**AWS ACADEMY LABS 1 - 3**

Austin Tran-Struthers

******

***Purpose***

To learn the basics of AWS by doing labs which focus on Identity Access Management (IAM), Virtual Private Cloud (VPC), and Elastic Compute Cloud (EC2)

**Lab 1 - Introduction to IAM:**

***Background***

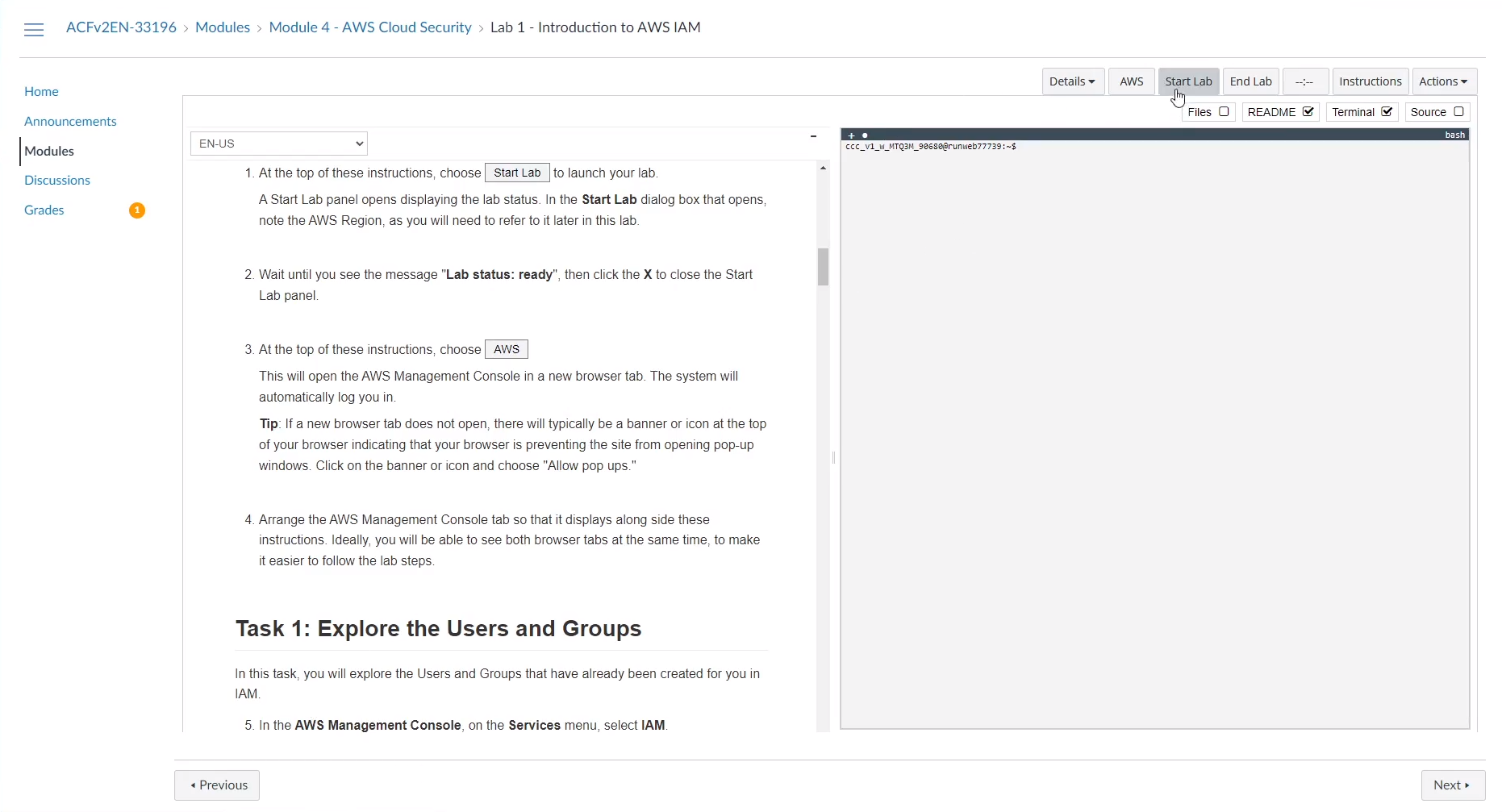
Amazon IAM (Identity and Access Management) is a service provided by AWS that enables users to manage access and permissions to AWS resources. IAM allows businesses to create and manage user accounts, assign specific permissions to those accounts, and control access to various AWS services and resources. It provides a centralized and secure way to manage user authentication and authorization, helping organizations maintain control over their AWS environments and data.

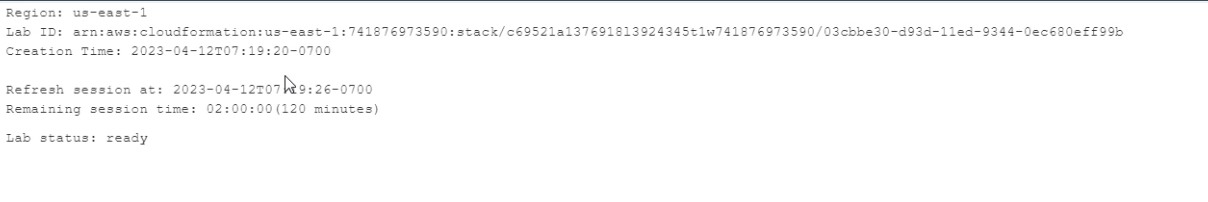
***Lab summary***

Investigating how IAM structures its users, user groups and policies applied to them, as well as how to navigate the IAM service. In addition, there is practice on moving users into user groups and testing their permissions.

