Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up an S3-Compatible Storage Locally: Use MinIO to create a local object storage service. Upload and download files using the web interface or CLI.

Name: Hema S

Department: ECE

Introduction

MinIO is an open-source, high-performance, and S3-compatible object storage solution. It allows users to store and manage data similar to Amazon S3 but runs locally on their own machines. This PoC explores how to install and configure MinIO using Docker, create storage buckets, and perform basic file operations.

Overview

This PoC demonstrates the setup of MinIO on a local system using Docker. The steps include:

1. Installing and running MinIO in a Docker container.
2. Accessing the MinIO web interface.
3. Creating a storage bucket.
4. Uploading and downloading files.
5. Verifying storage operations.

MinIO serves as an excellent alternative to cloud-based object storage solutions, providing a local environment for development, testing, and learning.

Objectives

The primary objectives of this PoC are:

1. To understand and implement an **S3-compatible** object storage solution locally.

2. To configure and manage MinIO using the **Docker container.**

3. To perform **basic storage operations** (creating buckets, uploading, and downloading files).

4. To explore **how MinIO can be used for cloud and DevOps workflows.**

Importance

1. **Hands-on Experience with Object Storage** – MinIO provides a real-world S3-compatible storage experience.

2. **Local Development & Testing** – It eliminates dependency on cloud storage services, reducing costs.

3. **Easy Integration with DevOps Tools** – MinIO is widely used for Kubernetes, CI/CD pipelines, and big data workloads.

4. **Foundation for Cloud & AWS S3** – Understanding MinIO helps in working with AWS S3 and similar cloud-based object storage services.

Step-by-Step Overview

Step 1:

Pull the MinIO Docker Image:

In the Command Prompt, execute:

docker pull minio/minio

This command downloads the latest MinIO image from Docker Hub.

```
C:\Users\DELL>docker pull minio/minio
Using default tag: latest
latest: Pulling from minio/minio
f85b91ff2bfd: Pull complete
e3a2c2426f91: Pull complete
2c2d0fc1bc01: Pull complete
dceacb66a9de: Pull complete
72729072f786: Pull complete
ad57128305dd: Pull complete
f634493814a0: Pull complete
bca14f032577: Pull complete
2390426de7ed: Pull complete
d8d02ca4ca98: Pull complete
Digest: sha256:a929054ae025fa7997857cd0e2a2e3029238e31ad89877326dc032f4c1a14259
Status: Downloaded newer image for minio/minio:latest
docker.io/minio/minio:latest
```

Step 2:

Run the MinIO Container:

After the image is downloaded, start the MinIO container by running:

Command with the **default** MinIO credentials:

**docker run -p 9000:9000 -p 9001:9001 --name minio **
**-e "MINIO_ROOT_USER=minioadmin" **
**-e "MINIO_ROOT_PASSWORD=minioadmin" **
minio/minio server /data --console-address ":9001"

```
C:\Users\DELL>docker run -p 9000:9000 -p 9001:9001 --name minio -e "MINIO_ROOT_USER=minioadmin" -e "MINIO_ROOT_PASSWORD=mini
oadmin" minio/minio server /data --console-address ":9001"
INFO: Formatting 1st pool, 1 set(s), 1 drives per set.
INFO: WARNING: Host local has more than 0 drives of set. A host failure will result in data becoming unavailable.
MinIO Object Storage Server
Copyright: 2015-2025 MinIO, Inc.
License: GNU AGPLv3 - https://www.gnu.org/licenses/agpl-3.0.html
Version: RELEASE.2025-02-28T09-55-16Z (go1.23.6 linux/amd64)

API: http://172.17.0.2:9000  http://127.0.0.1:9000
WebUI: http://172.17.0.2:9001 http://127.0.0.1:9001

Docs: https://docs.min.io
WARN: Detected default credentials 'minioadmin:minioadmin', we recommend that you change these values with 'MINIO_ROOT_USER'
 and 'MINIO_ROOT_PASSWORD' environment variables
```

Step 3:

Access the MinIO Web Interface

1. Open your browser and go to:

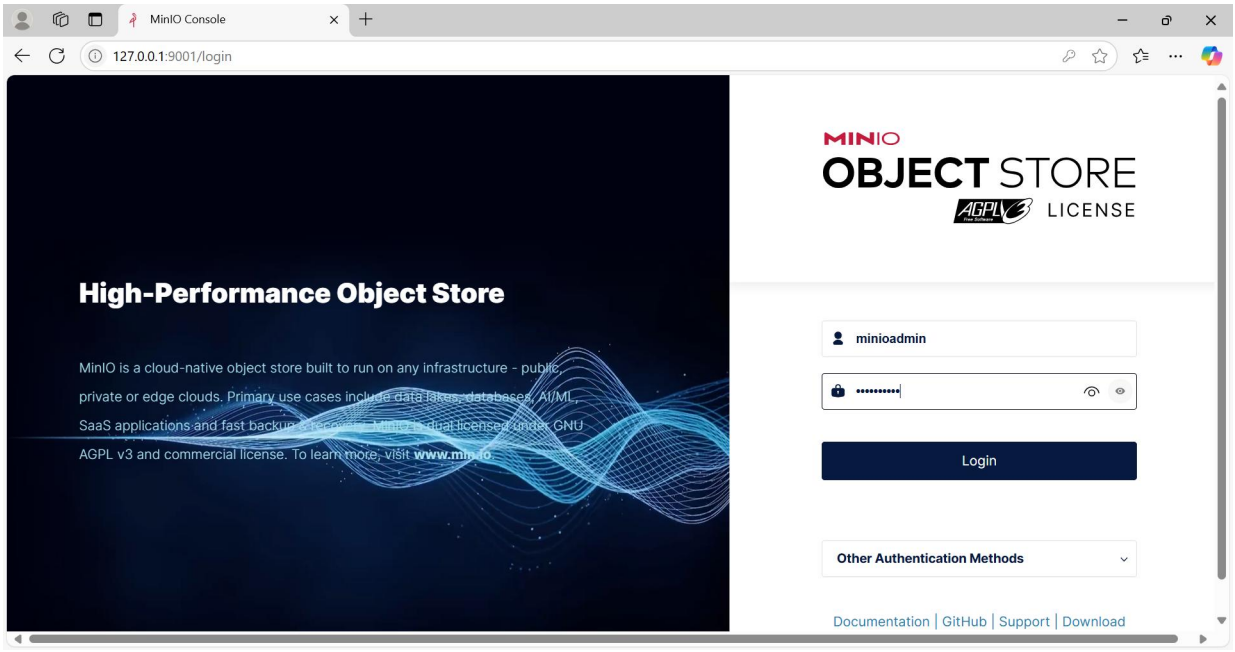
http://127.0.0.1:9001

2. Login with the default credentials:

Access Key: minioadmin

Secret Key: minioadmin

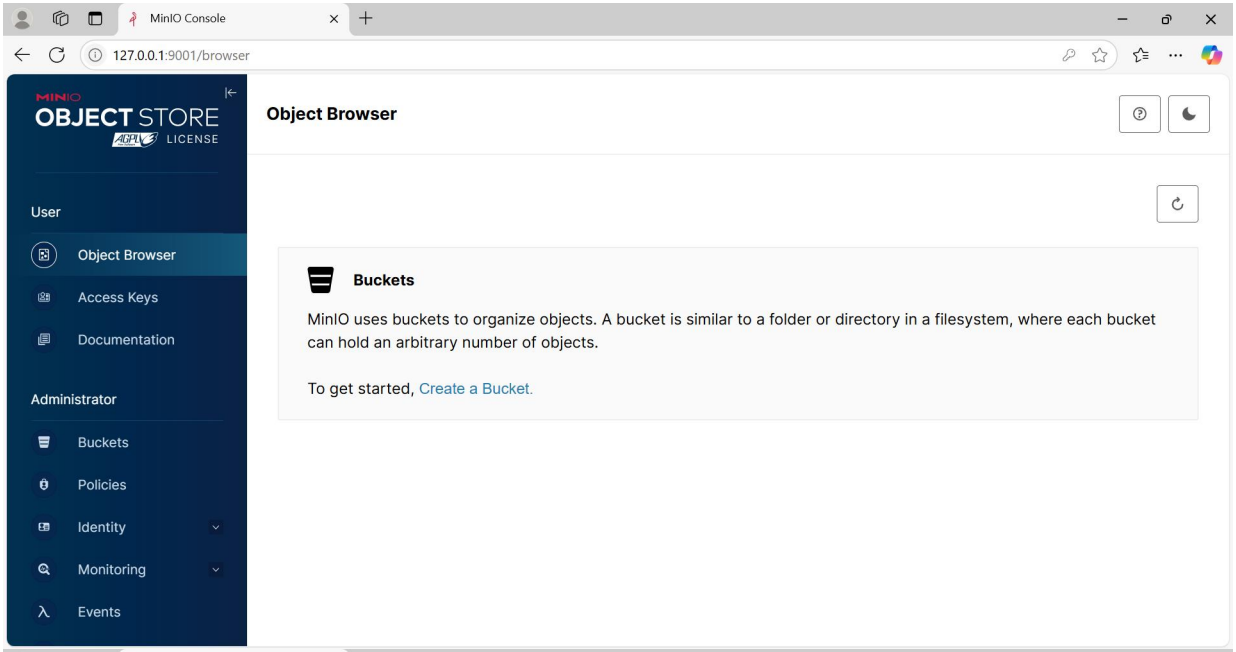
3. Click "Sign In"

