

U18CSI7201L - CLOUD COMPUTING

LABORATORY OBSERVATION

Nithin Prasath C – 22BCS081

Ex1 - Creating a virtual server in AWS public cloud

AIM:

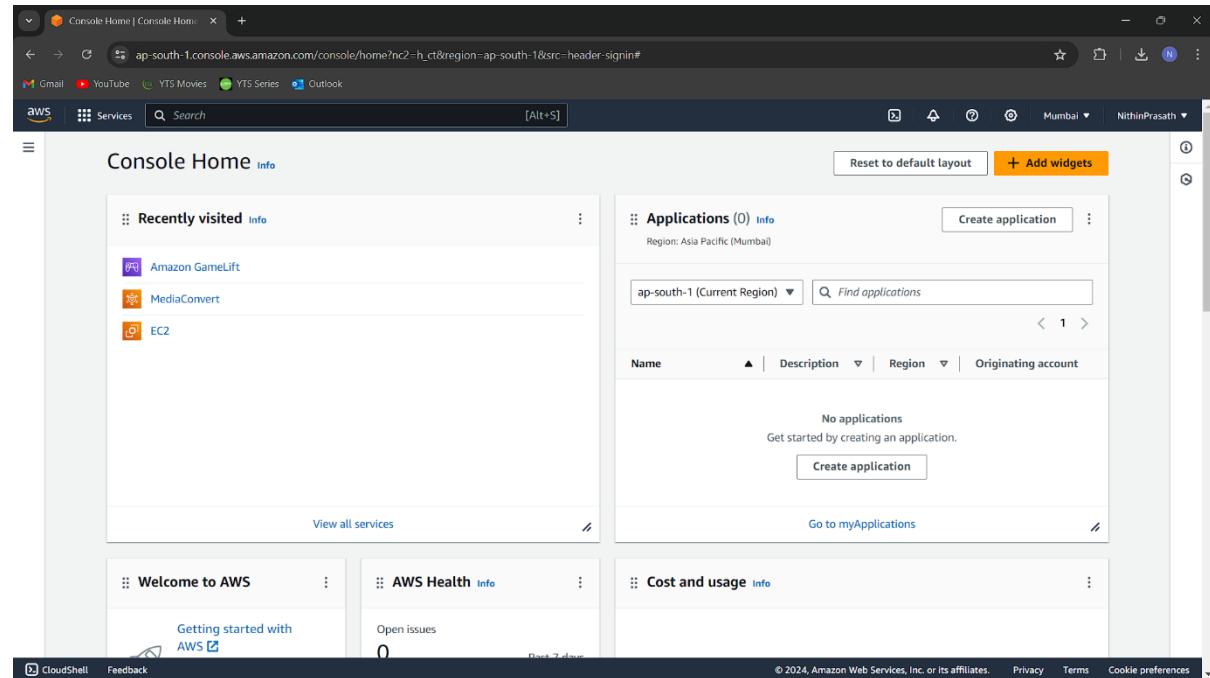
To provide step-by-step instructions for creating a virtual server in AWS public cloud, including screenshots for each step.

DESCRIPTION:

Amazon Web Services (AWS) is a comprehensive and widely adopted cloud platform that offers a range of services including computing power, storage, and databases. AWS enables users to run virtually any application in the cloud with flexible, scalable, and cost-effective infrastructure. One of the core services provided by AWS is the Elastic Compute Cloud (EC2), which allows users to create and manage virtual servers, known as instances. In this exercise, we will create a virtual server using AWS EC2.

OUTPUT:

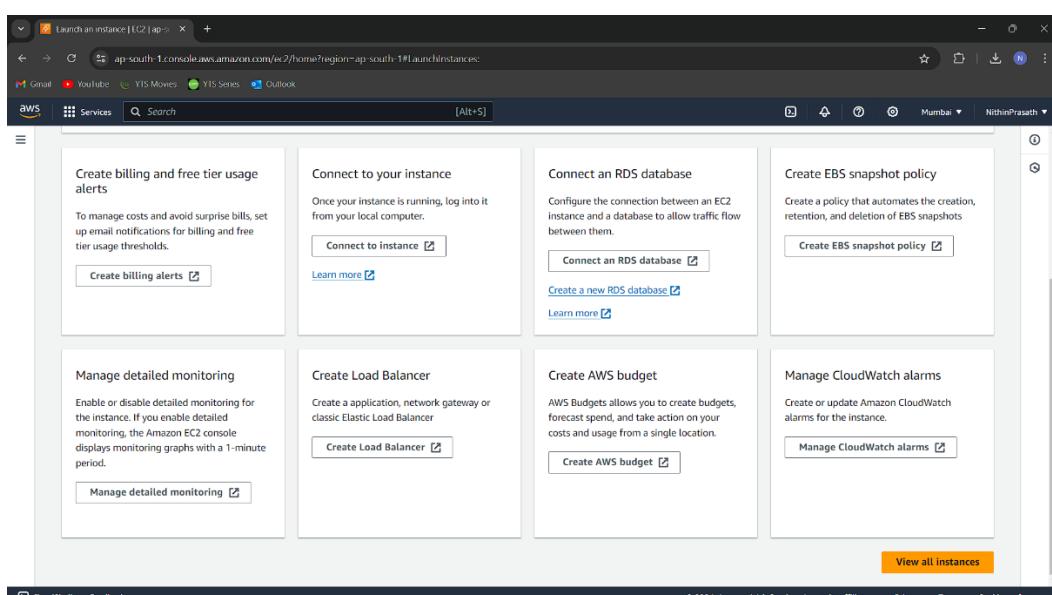
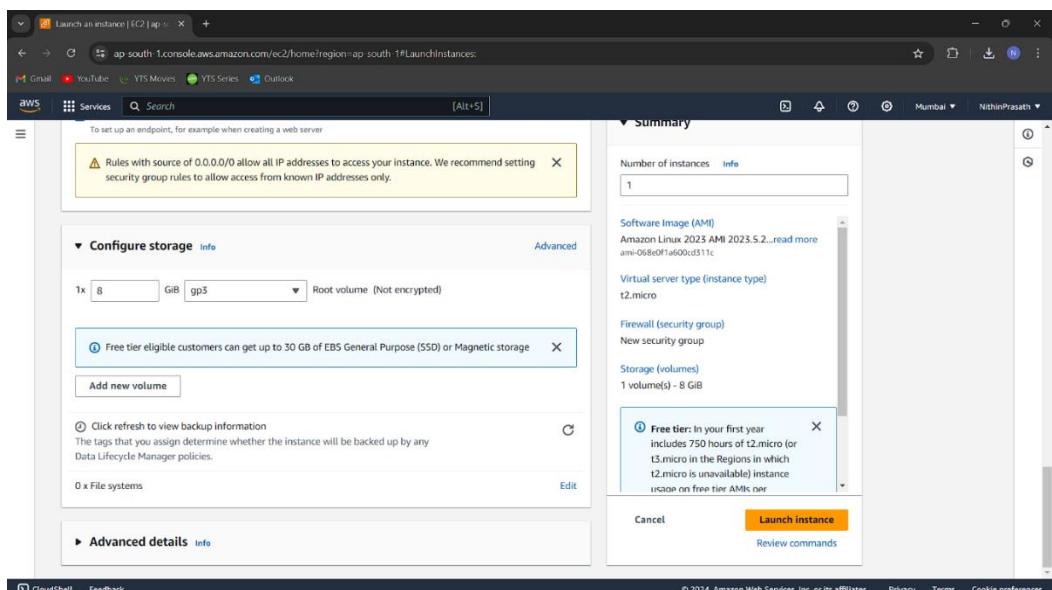
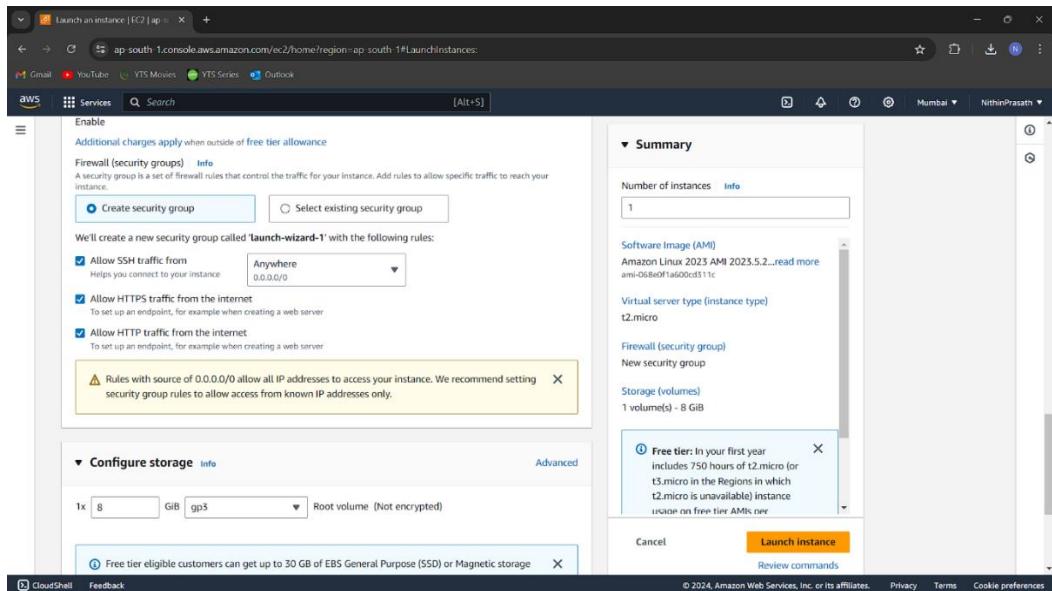
Amazon Image:



The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like Instances, Images, Elastic Block Store, and Network & Security. The main area has a "Resources" section with a table showing counts for various EC2 components. Below it is a "Launch instance" section with a "Launch instance" button and a "Migrate a server" option. To the right, there's a "Service health" section showing the Asia Pacific (Mumbai) Region is operating normally, and a "Zones" section. A large sidebar on the right details "EC2 Free Tier Info" with sections for "2 EC2 free tier offers in use" and "Offer usage (monthly)".

This screenshot shows the first step of the "Launch instance" wizard. It asks for a "Name" (set to "test1") and provides a "Search our full catalog including 1000s of application and OS images" bar. Below is a "Quick Start" section with icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A "Software Image (AMI)" section shows "Amazon Linux 2023 AMI" as the selected option. A callout box highlights the "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs." At the bottom are "Cancel" and "Launch instance" buttons.

This screenshot shows the second step of the "Launch instance" wizard, "Create key pair". It asks for a "Key pair name" (set to "as") and "Key pair type" (set to "RSA"). It also specifies "Private key file format" as ".pem". A callout box provides instructions for storing the private key securely. At the bottom are "Cancel" and "Create key pair" buttons.



The screenshot shows the AWS CloudShell interface. At the top, a green success banner reads "Successfully initiated launch of instance i-0da80cda8a4f7370a". Below it, a "Launch log" section lists several steps with green "Succeeded" status icons. The "Next Steps" section contains four cards: "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy".

The screenshot shows the AWS Instances page. A table lists one instance: "test1" (i-0da80cda8a4f7370a), which is "Running". The "Actions" dropdown menu for this instance includes options like "Stop", "Start", "Reboot", "Terminate", and "Launch instances". The left sidebar shows navigation links for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security.

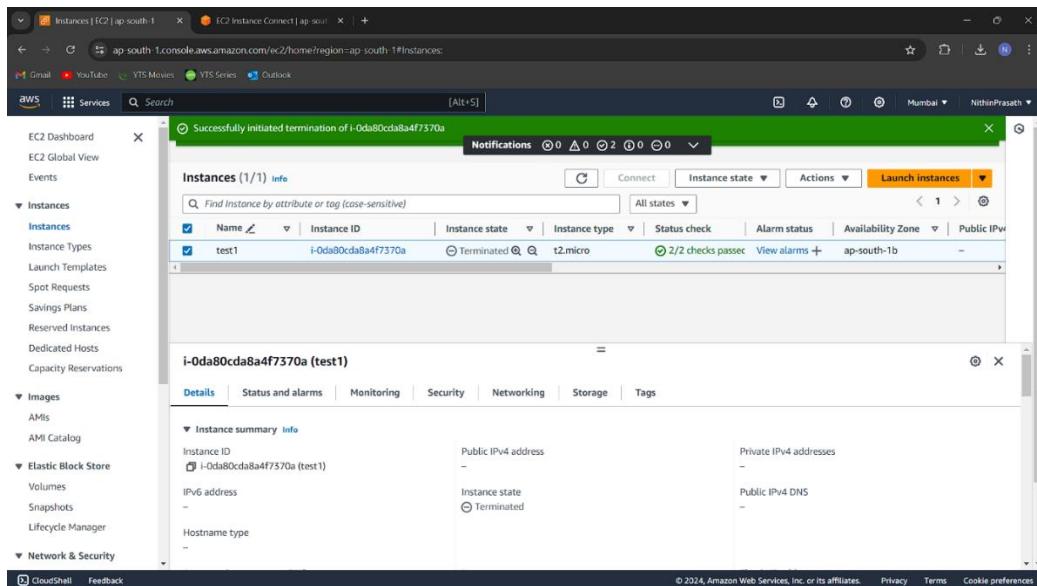
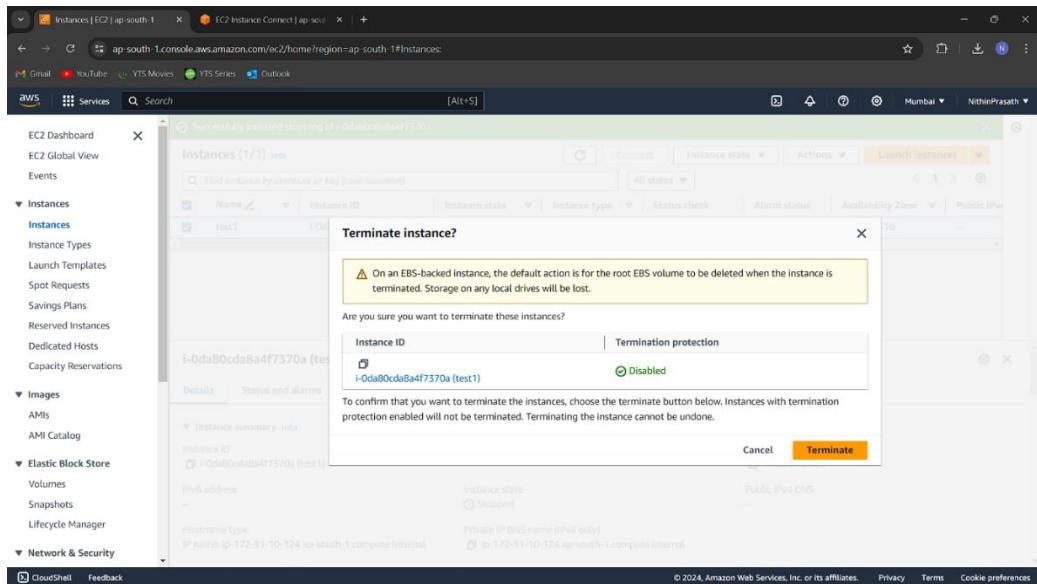
The screenshot shows the "Connect to instance" dialog box. It displays connection options for instance "i-0da80cda8a4f7370a (test1)". The "EC2 Instance Connect" tab is selected. Other tabs include "Session Manager", "SSH client", and "EC2 serial console". Under "Connection Type", the "Connect using EC2 Instance Connect" option is selected. The "Public IP address" field shows "43.204.218.133". The "Username" field contains "ec2-user". A note at the bottom states: "Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." A "Connect" button is at the bottom right.

The figure consists of three vertically stacked screenshots of the AWS CloudShell interface.

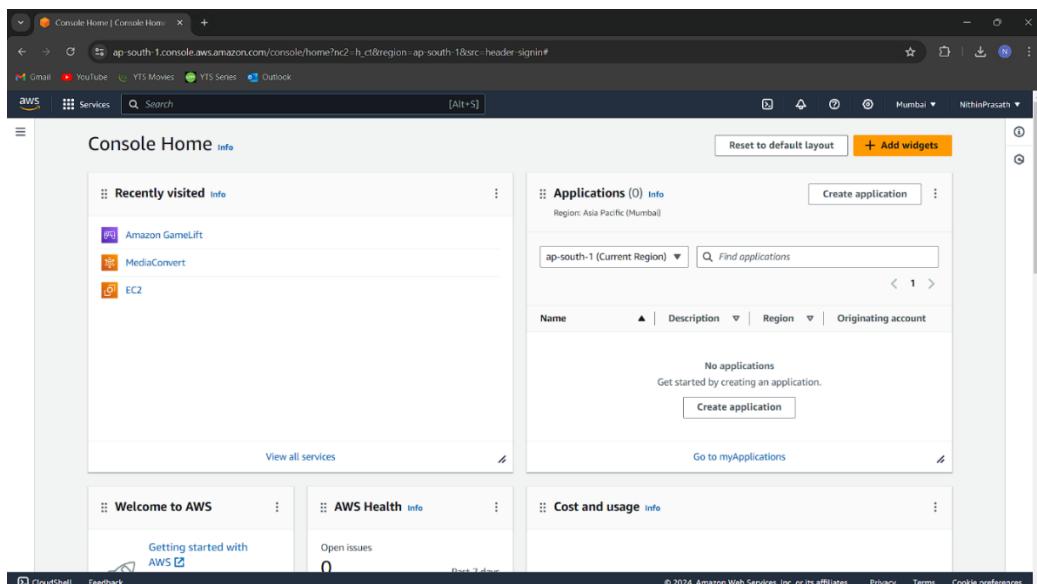
- Top Screenshot:** Shows a terminal session on an EC2 instance (i-0da80cda8a4f7370a). The user runs commands to create a directory named "sample":

```
[ec2-user@ip-172-31-10-124 ~]$ mkdir sample
[ec2-user@ip-172-31-10-124 ~]$ ls
sample
[ec2-user@ip-172-31-10-124 ~]$
```

- Middle Screenshot:** Shows the EC2 Instances page. A modal dialog titled "Stop instance?" is open for the instance i-0da80cda8a4f7370a (test1). The dialog contains a warning message about stopping the instance and its associated costs.
- Bottom Screenshot:** Shows the EC2 Instances page after the instance has been stopped. The instance status is now "Stopped". A green success message at the top indicates "Successfully initiated stopping of i-0da80cda8a4f7370a".



Windows/Linux/Mac Server



The screenshots illustrate the process of launching a new Amazon EC2 instance:

- Screenshot 1: EC2 Dashboard**
Shows the EC2 dashboard with various resource counts (Instances running: 0, Auto Scaling Groups: 0, Dedicated Hosts: 0, etc.). It includes sections for Launch instance, Service health, Offer usage (monthly), and Account attributes.
- Screenshot 2: Launch instance**
Shows the "Launch instance" wizard. Step 1: Set details (Region: Asia Pacific (Mumbai), Status: This service is operating normally). Step 2: Choose AMI (Name: test2). Step 3: Application and OS Images (Amazon Machine Image) (Microsoft Windows Server 2022 Base selected). Step 4: Configure Instance Details (Instance Type: t2.micro, Volume: 1 volume(s) - 30 GiB). Step 5: Add Tags (Name: test2). Step 6: Set Security Group (New security group selected). Step 7: Set Network and Storage (Leave default). Step 8: Set Key Pair (Create key pair selected). Step 9: Review and Launch. A tooltip for the "Free tier" is shown: "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month".
- Screenshot 3: Create key pair**
Shows the "Create key pair" dialog. Key pair name: as1, Key pair type: RSA (selected), Private key file format: pem (selected). A warning message states: "When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance." A "Create key pair" button is at the bottom.

The screenshot shows the 'Network settings' step of the EC2 launch wizard. It includes fields for Network (vpc-099b267a2bf937f83), Subnet (No preference), Auto-assign public IP (Enable), and Firewall (security groups). A note states: 'We'll create a new security group called "launch-wizard-2" with the following rules:'. Under this, three checkboxes are checked: 'Allow RDP traffic from Anywhere (0.0.0.0/0)', 'Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server', and 'Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server'. A warning message at the bottom says: '⚠ 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.' On the right, a summary panel shows 1 instance, Microsoft Windows Server 2022 AMI, t2.micro instance type, and 1 volume(s) - 30 GiB storage. A tooltip for the free tier indicates: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per year.'

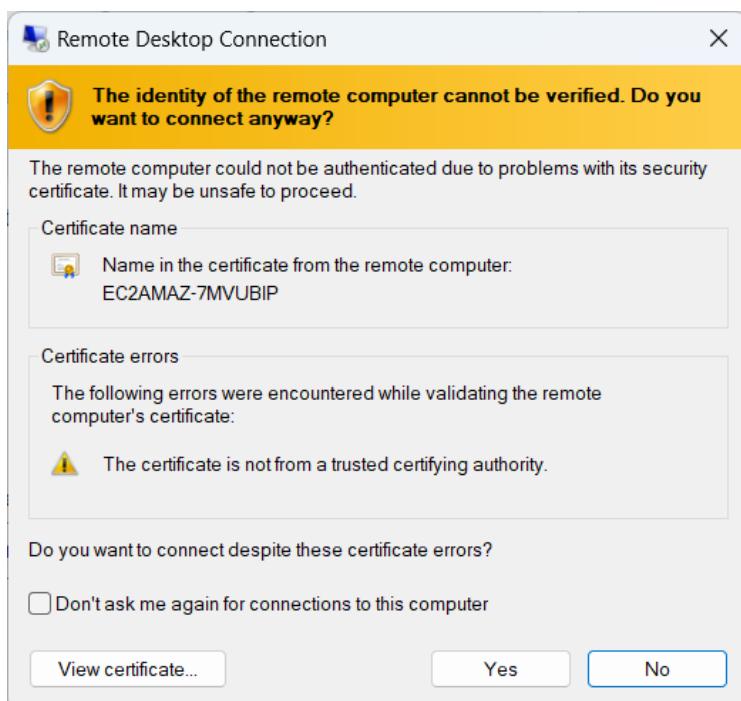
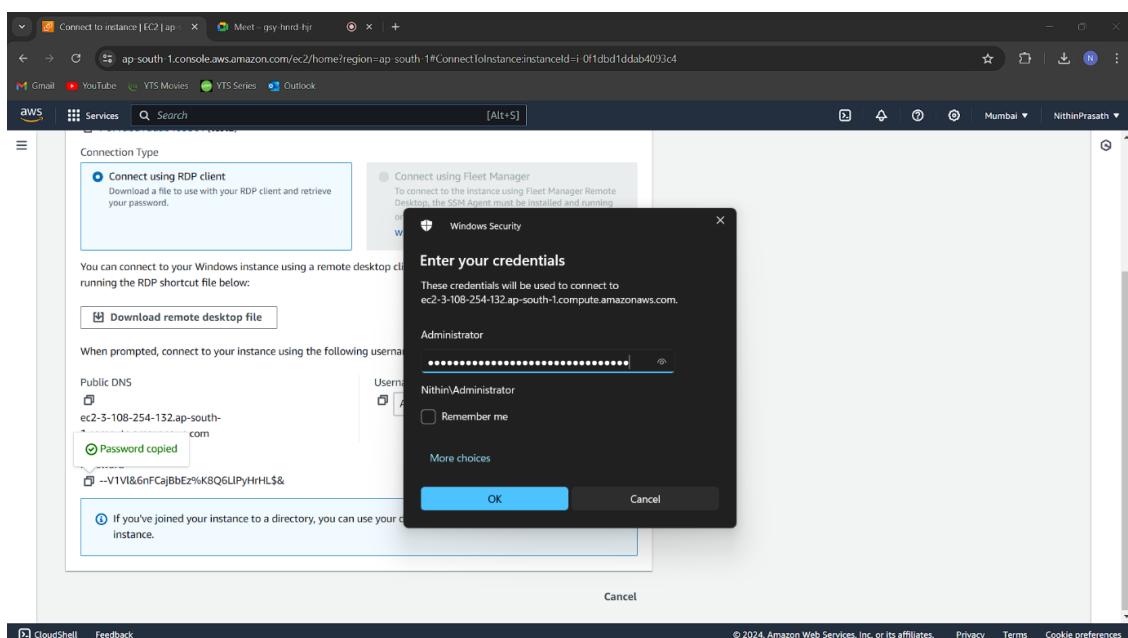
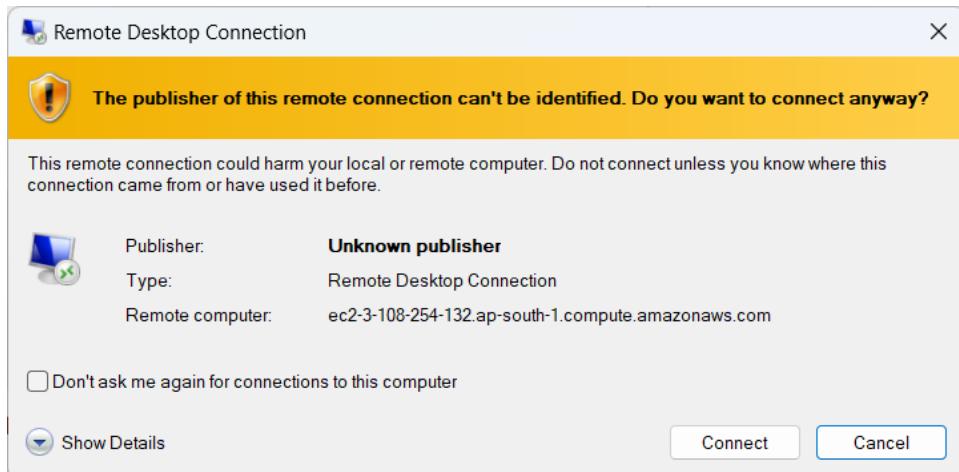
The screenshot shows the 'Launch an instance' success page. It displays a green banner: 'Success Successfully initiated launch of instance i-0f1dbd1ddab4093c4'. Below this, a 'Launch log' section shows five successful steps: 'Initializing requests', 'Creating security groups', 'Creating security group rules', 'Launch initiation', and 'Launch log'. Under 'Next Steps', there are four cards: 'Create billing and free tier usage alerts' (with a 'Create billing alerts' button), 'Connect to your instance' (with a 'Connect to instance' button), 'Connect an RDS database' (with a 'Connect an RDS database' button), and 'Create EBS snapshot policy' (with a 'Create EBS snapshot policy' button). A search bar at the top asks: 'What would you like to do next with this instance, for example "create alarm" or "create backup"?'.

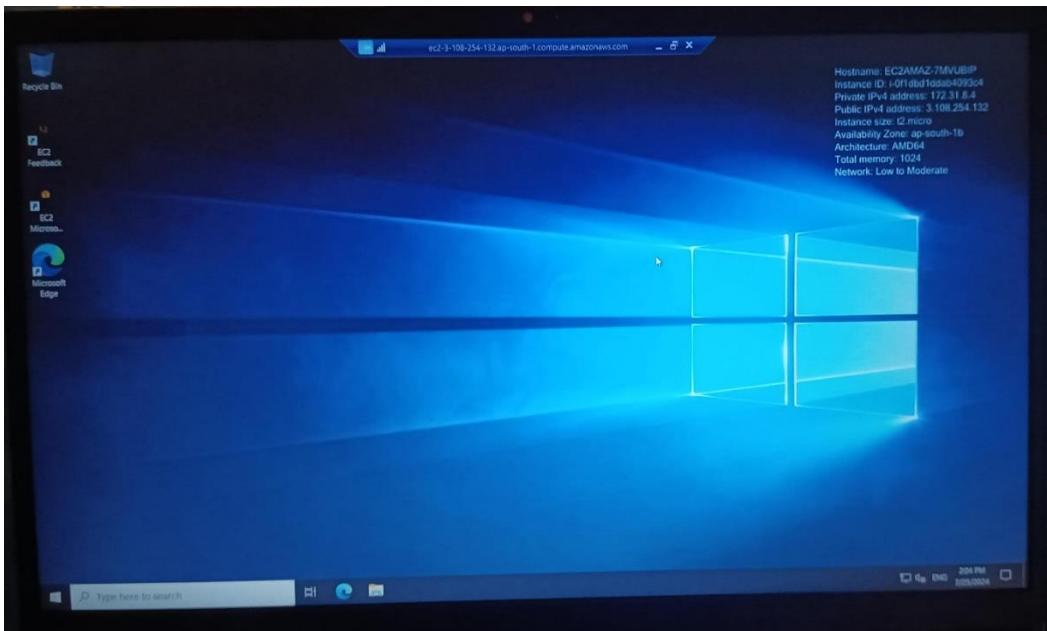
The screenshot shows the 'Launch an instance' page again, this time with a different layout. It features eight cards arranged in a grid: 'Create billing and free tier usage alerts' (with a 'Create billing alerts' button), 'Connect to your instance' (with a 'Connect to instance' button), 'Connect an RDS database' (with a 'Connect an RDS database' button), 'Create EBS snapshot policy' (with a 'Create EBS snapshot policy' button); 'Manage detailed monitoring' (with a 'Manage detailed monitoring' button), 'Create Load Balancer' (with a 'Create Load Balancer' button), 'Create AWS budget' (with a 'Create AWS budget' button), and 'Manage CloudWatch alarms' (with a 'Manage CloudWatch alarms' button). At the bottom right is a large orange 'View all instances' button.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security. The main content area displays a table titled 'Instances (1/1) Info'. It shows one instance: 'test2' (Instance ID: i-0f1dbd1ddab4093c4, State: Running, Type: t2.micro). Below the table is a detailed view for 'i-0f1dbd1ddab4093c4 (test2)', showing details like Public IPv4 address (3.108.254.132), Instance state (Running), and Private IP4 DNS (ip-172-31-8-4.ap-south-1.compute.internal).

