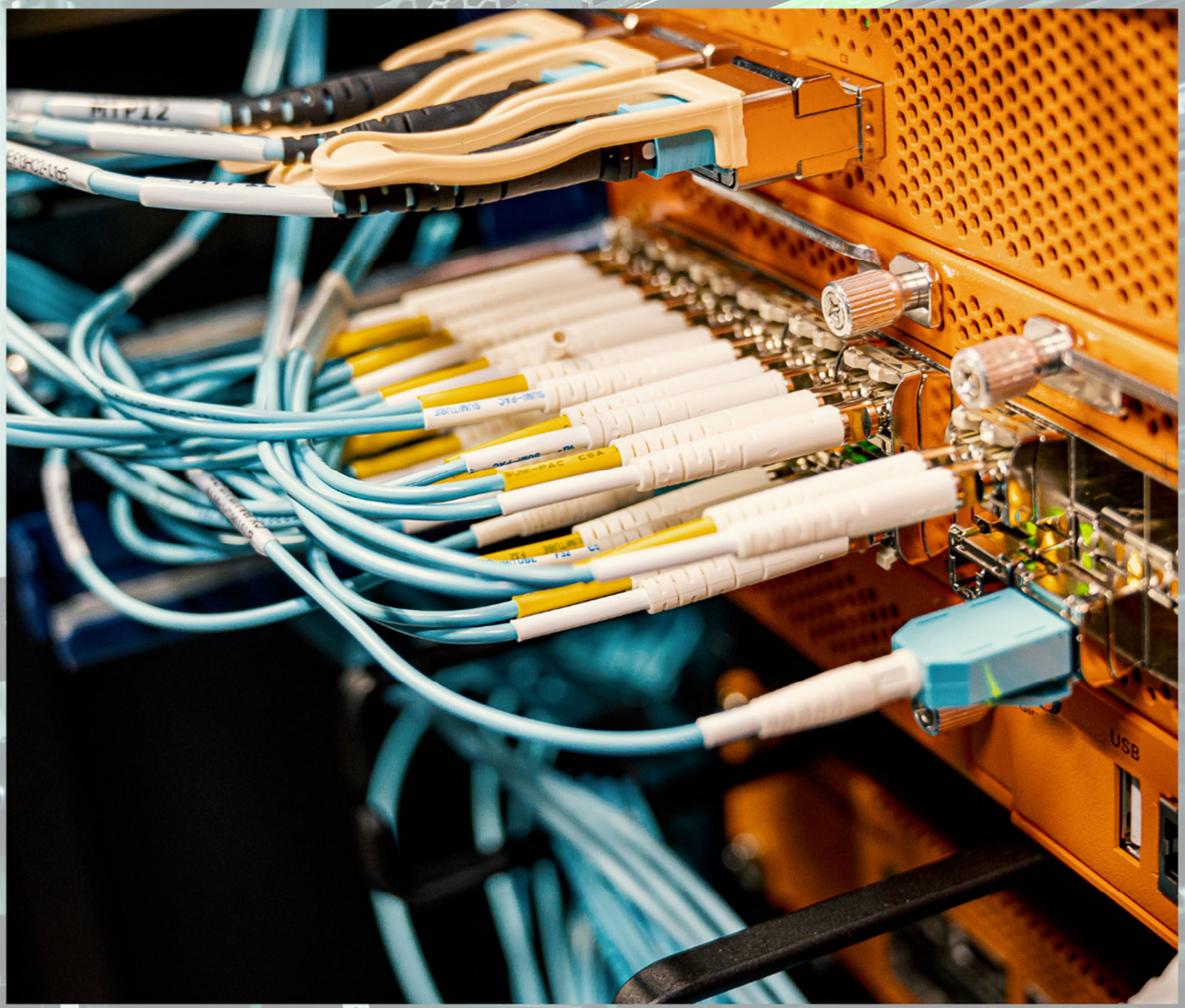
Alysia Chen

CCNP 3-4

Table of Contents

Lab 1: Windows Format and Installation



PURPOSE:

To learn how to set up operating systems.

BACKGROUND INFORMATION ON LAB CONCEPTS:

An operating system is a program that, after being booted, is used to manage other computer programs. In our case, we install Windows.

Windows is a Microsoft-proprietary operating system. First released in 1985, the latest version of Windows is now Windows 11. Additionally, Windows is the most popular operating system in the world (with 70% of the market share). Windows 11 was released in 2021.

Putty is used to access networking devices via SSH or Telnet. SSH is used to securely send commands over an insecure network, while Telnet does not provide this secure feature. This will facilitate configuring networking devices and the network in general, among other tasks important in Cisco Networking Academy.

Office 365 is used for many Microsoft Office programs in the classroom setting. For example, it contains Word (which is used to write documents), Excel (which is used to make spreadsheets), Outlook (which is used to communicate via electronic mail), among others. In this class, it is important to write lab writeups, which will rely heavily on applications like Microsoft Word.

Lenovo Commercial Vantage is used for configuring Windows 10 and 11. It is where updates are checked and run.

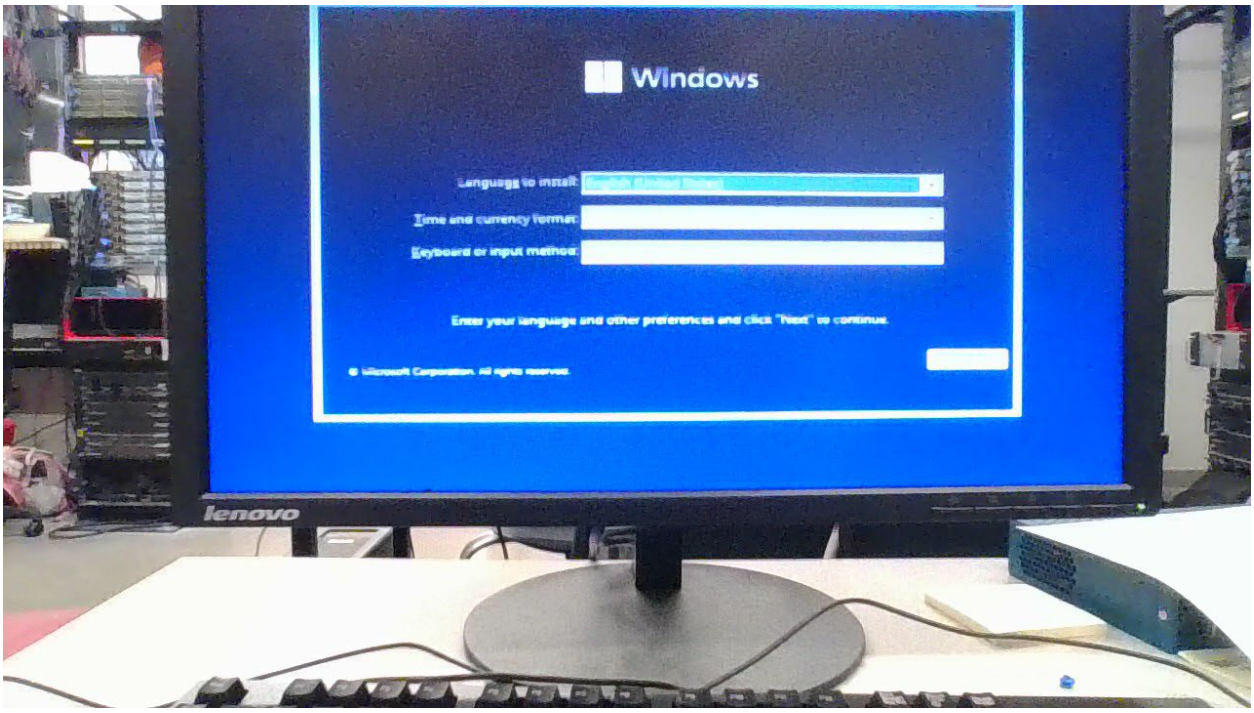
LAB SUMMARY:

Install operating system (Windows), install Putty, install Office 365, and install Lenovo Commercial Vantage.

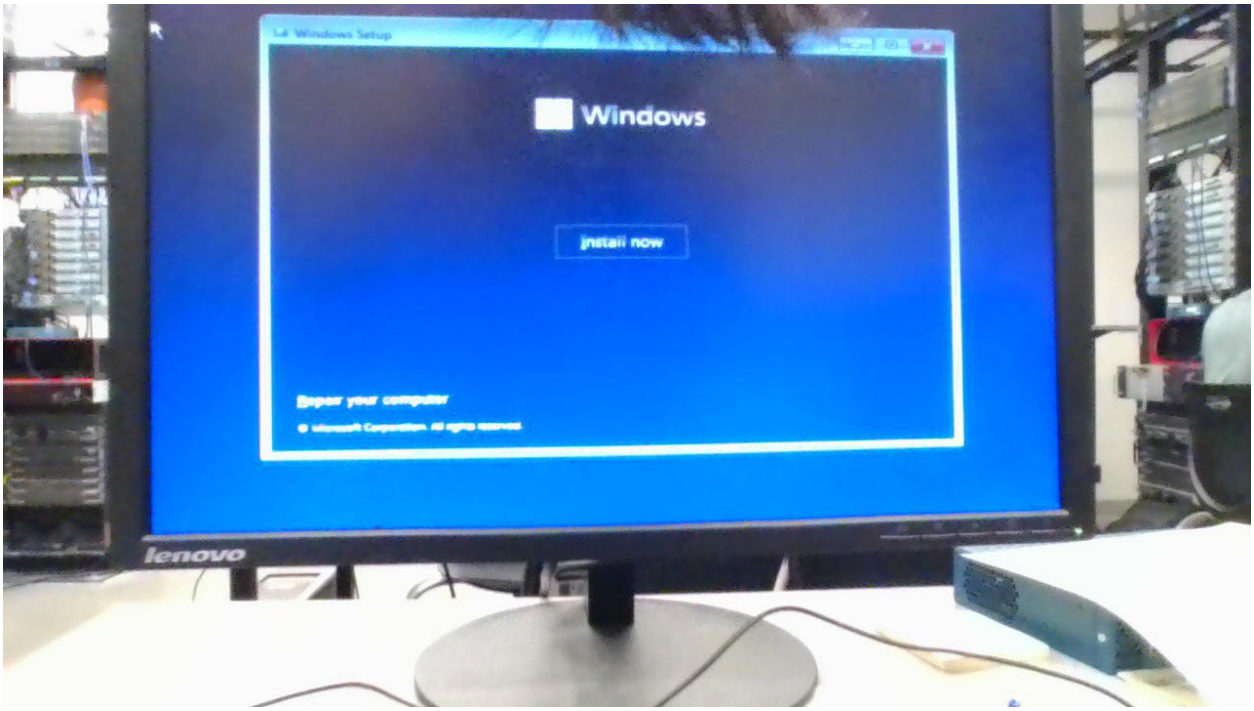
LAB COMMANDS:

First, plug the USB as well as the drive into the computer. Start the computer, spamming the F12 key.

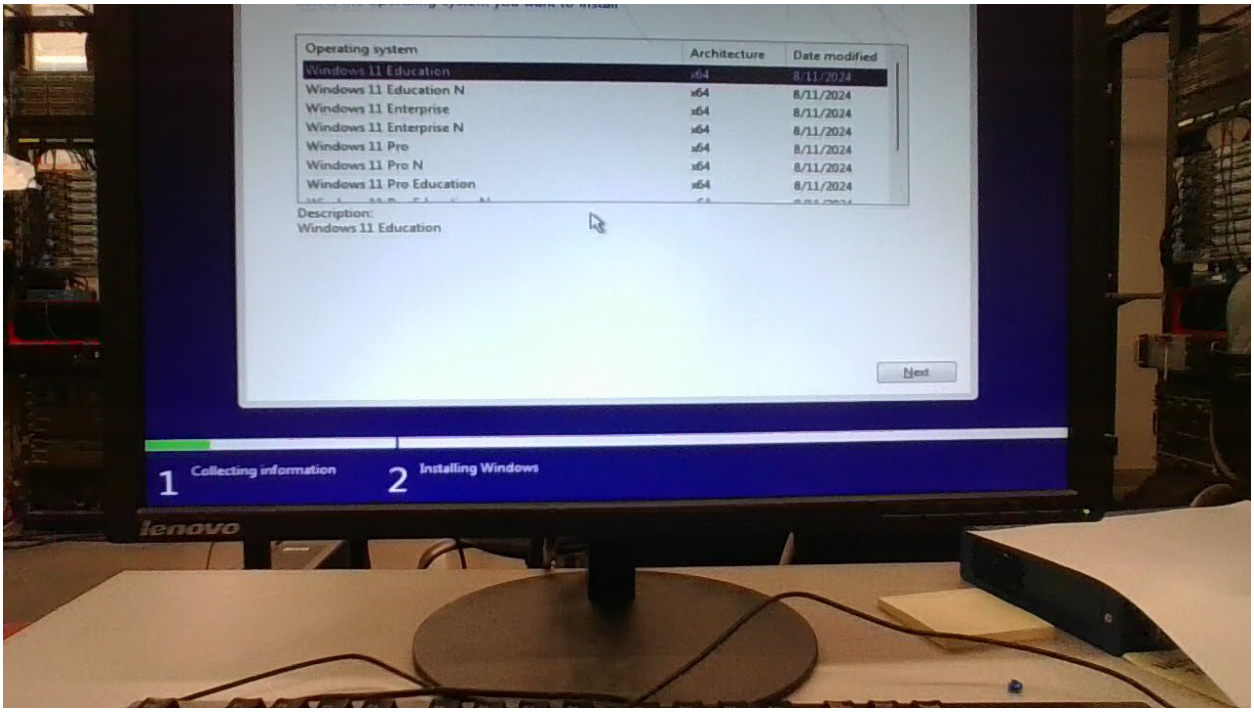
After a while, you will be taken to this page:



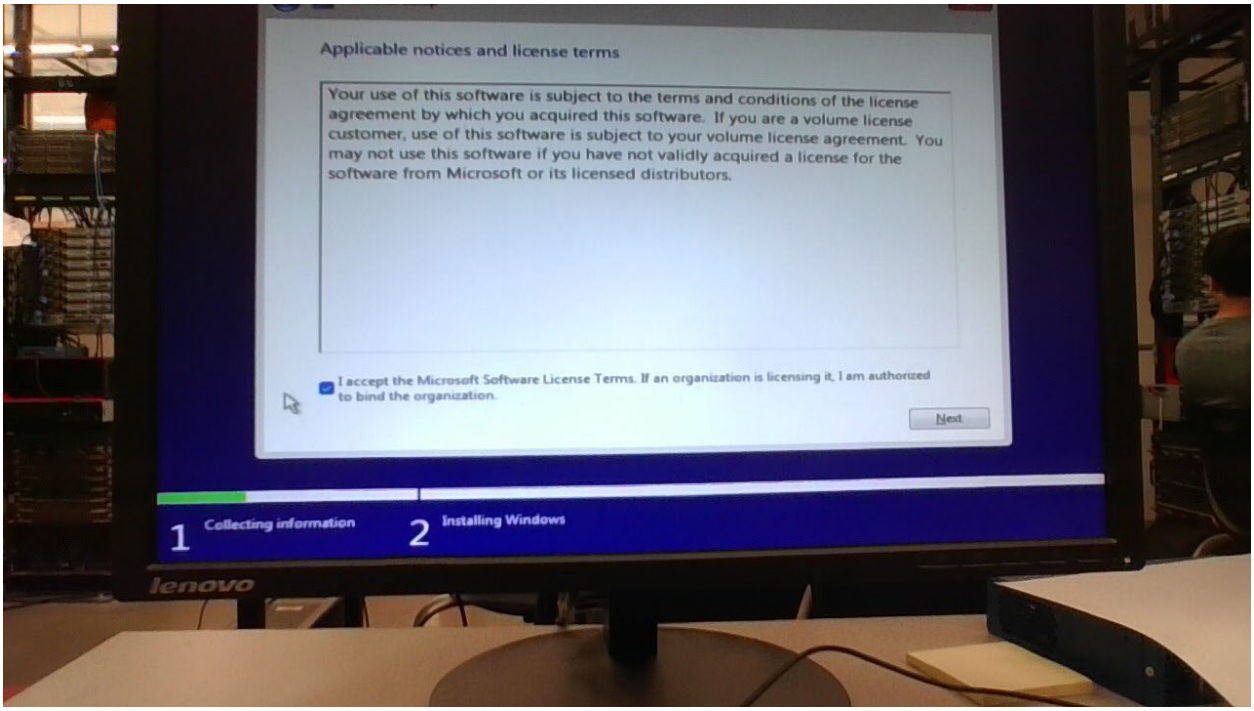
Select your language preference and click next.



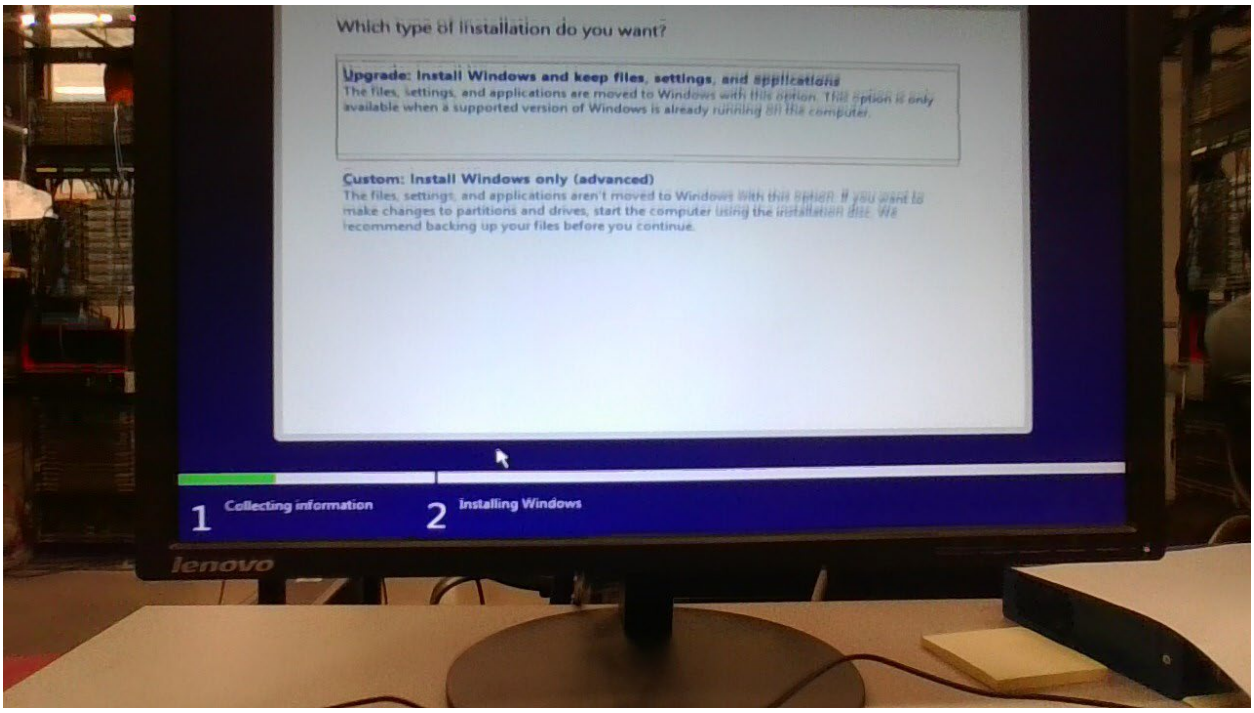
Click “Install Now”. After a while, a new page will load.



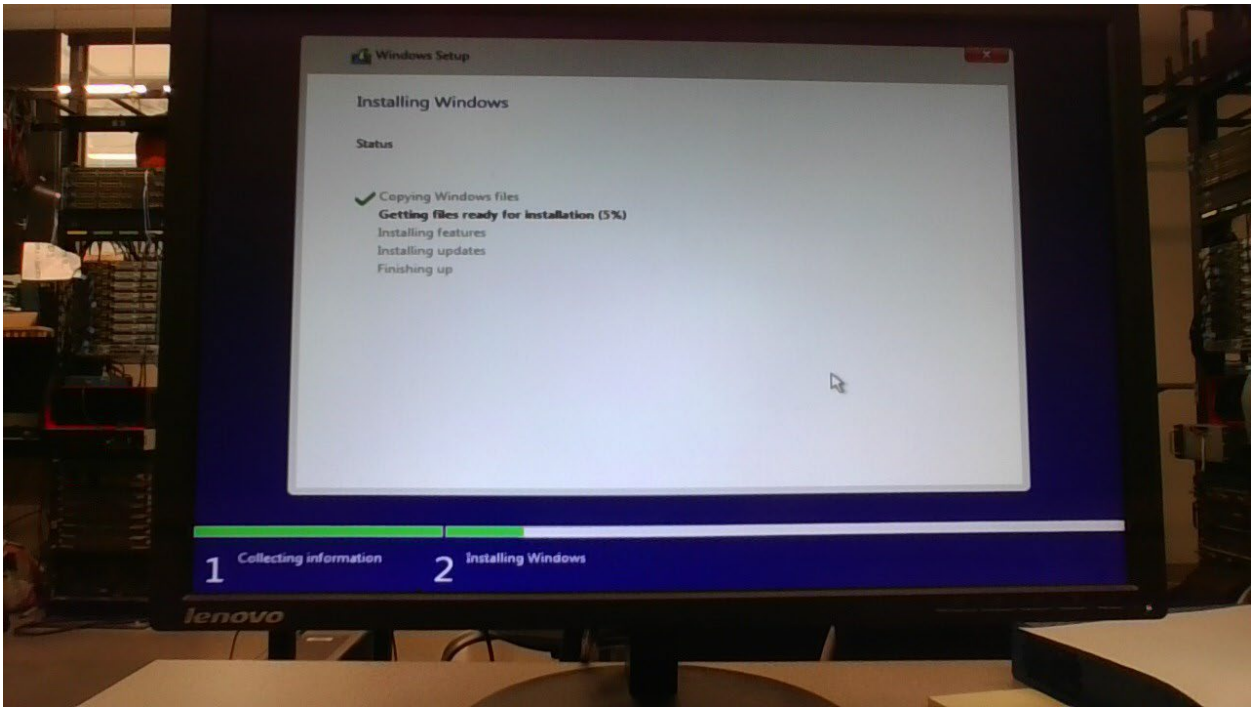
From the options above, select the Windows 11 Education option.



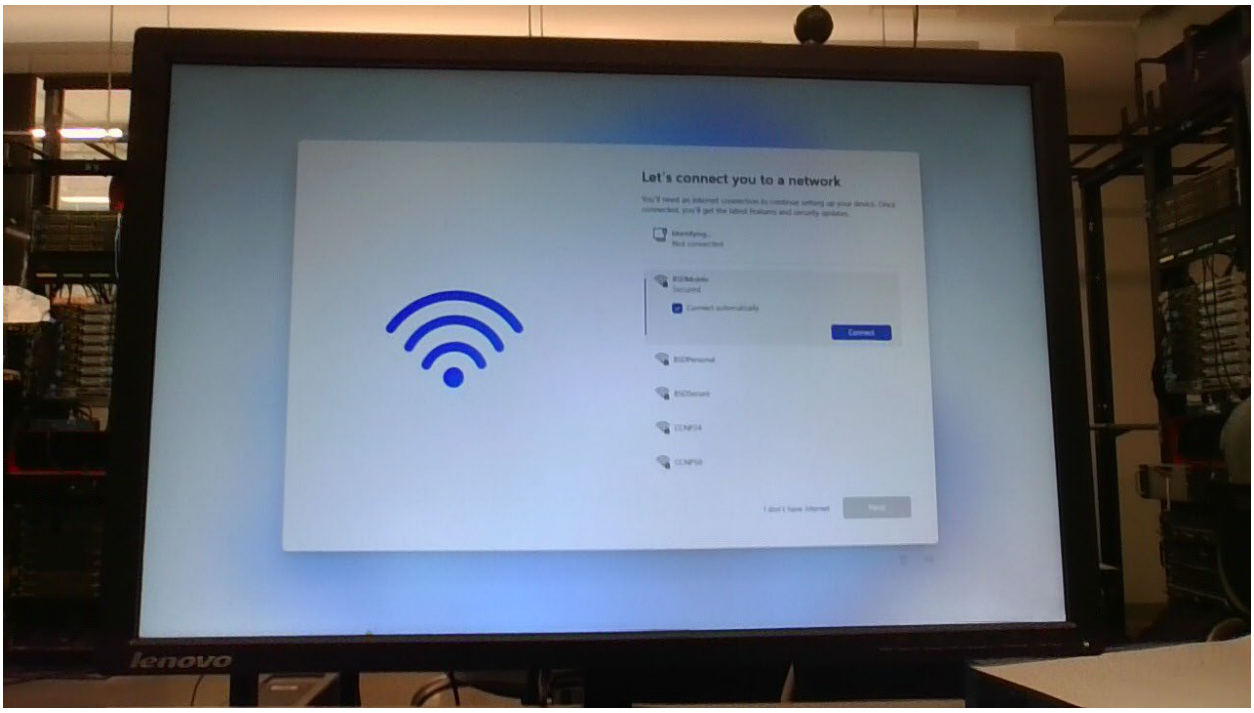
Click next.



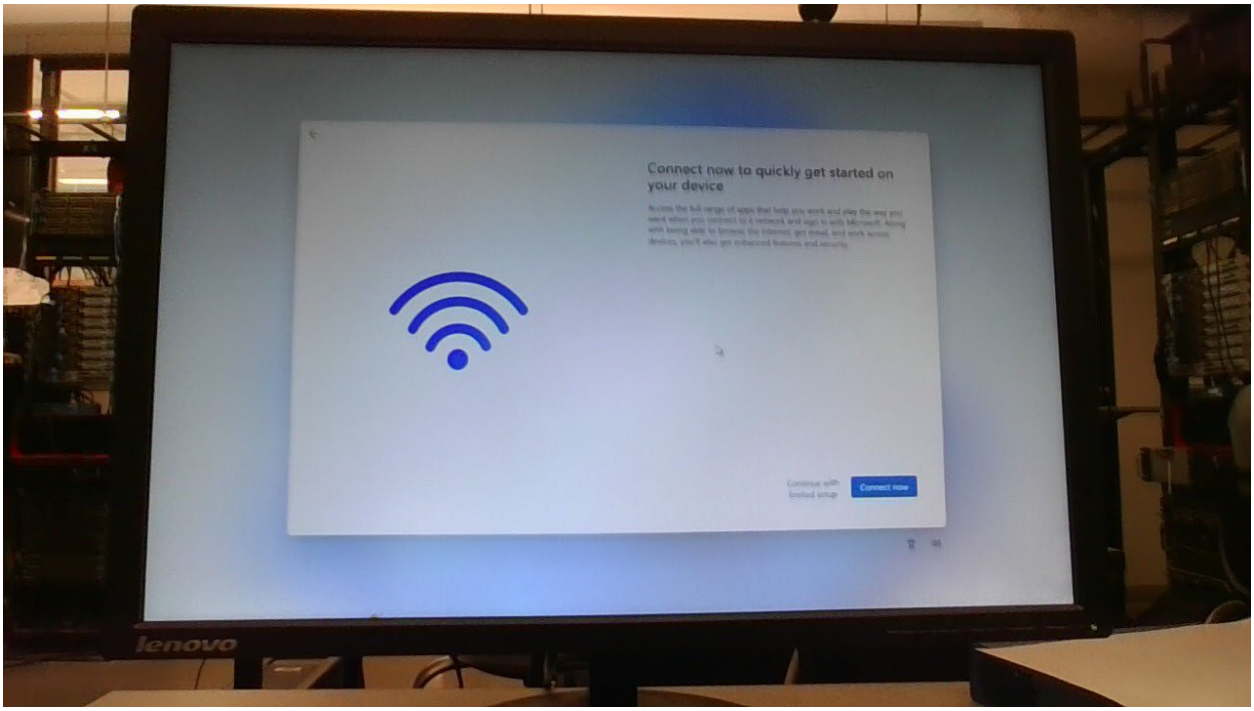
Click on “custom”.



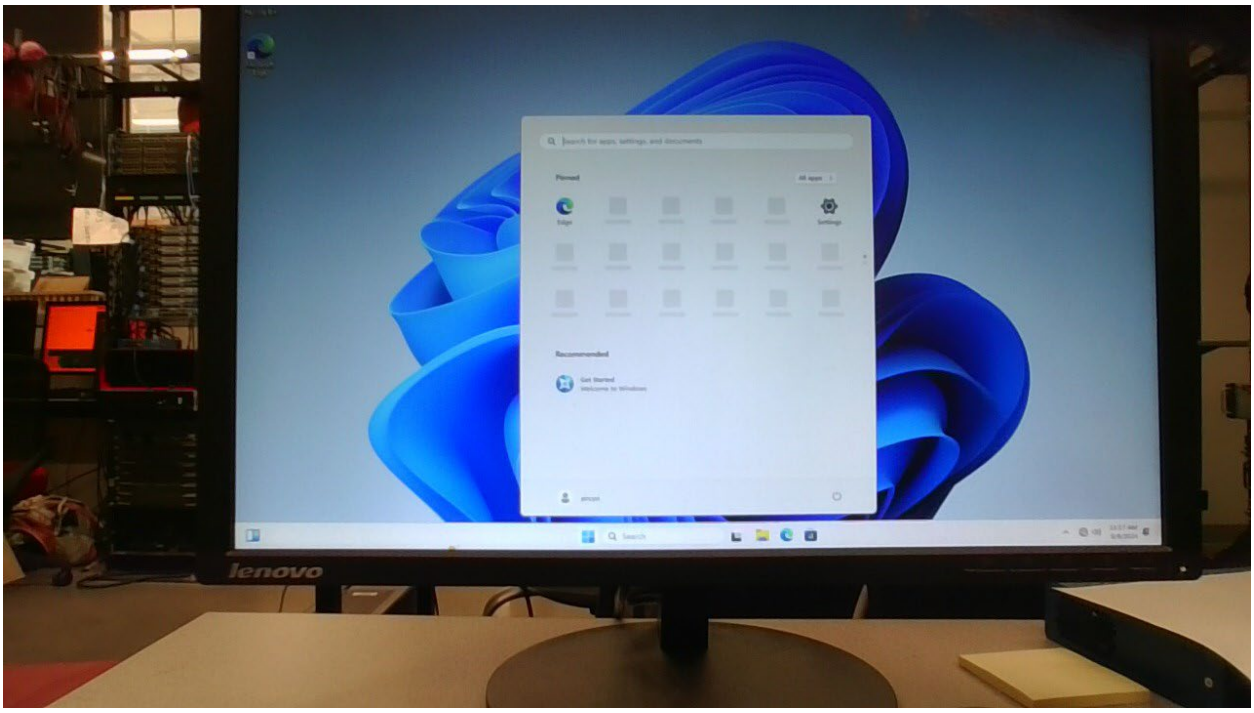
Windows is installing!



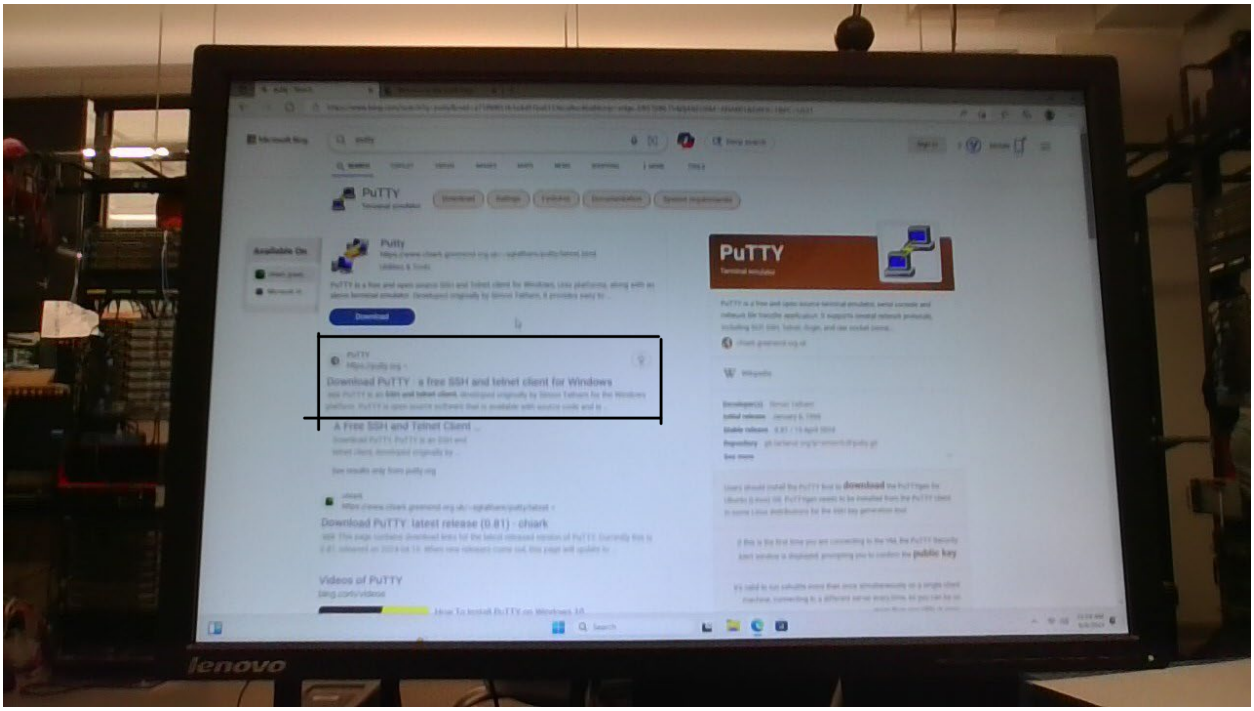
Do not connect to network. Click on “I don’t have internet” at the bottom right.



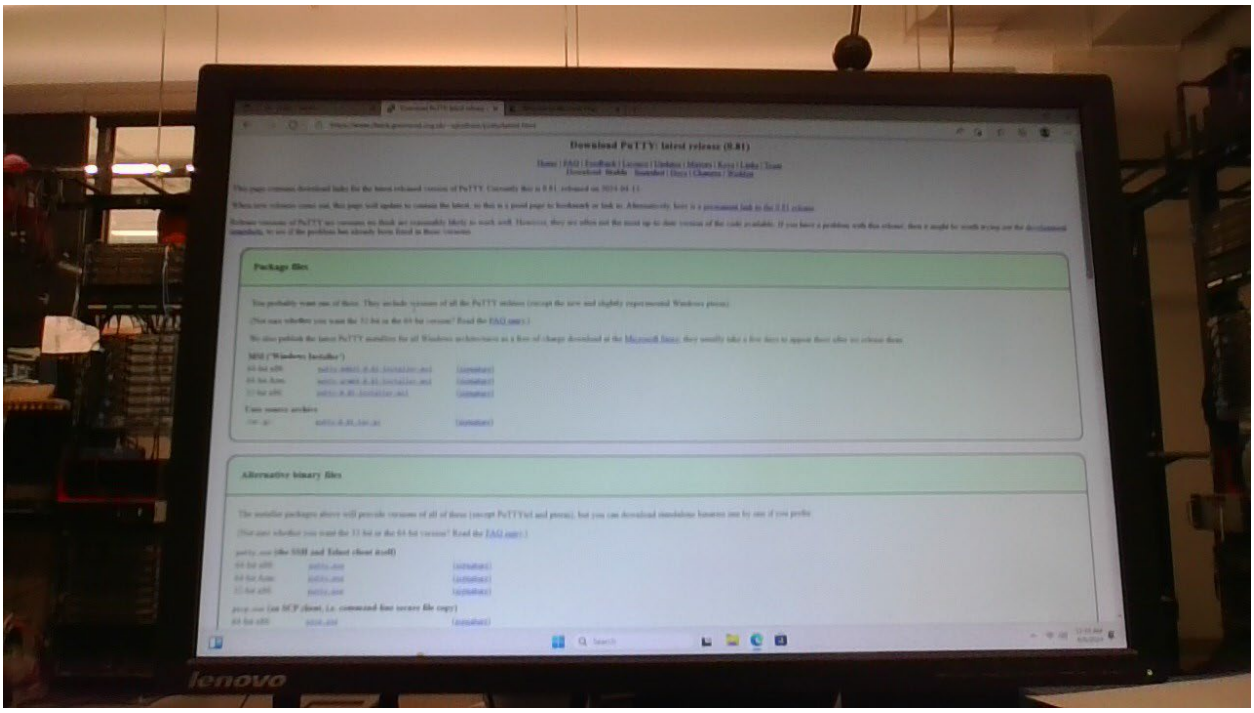
Click “continue with limited settings”.



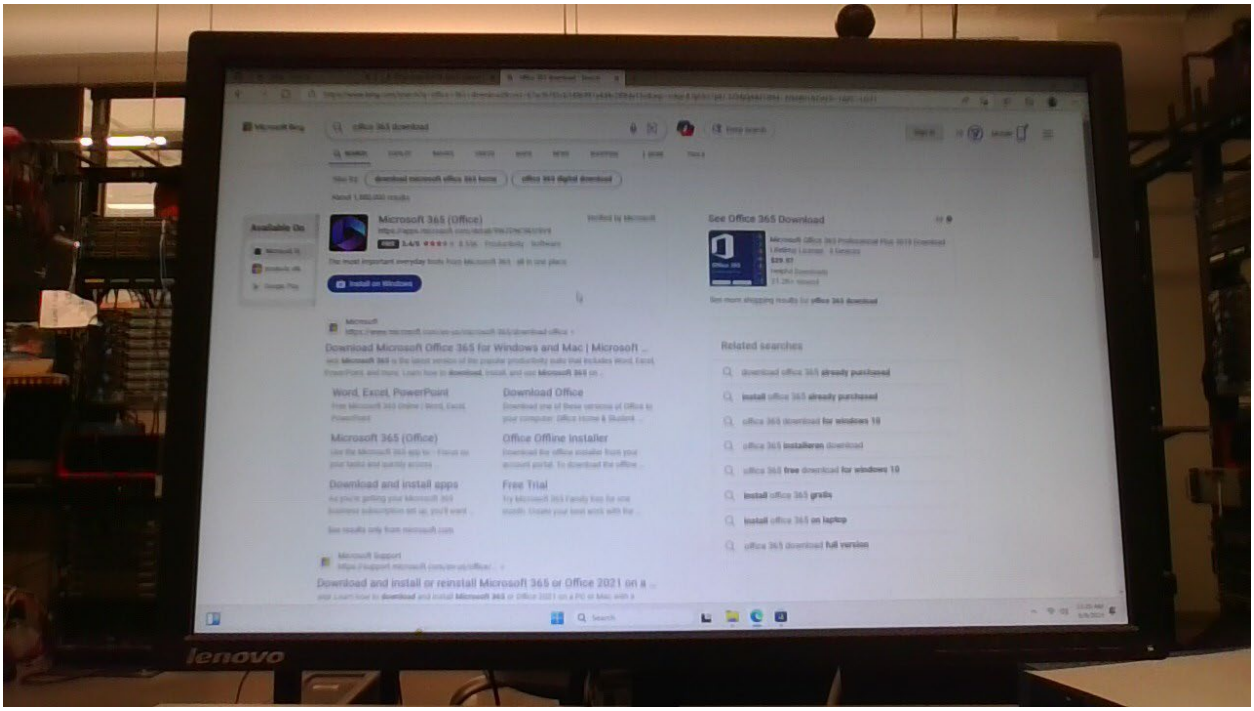
Your computer should start up now!



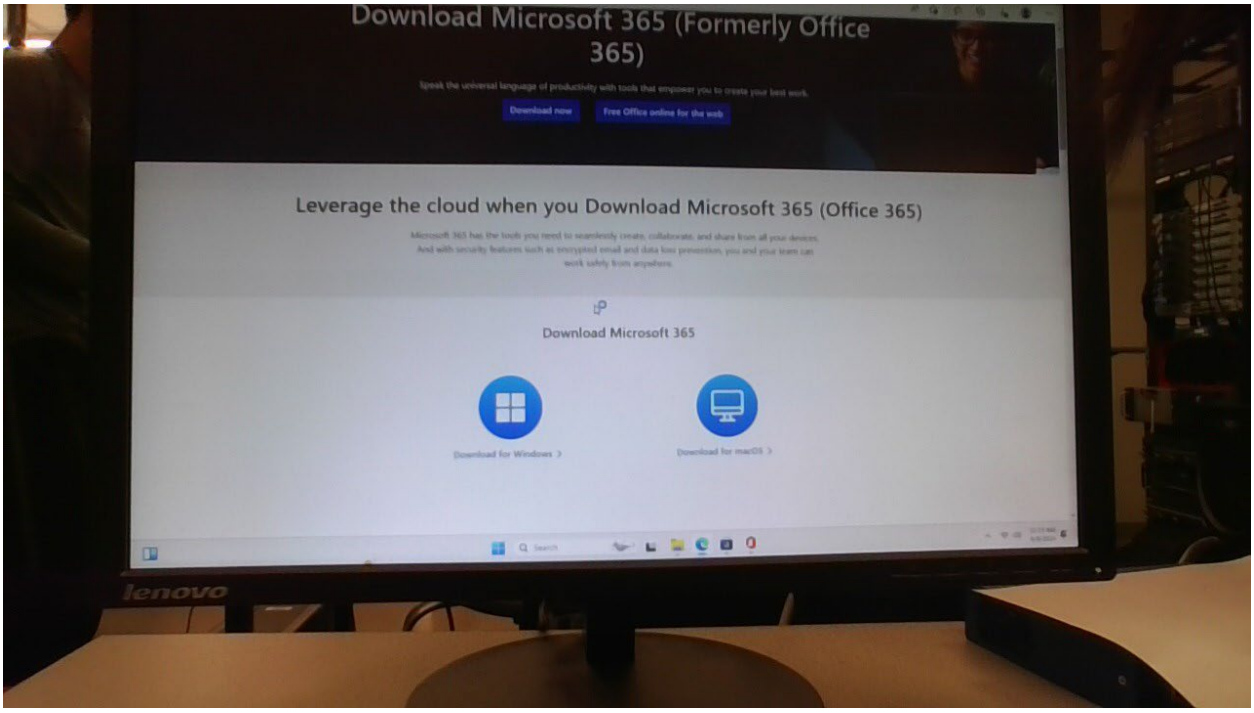
Open Microsoft Edge and search for “PuTTY”. Click on the boxed hyperlink.



Download the very first option on the webpage.

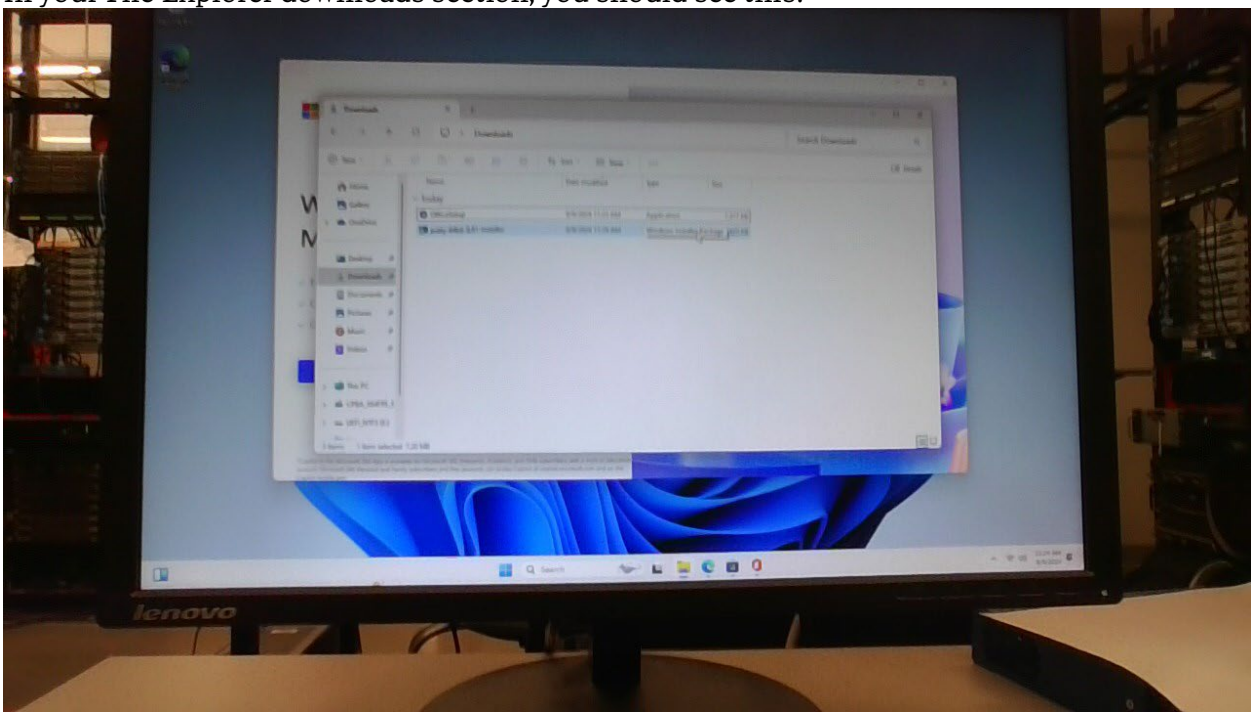


Search for Office 365 and download Microsoft Office 365 for Windows.

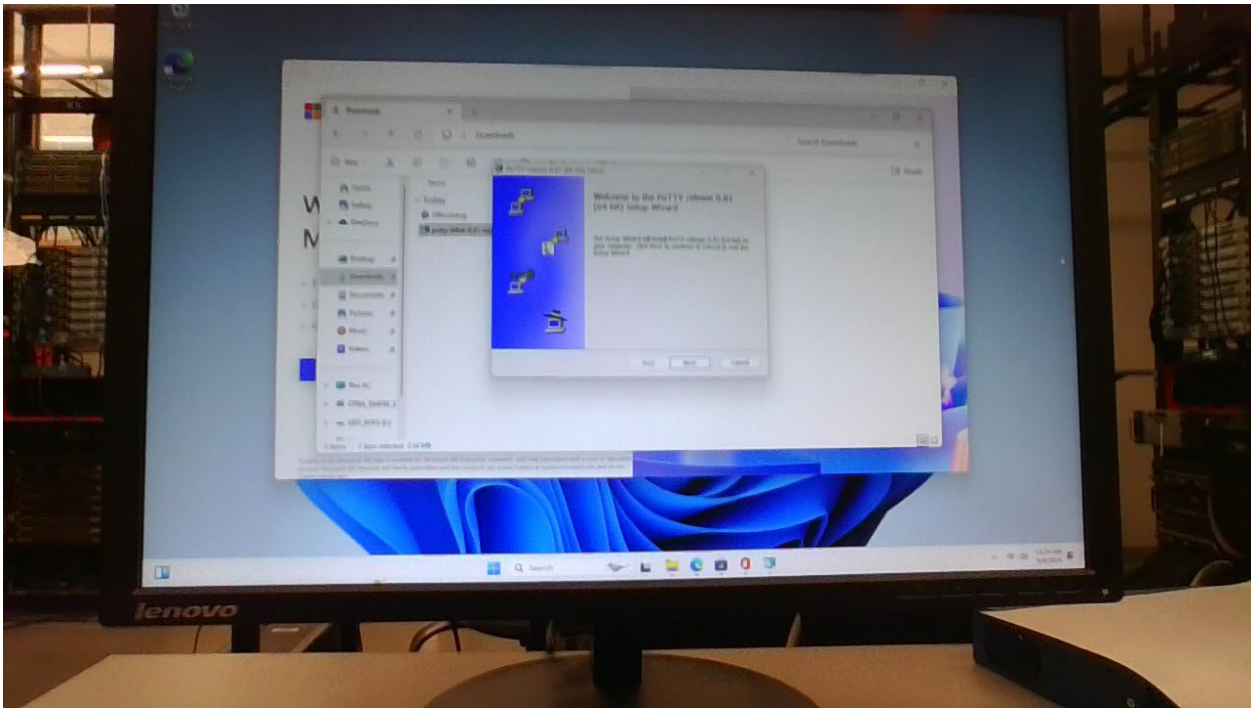


You should reach the page shown above.

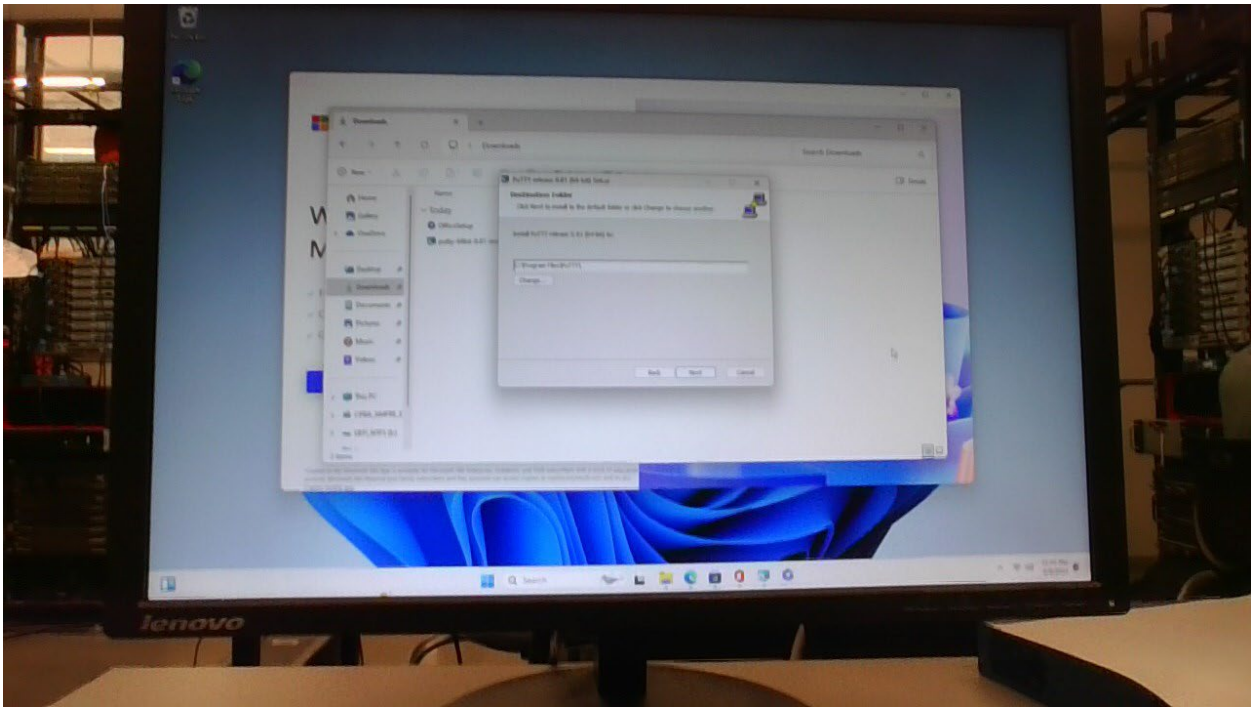
In your File Explorer downloads section, you should see this:



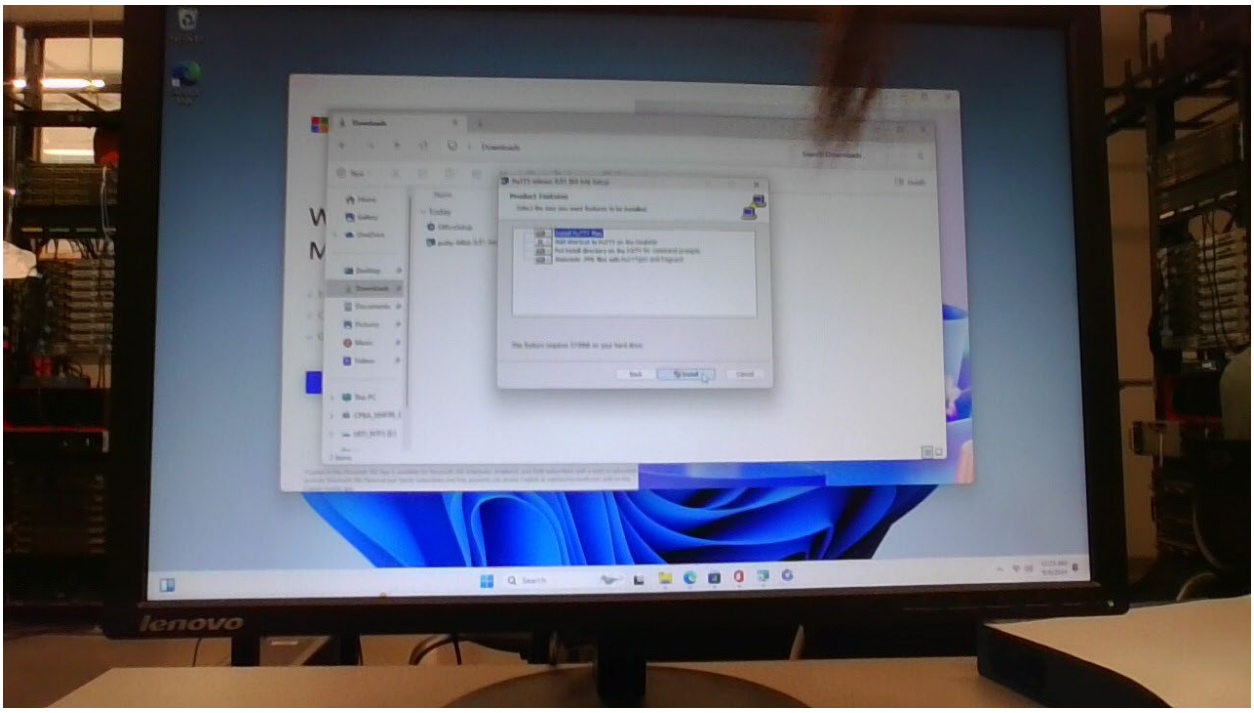
Open both files to start the setup process for PuTTY and Office 365.



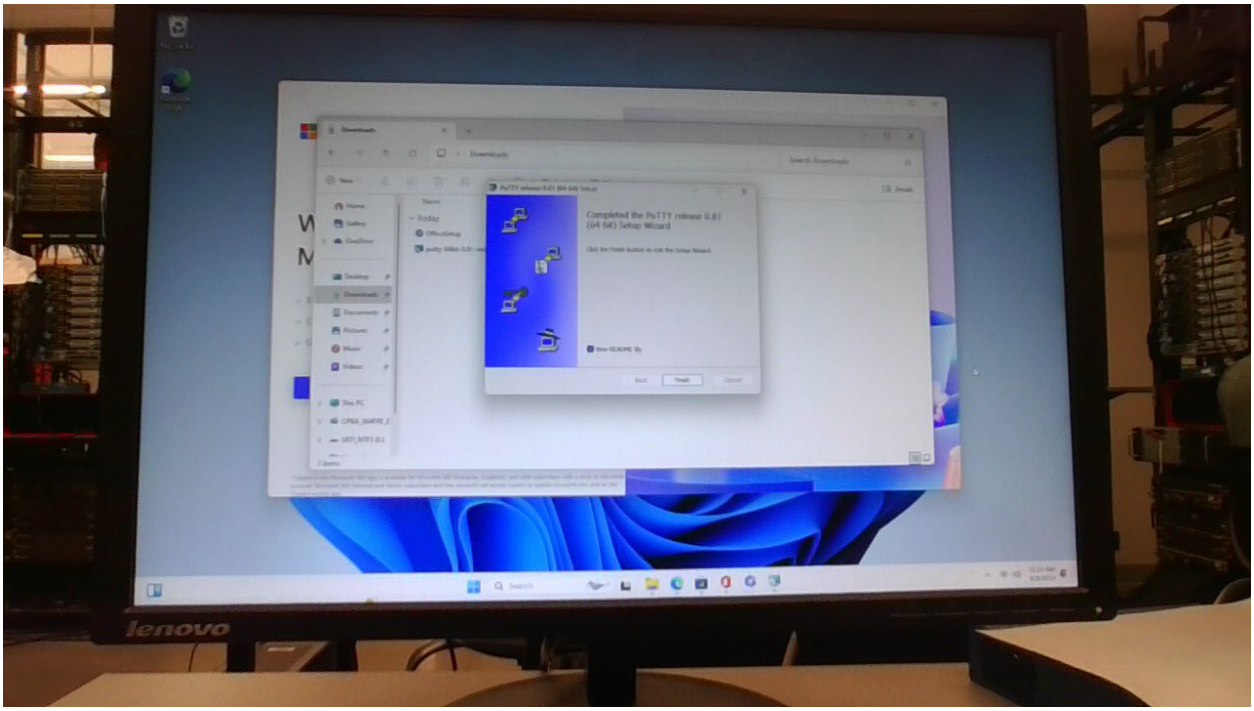
Click next.



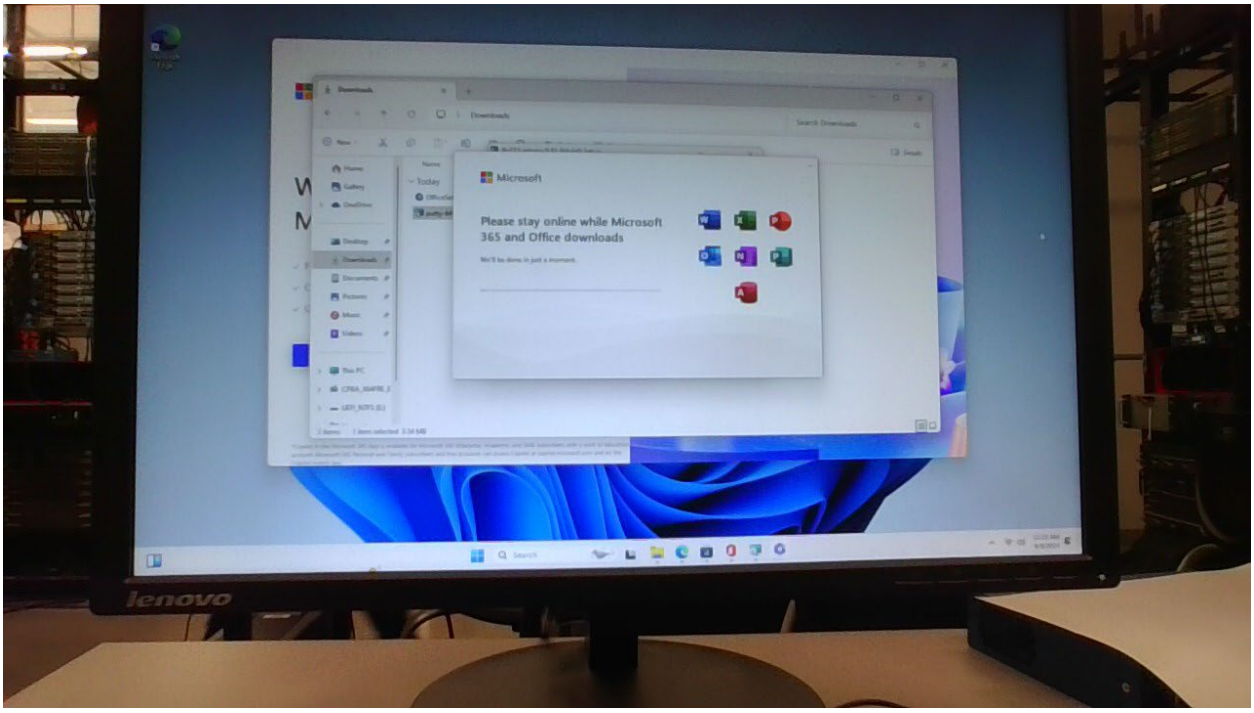
Click next.



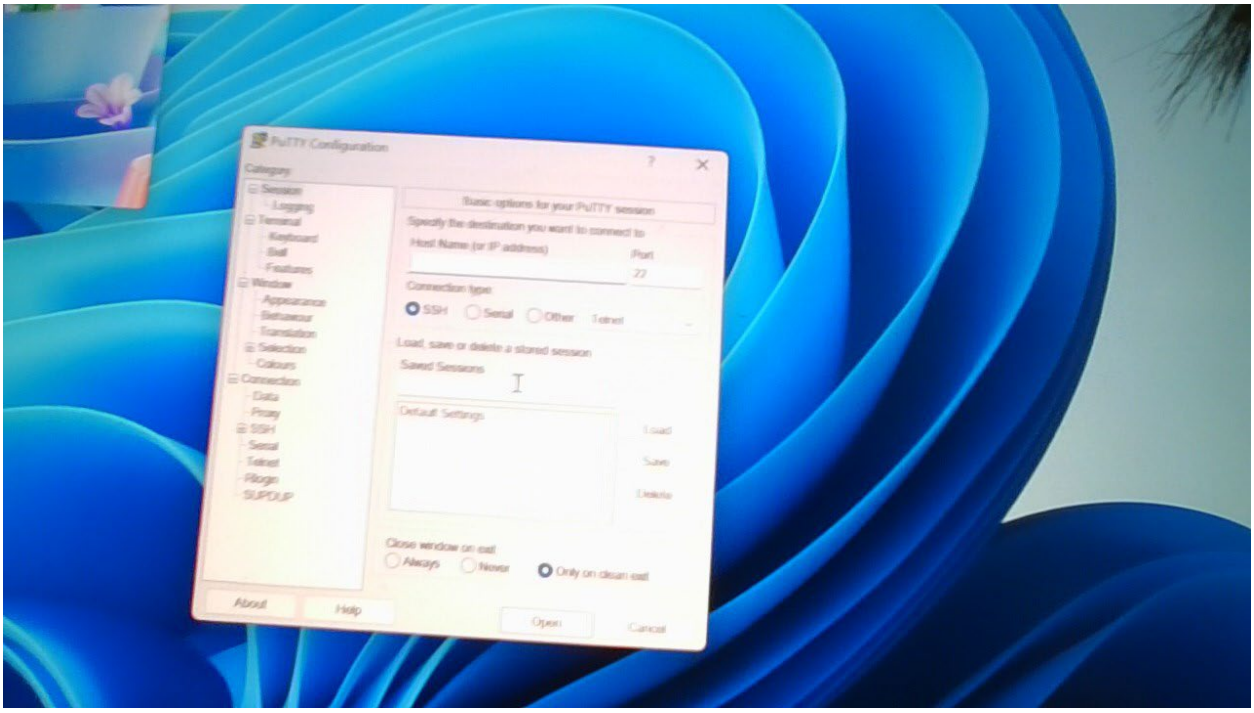
Click install.



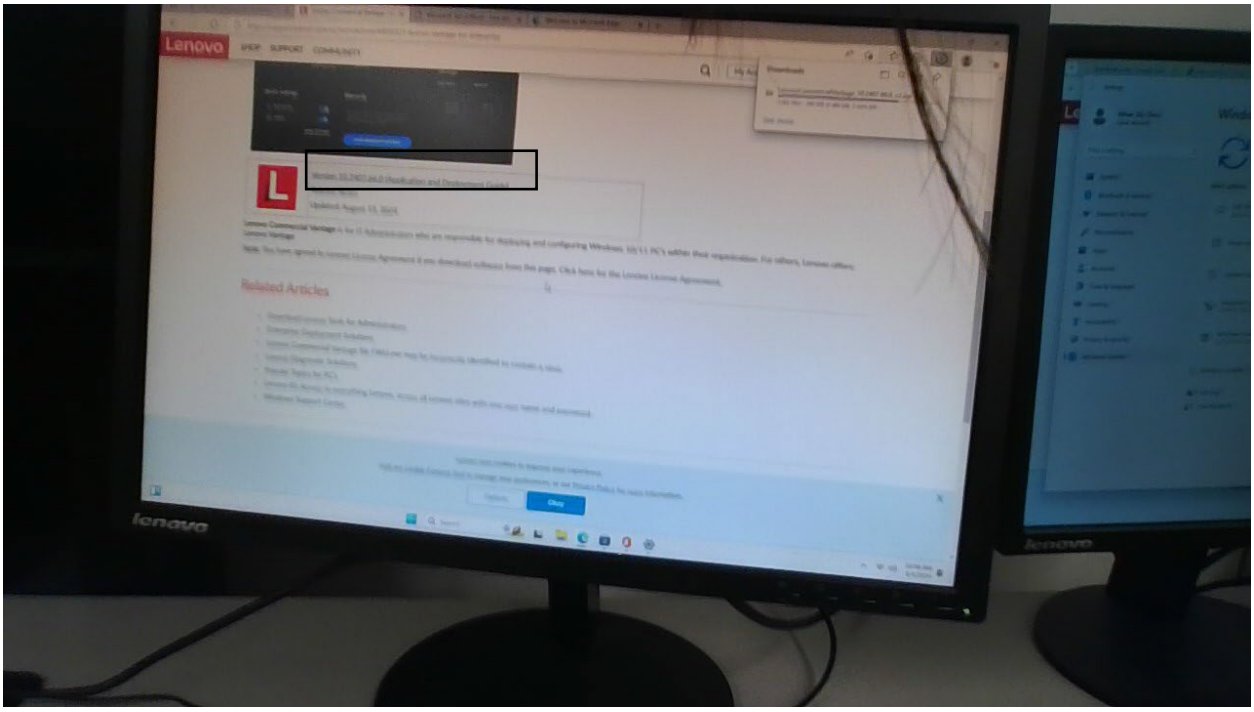
Click finish.



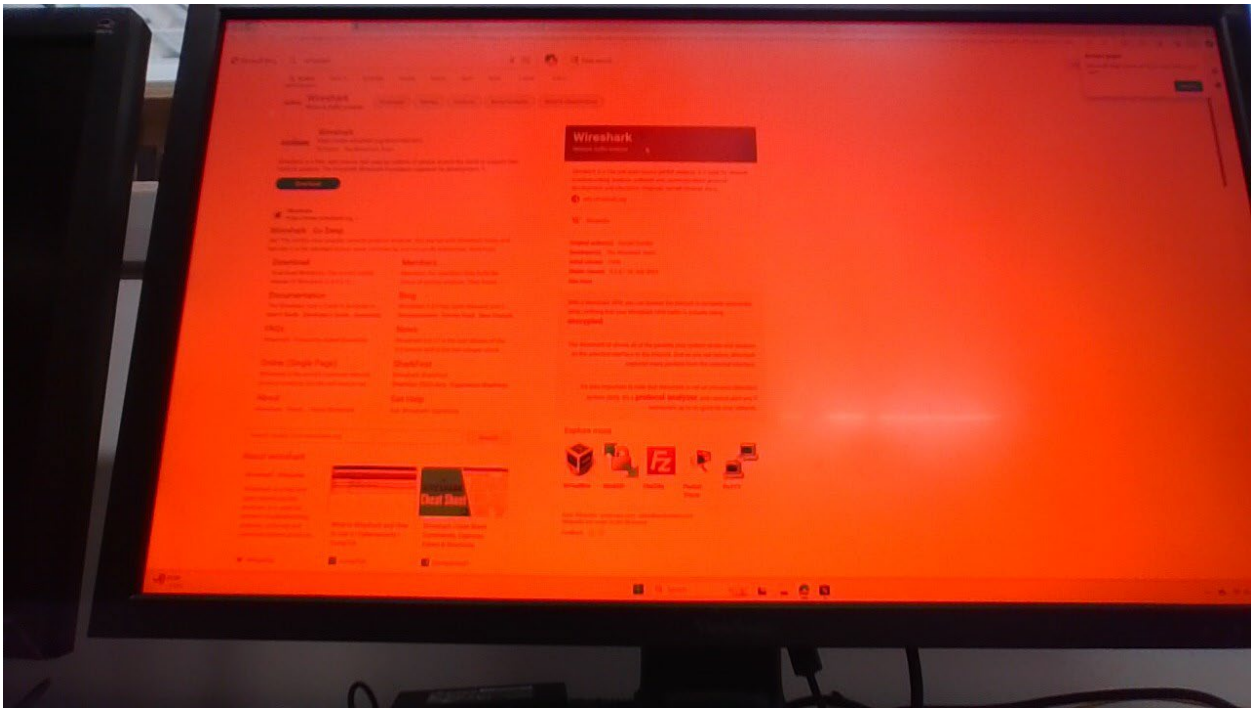
Wait for Office 265 to fully download. Once PuTTY and Office 365 are installed, click allow changes to the computer when prompted. Check to make sure you can make open the two applications, as shown below:

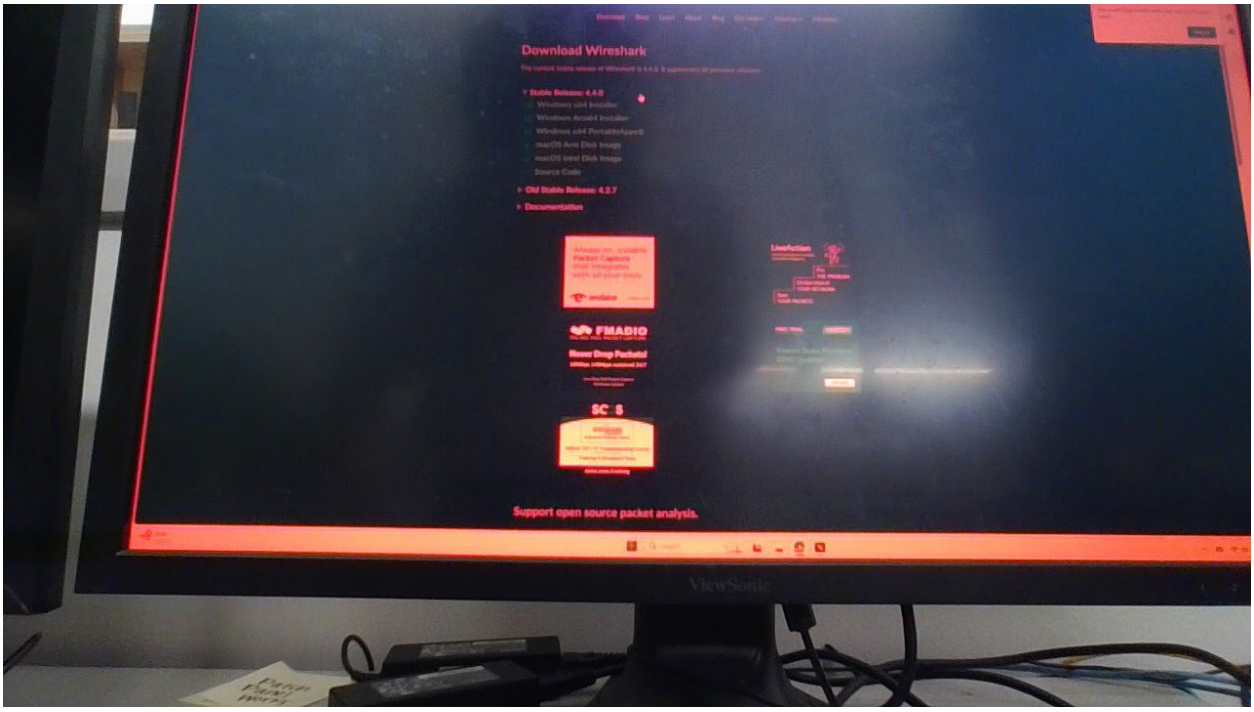


Next, search on Microsoft Edge for Lenovo Commercial Vantage. Install the application the same way PuTTY and Office 365 were installed.

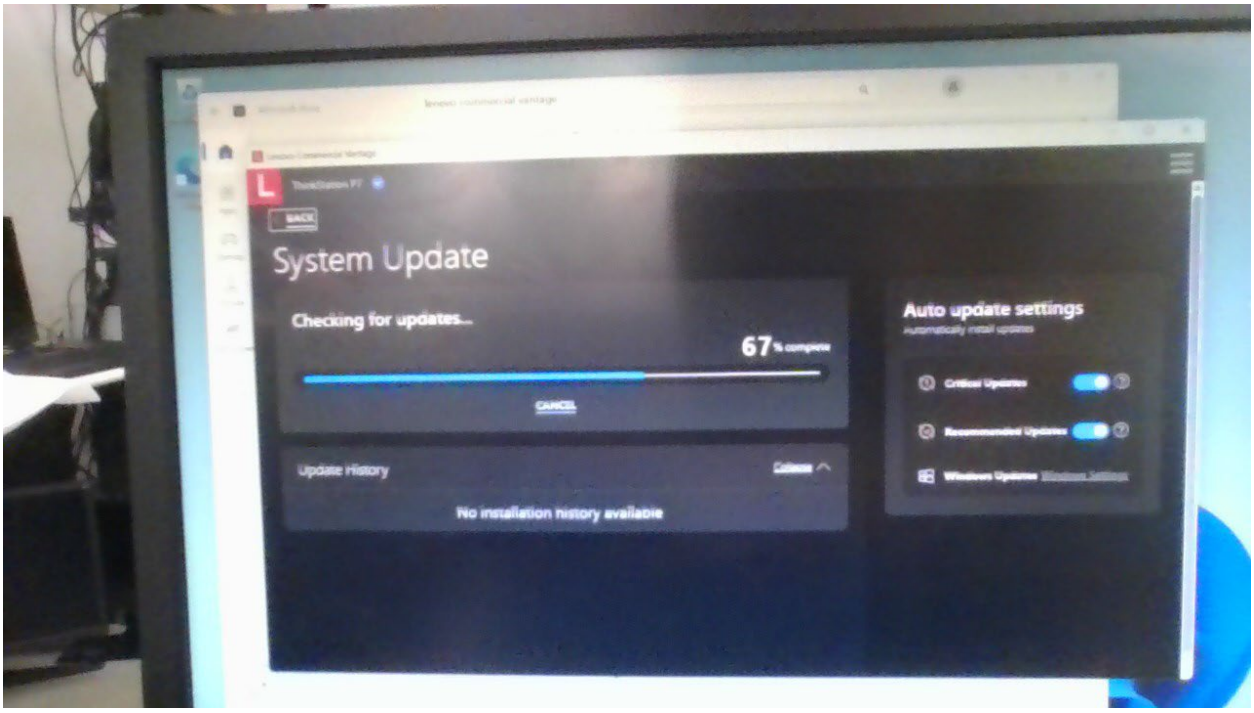


Afterwards, install Wireshark by searching for it on Microsoft Edge, clicking on the Windows x64 Installer option, opening the file, and clicking next for all options presented.

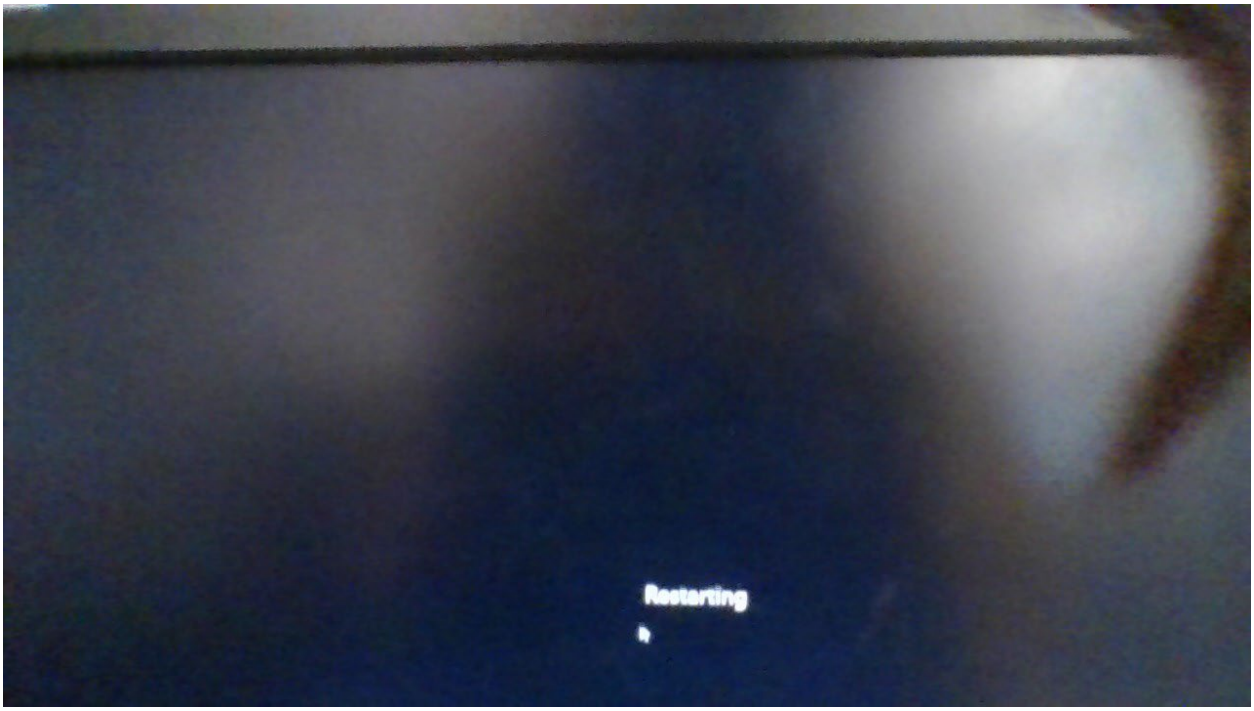




Click on the boxed hyperlink to start the download. However, I found downloading through Microsoft Store (which is already installed) to be much easier, so consider downloading through there instead.



Check for updates using Lenovo Commercial Vantage and install as necessary. You may need to restart.



PROBLEMS

To use a local Microsoft account, we had to select “I don’t have internet” option during the initial setup, which was confusing, and then later manually connect.

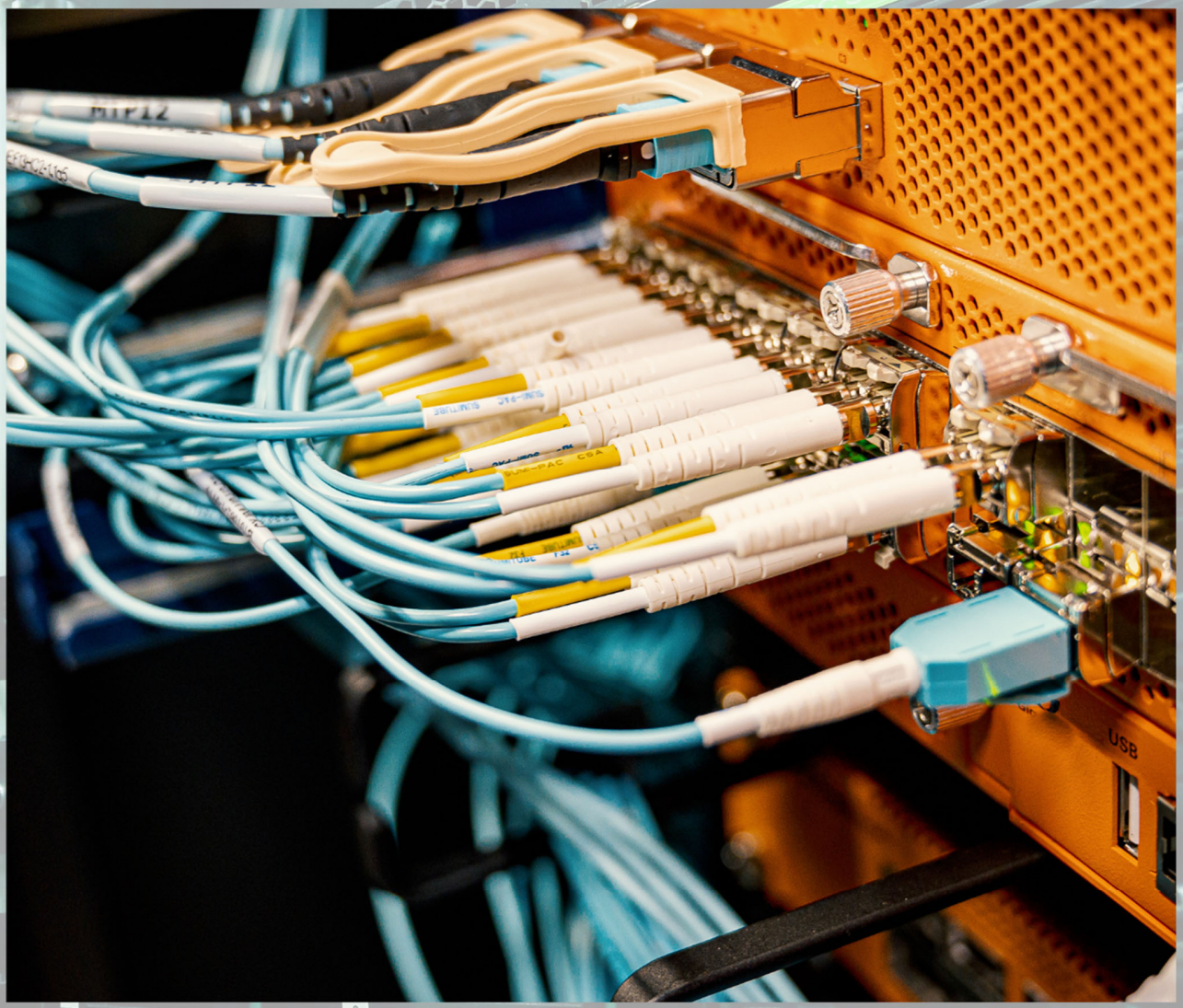
I found installing the Lenovo Commercial Vantage using Microsoft Edge to be confusing; however, using the Microsoft Store facilitated the process.

While updating, one of the desktops would not turn on after restarting, so I had to use another computer.

CONCLUSION

I learned how to setup a computer in its most basic form. However, there were some challenges, and I’m glad I was able to troubleshoot all of them.

Lab 2: Multi-Area OSPF



PURPOSE:

Review how to configure OSPFv2 and OSPFv3 (multiarea) and understand how to use a multi-layer switch as part of the configuration.

BACKGROUND INFORMATION ON LAB CONCEPTS:

OSPF is a routing protocol that uses the concept of areas. A network administrator can divide the routing domain into distinct areas that help control routing and update traffic, increasing efficiency.

OSPF, which stands for Open Shortest Path First, was developed as an alternative for the Routing Information Protocol (RIP). OSPF offers faster network convergence and can scale to much larger network implementations.

OSPF is able to detect changes in network topology within seconds. Using Dijkstra's algorithm, OSPF can compute the shortest path for each route. It calculates link metrics—helping judge the quality of potential links—based on distance of a router, amount of data, and link availability and reliability.

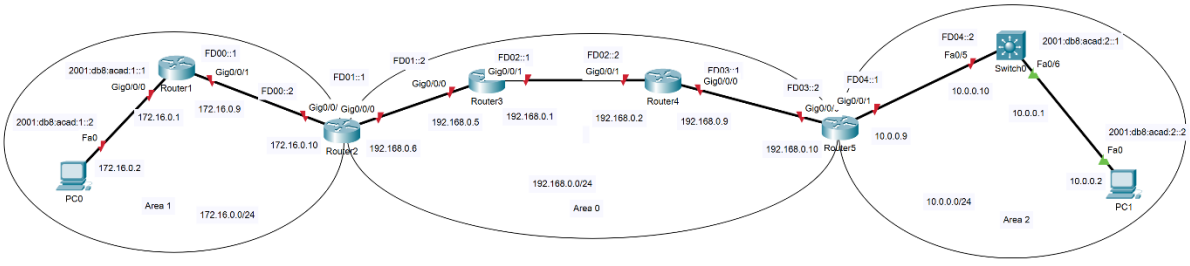
By convention, area 0 represents the backbone area of an OSPF network. Each additional area must have a connection to the OSPF backbone area.

OSPF does not use a transport protocol, unlike routing protocols like RIP and the Border Gateway Protocol. Instead, it encapsulates its data directly.

LAB SUMMARY:

In this lab, we configured multiarea OSPF, both for IPv4 and IPv6. We also were able to find out how to set up a multi-layer switch using OSPF.

NETWORK DIAGRAM:



LAB COMMANDS:

Follow the following steps for each routing device (router or multilayer switch) to configure the OSPF process ID, router ID, and interfaces.

MultilayerSwitch>en

MultilayerSwitch#config

MultilayerSwitch(config)#ip routing

MultilayerSwitch(config)#router ospf 1

This command enables OSPF with a process ID of 1.

MultilayerSwitch(config-router)#router-id 6.6.6.6

This command sets the router ID to 6.6.6.6.

MultilayerSwitch(config)#int fa0/5

For the multilayer switch, there is an additional step since we want the switch to be in routing, not switching, mode. As such, we have to use the following command:

MultilayerSwitch(config-if)#no switchport

Next, we continue by configuring the IP addresses and OSPF on each interface.

MultilayerSwitch(config-if)#ip add 10.0.0.10 255.255.255.252

MultilayerSwitch(config-if)#ip ospf 1 area 2

This command above configures the area in which the multilayer switch will belong to.

MultilayerSwitch(config-if)#exit

Repeat this process by configuring additional interfaces (using the no switchport command for interfaces on the multilayer switch).

We proceed by configuring the IPv6 addresses.

MultilayerSwitch(config)#ipv6 unicast-routing

MultilayerSwitch(config)#int f0/5

MultilayerSwitch(config-if)#ipv6 add fd04::2/64

MultilayerSwitch(config-if)#exit

Afterwards, we configure OSPFv3.

MultilayerSwitch(config)#ipv6 router ospf 1

MultilayerSwitch(config-rtr)#router-id 6.6.6.6

MultilayerSwitch(config-rtr)#exit

MultilayerSwitch(config)#int r fa0/5-6

MultilayerSwitch(config-if-range)#ipv6 ospf 1 area 2

We configure similarly for OSPFv2 and OSPFv3, on the routers:

Router 1 2001:db8:acad:1::1; 172.16.0.9

Router1>en

Router1#config

Router1(config)#ip routing

Router1(config)#router ospf 1

Router1(config-router)#router-id 1.1.1.1

Router1(config)#int g0/0/0

Router1(config-if)#ip add 172.16.0.1 255.255.255.252

Router1(config-if)#ip ospf 1 area 1

Router1(config)#int g0/0/1

Router1(config-if)#ip add 172.16.0.9 255.255.255.252

Router1(config-if)#ip ospf 1 area 1

Router1(config-if)#exit

Router1(config)#ipv6 router ospf 1

Router1(config-rtr)#router-id 1.1.1.1

Router1(config-rtr)#exit

Router1(config)#ipv6 unicast-routing

Router1(config)#int g0/0/0

Router1(config-if)#ipv6 add 2001:db8:acad:1::1/64

Router1(config-if)#ipv6 ospf 1 area 1

Router1(config)#int g0/0/1

Router1(config-if)#ipv6 add fd00::1/64

Router1(config-if)#ipv6 ospf 1 area 1

Repeat the above for routers R2, R3, R4, and R5.

Finally, configure PC0 and PC1's IP and IPv6 addresses. Test for connectivity and troubleshoot as needed.

CONFIGURATION:

Note that certain output has been removed for clarity and relevance.

