Amazon Web Service Documentation

Name: Ravishanker Vishwakarma
AKA: theblackthreat
Documentation Date: 18-05-2021
Purpose: Tutorials for College level.

Visit Website: https://theblackthreat.wixsite.com/theblackthreat
AWS: https://aws.amazon.com/

Contents

1.Objective: Creating Amazon EC2 instances with Microsoft Windows

Requirements : Amazon web service account weather it is free or paid tier.

EC2: EC2 stands for Elastic cloud computing used to create instance hosting services and storages publicly and privately.

Steps:1 Open AWS console to create a window instance.

Navigate this URL: https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:

Step 2: Launch Instance ---> choose a OS windows (windows based 2019 with free tier eligible)

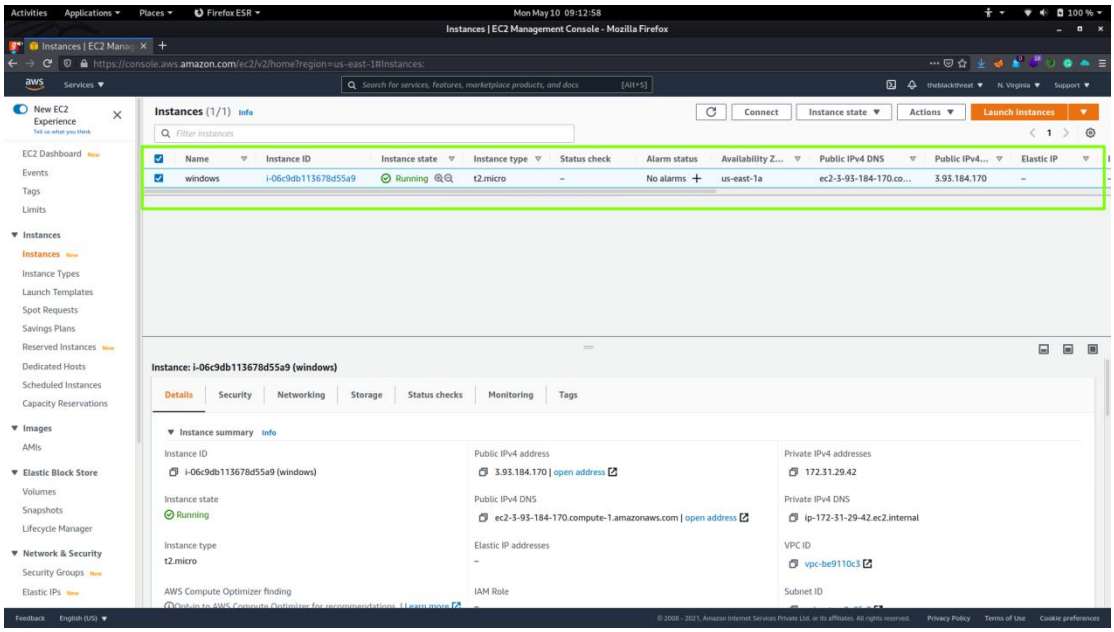
Step 3: check the box named **t2.micro** for free tier ---> click next to configure

Step 4: If you want ot choose specific VPC or virtually computing onto you specified space then you can select the vpc but in my case i'm putting it as default.---> click next to add storage.

Step 5: We can keeping 30 GB but if you want to more storage then change it as per requirement.

Step 6: Review and Launch.---> launch
Step 7: Now you are popped with a pair key click to create a pair key if you haven't created it yet. and named as "window-key" ---> download key pair. ---> save it.

Step 8: launch Instances ---> view Instances



Step 9: check your instance and then click to the connect at above pannel. ---> click on RDP client --->get password ---> browse your saved key named as"windows-key.pem" ---> decrypt password.

Now just save the passwrod for the future.

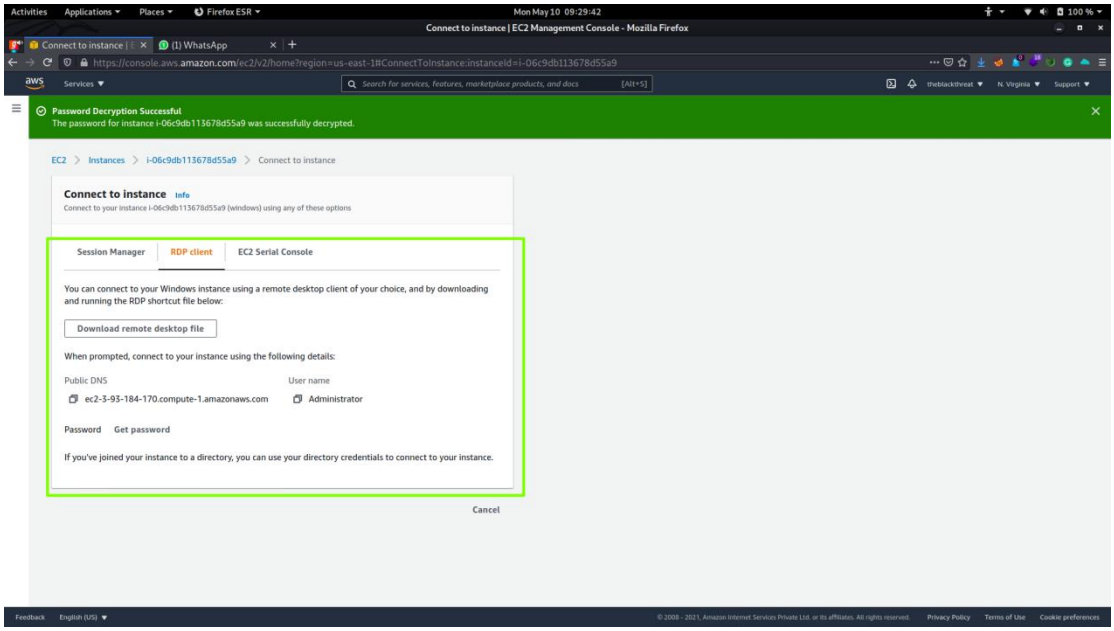
we have completely created A windows EC2 instance.

Step 10: to open it on your device. click on "download Remote desktop file" and save it

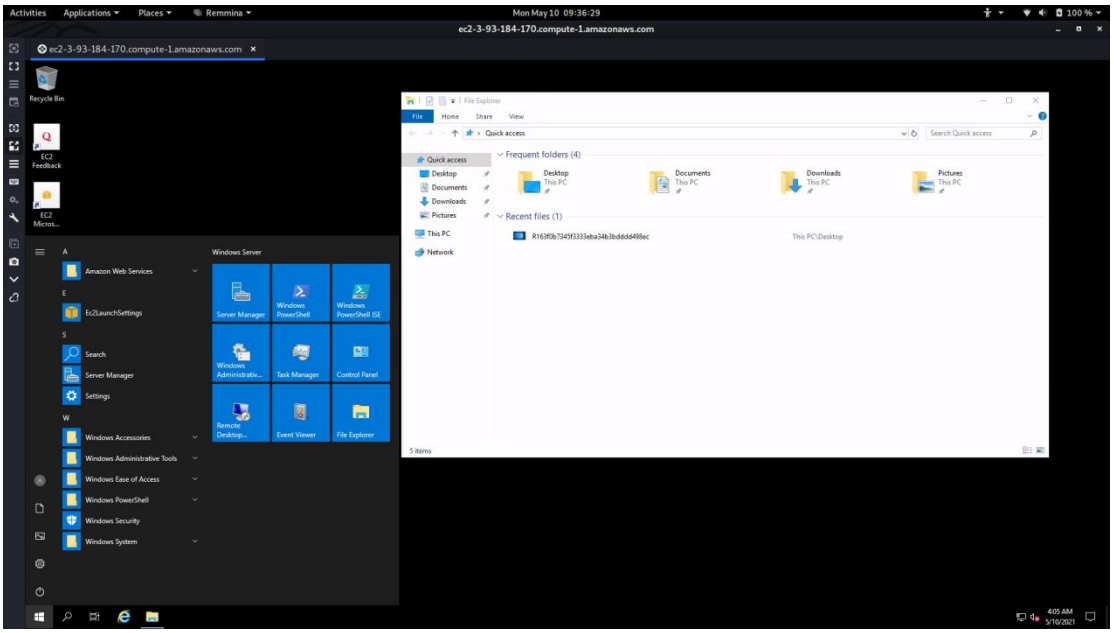
Open RDP client application on windows im using my Linux OS so open

for windows:
https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/clients/remote-desktop-clients
for linux : Run “Remmina”

double click on downloaded file windows.rdp file enter password and username form the



use these credential to login on RDP file and it will run on your machine.



Contrats we have done this.

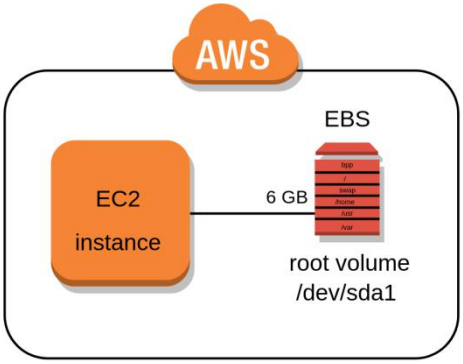
2. **Objectives**: Working with Amazon Elastic Block Store (EBS)

Requirements : Amazon web service account weather it is free or paid tier.

EBS: EBS is store drive for EC2 Instances(just as hard drive). EBS is provide block level volume storages for EC2 instances. EBS volumns are highly reliable and avialable drive that can be attached with any instance that is in the same availibility zone.
All instances have its own volumns but after termination of the instance volums are also deleted.. but EBS provide a comman volums that can be used with multple instances and the can share the storage. with an aditional volumes.
Volumes: root volumes and aditional volumes for instances. aditional volumes concern with EBS.
Snapshots: It is an Image to create a backups or used to create a duplicate

Step to handle it:

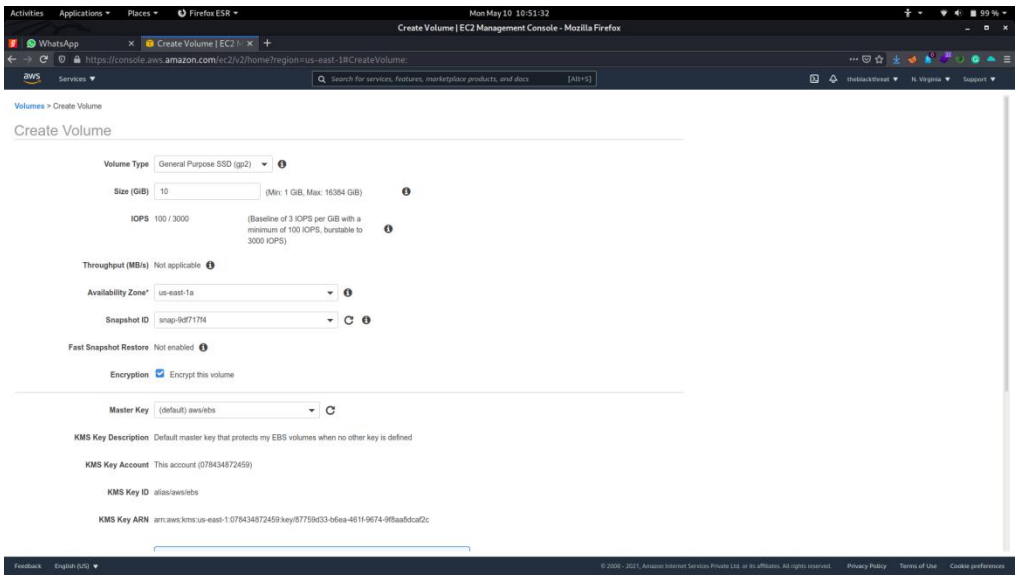
Step 1: Go to EC2 ---> click on Elastic block store
Contains with Volumes
Snapshots
Lifecycle manager.



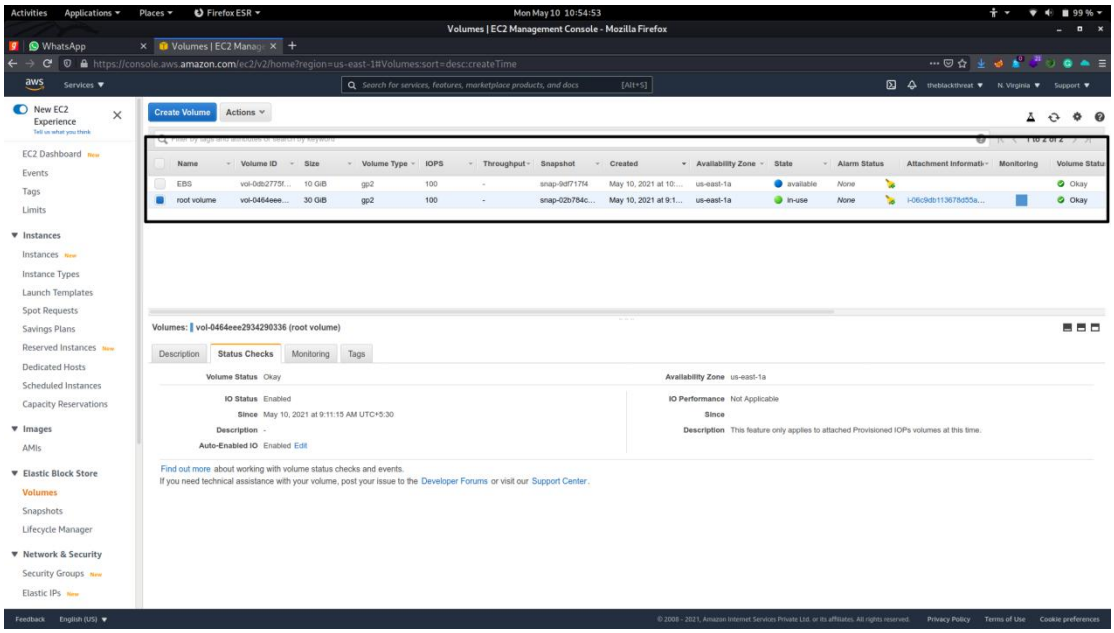
Before I have Created a Windows Instance having 30 GB root storage that is also termed as EBS volume or we can create another EBS volume SSD (In Physically we can tareget it as Hard drive but aws provide a SSD service for EBS to increase High performance level.)
Creation of New EBS volume:

Click on Create volume then fill the text area as per your demand.

IOPS: Input output per second (100/3000 by default)



Step 2: Create volume : now we have succesfully created volumes.



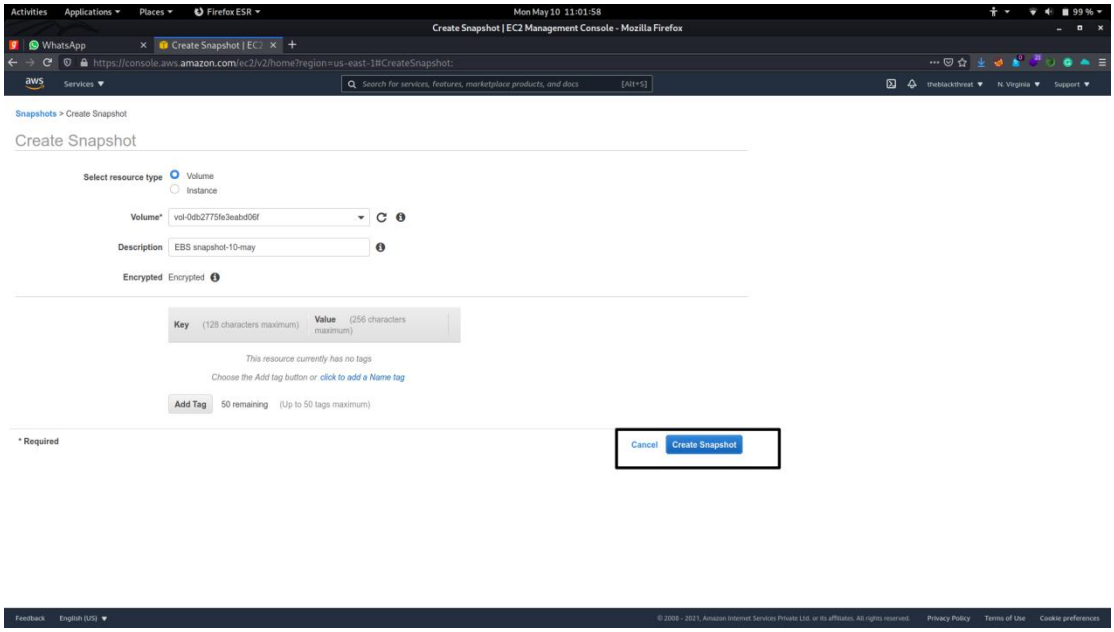
Snapshots: It it the important to know that if you have deleted volumes by mistake then you wont be able to recover it so large organization always creating their backups and store all data and Volumes.

To create snapshot let take a example :

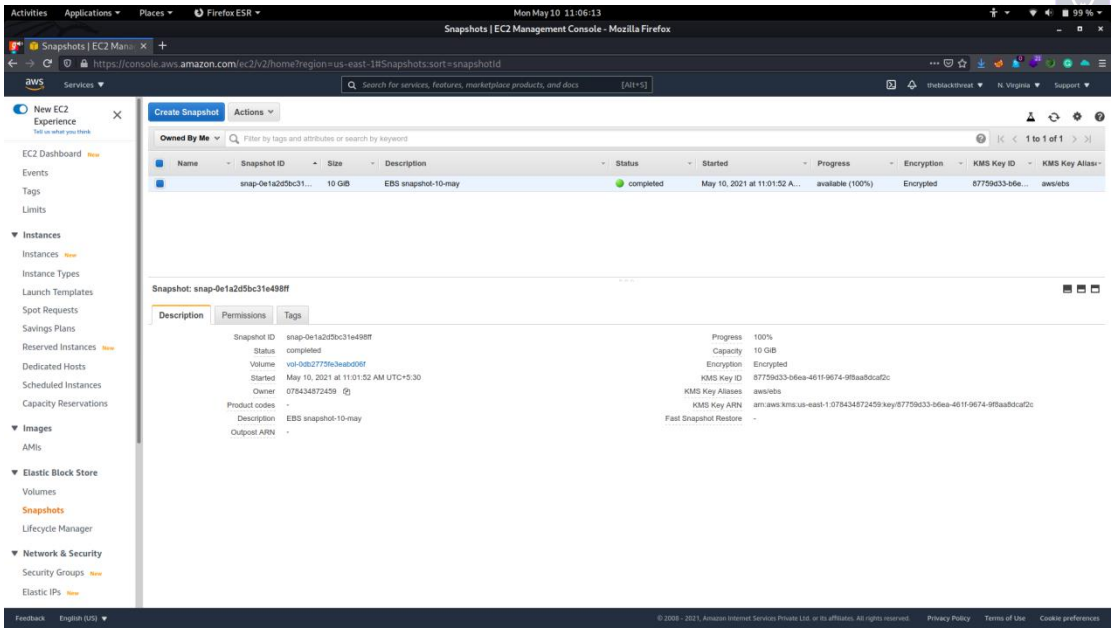
Step 1: Below Volumes there is an option named snapshot. click on it

Step 2: create snapshot chose your volume
Note: Id you want to create snapshot for your instance then click on instance.

Step 3: select you volume to snap.



It will take some time to complete you can delete it as well from the **action** nevigation bar.



We have successfully created a snapshot of EBS volume.
If we want to manage all lifecycle of Elastic block store(EBS) then click on **Lifecycle manager.**

Create Lifecycle policy
Data Lifecycle Manager enables you to automate the creation, retention, copy and deletion of EBS snapshots and EBS-backed AMIs. It also enables you to automate cross-account snapshot copy actions for snapshots that are shared with you, based on Amazon CloudWatch events.

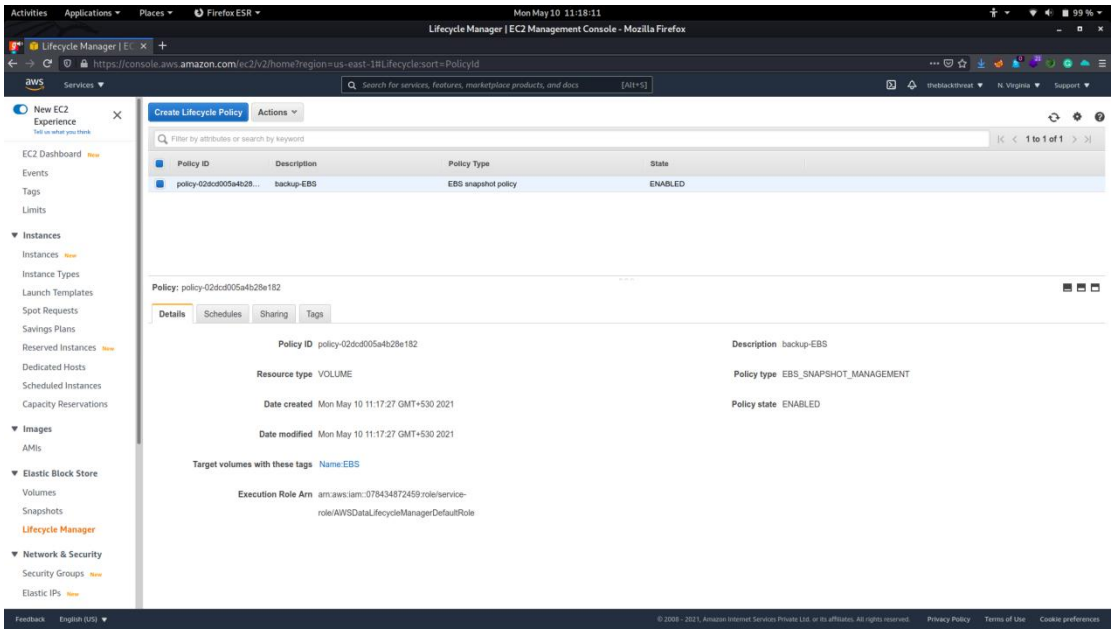
It will allow to create a periodic snapshot related to the policy

Step 1: I want to create a lifecycle of EBS so fill the information

IAM role : This policy must be associated with an IAM role that has the appropriate permissions.

Step 2**:** Set it by default\

Step 3: Set you policy schedule for a backups.then leave all option to defaults then click on **create**



3. **Objectives**: Build Your Virtual Private Cloud (VPC) and Launch a Web Server.

Requirments : Amazon web service account weather it is free or paid tier.

Building the Virtual private Cloud:

VPC : Virtual private cloud service allows us to create a virtual network for instance and services it allows to isolated services that we define we can use both IPv4 and IPv6. this service lies in network and content delivery section

To create a VPC we need to go for some steps other wise AWS allows us to default VPC.

Step 1: Go through the navigation bar and Find Netwrk and Content delivery then click on VPC.

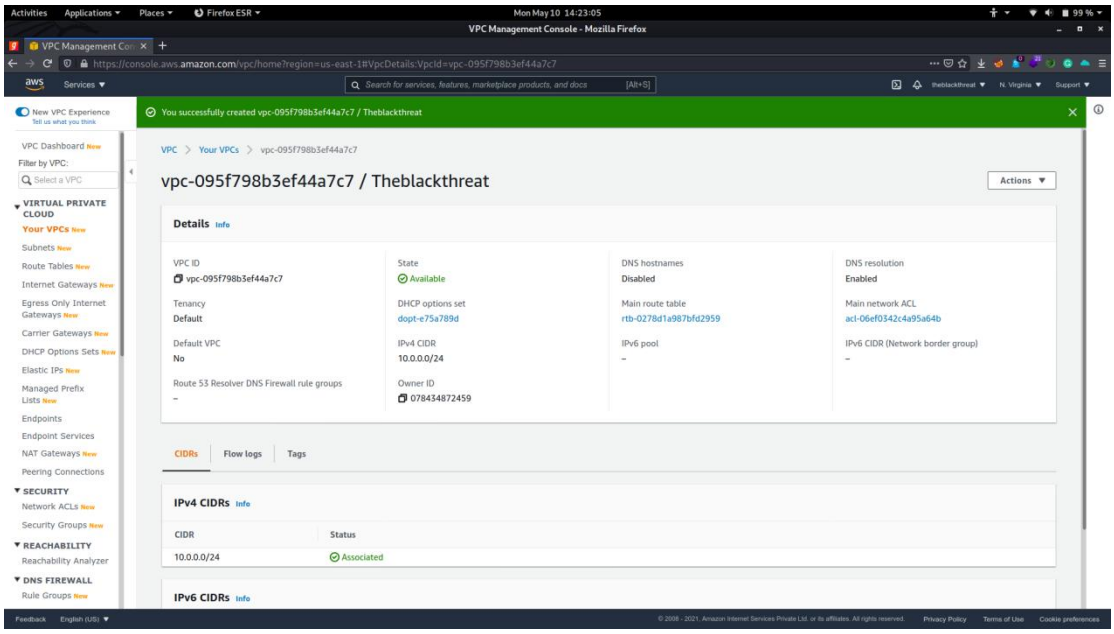
STep 2: Click on Create VPC

IPv4 CIDR block

You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. A CIDR block size must be between a /16 netmask and /28 netmask.

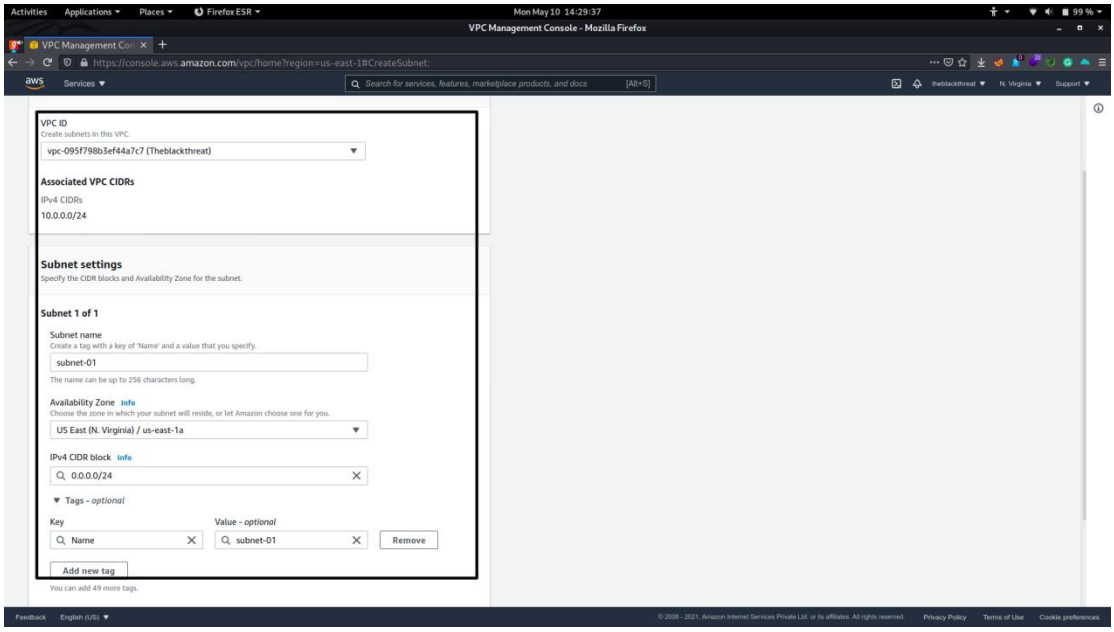
You can provide IPv6 but in my case i have small network so I'm creating it with IPv4.

Step 3: Create VPC we have created it so far but we need to understand it properly.

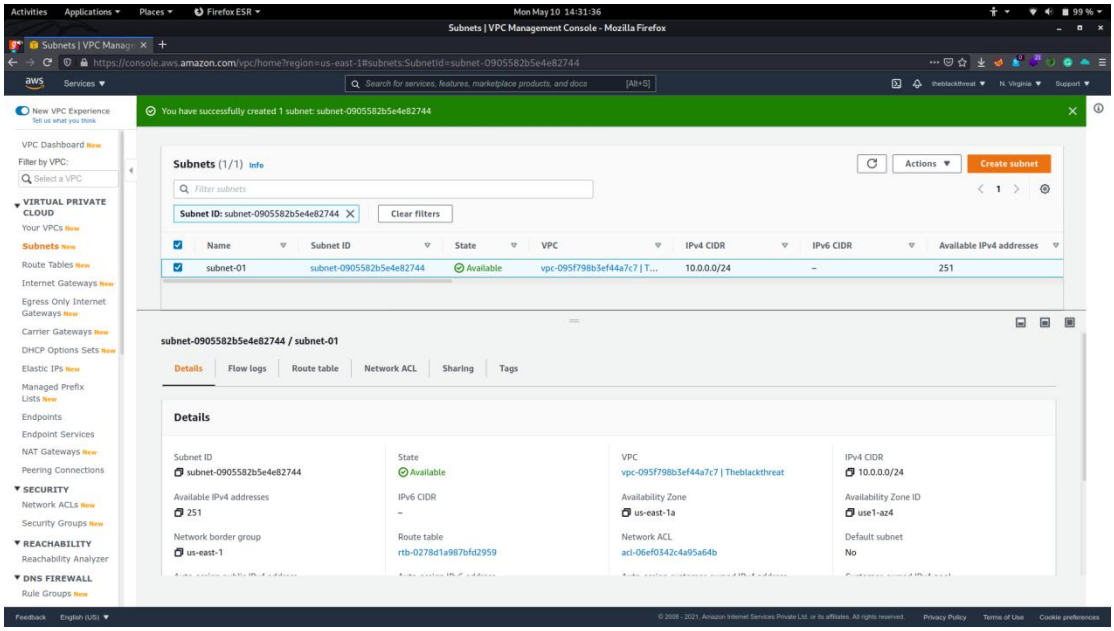


Subnet: A subnet is a range of IP addresses in your VPC. After creating a VPC, you can add one or more subnets in each Availability Zone.
In my case i have 6 Availability IP's Subnets
I'm selecting a first one for my small subnet.

