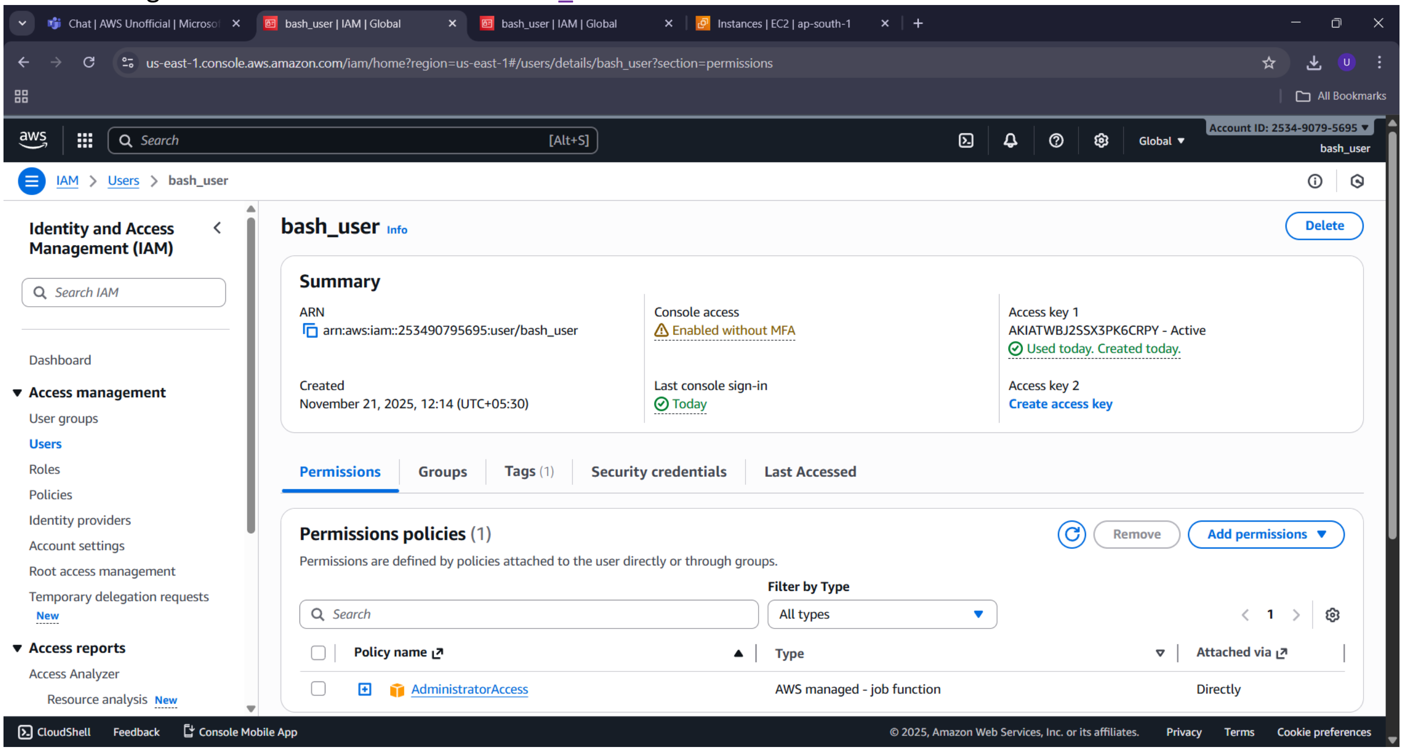
NAME : UDDHAV BAPU GUND DESIGNATION : JR.AWS DEVELOPER
TASK: Automate AWS EC2 & S3 Resource Creation and Cleanup Using Shell Scripts

PREREQUISITES:

1.Installed Git Bash(Required for .sh files)
2.Install AWS CLI on Windows.

STEP 1: CONFIGURE AWS CREDENTIALS AND CREATED TASK FOLDER

1.AWS user generated from AWS Console bash_user



bash_user Info

Summary

ARN
arn:aws:iam::253490795695:user/bash_user

Created
November 21, 2025, 12:14 (UTC+05:30)

Console access
Enabled without MFA

Last console sign-in
Today

Access key 1
AKIATWBJ2SSX3PK6CRPY - Active
Used today. Created today.

Access key 2
Create access key

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

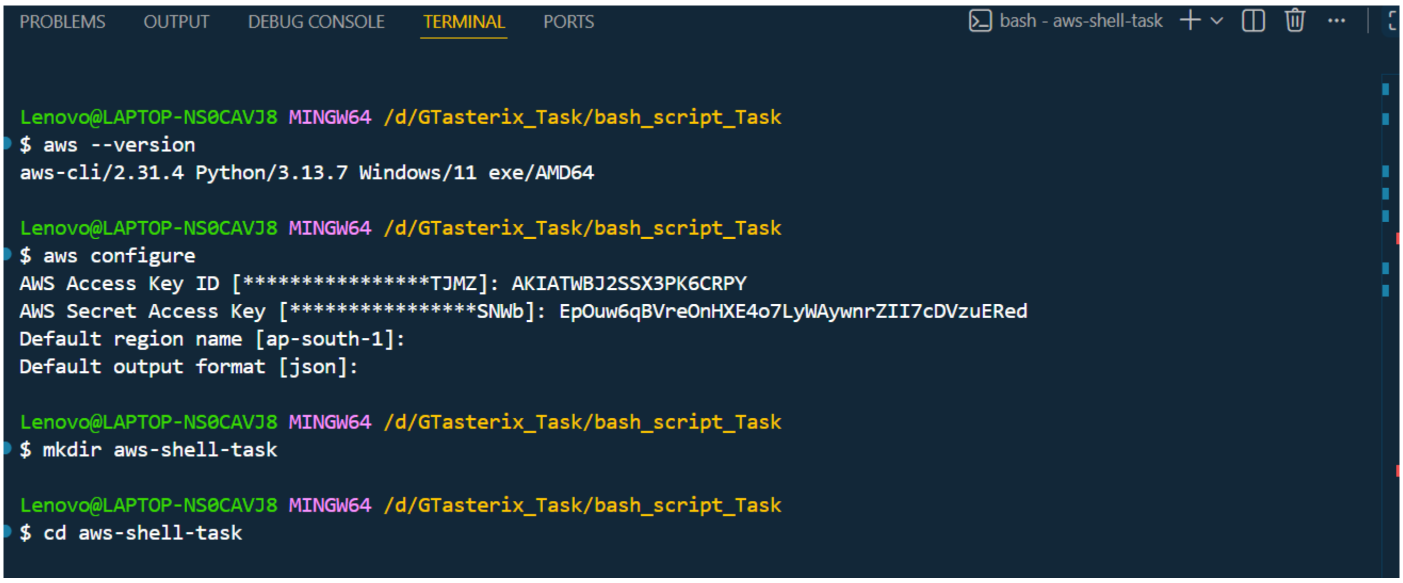
Permissions policies (1)
Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

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

2.Generated Access Key
3.Inside GitBash run:
    aws configure
4.Enter Details.

