EC2 AUTO SCALING AND LOAD BALANCING

CLOUD PROJECT REPORT

Presented by: Vanshika Munjal & Vaani Jindal (Group 37)
500121784 500119144

Batch: B2 CCVT

INTRODUCTION

Cloud Computing Overview:

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, organizations can access computing services—such as servers, storage, databases, networking, software, and analytics—from a cloud provider.

Amazon Web Services (AWS) Overview:

Amazon Web Services (AWS) is the world's most comprehensive and widely adopted cloud platform. It offers over 200 fully featured services, including computing, storage, networking, machine learning, security, and analytics.

PURPOSE OF PROJECT

This project focuses on building a scalable and highly available system using AWS services. By implementing EC2 Auto Scaling, we ensure that our application automatically adjusts the number of instances based on demand. Load Balancing, on the other hand, distributes incoming traffic efficiently across multiple EC2 instances, improving performance and reliability. Together, these features help build a fault-tolerant and cost-effective cloud infrastructure.

PROJECT GOALS

The goal of this project is to design and deploy a scalable, resilient, and cost-efficient cloud infrastructure using AWS EC2 Auto Scaling and Elastic Load Balancing. This system is intended to handle unpredictable traffic patterns by automatically adjusting compute capacity, ensuring high availability, fault tolerance, and optimal performance at all times.

Through this project, we aim to demonstrate how cloud-native technologies can eliminate manual resource management, reduce downtime, and support real-world applications with dynamic workloads effectively.

PROJECT OBJECTIVES

- Implement EC2 Auto Scaling to automatically add or remove instances based on demand.
- Integrate IAM, S3, EBS, and Lambda
- Configure an Elastic Load Balancer (ELB) to distribute incoming traffic evenly across instances.
- Optimize resource usage and cost by scaling the infrastructure dynamically.
- Monitor system performance using AWS tools like CloudWatch for real-time insights.
- Ensure best security practices using IAM roles and encryption

AWS SERVICES USED

IAM (Identity and Access Management) Roles & Permissions setup:
To ensure secure access to AWS services, Identity and Access Management (IAM) was configured as follows:

- Created IAM Group: EC2Group with AmazonEC2FullAccess policy.
- Created IAM User: Assigned to EC2Group, enabling programmatic access with generated Access Key and Secret Key.
- IAM Role for EC2 (EC2AutoScalingRole):
- Policy Attached: AmazonEC2RoleforSSM and AmazonS3FullAccess.

This role was associated with the EC2 Launch Template to allow EC2 instances to access S3 and use SSM for remote access.

EC2 Instance Configurations:
Amazon EC2 was used to launch virtual servers with the following setup:

- Launch Template: Named MyAutoScaleTemplate with:
- AMI: Amazon Linux 2
- Instance type: t2.micro
- Key pair: Used for SSH access
- Security Group: Allows SSH (port 22) and HTTP (port 80)
- IAM instance profile: EC2AutoScalingRole
- Auto Scaling Group:
- Min Size: 1, Max Size: 5, Desired Capacity: 2
- Subnets used: subnet-09aa416d352d8792e, subnet-081ae8c6100435c9a, subnet-0e68c9beb9282a1ff
- Scaling policy triggered by CPU utilization

AWS SERVICES USED

S3 Bucket Settings and EBS Attachment:

S3 Bucket:

- Created using CLI
- Used for file storage and syncing files with EC2 instances (aws s3 sync, cp, rm commands)
- Versioning and lifecycle rules were considered for optimization.

EBS Volume:

- Created and attached to EC2 instances
- Mounted to /data directory after formatting with mkfs
- Commands used: aws ec2 create-volume, attach-volume, followed by SSH-based Linux commands to mount and persist.

Lambda Function:

- Function Name: EC2CleanupFunction
- Purpose: Automatically stops/terminates idle EC2 instances based on CloudWatch Events
- Runtime: Python 3.x
- Trigger: Scheduled EventBridge rule EC2CleanupRule (e.g., every 1 hour)

Permissions:

- Lambda permission added for EventBridge to invoke it (lambda:InvokeFunction)
- IAM role with ec2:DescribeInstances, ec2:StopInstances, and logs:* permissions

IMPLEMENTATION DETAILS AND CODE SNIPPETS

Configuring AWS

```
C:\Users\VAANI JINDAL>aws configure
AWS Access Key ID [****************VKVW]: AKIA5G2VG5EDEAD2FZ6P
AWS Secret Access Key [****************QQVg]: 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
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
```

```
C:\Users\VAANI JINDAL>aws sts get-caller-identity
{
    "UserId": "AIDA5G2VG5EDLGUMCLZ7I",
    "Account": "908027422982",
    "Arn": "arn:aws:iam::908027422982:user/user_1"
}
```

The command aws sts get-caller-identity is used to retrieve details about the AWS identity making the request. It helps you confirm which AWS account, user, or role is being used when executing AWS CLI commands.

```
C:\Users\VAANI JINDAL>notepad trust-policy.json

C:\Users\VAANI JINDAL>type trust-policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

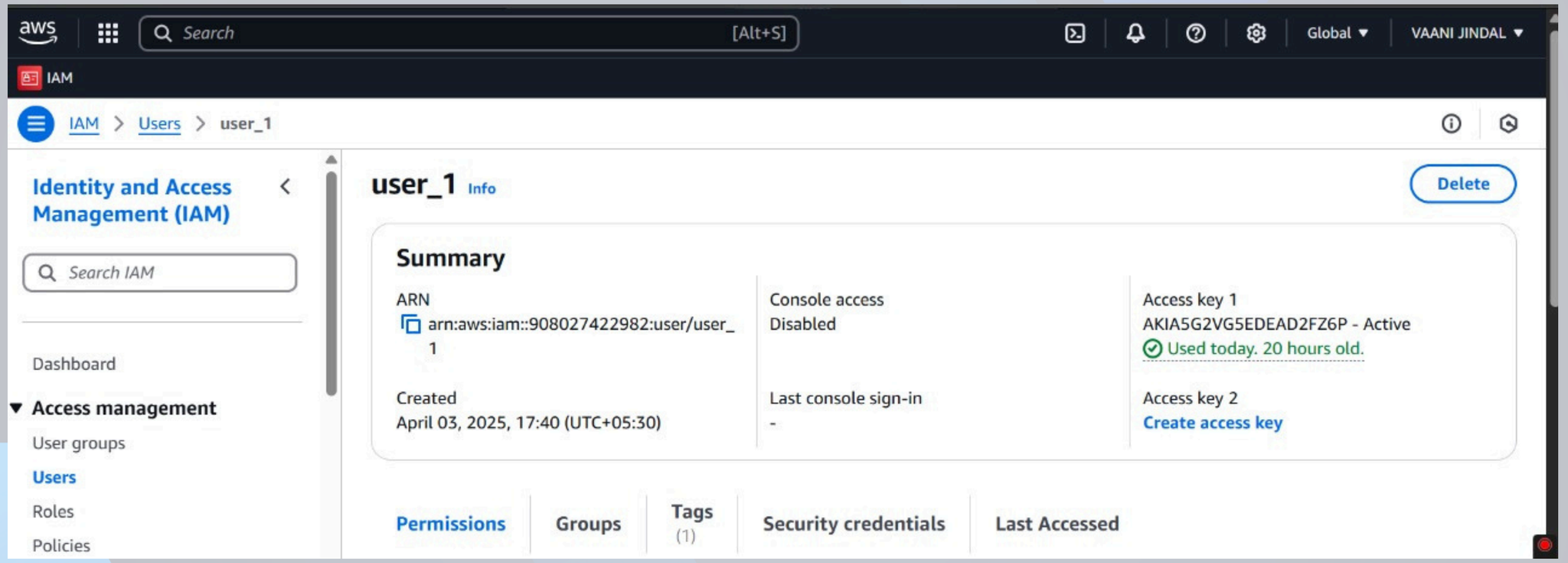
The trust-policy.json file is created to define the Trust Policy for an IAM Role in AWS.

- notepad trust-policy.json ..... created json file
- type trust-policy.json .....confirmation that created successfully

Setting up IAM Role

