Brit Stevens 12/7/23

AWS Labs 1-3

**Purpose:**

The purpose of these labs was to know what AWS services look like in a browser and how we would interact with it like in real life. These labs are also meant to better prepare us for our AWS exams as doing hands on assignment better our understanding of these services and scenarios. It introduced us to many of the services like EC2, S3, VPC, and every concept about the configuration of those services.

**Background Information on lab concepts:**

**Amazon Webservices**: Amazon Webservices or AWS as it is more commonly known as is a wide array of services that are provided by Amazon. They handle the physical aspects of cloud infrastructure and allow you to focus on the cloud side. Amazon uses a pay by use model and as AWS grows it will be cheaper and cheaper as the pool of users increases. AWS is commonly used by the military, global banks, and companies because of its security and credibility.

**IAM:** IAM (Identity and Access Management) is a webservice that is a security and authorization system that control what different users on your AWS dashboard (where you can view all your AWS services) can view and interact with. This service uses authorization and authentication to achieve this. Authentication ensure that the user has a log in with certain authorization allowing them to only view certain services on the dashboard. Instead of managing permissions of users individually you can put them into a group and apply policies to a group centrally. This is more efficient and leaves less room for error in applying permissions.

**IAM Roles:** An IAM Role is similar to a user account, it is an AWS identity with permission policies that determine what that role can and cannot do in AWS. The difference is instead of being uniquely associated with one person, a Role is intended to be assumable by anyone who needs it and can be applied to many users.

**IAM Policy:** These are policy that are applied to an IAM user group that allows or forbids certain actions or viewing of services for those groups. The basic structure of the statements in an IAM Policy is:

* **Effect** says whether to Allow or Deny the permissions.
* **Action** specifies the API calls that can be made against an AWS Service
* **Resource** defines the scope of entities covered by the policy rule (specific S3 Buckets or EC2 Instances)

Inline Policy, Different than management policies, which is a policy assigned to just one User or Group. Inline Policies are typically used to apply permissions for one-off situations for instance if you promote a worker to admin changing their user account to an administrator.

**API:** An Application Programing Interface or API for short are mechanisms that enable two software components to communicate with each other using a distinguished set of definitions and protocols exchange information. APIs are everywhere, for example watching YouTube videos on your phone requires an API to bring the video from a Google Database to your phone app or browser.

**S3**: S3 is an object storage system located in the AWS cloud. It is highly scalable and very secure like the rest of AWS services. It works by moving your data of any kind into the S3 buckets and lets it be accessed by anyone who you give access to in regions you select. An S3 bucket is what AWS decided to name how the data is stored, you can think of it like all your objects (data) are stored in the bucket and the bucket can be as big as you want. One object can be up to 5 Terabytes in size, but your bucket can go up to any size with an increase in price. You can also store your snapshots, essentially backups of all your data in a volume, in S3 buckets. With S3 you can manage your data by applying object tags to certain sets of data that are linked to easily search and create reports on those sets which is another feature of AWS. You can customize and configure your S3 service to generate reports for different buckets and tags and see how often that data is accessed to determine if you use a not as highly available storage option to save money. Whatever region you choose to primarily store your data will have the fastest access to the data. Because of this S3 has the option to duplicate your data across buckets in different regions for quicker access at additional charges.

**CIDR:** Classless Inter-Domain Routing (CIDR) is used to improve the efficiency of data routing on the internet through IP address allocation. Since every device has an IP address organizations use CIDR to give out these IP addresses efficiently without doing it manually. Unlike Classful address that use a set Network prefix length (8, 16, or 24) that you must purchase Classless use a VLSM (Variable Length Subnet Masking) allowing greater customization of the size of each subnet. Instead of having to force groups into a certain classful group of IP addresses which may be too big or too small you can only create as many addresses as you need with Classless making management more efficiently and standardized.

**Subnets:** Subnets are just groups of IP addresses that share a network address. IP addresses are structured with a network address at the start of the IP address and a host address at the end of the IP address made up of 32 bits. Each string of numbers separated by the periods consist of 8 bits and show a number between 0 and 255. Where the network address ends, and host address starts depends on the Subnet length it has. If a subnet has more bits, then there are more host addresses available in that subnet.

**Private vs Public Subnets:** Private addresses are for internal data transmission for example one desktop to another on the same network. Public addresses are ones meant to go out to the internet.

**Nat gateway:** This allows public addresses to be converted into a unique address when sent to the internet. This has to be done so each address can be unique on the internet allowing the data to come back to the same desktop once sent out. NAT is that process of conversion to a new address ready to be sent to the internet.

**Routes:** Each time any data is sent across a network it requires a route. This route is in the form of a header packet including the IP address of where you want to send the data and the MAC address (A physical number unique to every device) of the device is has to hop to next to reach that IP. Each time the packet reaches its next hop the MAC address changes to the next hop and so on until it reaches the final destination.

**VPC:** Amazon Virtual Private Cloud (VPC) enables you to launch AWS resources into a virtual network that you defined. This virtual network closely resembles a traditional network like any company but has the benefits of using the scalable infrastructure of AWS. You can create a VPC that spans multiple Availability Zones in a region.

**Availability zones**: where all the data is store for your EC2 and instances. They are massive servers that hold data inside of a region like US-east. There are many availability zones in a region.

**DNS:** Domain Name Server (DNS) translate the domain of server names (like amazon.com) to an IP address allowing your desktop to connect it. This system was put into place so each time you wanted to access any website you wouldn’t need to memorize the IP address associated with that website. This is a global system used by the internet and whenever your desktop requests to access a website it is called a Query.

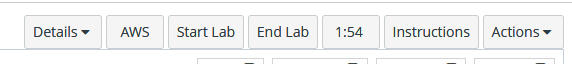
**AMI:** provides the information required to launch an instance, which is a virtual server in the cloud. It includes a template for the root volume for the instance (could be an operating system like windows or Linux). Whatever you choose will be the root volume that the EC2 instances you create will run on. You can choose a different AMI for all your instances. Also provides launch permissions based off of the IAM policies of who can access use this AMI as an option. It also records what volumes or S3 buckets are attached to the instance once launched.

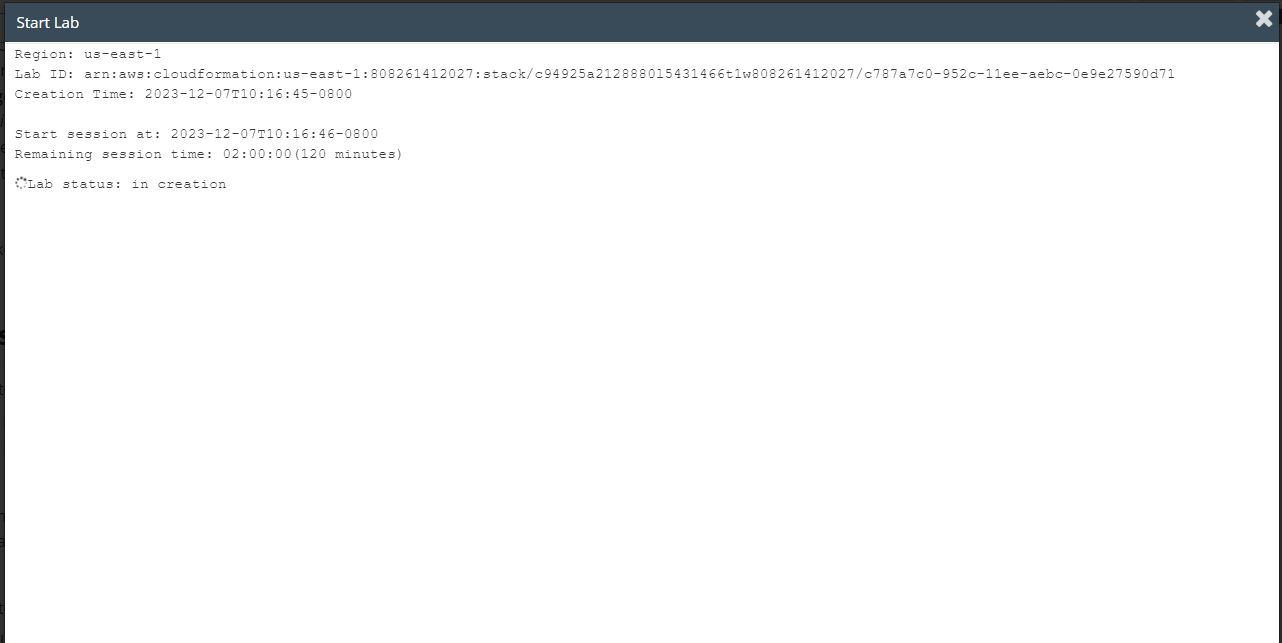
**EC2:** is a web service that provides resizable compute capacity in the cloud mainly to aid developers in creating web application. Amazon EC2's simple web interface allows you to obtain and configure capacity with ease, one reason it is so used. It provides you with complete control of your computing resources and lets you run on Amazon's computing environment with their mass amounts of server resources. As your computing requirements change, you can scale up or down as much as you please due to their highly scalable architecture that is the pay for what you use model of billing. EC2 also provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios with the use of multiple availabilities zones and S3 snapshots.

