

## INDEX

<b>S.NO</b>	<b>CONTENTS</b>	<b>PAGE NO</b>
I	Vision/Mission /PEOs/POs/PSOs	1
II	Syllabus	7
<b>Exp No:</b>	<b>List of Experiments</b>	
1	Create an AWS Free Tier account and budget alert to keep track of your AWS bill.	9
2	Configure and connect to an EC2 instance with manual settings.	18
3	Create an EC2 instance and access docker image to host a simple webpage.	32
4	Deploy a Machine learning model using AWS EC2	40
5	Create virtual storage with AWS S3 Bucket.	56
6	Create extended memory storage using Amazon EBS	62
7	Create an AWS IAM User with attached policy	82
8	Implement a role based access between AWS services	98
9	Migrate a website from local server to Cloud	102
10	Automate Machine Learning Model with AWS Amazon Sage maker	128



**Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda,  
Hyderabad – 500029**

**Department of Computer Science & Engineering (DS)**

**Vision of the Institution:**

To be the fountain head of latest technologies, producing highly skilled, globally competent engineers.

**Mission of the Institution:**

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
- To establish Industry Institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students



## KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY



(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda,  
Hyderabad – 500029

### Department of Computer Science & Engineering (DS)

#### **Vision of the Department:**

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

#### **Mission of the Department:**

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefits the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.



(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda,  
Hyderabad – 500029

## Department of Computer Science & Engineering(DS)

### PROGRAM OUTCOMES (POs)

**PO1: Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

**PO6: The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7: Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and

norms of the engineering practice.

**PO9: Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12: Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



## KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda,  
Hyderabad - 500029



### Department of Computer Science & Engineering(DS)

#### PROGRAM SPECIFIC OUTCOMES (PSOs)

**PSO1:** An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftment.

**PSO2:** Shall have expertise on the evolving technologies like Python, Machine Learning, Deep Learning, Internet of Things (IOT), Data Science, Full stack development, Social Networks, Cyber Security, Big Data, Mobile Apps, CRM, ERP etc.



**KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

**(AN AUTONOMOUS INSTITUTE)**

**Accredited by NBA & NAAC, Approved by AICTE, Affiliated to  
JNTUH, Narayanguda, Hyderabad – 500029**

**Department of Computer Science & Engineering(DS)**

### **PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

**PEO1:** Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

**PEO2:** Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

**PEO3:** Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

**PEO4:** Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

## EXPERIMENT 1

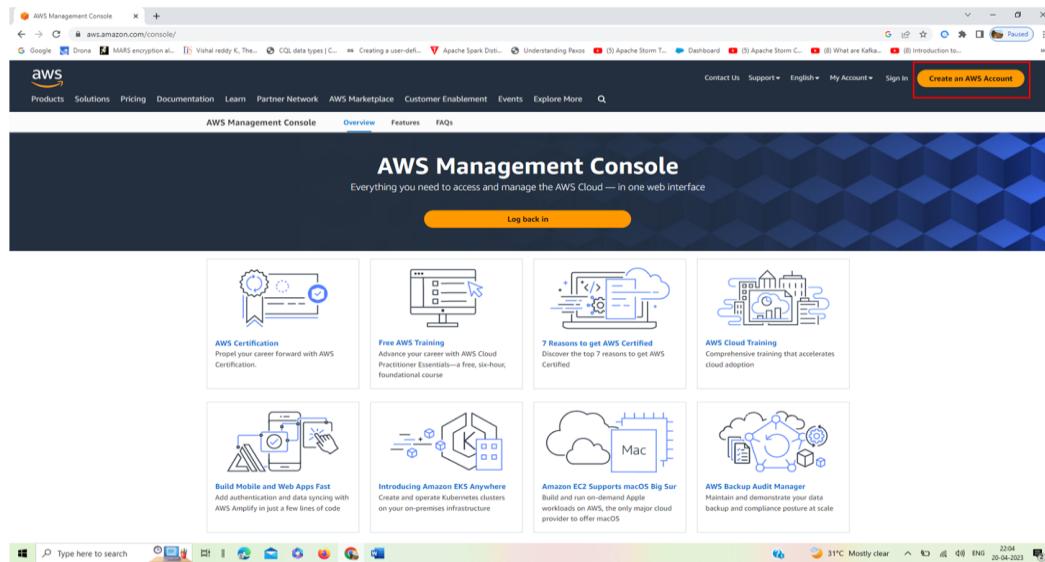
### Create an AWS Free Tier account and budget alert to keep track of your AWS bill.

**AIM:** To create an AWS Free Tier account and budget alert to keep track of your AWS bill.

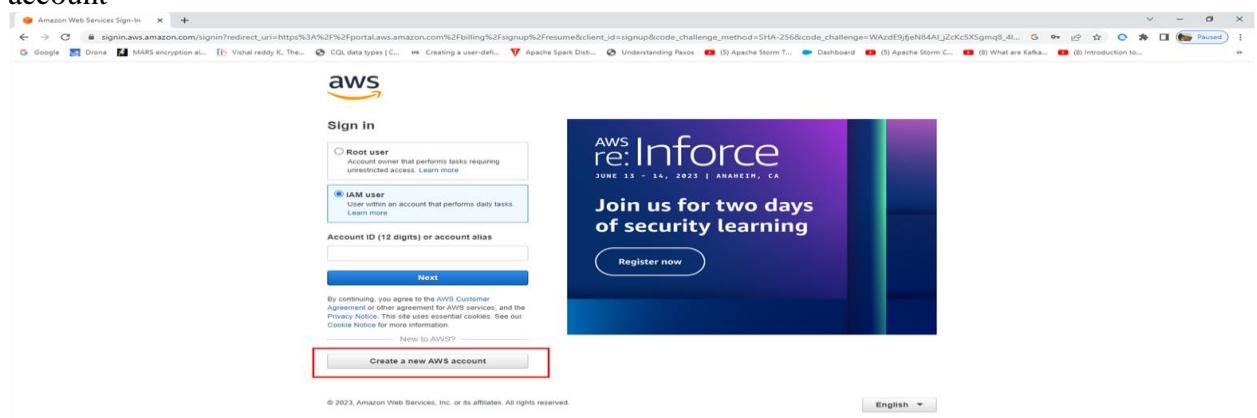
**REQUIREMENTS:** AWS active account and desktop.

#### **PROCEDURE:**

1. Navigate to <https://aws.amazon.com/>

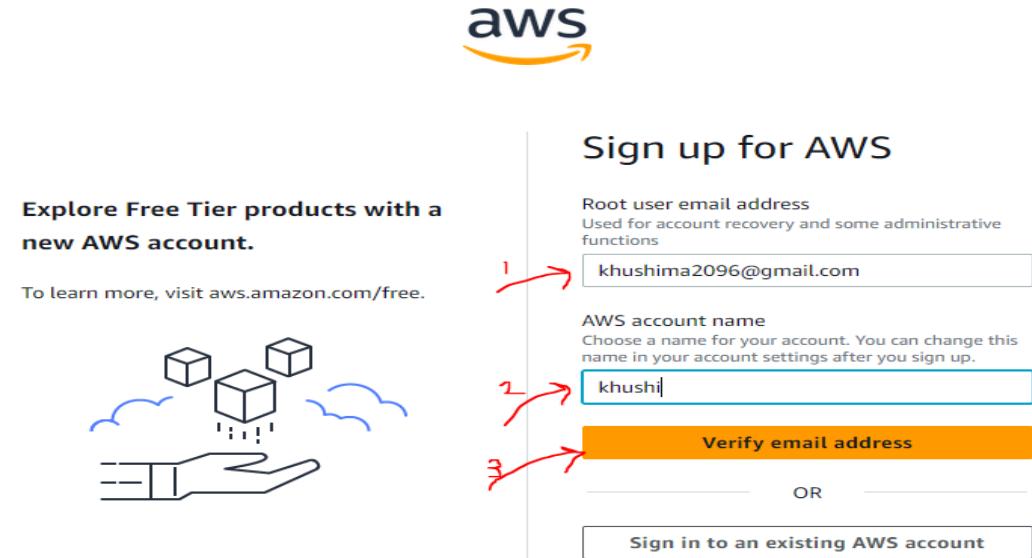


2. click on sign up for new AWS account

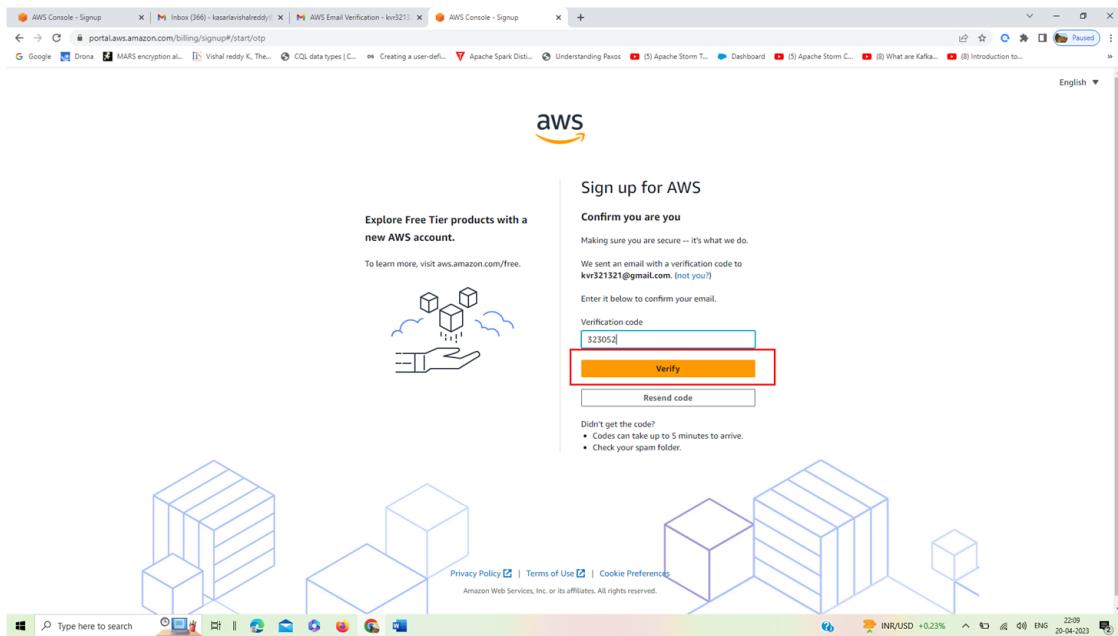


3. follow the below steps

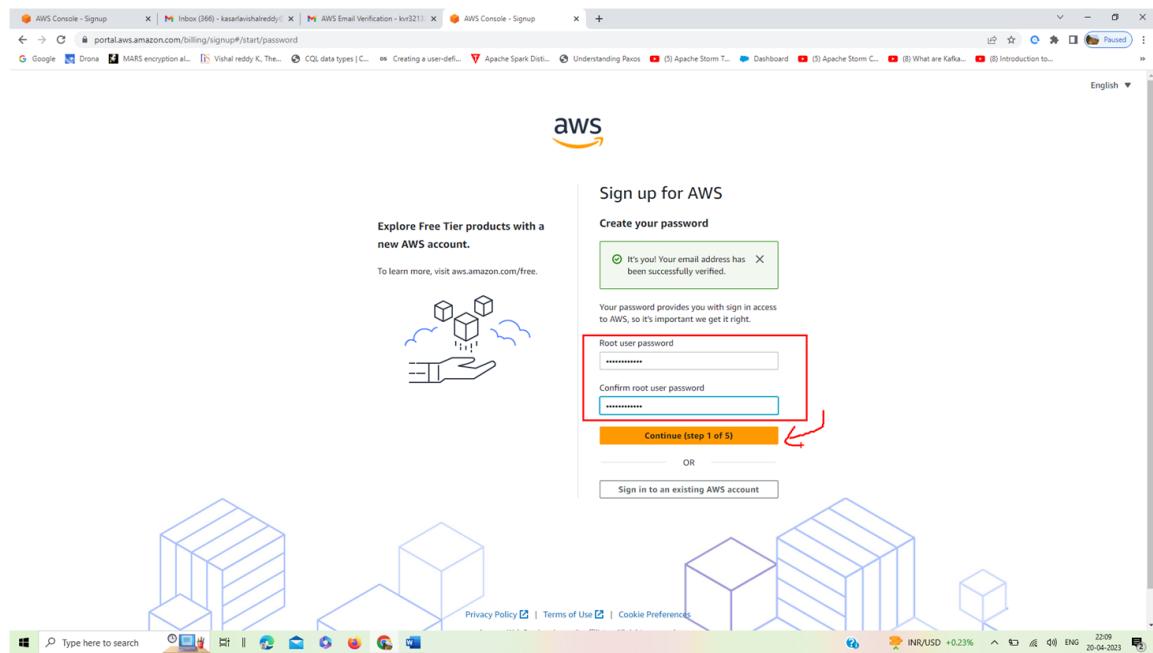
- i) enter email id
- ii) give an name for your account profile
- iii) verify the email address



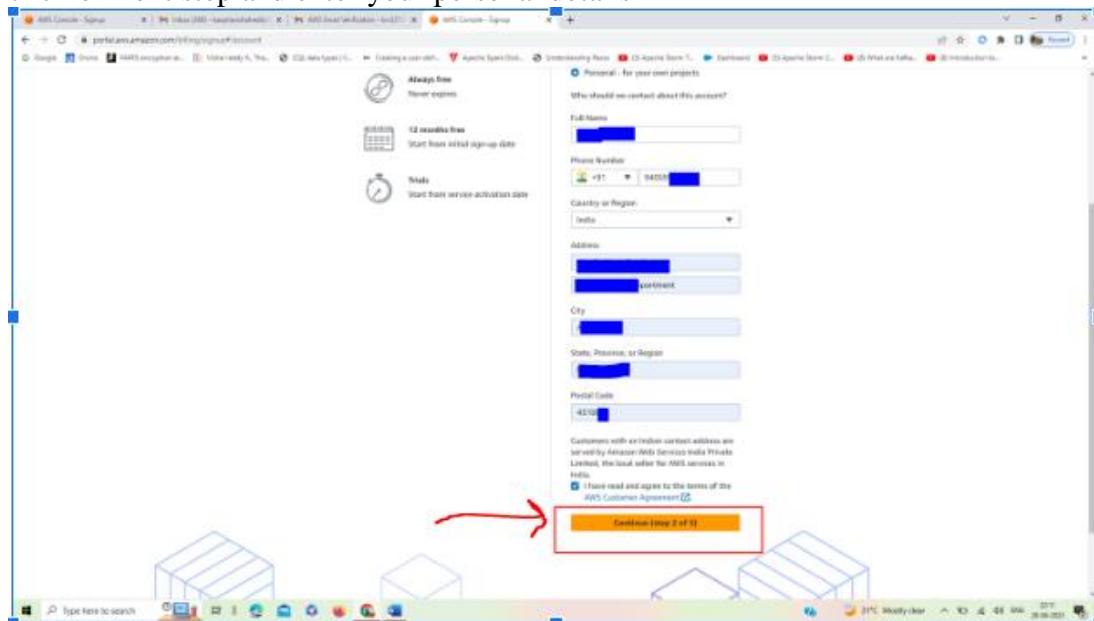
#### 4. verify the mail address with the verification code



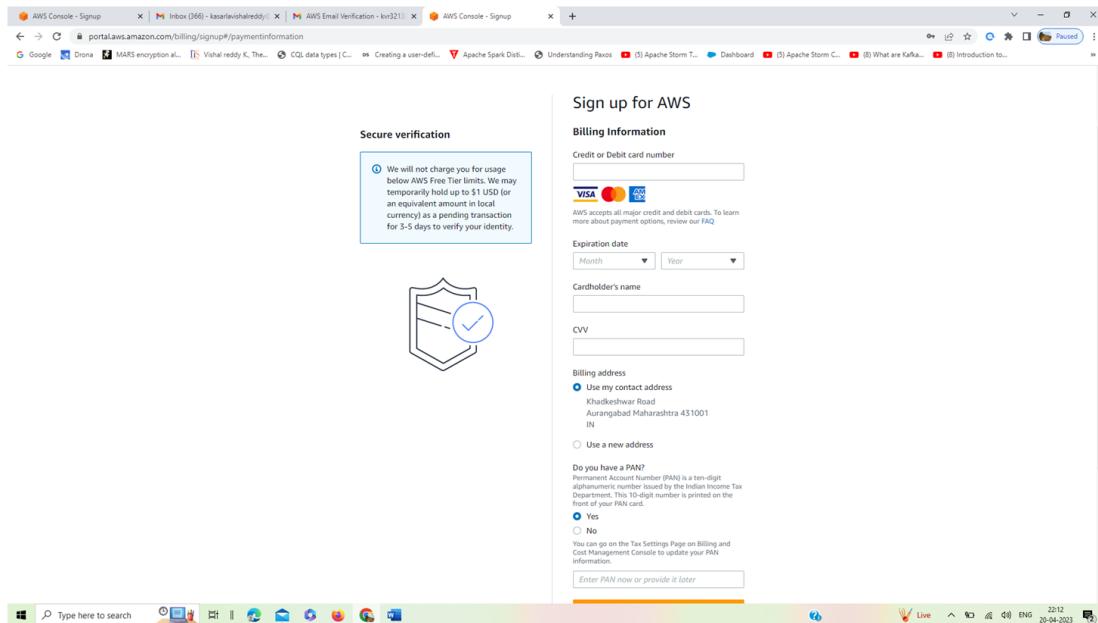
5. once verified enter login password for future use.



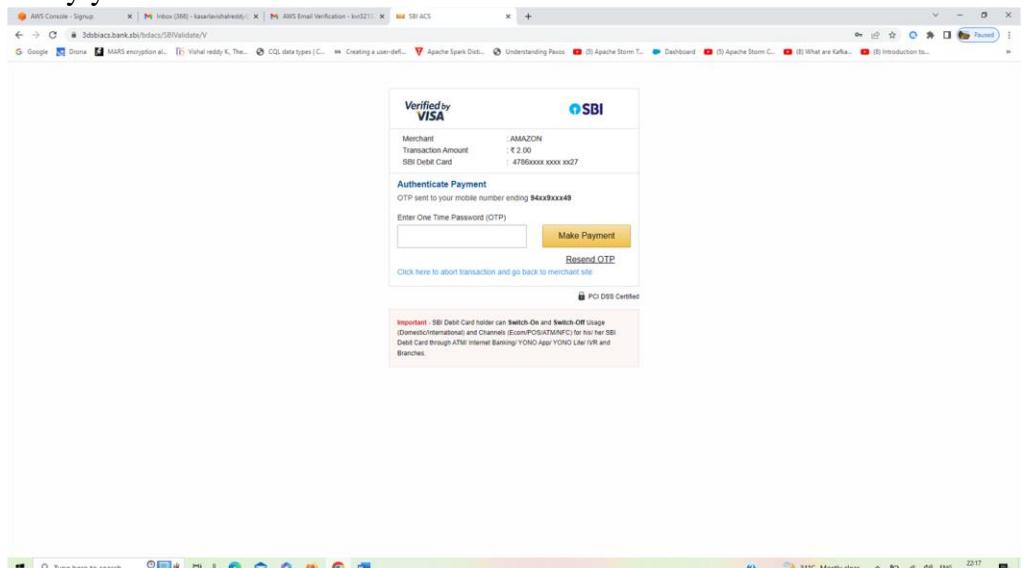
6. click on next step and enter your personal details



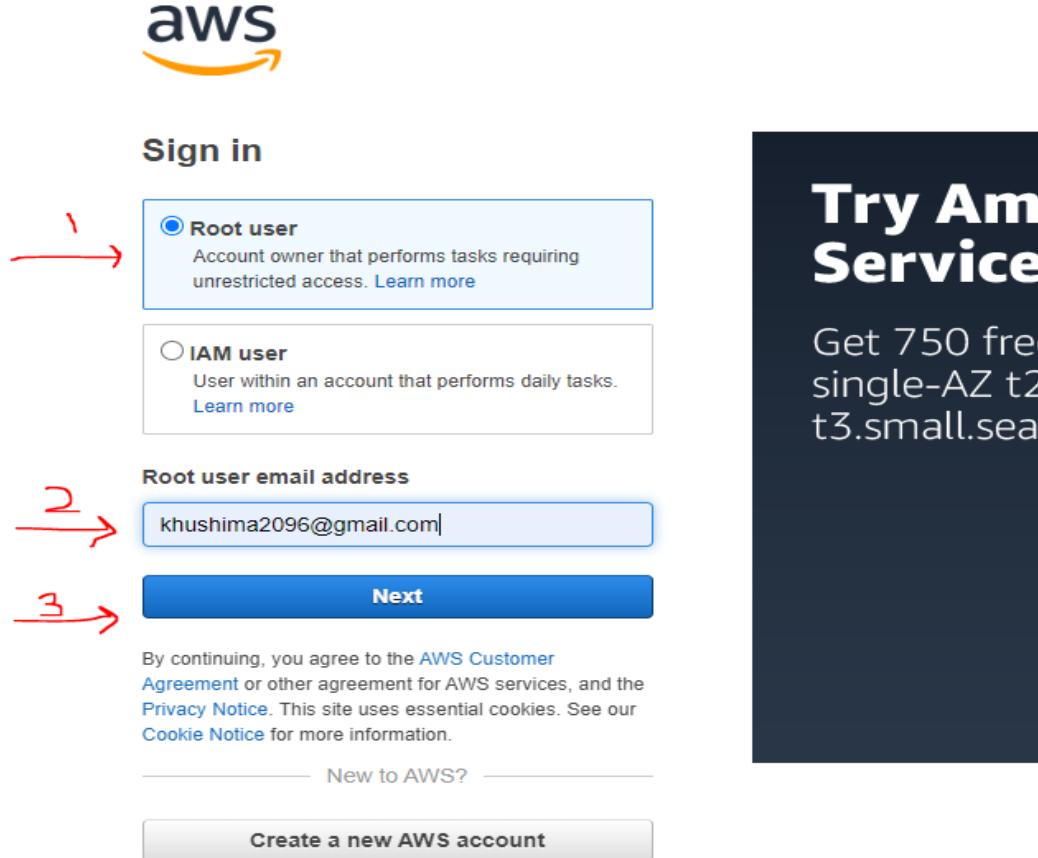
7. click on continue and enter the card details



8. verify your card details and click on continue

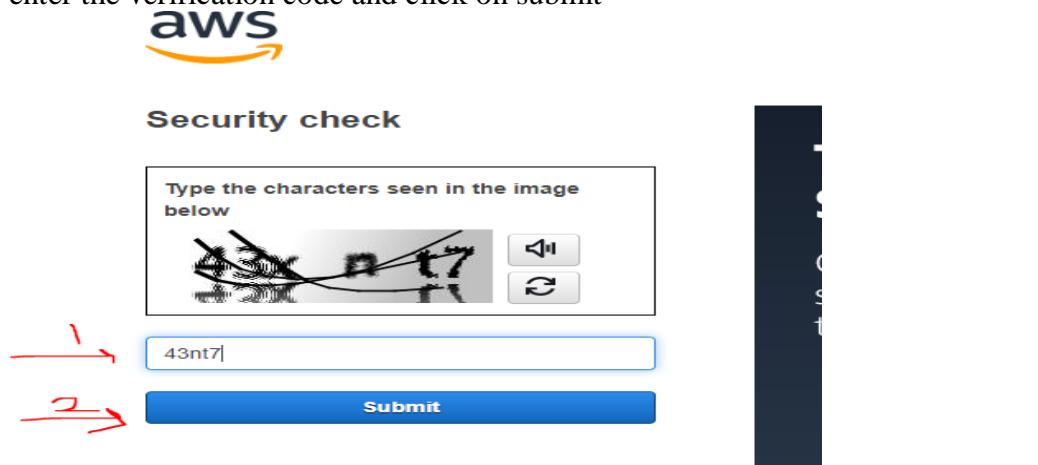


9. make payment and sign into your account
  - i) select root user
  - ii) enter your registered mail id
  - iii) click on next



The image shows the AWS Sign In page. At the top is the AWS logo. Below it is a "Sign in" button. The main area has two radio button options: "Root user" (selected) and "IAM user". A red arrow labeled "1" points to the "Root user" option. Below these are fields for "Root user email address" containing "khushima2096@gmail.com" (red arrow labeled "2") and a "Next" button (red arrow labeled "3"). At the bottom, there is a terms and conditions agreement section with links to "AWS Customer Agreement", "Privacy Notice", and "Cookie Notice". There is also a "New to AWS?" link and a "Create a new AWS account" button.

10. enter the verification code and click on submit

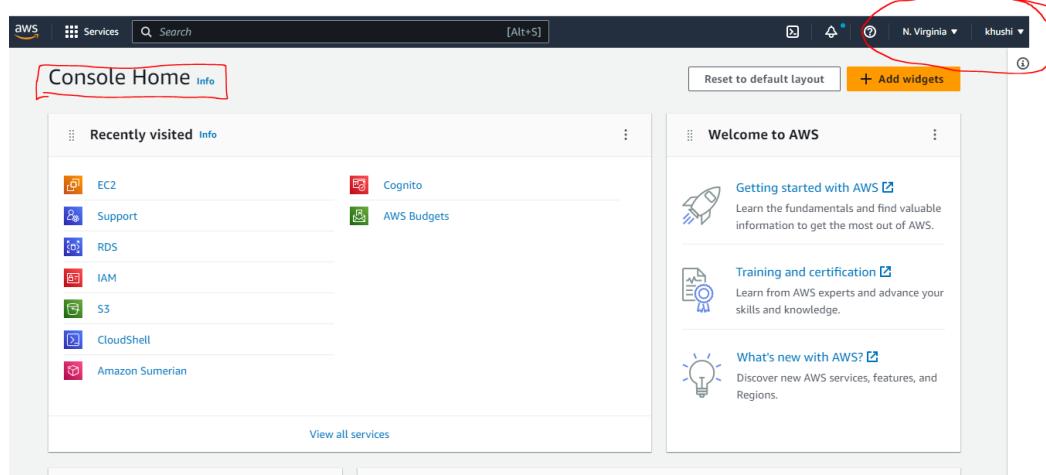


The image shows the AWS Security check page. At the top is the AWS logo. Below it is a "Security check" button. The main area contains a CAPTCHA challenge: "Type the characters seen in the image below" with a distorted text "43nt7" and a refresh button. A red arrow labeled "1" points to the CAPTCHA input field containing "43nt7" (red arrow labeled "2") and a "Submit" button (red arrow labeled "3"). To the right of the form is a dark sidebar.

11. enter password and click on sign in.



12. the AWS management Console will open



13. in order to see the bills search for billing

Search results for 'billing'

**Billing**

Access, analyze, and control your AWS costs and usage.

**AWS Billing Conductor**

Simplifying your billing practice

**AWS Wickr**

Secure communication with end-to-end encryption

**IoT TwinMaker**

Easily create digital twins of real-world systems to optimize operations

14. click on billing in order to see the dashboard

**AWS Billing Dashboard**

Page refresh time: Friday, April 21, 2023 at 1:04:21 PM GMT+5:30

**AWS summary**

Current month's total forecast	Current MTD balance	Prior month
No data to display	No data to display	No data to display
Total number of active services	Total number of active AWS accounts	Total number of active AWS accounts
2	No data to display	2

15. click on bills in order to view bills month wise

**Bills**

**Bills**

Page refresh time: Friday, April 21, 2023 at 1:05:53 PM GMT+5:30

**AWS estimated bill summary**

Total charges and payment information

Account ID	Billing period info	Bill status info
612447784518	April 1 - April 30, 2023	Pending

**Amazon Web Services India Private Limited**

**Estimated grand total:** **USD 0.00**

you can change the month in order to view ant previous bills

16. you can make the payments pending from here

Bills [Info](#)

Page refresh time: Friday, April 21, 2023 at 1:08:38 PM GMT+5:30

[Download all to CSV](#) [Print](#) [January 2023 ▾](#) [⚙️](#)

**AWS bill summary [Info](#)**  
Total charges and payment information

Account ID: 612447784518 | Billing period [Info](#): January 1 - January 31, 2023 | Bill status [Info](#): **Issued 02/02/2023**

Service provider: Amazon Web Services India Private Limited | Total in USD: **USD 0.13**

Grand total: **USD 0.13**

[▶ Payment information \[Info\]\(#\)](#)

17. you can also view the breakup of the billing charges here

Amazon Web Services India Private Limited charges by service <a href="#">Info</a>		
<a href="#">Expand all</a>		<a href="#">Settings</a>
Total active services <b>6</b>		Total pre-tax service charges in USD <b>USD 0.11</b>
<input type="text"/> Filter by service name or region name		< 1 >
Description Usage Quantity Amount in USD		
Relational Database Service		USD 0.11
CloudWatch		USD 0.00
Data Transfer		USD 0.00
Elastic Compute Cloud		USD 0.00
Service Catalog		USD 0.00
Simple Storage Service		USD 0.00
<b>Total tax</b>		<b>USD 0.02</b>

18. once you have made payment click on the payments option

AWS Billing > Payments

**Payments**

**Payments due** **Unapplied funds** **Transactions**

**Payments due summary** [Info](#)

No payments due summary  
No payments due summary to display.

**Payments due (0)**

**Download** **Cancel payment**

Filter due payments by text, property or value

Issued date ▾ ID ▾ Type ▾ Due date ▾ Status ▾ Currency ▾ Amount ▾

you can also view the previous transactions and download them.

19. click on free tier in order to see free tier AWS available

AWS Billing > AWS Free Tier

**AWS Free Tier** [Info](#)

**Summary (3)**

Service	AWS Free Tier usage limit	Current usage	Forecasted usage	MTD actual
Amazon Elastic Compute Cloud	30.0 GB-Mo for free for 12 months as part of AWS Free Usage Tier (Global-EBS:VolumeUsage)	16 GB-Mo	23 GB-Mo	—
Amazon Elastic Compute Cloud	750.0 Hrs for free for 12 months as part of AWS Free Usage Tier (Global-BoxUsage:freetier:micro)	0 Hrs	0 Hrs	—
AWS Data Transfer	100.0 GB for free per month as part of AWS Free Usage Tier (Global-DataTransfer-Out-Bytes)	0 GB	0 GB	—

## EXPERIMENT 2

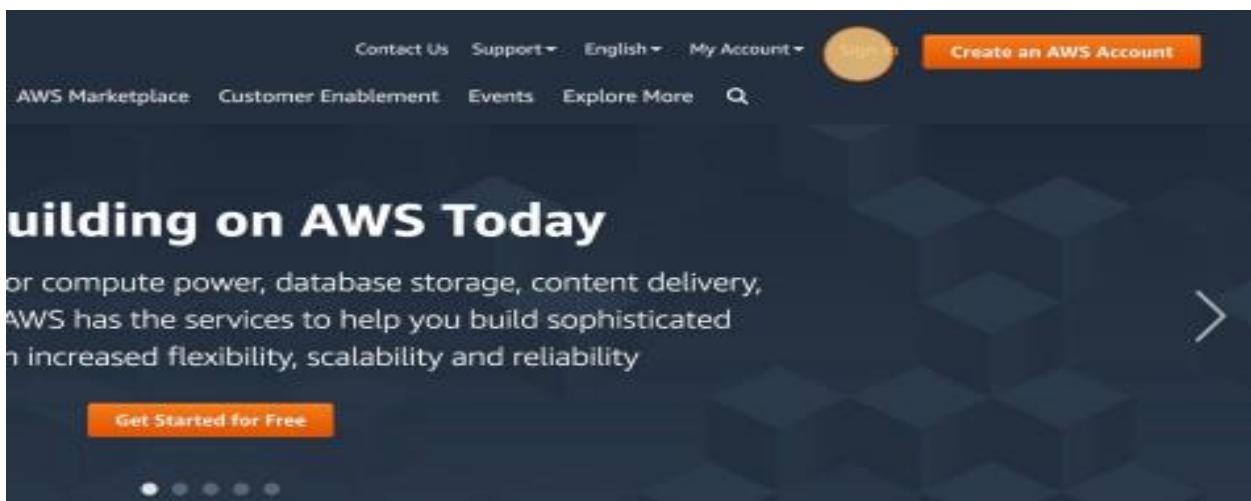
### Configure and connect to an EC2 instance with manual settings.

**AIM:** Establish an AWS account. Use the AWS Management Console to launch an EC2 instance and connect to it.

**REQUIREMENTS:** AWS active account and desktop.

#### **PROCEDURE:**

1. Navigate to <https://aws.amazon.com/>
2. Click "Sign In"

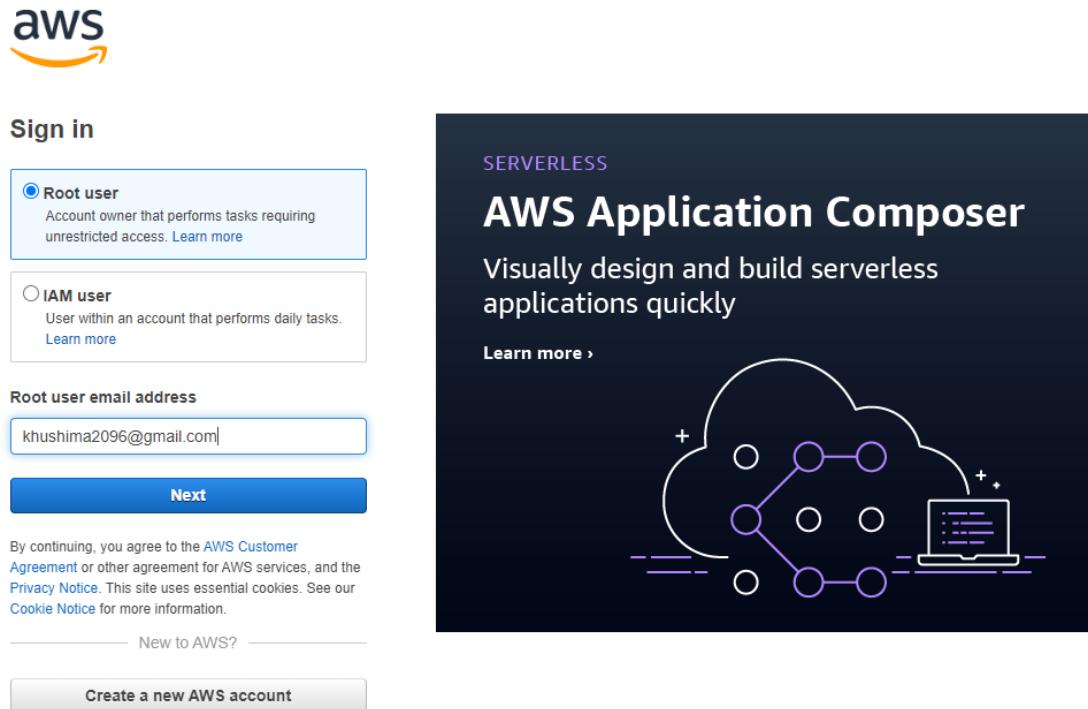


3. Click the "Root user" radio button.

The screenshot shows the first step of the "Create a new AWS account" wizard. It asks the user to choose their account type. Two options are available: "Root user" (selected) and "IAM user". The "Root user" option is described as an "Account owner that performs tasks requiring unrestricted access". The "IAM user" option is described as a "User within an account that performs daily tasks". Below this is a field for "Root user email address" containing "username@domain.com" and a yellow circular placeholder icon. A blue "Next" button is at the bottom. At the very bottom, there is a small legal notice about cookie usage and a "Create a new AWS account" link.