Then click on Create subnet. ---> Select your VPC ID
Fill all required inputs



then click on Create Subnet.

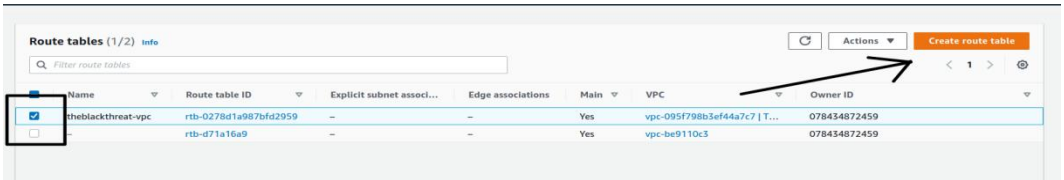


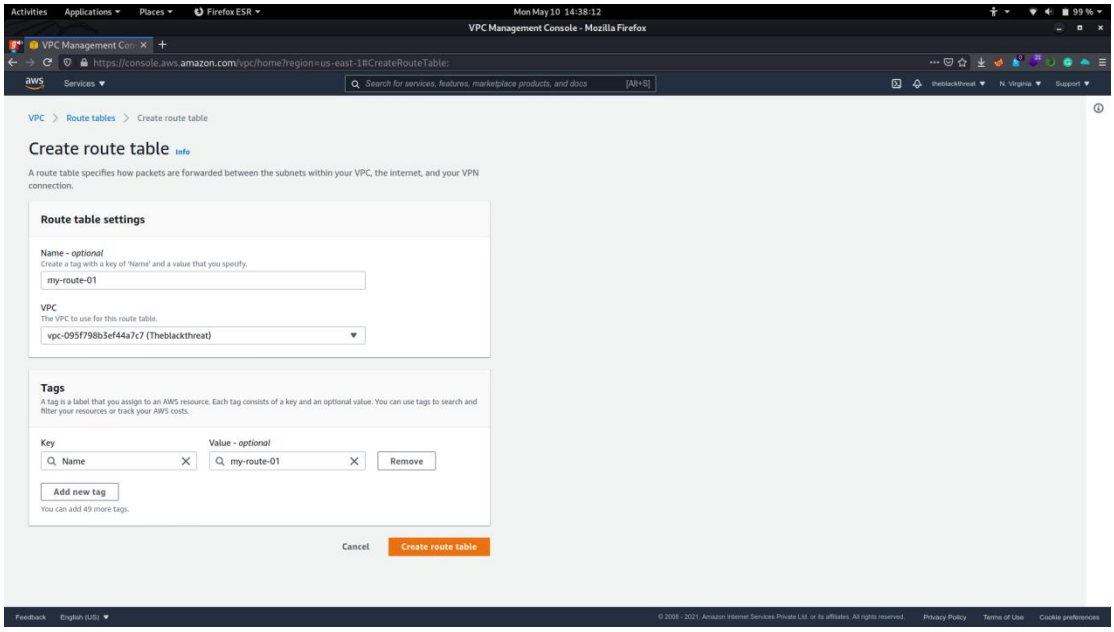
You can share your subnet as per your demand.

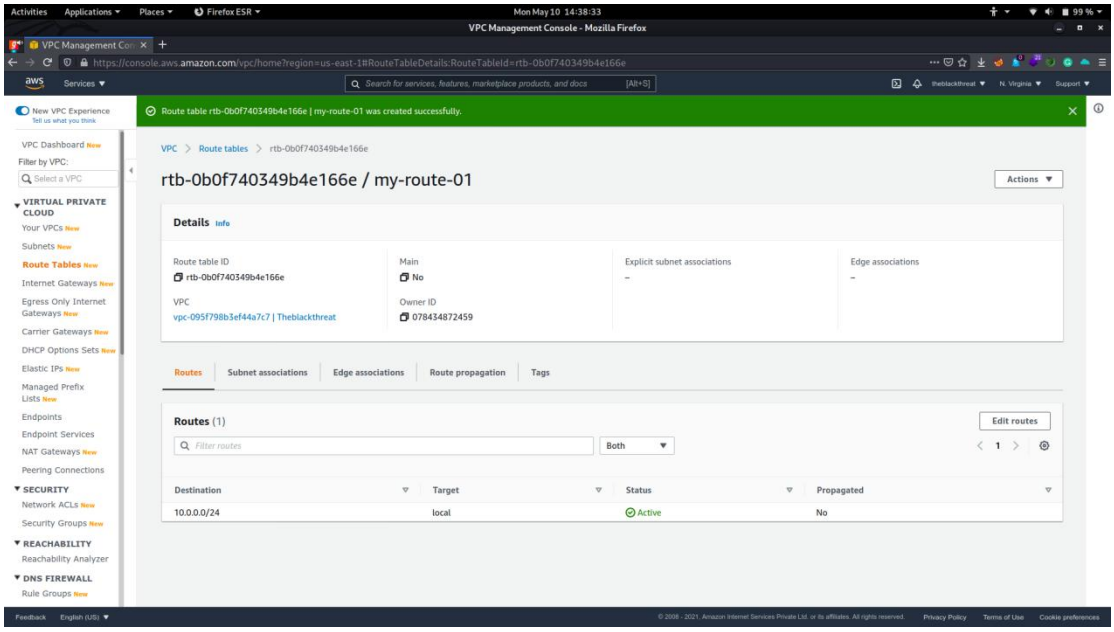
So i have set up with my subnet in VPC

Route Tables:
After this go through the Routing table to transmission of data and service over internet.

Step1: Go through the route table then click on you VPC
Step2: create a route table







Now we have successfully set up with route table

Now lets go with the **Internet gatways.**

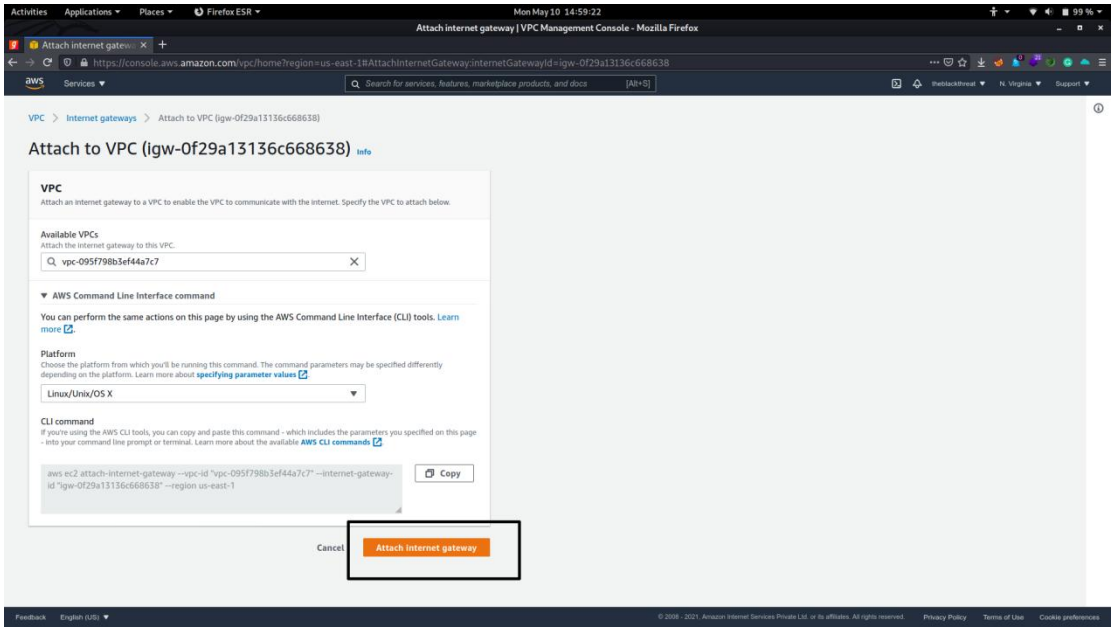
An internet gateway is a horizontally scaled, redundant, and highly available VPC component that enables communication between your VPC and the internet.

To use an internet gateway, attach it to your VPC and specify it as a target in your subnet route table for internet-routable IPv4 or IPv6 traffic. An internet gateway performs network address translation (NAT) for instances that have been assigned public IPv4 addresses.

Step 1: create a gatway. named it and

Step2: top right corner click on action and then click on **attach to VPC**

Step 3: select your VPC and then attach it to the gatway.(you can do it by CLI interface as well.)



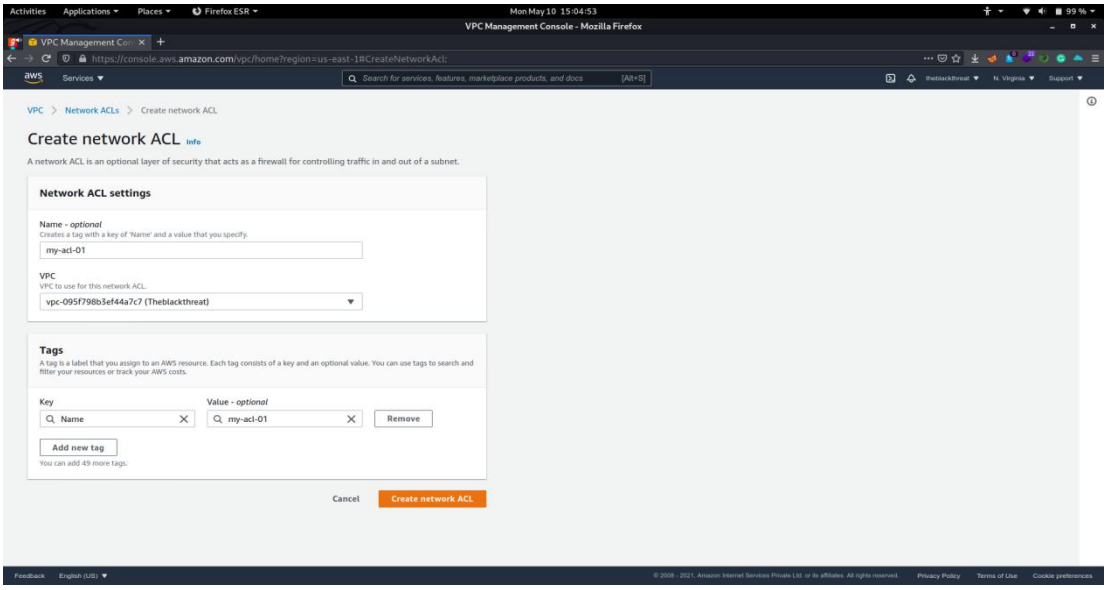
Now we have successfully connected the gateway to the VPC.

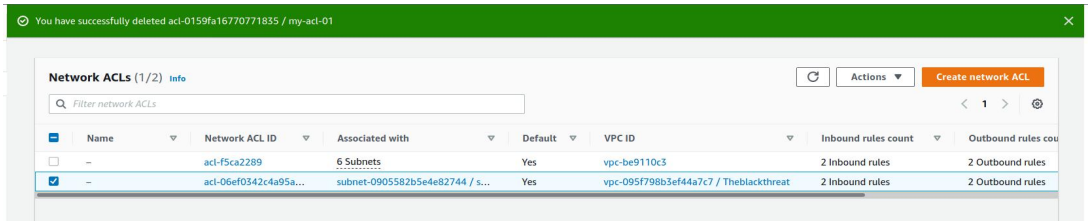
So now i'm not going through others these set is enough for my webapp which we will be creating so For more security of your VPC connection go through the

Security

we can Connect it to ACL(Access controll layer) and Security group.

Step 1: Go to security ---> Create network ACL





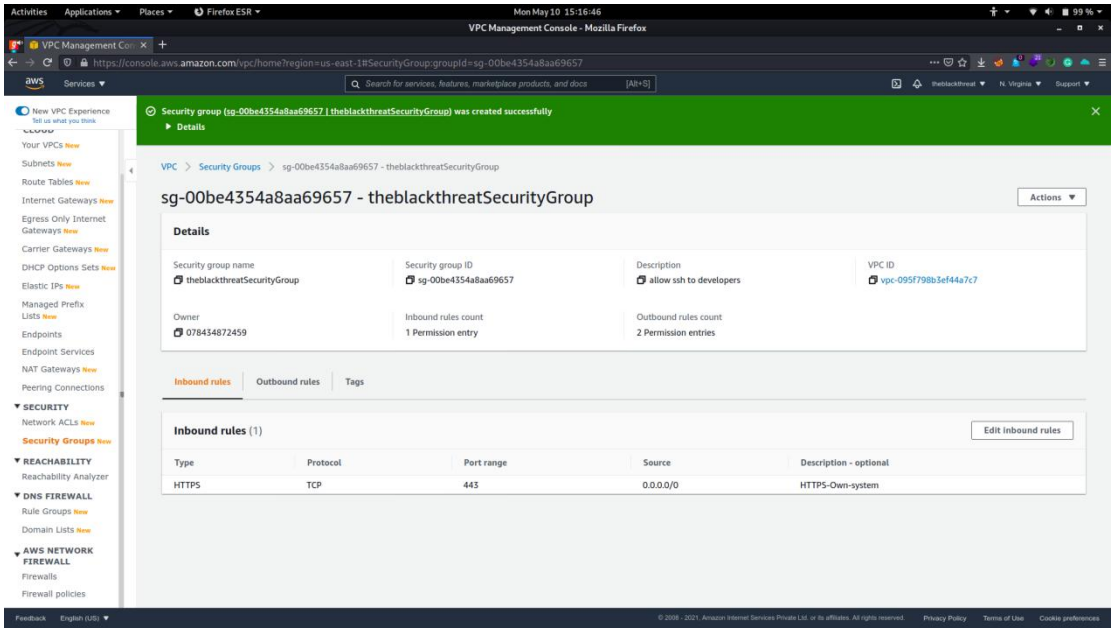
Configure with Security Group in VPC.

Step 1: go through it and then check your VPC then click on Create Security Group.

Step 2: add your security group name and the vpc that you want to secure

after this add inbound and outbound rule by all TCP

Step:3 click on Create Security Group.



We can set Network firewall to enhance more security.

Optional: Go to https://console.aws.amazon.com/vpc/ and create VPC wizard
enter vpc name “theblackthreat”
advantage of wizard it will automatically set up all Settings on VPC so saving much time

Windows webserver is used to host web application over internet AWS provide services to host own webpage private and publicly.
To Create a server we need to start a new instance into the VPC that we had created earlier.

Step 1: Go to EC2 ---> Select New Instances (Windows) ---> choose Instance Type(i.e Free tier)

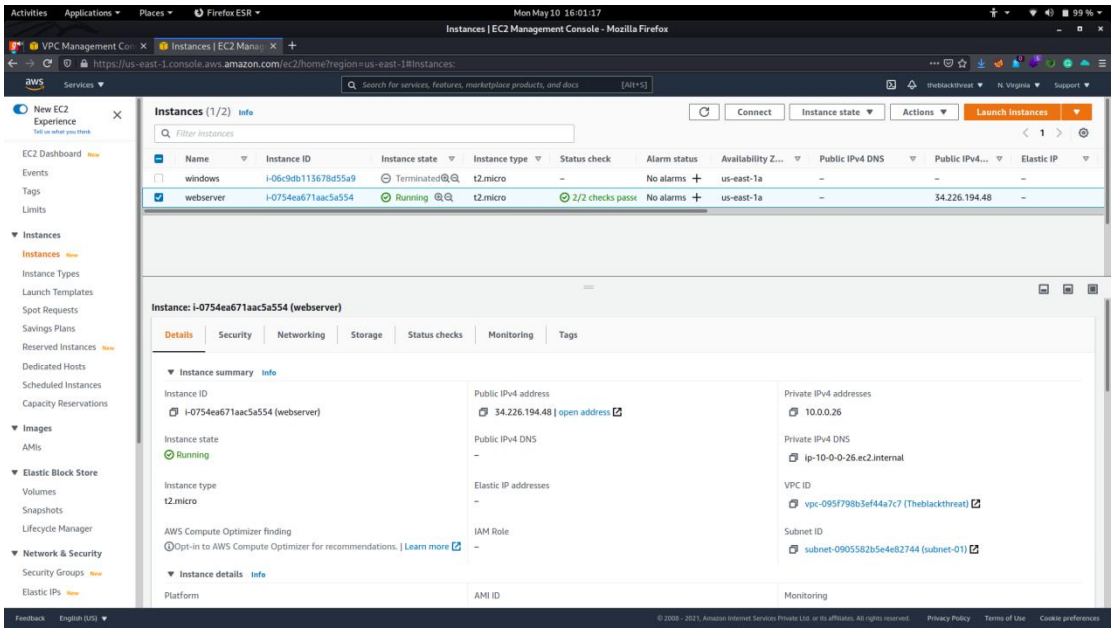
Step 2: Click next and Configure you Instance

(Note: I'm creating instance inot my own VPC that is thebalckthreat because we have already setup all service into VPC so I'm Selecting this to reduce time if you haven't do this go previous pages and Do it.)

Step 3: Add your storage mine is 30 GB and it is Enough for me.(but large organization change it accordingly.) ---> add security group that you create previously.

Step 4: Review and launch it.

Step 5: Create a key pair for connection to the RDP(Remote desktop Protocal) connection and Download it.



Step 6: now connect it and then get the password using key pair.

Step 7: Start you "filename.rdp" on Your System.

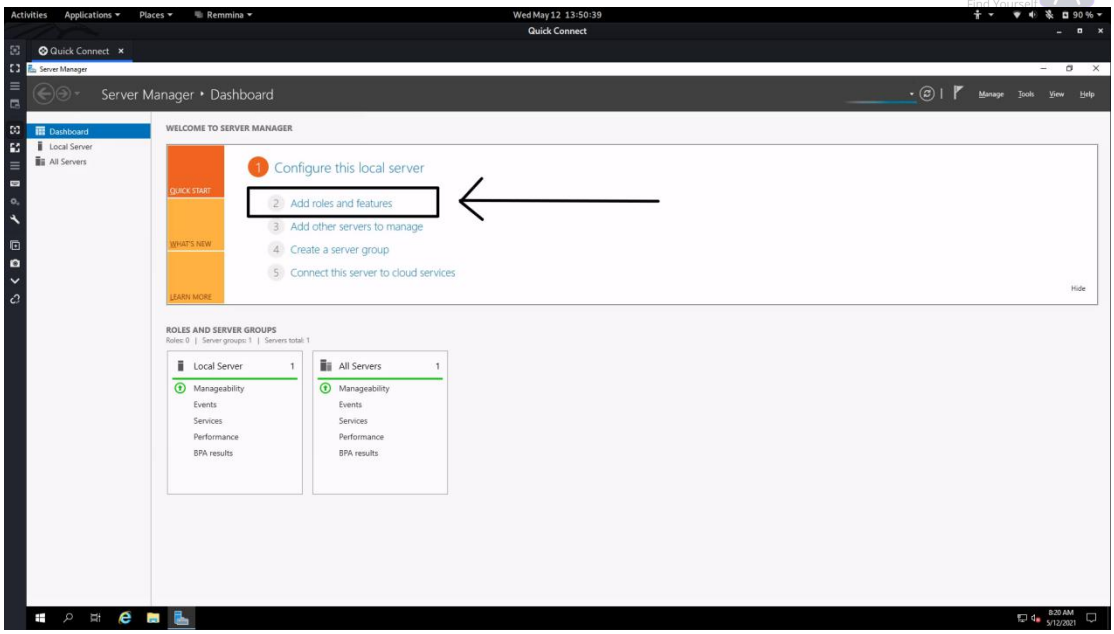
Troubleshoot to connection:

(Make sure you have enable the DNS hostname onto VPC otherwise it will not be connect.)
To do this go to your VPC select it and go to action and click on DNS hostname and enable it.

(Make sure you have connected elastic IP to do this just go to Elastic IP's on EC2 and Associate it with your instances.)

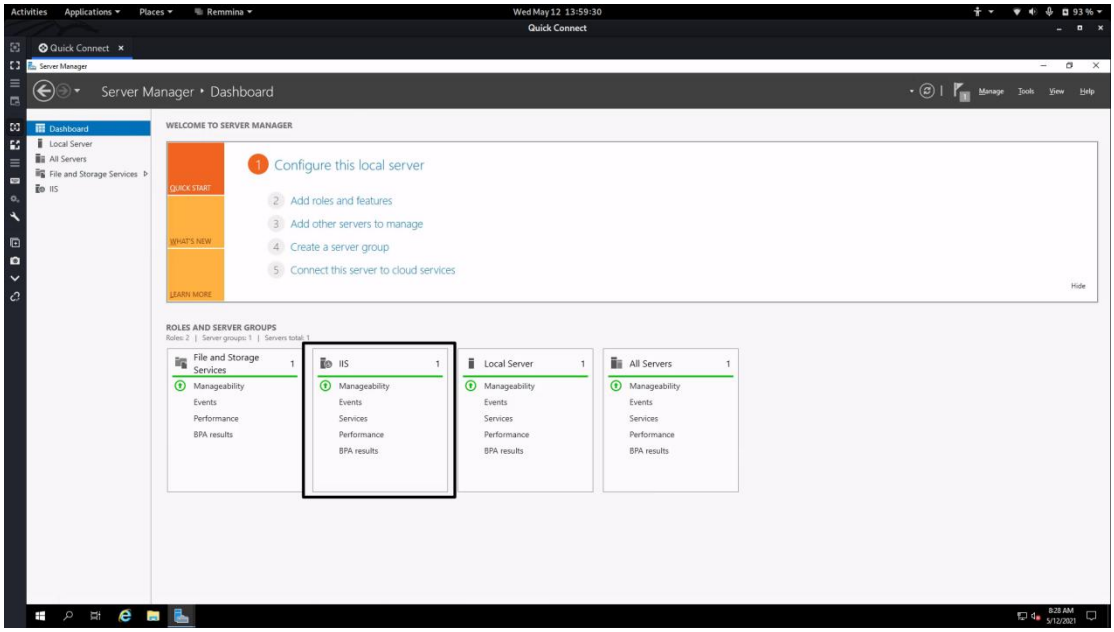
Step 8: After your instance is running then set for your webserver.

GO to Server Manager---> add role and features



Click on Next and set it all as per your requirement I'm Server role --->web server.---> click on Add features.---> Features ---> leave it as default---> next---> install.

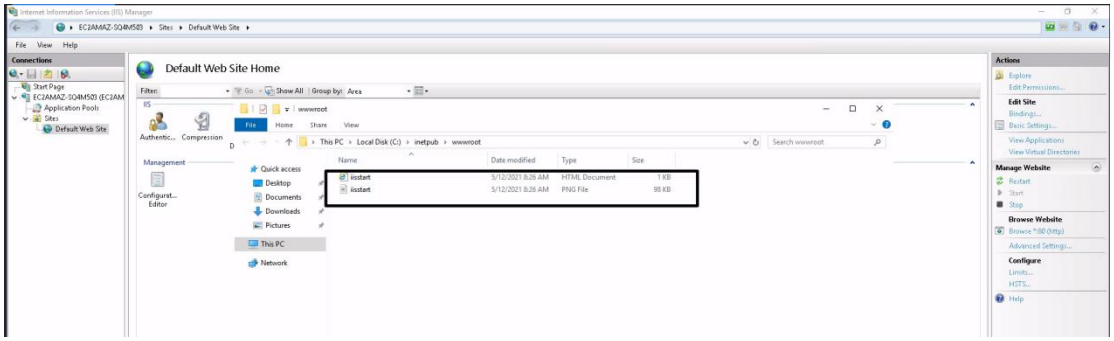
after completing the instalation process we it will look like this.



Step 9: Click on window button---> click on Windows administrative tool then find Internet information services(IIS) click on it

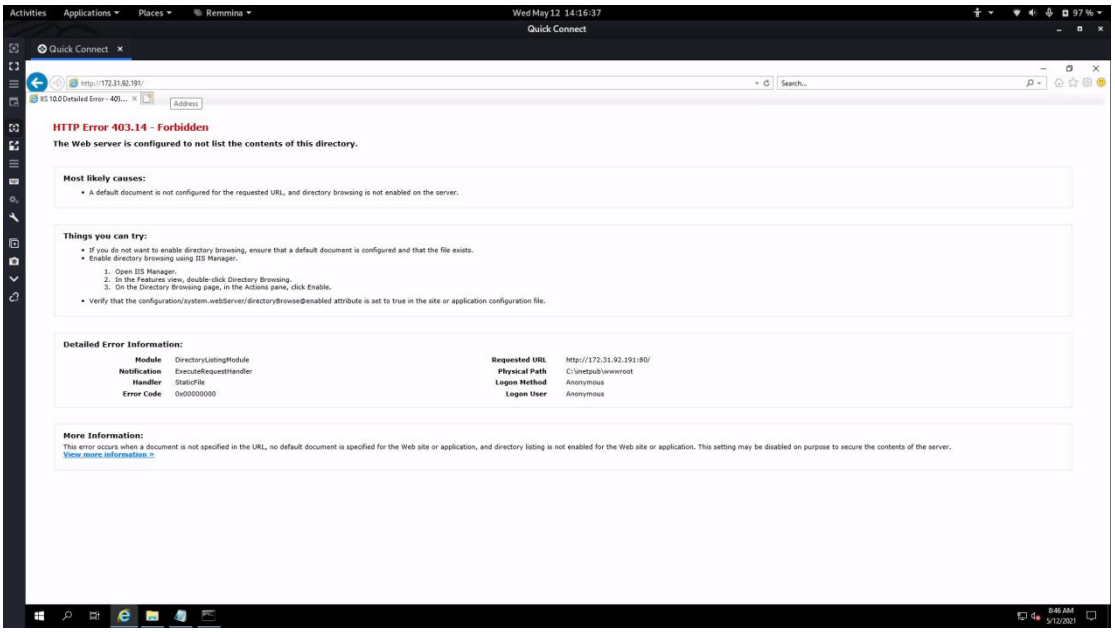
Step 10: click on Default webpage that is on left navigation bar you can find more services you can choose so .

Click on Explore and set up yuor ow web page here. ----->> one default file just remove it and create you own site here.



I have deleted the default file and created my owns

it will run on my machine.



It will showing that there is an error because we didn't given our webpage name. this will shows that our site seurity is running properly.

4. **Objective**: Introduction to AWS Identity and Access Management (IAM).

Requirments : Amazon web service account weather it is free or paid tier.

Working With AWS-Identity and Access Management.

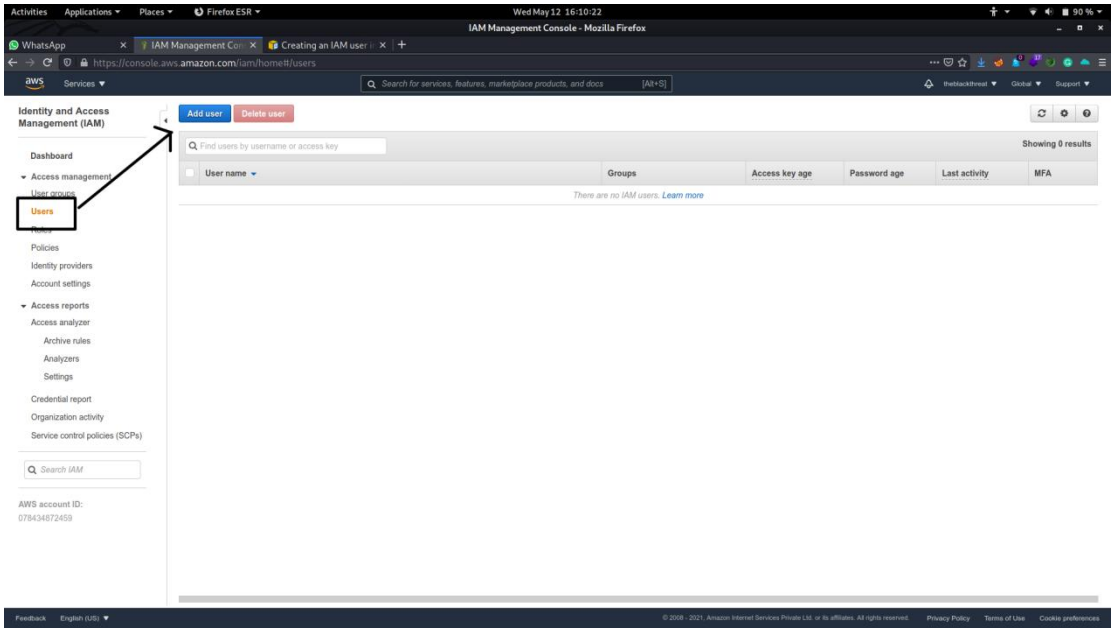
AWS -IAM user enable to manage access to the services and resoureces securily on AWS. IAM user can change the access management for the clients or the user and groups.

Root User have access to control the access for IAM user and he can create one or more IAM users into there account to access the services.