**EC2 Instances:** The instance type defines the hardware resources assigned to the instance with many different options each optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. If you are looking for an EC2 instance mainly for storing data from users than you may choose to have a high amount of data and lower processing power. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

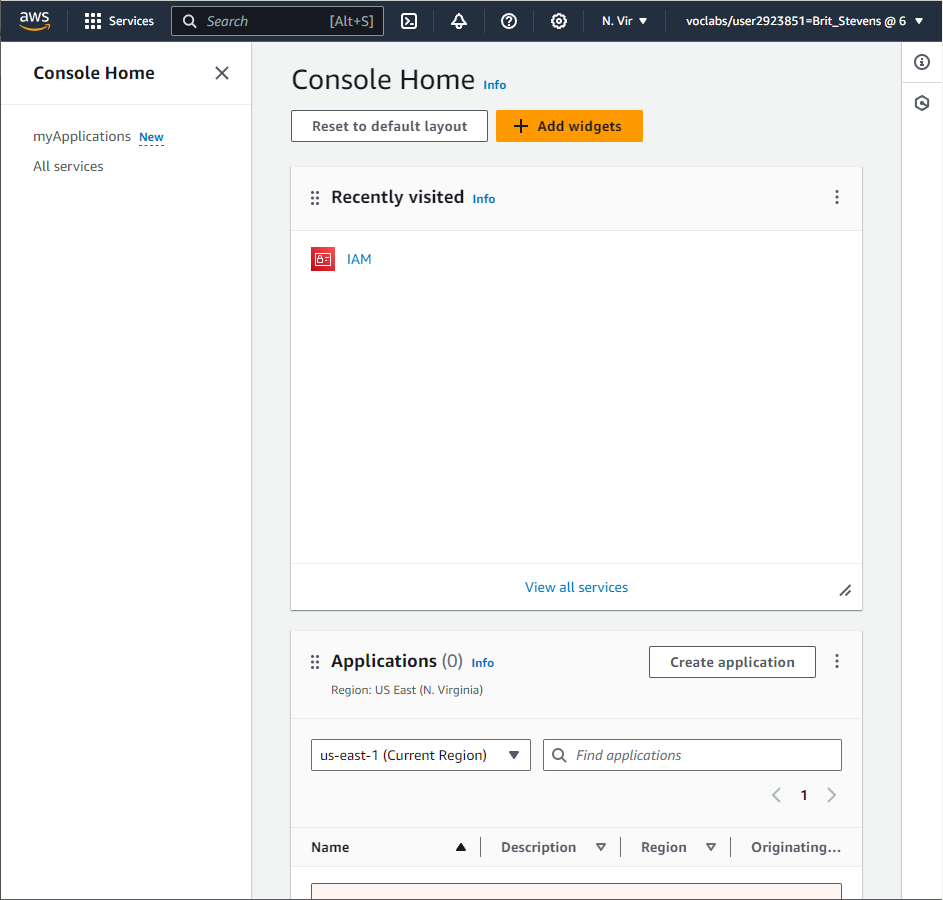
**Tags:** The name you give this instance will be stored as a tag. Tags enable you to categorize your AWS resources in different ways, for example, their purpose, owner, or environment in which they will be used. This is useful when you have many resources of the same type; you can quickly identify a specific resource based on the tags you have assigned to it. Each tag consists of a Key and a Value, both of which you define. You can define multiple tags to associate with the instance if you want to. These tags are key for efficient documentation will aid in timely changes.

**Lab Summary: AWS IAM Lab**

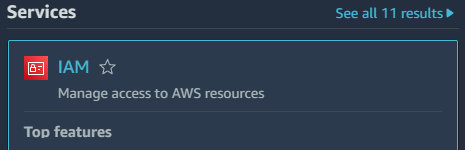
****

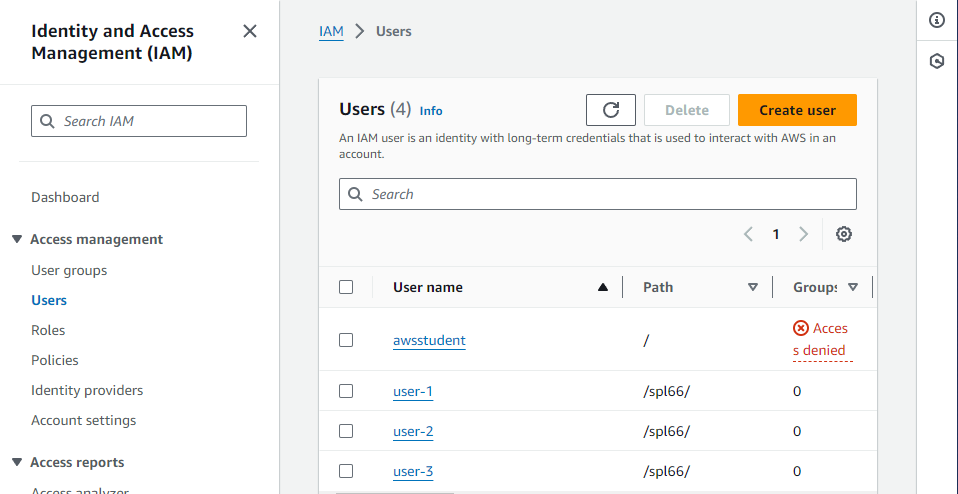
****

click start lab and wait until the prompt box says lab is ready. Once lab is ready click AWS to go to the EC2 instance.

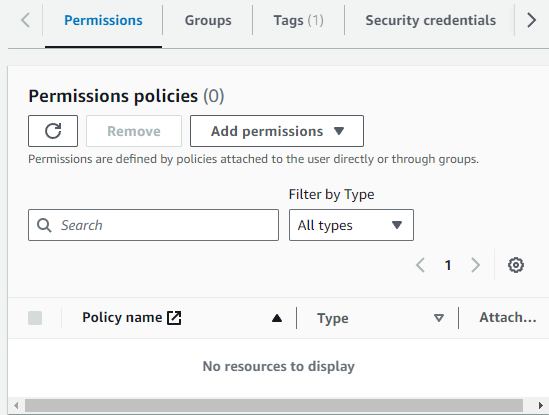
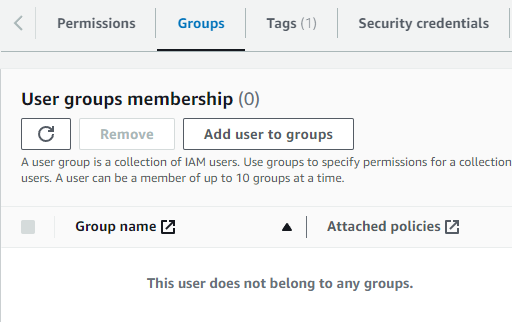


You will see a website like this. Click on services and search for IAM.

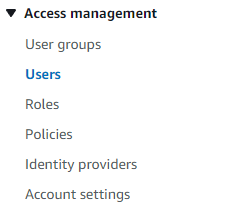
****

****

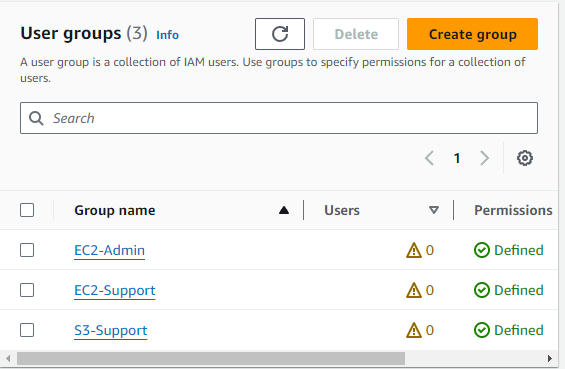
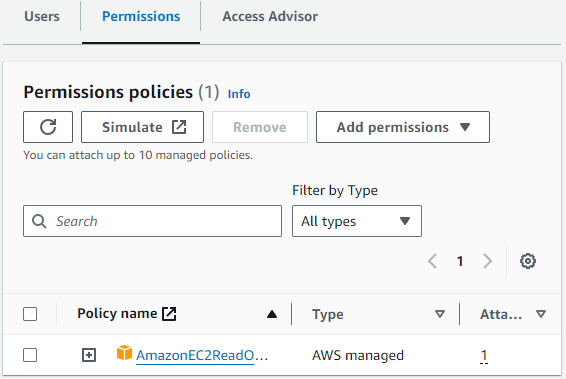
Once in IAM go to users and click on the already created user-1.

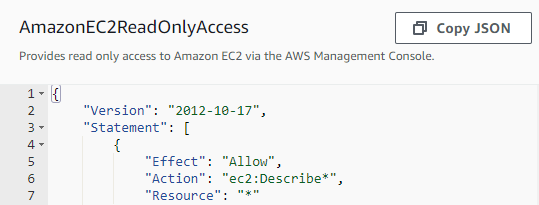
User-1 has no permissions and is not apart of any group. Lets assign the user to a group to apply a range of permissions.



Click on user groups.

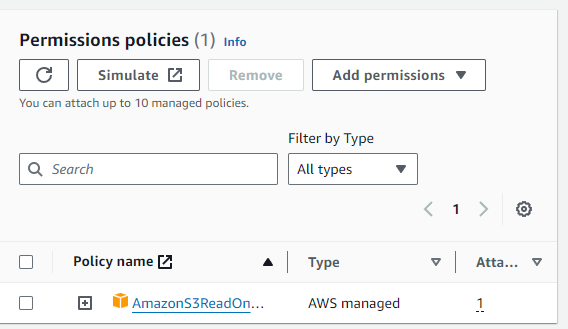
There are 3 preconfigured groups, select EC2-Support. It has permissions to read only on the EC2 Instance. If you click the plus icon next to the policy, you can read the Json code in the rule.



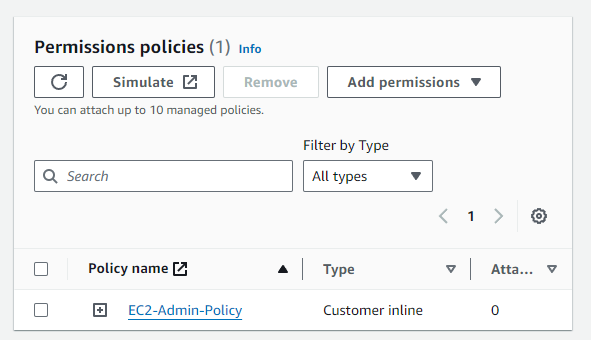
Effect is for allow or deny.

Action is what API calls can be made (can describe EC2).

Resource is what it can pull from or the scope of the resource (\* is any resource).

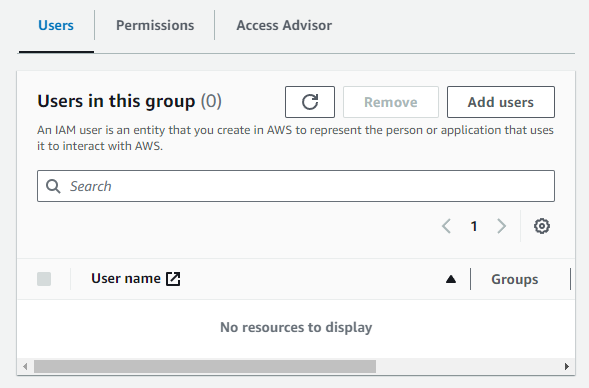


Looking at S3-Support group, it has the permission to read amazon S3.

