Cover Image Attribution: Joe Mabel CC-BY-SA 4.0

AWS Academy Cloud Foundations: Configuring Identity and Access Management, Virtual Private Cloud, and Elastic Compute Cloud

Colin J. Faletto, CCNA

aws academy

Purpose

This write-up is intended to document and explain the first three labs in the AWS Academy Cloud Foundations course. These labs are intended to provide an introduction to very basic AWS concepts, such as basic security, cloud networking, and virtual machines. These concepts are essential for new cloud engineers to provide parallels to the core skills of traditional IT. These labs also provide a strong foundation of knowledge for the AWS management console, as they all take place primarily inside this console.

Background

Amazon is a company based in Seattle that runs the biggest e-commerce platform in the world. They were started in 1994 by former CEO Jeff Bezos and have grown from a small online bookstore to a giant online store offering a wide variety of products. Amazon also has a strong physical retail presence in the grocery space with their Amazon Fresh and Whole Foods chains, and has a strong online media presence through Twitch, Prime Video, and Amazon Music, which provide entertainment in the form of livestreams, movies and television shows, and music respectively. Amazon also has a popular line of e-readers and tablets with their Kindle brand and has a successful brand of artificial intelligence assistants with their Amazon Alexa A.I. and their Amazon Echo line of smart speakers.

Amazon Web Services, or AWS, is Amazon's cloud computing division. It was created in 2002 to provide simple web services to customers and expanded to cloud storage and computing in 2006. It is the leading cloud service provider and is popular for its pay-as-you-go service model. AWS provides services to everyone from small businesses to massive companies like Coca-Cola and Apple, and even provides web infrastructure to government branches. AWS takes the responsibility and cost of managing a data center out of the hands of businesses and maintains a massive global network of Amazon data centers that split customer traffic among them. AWS currently has 34 geographic regions, each of which have multiple availability zones which themselves contain multiple data centers. These data centers are in undisclosed locations for security reasons, though their general position is published. AWS offers services for virtual machines, cloud storage, database management, machine learning, IoT services, cloud networking, and much more.

Amazon Elastic Compute Cloud, or EC2, is an AWS service that allows customers to create virtual machines in the AWS cloud. These machines are very versatile, as they can be allocated as many or as few resources (CPU, RAM, GPU) as needed and can run nearly any operating system. By default, EC2 instances will run Amazon Linux, which is a version of Linux optimized for AWS servers. Amazon's e-commerce platform, its primary source of revenue, has been running on EC2 instances for over a decade. EC2 instances have a variety of different types, which are optimized for different purposes such as memory (R series, X series), compute (C series), and storage (H series, I series, D series).

Amazon Virtual Private Cloud, or VPC, is an AWS service that provides a virtual network inside the AWS cloud. This service allows AWS objects, such as EC2 instances, to communicate with each other. In a VPC, each EC2 machine is assigned a

unique private IPv4 address, which is then connected via NAT to a public address on an internet gateway. Using this gateway, machines in the VPC can communicate with other AWS VPCs and other Internet-connected machines. Amazon VPC is provided at no additional charge to customers using EC2 instances.

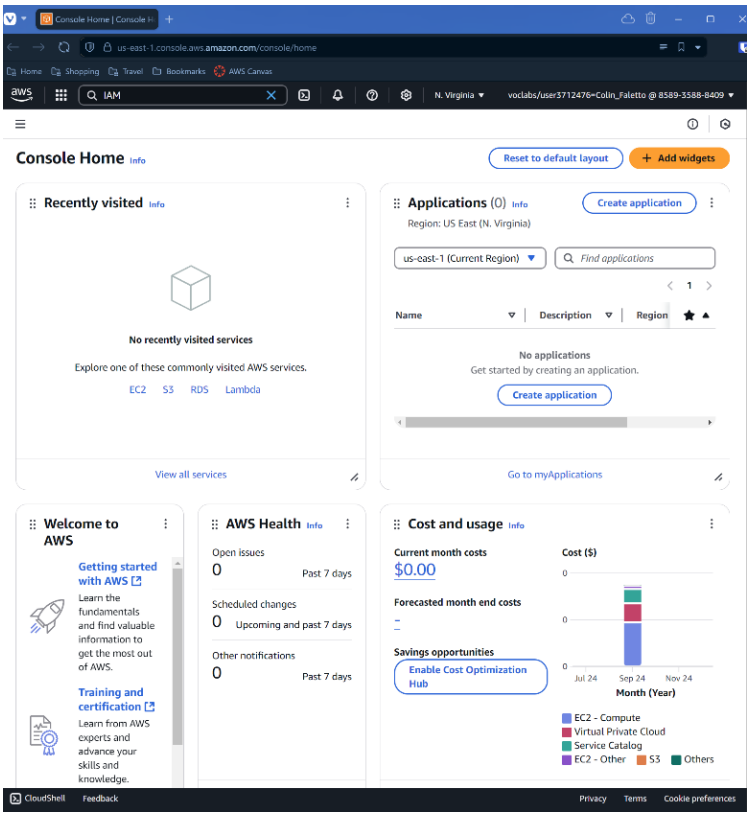
Amazon Identity and Access Management, or IAM, is an AWS service that provides a layer of security to customers by limiting the resources different users can access. IAM follows the principle of least privilege, meaning that by default, all AWS controls are blocked for users unless they have been explicitly granted permissions. IAM represents a portion of the customer responsibilities in the AWS shared responsibility model, which is a model outlining that AWS is responsible for the physical security of data centers and networks while the customer is responsible for keeping their customer data and configurations safe. One of IAM's unique features is its role feature, which creates identities with elevated permissions that can be temporarily assigned to users. This feature works similarly to the "sudo" command in unix-based operating systems.

Lab Summary

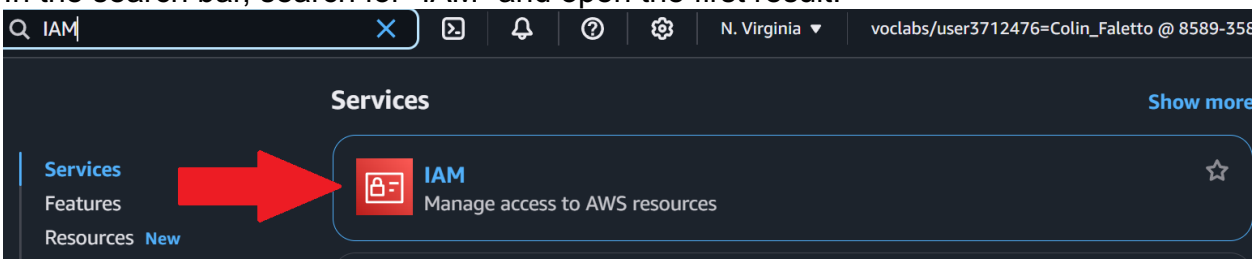
This write-up covers three different AWS labs. The first lab covers AWS IAM. The lab entails assigning permissions to three different users to allow them to interact with S3 and EC2 services. Two of these users are support users, who are assigned read-only permissions to EC2 and S3 respectively. The other user is an administrator with start and stop access to EC2 instances. The second lab covers AWS VPC, and entails creating a VPC with four different subnets, two public and two private, spread across two different availability zones. The lab also involves creating a security group to allow HTTP access to machines in the second public subnet. The third lab involves creating a web server EC2 instance, allowing HTTP access through a security group, then resizing the instance and testing the stop protection feature.

Lab Commands (Lab 1 IAM)

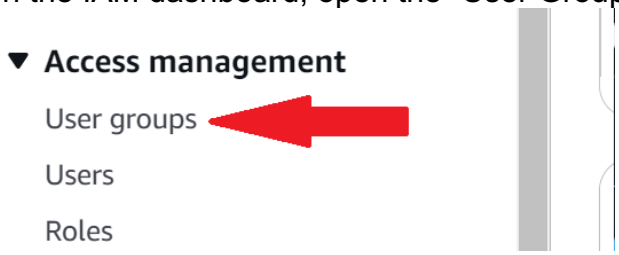
Log into the AWS management console.



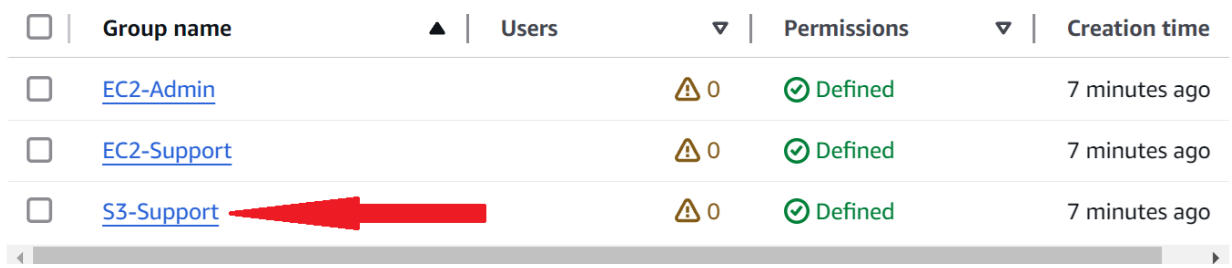
In the search bar, search for “IAM” and open the first result.



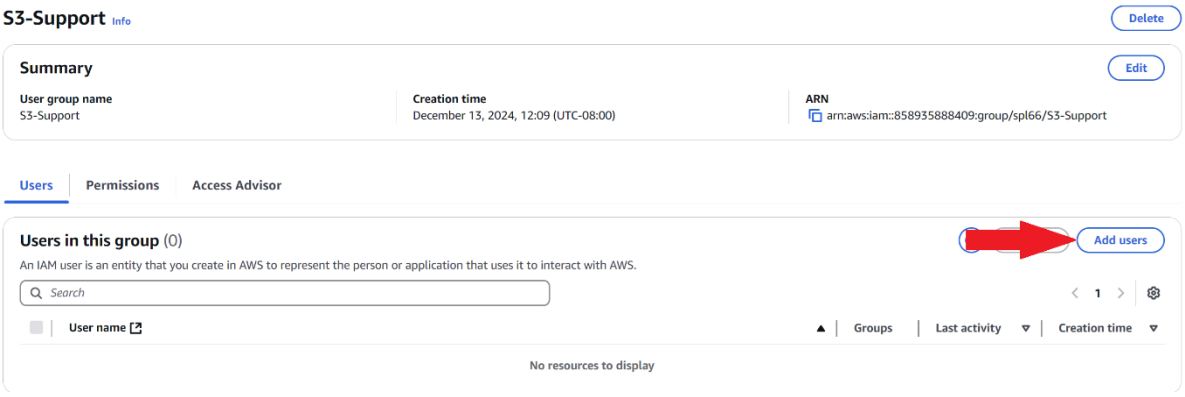
In the IAM dashboard, open the “User Groups” tab from the sidebar.



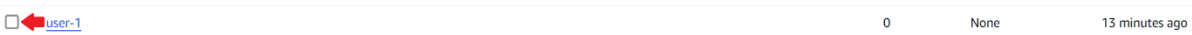
Open the “S3-Support” group.



Click “Add Users”.

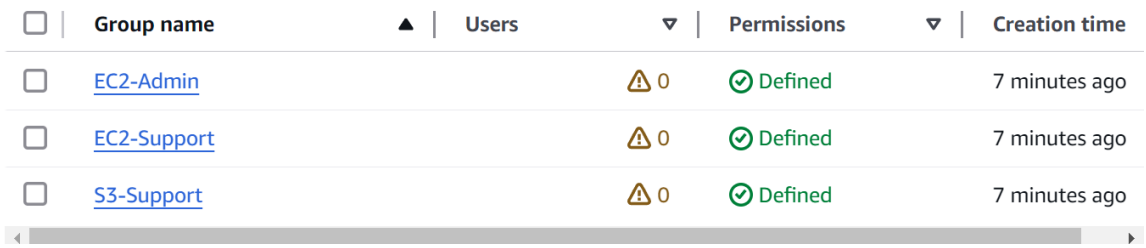


Check the box next to “user-1”.

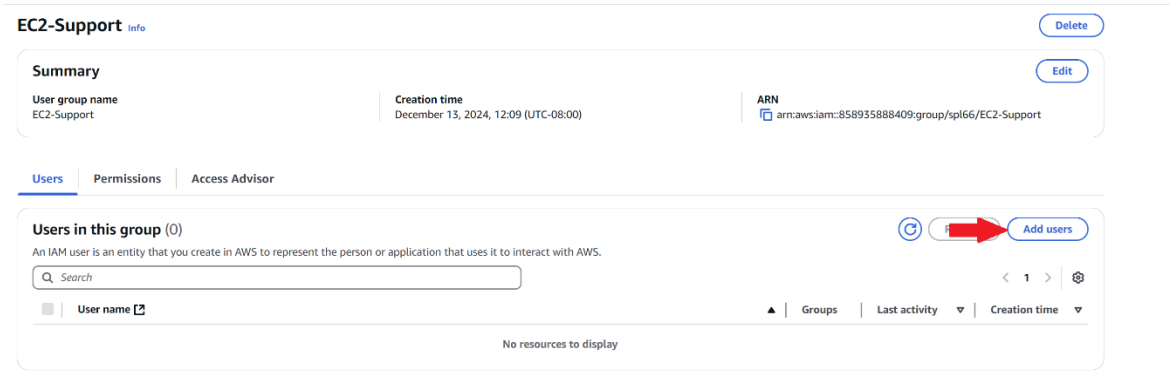


Click “Add Users”.

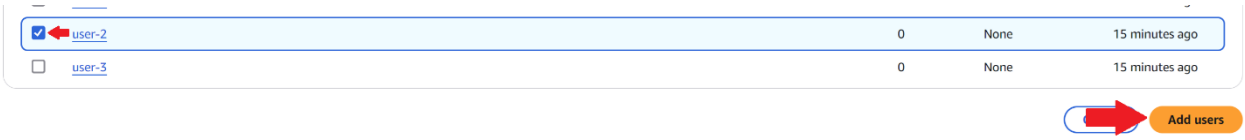
Return to the “User Groups” section and select “EC2-Support”.



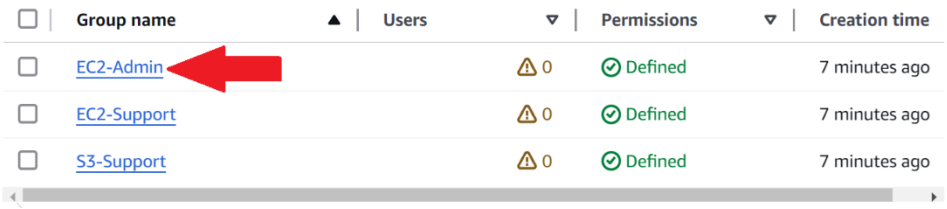
Click “Add Users”.



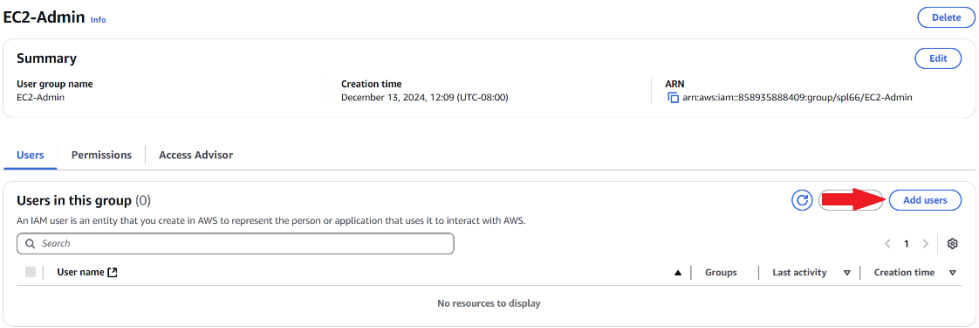
Select “user-2” and click “Add Users”.



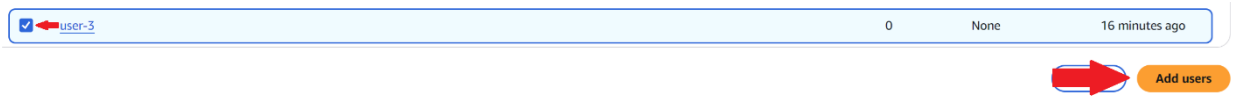
Return to the “User Groups” section and click on “EC2-Admin”.



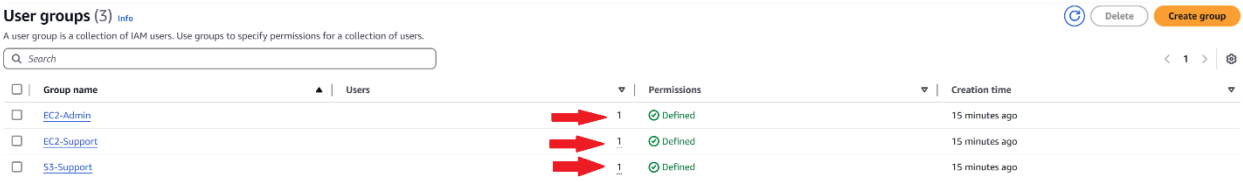
Click “Add Users”.



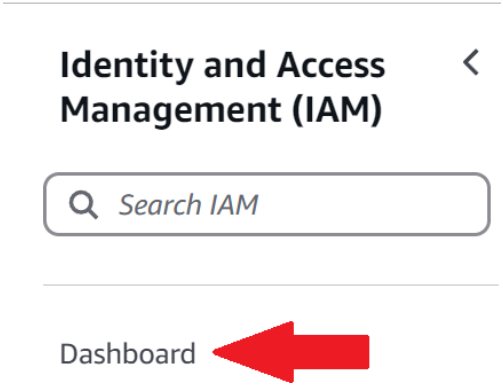
Click the checkbox next to “user-3” and click “Add Users”.



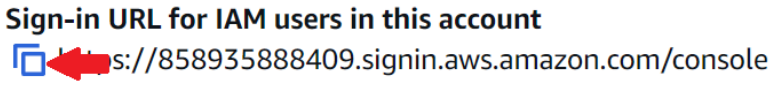
If done correctly, each user group should have a “1” next to it.



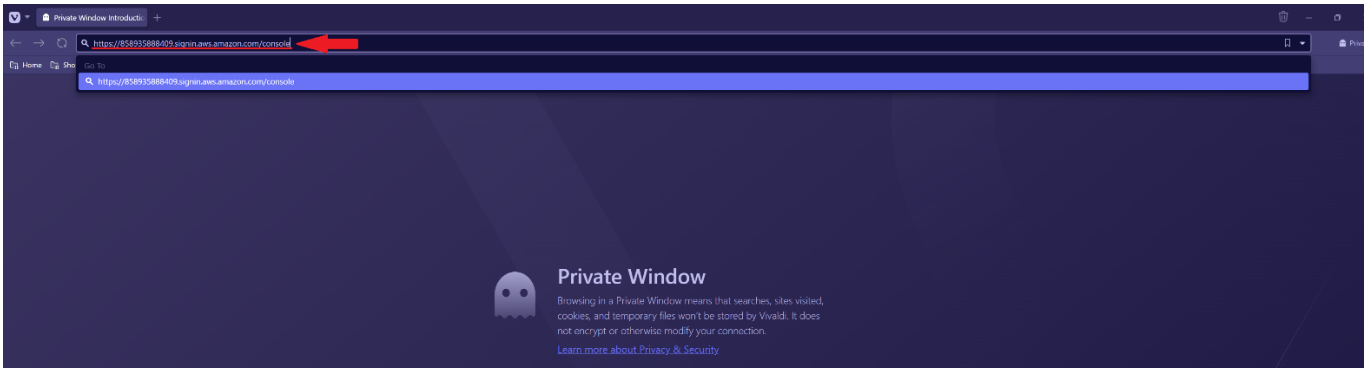
Return to the dashboard.



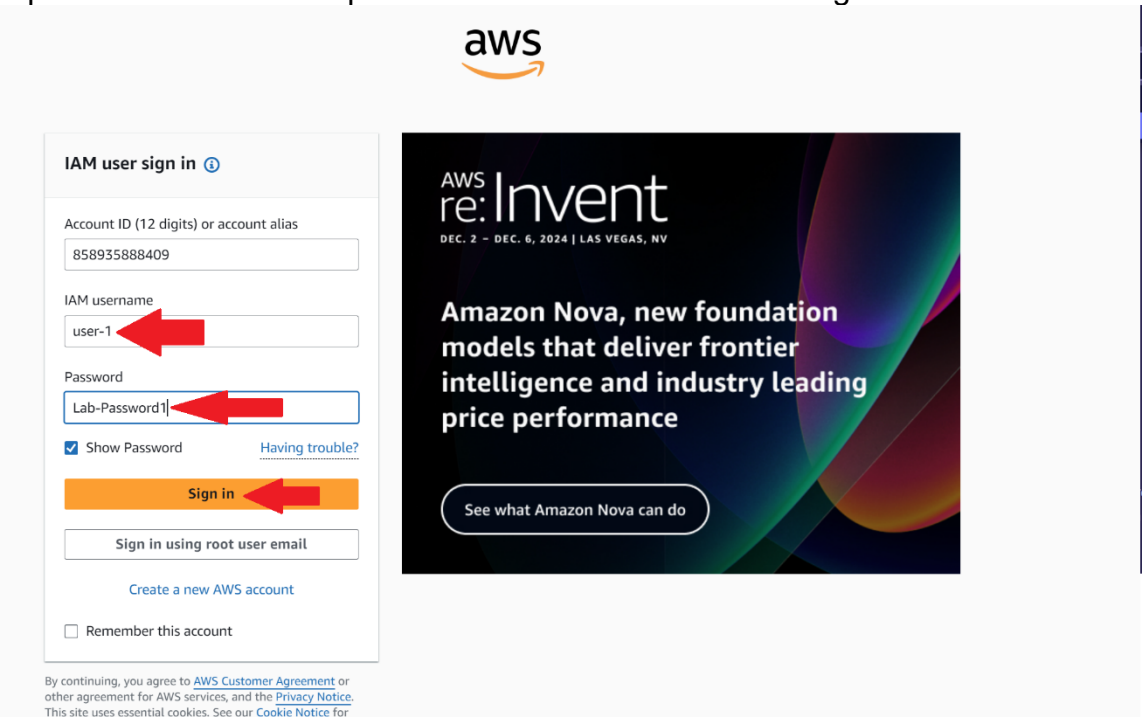
On the dashboard page, find and copy the sign-in URL for IAM users. This URL will be used to test the permissions that have just been set up.