This screenshot shows the 'Connect to instance' page for the instance 'i-0f1dbd1ddab4093c4 (test2)'. It features three tabs: Session Manager, RDP client (selected), and EC2 serial console. Under 'RDP client', it says 'Connect using RDP client' and provides a download link for the remote desktop file. It also shows the Public DNS (ec2-3-108-254-132.ap-south-1.compute.amazonaws.com) and a dropdown for the Username (Administrator). A note indicates that to connect using Fleet Manager, the SSM Agent must be installed and running on the instance.

This screenshot shows the 'Get Windows password' page for the same instance. It asks for a private key to retrieve the initial Windows administrator password. It lists the Instance ID (i-0f1dbd1ddab4093c4) and the key pair associated with it (as1). It has a section for uploading a private key file, with 'as1.pem' selected. Below is a text area for 'Private key contents - optional' containing an RSA private key. At the bottom are 'Cancel' and 'Decrypt password' buttons.





Screenshot of the AWS Cloud9 IDE interface. The left sidebar shows the AWS Services menu with 'Instances' selected. The main pane displays the EC2 Instances page with one instance named 'test2' listed. The instance details show it is running, has a t2.micro type, and is in the ap-south-1b availability zone. A modal window titled 'Stop instance?' is open, asking if the user wants to stop the instance. It includes a warning message about usage fees and a confirmation button.

Screenshot of the AWS Cloud9 IDE interface. The left sidebar shows the AWS Services menu with 'Instances' selected. The main pane displays the EC2 Instances page with one instance named 'test2' listed. The instance details show it is stopped, has a t2.micro type, and is in the ap-south-1b availability zone. A modal window titled 'Terminate instance?' is open, asking if the user wants to terminate the instance. It includes a warning message about the default action for EBS-backed instances and a confirmation button.

The screenshot shows the AWS EC2 Instances page. A green banner at the top indicates "Successfully initiated termination of i-0f1dbd1ddab4093c4". The main table lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
test2	i-0f1dbd1ddab4093c4	Terminated	t2.micro	2/2 checks passed	View alarms	ap-south-1b	-

The instance details page for "i-0f1dbd1ddab4093c4 (test2)" is shown. The "Details" tab is selected, displaying the following information:

- Instance ID:** i-0f1dbd1ddab4093c4 (test2)
- IPv6 address:** -
- Hostname type:** -
- Answer private resource DNS name:** -
- Auto-assigned IP address:** -
- Public IPv4 address:** -
- Instance state:** Terminated
- Instance type:** t2.micro
- VPC ID:** -
- Private IPv4 addresses:** -
- Public IPv4 DNS:** -
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** -

Result:

Successfully created a virtual server in the AWS public cloud using the EC2 service. The instance is now running and accessible. Screenshots are attached to demonstrate each step.

Ex2 - Attaching AWS EBS volume to Amazon EC2.

Nithin Prasath C – 22BCS081

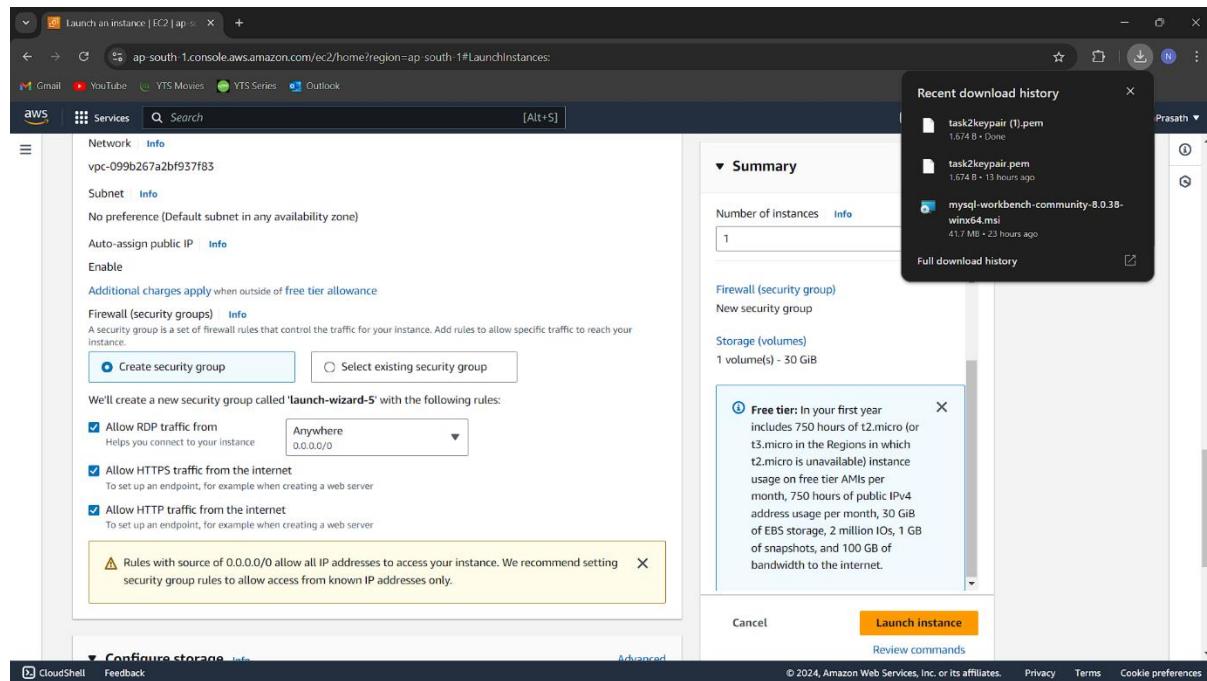
AIM:

To attach AWS EBS volume to Amazon EC2.

DESCRIPTION:

Attaching an AWS EBS volume to an EC2 instance involves creating a volume in the same availability zone, then linking it to the desired instance. This process allows the instance to utilize additional storage for data persistence. Once attached, the volume can be formatted and mounted within the instance's operating system, providing extra space for applications and data storage.

OUTPUT:



Create key pair

Key pair name
task2keypair

Key pair type
 RSA RSA encrypted private and public key pair
 ED25519 ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format
 .pem For use with OpenSSH
 .ppk For use with PuTTY

Note: When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance.

Cancel **Create key pair**

Launching instance

Launch initiation

79%

Please wait while we launch your instance.
Do not close your browser while this is loading.

Success
Successfully initiated launch of instance (i-0aebe01791fe858da)

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

1 2 3 4 5 6 >

Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
[Create billing alerts](#)

Connect to your instance
Once your instance is running, log into it from your local computer.
[Connect to instance](#) [Learn more](#)

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
[Connect an RDS database](#) [Create a new RDS database](#) [Learn more](#)

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots.
[Create EBS snapshot policy](#)

Screenshot 1: AWS EC2 Instances Overview

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store (Volumes selected), Snapshots, Lifecycle Manager, and Network & Security. The main content area displays a table for 'Instances (1/1) info'. The instance listed is 'task2' with ID 'i-031d8a90c9f63de64', which is 'Running' and has an 't2.micro' instance type. It is located in the 'ap-south-1a' availability zone and has a public IPv4 address '13.201.135.82'. The right side of the screen shows detailed information for this specific instance.

Screenshot 2: AWS Volumes Overview

The screenshot shows the AWS Volumes page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store (Volumes selected), Snapshots, Lifecycle Manager, and Network & Security. The main content area displays a table for 'Volumes (1/1) info'. The volume listed is 'vol-0fea5409063d97607', which is a 'gp2' type volume of 30 GB, attached to the 'task2' instance. It was created on Aug 01, 2024.

Screenshot 3: AWS Create Volume Form

The screenshot shows the 'Create volume' form. The top navigation bar includes 'EC2 > Volumes > Create volume'. The main section is titled 'Create volume' with an 'Info' link. It says 'Create an Amazon EBS volume to attach to any EC2 Instance in the same Availability Zone.' Below this, there's a 'Volume settings' section with fields for 'Volume type' (set to 'General Purpose SSD (gp3)'), 'Size (GiB)' (set to 100), 'IOPS' (set to 3000), and 'Throughput (MiB/s)' (set to 125). A note at the bottom states 'gp3 is now the default selection. gp3 provides up to 20% lower cost per GB than gp2.' There are also 'Learn More' and 'Create volume' buttons.

The screenshot shows the AWS Cloud9 IDE interface. A terminal window titled "Volumes | EC2 | ap-south-1" is open, displaying the message "Successfully created volume vol-0c68e69b56b1c598a." Below this, the "Volumes (1/2) info" table shows two volumes:

Snapshot ID	Created	Availability Zone	Volume state	Alarm status	Attached resources	Volume sta...
snap-01c4a28...	2024/08/01 07:29 GMT+5:...	ap-south-1a	In-use	No alarms	+ i-031d8a90c9f63de64 (task...)	Okay
-	2024/08/01 07:32 GMT+5:...	ap-south-1a	Available	No alarms	+ -	Okay

Details for the newly created volume:

Volume ID	vol-0c68e69b56b1c598a	Size	100 GiB	Type	gp3	Volume status	Okay
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations. [Learn more]	Volume state	Available	IOPS	3000	Throughput	125
Fast snapshot restored	No	Availability Zone	ap-south-1a	Created	Thu Aug 01 2024 07:32:12 GMT+0530 (India Standard Time)	Multi-Attach enabled	No
Snapshots	-	Attached resources	-	Outposts ARN	-	-	-

The screenshot shows the AWS Cloud9 IDE interface. A modal dialog titled "Attach volume" is open. It displays the "Basic details" section with the following information:

- Volume ID: vol-0c68e69b56b1c598a
- Availability Zone: ap-south-1a
- Instance: i-031d8a90c9f63de64
- Device name: xvdf

At the bottom right of the dialog are the "Cancel" and "Attach volume" buttons.

The screenshot shows the AWS Cloud9 IDE interface. A terminal window titled "Volumes | EC2 | ap-south-1" is open, displaying the message "Successfully attached volume vol-0c68e69b56b1c598a to instance i-031d8a90c9f63de64." Below this, the "Volumes (1/2) info" table shows the attached volume:

Snapshot ID	Created	Availability Zone	Volume state	Alarm status	Attached resources	Volume sta...
snap-01c4a28...	2024/08/01 07:29 GMT+5:...	ap-south-1a	In-use	No alarms	+ i-031d8a90c9f63de64 (task...)	Okay
-	2024/08/01 07:32 GMT+5:...	ap-south-1a	In-use	No alarms	+ i-031d8a90c9f63de64 (task...)	Okay

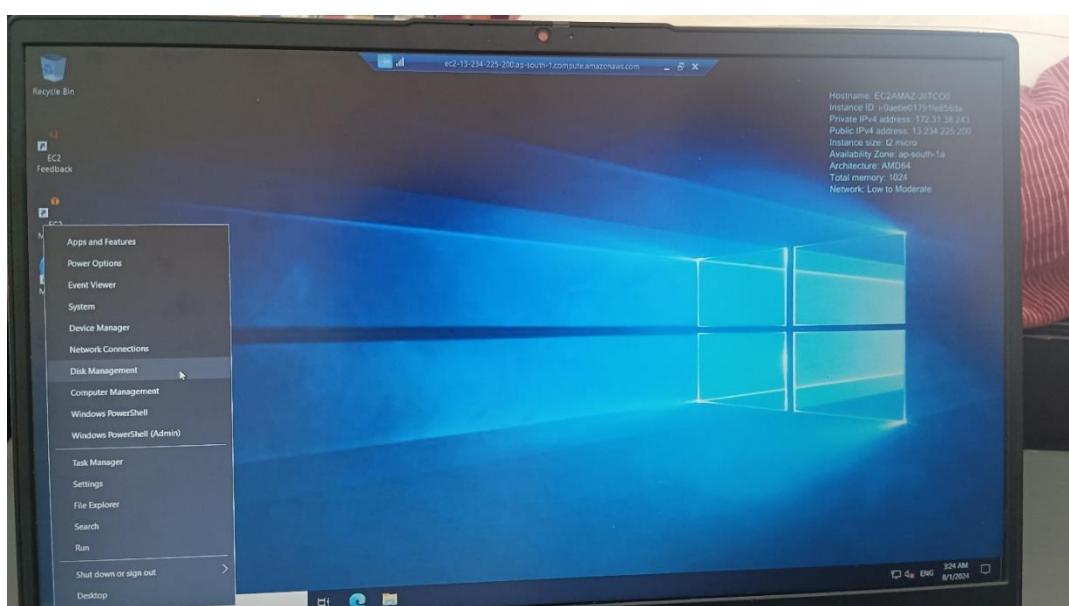
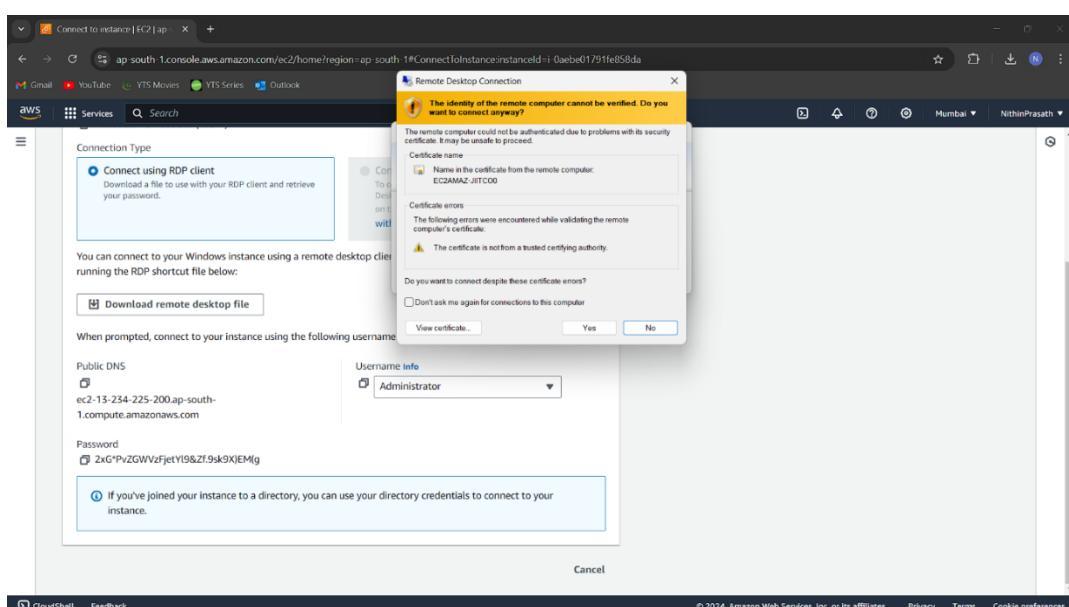
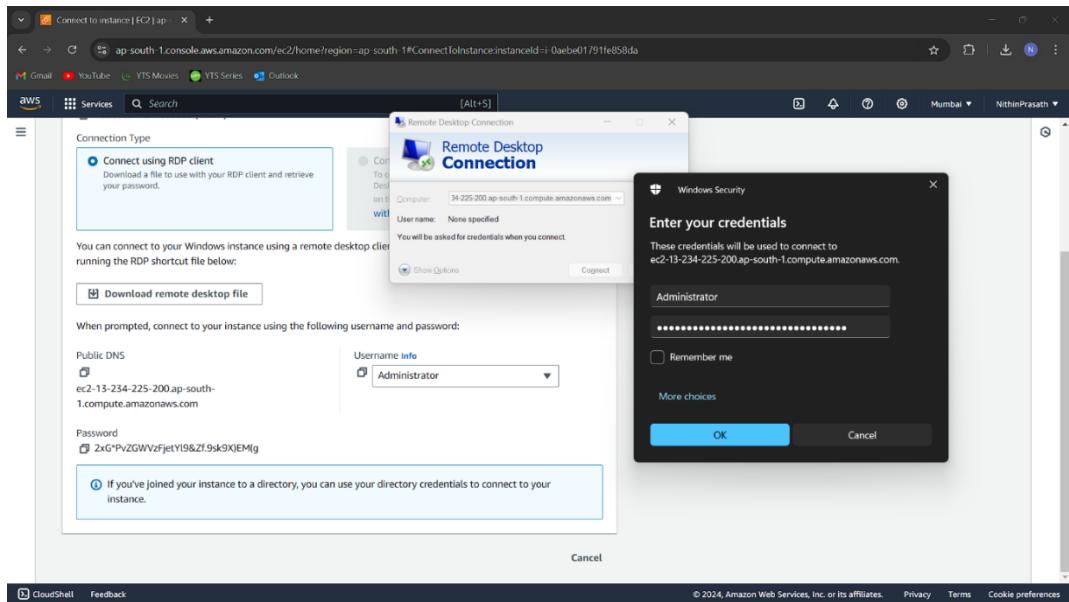
Details for the attached volume:

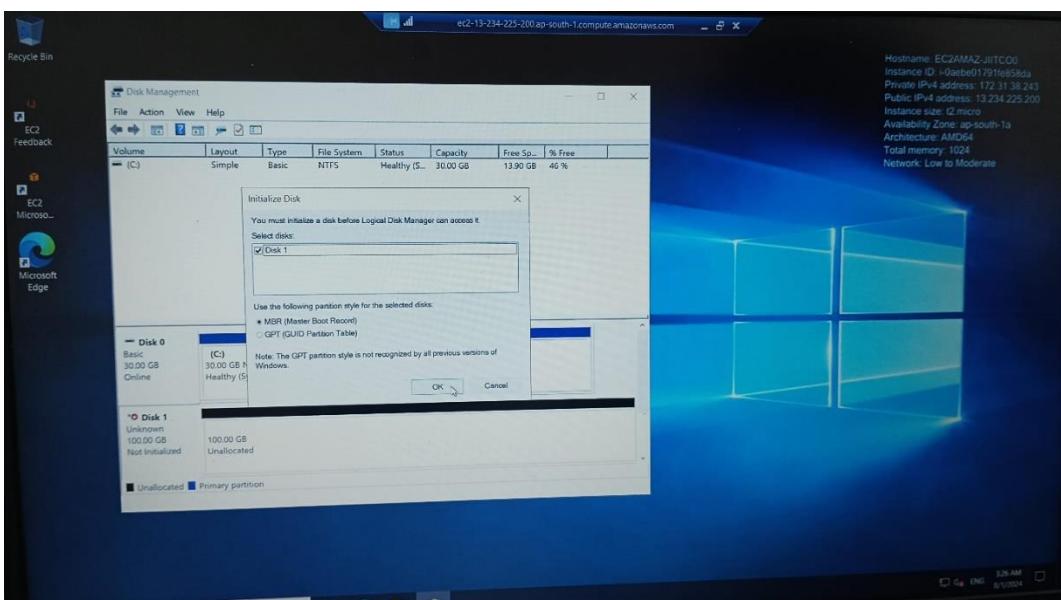
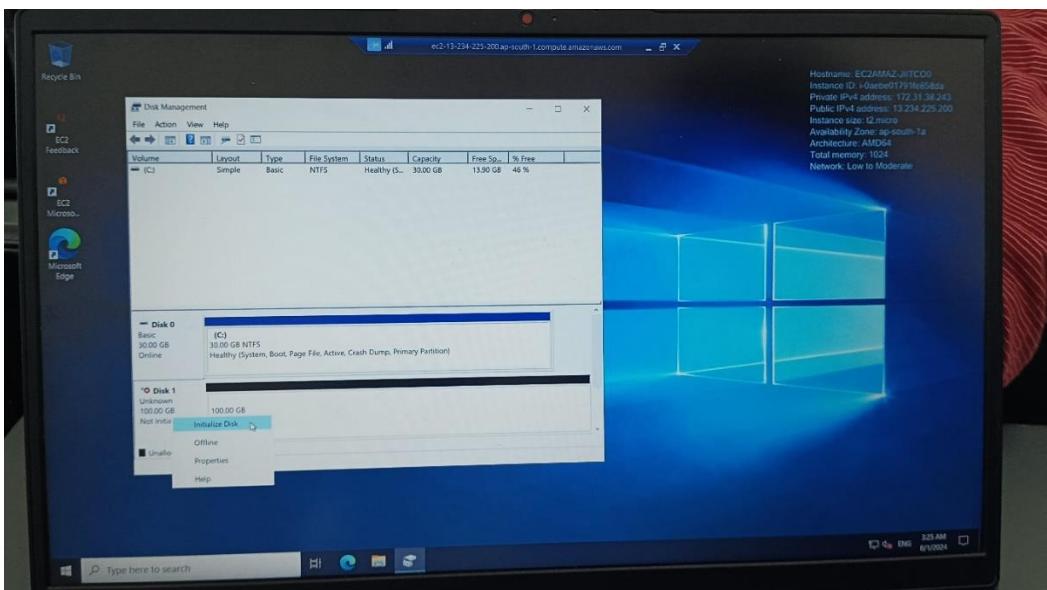
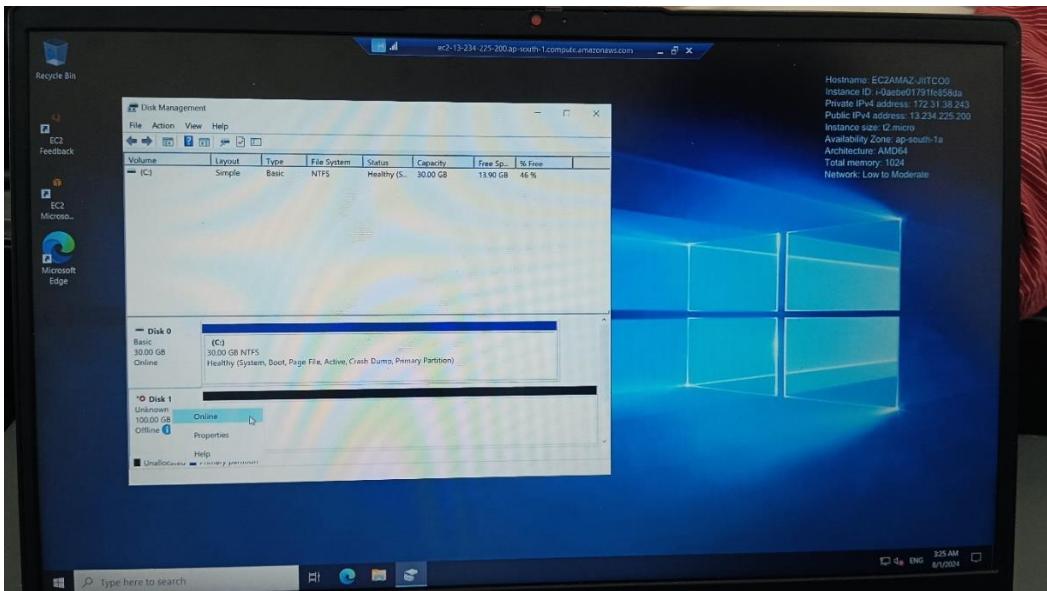
Volume ID	vol-0c68e69b56b1c598a	Size	100 GiB	Type	gp3	Volume status	Okay
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations. [Learn more]	Volume state	In-use	IOPS	3000	Throughput	125
Fast snapshot restored	No	Availability Zone	ap-south-1a	Created	Thu Aug 01 2024 07:32:12 GMT+0530 (India Standard Time)	Multi-Attach enabled	No
Snapshots	-	Attached resources	i-031d8a90c9f63de64 (task2): xvdf (attaching)	Outposts ARN	-	-	-

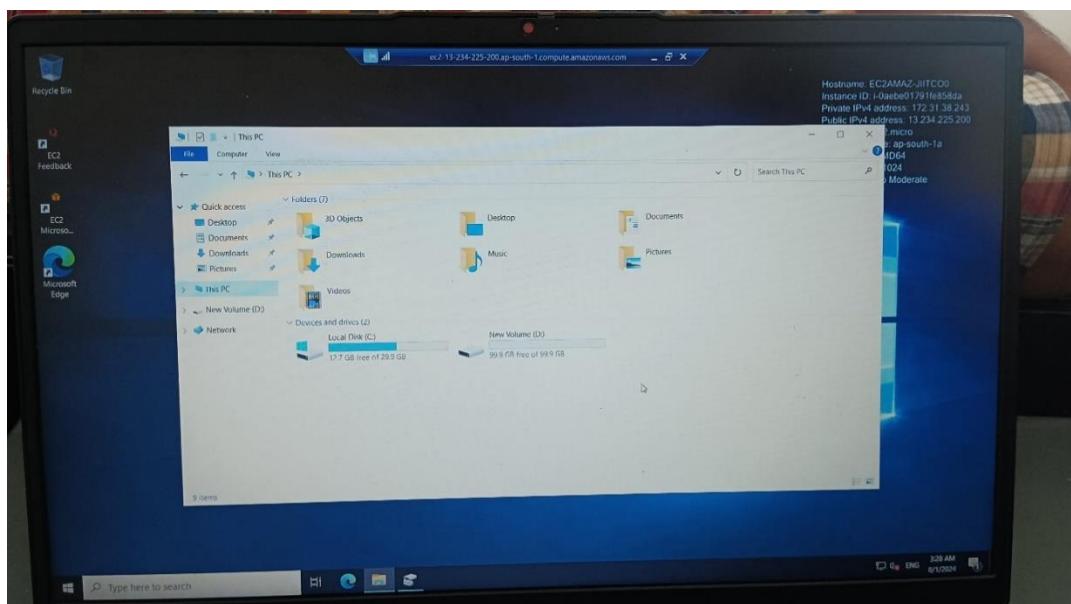
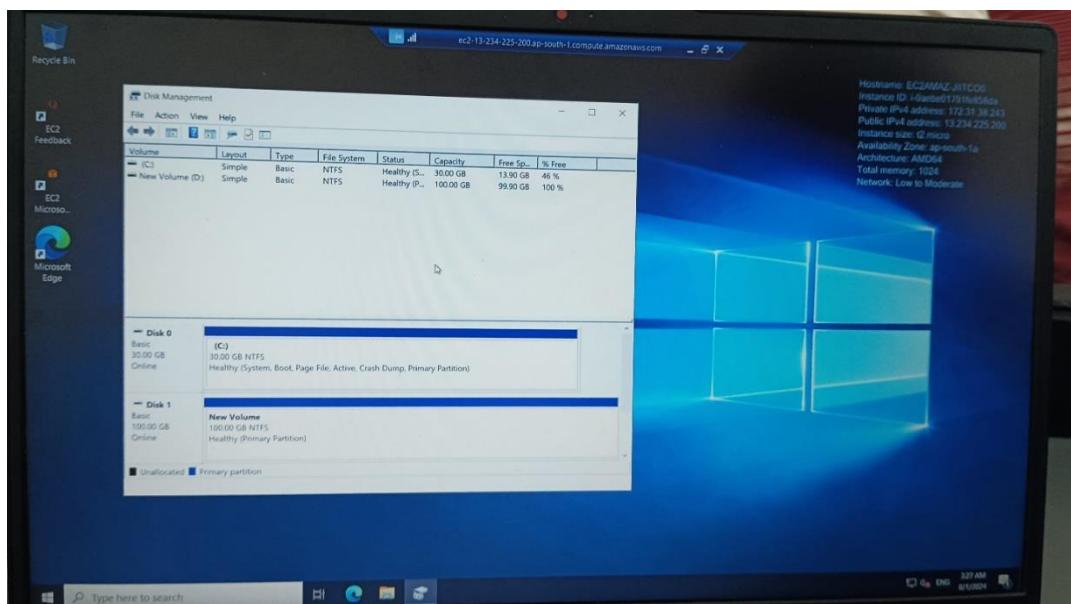
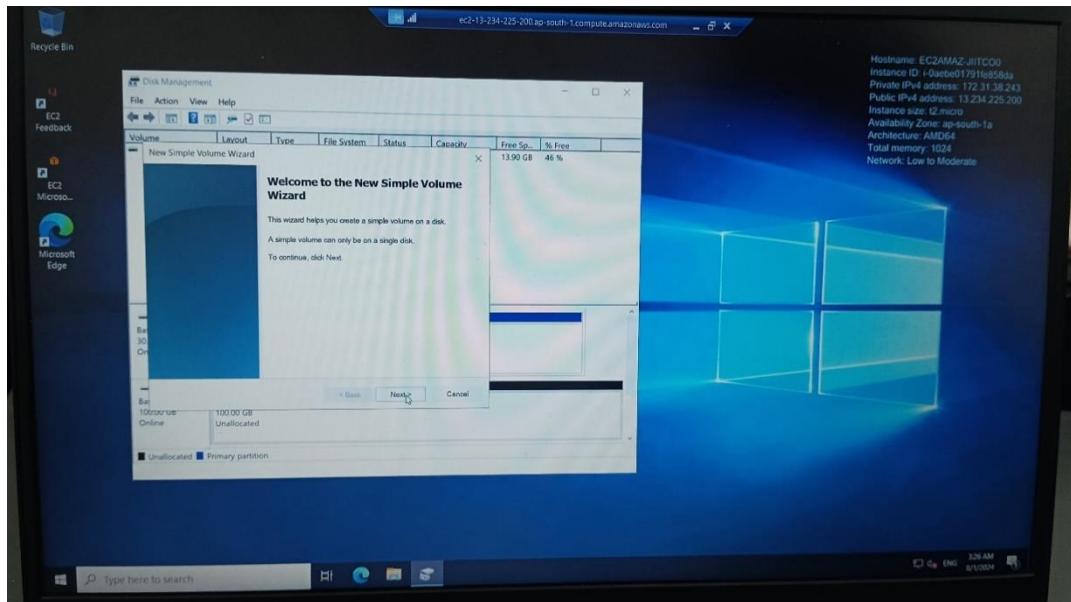
The screenshot shows the AWS EC2 Connect to instance page. It displays two connection methods: 'Connect using RDP client' (selected) and 'Connect using Fleet Manager'. Below these are instructions for connecting using a remote desktop client and a download link for the RDP shortcut file. A note indicates that if the instance is joined to a directory, directory credentials can be used. On the right, there's a 'Username info' dropdown set to 'Administrator'. At the bottom, there are 'Password' and 'Get password' buttons, and a note about joined instances.