***Starting up the lab***

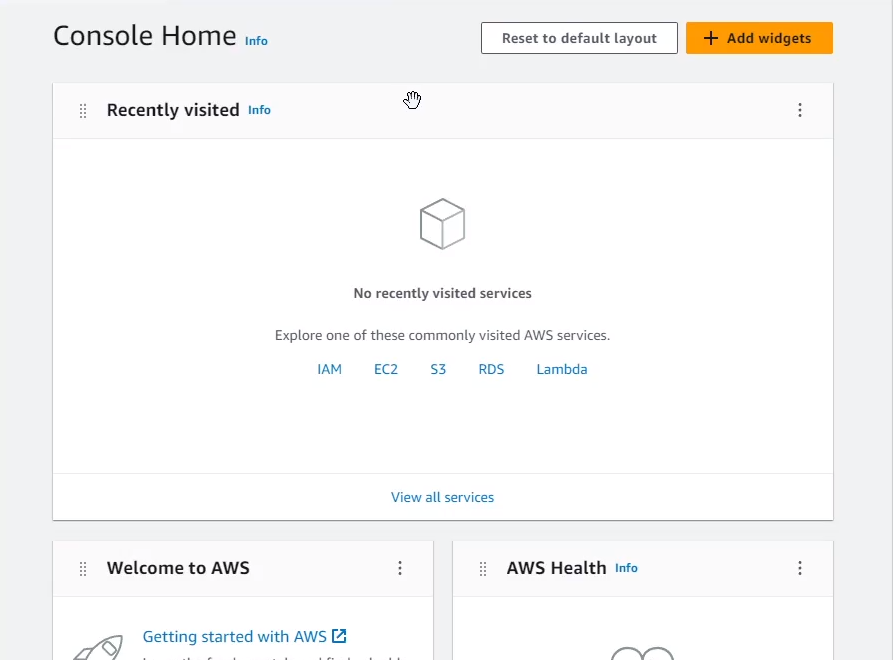
To begin you first need to start an AWS instance, the lab makes this simple enough by creating a simple Start Lab button near the top right of the starting page.



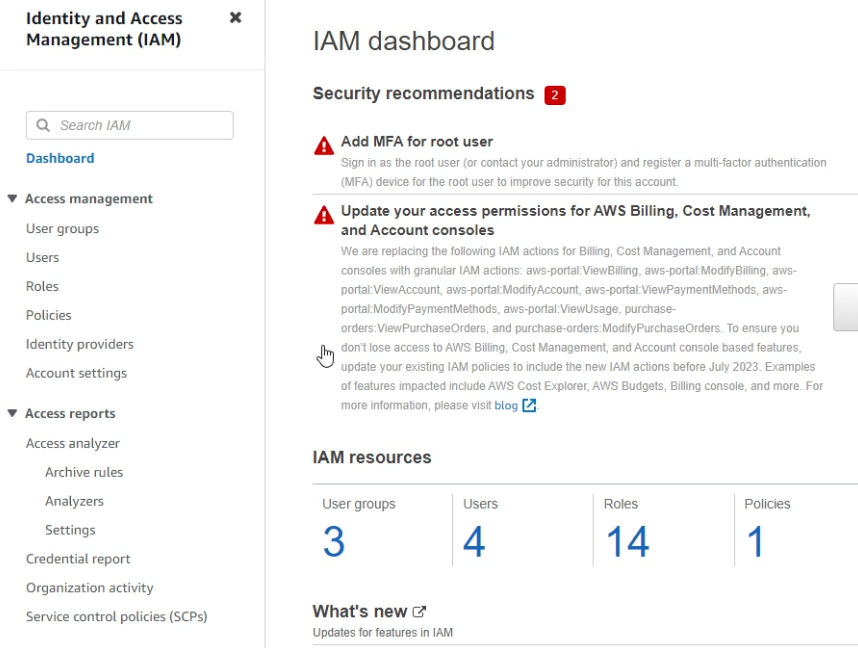
This will open up a pop-up which will tell you the region where the AWS instance starts, the lab’s creation time and how much time is remaining in the lab session.   
  
Wait until the lab status changes from “in creation” to “ready”, this took around 5 minutes.   
  
This means the AWS instance is ready to use and you can close the popup.

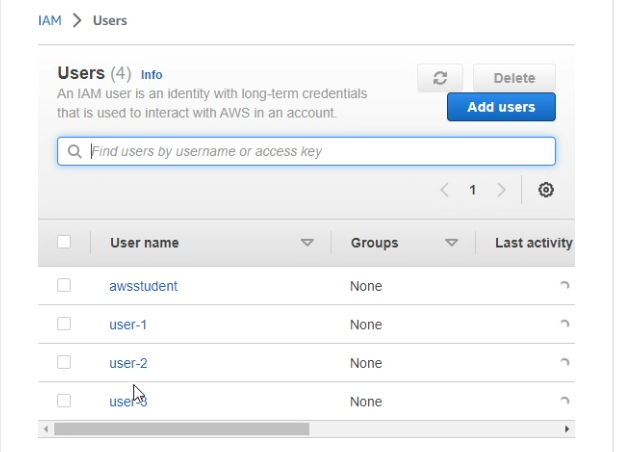
Pressing the AWS button to the left of the ‘Start Lab’ button will open the AWS management console and will allow you to begin the lab.

**An Investigation Into IAM**

  
When you arrive at the management console home page, you should see the screen above.

From here there are multiple ways to arrive at the IAM Management Console. You could press the hyperlink in the center of the page where the commonly visited AWS services are or use the Services drop down menu and search for the IAM service.

