



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

SN.	Topics	Page No.
01	Creating Amazon EC2 instances with Microsoft Windows	03-04
02	Working with Amazon Elastic Block Store (EBS)	05-07
03	Build Your Virtual Private Cloud (VPC) and Launch a Web Server	08-15
04	Introduction to AWS Identity and Access Management (IAM)	16-19
05	Deploy a Web Application on AWS	20-15
06	Using Auto Scaling with AWS Lambda	26-29
07	Launching EC2 Spot Instances with Auto Scaling and Amazon CloudWatch	30-28
08	Working with Amazon Machine Image	39-40
09	Working with load balancers	41-45
10	Working with EFS	46-48
11	Accessing relational database using Linux machine	49-52
12	VMs using VMware.	53-57
13	Virtualization using VMware Cloud	58-59

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

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with various navigation options like New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and Elastic IPs. The main area displays a table titled 'Instances (1/1) Info' with one row for a 'windows' instance. The instance details are as follows:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z...	Public IPv4 DNS	Public IPv4...	Elastic IP
windows	i-06c9db113678d55a9	Running	t2.micro	-	No alarms	us-east-1a	ec2-3-93-184-170.co...	3.93.184.170	-

Below the table, a detailed view for the 'windows' instance is expanded. It shows the following information:

Instance: i-06c9db113678d55a9 (windows)		
Details		
Instance ID	Public IPv4 address	Private IPv4 addresses
i-06c9db113678d55a9 (windows)	3.93.184.170 open address	172.31.29.42
Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	ec2-3-93-184-170.compute-1.amazonaws.com open address	ip-172-31-29-42.ec2.internal
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-be9110c3
AWS Compute Optimizer finding	IAM Role	Subnet ID
(Outstanding AWS Compute Optimizer recommendations)		

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

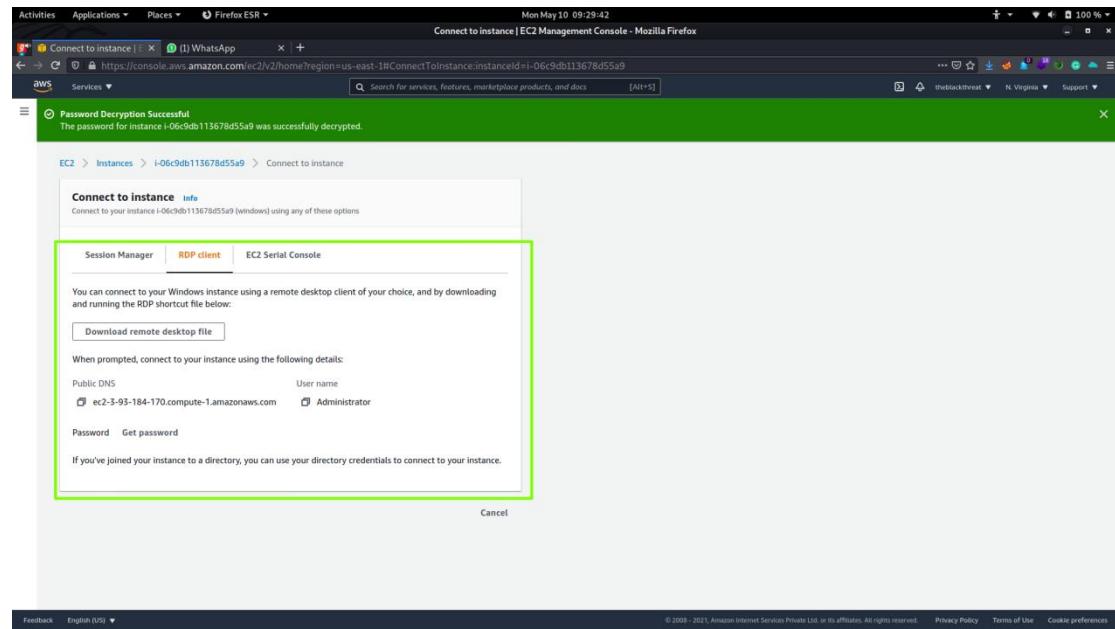
Open RDP client application on windows im using my Linux OS so open

for windows:

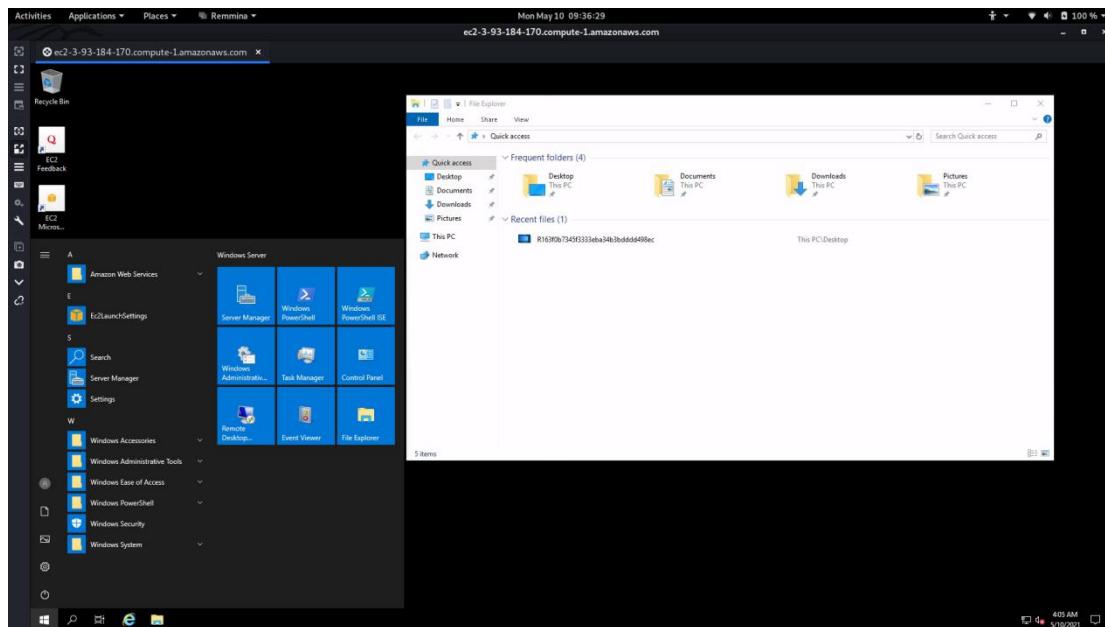
<https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/clients/remote-desktop-clients>

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 volumes are highly reliable and available drive that can be attached with any instance that is in the same availability zone.

All instances have its own volumes but after termination of the instance volumes are also deleted.. but EBS provide a common volumes that can be used with multiple instances and they can share the storage. with an additional volumes.

Volumes: root volumes and additional volumes for instances. Additional volumes concern with EBS.

Snapshots: It is an image to create a backup or used to create a duplicate

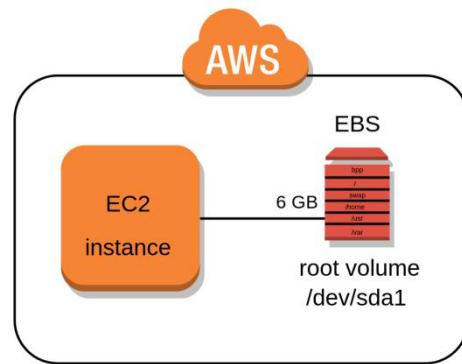
Step to handle it:

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 target it as Hard drive but AWS provide a SSD service for EBS to increase High performance level.)

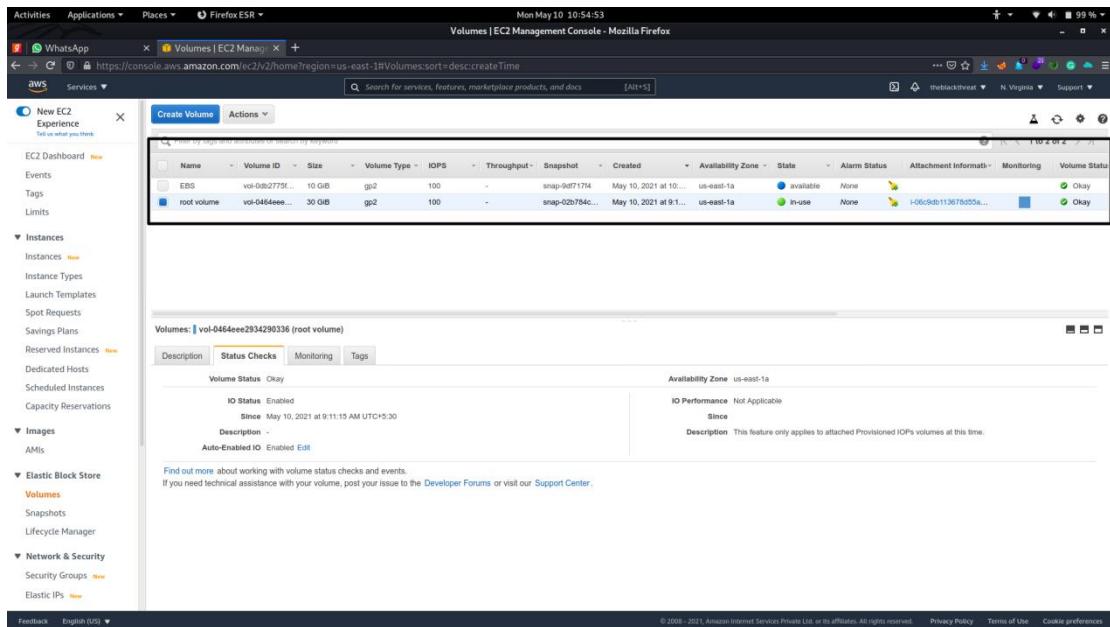
Creation of New EBS volume:

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

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

The screenshot shows the 'Create Volume' wizard in the AWS EC2 Management Console. The form includes fields for Volume Type (General Purpose SSD (gp2)), Size (GB) (10), IOPS (100 / 3000), Availability Zone (us-east-1a), Snapshot ID (snap-0df717f4), Encryption (checked), and Master Key (alias/awsesb). At the bottom, there are KMS Key Description, KMS Key Account, KMS Key ID, and KMS Key ARN fields, along with a note about the default master key.

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



Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status	Attachment Information	Monitoring	Volume Status
EBS	vol-0db2775fe3eabd0f	10 GiB	gp2	100	-	snap-0d717f4...	May 10, 2021 at 9:11:15 AM UTC+5:30	us-east-1a	available	None		Okay	
root volume	vol-0464eee...	30 GiB	gp2	100	-	snap-02b784c...	May 10, 2021 at 9:11:15 AM UTC+5:30	us-east-1a	In-use	None		Okay	

Snapshots: It is important to know that if you have deleted volumes by mistake then you won't be able to recover it so large organization always creates their backups and stores all data and Volumes.

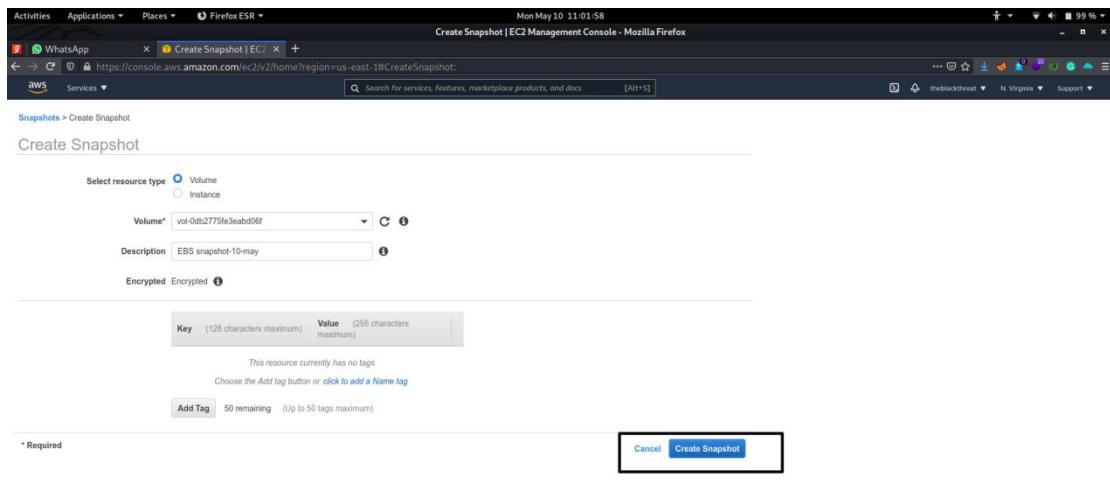
To create snapshot let's take an example :

Step 1: Below Volumes there is an option named Snapshot. Click on it

Step 2: Create snapshot choose your volume

Note: If you want to create snapshot for your instance then click on instance.

Step 3: Select your volume to snap.



It will take some time to complete; you can delete it as well from the action navigation 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.



Requirements : 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.

The screenshot shows the AWS VPC Management Console. The main view displays the details of a VPC named 'vpc-095f798b3ef44a7c7 / Theblackthreat'. The 'Details' tab is selected, showing the following information:

VPC ID	State	DNS hostnames	DNS resolution
vpc-095f798b3ef44a7c7	Available	Disabled	Enabled
Tenancy	DHCP options set	Main route table	Main network ACL
Default	dopt-e75a789d	rtb-0278d1e987bfd2959	acl-06ef0342c4a95a64b
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network border group)
No	10.0.0.0/24	-	-
Route 53 Resolver DNS Firewall rule groups	Owner ID		
-	078454872459		

Below the details, there are tabs for 'CIDRs' (selected), 'Flow logs', and 'Tags'. The 'CIDRs' tab shows one entry:

CIDR	Status
10.0.0.0/24	Associated

The 'REACHABILITY' section shows 'Reachability Analyzer' and 'DNS FIREWALL' status. At the bottom, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

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

VPC ID
Create subnets in this VPC.
vpc-095f798b3ef44a7c7 | Theblackthreat

Associated VPC CIDRs
IPv4 CIDRs
10.0.0.0/24

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
subnet-01

The name can be up to 256 characters long.

Availability Zone info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 CIDR block Info
Q. 0.0.0.0/24

Tags - optional
Key: Q. Name Value - optional: Q. subnet-01 Remove
Add new tag

You can add 49 more tags.

then click on Create Subnet.

New VPC Experience
Tell us what you think

VIRTUAL PRIVATE CLOUD
Your VPCs New
Subnets New
Route Tables New
Internet Gateways New
Egress Only Internet Gateways New
Carrier Gateways New
DHCP Options Sets New
Elastic IPs New
Managed Prefix Lists New
Endpoints
Endpoint Services
NAT Gateways New
Peering Connections

SECURITY
Network ACLs New
Security Groups New

REACHABILITY
Reachability Analyzer

DNS FIREWALL
Rule Groups New

Subnets (1/1) Info

Subnet ID: subnet-0905582b5e4e82744 Clear filters

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses
subnet-01	subnet-0905582b5e4e82744	Available	vpc-095f798b3ef44a7c7 Theblackthreat	10.0.0.0/24	-	251

subnet-0905582b5e4e82744 / subnet-01

Details Flow logs Route table Network ACL Sharing Tags

Details

Subnet ID subnet-0905582b5e4e82744	State Available	VPC vpc-095f798b3ef44a7c7 Theblackthreat	IPv4 CIDR 10.0.0.0/24
Available IPv4 addresses 251	IPv6 CIDR -	Availability Zone us-east-1a	Availability Zone ID use1-az4
Network border group us-east-1	Route table rtb-0278d1a987bfd2959	Network ACL acl-06ef0342c495a64b	Default subnet No

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

Route tables (1/2) Info

Filter route tables

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
theblackthreat-vpc	rtb-0278d1a987bfd2959	-	-	Yes	vpc-095f798b3ef44a7c7 Theblackthreat	078434872459
	rtb-d71a16a9	-	-	Yes	vpc-be9110c3	078434872459

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="my-route-01"/>
Remove	
Add new tag	

You can add 49 more tags.