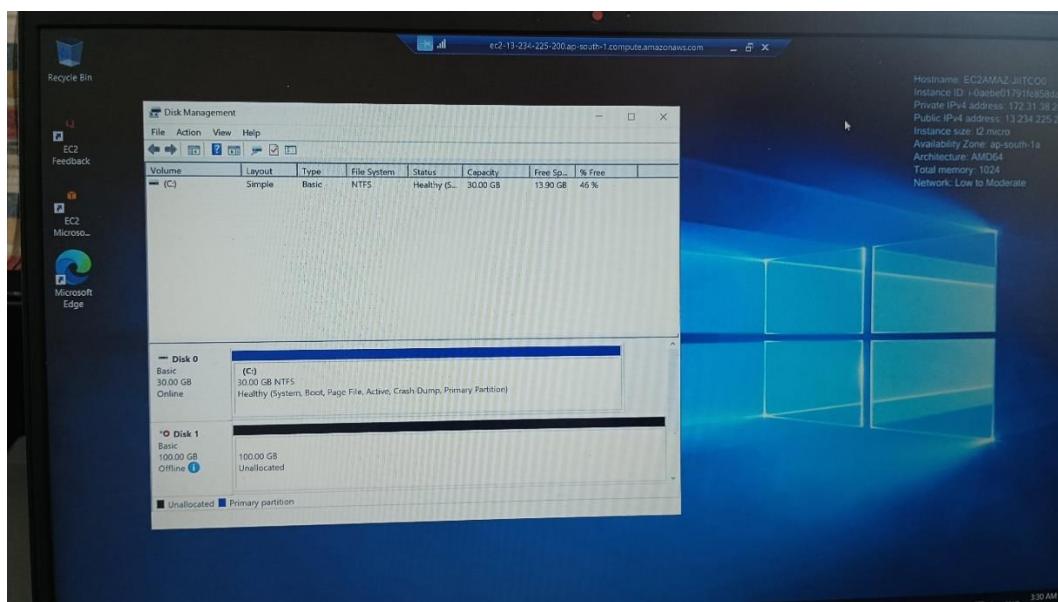
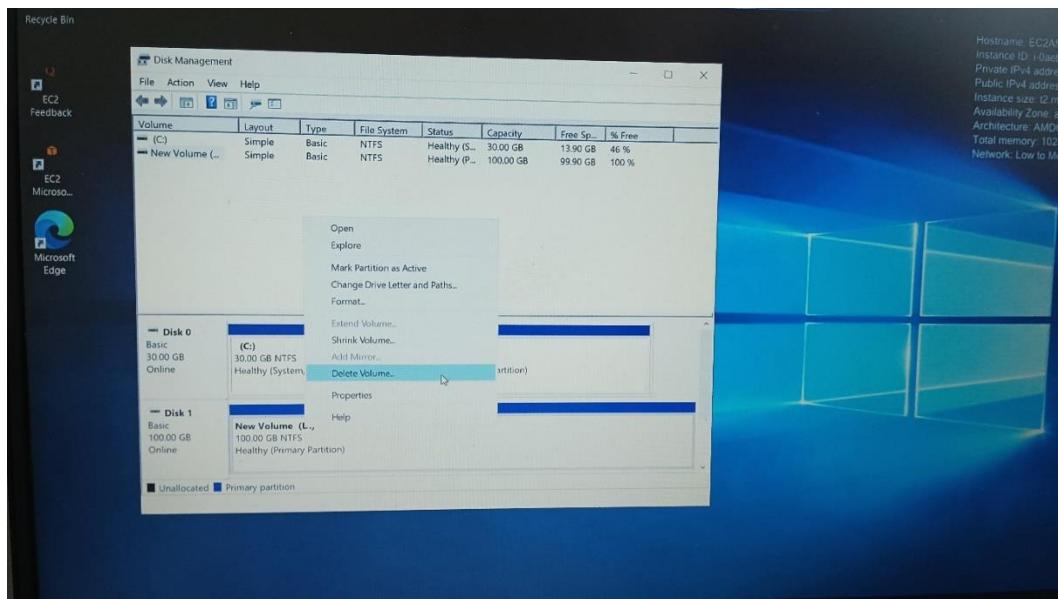
The screenshot shows the 'Get Windows password' page. It asks for a private key to decrypt the initial Windows administrator password. It lists the instance ID (i-0aebe01791fe858da), the key pair (task2keypair), and the private key file (task2keypair (1).pem). The private key content is displayed as a large block of encrypted text. At the bottom, there are 'Cancel' and 'Decrypt password' buttons.

This screenshot is identical to the one at the top of the page, showing the 'Connect to instance' page with the 'Connect using RDP client' option selected. It includes the RDP connection details, the 'Username info' dropdown set to 'Administrator', and the note about joined instances.









A screenshot of the AWS CloudShell interface. The main window shows the AWS Management Console with the 'Volumes (1/2)' page. A modal dialog box titled 'Detach vol-026e3d7055a9f3dea?' is open, containing the message: 'After you detach a volume, you might still be charged for volume storage. If you no longer need the volume, delete it to stop incurring charges.' Below this, there is a question: 'Are you sure that you want to detach volume vol-026e3d7055a9f3dea?'. At the bottom of the dialog are 'Cancel' and 'Detach' buttons. The status bar at the bottom right indicates 'Volume status: Okay'.

The image consists of three vertically stacked screenshots from the AWS Cloud Console, illustrating the steps to delete an EBS volume and stop an EC2 instance.

Screenshot 1: Successfully detached volume.

This screenshot shows the EBS Volumes page. A confirmation message at the top states "Successfully detached volume." Below is a table of volumes:

Type	Size	IOPS	Throughput	Snapshot ID	Created	Availability Zone	Volume state	Alarm status
gp2	30 GiB	100	-	snap-01c4a28...	2024/08/01 08:46 GMT+5...	ap-south-1a	In-use	No alarms
gp3	100 GiB	3000	125	-	2024/08/01 08:52 GMT+5...	ap-south-1a	Available	No alarms

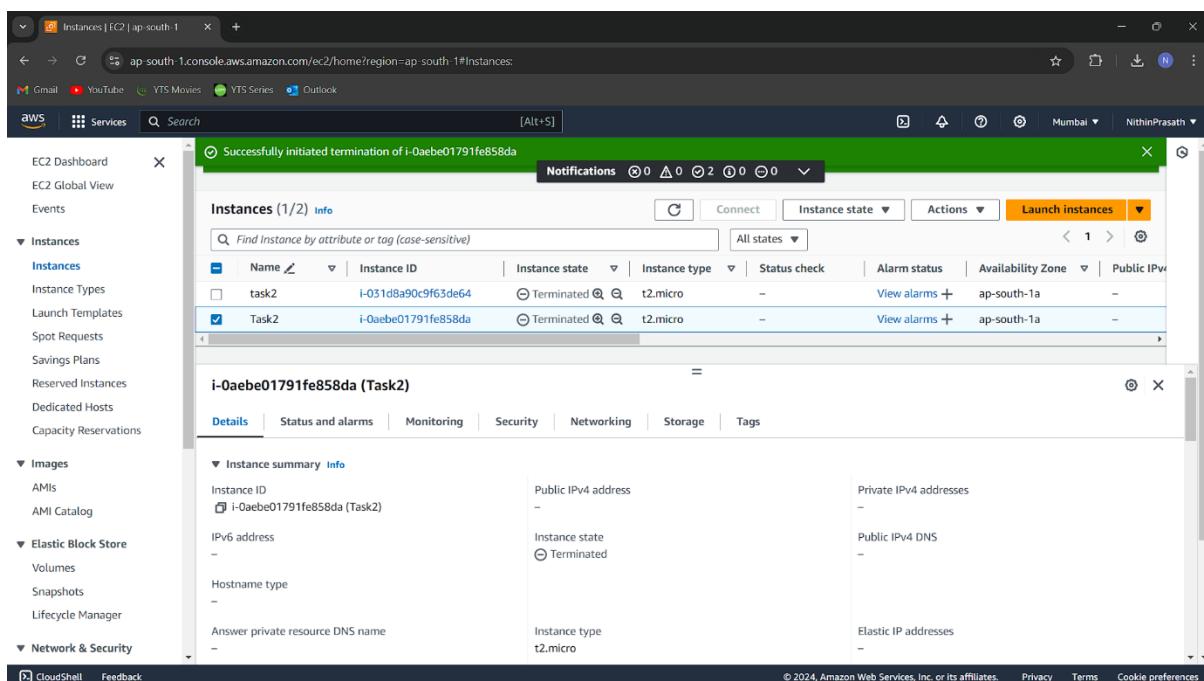
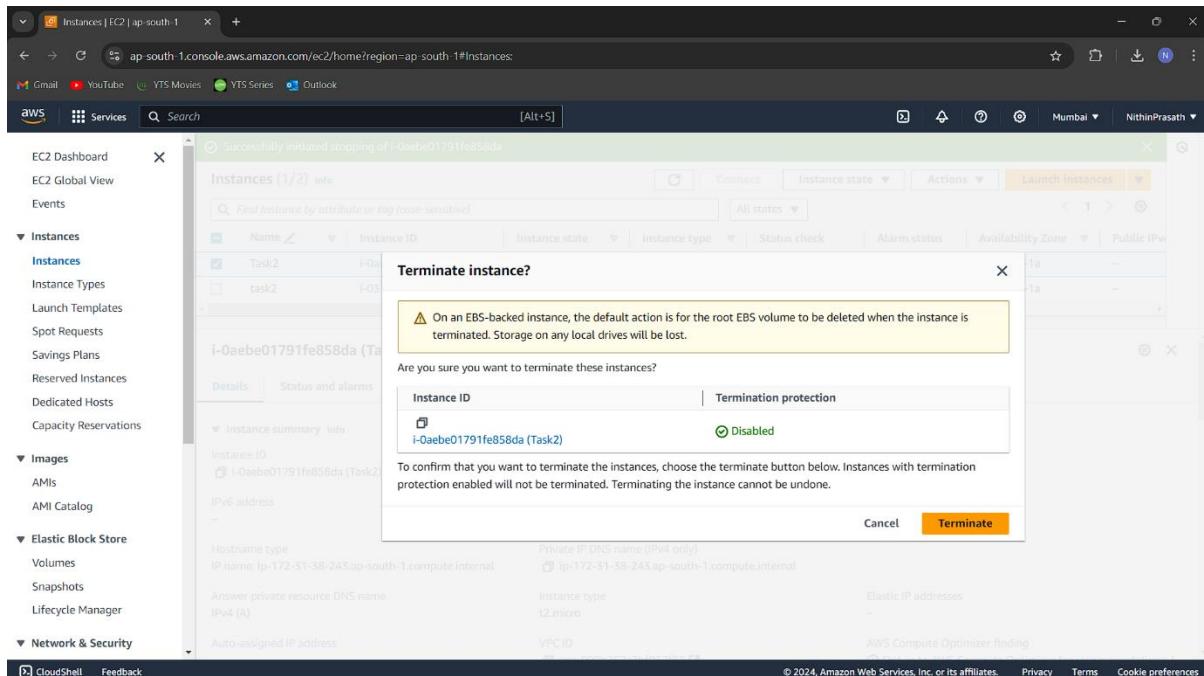
A modal window titled "Fault tolerance for all volumes in this Region" is open, showing "Snapshot summary" with "0 / 1" recently backed up volumes.

Screenshot 2: Delete vol-026e3d7055a9f3dea?

This screenshot shows the details of the volume "vol-026e3d7055a9f3dea". A confirmation dialog box asks "Are you sure that you want to delete vol-026e3d7055a9f3dea? To confirm deletion, type delete in the field." The user has typed "delete" into the input field.

Screenshot 3: Stop instance?

This screenshot shows the EC2 Instances page. A confirmation dialog box asks "To confirm that you want to stop the instance, choose the Stop button below." The "Stop" button is highlighted in orange.



Result:

Successfully attached AWS EBS volume to Amazon EC2. The instance is now running and accessible. Screenshots are attached to demonstrate each step.

Ex3 - Hosting Web application in cloud

Nithin Prasath C – 22BCS081

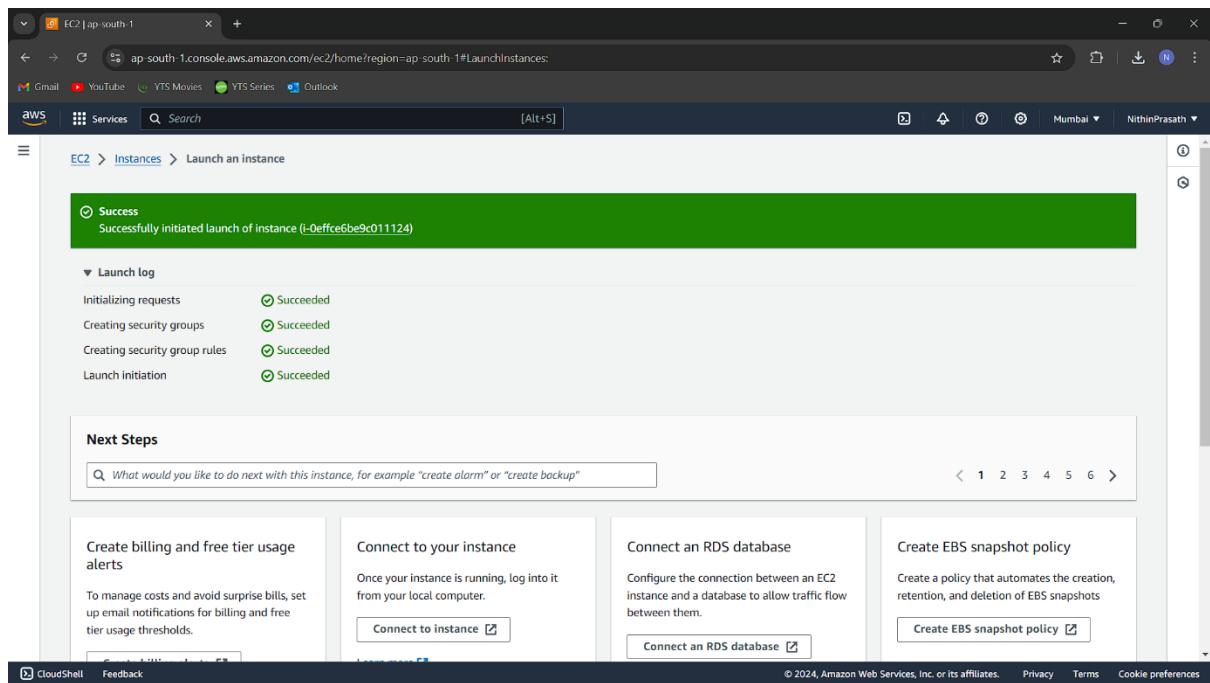
AIM:

To Create a web application in virtual Machine and hosting it in a host machine using the public IP and decrypted password which was created while launching the instance in host system.

DESCRIPTION:

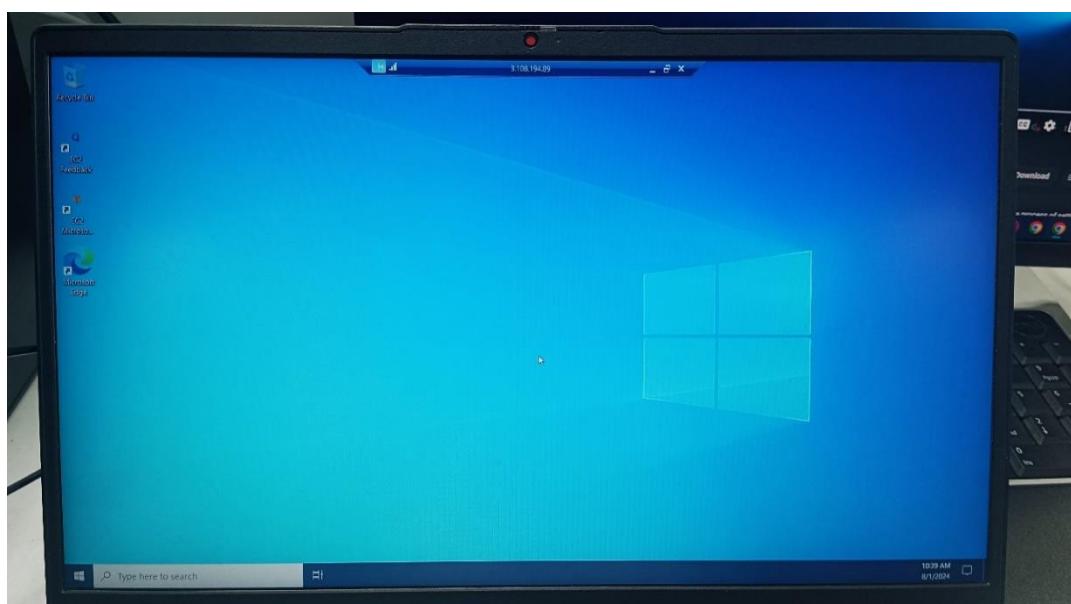
To host a web application on Amazon EC2, create an AWS account, set up an EC2 instance, and install the necessary web server software and dependencies. Deploy your application files to the instance and update the DNS settings to point your domain to the EC2 instance's public IP address.

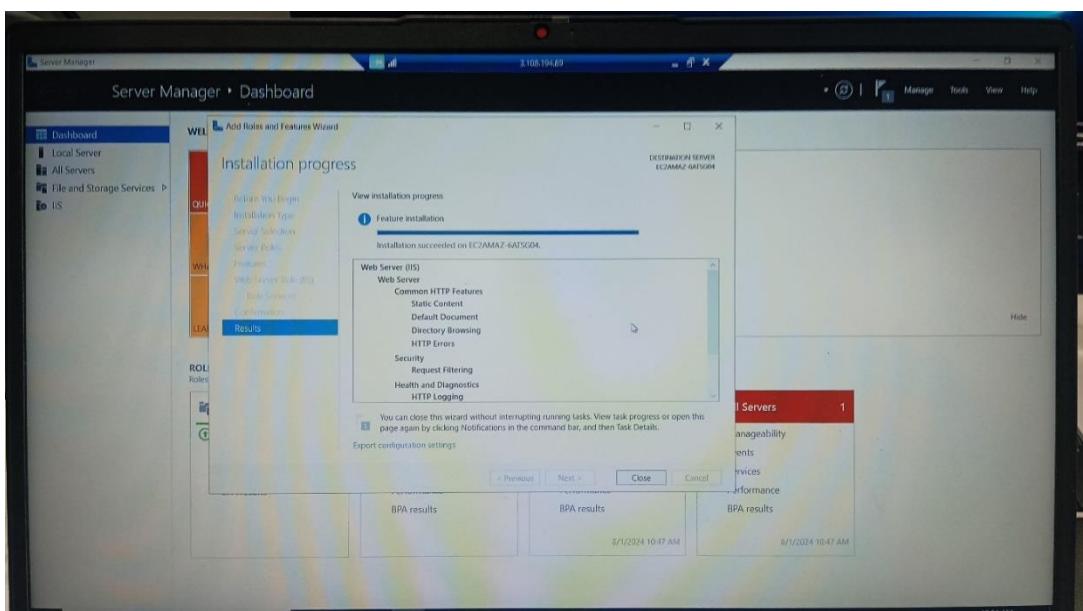
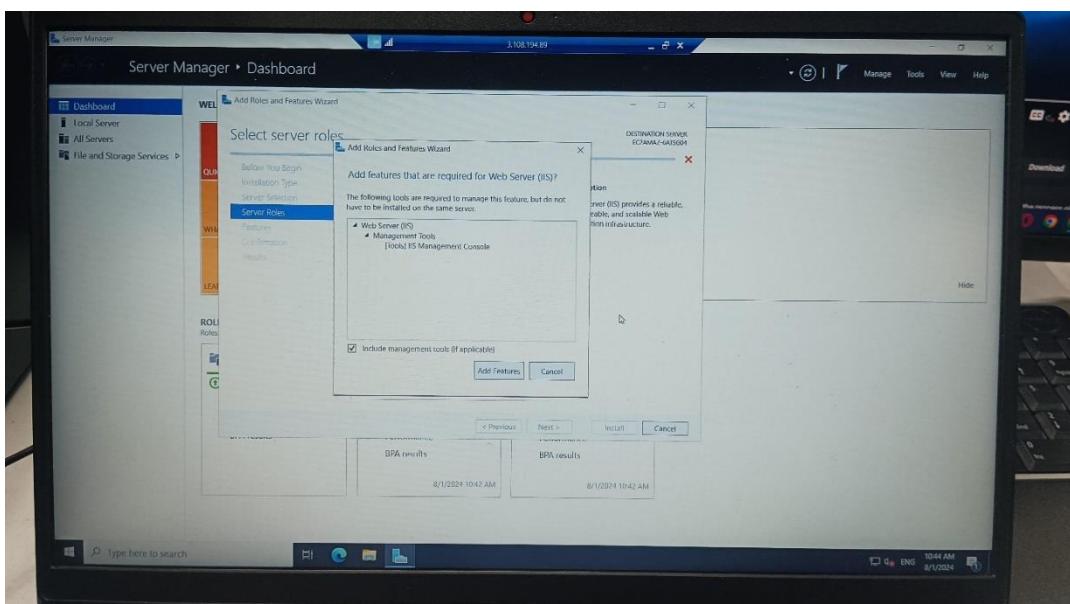
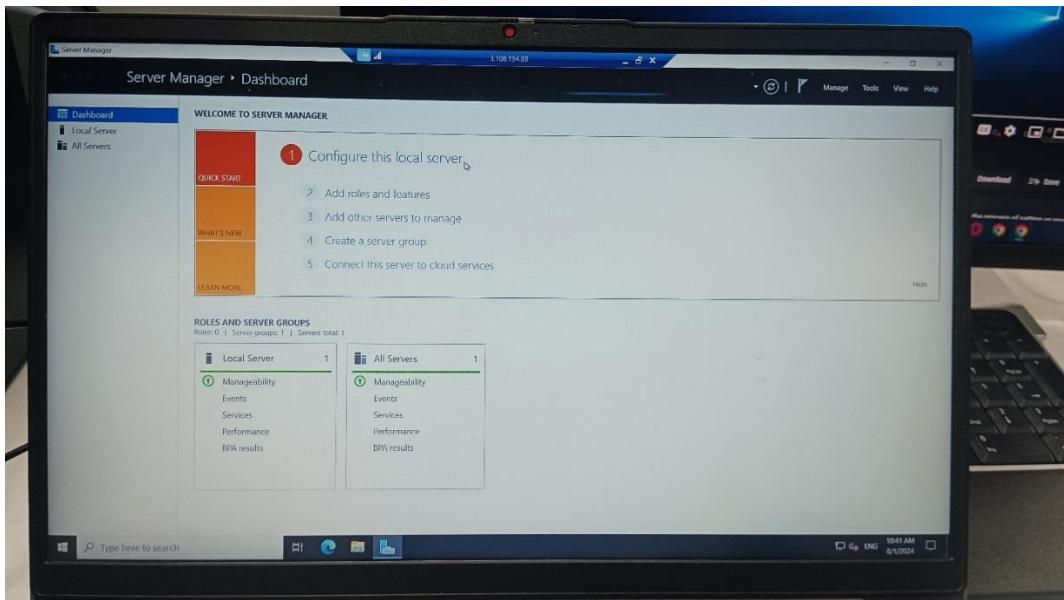
OUTPUT:

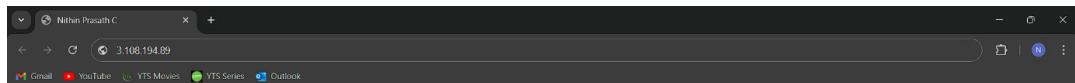


The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, Services, and Search. The main area displays a table of instances. One instance is selected, labeled "Task3" with the ID "i-0effce6be9c011124". The instance is shown as "Running" with the type "t2.micro" and status "Initializing". It has a Public IPv4 DNS of "ec2-3-108-194-89.ap-south-1.compute.amazonaws.com" and a Public IPv4 address of "3.108.194.89". A "Details" tab is open, showing more instance details such as Instance ID, Instance state, IP address, Hostname type, and Instance type.

This screenshot shows the "Connect to instance" page for the same EC2 instance. A "Remote Desktop Connection" dialog box is open, indicating it is "Configuring remote session". Below the dialog, there's a "Connection Type" section with a radio button for "Connect using RDP client". A note says you can download an RDP file. The "Public DNS" field contains "ec2-3-108-194-89.ap-south-1.compute.amazonaws.com". A "Username copied" message is displayed above the "Administrator" dropdown. A note at the bottom says you can use directory credentials if joined to a domain.

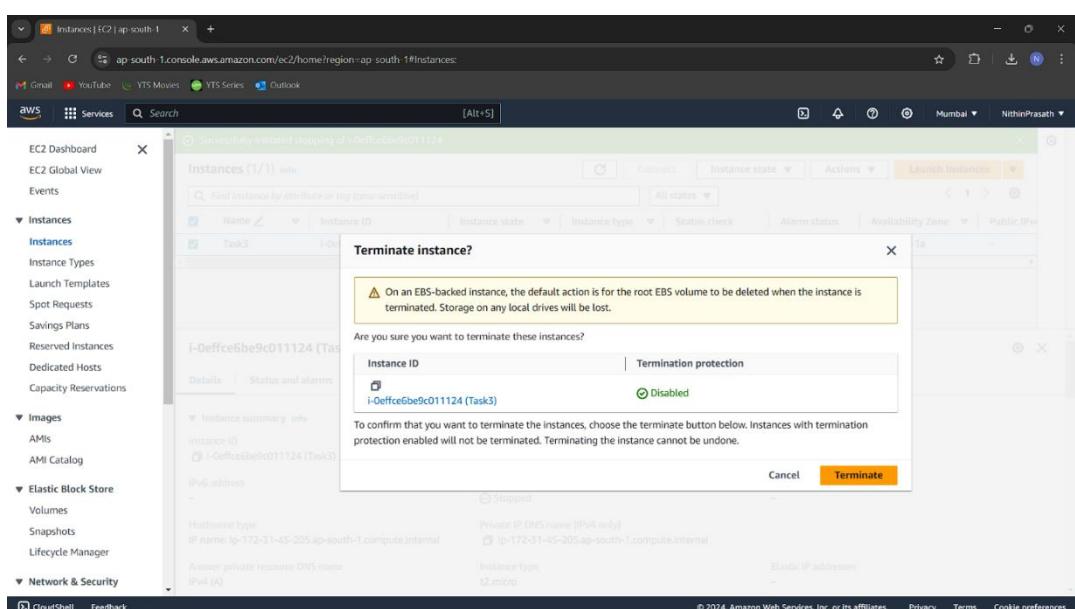
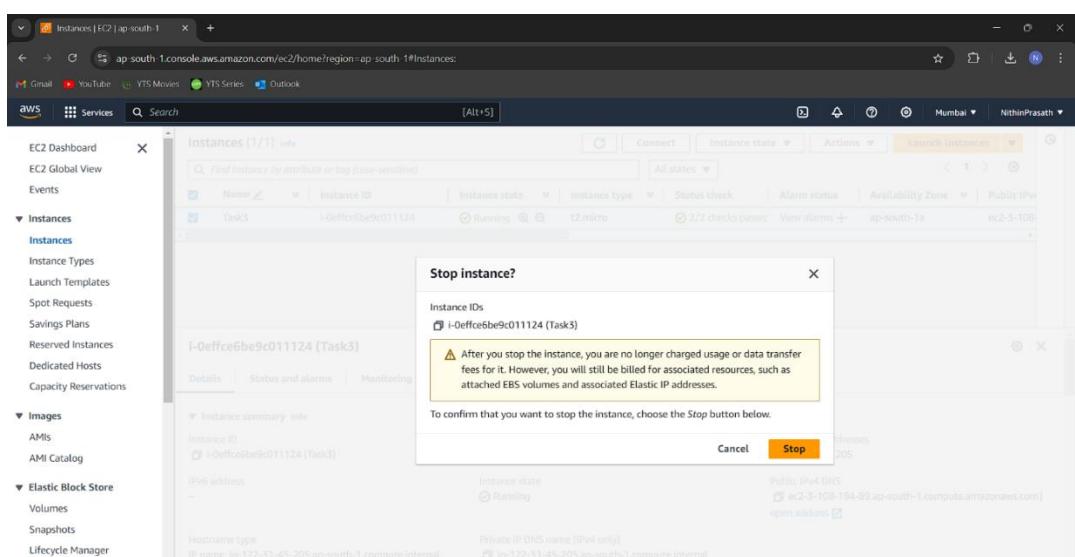






Nithin Prasath C - 22BCS081

Ex 3 : Hosting Web application in cloud



The screenshot shows the AWS EC2 Instances page. A green notification bar at the top reads "Successfully initiated termination of i-0effce6be9c011124". The main table lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Task3	i-0effce6be9c011124	Terminated	t2.micro	2/2 checks passed	View alarms	ap-south-1a	-

Below the table, the instance details for "i-0effce6be9c011124 (Task3)" are shown. The "Details" tab is selected, displaying the following information:

Instance summary	Info
Instance ID	i-0effce6be9c011124 (Task3)
IPv6 address	-
Hostname type	-
Answer private resource DNS name	-
Public IPv4 address	-
Instance state	Terminated
Instance type	t2.micro
Private IPv4 addresses	-
Public IPv4 DNS	-
Elastic IP addresses	-

RESULT:

Successfully hosted web application in cloud to Amazon EC2. The instance is now running and accessible. Screenshots are attached to demonstrate each step.

