CCNP Portfolio

Blizzard, Harrison J
6/18/2025

Table of Contents

9/10/2024

Windows Write Up

Lab 1

Blizzard, Harrison J

Purpose

The purpose of this lab was to set up and configure personal installs of windows 11 education for the new CCNP class of the 2024-2025 school year.

Background Information/Lab Concepts

In this lab, multiple important applications were installed on my Windows 11 education edition install. This edition was made for educational institutions and students, offering additional features compared to the standard Windows 11 Home or Pro editions. It includes advanced security tools for protecting sensitive information, such as enhanced encryption and privacy controls, as well as IT management tools that simplify the configuration and maintenance of multiple devices across a network. These features are particularly valuable in educational settings where managing and securing many devices is essential.

Office 365, a suite of productivity applications developed by Microsoft, includes popular tools such as Word, Excel, PowerPoint, and Outlook. These applications are good for creating documents and managing data, which can be used for daily tasks and professional work.

PuTTY is a good application used to connect to other computers or network devices. It can be used for device management and configuration of servers and network equipment remotely through ssh or telnet. It is great for controlling network devices and performing administrative tasks without being physically there unless you want to console into it.

Wireshark is a tool that captures and examines data packets traveling through a network. This tool can help people diagnose network issues, monitor traffic, and ensure network security and performance by providing insights into the data flow within the network.

Lenovo Commercial Vantage is an application made for Lenovo computers. It helps manage and optimize system performances by offering updates to things such as firmware, , and customization options. This tool is essential for maintaining the computer’s efficiency and ensuring reliable operation in network management and other tasks.

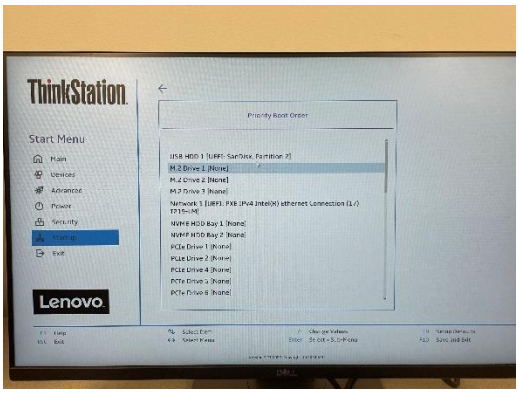
Lab Summary

First I took my SSD and inserted into the computer, then I took the flash drive with the image of win 11 and plugged that in as well, then I turned on the desktop and spammed f12 until I was in the bios where I changed the boot order to boot first through the flash drive. When it booted up, I then selected my language and selected windows 11 education edition and once it was done I signed into it, once I was on the desktop I immediately downloaded Mozilla Firefox because google and Microsoft have enough of my data and will never taste it in this lab, I then downloaded Wireshark which would be helpful in later labs. I also installed PuTTY which will also be needed in future labs, and which is used to console, SSH, or telnet into certain devices on

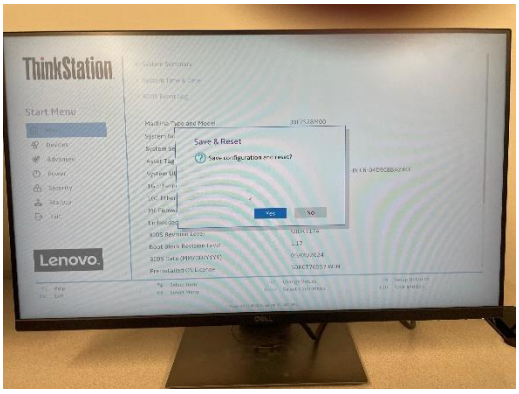
the same network. I also then logged into my personal Microsoft account into office 365 to have them, then I installed all the essential 365 apps like word, ppt, etc. After, I went into the Microsoft store and installed Lenovo Commercial Vantage, then after I went to windows update in settings and installed all the latest updates, once that was done and I restarted, I went to Lenovo Commercial Vantage to update my bios which it installed and when it was finish I was set up and ready for the next labs we will start to do in the future.

Lab Commands

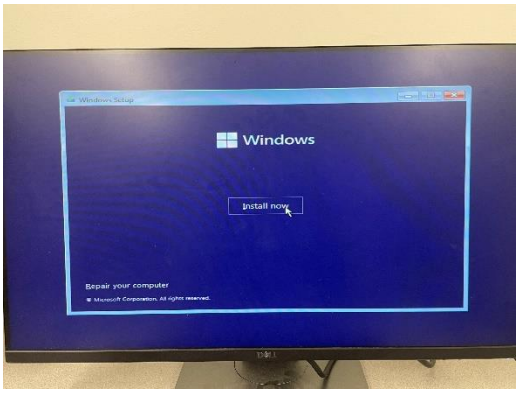
I first plugged in the USB Flash Drive and turned on the computer to open bios



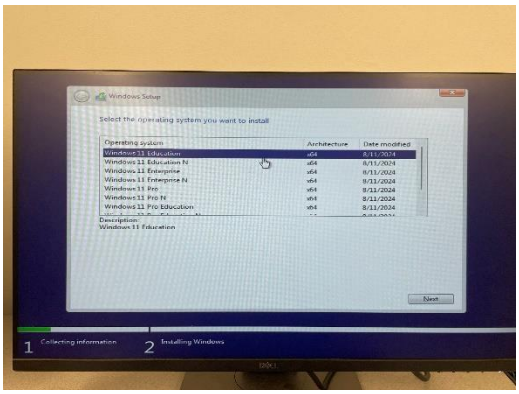
First I changed the boot order to start from the Flash drive and Saved the bios Config and rebooted.



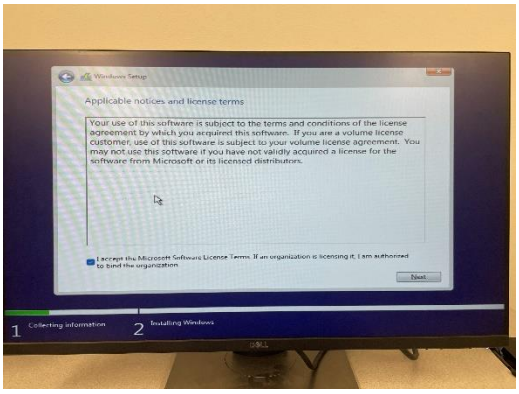
I had all the standard languages set to English.



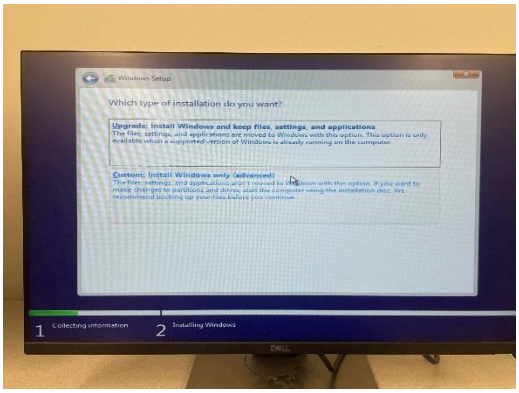
I hit install



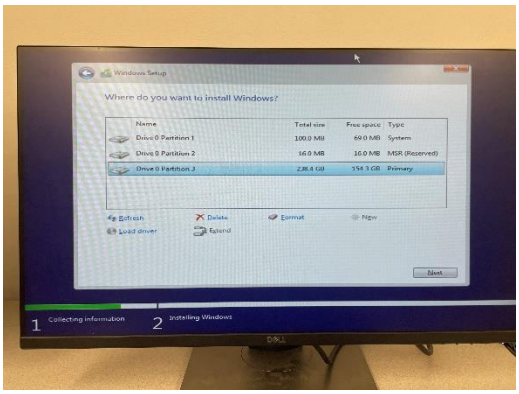
I was then into the windows install window and selected windows 11 education edition



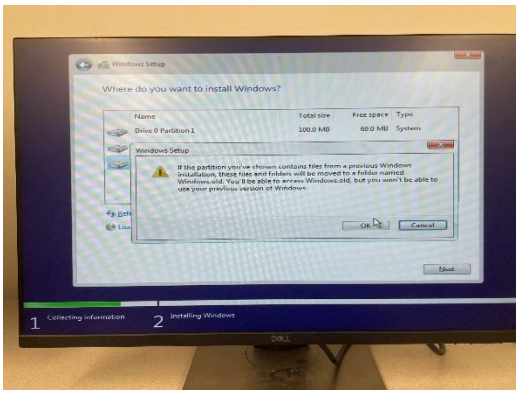
Hit next



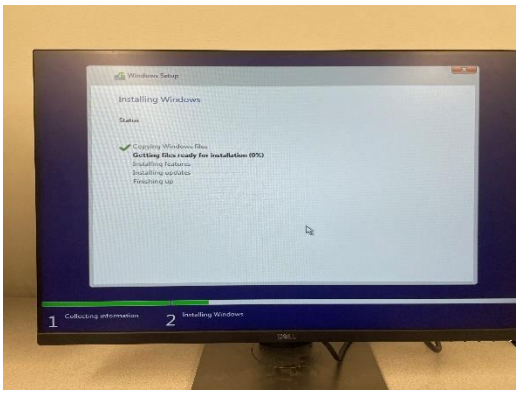
I hit custom install



I selected the partition with the most total size

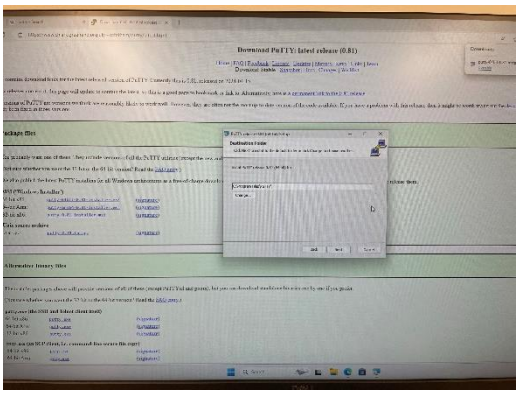


Formatted and erased the partition

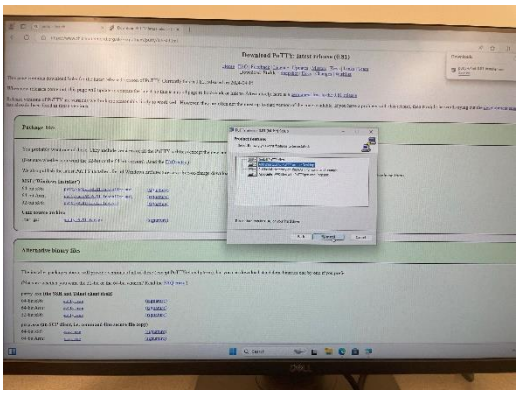


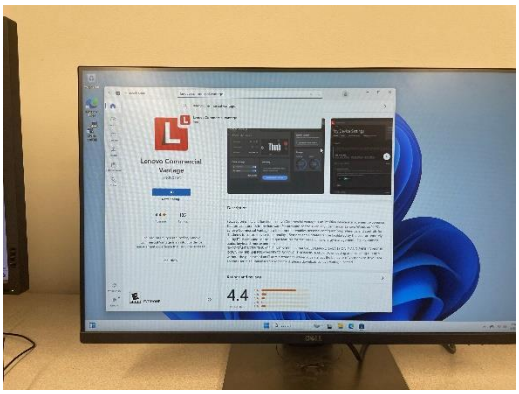
Wait for installation to finish

Windows fully booted up and I was able to now install all the software I needed

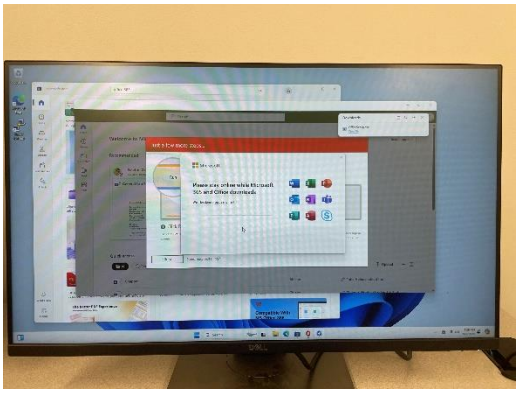


I went to the putty website and installed well, putty

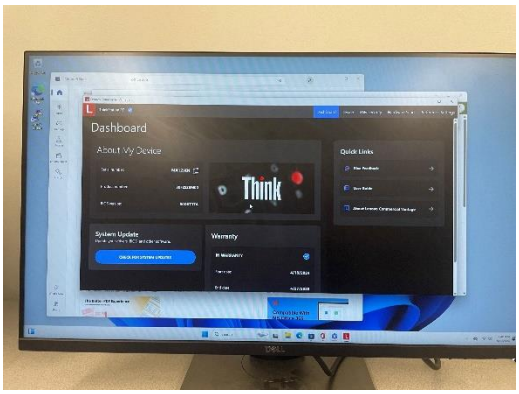


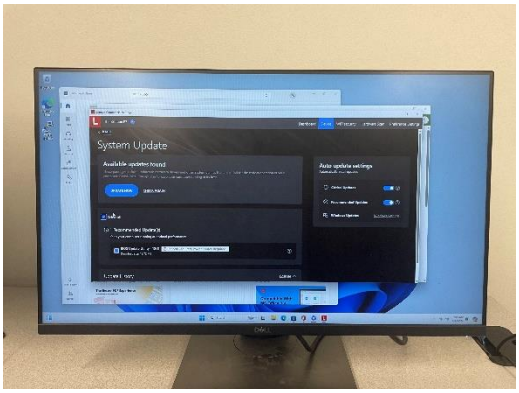


I then went to the Microsoft store to download Lenovo commercial vantage because it would be easier then using an install wizard

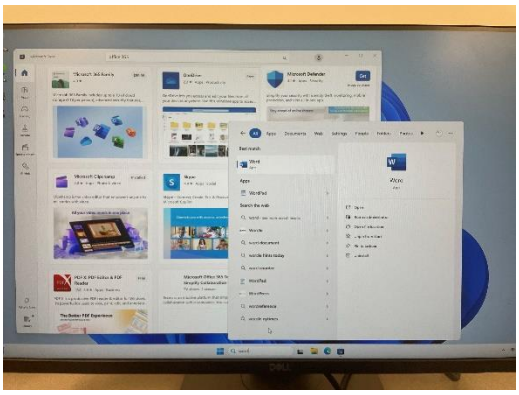


I also installed office 365 which started downloading

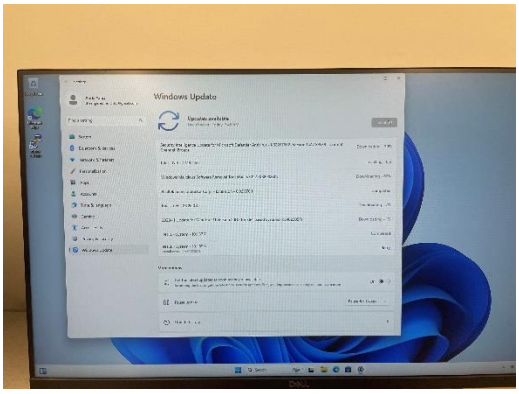




I hit update and then when it was done I restarted to update the bios



Office had finished installing



I then completed all the windows updates available

After I then installed firefox because edge sucks

Problems

The only couple of problems I had was with the boot order, I forgot to change it so I went back into the bios and put the SSD at the top so it would boot into it first. And one of the windows updates would not download or install no matter what, it was something to do with intel.

Conclusion

In my opinion this lab will be helpful to setup everything we need for future labs plus some customization and a tad of freedom and was a good way to start off the year. (Most of this document had to be rewritten because it was lost, and it got Auto Recovered)

9/29/2024

OSPF Write-Up

Lab 2

Blizzard, Harrison J

Purpose

The purpose of this lab was to reinform and to review the concepts and commands of the OSPF routing protocol for future labs and to get us in the sense of what later labs will be like.

Background Information/Lab Concepts

OSPF:

OSPF, or Open shortest path first is a routing protocol used by many. It's an Interior Gateway Protocol that's commonly used in large enterprise networks. OSPF is based on the Shortest Path First algorithm, which uses graph theory to find the shortest path between nodes in a network.

Layer 3 Switch:

A Layer 3 switch is different from a layer 2 switch by combining the functions of a switch and a router, layer 3 switches are great and compact because instead of having two network devices, switch and router, it combines them both into one where you can change individual ports to either be routing ports or switching ports.

Lab Summary

For this lab we started off by setting up each of the computers to use specific static ip addresses, we then turned off the firewalls and plugged them into their corresponding routers, after we then consoled into each router and began adding Ip addresses and adding ospf and using other commands, we then consoled into the layer 3 switch where we set it up to route and use ospf as well.

Lab Commands

R1:

Building configuration...

Current configuration : 4043 bytes

```
Last configuration change at 23:52:53 UTC Tue Sep 17 2024

version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core

hostname R1

boot-start-marker
boot-end-marker



vrf definition Mgmt-intf

 address-family ipv4
 exit-address-family

 address-family ipv6
 exit-address-family



no aaa new-model
```

login on-success log

subscriber templating

vtp domain cisco

vtp mode transparent

ipv6 unicast-routing

multilink bundle-name authenticated

```
crypto pki trustpoint TP-self-signed-2189345785
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2189345785
 revocation-check none
 rsakeypair TP-self-signed-2189345785


crypto pki certificate chain TP-self-signed-2189345785
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32313839 33343537 3835301E 170D3234 30393137 31383131
  33395A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 31383933
  34353738 35308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100B735 4EE15E99 BC8CE83C D1B73820 3E0BAA9D 88FF44A3 051FD0D6
  E2961C59 9200771E 215E8FBA EDB8A8B9 CCC35AF0 34819A51 81B37C4D 26925338
  C9F36FC8 27F6A707 56BF490A 6865A9A2 F33127B3 8E7FB7FD D910AE27 8068C356
  FD717FA9 1B441622 31CE9A92 335BE2EC 6E43B317 2EECF4A0 B24E1D23 0AD760A1
  CDCC125C 52AB5CD4 56518416 57FB0057 5EB49790 C6478508 5ABBF9B9 1E17378F
  9775E505 68785B36 5BF50CB9 6E2FCCB5 478C1BB2 5B46B826 CBE7722C AAA370D8
  201DD1C2 D0B26DD2 5AAEF890 91D77964 F2FFF4E3 801BFDED 9C7BD44F 07CABCDB
  E89530A9 49724372 EA373A08 CFE24DFC 3AA5390D D85C15C9 6C591632 0B00E8F8
```

ABE30B05 30010203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14BD6309 E506C7CC D1EFC799 D372A2B8 7EB5F9FD
5F301D06 03551D0E 04160414 BD6309E5 06C7CCD1 EFC799D3 72A2B87E B5F9FD5F
300D0609 2A864886 F70D0101 05050003 82010100 09642D9D 5CE0998F 9F97F23F
200E7E82 CFBD1F81 F93CE174 540CD20D 842D1ECA B93DBD11 56F3CACF AF2A801B
6BDBFDAA E8AA477F EC705723 5F2C7C9F 78684873 F26A748F E070E4E3 E14330F7
3A1AB10F 1671FBE2 F60ECC4D FEEB8B27 3A1E9860 C570613E 3445489B 4DFB1B1B
AC89DD75 2B0AA853 5A4E1F94 419C39C0 B30AEB4A 86A9486D 9C6A4870 7A7163ED
F62D2833 D3DFF59E 9B31F9BC 2055F995 4879C3C8 D6F07663 ECA252FE FD37975E
515BD6E5 B75EF5B7 C1F7C173 FA9908DE 04911CB9 88BE8D0D 3F628CF7 EAECA606
1E821AE0 62FE5CFE 52F2B4E6 15D09B51 4F7F2E6E 31C8F99A 75CDB588 09E15C3A
76470202 CCE61BA1 1CBC0C15 E11CADBA 839F33BE

quit

license udi pid ISR4321/K9 sn FDO21482DXE

license boot level appxk9

no license smart enable

diagnostic bootup level minimal

spanning-tree extend system-id

redundancy

mode none

```
interface GigabitEthernet0/0/0
 ip address 192.168.0.3 255.255.255.0
 negotiation auto
 ipv6 address 2001:DB8:ACAD:1::3/64
 ipv6 ospf 1 area 1

interface GigabitEthernet0/0/1
 ip address 192.168.2.1 255.255.255.0
 negotiation auto
 ipv6 address 2001:DB8:ACAD:3::1/64
 ipv6 ospf 1 area 1

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 1.1.1.1
 network 192.168.0.0 0.0.0.255 area 1
 network 192.168.2.0 0.0.0.255 area 1

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
```

```
ip tftp source-interface GigabitEthernet0



ipv6 router ospf 1
 router-id 1.1.1.1




control-plane



line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
```

End

R2:

Building configuration...

Current configuration : 3993 bytes

Last configuration change at 22:49:37 UTC Tue Sep 17 2024

version 16.9

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

platform punt-keepalive disable-kernel-core

hostname R2

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

login on-success log

subscriber templating

```
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated



crypto pki trustpoint TP-self-signed-2557841031
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2557841031
 revocation-check none
 rsakeypair TP-self-signed-2557841031


crypto pki certificate chain TP-self-signed-2557841031
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32353537 38343130 3331301E 170D3234 30393137 31373438
  35345A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 35353738
  34313033 31308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100BB8C 65EDEDC7 D75E0E69 797EB9ED 17C2496F D7F12267 04CA008E
```

4E067E31 6F95837F 5BC8B395 BABA53EB 8AD931AC A8A59EE3 636275F4 2E8DC79A
0EC5B225 4B33D74E 4A0D2A69 A110B2EC 262E6DD8 DBAC37A0 F924A287 E2FBDA07
A068A8D6 C22E8153 BAEE46D9 59776838 D5F268E6 21F6E4CB 9F359DD2 5CC2DF87
C2AB50F0 24834C14 4CAFFE6C 4BF5BD84 E5F85FCB FC1AB96A EFC260E1 463B4242
9F35AD23 617D3280 FFBAA3FA 53B983E6 CAABB854 A7A18B13 AFA10581 88F053F1
5A4A81DD A5D4A238 AA1AB08C C3312234 FED7DE63 54515775 C504CDBF B5D88E12
F7AC59A5 0372ADC4 BED701C5 E3F9D15B 996D616B 50DA011F 736343AD 3773C054
27FE6242 0D2F0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14223169 B873BBA0 14184051 07CFDD3D 87D1FDC9
5C301D06 03551D0E 04160414 223169B8 73BBA014 18405107 CFDD3D87 D1FDC95C
300D0609 2A864886 F70D0101 05050003 82010100 44C6DE61 D418E499 5F08BBEC
F95A58AF E497C771 1B8CD616 556CCF6F E20F70E9 9B682D53 329F8B54 93E0E00D
7CC2D100 764E1DC6 D2677BA8 958F5B81 6BF7FE9E 9FFAE690 19C4E435 C197B824
AC4F0215 AB919DD2 BEE87E95 913EF489 B4710FF1 3C1B29EC 06913B80 D8FBE47C
5483EB5E 5C42B5C1 3A87A9B4 A58C5611 B01934AC AD270ACE B889BEF1 FA1D7C22
7BF2CE45 53D3FBB2 2C32FA74 4DEDAECB 0D7ADECE 0C7A7FEA 2D4926C1 4D28FE23
74BE9E57 96F937EF 95B8D809 341324B5 14EDC053 E1B3D91D A4C581B0 4D17872F
F5C4675F BAFC1127 0C5A339C 897308E3 630A907D E35A50D1 328E7FC2 41682A16
4E158937 630A0EC6 A1731298 8B5F2A89 45345885

quit

license udi pid ISR4321/K9 sn FDO21500G1N

license boot level appxk9

no license smart enable

diagnostic bootup level minimal

```
spanning-tree extend system-id

redundancy
 mode none

interface GigabitEthernet0/0/0
 ip address 192.168.3.1 255.255.255.0
 negotiation auto
 ipv6 address 2001:DB8:ACAD:4::1/64
 ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
 ip address 192.168.2.2 255.255.255.0
```

```
 negotiation auto
 ipv6 address 2001:DB8:ACAD:3::2/64
 ipv6 ospf 1 area 1

interface Serial0/1/0

interface Serial0/1/1

interface GigabitEthernet0/2/0
 negotiation auto

interface GigabitEthernet0/2/1
 negotiation auto

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

router ospf 1
 router-id 2.2.2.2
 network 192.168.2.0 0.0.0.255 area 1
 network 192.168.3.0 0.0.0.255 area 0

ip forward-protocol nd
```

```
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0


ipv6 router ospf 1
 router-id 2.2.2.2



control-plane


line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
```

End

R3:

Building configuration...

Current configuration : 3818 bytes

Last configuration change at 23:10:27 UTC Tue Sep 17 2024

version 16.9

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

platform punt-keepalive disable-kernel-core

hostname R3

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

login on-success log

subscriber templating

vtp domain cisco

vtp mode transparent

ipv6 unicast-routing

multilink bundle-name authenticated

```
crypto pki trustpoint TP-self-signed-318861592
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-318861592
 revocation-check none
 rsakeypair TP-self-signed-318861592


crypto pki certificate chain TP-self-signed-318861592
 certificate self-signed 01
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33313838 36313539 32301E17 0D323430 39313731 38313535
  375A170D 33303031 30313030 30303030 5A303031 2E302C06 03550403 1325494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3331 38383631
  35393230 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 B8DE2383 C4CE9B87 1A982D45 655E9102 32FA0EA0 7C46D653 C277DD84
  E22AA44E E0C338C2 A079B689 3B4D6779 45B5A4F1 211630D3 88BDAE24 7F3A1618
  95135EEE 414CAC1C E7EF798E 2FB8FE13 768B5500 21AB669A B23AAA7E 0C32FB54
  CEDB30D5 BB09EF96 6F5707FB DDDB2B32 7073997D 80F5C5FE 50725F13 FBD0FB2B
  7A4BE1BA 190E16EE B2476BED E66FECD0 684E79E8 8B56694C 4B379B36 CEFEF213
  A6858367 A237BAC1 3A21C27A 25A609E8 CC92F21A EAA2FC11 9DB7445A C3F10322
  31C80EBD D7DE2F96 6FEC5CB9 D0E26FA9 605111FD F205D4D0 1FA466E4 3D6FF15D
  C2152F88 FB4750F5 15841845 B102AE63 E92B027E 3E27FDF6 DDA870B9 63422563
  C4114AB1 02030100 01A35330 51300F06 03551D13 0101FF04 05300301 01FF301F
```

0603551D 23041830 168014A2 B67F26FD 34496FA3 C134B9F8 83433194 67334030
1D060355 1D0E0416 0414A2B6 7F26FD34 496FA3C1 34B9F883 43319467 3340300D
06092A86 4886F70D 01010505 00038201 01006351 4C00BC47 606A6056 EAE09994
EC2F3C1E 40985F68 8F75419F 9DED110B 061C2388 43E6D9AD 2A0BD939 42476A9A
FD816DBB BBAFA627 06255C8D 1375147C BFC2B0A6 AF7FDDCB E418F829 B6A2114B
9581829E D229DC9C 3514F6D7 D7236147 E81D621B 9B443EE5 4258F5E2 0F586C2B
FEB2760B 13478669 03D2161B 97F0D660 A1638159 2CE8C0D0 9F9DBB4C 5B3BECA4
0B093C50 DDD4C020 299E3B49 2CAFA9EA 4996455C CAFE5B29 6B6CC781 EE177E26
21ADCF64 B2181B99 93DE5751 797DD6C6 D155BAC5 23F419B3 F8544D29 88603C64
161698BE 7C1B9D1B 6BA34086 035C6C3F C645BA8A D22F79DA EA97BF79 6E0EADB7
2432AC20 A25763B7 A9D2B106 4EAE2756 ED79

quit

license udi pid ISR4321/K9 sn FDO214420HM

no license smart enable

diagnostic bootup level minimal

spanning-tree extend system-id

redundancy

mode none

```
interface GigabitEthernet0/0/0
 ip address 192.168.3.2 255.255.255.0
 negotiation auto
 ipv6 address 2001:DB8:ACAD:4::2/64
 ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
 ip address 192.168.4.1 255.255.255.0
 negotiation auto
 ipv6 address 2001:DB8:5::1/64
 ipv6 ospf 1 area 2

interface Serial0/1/0

interface Serial0/1/1

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

```
 negotiation auto

router ospf 1
 router-id 3.3.3.3
 network 192.168.3.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 2

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0


ipv6 router ospf 1
 router-id 3.3.3.3




control-plane


line con 0
 transport input none
```

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

End

R5:

Building configuration...

Current configuration : 3907 bytes

Last configuration change at 23:36:51 UTC Tue Sep 17 2024

version 16.9

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

platform punt-keepalive disable-kernel-core

```
hostname R5

boot-start-marker
boot-end-marker


vrf definition Mgmt-intf

 address-family ipv4
 exit-address-family

 address-family ipv6
 exit-address-family


no aaa new-model



login on-success log
```

subscriber templating

vtp domain cisco

vtp mode transparent

ipv6 unicast-routing

multilink bundle-name authenticated

crypto pki trustpoint TP-self-signed-4153289585

enrollment selfsigned

subject-name cn=IOS-Self-Signed-Certificate-4153289585

revocation-check none

rsakeypair TP-self-signed-4153289585

crypto pki certificate chain TP-self-signed-4153289585

certificate self-signed 01

30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030

31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274

69666963 6174652D 34313533 32383935 3835301E 170D3234 30393137 31383135

30355A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649

4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D34 31353332

38393538 35308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201

0A028201 0100B3CC CCCB307D EC0C608A 987EAB0E 702BA29D EC303C1A 90FC5823

F48648AC C589FCB4 7F0ECBB7 CCF6A7DE 897F9CFF A6B7689E 7A349C16 A4CD481E
7235B4D8 0CB96D33 653A14C0 7B7DC206 A7AB7215 6088150A A0DD7107 5767FFE5
7C400DA3 0B403720 7C3B1B46 F8F88F13 2FF76B1C 223110A5 8922CB2A C7481A9E
FF4DAD1D C53E85CF 02310033 470D2ECE 8C272C37 B6DD4033 279C9784 D351830F
FB2CEF86 B8C07A6D 8F7DD9CE 94FD563B 4A979A10 44170032 D27053A7 DE7FEF0F
CD33AF9C 08866BA9 42BF4E9A 9A667E4C 4D66CB27 DE3C4CF1 0D78D64A 1714DC78
8D52BF77 418A96DC A5FB0F1D BC3F46F7 C20EE8A8 727399A3 4E5676EF F378412C
4039D6FB 95CF0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14525744 B7AFCFE1 3AE799B2 4FB6F56A D5C2E17D
5B301D06 03551D0E 04160414 525744B7 AFCFE13A E799B24F B6F56AD5 C2E17D5B
300D0609 2A864886 F70D0101 05050003 82010100 4EAF7BD8 A1A711D1 7B47D9E5
320896C7 A2E614DA B7FBAEB4 A91E1A07 81E30D3F 82C2A480 C92CC3E3 2ED49C8D
A1F5866D E76E8C96 7B566AC1 3EBD815B F508B7D9 3CA015FB B835CC83 6177915A
B4B9EABD 900E153F 18A701B7 2EE2930D 406A29BF DC97245C 3210E518 DE6318C9
4A0E8C6D 32C29EBD 9C9805A6 911A5BC3 DA756580 920CAF03 5F480DA3 2B1411A1
F5C30F7C 969B638C 0C2E265F 2C58DD90 C35DEC6A CD7F7585 6FCF0ABE D4CF195B
AC2B46A4 341E3938 A13AD8D3 0CC31E23 B1D795DA 433F136D 284294B7 10D35919
85ECC583 3B8A6855 6F4AD31E 6FEFBBD3 EF23848B 38BCF4BF 81DB75ED D283DCEC
DDE3D187 ABE864F3 A3C99811 09C1495D F7B2C9C3

quit

license udi pid ISR4321/K9 sn FDO214417Q4

no license smart enable

diagnostic bootup level minimal

spanning-tree extend system-id

```
redundancy
 mode none
```

```
interface GigabitEthernet0/0/0
 ip address 192.168.5.2 255.255.255.0
 negotiation auto
 ipv6 address 2001:DB8:ACAD:6::2/64
 ipv6 enable
 ipv6 ospf 1 area 2

interface GigabitEthernet0/0/1
 ip address 192.168.6.1 255.255.255.0
 negotiation auto
 ipv6 address 2001:DB8:ACAD:7::1/64
 ipv6 enable
```

```
 ipv6 ospf 1 area 2

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 5.5.5.5
 network 192.168.5.0 0.0.0.255 area 2
 network 192.168.6.0 0.0.0.255 area 2

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
```

```
ipv6 router ospf 1
 router-id 5.5.5.5




control-plane



line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login




End
```

R6:

Building configuration...

Current configuration : 1538 bytes

Last configuration change at 18:58:39 UTC Tue Sep 17 2024

version 16.9

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

platform punt-keepalive disable-kernel-core

hostname R6

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

login on-success log

subscriber templating

ipv6 unicast-routing

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO215009QY

no license smart enable

```
diagnostic bootup level minimal

spanning-tree extend system-id




redundancy
 mode none




interface GigabitEthernet0/0/0
 ip address 192.168.5.1 255.255.255.0
 negotiation auto
 ipv6 enable
 ipv6 ospf 1 area 2

interface GigabitEthernet0/0/1
 ip address 192.168.4.2 255.255.255.0
 negotiation auto
```

```
 ipv6 enable
 ipv6 ospf 1 area 2

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

router ospf 1
 router-id 6.6.6.6
 network 192.168.4.0 0.0.0.255 area 2
 network 192.168.5.0 0.0.0.255 area 2

ip forward-protocol nd
no ip http server
ip http secure-server
ip tftp source-interface GigabitEthernet0


ipv6 router ospf 1
```

router-id 4.4.4.4

control-plane

line con 0

transport input none

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

End

S1:

Building configuration...

Current configuration : 2560 bytes

version 12.2

no service pad

service timestamps debug uptime

service timestamps log uptime

no service password-encryption

hostname LAYER3SWITCH

boot-start-marker

boot-end-marker

no aaa new-model

system mtu routing 1500

ip routing

ipv6 unicast-routing

```
spanning-tree mode pvst
spanning-tree extend system-id

vlan internal allocation policy ascending



interface FastEthernet0/1
 switchport access vlan 10
 switchport mode access

interface FastEthernet0/2
 switchport access vlan 10
 switchport mode access

interface FastEthernet0/3

interface FastEthernet0/4

interface FastEthernet0/5

interface FastEthernet0/6
```

```
interface FastEthernet0/7
 switchport mode access

interface FastEthernet0/8

interface FastEthernet0/9

interface FastEthernet0/10

interface FastEthernet0/11

interface FastEthernet0/12

interface FastEthernet0/13

interface FastEthernet0/14

interface FastEthernet0/15

interface FastEthernet0/16

interface FastEthernet0/17

interface FastEthernet0/18

interface FastEthernet0/19
```

interface FastEthernet0/20

interface FastEthernet0/21

interface FastEthernet0/22

interface FastEthernet0/23

interface FastEthernet0/24

interface FastEthernet0/25

interface FastEthernet0/26

interface FastEthernet0/27

interface FastEthernet0/28

interface FastEthernet0/29

interface FastEthernet0/30

interface FastEthernet0/31

interface FastEthernet0/32

interface FastEthernet0/33

interface FastEthernet0/34

interface FastEthernet0/35

interface FastEthernet0/36

interface FastEthernet0/37

interface FastEthernet0/38

interface FastEthernet0/39

interface FastEthernet0/40

interface FastEthernet0/41

interface FastEthernet0/42

interface FastEthernet0/43

interface FastEthernet0/44

interface FastEthernet0/45

```
interface FastEthernet0/46

interface FastEthernet0/47

interface FastEthernet0/48

interface GigabitEthernet0/1

interface GigabitEthernet0/2

interface GigabitEthernet0/3

interface GigabitEthernet0/4

interface Vlan1
 ip address 192.168.10.2 255.255.255.0

interface Vlan10
 ip address 192.168.0.1 255.255.255.0
 ipv6 address 2001:DB8:ACAD:1::1/64
 ipv6 ospf 1 area 1

router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.0.255 area 1
```

```
ip classless
ip http server
ip http secure-server


ipv6 router ospf 1
 router-id 10.10.10.10
 log-adjacency-changes


tftp-server flash:c3560-ipservicesk9-mz.SE7
tftp-server flash:c3560-ipservicesk9-mz.SE7 alias ipservices


line con 0
line vty 0 4
 login
line vty 5 15
 login

End
```

Extra General Commands Concepts:

Pings

TraceRoute

ip ospf neighbors (v4 and v6)

ip ospf interface

topology

ip route tables (v4 and v6)

Anything special

- command "enable ipv6" on vlan on switch

- ipv6 routing in global config on switch

- ipv6 unicast-routing in global config on switch

- switchport mode access and switchport access vlan x on switch interfaces

- do ipv6 ospf x area y on any interface that is connected through OSPFv3

- set router-id for all ospf (v2 and v3)

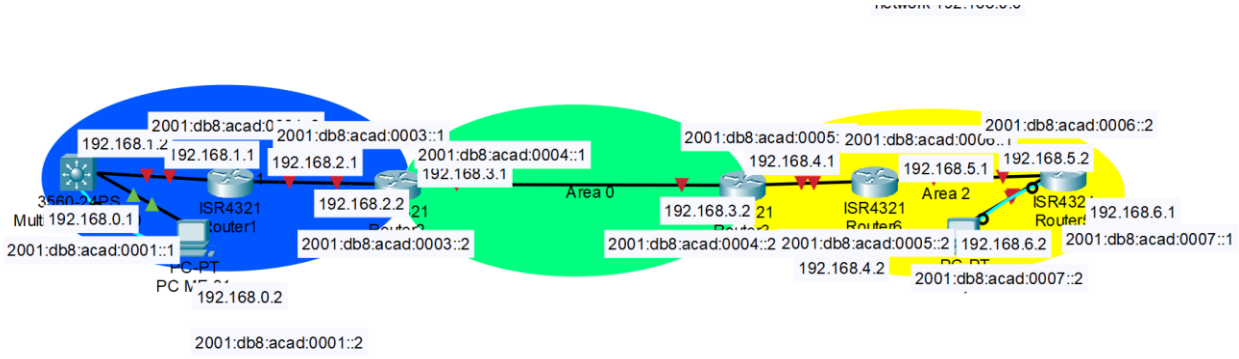
- router ospf (x) to enable OSPF on routers/switch\

- network statements inside the ospf interface to ensure that all addresses are advertised to everywhere through OSPF

- set ip addresses through "ip add (xxx.xxx.xxx.xxx xxx.xxx.xxx.xxx)" or "ipv6 add (xxxx::x/xx)"

- use "ipv6 router ospf" to enter/enable OSPFv3 (ipv6) and set ipv6 network statements for them as well.

Lab Diagram



Problems

We couldn't get the routers to route to the switch and ping any interface on it.

Conclusion

In conclusion I think this lab was great review and a good start to what future labs will be like and the amount of work we must complete then.

12/29/2024

eBGP Write-Up

Blizzard, Harrison J

Purpose

This lab's purpose was focused on how to teach us to correctly use and configure exterior Border Gateway Protocol to allow many different network protocols like OSPF and EIGRP to communicate. As communication between separate network types can be invaluable in networking jobs this lab was very useful.

Background Information/Lab Concepts

eBGP, or External Border Gateway Protocol, is a routing protocol that enables different autonomous systems (ASes) to exchange routing information. It ensures that networks can communicate beyond their own boundaries by sharing the necessary routing data to reach external destinations. As the most widely used exterior routing protocol, eBGP plays a key role in keeping the internet connected.

Autonomous systems (ASes) are large, interconnected networks that make up the backbone of the internet. Data moves across the internet by hopping from one AS to another until it reaches its destination. Each AS consists of one or more IP address prefixes managed by a single administrative entity, such as an enterprise, university, or service provider. These ASes operate under a defined routing policy—using protocols like OSPF, EIGRP, IS-IS, or RIP—which they present to the internet. The agreements between ASes to exchange routing information are known as "peering." Each AS is assigned a unique autonomous system number (ASN) by Regional Internet Registries (RIRs), which helps facilitate routing between networks. In essence, ASes are the foundation of internet communication, ensuring data flows smoothly between networks.

However, ASes alone aren’t enough to keep the internet running. They need a way to communicate with each other. That’s where eBGP comes in. eBGP enables ASes to share routing information by establishing sessions between routers at the edges of their respective networks, known as border or edge routers. These routers act as eBGP peers, exchanging route updates to ensure data can reach external destinations. The challenge with eBGP is that it must be carefully configured with the correct parameters, including authentication methods, neighbor addresses, and AS numbers. Unlike iBGP, which operates within a single AS, eBGP is specifically designed for communication between separate networks.

Different routing protocols handle information exchange in their own ways, which is why eBGP is necessary to connect them. OSPF, for example, calculates the shortest path through a network using an algorithm and shares that data with neighboring routers. EIGRP, on the other hand, uses its DUAL algorithm to establish loop-free routes and quickly adapt to changes, providing backup paths in case of failure. IS-IS, a protocol from the 1980s, uses

hello packets to detect topology changes and relies on a shortest path first (SPF) algorithm combined with a link-state protocol to determine the best routes. Since these protocols operate within separate networks, they cannot natively share data with each other. This is where eBGP is essential, as it acts as the bridge between different ASes and their routing policies.

Lab Summary

For this lab, we began by subnetting three IPv4 and three IPv6 networks. The setup included six routers in total, with two routers per network, forming three distinct networks. Each network operated under its own internal routing protocol, while eBGP was responsible for transferring data between them.

We started by configuring one router per network. The first step was assigning the correct IP addresses to each router's interfaces based on our topology. In the first network, we implemented OSPF between the two routers. The middle network was configured with EIGRP, and the final network used IS-IS, which required several adjustments to function properly.

Once the internal routing protocols were set up, we configured eBGP between the networks. This involved creating address families and redistributing neighbors as needed for each protocol. During this process, modifying the IS-IS routing level and adjusting its metric were critical for ensuring proper connectivity.

Finally, we assigned IPv4 and IPv6 addresses to the PCs connected to the edge routers of the outer networks, following the subnetting plan. After disabling firewalls, we successfully pinged across the entire network, confirming that eBGP was properly implemented and allowing full communication between all routers on both IPv4 and IPv6. This validated that my partner and I had successfully set up eBGP for cross-network communication.

Lab Commands

interface loopback0

Creates and enters configuration mode for the loopback0 interface.

ip address [ip-address] [subnet-mask]

Assigns an IP address to the loopback0 interface.

router bgp [autonomous-system-number]

Creates and enters configuration mode for a BGP instance with the specified autonomous system number (terminal configuration mode).

neighbor [ip-address] remote-as [autonomous-system-number]

Establishes a BGP neighbor relationship. For iBGP, the remote autonomous system number must match the local AS.

neighbor [ip-address] update-source loopback0

Configures the router to use its loopback interface as the source address for BGP packets, improving session stability.

ip route [destination-ip] [subnet-mask] [next-hop-ip]

Creates a static route to ensure the reachability of BGP peer loopback addresses (if not using OSPF or EIGRP).

network [ip-address] mask [subnet-mask]

Advertises a specific network to other BGP peers. The network must be in the routing table to be advertised.

neighbor [ip-address] route-reflector-client (optional)

Configures a neighbor as a route reflector client if using a route reflector to reduce the full mesh requirement in iBGP.

router ospf [process-id]

Enables OSPF for IGP reachability (if using OSPF as the underlying protocol).

network [ip-address] [wildcard-mask] area [area-id]

Defines OSPF networks and assigns them to a specific area.

redistribute [protocol] [autonomous-system-number] [metric {metric value}] [level]

Redistributes IPv6 routes from a specified protocol and assigns a metric.

show ip bgp summary

Displays a summary of BGP peers, including state and established sessions.

show ip bgp

Shows the BGP routing table with learned and advertised routes.

ping [ip-address] source loopback0

Verifies reachability of the neighbor’s loopback address using the local loopback as the source.

Configurations

R1 EBGP-

Current configuration : 4163 bytes

Last configuration change at 19:04:55 UTC Tue Jan 28 2025

version 16.9

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

platform punt-keepalive disable-kernel-core

hostname R1

boot-start-marker

boot-end-marker

```
vrf definition Mgmt-intf
address-family ipv4
exit-address-family
address-family ipv6
exit-address-family

no aaa new-model

no ip domain lookup

login on-success log

subscriber templating

vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated

crypto pki trustpoint TP-self-signed-2189345785
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-2189345785
revocation-check none
rsakeypair TP-self-signed-2189345785

crypto pki certificate chain TP-self-signed-2189345785
certificate self-signed 01
```

30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 32313839 33343537 3835301E 170D3235 30313238 31383339
31325A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 31383933
34353738 35308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100DD5F 81FAD8F6 F71B48B4 D3C9EF9B EEC4E341 4A12F5F5 287FAF81
CFD65AF9 C85F83D2 36B34516 41897177 7ECFA446 C92DB6B5 1289B20B 9331B07D
2B861ADA F98D97EB 05F17648 EB657CC7 5E8BB360 E22AA13F E7BCE6E9 41E17054
6D8DF124 305B03A4 AB7BD6F3 05F91ABF CFB15ADE B01C9B61 B8BF235D A8A0EE67
9831559F 64C719DF 328BA3A3 73F942EE 8EC2208C 07B148FE 979D079F 9E41CBF1
3E7613CD B0A68F31 D902FC0F 990D0438 210474BE 8FB75338 0FFA7E4E 16039066
046EA8B8 AEBD29FD 47201FF7 AAF53C35 B5D20FB3 A9293B95 4FFADEB1 39F72A40
29ACD579 1BF22771 96E06287 B03940FB B960E978 1B148950 F65D13B5 12B9A035
4C581DB6 98CF0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 143207B8 925C52C7 4247A369 831D7739 3EB0B6F6
80301D06 03551D0E 04160414 3207B892 5C52C742 47A36983 1D77393E B0B6F680
300D0609 2A864886 F70D0101 05050003 82010100 5220A5B9 78BA22A0 C855C02C
B69EAA47 026C0E88 676140F0 E4364E3F CB4CBA87 1B9A064A 1C190ACD FE40942D
322AAC49 578167DC 7E6C0CC7 FF9F363C B68FE8A2 377BCEE6 5AA8A2CB 06464449
41876C56 B7B53E77 0B4FC400 57E6C301 7BBE32A0 D57C78C1 3550F32B 6A52EC42
B8B22EF1 A4C262B1 CFE12747 7221C8F5 48698396 412B2D45 D0B6A350 9B5283CC
7A6EB1EF 2043905B DC62DD12 C8DB369E 50B5D376 448DC398 C33AEBA1 639AC1E3
695B058B 9715CF8B 046EA47A 9AB1CF68 D2216F15 A855A0B4 A5CDFCE8 A476F2E6
B78D106C C3A3DEC5 98DE143B A20C09A6 8A3461BA CC5B0F3D 4EAB7982 B0CBEAA7
F2C71A3C E138CD71 F51E2DD9 938B59E7 41C209FD
quit

```
license udi pid ISR4321/K9 sn FDO21482DXE
license boot level appxk9
no license smart enable
diagnostic bootup level minimal

spanning-tree extend system-id

redundancy
mode none

interface GigabitEthernet0/0/0
ip address 10.0.2.1 255.255.255.0
negotiation auto
ipv6 address FD02::1/64
ipv6 eigrp 1

interface GigabitEthernet0/0/1
ip address 10.0.1.1 255.255.255.0
negotiation auto
ipv6 address FD01::1/64
ipv6 eigrp 1

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

router eigrp 1
default-metric 10000 100 255 1 1500
network 10.0.2.0 0.0.0.255
redistribute connected
eigrp router-id 1.1.1.1

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
```

ip tftp source-interface GigabitEthernet0

ipv6 router eigrp 1
eigrp router-id 1.1.1.1
redistribute connected
default-metric 10000 100 255 1 1500

control-plane

line con 0
transport input none
stopbits 1

line aux 0
stopbits 1

line vty 0 4
login

end
```
R2 EBGP-
Current configuration : 4566 bytes

Last configuration change at 18:57:50 UTC Tue Jan 28 2025

version 16.9
service timestamps debug datetime msec
```

```
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core

hostname R2

boot-start-marker
boot-end-marker

vrf definition Mgmt-intf

 address-family ipv4
 exit-address-family

 address-family ipv6
 exit-address-family

no aaa new-model

login on-success log

subscriber templating

vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
```

```
crypto pki trustpoint TP-self-signed-2557841031
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2557841031
 revocation-check none
 rsakeypair TP-self-signed-2557841031

crypto pki certificate chain TP-self-signed-2557841031
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32353537 38343130 3331301E 170D3235 30313238 31383139
  30315A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 35353738
  34313033 31308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100DFAF 716405BF 540A7BEF 32E89F06 D70B5651 1B1DBBE9 30B7E7C1
  90452399 E6DE7436 69EFC00D F3FA6313 70CF1060 921D5301 92675195 B0787662
  913B9C6A 3EE0E42A 0C98FF36 AC355C0E B4E90D4E A1863630 1746BE48 35E7B32C
  85751069 A23B849E E1776709 E2CD52A0 3E1BDC3A F4169C45 72E7B338 F560F0C4
  A87A0080 FF557ADA FB599DDE EC6BFE9C B3458F13 DDE97947 637C31B3 E17582E9
  3CCD4D7F 54F2273D A8C69CCA 57AB08CA F8C3C530 7672E835 F9E6F9CC 61E9E8A3
  0CE36F4B D5D24FFB 845E66FD 82F2F70A 313C7F8E 2EAD0450 B6ABAAAD E8B3343E
  9A47317E 5A3844B1 B4D0604B 92995576 05B1FCEF 71A226C9 4E7BE71F 5006E360
  E7E281FF 52050203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
  301F0603 551D2304 18301680 1444BBF5 6693AA1E 6FF24478 4B433A52 BEC40CDE
  72301D06 03551D0E 04160414 44BBF566 93AA1E6F F244784B 433A52BE C40CDE72
  300D0609 2A864886 F70D0101 05050003 82010100 35FA9583 636A2AC7 0148209A
  27C9F03D D82FEEF4 AC19A04E DA174CFC B643D55C 125428A9 D88CDD23 0F6795A7
  E6964328 D1AFF508 6A045C4B 39B24C36 4D269AED 173BB690 874D0C8C 76B6A419
```

```
 5D61B0BC B7BB18BE 842B55D7 065A6CAB 31BE2B9C EA8DFA61 486249A6 ED108608
 8BFDE8E5 D531DB5D A2380D01 B480F271 E979FE0C 8CE794D2 965322E2 03469A01
 3B520354 5C9276F6 8B2EC202 899261D4 B905C75D 2AAB989B 3D55DF2B AB4E093F
 005AFC3C 89B9D1CE DA2681A5 B5E5CC64 91AD209D 10A98E3D 24DCF10E 7F266900
 129C412B FEB05C0C 8602A9BF A9CDF3EE 785A19F8 A423AA96 76F8F6E9 4480A6B8
 BB8B9958 504AA170 5CE2104A 6439F414 513A8C41
      quit

license udi pid ISR4321/K9 sn FDO21500G1N
license boot level appxk9
no license smart enable
diagnostic bootup level minimal

spanning-tree extend system-id

redundancy
 mode none

interface GigabitEthernet0/0/0
 ip address 10.0.2.2 255.255.255.0
 negotiation auto
 ipv6 address FD02::2/64
 ipv6 eigrp 1

interface GigabitEthernet0/0/1
 ip address 10.0.3.1 255.255.255.0
 negotiation auto
 ipv6 address FD03::1/64
```

```
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

router eigrp 1
 default-metric 10000 100 255 1 1500
 network 10.0.2.0 0.0.0.255
```

```
 redistribute connected
 redistribute bgp 1
 redistribute ospf 1
 eigrp router-id 2.2.2.2

router bgp 1
 bgp log-neighbor-changes
 neighbor 10.0.3.2 remote-as 2
 neighbor FD03::2 remote-as 2

 address-family ipv4
  redistribute connected
  redistribute eigrp 1
  neighbor 10.0.3.2 activate
  no neighbor FD03::2 activate
 exit-address-family

 address-family ipv6
  redistribute connected
  redistribute eigrp 1
  neighbor FD03::2 activate
 exit-address-family

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
```

```
ipv6 router eigrp 1
 eigrp router-id 2.2.2.2
 redistribute connected
 redistribute bgp 1
 default-metric 10000 100 255 1 1500

control-plane

line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login

end
```

R3 EBGP-

```
Current configuration : 4472 bytes

Last configuration change at 19:26:50 UTC Tue Jan 28 2025

version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
```

```
hostname R3

boot-start-marker
boot-end-marker

vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family

no aaa new-model

login on-success log

subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated

crypto pki trustpoint TP-self-signed-318861592
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-318861592
 revocation-check none
 rsakeypair TP-self-signed-318861592
```

```
crypto pki certificate chain TP-self-signed-318861592
 certificate self-signed 01
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33313838 36313539 32301E17 0D323530 31323831 38343035
  385A170D 33303031 30313030 30303030 5A303031 2E302C06 03550403 1325494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3331 38383631
  35393230 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 C1052458 4E6DCCF3 8C3F7B8D E1E22B8A 84241348 7F9A55EF A5FBC8CB
  F078FC59 3DE812B5 1C6CDFE4 40C7CCEF F3BD1778 F1354632 0A23754C DF3D838D
  2A10734B A199FB2A 29EB17FF F99C3D7F 23FE21D3 070A570B 4FD8B79C 7DC46789
  1D7A405F 22AD2CF5 9E7FC51E 0F0F0D6B 38063AE6 9A98252C C7435C18 97B58FFC
  7DB2588D E7ABA128 FD0E627E 4CD80BB7 E626654B 18D4DB39 A8D44F7B 3D64F32E
  CADA5881 1FD34844 F7FFCC5F 28FF6B64 6EAEE724 99F532C6 E5745E93 B436C5C8
  A0DF18AE DEC241DF D2978C6A D6DD8136 C96C8D2B 040A5AA3 A01A9A2F 1E4BC33E
  8A0BC530 85B79D87 534404C2 065B86D0 D7AC1E19 86DC3601 02F570F8 B00B27A4
  0331D6F3 02030100 01A35330 51300F06 03551D13 0101FF04 05300301 01FF301F
  0603551D 23041830 168014C4 536047FA 2E15B421 E0E9959C 1752FAC3 640F7830
  1D060355 1D0E0416 0414C453 6047FA2E 15B421E0 E9959C17 52FAC364 0F78300D
  06092A86 4886F70D 01010505 00038201 0100901C 8E1DEDDA 4A573991 47EAF40E
  9B7F7954 350B7248 293815F3 60CD4C25 7F5AA53C 8C7C90F8 FFFD0BE4 2DCD9A59
  E64A2B78 418AE424 30CC630D 6F07627F 6C63EA51 D31BFF04 D7140365 A2C0F67C
  EBD7EDA8 4C9F7EB1 CDAD1BCE CCDF30CD 239ACB46 34AE145E 2DB31F9F 2B2F822E
  26B95B8A F5A2A893 03212178 94F3BB55 D7423F39 75F2109B 980BCAF6 C7FA0500
  8A7B75CB C178B3B0 0B2585E5 94CF1EBA 433A458C 7B65A500 2BA3B5E9 6C1B7DC5
  930B88B8 AFAFD537 9E9BDE6E BC51F44D 1B72B11E B7A67F52 D0B894F4 0A910F65
  6A1D2F4A 946566FD 1273E060 BA9F03F9 80722F8F F248E7D3 DEF216D6 BD0293CF
  C8696E67 40C62B87 1D305ADF 658BA383 742D
```

```
    quit

license udi pid ISR4321/K9 sn FDO214420HM
no license smart enable
diagnostic bootup level minimal

spanning-tree extend system-id

redundancy
 mode none

interface GigabitEthernet0/0/0
 ip address 10.0.4.1 255.255.255.0
 negotiation auto
 ipv6 address FD04::1/64
 ipv6 ospf 1 area 0

interface GigabitEthernet0/0/1
 ip address 10.0.3.2 255.255.255.0
 negotiation auto
 ipv6 address FD03::2/64

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

router ospf 1
 redistribute connected subnets
 redistribute bgp 2 subnets
 network 10.0.4.0 0.0.0.255 area 0

router bgp 2
 bgp log-neighbor-changes
 neighbor 10.0.3.1 remote-as 1
 neighbor FD03::1 remote-as 1
 address-family ipv4
```

```
 redistribute connected
 redistribute ospf 1 match internal external 1 external 2
 redistribute ospf 3
 neighbor 10.0.3.1 activate
 no neighbor FD03::1 activate
exit-address-family
address-family ipv6
 redistribute connected
 redistribute ospf 1 match internal external 1 external 2
 neighbor FD03::1 activate
exit-address-family

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0

ipv6 router ospf 1
 redistribute connected
 redistribute bgp 2

control-plane

line con 0
 transport input none
 stopbits 1
line aux 0
```

```
 stopbits 1
line vty 0 4
 login

end
```

R4 EBGP-

```
Current configuration : 4316 bytes
Last configuration change at 19:19:15 UTC Tue Jan 28 2025
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname R4
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
address-family ipv4
exit-address-family
address-family ipv6
exit-address-family
no aaa new-model
login on-success log
subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated
```

crypto pki trustpoint TP-self-signed-2240717686

enrollment selfsigned

subject-name cn=IOS-Self-Signed-Certificate-2240717686

revocation-check none

rsakeypair TP-self-signed-2240717686

crypto pki certificate chain TP-self-signed-2240717686

certificate self-signed 01

30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030

31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274

69666963 6174652D 32323430 37313736 3836301E 170D3235 30313238 31383334

30365A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649

4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 32343037

31373638 36308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201

0A028201 0100C71F 6328FE7B 5C56EAF2 E2F4A080 D9BB34DB CB388C86 4137B931

91EBEB83 9AEAB14B 8BB3AACC C5CEB490 0FBC3357 9DC5A1FB 6A3DEF34 CC93452A

B6F3DB7F F1CB7CAF 8883344F 611C25AB 3F5FF2A8 388F0EA8 1E5F4805 3BD4528F

40EFB4DF 8171E39D D0891673 38C9273E C8296F83 B56DC52C 5245D645 8C263B6F

C62FED46 5A5BF9A6 7252DEA6 B210E744 50CEE9A1 0AA63119 FDEA3EFE 33355A84

D210A649 B775B12D 2B95F1B3 627CBB85 C3702AAF 1926B1C3 986ECC50 D3D7A05F

BEDF417F 11C793B5 D8B50775 1B644F8B 533D3FE6 7F6E2E88 BD3AF456 38DBF2FD

7A1E91A2 B4F71FFE DDFAA96B 63B86D16 CE7AB91F 70CB7B9F 003C4D9E 0B65FD7A

75C15889 6ADD0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF

301F0603 551D2304 18301680 14ED3C31 4C52E0A2 5971CB17 EFBAEC96 863CEAD0

1E301D06 03551D0E 04160414 ED3C314C 52E0A259 71CB17EF BAEC9686 3CEAD01E

300D0609 2A864886 F70D0101 05050003 82010100 A03E40F3 D82A2F58 B12FA90C

194EA52F FD3D5E2D 823BDC9F C490AA55 95BD8A15 5957020D 59C91172 46563A0C

ECE58901 DF44693C D0F08188 196E29FC CE6E0060 F8E0A033 A3E34CFF F8912AB7

35ADEB21 BBAAE773 4CA6E267 1F050C71 52FD40F8 79054C8B E86E6C97 DF62400F

58B51B7A 965C5FC3 D868C749 A7F4EFB5 1B0502E9 1A16F2BC F14FD5DA 037A2A66
9D69F92A FB33D26E 340BEA69 A20AA489 82E4FF2D 059C2975 8FFC1535 0820433D
4C3913F9 43B36496 D63C8452 77CE8A41 3EC7C7EF CA215046 EA8D448F EF16EE4E
9EA8956C 06F57162 3FC11907 8AD0AD7B E0474B47 82DFBC2F 289D2868 50F266EA
0F59D608 444433D3 20F3ECBB A4536DAD D42284A0
quit
license udi pid ISR4321/K9 sn FDO214414DZ
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
mode none
interface GigabitEthernet0/0/0
ip address 10.0.4.2 255.255.255.0
negotiation auto
ipv6 address FD04::2/64
ipv6 ospf 1 area 0
interface GigabitEthernet0/0/1
ip address 10.0.6.1 255.255.255.0
negotiation auto
ipv6 address FD06::1/64
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 4.4.4.4
redistribute connected subnets
redistribute bgp 2 subnets
network 10.0.4.0 0.0.0.255 area 0
router bgp 2
bgp log-neighbor-changes
neighbor 10.0.6.2 remote-as 3
neighbor FD06::2 remote-as 3
address-family ipv4
redistribute connected
redistribute ospf 1 match external 1 external 2
neighbor 10.0.6.2 activate
no neighbor FD06::2 activate
exit-address-family
address-family ipv6
redistribute connected
redistribute ospf 1 match internal external 1 external 2
neighbor FD06::2 activate
exit-address-family
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
```

ip tftp source-interface GigabitEthernet0

ipv6 router ospf 1

redistribute connected

redistribute bgp 2

control-plane

line con 0

transport input none

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

R5 EBGP-

Current configuration : 4248 bytes

Last configuration change at 19:26:56 UTC Tue Jan 28 2025

version 16.9

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

platform punt-keepalive disable-kernel-core

hostname R5

boot-start-marker

boot-end-marker

```
vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family

no aaa new-model

login on-success log

subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated

crypto pki trustpoint TP-self-signed-4153289585
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-4153289585
 revocation-check none
 rsakeypair TP-self-signed-4153289585

crypto pki certificate chain TP-self-signed-4153289585
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 34313533 32383935 3835301E 170D3235 30313238 31383431
  35355A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
```

4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D34 31353332
38393538 35308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100BE92 ED7C92B4 DE919C7A B92F492D 9E1E8F59 D849081F 4FF06F63
0D06744D B8C5AAD7 6E71035C 2F757983 530B822F 88AD2632 9BF331F7 7ECFFCF4
0799DC6C D9FECCBA 4D93FFD2 5B969FC3 1127BE6B EC1B7D35 E631D67B 43D675D0
F7FBFEF0 A00F0E07 9ABE00B9 719921E1 E5C434A1 D3362FC2 2493B46F 6F25978C
A793DA3C D9B3900D A32B26E8 9461676B 0B0F68C2 7317FD29 A416F91F B037BC16
EC443AD3 97BBD50D 9D21CCEE 5F5DB765 55C62786 A20B613B EC2FD8CD 8452543A
E130A136 F2E70C49 A521555B 279FC4EF 1009DBFD 0C3C4008 5D8BD061 BB2282B8
5D9A31A7 C5463B45 9CADEF1D 64053CF5 6AE9D609 CF69E086 3DC7CADD D62E1B04
FF461B7F 00CB0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14F35E65 A7C099C8 34FE73C7 B33F5E06 ADCE56F6
14301D06 03551D0E 04160414 F35E65A7 C099C834 FE73C7B3 3F5E06AD CE56F614
300D0609 2A864886 F70D0101 05050003 82010100 6820AC24 A13A82AE CE9F2F28
A9EEB181 BC8C1C0E 399ECC5F 37684397 5D1F000D DE35939F CA4F417C D7EC2615
939792A9 3E5CE5EB 2678A2BB BAE64F23 2F358FD1 C18CF7C8 38FC4B14 289FF676
E9062FB5 BBEDEE08 A4259B1B B787A933 83AC24D7 18717A12 EFFCD7E1 D14D9947
F35A1C26 A30B0BA6 F37B4857 688E4D3C FA87BA35 2474977D 376933E4 1C3B5357
1638804E 1DE96B64 B4B08897 D2BFECD1 7B0B761F 536E81F6 329ED8C2 168E4751
50AB27FF 1E8E216C 1359241F F4283B7B 0084815F 6DC3E862 A243811E 6D87BB03
C82E62A7 84EBA956 7AB74BF8 BE3D00C1 4E09203C E054BAE2 4055CF76 1F0C60B3
A7D83610 BC291C5D 13F1C0CE 83FFCA64 984D6169

quit

license udi pid ISR4321/K9 sn FDO214417Q4

no license smart enable

diagnostic bootup level minimal

```
spanning-tree extend system-id

redundancy
 mode none

interface GigabitEthernet0/0/0
 ip address 10.0.7.1 255.255.255.0
 negotiation auto
 ipv6 address FD07::1/64
 ipv6 ospf 3 area 1

interface GigabitEthernet0/0/1
 ip address 10.0.6.2 255.255.255.0
 negotiation auto
 ipv6 address FD06::2/64

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 3
 router-id 5.5.5.5
 redistribute connected subnets
 redistribute bgp 3 subnets
 network 10.0.7.0 0.0.0.255 area 1

router bgp 3
 bgp log-neighbor-changes
 neighbor 10.0.6.1 remote-as 2
 neighbor FD06::1 remote-as 2
 address-family ipv4
  redistribute connected
  redistribute ospf 3
  neighbor 10.0.6.1 activate
  no neighbor FD06::1 activate
 exit-address-family
 address-family ipv6
  redistribute connected
  neighbor FD06::1 activate
 exit-address-family

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
```

```
ipv6 router ospf 3
 router-id 5.5.5.5
 redistribute connected
 redistribute bgp 3

control-plane

line con 0
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login

end
```

R7 EBGP-

```
Current configuration : 3884 bytes

Last configuration change at 19:31:17 UTC Tue Jan 28 2025

version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
```

```
hostname R7

boot-start-marker
boot-end-marker

vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family

no aaa new-model

login on-success log

subscriber templating
vtp domain cisco
vtp mode transparent
ipv6 unicast-routing
multilink bundle-name authenticated

crypto pki trustpoint TP-self-signed-1457377718
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1457377718
 revocation-check none
 rsakeypair TP-self-signed-1457377718

crypto pki certificate chain TP-self-signed-1457377718
```

certificate self-signed 01

30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 31343537 33373737 3138301E 170D3235 30313238 31383336
30395A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D31 34353733
37373731 38308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100BABE FFEA5372 979D4FEE F13A3FB6 F3993E37 9B9AE265 1A29E643
3239E552 FA0CC9C3 60329A2D 168FBA65 2444F57F 13C99D26 42AA7F65 41EFC678
81F02E9A 0AC7AE66 11099860 F76D1FB6 46AF7321 5C6B4823 CAD1383B 872B3785
29DD9793 A40E04AE B1124F11 FDCA4DFD 40D82FBE C880AF8E 6D64EC82 E6D9A049
D4EA323A 04147DC3 AE453B9B 2CE4FCF1 046A7F6B 0575B1AD A4AD0282 BAF10243
E275E70F 5AD06549 293B84E0 C446B771 0A6269CD 6B4CE9BB 4490CDCA 933C342E
4D136686 6C89D2C0 30EA7D27 92B04106 E7BA4F50 5F4A9849 7F9BC040 096E36B3
59207C79 F479B0C5 5007ACC1 5D7A471F 77E77578 4CBE9ACE E29B72AE C6331677
4A6E177B F01D0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14C4B11A C877535E A4FE41D2 DC8BD221 8238B172
4A301D06 03551D0E 04160414 C4B11AC8 77535EA4 FE41D2DC 8BD22182 38B1724A
300D0609 2A864886 F70D0101 05050003 82010100 1FB713EC 229FD6EF 707A6F14
302D6A72 8C74519D 8F370FA5 21279426 CC3F1F6B 8E59154E 004090DD 342EBBCB
2F58E826 A357D48D F845E7DD A95F5CA8 4994095D E47F3C00 2AF9CE01 76DEE460
0BFF7300 4B3C491B 011C4976 86810F90 E801D46B 9DFB89E2 EAEF1F2C C0210F20
E186860D E0B9E862 14351FCE B6CA5466 69433332 AD436879 31B3F528 FCC5DC68
2BBE927C 326486A1 26B348C6 E8335F3B FE9A1BD3 ED07776E A9C6E91A 4505B483
DD0F7518 91161217 ECC9E373 E5C68B1C A0C2F197 0B1622B5 AF7D0A65 C5ABA2F8
7F3193B7 D690C396 B9BB6657 700FA863 00EDB8BC 387CFA79 26A0738A 386DB62E
9B615BF3 5F79AF78 24FD568A 64629371 C9E559DD

quit

```
license udi pid ISR4321/K9 sn FDO21441WDF
no license smart enable
diagnostic bootup level minimal

spanning-tree extend system-id

redundancy
 mode none

interface GigabitEthernet0/0/0
 ip address 10.0.7.2 255.255.255.0
 negotiation auto
 ipv6 address FD07::2/64
 ipv6 ospf 3 area 1

interface GigabitEthernet0/0/1
 ip address 10.0.8.1 255.255.255.0
 negotiation auto
 ipv6 address FD08::1/64
 ipv6 ospf 3 area 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 ospf 3
 router-id 6.6.6.6
 redistribute connected subnets
 network 10.0.7.0 0.0.0.255 area 1
 network 10.0.8.0 0.0.0.255 area 1

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0

ipv6 router ospf 3
 redistribute connected

control-plane

line con 0
 transport input none
 stopbits 1
```

line aux 0

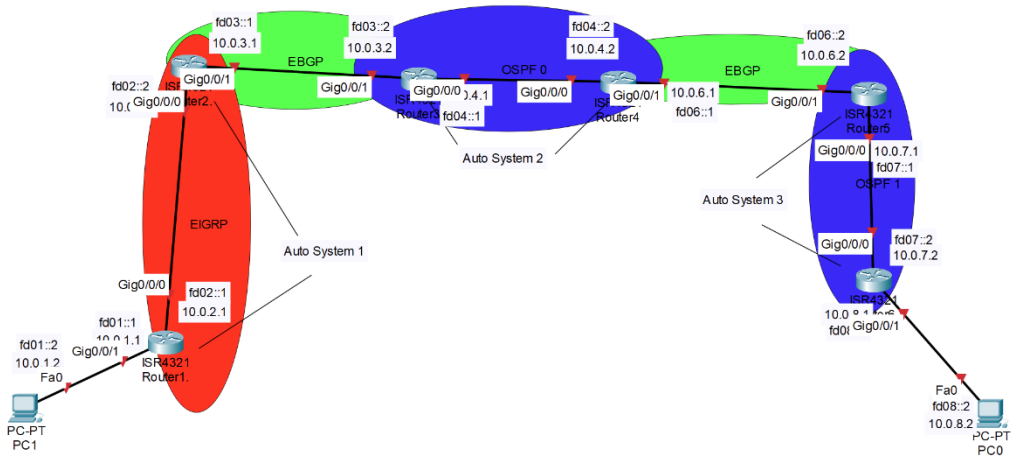
stopbits 1

line vty 0 4

login

end

Lab Diagram



Problems

No Major issues, but small issues with forgetting to put redistributes and external redistributes

Conclusion

eBGP is a highly valuable protocol, it is used to Interconnecting different network types is an inevitable aspect of networking, and eBGP provides the necessary solution. Ultimately, the knowledge gained from this lab will be invaluable moving forward.