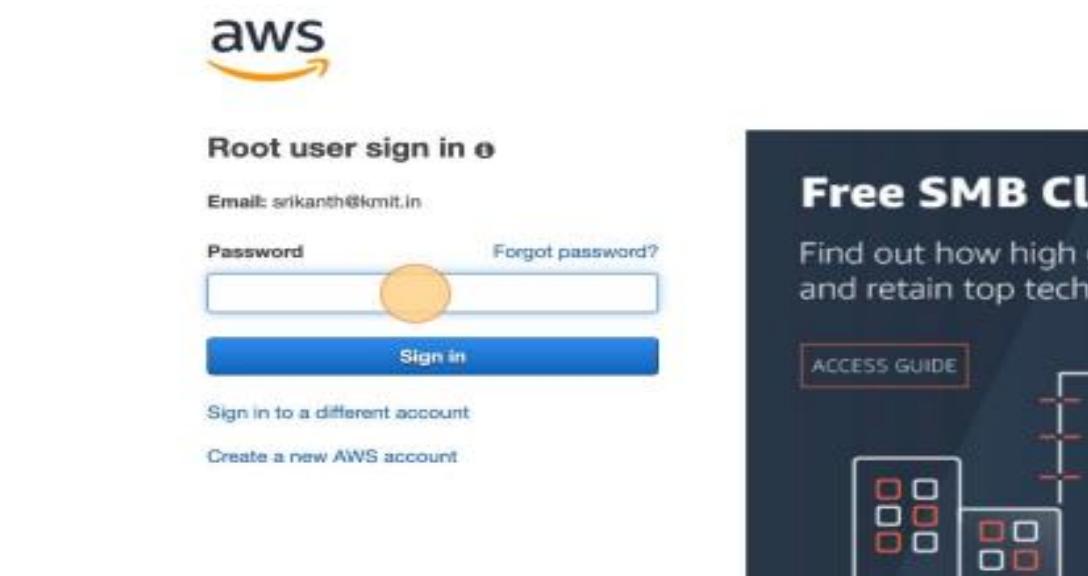


4. Type "Username you have registered for free tier"
5. Click "Next"



The image shows two side-by-side screenshots. On the left is the AWS sign-in page. It features the AWS logo at the top, followed by a "Sign in" heading. There are two radio button options: "Root user" (selected) and "IAM user". Below each option is a brief description and a "Learn more" link. A text input field for "Root user email address" contains the value "khushima2096@gmail.com". A large blue "Next" button is centered below the input field. At the bottom of the sign-in form, there is a small paragraph about AWS Customer Agreement and Privacy Notice, followed by a "New to AWS?" link and a "Create a new AWS account" button. On the right is a screenshot of the AWS Application Composer landing page. It has a dark background with the word "SERVERLESS" in purple at the top. Below it is a large heading "AWS Application Composer" in white. Underneath the heading is the tagline "Visually design and build serverless applications quickly". There is a "Learn more >" link in purple. The main visual is a diagram showing a cloud icon connected to a laptop icon via a network of nodes and lines.

6. Click this password field.



The image shows two side-by-side screenshots. On the left is the "Root user sign in" page. It features the AWS logo at the top, followed by the heading "Root user sign in". Below the heading is an "Email" field containing "srikanth@kmit.in". To the right of the email field is a "Forgot password?" link. Below the email field is a "Password" field, which is highlighted with a yellow circle. A "Sign in" button is located below the password field. At the bottom of the sign-in form are links for "Sign in to a different account" and "Create a new AWS account". On the right is a screenshot of a webpage titled "Free SMB Client". The visible text on the page includes "Find out how high g" and "and retain top tech". There is a "ACCESS GUIDE" button. The background of the page features a graphic of a server rack with red and white squares.

## Root user sign in

Email: srikanth@kmit.in

Password

[Forgot password?](#)

\*\*\*\*\*

[Sign in](#)



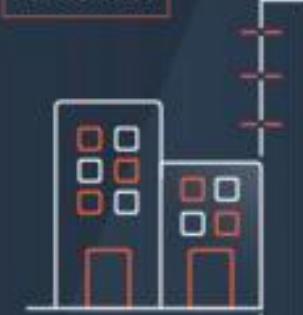
[Sign in to a different account](#)

[Create a new AWS account](#)

## Free SMB Cloud

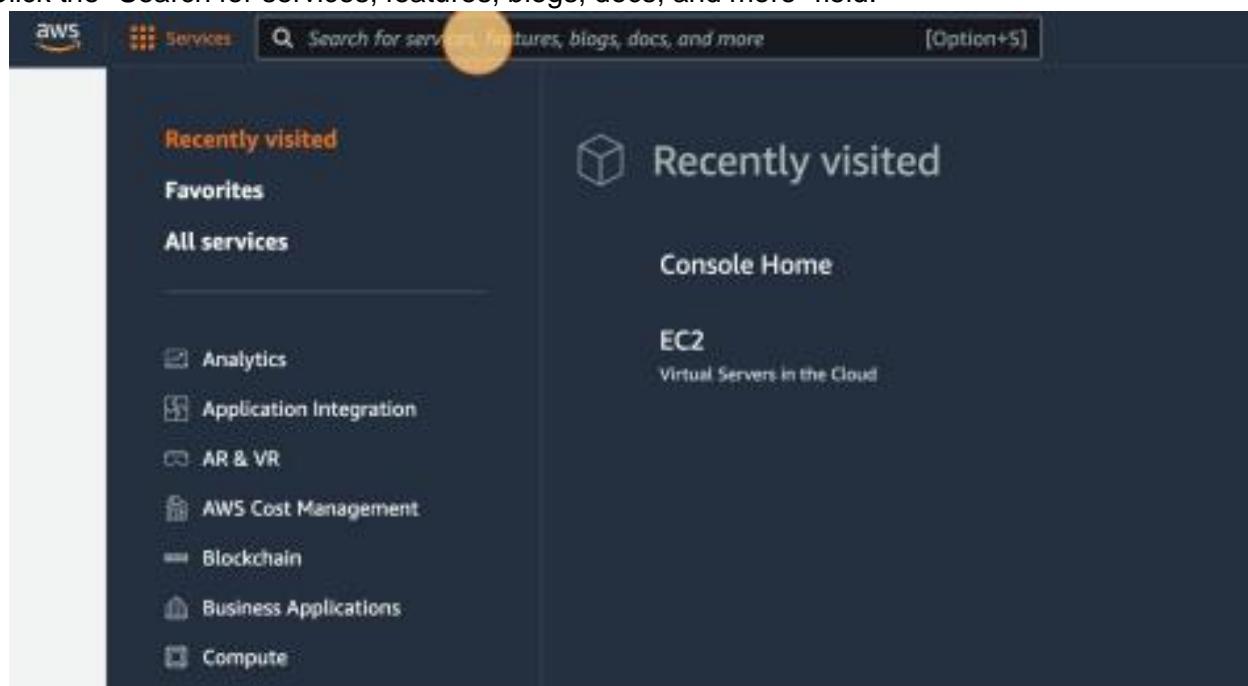
Find out how high growth companies find and retain top technical talent.

[ACCESS GUIDE](#)



7.Click "Services"

8. Click the "Search for services, features, blogs, docs, and more" field.

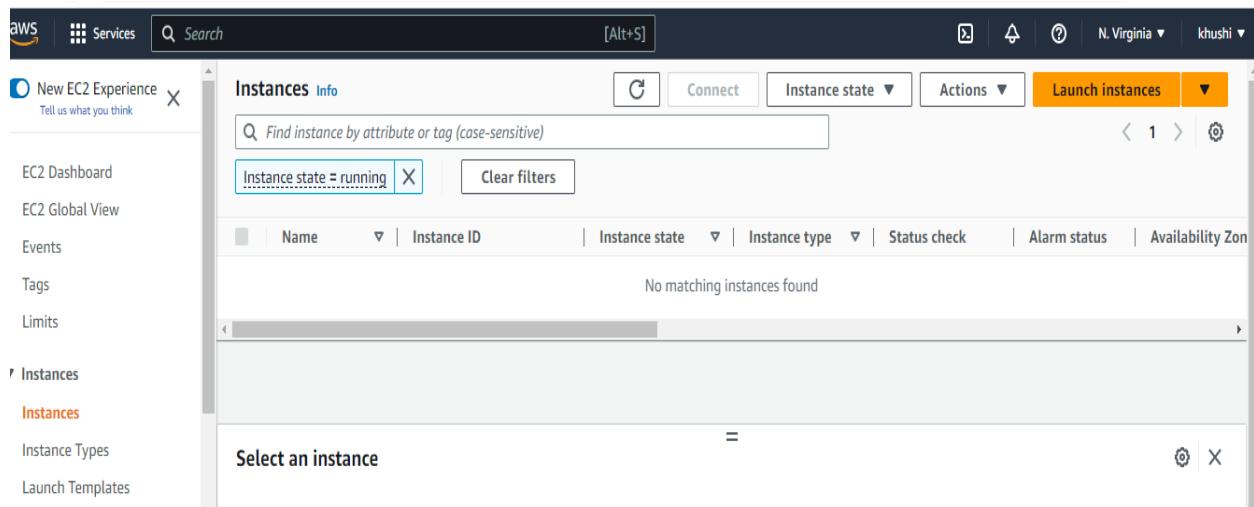


9. Type "EC2 [[enter]]"

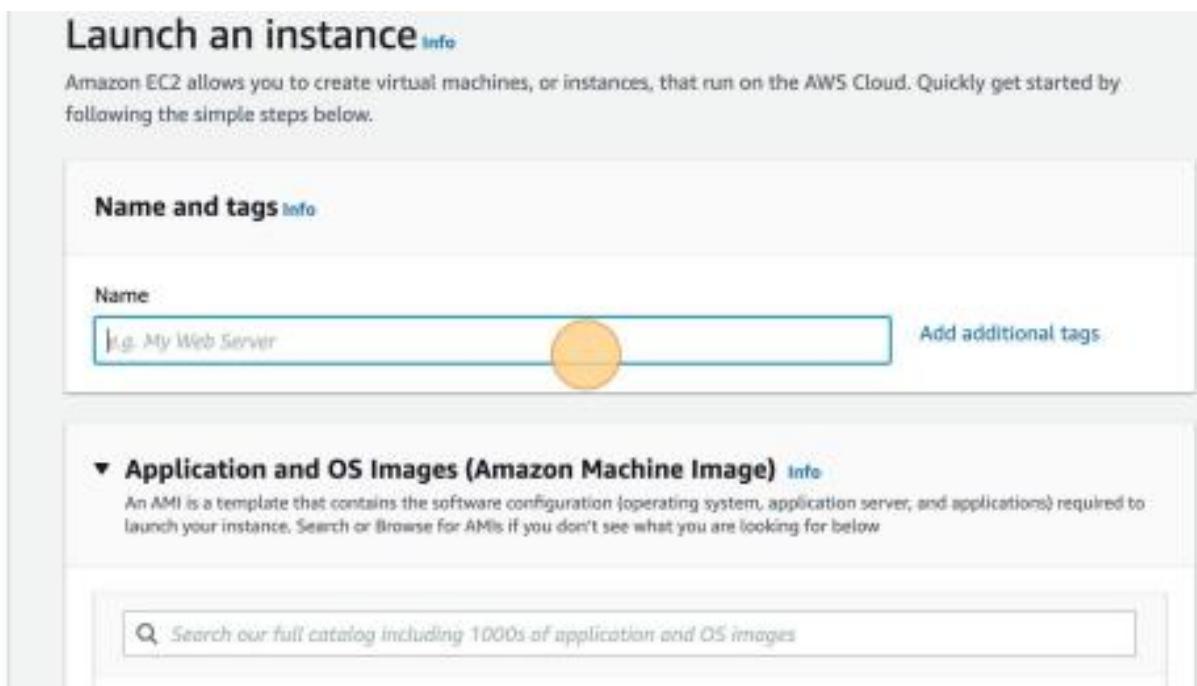
10. Click "Instances"

A screenshot of the AWS EC2 Dashboard. On the left, there's a sidebar with links like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (which is expanded and has 'Instances [new]' highlighted with a yellow circle), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances [new]', 'Dedicated Hosts', and 'Capacity Reservations'. The main content area is titled 'Resources' and shows a summary of EC2 resources in the US East (Ohio) Region. It includes tables for 'Instances (running)', 'Dedicated Hosts', 'Elastic IPs', 'Instances', 'Key pairs', 'Load balanc', 'Placement groups', 'Security groups', and 'Volumes'. Below this, there's a callout box with a tip about launching Microsoft SQL Server Always On availability groups. At the bottom, there are sections for 'Launch instance' (with a note about getting started) and 'Service health' (with a link to 'AWS Health D...').

11. Click "Launch instances"



12. Click the "Name" field.



13. give a name under Name and tags field and press enter

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name  [Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

14. under application and os images field select amazon aws

Recents Quick Start

Search our full catalog including 1000s of application and OS images

Amazon Linux Ubuntu Windows Red Hat SUSE Linux >

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type [Free tier eligible](#)

ami-02d1e544b84bf7502 (64-bit (x86)) / ami-03e57de632660544c (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Boot device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0 20220606.1 vRF G4 HVM en2

15. Click "Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type"

The screenshot shows the AWS Quick Start interface. At the top, there are tabs for 'Recents' and 'Quick Start'. Below the tabs, there is a grid of icons representing different AMIs: Amazon Linux, Ubuntu, Windows, Red Hat, and SUSE Linux. A search bar is located on the right side. Below the grid, the text 'Amazon Machine Image (AMI)' is displayed. A specific AMI is highlighted with a yellow oval: 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type'. This item includes details: 'ami-02d1e544b84bf7502 (64-bit (x86)) / ami-03e57de632660544c (64-bit (Arm))', 'Virtualization: hvm', 'ENA enabled: true', and 'Root device type: ebs'. To the right of this box is a 'Free tier eligible' dropdown menu. The bottom section contains fields for 'Description' (Amazon Linux 2 Kernel 5.10 AMI 2.0.20220606.1 x86\_64 HVM gp2), 'Architecture' (set to '64-bit (x86)'), and 'AMI ID' (ami-02d1e544b84bf7502). A blue circular icon with a white number '2' is visible on the left.

16. Select 64-bit architecture.

The screenshot shows the AWS Quick Start interface with the same AMI selected as in the previous step. The 'Architecture' field now has a dropdown menu open, showing three options: '64-bit (x86)', '64-bit (x86)' (which is highlighted with a yellow oval), and '64-bit (Arm)'. The other fields remain the same: 'Description' (Amazon Linux 2 Kernel 5.10 AMI 2.0.20220606.1 x86\_64 HVM gp2), 'AMI ID' (ami-02d1e544b84bf7502), and the 'Free tier eligible' dropdown. A blue circular icon with a white number '2' is visible on the left.

17. select instance type t2.micro free tier eligible

The screenshot shows the 'Instance type' section of the AWS instance creation wizard. The 't2.micro' instance type is selected, and it is labeled as 'Free tier eligible'. Other details shown include family (t2), 1 vCPU, 1 GiB Memory, and various On-Demand and Sustained Use Pricing options. A 'Compare instance types' link is also present.

18. create a key pair login or you can use an already existing key pair also.

The screenshot shows the 'Key pair (login)' section. It includes a note about using a key pair for secure connection. A dropdown menu labeled 'Select' is open, and a 'Create new key pair' button is visible.

The screenshot shows the 'Key pair (login)' section again. This time, a key pair named 'abc1' has been selected from the dropdown menu. The 'Create new key pair' button is also visible.

19. Click the "Select existing security group" field.

The screenshot shows the 'Network settings' section. It includes fields for Network (vpc-0c21e57c183e43fb6), Subnet (No preference), Auto-assign public IP (Enable), and Firewall (security groups). Two buttons at the bottom are 'Create security group' (selected) and 'Select existing security group'.

We'll create a new security group called 'launch-wizard-4' with the following rules:

## 20. select existing security group

**Firewall (security groups) [Info](#)**  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

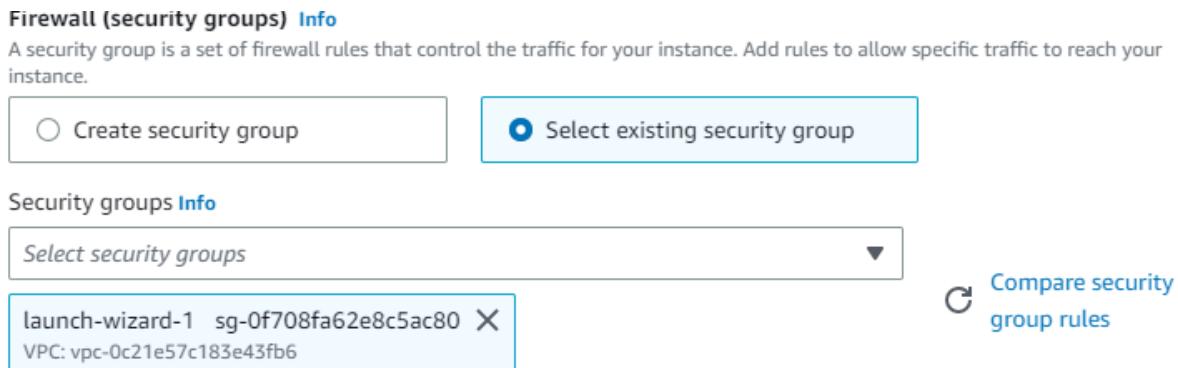
Create security group       Select existing security group

**Security groups [Info](#)**

Select security groups ▾

launch-wizard-1 sg-0f708fa62e8c5ac80 X  
VPC: vpc-0c21e57c183e43fb6

 [Compare security group rules](#)



## 21. under configure storage by default we have 8GiB change it to 10GiB and gp2

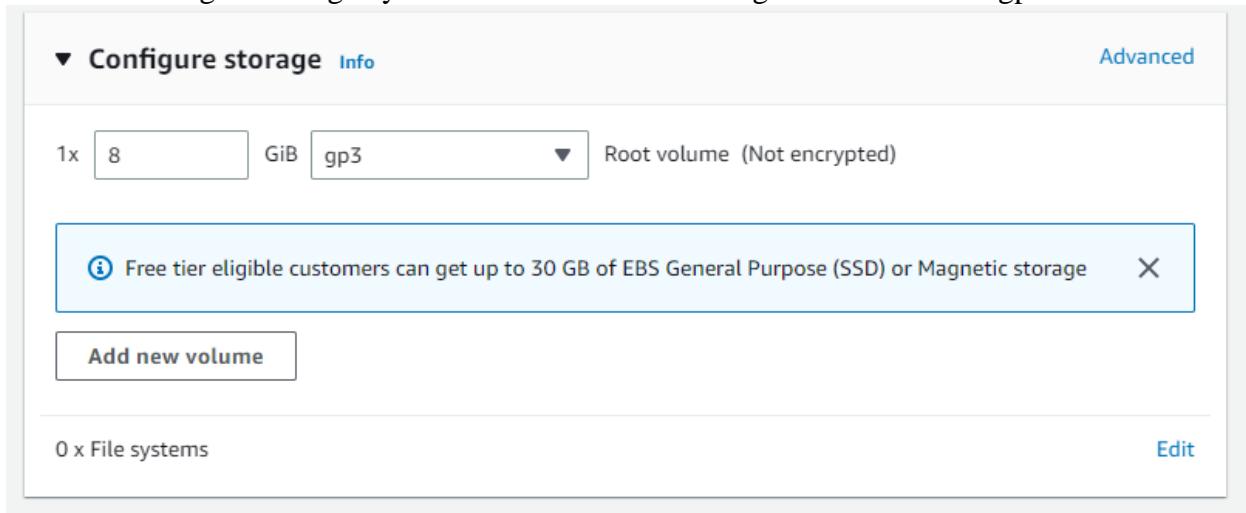
**▼ Configure storage [Info](#)** Advanced

1x  GiB  Root volume (Not encrypted)

 Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

[Add new volume](#)

0 x File systems [Edit](#)



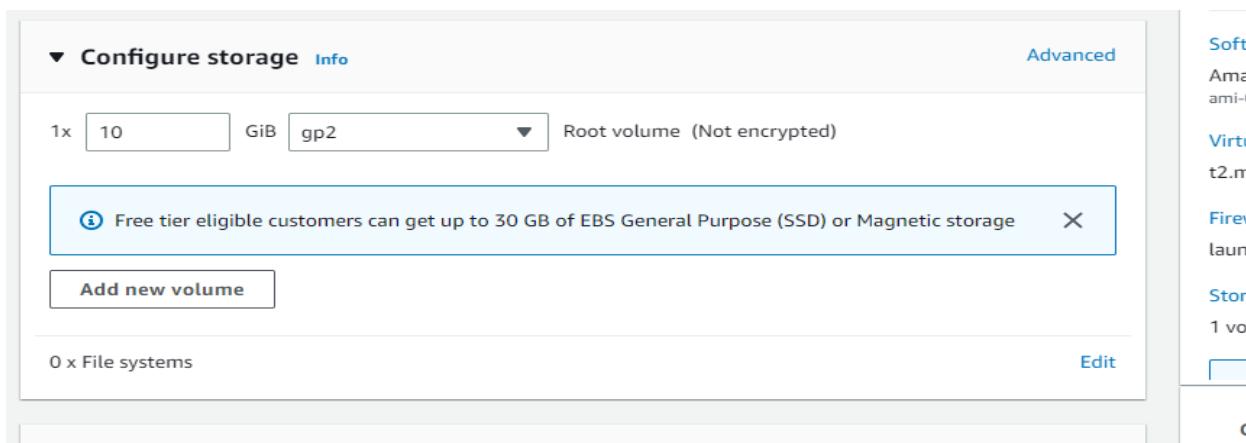
**▼ Configure storage [Info](#)** Advanced

1x  GiB  Root volume (Not encrypted)

 Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

[Add new volume](#)

0 x File systems [Edit](#)



Softw  
Ama:  
ami-0  
  
Virtu  
t2.mi  
  
Firew  
launc  
  
Store  
1 vol  
  
C

22. click on advanced details info and leave it as it is.

The screenshot shows the 'Advanced details' section of a Lambda function configuration. It includes a summary table with the following data:

Setting	Value
Number of instances	1
Software Image (AMI)	Amazon Linux 2023 AMI 2023.0.2... <a href="#">read more</a> ami-06e46074ae430fba6
Virtual server type (instance type)	t2.micro
Firewall (security group)	launch-wizard-1
Storage (volumes)	1 volume(s) - 10 GiB

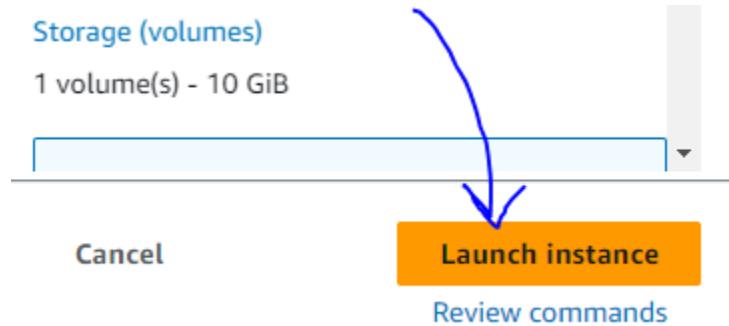
23. Click the "Number of instances Info" field.

The screenshot shows the 'Number of instances' field selected for editing. The configuration table remains the same as in the previous step:

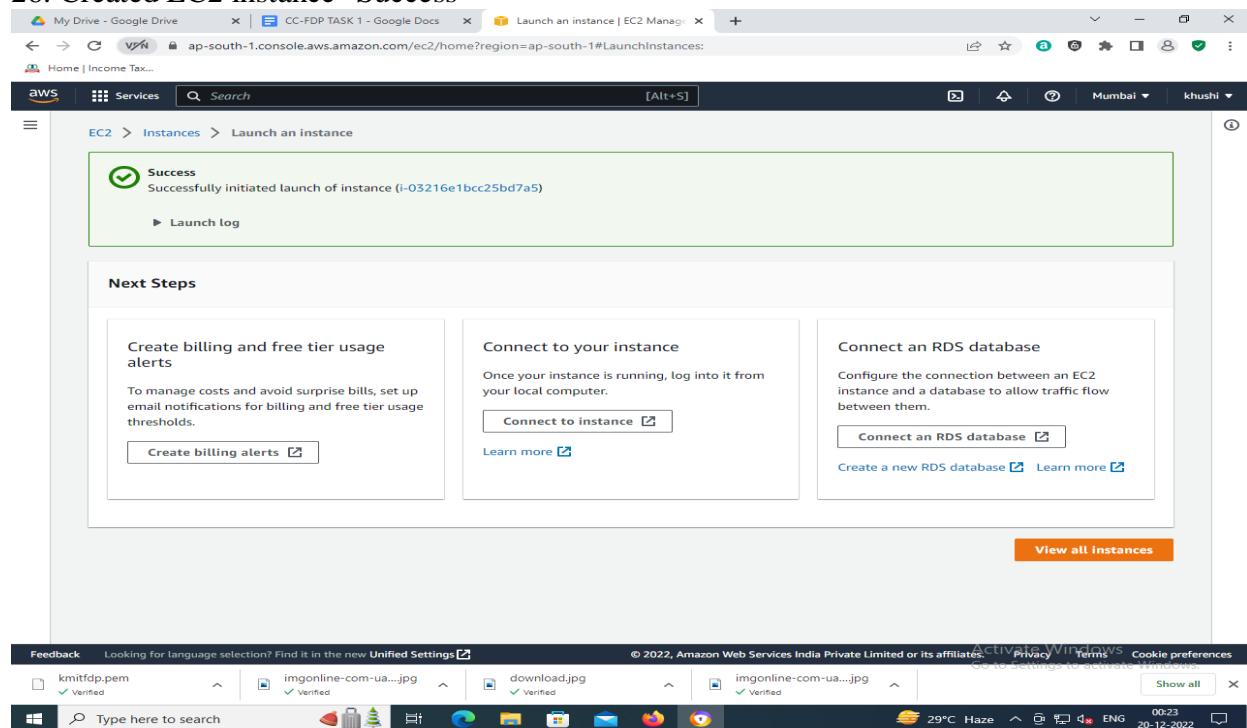
Setting	Value
Number of instances	1
Software Image (AMI)	Amazon Linux 2023 AMI 2023.0.2... <a href="#">read more</a> ami-06e46074ae430fba6
Virtual server type (instance type)	t2.micro
Firewall (security group)	launch-wizard-1
Storage (volumes)	1 volume(s) - 10 GiB

At the bottom of the screen, there are three buttons: 'Cancel', 'Launch instance' (which is highlighted in orange), and 'Review commands'.

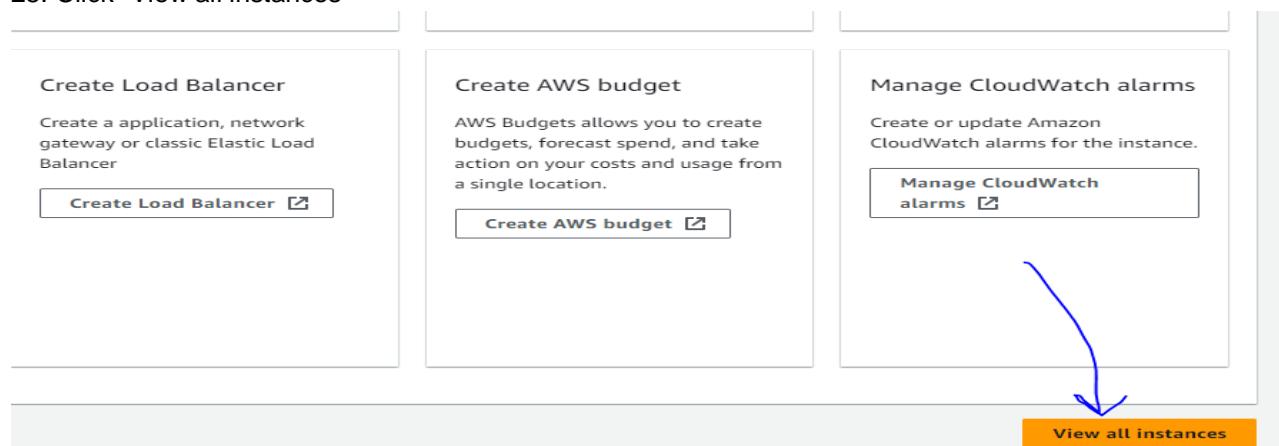
24. click on launch instance



26. Created EC2 instance "Success"



25. Click "View all instances"



26. Right Click on the instance name.

A screenshot of the AWS EC2 Instances page. At the top, there's a search bar labeled "Find instance by attribute or tag (case-sensitive)". Below it is a table header with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. A single instance is listed: "myec2" (Instance ID: i-0f6cbc8b268b72eb9, Instance state: Running, Instance type: t2.micro, Status check: Initializing, Alarm status: No alarms, Availability Zone: us-east-1d). A blue arrow points from the text "Right Click on the instance name." to the "myec2" instance name.

27. Click "Connect"

A screenshot of the AWS EC2 Instances page for the instance "myec2". The instance details are shown: Name (myec2), Instance ID (i-0f6cbc8b268b72eb9), Instance state (Running), and Instance type (t2.micro). A context menu is open over the instance name, listing options: Launch instances, Launch instance from template, Migrate a server, Connect (highlighted with a yellow box and a blue arrow), Stop instance, Start instance, Reboot instance, Hibernate instance, Terminate instance, Instance settings, Networking, Security, Image and templates, and Monitor and troubleshoot. To the right of the instance details, there are tabs for Storage, Status checks, and Monitoring. Under the Storage tab, it shows a Public IPv4 address (3.83.100.145) and an open address. Under the Status checks tab, it shows the Instance state as Running.

28. Click "EC2 Instance Connect"

A screenshot of the "Connect to instance" page for the instance "myec2". The URL is EC2 > Instances > i-0f6cbc8b268b72eb9 > Connect to instance. The "EC2 Instance Connect" tab is selected. It shows the instance ID (i-0f6cbc8b268b72eb9 (myec2)) and the Public IP address (3.83.100.145). Under "User name", it says "Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user." A text input field contains "ec2-user". A note at the bottom states: "Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name."

## 29. Click "Connect"

The screenshot shows the EC2 Instance Connect interface. At the top, there are four tabs: EC2 Instance Connect (selected), Session Manager, SSH client, and EC2 serial console. Below the tabs, the instance ID is listed as **i-0f6cbc8b268b72eb9 (myec2)**. The public IP address is listed as **3.83.100.145**. The user name is set to **ec2-user**. A note in a blue-bordered box states: **Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.**. At the bottom right, there are **Cancel** and **Connect** buttons, with a blue arrow pointing to the **Connect** button.

## 30. Instance will open in browser.

The screenshot shows a web browser window with multiple tabs. The active tab is titled "EC2 Instance Connect" and shows the URL <ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-03216e1bcc25bd7a5&osUs...>. The browser interface includes a search bar, a services menu, and a user profile for "khushi". The main content area displays a black screen with the text "Establishing Connection ..." and a progress bar. Below the content, the instance ID is listed as **i-03216e1bcc25bd7a5 (test)**, with Public IPs: 15.206.117.76 and Private IPs: 172.31.34.82. The bottom of the screen shows the Windows taskbar with various pinned icons and system status information.

**31. You can now start interacting with the instance.**

```
last login: Thu Jun 16 06:46:41 2022 from ec2-3-16-146-2.us-east-2.compute.amazonaws.com
[ec2-user@ip-172-31-31-32 ~]$ ls
Amazon Linux 2 AMI
[ec2-user@ip-172-31-31-32 ~]$ cd /home/ec2-user
[ec2-user@ip-172-31-31-32 ~]$ ls
[ec2-user@ip-172-31-31-32 ~]$
```

## EXPERIMENT 3

### Create an EC2 instance and access docker image to host a simple webpage.

**AIM:** To create an EC2 instance and access docker image to host a simple webpage

**REQUIREMENTS:** AWS active account and desktop.

#### **PROCEDURE:**

1. Create an Account in docker hub cloud repository.

<https://hub.docker.com/signup>

The screenshot shows the Docker Hub sign-up page with a blue header and sidebar. The main title is "Get Started Today for Free". Below it, there's a link "Already have an account? [Sign In](#)". The form fields are as follows:

- Username: khushi
- Email: khushima2096@gmail.com
- Password: (redacted)

Below the form are two checkboxes:

- Send me occasional product updates and announcements.
- I agree to the [Subscription Service Agreement](#), [Privacy Policy](#), and [Data Processing Terms](#).

At the bottom, there's a reCAPTCHA field with the text "I'm not a robot" and a checkbox. Next to it is a "reCAPTCHA" logo with "Privacy + Terms" below it. A handwritten mark resembling a checkmark is drawn over the "Activate Windows" button. The "Sign Up" button is at the bottom.

## 2. select images you want to pull

The screenshot shows the Docker Hub search interface. The search bar at the top contains the query 'images'. Below the search bar, there are filters on the left, including 'Products' (with 'Images' checked) and 'Trusted Content' (with 'Docker Official Image' checked). The search results show two items:

- alpine**: DOCKER OFFICIAL IMAGE · 1B+ · 9.9K Pulls. Updated 20 days ago. Tags: Linux, riscv64, x86-64, ARM, ARM 64, 386, PowerPC 64 LE, IBM Z.
- nginx**: DOCKER OFFICIAL IMAGE · 1B+ · 10K+ Pulls. Updated 6 days ago. Tags: official, build, nginx.

Each item has a 'Learn more' link and a blue line graph representing pull requests over time.

## 3. Build docker image of the web project and push it to docker-hub cloud repository.

### i) Here we download the fileshare project from Github of srikanth reddy-lectures

The screenshot shows the GitHub repository page for `srikanthreddy-lectures/fileshare`. The repository is public and has 1 branch and 0 tags. The code tab is selected. The repository details on the right include:

- About**: No description, website, or topics provided.
- Code**: Readme, 0 stars, 1 watching, 0 forks.

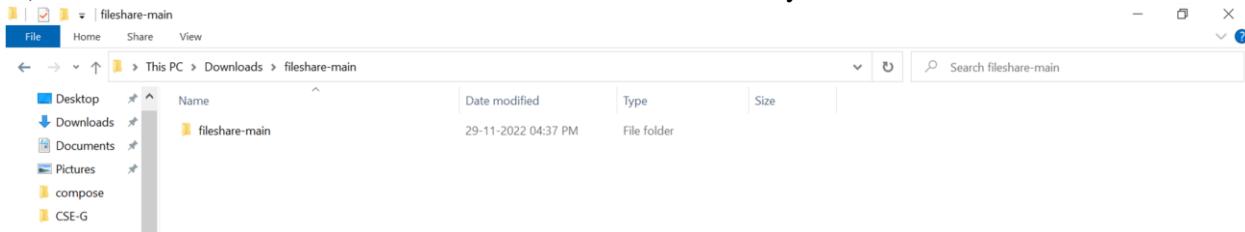
### ii) Here we go for code ->click on it ->go for Download Zip option

The screenshot shows the GitHub repository page for `srikanthreddy-lectures/fileshare`. The repository is public and has 1 branch and 0 tags. The code tab is selected. The repository details on the right include:

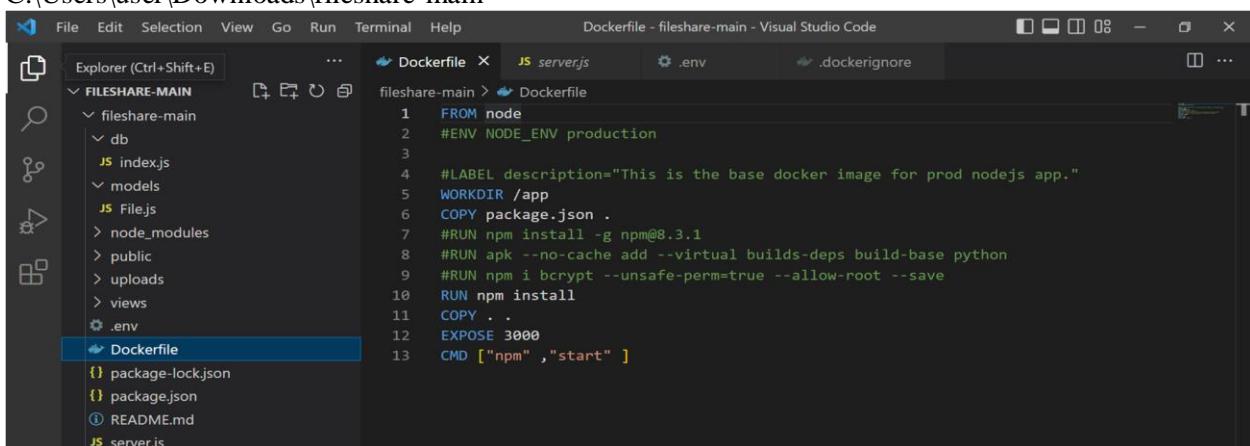
- About**: No description, website, or topics provided.
- Code**: Readme, 0 stars, 1 watching, 0 forks.
- Releases**: No releases published.