```
C:\Users\VAANI JINDAL>aws iam create-role --role-name EC2AutoScalingRole  --assume-role-policy-document file://"C:\\Users\\VAANI JINDAL\\OneDrive\\Desktop\\trust-policy.json"
{
    "Role": {
        "Path": "/",
        "RoleName": "EC2AutoScalingRole",
        "RoleId": "AROA5G2VG5EDPAZTDQ7YK",
        "Arn": "arn:aws:iam::908027422982:role/EC2AutoScalingRole",
        "CreateDate": "2025-04-03T14:55:06+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

IAM user_1


aws
Search
[Alt+S]
Global
VAANI JINDAL
IAM
IAM > Users > user_1
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
user_1 Info
Delete
Summary
ARN
arn:aws:iam::908027422982:user/user_1
Console access
Disabled
Access key 1
AKIA5G2VG5EDEAD2FZ6P - Active
Used today. 20 hours old.
Created
April 03, 2025, 17:40 (UTC+05:30)
Last console sign-in
-
Access key 2
Create access key
Permissions
Groups
Tags (1)
Security credentials
Last Accessed

Attaching Required Policies to IAM Role

```
C:\Users\VAANI JINDAL>aws iam attach-role-policy --role-name EC2AutoScalingRole --policy-arn arn:aws:iam::aws:policy/
AmazonEC2FullAccess

C:\Users\VAANI JINDAL>aws iam attach-role-policy --role-name EC2AutoScalingRole --policy-arn arn:aws:iam::aws:policy/
service-role/AWSLambdaBasicExecutionRole

C:\Users\VAANI JINDAL>aws iam attach-role-policy --role-name EC2AutoScalingRole --policy-arn arn:aws:iam::aws:policy/
AmazonS3FullAccess
```

Verification of the attached policies

```
C:\Users\VAANI JINDAL>aws iam list-attached-role-policies --role-name EC2AutoScalingRole
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonEC2FullAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
        },
        {
            "PolicyName": "AWSLambdaBasicExecutionRole",
            "PolicyArn": "arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole"
        },
        {
            "PolicyName": "AmazonS3FullAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
        }
    ]
}
```

Launch EC2 Instance Process

```
C:\Users\VAANI JINDAL> aws ec2 create-key-pair --key-name MyEC2Key --query "KeyMaterial" --output text > MyEC2Key.pem

C:\Users\VAANI JINDAL>chmod 400 MyEC2Key.pem
'chmod' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\VAANI JINDAL>icacls MyEC2Key.pem /inheritance:r
processed file: MyEC2Key.pem
Successfully processed 1 files; Failed processing 0 files

C:\Users\VAANI JINDAL>icacls MyEC2Key.pem /grant:r "%username%:F"
processed file: MyEC2Key.pem
Successfully processed 1 files; Failed processing 0 files

C:\Users\VAANI JINDAL>icacls MyEC2Key.pem /deny "NT AUTHORITY\SYSTEM:(F)"
processed file: MyEC2Key.pem
Successfully processed 1 files; Failed processing 0 files
```

- Setting up key-pair
- Remove inherited permissions (so only your user can access it)
- Grant full access only to your current user
- Deny access to the SYSTEM account (extra security)

Now the .pem file is secured and we can use it to connect to ourEC2 instance!

```
C:\Users\VAANI JINDAL>dir MyEC2Key.pem
 Volume in drive C is Windows
 Volume Serial Number is 3E9D-5BFC

 Directory of C:\Users\VAANI JINDAL

03-04-2025  23:38             1,702 MyEC2Key.pem
               1 File(s)          1,702 bytes
               0 Dir(s)  13,773,385,728 bytes free
```

Running this command in cmd to verify if .pem file exists

Created a Security Group

A Security Group is required to allow inbound and outbound traffic to your EC2 instance. This creates a security group named MySecurityGroup.

```
C:\Users\VAANI JINDAL>aws ec2 create-security-group --group-name MySecurityGroup --description "Security group for EC
2"
{
    "GroupId": "sg-0f4320053ede4b289",
    "SecurityGroupArn": "arn:aws:ec2:ap-south-1:908027422982:security-group/sg-0f4320053ede4b289"
}
```

```
C:\Users\VAANI JINDAL>aws ec2 authorize-security-group-ingress --group-name MySecurityGroup --protocol tcp --port 22
--cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-044ffd45a47f6167e",
            "GroupId": "sg-0f4320053ede4b289",
            "GroupOwnerId": "908027422982",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:ap-south-1:908027422982:security-group-rule/sgr-044ffd45a47f6167e"
        }
    ]
}


C:\Users\VAANI JINDAL>aws ec2 authorize-security-group-ingress --group-name MySecurityGroup --protocol tcp --port 80
--cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-032b21b7aad4aa1b3",
```

This allows you to connect via SSH from any IP.
Now, your EC2 can be accessed via SSH and HTTP

To get latest amazon Linux 2 AMI Id

```
C:\Users\VAANI JINDAL>aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2"
 --region ap-south-1 --query "Images | sort_by(@, &CreationDate)[-1].ImageId" --output text
ami-0019610b50e1fe6bb
```

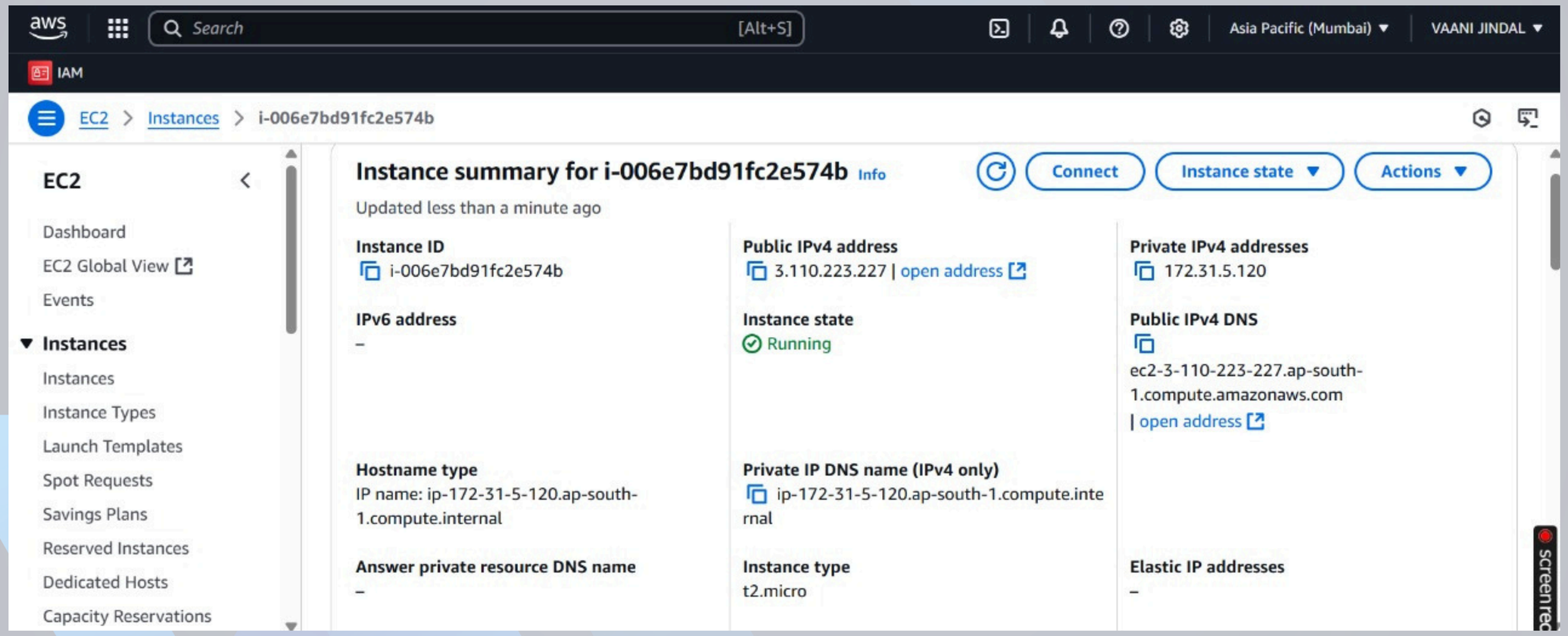
Launch EC2 Instance
(region ap-south-1)

```
C:\Users\VAANI JINDAL>aws ec2 run-instances --image-id ami-0019610b50e1fe6bb --instance-type t2.micro --key-name MyEC
2Key --security-group-ids sg-0f4320053ede4b289 --count 1 --region ap-south-1
{
    "ReservationId": "r-034bca6995523fe53",
    "OwnerId": "908027422982",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "d2fc96a0-b0e6-4005-88cc-831ed17fa06f",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-04-03T18:50:02+00:00",
                        "AttachmentId": "eni-attach-0e7098b1eecf369cf",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
```

