

# **DAY 1**

## **DATE:15.12.25(MONDAY)**

### **TOPICS**

Intro to cloud computing

Networking basics

Intro to devops

Tools used

### **INTRODUCTION TO CLOUD COMPUTING**

Cloud computing is the delivery of computing services—servers, storage, databases, networking, software, analytics—over the internet

Instead of keeping hardware locally, organizations use resources hosted on cloud platforms like AWS, Azure, GCP.

### **NETWORKING BASICS**

Networking refers to the process of connecting multiple devices (computers, servers, sensors) to share data and resources.

Examples: Sharing files, accessing the internet, communication between systems.

### **INTRODUCTION TO DEVOPS**

DevOps is a combination of Development + Operations that helps deliver software faster, efficiently, and with higher reliability.

### **Background**

DevOps evolved after two traditional methodologies:

1. Waterfall Model – linear, step-by-step
2. Agile Model – iterative, cyclic development
3. Code and configuration are stored in a distributed version-controlled system.
4. When code is shared for testing, any modifications done afterwards are saved as separate versions, without altering the original code.
5. This ensures traceability, reliability, and collaboration.

### **Examples of Platforms Using DevOps Concepts**

- AWS

- Azure
- GCP (Google Cloud Platform)

## TOOLS:

GitHub Actions

Docker

AWS / Cloud Platforms

Terraform – Cloud Infrastructure Automation

Grafana – Visualization & Monitoring

Git – Version Control System

GitHub – Code Hosting Platform

Kubernetes (K8s) – Container Orchestration

- EKS: Elastic Kubernetes Service (AWS Managed K8s)

## GIT ACTIONS:

GitHub Actions is an automation tool used for CI/CD (Continuous Integration / Continuous Deployment) inside GitHub.

### Key Points

- Uses YAML files to define workflows
- Automates tasks like build → test → deploy
- Triggered by GitHub events (push, pull request, etc.)

## DOCKER:

Docker is a containerization platform, NOT a storage database. Normally applications need OS, libraries, dependencies. Docker packages everything into a container so it can run anywhere consistently.

## **DAY 2**

**DATE:16.12.2025(TUESDAY)**

### **TOPICS**

**BASIC SERVICES OF AWS**

### **AWS**

Cloud Platform provided by Amazon  
Compared to Azure and gcp this is widely used.  
It works on the basis of pay-as-you-go  
It offers IT-Demand resources over the internet.

### **BASIC SERVICES**

VPC

EC2-works as a server

IAM-Security based

S3-Storage

RDS

#### **❖ Compute services**

- Server-for communication(client-intermediate-database)
- Web server-Installed in server for speed
- Serverless-without intermediate

So AWS provides virtual servers, containers, serverless computing and specialised computing sources(gpu)

#### **❖ Storage services**

- Unstructured data is stored (image,audio,video,folders)
- Can also do static host
- Examples
  - Amazon S3(independent-doesn't need any server)-The stored data are called objects.

Amazon EBS(Elastic Block Storage)-The data stored are called block storage.Used for EC2 instances. It requires a server.

#### **❖ Networking services**

- VPC(Virtual private cloud)
- Security groups(To allow which port can be accessed)

#### **❖ Database services**

RDS(Relational Database service)-sql, mysql,postgreSQL, oracle(structured data)

DynamoDB-No sql database for keyvalue and document data.

Aurora comes within RDS which is twice faster than RDS. It is expensive

### **AWS Global Infrastructure**

Physical backbone that provides all .AWS cloud services.Each region consists of multiple availability zones. For Example, In Mumbai it is split into 3 regions,ie

ap1a,ap1b,ap1c(asia- pacific south).So when the data enters a it gets copied in b and c too.  
So if there is any problem in any one of these regions the data will be retrieved.