[Cancel](#) [Create route table](#)

Route table rtb-0b0f740349b4e166e | my-route-01 was created successfully.

rtb-0b0f740349b4e166e / my-route-01

Details Info

Route table ID <input type="text" value="rtb-0b0f740349b4e166e"/>	Main <input checked="" type="radio"/> No	Explicit subnet associations -	Edge associations -
VPC <input type="text" value="vpc-095f798b3ef44a7c7 Theblackthreat"/>	Owner ID <input type="text" value="078434872459"/>	Actions	

[Routes](#) [Subnet associations](#) [Edge associations](#) [Route propagation](#) [Tags](#)

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/24	local	Active	No

[Edit routes](#)

Now we have successfully set up with route table

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

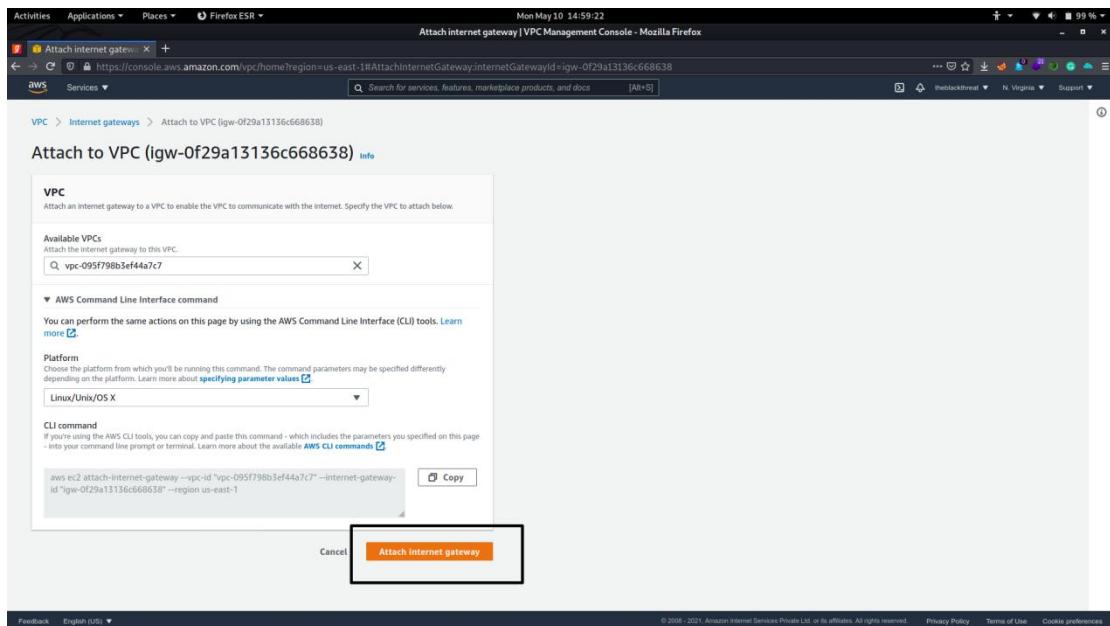
An internet gateway is a horizontally scaled, redundant, and highly available VPC component that enables communication between your VPC and the internet.

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 gateway. 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 gateway.(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

Name	Value	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count
acl-f5ca2289	6 Subnets	Yes	vpc-be9110c3	2 Inbound rules	2 Outbound rules	
acl-06ef0342c4a95a...	subnet-0905582b5e4e82744 / s...	Yes	vpc-095f798b3ef44a7c7 / Theblackthreat	2 Inbound rules	2 Outbound rules	

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.

The screenshot shows the AWS VPC Management Console in Mozilla Firefox. A green success message at the top states: "Security group (sg-00be4354a8aa69657 | theblackthreatSecurityGroup) was created successfully". Below this, the security group details are shown: Security group name is "theblackthreatSecurityGroup", Security group ID is "sg-00be4354a8aa69657", Description is "allow ssh to developers", and VPC ID is "vpc-095f798b3ef44a7c7". The Owner is listed as "078434872459". Inbound rules count is 1 Permission entry, and Outbound rules count is 2 Permission entries. The "Inbound rules" tab is selected, showing one rule: Type is HTTPS, Protocol is TCP, Port range is 443, Source is 0.0.0.0/0, and Description - optional is "HTTPS-Own-system". The left sidebar shows various AWS services like VPC, Route Tables, Subnets, and more under the SECURITY section.

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

Creating A Windows IIS-Webserver on AWS



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 Protocol) 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.

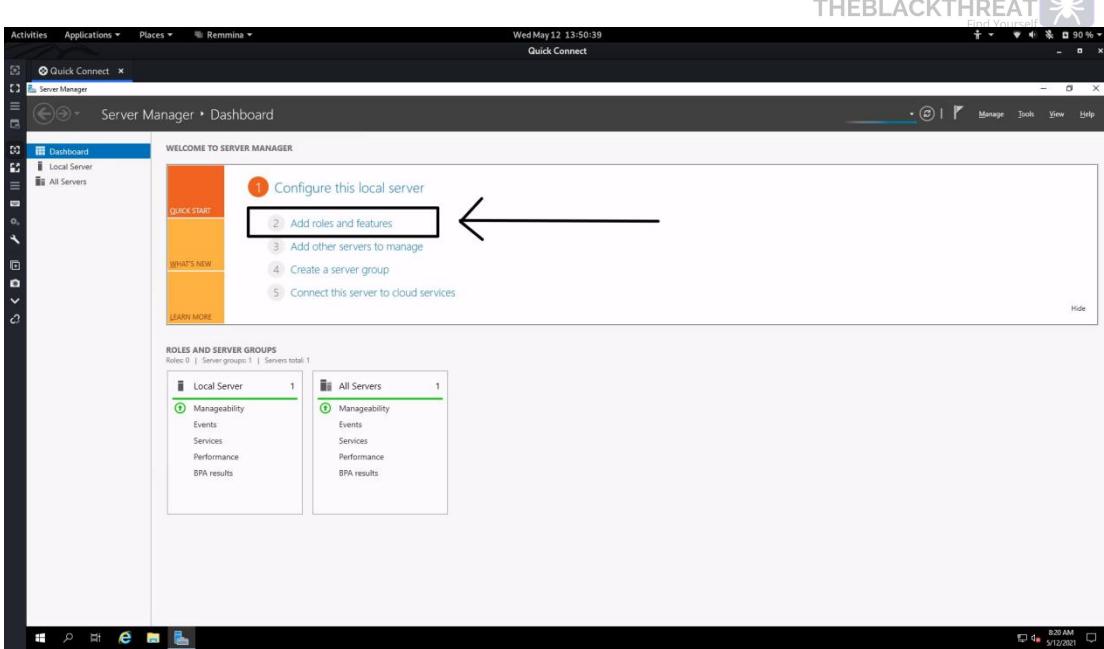
Troubleshoot to connection:

(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.)

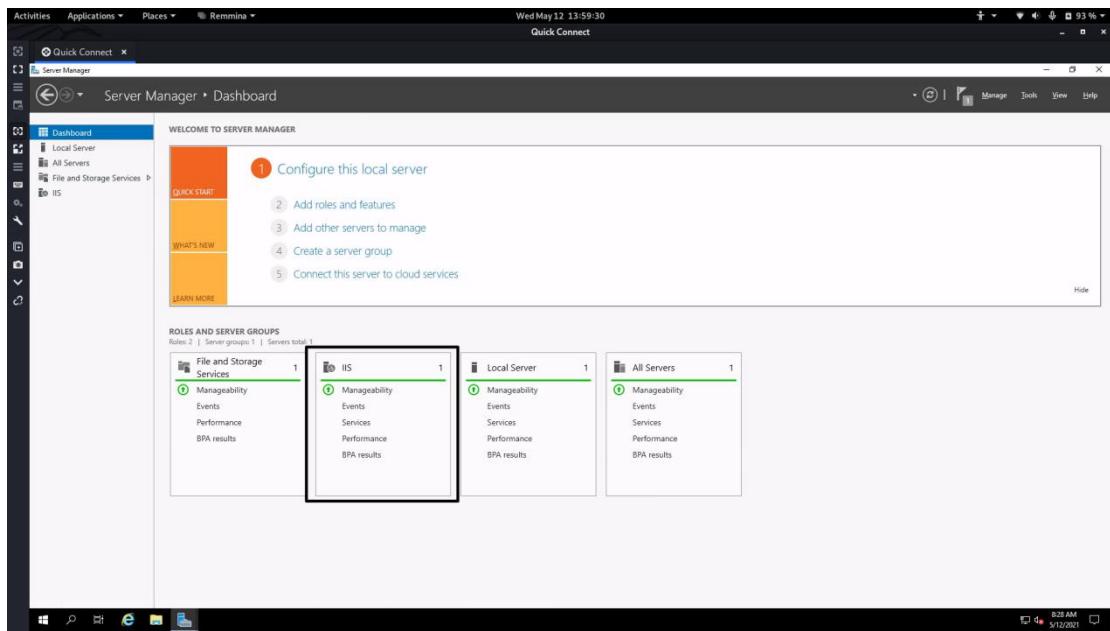
Step 8: After your instance is running then set for your webserver.

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.

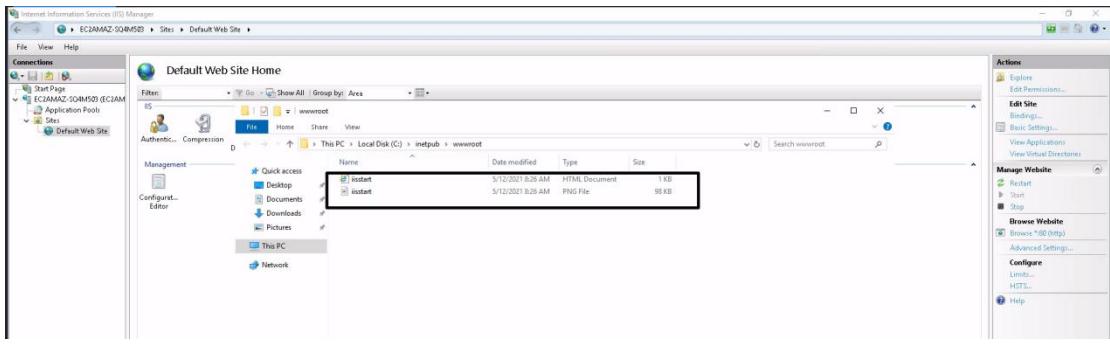
after completing the instalation process we it will look like this.



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

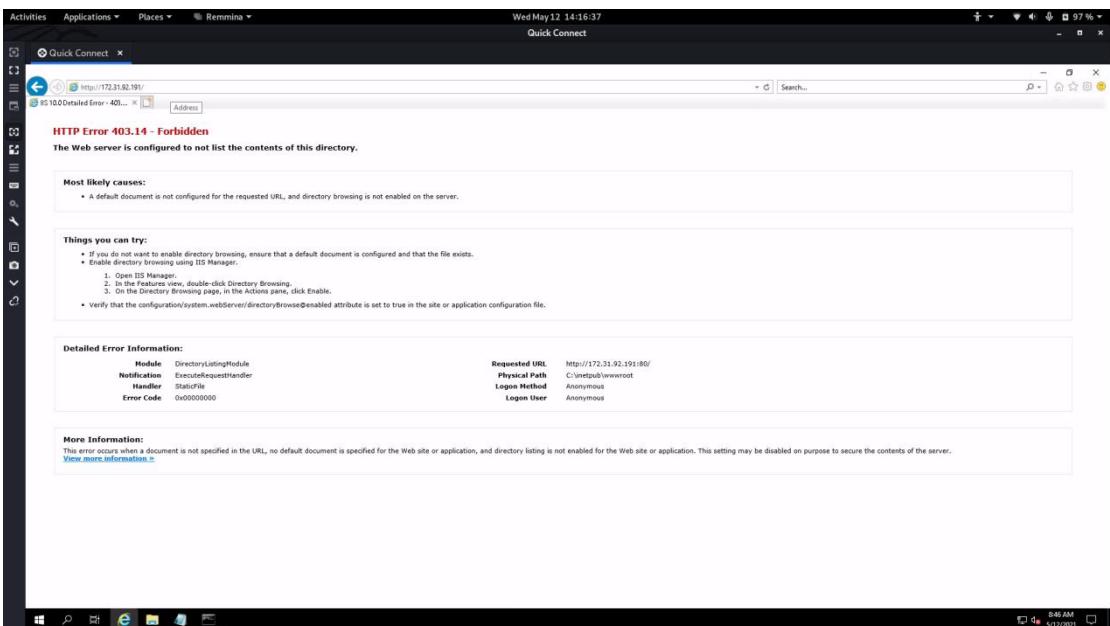
Step 10: click on Default webpage that is on left navigation bar you can find more services you can choose so .

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



I have deleted the default file and created my own.

it will run on my machine.



It will show that there is an error because we didn't give our webpage name. this will shows that our site security is running properly.



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



Requirements : 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 resources securely 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 their 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 Panel you can see the Users, groups, Role click on Users.

The screenshot shows the AWS IAM Management Console. In the top-left corner, there's a navigation bar with links for Activities, Applications, Places, and FireFox ESR. Below that is the AWS logo and a Services dropdown. The main content area has a title 'IAM Management Console - Mozilla Firefox'. At the top right, there are tabs for 'theblackthreat', 'Global', and 'Support'. A search bar at the top right contains the placeholder 'Search for services, features, marketplace products, and docs [Alt+S]'. The left sidebar is titled 'Identity and Access Management (IAM)' and includes sections for Dashboard, Access management, User groups, and **Users**. The 'Users' section is highlighted with a red box and has an orange arrow pointing to it from the text above. Below the sidebar, there's a table header with columns for 'User name', 'Groups', 'Access key age', 'Password age', 'Last activity', and 'MFA'. A message at the bottom of the table says 'Showing 0 results' and 'There are no IAM users. Learn more'. At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

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.

The screenshot shows the 'Add user' wizard, step 2: Set permissions. At the top, there are five numbered steps: 1, 2, 3, 4, 5. Step 2 is highlighted with a blue circle. The main area is titled 'Set permissions' and contains three buttons: 'Add user to group' (highlighted with a red box), 'Copy permissions from existing user', and 'Attach existing policies directly'. Below these buttons is a note: 'Get started with groups' followed by the text 'You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. Learn more' and a 'Create group' button. At the bottom, there's a link 'Set permissions boundary'.

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 important 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

Add user

Review

User details

User name	theblackthreat
AWS access type	Programmatic access and AWS Management Console access
Console password type	Custom
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonEC2FullAccess
Managed policy	AmazonDynamoDBFullAccess
Managed policy	IAMUserChangePassword

Tags

No tags were added.



Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://078434872459.signin.aws.amazon.com/console>

Download.csv

User	Access key ID	Secret access key	Email login instructions
theblackthreat-01	AKIAREQYVFSFZ28HRFPJ	***** Show	Send email

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

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

Login as IAM.

We have privileges to create Compute machines, VPC's and Create databases.

Lets create Instance using IAM user.