A modal window is open over the repository details, showing the 'Code' dropdown menu with the 'Download ZIP' option highlighted.

iii) Here we can see that fileshare-main is there in our standalone system



iv) After downloading once we have to check the Dockerfile in fileshare project and copy the path C:\Users\user\Downloads\fileshare-main



v. Open the powershell and check for docker --version

```
PS C:\Users\user> docker --version
Docker version 20.10.20, build 9fdeb9c
PS C:\Users\user> cd C:\Users\user\Downloads\fileshare-main
PS C:\Users\user\Downloads\fileshare-main>
```

Vi. Change the path and build the fileshare project

```

PS C:\Users\user\Downloads\fileshare-main\fileshare-main> docker build -t fileshare .
[+] Building 9.5s (4/10)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 389B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/node:latest
=> [auth] library/node:pull token for registry-1.docker.io
=> [1/5] FROM docker.io/library/node@sha256:50b76fc6dc5f03cb3d14c71b8564948aed2bc5124325f35830b2f3be21 5.0s
=> => resolve docker.io/library/node@sha256:50b76fc6dc5f03cb3d14c71b8564948aed2bc5124325f35830b2f3be21 0.1s
=> => sha256:9bf5846b28f63acab0ccb0a39a245fbca414e6b7ecd467282f58016537c06e159 2.21kB / 2.21kB 0.0s
=> => sha256:eaed4885733a5359619ad81e5d728eaead930972e8d8d14dd8e95d75fe895 7.51kB / 7.51kB 0.0s
=> => sha256:5c8cfbf51e6e6869f1af2a1e7067b07fd6733036a333c9d29f743b0285e26037 2.10MB / 5.16MB 4.9s
=> => sha256:50b76fc6dc5f03cb3d14c71b8564948aed2bc5124325f35830b2f3be21950af1 1.21kB / 1.21kB 0.0s
=> => sha256:aa3a609d15798d35c1484072876b7d22a316e98de6a097de33b9dade6a689cd1 8.39MB / 10.88MB 4.9s
=> => sha256:f2f58072e9ed1aa1b0143341c5ee83815c00ce47548309fa240155067ab0e698 2.10MB / 55.04MB 4.9s
=> [internal] load build context
=> => transferring context: 2.79MB 5.0s
=> => transferring context: 2.79MB 4.8s

```

vi. Now the check images ,will get the image of fileshare

```

PS C:\Users\user\Downloads\fileshare-main\fileshare-main> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
Fileshare           latest   c962e4ee9e49  2 minutes ago  1.07GB
archanareddyccse/ubuntugit  latest   facb5c4f563c  34 hours ago   193MB
Ubuntu-Git          latest   facb5c4f563c  34 hours ago   193MB
test22             latest   f0b78d71d2cd  5 days ago    206MB

```

vii. Now run the image with port number 3030

```

PS C:\Users\user\Downloads\fileshare-main\fileshare-main> docker run --name filesharing -d -p 3030:3000 fileshare
6bc612ce02a4e70aa6af12c2b5cffd0fd655d61bf9d5a853ed47c669e1f9f3
PS C:\Users\user\Downloads\fileshare-main\fileshare-main>

```

viii. Now we can see the container by name fileshare of port number 3030

```

PS C:\Users\user\Downloads\fileshare-main\fileshare-main> docker container ps -a
CONTAINER ID   IMAGE      COMMAND      CREATED        STATUS          PORTS
NAMES
6bc612ce02a4   fileshare   "docker-entrypoint.s..."  54 seconds ago   Up 51 seconds   0.0.0.0:3030->3000/tcp
filesharing
PS C:\Users\user\Downloads\fileshare-main\fileshare-main>

```

ix. Now tag the image with the <docker hub id>/<imagename> and push into dockerhub

Here iam using my dockerhub id archanareddyccse/fileshare1

```

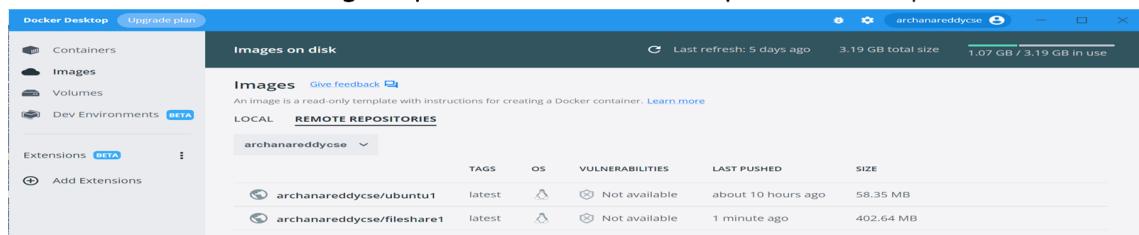
PS C:\Users\user\Downloads\fileshare-main\fileshare-main> docker tag fileshare archanareddyccse/fileshare1
PS C:\Users\user\Downloads\fileshare-main\fileshare-main> docker push archanareddyccse/fileshare1

```

x. Now push into hub by archanareddyccse/fileshare1

```
PS C:\Users\user\Downloads\fileshare-main\fileshare-main> docker push archanareddyccse/fileshare1
Using default tag: latest
The push refers to repository [docker.io/archanareddyccse/fileshare1]
f467781874d7: Pushed
b96c3920cd41: Pushed
1e0a2a2a444d: Pushed
afdadabceebf: Pushed
bd64d88b305d: Mounted from library/node
a5ba4dd1f9d2f: Mounted from library/node
d29a2b8b4664: Mounted from library/node
5461a69cb3a7: Mounted from library/node
4ff8848a7748: Mounted from library/node
b772748906: Mounted from library/node
d247c0fb37b: Mounted from library/node
cfdd5c3bd77e: Mounted from library/node
870a241bfbed: Mounted from library/node
latest: digest: sha256:19aeeeaeefdf9a20d4377ddb4b92c4ffa524c4bf6b93db18f279532c591790fa7 size: 3053
PS C:\Users\user\Downloads\fileshare-main\fileshare-main>
```

xi. Here we see the image is present the remote repositories of your docker desktop



xii. We go for hub.docker.com, here we can see the image with pull option



xiii. Now open the browser with localhost

4) Connect to AWS-EC2 instance using OpenSSH for powershell/terminal using keypair file.

```
PS C:\Users\user\Downloads> ssh -i "kmitfdp.pem" ubuntu@ec2-3-21-165-152.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-21-165-152.us-east-2.compute.amazonaws.com (3.21.165.152)' can't be established
.
ECDSA key fingerprint is SHA256:E0J77mipGHGg3yk0/jzNPWo4+huSwH292PzI64lM920.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-21-165-152.us-east-2.compute.amazonaws.com,3.21.165.152' (ECDSA) to the list
of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)

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

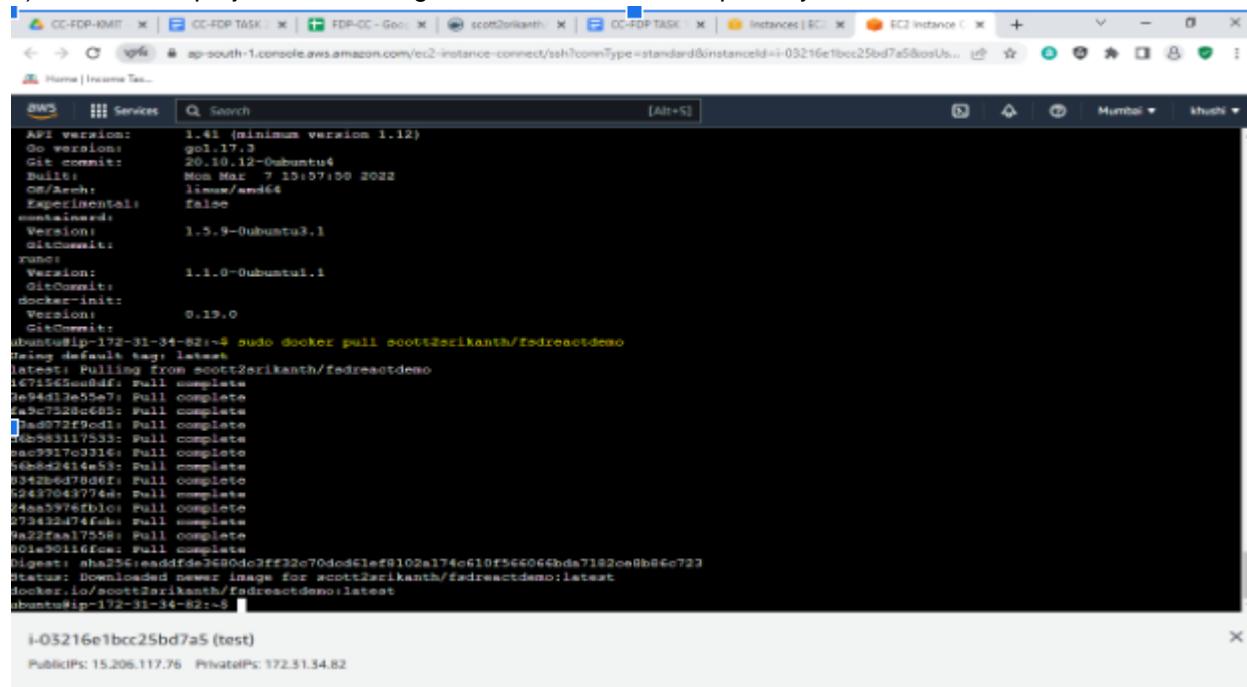

```

5) Install docker using linux commands.

```
ubuntu@ip-172-31-38-84:~$ sudo su -
root@ip-172-31-38-84:~# apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [756 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [169 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [11.4 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [485 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [74.3 kB]

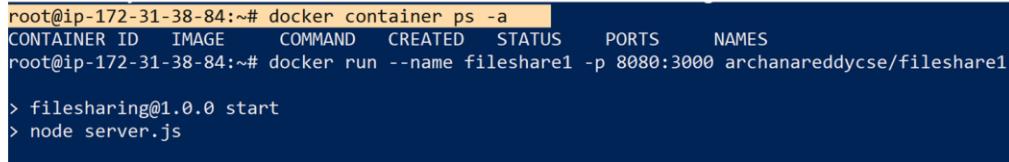
root@ip-172-31-38-84:~# apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 12 not upgraded.
Need to get 65.8 MB of archives.
```

6) Pull the web project docker image from the docker-hub repository.



```
ubuntu@ip-172-31-34-82:~$ sudo docker pull scott2srikanth/fedreactdemo
Using default tag: latest
latest: Pulling from scott2srikanth/fedreactdemo
1c715e5ea0df: Pull complete
2e94d13e55e7: Pull complete
fa9c7328c685: Pull complete
5add072f9cod1: Pull complete
1cc583117533: Pull complete
pac9317c3316: Pull complete
b6bd0d414e53: Pull complete
0342bd4d79d6f: Pull complete
0433043774a: Pull complete
74nn5976fb1c: Pull complete
073432474fcab: Pull complete
9a22ffaa17558: Pull complete
001e90116fcce: Pull complete
Digest: sha256:eadddfd3290dc3ff32c70dcd61ef9102a374c610f566066bda7102ce0b86c723
Status: Downloaded newer image for scott2srikanth/fedreactdemo:latest
docker.io/scott2srikanth/fedreactdemo:latest
ubuntu@ip-172-31-34-82:~$
```

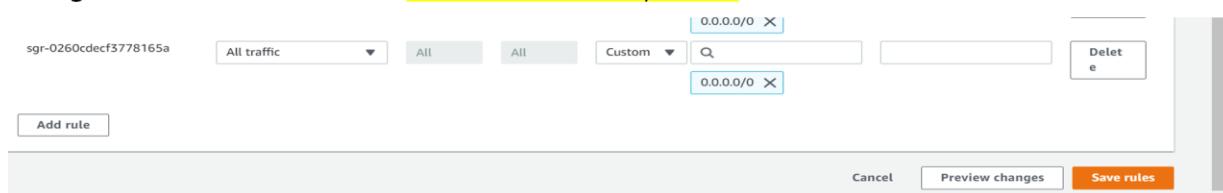
7) Run the container with specifying the port(e.g 8080), so that user can access.



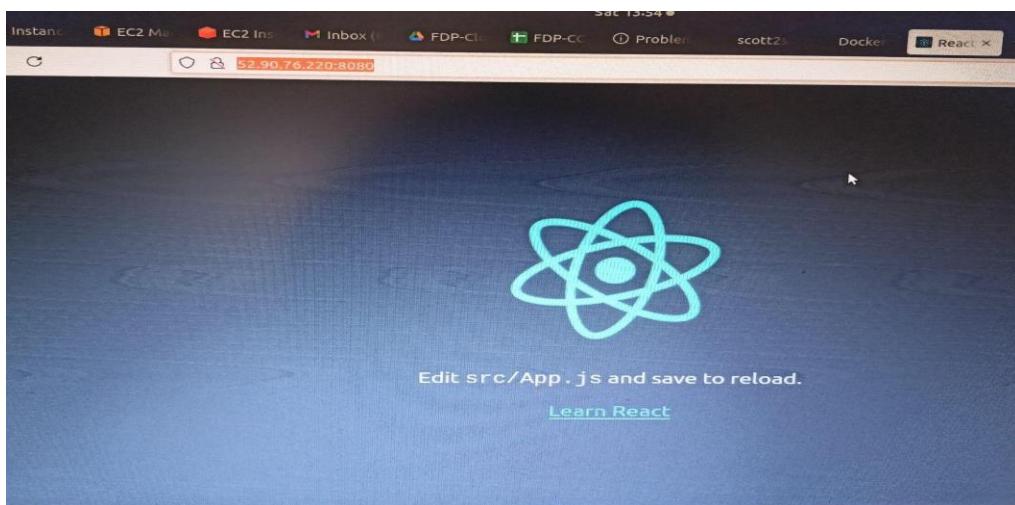
```
root@ip-172-31-38-84:~# docker container ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS      NAMES
root@ip-172-31-38-84:~# docker run --name fileshare1 -p 8080:3000 archanareddycse/fileshare1
> filesharing@1.0.0 start
> node server.js
```

8) Configure the inbound rule in AWS-EC2 instance to allow any(TCP) request from computer to access the port 8080 from the browser.

Change the inbound rule from **All traffic and Anywhere**



9) Access the home webpage of web application with public IP and port-no of AWS EC2 instance.



## EXPERIMENT 4

### Deploy a Machine learning model using AWS EC2

**AIM:** To Deploy a Machine learning model using AWS EC2

**REQUIREMENTS:** AWS active account and desktop.

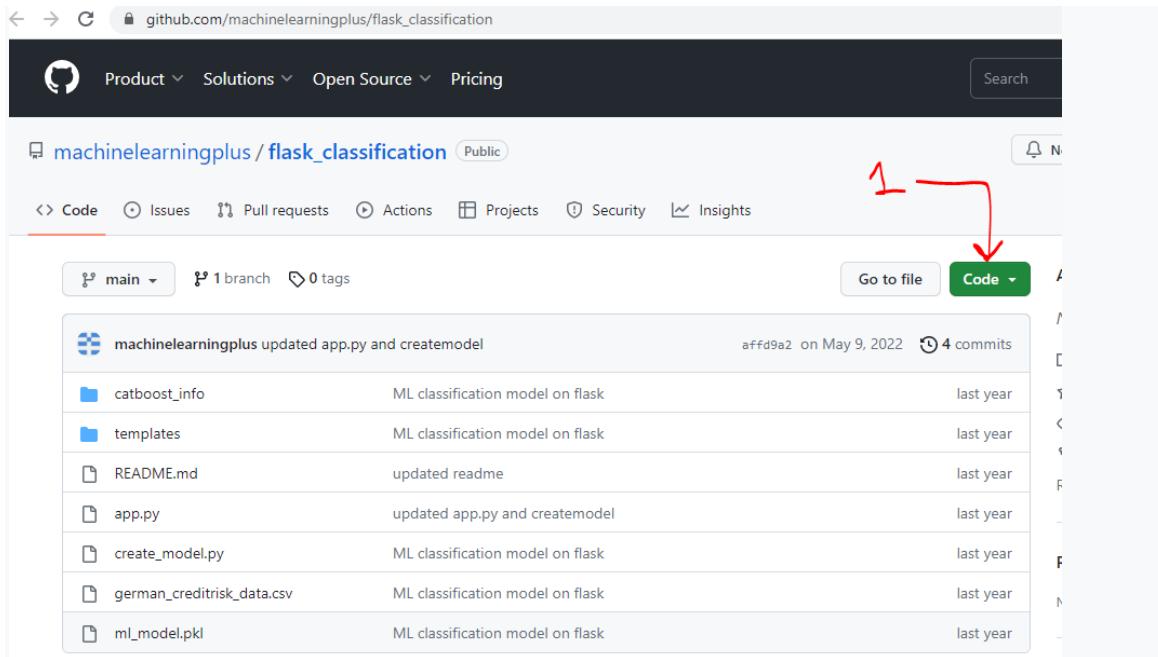
#### **PROCEDURE:**

##### Pre Requisites

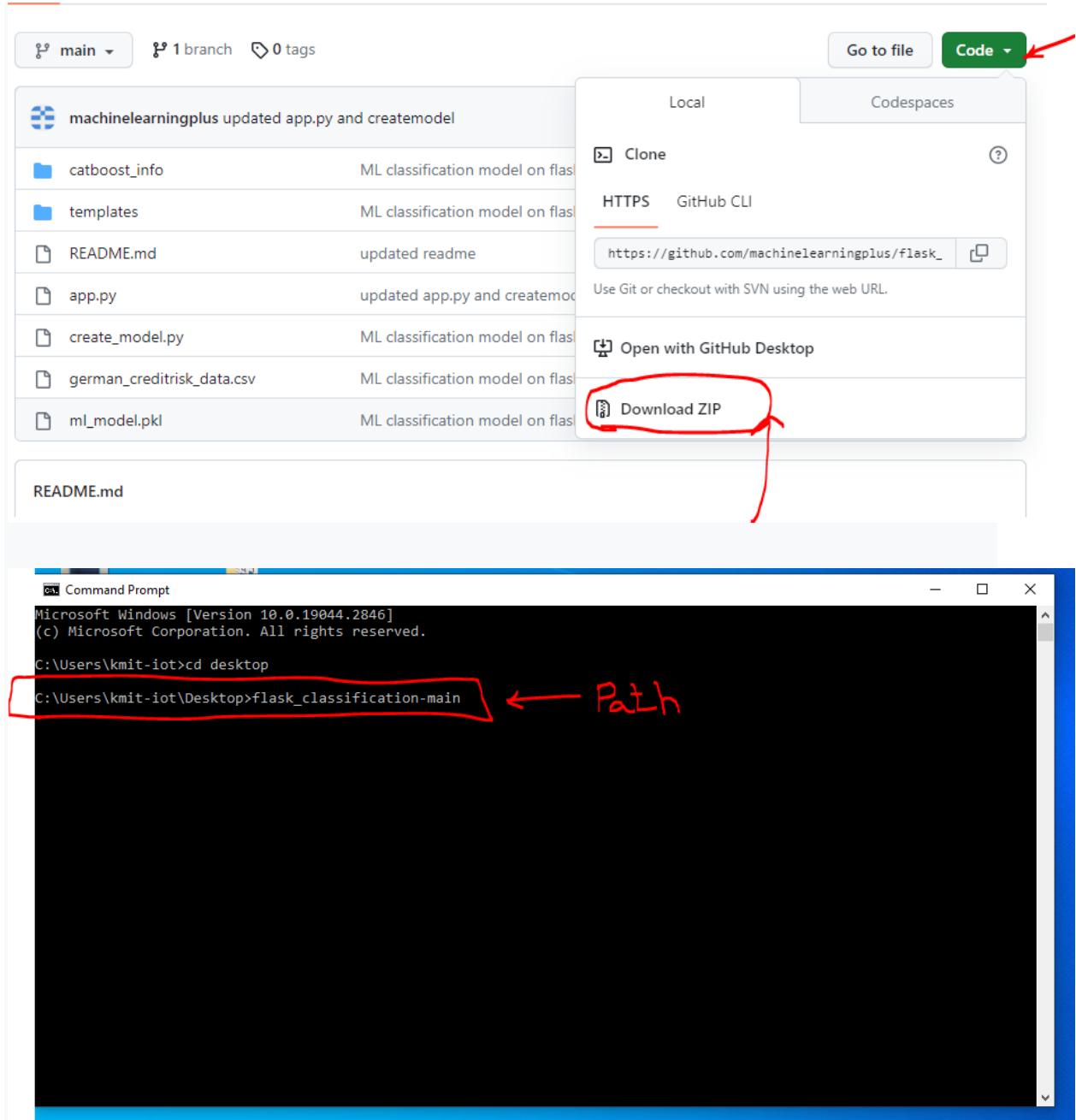
1. A working ML model built as a REST API, preferably with flask or similar framework.  
You can download the code for the model app here.
2. [https://github.com/machinelearningplus/flask\\_classification](https://github.com/machinelearningplus/flask_classification)
3. AWS Account. If you don't have one, you can create it here. You will need a credit card to create one, however, to deploy your model, we are going to be using a free tier instance, which will not incur cost.
4. You will need Python with an IDE installed to build, debug and serve your app locally. Recommend anaconda for this.  
Alternates: VS Code, PyCharm or Spyder.

#### Step 1: Run the ML flask app in your local computer

1. Download the code directory zip file and extract contents.



2. Open terminal or command prompt and change directory to the code folder: `cd downloaded\_code\_directory`

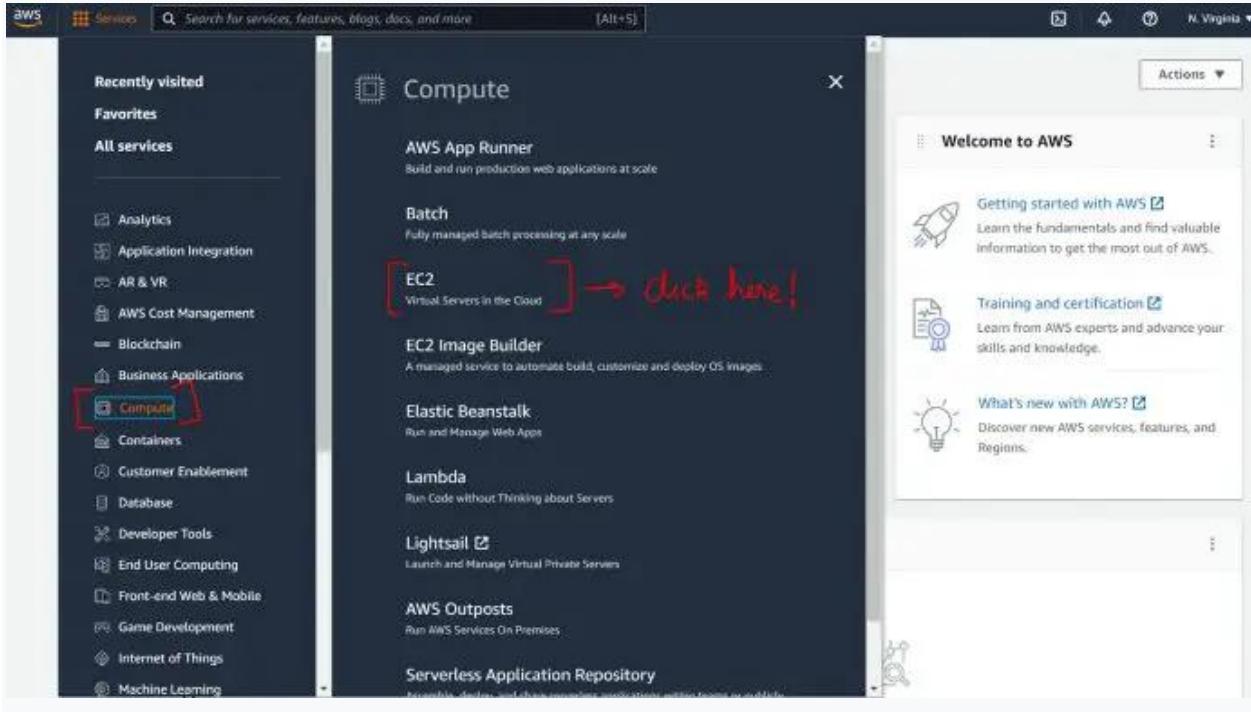


3. Run `python app.py`

## 6. Launch an ec2 instance on AWS free tier account

1. Login to your AWS account from console.aws.amazon.com.

Then, Search ‘EC2’ in the search box in the top. Or you can find it in the list of services under ‘Compute’



## 2. Launch an EC2 Instance

To launch an EC2 instance, you will have to go through a sequence of steps. An EC2 instance is nothing but a remote computer that will run at an Amazon Data Center that we can lease to host our ML app.

Let's go over the steps one by one.

- (i) Click the **launch instance** button from the EC2 dashboard.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (selected), 'AMIs', and 'Elastic Block Store'. The main area is titled 'Resources' and shows a summary of Amazon EC2 resources in the US East (N. Virginia) Region. It includes tables for Instances (running), Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. A callout box points to the 'Launch instance' button in the 'Launch instance' section, which is highlighted with a red box and has a handwritten note 'Click to launch new instance' written next to it. The 'Service health' section shows the region as US East (N. Virginia) and the status as 'This service is operating normally'.

this will get you started with creating a new EC2 instance.

This screenshot is nearly identical to the previous one, showing the AWS EC2 Dashboard. The sidebar and resource summary are the same. The 'Launch instance' section is visible, but the red annotation and handwritten note 'Click to launch new instance' have moved to point at the 'Migrate a server' button instead of the 'Launch instance' button.

## ii) Choose an AMI Image that is eligible for Free Tier

The following screen will list the available EC2 instances, also called **Amazon Machine Image (AMI)**. Pick one that is '**Free Tier Eligible**'. Other instance types will accrue cost.

We are replacing this launch experience with a new launch experience, which we will continue to improve based on your feedback. Opt-in to the new experience by selecting the button on the right and give us feedback. For now you can still opt out once you have tried it.

Opt-in to the new experience

1. Choose AMI   2. Choose instance type   3. Configure instance   4. Add storage   5. Add tags   6. Configure Security Group   7. Review

Step 1: Choose an Amazon Machine Image (AMI)

| AMI Name                                              | Description                                                             | Root Device Type                                                                                                                                                                                                                                                   | Virtualization Type   | ENI Enabled              | Action           |               |
|-------------------------------------------------------|-------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|--------------------------|------------------|---------------|
| <b>Ubuntu Server 22.04 LTS (HVM), SSD Volume Type</b> | ami-09d56f9956ab235b (64-bit x86) / ami-02ddaf75d21f25213 (64-bit Arm)  | Ubuntu Server 22.04 LTS (HVM) (SSD General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cisad/services">http://www.ubuntu.com/cisad/services</a> ))                                                                | Root device type: ebs | Virtualization type: hvm | ENI Enabled: Yes | <b>Select</b> |
| Ubuntu Server 20.04 LTS (HVM), SSD Volume Type        | ami-0c4f7029847b90238 (64-bit x86) / ami-0d70a59d7191a8079 (64-bit Arm) | Ubuntu Server 20.04 LTS (HVM)(SSD General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ))                                                                 | Root device type: ebs | Virtualization type: hvm | ENI Enabled: Yes | <b>Select</b> |
| Microsoft Windows Server 2019 Base                    | ami-08ed25c5dd62794ec0                                                  | Microsoft Windows 2019 Datacenter edition: [English]                                                                                                                                                                                                               | Root device type: ebs | Virtualization type: hvm | ENI Enabled: Yes | <b>Select</b> |
| Deep Learning AMI (Ubuntu 18.04) Version 60.1         | ami-07f4dcf20ed5ed5d                                                    | MXNet-1.8, TensorFlow-2.7, PyTorch-1.10, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check <a href="https://www.amazonaws.com/sagemaker">https://www.amazonaws.com/sagemaker</a> | Root device type: ebs | Virtualization type: hvm | ENI Enabled: Yes | <b>Select</b> |
| Deep Learning AMI GPU PyTorch 1.11.0 (Amazon Linux 2) | ami-023d5ea1ee956059c                                                   |                                                                                                                                                                                                                                                                    |                       |                          |                  | <b>Select</b> |

## (iii) Choose the instance type that belong to the selected AMI

In the following screen, it will ask to select the instance type with number of CPUs, RAM, memory limit etc.

For our app, since we are going with the Free Tier, pick the one with '**t2.micro**' in green with one CPU, 1GB memory. Then click '**Review and Launch**' blue button at the bottom.

**Step 2: Choose an Instance Type**

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All Instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (1 vCPU, 1.0 GiB memory, EBS only)

| Family | Type      | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support |
|--------|-----------|-------|--------------|-----------------------|-------------------------|---------------------|--------------|
| t2     | t2.nano   | 1     | 0.5          | EBS only              | -                       | Low to Moderate     | Yes          |
| t2     | t2.micro  | 1     | 1            | EBS only              | -                       | Low to Moderate     | Yes          |
| t2     | t2.small  | 1     | 2            | EBS only              | -                       | Low to Moderate     | Yes          |
| t2     | t2.medium | 2     | 4            | EBS only              | -                       | Low to Moderate     | Yes          |
| t2     | t2.large  | 2     | 8            | EBS only              | -                       | Low to Moderate     | Yes          |
| t2     | t2.xlarge | 8     | 16           | EBS only              | -                       | Moderate            | Yes          |
| t3     | t3.nano   | 2     | 0.5          | EBS only              | Yes                     | Up to 5 Gigabit     | Yes          |
| t3     | t3.micro  | 2     | 1            | EBS only              | Yes                     | Up to 5 Gigabit     | Yes          |
| t3     | t3.small  | 2     | 2            | EBS only              | Yes                     | Up to 5 Gigabit     | Yes          |
| t3     | t3.medium | 2     | 4            | EBS only              | Yes                     | Up to 5 Gigabit     | Yes          |

Cancel Previous Review and Launch Next: Configure Instance Details

#### (iv) Review and Launch

**Step 7: Review Instance Launch**

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**AMI Details**

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type - ami-09d56ff996ab235b3

**Instance Type**

| Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|------|-------|--------------|-----------------------|-------------------------|---------------------|
| t2.micro      | -    | 1     | 1            | EBS only              | -                       | Low to Moderate     |

**Security Groups**

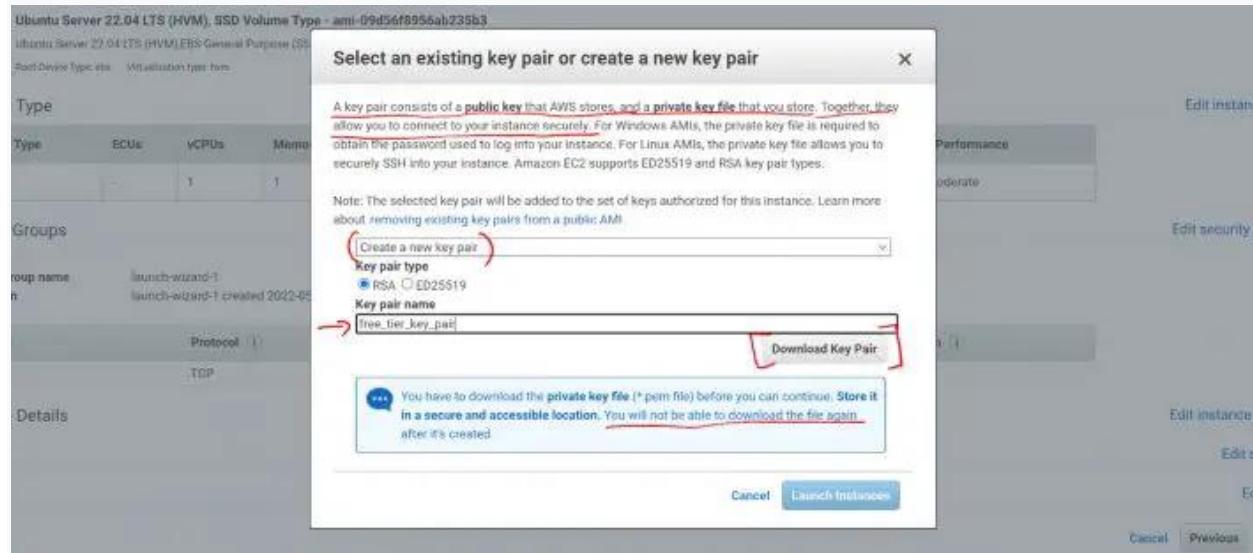
| Type | Protocol | Port Range | Source    | Description     |
|------|----------|------------|-----------|-----------------|
| SSH  | TCP      | 22         | 0.0.0.0/0 | launch-wizard-1 |

Click Launch

Cancel Previous Launch

#### (v) Create a Key Pair (don't ignore)

You will be greeted with a screen that will allow you to create a key pair. This step is important. A key pair is a file that is needed to connect to your AWS instance. You will be allowed to download the key pair only once. So download it now and store it safely. This is an additional layer of security that AWS imposes.

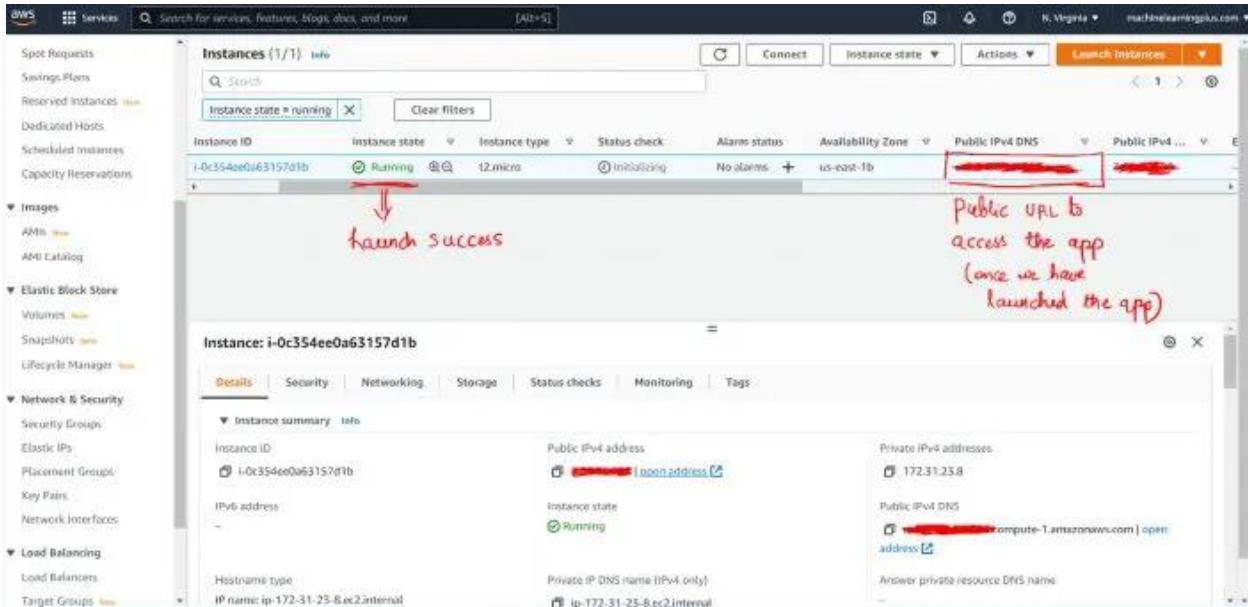


Clicking 'Download Key Pair' will download a `\*.pem` file. In this case it will be called 'free\_tier\_key\_pair.pem'.

Let's keep this safe. Once downloaded, click 'Launch Instances'.

Wait for a few seconds and the instance will be launched.

You then will be able to see a live instance by clicking the instances button on the EC2 dashboard. We will be launching our flask app in this instance.



Next, Let's create a security group.

## Create a Security Group

A security group lets us control who can send requests to the server (instance).

Under 'Network and Security' tab → Select 'Security Groups'. Then Click 'Create Security Group' to create one.

The screenshot shows the AWS EC2 console with the 'Security Groups' section selected. On the left, there's a navigation menu with various services like Capacity Reservations, Images, Elastic Block Store, Network & Security (with 'Security Groups' highlighted), Load Balancing, Auto Scaling, and more. The main pane displays a table of security groups with columns for Name, Security group ID, Security group name, VPC ID, Description, and Owner. Two entries are listed: 'launch-wizard-1' and 'default'. In the top right of the main pane, there's a 'Create security group' button, which is also highlighted with a red box.

Give a name to the security group and keep it default.

The screenshot shows the 'Create security group' wizard. The first step, 'Basic details', is completed with the name 'basicgroup' entered into the 'Name' field. A red arrow points to this field. Below it, there's a 'Description' field with the value 'basicgroup'. Under 'VPC', a dropdown menu is open, showing 'vpc-0d3441db571ebca7c' as the selected option. The 'Inbound rules' and 'Outbound rules' sections are partially visible below, each with their own configuration fields for Type, Protocol, Port range, Source, and Destination.