Example: US East(N. Virginia)-> us-east-1 (First launched)  
Asia Pacific(MUMBAI)->ap-south-1

## DAY 3

**DATE:17.12.2025(WEDNESDAY)**

### VIRTUAL PRIVATE CLOUD(VPC)

Private space between client and server

Subnet has 2 types---Pvt and public

Pvt-internet gateway not connected

#### IP Address

IPV4(Numeric)--32 bits

IPV6(Alpha numeric)--64 bits

#### CIDR

Gives the range upto how many IPs can be connected to VPC

For Example:10.0.0.0/16-gives 65,536 IPs to use inside VPC

#### Subnets

Subdivisions under vpc

Public subnets-requires internet access (websites)

Private subnets-requires internet access (websites)

#### NAT Gateway

Lets pvt servers go out the internet but keeps them hidden from outsiders.

#### Route Tables

Used for connecting 2 or more servers

#### Security groups

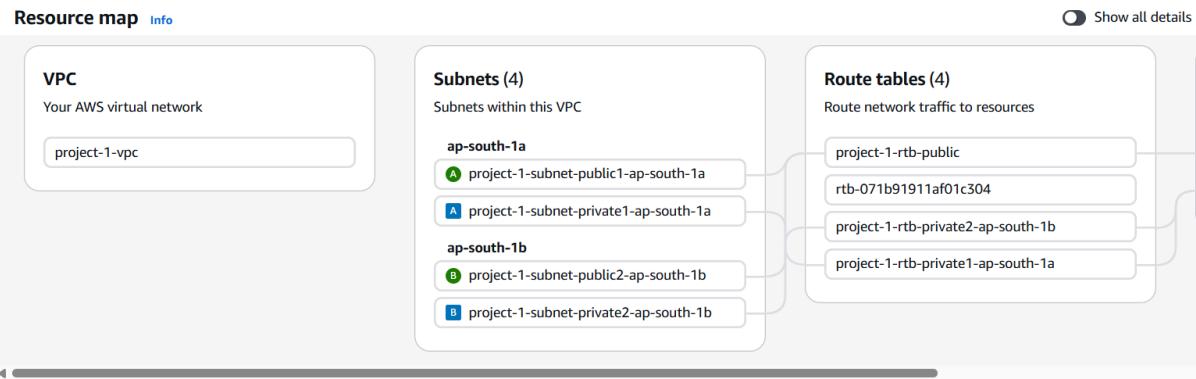
Port can be given in 2 ways

Inbuilt([https-443](https://443)(port no))

Customised-We create our own port

NACL-Acts as a Firewall for Subnets

vpc-0d2ce0b74d1876d82 / project-1-vpc				<a href="#">Actions ▾</a>
Details		Info		
VPC ID <a href="#">vpc-0d2ce0b74d1876d82</a>	State <a href="#">Available</a>	Block Public Access <input checked="" type="radio"/> Off	DNS hostnames Enabled	
DNS resolution Enabled	Tenancy default	DHCP option set <a href="#">dopt-01b3179112756fa73</a>	Main route table <a href="#">rtb-071b91911af01c304</a>	
Main network ACL <a href="#">acl-02505a30f179e83db</a>	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID <a href="#">347363494863</a>	
Encryption control ID -	Encryption control mode -			



