

DAY 1

DATE : 15/22/25

DAY : MONDAY

CLOUD COMPUTING: Cloud computing means using computers, storage, and software through the Internet instead of your own system. Data & apps run on remote servers

For example google drive,icloud.firebaseio etc,..

Cloud platforms:

- AWS
- AZURE
- GCP

Methodology used before:

- Agile
- Waterfall

DevOps : developing + operations

Integrating software development and IT operations

Distributives - it's like a storage unit

Data is split, replicated, and distributed across multiple nodes for reliability, scalability, and availability.

What exactly is this na.. In general while working in it we could handle multiple operations so if one person works on developing part then Other needs to work on operations and hence to avoid clash of that copy it will get cloned .So the work done by both members will be saved and used effectively.

Git is smtg we use to develop github

For continuous integration

Jenkins → Automates build, test, deploy

TOOLS USED:

1. Git - Source code versioning
2. Github - Code hosting, collaboration
3. Github actions - CI/CD inside GitHub
4. Docker - Package app + dependencies into containers
5. aws/cloud platform - Cloud infrastructure services
6. Tereform - Cloud infrastructure automation
7. Grafana - Data visualization dashboards
8. Kubernetes - EKS
9. Promathesis - calculate visualizing from grafana in simple words it'll show the outputs./Metrics monitoring.

Docker : A kind of storage database.Also it is a tool that helps you package an application with everything it needs (code, libraries, settings) and run it anywhere without problems.

DAY 2

DATE :16/22//25

DAY : TUESDAY

AWS

Amazon web services - used widely in many it industries

- Cloud computing platform provided by amazon.
- It works on the basis of Pay-as-you-go
- It offers IT demand resources over the internet - things like servers,storage,databases,networking,machine learning.

VPC

EC2 - works as server

IAM- security based

S3 - storage

RDS

AMAZON COMPUTE SERVICES:

Grp of AWS services tha provide virtual servers,containers,serverless computing, and specialized compute resources(like GPUs and HPC).

Compute services- sever

- Server - for communication (client - intermediate - database) where intermediate is the server
- Webserver - communication occurs through websites
- Serverless - for communication but no need of intermediate

To do : what is reverse proxy & forward proxy.

FORWARD PROXY : A forward proxy hides the user (client) from the internet.

REVERSE PROXY: A reverse proxy hides the server from users.

AMAZON STORAGE SERVICES:

Set of cloud storage solutions provided by amazon web services.

- It stores max of unstructured data like image,video,audio etc.,
- It can also be hosted statically through S3.

Examples :AMAZON S3(Simple storage services)

AMAZON EB2(Elastic block store)

Things inside S3 are objects- it works independently
THings inside EBS are block- it works with servers

AMAZON NETWORKING SERVICES:

Collection of cloud networking tools that help you connect, secure and deliver

VPC

Security grp present inside vpc that says which is to be allowed or not like https.
Database - RDS (relational database state) sql,mysql,postgresql are used here for communication
since it use structured dataset

AMAZON DATABASE SERVICES:

It is a fully managed database solution provided by AWS.

They let u store, manage, and analyse structured and unstructured data

AMAZON RDS

AMAZON DynamOB

AWS GLOBAL INFRASTRUCTURE

Physical backbone that provides all AWS cloud services worldwide.

Main parts are Regions and Availability zones(AZs)

Regions is a geographical area

Each region consists of multiple availability zones.

Regions are isolated from each other for data sovereignty ,compliance and fault isolation. They choose regions based on proximity to the users, legal requirements and redundancy.

Example:

US east(N.Virginia)=us-east-1

Asia Pacific(Mumbai)=ap-south-1

DAY 3

DATE : 17/22/25

DAY : WEDNESDAY

ABOUT PROJECT:

- 1.Deployment - static webpage- FOUNDATION
- 2.Deployment - backend REST API-INDUSTRY-LEVEL-PROJECT

VIRTUAL PRIVATE CLOUD

Private space btw client and server it all happens inside the network

Public - connected

Private - internet not connected

Subnet- inside instance which is called servers.

IP Address are of two types:

- IPV4-version 4 (only in numerical)-32 bits
- IPV6-version 6 (alphanumeric)- 64 bits

One system has one IP address

Basic a we'll be using 16 (divided by 16 na we'll be 60,000+ IPs to use inside VPC

Subnet - present inside vpc

Public- hidden data handlers

Private - has internet connection

NAT GATEWAY: update happens inside the server but it should not be known to the client.

ROUT TABLE:It is used for connection.

SECURITY GROUP:it must allow certain port and port numbers

Two types of port can be given:

Inbuilt - can't change the port number

Example : https-443

Customized - we can create a port and port number

RDP - windows

SSL - linux

Nacl - it's kinda security firewall

VPC CREATION:

▼ Details

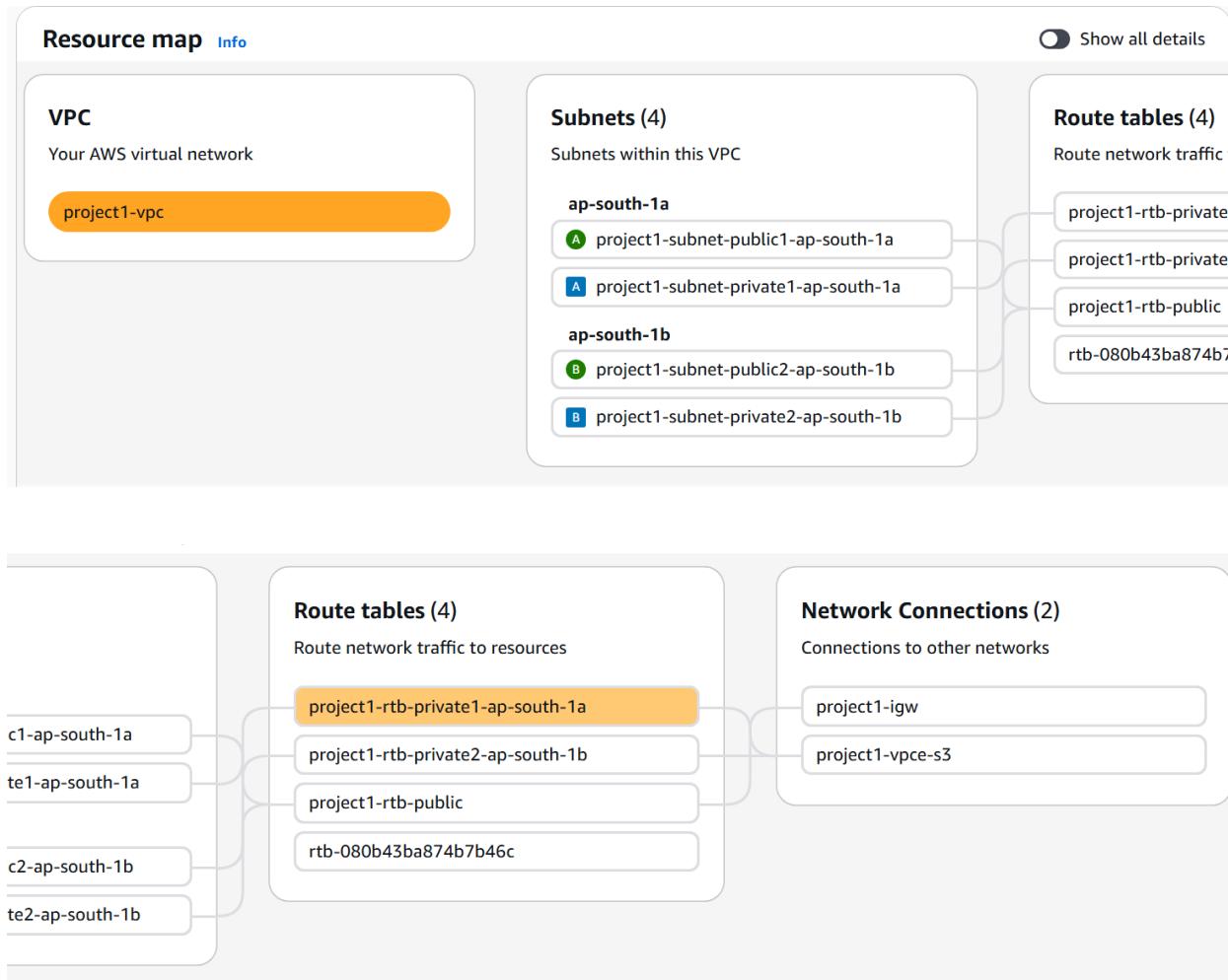
- ✓ Create VPC: [vpc-08cb7e670e85d6bd8 ↗](#)
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: [vpc-08cb7e670e85d6bd8 ↗](#)
- ✓ Create S3 endpoint: [vpce-05582130190d25a07 ↗](#)
- ✓ Create subnet: [subnet-0e95c52a52798411a ↗](#)
- ✓ Create subnet: [subnet-0d6d82e322e596f2e ↗](#)
- ✓ Create subnet: [subnet-0a4ad58627afce85b ↗](#)
- ✓ Create subnet: [subnet-014d8aeedc236673d ↗](#)
- ✓ Create internet gateway: [igw-04b7fb63f1d1f71d3 ↗](#)
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: [rtb-0ecc3143e81fee86a ↗](#)
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Create route table: [rtb-080f81657d69d6bf7 ↗](#)
- ✓ Associate route table
- ✓ Create route table: [rtb-051f9554a6a0b8a97 ↗](#)
- ✓ Associate route table
- ✓ Verifying route table creation
- ✓ Associate S3 endpoint with private subnet route tables: [vpce-05582130190d25a07 ↗](#)

vpc-08cb7e670e85d6bd8 / project1-vpc

Actions ▾

Details		Info	
VPC ID	vpc-08cb7e670e85d6bd8	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-05461d3b60c0a38c7	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	IPv4 CIDR 10.0.0.0/16
Encryption control ID	-	Encryption control mode	Route 53 Resolver DNS Firewall rule groups -
DNS hostnames Enabled			
Main route table rtb-080b43ba874b7b46c			
IPv6 pool -			
Owner ID 573247409194			

[Resource map](#) | [CIDRs](#) | [Flow logs](#) | [Tags](#) | [Integrations](#)



NETWORKING:

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

It has two types:

- physical topology- How devices are actually connected with cables.
- Logical topology - How data flows inside the network (can be different from the physical layout).

Types of network topology:

Bus,star

NETWORK PROTOCOLS

COMMUNICATION PROTOCOL:

TCP - two way communication-watsapp

UDP - one way communication-gaming

Security protocols:

APPLICATION PROTOCOLS:

Http - used for website but not so secured
Ftp - for files transferring
Sntp - for mail transferring

Common TCP & UDP Ports:

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 : 18/22/25

DAY : THURSDAY

ELASTIC COMPUTE CLOUD

EC2 - can be worked in ubuntu and windows

ubuntu-SSH;windows-RDP

EC2 is mandatory to create anything inside AWS.

Important concept:

Autoscaling:If a server gets down it'll automatically create another server itself.

Load balancing:it has three servers, it'll split data for multiple user to handle the server.

Types of LB:

application;network;S3Anyone can access it so we give it all as 0

The screenshot shows two main views of the AWS EC2 service.

Top View (Launch an instance): This view is titled "Launch an instance | EC2 | ap-south-1" and shows the "Instances" section. It includes fields for "Name and tags" (with "web_server1" entered), "Application and OS Images (Amazon Machine Image)" (with "Ubuntu Server 24.04 LTS (HVM, SSD Volume Type)" selected), and "Virtual server type (instance type)" (set to "t3.micro"). A large orange "Launch instance" button is prominent at the bottom right. The status bar indicates "Number of instances: 1".

Bottom View (Instances dashboard): This view is titled "Instances (1/1) | Info" and shows a single instance named "project1" with the ID "i-0e3adc7e47ba93590". The instance is listed as "Running" (status check "3/3 checks passed"). The dashboard includes tabs for "Details", "Status and alarms" (selected), "Monitoring", "Security", "Networking", "Storage", and "Tags". Under "Status checks", it lists "System status checks" (System reachability check passed) and "Attached EBS status checks" (Attached EBS reachability check passed). The sidebar on the left provides navigation links for EC2, Dashboard, AWS Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images (selected), AMIs, and AMI Catalog.