12/31/2024

IBGP Write-Up

Blizzard, Harrison J

Purpose

The purpose of this lab was to learn how to configure interior Border Gateway Protocol to connect different network protocols allowing them to communicate with each other easily in an autonomous system. Communication between routers in this manner is essential and highly valuable in networking roles.

Background Information/Lab Concepts

iBGP:

iBGP, or Internal Border Gateway Protocol, is a routing protocol used within an Autonomous System to exchange routing information between routers. Unlike eBGP, which operates between different AS's, iBGP stays inside a single AS and works separately from other internal routing protocols like OSPF, EIGRP, or ISIS.

iBGP makes sure all routers in the AS know about external routes, keeping the network running efficiently and consistently. It's important when multiple routers exist inside the same AS since it allows for better routing and redundancy. iBGP works by setting up peering sessions between routers, ensuring routing information spreads across the AS. This prevents single points of failure and helps with load balancing.

One way iBGP avoids needing a full mesh of connections is with route reflectors, which help forward route information without every router needing to peer with every other router. This improves scalability while keeping the same benefits.

The big difference between iBGP and eBGP is where they operate. eBGP moves data between AS's, while iBGP stays inside one. iBGP routers need direct peering relationships, usually in a full mesh unless route reflectors are used. Unlike eBGP, iBGP doesn't automatically share routes between peers unless specific mechanisms are in place. It also keeps the next-hop attribute unchanged from the eBGP peer and runs over TCP.

Lab Summary

I started the lab with only three Auto systems based off what i configured with eBGP. I didn't use any host PC's and only configured and pinged using the seven routers that were connected to one another by ethernet cables. For connectivity to work between all the routers would need IPv4 and IPv6 routes to redistribute in and out of eBGP. But, in the second Auto system located midway in the topology between AS 1 and 3, iBGP is used to

provide network connectivity between the two ASs with EIGRP being used as the routing protocol for the left AS.

Lab Commands

interface loopback0

Creates and enters configuration mode for the loopback0 interface.

ip address [ip-address] [subnet-mask]

Assigns an IP address to the loopback0 interface.

router bgp [autonomous-system-number]

Creates and enters configuration mode for a BGP instance with the specified autonomous system number (terminal configuration mode).

neighbor [ip-address] remote-as [autonomous-system-number]

Establishes a BGP neighbor relationship. For iBGP, the remote autonomous system number must match the local AS.

neighbor [ip-address] update-source loopback0

Configures the router to use its loopback interface as the source address for BGP packets, improving session stability.

ip route [destination-ip] [subnet-mask] [next-hop-ip]

Creates a static route to ensure the reachability of BGP peer loopback addresses (if not using OSPF or EIGRP).

network [ip-address] mask [subnet-mask]

Advertises a specific network to other BGP peers. The network must be in the routing table to be advertised.

neighbor [ip-address] route-reflector-client (optional)

Configures a neighbor as a route reflector client if using a route reflector to reduce the full mesh requirement in iBGP.

router ospf [process-id]

Enables OSPF for IGP reachability (if using OSPF as the underlying protocol).

network [ip-address] [wildcard-mask] area [area-id]

Defines OSPF networks and assigns them to a specific area.

redistribute [protocol] [autonomous-system-number] [metric {metric value}] [level]

Redistributes IPv6 routes from a specified protocol and assigns a metric.

show ip bgp summary

Displays a summary of BGP peers, including state and established sessions.

show ip bgp

Shows the BGP routing table with learned and advertised routes.

ping [ip-address] source loopback0

Verifies reachability of the neighbor's loopback address using the local loopback as the source.

Configurations

R1:

Current configuration : 1602 bytes

Last configuration change at 16:41:46 UTC Fri Nov 8 2024

version 16.9

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

platform punt-keepalive disable-kernel-core

hostname R1

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

login on-success log

subscriber templating

ipv6 unicast-routing

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FLM2408005M

no license smart enable

diagnostic bootup level minimal

spanning-tree extend system-id

redundancy

mode none

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 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 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
ip forward-protocol nd
no ip http server
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router ospf 1
router-id 1.1.1.1
control-plane
line con 0
transport input none
stopbits 1
line aux 0
```

stopbits 1

line vty 0 4

login

end

R2:

Current configuration : 2033 bytes

Last configuration change at 16:30:40 UTC Fri Nov 8 2024

version 16.9

service timestamps debug datetime msec

service timestamps log datetime msec

platform qfp utilization monitor load 80

platform punt-keepalive disable-kernel-core

hostname R2

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

login on-success log

subscriber templating

ipv6 unicast-routing

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FLM24060912

```
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
mode none
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 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 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
ip forward-protocol nd
no ip http server
ip http secure-server
ipv6 router ospf 1
router-id 2.2.2.2
redistribute bgp 1 metric 10
control-plane
line con 0
transport input none
stopbits 1
```

line aux 0

stopbits 1

line vty 0 4

login

end

R3:

Current configuration : 2530 bytes

Last configuration change at 17:41:29 UTC Fri Nov 8 2024

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname R3

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ipv6 unicast-routing

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214420G7

spanning-tree extend system-id

redundancy

```
mode none
vlan internal allocation policy ascending
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
interface Vlan1
```

```
no ip address
shutdown
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
ip forward-protocol nd
no ip http server
no ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router eigrp 1
eigrp router-id 3.3.3.3
control-plane
line con 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
end
```

R4:

```
Current configuration : 2498 bytes
Last configuration change at 16:52:45 UTC Fri Nov 8 2024
```

```
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
hostname R4
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
address-family ipv4
exit-address-family
address-family ipv6
exit-address-family
no aaa new-model
ipv6 unicast-routing
subscriber templating
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO211216BL
spanning-tree extend system-id
redundancy
mode none
vlan internal allocation policy ascending
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
vrf forwarding Mgmt-intf
no ip address
negotiation auto
interface Vlan1
no ip address
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

ip forward-protocol nd
no ip http server
no ip http secure-server
```

ip tftp source-interface GigabitEthernet0

ipv6 router eigrp 1

eigrp router-id 4.4.4.4

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

R5:

Current configuration : 2042 bytes

Last configuration change at 16:13:11 UTC Fri Nov 8 2024

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname R5

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

```
ipv6 unicast-routing
subscriber templating
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FDO214421CF
spanning-tree extend system-id
redundancy
mode none
vlan internal allocation policy ascending
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 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 49.0012.0000.0000.0005.00
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
!ip forward-protocol nd
```

```
no ip http server
no ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
line con 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
end
```

R6:

```
Current configuration : 3901 bytes
Last configuration change at 16:47:28 UTC Fri Nov 8 2024
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname R6
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
address-family ipv4
exit-address-family
address-family ipv6
```

exit-address-family

no aaa new-model

login on-success log

subscriber templating

ipv6 unicast-routing

multilink bundle-name authenticated

crypto pki trustpoint TP-self-signed-2990358516

enrollment selfsigned

subject-name cn=IOS-Self-Signed-Certificate-2990358516

revocation-check none

rsakeypair TP-self-signed-2990358516

crypto pki certificate chain TP-self-signed-2990358516

certificate self-signed 01

30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030

31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274

69666963 6174652D 32393930 33353835 3136301E 170D3234 31313038 31363436

30325A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649

4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 39393033

35383531 36308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201

0A028201 0100B31A E32D972E FEF6E67C 743C4ECF DE52951C 3640258F 186152B4

19FBE7B1 7F1D7357 45D715AF ADB62226 FC876C99 AD601B63 41F55279 8823C3F3

84082004 F6698074 DE7489BA 09F948FB 0CAEBBC5 CB931924 C86E593C D112AC3F

DAE63118 8538F358 3472152C C8F83F8D A22797F9 A6A2B9EA EE7AE860 7A517E56

7F3823AB 2FBC06CC 536EA525 C17C5683 673258BD 8A8E7FD3 B4A7D98A E6830604

AD032515 3A050C89 90A1B980 201A51DB DBCB1522 BE49802F 8B15FA69 372CEA21

E067A65D 485A9DA2 507739E3 27B20661 80526F7B 61B46AB4 895DA115 4D708458

F8EDB9E8 E3AAA2E8 BFC69E74 06F16026 26744C84 23391ED4 C767B534 751D1DCA

C930D6B7 EC970203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF

301F0603 551D2304 18301680 14DA6337 3B53AEB7 55A6557A 33FD28AD 91961F8F
3C301D06 03551D0E 04160414 DA63373B 53AEB755 A6557A33 FD28AD91 961F8F3C
300D0609 2A864886 F70D0101 05050003 82010100 3731D78A A554756C 6AAADCE7
89D2E39C 620BC32F 3856B305 F322743C FEE21F44 47D5383F A1AB371F F799C996
4A738D8F FA76FDDD FC6BADCF 0B89D106 AB61B4E2 C09673F8 94960F54 C5F87DE3
28D2EB0F 347A2B0D 25AEAD49 EA201E70 8E12CCEB 31B4B3CC B37A2DDE DC45CCB2
24954FCC AAFFA407 E810FF2F 1A5EC2FC 5FBAB6F4 B6F8A310 D6B23D49 E0F0EB74
5F1C3256 4D597712 2C5FAAD6 608CE5A5 B20826EC 653024ED 60496725 8C532F65
A8B8EDBF AA03AEAF E3E52F3A B4AF5A03 98A20365 DD6C9682 CF62DEC7 A2D117EB
60662382 4120E1A3 657CCBB0 CA045830 BB833538 925F7BB6 D61B651F 56280B8E
3FA2AD05 A368C113 7498BE08 F74528A1 2ABE9D0F

quit

license udi pid ISR4321/K9 sn FDO21441WBL

no license smart enable

diagnostic bootup level minimal

spanning-tree extend system-id

redundancy

mode none

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
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 isis
net 49.0012.0000.0000.0006.00
is-type level-1
metric-style wide
log-adjacency-changes
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
```

```
line con 0
transport input none
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
end
```

R7:

```
Current configuration : 4714 bytes
Last configuration change at 16:50:12 UTC Fri Nov 8 2024
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname R7
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
address-family ipv4
exit-address-family
address-family ipv6
exit-address-family
no aaa new-model
login on-success log
subscriber templating
```

ipv6 unicast-routing

multilink bundle-name authenticated

crypto pki trustpoint TP-self-signed-3937601207

enrollment selfsigned

subject-name cn=IOS-Self-Signed-Certificate-3937601207

revocation-check none

rsakeypair TP-self-signed-3937601207

crypto pki certificate chain TP-self-signed-3937601207

certificate self-signed 01

30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 33393337 36303132 3037301E 170D3234 31313038 31363034
33335A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D33 39333736
30313230 37308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100C231 05E93E32 79526AA5 7164ED9B 45B5F3E9 606C1778 80895110
A9591B96 1947271C 0CB2F5AF 2F775A07 9959705C 2D3BB42E 1903298E 782E60D5
771FC717 5F4E8D1B 2CADED8C 1ED7D6ED 48953C64 6C4C8F8E 0A921180 3FE2EF6E
AE0EE4B8 70D7F2D0 EDDCB7BE CC6F6A14 9E5AAAAD 0DB20575 C390B782 086CDEF2
02CDBBFB AD902EB7 73384913 9EC727A8 E64E4ECB C9A3A3DE A373553C E9D4C64F
9156E9CE F959874B 054B6A81 77811DD4 2194351F AA282E10 ACF77962 D93562E8
DA914758 FA46F64F 94F3DAB1 F142DAE8 2ABAD7FC D42B03A6 92AF0A43 254B5C10
7DF42E74 CA2EEBFF ED22FFB2 3E16EAA9 9D4DD919 18D15567 E9E0F157 18D47571
D5FE7F01 A8650203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 148B780C 0425D9CD C7DD9370 3E5002F7 05AB6C2F
D6301D06 03551D0E 04160414 8B780C04 25D9CDC7 DD93703E 5002F705 AB6C2FD6
300D0609 2A864886 F70D0101 05050003 82010100 56F87742 64535A5E 5A8B6771
464815A8 BF7B4CCC F67C3994 77698096 47921092 FD1BA44F D503422D 70AD705F

```
 60963D1B AA52F043 5DA9ACE8 4BB1C544 D2A0AA29 661BE36B D6C33A39 AFA6EB50
 FB139D4D 5CDB357A 70A5534E E3AB84E5 5D79AC48 C856ADDC F595B98C E6976DBD
 77630747 7CAFCA25 909210B4 6F3C911D 4A7606BD D736BF9B 37032C8D F1FA099F
 B0B0D0F4 97AD13EA F144243A 01D8BAD0 D27E5643 BA62111C A236BF56 94621F29
 5D51C336 F60080C4 8BF25524 E3E64392 B4A1C7B7 A58A6C3F F1681670 6A2CC27B
 CA86AC47 15338D97 7B6B18D4 8B8EAABC C8AE9A9E 43F01B89 5B684014 C85F8315
 6F723084 AD9C74C5 D62A0091 F8D5A78E 2911BB3B
    quit
license udi pid ISR4321/K9 sn FDO214421BX
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
mode none
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
```

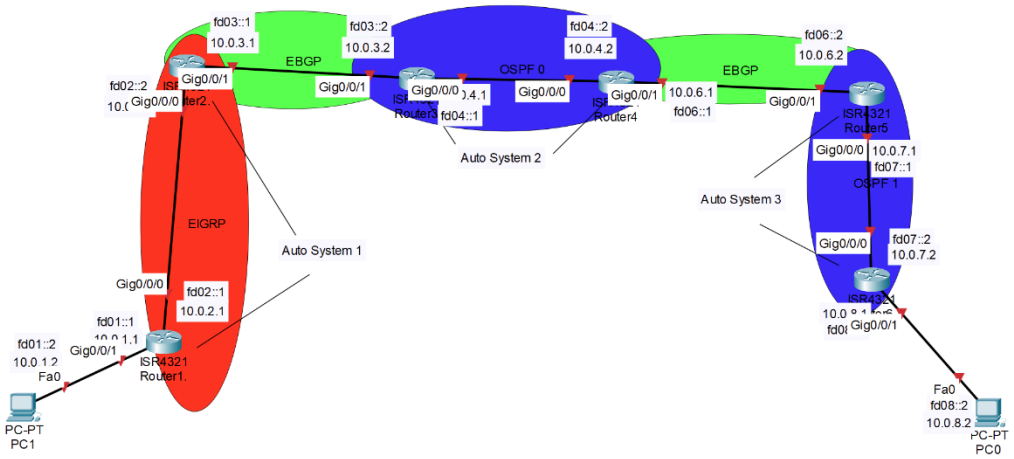
```
interface Serial0/1/0
no ip address
shutdown
interface Serial0/1/1
no ip address
shutdown
interface GigabitEthernet0
vrf forwarding Mgmt-intf
no ip address
shutdown
negotiation auto
router eigrp 1
network 7.7.7.7 0.0.0.0
network 10.0.1.0 0.0.0.255
network 10.0.2.0 0.0.0.255
eigrp router-id 7.7.7.7
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
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
ipv6 router eigrp 1
eigrp router-id 7.7.7.7
control-plane
line con 0
transport input none
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
```

end

Topology

(couldn't find my iBGP topology, I forgot to send it and packet tracer doesn't work on laptops so pretend theres an extra router with ips on it on subnet 10.0.5.0-10.0.5.2 and fd05::0)(also ignore PCs we never used them)



Problems

None that I remember clearly, only small problems that required 1-2 commands to fix

Conclusion

This lab provided an extremely valuable and unique skill for our time, demonstrating the ability to successfully implement advanced internal routing techniques for routers within an Autonomous System. By configuring peer relationships using loopback addresses and proper route advertising, we observed the scalability and stability advantages of iBGP. Additionally, the lab reinforced the importance of neighbor configurations in preventing routing loops. Overall, we enhanced our understanding of BGP and its impact on a configured network.

CCNP Portfolio
AWS 1-3 LABS

Blizzard, Harrison J

Purpose

The purpose of these labs is to provide a deeper understanding of the widely-used and powerful AWS platform. Specifically, the first three labs focus on teaching key features of Amazon's services. The introduction to IAM is essential for understanding secure web service policies, both in general and within the context of AWS. Creating a VPC and deploying a web server is a fundamental AWS task that enables users to build scalable, reliable, and flexible architecture. Finally, launching EC2 instances is a crucial feature of AWS, and mastering it is vital to fully leveraging the platform. Overall, these labs offer an excellent introduction to one of the most widely used IaaS platforms today.

Background Information/Lab Concepts

Amazon Web Services (AWS) is a widely-used cloud platform that offers a broad range of cloud computing services, including computing power, storage, databases, networking, and artificial intelligence. As an Infrastructure as a Service (IaaS) model, AWS enables customers to pay for access to Amazon's physical hardware for storage and computing, tailored to their specific needs.

Launched in 2006, AWS quickly became a leading player in the cloud services market due to its ability to provide scalable, flexible, and cost-effective infrastructure solutions. Before its official release, Amazon had already been using AWS internally to support its e-commerce platform, recognizing cloud computing as the most efficient method for its business needs. This eventually led to offering the services externally to customers starting in 2006. Today, AWS serves millions of customers, ranging from startups to government agencies.

One of the major advantages of AWS is its pay-as-you-go pricing model, which allows businesses to scale their operations without committing to long-term financial investments in physical hardware. This pricing flexibility has made AWS particularly appealing to smaller businesses, providing them with affordable access to robust cloud services. Furthermore, AWS offers high reliability and security through a global network of data centers called Availability Zones, ensuring services remain accessible even in the event of failures or regional issues.

The first lab in this guide focuses on AWS Identity and Access Management (IAM), a critical service that helps customers securely manage access to AWS resources. IAM allows the creation and management of users, groups, and permissions, ensuring that only the right people have access to the necessary services. It also supports granular access control, enabling permissions to be granted on a service-by-service or even resource-by-resource basis, minimizing risk and ensuring employees have just the right level of access for their tasks.

The second lab introduces users to building a Virtual Private Cloud (VPC) and deploying a web server. A VPC is an isolated network within AWS, allowing users to define their own virtual network topology, including subnets, routing tables, and security settings. This is essential for creating the secure, scalable, and customizable network infrastructure that AWS is known for.

The third lab covers Amazon EC2 (Elastic Compute Cloud), a core AWS service that provides resizable compute capacity in the cloud. EC2 allows users to launch virtual servers, known as instances, which can be scaled up or down depending on the current demand. With a wide variety of instance types, storage options, and operating systems, EC2 offers incredible flexibility,

making it suitable for everything from simple web applications to complex machine learning models.

Together, these labs provide an essential introduction to AWS and its core services, equipping users with the knowledge needed to effectively utilize the platform's powerful capabilities.

Lab Summary

To begin using IAM, I first explored users and groups. I opened the IAM console and selected one of the users. This user had no permissions or group assignments, and was only assigned a Console password. In the User Groups section, I clicked on the EC2-Support group link, navigated to the Permissions tab, and viewed the policy details. I repeated this process for the other groups. Next, I referred to the 'Business Scenario' in the lab documentation to proceed.

The next task was adding users to groups. I selected the User Groups section, then in the Users tab, I added users to the respective groups, starting with S3-Support. I repeated this for each group accordingly. For task three, which involved signing in and testing users, I opened the IAM dashboard, copied the IAM users sign-in URL, and pasted it into a private window. After signing in with each user account, I confirmed that each user had the necessary permissions.

For the VPC and web server task, the first step was to create the VPC. I verified my region and chose to create a VPC from the VPC dashboard. Using the documentation as a guide, I configured the VPC and confirmed the settings. The second task was to create additional subnets. I went to the Subnets section, created two subnets, and configured them with the correct details. Then, I used the Route Tables section to apply the subnets. Task three involved creating a VPC security group, so I created a new security group, added a rule, and confirmed its creation.

Task four focused on launching a web server instance. I opened EC2 and launched an instance, configuring and launching it. Afterward, I waited for the instance to pass its status checks. Once complete, I copied the public DNS value, and upon pasting it, I saw the AWS logo, confirming the deployment of the correct network architecture.

For the EC2 introduction, the first step was launching an Amazon EC2 instance. I opened the EC2 console, chose to launch an instance, and completed the configuration. The next step was to monitor the instance, which I did by navigating to the Monitoring tab and setting the system to capture system logs and screenshots.

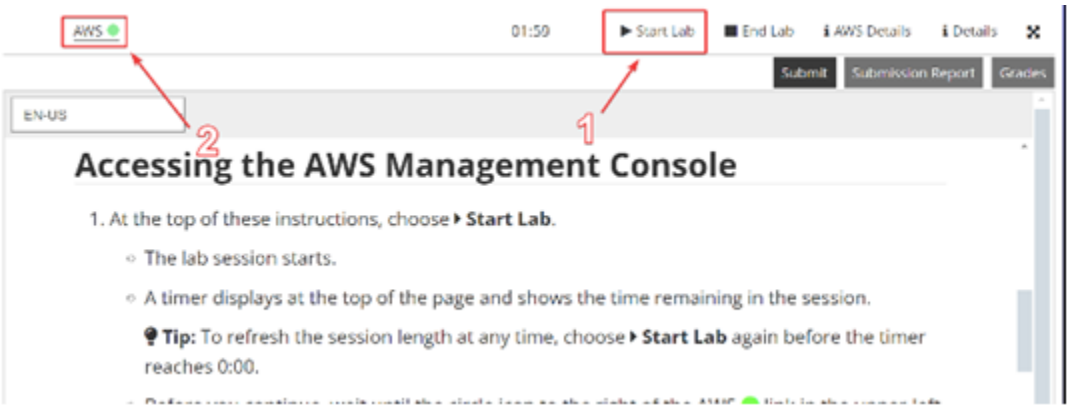
In task three, I updated the security group and accessed the web server. I pasted the public IP of the web server into a new window. Then, I returned to the EC2 console, edited the inbound rules, and confirmed the changes. Task four involved resizing the instance, instance type, and EBS

volume. I stopped the instance, used the Actions dropdown to change the instance type, enabled stop protection, and resized the EBS volume. Afterward, I launched the resized instance.

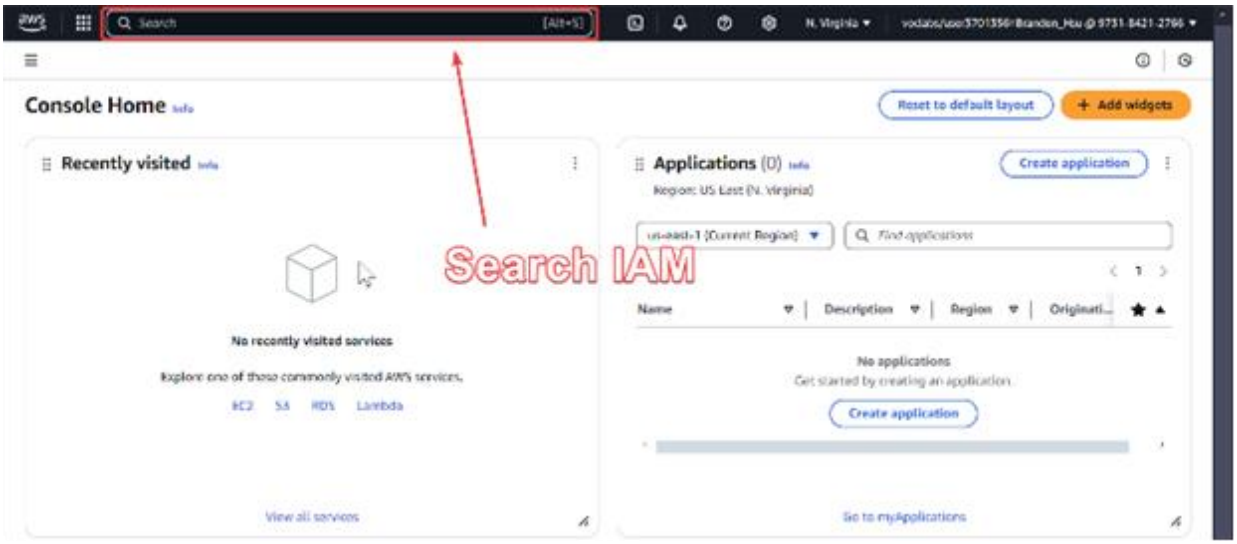
For task five, I explored the EC2 instance by using the service quotas to check the EC2 limits. In task six, I tested the stop protection feature. I attempted to stop the instance, which failed. I then unchecked the stop protection option under instance settings in the Actions menu, which allowed the instance to stop successfully.

Lab Commands

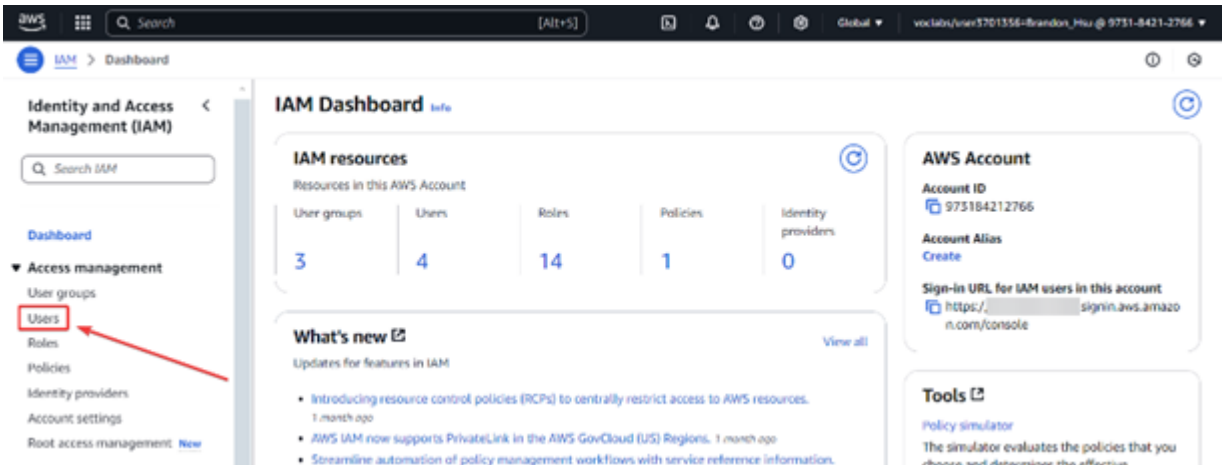
LAB 1:



Click Start Lab, then click “AWS” when the circle becomes green

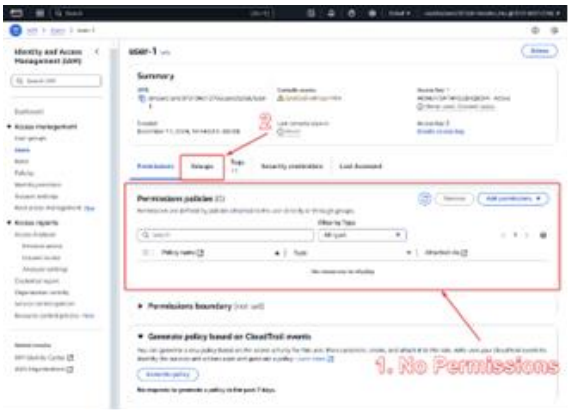


Search IAM and go into the IAM dashboard



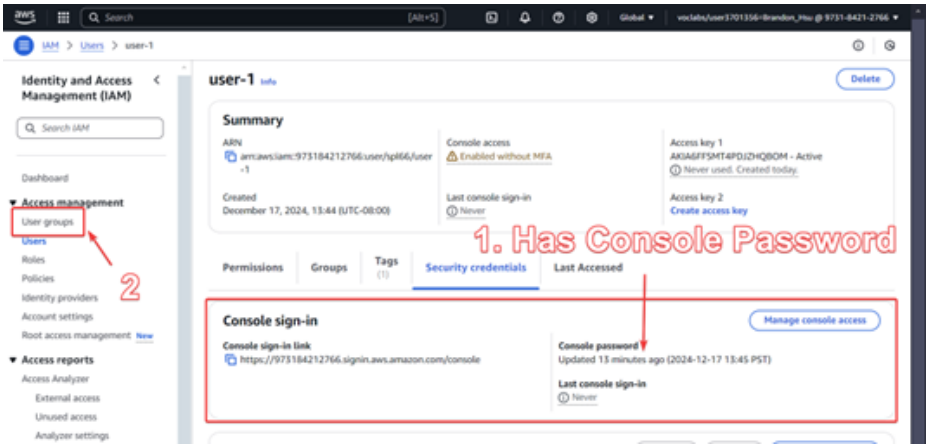
First search for the IAM console page to start viewing the pre-configured configurations of the lab

Select “User-1”

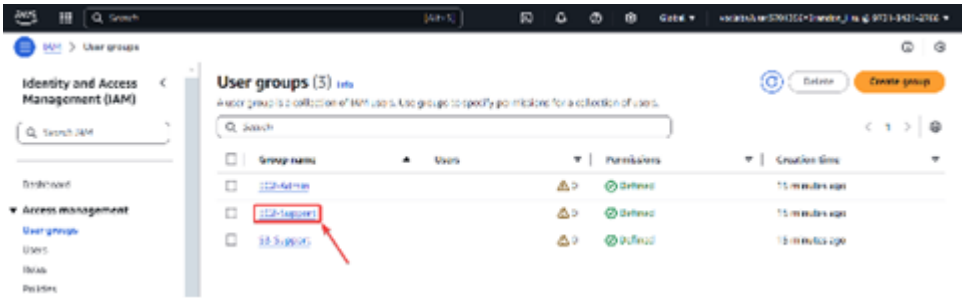


Notice how it has no permissions, then go to groups tab

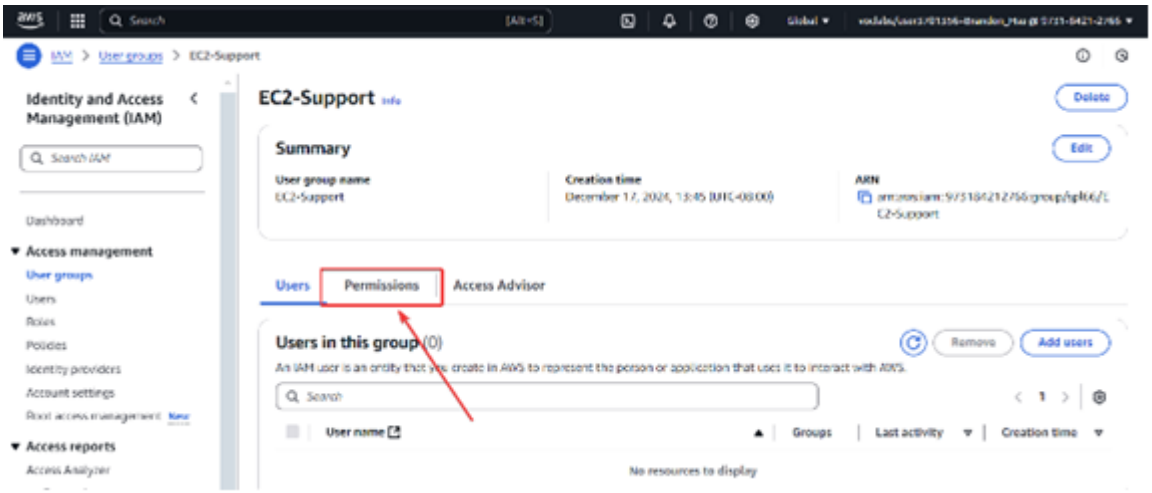
Notice how User-1 doesn't have any groups, then go to the security credentials tab



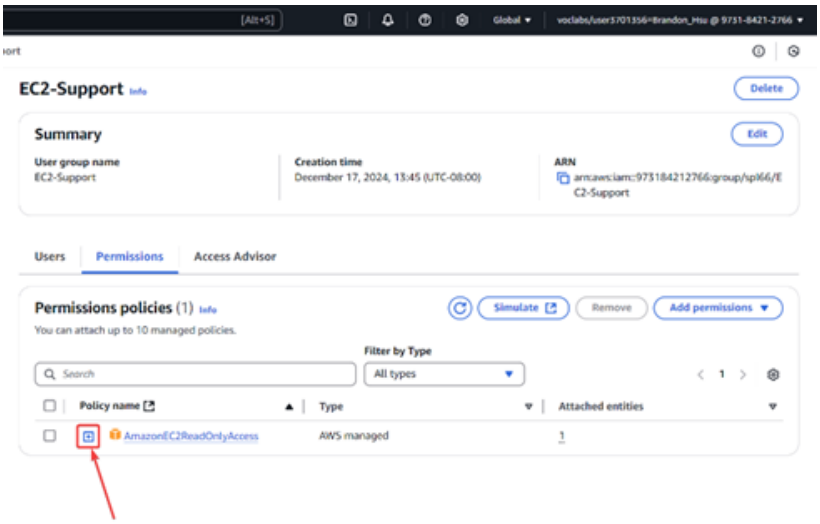
Notice that User-1 has a console password, now on the left navigation panel, select User Groups



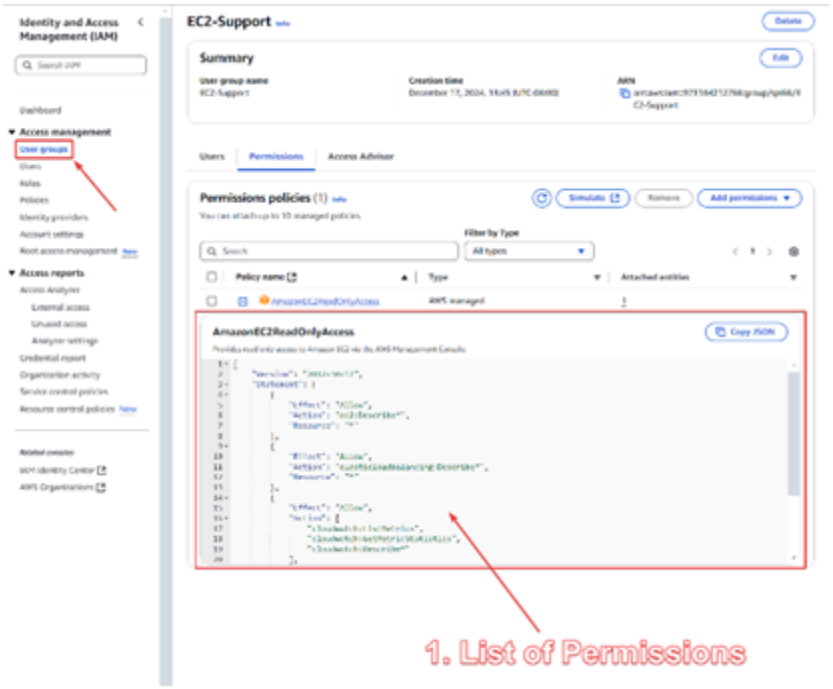
Go to the EC2-Support group



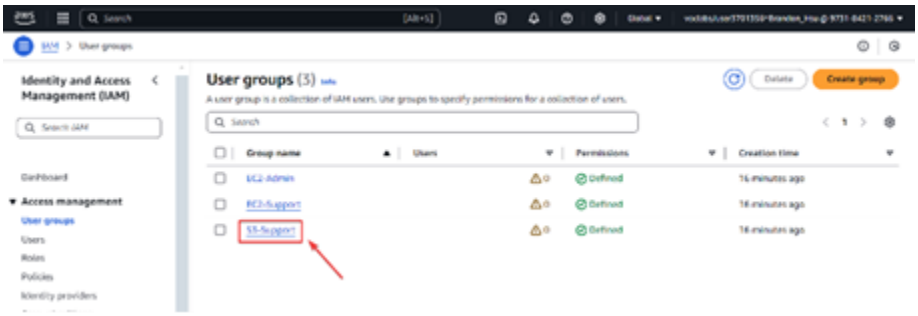
Select the Permissions tab



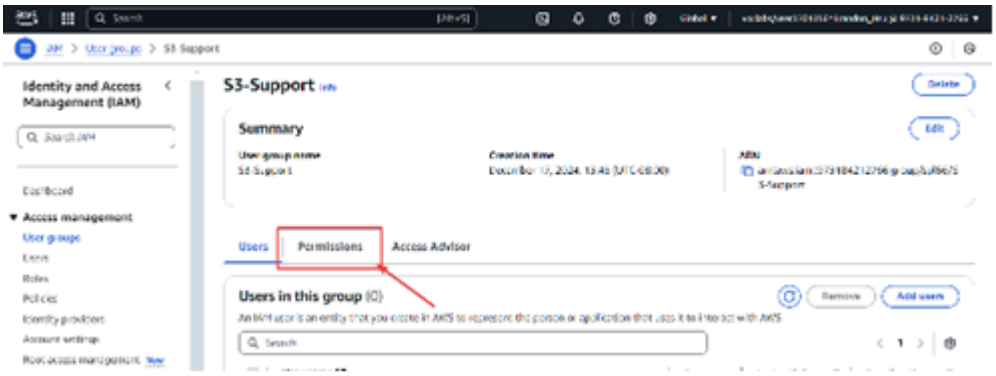
Select the box for more detailed information on the EC2-Support group



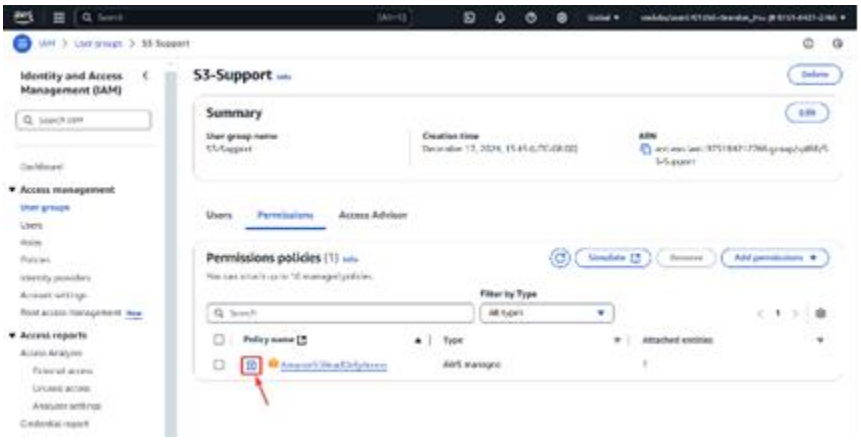
Look at the list of permissions, then go back to the User Groups in the left navagational panel



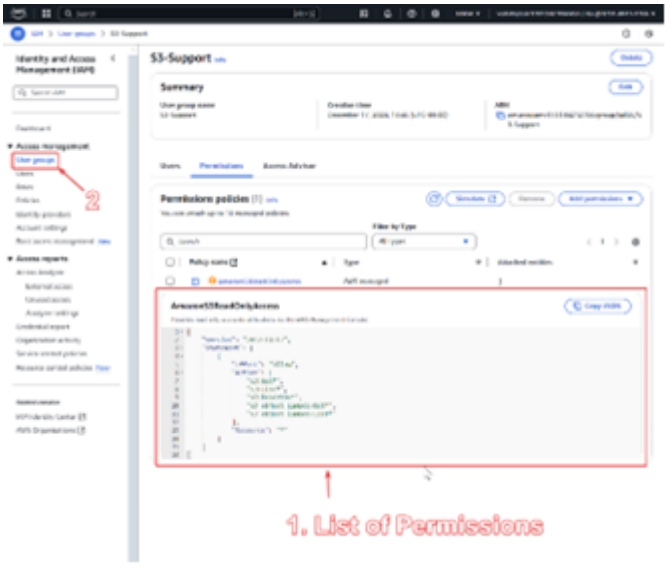
Now go into S3-Support group



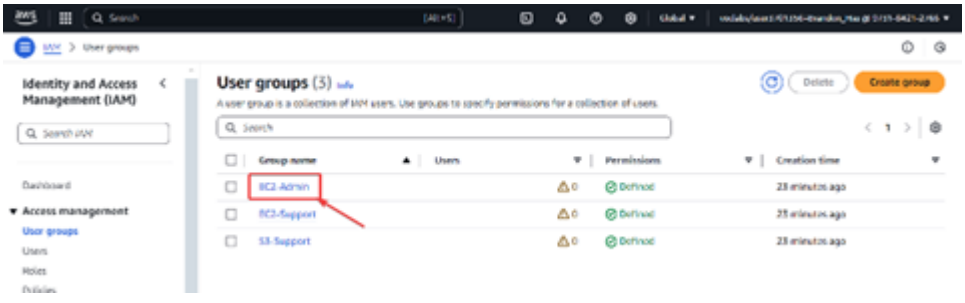
Go to the Permissions tab



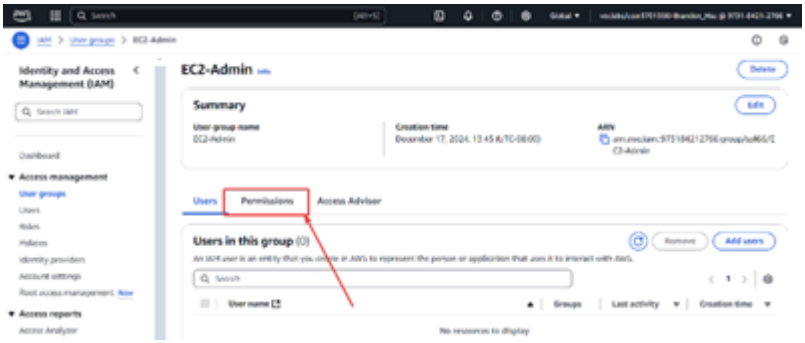
Again, click the box for more detailed information



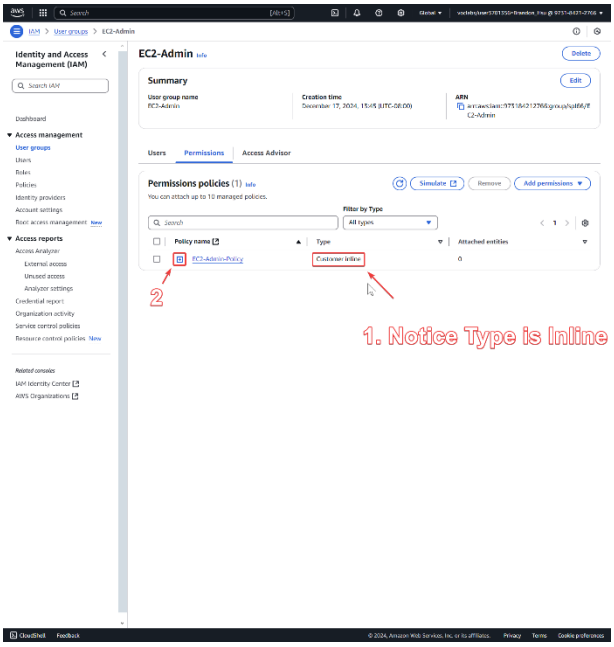
View the list of permissions, go back to User Groups from the left navagational panel



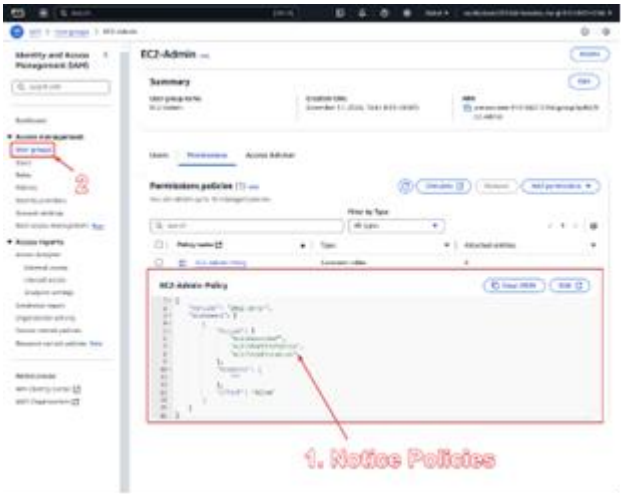
Now go to the final group, the EC2-Admin group



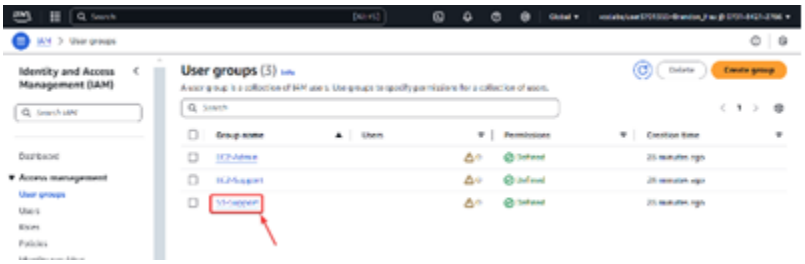
Go to the Permissions tab



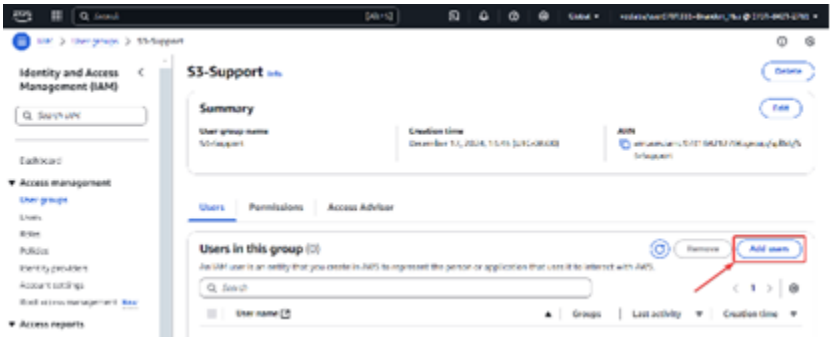
See how the type of poicy is inline, and view the more details by clicking the box



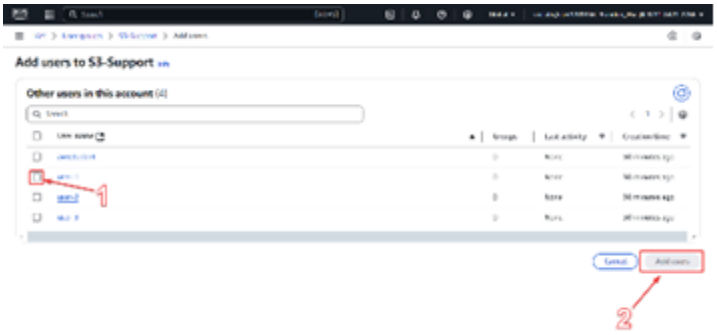
Notice the Policies, then go back to the User groups section in the left navagational panel to start adding users to the groups



Go to the S3-Support group

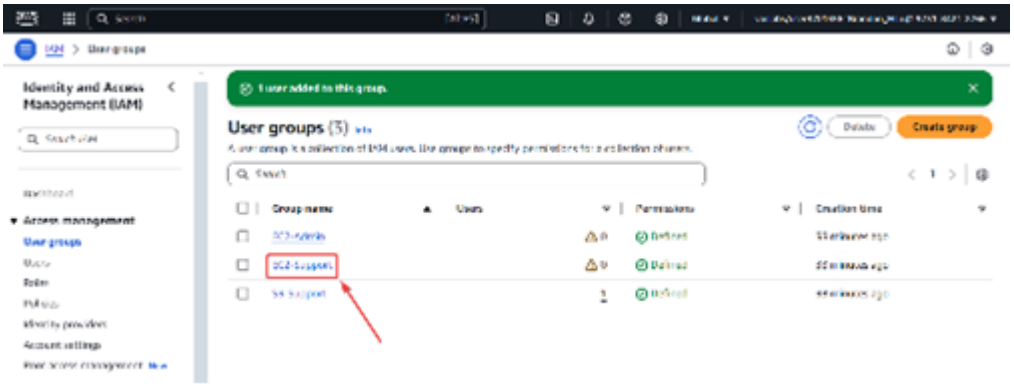


Select “Add users”

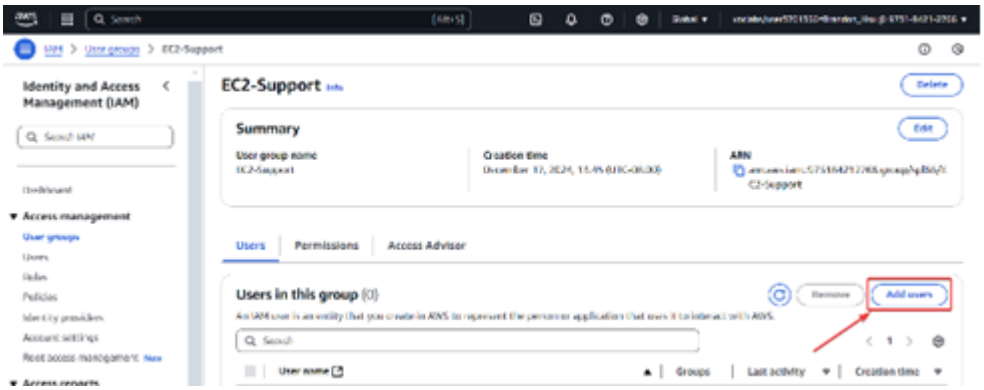


Select user-1’s check box and then select “Add users”

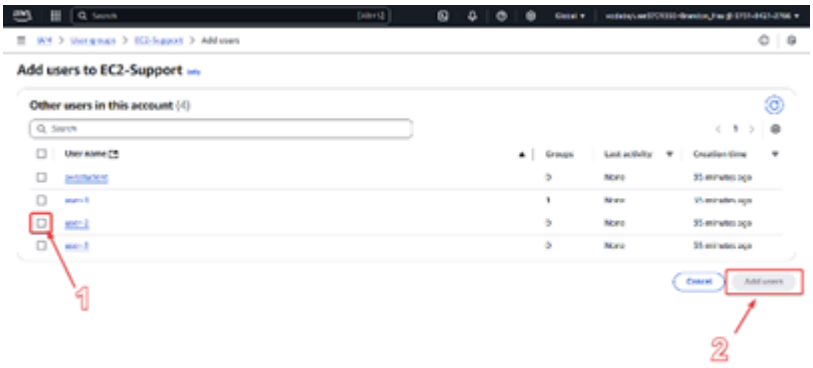
Notice the green box confirming User-1 has been added to the S3-Support group



Next go into the EC2-Support group

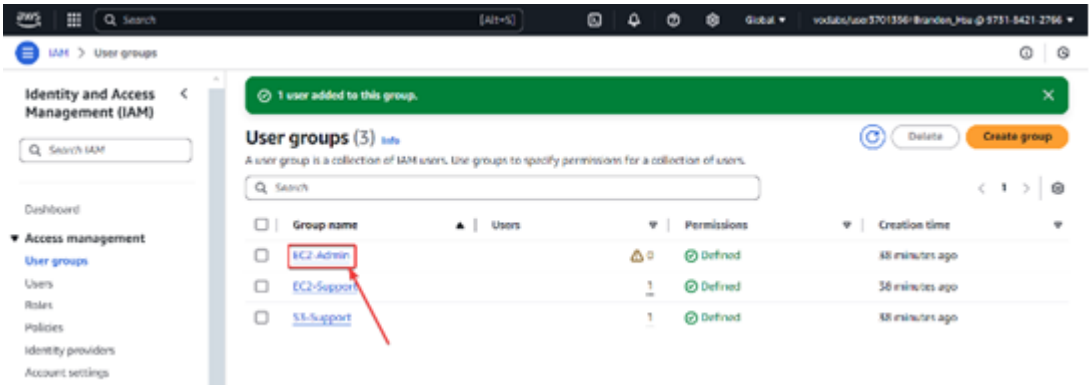


Select “Add users”

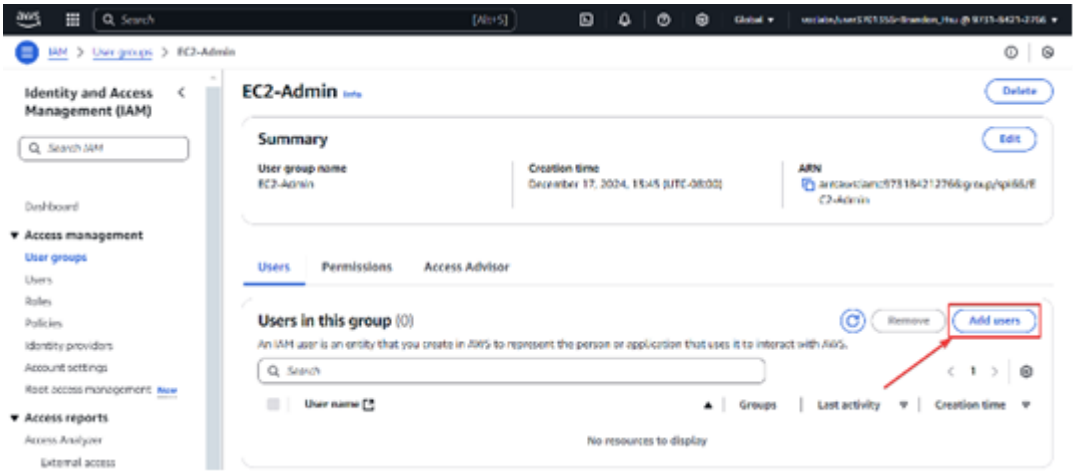


Same as user-1, check the box next to user-2 and then click “Add users”

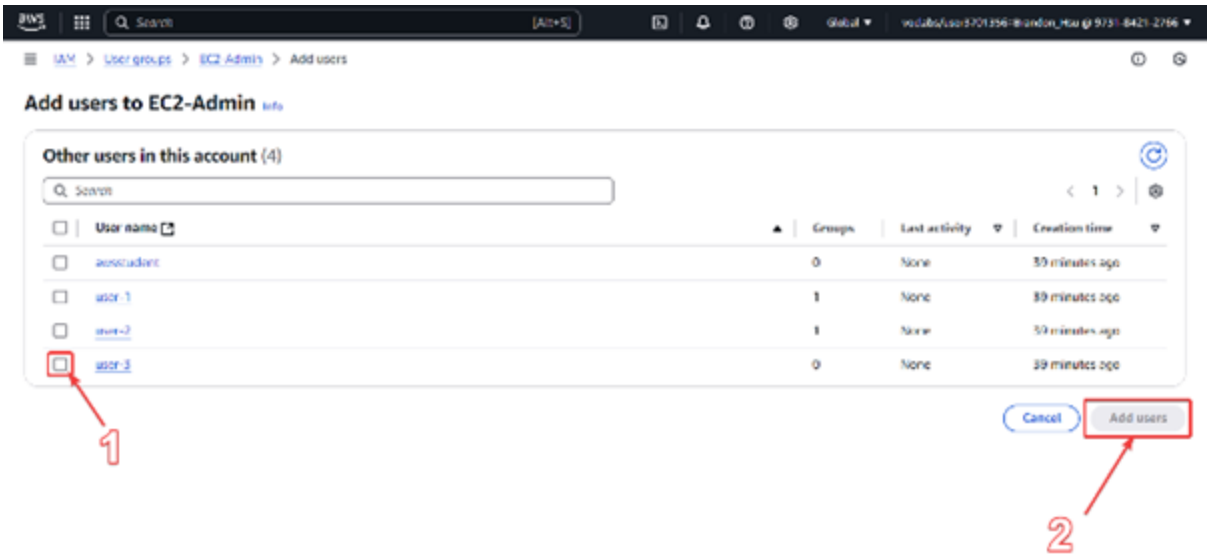
Notice how user-2 has been added, then go back to add user-3 to the EC2-Admin



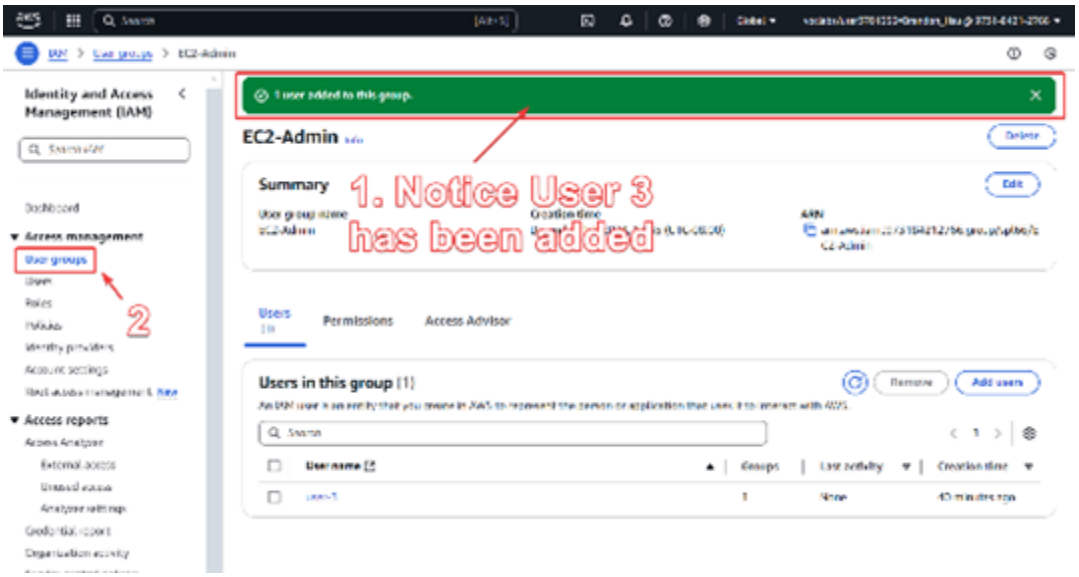
Next go to EC2-Admin Group



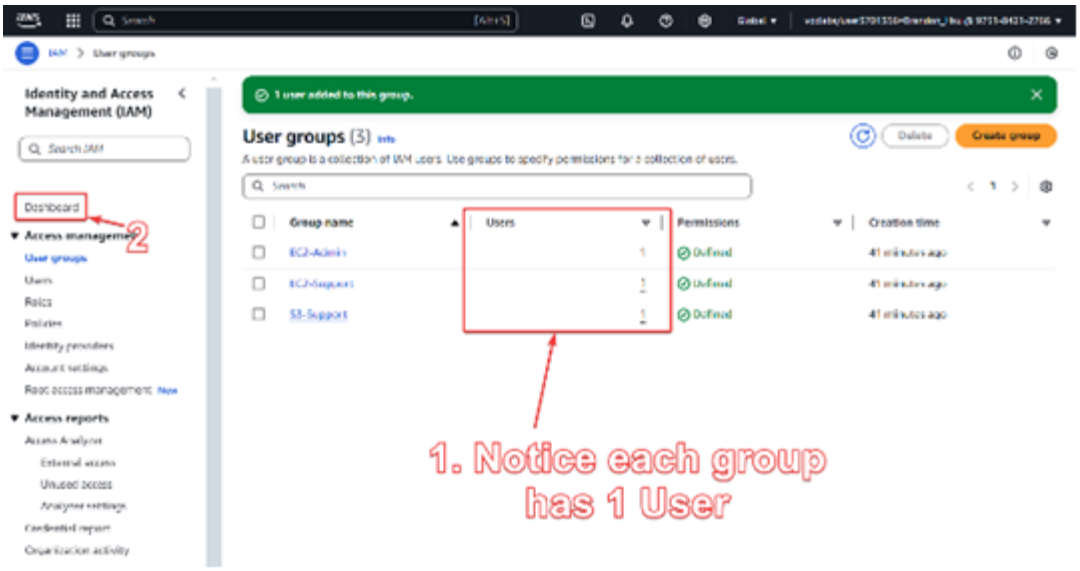
Then click “Add users”



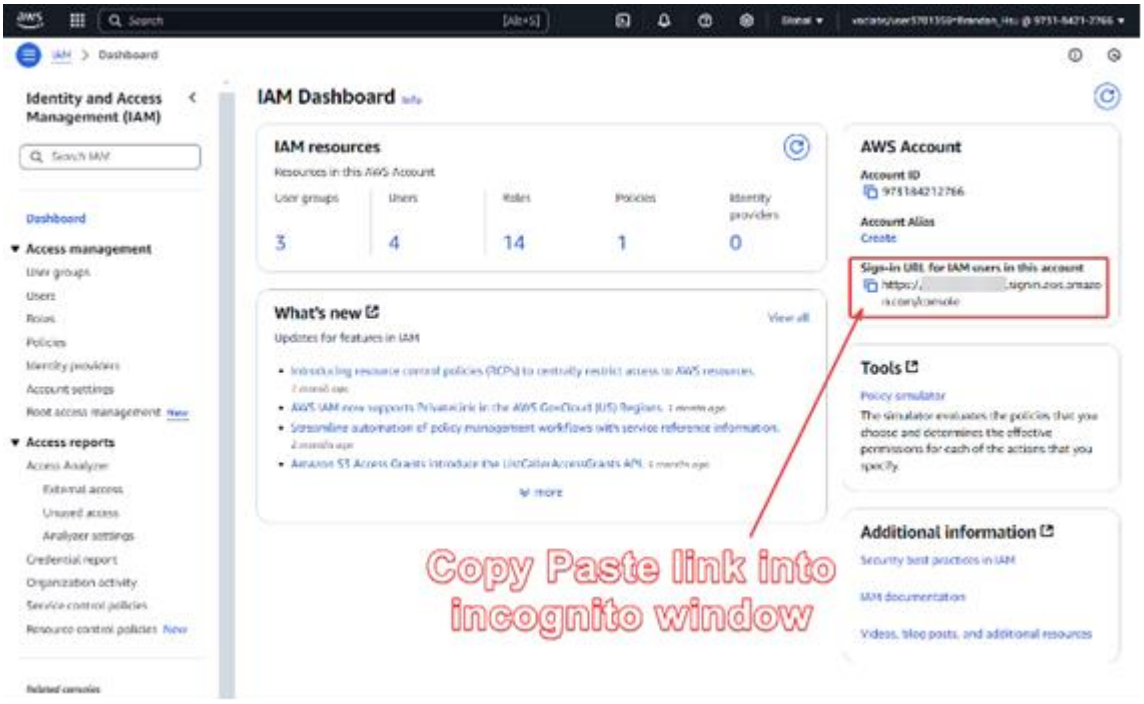
Click the select box on “user-3” then click “Add users”



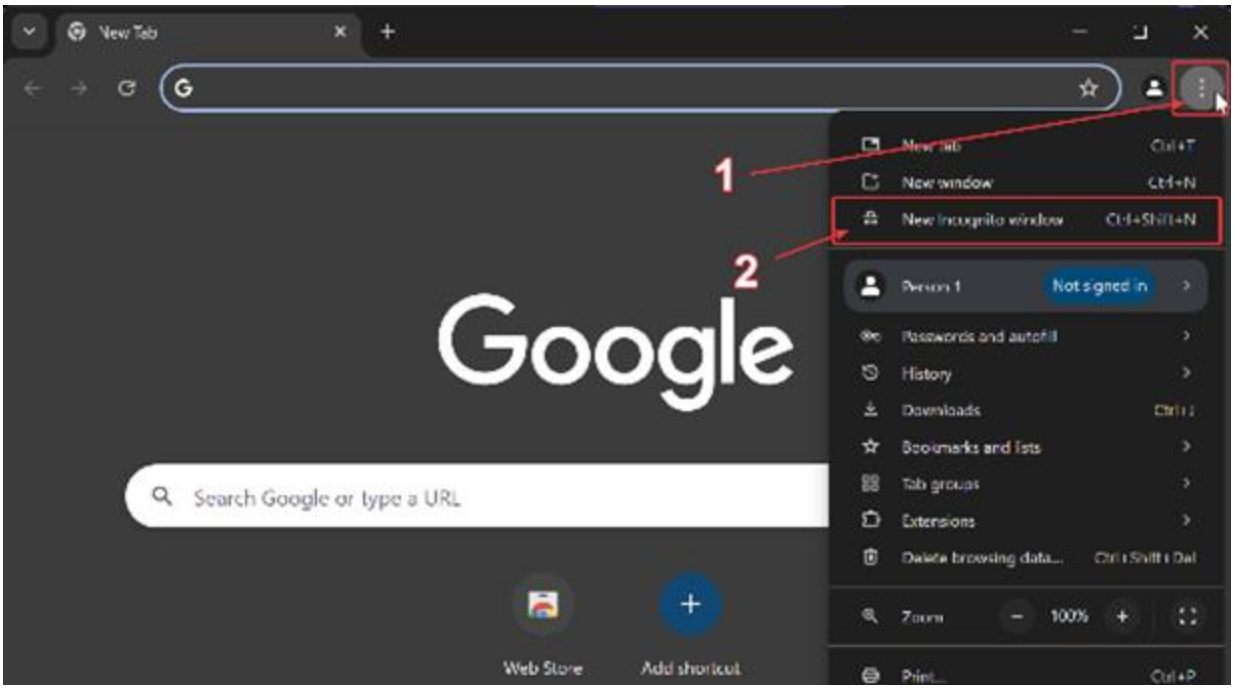
Notice how user-3 has been added, then go to user groups from the left navagation panel



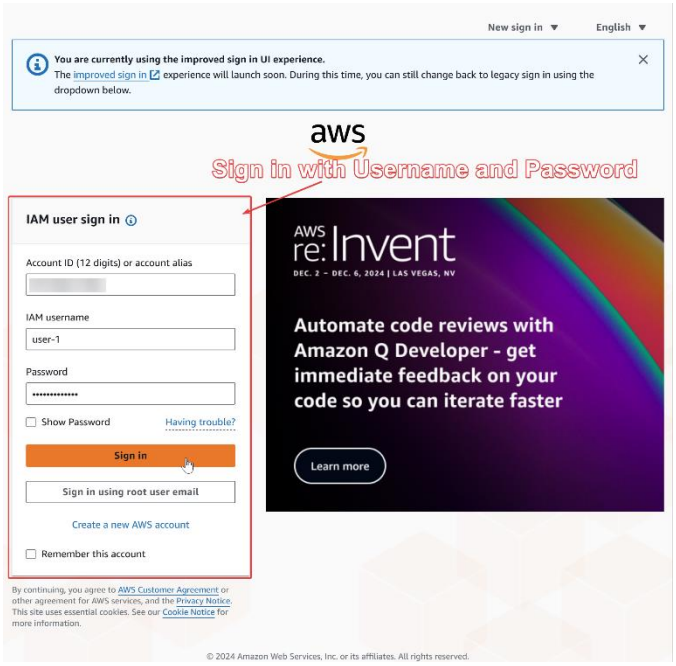
First see how each group has a user, then go to dashboard in the left navagational panel



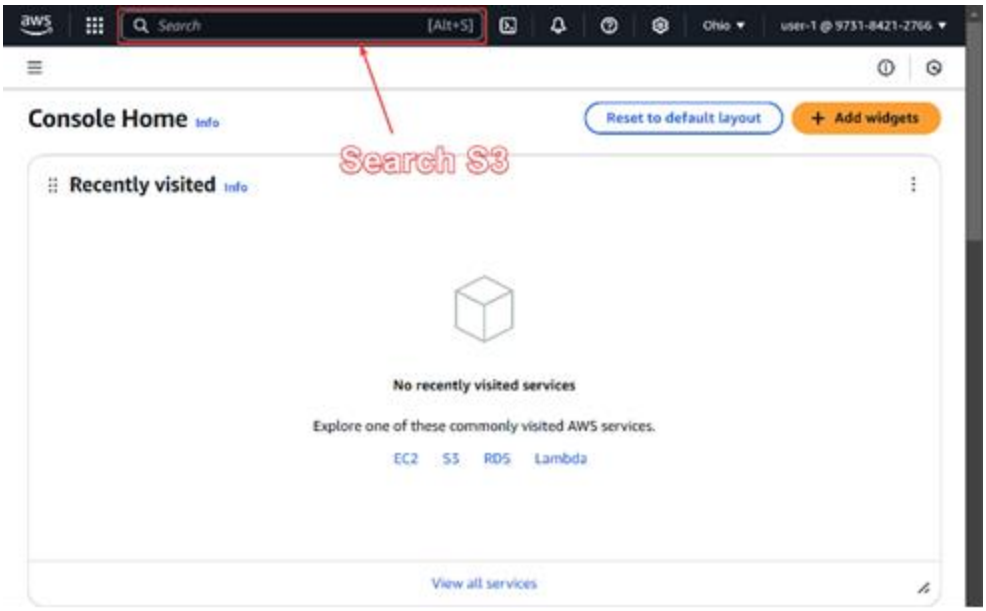
Copy the url for IAM users from the AWS account