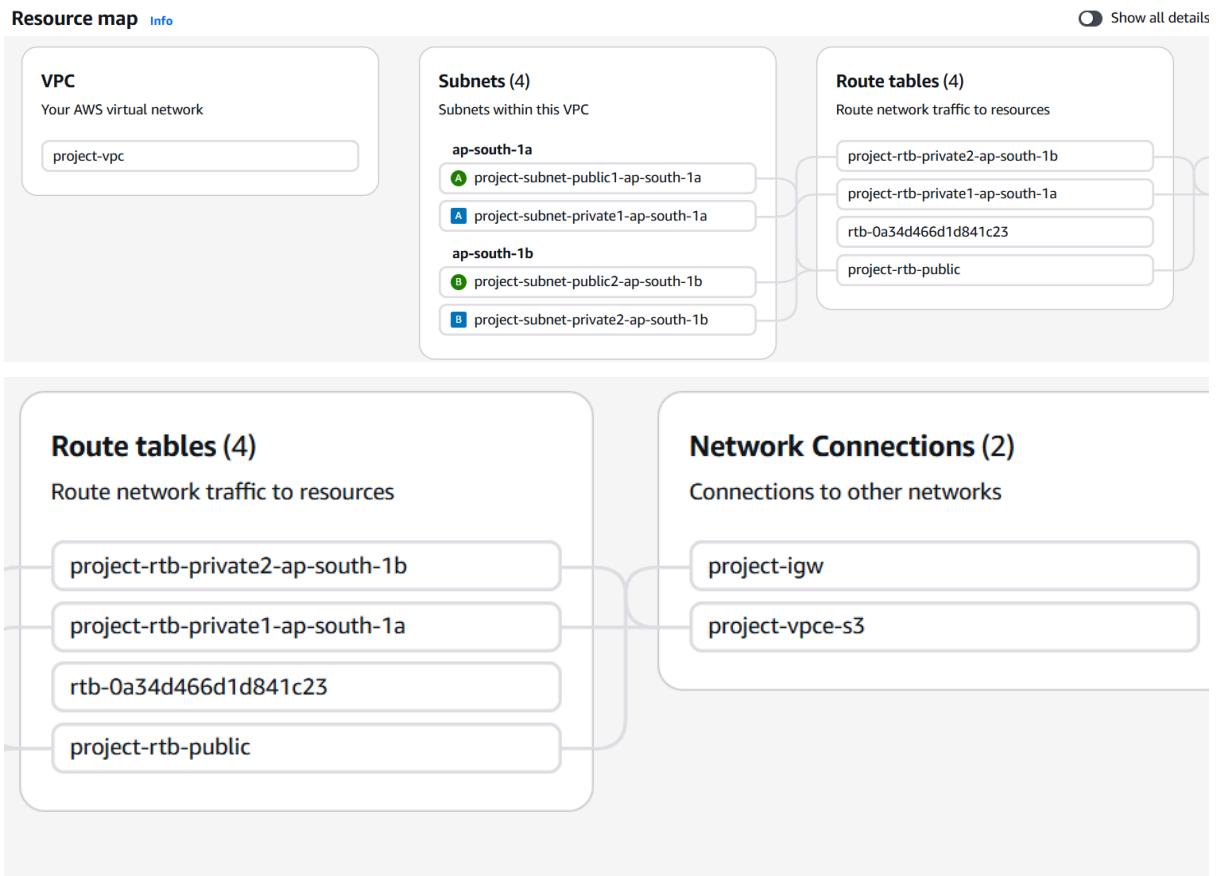
Success

## ▼ Details

- ✓ Create VPC: [vpc-0d1f90887075fb6cb](#)
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: [vpc-0d1f90887075fb6cb](#)
- ✓ Create S3 endpoint: [vpce-034620d08685566df](#)
- ✓ Create subnet: [subnet-0e9f7691f5e9dad70](#)
- ✓ Create subnet: [subnet-0fe0db4b309cc1510](#)
- ✓ Create subnet: [subnet-0d6961807e3d9d291](#)
- ✓ Create subnet: [subnet-0f9026e8dc7d88860](#)
- ✓ Create internet gateway: [igw-0fd267e6008889c97](#)
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: [rtb-0bff9cd0fff9e8e19](#)
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Create route table: [rtb-0985ac8ec9bc3bf70](#)
- ✓ Associate route table
- ✓ Create route table: [rtb-0a4b773fb37f97e8](#)
- ✓ Associate route table
- ✓ Verifying route table creation
- ✓ Associate S3 endpoint with private subnet route tables: [vpce-034620d08685566df](#)

## Details [Info](#)

VPC ID <a href="#">vpc-0d1f90887075fb6cb</a>	State Available	Block Public Access Off	DNS hostnames Enabled
DNS resolution Enabled	Tenancy default	DHCP option set <a href="#">dopt-01b3179112756fa73</a>	Main route table <a href="#">rtb-0a34d466d1d841c23</a>
Main network ACL <a href="#">acl-0523c1dc65a3cba23</a>	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID <a href="#">347363494863</a>
Encryption control ID -	Encryption control mode -		



## NETWORKING

### Network Topology:

Network topology is the arrangement(layout) of services and connections in a computer network. It shows how computers, switches, routers and other devices are connected and how data flows between them.