Router 1:

```
ipv6 unicast-routing

interface GigabitEthernet0/0/0
ip address 192.168.1.1 255.255.255.0
negotiation auto
ipv6 address 2001:DB8:ACAD:F::2/64
ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
ip address 192.168.0.1 255.255.255.0
negotiation auto
ipv6 address 2001:DB8:ACAD:A::2/64
ipv6 ospf 1 area 1

router ospfv3 1
router-id 0.0.0.1
address-family ipv6 unicast
exit-address-family

router ospf 1
router-id 0.0.0.1
network 192.168.0.0 0.0.0.255 area 1
network 192.168.1.0 0.0.0.255 area 0
```

Router 2:

```
ipv6 unicast-routing

interface GigabitEthernet0/0/0
ip address 192.168.1.2 255.255.255.0
negotiation auto
ipv6 address 2001:DB8:ACAD:B::1/64
ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
```

```
ip address 192.168.0.2 255.255.255.0
negotiation auto
ipv6 address 2001:DB8:ACAD:A::2/64
ipv6 ospf 1 area 1

router ospfv3 1
router-id 0.0.0.2
address-family ipv6 unicast
exit-address-family

router ospf 1
router-id 0.0.0.2
network 192.168.0.0 0.0.0.255 area 1
```

Router 3:

```
ipv6 unicast-routing

interface GigabitEthernet0/0/0
ip address 192.168.0.5 255.255.255.252
negotiation auto
ipv6 address FD01::2/64
ipv6 enable
ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
ip address 192.168.0.1 255.255.255.252
negotiation auto
ipv6 address FD02::1/64
ipv6 enable
ipv6 ospf 1 area 0

router ospf 1
router-id 0.0.0.3
network 192.168.0.0 0.0.0.255 area 0
```

Router 4:

```
ipv6 unicast-routing
```

```
interface Loopback0
ip address 4.4.4.4 255.255.255.255

interface GigabitEthernet0/0/0
ip address 192.168.0.9 255.255.255.252
negotiation auto
ipv6 address FD03::1/64

interface GigabitEthernet0/0/1
ip address 192.168.0.2 255.255.255.252
negotiation auto
ipv6 address FD02::2/64

router ospf 1
router-id 4.4.4.4
network 192.168.0.0 0.0.0.255 area 0
```

Router 5:

```
ipv6 unicast-routing

interface Loopback0
ip address 5.5.5.5 255.255.255.255

interface GigabitEthernet0/0/0
ip address 192.168.0.10 255.255.255.252
negotiation auto
ipv6 address FD03::2/64

interface GigabitEthernet0/0/1
ip address 10.0.0.9 255.255.255.0
negotiation auto
ipv6 address FD04::1/64

router ospf 1
router-id 5.5.5.5
network 10.0.0.0 0.0.0.255 area 2
network 192.168.0.0 0.0.0.255 area 0
```

Multi-Layer Switch:

```
ipv6 unicast-routing

interface FastEthernet0/5
no switchport
ip address 10.0.0.10 255.255.255.252
ipv6 address FD04::2/64
ipv6 ospf 1 area 2

interface FastEthernet0/6
no switchport
ip address 10.0.0.1 255.255.255.252
ipv6 address 2001:DB8:ACAD:2::1/64
ipv6 ospf 1 area 2

interface Vlan1
ip address 192.168.10.1 255.255.255.0
router ospf 1
router-id 6.6.6.6
log-adjacency-changes
network 10.0.0.0 0.0.0.255 area 2

ipv6 router ospf 1
router-id 6.6.6.6
```

PROBLEMS:

We encountered many problems during the configuration.

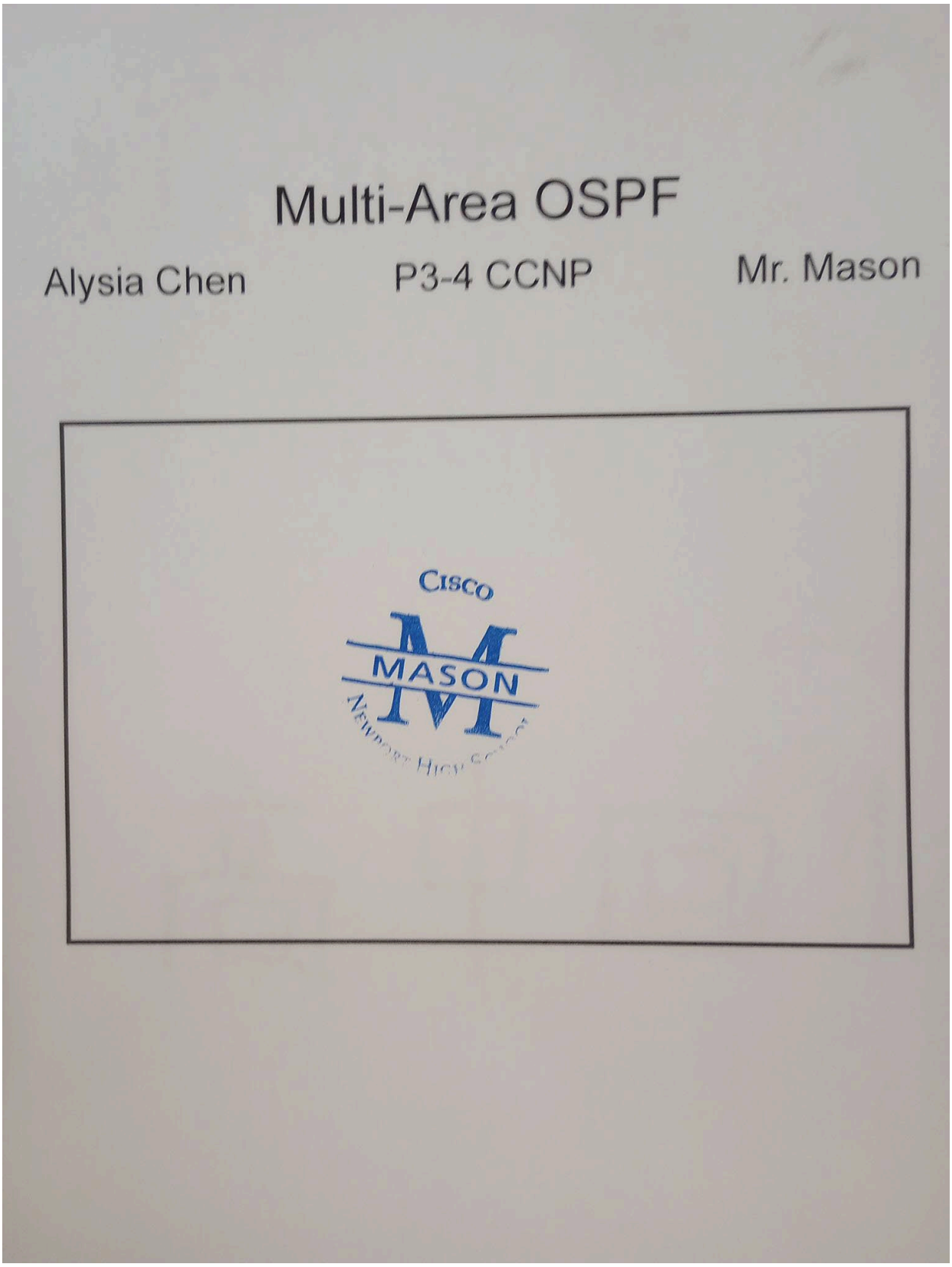
Firstly, there was a hardware issue on our original multilayer switch that prevented it from connecting to the network. When we tried using another multilayer switch, the incorrect Cisco IOS version was installed. For our lab configuration to work, we needed the IP Services version, but we originally started with the IP Base version. Therefore, to install IP Services, we had to borrow another multilayer switch to transfer and download the correct Cisco IOS version.

However, there were a few problems during the process. Firstly, transferring the IOS file directly via a USB drive didn't work. Secondly, transferring directly from one TFTP server to our TFTP server didn't work either. Lastly, we tried transferring from the multilayer switch with the correct version installed to the PC we borrowed to our PC. This was done through TFTPD64, which sets up the server. The last solution worked, and we were able to configure our multilayer switch smoothly and finish our lab.

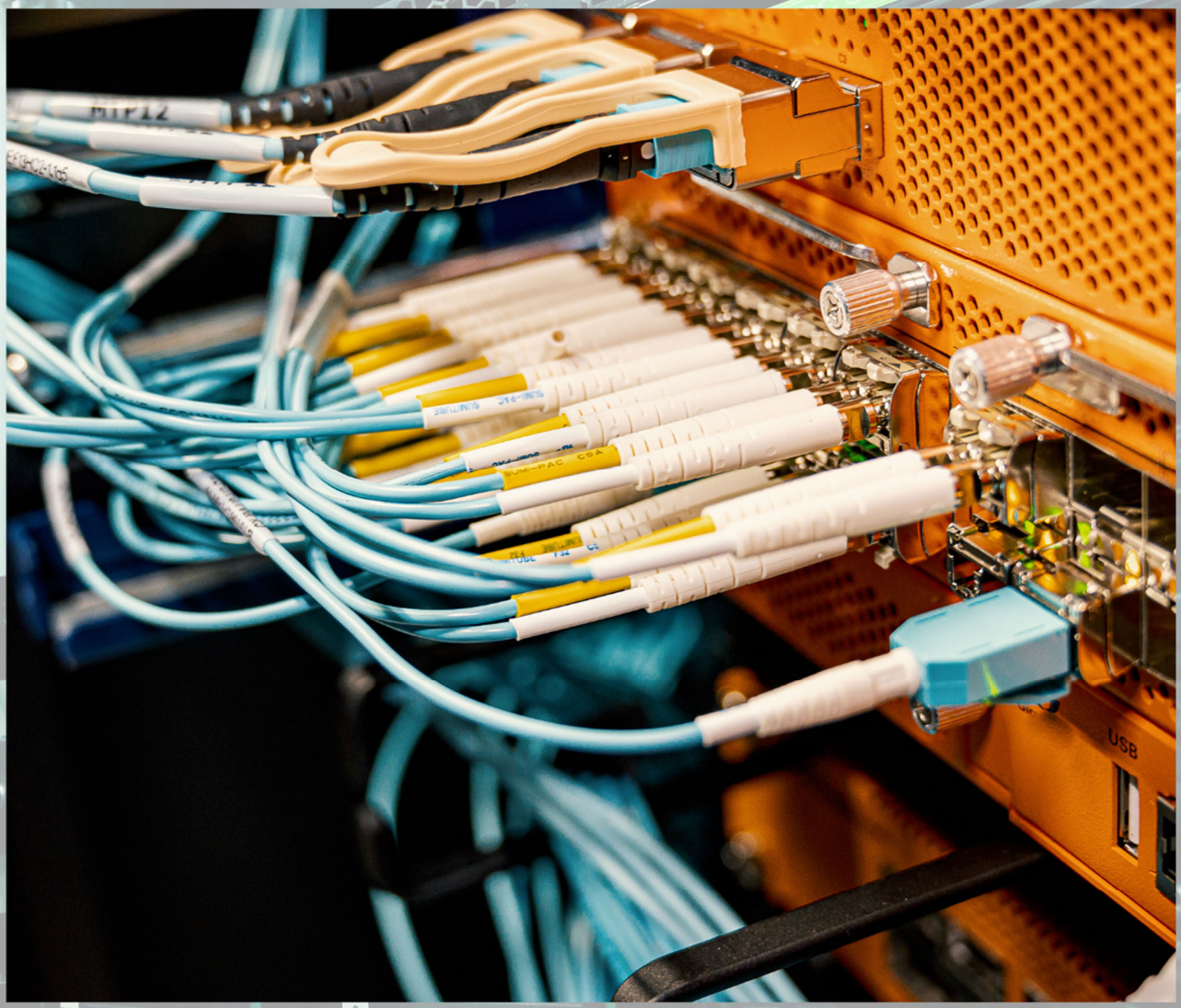
CONCLUSION:

I learned how to set up OSPF, both version 2 and version 3, as well as on a multilayer switch. However, there were lots of challenges along the way, and I'm glad I was able to troubleshoot and be able to finish the lab successfully.

TEACHER SIGN-OFF:



Lab 3: eBGP



PURPOSE:

Learn to configure the external Border Gateway Protocol, while using OSPF, EIGRP, and IS-IS as the routing protocols within each autonomous system.

BACKGROUND INFORMATION ON LAB CONCEPTS:

eBGP:

eBGP is also known as external Border Gateway Protocol (BGP). It uses autonomous systems (ASes), which are the networks within the network using eBGP.

BGP finds the most efficient path through attributes like weight (which local paths are preferred), local preference (which outbound path), and AS path length (router should prefer shorter paths).

BGP is used to connect networks worldwide to the Internet. In this lab, however, we only use it to connect IS-IS to OSPF to EIGRP. Some potential flaws of BGP are due to how wide-scale it is, routes being accidentally advertised or accidentally becoming the preferred paths may cause a widespread Internet outage, but these disruptions may take a while to fix due to a slow convergence.

OSPF:

OSPF stands for Open Shortest Path First. It is a link-state routing protocol, meaning that it only sends updates when a change occurs in the routing table and that all routers in the protocol know the full topology.

OSPF is able to detect changes in network topology within seconds, calculating link metrics—which help judge the quality of potential links—based on amount of data, distance of a router, and link availability and reliability.

EIGRP:

EIRGRP, or Enhanced Interior Gateway Routing Protocol, allows for faster convergence than networks like BGP. It calculates routes through evaluating the distance and whether a destination path has loops or not.

This routing protocol has both link-state and distance vector characteristics—an EIGRP router only sends updates when a change occurs in the routing table but it also only advertises its best route to a neighbor (which is more like distance vector) and does not know the entire topology.

IS-IS:

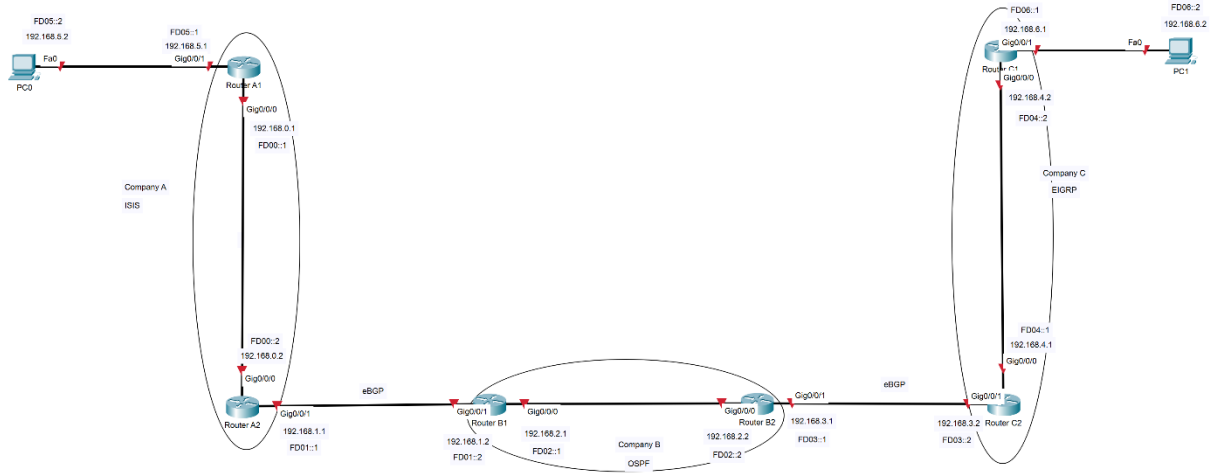
IS-IS (Intermediate System to Intermediate System) is designed to be an interior gateway protocol. It is also a link-state routing protocol and uses the same algorithm as OSPF to find the best path.

However, IS-IS is different from OSPF in that there are three types of routers used to connect different areas together: Level 1, Level 1-2, and Level 2. Level 1 can only establish neighbor adjacencies for routers in the same area; Level 2 can only establish neighbor adjacencies for routers in different areas; and Level 1-2 and perform both functions.

LAB SUMMARY:

In this lab, we configured external BGP, both for IPv4 and IPv6, to run through networks with IS-IS, OSPF, and EIGRP. We ended up connecting the two end devices—one on the network with IS-IS and one on the network with EIGRP—after successful configuration.

NETWORK DIAGRAM:



LAB COMMANDS:

First, after configuring network addresses for interfaces on each router, set up IS-IS, OSPF, and EIGRP, respectively:

IS-IS:

ip router isis enables an interface to participate in IS-IS for IPv4; *ipv6 router isis* enables an interface to participate in IS-IS for IPv6.

Since the routers will not use different areas, both of the routers (A1 and A2) will be Level 1. Enter the command *router isis* and use the global command *is-type level-1* to configure this. Specify a Network Entity Title (NET), which is the "ID" of the router in IS-IS. For example: *net 49.0001.1920.1680.0001.00.*

Then, use *redistribute connected* and *redistribute bgp [chosen Autonomous System number] [router level]*, noting to only use these commands for routers that are directly connected to a router with a different router protocol. Go into *address-family ipv6* and configure the same. These commands are used to advertised routes learned through BGP and directly connected routes into the IS-IS domain, but configuring them in this step will save some time.

OSPF:

ip ospf 1 area 0 configures the OSPF process ID (1) and the area (0, also known as the backbone area) for an interface. Process IDs are only for local identification (within the router); since only one area is needed for this lab, Area 0 is used. Repeat for IPv6, replacing *ip ospf 1 area 0* with *ipv6 ospf 1 area 0.*

In *router ospf 1*, where "1" represents the process ID, configure network statements to represent the different networks covered by OSPF, like: *network 192.168.2.0 0.0.0.255 area 0.* Next, configure the router ID, which identifies the router in the OSPF network. Do the same for IPv6, replacing *router ospf 1* with *ipv6 router ospf 1.*

Finally, use *redistribute connected* and *redistribute bgp [chosen Autonomous System number]*, noting to only use these commands for routers that are directly connected to a router with a different router protocol. These commands are used to advertised routes learned through BGP and directly connected routes into the OSPF domain, but configuring them in this step will save some time.

EIGRP:

Enter the *router eigrp 3* command to configure globally, where "3" is the chosen Autonomous System number. Configure the default metric values for routes that are redistributed through EIGRP. An example is *default-metric 10000 100 255 1 1500,* where 10000 represents the bandwidth, 100 represents the delay, 255 represents the reliability (highest reliability), 1 is the load (least load), and 1500 is the MTU.

Next, configure the network statements for the networks that the EIGRP routing protocol will cover. An example is *network 192.168.4.0 0.0.0.3.* Then, use *redistribute connected* and *redistribute bgp [chosen Autonomous System number]*, noting to only use these commands for routers that are directly connected to a router with a different router protocol. These commands are used to advertised routes learned through BGP and directly connected routes into the EIGRP domain, but configuring them in this step will save some time.

Then, type in the command *ipv6 router eigrp 3* to configure the router ID as such: *eigrp router-id 6.6.6.6*; also configure the *redistribute* commands. Configure the default metric values for IPv6 as well; the same values may be used for IPv4 and IPv6.

Configure interfaces using the command *ipv6 eigrp 3* for IPv6 only on links that are part of the EIGRP Autonomous System (AS 3).

Next, follow the following steps for each routing device to configure **eBGP**.

Enter the BGP router configuration with the global command *router bgp [router Autonomous System number].* Configure neighbors using commands like *neighbor 192.168.3.1 remote-as 3333* for IPv4 and *neighbor FD03::1 remote-as 3333* for IPv6. Next, enter the commands *address-family ipv4* and *address-family ipv6*, respectively, to configure IPv4 and IPv6 BGP. In both, enter *redistribute connected* and *redistribute [either IS-IS, OSPF, or EIGRP, depending on the router, with the associated*

router level, process ID, or Autonomous System number, respectively]. If needed, specific networks may be advertised using the *network [IPv4 address/IPv6 address]* command. This will allow for connected routes and routes learned through EIGRP to be advertised through BGP and make them visible across the entire network.

CONFIGURATIONS:

A1:

```
IPv4 Routing Table:
      4.0.0.0/32 is subnetted, 1 subnets
i L1     4.4.4.4 [115/10] via 192.168.0.2, 00:12:43,
GigabitEthernet0/0/0
      5.0.0.0/32 is subnetted, 1 subnets
i L1     5.5.5.5 [115/10] via 192.168.0.2, 00:12:43,
GigabitEthernet0/0/0
      6.0.0.0/32 is subnetted, 1 subnets
i L1     6.6.6.6 [115/10] via 192.168.0.2, 00:08:10,
GigabitEthernet0/0/0
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.0.1/32 is directly connected,
GigabitEthernet0/0/0
i L1  192.168.1.0/24 [115/20] via 192.168.0.2, 00:15:03,
GigabitEthernet0/0/0
i L1  192.168.2.0/24 [115/10] via 192.168.0.2, 00:13:32,
GigabitEthernet0/0/0
      192.168.3.0/30 is subnetted, 1 subnets
i L1     192.168.3.0 [115/10] via 192.168.0.2, 00:12:43,
GigabitEthernet0/0/0
      192.168.4.0/30 is subnetted, 1 subnets
i L1     192.168.4.0 [115/10] via 192.168.0.2, 00:08:40,
GigabitEthernet0/0/0
      192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.5.0/24 is directly connected,
GigabitEthernet0/0/1
L        192.168.5.1/32 is directly connected,
GigabitEthernet0/0/1
      192.168.6.0/30 is subnetted, 1 subnets
i L1     192.168.6.0 [115/10] via 192.168.0.2, 00:08:10,
GigabitEthernet0/0/0
```

```
IPv6 Routing Table:
C   FD00::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   FD00::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
I1  FD01::/64 [115/20]
     via FE80::227:90FF:FED4:F30, GigabitEthernet0/0/0
I1  FD02::/64 [115/10]
     via FE80::227:90FF:FED4:F30, GigabitEthernet0/0/0
I1  FD03::/64 [115/10]
     via FE80::227:90FF:FED4:F30, GigabitEthernet0/0/0
I1  FD04::/64 [115/10]
     via FE80::227:90FF:FED4:F30, GigabitEthernet0/0/0
C   FD05::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   FD05::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
I1  FD06::/64 [115/10]
     via FE80::227:90FF:FED4:F30, GigabitEthernet0/0/0
L   FF00::/8 [0/0]
     via Null0, receive
```

Configuration:

```
ipv6 unicast-routing

interface GigabitEthernet0/0/0
 ip address 192.168.0.1 255.255.255.0
 ip router isis
 negotiation auto
 ipv6 address FD00::1/64
 ipv6 router isis
 isis circuit-type level-1

interface GigabitEthernet0/0/1
 ip address 192.168.5.1 255.255.255.0
 ip router isis
 negotiation auto
 ipv6 address FD05::1/64
 ipv6 router isis
```

```
 isis circuit-type level-1

router isis
 net 49.0001.1920.1680.0001.00
 is-type level-1
 metric-style narrow
 redistribute connected
```

A2:

```
IPv4 Routing Table:
      4.0.0.0/32 is subnetted, 1 subnets
B        4.4.4.4 [20/20] via 192.168.1.2, 00:11:57
      5.0.0.0/32 is subnetted, 1 subnets
B        5.5.5.5 [20/1] via 192.168.1.2, 00:11:57
      6.0.0.0/32 is subnetted, 1 subnets
B        6.6.6.6 [20/1] via 192.168.1.2, 00:07:24
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.0.2/32 is directly connected,
GigabitEthernet0/0/0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected,
GigabitEthernet0/0/1
L        192.168.1.1/32 is directly connected,
GigabitEthernet0/0/1
B     192.168.2.0/24 [20/0] via 192.168.1.2, 00:12:47
      192.168.3.0/30 is subnetted, 1 subnets
B        192.168.3.0 [20/20] via 192.168.1.2, 00:11:57
      192.168.4.0/30 is subnetted, 1 subnets
B        192.168.4.0 [20/1] via 192.168.1.2, 00:07:54
i L1  192.168.5.0/24 [115/20] via 192.168.0.1, 00:15:33,
GigabitEthernet0/0/0
      192.168.6.0/30 is subnetted, 1 subnets
B        192.168.6.0 [20/1] via 192.168.1.2, 00:07:24

IPv6 Routing Table:
C   FD00::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   FD00::2/128 [0/0]
     via GigabitEthernet0/0/0, receive
```

```
C    FD01::/64 [0/0]
      via GigabitEthernet0/0/1, directly connected
L    FD01::1/128 [0/0]
      via GigabitEthernet0/0/1, receive
B    FD02::/64 [20/0]
      via FE80::B6A8:B9FF:FE47:9471, GigabitEthernet0/0/1
B    FD03::/64 [20/20]
      via FE80::B6A8:B9FF:FE47:9471, GigabitEthernet0/0/1
B    FD04::/64 [20/1]
      via FE80::B6A8:B9FF:FE47:9471, GigabitEthernet0/0/1
I1   FD05::/64 [115/20]
      via FE80::2C1:B1FF:FE68:AB90, GigabitEthernet0/0/0
B    FD06::/64 [20/1]
      via FE80::B6A8:B9FF:FE47:9471, GigabitEthernet0/0/1
L    FF00::/8 [0/0]
      via Null0, receive
```

Configuration:

```
ipv6 unicast-routing

interface GigabitEthernet0/0/0
 ip address 192.168.0.2 255.255.255.0
 ip router isis
 negotiation auto
 ipv6 address FD00::2/64
 ipv6 router isis
 isis circuit-type level-1

interface GigabitEthernet0/0/1
 ip address 192.168.1.1 255.255.255.0
 ip router isis
 negotiation auto
 ipv6 address FD01::1/64
 ipv6 router isis
 isis circuit-type level-1

router isis
 net 49.0001.1920.1680.0002.00
 is-type level-1
```

```
 redistribute connected
 redistribute bgp 2222 level-1
 address-family ipv6
  redistribute connected
  redistribute bgp 2222 level-1

router bgp 2222
 bgp router-id 2.2.2.2
 bgp log-neighbor-changes
 neighbor 192.168.1.2 remote-as 3333
 neighbor FD01::2 remote-as 3333
 neighbor FD01::2 update-source GigabitEthernet0/0/1
 address-family ipv4
  network 192.168.1.0
  redistribute connected
  redistribute isis level-1
  neighbor 192.168.1.2 activate
 address-family ipv6
  redistribute connected
  redistribute isis level-1
  neighbor FD01::2 activate
```

B1:

```
IPv4 Routing Table:
     4.0.0.0/32 is subnetted, 1 subnets
O E2     4.4.4.4 [110/20] via 192.168.2.2, 00:11:13,
GigabitEthernet0/0/0
      5.0.0.0/32 is subnetted, 1 subnets
O E2     5.5.5.5 [110/1] via 192.168.2.2, 00:11:13,
GigabitEthernet0/0/0
      6.0.0.0/32 is subnetted, 1 subnets
O E2     6.6.6.6 [110/1] via 192.168.2.2, 00:06:40,
GigabitEthernet0/0/0
B     192.168.0.0/24 [20/0] via 192.168.1.1, 00:13:27
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected,
GigabitEthernet0/0/1
L        192.168.1.2/32 is directly connected,
GigabitEthernet0/0/1
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
```

```
C        192.168.2.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.2.1/32 is directly connected,
GigabitEthernet0/0/0
      192.168.3.0/30 is subnetted, 1 subnets
O E2     192.168.3.0 [110/20] via 192.168.2.2, 00:11:13,
GigabitEthernet0/0/0
      192.168.4.0/30 is subnetted, 1 subnets
O E2     192.168.4.0 [110/1] via 192.168.2.2, 00:07:10,
GigabitEthernet0/0/0
B     192.168.5.0/24 [20/20] via 192.168.1.1, 00:13:27
      192.168.6.0/30 is subnetted, 1 subnets
O E2     192.168.6.0 [110/1] via 192.168.2.2, 00:06:40,
GigabitEthernet0/0/0

IPv6 Routing Table:
B   FD00::/64 [20/0]
     via FE80::227:90FF:FED4:F31, GigabitEthernet0/0/1
C   FD01::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   FD01::2/128 [0/0]
     via GigabitEthernet0/0/1, receive
C   FD02::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   FD02::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
OE2 FD03::/64 [110/20]
     via FE80::267E:12FF:FE55:5720, GigabitEthernet0/0/0
OE2 FD04::/64 [110/1]
     via FE80::267E:12FF:FE55:5720, GigabitEthernet0/0/0
B   FD05::/64 [20/20]
     via FE80::227:90FF:FED4:F31, GigabitEthernet0/0/1
OE2 FD06::/64 [110/1]
     via FE80::267E:12FF:FE55:5720, GigabitEthernet0/0/0
L   FF00::/8 [0/0]
     via Null0, receive
```

Configuration:

```
ipv6 unicast-routing
```

```
interface GigabitEthernet0/0/0
 ip address 192.168.2.1 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 ipv6 address FD02::1/64
 ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
 ip address 192.168.1.2 255.255.255.0
 negotiation auto
 ipv6 address FD01::2/64

router ospf 1
 router-id 3.3.3.3
 redistribute connected subnets
 redistribute bgp 3333 subnets
 network 192.168.2.0 0.0.0.255 area 0

router bgp 3333
 bgp router-id 3.3.3.3
 bgp log-neighbor-changes
 neighbor 192.168.1.1 remote-as 2222
 neighbor FD01::1 remote-as 2222
 neighbor FD01::1 update-source GigabitEthernet0/0/1
 address-family ipv4
  network 192.168.1.0
  network 192.168.2.0
  redistribute connected
  redistribute ospf 1 match external 1 external 2
  neighbor 192.168.1.1 activate
 address-family ipv6
  redistribute connected
  redistribute ospf 1 match external 1 external 2
  neighbor FD01::1 activate

ipv6 router ospf 1
 router-id 3.3.3.3
 redistribute connected
 redistribute bgp 3333
```

B2:

```
IPv4 Routing Table:
      4.0.0.0/32 is subnetted, 1 subnets
C        4.4.4.4 is directly connected, Loopback0
      5.0.0.0/32 is subnetted, 1 subnets
B        5.5.5.5 [20/0] via 192.168.3.2, 00:08:19
      6.0.0.0/32 is subnetted, 1 subnets
B        6.6.6.6 [20/130816] via 192.168.3.2, 00:03:27
O E2  192.168.0.0/24 [110/1] via 192.168.2.1, 00:08:04,
GigabitEthernet0/0/0
O E2  192.168.1.0/24 [110/20] via 192.168.2.1, 00:08:04,
GigabitEthernet0/0/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.2.2/32 is directly connected,
GigabitEthernet0/0/0
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/30 is directly connected,
GigabitEthernet0/0/1
L        192.168.3.1/32 is directly connected,
GigabitEthernet0/0/1
      192.168.4.0/30 is subnetted, 1 subnets
B        192.168.4.0 [20/0] via 192.168.3.2, 00:03:57
O E2  192.168.5.0/24 [110/1] via 192.168.2.1, 00:08:04,
GigabitEthernet0/0/0
      192.168.6.0/30 is subnetted, 1 subnets
B        192.168.6.0 [20/3072] via 192.168.3.2, 00:03:27

IPv6 Routing Table:
OE2 FD00::/64 [110/1]
     via FE80::B6A8:B9FF:FE47:9470, GigabitEthernet0/0/0
OE2 FD01::/64 [110/20]
     via FE80::B6A8:B9FF:FE47:9470, GigabitEthernet0/0/0
C   FD02::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   FD02::2/128 [0/0]
     via GigabitEthernet0/0/0, receive
C   FD03::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
```

```
L   FD03::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
B   FD04::/64 [20/0]
     via FE80::CE7F:76FF:FE83:16D1, GigabitEthernet0/0/1
OE2 FD05::/64 [110/1]
     via FE80::B6A8:B9FF:FE47:9470, GigabitEthernet0/0/0
B   FD06::/64 [20/281856]
     via FE80::CE7F:76FF:FE83:16D1, GigabitEthernet0/0/1
L   FF00::/8 [0/0]
     via Null0, receive
```

Configuration:

```
ipv6 unicast-routing

interface Loopback0
 ip address 4.4.4.4 255.255.255.255

interface GigabitEthernet0/0/0
 ip address 192.168.2.2 255.255.255.0
 ip ospf 1 area 0
 negotiation auto
 ipv6 address FD02::2/64
 ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
 ip address 192.168.3.1 255.255.255.252
 negotiation auto
 ipv6 address FD03::1/64

router ospf 1
 router-id 4.4.4.4
 redistribute connected subnets
 redistribute bgp 3333 subnets
 network 192.168.2.0 0.0.0.255 area 0

router bgp 3333
 bgp router-id 4.4.4.4
 bgp log-neighbor-changes
 neighbor fd03::2/64 remote-as 3
```

```
 neighbor 192.168.3.2 remote-as 3
 address-family ipv4
  network 192.168.2.0
  network 192.168.3.0
  redistribute connected
  redistribute ospf 1 match external 1 external 2
  neighbor 192.168.3.2 activate
 address-family ipv6
  redistribute connected
  redistribute ospf 1 match external 1 external 2
  network FD02::/64
  network FD03::/64
  neighbor fd03::2 activate

ipv6 router ospf 1
 router-id 4.4.4.4
 redistribute connected
 redistribute bgp 3333
```

C1:

```
IPv4 Routing Table:
      4.0.0.0/32 is subnetted, 1 subnets
B        4.4.4.4 [20/0] via 192.168.3.1, 00:06:44
      5.0.0.0/32 is subnetted, 1 subnets
C        5.5.5.5 is directly connected, Loopback0
      6.0.0.0/32 is subnetted, 1 subnets
D EX     6.6.6.6 [170/130816] via 192.168.4.2, 00:02:20,
GigabitEthernet0/0/0
B     192.168.0.0/24 [20/1] via 192.168.3.1, 00:06:15
B     192.168.1.0/24 [20/20] via 192.168.3.1, 00:06:15
B     192.168.2.0/24 [20/0] via 192.168.3.1, 00:06:44
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/30 is directly connected,
GigabitEthernet0/0/1
L        192.168.3.2/32 is directly connected,
GigabitEthernet0/0/1
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/30 is directly connected,
GigabitEthernet0/0/0
```

```
L        192.168.4.1/32 is directly connected,
GigabitEthernet0/0/0
B     192.168.5.0/24 [20/1] via 192.168.3.1, 00:06:15
      192.168.6.0/30 is subnetted, 1 subnets
D        192.168.6.0 [90/3072] via 192.168.4.2, 00:02:16,
GigabitEthernet0/0/0

IPv6 Routing Table:
B   FD00::/64 [20/1]
     via FE80::267E:12FF:FE55:5721, GigabitEthernet0/0/1
B   FD01::/64 [20/20]
     via FE80::267E:12FF:FE55:5721, GigabitEthernet0/0/1
B   FD02::/64 [20/0]
     via FE80::267E:12FF:FE55:5721, GigabitEthernet0/0/1
C   FD03::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   FD03::2/128 [0/0]
     via GigabitEthernet0/0/1, receive
C   FD04::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   FD04::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
B   FD05::/64 [20/1]
     via FE80::267E:12FF:FE55:5721, GigabitEthernet0/0/1
EX  FD06::/64 [170/281856]
     via FE80::521C:B0FF:FE42:AF80, GigabitEthernet0/0/0
L   FF00::/8 [0/0]
     via Null0, receive
```

Configuration:

```
ipv6 unicast-routing

interface Loopback0
 ip address 6.6.6.6 255.255.255.255

interface GigabitEthernet0/0/0
 ip address 192.168.4.2 255.255.255.252
 negotiation auto
 ipv6 address FD04::2/64
```

```
 ipv6 eigrp 3

interface GigabitEthernet0/0/1
 ip address 192.168.6.1 255.255.255.252
 negotiation auto
 ipv6 address FD06::1/64

router eigrp 3
 default-metric 10000 100 255 1 1500
 network 192.168.4.0 0.0.0.3
 network 192.168.6.0
 redistribute connected

ipv6 router eigrp 3
 eigrp router-id 6.6.6.6
 redistribute connected
 default-metric 10000 100 255 1 1500
```

C2:

```
IPv4 Routing Table:
     4.0.0.0/32 is subnetted, 1 subnets
D EX     4.4.4.4 [170/281856] via 192.168.4.1, 00:00:21,
GigabitEthernet0/0/0
      5.0.0.0/32 is subnetted, 1 subnets
D EX     5.5.5.5 [170/130816] via 192.168.4.1, 00:00:21,
GigabitEthernet0/0/0
      6.0.0.0/32 is subnetted, 1 subnets
C        6.6.6.6 is directly connected, Loopback0
D EX  192.168.0.0/24
           [170/281856] via 192.168.4.1, 00:00:21,
GigabitEthernet0/0/0
D EX  192.168.1.0/24
           [170/281856] via 192.168.4.1, 00:00:21,
GigabitEthernet0/0/0
D EX  192.168.2.0/24
           [170/281856] via 192.168.4.1, 00:00:21,
GigabitEthernet0/0/0
      192.168.3.0/30 is subnetted, 1 subnets
D        192.168.3.0 [90/3072] via 192.168.4.1, 00:00:21,
GigabitEthernet0/0/0
```

```
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/30 is directly connected,
GigabitEthernet0/0/0
L        192.168.4.2/32 is directly connected,
GigabitEthernet0/0/0
D EX  192.168.5.0/24
           [170/281856] via 192.168.4.1, 00:00:21,
GigabitEthernet0/0/0
      192.168.6.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.6.0/30 is directly connected,
GigabitEthernet0/0/1
L        192.168.6.1/32 is directly connected,
GigabitEthernet0/0/1

IPv6 Routing Table:
EX  FD00::/64 [170/281856]
     via FE80::CE7F:76FF:FE83:16D0, GigabitEthernet0/0/0
EX  FD01::/64 [170/281856]
     via FE80::CE7F:76FF:FE83:16D0, GigabitEthernet0/0/0
EX  FD02::/64 [170/281856]
     via FE80::CE7F:76FF:FE83:16D0, GigabitEthernet0/0/0
EX  FD03::/64 [170/281856]
     via FE80::CE7F:76FF:FE83:16D0, GigabitEthernet0/0/0
C   FD04::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   FD04::2/128 [0/0]
     via GigabitEthernet0/0/0, receive
EX  FD05::/64 [170/281856]
     via FE80::CE7F:76FF:FE83:16D0, GigabitEthernet0/0/0
C   FD06::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   FD06::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Configuration:

```
ipv6 unicast-routing

interface Loopback0
```

```
 ip address 5.5.5.5 255.255.255.255

interface GigabitEthernet0/0/0
 ip address 192.168.4.1 255.255.255.252
 negotiation auto
 ipv6 address FD04::1/64
 ipv6 eigrp 3

interface GigabitEthernet0/0/1
 ip address 192.168.3.2 255.255.255.252
 negotiation auto
 ipv6 address FD03::2/64

router eigrp 3
 default-metric 10000 100 255 1 1500
 network 192.168.3.0
 network 192.168.4.0 0.0.0.3
 redistribute connected
 redistribute bgp 3

router bgp 3
 bgp log-neighbor-changes
 neighbor 192.168.3.1 remote-as 3333
 neighbor FD03::1 remote-as 3333
 address-family ipv4
  redistribute connected
  redistribute eigrp 3
  neighbor 192.168.3.1 activate
 address-family ipv6
  redistribute connected
  redistribute eigrp 3
  network FD03::/64
  neighbor FD03::1 activate

ipv6 router eigrp 3
 eigrp router-id 5.5.5.5
 redistribute bgp 3
 redistribute connected
 default-metric 10000 100 255 1 1500
```

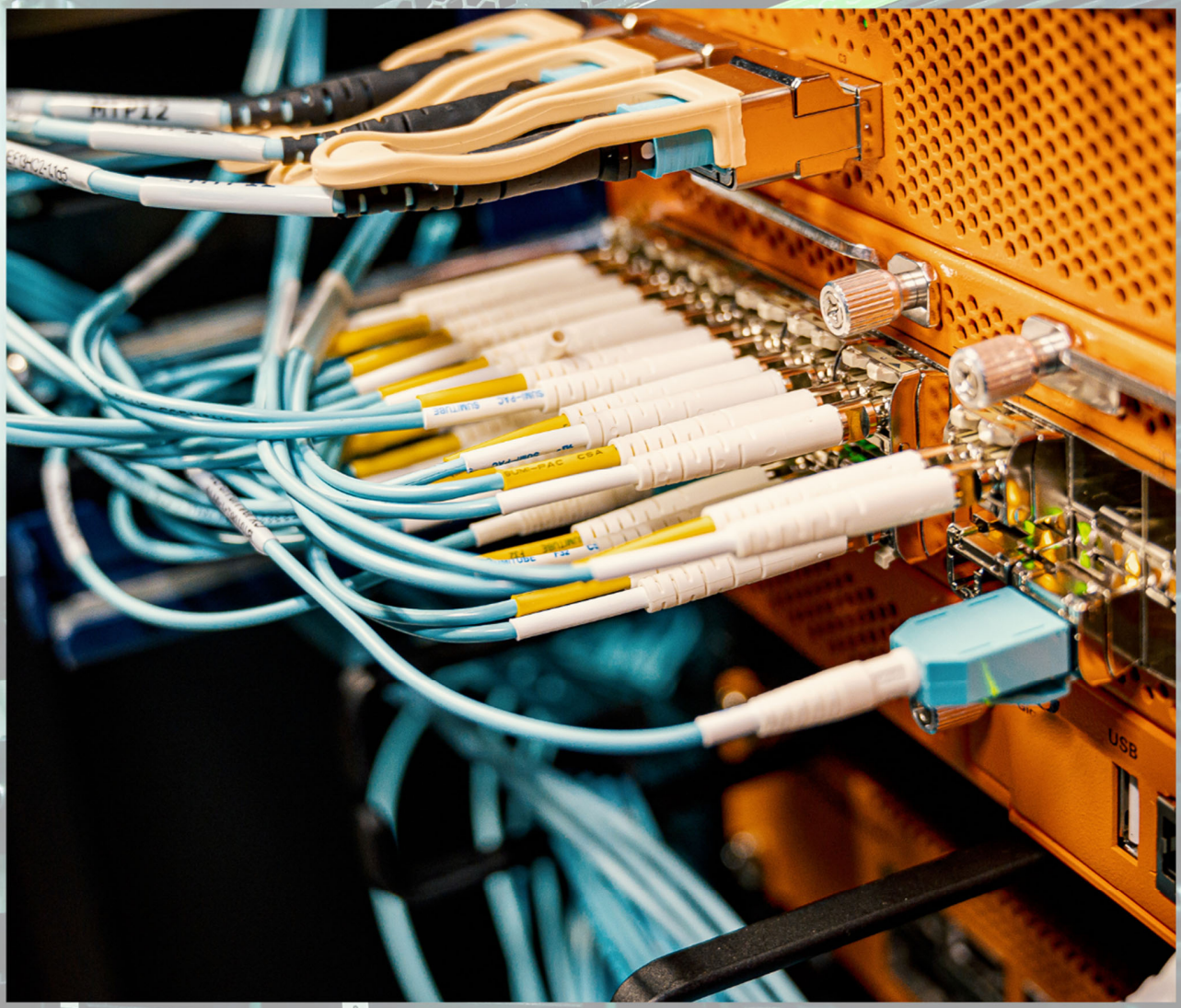
PROBLEMS:

I mostly struggled with connectivity through EIGRP. However, I soon realized that for EIGRP, I needed to have a default metric for it to work. Additionally, while configuring EIGRP, the routing protocol could not be seen through the commands *show ip route* and *show ipv6 route* until I figured out that for EIGRP, we should only configure interfaces on an EIGRP router that connect to other routers with EIGRP. After configuring a metric and fixing the interface configurations, the routers with EIGRP running were able to successfully ping other networks.

CONCLUSION:

I learned how to set up eBGP—something I found very complex as it not only involved setting up routing protocols but also a routing protocol that connected each of the domains to allow for end-to-end connectivity. Although there challenging moments, I'm glad I was able to finish the lab successfully.

Lab 4: iBGP



PURPOSE:

Learn to configure the internal Border Gateway Protocol, while using OSPF, EIGRP, and IS-IS, as well as the external Border Gateway Protocol to route between Autonomous Systems.

BACKGROUND INFORMATION ON LAB CONCEPTS:

BGP (iBGP and eBGP):

BGP is a routing protocol that uses autonomous systems (AS). It makes routing decisions by considering paths, network policies, and rule-sets (which are configured by a network administrator). BGP is the only commonly used routing protocol with TCP as its transport protocol. When BGP uses the same AS number and therefore the same autonomous systems for peers, it is called iBGP.

A BGP peer uses a finite-state machine with six states for connections with peers: Idle, Connect, Active, OpenConfirm, and Established. All routers within an AS, like in iBGP, must be configured in a full mesh topology, meaning that each router must be a peer to every other router in the network. However, this may hinder scalability. However, BGP is said to be the "most scalable of all routing protocols," due to route reflectors that reduce the number of connections required.

An analogy that could be used to explain iBGP (internal BGP) and eBGP (external BGP) usage within this lab involves a group of oranges. Imagine they reside in different farms, but they need to coordinate together to escape and return to Orange Land. To do that, they would need a communication system like eBGP that would ensure that oranges from different farms—similar to different networks—can still communicate with each other. They would need to do so as unified groups, which requires iBGP to act like a communication network with walkie-talkies that allows all oranges within a farm to receive the escape plans.

OSPF:

OSPF stands for Open Shortest Path First. It is a link-state routing protocol, meaning that it only sends updates when a change occurs in the routing table and that all routers in the protocol know the full topology.

OSPF is able to detect changes in network topology within seconds, calculating link metrics—which help judge the quality of potential links—based on amount of data, distance of a router, and link availability and reliability.

EIGRP:

EIRGRP, or Enhanced Interior Gateway Routing Protocol, allows for faster convergence than networks like BGP. It calculates routes through evaluating the distance and whether a destination path has loops or not.

This routing protocol has both link-state and distance vector characteristics—an EIGRP router only sends updates when a change occurs in the routing table but it also only advertises its best route to a neighbor (which is more like distance vector) and does not know the entire topology.

IS-IS:

IS-IS (Intermediate System to Intermediate System) is designed to be an interior gateway protocol. It is also a link-state routing protocol and uses the same algorithm as OSPF to find the best path.

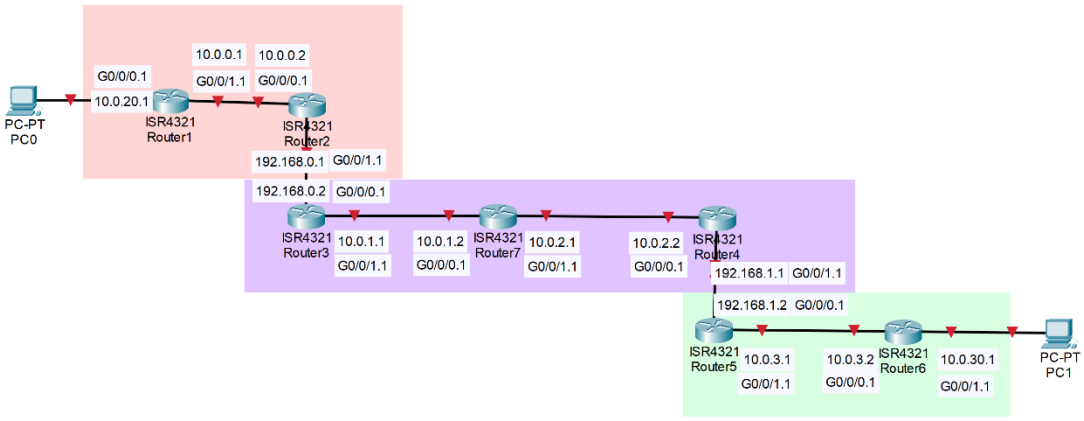
However, IS-IS is different from OSPF in that there are three types of routers used to connect different areas together: Level 1, Level 1-2, and Level 2. Level 1 can only establish neighbor adjacencies for routers in the same area; Level 2 can only establish neighbor adjacencies for routers in different areas; and Level 1-2 and perform both functions.

LAB SUMMARY:

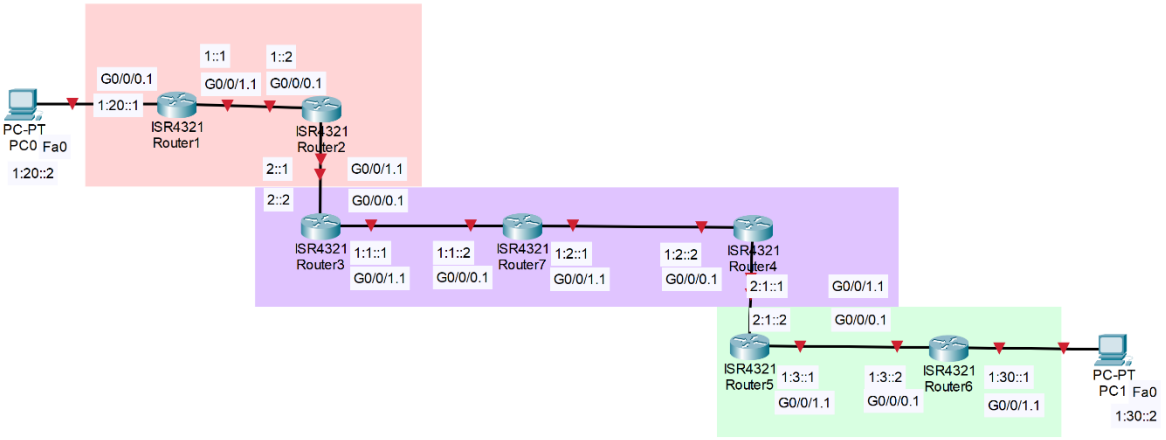
In this lab, we configured internal BGP, both for IPv4 and IPv6, within AS 2; the middle router uses EIGRP as the interior gateway protocol. eBGP sessions were established between AS 1 and AS 2, as well as between AS 2 and AS 3, with its routes carried through iBGP. We ended up connecting the two end devices after successful configuration.

NETWORK DIAGRAMS:

IPv4:



IPv6:



Router	Routing Protocols Used
R1	OSPF (IPv4 & IPv6)
R2	OSPF (IPv4 & IPv6), eBGP
R3	EIGRP (IPv4 & IPv6), eBGP, iBGP
R4	EIGRP (IPv4 & IPv6), eBGP, iBGP
R5	IS-IS (IPv4 & IPv6), eBGP
R6	IS-IS (IPv4 & IPv6)
R7	EIGRP (IPv4 & IPv6), iBGP

LAB COMMANDS:

First, after configuring network addresses for interfaces on each router, set up IS-IS, OSPF, and EIGRP, respectively:

IS-IS:

ip router isis enables an interface to participate in IS-IS for IPv4; *ipv6 router isis* enables an interface to participate in IS-IS for IPv6.

Since the routers will not use different areas, both of the routers (A1 and A2) will be Level 1. Enter the command *router isis* and use the global command *is-type level-1* to configure this. Specify a Network Entity Title (NET), which is the "ID" of the router in IS-IS. For example: *net 49.0001.1920.1680.0001.00.*

Then, use *redistribute connected* and *redistribute bgp [chosen Autonomous System number] [router level]*, noting to only use these commands for routers that are directly connected to a router with

a different router protocol. Go into *address-family ipv6* and configure the same. These commands are used to advertised routes learned through BGP and directly connected routes into the IS-IS domain, but configuring them in this step will save some time.

OSPF:

ip ospf 1 area 0 configures the OSPF process ID (1) and the area (0, also known as the backbone area) for an interface. Process IDs are only for local identification (within the router); since only one area is needed for this lab, Area 0 is used. Repeat for IPv6, replacing *ip ospf 1 area 0* with *ipv6 ospf 1 area 0.*

In *router ospf 1*, where "1" represents the process ID, configure network statements to represent the different networks covered by OSPF, like: *network 192.168.2.0 0.0.0.255 area 0.* Next, configure the router ID, which identifies the router in the OSPF network. Do the same for IPv6, replacing *router ospf 1* with *ipv6 router ospf 1.*

Finally, use *redistribute connected* and *redistribute bgp [chosen Autonomous System number]*, noting to only use these commands for routers that are directly connected to a router with a different router protocol. These commands are used to advertised routes learned through BGP and directly connected routes into the OSPF domain, but configuring them in this step will save some time.

EIGRP:

Enter the *router eigrp 3* command to configure globally, where "3" is the chosen Autonomous System number in this example. Configure the default metric values for routes that are redistributed through EIGRP. An example is *default-metric 10000 100 255 1 1500*, where 10000 represents the bandwidth, 100 represents the delay, 255 represents the reliability (highest reliability), 1 is the load (least load), and 1500 is the MTU.

Next, configure the network statements for the networks that the EIGRP routing protocol will cover. An example is *network 192.168.4.0 0.0.0.3.* Then, use *redistribute connected* and *redistribute bgp [chosen Autonomous System number]*, noting to only use these commands for routers that are directly connected to a router with a different router protocol. These commands are used to advertised routes learned through BGP and directly connected routes into the EIGRP domain, but configuring them in this step will save some time.

Then, type in the command *ipv6 router eigrp 3* to configure the router ID as such: *eigrp router-id 6.6.6.6*; also configure the *redistribute* commands. Configure the default metric values for IPv6 as well; the same values may be used for IPv4 and IPv6.

Configure interfaces using the command *ipv6 eigrp 3* for IPv6 only on links that are part of the EIGRP Autonomous System (AS 3).

Next, follow the following steps for each routing device to configure **BGP**.

Enter the BGP router configuration with the global command *router bgp [router Autonomous System number].* Configure neighbors using commands like *neighbor 192.168.3.1 remote-as 3333* for IPv4 and *neighbor FD03::1 remote-as 3333* for IPv6. For iBGP, make sure to use the same Autonomous System number for both *router bgp* and *neighbor* commands. For eBGP, make sure to use the neighboring network's Autonomous System number. Next, enter the commands *address-family ipv4* and *address-family ipv6*, respectively, to configure IPv4 and IPv6 BGP. In both, enter *redistribute connected* and *redistribute [either IS-IS, OSPF, or EIGRP, depending on the router, with the associated router level, process ID, or Autonomous System number, respectively].* If needed, specific networks may be advertised using the *network [IPv4 address/IPv6 address]* command. This will allow for connected routes and routes learned through other routing protocols to be advertised through BGP and make them visible across the entire network.

The following are some iBGP-specific commands. We must use the *next-hop-self [router 7 IP address]* command on routers 3 and 4 within both IPv4 and IPv6 address families. This will allow Router 7—the middle router—to reach the next-hop routes otherwise unavailable. We will also need to use the *neighbor [IP address] route-reflector-client* command that, while optional, will simplify the topology, especially if we were to upscale the network. Since Router 7 is a neighbor to the two other routers in the AS, Router 3 and Router 4, we will designate Router 7 as the Route Reflector and configure this command in its IPv4 and IPv6 address families.

CONFIGURATIONS:

Trace routes from PC1 to PC0, with IPv4 and IPv6, respectively:

```
Tracing route to DESKTOP-27SAP3J [10.0.20.2]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  10.0.30.1
  2    <1 ms    <1 ms    <1 ms  10.0.3.1
  3     1 ms    <1 ms    <1 ms  192.168.1.1
  4    <1 ms    <1 ms    <1 ms  10.0.2.1
  5     1 ms    <1 ms    <1 ms  10.0.1.1
  6     1 ms     1 ms     1 ms  192.168.0.1
  7     1 ms     1 ms     1 ms  10.0.0.1
  8     1 ms     1 ms     1 ms  DESKTOP-27SAP3J [10.0.20.2]

Trace complete.

Tracing route to 1:20::2 over a maximum of 30 hops

  1    <1 ms    <1 ms    <1 ms  1:30::1
  2    <1 ms    <1 ms    <1 ms  1:3::1
  3     1 ms    <1 ms     1 ms  2:1::1
  4     1 ms    <1 ms    <1 ms  1:2::1
  5     1 ms     1 ms     1 ms  1:1::1
  6     1 ms     1 ms     1 ms  2::1
```

```
  7     1 ms     1 ms     1 ms  1::1
  8     1 ms     1 ms     1 ms  1:20::2

Trace complete.
```

Hostname R1:

```
ipv6 unicast-routing

interface GigabitEthernet0/0/0
 ip address 10.0.20.1 255.255.255.0
 negotiation auto
 ipv6 address 1:20::1/64
 ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
 ip address 10.0.0.1 255.255.255.0
 negotiation auto
 ipv6 address 1::1/64
 ipv6 ospf 1 area 0

interface Serial0/1/0
 no ip address
 shutdown

interface Serial0/1/1
 no ip address
 shutdown

interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto

router ospf 1
 router-id 1.1.1.1
 network 10.0.0.0 0.0.0.255 area 0
 network 10.0.20.0 0.0.0.255 area 0

ipv6 router ospf 1
 router-id 1.1.1.1
```

Hostname R2:

```
ipv6 unicast-routing
```

```
interface GigabitEthernet0/0/0
 ip address 10.0.0.2 255.255.255.0
 negotiation auto
 ipv6 address 1::2/64
 ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
 ip address 192.168.0.1 255.255.255.0
 negotiation auto
 ipv6 address 2::1/64
 ipv6 ospf 1 area 0

interface Serial0/1/0

interface Serial0/1/1

interface GigabitEthernet0/2/0
 no ip address
 shutdown
 negotiation auto

interface GigabitEthernet0/2/1
 no ip address
 shutdown
 negotiation auto

interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto

router ospf 1
 router-id 2.2.2.2
 redistribute bgp 1 metric 10 subnets
 network 10.0.0.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0

router bgp 1
 bgp router-id 2.2.2.2
 bgp log-neighbor-changes
 neighbor 2::2 remote-as 2
 neighbor 192.168.0.2 remote-as 2

 address-family ipv4
  redistribute connected
  redistribute ospf 1
```

```
  no neighbor 2::2 activate
  neighbor 192.168.0.2 activate
 exit-address-family

 address-family ipv6
  redistribute connected
  redistribute ospf 1 metric 10
  neighbor 2::2 activate
 exit-address-family

ipv6 router ospf 1
 router-id 2.2.2.2
 redistribute bgp 1 metric 10
```

Hostname R3:

```
ipv6 unicast-routing

interface Loopback0
 ip address 3.3.3.3 255.255.255.255
 ipv6 address 100:3::3/128
 ipv6 eigrp 1

interface GigabitEthernet0/0/0
 ip address 192.168.0.2 255.255.255.0
 negotiation auto
 ipv6 address 2::2/64
 ipv6 eigrp 1

interface GigabitEthernet0/0/1
 ip address 10.0.1.1 255.255.255.0
 negotiation auto
 ipv6 address 1:1::1/64
 ipv6 eigrp 1

interface Serial0/1/0
 no ip address
 shutdown

interface Serial0/1/1
 no ip address
 shutdown

interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
```

```
 negotiation auto

router eigrp 1
 network 3.3.3.3 0.0.0.0
 network 10.0.1.0 0.0.0.255
 network 192.168.0.0
 eigrp router-id 3.3.3.3

router bgp 2
 bgp router-id 3.3.3.3
 bgp log-neighbor-changes
 neighbor 2::1 remote-as 1
 neighbor 100:4::4 remote-as 2
 neighbor 100:4::4 update-source Loopback0
 neighbor 100:7::7 remote-as 2
 neighbor 100:7::7 update-source Loopback0
 neighbor 4.4.4.4 remote-as 2
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 7.7.7.7 remote-as 2
 neighbor 7.7.7.7 update-source Loopback0
 neighbor 192.168.0.1 remote-as 1

 address-family ipv4
  redistribute connected
  no neighbor 2::1 activate
  no neighbor 100:4::4 activate
  no neighbor 100:7::7 activate
  neighbor 4.4.4.4 activate
  neighbor 7.7.7.7 activate
  neighbor 7.7.7.7 next-hop-self
  neighbor 192.168.0.1 activate
 exit-address-family

 address-family ipv6
  redistribute connected
  neighbor 2::1 activate
  neighbor 100:4::4 activate
  neighbor 100:7::7 activate
  neighbor 100:7::7 next-hop-self
 exit-address-family

ipv6 router eigrp 1
 eigrp router-id 3.3.3.3
```

Hostname R4:

```
ipv6 unicast-routing
```

```
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
 ipv6 address 100:4::4/128
 ipv6 eigrp 1

interface GigabitEthernet0/0/0
 ip address 10.0.2.2 255.255.255.0
 negotiation auto
 ipv6 address 1:2::2/64
 ipv6 eigrp 1

interface GigabitEthernet0/0/1
 ip address 192.168.1.1 255.255.255.0
 negotiation auto
 ipv6 address 2:1::1/64
 ipv6 eigrp 1

interface Serial0/1/0
 no ip address

interface Serial0/1/1
 no ip address

interface GigabitEthernet0/2/0
 no ip address
 negotiation auto

interface GigabitEthernet0/2/1
 no ip address
 negotiation auto

interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 negotiation auto

router eigrp 1
 network 4.4.4.4 0.0.0.0
 network 10.0.2.0 0.0.0.255
 network 192.168.1.0
 eigrp router-id 4.4.4.4

router bgp 2
 bgp router-id 4.4.4.4
 bgp log-neighbor-changes
 neighbor 2:1::2 remote-as 3
```

```
 neighbor 100:3::3 remote-as 2
 neighbor 100:3::3 update-source Loopback0
 neighbor 100:7::7 remote-as 2
 neighbor 100:7::7 update-source Loopback0
 neighbor 3.3.3.3 remote-as 2
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 7.7.7.7 remote-as 2
 neighbor 7.7.7.7 update-source Loopback0
 neighbor 192.168.1.2 remote-as 3

 address-family ipv4
  redistribute connected
  no neighbor 2:1::2 activate
  no neighbor 100:3::3 activate
  no neighbor 100:7::7 activate
  neighbor 3.3.3.3 activate
  neighbor 7.7.7.7 activate
  neighbor 7.7.7.7 next-hop-self
  neighbor 192.168.1.2 activate
 exit-address-family

 address-family ipv6
  redistribute connected
  neighbor 2:1::2 activate
  neighbor 100:3::3 activate
  neighbor 100:7::7 activate
  neighbor 100:7::7 next-hop-self
 exit-address-family

ipv6 router eigrp 1
 eigrp router-id 4.4.4.4
```

Hostname R5:

```
ipv6 unicast-routing

interface GigabitEthernet0/0/0
 ip address 192.168.1.2 255.255.255.0
 ip router isis
 negotiation auto
 ipv6 address 2:1::2/64
 ipv6 router isis

interface GigabitEthernet0/0/1
 ip address 10.0.3.1 255.255.255.0
 ip router isis
 negotiation auto
```

```
 ipv6 address 1:3::1/64
 ipv6 router isis

interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto

router isis
 net 49.0012.0000.0000.0005.00
 is-type level-1
 metric-style wide
 log-adjacency-changes
 redistribute bgp 3 metric 30 level-1
 address-family ipv6
  redistribute bgp 3 metric 30 level-1
 exit-address-family

router bgp 3
 bgp router-id 5.5.5.5
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 2:1::1 remote-as 2
 neighbor 192.168.1.1 remote-as 2

 address-family ipv4
  redistribute connected
  redistribute isis level-1 metric 10
  neighbor 192.168.1.1 activate
 exit-address-family

 address-family ipv6
  redistribute connected
  redistribute isis metric 10 level-1
  neighbor 2:1::1 activate
 exit-address-family
```

Hostname R6:

```
ipv6 unicast-routing

interface GigabitEthernet0/0/0
 ip address 10.0.3.2 255.255.255.0
 ip router isis
 negotiation auto
 ipv6 address 1:3::2/64
```

```
 ipv6 router isis

interface GigabitEthernet0/0/1
 ip address 10.0.30.1 255.255.255.0
 ip router isis
 negotiation auto
 ipv6 address 1:30::1/64
 ipv6 router isis

interface Serial0/1/0
 no ip address
 shutdown

interface Serial0/1/1
 no ip address
 shutdown

interface Service-Engine0/2/0
 no ip address
 shutdown

interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto

interface Vlan1
 no ip address
 shutdown

router isis
 net 49.0012.0000.0000.0006.00
 is-type level-1
 metric-style wide
 log-adjacency-changes
```

Hostname R7:

```
ipv6 unicast-routing

interface Loopback0
 ip address 7.7.7.7 255.255.255.255
 ipv6 address 100:7::7/128
 ipv6 eigrp 1

interface GigabitEthernet0/0/0
```

```
 ip address 10.0.1.2 255.255.255.0
 negotiation auto
 ipv6 address 1:1::2/64
 ipv6 eigrp 1

interface GigabitEthernet0/0/1
 ip address 10.0.2.1 255.255.255.0
 negotiation auto
 ipv6 address 1:2::1/64
 ipv6 eigrp 1

interface Serial0/1/0
 no ip address

interface Serial0/1/1
 no ip address

interface GigabitEthernet0/2/0
 no ip address
 negotiation auto

interface GigabitEthernet0/2/1
 no ip address
 negotiation auto

interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 negotiation auto

interface Vlan1
 no ip address

router eigrp 1
 network 7.7.7.7 0.0.0.0
 network 10.0.1.0 0.0.0.255
 network 10.0.2.0 0.0.0.255
 eigrp router-id 7.7.7.7

router bgp 2
 bgp router-id 7.7.7.7
 bgp log-neighbor-changes
 neighbor 100:3::3 remote-as 2
 neighbor 100:3::3 update-source Loopback0
 neighbor 100:4::4 remote-as 2
 neighbor 100:4::4 update-source Loopback0
 neighbor 3.3.3.3 remote-as 2
```

```
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 4.4.4.4 remote-as 2
 neighbor 4.4.4.4 update-source Loopback0

address-family ipv4
 no neighbor 100:3::3 activate
 no neighbor 100:4::4 activate
 neighbor 3.3.3.3 activate
 neighbor 3.3.3.3 route-reflector-client
 neighbor 4.4.4.4 activate
 neighbor 4.4.4.4 route-reflector-client
exit-address-family

address-family ipv6
 neighbor 100:3::3 activate
 neighbor 100:3::3 route-reflector-client
 neighbor 100:4::4 activate
 neighbor 100:4::4 route-reflector-client
exit-address-family