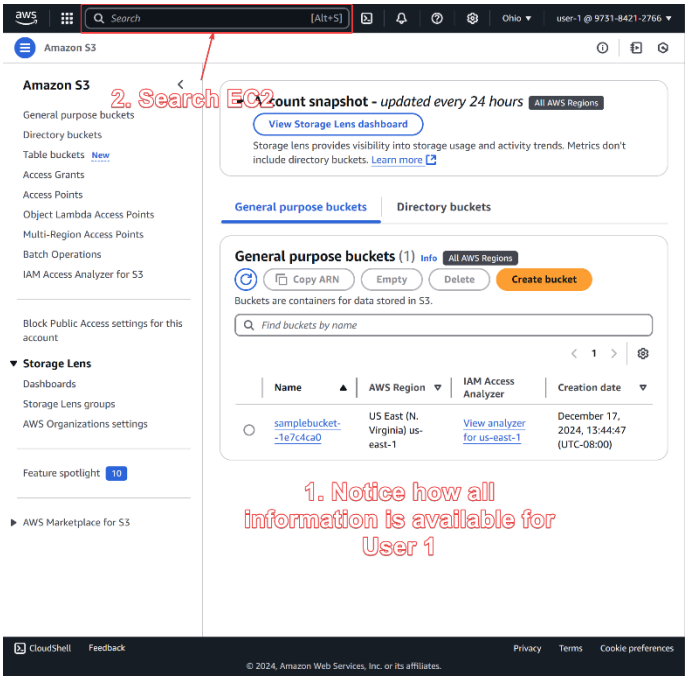
Next open a incognido window in a web browser, then paste the URL you copied in the previous step



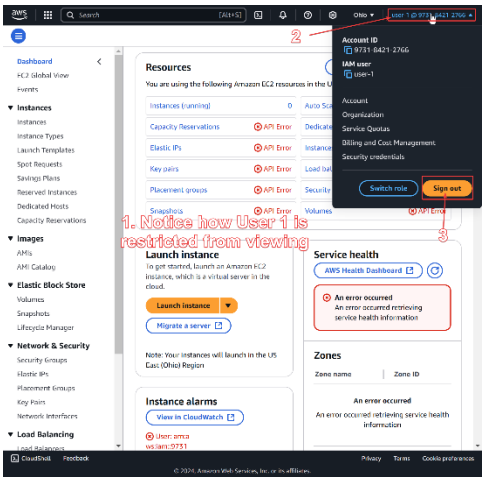
Sign in using your account ID and user 1’s specific username and password



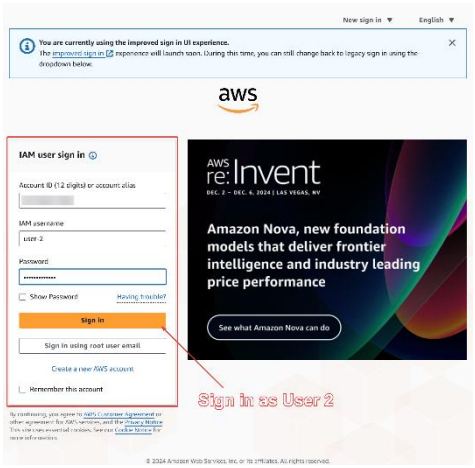
Search and select S3



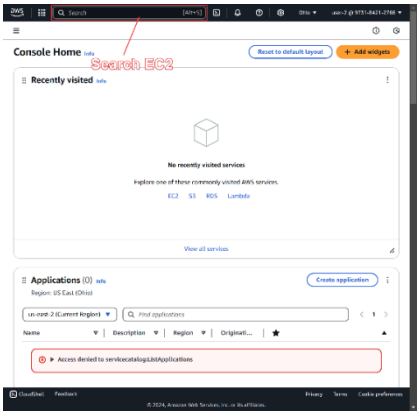
Notice all information is available on S3, then go to EC2



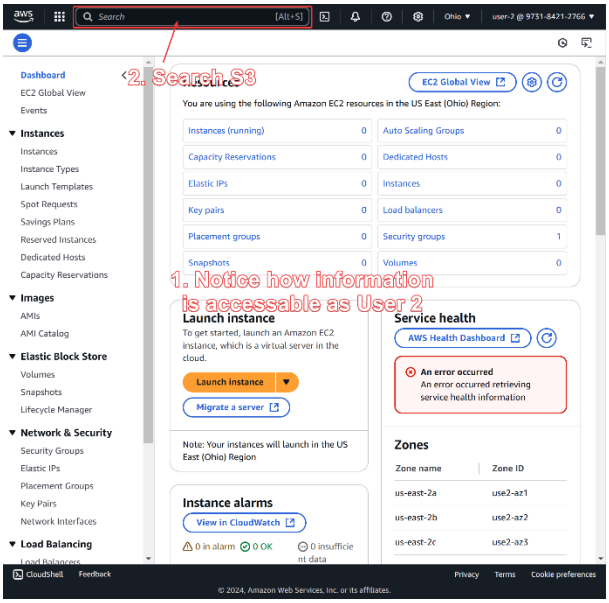
Notice how all the information is blocked, the log out of user-1



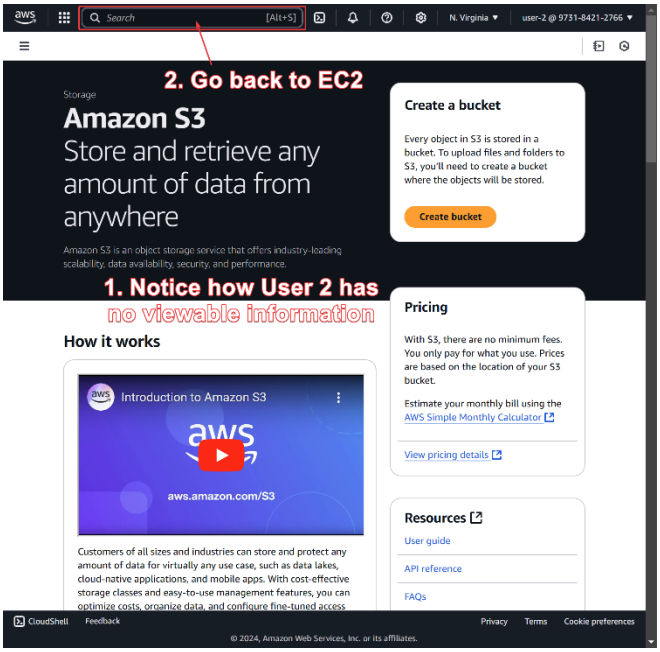
Sign in with user 2’s username and password with the same account ID as before



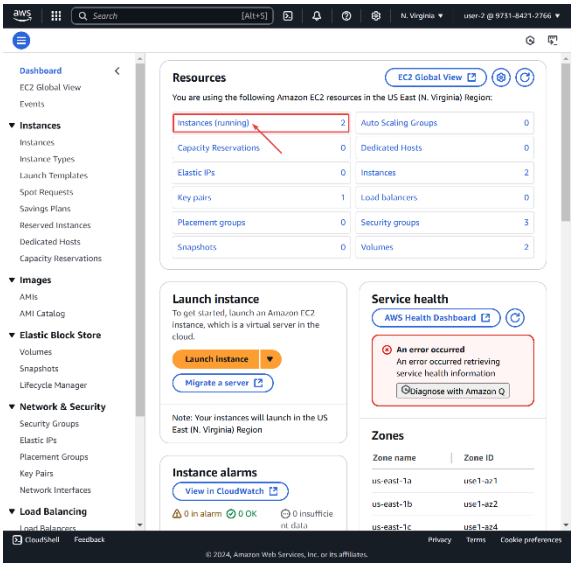
Search for EC2



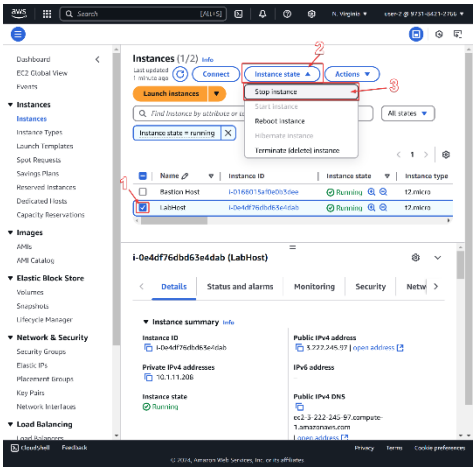
See how some of the information is available to user 2



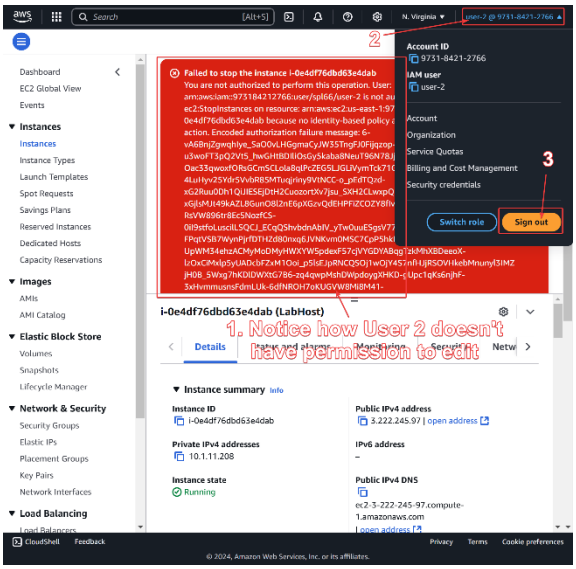
Notice how on User 2 it has no viewable information, then go to the search and go back to EC2



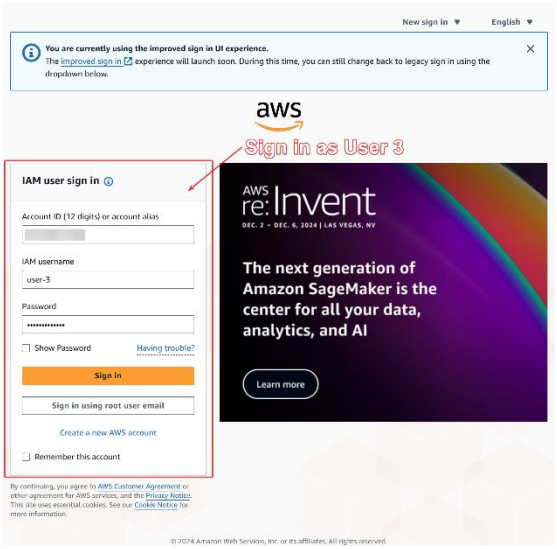
Check the running instances by clicking “instances”



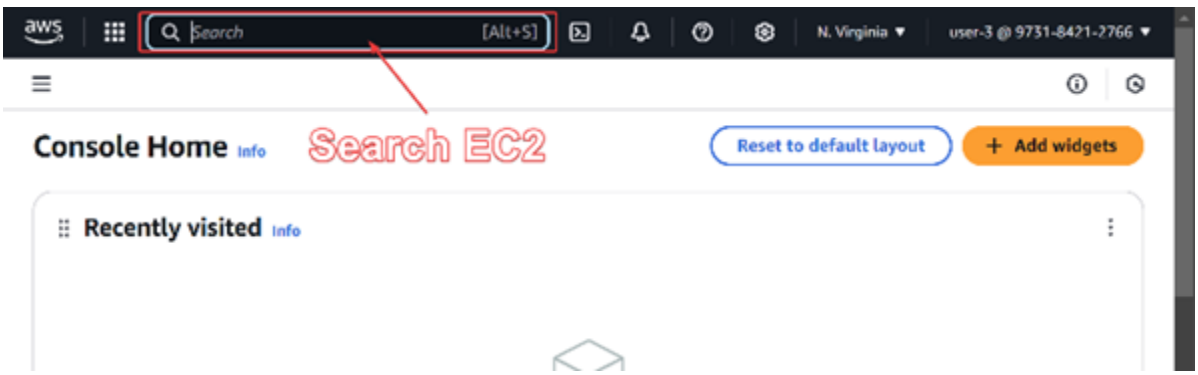
Select “LabHost” and try to stop it using the instance state drop down menu



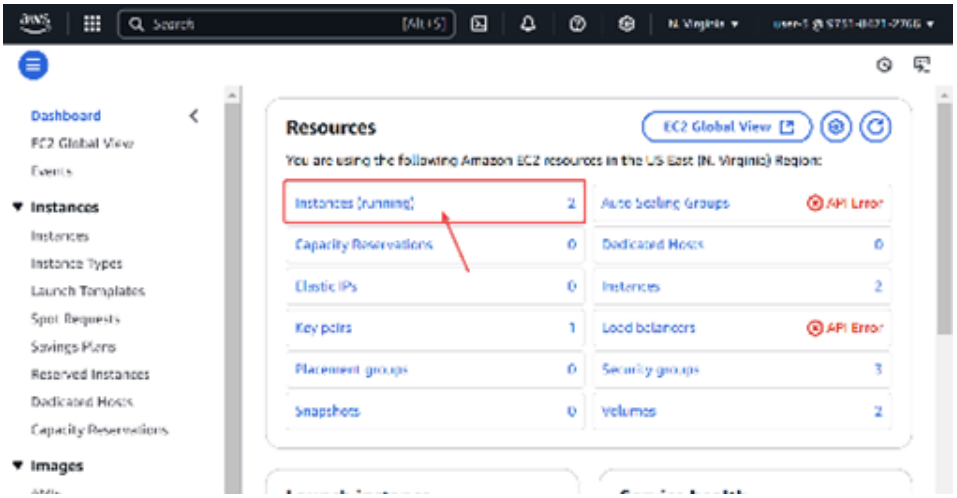
Notice how User-2 doesn't have the authorization to stop the instance from running



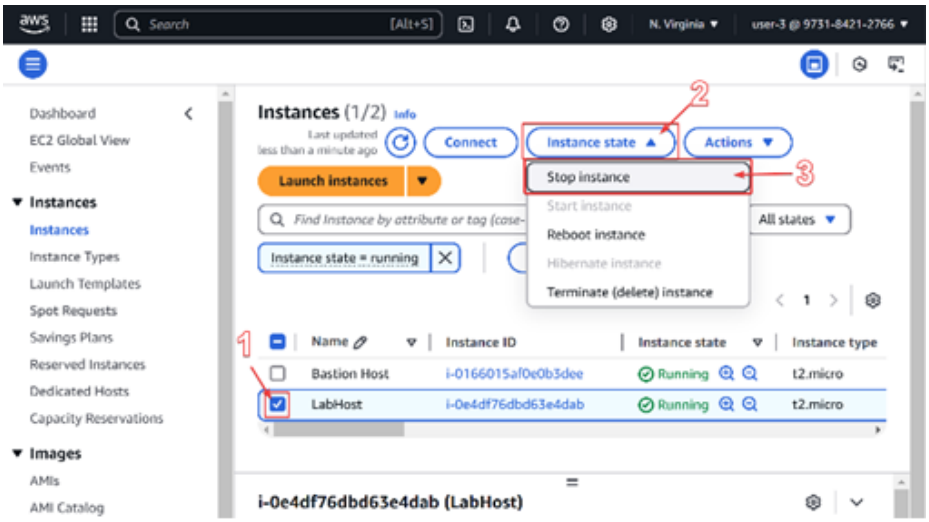
Finally sign into user-3, using the same account ID and user-3's specific username and password



Search EC2



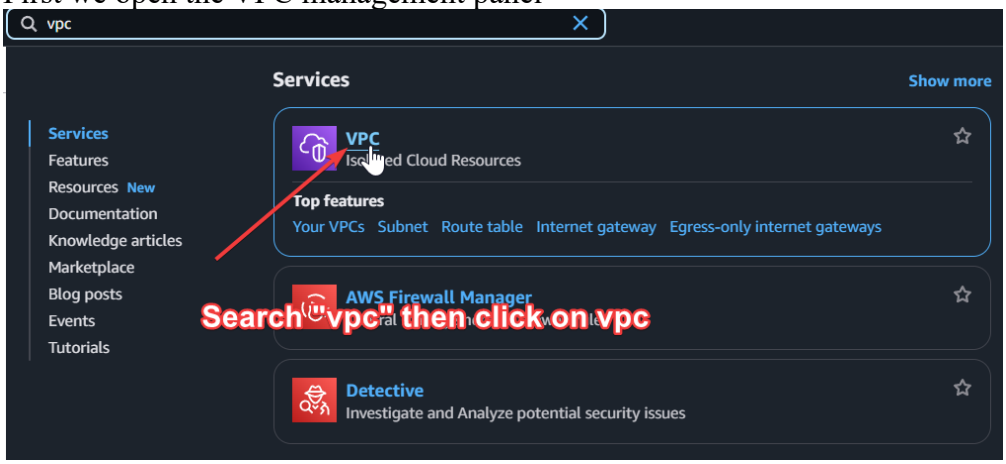
Select “Instances”



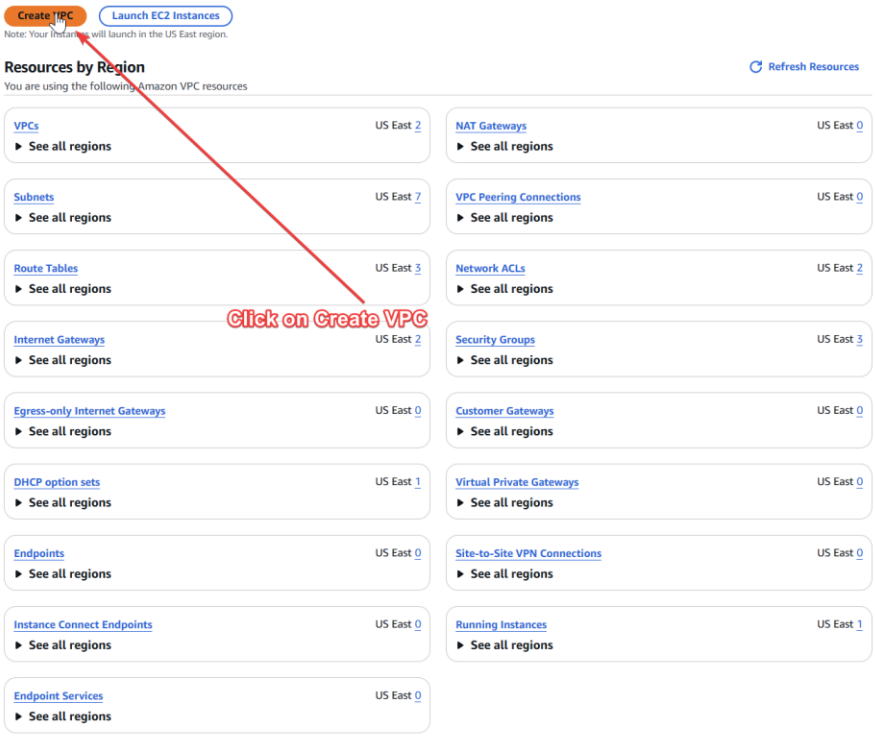
Stop the instance by selecting “LabHost” then the instance state drop down menu

LAB 2:

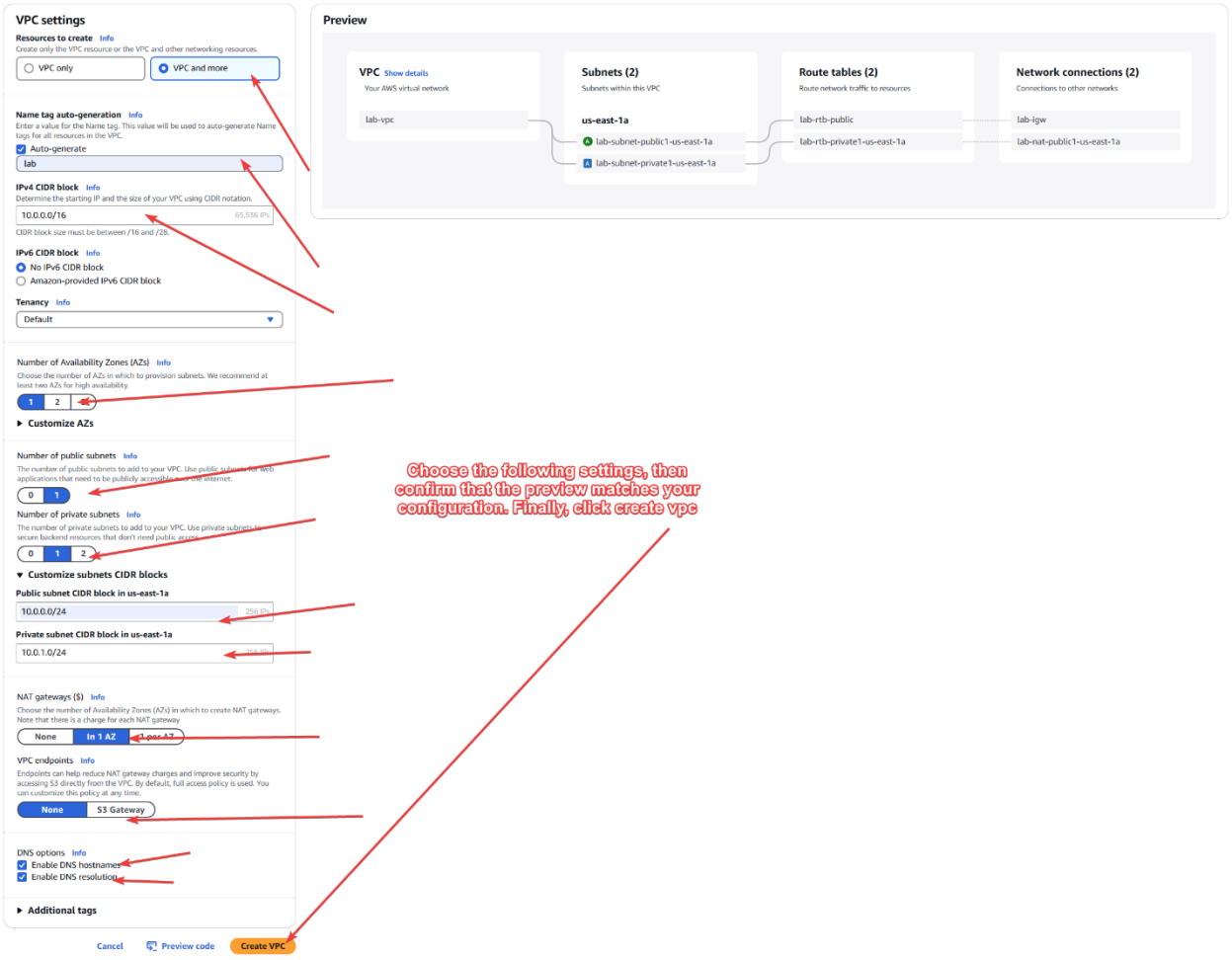
First we open the VPC management panel



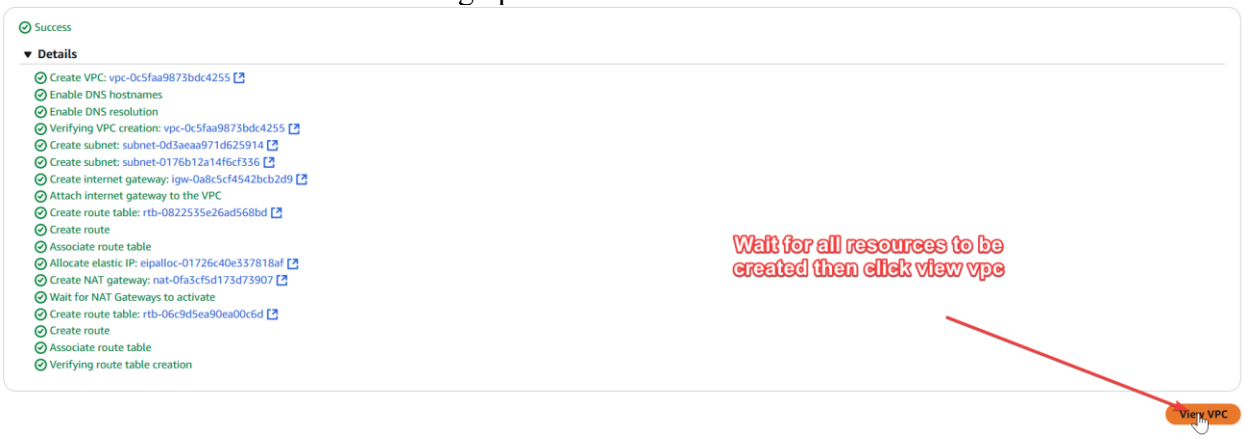
Next we open the VPC configuration menu



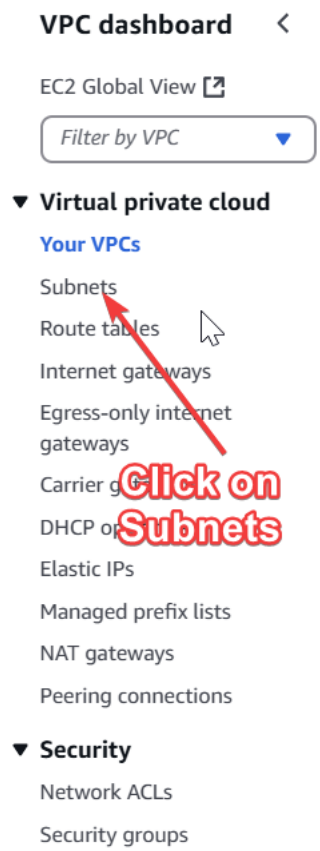
Next we will configure the basic architecture



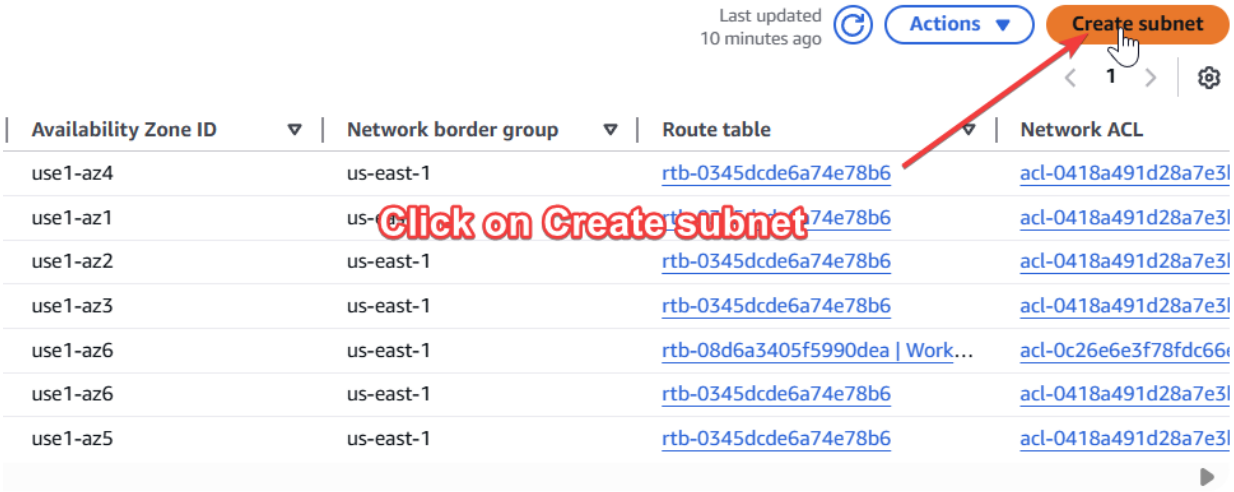
We wait for the VPC to finish setting up



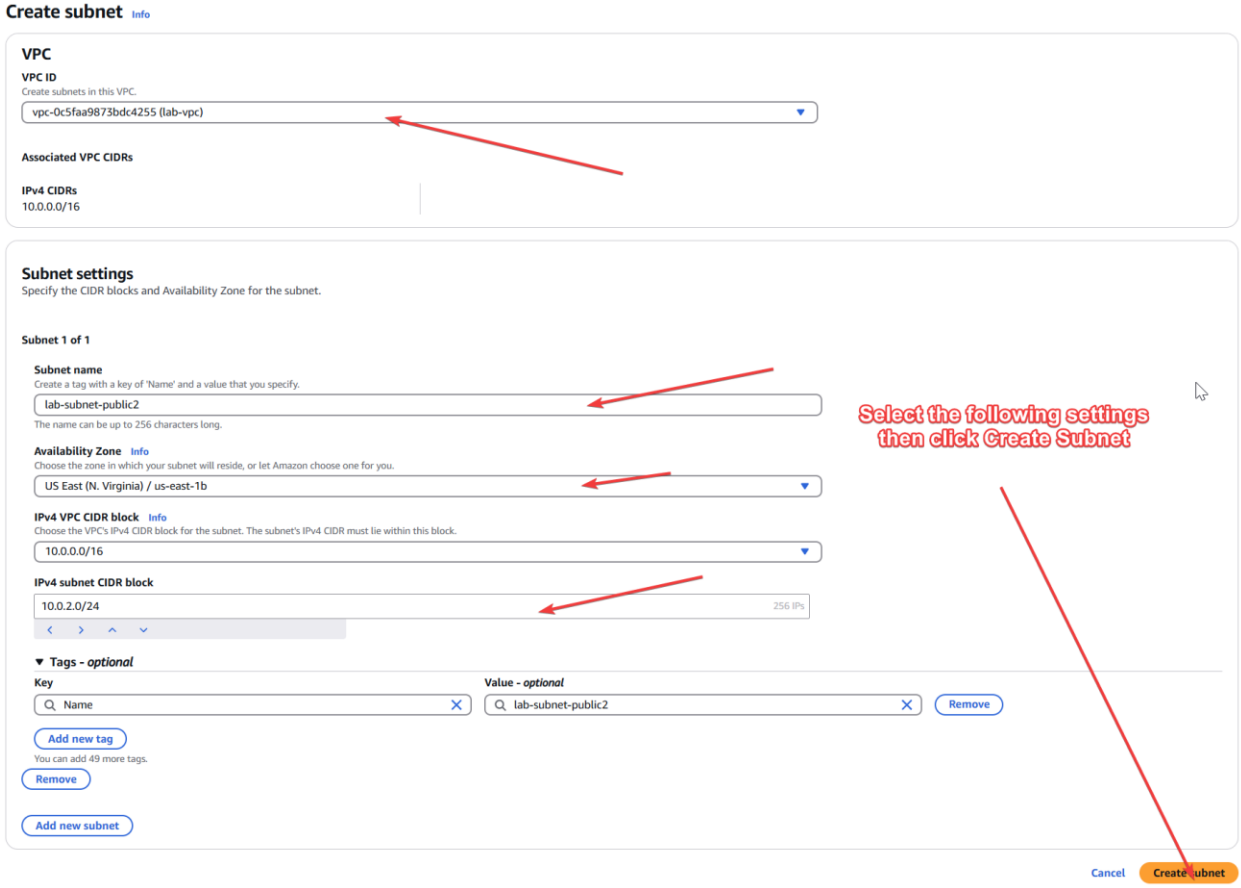
Now we open the subnets menu to configure them



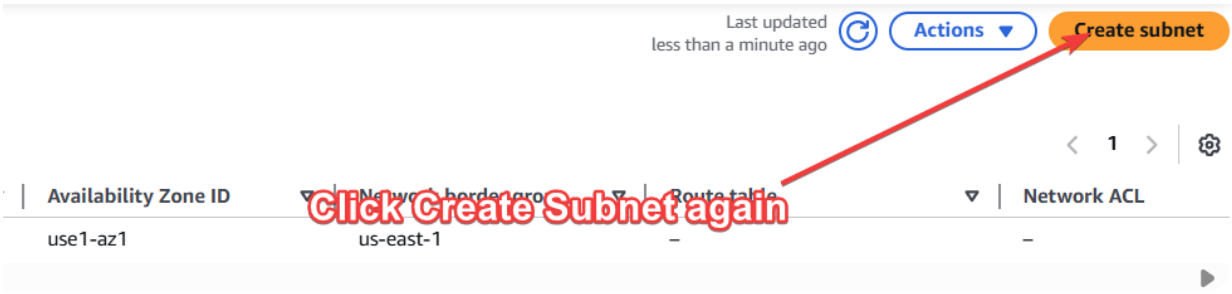
Next we open the subnet configuration menu



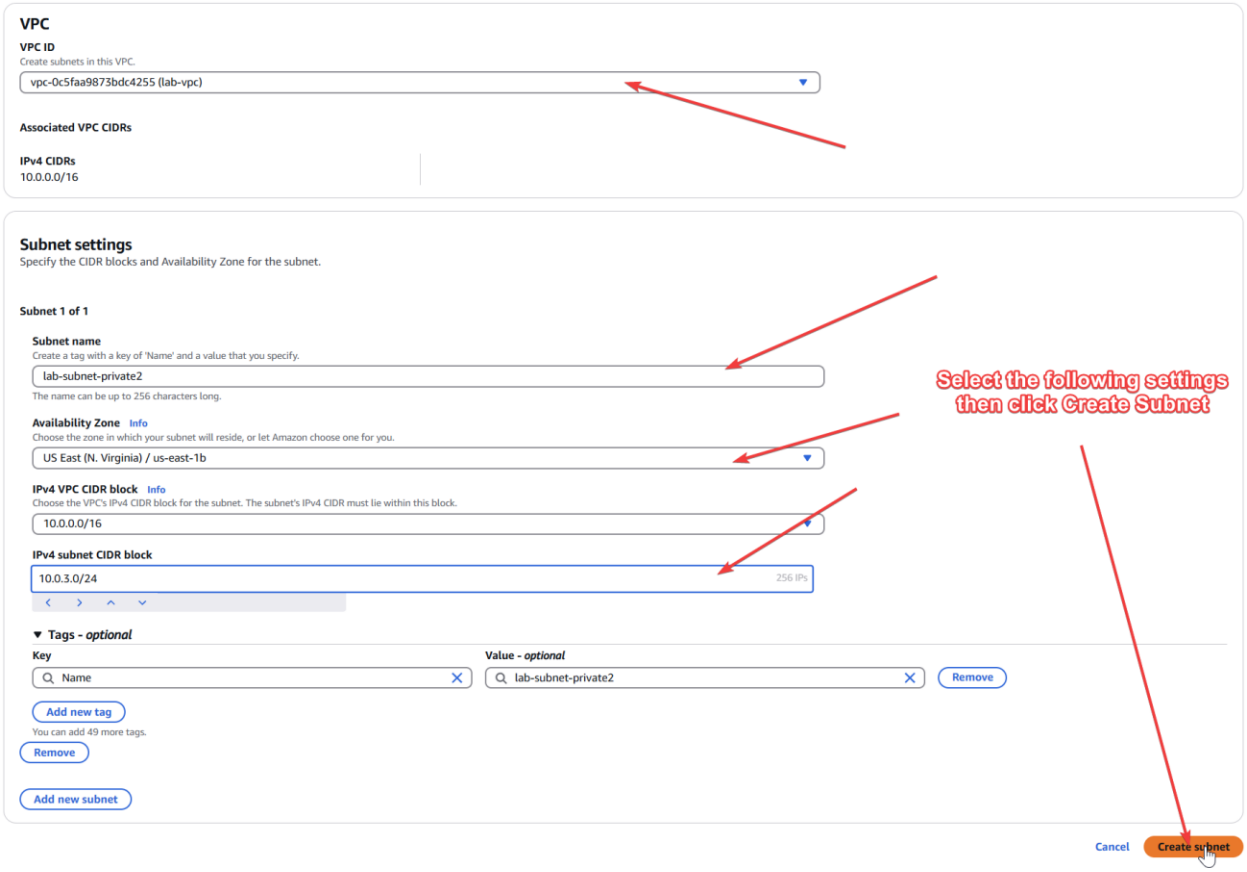
Configure the basic settings for the public subnet then create it



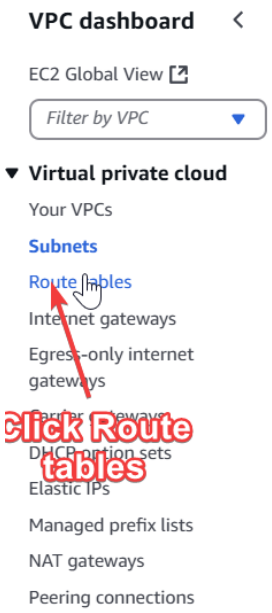
Now click create subnet to configure the private subnet



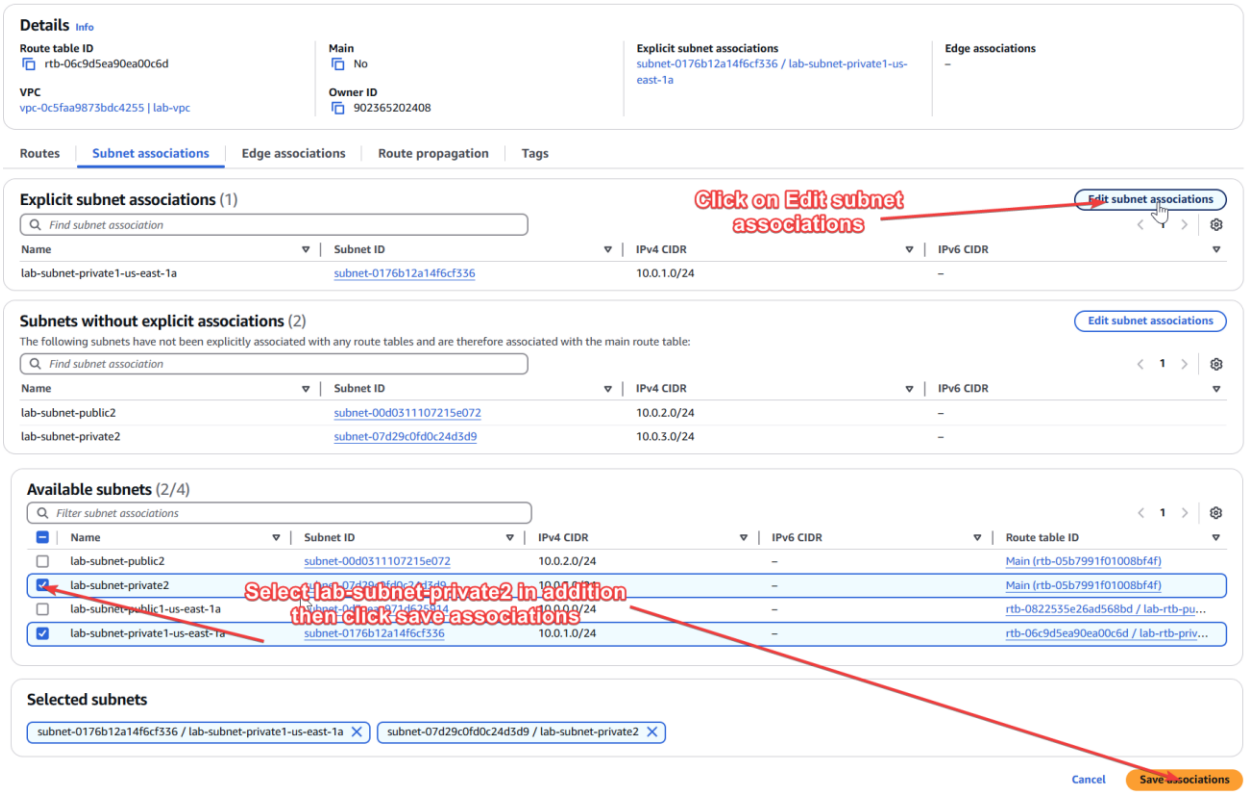
Configure the basic settings and continue onward



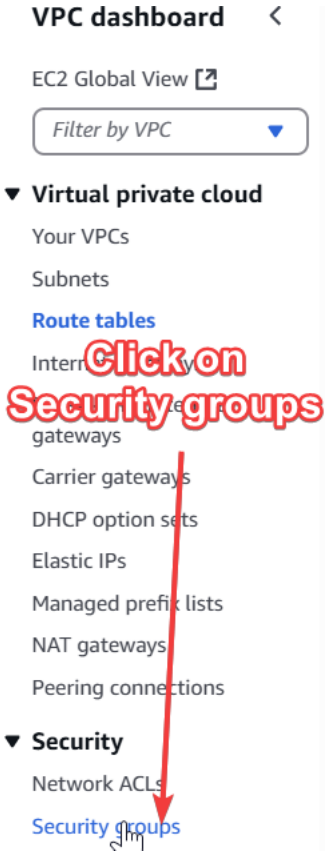
Now review the route tables configured with the subnets



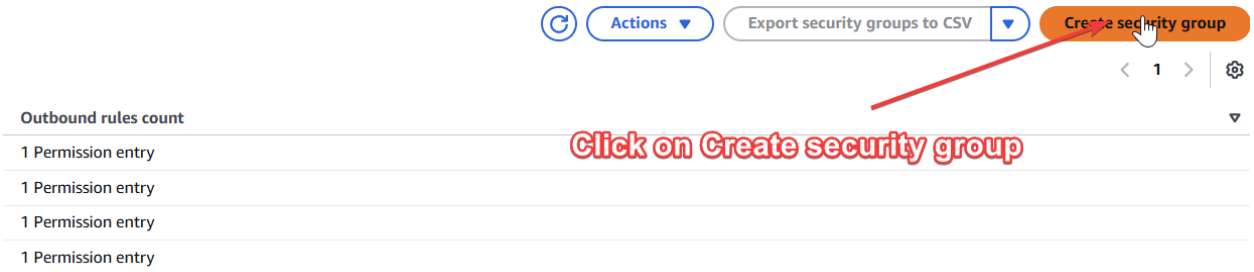
Now associate the subnet associations to enable connection to the subnets over the internet



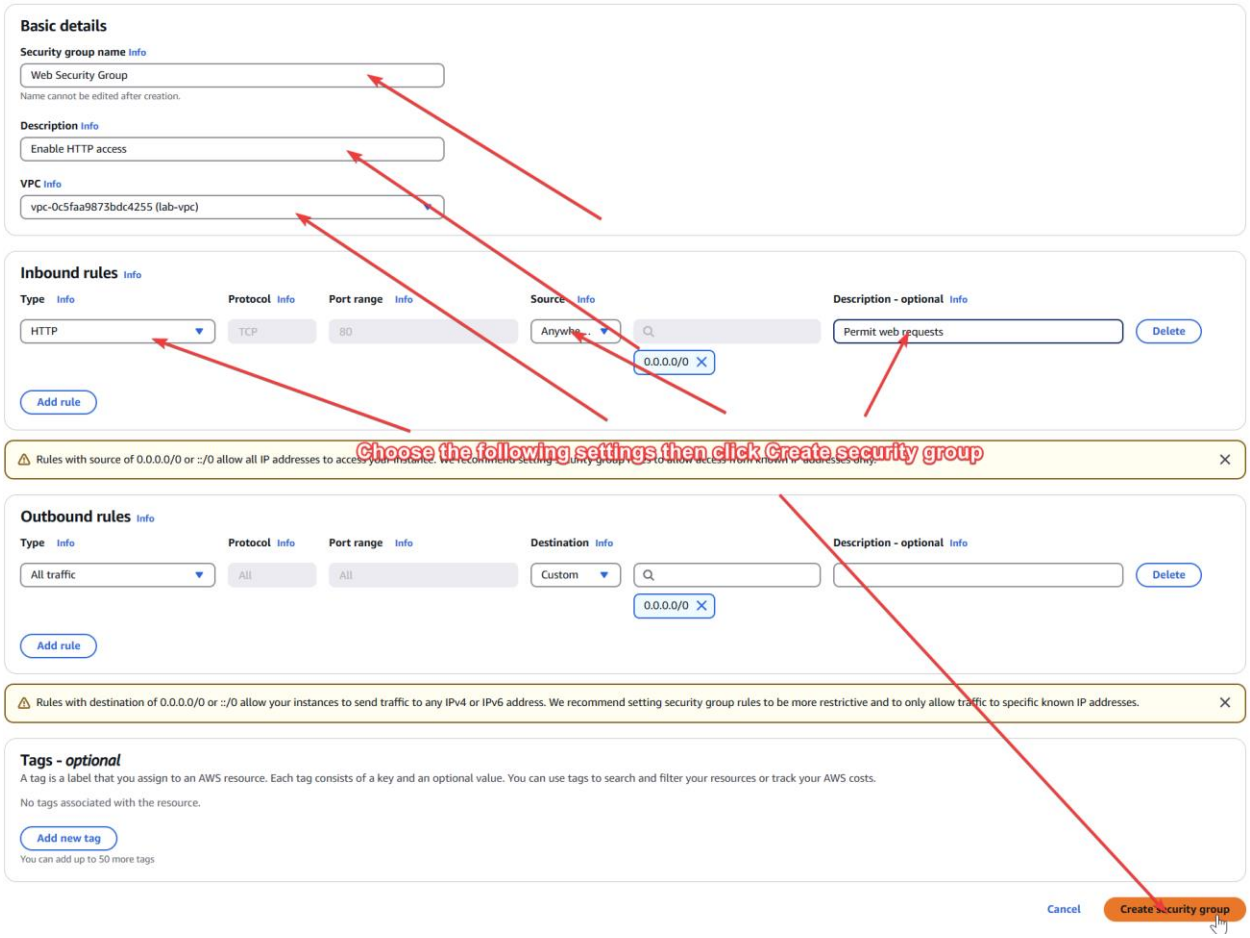
Now we will configure the inbound rules on the VPC, open the security group dashboard



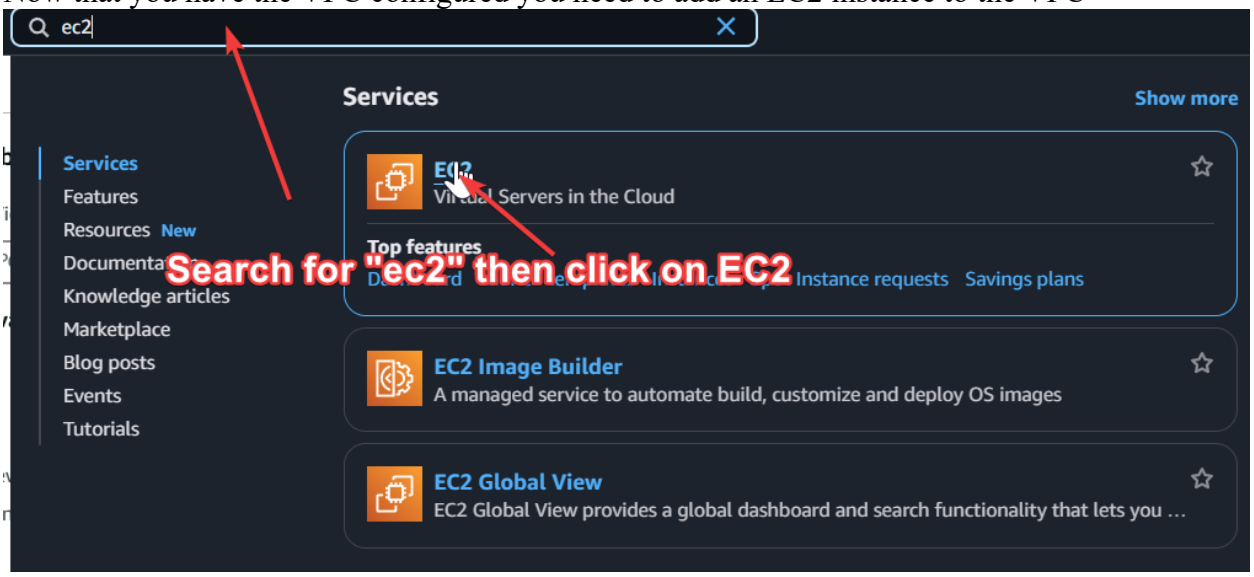
Now configure a security group



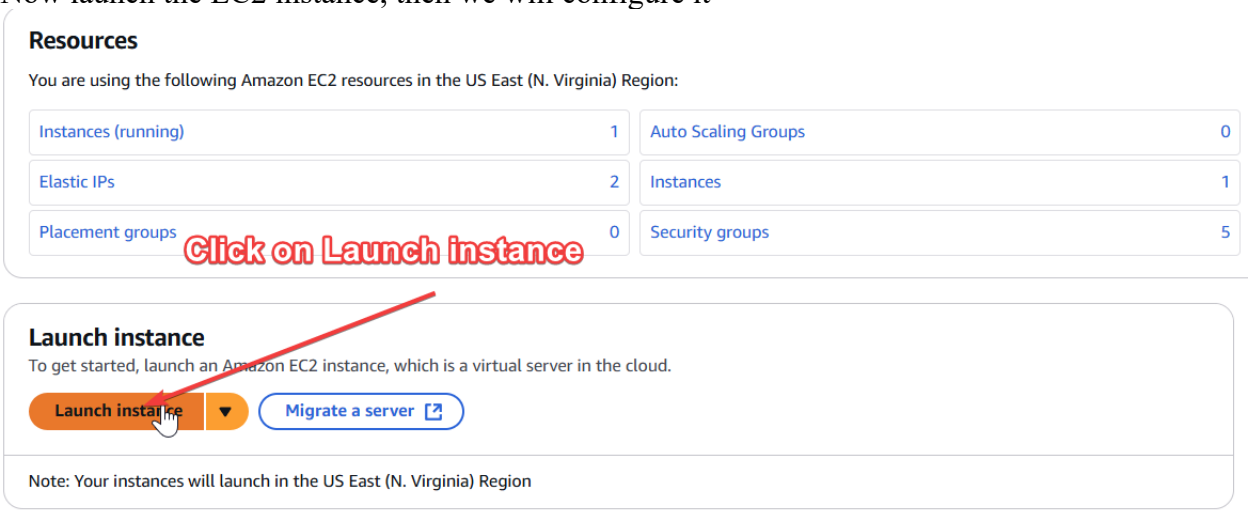
Now configure the basic settings for the security group, note that you only add an inbound HTTP rule



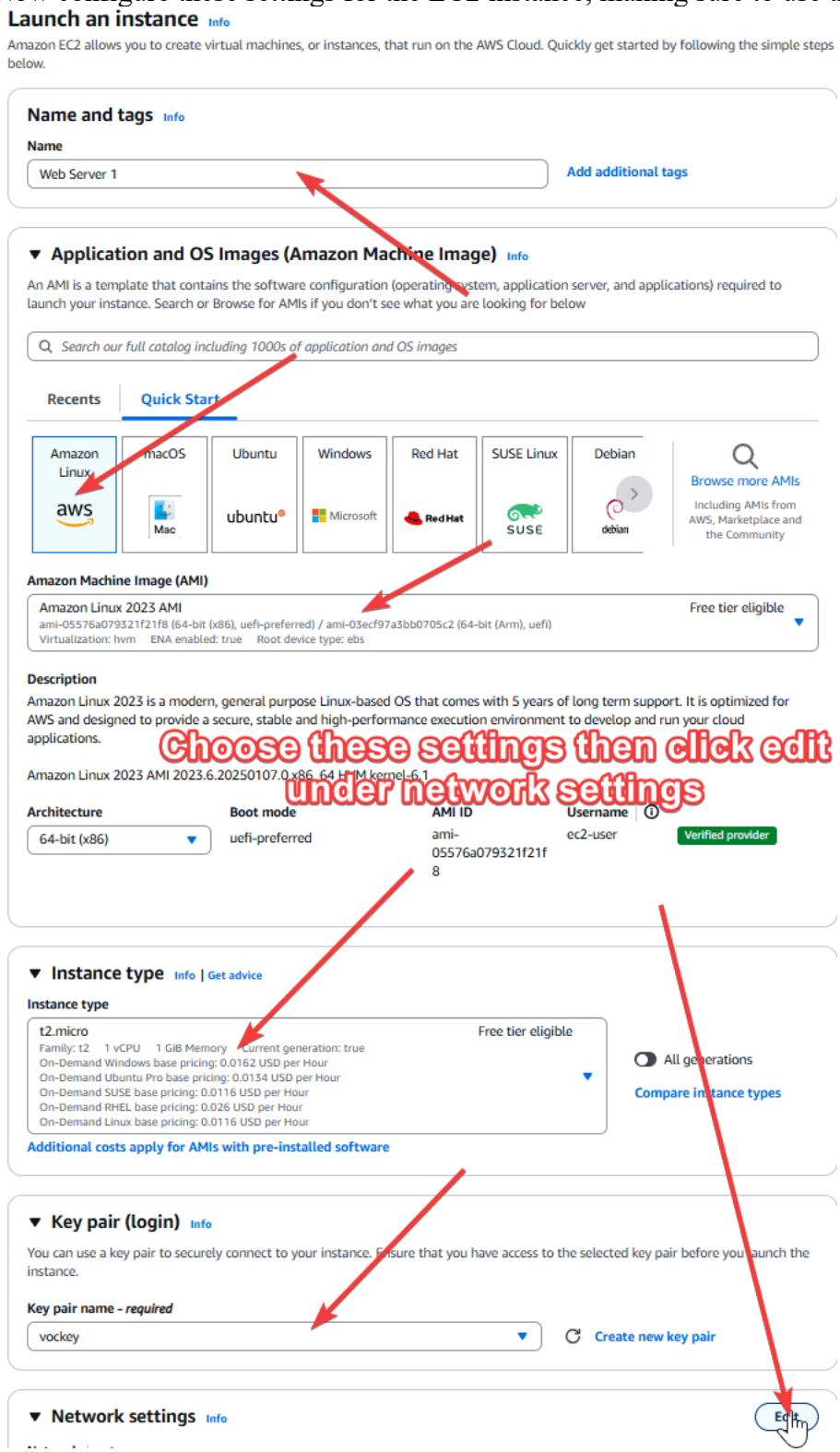
Now that you have the VPC configured you need to add an EC2 instance to the VPC



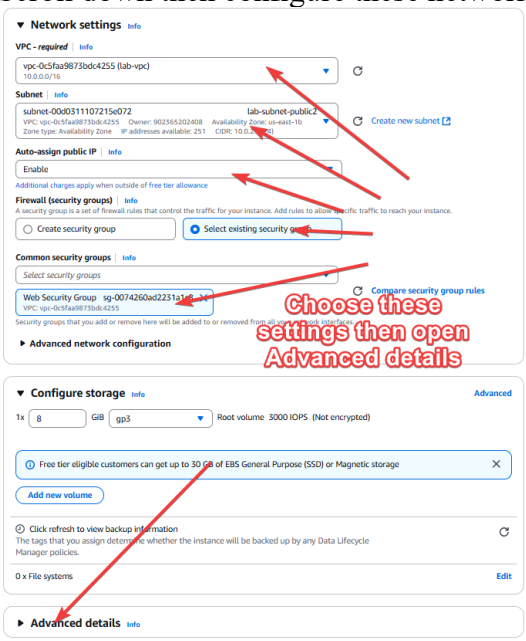
Now launch the EC2 instance, then we will configure it



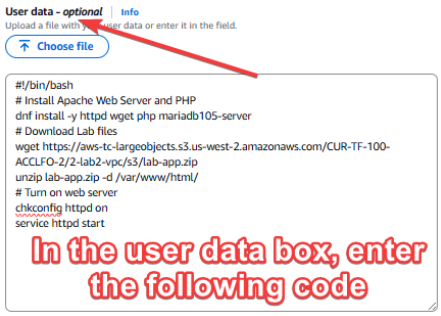
Now configure these settings for the EC2 instance, making sure to use the AMI Linux image



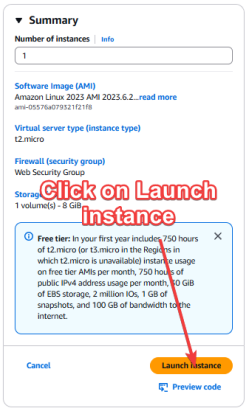
Scroll down then configure these network settings to control which subnet it is in the VPC



Scroll all the way down and finally enter the following code in order to boot the EC2 instance upon launch

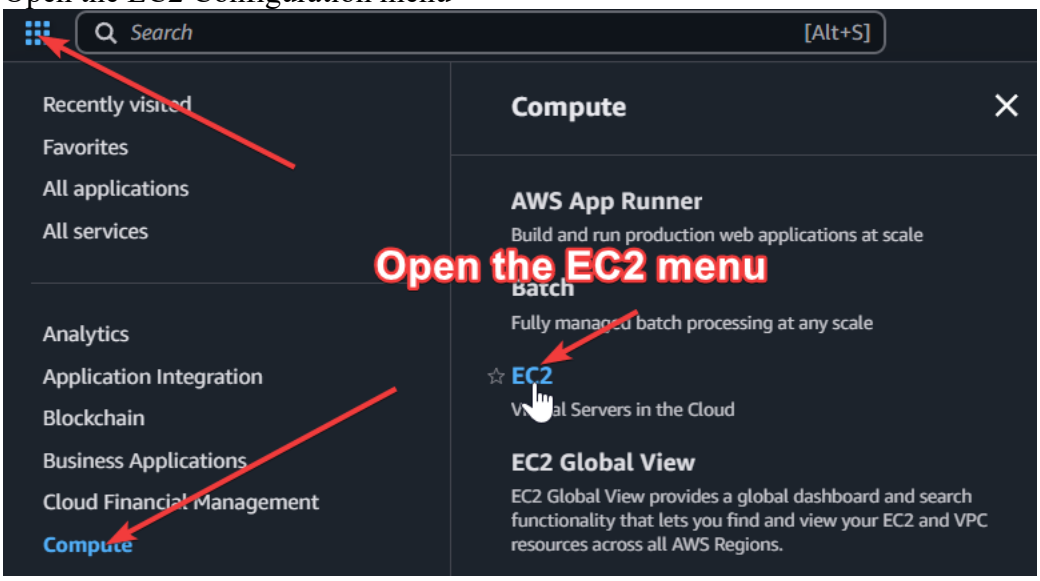


Finally in the right panel click launch instance, and the web server should be running

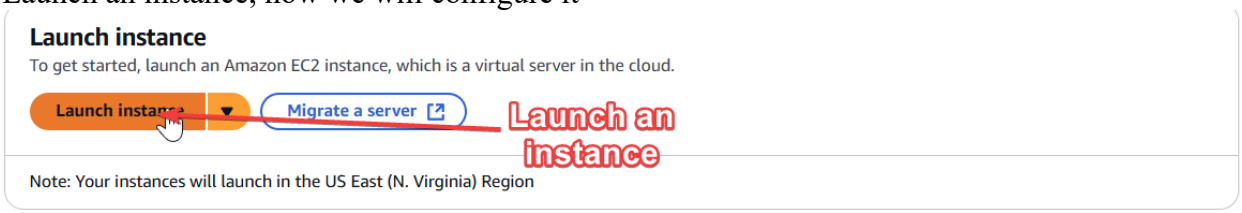


LAB 3:

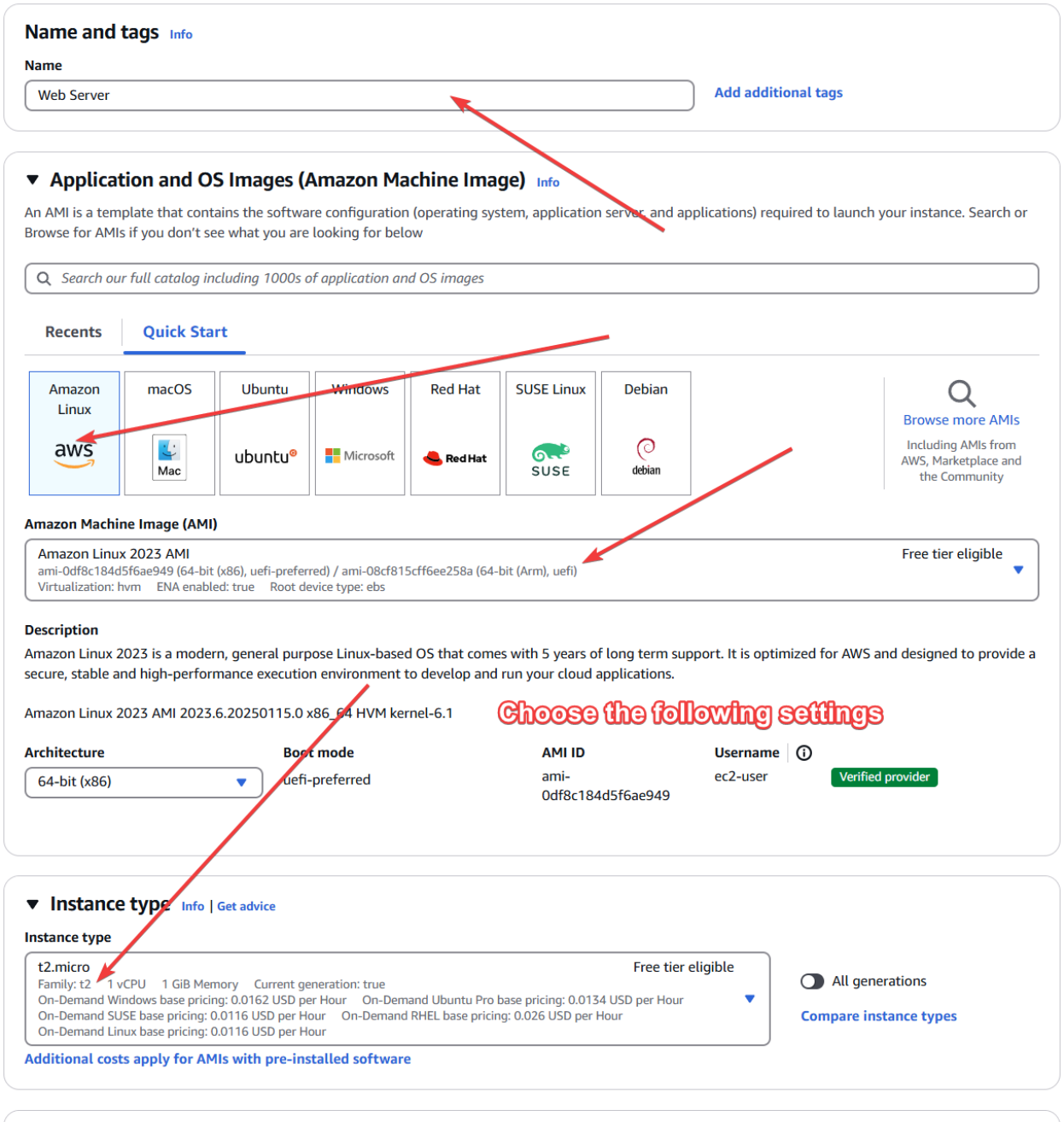
Open the EC2 Configuration menu



Launch an instance, now we will configure it

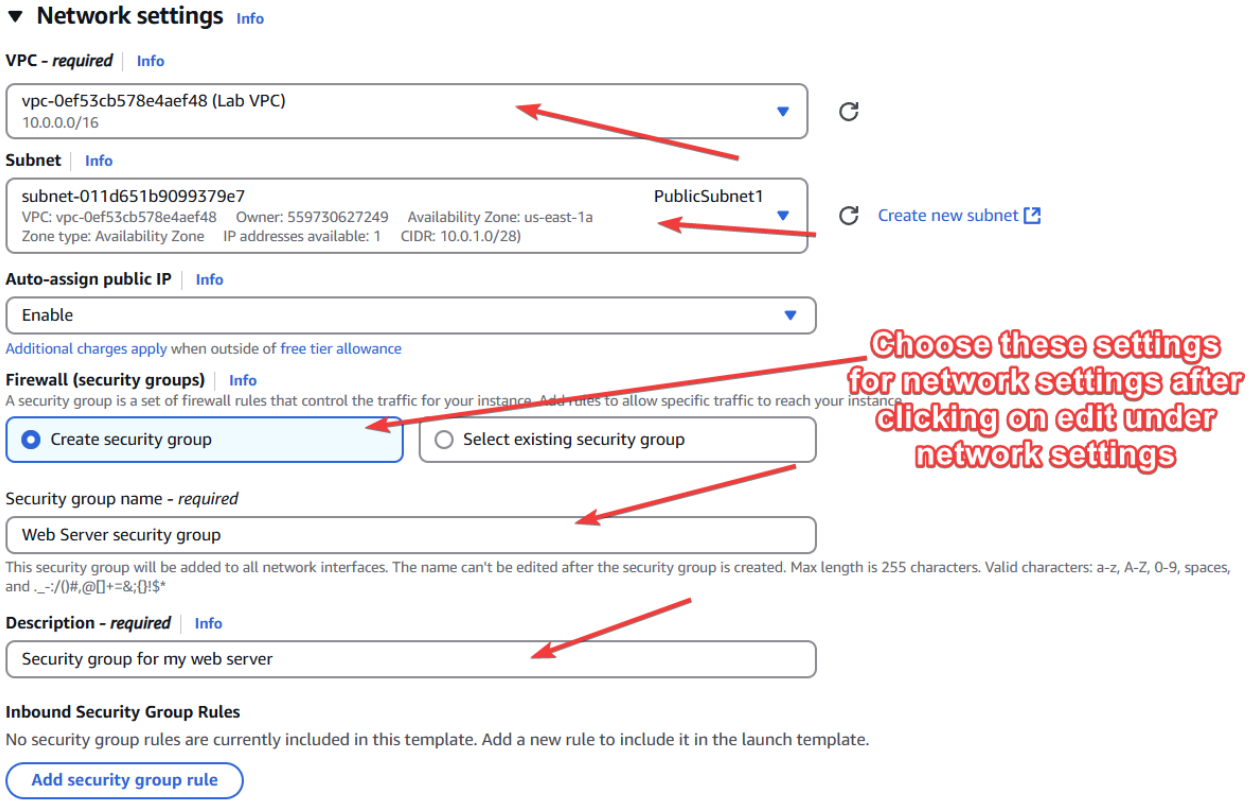


Start with the basic configuration, and make sure to choose the Amazon Linux AMI



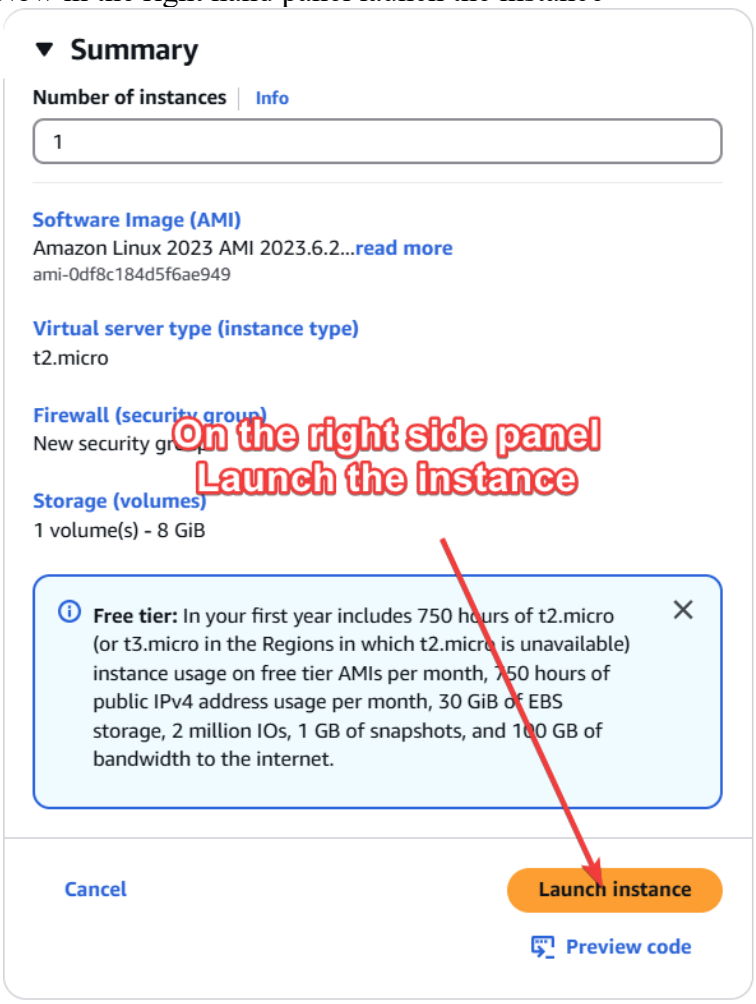
Enable termination configuration to stop it from turning off

In network settings configure the following settings to choose the VPC, subnet, and security group

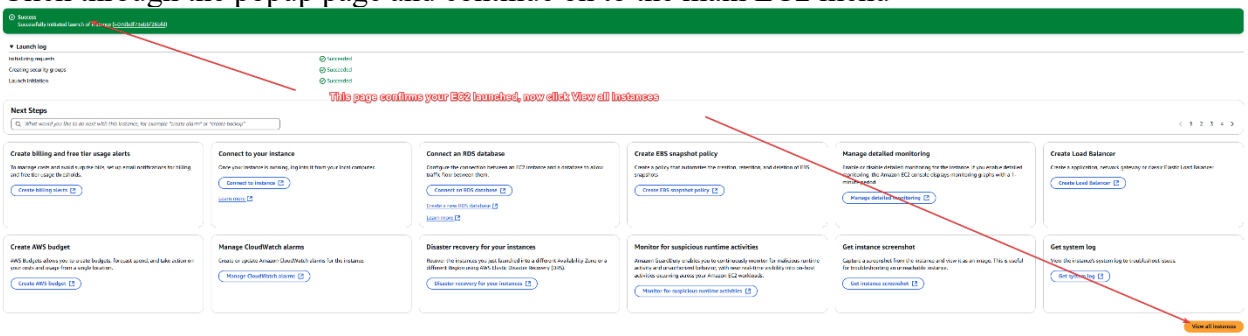


In the bottom of the page enter the following code to start the instance upon launch

