**AWS ACADEMY LABS 1 - 3**

******

***BY: ALEKSIE SANCHEZ MEDEIROS***

******

***Purpose***

To gain an understanding of 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***

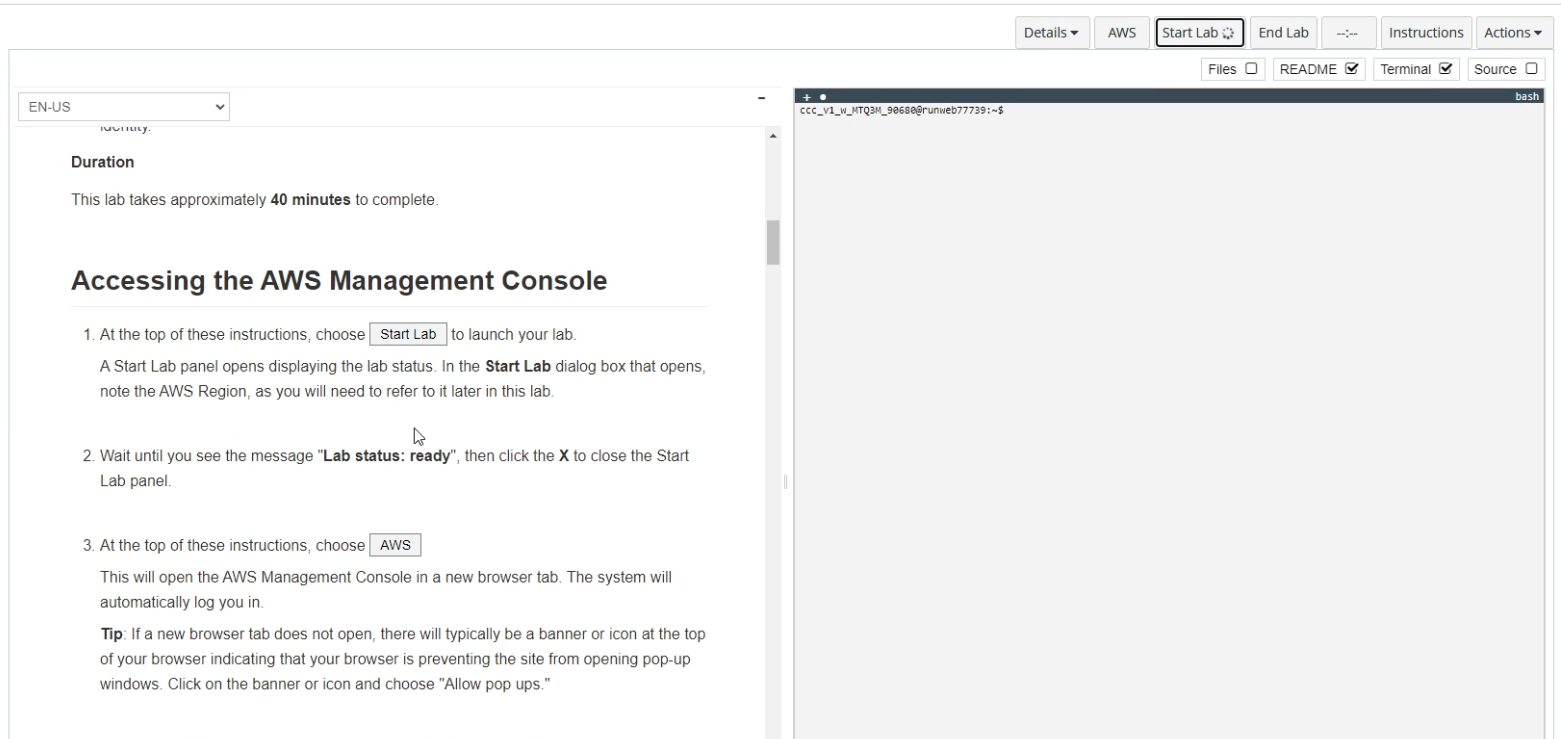
Amazon’s Identity and Access Management (IAM) is one of their many services offered for AWS customers. The IAM service focuses on providing those customers with the ability to ensure that the services that are being used are restricted to only users who should be able to use it. For example, a developer whose work only focuses on using EC2 should only have the ability to access EC2 instead of all the AWS services where that developer could cause damage to the system.