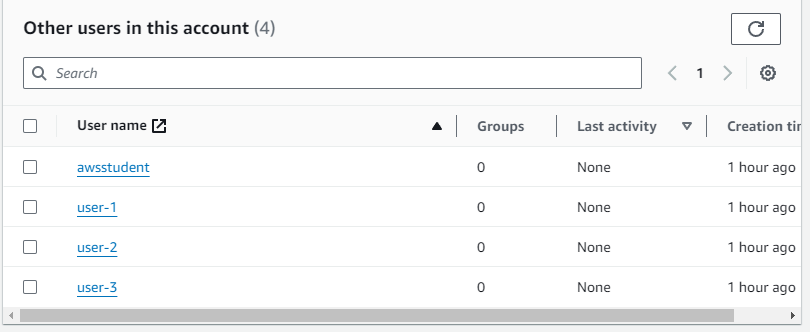


EC2-Admin is another group with a slightly different policy type. Customer Inline means its only applied to one group or user.

Now we know what each group does we can apply they to users accordingly.



We want user-1 to supply support for S3 so they would be put into group S3-Support. Go to that group and click on users. Under users click add users.

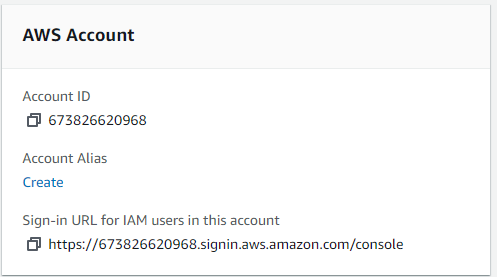


Select user-1 to add it to the group. Repeat for user 1 and 2.

We want user-2 to support the EC2 instance. Add user-2 to the EC2-Support group.

We want user-3 to administer the EC2 instance. Add user-3 to the EC2-Admin group.

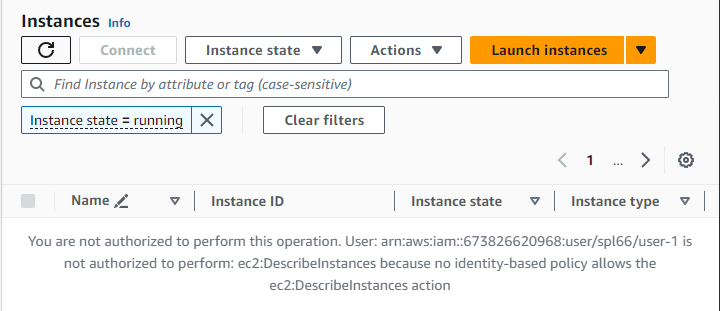
Now test if these policies work.



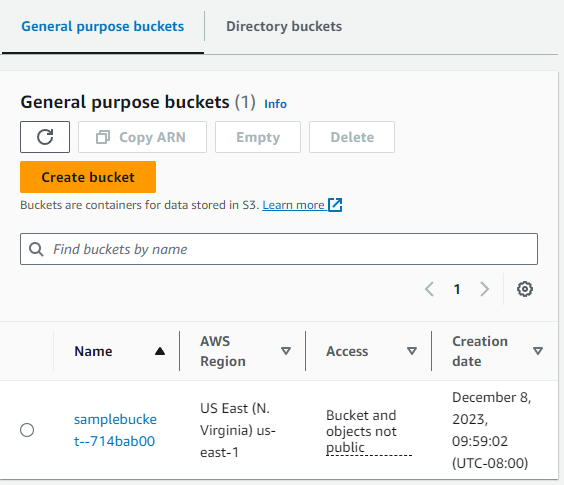
Still under IAM, click on dashboard to see the IAM user sign in URL to test. Copy the link and paste it into a browser.



Putting in the User-1 login information you are now in that account with access to S3 buckets.



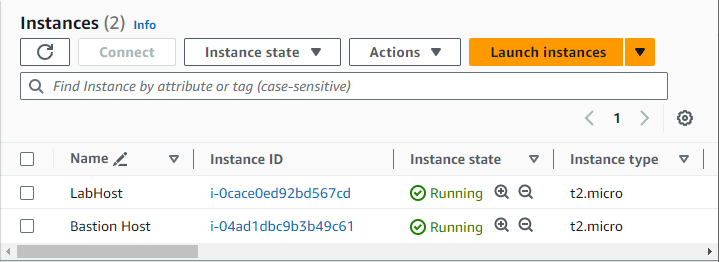
When you try to look at EC2 instances, you do not have permissions to like is supposed to happen.



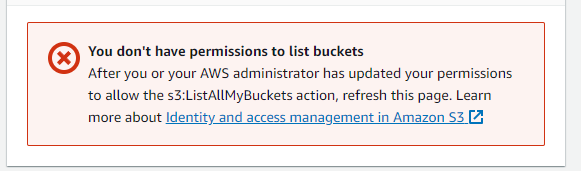
User 1 does have permissions to view buckets.



Now in user-2 you should be able to view EC2 instances.



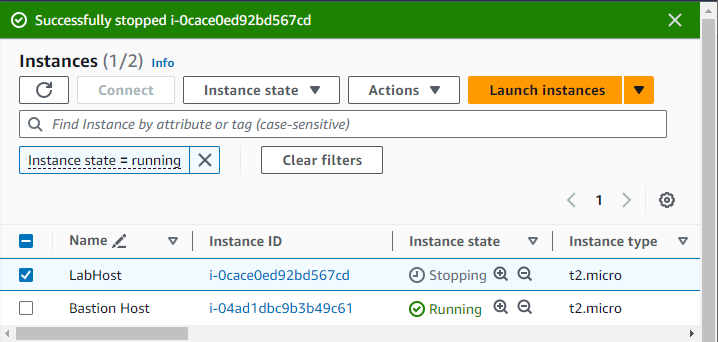
User-2 can see the instances but cannot edit them like we expected. When user-2 tries to stop an instance, nothing happens.



User-2 cannot view the buckets like expected.

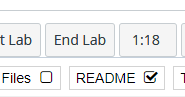


Now sign into user-3.

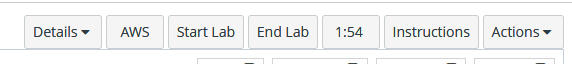


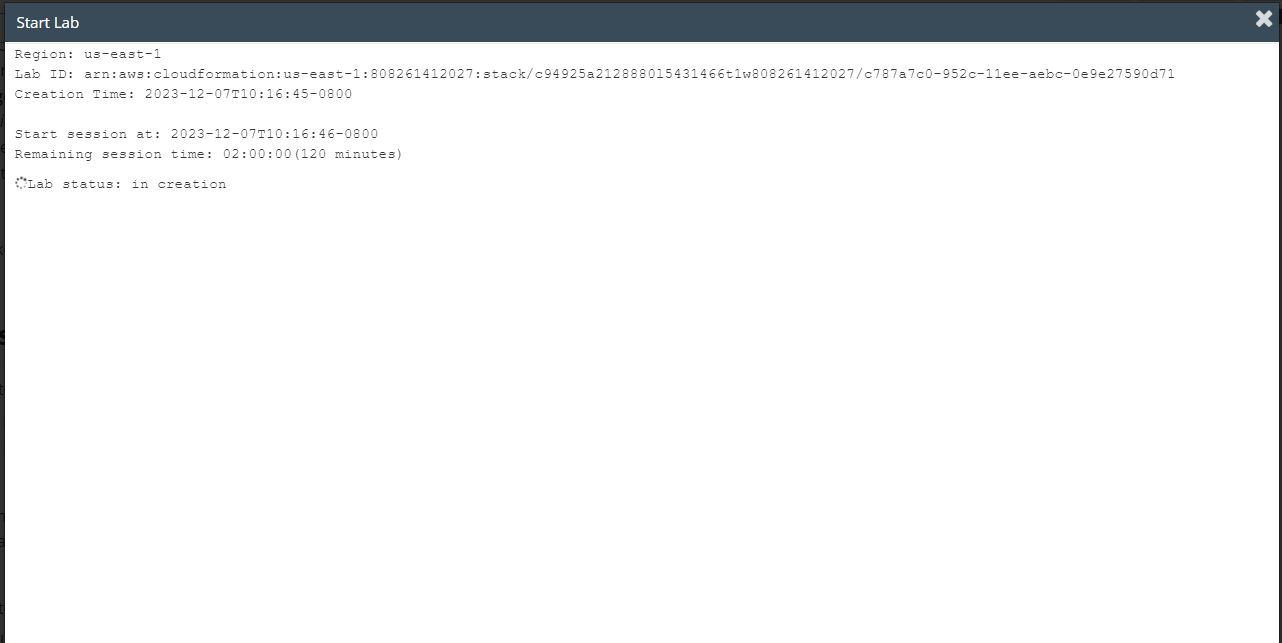
User 3 can successfully stop an instance because of their permissions.

Now click end lab and it’s complete.