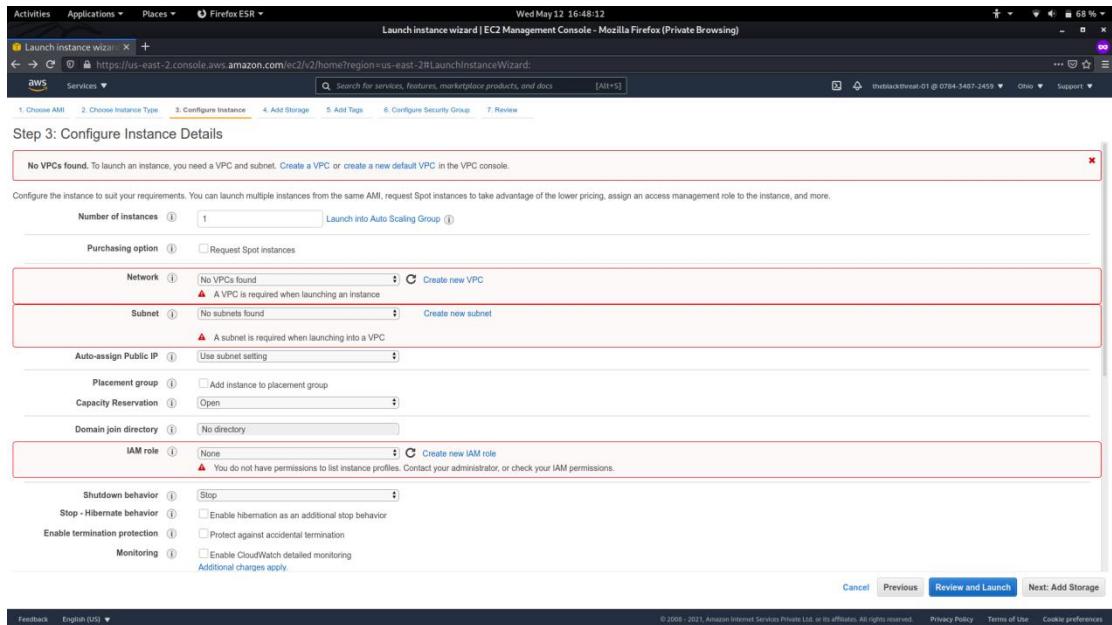
Step 1: GO to EC2 ---> select your instance machine and then choose your type of Instance.
 Im using my free tier Instances.

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.



Step 3: Configure Instance Details

No VPCs found. To launch an instance, you need a VPC and subnet. [Create a VPC](#) or [create a new default VPC](#) in the VPC console.

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option: Request Spot instances

Network: No VPCs found [Create new VPC](#)
 A VPC is required when launching an instance

Subnet: No subnets found [Create new subnet](#)
 A subnet is required when launching into a VPC

Auto-assign Public IP: Use subnet setting

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory

IAM role: None [Create new IAM role](#)
 You do not have permissions to list instance profiles. Contact your administrator, or check your IAM permissions.

Shutdown behavior: Stop Enable hibernation as an additional stop behavior

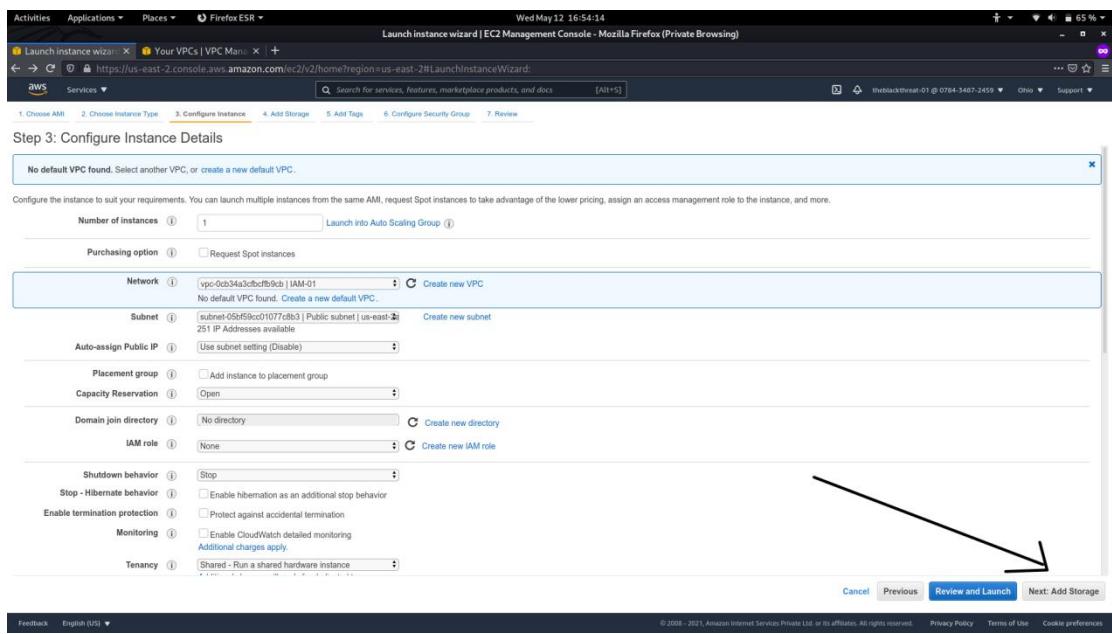
Stop - Hibernate behavior:

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring
 Additional charges apply.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

create VPC and then click to further.



Step 3: Configure Instance Details

No default VPC found. Select another VPC, or [create a new default VPC](#).

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option: Request Spot instances

Network: vpc-0cd34a3fbcbf9dc | IAM-01 [Create new VPC](#)
 No default VPC found. [Create a new default VPC](#).

Subnet: subnet-05f9fc01077c8b3 | Public subnet | us-east-2 [Create new subnet](#)
 251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory [Create new directory](#)

IAM role: None [Create new IAM role](#)

Shutdown behavior: Stop Enable hibernation as an additional stop behavior

Stop - Hibernate behavior:

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring
 Additional charges apply.

Tenancy: Shared - Run a shared hardware instance

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

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

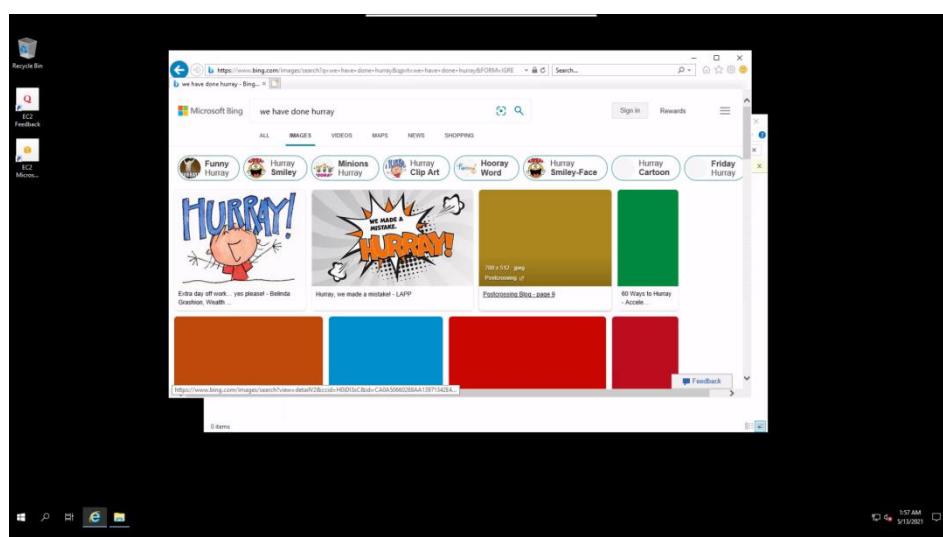
The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like Events, Tags, Limits, and Network & Security. The main area displays two instances: Ubuntu-IAM-01 (Stopped) and Window-IAM-01 (Running). The instance details for Window-IAM-01 are expanded, showing its Public IPv4 address (18.116.130.179), Private IPv4 address (10.0.0.227), and VPC ID (vpc-0cb54a3cfbcffbb5cb). Other details include its Public IPv4 DNS (ec2-18-116-130-179.us-east-2.compute.amazonaws.com), Elastic IP (3.131.62.91), and Availability Zone (us-east-2a).

Connecting this machine via RDp client.

Step 5: Download RDP client file and dycrypt your password. and connect it to your host.

The screenshot shows the 'Connect to instance' page for the Windows-IAM-01 instance. It provides options for Session Manager, RDP client, and EC2 Serial Console. The RDP client tab is selected. It shows session details like Public DNS (ec2-18-116-130-179.us-east-2.compute.amazonaws.com) and Administrator. A pink arrow points from the 'Administrator' link to a certificate acceptance dialog box titled 'Windows-IAM-01'. The dialog box shows 'Accept certificate?' with 'Yes' and 'No' buttons, and contains certificate details: Subject: CN = EC2AMAZ-OROL8N6, Issuer: CN = EC2AMAZ-OROL8N6, Fingerprint: 03:58:08:e7:1a:6a:0a:8d:0a:56:e7:b3:10:fb:e4:1a:46:28:b4:ad:18:de:3e:b0:c3:55:68:00:5e:15:e1:3d.

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



Requirements : 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 Interactive 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.

Version label	Description	Date created	Source	Deployed to
Sample Application-1		2021-05-13T09:08:42+05:30	20211133CVq-pro_teal_756.zip	Theblackthreattestwebapp-env
Sample Application		2021-05-13T08:46:23+05:30	Sample Application	-

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

Environment name	Health	Application name	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
Theblackthreattestwebapp-env	Warning	theblackthreat-test-webapp	2021-05-13 08:46:25 UTC+0530	2021-05-13 09:09:48 UTC+0530	Theblackthreattestwebapp-env.eba-czup5fx8.us-east-1.elasticbeanstalk.com	Sample Application-1	PHP 8.0 running on 64bit Amazon Linux 2	Supported	WebServer

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

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with navigation links like 'Activities', 'Applications', 'Places', and 'Services'. Under 'Services', 'Elastic Beanstalk' is selected. In the main content area, the application 'Theblackthreattestwebapp-env' is highlighted. The URL 'Theblackthreattestwebapp-env.eba-cupzfx8.us-east-1.elasticbeanstalk.com' is shown with its environment ID 'e-5sd7nzaayl'. The application name is 'theblackthreat-test-webapp'. The health status is 'Warning' (orange exclamation mark icon). The running version is 'Sample Application-1' (with a 'Upload and deploy' button). The platform is 'php' (with a PHP icon). Below this, a table lists recent events with columns for Time, Type, and Details. The details for each event are as follows:

Time	Type	Details
2021-05-13 09:10:16 UTC-0530	WARN	Service role 'arn:aws:iam::078434872459:role/aws-elasticbeanstalk-service-role' is missing permissions required to check for managed updates. Verify the role's policies.
2021-05-13 09:09:48 UTC-0530	INFO	Environment update completed successfully.
2021-05-13 09:09:48 UTC-0530	INFO	New application version was deployed to running EC2 instances.
2021-05-13 09:09:28 UTC-0530	INFO	Instance deployment completed successfully.
2021-05-13 09:09:26 UTC-0530	INFO	Instance deployment: You didn't include a 'composer.json' file in your source bundle. The deployment didn't install Composer dependencies.

At the bottom, there are links for 'Feedback', 'English (US)', and 'Cookie preferences'.

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

The screenshot shows the 'Pro Teal' free website template. The header has a dark teal background with the title 'Pro Teal' and a search bar. The main content area has a white background. It includes several sections: 'FOLLOW US on Twitter' (with a blue bird icon), 'SUBSCRIBE our feed' (with an orange RSS icon), 'Featured Work' (with a small image of a garden), 'Promotion' (with a red ribbon icon), and 'What we do?' (with a green checkmark icon). The 'What we do?' section contains text about the template being a full-site (5 pages) website template provided by templatemo.com and credits to FreePhotos for photos and SmileyWallpaper.com for icons used in the template. There are also 'details' buttons for each section.

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 Database.
 - 3.create elastic beanstalk.

Creating RDS database service.

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

The screenshot shows the 'Create database' wizard in the AWS RDS console. The 'Standard create' method is selected. Under 'Engine type', MySQL is chosen. Other options like Amazon Aurora, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server are also listed. The MySQL section includes a note about known issues and limitations. At the bottom, there are links for feedback, English (US), and cookie preferences.

- 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.

The screenshot shows the 'Databases' page in the AWS RDS console. A success message 'Successfully created database freelancedb' is displayed. The database table lists 'freelancedb' with details: Instance: MySQL Community, Region & AZ: us-east-1a, Size: db.t2.micro, Status: Available, Current activity: none, Maintenance: none, and VPC: vpc-...'. The left sidebar shows navigation options like Dashboard, Databases, Query Editor, etc.

Creating Application on Elastic Beanstalks



Step 1: First Create an Instance or just go through the Elastic Beanstalk. AWS automatic 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 application and upload it on Elastic beanstalks. and then select for PHP

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

The screenshot shows the AWS Elastic Beanstalk Applications console in a Firefox browser. The URL is https://console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/applications. The left sidebar shows 'Recent environments' with 'Freelance-env' and 'Theblackthreat-test-webapp-env'. The main area displays a table titled 'All applications' with columns: Application name, Environments, Date created, Last modified, and ARN. Two rows are listed:

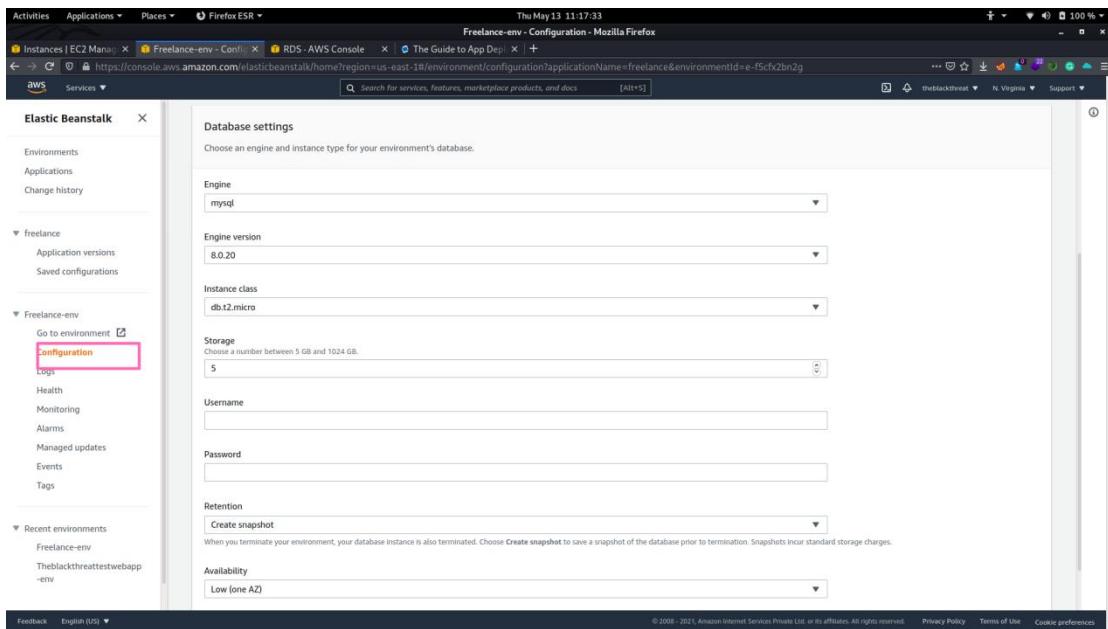
Application name	Environments	Date created	Last modified	ARN
freelance	Freelance-env	2021-05-13 11:01:55 UTC+0530	2021-05-13 11:01:55 UTC+0530	arn:aws:elasticbeanstalk:us-east-1:078434872459:application/freelance
theblackthreat-test-webapp	Theblackthreat-test-webapp-env	2021-05-13 08:43:46 UTC+0530	2021-05-13 08:43:46 UTC+0530	arn:aws:elasticbeanstalk:us-east-1:078434872459:application/theblackthreat-test-webapp