Creating A **IAM** User into **Root** account

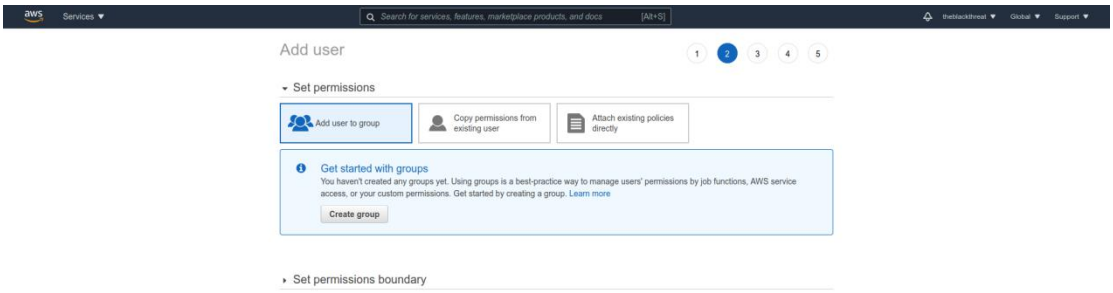
Step 1: Click on the user account and go on **My security Credentials.**

Step 2: On the left Navigation Pannel you can see the Users, groups, Role .... click on Users.



Step 3: Add user ---> add name of the user or [username]
We can give access to the user for CLI interfaces by enables the access keys ...

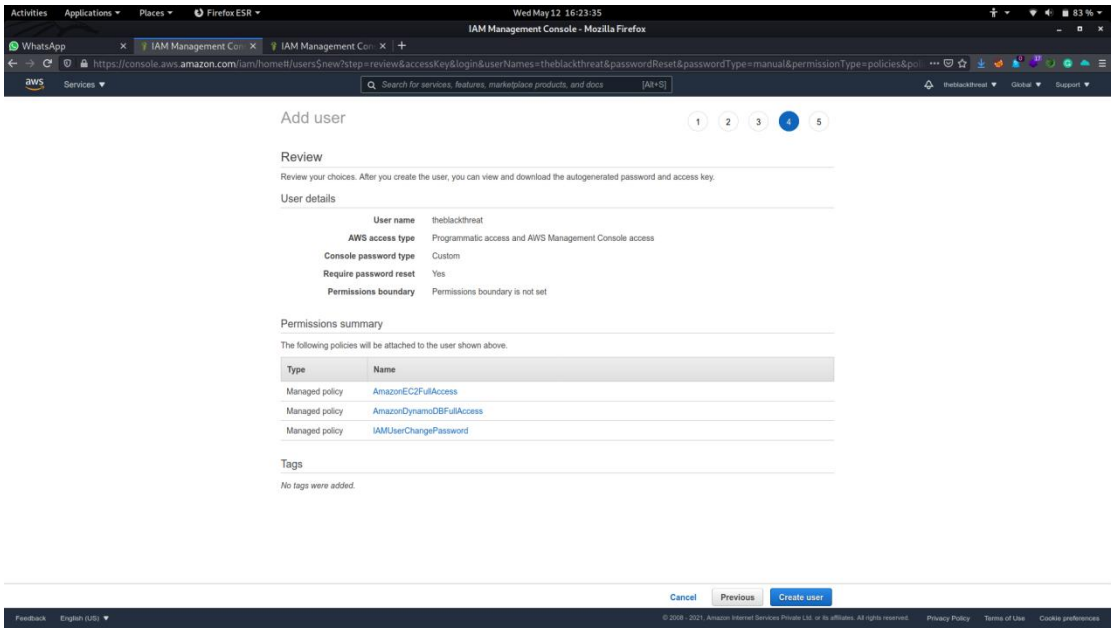
GIve you credentials then click on Next permissions.

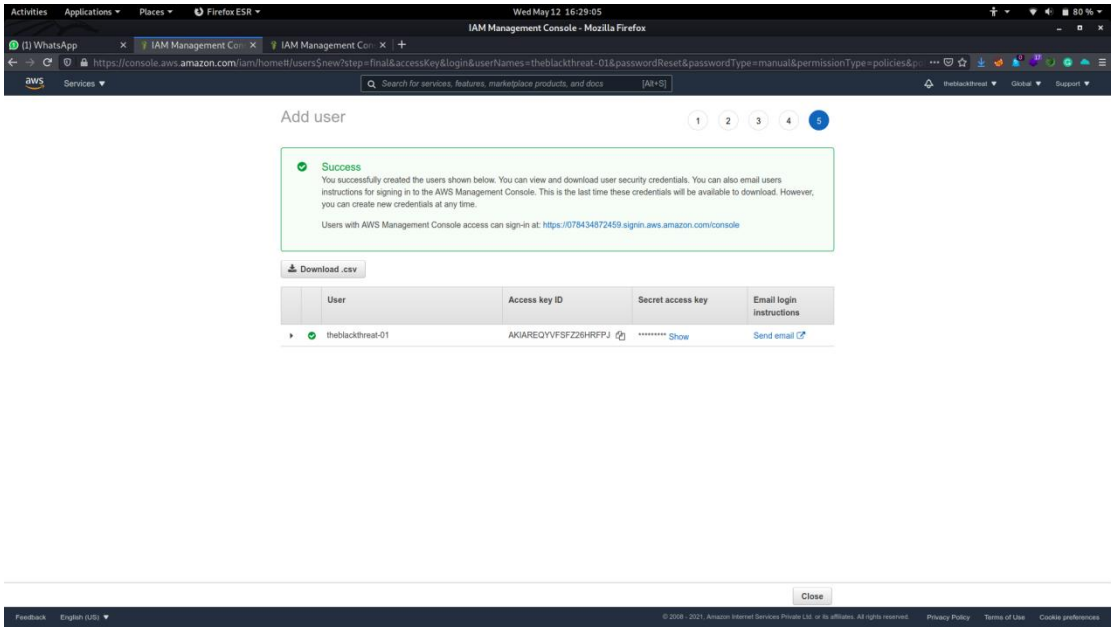


Step 4: you can add user to the groups if you have created it so far. In My case I'm not adding anything here. Now most impartant thing is Policies we can give access to the user by creating custom policies.

Click on Create policy or use can use default as well.
I'm giving "**AmazonEC2FullAccess" "IAMFullAccess" and "AmazonDynamoDBFullAccess**" **ComputeOptimizerReadOnlyAccess.**

Step 5: click no next--->Create User





Now we have succesfully created a IAM user. to check logout from the session and againg login as a IAM user.

https://console.aws.amazon.com/console/home

Login as IAM.

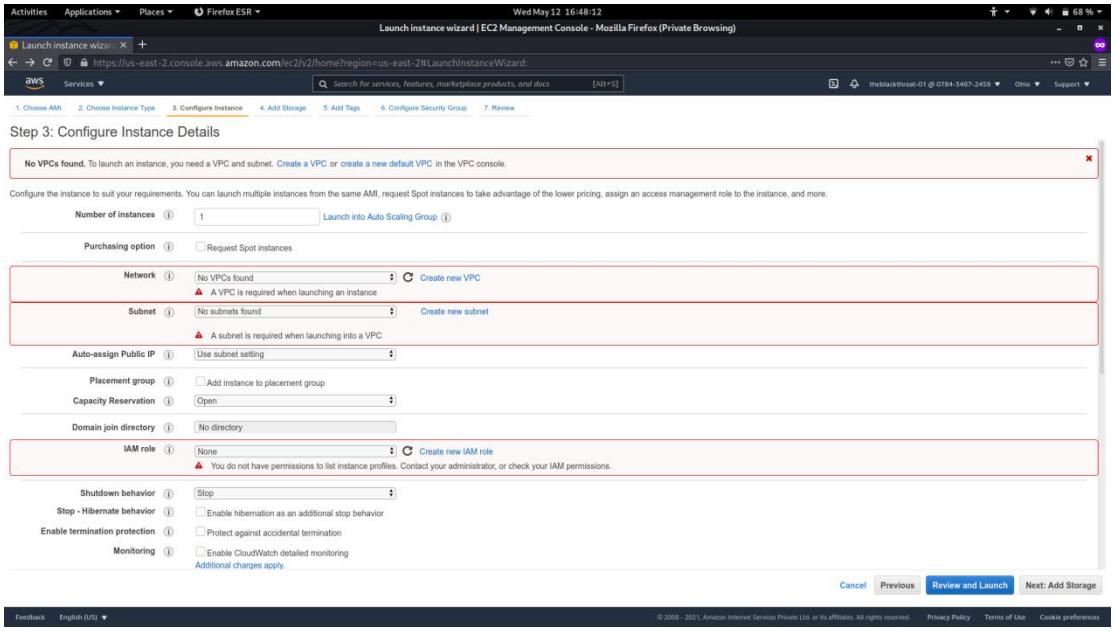
We have privillages to create Comute machines, VPC's and Create databases.

Lets create Instance using IAM user.

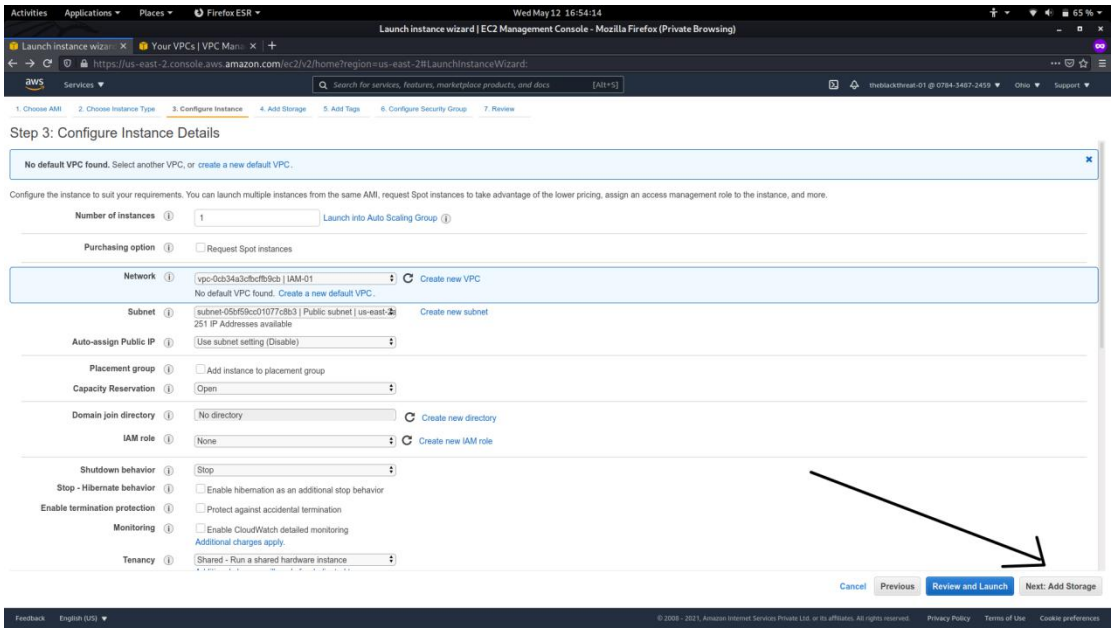
Step 1: GO to EC2 ---> select your instance machine and then choose your type of Instance.
Im using my free tier Instances.

Step 2: Launch instances.

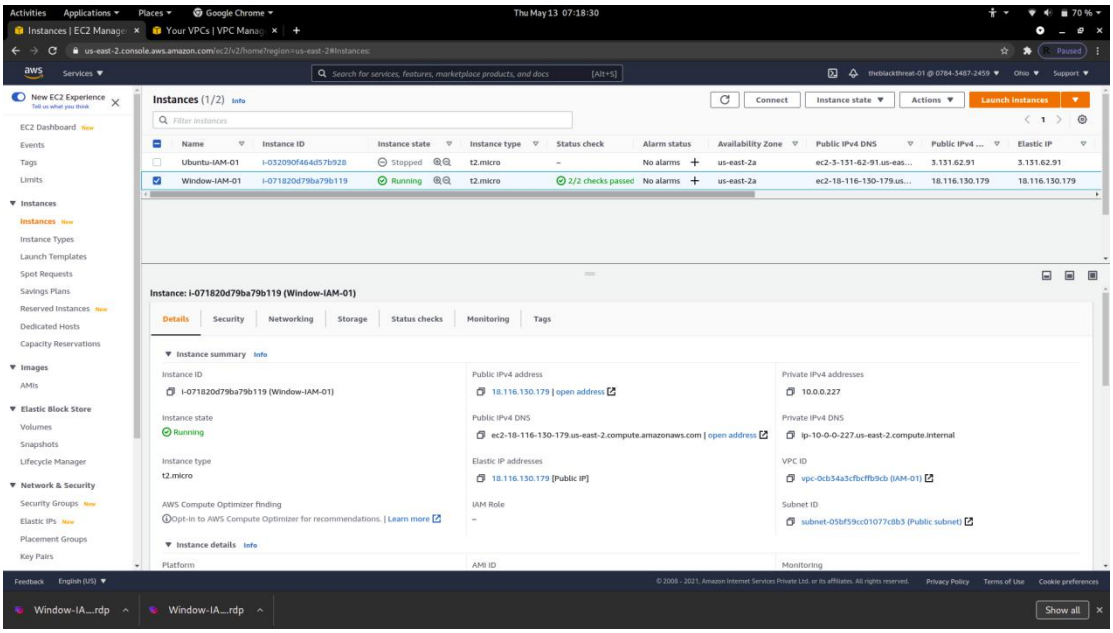
(I'm Creating a Windows Machine named **windows-IAM-01**)
Step 3: Repeat the above process to create instance
Step 4: Create Default VPC as we haven't created earlier.



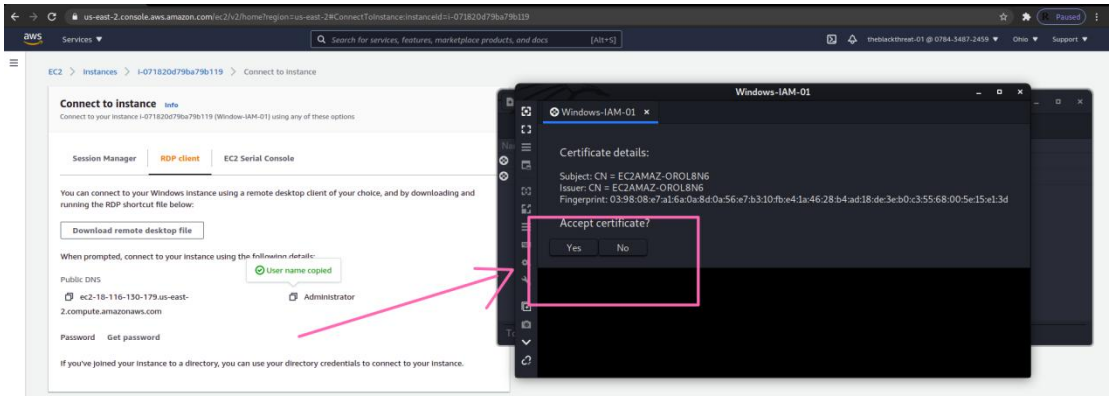
create VPC and then click to further.



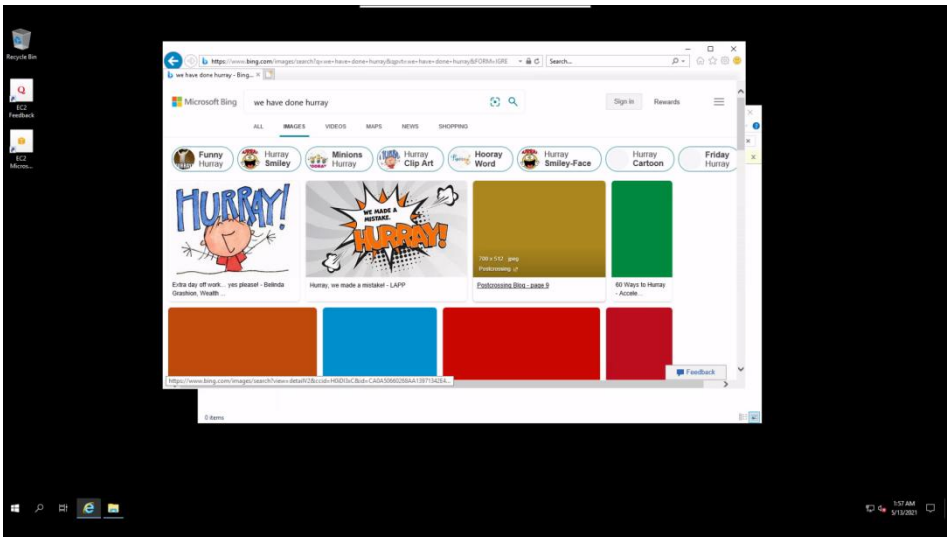
Go to reapeat process after instance will be created then start it onto your machine.



Connecting this machine via RDp client.
Step 5: Download RDP client file and dycrypt your password. and connect it to your host.



It will connect to the Server and now we have created the Instance using IAM Accesses.
We can host a webserver as well by doing the same thing as above mentaioned.



5. **Objective:** Deploy a Web Application on AWS

Requirments : Amazon web service account weather it is free or paid tier.

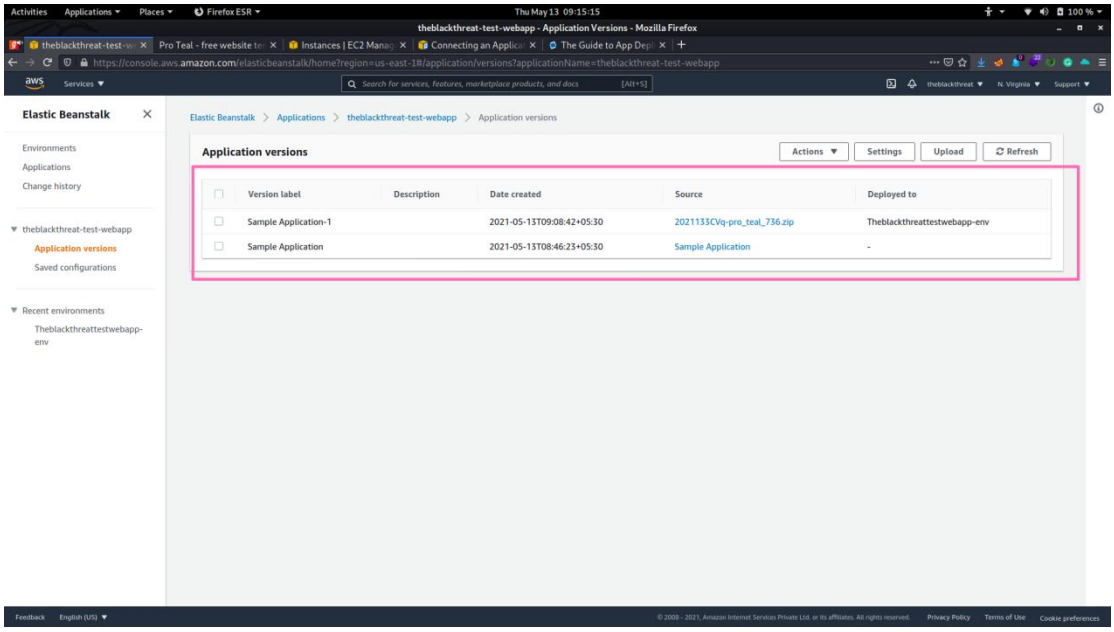
Ceating a Web app using **Elastic beanstalk**

Elastic beanstalk service provided by AWS it has very Internactive Interface and we are able to create a Static or dynamic web app by just creating the collection of code or just upload your application in it.

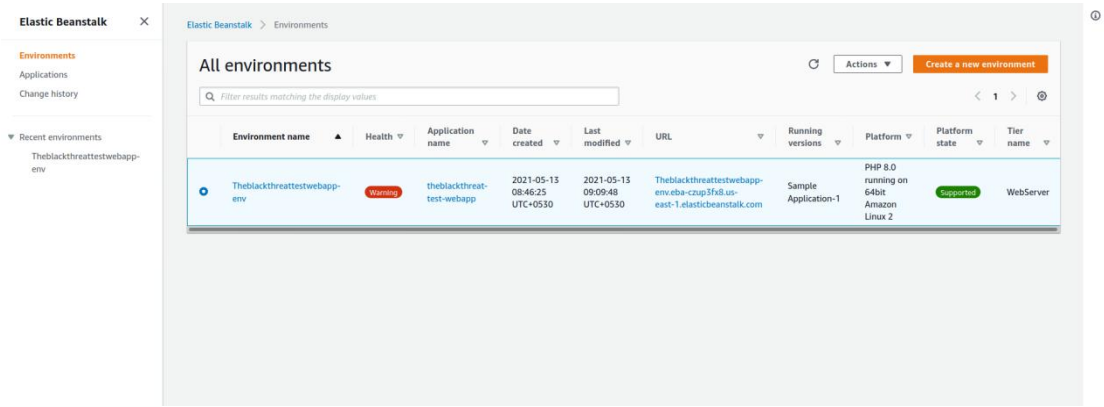
we are creating a Static then we go for dynamic webapp.

Step 1: Go to compute ---->Elastic beanstalk----> create an Application.---> name you webapp *"Theblackthreattestwebapp-env"*
Step 2: configure your settings then click on upload your application.
(It will take some time to setting up)

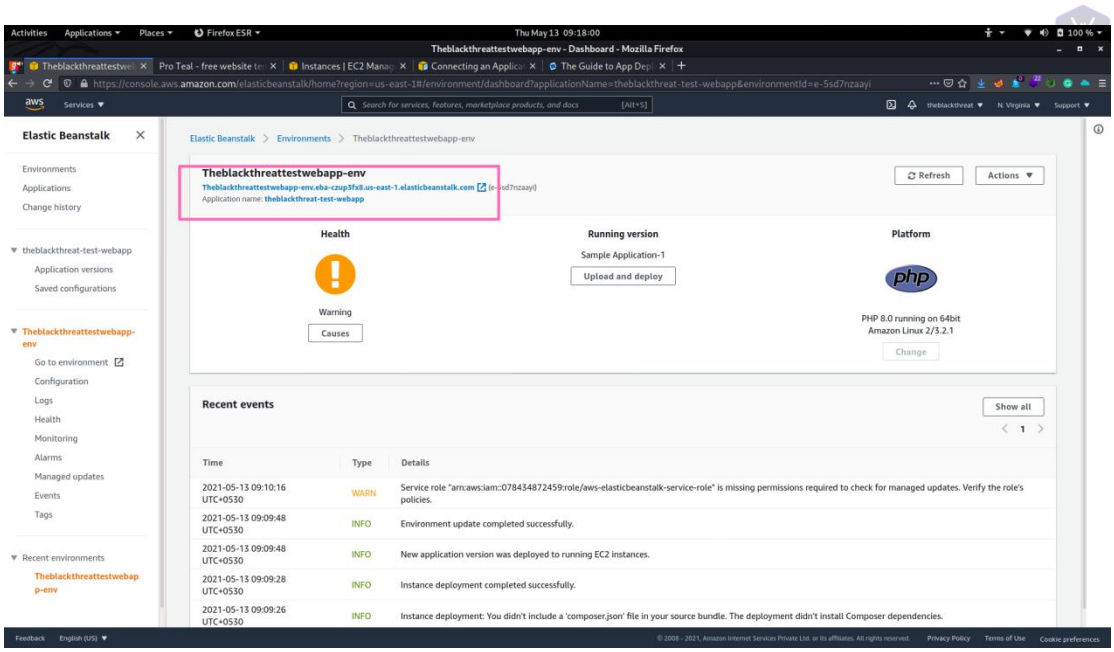
I have used a ***sample application-1*** we can run it as publically.



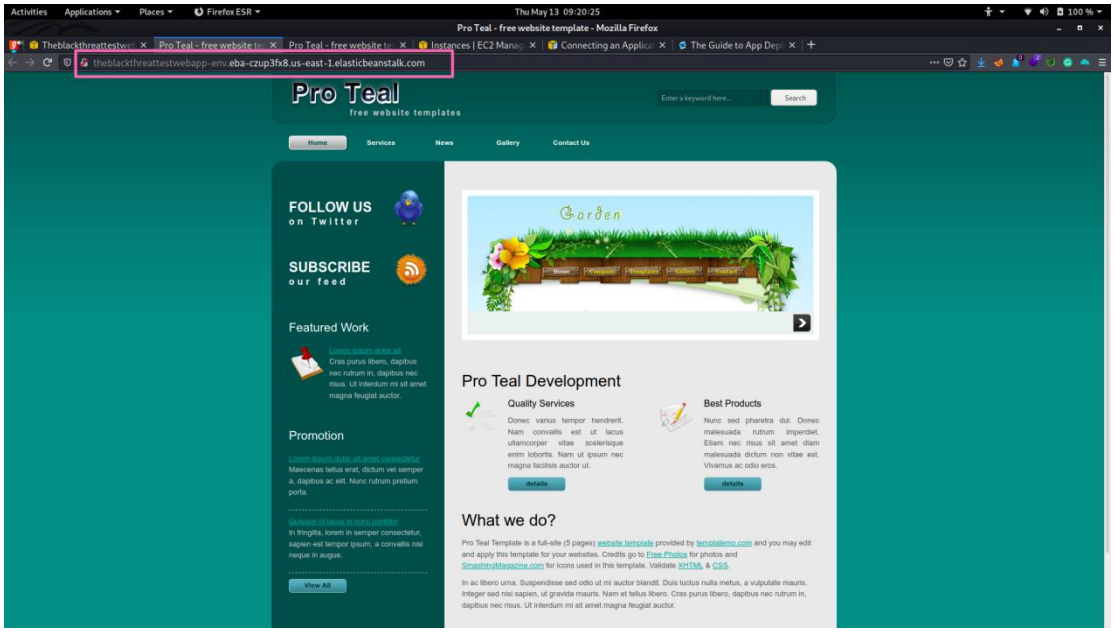
Step 3: Now click on applicaiton then select you web app.



Step 4: click on the url that shown on above to view your upladed app.



Don’t bother about warning it will just about health of our application server later on we will solve it.
Click on the url



Here is my application that run on publically.
So we have done the **Static Part**.

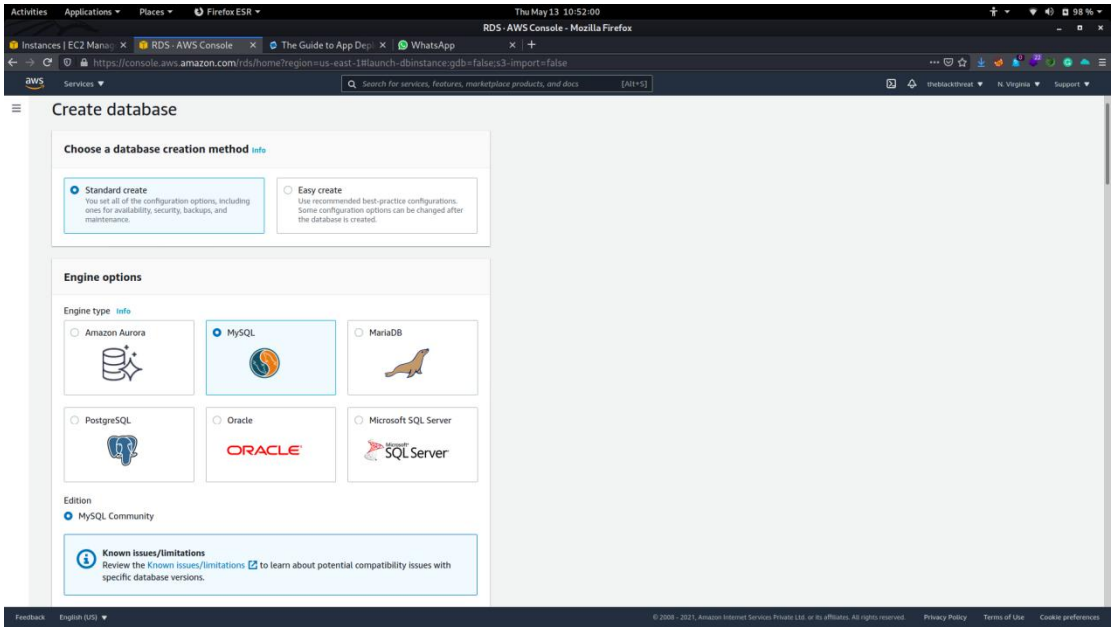
Creating a Dynamic Web application.

To create Dynamic web page we need to create a database on AWS and then it must connected to the Web app which is running on the Instances.

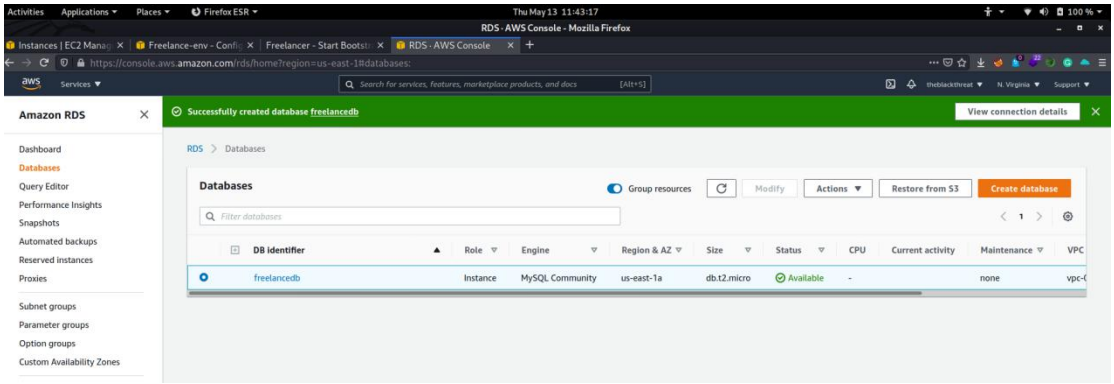
Step to produce : 1. create an Instance
2.Create RDS and Connect to the Webapp by configrating the Databse.
3.create elastic beanstalk.

