https://linkedin.com/in/prafulpatel16

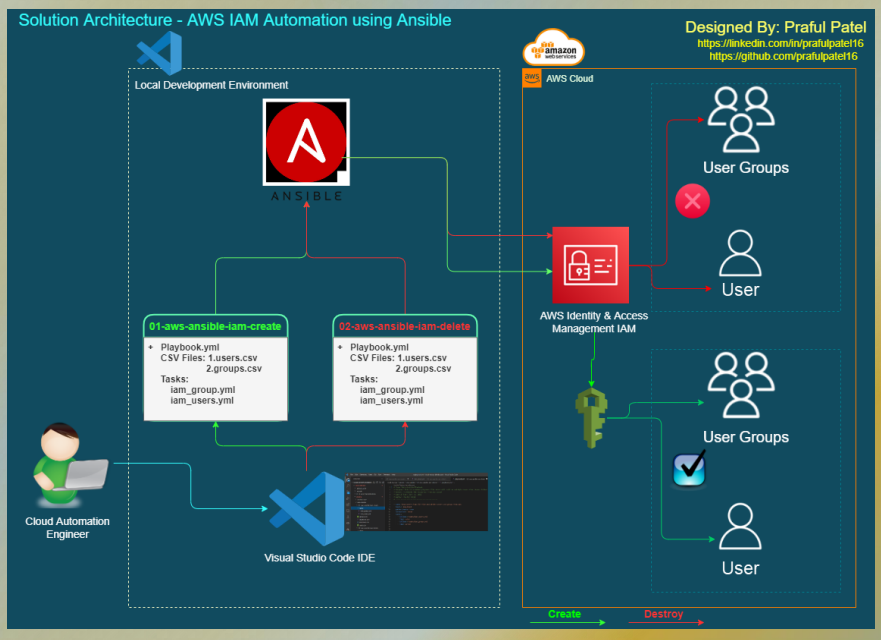
https://github.com/prafulpatel16

AWS - ANSIBLE AUTOMATION PROJECT

CHALLENGE: AWS IAM USERS & GROUP CREATION & DESTROY

USING ANSIBLE 100% AUTOMATION

SOLUTION DESIGN & IMPLEMENTATION BY: PRAFUL PATEL



Date: June 10, 2022

- **Project:**

  AWS IAM USERS & GROUPS CREATION & DESTROY USING ANSIBLE 100% AUTOMATION

- **Project Description:**

  Solution: IAM Automation Solution using Ansible Tool
  Cloud: AWS Cloud
  Cloud Services: Identity & Access Management (IAM)
  Automation Tool: Ansible

  An IT services provider, PRAfect Systems Inc., is engaged in providing Cloud/DevOps & software development solutions. The company recently migrated its entire workload to the AWS Cloud. So Along with all the application and database servers, there was a challenge that they had to create 1000 users who needed to assign to the relevant groups and roles and it's a challenge to create manually and time-consuming tasks which could be error-prone.

  Solution:

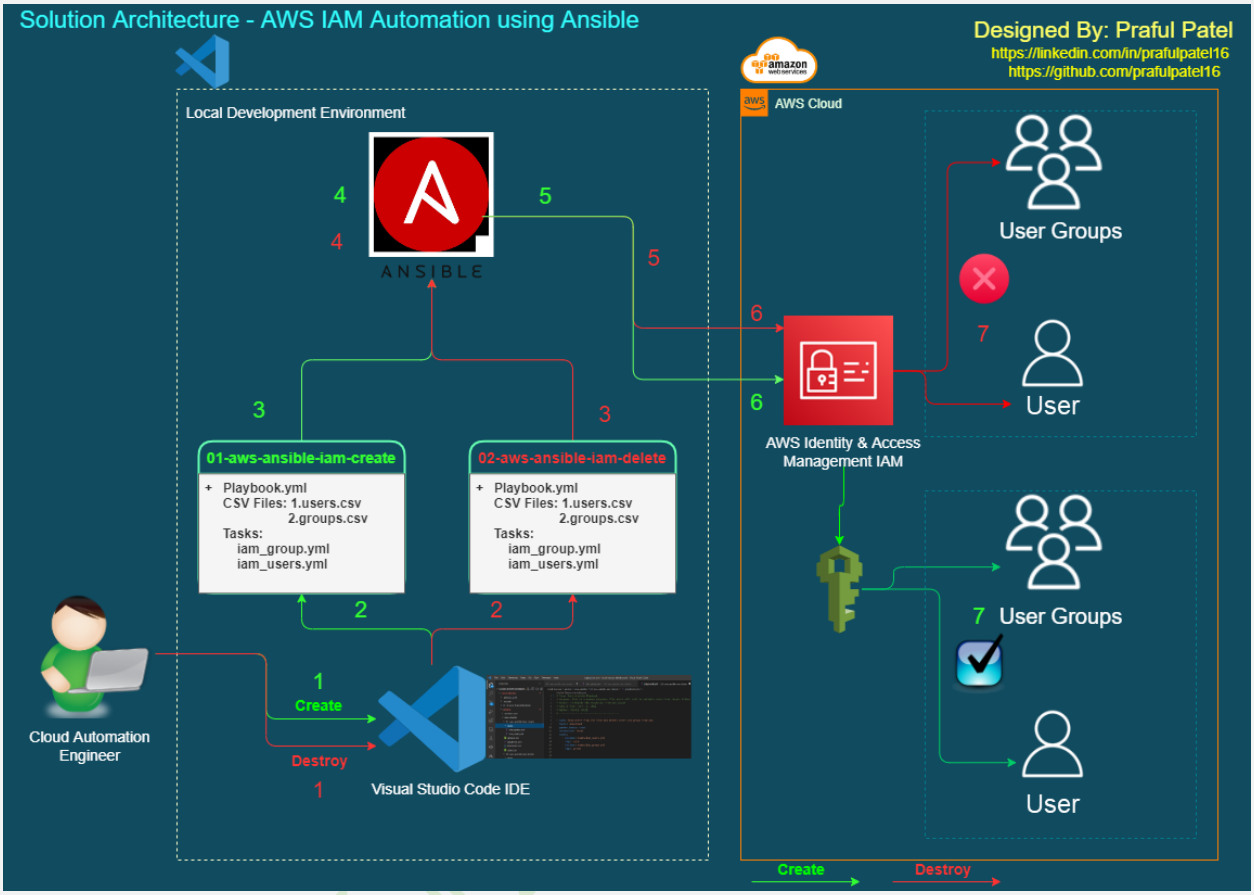
  This project demonstrates an experience of creating an automation solution using an Ansible playbooks for AWS IAM users and group creation on AWS cloud.

- **Project Cost Estimation:**
  (Note: This cost is Not any actual cost, it's just an estimation based on high level requirement. Price may be vary based on adding and removing services based on requirement.)

- **Tools & Technologies covered:**

  - AWS Cloud
  - AWS Identity & Access Management (IAM)
  - Ansible (Configuration Management Automation Tool)
  - Visual studio code IDE
  - GitHub
  - GitBash
  - Draw.io

- **Solution Architecture:**



This project will be completed in following implementation phases.

- **Project implementation Phase:**

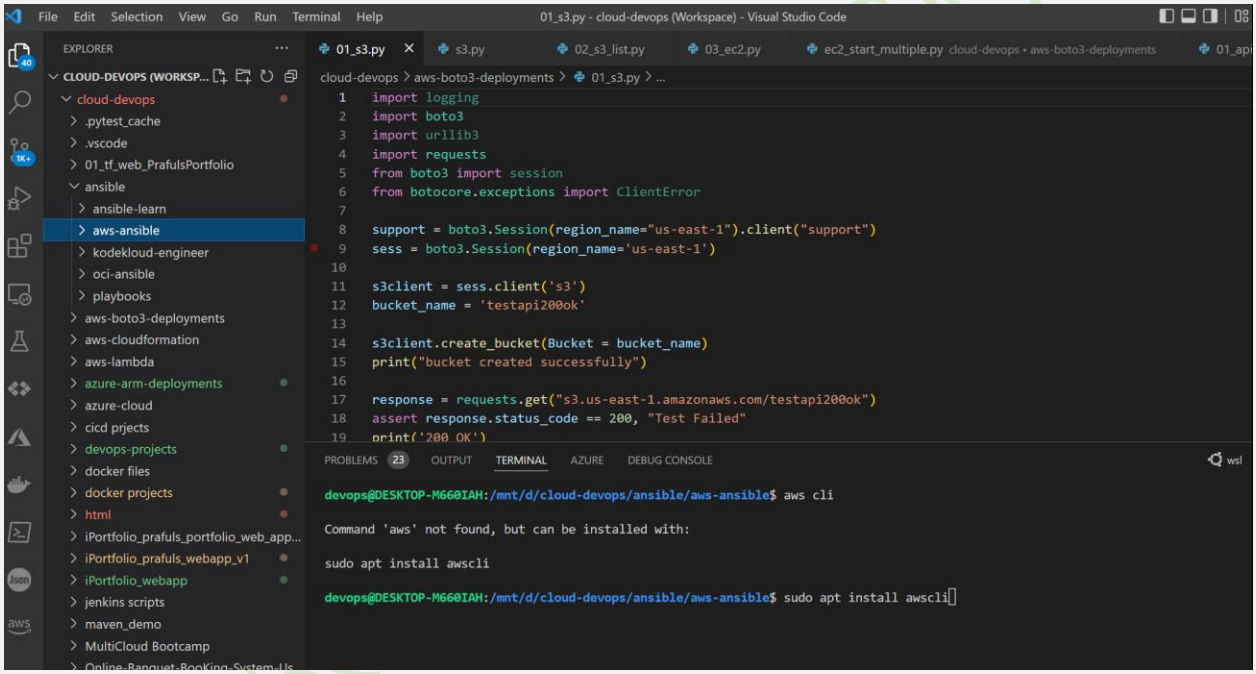
  - Phase 1: Development IDE configuration
  - Phase 2: AWS CLI and Ansible configuration
  - Phase 3: Create an IAM group and users in AWS using Ansible
  - Phase 4: Remove an IAM group and users in AWS using Ansible

➢ Implementation in an Action:

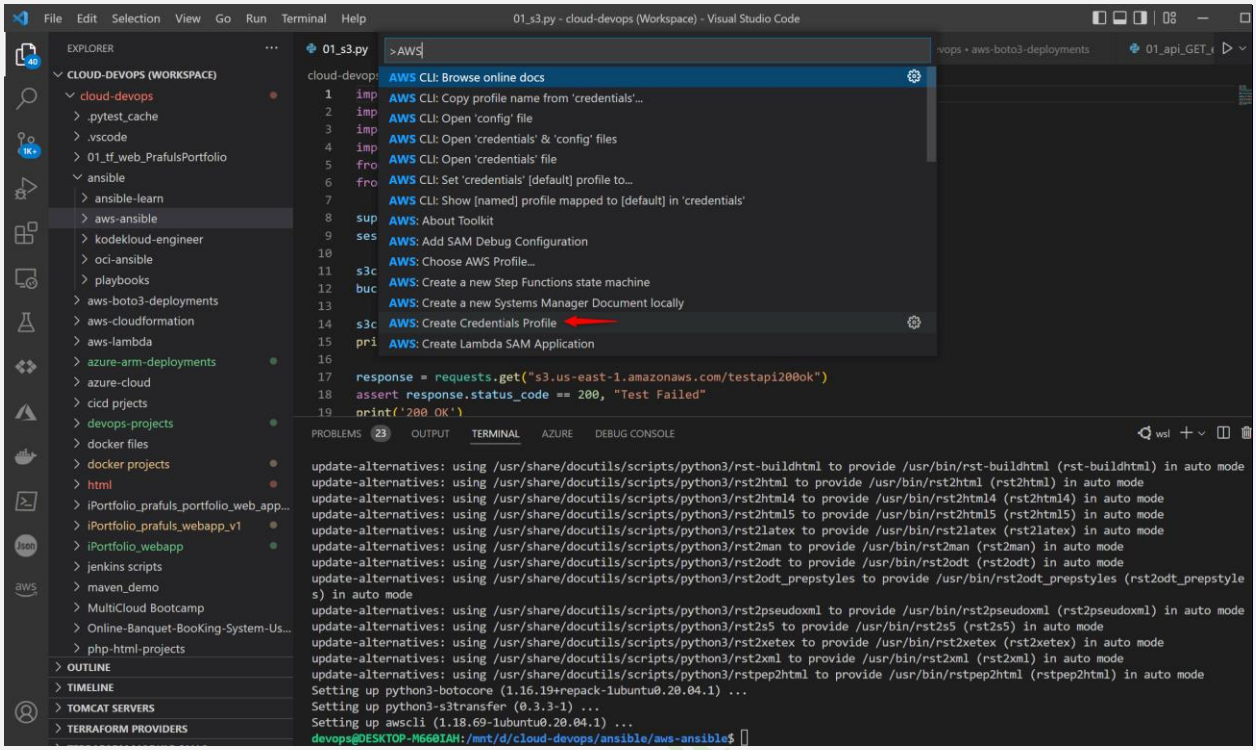
Phase 1: Development IDE configuration

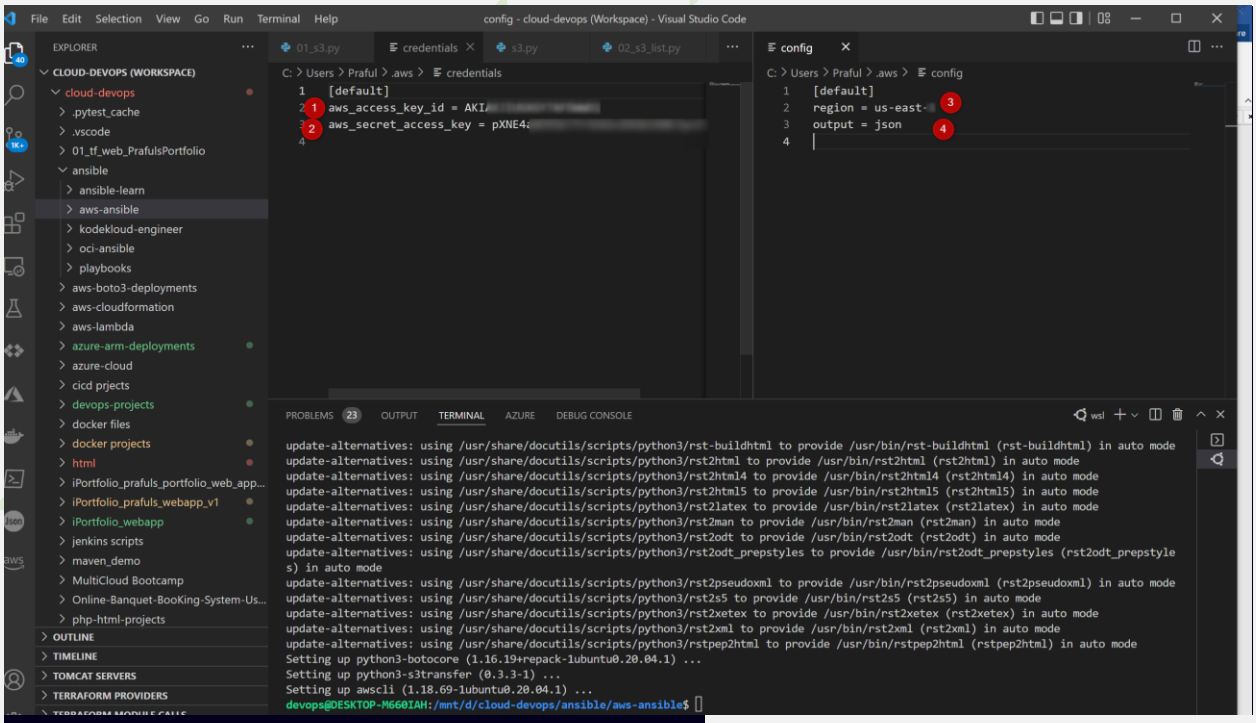
✓ Install AWS cli in VS code Ubuntu WSL
Sudo apt install awscli
Pip install boto

✓ Configure aws credentials in to vscode
Sudo apt install ansible



Configure Aws Profile

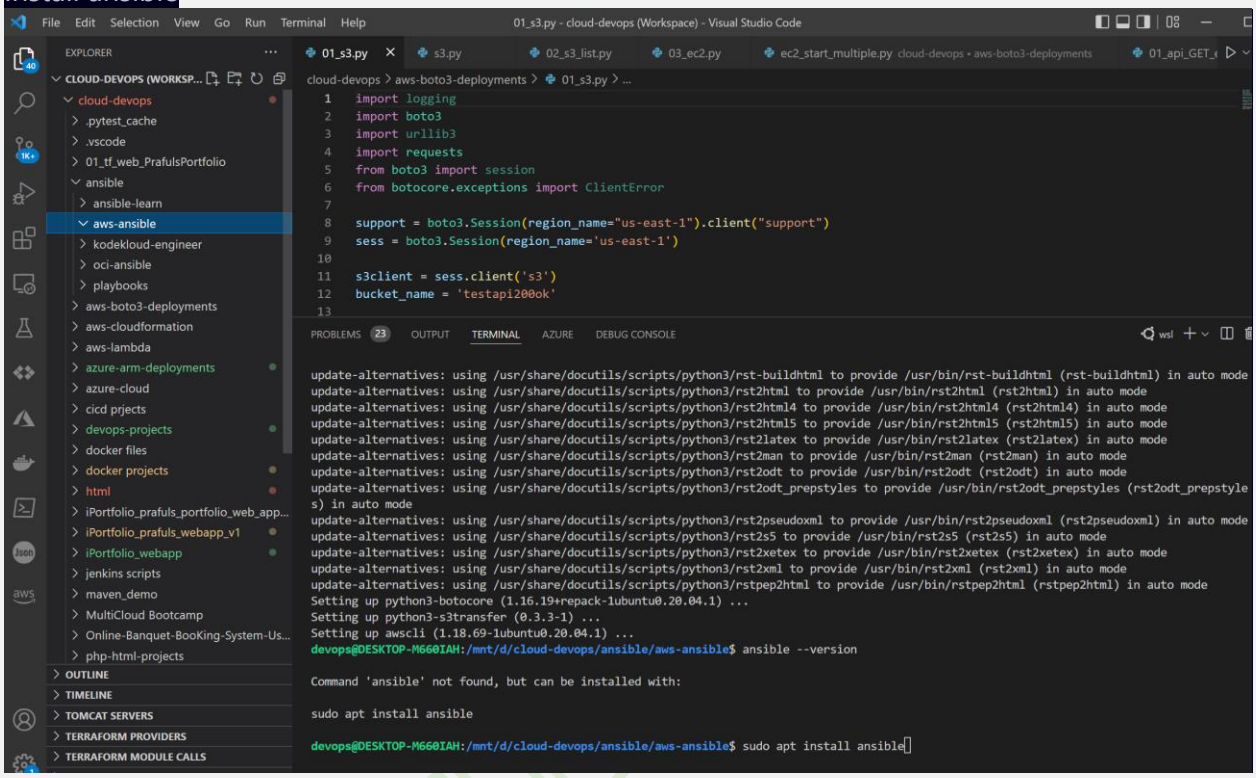




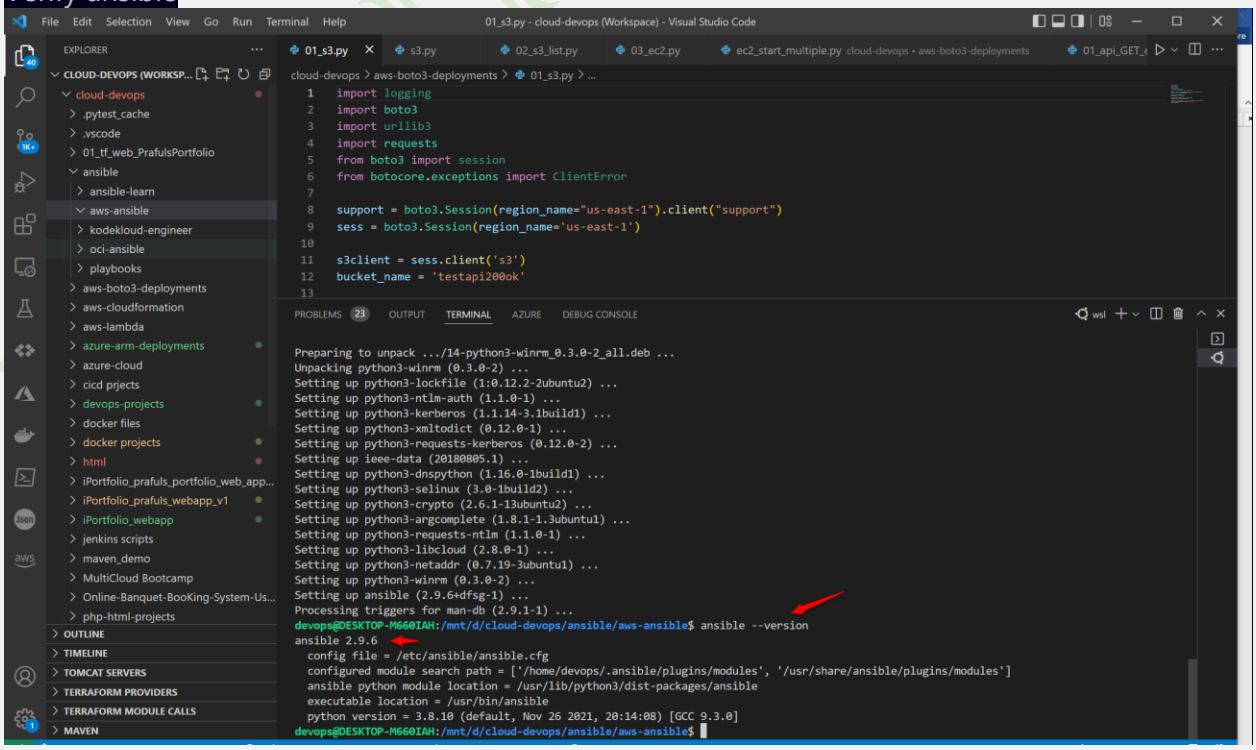
Source: https://github.com/xchangebit/ansible-aws-cli

- Phase 2: AWS cli and Ansible configuration

Install ansible

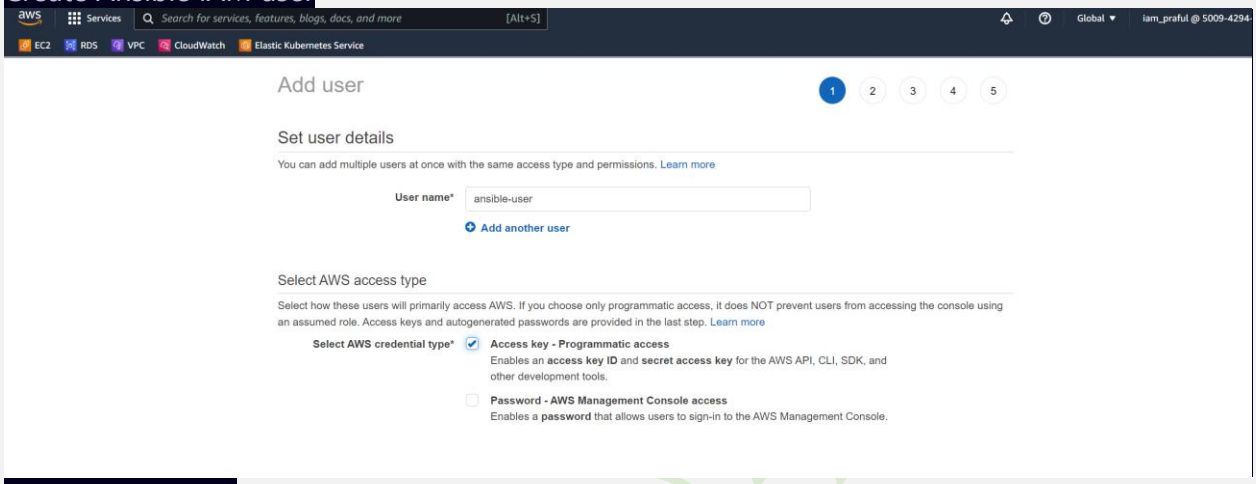


Verify ansible

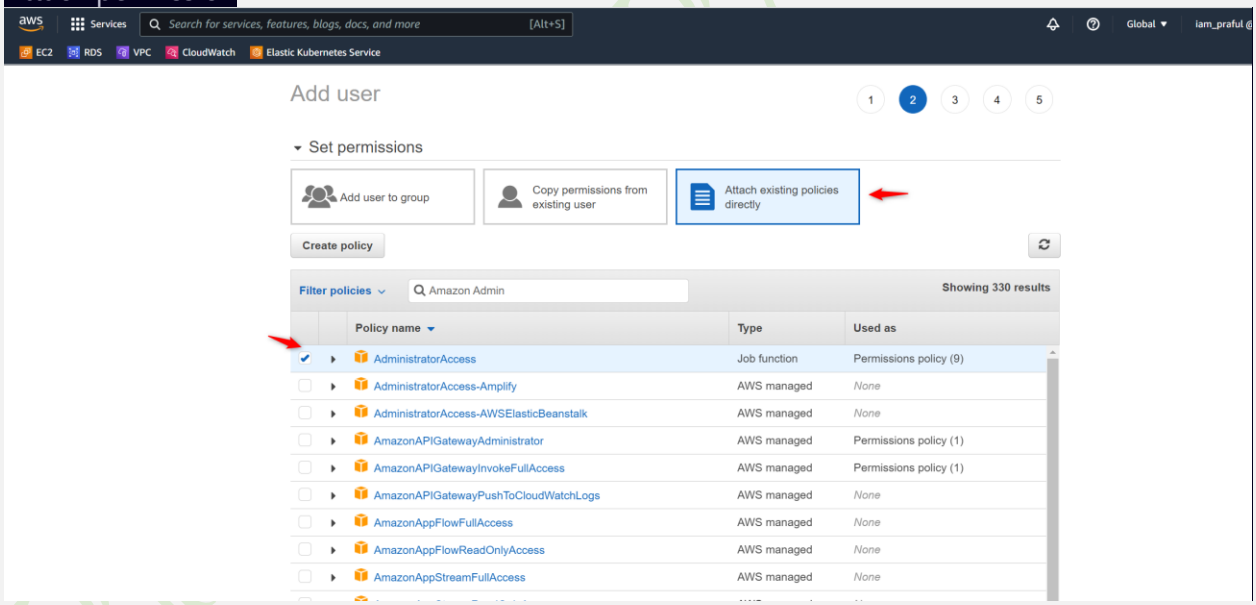


https://docs.ansible.com/ansible/latest/collections/community/aws/iam_user_module.html

