



<https://linkedin.com/in/prafulpatel16>

<https://github.com/prafulpatel16>

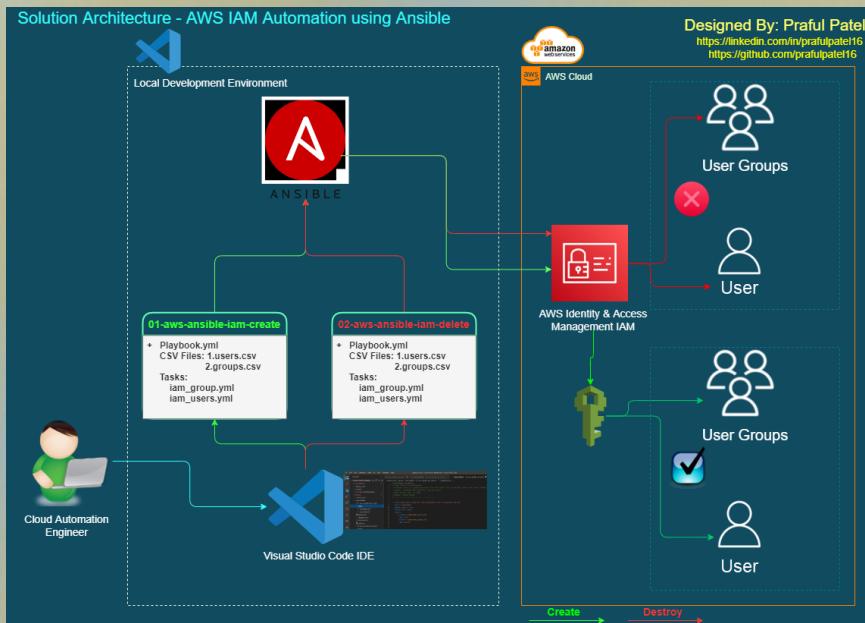


AWS - ANSIBLE AUTOMATION PROJECT

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.

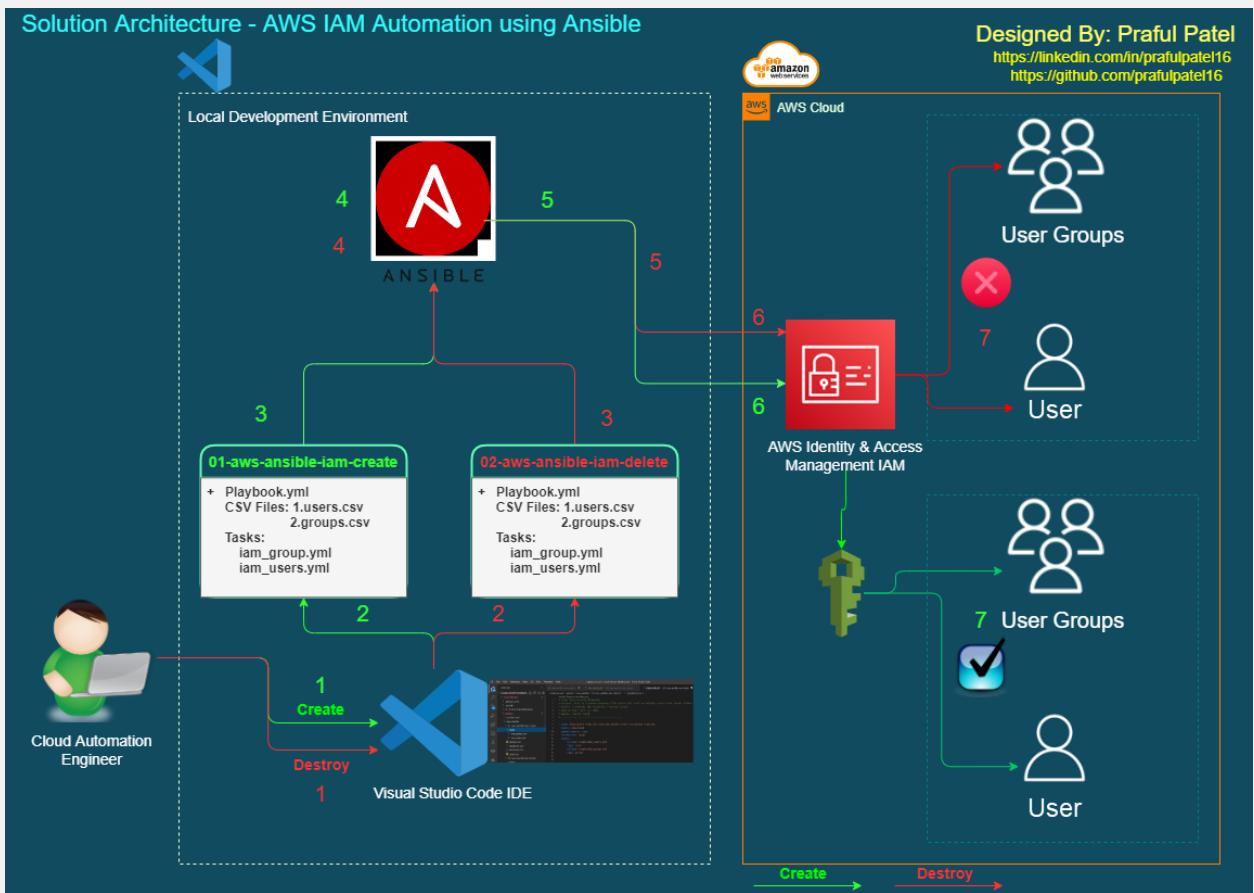
➤ **Project Cost Estimation:**

(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**

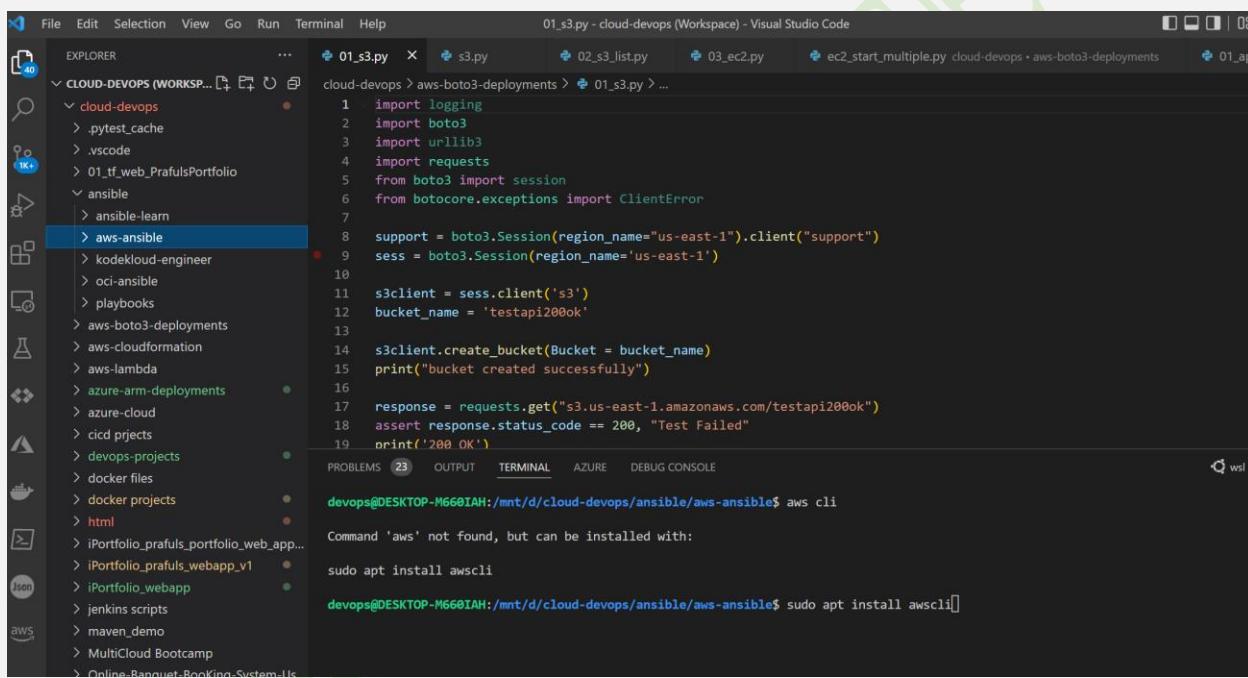
- ✓ Install AWS cli in VS code Ubuntu WSL