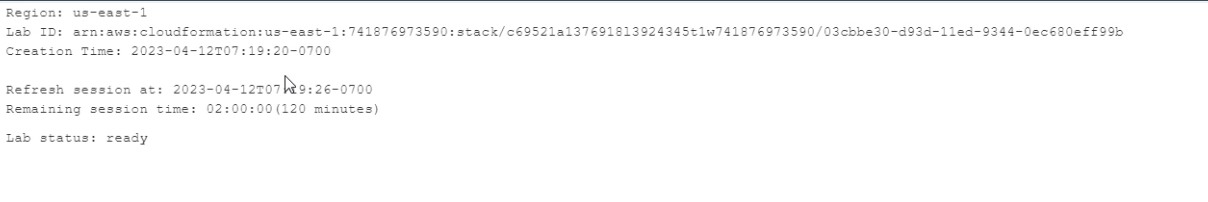
***Lab summary***

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***

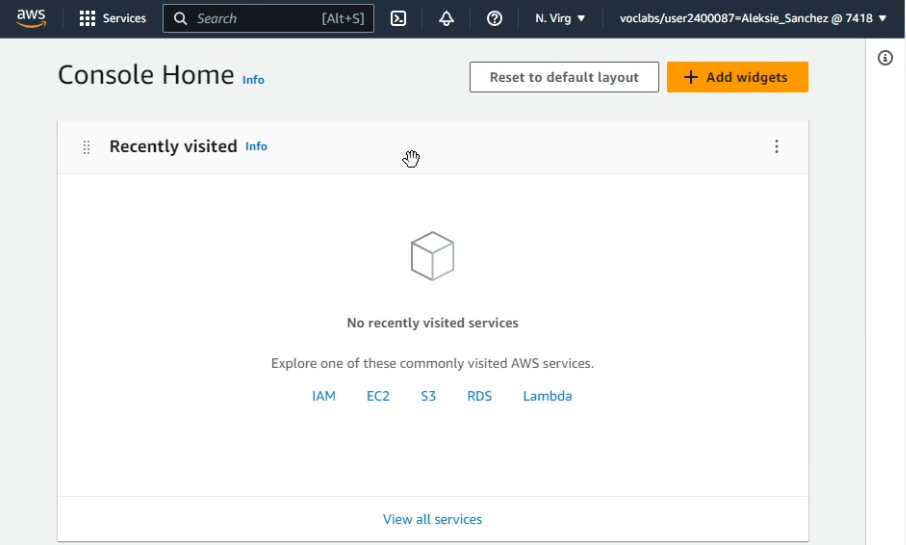
To begin you first need to start an AWS instance, the lab makes this simple enough by creating a simple Start Lab button near the top right of the landing page.



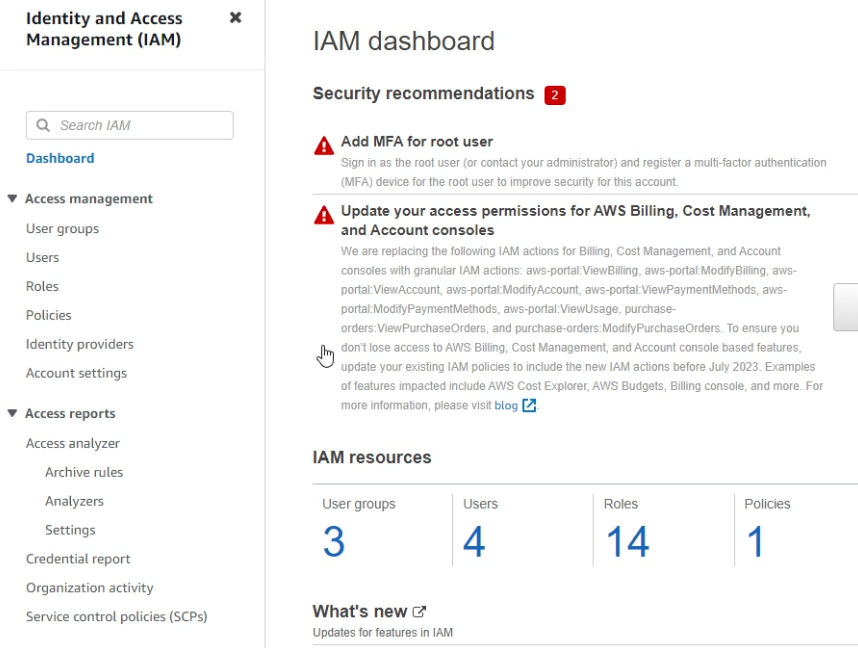
This will open up a pop-up which will tell you the region where the AWS instance starts, the lab’s creation time and how much time is remaining in the lab session.   
  
Wait until the lab status changes from ‘in creation’ to ‘ready’, this took around 3 minutes.   
  
This means the AWS instance is ready for use and you can close the popup.

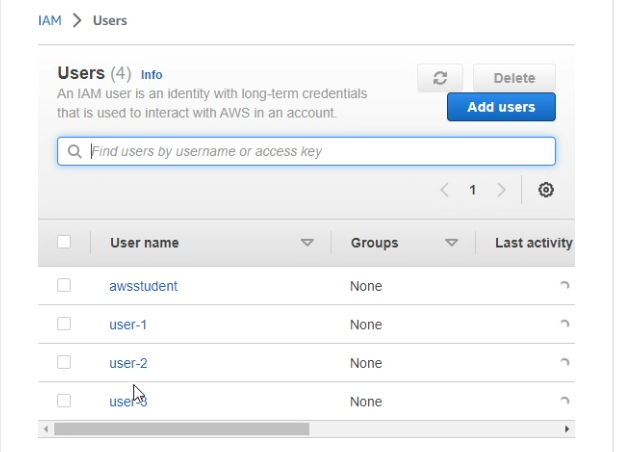
Pressing the AWS button to the left of the ‘Start Lab’ button will open the AWS management console and allow you to begin the lab.

**An Investigation Into IAM**

  
When you arrive at the management console home page, you should see a screen similar to the one above.