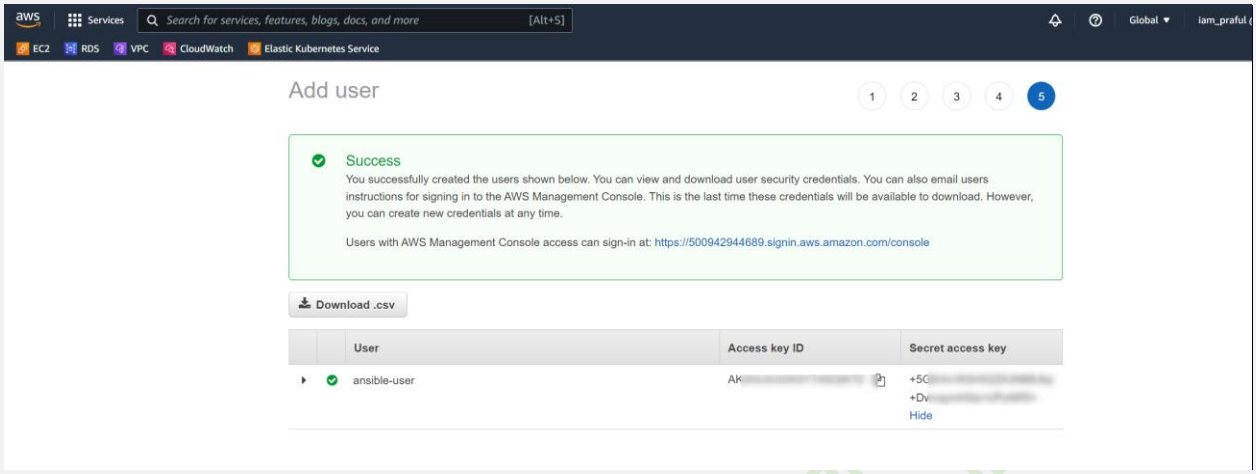
Create Ansible IAM user



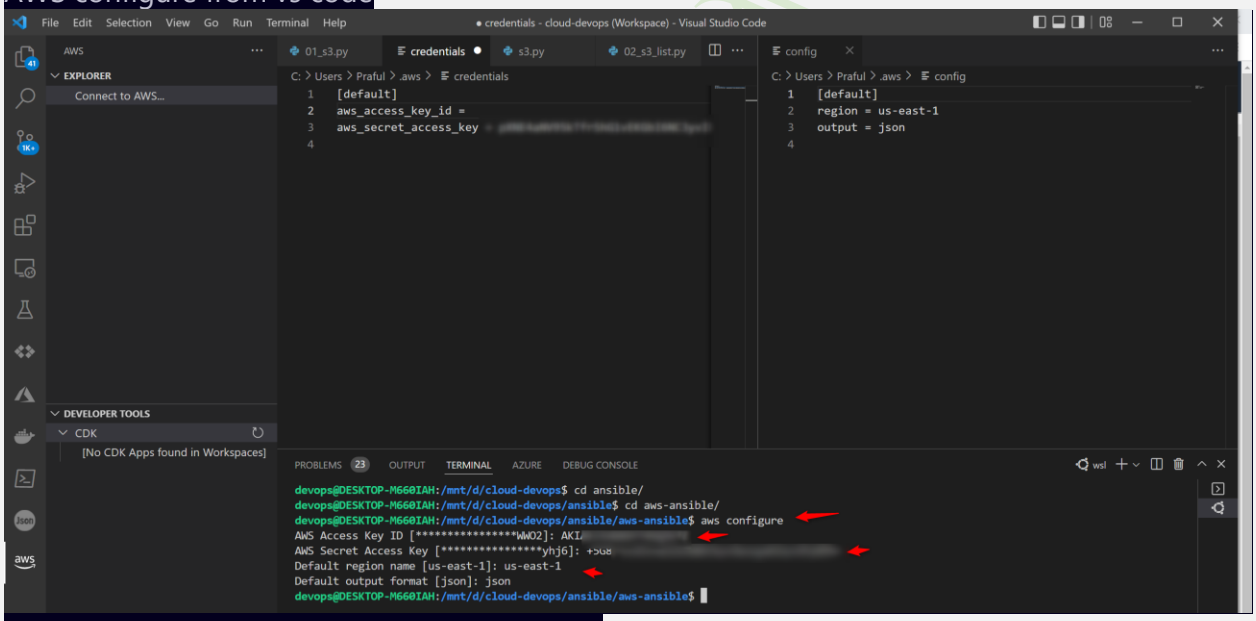
Attach permission



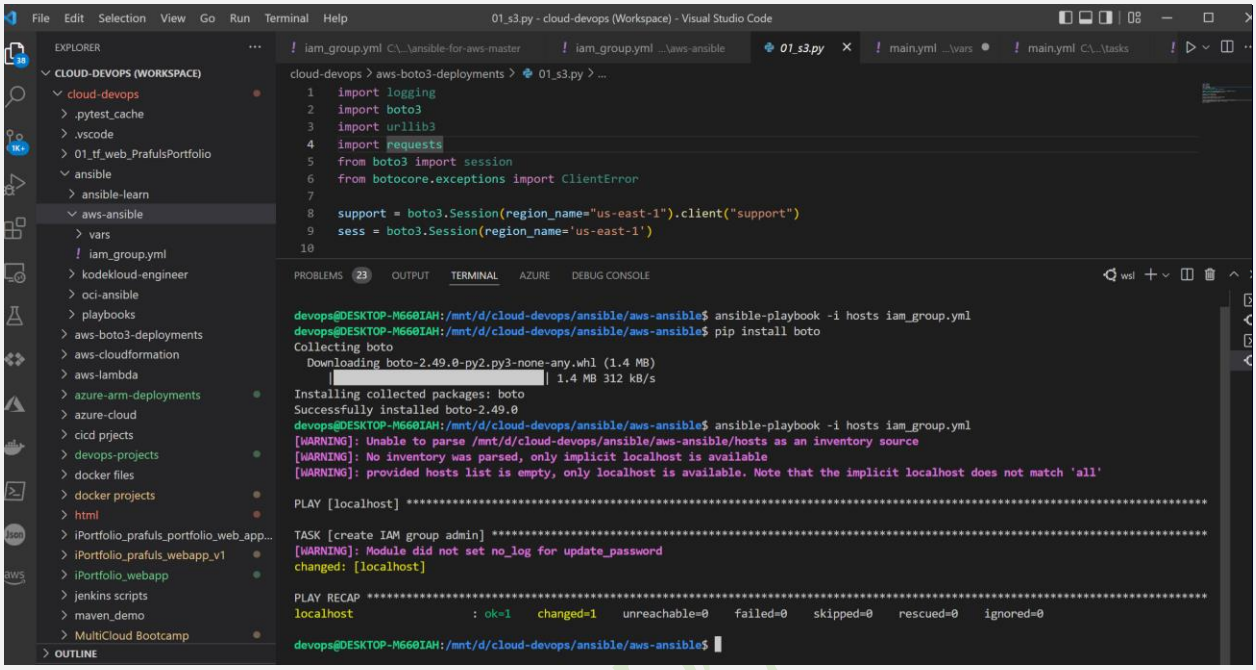
Copy secret key and access key



AWS configure from vs code

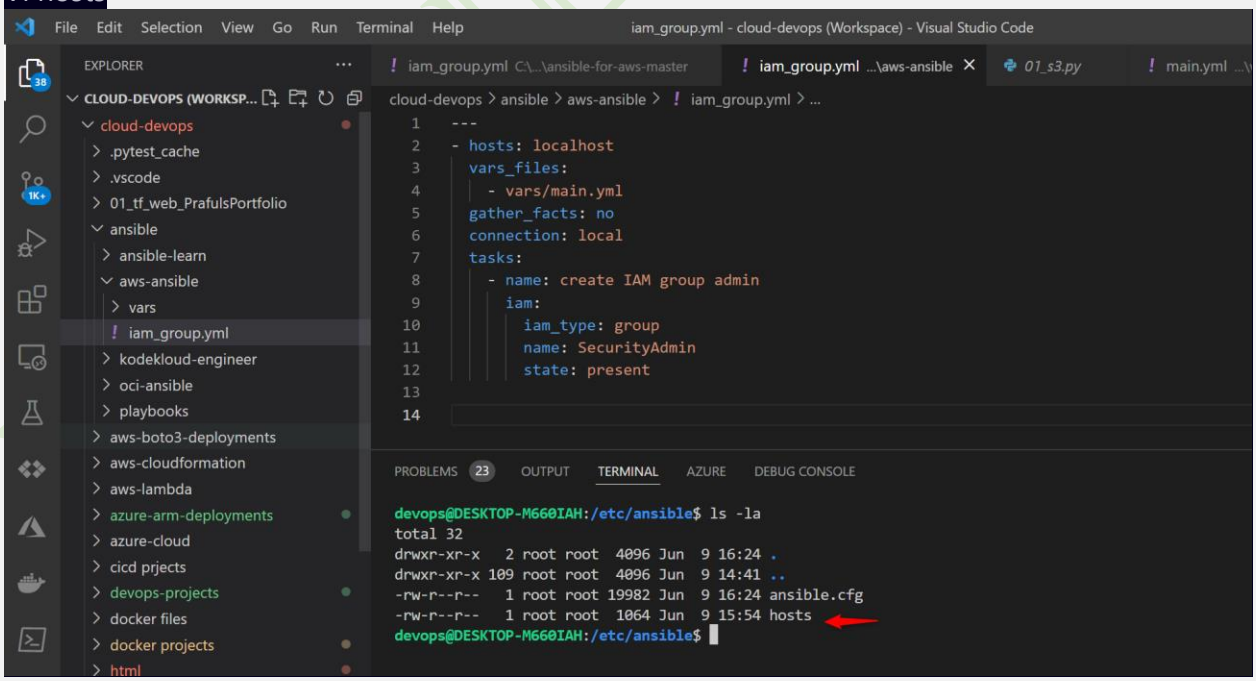


https://github.com/xchangebit/ansible-aws-cli
Install
Pip install boto



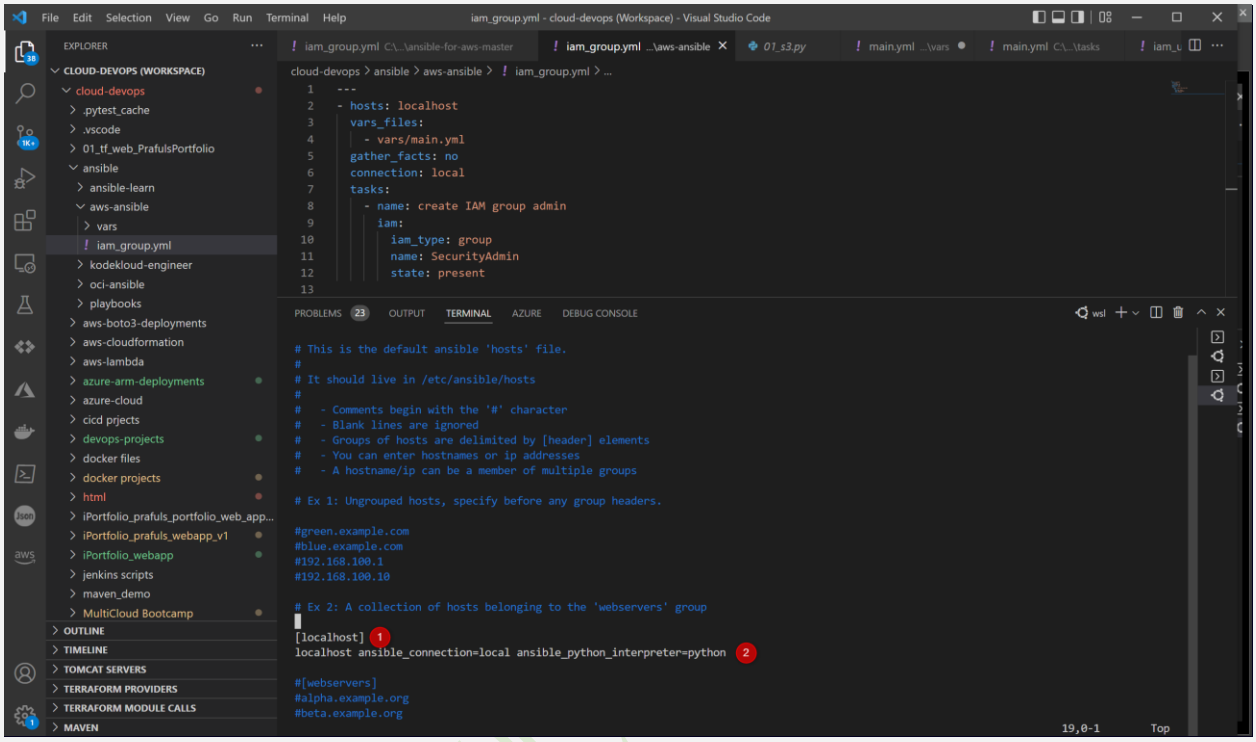
Configure Inventory

Cd /etc/ansible
Vi hosts



Add this line to locahost
[localhost]

localhost ansible_connection=local ansible_python_interpreter=python



Create vars directory for aws credentials
Create main.yml

```
---
aws_system_user: root
aws_profile: default
aws_access_key: "<aws-accesss-key>"
aws_secret_key: "<aws-secret-key>"
aws_region: eu-west-2
aws_format: table
```