There are 2 types of topology

Physical-How devices are actually connected with cables.

Logical-How data flows inside the network(can be different from the physical layout)

### TYPES OF NETWORK TOPOLOGY

- Bus
- Star
- Tree
- Mesh
- Point to point
- Hybrid

### PROTOCOLS

1. Communication protocol
2. Security protocol
3. Application protocol

#### 4.Management protocol

#### **TCP & UDP Ports\_TCP-two way communication**

Port Number	TCP Service	UDP Service
20/21	FTP(File Transfer Protocol)	-----
22	SSH(Secure Shell)	-----
25	SMTP(Send Mail)	-----
53	DNS(Zone Transfer, TCP)	DNS(Queries,UDP)
67/68	-----	DHCP(Dynamic Host Configuration)
80	HTTP(Web)	-----
110	POP3>Email receive)	-----
123	-----	NTP(Network Time Protocol)
143	IMAP>Email receive)	-----
161/162	-----	SNMP(Monitoring)
443	HTTPS(Secure Web)	-----
3389	RDP(Remote desktop)	-----

## DAY 4

DATE:18.12.2025(THURSDAY)

### EC2(ELASTIC COMPUTE CLOUD)

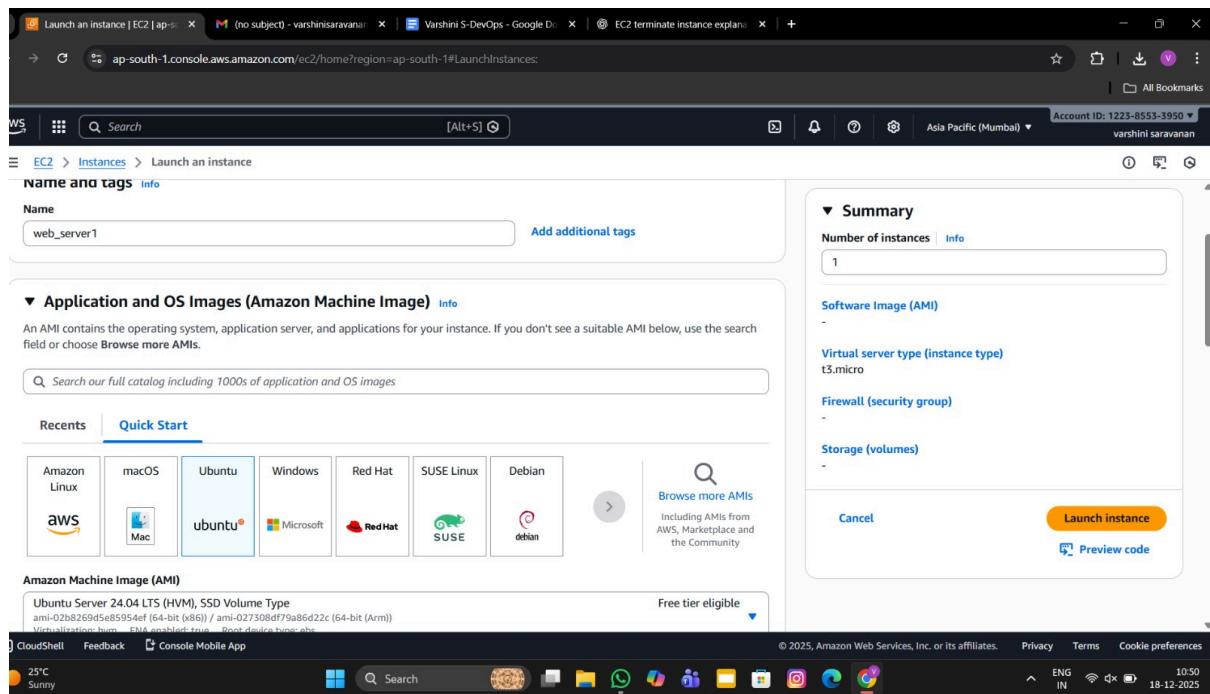
Backbone of AWS.