Ex4 - Attaching additional virtual servers to existing application

Nithin Prasath C – 22BCS081

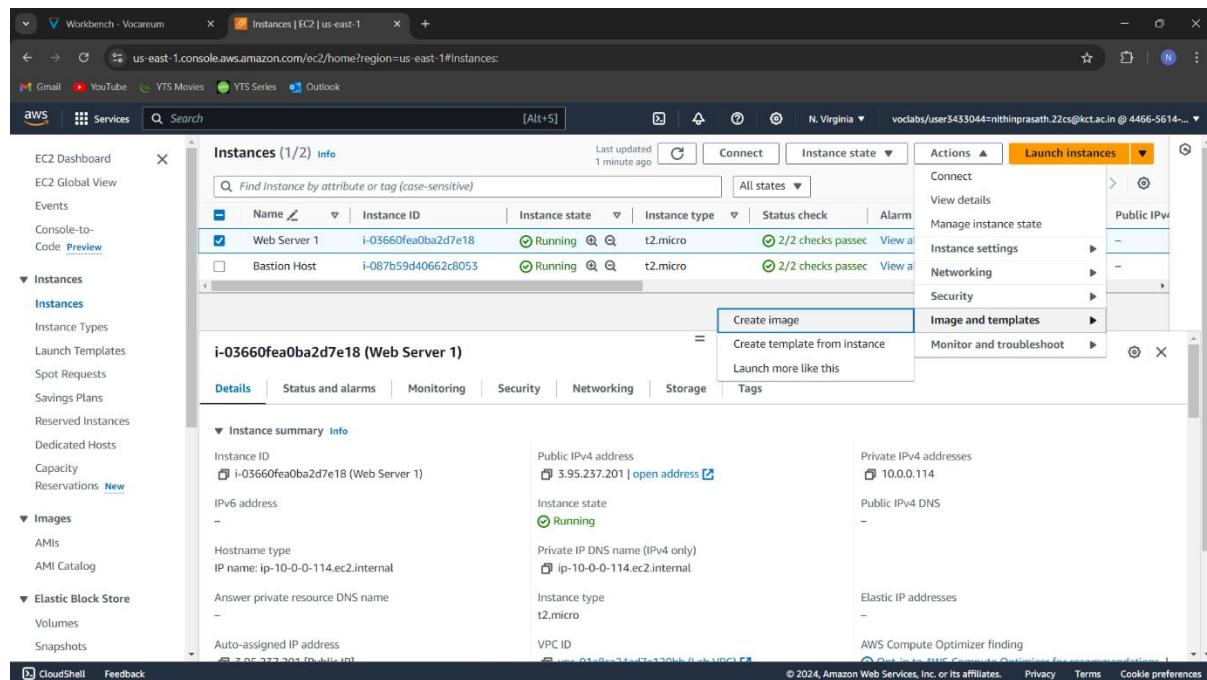
AIM:

To attach additional virtual servers to an existing application in a WebSphere Application Server (WAS) load balancer, enhancing scalability and availability.

DESCRIPTION:

The task involves configuring and adding new virtual servers to distribute application traffic across multiple servers using a load balancer, ensuring efficient load distribution and improved system performance.

OUTPUT:



The image consists of three vertically stacked screenshots from the AWS EC2 console.

Screenshot 1: Create Image (Step 1)

This screenshot shows the "Create image" configuration page. The instance ID is i-03660fea0ba2d7e18 (Web Server 1). The image name is "WebServerAMI". The image description is "Lab AMI for Web Server". The "Reboot instance" checkbox is checked. Under "Instance volumes", there is one EBS volume attached. The volume is an EBS General Purpose (SSD) volume with a size of 3000 GiB, IOPS of 1000, Throughput of 3000 MiB/s, Delete on termination, and Encrypted.

Screenshot 2: Step 1 Create target group (Step 1)

This screenshot shows the "Specify group details" step of creating a target group. It asks to "Specify group details" for a load balancer. Under "Basic configuration", the "Instances" target type is selected. It lists benefits such as supporting load balancing to instances within a specific VPC and facilitating the use of Amazon EC2 Auto Scaling. Other options like "IP addresses" and "Lambda function" are also shown.

Screenshot 3: Step 2 Create target group (Step 2)

This screenshot shows the "Register targets" step. It lists "Available instances (2)" in the "Instances" section. Two instances are listed: "Bastion Host" (Instance ID i-087b59d40662c8053) and "Web Server 1" (Instance ID i-03660fea0ba2d7e18). Both are marked as "Running" and belong to the "Web Security Group". Below the table, it says "0 selected" and "Ports for the selected instances". A dropdown menu shows "80" and "1-65535 (separate multiple ports with comma)". A button "Include as pending below" is visible.

Screenshot 1: Target group details (EC2)

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
0	0	0	0	0	0
	0 Anomalous				

Screenshot 2: Load balancers (EC2)

Screenshot 3: Select Create LB Wizard

Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.	Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.	Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.
Create	Create	Create

Screenshot 1: Create Application Load Balancer

The screenshot shows the 'Create Application Load Balancer' wizard. The 'Basic configuration' step is selected. A 'Load balancer name' field contains 'LabELB'. The 'Scheme' dropdown is set to 'Internet-facing'. Under 'Load balancer IP address type', 'IPv4' is selected. The status bar at the bottom indicates: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Screenshot 2: Load balancer details | LabELB

This screenshot shows the 'LabELB' load balancer details page. It lists the VPC (VPC-01a8ca24ad7e129bb), Hosted zone (Z355XDOTRQ7X7K), and Availability Zones (subnet-077de699085a2a43b, subnet-06f1c68beef4209b). The DNS name is LabELB-456857120.us-east-1.elb.amazonaws.com. The status is 'Provisioning'. The date created was September 16, 2024, at 22:39 (UTC+05:30). The 'Listeners and rules' tab is active.

Screenshot 3: EC2 launch templates

This screenshot shows the 'EC2 launch templates' page. It features a large heading 'EC2 launch templates' and sub-headings 'Streamline, simplify and standardize instance launches'. A call-to-action button 'Create launch template' is visible. The left sidebar includes 'Launch Templates' under the 'Compute' section. The status bar at the bottom indicates: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Step 5 - optional

Add notifications

Step 6 - optional

Add tags

Step 7

Review

Network info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-01a6ca24ad7e129bb (Lab VPC)
10.0.0.0/16

Create a VPC

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets
us-east-1a | subnet-087755c7326e5dd8 (Private Subnet 1)
10.0.1.0/24

us-east-1b | subnet-0bd9df015b5eb75d4 (Private Subnet 2)
10.0.3.0/24

Create a subnet

Cancel Skip to review Previous Next

Step 2

Choose instance launch options

Step 3 - optional

Configure advanced options

Step 4 - optional

Configure group size and scaling

Step 5 - optional

Add notifications

Step 6 - optional

Add tags

Step 7

Review

Load balancing

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

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
LabGroup | HTTP Application Load Balancer: LabELB

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Automatic scaling - optional

Choose whether to use a target tracking policy [Info](#)
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name
LabScalingPolicy

Metric type [Info](#)
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.
Average CPU utilization

Target value
60

Instance warmup [Info](#)
300 seconds

Disable scale-in to create only a scale-out policy

Instance maintenance policy [Info](#)
Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The image consists of three vertically stacked screenshots from the AWS EC2 console, all sharing the same browser tab bar at the top.

- Top Screenshot:** Shows the "Auto Scaling groups" page. It displays one group named "Lab Auto Scaling Group". The group has a status of "Updating capacity..." and is set to a minimum of 2, maximum of 6, and availability zone us-east-1a. A green banner at the top indicates "Lab Auto Scaling Group, 1 Scaling policy created successfully. Group metrics collection is enabled."
- Middle Screenshot:** Shows the "Target groups" page. It displays one target group named "LabGroup". The target group is associated with the ARN "arn:aws:elasticloadbalancing:us-east-1:456857120.us-east-1:targetgroup/LabGroup" and port 80, using the HTTP protocol. The target type is "Instance" and no load balancer is associated.
- Bottom Screenshot:** Shows the "Load balancers" page. It displays one load balancer named "LabELB". The load balancer is active, using the VPC "vpc-01a8ca24ad7e129bb" and has two availability zones: "us-east-1b (use1-az2)" and "us-east-1a (use1-az2)". The scheme is "Internet-facing" and the hosted zone is "Z355XDOTRQ7X7K". The ARN is "arn:aws:elasticloadbalancing:us-east-1:446656146075:loadbalancer/app/LabELB/cb57febedf61bb51". The DNS name is "LabELB-456857120.us-east-1.elb.amazonaws.com (A Record)".

CloudWatch > Alarms

Alarms (2)

Name	State	Last state update (UTC)	Conditions	Actions
TargetTracking-Lab Auto Scaling Group-AlarmHigh-0ea0709e-73f2-49fb-8157-debd0d604a	OK	2024-09-16 17:25:47	CPUUtilization > 60 for 3 datapoints within 3 minutes	Actions enabled
TargetTracking-Lab Auto Scaling Group-AlarmLow-412c1183-99e1-46f2-b923-3f5098a15945	Insufficient data	2024-09-16 17:23:50	CPUUtilization < 54 for 15 datapoints within 15 minutes	Actions enabled

EC2 > Auto Scaling groups

Auto Scaling groups (1/1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
Lab Auto Scaling Group	LabConfig Version Default	2	-	2	2

Auto Scaling group: Lab Auto Scaling Group

Dynamic scaling policies (1/2) Info

Actions	Create dynamic scaling policy
Enable	< 1 >
Disable	
Execute	

LabScalingPolicy

Policy type: Target tracking scaling
Enabled or disabled: Enabled
Execute policy when:

CloudWatch > Alarms

Alarms (1/2)

Name	State	Last state update (UTC)	Conditions	Actions
TargetTracking-Lab Auto Scaling Group-AlarmLow-949ee0a-14c1-44ad-b625-83ce7bd435e9	In alarm	2024-09-16 17:33:50	CPUUtilization < 37.5 for 15 datapoints within 15 minutes	Actions enabled
TargetTracking-Lab Auto Scaling Group-AlarmHigh-241cef92-58a6-47ea-9284-57...	In alarm	2024-09-16 17:33:50	CPUUtilization > 50 for 3 datapoints within 3 minutes	Actions enabled

The screenshot shows the AWS CloudFront console with the URL <https://us-east-1.console.aws.amazon.com/cloudfront/home?region=us-east-1#distribution:>. The distribution is labeled "labelb-456857120.us-east-1.elb.amazonaws.com". The "CloudFront Test" section displays the response from the EC2 instance, showing the instance ID and availability zone.

Meta-Data	Value
InstanceId	i-02951acccf42a4716
Availability Zone	us-east-1a

Current CPU Load: 0%

RESULT:

Successfully attached the virtual servers, allowing the load balancer to distribute requests across the newly added servers, leading to better resource utilization and higher application availability.

Ex5 - Host a static webpage

Nithin Prasath C – 22BCS081

AIM:

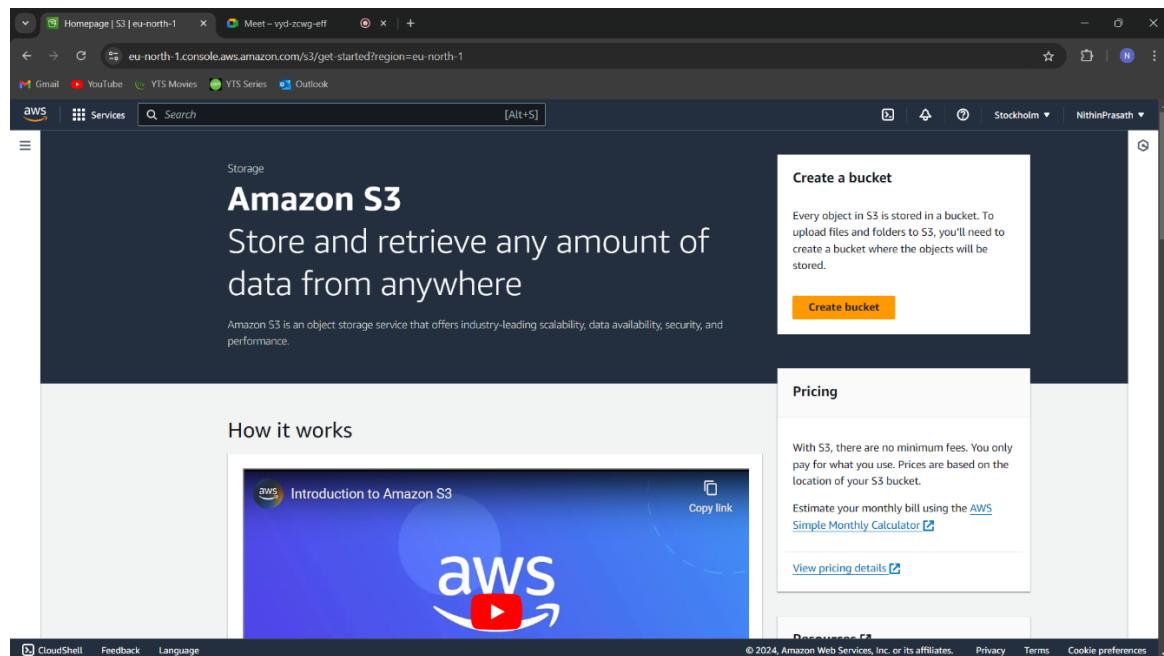
To host a static webpage on an Amazon S3 bucket, making the website publicly accessible via the web, using AWS's scalable storage infrastructure.

DESCRIPTION:

The process involves several steps:

1. Create a static webpage (HTML, CSS, JavaScript, etc.) that will be hosted on AWS S3.
2. Log in to the AWS Management Console and create an S3 bucket, ensuring the bucket name is globally unique.
3. Upload all the necessary website files (HTML, CSS, images, etc.) into the S3 bucket.
4. Configure the S3 bucket for static website hosting by enabling the "Static Website Hosting" feature under the bucket properties and specifying the index and error documents (e.g., index.html and error.html).
5. Adjust the bucket's permissions to make it publicly accessible, allowing users to visit the website via the S3 bucket's endpoint URL.
6. Optionally, configure custom domain hosting by linking the S3 bucket to a domain using Amazon Route 53 or an external DNS provider.

OUTPUT:



The screenshots show the 'Create bucket' wizard in the AWS Management Console. The first step, 'General configuration', shows the bucket name 'myawsbucket_nithin' and the choice of bucket type: 'General purpose' (selected) or 'Directory'. The second step, 'Block Public Access settings for this bucket', shows the 'Block all public access' checkbox being selected. A warning message states: 'Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.' The third step, 'Bucket Versioning', shows the 'Disable' option selected.

The screenshots illustrate the process of creating and managing an AWS S3 bucket named "myawsbucketnithin".

- Screenshot 1: Bucket Creation**
Shows the "General purpose buckets" list with one item: "myawsbucketnithin" (Europe (Stockholm) eu-north-1). The creation date is September 29, 2024, at 12:48:24 (UTC+05:30).
- Screenshot 2: Bucket Details**
Shows the "Objects" tab for the bucket. It displays a message: "No objects. You don't have any objects in this bucket." There is a prominent "Upload" button.
- Screenshot 3: Upload Operation**
Shows the "Upload" step of a file upload process. The destination is set to "s3://myawsbucketnithin". A message at the top says: "Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more." Below is a large dashed box for dragging files and a "Destination info" section.

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Files and folders (1 Total, 2.0 KB)

Name	Folder	Type
index.html	-	text/html

Destination info

Destination
s3://myawsbucketnithin

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://myawsbucketnithin	1 file, 4.6 KB (100.0%)	0 files, 0 B (0%)

Files and folders

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	4.6 KB	Succeeded	-

Objects (1) info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
index.html	html	September 29, 2024, 12:56:19 (UTC+05:30)	4.6 KB	Standard

Object Lock

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. [Learn more](#)

Requester pays

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [Learn more](#)

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Hosting type

Host a static website
Use the bucket endpoint as the web address. [Learn more](#)

Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

Index document

Specify the home or default page of the website.

`index.html`

Error document - optional

Successfully edited static website hosting.

myawsbucketnithin

Bucket overview

AWS Region Europe (Stockholm) eu-north-1	Amazon Resource Name (ARN) <code>arn:aws:s3:::myawsbucketnithin</code>	Creation date September 29, 2024, 12:48:24 (UTC+05:30)
---	---	---

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disabled

Multifactor authentication (MFA) delete

An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

The screenshot shows the AWS S3 Bucket Policy editor. On the left, a sidebar lists various S3 features like Buckets, Access Grants, and Storage Lens. The main area displays the current bucket ARN (arn:aws:s3:::myawsbucketnithin) and the JSON policy code:

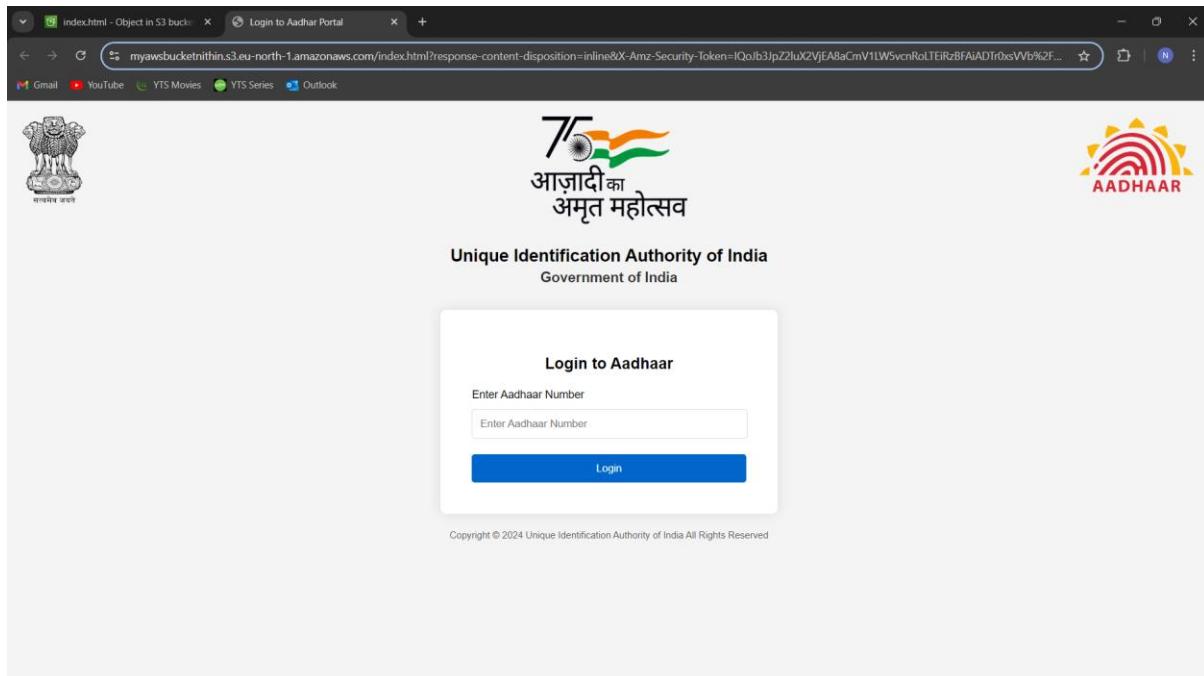
```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Sid": "PublicReadGetObject",
6              "Effect": "Allow",
7              "Principal": "*",
8              "Action": [
9                  "s3:GetObject"
10             ],
11             "Resource": [
12                 "arn:aws:s3:::myawsbucketnithin"
13             ]
14         }
15     ]
16 }

```

A modal window titled "Edit statement" is open on the right, containing the message "Select a statement" and a button "+ Add new statement".

The screenshot shows the AWS S3 Bucket Permissions page for the bucket "myawsbucketnithin". A green success message at the top says "Successfully edited bucket policy." The main content area shows the "Permissions overview" and the "Block public access (bucket settings)" section. Under "Block all public access", it says "Off" and "Individual Block Public Access settings for this bucket".



RESULT:

The static webpage was successfully hosted in the S3 bucket, accessible via a public URL, demonstrating a fully functional website with proper content and layout, leveraging AWS's reliable and scalable infrastructure.

Ex6 – Create a VPC

Nithin Prasath C – 22BCS081

AIM:

To create and configure a Virtual Private Cloud (VPC) in Amazon Web Services (AWS) for secure and isolated networking, enabling efficient management of resources such as EC2 instances and databases within a private network.

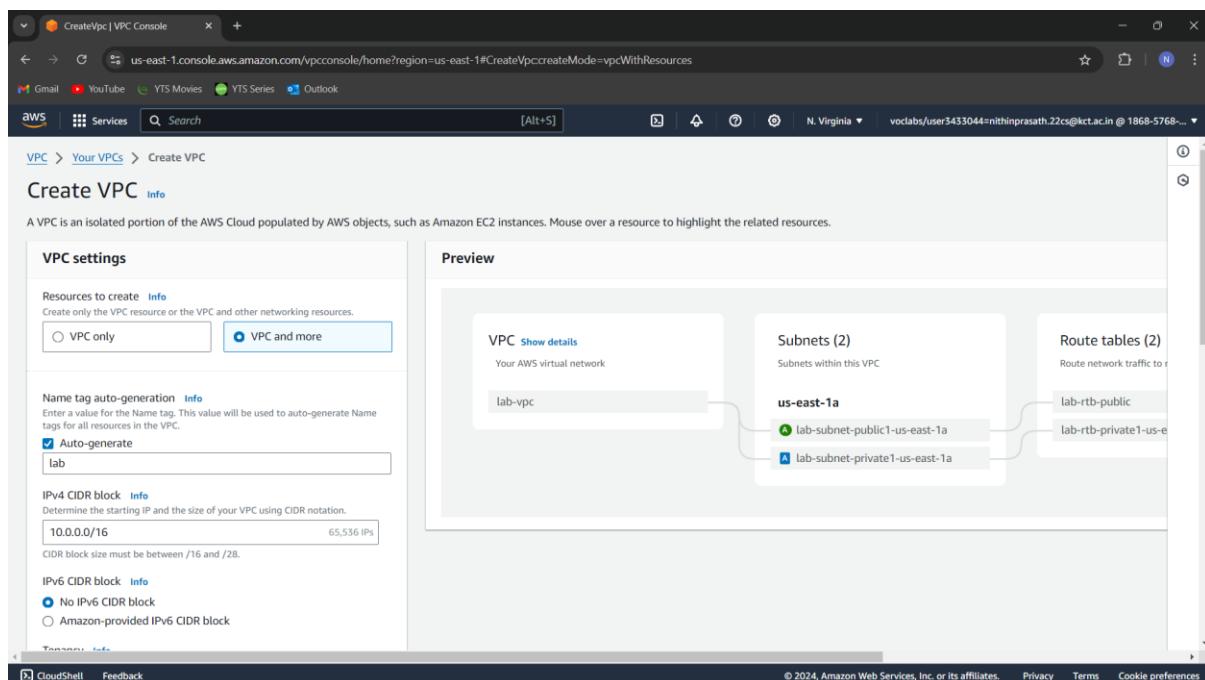
DESCRIPTION:

In this task, we will create a custom Virtual Private Cloud (VPC) in AWS, which allows you to define your own isolated network environment. This includes:

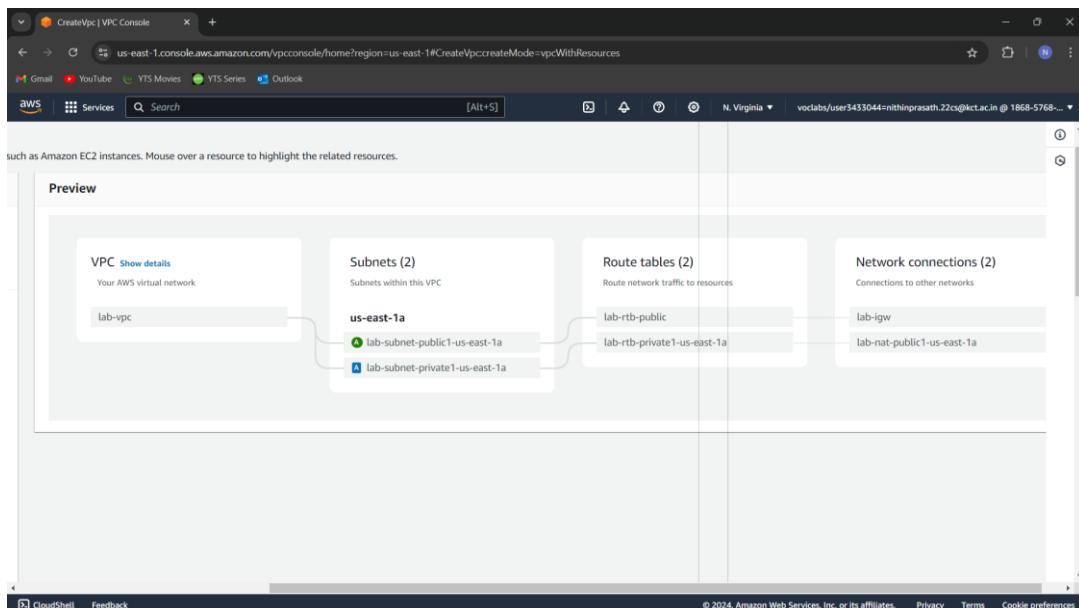
1. **Setting up the VPC** with an IP range (CIDR block).
2. **Creating subnets** for dividing the VPC across availability zones.
3. **Configuring an Internet Gateway** to enable internet access for the VPC.
4. **Setting up route tables** to direct traffic between subnets and the internet.
5. **Configuring security groups** and **network ACLs** to control inbound and outbound traffic for your resources.

This provides a controlled networking environment where instances like EC2 can be deployed securely within private or public subnets.

OUTPUT:



This screenshot shows the 'CreateVpc' page in the AWS VPC console. It includes fields for 'Number of private subnets' (set to 2), 'Public subnet CIDR block in us-east-1a' (10.0.0.0/20), 'Private subnet CIDR block in us-east-1a' (10.0.1.0/20), 'NAT gateways (\$)' (set to 1 AZ), 'VPC endpoints' (set to None), and 'DNS options' (checkboxes for 'Enable DNS hostnames' and 'Enable DNS resolution' both checked). The status bar at the bottom indicates '© 2024, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.



This screenshot shows the 'Create VPC workflow' page in the AWS VPC wizard. It lists 21 steps under the heading 'Success', all of which are completed with green checkmarks. The steps include creating the VPC, subnets, route tables, and various network components. A large orange 'View VPC' button is visible at the bottom. The status bar at the bottom indicates '© 2024, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

VPC dashboard > Your VPCs > **vpc-071e43b1e17b74413 / lab-vpc**

Details		Info	
VPC ID	vpc-071e43b1e17b74413	State	Available
Tenancy	Default	DNS hostnames	Enabled
Default VPC	No	Main route table	rtb-05faedbf36c35395
Network Address Usage metrics	Disabled	IPv6 pool	-
		Owner ID	186857684047

Resource map | CIDRs | Flow logs | Tags | Integrations

Resource map | Info

- VPC Show details
- Subnets (2) Subnets within this VPC
- Route tables (3) Route network traffic to resources
- Network interface Connect to this VPC

Subnets | VPC Console

Subnets (1/7) Info						
Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Last updated
-	subnet-08cf61f408c2ff592	Available	vpc-0705919ff34efaa57	172.31.80.0/20	-	12 minutes ago
-	subnet-0eb0ca6ecc5ac1ac	Available	vpc-0705919ff34efaa57	172.31.48.0/20	-	
-	subnet-0312d088170512988	Available	vpc-0705919ff34efaa57	172.31.0.0/20	-	
-	subnet-0bbff076ba30cdf0c6	Available	vpc-0705919ff34efaa57	172.31.32.0/20	-	
<input checked="" type="checkbox"/> Work Public Subnet	subnet-0b355e083ebb091c5	Available	vpc-0fdff40958321dd37 Work Public Subnet	10.0.0.0/24	-	
-	subnet-0ea08993cebb271bc	Available	vpc-0705919ff34efaa57	172.31.16.0/20	-	
-	subnet-00517500acd2b551e	Available	vpc-0705919ff34efaa57	172.31.64.0/20	-	

subnet-0b355e083ebb091c5 / Work Public Subnet

Details | Flow logs | Route table | Network ACL | CIDR reservations | Sharing | Tags

Subnet ID	subnet-0b355e083ebb091c5	Subnet ARN	arn:aws:ec2:us-east-1:186857684047:subnet/subnet-0b355e083ebb091c5	State	Available	IPv4 CIDR	10.0.0.0/24
Available IPv4 addresses	250			IPv6 CIDR association ID	-	Availability Zone	us-east-1a
				IPv6 CIDR	-		

CreateSubnet | VPC Console

VPC > Subnets > Create subnet

Create subnet | Info

VPC

VPC ID: vpc-071e43b1e17b74413 (lab-vpc)

Associated VPC CIDRs: 10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name: lab-subnet-public2

The name can be up to 256 characters long.