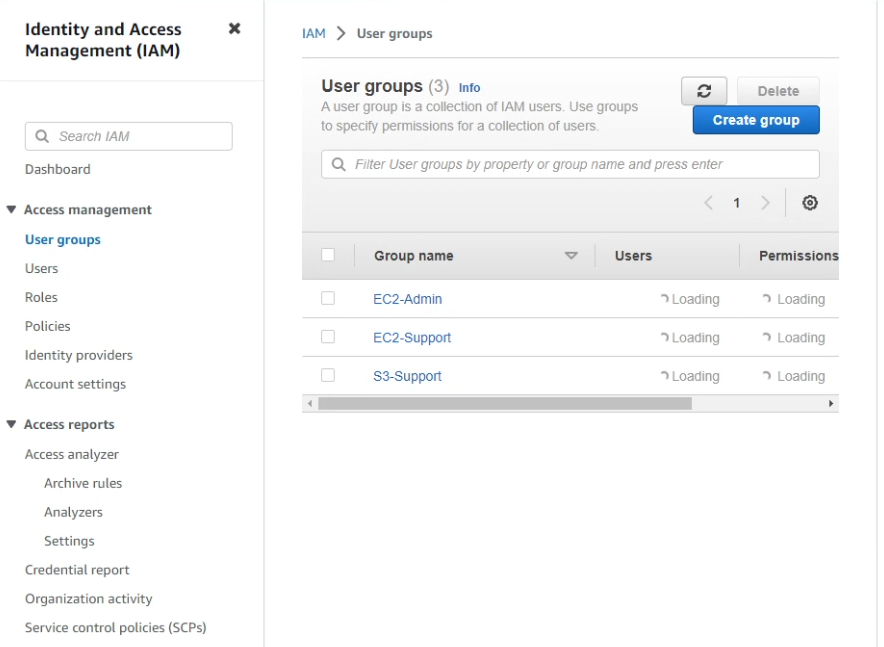
From here there are multiple ways to arrive at the IAM Management Console. You could press the hyperlink in the center of the page where the commonly visited AWS services are or use the Services drop down menu and search for the IAM service

Using either of the methods, you should arrive at the IAM dashboard, this 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.

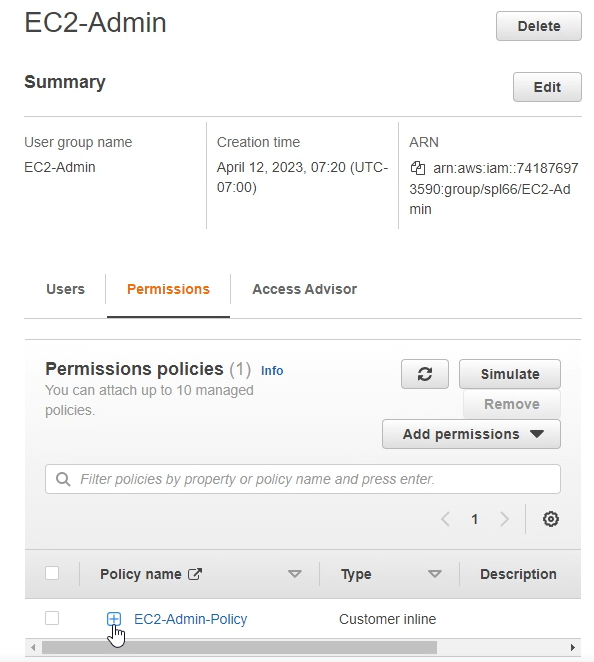


When you arrive at this page, you should see 3 existing user groups.  
Select the EC2 Support group, although this group has no users, it does have a permission policy which you can view by selecting the permissions tab from the menu.

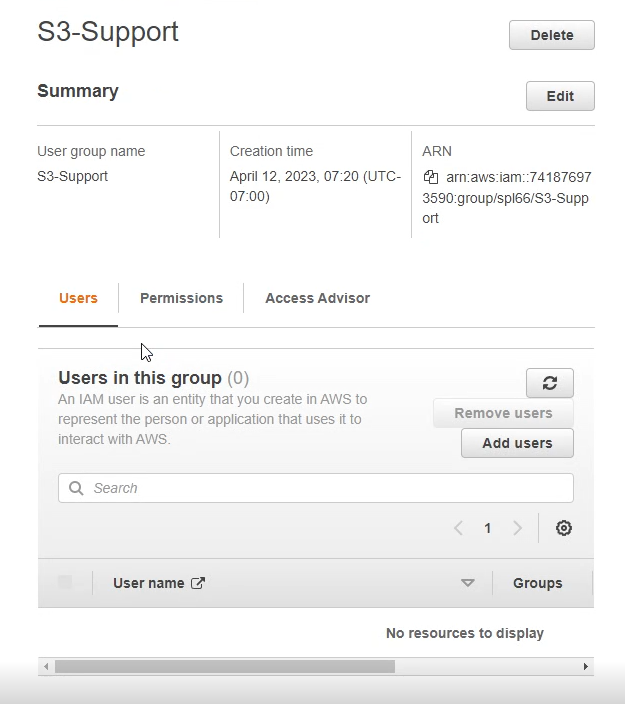
This permissions policy’s name tells you almost exactly what it does. This policy allows any users inside the group to only have the ability to view EC2 resources and not modify them.  
Pressing the + icon on the policy will show it’s details, specifically it will show its JSON file. The JSON file follows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

**Action** - which specifies the API call being used on a service (ex: viewing resources on an EC2 instance)  
**Resource** - which sets boundaries for what is covered by the policy (the use of a \* signifies any resource)

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.

