2.2. Student Handout

AWS CLI for Beginners: Student Handout

Introduction to AWS CLI

AWS CLI (Command Line Interface) is a tool that allows you to manage AWS services directly from your terminal or command prompt. It is useful for automation, scripting, and efficient resource management.

Core AWS CLI Commands

1. Configuring AWS CLI

- **Command**: `aws configure`
- **Purpose**: Set up AWS CLI with your credentials.
- **Prompts**:
- AWS Access Key ID
- AWS Secret Access Key
- Default region name (e.g., `us-east-1`)
- Default output format (e.g., `json`)

2. Getting Help

- **Command**: `aws help`
- **Purpose**: Provides help and documentation for AWS CLI commands.

3. General Command Format

- **Format**: `aws <service> <action>`
- **Example 1**: List all S3 buckets
- `aws s3 ls`
- **Example 2**: Describe EC2 instances
- `aws ec2 describe-instances`
- **Example 3**: List RDS instances
- `aws rds describe-db-instances`

Managing S3 with AWS CLI

1. Creating an S3 Bucket

- **Command**: `aws s3 mb s3://my-bucket-name`
- **Example 1**: `aws s3 mb s3://student-bucket-123`
- **Example 2**: `aws s3 mb s3://project-files-bucket`
- **Example 3**: `aws s3 mb s3://backup-storage-bucket`

2. Uploading Files to S3

- **Command**: `aws s3 cp myfile.txt s3://my-bucket-name/`
- **Example 1**: `aws s3 cp report.pdf s3://student-bucket-123/`
- **Example 2**: `aws s3 cp image.png s3://project-files-bucket/`
- **Example 3**: `aws s3 cp data.csv s3://backup-storage-bucket/`

3. Downloading Files from S3

- **Command**: `aws s3 cp s3://my-bucket-name/myfile.txt ./`
- **Example 1**: `aws s3 cp s3://student-bucket-123/report.pdf ./`
- **Example 2**: `aws s3 cp s3://project-files-bucket/image.png ./`
- **Example 3**: `aws s3 cp s3://backup-storage-bucket/data.csv ./`

4. Setting Permissions

- **Command**: `aws s3api put-object-acl --bucket my-bucket-name --key myfile.txt --acl public-read`
- **Example 1**: Make `report.pdf` public
- **Example 2**: Make `image.png` public
- **Example 3**: Make `data.csv` public

Managing EC2 with AWS CLI

1. Launching an EC2 Instance

- **Command**: `aws ec2 run-instances --image-id ami-12345678 --instance-type t2.micro --key-name my-key-pair --security-group-ids sg-12345678`

- **Example 1**: Launch with `ami-87654321`
- **Example 2**: Launch with `t2.small` instance type
- **Example 3**: Launch with `sg-87654321` security group

2. Terminating an EC2 Instance

- **Command**: `aws ec2 terminate-instances --instance-ids i-12345678`
- **Example 1**: Terminate instance `i-87654321`
- **Example 2**: Terminate instance `i-23456789`
- **Example 3**: Terminate instance `i-34567890`

3. Managing Key Pairs and Security Groups

- **Create Key Pair**: `aws ec2 create-key-pair --key-name my-new-key`
- **Example 1**: `aws ec2 create-key-pair --key-name student-key`
- **Example 2**: `aws ec2 create-key-pair --key-name project-key`
- **Example 3**: `aws ec2 create-key-pair --key-name backup-key`
- **Create Security Group**: `aws ec2 create-security-group --group-name my-security-group --description "My security group"`
- **Example 1**: `aws ec2 create-security-group --group-name student-sg --description "Student security group"`
- **Example 2**: `aws ec2 create-security-group --group-name project-sg --description "Project security group"`
- **Example 3**: `aws ec2 create-security-group --group-name backup-sg --description "Backup security group"`

Managing RDS with AWS CLI

1. Creating an RDS Instance

- **Command**: `aws rds create-db-instance --db-instance-identifier mydbinstance --db-instance-class db.t2.micro --engine mysql --master-username admin --master-user-password password123`
- **Example 1**: `aws rds create-db-instance --db-instance-identifier studentdb --db-instance-class db.t2.small --engine postgres --master-username student --master-user-password studentpass`
- **Example 2**: `aws rds create-db-instance --db-instance-identifier projectdb --db-instance-class db.t3.micro --engine mysql --master-username project --`

`master-user-password projectpass`

- **Example 3**: `aws rds create-db-instance --db-instance-identifier backupdb --db-instance-class db.t2.micro --engine oracle-se2 --master-username backup --master-user-password backuppass`

2. Backing Up and Restoring Databases

- **Create Snapshot**: `aws rds create-db-snapshot --db-snapshot-identifier mydbsnapshot --db-instance-identifier mydbinstance`
- **Example 1**: `aws rds create-db-snapshot --db-snapshot-identifier studentsnapshot --db-instance-identifier studentdb`
- **Example 2**: `aws rds create-db-snapshot --db-snapshot-identifier projectsnapshot --db-instance-identifier projectdb`
- **Example 3**: `aws rds create-db-snapshot --db-snapshot-identifier backupsnapshot --db-instance-identifier backupdb`
- **Restore from Snapshot**: `aws rds restore-db-instance-from-db-snapshot --db-instance-identifier mynewdbinstance --db-snapshot-identifier mydbsnapshot`
- **Example 1**: `aws rds restore-db-instance-from-db-snapshot --db-instance-identifier newstudentdb --db-snapshot-identifier studentsnapshot`
- **Example 2**: `aws rds restore-db-instance-from-db-snapshot --db-instance-identifier newprojectdb --db-snapshot-identifier projectsnapshot`
- **Example 3**: `aws rds restore-db-instance-from-db-snapshot --db-instance-identifier newbackupdb --db-snapshot-identifier backupsnapshot`

Hands-on: Building a Simple AWS System

Step 1: Create an S3 Bucket

- **Command**: `aws s3 mb s3://my-simple-system-bucket`
- **Example**: `aws s3 mb s3://simple-system-bucket`

Step 2: Launch an EC2 Instance

- **Command**: `aws ec2 run-instances --image-id ami-12345678 --instance-type t2.micro --key-name my-key-pair --security-group-ids sg-12345678`
- **Example**: `aws ec2 run-instances --image-id ami-87654321 --instance-type t2.micro --key-name simple-key --security-group-ids sg-87654321`

Step 3: Create an RDS Instance

- **Command**: `aws rds create-db-instance --db-instance-identifier mydbinstance --db-instance-class db.t2.micro --engine mysql --master-username admin --master-user-password password123`
- **Example**: `aws rds create-db-instance --db-instance-identifier simplesystemdb --db-instance-class db.t2.micro --engine mysql --master-username admin --master-user-password adminpass`

Conclusion

This handout provides a concise overview of using AWS CLI to manage AWS services like S3, EC2, and RDS. Practice these commands to gain hands-on experience and deepen your understanding of AWS CLI.