**AWS Project using SHELL SCRIPTING for DevOps**

## **Introduction**

In the real-world DevOps scenario, The **AWS Resource Tracker** script is widely used to provide an overview of the AWS resources being utilised within an environment.

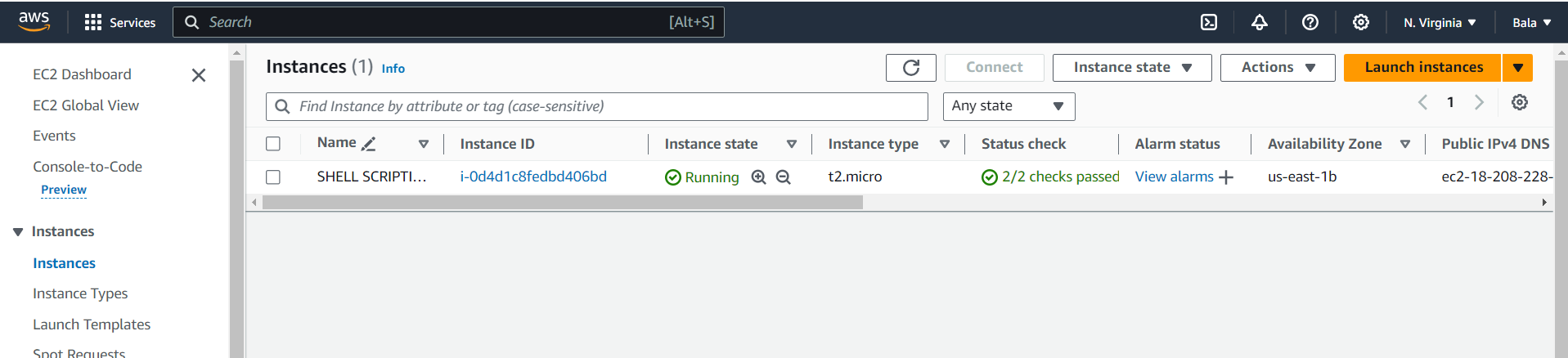
It aims to help organisations monitor and manage their AWS resources effectively. The script utilises the AWS Command Line Interface (**CLI**) to fetch information about different AWS services, such as **S3 buckets, EC2 instances, Lambda functions, and IAM users**.

## **What does this project do?**

By running the AWS Resource Tracker script, users can quickly obtain a list of S3 buckets, EC2 instances, Lambda functions, and IAM users associated with their AWS account.

This information can be valuable for various purposes, including auditing, inventory management, resource optimisation, and security assessment.

### **Create EC2 Instance**



**Copy the public IP address and Open command prompt**

* Give the below command

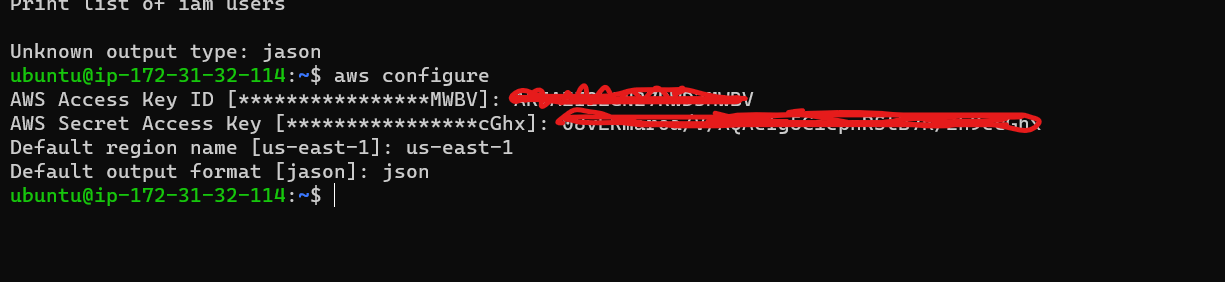
**ssh -i /Users/reshm/Downloads/demo.pem** [**ubuntu@18.208.228.31**](mailto:ubuntu@18.208.228.31)

To install the AWS Command Line Interface (CLI) on an Ubuntu machine, you can follow these steps:

1. sudo apt update
2. sudo apt install python3 python3-pip
3. sudo pip3 install awscli
4. aws –version