Creating RDS database service.

Step 1: Got to Database----> RDS --->select Mysql service



Step 2: Selet MySql service---> create a RDS named “FreelanceDB”--->then set it for following
Edition
Version
Template(free tier)
settings --> databse name
leave all things as default and create database.

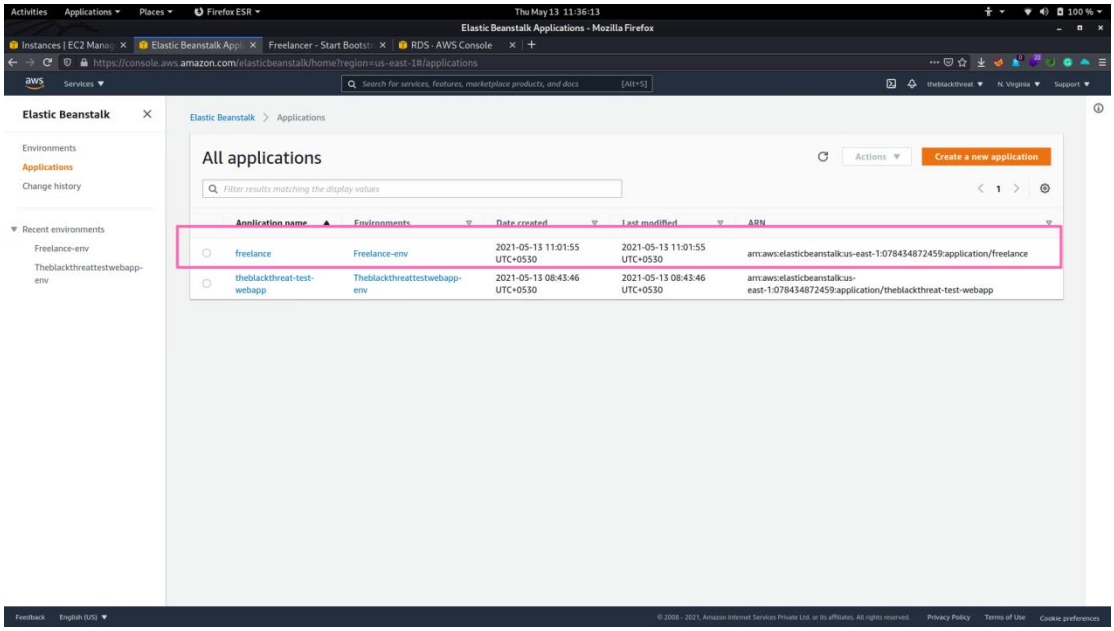


Creating Application on Elastic Beanstalks

Step 1: First Create an Instance or just go through the Elastic Beanstalk. AWS autometic create an instance for you.
(Sample PHP code for webapp:
https://github.com/theblackthreat/AWS-console/blob/main/webapp%20samples.zip)

Step 2: create an applcation and upload it on Elastic beanstalks. and then select for PHP

I have created a sample php page “Named **freelance**”

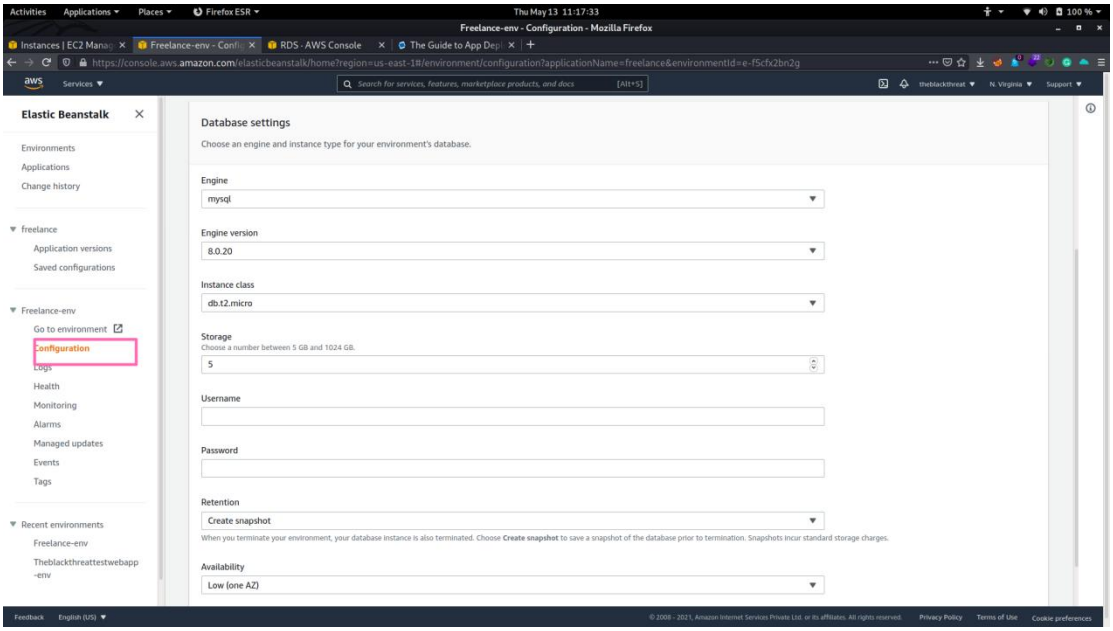


Step 3: click on the application---->click on Enviroment name ---> left panel find Configuration----->

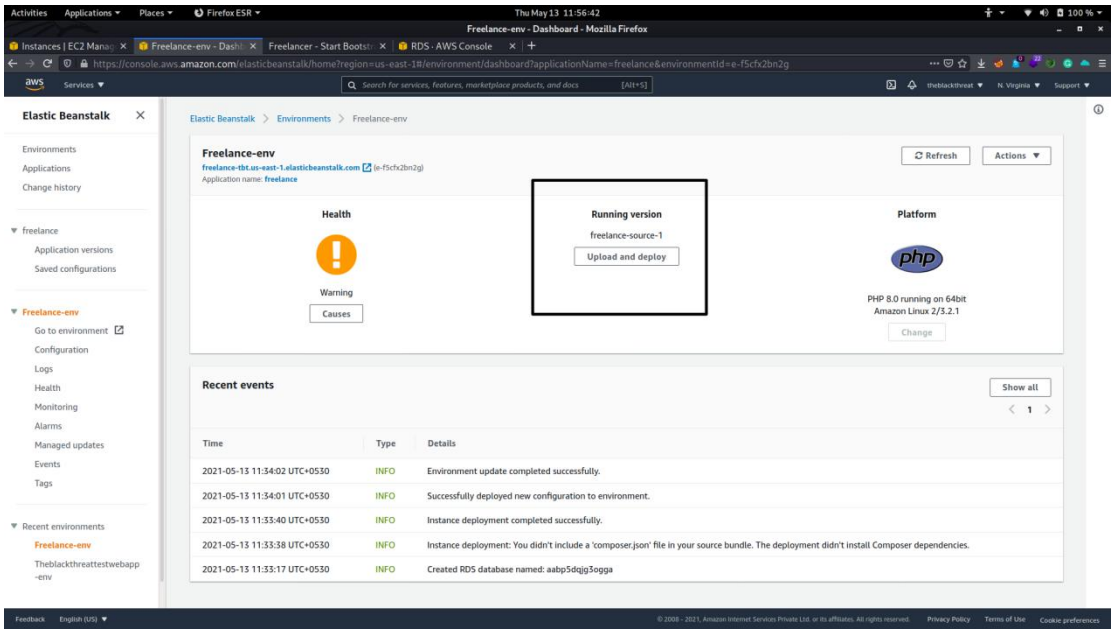
Click on Configuration To create another security group in the environment, you should do the following:

Open the Elastic Beanstalk console and find the management page of your environment. --->Select “Configuration” and choose “Modify” for “Instances”--->Select the required security group that you need to attach to the instances and apply--->After reading the warning, press “Confirm”

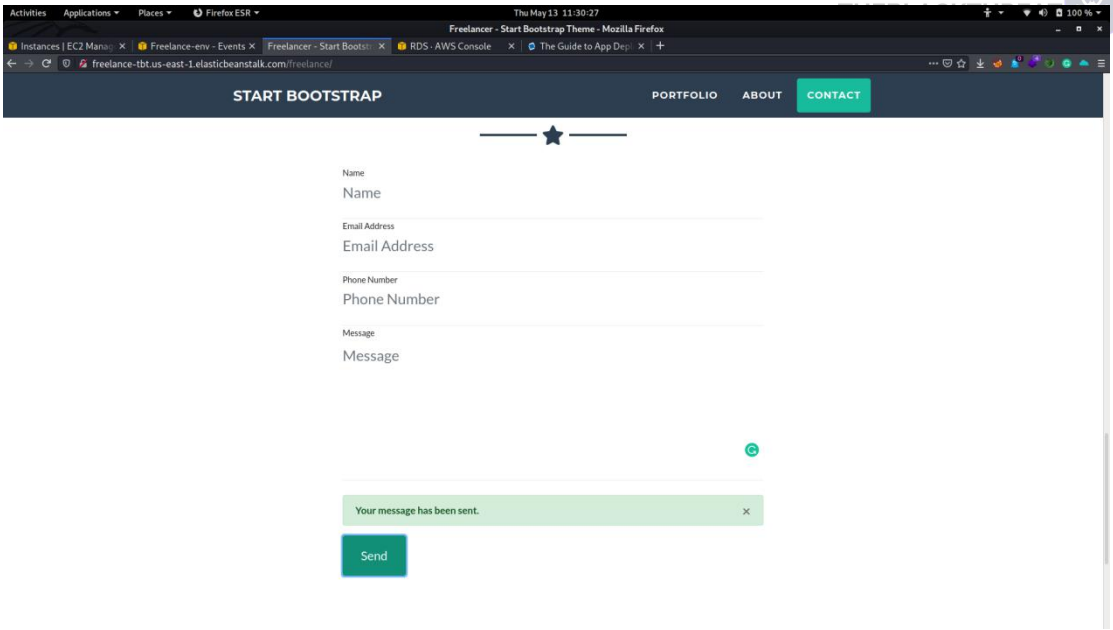
Syep 4: To connect to the database scroll down and click on Database---Edit
It will fetch you RDS databse and then we cna set up the credentials. put your username and Password form RDS database which you have created before. ----> Apply.



Step 5: After configuration of Database and Software modification save it and go to the application select your wep applicatio and then upload you sample.zip file here. and click on deply,



Wait for some time and after a while your application is able to run and is automatiallay connected to the database. For test Just check its working or not. go to the contact page and fill rhe form it will connect to the server.


Activities
Applications
Places
Firefox ESR
Thu May 13 11:30:27
Freelancer - Start Bootstrap Theme - Mozilla Firefox
Instances | EC2 Manag
Freelance-env - Events
Freelancer - Start Bootst
RDS - AWS Console
The Guide to App Depl
freelance-tbt.us-east-1.elasticbeanstalk.com/freelance/
START BOOTSTRAP
PORTFOLIO
ABOUT
CONTACT
Name
Name
Email Address
Email Address
Phone Number
Phone Number
Message
Message
Your message has been sent.
Send

----------000----------

6. Objective: Using Auto Scaling with AWS Lambda

Requirments : Amazon web service account weather it is free or paid tier.

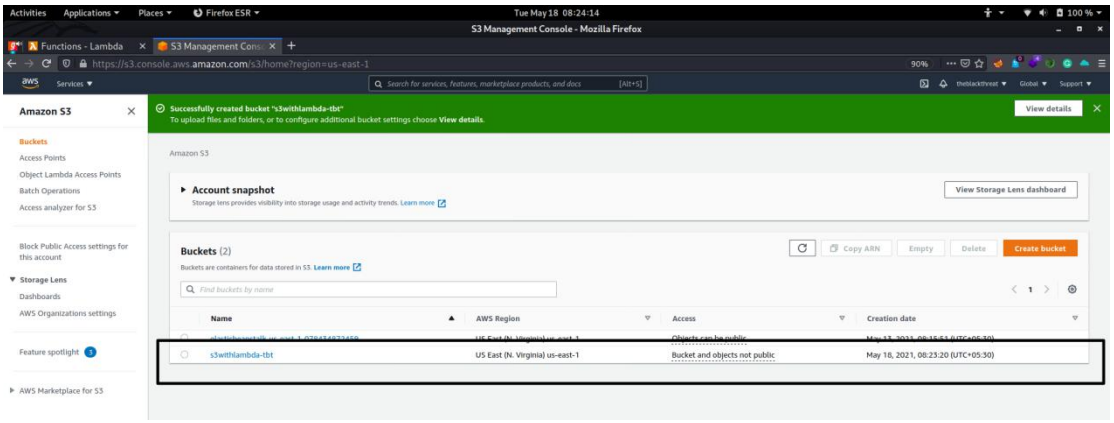
Lambda : AWS provide a serverless service named Lambda that runs our code and manges the triggred services for ususing the resources.it is manage the services parallaly and manage scalling. we can create lambda functions to manage concurrent traffic. we can use our own logics to handle the services by our own methods.

Creating a lambda function:

Our Aim : first wee need to create a s3 bucket.
then we create lambda function
provide lmabda function to work with s3 bucket so we will give permission from IAM role.

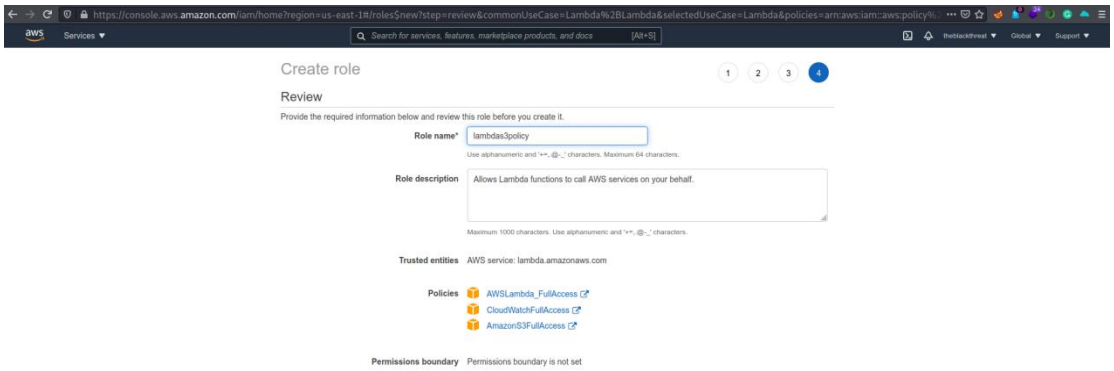
trigger point : When we will upload a file into bucket it will trigger the function that will show us that our file is uploaded successfully on cloudwatch.

Step 1: create a s3 bucket named "**s3withlambda-tbt**" leave all option as defaults ----> create bucket

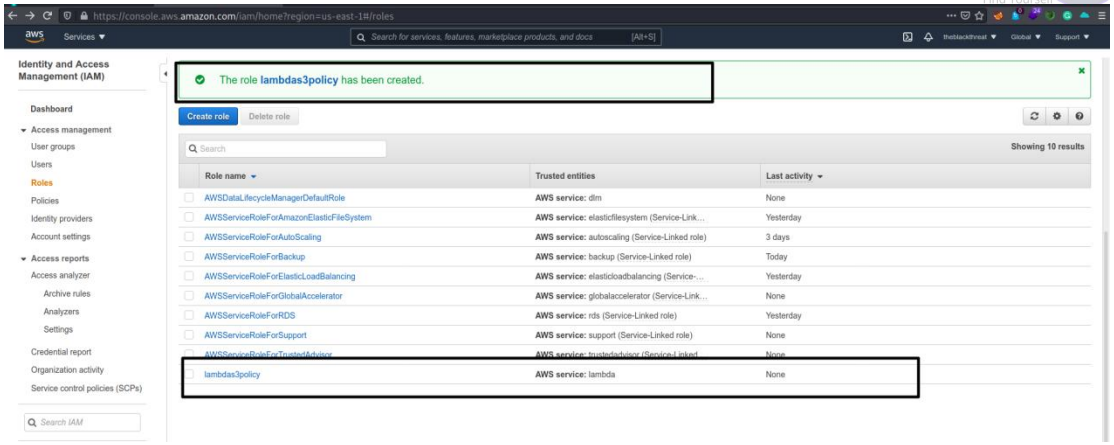


Step 2: Now go to IAM to give permission to the s3 and lambda.
https://console.aws.amazon.com/iam/home?region=us-east-1

Role---> Create role--->select lambda---> click on next--->attach policies as given
Amazons3FullAccess
AWSLambdaFull_Access
CloudWatchFullAccess
click on Next.



Step 3: Cleck on Create role.



We have create IAM role and S3 now lets move onto the lambda.
Step 4: find lambda on Services on AWS --> Create Function--->name your lambda function as "**workinglambdas3-tbt**".

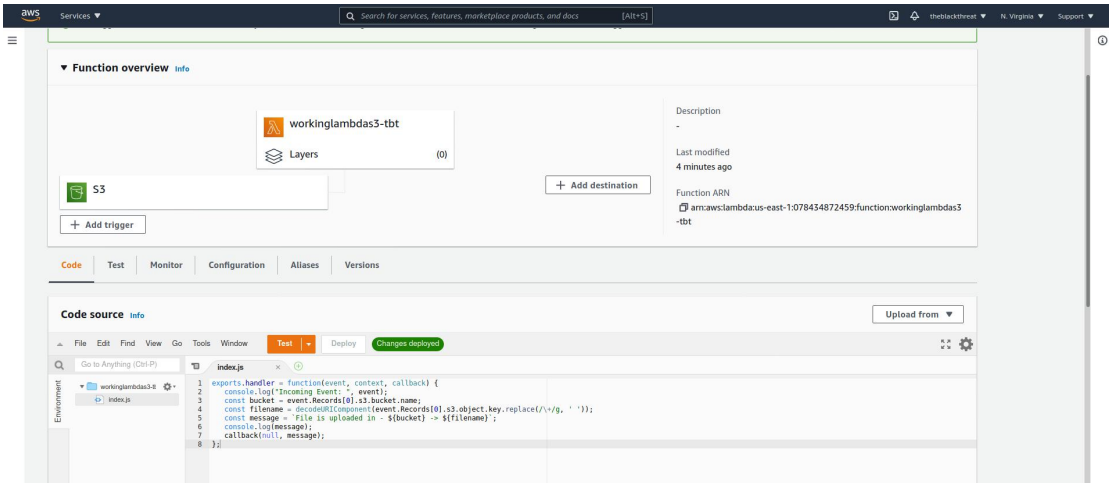
Code ---> Node js latest version.(14.x)

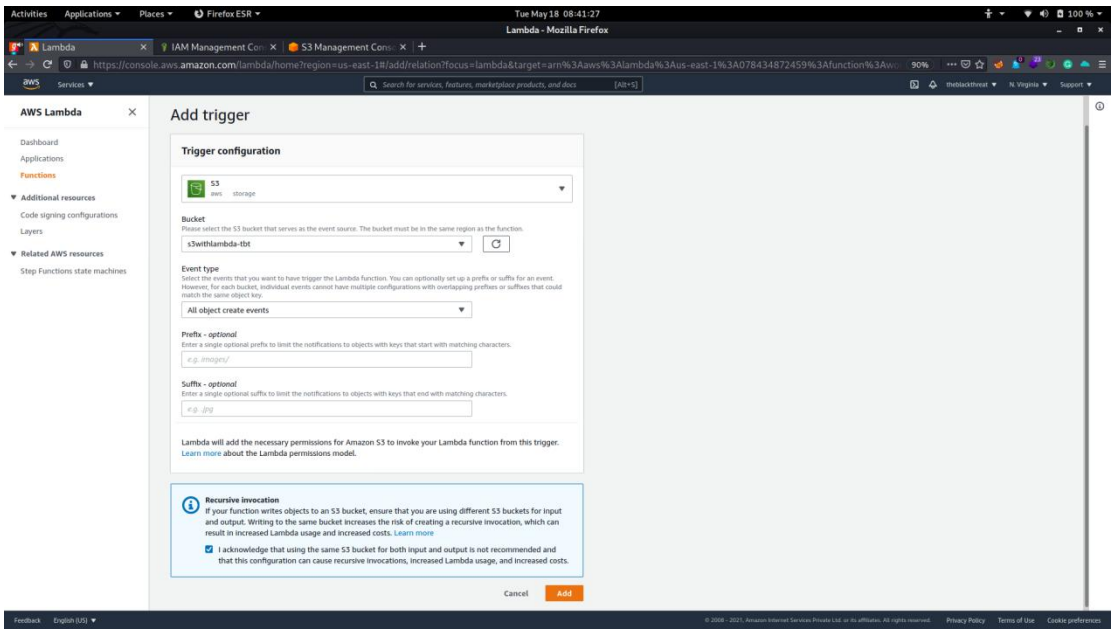
Choose permission ---> go to execution role(use an existing role)--->> select role that you have created on IAM "**lambdas3policy**". then next.
add this code and deploy it.

```
exports.handler = function(event, context, callback) {
    console.log("Incoming Event: ", event);
    const bucket = event.Records[0].s3.bucket.name;
    const filename = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
    const message = `File is uploaded in - ${bucket} -> ${filename}`;
    console.log(message);
    callback(null, message);
};
```

Add trigger as show in Above function ---> S3-->

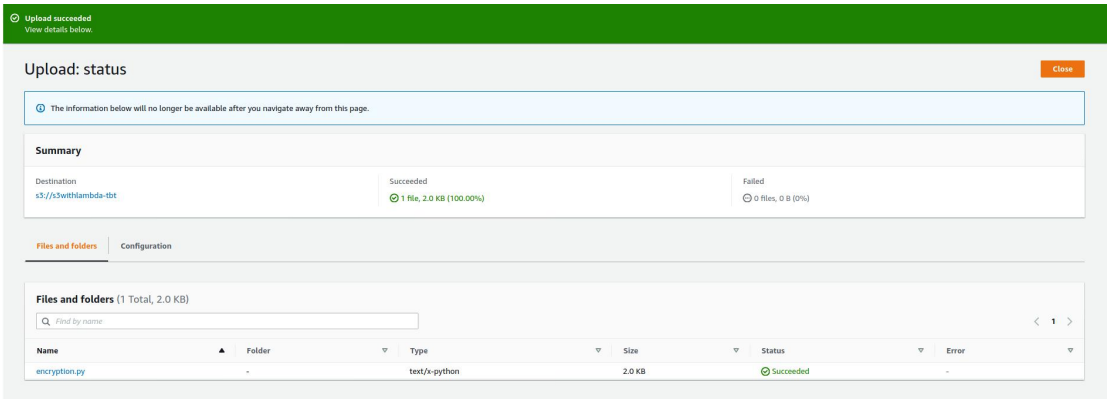
choose event type **all object created events ---> add**



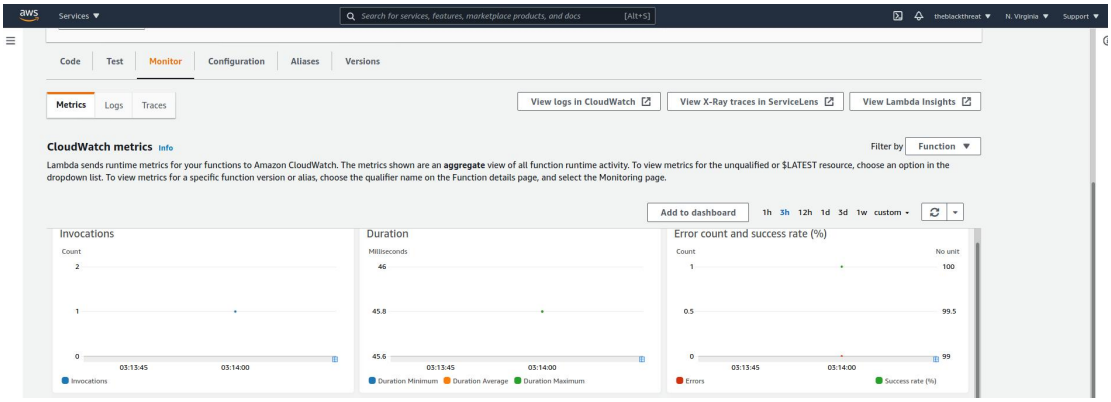


Now We have done all things lets check wheather its working or not.

Step 5: Go to s3 and upload any objects in your "**s3withlambda-tbt**"



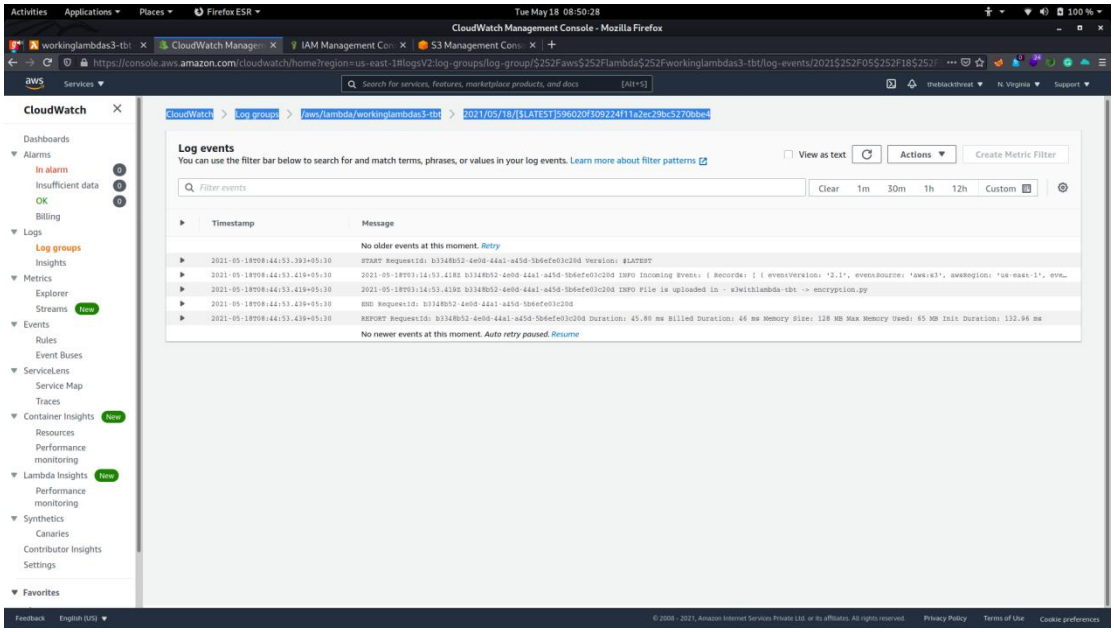
Step 6: TO check our lambda function will trigger or not go throug the lambda monitor section.



Yes its working our function will trigger as shown in dots. now click on **view logs on cloudwatch.**

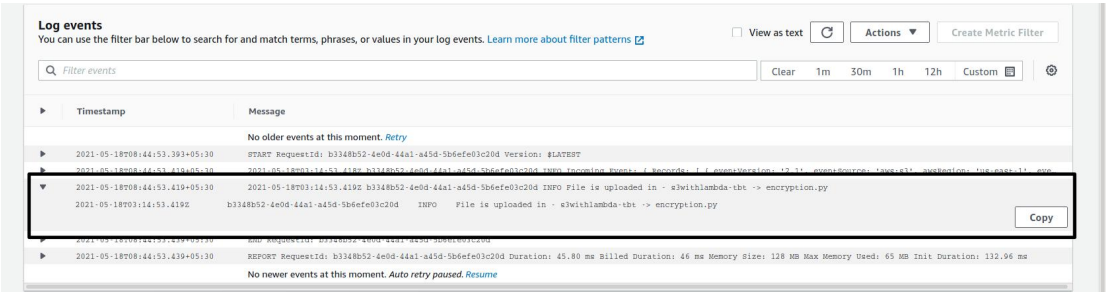
IN the cloud watch we can see all things that we have done so far is logged here .

· CloudWatch >> Log groups >> /aws/lambda/workinglambdas3-tbt >> 2021/05/18/[$LATEST]596020f309224f11a2ec29bc5270bbe4



Now find your file that name that you have uplaoded. It will look like below

2021-05-18T03:14:53.419Z b3348b52-4e0d-44a1-a45d-5b6efe03c20d INFO File is uploaded in - s3withlambda-tbt -> encryption.py



Now we have completely done with lambda function
lambda fuction will allow us a serverless execution that perform verious task in the backgroud.

Example : a clicnt logged into the webapp the data will trigger the lambda fuction and lambda can do what a developer wants. he can checks the google plugin that email is valid or not or password is right or not such this without direct sql interection of with the database. it will work as intermediate fuction between the client and database.

7.Objective: Launching EC2 Spot Instances with Auto Scaling and Amazon CloudWatch

EC2 auto scalling group that maintain, scale-up, or scale-down the number of instances needed based on scaling policies that are defined based on the demands of that particular application.

EC2 allows automatically add or remove the instances as needed. Dynamic **scaling** responds to changing demand and predictive **scaling automatically** schedules the right number of **EC2** instances based on predicted demand.