SAN (5732-4740-9194) ▾

Instances (1/1) Info Last updated 2 minutes ago

Find Instance by attribute or tag (case-sensitive)

All states ▾

Instance ID	Instance state	Instance type	Status check	Alarm status	Available
i-0e3adc7e47ba93590	Running	t3.micro	3/3 checks passed	View alarms +	ap-sou

i-0e3adc7e47ba93590 (project1)

Details Status and alarms Monitoring Security Networking Storage Tags Actions

Status checks Info

Status checks detect problems that may impair i-0e3adc7e47ba93590 (project1) from running your applications.

System status checks System reachability check passed

Instance status checks Instance reachability check passed

Attached EBS status checks Attached EBS reachability check passed

Metrics

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Launch an instance | EC2 | ap-south-1 | Varnshini S-DevOps - Google Doc | EC2 terminate instance explaination | +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

aws Search [Alt+S] Asia Pacific (Mumbai) Account ID: 1234-5678-9012 varshini saravanan

EC2 Instances Launch an instance

Create key pair

Key pair name web_server11

Key pairs allow you to connect to your instance securely. The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type RSA ED25519

RSA RSA encrypted private and public key pair

ED25519 ED25519 encrypted private and public key pair

Private key file format .pem .ppk

.pem For use with OpenSSH

.ppk For use with PuTTY

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

Cancel Create key pair

Summary Number of instances Info

Software Image (AMI) Canonical, Ubuntu, 24.04, amd6...read more

Virtual server type (instance type) t3.micro

Network security group New security group

Storage (volumes) Volume(s) - 8 GiB

Launch instance Preview code

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

25°C Sunny ENG IN 10:50 18-12-2025

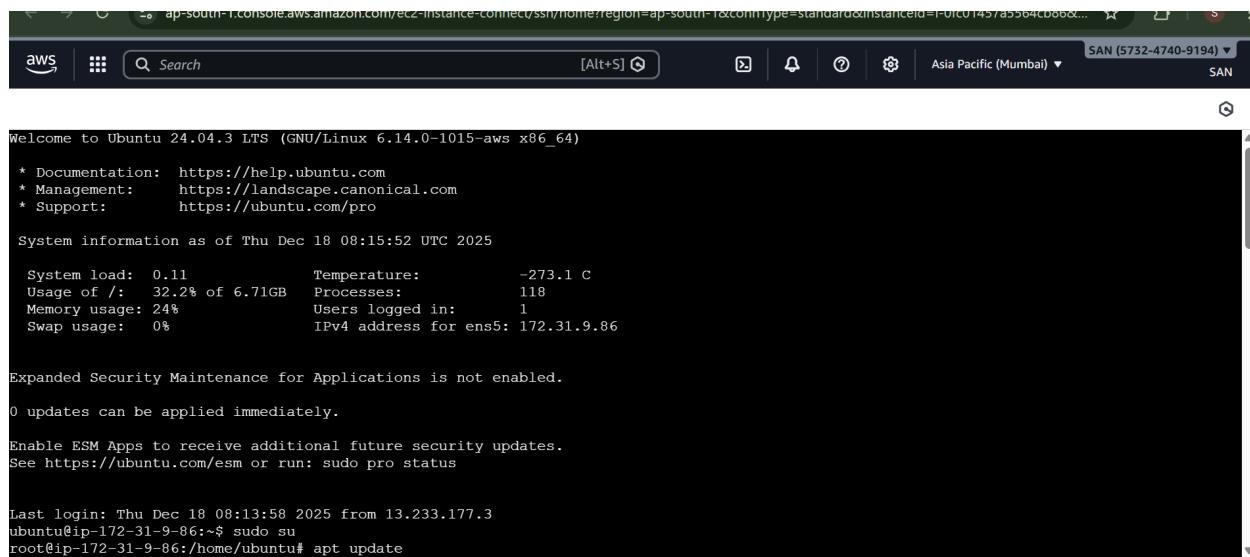
Classic server -

Application server- path based router- occurs only inside application

Network server - Port based router (we create port numbers manually) - can run in multiple application and also works in millisecond

Gateway server - to run third part in my port

WEB POSTING:



```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Thu Dec 18 08:15:52 UTC 2025

System load: 0.11 Temperature: -273.1 C
Usage of /: 32.2% of 6.71GB Processes: 118
Memory usage: 24% Users logged in: 1
Swap usage: 0% IPv4 address for ens5: 172.31.9.86

Expanded Security Maintenance for Applications is not enabled.

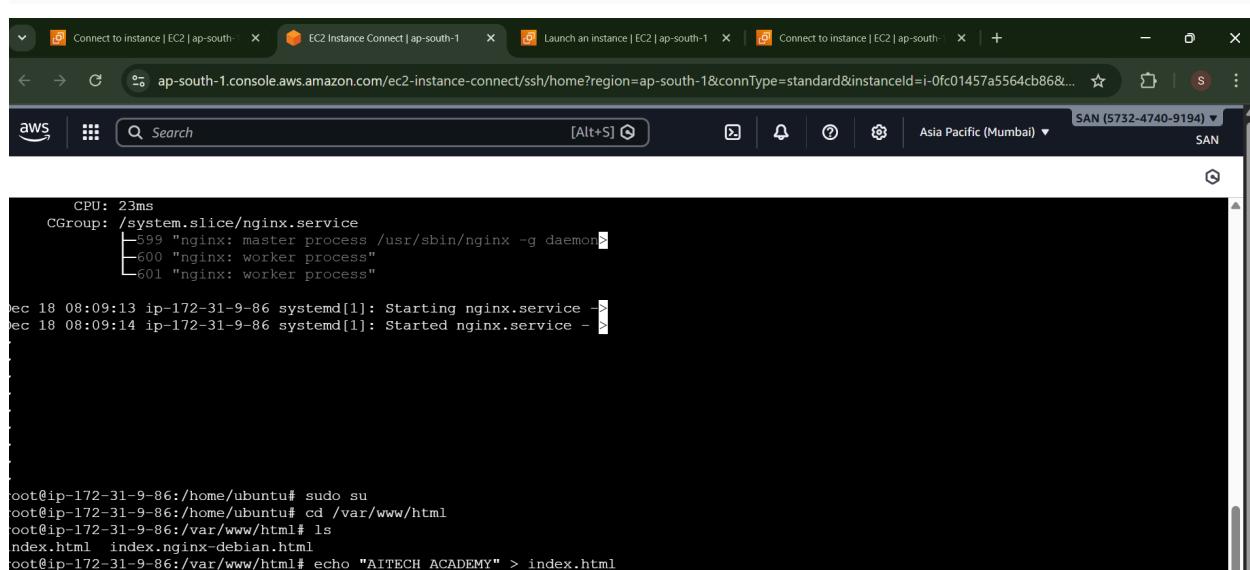
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Dec 18 08:13:58 2025 from 13.233.177.3
ubuntu@ip-172-31-9-86:~$ sudo su
root@ip-172-31-9-86:/home/ubuntu# apt update
```

i-0fc01457a5564cb86 (project1)