Now, we need to change the security group for the instance to the new group we just created.

To do that, go to "Network and Security" → "Network Interfaces" → Right click on the instance and select "Change security groups".

The screenshot shows the AWS Management Console with the Network & Security section selected. Under Network Interfaces, a single network interface is listed. A context menu is open over this interface, with the "Change security groups" option highlighted and surrounded by a red box. A handwritten note "Select Network Interfaces" is written next to the left sidebar under the Network Interfaces heading.

Then select the group we just created (*basicgroup*) and hit Save.

Change security groups Info

Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic to and from your instance. You can use this window to add and remove security groups.

**Network interface details**

Network interface ID  
eni-0cb74421652cbb39f

**Associated security groups**

Add one or more security groups to the network interface. You can also remove security groups.

Select security groups

- launch-wizard-1 ( sg-01c5fc837c8938177 )
- launch-wizard-1
- default ( sg-042eefc741f4b0e9b )
- default
- basicgroup ( sg-08a4ede2e5cf9a295 )
- basicgroup
- launch-wizard-1

sg-01c5fc837c8938177

Remove

Cancel Save

## Step 3. Connect to AWS EC2 instance using ssh

New EC2 Experience Tell us what you think.

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Instances (1/1) Info

Search

| Name | Instance ID         | Instance state | Instance type | Status check |
|------|---------------------|----------------|---------------|--------------|
| -    | i-0c354ee0a63157d1b | Running        | t2.r          | passed       |

Actions

- Launch instances
- Launch instance from template
- Migrate a server
- Connect [ ] → Takes to a screen that gives diff options to connect
- Stop instance
- Start instance
- Reboot instance
- Hibernate instance
- Terminate instance
- Instance settings
- Networking
- Security
- Image and templates
- Monitor and troubleshoot

Instance: i-0c354ee0a63157d1b

Details Security Networking Storage Status checks Mon

Instance summary Info

Instance ID: i-0c354ee0a63157d1b Public IPv4 address: 3.138.121.132

The following screen shows the instructions on how to connect to the ubuntu AWS EC2 instance from your local computer.

The screenshot shows the AWS Management Console interface for connecting to an EC2 instance. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar containing 'Search for services, features, blogs, docs, and more.', and a keyboard shortcut '[Alt+S]'. Below the navigation is a breadcrumb trail: 'EC2 > Instances > i-0c354ee0a63157d1b > Connect to instance'. The main content area has a title 'Connect to instance' with an 'Info' link. A sub-instruction says 'Connect to your instance i-0c354ee0a63157d1b using any of these options'. Below this are four tabs: 'EC2 Instance Connect', 'Session Manager', 'SSH client' (which is selected and highlighted in orange), and 'EC2 Serial Console'. Under the 'SSH client' tab, the 'Instance ID' is listed as 'i-0c354ee0a63157d1b'. Step-by-step instructions follow: 1. Open an SSH client. 2. Locate your private key file. The key used to launch this instance is `free_tier_key_pair.pem`. 3. Run this command, if necessary, to ensure your key is not publicly viewable:  
`chmod 400 free_tier_key_pair.pem`. 4. Connect to your instance using its Public DNS:  
`ec2-3-80-25-198.compute-1.amazonaws.com`. An 'Example:' section shows the command:  
`ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com`. A note in a callout box states: 'Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' At the bottom right of the main content area is a 'Cancel' button.

To make the connection with the remote AWS EC2 instance, `cd` to the folder that contains the Key Pair file, which in this case is `free\_tier\_key\_pair.pem`.

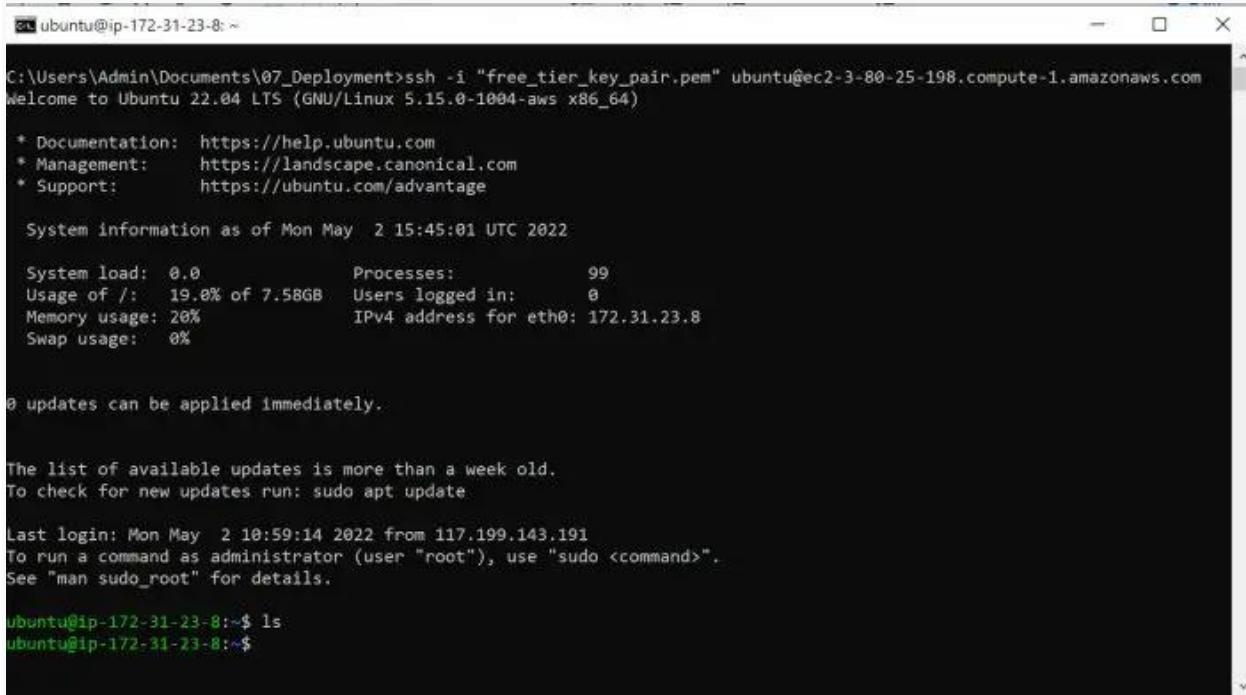
Then, as shown in the screen, type the following command from your command prompt if you are on windows, or the Terminal if you are on a Mac or Linux computer.

```
ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com
```

In the above command, you will need to replace the name of the pem file and the ec2 instance url.

The default username however in most cases is `ubuntu` so you can keep the `ubuntu@ec2..` part as it is.

This will make an `ssh` connection to the AWS EC2 instance. `ssh` stands for secure shell.



A screenshot of a terminal window titled "Ubuntu@ip-172-31-23-8: ~". The window displays the output of an SSH command. It starts with the welcome message for Ubuntu 22.04 LTS. Below that, it shows system documentation links and a summary of system information as of May 2, 2022. It includes details like system load (0.0), usage of / (19.0% of 7.58GB), memory usage (20%), swap usage (0%), and an IPv4 address (172.31.23.8). A note states that 0 updates can be applied immediately. At the bottom, it shows the last login details (Mon May 2 10:59:14 2022) and instructions for running commands as root using sudo. The prompt at the end is "ubuntu@ip-172-31-23-8:~\$".

```
C:\Users\Admin\Documents\07_Deployment>ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1004-aws x86_64)

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

System information as of Mon May  2 15:45:01 UTC 2022

System load:  0.0           Processes:      99
Usage of /:   19.0% of 7.58GB  Users logged in:   0
Memory usage: 20%           IPv4 address for eth0: 172.31.23.8
Swap usage:   0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon May  2 10:59:14 2022 from 117.199.143.191
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-23-8:~$ ls
ubuntu@ip-172-31-23-8:~$
```

## Step 4. Move your files to AWS Ec2 using Secure Copy (scp)

**Let's move the project folder to AWS.**

To do this, you need to be in the Windows command prompt. So open a new command prompt and cd to the folder that contains the project directory and issue the following command to secure copy the files to EC2 instance.

```
scp -r -i "free_tier_key_pair.pem" ./flask_classification  
ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com:~/
```

Again you will need to update the path to pem file and the ec2 url.



```
C:\Users\Admin\Documents\07_Deployment>scp -r -i "free_tier_key_pair.pem" ./flask_classification ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com:~/  
app.py  
catboost_training.json  
events.out.tfevents  
learn_error.tsv  
time_left.tsv  
create_model.py  
german_creditrisk_data.csv  
ml_model.pkl  
base-checkpoint.html  
index-checkpoint.html  
base.html  
index.html
```

Again ssh to the remote instance, and check if you can find the files there.

```
ubuntu@ip-172-31-23-8: ~
C:\Users\Admin\Documents\07_Deployment>ssh -i "free_tier_key_pair.pem" ubuntu@ec2-3-80-25-198.compute-1.amazonaws.com
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1004-aws x86_64)

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

 System information as of Mon May  2 16:08:24 UTC 2022

 System load: 0.0          Processes:           100
 Usage of /: 20.4% of 7.58GB   Users logged in:     0
 Memory usage: 23%          IPv4 address for eth0: 172.31.23.8
 Swap usage:  0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon May  2 15:45:02 2022 from 117.199.141.85
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-23-8:~$ ls
flask_classification
ubuntu@ip-172-31-23-8:~$
```

## Step 5. Install the necessary packages and run app.py to start the app

```
sudo apt-get update
```

```
sudo apt-get -y install python3-pip
```

```
pip3 install <each of the following packages>
```

Packages needed:

**catboost**

**flask**

Once the packages are installed, cd to the `flask_classification` directory and run `python app.py`. This should start the app and make it run from Amazon EC2 instance.

The screenshot shows a web browser window with the URL `ec2-54-242-1-245.compute-1.amazonaws.com:5001` in the address bar. The page title is "Assess the credit risk". A central modal dialog is titled "Enter the details". It contains several input fields and radio button groups:

- Age**: An input field with a blue underline.
- Sex**: Radio button groups for "Male" and "Female".
- Job**: Radio button groups for "0", "1", "2", and "3".
- Housing**: Radio button groups for "Own", "Free", and "Rent".
- Savings Account**: Radio button group for "Little".

## EXPERIMENT 5

### Create virtual storage with AWS S3 Bucket.

**AIM:** To Create virtual storage with AWS S3 Bucket.

**REQUIREMENTS:** AWS active account and desktop.

#### **PROCEDURE:**

1. sign into your console and select S3
2. Select buckets

The screenshot shows the AWS S3 landing page. On the left, there's a large 'Amazon S3' logo with a red circle around it. Below it, the text 'Store and retrieve any amount of data from anywhere' is displayed. A small note at the bottom says 'Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.' On the right, there's a 'Create a bucket' button with a red arrow pointing to it. Below the button, there's a section titled 'Pricing' with a note about no minimum fees and a link to the AWS Simple Monthly Calculator.

3. Click on create bucket

The screenshot shows the 'Create bucket' configuration page in the AWS S3 console. The 'General configuration' section is visible, with the 'Bucket name' field set to 'ccfdptest'. The 'AWS Region' dropdown is set to 'US East (N. Virginia) us-east-1'. The 'Object Ownership' section shows 'ACLs disabled (recommended)' selected. At the bottom, there are links for 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

#### 4. Give name and region

The screenshot shows the 'Create bucket' page in the AWS S3 console. In the 'General configuration' section, the 'Bucket name' field contains 'ccfdptest'. The 'AWS Region' dropdown is set to 'US East (N. Virginia) us-east-1'. Below these fields, there is a note about copying settings from an existing bucket, with a 'Choose bucket' button. In the 'Object Ownership' section, the 'ACLs disabled (recommended)' option is selected. The status bar at the bottom indicates the browser is looking for language selection.

#### 5. ACLS disabled

#### 6. Default encryption disable

The screenshot shows the 'Create bucket' page in the AWS S3 console. In the 'Default encryption' section, the 'Server-side encryption' dropdown is set to 'Disable'. There is also an 'Advanced settings' section with a note about uploading files after creation. At the bottom right, there are 'Cancel' and 'Create bucket' buttons. The status bar at the bottom indicates the browser is looking for language selection.

## 7. Click on create bucket

The screenshot shows the AWS S3 Management Console. On the left, there's a sidebar with options like 'Buckets', 'Access Points', 'Object Lambda Access Points', etc. The main area shows an 'Account snapshot' with storage usage information and a 'Buckets (1)' section. One bucket, 'ccfdptest', is listed with details: Name: ccfdptest, AWS Region: Asia Pacific (Mumbai) ap-south-1, Access: Objects can be public, Creation date: December 21, 2022, 13:38:09 (UTC+05:30). At the top right of the bucket list, there's a 'Create bucket' button.

8. once bucket is created select your bucket and click on it you will be able to see upload
  - Here you can upload any files or folders
  - And keep them for public view or restricted as your wish
  - Once image or file or folder uploaded you will get success message

The screenshot shows the 'Objects' tab for the 'ccfdptest' bucket. It displays two objects. At the top, there are buttons for 'Upload' (highlighted with a red arrow), 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', and 'Create folder'. Below these buttons is a search bar labeled 'Find objects by prefix'.

9. click on upload and then add files

The screenshot shows the AWS S3 Management Console with the 'Upload' interface. At the top, there's a message: '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 this is a large input area with a placeholder: 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folder.' A red arrow points to the 'Add files' button. Below the input area, there's a section titled 'Files and folders (0)' with a 'Remove' button, 'Add files' button, and 'Add folder' button. A search bar and a navigation bar with page numbers (1, 2) are also present. The main content area below says 'No files or folders' and 'You have not chosen any files or folders to upload.' On the left, there's a sidebar with 'Destination' and a dropdown menu.

10. select file or folder to copy from the list

The screenshot shows a Windows File Explorer window with the 'Open' dialog open. The dialog shows a list of quick access locations like Desktop, Downloads, Documents, Pictures, Music, Screenshots, SF, and Videos. The 'File name:' field is empty, and there are 'Open' and 'Cancel' buttons. To the right of the dialog, the main File Explorer window shows a folder structure under 'kmit'. A red dashed box highlights the 'Add folder' button in the dialog. The main window shows a message: 'CLI, AWS SDK or Amazon S3, or Add folder.' Below the dialog, there's a 'Destination' section with a dropdown set to 's3://ccfdptest'. The bottom of the screen shows the Windows taskbar with various pinned icons and the date/time.

11. click on open and you can view th file being uploaded

The screenshot shows the AWS S3 'Upload' interface. In the center, there's a large dashed blue box for dragging files or choosing them. Below it, a table lists the file: 'pixels-photo-1108099.webp' (image/webp, 28.2 KB). To the right, there are 'Remove', 'Add files', and 'Add folder' buttons. On the left, under 'Destination', it shows 's3://ccfdptest'. A 'Destination details' section is partially visible. At the bottom, there's a feedback bar and a Windows taskbar at the very bottom.

## 12. once image uploaded you can view the success status

This screenshot shows the 'Upload: status' page after the file was uploaded. It has a green header bar with the message 'Upload succeeded'. Below it, a summary table shows 'Succeeded' (1 file, 28.2 KB) and 'Failed' (0 files, 0 B). The 'Files and folders' tab is selected, showing the same table as the previous screenshot but with a green 'Succeeded' status next to the file entry. The Windows taskbar is visible at the bottom.

## 13. once image is uploaded you can copy the URL link and share it or paste it in browser to view it.

**10.000+ Best Pictures**

Inbox (90) - khushi | CC-FDP-KMIT - Go | CC-FDP TASK 6 - Go | FDP-CC - Google | CC-FDP TASK 3 - Go | ccfdfptest - S3 buck | 10.000+ Best Pictures | + | - | X

s3.console.aws.amazon.com/s3/buckets/ccfdfptest?region=ap-south-1&tab=objects

aws Services Search [Alt+S] Global khushi

Amazon S3 > Buckets > ccfdfptest

ccfdfptest Info

Objects Properties Permissions Metrics Management Access Points

**Objects (3)**

Objects are the fundamental entities in Amazon S3. 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

Object URL Copied

C Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

| Name                      | Type | Last modified                           | Size    | Storage class |
|---------------------------|------|-----------------------------------------|---------|---------------|
| download (1).jpg          | jpg  | December 21, 2022, 13:41:19 (UTC+05:30) | 9.7 KB  | Standard      |
| download (2).jpg          | jpg  | December 21, 2022, 13:47:09 (UTC+05:30) | 8.7 KB  | Standard      |
| pixels-photo-1108099.webp | webp | January 2, 2023, 12:52:19 (UTC+05:30)   | 28.2 KB | Standard      |

Feedback Looking for language selection? Find it in the new Unified Settings

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Show all

pixels-photo-11...webp

Type here to search

12:53 PM 1/2/2023

**14. Output**

Inbox (90) - khushi | CC-FDP-KMIT - Go | CC-FDP TASK 6 - Go | FDP-CC - Google | CC-FDP TASK 3 - Go | ccfdfptest - S3 buck | download+(2).jpg | Show all | + | - | X

s3.console.aws.amazon.com/download+(2).jpg

12:53 PM 1/2/2023

pixels-photo-11...webp

Type here to search

12:53 PM 1/2/2023

**15. you can delete the bucket with all your files permanently**

**Empty bucket** Info

**⚠** • Emptying the bucket deletes all objects in the bucket and cannot be undone.  
 • Objects added to the bucket while the empty bucket action is in progress might be deleted.  
 • To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

[Learn more](#)

If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket. [Learn more](#)

[Go to lifecycle rule configuration](#)

**Permanently delete all objects in bucket "ccfdptest"?**

To confirm deletion, type *permanently delete* in the text input field.

permanently delete

[Cancel](#) [Empty](#)

## 16. select bucket to be deleted and click on empty bucket

**Successfully emptied bucket "ccfdptest"**  
 View details below. If you want to delete this bucket, use the [delete bucket configuration](#).

**Empty bucket: status**

The details below are no longer available after you navigate away from this page.

| Summary                  |                                            |                               |
|--------------------------|--------------------------------------------|-------------------------------|
| Source<br>s3://ccfdptest | Successfully deleted<br>3 objects, 46.6 KB | Failed to delete<br>0 objects |

**Failed to delete (0)**

| Name                       | Prefix | Version ID | Type | Last modified | Size | Error |
|----------------------------|--------|------------|------|---------------|------|-------|
| No failed object deletions |        |            |      |               |      |       |

## 17. click on delete bucket to completely delete bucket items along with storage

The screenshot shows two windows side-by-side. The top window is the AWS S3 'Delete bucket' confirmation dialog. It displays a warning message: 'Deleting a bucket cannot be undone.' and 'Bucket names are unique. If you delete a bucket, another AWS user can use the name.' Below this is a text input field containing 'ccfdptest'. At the bottom are 'Cancel' and 'Delete bucket' buttons. The bottom window is a Windows desktop taskbar showing the AWS S3 icon. A green notification bar at the top of the screen says 'Successfully deleted bucket "ccfdptest"' with a small info icon. A red arrow points from the text in the notification bar to the same text in the AWS dialog. The taskbar also shows other icons for File Explorer, Edge, Google Photos, Mail, and Google Chrome.

## EXPERIMENT 6

### Create extended memory storage using Amazon EBS

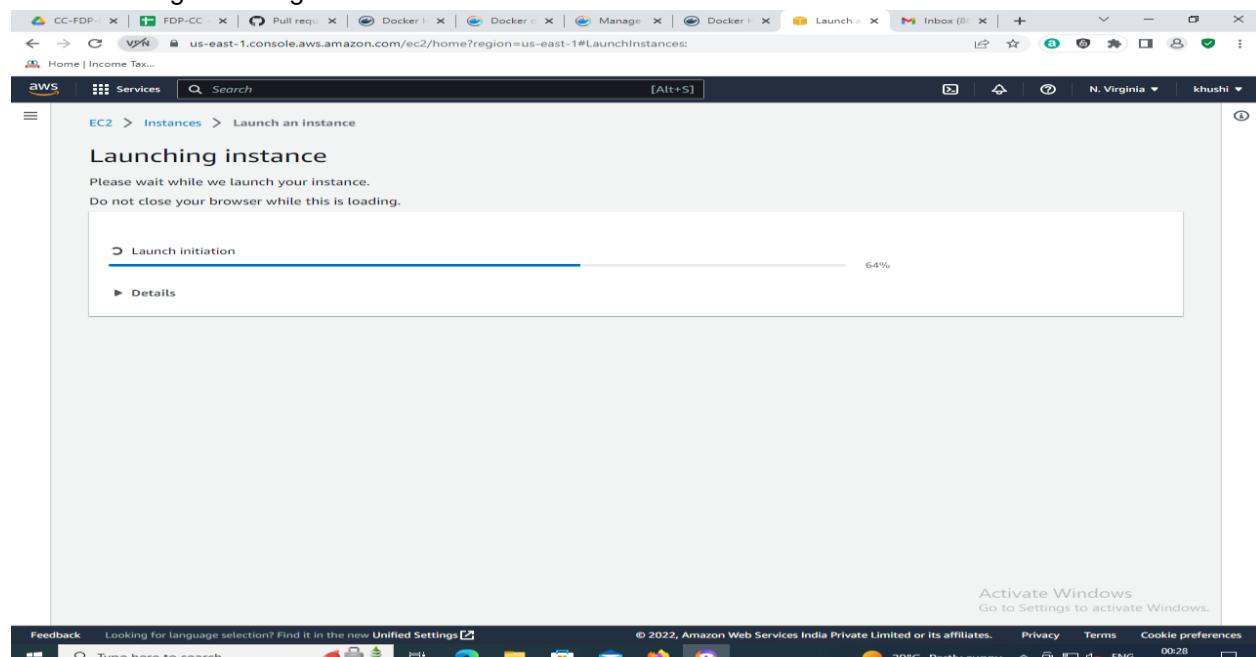
**AIM:** To Create extended memory storage using Amazon EBS, Show Rapid elasticity using EBS (EC2 windows instance + create EBS volumes)

**REQUIREMENTS:** AWS active account and desktop.

#### **PROCEDURE:**

Step1: create an ec2 instance

- i) go to my account
- ii) management console
- iii) console home
- iv) search ec2 and select instances
- v) launch an instance
  1. EBS test
  2. Windows
  3. Free tier
  4. Keypair old one select .pem file
  5. n/w security as existing
  6. Configure storage 30GB



- vi) connect to instance

**Success**  
Successfully initiated launch of instance (i-041e0d7cc8d195b7e)

**Next Steps**

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database

**Create billing alerts** [View](#)

Once your instance is running, log into it from your local computer.

**Connect to instance** [View](#)

**Learn more** [View](#)

**Connect an RDS database** [View](#)

**Create a new RDS database** [View](#) **Learn more** [View](#)

**View all instances**

Activate Windows  
Go to Settings to activate Windows.

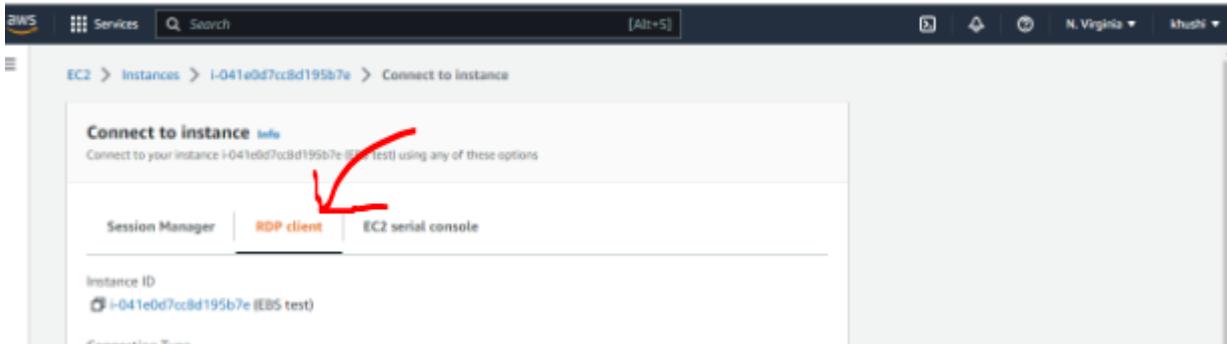
## Step2:select an instance, right click on it and select connect to instance

**Instances (2) Info**

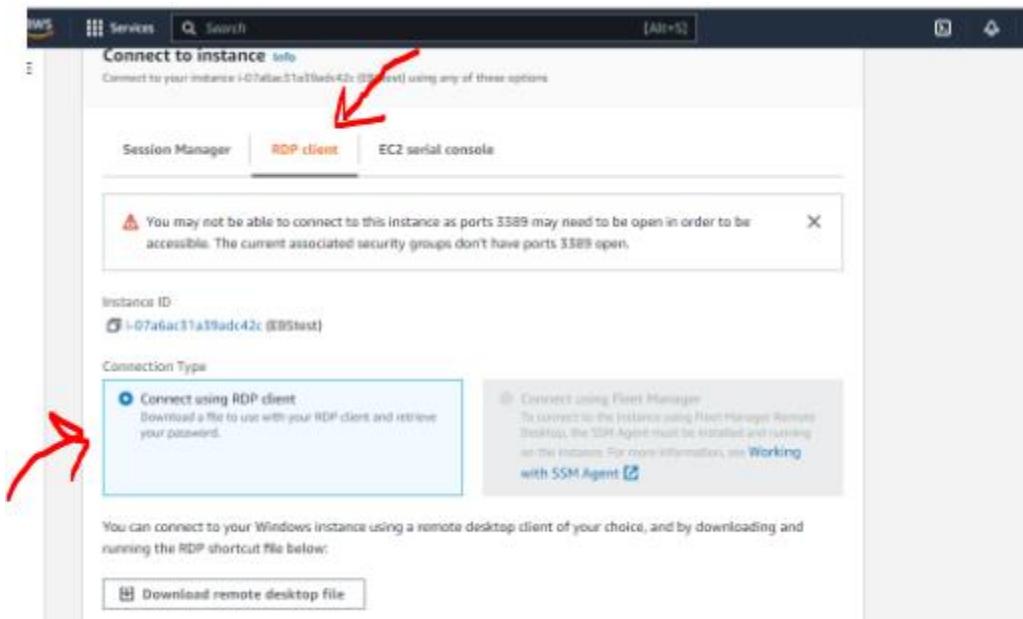
| Name     | Instance ID         | Instance state | Instance type | Status check | Alarm status | Ava  |
|----------|---------------------|----------------|---------------|--------------|--------------|------|
| test1    | i-05b263bb0a4d66a8d | Stopped        | t2.micro      | -            | No alarms    | us-e |
| EBS test | i-041e0d7cc8d195b7e | Pending        | t2.micro      | -            | No alarms    | us-e |

**Select an instance**

i) select RDP client



ii) connect using RDP client



iii) get password (not available) (keep reloading page till you get the password)

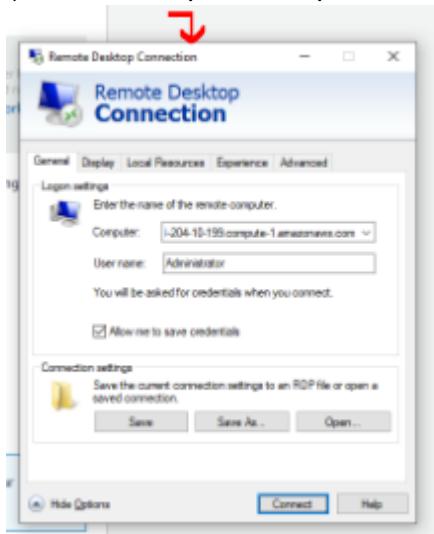
iv) once you get the password click on it

- > copy the key pair
- >you will get a .pem file
- > copy the public DNS



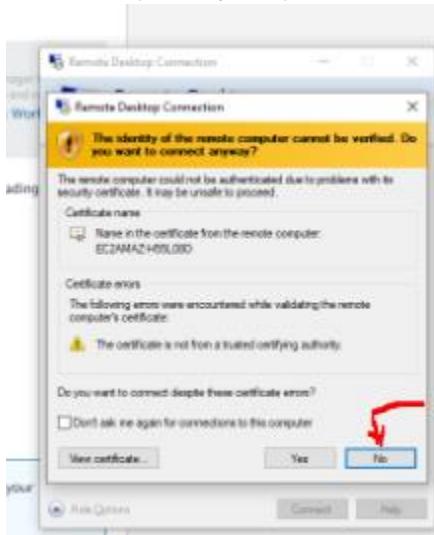
### Step 3: Add PC

- PC name-> paste the public DNS

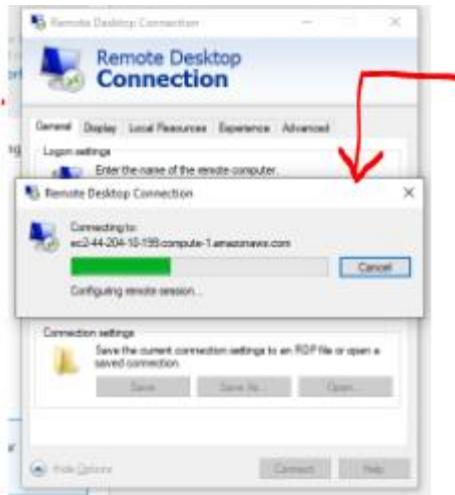


- add user account

- > username administrator
- > paste your password under password



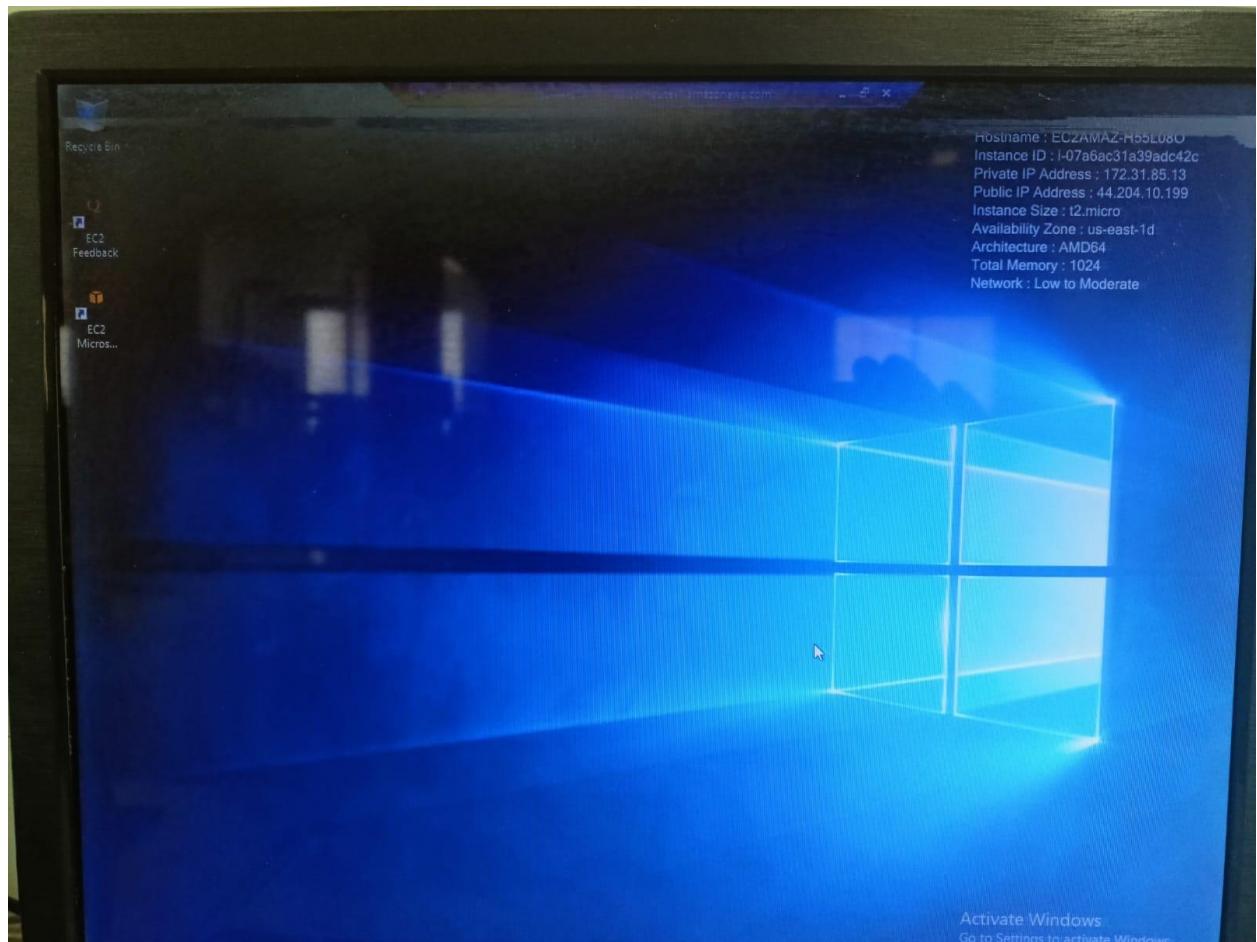
- click on add



iv) reconnect select and finally click on add.

Step 4: now select the new PC

- Click on connect
- Wait for the new PC to initialise



step5) now go back to your AWS account and select volumes and click on create volumes

The screenshot shows the AWS EC2 Volumes page. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Images, and Elastic Block Store. Under EBS, 'Volumes' is selected. The main area displays a table titled 'Volumes (1/1)' with one row. The row details are:

| Name | Volume ID             | Type | Size   | IOPS | Throughput | Snapshot        |
|------|-----------------------|------|--------|------|------------|-----------------|
| -    | vol-091d7d84ad4c2af2e | gp2  | 30 GiB | 100  | -          | snap-0c3ab6d... |

A message at the top right says: "You can now create Amazon Data Lifecycle Manager policies to automate snapshot management directly from this screen. Select the volumes to back up, and then choose Actions, Create snapshot lifecycle policy. For more information, see the Knowledge Center article." There are 'Actions' and 'Create volume' buttons.

At the bottom, there's a 'Details' tab and other tabs for Status checks, Monitoring, and Tags. The status bar at the bottom shows 'abc1.pem' and 'Verified'.

The screenshot shows the 'Create volume' wizard in the AWS Management Console. The user is configuring a new EBS volume with the following parameters:

- Volume type:** General Purpose SSD (gp2)
- Size (GiB):** 20
- IOPS:** 100 / 3000
- Throughput (MiB/s):** Not applicable
- Availability Zone:** us-east-1d
- Snapshot ID - optional:** Don't create volume from a snapshot
- Encryption:** abc1.pem (Verified)

The browser tab bar shows multiple open tabs, including 'Create volume | EC2' which is currently active. The Windows taskbar at the bottom displays various pinned icons and system status.