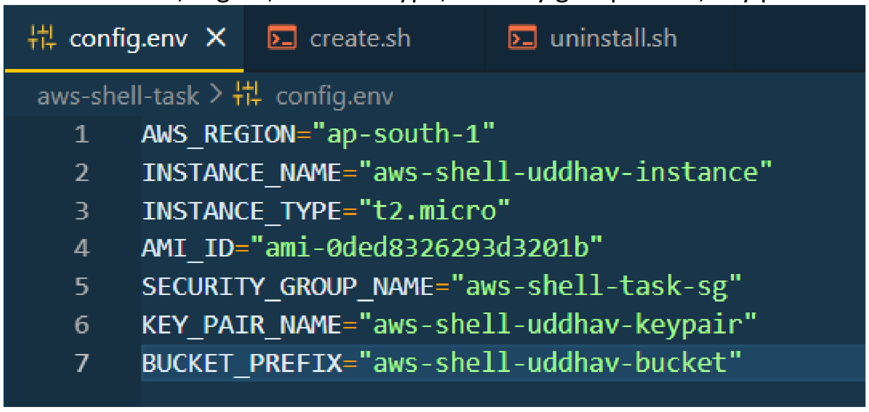
5.CREATED TASK FOLDER
    mkdir aws-shell-task
    cd aws-shell-task



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task
$ aws --version
aws-cli/2.31.4 Python/3.13.7 Windows/11 exe/AMD64

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task
$ aws configure
AWS Access Key ID [****************TJMZ]: AKIATWBJ2SSX3PK6CRPY
AWS Secret Access Key [****************SNWb]: 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
Default region name [ap-south-1]:
Default output format [json]:

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task
$ mkdir aws-shell-task

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task
$ cd aws-shell-task
```

STEP 2 : Created config.env File

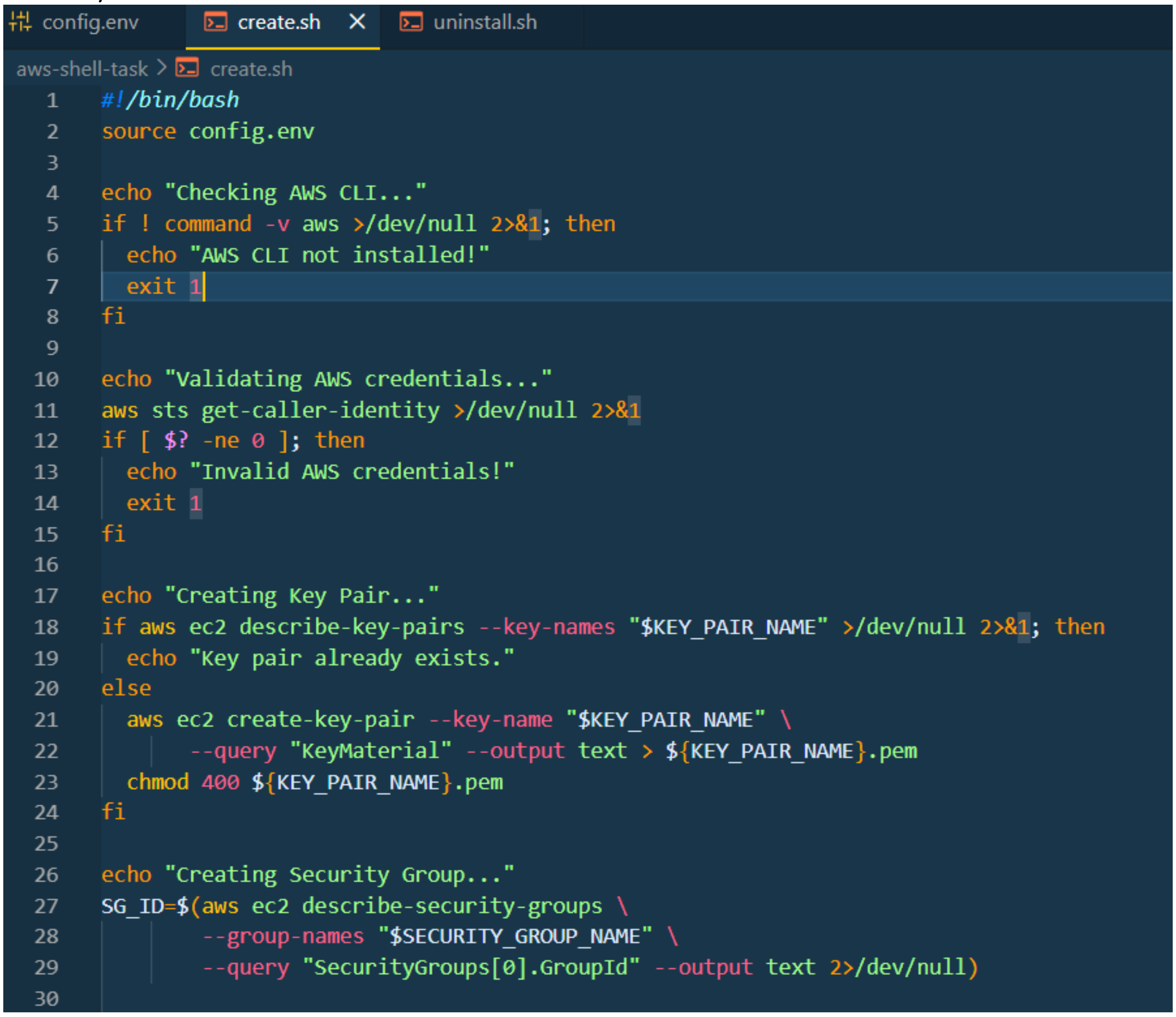
Inside the project directory, I created the **config.env** file and populated it with the required environment variables such as AMI ID, region, instance type, security group name, key pair name, and S3 bucket prefix.



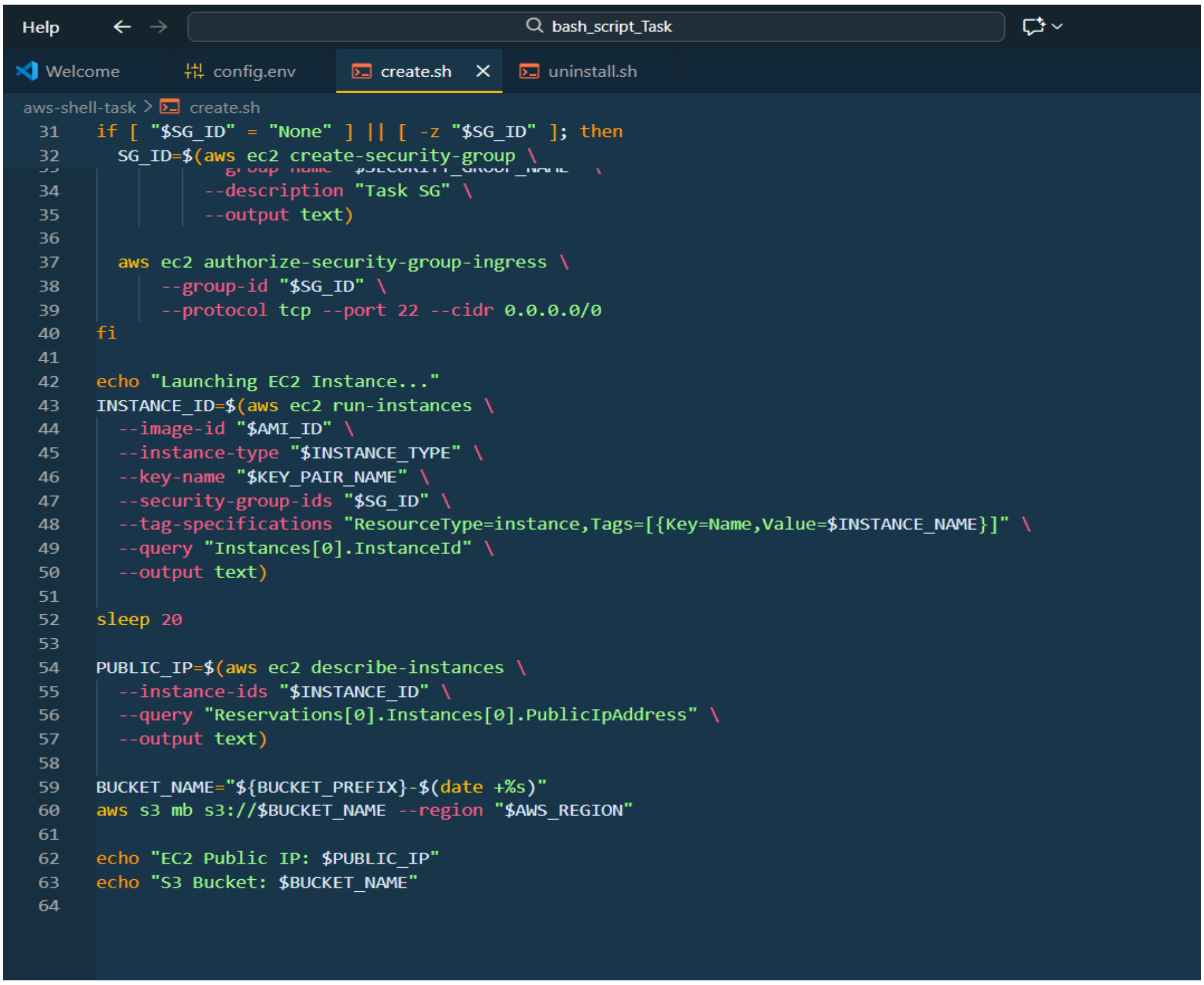
```
AWS_REGION="ap-south-1"
INSTANCE_NAME="aws-shell-uddhav-instance"
INSTANCE_TYPE="t2.micro"
AMI_ID="ami-0ded8326293d3201b"
SECURITY_GROUP_NAME="aws-shell-task-sg"
KEY_PAIR_NAME="aws-shell-uddhav-keypair"
BUCKET_PREFIX="aws-shell-uddhav-bucket"
```

STEP 3 : Created create.sh file

Inside that file consist of Load values from config.env,Validate AWS CLI installation,Validate AWS credentials , Creating Key Pair,Security Group,EC2 Instance,S3 Bucket with prefix,Save the generated S3 bucket name + EC2 public IP in the summary.



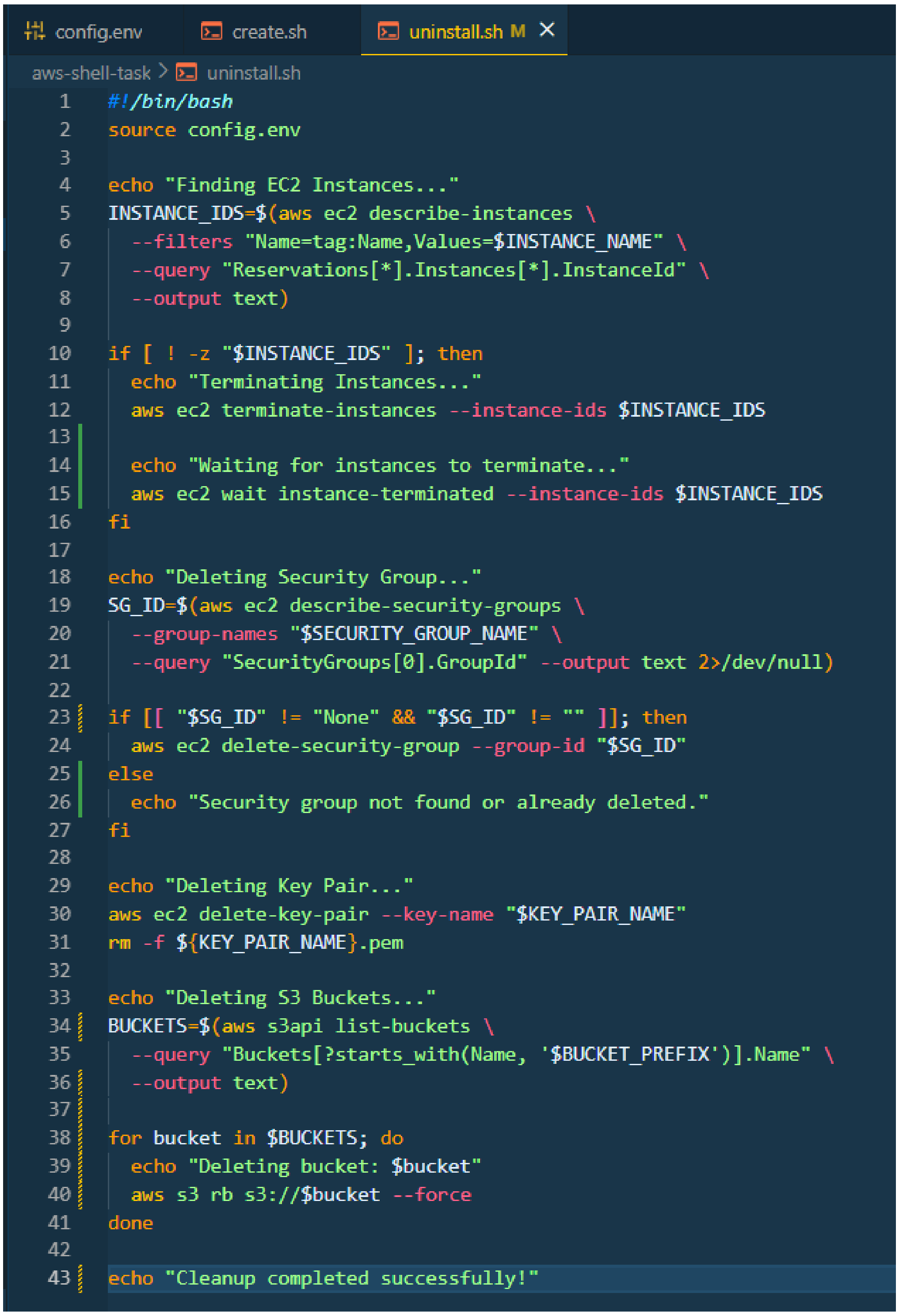
```
#!/bin/bash
source config.env