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

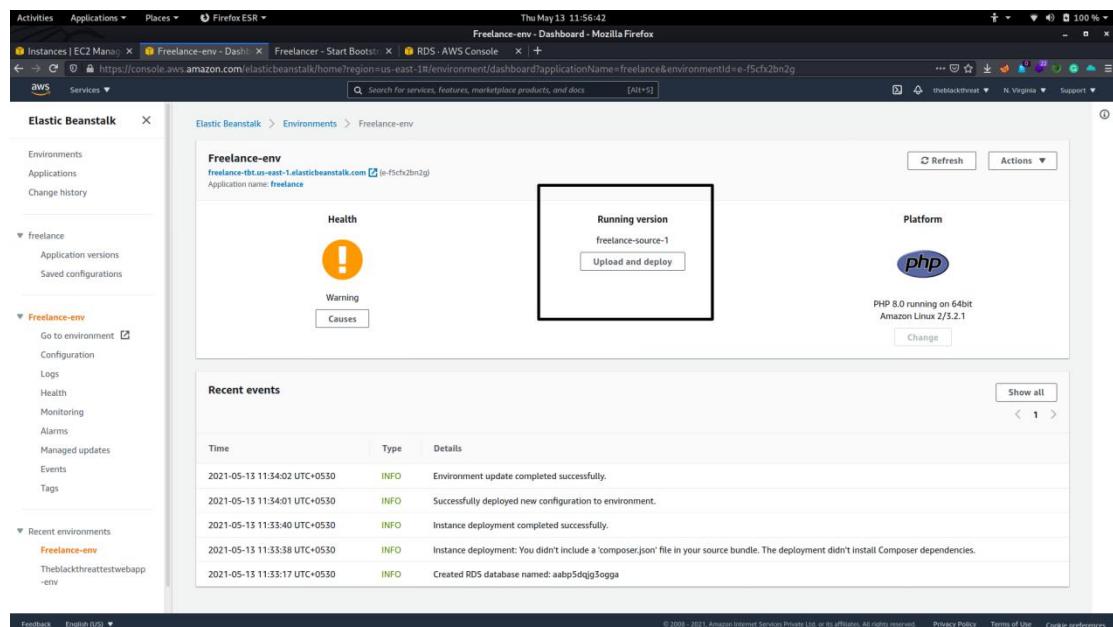
Click on Configuration To create another security group in the environment, you should do the following:

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”

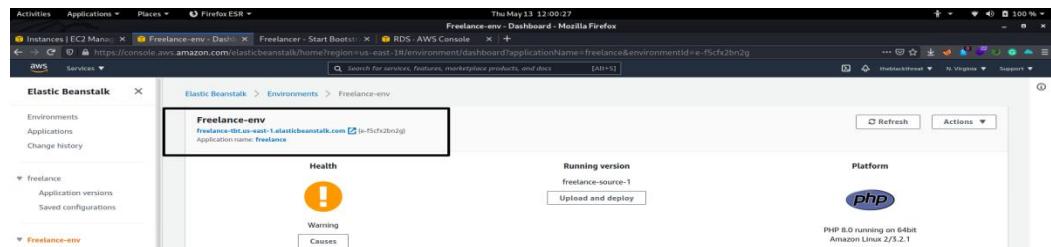
Syep 4: To connect to the database scroll down and click on Database---Edit
It will fetch you RDS database and then we can set up the credentials. put your username and Password form RDS database which you have created before. ----> Apply.

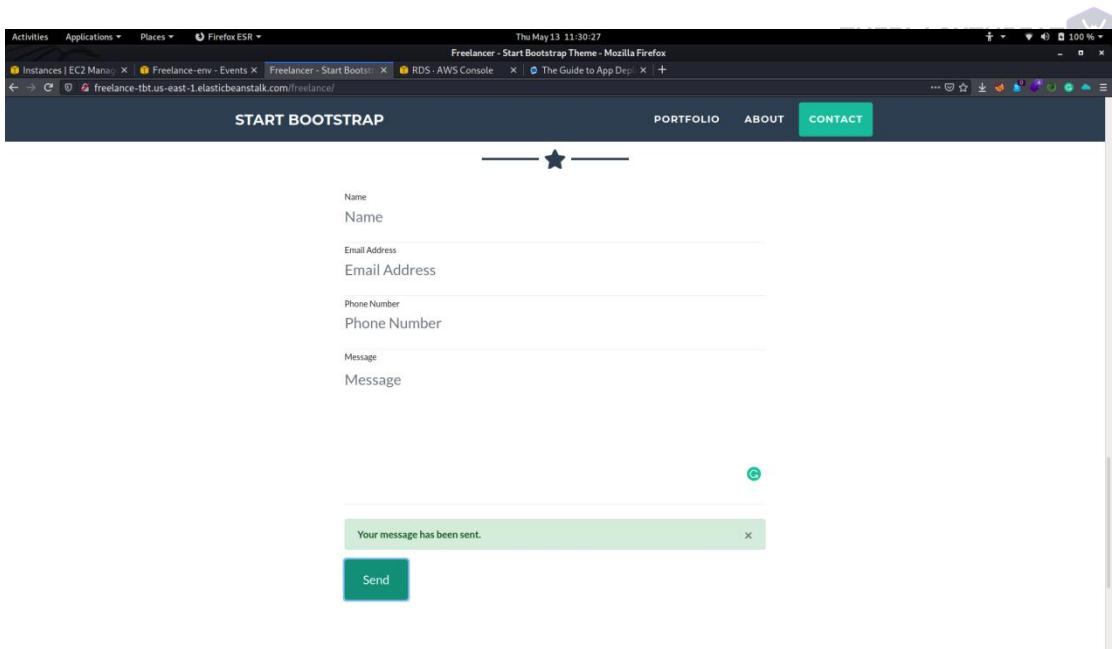


Step 5: After configuration of Database and Software modification save it and go to the application select your wep applicatio and then upload you sample.zip file here. and click on deploy,



Wait for some time and after a while your application is able to run and is automatially 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.





-----000-----

6. Objective: Using Auto Scaling with AWS Lambda



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

Lambda : AWS provide a serverless service named Lambda that runs our code and manages the triggered services for us using the resources. It is managed by the services parallelly and manages scaling. We can create lambda functions to manage concurrent traffic. We can use our own logic to handle the services by our own methods.

Creating a lambda function:

Our Aim : first we need to create an S3 bucket.

then we create lambda function

provide lambda 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 an S3 bucket named “**s3withlambda-tbt**” leave all options as defaults ----> create bucket

The screenshot shows the AWS S3 Management Console. A green success message at the top says "Successfully created bucket 's3withlambda-tbt'. To upload files and folders, or to configure additional bucket settings choose View details." On the left sidebar, under the "Buckets" section, there is a table with one row for the newly created bucket "s3withlambda-tbt". The table columns include Name, AWS Region, Access, and Creation date. The bucket details are: Name - s3withlambda-tbt, AWS Region - US East (N. Virginia) us-east-1, Access - Bucket and objects not public, and Creation date - May 18, 2021, 08:23:20 (UTC+05:30).

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.

The screenshot shows the "Create role" wizard in the AWS IAM console, specifically Step 4: Review. The role name is "lambdas3policy". The role description is "Allows Lambda functions to call AWS services on your behalf." Under "Policies", three policies are selected: "AWSLambda_FullAccess", "CloudWatchFullAccess", and "AmazonS3FullAccess". The "Permissions boundary" is noted as "Permissions boundary is not set".

Step 3: Click on Create role.

The screenshot shows the AWS IAM Roles page. A success message at the top says "The role lambdas3policy has been created." Below it, a table lists various roles. The newly created role, "lambdas3policy", is highlighted with a black border. Its details are as follows:

Role name	Trusted entities	Last activity
AWSDataLifecycleManagerDefaultRole	AWS service: dlm	None
AWSServiceRoleForAmazonElasticFileSystem	AWS service: elasticfilesystem (Service-Linked role)	Yesterday
AWSServiceRoleForAutoScaling	AWS service: autoscaling (Service-Linked role)	3 days
AWSServiceRoleForBackup	AWS service: backup (Service-Linked role)	Today
AWSServiceRoleForElastiCacheBalancing	AWS service: elasticloadbalancing (Service-Linked role)	Yesterday
AWSServiceRoleForGlobalAccelerator	AWS service: globalaccelerator (Service-Linked role)	None
AWSServiceRoleForRDS	AWS service: rds (Service-Linked role)	Yesterday
AWSServiceRoleForSupport	AWS service: support (Service-Linked role)	None
AWSServiceRoleForTrusted Advisor	AWS service: trustedadvisor (Service-Linked role)	None
lambdas3policy	AWS service: lambda	None

We have created 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 “**workinglambda3-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.

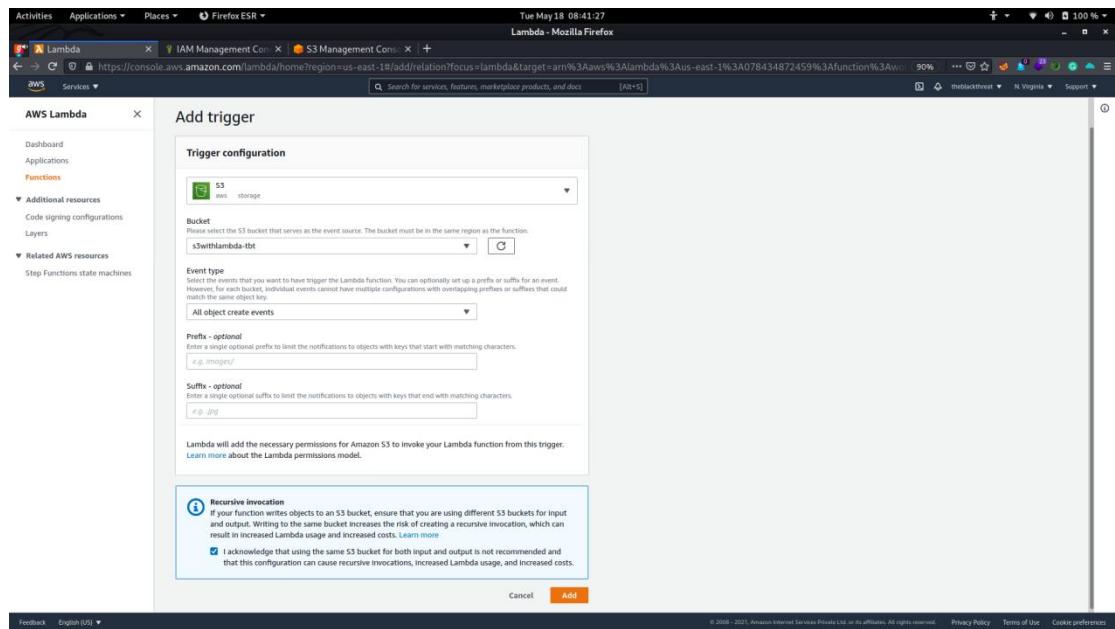
```
exports.handler = function(event, context, callback) {
    console.log("Incoming Event: ", event);
    const bucket = event.Records[0].s3.bucket.name;
    const filename = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
    const message = `File is uploaded in - ${bucket}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

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

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

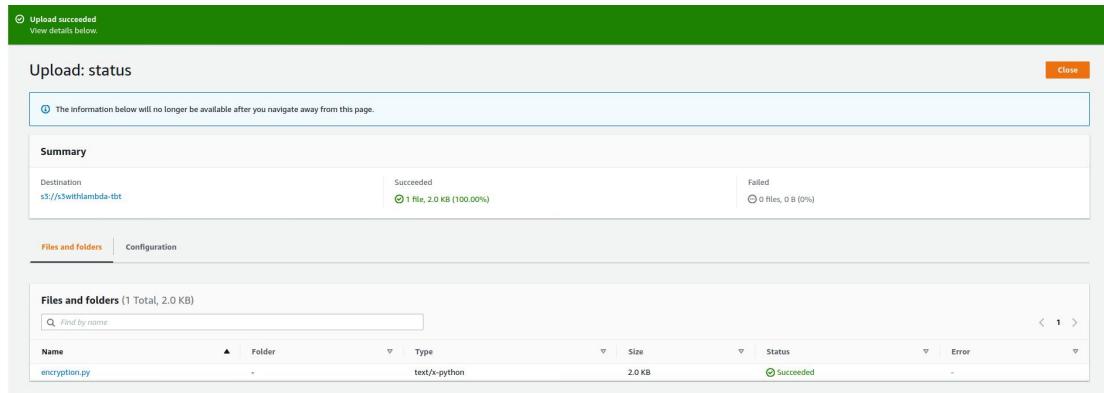
The screenshot shows the AWS Lambda Function Overview page for the function "workinglambda3-tbt". The function ARN is listed as "arn:aws:lambda:us-east-1:078454872459:function:workinglambda3-tbt". The "Code source" tab is selected, showing the "index.js" file with the following code:

```
1 exports.handler = function(event, context, callback) {
2     console.log("Incoming Event: ", event);
3     const bucket = event.Records[0].s3.bucket.name;
4     const filename = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
5     const message = `File is uploaded in - ${bucket} -> ${filename}`;
6     console.log(message);
7     callback(null, message);
8 };
```

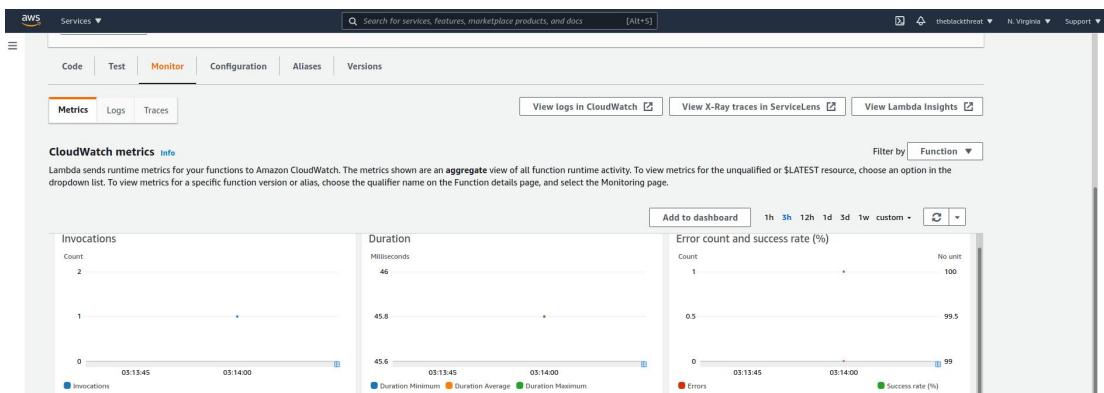


Now We have done all things lets check wheather its working or not.

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

The screenshot shows the AWS CloudWatch Management Console interface. On the left, there's a navigation sidebar with various services like CloudWatch, IAM, and S3. The main area is titled 'Log events' and displays a list of log entries. The first few entries are:

- 2021-05-18T08:44:53.393+05:30 START RequestId: b3348b52-4e0d-44a1-a45d-5b6efe03c20d Version: \${LATEST}
- 2021-05-18T08:44:53.419+05:30 2021-05-18T03:14:53.419Z b3348b52-4e0d-44a1-a45d-5b6efe03c20d INFO Incoming Event: { Records: [{ eventVersion: '2.1', eventName: 'aws:s3:ObjectCreated:Put', eventSource: 'aws:s3', awsRegion: 'us-east-1', s3: { bucket: 's3withlambda-tbt', key: 'encryption.py' } }] }
- 2021-05-18T08:44:53.419+05:30 END RequestId: b3348b52-4e0d-44a1-a45d-5b6efe03c20d
- 2021-05-18T08:44:53.439+05:30 REPORT RequestId: b3348b52-4e0d-44a1-a45d-5b6efe03c20d Duration: 45.80 ms Billed Duration: 46 ms Memory Size: 128 MB Max Memory Used: 65 MB Init Duration: 132.96 ms
- 2021-05-18T08:44:53.439+05:30 No newer events at this moment. Auto retry paused. Resume

Now find your file that name that you have uploaded. 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

The screenshot shows the AWS CloudWatch Management Console interface. On the left, there's a navigation sidebar with various services like CloudWatch, IAM, and S3. The main area is titled 'Log events' and displays a list of log entries. The first few entries are:

- 2021-05-18T08:44:53.393+05:30 START RequestId: b3348b52-4e0d-44a1-a45d-5b6efe03c20d Version: \${LATEST}
- 2021-05-18T08:44:53.419+05:30 2021-05-18T03:14:53.419Z b3348b52-4e0d-44a1-a45d-5b6efe03c20d INFO File is uploaded in - s3withlambda-tbt -> encryption.py
- 2021-05-18T08:44:53.419Z b3348b52-4e0d-44a1-a45d-5b6efe03c20d INFO File is uploaded in - s3withlambda-tbt -> encryption.py
- 2021-05-18T08:44:53.419+05:30 END RequestId: b3348b52-4e0d-44a1-a45d-5b6efe03c20d
- 2021-05-18T08:44:53.439+05:30 REPORT RequestId: b3348b52-4e0d-44a1-a45d-5b6efe03c20d Duration: 45.80 ms Billed Duration: 46 ms Memory Size: 128 MB Max Memory Used: 65 MB Init Duration: 132.96 ms
- 2021-05-18T08:44:53.439+05:30 No newer events at this moment. Auto retry paused. Resume

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 client 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 scaling 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

The screenshot shows the 'Create Load Balancer' wizard on the AWS Management Console. The current step is 'Step 1: Configure Load Balancer'. The 'Basic Configuration' section includes fields for 'Name' (MyLoaderBalancer), 'Scheme' (set to 'internet-facing'), and 'IP address type' (set to 'IPv4'). Below this, the 'Listeners' section shows a single listener for 'HTTP' on port 80. The 'Availability Zones' section lists two zones: 'us-east-1a' and 'us-east-1b', each associated with a specific subnet. At the bottom, there are 'Cancel' and 'Next: Configure Security Settings' buttons.

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.

Fri May 14 09:21:34 Create Load Balancer | EC2 Management Console - Mozilla Firefox

Using Auto Scaling w... | S3 Management Cons... | How to use EC2 Auto ... | Create Load Balancer | EC2 Management Con... +

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1&V2CreateELBWizard.type=application

aws Services Search for services, features, marketplace products, and docs [Alt+5]

theblackthreat N. Virginia Support

1. Configure Load Balancer 2. Configure Security Groups 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 4: Configure Routing
Your load balancer routes requests to the targets in this target group using the protocol and port that you specify here. It also performs health checks on the targets using these settings. The target group you specify in this step will apply to all of the listeners configured on this load balancer. You can edit or add listeners after the load balancer is created.

Target group

Name: MyTargetGroup
Target type: Instance
Protocol: HTTP
Port: 80
Protocol version: HTTP1.1
Health checks: Protocol: HTTP, Path: /
Tags: Advanced health check settings

Cancel Previous Next: Register Targets

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

Fri May 14 09:22:09 Create Load Balancer | EC2 Management Console - Mozilla Firefox

Using Auto Scaling w... | S3 Management Cons... | How to use EC2 Auto ... | Create Load Balancer | EC2 Management Con... +

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1&V2CreateELBWizard.type=application

aws Services Search for services, features, marketplace products, and docs [Alt+5]

theblackthreat N. Virginia Support

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 6: Review
Please review the load balancer details before continuing

Load balancer

Name: MyLoaderBalancer
Scheme: internet-facing
Listeners: Port 80 - Protocol: HTTP
IP address type: ipv4
VPC: vpc-009ec44ef41f571bf
Subnets: subnet-0aee8556e5603909, subnet-026bebbaed7ea59, subnet-0070418681209409, subnet-09d5678e7c968711, subnet-0f1181fd636fea59, subnet-07ed918945369bc9
Tags:

Security groups

Security groups: load-balancer-wizard-1

Routing

Target group: New target group
Target group name: MyTargetGroup
Port: 80
Target type: instance
Protocol: HTTP
Protocol version: HTTP1.1
Health check protocol: HTTP
Path:
Health check port: traffic port
Healthy threshold: 5
Unhealthy threshold: 2
Timeout: 5
Interval: 30
Success codes: 200

Targets

Instances:

Cancel Previous Create

Create Load Balancer Actions

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
awseb-AWSEB-IKKAVNR4D...	awseb-AWSEB-IKKAVNR4D...	active	vpc-009ec44ef41f571bf	us-east-1c, us-east-1b, ...	application	May 13, 2021 at 11:02:29 A...	
MyLoaderBalancer	MyLoaderBalancer-2383084...	provisioning	vpc-009ec44ef41f571bf	us-east-1c, us-east-1b, ...	application	May 14, 2021 at 9:22:11 AM ...	

Load balancer: MyLoaderBalancer

Description Listeners Monitoring Integrated services Tags

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 will create later if needed.)

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

Create launch configuration [Info](#)

Launch configuration name

Name
Auto scaling-EC2

Amazon machine image (AMI) [Info](#)

AMI
Cortex3_20210413_1618302222-7893c21e-ae95-4a33-8767-6feb7734f8dd

Instance type [Info](#)

Instance type
a1.medium (1 vCPUs, 2 GiB, EBS Only) [Choose instance type](#)

Additional configuration - optional

Purchasing option [Info](#)
 Request Spot Instances

Current price

- us-east-1a: \$0.008400
- us-east-1b: \$0.008400
- us-east-1d: \$0.008400

Maximum price (per instance/hour)
\$ 1

IAM instance profile [Info](#)
aws-elasticbeanstalk-ec2-role

Monitoring [Info](#)
 Enable EC2 instance detailed monitoring within CloudWatch

EBS-optimized instance
 Launch as EBS-optimized instance

► Advanced details

ⓘ Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

i Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

Storage (volumes) Info

EBS volumes

Volume type	Devices	Snapshot	Size (GiB)	Volume type
Root	/dev/sda1	snap-02420cd3d2dea1bc0	8	General purpose (SSD)
<input checked="" type="checkbox"/> EBS	/dev/sdh	No snapshot	30	General purpose (SSD)
<input type="checkbox"/> EBS	/dev/sdi	No snapshot	20	General purpose (SSD)

+ Add new volume

i Free tier eligible customers can get up to 30 GB of EBS storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Security groups Info

Assign a security group

- Create a new security group
 Select an existing security group

Security group name

AutoScaling-Security-Group-2

Description

AutoScaling-Security-Group-2 (2021-05-14T05:01:16.024Z)

Rules

Type	Protocol	Port range	Source type	Source
<input type="checkbox"/> Custom TCP rule	TCP	0	Custom IP	0.0.0.0/0
<input type="checkbox"/> All TCP	TCP	0 - 65535	Anywhere	0.0.0.0/0

+ Add new rule

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

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 **Autoscaling group**.

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

- Step 1 >> Choose launch template or configuration

The screenshot shows the 'Create Auto Scaling group' wizard at Step 1. The 'Name' field is filled with 'Auto-scaling-group'. The 'Launch configuration' dropdown is set to 'auto-scaling-EC2'. The 'Next' button is highlighted with a red box.

- Step 2 >> Configure settings

The screenshot shows the 'Configure settings' step at Step 2. The 'VPC' dropdown is set to 'vpc-Q09ec44ef41f571bf'. The 'Subnets' section lists several subnets, with checkboxes checked for 'us-east-1a | subnet-03eef856d5603f909', 'us-east-1b | subnet-0286ebbadec67ea59', 'us-east-1c | subnet-00704186812091409', 'us-east-1d | subnet-09d5678bf7f966711', 'us-east-1e | subnet-0f118fffd656feab8', and 'us-east-1f | subnet-07ed918945365bbc9'. The 'Next' button is highlighted with a red box.

Select all Subnets



No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer
Choose from your existing load balancers.

Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▾ MyTargetGroup | HTTP X

MyTargetGroup | HTTP
Application Load Balancer: MyLoaderBalancer

Health checks - optional

Health check type [Info](#)
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

EC2 ELB

Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

300 ▲ ▼ seconds

Additional settings - optional

Monitoring [Info](#)
 Enable group metrics collection within CloudWatch

Cancel Previous Skip to review Next

click on Next

Minimum capacity
1

Maximum capacity
1

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

None

Scaling policy name
Target Tracking Policy

Metric type
Average CPU utilization

Target value
50

Instances need
300 seconds warm up before including in metric

Disable scale in to create only a scale-out policy

Instance scale-in protection - optional

Instance scale-in protection
If instance scale-in is enabled, newly launched instances will be protected from scale in by default.

Enable instance scale-in protection

[Cancel](#) [Previous](#) [Skip to review](#) [Next](#)

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

Auto Scaling groups (1/3)											Create an Auto Scaling group
Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones				
<input checked="" type="checkbox"/> auto scaling group-EC2	auto-scaling-EC2	0	Updating cap...	1	1	1	us-east-1a, us-east-1b, us-east-1c, us-east-1d				
<input type="checkbox"/> awseb-e-f5cfx2bn2g-stack-A1	AWSEBEC2LaunchTemplate_MKUHKB	1	-	1	1	4	us-east-1a, us-east-1b, us-east-1d				
<input type="checkbox"/> awseb-e-5sd7nzaayi-stack-A1	AWSEBEC2LaunchTemplate_ICZZxWFi	1	-	1	1	1	us-east-1a, us-east-1b, us-east-1d				

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 Templetees 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

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z...	Public IPv4 DNS	Public IPv4...	Elastic IP
-	i-066df060cdcb42b2b	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-207-145-16.c...	54.207.145.16	-
-	i-0cfb1187f07546	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-235-138-228...	54.235.138.228	-
<input checked="" type="checkbox"/>	i-022ccbe2c8d982e50	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-52-91-130-93.co...	52.91.130.93	-
-	i-0fbfe89d6e816f45a	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-52-23-156-37.co...	52.23.156.37	-
-	i-09e153a7a47f863d0	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-164-39-22.co...	54.164.39.22	-
-	i-076a567acc9ea68d	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-164-111-250...	54.164.111.250	-
-	i-074c6e6f715a6209f	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-18-208-129-16.c...	18.208.129.16	-
-	i-004abda1fa1414ff	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-90-216-102.c...	54.90.216.102	-
-	i-01d10c10b973d881a	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-160-163-165...	54.160.163.165	-
-	i-0196ed03c86c23b06	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-91-250-137.c...	54.91.250.137	-
-	i-0865df3e0e4c3726	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-165-38-77.co...	54.165.38.77	-
<input checked="" type="checkbox"/>	i-04c61dc210a85c347	Running @Q	t2.micro	2/2 checks pass	No alarms +	us-east-1a	ec2-54-84-125-108.co...	5.84.125.108	-

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

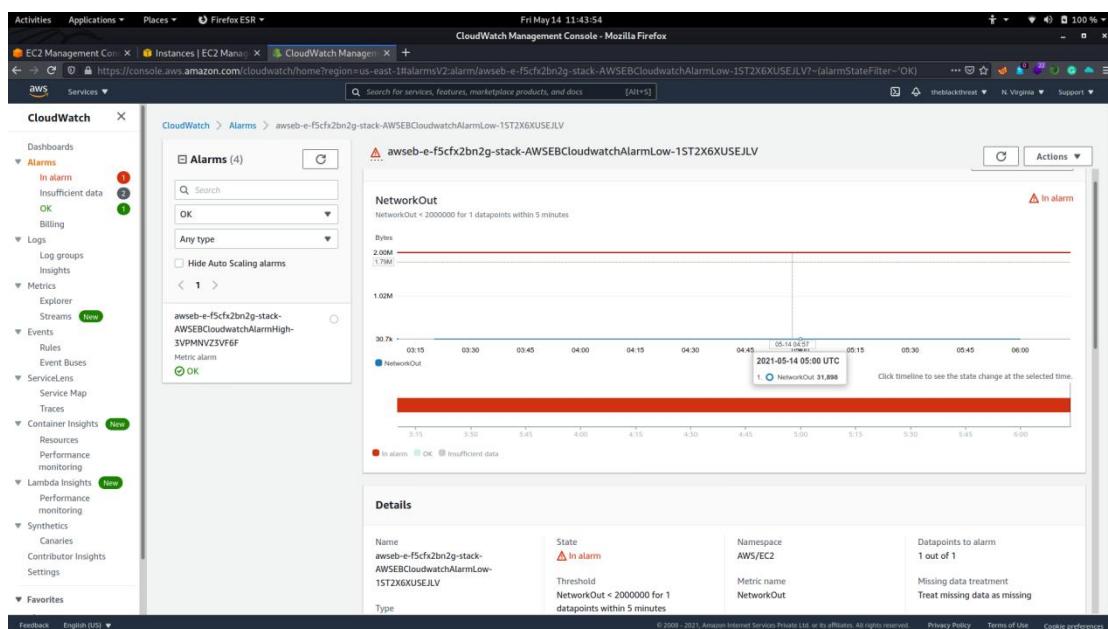
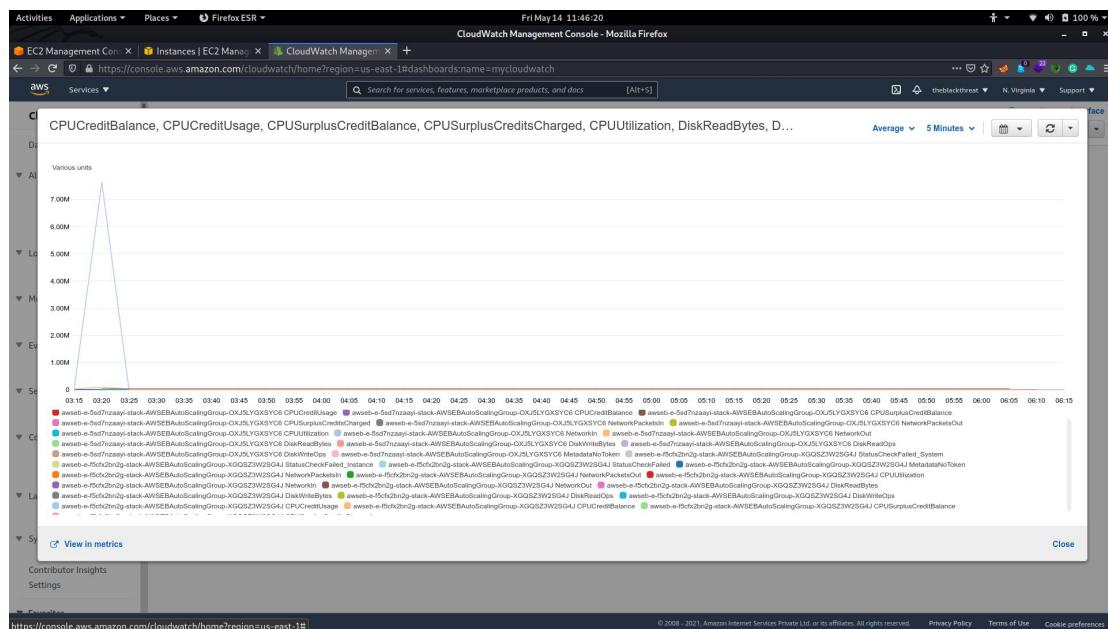
Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-0473af907d2e2ac	At 2021-05-14T05:42:56Z an instance was launched in response to an instance refresh.	2021 May 14, 11:12:58 AM +0:30	2021 May 14, 11:15:30 AM +0:30
Successful	Terminating EC2 instance: i-05b40f846f46bd63	At 2021-05-14T05:42:56Z an instance was taken out of service in response to an instance refresh. At 2021-05-14T05:42:56Z instance i-05b40f846f46bd63 was selected for termination.	2021 May 14, 11:12:56 AM +0:30	2021 May 14, 11:13:59 AM +0:30
Successful	Launching a new EC2 instance: i-05b40f846f46bd63	At 2021-05-14T03:22:39Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2021 May 14, 08:52:42 AM +0:30	2021 May 14, 08:53:13 AM +0:30
Successful	Terminating EC2 instance: i-0df26b0355b906724	At 2021-05-14T03:22:19Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2021 May 14, 08:52:19 AM +0:30	2021 May 14, 08:52:42 AM +0:30
Successful	Launching a new EC2 instance: i-0df26b0355b906724	At 2021-05-13T17:21:36Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2021 May 13, 10:51:38 PM +0:30	2021 May 13, 10:52:10 PM +0:30