Connect to Your EC2 Instance

```
C:\Users\VAANI JINDAL>aws ec2 describe-instances --query "Reservations[].Instances[].[InstanceId,PublicIpAddress]" --
output table
----------------------------------------
|           DescribeInstances          |
+---------------------+----------------+
|  i-069bf6155cdbe2690 |  None          |
|  i-006e7bd91fc2e574b |  3.110.223.227 |
+---------------------+----------------+
```

EC2 Instance



Verify public ip

```
C:\Users\VAANI JINDAL>aws ec2 describe-instances --instance-ids i-006e7bd91fc2e574b --query "Reservations[].Instances
[].[State.Name, PublicIpAddress]" --output table
-----------------------------
|      DescribeInstances     |
+----------+-----------------+
|  running |  3.110.223.227  |
+----------+-----------------+
```

To connect

```
C:\Users\VAANI JINDAL>ssh -i "MyEC2Key.pem" ec2-user@3.110.223.227
The authenticity of host '3.110.223.227 (3.110.223.227)' can't be established.
ED25519 key fingerprint is SHA256:K7tUsUqKOF04np937kwBHTSHzWlq0YuSM4PBseBZZrw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.110.223.227' (ED25519) to the list of known hosts.
          #
```

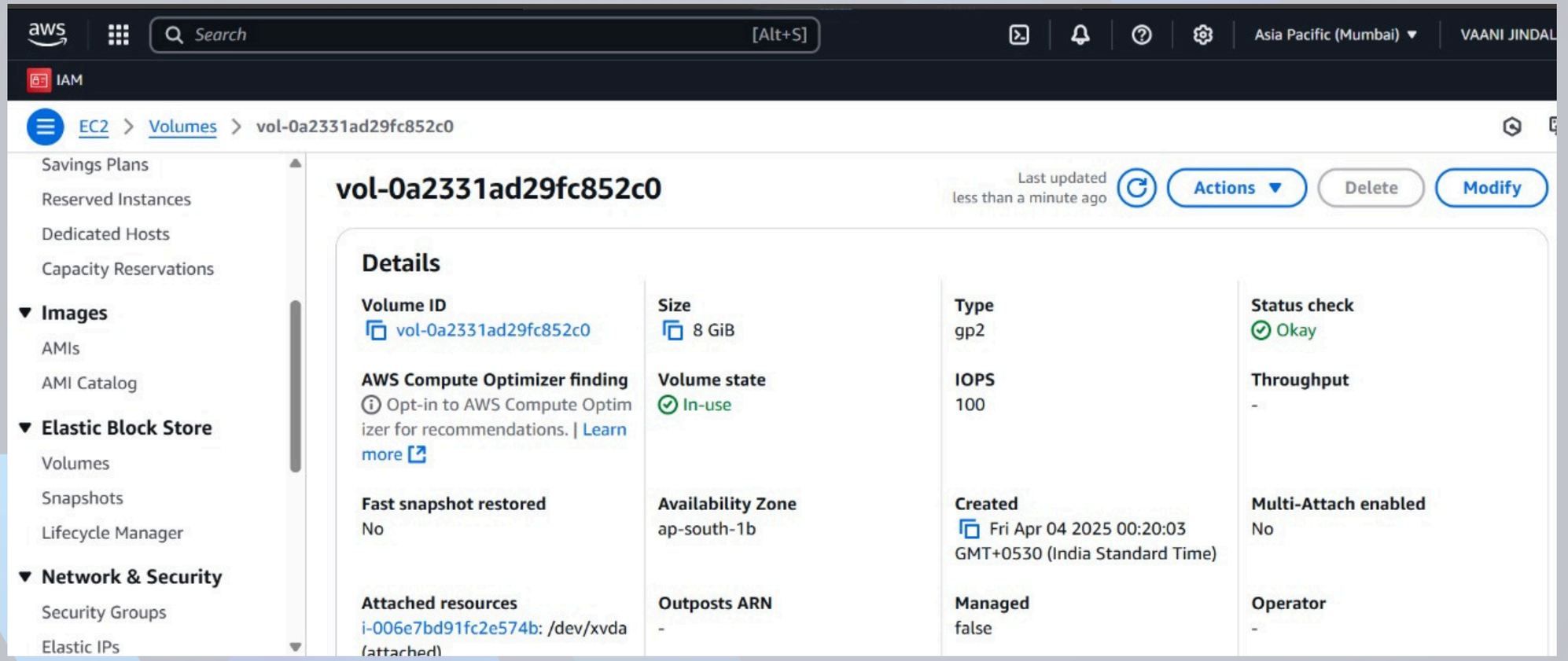
AWS configure and then creating EBS volume

```
[ec2-user@ip-172-31-5-120 ~]$ aws configure
AWS Access Key ID [None]: AKIA5G2VG5EDEAD2FZ6P
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
[ec2-user@ip-172-31-5-120 ~]$ aws ec2 create-volume --availability-zone ap-south-1b --size 10 --volume-t
ype gp2 --reg
ion ap-south-1
{
    "AvailabilityZone": "ap-south-1b",
    "MultiAttachEnabled": false,
    "Tags": [],
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-0b286b8328c3fae22",
    "State": "creating",
    "Iops": 100,
    "SnapshotId": "",
    "CreateTime": "2025-04-03T19:30:42.000Z",
    "Size": 10
}
```

Attaching the EBS Volume

```
[ec2-user@ip-172-31-5-120 ~]$ aws ec2 attach-volume --volume-id vol-0b286b8328c3fae22 --instance-id i-00
6e7bd91fc2e574b --device /dev/xvdf --region ap-south-1
{
    "AttachTime": "2025-04-03T19:32:45.219Z",
    "InstanceId": "i-006e7bd91fc2e574b",
    "VolumeId": "vol-0b286b8328c3fae22",
    "State": "attaching",
    "Device": "/dev/xvdf"
}
```

EBS volume


aws
Search
[Alt+S]
Asia Pacific (Mumbai)
VAANI JINDAL
IAM
EC2 > Volumes > vol-0a2331ad29fc852c0
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
vol-0a2331ad29fc852c0
Last updated
less than a minute ago
Actions
Delete
Modify
Details
Volume ID
vol-0a2331ad29fc852c0
Size
8 GiB
Type
gp2
Status check
Okay
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
Volume state
In-use
IOPS
100
Throughput
-
Fast snapshot restored
No
Availability Zone
ap-south-1b
Created
Fri Apr 04 2025 00:20:03 GMT+0530 (India Standard Time)
Multi-Attach enabled
No
Attached resources
i-006e7bd91fc2e574b: /dev/xvda (attached)
Outposts ARN
-
Managed
false
Operator
-

To enable automatic backup

```
[ec2-user@ip-172-31-5-120 ~]$ aws ec2 create-snapshot --volume-id vol-0b286b8328c3fae22 --description "A
utomatic backup"
{
    "Description": "Automatic backup",
    "Tags": [],
    "Encrypted": false,
    "VolumeId": "vol-0b286b8328c3fae22",
    "State": "pending",
    "VolumeSize": 10,
    "StartTime": "2025-04-03T20:12:25.361Z",
    "Progress": "",
    "OwnerId": "908027422982",
    "SnapshotId": "snap-0951794db846f927a"
}
```