Sudo apt install awscli

Pip install boto

- ✓ Configure aws credentials in to vscode

Sudo apt install ansible

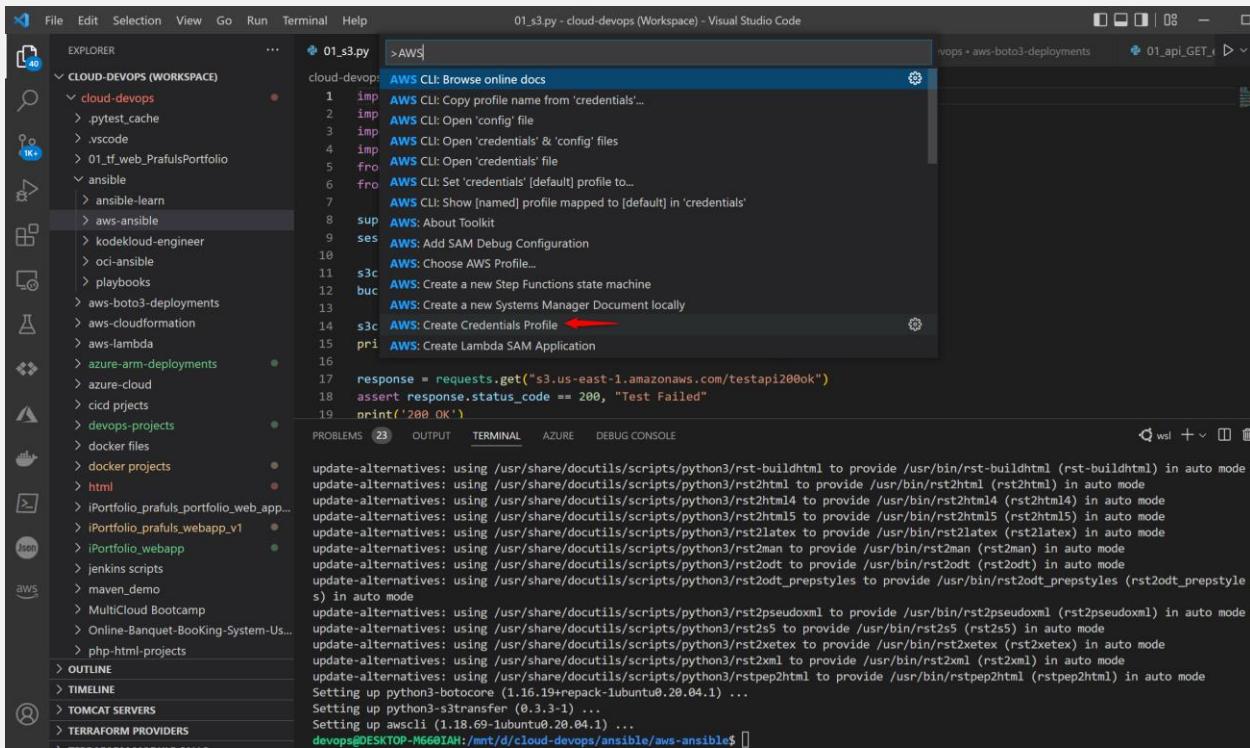


```

File Edit Selection View Go Run Terminal Help
01_s3.py - cloud-devops (Workspace) - Visual Studio Code
cloud-devops > aws-boto3-deployments > 01_s3.py > ...
1 import logging
2 import boto3
3 import urllib3
4 import requests
5 from boto3 import session
6 from botocore.exceptions import ClientError
7
8 support = boto3.Session(region_name='us-east-1').client("support")
9 sess = boto3.Session(region_name='us-east-1')
10
11 s3client = sess.client('s3')
12 bucket_name = 'testapi200ok'
13
14 s3client.create_bucket(Bucket = bucket_name)
15 print("bucket created successfully")
16
17 response = requests.get("s3.us-east-1.amazonaws.com/testapi200ok")
18 assert response.status_code == 200, "Test Failed"
19 print('200 OK')
PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ aws cli
Command 'aws' not found, but can be installed with:
  sudo apt install awscli
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ sudo apt install awscli

```

Configure Aws Profile



File Edit Selection View Go Run Terminal Help

01_s3.py - cloud-devops (Workspace) - Visual Studio Code

EXPLORER CLOUD-DEVS (WORKSPACE)

cloud-devops

cloud-devops

ansi... 01_s3.py > AWS

```

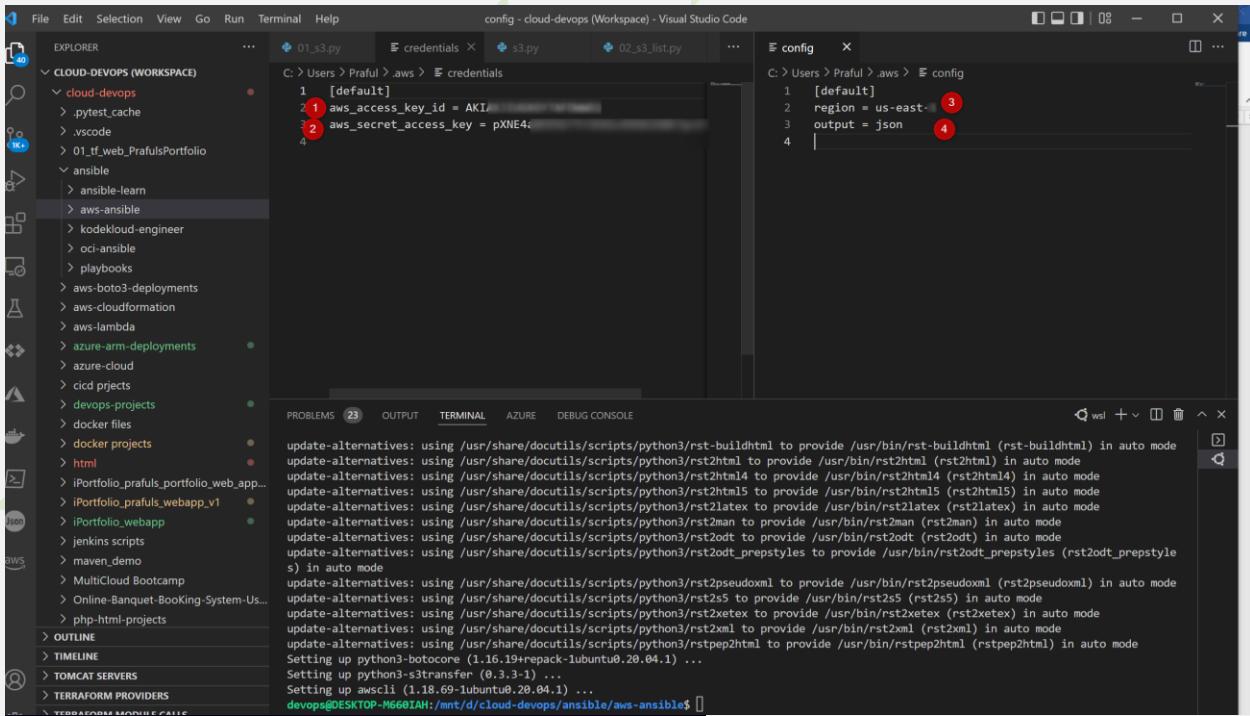
1 imp AWS CLI: Copy profile name from 'credentials'...
2 imp AWS CLI: Open 'config' file
3 imp AWS CLI: Open 'credentials' & 'config' files
4 imp AWS CLI: Open 'credentials' file
5 fro AWS CLI: Open 'credentials' file
6 fro AWS CLI: Set 'credentials' [default] profile to...
7 AWS CLI: Show [named] profile mapped to [default] in 'credentials'
8 sup AWS: About Toolkit
9 ses AWS: Add SAM Debug Configuration
10 s3c AWS: Choose AWS Profile...
11 buc AWS: Create a new Step Functions state machine
12 s3c AWS: Create a new Systems Manager Document locally
13 s3c AWS: Create Credentials Profile (arrow)
14 pri AWS: Create Lambda SAM Application
15
16
17 response = requests.get("s3.us-east-1.amazonaws.com/testapi200ok")
18 assert response.status_code == 200, "Test Failed"
19 print('200 OK')

```

PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE

wsl + □ □

update-alternatives: using /usr/share/docutils/scripts/python3/rst-buildhtml to provide /usr/bin/rst-buildhtml (rst-buildhtml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2html to provide /usr/bin/rst2html (rst2html) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2html4 to provide /usr/bin/rst2html4 (rst2html4) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2html5 to provide /usr/bin/rst2html5 (rst2html5) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2latex to provide /usr/bin/rst2latex (rst2latex) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2man to provide /usr/bin/rst2man (rst2man) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2odt to provide /usr/bin/rst2odt (rst2odt) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2odt_prepstyles to provide /usr/bin/rst2odt_prepstyles (rst2odt_prepstyle) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2pseudoxml to provide /usr/bin/rst2pseudoxml (rst2pseudoxml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2ss to provide /usr/bin/rst2ss (rst2ss) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xetex to provide /usr/bin/rst2xetex (rst2xetex) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xml to provide /usr/bin/rst2xml (rst2xml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rstpep2html to provide /usr/bin/rstpep2html (rstpep2html) in auto mode
Setting up python3-botocore (1.16.19+repack-lubuntu0.20.04.1) ...
Setting up python3-s3transfer (0.3.3-1) ...
Setting up awscli (1.18.69-lubuntu0.20.04.1) ...
Setting up awscli (1.18.69-lubuntu0.20.04.1) ...
devops@DESKTOP-M6G0IAH:/mnt/d/cloud-devops/ansible/aws-ansible\$



File Edit Selection View Go Run Terminal Help

config - cloud-devops (Workspace) - Visual Studio Code

EXPLORER CLOUD-DEVS (WORKSPACE)

cloud-devops

ansi... 01_s3.py > credentials > s3.py > 02_s3_list.py > config

C: > Users > Praful > .aws > config

```

1 [default]
2 aws_access_key_id = AKI/...
3
4 aws_secret_access_key = pXNE4;

```

C: > Users > Praful > .aws > config

```

1 [default]
2 region = us-east-1
3
4 output = json