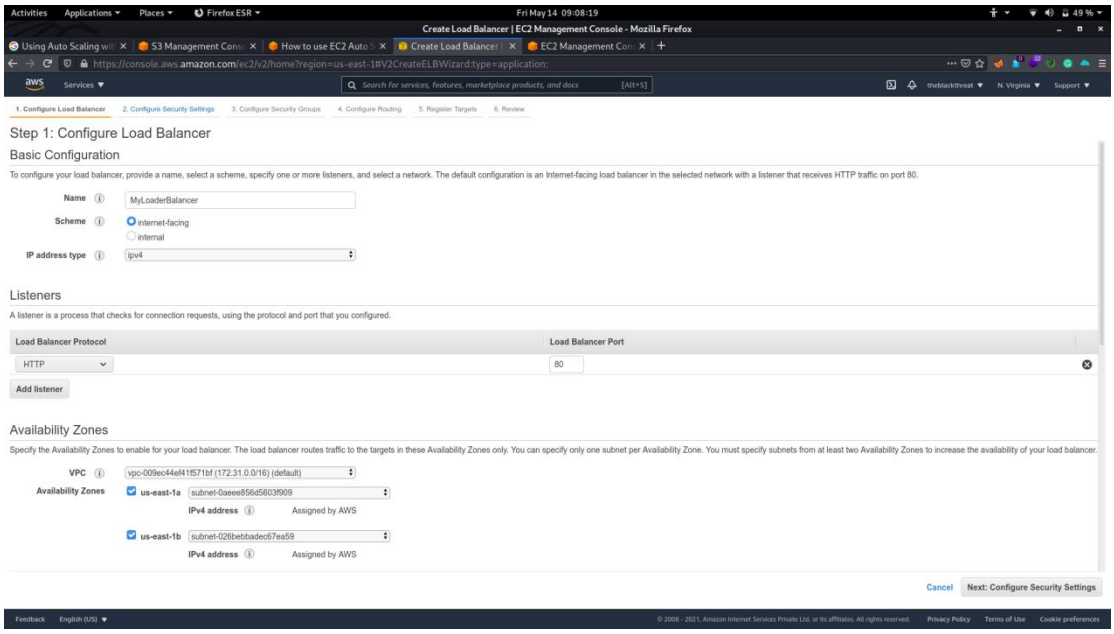
EC2 Spot Instance : spot instance allows us to request for unused EC2 instances in the lower cost.

To create the EC2 Autoscalling we need to create load balancers first.
Creating a Load balancer:

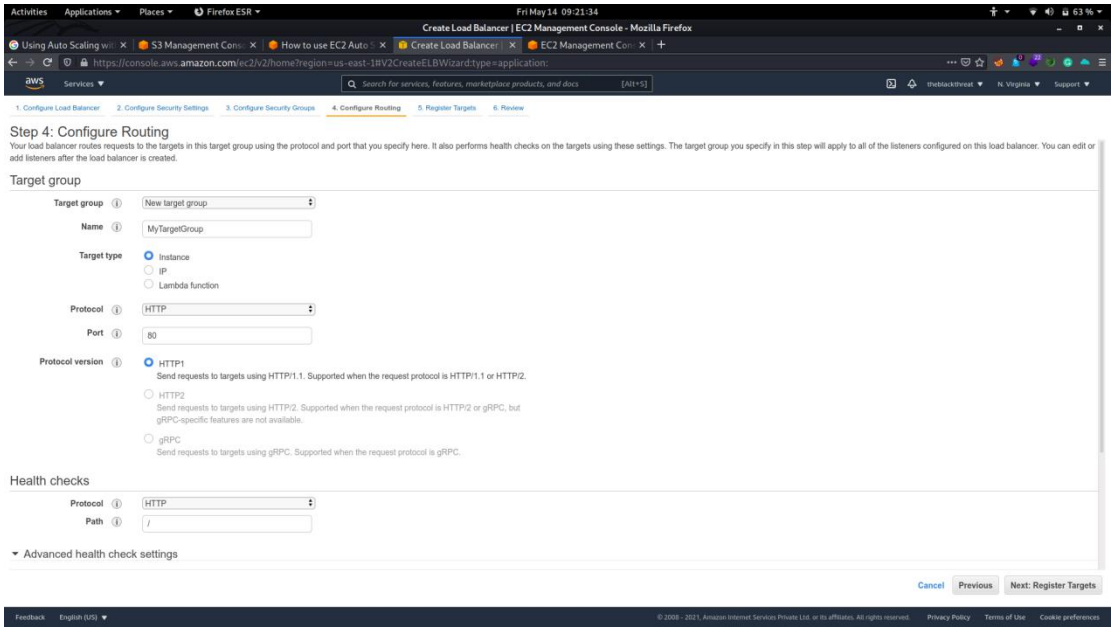
Step 1: EC2 Spot Instances
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers:sort=loadBalancerName

the load balancer will create requests to EC2 spot Instances.

Create Load Balancer ---> select HTTPS/HTTPs

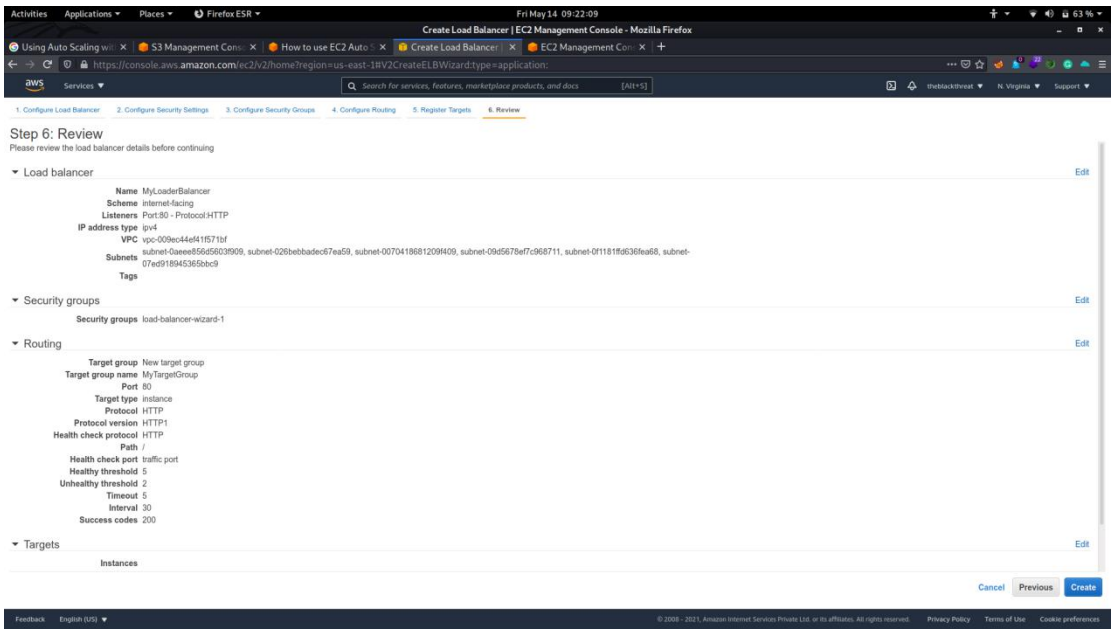


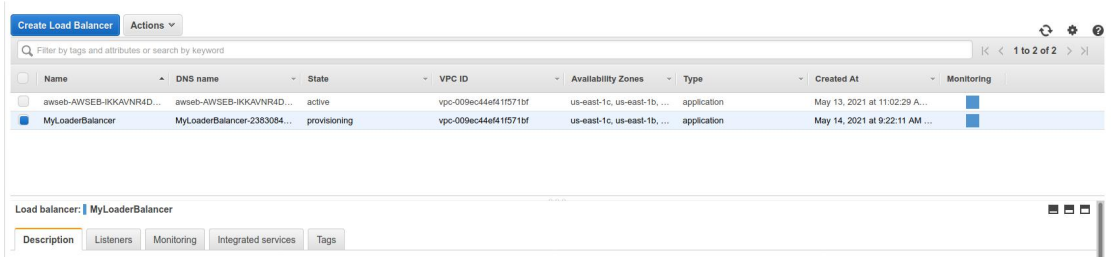
Step 2: click on Next and then and create a new security load balancer leave it as default and click on next.---> add configure routing.



.

Step 4: Register for targets ----> review and create.

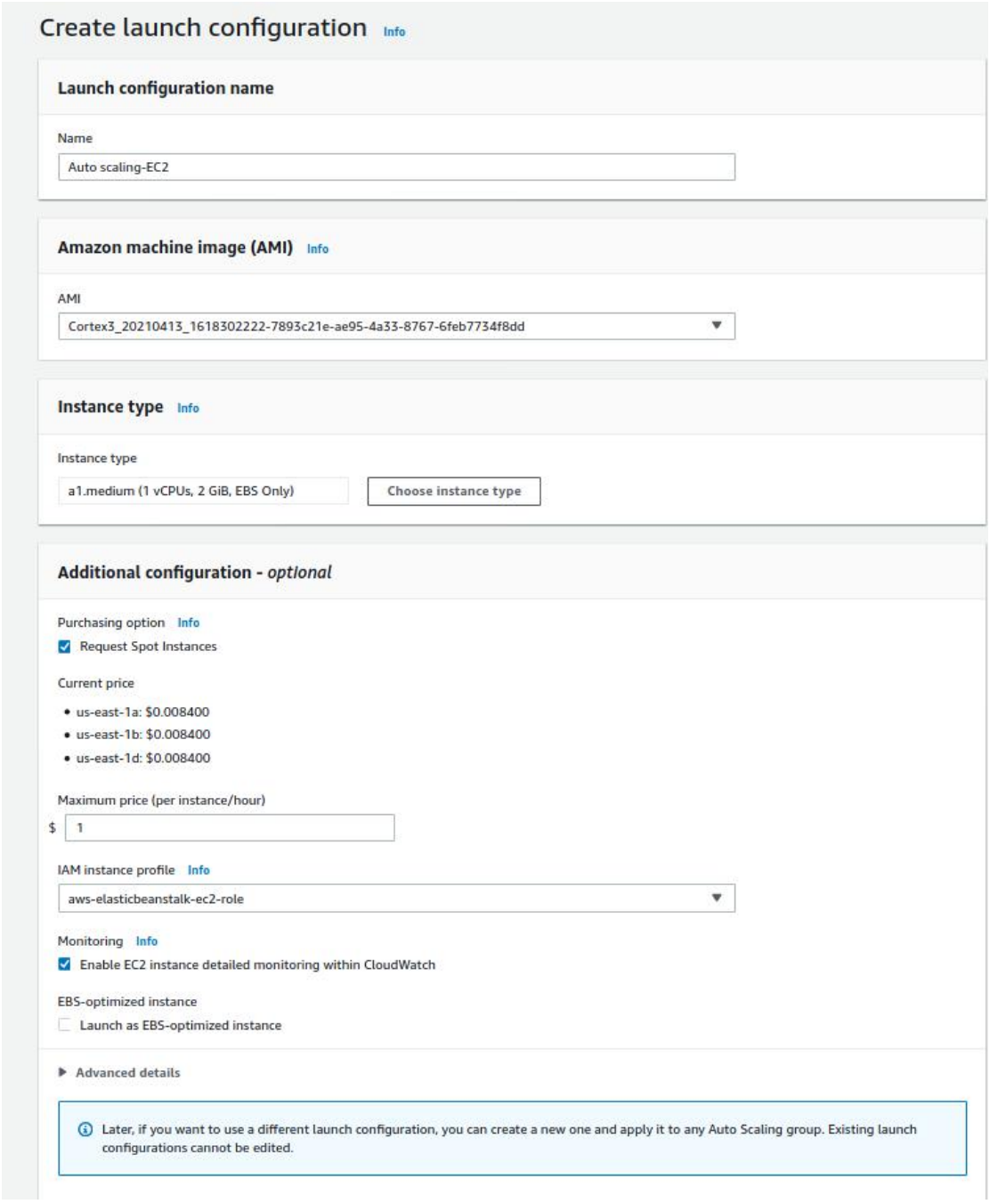


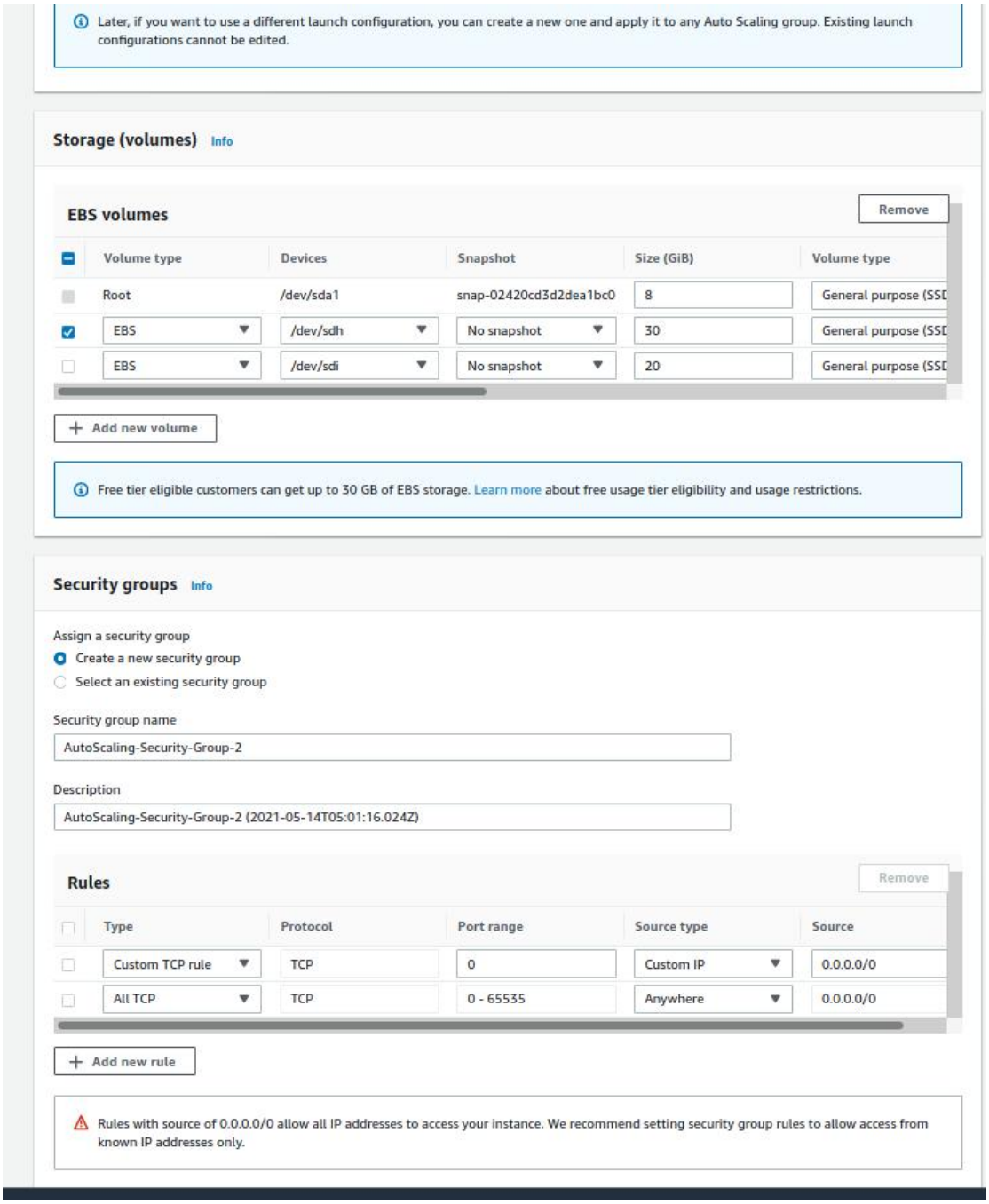


Creating the launch template for the Amazon EC2 Auto Scaling Group

Step 1: Creating a launch configuration (this will create the configuration of instances that EC2 spot wil create later if needed.)

Step 2: Navigate ot EC2--->Lauch cofiguration(scroll down)



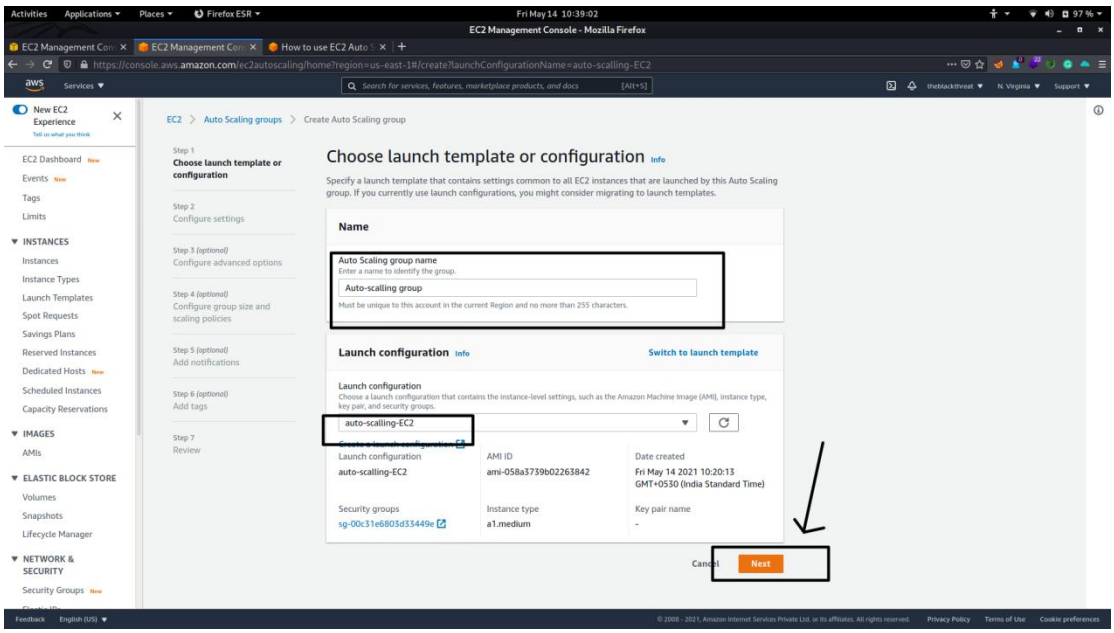


Step 3: Choose key pair then click on Create launch configuration.

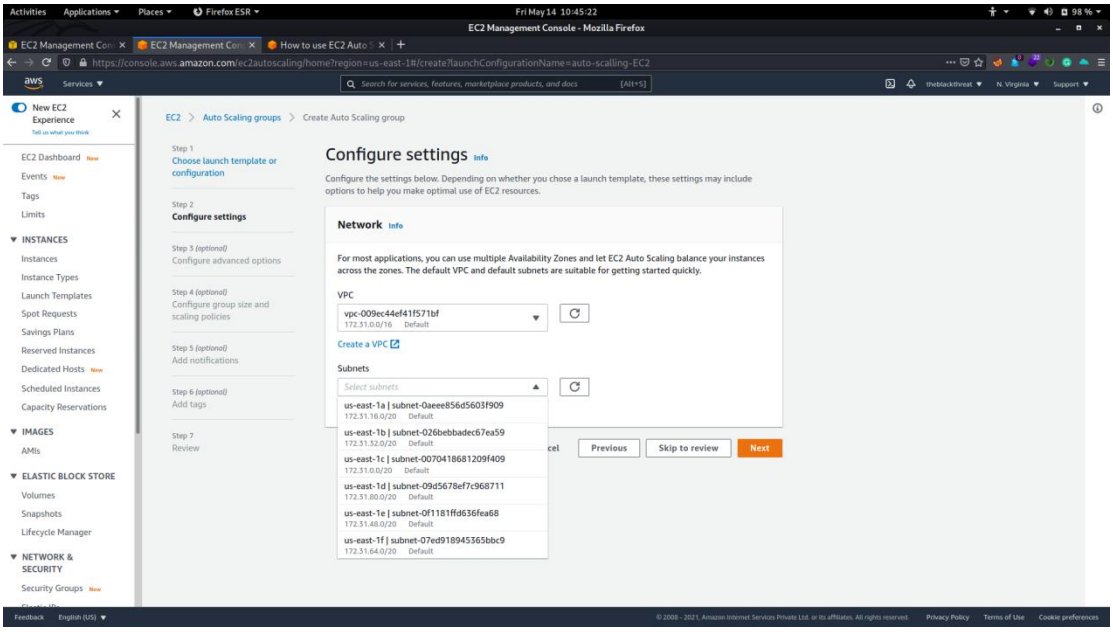
Step 4: Go to the dashboard of launch configuration and click on instance then go to action click on create **Autoscalling group.**

- EC2 >>Auto Scaling groups>>Create Auto Scaling group\

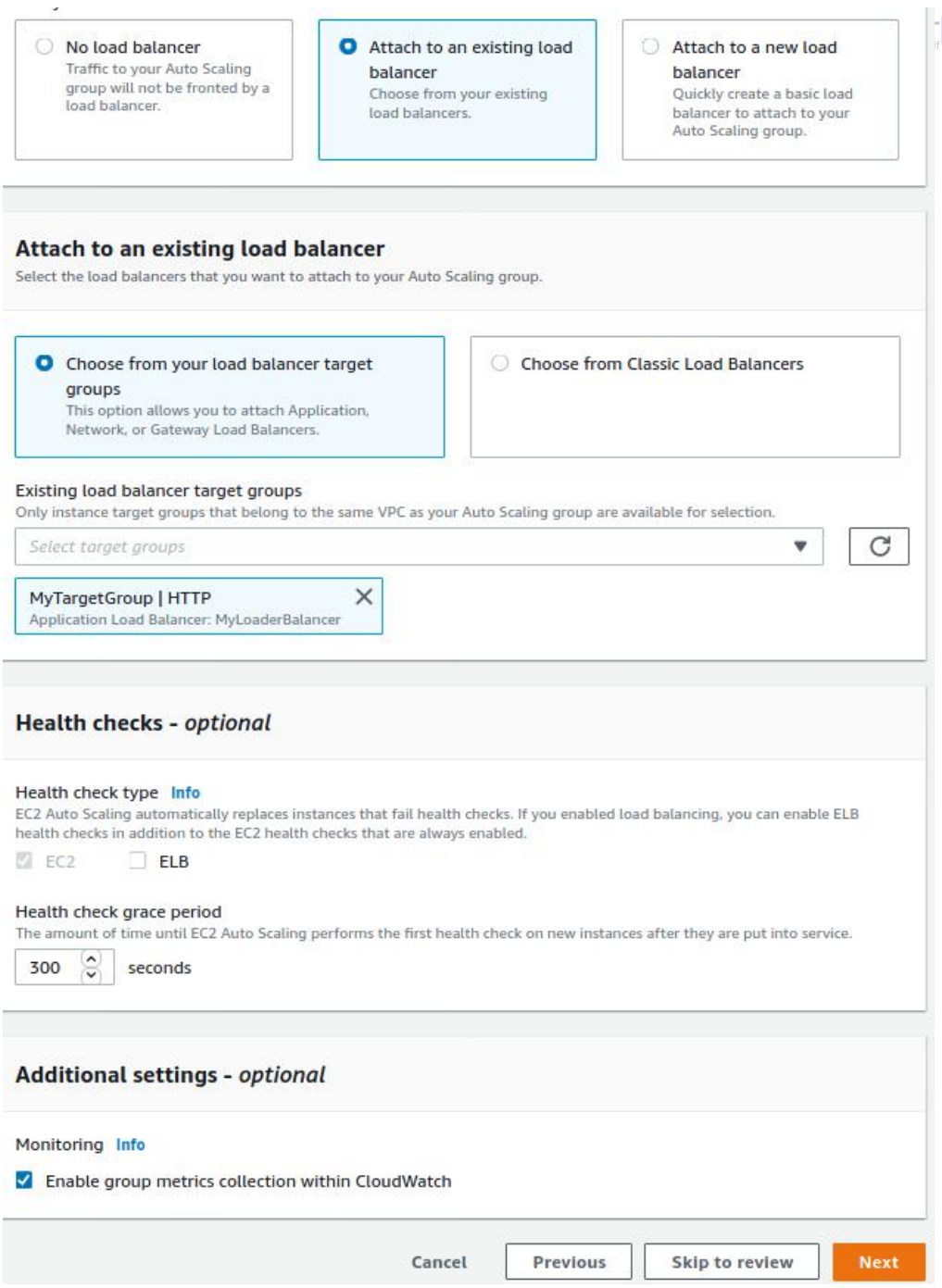
- Step 1 >> Choose launch template or configuration



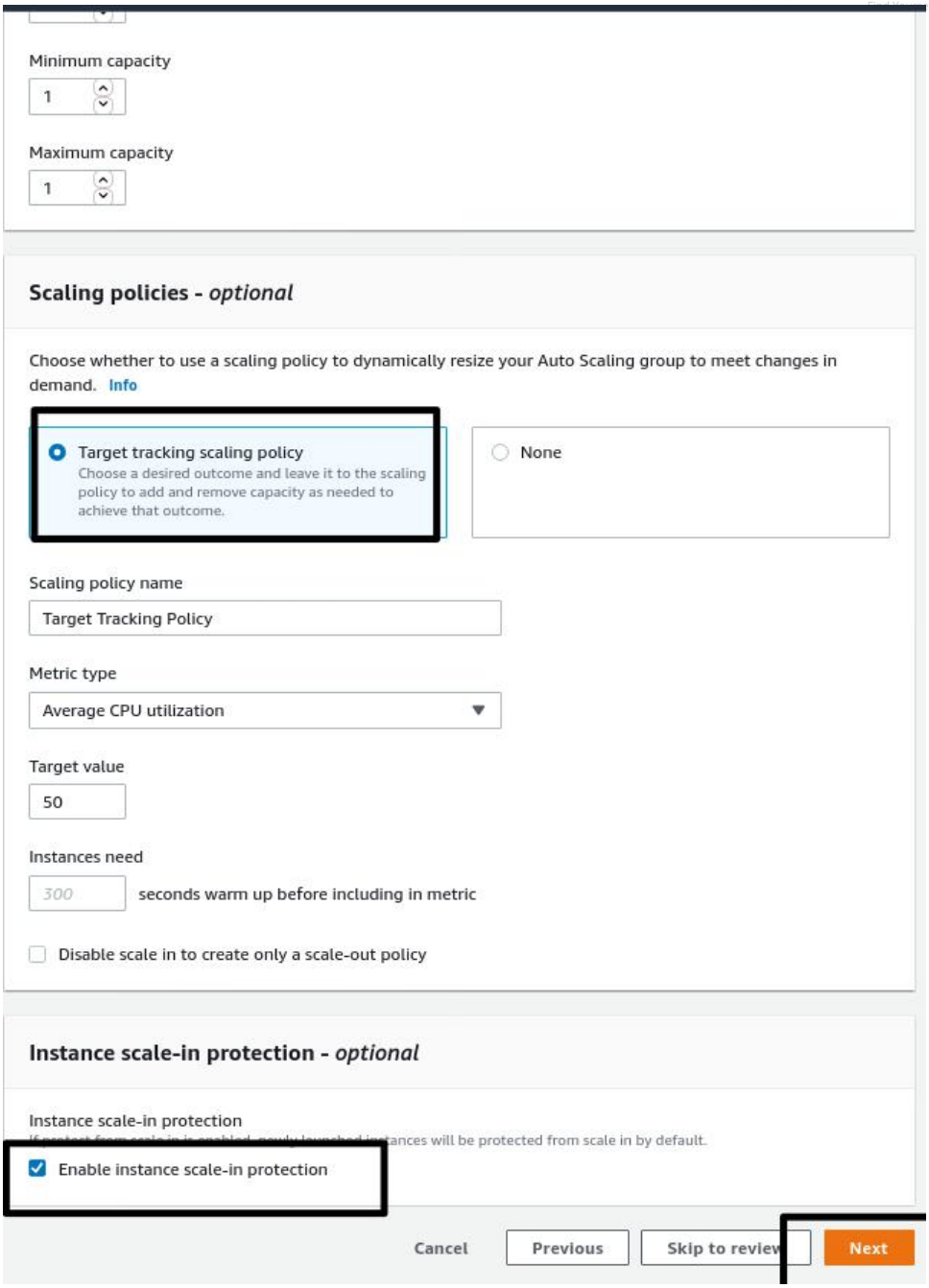
- · Step 2 >> Configure settings



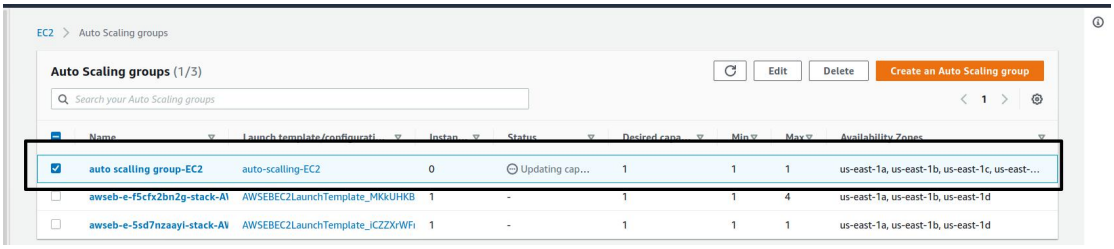
Select all Subnets



click on Next



Step 4: Add notification it will notify you on your added email ID.
Click next-->next-->review and create.