To schedule backups automatically

```
[ec2-user@ip-172-31-5-120 ~]$ aws backup create-backup-vault --backup-vault-name MyBackupVault
{
    "BackupVaultArn": "arn:aws:backup:ap-south-1:908027422982:backup-vault:MyBackupVault",
    "CreationDate": 1743711954.947,
    "BackupVaultName": "MyBackupVault"
}
[ec2-user@ip-172-31-5-120 ~]$ aws backup create-backup-plan --backup-plan '{"BackupPlanName":"MyBackupPl
an","Rules": [{"RuleName":"DailyBackup","TargetBackupVaultName":"MyBackupVault","ScheduleExpression":"cr
on(0 12 * * ? *)","StartWindowMinutes":60,"CompletionWindowMinutes":120}]}'
{
    "BackupPlanArn": "arn:aws:backup:ap-south-1:908027422982:backup-plan:e4086bd7-482e-43f5-9e08-41e5a77
3cba4",
    "VersionId": "M2JlOGI1ZDktNzEzZC00ZDQ5LWE0YzQtNzhlYjJkM2FiZmYz",
    "CreationDate": 1743712036.105,
    "BackupPlanId": "e4086bd7-482e-43f5-9e08-41e5a773cba4"
```

These permissions attached to iam user

- AWSBackupFullAccess
- AWSBackupOperatorAccess

To schedule backups automatically: AWS Backup stores all backups in a Backup Vault. By default, AWS creates a vault named "Default", but in our case, it might not exist.

command 1 for Create a Backup Vault
command 2 for To schedule backups:

CREATING S3 BUCKET

```
. Please try again.
[ec2-user@ip-172-31-5-120 ~]$ aws s3 mb s3://autoscaling-bucket-new1 --region ap-south-1
make_bucket: autoscaling-bucket-new1
```

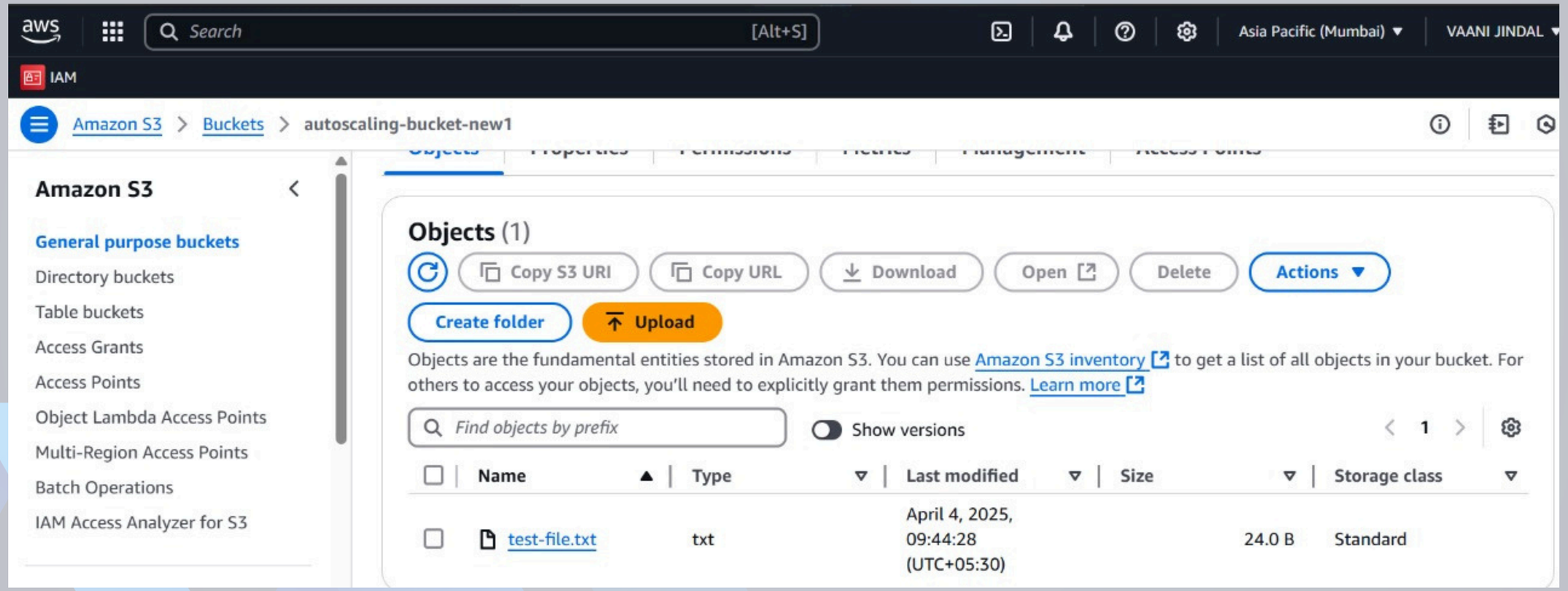
Cd command

```
C:\Users\VAANI JINDAL\OneDrive\Desktop>cd C:\Users\VAANI JINDAL\OneDrive\Desktop
```

Upload a File to S3 using Python

```
C:\Users\VAANI JINDAL\OneDrive\Desktop>python upload_s3.py
File C:\Users\VAANI JINDAL\OneDrive\Desktop\testfile.txt uploaded to S3 bucket autoscaling-bu
cket-new1 as test-file.txt
```

S3 BUCKET


aws
Search
[Alt+S]
Asia Pacific (Mumbai)
VAANI JINDAL
IAM
Amazon S3 > Buckets > autoscaling-bucket-new1
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
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
Find objects by prefix
Show versions
1
Name
Type
Last modified
Size
Storage class
test-file.txt
txt
April 4, 2025, 09:44:28 (UTC+05:30)
24.0 B
Standard

Upload a File to S3 using Python

upload_s3.py - C:/Users/VAANI JINDAL/OneDrive/Desktop/upload_s3.py (3.13.2)

File Edit Format Run Options Window Help

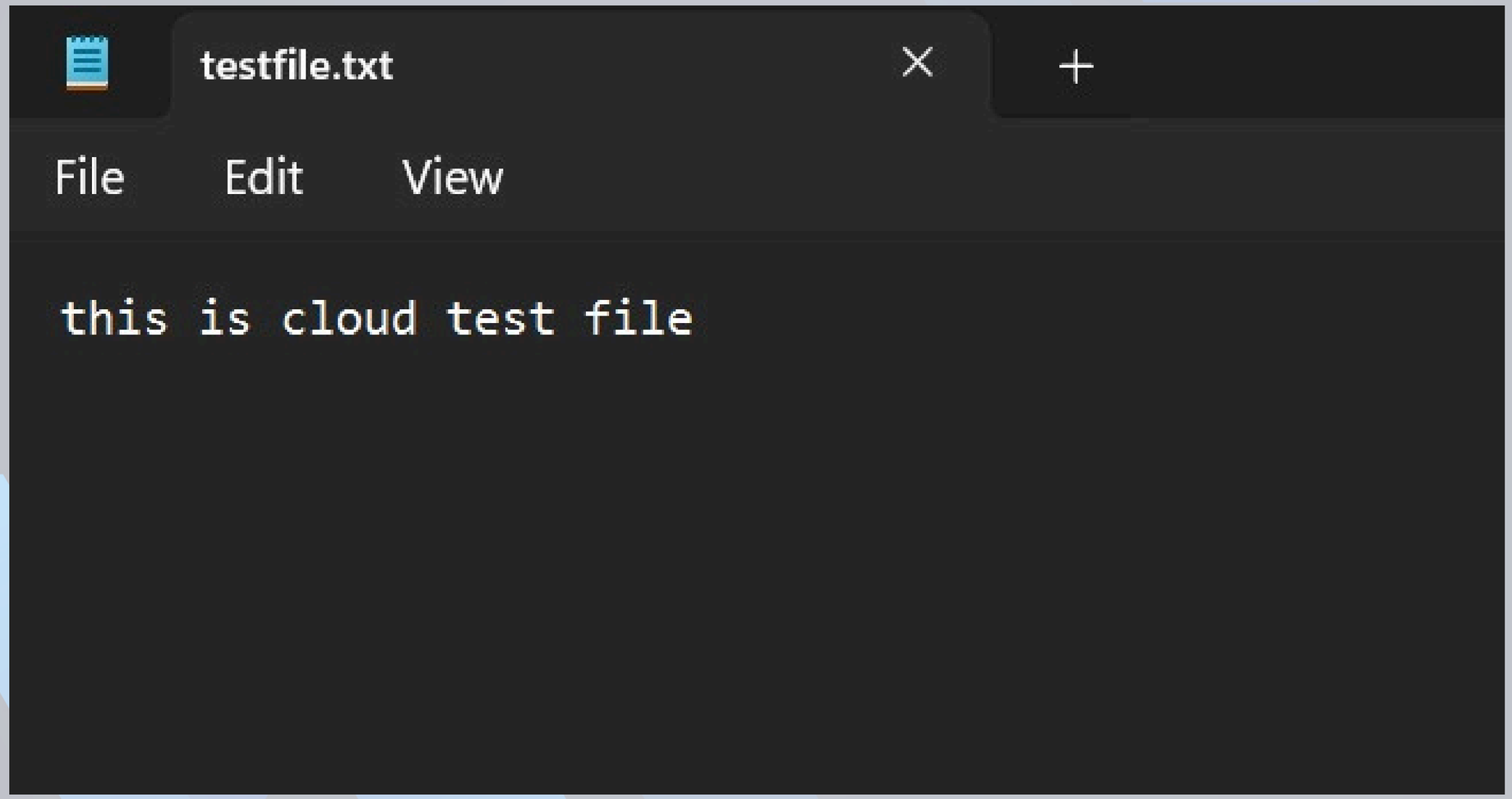
```
import boto3

# Initialize S3 client
s3 = boto3.client('s3')

bucket_name = "autoscaling-bucket-new1"
file_name = "C:\\Users\\VAANI JINDAL\\OneDrive\\Desktop\\testfile.txt"  # Update
object_name = "test-file.txt"  # Name in S3 bucket

# Upload file
s3.upload_file(file_name, bucket_name, object_name)
print(f"File {file_name} uploaded to S3 bucket {bucket_name} as {object_name}")
```

