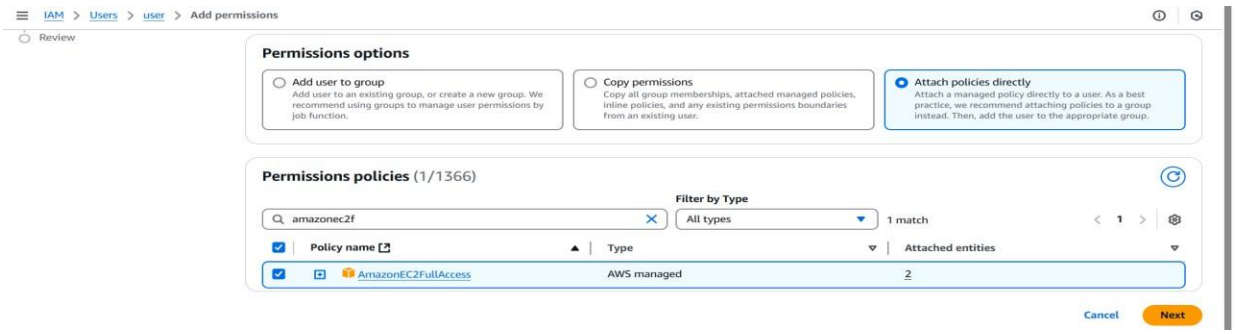
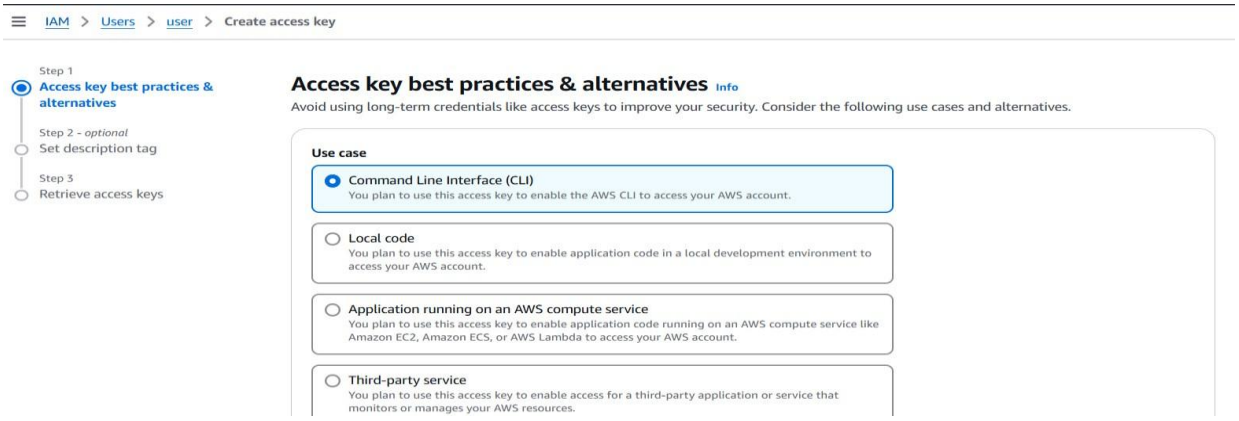
Task 3: An organisation has multiple ec2 instances running in a VPC on AWS. Each instance has multiple tags like “Department” and “Hostname” associated with it. The value of the Department tag is one of Frontend, Backend, DevOps, QA, ITOps, etc. The organisation has requested you to change the Department tags of a few servers and have shared a csv file with the old and new values for Hostname.
Write a python or Java Code for modifying the value of the Department tag by reading the csv file and accordingly changing the values of the following instances.

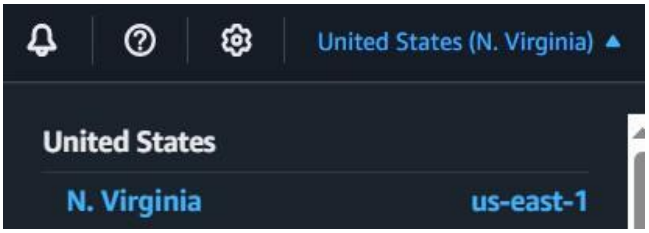
AWS CLI is installed and configured:
open the IAM and configure the process User->Create User->give the user name.
Give full acess to the ec2 user.->create user.



Open the created user and give the access key->Security Credentials->CLI->Create access key.