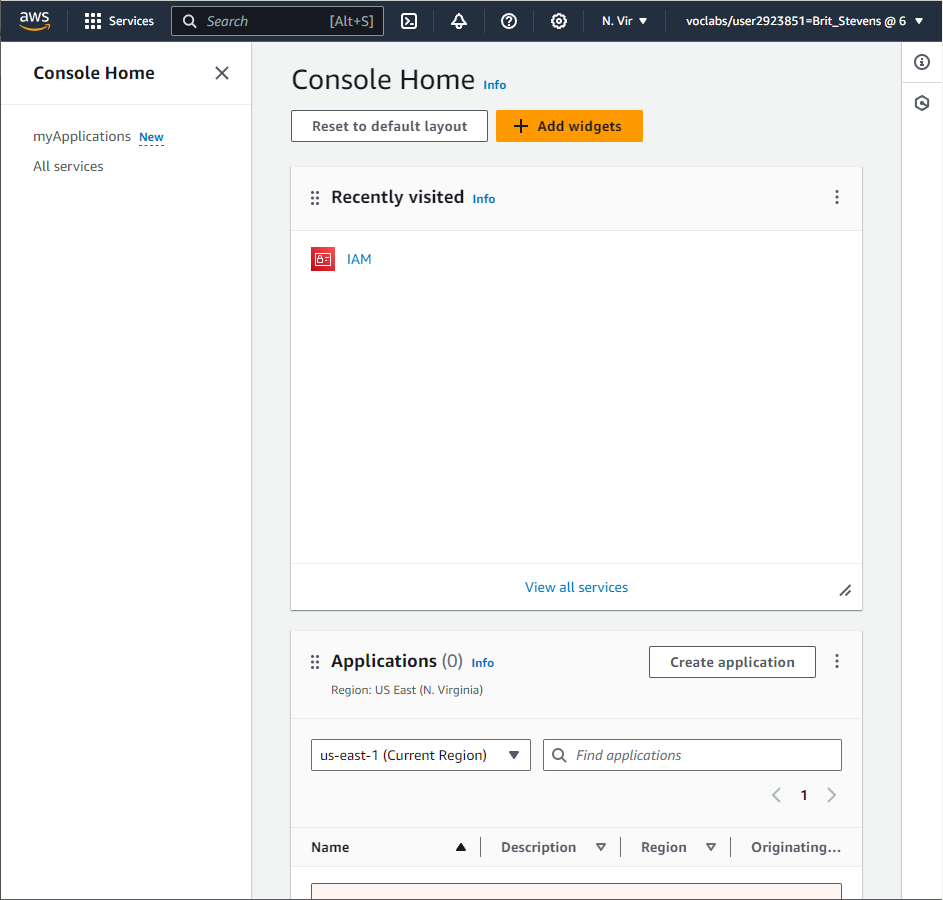


**Lab Summary: AWS VPC and Web Server**

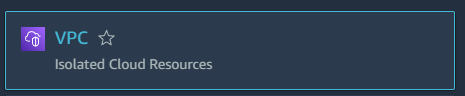
****

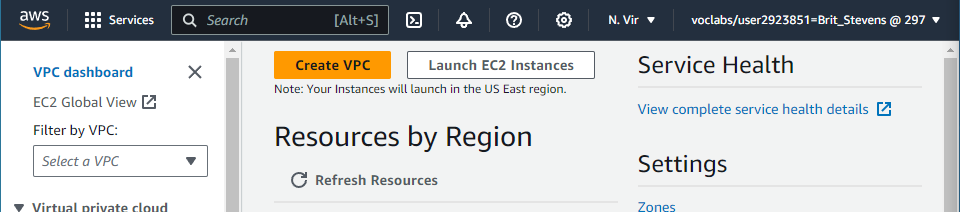
****

click start lab and wait until the prompt box says lab is ready. Once lab is ready click AWS to go to the EC2 instance.

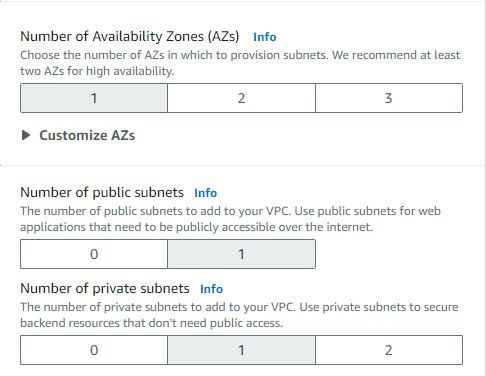
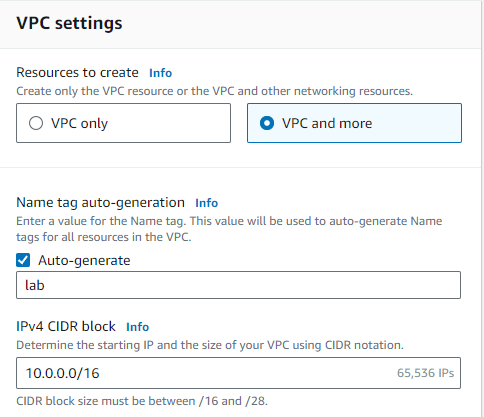


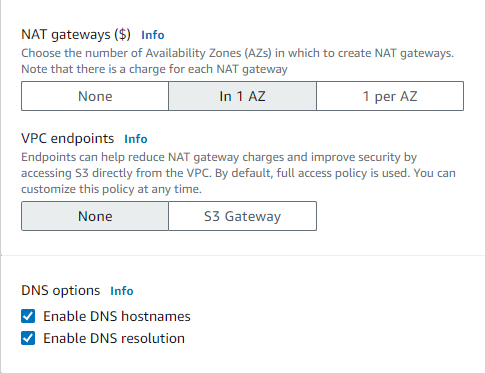
You will see a website like this. Click on services and search for VPC.

****

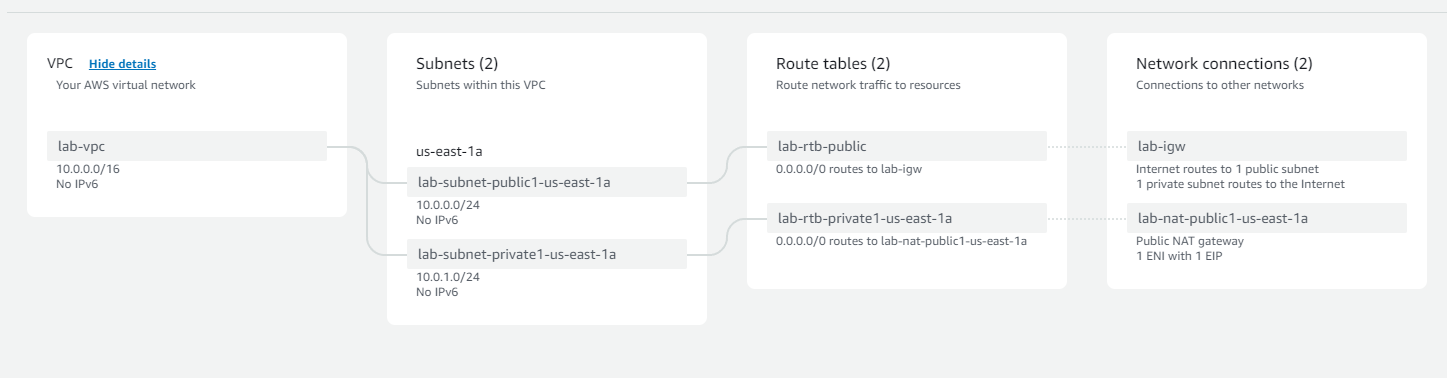
****

Once in VPC click on Create VPC.

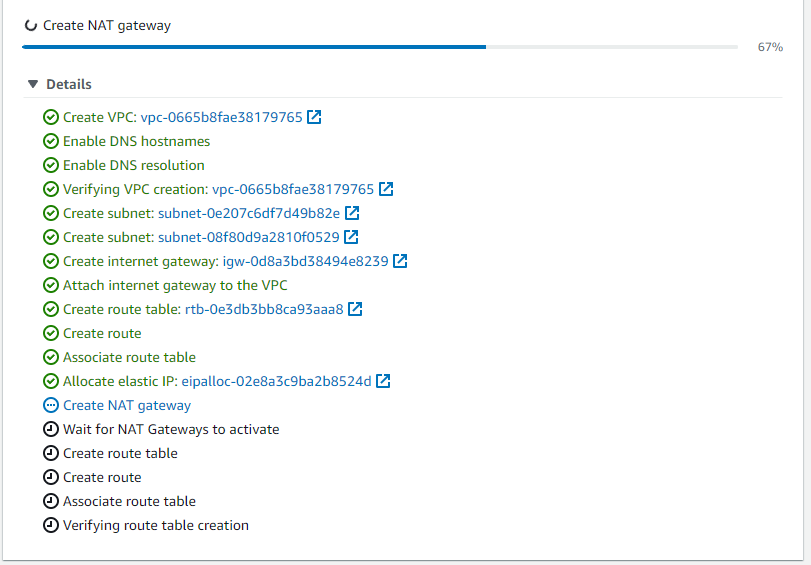




Under the creation settings for the VPC select VPC and more to create the VPC. Change the tag for auto generate to lab. Leave the IPv4 CIDR block the default. Change the number of Availability Zones to 1. Set both private and public subnets to 1. Expand the Customized subnets CIDR blocks and set public to 10.0.0.0/24 and private to 10.0.1.0/24. Set NAT gateway to In 1 AZ and VOC endpoints to none. Leave both DNS options enabled.

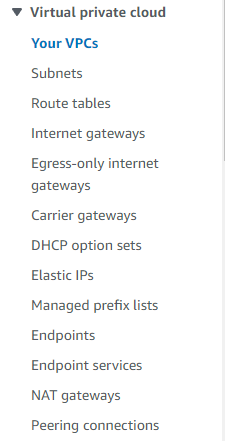


After confirming the settings are correct by looking at the preview, click create VPC at the bottom of the page.

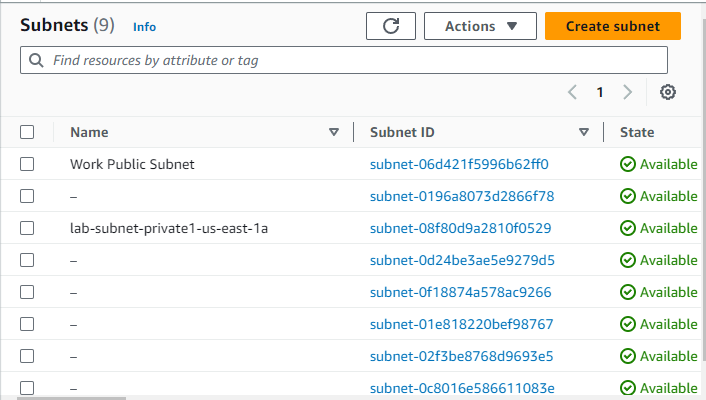


Wait for the prompt to complete and click view VPC.

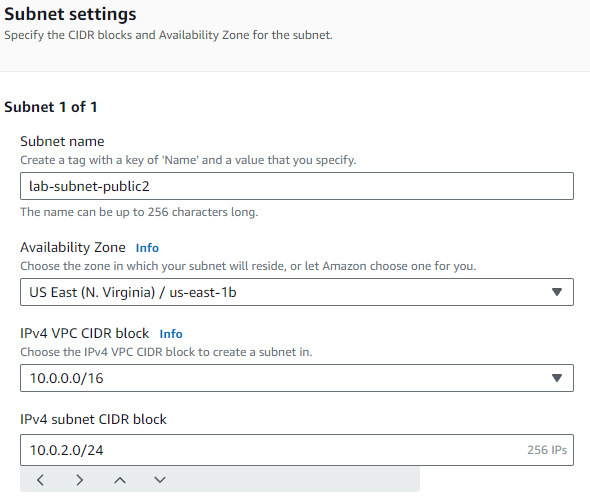
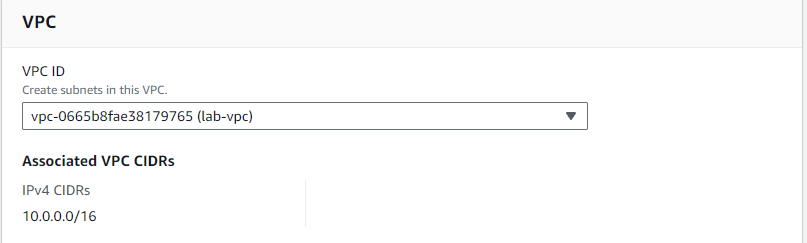
The VPC wizard has now made a NAT gateway and Internet gateway to allow communication between the VPC and internet.