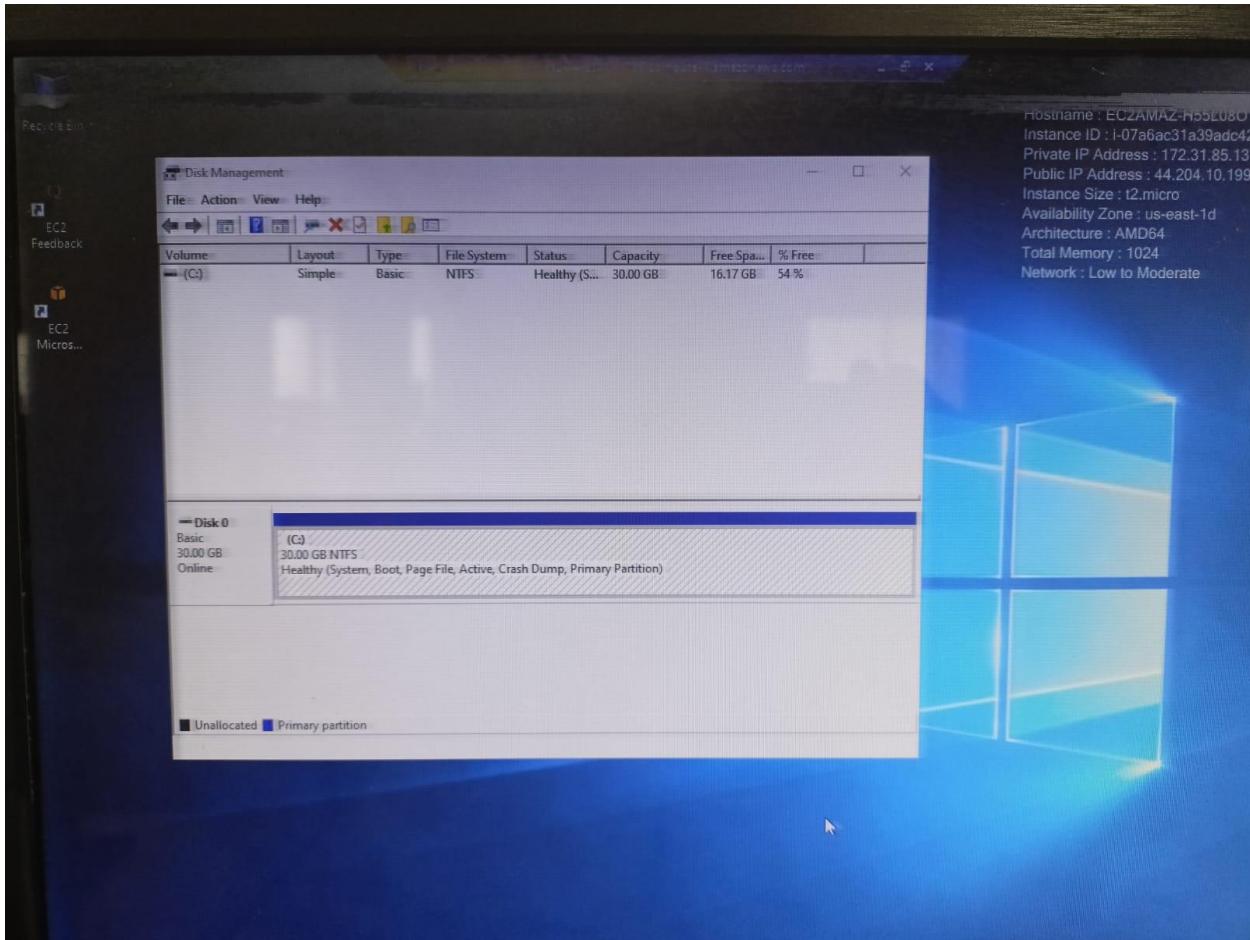
The screenshot shows the AWS EC2 Volumes page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Images, and Elastic Block Store. Under Elastic Block Store, 'Volumes' is selected. A success message at the top says 'Successfully created volume vol-0088f0a9eb46a42d1.' Below it, a note suggests creating Amazon Data Lifecycle Manager policies. The main area displays a table of volumes:

| Name | Volume ID             | Type | Size   | IOPS | Throughput | Snapshot        |
|------|-----------------------|------|--------|------|------------|-----------------|
| -    | vol-091d7d84ad4c2af2e | gp2  | 30 GiB | 100  | -          | snap-0c3ab6d... |
| -    | vol-0088f0a9eb46a42d1 | gp2  | 20 GiB | 100  | -          | -               |

At the bottom of the page, there are links for Feedback, Terms, Privacy, and Cookie preferences.

## Step 5: in the new PC

- Create and format the disk partitions
- Disk management will open
- We can see the 30GB space allocated



Step 6: go back to your instance and select EBS

- SELECT VOLUMES
- Select any one volume which is available and click on create volume

Step 7: create a 20gb volume and click on create volume

(NOTE: make sure your EBS instance and volume space created under the same zone)

- Wait till the volume space becomes available
- Select new volume
- Click on actions -> attach volume

The screenshot shows the AWS Management Console interface for the Elastic Block Store (EBS) service. The user has successfully created a new volume, as indicated by the green success message at the top: "Successfully created volume vol-0088f0a9eb46a42d1".

The main table displays two volumes:

| Name | Volume ID             | Type | Size   | IOPS | Throughput | Snapshot        |
|------|-----------------------|------|--------|------|------------|-----------------|
| -    | vol-091d7d84ad4c2af2e | gp2  | 30 GiB | 100  | -          | snap-0c3ab6d... |
| -    | vol-0088f0a9eb46a42d1 | gp2  | 30 GiB | 100  | -          | 2012-12-21      |

A context menu is open over the second volume (vol-0088f0a9eb46a42d1). The menu options include:

- Create volume
- Modify volume
- Create snapshot
- Create snapshot lifecycle policy
- Delete volume
- Attach volume** (highlighted)
- Detach volume
- Force detach volume
- Manage auto-enabled I/O
- Manage tags

Below the table, the Volume ID is displayed as vol-0088f0a9eb46a42d1. The Details tab is selected.

The bottom navigation bar includes links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences. The system status bar shows the date (21-12-2022), time (01:09), temperature (30°C), battery level, signal strength, and language (ENG).

The screenshot shows a browser window for the AWS Management Console with the URL [us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AttachVolume:volumId=vol-0088f0a9eb46a42d1](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AttachVolume:volumId=vol-0088f0a9eb46a42d1). The page displays the 'Attach volume' dialog for a specific volume ID. The dialog includes fields for 'Volume ID' (vol-0088f0a9eb46a42d1), 'Availability Zone' (us-east-1d), 'Instance' (i-07a6ac31a39adc42c), and 'Device name' (xvdf). Below these fields, a note states: 'Only instances in the same Availability Zone as the selected volume are displayed.' At the bottom right of the dialog are 'Cancel' and 'Attach volume' buttons. The browser's address bar also shows the URL. The taskbar at the bottom of the screen includes icons for File Explorer, Edge, File Explorer, Mail, Firefox, Task View, File Explorer, and Task View, along with system status indicators like battery level, temperature (30°C), and date/time (21-12-2022).

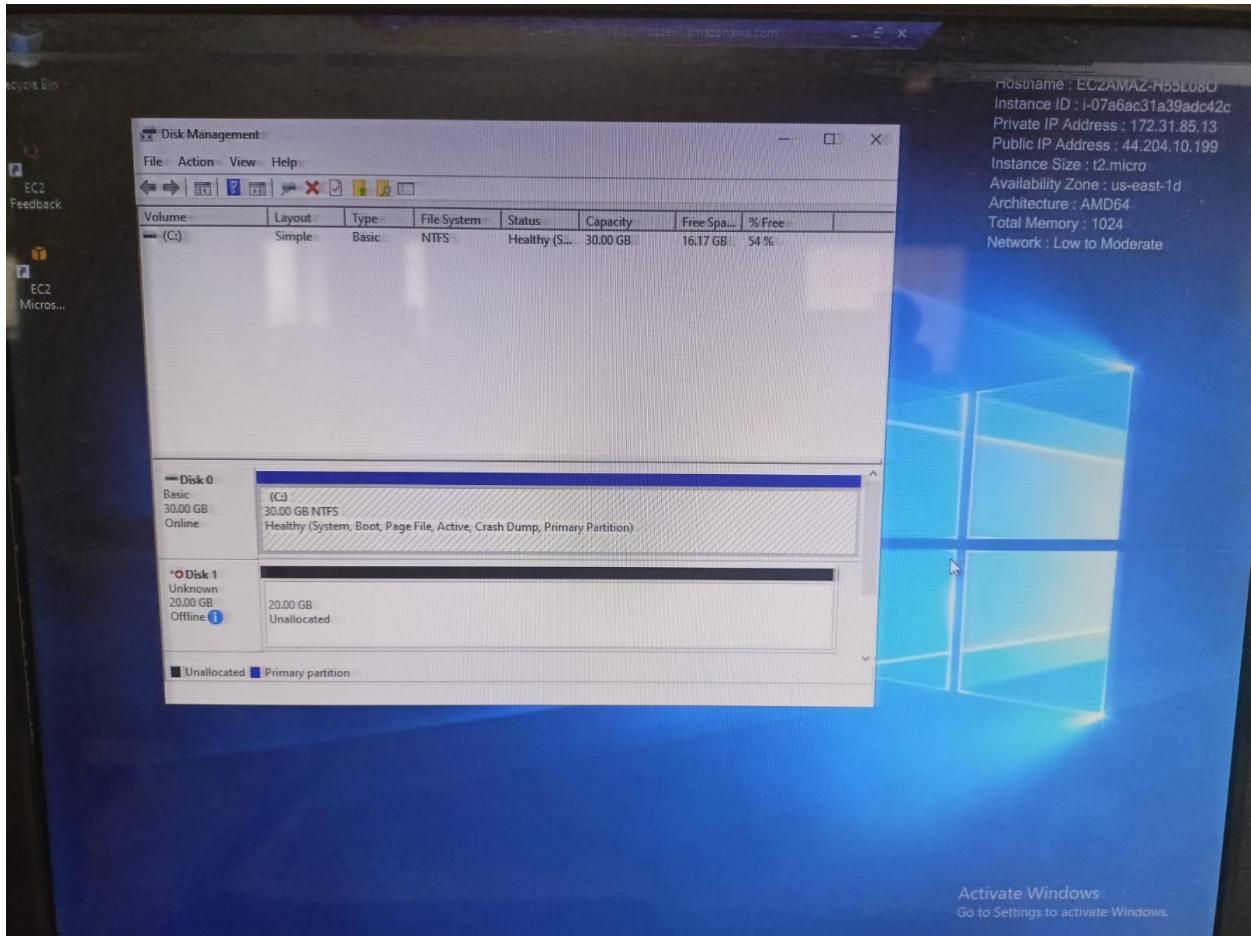
The screenshot shows the AWS Management Console with multiple tabs open at the top, including 'CC-FDP-KMIT - Go...', 'CC-FDP TASK 4 - Go...', 'FDP-CC - Google Slides', 'EBSDemo.mp4 - Google Slides', 'Volumes | EC2 Man...', and 'Inbox (88) - khushi...'. The main window is titled 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Volumes:' and displays the 'New EC2 Experience' interface. A success message at the top states: 'Successfully attached volume vol-0088f0a9eb46a42d1 to instance i-07a6ac31a39adc42c.' Below this, a note says: 'You can now create Amazon Data Lifecycle Manager policies to automate snapshot management directly from this screen. Select the volumes to back up, and then choose Actions, Create snapshot lifecycle policy. For more information, see the Knowledge Center article.' The 'Volumes (2)' table lists two volumes:

| Size   | IOPS | Throughput | Snapshot        | Created                | Availability Zone | Volume state |
|--------|------|------------|-----------------|------------------------|-------------------|--------------|
| 30 GiB | 100  | -          | snap-0c3ab6d... | 2022/12/21 00:49 GMT-8 | us-east-1d        | In-use       |
| 20 GiB | 100  | -          | -               | 2022/12/21 01:08 GMT-8 | us-east-1d        | In-use       |

The left sidebar shows navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The 'Feedback' bar at the bottom suggests looking for language selection in Unified Settings. The taskbar at the bottom includes icons for File Explorer, Edge, File Explorer, Mail, Firefox, Task View, and File Explorer, along with system status indicators for temperature (30°C), battery (ENG), and date/time (21-12-2022, 01:10).

(NOTE: wait for the new volume to be initialised)

Step 8: Go back to MY PC and you can see the disk space added

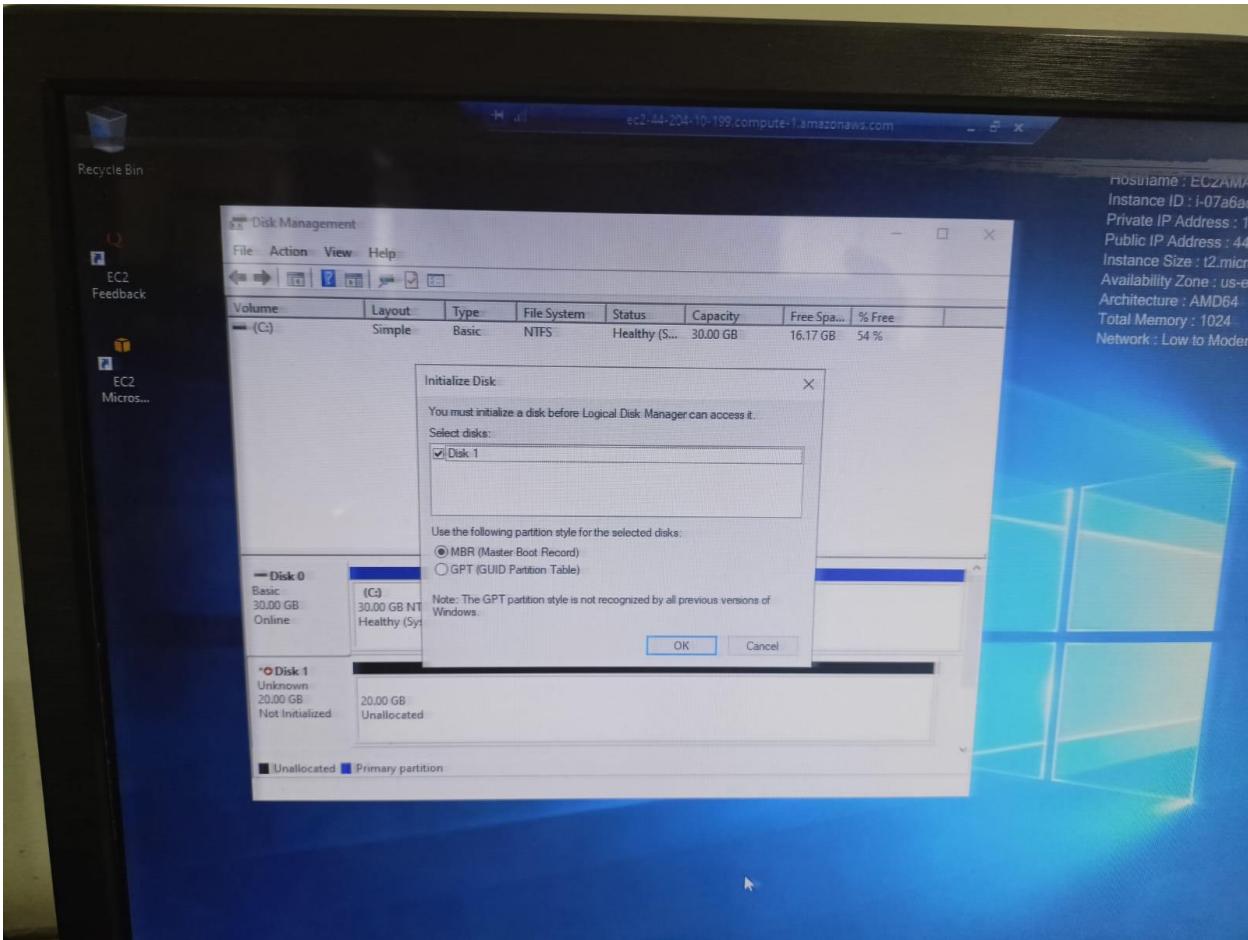


#### Step 9: right click on new disk space

- Select properties
- Make it online
- Click on initialise disk
- Click on ok

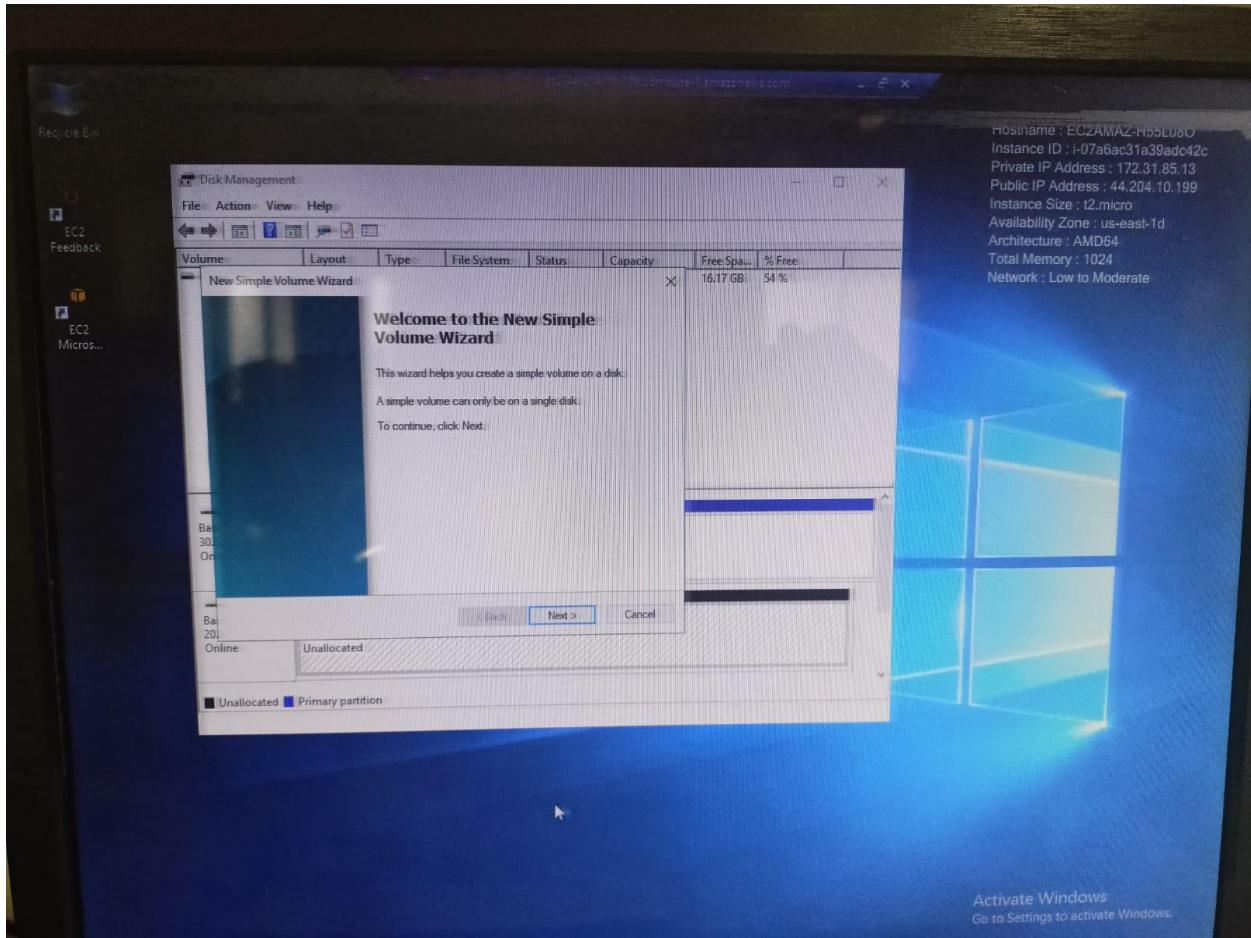
Observe the changes to disk1

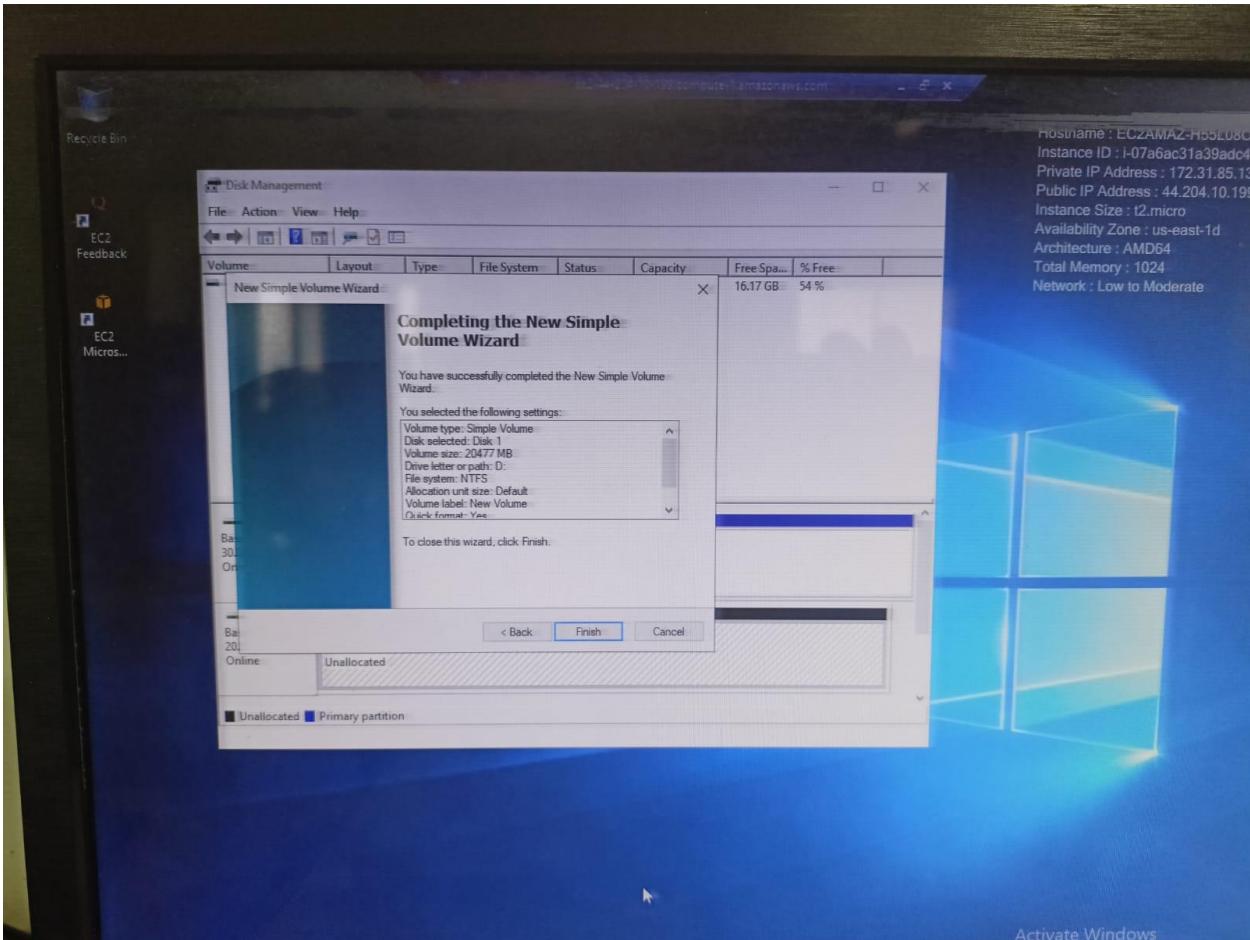
- Basic
- 20gb
- online

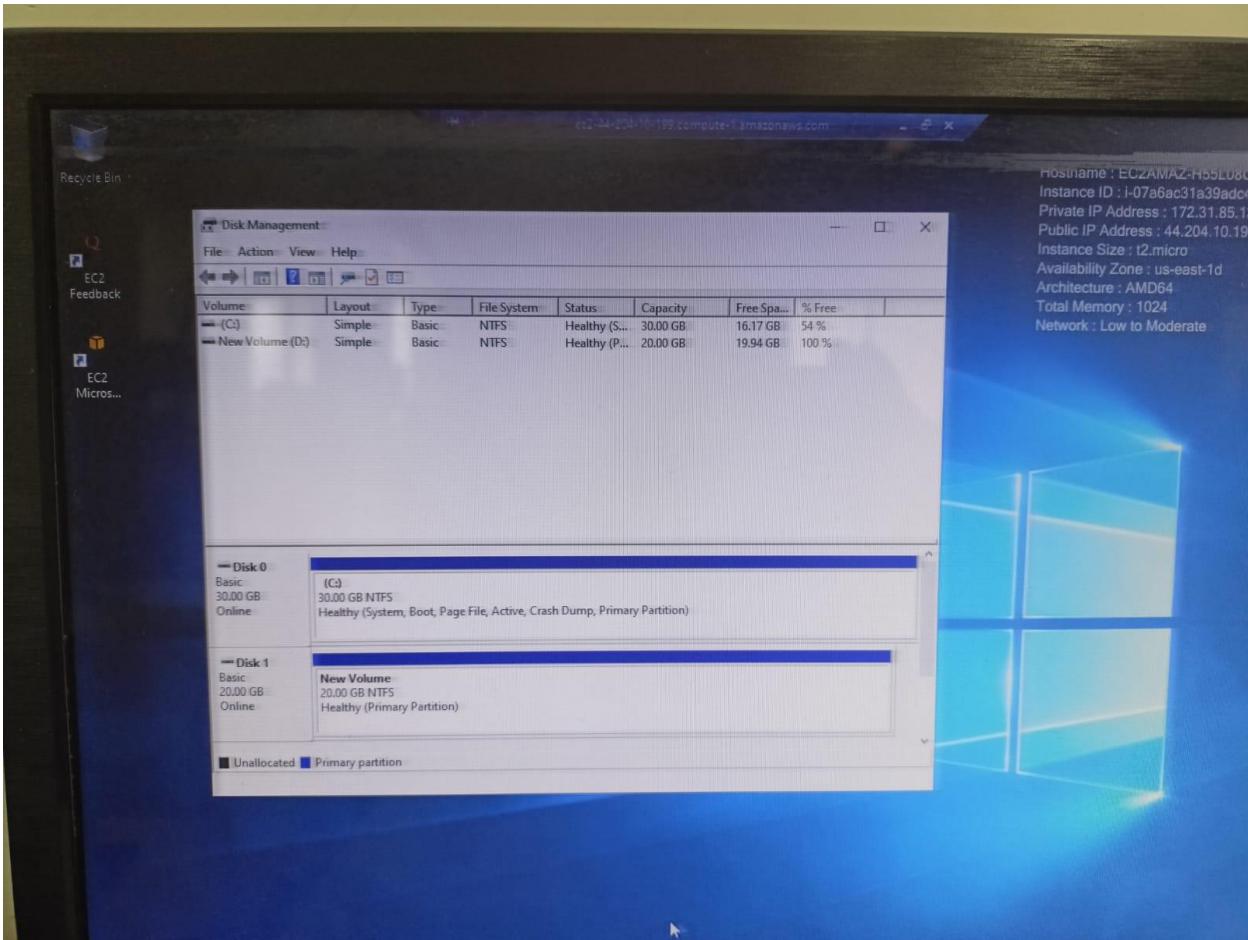


Step 10: right click on the 20gb unallocated space and select new simple volume wizard and click on **next-> next -> assign letter D -> next -> next -> finish**

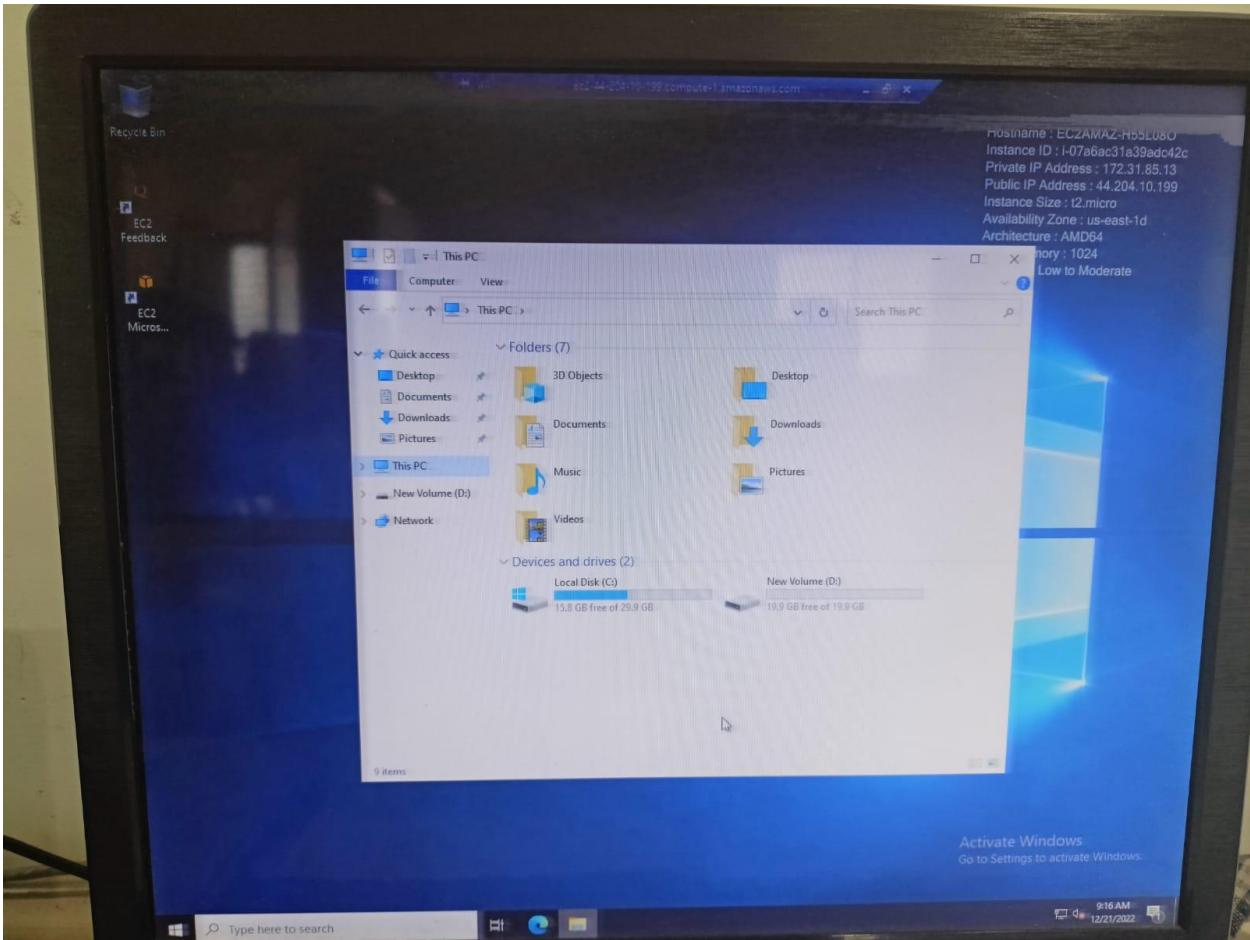
**(observe the volume name changes to new volume(D) and healthy)**







Step 11: goto my files in this PC you can see the two drives



## EXPERIMENT 7

### Create an AWS IAM User with attached policy

**AIM:** To Create IAM Role to access S3 bucket as read only mode from EC2 instance.

**REQUIREMENTS:** AWS active account and desktop.

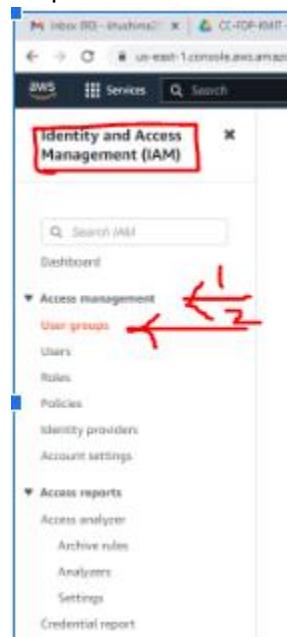
#### **PROCEDURE:**

Step 1:sign into your AWS management console

Step2: select IAM in console services



Step3: under access management select user groups and select administrator access



Step4: click on create user group

- Give name
- Search for admin permission policy
- Select admin access
- Click on create group

Identity and Access Management (IAM)

User groups

| <input type="checkbox"/>            | SystemAdministrator                      | AWS managed - job function |
|-------------------------------------|------------------------------------------|----------------------------|
| <input checked="" type="checkbox"/> | AdministratorAccess                      | AWS managed - job function |
| <input type="checkbox"/>            | AmazonWorkSpacesAdmin                    | AWS managed                |
| <input type="checkbox"/>            | AmazonAPIGatewayAdministrator            | AWS managed                |
| <input type="checkbox"/>            | AWSPrivateEndpointAdministratorAccess    | AWS managed                |
| <input type="checkbox"/>            | AWSSMAdminReadOnlyAccess                 | AWS managed                |
| <input type="checkbox"/>            | AWSSyncAdministrator                     | AWS managed                |
| <input type="checkbox"/>            | AWSSMAdminFullAccess                     | AWS managed                |
| <input type="checkbox"/>            | NetworkAdministrator                     | AWS managed - job function |
| <input type="checkbox"/>            | AmazonWorkSpacesApplicationManagerA...   | AWS managed                |
| <input type="checkbox"/>            | AWSCloudBuildAdminAccess                 | AWS managed                |
| <input type="checkbox"/>            | AWSServiceCatalogAdminFullAccess         | AWS managed                |
| <input type="checkbox"/>            | AWSEligibleActions_HotelPolicyForReso... | AWS managed                |
| <input type="checkbox"/>            | AdministratorAccess_Amplify              | AWS managed                |

Cancel Create group

- You will get success
- Download.csv file

### Step5: under user

Identity and Access Management (IAM)

User groups

Users

Roles

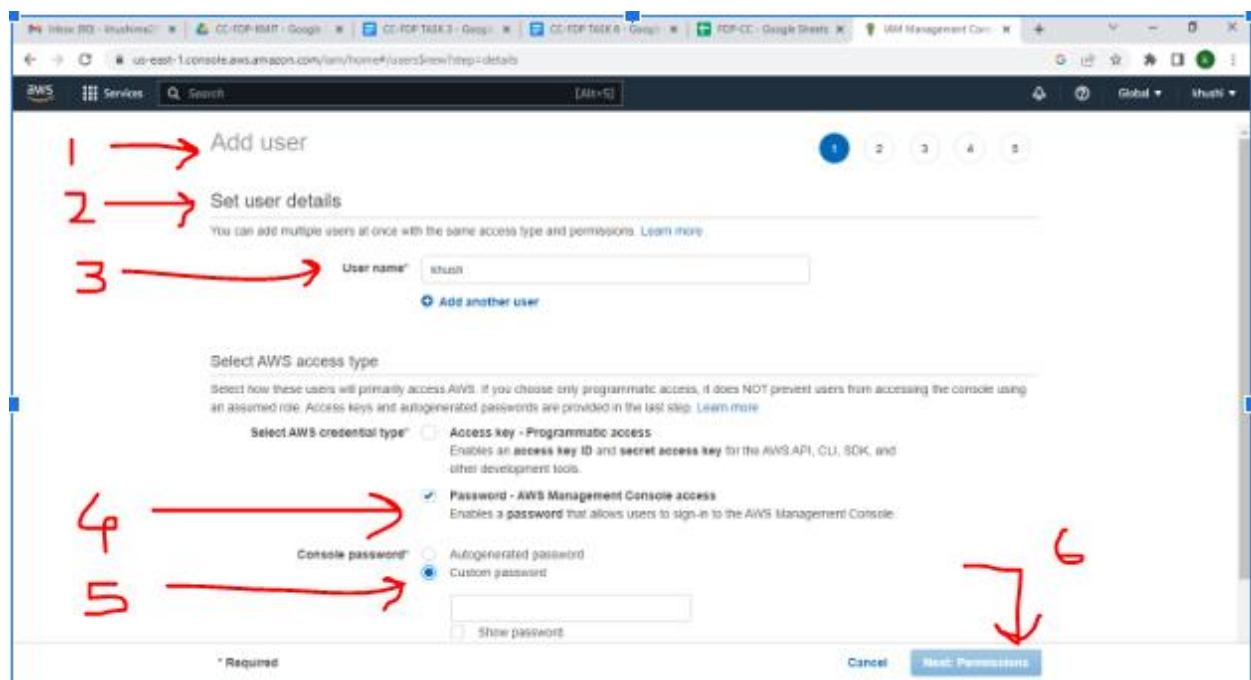
Policies

Identity providers

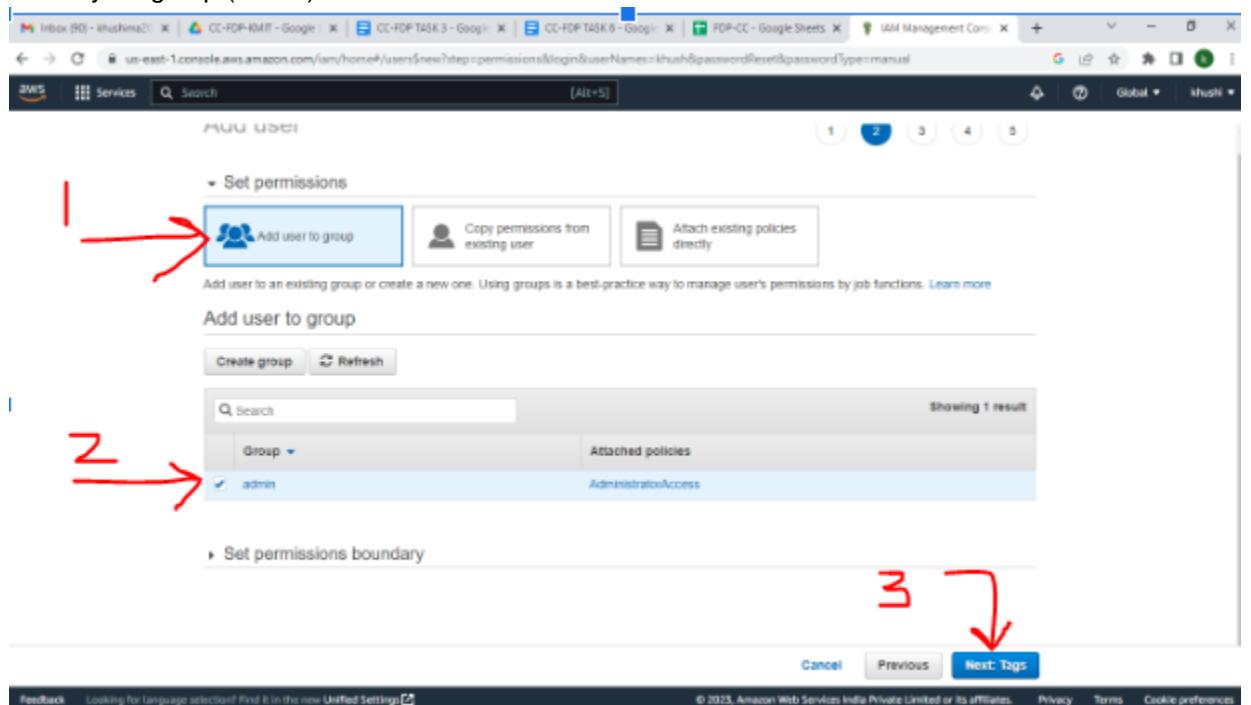
Account settings

Add user

- Select add users



- Set user details
- Username: khushi
- Psswd: -----
- Custom password
- Next permissions
- Add user to group
- Select your group (admin)



- Next tags (leave at default)

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. Learn more

| Key                                      | Value (optional)     | Remove                                |
|------------------------------------------|----------------------|---------------------------------------|
| <input type="text" value="Add new key"/> | <input type="text"/> | <input type="button" value="Remove"/> |

You can add 50 more tags.

default

Cancel Previous Next: Review

- Next review (leave at default)

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

|                        |                                                 |
|------------------------|-------------------------------------------------|
| User name              | khush                                           |
| AWS access type        | AWS Management Console access - with a password |
| Console password type  | Custom                                          |
| Require password reset | Yes                                             |
| Permissions boundary   | Permissions boundary is not set                 |

Permissions summary

The user shown above will be added to the following groups.

| Type           | Name                  |
|----------------|-----------------------|
| Group          | admin                 |
| Managed policy | IAMUserChangePassword |

Tags

No tags were added.