Successful	Terminating EC2 instance: i-05af193f0162a407	At 2021-05-13T17:21:16Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2021 May 13, 10:51:16 PM +0:30	2021 May 13, 10:51:18 PM +0:30
Successful	Launching a new EC2 instance: i-05af193f0162a407	At 2021-05-13T17:15:21Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2021 May 13, 10:45:24 PM +0:30	2021 May 13, 10:45:56 PM +0:30
Successful	Terminating EC2 instance: i-0ceaf1192333cd8	At 2021-05-13T17:15:01Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2021 May 13, 10:45:01 PM +0:30	2021 May 13, 10:45:25 PM +0:30
Successful	Launching a new EC2 instance: i-0ceaf1192333cd8	At 2021-05-13T07:08:23Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2021 May 13, 12:38:26 PM +0:30	2021 May 13, 12:38:57 PM +0:30
Successful	Terminating EC2 instance: i-03fae07cd480ccdf	At 2021-05-13T07:08:02Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2021 May 13, 12:38:02 PM +0:30	2021 May 13, 12:38:24 PM +0:30

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.

Step 7: Review Instance Launch

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details

Microsoft Windows Server 2019 Base - ami-0fa60543f60171fe3

Free tier eligible

Root Device Type: ebs Virtualization type: hvm

If you plan to use this AMI for an application that benefits from Microsoft License Mobility, fill out the License Mobility Form. Don't show me this again

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
I2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	0.0.0.0/0	
All TCP	TCP	0 - 65535	0.0.0.0/0	
All TCP	TCP	0 - 65535	:/0	

Instance Details

Storage

Tags

[Edit instance details](#) [Edit storage](#) [Edit tags](#)

[Cancel](#) [Previous](#) [Launch](#)

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.)

EC2 > Instances > i-0e83e6a5503678ddc > Create image

Create image [Info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID
i-0e83e6a5503678ddc (Window-sample-AMI)

Image name

Maximum 127 characters. Can't be modified after creation.

Image description - optional

Maximum 255 characters

No reboot
 Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/s...	Create new snapshot ...	30	EBS General Purpose ...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

i During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag image and snapshots together
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately
Tag the image and the snapshots with different tags.

No tags associated with the resource.
[Add tag](#)

You can add 50 more tags.

[Cancel](#) [Create Image](#)

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.

Owned by me Add filter

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	Platform	Root Device Type	Virtualization
AMI-sample-W...	AMI-sample-W...	ami-0528c23df07013cc	ami-0528c23df07013cc	078434872459	Private	available	May 16, 2021 at 7:12:30 AM UTC+5:30	Windows	ebs	hvm

Image: ami-0528c23df07013cc

[Edit](#)

AMI ID	AMI Name	Source	Owner	Visibility	Status	Creation Date	Platform	Root Device Type	Virtualization
ami-0528c23df07013cc	AMI-sample-Window	078434872459/AMI-sample-Window	078434872459	Private	available	May 16, 2021 at 7:12:30 AM UTC+5:30	Windows	ebs	hvm

AMI Name: AMI-sample-Window
Source: 078434872459/AMI-sample-Window
Status: available
State Reason:
Platform details: Windows
Usage operation: RunInstances.0002
Virtualization type: hvm
Root Device Name: /dev/vda1
Root Device Type: ebs
Kernel ID: -
Product Codes: -
Boot mode: -

Now we can launch multiple instances with this sample.

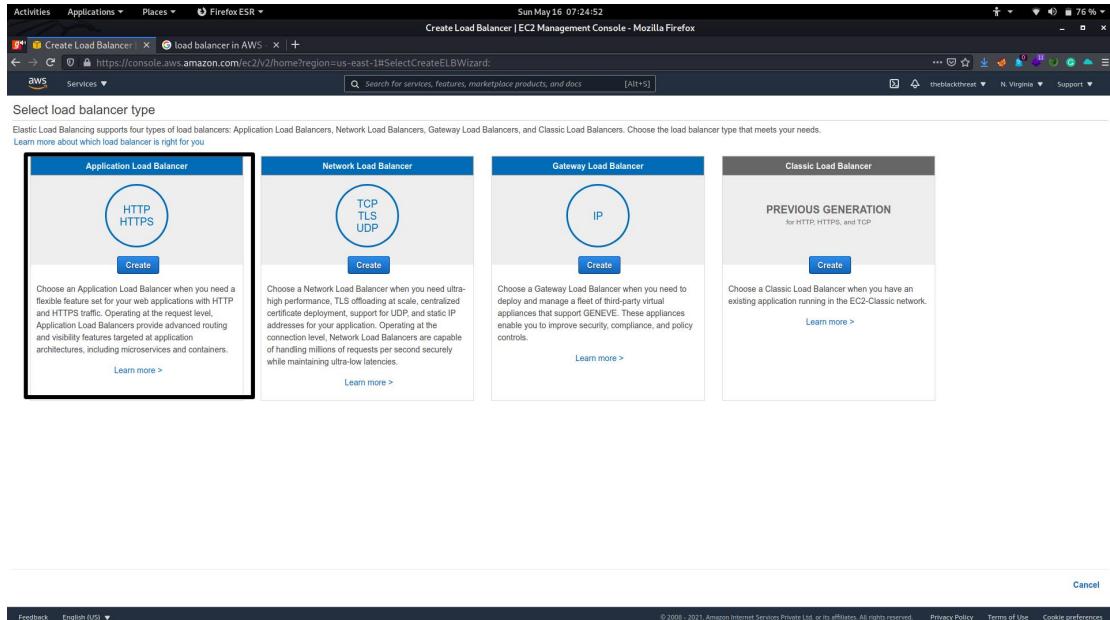
9. Objective: Working with load balancers

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 .

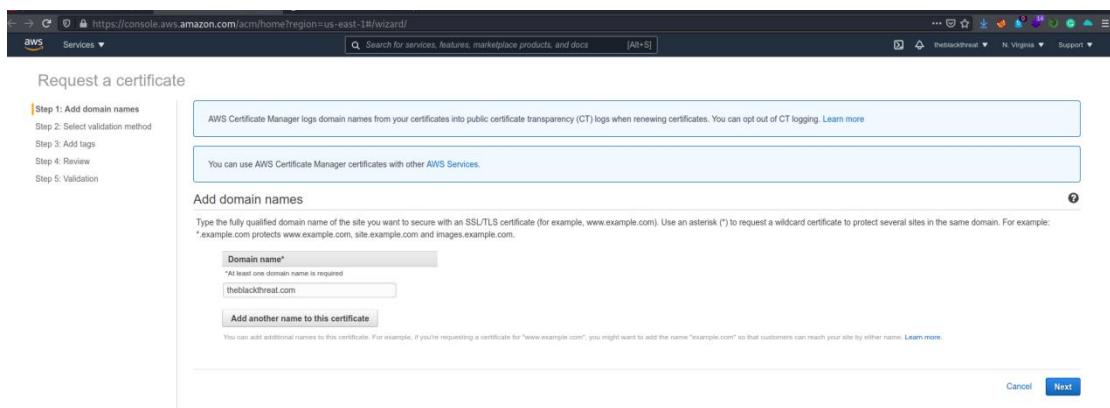


Step 2: Fill the basic configuration with name, interface, listeners, VPC available zones.

Note: If you choose HTTPS listeners 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



1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups **4. Configure Routing** 5. Register Targets 6. Review

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify here. It also performs health checks or add listeners after the load balancer is created.

Target group

Target group (i)	New target group load-balancer-for-server
Name (i)	<input type="text" value="load-balancer-for-server"/>
Target type	<input checked="" type="radio"/> Instance <input type="radio"/> IP <input type="radio"/> Lambda function
Protocol (i)	HTTP
Port (i)	<input type="text" value="80"/>
Protocol version (i)	<input checked="" type="radio"/> HTTP1 Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2. <input type="radio"/> HTTP2 Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available. <input type="radio"/> gRPC Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

Protocol (i)	HTTP
Path (i)	<input type="text" value="/"/>

» Advanced health check settings

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---->

Select load balancer type

Elastic Load Balancing supports four types of load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. Choose the load balancer type that meets your needs.

Learn more about which load balancer is right for you

The screenshot shows the AWS Elastic Load Balancing service page. It features four main sections for different load balancer types:

- Application Load Balancer:** Handles HTTP and HTTPS traffic. A 'Create' button is present.
- Network Load Balancer:** Handles TCP, TLS, and UDP traffic. A 'Create' button is present.
- Gateway Load Balancer:** Handles IP traffic. A 'Create' button is present.
- Classic Load Balancer (highlighted):** Handles previous generation traffic for HTTP, HTTPS, and TCP. A 'Create' button is present.

Each section includes a 'Learn more' link.

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

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

Step 1: Define Load Balancer

Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name:

Create LB Inside:

Create an internal load balancer:

Enable advanced VPC configuration:

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

Add

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

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time.

Assign a security group: Create a new security group
 Select an existing security group

Security Group ID			Name	Description	Actions
<input checked="" type="checkbox"/>	sg-096e92709b93effeb	default	default VPC security group		Copy to new
<input type="checkbox"/>	sg-0d581100217069f1d	launch-wizard-1	launch-wizard-1 created 2021-05-16T07:02:07.536+05:30		Copy to new

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

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

Step 5: Add EC2 Instances

The table below lists all your running EC2 instances. Check the boxes in the Select column to add those instances to this load balancer.

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-0e83e6a5503678ddc	stopped	launch-wizard-1	us-east-1a	subnet-01b9dc83...	172.31.16.0/20
<input checked="" type="checkbox"/>	Window-sample-AMI	stopped	launch-wizard-1	us-east-1a	subnet-01b9dc83...	172.31.16.0/20

Availability Zone Distribution

1 instance in us-east-1a

Enable Cross-Zone Load Balancing

Enable Connection Draining

30 | seconds

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

Load Balancer Creation Status

Successfully created load balancer Load balancer load-balancer-1 was successfully created. Note: It may take a few minutes for your instances to become active in the new load balancer.
Close

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

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



1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Cancel and Exit

Search for an AMI by entering a search term e.g. "Windows"

My AMIs

AMI-sample-Windows - ami-0528c23df07013cc

This is an AMI for windows instance

Root device type: ebs Virtualization type: hvm Owner: 079434872459 ENA Enabled: Yes

Select

64-bit (x86)

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

EC2 > Instances > i-0e83e6a5503678ddc > Connect to instance

Connect to instance Info

Connect to your instance i-0e83e6a5503678ddc (Window-sample-AMI) using any of these options

Session Manager | **RDP client** | EC2 Serial Console

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download remote desktop file](#)

When prompted, connect to your instance using the following details:

Public DNS User name
 ec2-18-214-96-70.compute-1.amazonaws.com Administrator

Password Get password

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

Step 9: connect it.

Activities Applications Places Remmina Sun May 16 07:59:38 ec2-18-214-96-70.compute-1.amazonaws.com

Recycle Bin

Feedback

Micos...

Amazon Web Services

Ec2LaunchSettings

Search

Server Manager

Settings

Windows Accessories

Windows Administrative Tools

Windows Ease of Access

Windows PowerShell

Windows Security

Windows System

Windows Server

Server Manager

Windows PowerShell

Windows PowerShell ISE

Windows Administrative Tools

Task Manager

Control Panel

Run...

Event Viewer

File Explorer

Networks

Network 2

Do you want to allow your PC to be discoverable by other PCs and devices on this network?

We recommend allowing this on your home and work networks, but not public ones.

Yes No

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

Sun May 16 08:02:40 EC2 Management Console - Mozilla Firefox

<https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers:sort=loadBalancerName>

New EC Experience Tell us what you think

EC2 Dashboard Events Tags Limits

Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups Elastic IPs

Feedback English (US) ▾

Load balancer: load-balancer-1

Instances Instances Health check Listeners Monitoring Tags Migration

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-0ef3efad503678ddc	Window-sample-AMI	us-east-1a	OutOfService ⓘ	Remove from Load Balancer

Edit Availability Zones

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
us-east-1f	subnet-0af9c34d8e9fb0a8f2	172.31.84.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1e	subnet-01c21e15964947894	172.31.48.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1d	subnet-065287071420321a	172.31.80.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1c	subnet-01a090de6f65748e	172.31.0.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1b	subnet-07e776bab0c520c0a	172.31.32.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1a	subnet-01b9e650d50a73b66	172.31.16.0/20	1	No (Availability Zone contains no healthy targets)	Remove from Load Balancer

© 2006 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Sun May 16 08:02:51 EC2 Management Console - Mozilla Firefox

<https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers:sort=loadBalancerName>

New EC Experience Tell us what you think

EC2 Dashboard Events Tags Limits

Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups Elastic IPs

Feedback English (US) ▾

Load balancer: load-balancer-1

ELB 5XXs Count HTTP 4XXs Count Backend Connection Errors Count Surge Queue Length (Count) HTTP 2XXs (Count)

HTTP 3XXs (Count) ELB 4XXs (Count) Spillover Count (Count) Estimated ALB Consumed LCUs (Count) Estimated ALB Active Connection Count (Count)

Estimated ALB New Connection Count (Count) Estimated Processed Bytes (Bytes)

© 2006 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

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

10. Objective: Working with EFS

Elastic File System(EFS): Amazon EFS is cloud based storage service provided by AWS. This is designed to be 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 accessible for single EC2 Instance with available regions.

EFS allows mount file system across multiple Instances 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.

Create file system

Create an EFS file system with service recommended settings. [Learn more](#)

Name - optional
Name your file system.

Name must not be longer than 256 characters, and must only contain letters, numbers, and these characters: + - . _ : /

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 Instances to connect to your file system. [Learn more](#)

default

Availability and Durability
Choose Regional (recommended) to create a file system using regional storage classes. Choose One Zone to create a file system using One Zone storage classes. [Learn more](#)

Regional
Stores data redundantly across multiple AZs

One Zone
Stores data redundantly within a single AZ

[Cancel](#) [Customize](#) [Create](#)

Step 2: create file system

File systems (1)									
View details Delete Create file system									
Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)	File system state	Creation time	Availability Zone
theblackthreat-EFS	fs-02a50bb6	Encrypted	6.00 KB	6.00 KB	0 Bytes	-	Available	Sun, 16 May 2021 03:02:02 GMT	Regional

Step 3: Now launch an Instance(Amazon Linux 2 AMI) and make sure you have chosen 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.

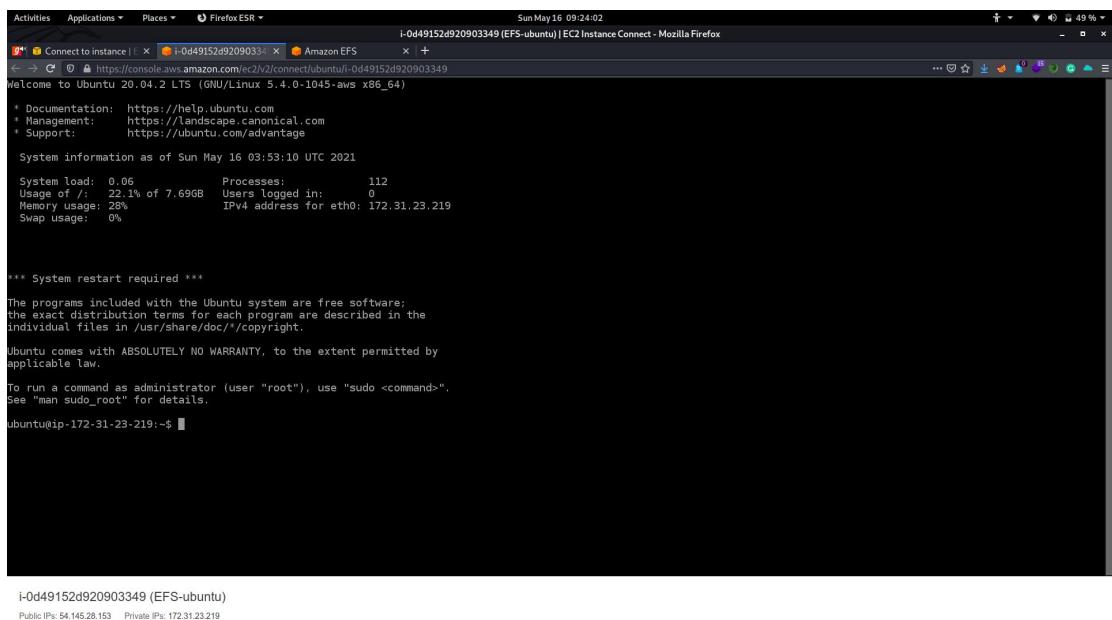
4: Configure Security Group, set Assign a security group to Select an existing security group. Choose the default security group to make sure that it can access your EFS file system.

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 chosen linux system then you can attach it.

Step 5: after complete connection 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 our instance 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.

The screenshot shows the Amazon RDS console with the following details:

- DB identifier:** database-tbt
- Status:** Available
- Engine:** MySQL Community
- Class:** db.t2.micro
- Region & AZ:** us-east-1b

Now Step 2:

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

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z...	Public IPv4 DNS	Public IPv4...	Elastic IP
Linux-RDS	i-08ae801ef2351c241	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-100-26-61-132.c...	100.26.61.132	-