echo "Checking AWS CLI..."
if ! command -v aws >/dev/null 2>&1; then
  echo "AWS CLI not installed!"
  exit 1
fi

echo "Validating AWS credentials..."
aws sts get-caller-identity >/dev/null 2>&1
if [ $? -ne 0 ]; then
  echo "Invalid AWS credentials!"
  exit 1
fi

echo "Creating Key Pair..."
if aws ec2 describe-key-pairs --key-names "$KEY_PAIR_NAME" >/dev/null 2>&1; then
  echo "Key pair already exists."
else
  aws ec2 create-key-pair --key-name "$KEY_PAIR_NAME" \
      --query "KeyMaterial" --output text > ${KEY_PAIR_NAME}.pem
  chmod 400 ${KEY_PAIR_NAME}.pem
fi

echo "Creating Security Group..."
SG_ID=$(aws ec2 describe-security-groups \
        --group-names "$SECURITY_GROUP_NAME" \
        --query "SecurityGroups[0].GroupId" --output text 2>/dev/null)

```



```bash
if [ "$SG_ID" = "None" ] || [ -z "$SG_ID" ]; then
  SG_ID=$(aws ec2 create-security-group \
          --group-name "$SECURITY_GROUP_NAME" \
          --description "Task SG" \
          --output text)

  aws ec2 authorize-security-group-ingress \
      --group-id "$SG_ID" \
      --protocol tcp --port 22 --cidr 0.0.0.0/0
fi

echo "Launching EC2 Instance..."
INSTANCE_ID=$(aws ec2 run-instances \
  --image-id "$AMI_ID" \
  --instance-type "$INSTANCE_TYPE" \
  --key-name "$KEY_PAIR_NAME" \
  --security-group-ids "$SG_ID" \
  --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=$INSTANCE_NAME}]" \
  --query "Instances[0].InstanceId" \
  --output text)

sleep 20

PUBLIC_IP=$(aws ec2 describe-instances \
  --instance-ids "$INSTANCE_ID" \
  --query "Reservations[0].Instances[0].PublicIpAddress" \
  --output text)

BUCKET_NAME="${BUCKET_PREFIX}-$(date +%s)"
aws s3 mb s3://$BUCKET_NAME --region "$AWS_REGION"

echo "EC2 Public IP: $PUBLIC_IP"
echo "S3 Bucket: $BUCKET_NAME"
```

STEP 4 : Created unistall.sh file

This file consists of This script must: Load values from config.env,Automatically detect all EC2 instances with the tag Name = INSTANCE_NAME, Terminate all matching EC2 instances, Delete the security group if it exists,Delete the key-pair + local .pem file, Identify and delete all S3 buckets starting with BUCKET_NAME_PREFIX, Display final cleanup summary.



config.env | create.sh | uninstall.sh M

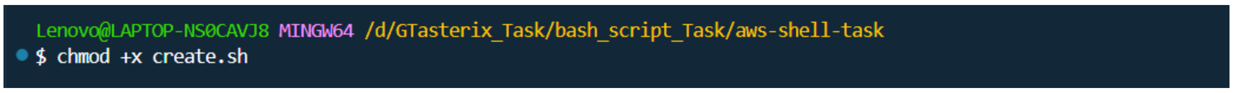
aws-shell-task > uninstall.sh