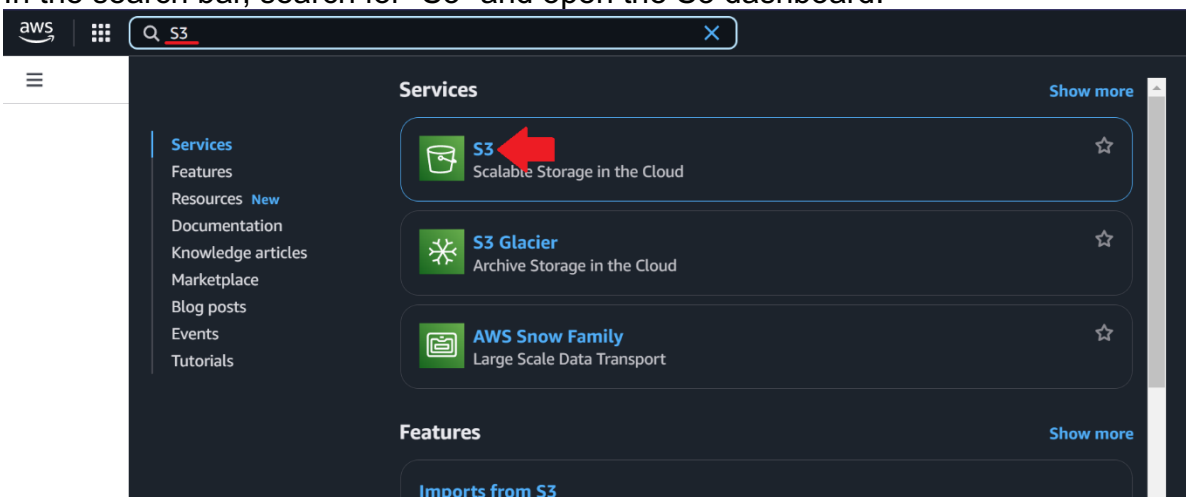
Open a new private browser window and input the URL.



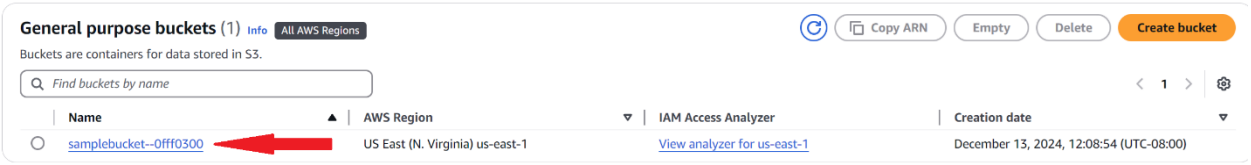
Input the username and password for “user-1” and click “Sign In”.



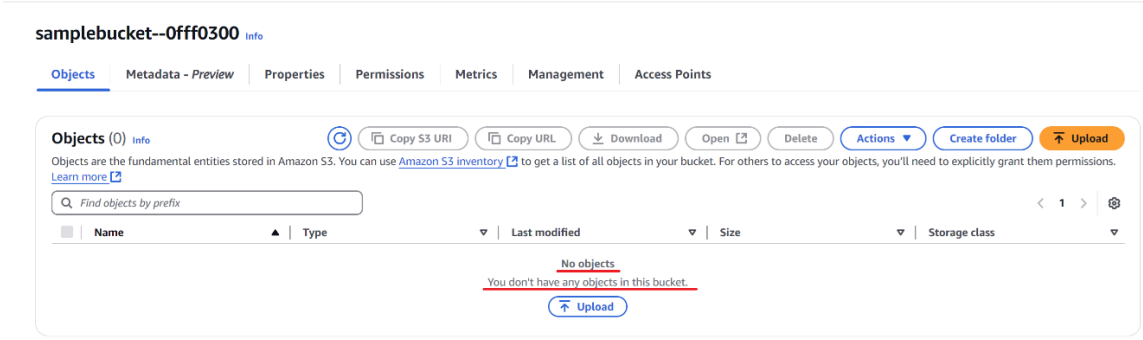
In the search bar, search for “S3” and open the S3 dashboard.



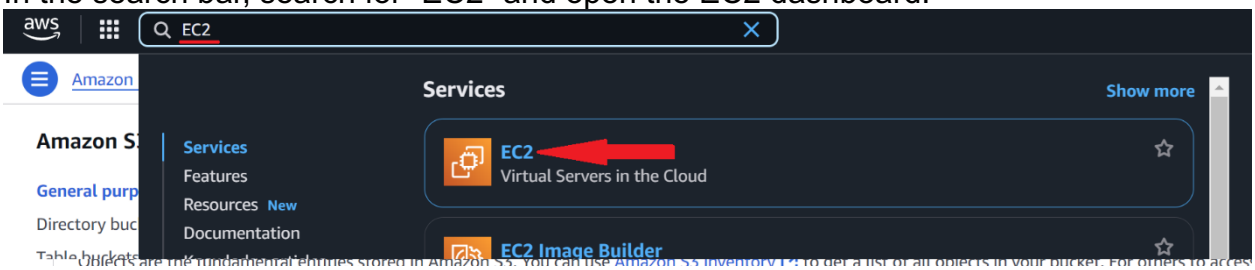
From the S3 dashboard, click on the sample bucket shown.



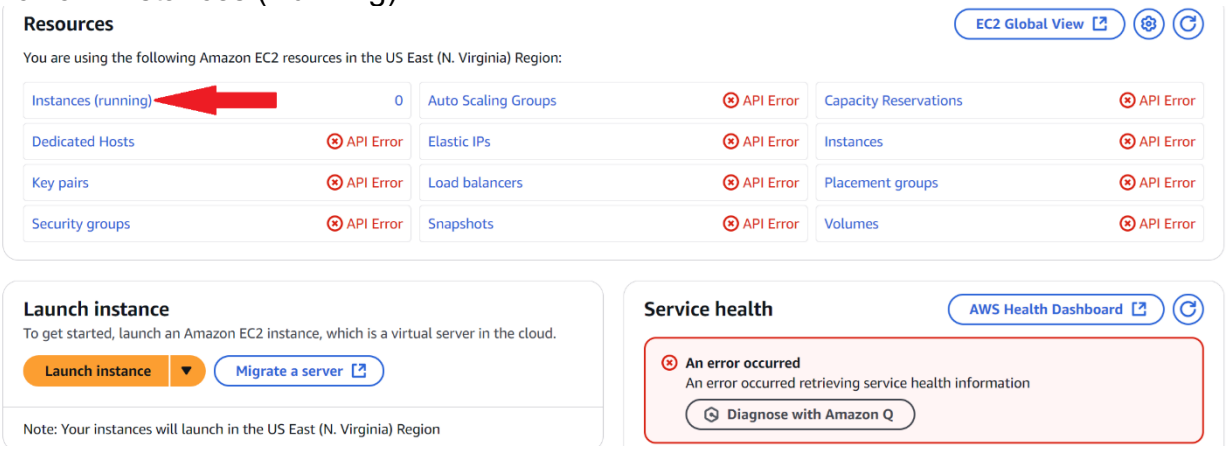
If permissions have been set correctly, user-1 should be able to see the objects in the bucket. In this case, there are no objects in the bucket, which should be evident in a message displayed to the user.



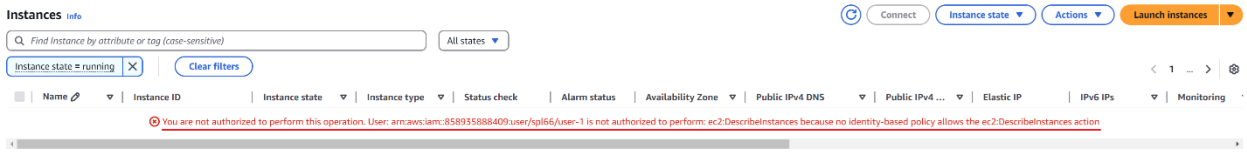
In the search bar, search for “EC2” and open the EC2 dashboard.



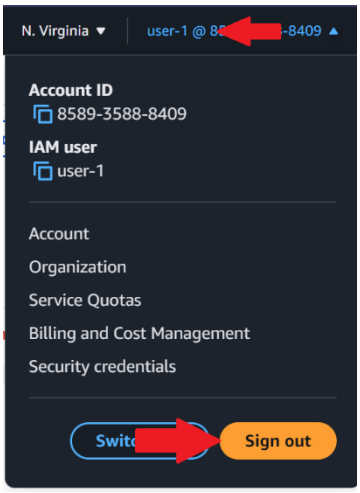
Click on “Instances (Running)”.



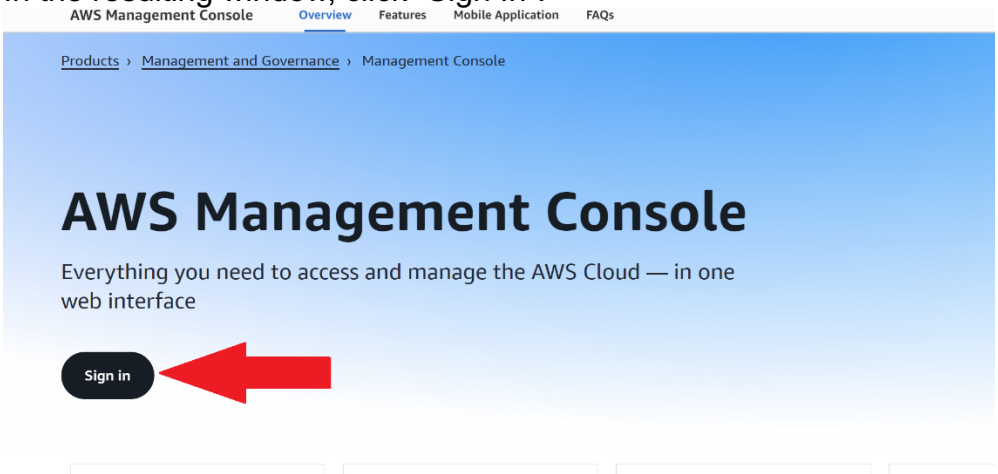
If permissions are set up correctly, the following error will occur:



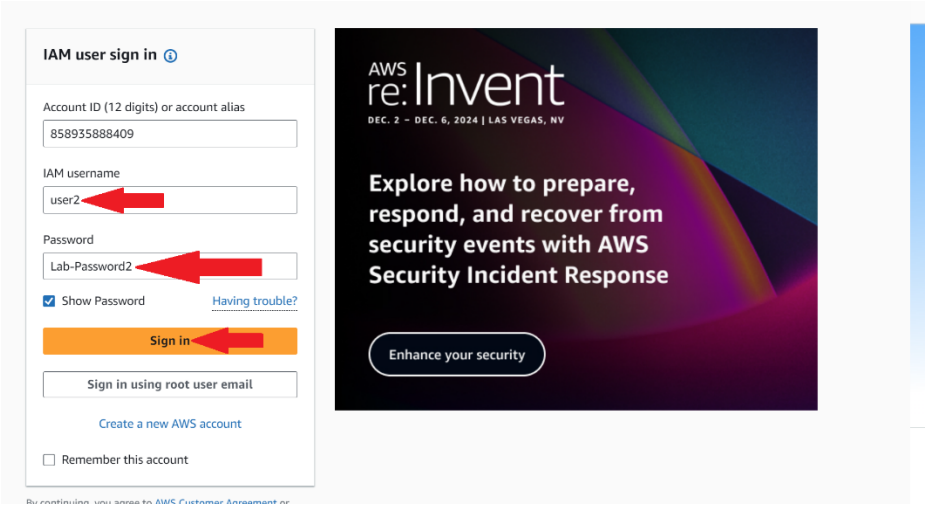
This is intended behavior, as user-1 is only authorized to access S3 services.
Click the “user-1” button in the top right corner and click “Sign Out”.



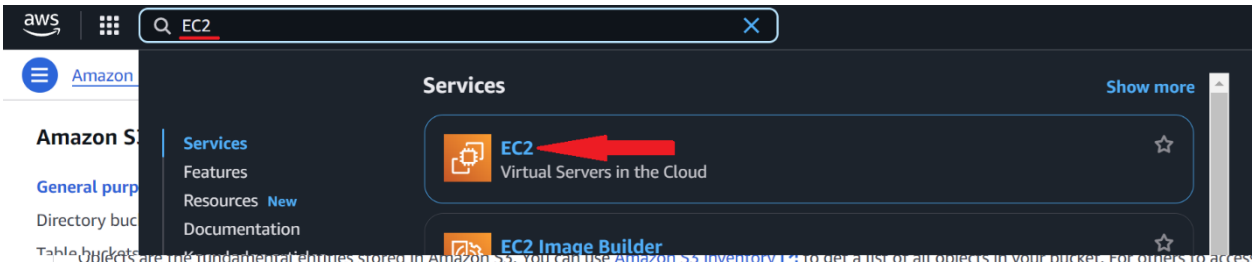
In the resulting window, click “Sign In”.



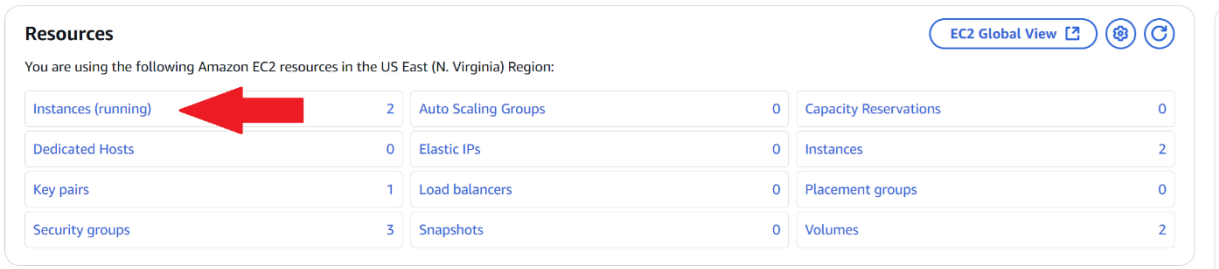
Enter the username and password for user-2 and click “Sign In”. Your AWS account ID should be automatically filled, but if it isn’t, you can paste the same sign-in URL from before.



Return to the EC2 dashboard.



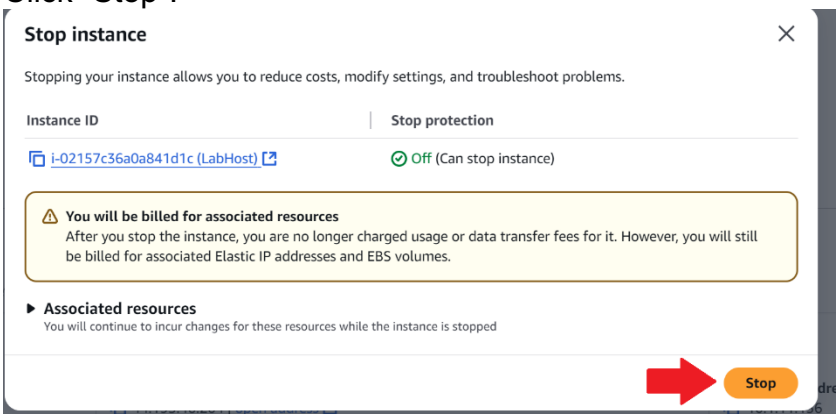
Click on “Instances (running)”. You’ll notice a distinct lack of “API Error” messages, unlike the page for user-1.



You’ll notice that the instances will properly display this time. Select the “LabHost” instance.

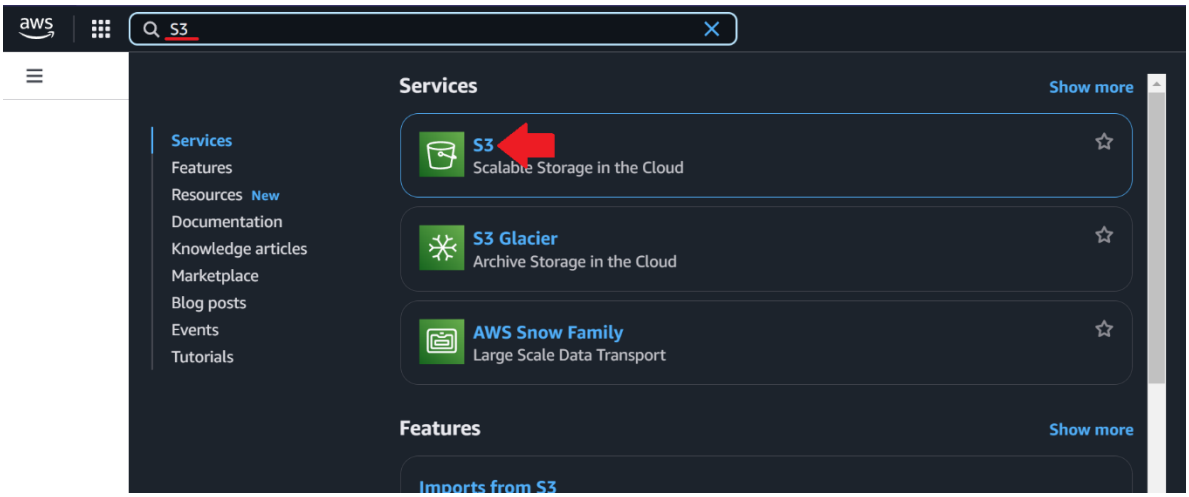
Click on “Instance state” > “Stop Instance”.

Click “Stop”.

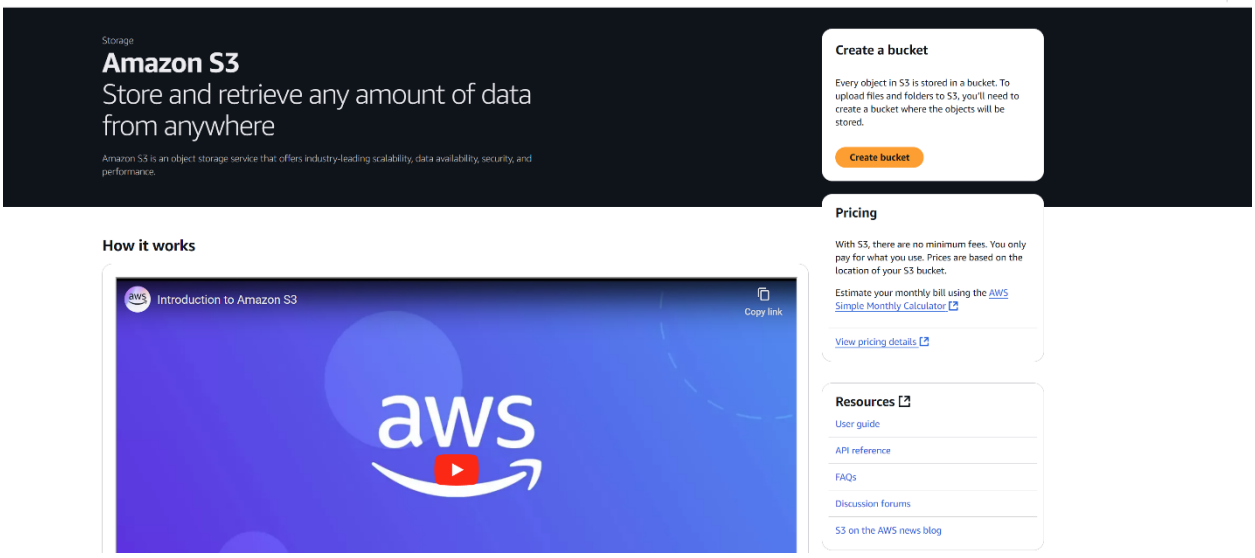


Since user-2 doesn’t have the appropriate permissions, you should get the following error message:

To confirm that user-2 doesn't have access to S3, access the S3 dashboard from the search bar.

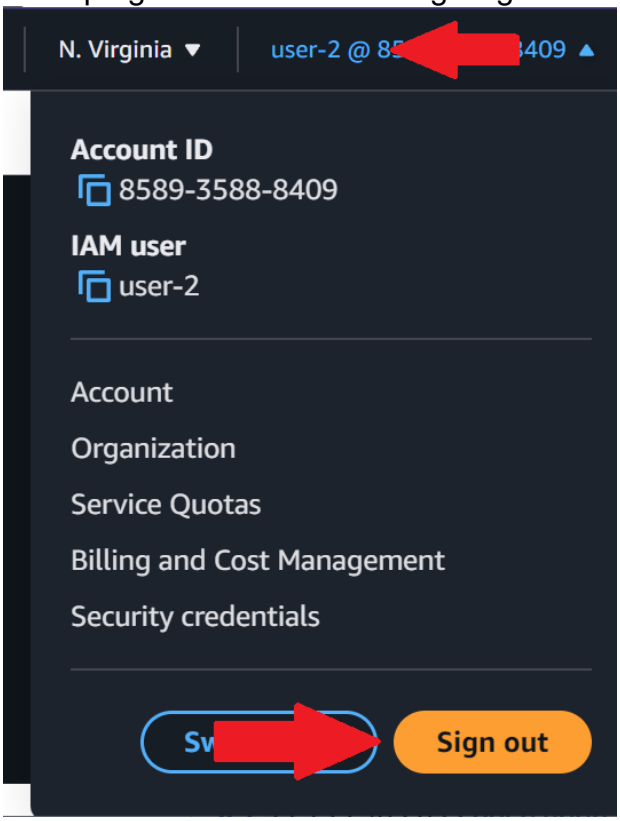


You should see the following page, where it appears that no S3 buckets have been set up (user-2 is unable to see these buckets, so AWS treats them as if they don't exist here).