ipv6 router eigrp 1
 eigrp router-id 7.7.7.7
```

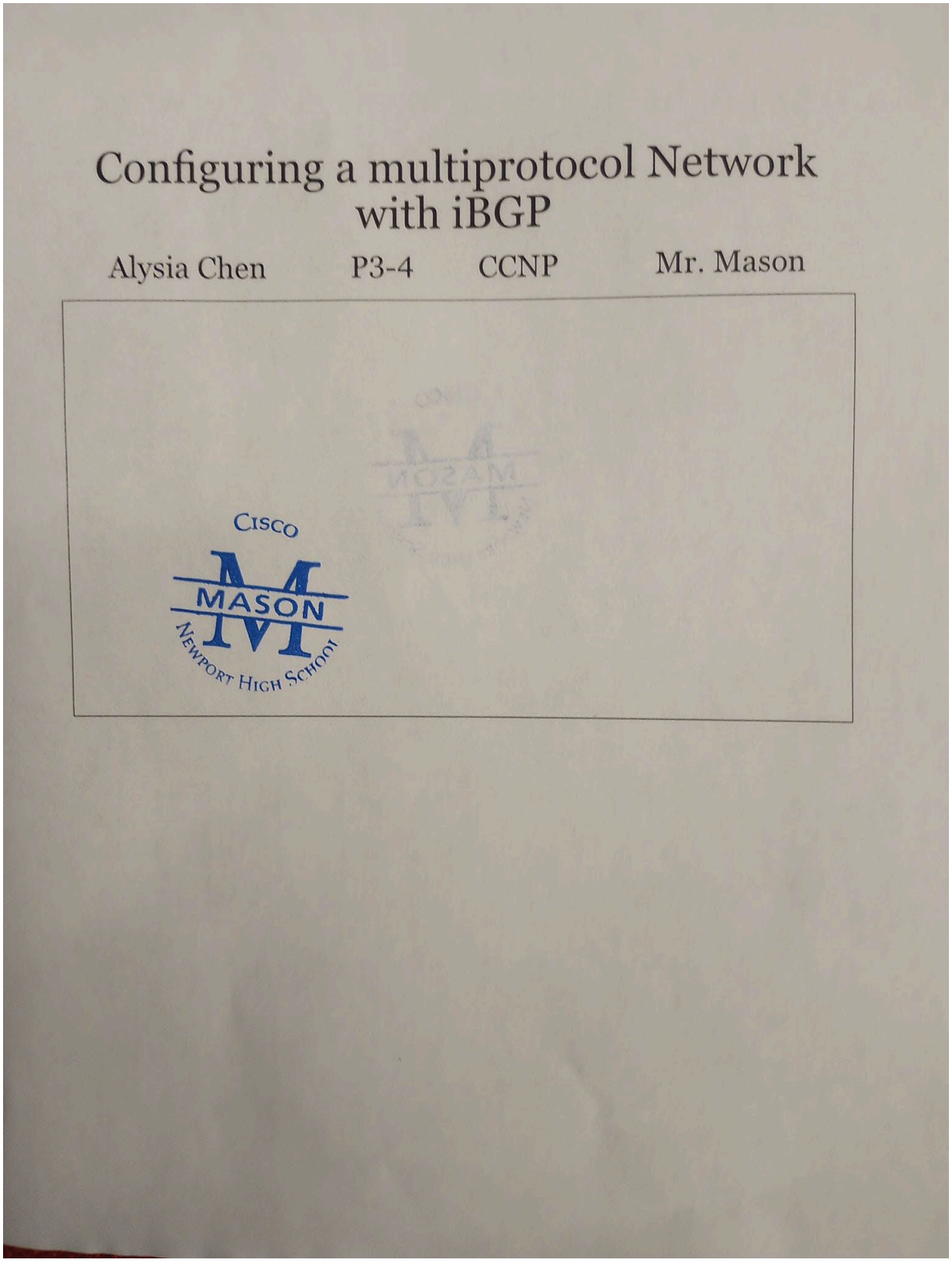
PROBLEMS:

Although there weren't significant challenges, when we were testing out our pings from our PCs, we realized that we couldn't ping from the end devices at all, even though Router 1 could ping Router 6. We soon realized that the physical Ethernet connection between PC0 and Router 1 and between PC1 and Router 6 was not set up; after that, we were successfully able to ping from end-to-end!

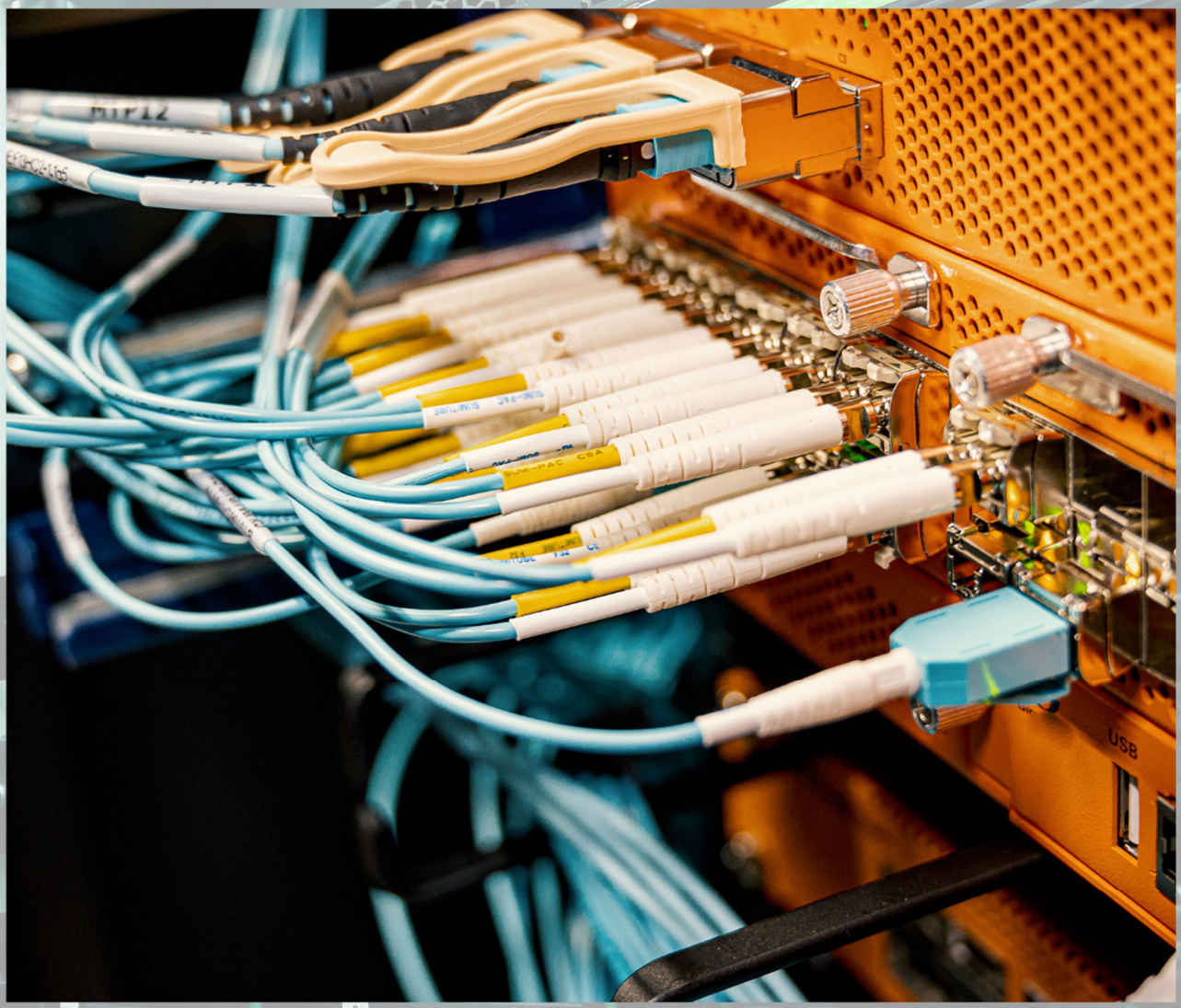
CONCLUSION:

This lab was very complex and took some time for me to understand due to how iBGP needed to run inside the Interior Gateway Protocol (EIGRP in our case), eBGP needed to connect the different Autonomous Systems, and we still needed to set up IS-IS and OSPF. However, after extensive research, I feel that I've gained a lot of knowledge from this lab and got reminded to never forget to plug in the Ethernet cables to the PCs when setting up the topology!

TEACHER SIGN-OFF:



Lab 5: Amazon Cloud Foundations 1-3



PURPOSE:

Learn to configure AWS IAM, build a VPC and launch a web server, and learn about Amazon EC2.

BACKGROUND INFORMATION ON LAB CONCEPTS:

IAM:

AWS Identity and Access Management has many purposes. Some of these purposes include the management of IAM Users (which are individuals), including their credentials and permissions; the management of IAM Roles (which can be assigned to anyone and multiple people at once if needed), including their credentials and permissions; and the management of identity federation so that existing users can have access permissions without needing to create individual IAM User credentials. Authentication is provided by matching a new user's sign-in credentials to a principal (an IAM user, federated user, IAM role, or application) that is trusted; a request is made to grant the principal access to resources, which will be allowed only if the user has permission.

VPC:

Amazon Virtual Private Cloud, also known as Amazon VPC, enables one to launch Amazon Web Services (AWS) into a virtual network that is defined by the network administrator. This virtual network is similar to a traditional network, but with the added benefits of scalability and flexibility. A VPC may span multiple Availability Zones. VPC has subnets (a range of IP addresses) that must be in a single Availability Zone. The network administrator can assign IP addresses (both IPv4 and IPv6) to VPCs and subnets. Gateways connect a VPC to another network; endpoints connect a VPC to AWS privately (without using internet gateway or a NAT device). Transit gateways act like central hubs to route traffic between the networks of VPCs and other connections.

EC2:

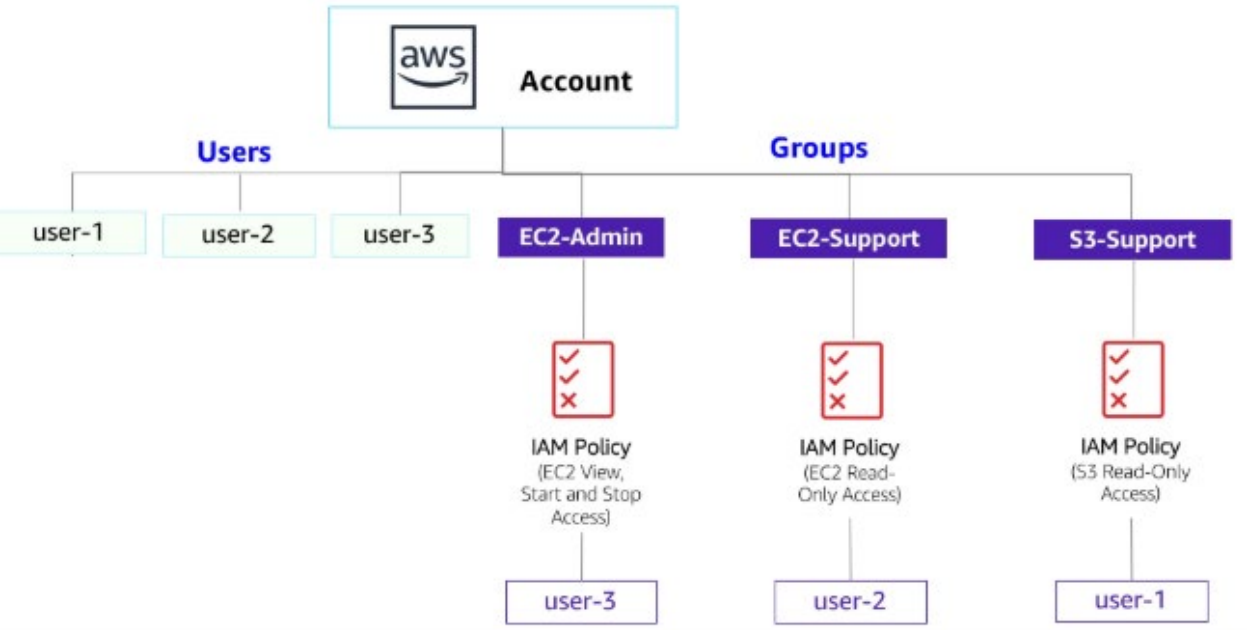
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that allows for scalability in the cloud. EC2 has over 750 instances and choice of the latest processor, storage, networking, operating system, and purchase model—the broadest and deepest compute platform. Benefits include on-demand infrastructure, secure compute, optimization, and more.

LAB SUMMARY:

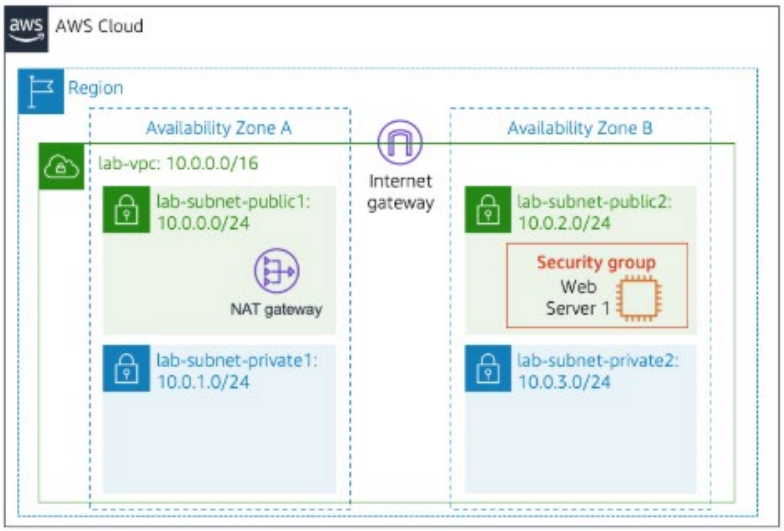
In these labs, I learned about different services that Amazon AWS offers—their purposes and how they work. I used the AWS Console and terminal sessions to complete different tasks, including configurations and checks.

DIAGRAMS:

Lab 1



Lab 2



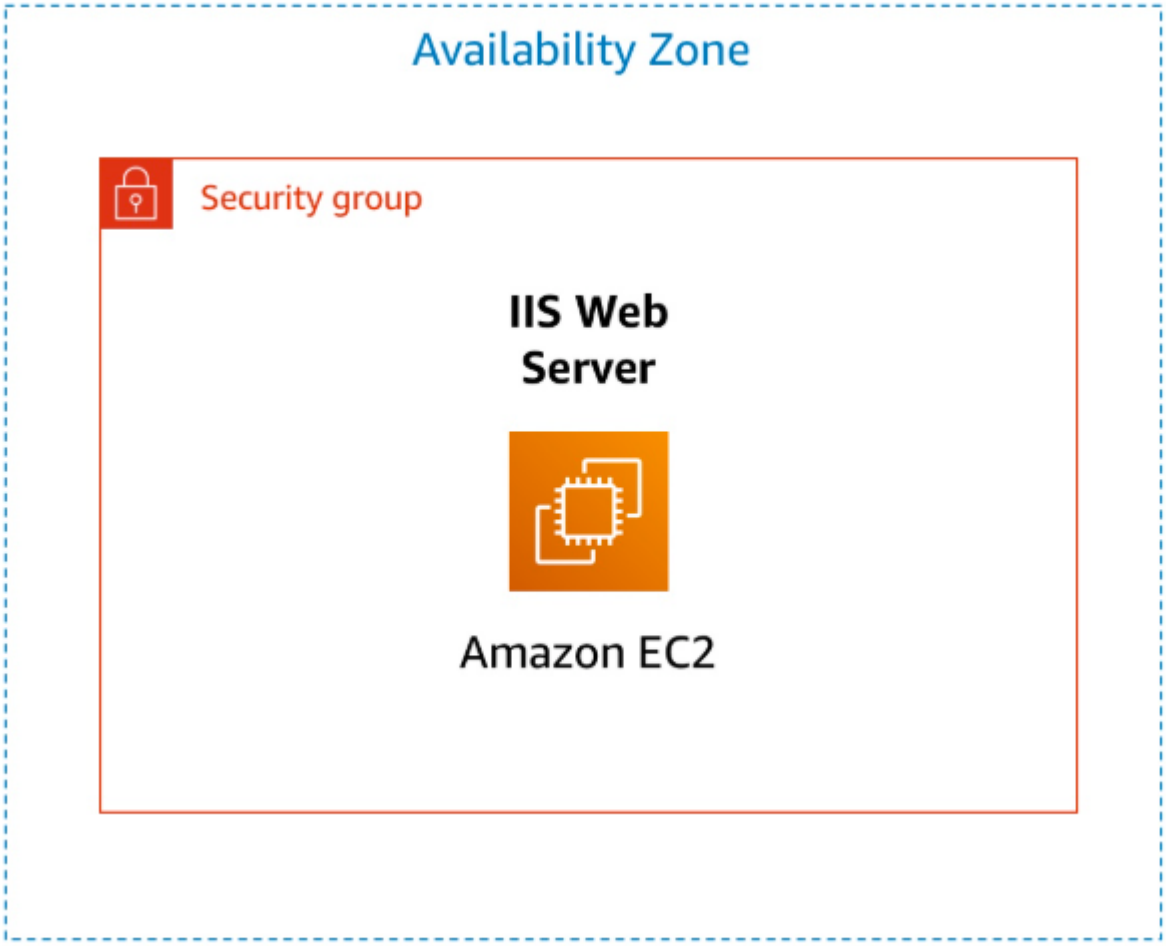
Public Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	Internet gateway

Private Route Tables

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	NAT gateway

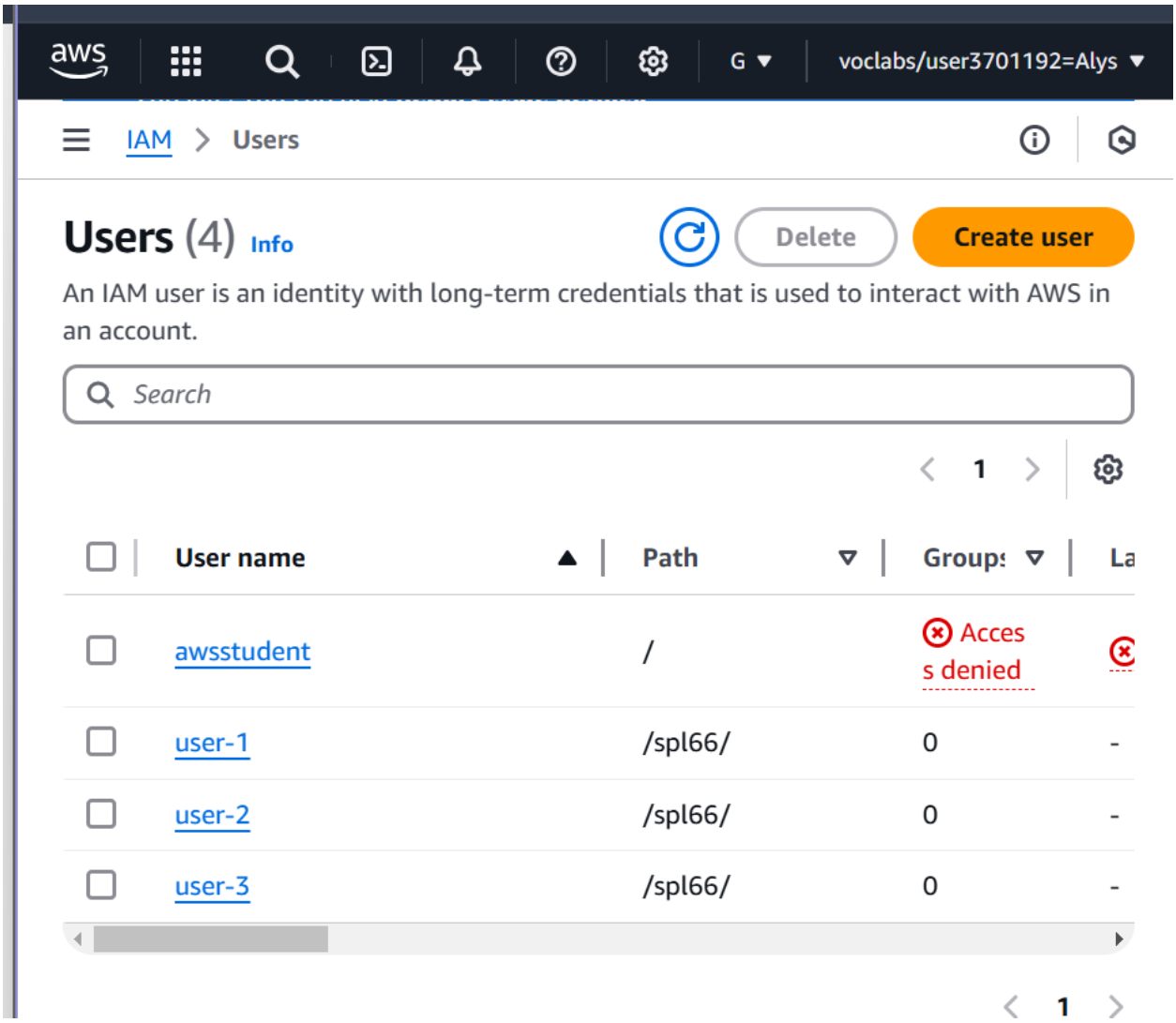
Lab 3



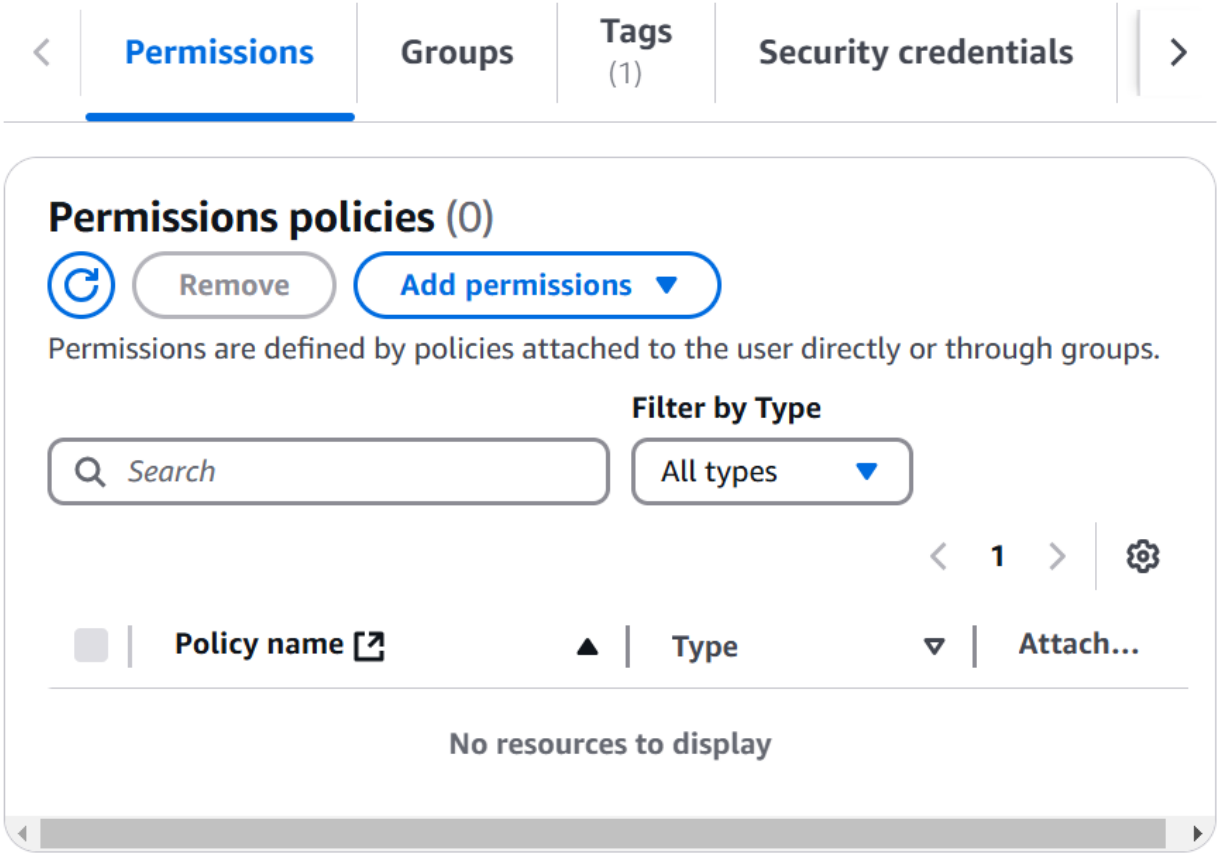
LAB COMMANDS:

LAB 1

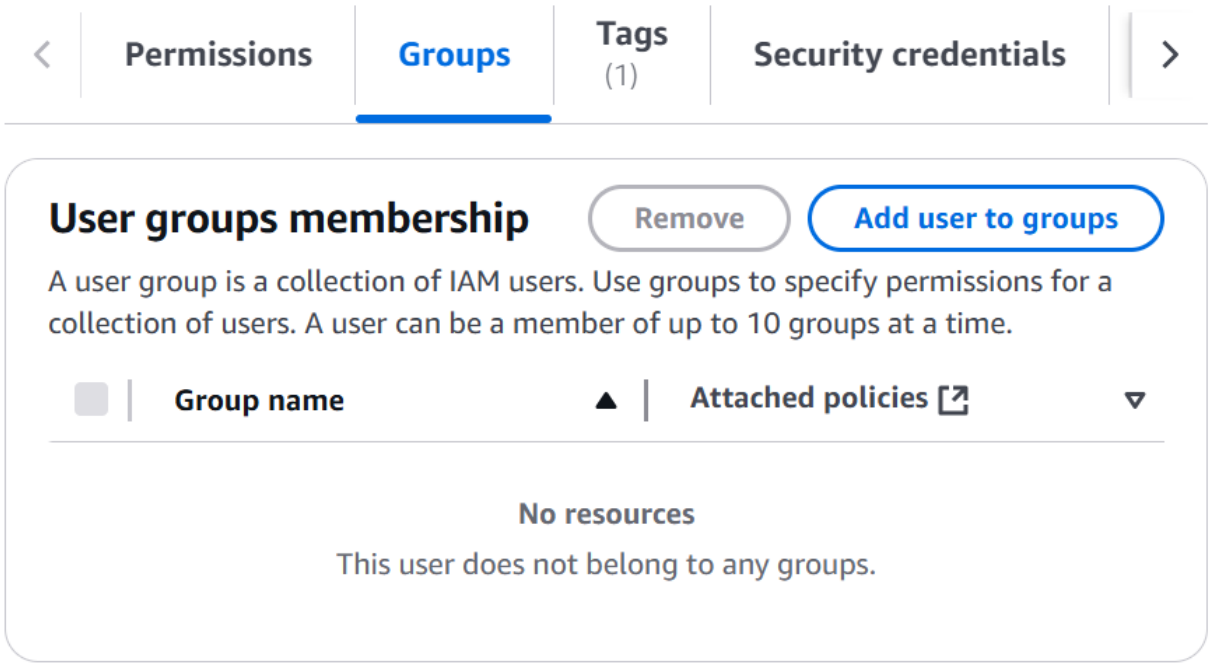
Go into the search box to open the IAM console and choose Users in the navigation pane.


aws
G
voclabs/user3701192=Alys
IAM > Users
Users (4) Info
Delete
Create user
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Search
1
User name
Path
Groups
La
awsstudent
/
Access denied
user-1
/spl66/
0
-
user-2
/spl66/
0
-
user-3
/spl66/
0
-
1

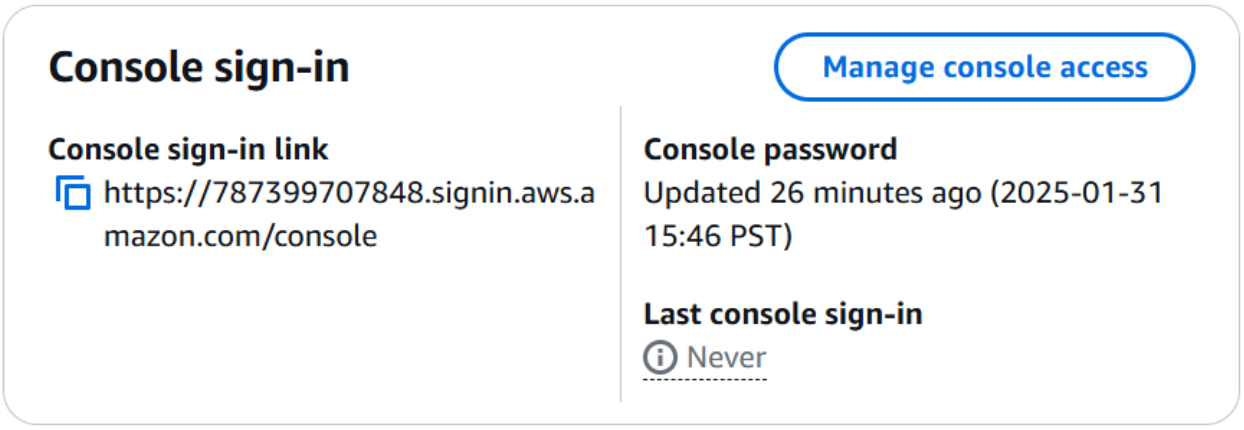
Click on the user-1 link, which will bring you to the Permissions tab as shown below:



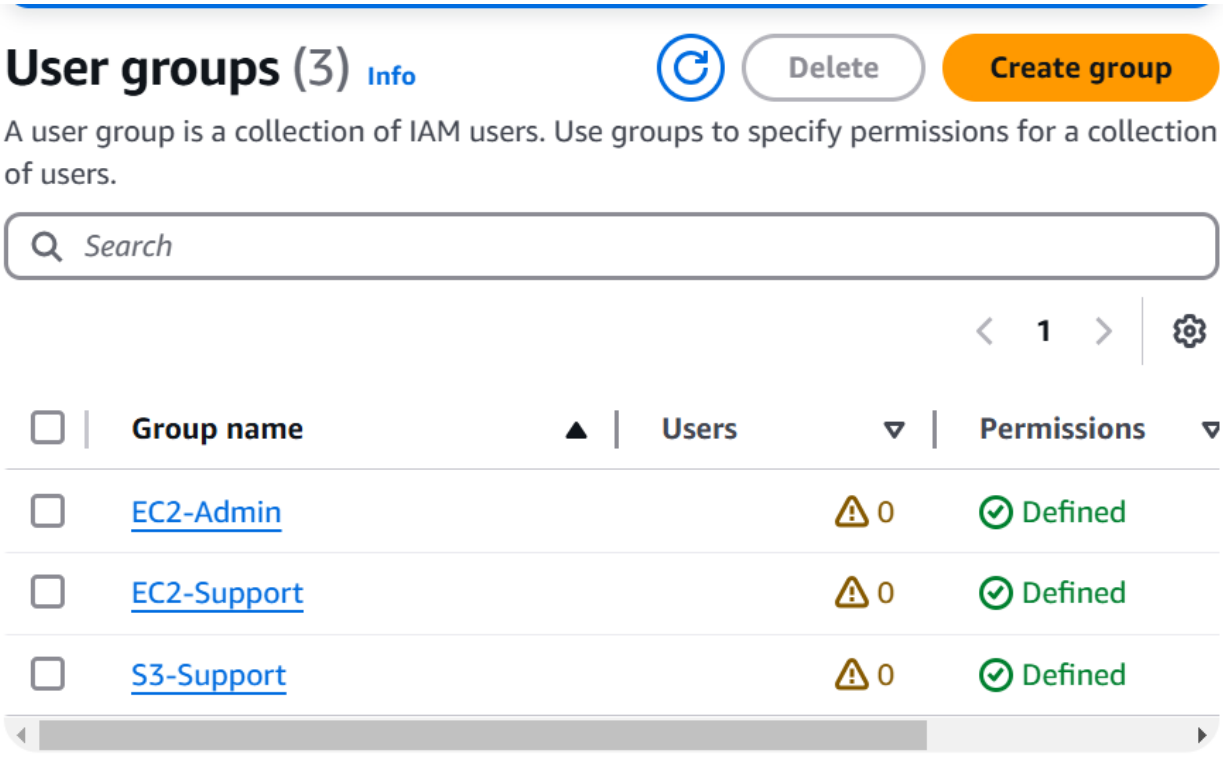
Choose the Groups tab.



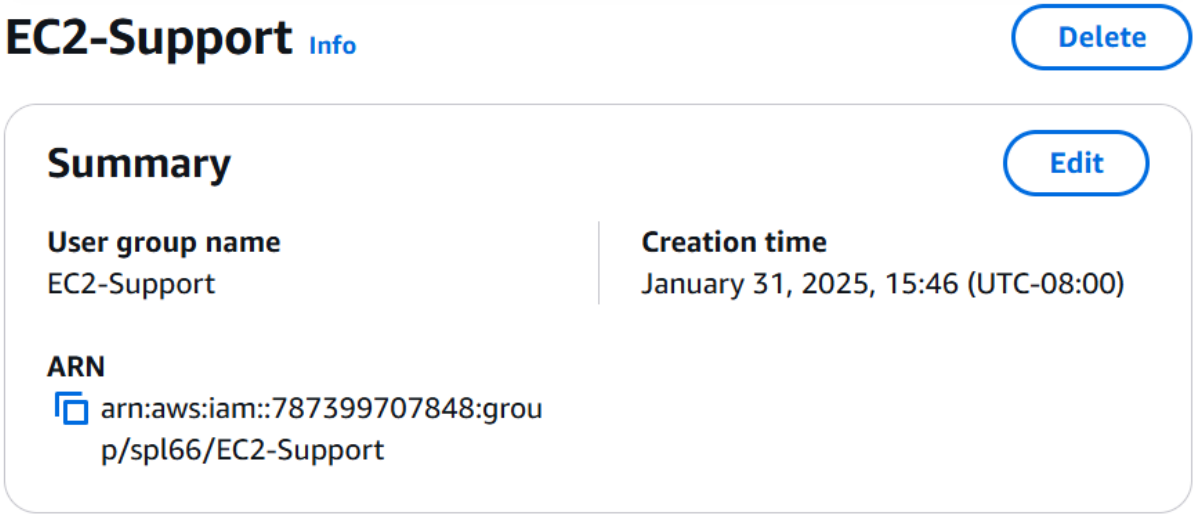
Then, look at the Security credentials tab; user-1 has a Console password assigned.

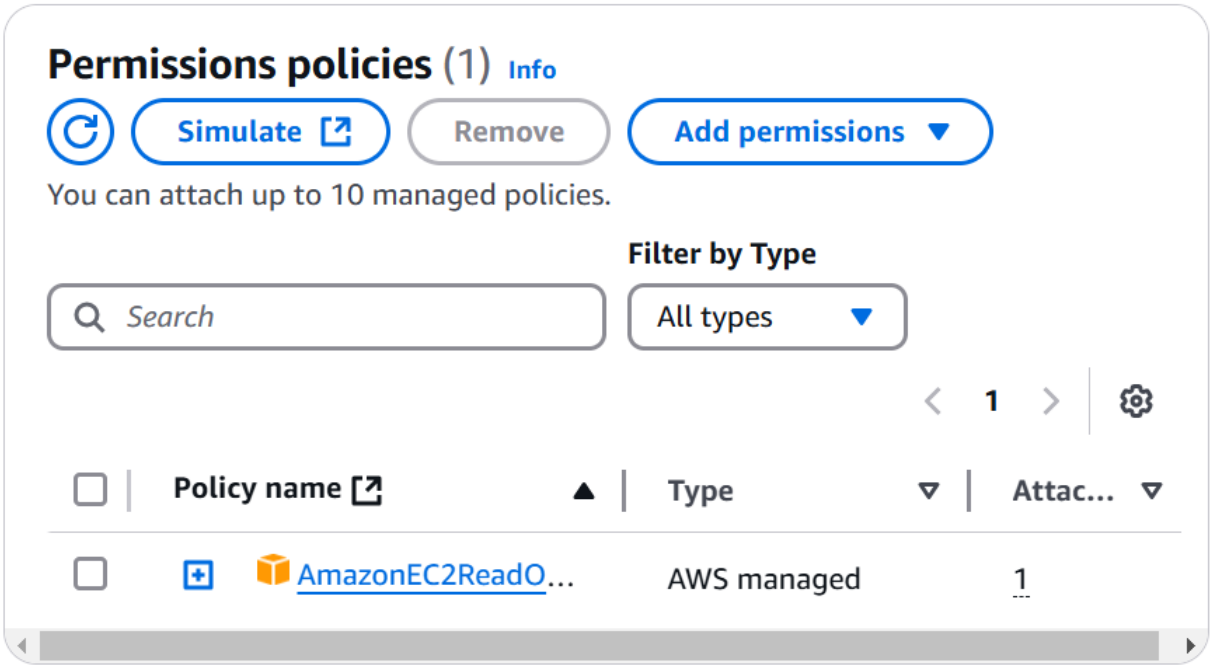


In the navigation pane, choose User groups.

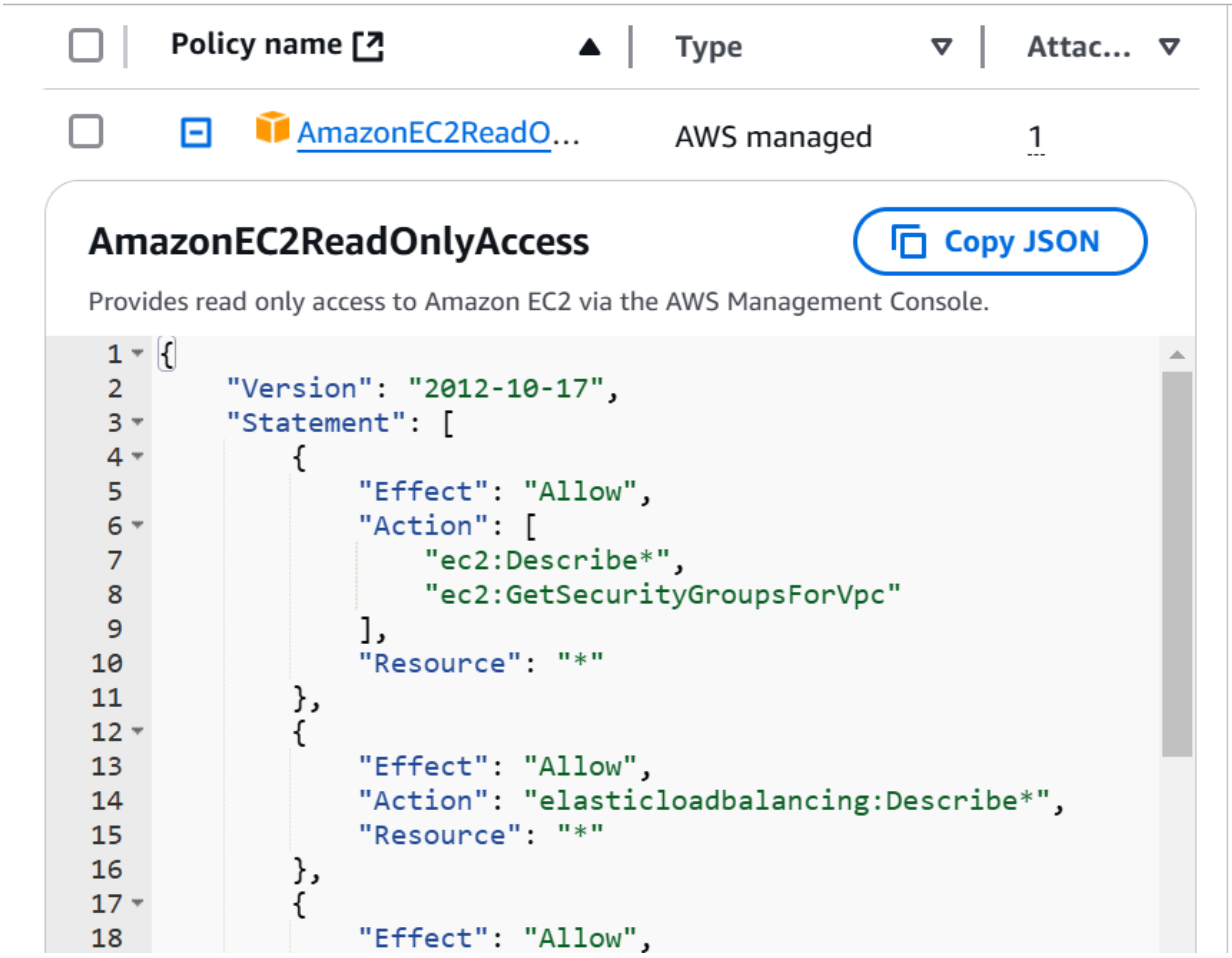


From these groups, click on the EC2-Support group. Choose the Permissions tab.

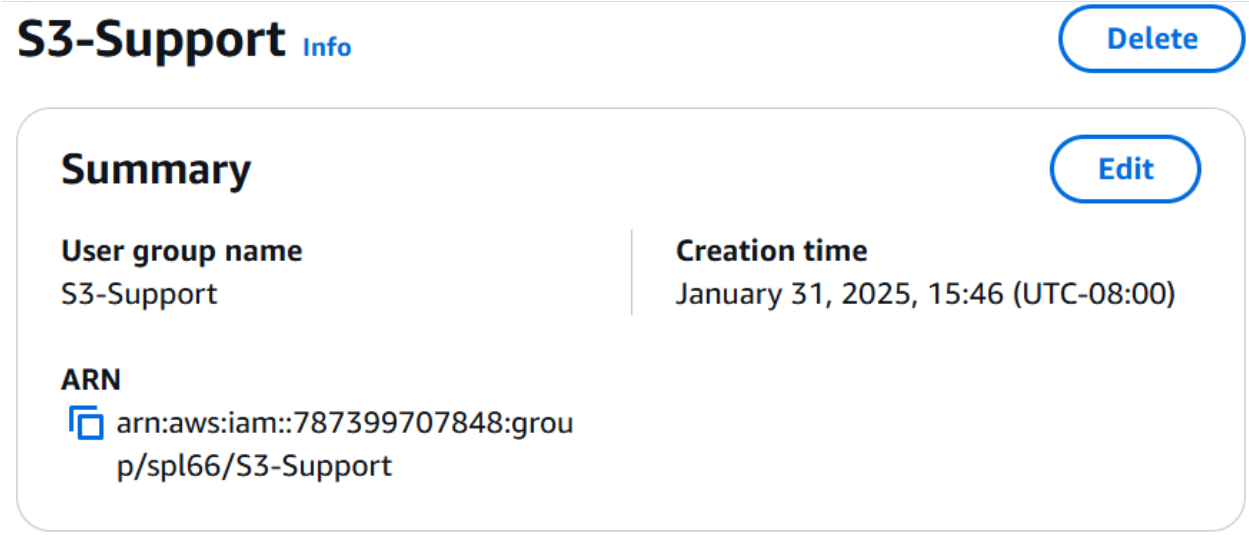


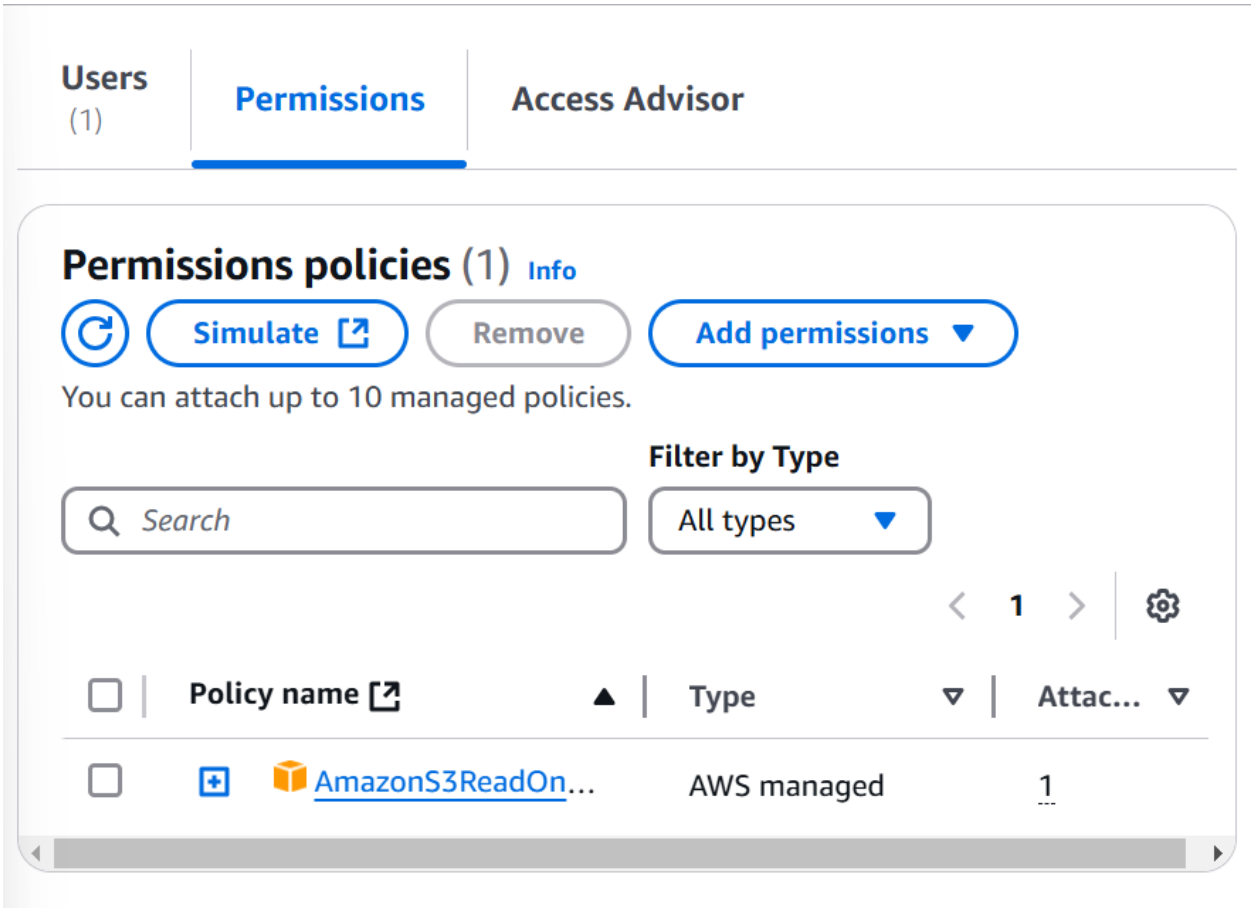


Click on the plus (+) icon to expand the AmazonEC2ReadOnlyAccess row. This policy is called a Managed Policy, and they are pre-built policies whose changes will be applied to all Users and Groups associated.

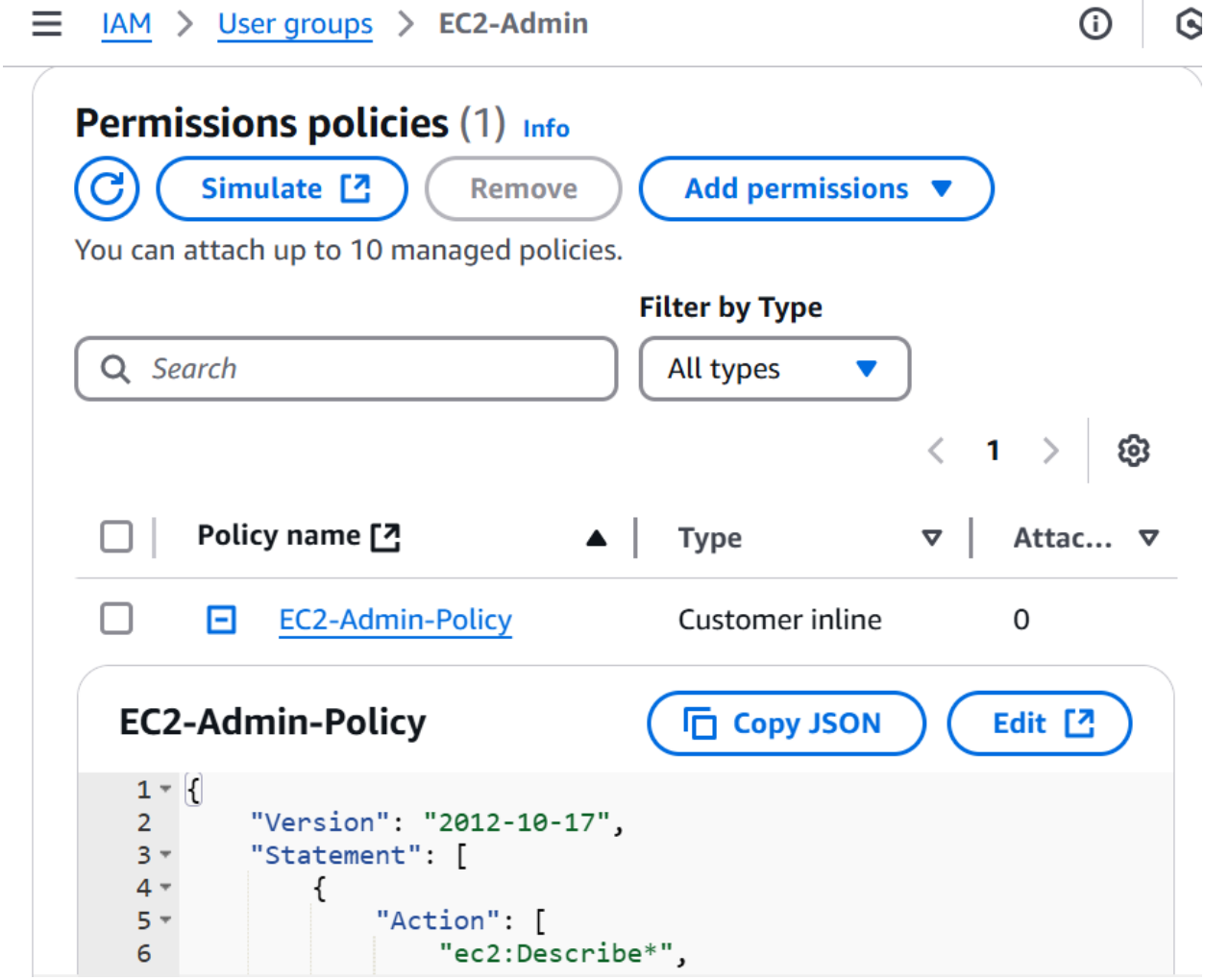


Enter the S3-Support group's Permissions tab, noticing how the S3-Support group has the AmazonS3ReadOnlyAccess policy.

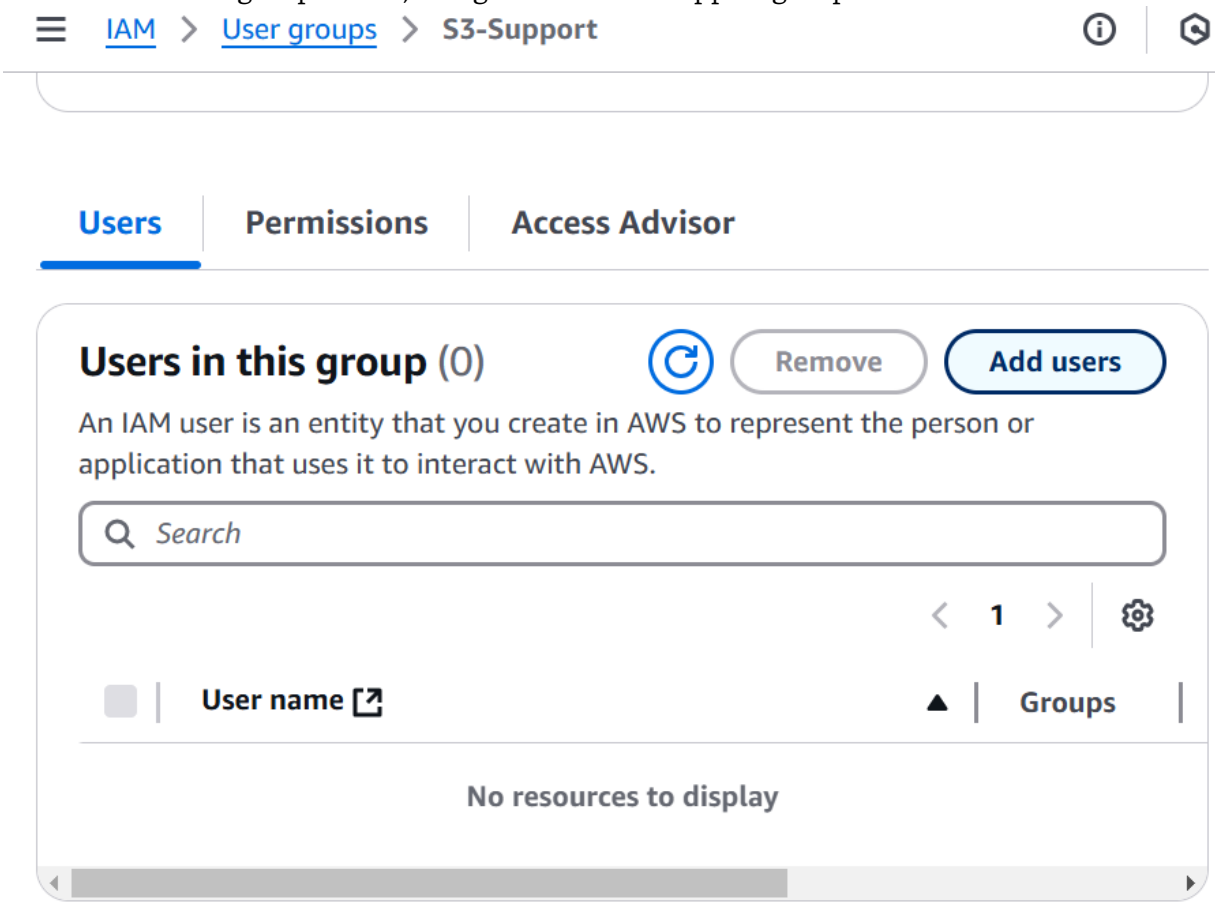



Users
(1)
Permissions
Access Advisor
Permissions policies (1) Info
Simulate
Remove
Add permissions
You can attach up to 10 managed policies.
Filter by Type
Search
All types
1
Policy name
Type
Attac...
AmazonS3ReadOn...
AWS managed
1

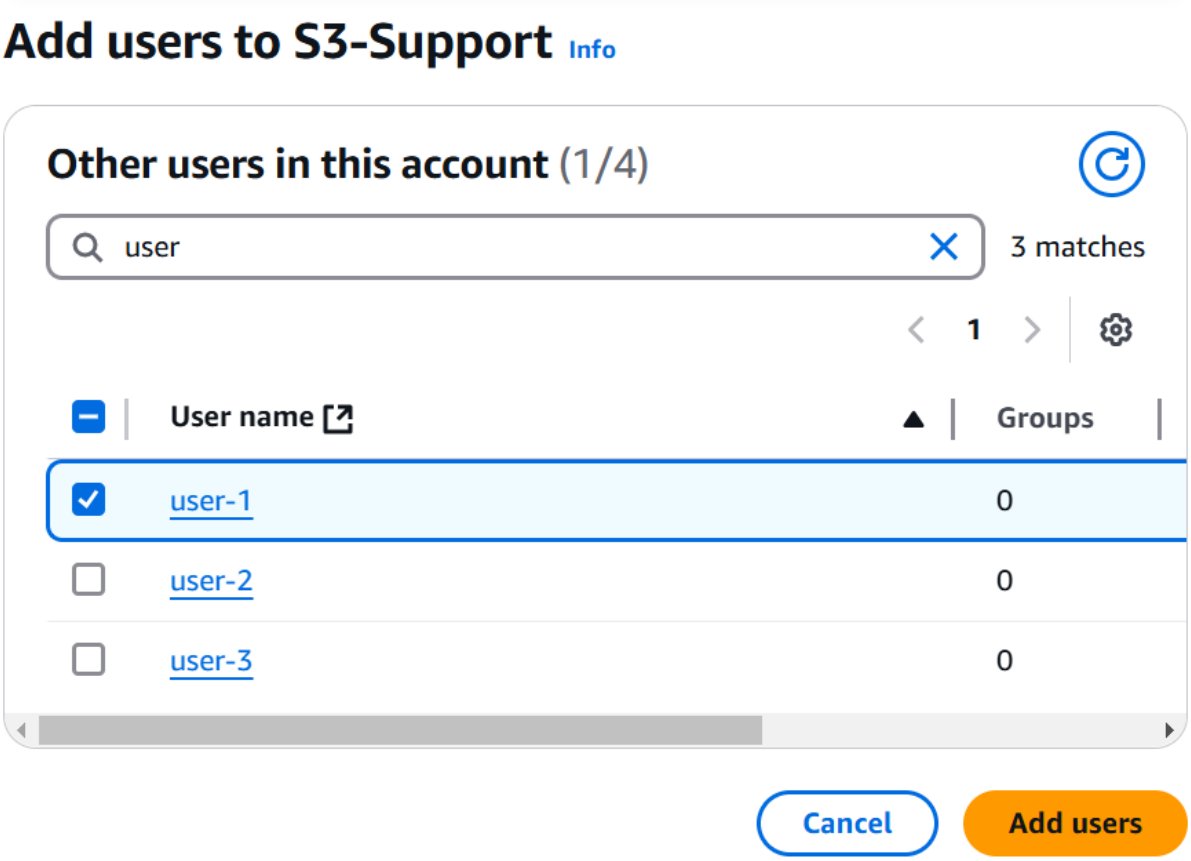
Next, go into the EC2-Admin group link and choose the Permissions tab. This group has an Inline Policy, which is used to apply permissions to exceptions.



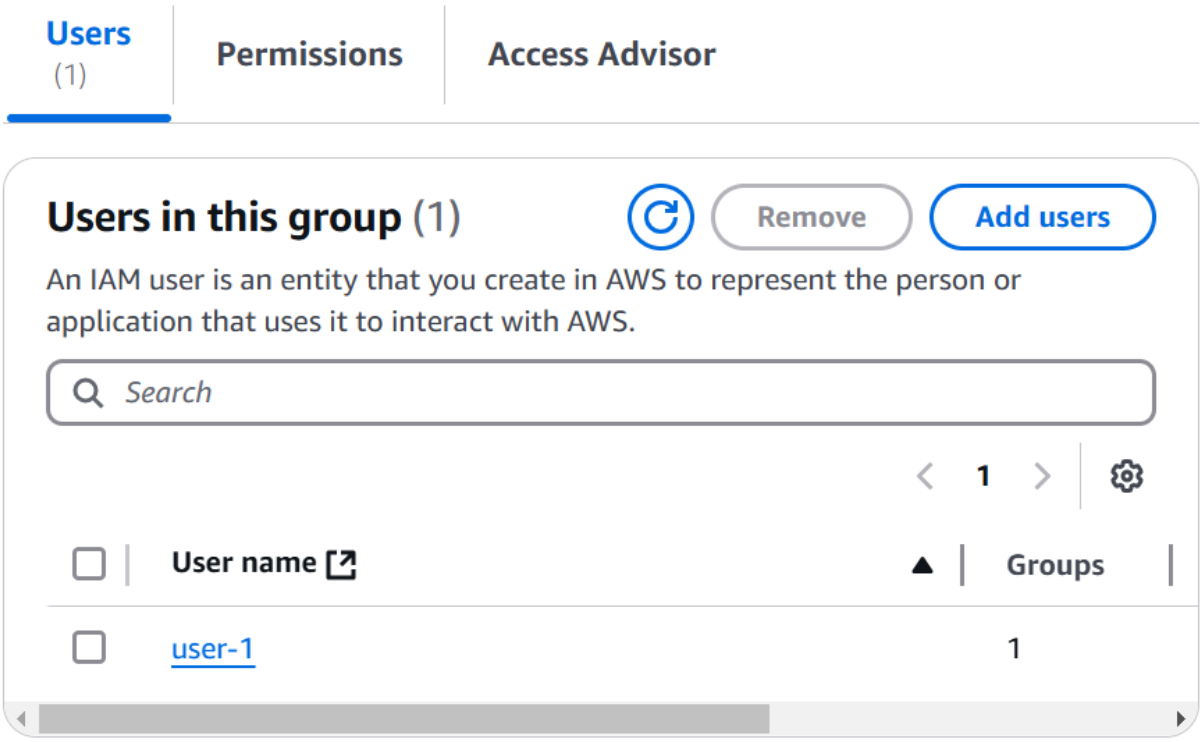
Let's add users to groups. First, navigate to the S3-Support group and choose the Users tab.



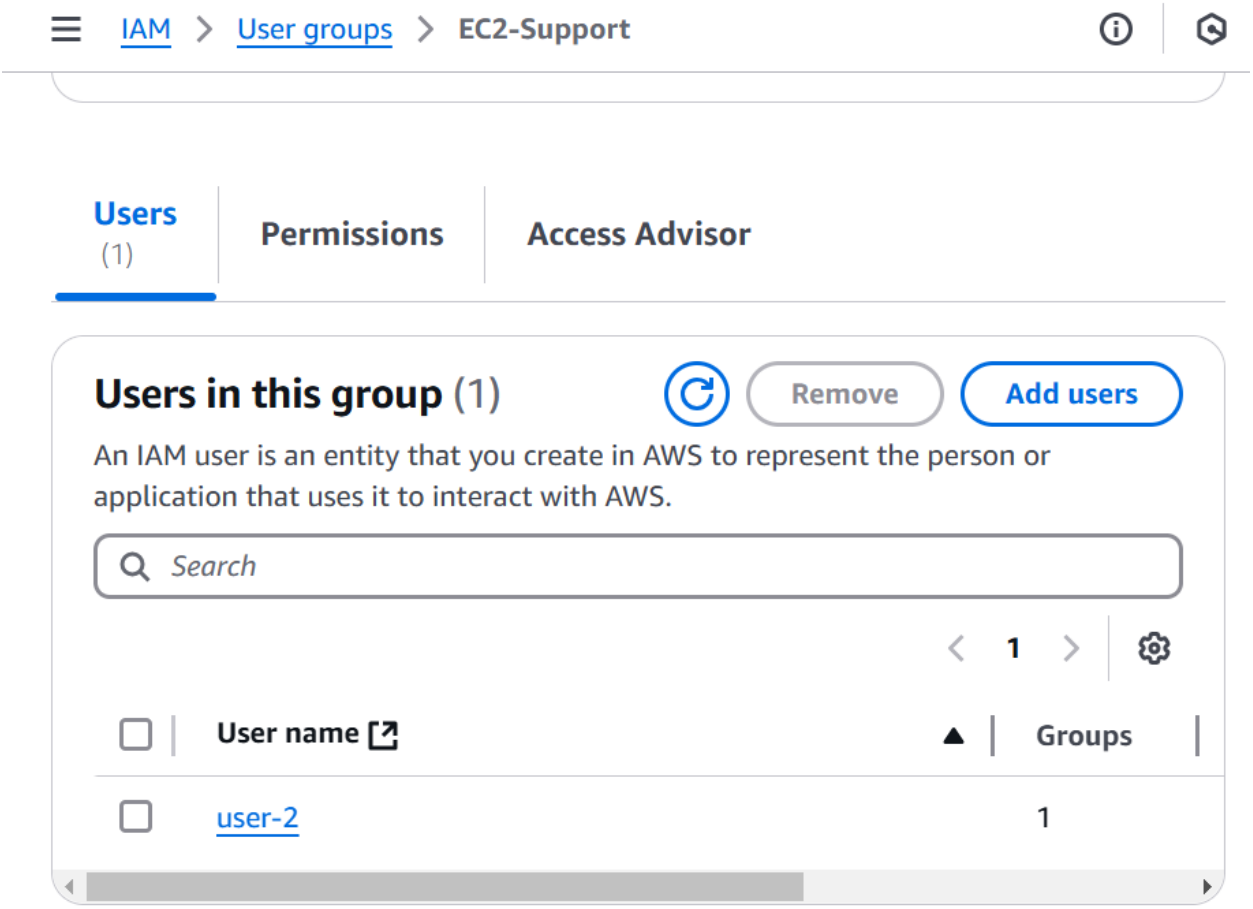
Add user-1 by selecting the button and checking the box next to user-1.



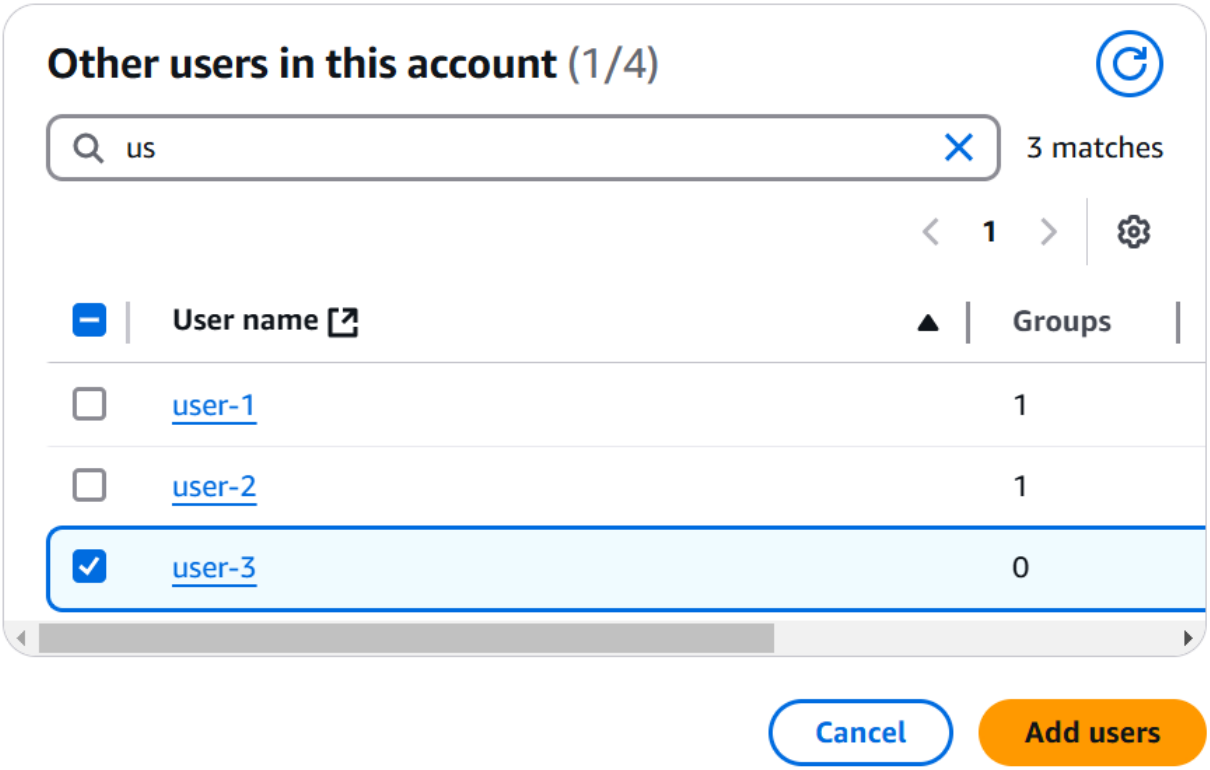
As you can see, user-1 has been added:

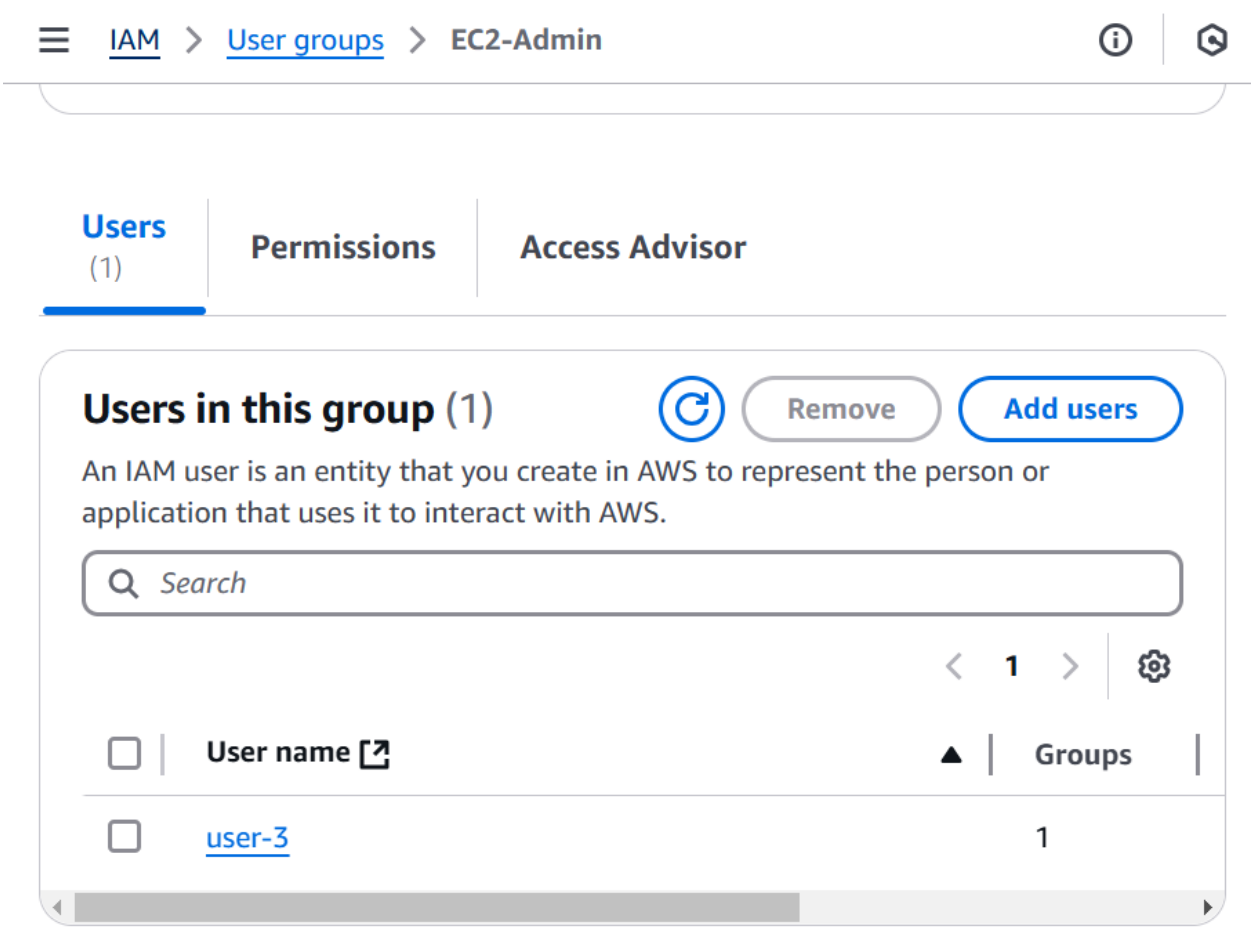


Add user-2 and user-3 to EC2-Support and EC2-Admin, respectively.

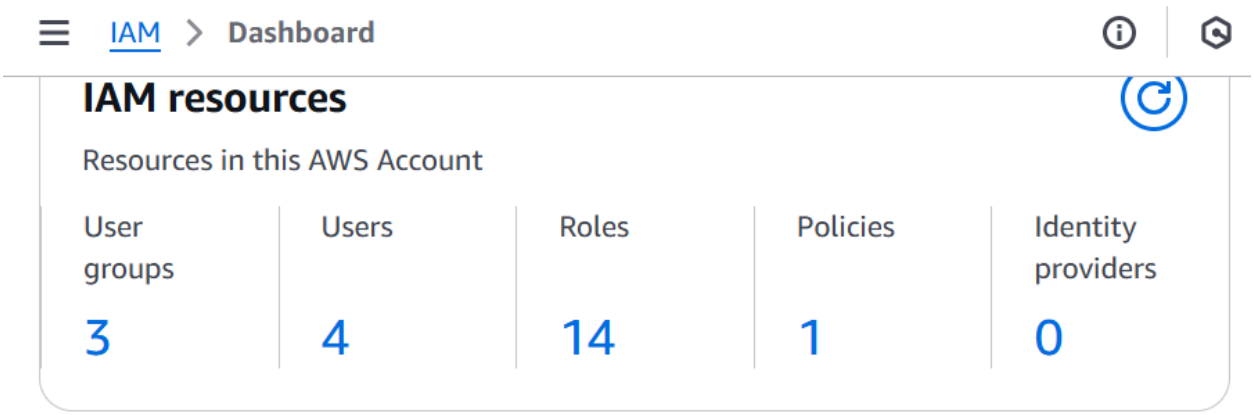


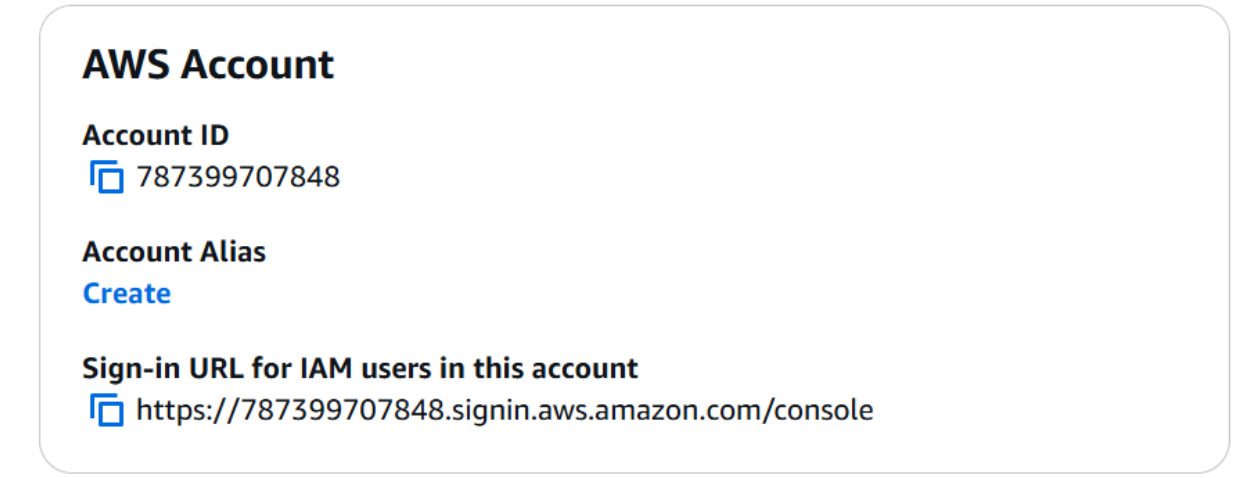
Add users to EC2-Admin Info



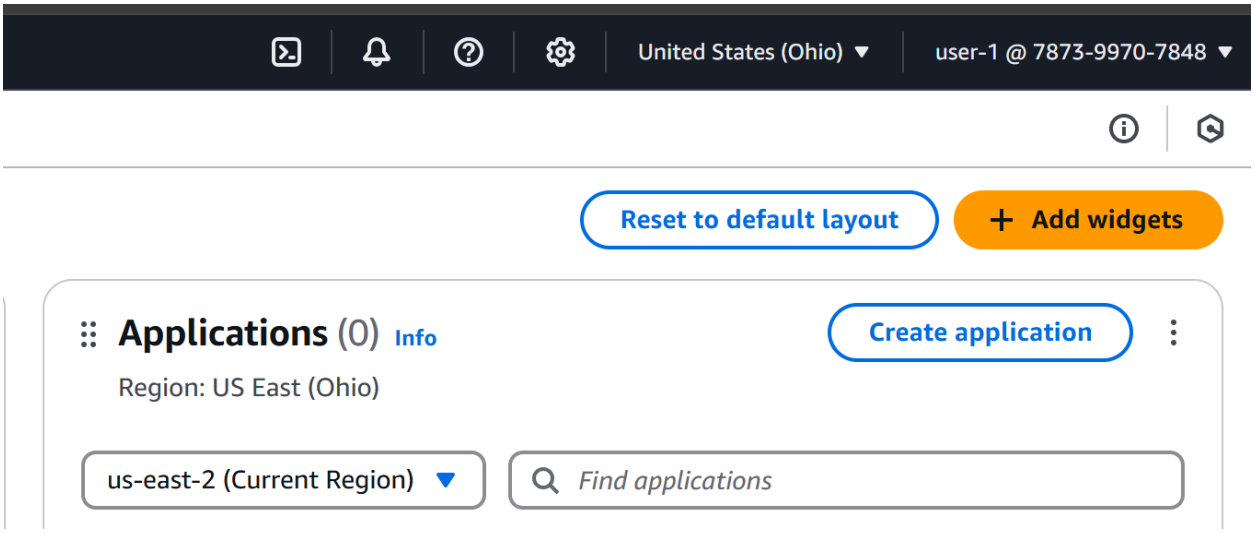


Next, navigate to Dashboard to access the sign-in URL for IAM users, opening it in a private window.

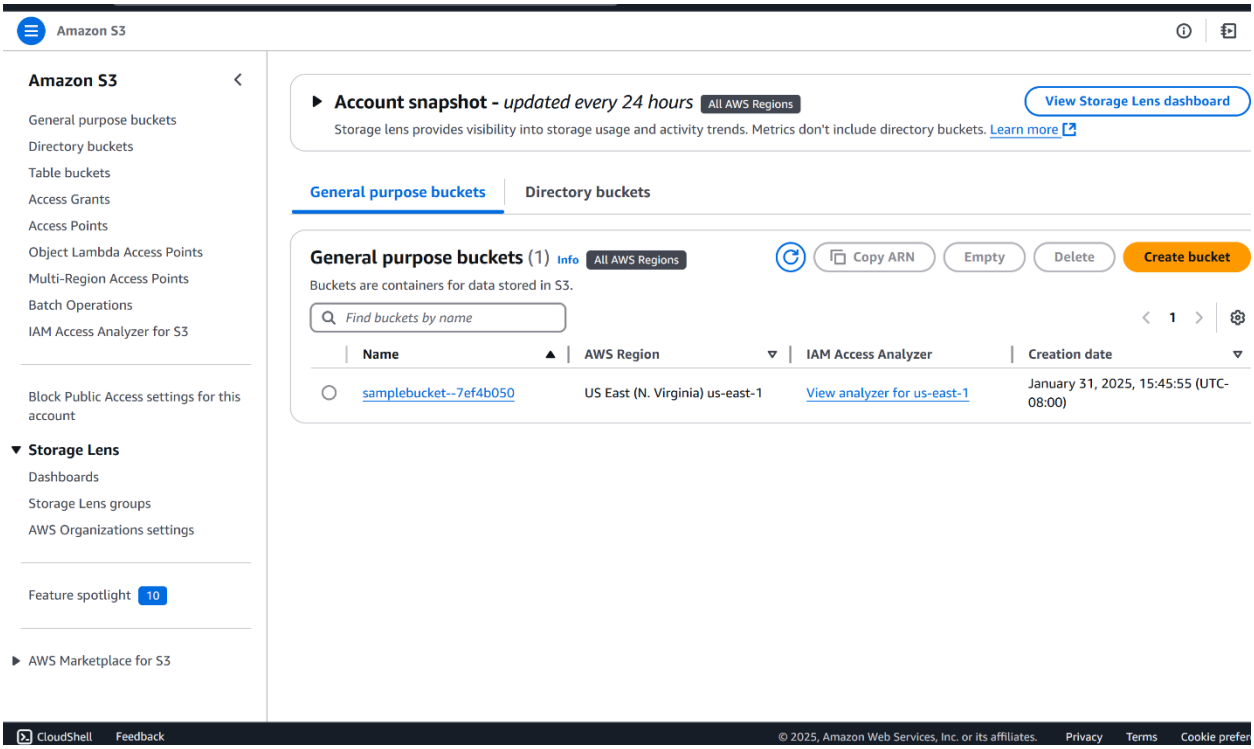


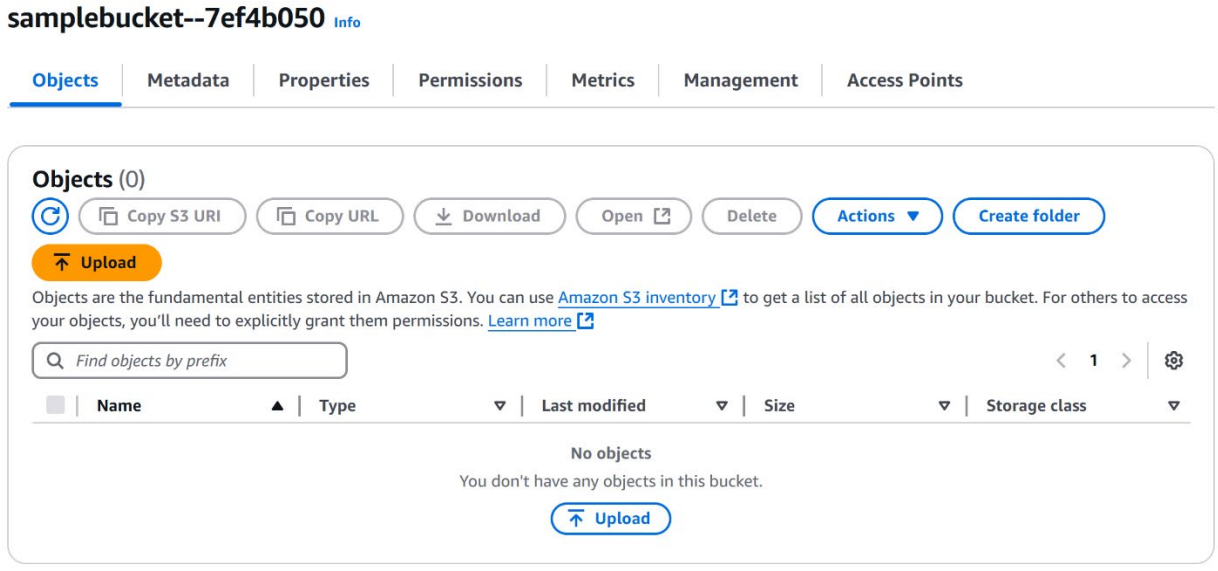


Access the user-1 account through the IAM user name **user-1** and password **Lab-Password1**.

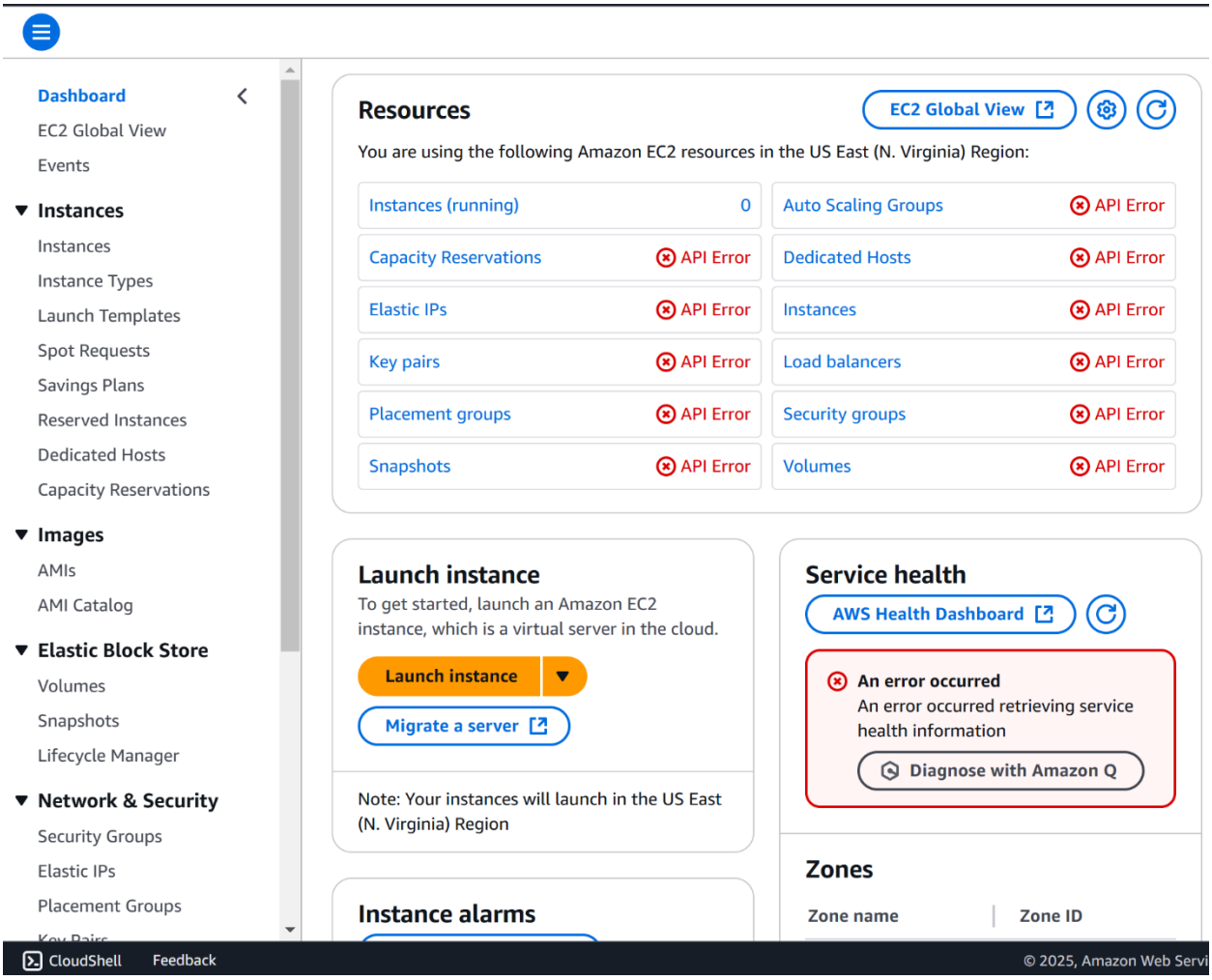


Search for and open the S3 console through the search box, in which the user should be able to view the list of buckets and contents as part of the S3-Support group.

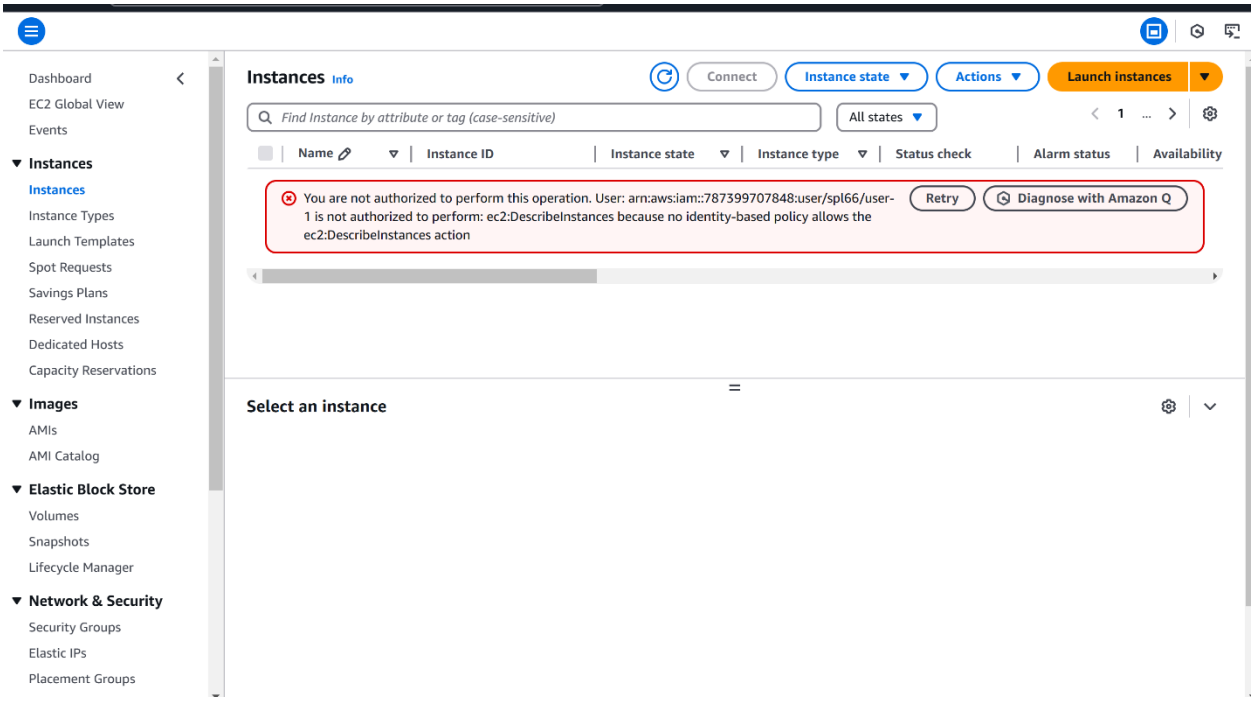




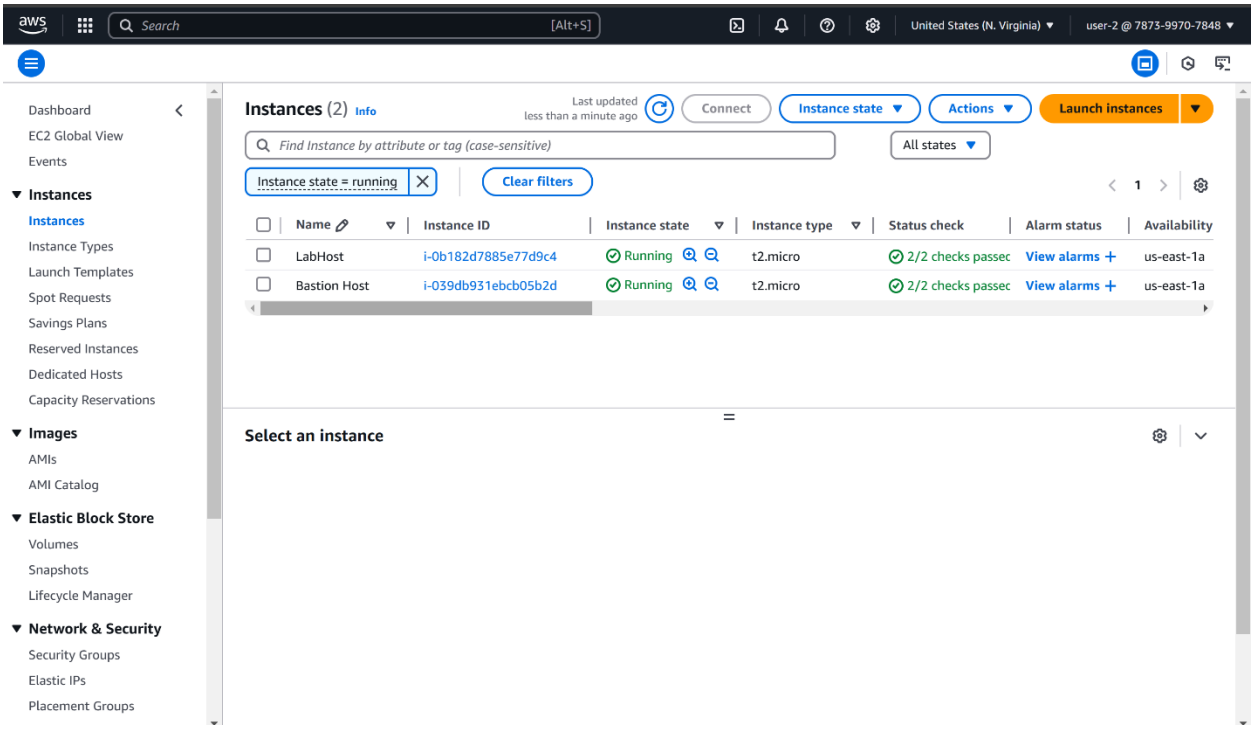
Next, go to the search box and choose EC2.



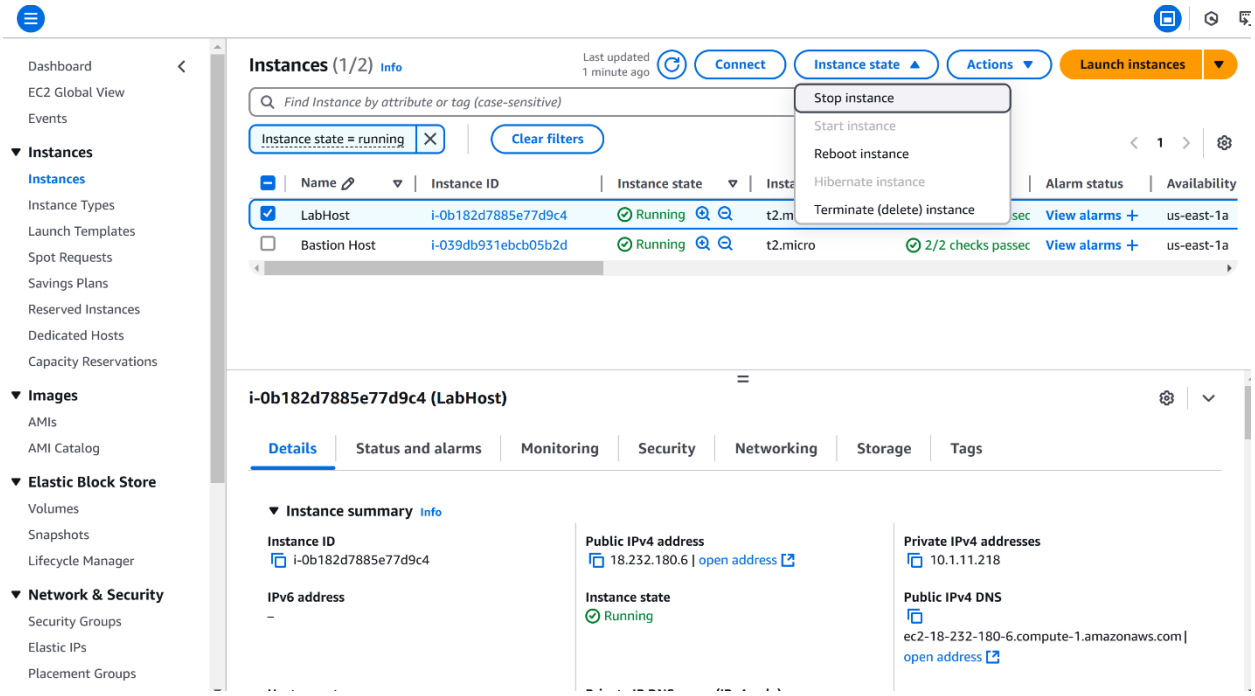
Choose Instances.



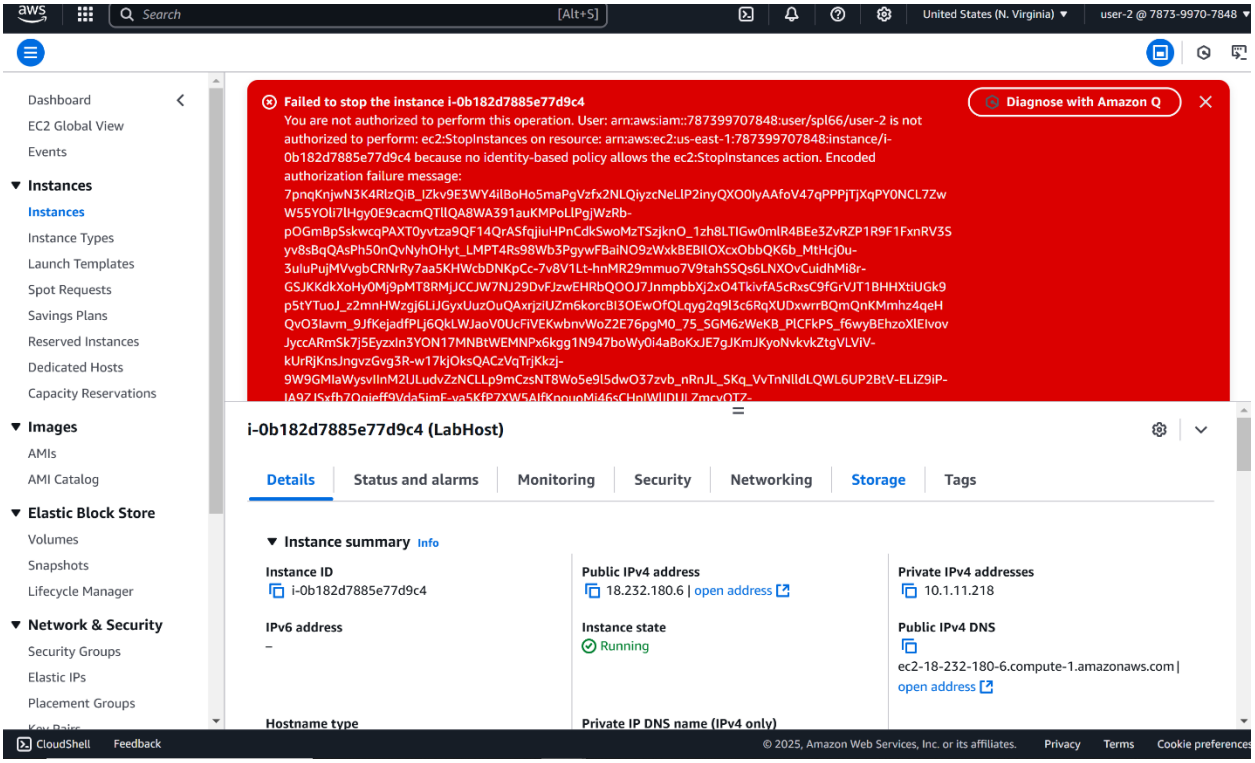
The user does not have permission to view instances. Now, sign out, use the URL from the IAM dashboard, and sign in as user-2 (IAM user name **user-2** and password **Lab-Password2**). Navigate to EC2 Instances.



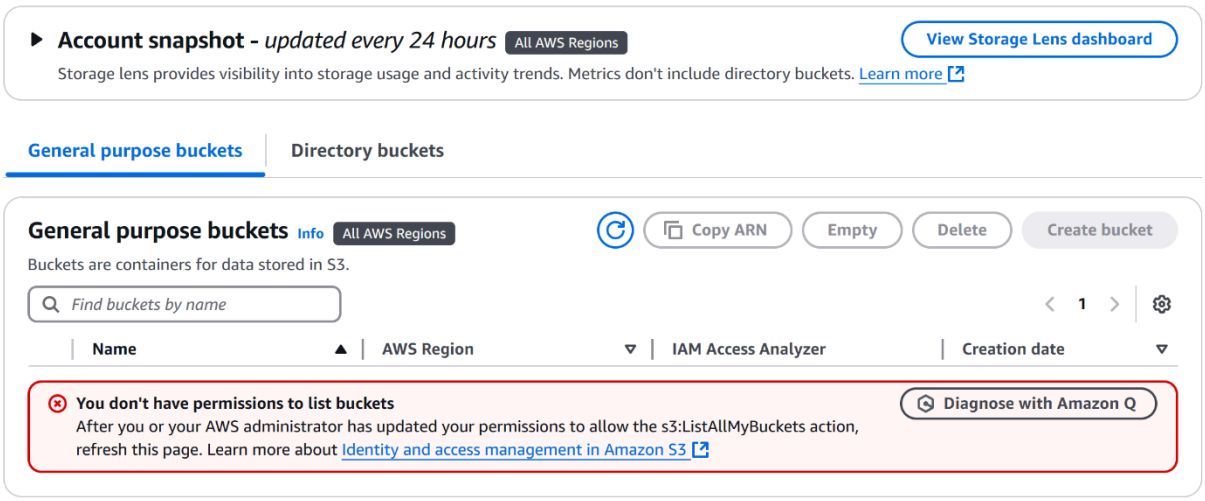
Select LabHost. In the Instance state menu, select Stop instance, as seen below:



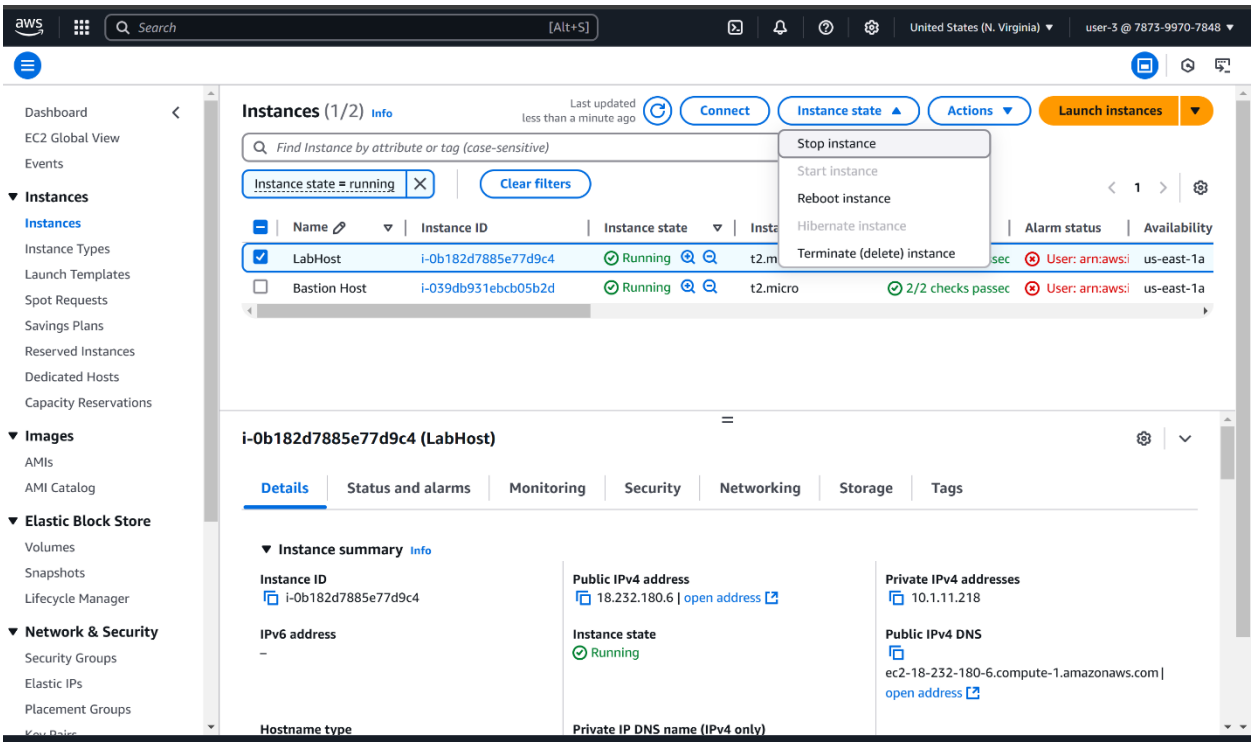
There is an error message because you are unauthorized to stop this instance.

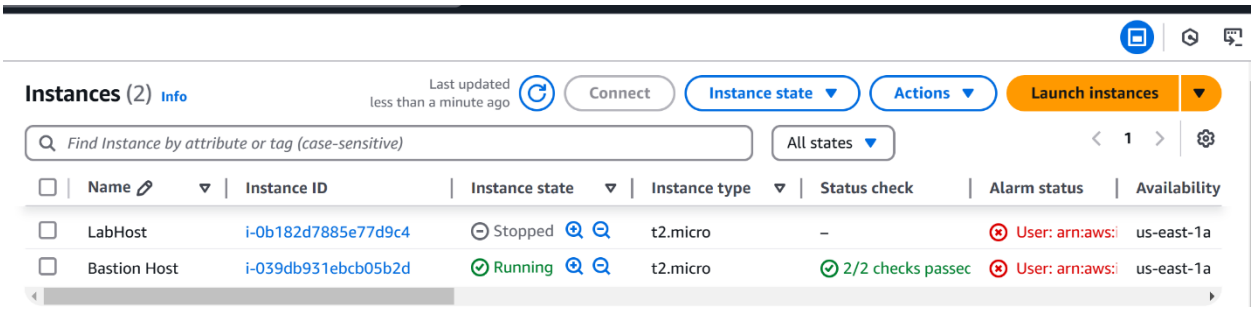


To check whether user-2 can also access S3 like user-1, search for the S3 console and view the buckets. It turns out that user-2 does not have permission to view the buckets.



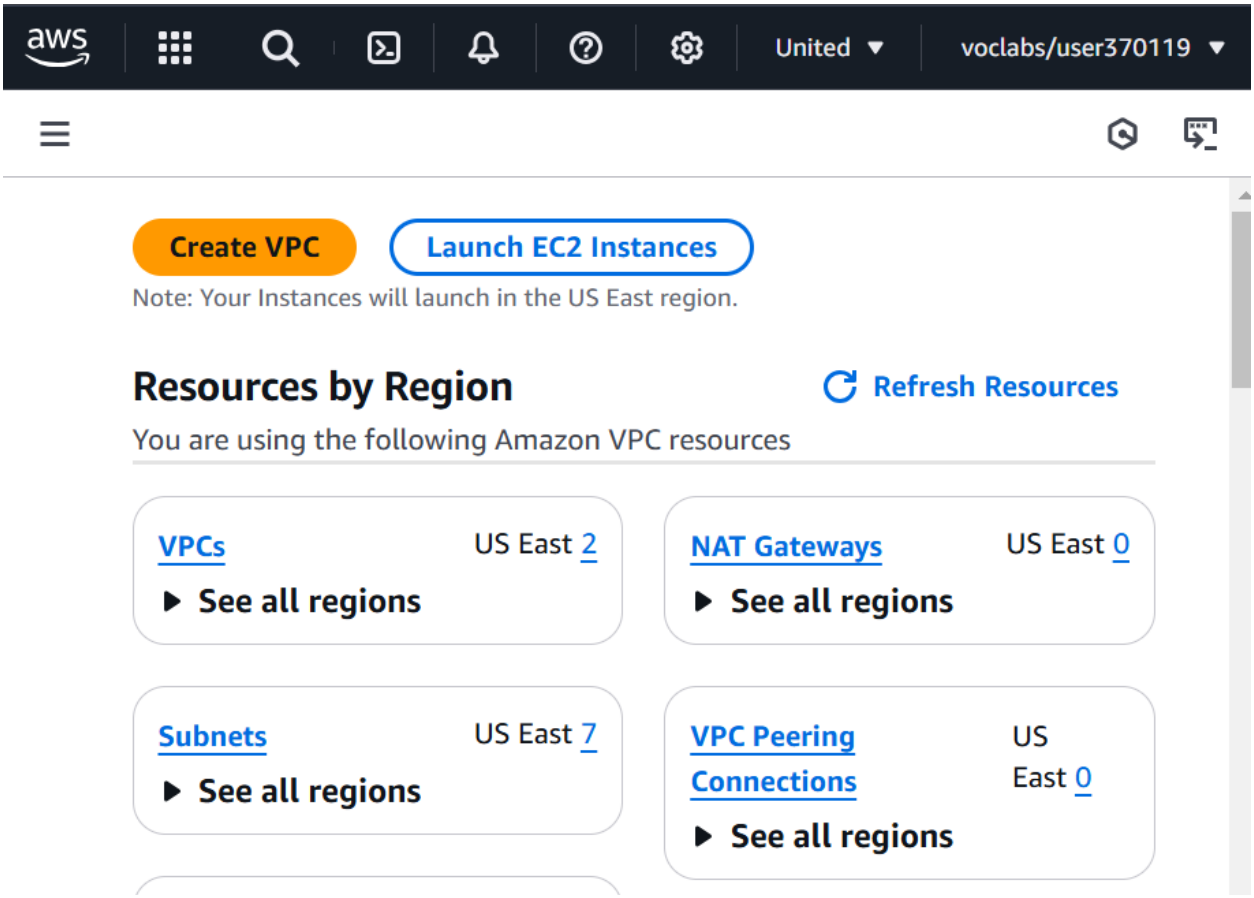
Sign out and sign in again as user-3 with the password Lab-Password3. Open the EC2 console and select Instances. You should be able to stop the LabHost instance this time.