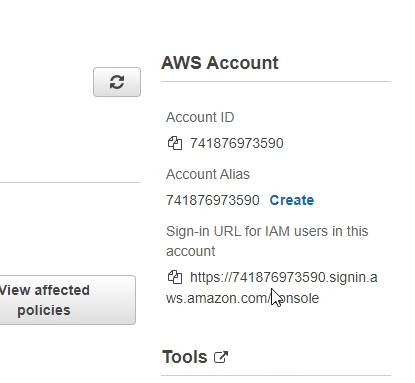
**Giving permissions to users by assigning them to groups**

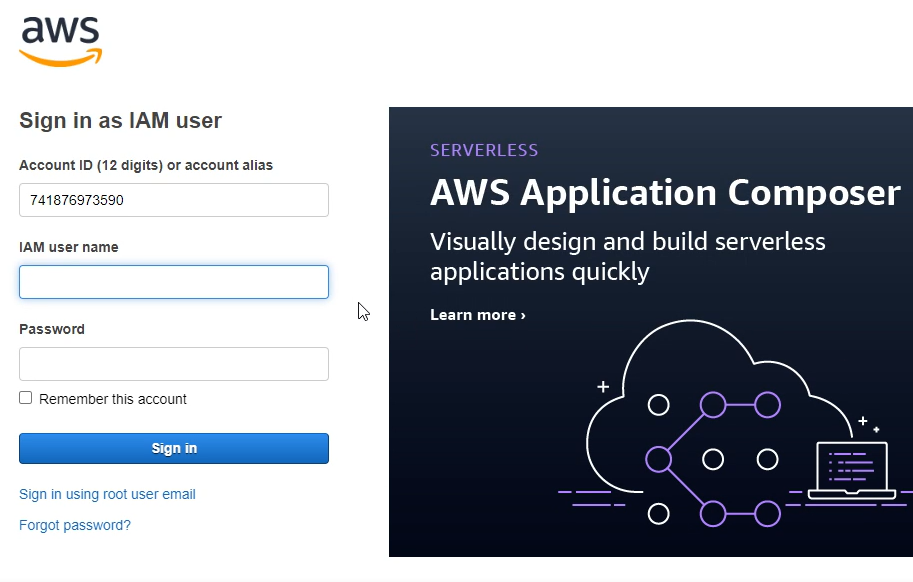
From the user groups menu, we can now assign each user to a different group, to do this select the group that you’re going to add a user to and hit the add users button near the middle of the screen, as shown below.

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.

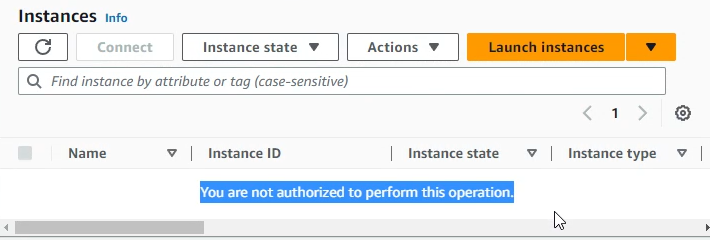
**Testing User Permissions**

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

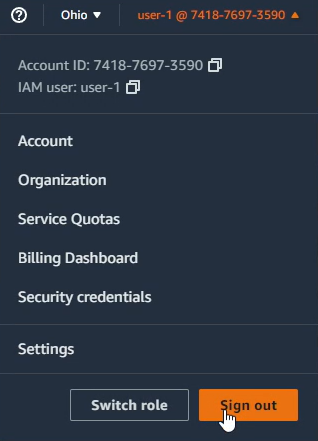
To do this go back to the dashboard and copy the url near the top right of the screen

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.  


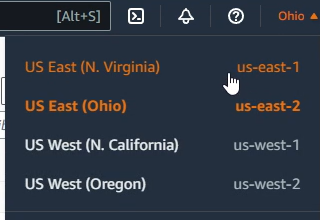
This is what the landing page looks like for the link that was copied, your account id may be different.  
  
Enter IAM username user-1 with password Lab-Password1 to test permissions for user-1.  
  
We can test if the user is able to view S3 buckets and their contents by entering the S3 service, as 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.



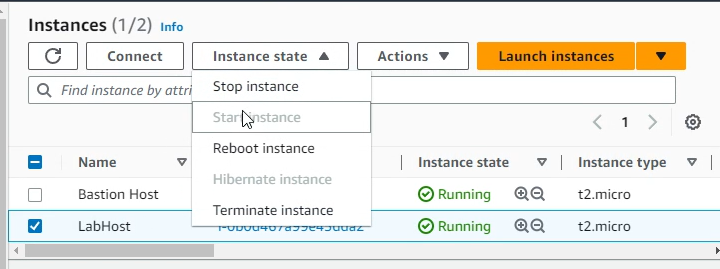
Sign out of the user-1 account by pressing the user dropdown near the top right of the screen.



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.