File uploaded to S3 Bucket (testfile.txt)



Download a File from S3 using Python

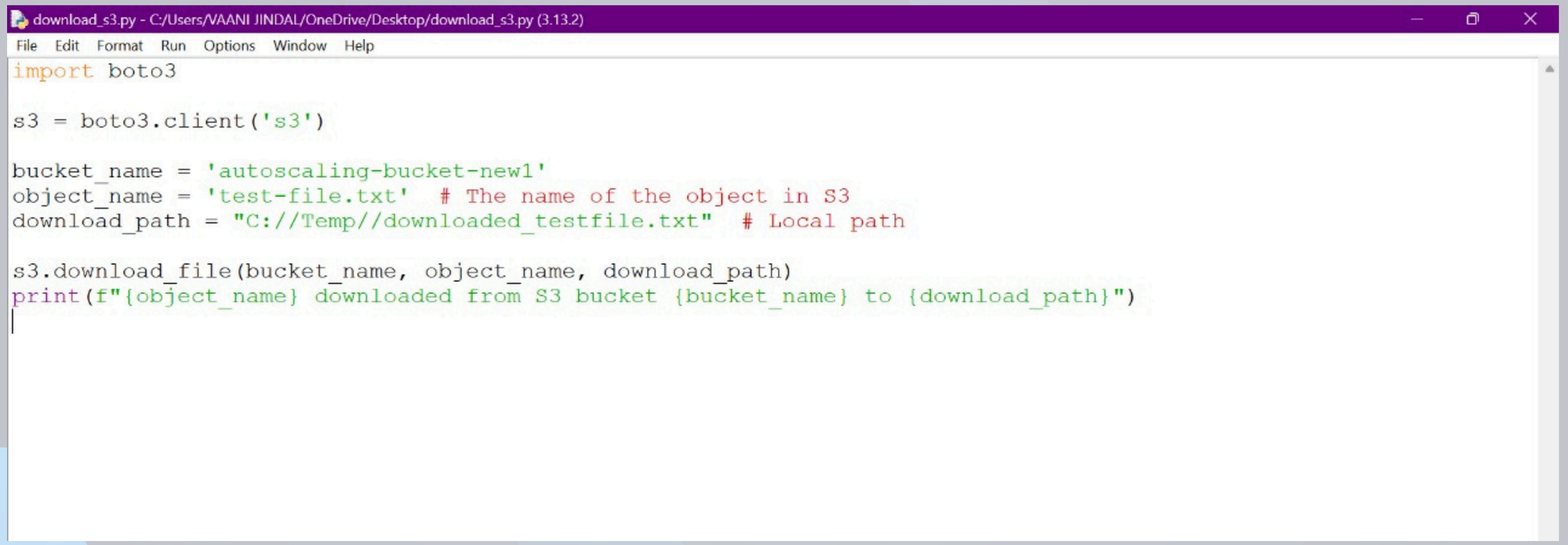
```
C:\Users\VAANI JINDAL\OneDrive\Desktop>mkdir C:\Temp

C:\Users\VAANI JINDAL\OneDrive\Desktop>mkdir C:\Temp
A subdirectory or file C:\Temp already exists.

C:\Users\VAANI JINDAL\OneDrive\Desktop>python download_s3.py
test-file.txt downloaded from S3 bucket autoscaling-bucket-new1 to C://Temp//downloaded_testf
ile.txt
```

Made a directory where to download the file

Download a File from S3 using Python



Enable S3 Bucket Versioning or Lifecycle Policies

To ensure data retention and automatic deletion of old files, you should enable S3 versioning and lifecycle policies.

Enabling S3 Bucket Versioning

```
C:\Users\VAANI JINDAL\OneDrive\Desktop>aws s3api put-bucket-versioning --bucket autoscaling-b
ucket-new1 --versioning-configuration Status=Enabled

C:\Users\VAANI JINDAL\OneDrive\Desktop>aws s3api get-bucket-versioning --bucket autoscaling-b
ucket-new1
{
    "Status": "Enabled"
}
```

Lifecycle Policies

```
C:\Users\VAANI JINDAL\OneDrive\Desktop>aws s3api put-bucket-lifecycle-configuration --bucket
autoscaling-bucket-new1 --lifecycle-configuration file://lifecycle_policy.json
{
    "TransitionDefaultMinimumObjectSize": "all_storage_classes_128K"
}


C:\Users\VAANI JINDAL\OneDrive\Desktop>aws s3api get-bucket-lifecycle-configuration --bucket
autoscaling-bucket-new1
{
    "TransitionDefaultMinimumObjectSize": "all_storage_classes_128K",
    "Rules": [
        {
            "Expiration": {
                "Days": 30
            },
            "ID": "DeleteOldFiles",
            "Filter": {
                "Prefix": ""
            },
            "Status": "Enabled"
        }
    ]
```

Make lifecycle_policy .json file

Command 1: apply the fixed policy command
Command 2: verify the lifecycle policy command

SETTING UP AUTO SCALING FOR EC2 INSTANCES

Launch Template

```
C:\Users\Vanshika Munjal\Desktop>aws ec2 create-launch-template --launch-template-name MyAutoScaleTemplate --version-description "v1"
 --launch-template-data "{\"ImageId\": \"ami-0019610b50e1fe6bb\", \"InstanceType\": \"t2.micro\", \"KeyName\": \"MyNewKeypair\", \"Se
curityGroupIds\": [\"sg-0d512e764a6a47df0\"], \"IamInstanceProfile\": {\"Name\": \"EC2AutoScalingProfile\"}}"
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-08bc6716bc23b51e4",
        "LaunchTemplateName": "MyAutoScaleTemplate",
        "CreateTime": "2025-05-06T21:48:27+00:00",
        "CreatedBy": "arn:aws:iam::426285640896:user/user-autoscaling",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1,
        "Operator": {
            "Managed": false
        }
    }
}
```

```
C:\Users\Vanshika Munjal\Desktop>aws ec2 describe-launch-templates --region ap-south-1
{
    "LaunchTemplates": [
        {
            "LaunchTemplateId": "lt-08bc6716bc23b51e4",
            "LaunchTemplateName": "MyAutoScaleTemplate",
            "CreateTime": "2025-05-06T21:48:27+00:00",
            "CreatedBy": "arn:aws:iam::426285640896:user/user-autoscaling",
            "DefaultVersionNumber": 1,
            "LatestVersionNumber": 1,
            "Operator": {
                "Managed": false
            }
        }
    ]
}
```