My Region :



commands:

sudo snap install aws-cli -- classic

```
control@control:~$ snap install aws-cli --classic
error: access denied (try with sudo)
control@control:~$ sudo snap install aws-cli --classic
[sudo] password for control:
aws-cli (v2/stable) 2.26.6 from Amazon Web Services (aws✓) installed
```

Then Configure the AWS with the command : aws configure

```
control@control:~$ aws configure
AWS Access Key ID [None]: AKIA3C6FMEXR5MAKCVME
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
control@control:~$
```

Update the CSV file .

```
control@control:~$ cat > tag_update.csv <<EOF
> Hostname,Current Department,New Department
> webserver01,Backend,Frontend
> webserver02,Frontend,Frontend
> database04,QA,Backend
> ansible,Devops,Backend
> scanner,,Devops
> activedirectory02,ITops,ITops
> database03,Frontend,Backend
> jenkins,,Devops
> EOF
```

Install the python and Boto3 library.

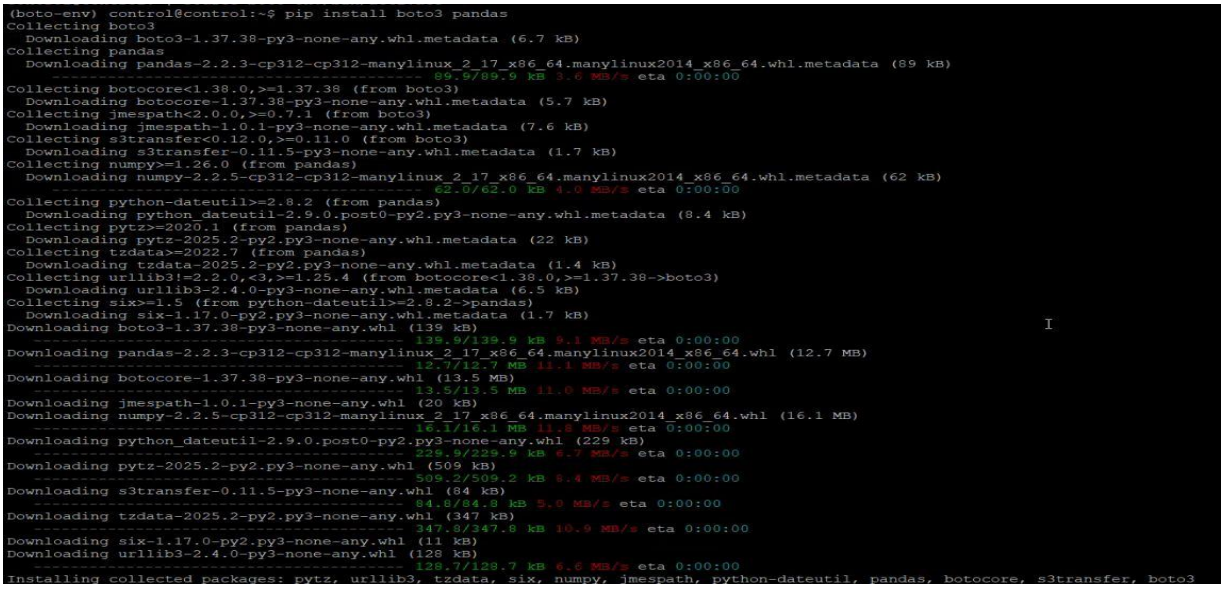
commands:

Sudo apt python3-pip

pip install boto3 pandas

Sudo python3 -m venv boto -env

python3 -m venv boto -env



For Creating the instance:
commands:
**aws ec2 run-instances **

**--image-id ami-0c02fb55956c7d316 **

**--instance-type t2.micro **

**--key-name my-key **

**--security-groups my-security-group **

--region us-east-1

TO RUN THE INSTANCE WITH THE TAG :
Then the department will correctly.