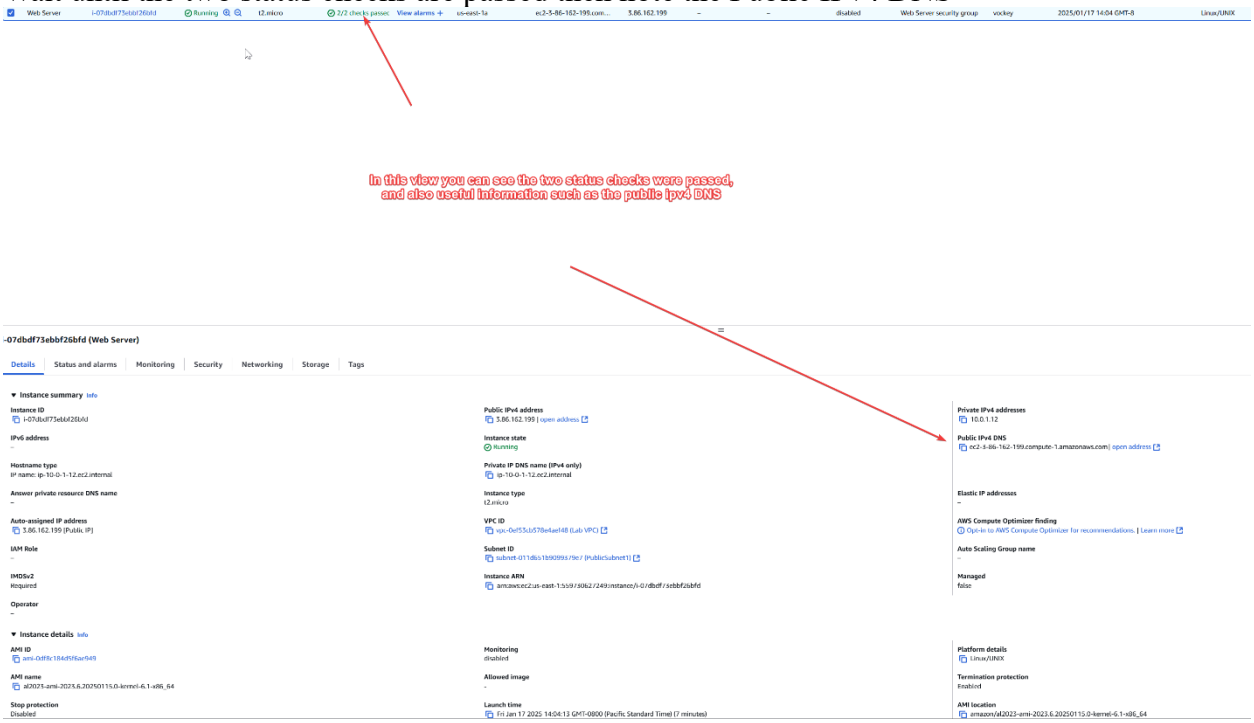
Now in the right hand panel launch the instance



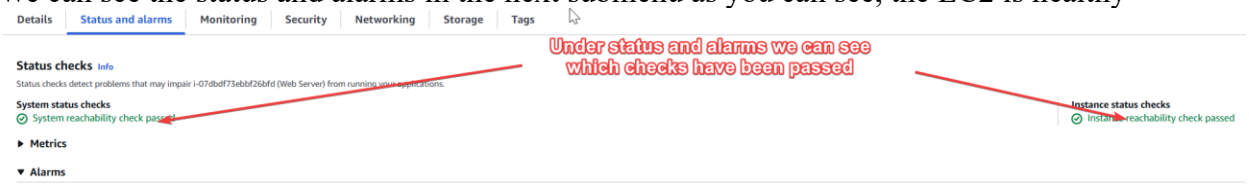
Click through the popup page and continue on to the main EC2 menu



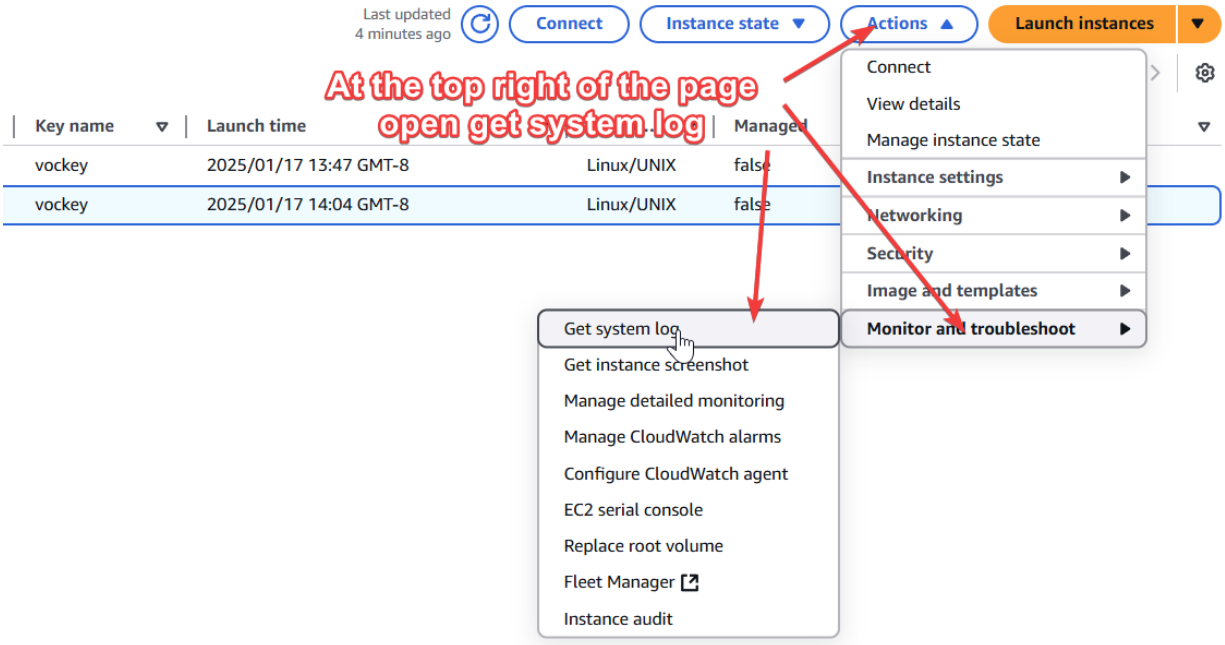
Wait until the two status checks are passed then note the Public IPv4 DNS



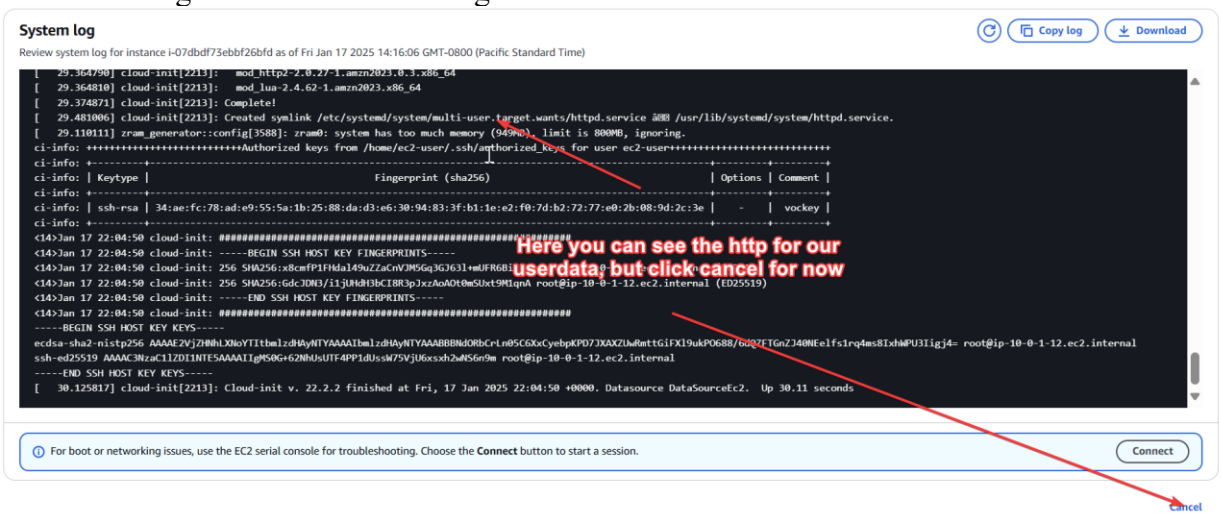
We can see the status and alarms in the next submenu as you can see, the EC2 is healthy



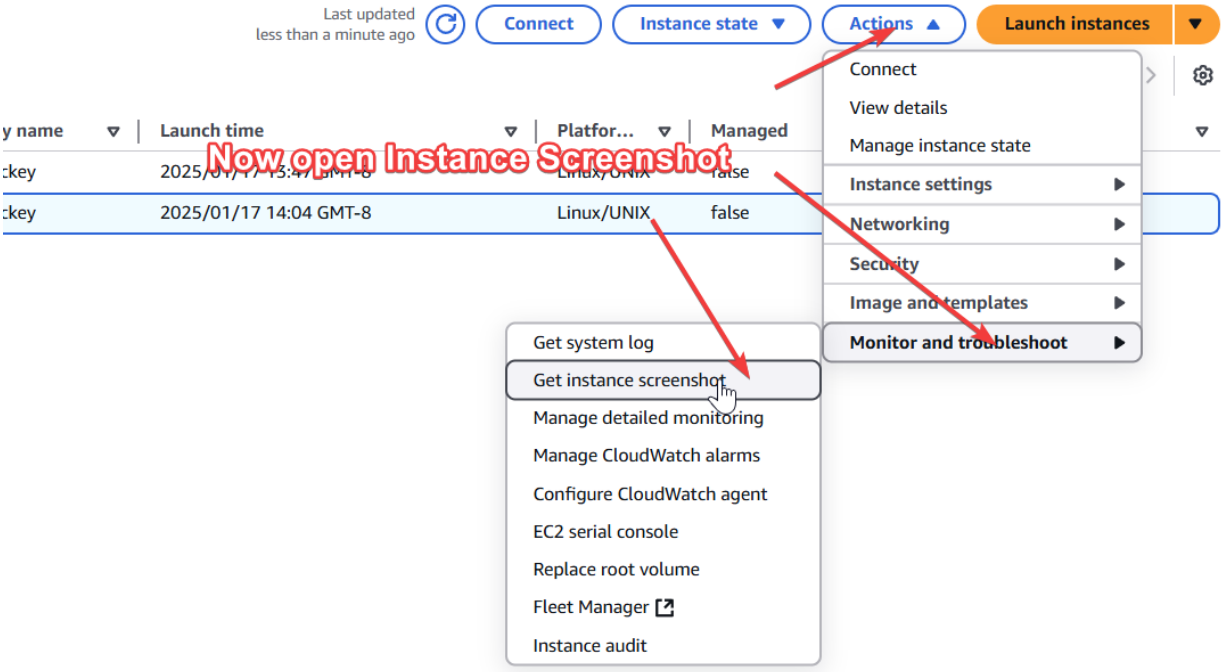
Now in the top open the system log to view the user data



As you can see the user data is visible in the system log, but we will continue on to the instance screenshot to get an idea of what the gui looks like

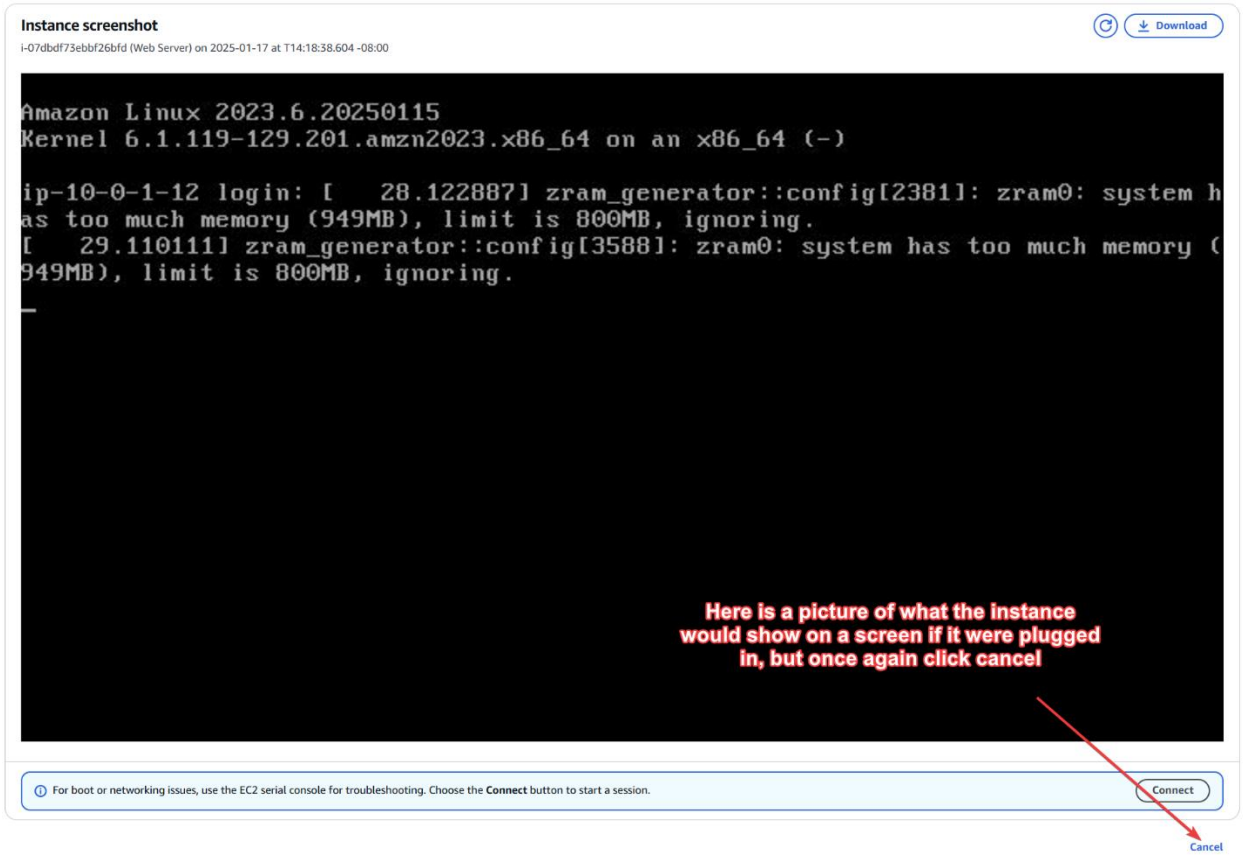


In the main menu, open the instance screenshot viewer

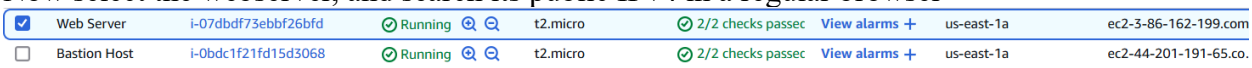


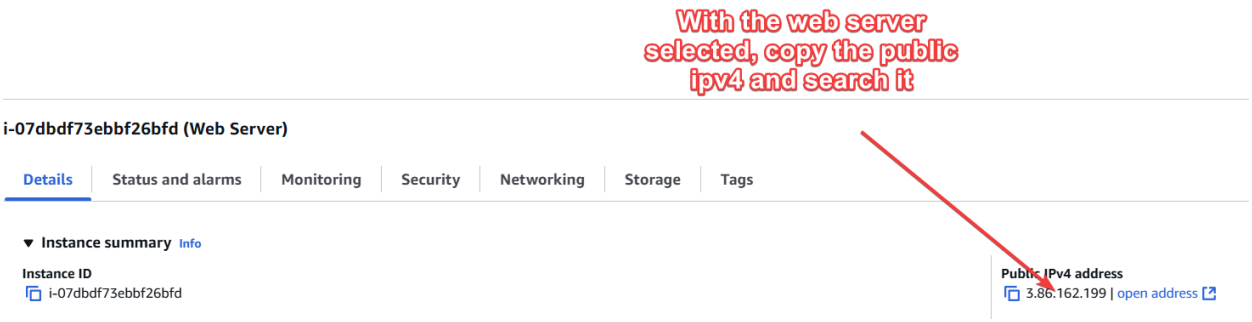
As you can see the linux AMI running doesn't actually have a GUI but rather just a command

line

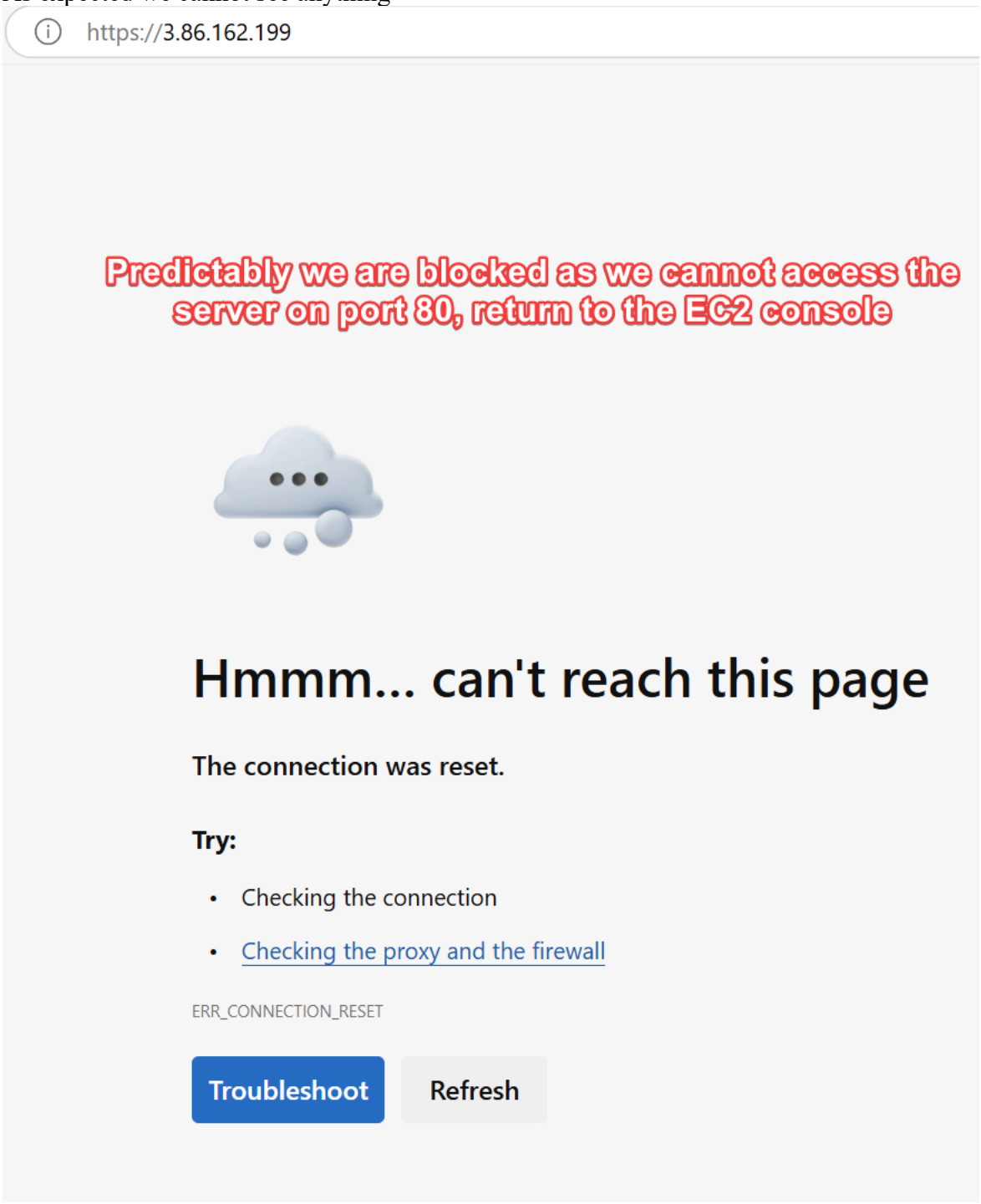

Instance screenshot
i-07dbdf73ebbf26bfd (Web Server) on 2025-01-17 at T14:18:38.604 -08:00
Download
Amazon Linux 2023.6.20250115
Kernel 6.1.119-129.201.amzn2023.x86_64 on an x86_64 (-)
ip-10-0-1-12 login: [28.122887] zram_generator::config[2381]: zram0: system h
as too much memory (949MB), limit is 800MB, ignoring.
[29.110111] zram_generator::config[3588]: zram0: system has too much memory (
949MB), limit is 800MB, ignoring.
Here is a picture of what the instance would show on a screen if it were plugged in, but once again click cancel
For boot or networking issues, use the EC2 serial console for troubleshooting. Choose the Connect button to start a session.
Connect
Cancel

Now select the webserver, and search its public IPv4 in a regular browser

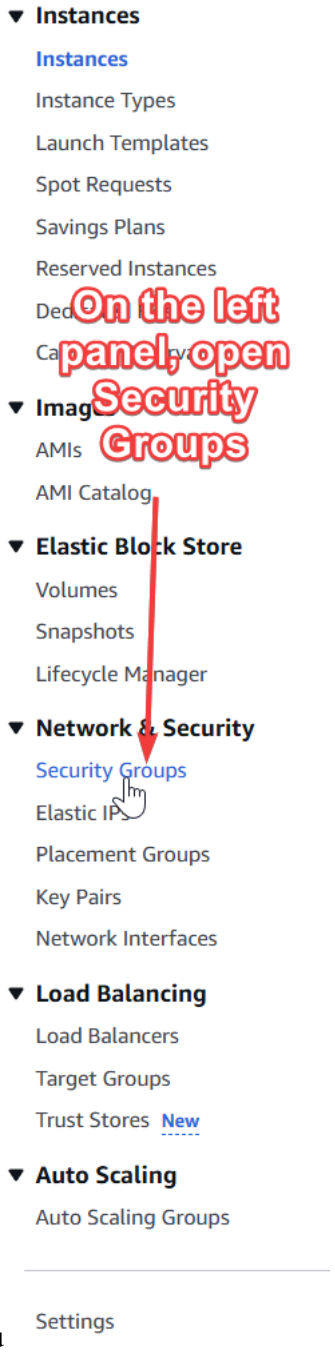




As expected we cannot see anything

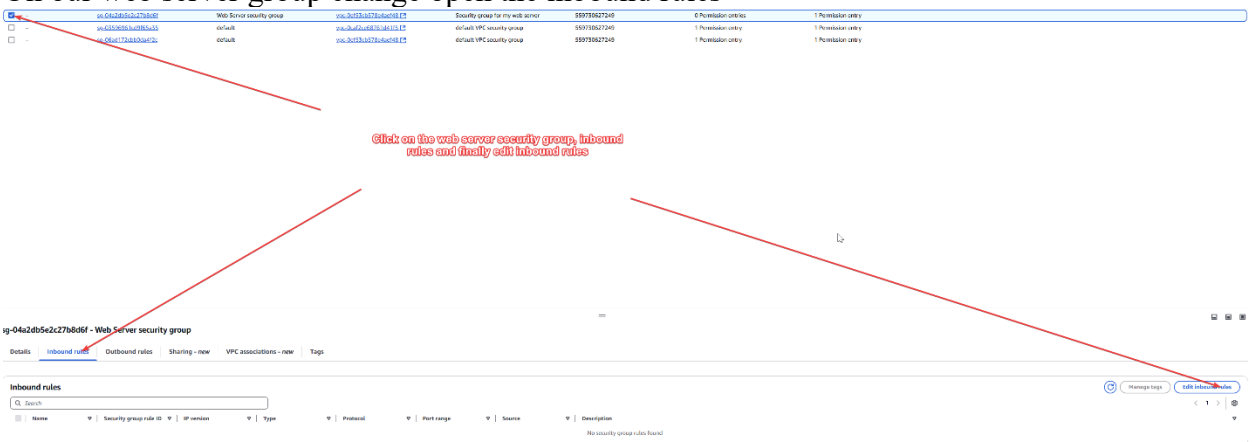


But we can change the security group inbound rules to change that, so open the security groups

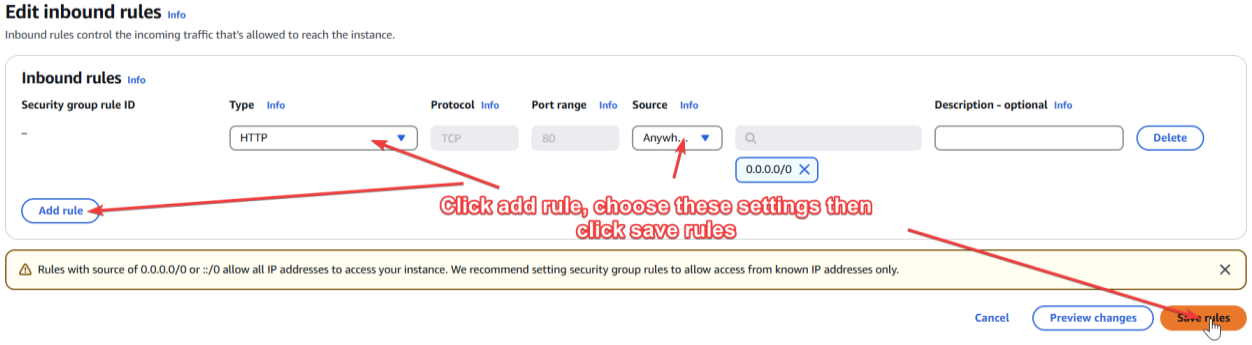


menu

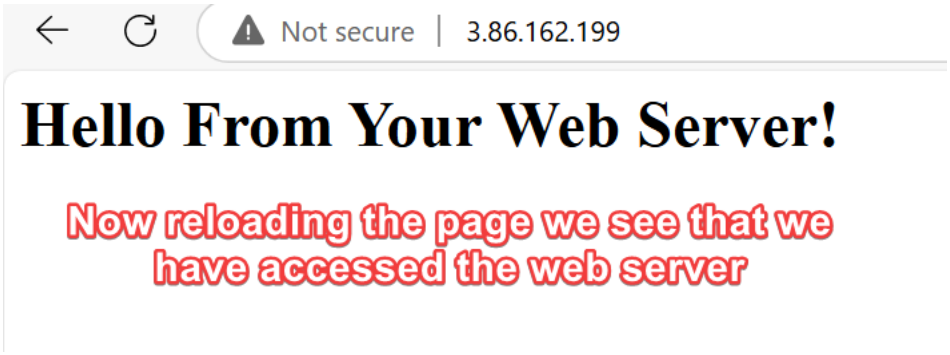
On our web server group change open the inbound rules



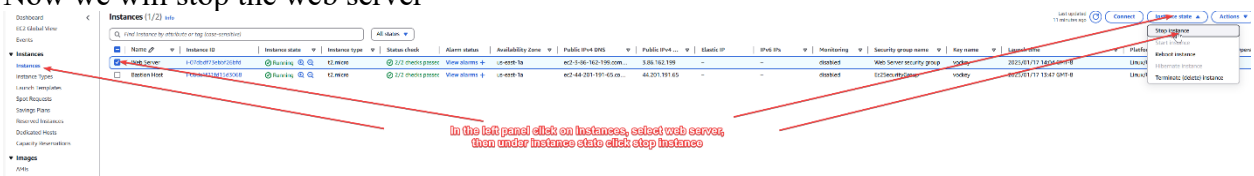
Edit the inbound rules to allow the HTTP connection through



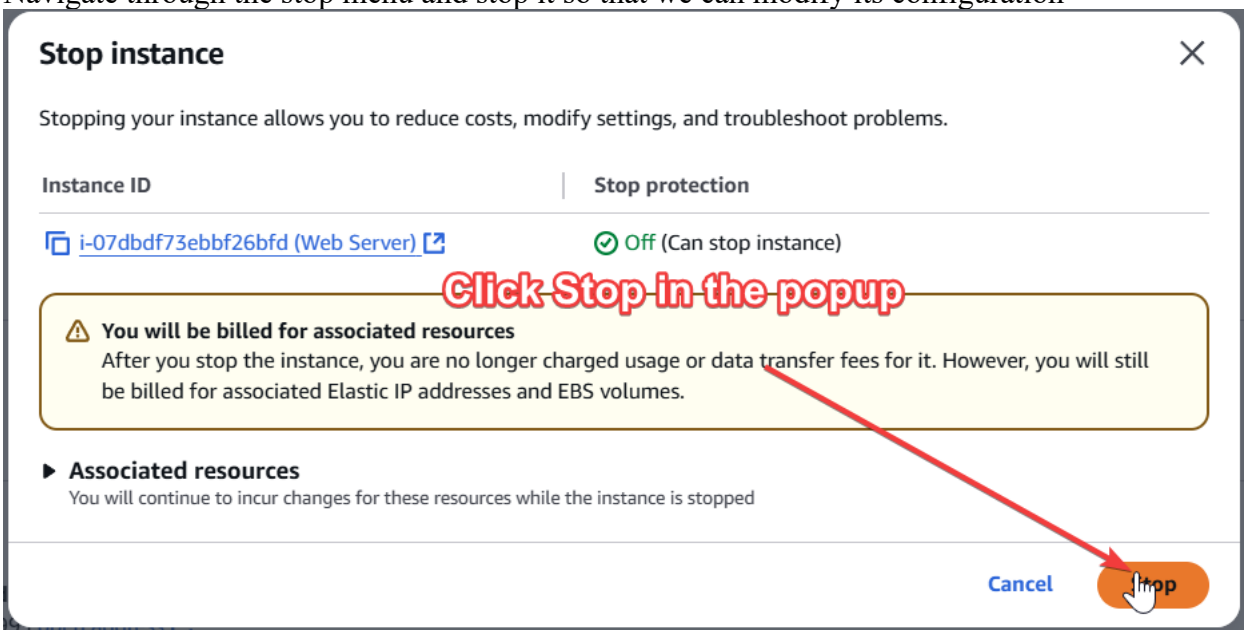
Now we can reload the web server and we see that we got through the firewall



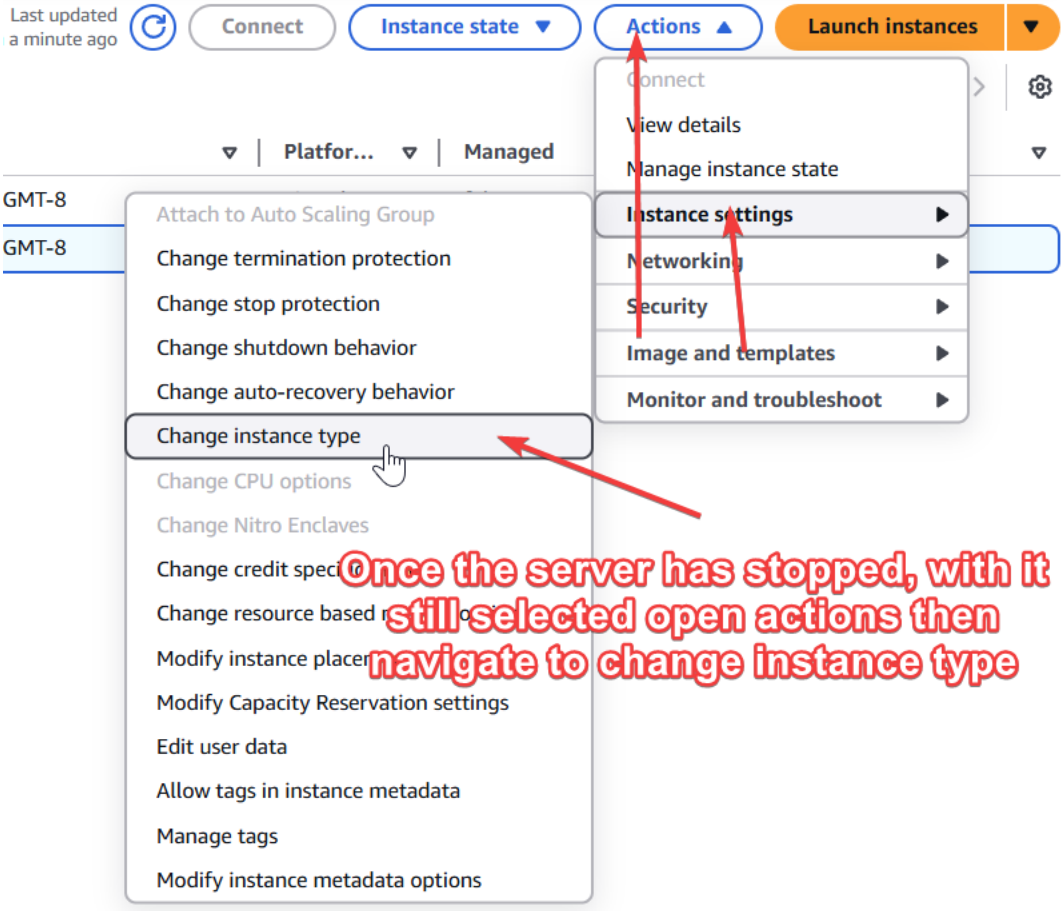
Now we will stop the web server



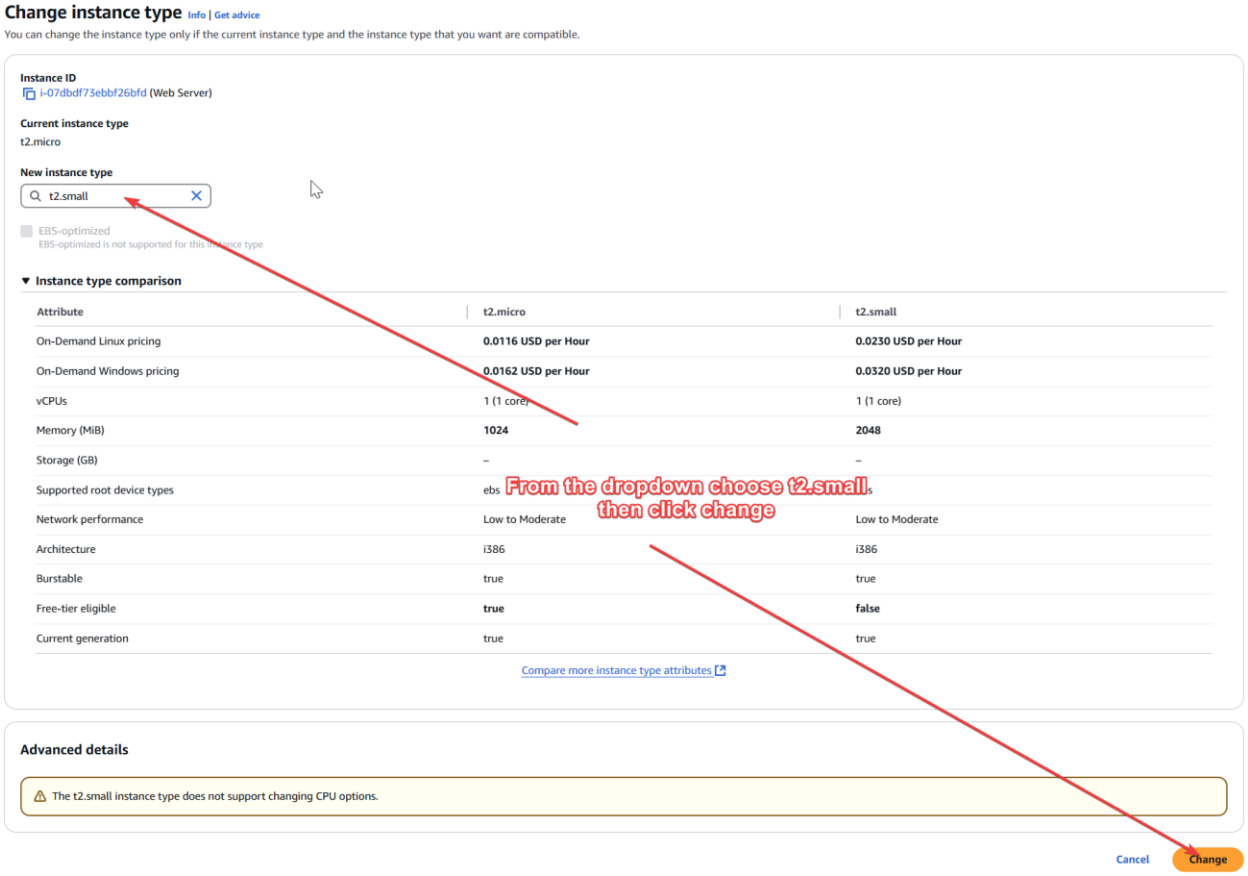
Navigate through the stop menu and stop it so that we can modify its configuration



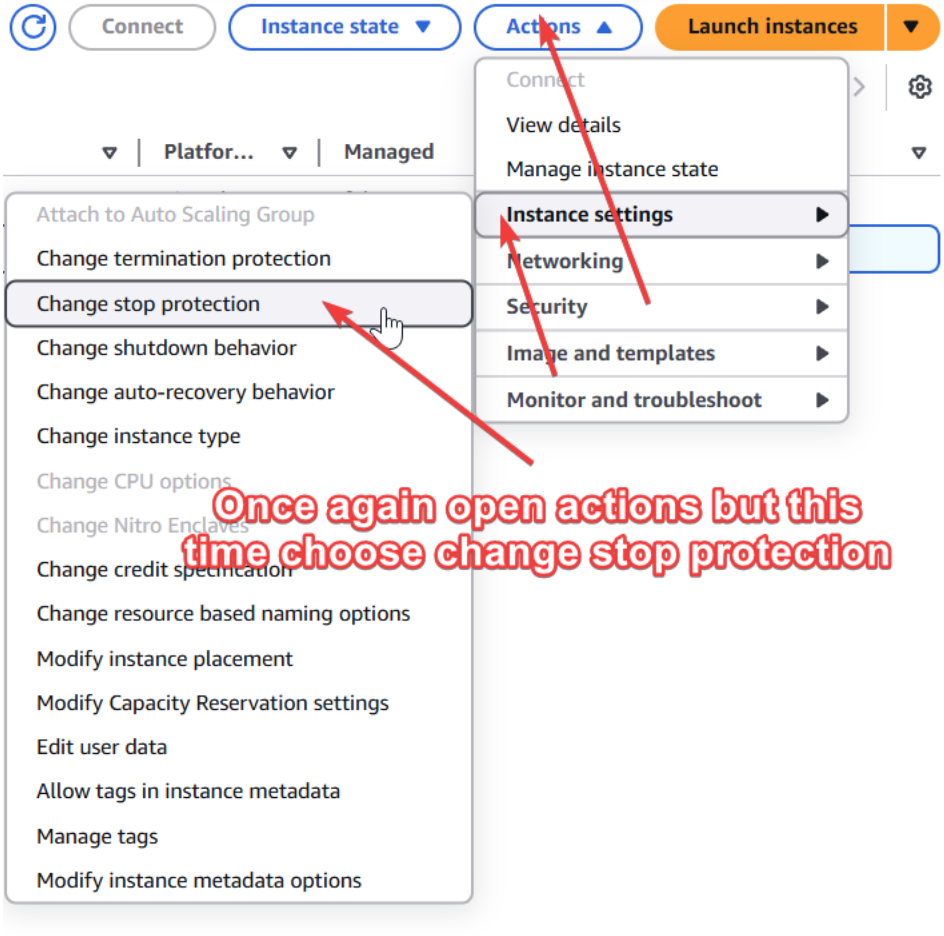
Back in the main menu change navigate through the actions menu to change the instance type



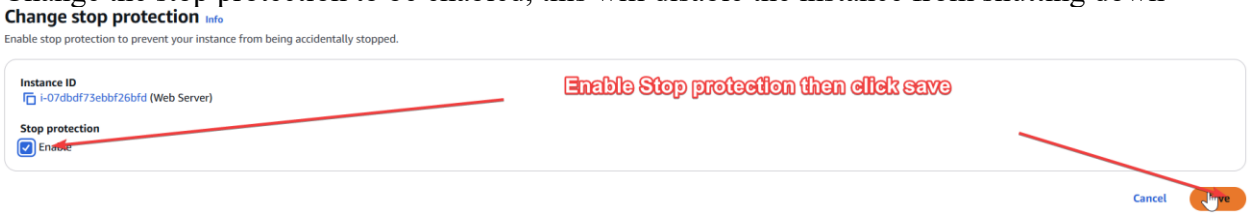
Make the instance a t2.small instance



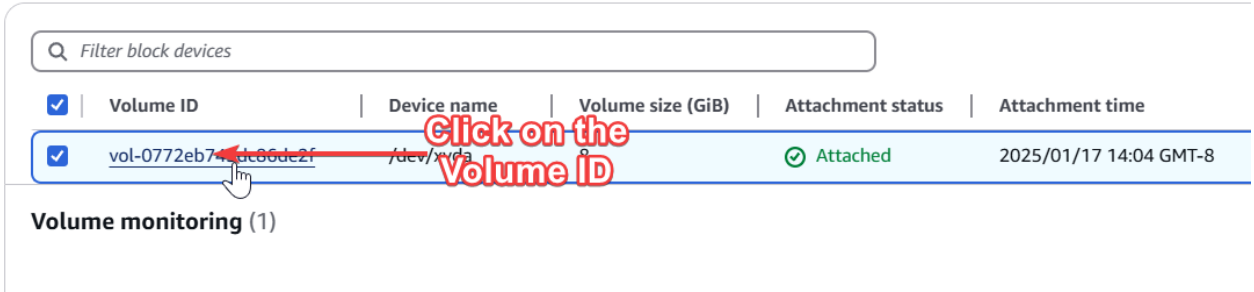
Now with the new instance navigate to change stop protection



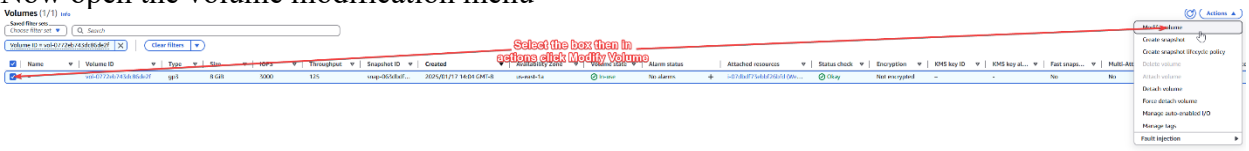
Change the stop protection to be enabled, this will disable the instance from shutting down



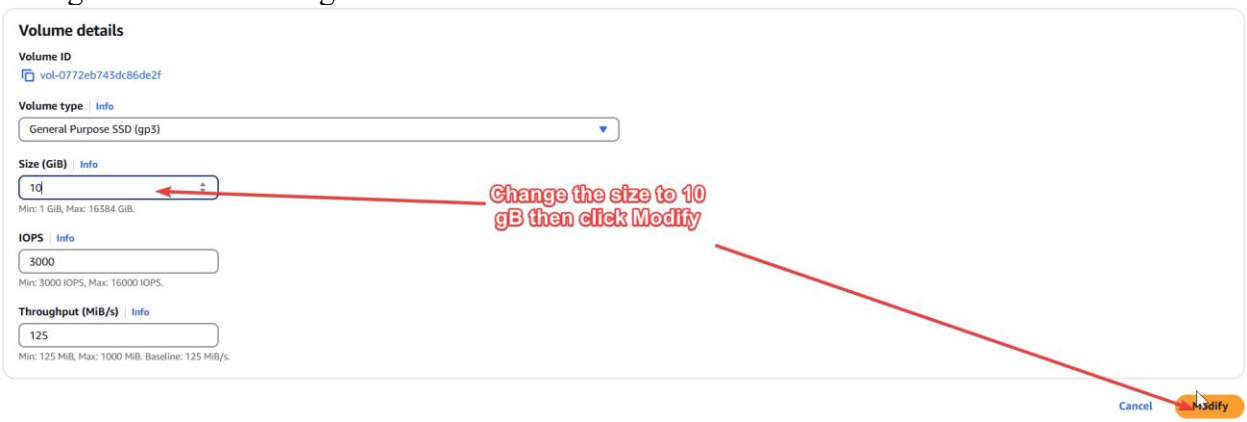
Now we will edit the storage configuration, select the volume



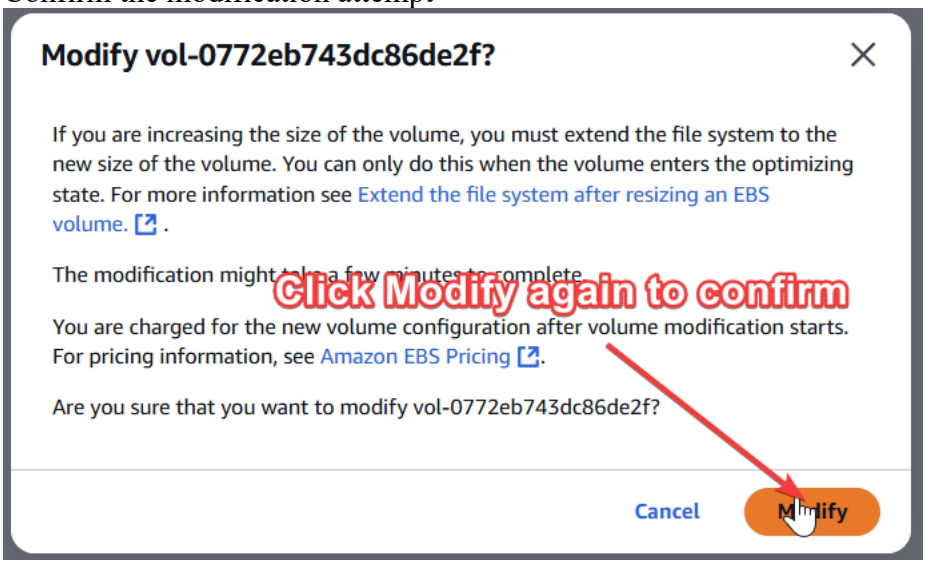
Now open the volume modification menu



Change the volume storage size

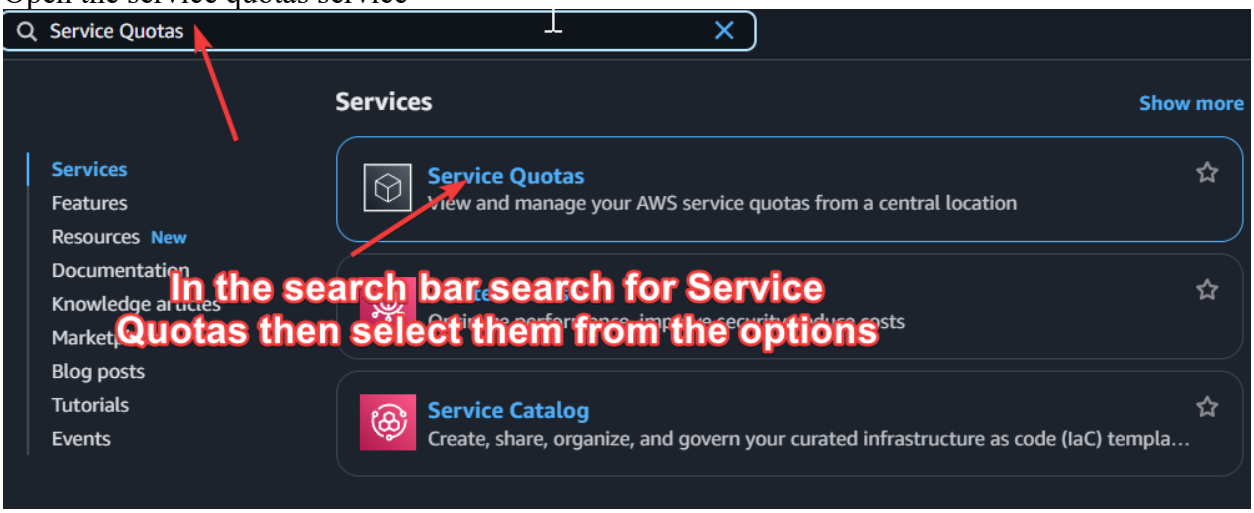


Confirm the modification attempt

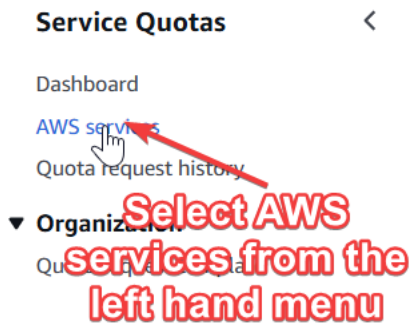


Now start the instance back up

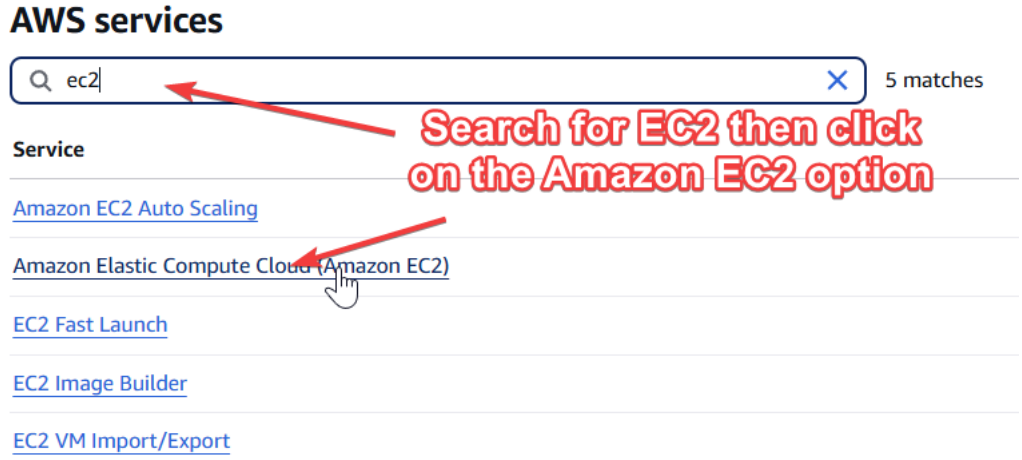
Open the service quotas service



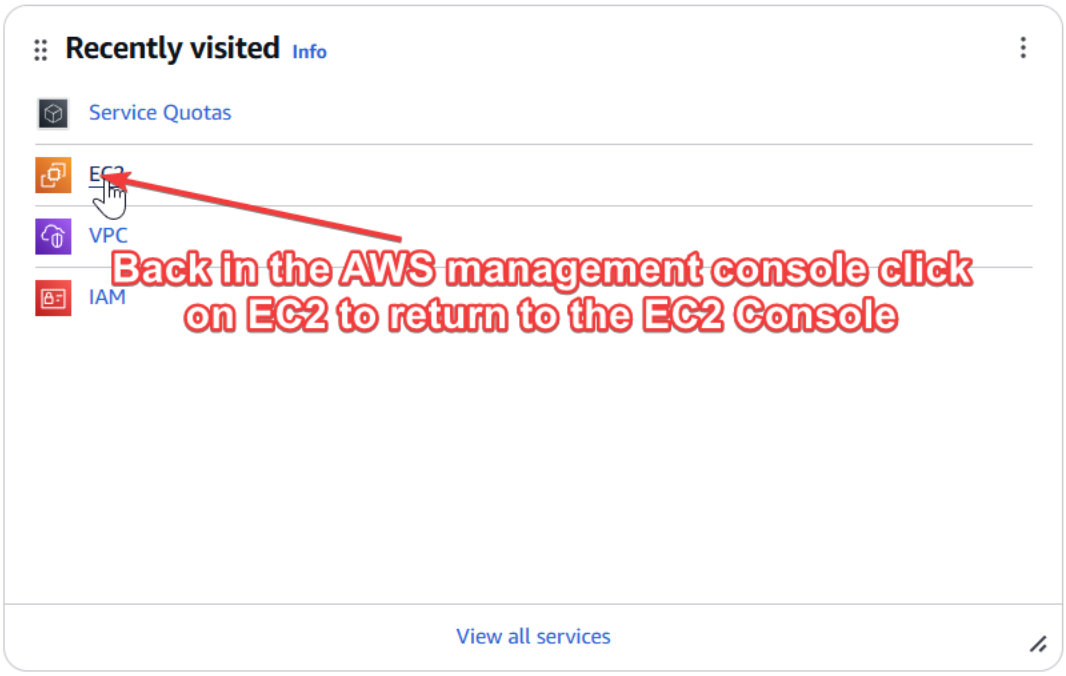
Select the AWS services menu from the left panel



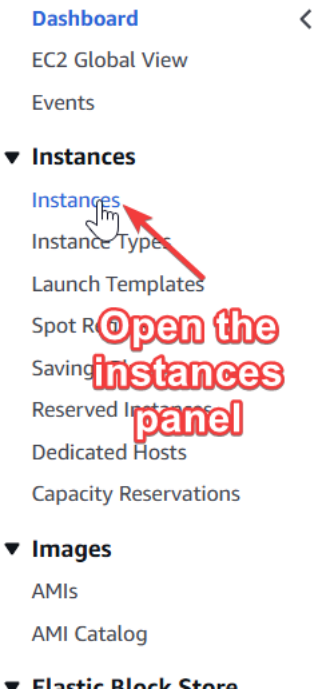
Now search for EC2 instances



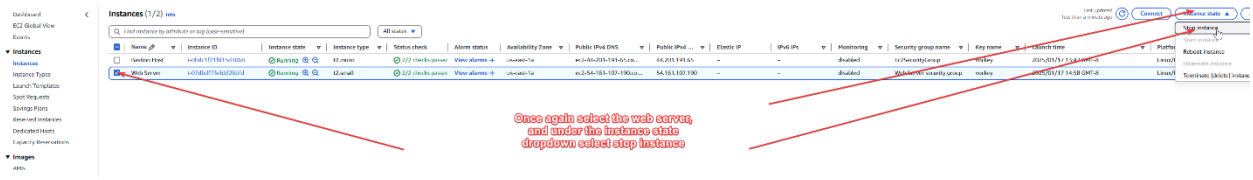
Tab back to the AWS management console and reopen the EC2 menu

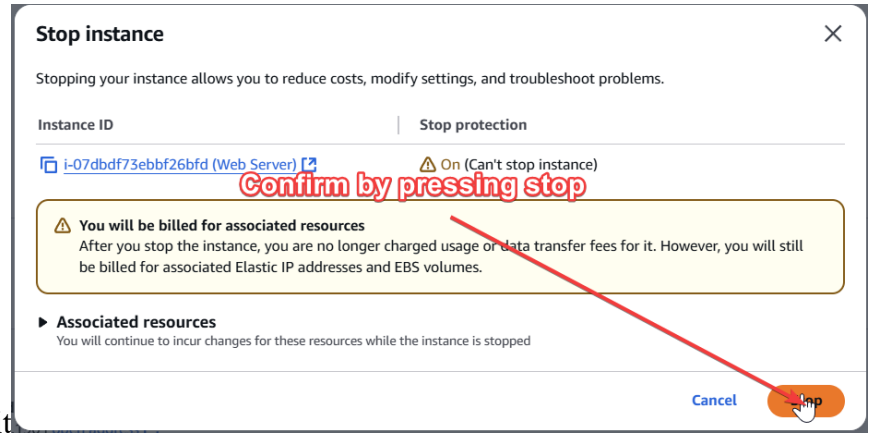


Open the instance view panel



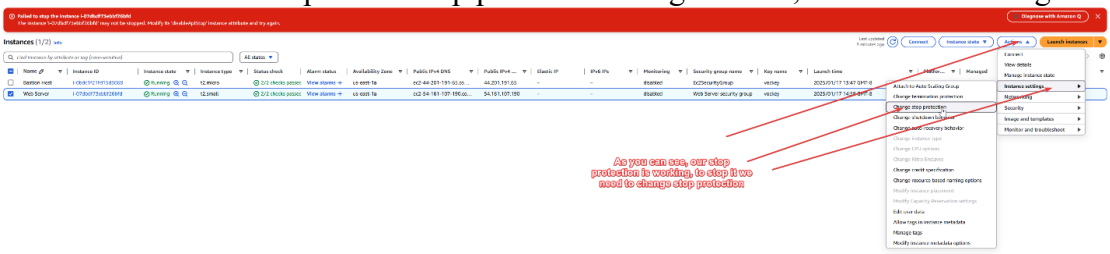
Now you will try to stop the Web Server instance again



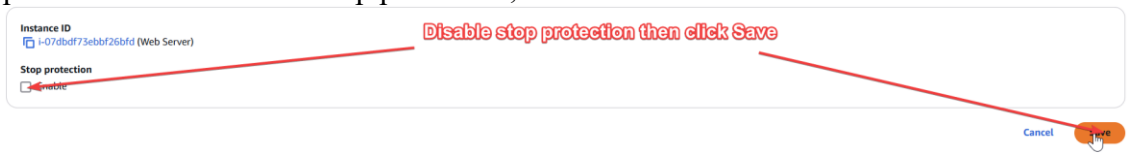


Confirm your choice to stop it

The instance fails to stop due to stop protection being enabled, now we will change this

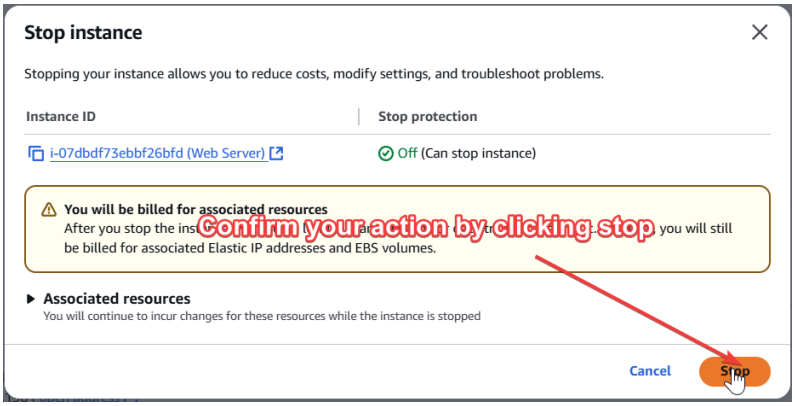


In the stop protection menu disable stop protection, remember this will enable shutoff



Now try to once again stop the instance

Confirm your choice, now the instance will stop



Problems

I did encounter problems that crunched my time including the lab not starting and having to wait forever on the yellow light and then accidentally not using a private window which logged me out of the main aws console.

Conclusion

These labs were incredibly valuable in teaching me the fundamental and core features of AWS. The lab documentation provided a smooth learning experience, with fewer challenges than I had expected. Overall, the lessons on IAM, VPCs, launching a web server, and EC2 have given me a solid foundation in cloud services, particularly IaaS, and a deeper understanding of AWS's unique features. Given that AWS is one of the most widely used cloud service platforms, this knowledge will be extremely beneficial as I continue to learn, both academically and professionally.

CCNP Portfolio
AWS 4-6 LABS

Blizzard, Harrison J

Purpose

The main purpose of these last three labs was to build upon the foundational knowledge gained in the first three, while also introducing new concepts essential for fully understanding AWS's capabilities. These labs covered AWS's EBS features, the process of building a database server, and the scaling and load balancing architecture within AWS. Overall, they served to enhance our expertise, particularly in key areas not addressed in the initial labs, and further developed our understanding of AWS's core functionalities.

Background Information/Lab Concepts

Amazon Web Services (AWS) emerged as a leader in the Infrastructure as a Service (IaaS) industry in 2006, building upon its internal use at Amazon in the early 2000s. AWS revolutionized the industry by offering cost-effective, scalable, and flexible network infrastructure, which appealed not only to large government networks but also to small startups. The use of Availability Zones enabled AWS to provide global connectivity with high redundancy, ensuring reliability and scalability. This combination of effectiveness and flexibility has made AWS a prominent force in the IaaS industry, with millions relying on it for network connectivity.

Amazon Elastic Block Store (EBS) is a block-level storage service designed to be used with Amazon EC2 instances. EBS provides persistent storage, meaning that data stored on an EBS volume is retained even if the instance it’s attached to is stopped or terminated. EBS is ideal for applications that require frequent updates and data integrity, such as databases or file services. Introduced in 2008, EBS was developed as a solution to the limitations of local instance storage, which could not retain data across reboots. EBS became a high-performance, persistent storage option to address this gap.

EBS volumes can be attached to EC2 instances to increase local storage, and these volumes can be formatted and accessed like traditional hard drives. EBS is highly scalable, allowing users to resize volumes, adjust performance, and change volume types (e.g., General Purpose SSD) based on specific needs. EBS is also designed for redundancy, as it is replicated within Availability Zones to ensure durability.

A database server on AWS typically refers to an EC2 instance running a database management system (DBMS) such as MySQL. AWS also offers fully managed database services like Amazon RDS (Relational Database Service) and Amazon Aurora, which simplify database setup and maintenance. These managed services are ideal for those who want to focus on their applications without dealing with complex technical tasks.

The history of database servers on AWS began in 2009 with the introduction of Amazon RDS. Prior to RDS, customers had to manage databases on EC2 instances, handling tasks such as backups, scaling, patching, and security themselves. RDS automated many of these administrative tasks, freeing developers to focus on application development. RDS became

popular due to its ease of use and scalability. In 2015, Amazon Aurora was launched as a high-performance, fully managed relational database that works alongside MySQL. Aurora offered better scalability and availability at a lower cost than other database solutions.

AWS provides flexibility in how database servers are managed. Customers can choose to use EC2 for self-managed database setups, allowing full control over the environment, or use Amazon RDS and Aurora for a fully managed experience. RDS and Aurora handle tasks like backups, patching, and scaling automatically, while also supporting multi-Availability Zone deployments for high availability, ensuring databases remain online even if one zone fails.

Scaling and load balancing are essential practices in AWS for ensuring that applications can handle varying traffic levels without compromising performance or availability. Auto Scaling dynamically adjusts the number of EC2 instances based on demand, while Elastic Load Balancing (ELB) distributes incoming traffic across multiple instances to maintain a balanced load and ensure high availability.

Introduced in 2009, ELB addresses the challenge of distributing traffic evenly to application servers. It automatically routes traffic to healthy instances in one or more Availability Zones, ensuring no single instance becomes overwhelmed. ELB supports both HTTP/HTTPS traffic and other types of network traffic, providing the flexibility needed for different use cases. Auto Scaling, on the other hand, allows applications to scale automatically based on metrics such as CPU usage or incoming request rates.

Together, Auto Scaling and ELB enable companies to scale applications dynamically based on traffic patterns. This approach ensures efficient use of resources, preventing overuse while maintaining application performance and user experience. As traffic rises, Auto Scaling launches additional EC2 instances, and ELB ensures those instances are added to the pool of servers handling requests. This scalable, fault-tolerant infrastructure allows businesses to meet demand without incurring unnecessary costs or compromising performance.

Lab Summary

To start the fourth lab, which focused on enhancing our skills with EBS, I began by creating and attaching an Amazon EBS volume to a new Amazon EC2 instance. I accessed the EC2 console and navigated to the "Volumes" section to create and configure a new volume. To attach the volume, I used the "Actions" menu within the EC2 volume settings. After that, I went to the "Instances" section, selected my instance, and clicked "Connect" in the instance connect tab. Using a few commands, I created and mounted the new storage volume. Next, I returned to the EC2 console and created a snapshot of my volume through the "Actions" menu. In the instance

connect session, I deleted the file I had created on the volume. For the final task, I created a new volume from the snapshot using the "Actions" menu in the EC2 console. After it was created, I again used the "Actions" menu to attach the volume and then mounted it through the instance connect session with a few commands.

In lab five, I started by creating a security group in the VPC to allow access from the web server to the RDS instance. Next, I created a DB subnet group, selecting subnets from two different Availability Zones. I then launched a Multi-AZ MySQL RDS instance with 20GB of SSD storage in the Lab VPC, ensuring the database was accessible from the web server. Once the database was available, I copied the endpoint and connected it to a web application. To test the application, I added, edited, and removed contacts, which were stored in the database and automatically replicated across the Availability Zones.

For the sixth lab, I began by creating an Amazon Machine Image (AMI) from the existing Web Server 1 to be used by the Auto Scaling group. I then set up an Application Load Balancer (ALB) and a Target Group to route traffic. Following that, I created a Launch Template for EC2 instances and configured an Auto Scaling Group to use the AMI, setting the scaling settings between 2 and 6 instances. I attached the Auto Scaling group to the ALB to distribute traffic. After verifying that the load balancer was functioning correctly, I tested Auto Scaling by generating CPU load, which triggered the creation of additional instances. Finally, I terminated Web Server 1, as it was no longer required.