```

Tue May 18 19:12:56
ubuntu@ip-172-31-30-72:~$ ssh -l practical pem
(theblackthreat@theblackthreat:~/Desktop/AWS) [~] ssh -l practical pem
$ chmod 400 practical.pem
[theblackthreat@theblackthreat:~/Desktop/AWS] [~] ssh -l practical pem ubuntu@ip-172-31-30-72
[sudo] password for theblackthreat:
The authenticity of host 'ec2-100-26-61-132.compute-1.amazonaws.com (100.26.61.132)' can't be established.
EDSA key fingerprint is SHA256:n9Ji9nG5XmB0oIhKzC76TuvvCTAoqd+xaqed3l4uc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-26-61-132.compute-1.amazonaws.com,100.26.61.132' (EDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1045-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Tue May 18 13:42:05 UTC 2021

 System load: 0.01              Processes: 92
 Usage of /: 14.7% of 7.60GB   Users logged in: 0
 Memory usage: 19%
 Swap usage: 0%                IP address for eth0: 172.31.30.72

 4. Connect to your instance using its Public DNS:
 0 packages can be updated, 0 of which are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright*.Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-30-72:~$ 

```

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

```

Activities Applications Places Terminal Tue May 18 19:53:46
ubuntu@ip-172-31-30-72:~$ mysql -h database-tbt.c6lcmw9ehorn.us-east-1.rds.amazonaws.com -P 3306 -u theblackthreat -p
Enter password:
ERROR 2002 (HY000): Can't connect to MySQL server on 'database-tbt.c6lcmw9ehorn.us-east-1.rds.amazonaws.com' (115)
mysql> CREATE DATABASE student;
Query OK, 1 row affected (0.00 sec)

mysql> USE student;
Database changed

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)

```

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 connects to the RDS DB instance.

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

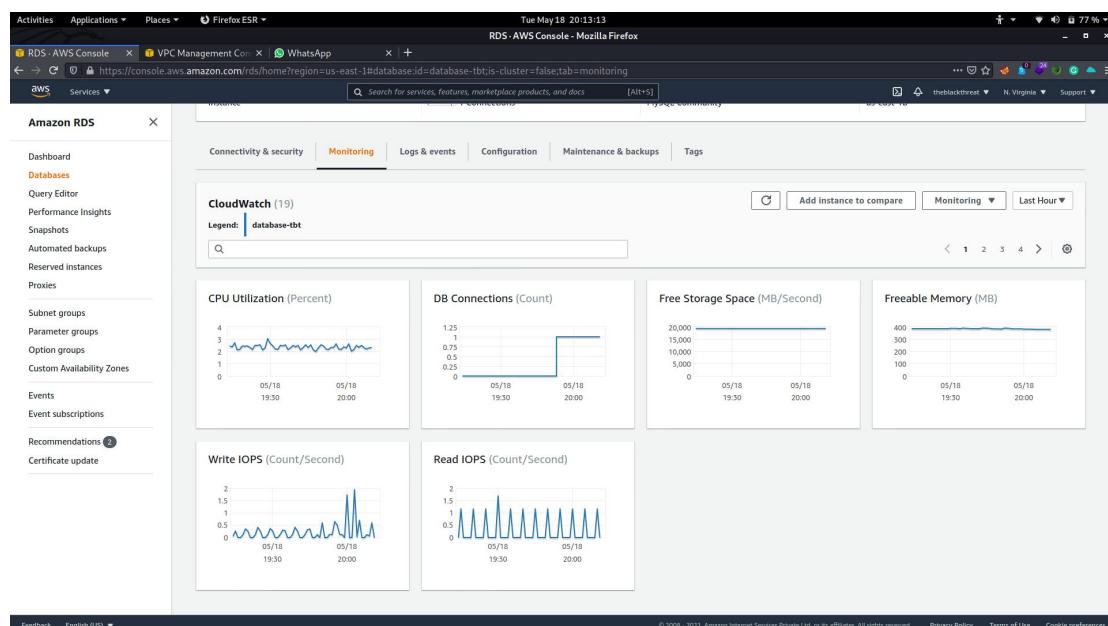
mysql> create table student_class(
    >     Name VARCHAR(20) NOT NULL,
    >     TITLE VARCHAR(10) NOT NULL,
    >     PRIMARY KEY (TITLE));
Query OK, 0 rows affected (0.03 sec)
```

```
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)
```

```
Activities Applications Places Terminal Tue May 18 20:08:41
Database changed phoenixos.com:3306 for create table movie
mysql> create table student_class;
ERROR 1113 (42000): A table must have at least 1 column
mysql> create table student_class
    --> name,
        --> ID;
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 'name,
class' at line 2
mysql> create table student_class(
    --> Name VARCHAR(20) NOT NULL,
    --> CLASS VARCHAR(12) NOT NULL,
    --> PRIMARY KEY (Name));
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 'CLASS VARCHAR(12) NOT NULL,' at line 1
mysql> create table student_class(
    --> Name VARCHAR(20) NOT NULL,
    --> CLASS VARCHAR(12) NOT NULL,
    --> PRIMARY KEY (Name));
DESCRIBE movies;
The terminal prints out information about the table:
Field | Type | Null | Key | Default | Extra
-----+-----+-----+-----+-----+-----+
| genre | varchar(20) | NO |   | NULL |   |
| director | varchar(60) | NO |   | NULL |   |
| release_year | int | NO |   | 0 |   |
| title | varchar(50) | NO | PRI | NULL |   |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

3. Insert movie information in column order—title, genre, director, and release year. Use the INSERT command.
INSERT INTO movies VALUES ('Avatar', 'Science Fiction', 'James Cameron', 2009);
```

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



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

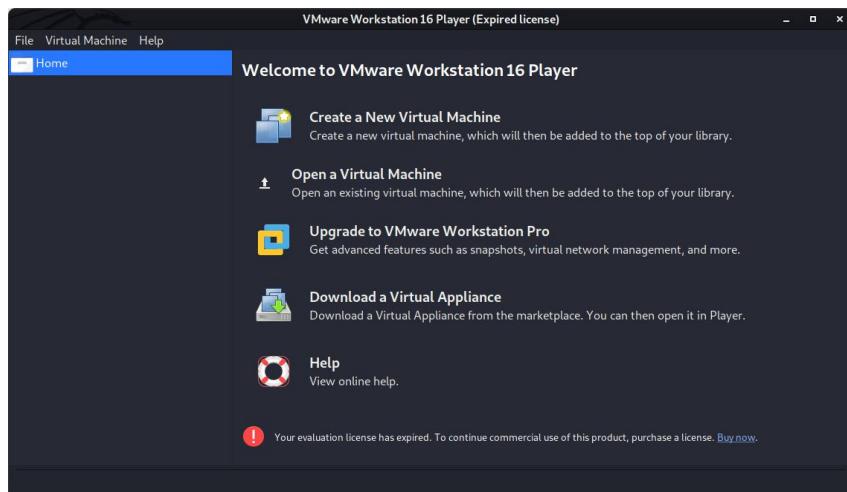
what is virtualization?

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).

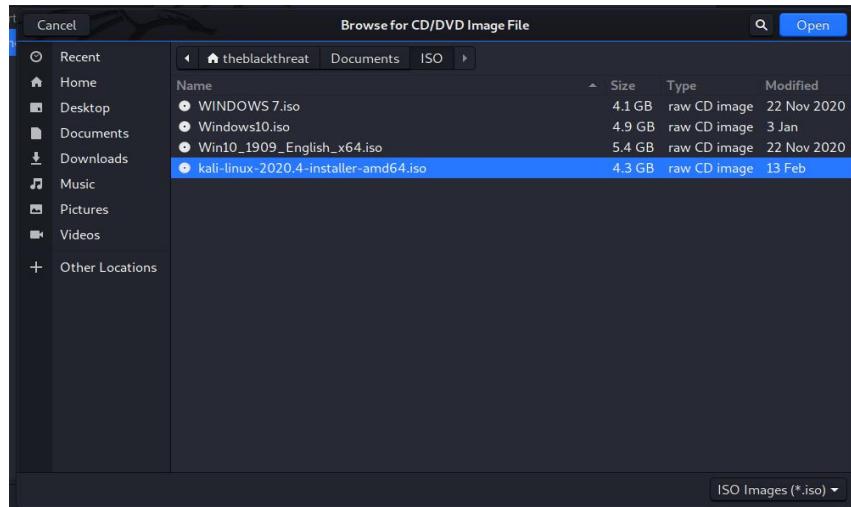
VMware: VMware Workstation is the first product ever released by the software company. It enables users to create and run VMs directly on a single Windows or Linux desktop or laptop. Those VMs run simultaneously with the physical machine. Each VM runs its own OS such as Windows or Linux.

Creating Kali linux in VMware:

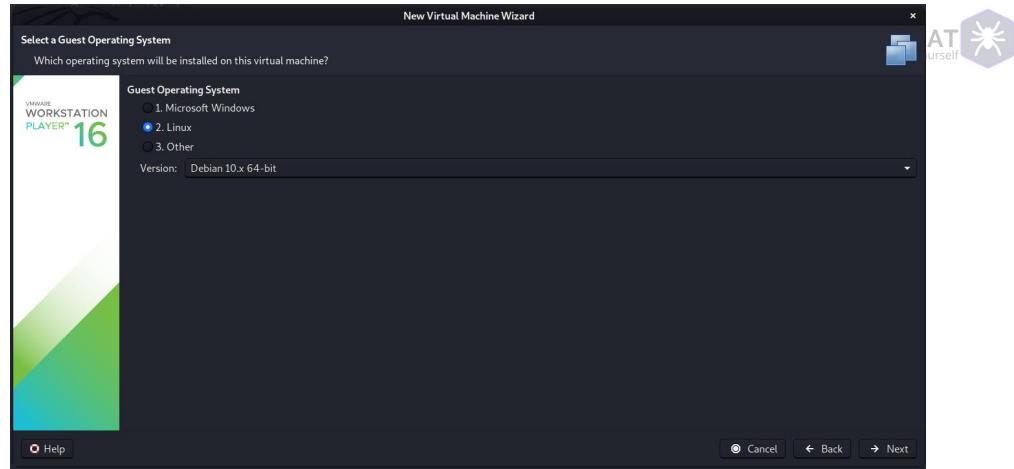
Step 1: Open VMware or download from [here](#)



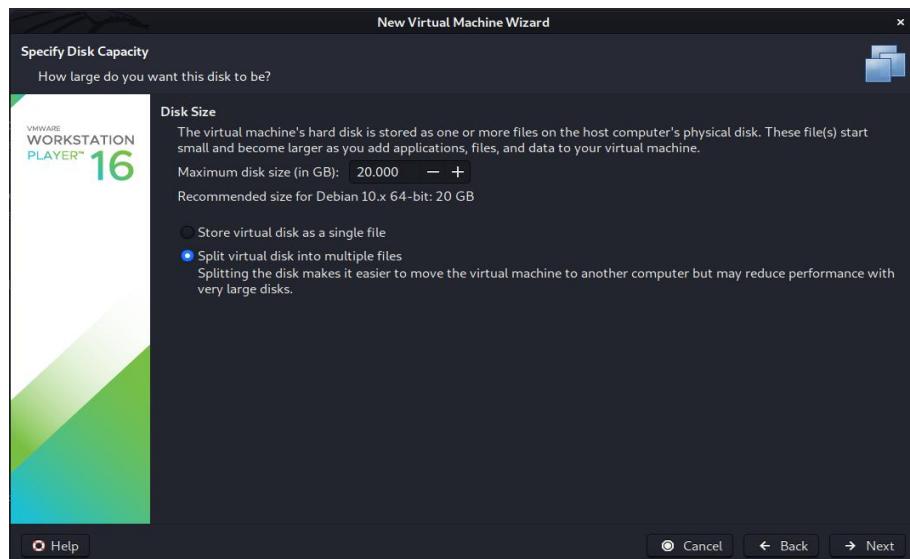
Step 2: Click on Create a New Virtual Machine. then browse your linux ISO file.



