AWS CLI Details!

In the fast-paced world of cloud computing, managing and interacting with your resources efficiently is key to success. Amazon Web Services (AWS) provides a powerful tool for this purpose — the AWS Command Line Interface (CLI). The AWS CLI empowers users to interact with AWS services directly from the command line, offering flexibility, speed, and automation capabilities.

Unlock the power of AWS CLI with our step-by-step guide on configuration and command execution. Delve into the heart of cloud computing as we walk you through the installation process, credential setup, and practical examples for running essential AWS CLI commands. Whether you're a newcomer or an experienced user, this guide ensures you confidently navigate the AWS command-line landscape for efficient cloud resource management.

Below are step-by-step instructions to configure the AWS CLI on your local machine or on EC2 instance and run multiple commands:

Step 1: Install AWS CLI

Ensure that you have the AWS CLI installed on your local machine. You can download and install it from the official AWS CLI download page.

Step 2: Open a Terminal or Command Prompt

Open your terminal or command prompt on your local machine.

Step 3: Run `aws configure` Command

In the terminal, run the following command:

```
aws configure
```

Step 4: Enter AWS Access Key ID

Enter the AWS Access Key ID associated with your IAM user.

Step 5: Enter AWS Secret Access Key

Enter the AWS Secret Access Key associated with your IAM user.

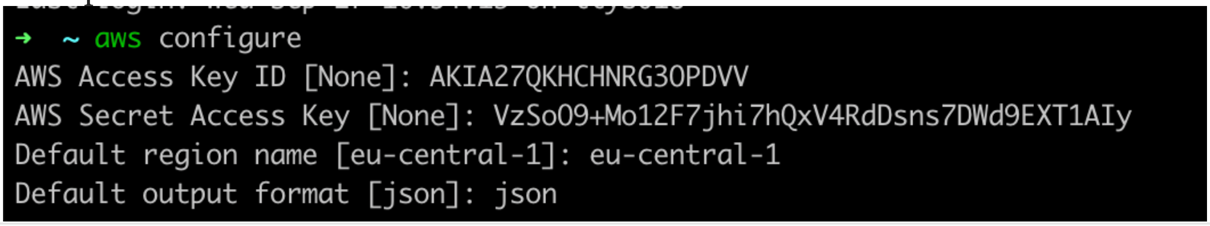
Step 6: Set Default Region

Enter the default region you want to use.

Step 7: Set Output Format

Choose the default output format.

Example Interaction:



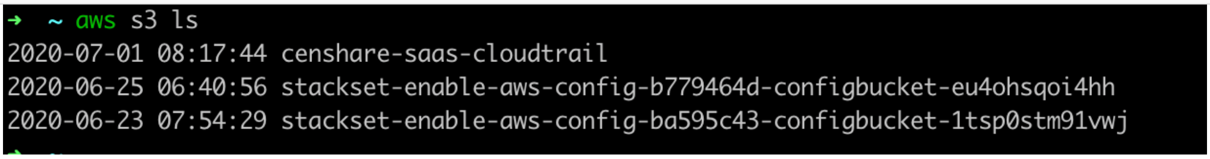
```
➜  ~ aws configure
AWS Access Key ID [None]: AKIA27QKHCHNRG3OPDVV
AWS Secret Access Key [None]: VzSo09+Mo12F7jhi7hQxV4RdDsns7DWd9EXT1AIy
Default region name [eu-central-1]: eu-central-1
Default output format [json]: json
```

```
AWS Access Key ID [None]: YOUR_ACCESS_KEY_ID
AWS Secret Access Key [None]: YOUR_SECRET_ACCESS_KEY
Default region name [None]: us-central-1
Default output format [None]: json
```

Step 8: Run AWS CLI Commands

Example 1: List S3 Buckets

```
aws s3 ls
```



```
➜  ~ aws s3 ls
2020-07-01 08:17:44 censhare-saas-cloudtrail
2020-06-25 06:40:56 stackset-enable-aws-config-b779464d-configbucket-eu4ohsqoi4hh
2020-06-23 07:54:29 stackset-enable-aws-config-ba595c43-configbucket-1tsp0stm91vwj
```

This command will list the S3 buckets in your AWS account.