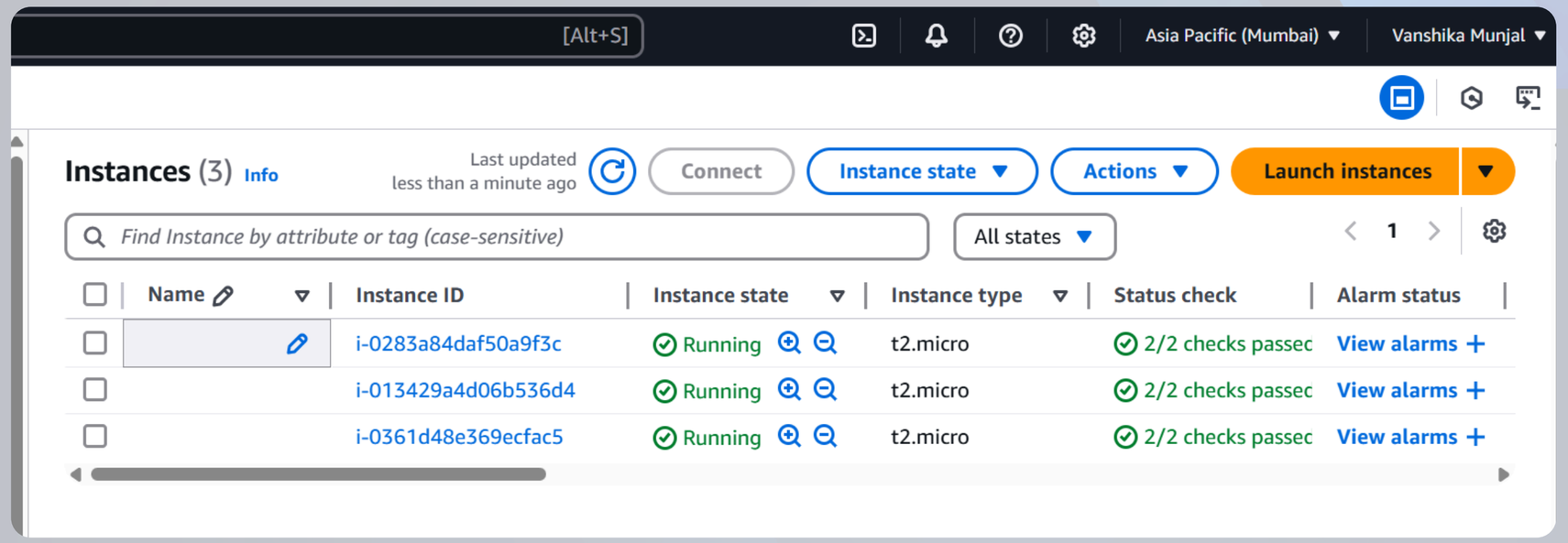
This will show all the launch templates including their names and versions

Auto Scaling Group Creation

```
C:\Users\Vanshika Munjal\Desktop>aws autoscaling create-auto-scaling-group --auto-scaling-group-name MyASG --launch-template "LaunchT
emplateName=MyAutoScaleTemplate,Version=1" --min-size 1 --max-size 5 --desired-capacity 2 --vpc-zone-identifier "subnet-09aa416d352d8
792e,subnet-081ae8c6100435c9a,subnet-0e68c9beb9282a1ff" --region ap-south-1
```


```
C:\Users\Vanshika Munjal\Desktop>aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names MyASG --region ap-south-1
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupName": "MyASG",
            "AutoScalingGroupARN": "arn:aws:autoscaling:ap-south-1:426285640896:autoScalingGroup:9064489b-98c8-441c-b80c-2a979972805d
:autoScalingGroupName/MyASG",
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-08bc6716bc23b51e4",
                "LaunchTemplateName": "MyAutoScaleTemplate",
                "Version": "1"
            },
            "MinSize": 1,
            "MaxSize": 5,
            "DesiredCapacity": 2,
            "DefaultCooldown": 300,
            "AvailabilityZones": [
                "ap-south-1c",
                "ap-south-1b",
                "ap-south-1a"
            ],
            "LoadBalancerNames": [],
            "TargetGroupARNs": [],
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": 0,
            "Instances": [
                {
                    "InstanceId": "i-013429a4d06b536d4",
                    "InstanceType": "t2.micro",
                    "AvailabilityZone": "ap-south-1a",
                    "LifecycleState": "InService",
                    "HealthStatus": "Healthy",
                    "LaunchTemplate": {
                        "LaunchTemplateId": "lt-08bc6716bc23b51e4",
```


Shows the detail of the auto scaling group, allows you to verify that everything is setup correctly.



Minimum number of EC2 instances to keep running at all times.
--max-size 5 Maximum number of EC2 instances that can be launched in the group.
--desired-capacity 2 Number of EC2 instances to launch initially. The Auto Scaling Group will try to maintain this number unless scaling policies change it.

```
C:\Users\Vanshika Munjal\Desktop>aws ec2 describe-vpcs --region ap-south-1
{
    "Vpcs": [
        {
            "OwnerId": "426285640896",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0be3d6771b3dd6738",
                    "CidrBlock": "172.31.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": true,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-08a94d26caf441f1f",
            "State": "available",
            "CidrBlock": "172.31.0.0/16",
            "DhcpOptionsId": "dopt-03b17e549da394ca1"
        }
    ]
}
```

This command will return a list of all VPCs in ap-south-1 region including their IDs

Create a Target Group

```
C:\Users\Vanshika Munjal\Desktop>aws elbv2 create-target-group --name MyTargetGroup --protocol HTTP --port 80 --vpc-id vpc-08a94d26ca
f441f1f --region ap-south-1
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:ap-south-1:426285640896:targetgroup/MyTargetGroup/6df82defd1ccdb7e",
            "TargetGroupName": "MyTargetGroup",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-08a94d26caf441f1f",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}
```

LOAD BALANCER

Attaching Load Balancer

```
C:\Users\Vanshika Munjal\Desktop>aws elbv2 create-load-balancer ^ --name MyAppLoadBalancer ^ --subnets subnet-0e68c9beb9282a1ff subne
t-09aa416d352d8792e subnet-081ae8c6100435c9a ^ --security-groups sg-0d512e764a6a47df0 ^ --scheme internet-facing ^ --type application
 ^ --ip-address-type ipv4 ^ --region ap-south-1
{
    "LoadBalancers": [
        {
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:ap-south-1:426285640896:loadbalancer/app/MyAppLoadBalancer/b4eda167e5251
3da",
            "DNSName": "MyAppLoadBalancer-1054520609.ap-south-1.elb.amazonaws.com",
            "CanonicalHostedZoneId": "ZP97RAFLXTNZK",
            "CreatedTime": "2025-05-06T22:14:26.780000+00:00",
            "LoadBalancerName": "MyAppLoadBalancer",
            "Scheme": "internet-facing",
            "VpcId": "vpc-08a94d26caf441f1f",
            "State": {
                "Code": "provisioning"
            },
            "Type": "application",
            "AvailabilityZones": [
                {
                    "ZoneName": "ap-south-1c",
                    "SubnetId": "subnet-0e68c9beb9282a1ff",
                    "LoadBalancerAddresses": []
                },
                {
                    "ZoneName": "ap-south-1a",
                    "SubnetId": "subnet-09aa416d352d8792e",
                    "LoadBalancerAddresses": []
                },
                {
                    "ZoneName": "ap-south-1b",
                    "SubnetId": "subnet-081ae8c6100435c9a",
                    "LoadBalancerAddresses": []
                }
            ],
            "SecurityGroups": [
```

Creating a listener

```
C:\Users\Vanshika Munjal\Desktop>aws elbv2 create-listener ^ --load-balancer-arn arn:aws:elasticloadbalancing:ap-south-1:426285640896
:loadbalancer/app/MyAppLoadBalancer/b4eda167e52513da ^ --protocol HTTP ^ --port 80 ^ --default-actions Type=forward,TargetGroupArn=ar
n:aws:elasticloadbalancing:ap-south-1:426285640896:targetgroup/MyTargetGroup/6df82defd1ccdb7e ^ --region ap-south-1
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:ap-south-1:426285640896:listener/app/MyAppLoadBalancer/b4eda167e52513da/4e60
ced0cc4911b7",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:ap-south-1:426285640896:loadbalancer/app/MyAppLoadBalancer/b4eda167e5251
3da",
            "Port": 80,
            "Protocol": "HTTP",
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:ap-south-1:426285640896:targetgroup/MyTargetGroup/6df82defd1ccdb7
e",
                    "ForwardConfig": {
                        "TargetGroups": [
                            {
                                "TargetGroupArn": "arn:aws:elasticloadbalancing:ap-south-1:426285640896:targetgroup/MyTargetGroup/6df
82defd1ccdb7e",
                                "Weight": 1
                            }
                        ],
                        "TargetGroupStickinessConfig": {
                            "Enabled": false
                        }
                    }
                }
            ]
        }
    ]
}
```