```bash
#!/bin/bash
source config.env

echo "Finding EC2 Instances..."
INSTANCE_IDS=$(aws ec2 describe-instances \
  --filters "Name=tag:Name,Values=$INSTANCE_NAME" \
  --query "Reservations[*].Instances[*].InstanceId" \
  --output text)

if [ ! -z "$INSTANCE_IDS" ]; then
  echo "Terminating Instances..."
  aws ec2 terminate-instances --instance-ids $INSTANCE_IDS

  echo "Waiting for instances to terminate..."
  aws ec2 wait instance-terminated --instance-ids $INSTANCE_IDS
fi

echo "Deleting Security Group..."
SG_ID=$(aws ec2 describe-security-groups \
  --group-names "$SECURITY_GROUP_NAME" \
  --query "SecurityGroups[0].GroupId" --output text 2>/dev/null)

if [[ "$SG_ID" != "None" && "$SG_ID" != "" ]]; then
  aws ec2 delete-security-group --group-id "$SG_ID"
else
  echo "Security group not found or already deleted."
fi

echo "Deleting Key Pair..."
aws ec2 delete-key-pair --key-name "$KEY_PAIR_NAME"
rm -f ${KEY_PAIR_NAME}.pem

echo "Deleting S3 Buckets..."
BUCKETS=$(aws s3api list-buckets \
  --query "Buckets[?starts_with(Name, '$BUCKET_PREFIX')].Name" \
  --output text)

for bucket in $BUCKETS; do
  echo "Deleting bucket: $bucket"
  aws s3 rb s3://$bucket --force
done

echo "Cleanup completed successfully!"
```

STEP 5 :Made the Scripts Executable

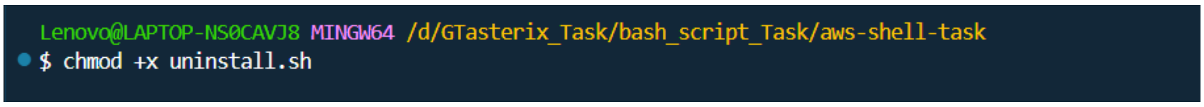
Before running the scripts, I provided execution permissions using:

chmod +x create.sh



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task
$ chmod +x create.sh
```

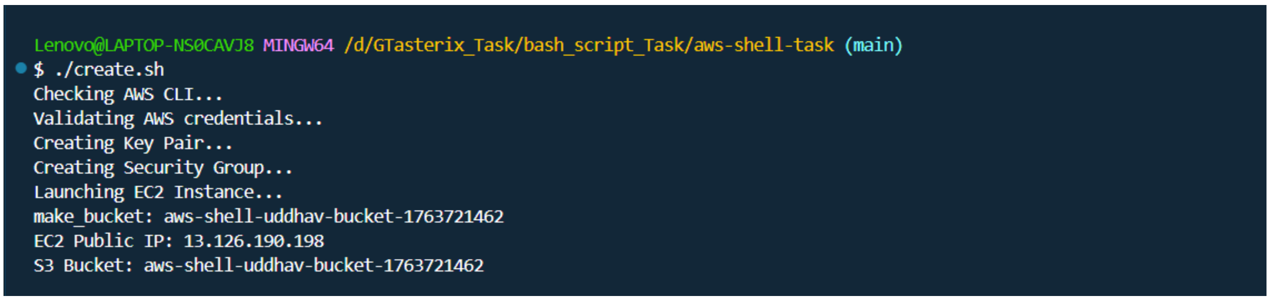
chmod +x uninstall.sh



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task
$ chmod +x uninstall.sh
```

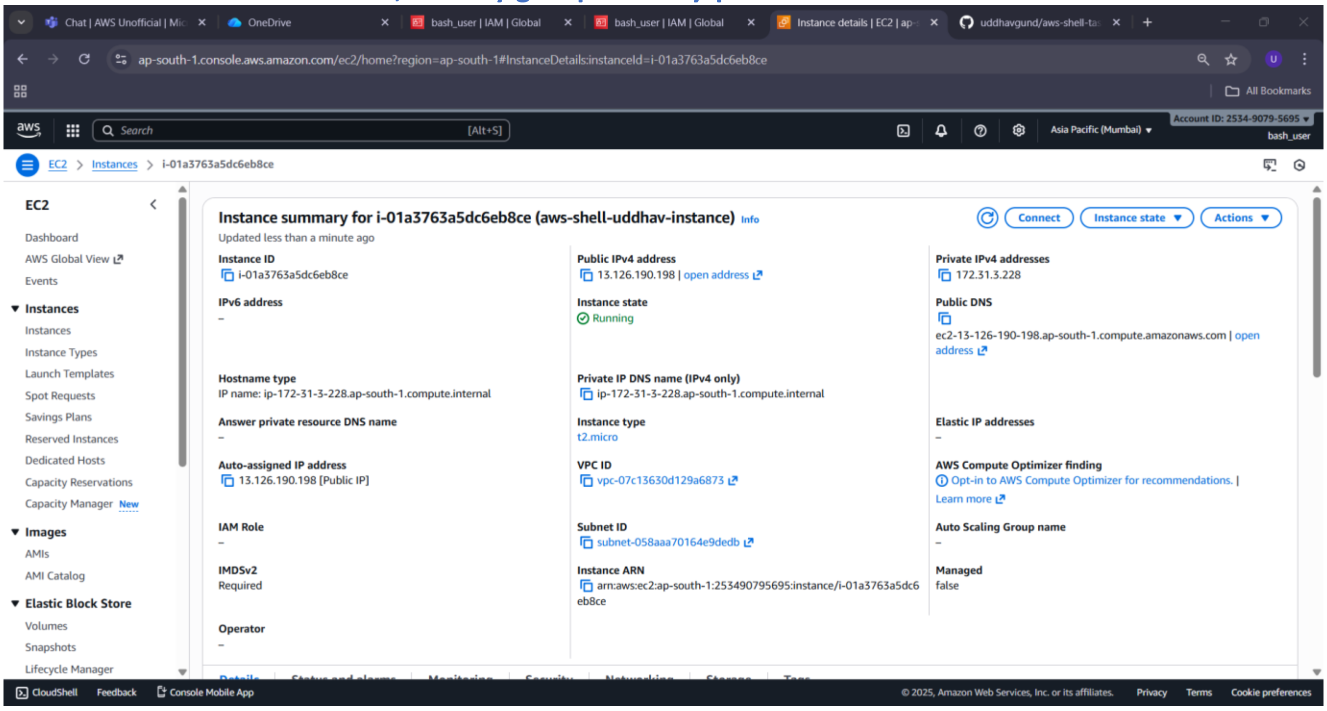
STEP 6 :Executed the Creation Script

I initiated AWS resource provisioning by running:
./create.sh



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ ./create.sh
Checking AWS CLI...
Validating AWS credentials...
Creating Key Pair...
Creating Security Group...
Launching EC2 Instance...
make_bucket: aws-shell-uddhav-bucket-1763721462
EC2 Public IP: 13.126.190.198
S3 Bucket: aws-shell-uddhav-bucket-1763721462
```

STEP 7 : Verified EC2 Instance,security group and key-pair from console and AWS CLI