Export aws credentials

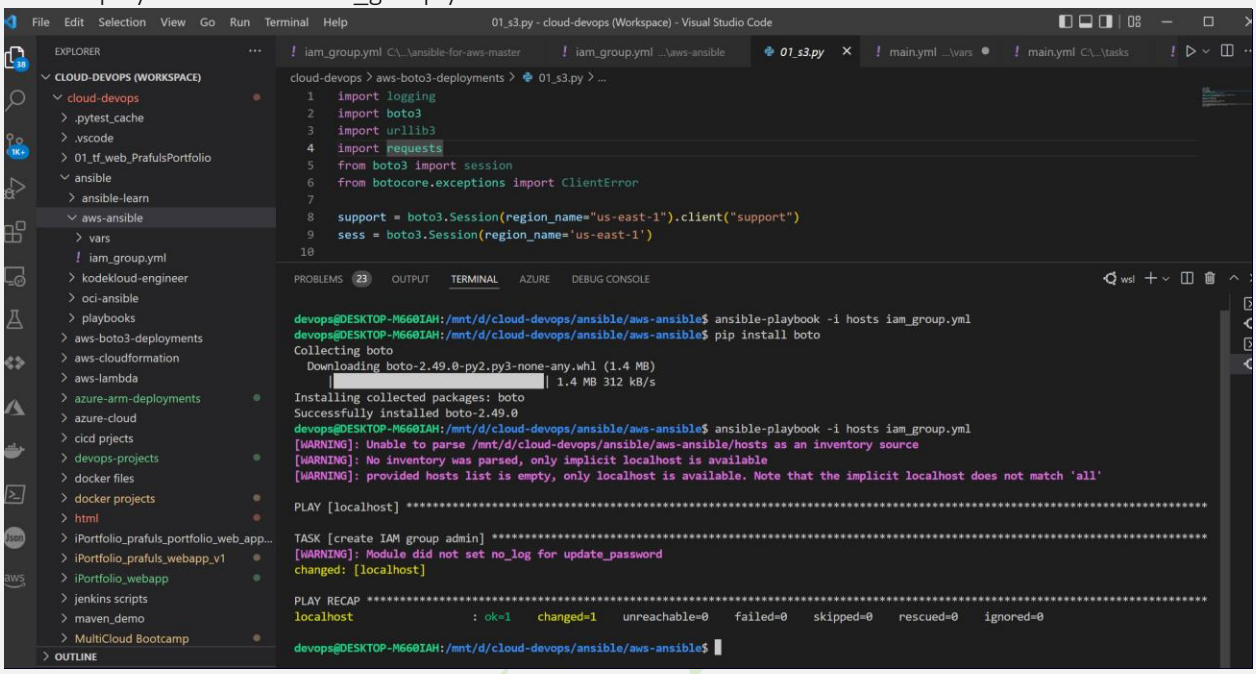
```
export AWS_ACCESS_KEY_ID={access key id}

export AWS_SECRET_ACCESS_KEY={secret access key}
```

- Test the sample script and connection between Ansible local with AWS

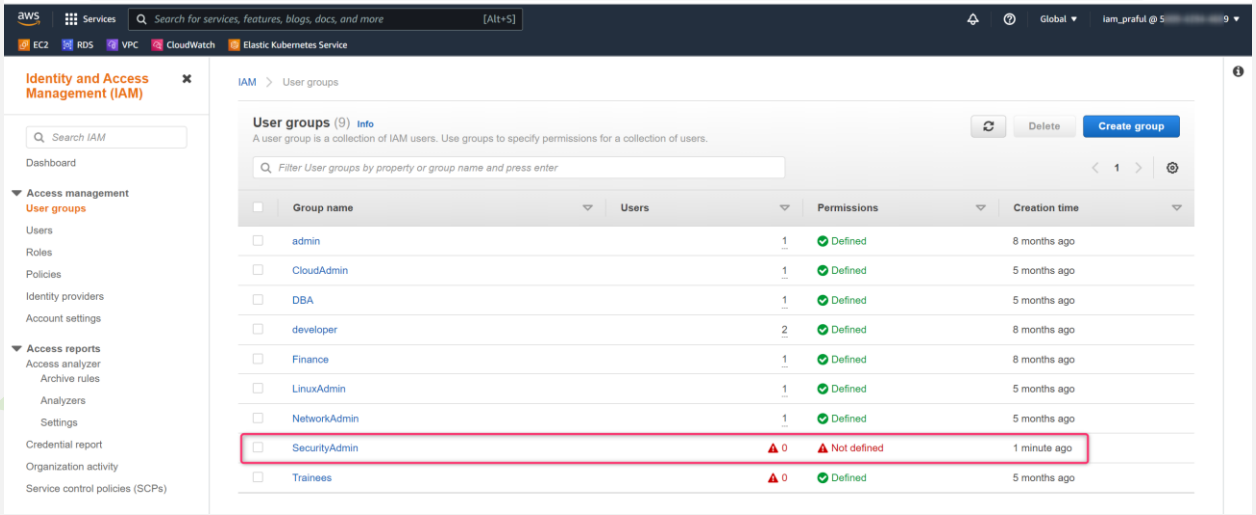
Group Creation: SecurityAdmin

Run ansible playbook
Ansible-playbook –I hosts iam_group.yml

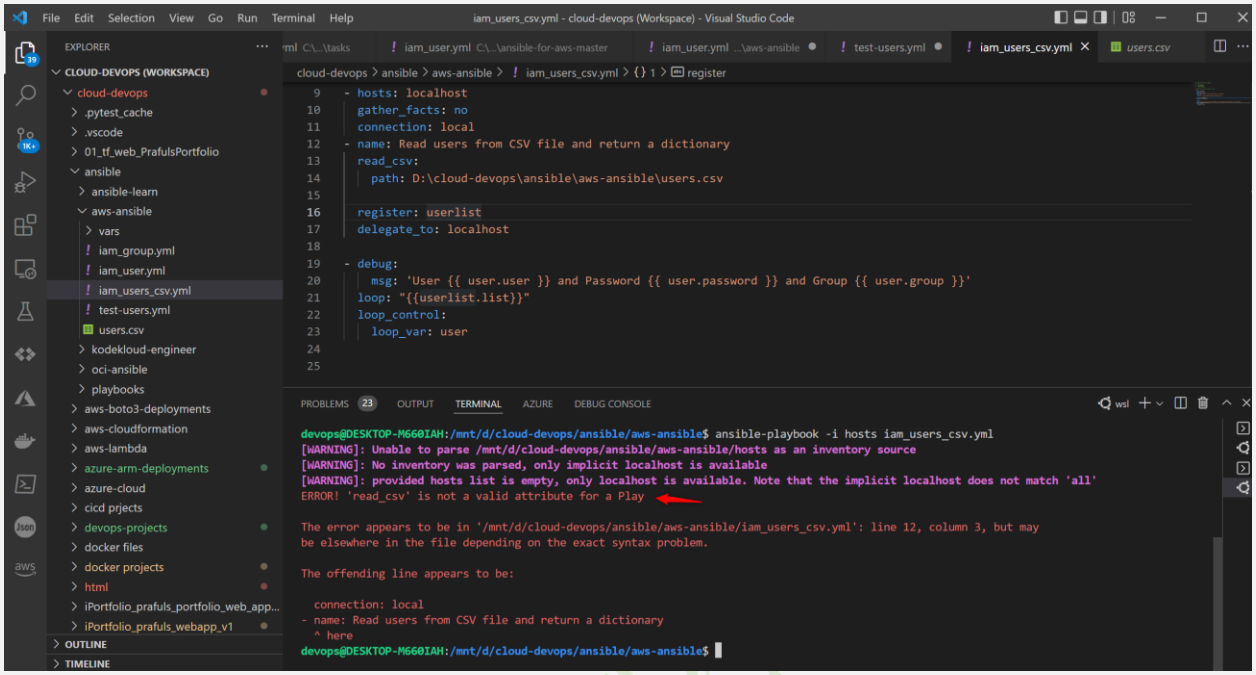


Playbook execution successful

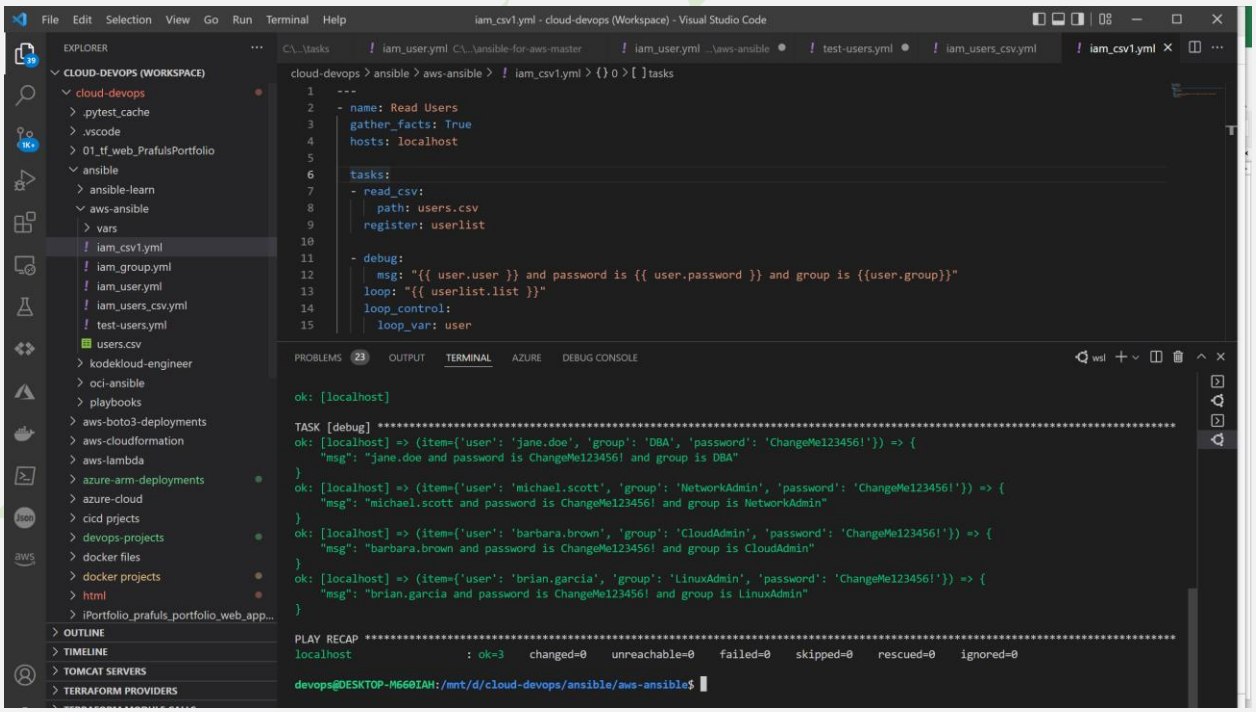
Verify that User Group has been created in AWS



