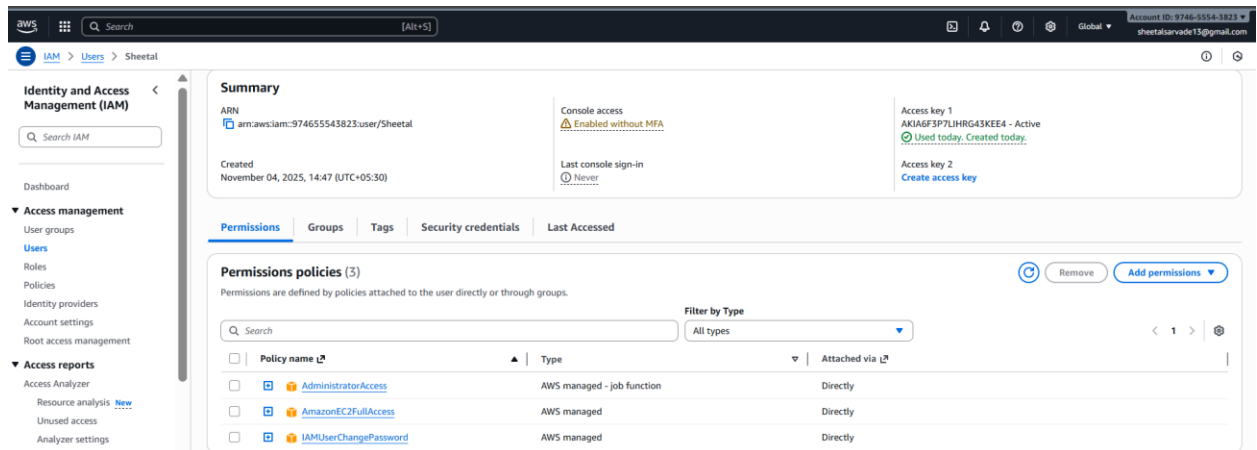


# Create Ec2 Machine using CLI

- Create IAM user and add Access key , Administrative Permissions



- Configure AWS CLI

aws configure

AWS Access Key ID

AWS Secret Access Key

Default region name - us-east-1

Default output format - json

- aws sts get-caller-identity

```
{
  "UserId": "AIDA6F3P7LIHVZAJAW5HU",
  "Account": "974655543823",
  "Arn": "arn:aws:iam::974655543823:user/Sheetal"
}
```

- Attach Permission to IAM user related to EC2
- Script to launch EC2 using CLI

```
#!/bin/bash
# -----
# Script: create-ec2-instance.sh
# Purpose: Launch an EC2 instance with security group & key pair
# Author: CloudOps Engineer
# -----
```

```

# ----- Configuration -----
REGION="us-east-1"
INSTANCE_TYPE="t2.micro"
AMI_ID="ami-0c02fb55956c7d316" # Amazon Linux 2 AMI (update for your
region)
KEY_NAME="my-ec2-key"
SECURITY_GROUP_NAME="my-ec2-sg"
TAG_NAME="CloudOps-EC2-Demo"

# ----- Step 1: Create Key Pair -----
echo "Creating key pair: ${KEY_NAME}"
aws ec2 create-key-pair \
  --region $REGION \
  --key-name $KEY_NAME \
  --query 'KeyMaterial' \
  --output text > "${KEY_NAME}.pem"

chmod 400 "${KEY_NAME}.pem"
echo "Key pair saved as ${KEY_NAME}.pem"

# ----- Step 2: Create Security Group -----
echo "Creating security group: ${SECURITY_GROUP_NAME}"
SG_ID=$(aws ec2 create-security-group \
  --group-name "$SECURITY_GROUP_NAME" \
  --description "Security group for EC2 demo" \
  --region $REGION \
  --query 'GroupId' \
  --output text)

echo "Security Group ID: $SG_ID"

# Add inbound rules for SSH (22) and HTTP (80)
aws ec2 authorize-security-group-ingress --group-id "$SG_ID" --protocol tcp --
port 22 --cidr 0.0.0.0/0 --region $REGION
aws ec2 authorize-security-group-ingress --group-id "$SG_ID" --protocol tcp --
port 80 --cidr 0.0.0.0/0 --region $REGION

# ----- Step 3: Launch EC2 Instance -----
echo "Launching EC2 instance..."
INSTANCE_ID=$(aws ec2 run-instances \
  --image-id $AMI_ID \
  --instance-type $INSTANCE_TYPE \
  --key-name $KEY_NAME \
  --security-group-ids $SG_ID \

```

```

--region $REGION \
--tag-specifications
"ResourceType=instance,Tags=[{Key=Name,Value=$TAG_NAME}]" \
--query 'Instances[0].InstanceId' \
--output text)

echo "Instance ID: $INSTANCE_ID"

# ----- Step 4: Wait for Instance to be Running -----
echo "Waiting for instance to enter 'running' state..."
aws ec2 wait instance-running --instance-ids $INSTANCE_ID --region $REGION

# ----- Step 5: Retrieve Public IP -----
PUBLIC_IP=$(aws ec2 describe-instances \
--instance-ids $INSTANCE_ID \
--region $REGION \
--query "Reservations[0].Instances[0].PublicIpAddress" \
--output text)

echo "Instance is ready!"
echo "Public IP: ${PUBLIC_IP}"
echo "You can SSH using: ssh -i ${KEY_NAME}.pem ec2-user@${PUBLIC_IP}"

# ----- Step 6: Display Summary -----
echo "-----"
echo "EC2 Instance Summary:"
echo "Region:    $REGION"
echo "Instance Type: $INSTANCE_TYPE"
echo "Instance ID: $INSTANCE_ID"
echo "Public IP:   $PUBLIC_IP"
echo "Key Pair:    ${KEY_NAME}.pem"
echo "Security Group: $SG_ID"
echo "Tag:        $TAG_NAME"
echo "-----"

```

- EC2 Machine Launched

The screenshot displays the AWS Management Console for the 'us-east-1' region. The 'Instances' page shows a single EC2 instance named 'CloudOps-EC2-Demo' with ID 'i-09160ddedbc589f98', which is in a 'Running' state. The instance type is 't2.micro' and it is located in the 'us-east-1c' availability zone. Below the console view, a CloudShell terminal window is open, showing the following output:

```
us-east-1 +
Instance ID: i-09160ddedbc589f98
Waiting for instance to enter 'running' state...
Instance is ready!
Public IP: 187.21.135.117
You can SSH using: ssh -i my-ec2-key.pem ec2-user@187.21.135.117
-----
EC2 Instance Summary:
Region: us-east-1
Instance Type: t2.micro
Instance ID: i-09160ddedbc589f98
Public IP: 187.21.135.117
Key Pair: my-ec2-key.pem
Security Group: sg-8c1639cf74adb2b43
Tag: CloudOps-EC2-Demo
-----
myscripts $ !
```