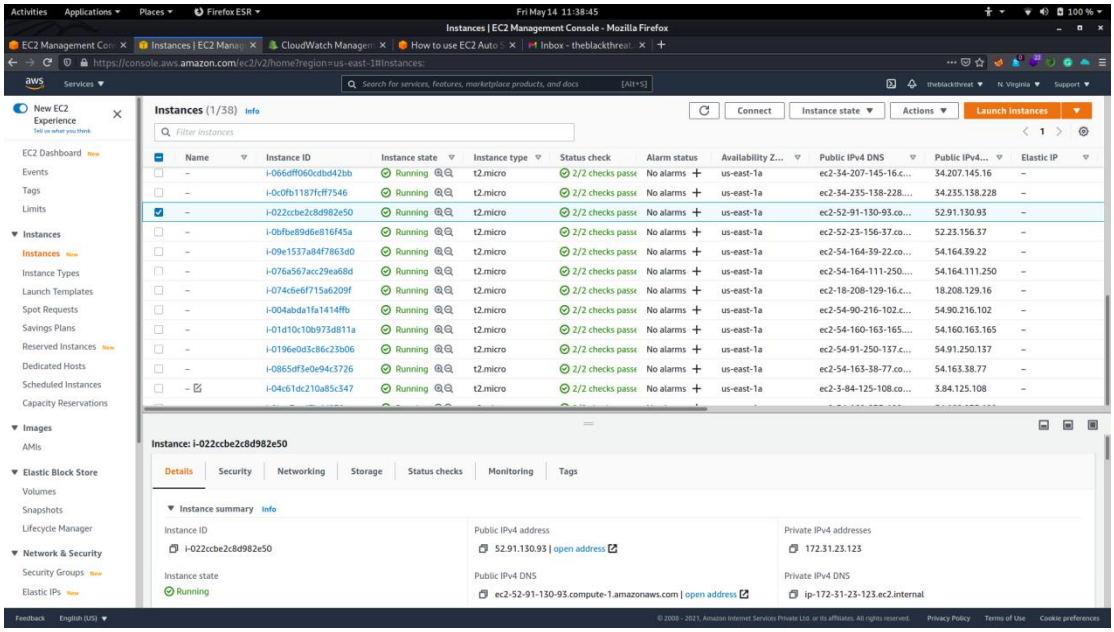


We have Created Autoscalling group so we need to check wheather it's working or not.

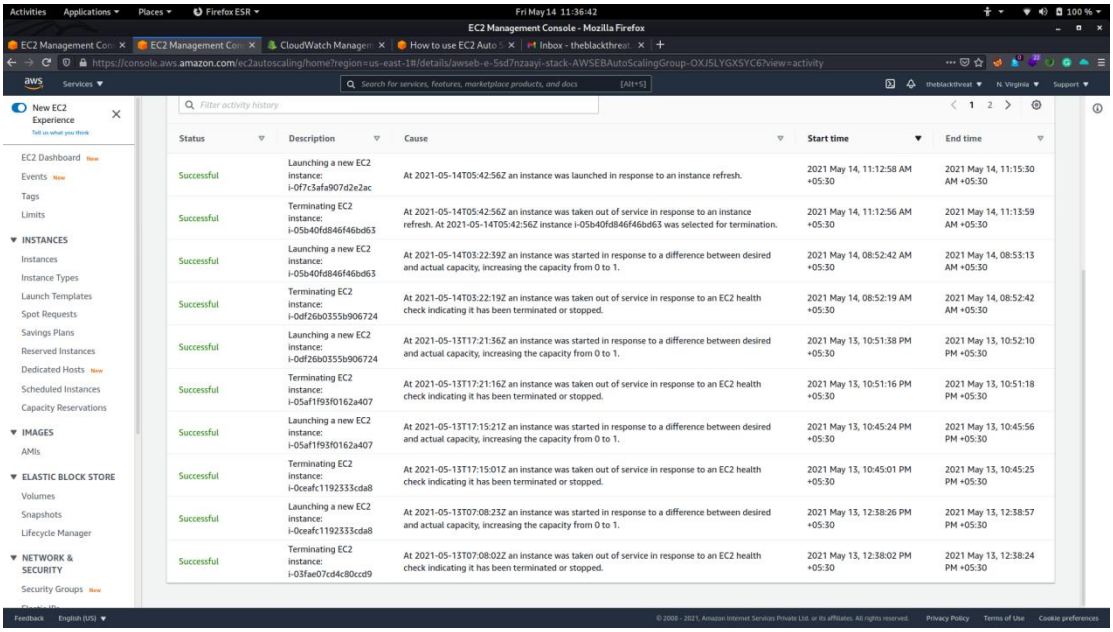
Monitor and test the Auto Scaling group

Step 1: Click on Launch Instances from Templeates select the template that we have created **myTemplate**.

Step 2: Set Number of Instances that you want to run at a time after some time all instances are running



Step 3:Click on the Details for all setup we have done then click on Management Instances

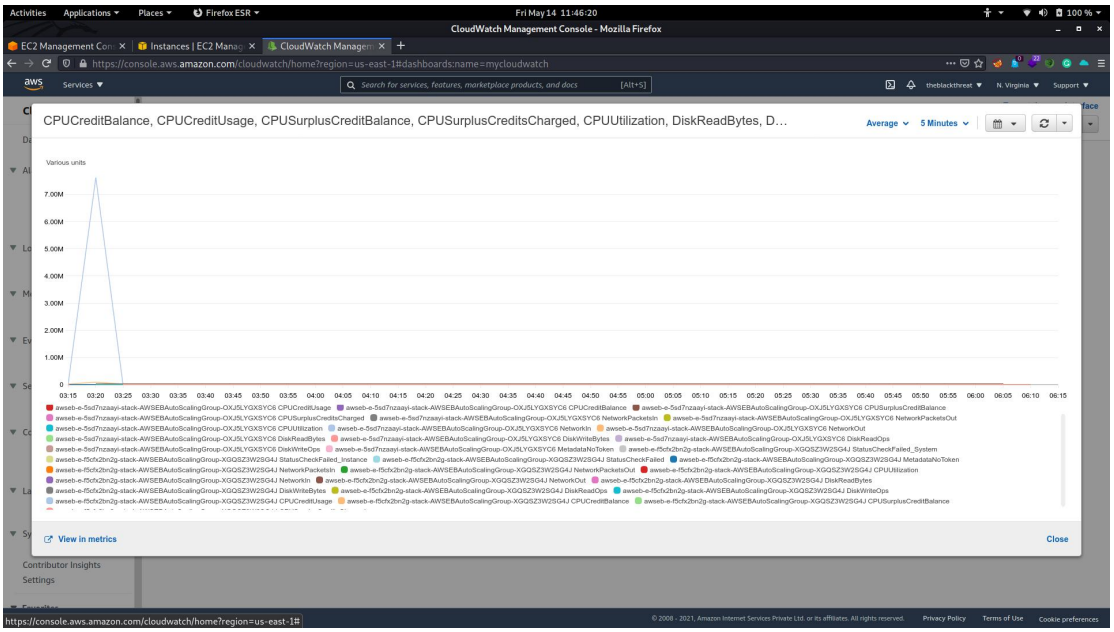


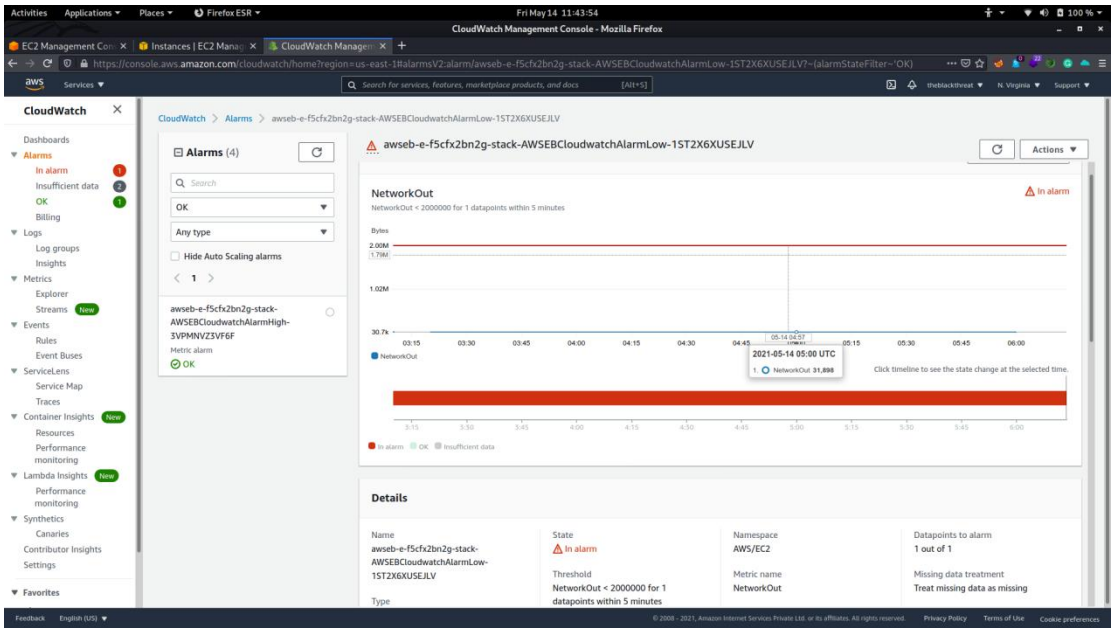
all Instances are successfully handled by Scaling groups.

Amazon CloudWatch

Amazon CloudWatch is a monitoring and management service that provides data and actionable insights for AWS, hybrid, and on-premises applications and infrastructure resources. With CloudWatch, you can collect and access all your performance and operational data in form of logs and metrics from a single platform.

For our instance spot we can create a cloud watch to monitor services.



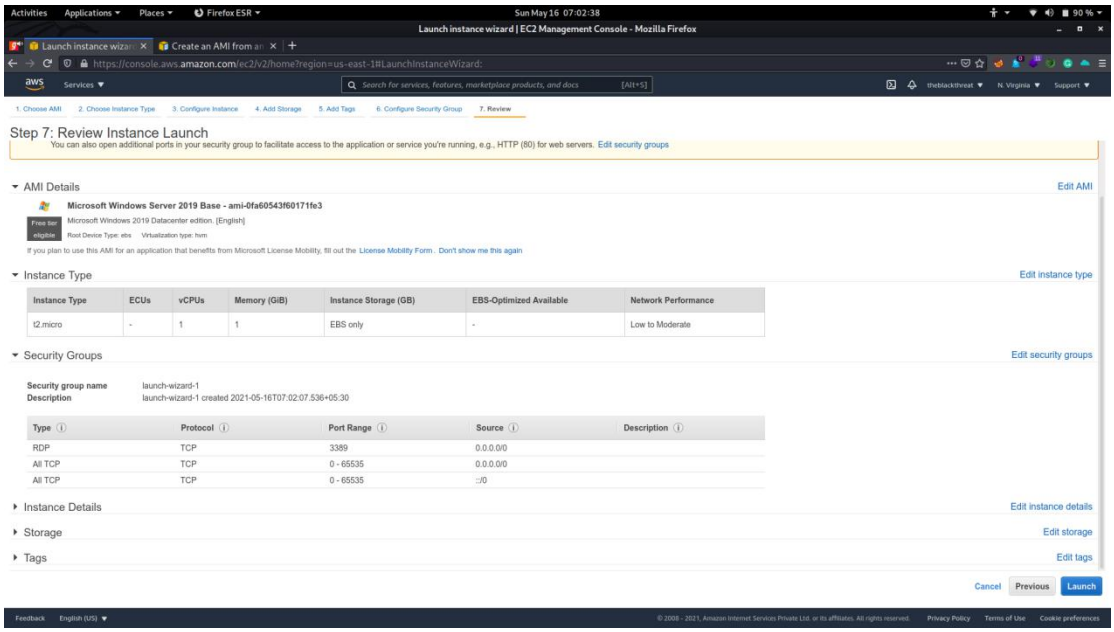


8.Objective: Working with Amazon Machine Image(AMI).

Amazon machine Image: This service on amazon provide a single configuration to launch multiple Instances. we can create multiple instances with single AMI configuration when needed same instance.

Step 1**:** First we need to launch an instance for sample:

free tier instance windows.

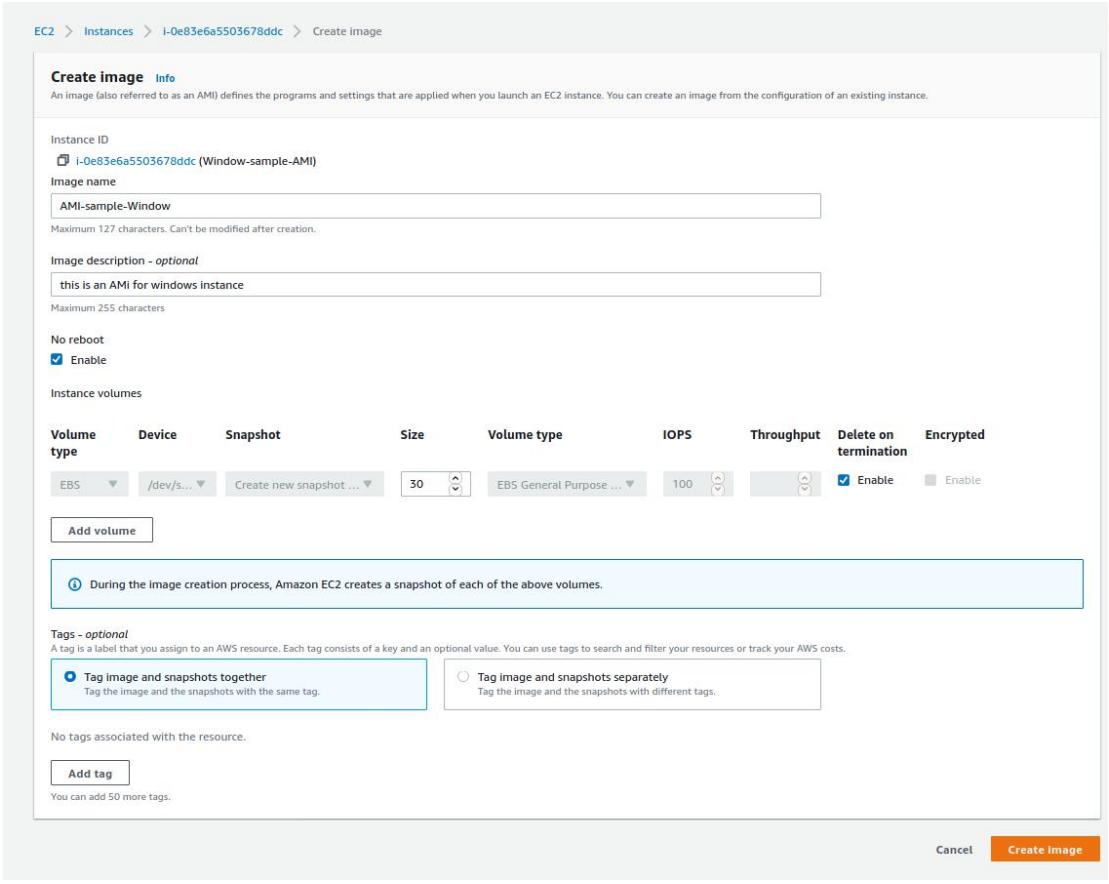


Review of Sample Instance

Step 2: after completion of instance connect instnaces via RDP client.

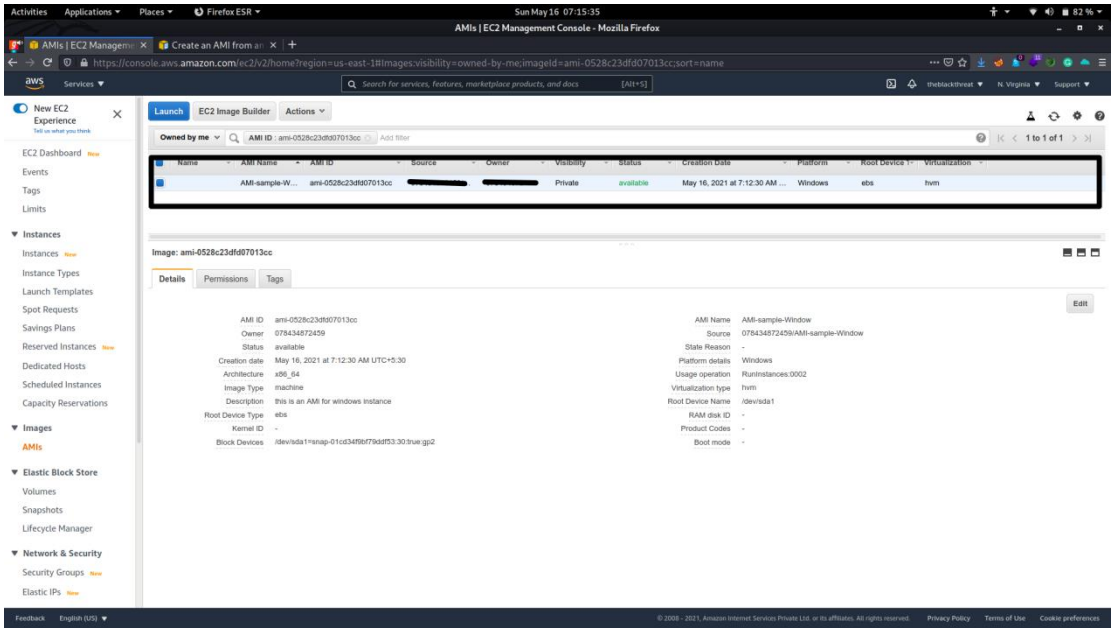
Step 3: Right-click the instance you want to use as the basis for your AMI, and choose **Create Image** from the context menu.

Step 4: Click on Create an Image then add a unique name and configuration. ----> click on create image.
(It will take some time to create an AMI, it will appear in the AMIs view in AWS Explorer.)



Click on enable to No root (If you choose No reboot, we can't guarantee the file system integrity of the created image.)

Wait for some time to be completion of pending status.

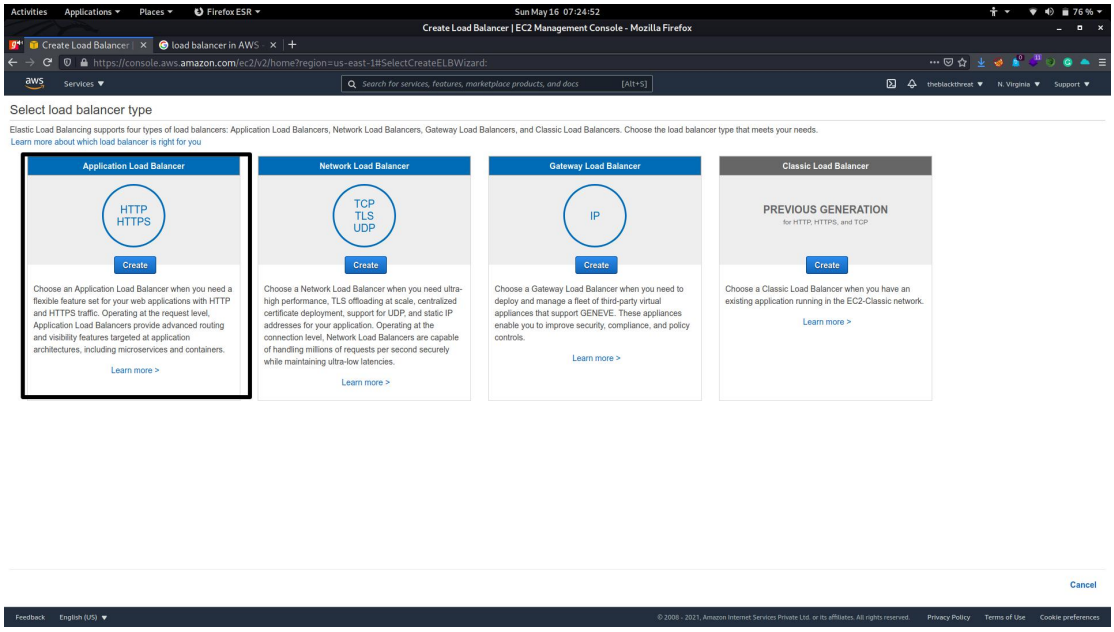


Now we can launch multiple instances with this sample.

9. Objective: Working with load balancers

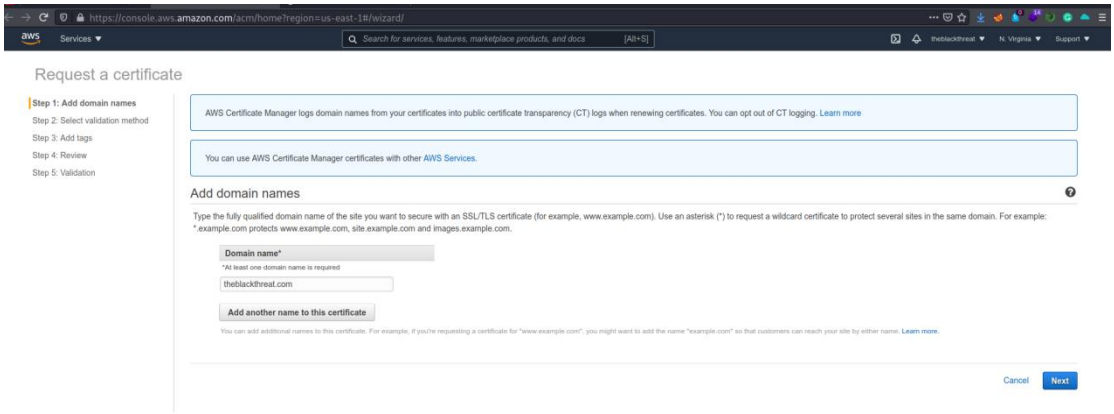
Load Balancers: To control the traffic request aws provide a service that is load balancer. It will distributes incoming application traffic across multiple EC2 instances in multiple Availability Zones. Load balancer detects the unhealthy instances and route the traffic only to healthy instances.

Creating a Load balancer:
Step 1: navigate to EC2 service and find load balancer on left pannel. ---> click on create a Laod balancer.
Select HTTP/HTTPS service .



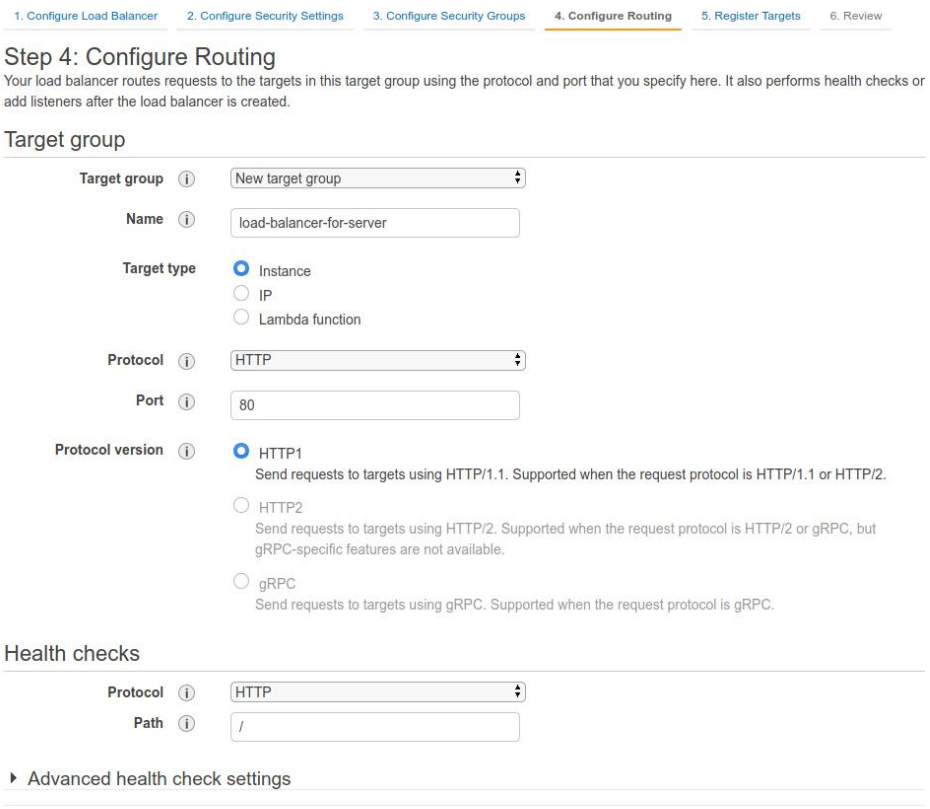
Step 2: Fill the basic configuration with name, interface, listeners, VPC avilable zones.
Note: If you choose HTTPS listners then it would call for domain that you should create further. otherwise you can go further settings.

Step 3: Configure Security Settings
If you don't have certificate then request a new one by given option on beside.



Step 4: Validation of certificates might take some time(~1 hour) after completion of this process go back to your load balancer then add a certificate.

Step 5: click on next to configuration routing ---> select your route or leave it as by default



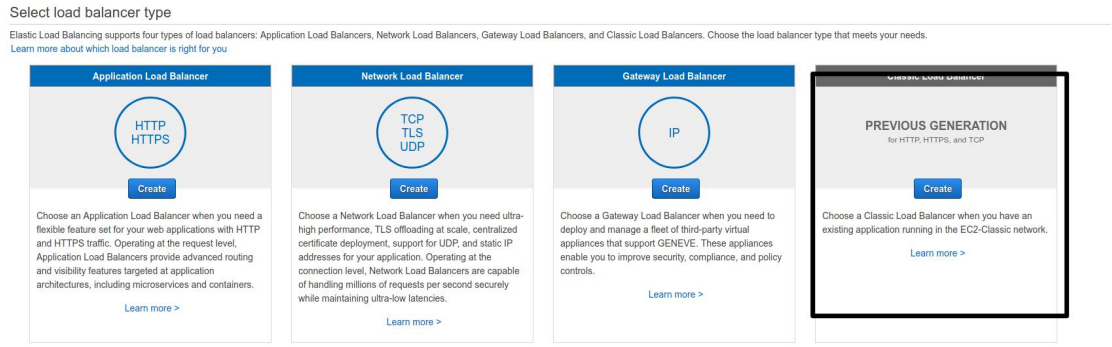
Step 6: click on next to configure Target register ----->select your registered domain and instance.

Step 7: click on review and create
Now you have done. congrats.

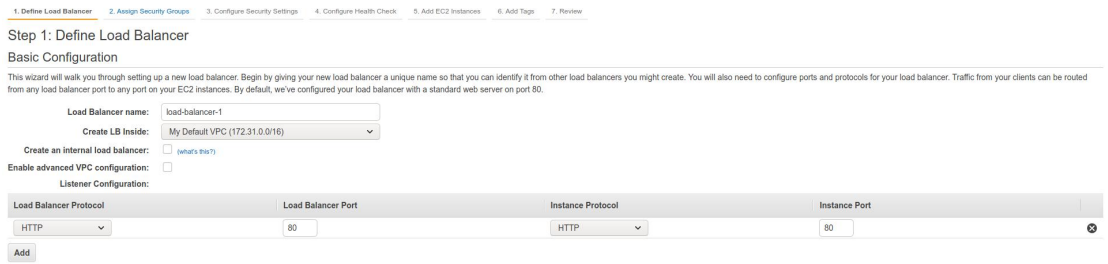
---------- ----------

A classic method to create Load balancer.

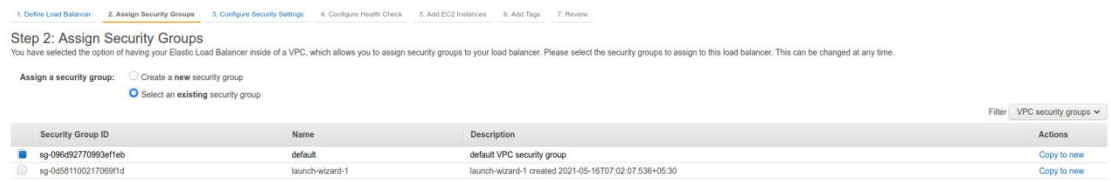
Step 1: Create on load balancer---->click on classic load balancer---->



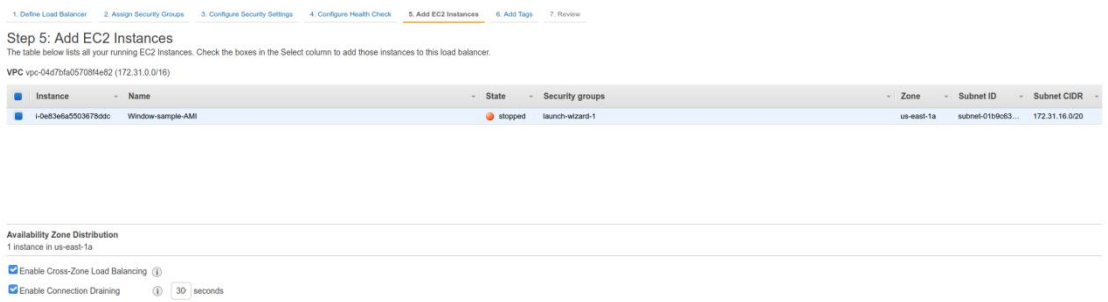
Step 2: name your balancer---> configure it



Step 3: click on next --->use your security group if you have your own otherwise click on defautl then go further.



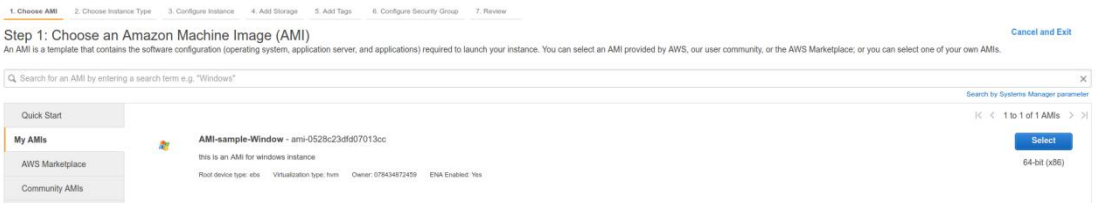
Step 4: click on next to set all by default then choose your instance.