```

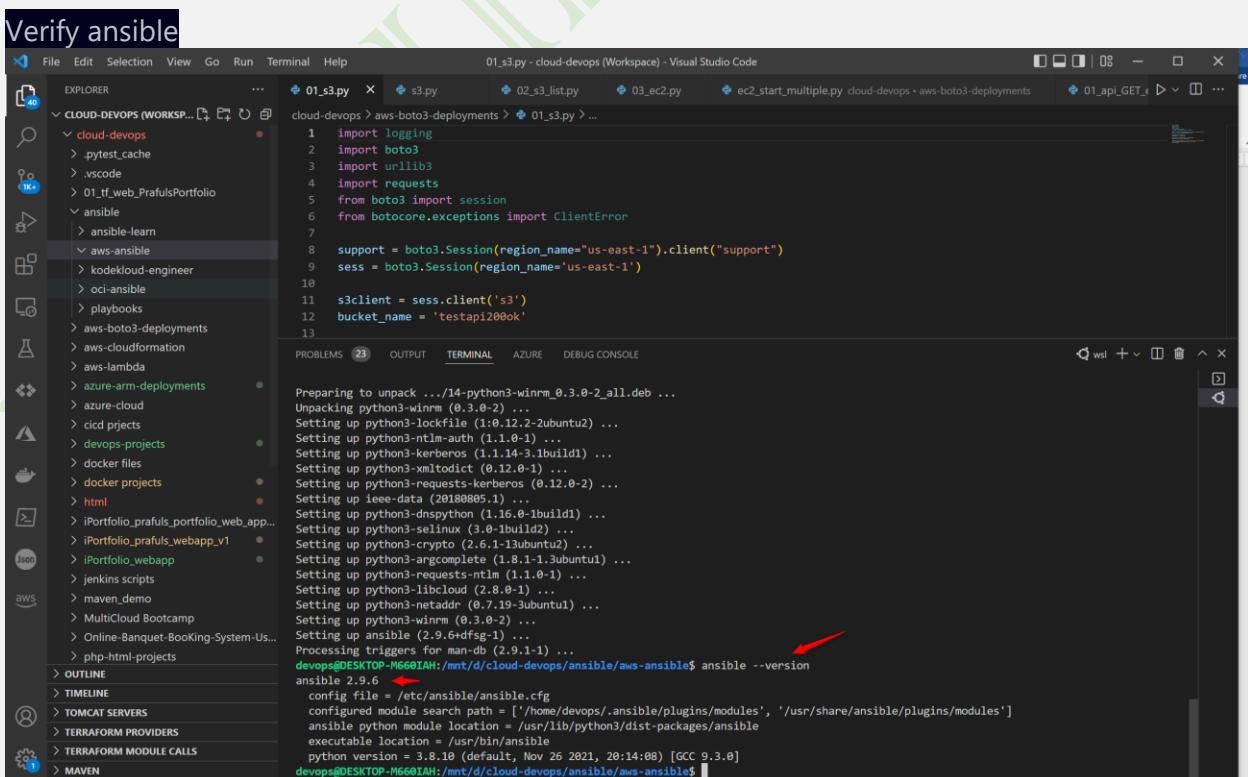
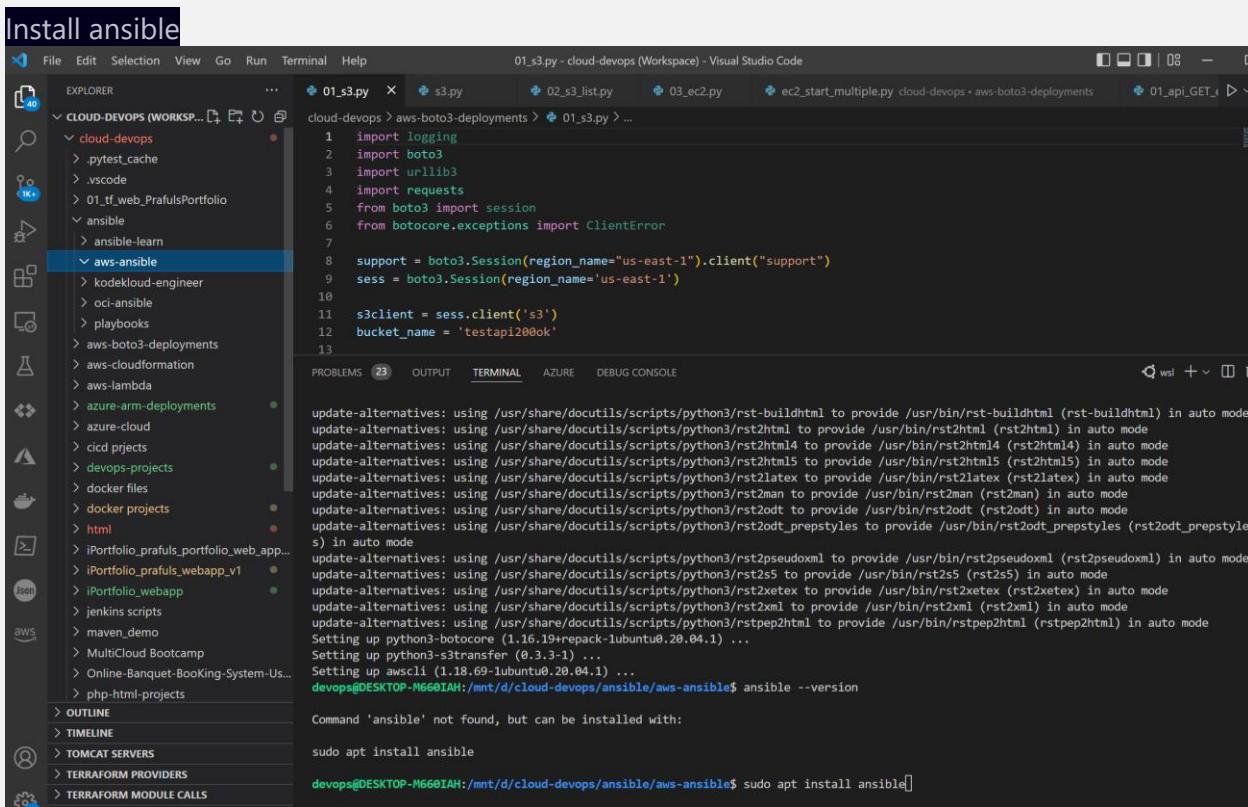
PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE

wsl + □ □

update-alternatives: using /usr/share/docutils/scripts/python3/rst-buildhtml to provide /usr/bin/rst-buildhtml (rst-buildhtml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2html to provide /usr/bin/rst2html (rst2html) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2html4 to provide /usr/bin/rst2html4 (rst2html4) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2html5 to provide /usr/bin/rst2html5 (rst2html5) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2latex to provide /usr/bin/rst2latex (rst2latex) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2man to provide /usr/bin/rst2man (rst2man) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2odt to provide /usr/bin/rst2odt (rst2odt) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2odt_prepstyles to provide /usr/bin/rst2odt_prepstyles (rst2odt_prepstyle) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2pseudoxml to provide /usr/bin/rst2pseudoxml (rst2pseudoxml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2ss to provide /usr/bin/rst2ss (rst2ss) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xetex to provide /usr/bin/rst2xetex (rst2xetex) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xml to provide /usr/bin/rst2xml (rst2xml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rstpep2html to provide /usr/bin/rstpep2html (rstpep2html) in auto mode
Setting up python3-botocore (1.16.19+repack-lubuntu0.20.04.1) ...
Setting up python3-s3transfer (0.3.3-1) ...
Setting up awscli (1.18.69-lubuntu0.20.04.1) ...
Setting up awscli (1.18.69-lubuntu0.20.04.1) ...
devops@DESKTOP-M6G0IAH:/mnt/d/cloud-devops/ansible/aws-ansible\$

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

Phase 2: AWS cli and Ansible configuration



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

Create Ansible IAM user

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* [Add another user](#)

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type* Access key - Programmatic access
Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access
Enables a password that allows users to sign-in to the AWS Management Console.

Attach permission

Add user

Set permissions

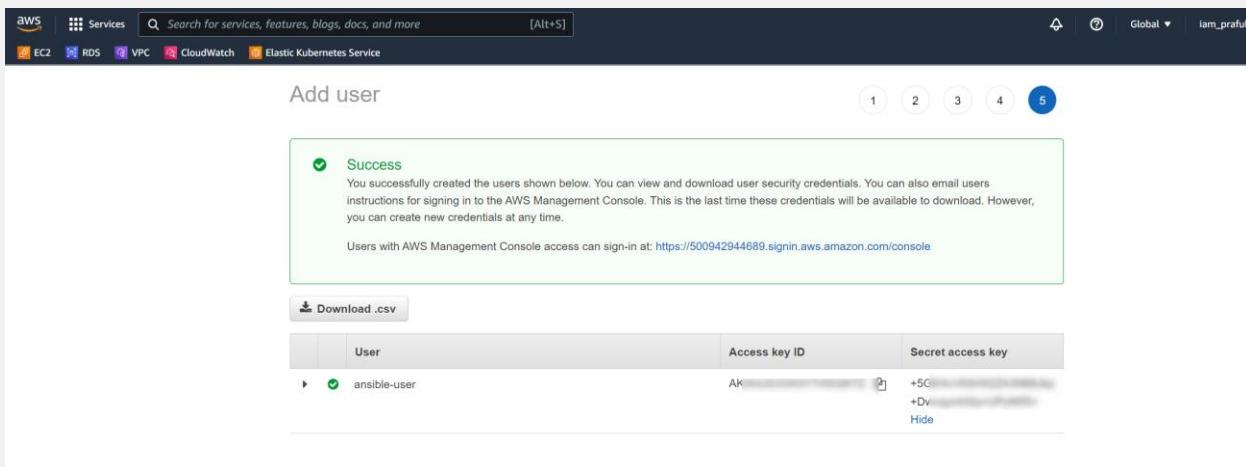
[Add user to group](#) [Copy permissions from existing user](#) [Attach existing policies directly](#)

[Create policy](#)

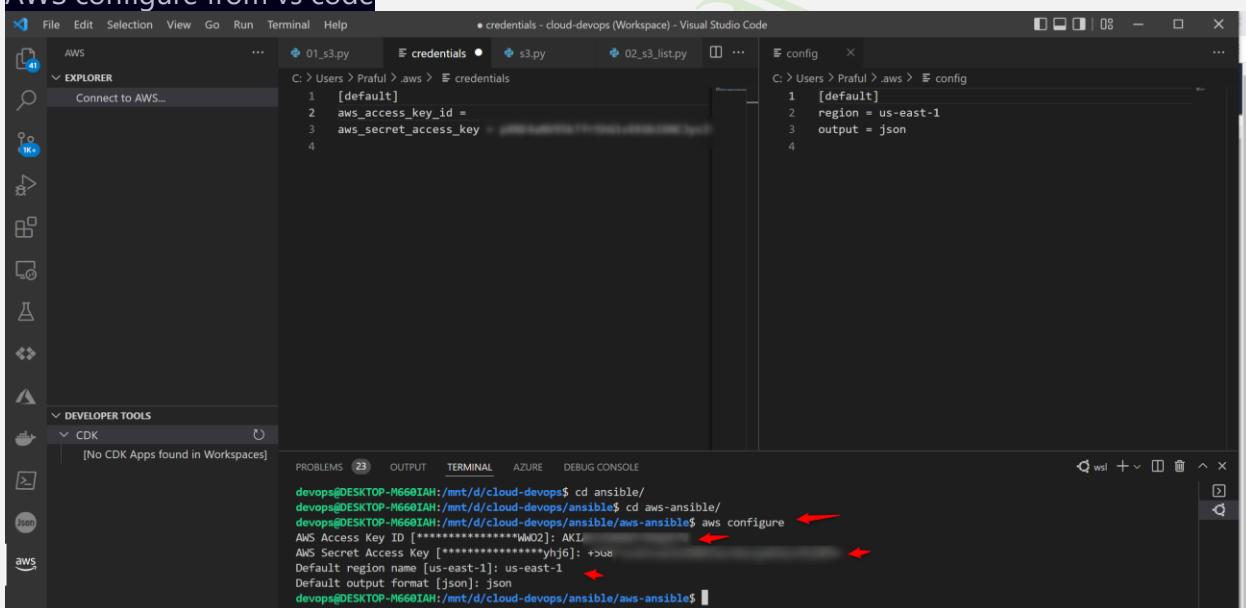
Filter policies Showing 330 results

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (9)
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	Permissions policy (1)
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	AWS managed	Permissions policy (1)
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	None
<input type="checkbox"/>	AmazonAppFlowFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonAppFlowReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AmazonAppStreamFullAccess	AWS managed	None