Lab Commands

LAB 4:

TASK 1:

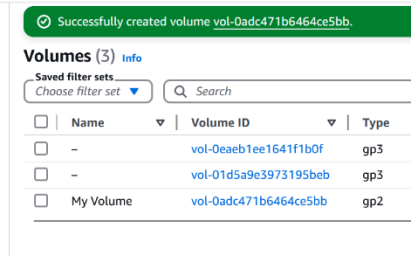
In the AWS management console use the search box in the top left to search and select EC2

In the left navigation panel, choose the Instances option, The availability zone should be similar to: us-east-1a

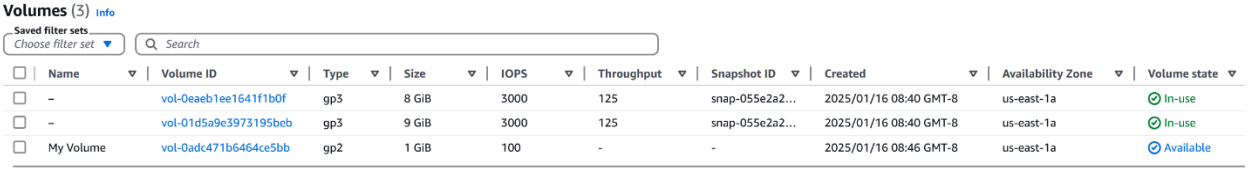
In the left navigation panel choose Volumes, then choose create volume in the top right corner

After choosing create volume, change volume type to *General Purpose SSD (gp2)*, Size (GiB) to *1*, and for Availability Zone choose the same availability zone as you noted in your EC2 instance

First chose to ‘Add tag’, then once inside the Tag Editor enter *Name* as Key and *My Volume* as value. Finally, select ‘Create Volume’ at the bottom

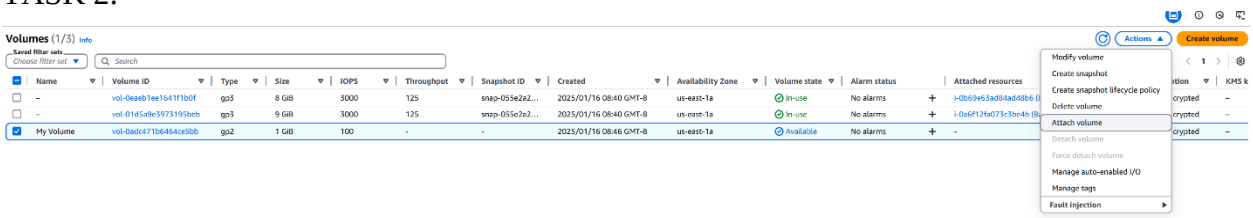


TASK 2:

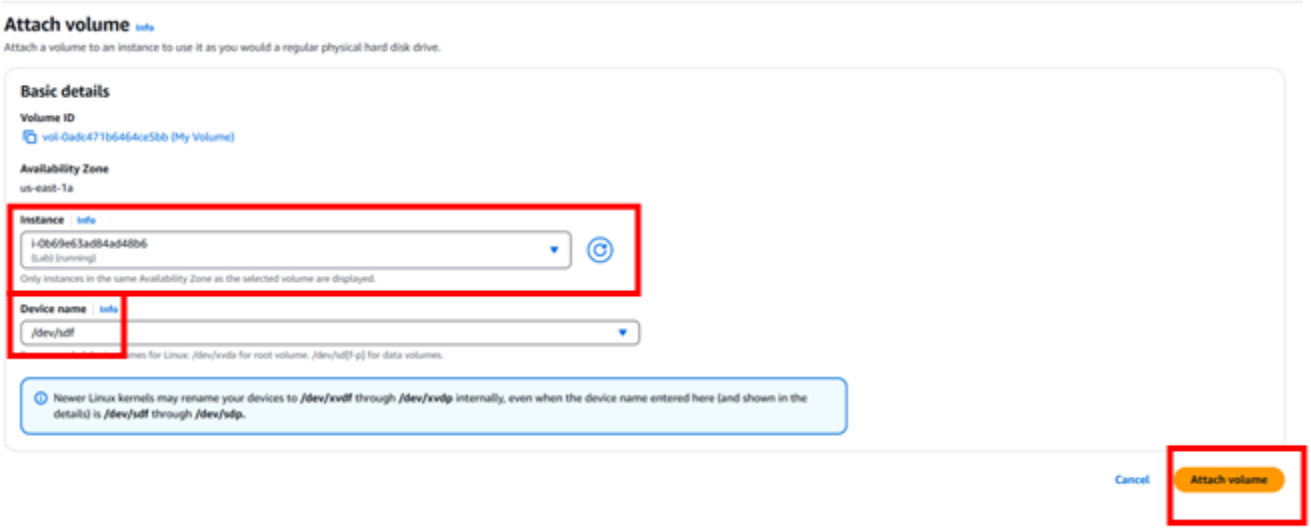


The new volume you created should appear and be named My Volume, after waiting and hitting refresh in the top right of the console, the volume state should turn to *Available*

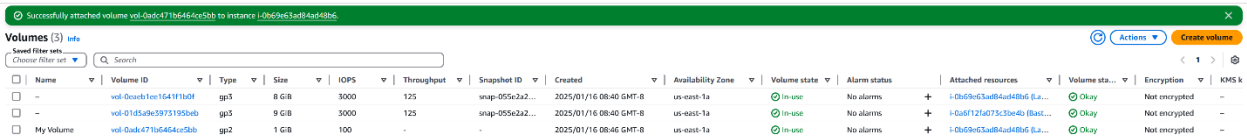
TASK 2:



Select *My Volume*, then drop down the Actions menu and select, Attach volume

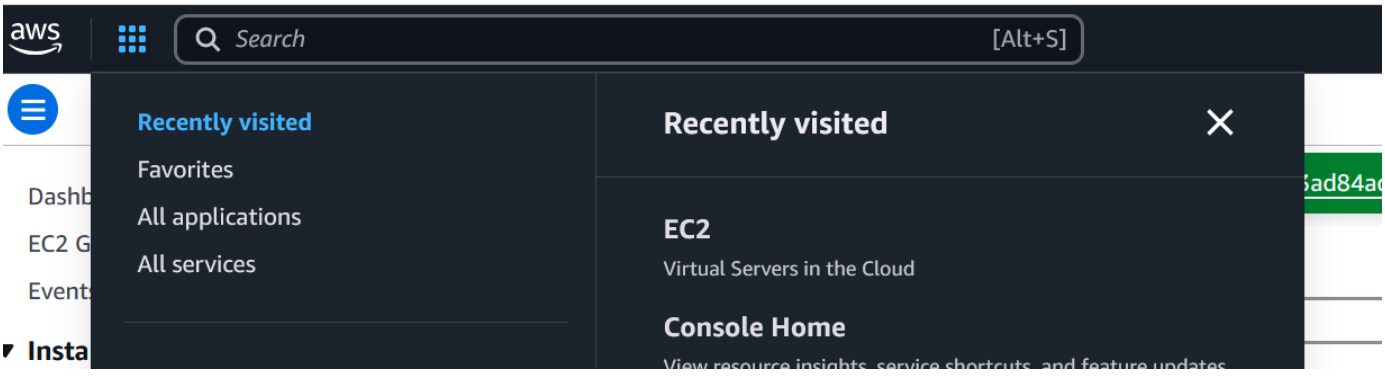


In the instance field select the one labeled, 'Lab', note the device name is set to /dev/sdf, choose to attach volume

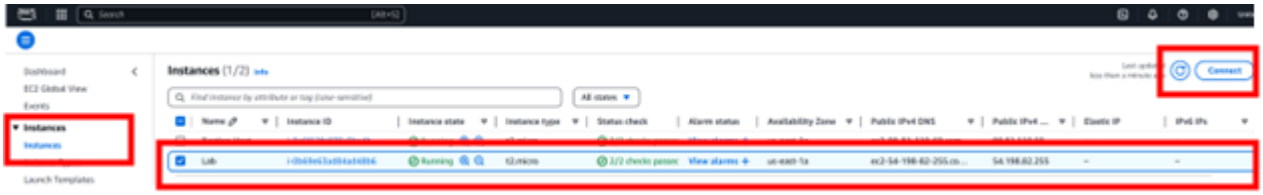


You should now see a 'Successfully attached volume' message

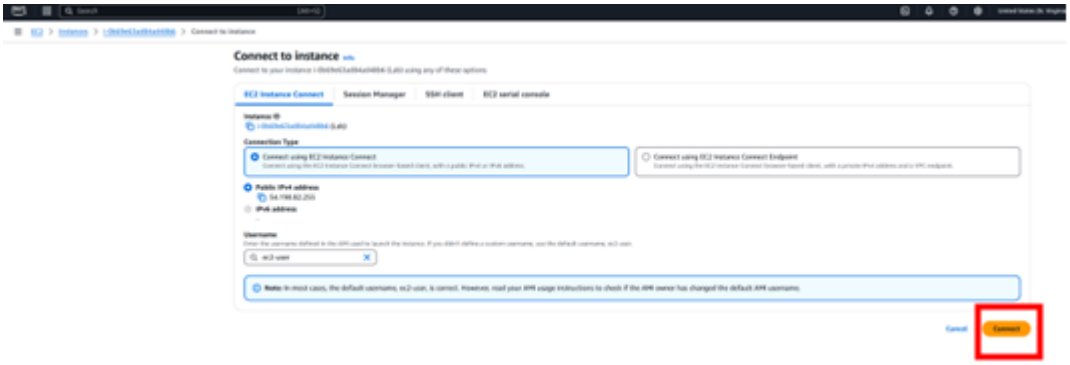
TASK 3:



In the AWS management console search for, and select, EC2



Choose Instances on the left panel and then select your lab instance, after doing this select connect in the top right



On the EC2 Instance Connect tab of the Connect to Instance select ’Connect’

TASK 4:

Enter **df –h** at the $ prompt to view available storage, you should see and output similar to this

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mkfs -t ext3 /dev/sdf
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 5f33ee49-d2c9-421f-ac18-c34a42b2f209
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[ec2-user@ip-10-1-11-4 ~]$
```

Create an ext3 file system on the new volume, enter **sudo mkfs -t ext3 /dev/sdf**, there should be an indication of a created filesystem

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mkdir /mnt/data-store
[ec2-user@ip-10-1-11-4 ~]$
```

Next create a directory for in order mount a new storage volume with the command **sudo mkdir /mnt/data-store**

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mount /dev/sdf /mnt/data-store
[ec2-user@ip-10-1-11-4 ~]$
```

Next mount that directory with the command **sudo mount /dev/sdf /mnt/data-store**

```
[ec2-user@ip-10-1-11-4 ~]$ echo "/dev/sdf   /mnt/data-store e
/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2
[ec2-user@ip-10-1-11-4 ~]$
```

To configure to mount this volume upon the instance starting, you need to add a line to the /etc/fstab, run this command, **echo "/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab**

```
[ec2-user@ip-10-1-11-4 ~]$ cat /etc/fstab
#
UUID=73e034f4-2887-4ec9-8b40-0d35c0091a37     /           xfs    defaults
UUID=9F37-3C35        /boot/efi       vfat    defaults,noatime,uid=0,gid=
/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2
[ec2-user@ip-10-1-11-4 ~]$
```

To find the setting on the last line you need to view the configuration file with the **cat /etc/fstab** command

```
[ec2-user@ip-10-1-11-4 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  452K  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-10-1-11-4 ~]$
```

Use the **df –h** command to once again view your available storage, the output should be similar to this. The output now lists */dev/xvdf* which is the new mounted volume

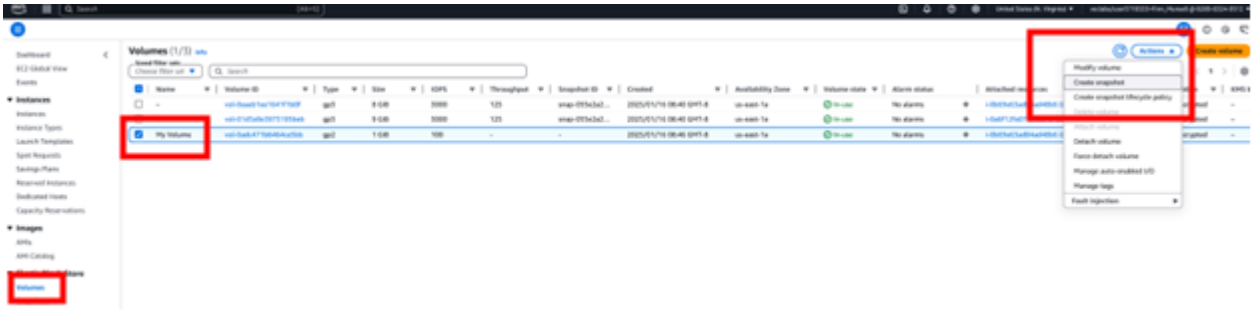
```
[ec2-user@ip-10-1-11-4 ~]$ sudo sh -c "echo some text has been writt
[ec2-user@ip-10-1-11-4 ~]$
```

On the mounted volume you viewed above create a file and add some text using **sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"**

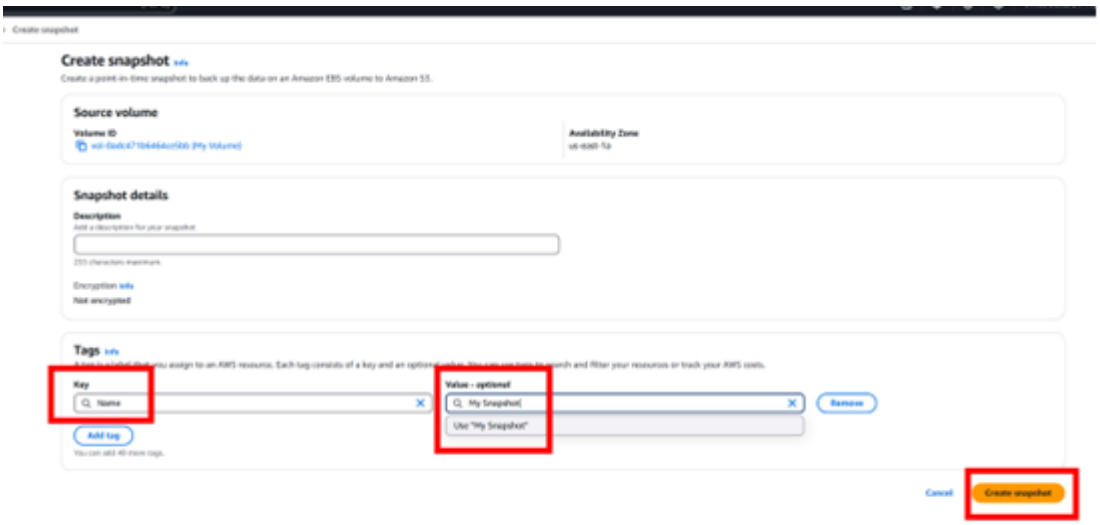
```
[ec2-user@ip-10-1-11-4 ~]$ cat /mnt/data-store/file.txt
some text has been written
```

Verify that text was written with **cat /mnt/data-store/file.txt** ensure your EC2 instance is still running and leave it that way

TASK 5:



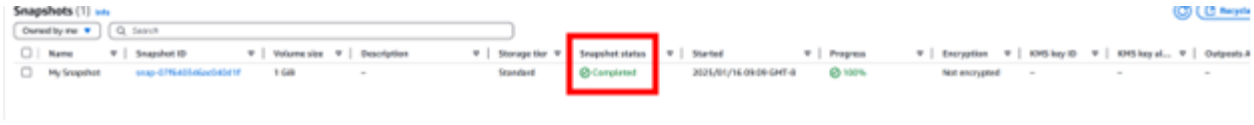
Return to the EC2 console, choose volumes on the left-hand panel, and select *My Volume* by checking the box next to it. Then open the actions menu and, ”*Create Snapshot*”



Choose to, ”*add tag*” near the bottom. For Key, enter *Name* and for Value enter *My Snapshot*. Finally, choose to *Create snapshot*

You should recive a message showing the successful creation of this snapshot

In the left-hand panel choose snapshots it should still be in a state of *Pending*



It will eventually turn to *Completed*

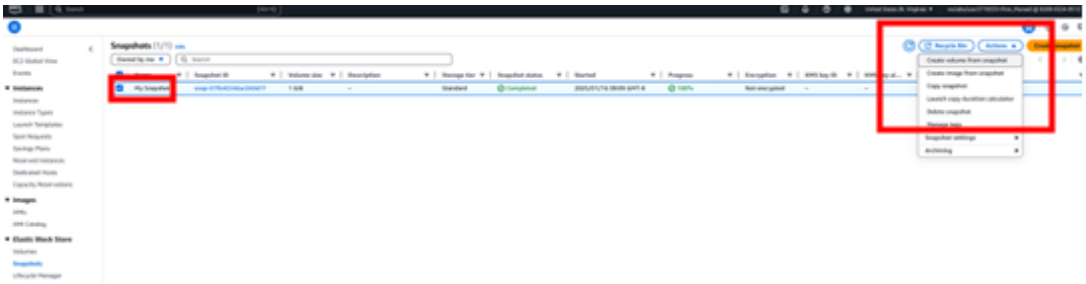
```
[ec2-user@ip-10-1-11-4 ~]$ sudo rm /mnt/data-store/file.txt
[ec2-user@ip-10-1-11-4 ~]$
```

In your EC2 Instance Connect session, delete the file that you created on your volume with **sudo rm /mnt/data-store/file.txt**

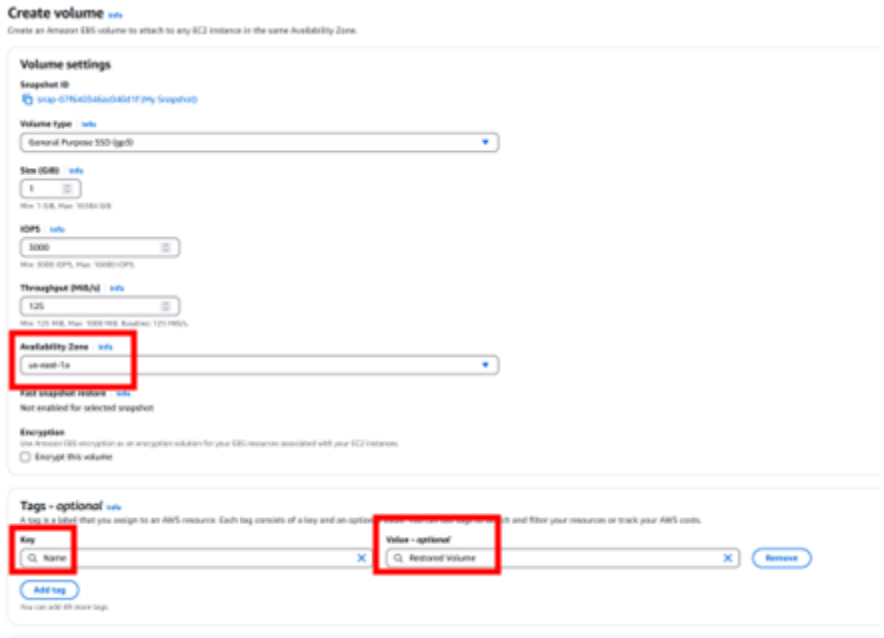
```
[ec2-user@ip-10-1-11-4 ~]$ ls /mnt/data-store/
lost+found
```

Verify said files deletion with **ls /mnt/data-store/**

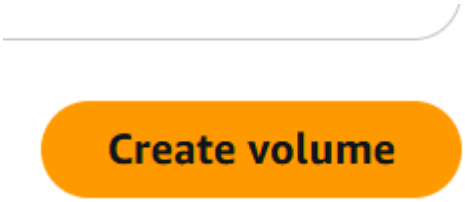
TASK 6:



Back in the EC2 console reselect *My Snapshot*, then select the actions menu and *Create volume from snapshot*



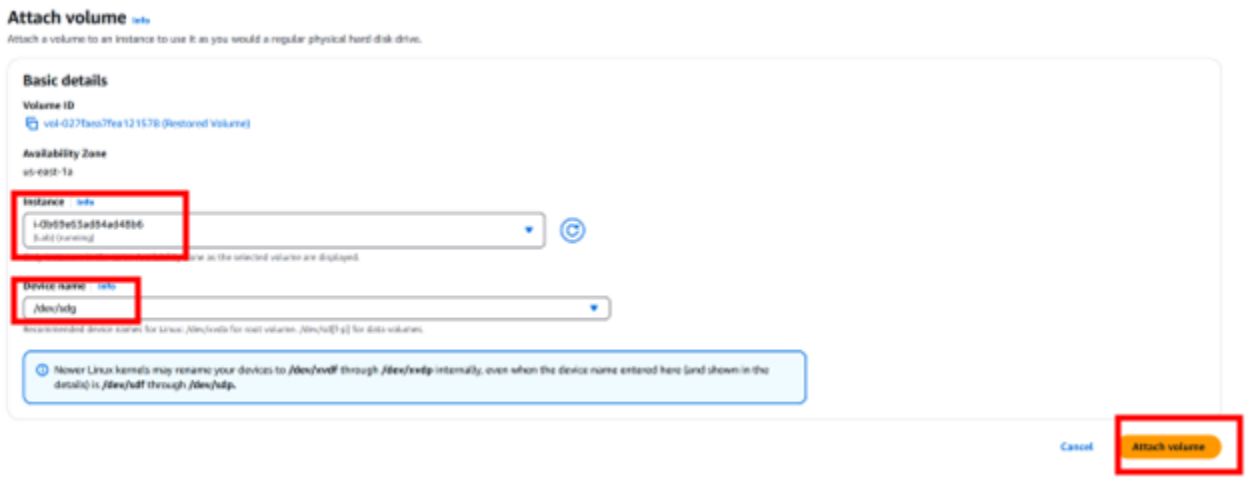
For availability zone, use the same one as earlier, then add a tag, use *Name* for Key and for value use *Restored Volume*



Finally, choose to create volume

ATTACH RESTORED VOLUME TO EC2 INSTANCE:

In the left-hand panel choose volumes, select your *Restored Volume*, in the actions menu ”*Attach volume*”



Select the Lab option in instance, note that Device should be set to */dev/sdg* this will be used later, then attach the volume

You should receive a successfully attached message, the volume state is now, *in-use*

MOUNT RESTORED VOLUME:

```
[ec2-user@ip-10-1-11-4 ~]$ sudo mkdir /mnt/data-store2
[ec2-user@ip-10-1-11-4 ~]$
```

Now return to the other EC2 window and create a directory for mounting the new storage volume with **sudo mkdir /mnt/data-store2**

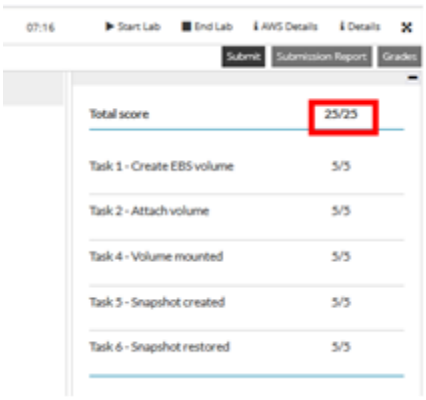
```
[ec2-user@ip-10-1-11-4 ~]$ sudo mount /dev/sdg /mnt/data-store2
[ec2-user@ip-10-1-11-4 ~]$
```

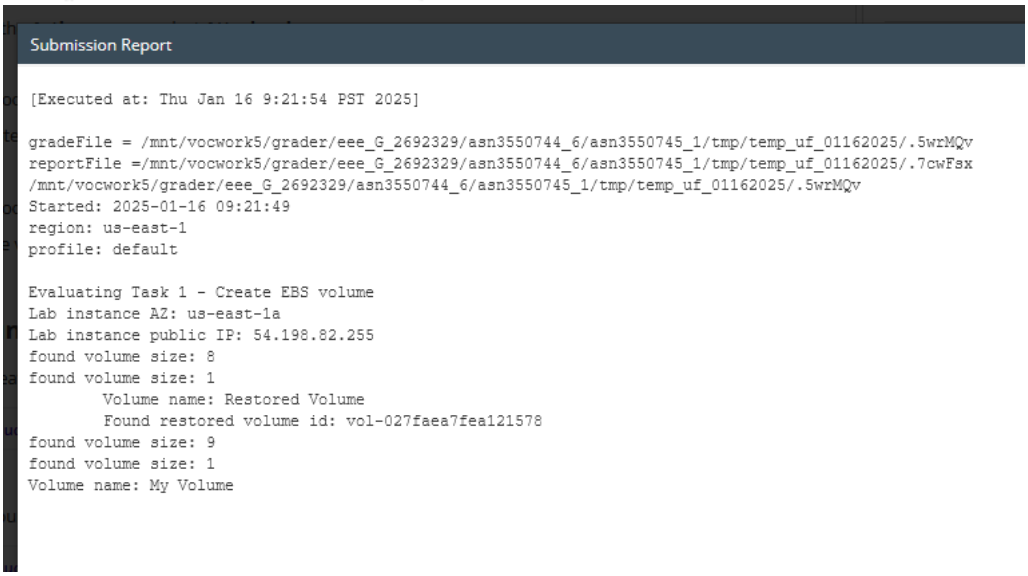
Mount that volume, **sudo mount /dev/sdg /mnt/data-store2**