Availability Zone: us-east-1a

The screenshot shows the 'CreateSubnet | VPC Console' page. In the 'Name' field, 'lab-subnet-public2' is entered. Under 'Availability Zone', 'US East (N. Virginia) / us-east-1b' is selected. The 'IPv4 CIDR block' is set to '10.0.0.0/16'. The 'IPv4 subnet CIDR block' is set to '10.0.2.0/24'. A tag 'Name: lab-subnet-public2' is added. The 'Create subnet' button is highlighted.

The screenshot shows the 'subnets | VPC Console' page. It displays a table of subnets, with 'lab-subnet-public2' listed. The subnet ID is 'subnet-0961188e85711c7d4', it is in state 'Available', and its IPv4 CIDR is '10.0.2.0/24'. The 'Create subnet' button is visible at the top right.

Name	Subnet ID	State	VPC	IPv4 CIDR
lab-subnet-public2	subnet-0961188e85711c7d4	Available	vpc-071e43b1e17b74413 lab-vpc	10.0.2.0/24

The screenshot shows the 'RouteTables | VPC Console' page. It displays a table of route tables, with 'lab-rtb-private1-us-east-1a' listed. The route table ID is 'rtb-018b9a779c9ef76ec', it is associated with subnet 'subnet-0c466a3b71a77a...', and its main VPC is 'vpc-071e43b1e17b74413'. The 'Create route table' button is visible at the top right.

Name	Route table ID	Explicit subnet associations	Main	VPC
lab-rtb-public	rtb-0117682b8fc87016	subnet-0dec2ee1c2b8c0...	No	vpc-071e43b1e17b74413
Work Public Route Table	rtb-017611280d62f0faf	subnet-0b355e083eb09...	No	vpc-0fd9f40958321dd37
lab-rtb-private1-us-east-1a	rtb-018b9a779c9ef76ec	subnet-0c466a3b71a77a...	Yes	vpc-071e43b1e17b74413

The screenshot shows the AWS VPC console interface. On the left, there's a sidebar with navigation links like EC2 Global View, Virtual private cloud, Route tables, and Security. The main area displays 'Route tables (1/6) Info'. It lists three route tables: 'rtb-019800d21ct838bea' (Main Yes, VPC vpc-0705919f54efaa57), 'Work Public Route Table' (Main No, VPC vpc-0df9f40958321d37), and 'rtb-055faed8f56c35395' (Main Yes, VPC vpc-071e43b1e17b74413). Below this, a specific route table 'rtb-018b9a779c9ef76ec / lab-rtb-private1-us-east-1a' is selected, showing its subnet associations.

This screenshot shows the 'Edit subnet associations' dialog for route table 'rtb-0117682b8fec87016'. In the 'Available subnets (2/3)' section, three subnets are listed: 'lab-subnet-public1-us-east-1a' (selected), 'lab-subnet-public2' (selected), and 'lab-subnet-private1-us-east-1a'. In the 'Selected subnets' section, two subnets are selected: 'subnet-0dec2ee1c2b8c063a / lab-subnet-public1-us-east-1a' and 'subnet-0961188e85711c7d4 / lab-subnet-public2'. At the bottom right, there are 'Cancel' and 'Save associations' buttons.

This screenshot shows the 'Create security group' dialog in the AWS EC2 console. Under 'Basic details', the 'Security group name' is set to 'Web Security Group'. The 'Description' field contains 'Enable HTTP access'. The 'VPC' field is set to 'vpc-071e43b1e17b74413 (lab-vpc)'. Under 'Inbound rules', it says 'This security group has no inbound rules.' and there is an 'Add rule' button. The bottom of the screen shows standard AWS navigation links.

The screenshot shows two consecutive pages from the AWS Cloud Console. The top page is titled 'CreateSecurityGroup | EC2 | us-east-1' and displays the configuration for a new security group. It includes fields for 'Description' (Enable HTTP access), 'VPC Info' (selected VPC), and 'Inbound rules'. An inbound rule is defined for 'HTTP' on port '80' from 'Anywhere' with a description 'Permit web requests'. A warning message at the bottom states: '⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The bottom page is titled 'SecurityGroup | EC2 | us-east-1' and shows the successful creation of the security group 'sg-0dcf1fcbb668fc16d'. The details page lists the security group ID, owner, and inbound/outbound rule counts. The 'Inbound rules' section shows one rule for 'HTTP' on port '80' from 'Anywhere'.

RESULT:

The VPC is now fully operational and ready to host resources. The instance is now running and accessible. Screenshots are attached to demonstrate each step.

Ex7 – Creating MySQL Instances

Nithin Prasath C – 22BCS081

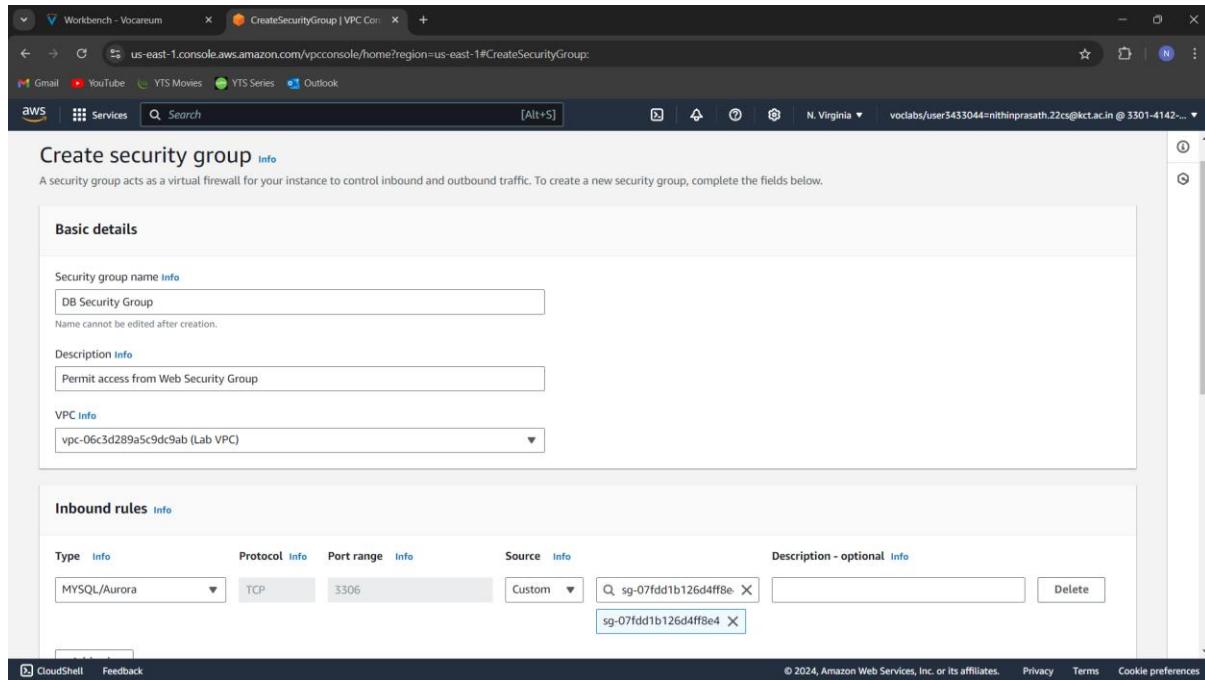
AIM:

To create and configure a MySQL database instance in Amazon Web Services (AWS) using the RDS (Relational Database Service) platform, enabling secure and scalable cloud-based database management.

DESCRIPTION:

This lab task involves creating a fully managed MySQL instance in AWS RDS. The steps include logging into the AWS Management Console, setting up a new MySQL instance, selecting instance class and storage options, configuring security (VPC, subnets, and security groups), and enabling multi-AZ deployments for high availability. Once the instance is created, you will configure the database with appropriate credentials and connect to it using MySQL Workbench or a similar SQL client. The task demonstrates the ability to provision, configure, and manage MySQL databases in a cloud environment, while ensuring security and availability using AWS RDS.

OUTPUT:



VPC dashboard > sg-005eacac8aa8f274e - DB Security Group

Details

Security group name	sg-005eacac8aa8f274e	Description	Permit access from Web Security Group
Owner	330141422282	Inbound rules count	1 Permission entry
		Outbound rules count	1 Permission entry

Inbound rules | Outbound rules | Tags

Inbound rules (1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-024d8d017c0805...	-	MySQL/Aurora	TCP	3306

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name: DB-Subnet-Group
You won't be able to modify the name after your subnet group has been created.

Description: DB Subnet Group

VPC: Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
Lab VPC (vpc-06c3d289a5c9dc9ab)

Add subnets

Availability Zones: Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone

Amazon RDS > Subnet Groups

Subnet groups (1)

Name	Description	Status	VPC
db-subnet-group	DB Subnet Group	Complete	vpc-06c3d289a5c9dc9ab

Successfully created DB-Subnet-Group. View subnet group

Screenshot 1: AWS EC2 Instances Overview

Screenshot 2: AWS RDS Create Database - MySQL Engine Options

Screenshot 3: AWS RDS Create Database - Templates

The screenshot shows the 'Create database - RDS Manager' interface on the AWS console. The 'Settings' tab is selected. In the 'DB instance identifier' field, 'database-1' is entered. Under 'Master username', 'admin' is specified. For 'Credentials management', the 'Self managed' option is selected, indicating the user will create their own password. A note states: 'Create your own password or have RDS create a password that you manage.' The 'Master password' field contains a masked password. On the right, a sidebar provides information about MySQL, including its popularity and various features like support for up to 64 TiB and automated backup.

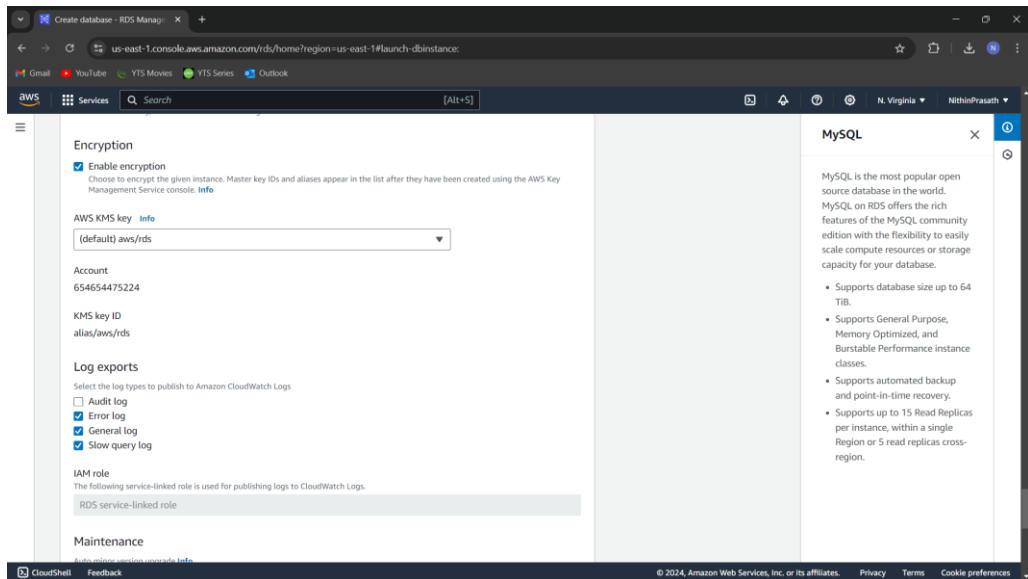
The screenshot shows the 'Create database - RDS Manager' interface on the AWS console. The 'Storage' tab is selected. The 'Storage type' dropdown is set to 'General Purpose SSD (gp3)'. An 'Allocated storage' input field shows '20' GiB. A note indicates that after modifying storage, the DB instance status will be in 'storage-optimization' and remains available. The 'Advanced settings' section notes baseline IOPS of 3,000 and throughput of 125 Mbps for allocated storage less than 400 GiB. The 'Storage autoscaling' section includes an 'Enable storage autoscaling' checkbox which is checked. A 'Maximum storage threshold' input field shows '1000' GiB. The right sidebar continues to provide MySQL information and features.

The screenshot shows the 'Create database - RDS Manager' interface on the AWS console. The 'VPC' tab is selected. A note states that after creating a database, its VPC cannot be changed. The 'DB subnet group' dropdown is set to 'default'. Under 'Public access', the 'Yes' option is selected, allowing public IP access from outside the VPC. The 'VPC security group (firewall)' dropdown is set to 'Choose existing' with 'Choose existing VPC security groups' selected. A note says to choose VPC security groups that allow appropriate incoming traffic. The 'Existing VPC security groups' dropdown shows 'default'. The 'Availability Zone' dropdown is set to 'No preference'. The right sidebar continues to provide MySQL information and features.

This screenshot shows the first step of creating a new MySQL database instance. It's titled 'Create database - RDS Manager'. On the left, there's a sidebar with 'Additional configuration' expanded. Under 'VPC security group (firewall)', there are two options: 'Choose existing' and 'Create new'. The 'Create new' button is highlighted. A sub-section for 'New VPC security group name' contains a text input field with 'task7vpcgroup'. Below it, 'Availability Zone' is set to 'No preference'. Under 'RDS Proxy', there's an option to 'Create an RDS Proxy' which is currently unchecked. A note says 'RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing.' A 'Certificate authority - optional' section follows, showing a dropdown menu with 'rds-ca-rsa2048-g1 (default)' selected. A note states 'If you don't select a certificate authority, RDS chooses one for you.' On the right, a 'MySQL' info panel provides general information about MySQL and its features.

This screenshot shows the second step of the creation wizard. It includes sections for 'Database authentication' and 'Monitoring'. In 'Database authentication', 'Password authentication' is selected. In 'Monitoring', 'Enable Enhanced Monitoring' is checked, and 'Granularity' is set to '60 seconds'. A note says 'Clicking "Create database" will authorize RDS to create the IAM role rds-monitoring-role'. The right side of the screen features the same MySQL info panel as the previous screenshot.

This screenshot shows the third step of the creation wizard. It includes sections for 'Additional configuration', 'Database options', and 'Backup'. Under 'Additional configuration', there's a note about encryption and backup settings. In 'Database options', 'Initial database name' is set to 'Database1'. Under 'DB parameter group', 'default.mysql8.0' is selected. In 'Option group', 'defaultmysql8-0' is selected. Under 'Backup', 'Enable automated backups' is checked. A note says 'Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details here.' A 'Backup retention period' section shows a dropdown set to '1 day'. The right side of the screen features the same MySQL info panel as the previous screenshots.



Amazon RDS

[Dashboard](#) [Databases](#) [Query Editor](#) [Performance insights](#) [Snapshots](#) [Exports in Amazon S3](#) [Automated backups](#) [Reserved instances](#) [Proxies](#)

[Subnet groups](#) [Parameter groups](#) [Option groups](#) [Custom engine versions](#) [Zero-ETL Integrations](#) [New](#)

[Events](#) [Event subscriptions](#)

[CloudShell](#) [Feedback](#)

lab-db

[RDS](#) > [Databases](#) > lab-db

Summary

DB identifier	Status	Role	Engine	Recommendations
lab-db	Modifying	Instance	MySQL Community	
CPU	Class	Current activity	Region & AZ	
	db.t3.micro	0 Connections	us-east-1a	

Connectivity & security

Endpoint & port	Networking	Security
Endpoint lab-db.clonsfz9vsoi.us-east-1.rds.amazonaws.com	Availability Zone us-east-1a	VPC security groups DB Security Group (sg-005eacaa8bf274de) Active
Port 3306	VPC	

[CloudShell](#) [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

[Workbench - Vocareum](#) [Database Details - RDS Manager](#) [Welcome to AWS Technical Escalation](#)

Not secure 3.90.52.36

[Gmail](#) [YouTube](#) [YTS Movies](#) [YTS Series](#) [Outlook](#)

aws [Load Test](#) [RDS](#)

Meta-Data	Value
InstanceId	i-077b7eb8553a8e15
Availability Zone	us-east-1b

Current CPU Load: 1%

The image contains two screenshots of a web browser window. The top screenshot shows a form for connecting to an AWS RDS database. The fields are: Endpoint (lab-db.clonsfz9vsoi.us-east-1.rds.amazonaws.com), Database (lab), Username (main), and Password (redacted). A 'Submit' button is at the bottom. The bottom screenshot shows an 'Address Book' page with a table of contacts:

Last name	First name	Phone	Email	Admin
Doe	Jane	010-110-1101	janed@someotheraddress.org	Edit Remove
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove

RESULT:

Successfully created a MySQL instance in AWS RDS. The database is now running, secure, and accessible. Screenshots are attached to demonstrate each step of the configuration process, from instance setup to successful connection using a MySQL client.

Ex3 - Hosting Web application in cloud

Nithin Prasath C – 22BCS081

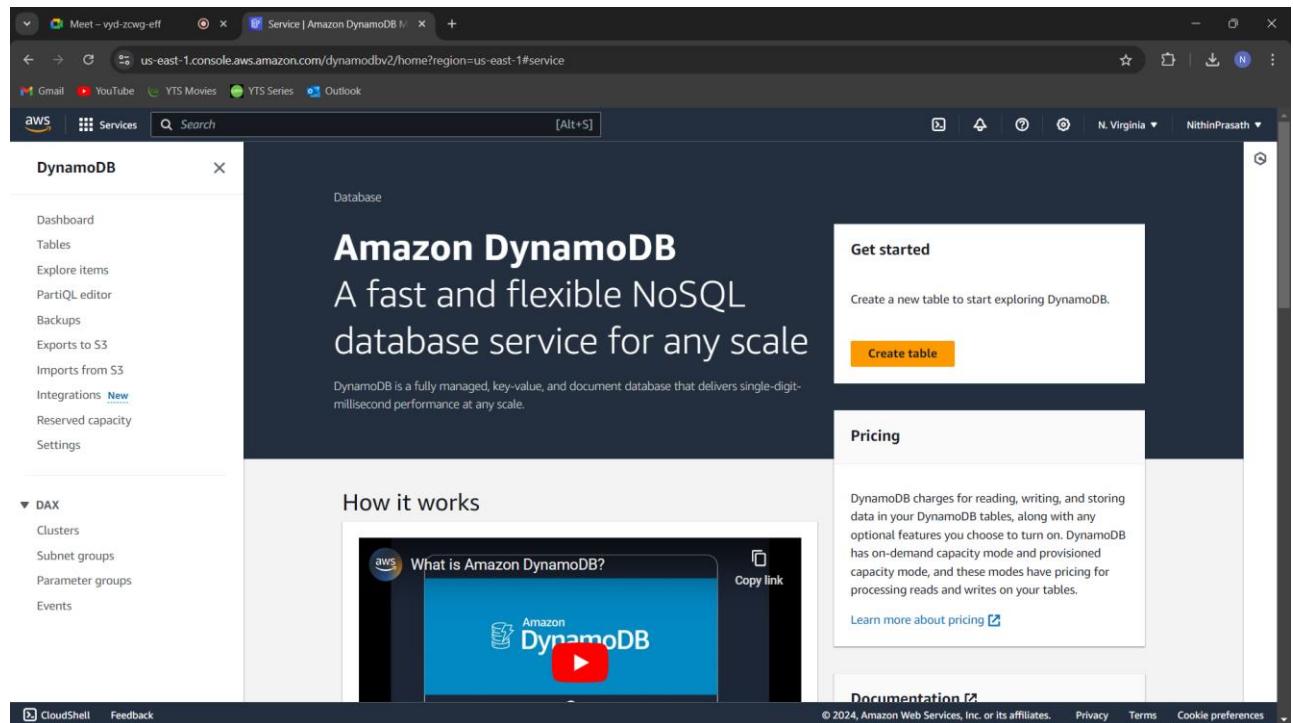
AIM:

To create and carryout Read and Write operations on DynamoDB.

DESCRIPTION:

In services, select DynamoDB and click on create table. Provide the table name and click on create table. Click on create item and fill the attributes in the table. Select PartiQL and perform read operation using select command to view the table. Perform write operation using insert to insert a new value into the table.

OUTPUT:



The screenshots illustrate the process of creating a new DynamoDB table:

- Screenshot 1: Create table (Step 1)**
Shows the 'Create table' wizard. The 'Table name' is set to 'task8table'. The 'Partition key' is 'task8PK' (String type). There is no 'Sort key' defined.
- Screenshot 2: Table settings (Step 2)**
Shows the 'Table settings' configuration. The 'Default settings' tab is selected. Under 'Default table settings', the table class is 'DynamoDB Standard', capacity mode is 'Provisioned', and both read and write capacities are set to 5. Auto scaling is enabled.
- Screenshot 3: List tables (Step 3)**
Shows the 'Tables' list after the table has been created. The table 'task8table' is listed with a status of 'Active', partition key 'task8PK', sort key '-', and both read and write capacities set to 'Provisioned (5)'.

The screenshot shows the AWS DynamoDB console with the table `task8table`. The table has a single item with the key `task8PK` and value `key1`. The table is provisioned with 1 RU and 1 WU. It has no active alarms and is set to Point-in-time recovery (PITR) Off.

The screenshot shows the AWS DynamoDB console with the table `task8table`. A scan operation was performed, returning 0 items. The table has a single item with the key `task8PK` and value `key1`.

The screenshot shows the AWS DynamoDB console with the table `task8table`. A new item is being created with the following attributes:

Attribute name	Value	Type
<code>task8PK</code> - Partition key	<code>key1</code>	String
<code>name</code>	<code>Nithin</code>	String
<code>age</code>	<code>19</code>	Number

The screenshots illustrate the use of filters in the AWS DynamoDB Item Explorer:

- Screenshot 1:** Shows the initial state where a single item has been successfully saved. The table contains one item: key1 (String), age (Number), name (String) with value Nithin.
- Screenshot 2:** Shows the "Scan or query items" section with two filters applied: name = Nithin and age = 19. The results show the same single item.
- Screenshot 3:** Shows the "Scan or query items" section with the same filters applied. The results show the same single item, and a message indicates that 0.5 read capacity units were consumed.

The screenshot shows the PartiQL editor interface within the AWS DynamoDB console. The sidebar on the left lists various AWS services, with 'DynamoDB' selected. Under 'PartiQL editor', there is a 'Tables (1)' section showing a single table named 'task8table'. A 'Query 1' tab is active, containing the SQL query: `select * from task8table`. Below the query are 'Run' and 'Clear' buttons. At the bottom, there are 'Table view' and 'JSON view' tabs, with 'Table view' currently selected.

This screenshot shows the results of the query execution from the previous screen. The 'Completed' status is displayed, indicating the query was run on 9/22/2024 at 1:13:29 PM with an elapsed time of 753ms. The 'Items returned (1)' section shows one item in the 'task8table': key1 with name Nithin and age 19. There is a 'Download results to CSV' button next to the items table.

This screenshot shows a new insert operation in the PartiQL editor. The 'Tables (1)' section shows 'task8table'. The 'Query 1' tab contains the SQL query: `insert into task8table values ('task8PK';'key2', 'name'; 'Koushik', 'age';21);`. The 'Run' and 'Clear' buttons are present below the query. The status bar at the bottom indicates the query is 'Completed'.

The screenshot shows the Amazon DynamoDB PartiQL editor interface. On the left, a sidebar menu includes options like Dashboard, Tables, Explore items, PartiQL editor (which is selected), Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Below this is a DAX section with Clusters, Subnet groups, Parameter groups, and Events. The main area displays a 'Tables (1)' list with 'task8table' selected. A 'Query 1' tab is open with the SQL query: `1 select * from task8table;`. Below the query is a 'Run' button and a 'Clear' button. At the bottom, there are 'Table view' and 'JSON view' tabs, and a status message 'Failed'.

This screenshot shows the same PartiQL editor interface after the query has been executed. The 'Run' button is now greyed out. The status bar at the bottom indicates the query was completed successfully. The 'Table view' tab is selected, showing the results of the query. The results table has columns: task8PK, name, and age. The data rows are: key2, Koushik, 21 and key1, Nithin, 19. There is also a 'Download results to CSV' button.

RESULT:

Thus, table was created and read and write operations were performed successfully on DynamoDB.

Ex - 9 : VM Migration

Nithin Prasath C – 22BCS081

AIM:

To perform and demonstrate Virtual Machine (VM) Migration, following the provided steps to transfer a running virtual machine from one host to another without interrupting its operation.

DESCRIPTION:

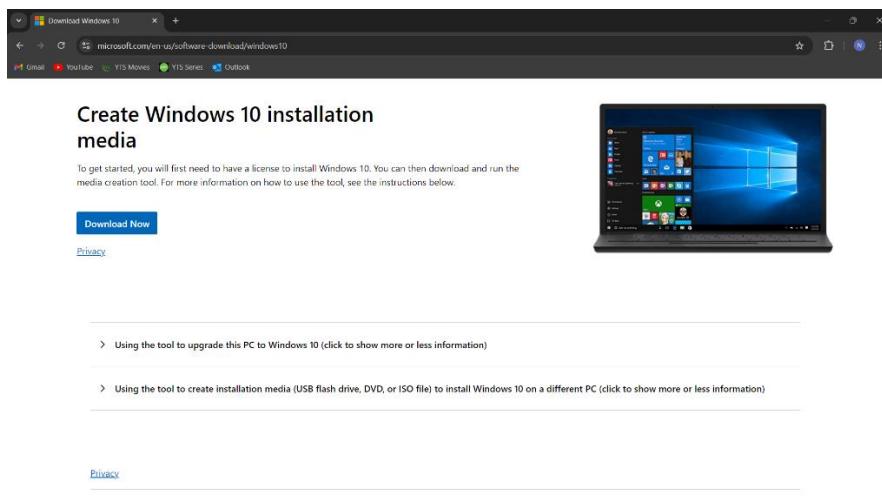
VM Migration involves moving a virtual machine (VM) from one physical host to another while maintaining its operational state. This process ensures that the VM continues to run smoothly without downtime, making it an essential practice in load balancing, system maintenance, or disaster recovery.