PublicIPs: 13.127.148.76 PrivateIPs: 172.31.9.86



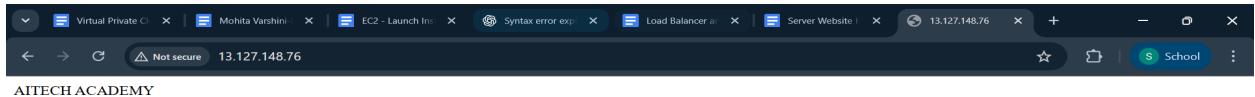
```
CPU: 23ms
CGroup: /system.slice/nginx.service
└─599 "nginx: master process /usr/sbin/nginx -g daemon"
   ├─600 "nginx: worker process"
   ├─601 "nginx: worker process"

Dec 18 08:09:13 ip-172-31-9-86 systemd[1]: Starting nginx.service -
Dec 18 08:09:14 ip-172-31-9-86 systemd[1]: Started nginx.service -
```

```
oot@ip-172-31-9-86:/home/ubuntu# sudo su
oot@ip-172-31-9-86:/home/ubuntu# cd /var/www/html
oot@ip-172-31-9-86:/var/www/html# ls
index.html index.nginx-debian.html
oot@ip-172-31-9-86:/var/www/html# echo "AITECH ACADEMY" > index.html
oot@ip-172-31-9-86:/var/www/html# cat index.html
AITECH ACADEMY
oot@ip-172-31-9-86:/var/www/html#
```

i-0fc01457a5564cb86 (project1)

PublicIPs: 13.127.148.76 PrivateIPs: 172.31.9.86



DAY 5

DATE : 19/22//25

DAY : FRIDAY

ELASTIC BLOCK STORE

It's inside ec2, it works like a harddisk (gp2, gp3).

It stores things as blocks and the part inside is called volume. In this volume storage can be increased but it can't decrease.

S3- it stores things as objects . Object (files) stored inside the container called bucket.

SIMPLE STORAGE SERVICE: Bucket work as container and we can store any datas and anyone from this world can access it

(Static webposting: it can be hosted inside the bucket itself)

The screenshot shows the AWS EC2 Volumes Create volume interface. The 'Volume settings' section is displayed with the following parameters:

- Volume type: General Purpose SSD (gp3)
- Size (GiB): 16
- IOPS: 3000
- Throughput (MiB/s): 125
- Availability Zone: aps1-aZ1 (ap-south-1a)

At the bottom of the page, there are links for CloudShell, Feedback, and Console Mobile App. The footer contains copyright information for 2025, Amazon Web Services, Inc. or its affiliates, and links for Privacy, Terms, and Cookie preferences.

Successfully created volume vol-03f727c8612808882.

Volumes (3) Info

Last updated less than a minute ago

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID
	vol-0c04d47492a856c51	gp3	8 GiB	3000	125	snap-02aec5d
	vol-03f727c8612808882	gp3	16 GiB	3000	125	-
	vol-0b1e051160b3c7e92	gp3	30 GiB	3000	125	snap-02be30c

Fault tolerance for all volumes in this Region

Snapshot summary

Recently backed up volumes / Total # volumes
0 / 2

Last updated on Fri, Dec 19, 2025, 10:46:36 AM (GMT+05:30)

Data Lifecycle Manager default policy for EBS Snapshots status
No default policy set up | Create policy

Details

Volume ID vol-03f727c8612808882	Size 16 GiB	Type gp3	Status check Okay
AWS Compute Optimizer finding <small>Opt-in to AWS Compute Optimizer for recommendations.</small>	Volume state Available	IOPS 3000	Throughput 125
Fast snapshot restored No	Availability Zone aps1-az1 (ap-south-1a)	Created Fri Dec 19 2025 10:50:33 GMT+0530 (India Standard Time)	Multi-Attach enabled No
Attached resources -	Outposts ARN -	Managed false	Operator -
Source			
Snapshot ID -	Source volume ID -		

Successfully created bucket "helluluhahbucket". To upload files and folders, or to configure additional bucket settings, choose View details.

General purpose buckets All AWS Regions **Directory buckets**

General purpose buckets (1) Info

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
helluluhahbucket	Asia Pacific (Mumbai) ap-south-1	December 19, 2025, 11:10:26 (UTC+05:30)

Account snapshot Info
Updated daily
[View dashboard](#)
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.

The screenshot shows the AWS S3 console with the path `Buckets > helluluahbucket`. On the left, a sidebar lists various management features like Access Points, Storage Lens, and IAM Access Analyzer. The main area is titled "Bucket policy" and contains the following JSON code:

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

A green success message at the top right says "Successfully edited bucket policy." There are "Edit" and "Delete" buttons above the JSON code, and a "Copy" button to its right.

The screenshot shows a web browser window with the URL `helluluahbuckets3.ap-south-1.amazonaws.com/test-dashboard.html`. The page is titled "Agrisahayak" and displays a welcome message: "Hi Test User!!! Welcome to Agrisahayak".

The interface includes two main sections:

- Ask Me:** A box with a speech bubble icon, labeled "Ask Me", and the subtext "Get answers to your problems". It has a "Home" link below it.
- Schemes:** A box with a document icon, labeled "Schemes", and the subtext "Government agricultural schemes". It also has a "Home" link below it.

Below these boxes, there's a text input field with placeholder "Type your question..." and a "Send" button. A "Logout" link is located at the top right of the header.