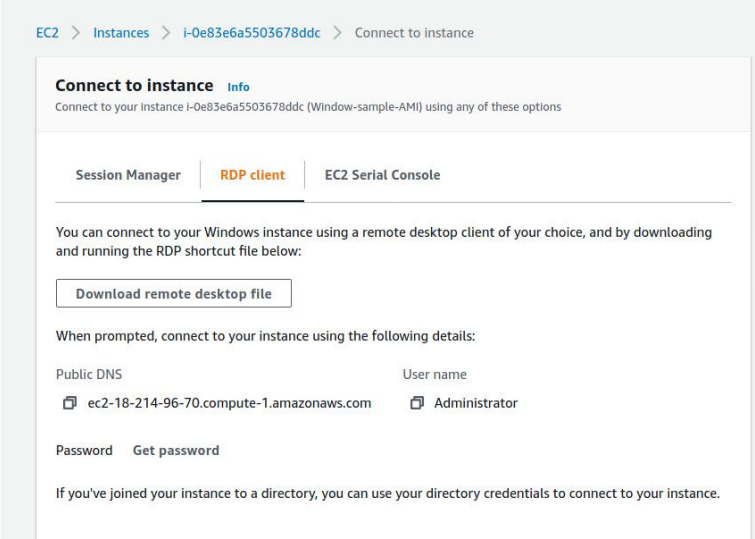
Step 5: add tag if needed, then click on **review and create**

Step 6: to verify that it working or not start your instance create an IIS server then traffic the request youcan monitor it on load balaners.

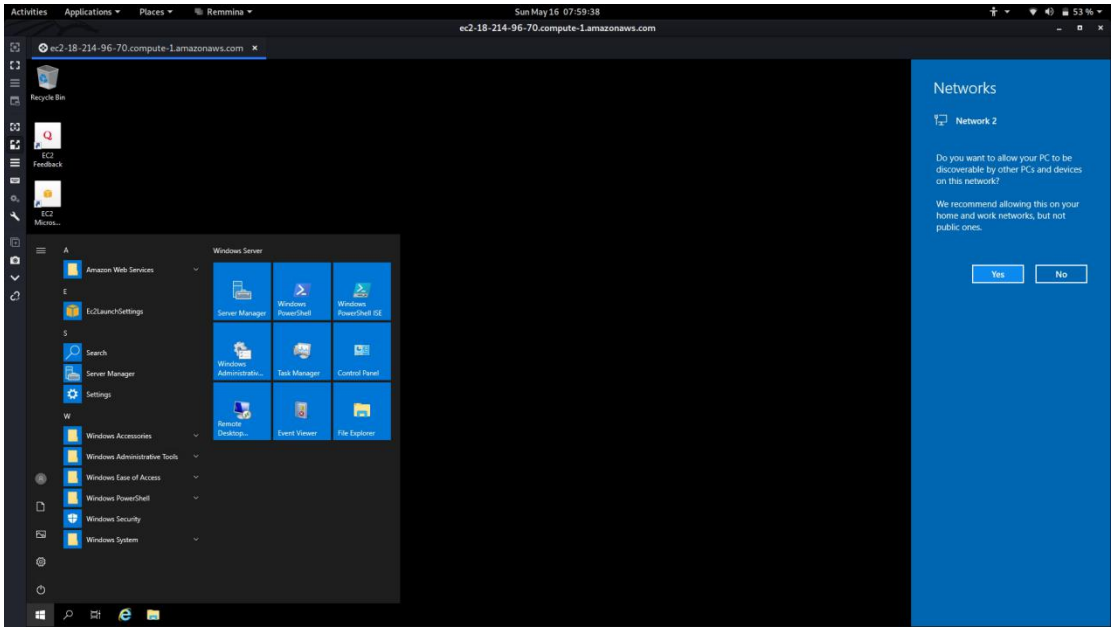
Step 7: Connect your instnace, go to the launch Instances----> my AMIs---> select your sample INstance.



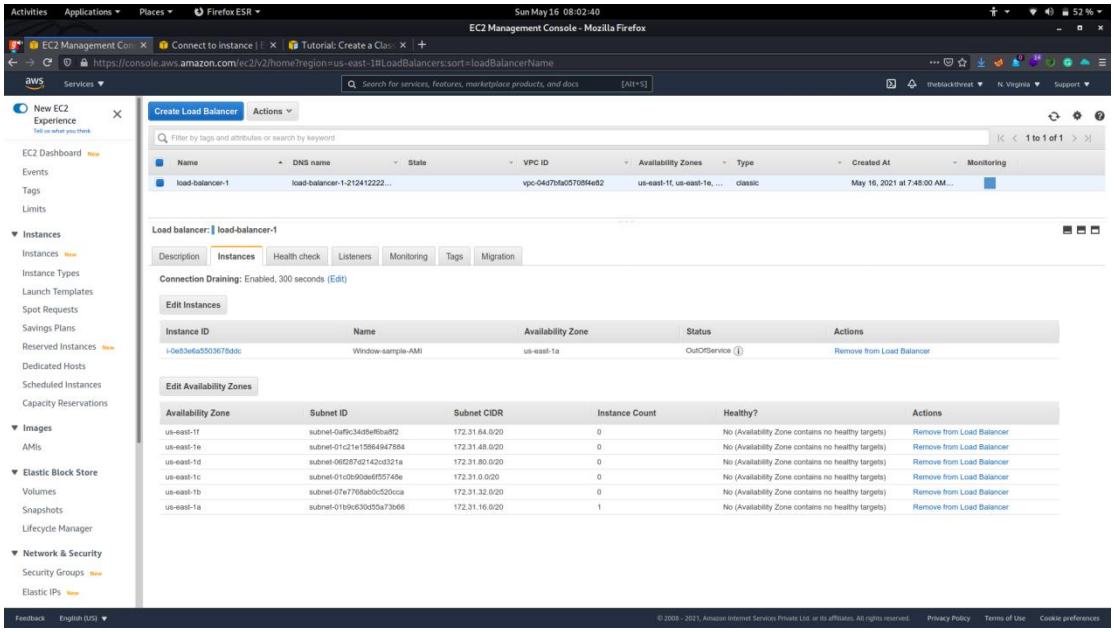
Step 8: Select and start your instance(run multiple instances.)

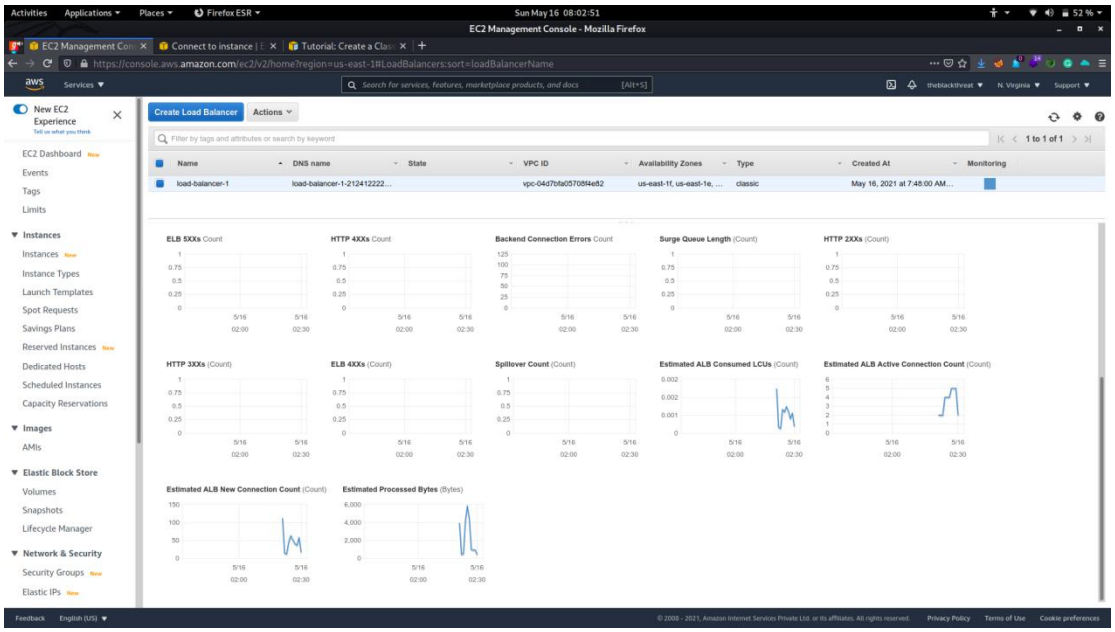


Step 9: connect it.



Step 10: go to load balancer and you can monitor it.





Now we have done both HTTP/HTTPS and classic load balancers.

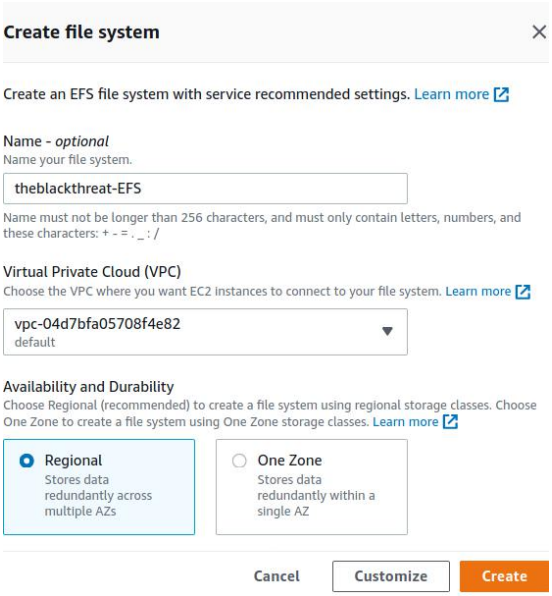
10. Objective: Working with EFS

Elastic File System(EFS): Amazone EFS is cloud based storage service that provide by AWS.This is designed to scalable, elastic, concurrent with some restrictions, and encrypted file storage for use with both AWS cloud services and on-premises resources.

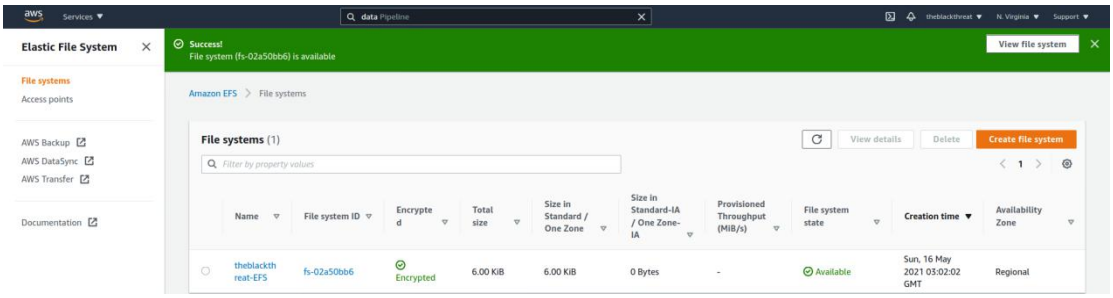
EBS and EFS are both faster than Amazon S3, with high IOPS and lower latency. ... EFS is best used for large quantities of data, such as large analytic workloads.
Differences:
EBS only accessable for single EC2 Instance with avilable regions.
EFS allows mount file system across multiple Instance and multiple regions.
Amazon S3 is an object store good at storing vast numbers of backups or user files.

Creating a EFS

Step 1:Navigate to Storage service find EFS click on it and create on create file system.



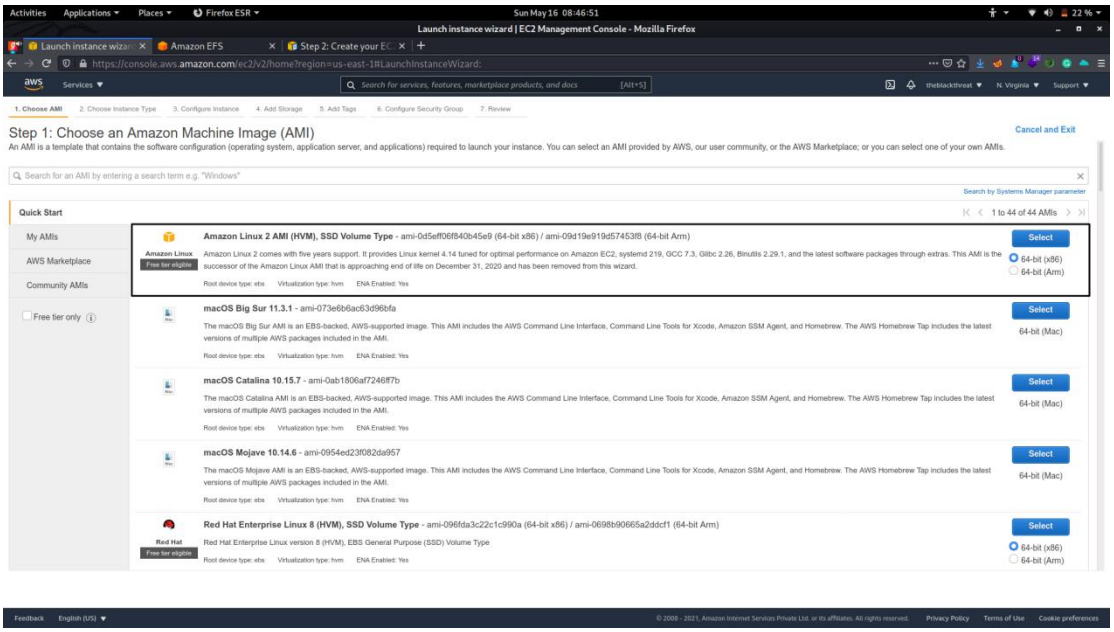
Step 2: create file system



Step 3: Now launch an Instance(Amazon Linux 2 AMI) and make sure you have choose EFS file system. lets do it.

1. Go to AMI choose your Instance (go to Objective 8)

Step 4: launch instances Amazon Linux 2 AMI----> configure it

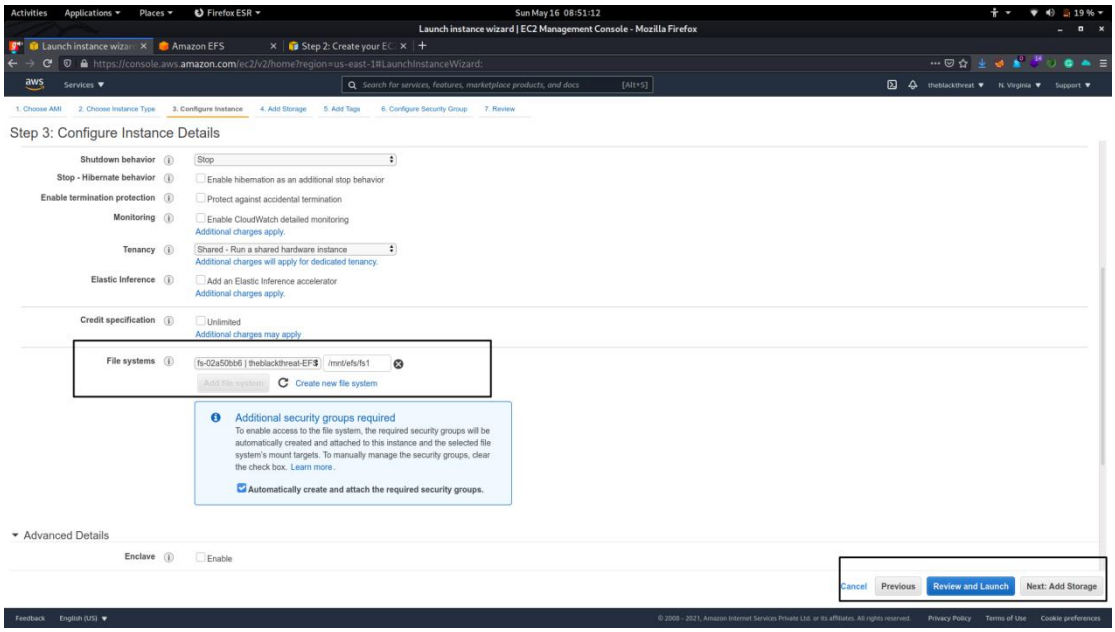


1: Choose an Amazon Machine Image (AMI), find an Amazon Linux 2 AMI at the top of the list and choose Select.
2: Choose an Instance Type, choose Next: Configure Instance Details.
3: Configure Instance Details, provide the following information:

Leave Number of instances at one.

Leave Purchasing option at the default setting.

For Network, choose the entry for the same VPC that you noted when you created your EFS file system in Step 1: Create your Amazon EFS file system.



For Subnet, choose a default subnet in any Availability Zone.

For File systems, make sure that the EFS file system that you created in Step 1: Create your Amazon EFS file system is selected. The path shown next to the file system ID is the mount point that the EC2 instance will use, which you can change.

The User data automatically includes the commands for mounting your Amazon EFS file system.

Choose Next: Add Storage and tags

Name your instance and choose Next: Configure Security Group.

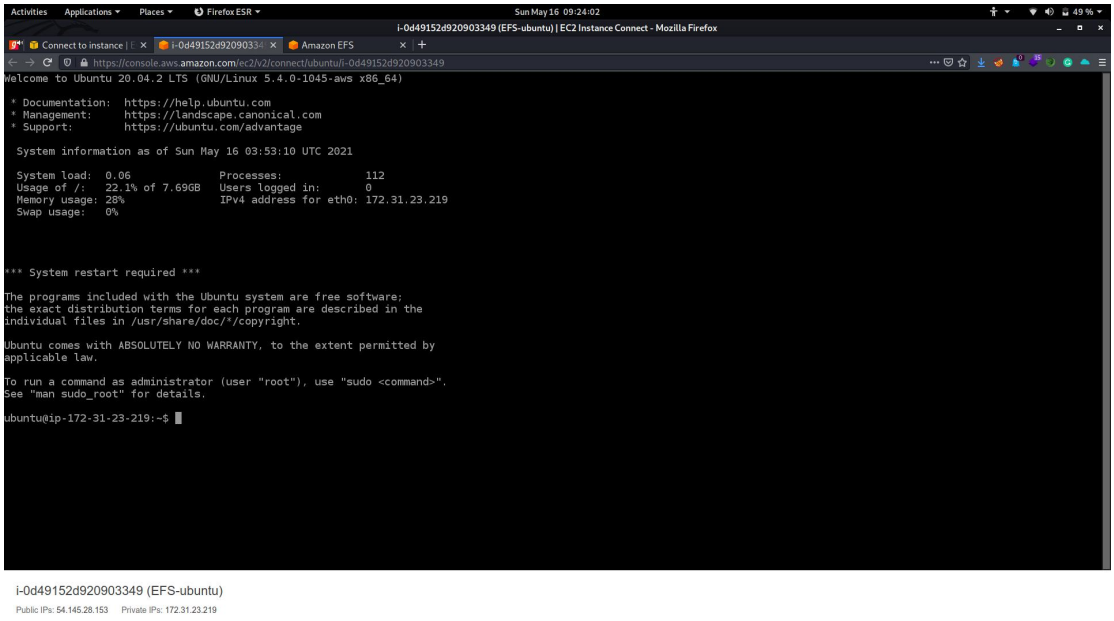
4: Configure Security Group, set Assign a security group to Select an existing security group. Choose the default security group to make sure that it can access your EFS file system.

Type: SSH
Protocol: TCP
Port Range: 22
Source: Anywhere 0.0.0.0/0

Choose Review and Launch.

Choose Launch.

Select the check box for the key pair that you created, and then choose Launch Instance



Step 5: If you have choosen linux system then you can attach it.

Step 5: after complete connetion of Linux box use these cmds to mount the EFS. and now you have done.

$ sudo mount -t efs -o tls fs-02a50bb6:/ efs
$ sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
fs-02a50bb6.efs.us-east-1.amazonaws.com:/ efs

After this operation we will mount aur instacne to the EFS.
so we have completed it.

11. Objective: Accessing relational database using Windows and Linux machine

1st Connecting the Relational Database using Linux Machine.

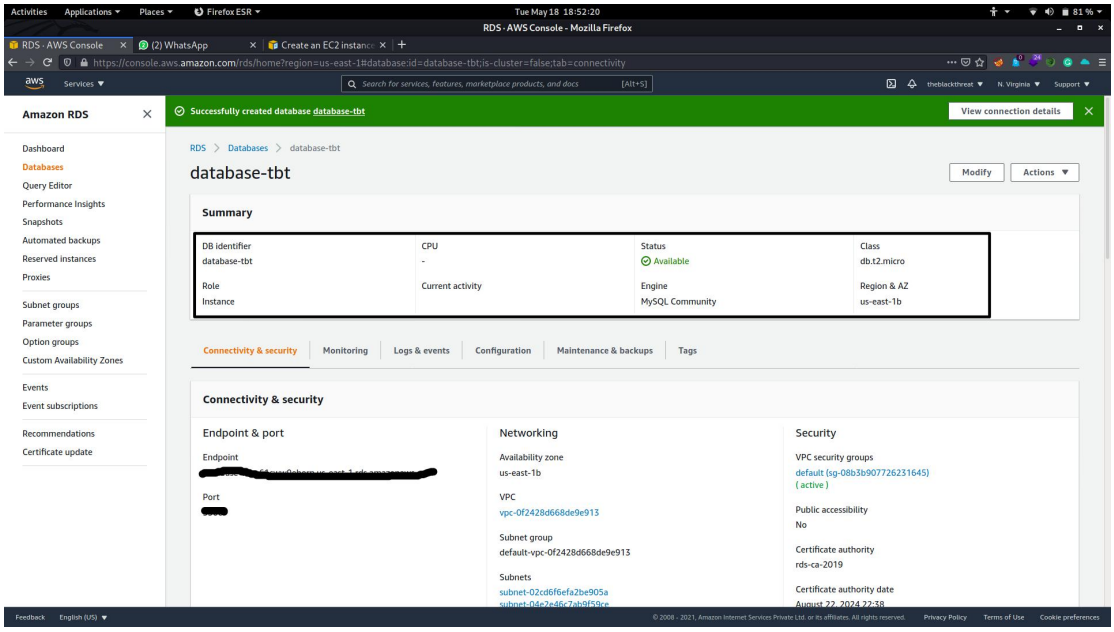
First we will creating a RDs database from Storage---> RDS.

Steps to follow.

A) Create a MySQL DB instance:
We will use Amazon RDS to create a MySQL DB Instance with db.t2.micro DB instance class, 20 GB of storage, and automated backups enabled with a retention period of one day

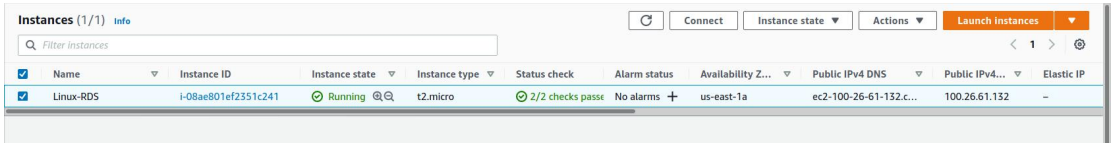
a) On the top right corner of the Amazon RDS console, select the Region in which you want to create the DB instance.
b) In the Create database section, choose Create database.
c) You now have options to select your engine. For this tutorial, click the MySQL icon, leave the default value of edition and engine version, and select the Free Tier template.
d) You will now configure your DB instance. The list below
shows the example settings you can use for this tutorial:
e) Settings: DB instance identifier: Type a name for the DB instance that is unique for your account in the Region that you selected.
Master username: Type a username that you will use to log in to your DB instance.Master password: Type a password that contains from 8 to 41 printable ASCII characters (excluding /,", and @) for your master user password.
Confirm password: Retype your password
f) Instance specifications: DB instance class: Select db.t2.micro --- 1vCPU, 1 GIB RAM. Storage type: Select General Purpose (SSD). Allocated storage: Select the default of 20 to allocate 20 GB of storage for your database.
Enable storage autoscaling: If your workload is cyclical or unpredictable, you would enable storage autoscaling to enable RDS to automatically scale up your storage when
needed.

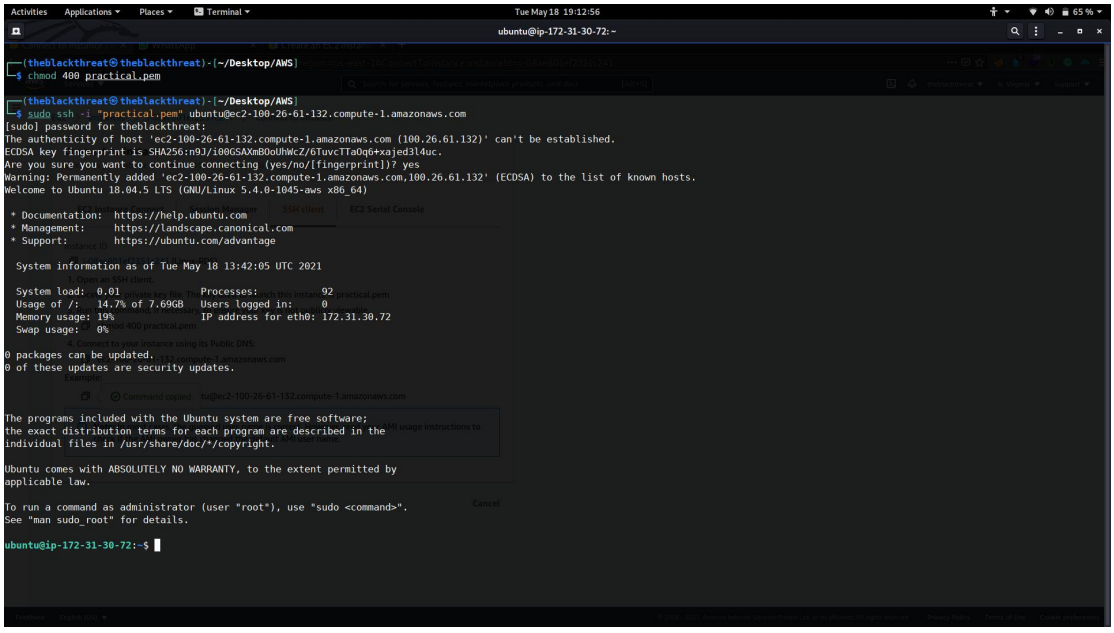
g) Connectivity
Virtual Private Cloud (VPC): Select Default VPC. For more information about VPC
h) Database options
Database name: Type a database name that is 1 to 64 alpha- numeric characters. If you do not provide a name, Amazon RDS will not automatically create a database on the DB instance you are creating.
DB parameter group: Leave the default value. For more
information, see Working with DB Parameter Groups.Option group: Leave the default value. Amazon RDS uses option groups to enable and configure additional features. For more information, see Working with Option Groups.



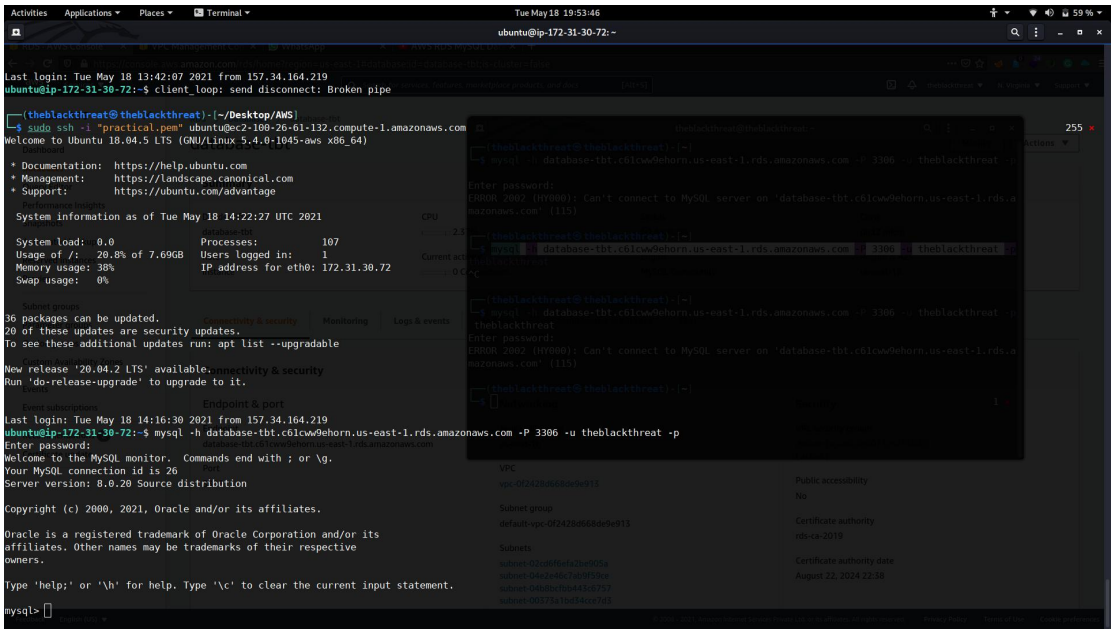
Now Step 2:

Create EC2 Linux Instances. go to first experiment. and Connect it.





STep 4: now to connect the database we need to create a local server on linux use following cmds to create server.



Note: if not connect then go throug the default security group and then edit inbound rule

All trafic--->all TCP ---ip(anywhere). then save it

after some time it will be connected.