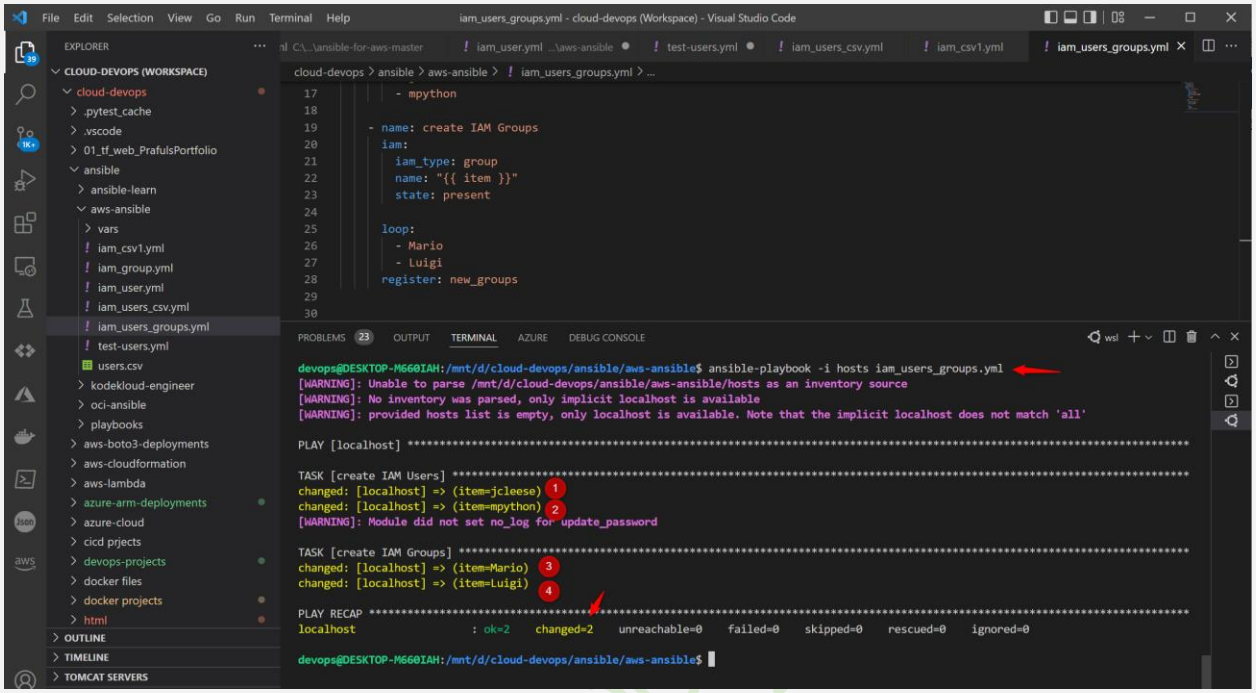
Error



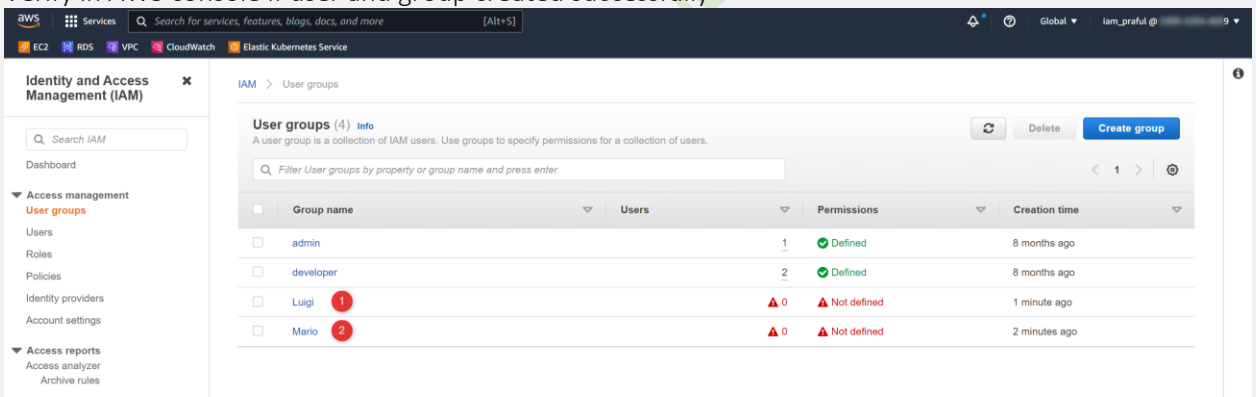
Test 2

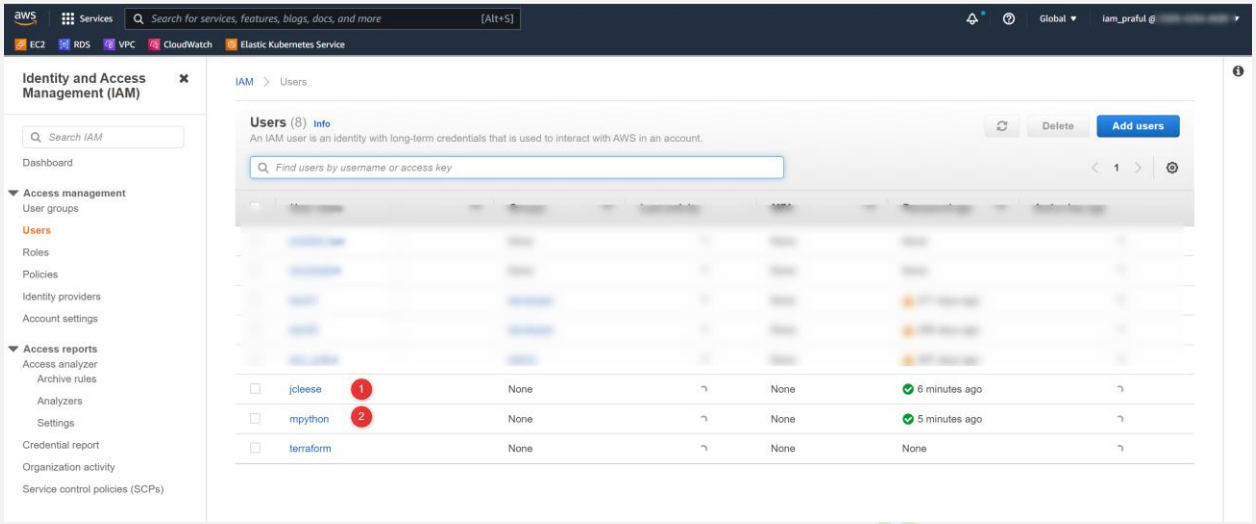


Test 3 Create Users and Groups



Verify in AWS console if user and group created successfully





Note: The above solution was just a research and experiment for the particular use case

- Phase 3: Create IAM group and users in AWS using Ansible

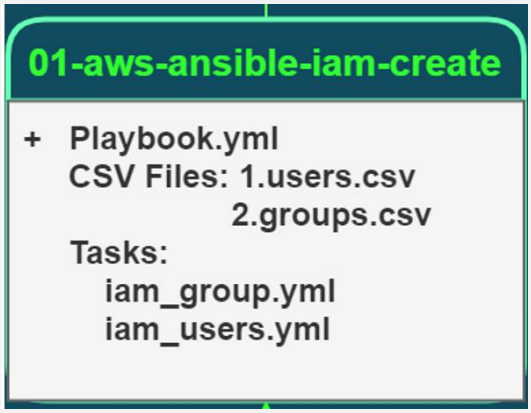
➢ Final Working Solution:

Solution 1: Create IAM group and users in AWS using Ansible

Implementation steps:

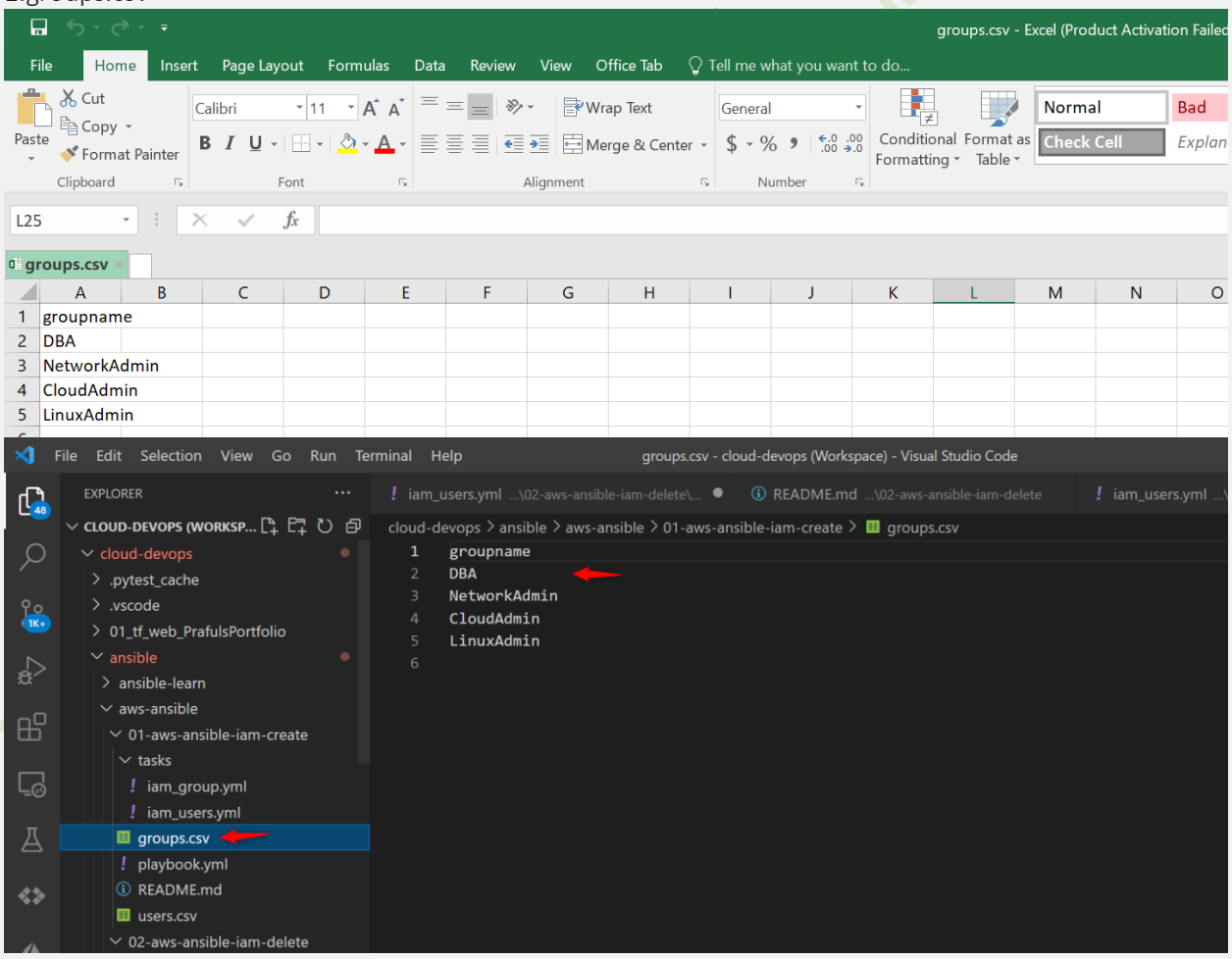
1.Create main working directory
2.Prepare groups.csv and users.csv data files
3.Create sub directory 'tasks'
4.Create two separate tasks file inside 'tasks' folder
    1. create iam_group.yml
    2. create iam_users.yml
5.Create main playbook file & include tasks folders file
    Playbook.yml
6.Run Playbook
7.Verify that users and group are created in aws cloud