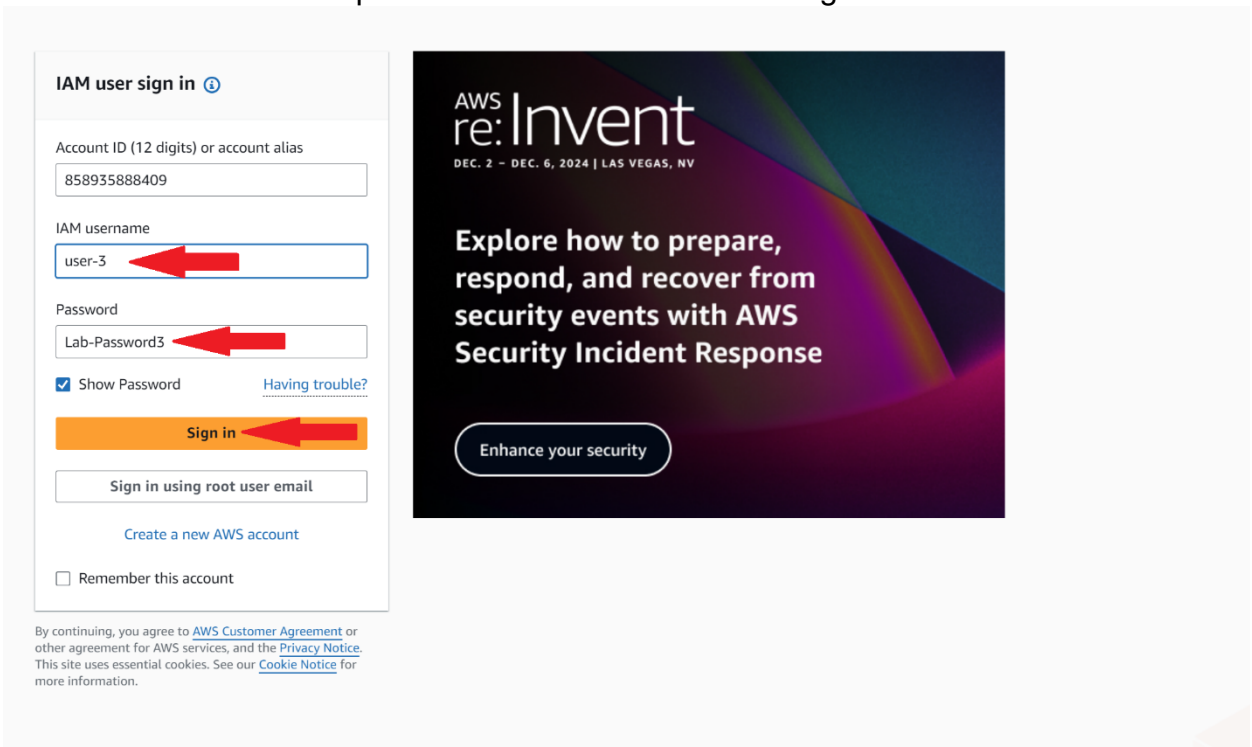


Attempting to create a bucket as user-2 will result in the following error message:

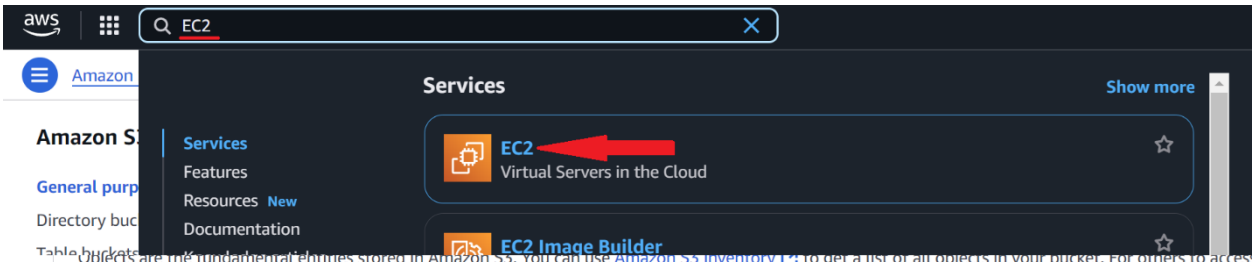
Now, switch to user-3. Sign out of the user-2 account by clicking the “user-2” button in the top right corner and clicking “Sign Out”.



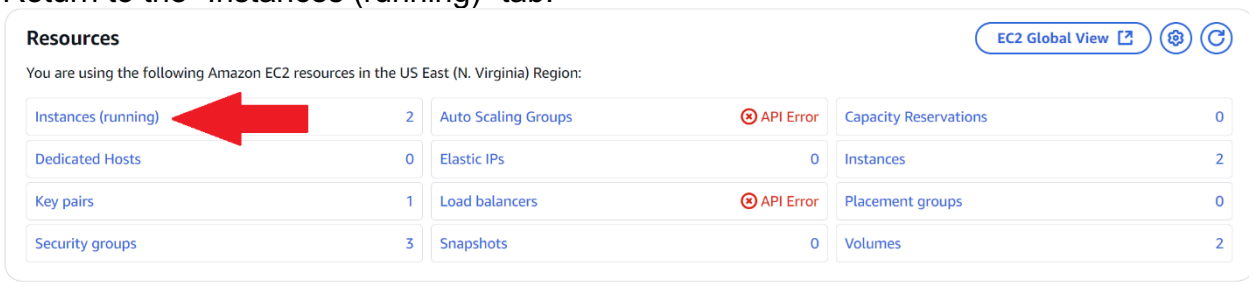
Enter the username and password for user-3 and click “Sign In”.



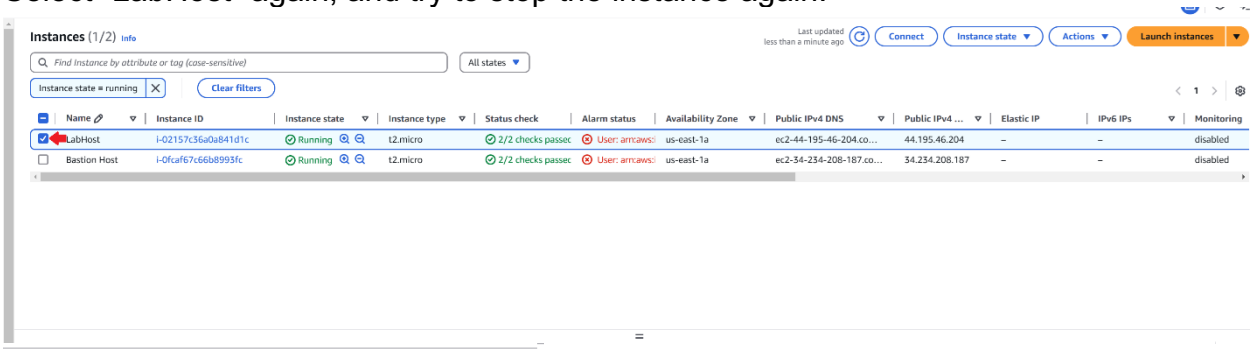
Return to the EC2 dashboard.



Return to the “Instances (running)” tab.



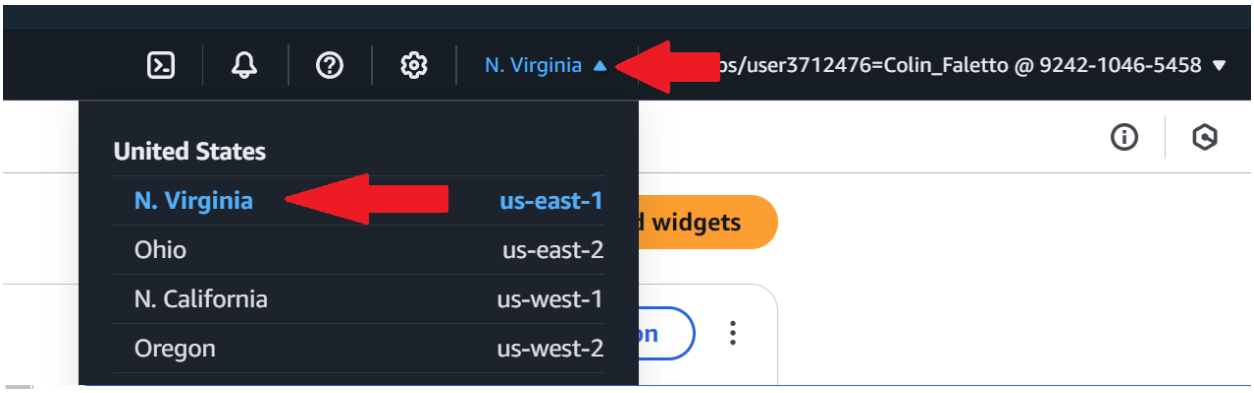
Select “LabHost” again, and try to stop the instance again.



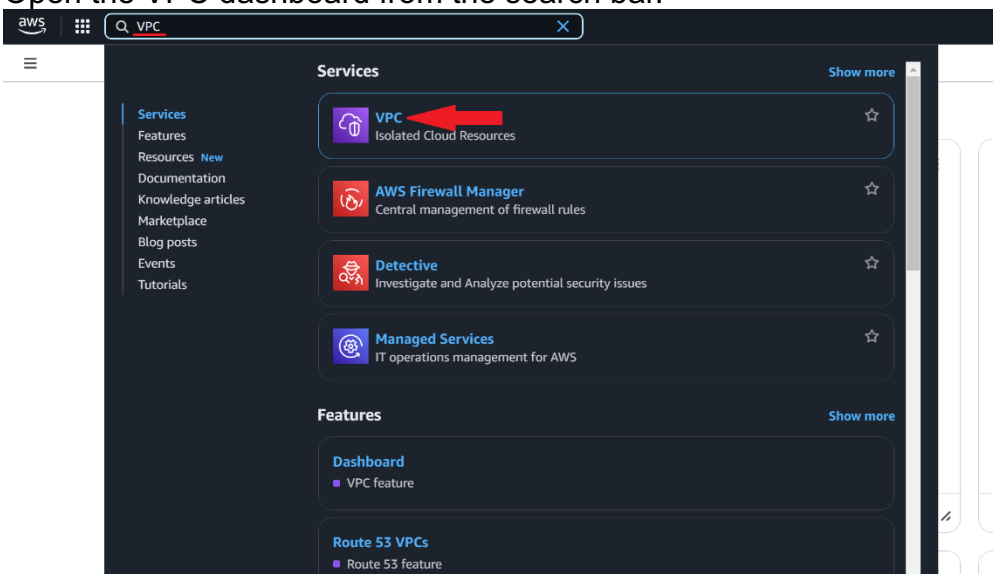
If user-3’s permissions are correctly set up, you should see the following success message:

Lab Commands (Lab 2 VPC)

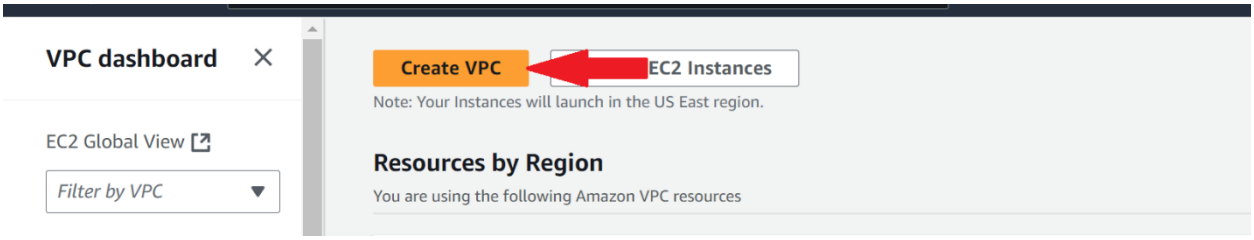
In the AWS management console, ensure that you have the “N. Virginia (us-east-1)” region selected.



Open the VPC dashboard from the search bar.

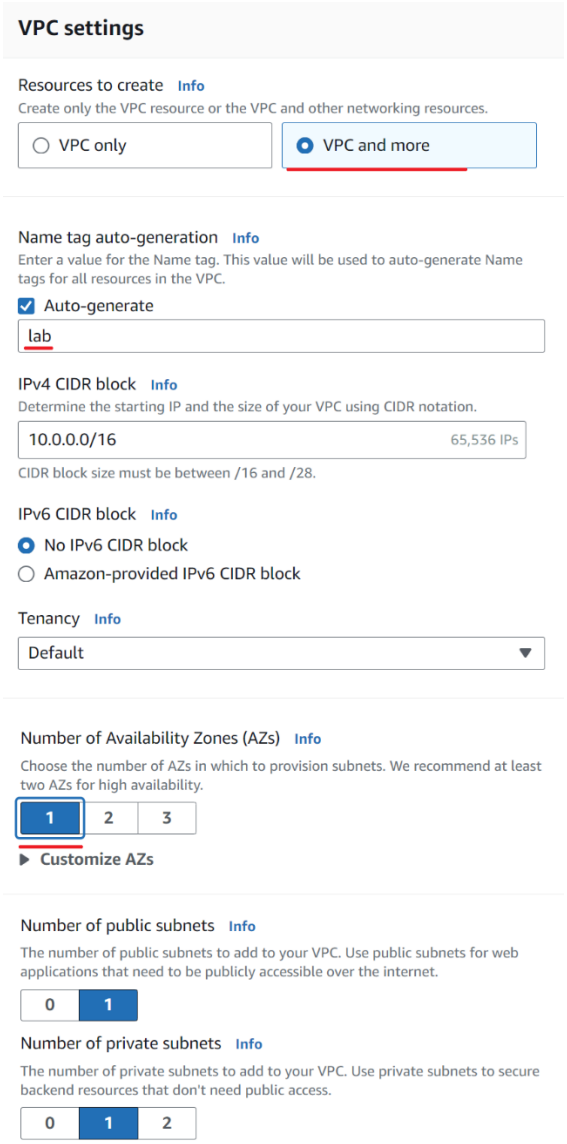


Click “Create VPC”.

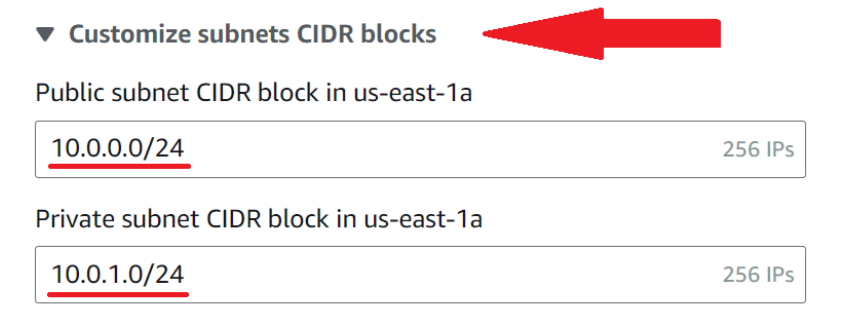


Keep all settings on their default values (ensure they match with the screenshot below) besides the following:

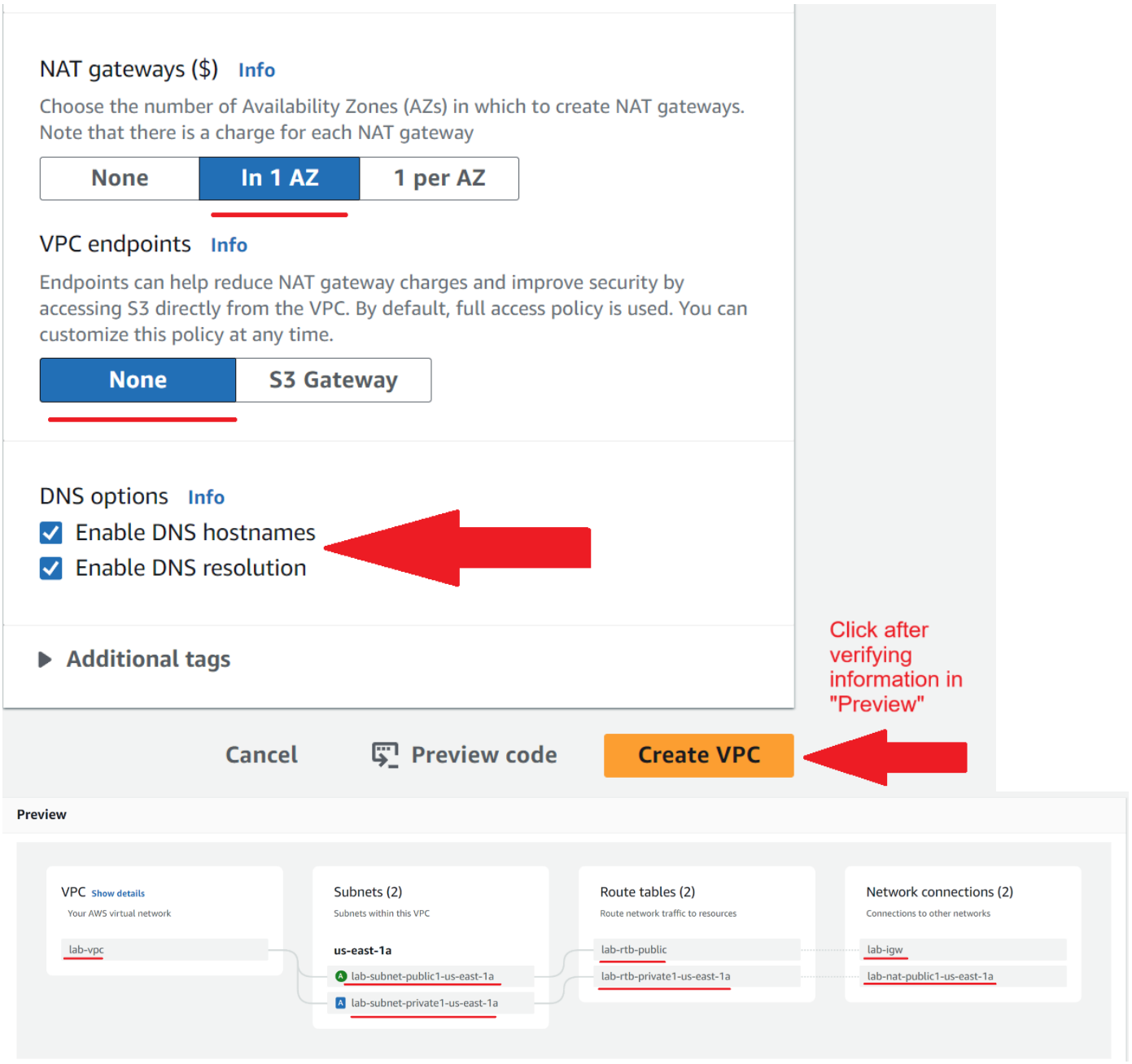
- Set “Resources to create” to “VPC and more”
- Set “Name tag auto-generation” to “lab”
- Set “Number of Availability Zones (AZs)” to “1”



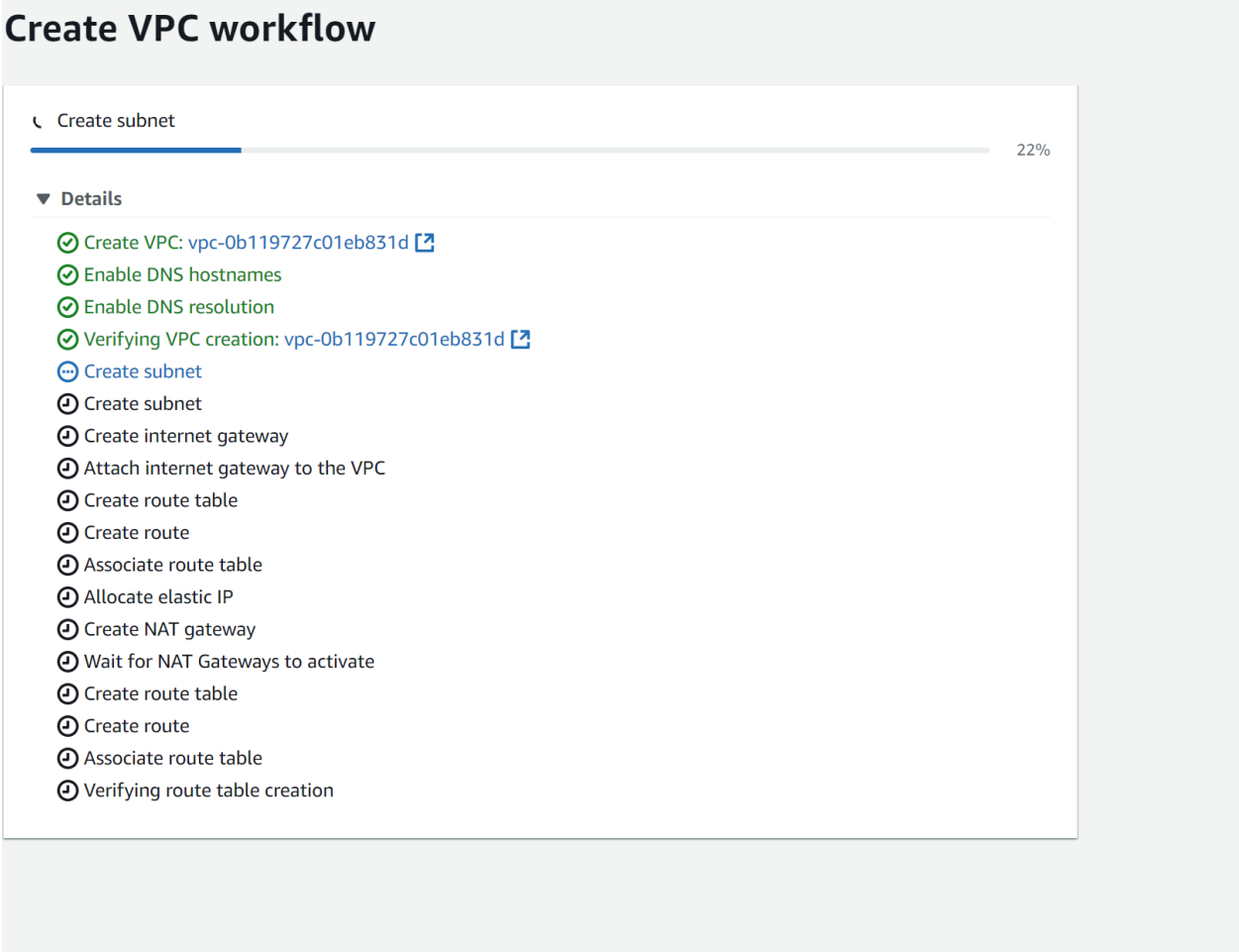
Open the “Customize subnets CIDR blocks” section and ensure that the public and private subnet CIDR blocks are set to 10.0.0.0/24 and 10.0.1.0/24 respectively.