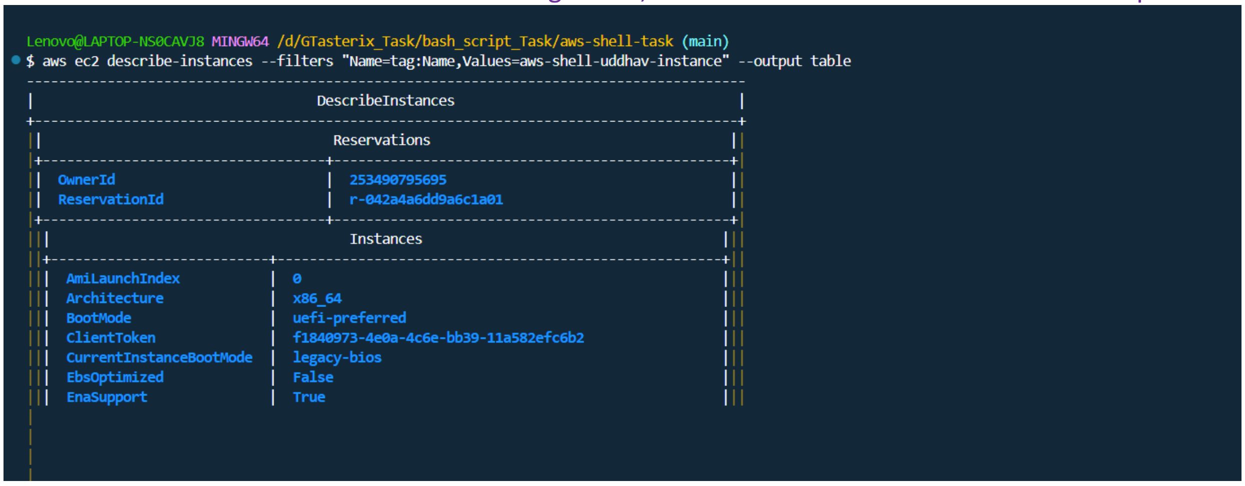


Instance summary for i-01a3763a5dc6eb8ce (aws-shell-uddhav-instance)

Updated less than a minute ago

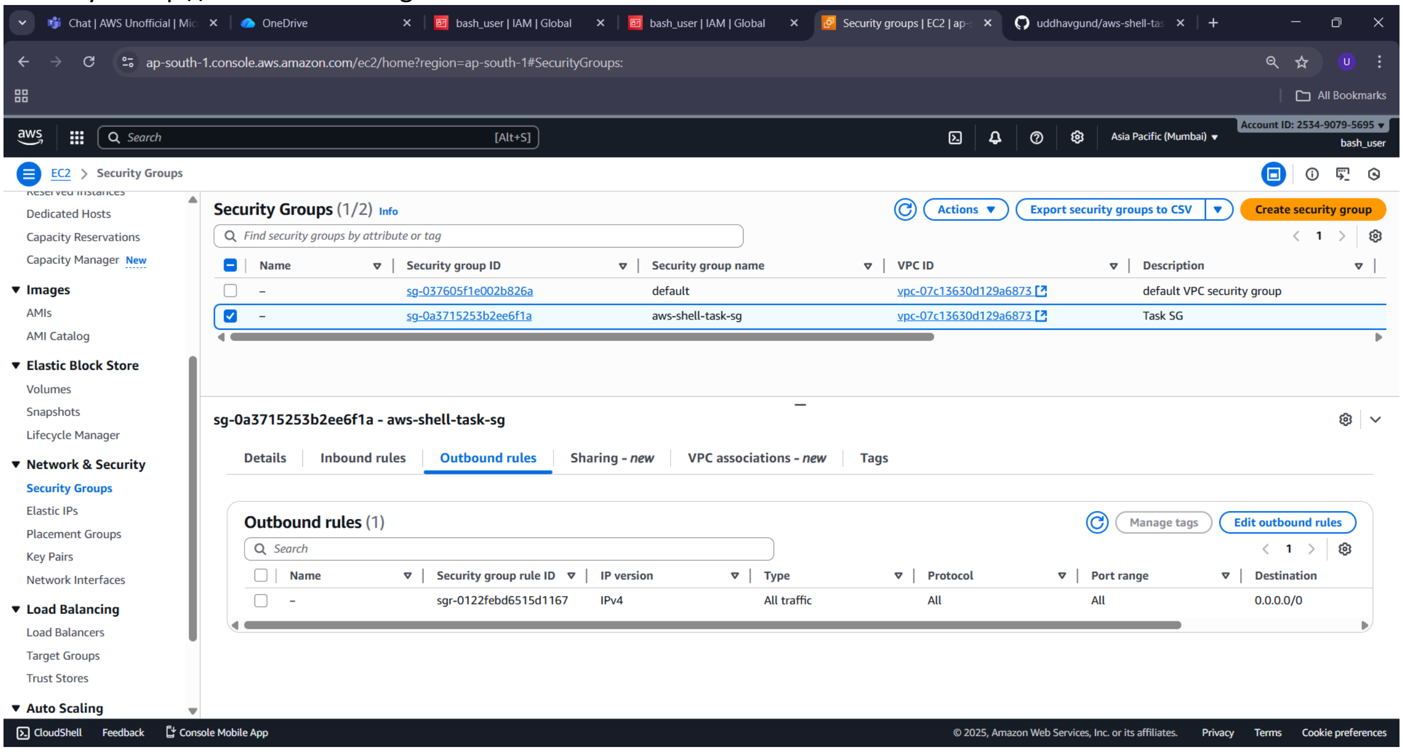
Instance ID: i-01a3763a5dc6eb8ce	Public IPv4 address: 13.126.190.198 | open address	Private IPv4 addresses: 172.31.3.228
IPv6 address: –	Instance state: Running	Public DNS: ec2-13-126-190-198.ap-south-1.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-3-228.ap-south-1.compute.internal	Private IP DNS name (IPv4 only): ip-172-31-3-228.ap-south-1.compute.internal	
Answer private resource DNS name: –	Instance type: t2.micro	Elastic IP addresses: –
Auto-assigned IP address: 13.126.190.198 [Public IP]	VPC ID: vpc-07c13630d129a6873	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –	Subnet ID: subnet-058aaa70164e9dedb	Auto Scaling Group name: –
IMDSv2: Required	Instance ARN: arn:aws:ec2:ap-south-1:253490795695:instance/i-01a3763a5dc6eb8ce	Managed: false
Operator: –		

Cheking EC2 deatails from AWS CLI
Using command :
aws ec2 describe-instances --filters "Name=tag:Name,Values=aws-shell-uddhav-instance" --output table