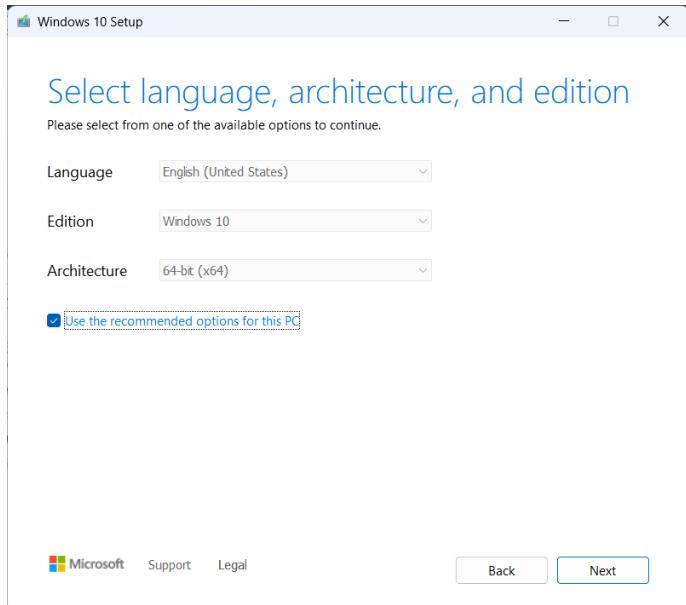
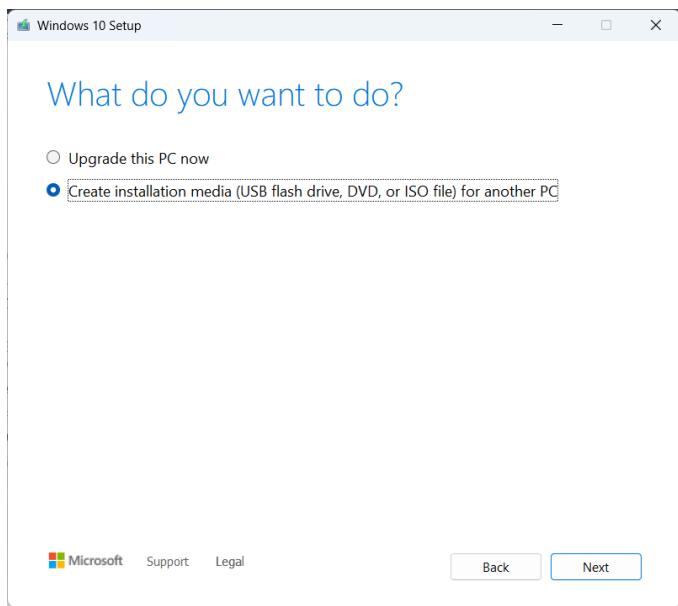
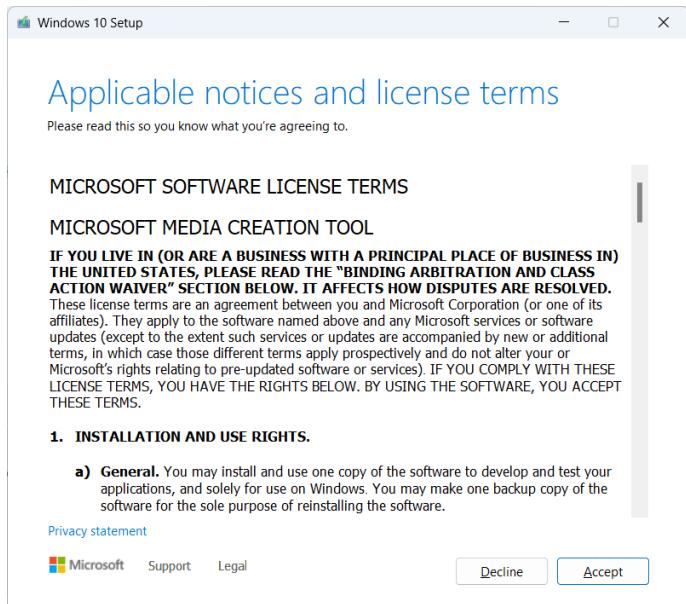
The migration process typically involves:

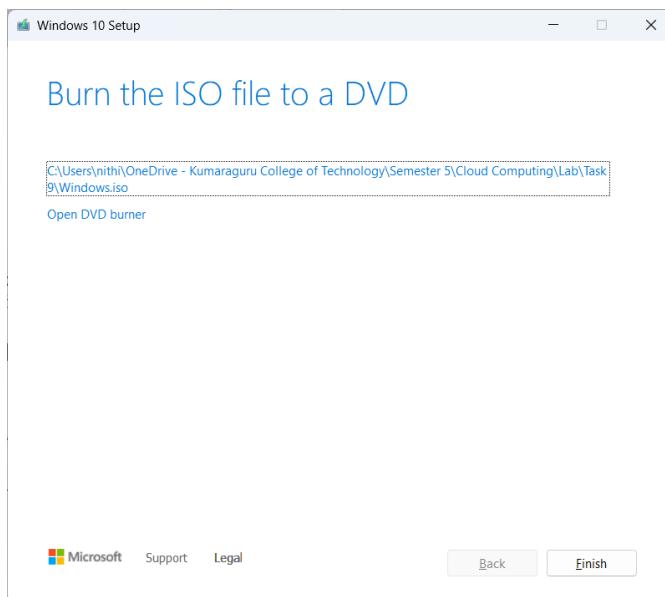
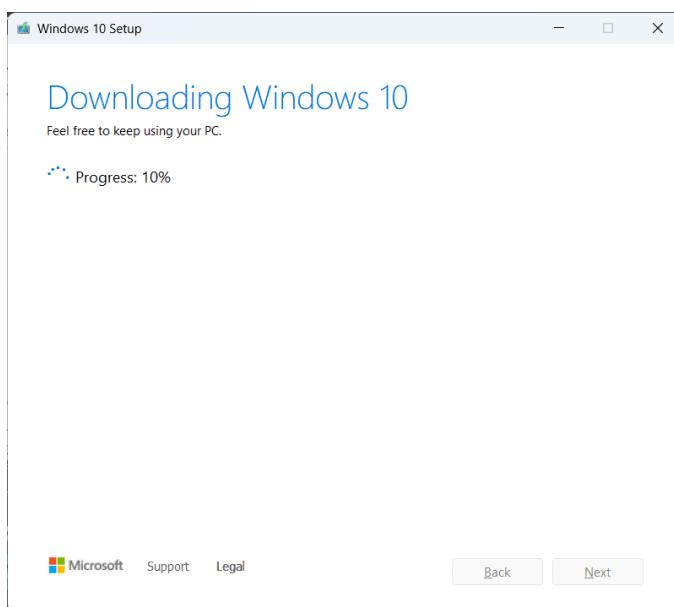
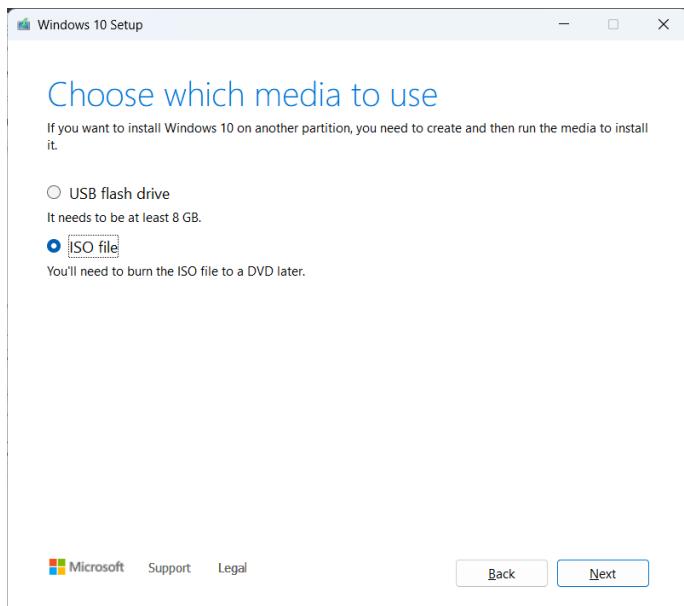
1. **Preparation of Source and Destination Hosts:** Ensure both hosts are configured properly and meet the requirements for the migration.
2. **Initiating the Migration:** Start the migration process by selecting the VM and specifying the destination host.
3. **Data Transfer:** The VM's memory, disk files, and state are transferred over the network to the destination host.
4. **Completion:** The VM is resumed on the destination host with its operations continuing as if uninterrupted, while it is removed from the source host.

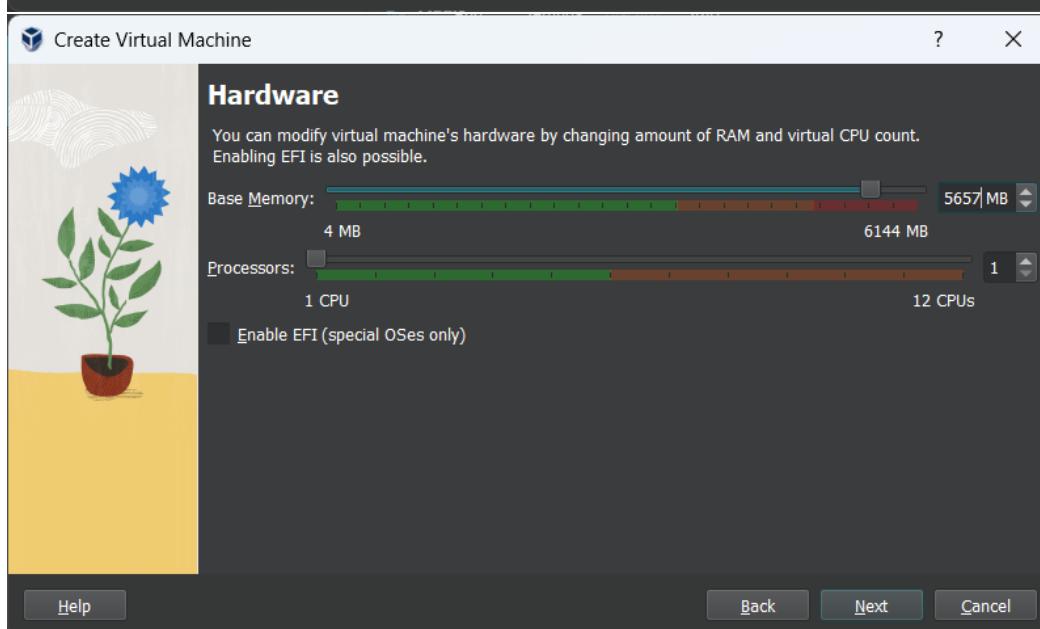
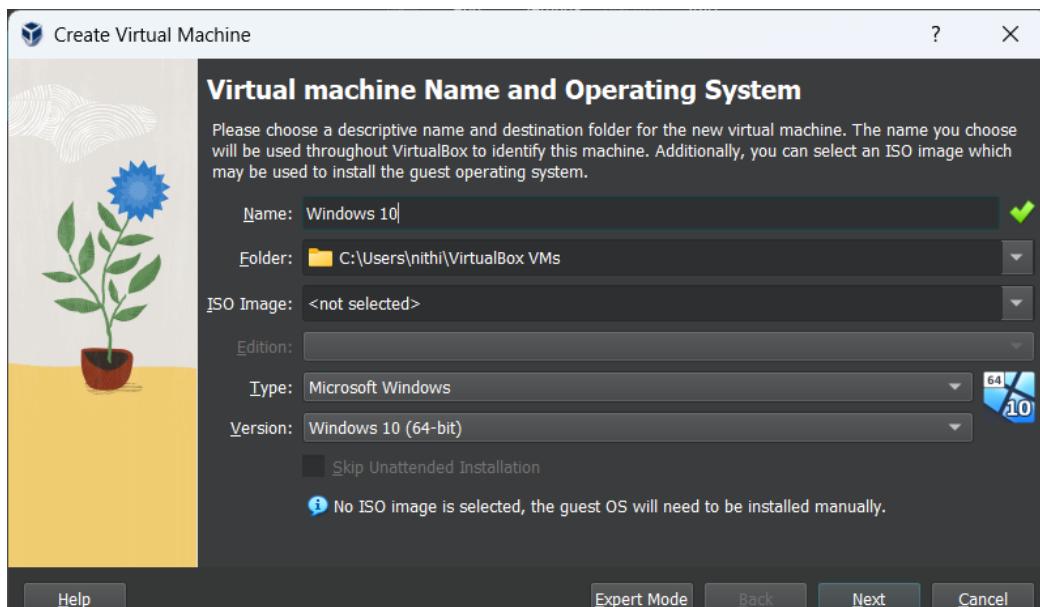
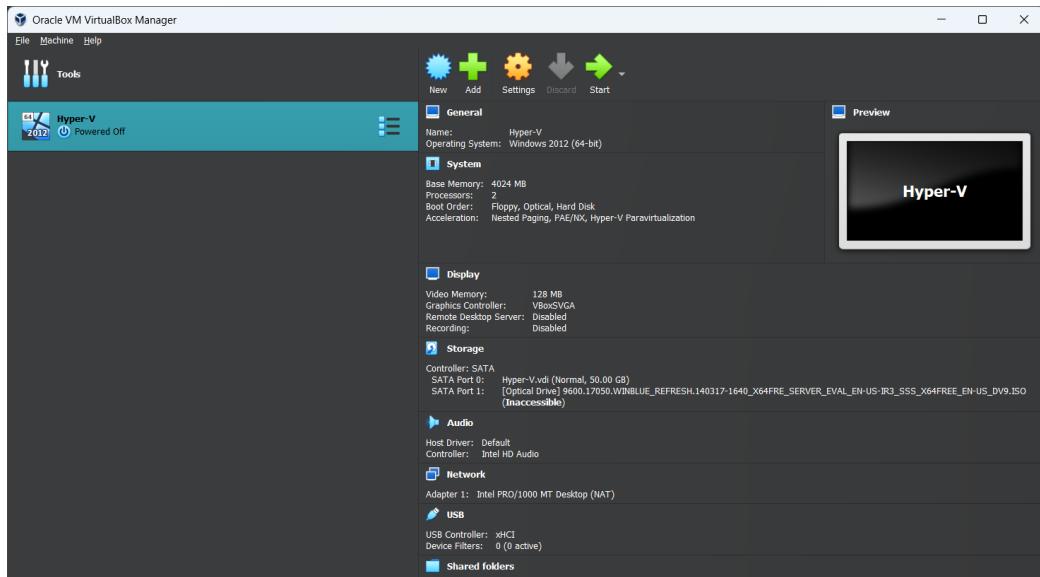
This exercise simulates real-world scenarios where VM migration is necessary for system upgrades, resource optimization, or failure recovery.

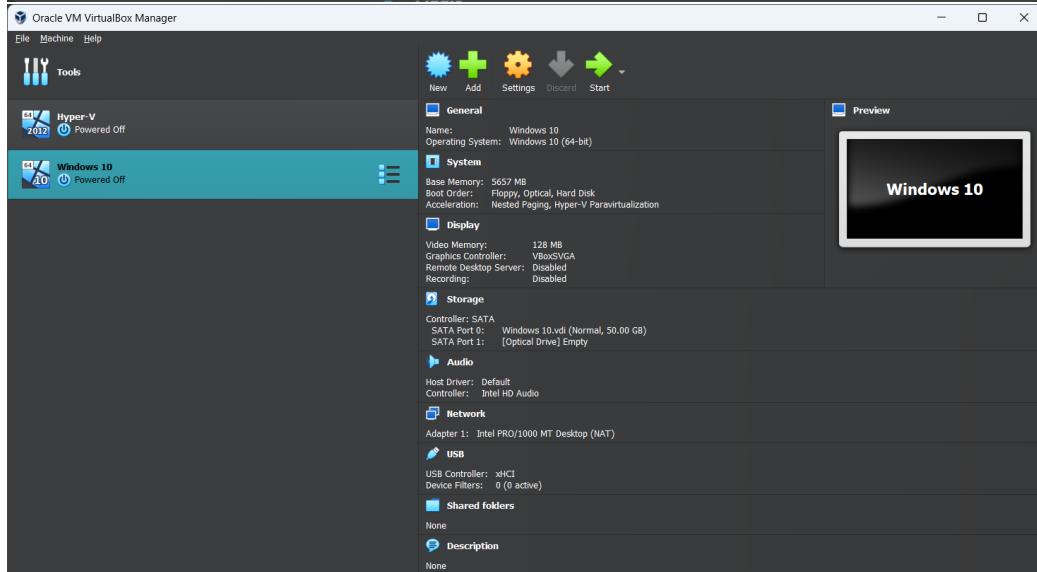
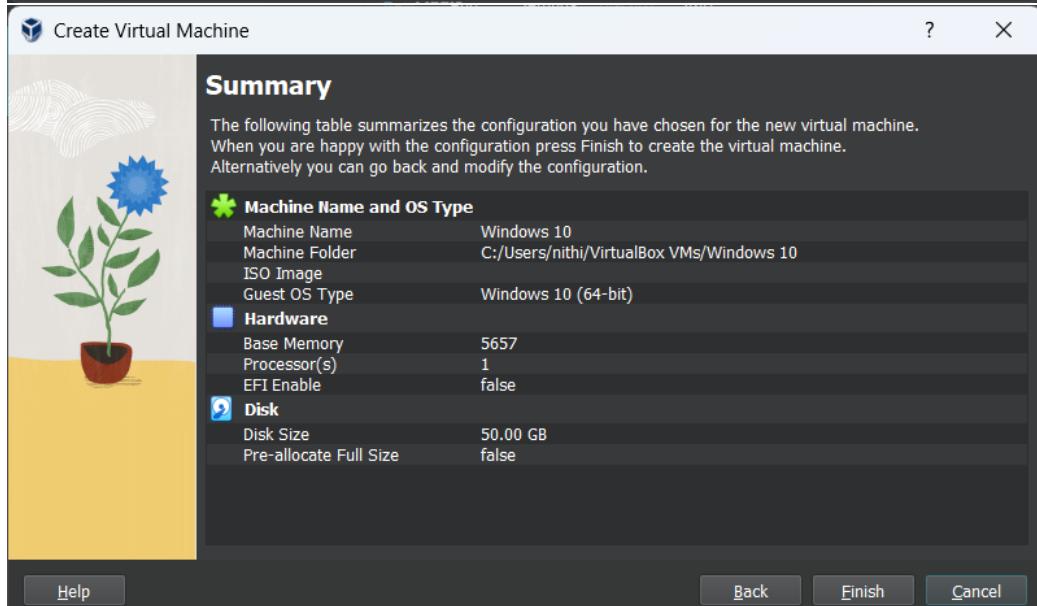
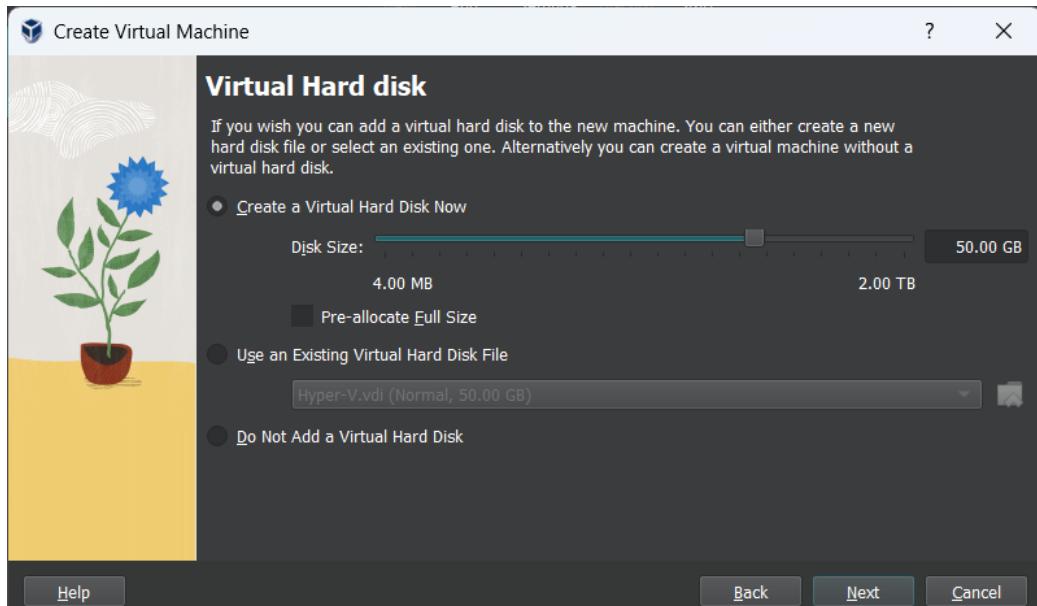
OUTPUT:

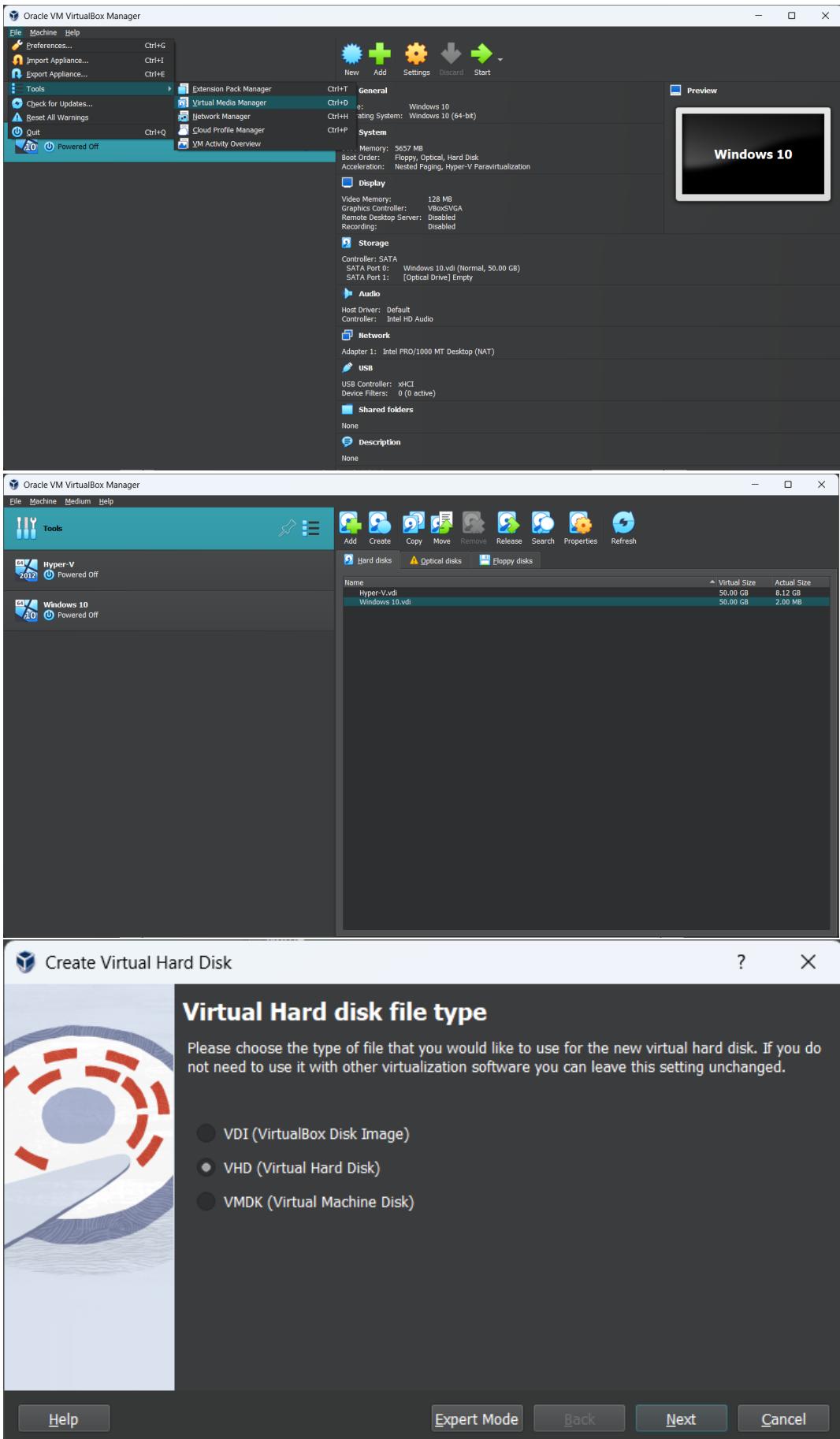


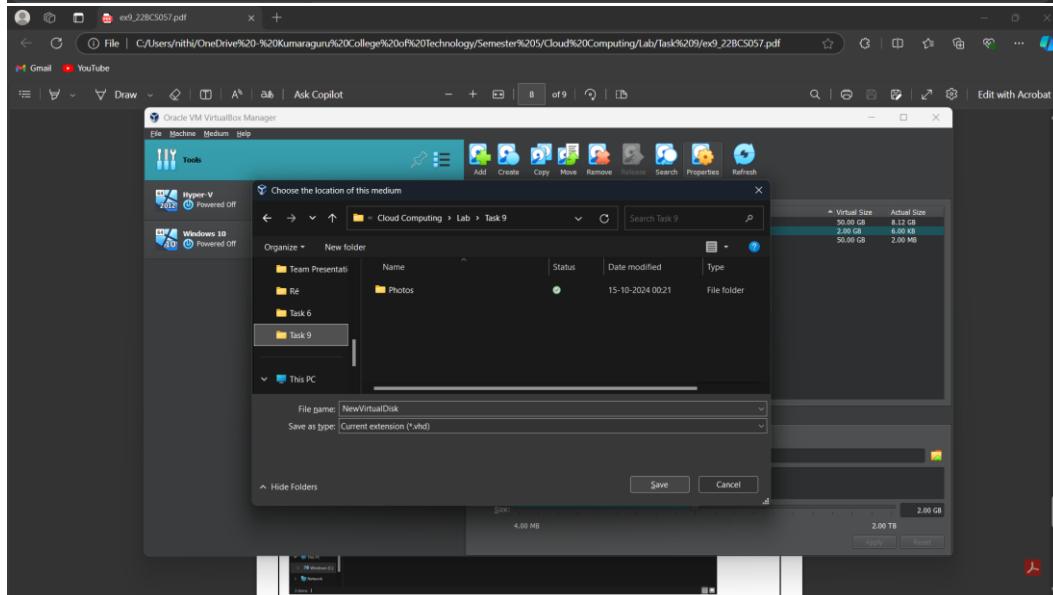
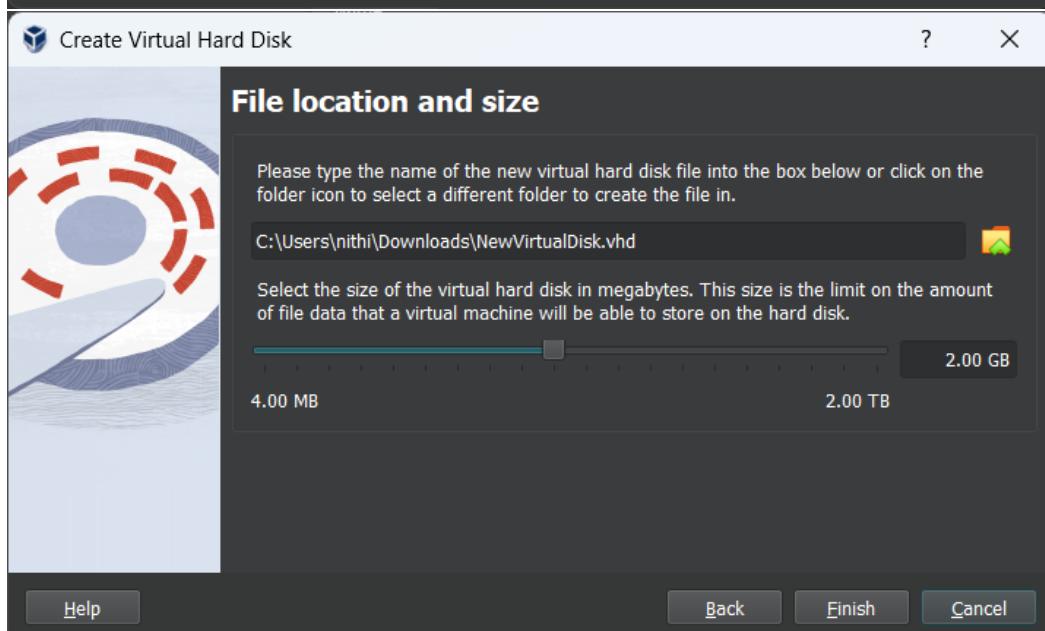
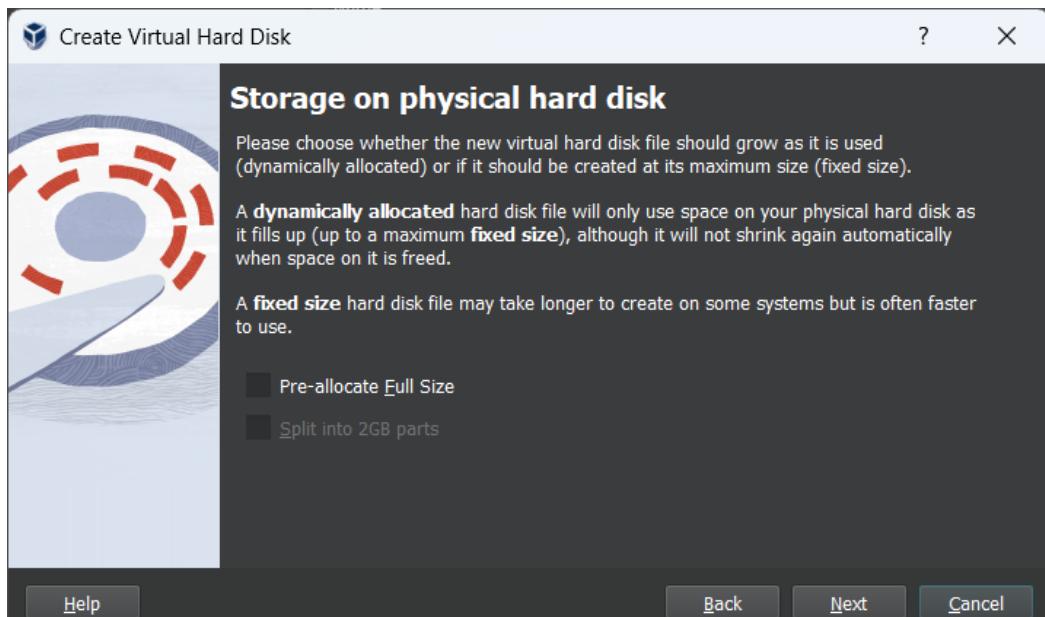


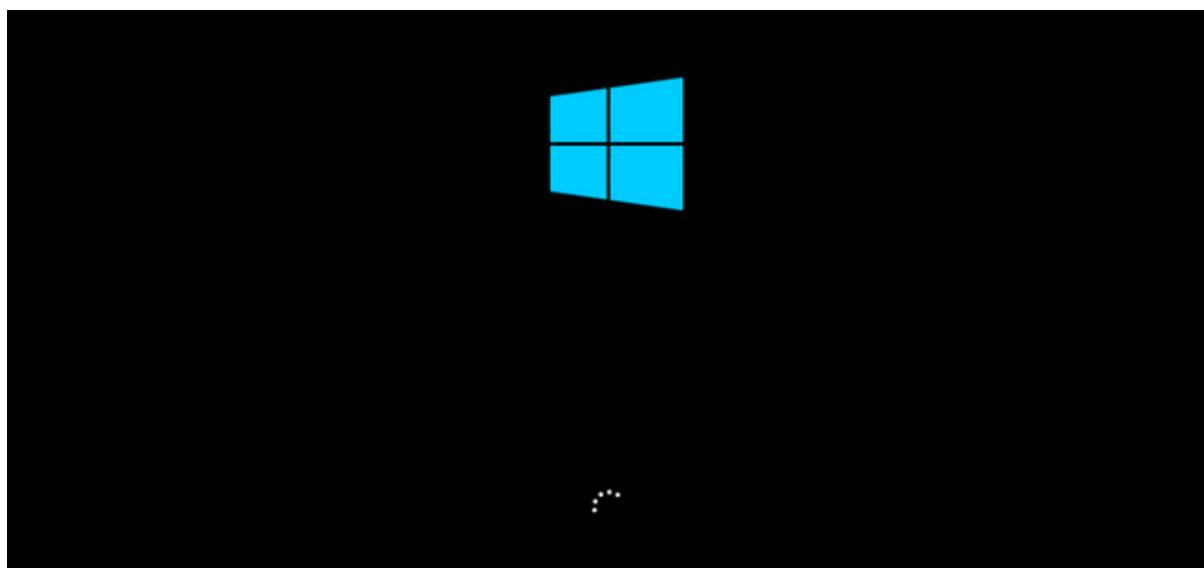
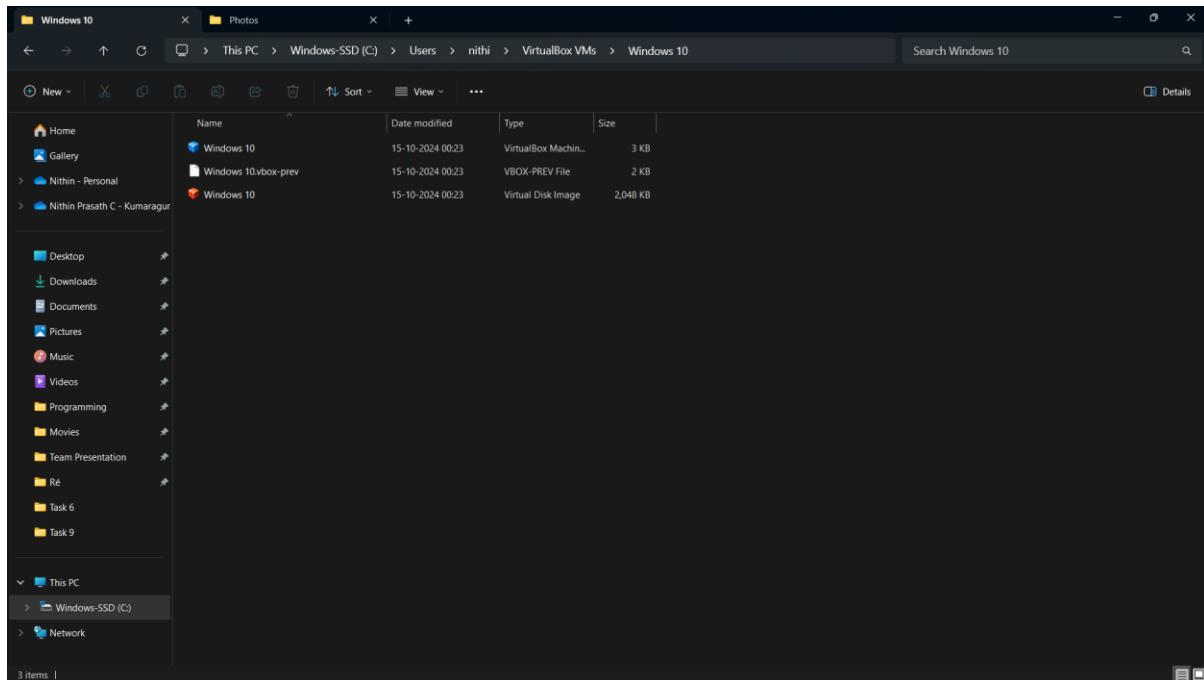












RESULT:

The VM migration was successfully performed by following the step-by-step procedure, demonstrating seamless transition with no downtime. Screenshots and logs confirmed the uninterrupted operation of the VM on the destination host.