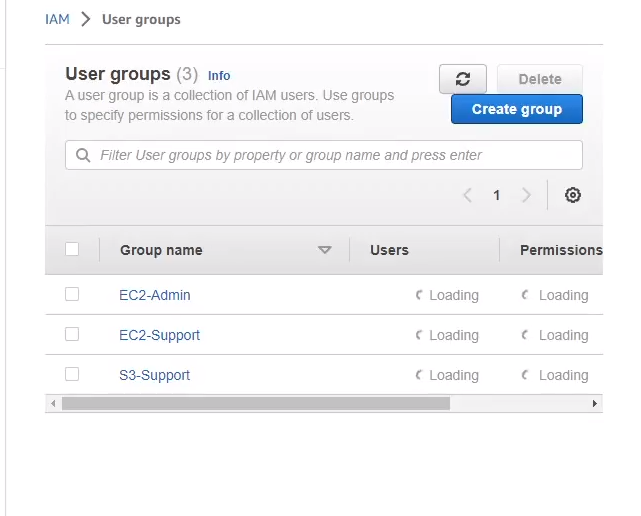
Using either of the methods, you should arrive at this, the IAM dashboard. It displays, among many things, the basic security recommendations for IAM and the currently active resources. In the case of this lab, you can ignore the security recommendations. In addition, this lab gives you pre existing users, roles, policies, and user groups that you explore and later use. For now, press the ‘Users’ option in the left menu under ‘Access Management’.



You should see 4 existing users in the menu.   
Select user-1.

By selecting the permissions, groups, and security credentials from the side-to-side menu you can view their respective details. However, user-1 does not have any existing permissions and is not part of any groups, although user-1 is given a console password.

Once you’re done viewing user-1, select the User groups option from the left navigation menu above where you chose users.

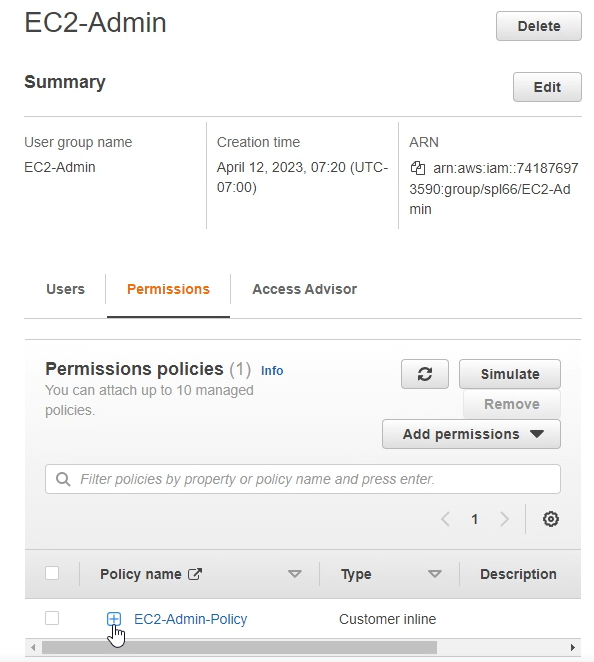


When you arrive at this page, there should be 3 existing user groups.  
Select the EC2 Support group, although this group has no users, it does have a permission policy which you can view by selecting the permissions tab from the menu.

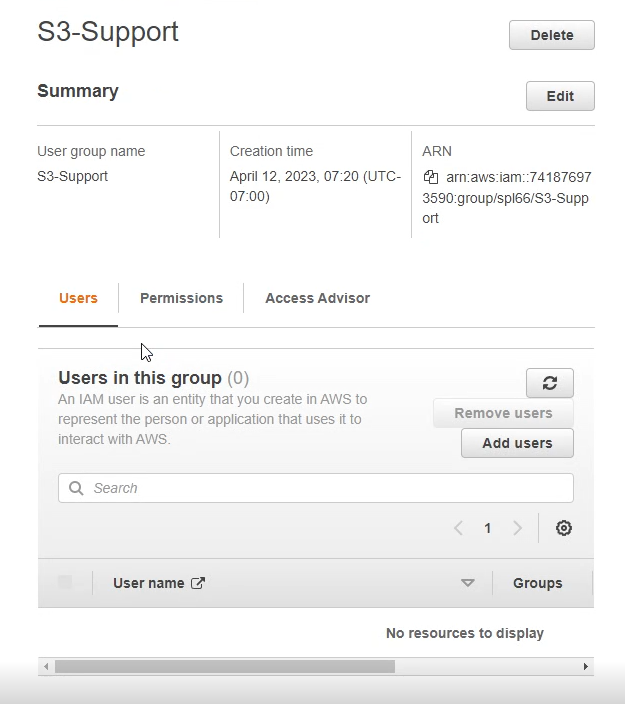
This permissions policy’s name tells you almost exactly what it does. This policy allows any users inside the group to only have the ability to view EC2 resources and not modify them.  
Pressing the + icon on the policy will show its details, specifically it will show its JSON file. The JSON file follows a simple format.  
**Effect** - whether to Allow or Deny the permissions

**Action** - specifies the API calls that can be used on an AWS Service  
**Resource** - which sets boundaries for what is covered by the policy

You can also view the S3-Support groups policy which is similar to the EC2-Support group in that it is read only.

  
Also look at the EC2-Admin policy which is different from the other two policies which are Managed Policies, this inline policy is meant to apply to just one User/Group and are for one-off situations such as administrator privileges. This policy still follows the JSON format previously established and allows the ability to stop and start EC2 instances as well as view their information.

