***Project: AWS Resource Usage Script***

**Objective**

* This project focuses on creating a comprehensive script using the AWS CLI to gather vital insights into AWS resource usage. By querying information about S3 buckets, EC2 instances, Lambda functions, and IAM users, the script aims to streamline the monitoring and management of AWS resources.

**Script Components**

The script consists of several AWS CLI commands to retrieve the information:

* *AWS S3 Buckets*
* This command is used to list all S3 buckets in the AWS account: ***aws s3 ls***
* *AWS EC2 Instances*
* This command is used to provide detailed information about all running AWS EC2 instances: ***aws ec2 describe-instances***
* *AWS Lambda Functions*
* This command is used to list all Lambda functions deployed in the AWS account: ***aws lambda list-functions***
* *AWS IAM Users*
* This command is used to retrieve a list of all IAM users associated with the AWS account: ***aws iam list-users***

**Additional Features:**

* *Debug Mode*
* By improvising the script, **“set -x”** command is added to enable the debug mode. This option displays each executed command and its output, enhancing visibility during the script execution.
* *Data Extraction with “jq” command*
* The **“jq”** command is used to extract specific information from the output of AWS CLI commands. Namely, the specific data can be printed from AWS Services. This option is developed the script functionality. For example:

***aws ec2 describe-instances | jq ".Reservations[].Instances[].InstanceId"***

This retrieves the **“InstanceId”** of each AWS EC2 instance.

**Running the Script (aws\_resourceTracker.sh)**

On Your Local Machine

* Execute the script using this command: ***./<YOUR FILE NAME>.sh***
* Edit or read the script using this command: ***vim <YOUR FILE NAME>.sh***
* If the PERMISSION issues arise, temporarily adjust permissions using this command: ***chmod 777 <YOUR FILE NAME>.sh***. Then, the script will properly work.

**AWS CLI Command Reference**

* If you don’t remember the commands, you can look at the AWS CLI Command Reference. Is it very straightforward.

***Initial Setup***

**AWS EC2 Instance**

You have to make sure that you created or selected existing AWS EC2 instance. Firstly, I would like to mention that you can do all these processes in AWS Console, however, this way is longer than connecting from the local terminal. Because, as a DevOps Engineer, you might connect to hundreds of servers every day. You cannot do this process each time since it isn’t efficient and time saving. As a DevOps Engineer, you must be efficient.

If you prefer to work on your local terminal, then you have to connect AWS EC2 instance from your terminal. My terminal is **MobaXterm**. To connect the EC2 instance, the **“KEY VALUE”** is needed. It is created while launching the new EC2 instance and must be downloaded. If you are using Ubuntu, you have to download it in **<PEM FILE>.pem**. This KEY VALUE is needed to login your EC2 instance from your terminal and you must not share it with anyone. It is confidential.

A screenshot of a computer

Description automatically generated

You connect your EC2 instance with the following command: ***ssh -i <YOUR PATH TO PEM FILE>/<PEM FILE>.pem ubuntu@<PUBLIC IPv4 ADDRESS>***

However, you can come across with permission problem. It can be solved by using this command: ***chmod 600 <YOUR PATH TO PEM FILE>/<PEM FILE>.pem***. Then, it will properly work.

**AWS CLI Installation**

You must make sure that you already have **AWS CLI (Command Line Interface)** in your computer. You can check it with ***“aws version”*** command. If a lot of commands pop up, it means you have CLI. However, if you don’t have, you can install by using ***“sudo apt install aws cli”*** command or via this link: <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

After downloading this, you have still some stuffs to do. Because you only downloaded AWS CLI binary which doesn’t have your user information or account information. That is why you have to authenticated with your AWS console. You must have **“Security credentials”** which consist of **“Access Key”** and **“Secret Access Key”**. You must write it in your terminal by using “***aws configure”*** command. Then, AWS **Access Key ID, AWS Secret Access Key, Default region name** and **Default output format** will be asked to enter. Once you entered, you will finish to authenticate with your AWS. Never share these keys with anyone.