Step 5: Install mysql server and update the instance

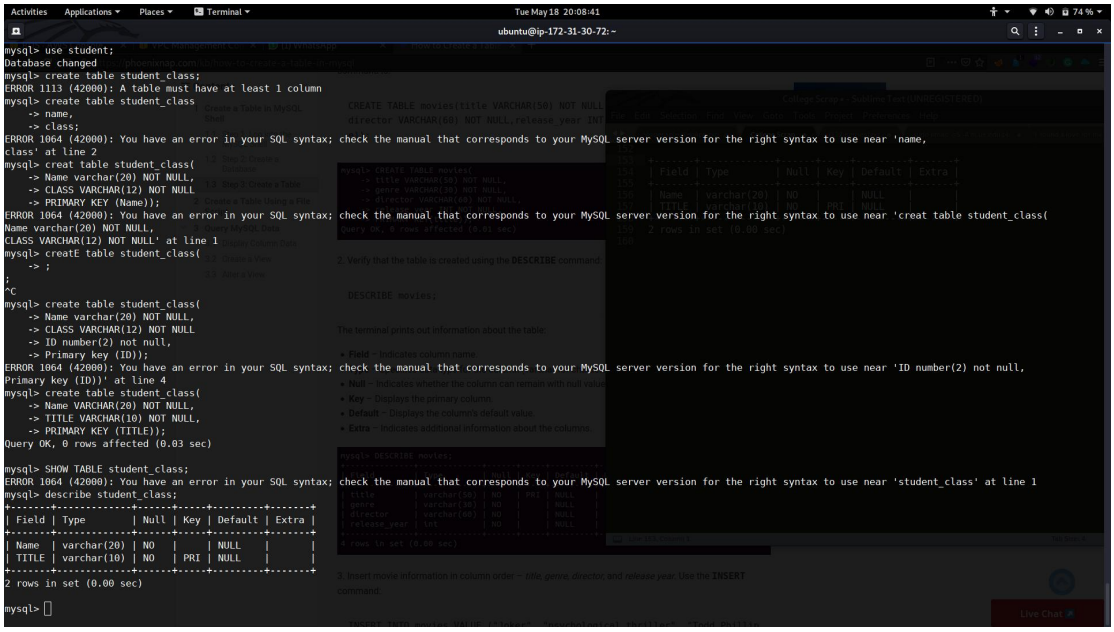
$ sudo apt update
$ sudo apt install mysql-server

Step 5: Now create databse that will conects to the RDS DB instance.

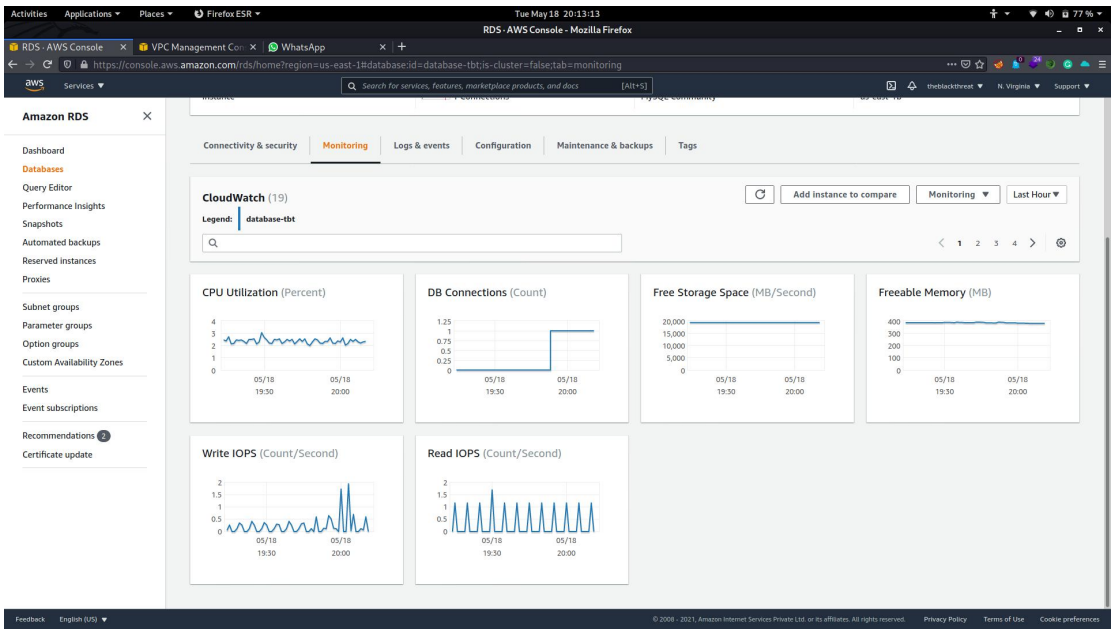
```
mysql> create db student
mysql> use student;

mysql> create table student_class(
    -> Name VARCHAR(20) NOT NULL,
    -> TITLE VARCHAR(10) NOT NULL,
    -> PRIMARY KEY (TITLE));
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW TABLE student_class;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near 'student_class' at line 1
mysql> describe student_class;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Name  | varchar(20) | NO   |     | NULL    |       |
| TITLE | varchar(10) | NO   | PRI | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)2 rows in set (0.00 sec)
```



to check that all things is good go to RDS database now click on monitering



we can see that all data traffic is working when we created a databse on terminal.
Now we have done this to connecting with linux Instance with RDS DB instance.

---

12. Virtual Machine using VMware

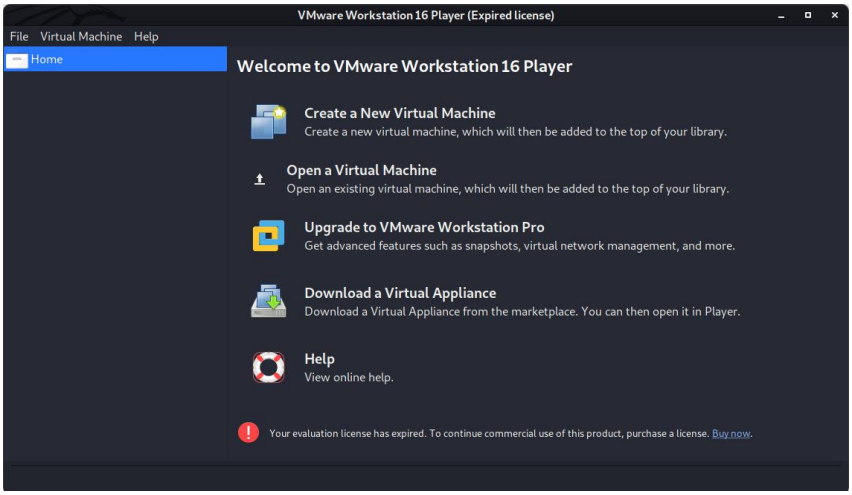
what is virtualization?

Virtualization uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer-processors, memory, storage and more-to be divided into multiple virtual computers, commonly called virtual machines (VMs).

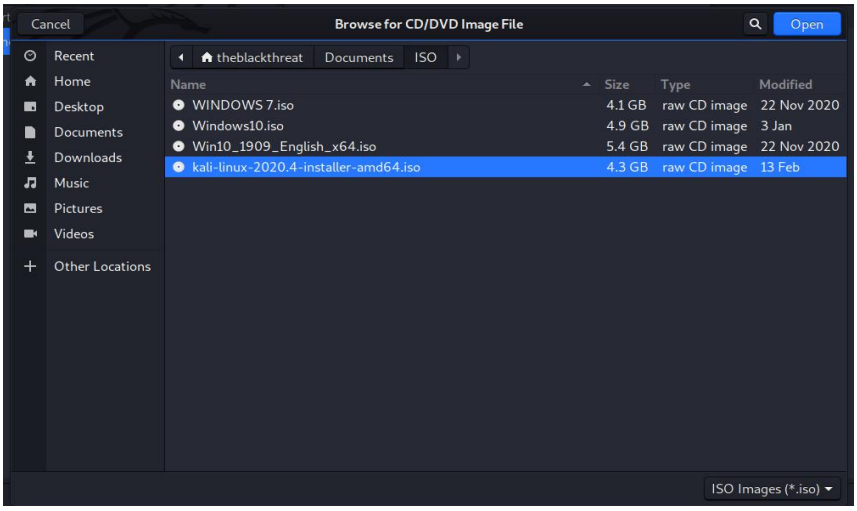
VMware: VMware Workstation is the first product ever released by the software company. It enables users to create and run VMs directly on a single Windows or Linux desktop or laptop. Those VMs run simultaneously with the physical machine. Each VM runs its own OS such as Windows or Linux.

Ceating Kali linux in VMware:

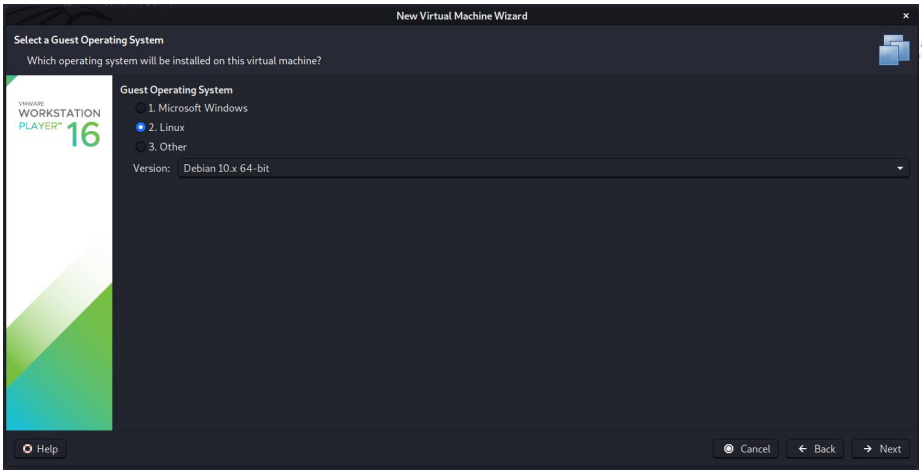
Step 1: Open VMware or download from here



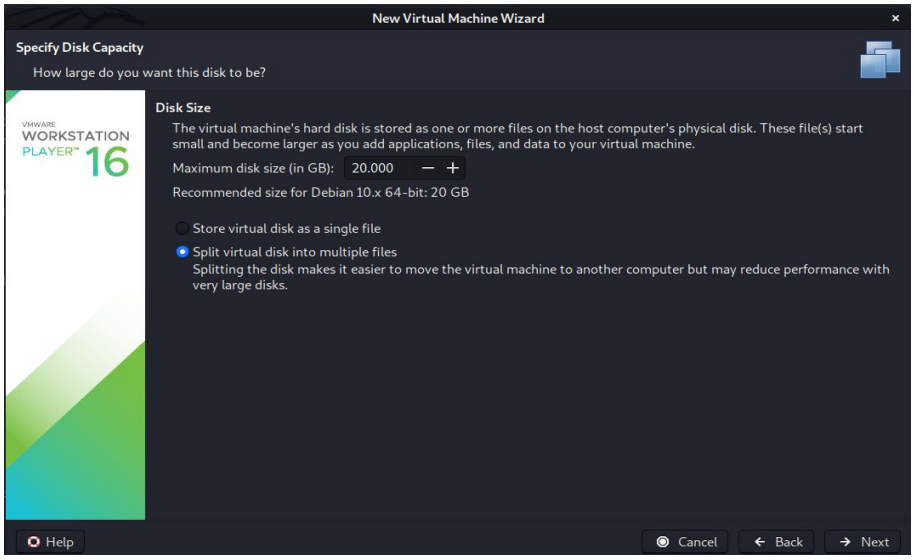
Step 2: Click on Create a New Virtual Machine. then browse your linux ISO file.



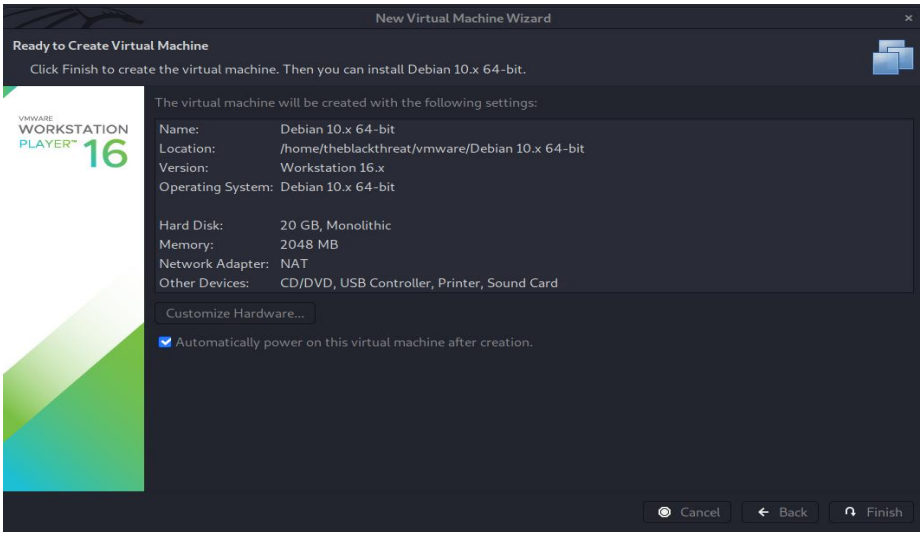
Step 3: Click Next and choose Linux ----> in context of kali--->Debian based version-->.Next



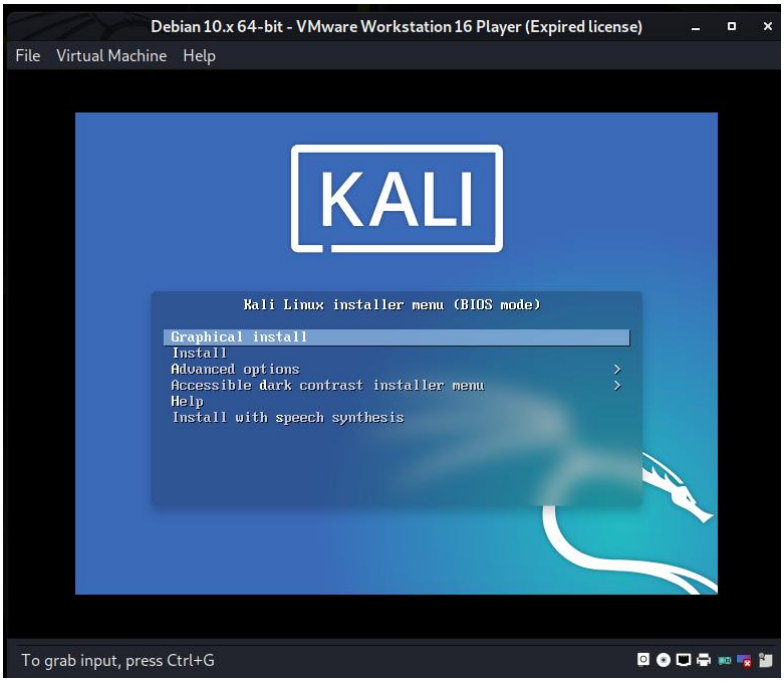
Step 4: Select your comfortable storage. and click on split it on single device ---> Next.

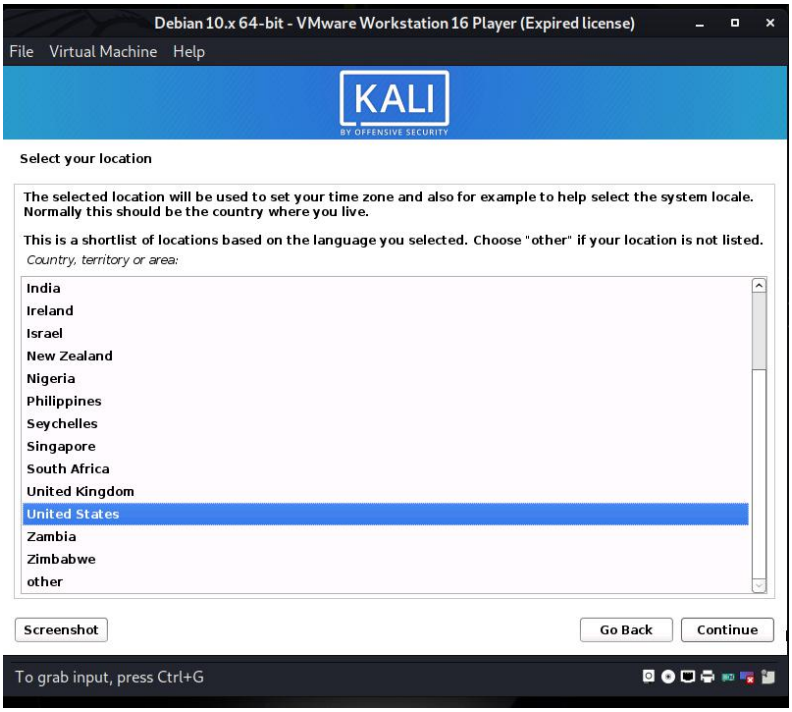


Step 5: See all configuration and then finish it.

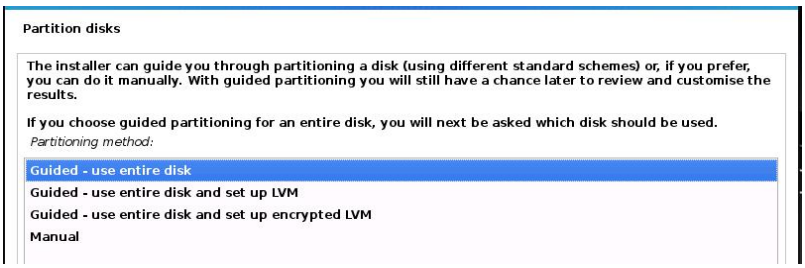


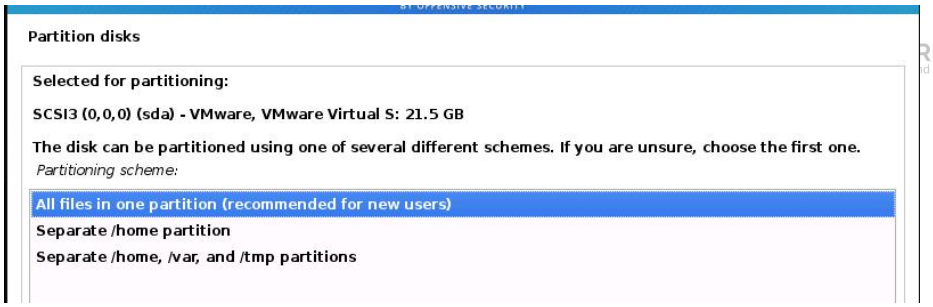
Step 6: after finish installation process if going.



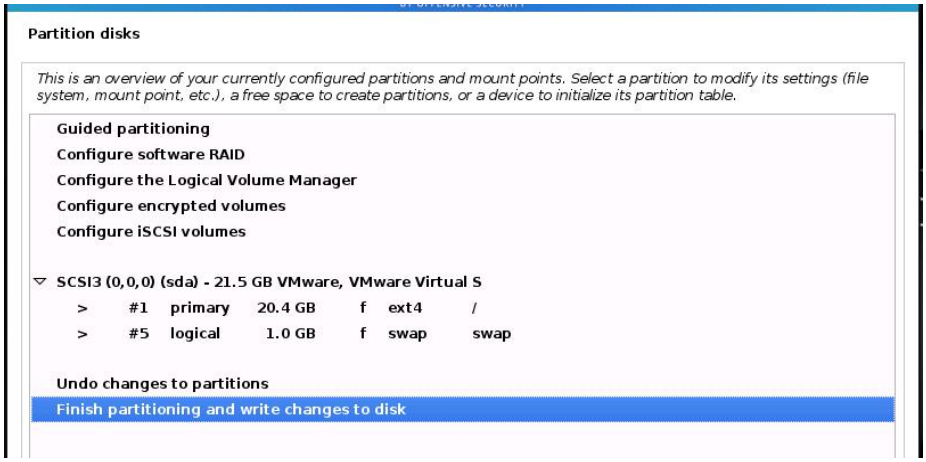


after this add system name and passowrd and continue

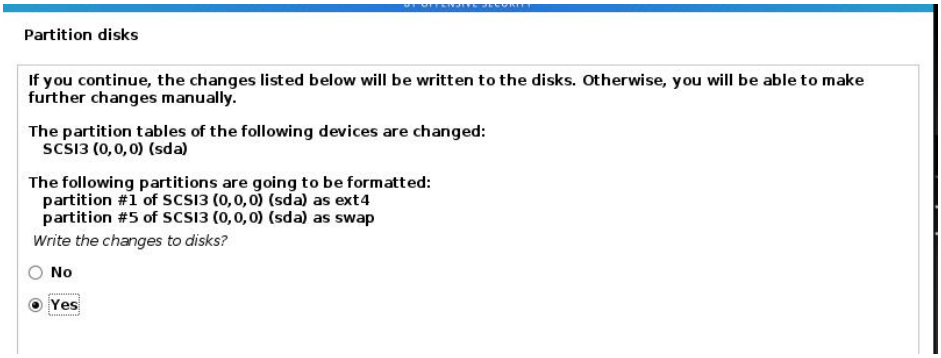




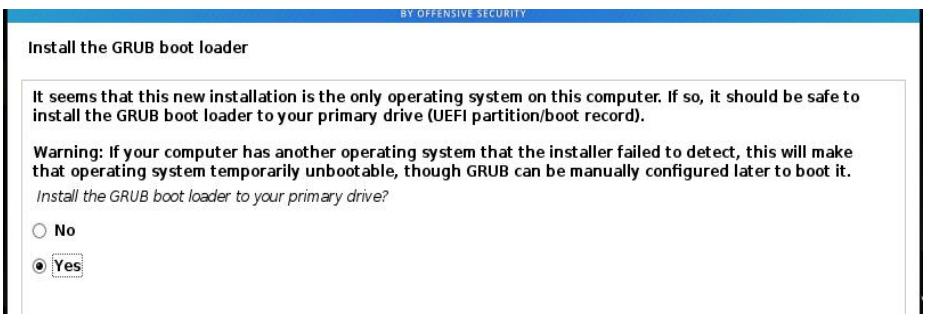
Stpe 7: Partition of disk all file are should be in same place so choose first one.



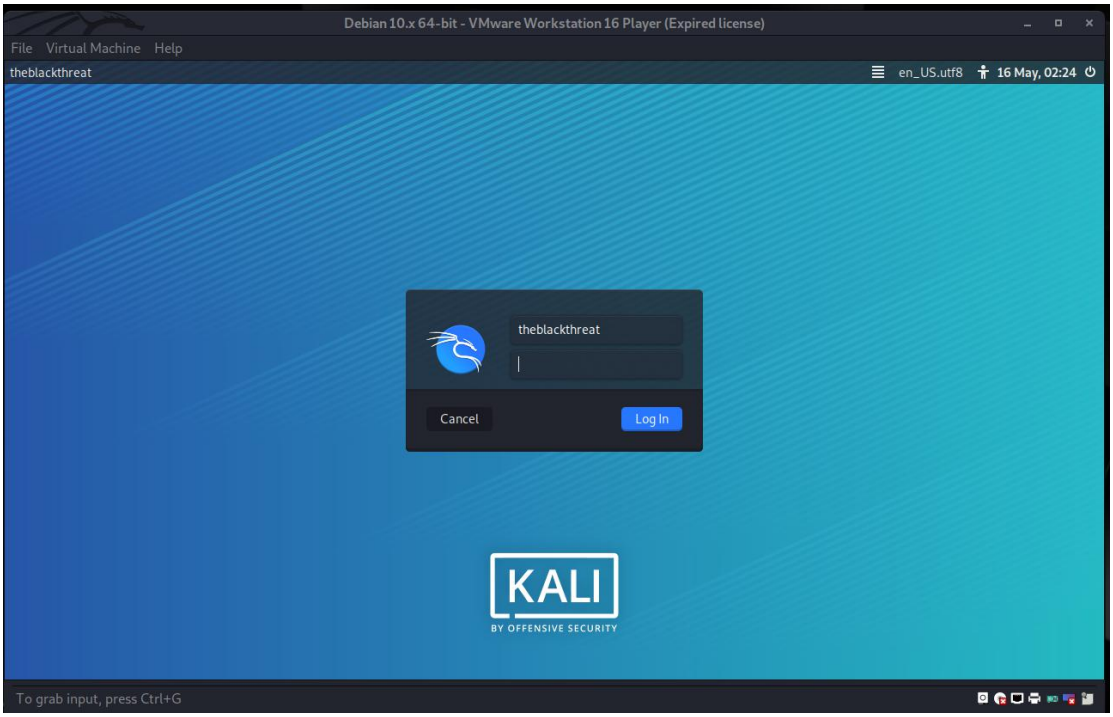
Step 8: give the permission for disk changes this will allow you to move further.



Step 9: Install GRUB Boot loader click on yes.



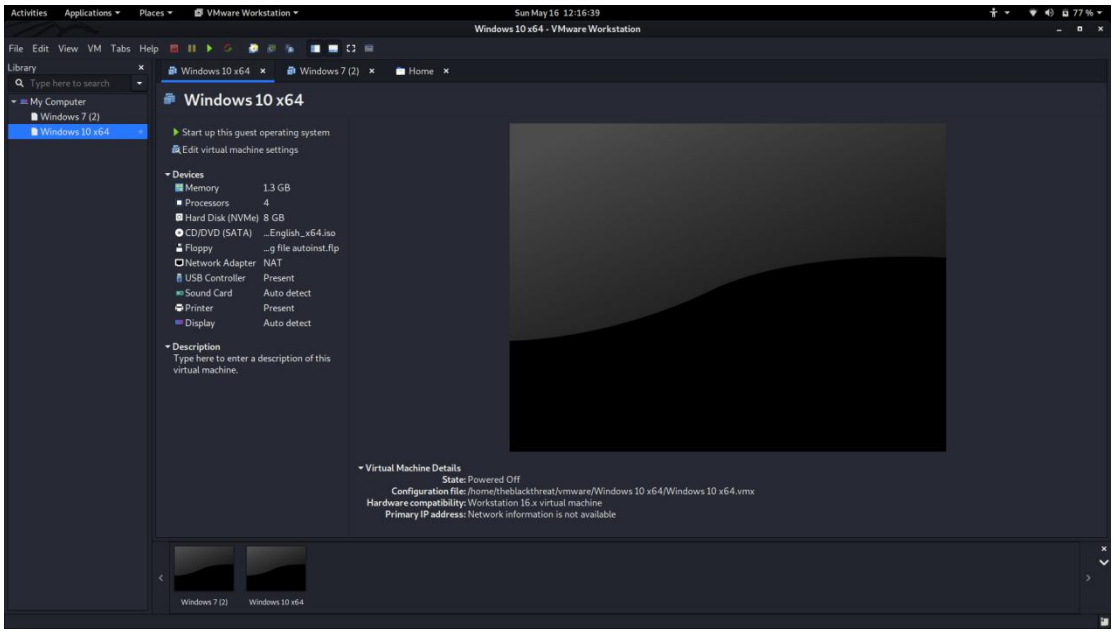
STep 10: after installation enter youre username and password and you will done.



Creating Snapshot on VMware for

snapshot make work more effiecient if there is probability of happening disater to the system. snapshot create a virtual memory and save the machine state so that we can again start our work from their.

I have created windows 10 and 7 in my work station



Go to VM on the top bar then click on create snap shot and save it.

13. Virtualization using VMware Cloud

Virtualization software allows multiple operating systems and applications to run on the same server at the same time, and, as a result, lowers costs and increases efficiency of a company's existing hardware. It's a fundamental technology that powers cloud computing. Virtualization thus emulates hardware.

Types of Virtualization:

1. Hardware Virtualization.
2. Operating system Virtualization.
3. Server Virtualization.
4. Storage Virtualization.

1) Hardware Virtualization:

When the virtual machine software or virtual machine manager (VMM) is directly installed on the hardware system is known as hardware virtualization.

The main job of hypervisor is to control and monitoring the processor, memory and other hardware resources.

After virtualization of hardware system we can install different operating system on it and run different applications on those OS.

Usage:

Hardware virtualization is mainly done for the server platforms, because controlling virtual machines is much easier than controlling a physical server.

2) Operating System Virtualization:

When the virtual machine software or virtual machine manager (VMM) is installed on the Host operating system instead of directly on the hardware system is known as operating system virtualization.

Usage:

Operating System Virtualization is mainly used for testing the applications on different platforms of OS.

3) Server Virtualization:

When the virtual machine software or virtual machine manager (VMM) is directly installed on the Server system is known as server virtualization.

Usage:

Server virtualization is done because a single physical server can be divided into multiple servers on the demand basis and for balancing the load.

4) Storage Virtualization:

Storage virtualization is the process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device.

Storage virtualization is also implemented by using software applications.

Usage:

Storage virtualization is mainly done for back-up and recovery purposes.

How does virtualization work in cloud computing?

Virtualization plays a very important role in the cloud computing technology, normally in the cloud computing, users share the data present in the clouds like application etc, but actually with the help of virtualization users shares the Infrastructure.

The **main usage of Virtualization Technology** is to provide the applications with the standard versions to their cloud users, suppose if the next version of that application is released, then cloud provider has to provide the latest version to their cloud users and practically it is possible because it is more expensive.

To overcome this problem we use basically virtualization technology, By using virtualization, all severs and the software application which are required by other cloud providers are maintained by the third party people, and the cloud providers has to pay the money on monthly or annual basis

Note: Delete service if not needed otherwise aws charge you.