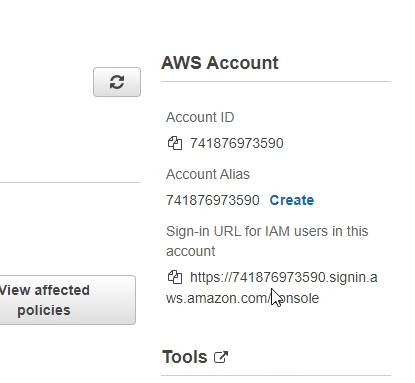
**Giving permissions to users by assigning them to groups**

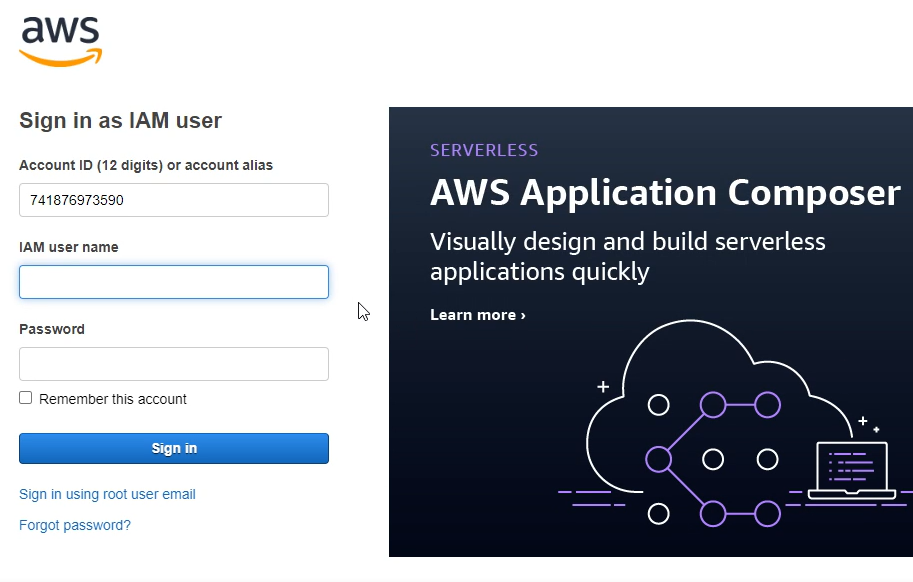
From the user groups menu, we can now assign each user to a different group, to do this select the group that you’re going to add a user to and hit the add users button near the middle of the screen, as shown below.

From here select the user you’re going to add and press add users. Repeat this process by putting user-1 into the S3-Support group, user-2 to the EC2-support group and user-3 into the EC2-admin group.

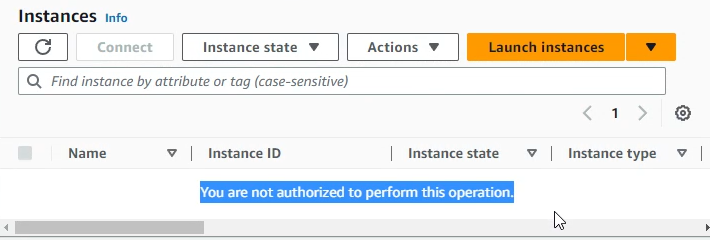
**Testing User Permissions**

Now we can test individual user permissions and ensure that each user only has access to what it needs and nothing else.

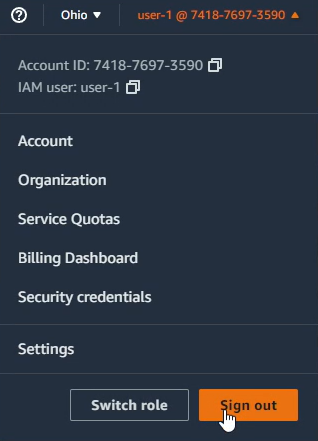
To do this go back to the dashboard and copy the url near the top right of the screen

This url is used to sign into the AWS account that is currently in use as well as its users.  
Access the link via a private window or a different browser.  


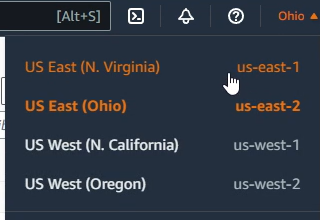
This is what the landing page looks like for the link that was copied, your account id may be different.  
  
Enter IAM username user-1 with password Lab-Password1 to test permissions for user-1.  
  
We can test if the user is able to view S3 buckets and their contents by entering the S3 service, as with IAM there are multiple methods of arriving at the S3 landing page.  
Then you can ensure that the user cannot access EC2 by arriving at the EC2 page from the services menu. You should see a message like the one below.



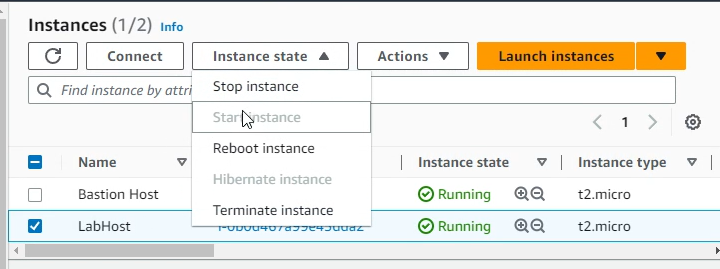
Sign out of the user-1 account by pressing the user dropdown near the top right of the screen.



Now sign into user-2 by using the username user-2 and the password Lab-Password2.  
  