Cancel Previous Create user

1 → Review  
2 → Group  
3 → Managed policy  
4 → Create user

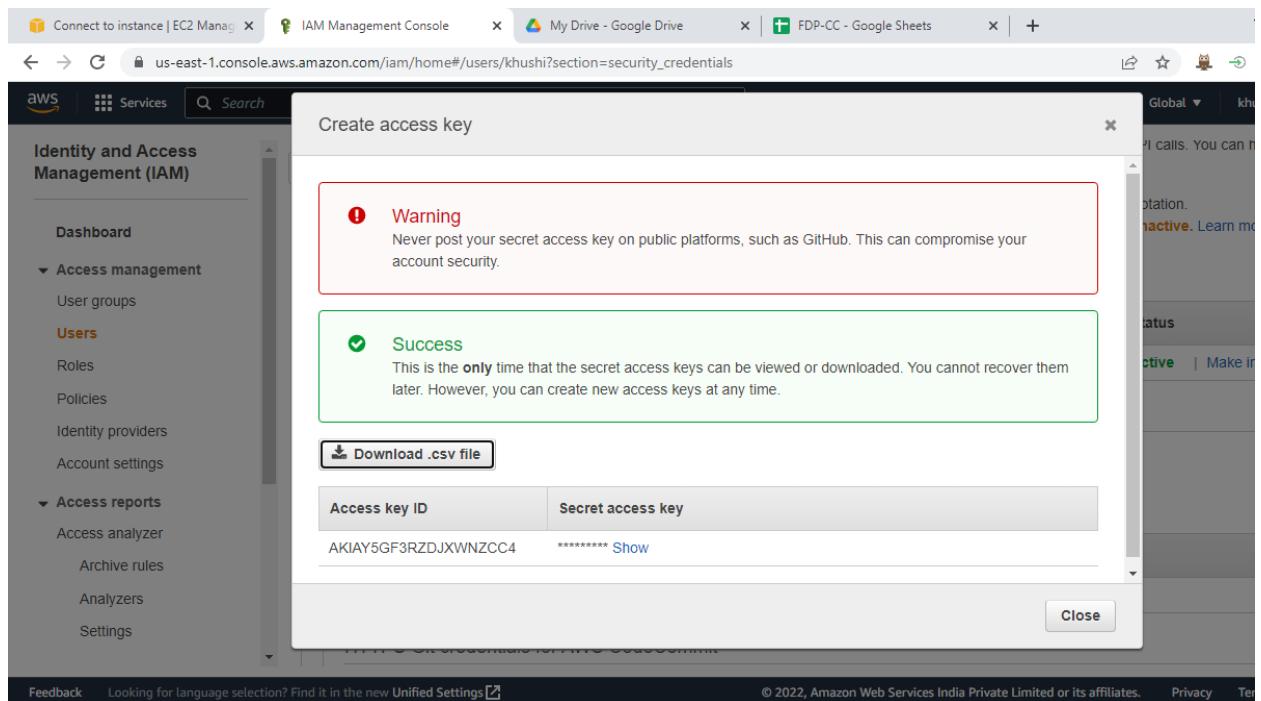
} default

- Click on create user
- You will get success download .csv file

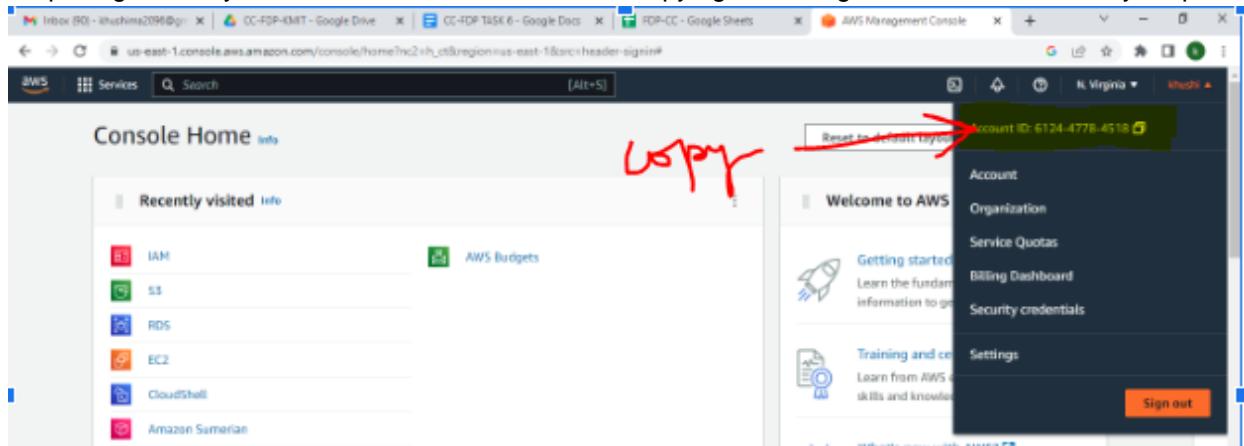
The screenshot shows a browser window with multiple tabs open, including Gmail, Google Sheets, and the AWS IAM Management Console. The main focus is on the AWS IAM Management Console where a new user has been successfully created. A green success message box states: "Success: You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time." Below this message is a link: "Users with AWS Management Console access can sign-in at: https://612447784518.signin.aws.amazon.com/console". There is a "Download .csv" button. A table lists one user: "User" (khush) and "Email login instructions" (Send email). The table has columns for User and Email login instructions. The bottom of the screen shows a Windows taskbar with icons for File Explorer, Edge, File Manager, Mail, and Google Chrome.

#### Step6: in your AWS console goto IAM services

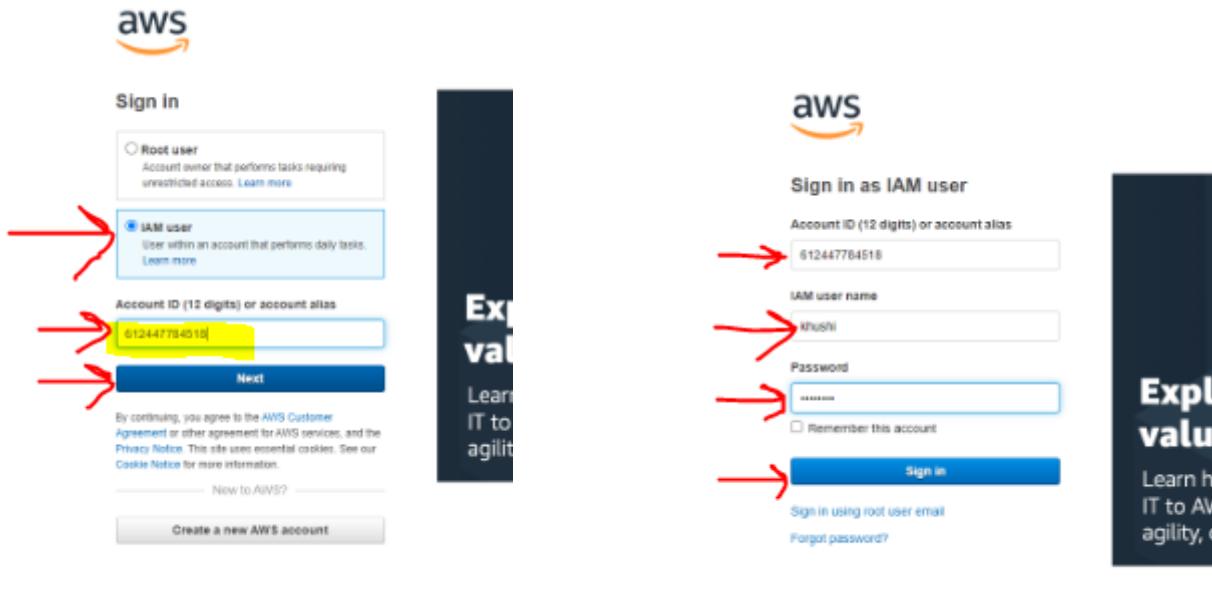
- Select your user
- Goto security credentials
- Access keys
- Create access key
- Copy your access key ID
- Download.csv file



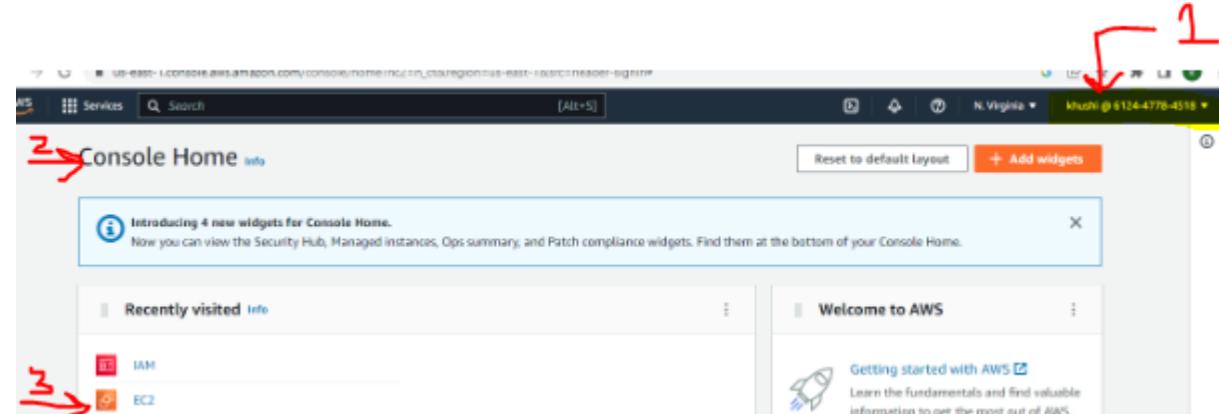
Step7: sign out of your AWS Root user account after copying the 12 digits account ID found in your profile



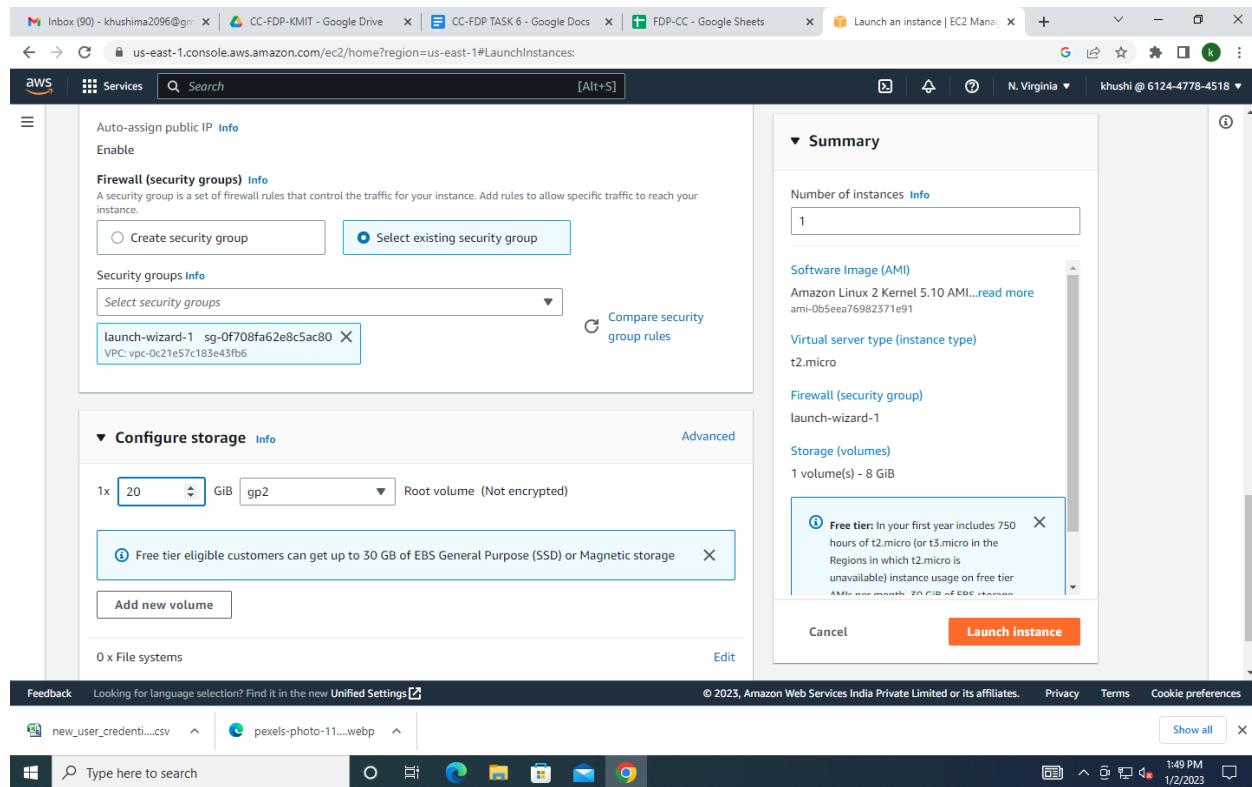
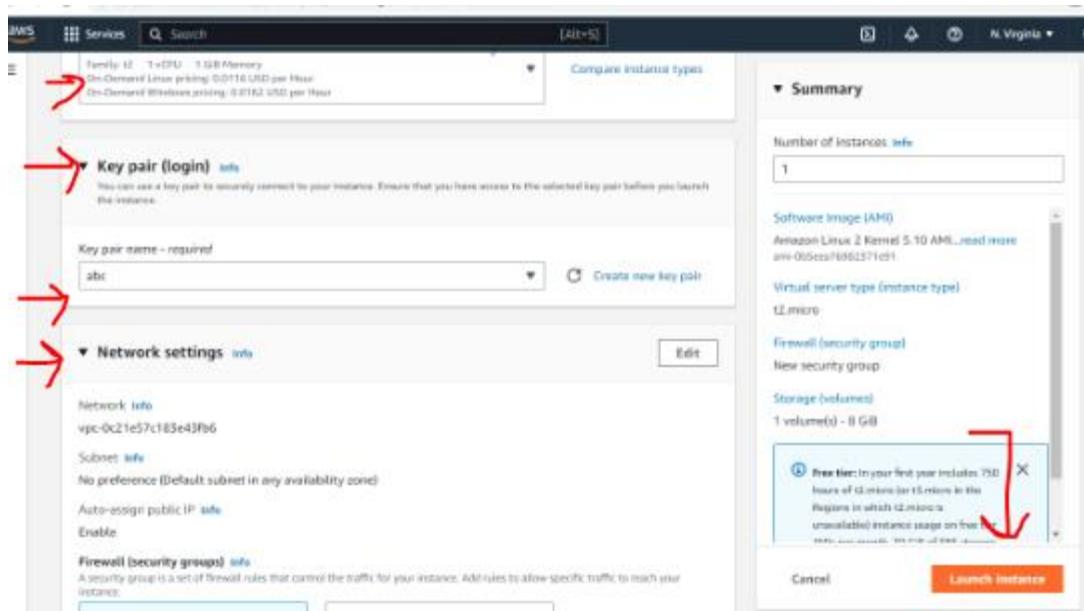
Step8: sign into IAM user account



- And create an ec2 instance



- And launch the instance



The screenshot shows the AWS EC2 Instances Launch an instance page. At the top, there are several tabs: 'Inbox (90) - khushim2096@gmail.com', 'CC-FDP-KMIT - Google Drive', 'CC-FDP TASK 6 - Google Docs', 'FDP-CC - Google Sheets', and 'Launch an instance | EC2 Manager'. Below the tabs, the AWS logo and 'Services' button are visible. A search bar contains 'Search [Alt+S]'. The top right shows 'N. Virginia' and 'khushi @ 6124-4778-4518'. The main content area has a title 'Launching instance' with the sub-instruction 'Please wait while we launch your instance. Do not close your browser while this is loading.' Below this is a progress bar labeled 'Launch initiation' with a value of '64%'.

This screenshot shows the Windows taskbar. It includes the Start button, a search bar with the placeholder 'Type here to search', and several pinned icons for File Explorer, Edge, File Explorer, Mail, and Google Chrome. The system tray shows the date and time as '1:49 PM 1/2/2023'.

The screenshot shows the AWS EC2 Instances Launch an instance page after a successful launch. The top section displays a green checkmark icon and the message 'Successfully initiated launch of instance (i-0840692f9c6280704)'. Below this is a 'Launch log' link. The main content area features a 'Next Steps' section with three cards: 'Create billing and free tier usage alerts', 'Connect to your instance', and 'Connect an RDS database'. Each card includes a descriptive text and a call-to-action button. At the bottom right of the main content area is a red 'View all instances' button.

This screenshot shows the Windows taskbar, identical to the one in the previous image, with the search bar, pinned icons for File Explorer, Edge, File Explorer, Mail, and Google Chrome, and the system tray showing the date and time as '1:49 PM 1/2/2023'.

- And connect to it

The screenshot shows the AWS EC2 Connect interface. At the top, there are tabs for 'EC2 Instance Connect', 'Session Manager', 'SSH client', and 'EC2 serial console'. The 'EC2 Instance Connect' tab is selected. Below it, the instance ID is listed as 'i-0840692f9c6280704 (test2)'. The public IP address is '54.89.146.129'. The user name is set to 'ec2-user'. A note states: 'In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' At the bottom are 'Cancel' and 'Connect' buttons.

Feedback Looking for language selection? Find it in the new Unified Settings

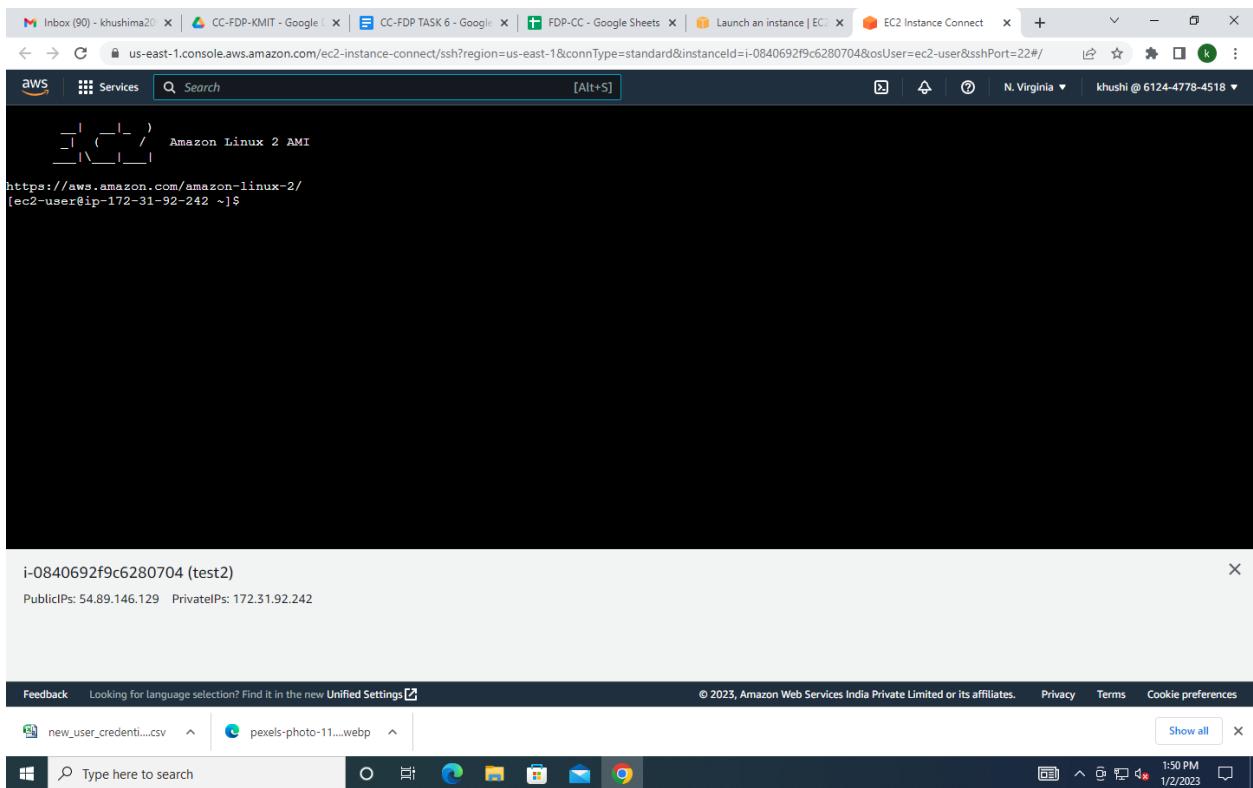
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

new\_user\_credential...csv pexels-photo-11...webp

Type here to search

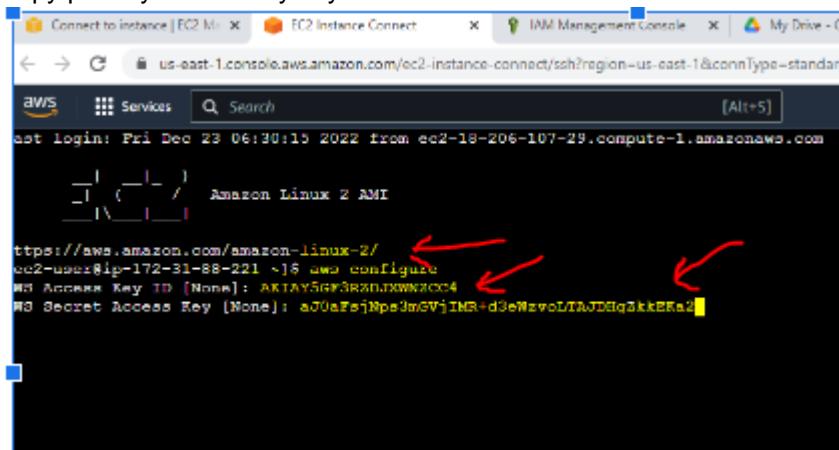
Establishing Connection ...

i-0840692f9c6280704 (test2)  
PublicIPs: 54.89.146.129 PrivateIPs: 172.31.92.242



Step9: once connected to the instance terminal

- Type aws configure
  - Copy paste your ID
  - Copy paste your security key

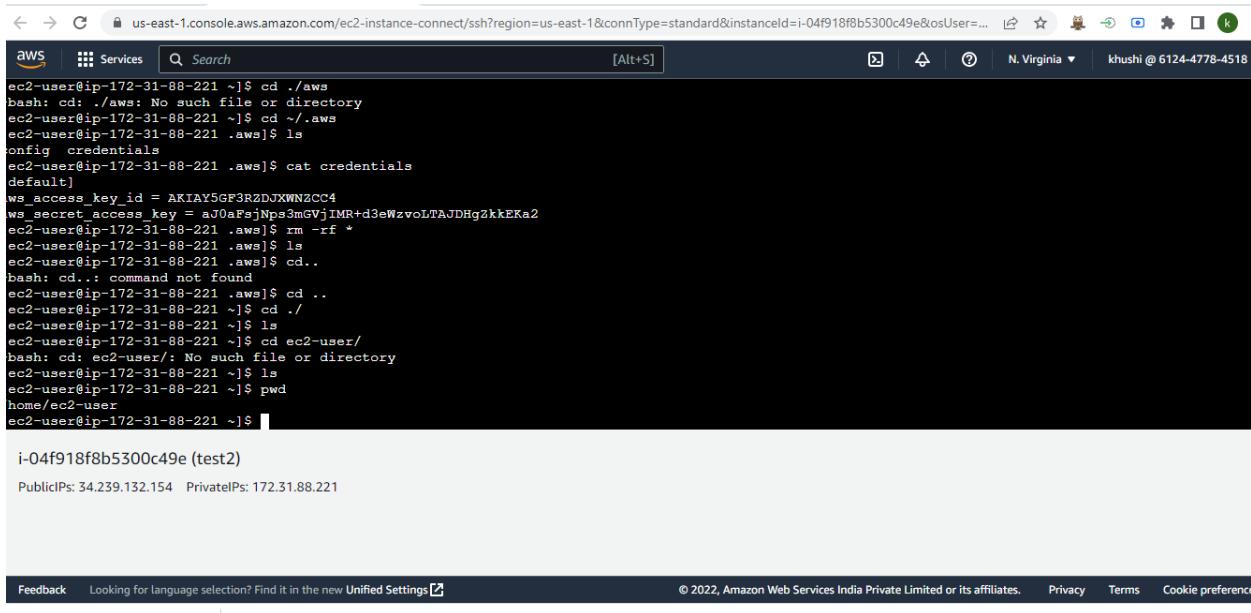


- Region: at default
  - Output format: at default

Step 10: type the following command

- Aws s3 ls
  - cd ~/.aws
  - Ls
  - Cat credentials
  - Rm -rf

- Ls
- Cd ..
- Cd..
- Ls
- Cd ec2-user/
- Ls
- Pwd
- Aws s3 ls



```

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-04f918f8b5300c49e&osUser=...
khushi @ 6124-4778-4518
AWS Services Search [Alt+S]
N. Virginia

ec2-user@ip-172-31-88-221 ~]$ cd ./aws
bash: cd: ./aws: No such file or directory
ec2-user@ip-172-31-88-221 ~]$ cd ~/.aws
ec2-user@ip-172-31-88-221 .aws]$ ls
config credentials
ec2-user@ip-172-31-88-221 .aws]$ cat credentials
default]
aws_access_key_id = AKIAY5GF3RZDJXWNZCC4
aws_secret_access_key = aJ0aFsjNps3mGVj1MR+d3eWzvoLT AJDHgZkkERa2
ec2-user@ip-172-31-88-221 .aws]$ rm -rf *
ec2-user@ip-172-31-88-221 .aws]$ ls
ec2-user@ip-172-31-88-221 .aws]$ cd ..
bash: cd.: command not found
ec2-user@ip-172-31-88-221 .aws]$ cd ..
ec2-user@ip-172-31-88-221 ~]$ ls
ec2-user@ip-172-31-88-221 ~]$ cd ec2-user/
bash: cd: ec2-user/: No such file or directory
ec2-user@ip-172-31-88-221 ~]$ ls
ec2-user@ip-172-31-88-221 ~]$ pwd
/home/ec2-user
ec2-user@ip-172-31-88-221 ~]$ 

i-04f918f8b5300c49e (test2)
PublicIPs: 34.239.132.154 PrivateIPs: 172.31.88.221

```

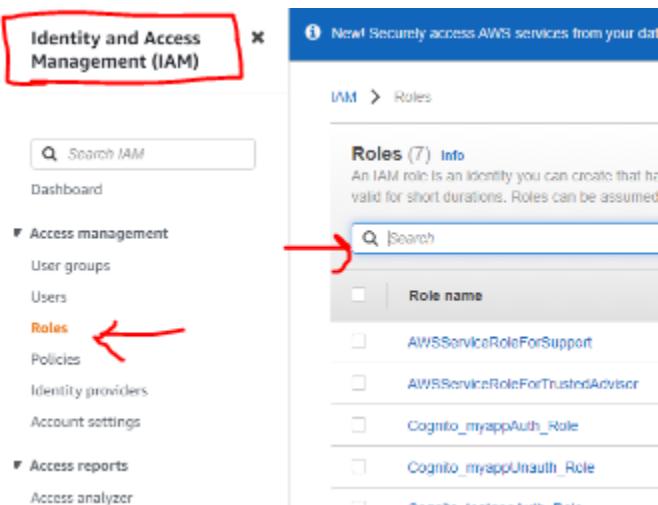
Feedback Looking for language selection? Find it in the new Unified Settings [\[?\]](#)

© 2022, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Show all

## Step 11: go to IAM services

- Roles create role



The screenshot shows the AWS IAM service interface. On the left, there's a navigation sidebar with links like Dashboard, User groups, Users, Roles (which is highlighted with a red arrow), Policies, Identity providers, and Account settings. The main area is titled "Roles (7) Info" and contains a search bar labeled "Search". Below the search bar is a table listing seven roles: AWSServiceRoleForSupport, AWSServiceRoleForTrustedAdvisor, Cognito\_myappAuth\_Role, Cognito\_myappUnauth\_Role, and others. A red arrow points to the search bar.

| Role name                             |
|---------------------------------------|
| AWSServiceRoleForSupport              |
| AWSServiceRoleForTrustedAdvisor       |
| Cognito_myappAuth_Role                |
| Cognito_myappUnauth_Role              |
| AmazonrekognitionFullAccess           |
| AmazonrekognitionTranscribe           |
| AmazonrekognitionTranscribeFullAccess |

- AWS services
- Select ec2

- Click on next

**Select trusted entity**

**Trusted entity type**

- AWS service  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy  
Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Common use cases**

- EC2  
Allows EC2 instances to call AWS services on your behalf.

- Policy search S3
- Select amazon S3 readonly access
- Click on next

**Permissions policies (Selected 1/801)**

Choose one or more policies to attach to your new role.

| Policy name                                                | Type        | Description                                                               |
|------------------------------------------------------------|-------------|---------------------------------------------------------------------------|
| AmazonDMSRedshiftFullAccess                                | AWS managed | Provides access to manage S3 settings for Redshift endpoints for DMS.     |
| AmazonS3FullAccess                                         | AWS managed | Provides full access to all buckets via the AWS Management Console.       |
| QuickSightAccessForRedshift                                | AWS managed | Policy used by QuickSight team to access customer data produced by S3 ... |
| <input checked="" type="checkbox"/> AmazonS3ReadOnlyAccess | AWS managed | Provides read only access to all buckets via the AWS Management Console.  |

**AmazonS3ReadOnlyAccess**

Provides read only access to all buckets via the AWS Management Console.

```

1. {
2.   "Version": "2012-10-17",
3.   "Statement": [
4.     {
5.       "Effect": "Allow",
6.       "Action": [
7.         "s3:Get*",

```

- You will get the role details

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

## Name, review, and create

### Role details

**Role name**  
Enter a meaningful name to identify this role.

s3ReadOnly

Maximum 64 characters. Use alphanumeric and '+-=\_,@-' characters.

**Description**  
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+-=\_,@-' characters.

### Step 1: Select trusted entities

Edit

1 [ ]  
2 "Version": "2012-10-17"

The screenshot shows the AWS IAM Roles page. A green banner at the top says 'Role s3ReadOnly created.' with a red arrow pointing to it. Below the banner, there's a 'View role' button. The main area shows a table with one row for the newly created role. The table has columns for 'Actions', 'Delete', and 'Create role'. The 'Actions' column for the role contains a red arrow pointing to the 'Modify IAM role' link.

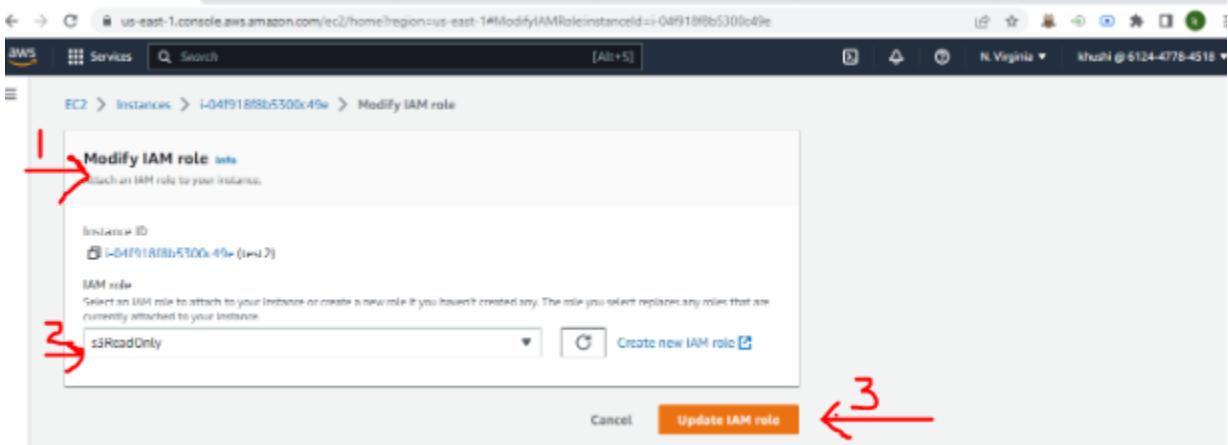
## Step 12: go to your instance

- select actions
- Security

The screenshot shows the AWS EC2 Instances page. A table lists one instance named 'test2'. A red arrow labeled '2' points to the instance name. Another red arrow labeled '3' points to the 'Actions' dropdown menu. Inside the menu, a red arrow labeled '4' points to the 'Modify IAM role' option. A red arrow labeled '5' points to the 'Modify IAM role' link in the instance details panel. A red arrow labeled '6' points to the 'Security' tab in the instance details panel.