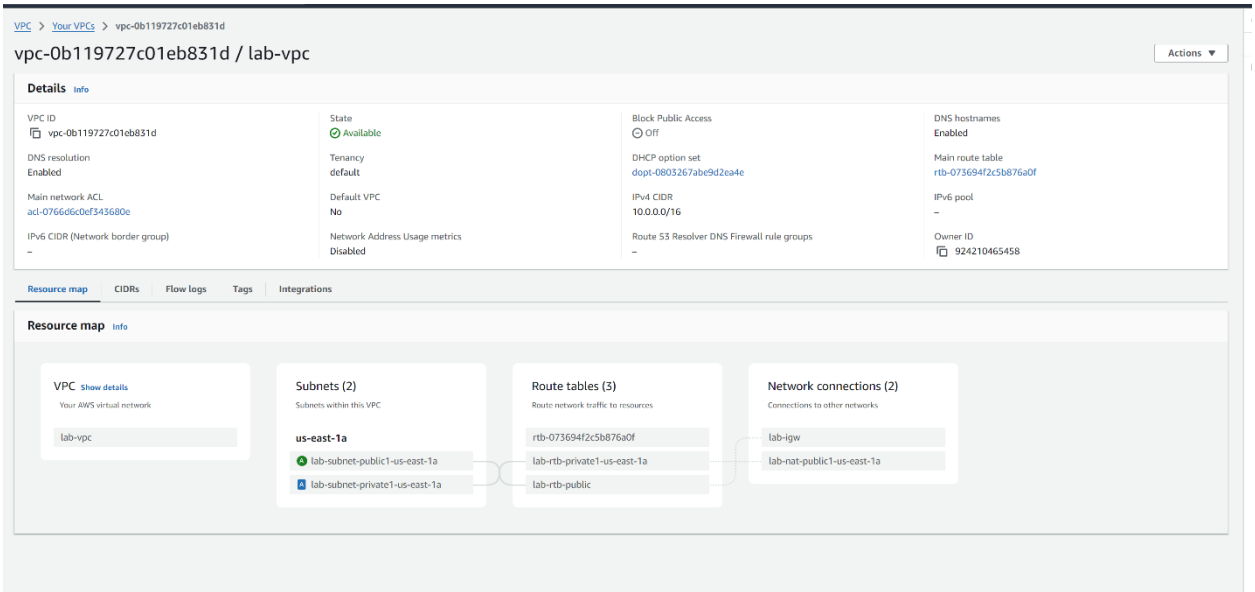
Set “NAT gateways” to “In 1 AZ”, set “VPC Endpoints” to “None”, and ensure that both boxes under “DNS options” are checked. Verify that the Subnets, Route Tables, and Network connections in the “Preview” section match the screenshot below, then click “Create VPC”.



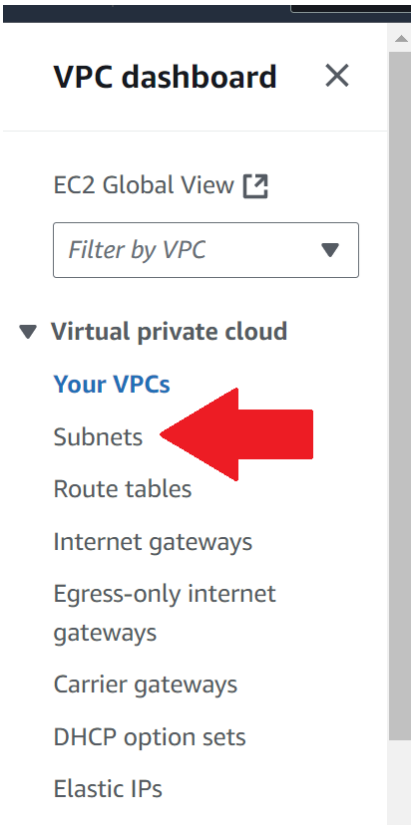
You should see a window like this while the VPC is being created:



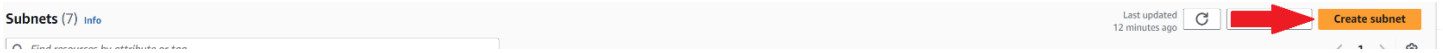
You should see a screen like this when the VPC has finished being created:



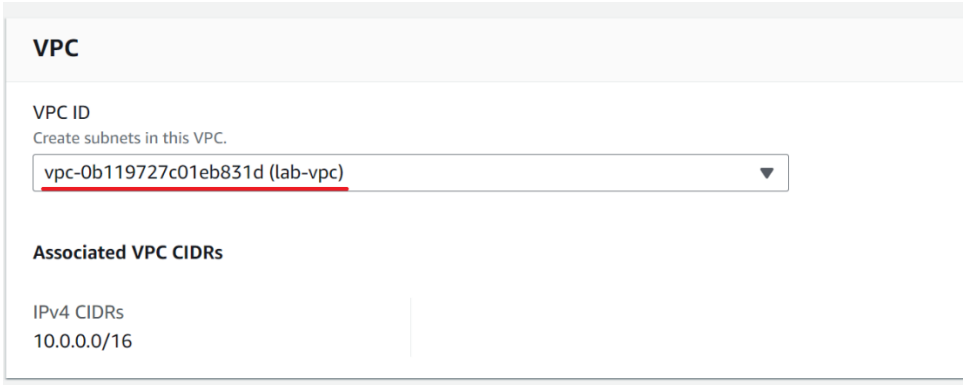
You will now create additional subnets in a second availability zone. From the sidebar, click on “Subnets”.



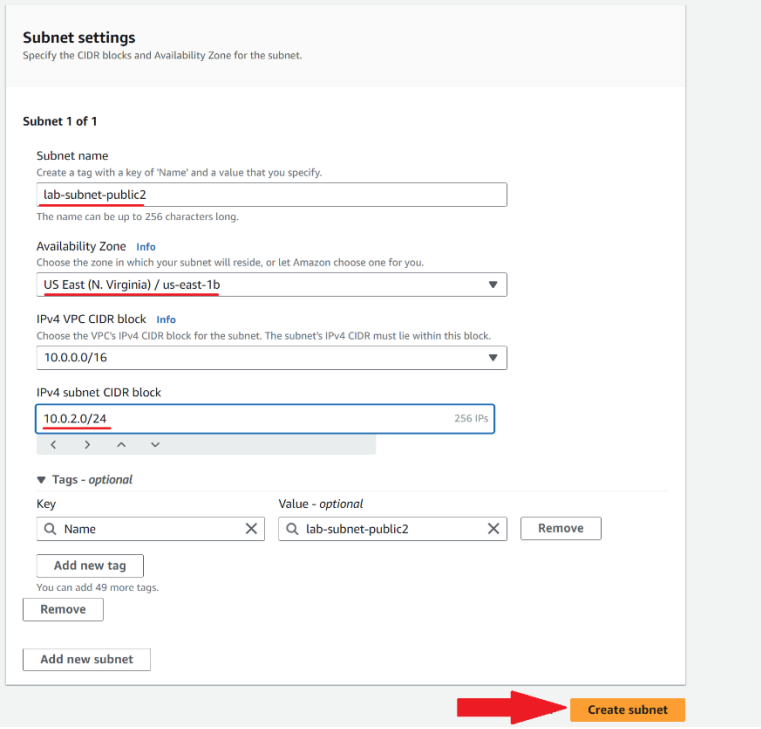
Click “Create Subnet”.



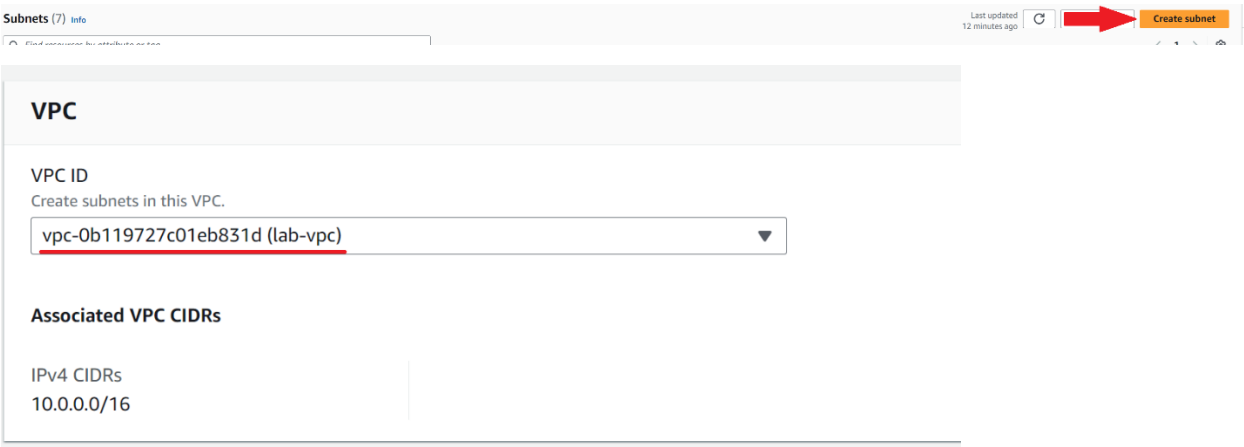
Under “VPC”, select “lab-vpc”.



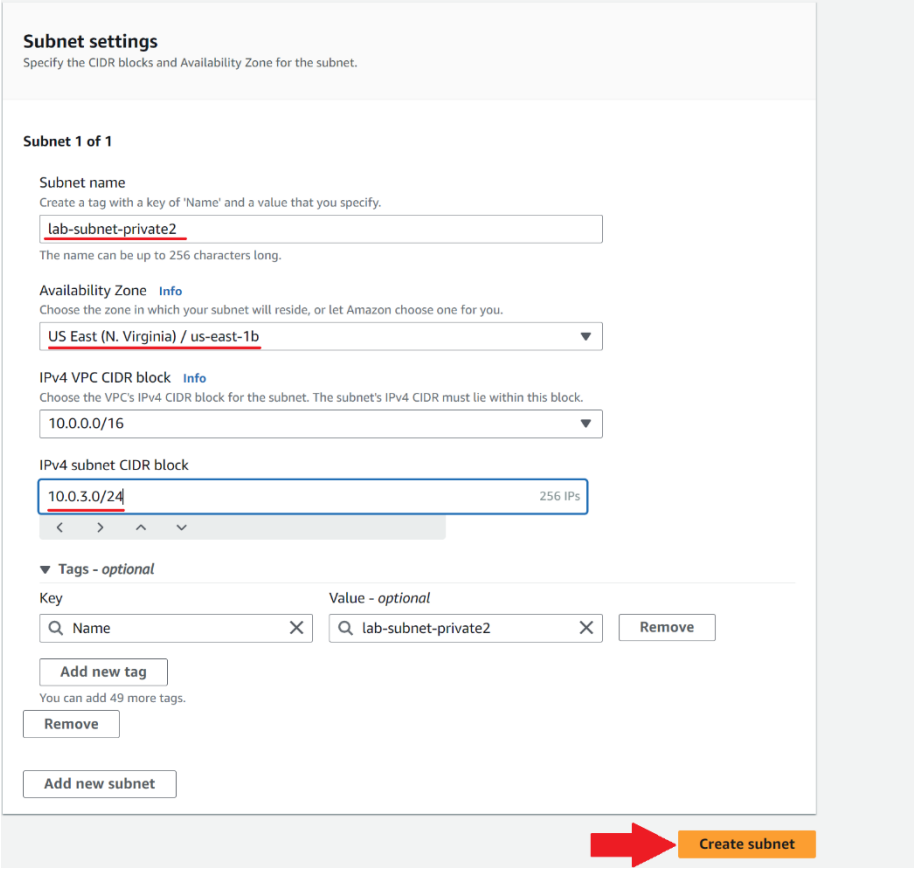
Under “Subnet settings”, set the name to “lab-subnet-public2”, the availability zone to “us-east-1b”, and the IPv4 subnet CIDR block to 10.0.2.0/24. Click “Create Subnet”.



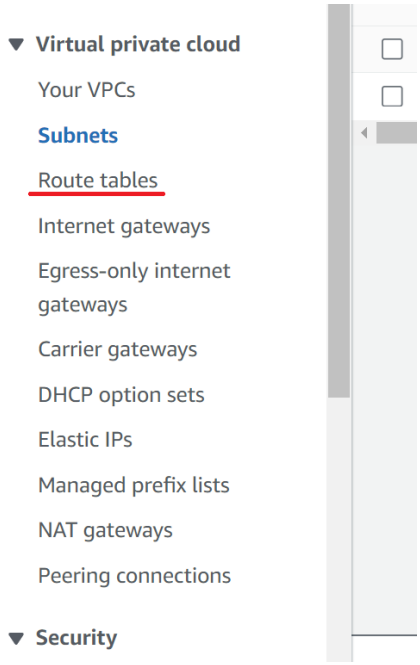
From the subnet dashboard again, select “Create Subnet” and set the VPC to “lab-vpc” again.



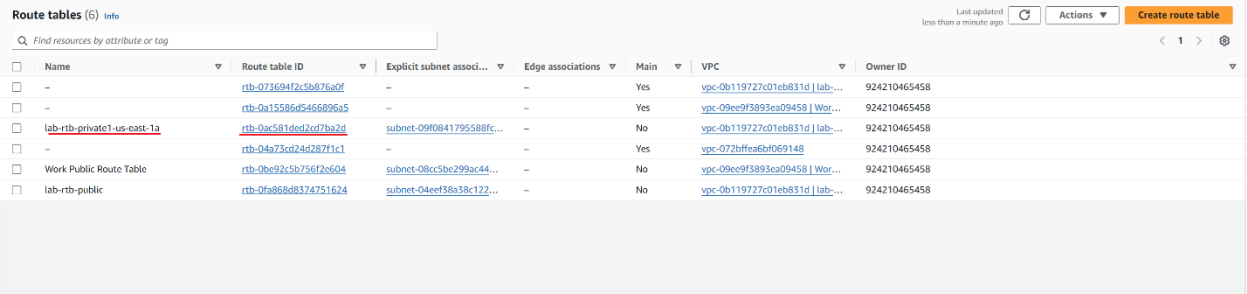
Under “Subnet settings” this time, set the name to “lab-subnet-private2”, the Availability Zone to “us-east-1b”, and the IPv4 Subnet CIDR block to 10.0.3.0/24. Click “Create Subnet”.



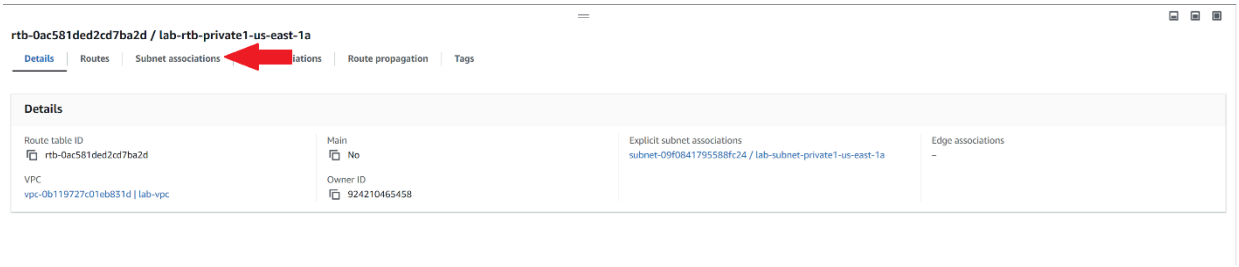
Next, set the second private subnet to share a routing table with the first private subnet. Go to the “Route tables” section from the sidebar.



Click on the “Route table ID” hyperlink associated with “lab-rtp-private1-us-east-1a”.

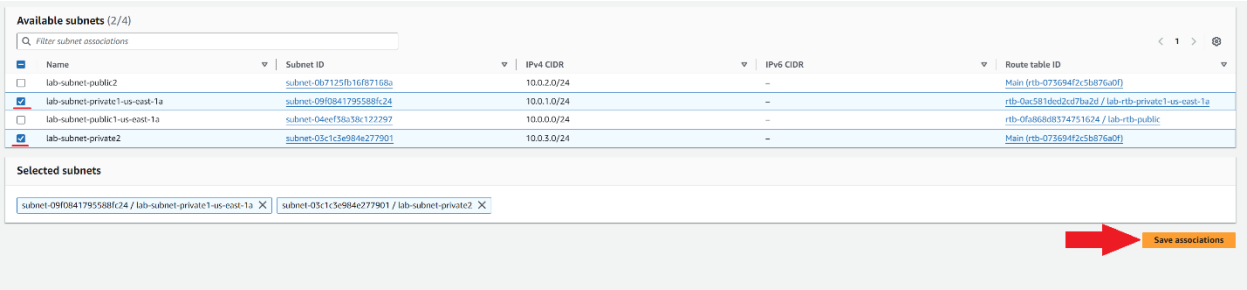


Click “Subnet associations”.

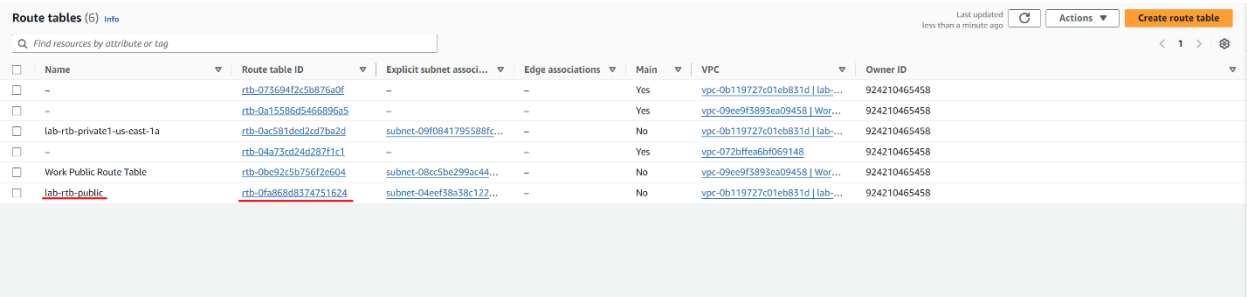


Click “Edit subnet associations”.

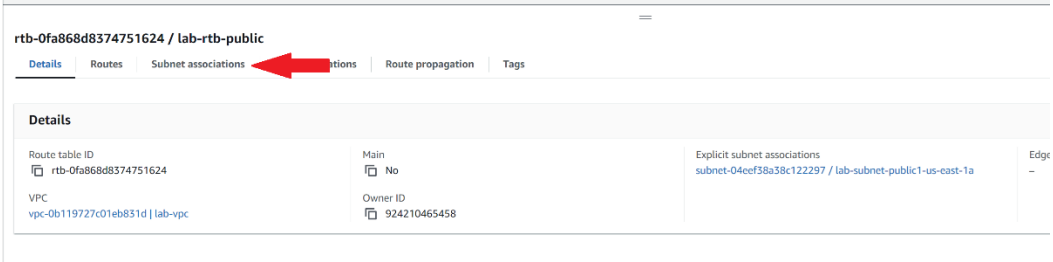
Select both private subnets, then click “Save associations”.



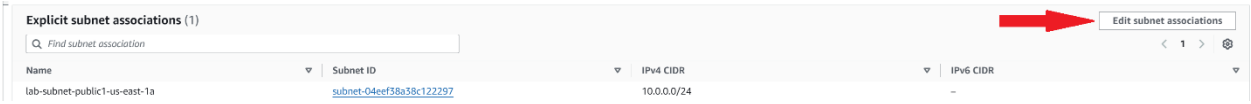
Return to the “Route tables” page. Click on the “Route table ID” hyperlink of the “lab-rtp-public” table.