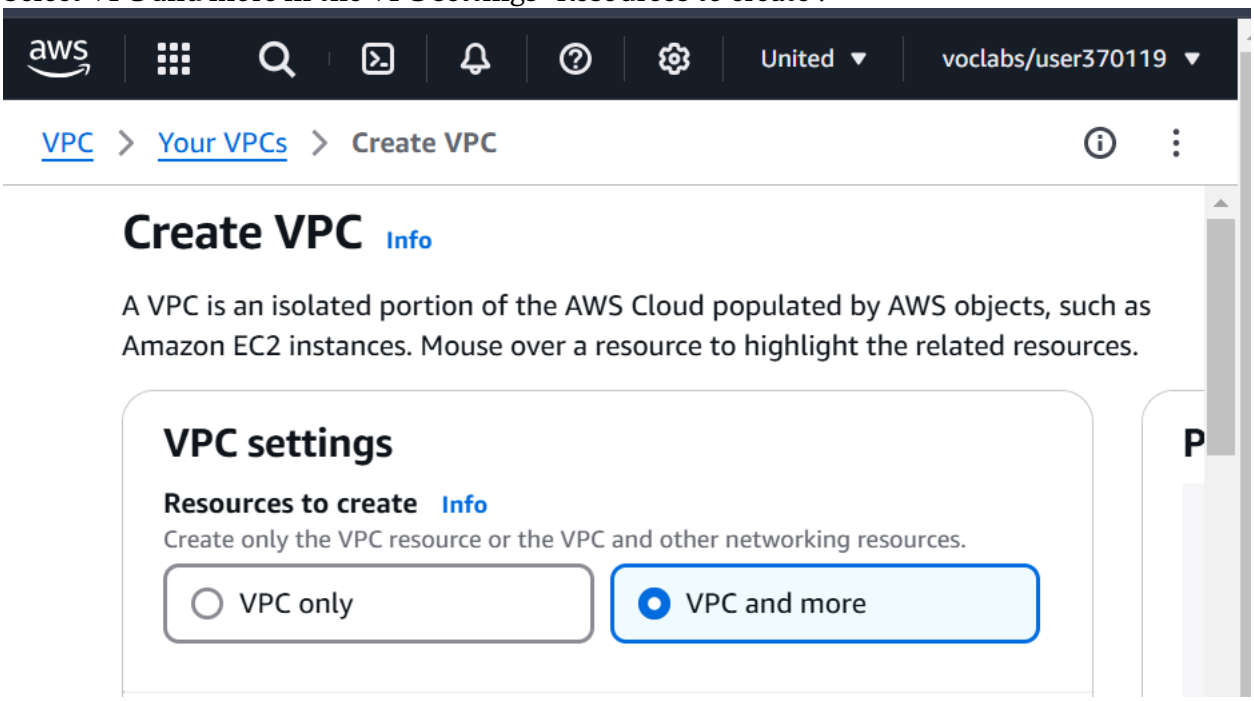


LAB 2

Navigate to the VPC console using the search box and click on Create VPC.



Select VPC and more in the VPC settings “Resources to create”.



Next, under Name tag auto-generation, set the value that’s auto-generated to **lab**. Keep the IPv4 CIDR block setting, number of public subnets, and number of private subnets. Set the number of availability zones to **1**.

VPC settings

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

- ○ VPC only
- ◉ VPC and more

Name tag auto-generation Info

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

- ☑ Auto-generate

lab

IPv4 CIDR block Info

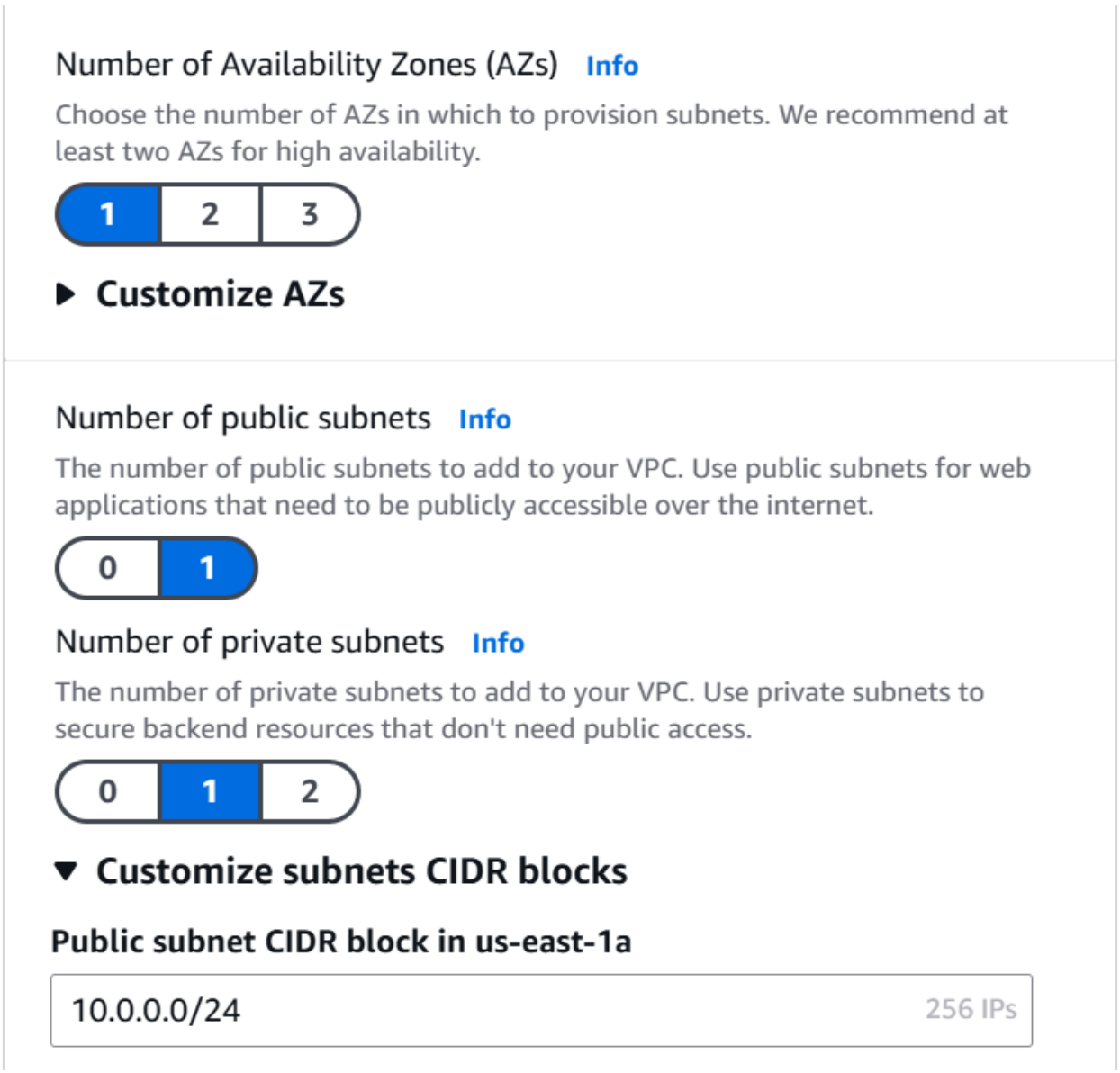
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

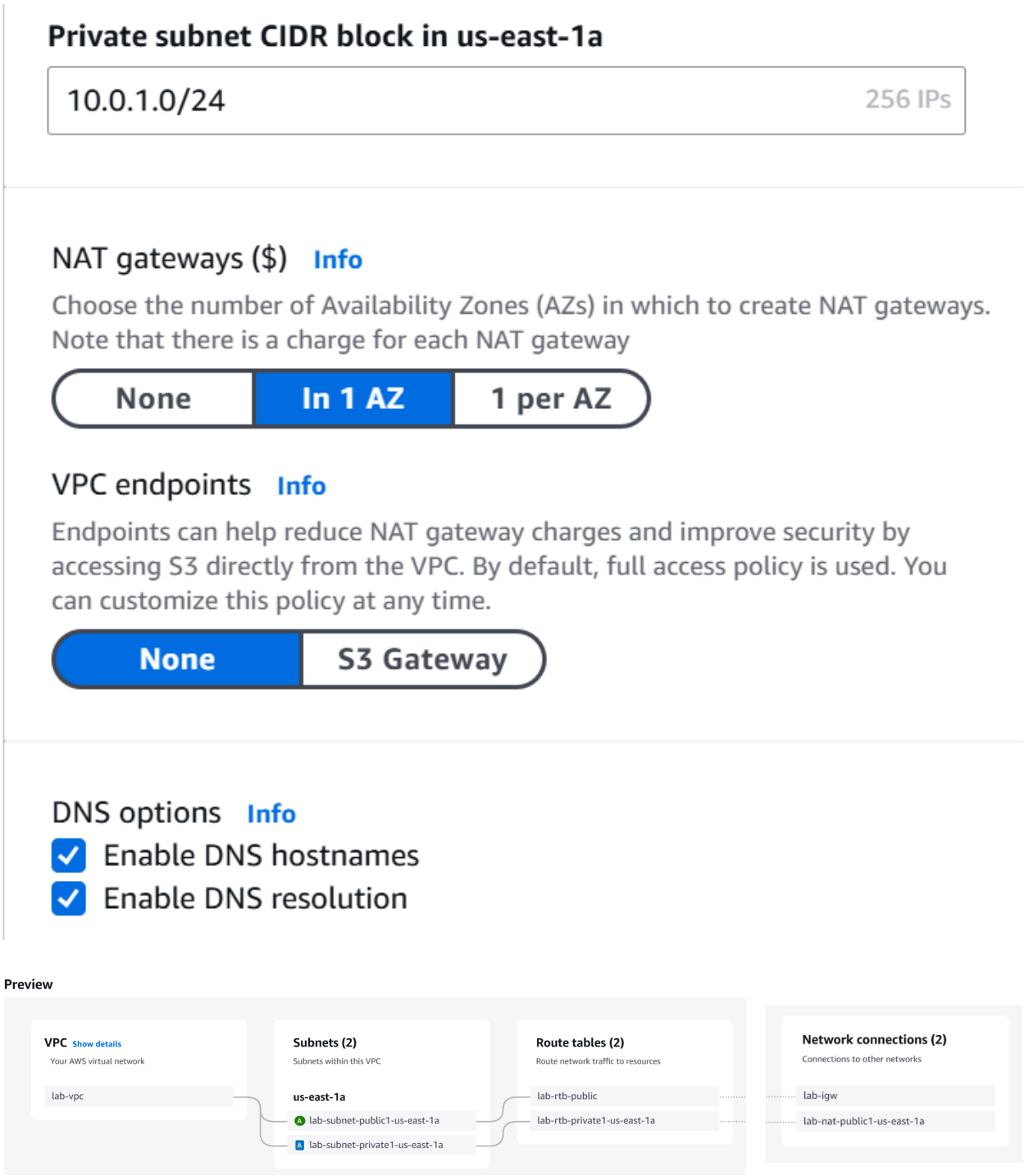
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info

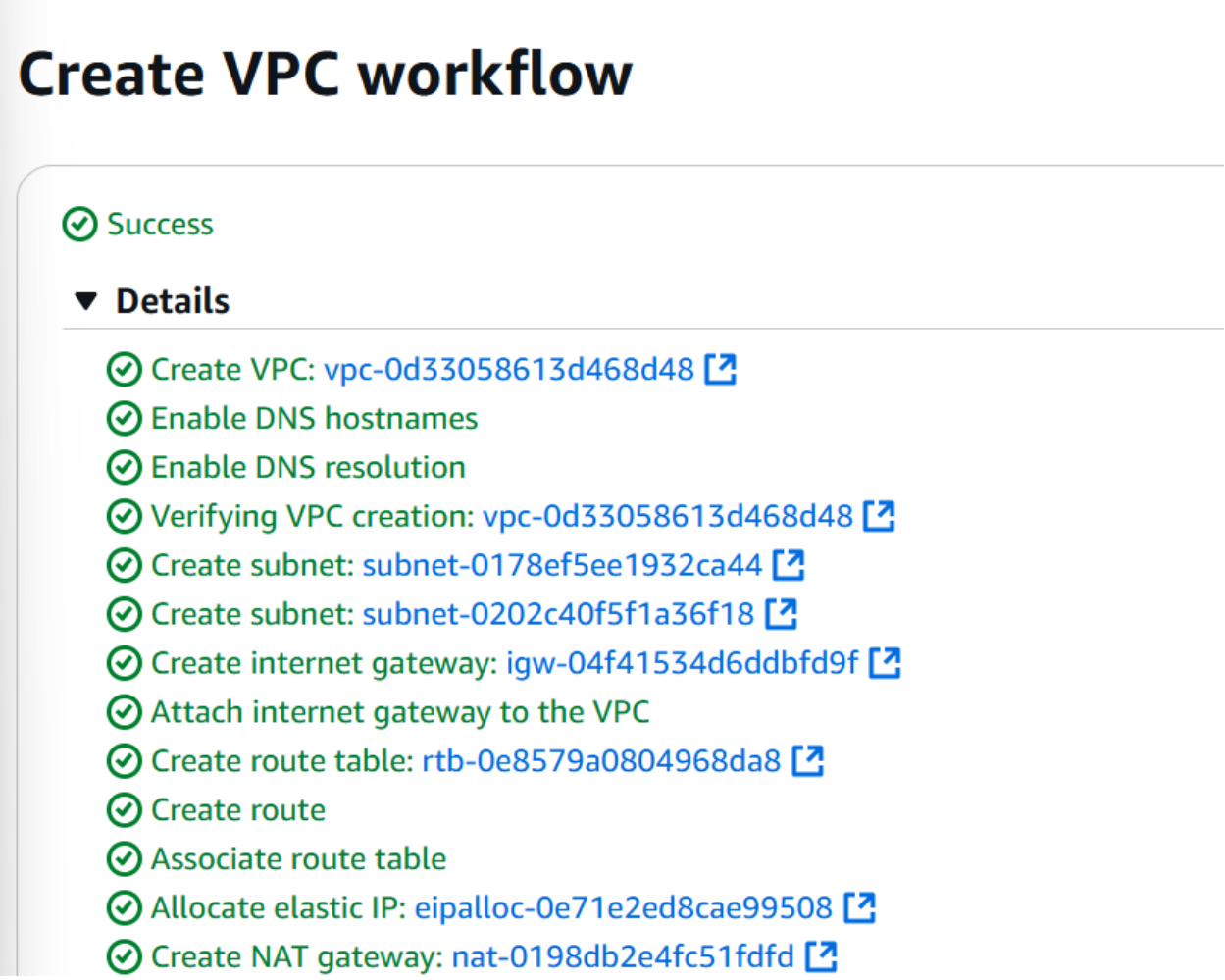
- ◉ No IPv6 CIDR block
- ○ Amazon-provided IPv6 CIDR block



Expanding the Customize subnets CIDR blocks section, change the Public subnet CIDR block in us-east-1a to **10.0.0.0/24** and change Private subnet CIDR block in us-east-1a to **10.0.1.0/24**. Set NAT gateways to **In 1 AZ**. Set VPC endpoints to **None**. Enable both DNS hostnames and DNS resolution.



After clicking Create VPC, make sure it is complete before continuing:



Choose View VPC to look at the settings using the VPC console links. Some options are to select Subnets and Route tables to look through those details.

subnet-0178ef5ee1932ca44 / lab-subnet-public1-us-east-1a

Actions

Details

Subnet ID
subnet-0178ef5ee1932ca44

IPv4 CIDR
10.0.0.0/24

Availability Zone
us-east-1a

Route table
rtb-0e8579a0804968da8 | lab-rtb-public

Auto-assign IPv6 address
No

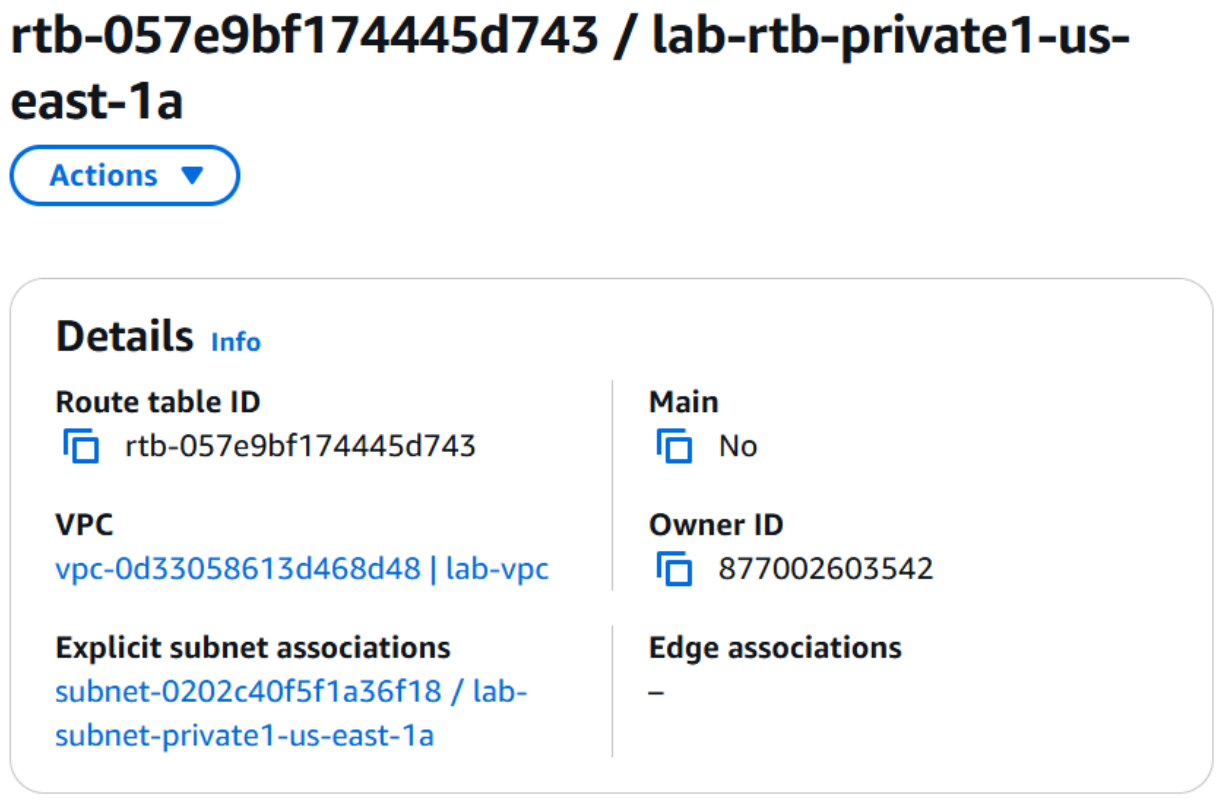
Subnet ARN
arn:aws:ec2:us-east-1:877002603542:subnet/subnet-0178ef5ee1932ca44

Available IPv4 addresses
250

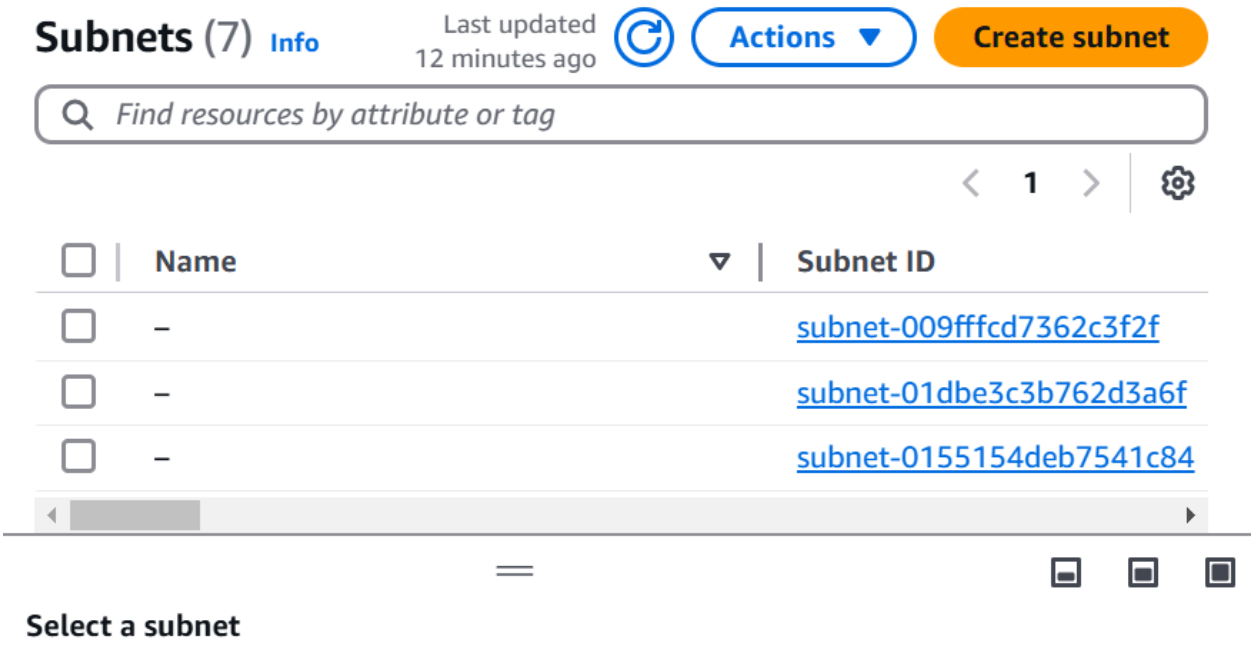
Availability Zone ID
use1-az2

Network ACL
acl-04dc06315b61731cc

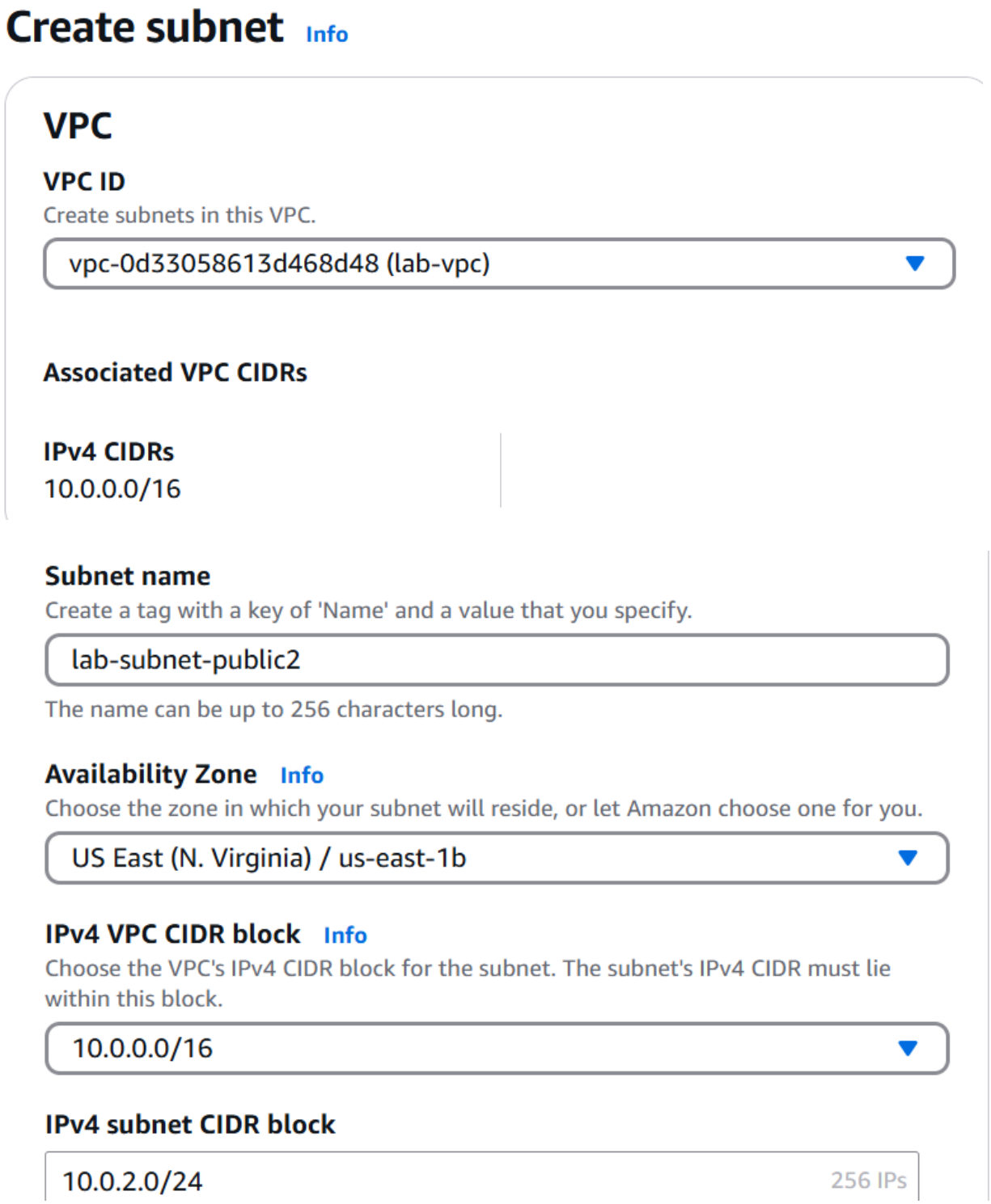
Auto-assign customer-owned IPv4 address



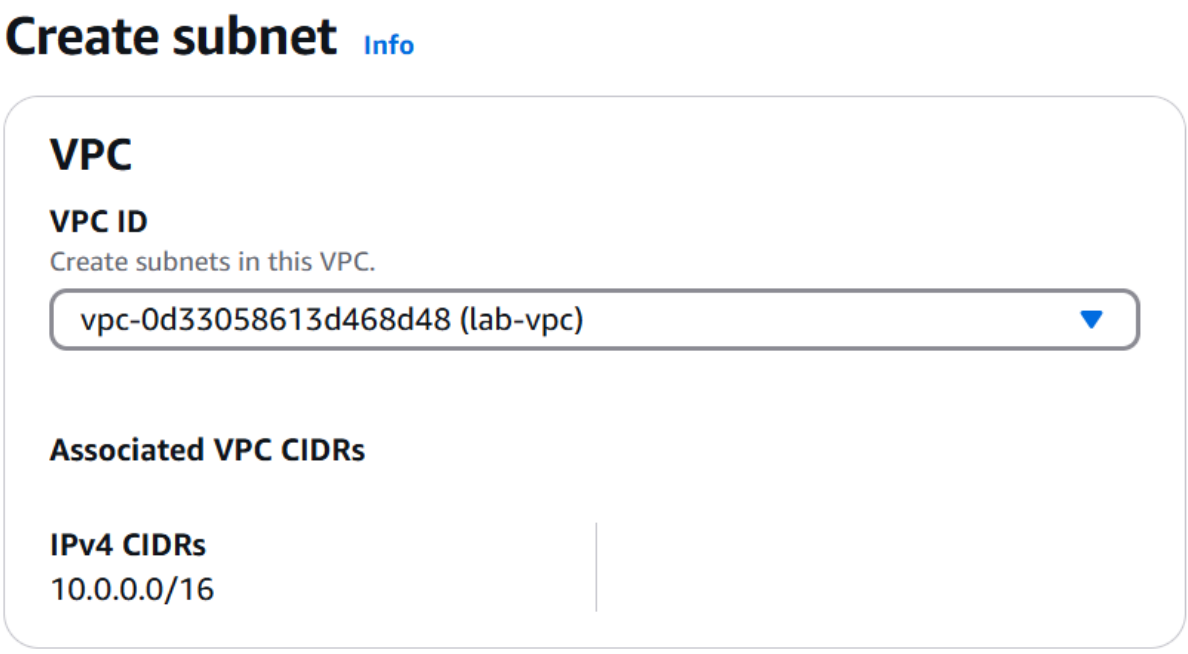
Now, create additional subnets by selecting Subnets from the left navigation pane. Choose Create subnet.

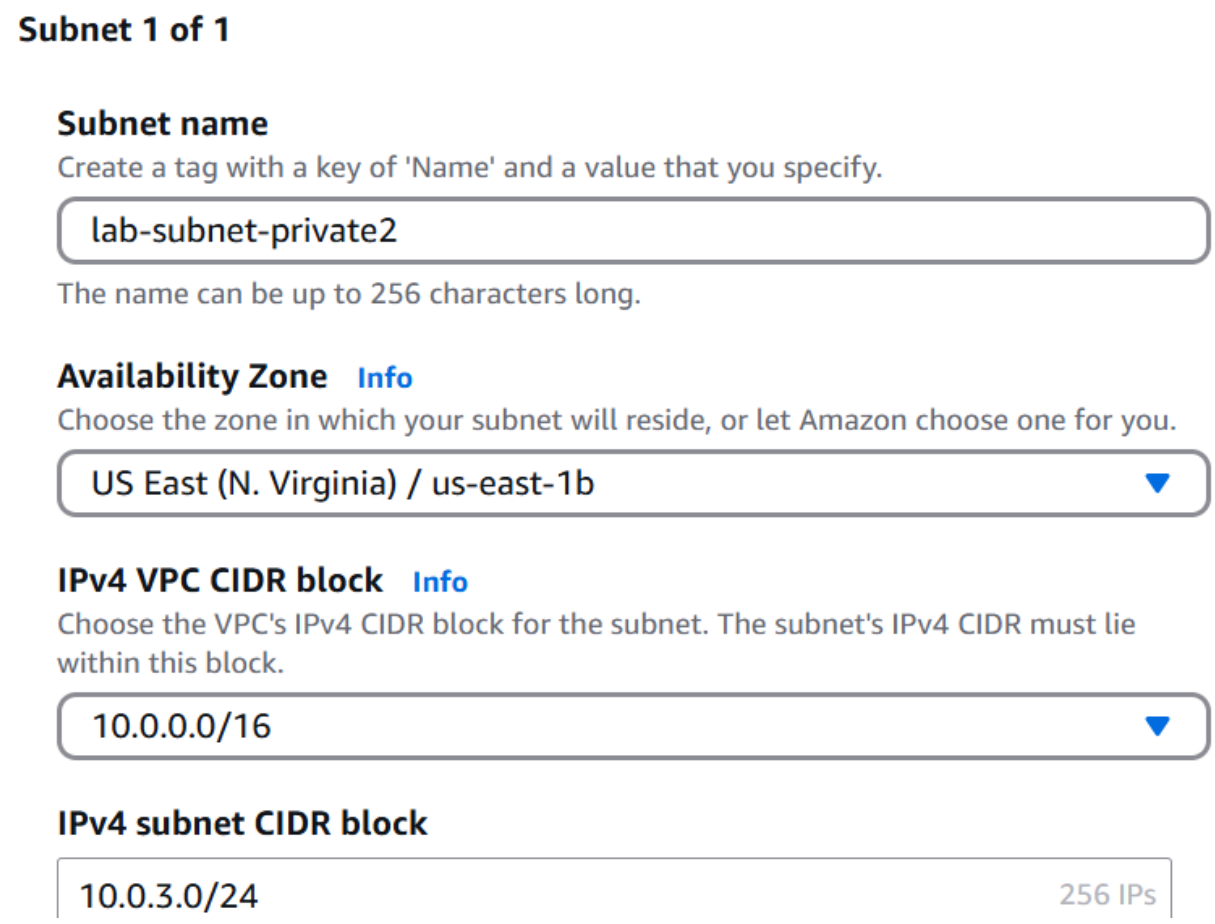


Configure the following: VPC ID (use **lab-vpc**), subnet name (**lab-subnet-public2**), Availability Zone (select the second Availability Zone from the top), and IPv4 CIDR block (**10.0.2.0/24**).

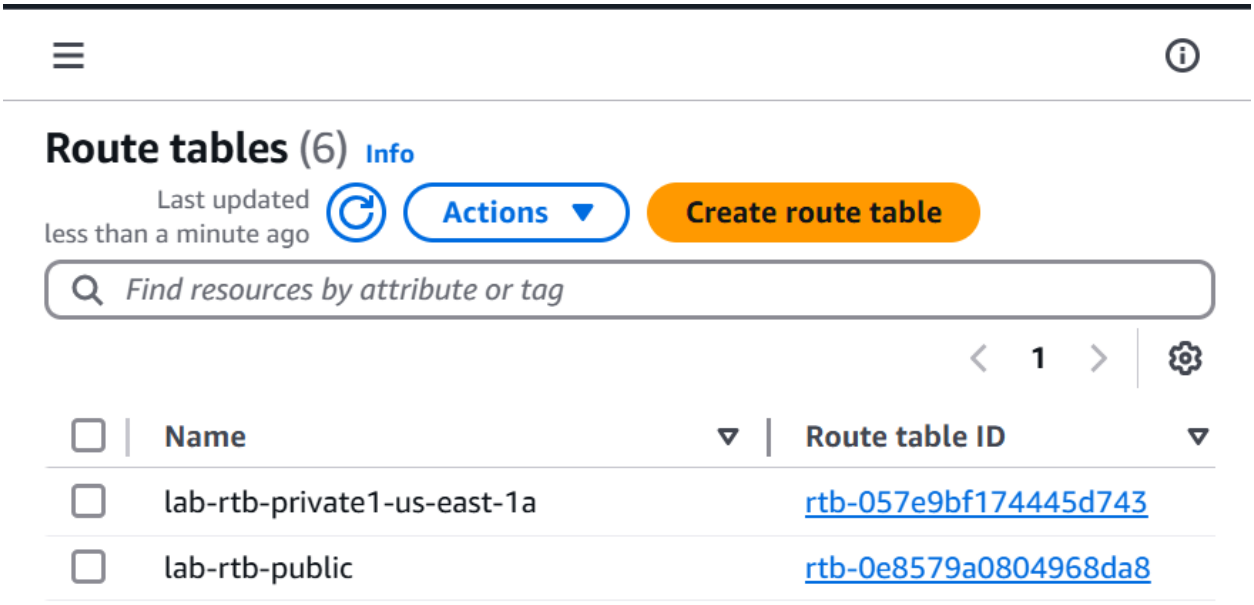


Next, create a second private subnet (since a second public subnet was created). Use the same settings, but change the subnet name to **lab-subnet-private2** and the IPv4 CIDR block to **10.0.3.0/24**.

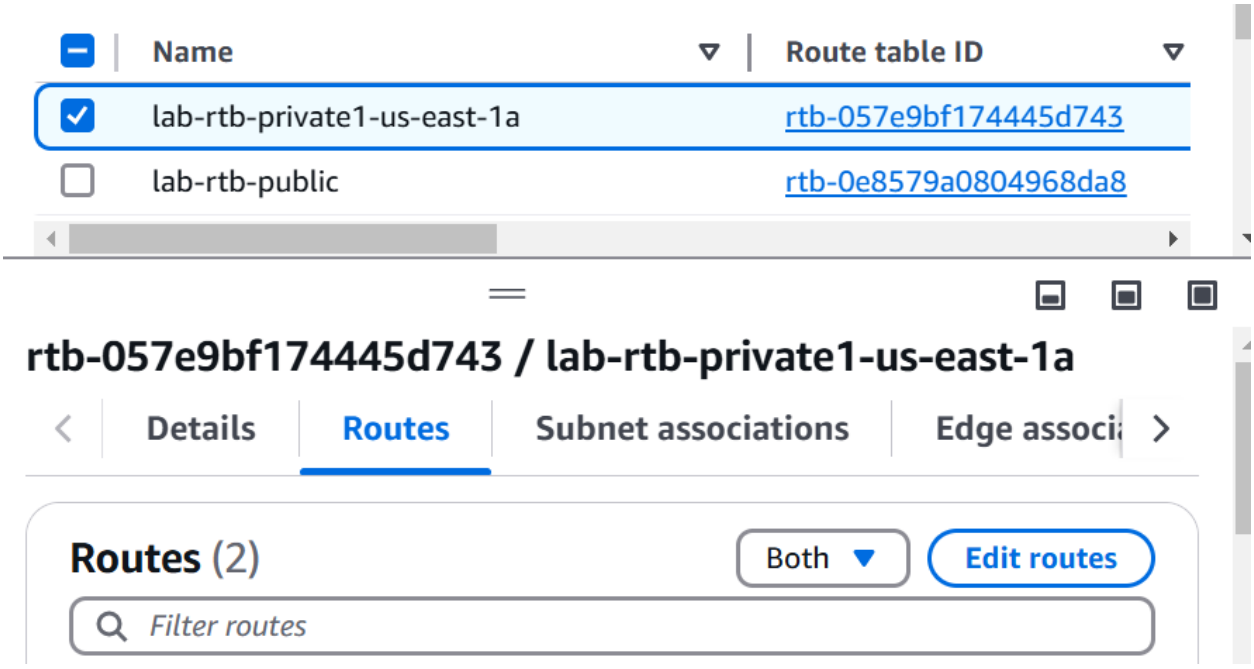




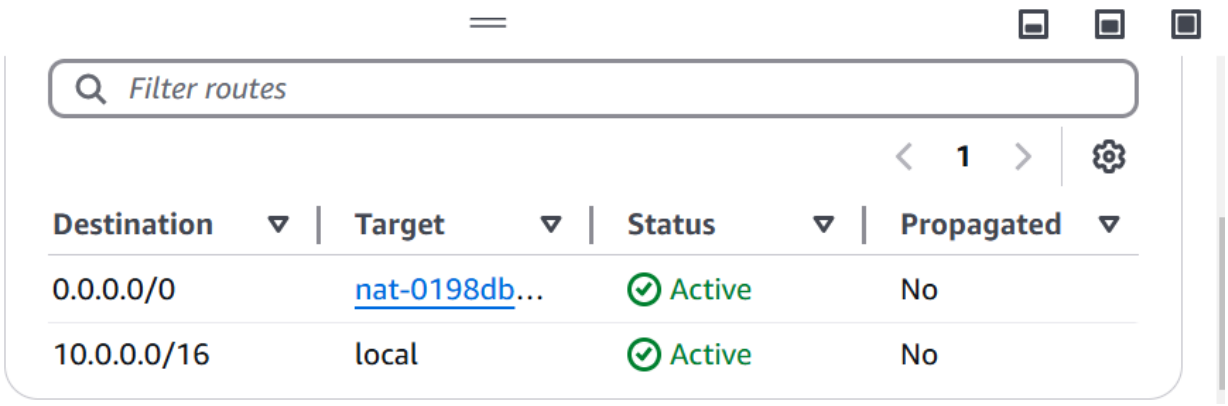
Next, in the left navigation pane, select Route tables.



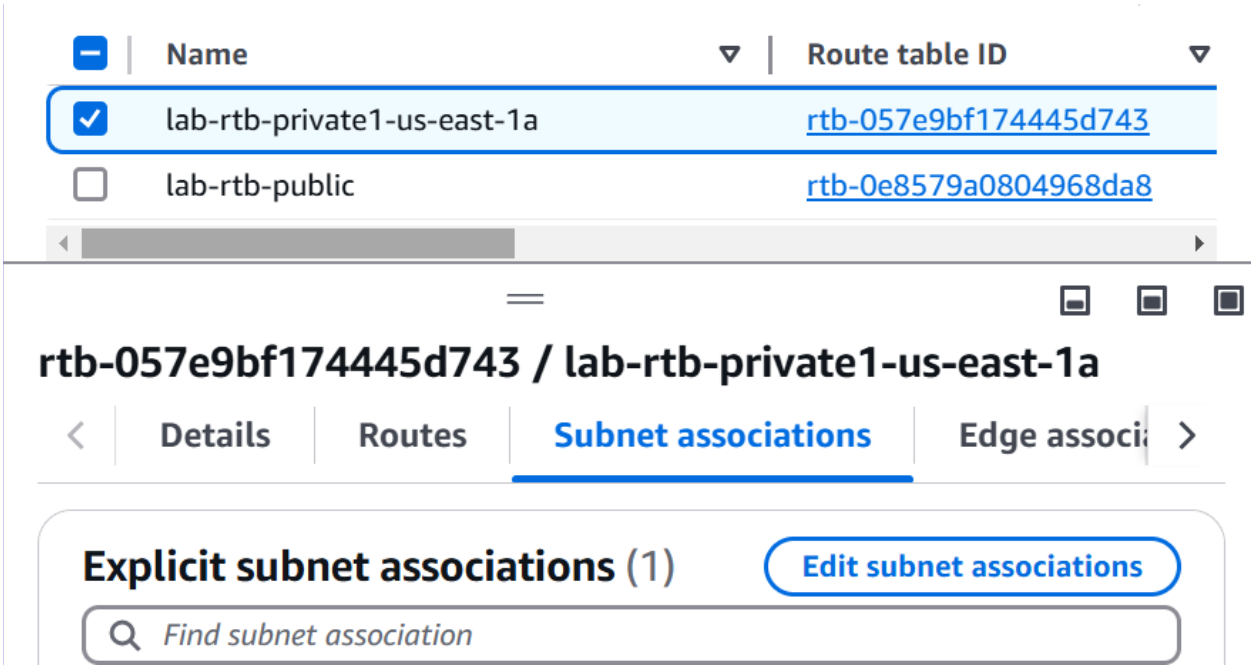
Check the box for the lab-rtb-private1-us-east-1a route table.



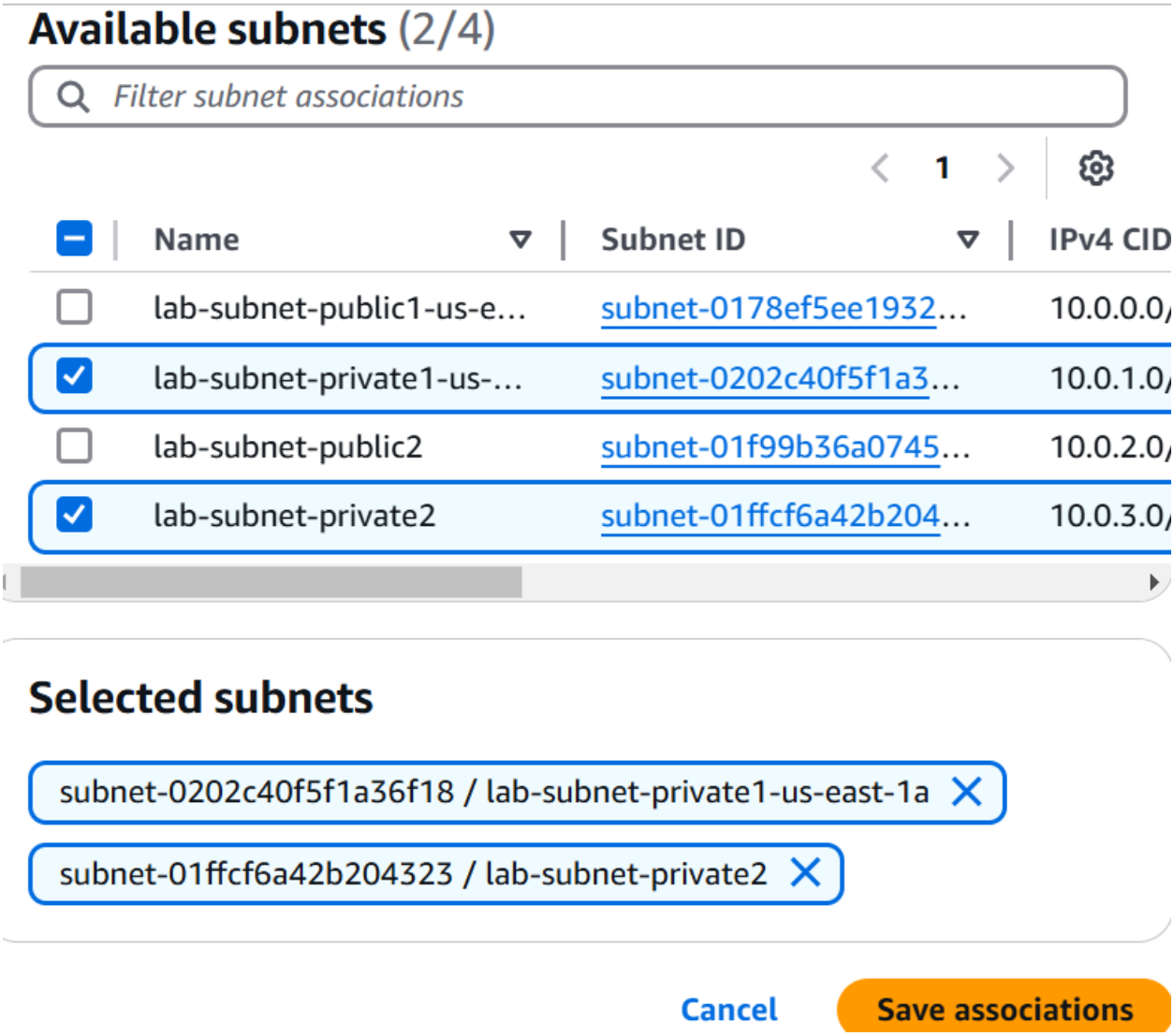
Notice that in the Routes tab pictured above, Destination 0.0.0.0./0 is set to Target nat-xxxxxxxx, showing that traffic going towards the Internet, which has an IP address of 0.0.0.0/0, will be sent to the NAT Gateway before being forwarded to the Internet.



Next, click on the Subnet associations tab and click on Edit subnet associations.

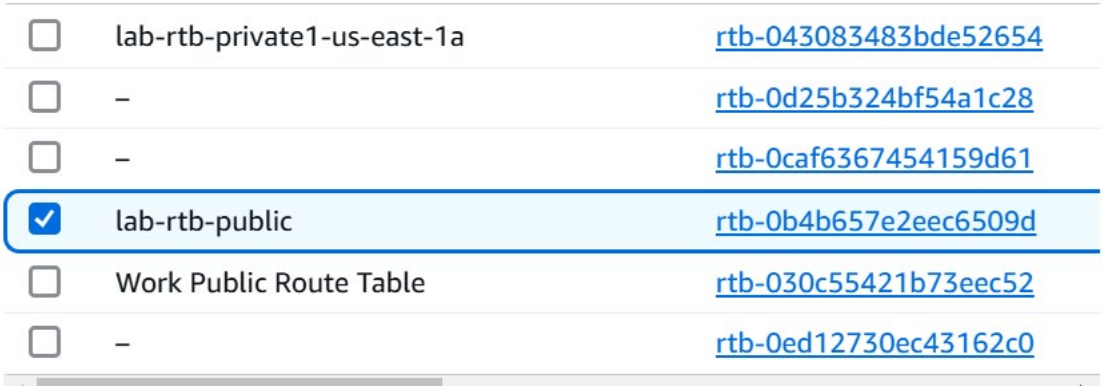


Select both lab-subnet-private1-us-east-1a and lab-subnet-private2, before selecting Save associations.

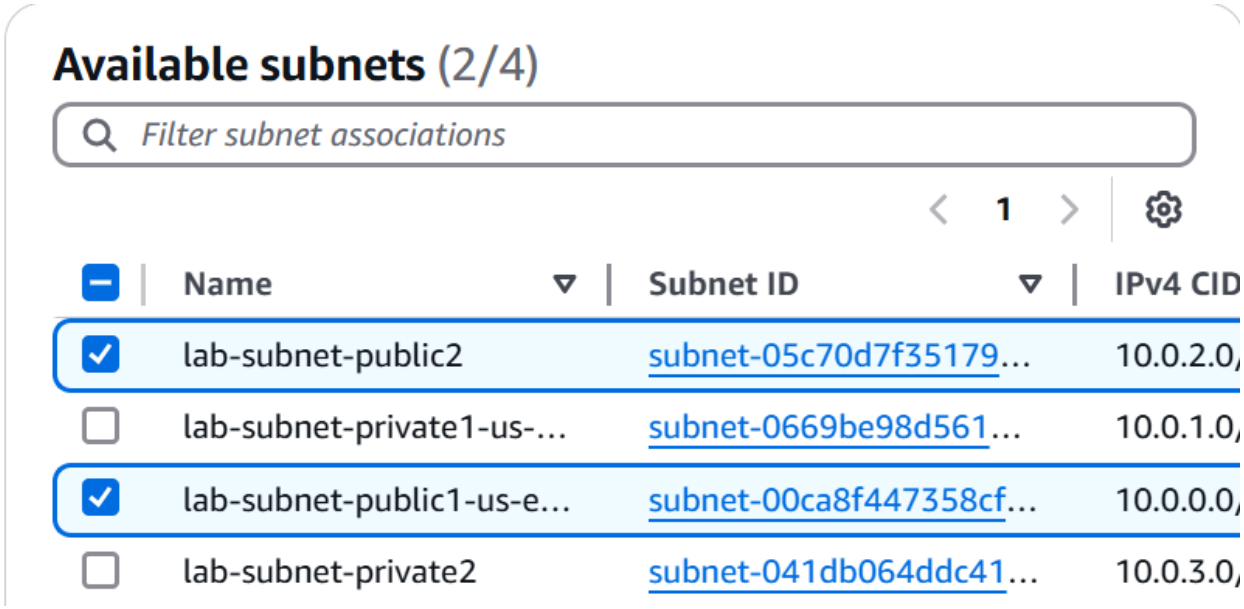


There should be a pop-up as such:

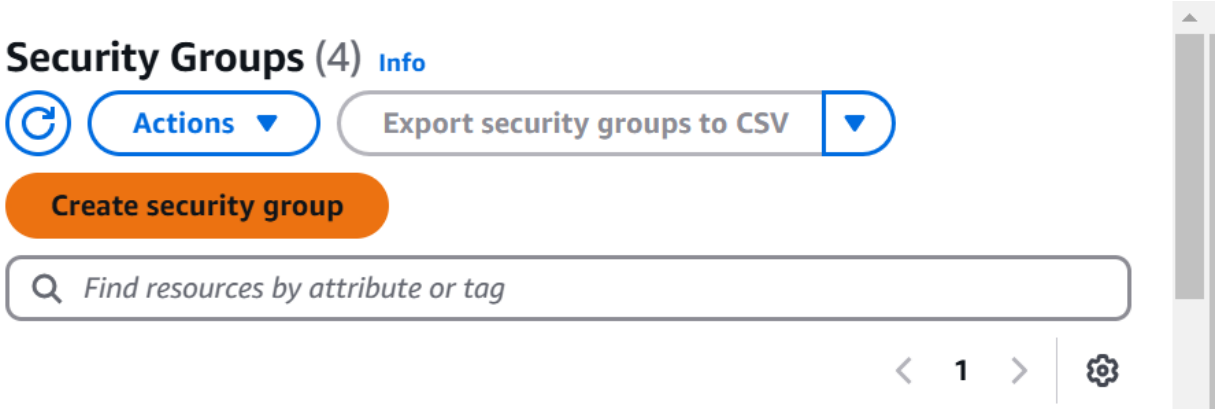
Next, deselect lab-rtb-private1-us-east-1a and select lab-rtb-public.



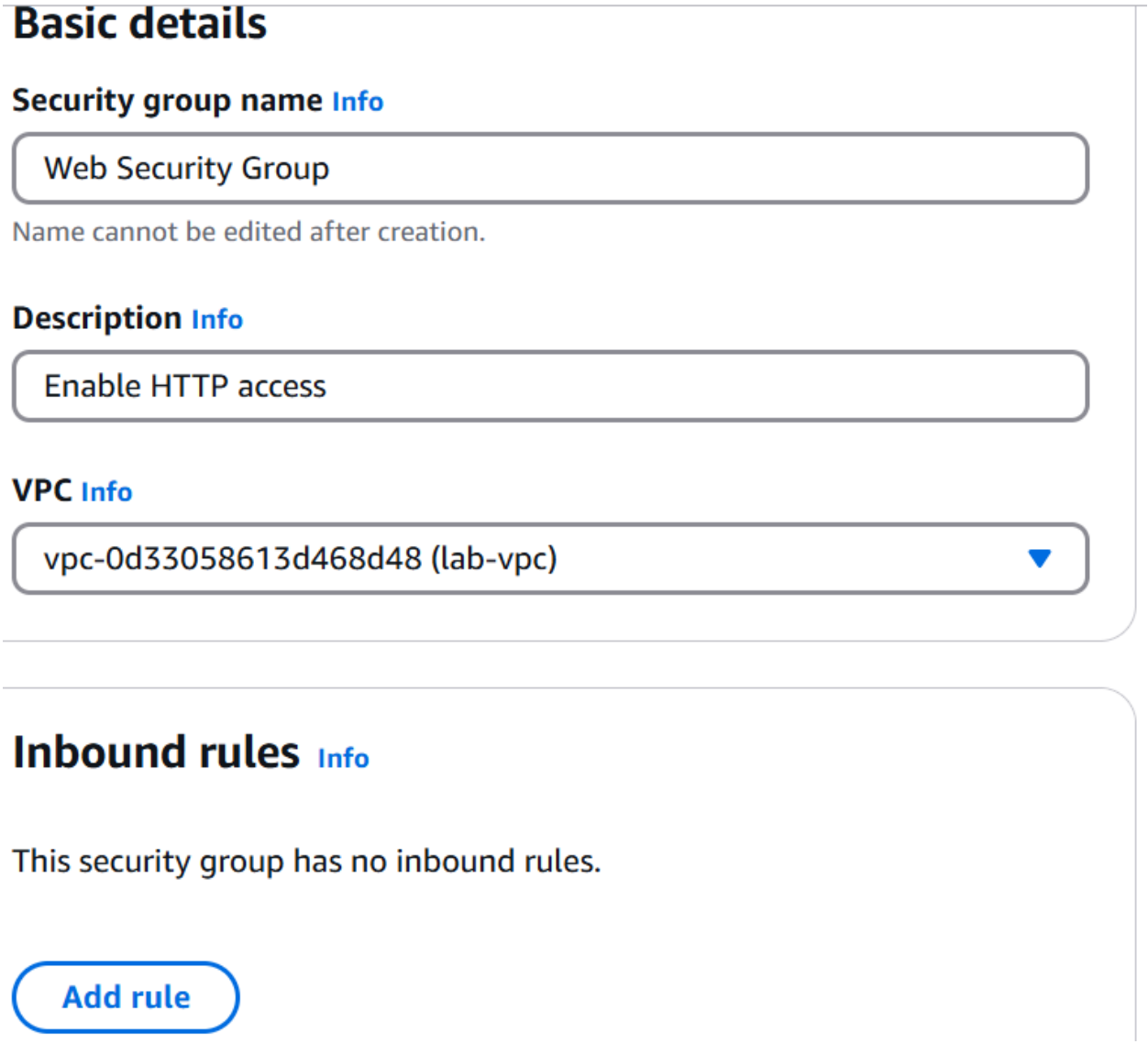
In the Subnet associations tab, choose Edit subnet associations and select both lab-subnet-public1-us-east-1a and lab-subnet-public2.



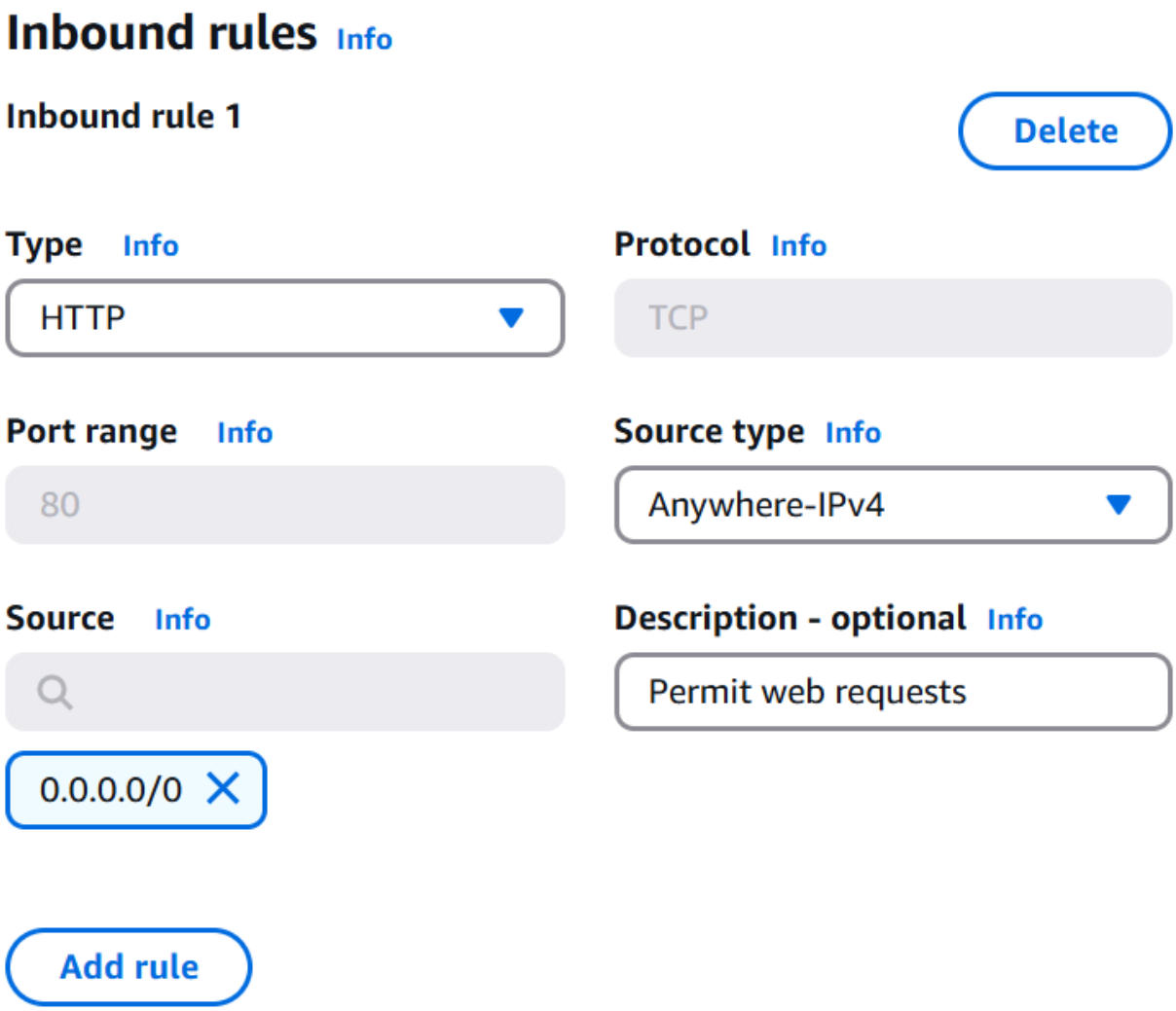
Click on Save associations. In the left navigation pane, select Security groups. Click on Create security group (as shown below).



Configure the following, as shown below:

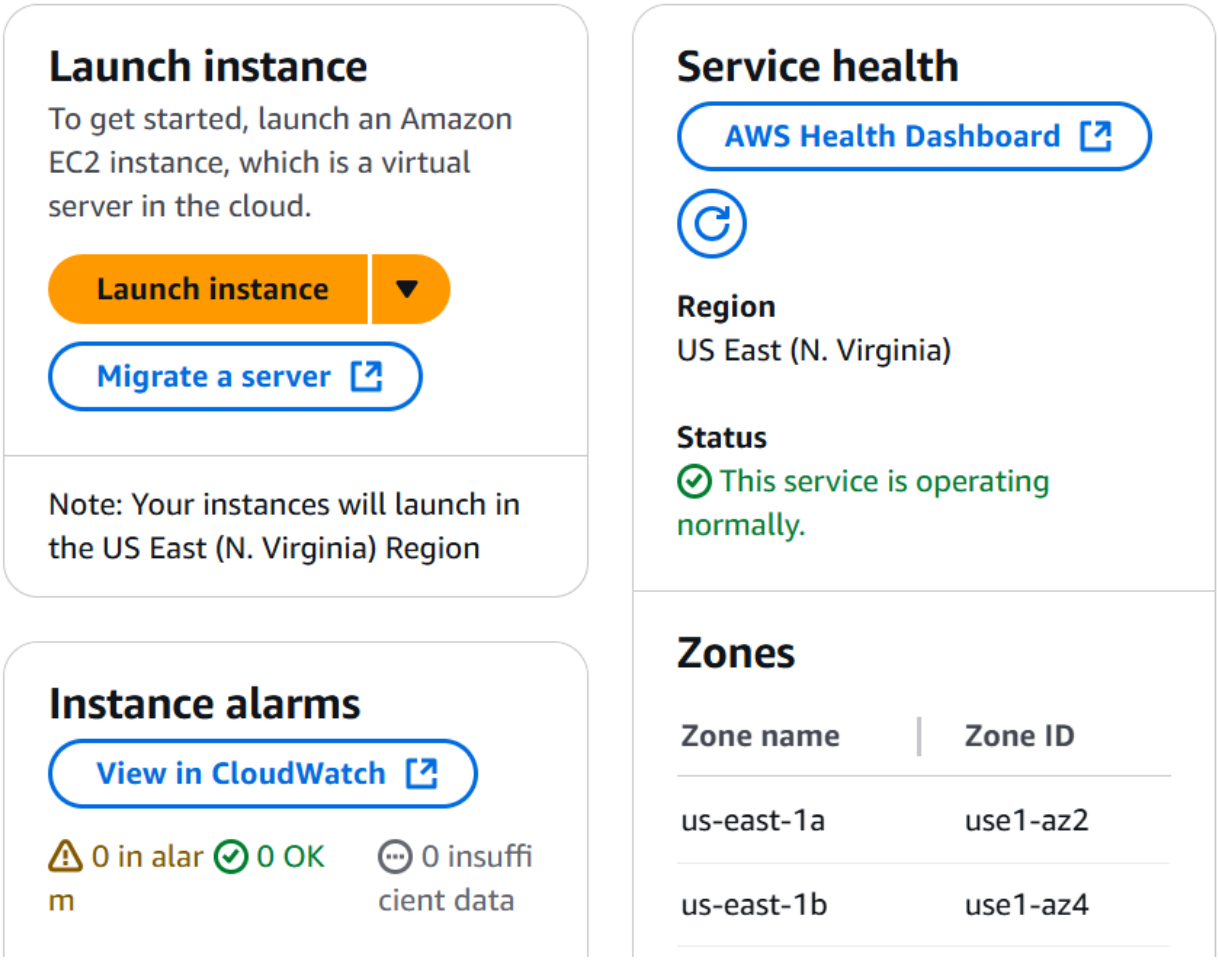


Click on Add rule under Inbound rules and configure the following:

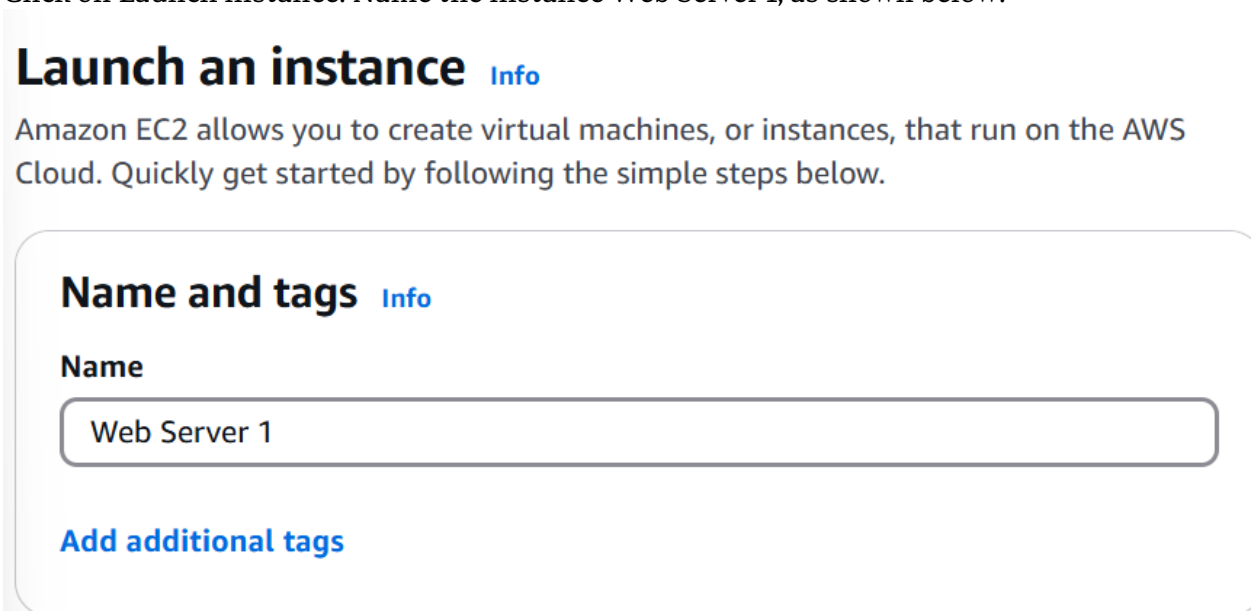


Scroll to the bottom and click on Create security group. There should be a pop-up signifying the successful creation of Web Security Group.

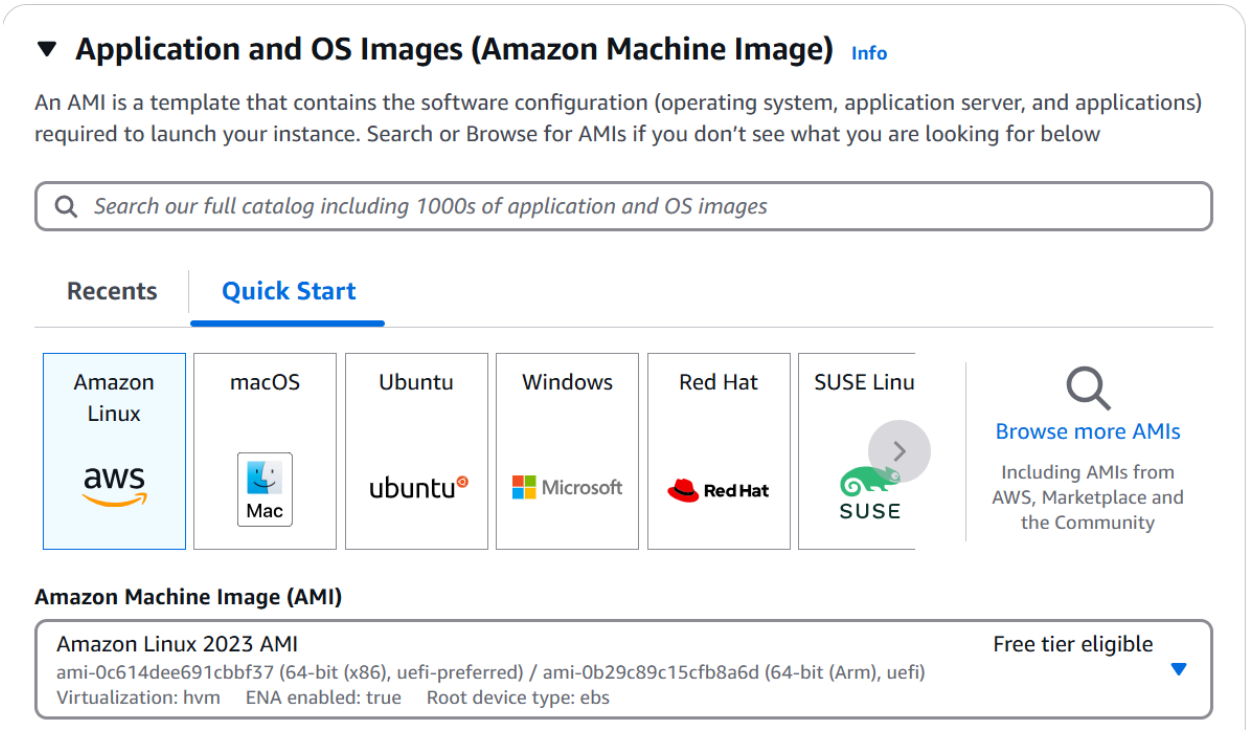
Search for EC2 to open the EC2 console.

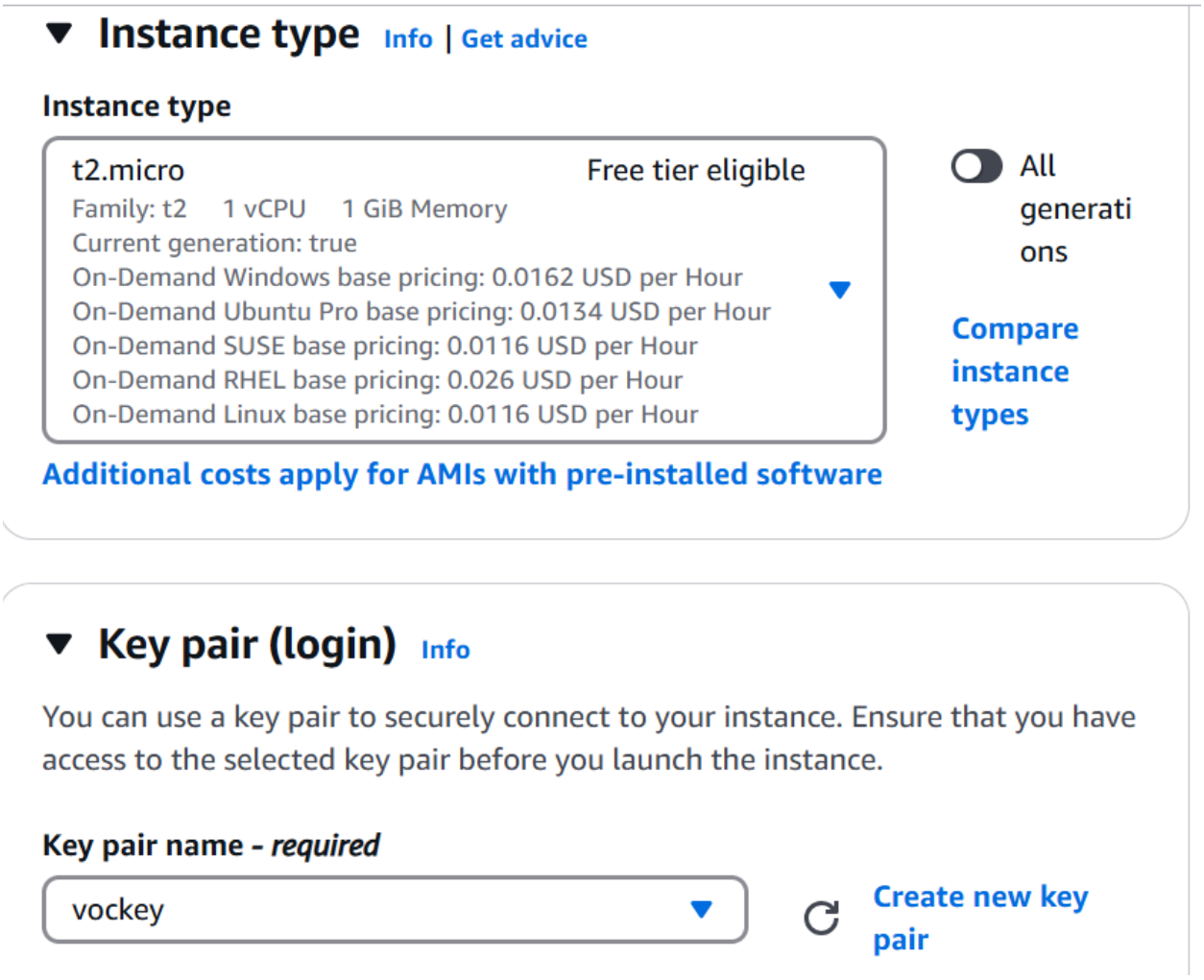


Click on Launch instance. Name the instance Web Server 1, as shown below:



Do not change any of the settings under Application and OS Images. Keep the Instance type as the default t2.micro; select **vockey** from the Key pair name menu.





Click on Edit next to network settings, configuring the VPC, subnet, and auto-assign public IP settings as shown below, as well as the security group. Do not change any settings under Configure storage.

Network settings Info

VPC - *required* Info

vpc-0d33058613d468d48 (lab-vpc)
10.0.0.0/16

Subnet Info

subnet-01f99b36a07454455
lab-subnet-public2
VPC: vpc-0d33058613d468d48
Owner: 877002603542
Availability Zone: us-east-1b
Zone type: Availability Zone
IP addresses available: 251 CIDR: 10.0.2.0/24)

Create new subnet

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

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

Common security groups Info

Select security groups

Web Security Group sg-0418161432183a8a8
VPC: vpc-0d33058613d468d48

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

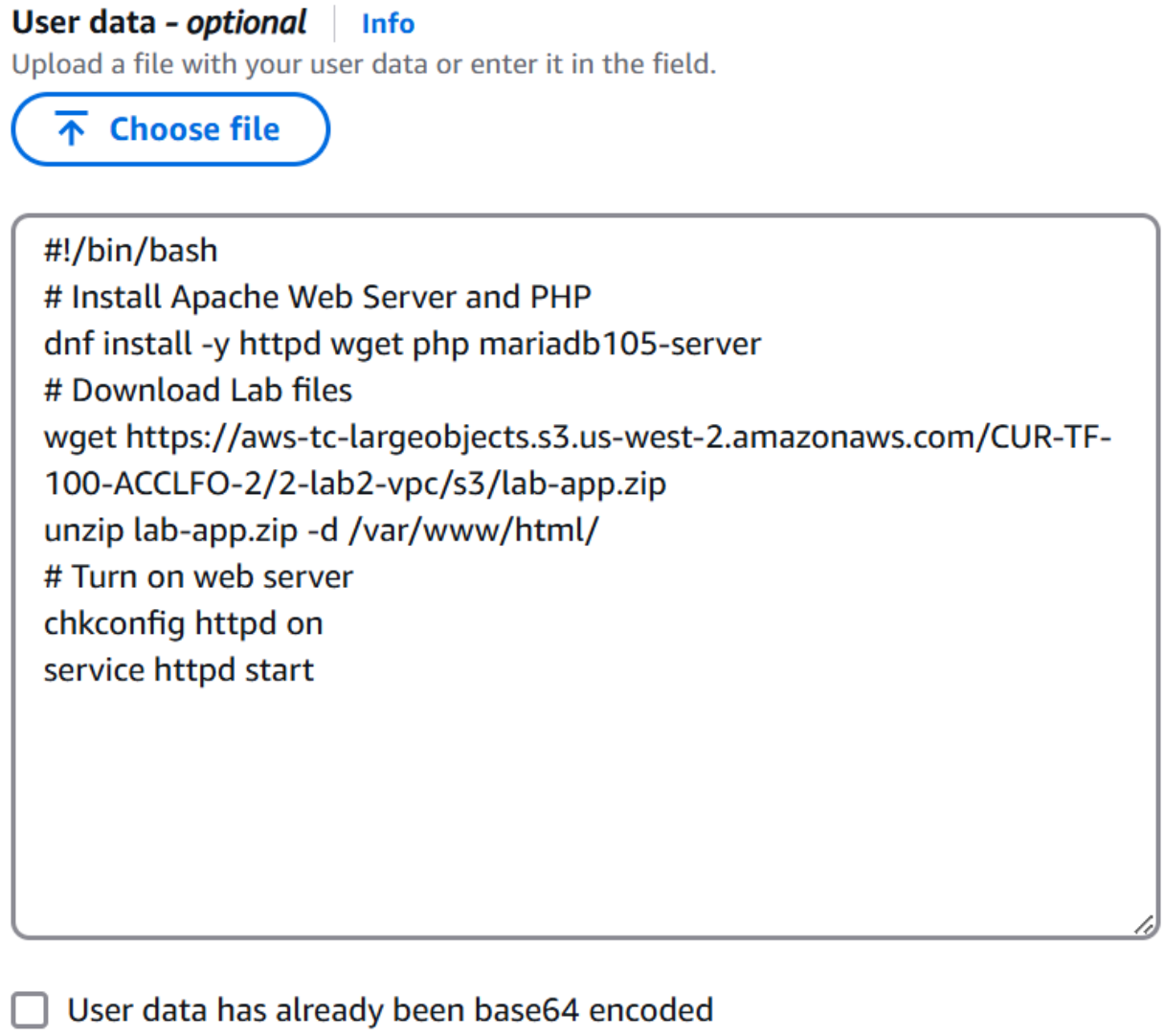
Advanced network configuration

Configure storage Info

Advanced

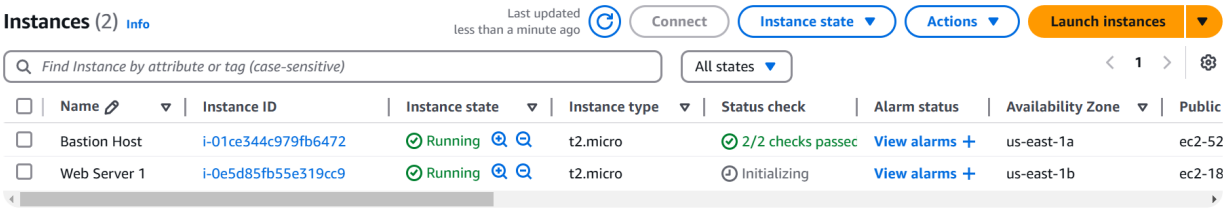
1x 8 GiB gp3 Root volume 3000 IOPS (Not encrypted)

Expand the Advanced details section, scrolling down and copying the following code into the User data box.

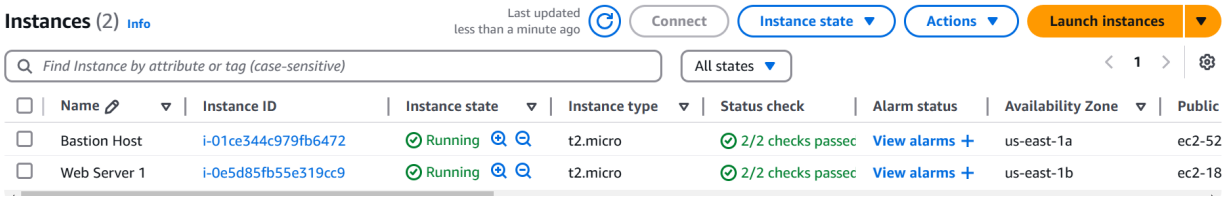


There should be a pop-up signaling the successful launch of an instance.

Scroll to the very bottom to click on View Instances, which will bring you to this page:

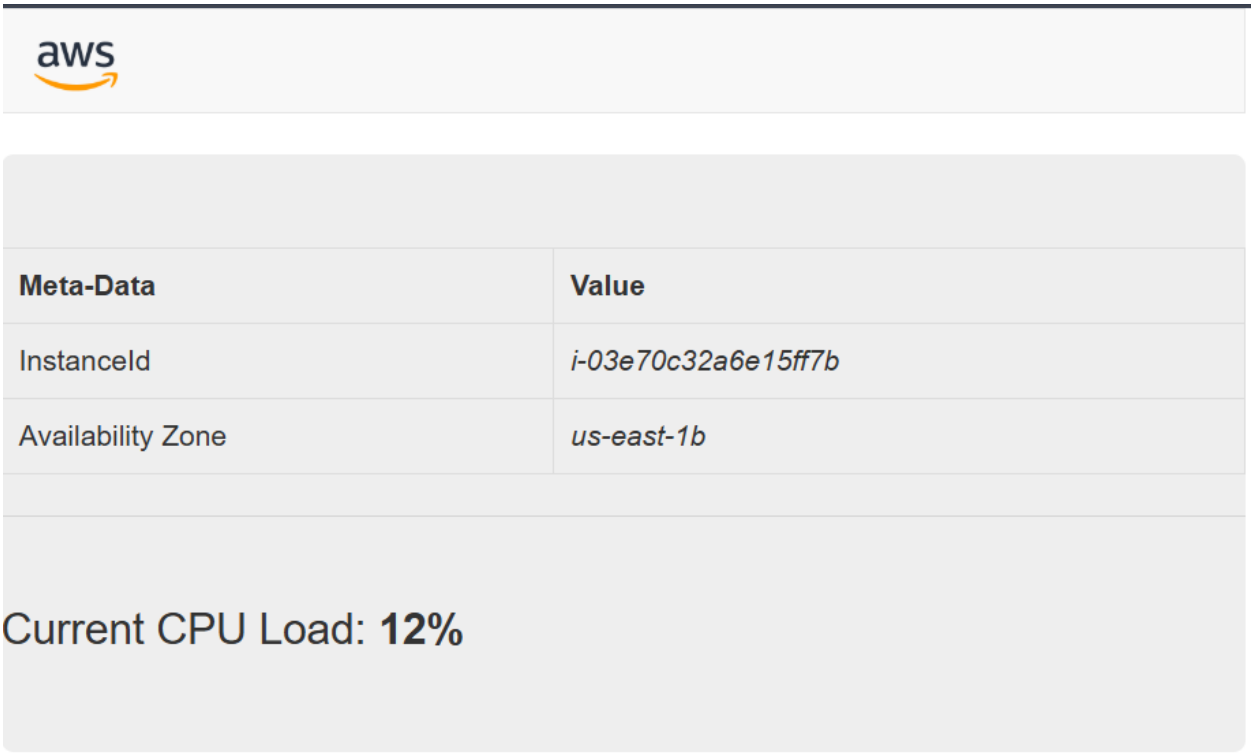


Wait until the Status for Web Server 1 reads 2/2 checks passed; you may need to select the Refresh icon a few times until then.



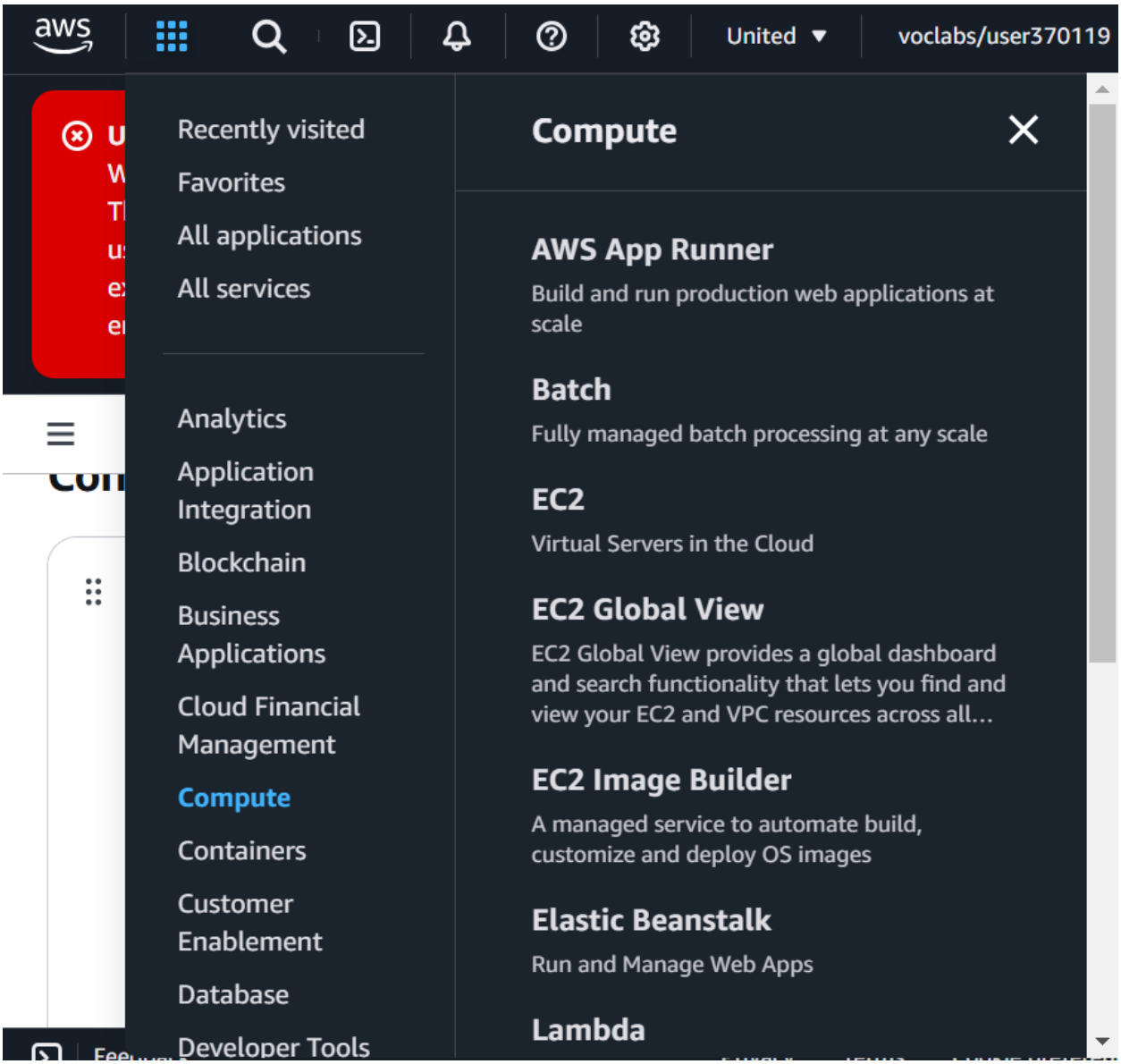
Select Web Server 1.

Copy and paste the Public IPv4 DNS address into a new browser tab. A web page similar to the following should load.

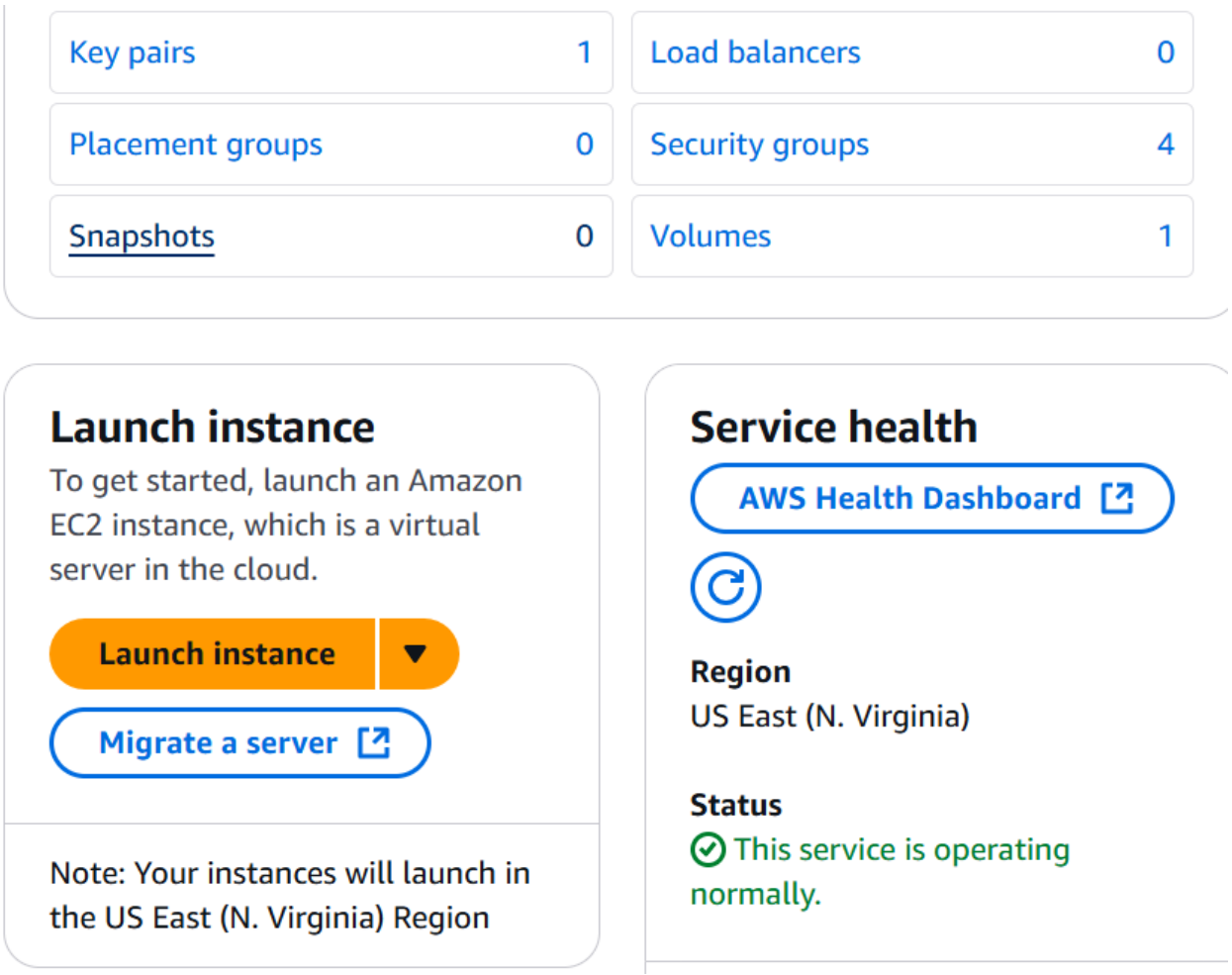


LAB 3

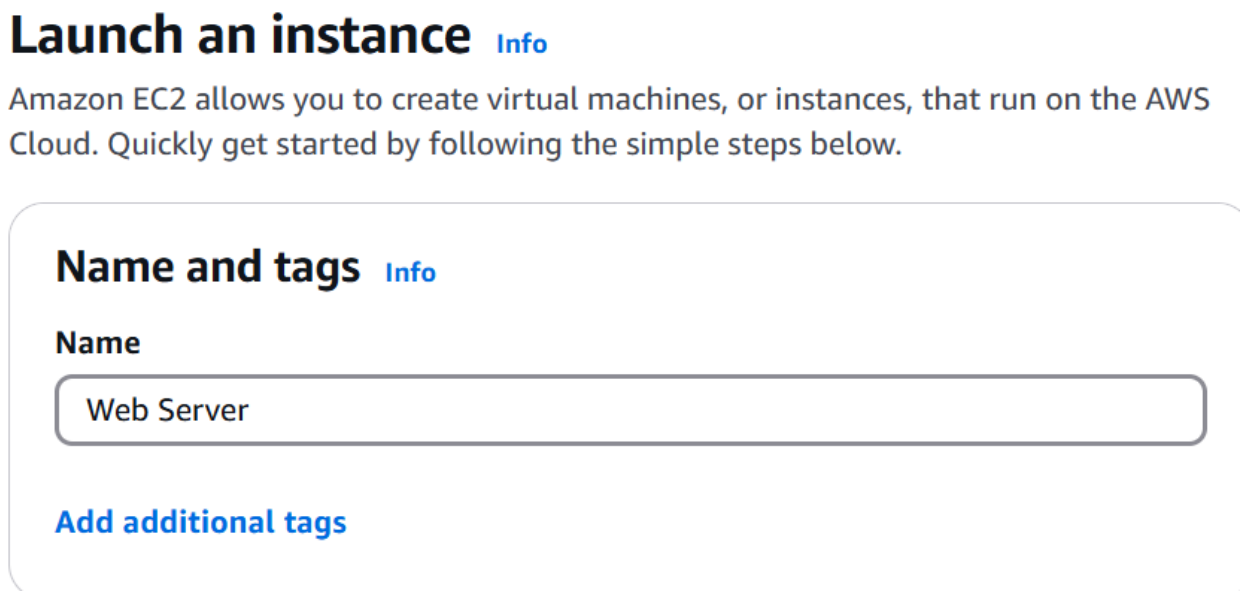
In the AWS Management Console, choose Services before clicking on Compute.



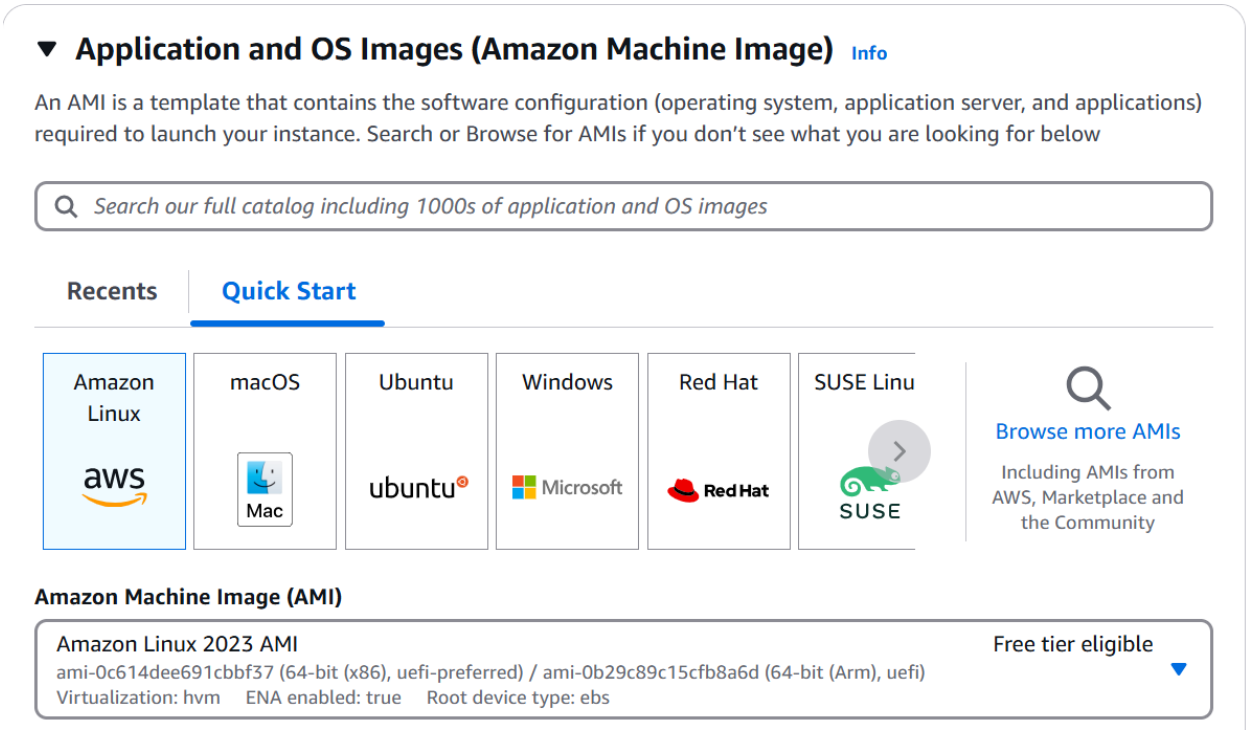
Click on EC2 under Compute.

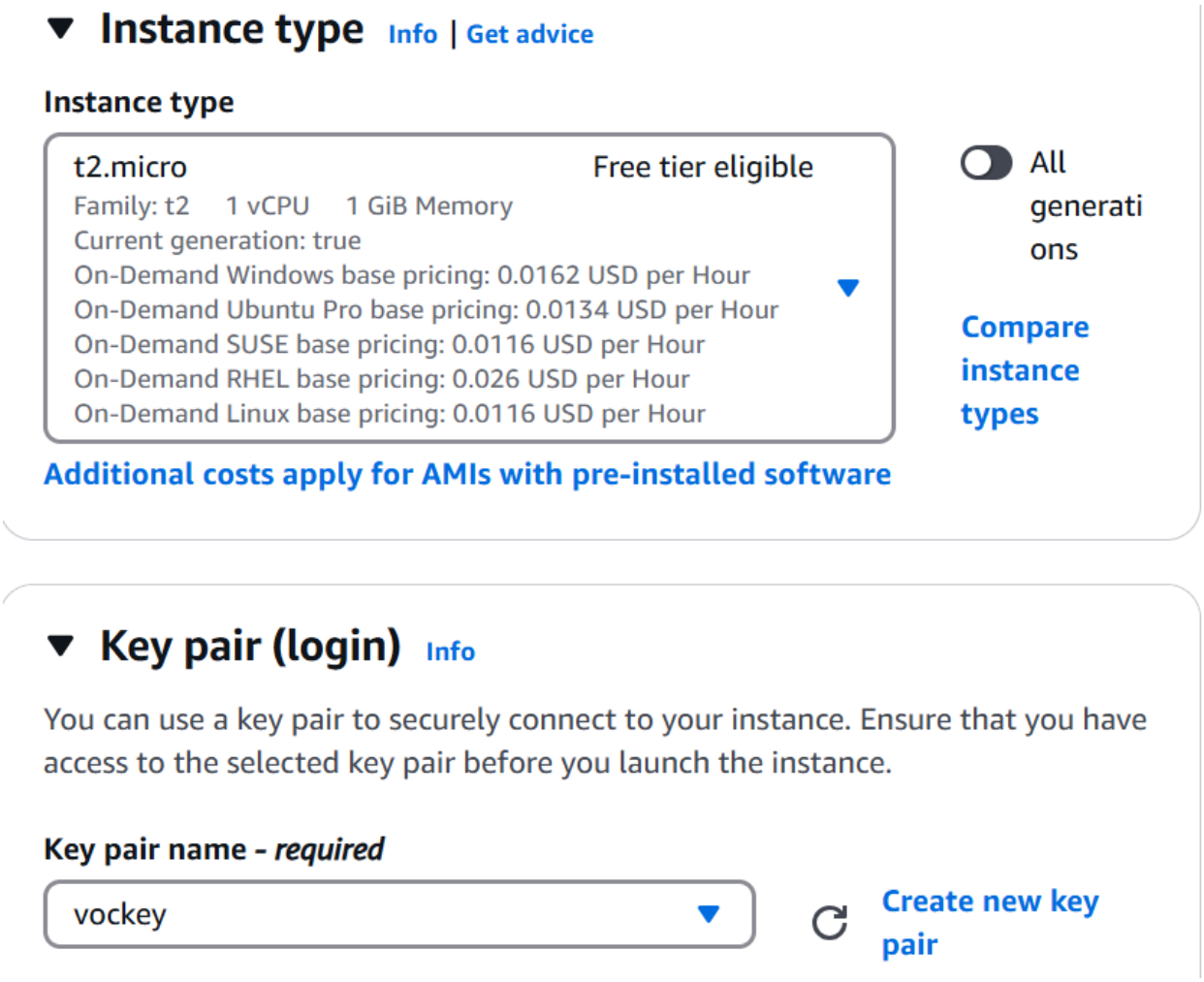


Select the Launch instance button from the Launch instance menu. Configure the name as show below:

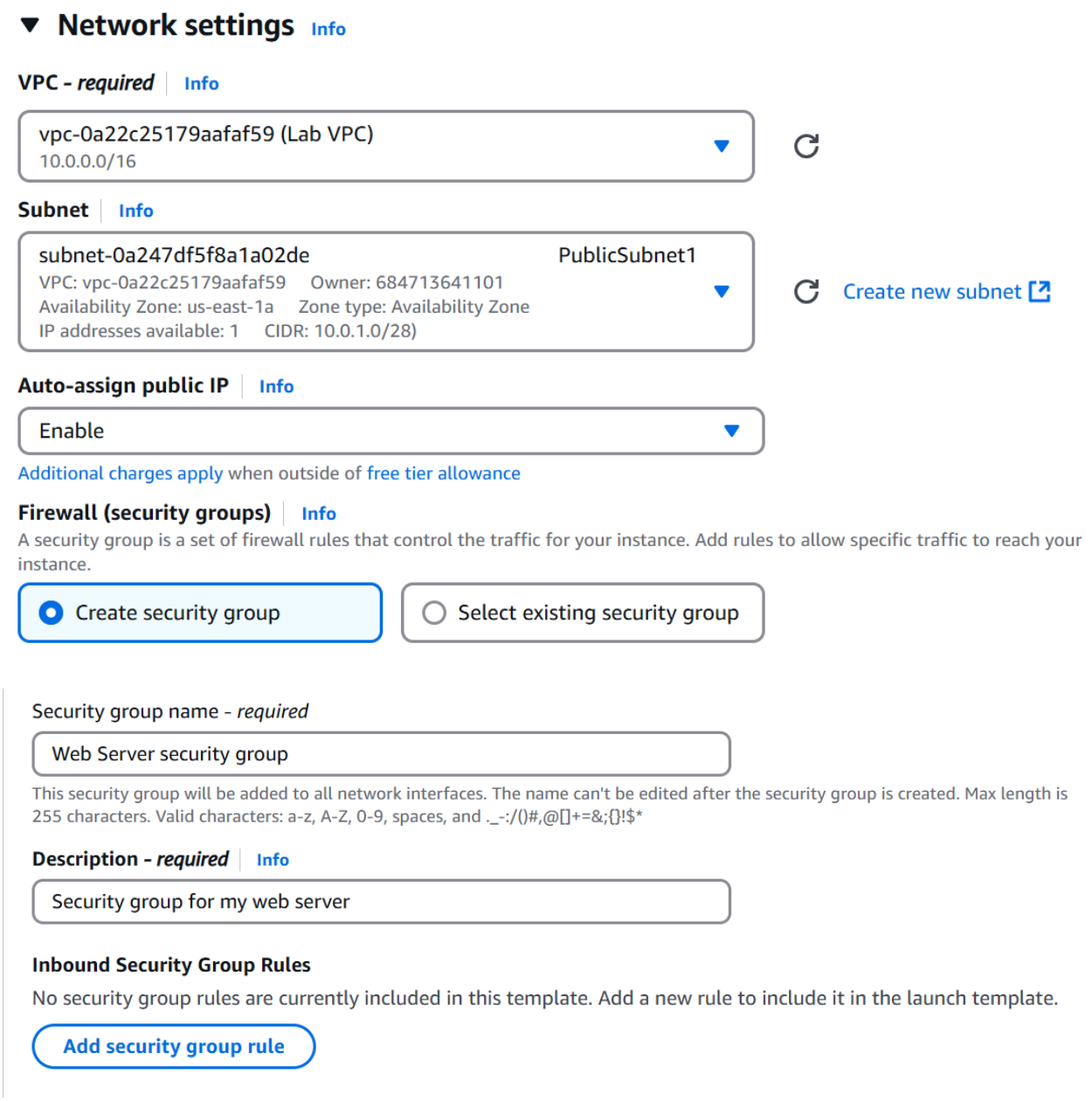


Do not change any of the settings under Application and OS Images. Keep the Instance type as the default t2.micro; select **vockey** from the Key pair name menu.


Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Mac
Ubuntu
Windows
Microsoft
Red Hat
SUSE Linu
SUSE
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 AMI
Free tier eligible
ami-0c614dee691cbbf37 (64-bit (x86), uefi-preferred) / ami-0b29c89c15cfb8a6d (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs



Click on Edit next to network settings, configuring the VPC, subnet, and auto-assign public IP settings as shown below, as well as the security group. Make sure to remove the Inbound Security Group rule. Do not change any settings under Configure storage.

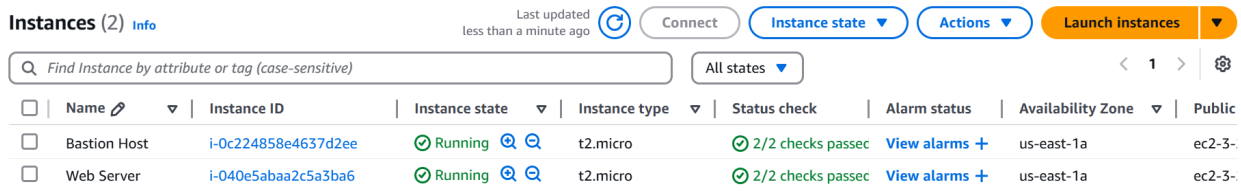


Then, expand Advanced details and enable Termination protection. Copy the following code into the User Data box:

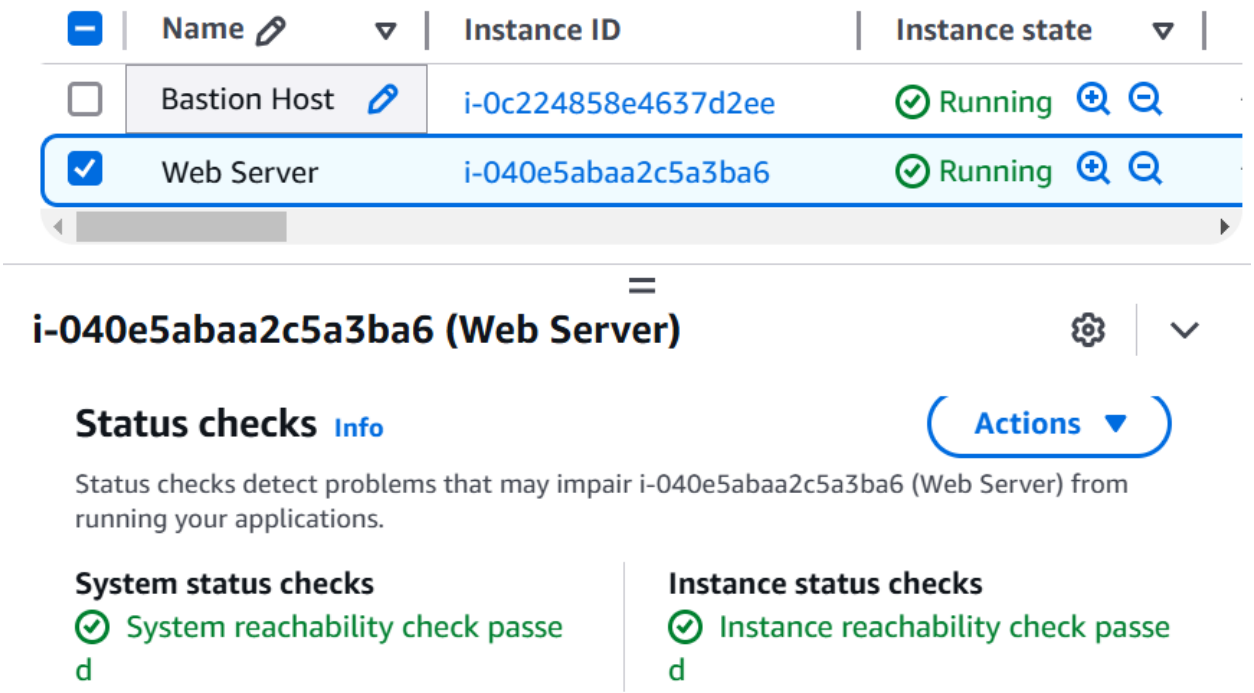
```
#!/bin/bash