Example 2: Describe EC2 Instances

```
aws ec2 describe-instances
```

This command will describe your EC2 instances in the default region.

Example 3: Create an S3 Bucket

```
aws s3 mb s3://your-new-bucket-name
```

Replace `your-new-bucket-name` with a unique name for your new S3 bucket.

Example 4: Upload a File to S3

```
aws s3 cp local-file.txt s3://your-new-bucket-name/
```

Replace `local-file.txt` with the path to a file on your local machine and `your-new-bucket-name` with the S3 bucket you just created.

Step 9: Verification

You can verify the results of your commands by checking the AWS Management Console or by running additional commands based on your needs.

Note:

Remember to replace placeholder values like `YOUR_ACCESS_KEY_ID`, `YOUR_SECRET_ACCESS_KEY`, `your-new-bucket-name`, and `local-file.txt` with your actual AWS credentials, bucket names, and file paths.

You can store multiple AWS CLI profiles by configuring named profiles in your AWS CLI configuration file (`~/.aws/config`). This allows you to switch between different AWS accounts or IAM roles easily.

a. Open or Create the AWS CLI Configuration File:

Open your terminal or command prompt and edit the AWS CLI configuration file. If the file doesn’t exist, it will be created:

```
nano ~/.aws/config
```

b. Define AWS CLI Profiles:

In the configuration file, you can define multiple profiles using the following syntax:

```
[profile profile_name]
region = aws_region
output = output_format
```

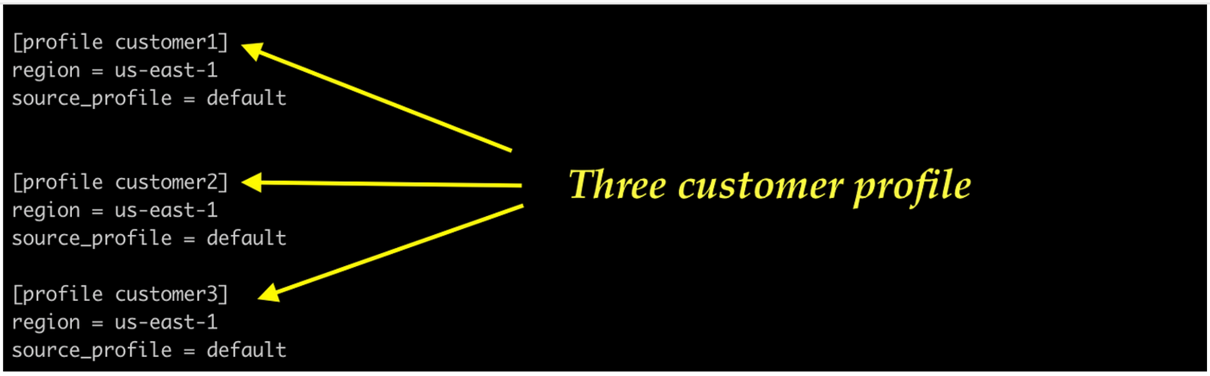
- `profile_name`: Replace this with a name for your profile.
- `aws_region`: Set the AWS region you want to associate with the profile.
- `output_format`: Choose the default output format for the profile, e.g., `json`, `text`, or `table`.