After installing aws-cli, now give aws configuration

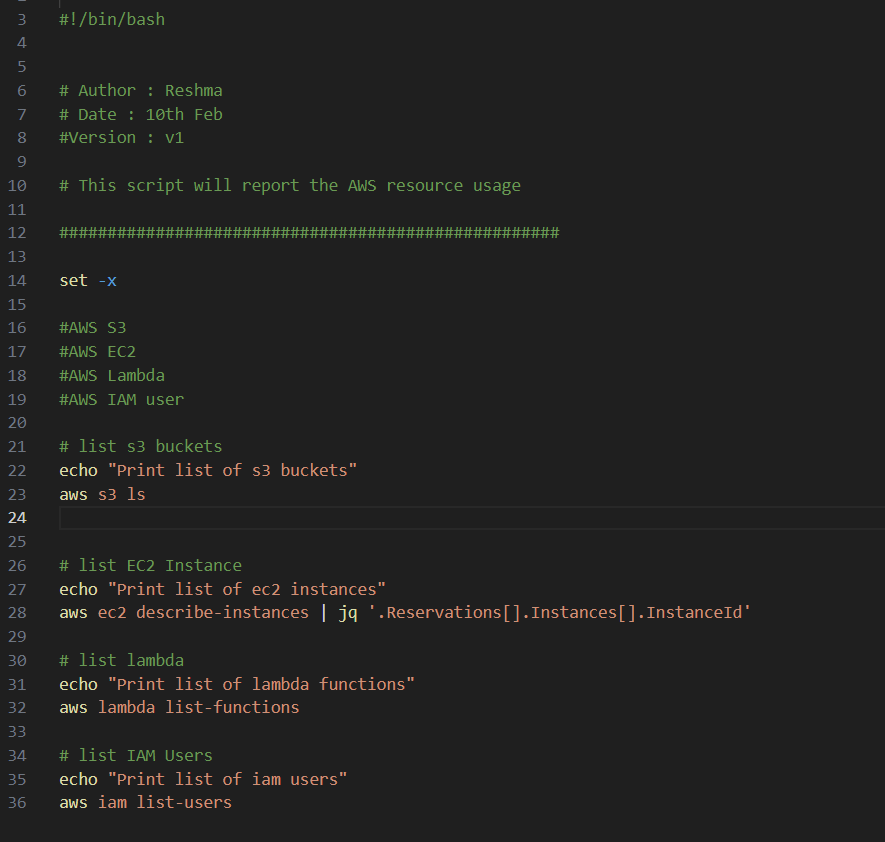
* Aws configure
* It will ask for access key and secret access key
* Get it from IAM user and give the credentials with region and default format.



* Open the vi editor vi aws\_resource\_tracker.sh
* It is a good convention to always provide your details so that it will make it easier for other developers to contribute if they want to ie, metadata.
* The command set -x is given here in order to debug the script, this will print the command which we are running and then prints the output.
* The commands aws s3 ls, aws lambda list-functions and aws iam list-users print the information of the given statement.

The command

aws ec2 describe instances | jq '.Reservations[].Instances[].InstanceId' will give you all the Instance IDs present in your aws in JSON format.