1.Create a main working directory: 01-aws-ansible-iam-create

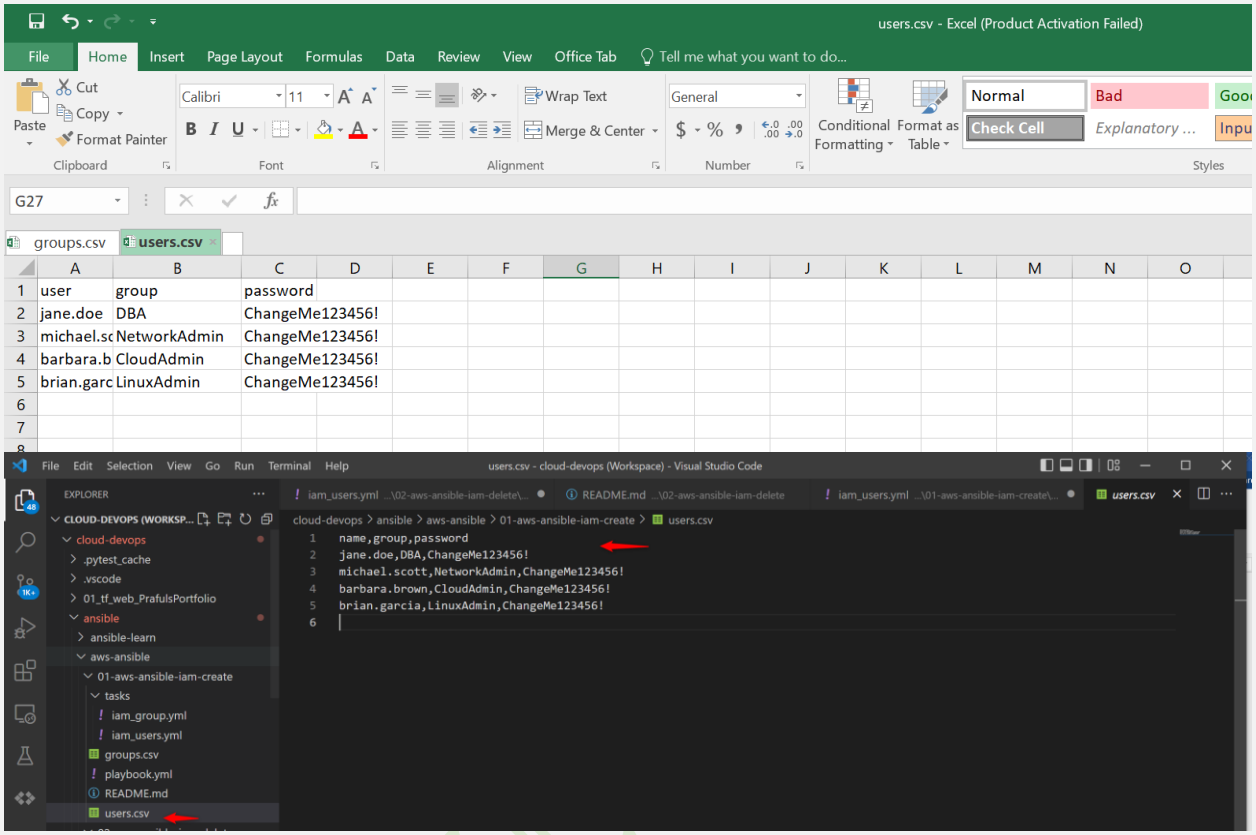


2.Prepare groups.csv and users.csv data files

Prepare CSV files:
1.groups.csv



2.users.csv



3.Create sub directory 'tasks'

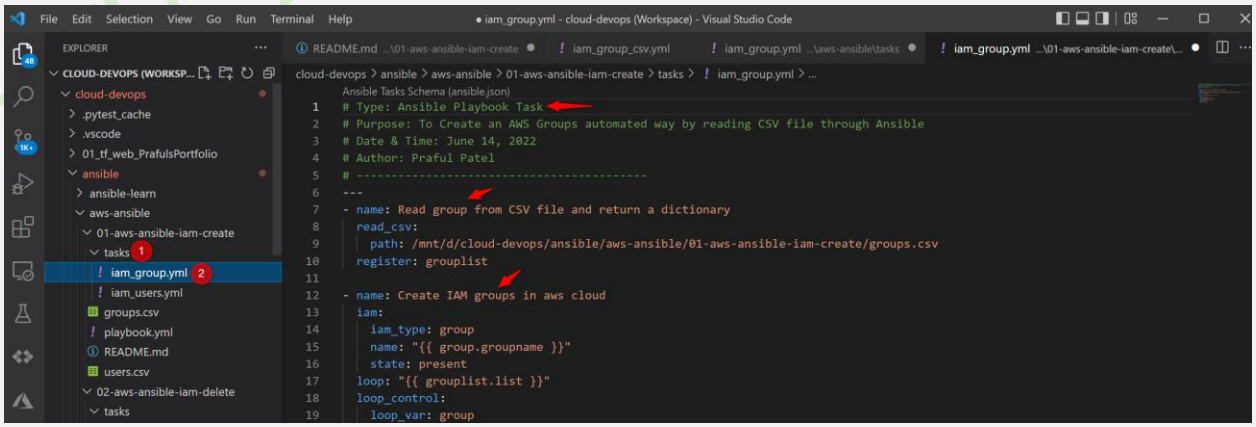
Folder: tasks ;

01-aws-ansible-iam-create/tasks/

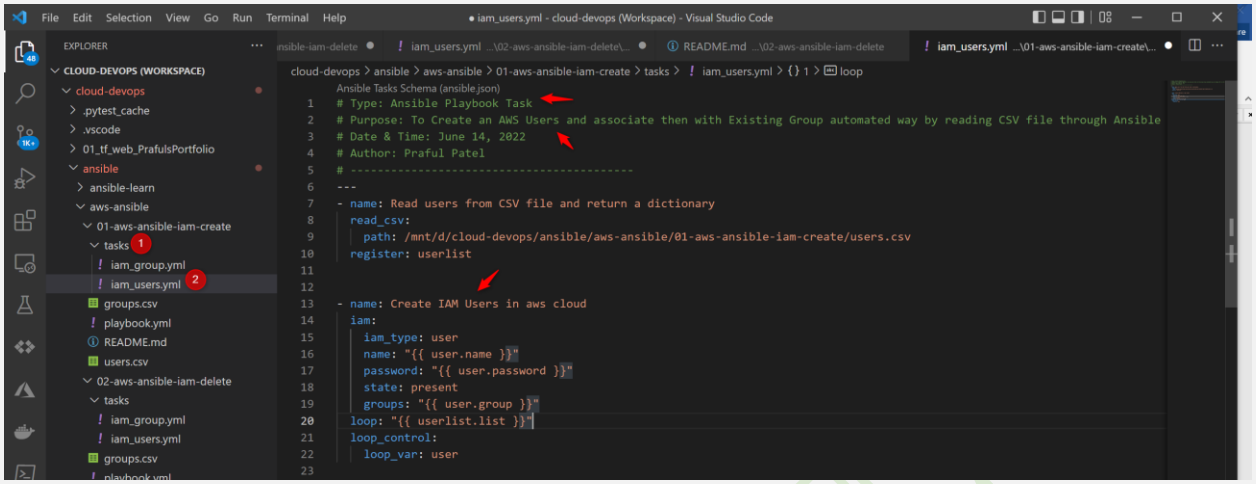
4.Create two separate tasks file inside 'tasks' folder