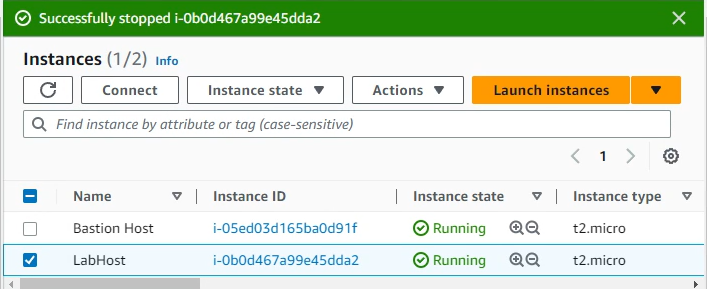
Now select the checkbox next to the LabHost instance and attempt to stop the instance by going to the instance state drop down menu and selecting Stop instance.



If permissions have been set correctly, you should see a long and winding 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.



Congratulations, you have finished Lab 1 and gained an understanding of the AWS service IAM. 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**

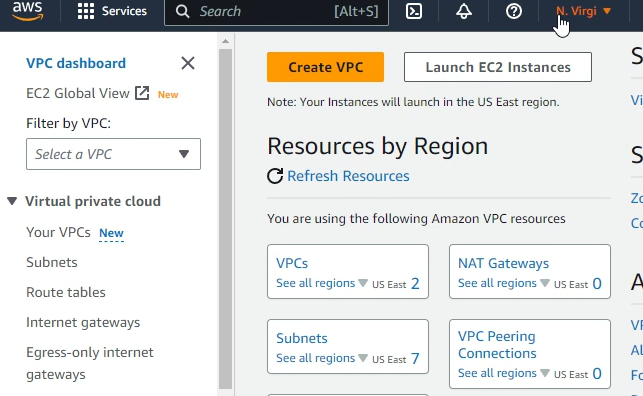
***Background***

Amazon’s Virtual Private Cloud (VPC) is one of their more versatile services made for AWS customers. The VPC service allows for network resources to be allocated to an AWS user which allows them to set up communication from AWS services to the internet for example. In addition, you can use VPC to allow communications between physical hardware and the cloud.

***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 the VPC wizard that will make creating a VPC a very streamlined process.

NOTE: SOME OF YOUR VPC SETTINGS WILL BE SET TO THE CORRECT ONE

To begin, ensure that VPC and more are selected on the resources to create subsection.

Next, change the value underneath auto-generate from project to lab.

Keep the IPv4 CIDR block the same.

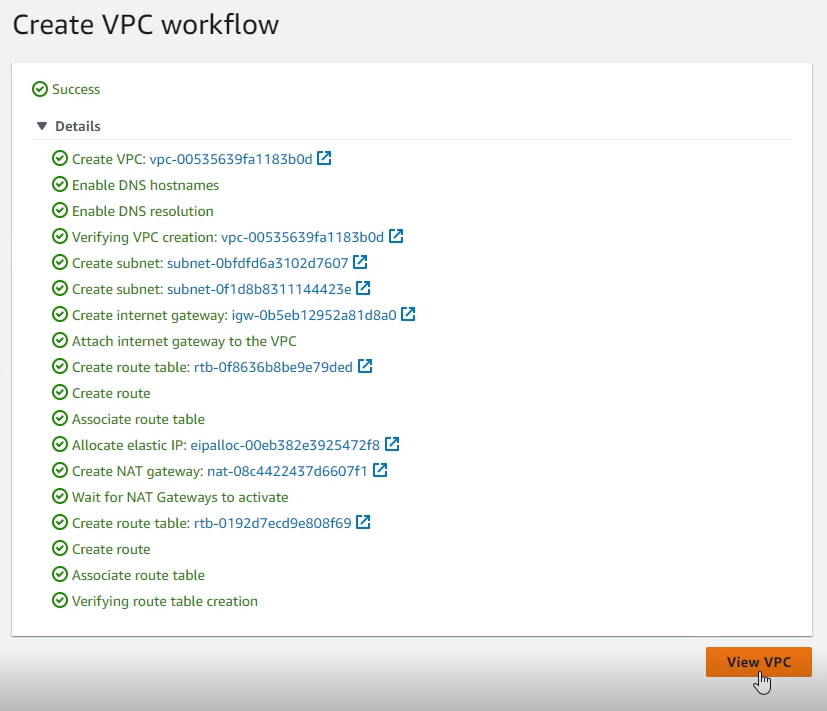
Set the Number of availability zones, the number of public and private subnets to 1.

Set the CIDR block for the public subnet to 10.0.0.0 /24 and the private to 10.0.1.0/24

Then set the NAT gateways to “In 1 AZ” and the VPC endpoints to None.  
Keep the DNS options enabled and create your VPC after ensuring the settings are right.

Creating the VPC workflow may take a while, it took around 2 minutes for it to be complete

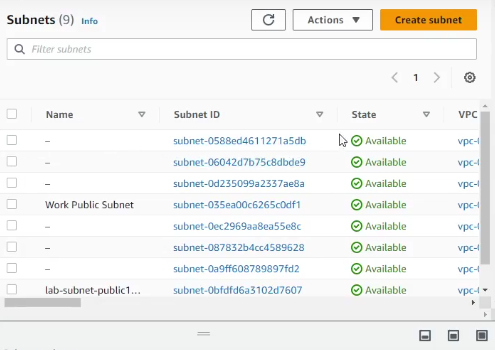
When all the checkmarks light up, select the View VPC button



The process of creating this VPC enables you to be able to create communications between EC2 instances and other AWS features or traditional physical networks for example.