Navigate to the EC2 landing page, you could use the services menu or look at the recently visited AWS services.  
If the EC2 landing page does not have any running instances, change your region to us-east-1 as shown below.



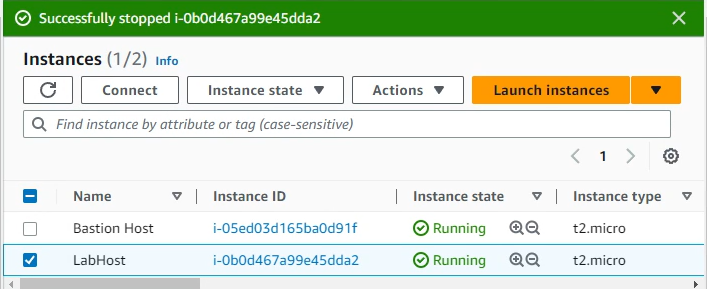
Now select the checkbox next to the LabHost instance and attempt to stop the instance by going to the instance state drop down menu and selecting Stop instance.



If permissions have been set correctly, you should see a long red message show up near the top of the screen telling you that user-2 cannot perform this action.  
Double check that user-2 isn’t able to see anything other than EC2 by attempting to look at the S3 buckets. Then sign out of user-2

And now for the final user, enter the username user-3 and the password Lab-Password3.

Navigate to the EC2 page and attempt to stop the instance. Since you are in the EC2-admin group, you should see the instance enter the stopping state or a successfully stopped message.



Make sure to sign out of user-3 and press the End Lab button next to the Start Lab button from before.

**Lab 2 -Building Virtual Private Clouds**

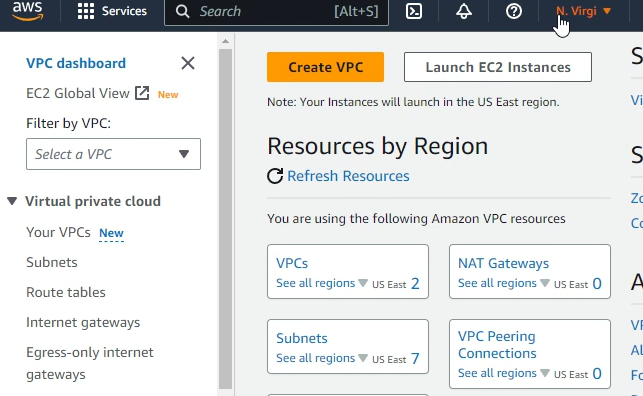
***Background***

Amazon VPC (Virtual Private Cloud) is a virtual network service offered by Amazon Web Services (AWS) that enables users to create a logically isolated section of the AWS cloud to launch resources and configure network settings. With VPC, users can define their own virtual network topology, including IP address ranges, subnets, routing tables, and network gateways, allowing them to have full control over their network environment within AWS.

***Lab summary***

In this lab you learn how to create a VPC with subnets in two Availability Zones which increases availability as well as a VPC security group that allows for HTTP traffic to flow to a web server created using an EC2 instance.

**Creating the VPC**

Once you enter the AWS Management Console page after starting the lab, use the services menu to arrive at the VPC Management Console. Ensure that your region is set to us-east-1 to avoid any variation. Select the orange Create VPC button from the VPC dashboard.

This is where you create the VPC, it is a very simple process

Choose the VPC and more setting  
Keep Auto-generated selected but change it to lab instead of project  
Keep the IPv4 CIDR block at 10.0.0.0/16  
Availability zones, and public and private subnets shout get set to 1  
Expand the Customize subnets CIDR blocks section

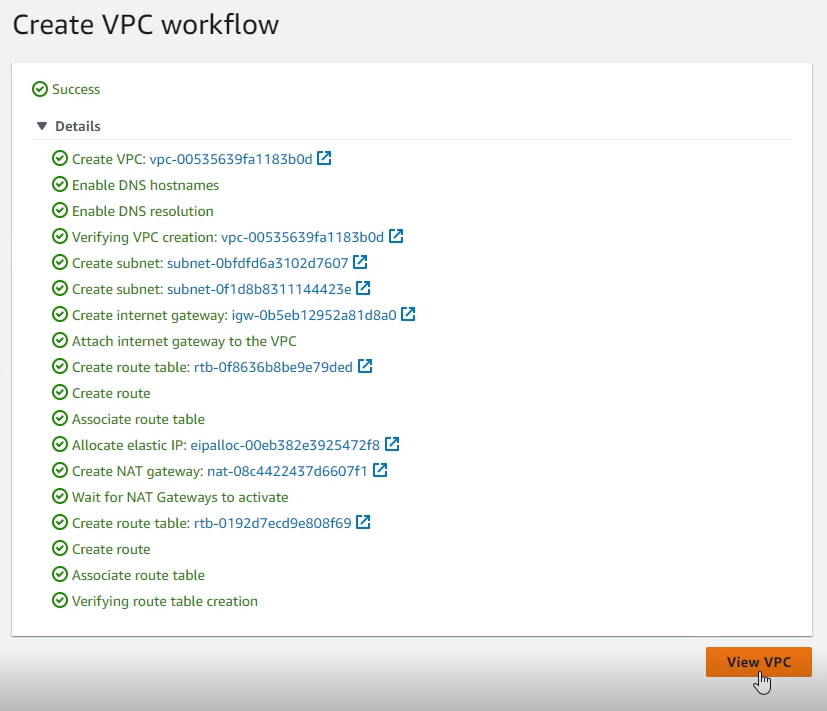
Change Public subnet CIDR block in us-east-1a to 10.0.0.0/24