At the bottom of the page, there are three sections:

- Agricultural Schemes:** A heading followed by "PM-KISAN" and a subtext "Income support of ₹6,000 per year to all farmer families". A "Link →" button is provided.
- Kisan Credit Card:** Another heading.

DAY 6

DATE : 23/22/25

DAY : TUESDAY

IDENTIFY AND ACCESS MANAGEMENT:

- It's mainly used for security purposes.
- IAM user - root user can only provide certain needed things to the IAM user
- Access providing thing is called policy(rules and regulations)

{1.user 2.policy 3.groups 4.Role}

1. User - one who has the need to use policy
2. Policy - more like a rules and regulations also like providing access
3. Groups - applying policy to certain grp of members
4. Role - service to service communication.(restricting service)

Policies:

- **Inline policy** - already an user is created and policy is created with them. If a user is deleted then the policy will also be deleted as both user and policies are created together
- **AWS managed policy** - policies are already created by them (inbuilt policy)
- **Customer managed policy** - policies are created by customer's wish even for creating this they might take or apply certain policies from AWS managed policy . Policy and user are created independently.

ACCESS KEY - Works as a password

SECRET ACCESS KEY- works as a hidden key

Muti factor authentication - can be used wherever we want sort of...

The screenshot shows the 'Specify permissions' step of the 'Create policy' wizard in the AWS IAM console. The left sidebar shows 'Step 1: Specify permissions' is selected, with 'Step 2: Review and create' available. The main area is titled 'Specify permissions' with an 'Info' link. It contains a 'Policy editor' section with tabs for 'Visual' (selected), 'JSON', and 'Actions'. The 'S3' service is selected under 'Specify what actions can be performed on specific resources in S3'. Under 'Actions allowed', there is a search bar 'Filter Actions' and a list of actions: 'All actions' (selected), 'All S3 actions (s3:*)', 'List (Selected 16/16)', and 'Read (Selected 62/62)'. The 'Effect' dropdown is set to 'Allow'. At the bottom right, there are 'Expand all' and 'Collapse all' buttons.

IAM > Users

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests
- New

Access reports

Access Analyzer

Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group	Last activity	MFA	Password age
hinatahyuga	/	1	1 hour ago	-	1 hour

Search

Delete Create user

IAM > Users > hinatahyuga

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests
- New

Access reports

Access Analyzer

Summary

ARN: arn:aws:iam::573247409194:user/hinatahyuga

Console access: Enabled without MFA

Created: December 23, 2025, 10:50 (UTC+05:30)

Last console sign-in: Today

Access key 1: Create access key

Permissions

Groups (1) Tags Security credentials Last Accessed

Permissions policies (3)

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly
AmazonEC2FullAccess	AWS managed	Group avengers

Filter by Type

Remove Add permissions

IAM > Policies

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles
- Policies**
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests
- New

Access reports

Access Analyzer

Policies (1/1440) Info

A policy is an object in AWS that defines permissions.

Policy name	Type	Used as	Description
AccessAnalyzerServiceRolePolicy	AWS managed	None	Allow Access Analyzer to analyze resources
AccountManagementPolicy	AWS managed	None	For use with accounts created through
AdministratorAccess	AWS managed - job function	Permissions policy (1)	Provides full access to AWS services across all accounts
AdministratorAccess	AWS managed	None	Grants account administrative permissions
AdministratorAccess	AWS managed	None	Grants account administrative permissions
AIOpsAssistantIncidentResponsePolicy	AWS managed	None	Provides permissions required by the /
AIOpsAssistantPolicy	AWS managed	None	Provides ReadOnly permissions required by the /
AIOpsConsoleAdministratorPolicy	AWS managed	None	Grants full access to Amazon AI Operations
AIOpsOperatorAccessPolicy	AWS managed	None	Grants access to the Amazon AI Operations

Actions Delete Create policy

CloudShell Feedback Console Mobile App

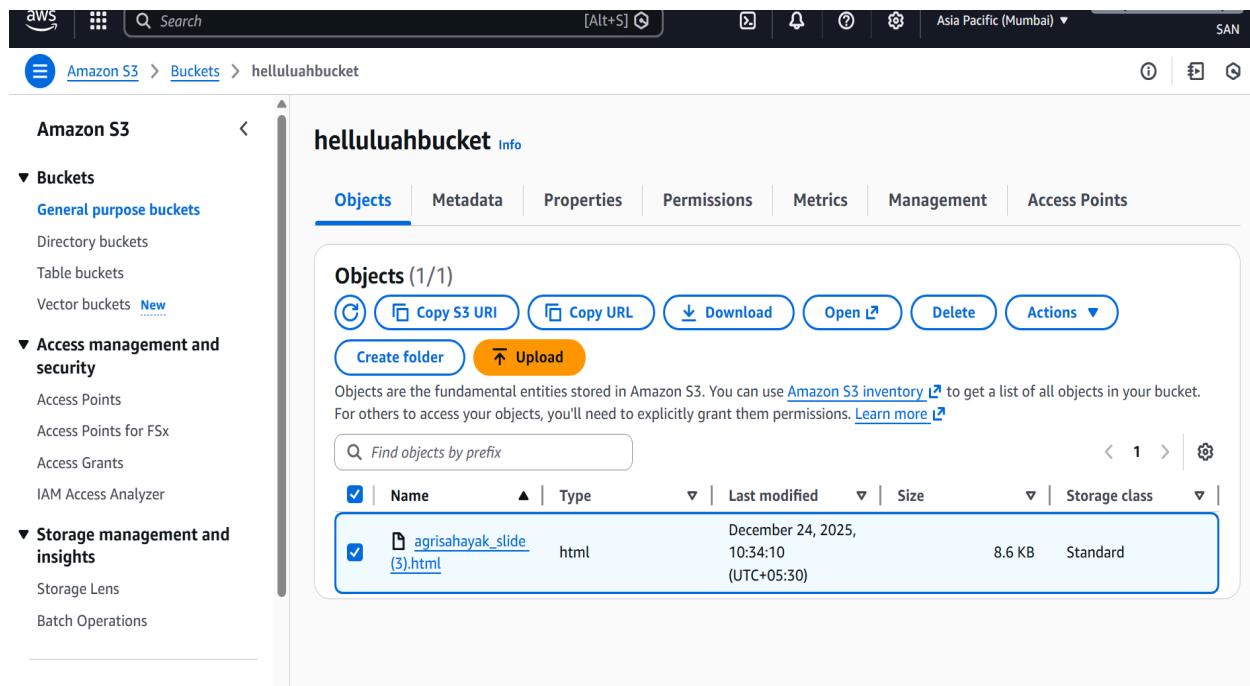
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