****

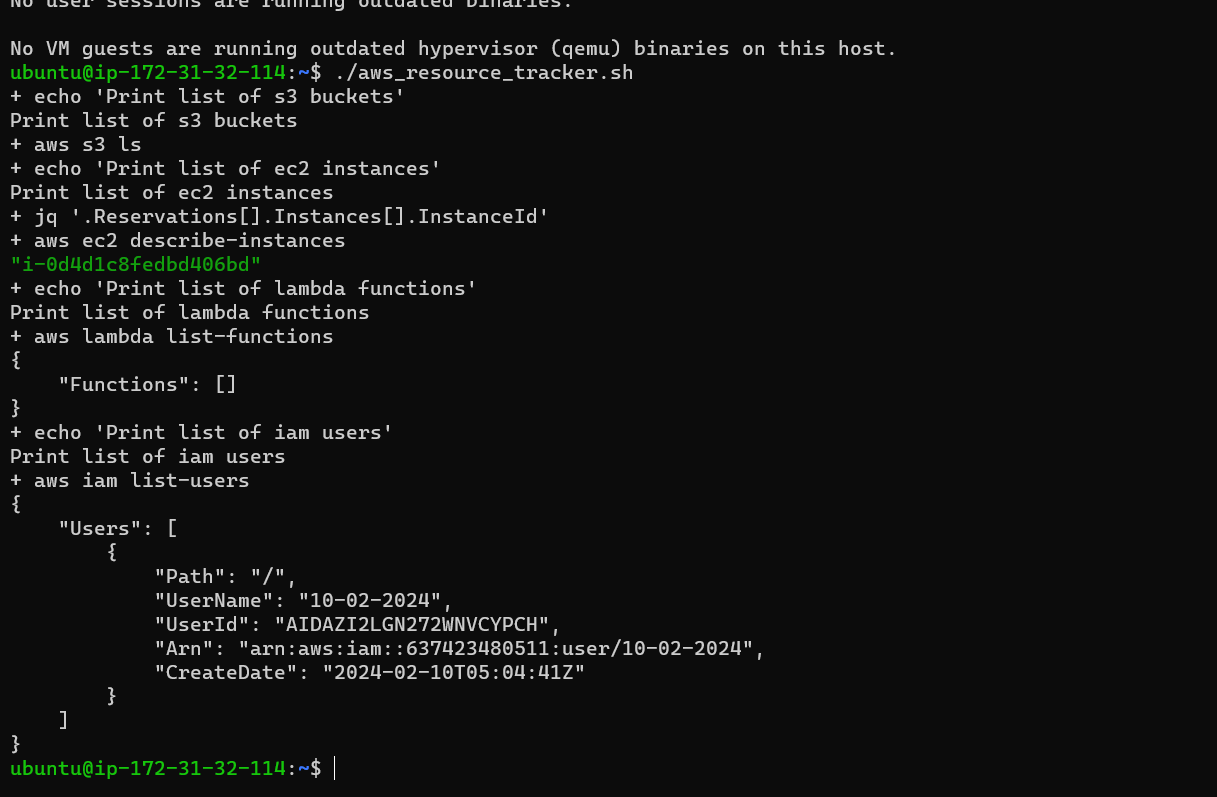
**:wq! And entre**

Run the below command before seeing the output.