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=aws-shell-uddhav-instance" --output table
 -------------------------------------------------------------------------------------------
|                                    DescribeInstances                                      |
+-------------------------------------------------------------------------------------------+
||                                      Reservations                                       ||
|+-------------------------------------+---------------------------------------------------+|
||  OwnerId                            |  253490795695                                     ||
||  ReservationId                      |  r-042a4a6dd9a6c1a01                              ||
|+-------------------------------------+---------------------------------------------------+|
|||                                       Instances                                       |||
||+----------------------------+--------------------------------------------------------+||
|||  AmiLaunchIndex            |  0                                                     |||
|||  Architecture              |  x86_64                                                |||
|||  BootMode                  |  uefi-preferred                                        |||
|||  ClientToken               |  f1840973-4e0a-4c6e-bb39-11a582efc6b2                  |||
|||  CurrentInstanceBootMode   |  legacy-bios                                           |||
|||  EbsOptimized              |  False                                                 |||
|||  EnaSupport                |  True                                                  |||
|
|
|
```

Security Group //aws-shell-task-sg



Security Groups (1/2) Info

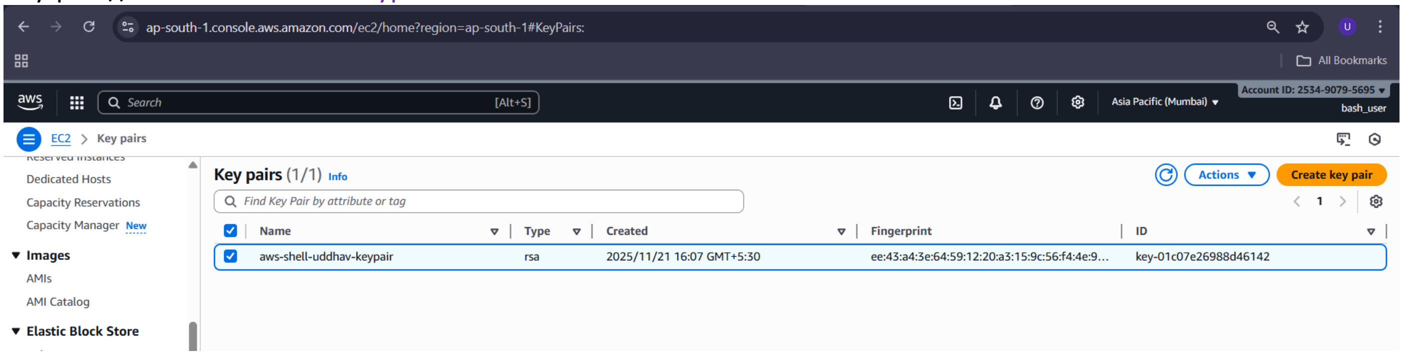
Name	Security group ID	Security group name	VPC ID	Description
–	sg-037605f1e002b826a	default	vpc-07c13630d129a6873	default VPC security group
–	sg-0a3715253b2ee6f1a	aws-shell-task-sg	vpc-07c13630d129a6873	Task SG

sg-0a3715253b2ee6f1a - aws-shell-task-sg

Outbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination
–	sgr-0122febd6515d1167	IPv4	All traffic	All	All	0.0.0.0/0

Key-pair //aws-shell-uddhav-keypair



Key pairs (1/1) Info

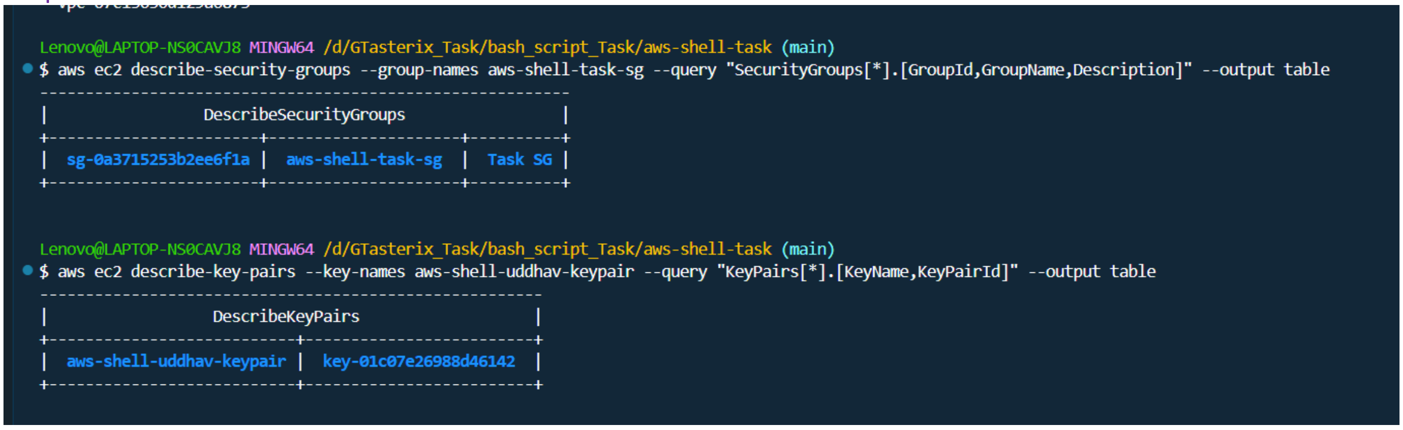
Name	Type	Created	Fingerprint	ID
aws-shell-uddhav-keypair	rsa	2025/11/21 16:07 GMT+5:30	ee:43:a4:3e:64:59:12:20:a3:15:9c:56:f4:4e:9...	key-01c07e26988d46142

Security Group and key pair from AWS CLI

Command:

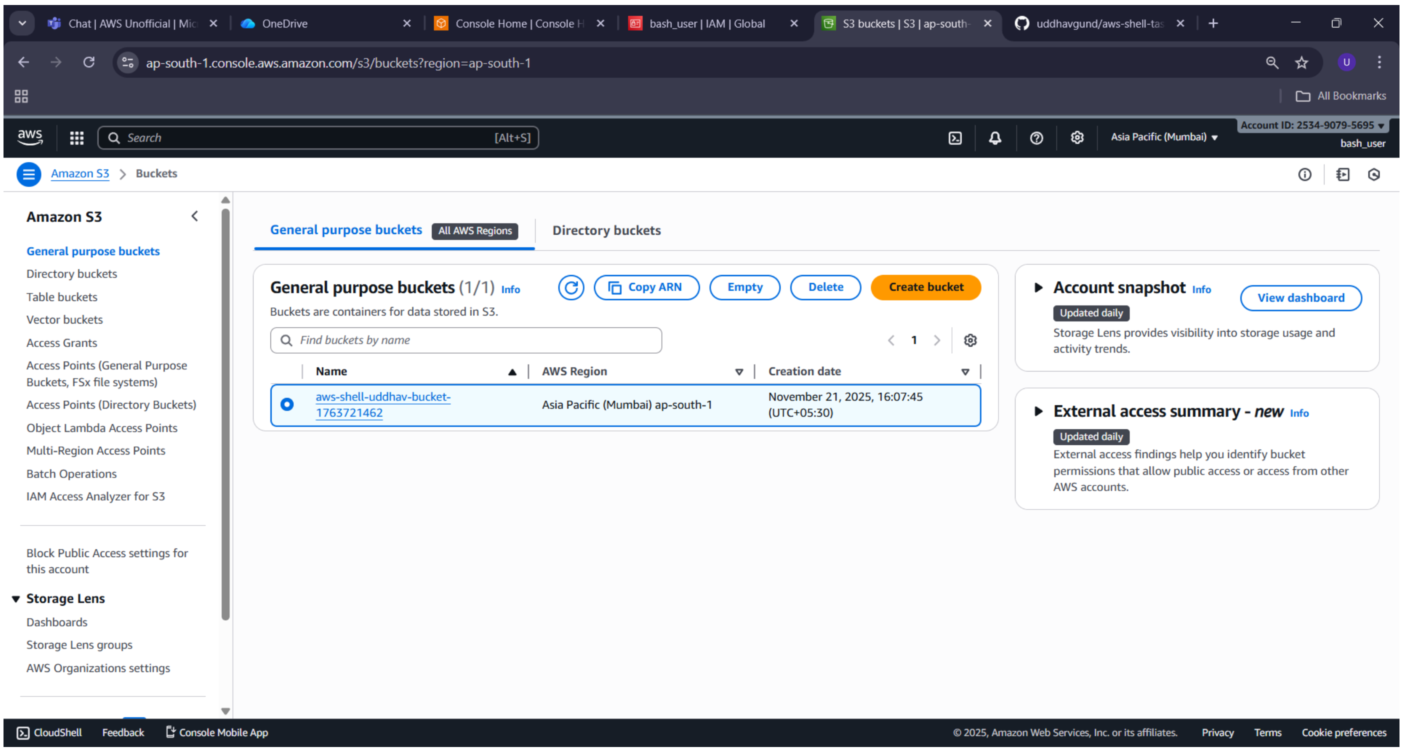
aws ec2 describe-security-groups --group-names aws-shell-task-sg --query "SecurityGroups[*].[GroupId,GroupName,Description]" --output table

aws ec2 describe-key-pairs --key-names aws-shell-uddhav-keypair --query "KeyPairs[*].[KeyName,KeyPairId]" --output table



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ aws ec2 describe-security-groups --group-names aws-shell-task-sg --query "SecurityGroups[*].[GroupId,GroupName,Description]" --output table
-----------------------------------------------------------
|                 DescribeSecurityGroups                  |
+-----------------------+----------------------+----------+
|  sg-0a3715253b2ee6f1a |  aws-shell-task-sg   |  Task SG |
+-----------------------+----------------------+----------+

Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ aws ec2 describe-key-pairs --key-names aws-shell-uddhav-keypair --query "KeyPairs[*].[KeyName,KeyPairId]" --output table
-------------------------------------------------------
|                   DescribeKeyPairs                  |
+---------------------------+-------------------------+
|  aws-shell-uddhav-keypair |  key-01c07e26988d46142  |
+---------------------------+-------------------------+
```