It works as a server. Within this web server is installed.

Ubuntu(SSH) and Windows(RDP) are used.

- Auto scaling-When the server is down, it automatically creates 2 instances with the same storage.
- Load balancing-Application and networking are important types. Similar to subnets but it is split by load here. Path based routing.

Classic  
Application  
Network  
Gateway



Screenshot of the AWS EC2 'Launch an instance' wizard step 1: Set instance details.

**Instance type:** t3.micro

**Additional costs apply for AMIs with pre-installed software**

**Key pair (login):** Info

You can use a key pair to securely connect to your instance. Ensure you have a key pair named 'web\_server1' available.

**Key pair name - required:** Select

**Network settings:** Info

Network: Info  
vpc-061d56746ad1af745  
Subnet: Info  
No preference (Default subnet in any availability zone)  
Auto-assign public IP: Info

**Create key pair:**

**Key pair name:** web\_server1

**Key pair type:** RSA (selected)

**Private key file format:** .pem (selected)

**Summary:**

- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 24.04, amd64 (ami-02b8269d5e85954ef)
- Virtual server type (instance type): t3.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

**Buttons:** Cancel, Create key pair, Launch instance, Preview code

Screenshot of the AWS EC2 'Launch an instance' wizard step 2: Set 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:** Create security group

**Security group name - required:** web\_server1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_~!@#\$%^&\*()\_~!@#\$%^&\*()

**Description - required:** Info

web\_server1 created 2025-12-18T05:19:52.872Z

**Inbound Security Group Rules:**

Type	Protocol	Port range	Action
All TCP	TCP	0-65535	Remove
Source type	Source	Description - optional	
Anywhere	Add CIDR, prefix list or security group	e.g. SSH for admin desktop	
0.0.0.0/0			

**Summary:**

- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 24.04, amd64 (ami-02b8269d5e85954ef)
- Virtual server type (instance type): t3.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

**Buttons:** Cancel, Launch instance, Preview code

The screenshot shows the AWS EC2 'Launch an instance' wizard. The current step is 'Creating security group rules'. A progress bar indicates 33% completion. Below the progress bar, there is a 'Details' section with the following text:

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

The screenshot shows the AWS EC2 'Instance details' page for an instance with the ID `i-0e904a2d002bd39f9`. The instance is associated with an IAM Role, an IMDSv2 ARN, and an Operator. It is currently in a Subnet with the ID `subnet-0de661fbff0ffddce5`. The Auto Scaling Group name is listed as '-' and Managed status is false. The page includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Status and alarms tab, it shows Status checks, System status checks (with one check passed), Instance status checks (with one check passed), and Attached EBS status checks (with one check passed). There is also a Metrics section and an Alarms section. A search bar at the top allows finding alarms by name. The left sidebar shows navigation links for EC2, Instances, Images, and Elastic Block Store.

The screenshot shows the AWS Management Console interface for launching a new EC2 instance.

**Launch Instance Wizard:**

- Name and tags:** A text input field contains "web\_server1".
- Application and OS Images (Amazon Machine Image):** A search bar is present, and a "Quick Start" tab is selected. It lists various AMI categories: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. An "ubuntu" icon is highlighted.
- Summary:** Shows 1 instance.
- Software Image (AMI):** Set to "t3.micro".
- Virtual server type (instance type):** Set to "t3.micro".
- Firewall (security group):** A dropdown menu is shown.
- Storage (volumes):** A dropdown menu is shown.
- Buttons:** "Cancel", "Launch instance", and "Preview code".

**Instances Page:**

- Instances (1/1) Info:** Shows 1 instance named "project-ec2" (ID: i-01d069f1ee75f99b6).
- Instance Details:** Status: Running, Type: t3.micro, Status check: 3/3 checks passed, Alarm status: View alarms +, Availability Zone: ap-south-1b, Public IPv4: ec2-3-109-5.
- Actions:** Buttons include "Connect", "Instance state", "Actions", and "Launch instances".