```
[ec2-user@ip-10-1-11-4 ~]$ ls /mnt/data-store2/
file.txt  lost+found
[ec2-user@ip-10-1-11-4 ~]$
```

Verify that volume you mounted has the file that you created earlier with *ls /mnt/data-store2/*, you should see file.txt

SUBMIT:

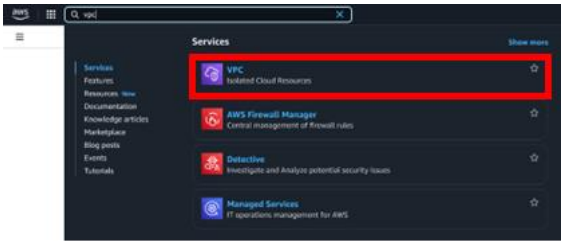




Wait approximately 5 minutes to submit, you can then view an accurate total score

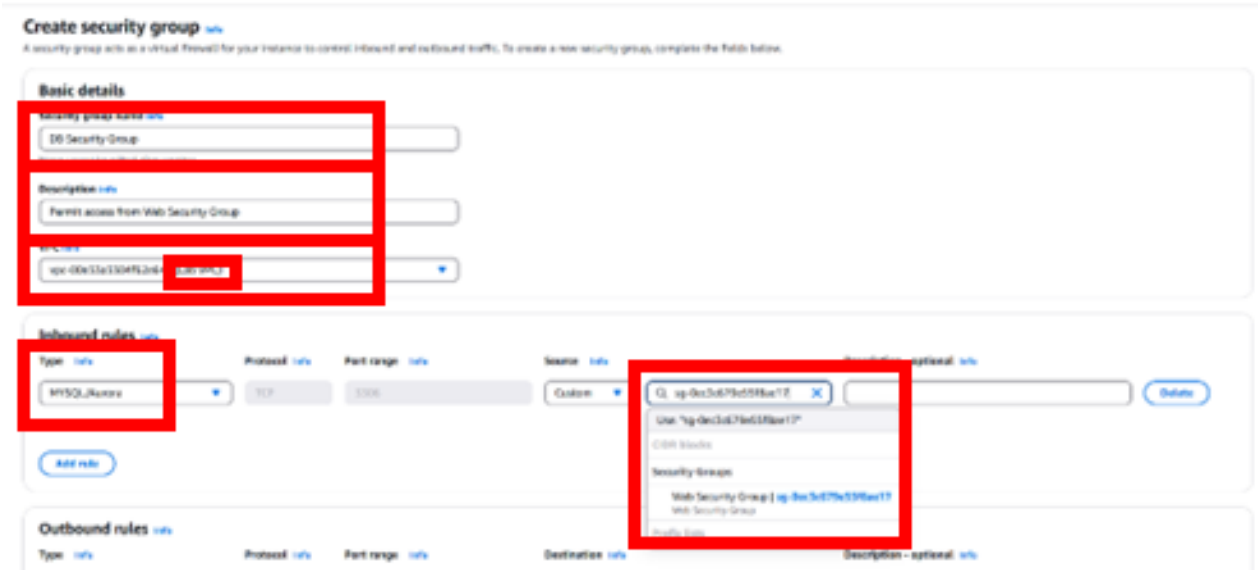
LAB 5:

TASK 1:

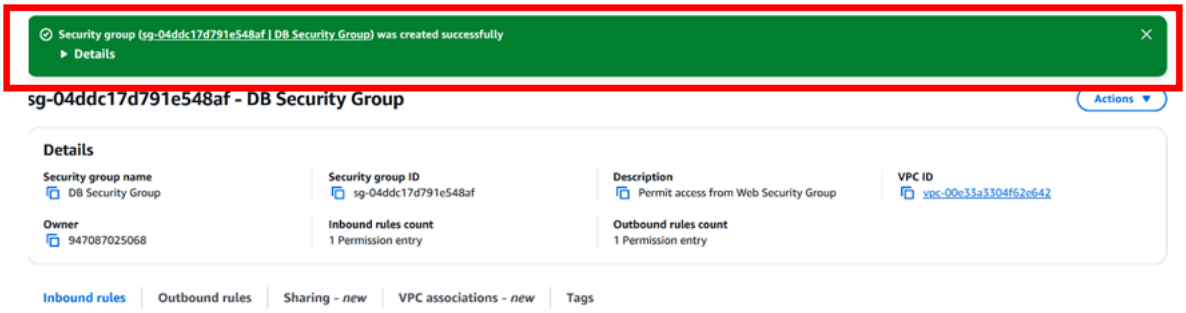


Open AWS management console and use the upper left search tool to find VPC, select it

In the left navigation panel choose Security groups, choose ‘Create security group’

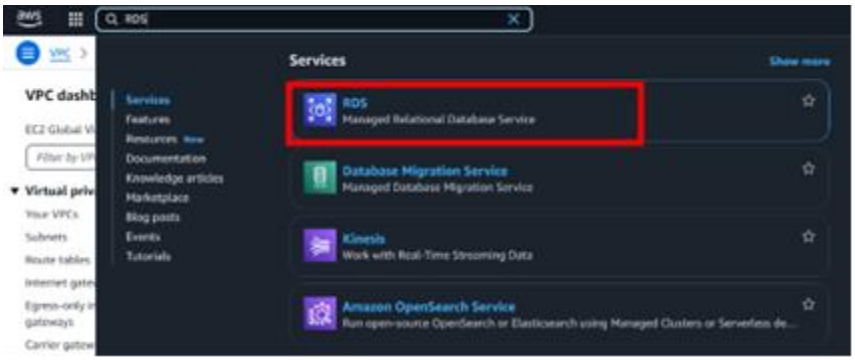


For Security group name use *DB Security Group*, for Description set it to *Permit access from Web Security Group*, for VPC use ‘Lab VPC’, choose the X next to the preselected VPC and choose Lab VPC from the menu. In Inbound rules choose “Add rule”, *MySQL/Aurora (3306)* should be the Type and then for source, place the cursor in the field to the right of Custom, type *sg*, and then select ‘Web Security Group’. Finally, create the security group



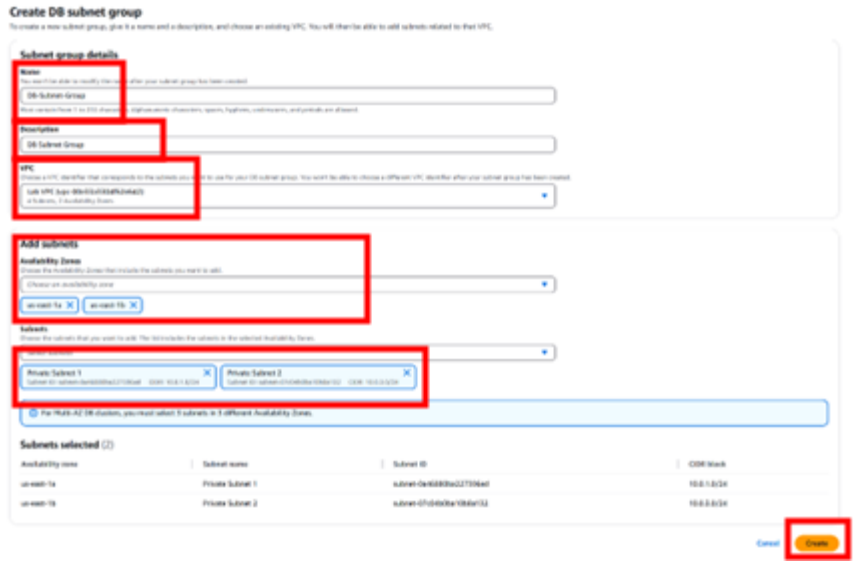
You should see a confirmation notification

TASK 2:



In the AWS management use the search box next to services to select RDS

In the left-hand navigation panel choose Subnet groups, choose ’Create DB subnet group’

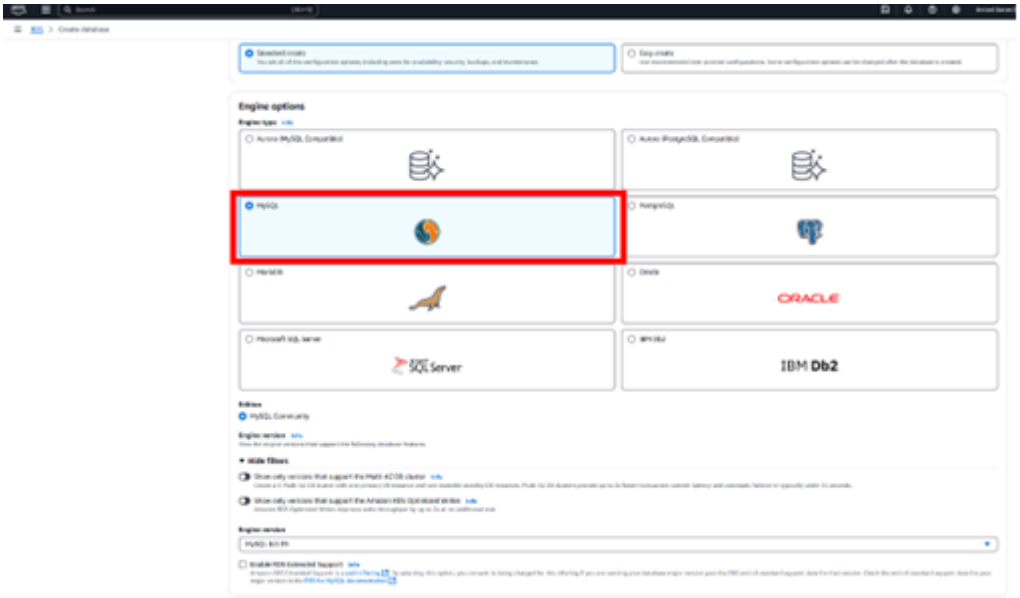


Configure Name as *DB-Subnet-Group*, the Description as *DB Subnet Group*, and VPC as ‘Lab VPC’. Then Go to the Add subnets section and select the list of values under Availability Zones and choose the first two zones, that should be *us-east-1a* and *us-east-1b*. Then expand the list of values under Subnets and select the subnet ranges *10.0.1.0/24* and *10.0.3.0/24*. These subnets should now be shown in the Subnets selected table. Finally, choose ‘Create’

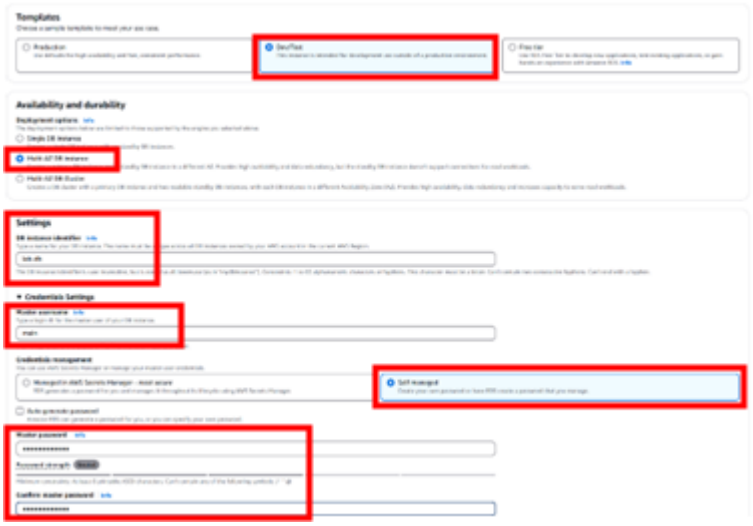
You should receive a successfully created message

TASK 3:

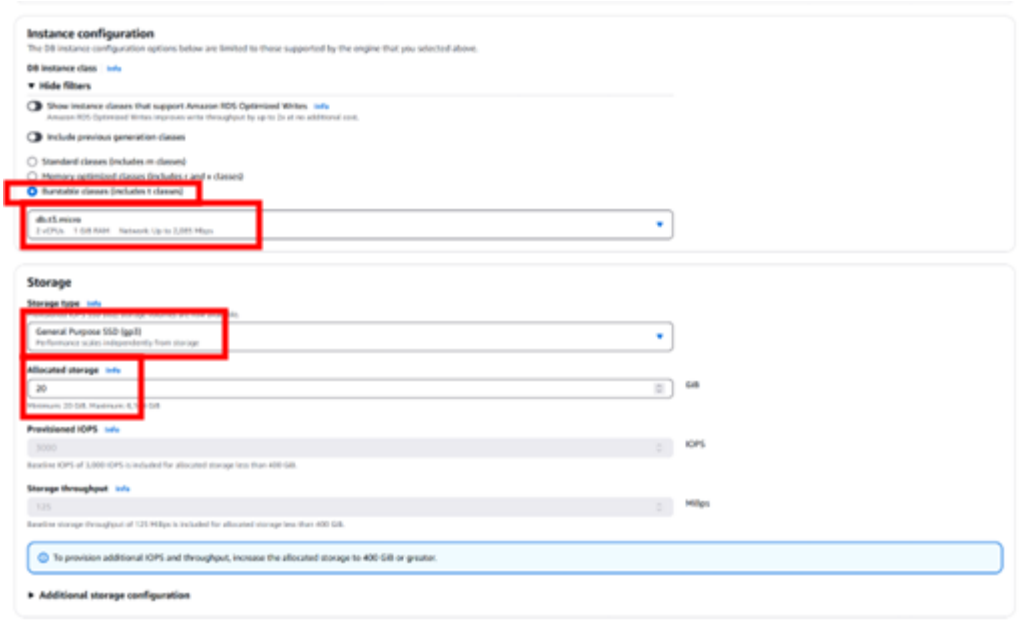
In the left-hand navigation panel, choose Databases and then choose ’Create database’



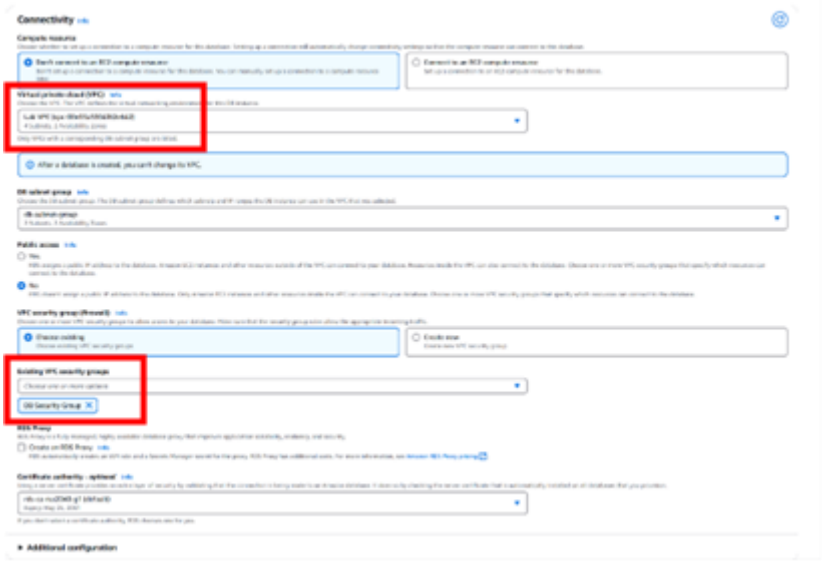
Choose ‘My SQL’ for the Engine Options



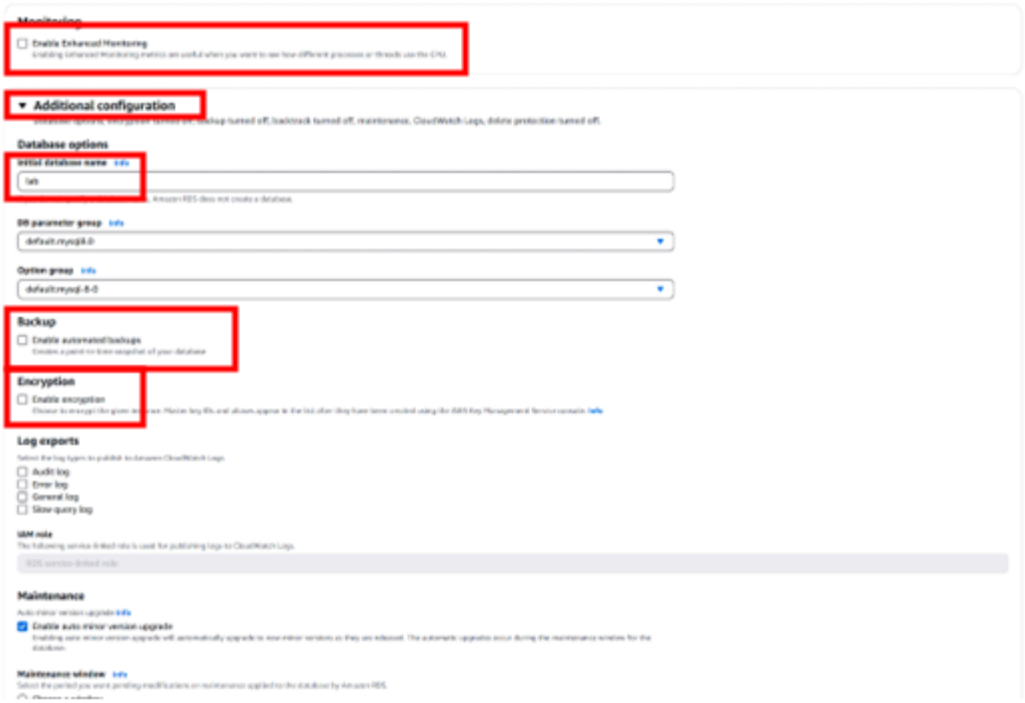
Under templates choose the option of ‘Dev/Test’, for availability and durability use the choice ‘Multi-AZ DB’ distance. For settings DB instance identifier: ‘lab-db’, Master username: *main*, Master password: *lab-password*, confirm password is the same as master password



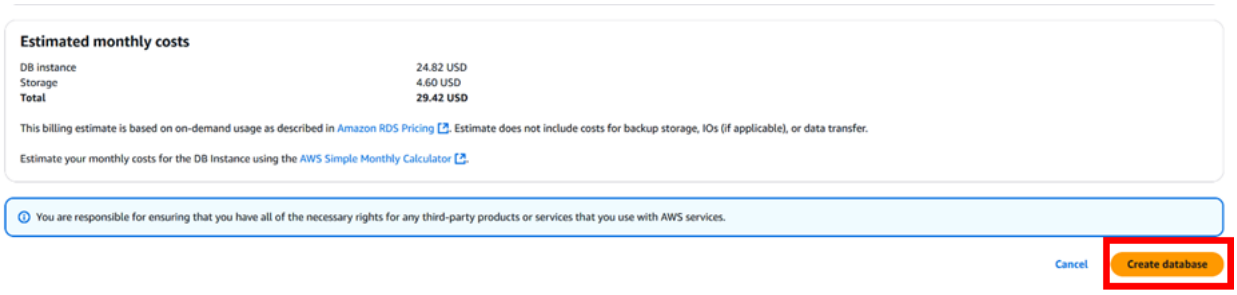
Under DB instances turn on ‘Burstable classes’ and then select ‘db.t3.micro’, for storage use ‘General purpose (SSD)’ as the storage type and for allocated storage use *20*



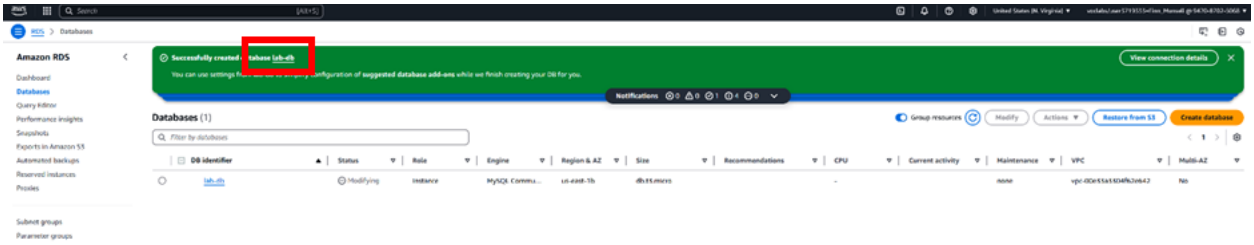
Under Connectivity, configure Virtual Private Cloud (VPC) as ‘Lab VPC’, Under Existing VPC security groups, from the dropdown list Choose ‘DB Security Group’ and be sure to deselect default



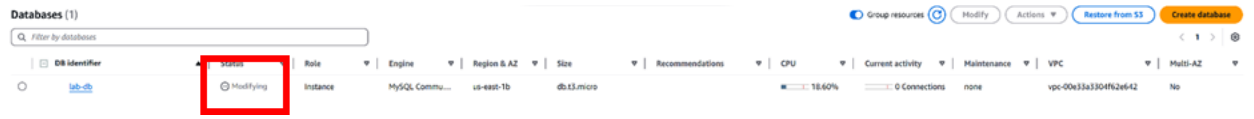
Under Monitoring expand ‘Additional configuration’. Uncheck ‘Enable Enhanced monitoring’. Under Additional configuration, configure the Initial database name as *lab*. Uncheck ‘Enable automatic backups’. ‘Uncheck Enable encryption’. This will make the database faster for this lab but usually it is not recommended



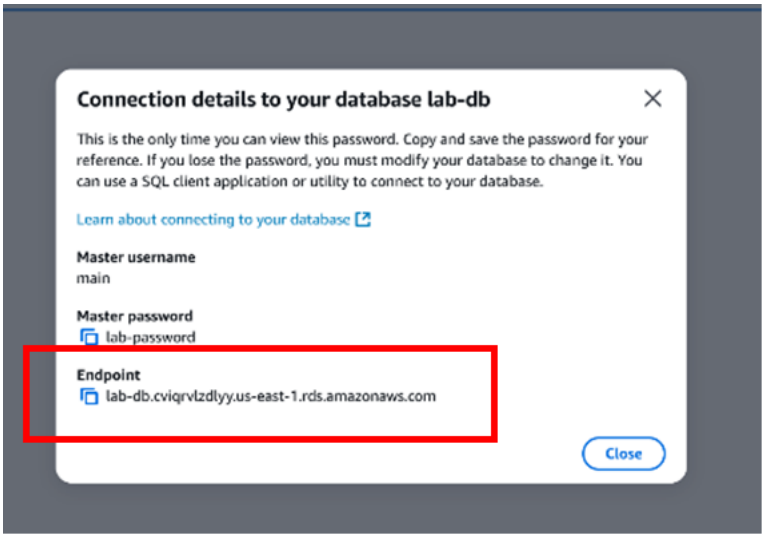
Choose ’Create database‘, your database will now be launched



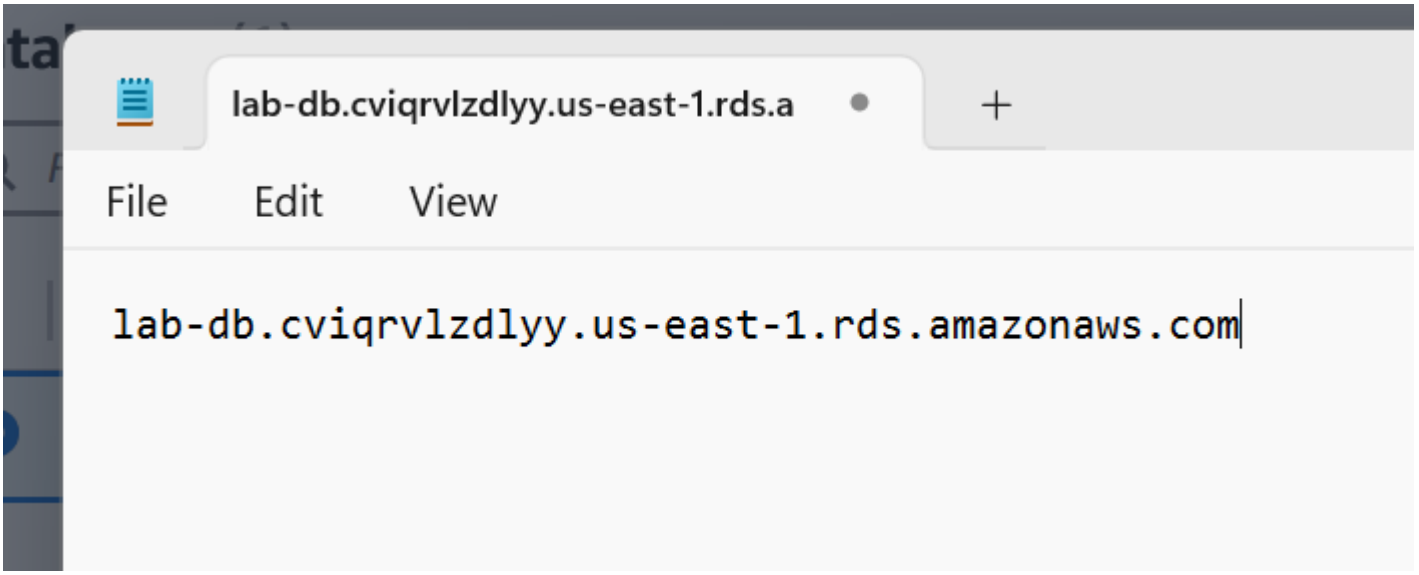
Choose ’lab-db‘ (choose the link itself)



You will now need to wait approximately 4 minutes for the database to be available. The deployment process is deploying a database in two different Availability zones. Wait until Info changes to ‘Modifying’ or ‘Available’.



Scroll down to the Connectivity & security section and copy the Endpoint field, it will look similar to: *lab-db.xxxx.us-east-1.rds.amazonaws.com*



Paste the Endpoint value into a text editor. You will use it later in the lab

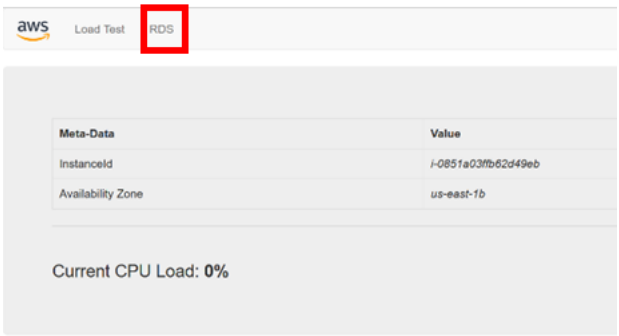
TASK 4:

Unable to load the graphic

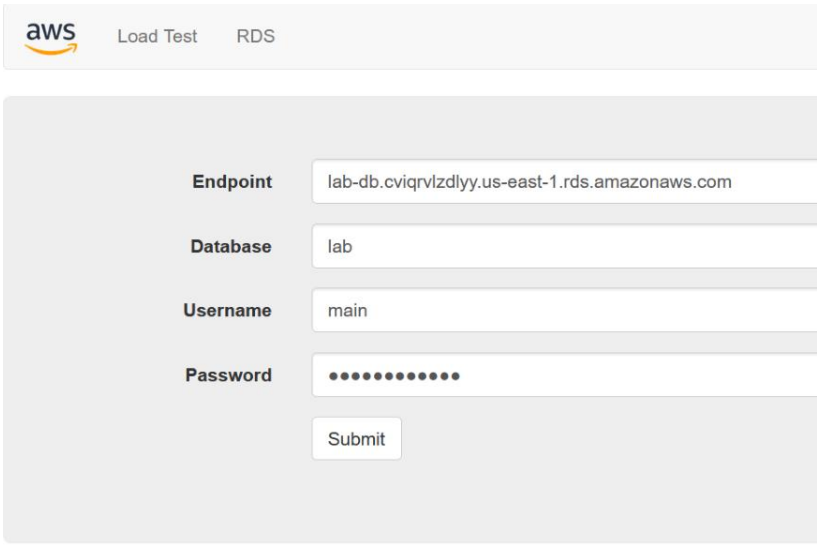
To discover the ‘WebServer’ IP address, choose on the AWS Details drop down menu above the instructions and copy the IP address value

Unable to load the graphic

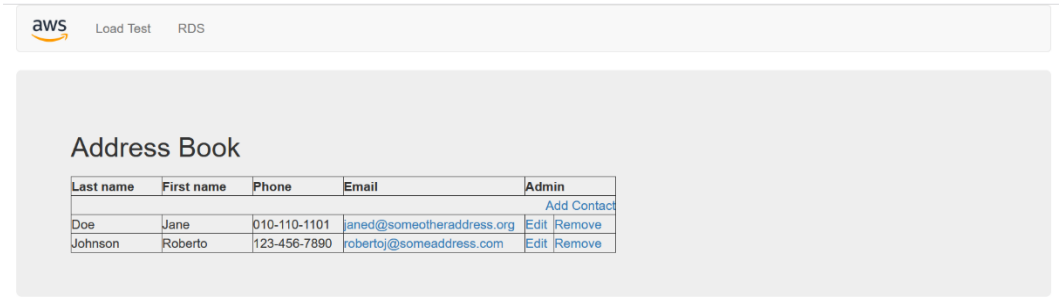
Open a new web browser tab, paste the WebServer IP address and press enter. Now the web application will be displayed, showing information about the EC2 instance



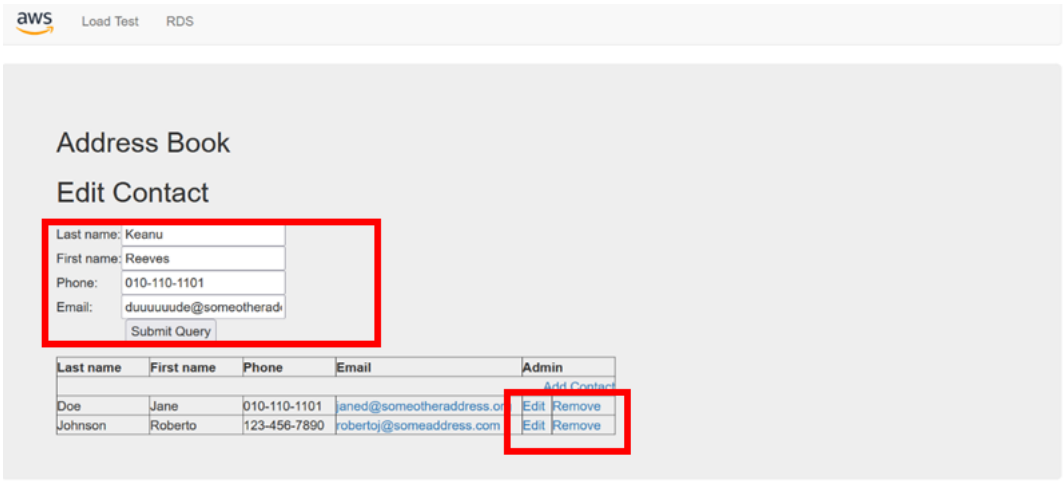
Choose the ‘RDS’ link at the top of the page

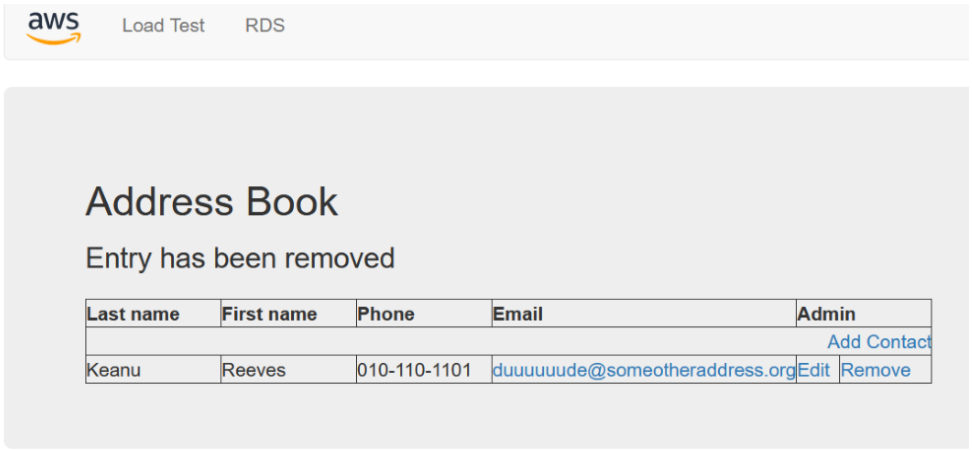


Configure with the following settings: for Endpoint paste the Endpoint you copied to a text editor earlier, for the Database use *lab*, for Username use *main*, and for the Password use *lab-password*, finally choose ’Submit’



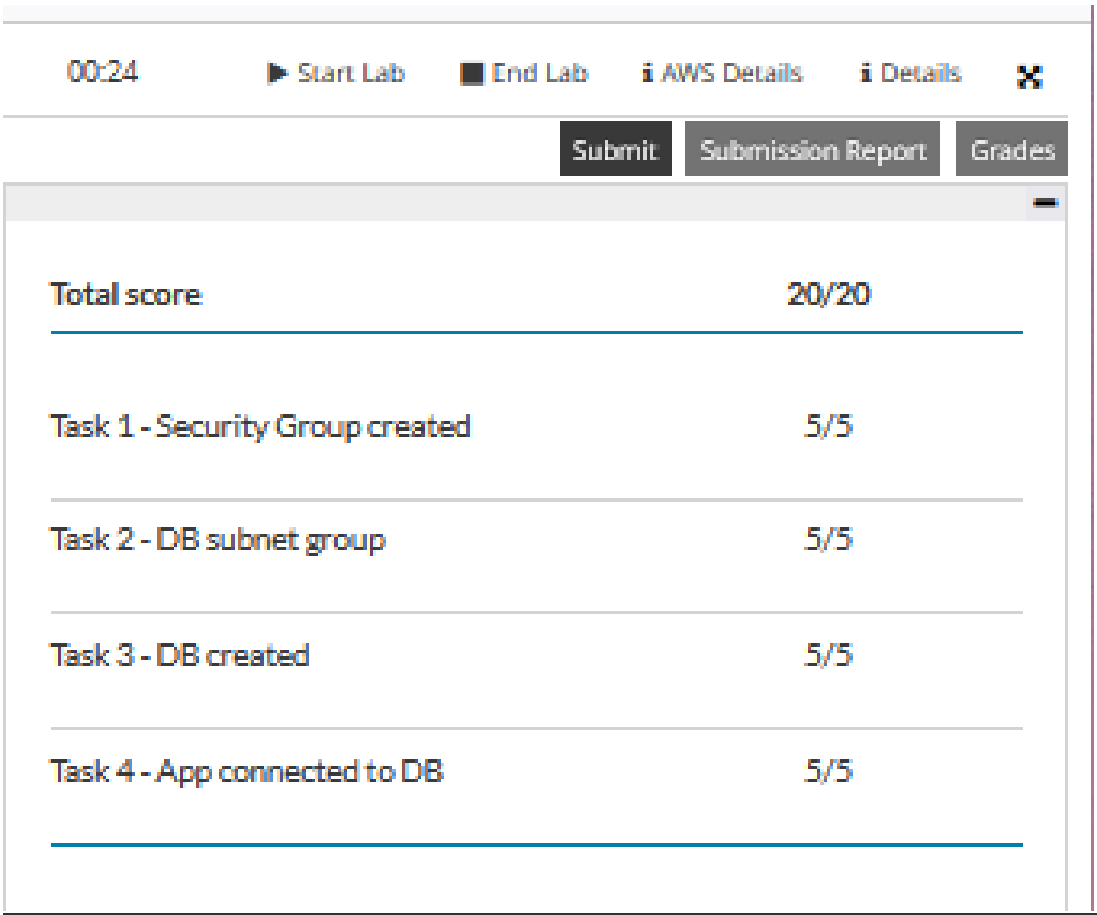
A message will appear explaining that the application is running a command to copy information to the database. After a few seconds the application will display an Address Book.

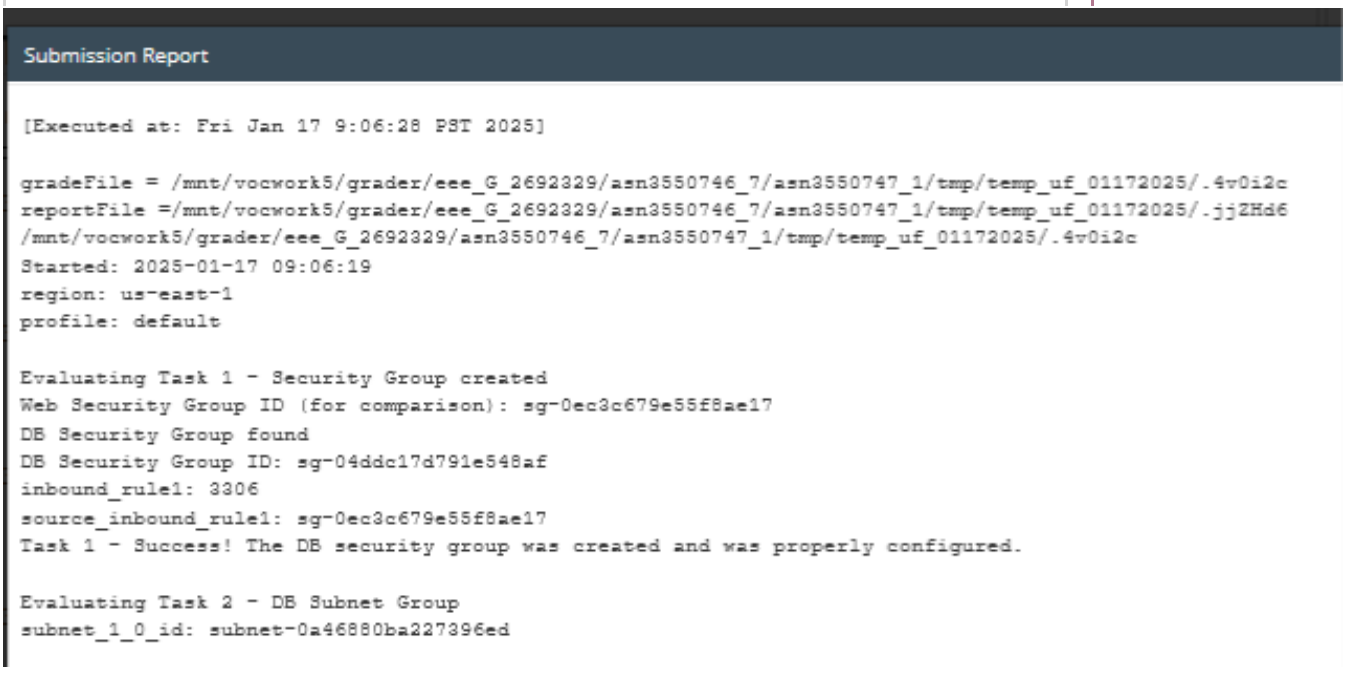




Test the web application by adding, editing and removing contacts, the data is being persisted to the database and is automatically replicating to the second Availability Zone

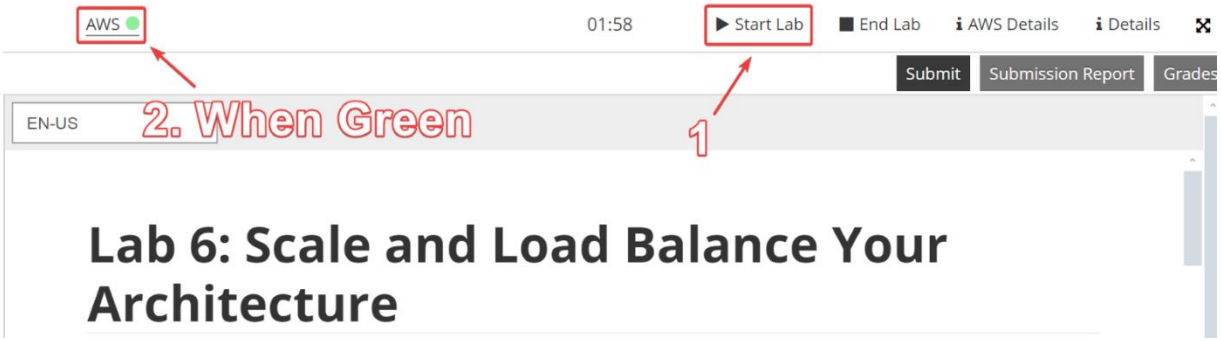
SUBMIT



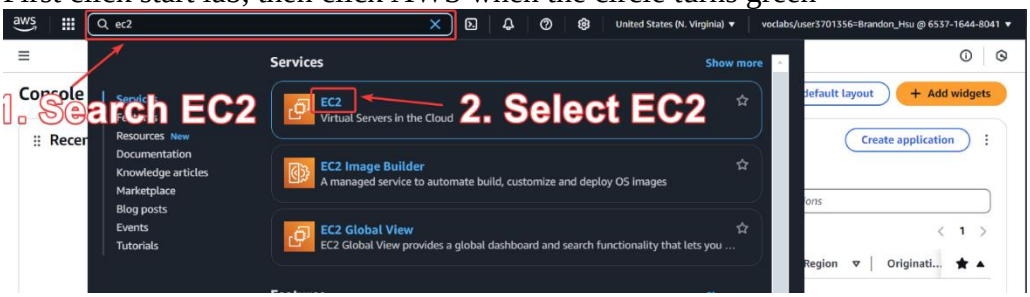


Wait approximately 5 minutes to submit, you can then view an accurate total score

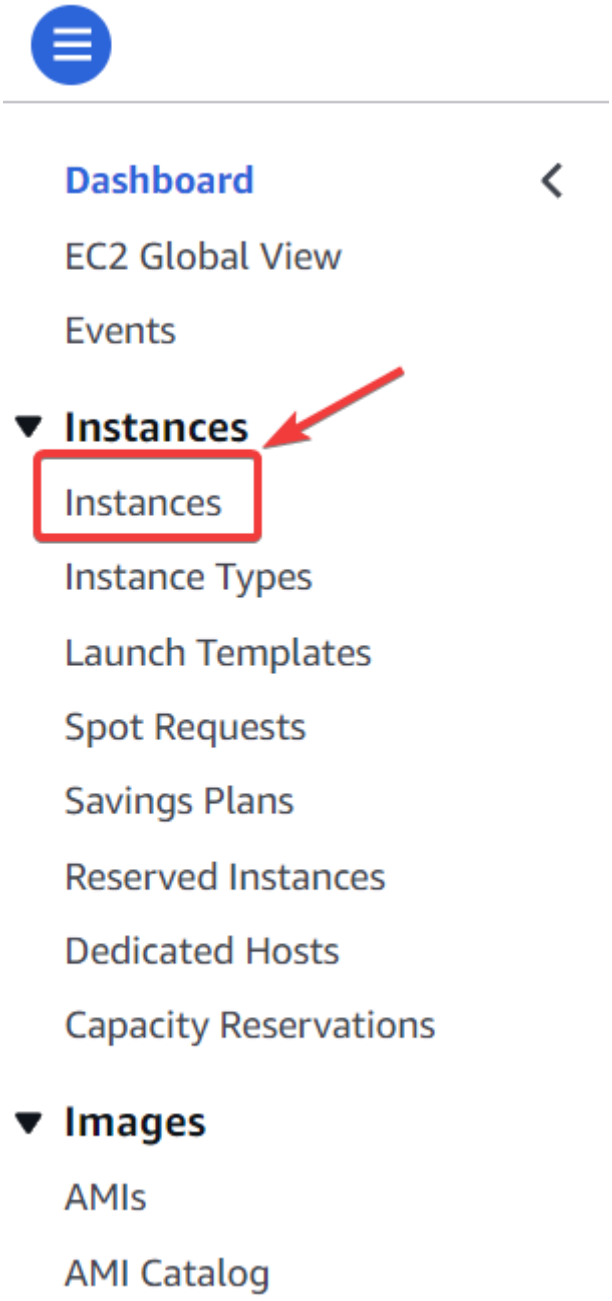
LAB 6:



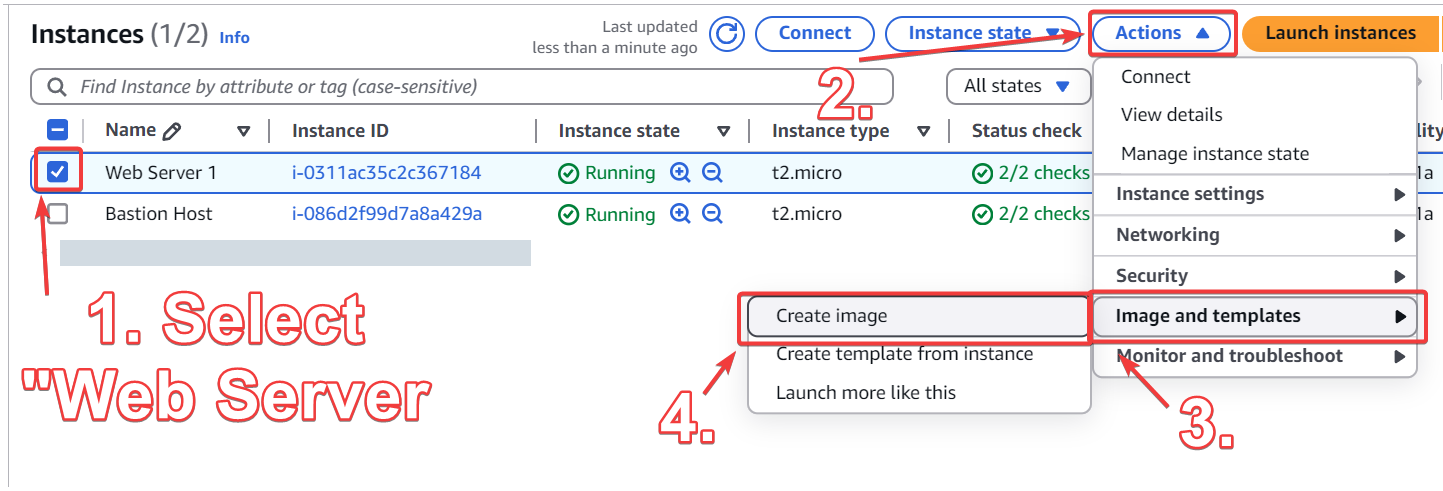
First click start lab, then click AWS when the circle turns green



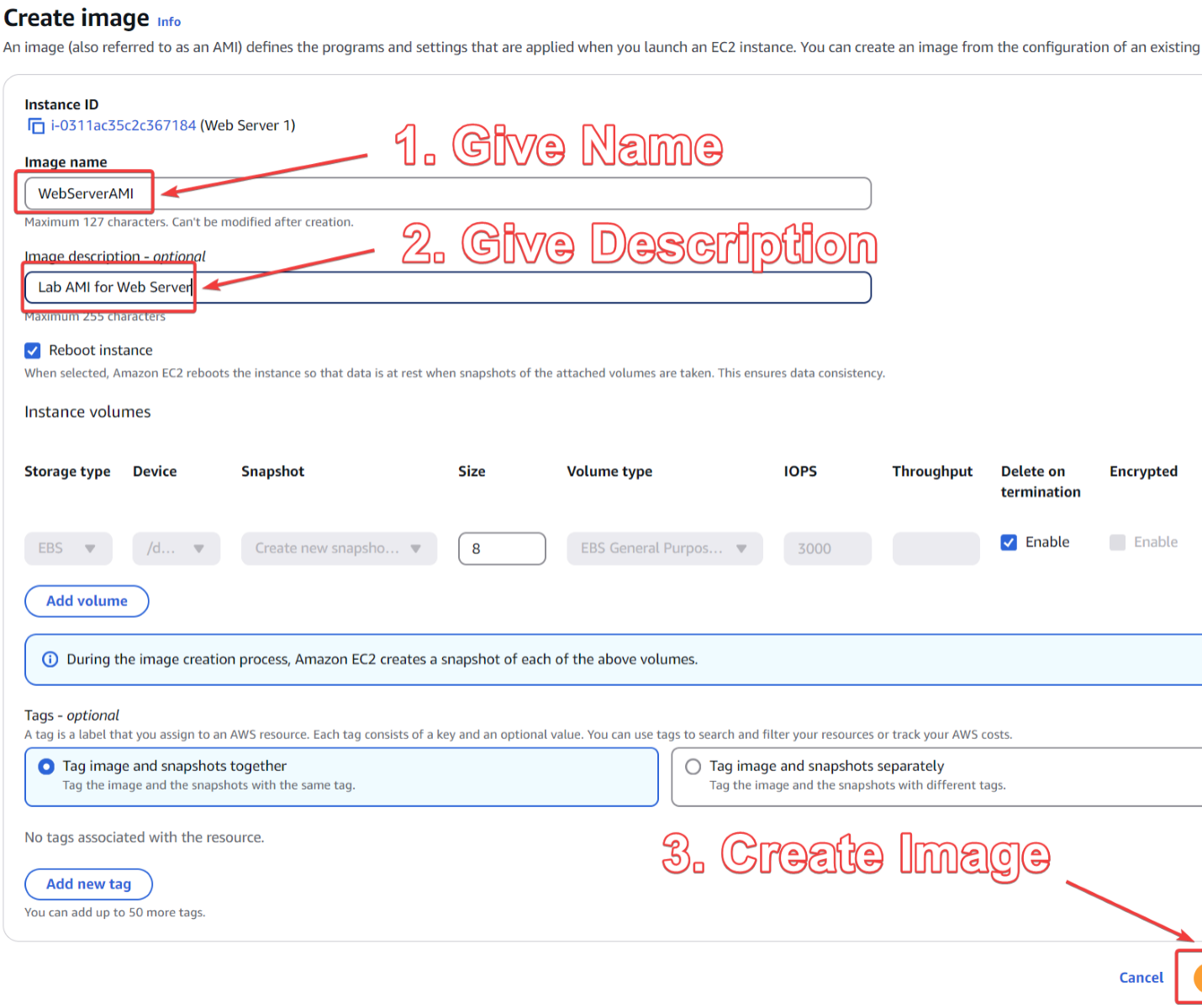
Search then go to the EC2 dashboard



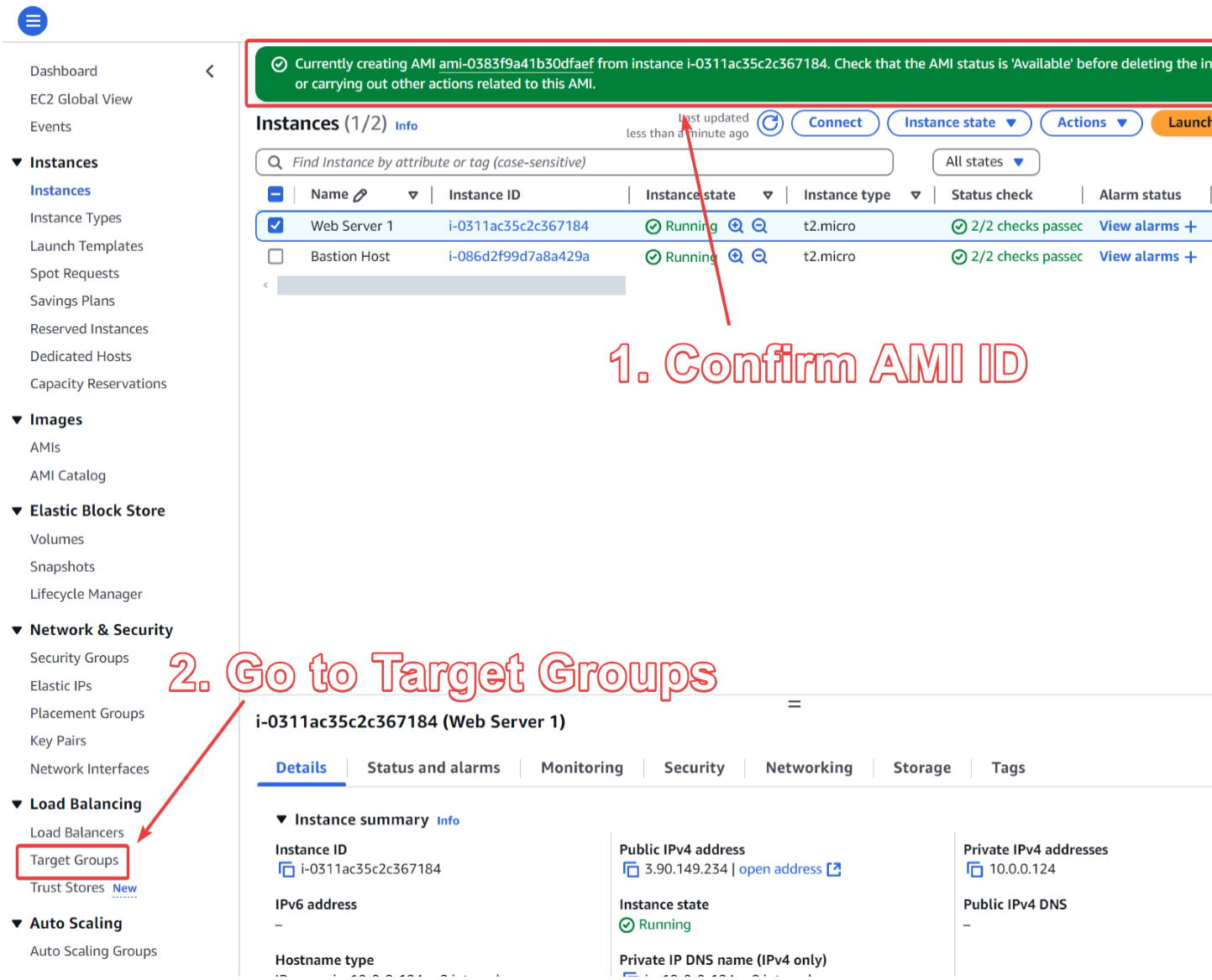
Go to the instances page from the left hand navagational panel



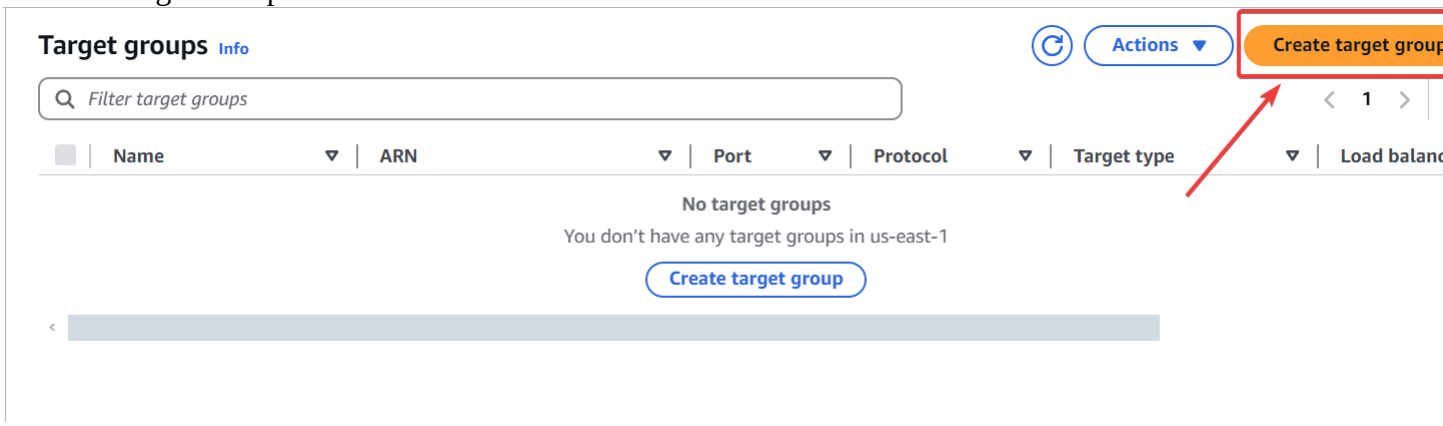
Select the box next to the pre-made Web Server 1, then go to actions followed by selecting Images and templates, then click Create image



Give the Image a name, then give a description, and then click Create Image



See that the AMI ID is correct, and the image is made correctly, then go to target groups from the left navigational panel

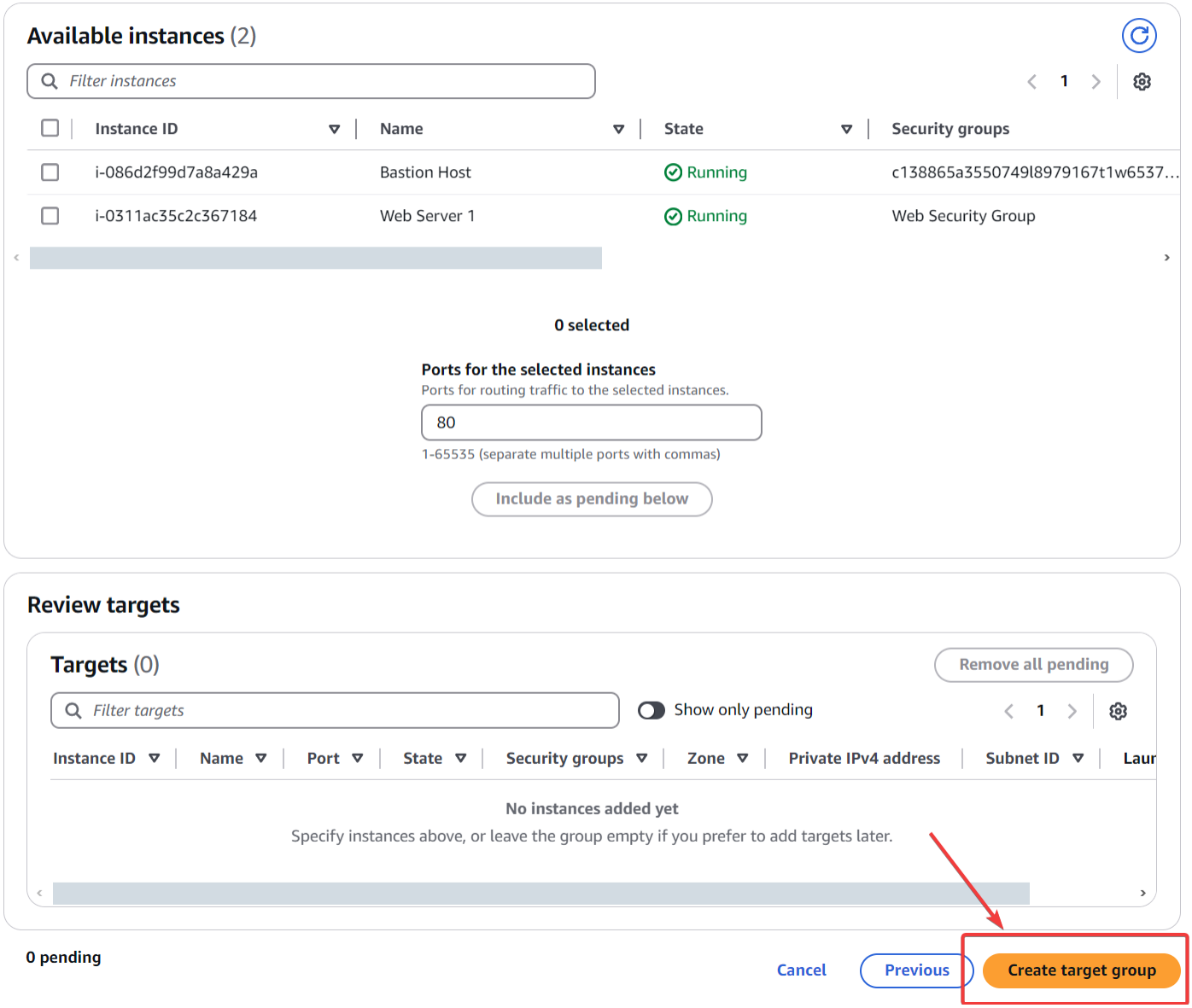


Click Create target group

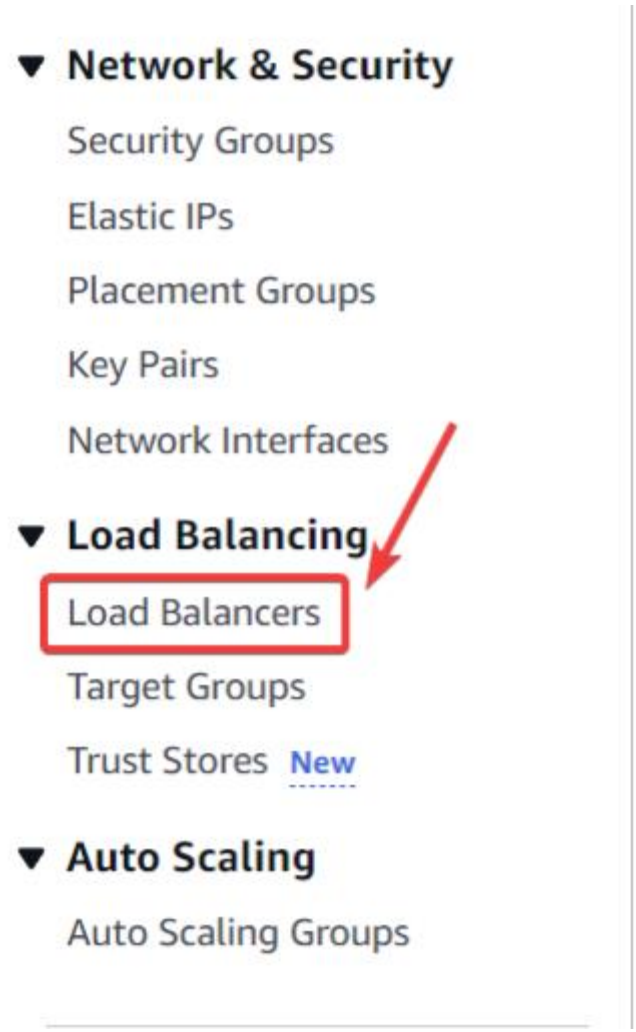
Select Instances, the create a name, select the Lab VPC, then scroll all the way down and click next

Register targets

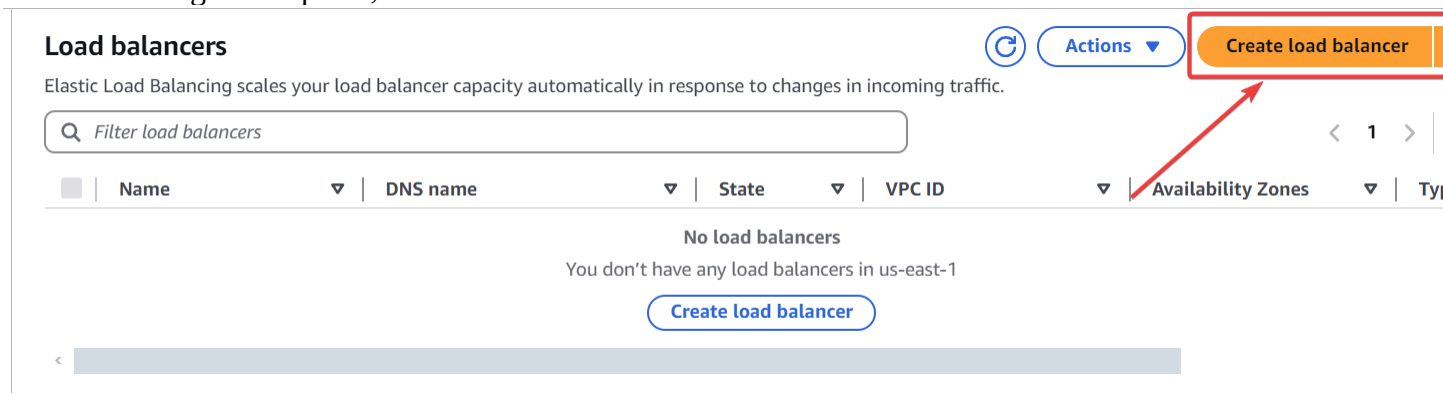
This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.



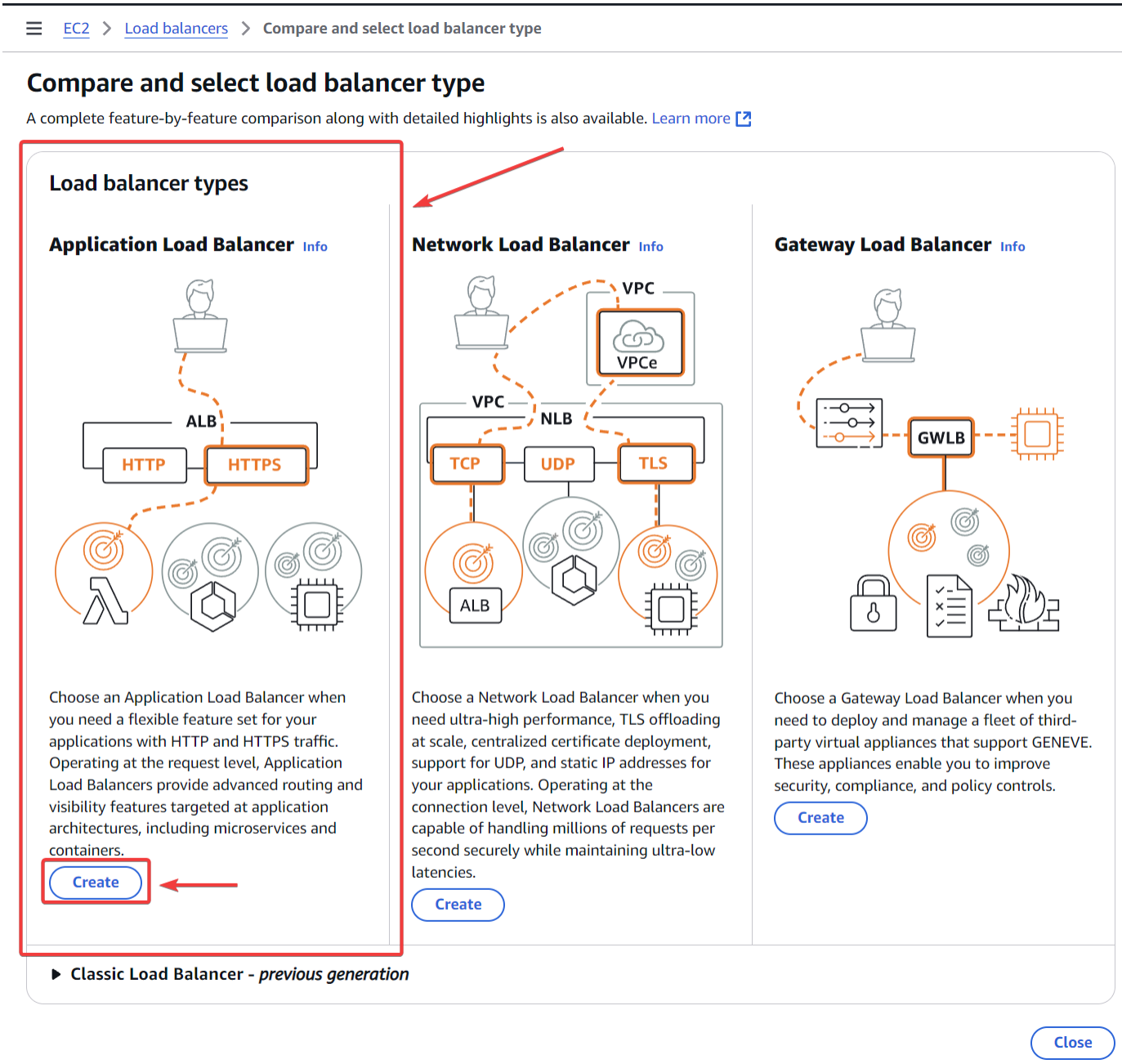
Confirm the information is correct, then select create target group



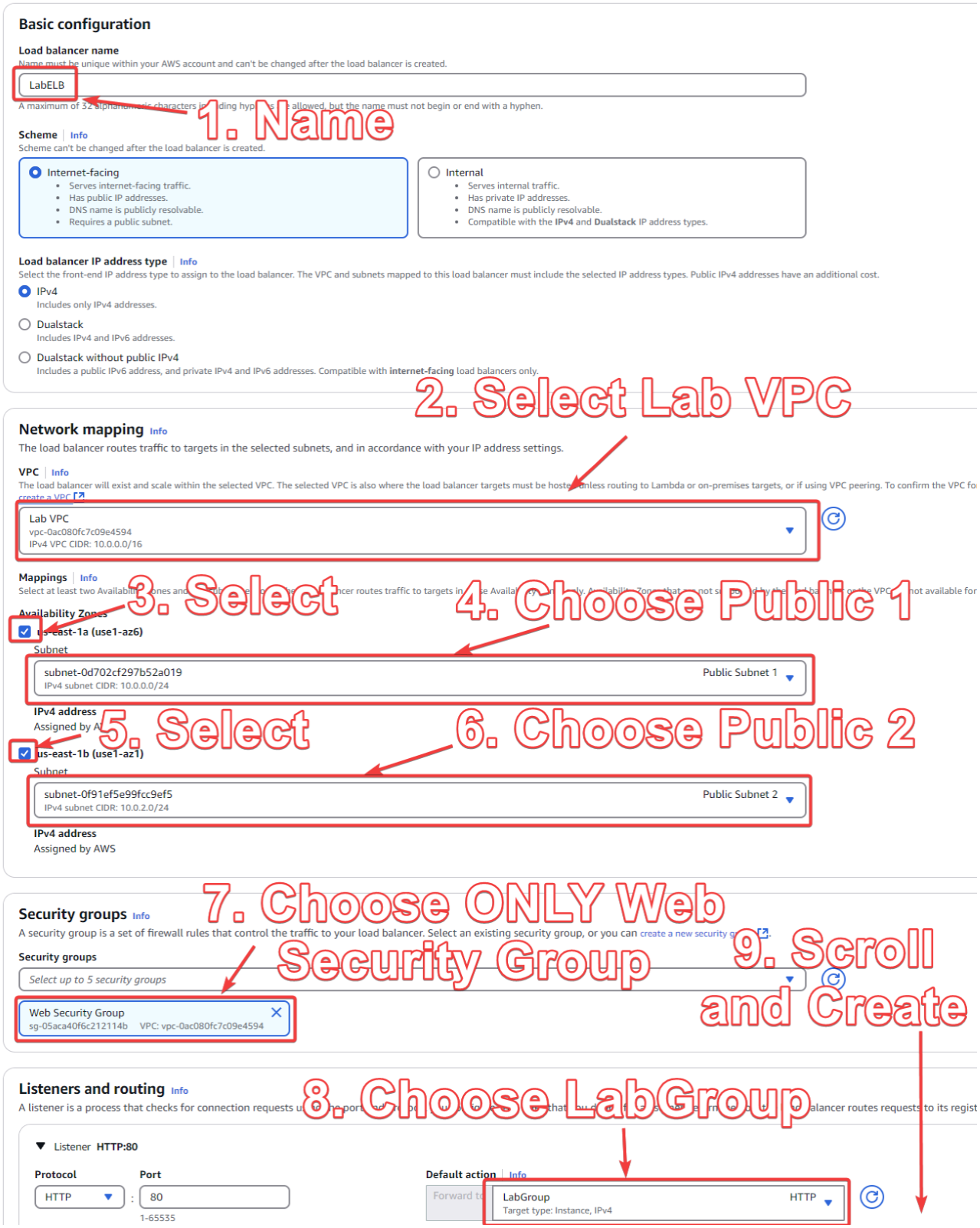
In the left navigational panel, select Load Balencers



Select Create load balancer

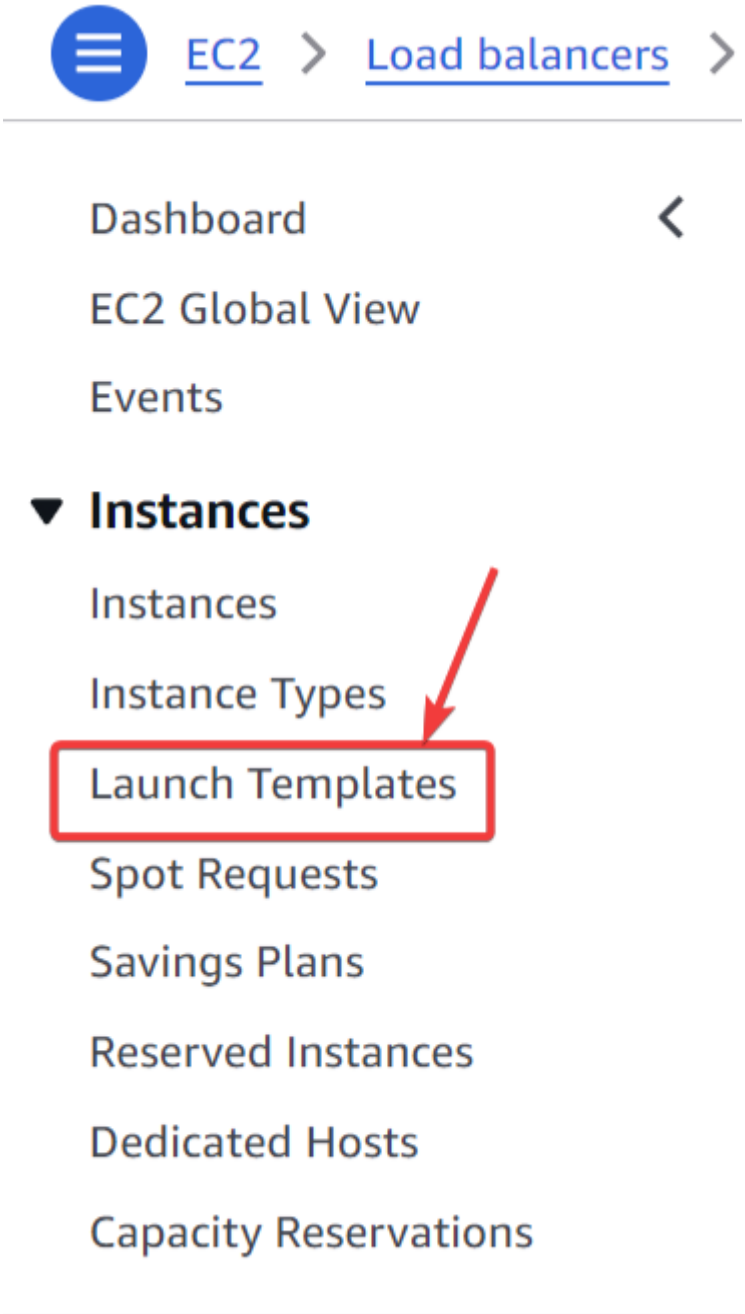


Select the Application Load Balancer and click create

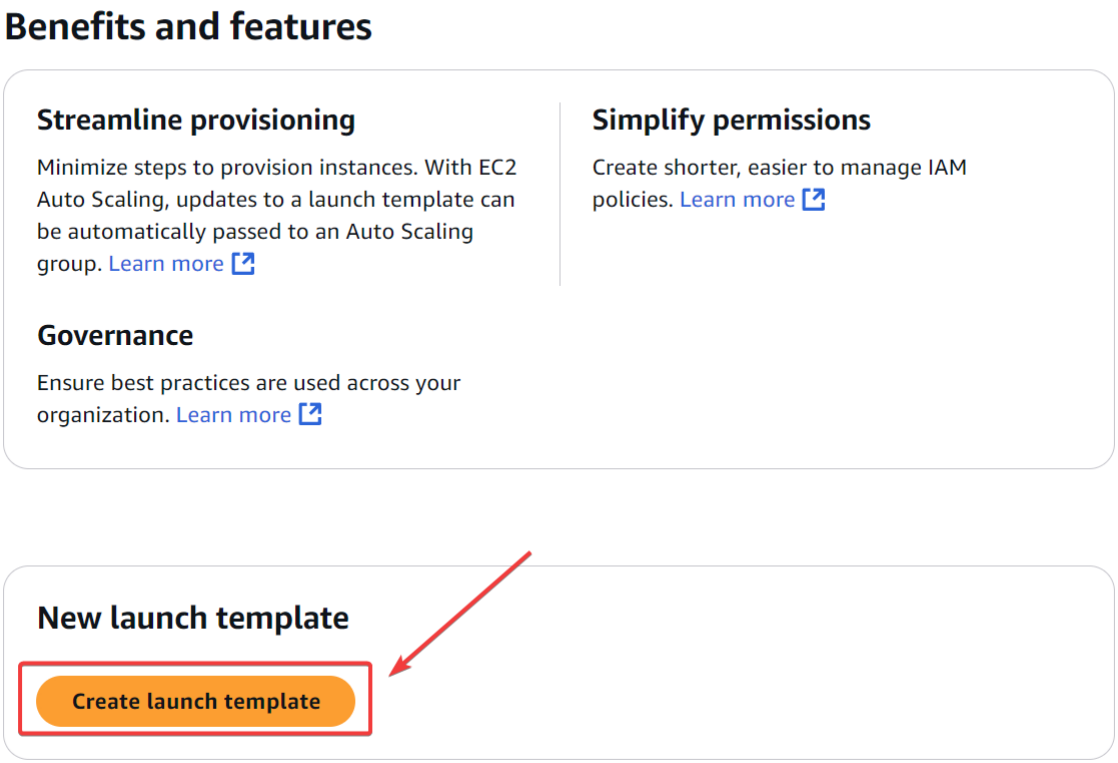


Create a Name, then select Lab VPC for the VPC, then select the first availability zone and choose public 1, select the second availability zone followed by choosing public 2, change the

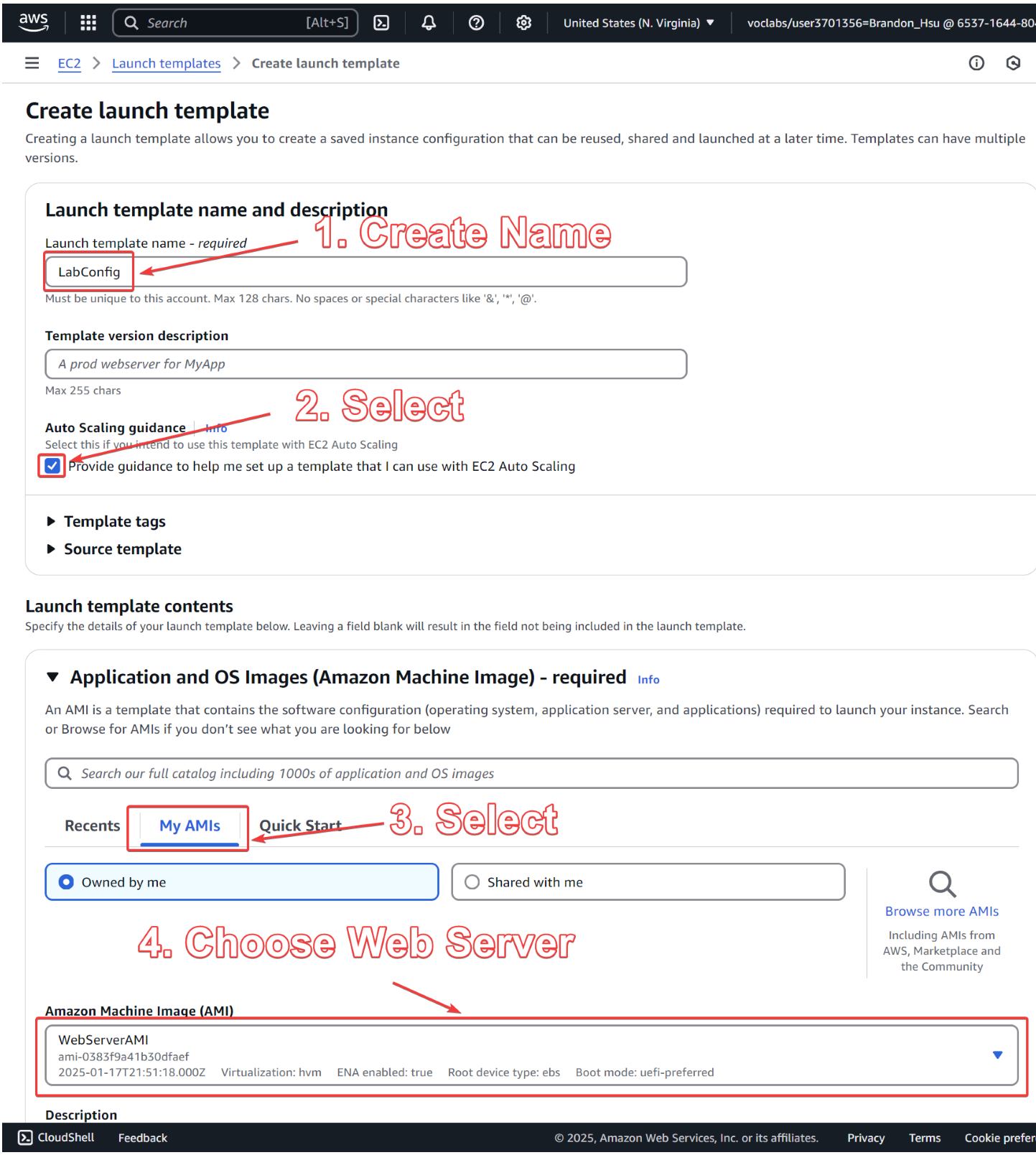
security group to only be Web Security Group, in listeners and routing, select LabGroup, then scroll down and select create



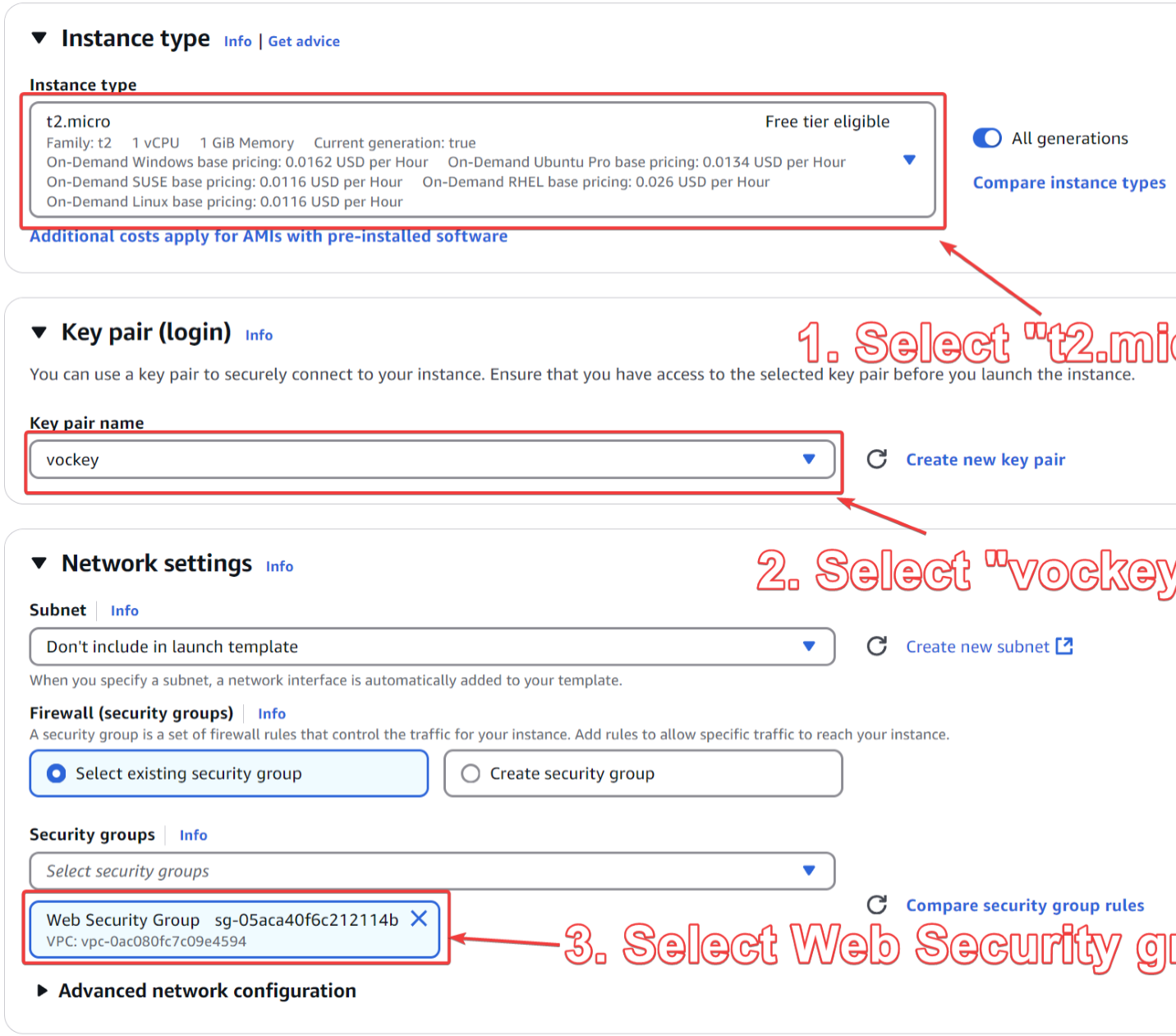
In the left navigational panel, select launch templates



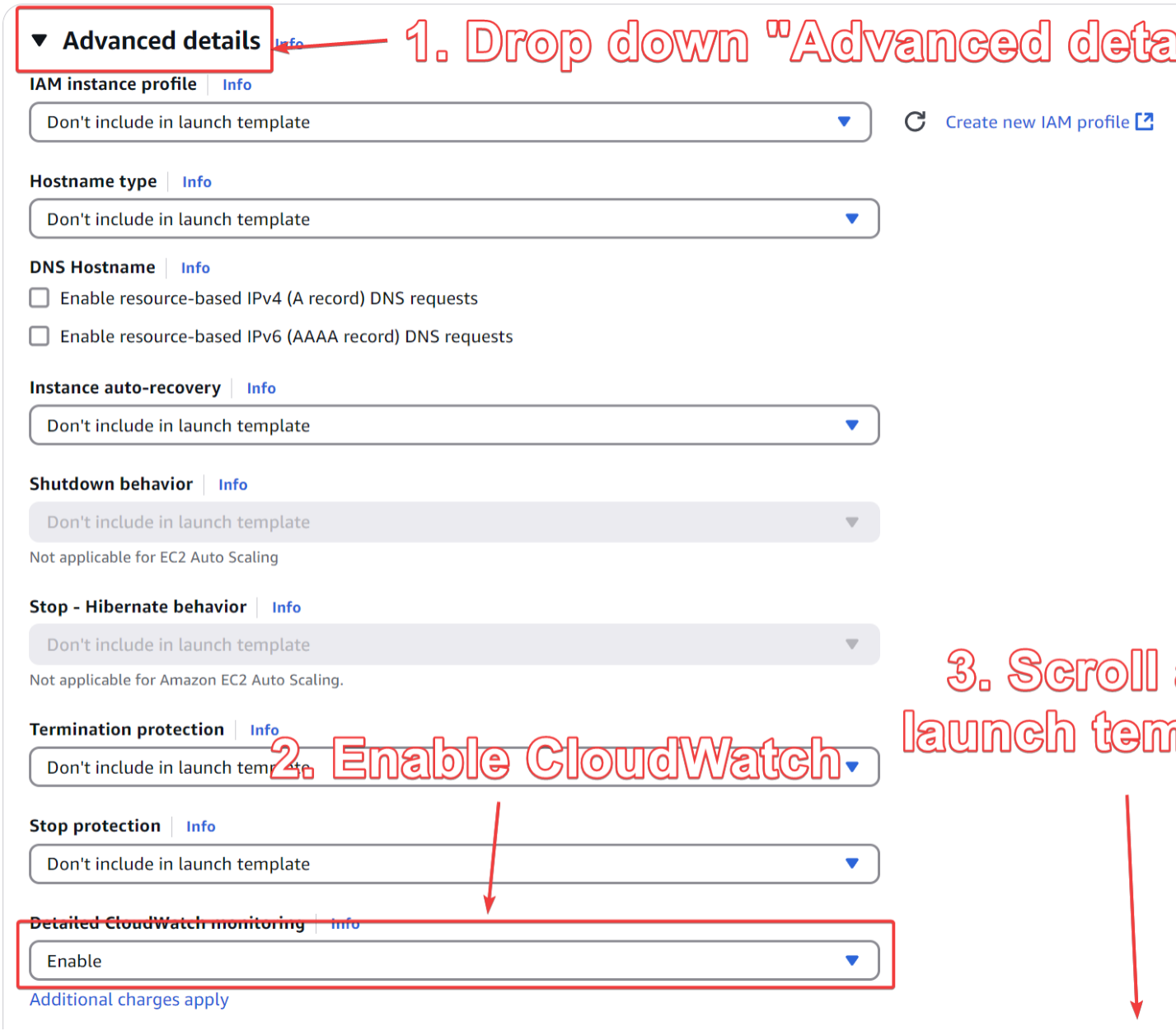
Select Create lanch template



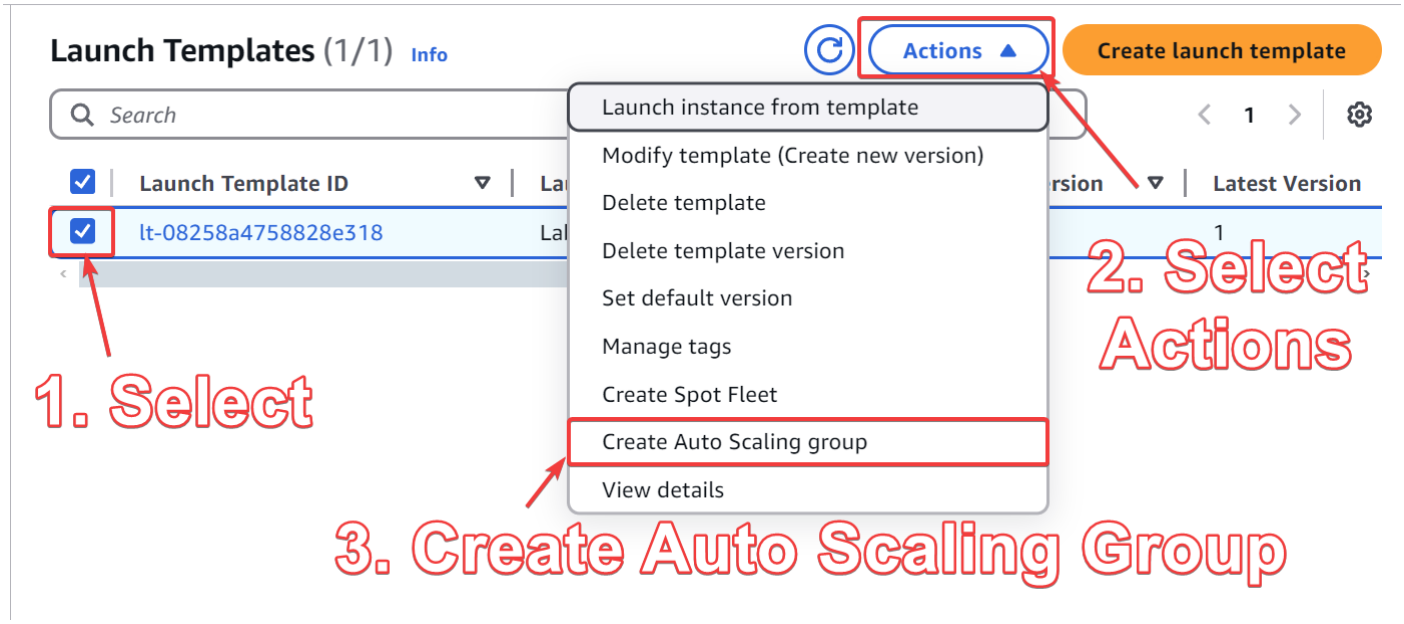
Create a name, select Auto Scaling Guidance, then select the My AMIs tab, then choose Web Server AMI



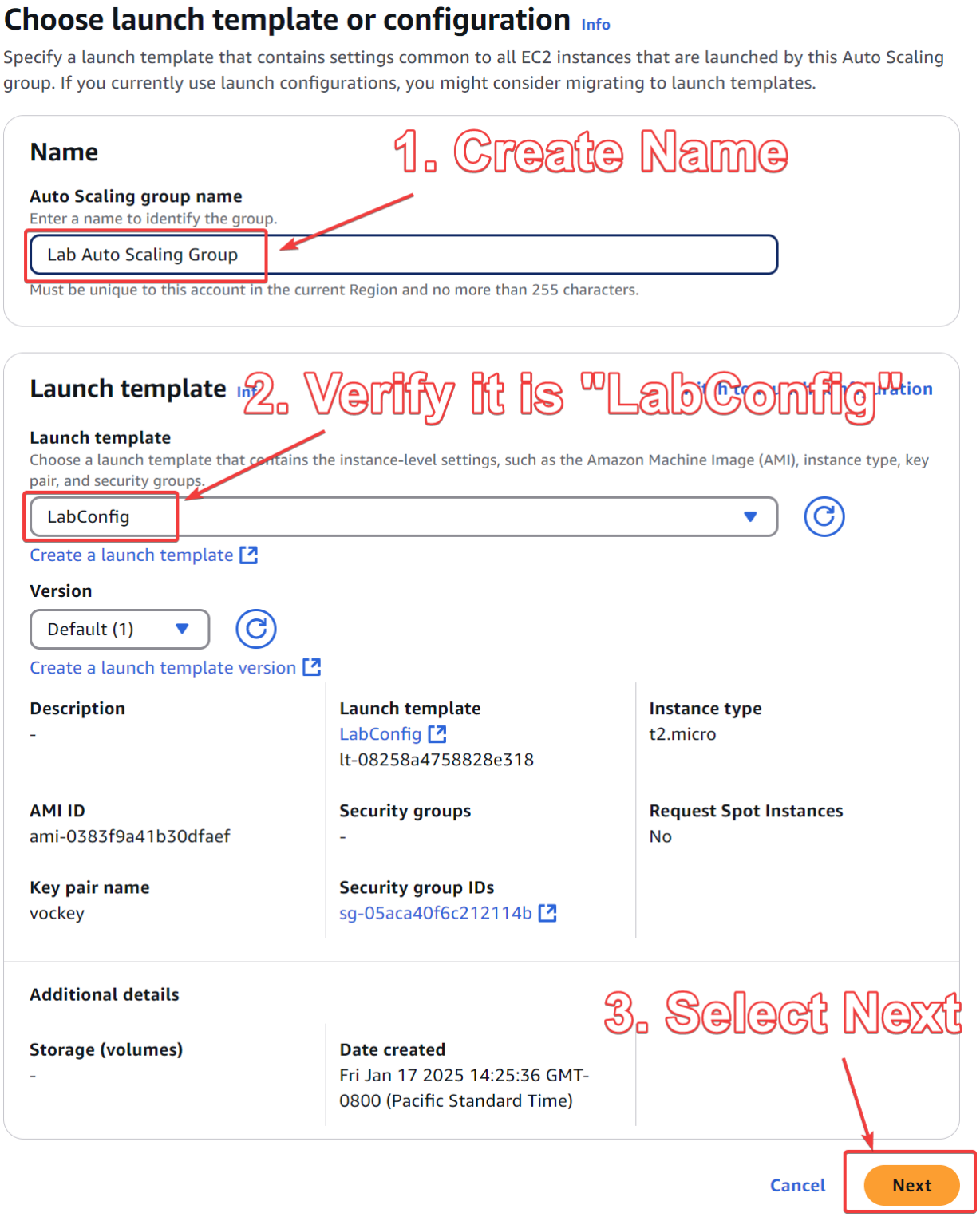
Continuing the Launch server creation, select the t2.micro instance type, then select vockey, and finally make sure Web Security group is the only security group



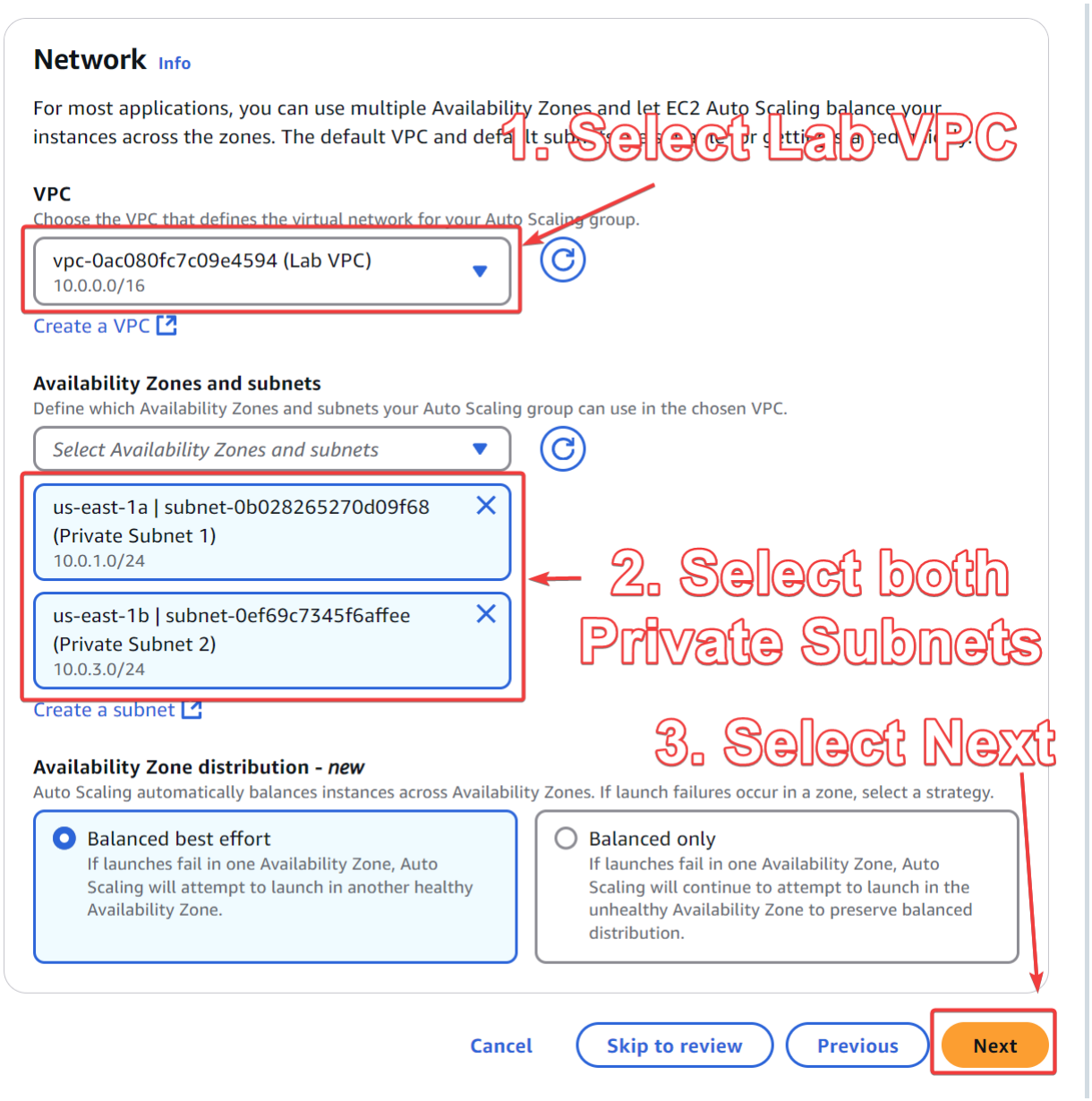
In the drop down menu of advanced details, enable cloudwatch, then scroll down and click launch template



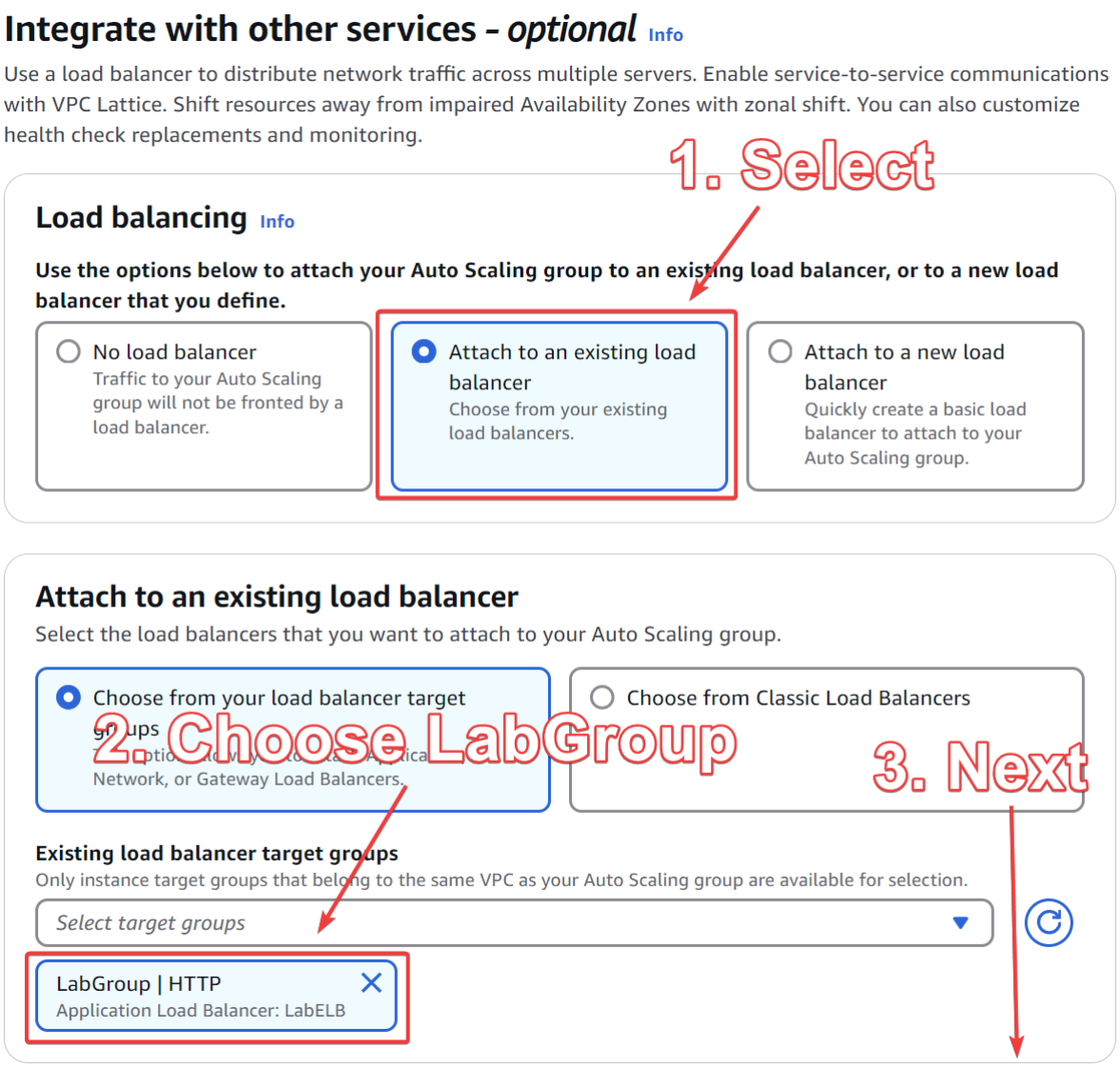
Select the new launch template and then in actions, create auto scaling group



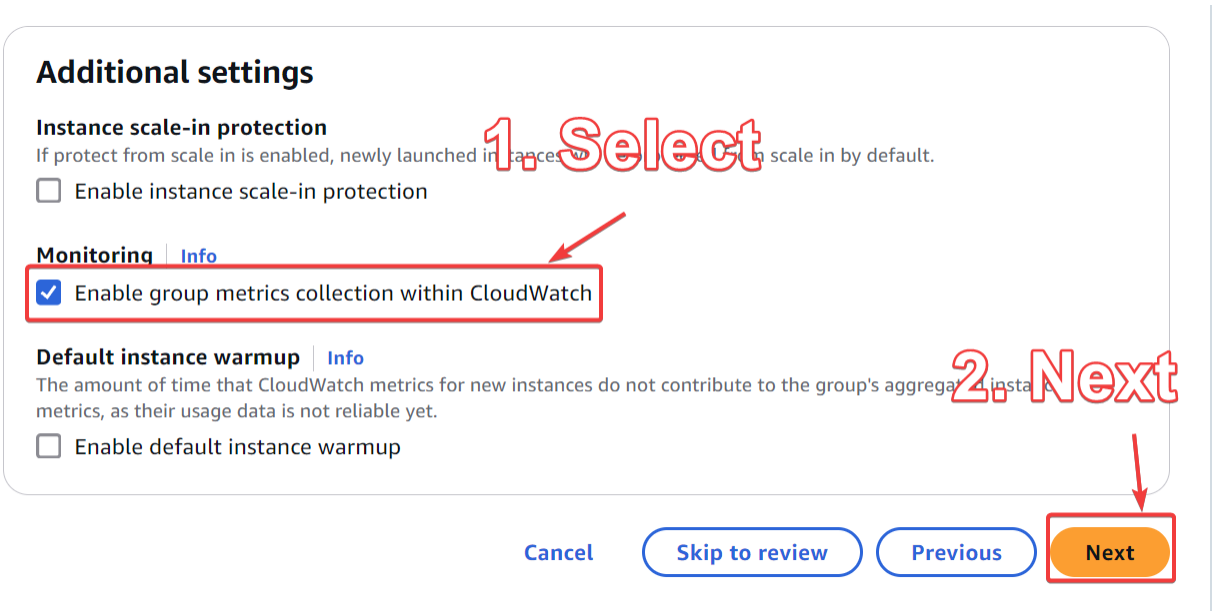
Create a Name, Verify that the launch tempalte is LabConfig, then click next



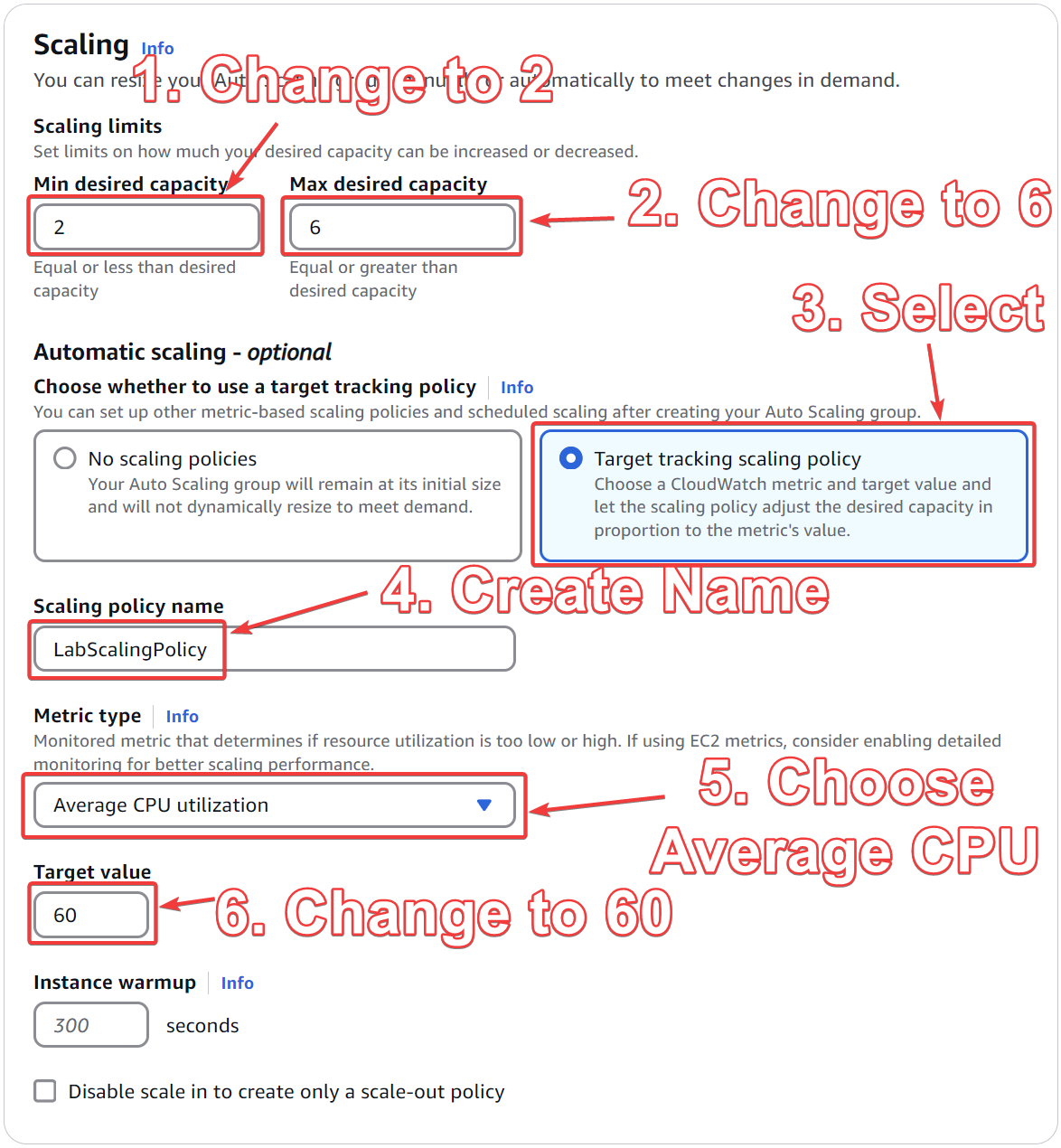
Select the Lab VPC, then select both private subnets, then select next



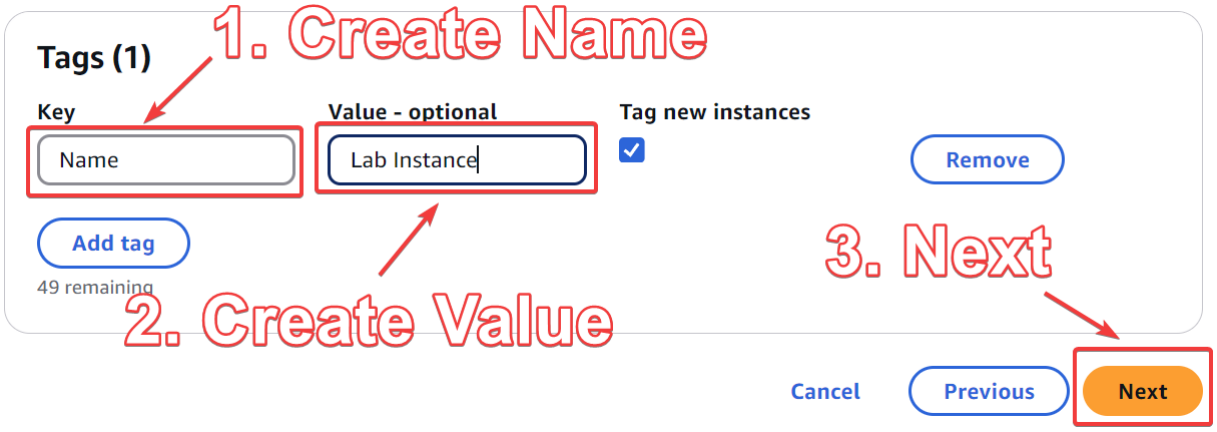
Select, Attach to existing load balancer, choose LabGroup | HTTP, then scroll to click Next



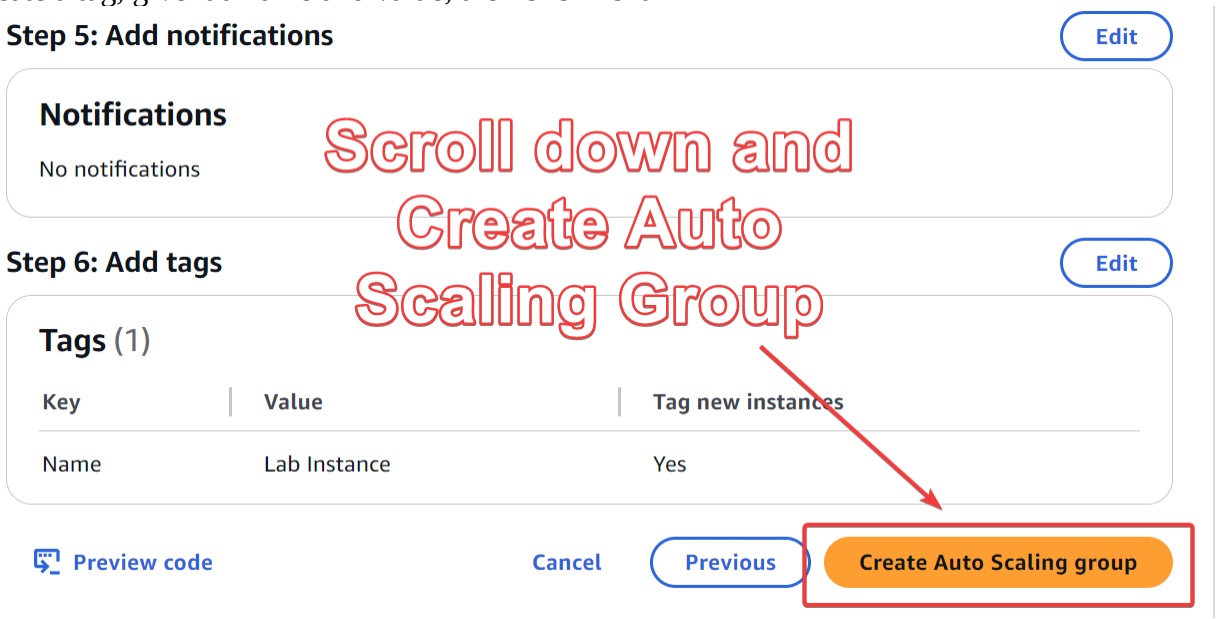
Select Enable group metrics collection within CloudWatch, then click Next



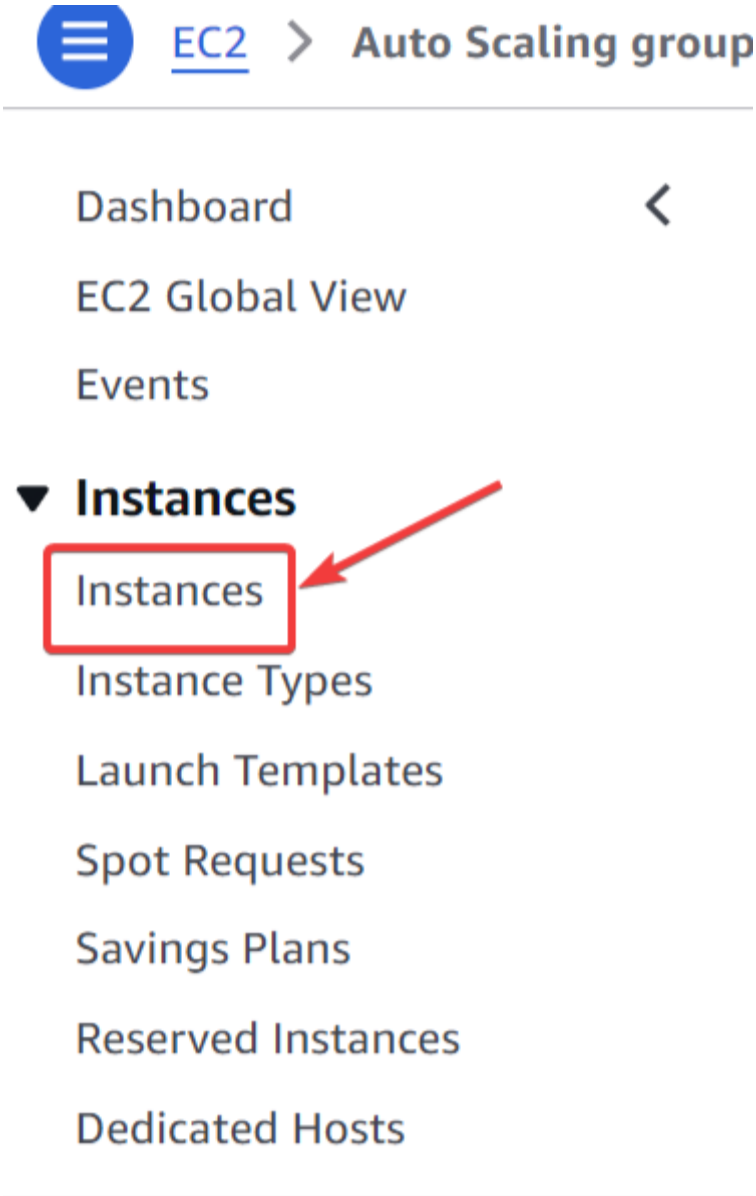
Change the min desired capacity to 2 and the maximum to 6, then select target tracking scaling policy, create a name then select average CPU in metric type, change the target volume to 60



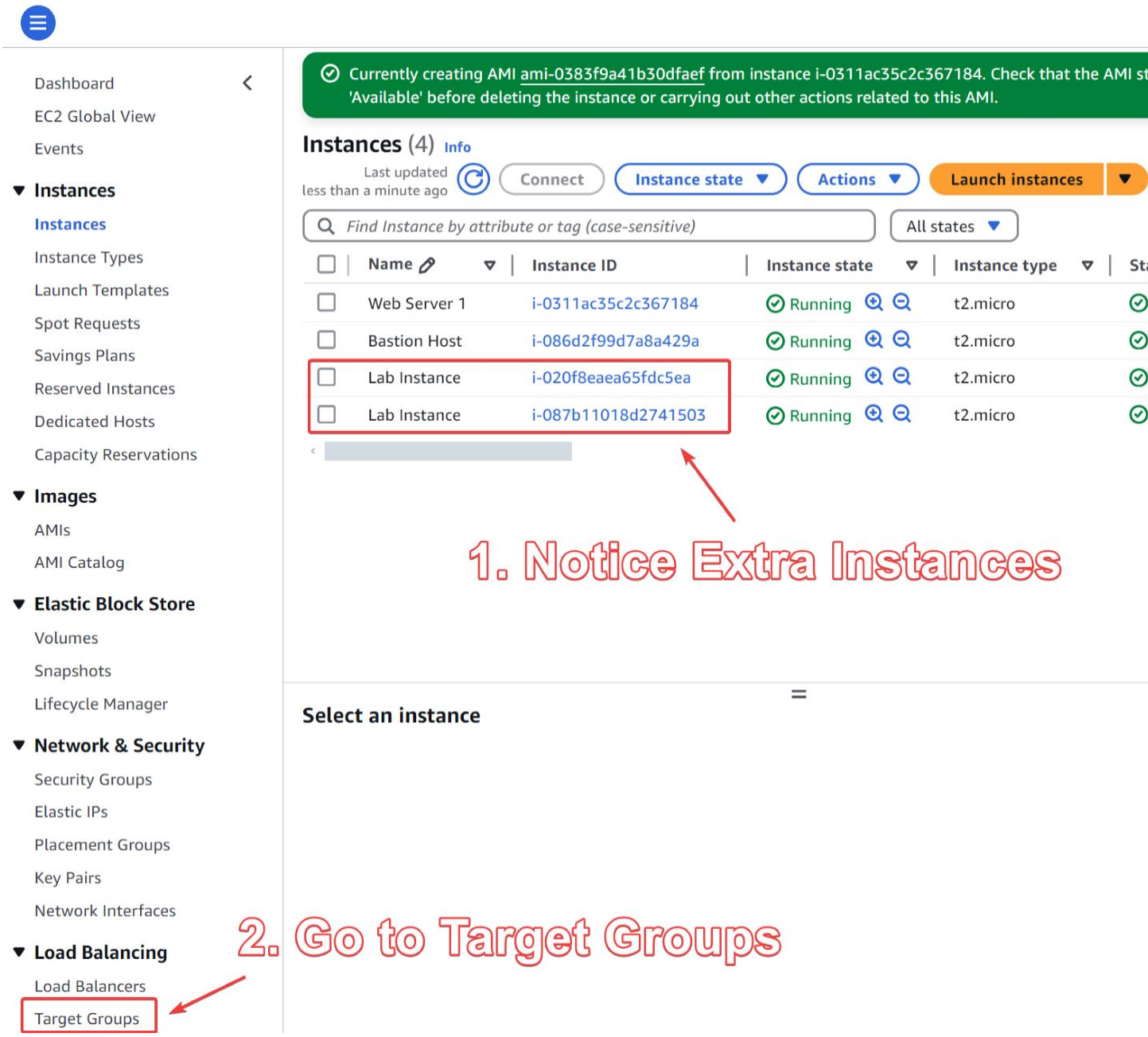
Create a tag, give it a name and value, then click next



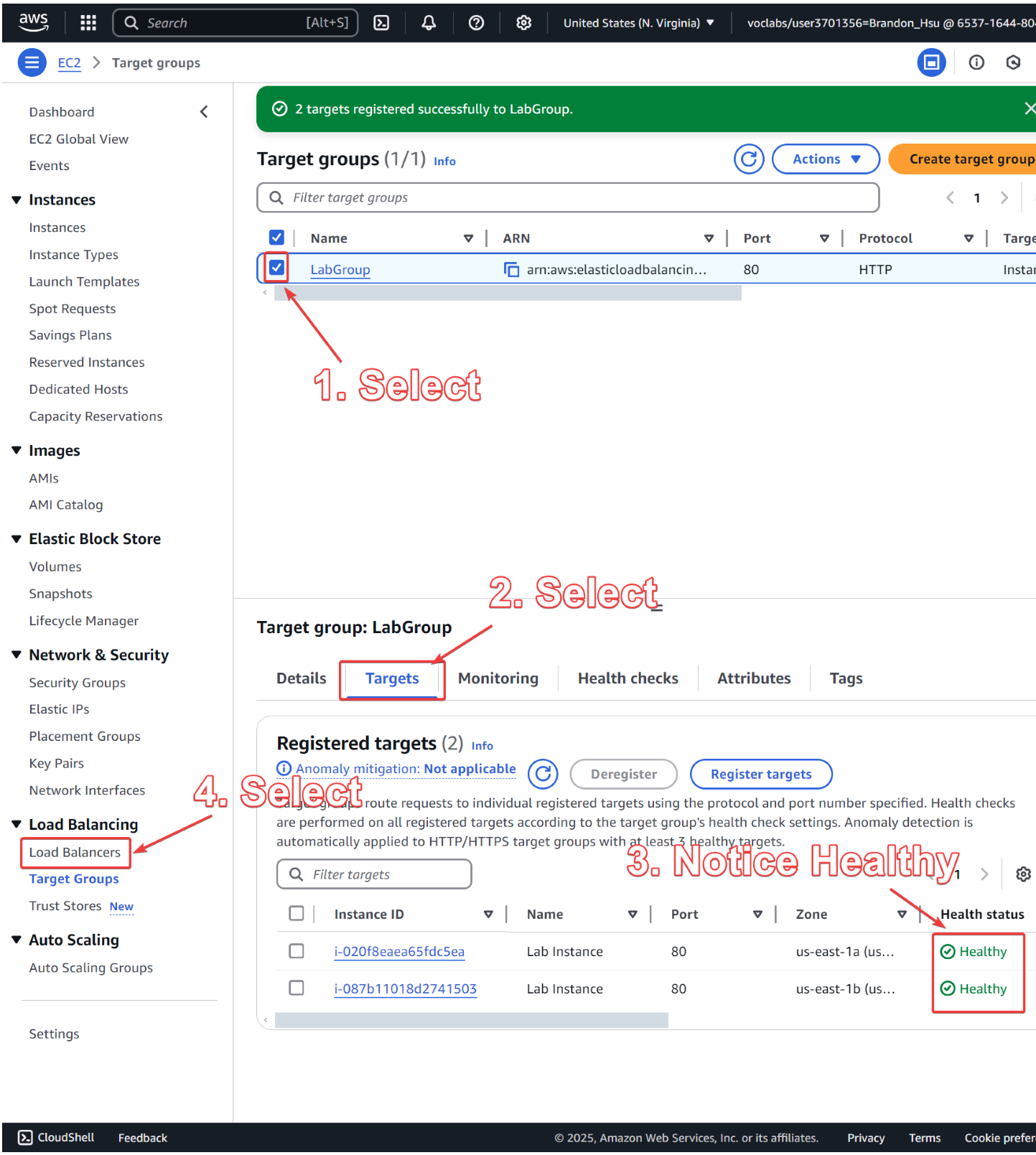
Scroll down and create auto scaling group



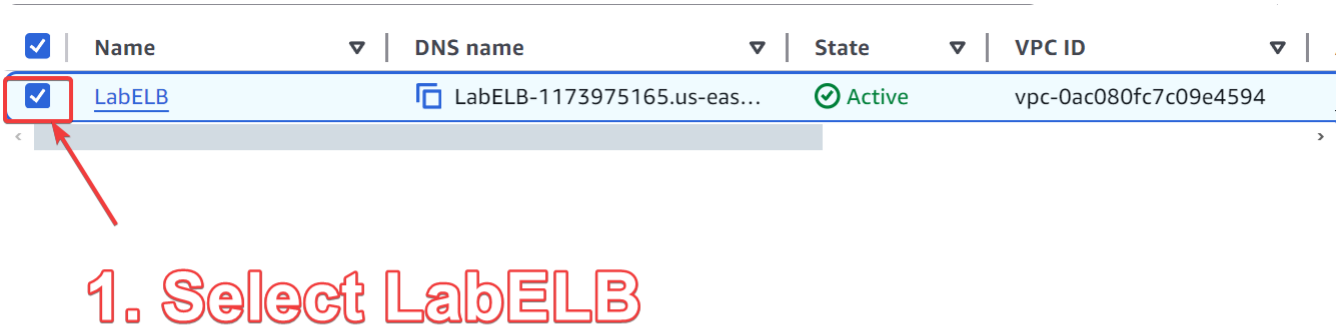
Click Instances in the left navigational panel

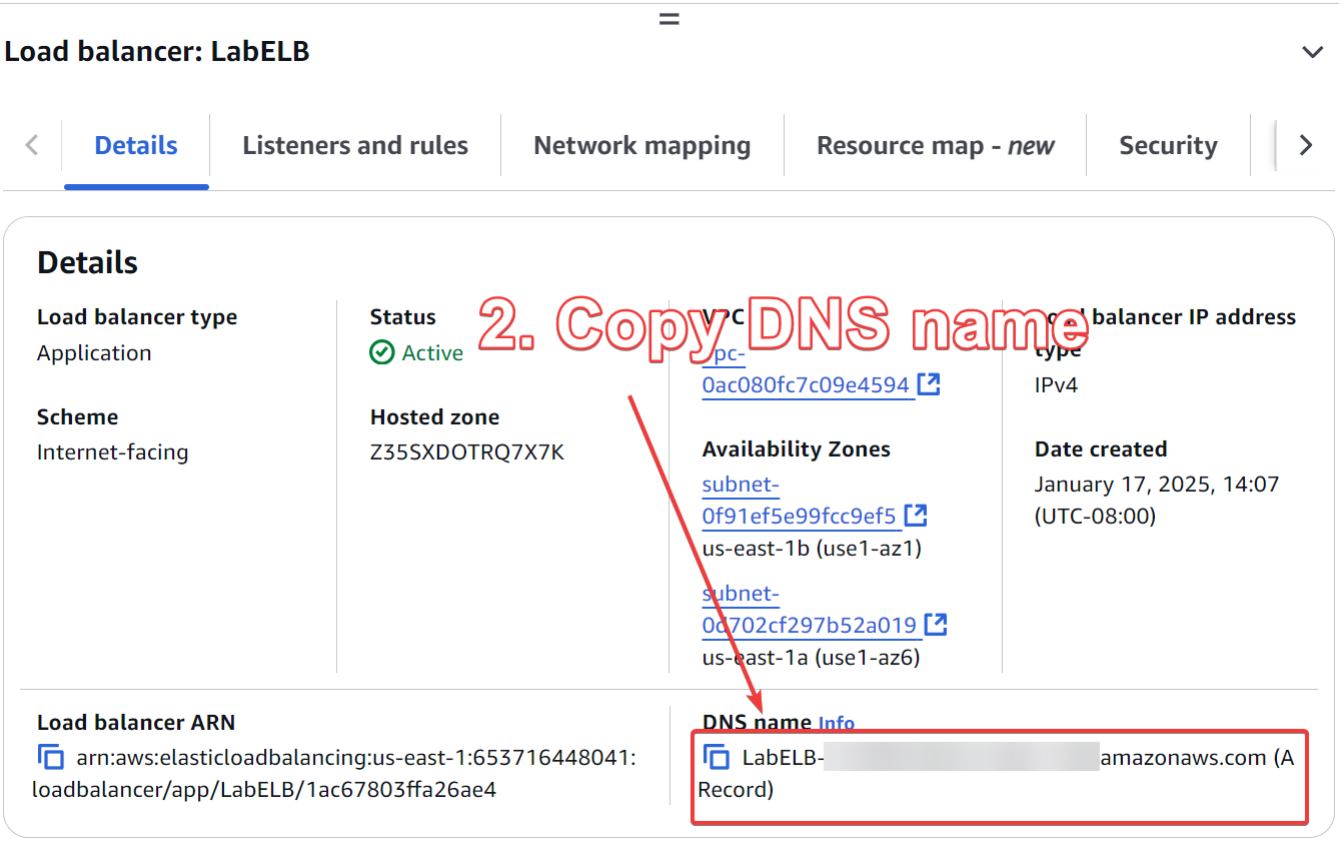


Notice how there are 2 extra instances, then go to Target Groups

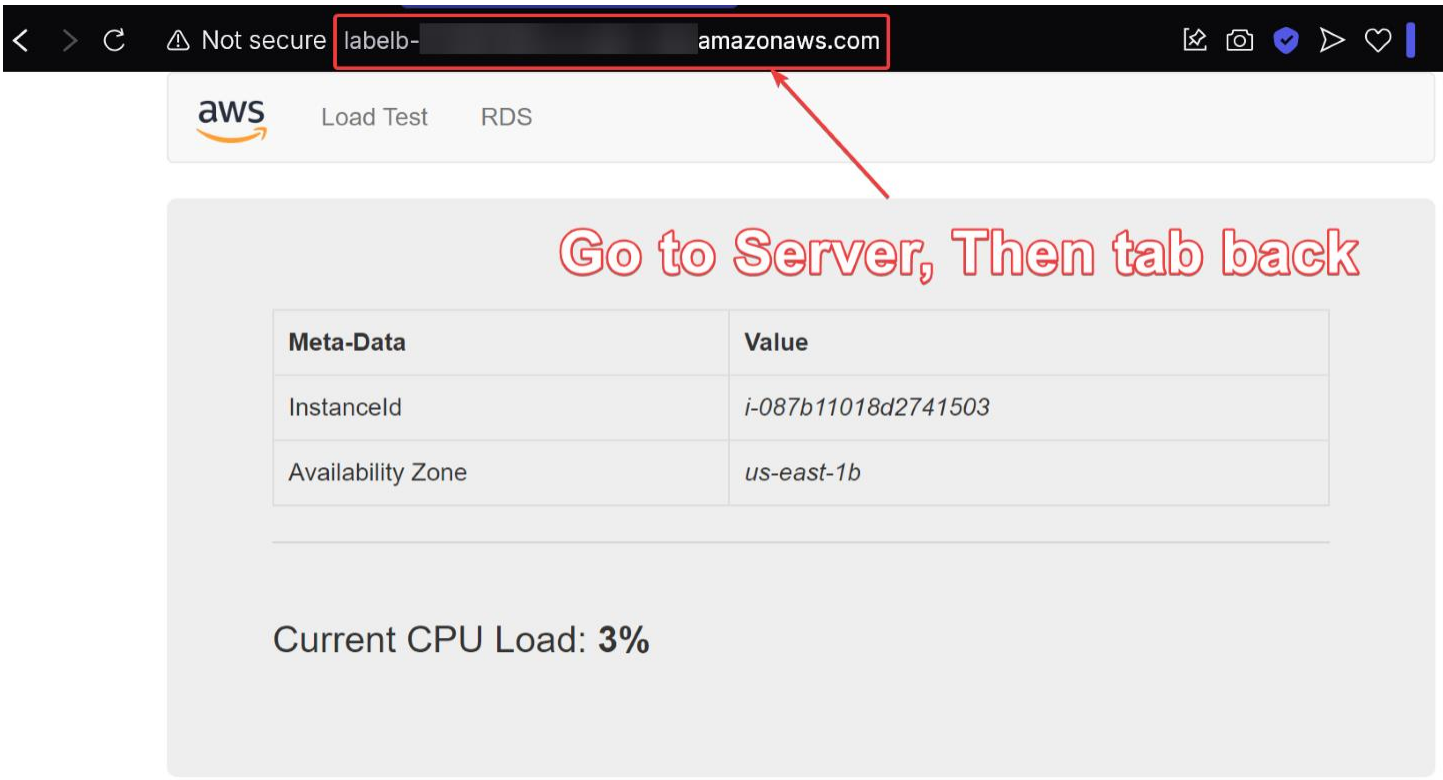


Select the target group and then click on the targets tab, then see that both targets are “Healthy” then in the left navigational panel, select Load Balancers

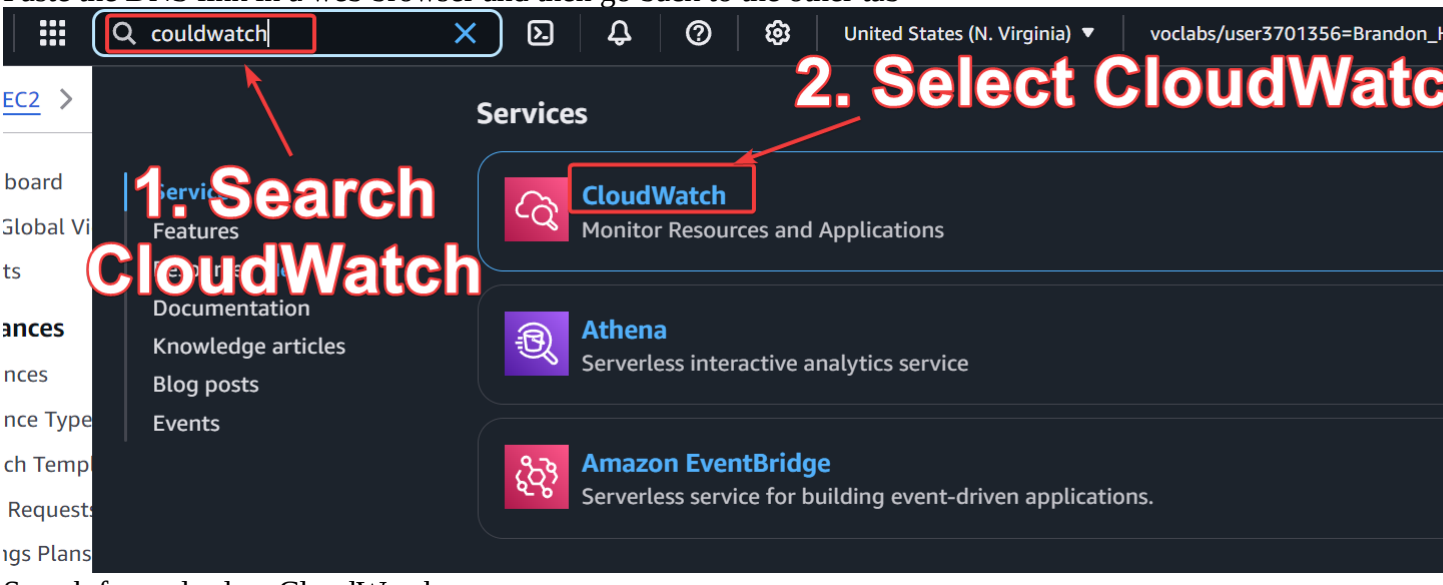




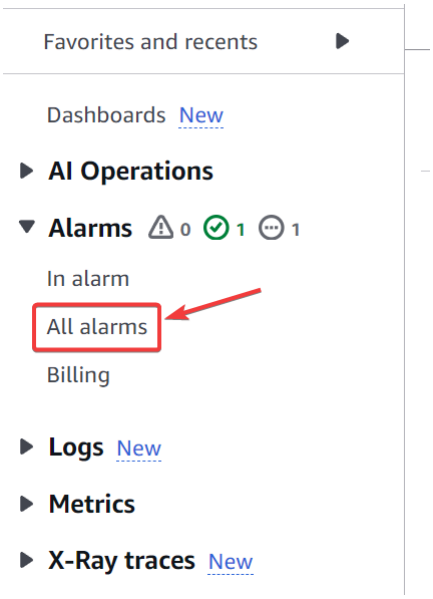
Select the LabELB, then copy the DNS name in the details tab



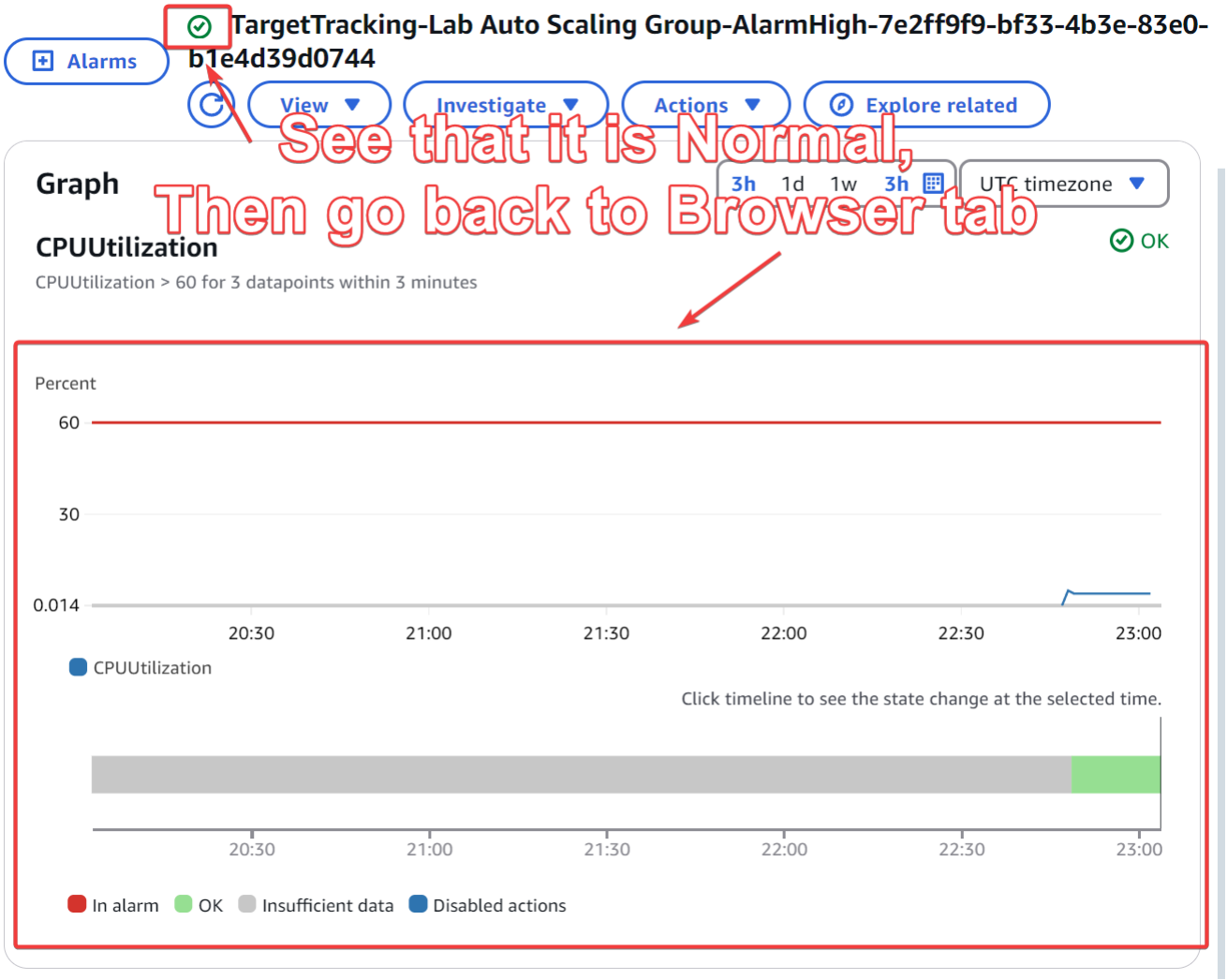
Paste the DNS link in a web browser and then go back to the other tab



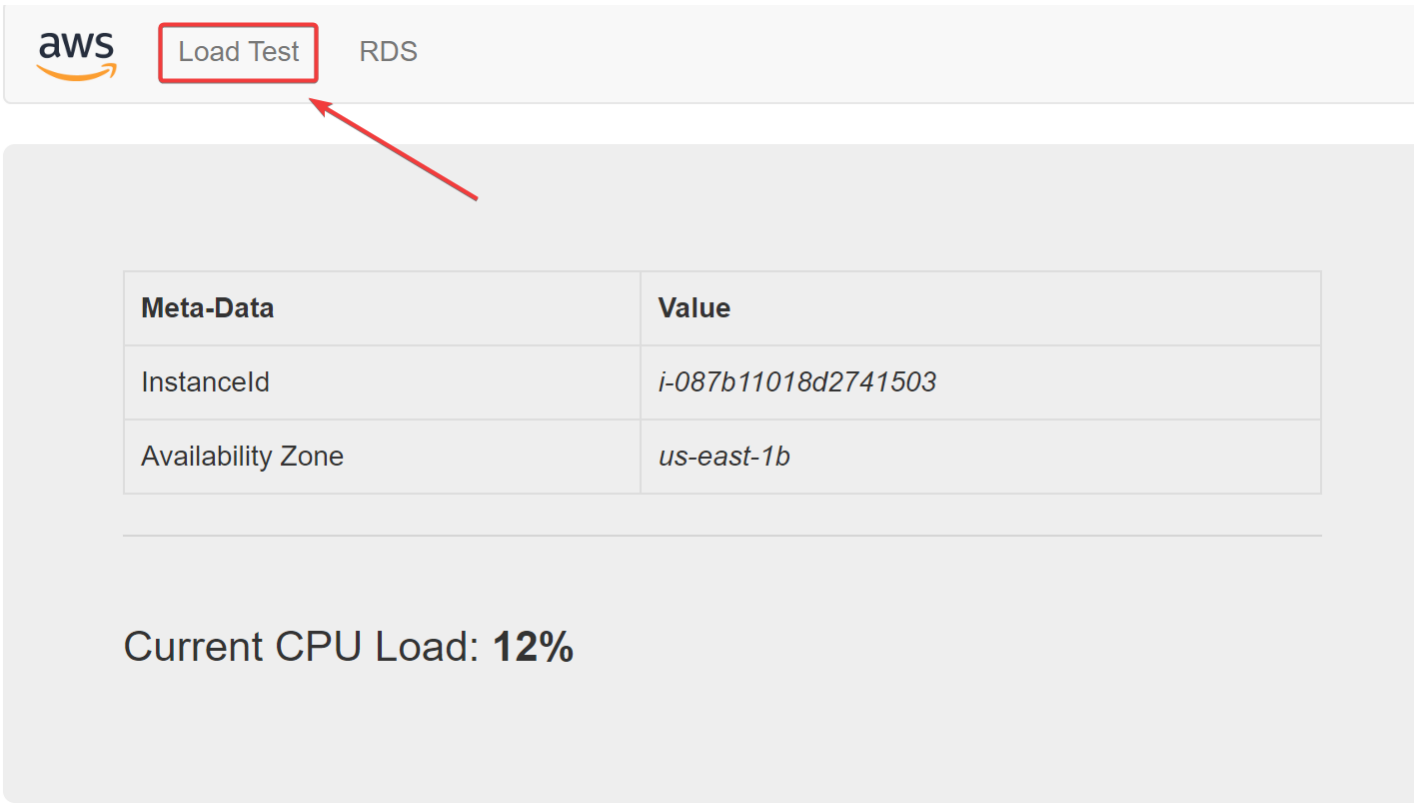
Search for and select CloudWatch



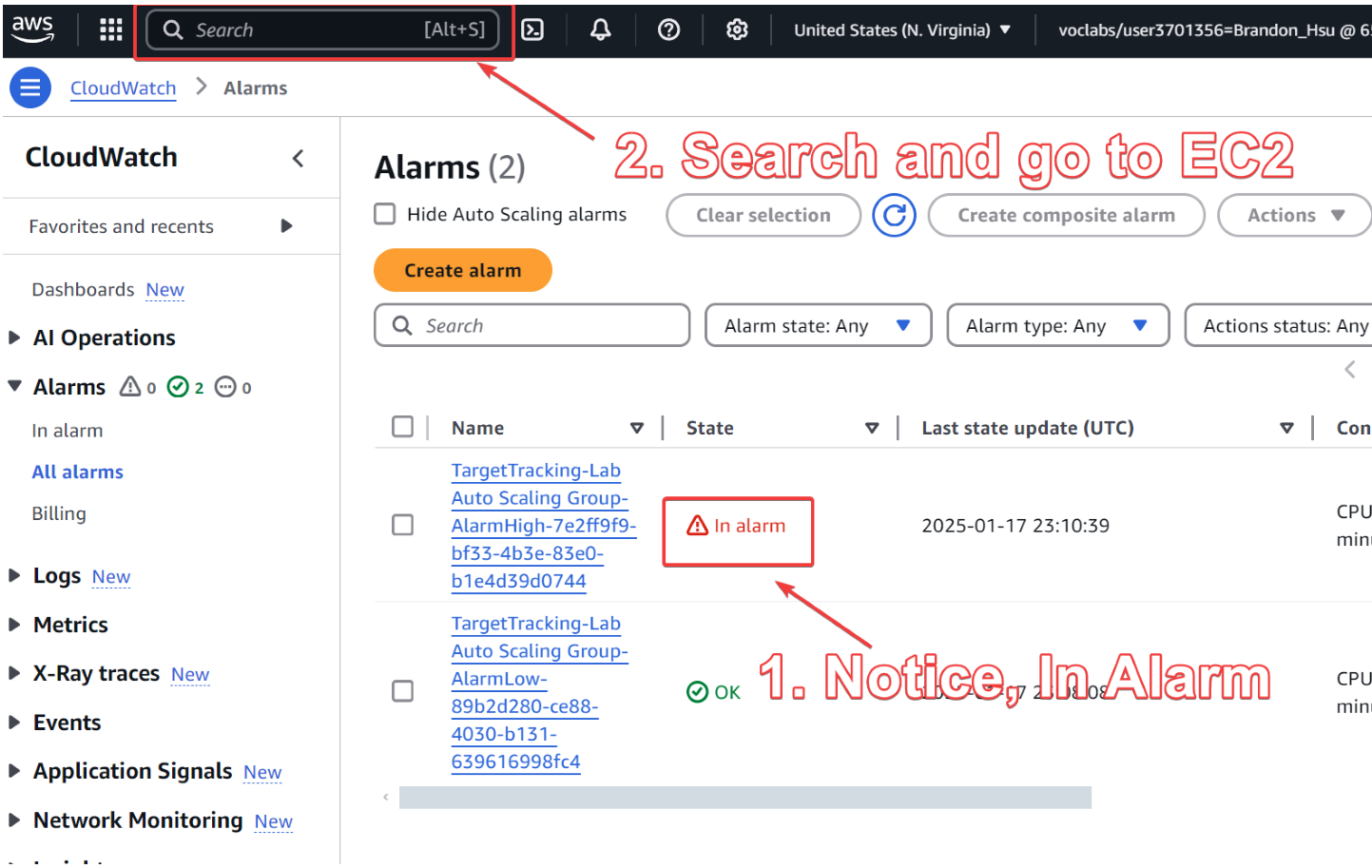
Select All alarms in the left navigational panel



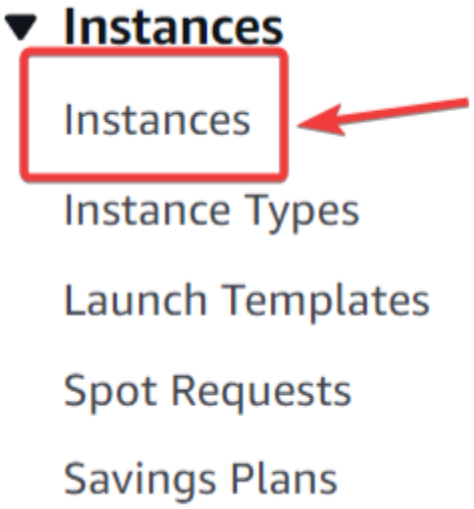
See that all alarms are good, then go to the browser tab



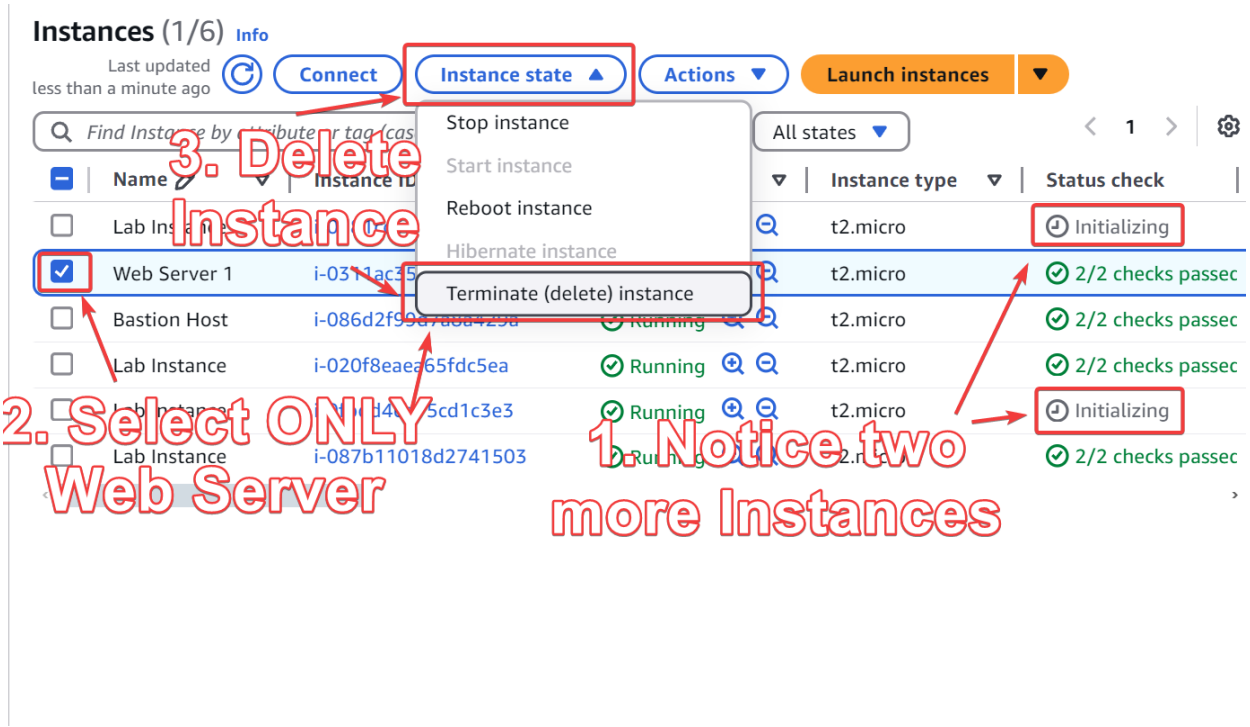
Click Load Test in the top left corner



Notice how the alarm state changed to in alarm, now go back to the EC2 dashboard



Select Instances in the left navigational panel



Notice how there are 2 extra instances, now select the Web Server 1 and then in the Instance state drop down menu, delete the instance
After it deletes, go to the AWS lab launch, then select submit for grading and end lab

Problems

It was more unstable and wouldn't launch a couple times, don't know the issue but I eventually got it to work for all labs.

Conclusion

In conclusion, these labs were incredibly valuable in teaching us not only the core features of AWS but also useful tips and tricks for further application. Specifically, working with EBS enhanced our understanding of how to provide persistent storage in cloud computing, one of the foundational capabilities of AWS. Building a database server showed us an essential tool for data management, backups, performance optimization, and security. The scaling and load balancing lab was particularly significant, as it highlighted a key feature that contributed to AWS's widespread popularity. This lab not only improved our understanding of performance optimization but also taught us about fault tolerance and scalability on a larger scale, which has numerous real-world applications. Overall, these labs expanded our knowledge beyond what we learned earlier and deepened our understanding of how to effectively use AWS, one of the most popular cloud computing platforms.

CCNP Portfolio
Cisco CCNP
IS-IS Lab

5/20/2025

Blizzard, Harrison J

Purpose

The purpose of this lab is to create a network that holds 3 areas having 2 routers per area (areas 1, 2, and 3 respectively) to route Multi-area Intermediate System to Intermediate System (IS-IS) through these areas. Throughout this lab we expect to learn the different uses of Level 1 and Level 1/2 routers to certify the function of the IS-IS network.

Additionally, the purpose was also about learning how to configure the basics of IS-IS in a P2P network when IS-IS runs over IPv4 differently compared to other link-state protocols like OSPF. Furthermore, we found that IS-IS uses hello packets to calculate neighbor adjacencies, and the commands to verify the adjacencies formed between the different routers using IS-IS were functioning properly. Another purpose was for this lab to be an introduction and a gateway to more complex IS-IS concepts like DIS election and multi-area IS-IS configurations.

The end goal of the lab was to experience and learn IS-IS to a point where we could explain how to configure the entire network and at a level that would be acceptable for an expert like Radia Perlman whom which this lab was specifically made for.

Background Information/Lab Concepts

Intermediate System to Intermediate System (IS-IS) runs as a link-state Interior Gateway Protocol (IGP) and is widely recognized for both its high scalability and rapid convergence. At its core, IS-IS finds the shortest route between nodes by employing Dijkstra's algorithm, which functions by finding the shortest path from a source node to all others in the network. It does this by continuously choosing the closest unvisited node and updating distances to neighboring unvisited nodes until all have been addressed.

Routers in an IS-IS environment are categorized into three distinct types. Level 1 routers are designed to function solely within a specific area. Routers classified as Level 1/2 can manage communication both inside a single area and across multiple areas. Meanwhile, Level 2 routers are strictly used for inter-area communication.

Both IS-IS and OSPF fall under the category of IGPs, relying on the link-state approach. They share some characteristics such as using link-state representations, aging mechanisms, and cost-based metrics. Additionally, both support link-state databases and apply Shortest Path First (SPF) algorithms in deciding the best paths. Their routing decision processes, periodic updates, and flooding techniques show notable similarities. However, IS-IS distinguishes itself from OSPF through many architectural and operational differences. IS-IS boasts greater scalability and a more adaptable area structure. It runs directly over the data link layer, enhancing security. In contrast, OSPF employs a more rigid

hierarchy, with a mandatory backbone area (area 0). IS-IS's peering process is more lenient, it doesn't require matching hello/dead intervals or subnet masks, unlike OSPF. Another key contrast lies in their network leadership roles: IS-IS chooses a single Designated Intermediate System (DIS), which can be replaced or preempted; OSPF elects a Designated Router (DR) and a Backup Designated Router (BDR), neither of which are preemptible. Although our lab scenario did not implement a DIS, such devices are crucial in IS-IS when working with multi-access network types. It's also worth noting that IS-IS doesn't support NBMA, point-to-multipoint setups, or virtual links due to its reliance on Layer 2 for operation.

Each IS-IS Network Entity Title (NET) is composed of three elements: the Area ID, System ID, and N-selector. The Area ID typically spans 6 bytes (e.g., 49.0002), while the System ID is 12 bytes long (e.g., 0000.0000.0004). The final part, the N-selector, must be set to 00 (2 bytes) to set up proper IS-IS adjacencies. A complete NET example would be: 49.0002.0000.0000.0004.00.

Lab Summary

Designations:

R1 (Level 1 router)

R2 (Level 1/2 router)

R3 (Level 1/2 router)

R4 (Level 1/2 router)

R5 (Level 1/2 router)

R6 (Level 1 router)

Connections:

PC1 is connected to R1.

R1 is connected to R2 (area 1)

R2 is connected to R3 (areas 1-2)

R3 is connected to R4 (area 2)

R4 is connected to R5 (areas 2-3)

R5 is connected to R6 (area 3)

R6 connected PC2.

(Topology diagram can be found after Lab Configs)

Lab Commands/Configs

“show ip route”: Displays the IPv4 routing table.

“ip routing”: A command to enable the sending of data packets for IPv4 across an IP network.

“traceroute [destination ip address]”: Figuring the data hops from the source ip address to the destination ip address.

“ip router isis”: Enables IS-IS as an IP routing protocol and enters router configuration mode for the protocol.

“show ip protocol”: displays information about the parameters and settings of active IPv4 routing protocols running on the router.

“show isis neighbors”: displays the status of all IS-IS neighbors configured on the router Is-type.

“**net [Area-ID. System-ID. N-Selector] Example Net-id: 49.0002.0000.0000.0004.00 (49.0002 is the Area-ID, 0000.0000.0004 is the system-ID, 00 is the N-Selector)**”: The Net-id acts a unique identifier for the router within the IS-IS domain.

“**is-type level- [level number]**”: sets the routing level for an IS-IS instance.

“**is-type level-1**”: configures a router that operates within an area that runs IS-IS

“is-type level-2”: configures a router to act solely as a router that communicates between areas, taking part in only level 2 routing with IS-IS

“**is-type level-1-2**”: configures a router to act both as a router that runs within an area and between routers with IS-IS

“**metric-style narrow**”: the older, default metric type used for calculating costs between routers.

“**show isis protocol**”: displays information about IS-IS protocol including its configuration and status.

Configs:

R1:

```
interface GigabitEthernet0/0/0
 ip address 10.0.0.4 255.255.255.0
 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 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 isis
 net 49.0001.0000.0000.0001.00
 is-type level-1
 metric-style narrow
```

R2:

```
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 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 isis
 net 49.0001.0000.0000.0002.00
 metric-style narrow


ip forward-protocol nd
no ip http server
ip http secure-server
```

R3:

```
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
 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 49.0002.0000.0000.0003.00
 ip forward-protocol nd
 no ip http server
 no ip http secure-server
 ip tftp source-interface GigabitEthernet0
```

R4:

```
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


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 49.0002.0000.0000.0004.00


ip forward-protocol nd
no ip http server
no ip http secure-server
ip tftp source-interface GigabitEthernet0
```

R5 Configuration:

```
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


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 49.0003.0000.0000.0005.00
```

R6:

```
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 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

router isis
 net 49.0003.0000.0000.0006.00
 is-type level-1
 metric-style narrow

ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip tftp source-interface GigabitEthernet0
```

Show IP Protocol:

R1:

```
 R1#show isis protocol
```

```
 IS-IS Router: <Null Tag> (0x10000)
   System Id: 0000.0000.0001.00  IS-Type: level-1
   Manual area address(es):
         49.0001
   Routing for area address(es):
         49.0001
   Interfaces supported by IS-IS:
         GigabitEthernet0/0/1 - IP
         GigabitEthernet0/0/0 - IP
   Redistribute:
     static (on by default)
   Distance for L2 CLNS routes: 110
   RRR level: none
   Generate narrow metrics: level-1-2
   Accept narrow metrics:   level-1-2
   Generate wide metrics:   none
   Accept wide metrics:     none
```

R2:

```
 R2#show isis protocol

 IS-IS Router: <Null Tag> (0x10000)
   System Id: 0000.0000.0002.00  IS-Type: level-1-2
   Manual area address(es):
         49.0001
   Routing for area address(es):
         49.0001
   Interfaces supported by IS-IS:
         GigabitEthernet0/0/1 - IP
         GigabitEthernet0/0/0 - IP
   Redistribute:
     static (on by default)
   Distance for L2 CLNS routes: 110
   RRR level: none
   Generate narrow metrics: level-1-2
   Accept narrow metrics:   level-1-2
   Generate wide metrics:   none
   Accept wide metrics:     none
```

R3:

```
R3#show isis protocol

 IS-IS Router: <Null Tag>
```

```
   System Id: 0000.0000.0003.00  IS-Type: level-1-2
   Manual area address(es):
         49.0002
   Routing for area address(es):
         49.0002
   Interfaces supported by IS-IS:
         GigabitEthernet0/0/1 - IP
         GigabitEthernet0/0/0 - IP
   Redistribute:
     static (on by default)
   Distance for L2 CLNS routes: 110
   RRR level: none
   Generate narrow metrics: level-1-2
   Accept narrow metrics:   level-1-2
   Generate wide metrics:   none
   Accept wide metrics:     none
```

R4:

```
R4#show isis protocol

System Id: 0000.0000.0004.00  IS-Type: level-1-2
  Manual area address(es):
        49.0002
  Routing for area address(es):
        49.0002
  Interfaces supported by IS-IS:
        GigabitEthernet0/0/1 - IP
        GigabitEthernet0/0/0 - IP
  Redistribute:
    static (on by default)
  Distance for L2 CLNS routes: 110
  RRR level: none
  Generate narrow metrics: level-1-2
  Accept narrow metrics:   level-1-2
  Generate wide metrics:   none
  Accept wide metrics:     none
```

R5:

```
R5#show isis protocol

IS-IS Router: <Null Tag>
  System Id: 0000.0000.0005.00  IS-Type: level-1-2
  Manual area address(es):
        49.0003
```

```
  Routing for area address(es):
        49.0003
  Interfaces supported by IS-IS:
        GigabitEthernet0/0/1 - IP
        GigabitEthernet0/0/0 - IP
  Redistribute:
    static (on by default)
  Distance for L2 CLNS routes: 110
  RRR level: none
  Generate narrow metrics: level-1-2
  Accept narrow metrics:   level-1-2
  Generate wide metrics:   none
  Accept wide metrics:     none
```

R6:

```
R6#show isis protocol
IS-IS Router: <Null Tag> (0x10000)
  System Id: 0000.0000.0006.00  IS-Type: level-1
  Manual area address(es):
        49.0003
  Routing for area address(es):
        49.0003
  Interfaces supported by IS-IS:
        GigabitEthernet0/0/1 - IP
        GigabitEthernet0/0/0 - IP
  Redistribute:
    static (on by default)
  Distance for L2 CLNS routes: 110
  RRR level: none
  Generate narrow metrics: level-1-2
  Accept narrow metrics:   level-1-2
  Generate wide metrics:   none
  Accept wide metrics:     none
```

> Show IP route/> Traceroute for IPv4:

R1:

```
R1#show ip route
 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
        ia - IS-IS inter area, * - candidate default, U - per-user
static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l -
LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides
from PfR

 Gateway of last resort is not set

 i L2  192.168.1.0/24 [115/20] via 192.168.2.1, 3d00h,
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 L1  192.168.4.0/24 [115/20] via 192.168.3.2, 3d00h,
GigabitEthernet0/0/0
 i L2  192.168.5.0/24 [115/30] via 192.168.3.2, 3d00h,
GigabitEthernet0/0/0
```

R2:

```
R2#show ip route
 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
        ia - IS-IS inter area, * - candidate default, U - per-user
static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l -
LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides
from PfR

 Gateway of last resort is not set

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, 3d00h,
GigabitEthernet0/0/1
 i L2  192.168.4.0/24 [115/30] via 192.168.2.2, 3d00h,
GigabitEthernet0/0/1
 i L2  192.168.5.0/24 [115/40] via 192.168.2.2, 3d00h,
GigabitEthernet0/0/1
```

R3:

```
 R3#show ip route
 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
        ia - IS-IS inter area, * - candidate default, U - per-user
static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l -
LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides
from PfR

 Gateway of last resort is not set

 i L2  192.168.1.0/24 [115/20] via 192.168.2.1, 3d00h,
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 L1  192.168.4.0/24 [115/20] via 192.168.3.2, 3d00h,
GigabitEthernet0/0/0
 i L2  192.168.5.0/24 [115/30] via 192.168.3.2, 3d00h,
GigabitEthernet0/0/0
```

R4:

```
 R4#show ip route
 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
```

```
        ia - IS-IS inter area, * - candidate default, U - per-user
static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l -
LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides
from PfR

 Gateway of last resort is not set

 i L2  192.168.1.0/24 [115/30] via 192.168.3.1, 3d00h,
GigabitEthernet0/0/0
 i L1  192.168.2.0/24 [115/20] via 192.168.3.1, 3d00h,
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, 4d00h,
GigabitEthernet0/0/1
```

R5:

```
 R5#show ip route
 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
        ia - IS-IS inter area, * - candidate default, U - per-user
static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l -
LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides
from PfR

 Gateway of last resort is not set

 i L2  192.168.1.0/24 [115/40] via 192.168.4.1, 3d00h,
GigabitEthernet0/0/1
 i L2  192.168.2.0/24 [115/30] via 192.168.4.1, 3d00h,
GigabitEthernet0/0/1
 i L2  192.168.3.0/24 [115/20] via 192.168.4.1, 3d00h,
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
```

R6:

```
R6#show ip route
 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
        ia - IS-IS inter area, * - candidate default, U - per-user
static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l -
LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides
from PfR

 Gateway of last resort is 192.168.5.1 to network 0.0.0.0

 i*L1  0.0.0.0/0 [115/10] via 192.168.5.1, 4d00h, GigabitEthernet0/0/0
 i L1  192.168.4.0/24 [115/20] via 192.168.5.1, 4d00h,
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
```

Traceroutes:

R1 to R6

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

R6 to R1

```
R6#traceroute 192.168.1.1
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.5.1 1 msec 1 msec 0 msec
  2 192.168.4.1 1 msec 1 msec 0 msec
  3 192.168.3.1 1 msec 1 msec 0 msec
  4 192.168.2.1 1 msec 1 msec 1 msec
  5 192.168.1.1 2 msec 1 msec *
```

“> show isis neighbors”:

R1 :

```
R1#show isis neighbors

 System Id       Type Interface     IP Address      State Holdtime
Circuit Id
 R2              L1   Gi0/0/0       192.168.1.2     UP    26
R1.01
```

R2:

```
R2#show isis neighbors

 System Id       Type Interface     IP Address      State Holdtime
Circuit Id
 R1              L1   Gi0/0/0       192.168.1.1     UP    7
R1.01
 R3              L2   Gi0/0/1       192.168.2.2     UP    24
R2.02
```

R3:

```
 R3#show isis neighbors

 System Id       Type Interface     IP Address      State Holdtime
Circuit Id
 R2              L2   Gi0/0/1       192.168.2.1     UP    9
R2.02
 R4              L1   Gi0/0/0       192.168.3.2     UP    23
```

```
R3.01
 R4              L2   Gi0/0/0       192.168.3.2     UP    23
R3.01
```

R4:

```
R4#show isis neighbors

 System Id       Type Interface     IP Address      State Holdtime
Circuit Id
 R3              L1   Gi0/0/0       192.168.3.1     UP    7
R3.01
 R3              L2   Gi0/0/0       192.168.3.1     UP    9
R3.01
 R5              L2   Gi0/0/1       192.168.4.2     UP    6
R5.02
```

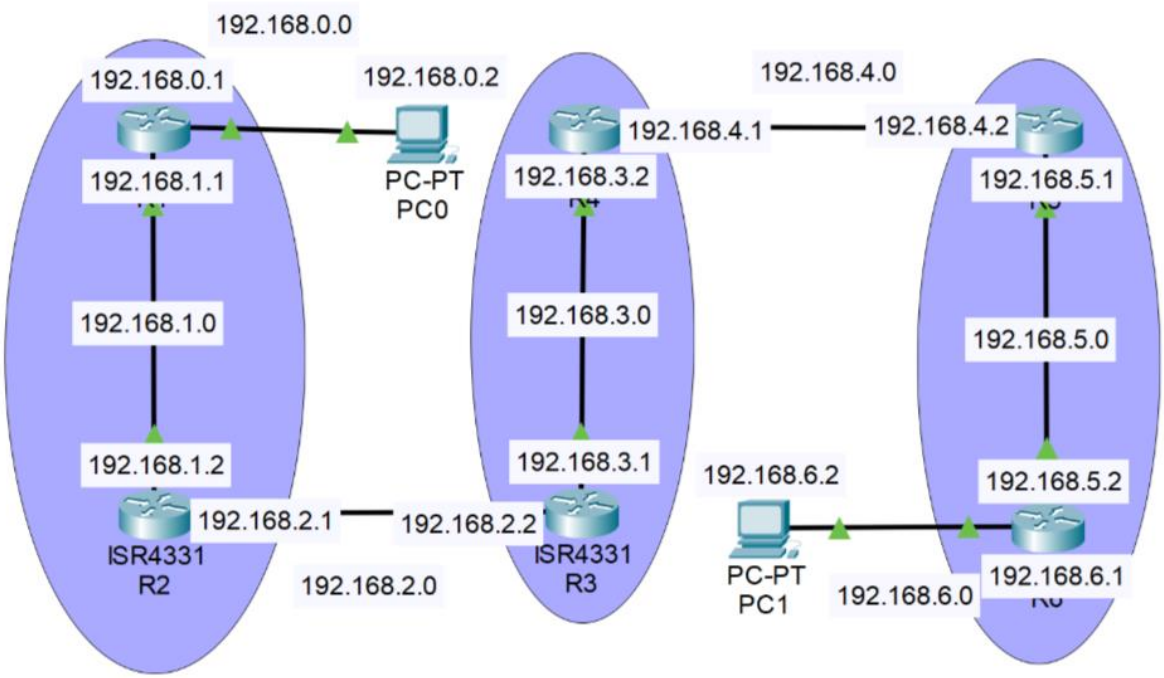
R5:

```
 R5#show isis neighbors

 System Id       Type Interface     IP Address      State Holdtime
Circuit Id
 R4              L2   Gi0/0/1       192.168.4.1     UP    22
R5.02
 R6              L1   Gi0/0/0       192.168.5.2     UP    26
R5.01
```

R6:

```
R6#show isis neighbors

 System Id       Type Interface     IP Address      State Holdtime
Circuit Id
 R5              L1   Gi0/0/0       192.168.5.1     UP    9
```

Lab Diagram



Problems

What was the most challenging:

At first, setting up the Net-ID in IS-IS was pretty confusing, mostly because of how OSPF doesn't use anything similar. This required us to familiarize ourselves with how Net-IDs function and how to properly configure them so that the routing setup worked without any flaws. We also found it a little challenging to fully understand the differences between the router levels (Levels 1, 2, and 1/2) and how to correctly assign these levels when building out a 3 area, P2P IS-IS topology. Additionally, adjusting to the IS-IS specific commands like "is-type", took time to change over to, since they are quite a bit different from the commands that we used previously with other routing protocols.

Concepts Covered:

Throughout the lab, we explored how IS-IS and OSPF and became familiar with how to correctly configure Net-IDs for IS-IS. We gained a clearer understanding of the several

types of Intermediate Systems: Level 1 routers run only within their own area, Level 2 routers function strictly between areas, and Level 1/2 routers are capable of both intra- and inter-area routing. One key distinction from OSPF that we learned is that all interfaces on a router in IS-IS are assigned to the same area. This contrasts with OSPF, where individual interfaces can belong to different areas.

What Caused Problems:
One of the significant issues we had involved misconfiguring the is-type setting. At the start we used only Level 1 routers, which caused a failure in pings between different areas. We resolved this by introducing Level 1/2 routers, which made proper inter-area communication function. Another mistake that led to issues was forgetting to assign a Net-ID entirely, and then later configuring it incorrectly by treating it like an OSPF area ID. Instead of using the correct format (Area ID, followed by System ID, and then N-selector) we applied the wrong structure. Moreover, we incorrectly placed interfaces from a single router into separate areas, like what we would do in OSPF. However, IS-IS requires that all interfaces on a router belong to the same area, which was something we had to change.

Conclusion

Through this lab, we had to create a network that holds 3 areas having 2 routers per area (areas 1, 2, and 3 respectively) to route Multi-area Intermediate System to Intermediate System (IS-IS) through these areas. Throughout this lab we expect to learn the different uses of Level 1 and Level 1/2 routers to certify the function of the IS-IS network.

Despite the problems, we were able to figure them out and work together to correctly configure and set up our Network. This will undoubtedly be helpful in understanding the diversity and architecture of routing protocols and help set us up for our eventual class talk with Radia Perlman.

5/20/2025

Wireless Multi-SSID Access Point Lab

Blizzard, Harrison J

Purpose

This lab focused on extending the range and accessibility of a wireless internet connection by setting up an access point. The goal was to allow users to connect from locations further away from the primary router. As part of the lab, we deployed a standalone Wireless Access Point configured with three different SSIDs, each offering a different level of network security.

The first SSID was configured as completely open with no security, the second SSID required a standard password for access, and the third SSID used security via a RADIUS server where usernames and passwords were needed for authentication. Each SSID was assigned to its own VLAN, and we implemented inter-VLAN routing through a switch which was connected to a router configured for router-on-a-stick, which provided access to the internet.

By having multiple SSIDs using both WPA2-PSK and WPA2-Enterprise security methods, we were able to create separate network environments set to different user groups and use cases. This kind of setup is especially valuable when there’s a need to have network traffic separated.

Background Information/Lab Concepts

An access point serves as the platform for establishing multiple separate Wi-Fi networks. It functions as a bridge, enabling wireless devices to communicate with wired networks. Additionally, it extends wireless coverage and increases the number of users who can connect to the same network.

Each wireless network is identified by an SSID, or Service Set Identifier. This label acts as the network’s name and helps differentiate it from other networks visible on a device.

VLANs (Virtual Local Area Networks), group devices logically as if they were on the same physical network—even if they're located in different physical locations. In this lab, VLANs were configured to isolate each SSID, allowing us to create multiple secure wireless networks that operate independently while sharing the same physical infrastructure.

WPA2-PSK is a common wireless security protocol that uses a pre-shared key (which is just a password) to allow access and encrypt transmitted data.

WPA2-Enterprise, on the other hand, offers a higher level of security than WPA2-PSK. It uses strong encryption along with individual user credentials—requiring both a unique username and password per user. In our lab, this protocol was also used on one of the three networks to demonstrate enterprise-grade access control.

DHCP (Dynamic Host Configuration Protocol), automates the assignment of IP addresses to devices on a network. In this lab, DHCP was used to dynamically assign IP addresses to clients connecting through each of the three VLANs tied to our SSIDs.

NAT (Network Address Translation) is a method that allows several devices on a private internal network to share a single public IP address when accessing the internet. We used NAT in the lab to help conserve public IP space and enable internet connectivity.

Lastly, the RADIUS server was the most important part in managing authentication for the third SSID. Which was used as a central point for Authentication, Authorization, and Accounting (AAA), the RADIUS server provided secure credential verification for devices trying to connect to the WPA2-Enterprise wireless network.

Lab Summary

Router from port G0/0/1 connected to the Internet.

Router from port G0/0/0 connected to Switch at port F0/19

Sub-interfaces RG0/0/0/0.10, RG0/0/0.20, and RG0/0/0.30 configured for access to the AP, PC, and the Radius server.

Switch from port f0/17 connected to Access-point

Switch from port f0/19 connected to RADIUS server.

Switch from port f0/20 connected to PC.

Lab Commands/Configs

“**show ip route**”: Displays the IPv4 routing table

“**ip routing**”: A command to enable the sending of data packets for IPv4 across an IP network

“**traceroute [destination ip address]**”: Figuring the data hops from the source ip address to the destination ip address.

“**Switchport mode access**”: Configures an interface to operate in access mode and carries traffic for only one VLAN.

“**Switchport trunk encapsulation dot1q**”: Configures an interface to operate in trunk mode and allows it to carry traffic for multiple vlans

“**Switchport trunk allowed vlan [vlan numbers separated by commas]**”: specifics which VLANs a trunk port is allowed to carry traffic for.

“**encapsulation dot1q [VLAN ID]**”: configures an interface to handle Ethernet frames with VLAN ID given.

“**ip dhcp pool [Pool_Name]**”: Creates a DHCP pool with the specified name which will define the parameters for IP address assignment on the network

“**Network [IP Address network] [Subnet Mask]**”: Specifies the IP address range that the DHCP pool will manage

“**Default-router [IP address]**”: Specifies the default gateway/router IP address that will be given to DHCP clients

“**Dns-server 8.8.8.8**”: Gives the DNS server IP address that will be handed out to DHCP clients, with 8.8.8.8 is the public DNS server operated by Google.

“**ip dhcp excluded-address [ip address range]**”: Instructs a DHCP server to prevent assigning IP addresses from the ip range given to DHCP clients.

“**ip nat inside source list 1 interface [specified interface] overload**”: enables NAT on the specified Interface which helps to translate source addresses from the network defined in the access list

“**access-list 1 permit [IP address] [Wildcard-mask]”:** defines a network (192.168.1.0/24) for which NAT will be applied.

Configurations:

AP CLI Configuration:

```
version 15.3
 no service pad
 service timestamps debug datetime msec
 service timestamps log datetime msec
 service password-encryption
 hostname Lab-AP
 logging rate-limit console 9
 enable secret 5 $1$3xv/$tCL154uWhyGwPSTjT2gBC1
 aaa new-model
 aaa group server radius rad_eap
 server name camera-ThinkStation-P7
 aaa group server radius rad_mac
 aaa group server radius rad_acct
 server name camera-ThinkStation-P7
```

```
aaa group server radius rad_admin
server name camera-ThinkStation-P7
aaa group server tacacs+ tac_admin
aaa group server radius rad_pmip
aaa group server radius dummy
aaa authentication login eap_methods group rad_eap
aaa authentication login mac_methods local
aaa authorization exec default local
aaa accounting network acct_methods start-stop group rad_acct
aaa session-id common
no ip source-route
no ip cef
dot11 pause-time 100
dot11 syslog
dot11 ssid GuestNet
  vlan 20
  authentication open
  guest-mode
dot11 ssid Radius-CorpoNet
  vlan 30
  authentication open eap eap_methods
  authentication network-eap eap_methods
  authentication key-management wpa
  mbssid guest-mode
dot11 ssid SecureNet
  vlan 10
  authentication open
  authentication key-management wpa version 2
  mbssid guest-mode
  wpa-psk ascii 7 032752180500024340071C0603
no ipv6 cef
username Cisco password 7 0802455D0A16
bridge irb
interface Dot11Radio0
no ip address
ssid GuestNet
antenna gain 0
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 Dot11Radio0.20
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
no ip address
encryption vlan 10 mode ciphers aes-ccm
encryption vlan 30 mode ciphers aes-ccm tkip
ssid Radius-CorpoNet
ssid SecureNet
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 Dot11Radio1.10
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 Dot11Radio1.30
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 Dot11Radio1.301
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 GigabitEthernet0.301
interface BVI1
mac-address 44d3.ca03.7dce
ip address 192.168.1.245 255.255.255.0
ipv6 address autoconfig
```

ipv6 enable
ip default-gateway 192.168.1.1
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
radius-server attribute 32 include-in-access-req format %h
radius server camera-ThinkStation-P7
address ipv4 192.168.1.114 auth-port 1812 acct-port 1813
key 7 0538232C13697A
bridge 1 route ip
line con 0
line vty 0 4
transport input all
end

Cisco Switch CLI Configuration:

version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname APSwitch
boot-start-marker
boot-end-marker
no aaa new-model
system mtu routing 1500
authentication mac-move permit
ip subnet-zero
spanning-tree mode pvst
spanning-tree etherchannel guard misconfig
spanning-tree extend system-id
vlan internal allocation policy ascending
interface FastEthernet0/1
interface FastEthernet0/2
interface FastEthernet0/3

```
interface FastEthernet0/4
interface FastEthernet0/5
interface FastEthernet0/6
interface FastEthernet0/7
interface FastEthernet0/8
interface FastEthernet0/9
interface FastEthernet0/10
interface FastEthernet0/11
interface FastEthernet0/12
interface FastEthernet0/13
interface FastEthernet0/14
interface FastEthernet0/15
interface FastEthernet0/16
interface FastEthernet0/17
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 1,10,20,30
switchport mode trunk
interface FastEthernet0/18
switchport trunk encapsulation dot1q
switchport mode access
interface FastEthernet0/19
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 1,10,20,30
switchport mode trunk
interface FastEthernet0/20
switchport trunk encapsulation dot1q
switchport mode trunk
interface FastEthernet0/21
interface FastEthernet0/22
interface FastEthernet0/23
interface FastEthernet0/24
interface FastEthernet0/25
interface FastEthernet0/26
interface FastEthernet0/27
interface FastEthernet0/28
interface FastEthernet0/29
interface FastEthernet0/30
interface FastEthernet0/31
```

```
interface FastEthernet0/32
interface FastEthernet0/33
interface FastEthernet0/34
interface FastEthernet0/35
interface FastEthernet0/36
interface FastEthernet0/37
interface FastEthernet0/38
interface FastEthernet0/39
interface FastEthernet0/40
interface FastEthernet0/41
interface FastEthernet0/42
interface FastEthernet0/43
interface FastEthernet0/44
interface FastEthernet0/45
interface FastEthernet0/46
interface FastEthernet0/47
interface FastEthernet0/48
interface GigabitEthernet0/1
interface GigabitEthernet0/2
interface GigabitEthernet0/3
interface GigabitEthernet0/4
interface Vlan1
no ip address
ip classless
ip http server
ip http secure-server
ip sla enable reaction-alerts
line con 0
logging synchronous
line vty 0 4
login
line vty 5 15
login
end
```

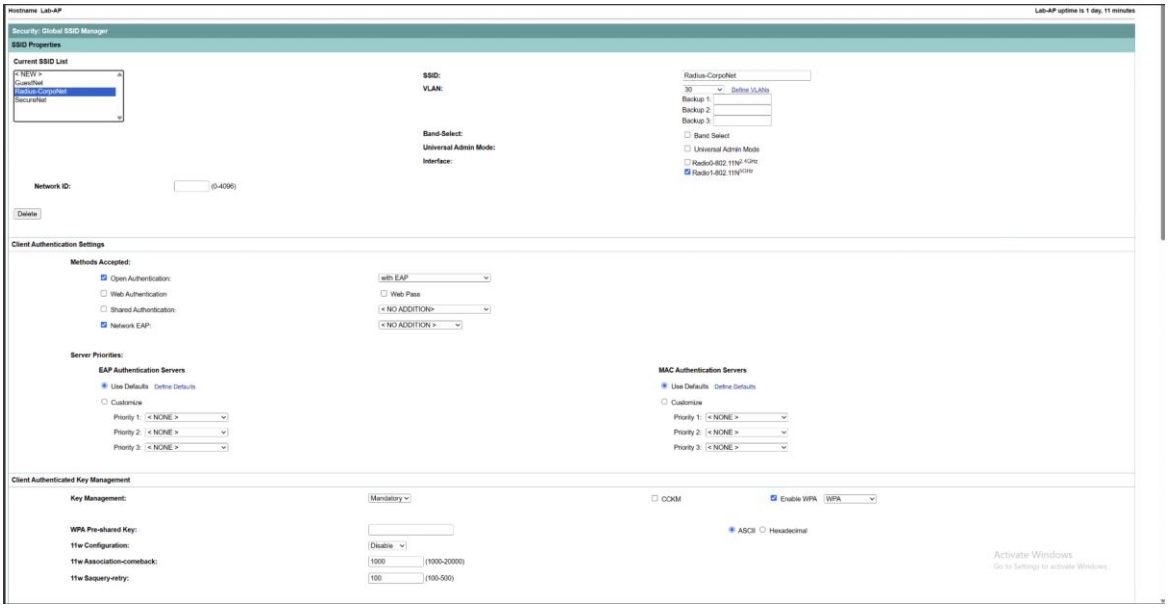
Cisco Router CLI Configuration:

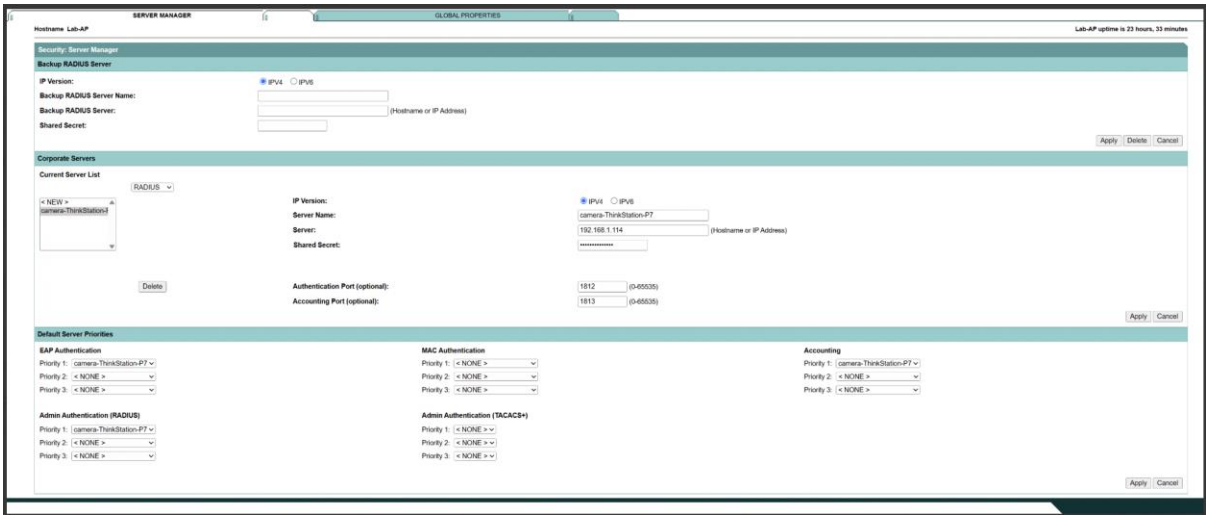
```
version 16.9
 service timestamps debug datetime msec
 service timestamps log datetime msec
 platform qfp utilization monitor load 80
 platform punt-keepalive disable-kernel-core
 hostname AP-Router
 boot-start-marker
 boot-end-marker
 vrf definition Mgmt-intf
 address-family ipv4
 exit-address-family
 address-family ipv6
 exit-address-family
 no aaa new-model
 ip dhcp excluded-address 192.168.10.1 192.168.10.10
 ip dhcp excluded-address 192.168.20.1 192.168.20.10
 ip dhcp excluded-address 192.168.30.1 192.168.30.10
 ip dhcp pool SECURE
 network 192.168.10.0 255.255.255.0
 default-router 192.168.10.1
 dns-server 8.8.8.8
 ip dhcp pool GUEST
 network 192.168.20.0 255.255.255.0
 default-router 192.168.20.1
 dns-server 8.8.8.8
 ip dhcp pool RADIUS
 network 192.168.30.0 255.255.255.0
 default-router 192.168.30.1
 dns-server 8.8.8.8
 login on-success log
 subscriber templating
 vtp domain cisco
 vtp mode transparent
 multilink bundle-name authenticated
 crypto pki trustpoint TP-self-signed-2731081846
 enrollment selfsigned
```

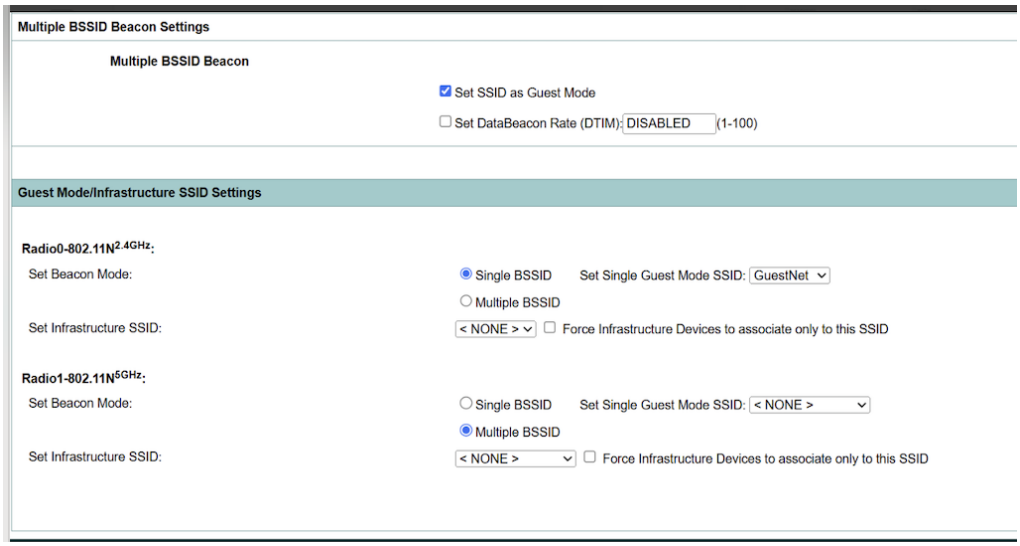
```
subject-name cn=IOS-Self-Signed-Certificate-2731081846
revocation-check none
rsakeypair TP-self-signed-2731081846
license udi pid ISR4321/K9 sn FDO21432ZLS
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
mode none
interface GigabitEthernet0/0/0
ip address 192.168.1.1 255.255.255.0
ip nat inside
negotiation auto
interface GigabitEthernet0/0/0.10
encapsulation dot1Q 10
ip address 192.168.10.1 255.255.255.0
ip nat inside
interface GigabitEthernet0/0/0.20
encapsulation dot1Q 20
ip address 192.168.20.1 255.255.255.0
ip nat inside
interface GigabitEthernet0/0/0.30
encapsulation dot1Q 30
ip address 192.168.30.1 255.255.255.0
ip nat inside
interface GigabitEthernet0/0/1
ip address dhcp
ip nat outside
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
```