dnf install -y httpd

systemctl enable httpd

systemctl start httpd
```

echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html

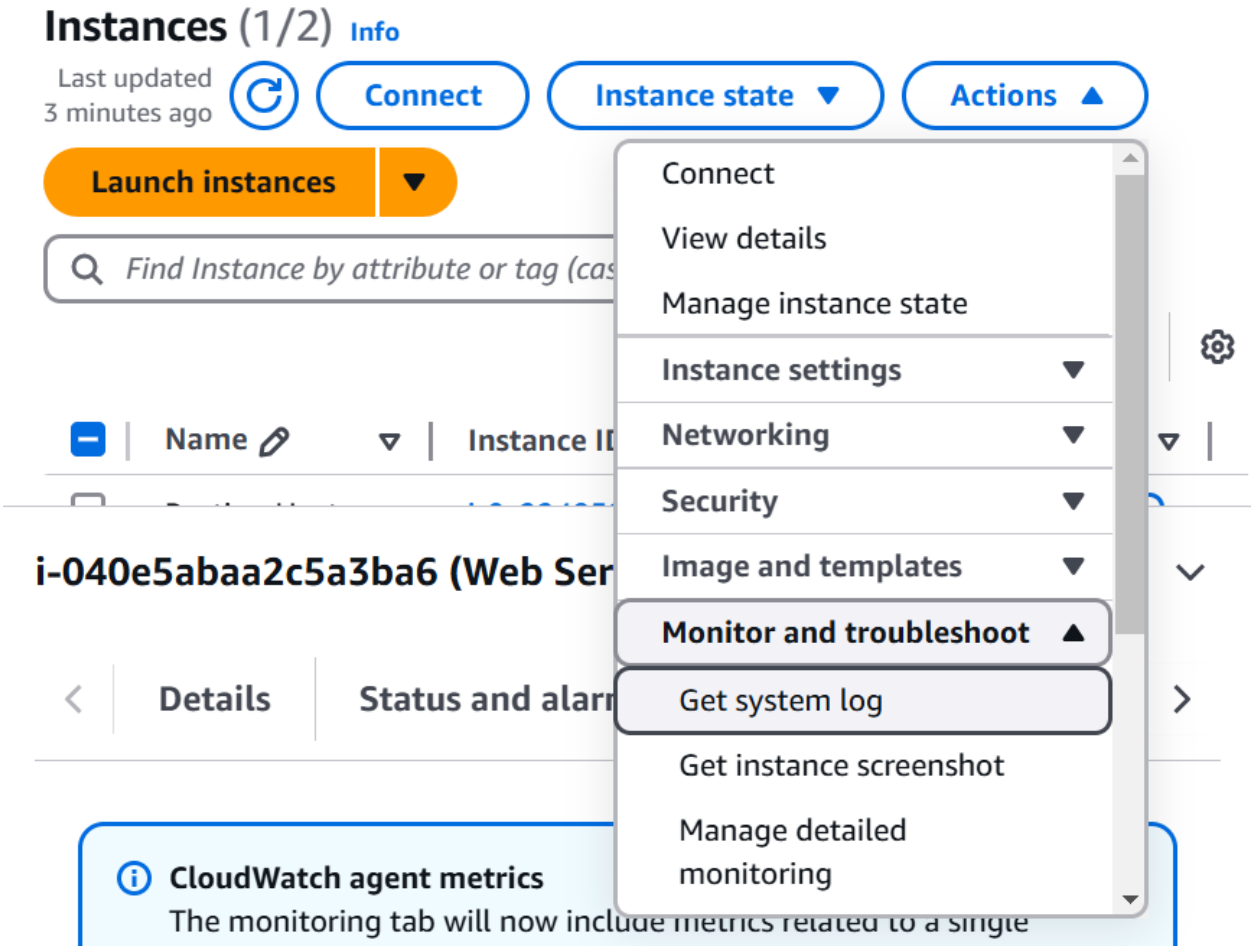
Select Launch instance and choose View all instances after a Success message pops up. Wait until the Instance State is Running and the two Status Checks are both passed, clicking on the Refresh icon:



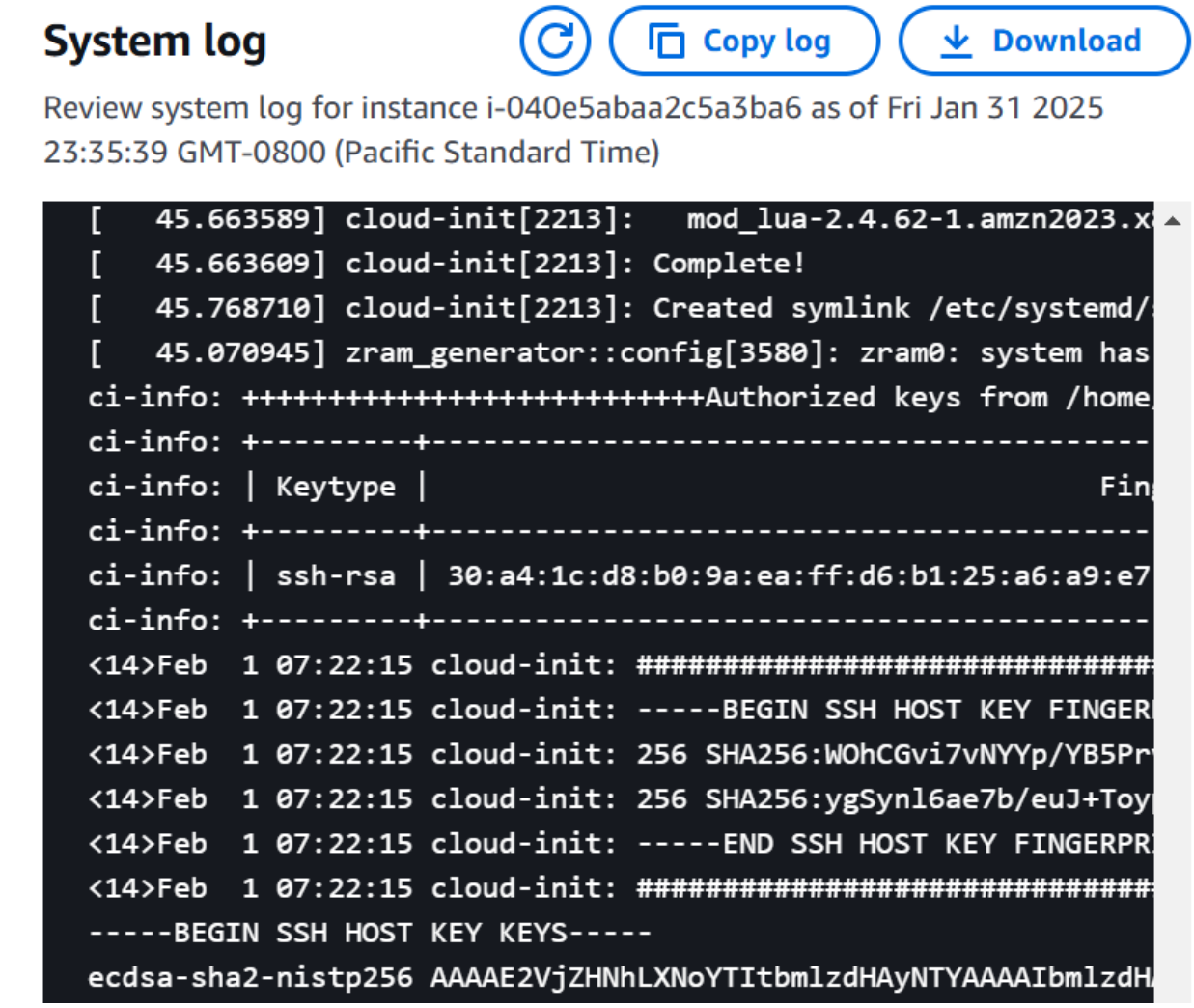
Select the Web Server using the checkbox, navigating to the Status checks tab. Ensure that both the system reachability and Instance reachability checks have passed.



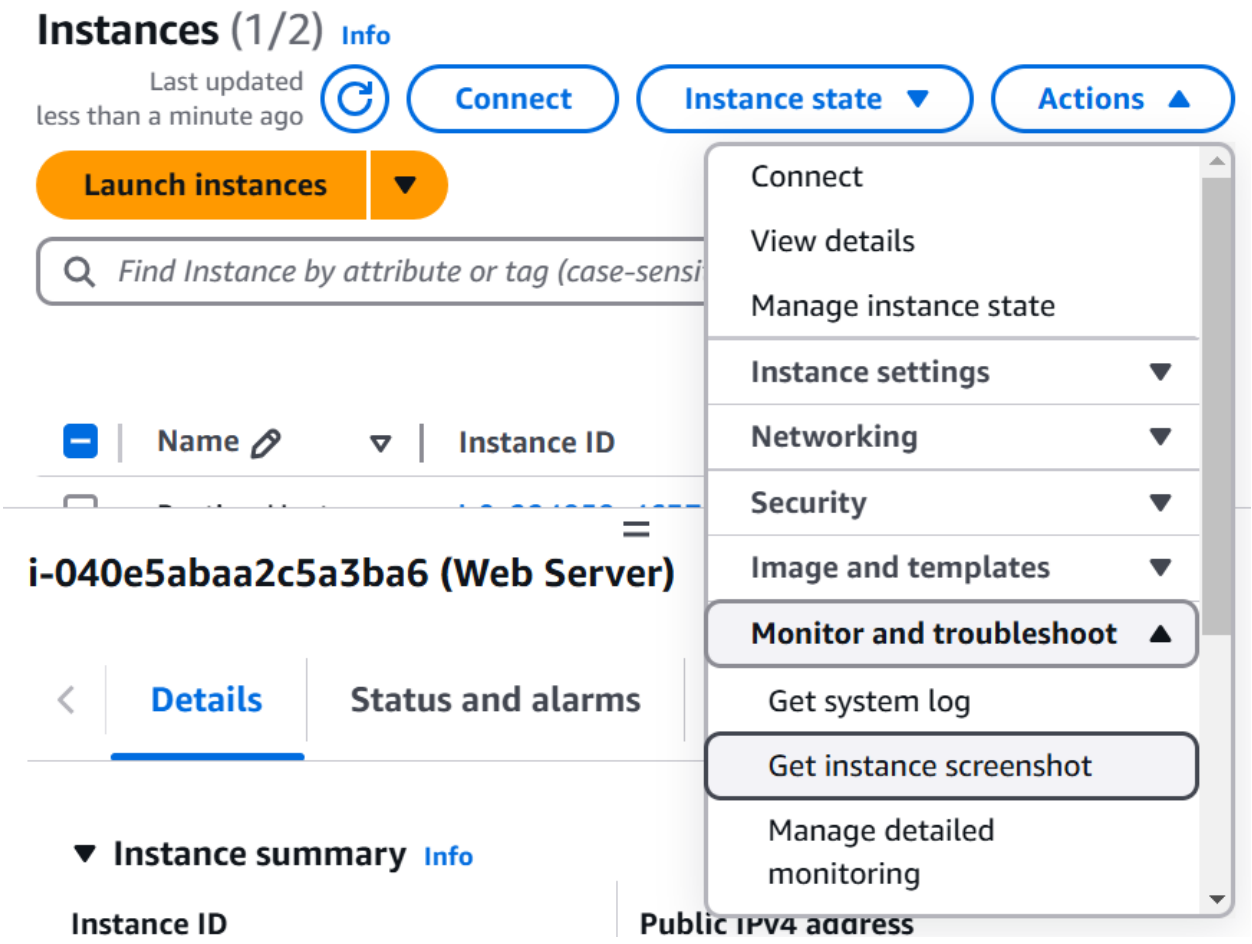
You may also use the Monitoring tab to view the Amazon CloudWatch metrics. Next, select the Actions drop-down menu, click on Monitor and troubleshoot, and select Get system log.


Instances (1/2) Info
Last updated 3 minutes ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (cas
Name
Instance ID
i-040e5abaa2c5a3ba6 (Web Ser
Details
Status and alarm
Connect
View details
Manage instance state
Instance settings
Networking
Security
Image and templates
Monitor and troubleshoot
Get system log
Get instance screenshot
Manage detailed monitoring
CloudWatch agent metrics
The monitoring tab will now include metrics related to a single

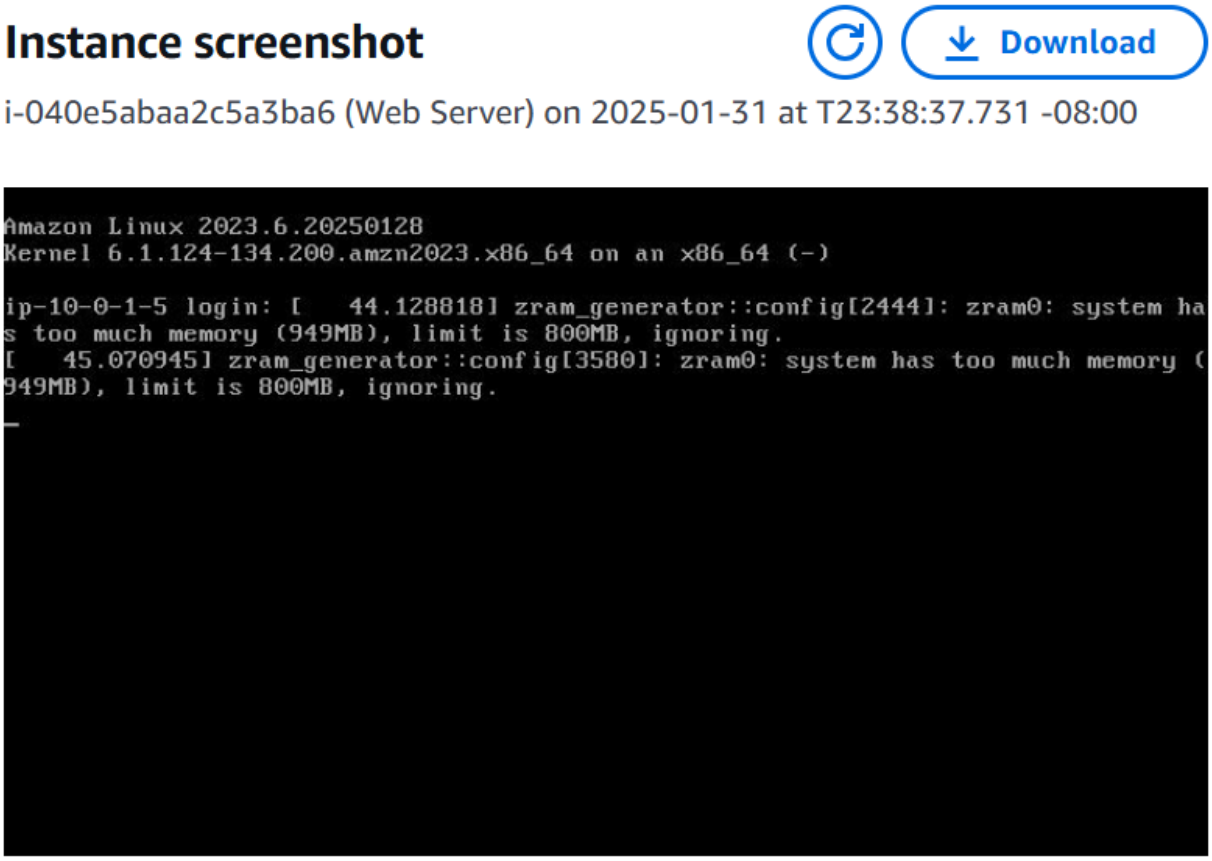
Note that the HTTP package was installed, as the system log shows.



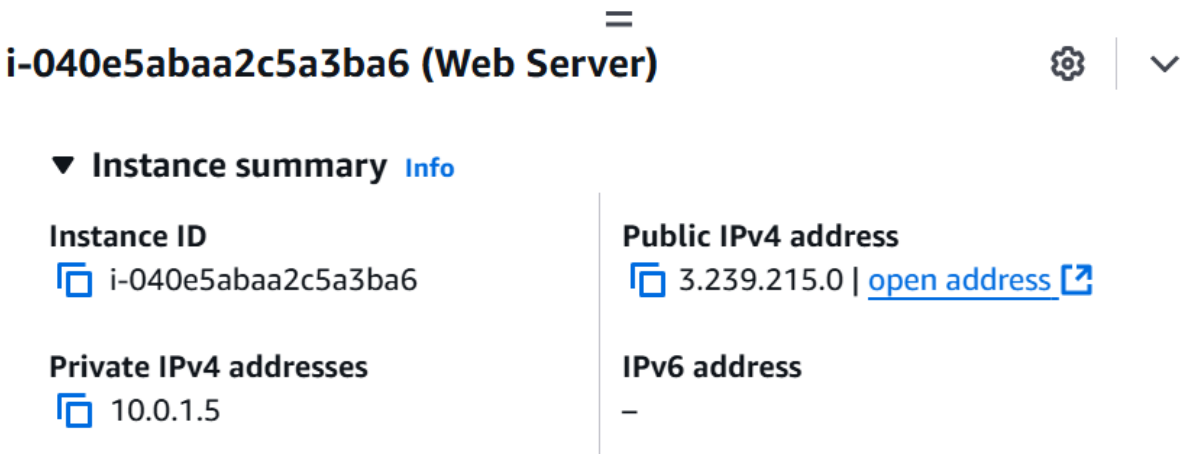
Choose Cancel at the very bottom to return to the previous page. With Web Server still selected, return to the Actions drop-down menu to select Monitor and troubleshoot. Click on Get instance screenshot.


Instances (1/2) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensi
Name
Instance ID
i-040e5abaa2c5a3ba6 (Web Server)
Details
Status and alarms
Instance summary Info
Instance ID
Public IPv4 address
Connect
View details
Manage instance state
Instance settings
Networking
Security
Image and templates
Monitor and troubleshoot
Get system log
Get instance screenshot
Manage detailed monitoring

This is the final way that will be explored in this lab on how to monitor an instance.

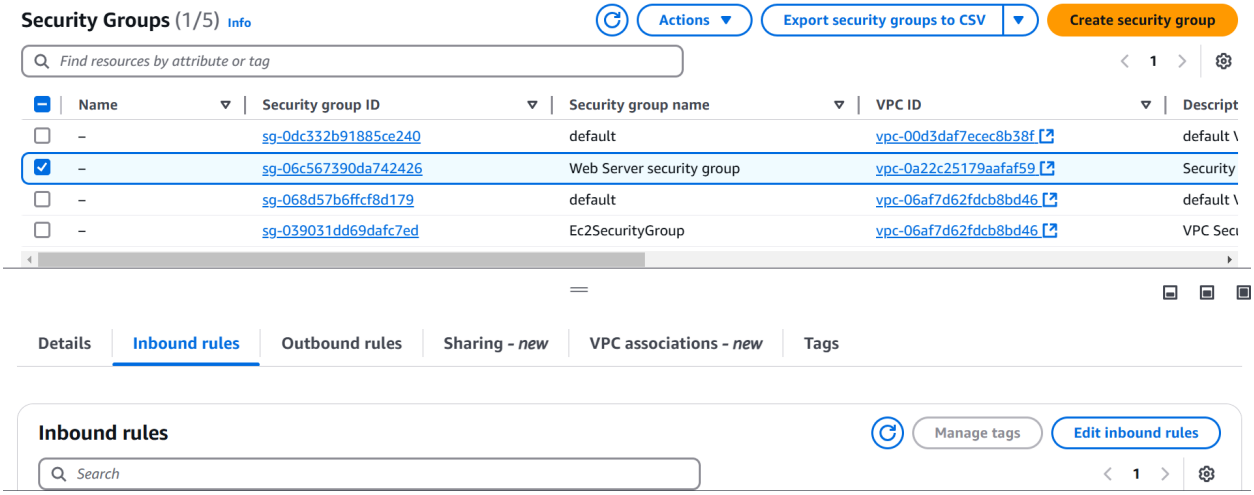


Choose cancel; while Web Server is still selected, navigate to the Details tab. Copy the Public IPv4 address and enter the address into a new browser tab.

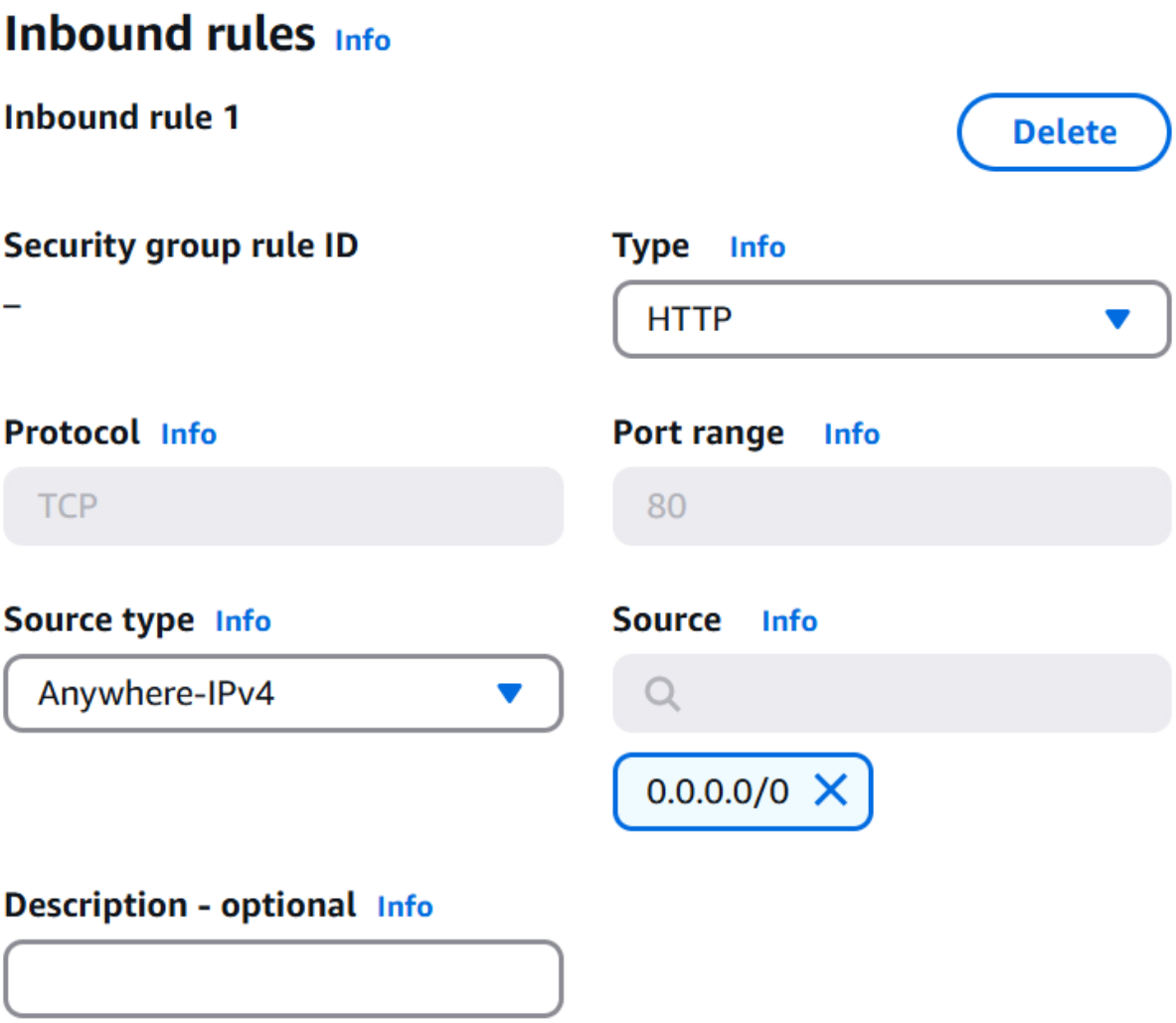


The web browser should not be able to access the web server due to the security group blocking inbound traffic on port 80. Return to the EC2 Console tab and select Security Groups from the

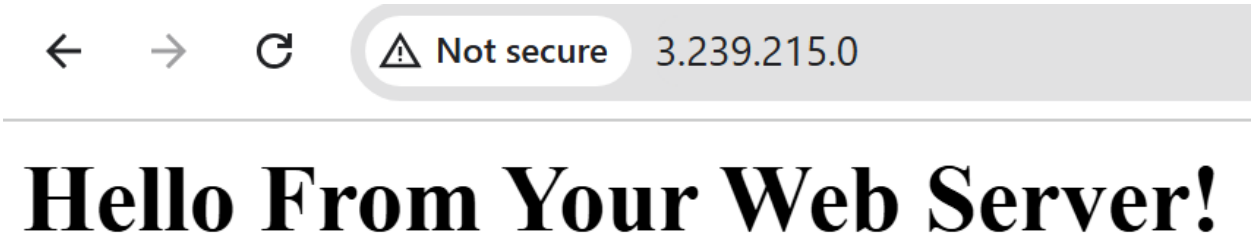
navigation pane. Check the box for Web Server security group and navigate to the Inbound rules tab.



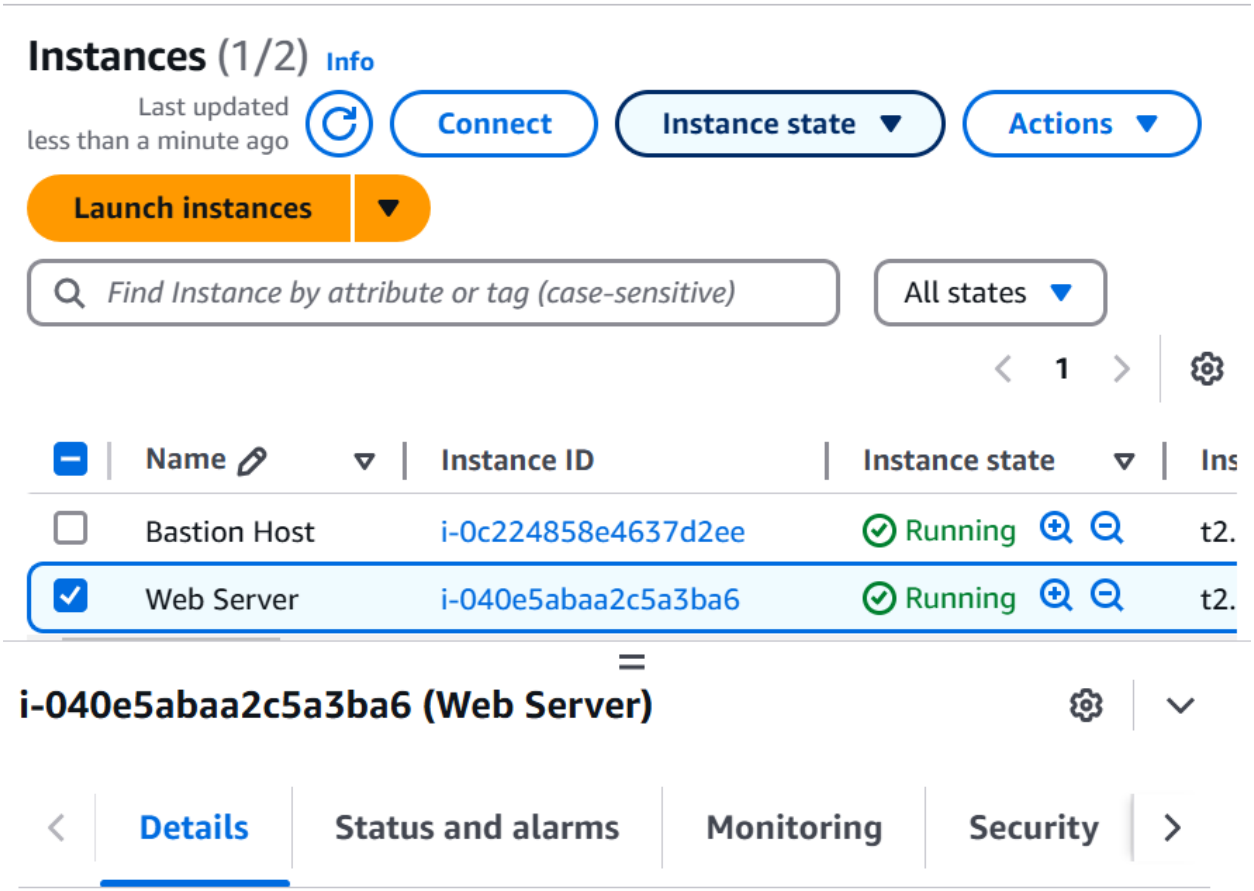
Edit inbound rules and select Add rule, configuring as shown:



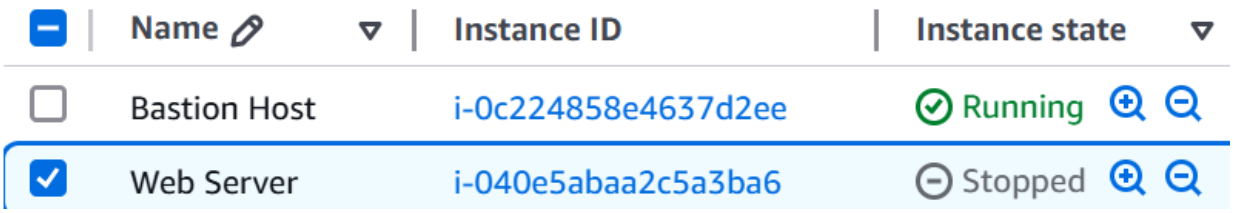
Save rules. When refreshing the web server tab, a message should appear, like the following:



Return to the Instances page and check the box for the Web Server instance.

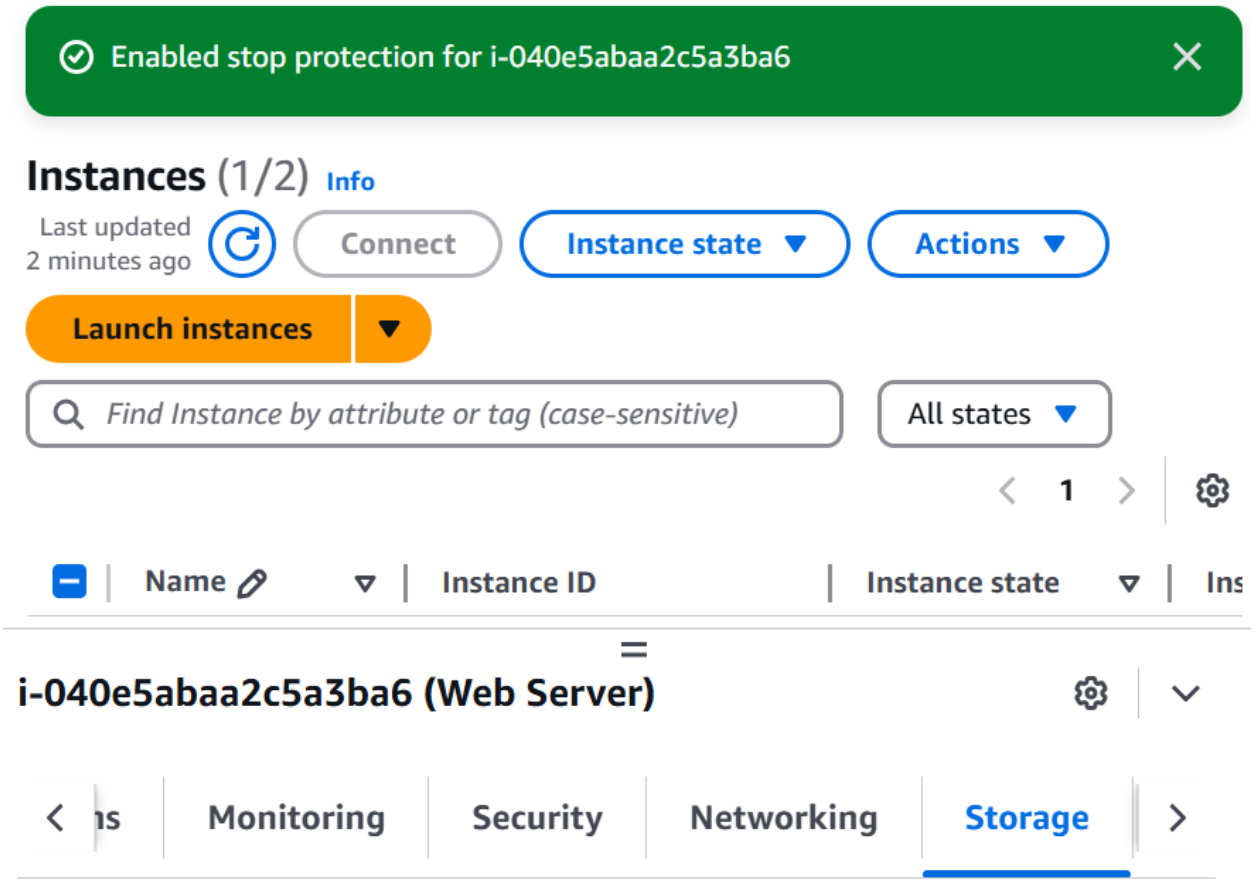


In the Instance state menu, select Stop instance and choose Stop. Wait until the Instance state is Stopped before continuing.

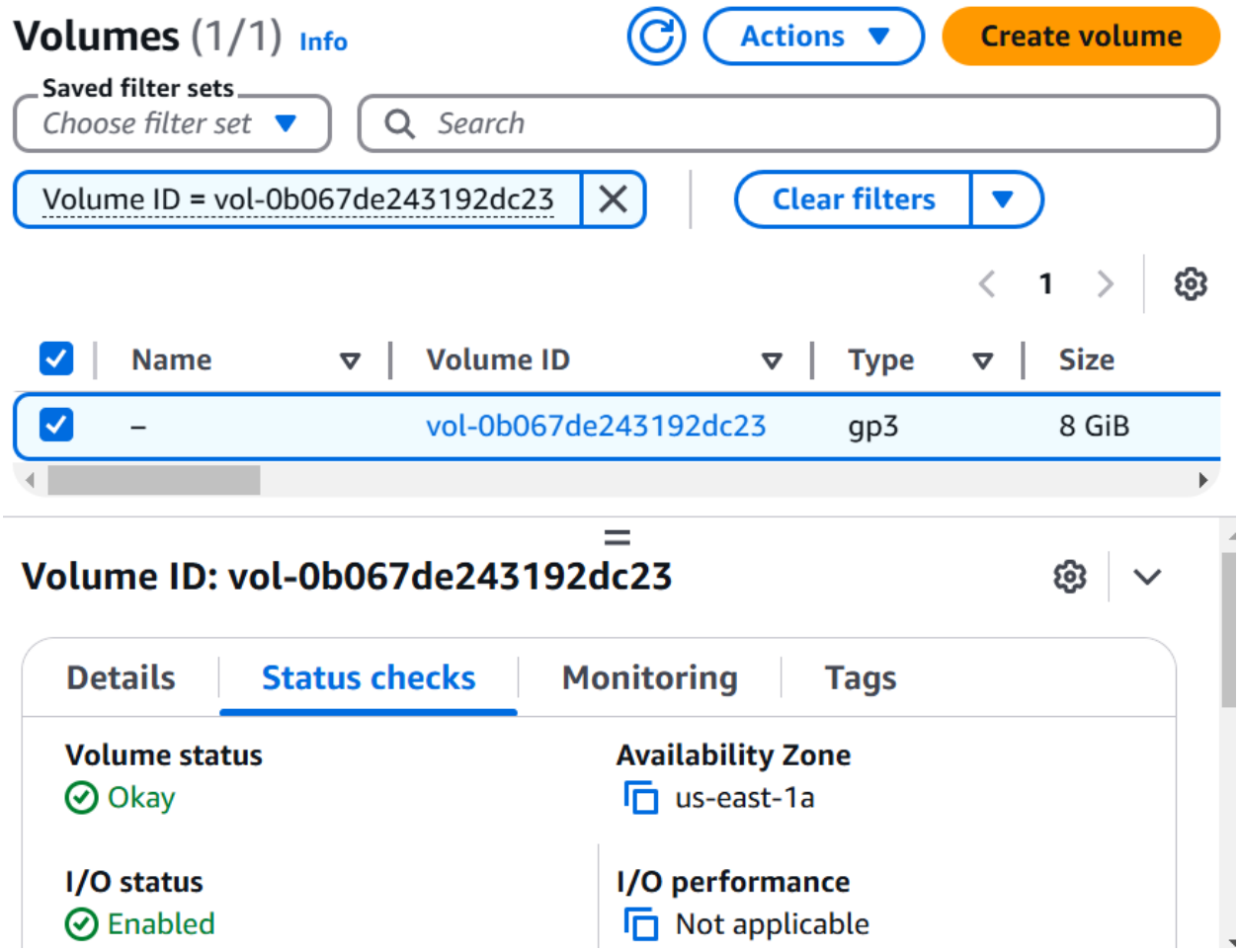


In the Actions menu, with the Web Server instance still selected, select Instance settings and Change instance type to change the Instance Type to **t2.small**. click Apply.There should be a pop-up success message appearing.

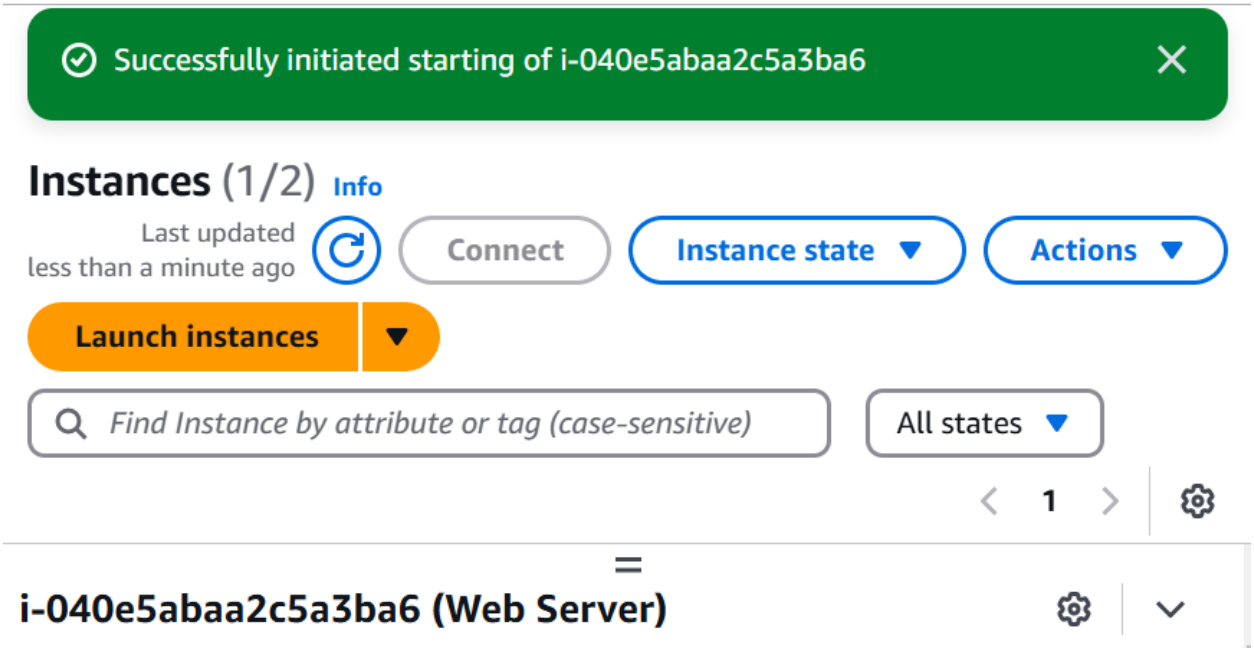
While the Web Server Instance is selected, go into the Actions drop-down menu, selecting Instance settings, then Change stop protection. Select Enable before pressing Save. Make sure the success message for Stop protection shows up and that the Web Server instance is still selected. Choose the Storage tab.



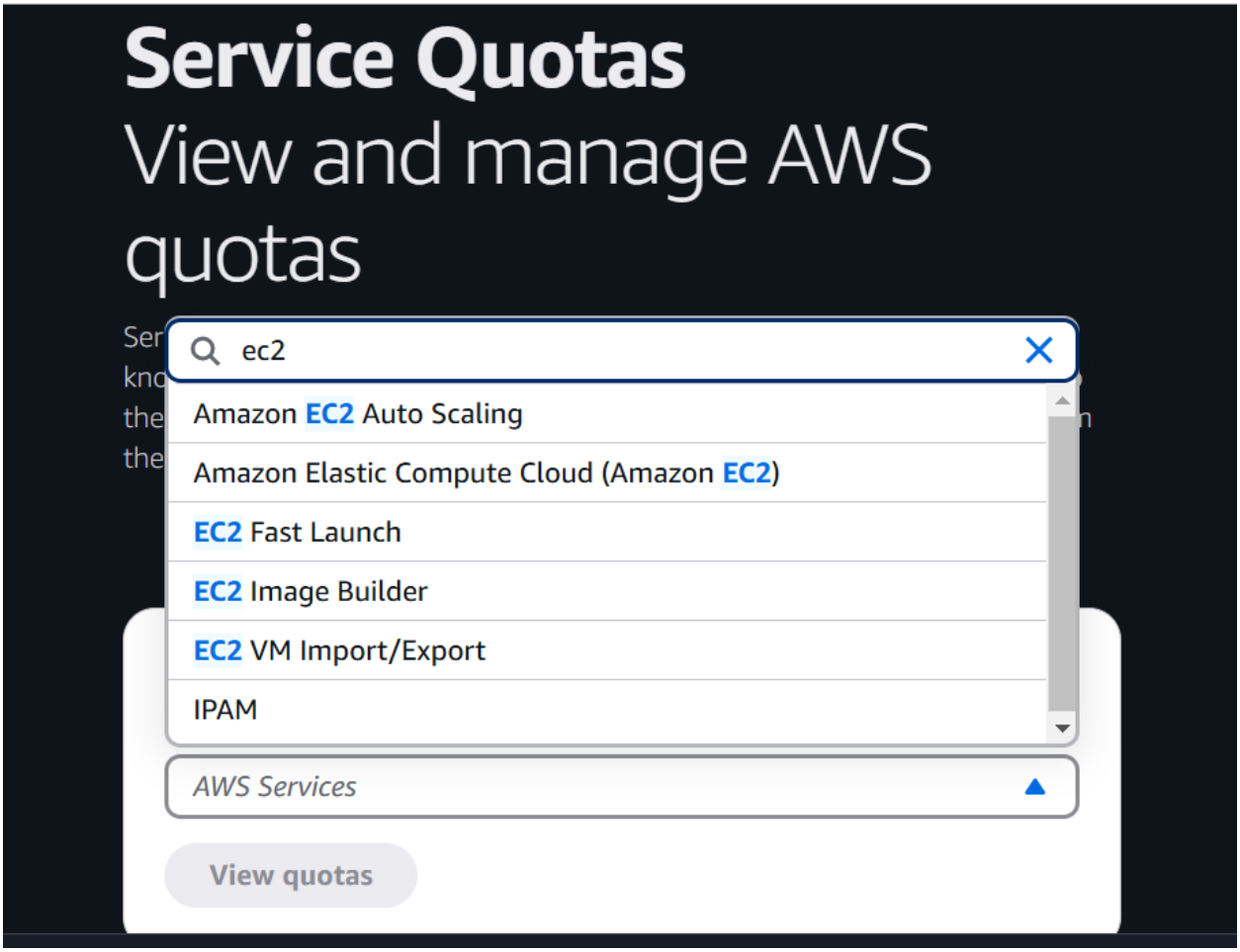
Select the name of the Volume ID and the checkbox next to the volume.



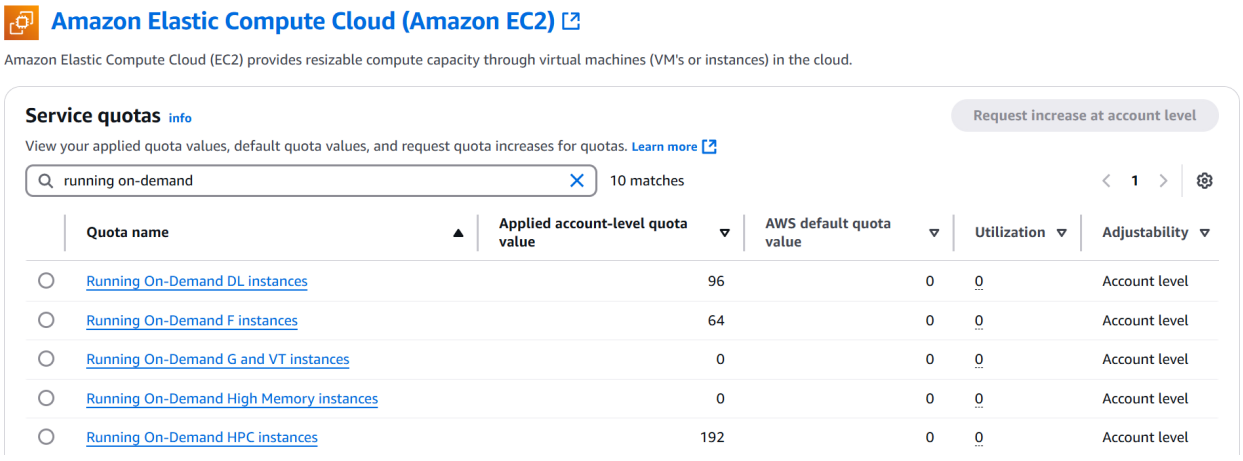
In the Actions menu, select Modify volume and change the size to 10 before clicking Modify. After the size is now 10 GiB, in the left navigation pane, choose Instances. Select the Web Server instance and in the Instance state drop-down menu, select Start instance.



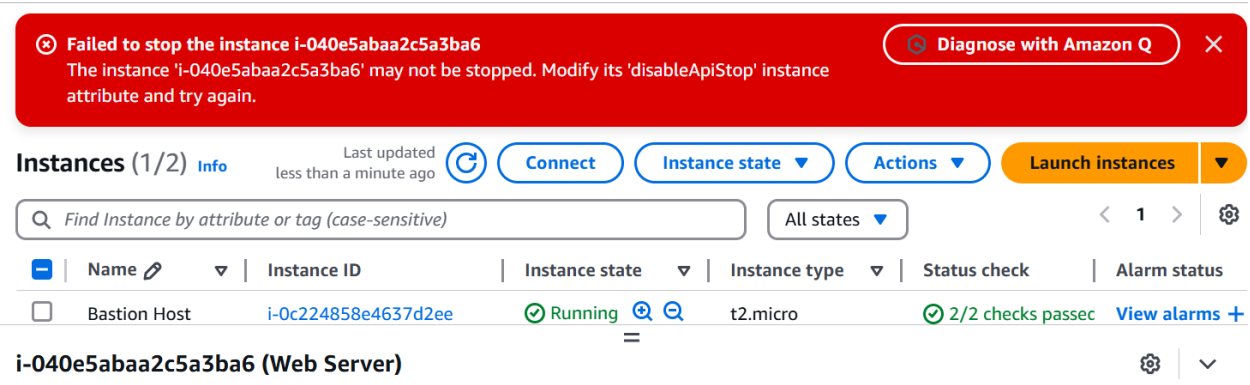
Next, search for and select Service Quotas, choosing AWS Services. In the Find services search bar, search for and select **Amazon Elastic Compute Cloud.**



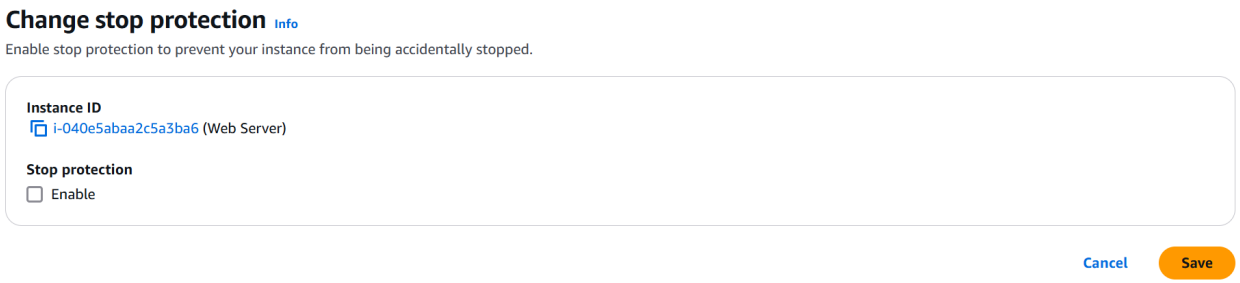
After selecting View quotas, enter into the View quotas search bar **running on-demand** and observe the search results.



Next, return to the EC2 console and choose Instances. Select the Web Server instance; in the Instance state drop-down menu, select Stop instance. Choose Stop. There should be an error message. This is due to the stop protection enabled earlier.



In the Actions drop-down menu, choose Instance setting and click on Change stop protection. Deselect the Enable setting and choose Save.



Try again to stop the instance. It should work now, as shown by the success message below:

PROBLEMS:

Although there weren't significant challenges, when I was configuring the web server in Lab 2, the time for the 2 checks to pass took an unexpectedly long time, delaying me while I was completing the tasks.

CONCLUSION:

These labs, although time-consuming, were mostly straightforward and easy-to-comprehend. At the same time, I learned a lot about AWS services offered by Amazon. I became a lot more familiar with the AWS Console used.

SIGN-OFF:

Lab 1:

Total score	40/40
TASK 2a - Added user-1 to S3-Support group	5/5
TASK 2b - Added user-2 to EC2-Support group	5/5
TASK 2c - Added user-3 to EC2-Admin group	5/5
TASK 3a - user-1 logged in	5/5
TASK 3b - user-2 logged in	5/5
TASK 3c - user-2 ec2 stop instance attempt	5/5
TASK 3d - user-3 logged in	5/5

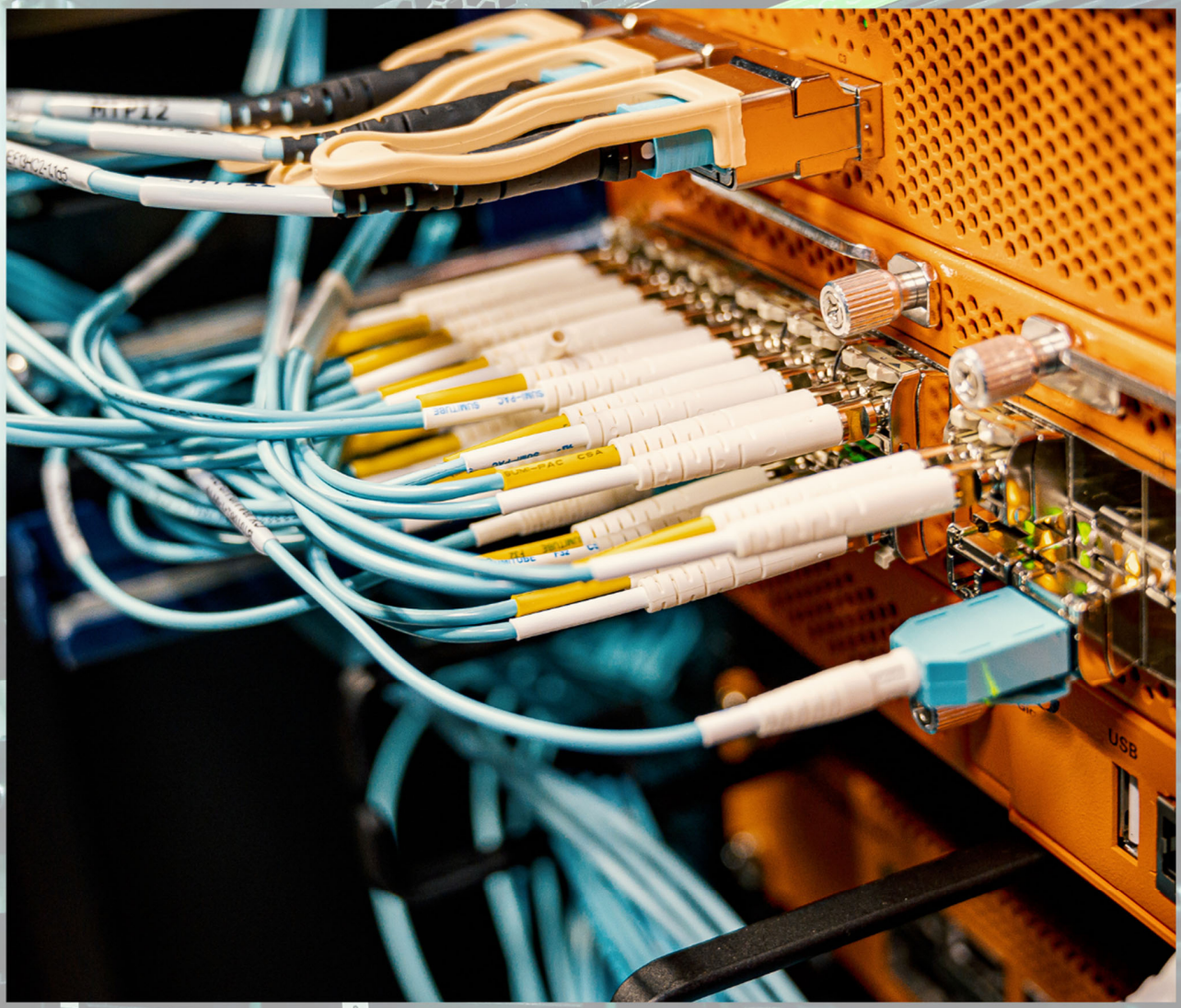
Lab 2:

Total score	30/30
Task 1 - VPC created correctly	5/5
Task 2a - New subnets created correctly	5/5
Task 2b - Subnet route table association	5/5
Task 3 - Security group created correctly	5/5
Task 4a - EC2 instance created correctly	5/5
Task 4b - EC2 instance website accessible	5/5

Lab 3:

Total score	25/25
Task 1 - EC2 instance created correctly	5/5
Task 2 - get system log requested	5/5
Task 3 - security group updated	5/5
Task 4 - EC2 instance updated	5/5
Task 6 - Instance stopped on second try	5/5

Lab 6: Amazon Cloud Foundations 4-6



PURPOSE:

Learn to work with EBS, build a DB server, and load balance through different AWS services.

BACKGROUND INFORMATION ON LAB CONCEPTS:

EBS:

Amazon Elastic Block Store (Amazon EBS) allows for storage for EC2 instances. EBS volumes are attached to the network and are independent from instances (so if instances become unavailable, volumes may still exist). EBS volumes can be used as an EC2 instances boot partition, which can be stopped and restarted, so you only need to pay for the time you're using them. EBS volumes also offer more durability over traditional EC2 instances because they're automatically replicated on the backend.

DB Server:

Amazon Relational Database Service (Amazon RDS) facilitates the process of setting up and operating a cloud database, while allowing room for scalability. At the same time, network administrators can choose which database engines to deploy and scale. Amazon RDS will also manage backup, software patching, and recovery, while detecting failures immediately. AWS will also manage the hardware layers, unlike a traditional database server.

ELB and Auto Scaling:

Elastic Load Balancing (ELB) automatically distributes traffic across EC2 instances, allowing for an easy way to achieve fault tolerance. It also allows customers to monitor the health and performance in real time and has additional security features. Auto Scaling allows for convenient scaling, according to your pre-defined conditions and works best for applications with stable demand patterns. Auto Scaling also increases the amount of EC2 instances for change in demand to maintain the performance while minimizing the costs through decreasing capacity at less in-demand times.

LAB SUMMARY:

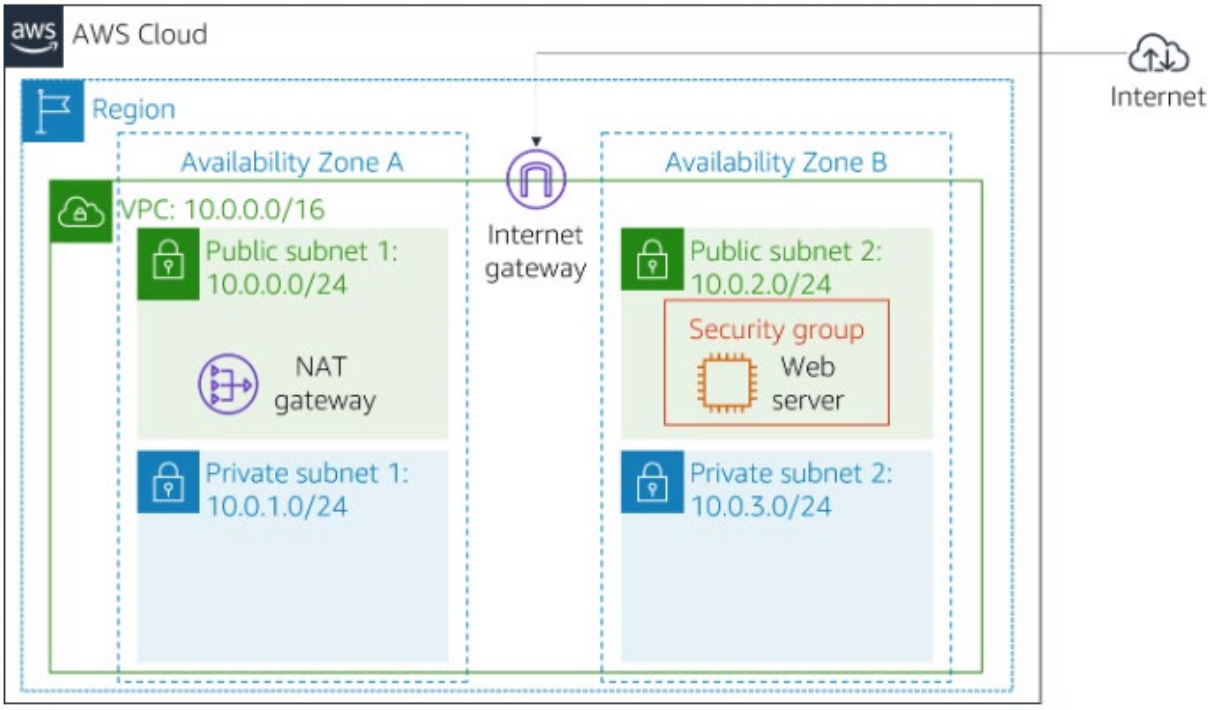
In these labs, I learned about different services that Amazon AWS offers—their purposes and how they work. I used the AWS Console and web browers to complete different tasks, including configurations and checks.

DIAGRAMS:

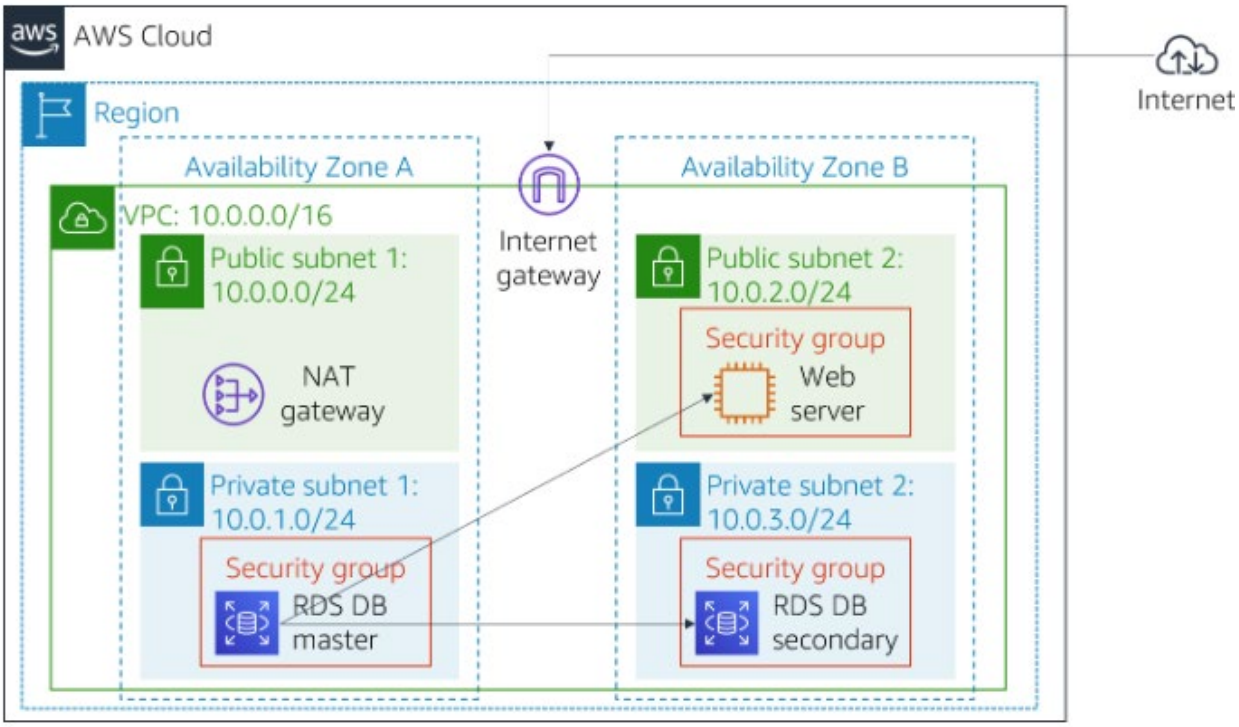
Lab 4

Lab 5

Beginning infrastructure:

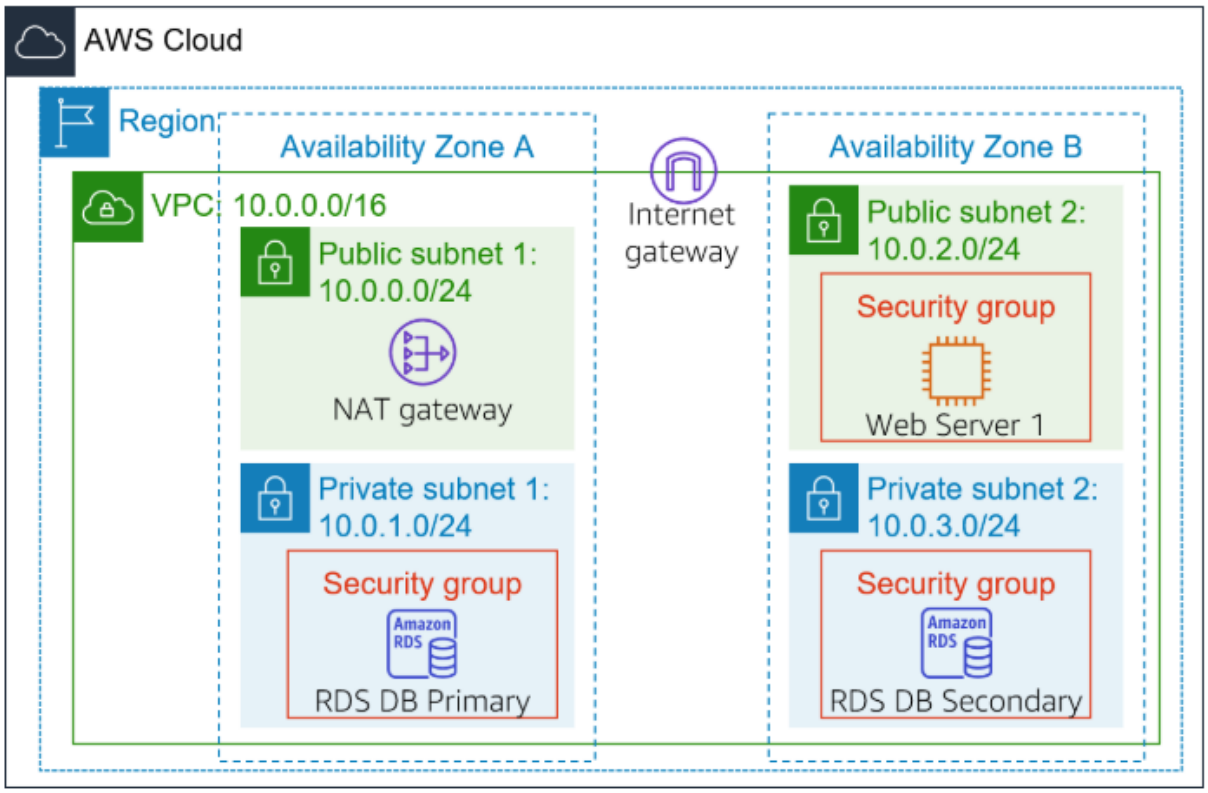


Ending infrastructure:

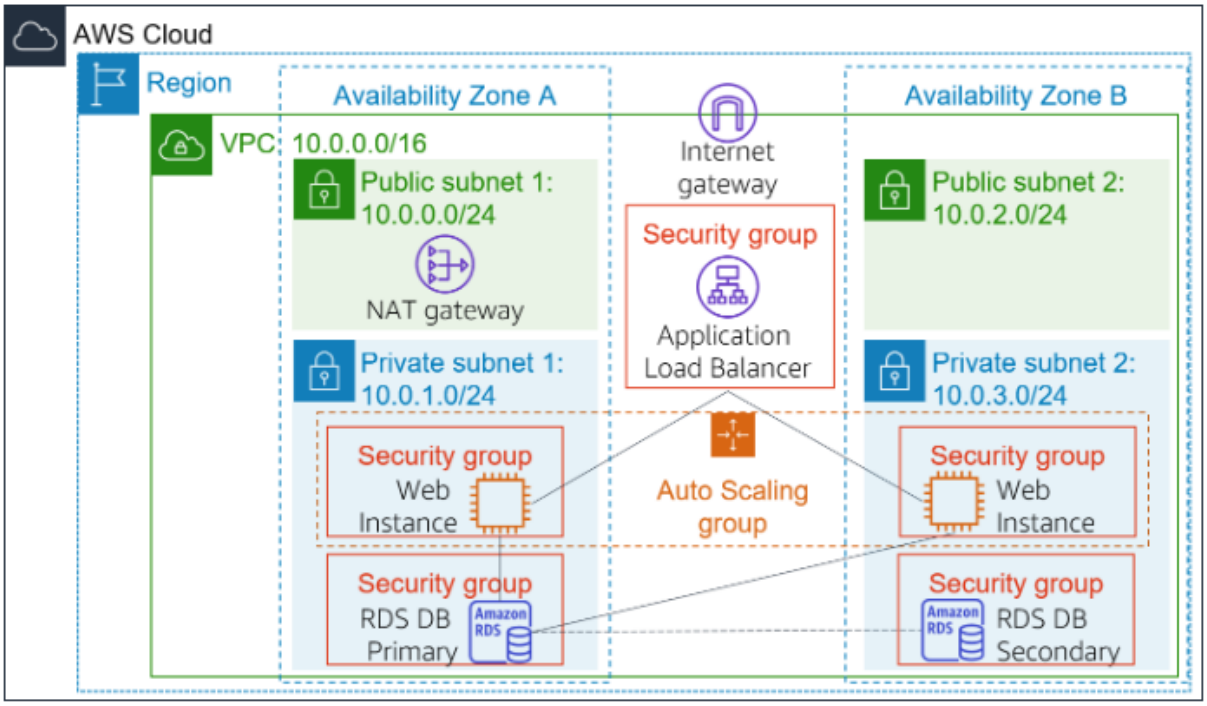


Lab 6

Beginning infrastructure:



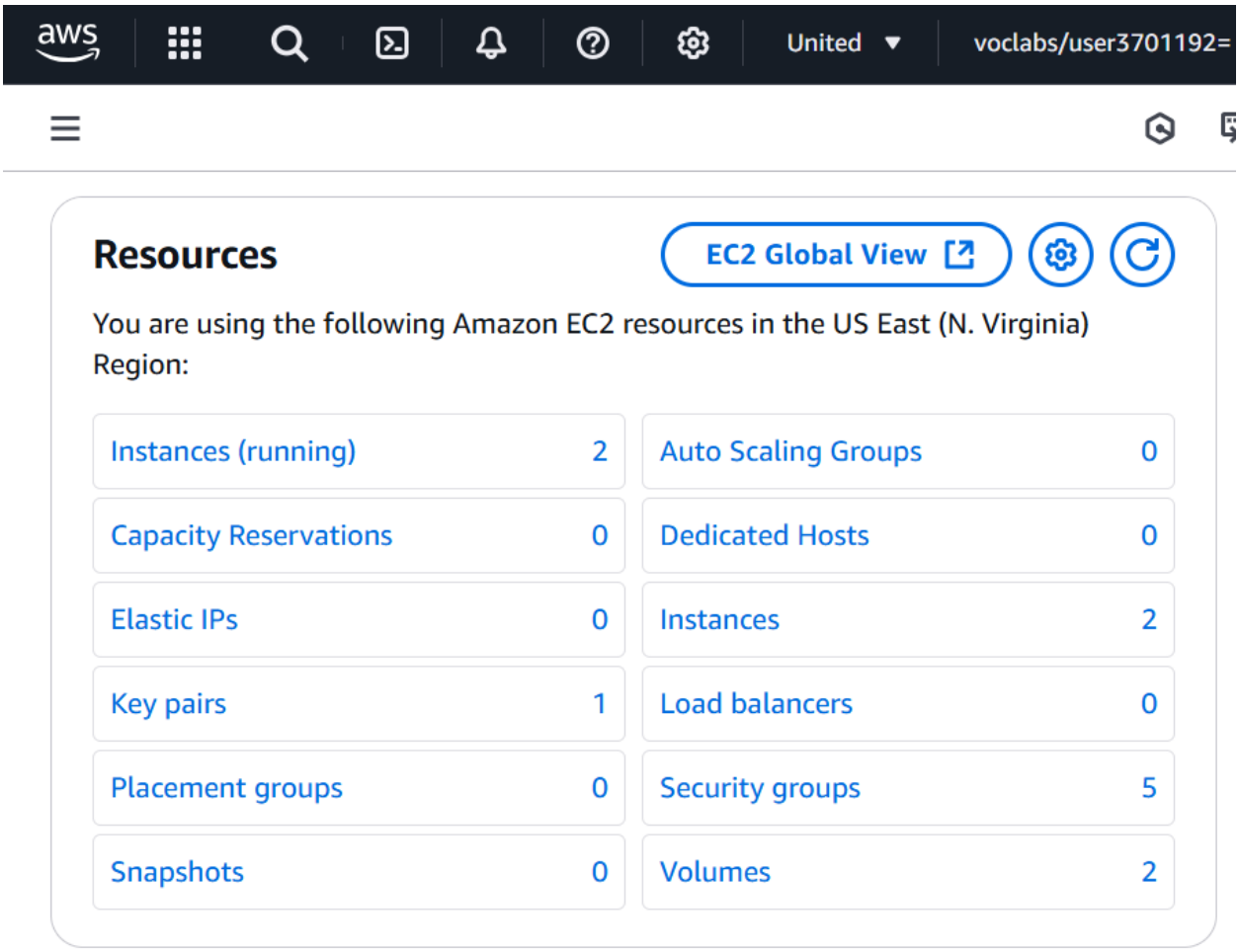
Ending infrastructure:



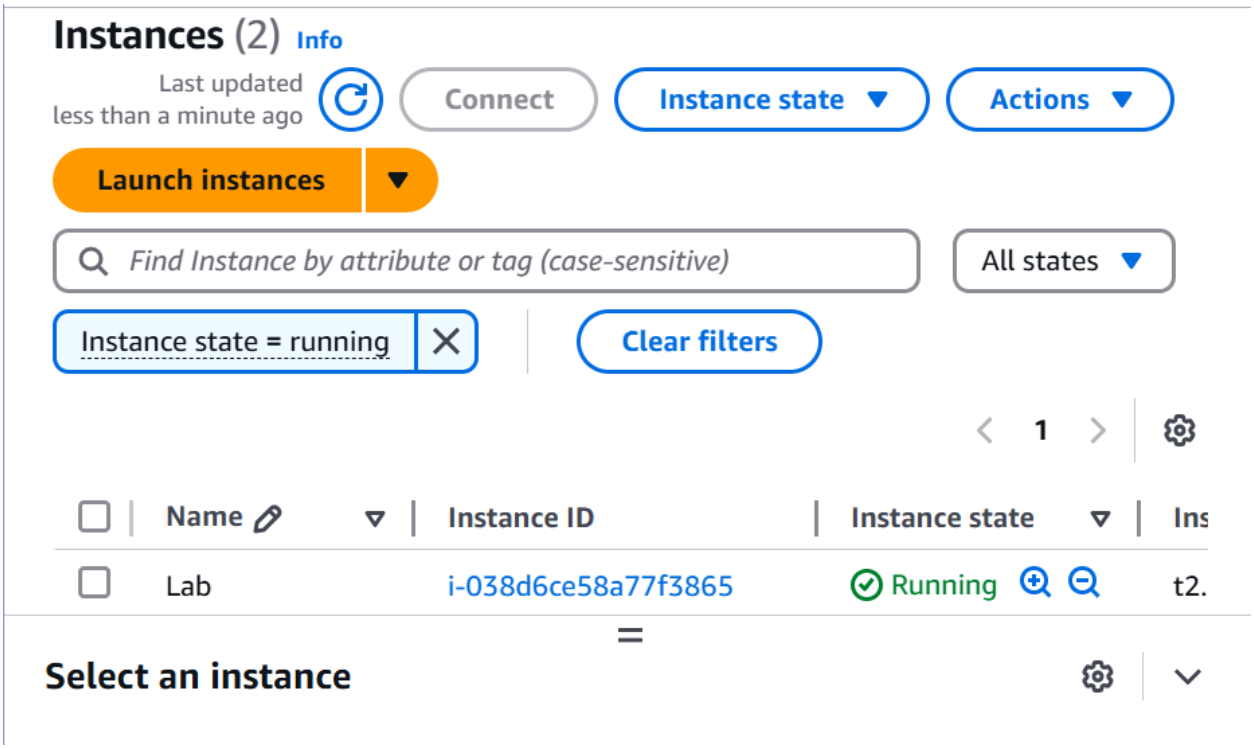
LAB COMMANDS:

LAB 4

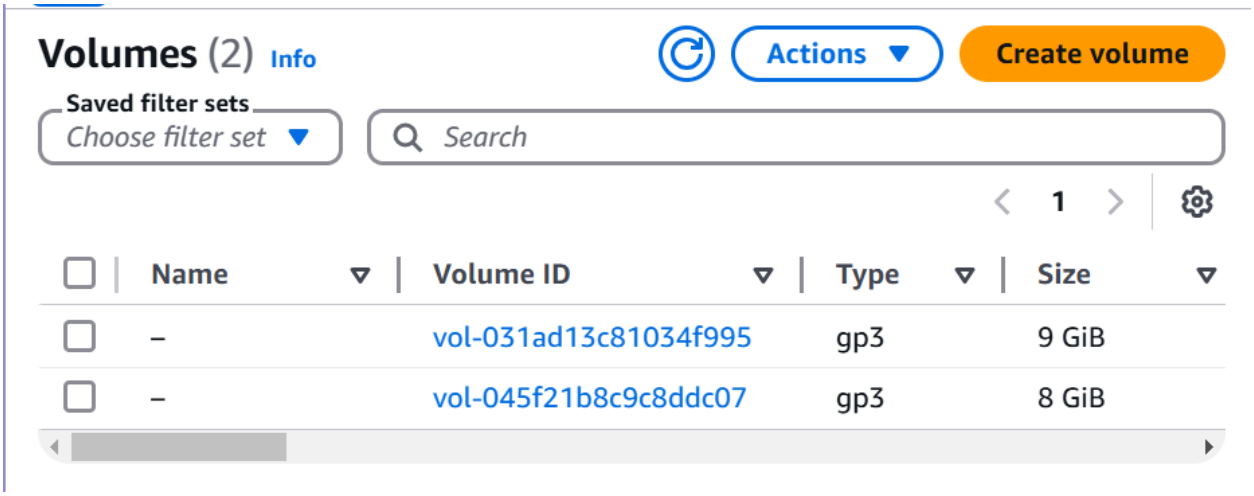
Navigate to the E2 Console page.



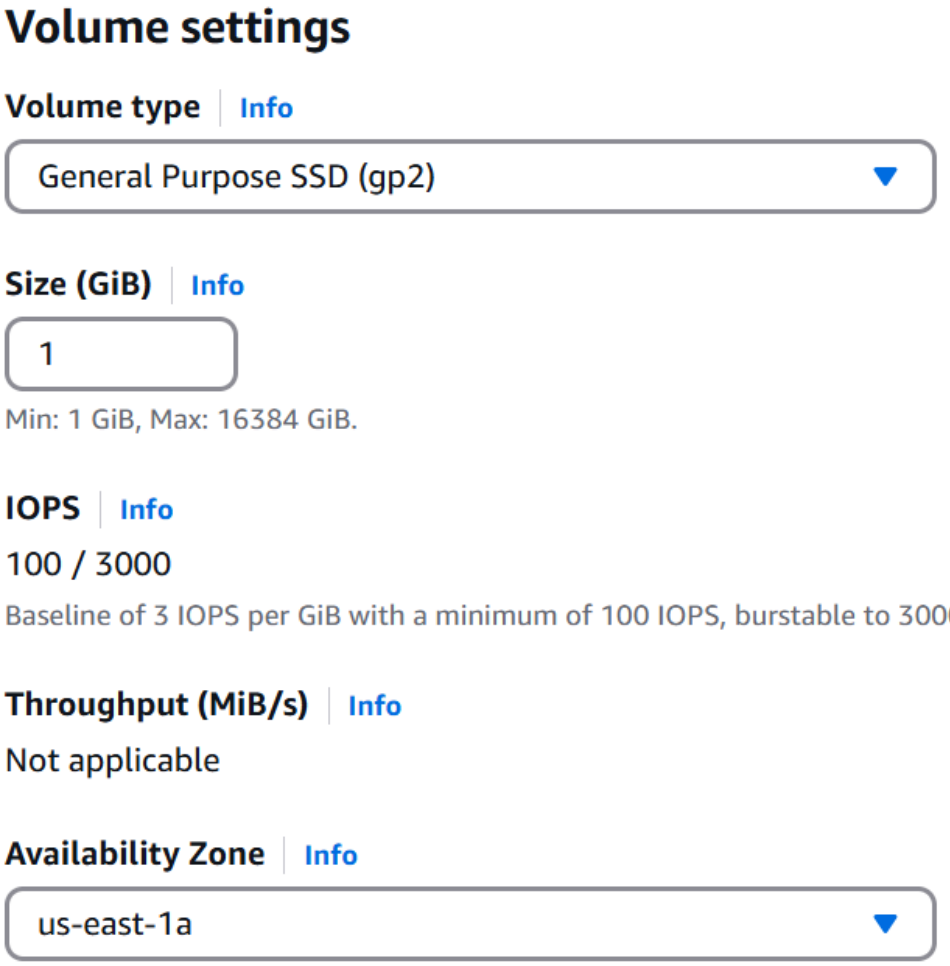
Next, click into Instances. Notice how an EC2 instance named Lab has already been set up.



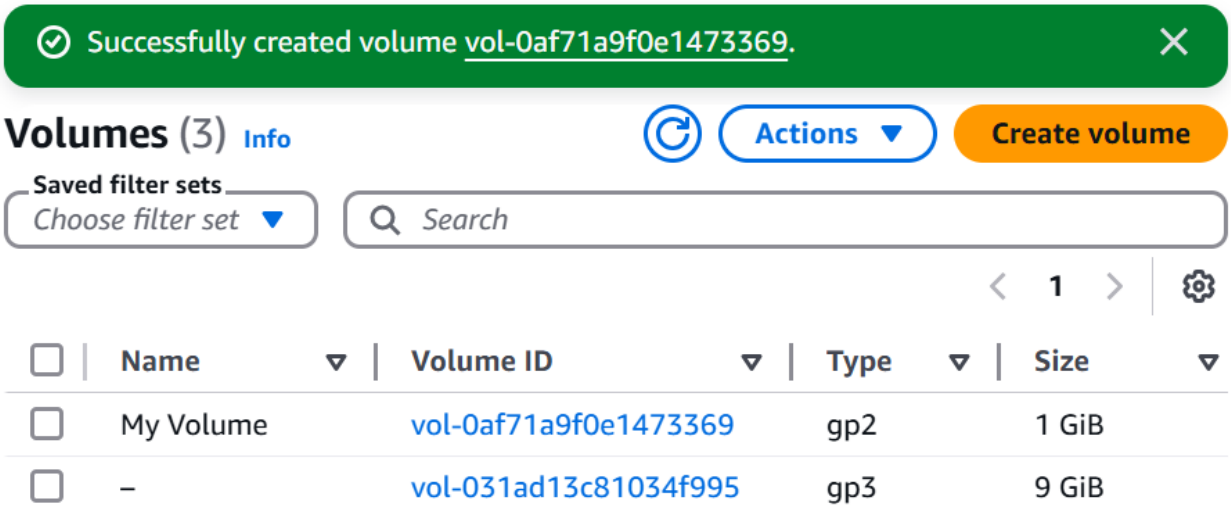
In the left navigation pane, select Volumes.



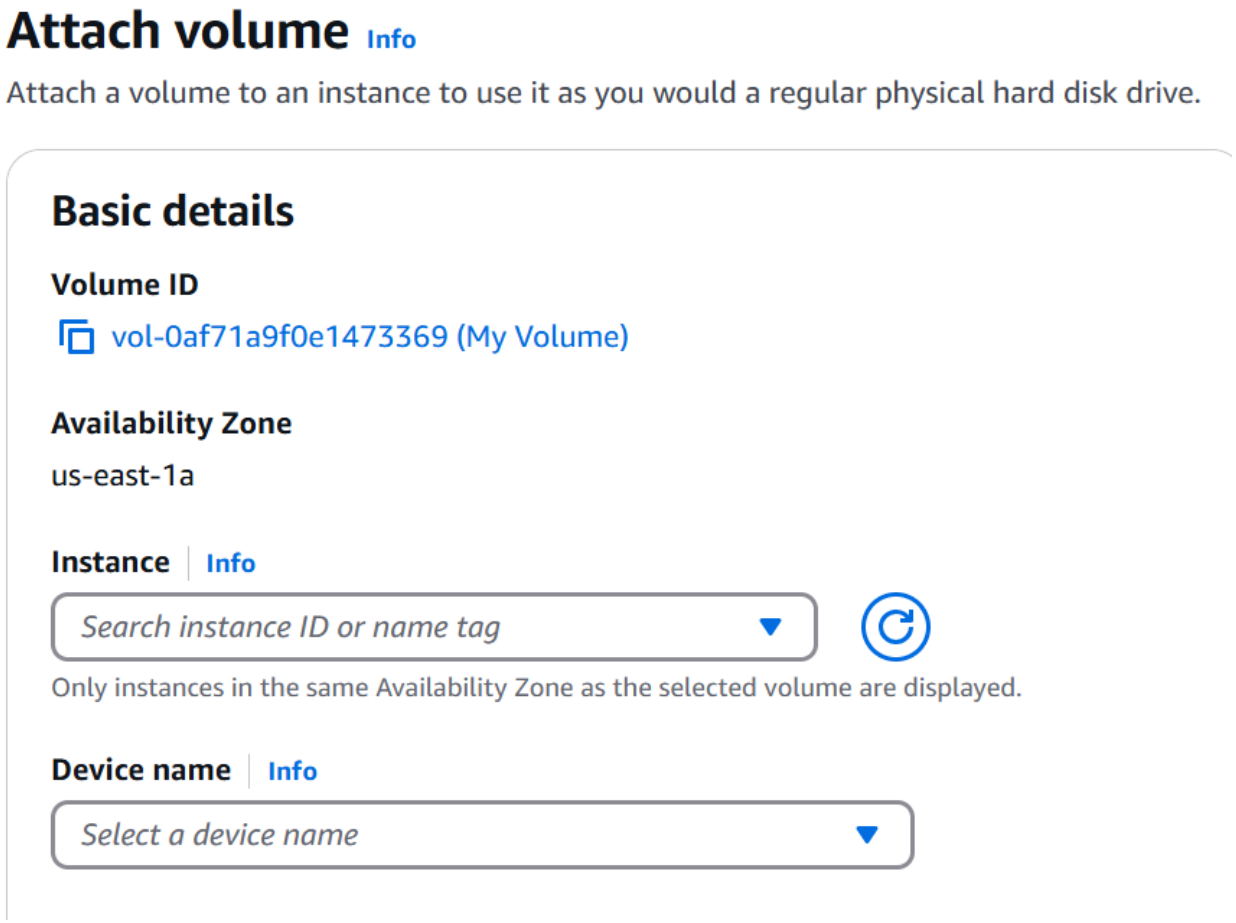
Click on Create volume and configure as shown below:



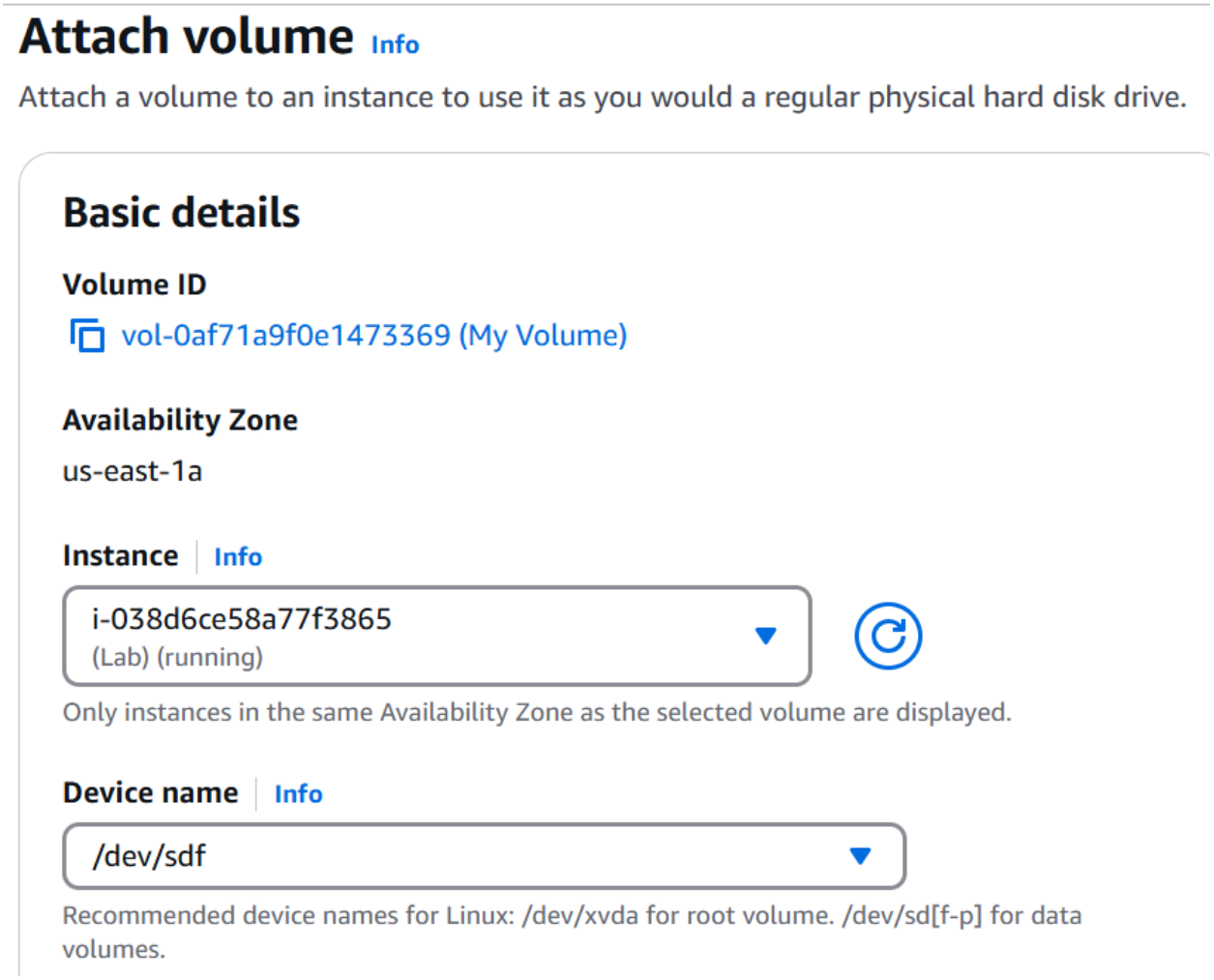
Click on Add tag, entering in a Key of **Name** and Value of **My Volume**. Choose Create Volume. You should see this screen:



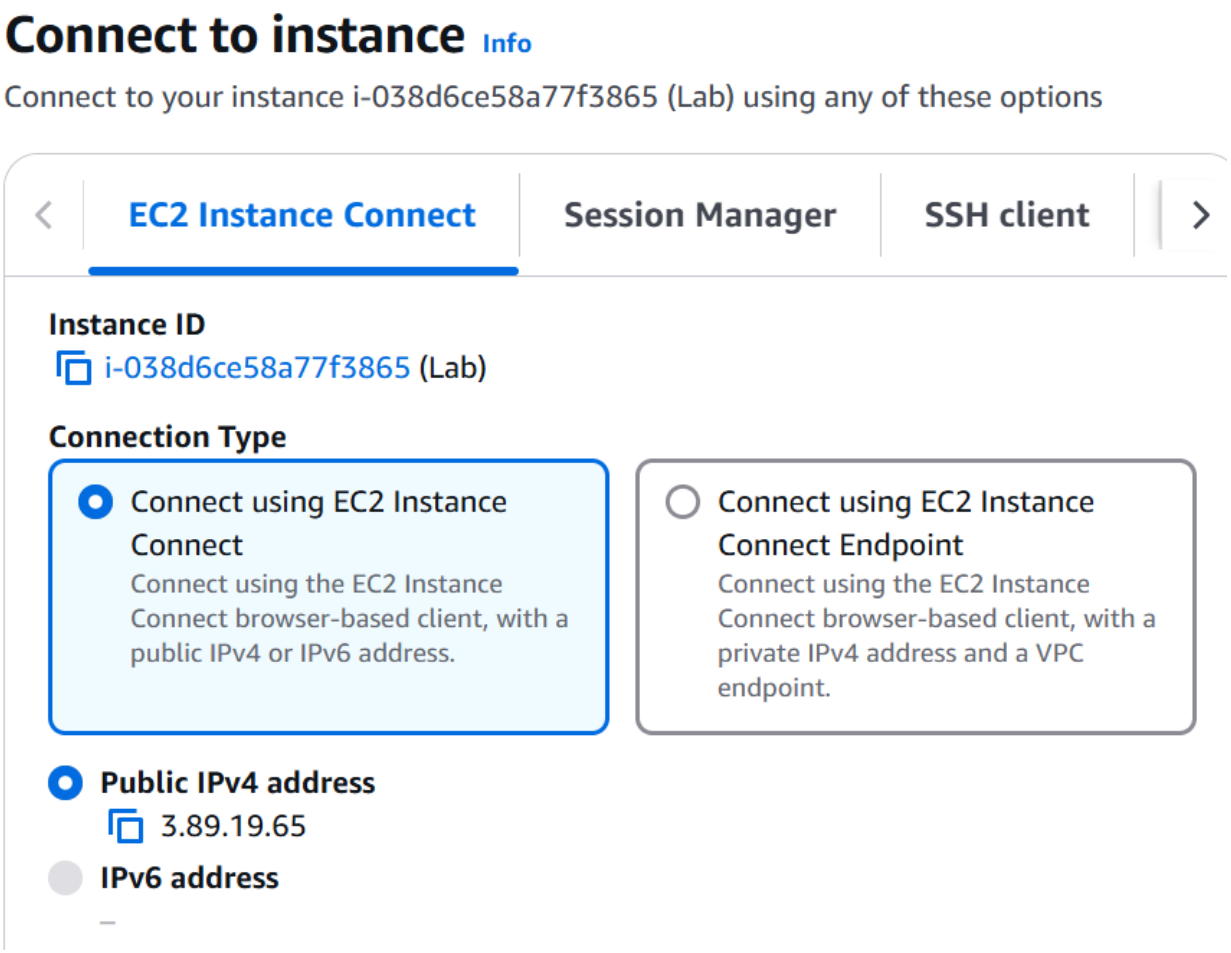
Select the checkbox next to My Volume and in the Actions drop-down menu, choose Attach volume. Note: you may need to wait and click on the refresh icon before you can do so.



Enter the Instance and Device name as shown below:



Click on Attach volume. Next, go back into EC2 Instances and click on the checkbox next to Lab instance, hitting Connect.



The above page should appear. Scroll to the very bottom and choose Connect. A terminal session should open, like the one below:

On the terminal session, run the following command: `df -h`. There should be an output like this:

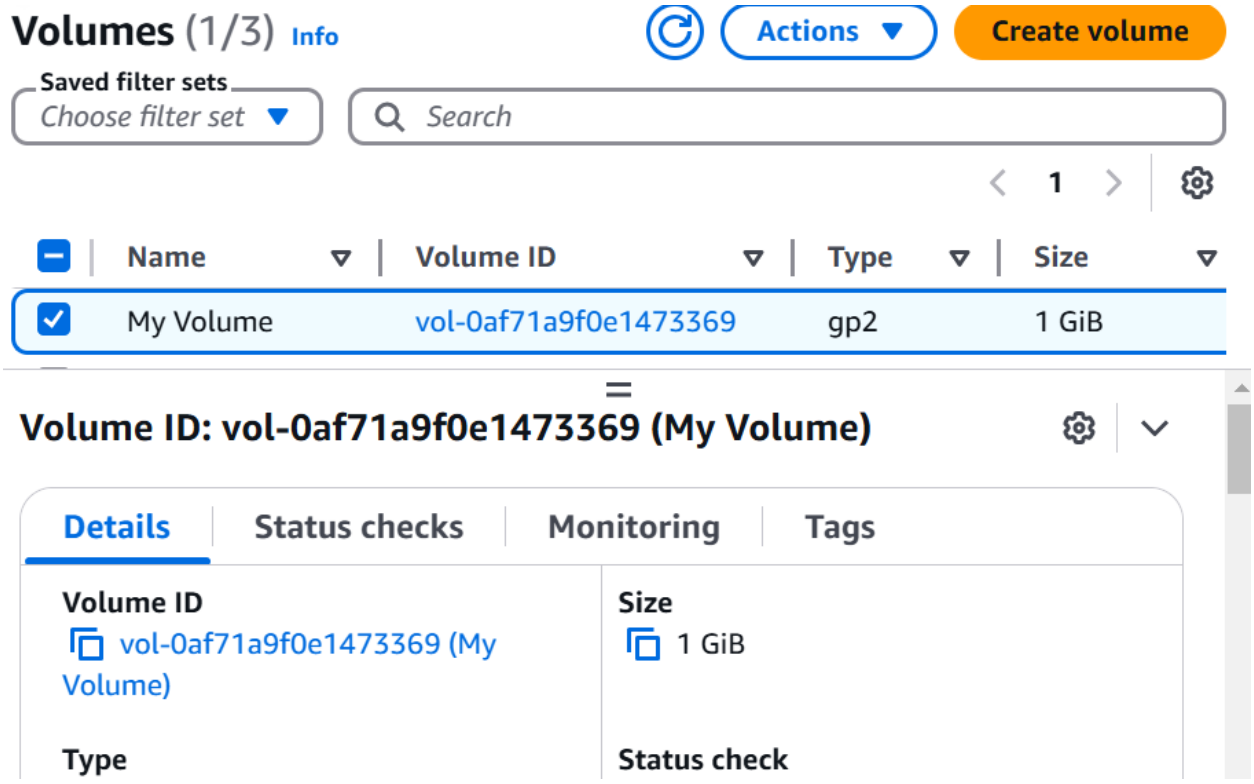
Next, enter this command: `sudo mkfs -t ext3 /dev/sdf`. This will create a new file system. `sudo mkdir /mnt/data-store` will create a directory for mounting the new storage volume; `sudo mount /dev/sdf mkdir /mnt/data-store` will mount the new volume. Run this: `echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab`; this will allow the volume to be mounted whenever the instance is started. View the configuration file using `cat /etc/fstab`; view the available storage again using `df -h` (see picture below for current storage).