## WINDOWS

aws | Search [Alt+S]

☰ EC2 > Instances > i-06fcff466a1c1af97 > Connect to instance

Session Manager RDP client EC2 serial console

**Record RDP connections**  
You can now record RDP connections using AWS Systems Manager just-in-time node access. [Learn more](#)

**Instance ID**  
 i-06fcff466a1c1af97 (project2)

**Connection Type**

Connect using RDP client  
Download a file to use with your RDP client and retrieve your password.

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

[Download remote desktop file](#)

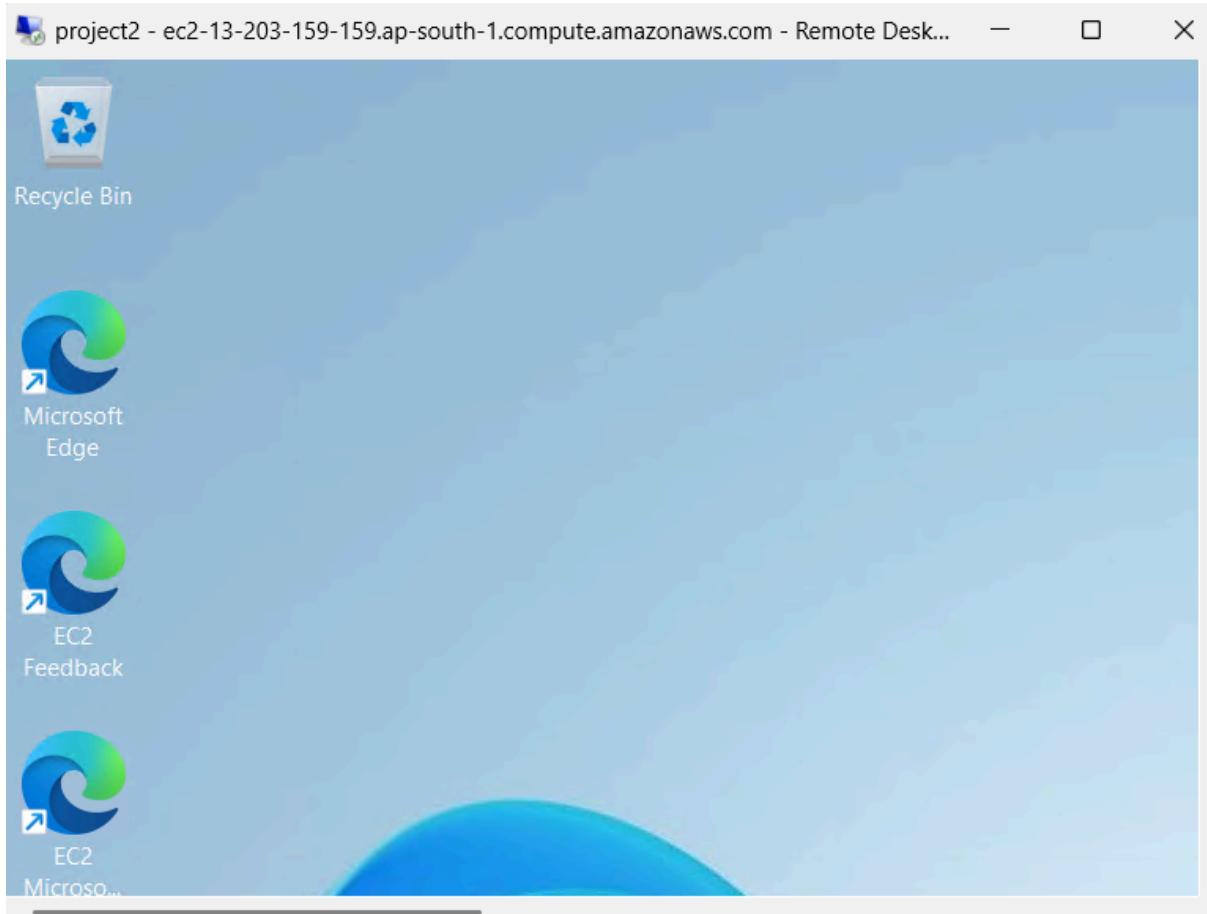
When prompted, connect to your instance using the following username and password:

**Public DNS**  
 ec2-13-203-159-159.ap-south-1.compute.amazonaws.com

Password copied

E(m)yhN25O)pNjIA)BC3XaPwQ&D(JvDe

User



DAY 5

DATE:19.12.2025(FRIDAY)

## ELASTIC BLOCK STORE

Inside EC2. Works like harddisk(gp2, gp3)

Stores data as blocks

Data is server oriented.

**Create volume** [Info](#)

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

**Volume settings**

**Volume type** [Info](#)  
General Purpose SSD (gp3)

**Size (GiB)** [Info](#)  
100  
Min: 1 GiB, Max: 65536 GiB.

**IOPS** [Info](#)  
3000  
Min: 3000 IOPS, Max: 80000 IOPS.

**Throughput (MiB/s)** [Info](#)  
125  
Min: 125 MiB, Max: 2000 MiB. Baseline: 125 MiB/s.

**Availability Zone** [Info](#)  
aps1-az1 (ap-south-1a)

**Snapshot ID - optional** [Info](#)

## S3-SIMPLE STORAGE SERVICE

Amazon S3 > Buckets

Successfully created bucket "sakthi-project-3"  
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

[View details](#) [X](#)

[General purpose buckets](#) [All AWS Regions](#) [Directory buckets](#)

**General purpose buckets (1) [Info](#)**  
Buckets are containers for data stored in S3.  
[Create bucket](#)

Name	AWS Region	Creation date
sakthi-project-3	Asia Pacific (Mumbai) ap-south-1	December 19, 2025, 11:11:24 (UTC+05:30)

**Account snapshot [Info](#)**  
Updated daily  
Storage Lens provides visibility into storage usage and activity trends.

**External access summary - new [Info](#)**  
Updated daily  
External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

Amazon S3 > Buckets > sakthi-project-3

Successfully edited static website hosting.

**Tags**

You can use tags to track storage costs, organize general purpose buckets, and specify permissions for a general purpose bucket. AWS-generated tags are created by AWS and are read-only. [Learn more](#)

S3 Console now uses s3>ListTagForResource, s3:TagResource, and s3:UntagResource APIs to manage tags on S3 general purpose buckets by default. To use these APIs for tagging, please provide permissions to s3>ListTagsForResource, s3:TagResource, and s3:UntagResource actions. [Learn more](#)

User-defined tags AWS-generated tags

User-defined tags (0) [Info](#)

Add new tag

Key Value - optional

There are no user-defined tags associated with this bucket.

Add new tag

**Default encryption**

Server-side encryption is automatically applied to new objects stored in this bucket.

Upload succeeded

For more information, see the [Files and folders](#) table.

After you navigate away from this page, the following information is no longer available.

**Summary**

Destination	Succeeded	Failed
s3://sakthi-project-3	1 file, 8.4 KB (100.00%)	0 files, 0 B (0%)

**Files and folders** (1 total, 8.4 KB)

Name	Folder	Type	Size	Status	Error
agri.html	-	text/html	8.4 KB	Succeeded	-

**Objects** (1/1)

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

Name	Type	Last modified	Size	Storage class
agri.html	html	December 19, 2025, 11:14:46 (UTC+05:30)	8.4 KB	Standard

## Bucket policy

[Edit](#)