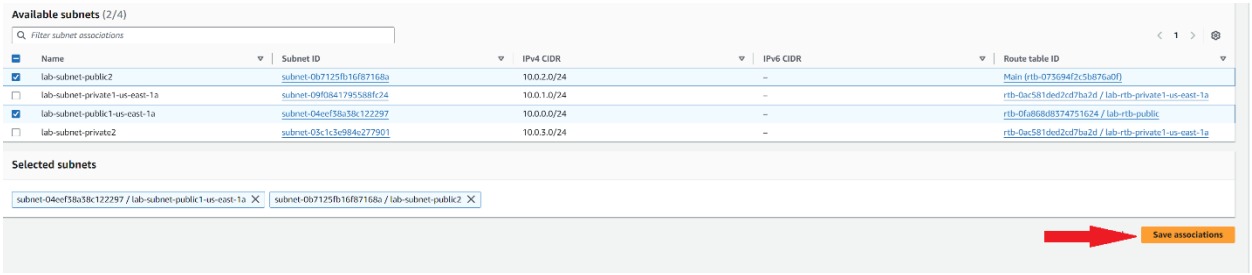
Click “Subnet associations”.



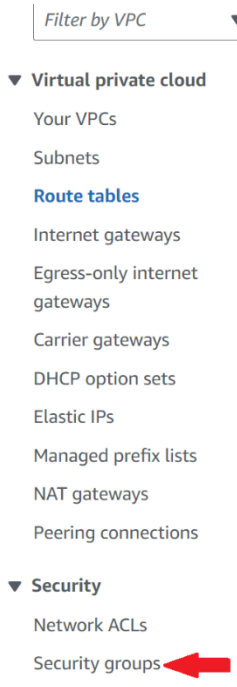
Click “Edit subnet associations”.



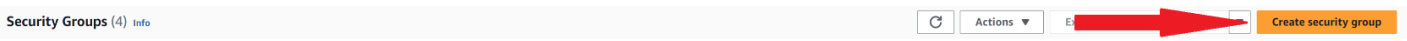
Select both public subnets, then click “Save associations”.



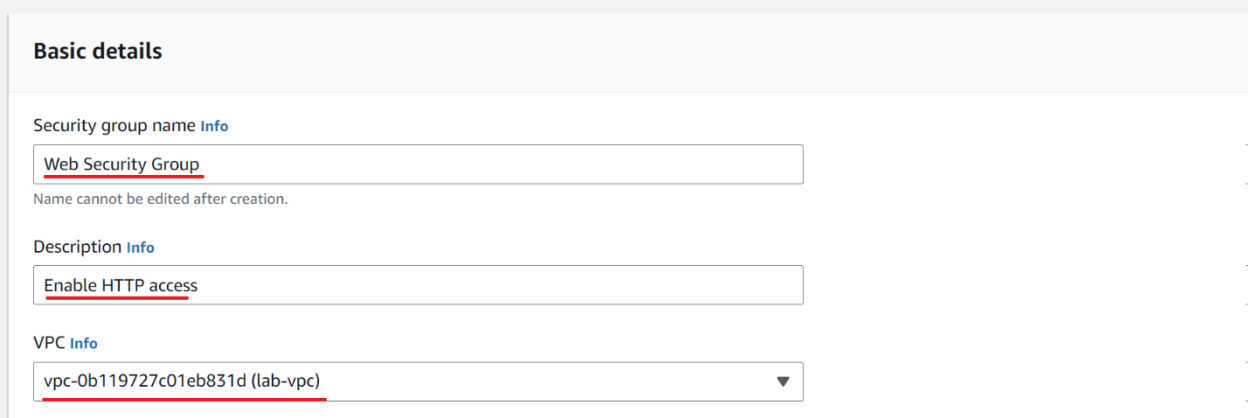
Next, you will configure a security group to permit HTTP access to a web server. Go to the “Security groups” page from the sidebar.



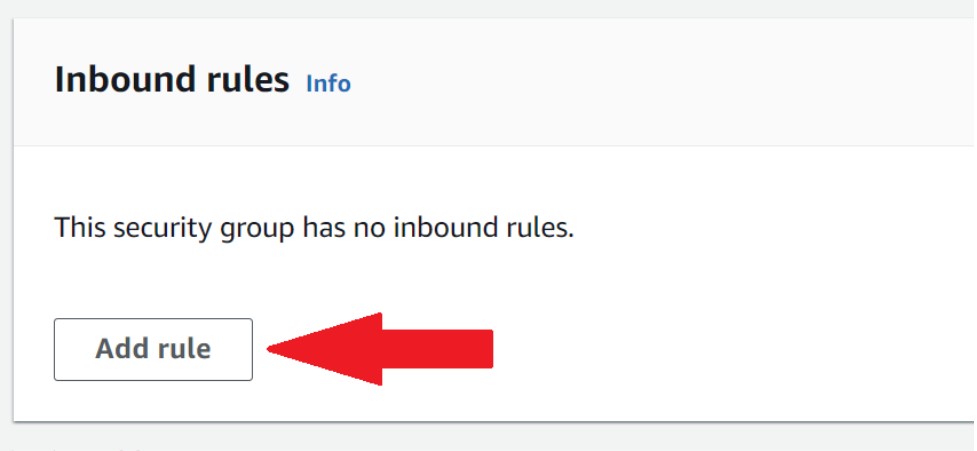
Click “Create security group”.



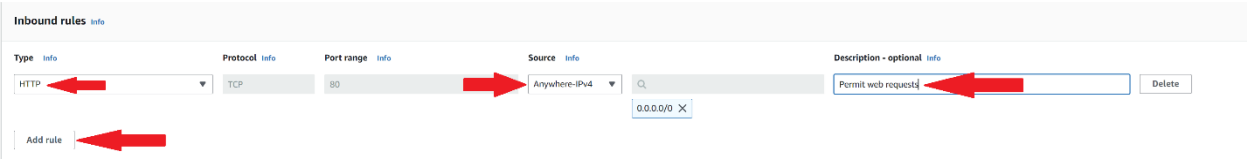
Under “Basic details”, set the name to “Web Security Group”, the description to “Enable HTTP Access”, and the VPC to “lab-vpc”.



Under “Inbound rules”, click “Add rule”.

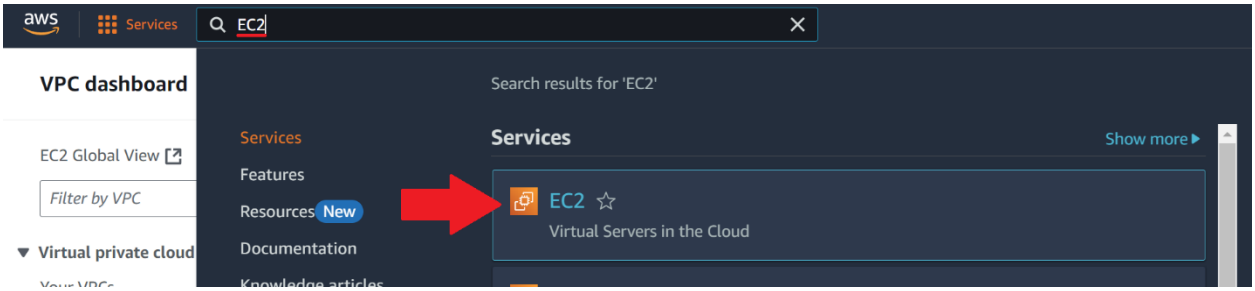


Set the rule type to “HTTP”, the source to “Anywhere-IPv4”, and the description to “Permit web requests”. Click “Add rule”.

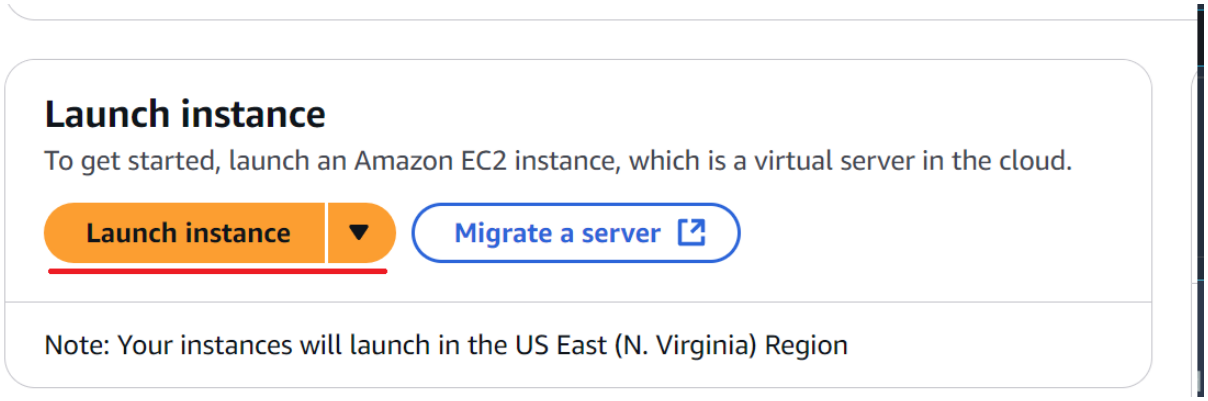


Click “Create security group”.

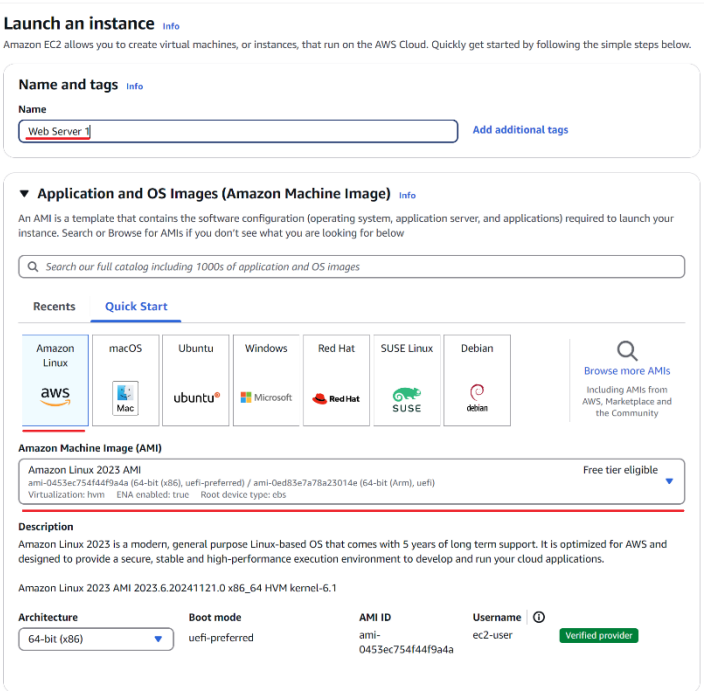
Next, we will create a web server with this security group applied. Open the EC2 dashboard from the search bar.



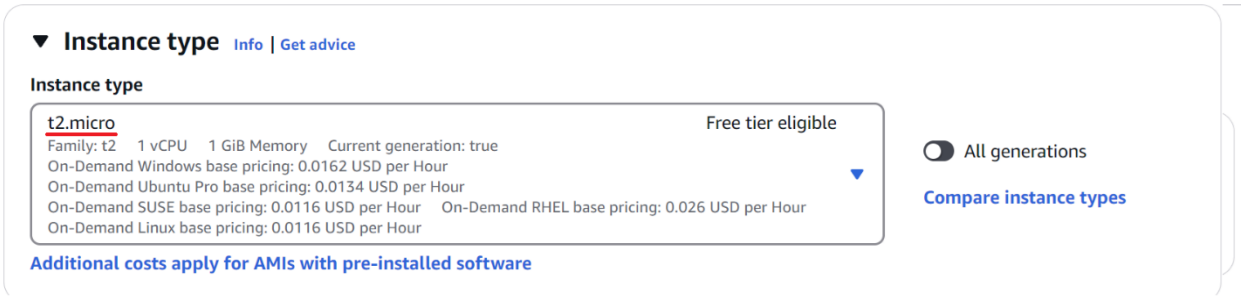
Click “Launch instance”.



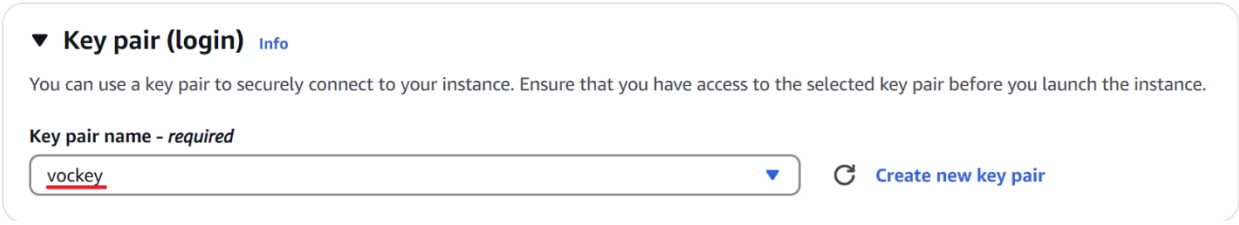
Set the name to “Web Server 1” and set the Amazon Machine Image to “Amazon Linux 2023 AMI”.



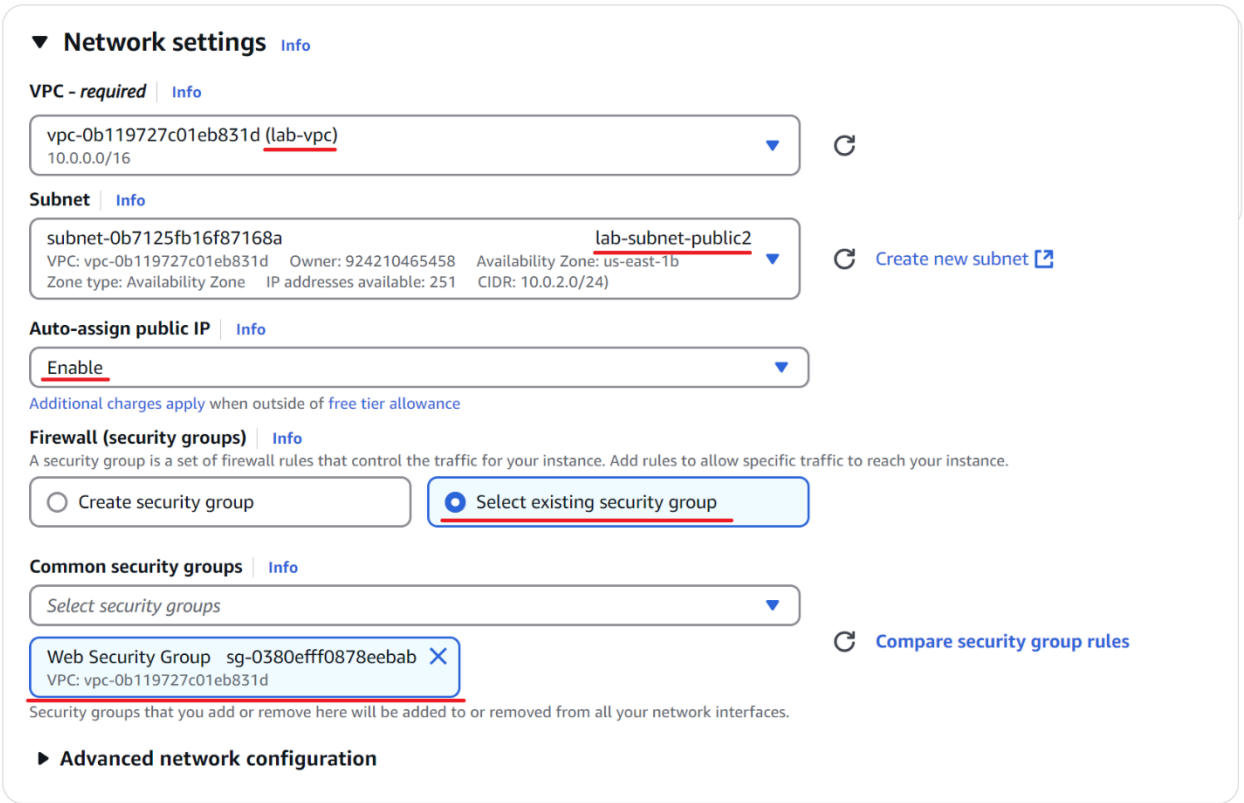
Set the instance type to “t2.micro”.



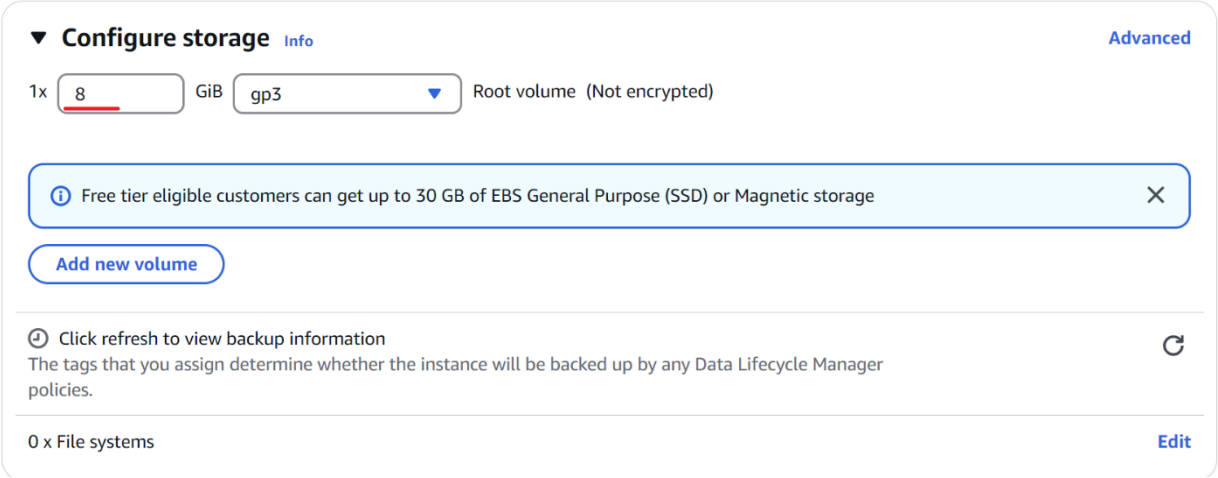
Set the Key pair to “vockey”.



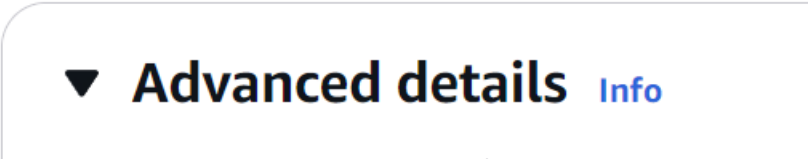
Under “Network settings”, set the VPC to “lab-vpc”, the subnet to “lab-subnet-public2”, and “Auto-assign public IP” to “Enable”. Set the “Firewall (security groups)” to “Select existing security groups” and add the “Web Security Group” created earlier.



Under “Configure storage”, set the storage to 8 Gibibytes.



Open the “Advanced details” tab.

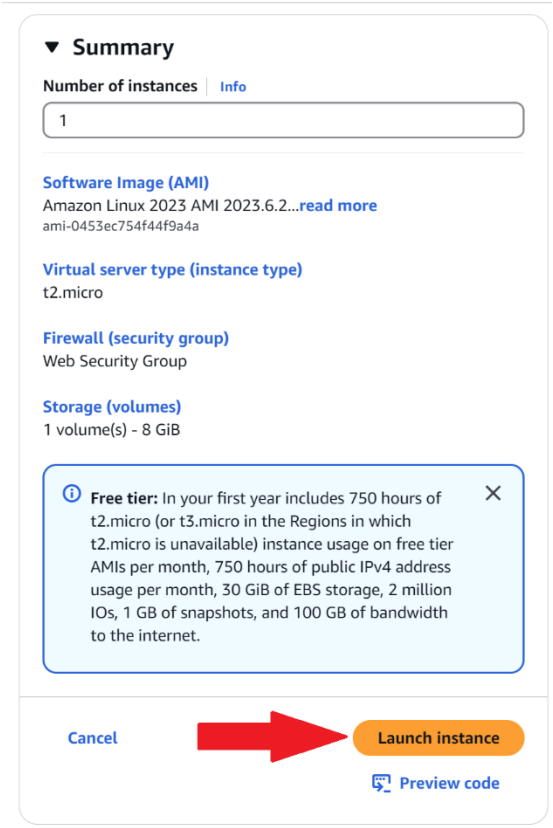


Scroll down to the “User data” section.

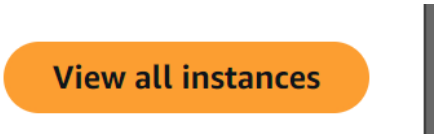
Add the following script:

```
#!/bin/bash
# Install Apache Web Server and PHP
dnf install -y httpd wget php mariadb105-server
# Download Lab files
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-ACCLFO-2/2-lab2-vpc/s3/lab-app.zip
unzip lab-app.zip -d /var/www/html/
# Turn on web server
chkconfig httpd on
service httpd start
```

Click “Launch Instance”.