STEP 8 : Verifed S3 Bucket from AWS Console and AWS CLI



ap-south-1.console.aws.amazon.com/s3/buckets?region=ap-south-1

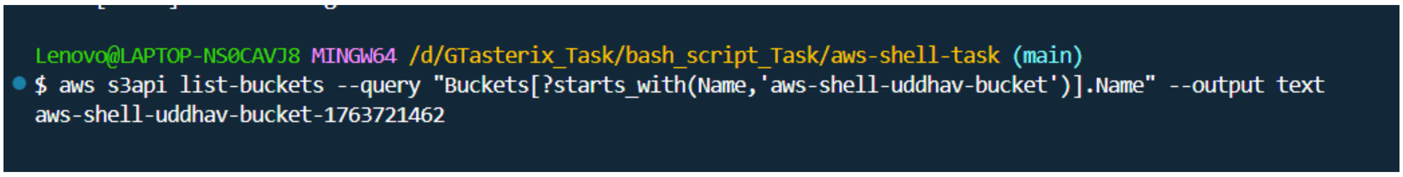
General purpose buckets (1/1)
Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
aws-shell-uddhav-bucket-1763721462	Asia Pacific (Mumbai) ap-south-1	November 21, 2025, 16:07:45 (UTC+05:30)

Storage Lens provides visibility into storage usage and activity trends.

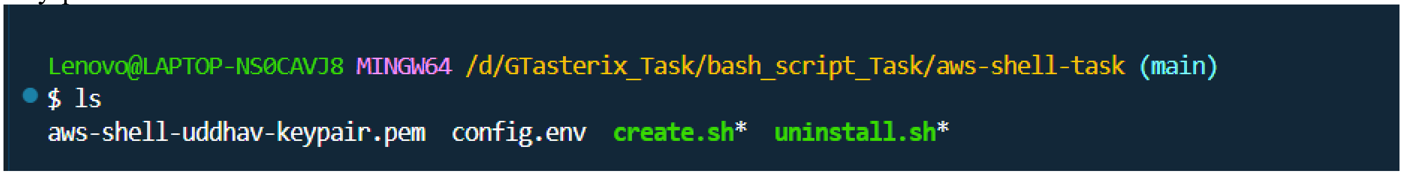
External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

Verified from AWS CLI
Command:
aws s3api list-buckets --query "Buckets[?starts_with(Name,'aws-shell-uddhav-bucket')].Name" --output text



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ aws s3api list-buckets --query "Buckets[?starts_with(Name,'aws-shell-uddhav-bucket')].Name" --output text
aws-shell-uddhav-bucket-1763721462
```

Key-pair file added at local folder



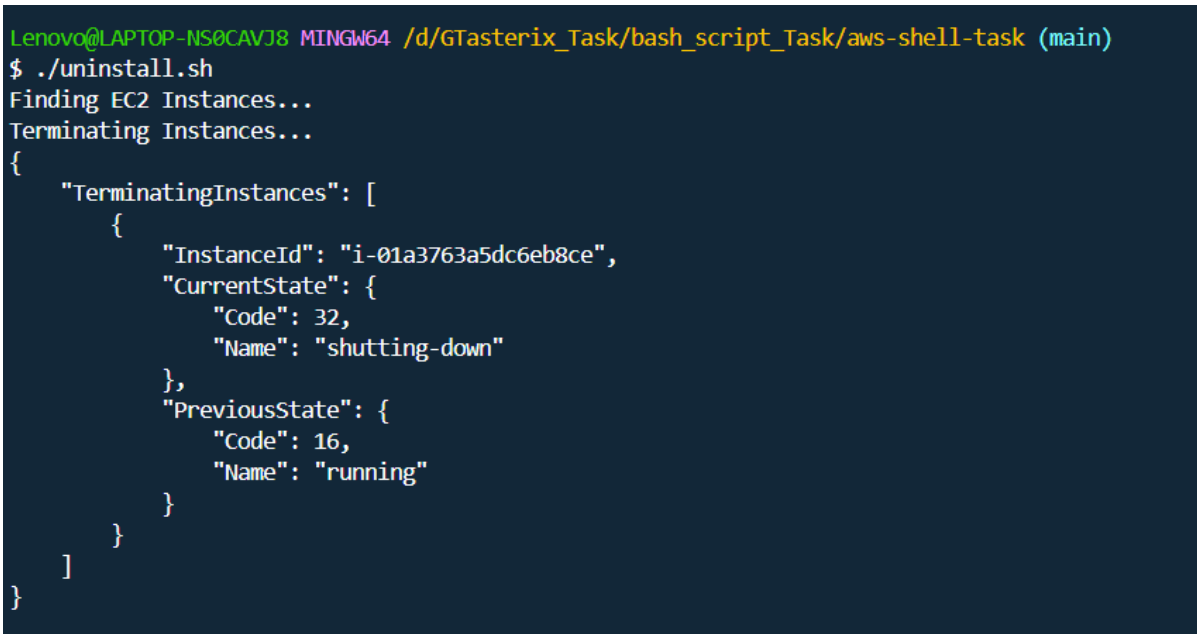
```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ ls
aws-shell-uddhav-keypair.pem  config.env  create.sh*  uninstall.sh*
```

STEP 9 : Executed the Cleanup Script

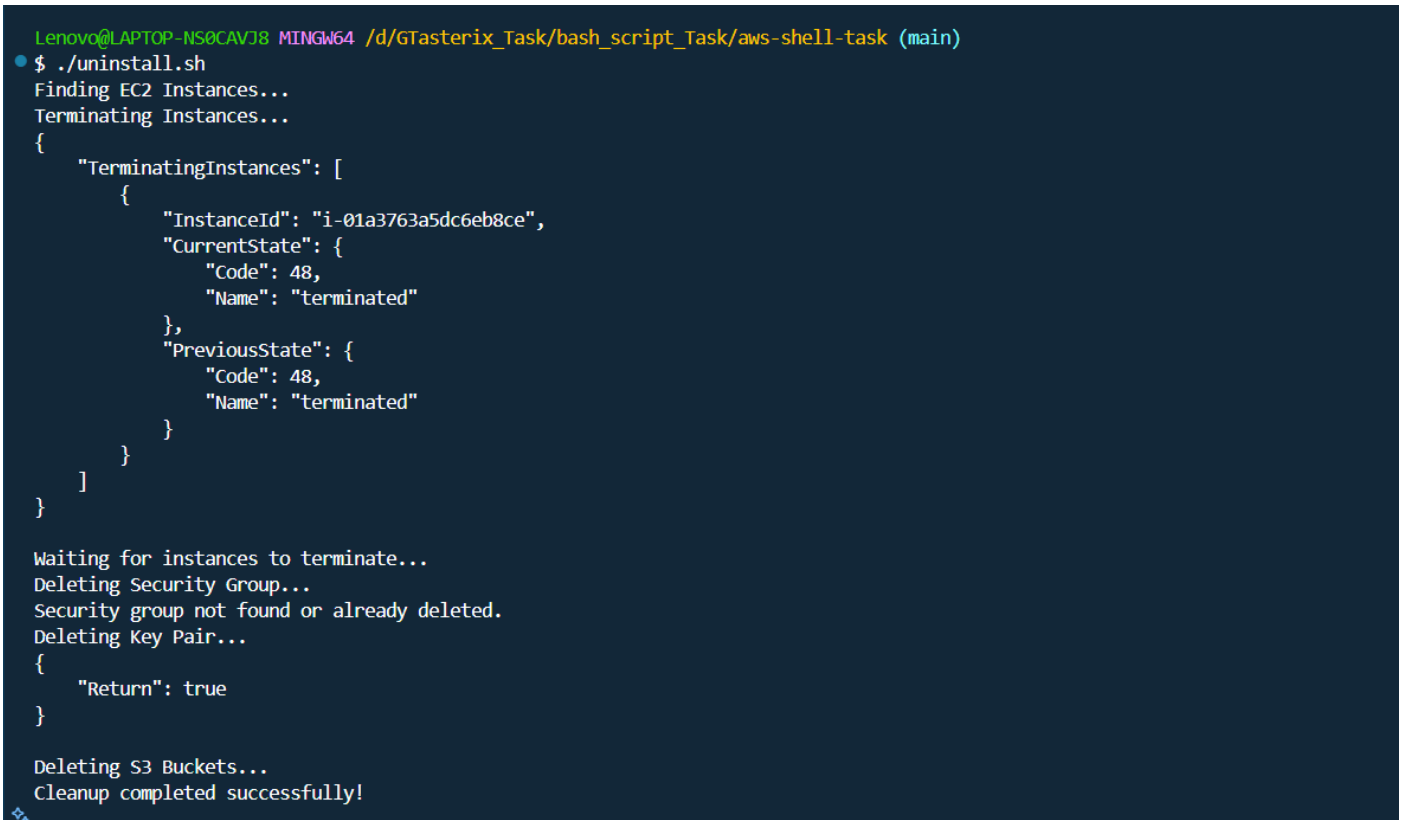
After testing the created resources, I executed the deletion script using:

./uninstall.sh

This script:
Detected and terminated the EC2 instance
Deleted the security group
Removed the AWS key pair and local .pem file
Identified and deleted all S3 buckets starting with the configured prefix
Displayed a final cleanup summary.



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ ./uninstall.sh
Finding EC2 Instances...
Terminating Instances...
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-01a3763a5dc6eb8ce",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```



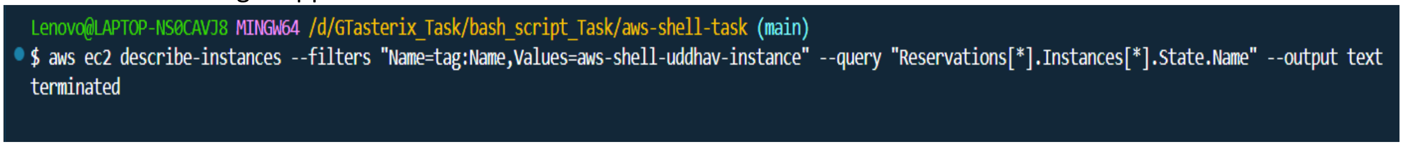
```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ ./uninstall.sh
Finding EC2 Instances...
Terminating Instances...
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-01a3763a5dc6eb8ce",
            "CurrentState": {
                "Code": 48,
                "Name": "terminated"
            },
            "PreviousState": {
                "Code": 48,
                "Name": "terminated"
            }
        }
    ]
}