1. iam_group,yml
2. iam_users.yml
3. iam_group_policy.yml

1.Iam_group.yml



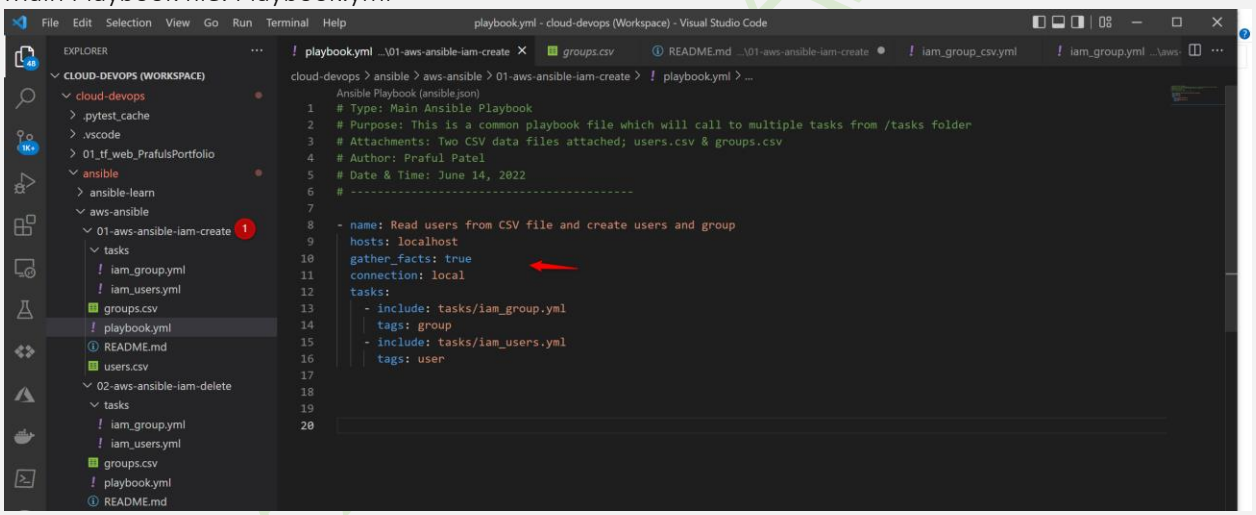
2.iam_users.yml



5.Create main playbook file & include tasks folders file

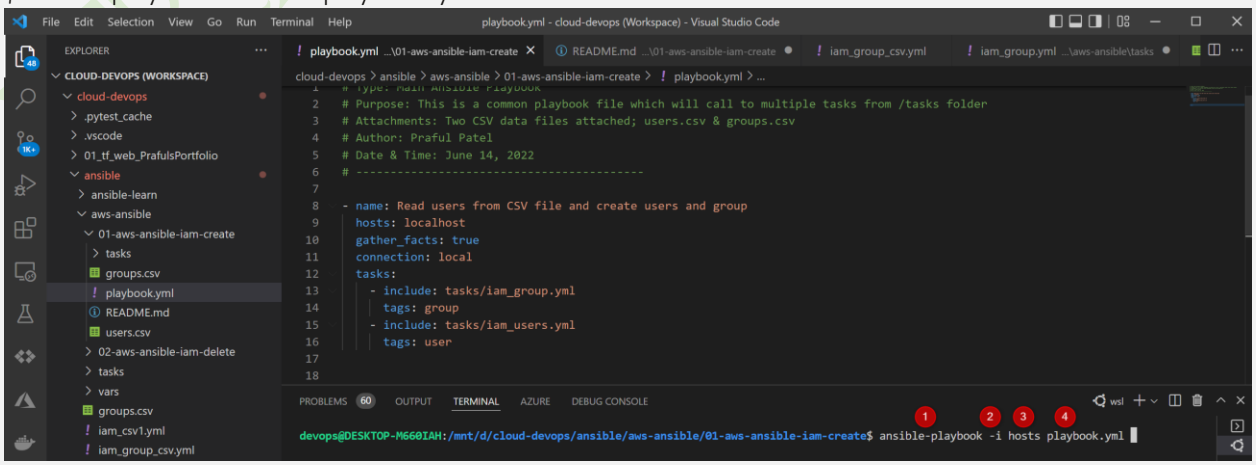
Playbook.yml

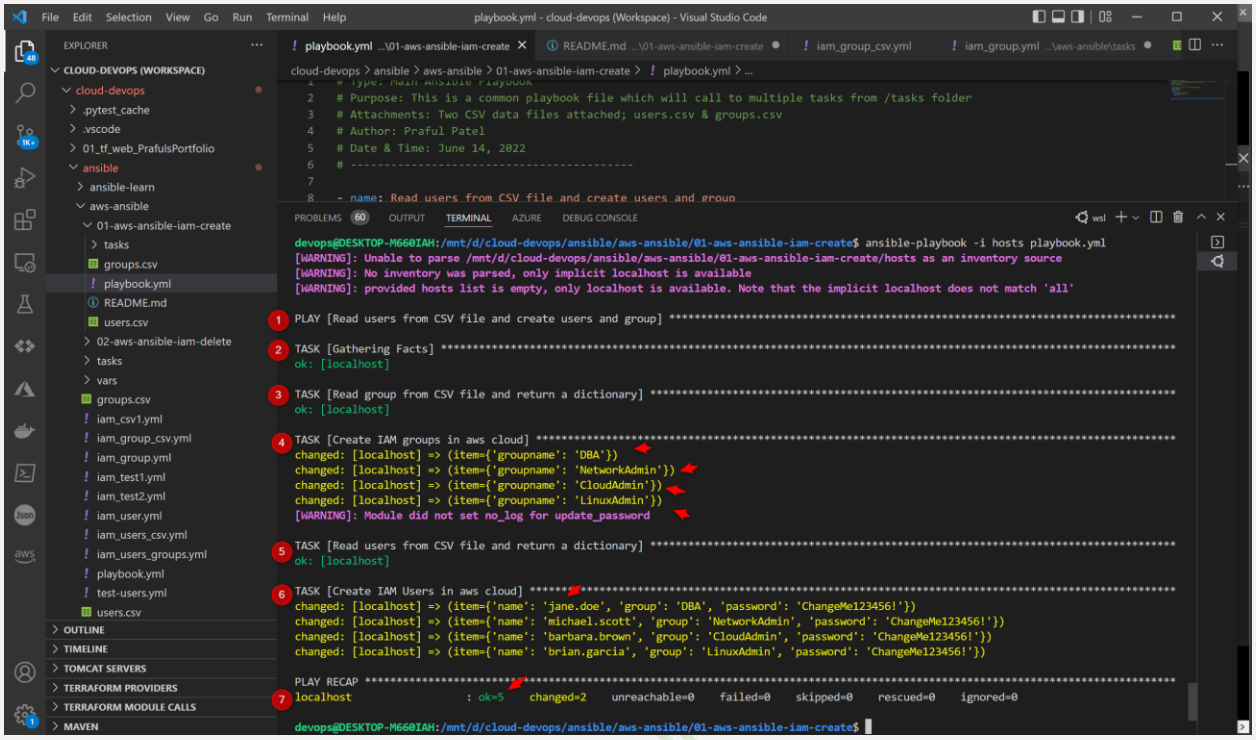
Main Playbook file: Playbook.yml



6.Run Playbook

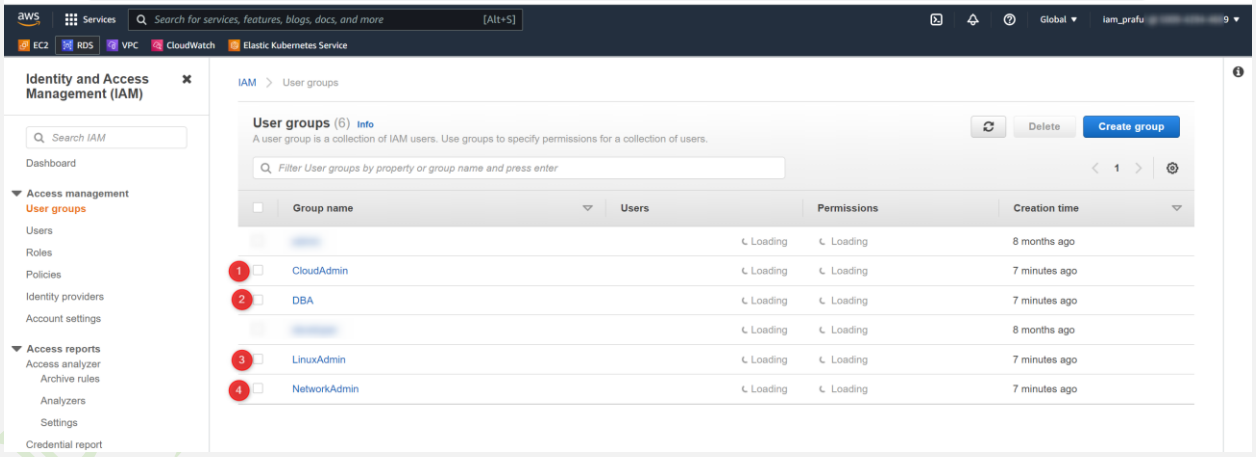
$ansible-playbook –I hosts playbook.yml



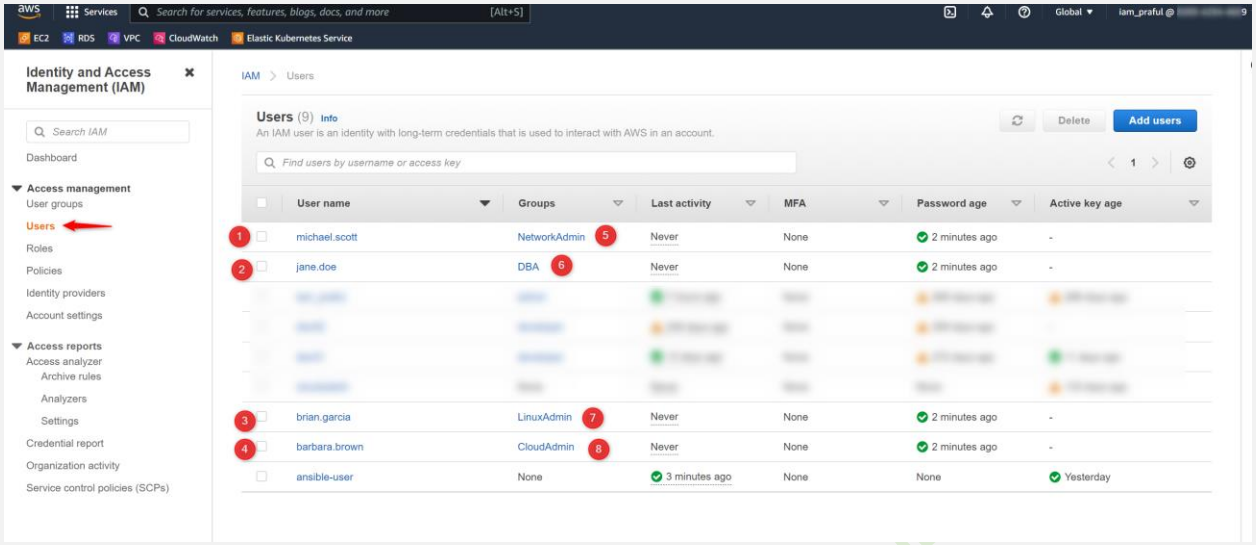


7.Verify that users and group are created in aws cloud

Groups: Groups successfully created automated way



Users: Users successfully created automated way



- Phase 4: Remove IAM group and users in AWS using Ansible

Solution 2: Create IAM group and users in AWS using Ansible

Source: GitHub: https://github.com/prafulpatel16/aws-ansible.git

Implementation steps:

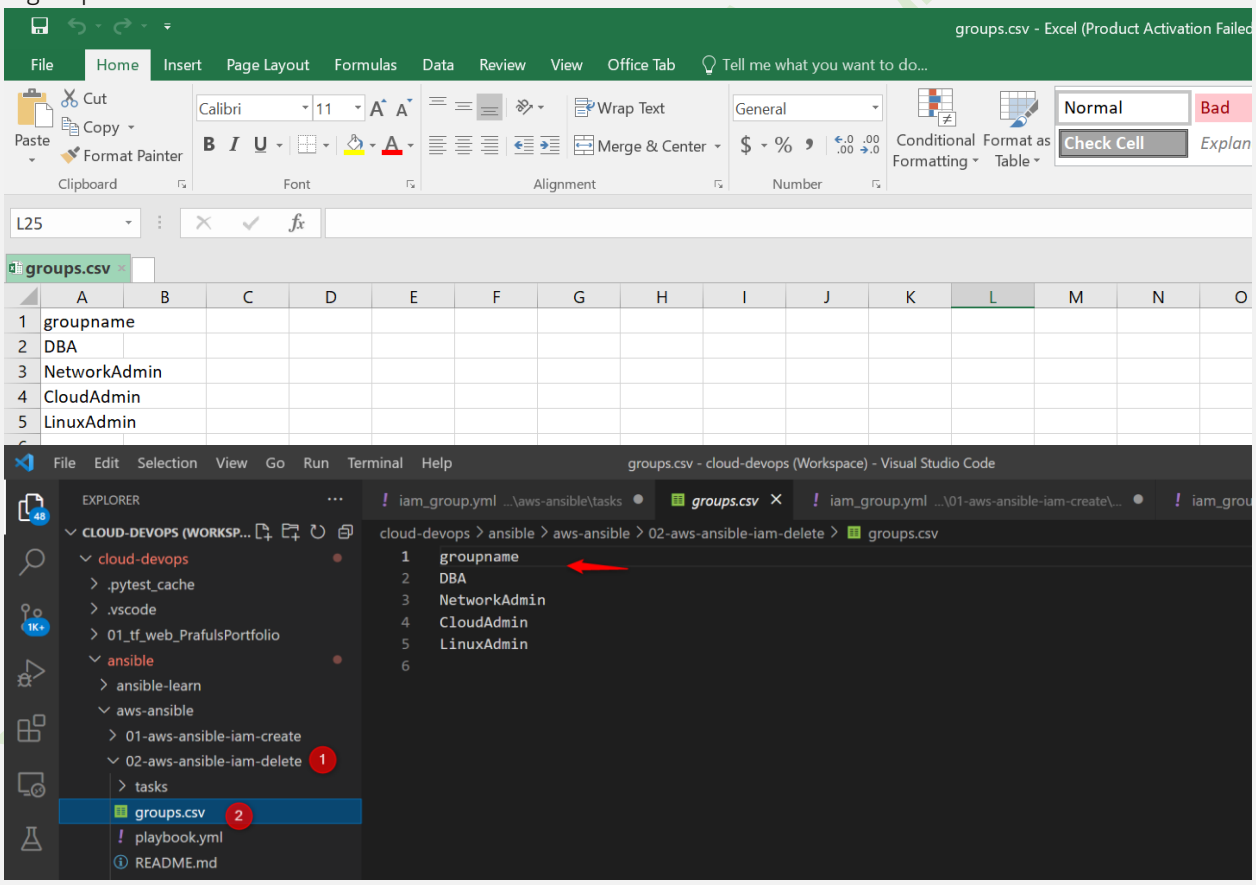
1.Create main working directory
2.Prepare groups.csv and users.csv data files
3.Create sub directory 'tasks'
4.Create two separate tasks file inside 'tasks' folder
    1. create iam_group.yml
    2. create iam_users.yml
5.Create main playbook file & include tasks folders file
    Playbook.yml
6.Run Playbook
7.Verify that users and group are created in aws cloud

1.Create a main working directory: 02-aws-ansible-iam-delete

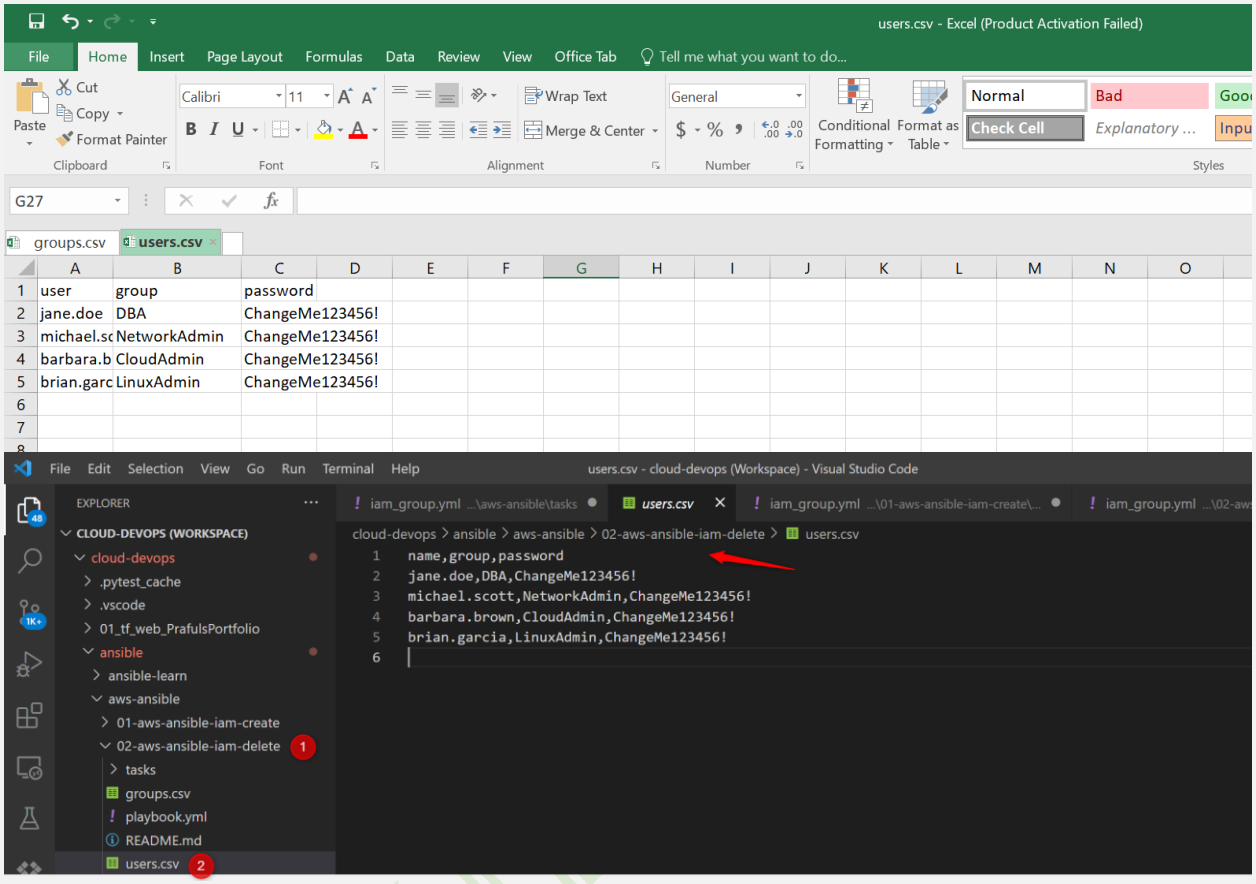
2.Prepare groups.csv and users.csv data files

Prepare CSV files:

1.groups.csv



2.users.csv



3.Create sub directory 'tasks'

