Title: Understand the working of AWS and creating an EC2 instance and a S3 bucket.

1. Objective:

The objective of this cloud lab experiment is to deploy an EC2 instance on AWS to run a basic web server and create an S3 bucket to upload and store files. The experiment aims to understand the configuration and management of compute and storage resources within AWS.

2. Background:

- ### Theory/Concepts:

Virtualization: AWS EC2 uses virtualization to provide scalable compute capacity, allowing multiple virtual servers (instances) to run on a single physical machine.

Cloud Deployment Models:

- Public Cloud: AWS operates as a public cloud, providing scalable resources that can be shared across multiple users.

Cloud Service Models:

- IaaS (Infrastructure as a Service): AWS EC2 is an IaaS offering, providing scalable virtual servers.
- SaaS (Software as a Service): While not directly used in this experiment, SaaS includes applications like AWS Cloud9 that run on AWS infrastructure.

Context: AWS (Amazon Web Services) is used for this experiment. AWS provides a range of cloud services including compute (EC2) and storage (S3) solutions. This lab focuses on configuring EC2 for web hosting and S3 for file storage.

3. Tools and Services

- **Cloud Services:**

  AWS EC2 (Elastic Compute Cloud): Provides resizable compute capacity in the cloud.

  AWS S3 (Simple Storage Service): Offers scalable object storage for data backup and archiving.

- **Software/Tools:**

  AWS CLI (Command Line Interface): A unified tool to manage AWS services via the command line.

  AWS Management Console: A web-based interface to interact with AWS services.

4. Experiment Setup

Step-by-Step Configuration:

Cloud Account Setup:

- Sign an AWS account at [AWS Sign-Up] (https://aws.amazon.com/). Provide necessary information.

Environment Configuration:

Create an EC2 Instance

- Log in to the AWS Management Console.
- Navigate to the EC2 Dashboard and click “Launch Instance.”
- Select an Amazon Machine Image (AMI), such as “Amazon Linux 2 AMI.”

- Choose an instance type (e.g., `t2.micro` for the free tier).
- Configure instance details (default settings are usually sufficient).
- Add storage (default size is typically sufficient).
- Configure a security group to allow HTTP (port 80) and SSH (port 22) access.
- Review and launch the instance. Download the key pair (.pem file) for SSH access.

Set Up S3 Bucket:

- Go to the AWS Management Console and navigate to the S3 service.
- Click “Create Bucket.”
- Enter a unique bucket name and select a region.
- Configure bucket settings, such as public access permissions (for this experiment, enable public access to allow file uploads).
- Review and create the bucket.
- Navigate to your S3 bucket and click “Upload.”
- Choose files from your local machine and click “Upload.”

Security Settings:

Configure Security Groups:

- Go to the EC2 Dashboard and select your instance’s security group.
- Ensure inbound rules allow HTTP (port 80) and SSH (port 22).

IAM Roles:

- If your EC2 instance requires access to other AWS services, create and attach an IAM role with appropriate permissions.

5. Execution

Tasks Performed:

Instance Launch: Launched an EC2 instance, configured it, and set up Apache.
Bucket Configuration: Created an S3 bucket and configured it for file uploads.
File Upload: Uploaded files to the S3 bucket using the AWS Management Console and AWS CLI.

6. Observations

Data Collected:

Performance Metrics:

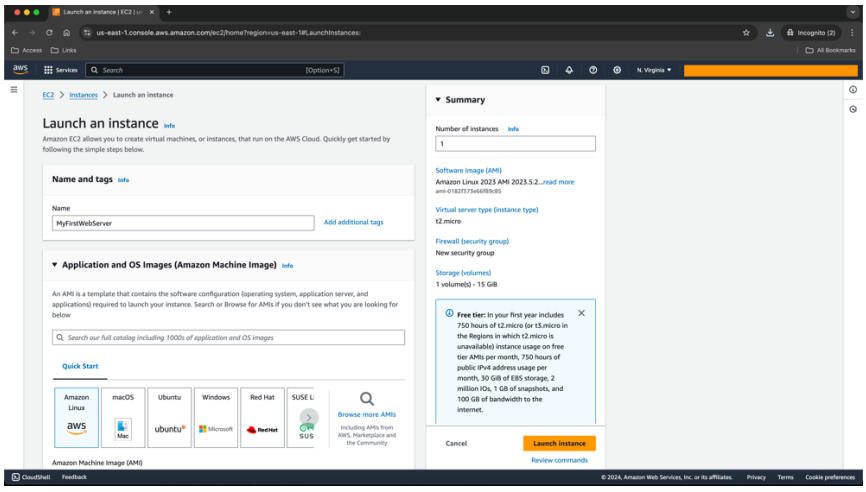
CPU Utilization: Monitored via CloudWatch.