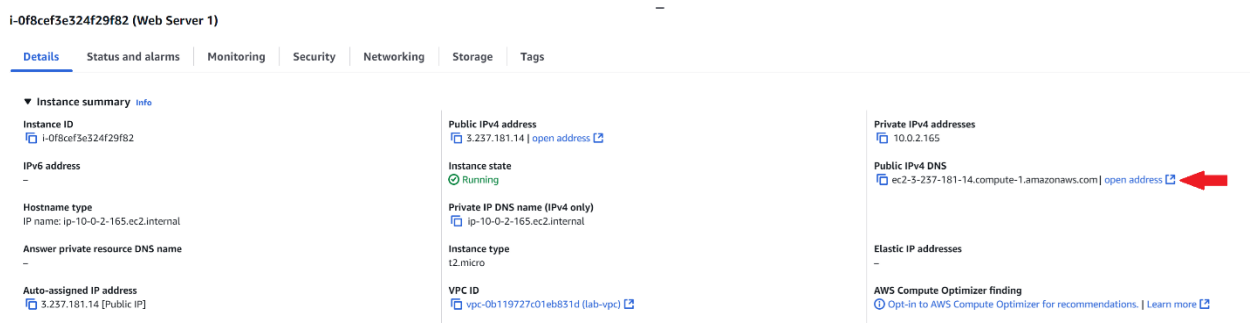


Click “View All Instances”.

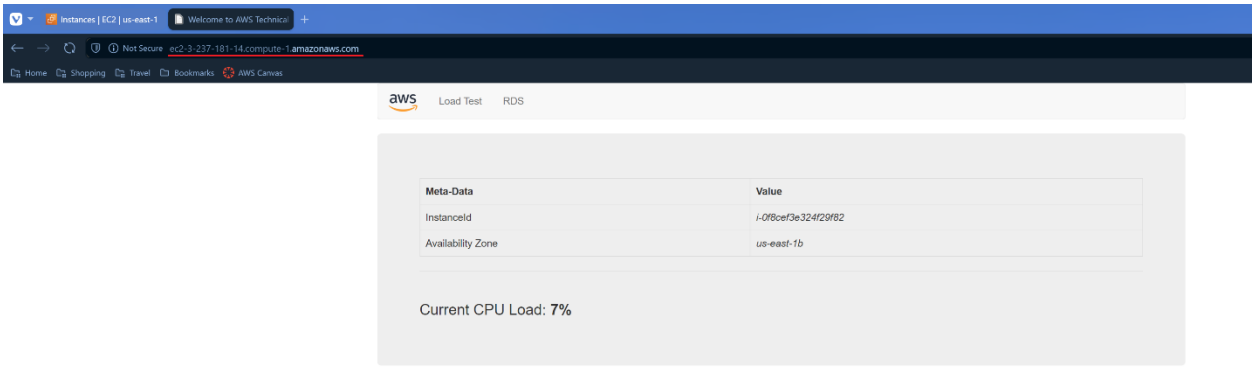


Click the checkbox next to “Web Server 1”.

After the “Status Check” section says “2/2 checks passed”, look for the public IPv4 address of the server and click the button to open the URL in a new tab:

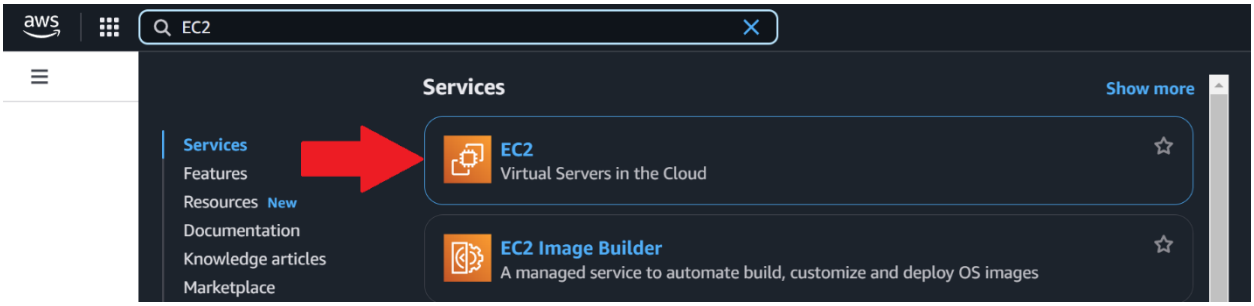


If done correctly, a page like this should open:

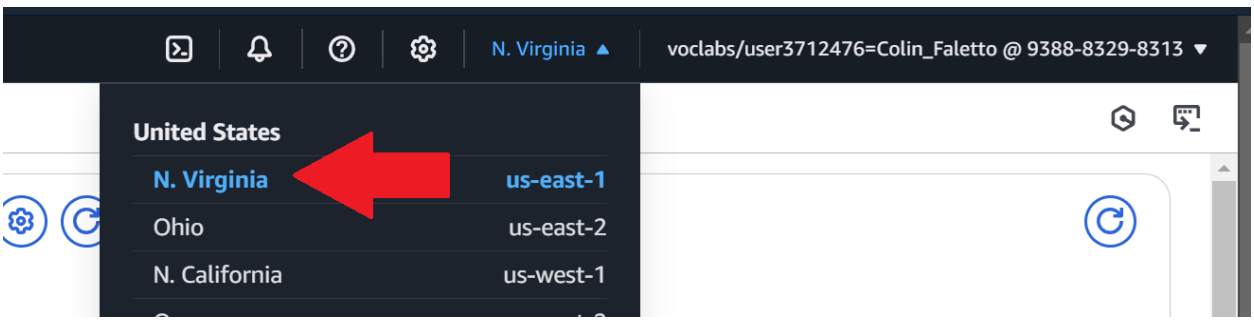


Lab Commands (Lab 3 EC2)

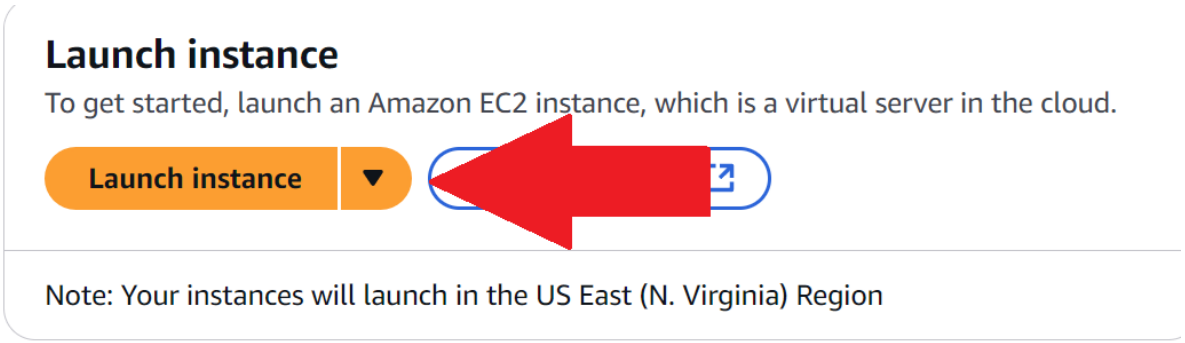
Open the EC2 console from the search bar.



Ensure you have the North Virginia region selected.

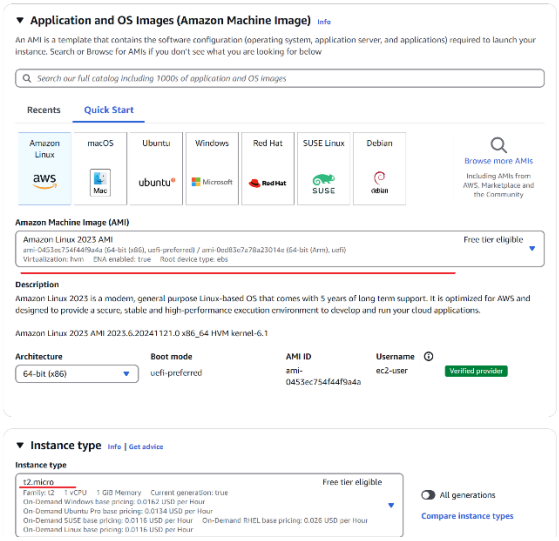


From the EC2 homepage, select “Launch Instance”.

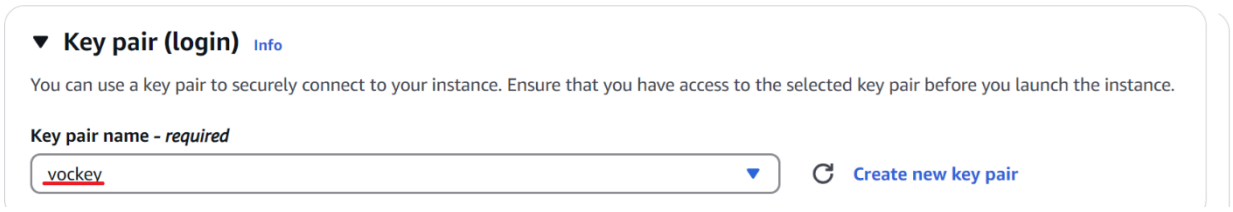


Name this instance “Web Server”.

Set the AMI to “Amazon Linux 2023 AMI” and set the instance type to “t2.micro”.

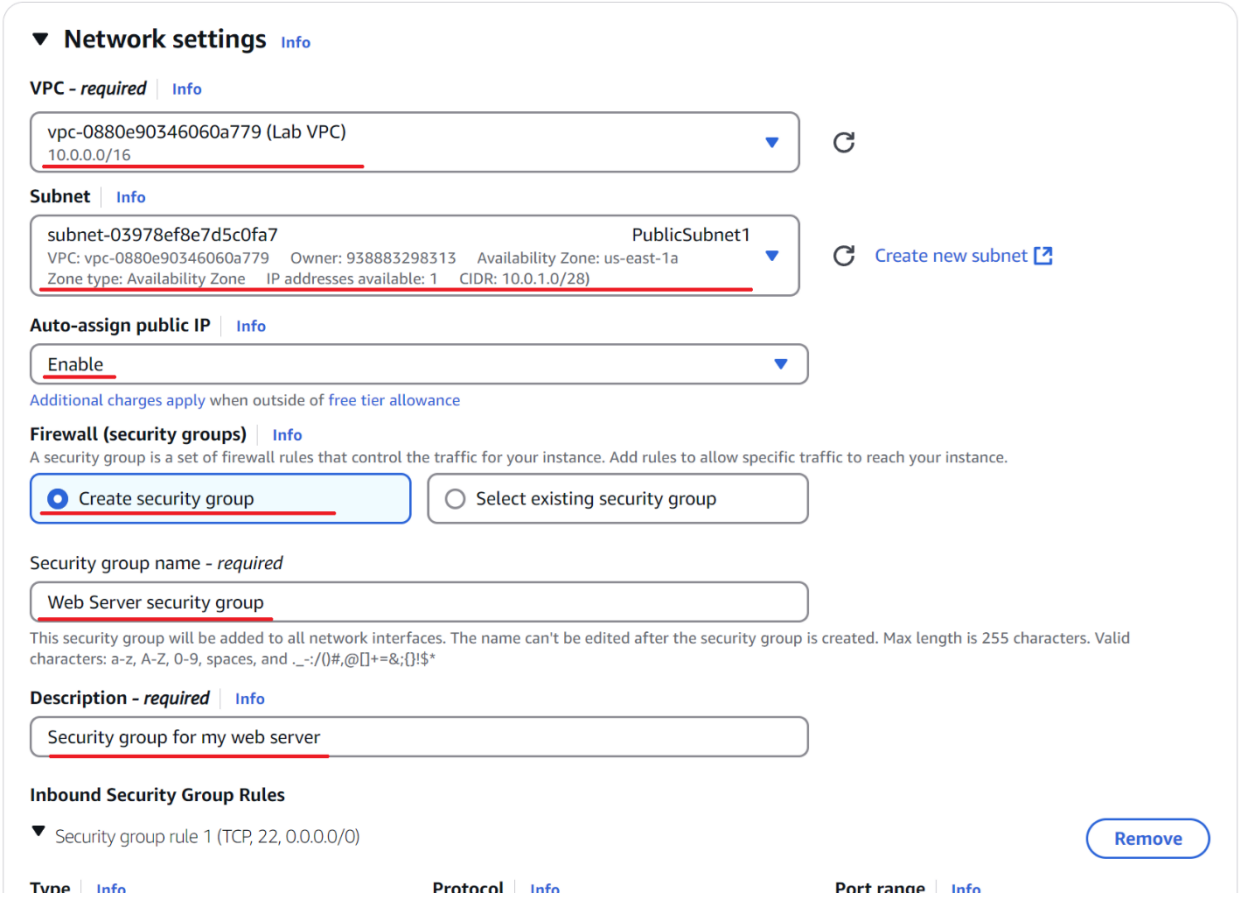


Set the key pair to “vockey”.

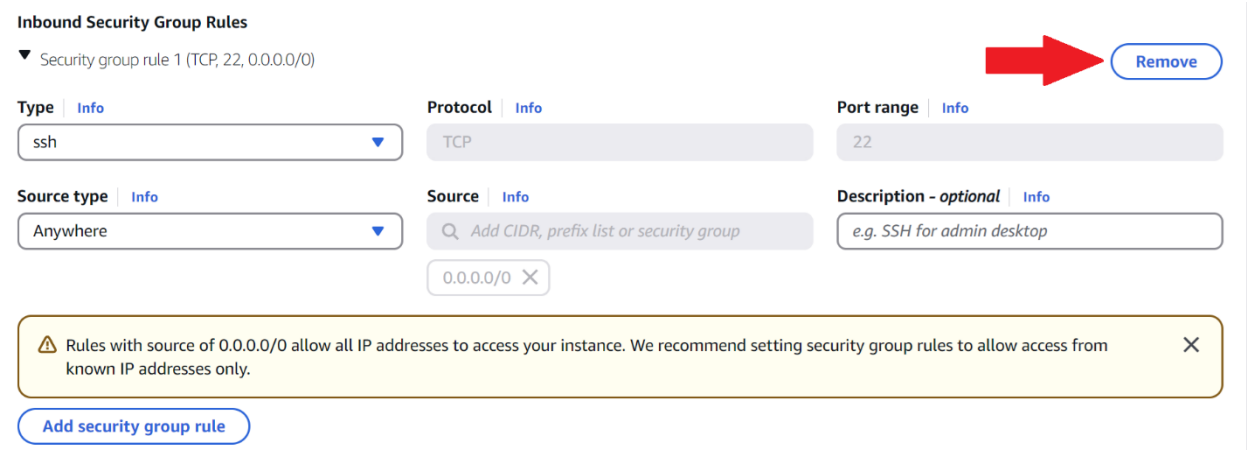


Under “Network Settings”, click “Edit”.

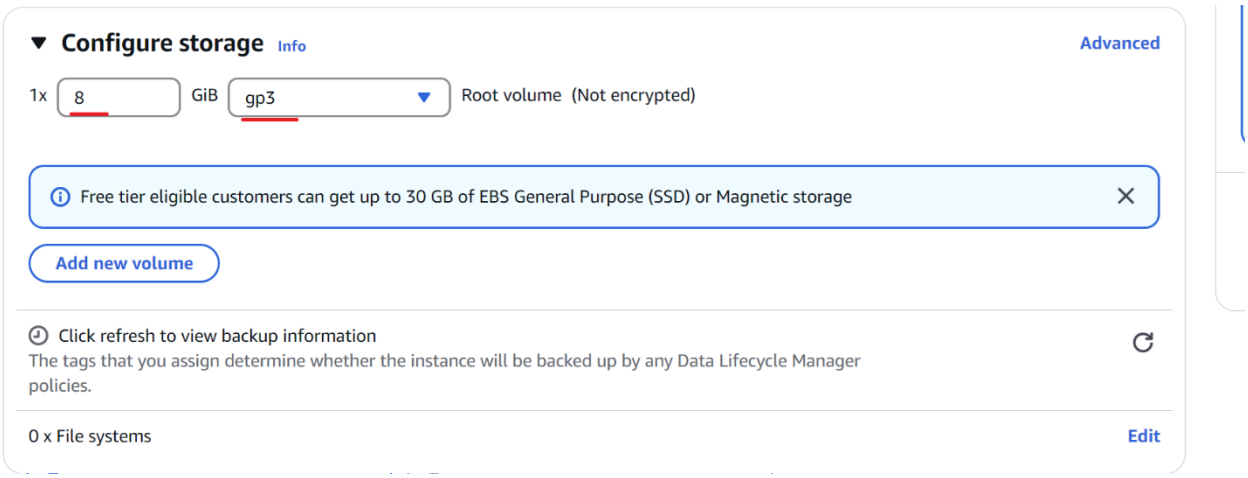
Set the VPC to “Lab VPC”, the subnet to “Public Subnet 1”, and the “Auto-Assign Public IP” setting to “enable”. Create a new security group with the name “Web Server security group” and the description “Security group for my web server”.



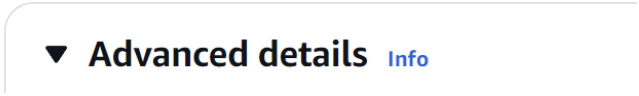
Make sure to remove “Security group rule 1”.



Under “Configure storage”, select 8 GiB of gp3.



Go the Advanced Details section.

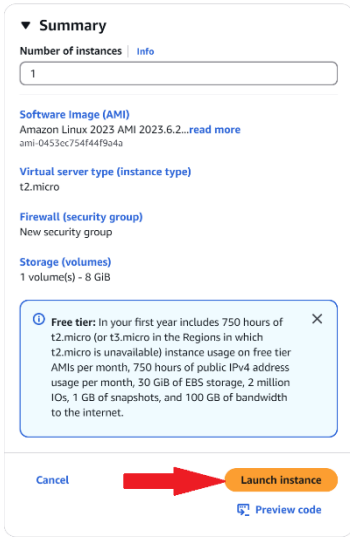


Enable termination protection.

Enter the following into the “User Data” section:

```
#!/bin/bash
dnf install -y httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Hello From Your Web Server!</h1></html>' >
/var/www/html/index.html
```

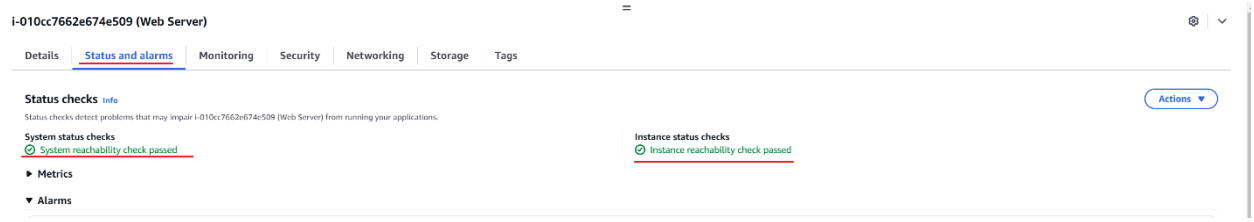
Under the “Summary” section, click “Launch Instance”.



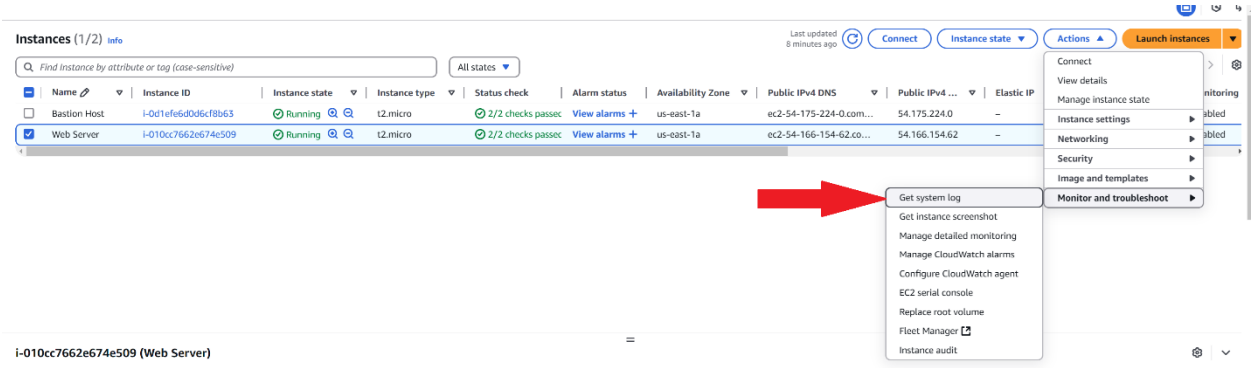
After launching the EC2 instance, click “View all instances”.

Wait for your web server to have the instance state of “Running”, then click the checkbox next to it.

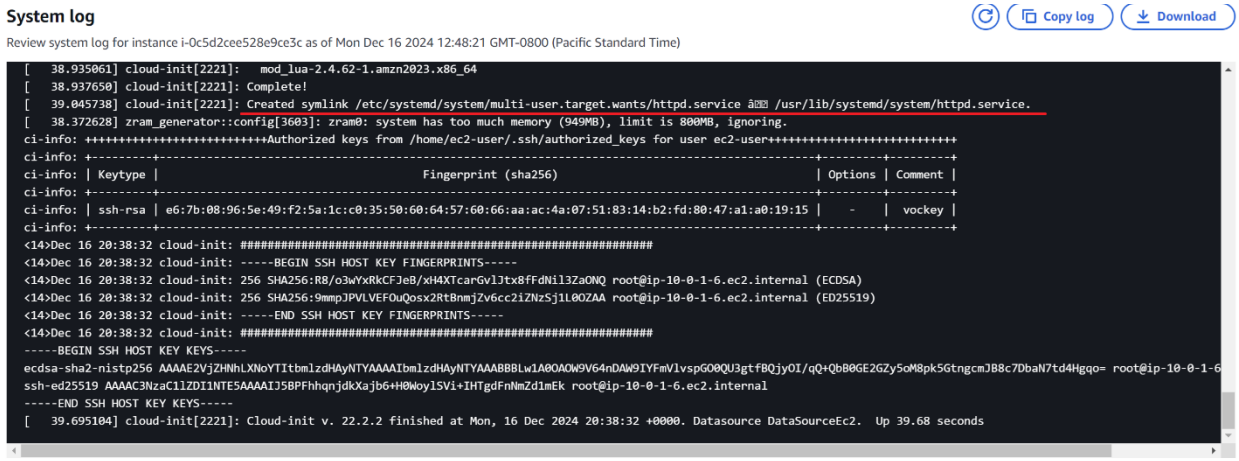
Under the “Status and Alarms” tab, ensure that both reachability checks have passed.