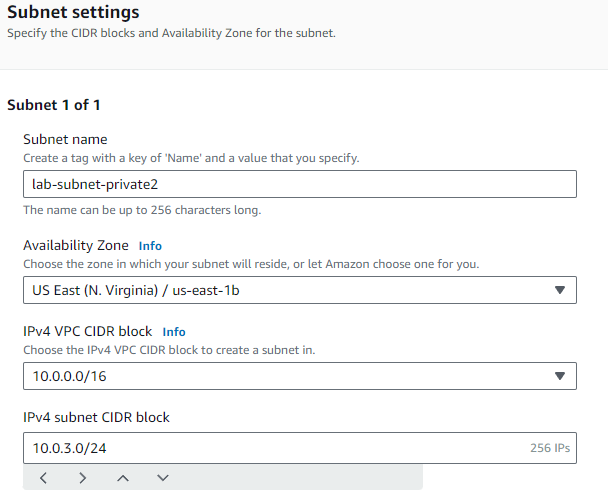
You can view your various settings by clicking on any of the options.



Select subnets and create a new subnet.

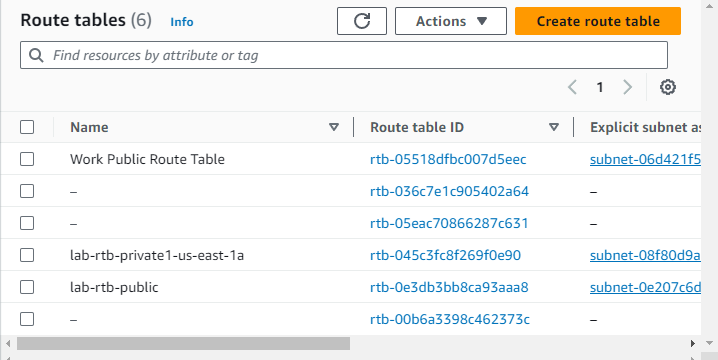


Select the VPC you configured previously and use the listed settings to create a new public subnet to allow for devices on your VPC.

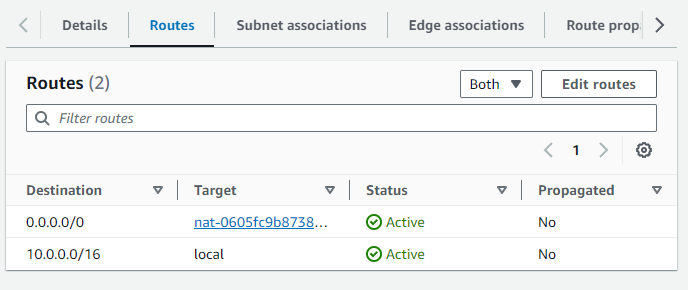


Create one more subnet for private IPv4 addresses using the listed settings.

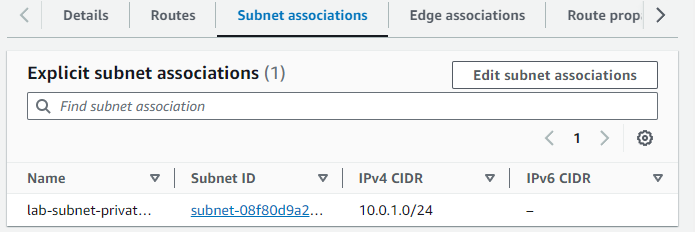
Now that you have created a new private subnet you will route any internet bound traffic to the NAT gateway.



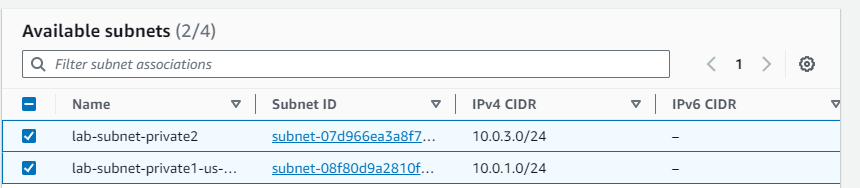
Go to Route tables and select lab-rtb-private1-us-east-1a.



The route 0.0.0.0/0 is a default route to the NAT which allows devices to reach the internet.



This route is only associated with our first private subnet so you must add the new private subnet by clicking edit subnet associations.

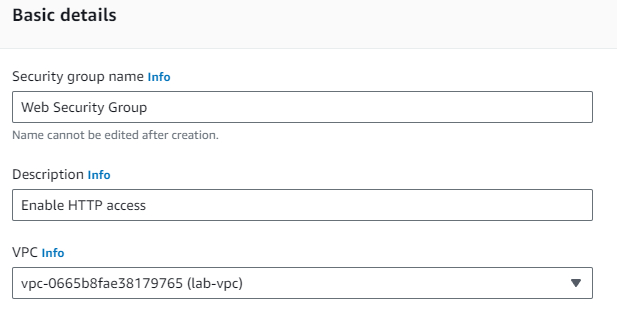
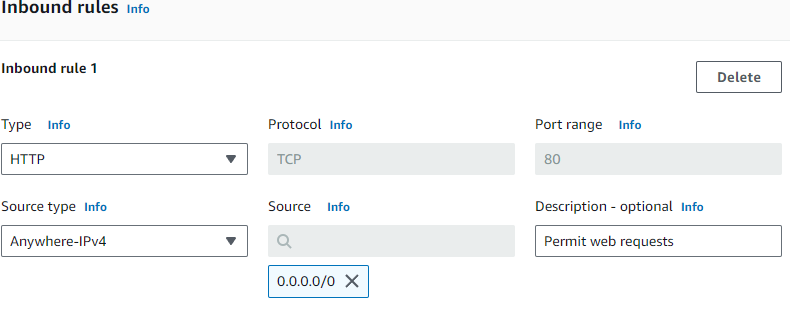


Click save at the bottom after adding new subnet.

Repeat steps with public routes.

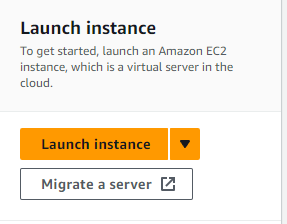
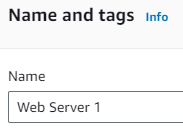
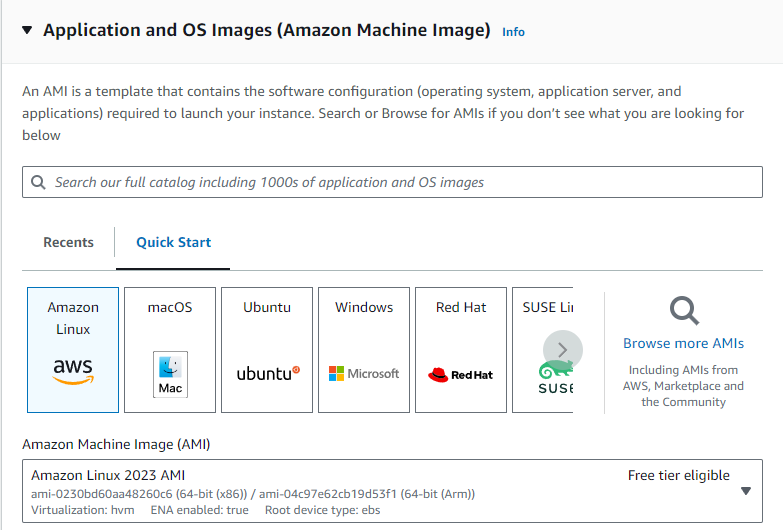
Select security groups and create new.

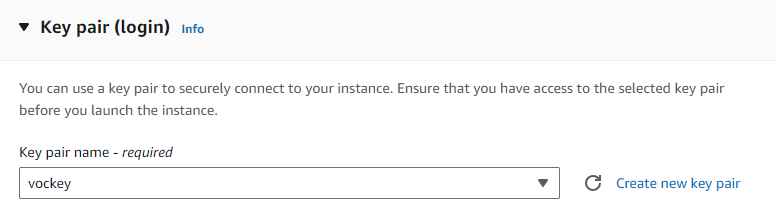
Use the shown settings and create the security group using the create at the bottom of the page.



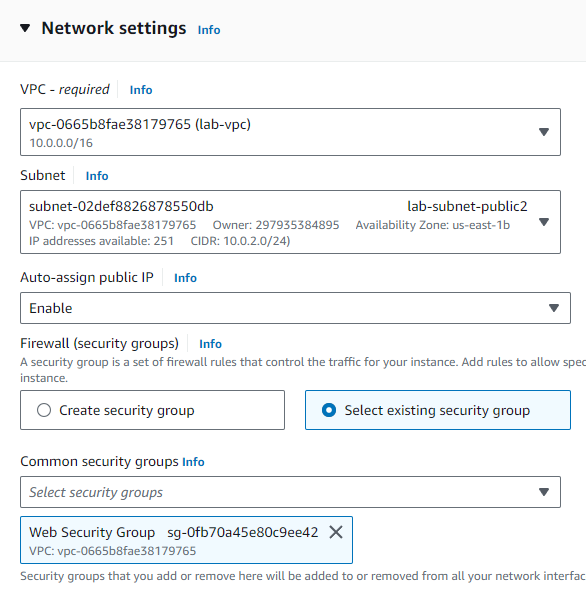
Now search for EC2 as you will put your new EC2 instance into the VPC.

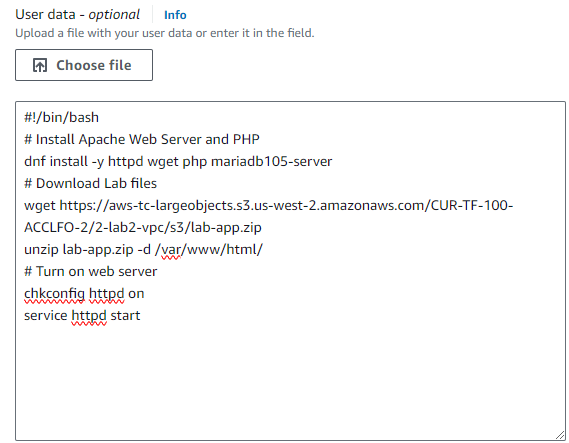
Choose launch instance with the name Web server 1. Select amazon Linux as the AMI. Use the default Instance type.