Change Private subnet CIDR block in us-east-1a to 10.0.1.0/24

Set NAT gateways to In 1 AZ  
Set VPC endpoints to None  
Finally keep both DNS hostnames and resolution enabled

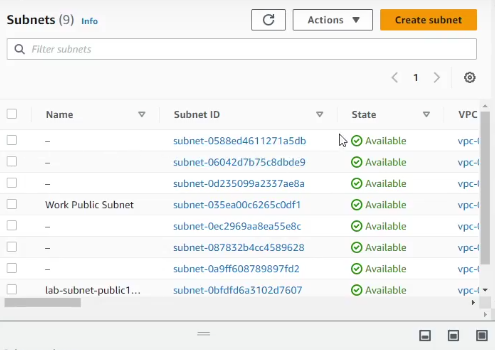
Make sure to confirm that all these settings in the Preview panel and then create your VPC

Once it is created and all the checkmarks are green click View VPC



**Creating subnets for different Availability Zones**

To create more subnets in our VPC, head to the subnets section using the left navigation bar. Then hit the create subnet button near the top right corner.

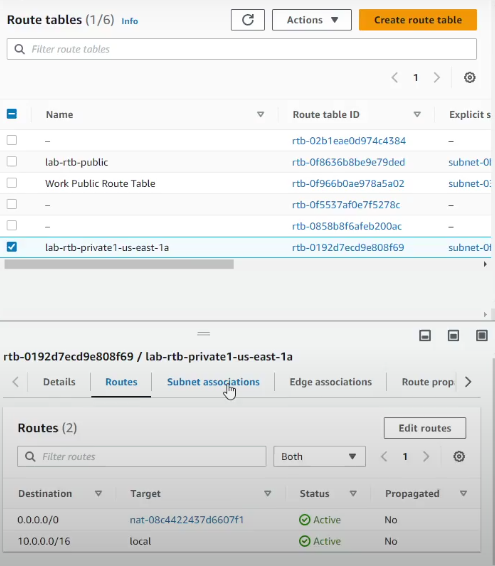


This sends us to this page which lets you configure another subnet for the availability zone us-east-1b.

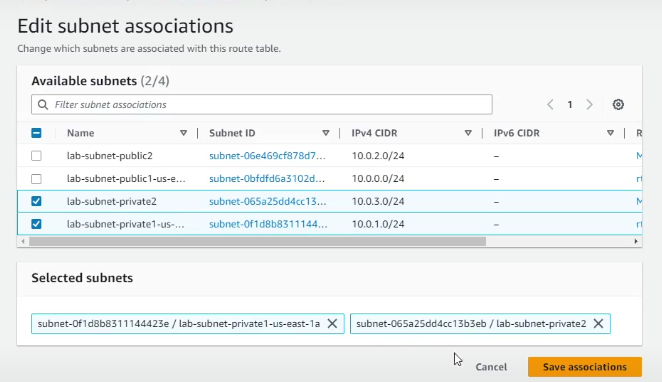
Set the VPC ID to the lab-vpc, set the name to lab-subnet-public2, set the availability zone to us-east-1b. The IPv4 address for the CIDR block will be 10.0.2.0/24

Repeat this to create a private subnet with the name lab-subnet-private2 and the IPv4 CIDR block to 10.0.3.0/24

With these two private subnets we can now view their route tables which will ensure that private subnets are properly routed.



Choose the routes tab in the navigation bar and select lab-rtb-private1-us-east-1a. This table is sending its information for the internet to the NAT gateway it should be.

Now select the subnet associations tab and click edit subnet associations, select lab-subnet-private2 and save the associations.

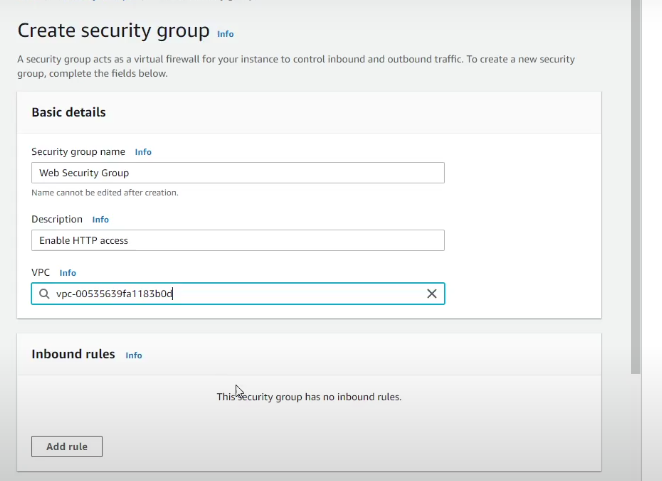
This makes sure that the new private subnet in the second availability zone sends traffic to the NAT gateway

Repeat this same process for the lab-rtb-public route table selecting the lab-subnet-public2 when you edit the subnet associations. With this we now have a VPC with private and public subnets in 2 AZs.

.

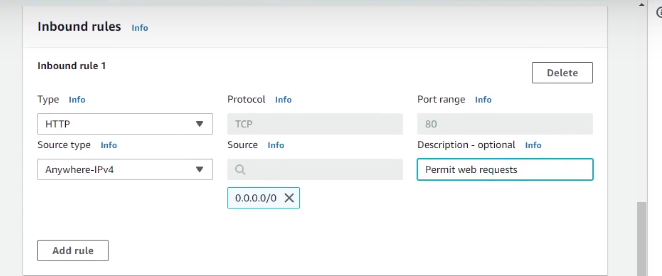
**VPC Security Groups**

Now we can create a VPC security group which acts as a virtual firewall. To begin, select the Security groups from the navigation window. Then choose create security group near the top right of the screen.