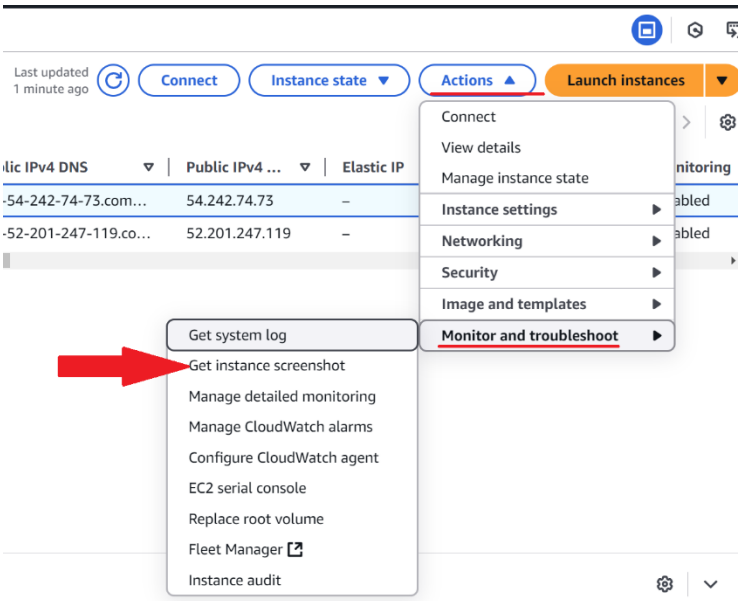
Click “Actions > Monitor and troubleshoot > Get system log”.



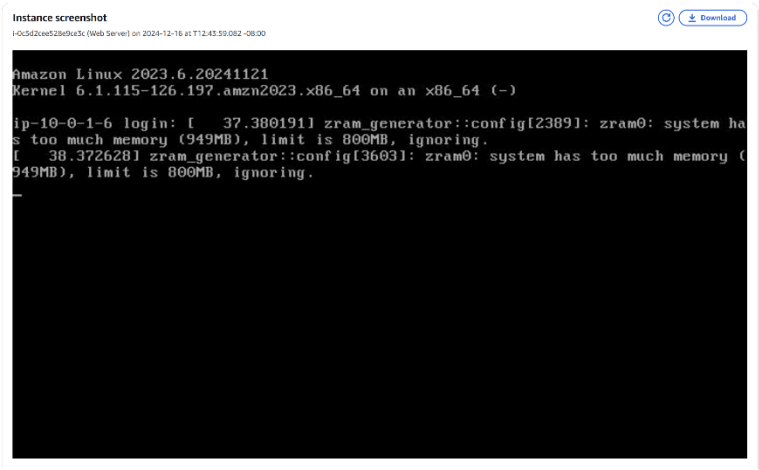
If done correctly, you should see a reference to the “httpd” service started in the User Data section.



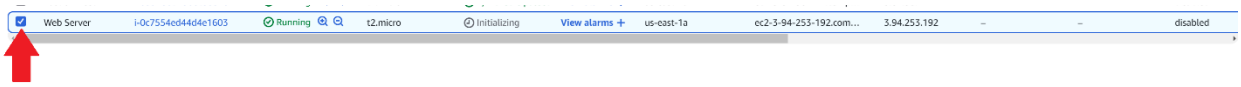
Return to the instances page and click Actrions > Monitor and troubleshoot > Get instance screenshot.

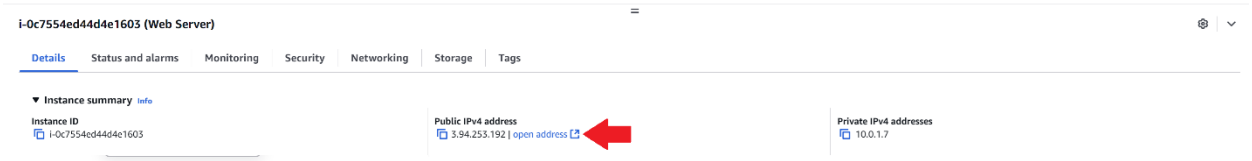


You should see a screenshot similar to this:

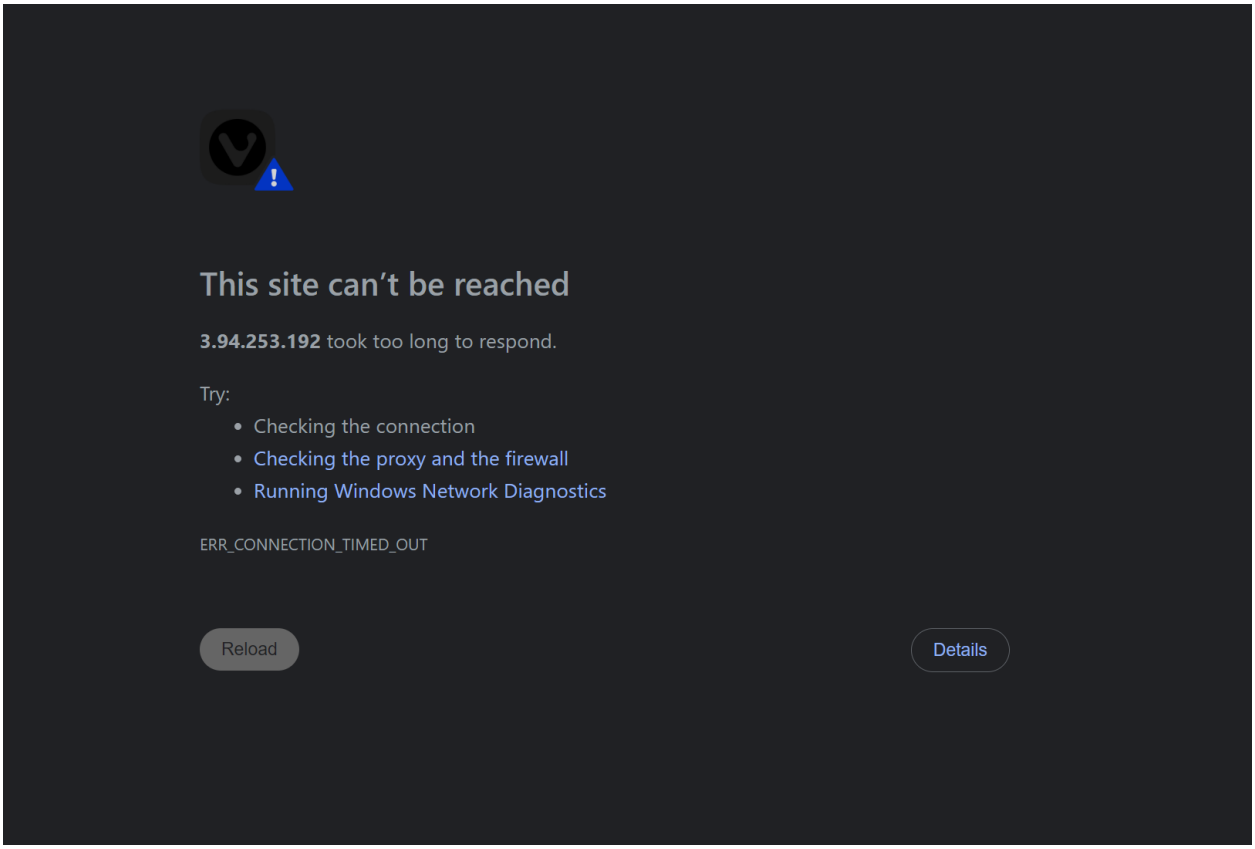


On the instances page, select Web Server and click open address under the Public IPv4 address field.

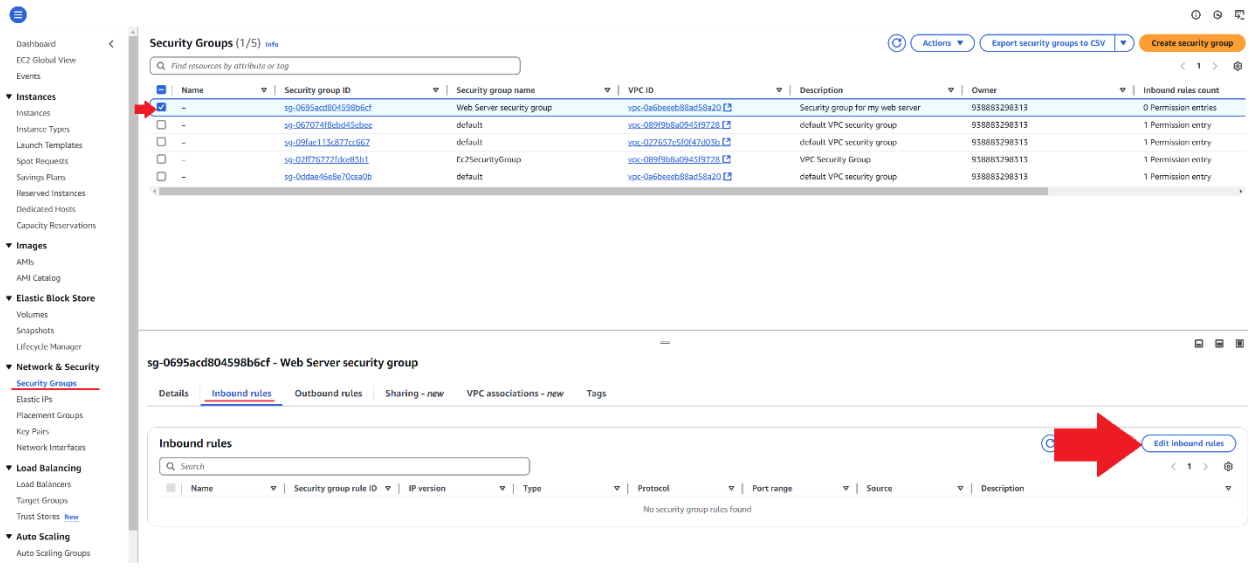




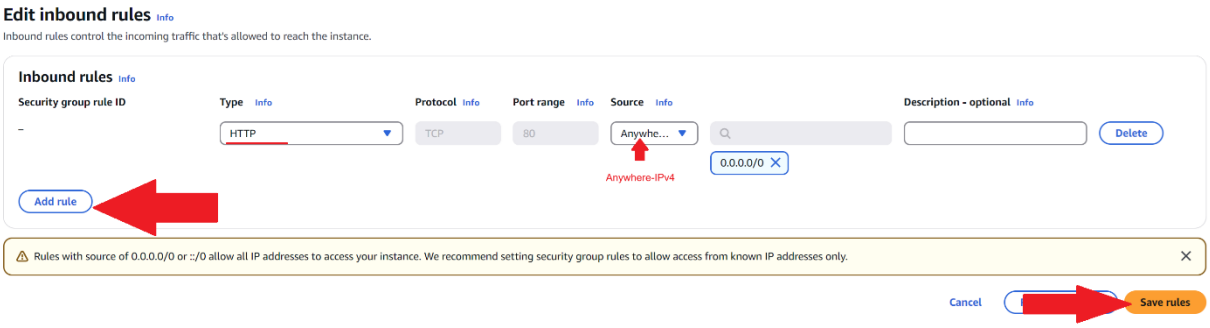
You should see an error message. This is because the security group for the instance has not yet been configured to permit HTTP access.



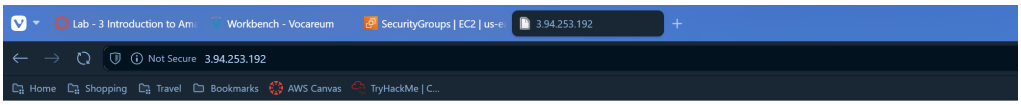
Next, you will configure a security group for the EC2 instance. Go to Security Groups > Select Web server security group > Inbound rules > Edit inbound rules.



Click Add rule, set the type to HTTP, the source to Anywhere-IPv4, then click Save rules.

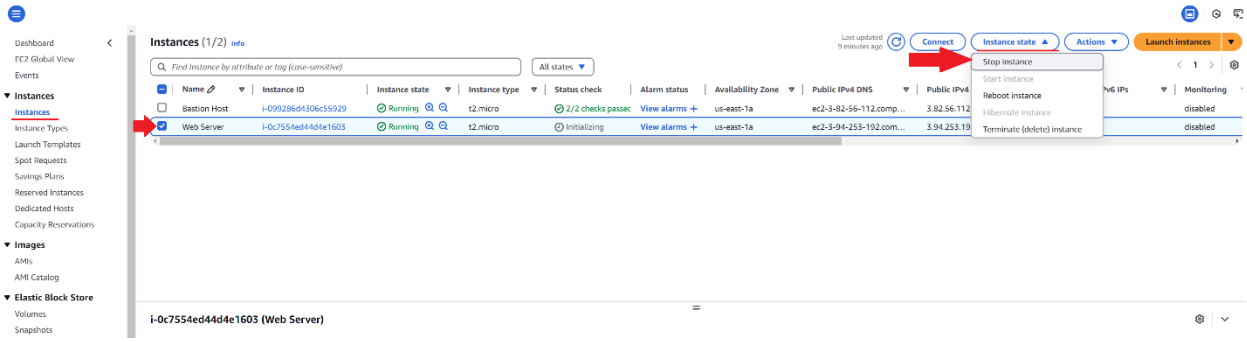


Refresh the web server page. You should see the following message:

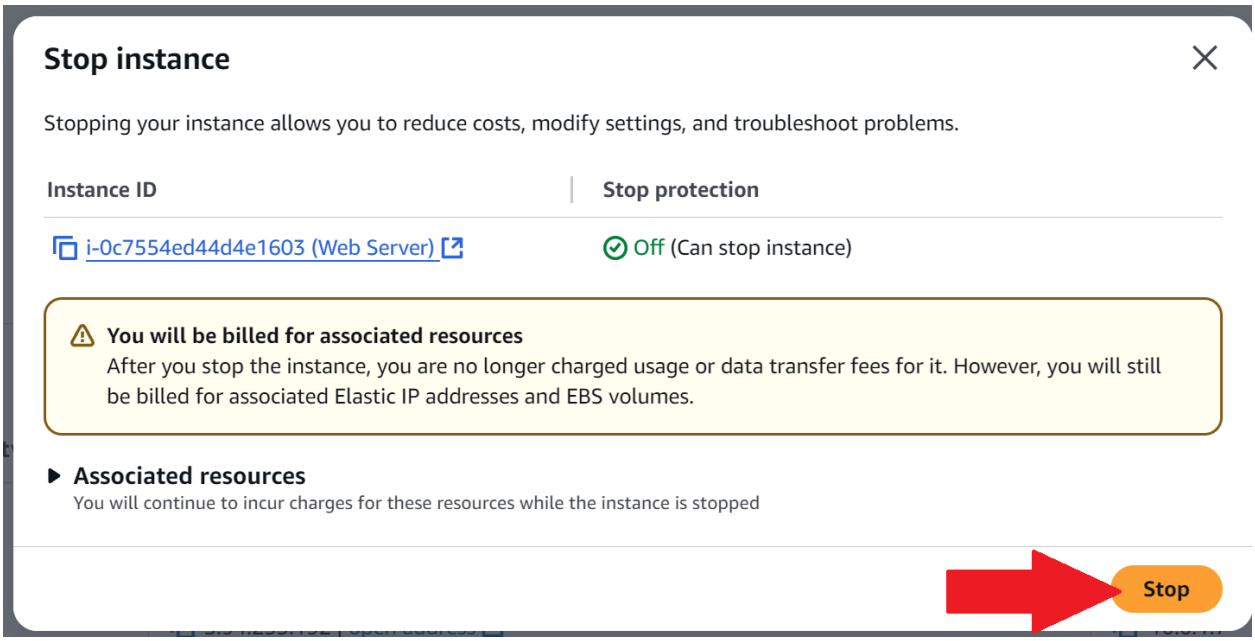


Hello From Your Web Server!

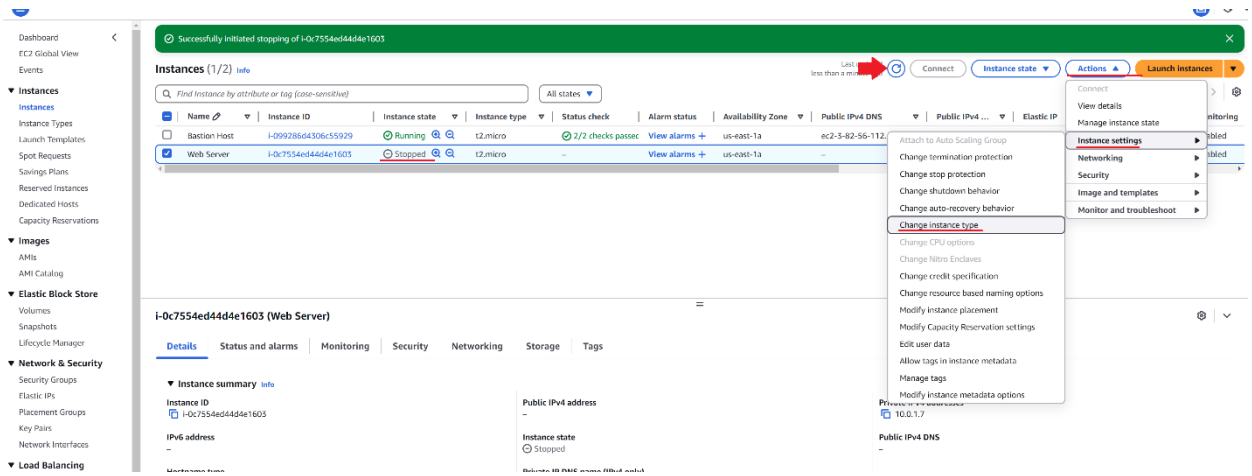
Next, you will resize the instance. Go to Instances > Select Web Server > Instance state > Stop instance.



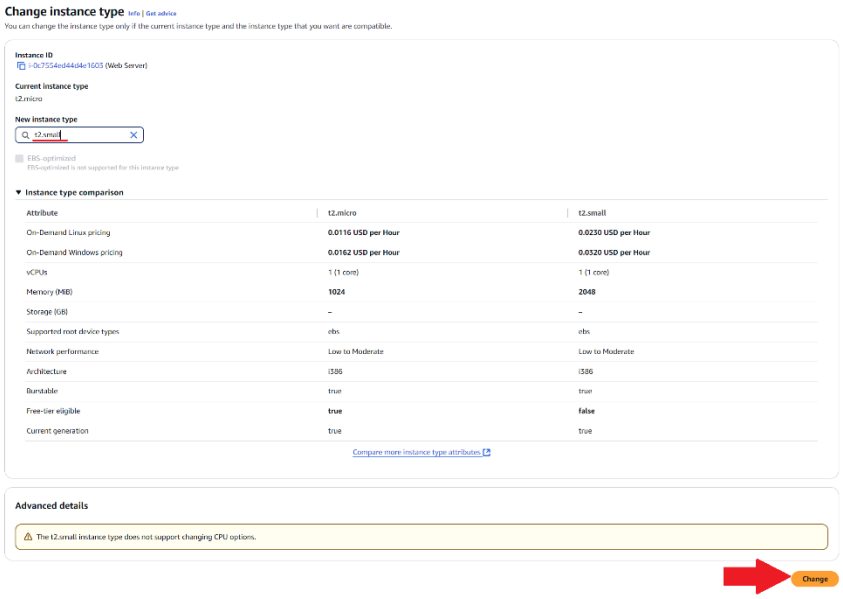
Click Stop.



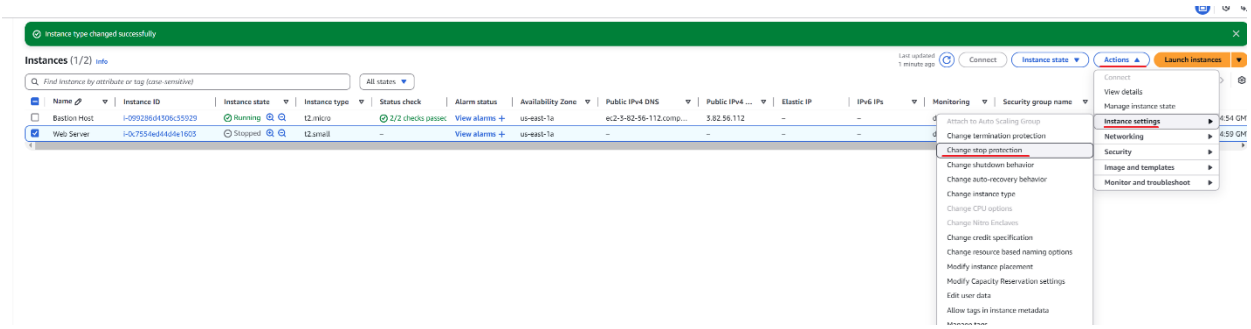
Click the refresh icon until the instance state shows Stopped, then go to Actions > Instance settings > Change instance type.