Copy secret key and access key



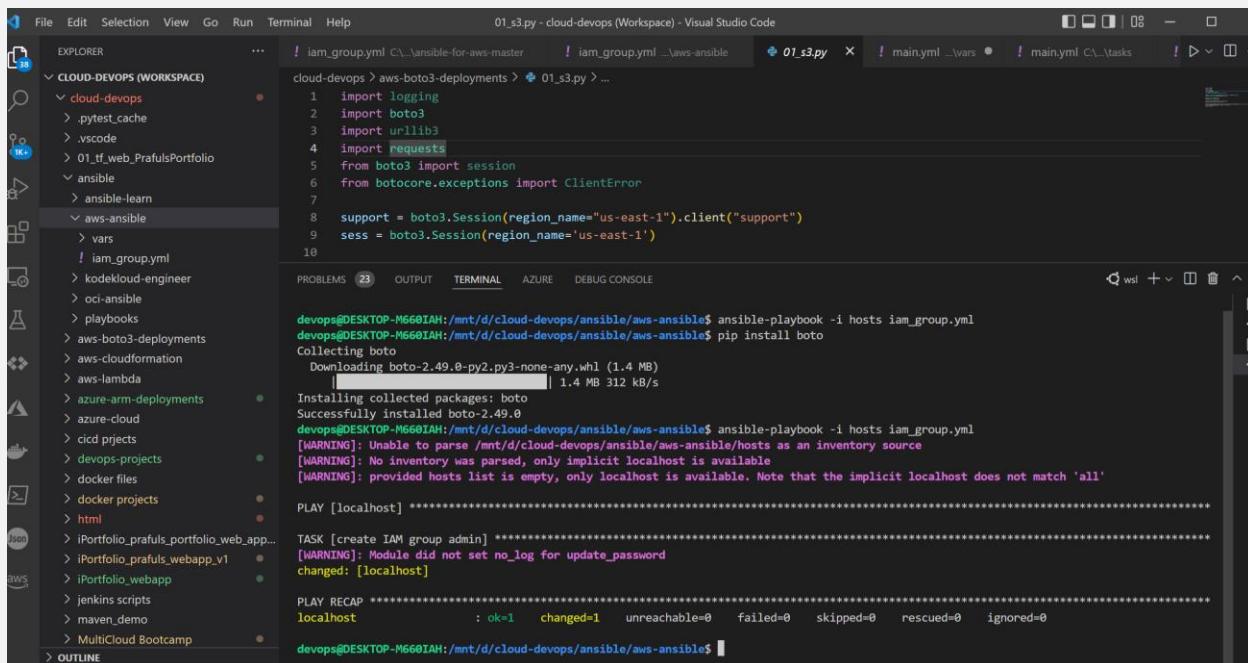
AWS configure from vs code



<https://github.com/xchangebit/ansible-aws-cli>

Install

Pip install boto



```

File Edit Selection View Go Run Terminal Help
01_s3.py - cloud-devops (Workspace) - Visual Studio Code
iam_group.yml C:\...\ansible-for-aws-master iam_group.yml ...aws-ansible 01_s3.py main.yml C:\...\tasks
cloud-devops > aws-boto3-deployments > 01_s3.py > ...
1 import logging
2 import boto3
3 import urllib3
4 import requests
5 from botocore import session
6 from botocore.exceptions import ClientError
7
8 support = boto3.Session(region_name='us-east-1').client("support")
9 sess = boto3.Session(region_name='us-east-1')
10

PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE

devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ ansible-playbook -i hosts iam_group.yml
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ pip install boto
Collecting boto
  Downloading boto-2.49.0-py2.py3-none-any.whl (1.4 MB)
  100% |████████████████████████████████| 1.4 MB 312 kB/s

Installing collected packages: boto
Successfully installed boto-2.49.0
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ ansible-playbook -i hosts iam_group.yml
[WARNING]: Unable to parse /mnt/d/cloud-devops/ansible/aws-ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [localhost] ****
TASK [create IAM group admin] ****
[WARNING]: Module did not set no_log for update_password
changed: [localhost]

PLAY RECAP ****
localhost : ok=1 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

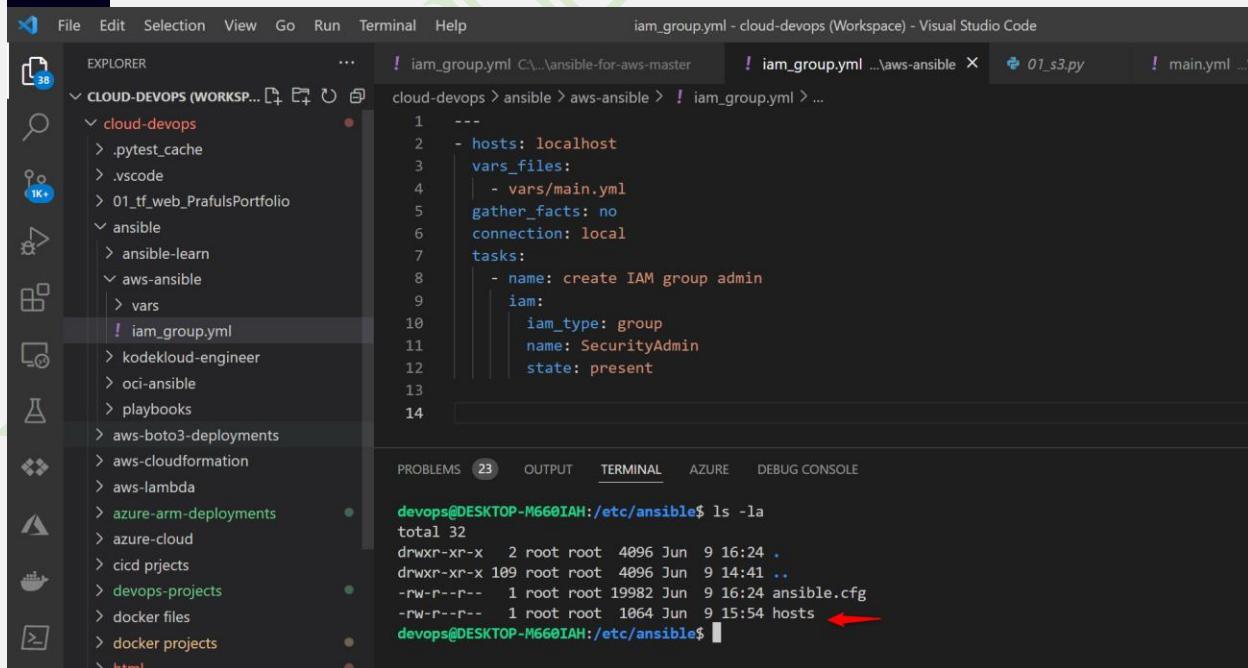
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ 

```

Configure Inventory

Cd /etc/ansible

Vi hosts



```

File Edit Selection View Go Run Terminal Help
iam_group.yml - cloud-devops (Workspace) - Visual Studio Code
iam_group.yml C:\...\ansible-for-aws-master iam_group.yml ...aws-ansible 01_s3.py main.yml ...
cloud-devops > ansible > aws-ansible > iam_group.yml > ...
1 ---
2   - hosts: localhost
3     vars_files:
4       - vars/main.yml
5     gather_facts: no
6     connection: local
7     tasks:
8       - name: create IAM group admin
9         iam:
10           iam_type: group
11           name: SecurityAdmin
12           state: present

PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE

devops@DESKTOP-M660IAH:/etc/ansible$ ls -la
total 32
drwxr-xr-x  2 root root  4096 Jun  9 16:24 .
drwxr-xr-x 109 root root  4096 Jun  9 14:41 ..
-rw-r--r--  1 root root 19982 Jun  9 16:24 ansible.cfg
-rw-r--r--  1 root root 1064 Jun  9 15:54 hosts
devops@DESKTOP-M660IAH:/etc/ansible$ 

```

Add this line to localhost

[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}
```

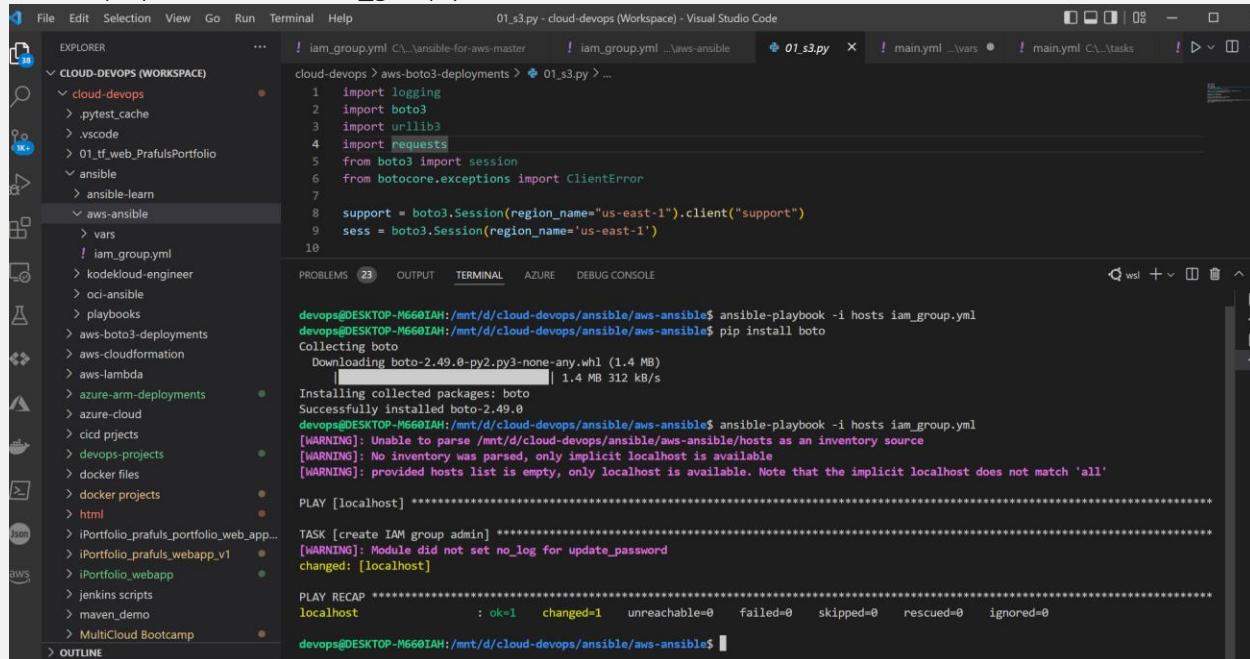
```
export AWS_SECRET_ACCESS_KEY={secret access key}
```

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

Group Creation: SecurityAdmin

Run ansible playbook

Ansible-playbook -i hosts iam_group.yml