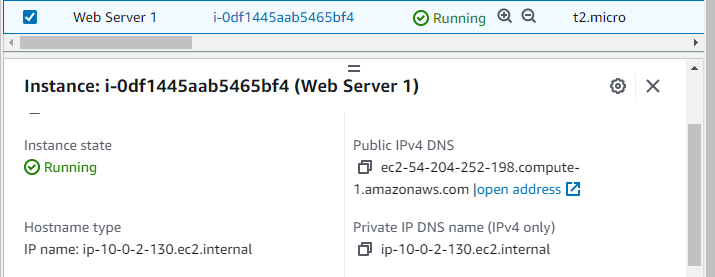
Use vockey as your key pair.

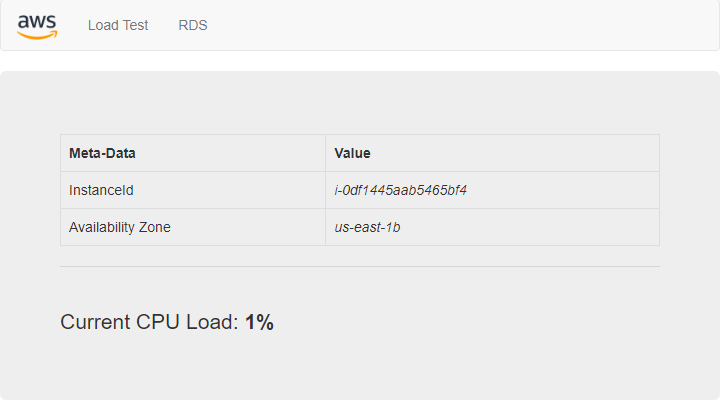


Under network settings use your VPC and the 2nd public subnet you made. Also select the security group you made previously. Leave the storage as default and expand advanced details.



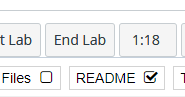
Copy this code into user data. Now launch the instance.

select your instance and copy the Public IPv4 DNS to paste into a web browser.

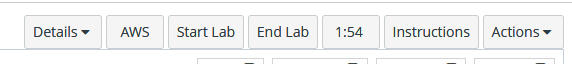


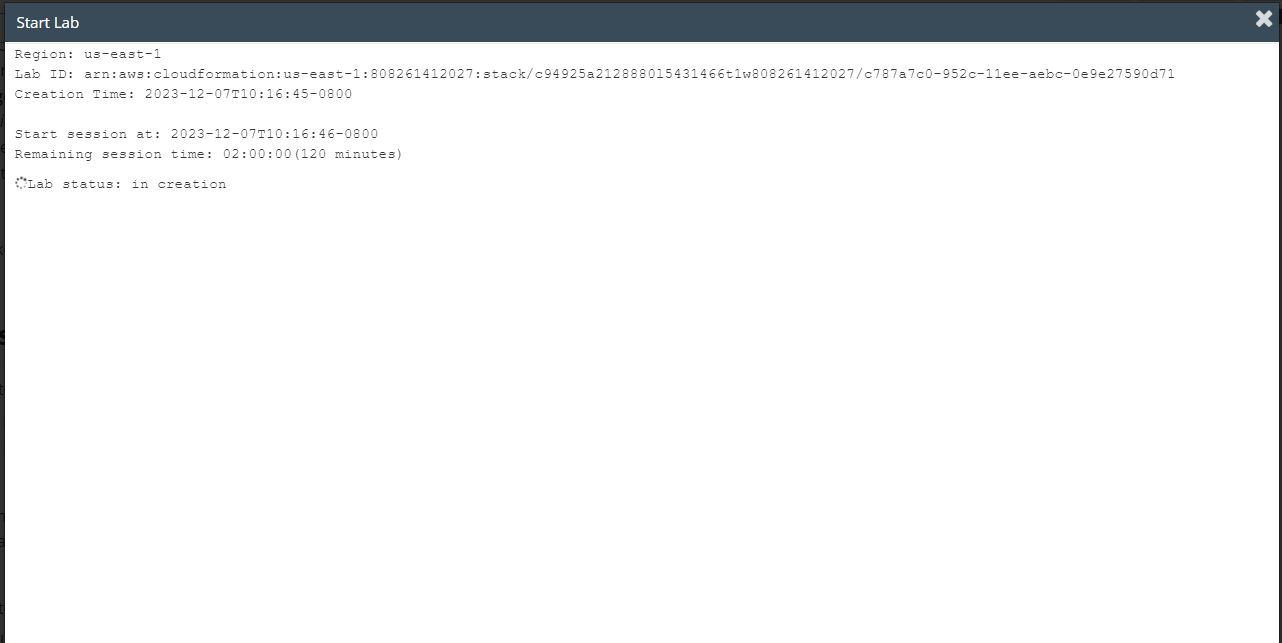
You will see these details about the VPC confirming in is available on the Internet.

Now click end lab and it’s complete.

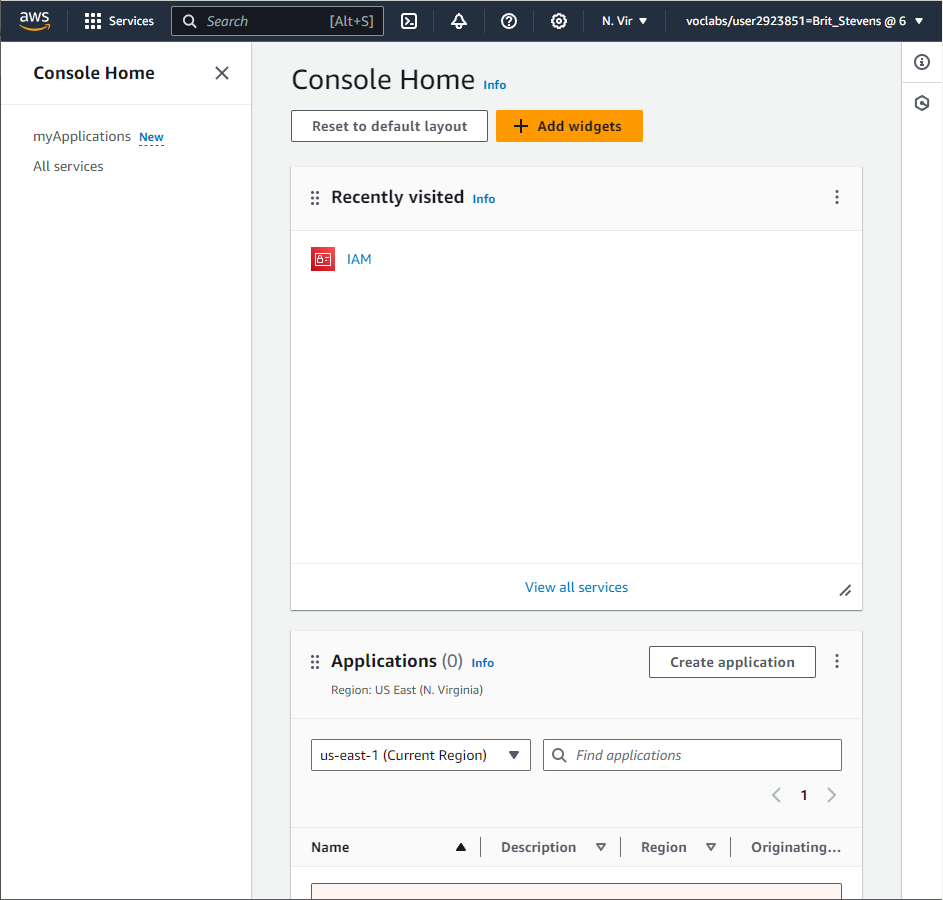


**Lab Summary: AWS EC2 Configuration and Management Lab**

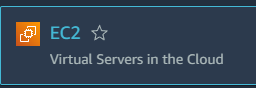
****

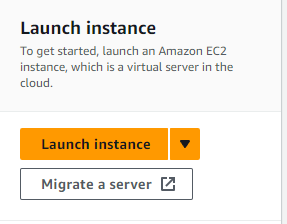
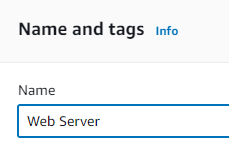
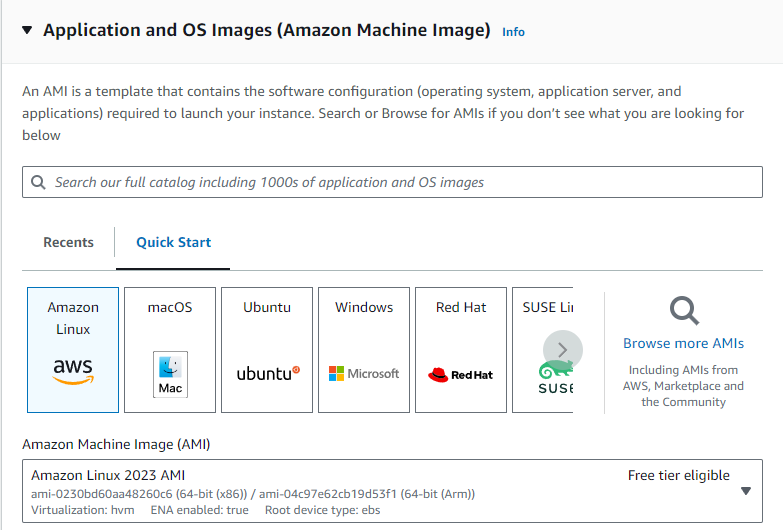
****

click start lab and wait until the prompt box says lab is ready. Once lab is ready click AWS to go to the AWS webpage.

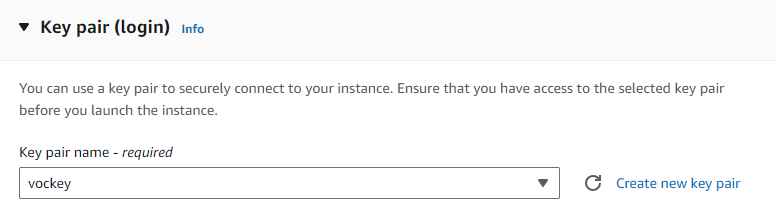


You will see a website like this. Click on services and search for EC2.