- Modify IAM role

- Change to S3 read only



Step 13: after updating the IAM role got to terminal and type **aws s3 ls**

```
[ec2-user@ip-172-31-88-221 ~]$ aws s3 ls
Unable to locate credentials. You can configure credentials by running "aws configure".
[ec2-user@ip-172-31-88-221 ~]$ aws s3 ls
2022-12-21 08:08:09 ccfdpstest
[ec2-user@ip-172-31-88-221 ~]$ 
```

i-04f918f8b5300c49e (test2)

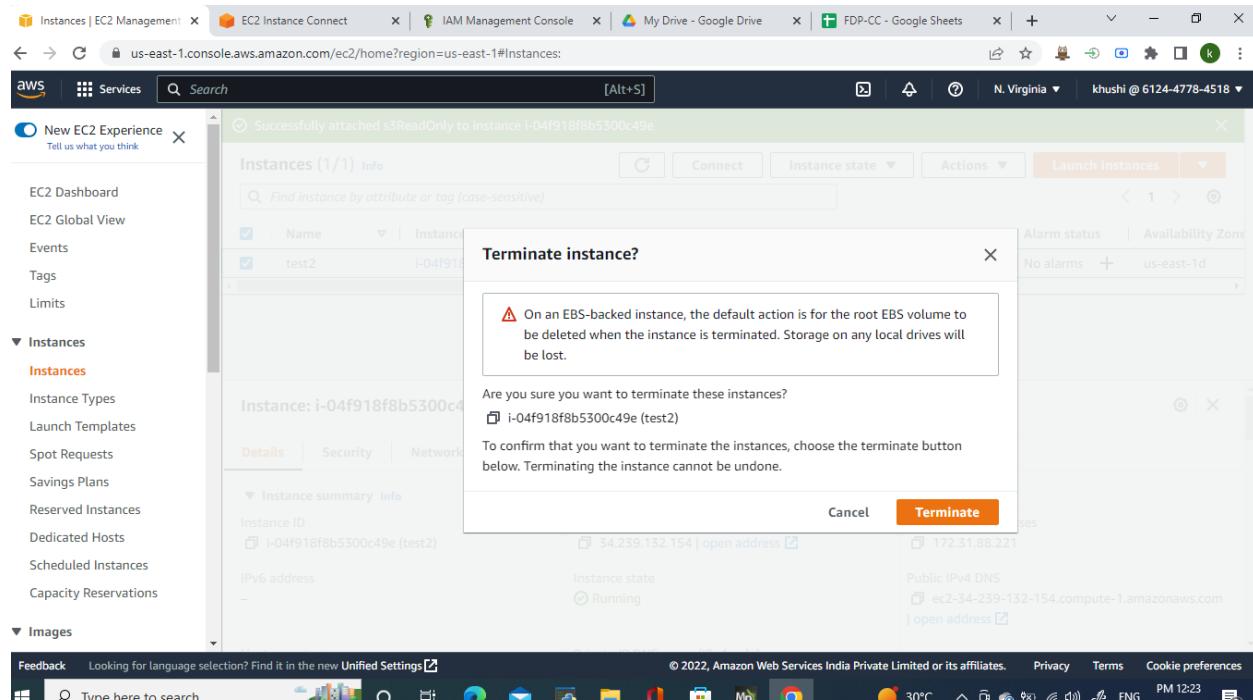
PublicIPs: 34.239.132.154 PrivateIPs: 172.31.88.221

https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1 Settings

© 2022, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

30°C PM 12:21 23-12-2022

Step 14: terminate instance



## EXPERIMENT 8

### Implement a role based access between AWS services

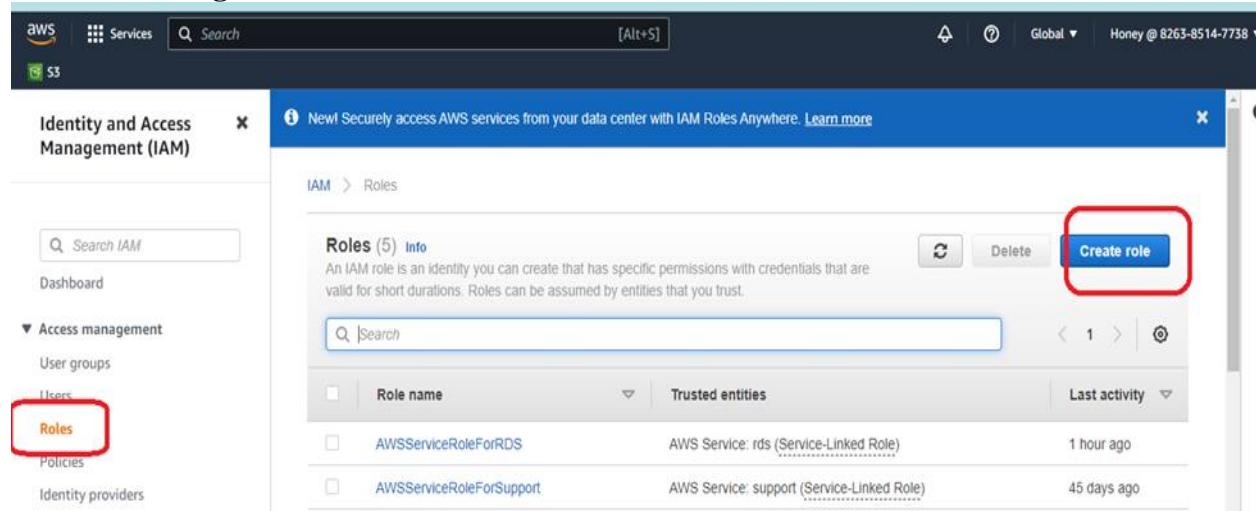
**AIM:** To Create IAM Role to access S3 bucket as read only mode from EC2 instance.

**REQUIREMENTS:** AWS active account and desktop.

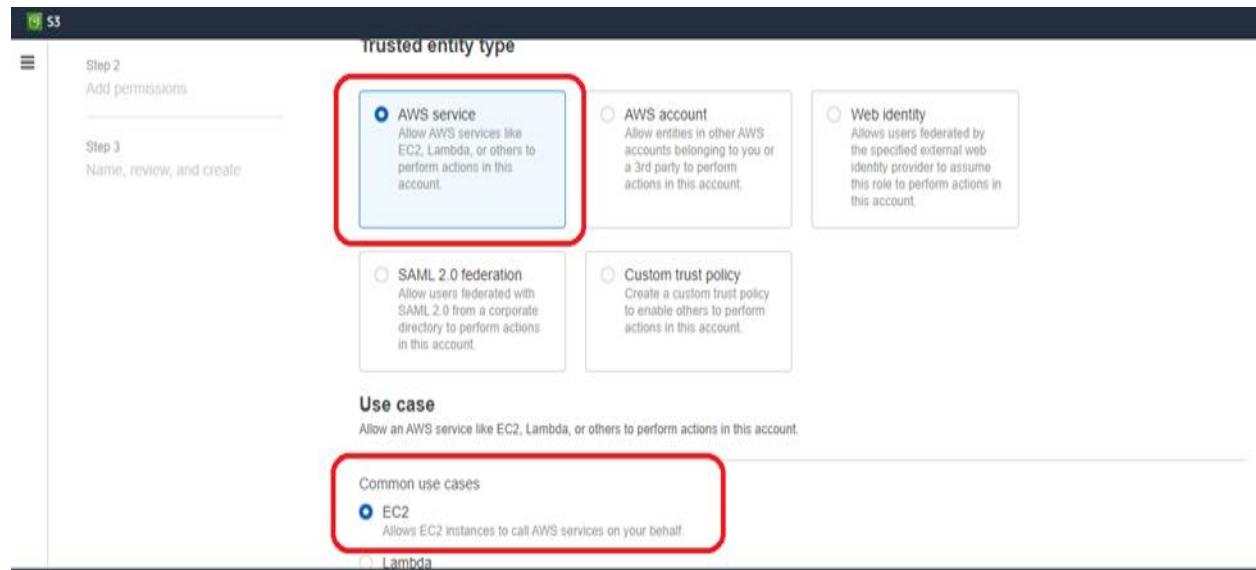
#### **PROCEDURE:**

##### **Step 1: Assigning Roles**

**Note:** Assuming EC2 & S3 Instance are created



The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' selected. Under 'Access management', 'Roles' is highlighted with a red box. The main area shows a table of existing roles: 'AWSServiceRoleForRDS' (AWS Service: rds (Service-Linked Role)) and 'AWSServiceRoleForSupport' (AWS Service: support (Service-Linked Role)). A blue 'Create role' button is at the top right of the table area, also highlighted with a red box.

The screenshot shows the 'Add permissions' step in the IAM Role creation wizard. It has three steps: Step 2 (selected), Step 3, and Step 4. In Step 2, under 'Trusted entity type', the 'AWS service' option is selected and highlighted with a red box. Other options include 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. In the 'Use case' section, the 'EC2' option is selected and highlighted with a red box. Other options include 'Lambda'.

## Step 2: Adding Permissions: S3 read only

The screenshot shows the AWS IAM Management Console. A modal window titled "Add permissions" is open, showing a list of policies. The "AmazonS3ReadOnlyAccess" policy is selected and highlighted with a blue border. Below the list, the policy's details are shown: "Provides read only access to all buckets via the AWS Management Console". A "Copy" button is available. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/iamv2/home#/roles/create?commonUseCase=EC2&step=addPermission&trustedEntityType=AWS_SERVICE`. The status bar at the bottom right indicates the date and time: 12/20/2022 12:36 AM.

## Step 3: Successfully S3 Role created with read-only permission

The screenshot shows the AWS IAM Management Console with a success message: "Role S3readonlynew created." A "View role" button is visible. The left sidebar shows the "Identity and Access Management (IAM)" navigation pane with "Access management" expanded, showing "Roles" as the selected item. The main content area displays a list of roles with columns for "Role name", "Trusted entities", and "Last activity". Three roles are listed: "AWSServiceRoleForRDS", "AWSServiceRoleForSupport", and "AWSServiceRoleForTrustedAdvisor". The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/iamv2/home#/roles`. The status bar at the bottom right indicates the date and time: 12/20/2022 12:43 AM.

## Step 4: EC2 Role based Access

The screenshot shows the AWS EC2 Instances page. A red box highlights the 'Modify IAM role' button in the bottom right corner of the instance details panel. Another red box highlights the 'Security' option in the Actions dropdown menu, which also includes 'Change security groups', 'Get Windows password', 'Image and templates', and 'Monitor and troubleshoot'.

The screenshot shows the 'Modify IAM role' dialog box. A red box highlights the 'Update IAM role' button at the bottom right. Inside the dialog, there is a warning message: '⚠ If you choose No IAM Role, any IAM role that is currently attached to the instance will be removed. Are you sure you want to remove from the selected instance?' Below the dialog are 'Cancel' and 'Update IAM role' buttons.

## Step 5: check role based Access on terminal



```
Last login: Thu Jan  5 18:26:41 2023 from ec2-18-206-107-28.compute-1.amazonaws.com
>Last login: Thu Jan  5 18:26:41 2023 from ec2-18-206-107-28.compute-1.amazonaws.com
[ec2-user@ip-172-31-89-176 ~]$ aws s3 ls
2022-12-18 08:38:45 ccbkmit
2022-11-01 10:29:19 cherrybucket23
2022-12-18 07:39:41 cherrycherry
2022-12-22 16:51:11 cmigrationbucket
2022-11-01 10:19:52 honeybucket1
[ec2-user@ip-172-31-89-176 ~]$
```

## EXPERIMENT 9

### Migrate a website from local server to Cloud

#### Migrate your on-premises workload walkthrough

In this walkthrough, the example WordPress blog is currently running as a two-tier stack in a simulated on-premises data center. The frontend uses Apache Server running on Ubuntu 16.04 LTS and the backend is supported by a mySQL server running on Ubuntu 16.04 LTS. All systems are hosted on a virtualized platform in a simulated data center.

The key steps of this migration process are:

- Create IAM user for AWS Replication Agent
- Create the Replication Settings template in the AWS MGN Console
- Install the AWS Replication Agents on source servers
- Configure the Launch Settings in the AWS MGN console
- Launch the test instances
- Launch the cutover instances
- Finalize cutover

#### Prerequisites

For this walkthrough, complete the following prerequisites:

- Have an [AWS account](#)
- Have a clear understanding of [Amazon Virtual Private Cloud \(Amazon VPC\)](#)
- Create the virtual networking environment for the purpose of this walkthrough as shown in Figure 1:
  - Create Amazon VPC, *MGN-Demo VPC*.
  - Create three subnets in the *MGN-Demo VPC*, *Staging Area Subnet*, *Migrated Resources Public Subnet* and *Migrated Resource Private Subnet*
  - Create Internet gateway and attach it to the *MGN-Demo VPC*
  - Create two Route tables, *Public-MGN-Demo-RouteTable* and *Private-MGN-Demo-RouteTable*
  - Add internet route to the *Public-MGN-Demo-RouteTable* (Destination 0.0.0.0/0 with Internet gateway as target)
  - Associate the *Staging Area Subnet* and *Migrated Resources Public Subnet* with the *Public-MGN-Demo-RouteTable* And associate the *Migrated resource private subnet* with the *Private-MGN-Demo-RouteTable*
  - Launch an [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) instance as a Bastion host in the *Migrated Resources Public Subnet*
  - Create two security group, *Public-MGN-Demo-SG* and *Private-MGN-Demo-SG*
    - Add inbound rules for HTTP and HTTPS ports from anywhere (0.0.0.0/0), and SSH port from the Bastion host public IP address to the *Public-MGN-Demo-SG*.
    - Add inbound rules for MYSQL ports from *Public-MGN-Demo-SG* security group to the *Private-MGN-Demo-SG* and SSH port from the Bastion host private IP address.

- AWS MGN requires network connectivity to and from your on-premises environment and within your Amazon VPC. For more information on the detailed network connectivity requirement, [check the Network Requirement Guide](#).

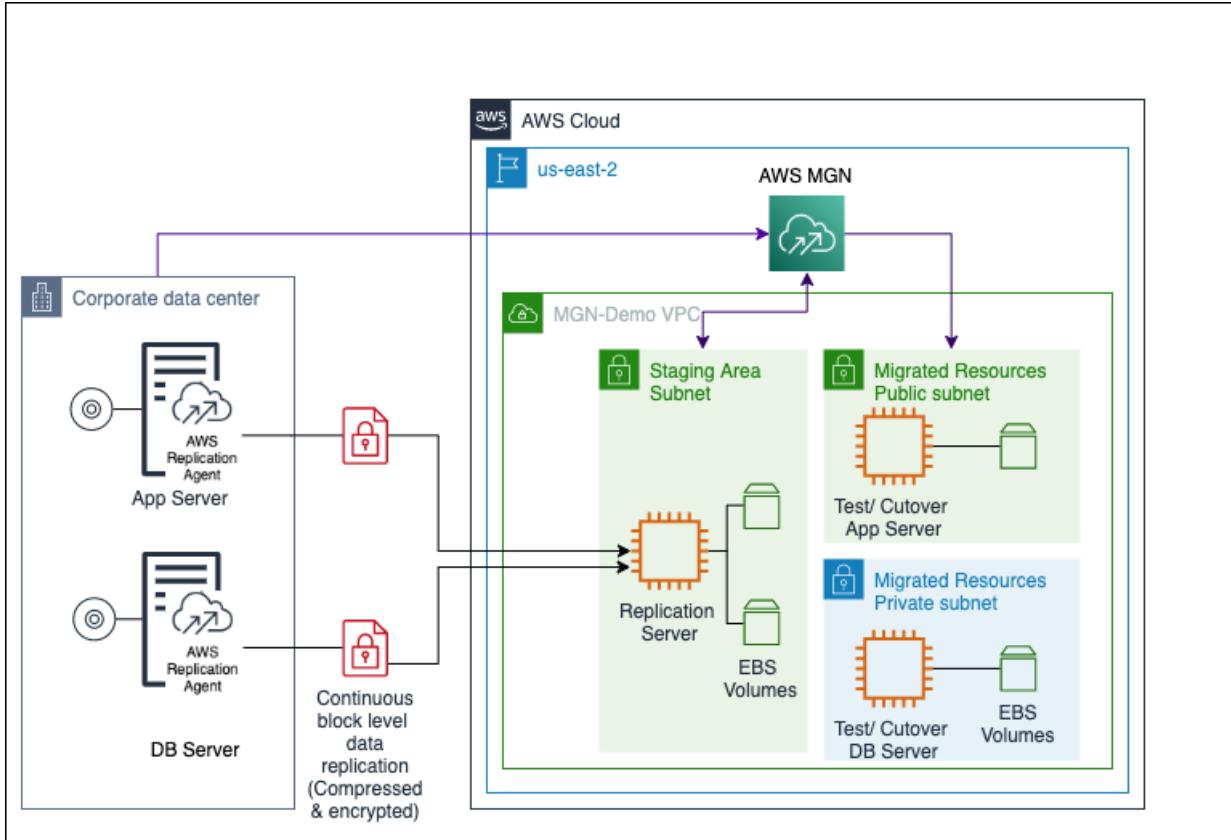


Figure 1 – AWS MGN migration architectural diagram.

#### Create IAM user for AWS Replication Agent

You need AWS credentials to use AWS MGN. Create an AWS Identity and Access Management (IAM) user and assign the proper permission policy to the user. Obtain an Access Key ID and Secret Access Key, which you would need to enter during the AWS replication agent installation on the source servers. Follow the steps found in [generating the required AWS credentials](#).

## Add user

1 2 3 4 5

### ▼ Set permissions

Filter policies ▾  Showing 9 results

|                                     | Policy name ▾                                 | Type        | Used as                |
|-------------------------------------|-----------------------------------------------|-------------|------------------------|
| <input type="checkbox"/>            | AWSApplicationDiscoveryAgentAccess            | AWS managed | None                   |
| <input type="checkbox"/>            | AWSApplicationDiscoveryServiceFullAccess      | AWS managed | None                   |
| <input checked="" type="checkbox"/> | AWSApplicationMigrationAgentPolicy            | AWS managed | None                   |
| <input type="checkbox"/>            | AWSApplicationMigrationConversionServerPolicy | AWS managed | Permissions policy (1) |
| <input type="checkbox"/>            | AWSApplicationMigrationEC2Access              | AWS managed | None                   |
| <input type="checkbox"/>            | AWSApplicationMigrationFullAccess             | AWS managed | None                   |
| <input type="checkbox"/>            | AWSApplicationMigrationMGHAccess              | AWS managed | Permissions policy (1) |
| <input type="checkbox"/>            | AWSApplicationMigrationReadOnlyAccess         | AWS managed | None                   |

[Cancel](#) [Previous](#) [Next: Tags](#)

Figure 2 – Creating an IAM User and attaching policy.

### Create the replication settings template

To migrate your servers from on-premises or another cloud service provider, create your replication settings to initialize the service for the first time in the AWS MGN console. You can access AWS MGN through the [AWS console](#).

**Application Migration Service**

Source Servers  
Launch history  
Settings

Migration Hub   
Documentation

Application Migration Service > Set up Application Migration Service

## Set up Application Migration Service

In order to use Application Migration Service in this region, the service must first be initialized by creating a Replication Settings template. After the template is created, Application Migration Service will automatically create the IAM roles required for the service to operate. The service can only be initialized by the Admin user of your AWS account, and must have multi-factor authentication configured.

### Create Replication Settings template

Every source server added to this console has Replication Settings that control how data is sent from the source server to AWS. These settings are created automatically based on this template, and can be modified at any time for any source server or group of source servers. The template itself can also be modified at any time (changes made will only affect newly added servers).

#### Replication Servers

Staging area subnet

Staging Area Subnet  
vpc-1

Replication Server instance type

t3.small

EBS volume type (for replicating disks over 500GB)

Faster, General Purpose SSD (gp2)

EBS encryption

Default

Security groups

Always use Application Migration Service security group

Additional security groups

Choose additional security groups

Data routing and throttling

Use private IP for data replication (VPN, DirectConnect, VPC peering)  
 Create public IP  
 Throttle network bandwidth (per server - in Mbps)

Figure 3 – Creating the MGN replication setting template.

## Install the AWS replication agents

After AWS MGN has been initialized with the creation of the replication settings template, and you have generated the required AWS credentials with the necessary permission, now install the AWS replication agent on the source servers on-premises. There are separate installation instructions for Linux and Windows operating systems:

- [Linux](#)
- [Windows](#)

For the illustration, both source servers are running on Ubuntu 16.04 LTS. The Linux installation steps are as follows:

1. Download the agent installer.

```
wget -O ./aws-replication-installer-init.py https://aws-application-migration-service-us-east-2.s3.amazonaws.com/latest/linux/aws-replication-installer-init.py
```

Remember to replace the region with the AWS Region into which you are replicating; in this walkthrough the replication is set to us-east-2.

2. Once you download the agent installer, run the installation script.

```
sudo python3 aws-replication-installer-init.py
```

3. The installer prompts you to enter your AWS Region Name, the AWS Access Key ID, and AWS Secret Access Key. Use the Access Key and Secret Access Key that you saved earlier.

4. Once you have entered the credentials, the installer identifies the volumes for replication, and prompts you to choose the disks to replicate, press enter.

```
ubuntu@ip-192-168-0-169:~$ sudo python3 aws-replication-installer-init.py
The installation of the AWS Replication Agent has started.
AWS Region Name: us-east-2
AWS Access Key ID: AK          'PM
AWS Secret Access Key:
Identifying volumes for replication.
Choose the disks you want to replicate. Your disks are: /dev/xvda
To replicate some of the disks, type the path of the disks, separated with a comma
Press Enter:
Identified volume for replication: /dev/xvda of size 8 GiB
All volumes for replication were successfully identified.
Downloading the AWS Replication Agent onto the source server... Finished.
Installing the AWS Replication Agent onto the source server... Finished.
Syncing the source server with the Application Migration Service Console... Finished.
The following is the source server ID: s-49f15a470a77d6f74.
The AWS Replication Agent was successfully installed.
ubuntu@ip-192-168-0-169:~$
```

Figure 4 – MGN replication agent installation.

Repeat the same AWS replication agent installation process on all source servers.

After installing the AWS replication agent on the source servers, return to the AWS MGN console to validate the source servers in the console. Please note that the source servers are added to the AWS MGN console by installing the AWS replication agent on the source servers.

| Source Servers (2)                                              |                   |        |                     |
|-----------------------------------------------------------------|-------------------|--------|---------------------|
| <input type="text"/> Filter source servers by property or value |                   |        |                     |
|                                                                 | Hostname          | Alerts | Migration lifecycle |
| <input type="checkbox"/>                                        | ip-192-168-0-169  | -      | Not ready           |
| <input type="checkbox"/>                                        | ip-192-168-10-156 | -      | Not ready           |

Figure 5 – Validating agent installation in MGN console.

After the initial sync is complete, the migration lifecycle displays Ready for testing and the data replication status displays as Healthy.

| Source Servers (2)                                              |                   |        |                     |
|-----------------------------------------------------------------|-------------------|--------|---------------------|
| <input type="text"/> Filter source servers by property or value |                   |        |                     |
|                                                                 | Hostname          | Alerts | Migration lifecycle |
| <input type="checkbox"/>                                        | ip-192-168-0-169  | -      | Ready for testing   |
| <input type="checkbox"/>                                        | ip-192-168-10-156 | -      | Ready for testing   |

Figure 6 – Source servers ready for testing in MGN console.

#### Configure the launch settings

Prior to testing or cutting over an instance, configure the launch settings. The launch settings are a set of instructions that determine how a test or cutover instance is launched for each source server in AWS.

Launch settings are comprised of two sections: general launch settings and Amazon EC2 launch templates.

Select each server, and select the tab for Launch Settings to edit the launch settings.

The screenshot shows the Migration dashboard interface. At the top, there are tabs: Migration dashboard, Server Info, Tags, Disks settings, Replication Settings, and Launch settings. The Launch settings tab is currently selected, indicated by an orange underline. Below this, there are two main sections: 'General launch settings' and 'EC2 Launch Template'.

**General launch settings** (Info) Edit

|                            |     |                            |      |
|----------------------------|-----|----------------------------|------|
| Instance type right sizing | On  | Operating system licensing | BYOL |
| Start instance upon launch | Yes | Transfer server tags       | No   |
| Copy private IP            | No  |                            |      |

**EC2 Launch Template** (Info) Modify

|                 |                                                   |                           |    |
|-----------------|---------------------------------------------------|---------------------------|----|
| Template ID     | lt-1                                              | Primary network interface |    |
| Instance type   | c4.xlarge                                         | Description               | *  |
| EBS volumes     | Volume 1 (8 GiB, EBS, Provisioned IOPS SSD (io1)) | Subnet                    | *  |
| Security groups | *                                                 | Public IP                 | No |
| Tenancy         | *                                                 | Private IP addresses      | *  |

Figure 7 – Configuring launch settings for each source server in MGN console.

For this walkthrough, we will keep the default settings for the general launch settings. And, we will be making changes to the Amazon EC2 launch template, to place the App Server and DB Server in the right subnets, which you have created with the necessary security groups.

Amazon EC2 launch template modification for the App Server (source server):  
Starting with the App Server, select modify in the Amazon EC2 Launch template window.

This would launch a new window for the Amazon EC2 Launch templates. Add a description for the Template version description.

## Modify template (Create new version)

Modifying a template allows you to create a new template version from an existing version. Using versions allows you to manage templates in a structured and controlled way. It also allows you to always use the default version of the template by rolling out updates to templates without having to change a reference to the template name or ID.

**Launch template name and version description**

Launch template name  
created-and-used-by-application-migration-service-s-

Template version description  
 Max 255 chars

Auto Scaling guidance [Info](#)  
Select this if you intend to use this template with EC2 Auto Scaling  
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Source template

### Launch template contents

Specify the details of your launch template version below. Leaving a field blank will result in the field not being included in the launch template version.

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

AMI

▼ Instance type [Info](#)

Instance type  
  
Family: c4 4 vCPU 7.5 GiB Memory  
On-Demand Linux pricing: 0.199 USD per Hour  
On-Demand Windows pricing: 0.383 USD per Hour

Compare instance types

Figure 8 – Modifying Amazon EC2 launch template.

Navigate down to Network Interfaces. Under Subnet, choose the respective subnet, *Migrated Resources Public Subnet* for the app server, the *Public-MGN-Demo-SG* for Security groups, and select Enable under Auto-assign public IP for the public facing app server. Once done, select Create template version.

▼ Network interfaces [Info](#)

**Network interface 1**

|                                                   |                                                                                                                                                  |                                                                                                                                                                      |                        |
|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| Device index <a href="#">Info</a><br>0            | Network interface <a href="#">Info</a><br>New interface                                                                                          | Description <a href="#">Info</a><br>My Primary ENI                                                                                                                   | <a href="#">Remove</a> |
| Subnet <a href="#">Info</a><br>subnet-1           | Security groups <a href="#">Info</a><br><a href="#">Select security groups ▾</a>                                                                 | Auto-assign public IP <a href="#">Info</a><br>Enable                                                                                                                 |                        |
| IP addresses available: 251                       |                                                                                                                                                  | <input checked="" type="checkbox"/> Hide all selected                                                                                                                |                        |
| Primary IP <a href="#">Info</a><br>123.123.123.1  | Secondary IP <a href="#">Info</a><br>Don't include in launch temp...                                                                             | IPv6 IPs <a href="#">Info</a><br>Don't include in launch temp...                                                                                                     |                        |
| Delete on termination <a href="#">Info</a><br>Yes | Elastic Fabric Adapter <a href="#">Info</a><br><input checked="" type="checkbox"/> Enable<br>EFA is only compatible with certain instance types. | Network card index <a href="#">Info</a><br><a href="#">Don't include in launch template ▾</a><br>The selected instance type does not support multiple network cards. |                        |
| <a href="#">Add network interface</a>             |                                                                                                                                                  |                                                                                                                                                                      |                        |

► Advanced details [Info](#)

[Cancel](#) [Create template version](#)

Figure 9 – Modifying Amazon EC2 launch template network interfaces for Source Server migration.

This launches a new window that shows you have successfully modified the EC2 Launch Template. Now, you would go ahead to change the default version that the EC2 launch template would use, select View Launch Template.



Success

Successfully modified [created-and-used-by-application-migration-service-s](#)

. A new version (version 5) has been created.

► Actions log

## Next steps

### Launch an instance

With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

[Launch instance from this template](#)

### Create an Auto Scaling group from your template

Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

[Create Auto Scaling group](#)

### Create Spot Fleet

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot Instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

[Create Spot Fleet](#)

[View launch templates](#)

Figure 10 – The successfully modified Amazon EC2 Launch template window.

In the Launch template window that shows up, select the launch template ID, and select Actions in the top right corner, and then select the Set default version.

*Hint: If you have more than one launch template ID, and you do not know the right one, return to the MGN console, select the right Source Server, under its Launch Settings, you would find the right Launch template ID.*

The screenshot shows the AWS EC2 Launch Templates page. At the top left, it says "EC2 > Launch templates". Below that, the title "Launch templates (1/2)" is followed by an "Info" link. A search bar with the placeholder "Filter by tags or properties or search by keyword" is present. The main table has columns for "Launch template ID", "Launch template name", and "Default". One row is selected, showing "lt-1" as the ID and "created-and-used-by-application-migration-service-s-" as the name. The number "5" is at the end of the row. The table also includes a header row with arrows for sorting.

| Launch template ID | Launch template name                                 | Default |
|--------------------|------------------------------------------------------|---------|
| lt-1               | created-and-used-by-application-migration-service-s- | 5       |

Figure 11 – Updating and setting a new default launch template version.

The Set default version dialog box appears; select the Template version you just created and set as default version.

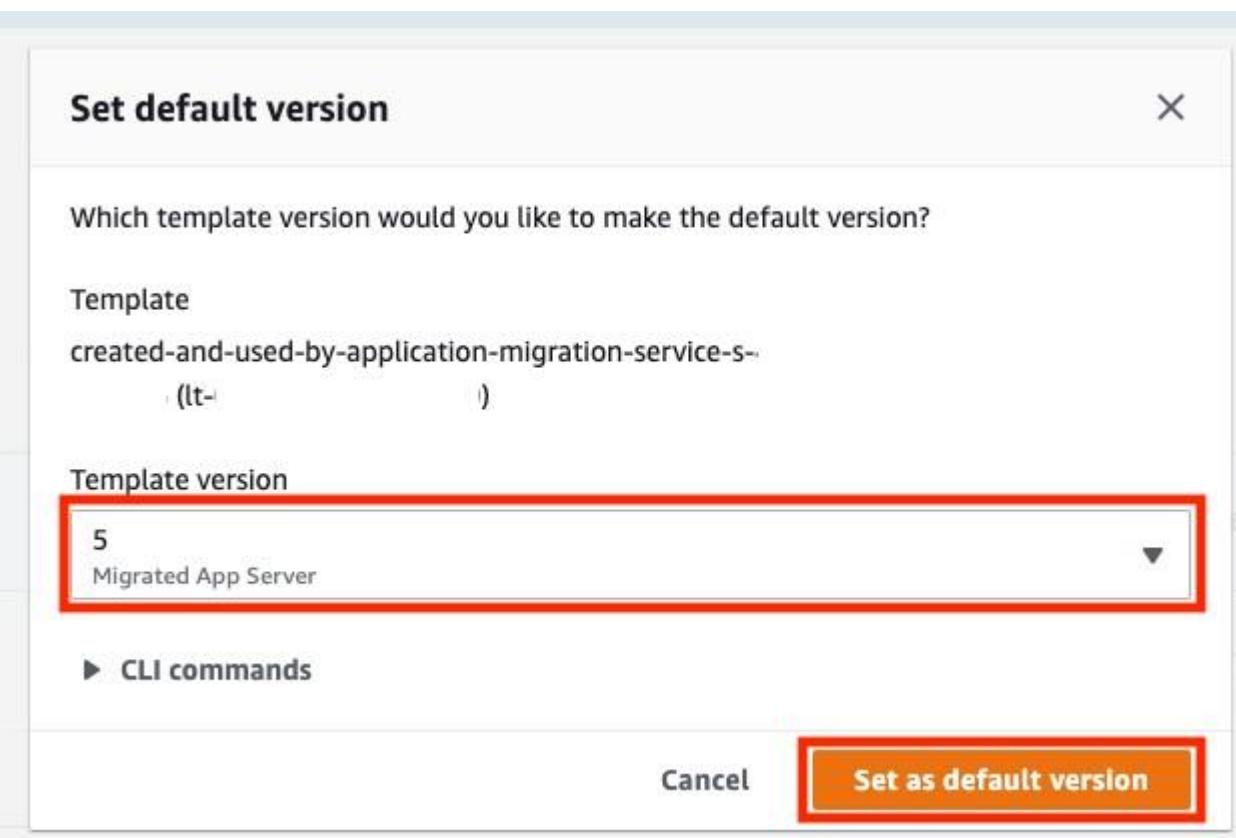


Figure 12 – Setting a new default launch template version.

Return to the AWS MGN console and select the source server for the app server. Under launch settings, find the updated Amazon EC2 Launch Settings.

| General launch settings <small>Info</small>       |                            |
|---------------------------------------------------|----------------------------|
| Instance type right sizing                        | Operating system licensing |
| On                                                | BYOL                       |
| Start instance upon launch                        | Transfer server tags       |
| Yes                                               | No                         |
| Copy private IP                                   |                            |
| No                                                |                            |
| EC2 Launch Template <small>Info</small>           |                            |
| Template ID                                       | Primary network interface  |
| lt-086c                                           | Description                |
| Instance type                                     | -                          |
| c4.xlarge                                         | Subnet                     |
| EBS volumes                                       | subnet-04a1                |
| Volume 1 (8 GiB, EBS, Provisioned IOPS SSD (io1)) | Public IP                  |
| Security groups                                   | Yes                        |
| sg-0f3                                            | Private IP addresses       |
| Tenancy                                           | -                          |
| -                                                 |                            |
| Placement group name                              | -                          |
| -                                                 |                            |

Figure 13 – Updated Amazon EC2 launch template in MGN console.

Amazon EC2 launch template modification for the DB Server (source server):  
Repeat steps 1-9 to configure the launch settings for the DB Server.

1. Return to the MGN console and select the source server for the DB Server
2. Select the Launch Settings tab
3. Select Modify under the EC2 Launch Template
4. Add a description for the Template Version description
5. Navigate to Network Interfaces:
  - a. Under Subnet, choose the respective subnet, *Migrated resource private subnet* for the DB Server,
  - b. The *Private-MGN-Demo-SG* for Security groups
  - c. And leave the auto-assign public IP as disabled for the DB Server in a private subnet.
  - d. Once done, select Create template version.
6. On the successfully modified Amazon EC2 Launch Template page, select View Launch Template.
7. In the Launch template window, select the launch template ID for the DB Server, and select Actions in the top right corner, and then click on the Set default version.
8. The Set default version window dialog box appears. Select the Template version you just created and set as default version.
9. Return to the AWS MGN console. Select the source server for the DB Server; under launch settings find the updated EC2 Launch Settings.

#### Launch the test instances

Now, that you configured the launch settings for each source server, you are ready to launch the servers as test instances. Best practice is to test instances before cutover.

On the MGN console page, make sure that your source servers are ready for testing by looking at the following indicators as shown in Figure 14. The source servers are Healthy and Ready for testing.

Application Migration Service > Source Servers

### Source Servers (2)

Filter source servers by property or value

| Hostname          | Alerts | Migration lifecycle | Data replication status | Next step             |
|-------------------|--------|---------------------|-------------------------|-----------------------|
| ip-192-168-0-169  | -      | Ready for testing   | Healthy                 | Launch test instance. |
| ip-192-168-10-156 | -      | Ready for testing   | Healthy                 | Launch test instance. |

Figure 14 – Source servers in healthy and ready for testing state.

Check the boxes to the left of each server that you want to launch a Test Instance. Select Test and Cutover in the top right corner, then select Launch test instances.

Application Migration Service > Source Servers

### Source Servers (2)

Filter source servers by property or value

| Hostname                                              | Alerts | Migration lifecycle | Data replication status | Next step             |
|-------------------------------------------------------|--------|---------------------|-------------------------|-----------------------|
| <input checked="" type="checkbox"/> ip-192-168-0-169  | -      | Ready for testing   | Healthy                 | Launch test instance. |
| <input checked="" type="checkbox"/> ip-192-168-10-156 | -      | Ready for testing   | Healthy                 | Launch test instance. |

Actions ▾ Replication ▾ Test and Cutover ▾

- Testing
- Launch test instances
- Mark as "Ready for cutover"
- Revert to "Ready for testing"
- Cutover
- Launch cutover instances
- Finalize cutover
- Revert to "Ready for cutover"
- Other
- Edit Launch Settings
- Terminate launched instances

Figure 15 – Launching a test instance in MGN console.

Select Launch in the Launch test instances dialog box.

