**Alex has been assigned the task of setting up an IAM user with specific permissions. These permissions consist of:**

* **Full Access IAM User.**
* **Full Access AWS Code Commit.**
* **Full Access S3 Service.**

1. Go to 'IAM' in AWS Console.
2. Click 'Create User'.
3. Give the User Name 'Alex' & click 'Next'.
4. Choose 'Attach Policies Directly'.
5. Search and Add 'IAMFullAccess'.
6. Search and Add 'AWSCodeCommitFullAccess'.
7. Search and Add 'AmazonS3FullAccess'.
8. Click on 'Next'.
9. Click on 'Create User'.
10. Go to 'Users'.
11. Click on 'Alex' User Name.
12. Go to 'Security Credentials'.
13. In Access Keys Section, click on 'Create Access Key'.
14. In Use Case section, select 'Command Line Interface (CLI)'.
15. Click 'Create Access Key'.
16. Download the '.csv file'.
17. Go to 'Security Credentials'.
18. In HTTPS Git credentials for AWS CodeCommit, click 'Generate Credentials'.
19. Click 'Download Credentials'.
20. Go to Console Sign-in Section, click 'Enable Console Access'.
21. Set Console Access to Enable.
22. Set a custom password and click 'Apply'.
23. Download the '.csv file'.

**In Rohan’s IT Company, there are two other team members, Shubham and Ronit, working on two different task:**

* **Shubham is working on ‘task1.py’ in the ‘test1’ branch and is responsible for uploading the data to Github.**
* **Ronit is working on ‘task2.py’ in the ‘test2’ branch and is responsible for uploading the data to Github.**
* **Rohan has a ‘master’ branch with a file named ‘requirement.txt’.**
* **Last merge ‘test1’ and ‘test2’ branch to the ‘master’ branch.**

1. Create a folder named 'Rohan IT Company' on desktop.

2. Open the folder in VS Code.

3. Create 'task1.py', 'task2.py' and 'requirement.txt' in the folder.

4. Open AWS and Search CodeCommit and Enter.

5. Click on 'Create Repository'

6. Name the repository 'Rohan IT Company' and Click 'Create'.

7. Click on 'Clone URL' and Click 'Clone HTTPS'

8. Open terminal of the folder.

9. Code the command in the terminal :-

git init

git remote add origin (Paste Clone HTTPS)

git add .\requirement.txt

git commit -m "master Branch"

git push -u origin master

git branch test1

git checkout test1

git add .\task1.py

git commit -m "test1 Branch"

git push -u origin test1

git branch test2

git checkout test2

git add .\task2.py

git commit -m "test2 Branch"

git push -u origin test2

git checkout master

git merge test1

git commit -m "Merged test1 into master"

git push -u origin master

git merge test2

git commit -m "Merged test2 into master"

git push -u origin master

**Create new EC2 with name EC2-my-server-rollno.**

**Configure for AWS-CLI (Inside EC2)**

**Display list of buckets in S3**

1. Access your AWS Console and navigate to the EC2 service.
2. In the EC2 dashboard, select "Instances" from the left-hand menu.
3. Click the "Launch an Instance" button to initiate the instance creation process.
4. Provide a name for your instance to help identify it later in your AWS environment.
5. In the "Application and OS Images" section, choose "Ubuntu" as the desired operating system and application.
6. In the "Key Pair" section, opt for "Proceed without a key pair" if you don't need SSH key authentication.
7. In the "Network Settings" section, configure the following security group rules:

* Allow SSH traffic from the internet
* Allow HTTPS traffic from the internet
* Allow HTTP traffic from the internet

1. Launch
2. Go to Instances and choose the created EC2 and click Start Instance from Instance State and click Connect from the Actions.
3. For configuration, download the package using the following commands.

sudo apt-get update

sudo apt-get install unzip

sudo apt-get install python-pip

sudo apt install python3-pip

pip install boto3

pip install mysql-connector-python

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

1. Enter the following code to configure

aws configure

AWS Access Key ID [None]: [Enter Access Key ID Of IAM User from access\_key.csv]

AWS Secret Access Key [None]: [Enter Secret Key]

Default region name [None]: [eu-north-1]

Default output format [None]: [JSON]

**What are the sequential steps for an employee in an IT company working on a project that involves the following actions:**

* **Utilizing the get.py file to download data from an S3 bucket.**
* **Establishing a CodeCommit repository.**
* **Uploading the get.py file to the created CodeCommit repository.**
* **Using AWS-CLI to upload a data.csv file to a newly created S3 bucket.**
* **Performing a Git Clone operation within an EC2 instance.**
* **Employing the Nano text editor to modify the get.py file by adding the statement print("The task is done").**
* **Executing the modified get.py file.**
* Create a data.csv in the folder:
* Create a new folder.
* Inside the folder, create a file named data.csv.
* Upload the data.csv to S3 Bucket using AWS CLI
* Open the terminal in the folder.
* Execute the following commands:

aws s3 mb s3://csvfolder0608

aws s3 cp ./data.csv s3://csvfolder0608

aws s3 ls s3://csvfolder0608

* Code the get.py file:
* Open the folder and create/get.py file.
* Add the following code to get.py:

import boto3

def download\_from\_s3(bucket\_name, s3\_key, file\_name):

s3 = boto3.client('s3')

try:

s3.download\_file(bucket\_name, s3\_key, file\_name)

print("File downloaded successfully from S3")

except Exception as e:

print(f"An error occurred: {str(e)}")

bucket\_name = 'csvfolder0608'

s3\_key = 'data.csv'

file\_name = 'DownloadedData.csv'

download\_from\_s3(bucket\_name, s3\_key, file\_name)

* Create a CodeCommit repo:
* Open the terminal in the folder.
* Execute the following commands:

git init

git remote add origin <codecommit HTTPS Link>

git add get.py

git commit -m "-"

git push -u origin master

* Clone CodeCommit Repo in EC2:
* Open the EC2 instance.
* Execute the following command:

git clone <codecommit HTTPS Link>

* Edit and run the get.py:

nano get.py

python3 get.py

ls