Ex - 10 : Virtualization

Nithin Prasath C – 22BCS081

AIM:

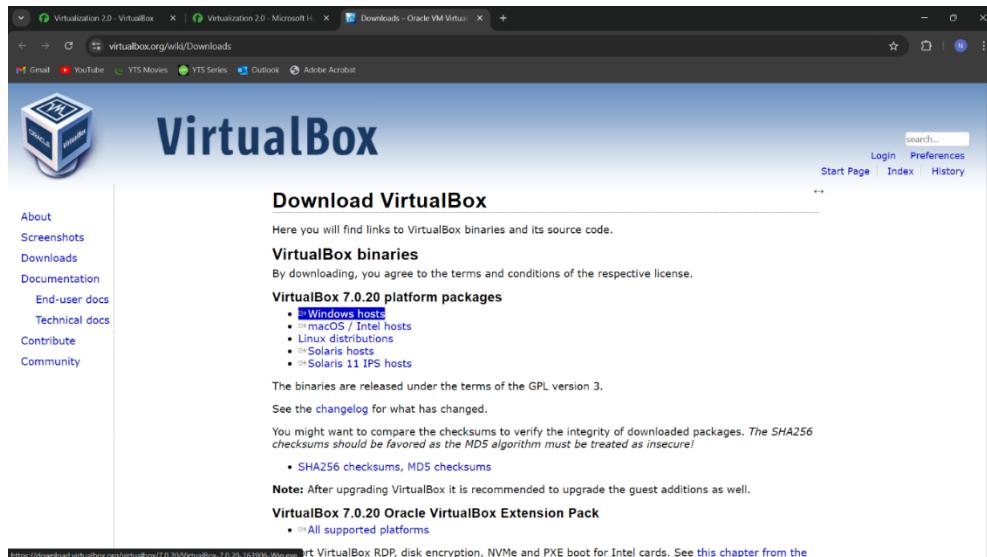
To install and configure Hyper-V on a Windows Server 2012 environment, enabling the server to create and manage virtual machines for the purpose of consolidating workloads, improving server utilization, and creating a flexible, efficient, and scalable infrastructure.

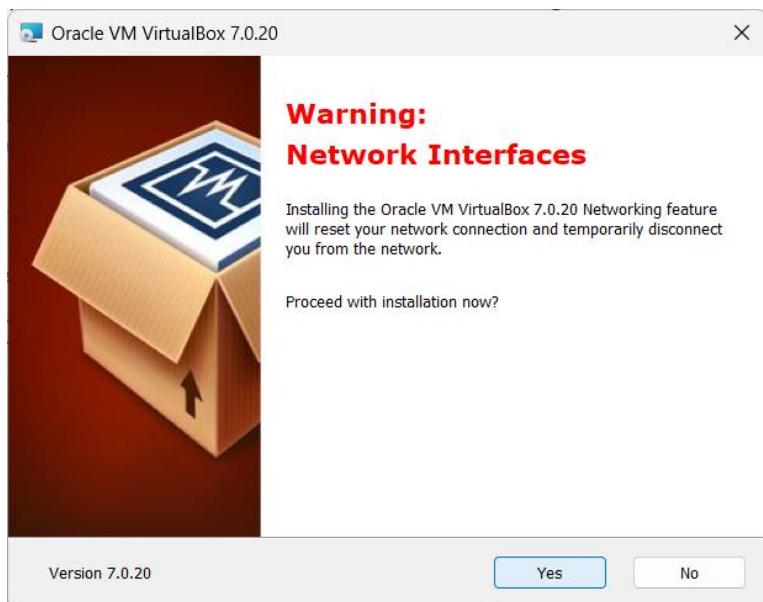
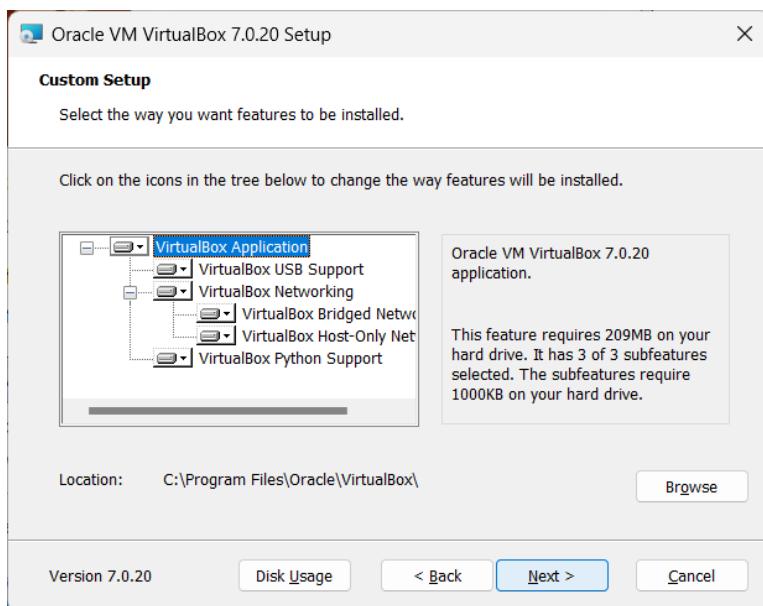
DESCRIPTION:

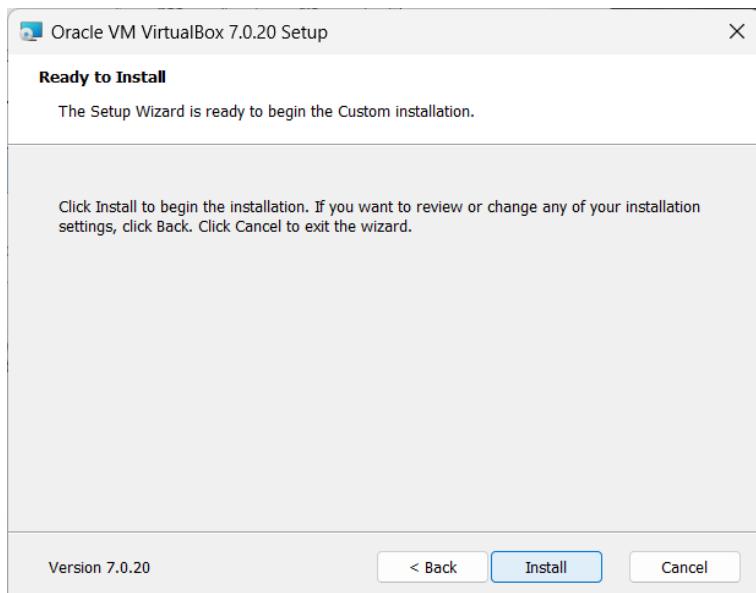
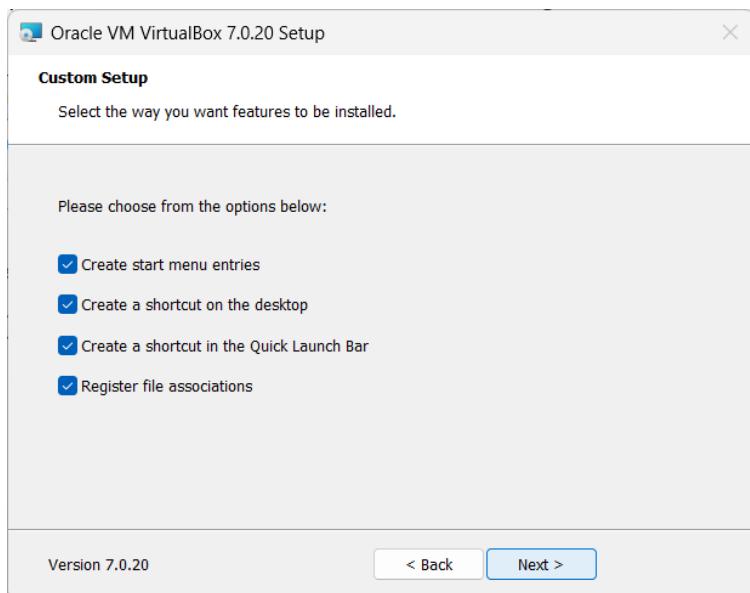
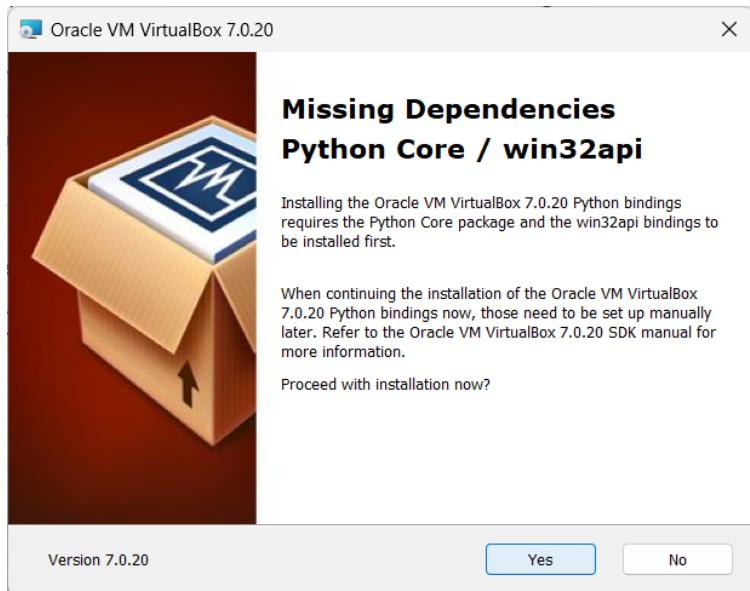
The task involves installing and configuring Hyper-V on a Windows Server 2012 environment. Hyper-V is a hypervisor-based technology that allows you to create and manage virtual machines on a single physical server. By enabling Hyper-V, the server can efficiently consolidate multiple workloads, reducing hardware costs and improving resource utilization. This setup provides a flexible and scalable infrastructure, ideal for testing, development, and production environments, enabling better management of computing resources and ensuring optimal performance across virtualized systems.

OUTPUT:

1. Downloading and installing Oracle VM VirtualBox in our PC:

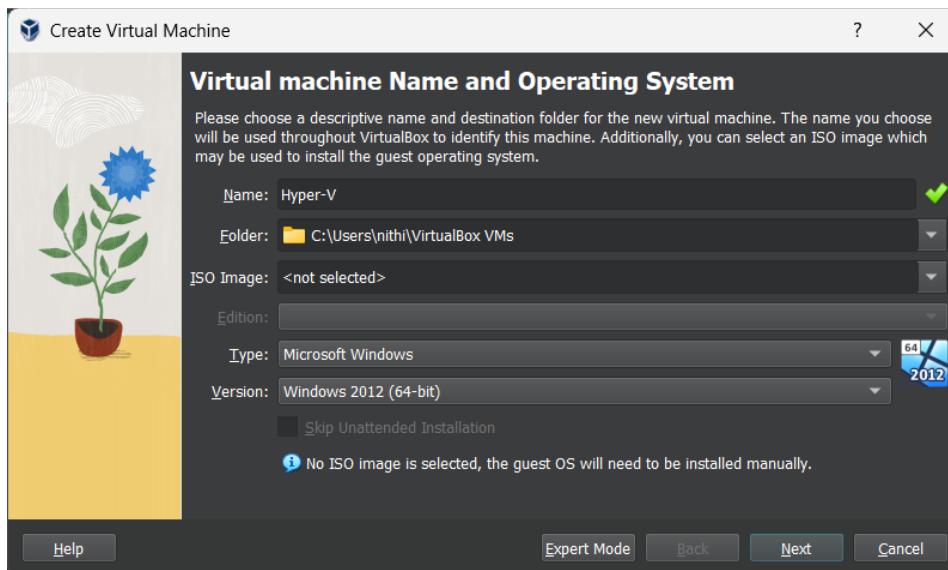
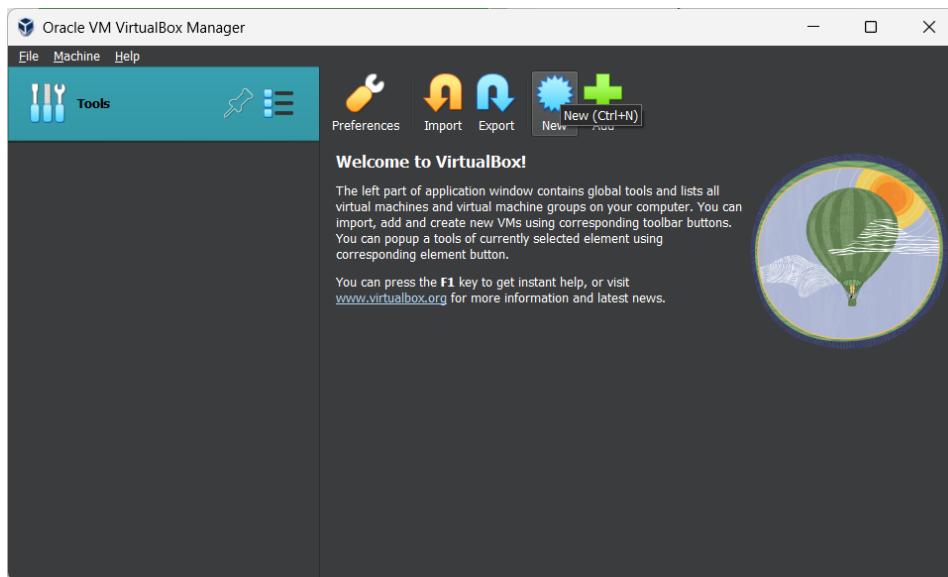


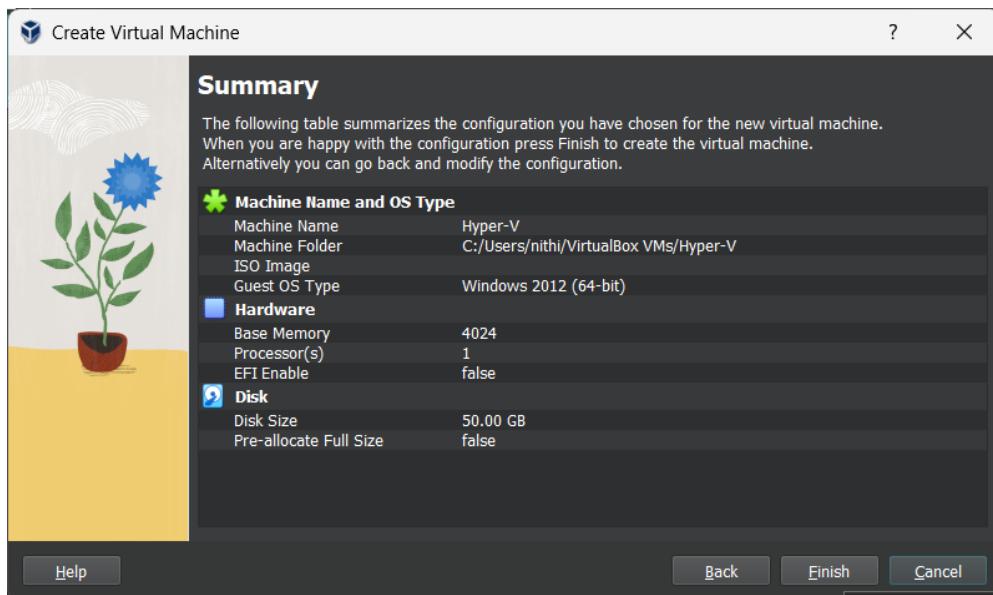
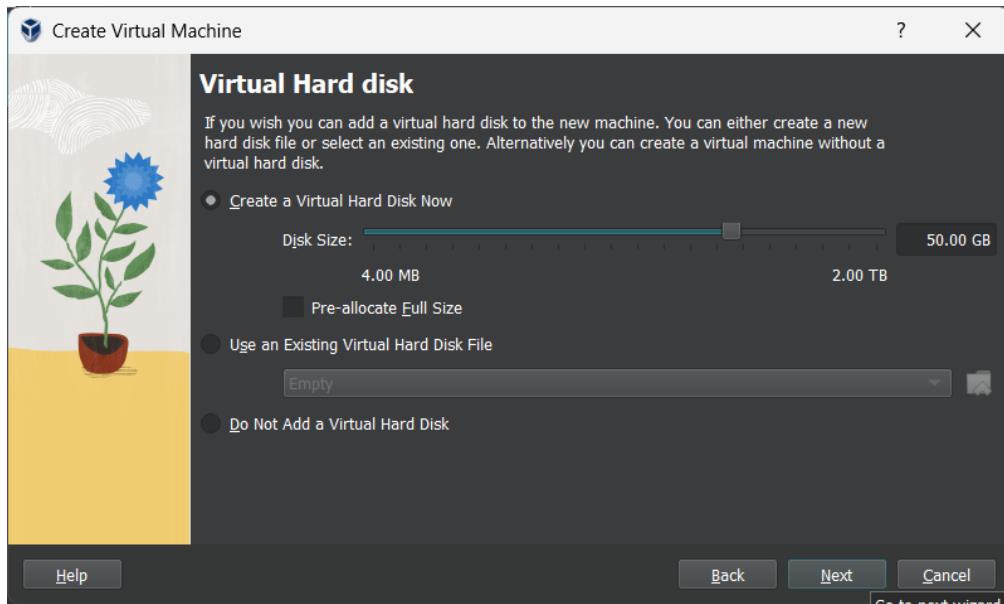
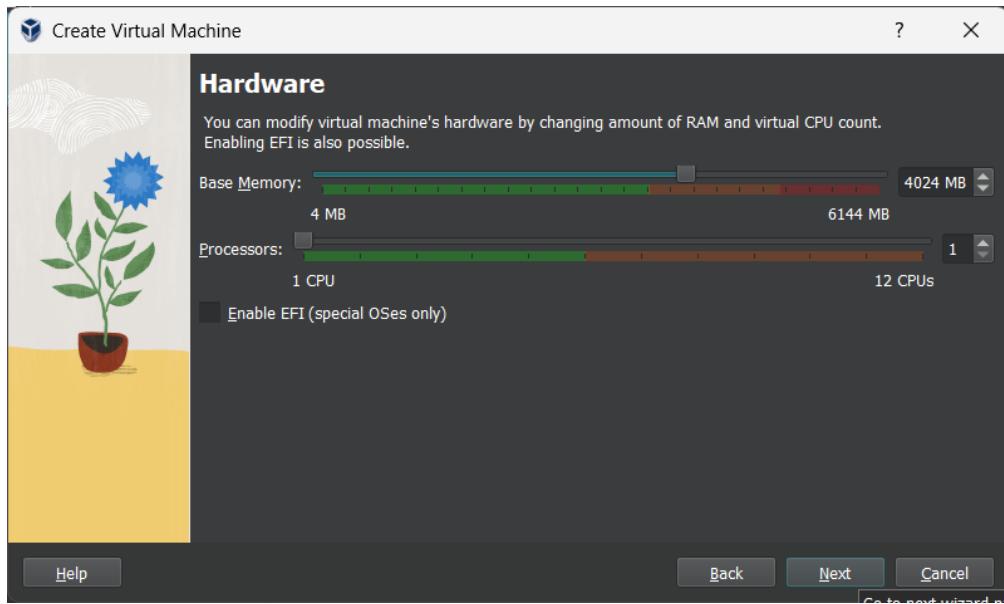


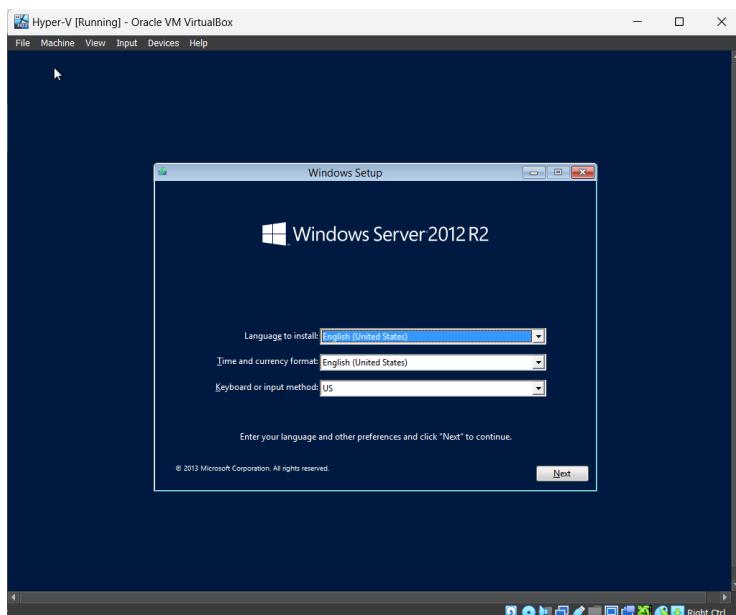
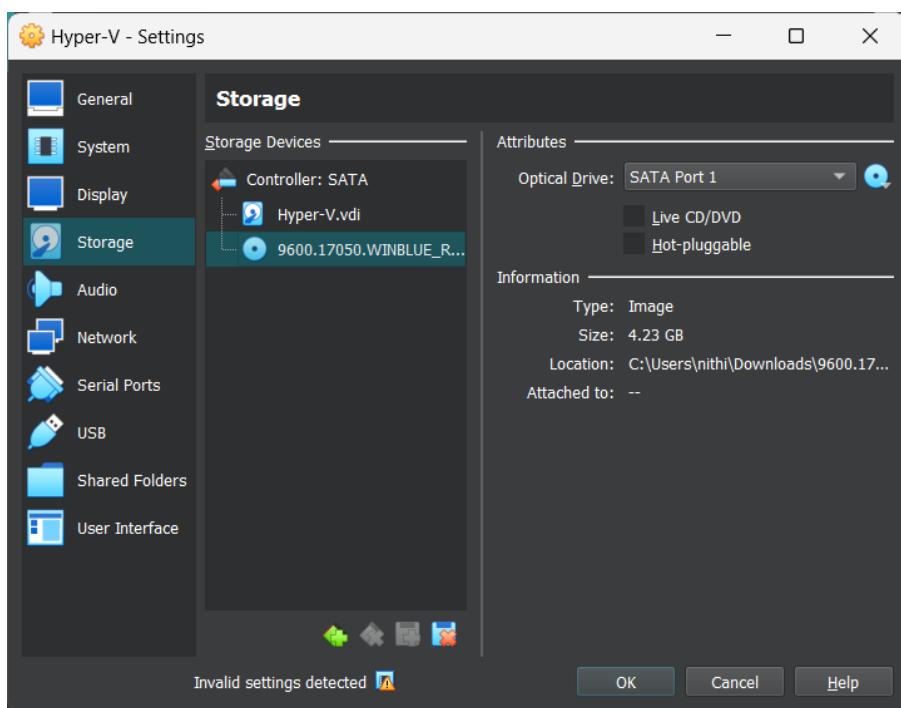
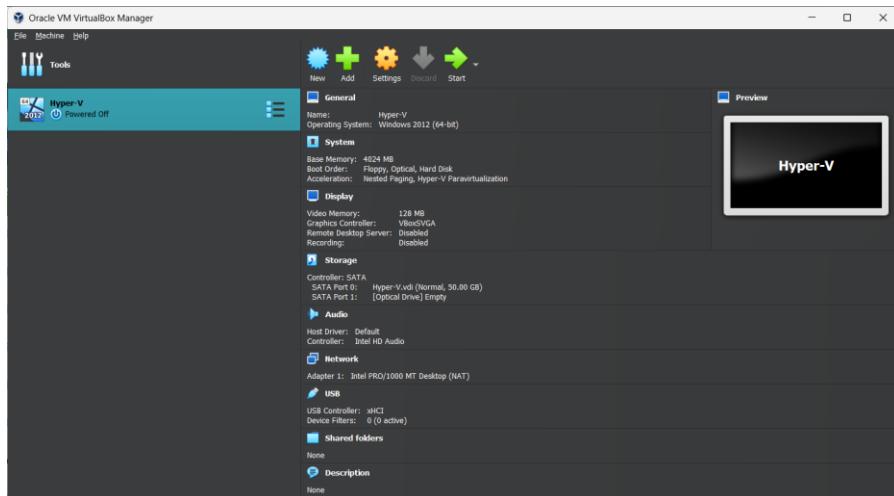