```
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  456K  190M   1% /run
/dev/xvda1      8.0G  1.6G  6.4G  20% /
tmpfs           475M     0  475M   0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs            95M     0   95M   0% /run/user/1000
/dev/xvdf       975M   60K  924M   1% /mnt/data-store
[ec2-user@ip-10-1-11-77 ~]$ cat /etc/fstab
#
```

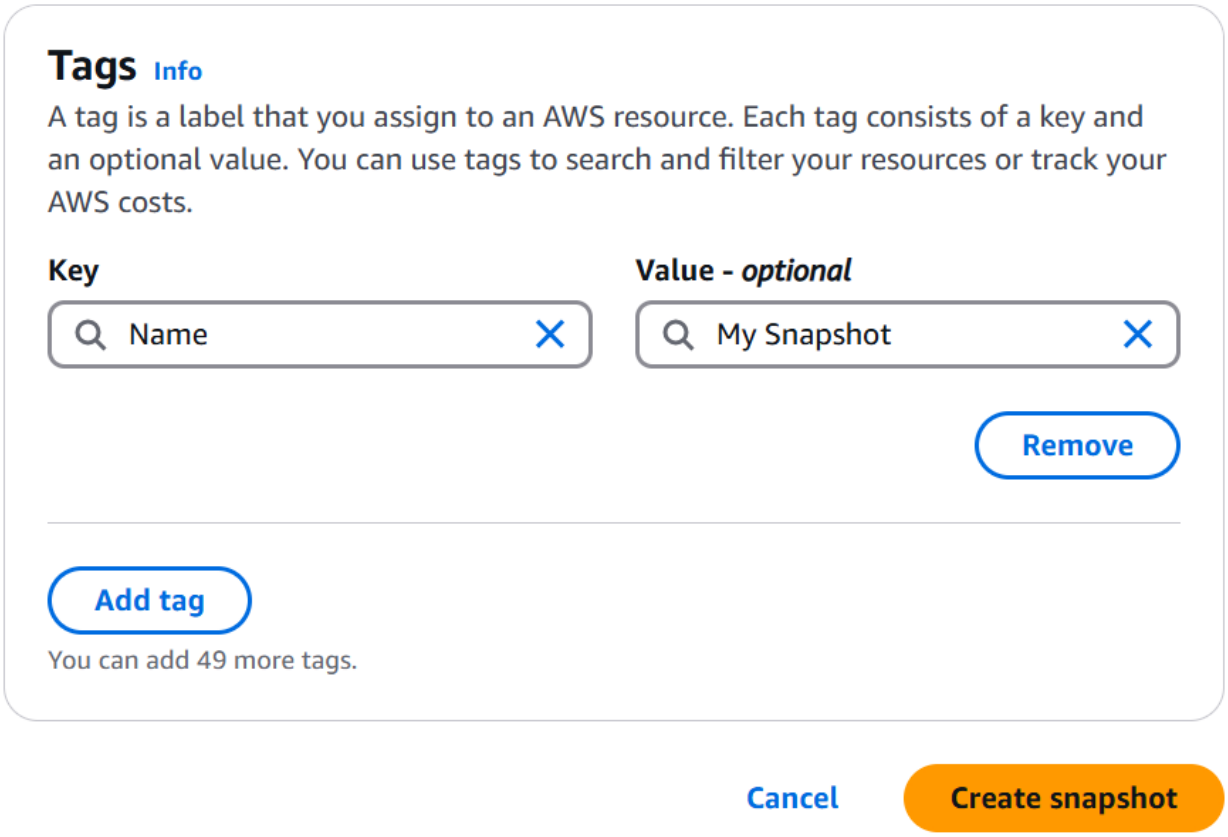
Next, create a file on the mounted volume and add some text using the first command shown below; the second command is used to verify the text has been written to the volume.

```
[ec2-user@ip-10-1-11-77 ~]$ sudo sh -c "echo some text has been wri
tten > /mnt/data-store/file.txt"
[ec2-user@ip-10-1-11-77 ~]$ cat /mnt/data-store/file.txt
some text has been written
[ec2-user@ip-10-1-11-77 ~]$
```

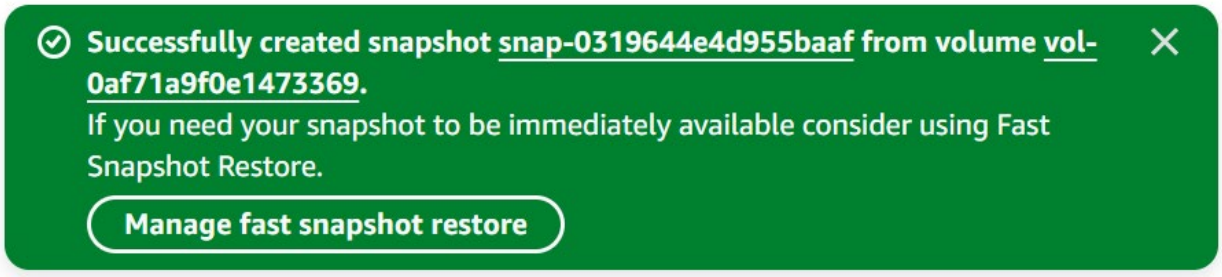
Go back into the EC2 Console, choose Volumes, and check the box next to My Volume.



In the Actions drop-down menu, select Create snapshot. Choose Add tag and configure Key as **Name** and Value as **My Snapshot**. Click Create snapshot.



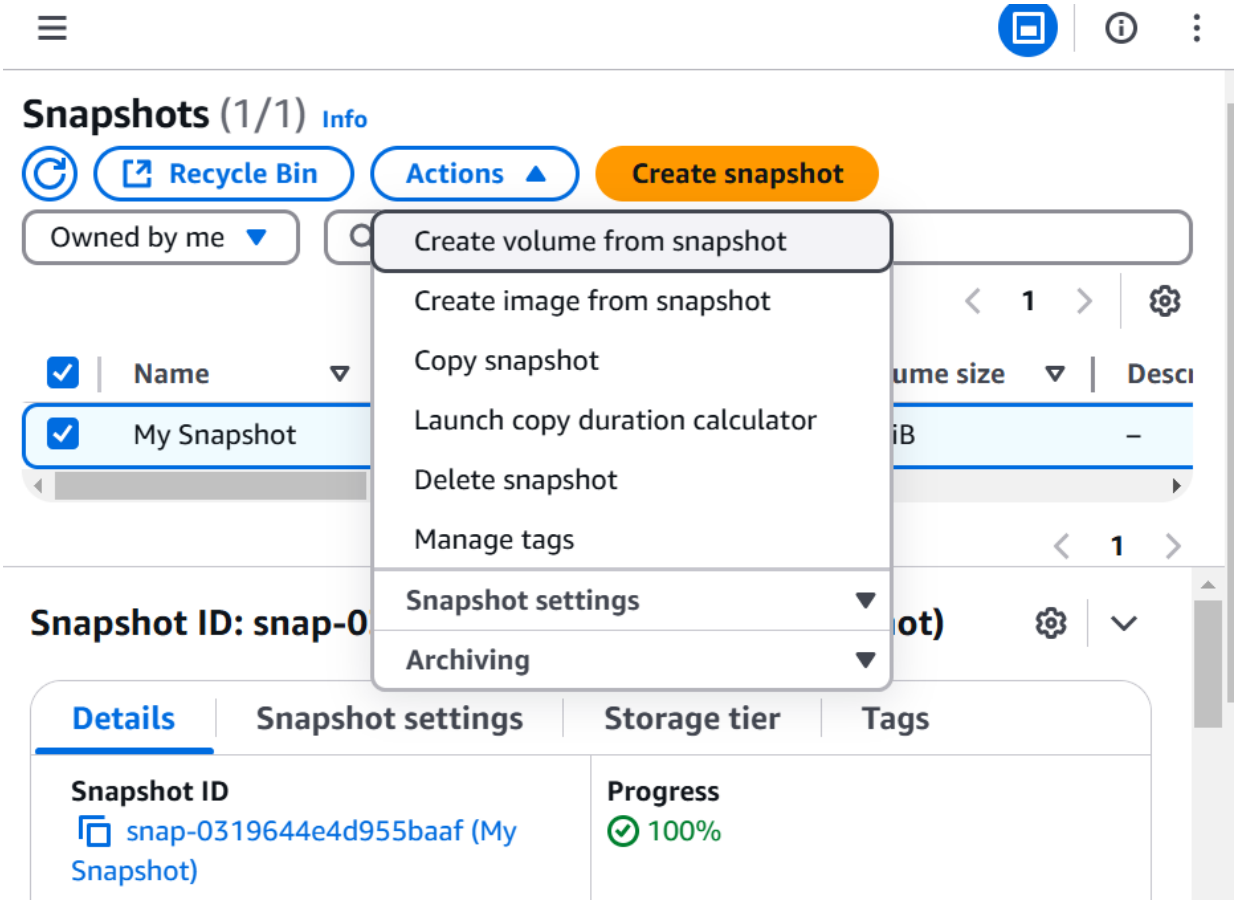
There should be a pop-up message like the following:



Next, in the left navigation pane, click on Snapshots, where the snapshot that was just created is displayed. In the EC2 Instance Connect terminal session, delete the file created on the volume using the command `sudo rm /mnt/data-store/file.txt`. Verify the file has been deleted using the command `ls /mnt/data-store/`.

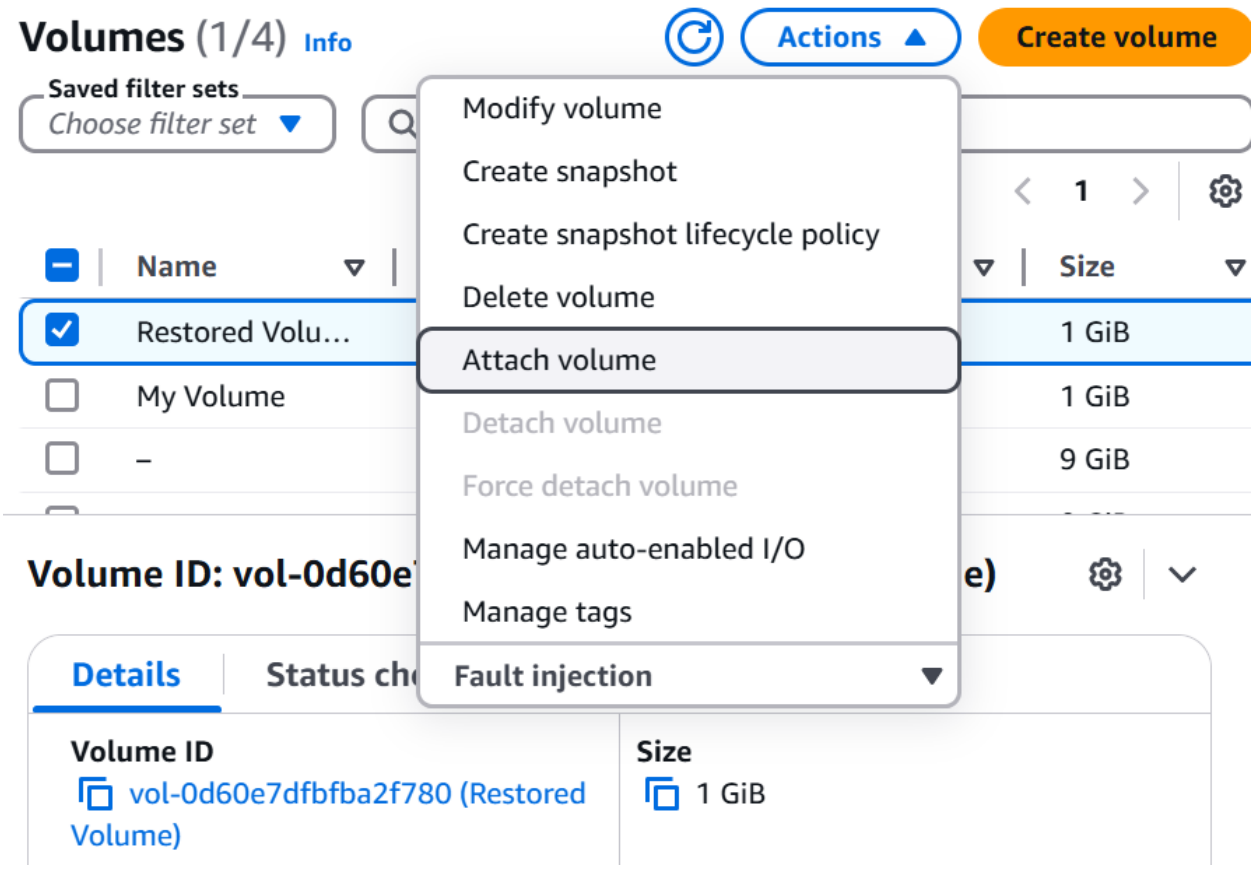
```
[ec2-user@ip-10-1-11-77 ~]$ sudo rm /mnt/data-store/file.txt
[ec2-user@ip-10-1-11-77 ~]$ ls /mnt/data-store
lost+found
```

In the EC2 console, check the box next to My Snapshot; in the Actions menu, click on Create volume from snapshot.

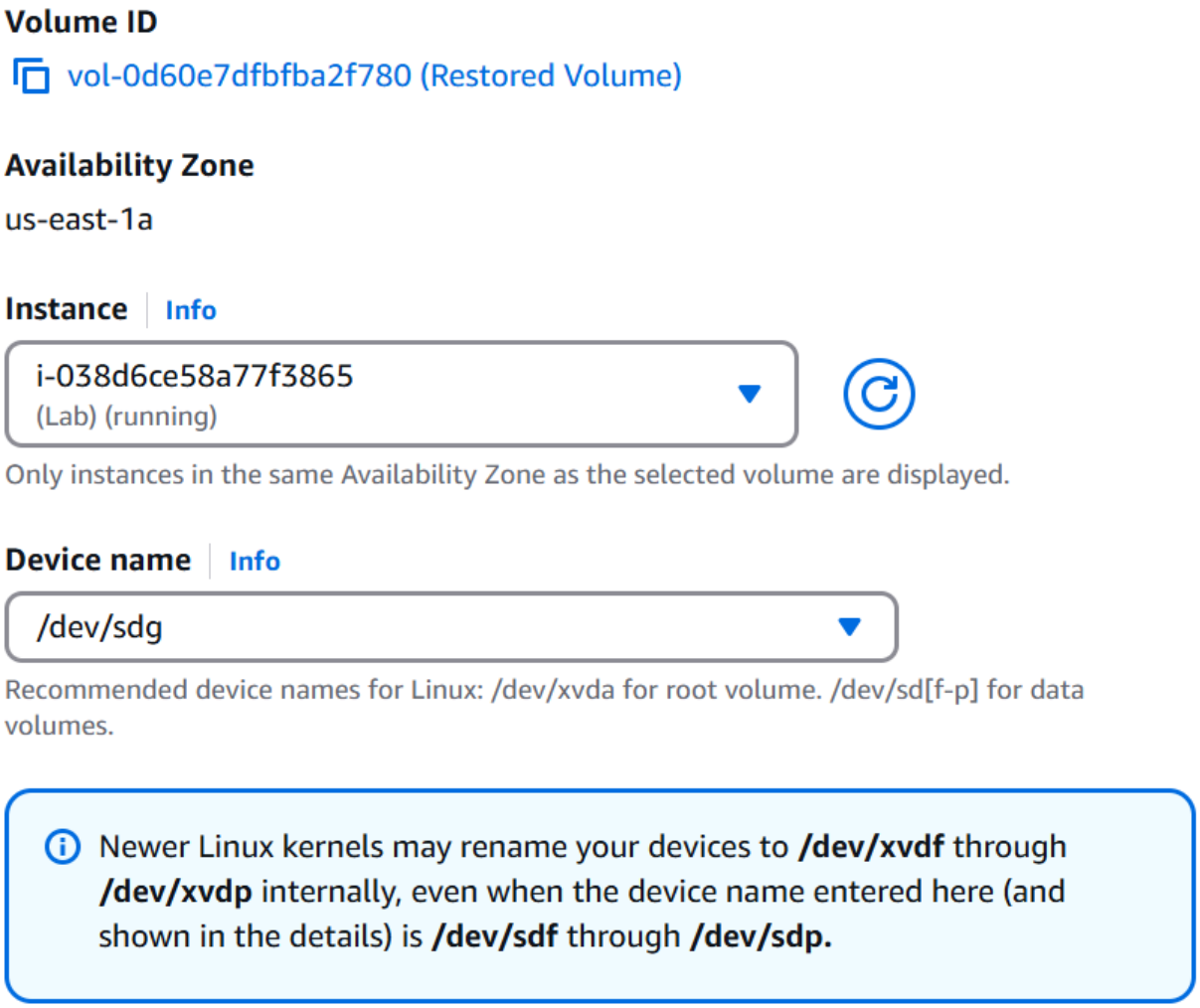


Use the same configurations for the original volume, but for the Value attribute under Add tag, instead of **Volume**, enter **Restored Volume**. A success message should pop up.

Next, in the left navigation pane, navigate back to Volumes and select the check box next to Restored Volume. In the Actions drop-down menu, click on Attach volume.



Configure as following:



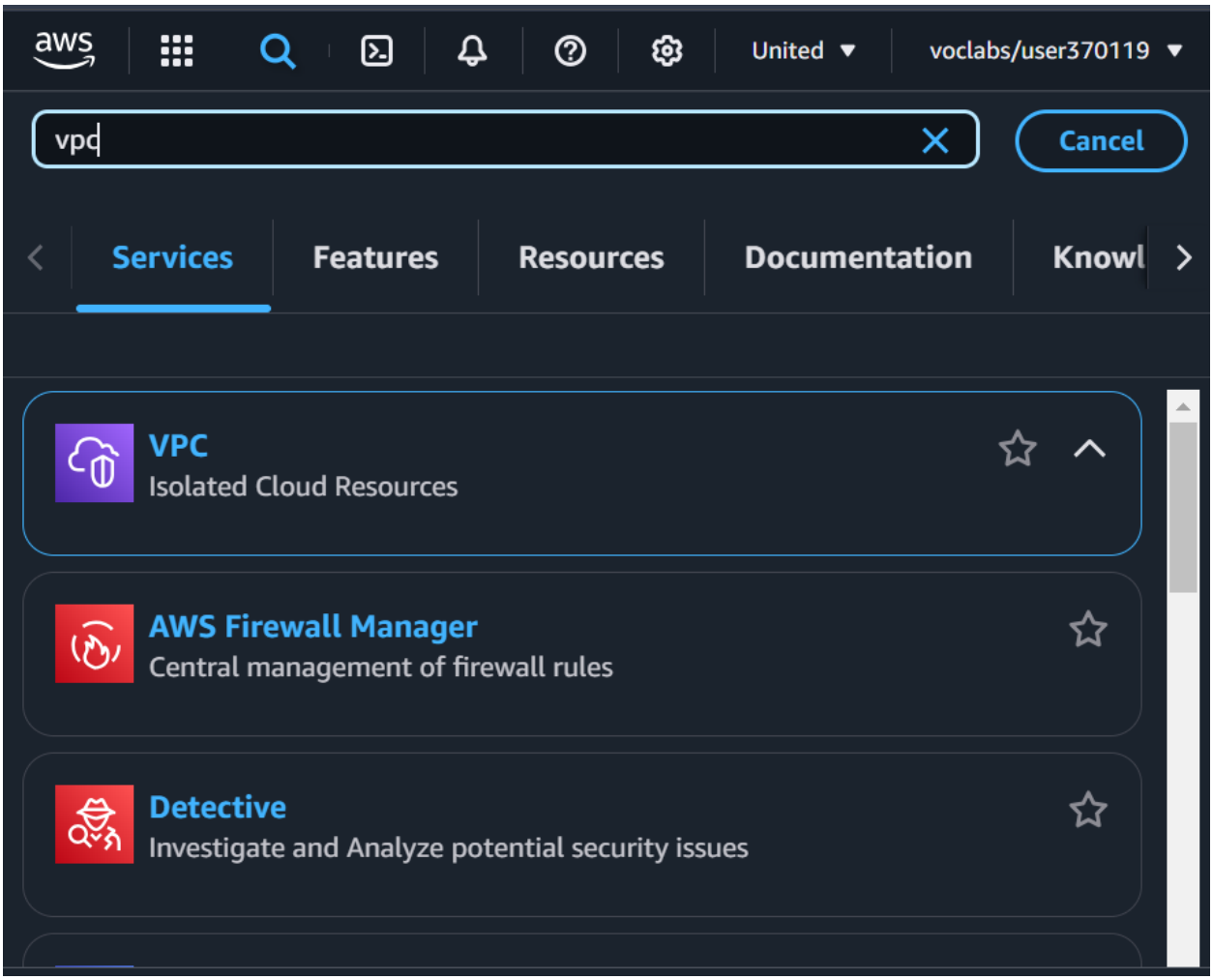
Choose Attach volume. Go back into the terminal session and enter the following commands: `sudo mkdir /mnt/data-store2` to create a directory for mounting the new storage volume, mount the volume using `sudo mount /dev/sdg /mnt/data-store2`, and verify this by using `ls /mnt/data-store2/`.