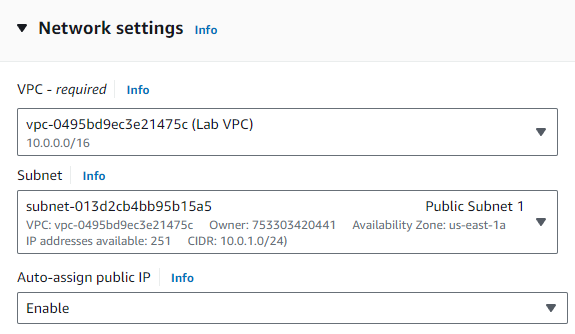
****

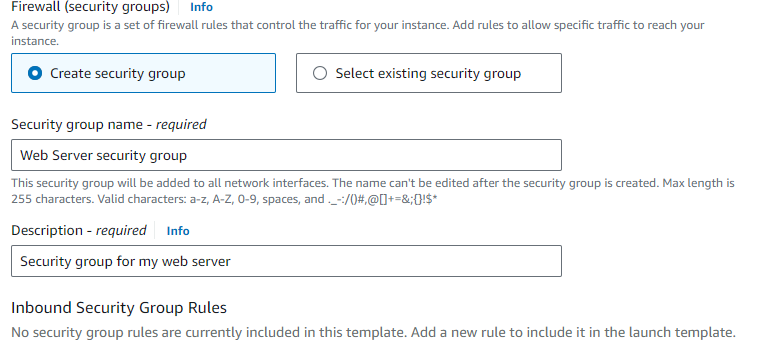
Choose launch instance with the name Web server. Select amazon Linux as the AMI. Use the default Instance type.



Use vockey as your key pair.

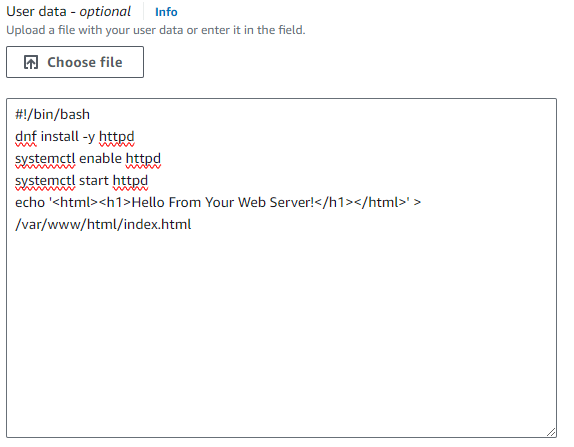


Under network settings use lab VPC and the 1st public subnet.



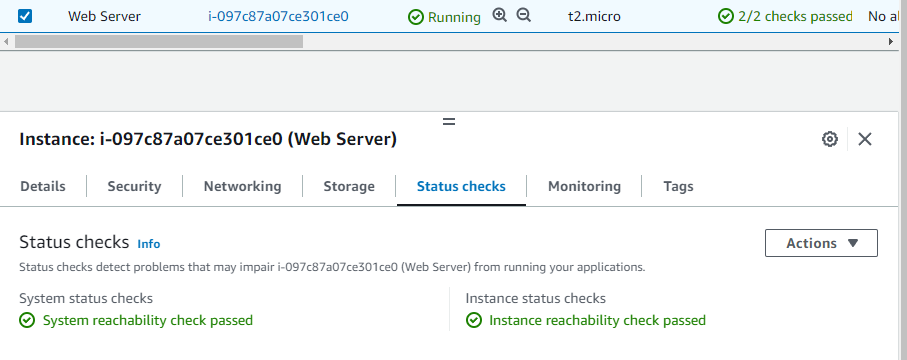
Under Firewall security select create new security group using the shown name and description. There is an inbound security group rule that you must delete. Leave the storage as default and expand advanced details.



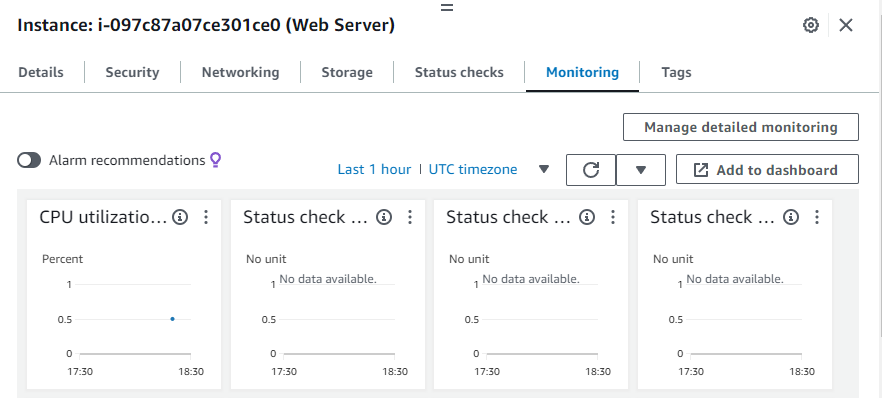


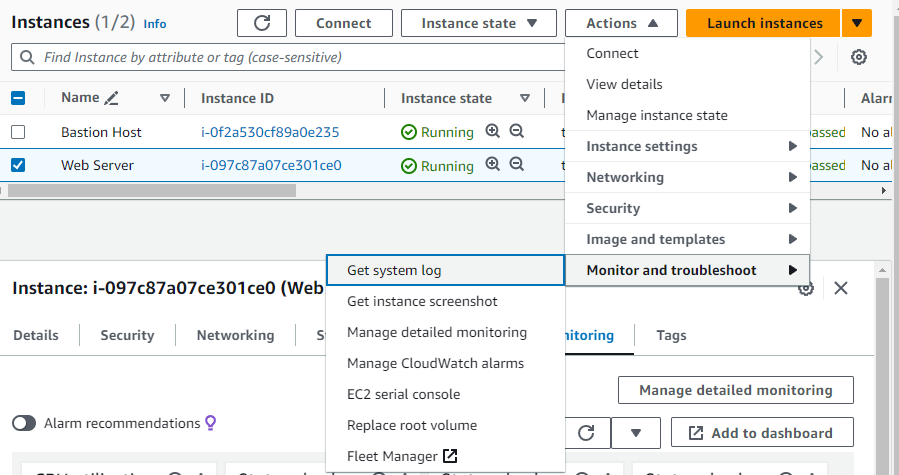
Enable termination protection and copy the shown text into user data. Now launch the instance.

Select view all instances and select the one you just made.



Check that the server has passed both status checks. By clicking monitoring you can view the resources the instance is using.



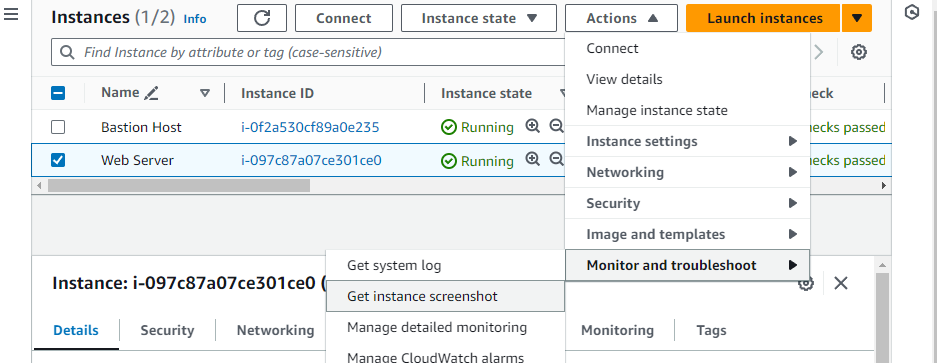


Click on actions and select get system log under monitor and troubleshoot.

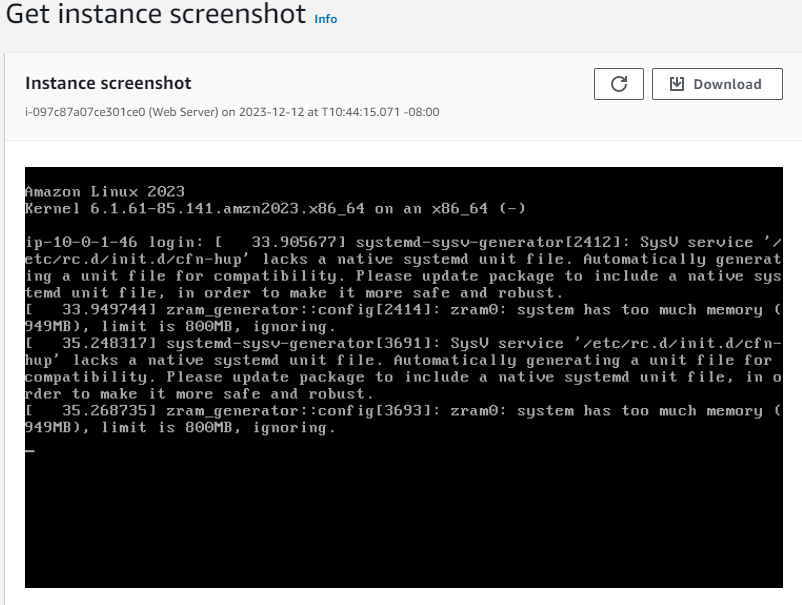


Here you can view some of the resources we downloaded with the user data earlier.

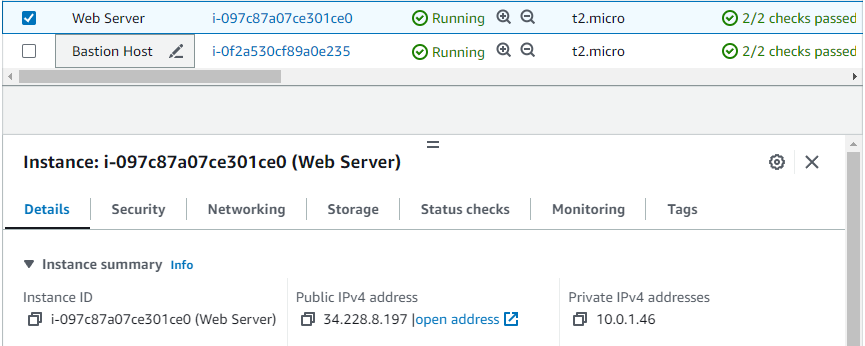
Click cancel to go back.



Click get instance screenshot.



These are different ways to monitor your instance.

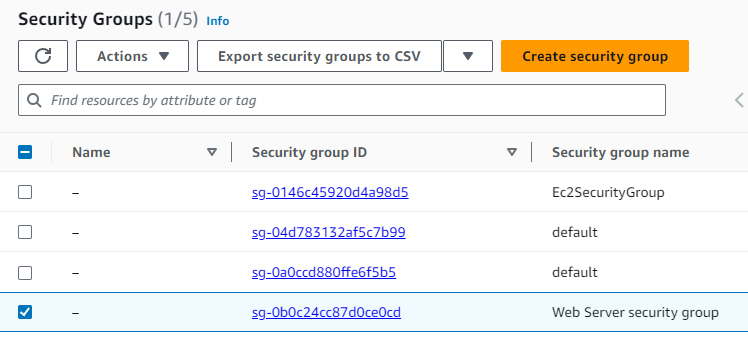


Go back under details and copy the public IPv4 address to access the EC2 from a web browser.