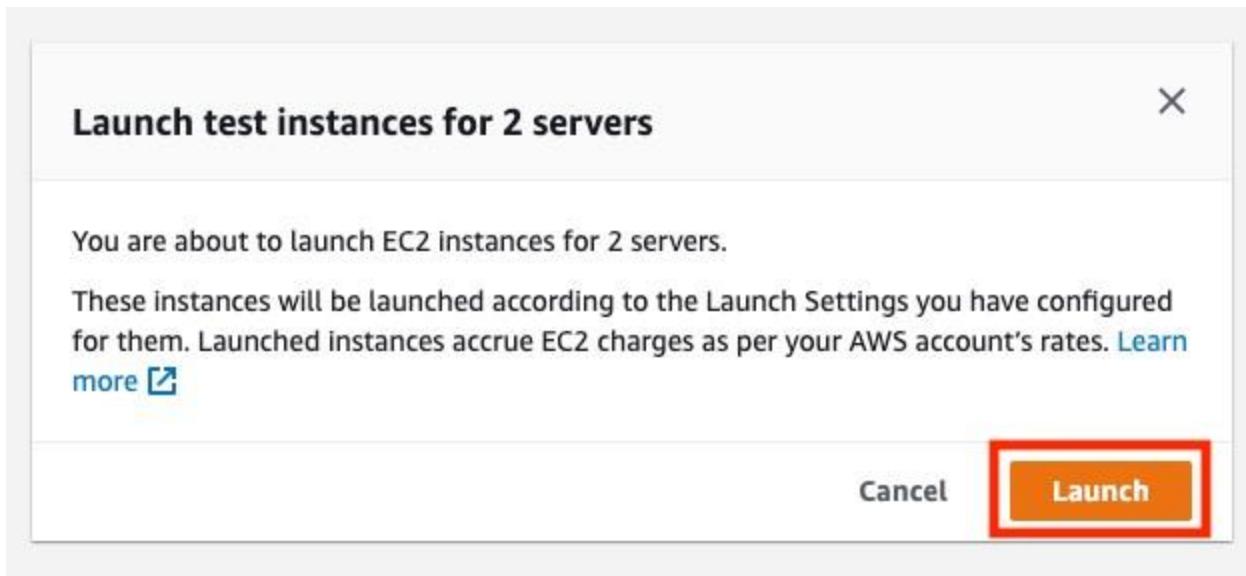


Figure 16 – Launch test instance window.

This takes you back to the MGN console page with the source servers. Make sure that the Migration lifecycle indicates that Test in progress, then select View job details in the top right corner to view the MGN job log details.

A screenshot of the Application Migration Service console showing the "Source Servers" list. At the top, a green header bar displays "Launch job mgnjob-41 created" and "Starting to launch test Instances for 2 servers." On the right, there is a "View job details" button with a red circle around it. Below the header, the "Source Servers (2)" section shows two entries. Each entry has a checkbox, a hostname, an alerts column, a "Migration lifecycle" column with a red circle containing the number "1" and a red box around the "Test in progress" status, an "Actions" dropdown, a "Replication" dropdown, and a "Test and Cutover" dropdown. The first server is listed as "ip-192-168-0-169" with "Test in progress" and "Healthy" status. The second server is listed as "ip-192-168-10-156" with "Test in progress" and "Healthy" status. A navigation bar at the bottom left shows "Application Migration Service > Source Servers".

| Hostname          | Alerts | Migration lifecycle | Data replication status | Next step                                         |
|-------------------|--------|---------------------|-------------------------|---------------------------------------------------|
| ip-192-168-0-169  | -      | Test in progress    | Healthy                 | Complete testing and mark as 'Ready for cutover'. |
| ip-192-168-10-156 | -      | Test in progress    | Healthy                 | Complete testing and mark as 'Ready for cutover'. |

Figure 17 – MGN console migration lifecycle status.

Check the job status as it changes from Started to Completed. You can also view the job logs for more details.

⌚ Launch job mgnjob-4128061418b35949 created  
Starting to launch test instances for 2 servers.

[View job](#)

Application Migration Service > Launch history > Job: mgnjob-412

## Job: mgnjob-412

| Details               |                |                       |
|-----------------------|----------------|-----------------------|
| Type                  | Status         | Initiated by          |
| Launch                | Started        | Launch test instances |
| Start time            | Completed time |                       |
| 4/19/2021, 1:25:09 PM | -              |                       |

### Job log Info

Q Filter job log by property or value.

| Time                  | Event                    | Additional Data                   |
|-----------------------|--------------------------|-----------------------------------|
| 4/19/2021, 1:25:09 PM | Job started              |                                   |
| 4/19/2021, 1:25:09 PM | Started taking snapshot  | Source server : ip-192-168-0-169  |
| 4/19/2021, 1:25:09 PM | Started taking snapshot  | Source server : ip-192-168-10-156 |
| 4/19/2021, 1:26:00 PM | Finished taking snapshot | Source server : ip-192-168-0-169  |
| 4/19/2021, 1:26:00 PM | Finished taking snapshot | Source server : ip-192-168-10-156 |
| 4/19/2021, 1:26:01 PM | Conversion started       | Source server : ip-192-168-0-169  |
| 4/19/2021, 1:26:01 PM | Conversion started       | Source server : ip-192-168-10-156 |

Figure 18 – MGN console job log.

Once completed, return to the MGN console Source Server page. Now the Alerts show up as Launched.

| Hostname          | Alerts   | Migration lifecycle | Data replication status | Next step                                         |
|-------------------|----------|---------------------|-------------------------|---------------------------------------------------|
| ip-192-168-0-169  | Launched | Test in progress    | Healthy                 | Complete testing and mark as 'Ready for cutover'. |
| ip-192-168-10-156 | Launched | Test in progress    | Healthy                 | Complete testing and mark as 'Ready for cutover'. |

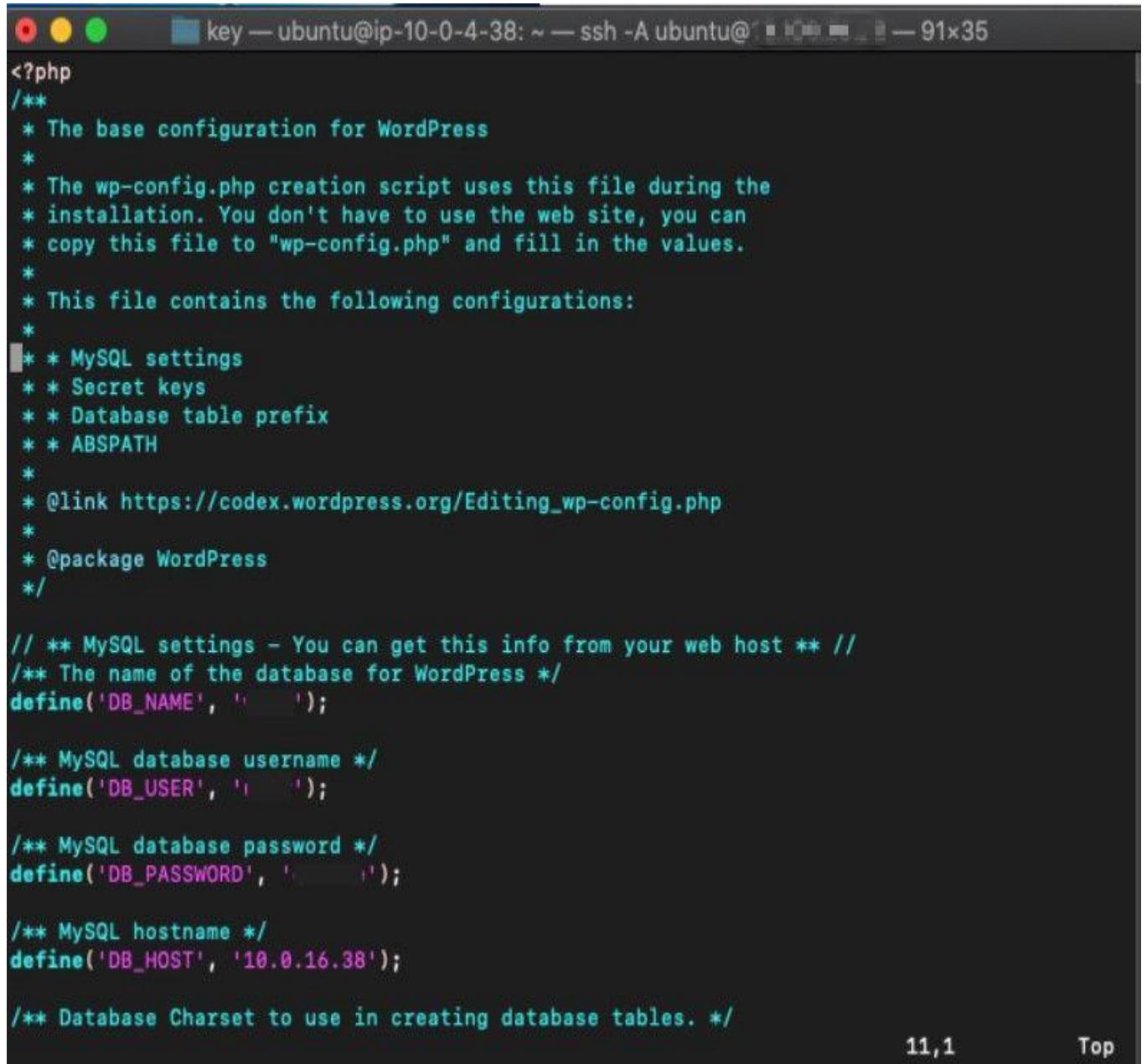
Figure 19 – MGN console launched server alert.

Making configuration updates to the WordPress App Server with the new DB Server host:

- On the MGN console, select the test DB source server to access the Migration Dashboard.
- Under Lifecycle, select View in Amazon EC2 console to see the Amazon EC2 page.
- Obtain the private IP address for the DB Server; you will be updating the WordPress app server configuration file with this information.
- Return to the MGN console, select the test app source server to access the Migration Dashboard.
- Under Lifecycle, select View in Amazon EC2 console to see the Amazon EC2 page.
- Obtain the public IP address.
- Login to the test app server for WordPress, and update the database connection string:

```
sudo vi /var/www/html/wp-config.php
```

- In the wp-config.php file, update the test DB Server IP address.



```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', ' ');

/** MySQL database username */
define('DB_USER', ' ');

/** MySQL database password */
define('DB_PASSWORD', ' ');

/** MySQL hostname */
define('DB_HOST', '10.0.16.38');

/** Database Charset to use in creating database tables. */

```

11,1

Top

Figure 20 – Changing the database host in wp-config.php.

Updating grant privileges in mySQL database:

Update your grant privileges in mySQL database for the test app server to assess the test database.

- Login to the database using the root password:

```
sudo mysql -u root -p
```

- Grant all privilege on the MySQL database for WordPress user. Replace *DB\_NAME* with database name, *DB\_HOST* with database IP address, and *DB\_PASSWORD* with the password for the database.

```
GRANT ALL PRIVILEGES ON '<DB NAME>'@'<DB HOST>' IDENTIFIED BY 'DB_PASSWORD';
```

- Flush privileges to reload the grant tables.

```
FLUSH PRIVILEGES;  
exit
```

```
key — ubuntu@ip-10-0-16-38: ~ — ssh -A ubuntu@ip-10-0-16-38 — 91x35
ubuntu@ip-10-0-16-38:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 5.7.33-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

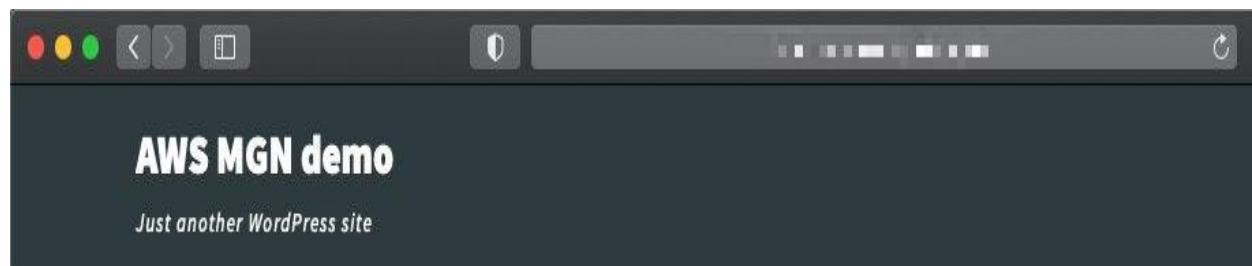
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> GRANT ALL PRIVILEGES ON    *.* TO 'user'@'10.0.4.38' IDENTIFIED BY 'c      e';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> █
```

*Figure 21 – Updating grant privileges in database.*

Launch the app in a web browser using the public IP address of the test app server.



*Figure 22 – Launched test server webpage in browser.*

With a successfully launched application in the web browser, you can select Mark as “Ready for Cutover”. Cutover.

## Source Servers (2)

| Source Servers (2)                                              |          |                     |
|-----------------------------------------------------------------|----------|---------------------|
| <input type="text"/> Filter source servers by property or value |          |                     |
| Hostname                                                        | Alerts   | Migration lifecycle |
| <input checked="" type="checkbox"/> ip-192-168-0-169            | Launched | Test in progress    |
| <input checked="" type="checkbox"/> ip-192-168-10-156           | Launched | Test in progress    |

Figure 23 – How to select servers and mark them as “Ready for cutover:” check the boxes next to the server hostname; select the “Test and Cutover” dropdown menu, and then from the dropdown, select “Mark as ‘Ready for Cutover.’”

When the “Mark servers as ready for cutover” dialog box appears, choose continue. The source servers show up as ready for cutover in the MGN console.

### Launch the cutover instances

After completing the testing of all the source servers, you are ready for cutover. Prior to launching cutover instances, ensure that the source servers are listed as Ready for cutover under Migration lifecycle and Healthy under Data replication status.

Once confirmed, check all source servers to be cutover, and then select Launch Cutover instances from the Test and Cutover dropdown menu.

The screenshot shows the 'Source Servers (2)' page in the Application Migration Service. At the top, there are three buttons: 'Actions ▾', 'Replication ▾', and 'Test and Cutover ▾'. The 'Test and Cutover ▾' button is highlighted with a red box and circled with number 3. A context menu is open from this button, listing several options: 'Testing', 'Launch test instances', 'Mark as "Ready for cutover"', 'Revert to "Ready for testing"', 'Cutover', 'Launch cutover instances' (which is highlighted with a red box and circled with number 4), 'Finalize cutover', 'Revert to "Ready for cutover"', 'Other', 'Edit Launch Settings', and 'Terminate launched instances'. The main table lists two source servers: 'ip-192-168-0-169' and 'ip-192-168-10-156', both in 'Ready for cutover' status with 'Healthy' replication. The 'Hostname' column has a red box around it and is circled with number 1. The 'Migration lifecycle' and 'Data replication status' columns have red boxes around them and are circled with number 2. The 'Next step' column contains 'Launch cutover instance' for each server.

| Hostname          | Alerts | Migration lifecycle | Data replication status | Next step               |
|-------------------|--------|---------------------|-------------------------|-------------------------|
| ip-192-168-0-169  | -      | Ready for cutover   | Healthy                 | Launch cutover instance |
| ip-192-168-10-156 | -      | Ready for cutover   | Healthy                 | Launch cutover instance |

Figure 24 – Launching cutover instances.

The Launch cutover instances dialog box appears. Choose Launch to start the cutover.

On the MGN console, the source servers show up as Cutover in progress. Select view job details in the top right corner of the page to view the job details.

⌚ Launch Job mgnjob-492eaeb60f3edb1d8 created  
Starting to launch cutover instances for 2 servers.

Application Migration Service > Source Servers

## Source Servers (2)

Filter source servers by property or value

Hostname



Alerts



Migration lifecycle

1

ip-192-168-0-169

-

Cutover in progress

ip-192-168-10-156

-

Cutover in progress

Figure 25 – Cutover in progress.

⌚ Launch job mgnjob-492i created  
Starting to launch cutover instances for 2 servers.

Application Migration Service > Launch history > Job: mgnjob-492e

### Job: mgnjob-492e

**Details**

| Type                   | Status         | Initiated by             |
|------------------------|----------------|--------------------------|
| Launch                 | Started        | Launch cutover instances |
| Start time             | Completed time | -                        |
| 4/20/2021, 11:15:25 AM | -              | -                        |

**Job log** Info

Filter job log by property or value

| Time                   | Event                   | Additional Data                   |
|------------------------|-------------------------|-----------------------------------|
| 4/20/2021, 11:15:26 AM | Job started             |                                   |
| 4/20/2021, 11:15:26 AM | Started taking snapshot | Source server : ip-192-168-0-169  |
| 4/20/2021, 11:15:26 AM | Started taking snapshot | Source server : ip-192-168-10-156 |

**Source servers (2)**

Filter source servers by property or value

| Hostname          | Status  |
|-------------------|---------|
| ip-192-168-0-169  | Pending |
| ip-192-168-10-156 | Pending |

Figure 26 – Job details of cutover.

Once the job is completed, return to the MGN console page. The Alerts show up as Launched.

Application Migration Service > Source Servers

## Source Servers (2)

Filter source servers by property or value

|                          | Hostname          | Alerts                | Migration lifecycle |
|--------------------------|-------------------|-----------------------|---------------------|
| <input type="checkbox"/> | ip-192-168-0-169  | <span>Launched</span> | Cutover in progress |
| <input type="checkbox"/> | ip-192-168-10-156 | <span>Launched</span> | Cutover in progress |

Figure 27 – Launched cutover instance.

Follow the same steps for ‘Making configuration updates to the WordPress App Server with the new DB Server host’ and “Updating grant privileges in mySQL database’.

Application Migration Service > Source Servers > ip-192-168-0-169

ip-192-168-0-169 (s-49f)

Actions ▾ Replication ▾ Test and Cutover ▾

Next actions [Info](#)

Launched

Finalize cutover.

Migration dashboard Server info Tags Disks settings Replication Settings Launch settings

Lifecycle [Info](#)

Not ready > Ready for testing > Test in progress > Ready for cutover > **Cutover in progress** > Cutover complete

Launch status Last test Cutover

Launched Job ID: [mgnjob-412](#) Cutover

First boot: Succeeded Job ID: [mgnjob-492](#)

[View in EC2 console](#) (3) Started: 4/19/2021, 1:25:09 PM Started: 4/20/2021, 11:15:26 AM

Data replication status [Info](#)

Healthy

Replication progress Total replicated storage Elapsed replication time

100% 8 of 8 GiB 5 d

Lag Last seen

- 4/20/2021, 4:24:00 PM

Backlog Replication start time

- 4/15/2021, 8:43:39 PM

Events and metrics [?](#)

The screenshot displays the AWS Application Migration Service migration dashboard for a specific source server. At the top, the navigation path is shown as Application Migration Service > Source Servers > ip-192-168-0-169. The main title is 'ip-192-168-0-169 (s-49f)'. Below the title are three dropdown menus: Actions ▾, Replication ▾, and Test and Cutover ▾. The 'Migration dashboard' tab is selected. The dashboard is divided into several sections: 'Next actions' (with a 'Launched' status and a 'Finalize cutover' button), 'Lifecycle' (showing a progress bar from 'Not ready' to 'Cutover complete', with a 'Cutover in progress' step highlighted in blue. A red circle highlights the 'View in EC2 console' link under the 'First boot: Succeeded' row), 'Data replication status' (showing 'Healthy' status with a 100% replication progress bar, 8 GiB total replicated storage, and a 5-day elapsed time. It also includes 'Lag' and 'Backlog' metrics with their respective last seen times and replication start times), and 'Events and metrics' (which is currently empty). The bottom of the dashboard features a horizontal navigation bar with links for Migration dashboard, Server info, Tags, Disks settings, Replication Settings, and Launch settings.

Figure 28 – The migration dashboard.

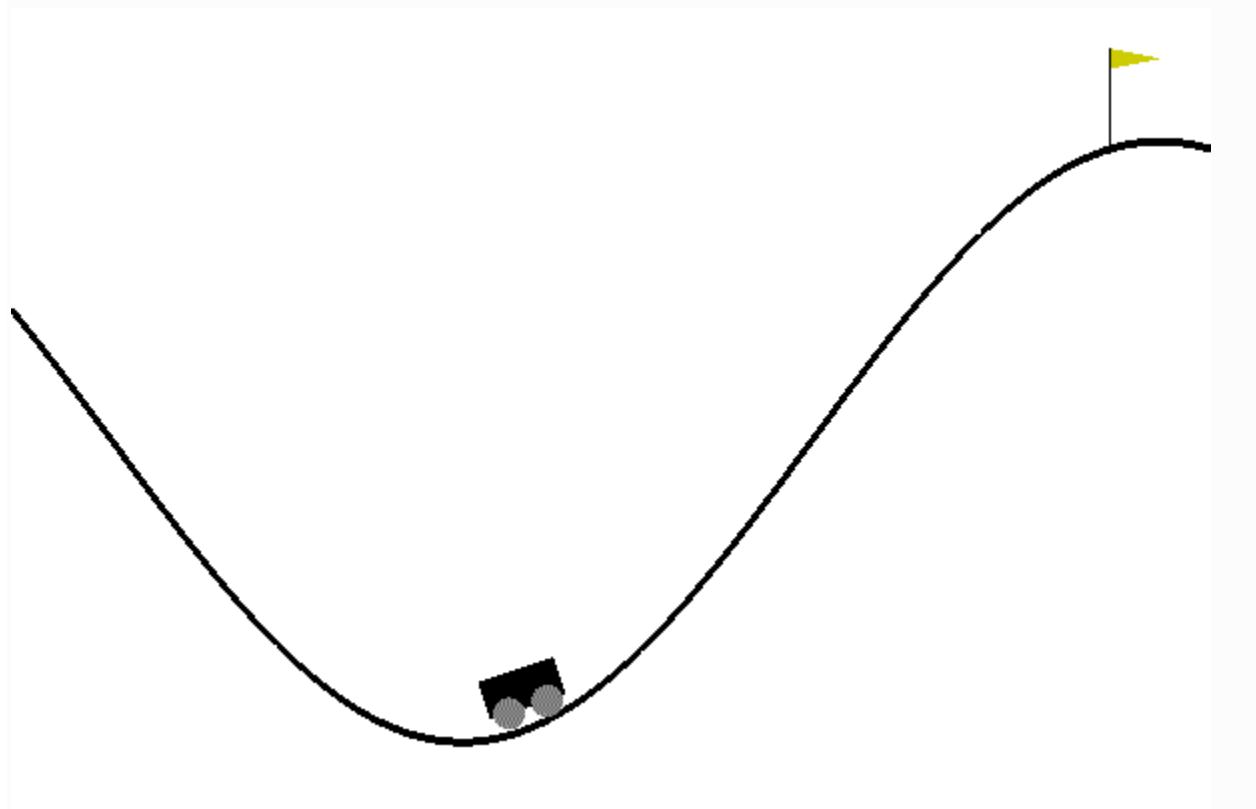
## EXPERIMENT 10

### Automate Machine Learning Model with AWS Amazon Sage maker

Introduction:

#### **Mountain Car with Amazon SageMaker RL**

Mountain Car is a classic control Reinforcement Learning problem that was first introduced by A. Moore in 1991 [1]. An under-powered car is tasked with climbing a steep mountain, and is only successful when it reaches the top. Luckily there's another mountain on the opposite side which can be used to gain momentum, and launch the car to the peak. It can be tricky to find this optimal solution due to the sparsity of the reward. Complex exploration strategies can be used to incentivise exploration of the mountain, but to keep things simple in this example we extend the amount of time in each episode from Open AI Gym's default of 200 environment steps to 10,000 steps, showing how to customise environments. We consider two variants in this example: [PatientMountainCar](#) for discrete actions and [PatientContinuousMountainCar](#) for continuous actions.



[PatientMountainCar](#) 3

1. Objective: Get the car to the top of the right hand side mountain.
2. Environment(s): Open AI Gym's [MountainCar-v0](#) that is extended to 10,000 steps per episode.

3. State: Car's horizontal position and velocity (can be negative).
4. Action: Direction of push (left, nothing or right).
5. Reward: -1 for every environment step until success, which incentivises quick solutions.

### PatientContinuousMountainCar

1. Objective: Get the car to the top of the right hand side mountain.
2. Environment(s): Open AI Gym's MountainCarContinuous-v0 that is extended to 10,000 steps per episode.
3. State: Car's horizontal position and velocity (can be negative).
4. Action: Magnitude of push (if negative push to left, if positive push to right).
5. Reward: +100 for reaching top of the right hand side mountain, minus the squared sum of actions from start to end.

[1] A. Moore, Efficient Memory-Based Learning for Robot Control, PhD thesis, University of Cambridge, November 1990.

## Pre-requisites

## Imports

To get started, we'll import the Python libraries we need, set up the environment with a few prerequisites for permissions and configurations.

```
[ ]:
import sagemaker
import boto3
import sys
import os
import glob
import re
import subprocess
import numpy as np
from IPython.display import HTML
import time
from time import gmtime, strftime

sys.path.append("common")
from misc import get_execution_role, wait_for_s3_object
from sagemaker.rl import RLEstimator, RLToolkit, RLFramework
```

## Setup S3 buckets

Set up the linkage and authentication to the S3 bucket that you want to use for checkpoint and the metadata.

```
[ ]:
sage_session = sagemaker.session.Session()
s3_bucket = sage_session.default_bucket()
s3_output_path = "s3://{}{}".format(s3_bucket)
print("S3 bucket path: {}".format(s3_output_path))
```

## Define Variables

We define variables such as the job prefix for the training jobs *and the image path for the container (only when this is BYOC)*.

```
[ ]:  
# create unique job name  
job_name_prefix = "rl-mountain-car"
```

## Configure settings

You can run your RL training jobs on a SageMaker notebook instance or on your own machine. In both of these scenarios, you can run in either ‘local’ (where you run the commands) or ‘SageMaker’ mode (on SageMaker training instances). ‘local’ mode uses the SageMaker Python SDK to run your code in Docker containers locally. It can speed up iterative testing and debugging while using the same familiar Python SDK interface. Just set `local_mode = True`. And when you’re ready move to ‘SageMaker’ mode to scale things up.

```
[ ]:  
# run in Local mode?  
local_mode = False
```

## Create an IAM role

Either get the execution role when running from a SageMaker notebook instance `role = sagemaker.get_execution_role()` or, when running from local notebook instance, use utils method `role = get_execution_role()` to create an execution role.

```
[ ]:  
try:  
    role = sagemaker.get_execution_role()  
except:  
    role = get_execution_role()  
  
print("Using IAM role arn: {}".format(role))
```

## Install docker for local mode

In order to work in `local` mode, you need to have docker installed. When running from you local machine, please make sure that you have docker or docker-compose (for local CPU machines) and nvidia-docker (for local GPU machines) installed. Alternatively, when running from a SageMaker notebook instance, you can simply run the following script to install dependenceis.

Note, you can only run a single local notebook at one time.

```
[ ]:  
# only run from SageMaker notebook instance  
if local_mode:  
    !/bin/bash ./common/setup.sh
```

## Setup the environments

We create a file called `src/patient_envs.py` for our modified environments. We can create a custom environment class or create a function that returns our environment. Since we’re using Open AI Gym environment and wrappers, we just create functions that take the classic control environments `MountainCarEnv` and `Continuous_MountainCarEnv` and wrap them with a `TimeLimit` where we specify the `max_episode_steps` to 10,000.

```
[ ]:  
!pygmentize src/patient_envs.py
```

## Configure the presets for RL algorithms

The presets that configure the RL training jobs are defined in the “preset-mountain-car-continuous-clipped-ppo.py” file which is also uploaded on the /src directory. Also see “preset-mountain-car-dqn.py” for the discrete environment case. Using the preset file, you can define agent parameters to select the specific agent algorithm. You can also set the environment parameters, define the schedule and visualization parameters, and define the graph manager. The schedule presets will define the number of heat up steps, periodic evaluation steps, training steps between evaluations.

These can be overridden at runtime by specifying the RLCOACH\_PRESET hyperparameter. Additionally, it can be used to define custom hyperparameters.

```
[ ]:  
!pygmentize src/preset-mountain-car-continuous-clipped-ppo.py
```

## Write the Training Codes

The training code is written in the file “train-coach.py” which is uploaded in the /src directory. We create a custom `SageMakerCoachPresetLauncher` which sets the default preset, maps and ties hyperparameters.

```
[ ]:  
!pygmentize src/train-coach.py
```

## Train the RL model using the Python SDK Script mode

If you are using local mode, the training will run on the notebook instance. When using SageMaker for training, you can select a GPU or CPU instance. The RLEstimator is used for training RL jobs.

1. Specify the source directory where the environment, presets and training code is uploaded.
2. Specify the entry point as the training code
3. Specify the choice of RL toolkit and framework. This automatically resolves to the ECR path for the RL Container.
4. Define the training parameters such as the instance count, job name, S3 path for output and job name.
5. Specify the hyperparameters for the RL agent algorithm. The RLCOACH\_PRESET can be used to specify the RL agent algorithm you want to use.
6. Define the metrics definitions that you are interested in capturing in your logs. These can also be visualized in CloudWatch and SageMaker Notebooks.

We use a variant of Proximal Policy Optimization (PPO) called Clipped PPO, which removes the need for complex KL divergence calculations.

```
[ ]:  
if local_mode:  
    instance_type = "local"  
else:  
    instance_type = "ml.m4.4xlarge"  
  
estimator = RLEstimator(  
    entry_point="train-coach.py",  
    source_dir="src",  
    dependencies=["common/sagemaker_rl"],  
    toolkit=RLToolkit.COACH,
```

```

        toolkit_version="0.11.0",
        framework=RLFramework.MXNET,
        role=role,
        instance_type=instance_type,
        instance_count=1,
        output_path=s3_output_path,
        base_job_name=job_name_prefix,
        hyperparameters={

            "RLCOACH_PRESET": "preset-mountain-car-continuous-clipped-ppo", # "preset-mountain-car-dqn",
            "discount": 0.995,
            "gae_lambda": 0.997,
            "evaluation_episodes": 3,
            # approx 100 episodes
            "improve_steps": 100000,
            # approx 5 episodes to start with
            "training_freq_env_steps": 75000,
            "training_learning_rate": 0.004,
            "training_batch_size": 256,
            # times number below by training_freq_env_steps to get total samples per policy training
            "training_epochs": 15,
            "save_model": 1,
        },
    )

estimator.fit(wait=local_mode)

```

## Store intermediate training output and model checkpoints

The output from the training job above is stored on S3. The intermediate folder contains gifs and metadata of the training.

```

[ ]:
job_name = estimator._current_job_name
print("Job name: {}".format(job_name))

s3_url = "s3://{}{}".format(s3_bucket, job_name)

if local_mode:
    output_tar_key = "{}/output.tar.gz".format(job_name)
else:
    output_tar_key = "{}/output/output.tar.gz".format(job_name)

intermediate_folder_key = "{}/output/intermediate/".format(job_name)
output_url = "s3://{}{}".format(s3_bucket, output_tar_key)
intermediate_url = "s3://{}{}".format(s3_bucket, intermediate_folder_key)

print("S3 job path: {}".format(s3_url))
print("Output.tar.gz location: {}".format(output_url))
print("Intermediate folder path: {}".format(intermediate_url))

tmp_dir = "/tmp/{}".format(job_name)
os.system("mkdir {}".format(tmp_dir))
print("Create local folder {}".format(tmp_dir))

```

## Visualizations

## Plot metrics for training job

We can pull the reward metric of the training and plot it to see the performance of the model over time.

```

[ ]:
%matplotlib inline
import pandas as pd

csv_file_name = "worker_0.simple_rl_graph.main_level.main_level.agent_0.csv"
key = os.path.join(intermediate_folder_key, csv_file_name)
wait_for_s3_object(s3_bucket, key, tmp_dir)

```

```

csv_file = "{}{}".format(tmp_dir, csv_file_name)
df = pd.read_csv(csv_file)
df = df.dropna(subset=["Training Reward"])
x_axis = "Episode #"
y_axis = "Training Reward"

if len(df) > 0:
    plt = df.plot(x=x_axis, y=y_axis, figsize=(12, 5), legend=True, style="b-")
    plt.set_ylabel(y_axis)
    plt.set_xlabel(x_axis)

```

## Visualize the rendered gifs

The latest gif file found in the gifs directory is displayed. You can replace the tmp.gif file below to visualize other files generated.

```

[ ]:
key = os.path.join(intermediate_folder_key, "gifs")
wait_for_s3_object(s3_bucket, key, tmp_dir)
print("Copied gifs files to {}".format(tmp_dir))
[ ]:
glob_pattern = os.path.join("{}/*.gif".format(tmp_dir))
gifs = [file for file in glob.iglob(glob_pattern, recursive=True)]
extract_episode = lambda string: int(
    re.search(".*episode-(\d*)_.*", string, re.IGNORECASE).group(1)
)
gifs.sort(key=extract_episode)
print("GIFs found:\n{}".format("\n".join([os.path.basename(gif) for gif in gifs])))
[ ]:
# visualize a specific episode
gif_index = -1 # since we want last gif
gif_filepath = gifs[gif_index]
gif_filename = os.path.basename(gif_filepath)
print("Selected GIF: {}".format(gif_filename))
os.system("mkdir -p ./src/tmp/ && cp {} ./src/tmp/{}.gif".format(gif_filepath, gif_filename))
HTML(''.format(gif_filename))

```

## Evaluation of RL models

We use the last checkpointed model to run evaluation for the RL Agent.

## Load checkpointed model

Checkpointed data from the previously trained models will be passed on for evaluation / inference in the checkpoint channel. In local mode, we can simply use the local directory, whereas in the SageMaker mode, it needs to be moved to S3 first.

```

[ ]:
wait_for_s3_object(s3_bucket, output_tar_key, tmp_dir)

if not os.path.isfile("{}output.tar.gz".format(tmp_dir)):
    raise FileNotFoundError("File output.tar.gz not found")
os.system("tar -xvzf {}output.tar.gz -C {}".format(tmp_dir, tmp_dir))

if local_mode:
    checkpoint_dir = "{}data/checkpoint".format(tmp_dir)
else:
    checkpoint_dir = "{}checkpoint".format(tmp_dir)

print("Checkpoint directory {}".format(checkpoint_dir))
[ ]:
if local_mode:
    checkpoint_path = "file://{}".format(checkpoint_dir)

```

```

    print("Local checkpoint file path: {}".format(checkpoint_path))
else:
    checkpoint_path = "s3://{}/{}/checkpoint/".format(s3_bucket, job_name)
    if not os.listdir(checkpoint_dir):
        raise FileNotFoundError("Checkpoint files not found under the path")
    os.system("aws s3 cp --recursive {} {}".format(checkpoint_dir, checkpoint_path))
    print("S3 checkpoint file path: {}".format(checkpoint_path))

```

## Run the evaluation steps

Use the checkpointer model to run the evaluation step.

```

[ ]:
estimator_eval = RLEstimator(
    role=role,
    source_dir="src/",
    dependencies=["common/sagemaker_rl"],
    toolkit=RLToolkit.COACH,
    toolkit_version="0.11.0",
    framework=RLFramework.MXNET,
    entry_point="evaluate-coach.py",
    instance_count=1,
    instance_type=instance_type,
    base_job_name=job_name_prefix + "-evaluation",
    hyperparameters={
        "RLCOACH_PRESET": "preset-mountain-car-continuous-clipped-ppo",
        "evaluate_steps": 10000 * 2, # evaluate on 2 episodes
    },
)

estimator_eval.fit({"checkpoint": checkpoint_path})

```

## Visualize the output

Optionally, you can run the steps defined earlier to visualize the output

## Model deployment

Since we specified MXNet when configuring the RLEstimator, the MXNet deployment container will be used for hosting.

```

[ ]:
predictor = estimator.deploy(
    initial_instance_count=1, instance_type=instance_type, entry_point="deploy-mxnet-coach.py"
)

```

We can test the endpoint with 2 samples observations. Starting with the car on the right side, but starting to fall back down the hill.

```

[ ]:
output = predictor.predict(np.array([0.5, -0.5]))
action = output[1][0]
action

```

We see the policy decides to move the car to the left (negative value). And similarly in the other direction.

```

[ ]:
output = predictor.predict(np.array([-0.5, 0.5]))
action = output[1][0]
action

```

## Clean up endpoints

```
[ ]:  
predictor.delete_endpoint()
```

The screenshot shows the Amazon EC2 Instances page. At the top, there's a search bar with 'search: i-0e8' and a 'Clear filters' button. To the right are three buttons: a blue 'C' button, a 'Connect' button, and a 'Instance state' dropdown set to 'Running'. Below the search bar is a table header with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, Elastic IP, and Private IP ad... The table contains one row for an instance named 'ip-192-168-0-169' with the ID 'i-0e8'. The instance is shown as 'Running' (green), of type 'c4.xlarge', with a status check of '2/2 checks passed', no alarms, in 'us-east-2b' availability zone, with a public DNS name 'ec2-13-59-65-140.us-e...', a public IP '13.59.65.140', and an elastic IP '10.0.4.7'.

Figure 29 – The Amazon EC2 console.

Once you're finished with the updates to the app and DB Servers, and connectivity is confirmed using a web browser to launch the application, and all other performance test has been performed, you can now make changes to the DNS records to point to the new app server.