```
[ec2-user@ip-10-1-11-77 ~]$ sudo mkdir /mnt/data-store2
[ec2-user@ip-10-1-11-77 ~]$ sudo mount /dev/sdg /mnt/data-store2
[ec2-user@ip-10-1-11-77 ~]$ ls /mnt/data-store2/
file.txt  lost+found
```

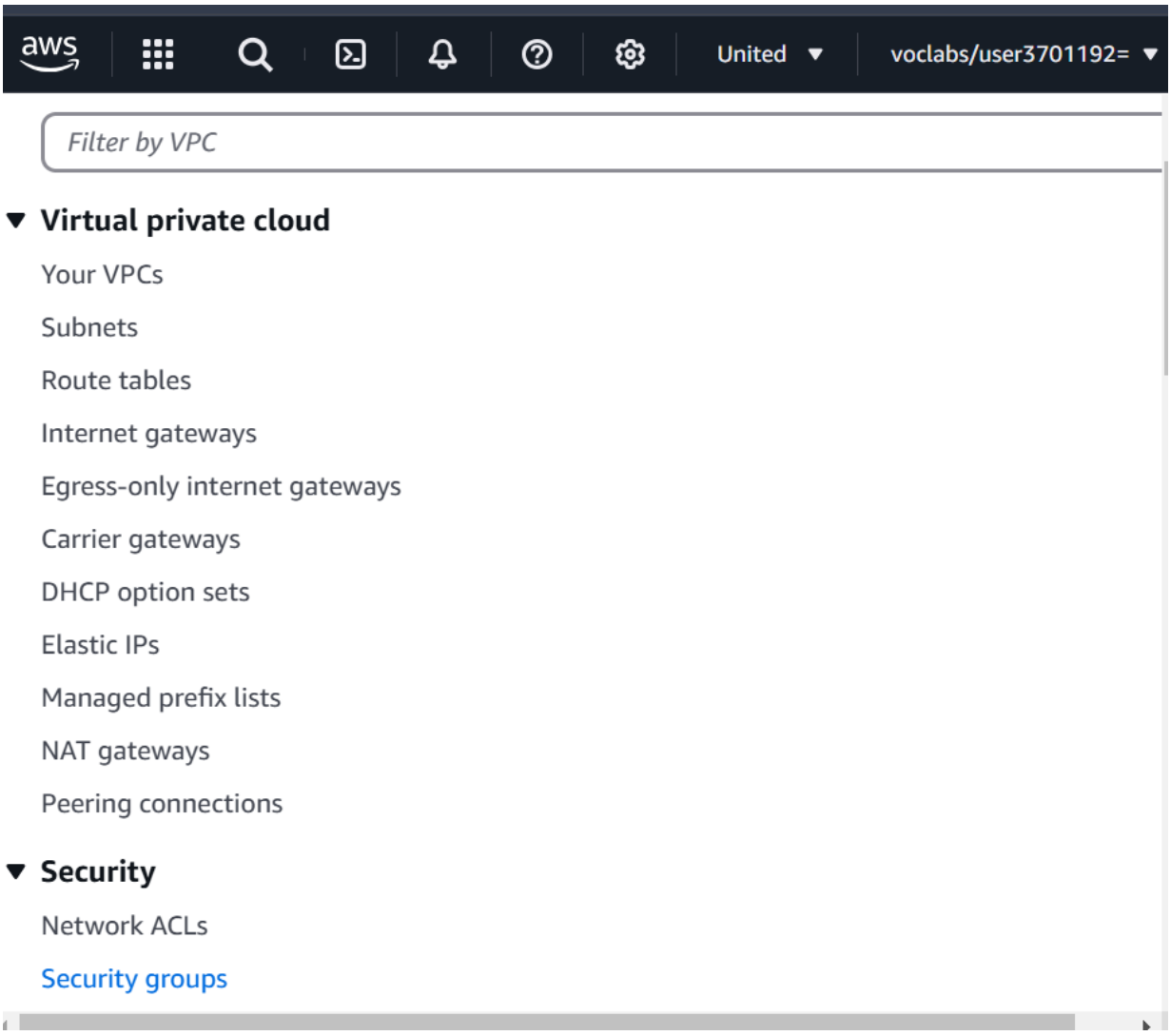
The file.txt is the verification that shows that mounting was successful.

LAB 5

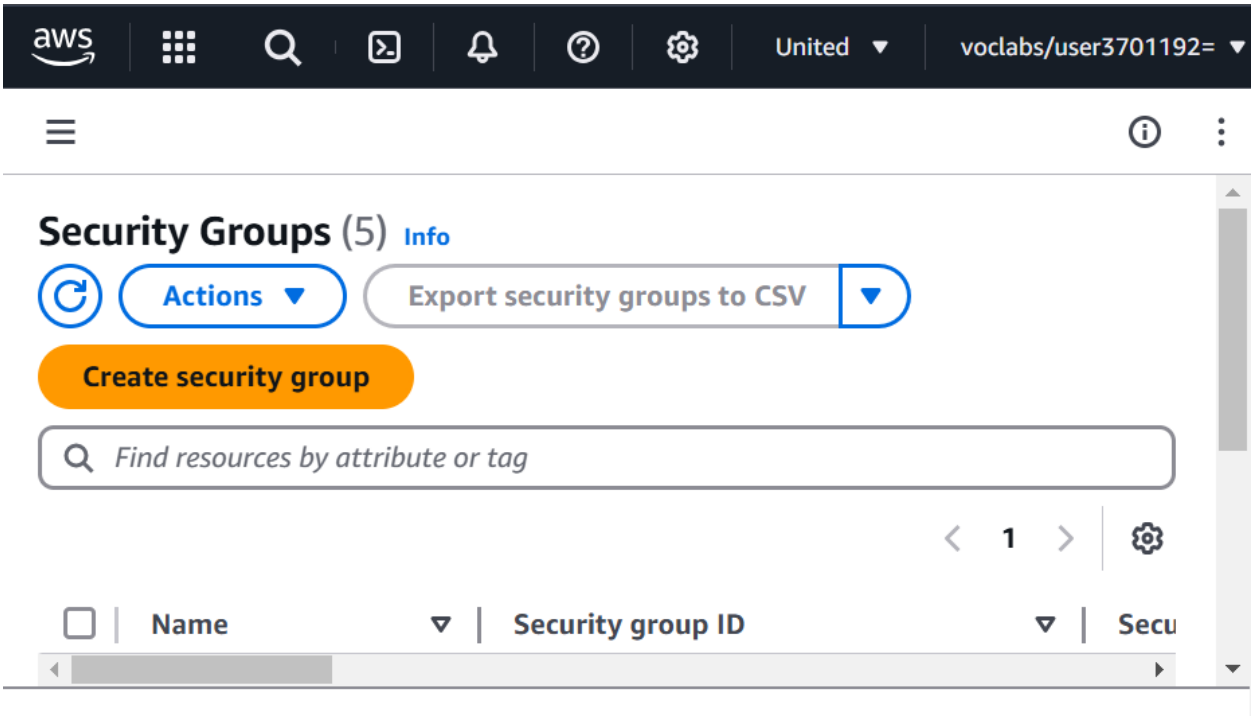
In the search box, search for and enter VPC.



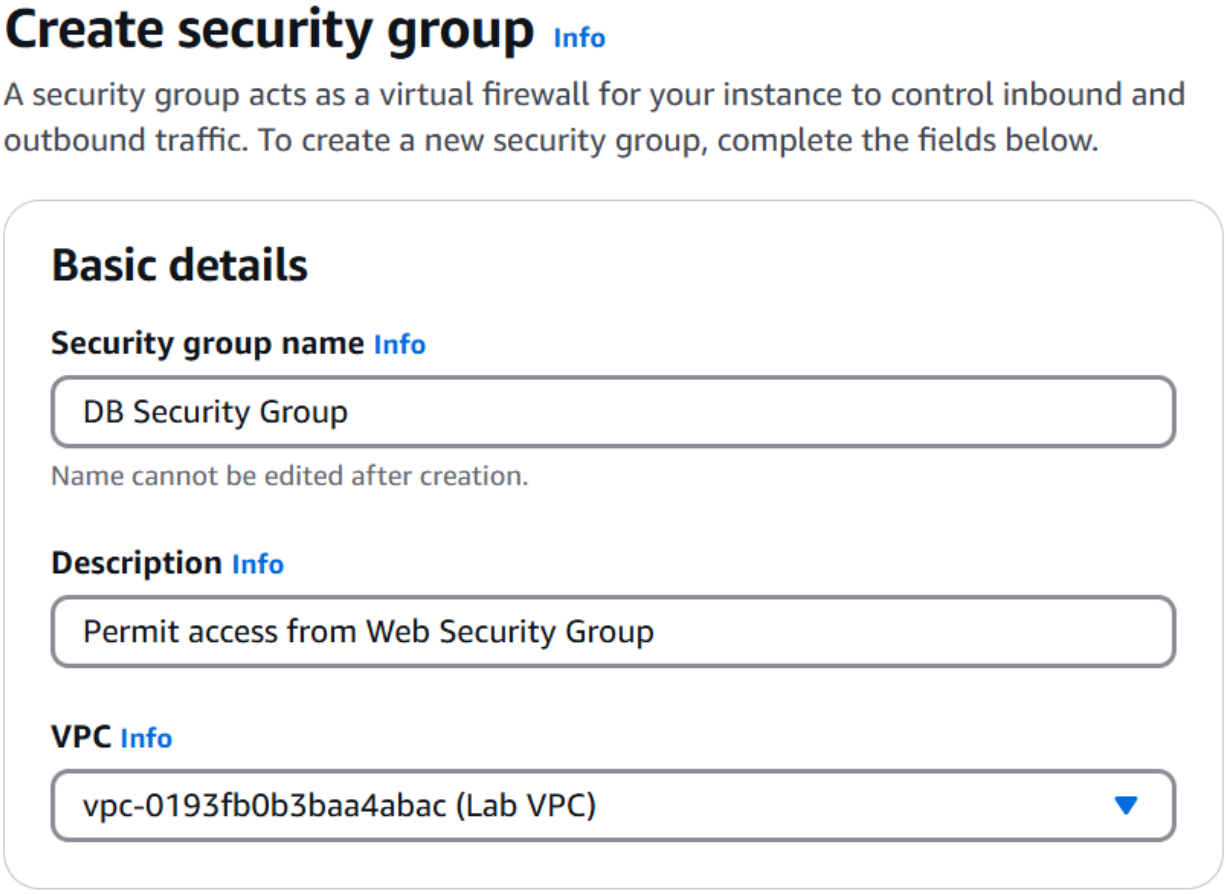
Select Security groups under the left navigation pane.



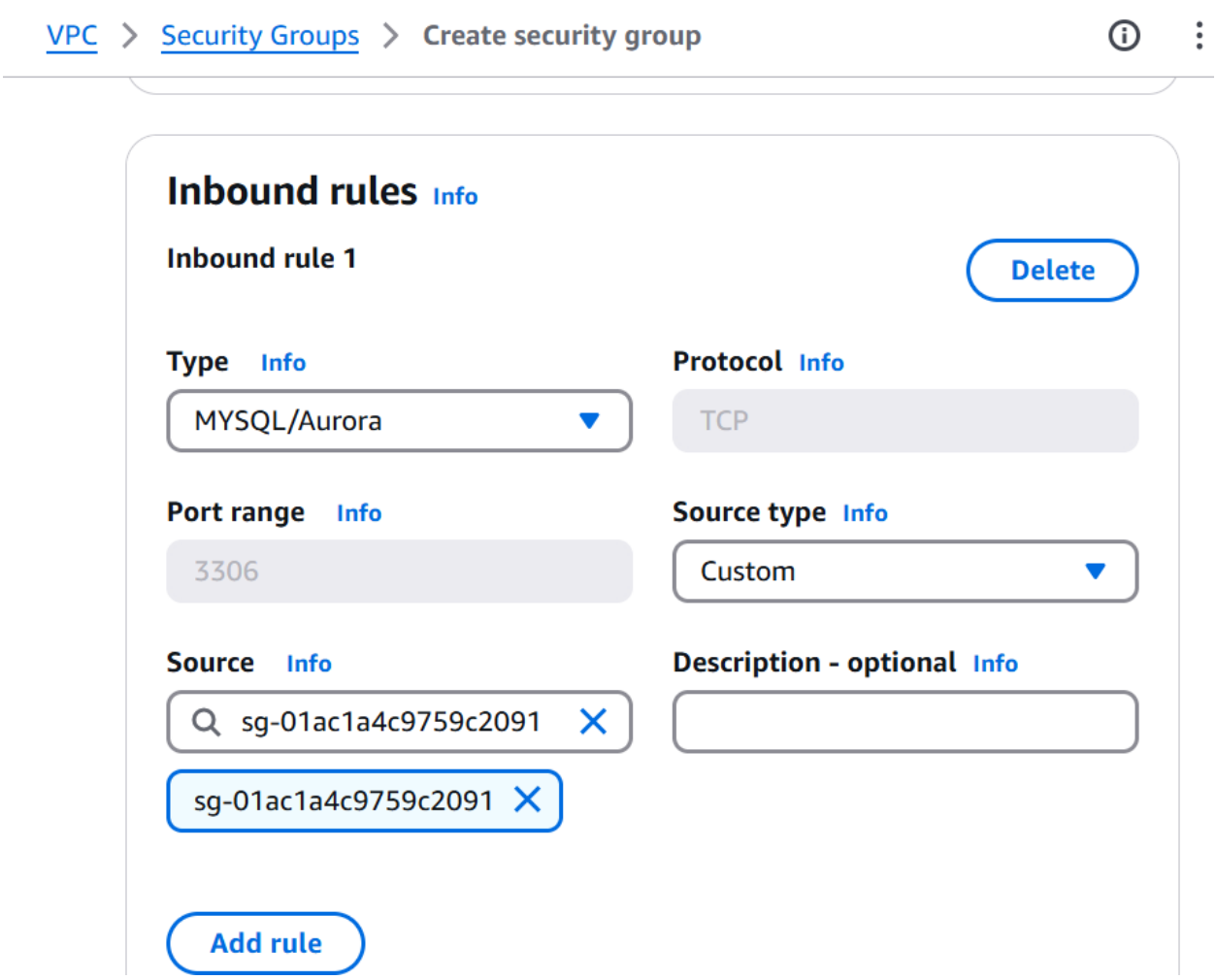
This should take you to a screen like the one below.



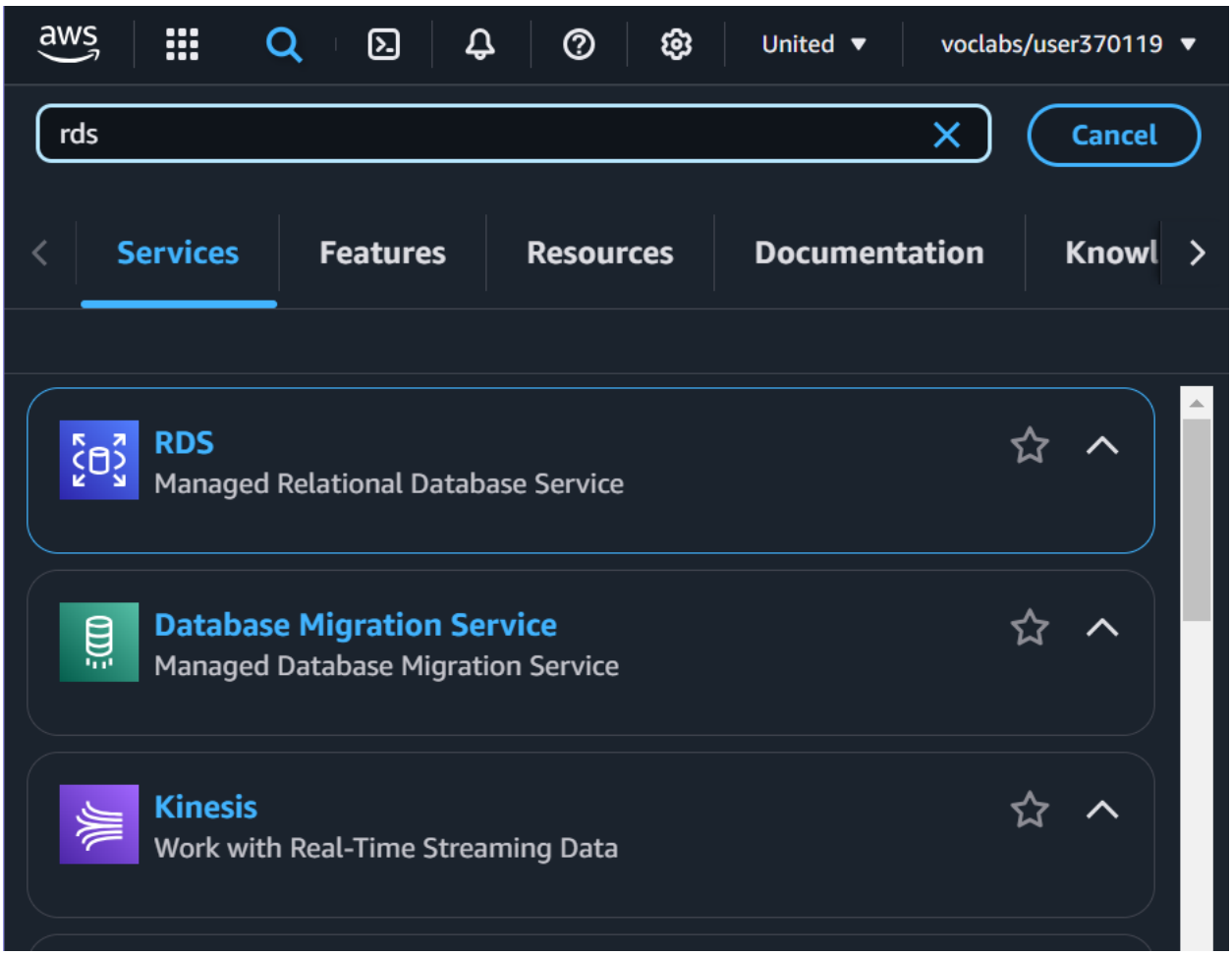
Select Create security group and configure as shown below.



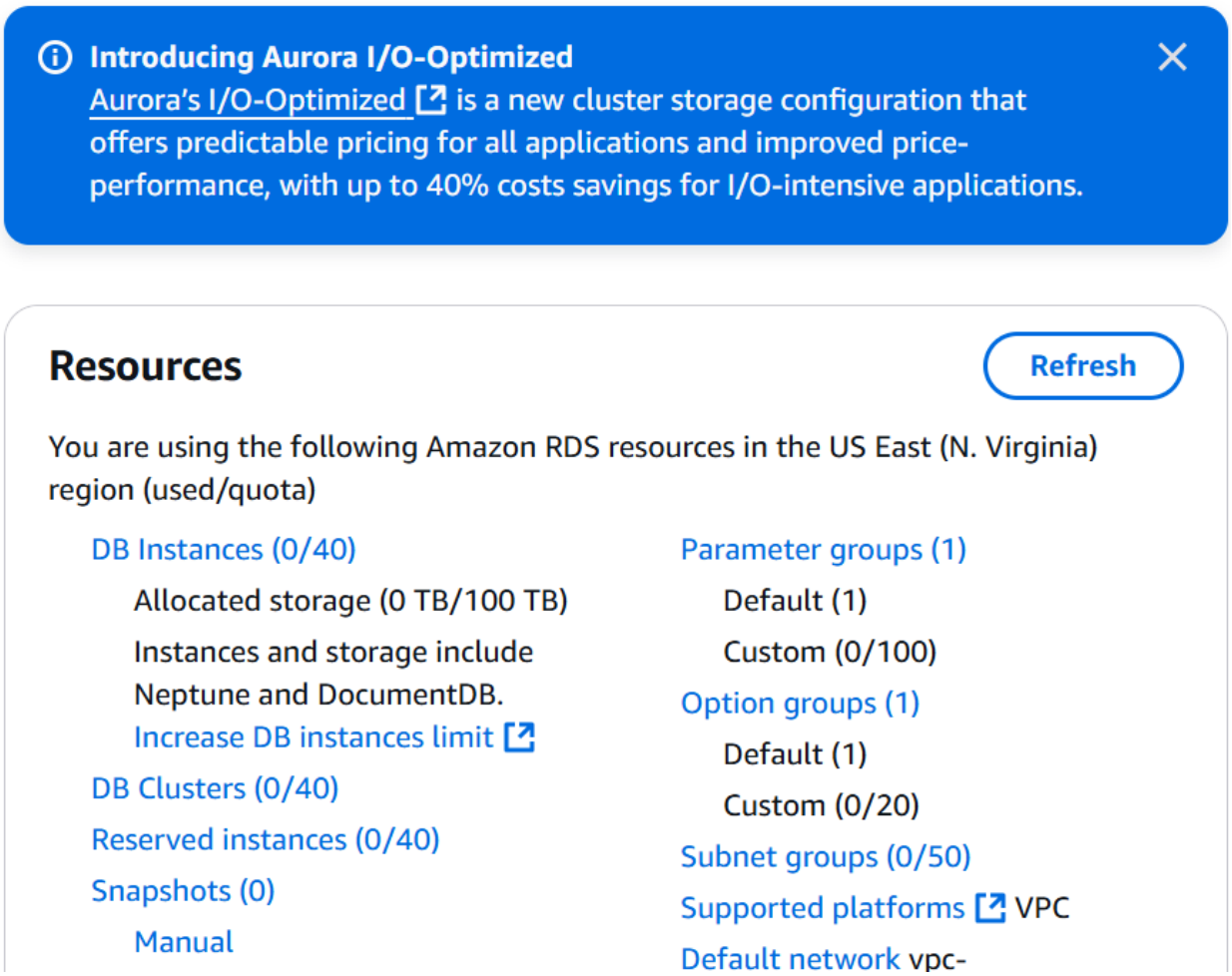
Add a rule under the Inbound rules section. Configure the following:



In the search box, open up RDS.



It should open up a page like the one below.



Next, in the left navigation pane, click on Subnet groups and click on Create DB Subnet Group, configuring as shown below:

Subnet group details

Name

You won't be able to modify the name after your subnet group has been created.

DB-Subnet-Group

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

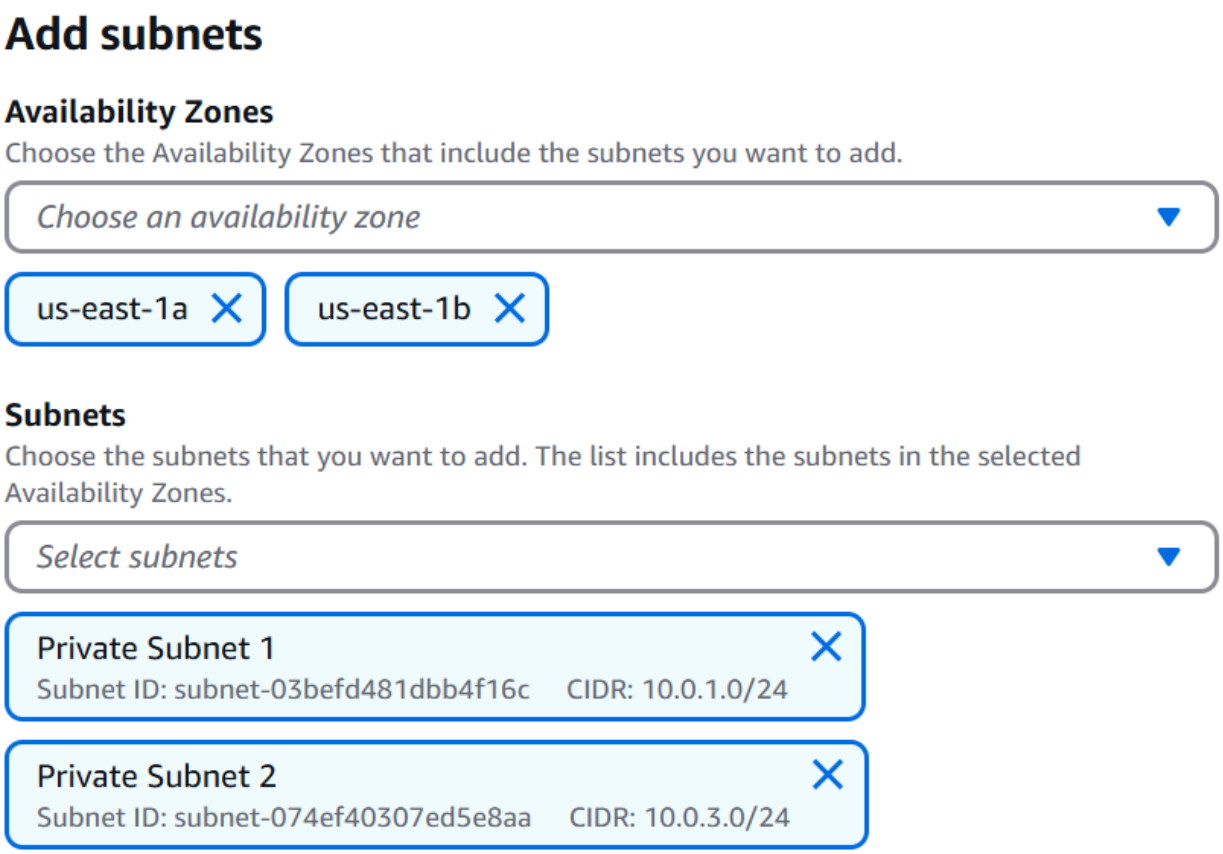
DB Subnet Group

VPC

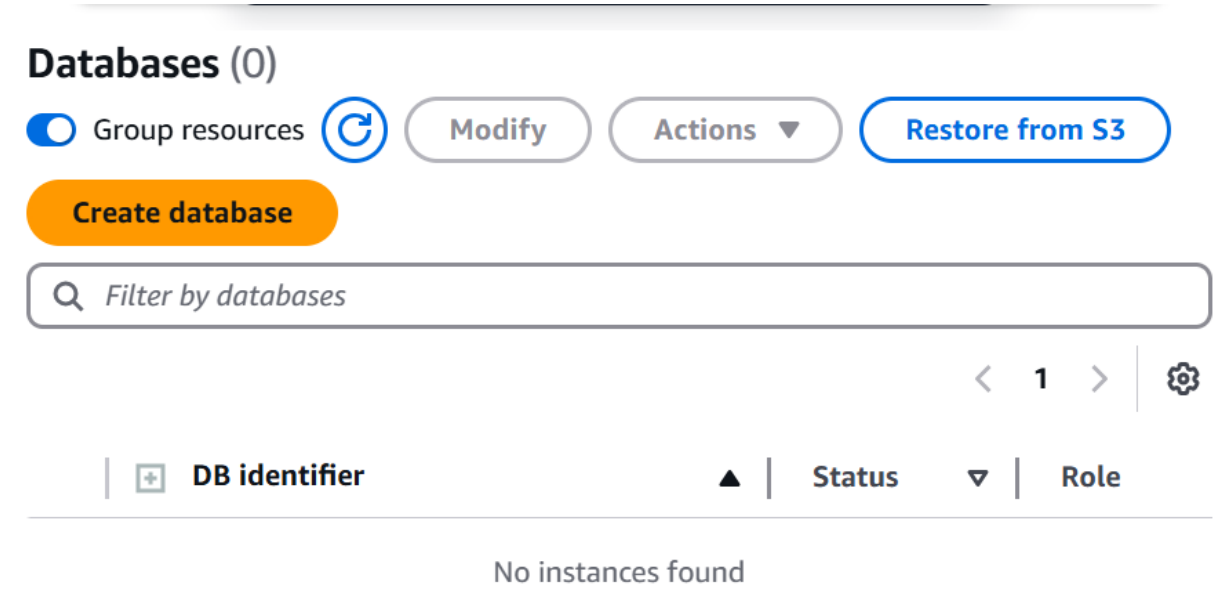
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

Lab VPC (vpc-0193fb0b3baa4abac)
4 Subnets, 2 Availability Zones

Scroll to the Add subnets part and configure the following.



Click on the Create button. Next, navigate to Databases from the left navigation pane.



Choose Create database. Configure to match the following.

Choose a database creation method

- (●) **Standard create**
  You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- ( ) **Easy create**
  Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

- ( ) Aurora (MySQL Compatible)

- ( ) Aurora (PostgreSQL Compatible)

- (●) MySQL

- ( ) PostgreSQL

Edition

(●) MySQL Community

Engine version Info
View the engine versions that support the following database features.

▼ **Hide filters**

- [ ] Show only versions that support the Multi-AZ DB cluster Info
  Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.
- [ ] Show only versions that support the Amazon RDS Optimized Writes Info
  Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine version

MySQL 8.0.40 ▼

- [ ] Enable RDS Extended Support Info
  Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for MySQL documentation.

Templates

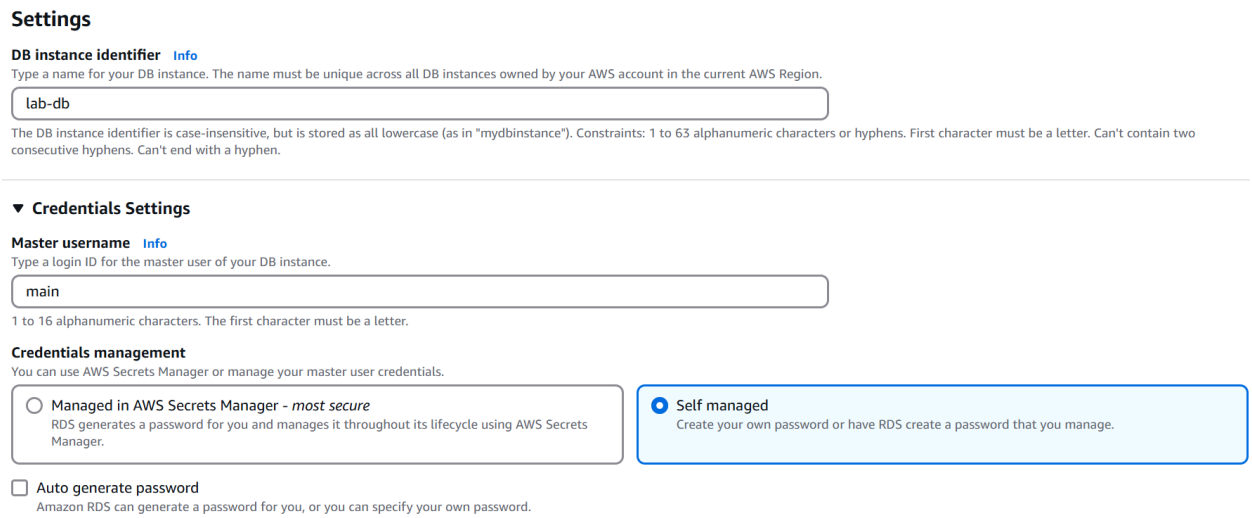
Choose a sample template to meet your use case.

- ( ) **Production**
  Use defaults for high availability and fast, consistent performance.
- (●) **Dev/Test**
  This instance is intended for development use outside of a production environment.
- ( ) **Free tier**
  Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

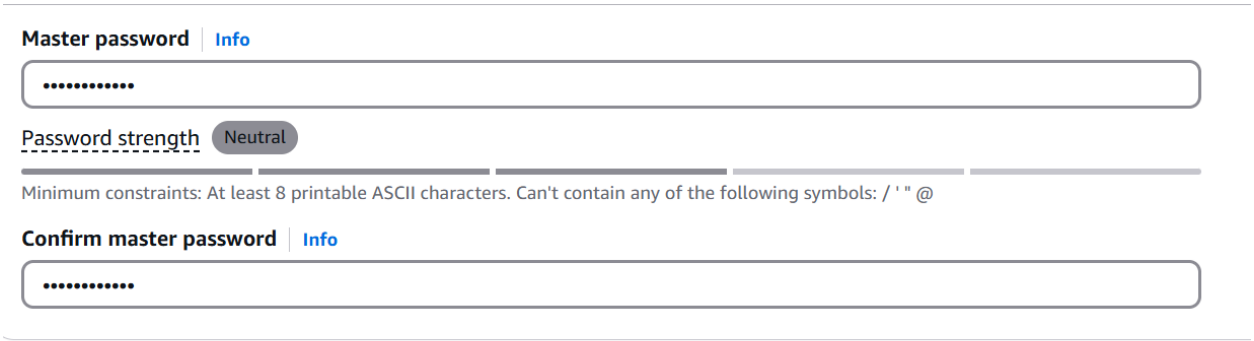
Availability and durability

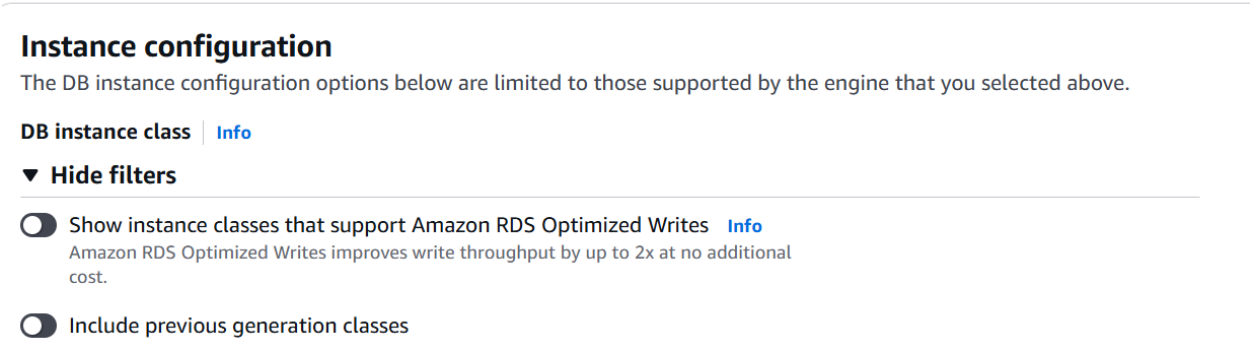
Deployment options Info
The deployment options below are limited to those supported by the engine you selected above.

- ( ) Multi-AZ DB Cluster
  Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- (●) Multi-AZ DB instance
  Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
- ( ) Single DB instance
  Creates a single DB instance with no standby DB instances.



Set the password as **lab-password**.





- ○ Standard classes (includes m classes)
- ○ Memory optimized classes (includes r and x classes)
- ◉ Burstable classes (includes t classes)

db.t3.micro
2 vCPUs 1 GiB RAM Network: Up to 2,085 Mbps

Storage

Storage type Info
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp3)
Performance scales independently from storage

Allocated storage Info

20 GiB

Minimum: 20 GiB. Maximum: 6,144 GiB

Connectivity Info

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

- ◉ Don't connect to an EC2 compute resource
  Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.
- ○ Connect to an EC2 compute resource
  Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Lab VPC (vpc-0193fb0b3baa4abac)
4 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

db-subnet-group
2 Subnets, 2 Availability Zones

Public access Info

- ○ Yes
  RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.
- ◉ No
  RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

- ◉ Choose existing
  Choose existing VPC security groups
- ○ Create new
  Create new VPC security group

Existing VPC security groups

Choose one or more options

DB Security Group ✕

RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

☐ Create an RDS Proxy Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing.

Certificate authority - *optional* Info

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expiry: May 25, 2061

If you don't select a certificate authority, RDS chooses one for you.

▶ **Additional configuration**

Tags - *optional*

A tag consists of a case-sensitive key-value pair.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Database authentication

Database authentication options Info

- (●) Password authentication
  Authenticates using database passwords.
- ( ) Password and IAM database authentication
  Authenticates using the database password and user credentials through AWS IAM users and roles.
- ( ) Password and Kerberos authentication
  Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Monitoring

- [ ] Enable Enhanced Monitoring
  Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

▼ Additional configuration

Database options, encryption turned off, backup turned off, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name Info

lab

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group Info

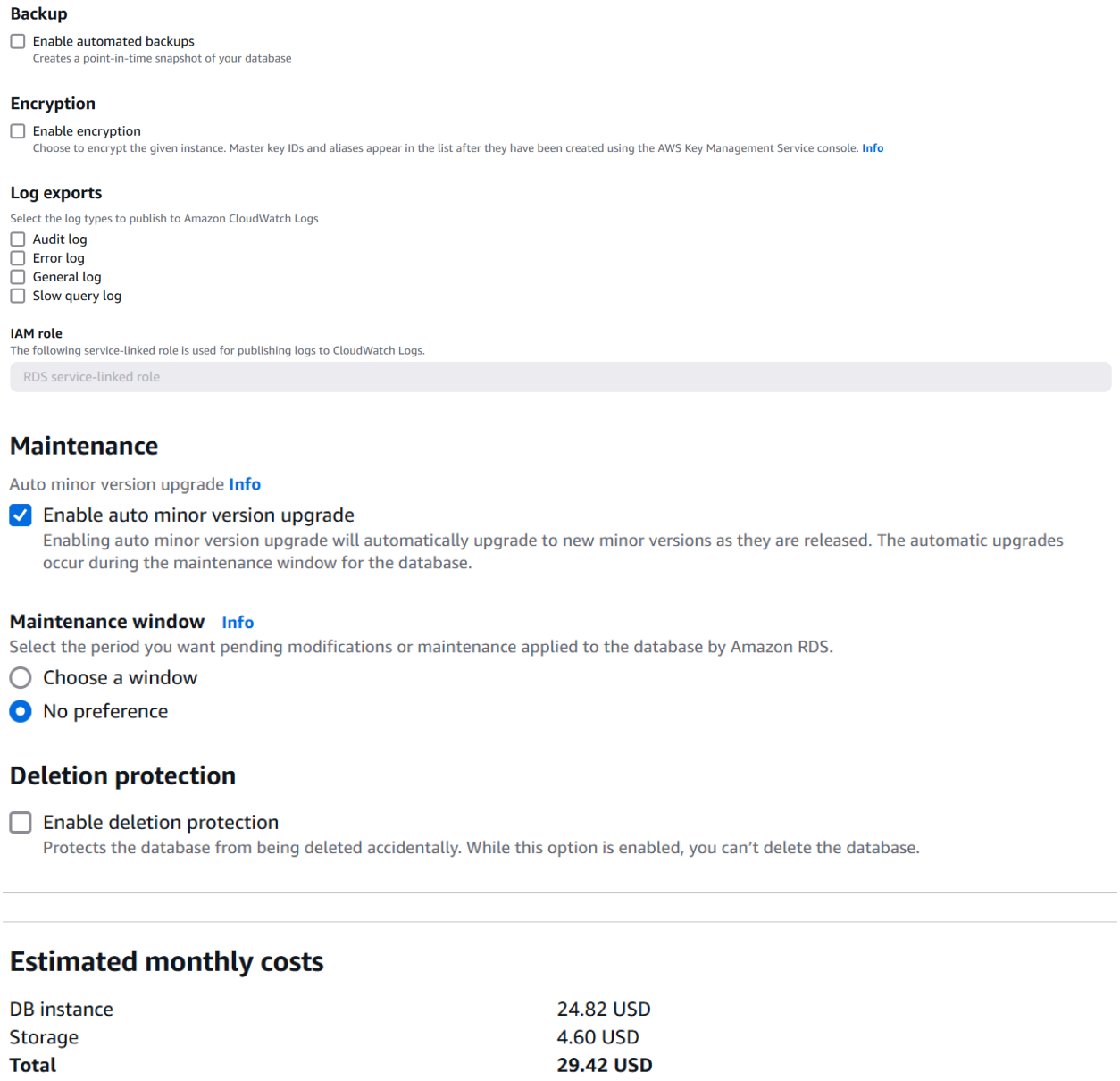
default.mysql8.0

Option group Info

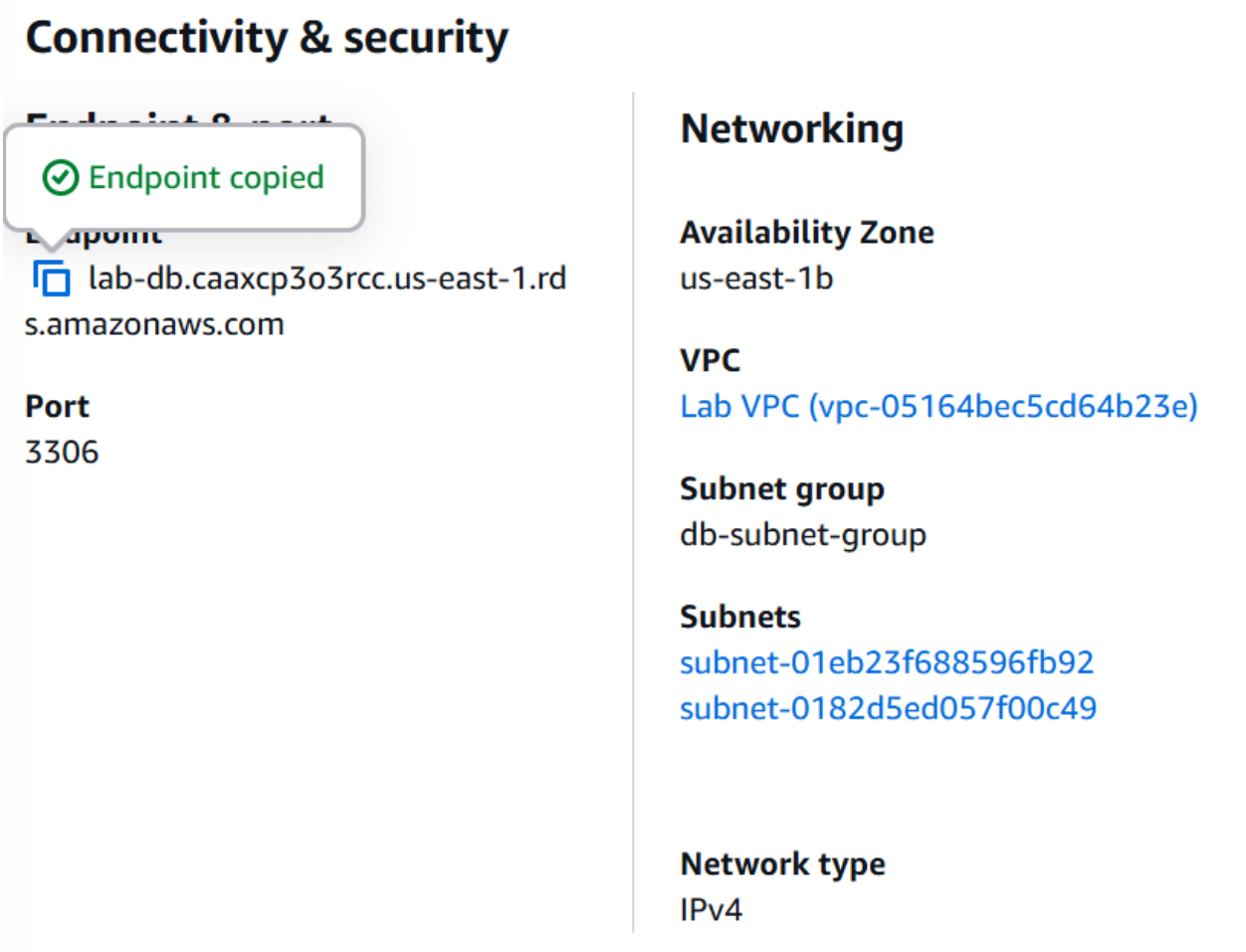
default:mysql-8-0

Backup

- [ ] Enable automated backups
  Creates a point-in-time snapshot of your database

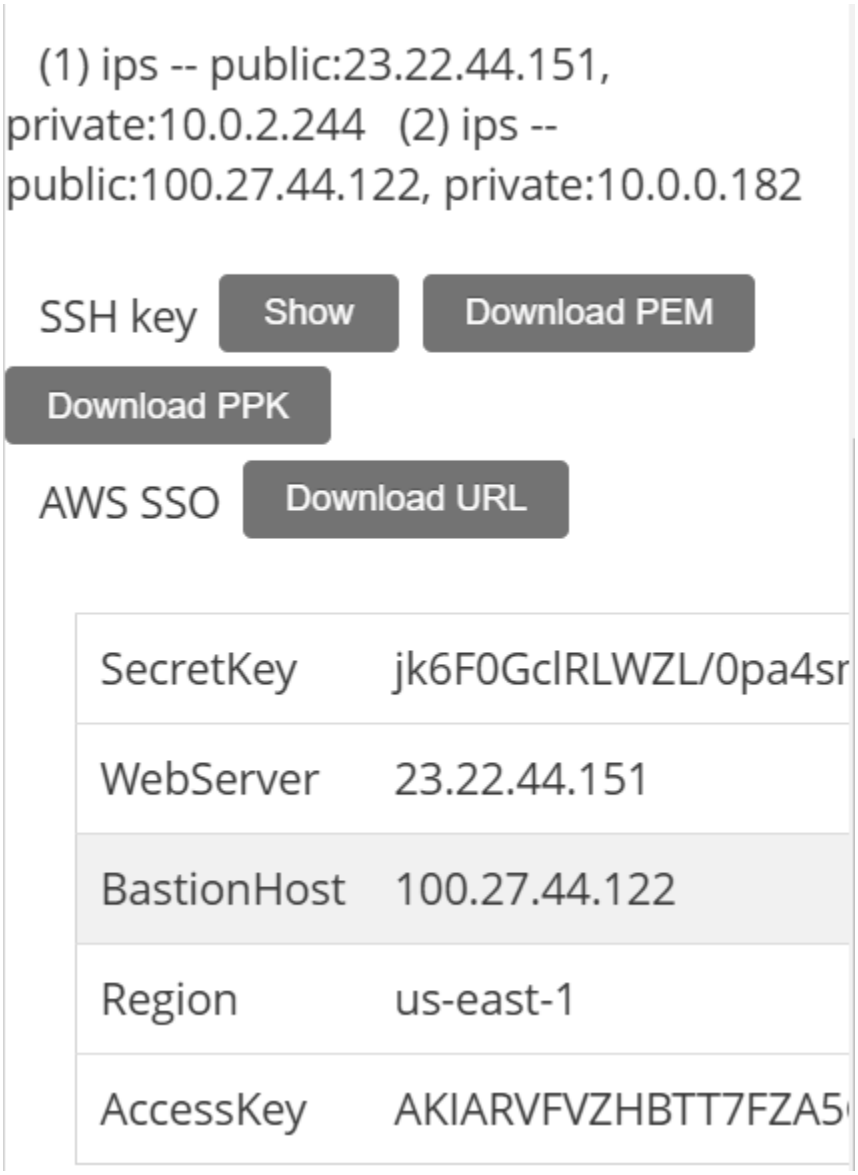


After the database is available, click on lab-db and make sure the status is Modifying or Available. Navigate to the Connectivity & security section and copy the Endpoint field into a document.

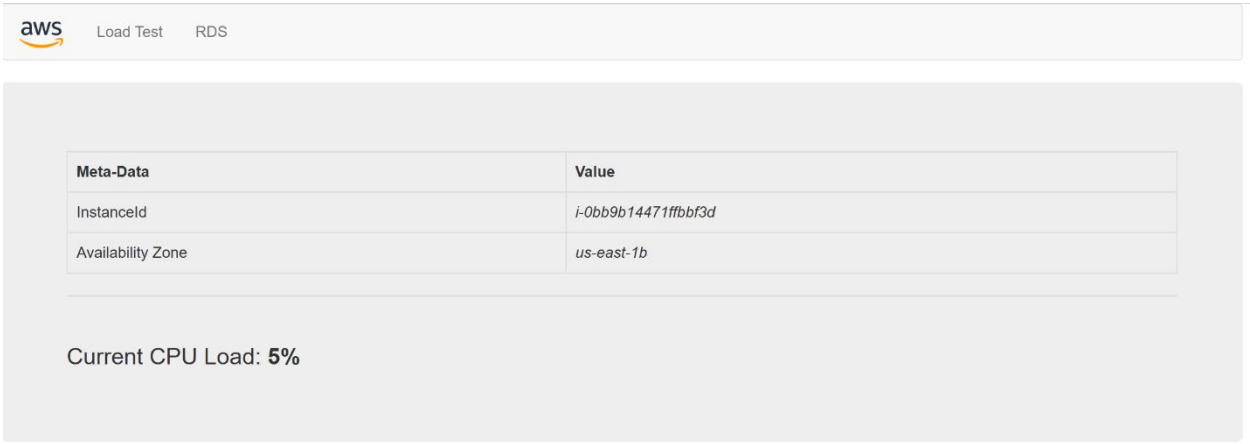


The following is my endpoint: lab-db.caaxcp3o3rcc.us-east-1.rds.amazonaws.com.

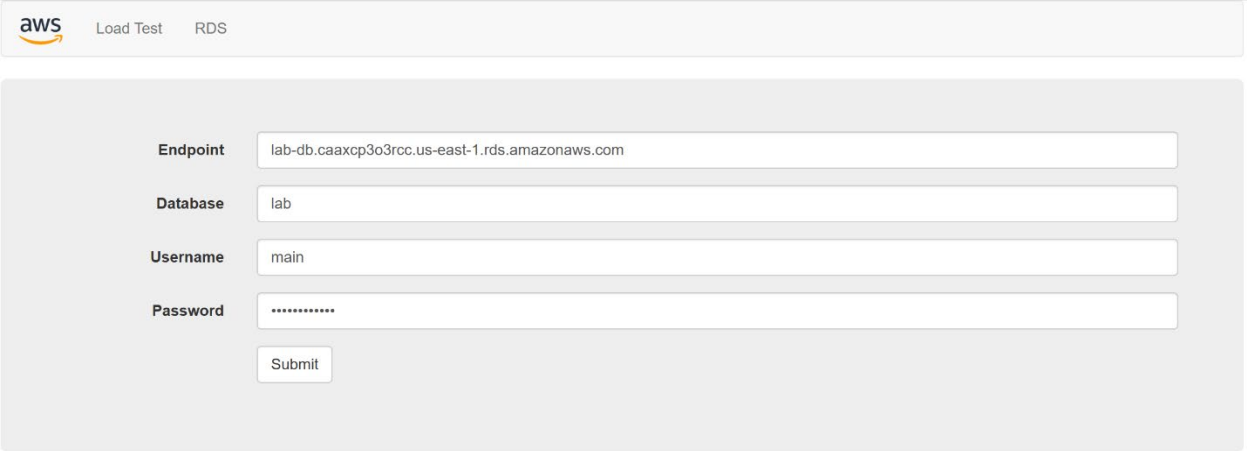
Next, to discover the WebServer IP address, click on the AWS Details drop-down menu above the lab instructions (not the AWS Console) and find the WebServer IP address. Copy IPv4 address into a new web browser. In this case, the IPv4 address is 23.22.44.151.



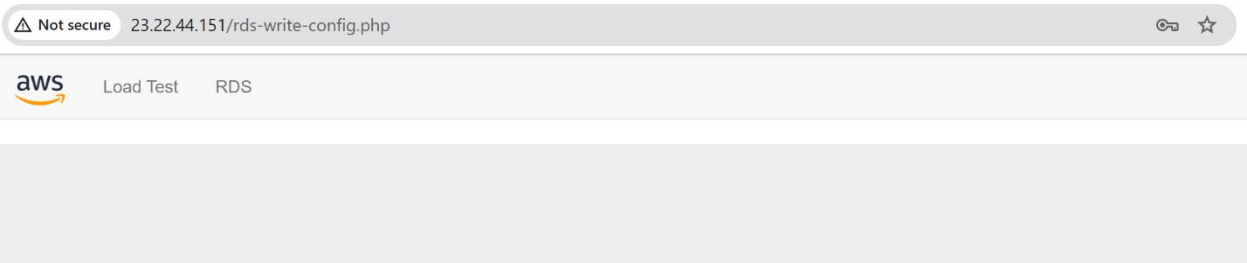
This should bring you to the page below.



Click on the RDS link on the top of the page and configure as shown below, where the endpoint is copied from before and the password is **lab-password**:



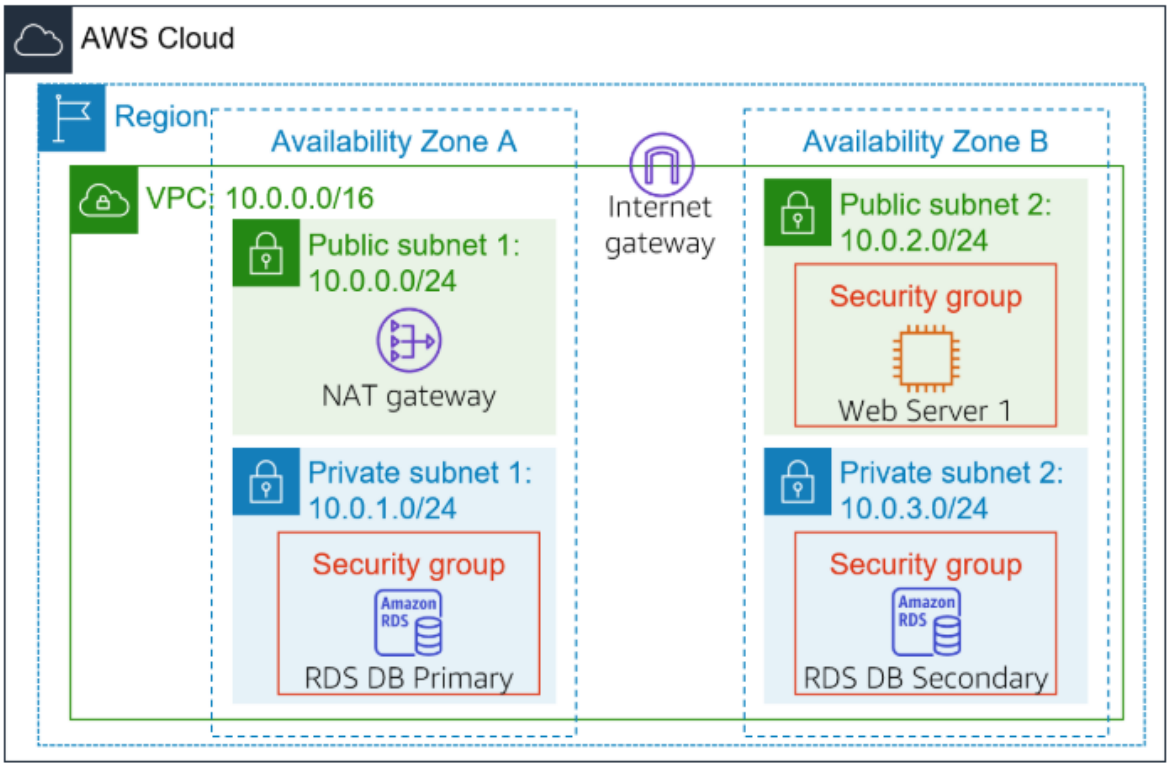
However, we find that, when the above is submitted per the directions, the web page does not display anything.



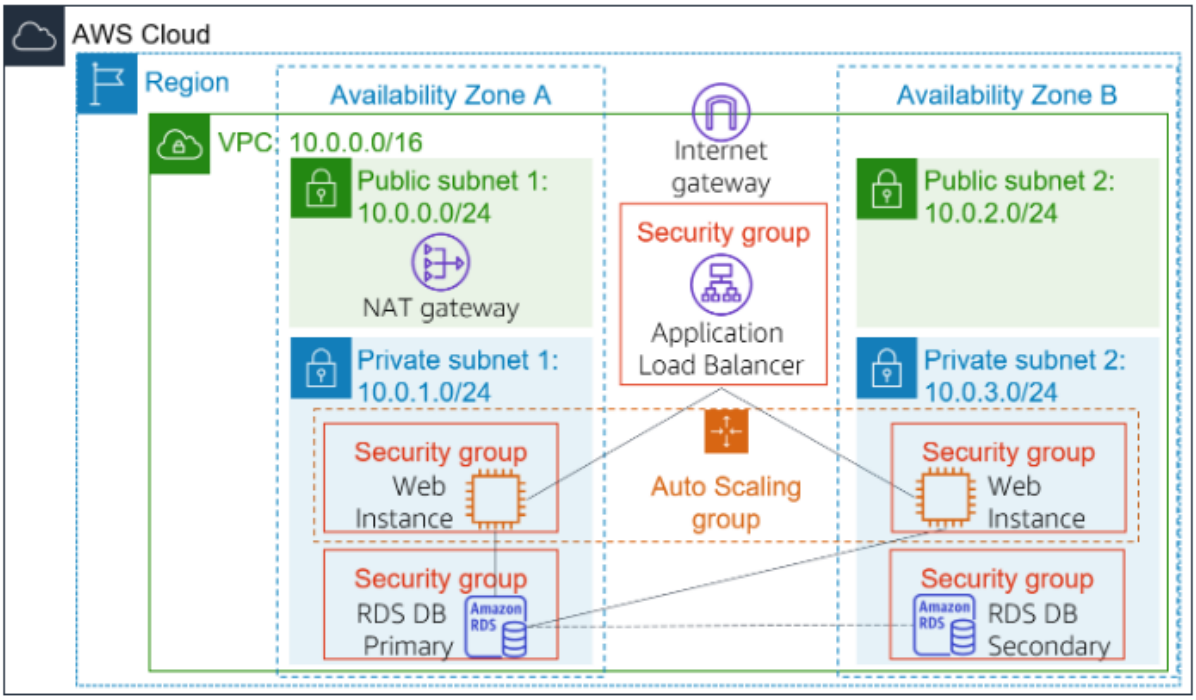
Therefore, we are unable to configure the web application to use the database.

LAB 6

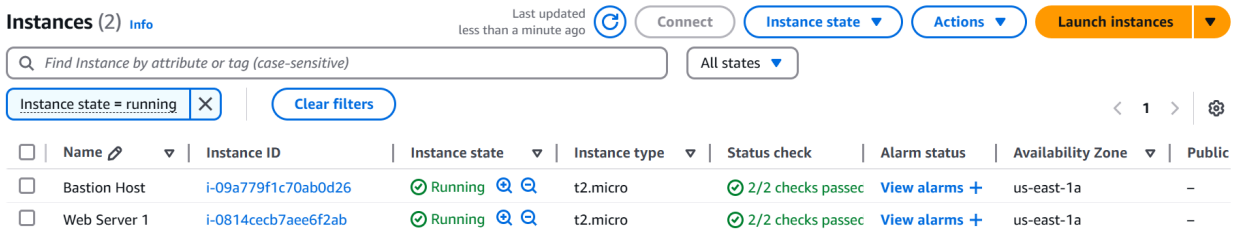
Start with this:



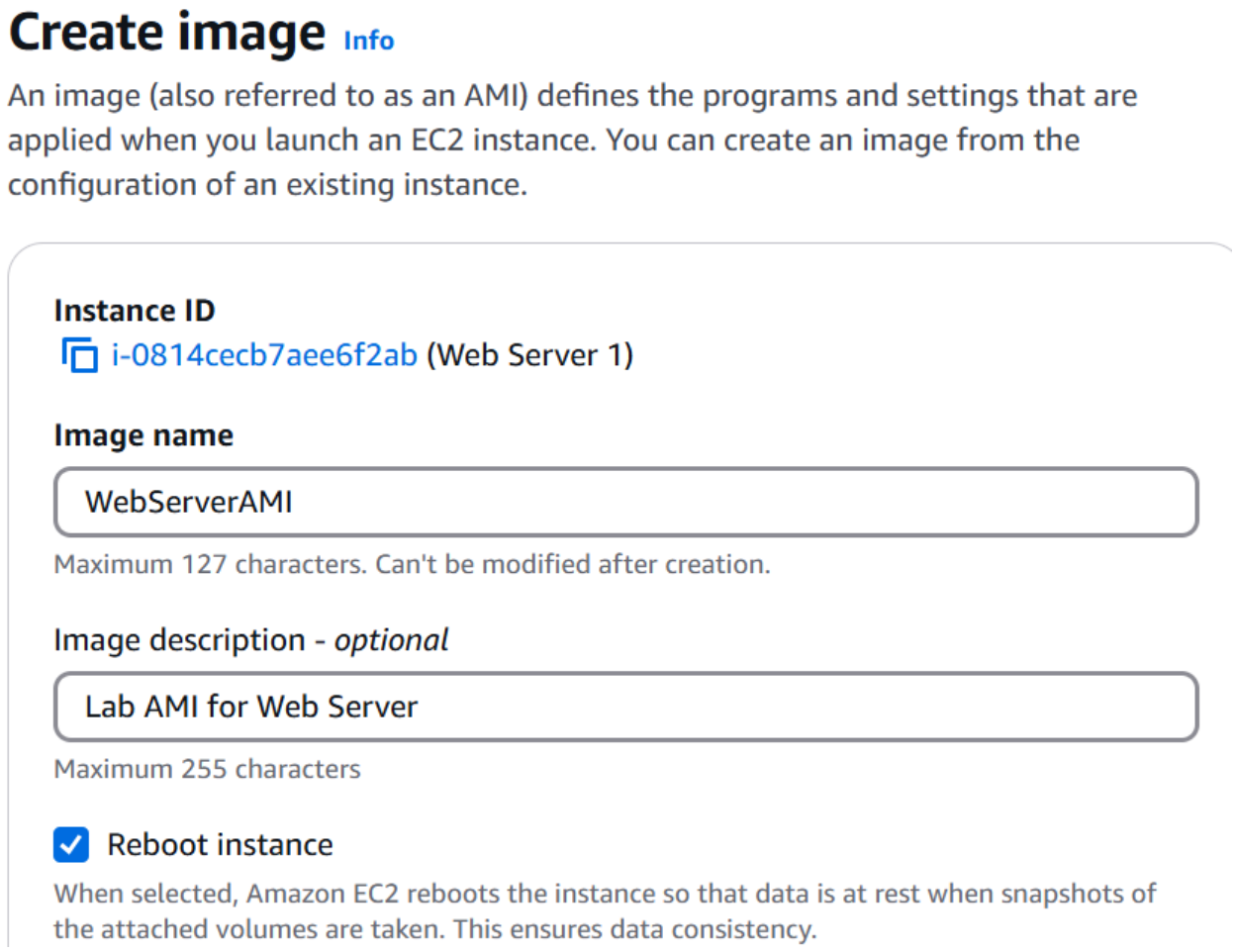
Finish with this:



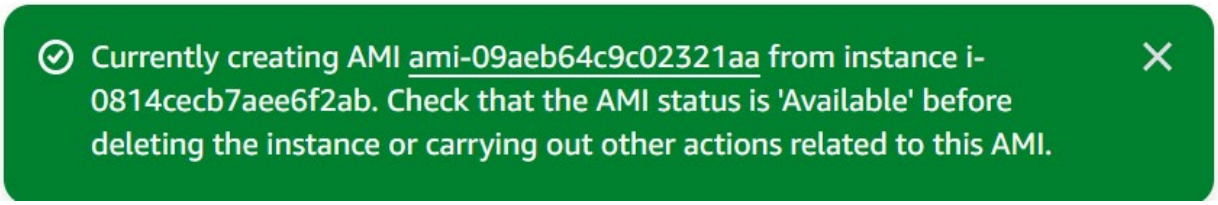
Search in the search box for EC2 to open it. Choose instances in the left navigation pane. Make sure the Web Server displays 2/2 checks passed before continuing, as shown below:



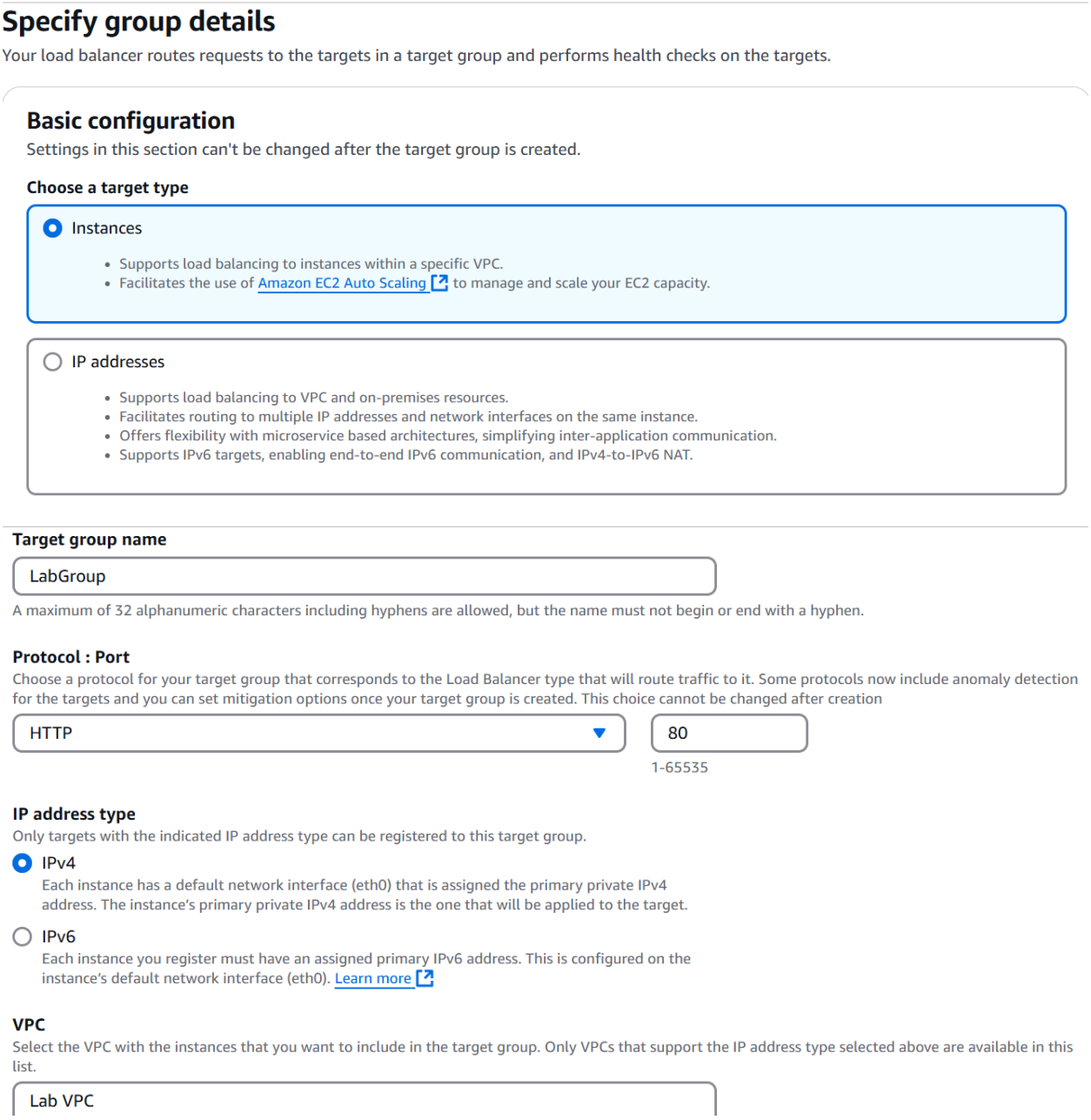
Check the box next to Web Server 1 and in the Actions menu, choose Image and templates and select Create image. Configure as shown in the following image.



There should be a success message with the AMI ID for the new AMI: ami-09aeb64c9c02321aa.

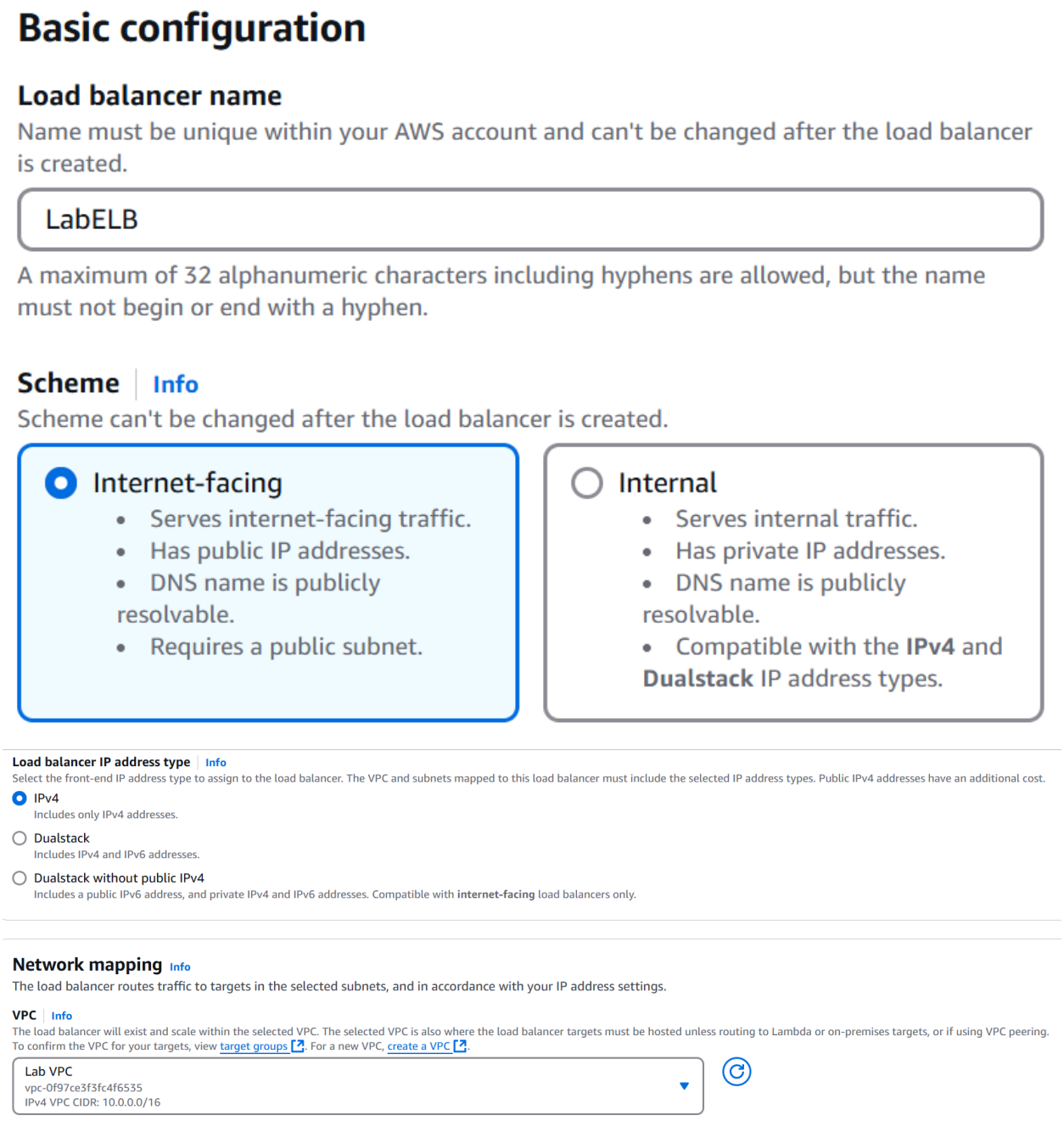


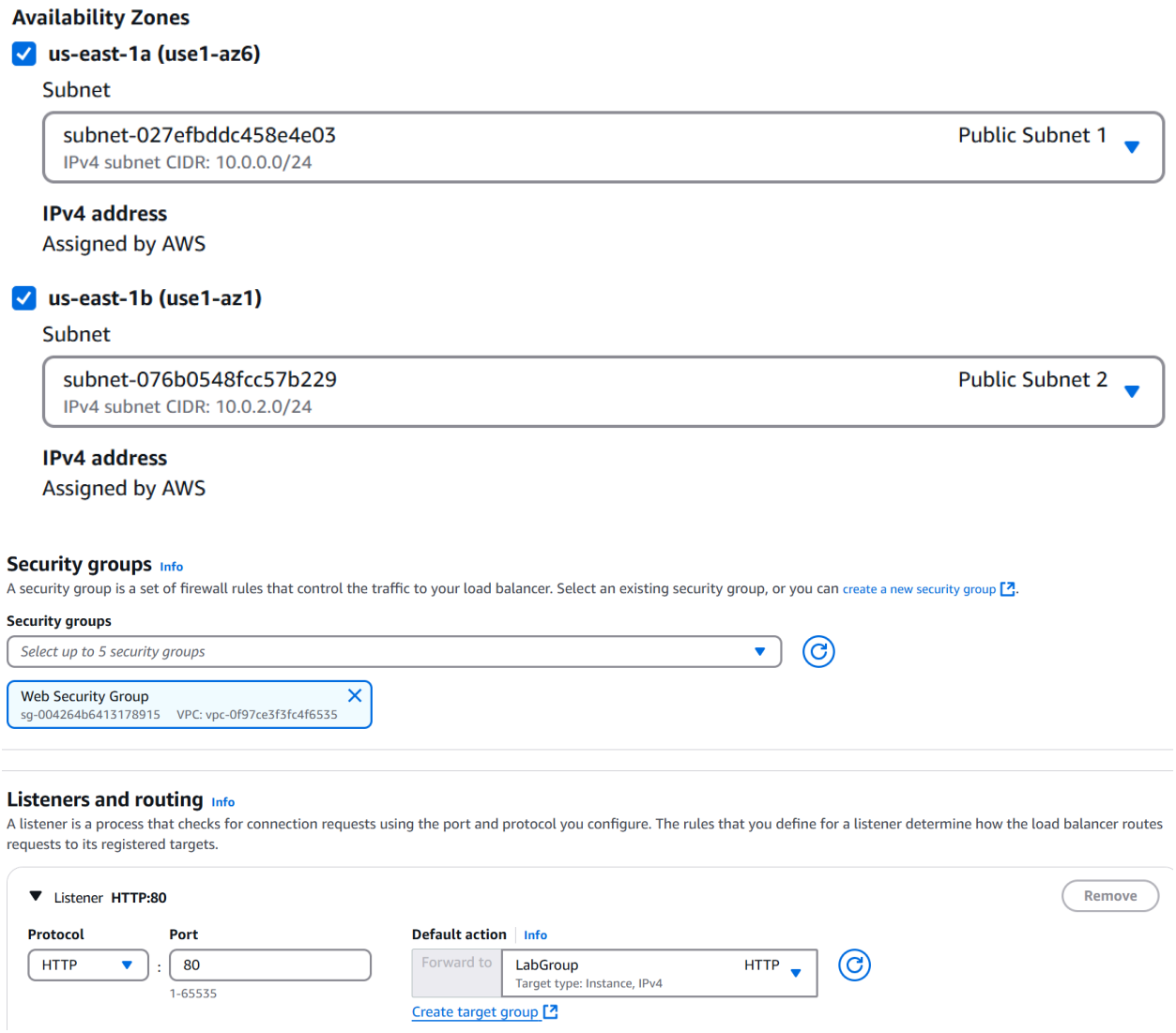
In the left navigation pane, select Target Groups, which define where to send traffic coming into the Load Balancer. Click on Create target group and configure similar to the following.



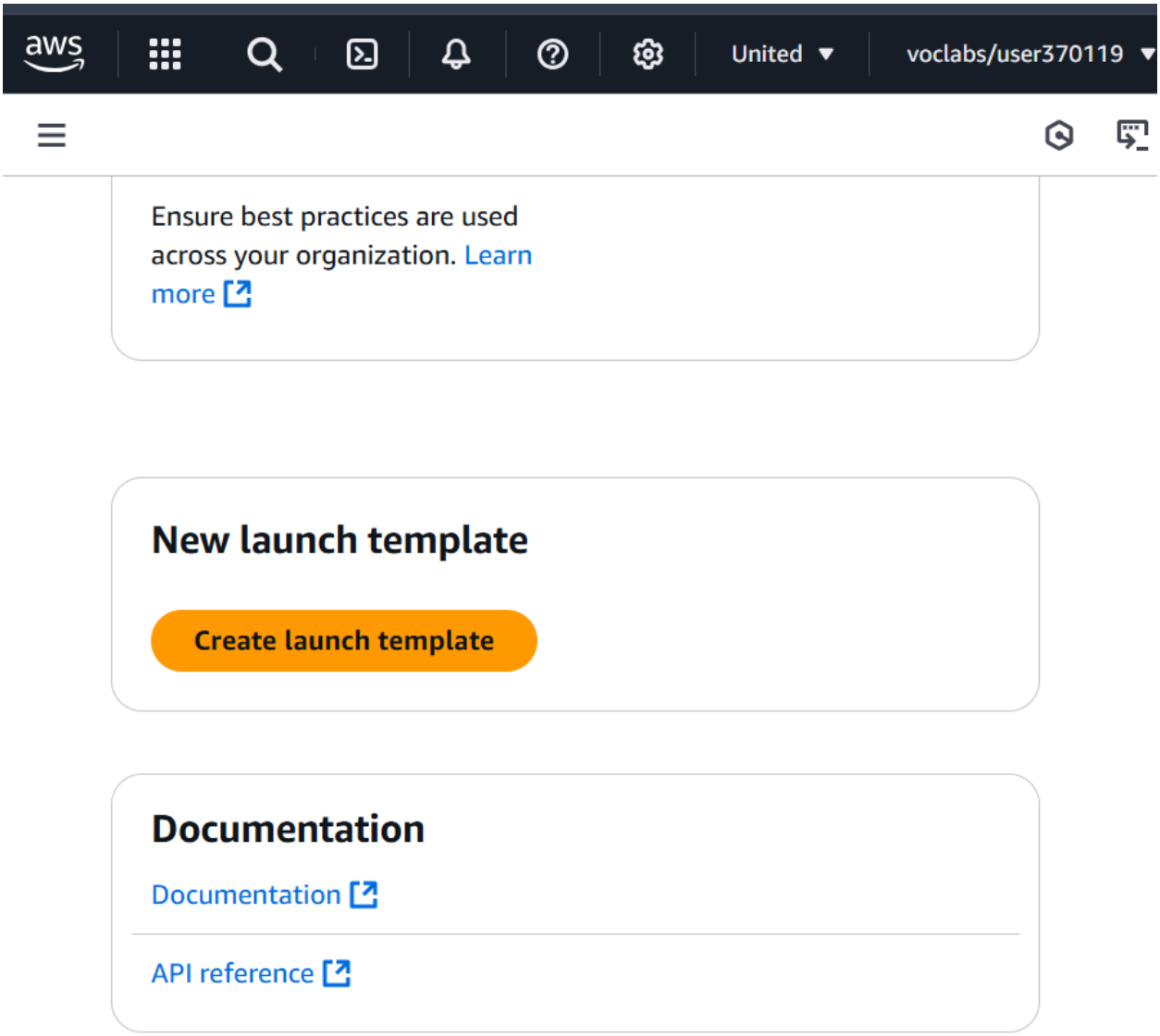
Click next after these configurations have been completed; do not change any other settings.

Choose Create target group at the bottom of the page; no other changes are needed. In the left navigation pane, choose Load Balancers and click on Create load balancer. Create one under Application Load Balancer and configure the following.





Then, click Create load balancer. In the left navigation pane, select Launch Templates and scroll until this page is seen:



Click on Create launch template and configure as shown below.

Launch template name and description

Launch template name - *required*

LabConfig

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

A prod webserver for MyApp

Max 255 chars

Auto Scaling guidance | Info

Select this if you intend to use this template with EC2 Auto Scaling

☑ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ **Template tags**

▶ **Source template**

▼ **Application and OS Images (Amazon Machine Image) - required** Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | My AMIs | Quick Start

◉ Owned by me ○ Shared with me

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

WebServerAMI
ami-09aeb64c9c02321aa
2025-02-02T23:19:13.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

Description
Lab AMI for Web Server

Architecture	AMI ID
x86_64	ami-09aeb64c9c02321aa

Instance type Info | Get advice

Advanced

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

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

vockey

Create new key pair

Network settings Info

Subnet Info

Don't include in launch template

Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

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.

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

Security groups | Info

Select security groups

Web Security Group sg-004264b6413178915
VPC: vpc-0f97ce3f3fc4f6535

Compare security group rules

▶ **Advanced network configuration**

▼ Storage (volumes) Info

EBS Volumes Hide details

▶ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3), 3000 IOPS)
AMI Volumes are not included in the template unless modified

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

▼ Resource tags Info

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

Add new tag

You can add up to 50 more tags.

▼ Advanced details Info

IAM instance profile | Info

Don't include in launch template

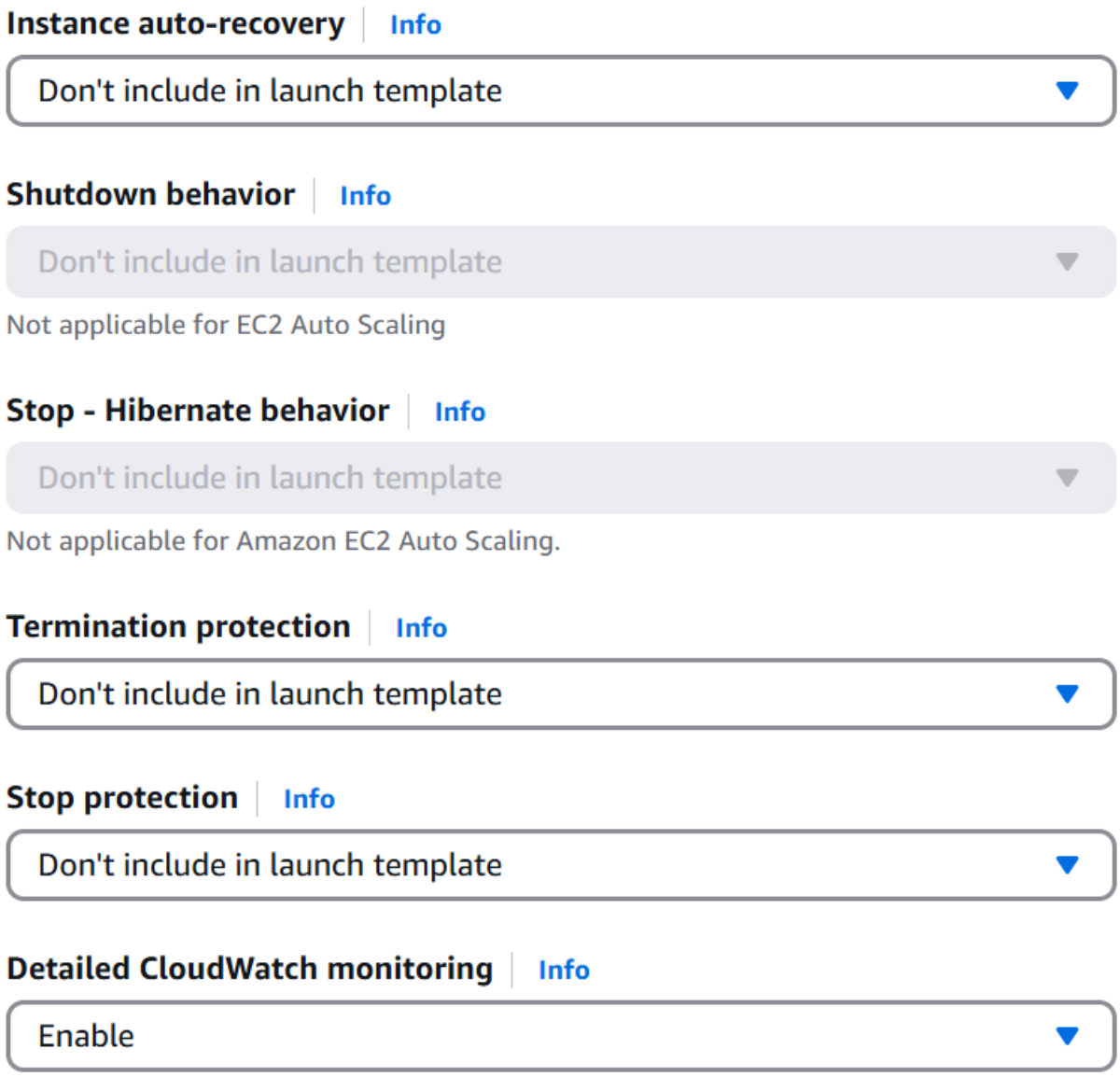
Create new IAM profile

Hostname type | Info

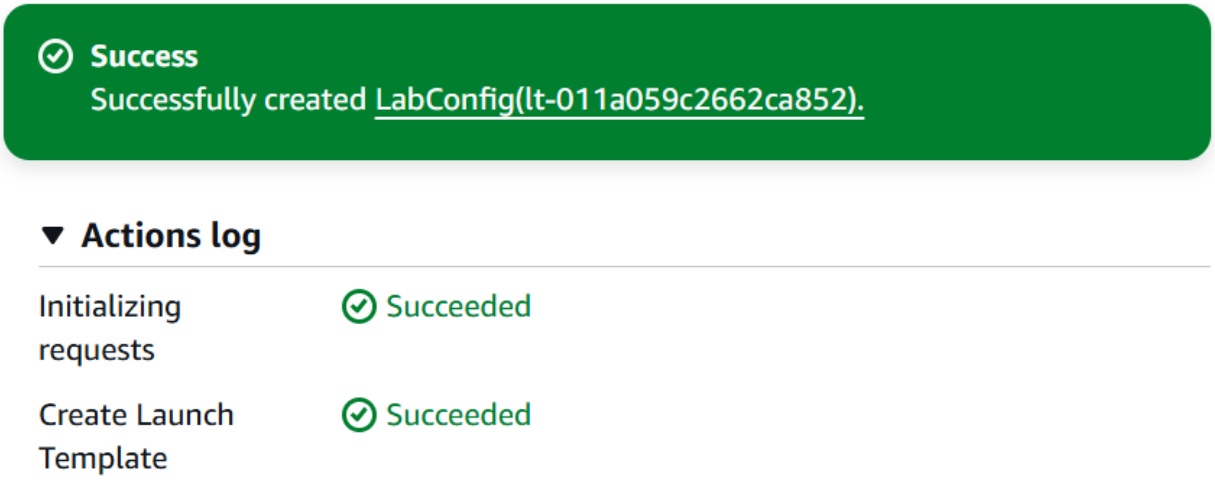
Don't include in launch template

DNS Hostname | Info

- [ ] Enable resource-based IPv4 (A record) DNS requests
- [ ] Enable resource-based IPv6 (AAAA record) DNS requests



There are no more configuration changes that need to be made. Click on Create launch template.



In the success message, click on the LabConfig launch template link. From the Actions drop-down menu, choose Create Auto Scaling group and configure as shown below; only change what is displayed in the images below.

Name

Auto Scaling group name

Enter a name to identify the group.

Lab Auto Scaling Group

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

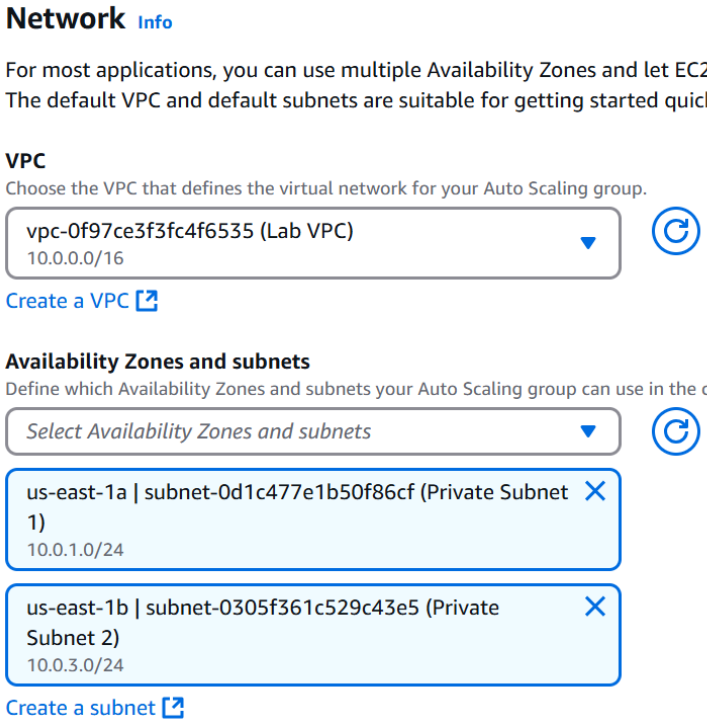
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template

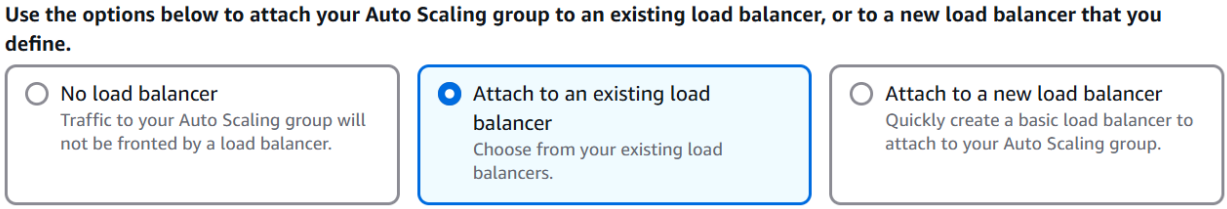
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

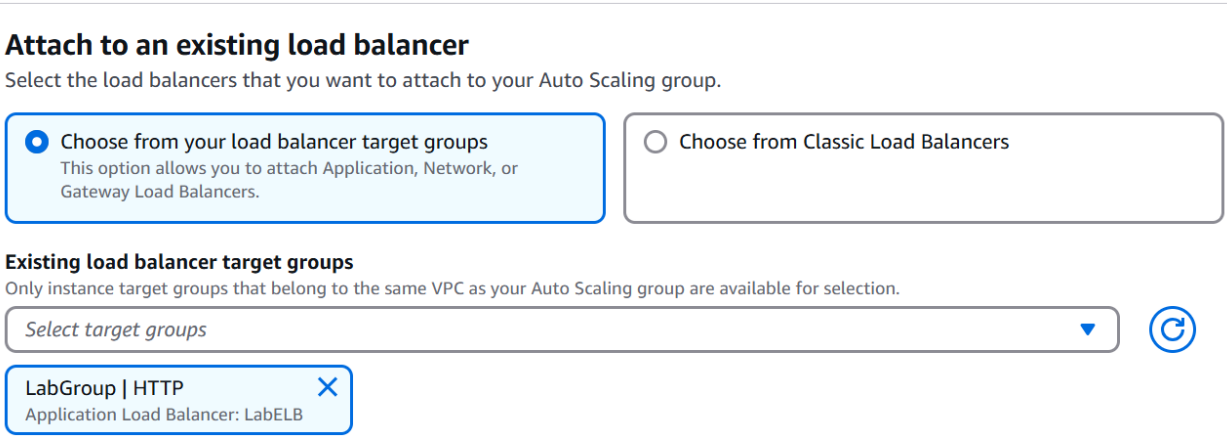
LabConfig

Click next to move onto the next step.



Click next to move onto the next step.





Click next to move onto the next step.

Configure group size and scaling - *optional* Info

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity

Specify your group size.

2

Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

2

Equal or less than desired capacity

Max desired capacity

6

Equal or greater than desired capacity

Automatic scaling - *optional*

Choose whether to use a target tracking policy Info

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

- ○ No scaling policies
  Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.
- ◉ Target tracking scaling policy
  Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name

LabScalingPolicy

Metric type Info

Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

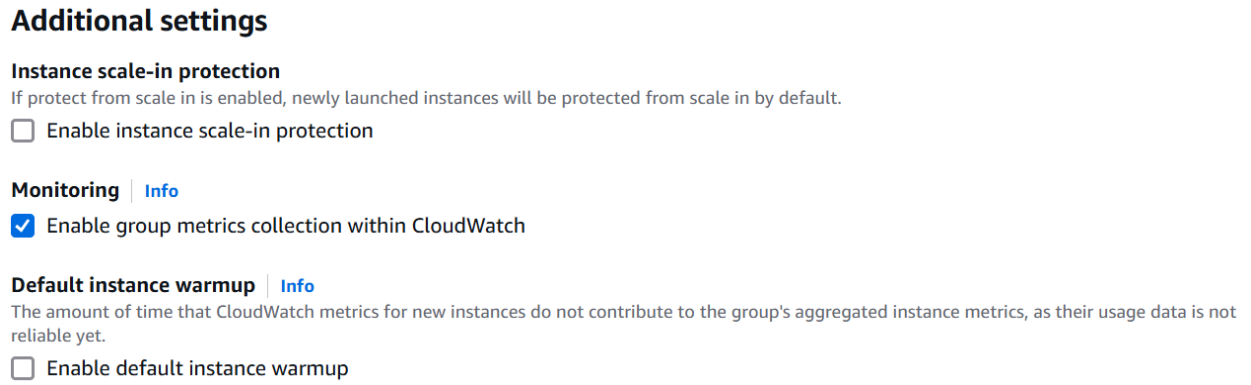
Target value

60

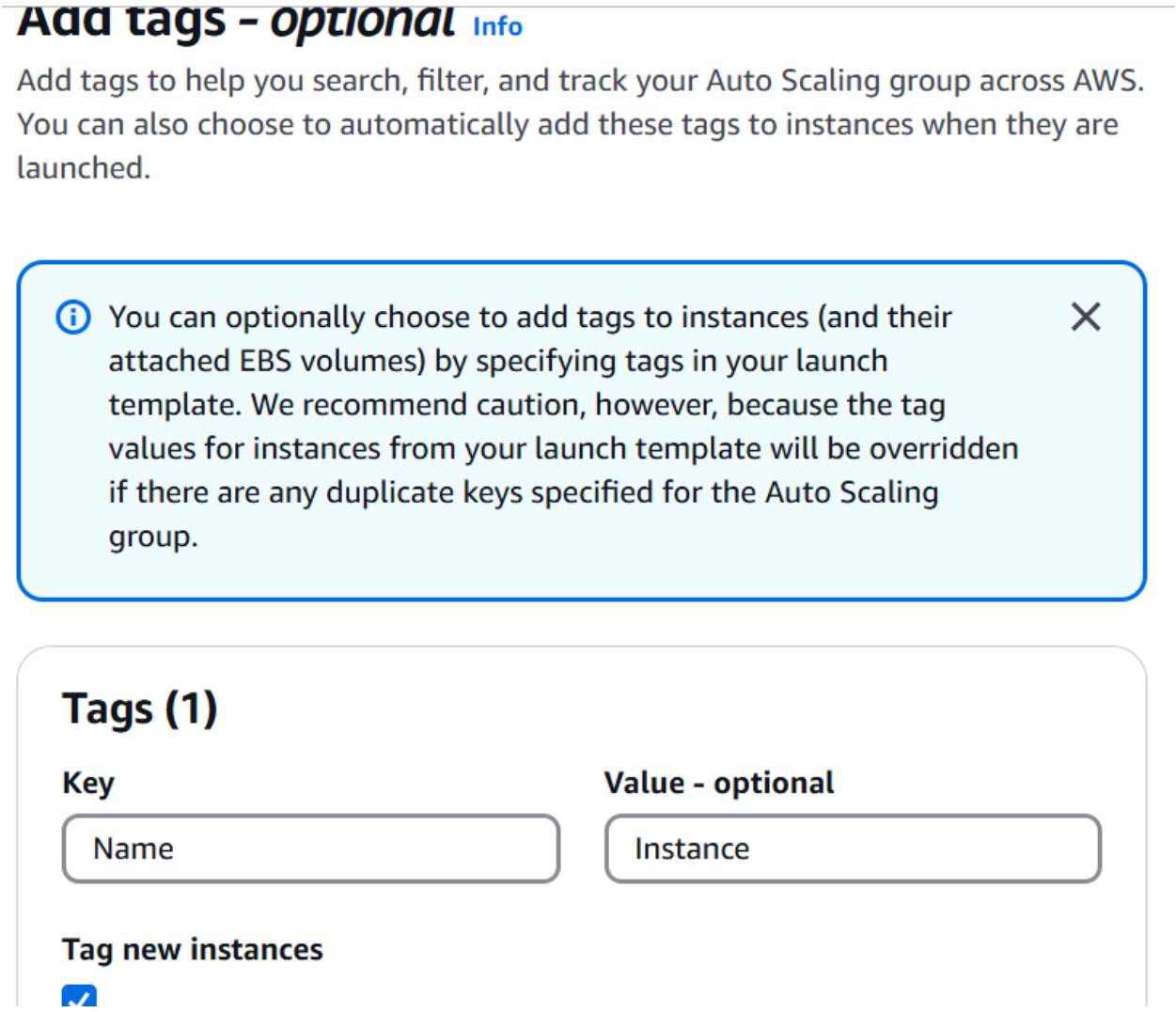
Instance warmup Info

300 seconds

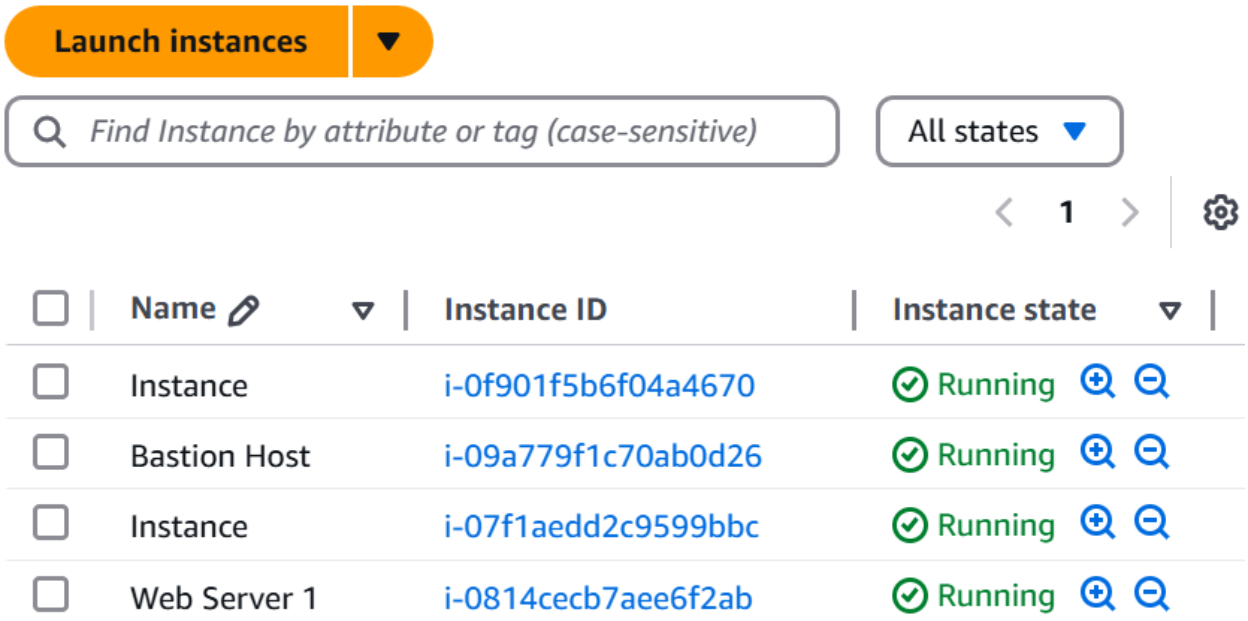
☐ Disable scale in to create only a scale-out policy



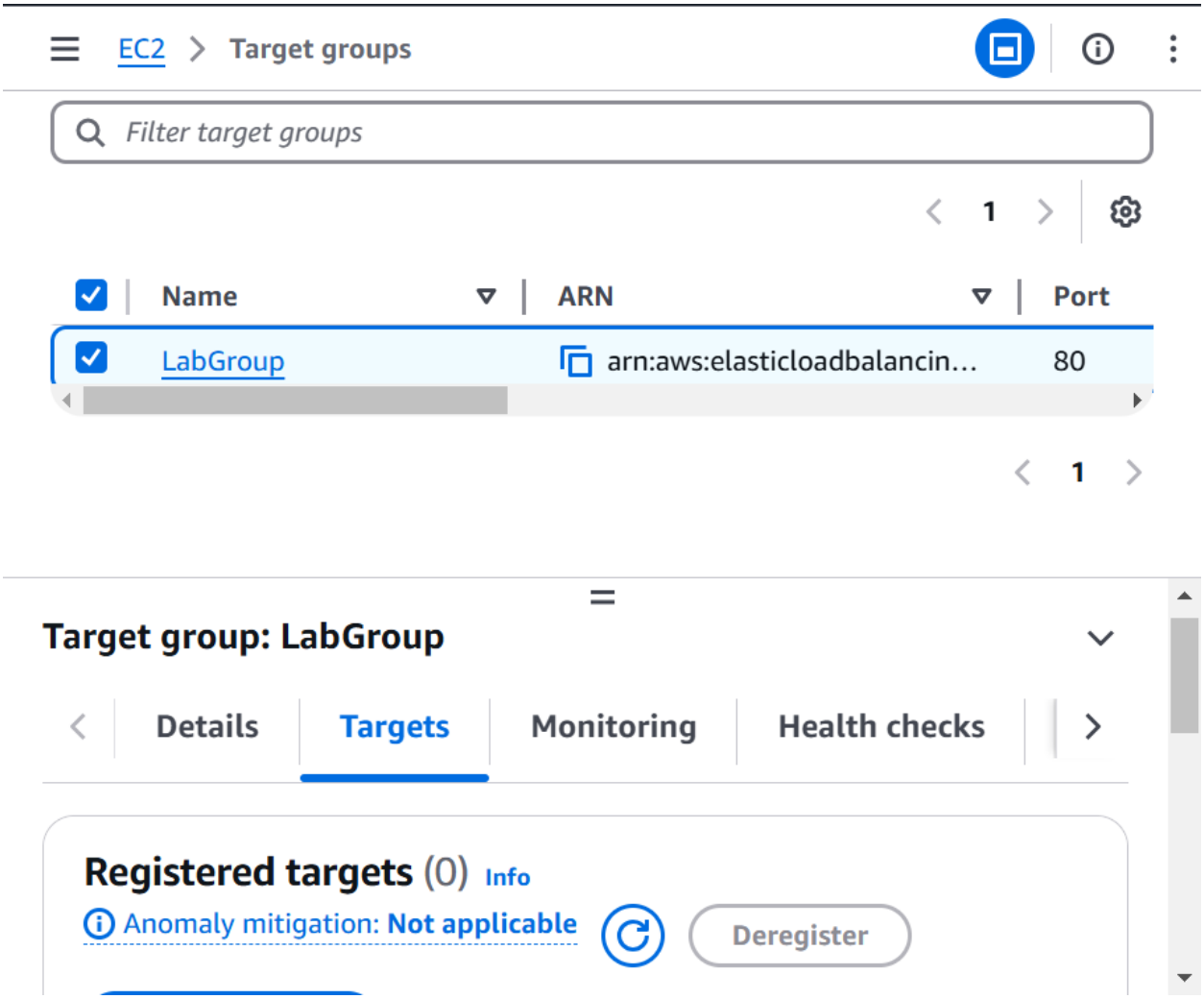
Click next to move onto the next step. In Step 5, no action is needed, so click next.



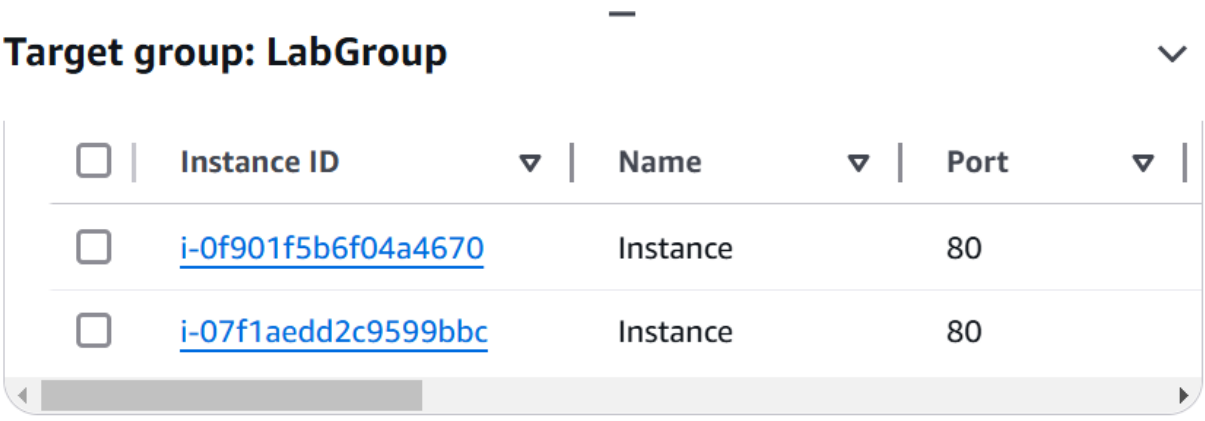
Click on Create Auto Scaling group. In the left navigation pane, go back to Instances. There should be two new instances; refreshing may be necessary.



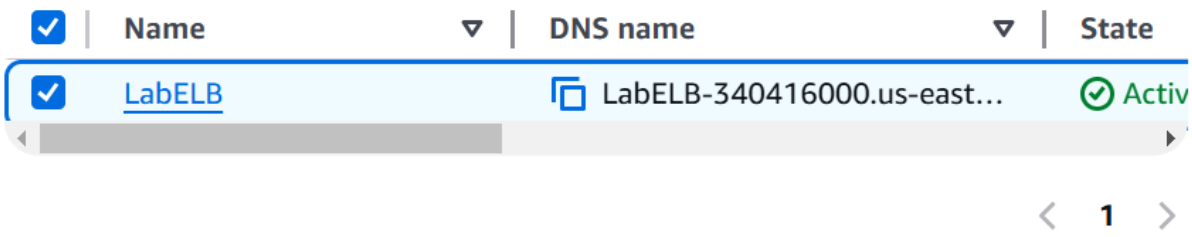
In the left navigation pane, navigate to Target Groups and check the box next to LabGroup. Choose the Targets tab, as shown below.

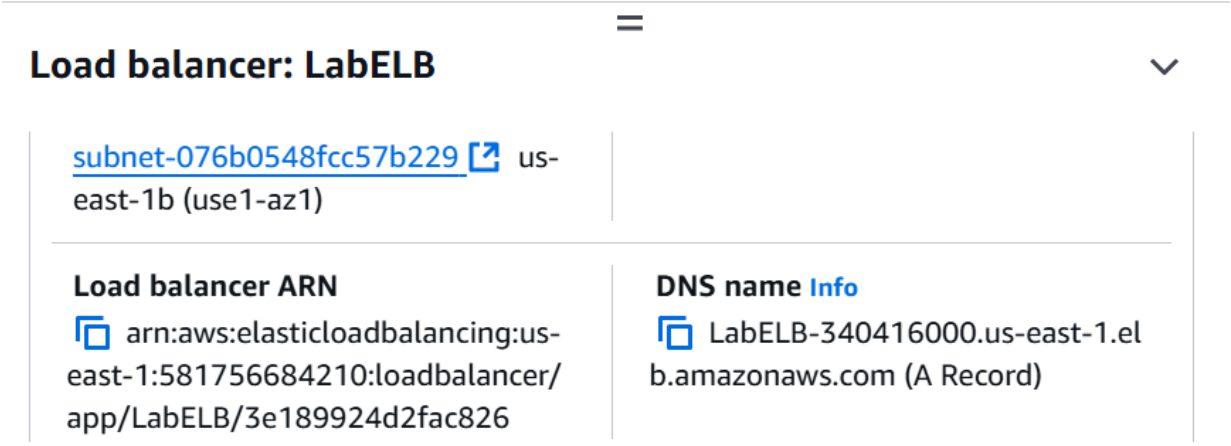


Refresh so that there are two Registered targets that appear under this tab, both with a status of Healthy.

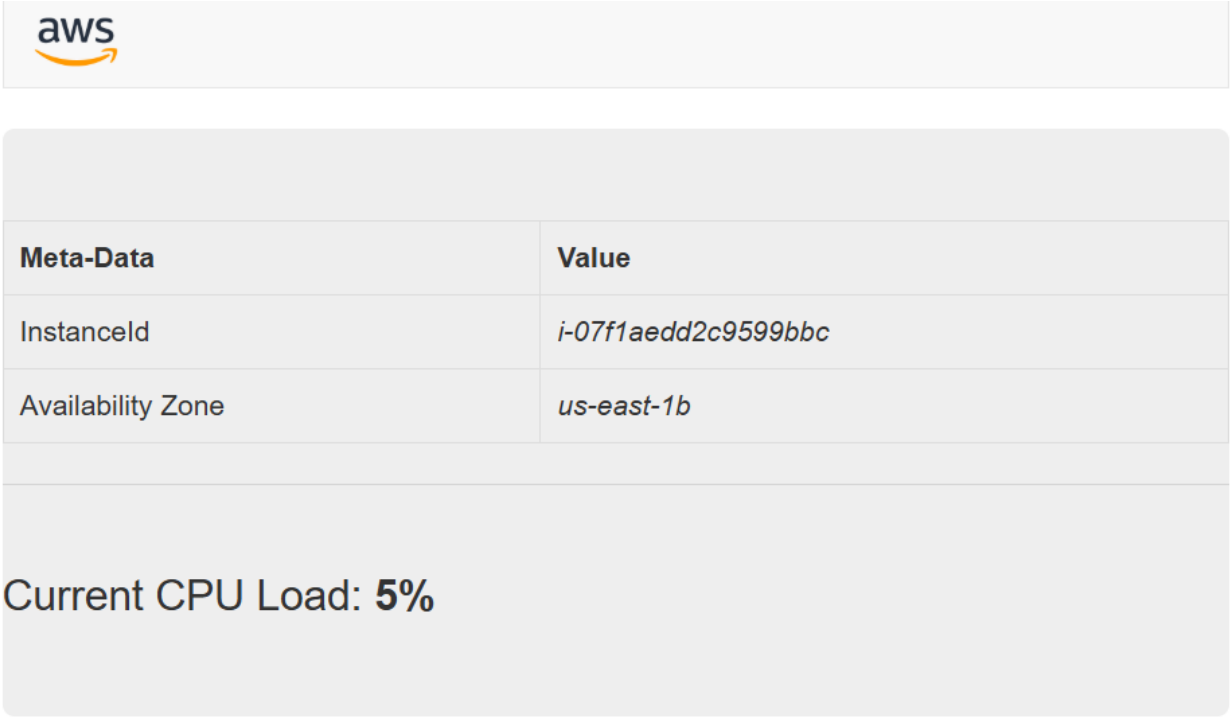


In the left navigation pane, select Load balancers and check the box next to LabELB. Under the details tab, find and copy the DNS name.

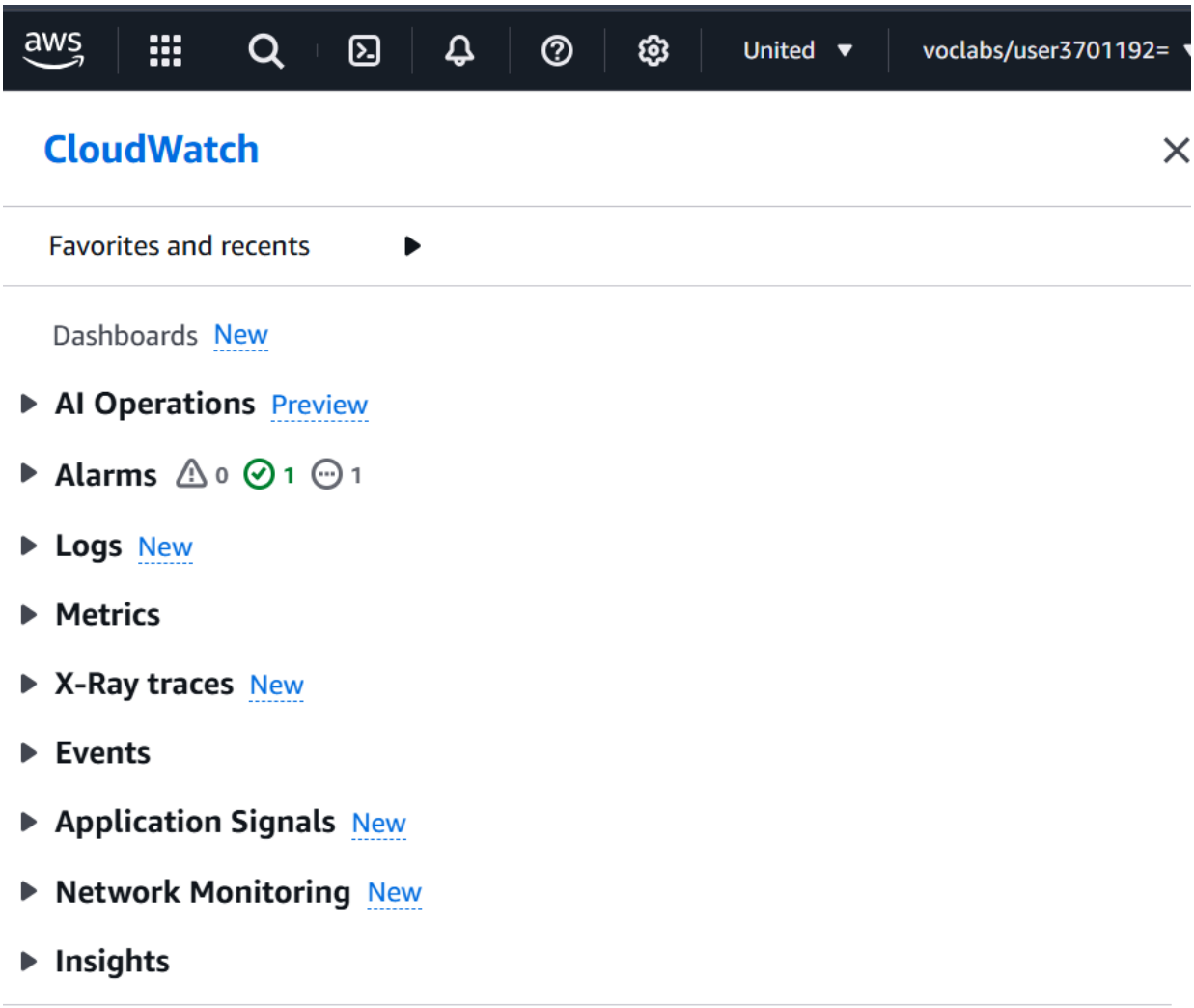




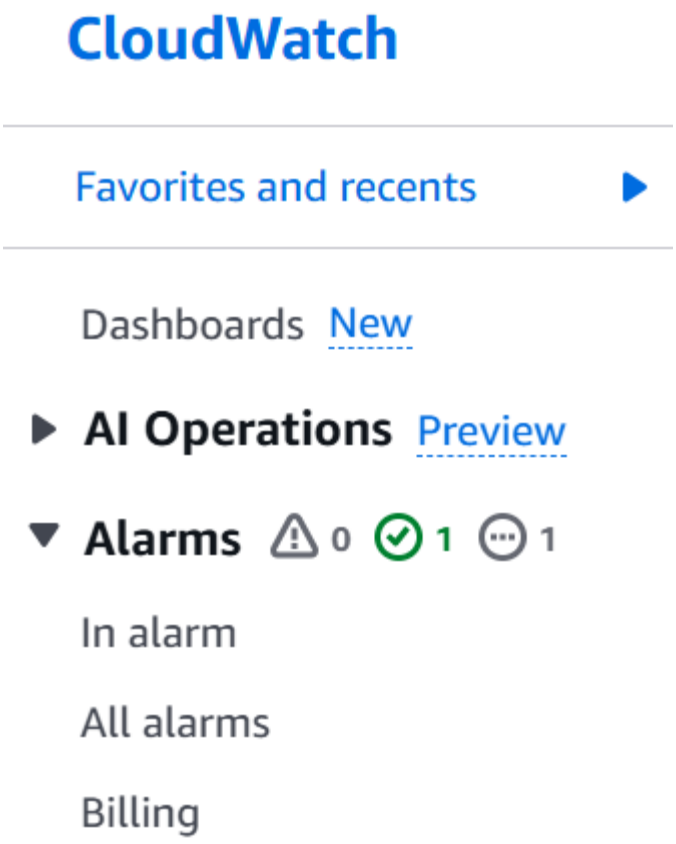
Paste the DNS name into a new web browser; the application should appear, indicating that the Load Balancer received and sent the request before passing back the result.



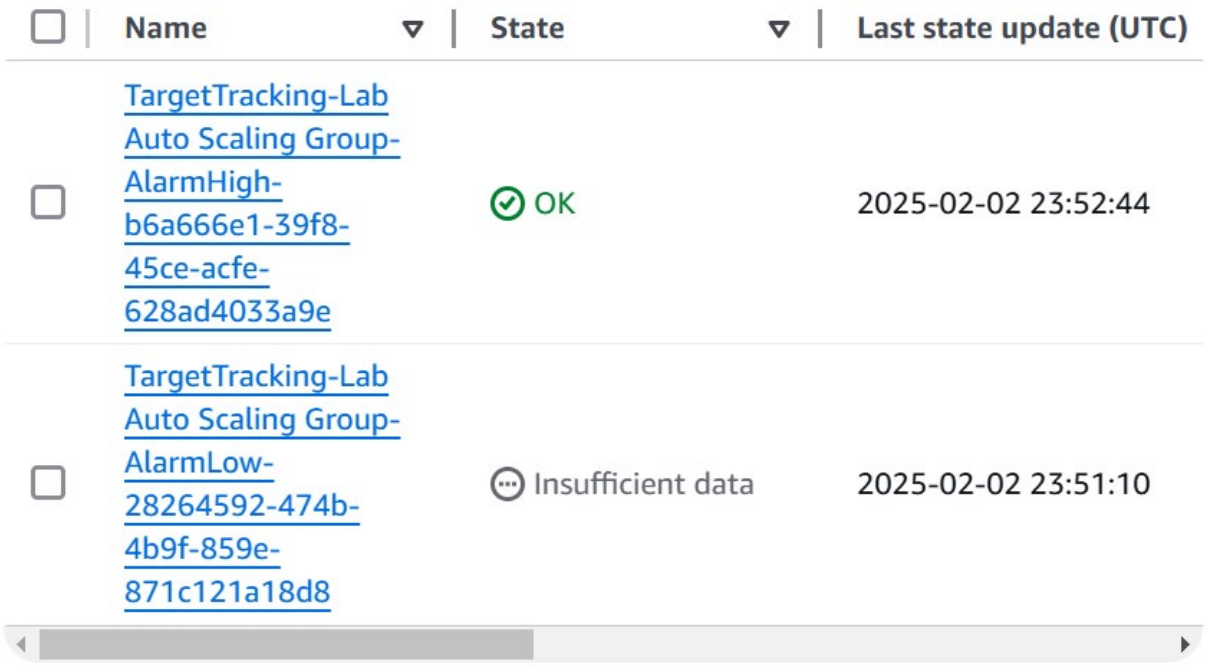
The browser tab may be similar to the one above when the application is running. Next, search for CloudWatch and open it without closing the application tab. The following screen should appear.



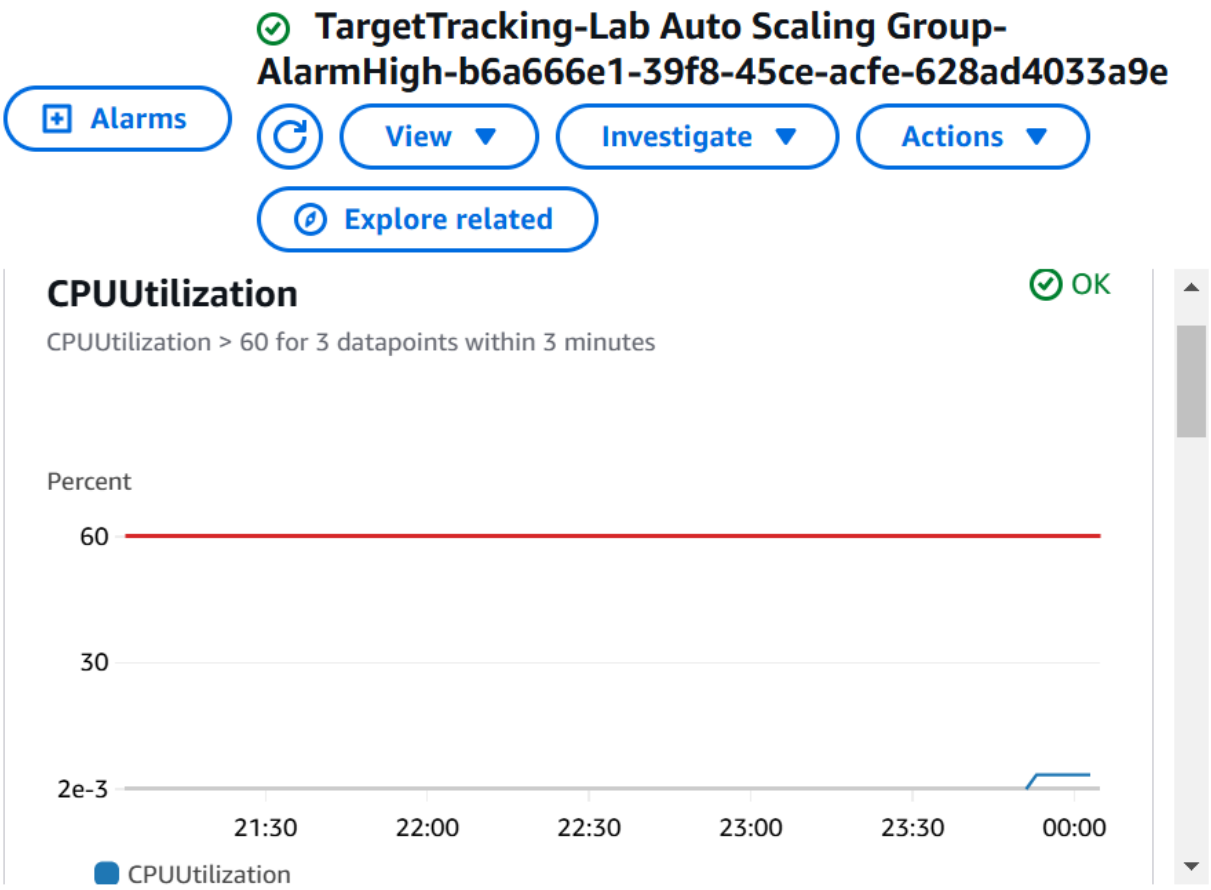
Click on the Alarms drop-down menu and select All alarms.



There should be two alarms displayed.



Choose the OK alarm (the alarm containing AlarmHigh in its name). The following page should load.



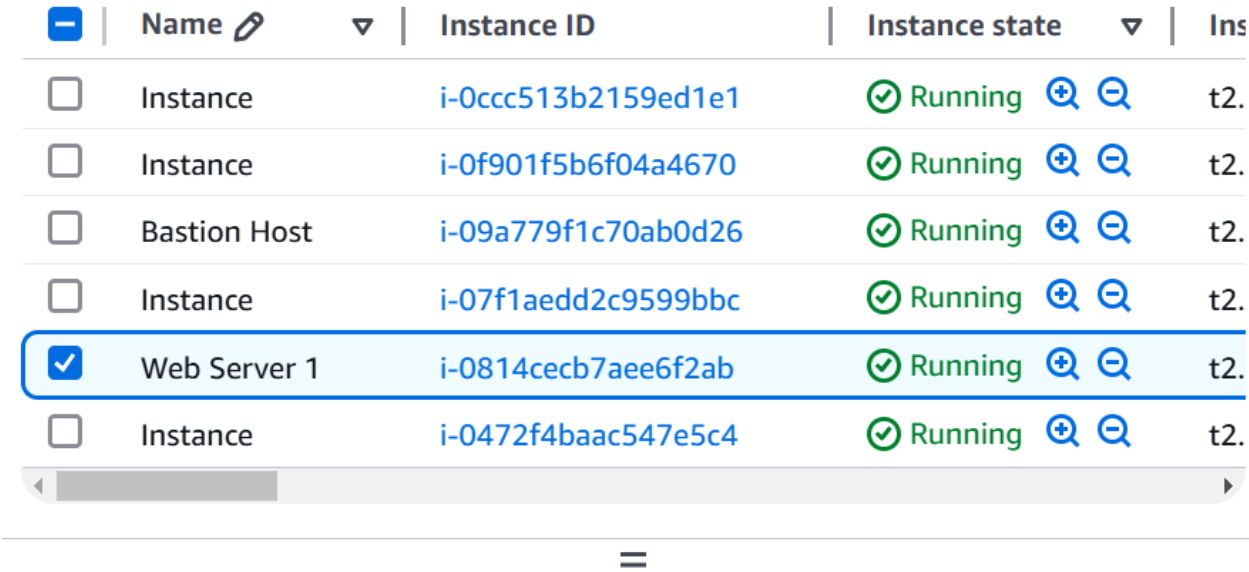
Return to the web application and click on Load Test. Return to the CloudWatch console page with the two alarms and wait until the AlarmLow alarm changes to OK and AlarmHigh alarm changes to In alarm.

Name	State	Last state update (UTC)
TargetTracking-Lab Auto Scaling Group-AlarmLow-28264592-474b-4b9f-859e-871c121a18d8	In alarm	2025-02-03 00:06:55
TargetTracking-Lab Auto Scaling Group-AlarmHigh-b6a666e1-39f8-45ce-acfe-628ad4033a9e	OK	2025-02-02 23:52:44

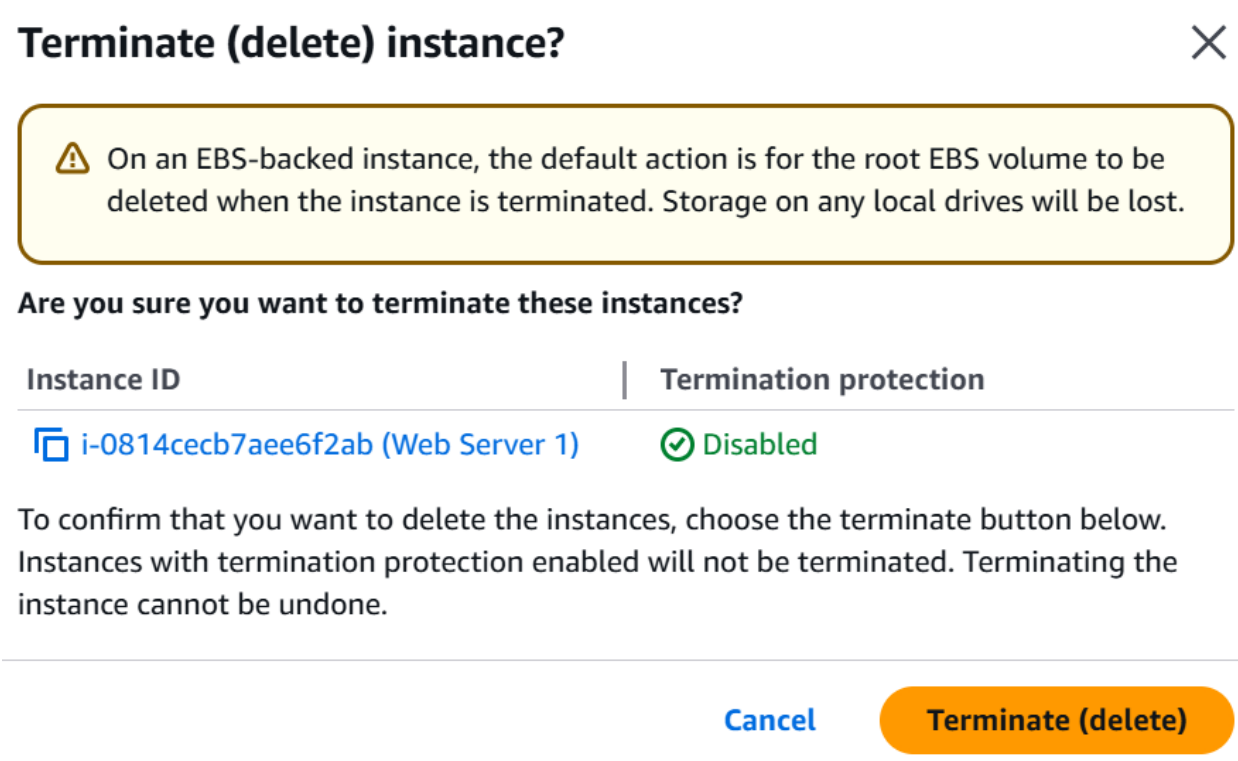
Then, go to EC2 and into Instances through the left navigation pane. More than two instances called Instance should be running in response to the CloudWatch alarm.

Name	Instance ID	Instance state
Instance	i-0ccc513b2159ed1e1	Running
Instance	i-0f901f5b6f04a4670	Running
Bastion Host	i-09a779f1c70ab0d26	Running
Instance	i-07f1aedd2c9599bbc	Running
Web Server 1	i-0814cecb7aee6f2ab	Running
Instance	i-0472f4baac547e5c4	Running

Select Web Server 1 and make sure it is the only instance selected.



In the Instance state drop-down menu, choose Instance State and then click on Terminate Instance.



Confirm to Terminate.

There should be a success message:

PROBLEMS:

In Lab 5, there were several problems that I encountered, one of which affected my scoring for the ab. First, I didn't uncheck Enable Enhanced Monitoring, so I had to remake the database and delete the current one. The second time, I had the wrong DB instance identifier entered, so I had to delete the database and redo the database configuration. Afterwards, my session timed out and I was logged out of the AWS Console, so I had to restart the lab. Finally, the web page did not display anything even after the correct credentials were submitted and everything prior to this segment was displayed as correctly configured; I was unable to fix this problem due to having no incorrect configurations and just left it as is.

In Lab 6, instead of an instance named Lab Instance, the instances that were created in response were all named Instance, so the grader thought I did not finish the Auto Scaling check task. Therefore, I had to change the instance names to Lab Instance for it to register in the system, which did work. I ended up getting a full score on Lab 6.

CONCLUSION:

These labs were a lot harder than the previous labs in my opinion. Although they were equally as straightforward, some steps had lots of opportunities for mistakes, leading me to spend more time on these labs than before. However, I'm glad I was able to troubleshoot and persevere—two skills critical to becoming a network administrator.

SIGN-OFF:

Lab 4:

Total score	25/25
Task 1 - Create EBS volume	5/5
Task 2 - Attach volume	5/5
Task 4 - Volume mounted	5/5
Task 5 - Snapshot created	5/5
Task 6 - Snapshot restored	5/5

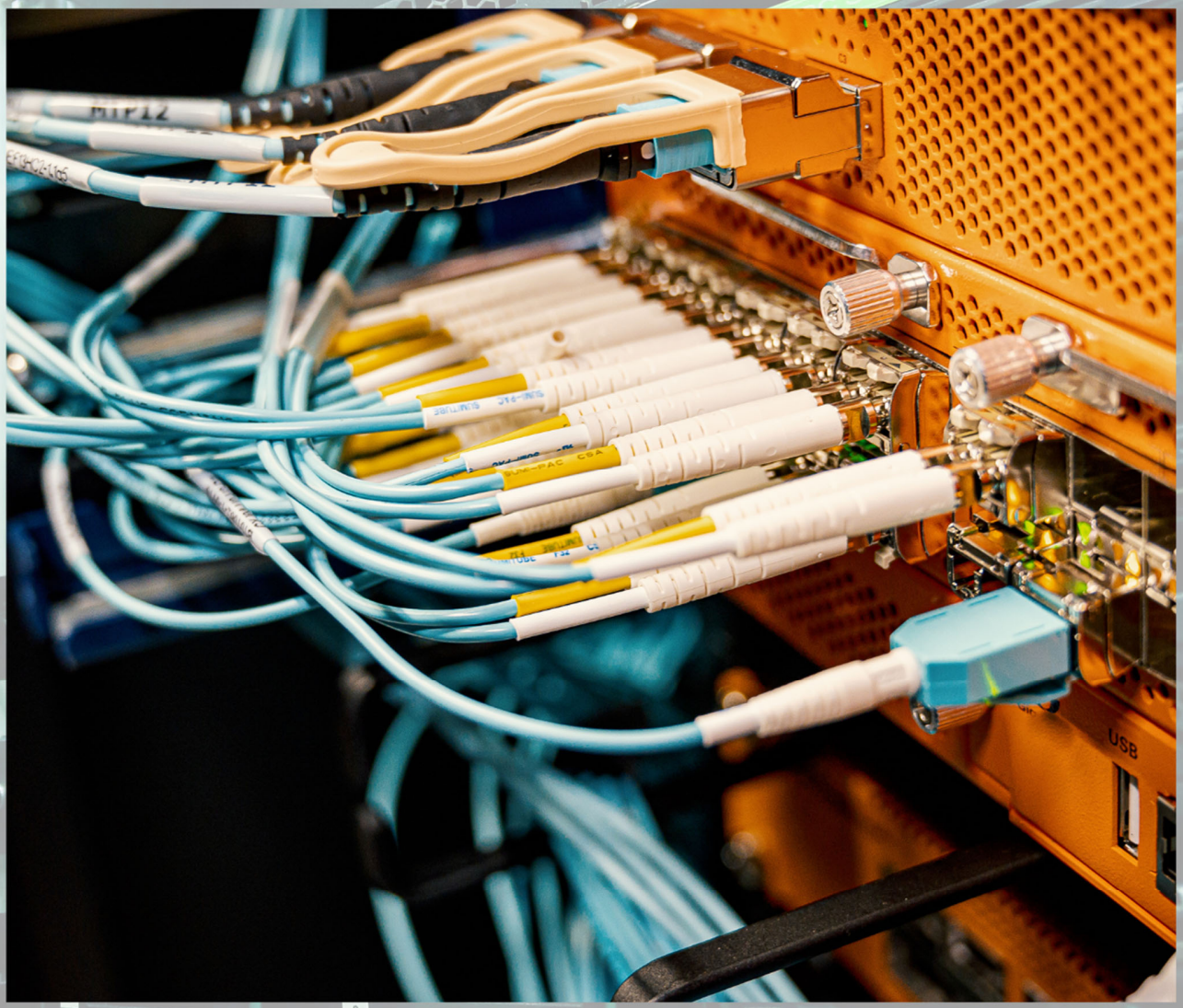
Lab 5:

Total score	15/20
Task 1 - Security Group created	5/5
Task 2 - DB subnet group	5/5
Task 3 - DB created	5/5
Task 4 - App connected to DB	0/5

Lab 6:

Total score	35/35
Task 1 - AMI created	5/5
Task 2 - Load Balancer created	5/5
Task 3a - Launch Template created	5/5
Task 3b - Auto Scaling Group created	5/5
Task 4 - Load Balancer check	5/5
Task 5 - Auto Scaling check	5/5
Task 6 - Web Server 1	5/5

Lab 7: AP Configuration with RADIUS



PURPOSE:

Learn to configure IS-IS, Intermediate System to Intermediate System, in preparation for a guest speaker who made IS-IS and will be visiting in a few weeks.

BACKGROUND INFORMATION ON LAB CONCEPTS:

IS-IS (Intermediate System to Intermediate System) is designed to be an interior gateway protocol. It is also a link-state routing protocol and uses the same algorithm as OSPF to find the best path. There are also some other similarities, like how their networks both converge quickly and they both have automated neighbor discovery.

However, IS-IS is different from OSPF in that there are three types of routers used to connect different areas together: Level 1, Level 1-2, and Level 2. Level 1 can only establish neighbor adjacencies for routers in the same area; Level 2 can only establish neighbor adjacencies for routers in different areas; and Level 1-2 can perform both functions. Therefore, Level 1 is often used for routers in the local area only. Level 1-2, on the other hand, is for routers that are adjacent to both a local router and a router in a different area. They bridge the two types of routers, allowing them to exchange information. Level 2 is for routing between different areas and only establish adjacencies with other Level 2 routers. They're responsible for inter-area routing.

An analogy to illustrate the role of different routers in IS-IS is the following. Imagine in a company, there are employees, secretaries, and high-level bosses. Employees can pass messages to other employees, but they need to inform a secretary to communicate with their high-level bosses. The high-level bosses do not individually talk to their employees unless their secretary mentions it. Here, employees are like the Level 1 routers; secretaries are like the Level 1-2 routers; and high-level bosses are like the Level 2 routers.

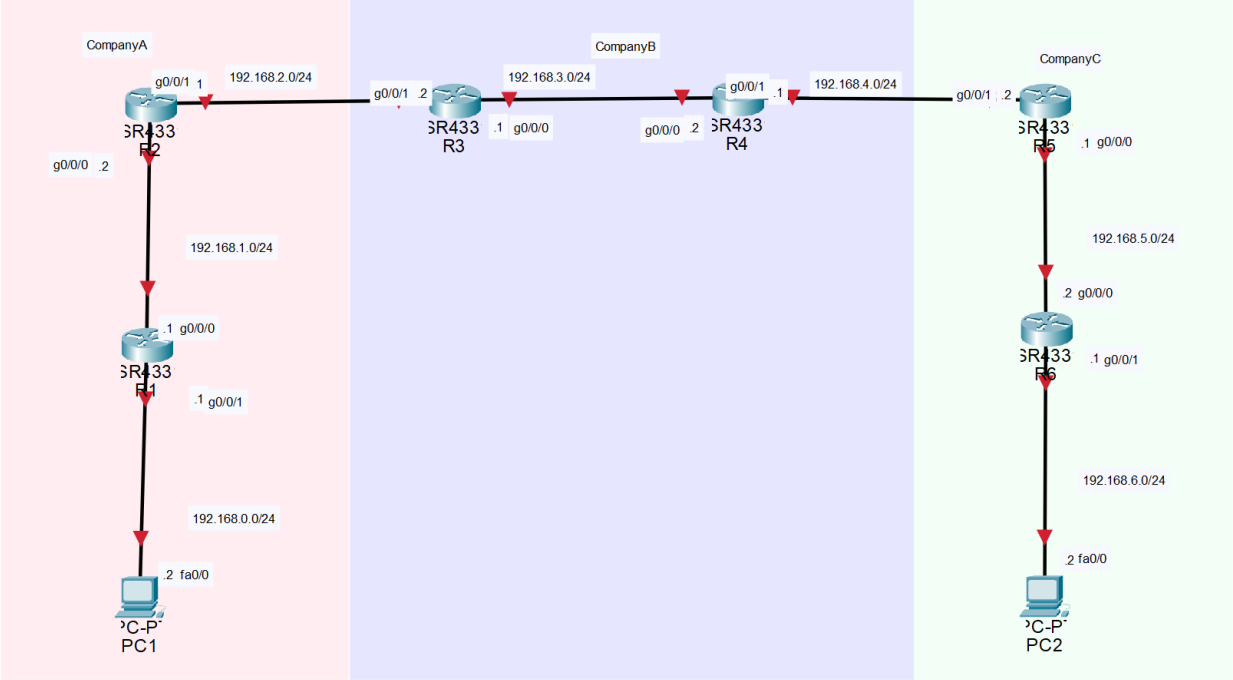
There are a few more differences between IS-IS and OSPF. For example, IS-IS operates directly over the data link layer and uses protocol data units for communication, unlike OSPF, which uses Layer 3 IP for encapsulation. Thus, IS-IS is not dependent on IP for routing and supports multiple network layer protocols instead of just IP, which offers more options for a network administrator. IS-IS also does not have two different versions for IPv4 and IPv6, instead supporting both within IS-IS itself, unlike OSPF, which has OSPFv2 and OSPFv3. This means that it is easier to configure. Instead of assigning clear-cut areas like OSPF (which also requires there be a strong distinction between Area 0 and other areas), IS-IS opts for a less rigid structure, only designating router levels, to allow for more room for customization for network architects.

IS-IS provides a big advantage in terms of large-scale networks for its flexibility and configurability, as well as its performance in more complex topologies; however, OSPF may be better in enterprise networks because its areas allow network administrators to be able to monitor and control the network more easily and efficiently. It also has been more widely tested and used in different networks, so enterprises are likely to find OSPF more reliable and easier to find documentation about.

LAB SUMMARY:

Created an IS-IS network infrastructure for 3 companies (Company A, Company B, and Company C) using six routers and two end devices.

NETWORK DIAGRAMS:



Router	IS-IS Level
R1	Level 1
R2	Level 1-2
R3	Level 2
R4	Level 2
R5	Level 1-2
R6	Level 1

LAB COMMANDS:

After configuring network addresses for interfaces on each router, set up IS-IS:

ip router isis enables an interface to participate in IS-IS for IPv4.

Since the routers will use different areas, different levels of IS-IS will be configured. Enter the command *router isis* and use the global command *is-type level-1* to configure Level 1. Similarly, use the global command *is-type level-1-2* and *is-type level-2* to configure Level 1-2 and Level 2, respectively.

Specify a Network Entity Title (NET), which is the "ID" of the router in IS-IS and also determines which area the router belongs to. For example: *net 49.0001.1920.1680.0001.00.* In this case, the .0001 signifies the area, 1, that the router is in.

CONFIGURATIONS:

Note that certain output has been removed for clarity and relevance.

Traceroute from R1 to R6:

```
R1#traceroute 192.168.6.1
Type escape sequence to abort.
Tracing the route to 192.168.6.1
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.1.2 0 msec 0 msec 0 msec
  2 192.168.2.2 0 msec 1 msec 0 msec
  3 192.168.3.2 1 msec 1 msec 0 msec
  4 192.168.4.2 1 msec 1 msec 1 msec
  5 192.168.5.2 1 msec 1 msec *
```

Hostname R1—show IP route and show running configuration:

```
R1#show ip route
i*L1  0.0.0.0/0 [115/10] via 192.168.1.2, 01:14:51,
GigabitEthernet0/0/0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.1.1/32 is directly connected,
GigabitEthernet0/0/0
i L1  192.168.2.0/24 [115/20] via 192.168.1.2, 01:15:01,
GigabitEthernet0/0/0

R1#show run
interface GigabitEthernet0/0/0
 ip address 192.168.1.1 255.255.255.0
 ip router isis
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.0.1 255.255.255.0
 ip router isis
 negotiation auto
!
```

```
interface Serial0/1/0
 no ip address
 shutdown
!
interface Serial0/1/1
 no ip address
 shutdown
!
interface GigabitEthernet0/2/0
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/2/1
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
!
interface Vlan1
 no ip address
 shutdown
!
router isis
 net 49.0001.0000.0000.0001.00
 is-type level-1
 log-adjacency-changes
```

Hostname R2—show IP route and show running configuration:

```
R2#show ip route
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.1.2/32 is directly connected,
GigabitEthernet0/0/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected,
GigabitEthernet0/0/1
L        192.168.2.1/32 is directly connected,
GigabitEthernet0/0/1
i L2  192.168.3.0/24 [115/20] via 192.168.2.2, 01:15:46,
GigabitEthernet0/0/1
```

```
i L2  192.168.4.0/24 [115/30] via 192.168.2.2, 01:15:13,
GigabitEthernet0/0/1
i L2  192.168.5.0/24 [115/40] via 192.168.2.2, 01:14:41,
GigabitEthernet0/0/1
i L2  192.168.6.0/24 [115/50] via 192.168.2.2, 00:02:57,
GigabitEthernet0/0/1

R2#show run
interface GigabitEthernet0/0/0
 ip address 192.168.1.2 255.255.255.0
 ip router isis
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.2.1 255.255.255.0
 ip router isis
 negotiation auto
!
interface Serial0/1/0
 no ip address
 shutdown
!
interface Serial0/1/1
 no ip address
 shutdown
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
!
interface Vlan1
 no ip address
 shutdown
!
router isis
 net 49.0001.0000.0000.0002.00
 log-adjacency-changes
```

Hostname R3—show IP route and show running configuration:

```
R3#show ip route
i L2  192.168.1.0/24 [115/20] via 192.168.2.1, 01:16:58,
GigabitEthernet0/0/1
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected,
GigabitEthernet0/0/1
```

```
L        192.168.2.2/32 is directly connected,
GigabitEthernet0/0/1
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.3.1/32 is directly connected,
GigabitEthernet0/0/0
i L2  192.168.4.0/24 [115/20] via 192.168.3.2, 01:15:55,
GigabitEthernet0/0/0
i L2  192.168.5.0/24 [115/30] via 192.168.3.2, 01:15:23,
GigabitEthernet0/0/0
i L2  192.168.6.0/24 [115/40] via 192.168.3.2, 00:03:39,
GigabitEthernet0/0/0

R3#show run
interface GigabitEthernet0/0/0
 ip address 192.168.3.1 255.255.255.0
 ip router isis
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.2.2 255.255.255.0
 ip router isis
 negotiation auto
!
interface Serial0/1/0
 no ip address
!
interface Serial0/1/1
 no ip address
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 negotiation auto
!
interface Vlan1
 no ip address
!
router isis
 net 49.0002.0000.0000.0003.00
 is-type level-2-only
 log-adjacency-changes
```

Hostname R4—show IP route and show running configuration:

```
R4#show ip route
```

```
i L2  192.168.1.0/24 [115/30] via 192.168.3.1, 01:17:07,
GigabitEthernet0/0/0
i L2  192.168.2.0/24 [115/20] via 192.168.3.1, 01:17:07,
GigabitEthernet0/0/0
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.3.2/32 is directly connected,
GigabitEthernet0/0/0
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected,
GigabitEthernet0/0/1
L        192.168.4.1/32 is directly connected,
GigabitEthernet0/0/1
i L2  192.168.5.0/24 [115/20] via 192.168.4.2, 01:16:01,
GigabitEthernet0/0/1

R4#show run
interface GigabitEthernet0/0/0
 ip address 192.168.3.2 255.255.255.0
 ip router isis
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.4.1 255.255.255.0
 ip router isis
 negotiation auto
!
interface Serial0/1/0
 no ip address
 shutdown
!
interface Serial0/1/1
 no ip address
 shutdown
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
!
interface Vlan1
 no ip address
 shutdown
!
router isis
```

```
 net 49.0002.0000.0000.0004.00
 is-type level-2-only
 log-adjacency-changes
```

Hostname R5—show IP route and show running configuration:

```
R5#show ip route
i L2  192.168.1.0/24 [115/40] via 192.168.4.1, 01:17:13,
GigabitEthernet0/0/1
i L2  192.168.2.0/24 [115/30] via 192.168.4.1, 01:17:13,
GigabitEthernet0/0/1
i L2  192.168.3.0/24 [115/20] via 192.168.4.1, 01:17:13,
GigabitEthernet0/0/1
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected,
GigabitEthernet0/0/1
L        192.168.4.2/32 is directly connected,
GigabitEthernet0/0/1
      192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.5.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.5.1/32 is directly connected,
GigabitEthernet0/0/0

R5#show run
interface GigabitEthernet0/0/0
 ip address 192.168.5.1 255.255.255.0
 ip router isis
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.4.2 255.255.255.0
 ip router isis
 negotiation auto
!
interface Serial0/1/0
 no ip address
 shutdown
!
interface Serial0/1/1
 no ip address
 shutdown
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
```

```
!
interface Vlan1
 no ip address
 shutdown
!
router isis
 net 49.0003.0000.0000.0005.00
 log-adjacency-changes
```

Hostname R6—show IP route and show running configuration:

```
R6#show ip route
i*L1  0.0.0.0/0 [115/10] via 192.168.5.1, 01:17:28,
GigabitEthernet0/0/0
i L1  192.168.4.0/24 [115/20] via 192.168.5.1, 01:17:28,
GigabitEthernet0/0/0
      192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.5.0/24 is directly connected,
GigabitEthernet0/0/0
L        192.168.5.2/32 is directly connected,
GigabitEthernet0/0/0

R6#show run
interface GigabitEthernet0/0/0
 ip address 192.168.5.2 255.255.255.0
 ip router isis
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.6.1 255.255.255.0
 ip router isis
 negotiation auto
!
interface GigabitEthernet0/1/0
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/1/1
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 shutdown
 negotiation auto
```

```
!
interface Vlan1
 no ip address
 shutdown
!
router isis
 net 49.0003.0000.0000.0006.00
 is-type level-1
 log-adjacency-changes
```

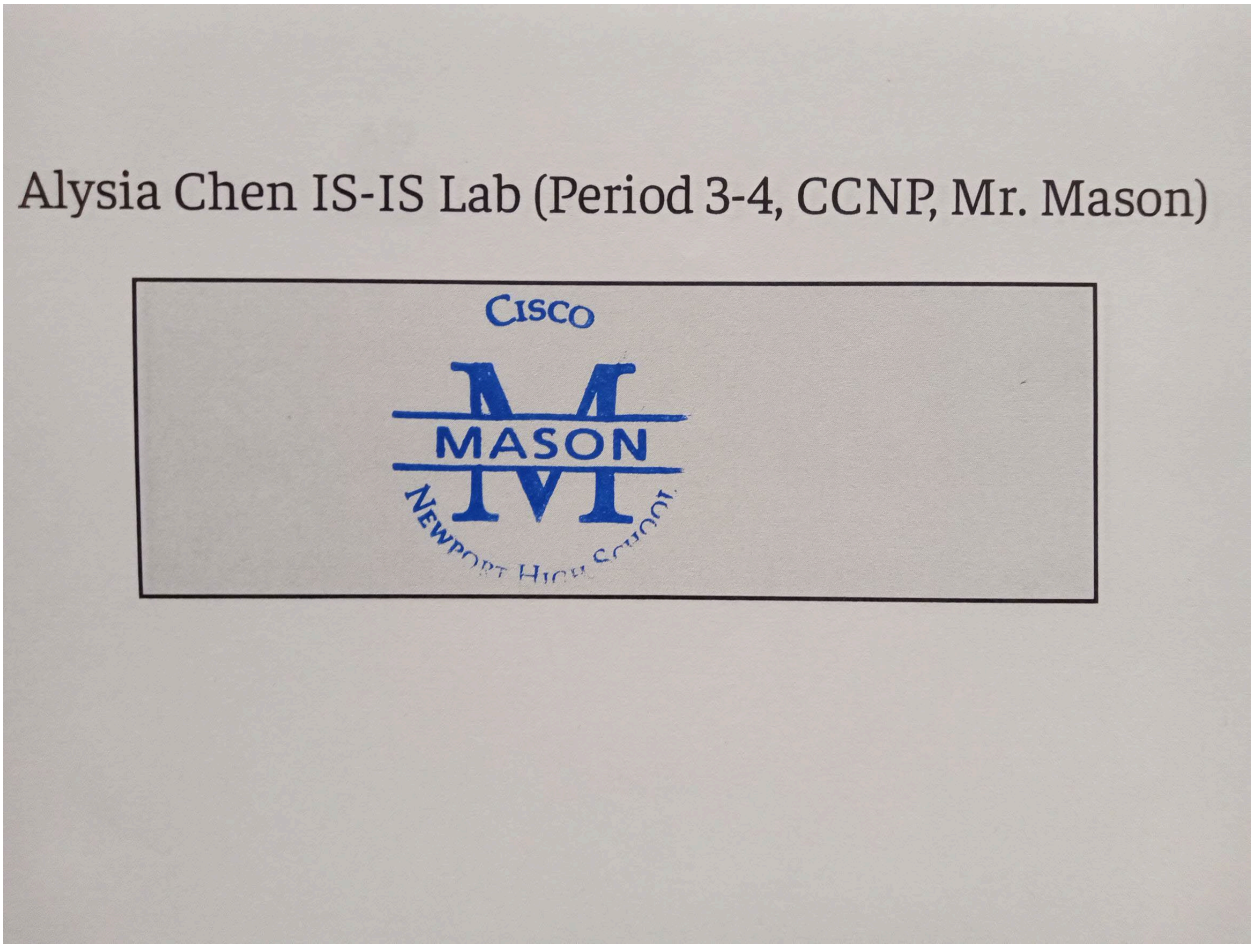
PROBLEMS:

As this is my first time working alone, it was very inefficient to console into 6 routers and configure them; the same work could have been done in half the time it took for me. Additionally, I spent a significant amount of time trying to make sure the running configuration was saved properly and that I didn't accidentally save the running configuration to the startup configuration.

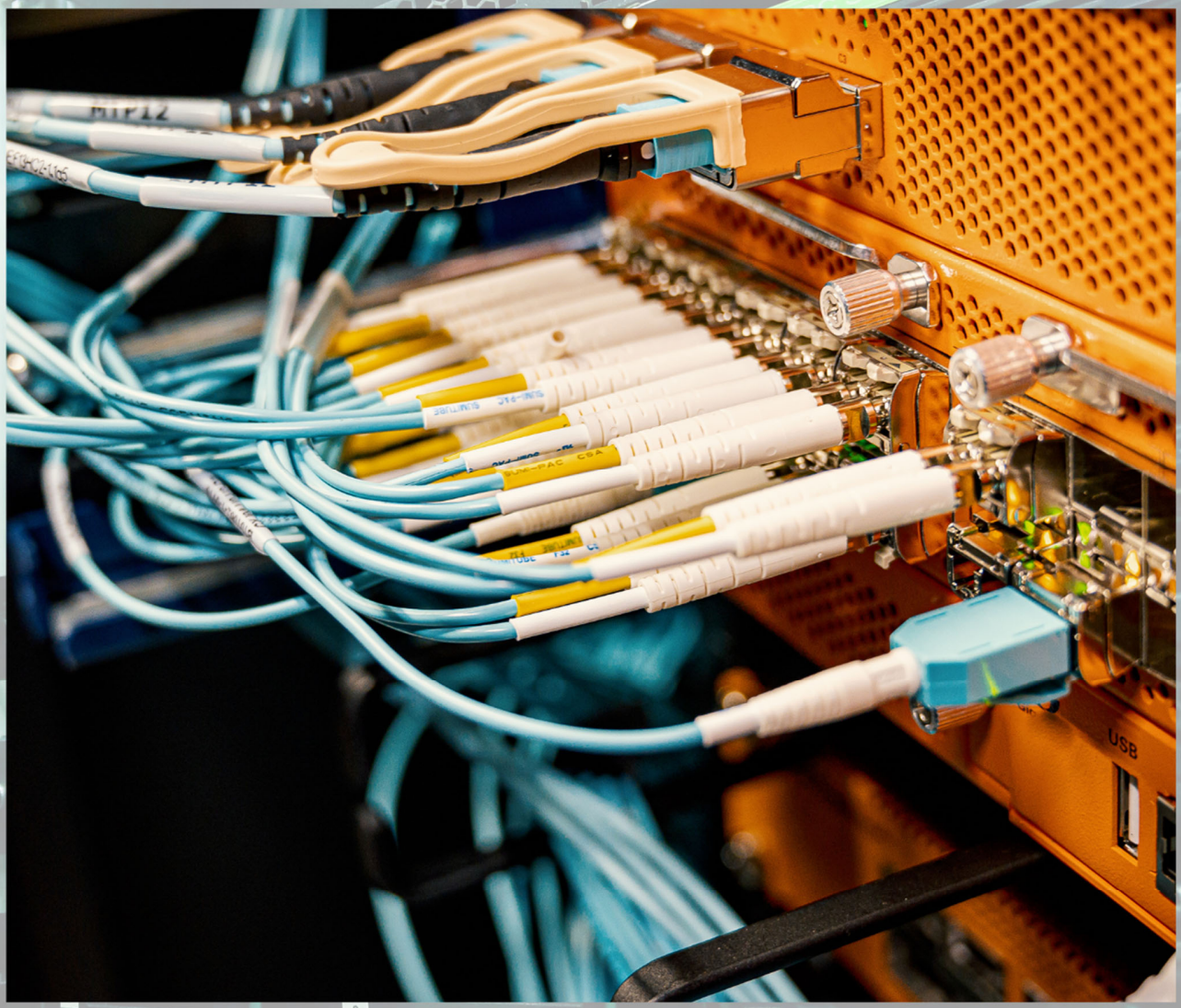
CONCLUSION:

This lab was fairly simple for me, but I'm especially proud I was able to complete it working solo! I hope that in CCNP I can not only learn how to work with others like I did in the past, but be able to solve problems on my own and through online resources to get good at both asking people for help and researching online for answers.

TEACHER SIGN-OFF:



Lab 8: IS-IS



PURPOSE:

Learn to configure three wireless networks using a wireless access point.

BACKGROUND INFORMATION ON LAB CONCEPTS:

An access point (AP) is a device that allows local, wireless devices to connect to the Internet by creating a wireless local area network (WLAN). It has security features and is able to support multiple devices, making it ideal for many different types of networks.

On the router, Port Address Translation (PAT), Dynamic Host Configuration Protocol, and access lists must be configured for hosts to be able to access the cloud (which is connected via a blue wire in this case).

Port Address Translation (PAT) is a specific type of Network Address Translation. Network Address Translation (NAT) is needed here because a private IP address cannot directly route to the internet unless it has a public IP address that it can "translate" to and that external networks can recognize. In other words, NAT translates a private IP address to a public IP address to route to the internet.

PAT is used in this lab because it is more scalable for our purposes than other types of NAT: it only uses one public IP address, assigning different router port numbers to each connection, and allows multiple devices through it. PAT uses both port number and IP address to translate a private IP address to the public IP address; the command *overload* for multiple devices to share the same public IP address, which distinguishes it from other subsets of Network Address Translation. This is especially important in this lab because there are three VLANs, or isolated networks, and thus three subnets of IP addresses for each, all requiring internet access through a single router.

Dynamic Host Configuration Protocol (DHCP) requires a pool of private IP addresses to be configured so that the router can automatically assign a private IP address to each device. Finally, access lists control what is permitted and what is not permitted to enter or exit a port, filtering traffic and allowing for the necessary packets of information to be transmitted.

There are many different servers being used here. The Domain Name System (DNS) server used in this particular lab can be found here: https://developers.google.com/speed/public-dns. DNS ensures that the domain name a user enters can be directly translated into the corresponding IP address so the user does not have to enter IP addresses to use the Internet. Another server is the DHCP server, which is just the router in this case (refer to the above explanation of DHCP). Finally, the last server is the RADIUS server, which is used for remote authentication, authorization, and accounting. In simpler terms, it is used to keep the network secure.

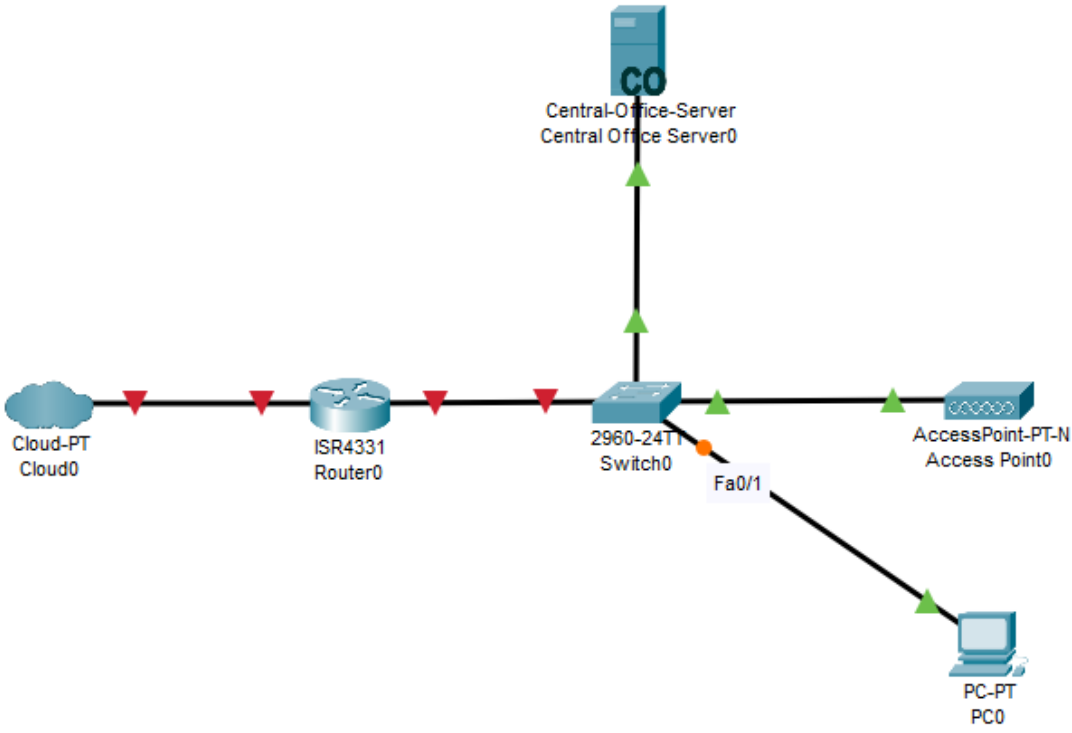
A trunk port carries traffic between different VLANs, while an access port carries traffic for only a single VLAN. A native trunk VLAN allows untagged traffic to be sent. Untagged traffic is traffic that does not have a VLAN tag but is assumed to be associated with the native VLAN. In this

case, since there are three wireless networks, three SSIDs (which are used to identify each network) and thus three VLANs to logically separate traffic are needed.

LAB SUMMARY:

Created 3 VLANs with 3 different SSIDs: one for Guest (no password), one that is Home (pre-shared WPA2 key), and one that is WPA2 Enterprise (using Linux and RADIUS).

NETWORK DIAGRAM:



Device (Interface)	IP Address
Switch VLAN 1	192.168.0.1/24
Switch VLAN 10	192.168.10.1/24
Switch VLAN 20	192.168.20.1/24
Switch VLAN 30	192.168.30.1/24
Router G0/0/1.1	192.168.0.2/24
Router G0/0/1.10	192.168.10.2/24

Router G0/0/1.20	192.168.20.2/24
Router G0/0/1.30	192.168.30.2/24
AP BVI1	192.168.0.3/24
PC0	192.168.0.4/24

LAB COMMANDS:

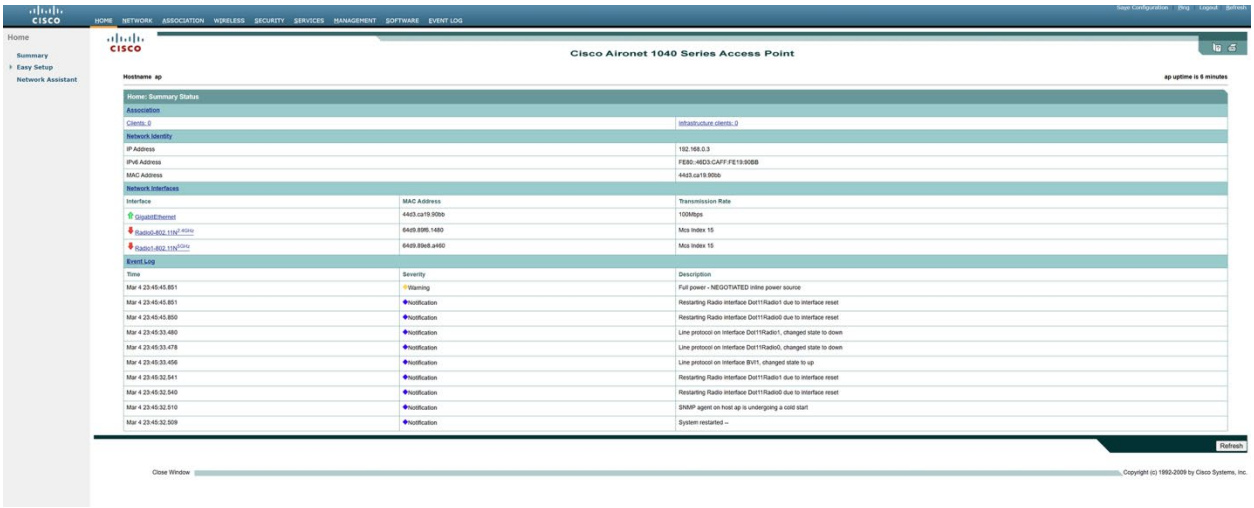
On the router, configure the appropriate IP addresses for each subinterface, with each subinterface corresponding to a VLAN, tagging each with the command *encapsulation dot1Q VLAN_number* and for the subinterface with the native VLAN, adding behind *VLAN_number* the keyword *native*. Make sure to create VLANs 10, 20, and 30 with the commands *vlan VLAN_number* and *name VLAN_name*.

Next, set up NAT. After determining which interface is used for connection to the internet, configure said interface to be the outside interface with the command *ip nat outside*. Configure the other interfaces with the command *ip nat inside*.

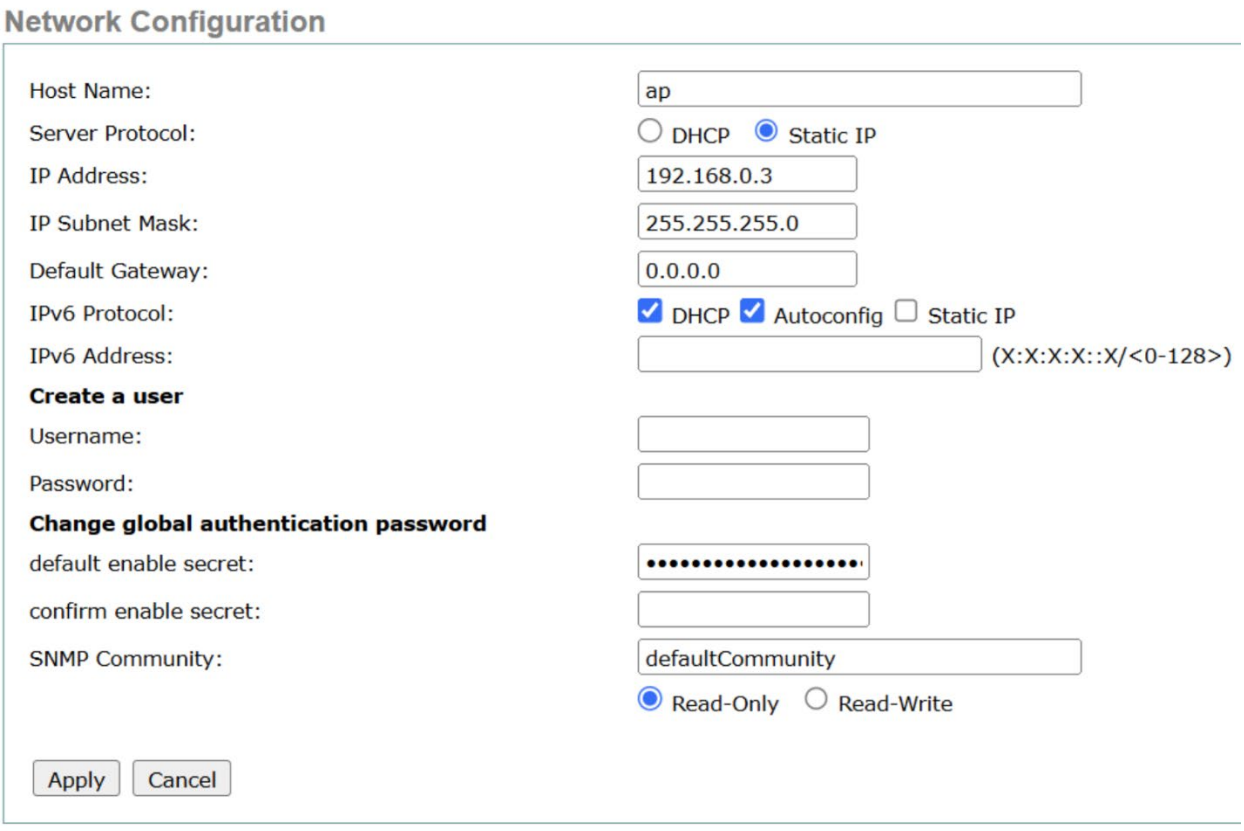
Afterwards, we will configure the access lists, permitting traffic through all of the VLANS with the commands *access-list 1 permit 192.168.0.0 0.0.0.255*, *access-list 1 permit 192.168.10.0 0.0.0.255*, *access-list 1 permit 192.168.20.0 0.0.0.255*, and *access-list 1 permit 192.168.30.0 0.0.0.255.*

The final configuration on the router involves DHCP. Enter the command *ip dhcp pool vlan[VLAN_number]* for each VLAN that is not the native VLAN. Configure the pool of DHCP addresses using the command *network [ip address] [subnet mask]*, *default-router [IP address of the router]*, and *dns-server 8.8.8.8 8.8.8.4*. Afterwards, configure the IP DHCP excluded addresses, the addresses that should not be leased out with DHCP, using the command *ip dhcp excluded-address [address 1] [address 2]*. Finally, configure the interface that is the outside interface for NAT with *ip address dhcp* to ensure it receives an address from DHCP.

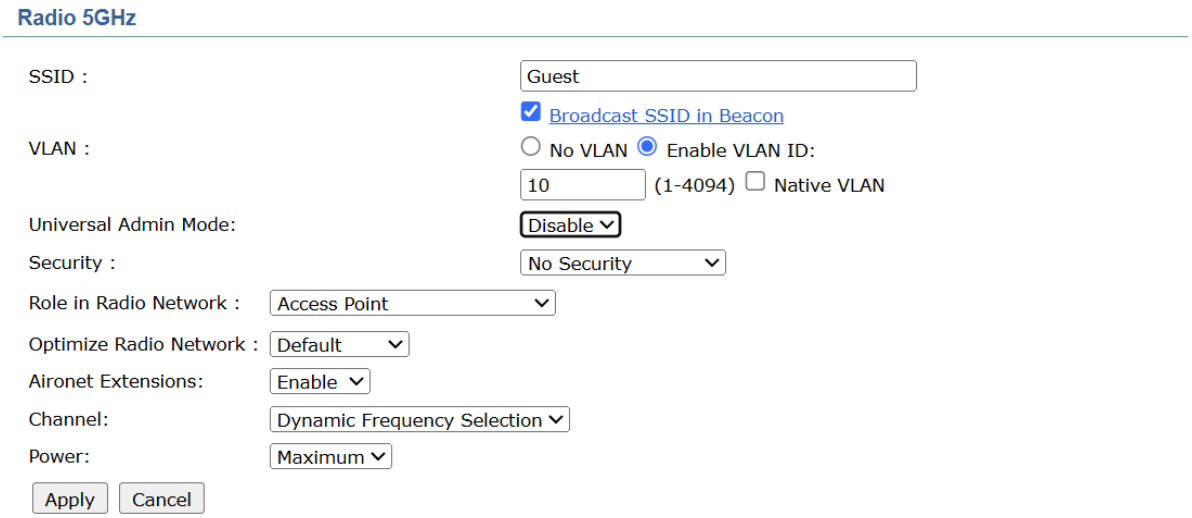
There are also a few commands that the switch should be configured with. For any interfaces directly connected to a router, switch, or any other network device besides a PC, set the interface to *switchport mode trunk* and enter *switchport trunk encapsulation dot1q*. For those connected to a PC (or RADIUS server), use the command *switchport mode access* and specify which VLAN with the command *switchport access vlan VLAN_number*.



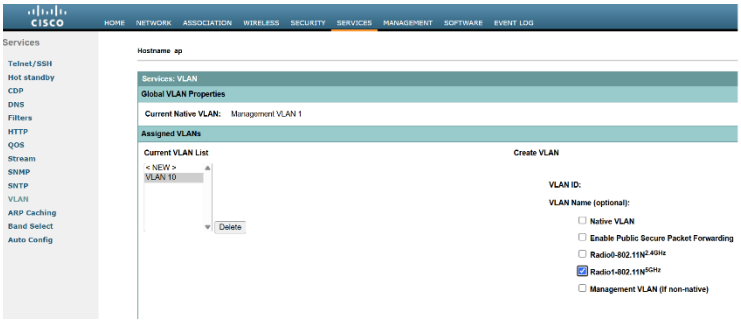
Next, navigate to under Easy Setup —> Network Configuration to ensure everything is set up correctly.



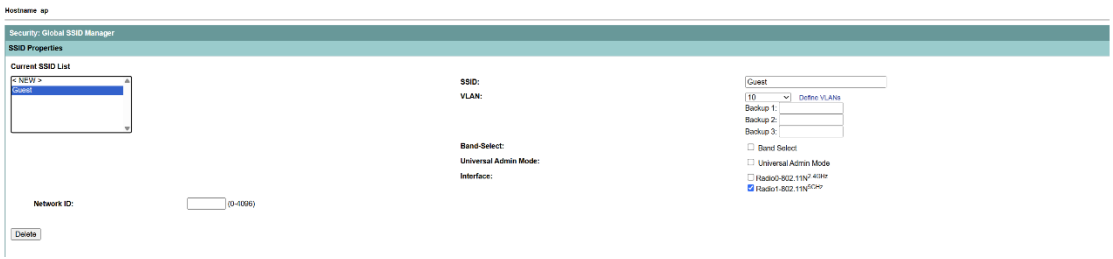
Under Radio Configuration, configure the following and ensure that Universal Admin Mode is disabled because that is used for a lightweight access point whereas in this lab we use an autonomous access point.



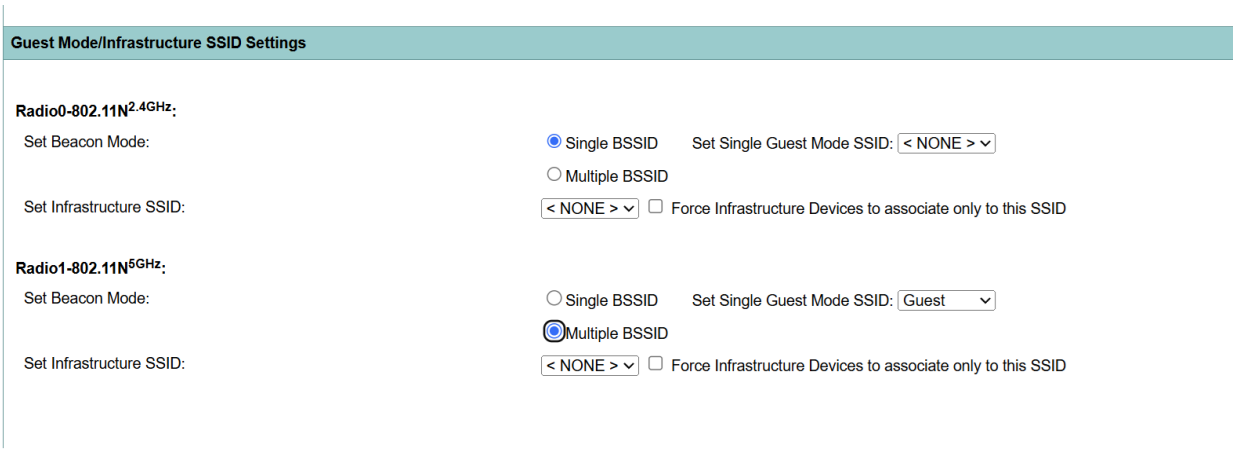
Under Services, make sure that the Radio1-802.11N^{5GHz} box is checked.



Under the Security tab, navigate to SSID Manager and verify that the previously configured SSID is present and select that SSID.



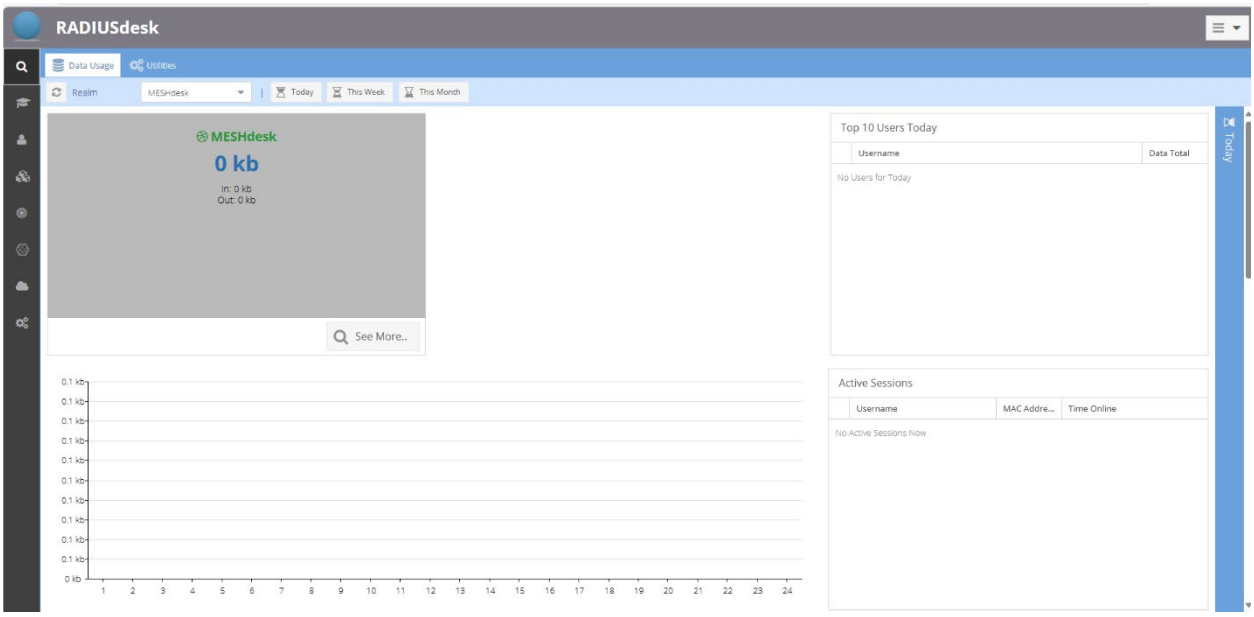
Turn on BSSID for the 5GHz option and hit apply. Repeat using 5GHz and create the SSIDs for Home and Enterprise.



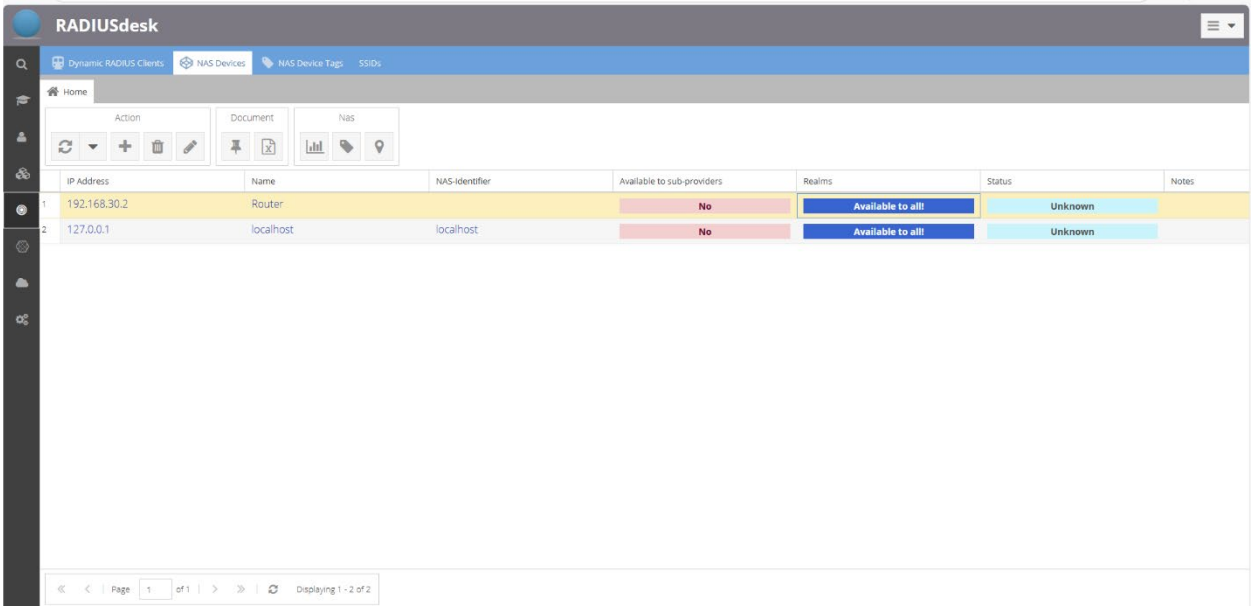
Next, we will configure the RADIUS server for authentication on the Enterprise network. First, download Oracle VirtualBox and radiusdesk-2017-0-3 ova files, completing set up. Open up the RADIUSdesk file in VirtualBox and login using the password *admin* and username *system*.

```
system@RADIUSdesk-2017-0-3:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:b3:75:b3
          inet6 addr: fe80::a00:27ff:feb3:75b3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6090 (6.0 KB)  TX bytes:16380 (16.3 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:24:81:ea
          inet6 addr: fe80::a00:27ff:fe24:81ea/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:32926 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4908 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:31594810 (31.5 MB)  TX bytes:343638 (343.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:3380 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3380 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:250645 (250.6 KB)  TX bytes:250645 (250.6 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.1.0.1  P-t-P:10.1.0.1  Mask:255.255.0.0
          UP POINTOPOINT RUNNING  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

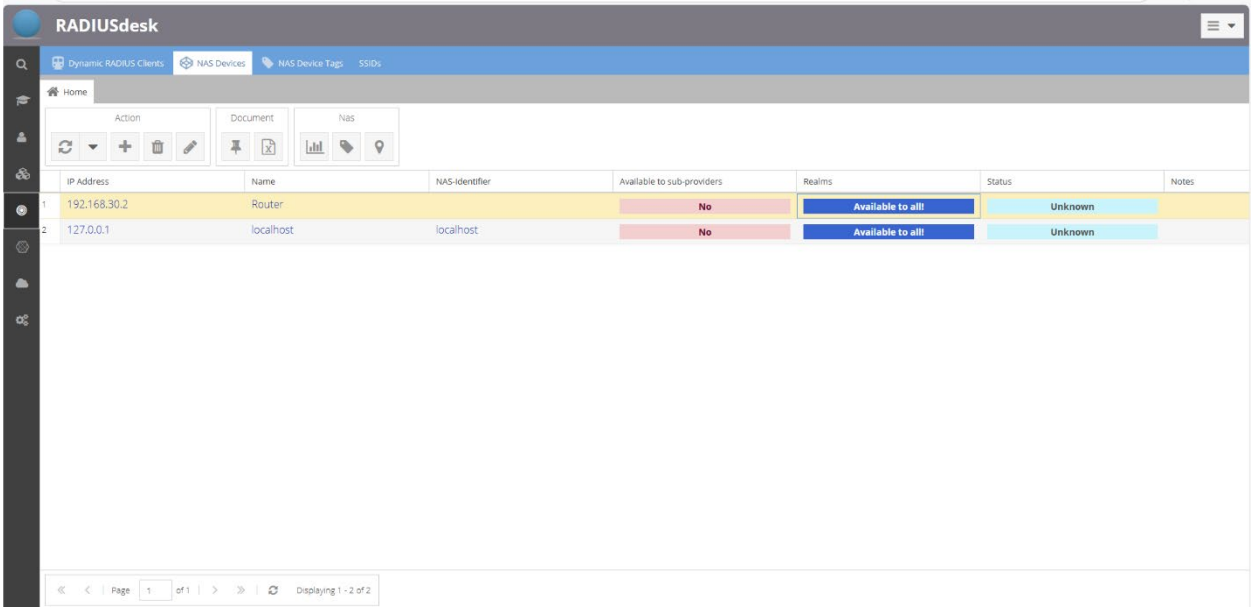
After typing *ifconfig*, find the eth0 IP address and type it into a web browser to access the graphical user interface of the RADIUS Server. The password for the graphical user interface is *admin*, and the username is *root*.



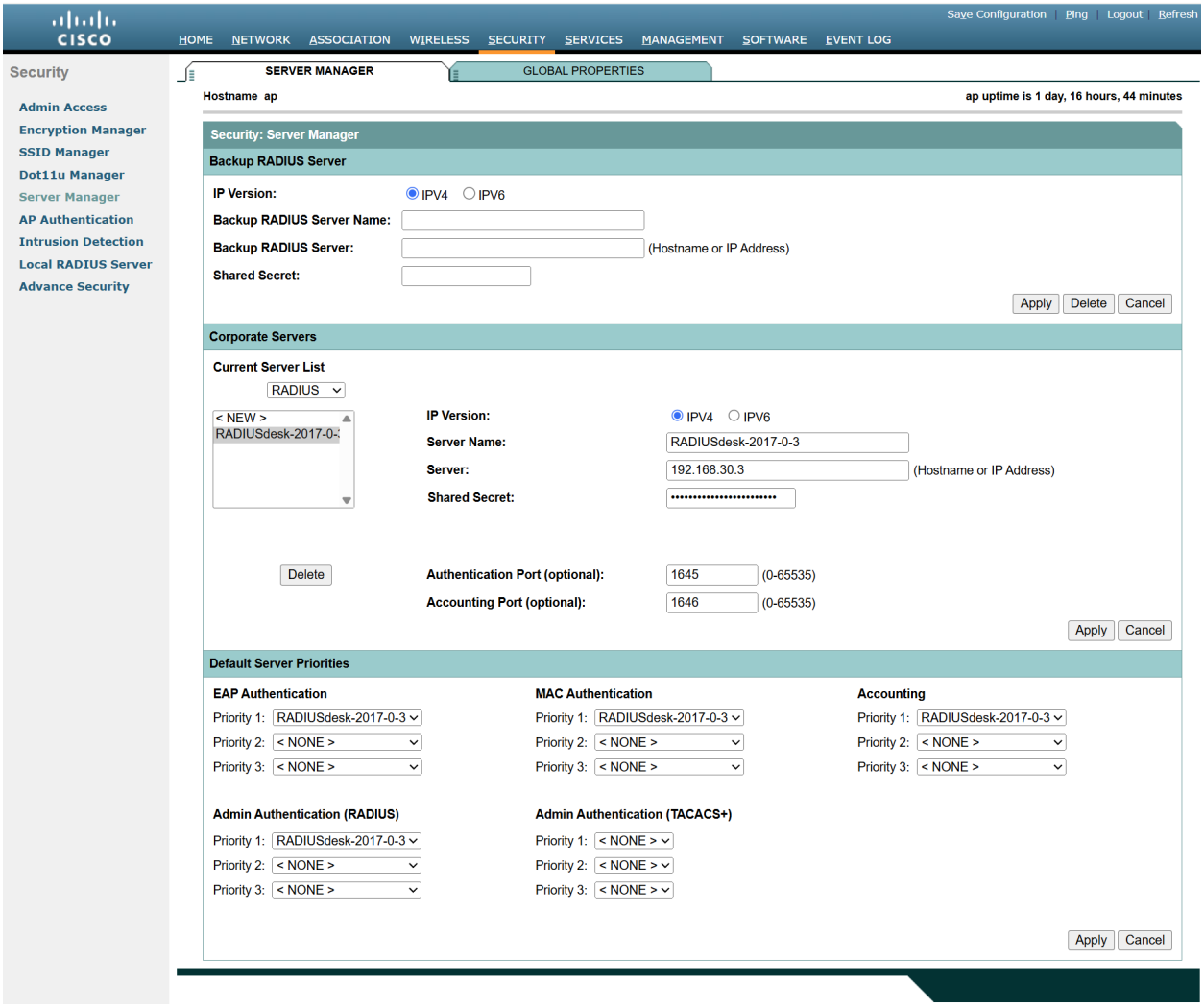
Under Action, click add, selecting admin_college:



Next, navigate to NAS Devices and click add, selecting admin_college and entering in the router IP address to add the router.



Afterwards, set the RADIUS server in the graphical interface of the access point.



Verify that the Enterprise network is able to use the RADIUS server with the correct user credentials to function properly, and that devices are able to connect to all three networks.

CONFIGURATIONS:

Note that certain output has been removed for clarity and relevance.

Hostname AP:

```
dot11 ssid Enterprise 2.0
   vlan 30
   authentication open eap eap_methods
   authentication network-eap eap_methods
   authentication key-management wpa version 2
   mbssid guest-mode
   no ids mfp client

dot11 ssid Guest
```

```
   vlan 10
   authentication open
   guest-mode
   mbssid guest-mode

dot11 ssid Home
   vlan 20
   authentication open
   authentication key-management wpa version 2
   mbssid guest-mode
   wpa-psk ascii 7 0822455D0A165445415A5E57

no ipv6 cef

username Cisco password 7 032752180500701E1D
username admin password 7 01300F175804575D72

bridge irb
interface Dot11Radio0
 no ip address
 encryption vlan 20 mode ciphers aes-ccm
 ssid Guest
 antenna gain 0
 speed  basic-1.0 basic-2.0 basic-5.5 basic-11.0 6.0 9.0 12.0
18.0 24.0 36.0 48.0 54.0 m0. m1. m2. m3. m4. m5. m6. m7. m8. m9.
m10. m11. m12. m13. m14. m15.
 station-role root
 bridge-group 1
 bridge-group 1 subscriber-loop-control
 bridge-group 1 spanning-disabled
 bridge-group 1 block-unknown-source
 no bridge-group 1 source-learning
 no bridge-group 1 unicast-flooding

interface Dot11Radio0.101
 encapsulation dot1Q 10
 bridge-group 10
 bridge-group 10 subscriber-loop-control
 bridge-group 10 spanning-disabled
 bridge-group 10 block-unknown-source
 no bridge-group 10 source-learning
 no bridge-group 10 unicast-flooding

interface Dot11Radio1
 no ip address
 !
 encryption vlan 20 mode ciphers aes-ccm
```

```
 !
 encryption vlan 30 mode ciphers aes-ccm
 !
 ssid Enterprise 2.0
 !
 ssid Home
 !
 antenna gain 0
 peakdetect
 dfs band 3 block
 mbssid
 channel dfs
 station-role root
 bridge-group 1
 bridge-group 1 subscriber-loop-control
 bridge-group 1 spanning-disabled
 bridge-group 1 block-unknown-source
 no bridge-group 1 source-learning
 no bridge-group 1 unicast-flooding

interface Dot11Radio1.20
 encapsulation dot1Q 20
 bridge-group 20
 bridge-group 20 subscriber-loop-control
 bridge-group 20 spanning-disabled
 bridge-group 20 block-unknown-source
 no bridge-group 20 source-learning
 no bridge-group 20 unicast-flooding

interface Dot11Radio1.101
 encapsulation dot1Q 10
 bridge-group 254
 bridge-group 254 subscriber-loop-control
 bridge-group 254 spanning-disabled
 bridge-group 254 block-unknown-source
 no bridge-group 254 source-learning
 no bridge-group 254 unicast-flooding

interface Dot11Radio1.301
 encapsulation dot1Q 30
 bridge-group 30
 bridge-group 30 subscriber-loop-control
 bridge-group 30 spanning-disabled
 bridge-group 30 block-unknown-source
 no bridge-group 30 source-learning
 no bridge-group 30 unicast-flooding
```

```
interface GigabitEthernet0
 no ip address
 duplex auto
 speed auto
 bridge-group 1
 bridge-group 1 spanning-disabled
 no bridge-group 1 source-learning

interface GigabitEthernet0.10
 encapsulation dot1Q 10
 bridge-group 10
 bridge-group 10 spanning-disabled
 no bridge-group 10 source-learning

interface GigabitEthernet0.20
 encapsulation dot1Q 20
 bridge-group 20
 bridge-group 20 spanning-disabled
 no bridge-group 20 source-learning

interface GigabitEthernet0.30
 encapsulation dot1Q 30
 bridge-group 30
 bridge-group 30 spanning-disabled
 no bridge-group 30 source-learning

interface BVI1
 mac-address 44d3.ca03.7dce
 ip address 192.168.0.3 255.255.255.0
 ipv6 address dhcp
 ipv6 address autoconfig
 ipv6 enable

ip default-gateway 192.168.0.2
ip forward-protocol nd
ip http server
no ip http secure-server
ip http help-
path http://www.cisco.com/warp/public/779/smbiz/prodconfig/help/
eag
ip radius source-interface BVI1

radius-server local
  nas 192.168.0.99 key 7 13261E010803557878
  user admin nthash 7
0322092E515776681D5E4F2635345D28520E0F777E6365734055335A2204080D
77
```

```
radius-server attribute 32 include-in-access-req format %h

radius server freeradius
 address ipv4 192.168.0.99 auth-port 1812 acct-port 1813
 key 7 13261E010803557878
```

Hostname AP_Router:

```
ip dhcp excluded-address 192.168.20.1 192.168.20.3
ip dhcp excluded-address 192.168.10.1 192.168.10.3
ip dhcp excluded-address 192.168.30.1 192.168.30.4

ip dhcp pool vlan10
 network 192.168.10.0 255.255.255.0
 default-router 192.168.0.2
 dns-server 8.8.8.8 8.8.8.4

ip dhcp pool vlan20
 network 192.168.20.0 255.255.255.0
 default-router 192.168.0.2
 dns-server 8.8.8.8 8.8.8.4

ip dhcp pool vlan30
 network 192.168.30.0 255.255.255.0
 default-router 192.168.0.2
 dns-server 8.8.8.8 8.8.8.4

vlan 10
 name Guest

vlan 20
 name Home

vlan 30
 name Enterprise

interface GigabitEthernet0/0/0
 ip address dhcp
 ip nat outside
 negotiation auto

interface GigabitEthernet0/0/1
 no ip address
 ip nat inside
 negotiation auto
```

```
interface GigabitEthernet0/0/1.1
 encapsulation dot1Q 1 native
 ip address 192.168.0.2 255.255.255.0
 ip nat inside

interface GigabitEthernet0/0/1.10
 encapsulation dot1Q 10
 ip address 192.168.10.2 255.255.255.0
 ip nat inside

interface GigabitEthernet0/0/1.20
 encapsulation dot1Q 20
 ip address 192.168.20.2 255.255.255.0
 ip nat inside

interface GigabitEthernet0/0/1.30
 encapsulation dot1Q 30
 ip address 192.168.30.2 255.255.255.0
 ip nat inside

ip nat inside source list 1 interface GigabitEthernet0/0/0
overload
ip forward-protocol nd
no ip http server
no ip http secure-server
ip tftp source-interface GigabitEthernet0

access-list 1 permit 192.168.0.0 0.0.0.255
access-list 1 permit 192.168.10.0 0.0.0.255
access-list 1 permit 192.168.20.0 0.0.0.255
access-list 1 permit 192.168.30.0 0.0.0.255
```

Hostname Switch:

```
interface FastEthernet0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk

interface FastEthernet0/2
 switchport trunk encapsulation dot1q
 switchport mode trunk

interface FastEthernet0/3
 switchport access vlan 30
 switchport trunk encapsulation dot1q
```

```
 switchport mode trunk

interface Vlan1
 ip address 192.168.0.1 255.255.255.0

interface Vlan10
 no ip address

interface Vlan20
 no ip address

interface Vlan30
 no ip address
```

PROBLEMS:

I had many problems throughout this lab. For example, although my Home Wi-Fi was able to work properly, the Guest Wi-Fi had a password, even though I did not configure a password. I ended up being able to fix this through going into the AP graphical user interface and selecting the Security section, finally navigating to Encryption Manager to turn off encryption on VLAN 10. However, once I tried connecting to the now-open network, there was no internet. After going into the command-line interface of the AP, I realized that interface dot11radio1.101, which is the subinterface used for guest, had a bridge group of 254 rather than 10. Upon changing that, I was able to get internet connection on the Guest Wi-Fi.

Unfortunately, I was unable to finish my lab. This was due to the multitude of problems I encountered while configuring my last Wi-Fi, the Enterprise one, which used a RADIUS server and required the server for authentication. Whenever I tried logging into the Wi-Fi, I received the message “Can’t connect to this network”.

Referring to the command-line interface of my AP, I see that the AP reads that the “authentication failed,” even though the username and password is the same as the one on the RADIUS server. Additionally, there are no firewalls blocking the port; the routing works because I am able to ping from my AP to the RADIUS server and the VLAN is correct because the configuration is very similar to the Guest and Home ones, which both work.

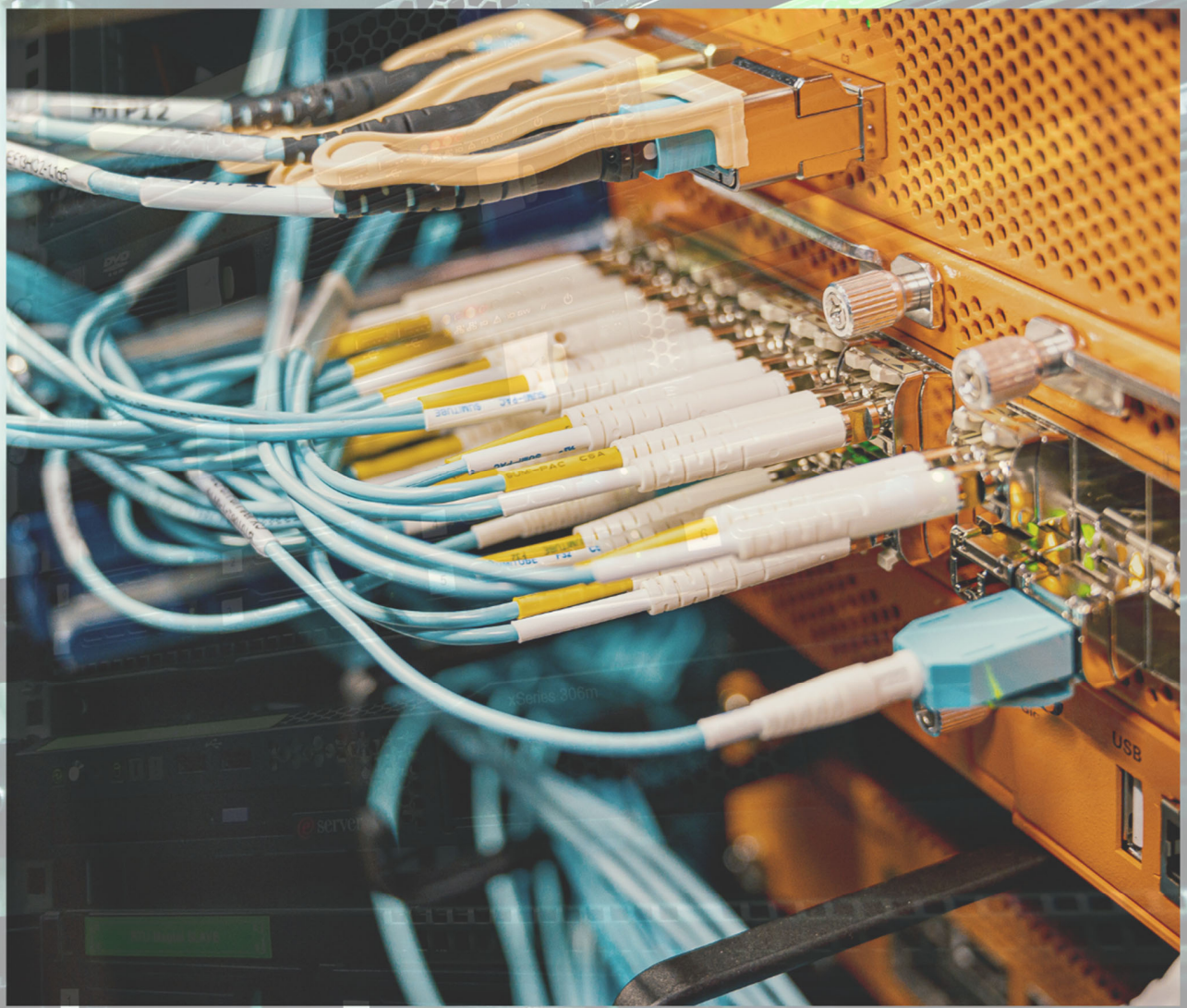
Additionally, I had tried using radtest to see if it is a problem with the RADIUS server not being able to run properly or having the wrong credentials configured, but the radtest was able to work for multiple user credentials. I also utilized the error logs. The error logs used to not show anything, but after restarting the RADIUS server, I was able to receive messages citing a connectivity issue.

After consulting with many online resources, my classmates, and my teacher, I was still unable to figure out the problem with my Enterprise network. However, I do believe the process of troubleshooting was highly valuable, and I do not regret trying my best in this lab.

CONCLUSION:

This lab was by far the most complex. Although I worked on it by my own, I had to ask for others' help multiple times due to the variety of troubleshooting methods I had to employ in order for me to get this far. Although I was not able to finish this lab, I believe that the valuable skills I obtained in communication, resilience, and practice with troubleshooting will allow me to advance my career in this field.

Lab 9: Layer 2 Attacks and Mitigations



PURPOSE:

Learn to mitigate three different types of attacks on networks.

BACKGROUND INFORMATION ON LAB CONCEPTS:

In networking, attacks are something network administrators have to take into account for when configuring a network. However, attacks not only come from external sources, but may also be coming from inside the network. In this lab, we will be using three different types of attacks, all of them occurring inside the local network: DHCP Starvation, ARP Spoofing, and MAC Flooding.

DHCP stands for Dynamic Host Configuration Protocol. It gives IP addresses to devices on a network through a server, which in this lab is a router leasing out addresses. Each time a device connects, it will ask the DHCP server for an IP address, and the DHCP server will provide one until all available IP addresses are used up. In a DHCP starvation attack, an attacker sends many fake requests to the DHCP server to lease out all the IP addresses of the server. Once the server runs out of addresses, devices within the network that actually need IP addresses cannot connect to the network, and thus users cannot connect to the internet or internal systems. Work will be significantly delayed due to this, and additionally, attackers may introduce a new, fake DHCP server to redirect or intercept user traffic, having complete control and access over the data.

ARP stands for Address Resolution Protocol. It helps devices find each other on the network using MAC addresses. For example, if a device wants to send data to IP 192.168.1.1, it sends an ARP request asking, “Who has 192.168.1.1?” The device with that IP replies with its MAC address, and the requesting device stores this information in its ARP cache and sends data to that MAC address. In ARP spoofing, an attacker sends fake ARP replies, falsely claiming that it has the requested MAC address. The victim updates its ARP cache with the attacker’s MAC address, so when the victim then tries to send data to that IP address, the data is sent to the attacker’s MAC address instead of the intended address.

Lastly, MAC flooding involves MAC addresses, which are unique IDs assigned to network devices. Switches have a table of these addresses to be able to “switch” properly, or direct network traffic to the correct device. In a MAC flooding attack, an attacker sends lots of fake MAC addresses to the switch to overload the switch’s memory. When this happens, the switch is unable to distinguish between devices and as a result, sends traffic to all devices on the network.

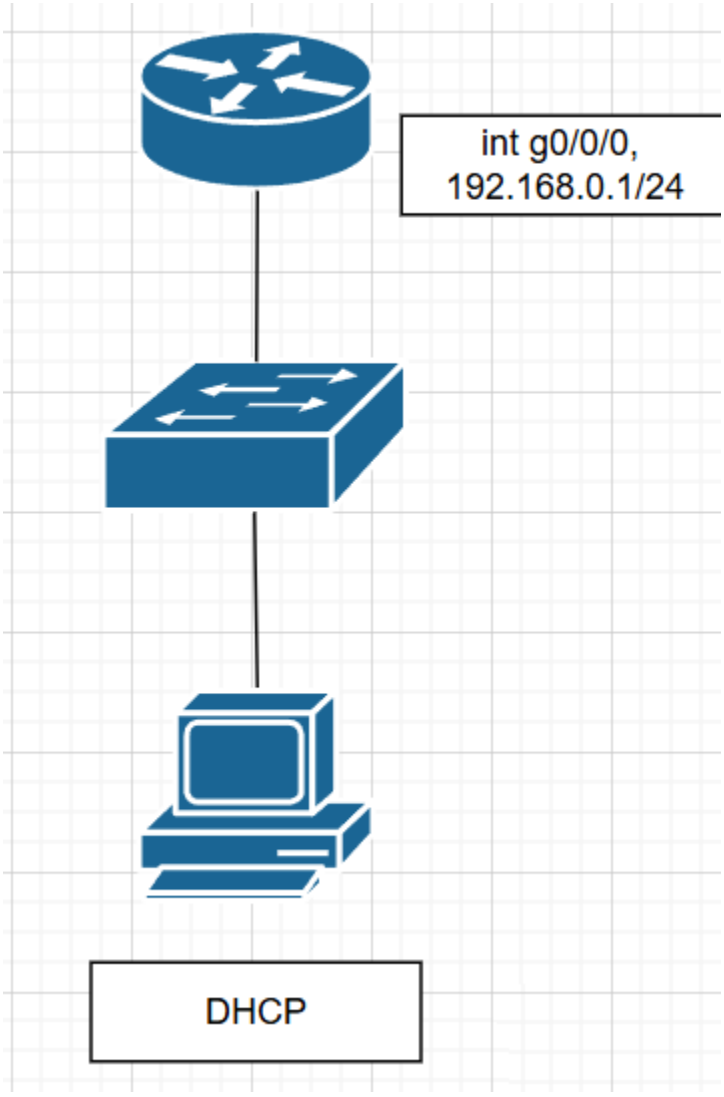
Thus, it is important to defend against any internal network security threats. Attacks from inside the network can potentially see confidential data, change data, or entirely prevent data from reaching the intended address. Lastly, note that although DHCP starvation, ARP snooping, and MAC flooding are common attacks, they are not all of the possible attacks that may act on a network.

LAB SUMMARY:

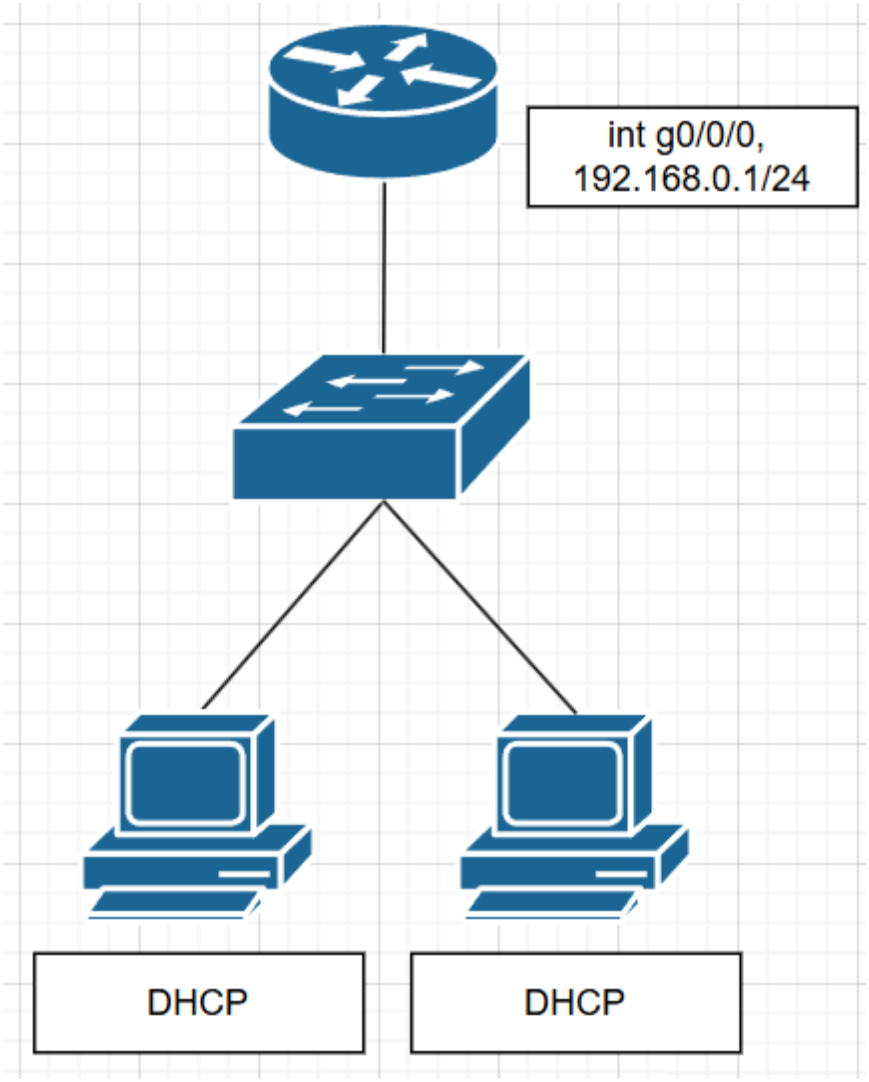
Attacked a simple router-on-a-stick network through three methods: DHCP starvation, ARP snooping, and MAC flooding. Proceeded to mitigate all three attacks.

NETWORK DIAGRAMS:

For DHCP starvation and MAC flooding:



For ARP snooping:



LAB COMMANDS:

Configure DHCP with the following commands on the router in global configuration mode. Enter the command *ip dhcp pool vlan[VLAN_number]* for each VLAN that is not the native VLAN. Configure the pool of DHCP addresses using the command *network [ip address] [subnet mask]*, *default-router [IP address of the router]*, and *dns-server 8.8.8.8 8.8.8.4*. Afterwards, configure the IP DHCP excluded addresses, the addresses that should not be leased out with DHCP, using the command *ip dhcp excluded-address [address 1] [address 2]*.

Set up a virtual machine on a computer and go into its terminal. Use the *sudo yersinia -G* command to open up a software to run attacks on the network. Make sure the computer has an Ethernet connection to the switch and thus is in the same network as the router and host. Then, find the attack function and select DHCP, then check the box next to DHCP discover before running the attack.

Verify that on the PC, which should be set with an IP address obtained through DHCP, is not able to obtain an IP address anymore after the attack. See below:

```
Router#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/              Lease expiration        Type       State      Interface
                Hardware address/
                User name
192.168.0.6     0104.d9c8.ba25.74       Jun 14 2025 05:12 PM    Automatic  Active     GigabitEthernet0/0/0
192.168.0.7     0148.2ae3.8eee.3f       Jun 14 2025 05:15 PM    Automatic  Active     GigabitEthernet0/0/0
192.168.0.8     76df.4a38.1e54          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.9     08e1.b642.b67d          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.10    709d.af4a.6a0d          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.11    8a80.8079.13c8          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.12    76e1.0700.3695          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.13    50db.ad29.b5d0          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.14    e216.535a.e61e          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.15    6838.421a.83cd          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.16    e66f.364e.6bfa          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.17    70a6.373c.386f          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.18    7a30.7b37.6706          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.19    c821.264b.15d0          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.20    b01f.7978.830e          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.21    ca4b.487a.b728          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.22    92c8.a90c.47b7          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.23    46ec.0c36.3921          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.24    be73.a03b.f8d1          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.25    867f.d919.fb9d          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.26    16e2.2137.077c          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.27    8459.af33.2e81          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.28    7ce0.c770.3f16          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.29    86e9.ad07.3c71          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.30    caaf.a530.d1b8          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.31    b266.7132.3ae5          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.32    86e6.0558.76bb          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.33    827d.f108.ea6d          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.34    320f.460b.cea6          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.35    da03.0531.44a5          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.36    82e4.0d0b.4760          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.37    b6fc.0c2c.e6b5          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.38    bae2.6944.3d5c          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.39    649c.595f.ec47          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.40    1ad7.2528.2d09          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.41    304e.cb61.e5dd          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.42    de78.387f.63a6          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.43    1e8a.8340.a53c          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.44    eacb.b20a.bf9f          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.45    76d5.7a0f.fd6f          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.46    7a4f.ea53.d3df          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.47    b8b2.a246.d02b          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.48    767b.f817.2922          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.49    88b3.4370.50a7          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.50    a087.3d5d.6d6e          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.51    2419.1464.88d7          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.52    3e78.ba6b.3edd          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.53    44b8.6903.b39e          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.54    9ebc.647c.bfe7          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.55    760d.384b.f39d          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.56    bab5.257d.a025          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.57    feac.7329.c8e3          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.58    d68c.e951.15af          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.59    6c4d.ef07.b454          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.60    dee1.fa37.cd4f          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.61    d0e1.d564.8bb6          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
192.168.0.62    22f6.975d.f625          Jun 13 2025 05:18 PM    Automatic  Selecting  GigabitEthernet0/0/0
```

Then, to mitigate, enter the following commands on the switch. *int range f0/1-24* to access all of the FastEthernet interfaces of the switch; *switchport and switchport mode access* in case any interface is not able to connect to end devices because it's in trunking mode; *switchport port-security* and *switchport port-security max 1* to actually secure the switchports and make sure unwanted traffic is never allowed.

Next, we run an ARP attack. First, connect a second host to the switch, as shown in the second network diagram. Configure the second host's IP address to be obtained by DHCP as well. Then, run in the terminal of the attacker's computer *sudo arpspoof -I enp3s0 -t [default gateway] [host_IP].* Then, wait a few seconds before pinging from the host computer to the default gateway (IP address of the router). On the terminal of the attacker's computer, run the command *sudo wireshark* to open Wireshark and capture ICMP packets.

To mitigate the ARP attack, run the below commands:

ip dhcp snoop vlan 1-10
no ip dhcp snoop info opt
ip dhcp snoop

ip arp inspect vlan 1-10
ip arp inspec vali ip
ip arp gratui none

int range f0/1-23
ip arp inspection trust
ip dhcp snooping trust

int f0/24
ip arp inspection limit none
no ip dhcp snooping trust

Note that interface f0/24 is the one connected to the attacker's laptop.

After mitigation, ping from the host computer to the router again, making sure that there are no ICMP packets appearing on Wireshark, meaning that the attacker is unable to intercept packets.

Finally, we simulate a MAC flooding attack. Disconnect the second host from the switch—there is no need for it for this segment—and run the command *sudo macof -1 enp3s0* on the attacker's computer. Run the following command on the switch to show the MAC address table: *show mac address-table*. There should be thousands of MAC addresses on the table. Then, run the command *clear mac address-table dynamic* and proceed with the mitigation.

The commands are same as the ones used for DHCP starvation mitigation: *int range f0/1-24* to access all of the FastEthernet interfaces of the switch; *switchport and switchport mode access* in case any interface is not able to connect to end devices because it's in trunking mode; *switchport port-security* and *switchport port-security max 1* to actually secure the switchports and make sure unwanted traffic is never allowed.

When viewing the MAC address table once more, there should be less than 10 addresses leased out, significantly less than before, the rest having been blocked by port security.

CONFIGURATIONS:

Note that certain output has been removed for clarity and relevance.

Hostname R1—show running configuration:

```
ip dhcp excluded-address 192.168.0.1 192.168.0.5

ip dhcp pool 1
network 192.168.0.0 255.255.255.0
default-router 192.168.0.1
dns-server 8.8.8.8

interface GigabitEthernet0/0/0
ip address 192.168.0.1 255.255.255.0
negotiation auto
```

No configuration was done on the switch, except during mitigation of each attack.

PROBLEMS:

I had a few problems getting the MAC flooding attack to work because the switch had built-in security protocols that protected it from such an attack. I included troubleshooting methods like waiting longer before stopping the attack to see if the MAC address would be more flooded and unplugging the port connected to the host to see if the host's MAC address would be unable to be seen in the address table. However, after running the command *sudo macof -1 enp3s0* instead of simply *sudo macof*, the attack worked right away.

CONCLUSION:

This was a fairly simple lab. The commands were not too complicated and there was not a lot of problems I encountered. However, I'm still proud of myself for finishing the lab very efficiently while still having a complete understanding of how attacks work and more importantly, how to mitigate them.

TEACHER SIGN-OFF