**Creating subnets for different Availability Zones**

To ensure that there is high availability we can create another subnet in a separate availability zone. 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.

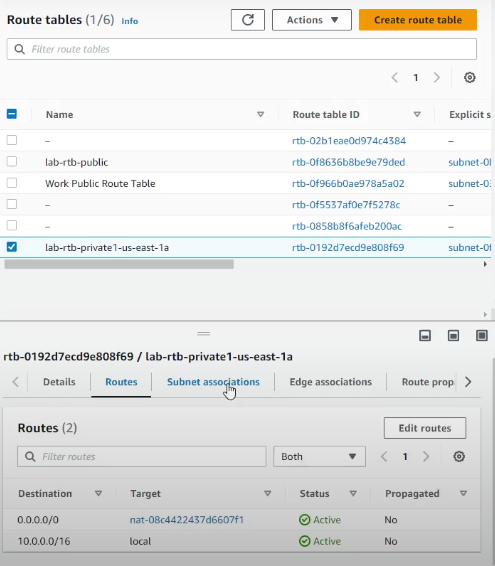


This sends us to a set menu, first we will create a public subnet for the availability zone us-east-1b.

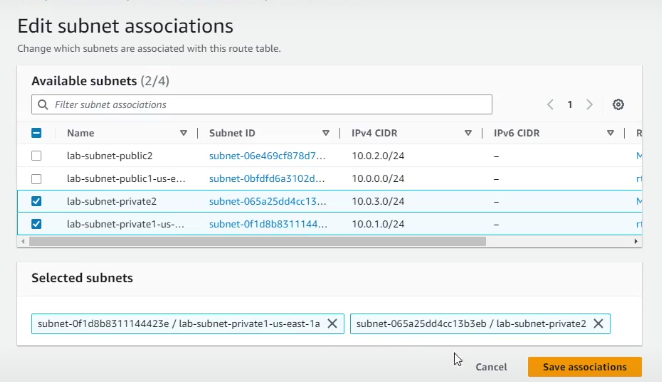
To do this, set the VPC ID to lab-vpc, the name to lab-subnet-public2 and the availability zone us-east-1b. This public subnet will have all the private IPv4 addresses in the 10.0.2.0/24 range.

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

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



To begin move to the route tables using the navigation bar and select the lab-rtb-private-us-east-1a, this table is currently sending all information destined to the internet towards the NAT gateway as it should be.

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

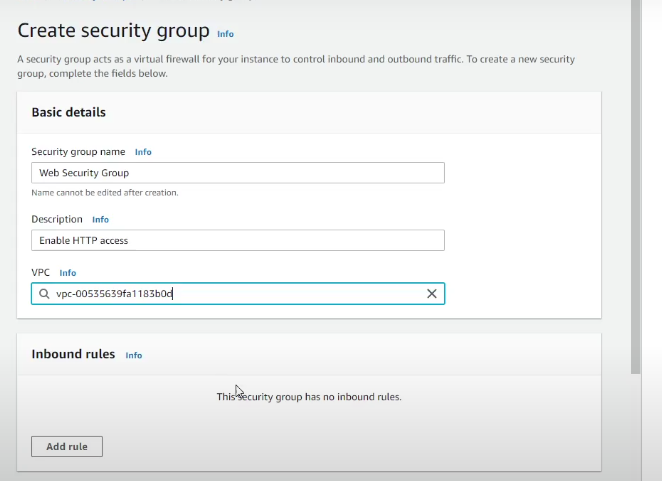
This makes sure that the new private subnet in the second availability zone sends traffic destined for the internet to the NAT gateway.

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.

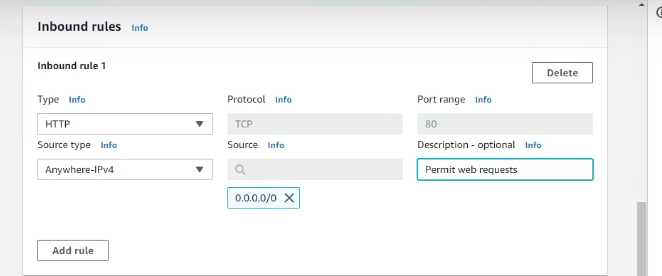
.

**VPC Security Groups**

Now we can create a VPC security group which will act similar to a firewall. It’s very similar to how access lists work in traditional infrastructure. To begin, select the Security groups from the navigation window. Then choose create security group near the top right of the screen.



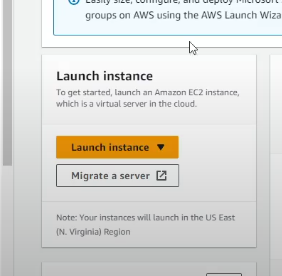
Set the security group name to Web Security Group, a description of Enable HTTP access and set the VPC to the lab-vpc that we have been using.

Then for the inbound rules, select add rule and set the type to HTTP and the source to anywhere-IPv4 then set the description to Permit web requests in case you forget what the rule does. The reason we have to allow, is because AWS has implicit deny on almost everything

Now we can test this security group as well as our VPC.