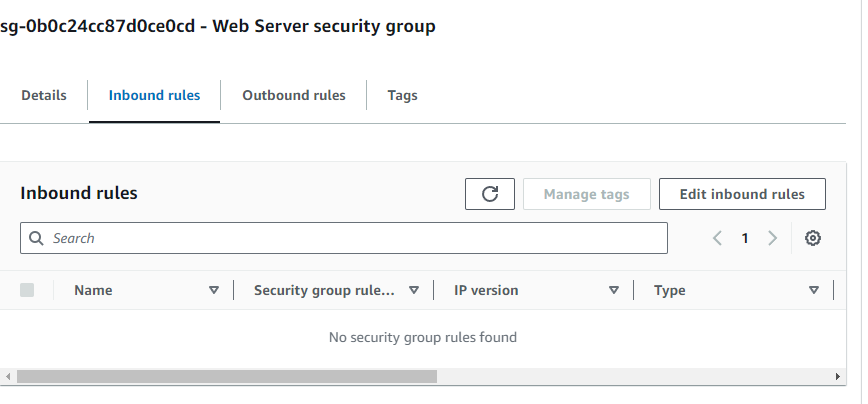
This should not work because we have no inbound rules apply meaning nothing can access the EC2 from the internet.



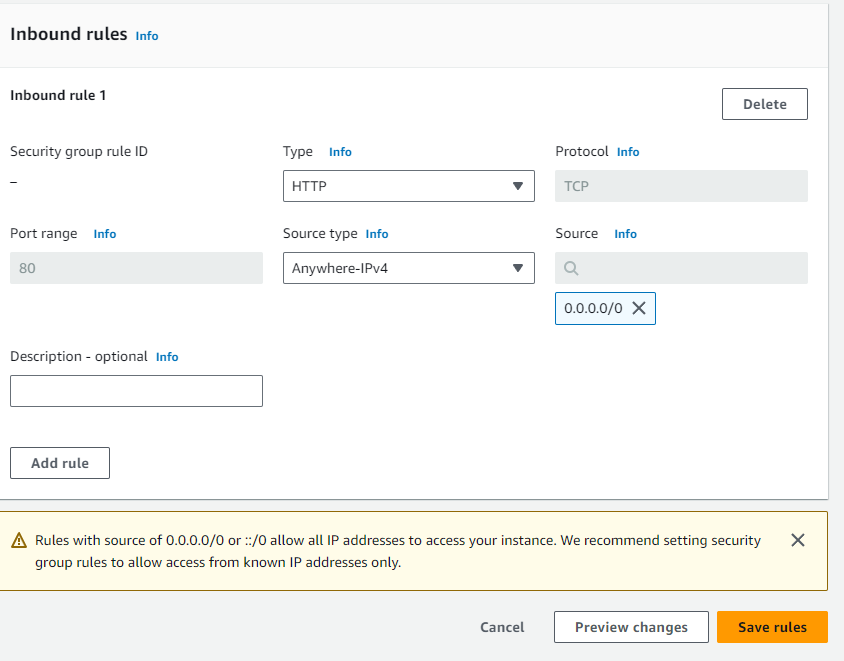
Navigate to the left panel and select security groups.



Find the web server security group and select it.

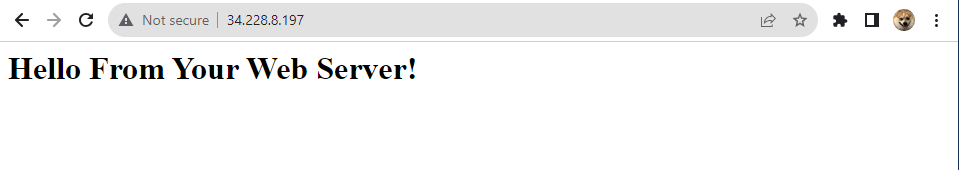


Select edit inbound rules to create a new rule.

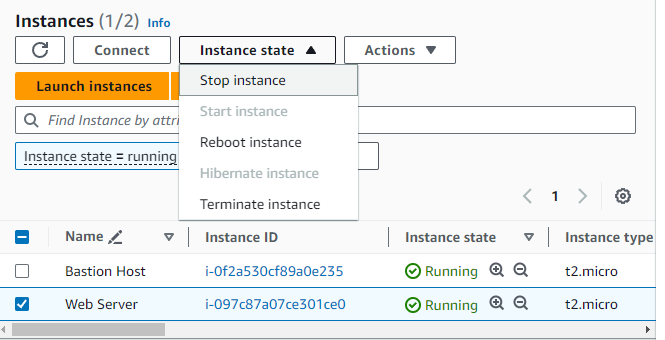


Add this rule to allow any IPv4 traffic using HTTP and click save rules to apply.

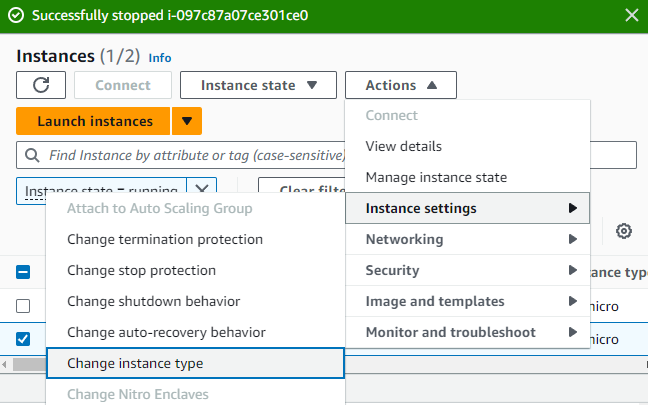
Go back to the web browser with the IPv4 address entered earlier and refresh. It should now load.



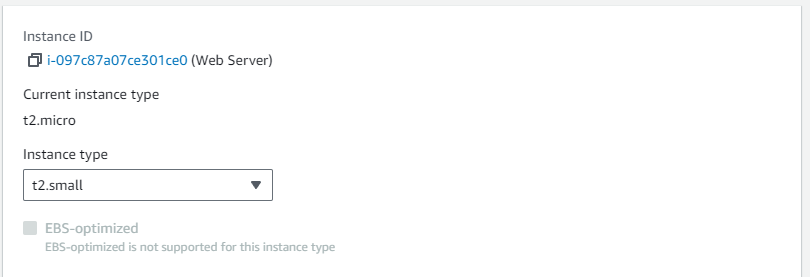
If you ever need to resize your instance you must stop it first.



Go back to the EC2 instances and select web server. Under instance state click stop instance.



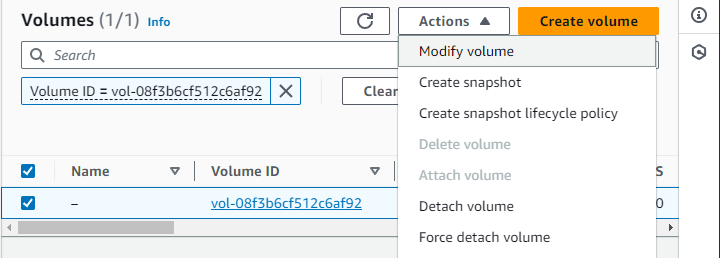
Under actions select change instance type.



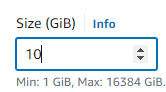
Select t2.small and save changes. This will double the memory of the EC2.



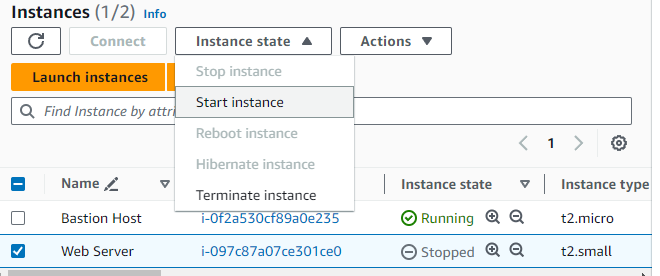
Click the Volume ID.



With the volume selected click actions and modify volume to give the EC2 more data storage.

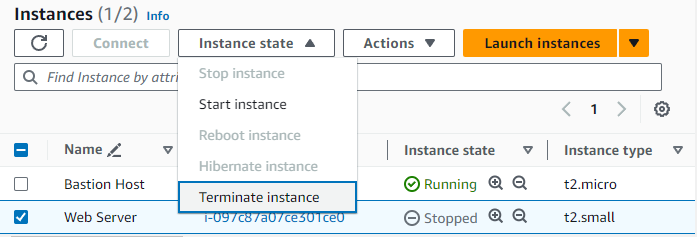


Enter 10 to increase the size by 2 Gigabytes and select modify.



Start the instance again.

You will now test termination protection.

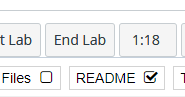


With web server selected click terminate instance.



It will fail because we have termination protection on. If you turn this off, then you can terminate your instance.

Now click end lab and it’s complete.



**Problems:**

* No problems in configuration
  + The lab is step by step so if you followed instructions there was no room for error.
* Thinking the lab failed
  + To start the lab, you must wait 5-10 minutes as it loads so I have closed it accidently the first few attempts.
* Errors with AWS
  + Occasionally the lab would begin in the wrong region and make it unusable.
    - Restarted lab until it used the correct region.
  + Some days AWS would be inaccessible to us halting our progress.
    - Needed to wait until AWS would allow us to login again.

**Conclusion:**

These were important labs because they were our first exposure to a real AWS instance and showed us how their web page functions. It also showed us permissions function and the role that groups play in security and distribution of those permissions. We also learned how configure EC2 instances and S3 buckets for the use case scenarios we were given. I was very skeptical that these labs would aid our learning because you just follow the guide and complete the but rewriting the guides in my own words truly does help. When I didn’t know a concept or service I was supposed to configure, I would add it to the background information and research what it was. This cycle of proceeding through the labs and listing terms made me understand every term to a fuller extent and did help my learning. I believe that is the biggest takeaway from these labs even though the services and what they provide are essential. Learning how to better understand concepts even if you don’t know what they are at first is an indispensable skill and can be applied to anything even out of networking courses. Overall, I believe these labs did help my learning of the AWS services and what they provide to better prepare me for the AWS certification exams.