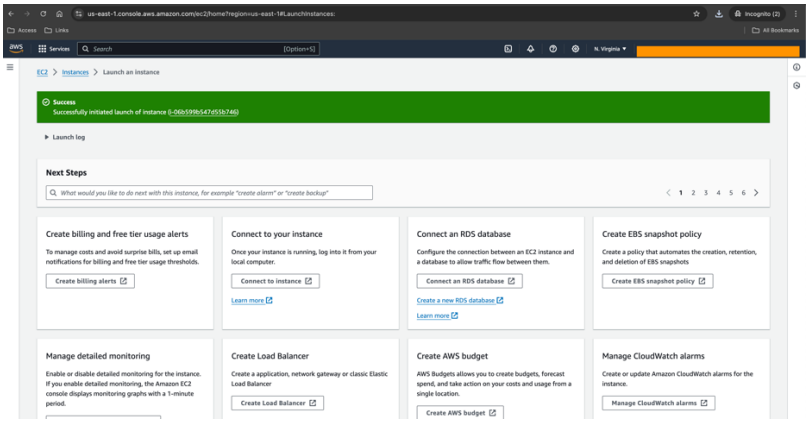
S3 Bucket:

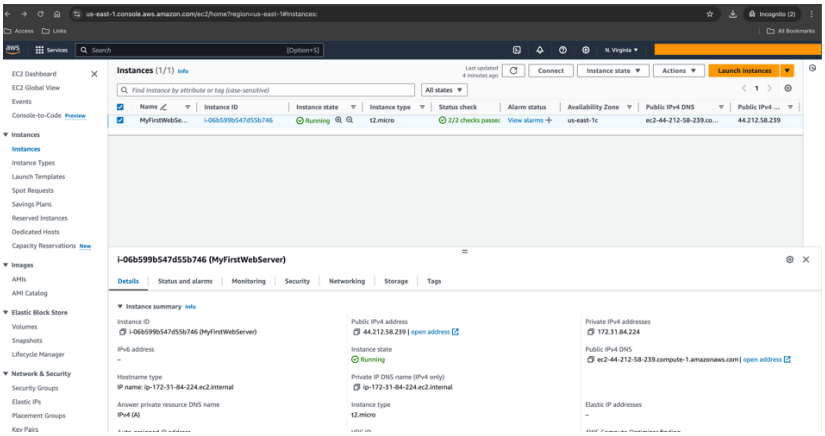
File upload success: Verified uploaded files in the S3 bucket.

Access permissions: Ensured files were publicly accessible if configured.