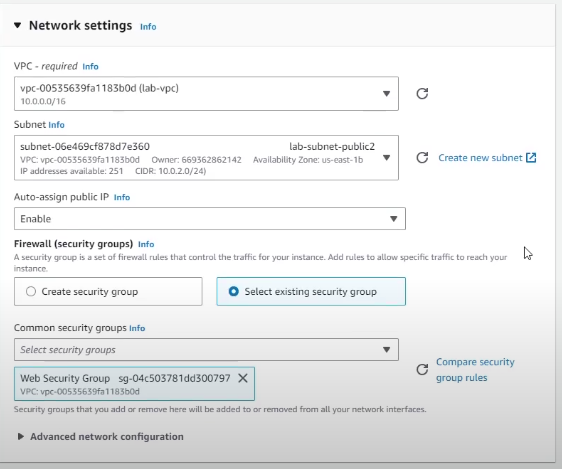
**Creating a Web Server with EC2**

Now we can integrate an EC2 instance into our new VPC. First head over to the EC2 service using the services box near the top left and looking for EC2. Now hit the Launch Instance button.

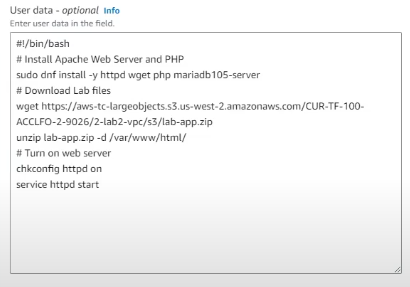


This will move you into a menu to create your new EC2 instance. First change the name to Web Server 1 and ensure that Amazon Linux is selected as the Amazon Machine Image. Keep the default instance type and set the key pair name to vockey using the dropdown.

Select the edit network setting button and set the network to lab-vpc, the subnet to the lab-subnet-public2 and enable auto assign public IP, for the firewall security group select an existing one and choose the Web Security Group we created earlier.



Keep the storage configuration the same and in the advanced details copy paste 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, if you are in a rush you can reload the instances screen until it says the check has finished. I waited around 3 minutes then hit the reload button.

When it’s set up, select Web Server 1 and copy the public IPv4 DNS and you should see the AWS lego and meta-data values, if so you have successfully created a VPC with subnets in two AZs that allows for HTTP traffic to a Web server integrated using EC2.

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

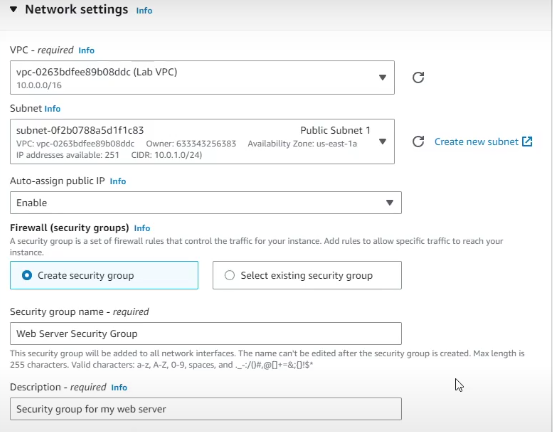
***Background***

Amazon’s Elastic Cloud Computing (EC2) is similar to a rental service but for computing power. Amazon will give you virtual computers which are internally referred to as instances which are entirely manageable within the web dashboard. These virtual machines are available at any moment which is why they are referred to as elastic and they can be scaled to whatever size is needed at a moment’s notice.

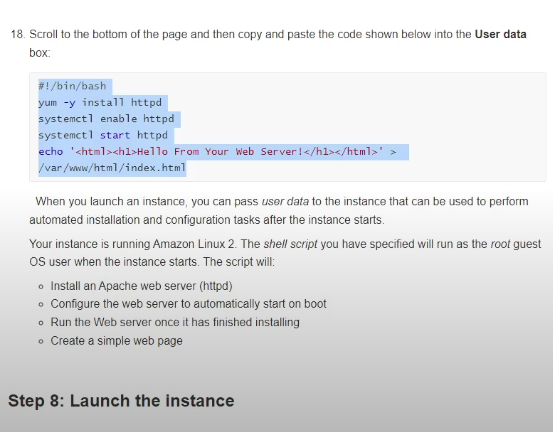
***Lab summary***

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

**Launching the EC2 instance**

To begin, ensure that the resources being managed are currently in the us-east-1 region, then enter the EC2 service by using the services menu. Then select the launch instance button. Set the name of this instance to Web server and ensure that the Amazon Linux AMI is selected as was done in the previous lab. Keep the instance type the same and set the key pair to vockey like before. Edit the network settings to the lab-vpc which has been preconfigured similar to the previous lab’s vpc.

Create a new security group with the name of Web Server Security group and a description like the one above. Then remove the inbound security group rule by hitting the remove button next to the rule. The storage configuration stays the same, the critical part is in the advanced details, where you have to turn on the termination protection to ensure that there is no accidental deletion of the instance. Then copy and paste the code into the user data.



With all this configuration setup properly now you can launch the EC2 instance.

Now we can monitor our instance to ensure that everything is working as intended.