Registering instances to target group

```
C:\Users\Vanshika Munjal\Desktop>aws elbv2 register-targets --target-group-arn arn:aws:elasticloadbalancing:ap-south-1:426285640896:t
argetgroup/MyTargetGroup/6df82defd1ccdb7e --targets Id=i-0283a84daf50a9f3c Id=i-013429a4d06b536d4 Id=i-0361d48e369ecfac5 --region ap-
south-1
```

Attach Target Group to Auto Scaling Group

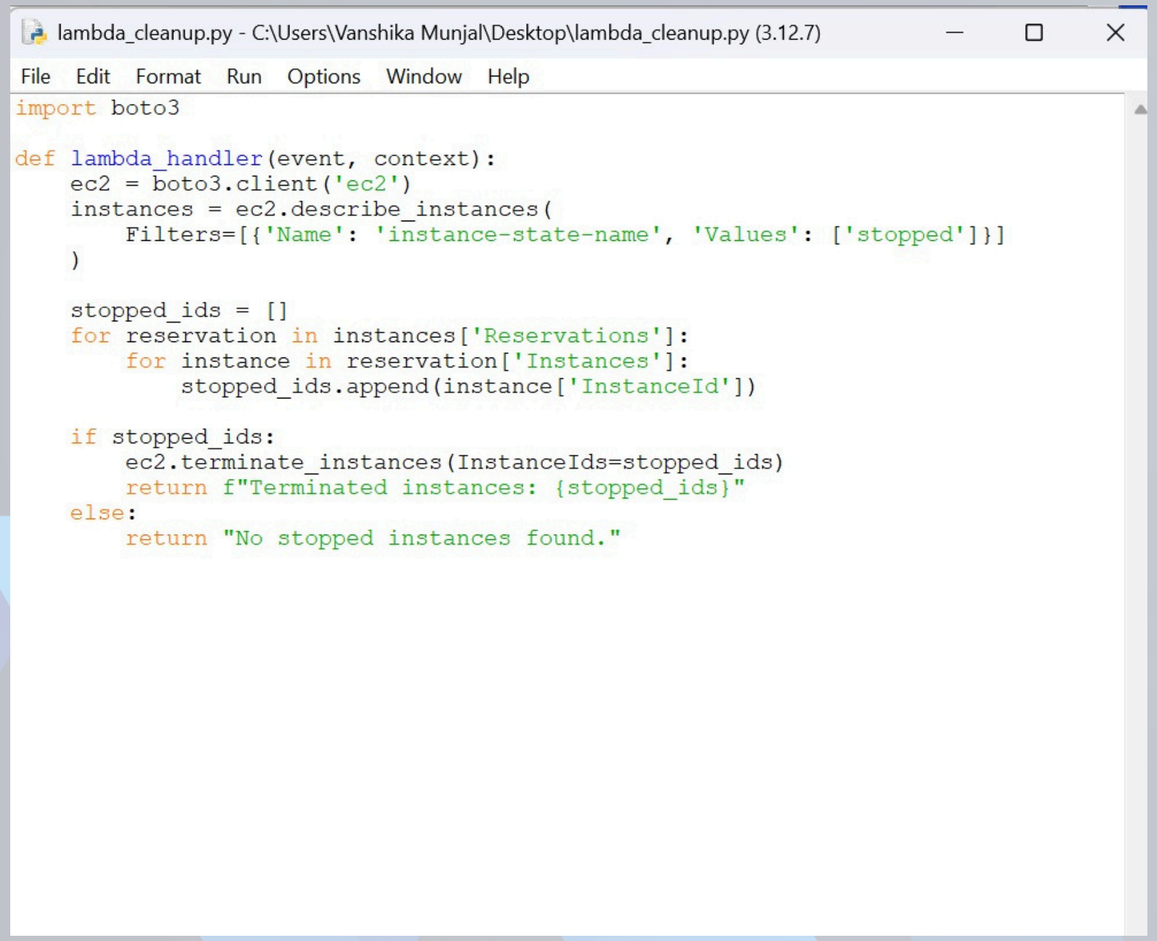
```
C:\Users\Vanshika Munjal\Desktop>aws autoscaling attach-load-balancer-target-groups ^ --auto-scaling-group-name MyASG ^ --target-grou
p-arns arn:aws:elasticloadbalancing:ap-south-1:426285640896:targetgroup/MyTargetGroup/6df82defd1ccdb7e ^ --region ap-south-1
```

Describing Load Balancer Target Groups

```
C:\Users\Vanshika Munjal\Desktop>aws autoscaling describe-load-balancer-target-groups --auto-scaling-group-name MyASG --region ap-sou
th-1
{
    "LoadBalancerTargetGroups": [
        {
            "LoadBalancerTargetGroupARN": "arn:aws:elasticloadbalancing:ap-south-1:426285640896:targetgroup/MyTargetGroup/6df82defd1c
cdb7e",
            "State": "Added"
        }
    ]
}
```

LAMBDA FUNCTION

Python file



Lambda Function deletes stopped EC2 Instances to save cost- runs on schedule using CloudWatch Events

Package Lambda Function

```
C:\Users\Vanshika Munjal\Desktop>aws cloudformation package ^ --template-file MyAutoScaleTemplate.yaml ^ --s3-bucket autoscaling-buck
et-new1 ^ --output-template-file output-template.yaml ^ --region ap-south-1

Successfully packaged artifacts and wrote output template to file output-template.yaml.
Execute the following command to deploy the packaged template
aws cloudformation deploy --template-file C:\Users\Vanshika Munjal\Desktop\output-template.yaml --stack-name <YOUR STACK NAME>
```

MyAutoScaleTemplate.yaml

File Edit View

```yaml
AWSTemplateFormatVersion: "2010-09-09"
Description: Auto Scaling Setup with ALB for EC2 instances

Parameters:
  KeyName:
    Type: String
    Default: MyNewKeypair
    Description: EC2 Key Pair name
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VPC ID
  SubnetIds:
    Type: List<AWS::EC2::Subnet::Id>
    Description: Subnets for Load Balancer

Resources:

  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: MyAutoScaleTemplate
      LaunchTemplateData:
        ImageId: ami-0019610b50e1fe6bb
        InstanceType: t2.micro
        KeyName: !Ref KeyName
        SecurityGroupIds:
          - sg-0d512e764a6a47df0
        IamInstanceProfile:
          Name: EC2AutoScalingProfile

  MyTargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      Name: MyTargetGroup
      Port: 80
      Protocol: HTTP
      VpcId: !Ref VpcId
```

Defines the template format version and a brief description

```
C:\Users\Vanshika Munjal\Desktop>aws iam create-role --role-name LambdaEC2CleanupRole --assume-role-policy-document file://lambda-tru
st-policy.json
{
    "Role": {
        "Path": "/",
        "RoleName": "LambdaEC2CleanupRole",
        "RoleId": "AROAWGQE2VTAOIJE55CUY",
        "Arn": "arn:aws:iam::426285640896:role/LambdaEC2CleanupRole",
        "CreateDate": "2025-05-07T11:31:50+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "lambda.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

Creating IAM Role for Lambda Function

lambda-trust-policy.json

File Edit View

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "lambda.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }]
}
```