```

File Edit Selection View Go Run Terminal Help
EXPLORER CLOUD-DEVOPS (WORKSPACE)
  cloud-devops
    .pytest_cache
    .vscode
  01_tf_web_PrafulsPortfolio
  ansible
    ansible-learn
    aws-ansible
      vars
      iam_group.yml
      kodakloud-engineer
      oci-ansible
      playbooks
      aws-boto3-deployments
      aws-cloudformation
      aws-lambda
      azure-arm-deployments
      azure-cloud
      cicd projects
      devops-projects
      docker files
      docker projects
      html
      iPortfolio_prafuls_portfolio_web_app...
      iPortfolio_prafuls_webapp_v1
      iPortfolio_webapp
      jenkins scripts
      maven_demo
      MultiCloud Bootcamp
  OUTLINE
  01_s3.py - cloud-devops (Workspace) - Visual Studio Code
  iam_group.yml C:\_ansible-for-aws-master\cloud-devops> 01_s3.py > ...
  1 import logging
  2 import boto3
  3 import urllib3
  4 import requests
  5 from botocore import session
  6 from botocore.exceptions import ClientError
  7
  8 support = boto3.Session(region_name="us-east-1").client("support")
  9 sess = boto3.Session(region_name='us-east-1')
  10

PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE

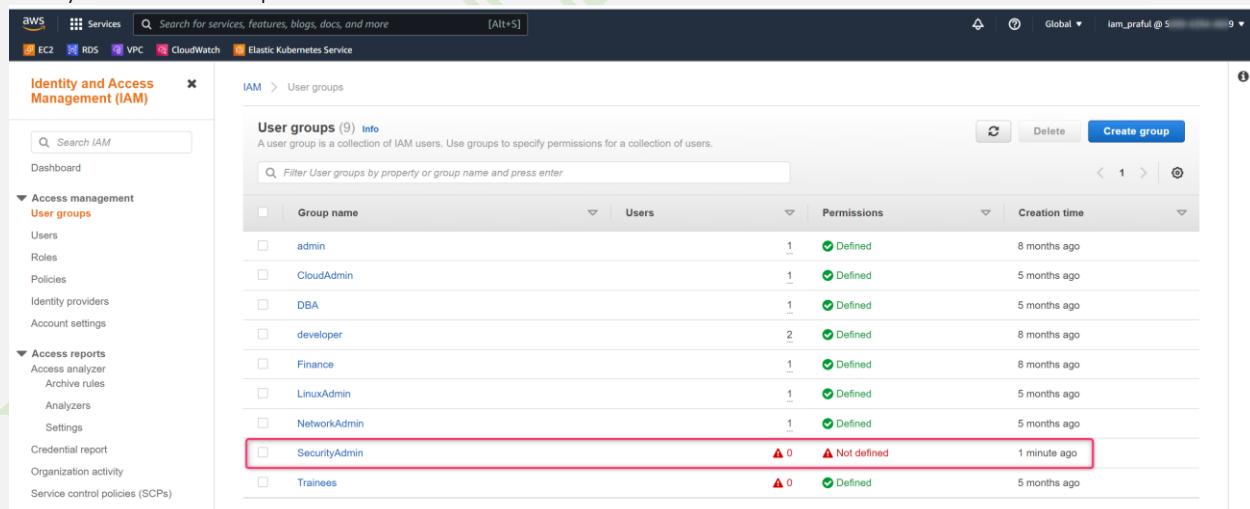
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ ansible-playbook -i hosts iam_group.yml
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ pip install boto
Collecting boto
  Downloading boto-2.49.0-py3-none-any.whl (1.4 MB)
  100% |████████████████████████████████| 1.4 MB 312 kB/s
  Installing collected packages: boto
  Successfully installed boto-2.49.0
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ ansible-playbook -i hosts iam_group.yml
[WARNING]: Unable to parse /mnt/d/cloud-devops/ansible/aws-ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
PLAY [localhost] ****
TASK [create IAM group admin] ****
[WARNING]: Module did not set no_log for update_password
changed: [localhost]

PLAY RECAP ****
localhost : ok=1 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ 

```

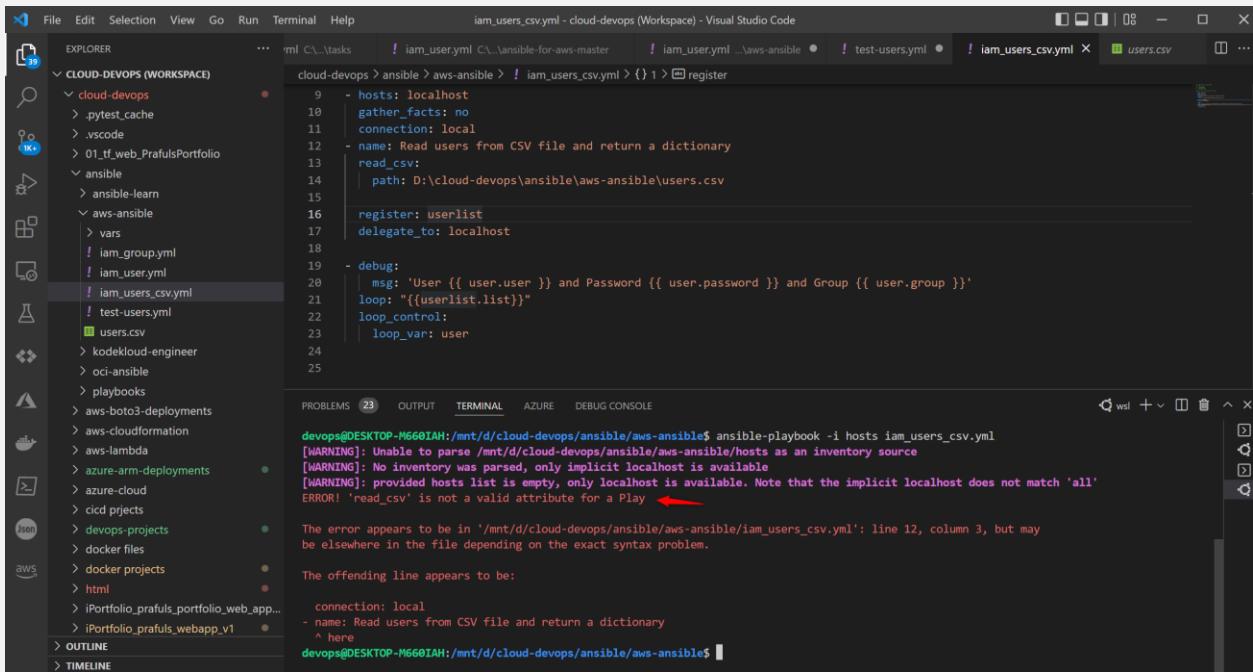
Playbook execution successful

Verify that User Group has been created in AWS



Group name	Users	Permissions	Creation time
admin	1	Defined	8 months ago
CloudAdmin	1	Defined	5 months ago
DBA	1	Defined	5 months ago
developer	2	Defined	8 months ago
Finance	1	Defined	8 months ago
LinuxAdmin	1	Defined	5 months ago
NetworkAdmin	1	Defined	5 months ago
SecurityAdmin	0	Not defined	1 minute ago
Trainees	0	Defined	5 months ago

Error



```

cloud-devops > ansible > aws-ansible > iam_users_csv.yml > {} 1 > register
  9 - hosts: localhost
 10   gather_facts: no
 11   connection: local
 12   name: Read users from CSV file and return a dictionary
 13   read_csv:
 14     path: D:\cloud-devops\ansible\aws-ansible\users.csv
 15
 16   register: userlist
 17   delegate_to: localhost
 18
 19   - debug:
 20     msg: "User {{ user.user }} and Password {{ user.password }} and Group {{ user.group }}"
 21     loop: "{{userlist.list}}"
 22     loop_control:
 23       loop_var: user
 24
 25

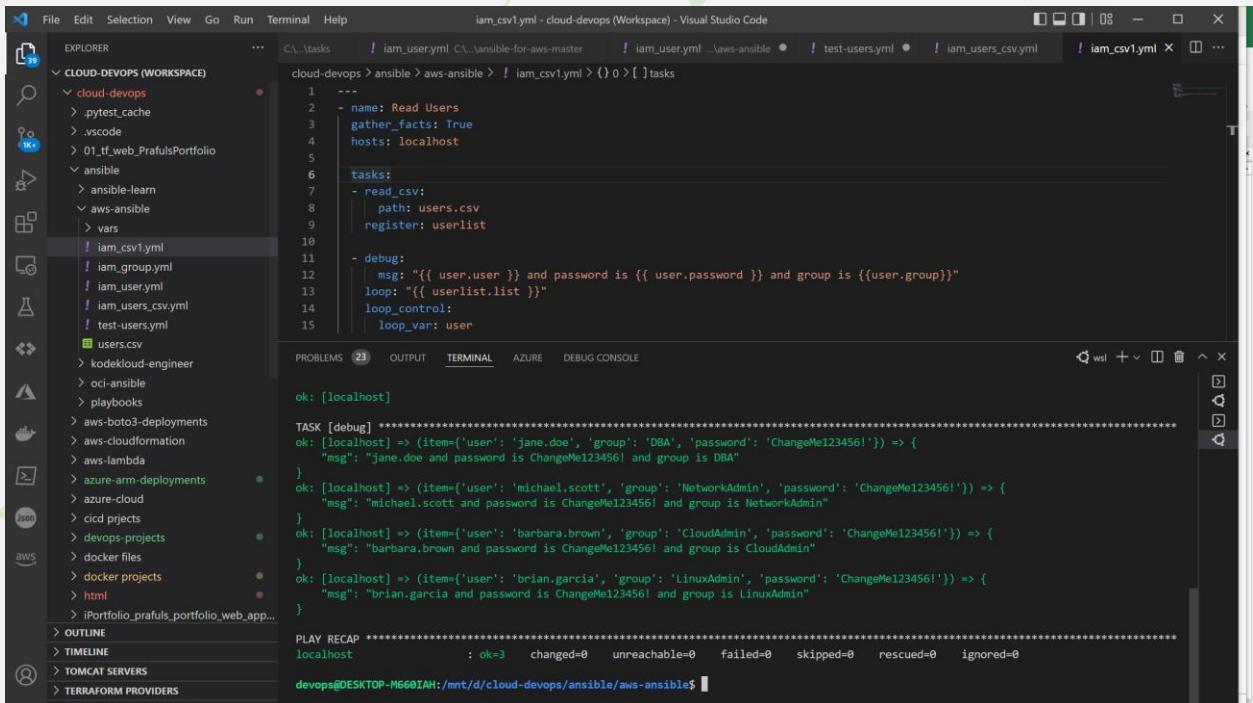
PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE

devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ ansible-playbook -i hosts iam_users_csv.yml
[WARNING]: Unable to parse /mnt/d/cloud-devops/ansible/aws-ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
ERROR! 'read_csv' is not a valid attribute for a Play ←