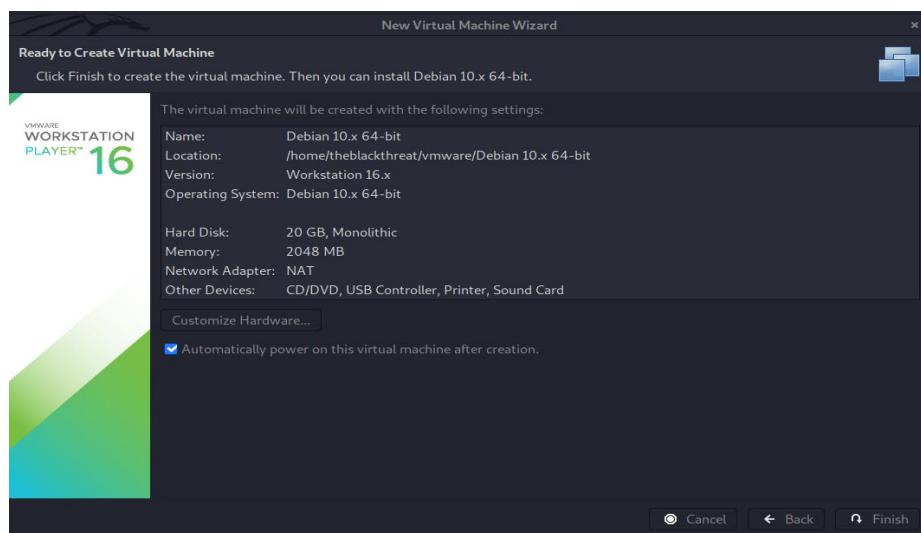
Step 3: Click Next and choose Linux ---> in context of kali--->Debian based version-->.Next



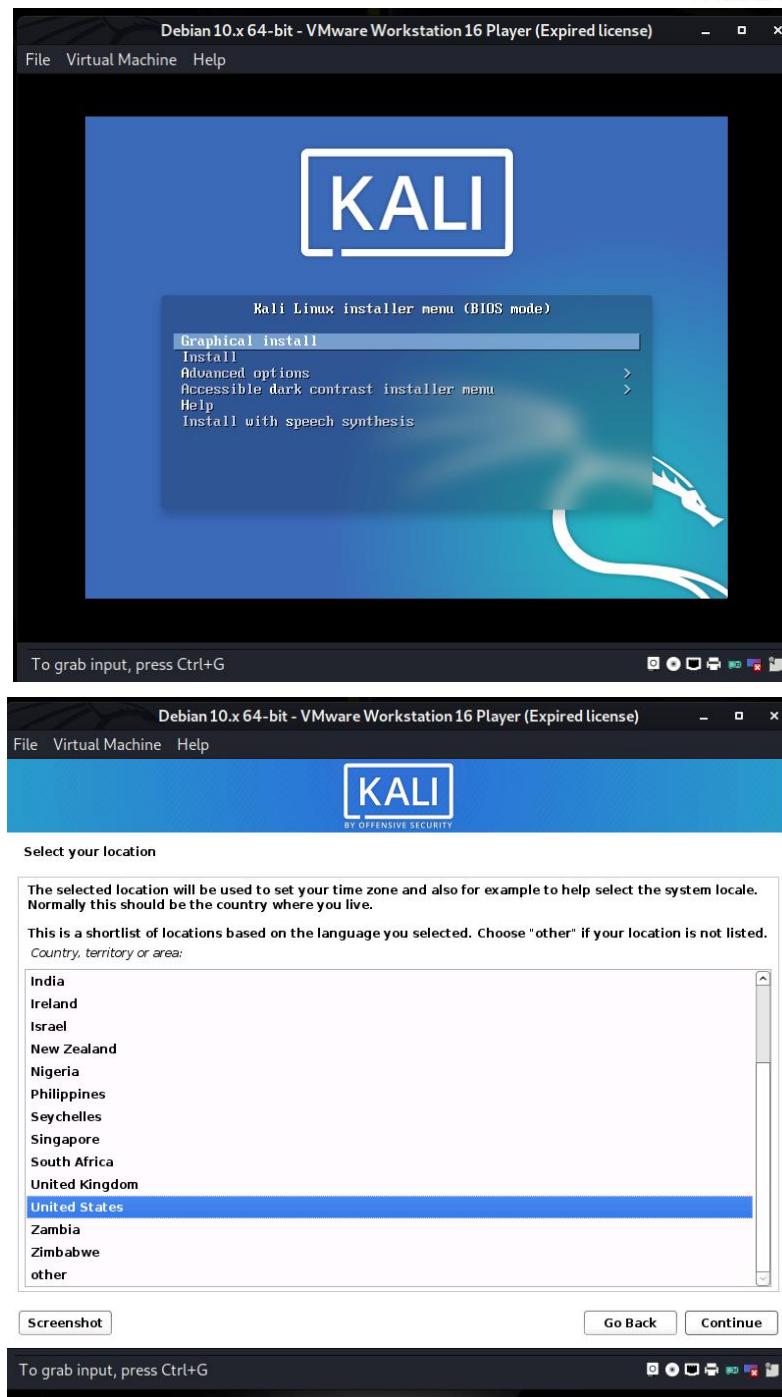
Step 4: Select your comfortable storage. and click on split it on single device ---> Next.



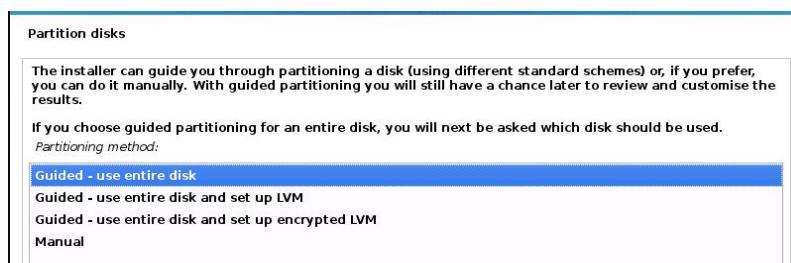
Step 5: See all configuration and then finish it.

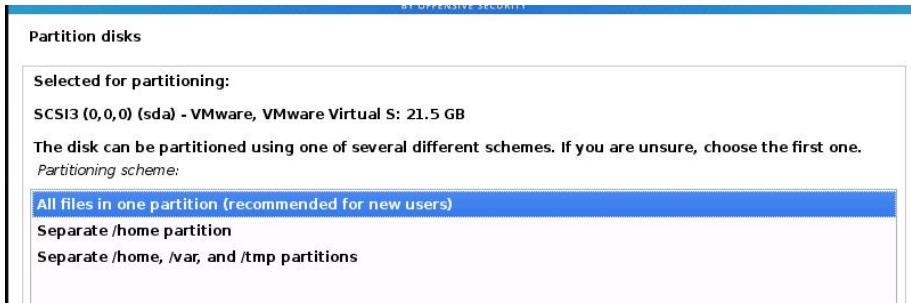


Step 6: after finish installation process if going.

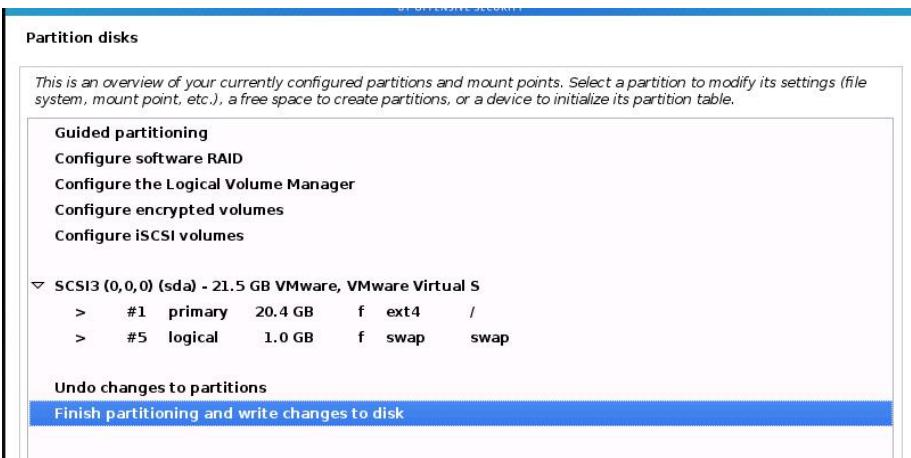


after this add system name and password and continue

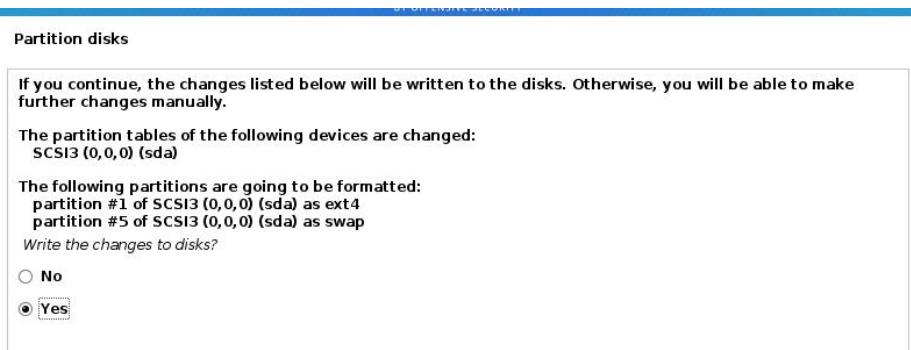




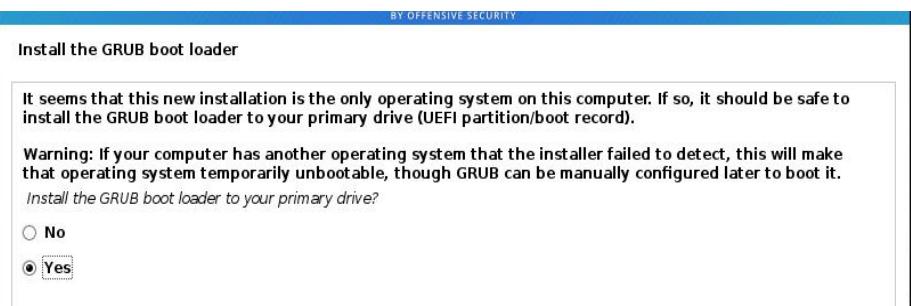
Step 7: Partition of disk all file are should be in same place so choose first one.



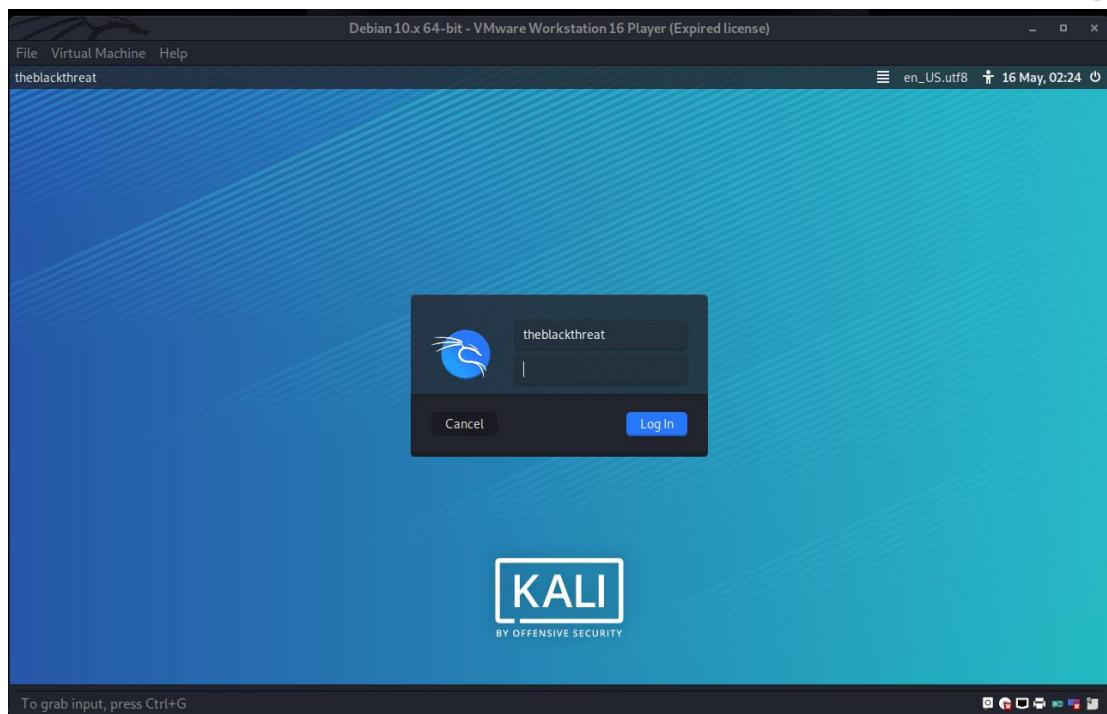
Step 8: give the permission for disk changes this will allow you to move further.



Step 9: Install GRUB Boot loader click on yes.



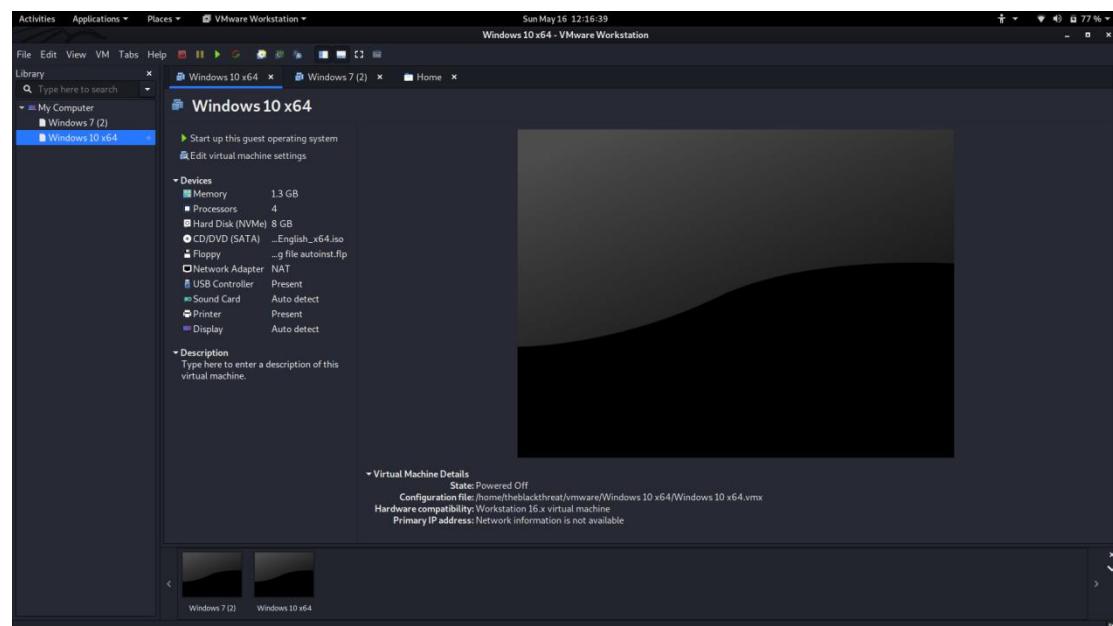
STep 10: after installation enter youre username and password and you will done.



Creating Snapshot on VMware for

snapshot make work more effiecent 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 servers 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.