Attach Policies

```
C:\Users\Vanshika Munjal\Desktop>aws iam attach-role-policy --role-name LambdaEC2CleanupRole --policy-arn arn:aws:iam::aws:policy/Ama
zonEC2FullAccess
```

```
C:\Users\Vanshika Munjal\Desktop>aws lambda create-function --function-name EC2CleanupFunction --runtime python3.12 --role arn:aws:ia
m::426285640896:role/LambdaEC2CleanupRole --handler lambda_cleanup.lambda_handler --zip-file fileb://lambda_cleanup.zip --region ap-s
outh-1
{
    "FunctionName": "EC2CleanupFunction",
    "FunctionArn": "arn:aws:lambda:ap-south-1:426285640896:function:EC2CleanupFunction",
    "Runtime": "python3.12",
    "Role": "arn:aws:iam::426285640896:role/LambdaEC2CleanupRole",
    "Handler": "lambda_cleanup.lambda_handler",
    "CodeSize": 487,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2025-05-07T11:36:38.790+0000",
    "CodeSha256": "fG6q7Ca8ap/kjrJ3kdSQloIqustiQoeXqwF4JLJz+Zo=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "2ecdaebc-425a-4487-b924-f6f0c7efcb77",
    "State": "Pending",
    "StateReason": "The function is being created.",
    "StateReasonCode": "Creating",
    "PackageType": "Zip",
    "Architectures": [
        "x86_64"
    ],
    "EphemeralStorage": {
        "Size": 512
    },
    "SnapStart": {
        "ApplyOn": "None",
        "OptimizationStatus": "Off"
    },
```

Deploy Lambda Function

Schedule Lambda with Cloud Watch Events

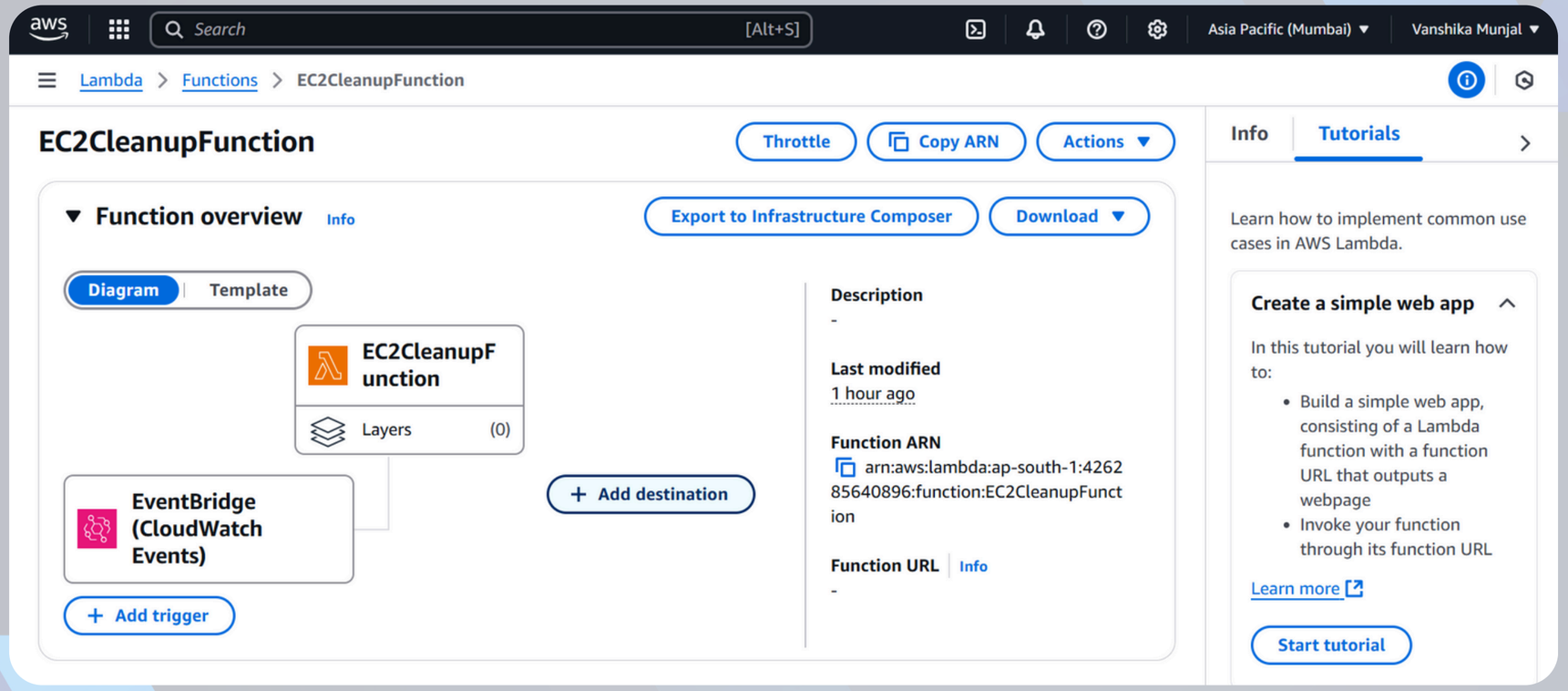
```
C:\Users\Vanshika Munjal\Desktop>aws events put-rule --schedule-expression "rate(1 hour)"  --name EC2CleanupRule
{
    "RuleArn": "arn:aws:events:ap-south-1:426285640896:rule/EC2CleanupRule"
}
```

Add Lambda as Target

```
C:\Users\Vanshika Munjal\Desktop>aws events put-targets ^ --rule EC2CleanupRule ^ --targets Id="1",Arn="arn:aws:lambda:ap-south-1:426
285640896:function:EC2CleanupFunction"
{
    "FailedEntryCount": 0,
    "FailedEntries": []
}
```

Give permissions to event bridge

```
C:\Users\Vanshika Munjal\Desktop>aws lambda add-permission ^ --function-name EC2CleanupFunction ^ --statement-id AllowExecutionFromEv
entBridge ^ --action lambda:InvokeFunction ^ --principal events.amazonaws.com ^ --source-arn arn:aws:events:ap-south-1:426285640896:r
ule/EC2CleanupRule ^ --region ap-south-1
{
    "Statement": "{\"Sid\":\"AllowExecutionFromEventBridge\",\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"
},\"Action\":\"lambda:InvokeFunction\",\"Resource\":\"arn:aws:lambda:ap-south-1:426285640896:function:EC2CleanupFunction\",\"Conditio
n\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:events:ap-south-1:426285640896:rule/EC2CleanupRule\"}}}"
}
```



You now have a Lambda function running hourly, scanning for unused EC2 instances (such as those spun up by Auto Scaling and later stopped), and terminating them automatically.

CHALLENGES FACED & SOLUTIONS IMPLEMENTED

1. IAM Role Permissions
Challenge: Lambda functions and EC2 instances initially failed to access required AWS resources (S3, CloudWatch).

Solution: Created and attached custom IAM roles with the minimum required permissions. Verified using the IAM Policy Simulator.

2. Lambda Permission Error
Challenge: EventBridge rule couldn’t trigger the Lambda function.

Solution: Used aws lambda add-permission to explicitly allow EventBridge to invoke the Lambda function with the correct source-arn.

Solution: Rechecked the full ARN in the Lambda console and replaced placeholders with the actual region and account ID.

CHALLENGES FACED & SOLUTIONS IMPLEMENTED

3. Invalid Target Group ARN
Challenge: Encountered ARN format errors while registering the Lambda function to EventBridge.

Solution: Rechecked the full ARN in the Lambda console and replaced placeholders with the actual region and account ID.

4. Load Balancer & Subnet Mismatch
Challenge: ELB failed to associate with the Auto Scaling Group due to incorrect or insufficient subnets.

Solution: Used aws ec2 describe-subnets to confirm subnets were in different Availability Zones and belonged to the correct VPC.

FUTURE IMPROVEMENTS

- Implement Scaling Policies: Add CPU or memory-based scaling policies to make Auto Scaling more intelligent.
- Add CI/CD Integration: Use AWS CodePipeline and CodeDeploy to automate deployments for Lambda and EC2.

- Introduce Step Scaling or Scheduled Scaling: Optimize resource usage during known traffic patterns.
- Implement HTTPS on ELB: Add SSL certificates to support secure connEctions.

- Enable Detailed Monitoring: Use detailed metrics in CloudWatch for better insights and alarms.

CONCLUSION

This project successfully demonstrated a dynamic and scalable cloud architecture using AWS services such as EC2, Auto Scaling, Elastic Load Balancer, IAM, S3, EBS, Lambda, and CloudWatch. The infrastructure auto-scales based on demand, ensures high availability, and is integrated with automation through EventBridge and Lambda functions.

THANK YOU