The error appears to be in '/mnt/d/cloud-devops/ansible/aws-ansible/iam_users_csv.yml': line 12, column 3, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:
connection: local
- name: Read users from CSV file and return a dictionary
  ^ here
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$
```

Test 2



```

cloud-devops > ansible > aws-ansible > iam_csv1.yml > {} 0 > [ ] tasks
  1 ---
  2   - name: Read Users
  3     gather_facts: True
  4     hosts: localhost
  5
  6     tasks:
  7       - read_csv:
  8         path: users.csv
  9         register: userlist
 10
 11     - debug:
 12       msg: "{{ user.user }} and password is {{ user.password }} and group is {{user.group}}"
 13     loop: "{{userlist.list}}"
 14     loop_control:
 15       loop_var: user

PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE

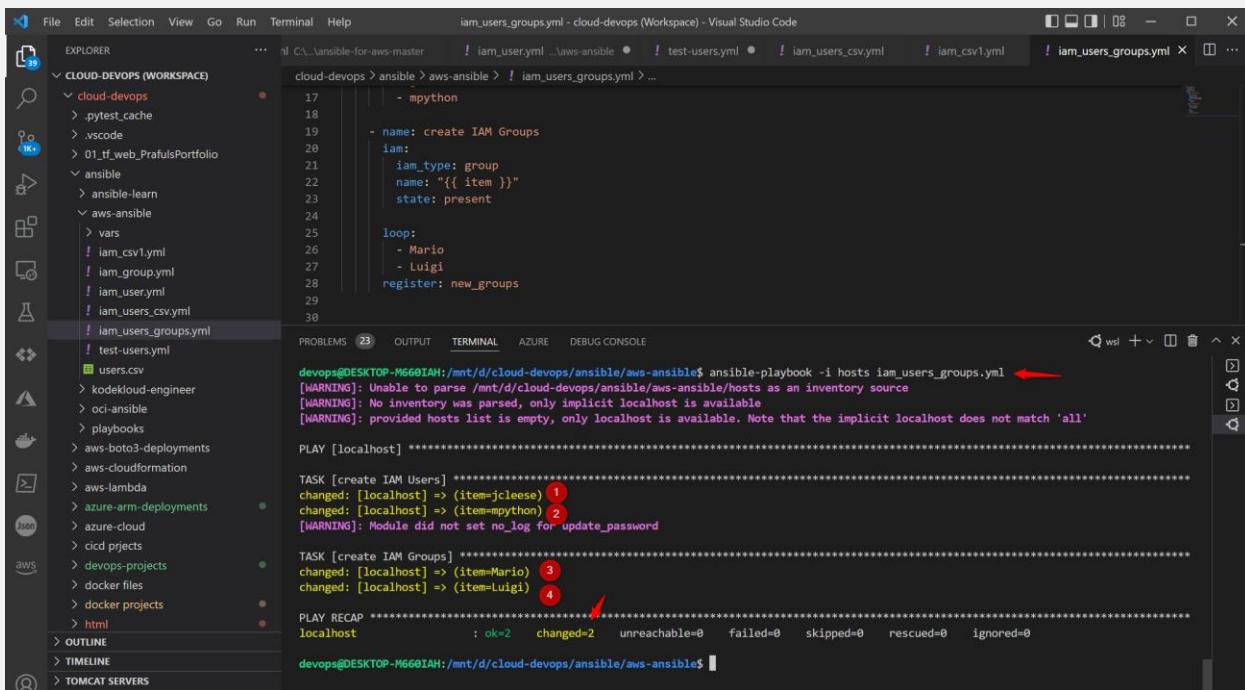
ok: [localhost]

TASK [debug] ****
ok: [localhost] => (item={'user': 'jane.doe', 'group': 'DBA', 'password': 'ChangeMe123456!'}) => {
    "msg": "jane.doe and password is ChangeMe123456! and group is DBA"
}
ok: [localhost] => (item={'user': 'michael.scott', 'group': 'NetworkAdmin', 'password': 'ChangeMe123456!'}) => {
    "msg": "michael.scott and password is ChangeMe123456! and group is NetworkAdmin"
}
ok: [localhost] => (item={'user': 'barbara.brown', 'group': 'CloudAdmin', 'password': 'ChangeMe123456!'}) => {
    "msg": "barbara.brown and password is ChangeMe123456! and group is CloudAdmin"
}
ok: [localhost] => (item={'user': 'brian.garcia', 'group': 'LinuxAdmin', 'password': 'ChangeMe123456!'}) => {
    "msg": "brian.garcia and password is ChangeMe123456! and group is LinuxAdmin"
}

PLAY RECAP ****
localhost          : ok=3    changed=0    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0

devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$
```

Test 3 Create Users and Groups

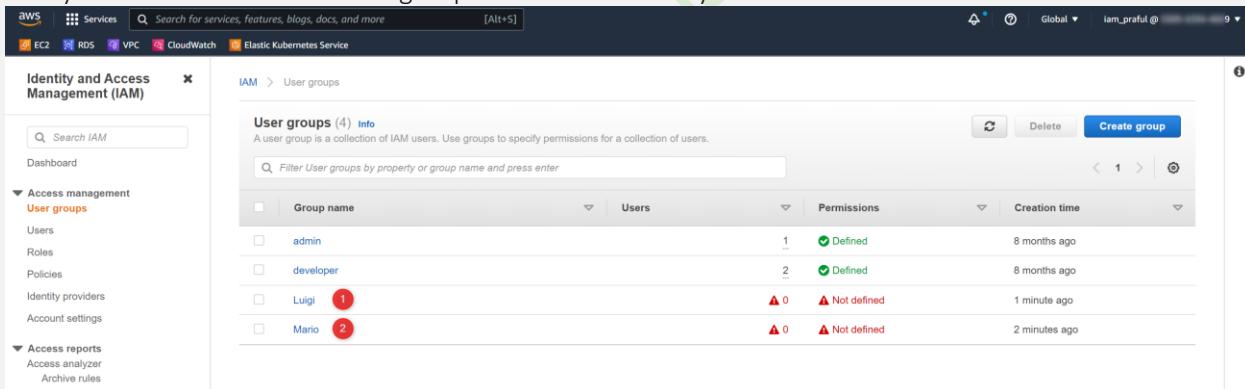


```

File Edit Selection View Go Run Terminal Help
iam_users_groups.yml - cloud-devops (Workspace) - Visual Studio Code
cloud-devops > ansible > aws-ansible > iam_users_groups.yml > ...
17     - mpython
18
19     - name: create IAM Groups
20     iam:
21         iam_type: group
22         name: "{{ item }}"
23         state: present
24
25     loop:
26         - Mario
27         - Luigi
28     register: new_groups
29
30
PROBLEMS 23 OUTPUT TERMINAL AZURE DEBUG CONSOLE
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ ansible-playbook -i hosts iam_users_groups.yml
[WARNING]: Unable to parse /mnt/d/cloud-devops/ansible/aws-ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
PLAY [localhost] ****
TASK [create IAM Users] ****
changed: [localhost] => (item=jcleese) 1
changed: [localhost] => (item=mpython) 2
[WARNING]: Module did not set no_log for update_password
TASK [create IAM Groups] ****
changed: [localhost] => (item=Mario) 3
changed: [localhost] => (item=Luigi) 4
PLAY RECAP ****
localhost : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
devops@DESKTOP-M660IAH:/mnt/d/cloud-devops/ansible/aws-ansible$ 

```

Verify in AWS console if user and group created successfully



Group name	Users	Permissions	Creation time
admin	1	Defined	8 months ago
developer	2	Defined	8 months ago
Luigi	0	Not defined	1 minute ago
Mario	0	Not defined	2 minutes ago

The screenshot shows the AWS IAM Users page. The left sidebar is titled 'Identity and Access Management (IAM)' and includes sections for 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings), 'Access reports' (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, Service control policies (SCPs)). The main right panel is titled 'Users (8) Info' and contains a table with the following data:

User	Access key ID	Access key status	Last used
jleese	1	None	6 minutes ago
mpython	2	None	5 minutes ago
terraform		None	None

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

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	groupname														
2	DBA														
3	NetworkAdmin														
4	CloudAdmin														
5	LinuxAdmin														

File Edit Selection View Go Run Terminal Help groups.csv - cloud-devops (Workspace) - Visual Studio Code

EXPLORER

- CLOUD-DEVOPS (WORKSP... ...
- cloud-devops
 - .pytest_cache
 - .vscode
 - 01_tf_web_PrafulsPortfolio
 - ansible
 - ansible-learn
 - aws-ansible
 - 01-aws-ansible-iam-create
 - tasks
 - iam_group.yml
 - iam_users.yml
 - groups.csv ←
 - playbook.yml
 - README.md
 - users.csv
- 02-aws-ansible-iam-delete

groups.csv