DAY 7

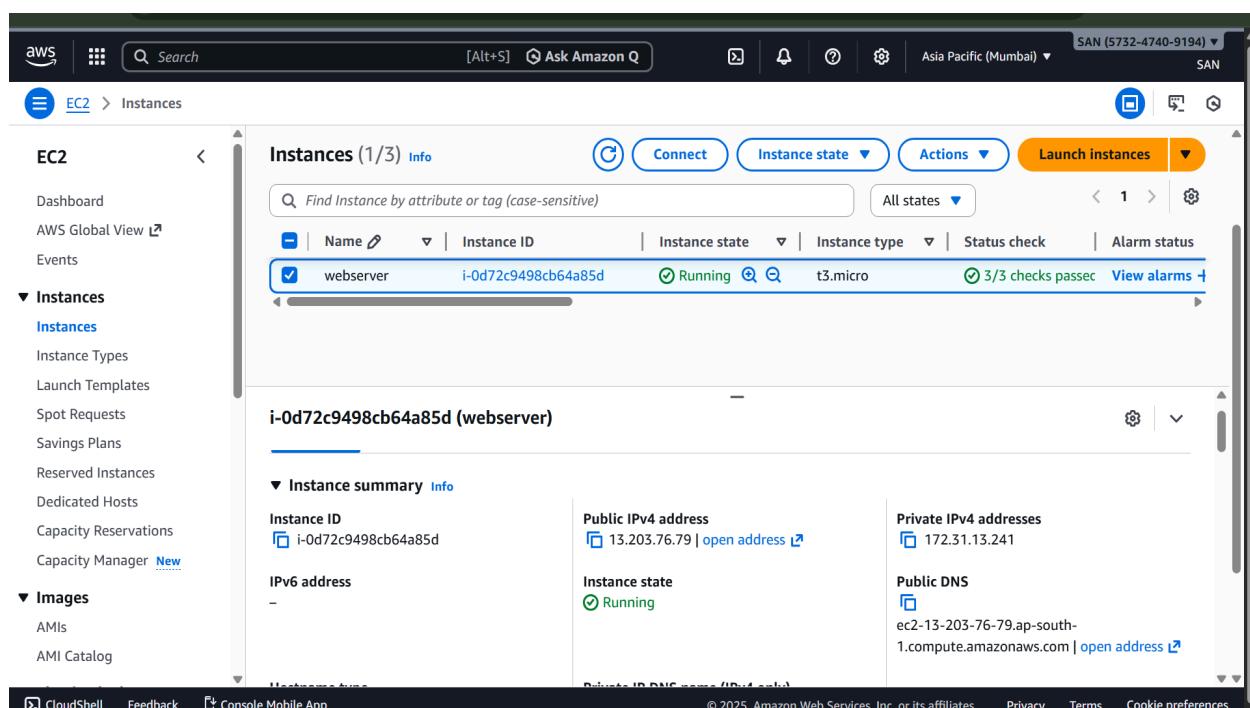
DATE : 24/22/25

DAY : WEDNESDAY

Task : The task was to create an S3 bucket, upload an HTML file, and host that file publicly using an EC2 web server inside a VPC with an Internet Gateway.



The screenshot shows the AWS S3 console. On the left, the navigation pane includes sections for Buckets, Access management and security, Storage management and insights, and Account and organization. The main area displays the 'helluluahbucket' bucket. It contains one object named 'agrisahayak_slide(3).html'. The object is listed with its name, type (HTML), last modified date (December 24, 2025, 10:34:10 UTC+05:30), size (8.6 KB), and storage class (Standard).



The screenshot shows the AWS EC2 console. The navigation pane includes sections for Instances, Images, and AMIs. The main area displays the 'Instances' page, which lists one instance named 'webserver' with the ID 'i-0d72c9498cb64a85d'. The instance is shown as 'Running' with a Public IPv4 address of 13.203.76.79 and a Private IPv4 address of 172.31.13.241. The Public DNS is ec2-13-203-76-79.ap-south-1.compute.amazonaws.com.

AWS EC2 Instances - i-0d72c9498cb64a85d

Instance summary for i-0d72c9498cb64a85d (webserver)

Actions

Public IPv4 address: 13.203.76.79 | [Open address](#)

Private IP4 addresses: 172.31.13.241

Public DNS: ec2-13-203-76-79.ap-south-1.compute.amazonaws.com | [Open address](#)

Private IP DNS name (IPv4 only): ip-172-31-13-241.ap-south-1.compute.internal

Instance type: t3.micro

VPC ID: vpc-05acb9318283f8efc

Elastic IP addresses: -

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.

```

inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/top_level.txt
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/INSTALLER
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/AUTHORS.rst
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/LICENSE
inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
ubuntu@ip-172-31-13-241:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
ubuntu@ip-172-31-13-241:~$ aws --version
aws-cli/2.32.23 Python/3.11 Linux/6.14.0-1015-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-13-241:~$ aws s3 ls
2025-12-19 06:19:22 helluluahbucket
ubuntu@ip-172-31-13-241:~$ sudo aws s3 cp s3://helluluahbucket/index.html /var/www/html/
fatal error: An error occurred (404) when calling the HeadObject operation: Key "index.html" does not exist
ubuntu@ip-172-31-13-241:~$ sudo aws s3 cp "s3://helluluahbucket/agrisahayak_slide (3).html" /var/www/html/
download: s3://helluluahbucket/agrisahayak_slide (3).html to ../../var/www/html/agrisahayak_slide (3).html
ubuntu@ip-172-31-13-241:~$ ls /var/www/html/
'agrisahayak_slide (3).html' index.html
ubuntu@ip-172-31-13-241:~$ 
```

i-0d72c9498cb64a85d (webserver)

Public IPs: 13.203.76.79 Private IPs: 172.31.13.241

helluluahbucket.s3.ap-south-1.amazonaws.com/agrisahayak_slide (3).html

Impact and Benefits – AgriSahayak

Social & Environmental Impact

- Supports sustainable farming
- Reduces crop losses & chemical misuse

Economic Benefits

- Increases farmer income
- Reduces dependency on middlemen

Instant Guidance

- Provides quick local-language answers
- Reduces decision delays

Farmer Empowerment

- Builds confidence & self-reliance
- Strengthens trust via expert validation

Inclusive Accessibility

- Usable by low-literacy farmers via voice, text, image
- Malayalam-first support ensures inclusivity

Empowering farmers with real-time, localized, and trusted advisory.