Here's an example of three profiles:

```
[profile customer1]
region = us-east-1
output = json

[profile customer2]
region = us-west-2
output = json

[profile customer3]
region = us-east-1
output = json
```



```
[profile customer1]
region = us-east-1
source_profile = default


[profile customer2]
region = us-east-1
source_profile = default

[profile customer3]
region = us-east-1
source_profile = default
```

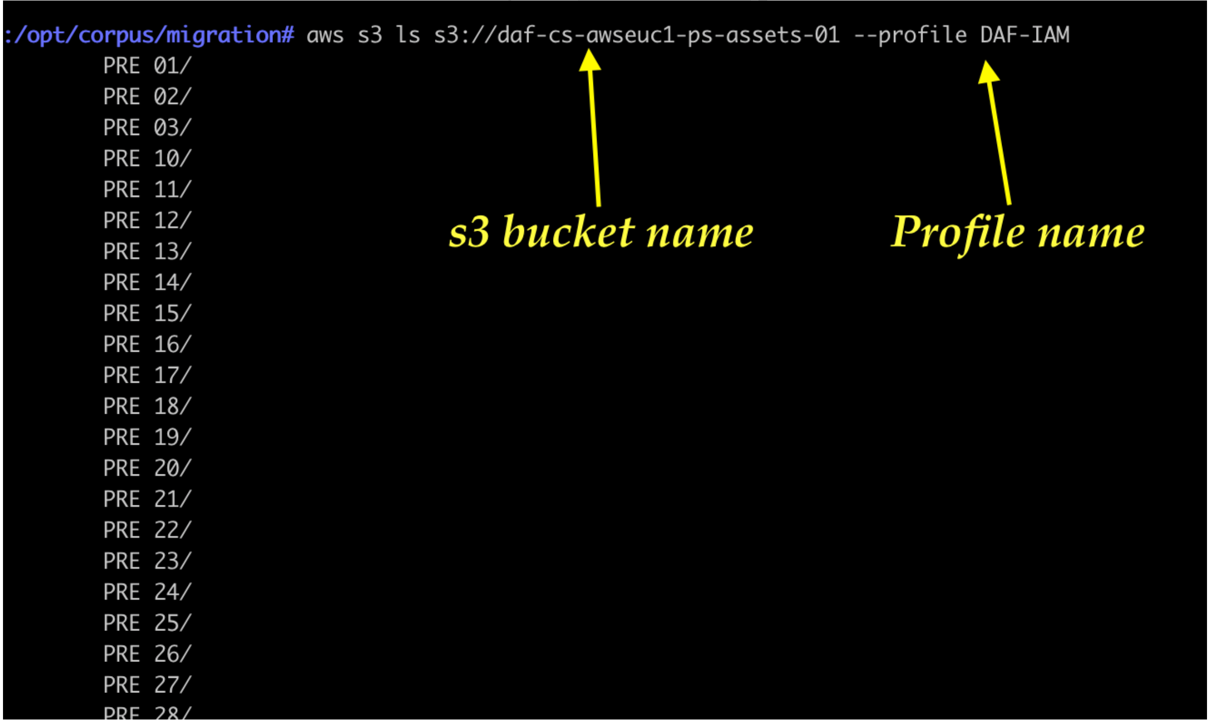
Save the changes to the configuration file and exit the text editor.

d. Configure AWS CLI to Use a Specific Profile:

You can specify the profile you want to use for a particular AWS CLI command by using the `--profile` option:

```
aws s3 ls s3://name-of-the-s3bucket --profile customer1
```

This command lists S3 buckets using the `customer1` profile.



```
:/opt/corpus/migration# aws s3 ls s3://daf-cs-awseuc1-ps-assets-01 --profile DAF-IAM
        PRE 01/
        PRE 02/
        PRE 03/
        PRE 10/
        PRE 11/
        PRE 12/
        PRE 13/
        PRE 14/
        PRE 15/
        PRE 16/
        PRE 17/
        PRE 18/
        PRE 19/
        PRE 20/
        PRE 21/
        PRE 22/
        PRE 23/
        PRE 24/
        PRE 25/
        PRE 26/
        PRE 27/
        PRE 28/
```

e. Switching Profiles:

To switch between profiles, simply use the `--profile` option with different profile names:

```
aws ec2 describe-instances --profile work
```

This command describes EC2 instances using the `work` profile.

By using named profiles, you can manage and switch between different AWS configurations easily, making it convenient to work with multiple AWS accounts or roles from the AWS CLI.