* **sudo apt-get update**
* **sudo apt-get install jq**

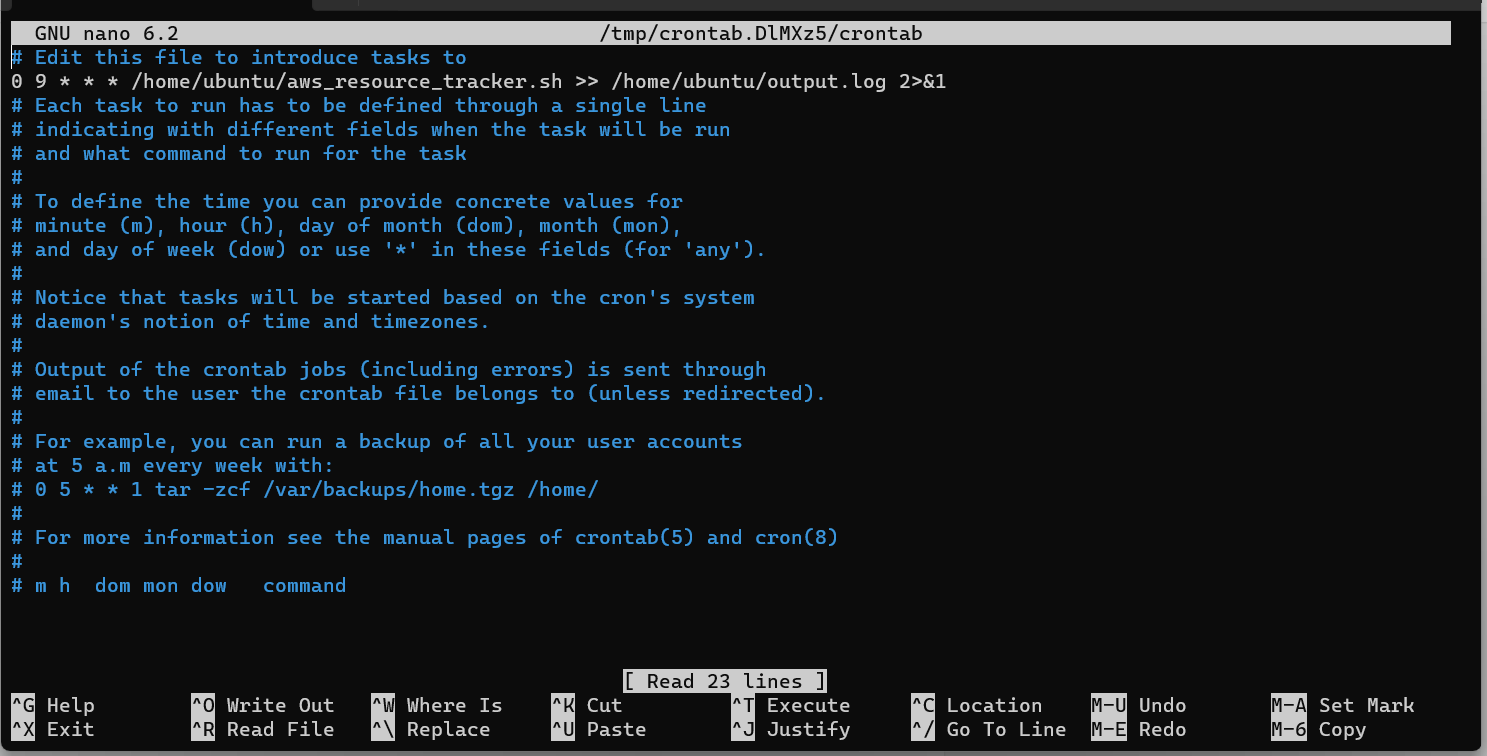
After installing run the command

* **./aws\_resource\_tracker.sh**

****

* Next We have to intricate this script to crontab and we get our result**:**

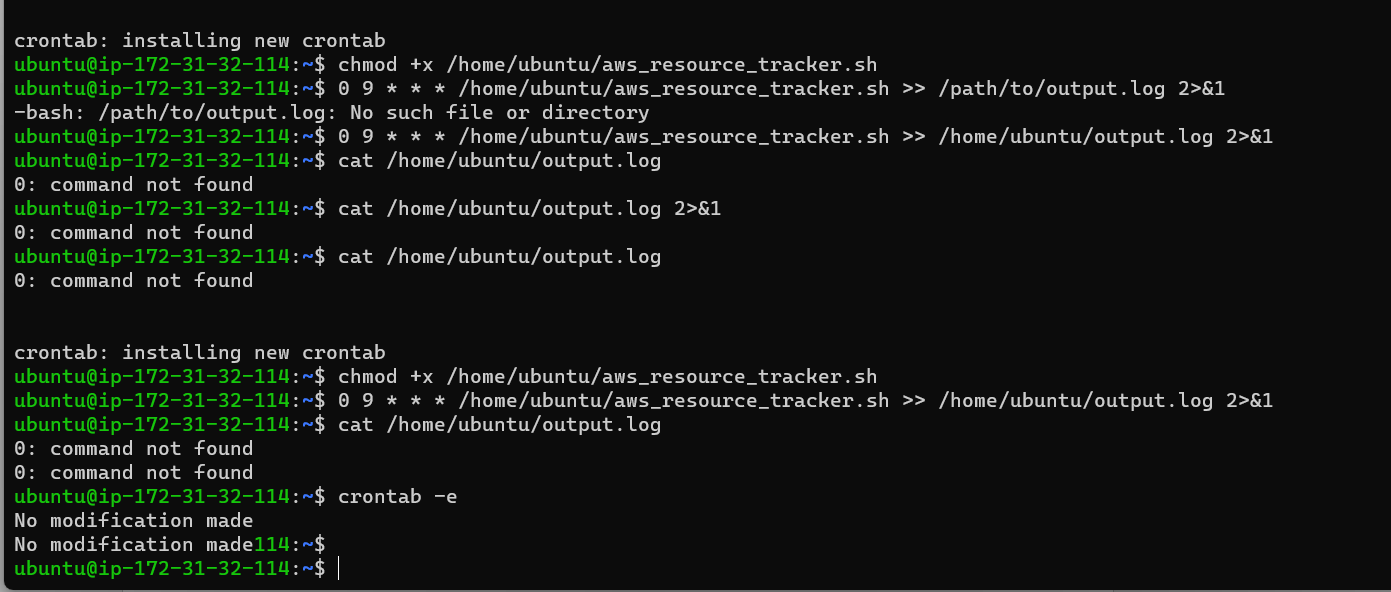
**crontab -e**

****

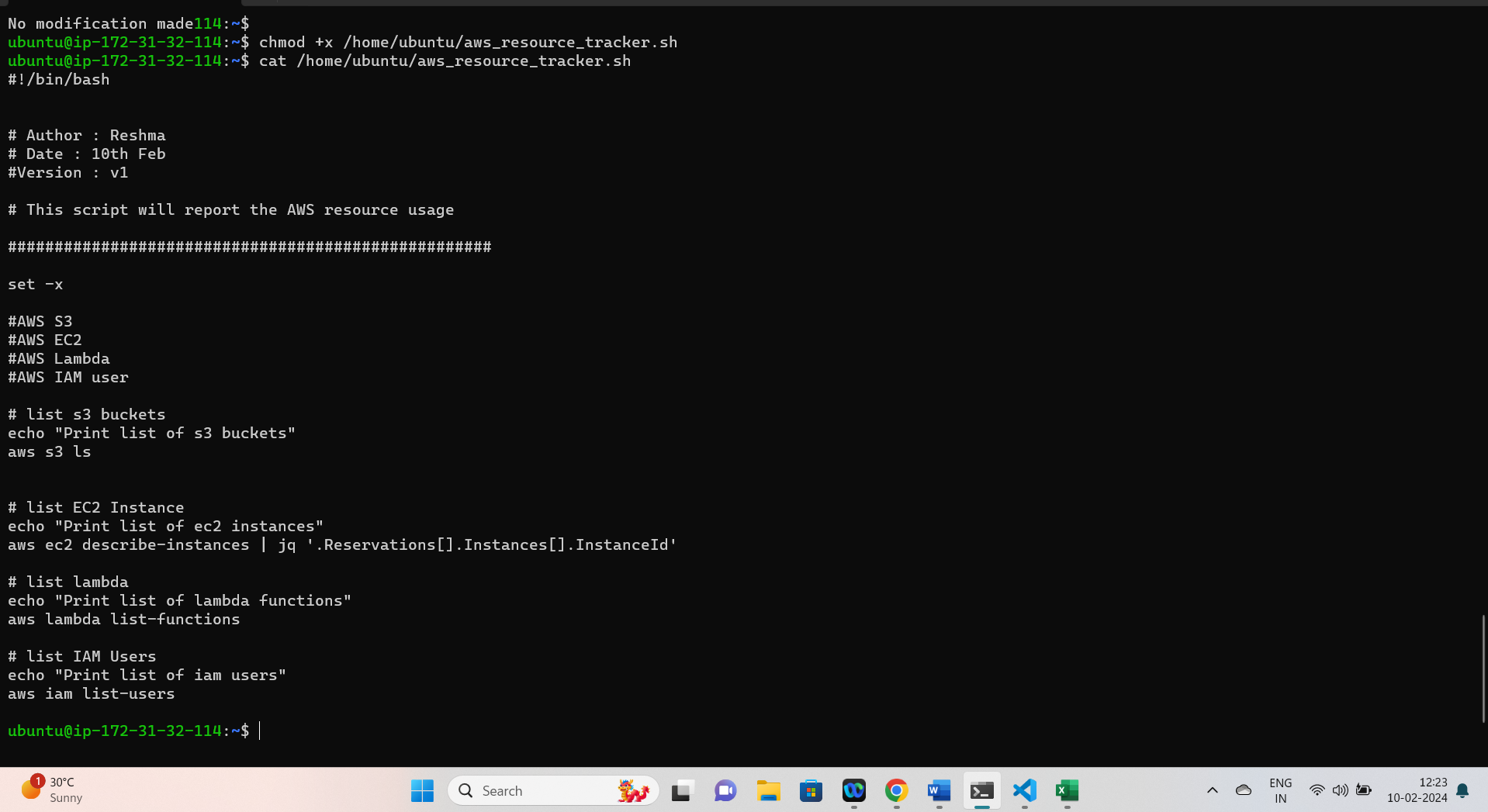
**Cntrl + O then entre**

**Cntrl + X exit**

* Giving permission**-** chmod +x /home/ubuntu/aws\_resource\_tracker.sh
* **0 9 \* \* \* /home/ubuntu/aws\_resource\_tracker.sh >> /home/ubuntu/output.log 2>&1**
* **cat /home/ubuntu/output.log**

****

* It showed error now give permission as given below we can see the output:

****

## **Use Cases**

* **Monitoring and auditing**: By running this script periodically using cron jobs, you can monitor and audit your AWS resources. It provides insights into the status and details of different resources, such as S3 buckets, EC2 instances, Lambda functions, and IAM users.
* **Resource inventory**: The script helps in maintaining an up-to-date inventory of your AWS resources. It lists the S3 buckets, EC2 instances, Lambda functions, and IAM users, allowing you to have a clear understanding of what resources exist in your environment.
* **Troubleshooting**: In case of any issues or incidents, this script can be used to quickly gather information about the relevant AWS resources. For example, if there is an issue with an EC2 instance, you can run the script to get the instance ID and other details for further investigation.
* **Automation and reporting**: The script can be integrated into an automated pipeline or workflow to generate regular reports about AWS resource usage. This information can be valuable for tracking costs, resource utilization, and compliance requirements.
* **Scalability and efficiency**: In larger environments with numerous AWS resources, manually retrieving information about each resource can be time-consuming and error-prone. By using this script, you can automate the process and retrieve resource details in a consistent and efficient manner.

Overall, this script simplifies the process of gathering information about AWS resources, enhances visibility into your infrastructure, and supports effective management and monitoring of your DevOps environment.