1.cluster - tasks are built inside this cluster

We gonna create task and cluster separately

Container is inside task and that container is application

Cluster - task - container

Repository (ECR) - docker images are set inside this repository , more like inserting files inside it.

Docker - It's just a tool. devops tool, creates all environment variables on its own.

Escape + key - to save

:wq - save and exit

PATH BASED ROUTING - APPLICATION

The screenshot shows the AWS ECR 'Create private repository' page. At the top, there is a navigation bar with the AWS logo, a search bar, and various icons. The main heading is 'Create private repository'. Below it, there are two sections: 'General settings' and 'Image tag settings'.

General settings: This section contains a 'Repository name' input field. The placeholder text is 'Enter a concise name. Repositories support namespaces, which you can use to group similar repositories.' The input field contains the value '573247409194.dkr.ecr.ap-south-1.amazonaws.com/demoone'. A note below the input field states: '7 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, and special characters _-./'.

Image tag settings: This section includes an 'Image tag mutability' setting. It has two options: 'Mutable' (selected) and 'Immutable'. The 'Mutable' option is described as 'Image tags can be overwritten.' The 'Immutable' option is described as 'Image tags can't be overwritten.' Below this, there is a 'Mutable tag exclusions' section with a note: 'Tags that match these filters will be immutable (can't be overwritten). Using wildcards (*) will match zero or more image tag characters.' There is a text input field and a 'Add filter' button.

DAY 8

DATE : 26/22/25

DAY : FRIDAY

VPC > Your VPCs > vpc-016d5961013bac5b5

/pc-016d5961013bac5b5 / deployment-project-VPC-vpc

Actions

Details [Info](#)

VPC ID vpc-016d5961013bac5b5	State Available	Block Public Access <input type="radio"/> Off	DNS hostnames Enabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-0fce1dd5bf2de2a0f	Main route table rtb-00aeecd324ff6fad71
Main network ACL acl-0173bff4707c7a011	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 573247409194
Encryption control ID —	Encryption control mode —		

Resource map [CIDRs](#) [Flow logs](#) [Tags](#) [Integrations](#)

Resource map [Info](#)

VPC [CloudWatch Metrics](#) [Logs](#) [Metrics](#) [Metrics Insights](#) [CloudWatch Metrics](#) [Logs](#) [Metrics](#) [Metrics Insights](#)

Subnets (4) [CloudWatch Metrics](#) [Logs](#) [Metrics](#) [Metrics Insights](#)

Route tables (4) [CloudWatch Metrics](#) [Logs](#) [Metrics](#) [Metrics Insights](#)

Show all details

IAM > Roles > ecsTaskExecutionRole

ecsTaskExecutionRole [Info](#)

Edit [Delete](#)

Summary

Creation date December 26, 2025, 11:13 (UTC+05:30)	ARN arn:aws:iam::573247409194:role/ecsTaskExecutionRole
Last activity 11 minutes ago	Maximum session duration 1 hour

Permissions [Trust relationships](#) [Tags](#) [Last Accessed](#) [Revoke sessions](#)

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Filter by Type

Search	All types
Policy name	Type
AmazonECSTaskExecutionRolePolicy	AWS managed

[Simulate](#) [Remove](#) [Add permissions](#)

IAM > Users

Users (1) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Create user

Search

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access keys
mohi	/	0	1 hour ago	-	-	-	Act

us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/details/mohi?section=permissions

mohi Info

Summary

ARN arn:aws:iam::573247409194:user/mohi	Console access Disabled	Access key 1 AKIAKYK6BVJQVEVUYTCPP - Active <small>Never used. Created today.</small>
Created December 26, 2025, 10:30 (UTC+05:30)	Last console sign-in -	Access key 2 Create access key

Permissions **Groups** **Tags (1)** **Security credentials** **Last Accessed**

Permissions policies (3)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type

Policy name	Type	Attached via
AmazonEC2ContainerServiceTaskExecutionRole	AWS managed	Directly

Amazon Elastic Container Service

- Clusters
- Namespaces
- Task definitions
- Account settings
- Amazon ECR
- Repositories
- AWS Batch
- Documentation
- Discover products
- Subscriptions

frontend-aitech-task123-service-69jjvto1 Info

Last updated December 26, 2025, 13:02 (UTC+5:30) **Update service**

Service overview Info

Status Active	Tasks (1 Desired) 0 Pending 1 Running	Task definition: revision frontend-aitech-task123:2	Deployment status Success
----------------------	--	--	----------------------------------

Deployments

Last deployment Info

Deployment ID cxbCBZ5cNQdU20OZdUq8E	Deployment status Success	Deployment controller type ECS	Deployment strategy Rolling update
Min and max running tasks	Deployment duration	Created at	Started at

Tasks

Tasks (1/1)

Status	Desired status	Task definition	Health status	Created at	Started by
Running	Running	frontend-aitech-task123:2	Unknown	26 minutes ago	ecs-svc/26867649292

Containers for task b6d4461731e24daa8edd93c71f864108

Containers (1)

Container name	Container runtime ID	Image URI	Image Digest	Status	Health
----------------	----------------------	-----------	--------------	--------	--------

SAN (5732-4740-9194) | Asia Pacific (Mumbai)

[Actions](#)

frontend-aitech-tg

Details

arn:aws:elasticloadbalancing:ap-south-1:573247409194:targetgroup/frontend-aitech-tg/c2549c2721e3295b

Target type IP	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-016d5961013bac5b5
IP address type IPv4	Load balancer frontend-aitech-alb123		
1 Total targets	0 Healthy	0 Unhealthy	0 Unused
	0 Anomalous		1 Initial
			1 Draining

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets | Monitoring | Health checks | Attributes | Tags