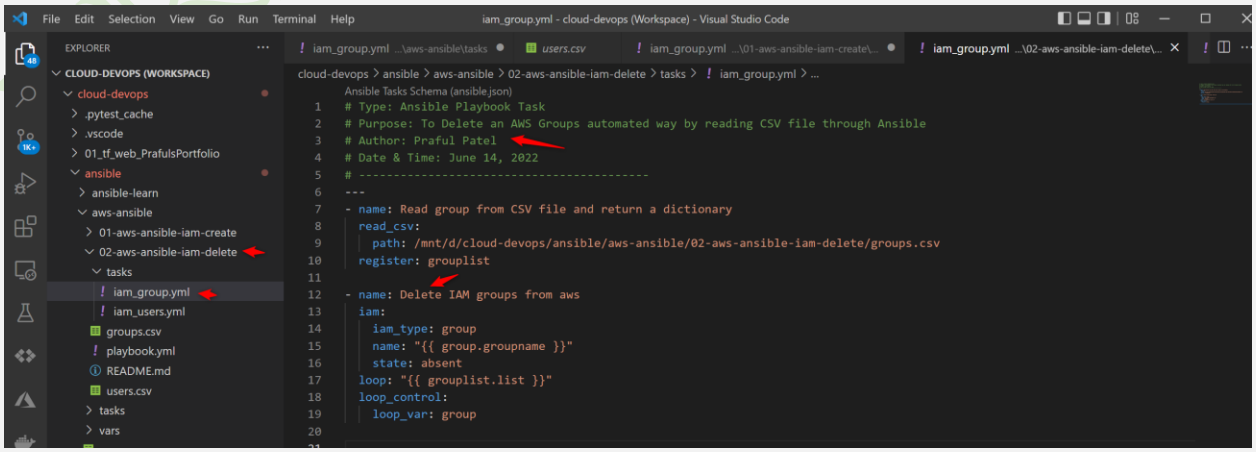
Folder: tasks ;

02-aws-ansible-iam-delete/tasks/

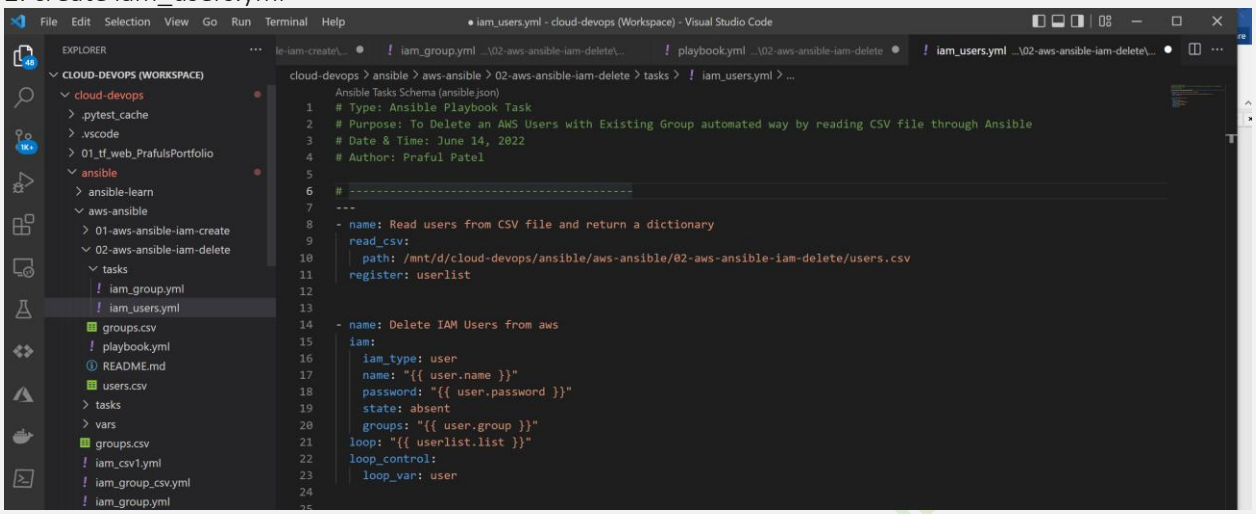
4.Create two separate tasks file inside 'tasks' folder

1. create iam_group,yml
2. create iam_users.yml

1.Iam_group.yml



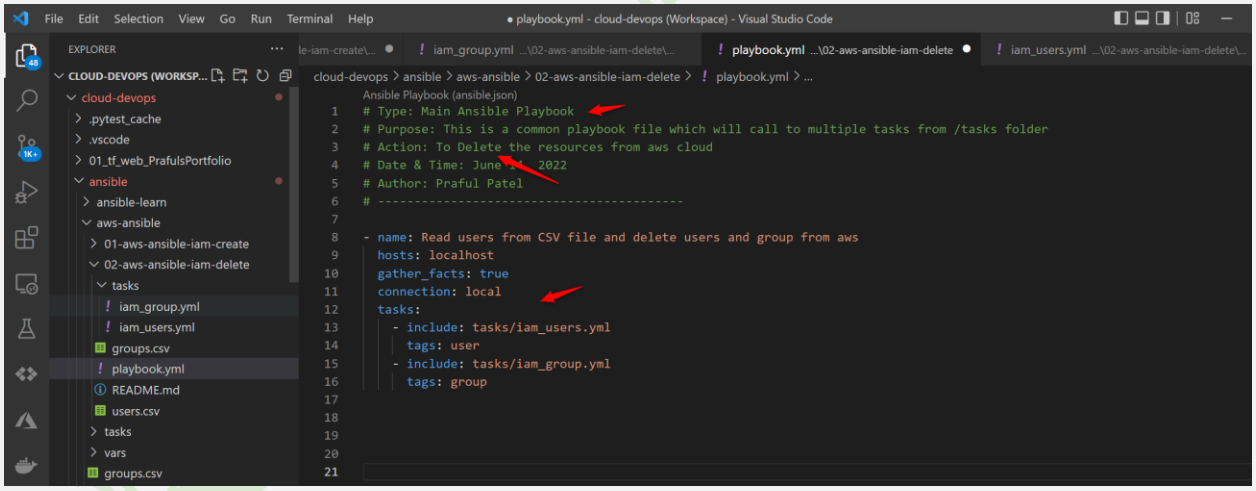
2. create iam_users.yml



5.Create main playbook file & include tasks folders file

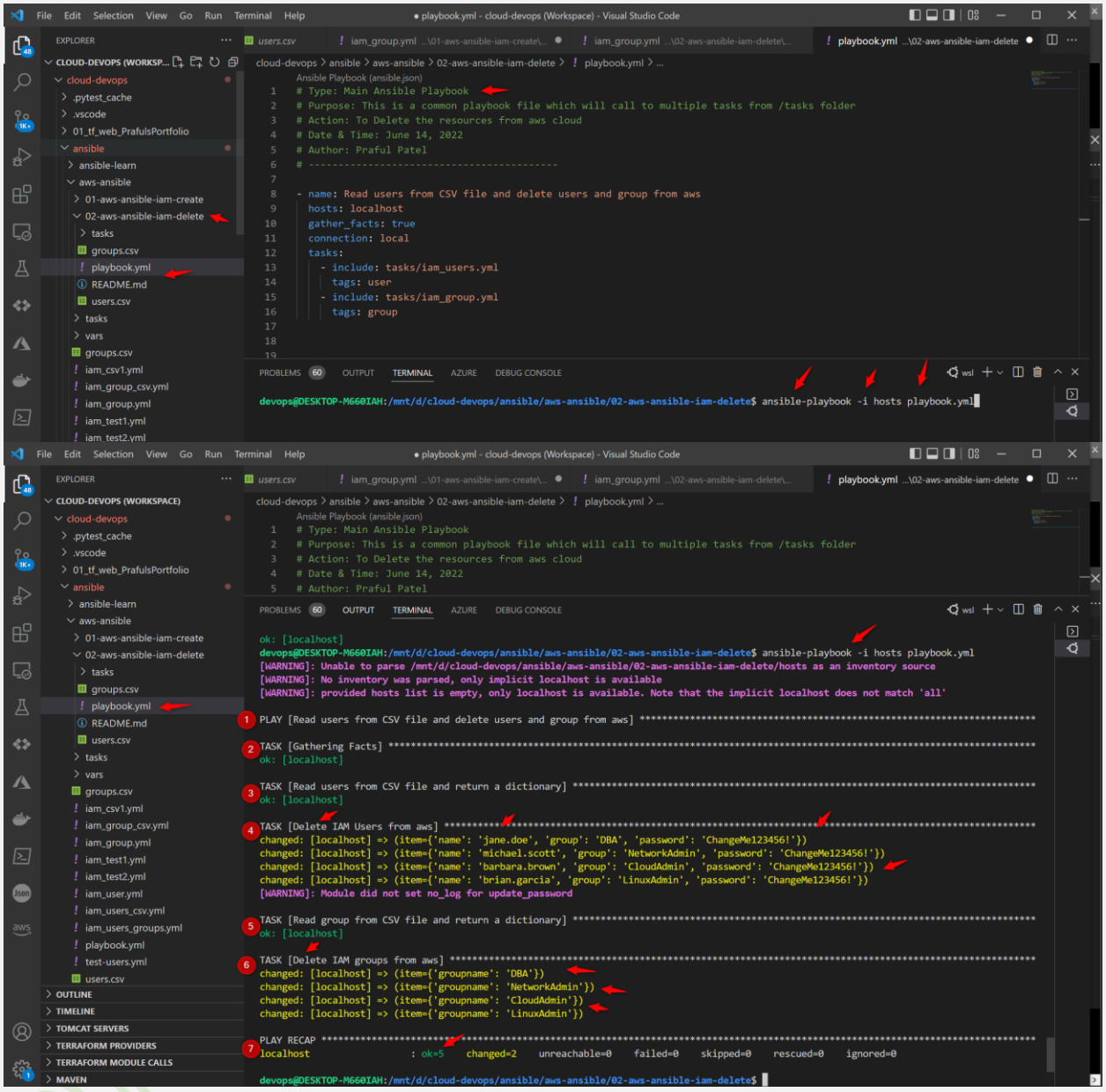
Playbook.yml

Main Playbook file: Playbook.yml

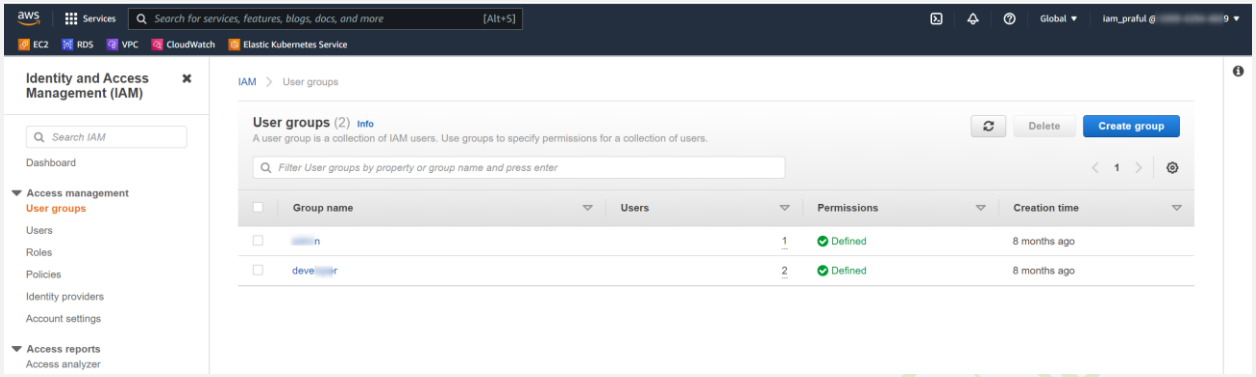


6.Run Playbook

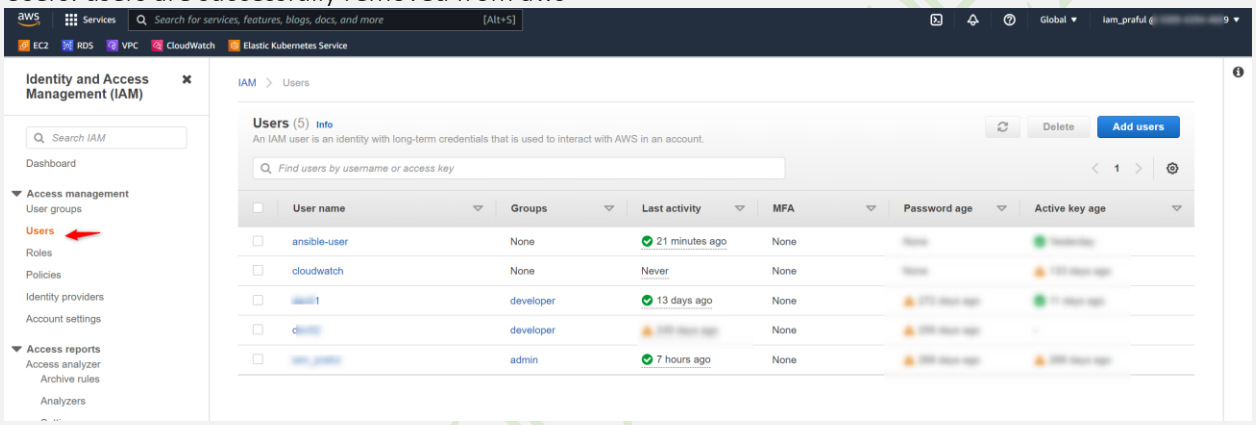
$ansible-playbook –I hosts playbook.yml



7.Verify that users and group are deleted from AWS cloud
Groups: Groups are successfully removed from aws



Users: users are successfully removed from aws



Congratulations!!!!