**aws ec2 run-instances **

**--image-id ami-0c02fb55956c7d316 **

**--instance-type t2.micro **

**--key-name my-key **

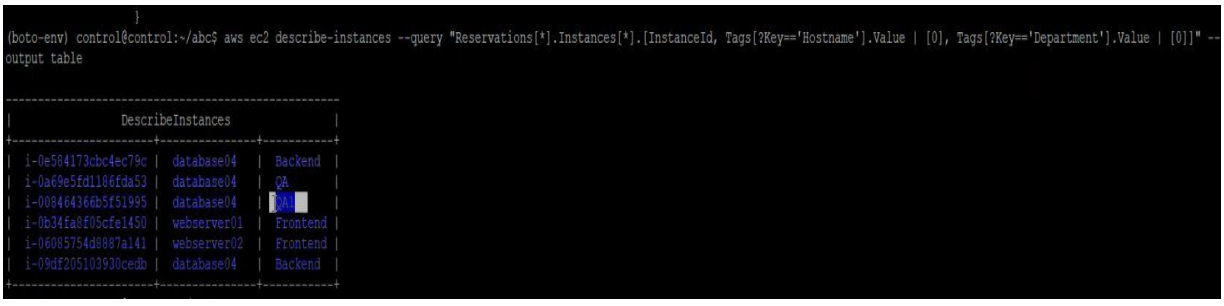
**--security-groups default **

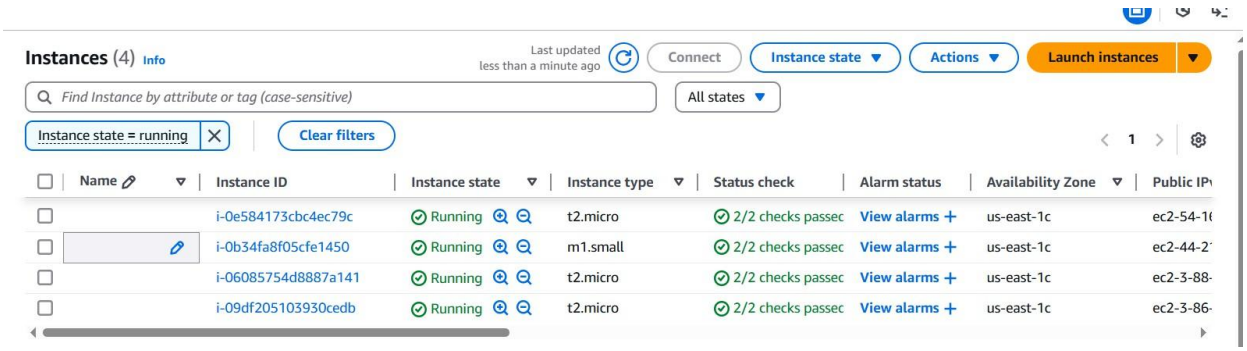
**--tag-specifications 'ResourceType=instance,Tags=[{Key=Hostname,Value=database04},{Key=Department,Value=QA1}]' **

--count 1

TO RUN THE COMMANDS FOR THE OUTPUT TABLE:

aws ec2 describe-instances \--query "Reservations[*].Instances[*].[InstanceId, Tags[?Key=='Hostname'].Value | [0], Tags[?Key=='Department'].Value | [0]]" \ --output table





update python.py

update python.py script is used to see the AWS Console and check the tags of those EC2 instances manually.

command:

aws ec2 describe-instances --region us-east-1 --query "Reservations[*].Instances[*].[InstanceId,Tags]"

Run this command to see the tags for the specific id value.

command:

aws ec2 describe-tags --filters "Name=resource-id,Values=i-06085754d8887a141"

```
(boto-env) control@control:~/abc$ vim tag_update.csv
(boto-env) control@control:~/abc$  aws ec2 describe-tags --filters "Name=resource-id,Values=i-0b34fa8f05cfe1450"
{
    "Tags": [
        {
            "Key": "Department",
            "ResourceId": "i-0b34fa8f05cfe1450",
            "ResourceType": "instance",
            "Value": "Frontend"
        },
        {
            "Key": "Hostname",
            "ResourceId": "i-0b34fa8f05cfe1450",
            "ResourceType": "instance",
            "Value": "webserver01"
        }
    ]
}
```

python de.py

Loading the CSV correctly. Using filters to identify the EC2 instances by Hostname and current Department.Updating the Department tag if the match is found.

```
(boto-env) control@control:~/abc$ python de.py
CSV Data Loaded:
          Hostname Current Department New Department
0       webserver01            Backend       Frontend
1       webserver02           Frontend       Frontend
2        database04                 QA        Backend
3           ansible            Devops        Backend
4           scanner               NaN         Devops
5  activedirectory02             ITops          ITops
6        database03           Frontend        Backend
7           jenkins               NaN         Devops
Processing instance webserver01 with current department: Backend and new department: Frontend
No matching instance found for webserver01 with Department Backend
Processing instance webserver02 with current department: Frontend and new department: Frontend
Found instance i-06085754d8887a141. Updating department tag...
Updated instance i-06085754d8887a141 with new Department: Frontend
Processing instance database04 with current department: QA and new department: Backend
Found instance i-0a69e5fd1186fda53. Updating department tag...
Updated instance i-0a69e5fd1186fda53 with new Department: Backend
Processing instance ansible with current department: Devops and new department: Backend
No matching instance found for ansible with Department Devops
Skipping scanner because it has no current department value.
Processing instance activedirectory02 with current department: ITops and new department: ITops
No matching instance found for activedirectory02 with Department ITops
Processing instance database03 with current department: Frontend and new department: Backend
No matching instance found for database03 with Department Frontend
Skipping jenkins because it has no current department value.
```

TASK 3 OUTPUT:

Modifying the value of the Department tag by reading the csv file and accordingly changing the values.