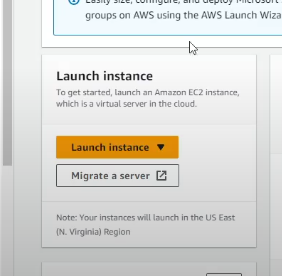
Set the Security Group Name to Web Security Group, make the Description Enable HTTP access and set the VPC to the lab-vpc that is being used.

For inbound rules set the type to HTTP, the source to Anywhere-IPv4 and the description to Permit web requests. Now scroll to the bottom and create the security group.

Now we can test the VPC and our security group using EC2.

**Creating a Web Server with EC2**

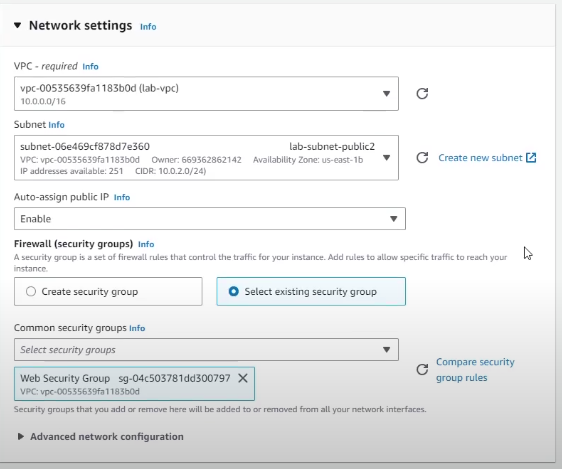
We will configure the EC2 instance to act as a web server.   
In the search box on the right of Services, choose EC2 to open the EC2 console. Select Launch instance.



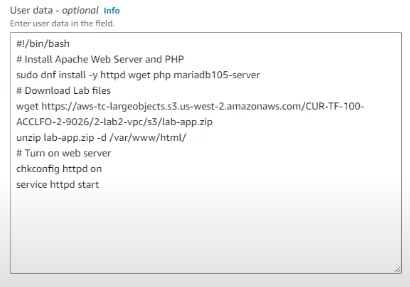
You will be put into a menu to create your EC2 instance. Name it Web Server and make sure that Amazon Linux is selected as the default AMI.

In the Instance type panel, make sure that t2.micro is selected and set the key pair name to vockey using the dropdown menu.

Go to Network Settings, choose edit and configure: Network to be lab-vpc, Subnet to lab-subnet-public2 and enable Auto-assign public IP. For Firewall choose Select existing security group, and for Common security groups, choose the Web Security Group created earlier.



Storage configuration should be kept at default settings and copy the code into the User Data box.



This does basic setup for the Web Server. Then select Launch Instance.

After you see the success message, select view all instances at the bottom.

Then wait until the status check is complete, refresh the page every 30 seconds or so to be aware of the status of the instance.

When it’s done, select Web Server 1 and copy the Public IPV4 DNS value in the Details tab. Paste the Public DNS into a new web browser tab and you should see a web page showing the AWS logo and instance metadata values

**Lab 3 - An Intro to Amazon EC2**

***Background***

Amazon EC2 (Elastic Compute Cloud) is a scalable cloud computing service provided by Amazon Web Services (AWS) that allows users to provision and manage virtual servers, known as instances, in the cloud. EC2 offers a wide range of instance types and configurations, providing users with flexibility and control over their computing resources, including the ability to scale capacity up or down based on demand.

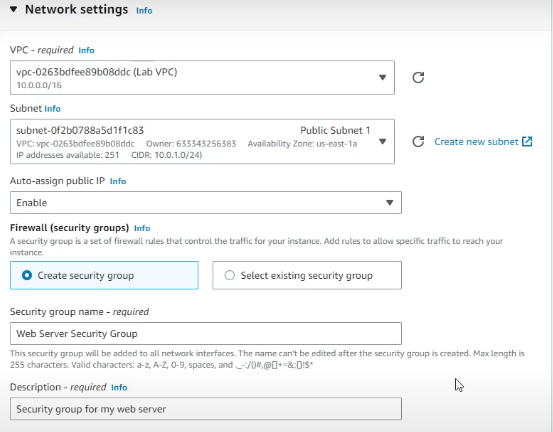
***Lab summary***

Creating and configuring a web server with virtual machines using Amazon’s EC2 service and then investigating all of its features.

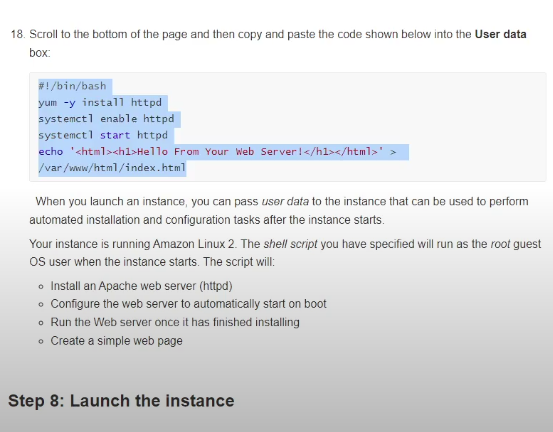
**Launching the EC2 instance**

In the services menu choose Compute then EC2. Go to Launch instance then launch the instance.

Set the name of the instance to Web Server and make sure that the AMI Amazon Linux is selected. Keep the instance type the default t2.micro and set the key pair to vockey. Choose edit and select Lab VPC



Create a new security group and name it Web Server Security Group, set the description to Security group for my web server. Configure storage should be kept default. Go to advanced details and enable Termination protection and copy paste the code into the User Data box.