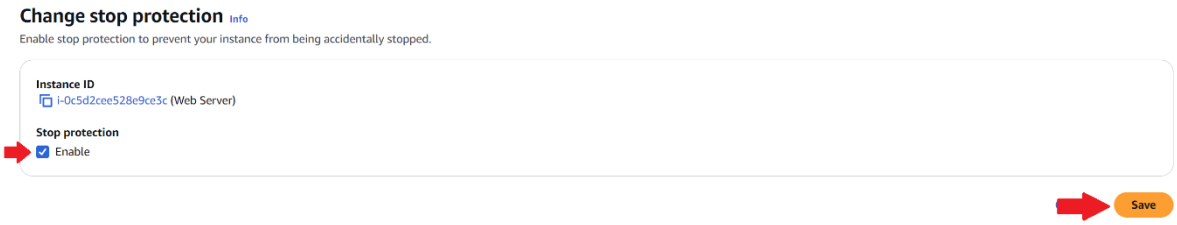
Set the instance type to t2.small, then click Change.



Go to Actions > Instance settings > Change stop protection.

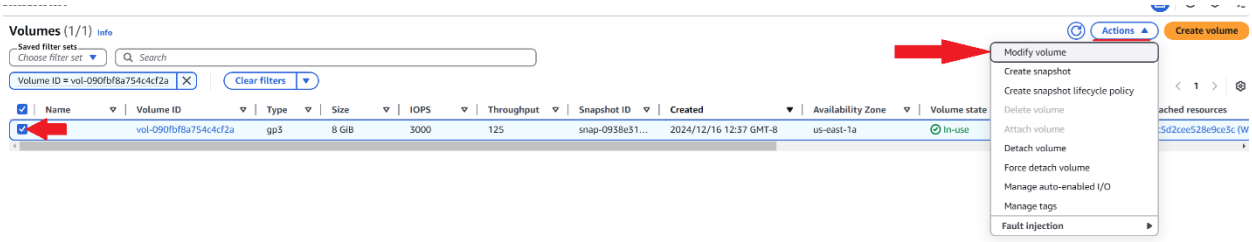


Click Enable, then click save.



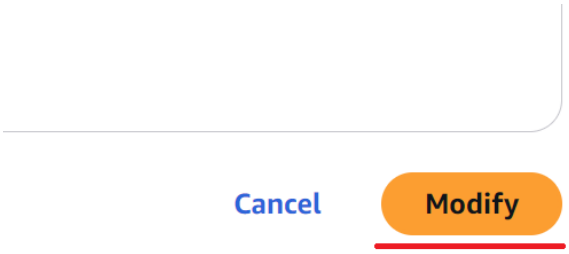
Next, go to the web server's Storage tab..

Select the volume created earlier, then click Actions > Modify volume.

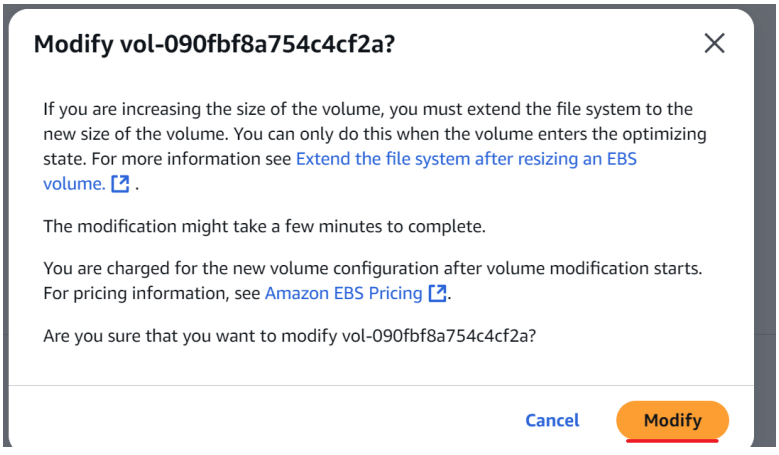


Change the size to 10 GiB.

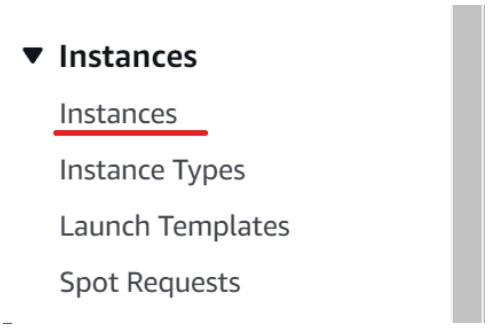
Click Modify.



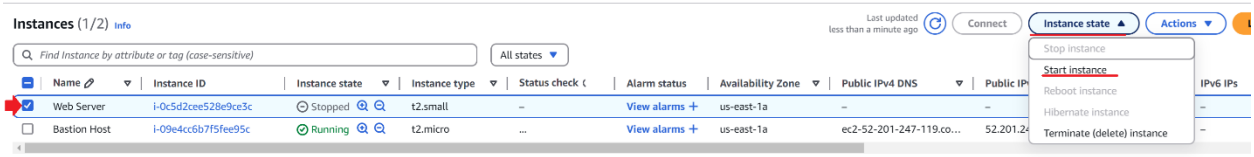
Click Modify again.



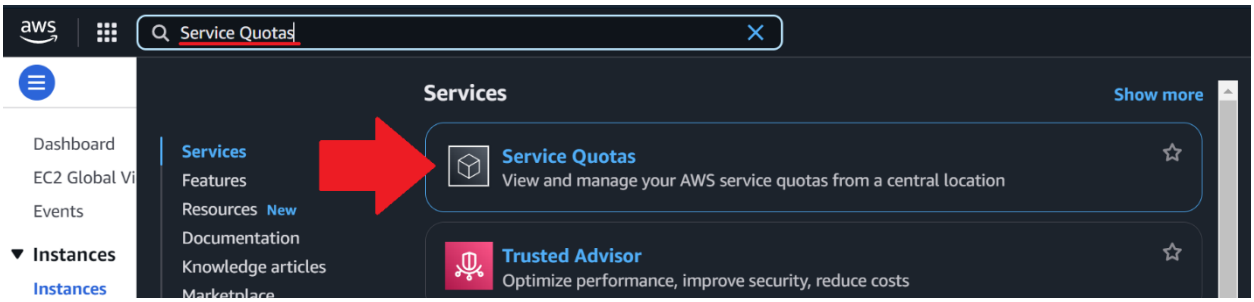
Return to the Instances section of the EC2 dashboard.



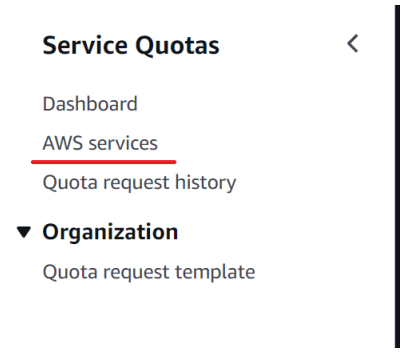
Select Web Server and click Instance state > Start instance.



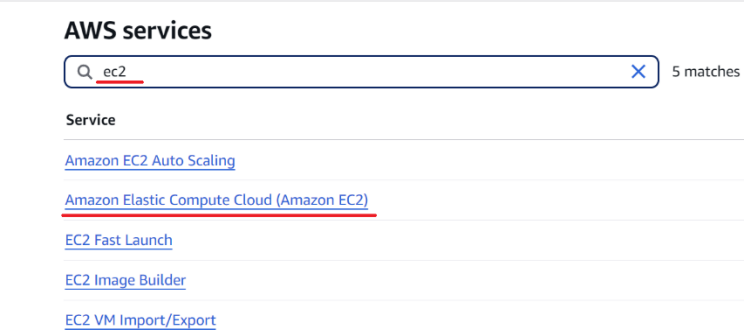
While the EC2 instance is starting, go to the Service Quotas dashboard from the search bar.



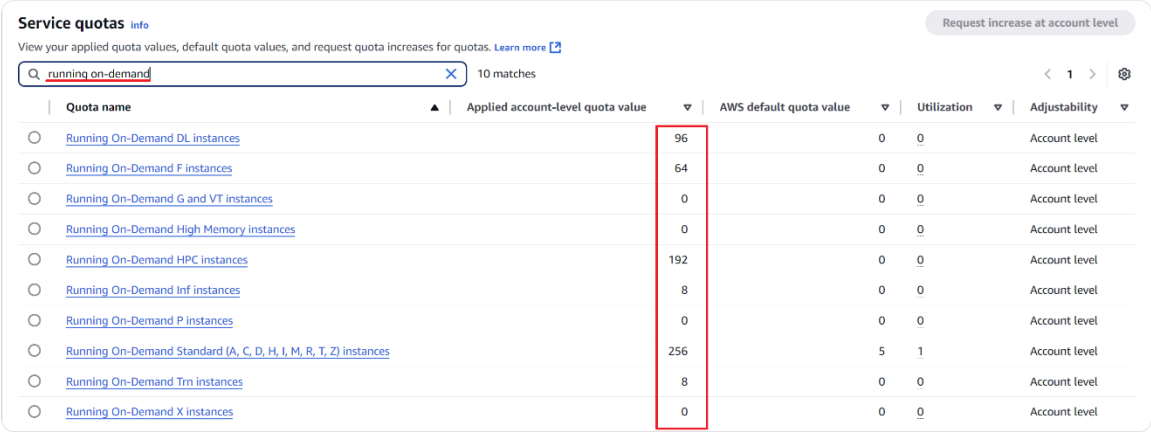
Select AWS services from the sidebar.



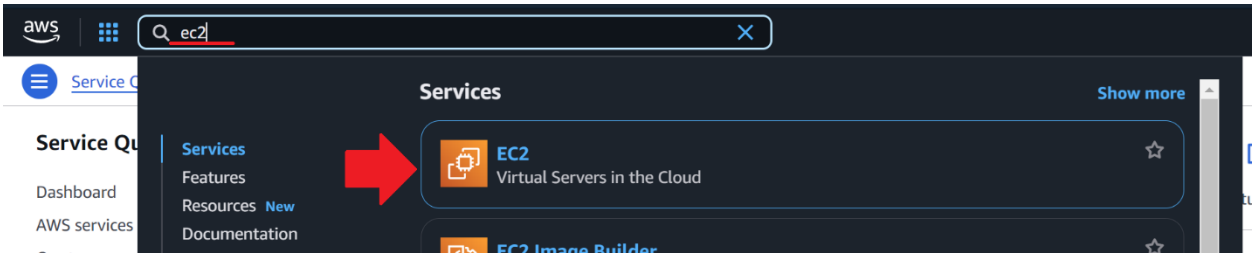
Search for and select Amazon Elastic Compute Cloud (Amazon EC2).



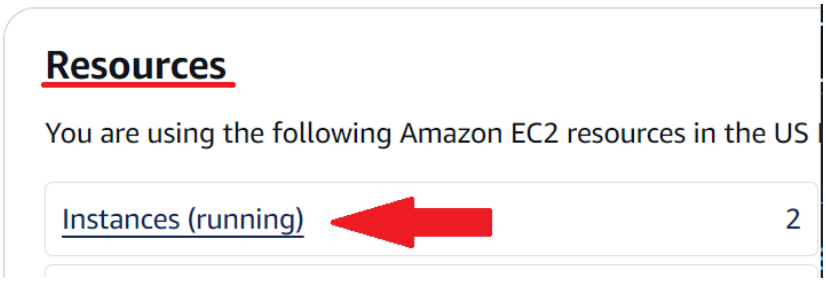
Search for running on-demand. Note the limits applied to how many concurrent EC2 instances can be run based on the instance type. Ensure that you do not surpass these limits while working with EC2.



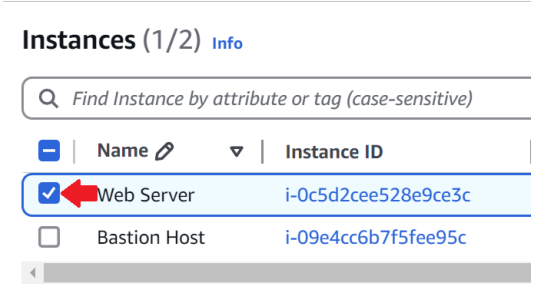
Return to the EC2 dashboard from the search bar.



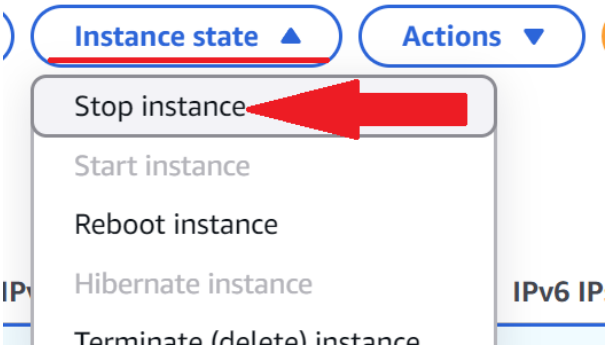
From the homepage, click Instances (running) under the Resources tab.



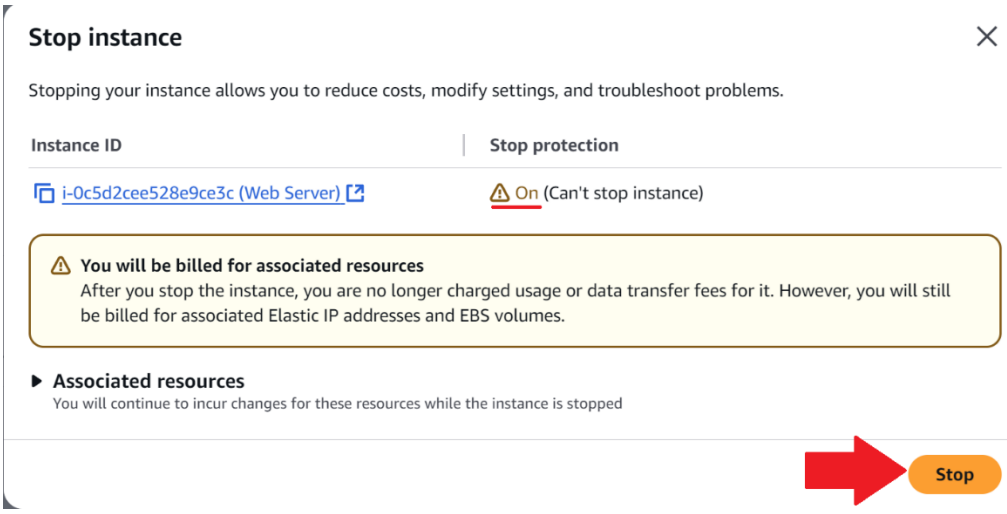
Select Web Server.



Select Instance state > Stop instance.

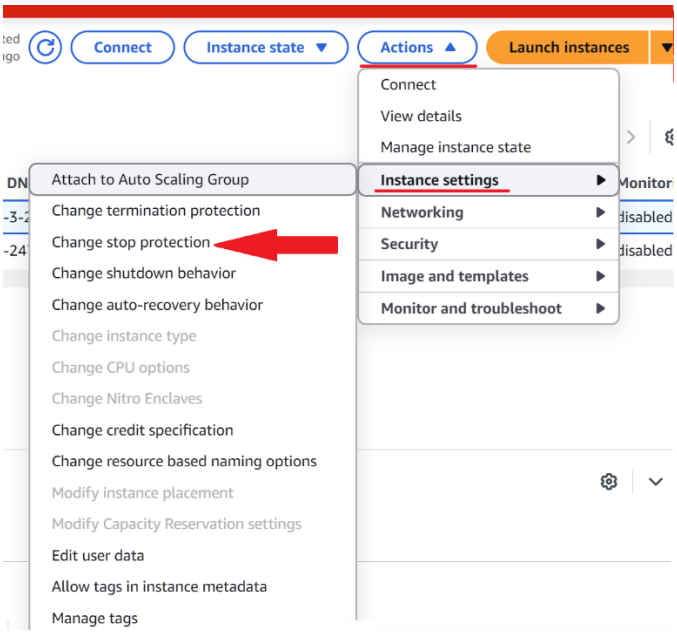


Click stop. Note the warning that stop protection is on.

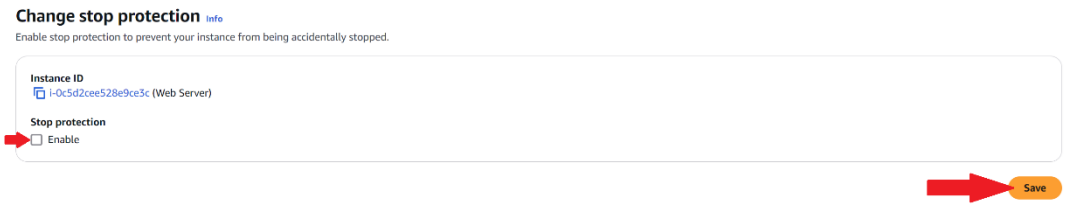


You will see the following error message.

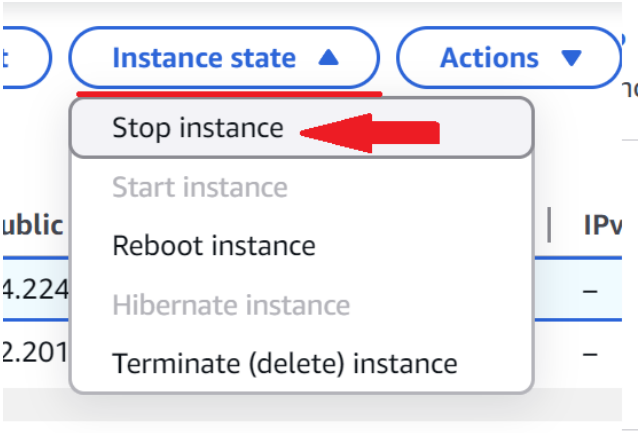
Next, you will disable stop protection. Go to Actions > Instance settings and click Change stop protection.



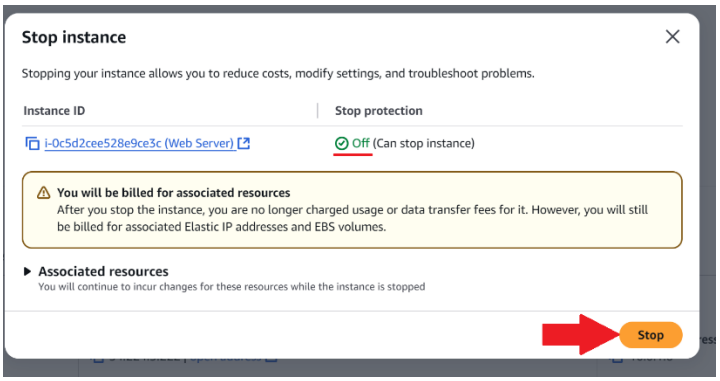
Uncheck Enable and click Save.



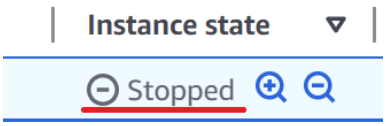
Now, click Instance state > Stop instance.



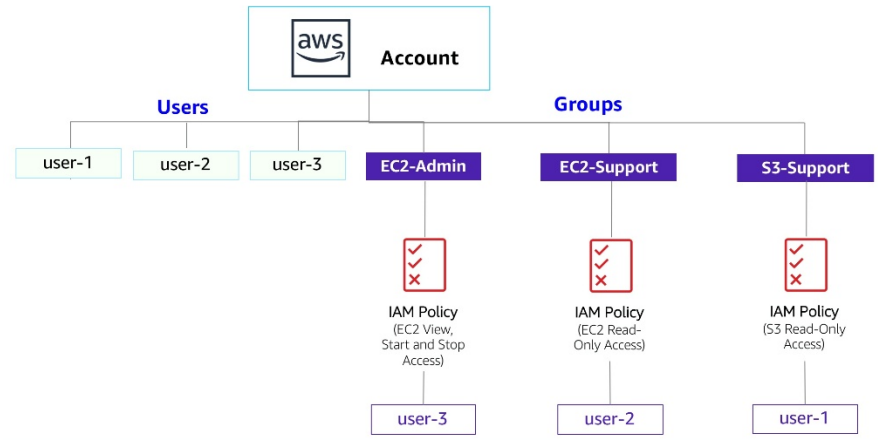
Click Stop. Note the confirmation that stop protection is now off.



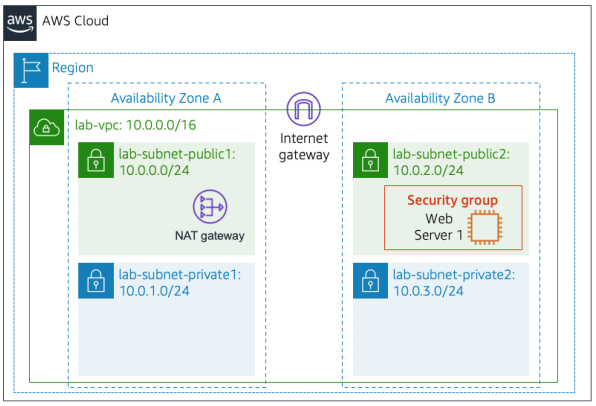
Wait a few minutes and refresh the Instances page. The web server's instance state should now show as Stopped.



Network Diagram (Lab 1 IAM)



Network Diagram (Lab 2 VPC)



Public Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	Internet gateway

Private Route Tables

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	NAT gateway

Network Diagram (Lab 3 EC2)

Problems

One problem I encountered was in Lab 3 (the EC2 lab), where even after correctly updating the Web server security group, the server's webpage refused to connect. This is because my web browser attempted to connect with HTTPS instead of HTTP. This was fixed simply by modifying the protocol in the URL to connect over HTTP.

Conclusion

To wrap up, this lab was a very useful introduction to the world of cloud computing. I'm grateful to have learned about exciting new cloud computing concepts such as complex security mechanisms, virtual routing through a private cloud, and easily scalable virtual machines. I'm now confident that I could help a company migrate basic services from traditional IT to the cloud, I'm hopeful that these foundational skills will help me secure a career in cloud services in the future.