Waiting for instances to terminate...
Deleting Security Group...
Security group not found or already deleted.
Deleting Key Pair...
{
    "Return": true
}

Deleting S3 Buckets...
Cleanup completed successfully!
```

All resources created during the task were removed successfully.

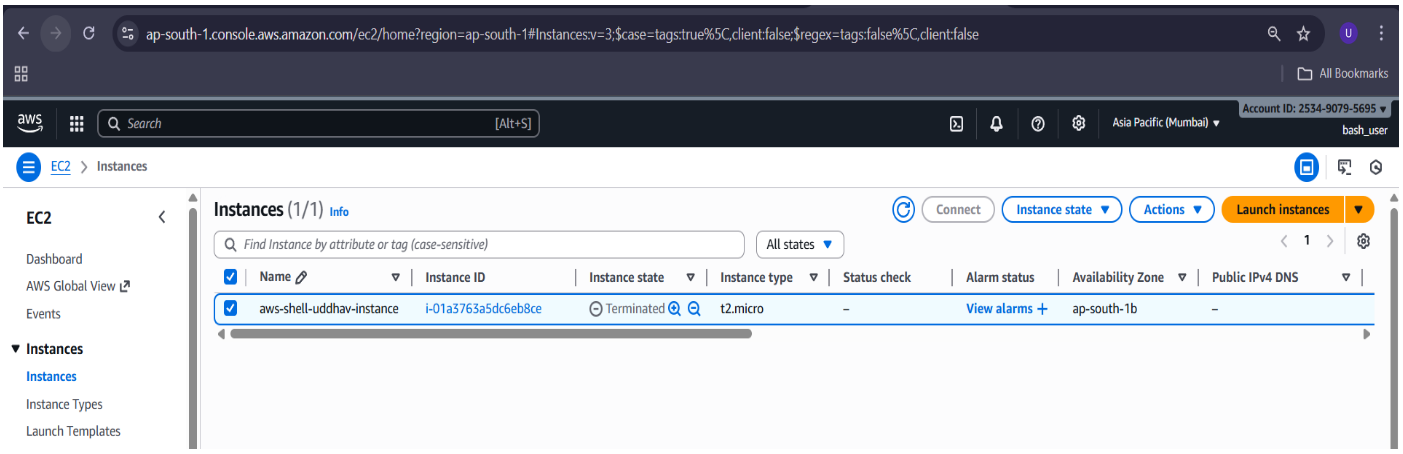
STEP 10 : Verified Resource Deletion

I confirmed that the automation completed successfully by verifying that:

The instance no longer appeared in EC2



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=aws-shell-uddhav-instance" --query "Reservations[*].Instances[*].State.Name" --output text
terminated
```



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

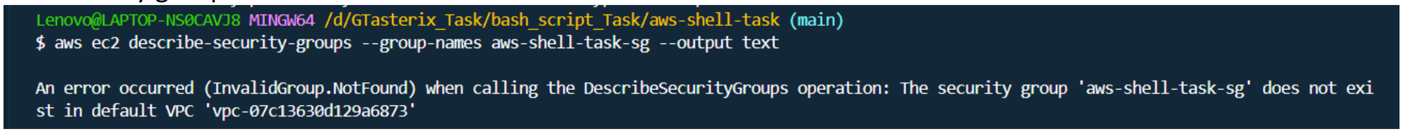
Account ID: 2534-9079-5695 · Asia Pacific (Mumbai) · bash_user

EC2 > Instances

Instances (1/1) Info

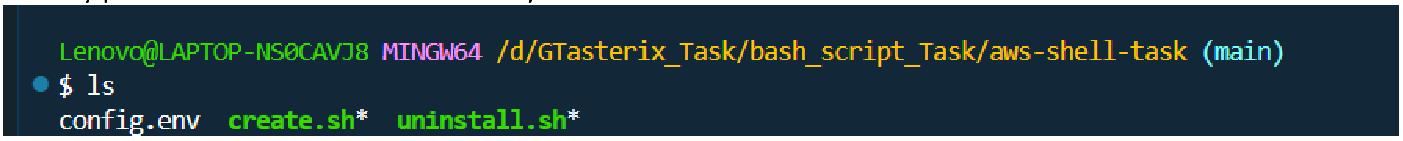
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
aws-shell-uddhav-instance	i-01a3763a5dc6eb8ce	Terminated	t2.micro	–	View alarms +	ap-south-1b	–

The security group was deleted



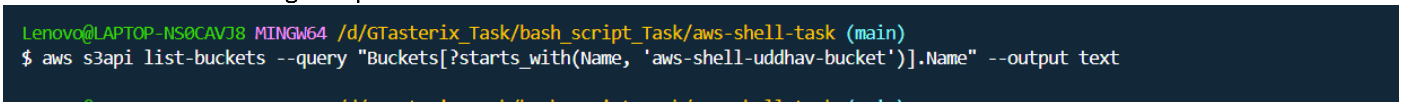
```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ aws ec2 describe-security-groups --group-names aws-shell-task-sg --output text

An error occurred (InvalidGroup.NotFound) when calling the DescribeSecurityGroups operation: The security group 'aws-shell-task-sg' does not exi
st in default VPC 'vpc-07c13630d129a6873'
```

The key pair was removed from AWS and my local folder



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ ls
config.env  create.sh*  uninstall.sh*
```

No S3 buckets with the given prefix remained



```
Lenovo@LAPTOP-NS0CAVJ8 MINGW64 /d/GTasterix_Task/bash_script_Task/aws-shell-task (main)
$ aws s3api list-buckets --query "Buckets[?starts_with(Name, 'aws-shell-uddhav-bucket')].Name" --output text
```

All verifications showed that the cleanup script executed correctly.