**Project Details:**

**- Sai Niveditha Sahu**

**- Team Name: Cloud 03**

**- Project Statement: AWS Backup Automation Script**

**Objective: The objective of this project is to create a script that automates the backup process of an application running on an EC2 instance and saves the backup data to an S3 bucket in the AWS cloud. The script should provide a reliable and efficient way to back up the application data and ensure the availability and durability of the backups.**

**Step by step instruction:**

**Step 1: Setup AWS Resources**

1.1 Create an EC2 instance, an IAM role, and an S3 bucket using the AWS Management Console.

**Step 2: Connect to EC2 Instance**

2.1 Connect to the EC2 instance and attach the IAM role to it.

**Step 3: Install Dependencies and Configure Cron Job**

3.1 Run the following commands on the terminal:

**sudo yum install python3.11 python3-pip**

**sudo yum install cronie -y**

**sudo systemctl enable crond.service**

**sudo systemctl start crond.service**

**python3 -m pip install paramiko boto3**

**Step 4: Create Directories and Backup Script**

4.1 Create two directories named 'appdata' and 'script'.

4.2 Navigate to the 'script' directory: cd script

4.3 Create a backup script: vim backupscript.py

4.4 Copy the backup script code into the vim text editor.

**import os**

**import shutil**

**import zipfile**

**import boto3**

**from botocore.exceptions import NoCredentialsError**

**def backup\_and\_upload\_to\_s3(local\_directory, s3\_bucket\_name):**

**try:**

**# Create a temporary directory for the backup**

**backup\_directory = "appdata\_backup\_temp"**

**os.makedirs(backup\_directory)**

**# Copy the /home/ec2-user/appdata directory to the temporary backup directory**

**appdata\_backup\_path = os.path.join(backup\_directory, "appdata\_backup")**

**shutil.copytree(local\_directory, appdata\_backup\_path)**

**# Create a ZIP file of the backup**

**zip\_filename = "appdata\_backup.zip"**

**with zipfile.ZipFile(zip\_filename, "w", zipfile.ZIP\_DEFLATED) as zip\_file:**

**for root, \_, files in os.walk(appdata\_backup\_path):**

**for file in files:**

**file\_path = os.path.join(root, file)**

**zip\_file.write(file\_path, arcname=os.path.relpath(file\_path, appdata\_backup\_path))**

**# Upload the ZIP file to S3**

**s3 = boto3.client('s3')**

**s3.upload\_file(zip\_filename, s3\_bucket\_name, zip\_filename)**

**print("Backup and upload to S3 completed successfully!")**

**except NoCredentialsError:**

**print("Credentials not available or invalid. Please configure AWS credentials.")**

**except Exception as e:**

**print(f"Backup and upload to S3 failed: {e}")**

**finally:**

**# Clean up temporary files**

**if os.path.exists(backup\_directory):**

**shutil.rmtree(backup\_directory)**

**if os.path.exists(zip\_filename):**

**os.remove(zip\_filename)**

**# Replace 'your-s3-bucket-name' with your actual S3 bucket name**

**backup\_and\_upload\_to\_s3(local\_directory='/home/ec2-user/appdata', s3\_bucket\_name='give bucketname'**)

**Step 5: Install Python and Pip on Windows**

5.1 Install Python on your Windows system.

5.2 Open PowerShell and download and install pip:

**curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py**

**python get-pip.py**

**python -m pip install paramiko**

**Step 6: Modify Backup Script and Execute on Windows**

6.1 Modify 'backupscript.py' with your S3 bucket name.

6.2 Navigate to the script location: cd .\Downloads\

6.3 Execute the backup script: python aws\_backup.py

**Step 7: Verify Cron Job**

7.1 Verify the cron job is scheduled on the Linux server:

**bash**

copy code

**crontab –l**

**\* \* \* \* \* python3 /home/ec2-user/script/backupscript.py**

**Step 8: File Creation and Verification**

8.1 Go back to the parent directory: cd ..

8.2 Create sample files in 'appdata': touch appdata/sample1 appdata/sample2

8.3 Verify that the files are backed up in the S3 bucket.

8.4 Verify the zip file by downloading on your system.

**Recommendations for Enhancement**

* **Utilize AWS Secret Manager to store SSH hostname and public IP securely.**
* **Enhance the backup script to prompt for the S3 bucket name as user input.**
* **Document procedures for Windows, utilizing Task Scheduler.**
* **Consider using Ansible for automation across different systems**