```
negotiation auto
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip http client source-interface GigabitEthernet0/0/0
ip nat inside source list 1 interface GigabitEthernet0/0/1 overload
access-list 1 permit 192.168.1.0 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
control-plane
line con 0
logging synchronous
transport input none
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
end
```

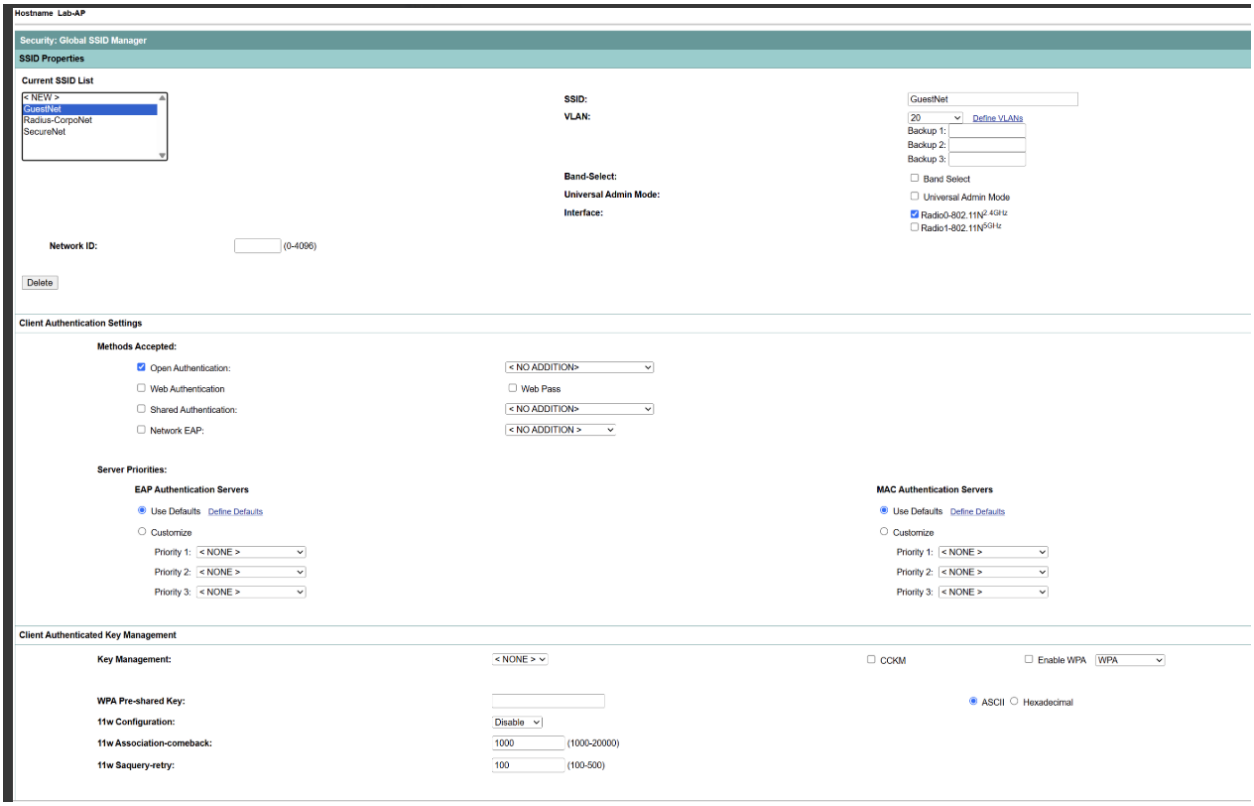
Radius Server on the AP Configuration:


Hostname Lab-AP
Lab-AP uptime is 1 day, 11 minutes
Security: Global SSID Manager
SSID Properties
Current SSID List
< NEW >
GuestNet
Radius-CorpoNet
SecureNet
SSID:
Radius-CorpoNet
VLAN:
30
Define VLANs
Backup 1:
Backup 2:
Backup 3:
Band-Select:
Band Select
Universal Admin Mode:
Universal Admin Mode
Interface:
Radio0-802.11N2.4GHz
Radio1-802.11N5GHz
Network ID:
(0-4096)
Delete
Client Authentication Settings
Methods Accepted:
Open Authentication:
with EAP
Web Authentication
Web Pass
Shared Authentication:
< NO ADDITION>
Network EAP:
< NO ADDITION >
Server Priorities:
EAP Authentication Servers
MAC Authentication Servers
Use Defaults
Define Defaults
Customize
Priority 1: < NONE >
Priority 2: < NONE >
Priority 3: < NONE >
Client Authenticated Key Management
Key Management:
Mandatory
CCKM
Enable WPA
WPA
WPA Pre-shared Key:
ASCII
Hexadecimal
11w Configuration:
Disable
11w Association-comeback:
1000
(1000-20000)
11w Saquery-retry:
100
(100-500)
Activate Windows
Go to Settings to activate Windows.

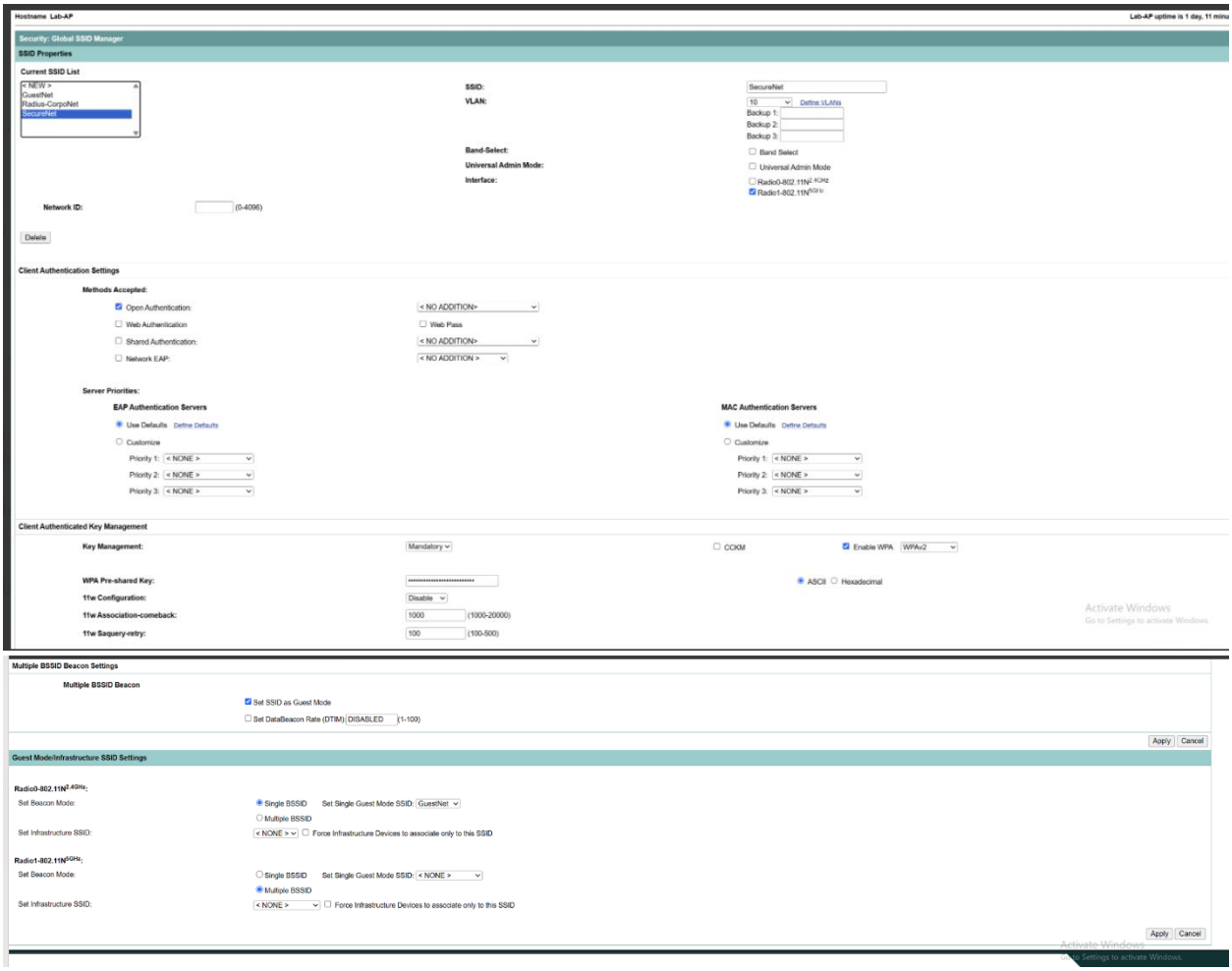

SERVER MANAGER
GLOBAL PROPERTIES
Hostname Lab-AP
Lab-AP uptime is 23 hours, 33 minutes
Security: Server Manager
Backup RADIUS Server
IP Version:
IPV4
IPV6
Backup RADIUS Server Name:
Backup RADIUS Server:
(Hostname or IP Address)
Shared Secret:
Apply
Delete
Cancel
Corporate Servers
Current Server List
RADIUS
< NEW >
camera-ThinkStation-P7
IP Version:
Server Name:
Server:
192.168.1.114
(Hostname or IP Address)
Shared Secret:
Delete
Authentication Port (optional):
1812
(0-65535)
Accounting Port (optional):
1813
(0-65535)
Default Server Priorities
EAP Authentication
MAC Authentication
Accounting
Priority 1: camera-ThinkStation-P7
Priority 2: < NONE >
Priority 3: < NONE >
Admin Authentication (RADIUS)
Admin Authentication (TACACS+)


Multiple BSSID Beacon Settings
Multiple BSSID Beacon
Set SSID as Guest Mode
Set DataBeacon Rate (DTIM):
DISABLED
(1-100)
Guest Mode/Infrastructure SSID Settings
Radio0-802.11N2.4GHz:
Set Beacon Mode:
Single BSSID
Set Single Guest Mode SSID:
GuestNet
Multiple BSSID
Set Infrastructure SSID:
< NONE >
Force Infrastructure Devices to associate only to this SSID
Radio1-802.11N5GHz:
Set Beacon Mode:
Single BSSID
Set Single Guest Mode SSID:
< NONE >
Multiple BSSID
Set Infrastructure SSID:
< NONE >
Force Infrastructure Devices to associate only to this SSID

GuestNet configurations on the AP Configuration:

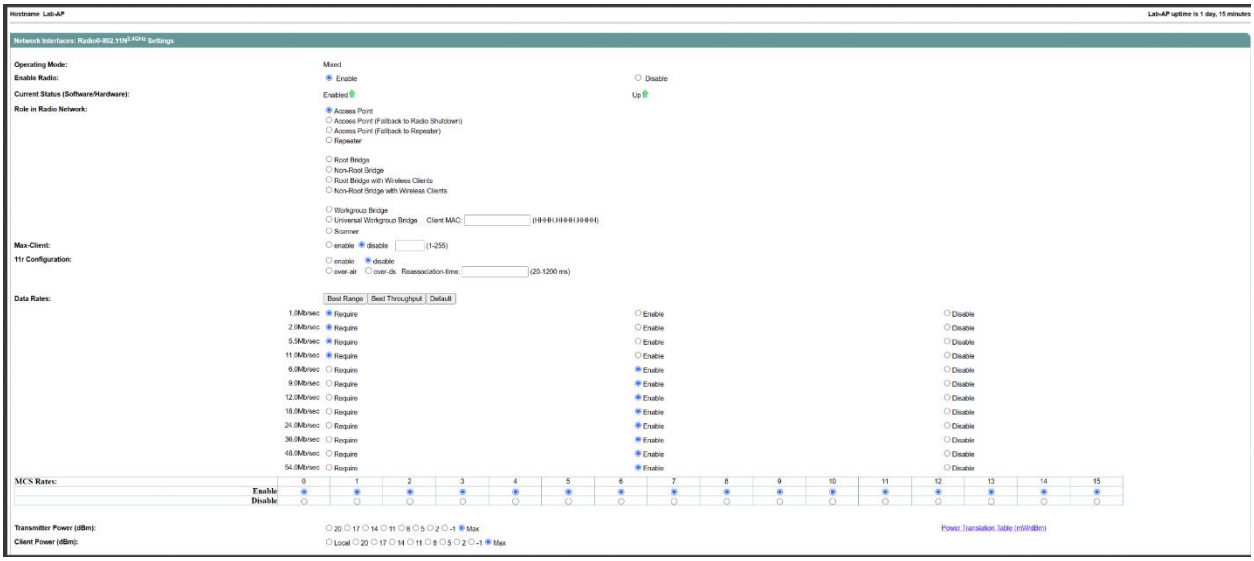


SecureNet configurations on the AP Configuration:

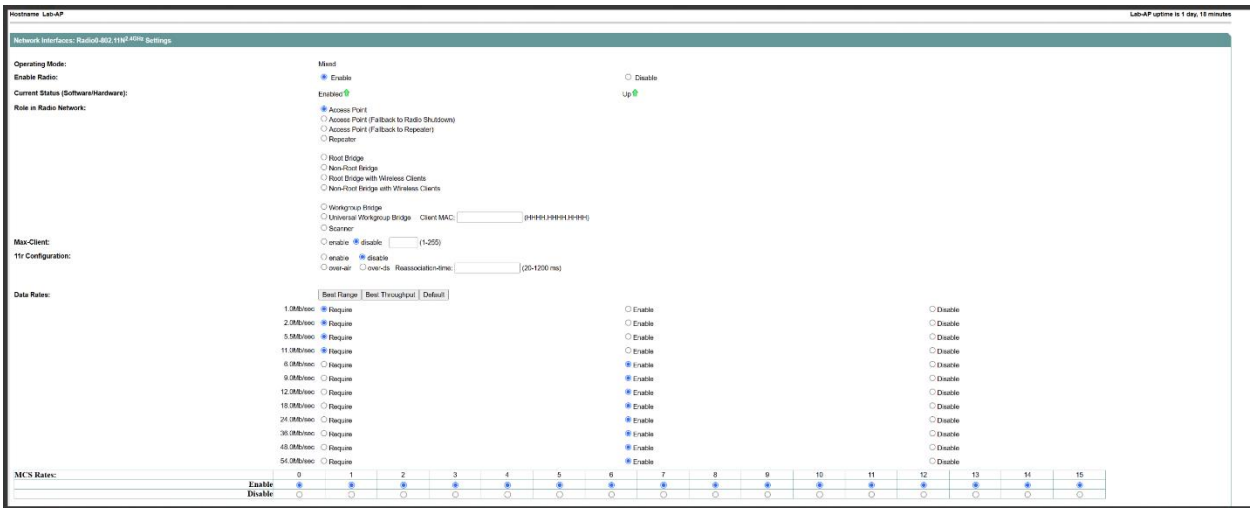

Hostname Lab-AP
Lab-AP uptime is 1 day, 11 minu
Security: Global SSID Manager
SSID Properties
Current SSID List
< NEW >
GuestNet
Radius-CorpoNet
SecureNet
SSID:
SecureNet
VLAN:
10
Define VLANs
Backup 1:
Backup 2:
Backup 3:
Band-Select:
Band Select
Universal Admin Mode:
Universal Admin Mode
Interface:
Radio0-802.11N2.4GHz
Radio1-802.11N5GHz
Network ID:
(0-4096)
Delete
Client Authentication Settings
Methods Accepted:
Open Authentication:
< NO ADDITION>
Web Authentication
Web Pass
Shared Authentication:
< NO ADDITION>
Network EAP:
< NO ADDITION >
Server Priorities:
EAP Authentication Servers
Use Defaults
Define Defaults
Customize
Priority 1: < NONE >
Priority 2: < NONE >
Priority 3: < NONE >
MAC Authentication Servers
Use Defaults
Define Defaults
Customize
Priority 1: < NONE >
Priority 2: < NONE >
Priority 3: < NONE >
Client Authenticated Key Management
Key Management:
Mandatory
CCKM
Enable WPA
WPAv2
WPA Pre-shared Key:
ASCII
Hexadecimal
11w Configuration:
Disable
11w Association-comeback:
1000
(1000-20000)
11w Saquery-retry:
100
(100-500)
Activate Windows
Go to Settings to activate Windows.

Multiple BSSID Beacon Settings
Multiple BSSID Beacon
Set SSID as Guest Mode
Set DataBeacon Rate (DTIM):
DISABLED
(1-100)
Apply
Cancel
Guest Mode/Infrastructure SSID Settings
Radio0-802.11N2.4GHz:
Set Beacon Mode:
Single BSSID
Set Single Guest Mode SSID:
GuestNet
Multiple BSSID
Set Infrastructure SSID:
< NONE >
Force Infrastructure Devices to associate only to this SSID
Radio1-802.11N5GHz:
Set Beacon Mode:
Single BSSID
Set Single Guest Mode SSID:
< NONE >
Multiple BSSID
Set Infrastructure SSID:
< NONE >
Force Infrastructure Devices to associate only to this SSID
Apply
Cancel
Activate Windows
Go to Settings to activate Windows.

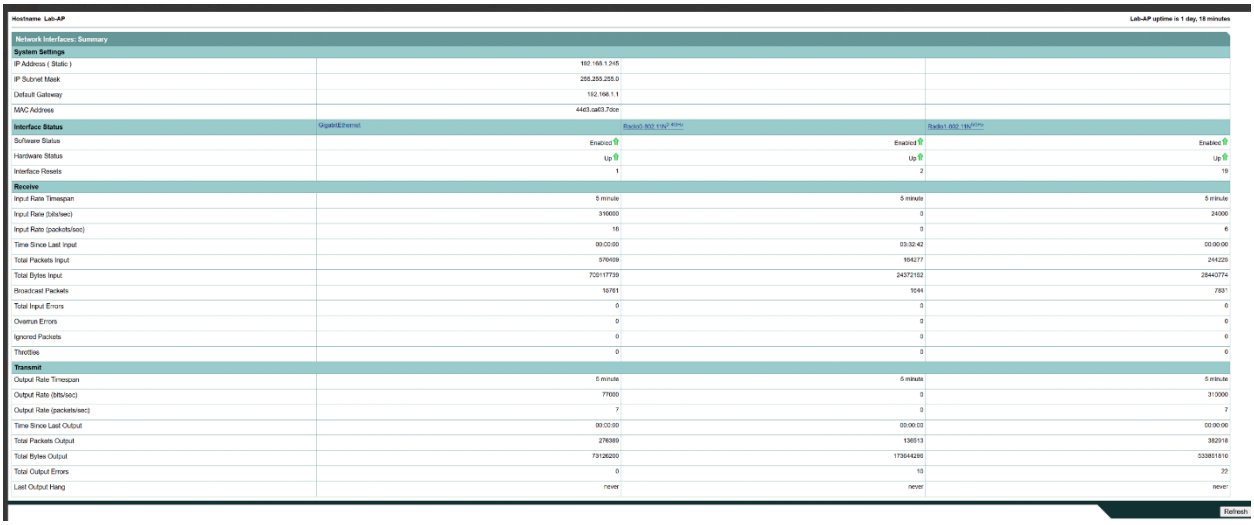
Radio0configurations on the AP Configuration:



Radoip1-5GHz Configurations:



AP Summary:

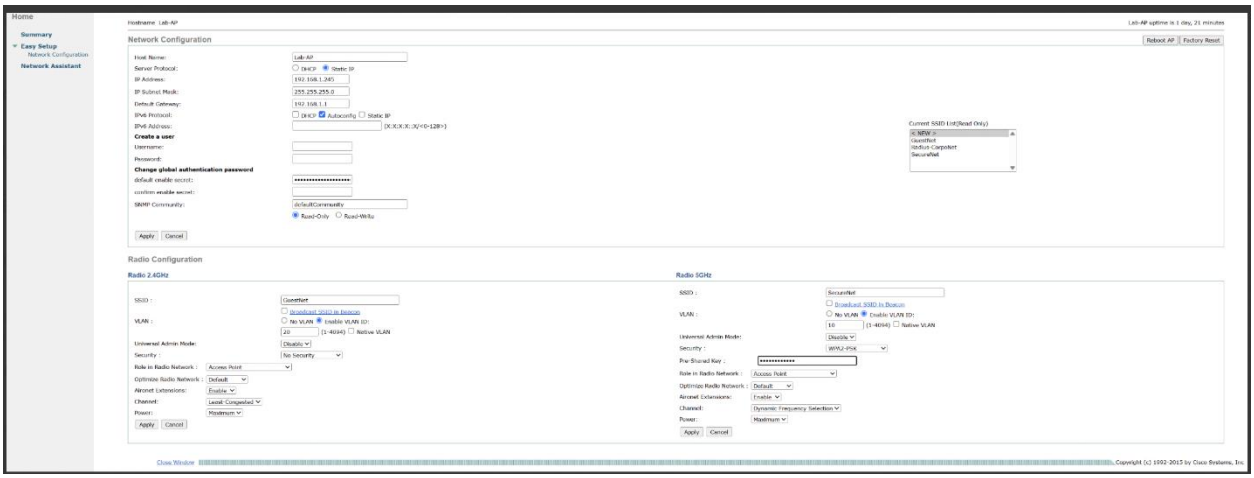


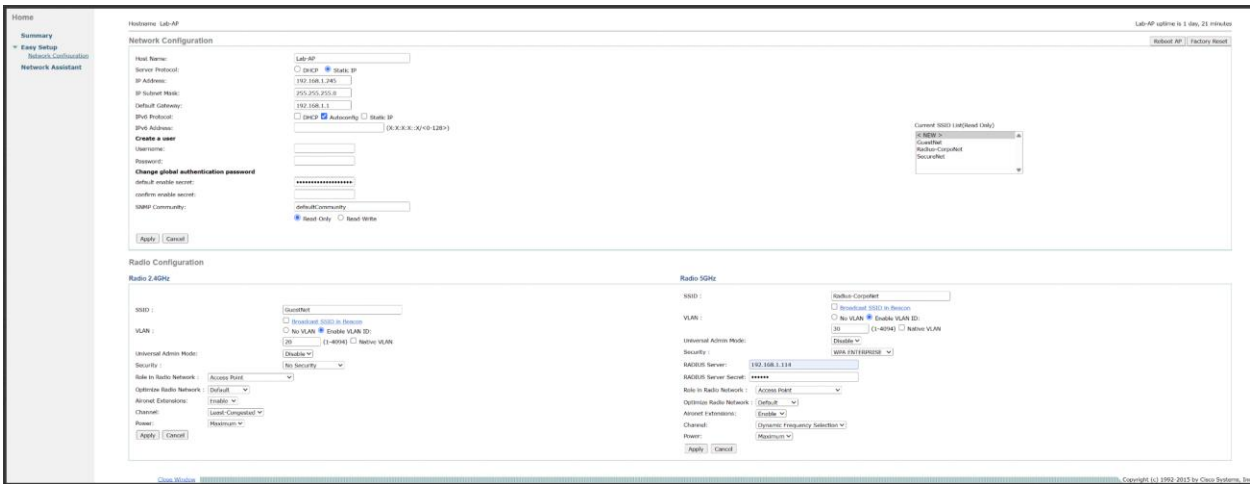
Hostname Lab-AP — Lab-AP uptime is 1 day, 18 minutes

Network Interfaces: Summary			
System Settings			
IP Address (Static)	192.168.1.245		
IP Subnet Mask	255.255.255.0		
Default Gateway	192.168.1.1		
MAC Address	4463.ca63.7dce		
Interface Status	GigabitEthernet	Radio0-802.11N^2.4GHz	Radio1-802.11N^5GHz
Software Status	Enabled	Enabled	Enabled
Hardware Status	Up	Up	Up
Interface Resets	1	2	19
Receive			
Input Rate Timespan	5 minute	5 minute	5 minute
Input Rate (bits/sec)	310000	0	24000
Input Rate (packets/sec)	16	0	6
Time Since Last Input	00:00:00	03:32:42	00:00:00
Total Packets Input	570499	104277	244225
Total Bytes Input	709117739	24372102	28443774
Broadcast Packets	16761	1044	7831
Total Input Errors	0	0	0
Overrun Errors	0	0	0
Ignored Packets	0	0	0
Throttles	0	0	0
Transmit			
Output Rate Timespan	5 minute	5 minute	5 minute
Output Rate (bits/sec)	77000	0	31000
Output Rate (packets/sec)	7	0	7
Time Since Last Output	00:00:00	00:00:03	00:00:00
Total Packets Output	276389	136513	382918
Total Bytes Output	73126200	173644286	533851810
Total Output Errors	0	10	22
Last Output Hang	never	never	never

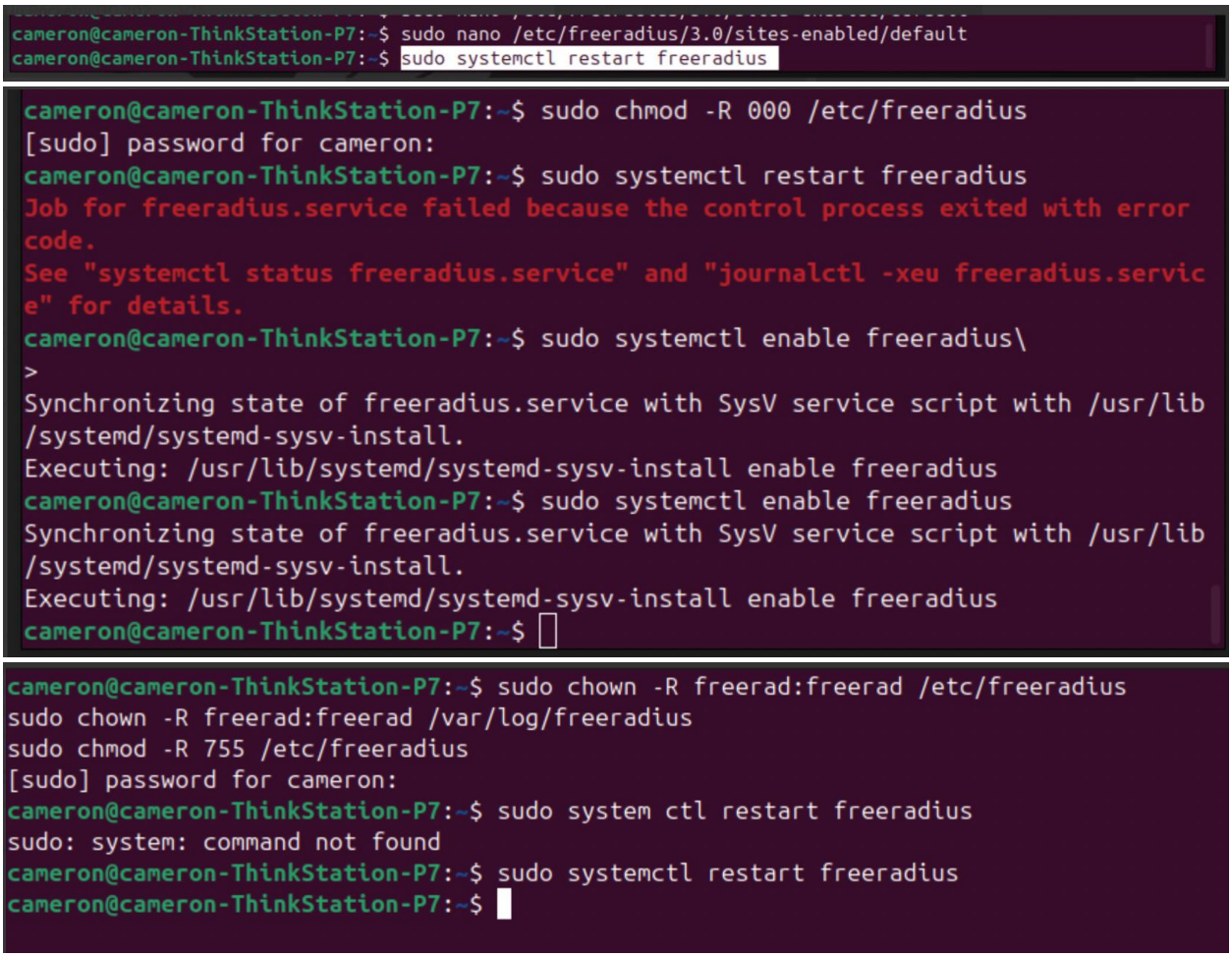
Refresh

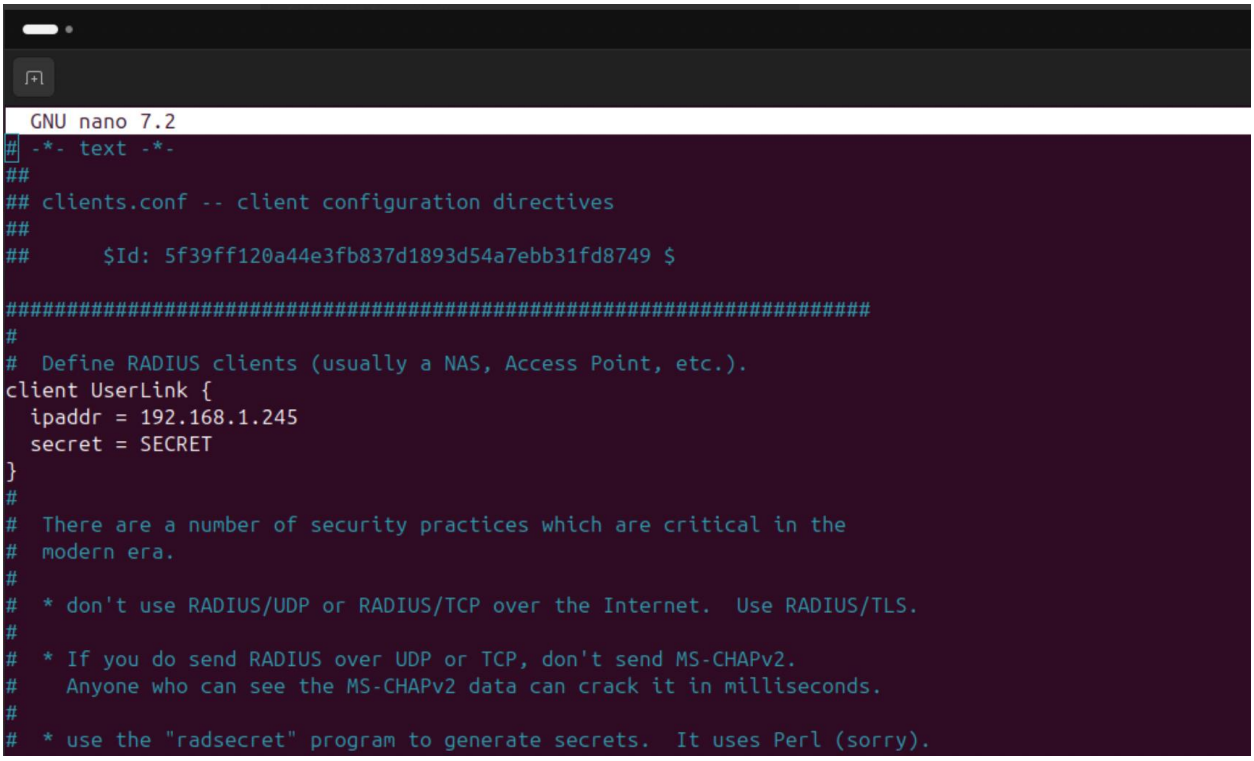
SSIDS:



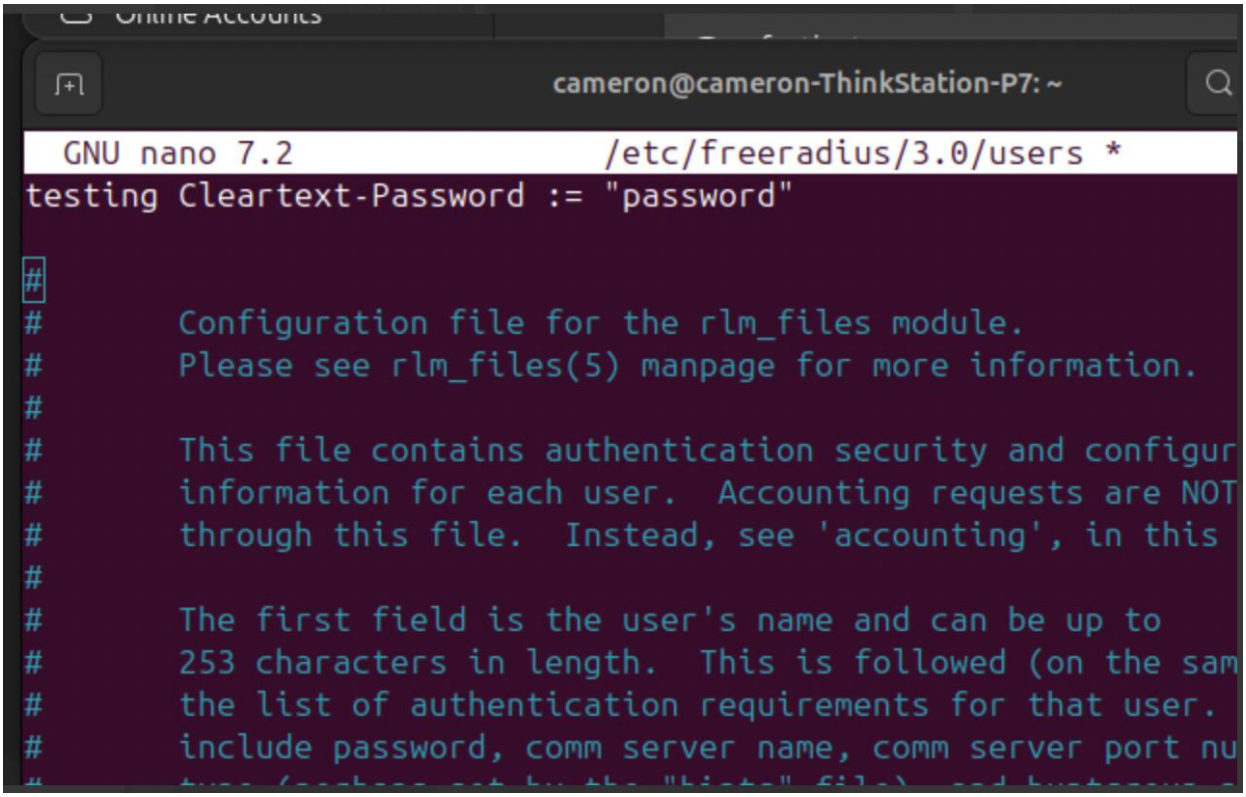


Radius Server commands:

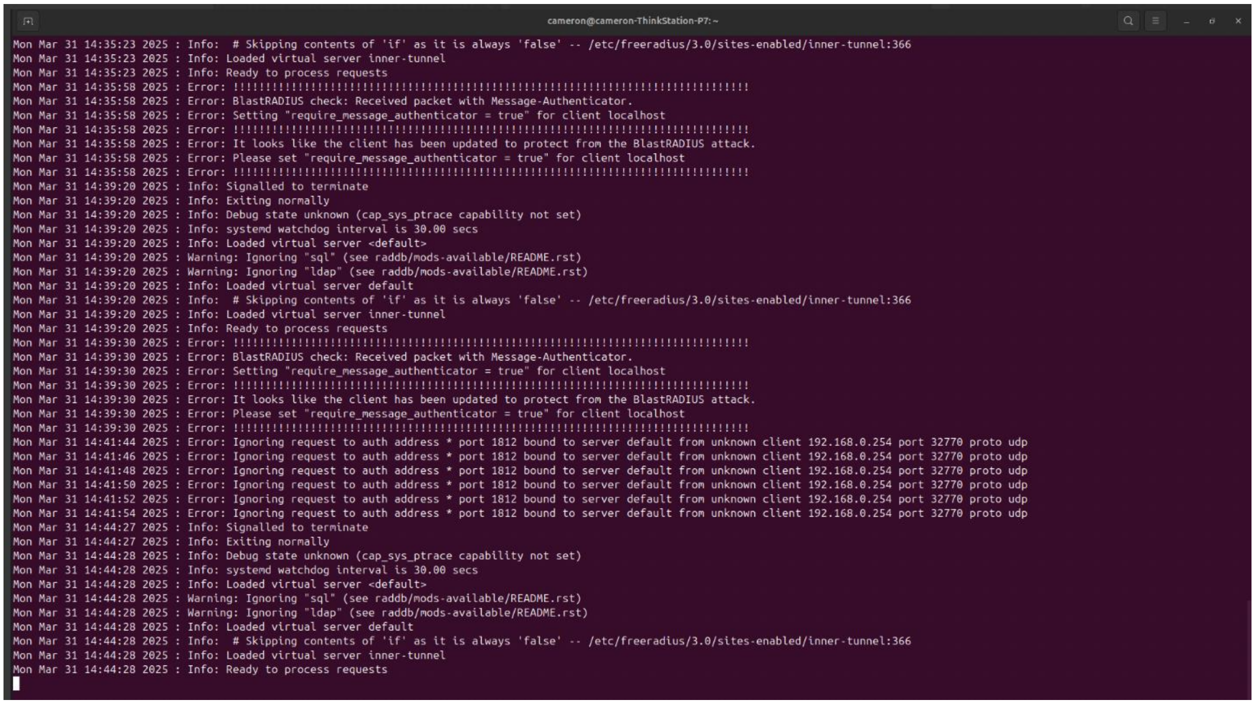


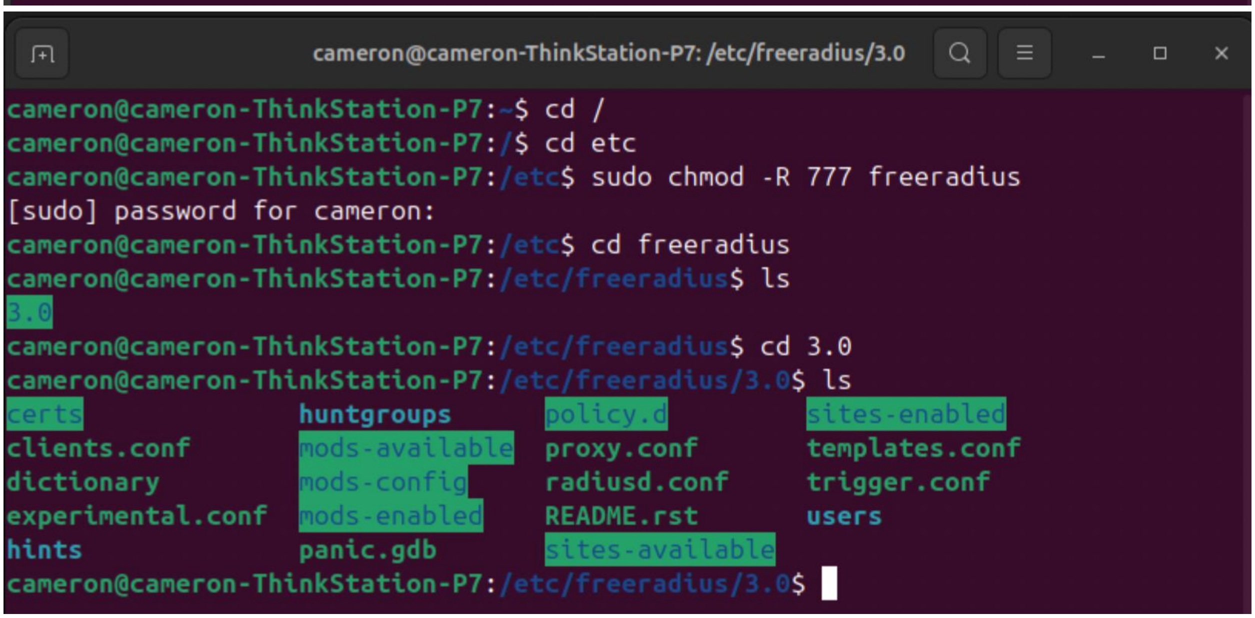


```
  GNU nano 7.2
# -*- text -*-
##
## clients.conf -- client configuration directives
##
##      $Id: 5f39ff120a44e3fb837d1893d54a7ebb31fd8749 $

#######################################################################
#
#  Define RADIUS clients (usually a NAS, Access Point, etc.).
client UserLink {
  ipaddr = 192.168.1.245
  secret = SECRET
}
#
#  There are a number of security practices which are critical in the
#  modern era.
#
#  * don't use RADIUS/UDP or RADIUS/TCP over the Internet.  Use RADIUS/TLS.
#
#  * If you do send RADIUS over UDP or TCP, don't send MS-CHAPv2.
#    Anyone who can see the MS-CHAPv2 data can crack it in milliseconds.
#
#  * use the "radsecret" program to generate secrets.  It uses Perl (sorry).
```



```
cameron@cameron-ThinkStation-P7: ~
  GNU nano 7.2                  /etc/freeradius/3.0/users *
testing Cleartext-Password := "password"

#
#       Configuration file for the rlm_files module.
#       Please see rlm_files(5) manpage for more information.
#
#       This file contains authentication security and configur
#       information for each user.  Accounting requests are NOT
#       through this file.  Instead, see 'accounting', in this
#
#       The first field is the user's name and can be up to
#       253 characters in length.  This is followed (on the sam
#       the list of authentication requirements for that user.
#       include password, comm server name, comm server port nu
```


cameron@cameron-ThinkStation-P7: ~
Mon Mar 31 14:35:23 2025 : Info: # Skipping contents of 'if' as it is always 'false' -- /etc/freeradius/3.0/sites-enabled/inner-tunnel:366
Mon Mar 31 14:35:23 2025 : Info: Loaded virtual server inner-tunnel
Mon Mar 31 14:35:23 2025 : Info: Ready to process requests
Mon Mar 31 14:35:58 2025 : Error: !!
Mon Mar 31 14:35:58 2025 : Error: BlastRADIUS check: Received packet with Message-Authenticator.
Mon Mar 31 14:35:58 2025 : Error: Setting "require_message_authenticator = true" for client localhost
Mon Mar 31 14:35:58 2025 : Error: !!
Mon Mar 31 14:35:58 2025 : Error: It looks like the client has been updated to protect from the BlastRADIUS attack.
Mon Mar 31 14:35:58 2025 : Error: Please set "require_message_authenticator = true" for client localhost
Mon Mar 31 14:35:58 2025 : Error: !!
Mon Mar 31 14:39:20 2025 : Info: Signalled to terminate
Mon Mar 31 14:39:20 2025 : Info: Exiting normally
Mon Mar 31 14:39:20 2025 : Info: Debug state unknown (cap_sys_ptrace capability not set)
Mon Mar 31 14:39:20 2025 : Info: systemd watchdog interval is 30.00 secs
Mon Mar 31 14:39:20 2025 : Info: Loaded virtual server <default>
Mon Mar 31 14:39:20 2025 : Warning: Ignoring "sql" (see raddb/mods-available/README.rst)
Mon Mar 31 14:39:20 2025 : Warning: Ignoring "ldap" (see raddb/mods-available/README.rst)
Mon Mar 31 14:39:20 2025 : Info: Loaded virtual server default
Mon Mar 31 14:39:20 2025 : Info: # Skipping contents of 'if' as it is always 'false' -- /etc/freeradius/3.0/sites-enabled/inner-tunnel:366
Mon Mar 31 14:39:20 2025 : Info: Loaded virtual server inner-tunnel
Mon Mar 31 14:39:20 2025 : Info: Ready to process requests
Mon Mar 31 14:39:30 2025 : Error: !!
Mon Mar 31 14:39:30 2025 : Error: BlastRADIUS check: Received packet with Message-Authenticator.
Mon Mar 31 14:39:30 2025 : Error: Setting "require_message_authenticator = true" for client localhost
Mon Mar 31 14:39:30 2025 : Error: !!
Mon Mar 31 14:39:30 2025 : Error: It looks like the client has been updated to protect from the BlastRADIUS attack.
Mon Mar 31 14:39:30 2025 : Error: Please set "require_message_authenticator = true" for client localhost
Mon Mar 31 14:39:30 2025 : Error: !!
Mon Mar 31 14:41:44 2025 : Error: Ignoring request to auth address * port 1812 bound to server default from unknown client 192.168.0.254 port 32770 proto udp
Mon Mar 31 14:41:46 2025 : Error: Ignoring request to auth address * port 1812 bound to server default from unknown client 192.168.0.254 port 32770 proto udp
Mon Mar 31 14:41:48 2025 : Error: Ignoring request to auth address * port 1812 bound to server default from unknown client 192.168.0.254 port 32770 proto udp
Mon Mar 31 14:41:50 2025 : Error: Ignoring request to auth address * port 1812 bound to server default from unknown client 192.168.0.254 port 32770 proto udp
Mon Mar 31 14:41:52 2025 : Error: Ignoring request to auth address * port 1812 bound to server default from unknown client 192.168.0.254 port 32770 proto udp
Mon Mar 31 14:41:54 2025 : Error: Ignoring request to auth address * port 1812 bound to server default from unknown client 192.168.0.254 port 32770 proto udp
Mon Mar 31 14:44:27 2025 : Info: Signalled to terminate
Mon Mar 31 14:44:27 2025 : Info: Exiting normally
Mon Mar 31 14:44:28 2025 : Info: Debug state unknown (cap_sys_ptrace capability not set)
Mon Mar 31 14:44:28 2025 : Info: systemd watchdog interval is 30.00 secs
Mon Mar 31 14:44:28 2025 : Info: Loaded virtual server <default>
Mon Mar 31 14:44:28 2025 : Warning: Ignoring "sql" (see raddb/mods-available/README.rst)
Mon Mar 31 14:44:28 2025 : Warning: Ignoring "ldap" (see raddb/mods-available/README.rst)
Mon Mar 31 14:44:28 2025 : Info: Loaded virtual server default
Mon Mar 31 14:44:28 2025 : Info: # Skipping contents of 'if' as it is always 'false' -- /etc/freeradius/3.0/sites-enabled/inner-tunnel:366
Mon Mar 31 14:44:28 2025 : Info: Loaded virtual server inner-tunnel
Mon Mar 31 14:44:28 2025 : Info: Ready to process requests


cameron@cameron-ThinkStation-P7: /etc/freeradius/3.0
cameron@cameron-ThinkStation-P7:~$ cd /
cameron@cameron-ThinkStation-P7:/$ cd etc
cameron@cameron-ThinkStation-P7:/etc$ sudo chmod -R 777 freeradius
[sudo] password for cameron:
cameron@cameron-ThinkStation-P7:/etc$ cd freeradius
cameron@cameron-ThinkStation-P7:/etc/freeradius$ ls
3.0
cameron@cameron-ThinkStation-P7:/etc/freeradius$ cd 3.0
cameron@cameron-ThinkStation-P7:/etc/freeradius/3.0$ ls
certs huntgroups policy.d sites-enabled
clients.conf mods-available proxy.conf templates.conf
dictionary mods-config radiusd.conf trigger.conf
experimental.conf mods-enabled README.rst users
hints panic.gdb sites-available
cameron@cameron-ThinkStation-P7:/etc/freeradius/3.0$

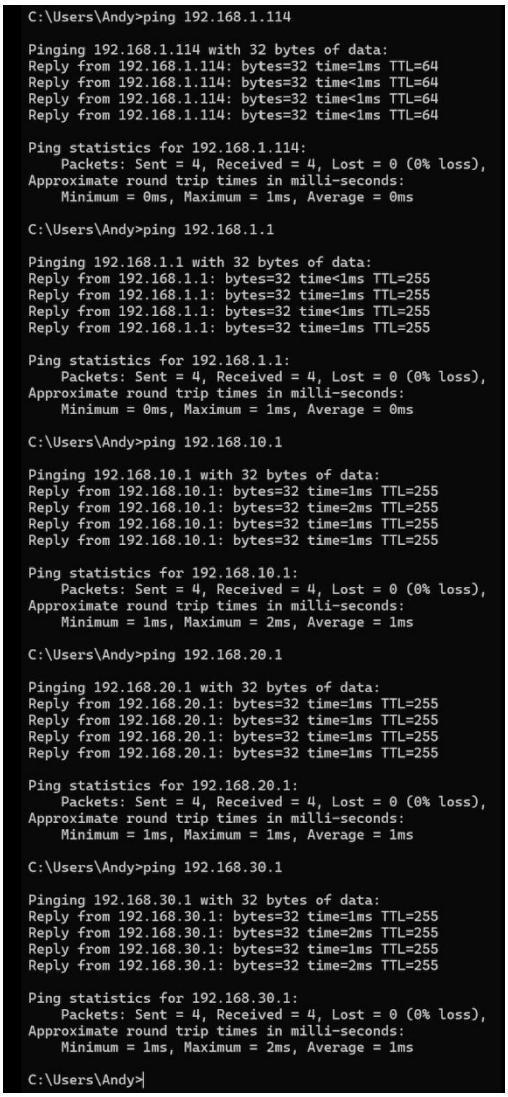
Show IP route/Traceroute for IPv4 and IPv6:

IPv4:

```
AP-Router#sh ip route
 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 192.168.40.1 to network 0.0.0.0

S*    0.0.0.0/0 [254/0] via 192.168.40.1
      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
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
 C       192.168.10.0/24 is directly connected, GigabitEthernet0/0/0.10
 L       192.168.10.1/32 is directly connected, GigabitEthernet0/0/0.10
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
 C       192.168.20.0/24 is directly connected, GigabitEthernet0/0/0.20
 L       192.168.20.1/32 is directly connected, GigabitEthernet0/0/0.20
      192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
 C       192.168.30.0/24 is directly connected, GigabitEthernet0/0/0.30
 L       192.168.30.1/32 is directly connected, GigabitEthernet0/0/0.30
 C    192.168.40.0/23 is directly connected, GigabitEthernet0/0/1
      192.168.40.0/32 is subnetted, 1 subnets
```

L 192.168.40.131 is directly connected, GigabitEthernet0/0/1

Traceroute/Pings:



```
C:\Users\Andy>tracert 192.168.1.245

Tracing route to 192.168.1.245 over a maximum of 30 hops

  1    <1 ms    <1 ms    <1 ms  192.168.1.245

Trace complete.

C:\Users\Andy>
```

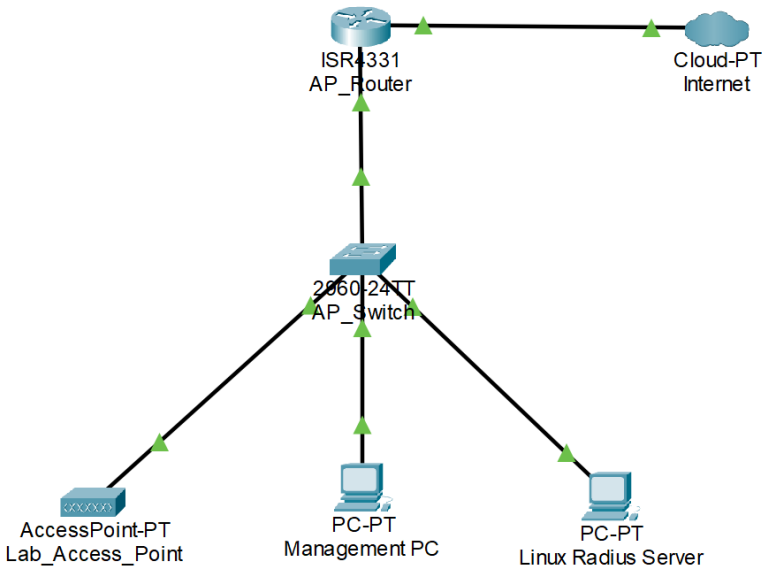
```
C:\Users\Andy>tracert 192.168.1.114

Tracing route to 192.168.1.114 over a maximum of 30 hops

  1    <1 ms    <1 ms    <1 ms  192.168.1.114

Trace complete.
```

```
C:\Users\Andy>tracert 192.168.1.1

Tracing route to 192.168.1.1 over a maximum of 30 hops

  1    <1 ms    <1 ms    <1 ms  192.168.1.1

Trace complete.
```

PC pinging AP:

```
C:\Users\Andy>ping 192.168.1.245

Pinging 192.168.1.245 with 32 bytes of data:
Reply from 192.168.1.245: bytes=32 time<1ms TTL=255
Reply from 192.168.1.245: bytes=32 time<1ms TTL=255
Reply from 192.168.1.245: bytes=32 time=1ms TTL=255
Reply from 192.168.1.245: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.1.245:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Lab Diagram



Problems

What Was Challenging:

We faced some difficulties during the lab, especially in setting up basic Layer 2 connectivity. Despite having assigned IP addresses on both ends of a direct connection the pings between devices failed which led us to change from the initial controller-based access point setup and instead use a standalone access point. But, switching brought its own challenges which were mainly due to the different GUI used by the controller and AP. Learning how to configure each and understand them became a little hard to do.

One of the more demanding parts was configuring the Linux-based RADIUS server, which was new to most my group but familiar to me. Without much familiarity and experience, we had to troubleshoot quite often with us changing things until we reached a working setup. On top of that, many of the CLI and GUI commands used with the access point were new, adding some complexity. The change from controller to standalone AP did require significant change to our setup and understanding.

Concepts We Learned:

This lab introduced us to using the GUI of a wireless access point to configure and manage multiple SSIDs, each with a distinct network name for identification. We also revisited DHCP configuration on the router, enabling dynamic IP address assignment to clients connected through different VLANs, each linked to an SSID. Another key takeaway was our implementation of NAT, allowing multiple devices to share a single public IP address while accessing the internet. A major new concept we encountered was the setup of a Linux-based RADIUS server for centralized authentication. This was our first experience working with such a system and gave us a better understanding of enterprise-grade security practices for wireless networking.

What Caused Issues:

Initially, we attempted to incorporate a controller into the lab design, but it failed to communicate properly with the access point. This communication problem led us to transition toward a standalone access point configuration, which we assumed would be simpler. However, this change introduced new complications, especially with the different configuration styles. We also struggled with integrating inter-VLAN routing with the “router-on-a-stick” setup. Both the Wireless LAN Controller GUI and the access point’s web interface required some time to understand, as we had to navigate and configure them at the same time while also taking care of network flaws.

Conclusion

In this lab, we built a wireless network composed of three different SSIDs each operating from the same access point, and with each associated with a different VLAN and authentication method. The first SSID is an open and unsecured network, the second used WPA2-PSK securing access with a shared passphrase, and the third using WPA2-Enterprise while also requiring people to have credentials which were managed by a RADIUS server. The router was configured as a DHCP server to automatically assign IP addresses to devices on each VLAN, ensuring network segmentation and connectivity across all networks.

6/1/2025

CCNP Portfolio
Layer 2 Attacks

Blizzard, Harrison J

Purpose

The purpose of this lab is to build a network that can defend itself against three different Layer 2 attacks coming from a singular Linux PC. The specific attacks we used in this lab were DHCP starvation, ARP spoofing, and MAC flooding. To defend against DHCP starvation, we used DHCP Snooping. For protection against ARP spoofing, we used Dynamic ARP Inspection (DAI), and for MAC flooding attacks we used port security on active switch ports.

Before we secured the network, we carried out the attacks: ARP spoofing, DHCP starvation, and MAC flooding using a Linux PC to target our network switch. This part of the lab was to give us a better understanding of how Layer 2 vulnerabilities are used by attackers. Getting experience with both sides of the attacks and defenses was key to understand how to apply this knowledge to secure small networks and prepare for protecting larger enterprise networks in professional settings.

Background Information/Lab Concepts

A DHCP Starvation attack is a type of network-based threat where an attacker floods the DHCP server with numerous requests using spoofed MAC addresses, ultimately depleting the available IP address pool. This prevents legitimate clients from obtaining IP addresses, effectively denying network access.

ARP Spoofing involves a malicious actor sending fake ARP messages across a local network. The result is that the attacker's MAC address gets falsely associated with the IP address of a legitimate host, enabling interception or manipulation of data meant for that device.

MAC Flooding targets the memory limitations of a switch's MAC address table. By overwhelming it with a high volume of bogus or random MAC addresses, the attacker forces the switch to enter a fail-open state, broadcasting incoming frames to all ports and compromising the confidentiality of data traffic.

Dynamic ARP Inspection (DAI) is a protective mechanism designed to counter ARP spoofing attempts. It does this by inspecting ARP packets on untrusted ports and verifying that the MAC-to-IP bindings match those stored in either the DHCP Snooping database or configured ARP access control lists.

Port Security is a command used on cisco switches which strengthen network defense. It limits the devices allowed to connect to the switchport by filtering based on MAC addresses.

DHCP Snooping is a cisco switch command used to protect against malicious actors and spoofing attacks. It monitors and filters DHCP messages and building a trusted database of client MAC and IP address bindings. It then discards any DHCP traffic that originates from unauthorized sources.

Lab Summary

PC 1: Connects to Switch through F1/0/1 with an ipv4 address of (192.168.10.6)

Switch 1: Connects to Router through F1/0/2 (originally not configured with any defenses to make sure they work)

Router 1: Connects to Switch through Interface G0/0/0 with an ipv4 address of (192.168.10.1)

Linux PC: Connects to Switch through F1/0/3 with ipv4 address of (192.168.10.7)

PC 2: Connects to Switch through F1/0/4 with an ipv4 address of (192.168.10.8)

PC to switch, switch to router, Linux PC to switch. Linux PC used to run the attacks on the switch which we then try and configure the switch to mitigate and deny these attacks. PC 2 is used on MAC flooding to capture the packets as the Linux PC would crash after capturing its own mac flooding attack in Wireshark.

Lab Commands

show ip route: Displays the IPv4 routing table

ip routing: A command to enable the sending of data packets for IPv4 across an IP network

traceroute [destination ip address]: Figuring the data hops from the source ip address to the destination ip address.

show ip protocol: displays information about routing protocols on the device

ip arp inspection vlan [vlan_number]: used to enable Dynamic ARP Inspection (DAI) on specific VLANs within a network

ip arp inspection trust: used to configure an interface trusted for Dynamic ARP Inspection (DAI) on a Switch.

ip dhcp snooping: to enable DHCP snooping globally on a Cisco Switch

ip dhcp snooping vlan: enables DHCP snooping on specific VLANs on a switch

Ip dhcp snooping trust: designates a switch port as a trusted interface for DHCP traffic

ip dhcp snooping limit rate [number]: sets the number of dhcp requests that can be revived in a second.

sudo macof -i <interface>: enables mac flooding on interface

switchport mode access: configures a switchport to operate in access mode

Switchport port-security: a security feature on network switches that restricts network access to authorized devices by limiting the number of MAC addresses allowed on a port.

Lab Configs:

R1:

```
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
hostname Router
boot-start-marker
boot-end-marker
vrf definition Mgmt-intf
address-family ipv4
exit-address-family
address-family ipv6
exit-address-family
no aaa new-model
ip dhcp excluded-address 192.168.10.1
ip dhcp pool LAYER2ATTACK
network 192.168.10.0 255.255.255.0
default-router 192.168.10.1
dns-server 8.8.8.8
login on-success log
subscriber templating
```

```
multilink bundle-name authenticated
license udi pid ISR4321/K9 sn FLM2408005M
no license smart enable
diagnostic bootup level minimal
spanning-tree extend system-id
redundancy
mode none
interface GigabitEthernet0/0/0
ip address 192.168.10.1 255.255.255.0
negotiation auto
interface GigabitEthernet0/0/1
no ip address
negotiation auto
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
ip forward-protocol nd
no ip http server
ip http secure-server
ip tftp source-interface GigabitEthernet0
control-plane
line con 0
transport input none
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
end
```

S1:

```
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption

hostname Switch

boot-start-marker
boot-end-marker
```

```
no aaa new-model
system mtu routing 1500
ip arp gratuitous none

ip dhcp snooping

spanning-tree mode pvst
spanning-tree portfast default
spanning-tree extend system-id

vlan internal allocation policy ascending

interface Vlan1
no ip address
shutdown

ip http server
ip http secure-server
logging esm config
line con 0
line vty 5 15
end
```

PC1:

```
  Description . . . . . . . . . . . : Intel(R) Ethernet Connection
(17) I219-LM
    Physical Address. . . . . . . . . : 04-D9-C8-BA-24-6E
    DHCP Enabled. . . . . . . . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IPv4 Address. . . . . . . . . . . : 192.168.10.6(Preferred)
    Subnet Mask . . . . . . . . . . . : 255.255.255.0
    Lease Obtained. . . . . . . . . . : Monday, June 9, 2025 10:02:17
AM
    Lease Expires . . . . . . . . . . : Tuesday, June 10, 2025
11:03:25 AM
    Default Gateway . . . . . . . . . : 192.168.10.1
    DHCP Server . . . . . . . . . . . : 192.168.10.1
    DNS Servers . . . . . . . . . . . : 8.8.8.8
    NetBIOS over Tcpip. . . . . . . . : Enabled
```

Show IP route/Traceroute for IPv4,

R1:

```
Router#show ip route
 Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
        ia - IS-IS inter area, * - candidate default, U - per-user
static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l -
LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides
from PfR



Gateway of last resort is not set



      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
 C        192.168.10.0/24 is directly connected, GigabitEthernet0/0/0
 L        192.168.10.1/32 is directly connected, GigabitEthernet0/0/0
```

Show IP protocol:

```
Router#show ip protocol
 *** IP Routing is NSF aware ***



Routing Protocol is "application"
   Sending updates every 0 seconds
   Invalid after 0 seconds, hold down 0, flushed after 0
   Outgoing update filter list for all interfaces is not set
   Incoming update filter list for all interfaces is not set
   Maximum path: 32
   Routing for Networks:
   Routing Information Sources:
     Gateway         Distance      Last Update
   Distance: (default is 4)
```

MAC Flooding:

```
    Mac Address Table
 -------------------------------------------


Vlan    Mac Address       Type        Ports
 ----    -----------       --------    -----
 All    0100.0ccc.cccc    STATIC      CPU
 All    0100.0ccc.cccd    STATIC      CPU
 All    0100.0ccc.ccce    STATIC      CPU
 All    0180.c200.0000    STATIC      CPU
 All    0180.c200.0001    STATIC      CPU
 All    0180.c200.0002    STATIC      CPU
 All    0180.c200.0003    STATIC      CPU
 All    0180.c200.0004    STATIC      CPU
 All    0180.c200.0005    STATIC      CPU
 All    0180.c200.0006    STATIC      CPU
 All    0180.c200.0007    STATIC      CPU
 All    0180.c200.0008    STATIC      CPU
 All    0180.c200.0009    STATIC      CPU
 All    0180.c200.000a    STATIC      CPU
 All    0180.c200.000b    STATIC      CPU
 All    0180.c200.000c    STATIC      CPU
 All    0180.c200.000d    STATIC      CPU
 All    0180.c200.000e    STATIC      CPU
 All    0180.c200.000f    STATIC      CPU
 All    0180.c200.0010    STATIC      CPU
 Total Mac Addresses for this criterion: 20
```

DHCP Starvation:

```
Router# show ip arp
 Protocol  Address          Age (min)  Hardware Addr   Type
Interface
 Internet  192.168.10.1           -   cc7f.76d1.adc0  ARPA
GigabitEthernet0/0/0
 Internet  192.168.10.6           0   04d9.c8ba.246e  ARPA
GigabitEthernet0/0/0
 Internet  192.168.10.7           2   482a.e38e.e9c8  ARPA
GigabitEthernet0/0/0
 Internet  192.168.10.230         0   Incomplete      ARPA
 Internet  192.168.10.231         0   Incomplete      ARPA
 Internet  192.168.10.232         0   Incomplete      ARPA
 Internet  192.168.10.233         0   Incomplete      ARPA
 Internet  192.168.10.234         0   Incomplete      ARPA
 Internet  192.168.10.235         0   Incomplete      ARPA
 Internet  192.168.10.236         0   Incomplete      ARPA
 Internet  192.168.10.237         0   Incomplete      ARPA
 Internet  192.168.10.238         0   Incomplete      ARPA
 Internet  192.168.10.239         0   Incomplete      ARPA
 Internet  192.168.10.240         0   Incomplete      ARPA
 Internet  192.168.10.241         0   Incomplete      ARPA
 Internet  192.168.10.242         0   Incomplete      ARPA
 Internet  192.168.10.243         0   Incomplete      ARPA
 Internet  192.168.10.244         0   Incomplete      ARPA
```

ARP Spoofing:

```
Router# show ip arp
 Protocol  Address          Age (min)  Hardware Addr   Type
Interface
 Internet  192.168.10.1           -   cc7f.76d1.adc0  ARPA
GigabitEthernet0/0/0
 Internet  192.168.10.6           0   482a.e38e.e9c8  ARPA
GigabitEthernet0/0/0
 Internet  192.168.10.7           5   482a.e38e.e9c8  ARPA
GigabitEthernet0/0/0
```

Traceroute:

Router to PC

Router#ping 192.168.10.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#

PC to Router ping

C:\Users\Andy> ping 192.168.10.1

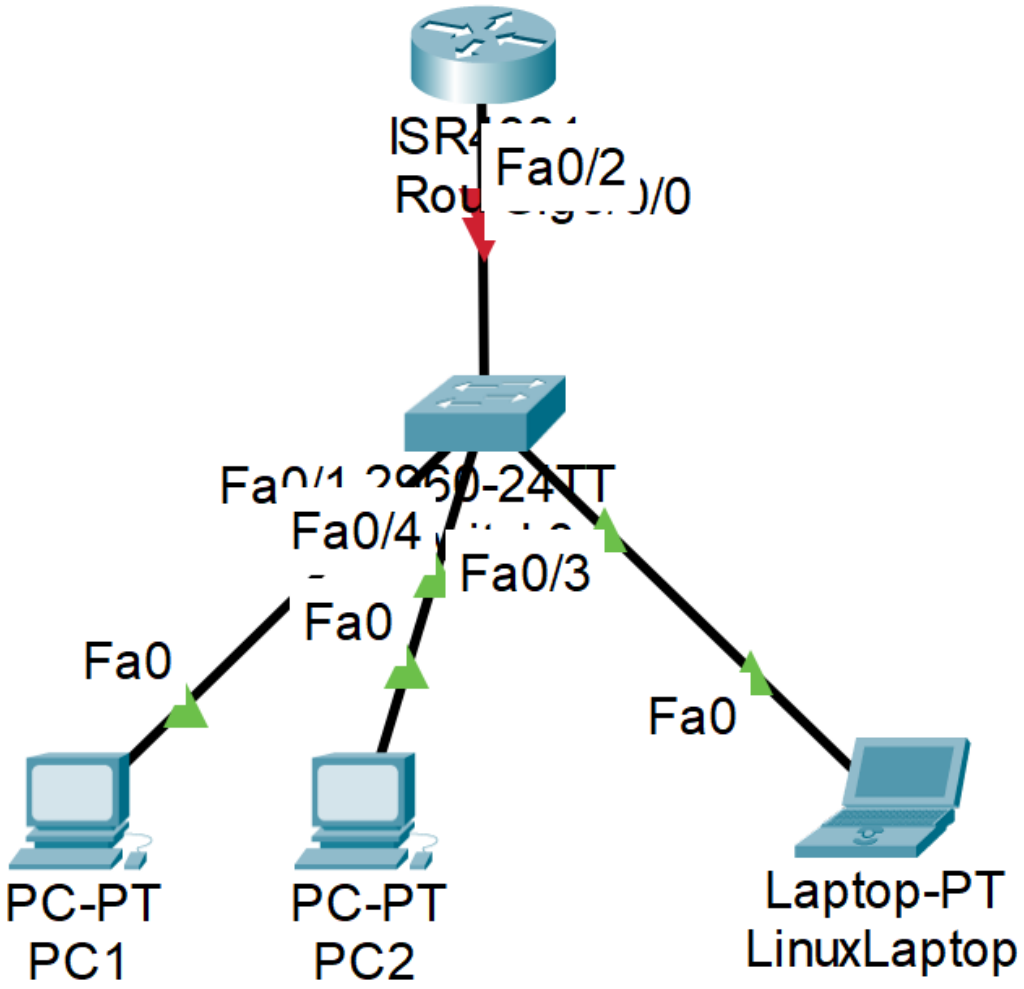
Pinging 192.168.10.1 with 32 bytes of data:
Reply from 192.168.10.1: bytes=32 time<1ms TTL=255
Reply from 192.168.10.1: bytes=32 time=1ms TTL=255
Reply from 192.168.10.1: bytes=32 time=1ms TTL=255
Reply from 192.168.10.1: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.10.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

Router to Linux PC ping

Router# ping 192.168.10.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms

Lab Diagram



Problems

What was particularly challenging:

One of the most difficult parts of this lab was successfully proving MAC flooding worked as intended. On the surface it seems MAC flooding should be the easiest yet the point of MAC flooding is to overwhelm the switch so it broadcasts every packet it received. We used Wireshark on the Linux PC But it ended up crashing everytime even with a filter because it would capture too many packets from itself. The way we fixed this problem was by adding

an additional PC. The additional PC would be connected to the switch via f1/0/4 and would listen using Wireshark to receive any ping packets meant for the router and not for the PC 2 itself.

Concepts we learned:
We learned how to carry out and recognize the three attacks, MAC flooding, DHCP starvation, and ARP spoofing. MAC flooding results in an full MAC table, DHCP starvation causes incomplete MAC entries which then denies further more entries, and ARP spoofing leads to multiple IP addresses being tied to a single MAC address as a Man in the middle attack. Additionally, we learned how to implement security measures such as port security, DHCP snooping, and dynamic ARP inspection to defend against these attacks.

What caused issues:
A major challenge was trying to run both the attacks and the corresponding defense mechanisms back-to-back. While we could successfully test them one at a time, combining them introduced problems especially with resetting the switch or ensuring our protections activated properly during the attack. Like I said earlier about MAC flooding, we weren't able to carry out the attacks while also capture packets on the same device so we used a new PC for the job.

Conclusion

In conclusion this lab involved carrying out three different Layer 2 attacks: DHCP starvation, ARP spoofing, and MAC flooding. While, executing them, we also set up configurations to defend the network from them. DHCP snooping was used against DHCP starvation, dynamic ARP inspection (DAI) was used to counter ARP spoofing, and port security was used on each active port for MAC flooding.