AWS CLI Fundamentals

The AWS Command Line Interface (CLI) is a tool that allows you to interact with AWS services from your terminal or command line. Here's a step-by-step guide to learning AWS CLI fundamentals:

1. Installation and Setup

Install AWS CLI

Windows / macOS / Linux:
Download the AWS CLI from the official AWS CLI installation page.
Follow the installation instructions for your operating system.

Verify Installation: Run the following command to verify that AWS CLI is installed:

```
aws --version
```

Output Example:

```
aws-cli/2.x.x Python/3.x.x Linux/x86_64
```

Configure AWS CLI

Set up your credentials and default region:

```
aws configure
```

You'll be prompted to enter:

- AWS Access Key ID: Found in the AWS Management Console under IAM.
- AWS Secret Access Key: Found in the same place as the Access Key ID.
- Default region: Specify your preferred region (e.g., us-east-1).
- Default output format: Choose json, text, or table.

2. AWS CLI Basics

List Available Services
To see the services AWS CLI supports:

```
aws help
```

View Command Options for a Service
For example, to view options for EC2:

```
aws ec2 help
```

3. Common AWS CLI Commands

S3 (Simple Storage Service)

List all S3 buckets:

```
aws s3 ls
```

Create an S3 bucket:

```
aws s3 mb s3://my-unique-bucket-name
```

Upload a file to an S3 bucket:

```
aws s3 cp myfile.txt s3://my-unique-bucket-name/
```

Download a file from an S3 bucket:

```
aws s3 cp s3://my-unique-bucket-name/myfile.txt .
```

Delete a bucket:

```
aws s3 rb s3://my-unique-bucket-name --force
```

EC2 (Elastic Compute Cloud)

List all EC2 instances:

```
aws ec2 describe-instances
```

Start an EC2 instance:

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Stop an EC2 instance:

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

Terminate an EC2 instance:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

IAM (Identity and Access Management)

List IAM users:

```
aws iam list-users
```

Create an IAM user:

```
aws iam create-user --user-name my-user
```

Delete an IAM user:

```
aws iam delete-user --user-name my-user
```

CloudFormation

List CloudFormation stacks:

```
aws cloudformation list-stacks
```

Deploy a CloudFormation stack:

```
aws cloudformation create-stack --stack-name my-stack --template-body file://template.yaml
```

Delete a CloudFormation stack:

```
aws cloudformation delete-stack --stack-name my-stack
```

AWS CLI Profiles

Create a Profile

Profiles allow you to use different credentials for multiple AWS accounts:

```
aws configure --profile myprofile
```

Use a Profile

To use a specific profile, include the --profile option:

```
aws s3 ls --profile myprofile
```

Filtering and Formatting

Filters

Use --query to filter the output:

```
aws ec2 describe-instances --query 'Reservations[*].Instances[*].InstanceId'
```

Output Formats

Specify output formats (json, text, or table) with --output:

```
aws ec2 describe-instances --output table
```

Automating Tasks with AWS CLI

Use AWS CLI in Scripts

You can include AWS CLI commands in Bash or PowerShell scripts for automation.

```
#!/bin/bash
aws s3 cp myfile.txt s3://my-unique-bucket-name/
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Use AWS CLI with Cron Jobs

Automate tasks like backups with a scheduled Cron job.

Debugging AWS CLI

Enable Debugging

Add --debug to any command to see detailed logs:

```
aws ec2 describe-instances --debug
```

Best Practices

- Secure Credentials: Use IAM roles and profiles instead of hardcoding keys.
- Limit Permissions: Follow the principle of least privilege for IAM policies.
- Use Output Redirection: Redirect outputs to files for logs or further processing.

  ```
  aws s3 ls > s3_list.txt
  ```