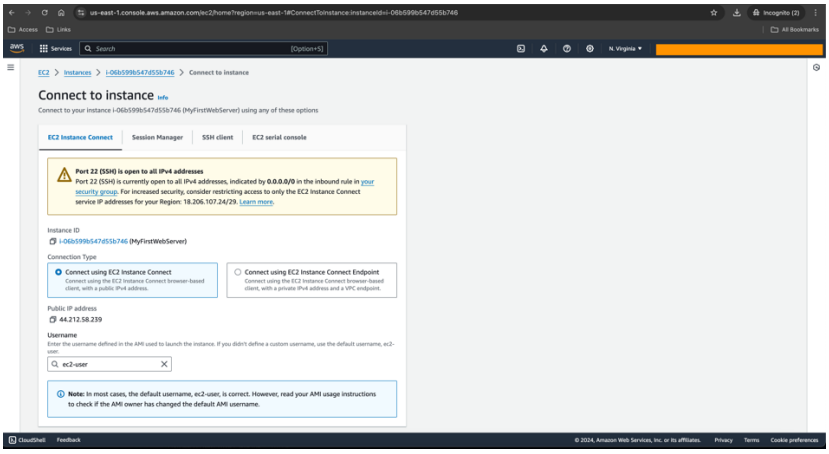
Launching an EC2 Instance

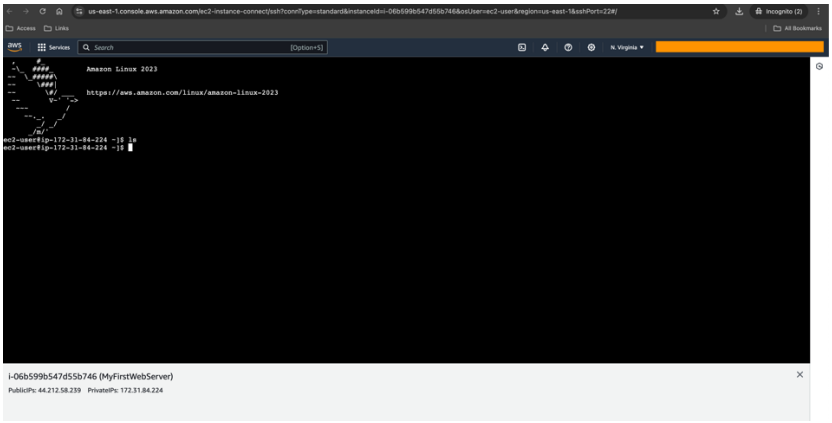




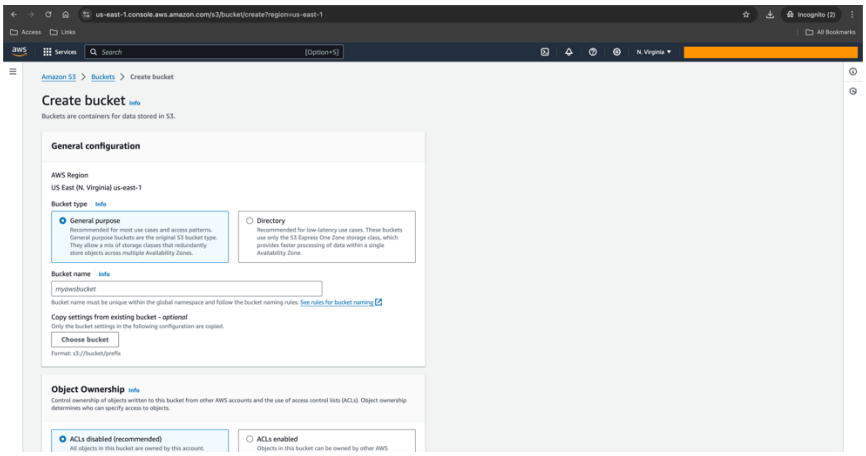


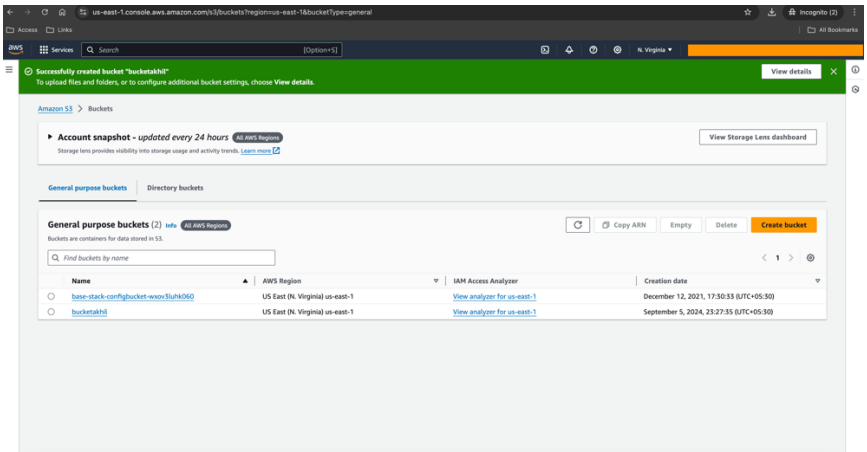
Connecting to an EC2 Instance



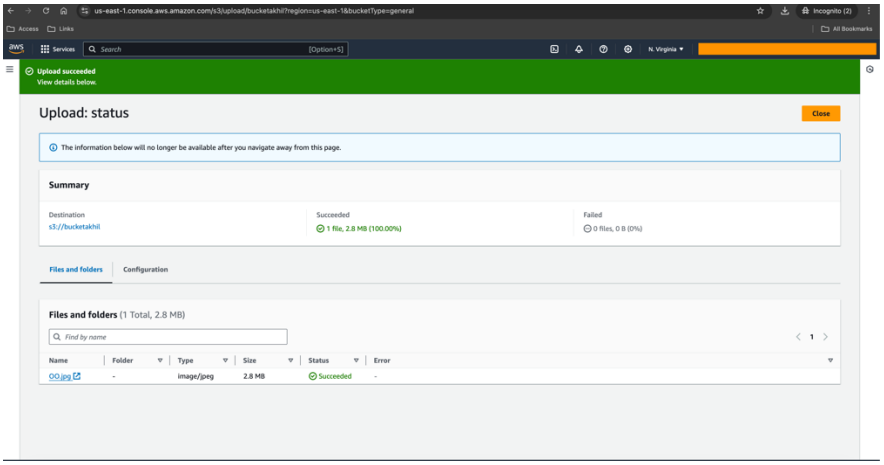


Creating an S3 Bucket





Uploading a file to an S3 Bucket



7. Results

- The EC2 instance was successfully launched. The S3 bucket was created, and files were successfully uploaded and retrieved.