You can now launch the instance and monitor it to make sure everything is working properly.

Selecting Status checks will tell you about System reachability and Instance reachability

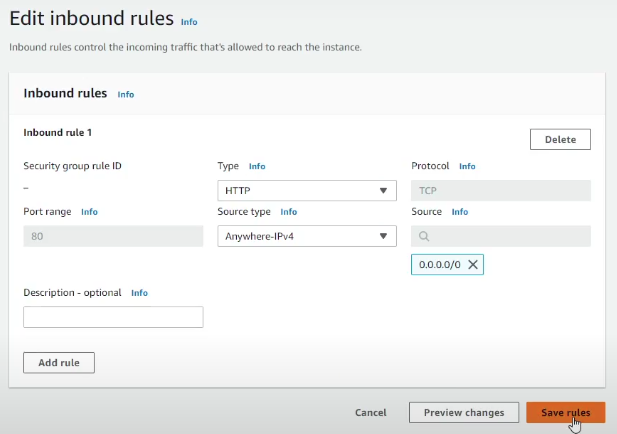
Selecting monitoring will give you Amazon CloudWatch metrics for the Instance

Selecting Instance Screenshot from the Monitor and troubleshoot tab will show you the console of the instance

**Upgrading the Security Group**

If we attempt to access the web server we won’t be able to since our security group doesn’t allow for any traffic to enter on port 80 which is HTTP.

To change this enter the details tab and scroll down to the inbound rules and edit them.

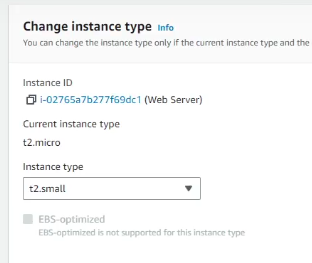


Set the type to allow HTTP traffic and anywhere-ipv4. Now you should be able to access the web server using the public ipv4 address in your web browser.

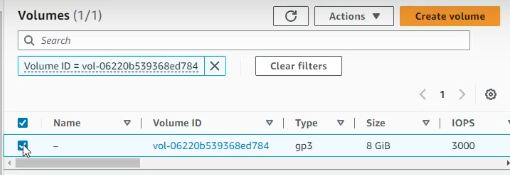
**Resizing the instance**

If the instance needs to be upgraded or downgraded due to overutilization or underutilization, you can change the instance type. If you need to change the size of the instance you need to stop it using the instance state dropdown.

In the Actions menu, choose instance settings and then change instance type. Configure the instance type to t2.small.

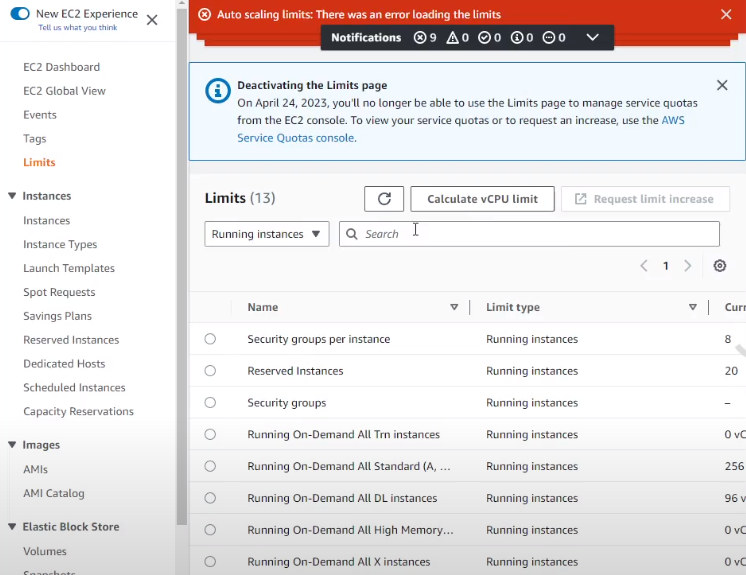


Then select the storage tab with the web server still selected and select volume. Select Modify volume and increase the size from 8 GiB to 10. Go back to instances and start the instance again.



**EC2 Limits**

EC2 provides many different resources that you can use. But these resources have limits set when you create an AWS account based on the region you are in. Search for Service Quotas to view the limits set by Amazon.

.

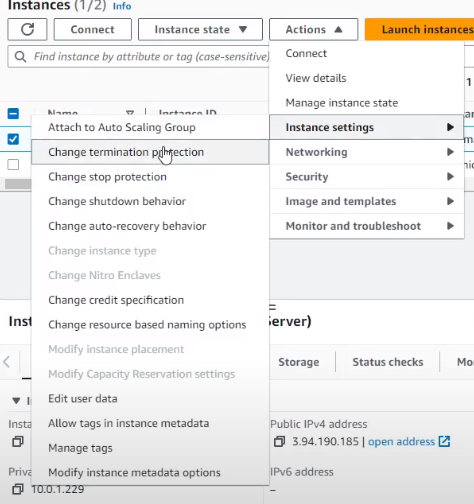
**Termination Protection**

When you are done with an instance, you will terminate it. You cannot connect to or restart an instance after it has been terminated. Go back to the EC2 console and choose instances. Select the Web Server and terminate it. An error message should pop up.



This means our instance has termination protection turned on, which is a handy safeguard to accidental deletion.

To turn off termination protection, select Actions > Instance Settings > Change Termination Protection and uncheck the box that says enable and save.



Now you should be able to terminate your instance.



The EC2 web server has been completed