SAN (5732-4740-9194) | Asia Pacific (Mumbai)

[Actions](#) | [Create load balancer](#)

Load balancers (1/1) [What's new?](#)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	State	Type	Scheme	IP address type	VPC ID	Availability Zon
frontend-aitech-alb123	Active	application	Internet-facing	IPv4	vpc-016d5961013bac5b5	2 Availability Zo

Load balancer: frontend-aitech-alb123

Internet-facing | ZP97RAFLXTNZK | December 26, 2025, 11:53 (UTC+05:30)

Load balancer ARN
[arn:aws:elasticloadbalancing:ap-south-1:573247409194:loadbalancer/app/fronten](#)
[d-aitech-alb123/614ece52b2e11f94](#)

DNS name copied | [frontend-aitech-alb123-904489280.ap-south-1.elb.amazonaws.com \(A Record\)](#)

```
=> [3/3] COPY frontend-aitech/ /user/share/nginx/html/          0.5s
=> exporting to image                                         2.5s
=> => exporting layers                                         2.0s
=> => exporting manifest sha256:bd37d93f17ec3c2b10666808c3f6  0.0s
=> => exporting config sha256:4fb2bf50dfddlc0e15e634b64810f  0.0s
=> => exporting attestation manifest sha256:odd4d9f8fb6f8f99  0.0s
=> => exporting manifest list sha256:676aae5e3962d8b2ab38cbe 0.0s
=> => naming to docker.io/library/frontend-aitech:latest      0.0s
=> => unpacking to docker.io/library/frontend-aitech:latest    0.4s
root@ip-10-0-13-115:/home/ubuntu/my-website# docker tag frontend-aitech:latest 573247409194.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest
root@ip-10-0-13-115:/home/ubuntu/my-website# docker push 573247409194.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech:latest
The push refers to repository [573247409194.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech]
b3d98ac97eb5: Pushed
105bf4a22968: Pushed
1074353eec0d: Layer already exists
567f84da6fb0: Layer already exists
da7c973d0b92: Layer already exists
a9c65653a275: Layer already exists
085c5e5aaa8e: Layer already exists
25f453064fd3: Layer already exists
33f95a0f3229: Layer already exists
de54cb021236: Layer already exists
0abf9e567266: Layer already exists
latest: digest: sha256:676aae5e3962d8b2ab38cbe9815a07d04ca2800b6fce07081b9c00d6c911195b size: 856
```

i-0104ed108f0b557e8 (deployment-project)

PublicIPs: 13.201.1.147 PrivateIPs: 10.0.13.115

REST API

```
Last login: Fri Dec 26 08:30:11 2025 from 13.233.177.4
ubuntu@ip-10-0-15-89:~$ sudo su
root@ip-10-0-15-89:/home/ubuntu# cd ~/rest-api
bash: cd: /root/rest-api: No such file or directory
root@ip-10-0-15-89:/home/ubuntu# mkdir rest-api
cd rest-api
mkdir: cannot create directory 'rest-api': File exists
root@ip-10-0-15-89:/home/ubuntu/rest-api# aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin 573247409194.dkr.ecr.ap-south-1.amazonaws.com/frontend-aitech

Command 'aws' not found, but can be installed with:
snap install aws-cli  # version 1.44.6, or
apt install awscli   # version 2.14.6-1
See 'snap info aws-cli' for additional versions.
error: cannot perform an interactive login from a non TTY device
root@ip-10-0-15-89:/home/ubuntu/rest-api#
```

i-09abe84cf35fa6a9d (webstack)

Public IPs: 13.126.196.163 Private IPs: 10.0.15.89

```

inflating: aws/dist/prompt_toolkit-3.0.31.dist-info/licenses/LICENSE
NSE
inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
You can now run: /usr/local/bin/aws --version
root@ip-10-0-15-89:/tmp# aws --version
aws-cli/2.32.23 Python/3.13.11 Linux/6.14.0-1015-aws exe/x86_64.ubuntu.24
root@ip-10-0-15-89:/tmp# aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin 573247409194.dkr.ecr.ap-south-1.amazonaws.com

Unable to locate credentials. You can configure credentials by running "aws login".
error: cannot perform an interactive login from a non TTY device
root@ip-10-0-15-89:/tmp#

```

Task definition successfully created
rest-api-task:1 has been successfully created. You can use this task definition to deploy a service or run a task.

rest-api-task:1

Last updated December 26, 2025, 15:48 (UTC+5:30)

Overview [Info](#)

ARN arn:aws:ecs:ap-south-1:573247409194:task-definition/rest-api-task:1	Status ACTIVE	Time created December 26, 2025, 15:48 (UTC+5:30)	App environment Fargate
Task role -	Task execution role ecsTaskExecutionRole	Operating system/Architecture Linux/X86_64	Network mode awsvpc
Fault injection Turned off			

Containers [JSON](#) [Task placement](#) [Volumes \(0\)](#) [Requires attributes](#) [Tags](#)

Task size

Task CPU	Task memory
CloudShell	Feedback
Console Mobile App	

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Amazon Elastic Container Service > Clusters > rest-api > Services

December 26, 2025, 16:40 (UTC+5:30)

Amazon Elastic Container Service

Express Mode

Clusters

Namespaces

Task definitions

Account settings

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

Tell us what you think

Cluster overview

ARN arn:aws:ecs:ap-south-1:347363494 863:cluster/rest-api	Status Active	CloudWatch monitoring Default	Registered container instances -
--	-------------------------	---	--

Services

Draining -	Active 1	Pending -	Running 1
----------------------	--------------------	---------------------	---------------------

Tasks

Services | **Tasks** | **Infrastructure** | **Metrics** | **Scheduled tasks** | **Configuration** | **Event history** | **Tags**

Services (1) Info

Last updated December 26, 2025, 16:40 (UTC+5:30)

Manage tags | Update | Delete service | Create

Filter services by value

Filter launch type: Any launch type

Filter scheduling strategy: Any scheduling strategy

Filter resource management type: Any resource management type

Service name	ARN	Status	Schedu...	L...	Task de...	Deployments and tasks
rest-api-backend-service-pim0j9d9	arn:aws:ecs:s...	Active	REPLICA	-	rest-api-b...	1/1 Tasks ru...