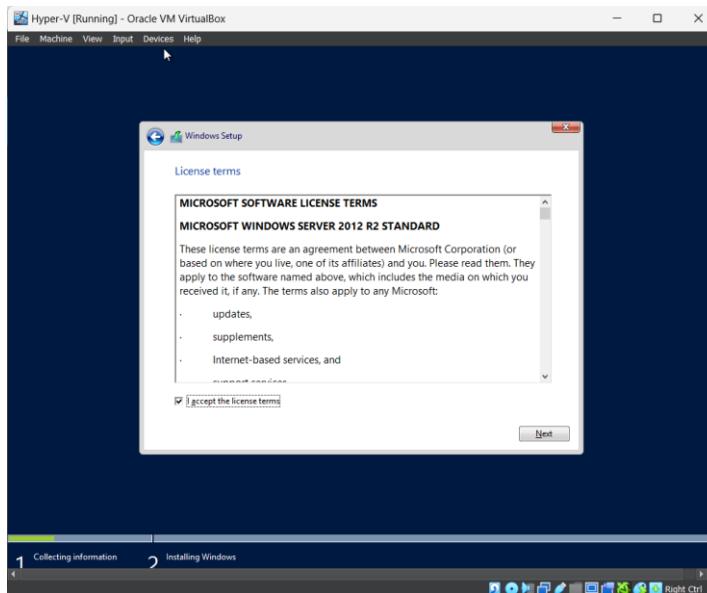
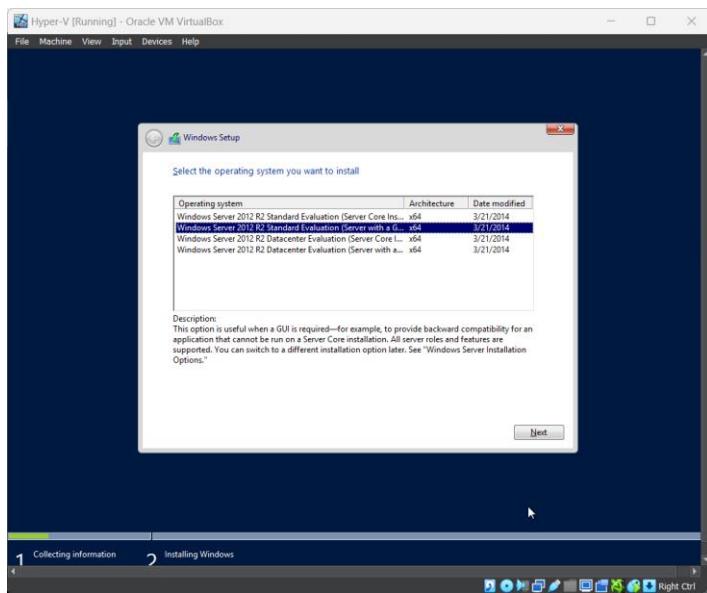
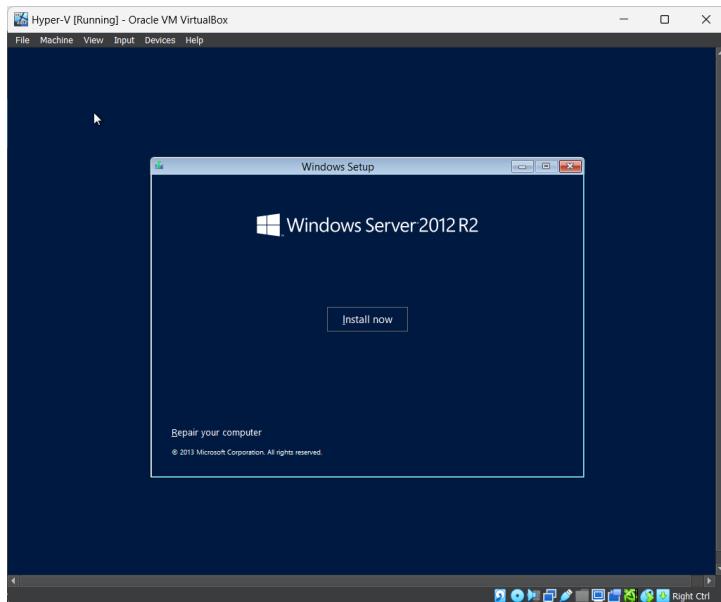


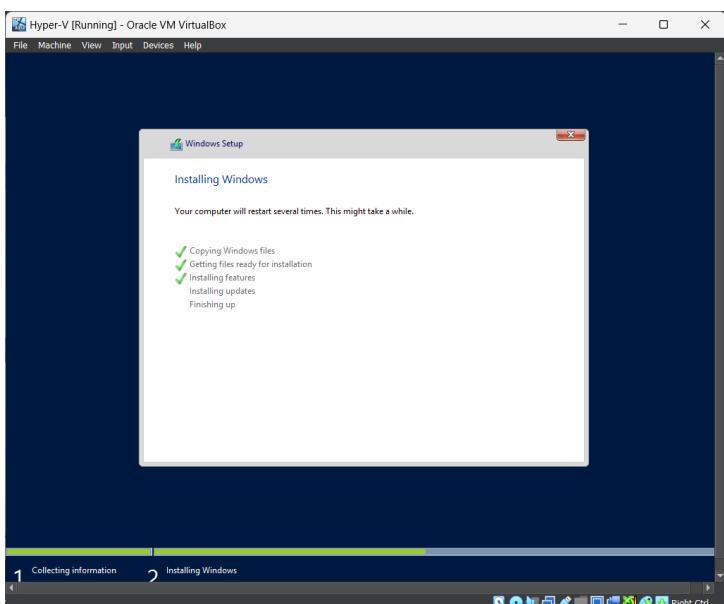
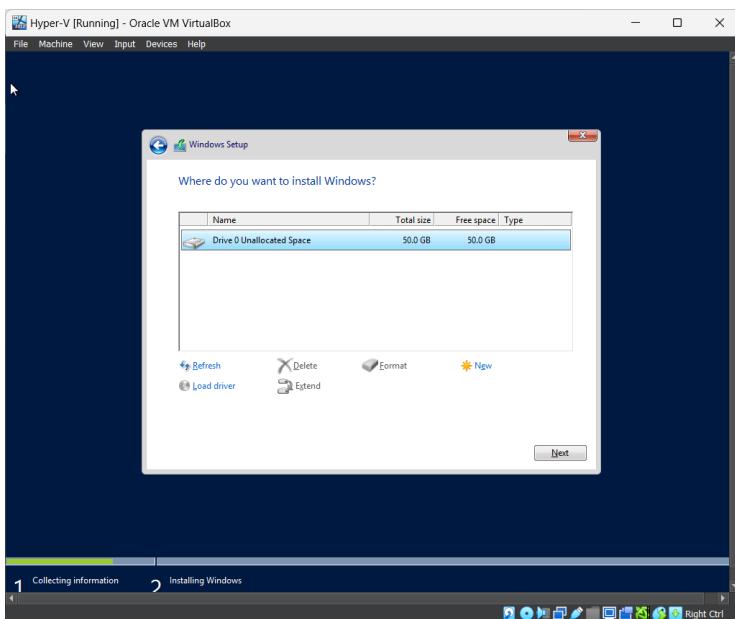
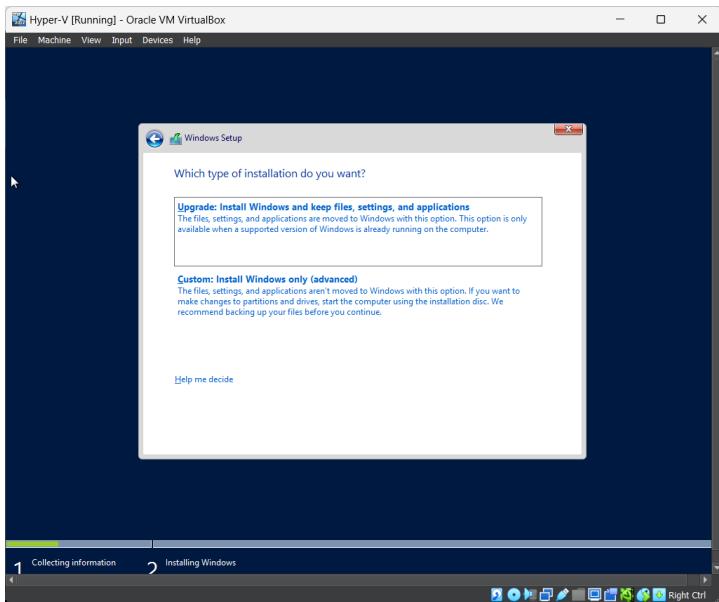
2. Installing Windows Server 2012 R2 in virtual Machine

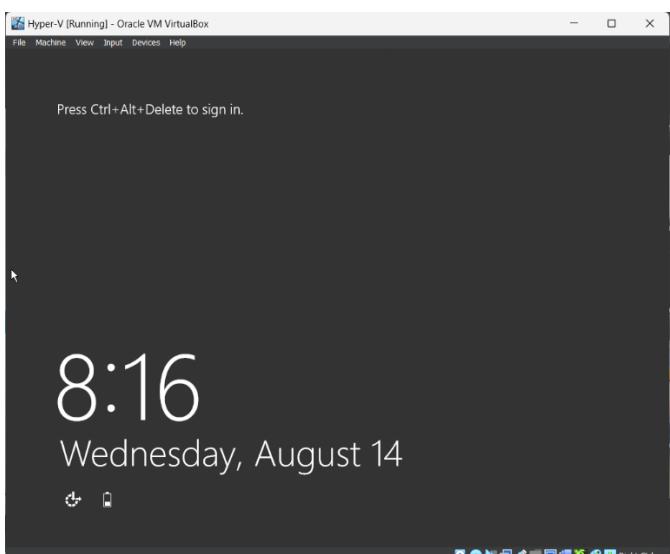
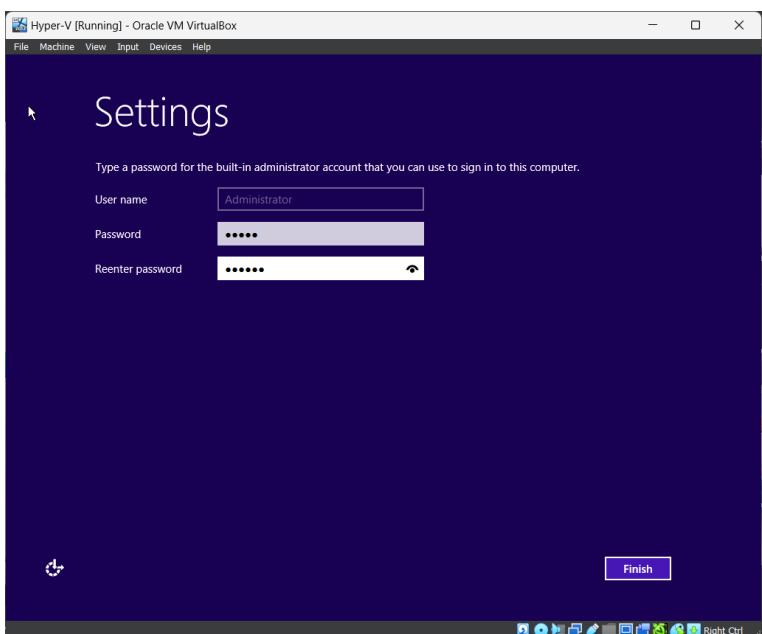
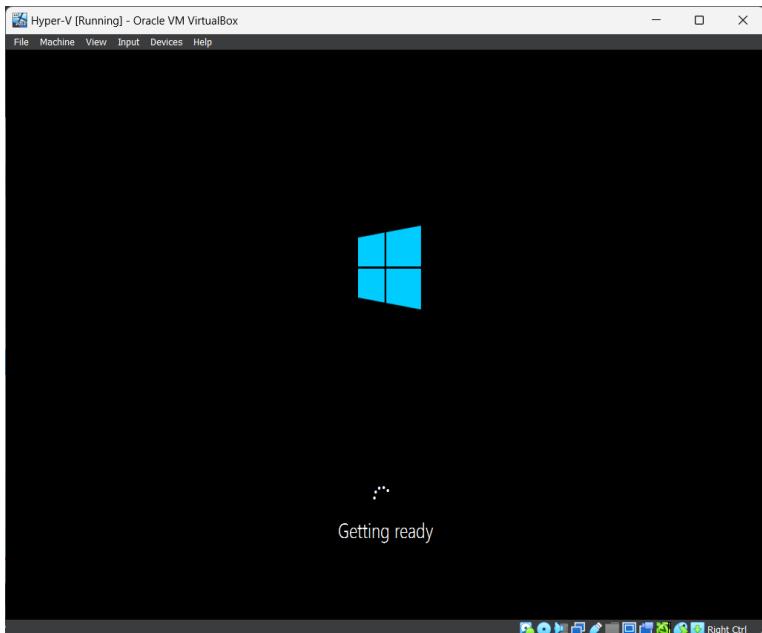




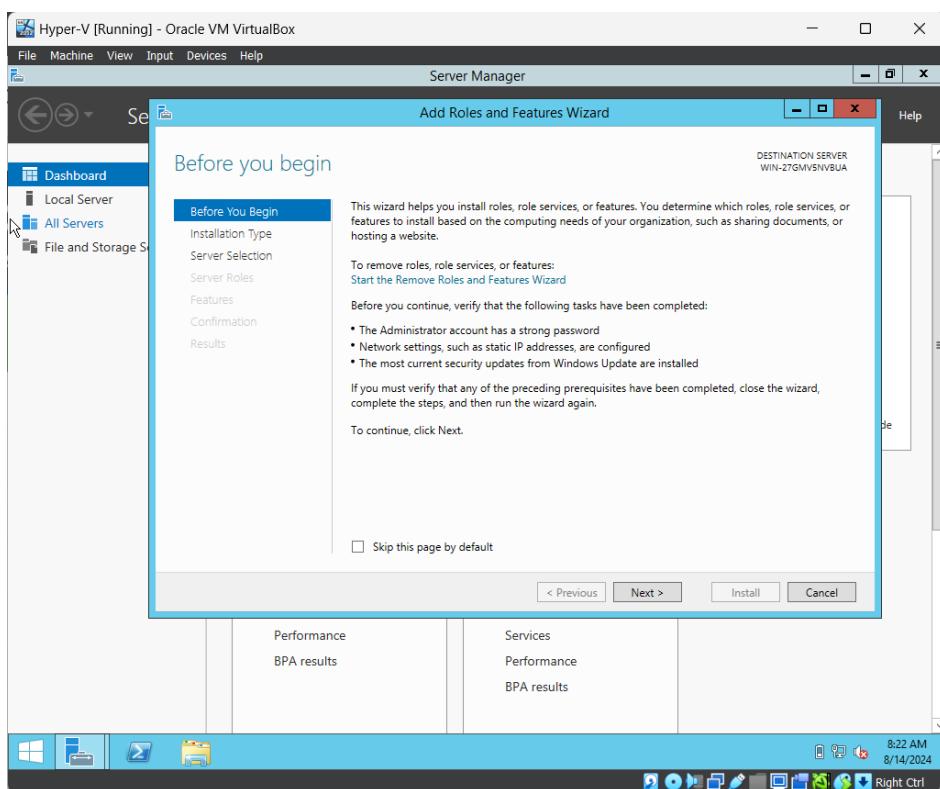
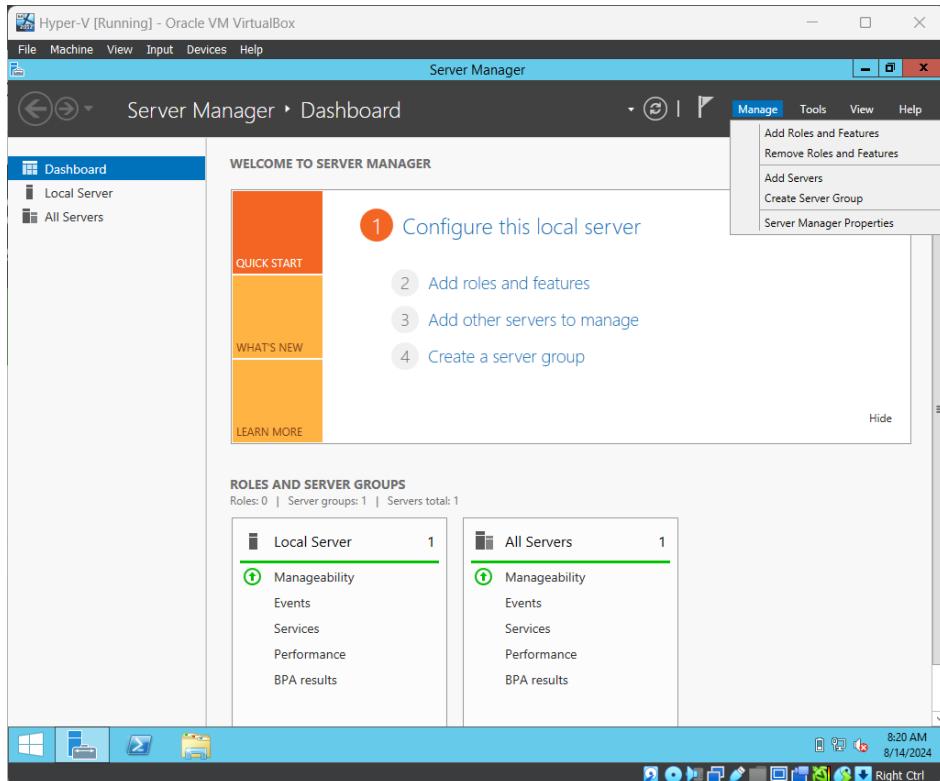


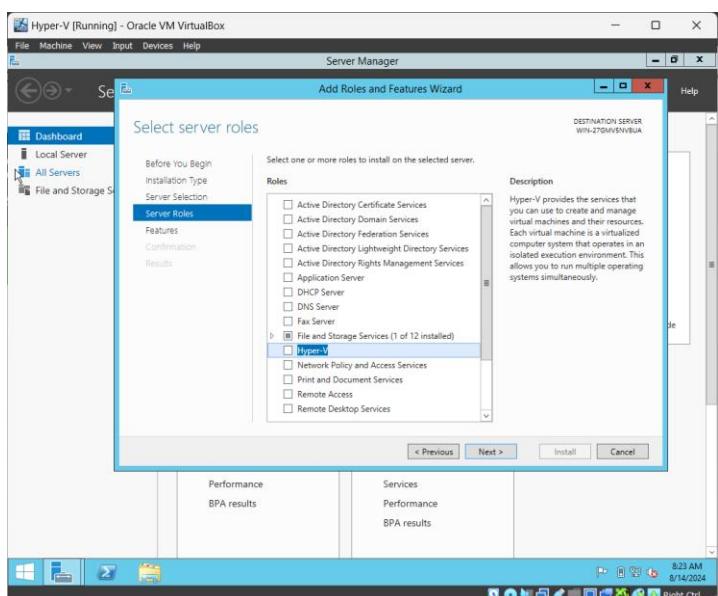
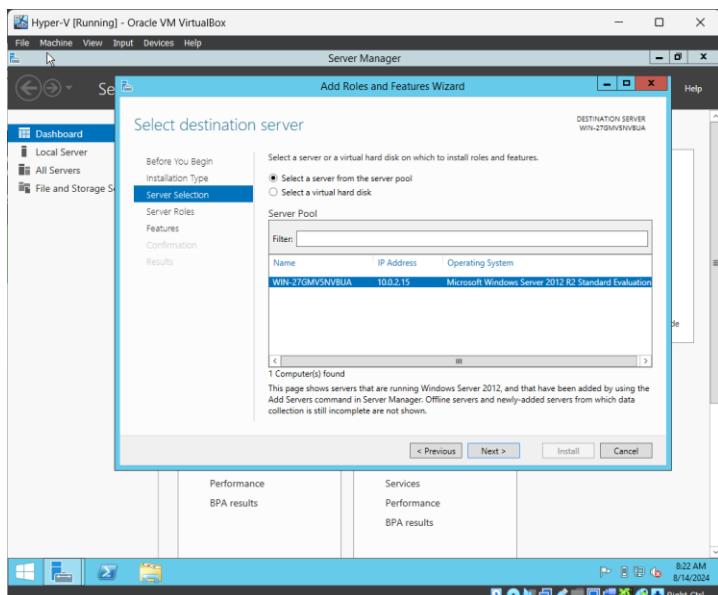
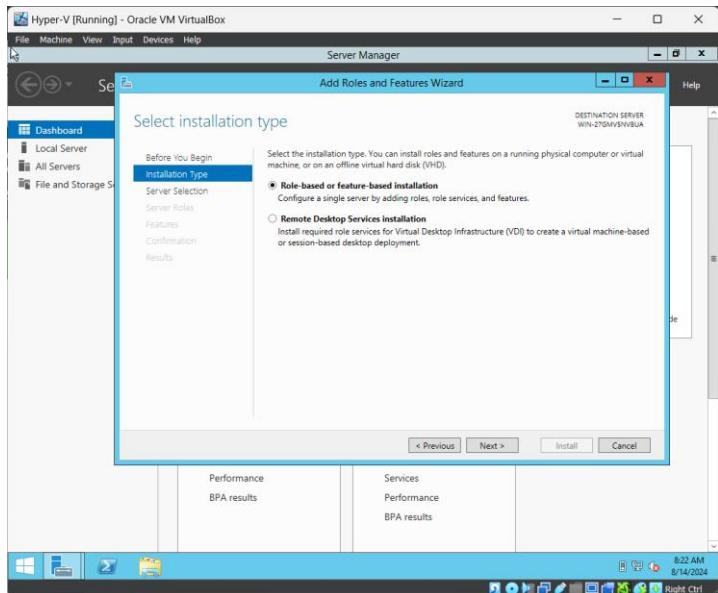


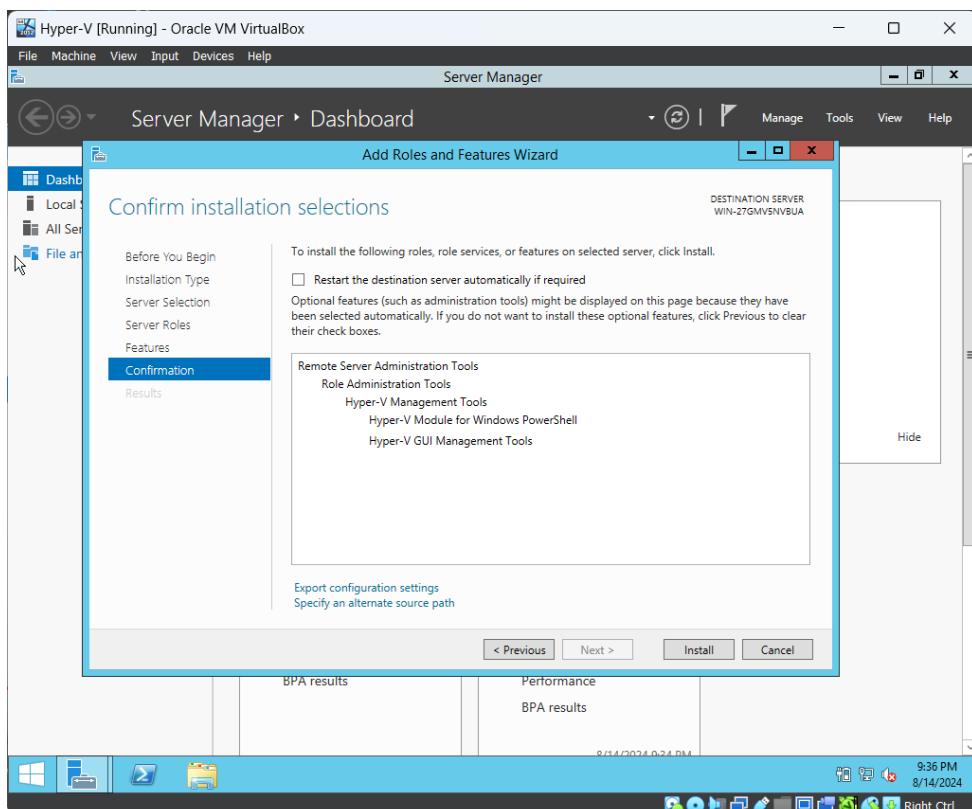
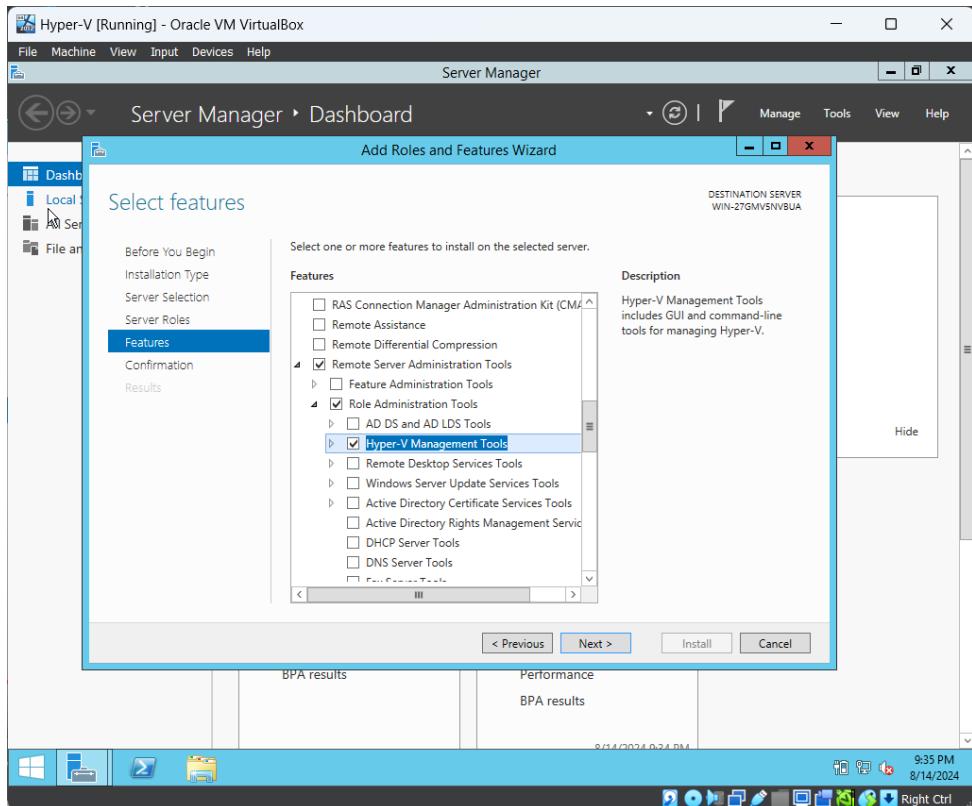


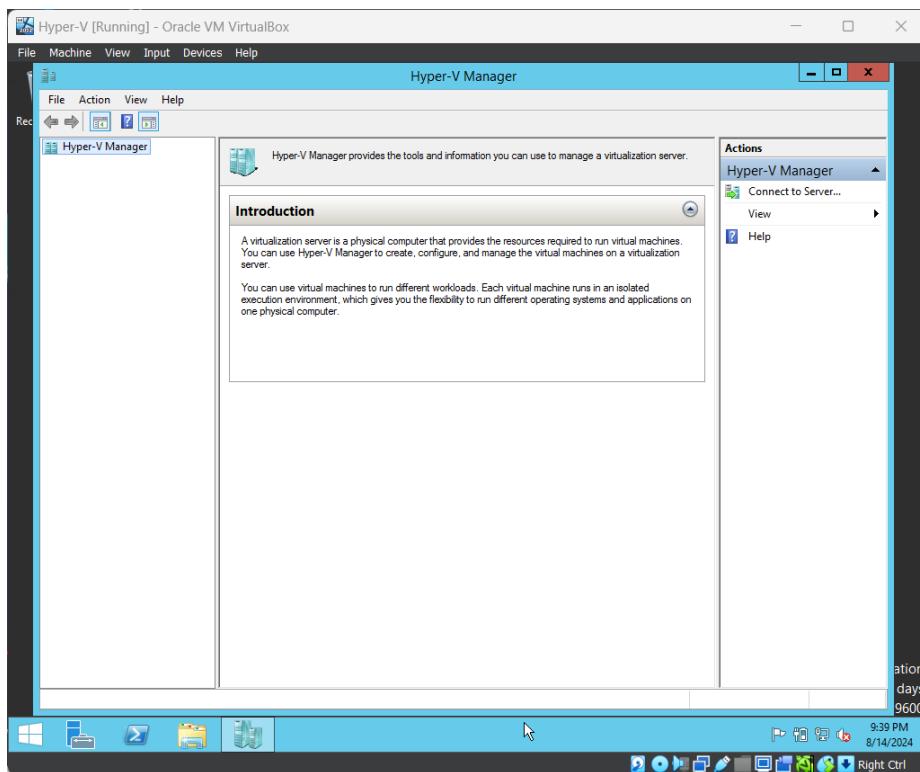
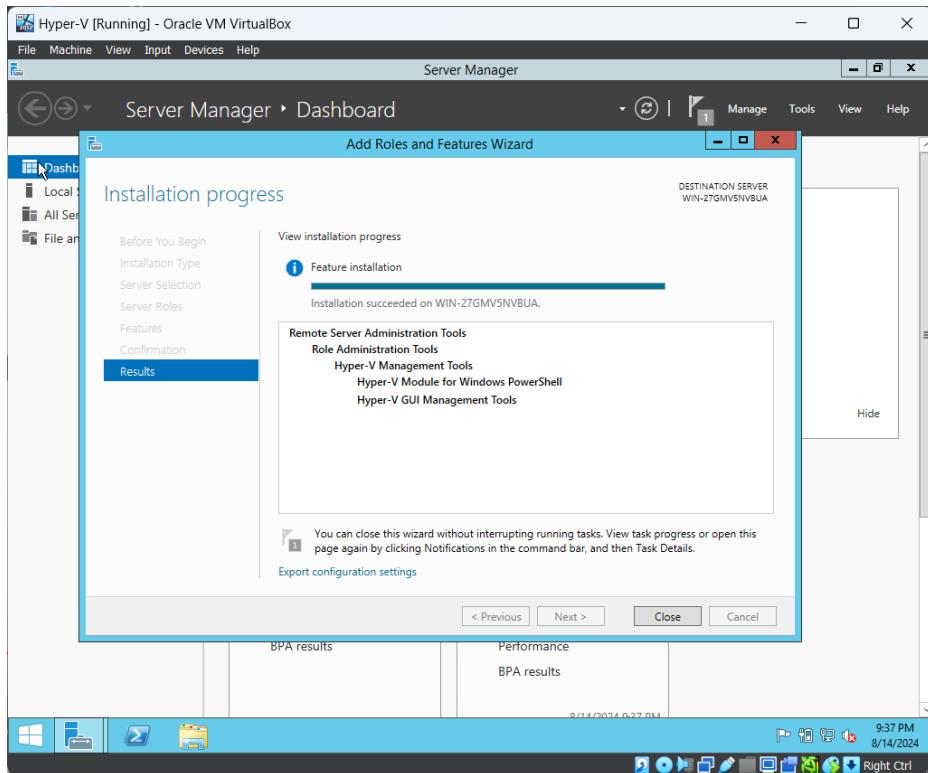


3. Installing Hyper-V in virtual machine:









RESULT:

The successful installation and configuration of Hyper-V on Windows Server 2012 enables the creation and management of virtual machines, optimizing resource utilization and providing a scalable virtualized environment.