```

1 groupname
2 DBA
3 NetworkAdmin
4 CloudAdmin
5 LinuxAdmin
6

```

2.users.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	user	group	password												
2	jane.doe	DBA	ChangeMe123456!												
3	michael.sc	NetworkAdmin	ChangeMe123456!												
4	barbara.b	CloudAdmin	ChangeMe123456!												
5	brian.garc	LinuxAdmin	ChangeMe123456!												
6															
7															
8															

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

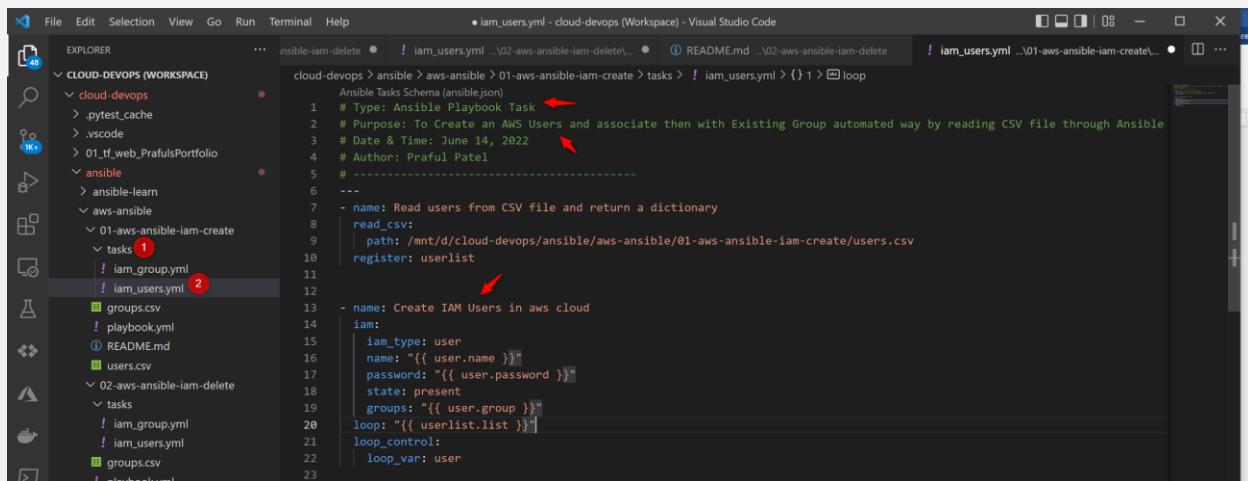
1. iam_group.yml

```

# Ansible Tasks Schema (ansible.json)
# Purpose: To Create an AWS Groups automated way by reading CSV file through Ansible
# Date & Time: June 14, 2022
# Author: Praful Patel
#
# -----
# name: Read group from CSV file and return a dictionary
read_csv:
  path: /mnt/d/cloud-devops/ansible/aws-ansible/01-aws-ansible-iam-create/groups.csv
  register: grouplist
#
# name: Create IAM groups in aws cloud
iam:
  iam_type: group
  name: "{{ group.groupname }}"
  state: present
  loop: "{{ grouplist.list }}"
  loop_control:
    loop_var: group

```

2. iam_users.yml



```

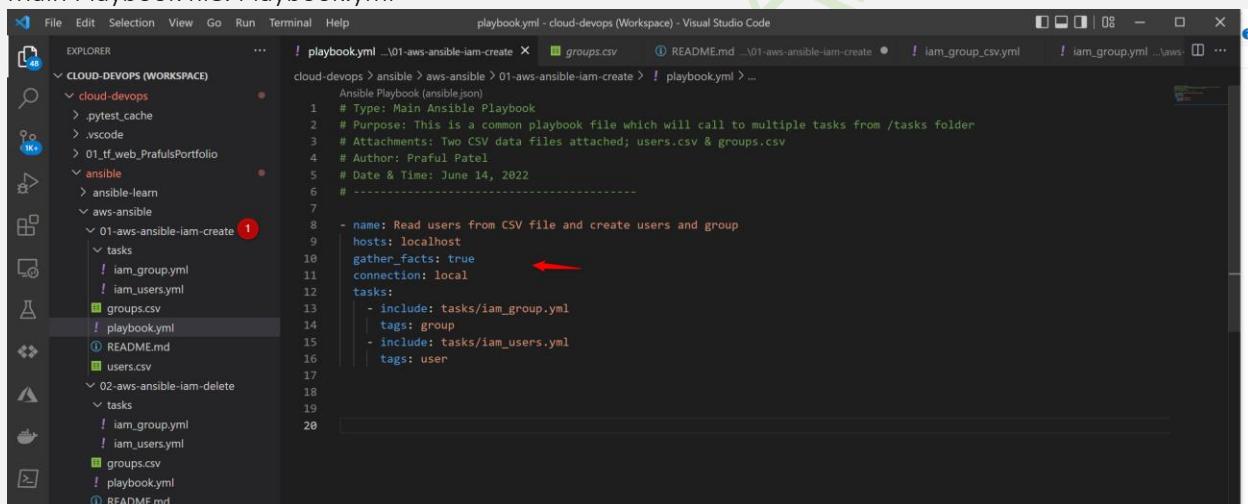
cloud-devops > ansible > aws-ansible > 01-aws-ansible-iam-create > tasks > ! iam_users.yml > {} 1 > loop
  Ansible Tasks Schema (ansible.json)
  1 # Type: Ansible Playbook Task
  2 # Purpose: To Create an AWS Users and associate then with Existing Group automated way by reading CSV file through Ansible
  3 # Date & Time: June 14, 2022
  4 # Author: Praful Patel
  5 # -----
  6
  7 - name: Read users from CSV file and return a dictionary
  8   read_csv:
  9     path: /mnt/d/cloud-devops/ansible/aws-ansible/01-aws-ansible-iam-create/users.csv
 10   register: userlist
 11
 12 - name: Create IAM Users in aws cloud
 13   iam:
 14     iam_type: user
 15     name: "{{ user.name }}"
 16     password: "{{ user.password }}"
 17     state: present
 18     groups: "{{ user.group }}"
 19   loop: "{{ userlist.list }}"
 20   loop_control:
 21     loop_var: user
 22
 23

```

5.Create main playbook file & include tasks folders file

Playbook.yml

Main Playbook file: Playbook.yml



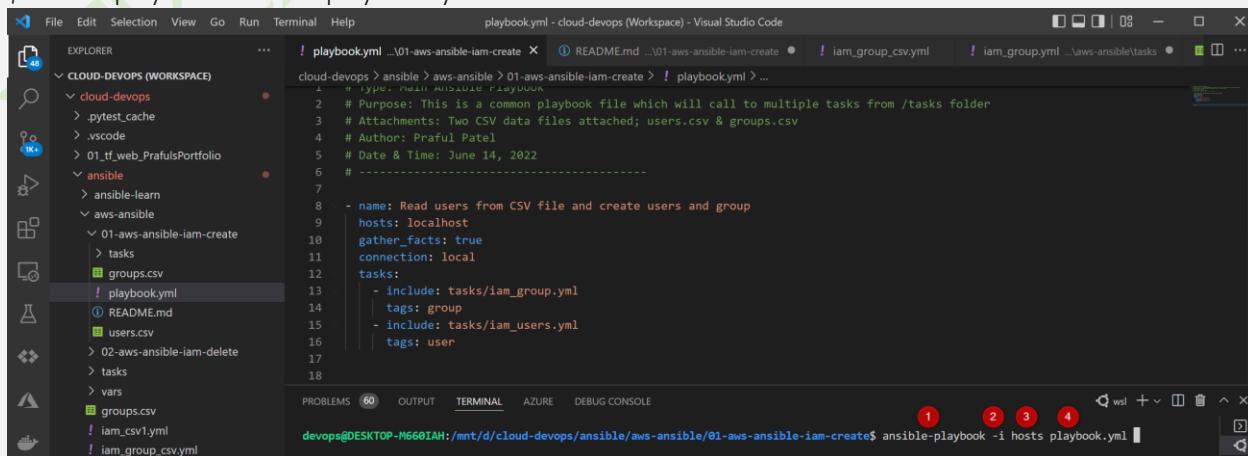
```

cloud-devops > ansible > aws-ansible > 01-aws-ansible-iam-create > playbook.yml > ...
  Ansible Playbook (ansible.json)
  1 # Type: Main Ansible Playbook
  2 # Purpose: This is a common playbook file which will call to multiple tasks from /tasks folder
  3 # Attachments: Two CSV data files attached; users.csv & groups.csv
  4 # Author: Praful Patel
  5 # Date & Time: June 14, 2022
  6 # -----
  7
  8 - name: Read users from CSV file and create users and group
  9   hosts: localhost
 10   gather_facts: true
 11   connection: local
 12   tasks:
 13     - include: tasks/iam_group.yml
 14       tags: group
 15     - include: tasks/iam_users.yml
 16       tags: user
 17
 18
 19
 20

```

6.Run Playbook

\$ansible-playbook -i hosts playbook.yml



```

cloud-devops > ansible > aws-ansible > 01-aws-ansible-iam-create > playbook.yml > ...
  # Type: Main Ansible Playbook
  1 # Purpose: This is a common playbook file which will call to multiple tasks from /tasks folder
  2 # Attachments: Two CSV data files attached; users.csv & groups.csv
  3 # Author: Praful Patel
  4 # Date & Time: June 14, 2022
  5 # -----
  6
  7
  8 - name: Read users from CSV file and create users and group
  9   hosts: localhost
 10   gather_facts: true
 11   connection: local
 12   tasks:
 13     - include: tasks/iam_group.yml
 14       tags: group
 15     - include: tasks/iam_users.yml
 16       tags: user
 17
 18
 19
 20

```

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

Groups: Groups successfully created automated way

Identity and Access Management (IAM)

User groups (6) [Info](#)

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Filter User groups by property or group name and press enter

Group name	Users	Permissions	Creation time	
CloudAdmin	1	↳ Loading	↳ Loading	8 months ago
DBA	2	↳ Loading	↳ Loading	7 minutes ago
LinuxAdmin	3	↳ Loading	↳ Loading	7 minutes ago
NetworkAdmin	4	↳ Loading	↳ Loading	7 minutes ago

Users: Users successfully created automated way

Users (9) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Find users by username or access key

User name Groups Last activity MFA Password age Active key age

1 michael.scott	NetworkAdmin	5	Never	None	2 minutes ago	-
2 jane.doe	DBA	6	Never	None	2 minutes ago	-
3 brian.garcia	LinuxAdmin	7	Never	None	2 minutes ago	-
4 barbara.brown	CloudAdmin	8	Never	None	2 minutes ago	-
ansible-user	None		3 minutes ago	None	None	Yesterday

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

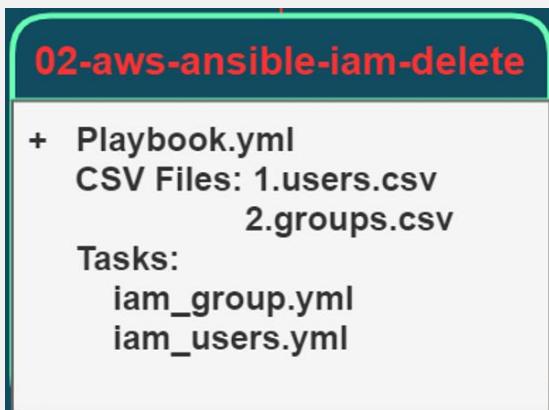
Solution 2: Create IAM group and users in AWS using Ansible

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

	groupname
1	DBA
2	NetworkAdmin
3	CloudAdmin
4	LinuxAdmin

 The file is located in a folder structure: cloud-devops > ansible > aws-ansible > 02-aws-ansible-iam-delete > groups.csv. A red arrow points to the 'groupname' header in the CSV file."/>

2.users.csv