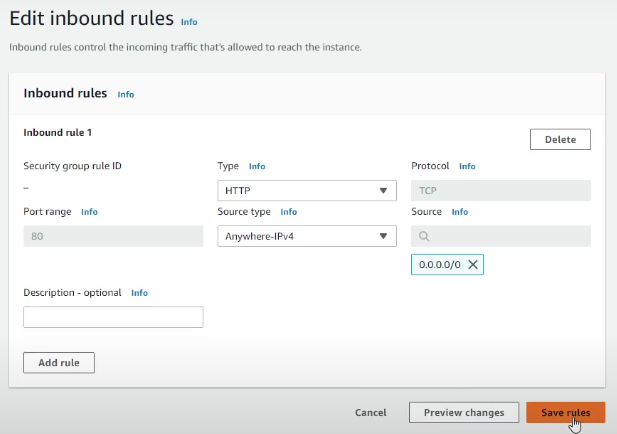
Selecting the status checks will tell you about the 2 checks amazon does

Selecting monitoring will give you important metrics

An instance screenshot will give you a view of your instance if it was physically there.

Now 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.

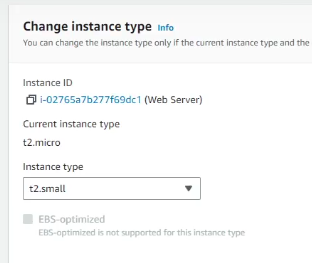
To change this enter the details tab and scroll down to the inbound rules and edit them.

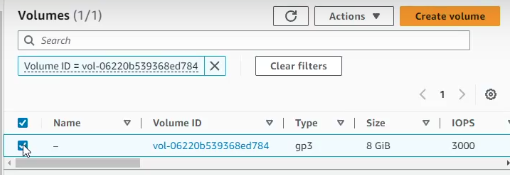


Set the type to allow HTTP traffic and anywhere-ipv4. Now you should be able to access the web server using the public ipv4 address in your web browser.

**Resizing the instance**

In case you ever need to change the size of the instance you can stop the instance using the instance state drop down

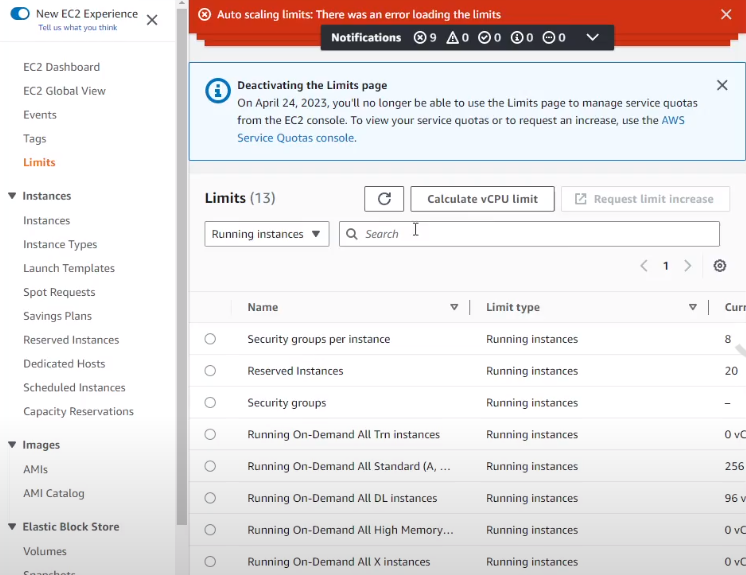
Then with the actions enter the instance settings and then scroll down through the drop down settings until you find t2.small then select apply.

Then select the storage tab with the web server still selected and select the volume hyperlink. This will lead you to a screen where you can select to modify from the actions dropdown and select modify volume.

Then increase the disk size to 10 gb and select modify.

With that go back to the instances page and start it up again.

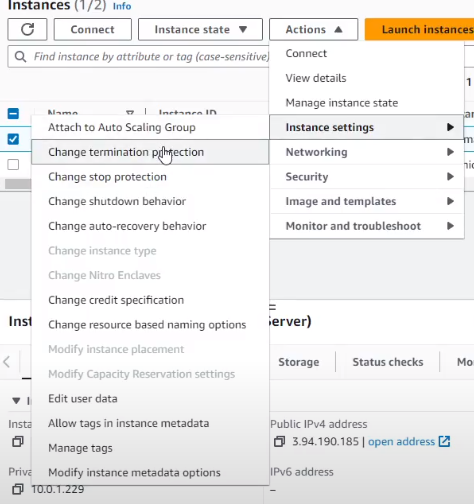
**EC2 Limits**

EC2 instances have built in limits; these limits may vary by region and may have rate limiters set by Amazon, all of this is viewable from the limits menu accessible from the left navigation bar.

**Termination Protection**

Now that we are all done with the investigation and set up process we have to terminate or delete the instance. To do this head back into the instances menu and select Terminate Instance from the instance state menu. You should see an error message.

This means our instance has termination protection turned on, which is in general a very good thing to avoid accidental deletion.

To turn off termination protection select actions –> instance settings → change termination protection and uncheck the box.

With that you should be able to terminate the instance.



Congratulations, you have successfully finished creating and configuring a basic web server using the AWS service EC2!