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::sakthi-project-3/*"  
    }  
  ]  
}
```

sakthi-project-3 - S3 bucket | S | AI Crop Health Dashboard | +

sakthi-project-3.s3.ap-south-1.amazonaws.com/agri.html

Action required

### Crop Dashboard

#### Dashboard Overview

Overview

Crop Health: Good

Weather: 28°C | 65% Humidity

Soil Moisture: Moderate

Pest Risk: High

**Crop Health Map (Time-Lapse)**

Use slider to view different days

**Soil Moisture Trend**

Day	Soil Moisture (%)
Day 1	44
Day 2	50
Day 3	42
Day 4	39
Day 5	41

## **DAY 6**

**DATE:23.12.25(TUESDAY)**

### **IAM-Identity and Access Management**

Security purpose

- 1) User
- 2) Policy
- 3) Group
- 4) Role-Service to service communication(restricting service)

### **POLICIES**

- 1)Inline policy-Policy is created within the user.
- 2)AWS managed policies
- 3)Customer managed policies-Derived from Aws managed policies. Policy and user are created separately.

### **USER**

Access key

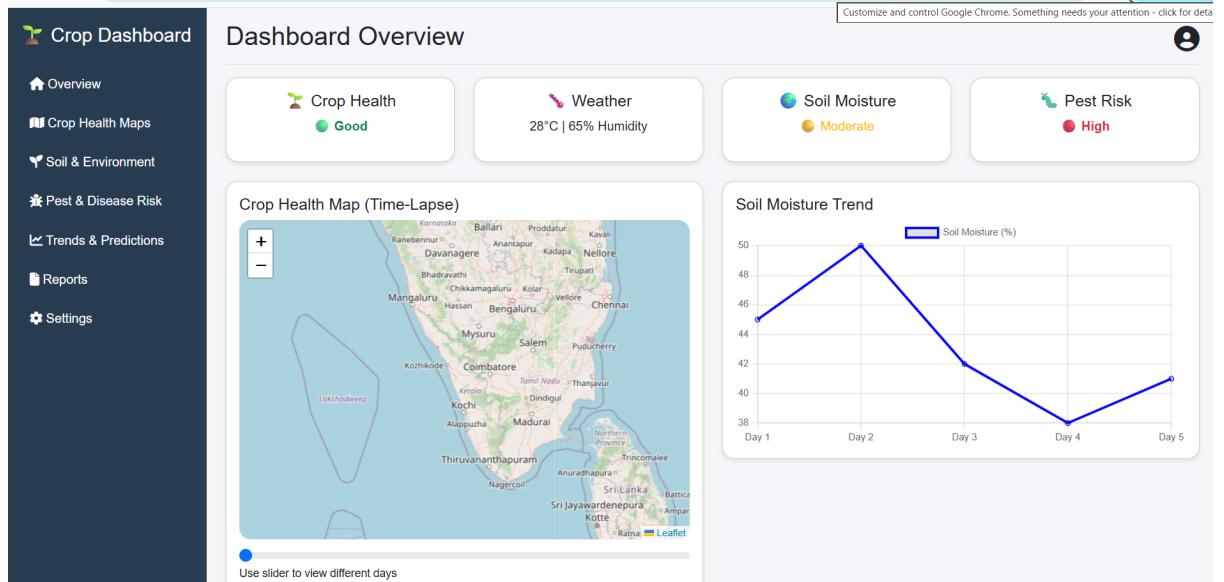
Secret access key

MFA-Multi factor authentication

# DAY 7

## DATE:24.12.25(WEDNESDAY)

### TASK -1



## Rest api

The screenshot shows the 'Amazon Elastic Container Service' cluster overview page. The left sidebar lists services like Express Mode, Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, Amazon Batch, Documentation, Discover products, and Subscriptions. The main content area shows a 'Cluster overview' for a cluster named 'arnaws:ecs:ap-south-1:347365494863:cluster/rest-api'. It displays status (Active), CloudWatch monitoring (Default), and registered container instances (none). Below this are sections for 'Services' (Draining 1, Pending 0, Running 1) and 'Tasks' (Pending 0, Running 1). At the bottom is a 'Services' table with one entry: 'rest-api-backend-service-pim0j9d9' (ARN: arnaws:ecs:ap-south-1:347365494863:service/rest-api-b..., Status: Active, Type: REPLICAS, Tasks: 1/1 Tasks running).