```

  name,group,password
  jane.doe,DBA,ChangeMe123456!
  michael.sc,NetworkAdmin,ChangeMe123456!
  barbara.b,CloudAdmin,ChangeMe123456!
  brian.garc,LinuxAdmin,ChangeMe123456!
  
```

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

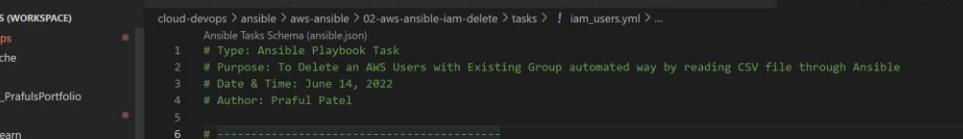
```

  # Type: Ansible Playbook Task
  # Purpose: To Delete an AWS Groups automated way by reading CSV file through Ansible
  # Author: Praful Patel
  # Date & Time: June 14, 2022

  - name: Read group from CSV file and return a dictionary
    read_csv:
      path: /mnt/d/cloud-devops/ansible/02-aws-ansible-iam-delete/groups.csv
      register: grouplist

  - name: Delete IAM groups from aws
    iam:
      iam_type: group
      name: "{{ grouplist.list }}"
      state: absent
    loop: "{{ grouplist.list }}"
    loop_control:
      loop_var: group
  
```

2. create iam_users.yml

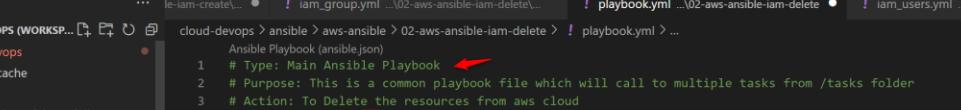


```
cloud-devops > ansible > aws-ansible > 02-aws-ansible-iam-delete > tasks > iam_users.yml > ...
Ansible Tasks Schema (ansible.json)
1 # Type: Ansible Playbook Task
2 # Purpose: To Delete an AWS Users with Existing Group automated way by reading CSV file through Ansible
3 # Date & Time: June 14, 2022
4 # Author: Praful Patel
5
6 #
7 ---
8 - name: Read users from CSV file and return a dictionary
9   read_csv:
10     path: /mnt/d/cloud-devops/ansible/aws-ansible/02-aws-ansible-iam-delete/users.csv
11     register: userlist
12
13
14 - name: Delete IAM Users from aws
15   iam:
16     iam_type: user
17     name: "{{ user.name }}"
18     password: "{{ user.password }}"
19     state: absent
20     groups: "{{ user.group }}"
21   loop: "{{ userlist.list }}"
22   loop_control:
23     loop_var: user
```

5. Create main playbook file & include tasks folders file

Playbook.yml

Main Playbook file: Playbook.yml



The screenshot shows the VS Code interface with the title bar "playbook.yml - cloud-devops (Workspace) - Visual Studio Code". The Explorer sidebar on the left lists files and folders related to the "CLOUD-DEVS (WORKSPACE)" project, including "iam_group.yml", "playbook.yml", and "iam_users.yml". The main editor area displays the "playbook.yml" file for an Ansible playbook. The file content is as follows:

```
Ansible Playbook (ansible.json)
1 # Type: Main Ansible Playbook
2 # Purpose: This is a common playbook file which will call to multiple tasks from /tasks folder
3 # Action: To Delete the resources from aws cloud
4 # Date & Time: June 14, 2022
5 # Author: Praful Patel
6 #
7
8 - name: Read users from CSV file and delete users and group from aws
9   hosts: localhost
10  gather_facts: true
11  connection: local
12  tasks:
13    - include: tasks/iam_users.yml
14      tags: user
15    - include: tasks/iam_group.yml
16      tags: group
```

Annotations with red arrows highlight the following text in the file:

- A red arrow points to the first line of the file: "# Type: Main Ansible Playbook".
- A red arrow points to the second line of the file: "# Purpose: This is a common playbook file which will call to multiple tasks from /tasks folder".
- A red arrow points to the third line of the file: "# Action: To Delete the resources from aws cloud".
- A red arrow points to the "gather_facts: true" line in the tasks section.
- A red arrow points to the "connection: local" line in the tasks section.
- A red arrow points to the "tags: user" line in the tasks section.
- A red arrow points to the "tags: group" line in the tasks section.

6. Run Playbook

\$ ansible-playbook -I hosts playbook.yml

```

# Type: Main Ansible Playbook
# Purpose: This is a common playbook file which will call to multiple tasks from /tasks folder
# Action: To Delete the resources from aws cloud
# Date & Time: June 14, 2022
# Author: Praful Patel

- name: Read users from CSV file and delete users and group from aws
  hosts: localhost
  gather_facts: true
  connection: local
  tasks:
    - include: tasks/iam_users.yml
      tags: user
    - include: tasks/iam_group.yml
      tags: group

```

```

PLAY [Read users from CSV file and delete users and group from aws] ****
ok: [localhost]
[WARNING]: Unable to parse /mnt/d/cloud-devops/ansible/aws-ansible/02-aws-ansible-iam-delete/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

TASK [Delete IAM Users from aws] ****
ok: [localhost]
changed: [localhost] => (item={'name': 'jane.doe', 'group': 'DBA', 'password': 'ChangeMe123456!'})
changed: [localhost] => (item={'name': 'michael.scott', 'group': 'NetworkAdmin', 'password': 'ChangeMe123456!'})
changed: [localhost] => (item={'name': 'barbara.brown', 'group': 'CloudAdmin', 'password': 'ChangeMe123456!'})
changed: [localhost] => (item={'name': 'brian.garcia', 'group': 'LinuxAdmin', 'password': 'ChangeMe123456!'})
[WARNING]: Module did not set no_log for update_password

TASK [Read group from CSV file and return a dictionary] ****
ok: [localhost]

TASK [Delete IAM groups from aws] ****
changed: [localhost] => (item={'groupname': 'DBA'})
changed: [localhost] => (item={'groupname': 'NetworkAdmin'})
changed: [localhost] => (item={'groupname': 'CloudAdmin'})
changed: [localhost] => (item={'groupname': 'LinuxAdmin'})

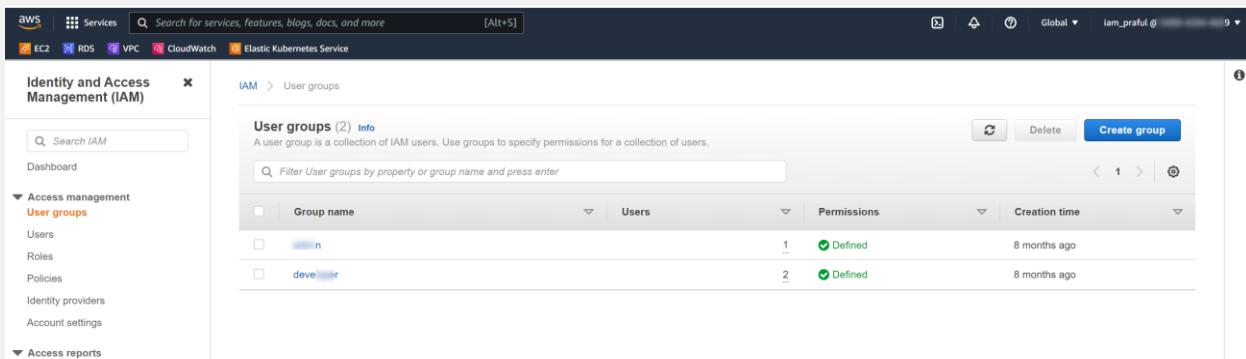
PLAY RECAP ****
localhost : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

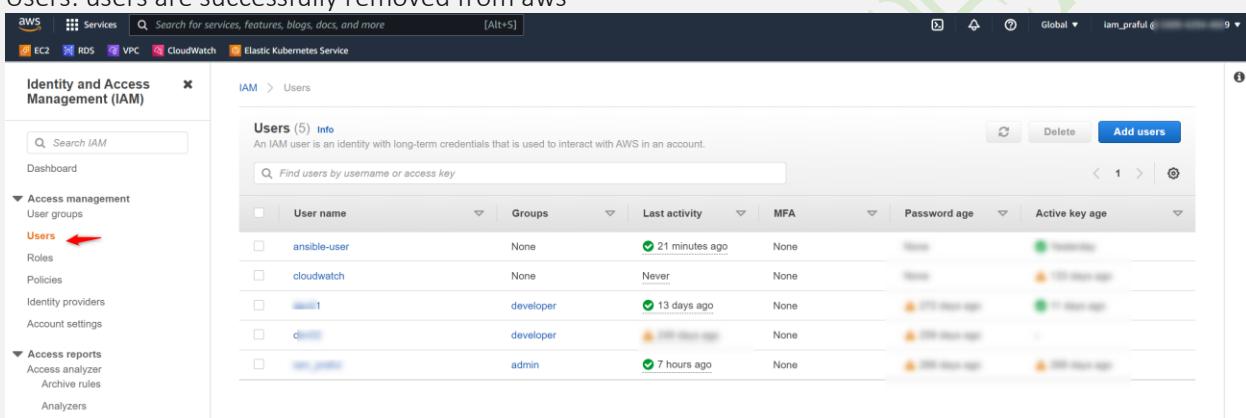
7.Verify that users and group are deleted from AWS cloud

Groups: Groups are successfully removed from aws

Users: users are successfully removed from aws



Group name	Users	Permissions	Creation time
n	1	Defined	8 months ago
deve	2	Defined	8 months ago



User name	Groups	Last activity	MFA	Password age	Active key age
ansible-user	None	21 minutes ago	None	Never	Recently
cloudwatch	None	Never	None	Never	100 days ago
n	developer	13 days ago	None	270 days ago	11 days ago
deve	developer	100 days ago	None	290 days ago	-
admin	admin	7 hours ago	None	290 days ago	290 days ago



Congratulations!!!! 🎉