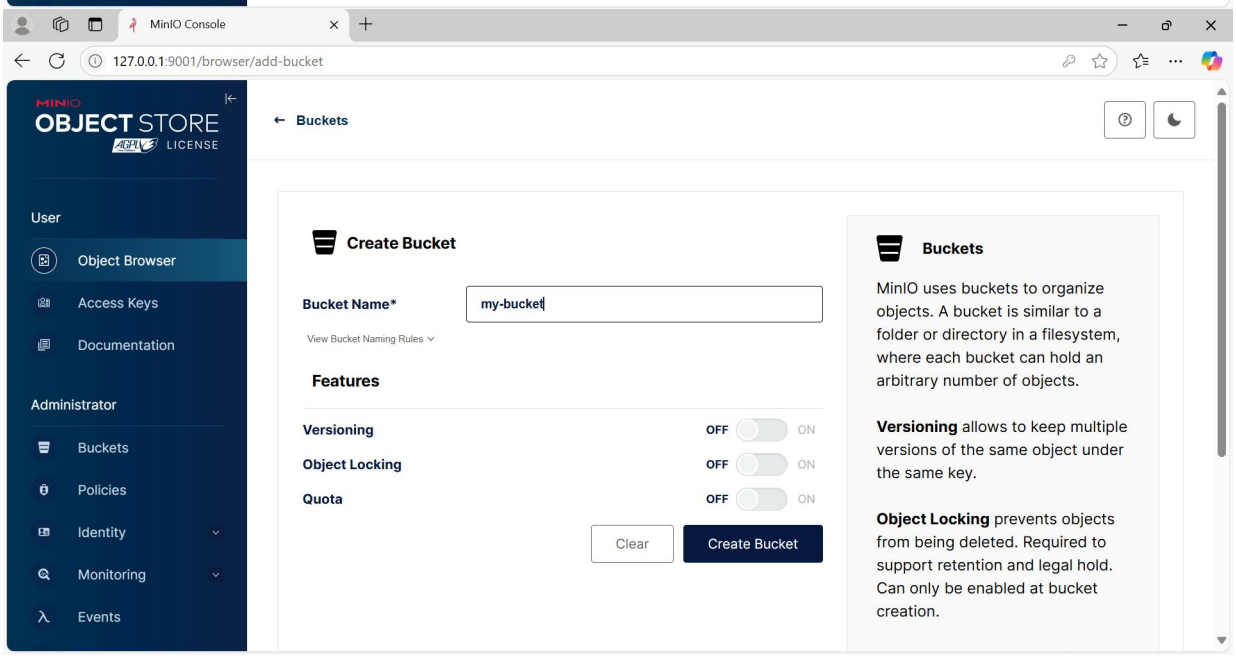


Step 4:

Create a New Bucket

1. On the **MinIO Web Interface**, go to the **Buckets** tab.
2. Click **"Create Bucket"** (top-right).
3. Enter a **bucket name** (e.g., my-bucket).
4. Click **"Create"**

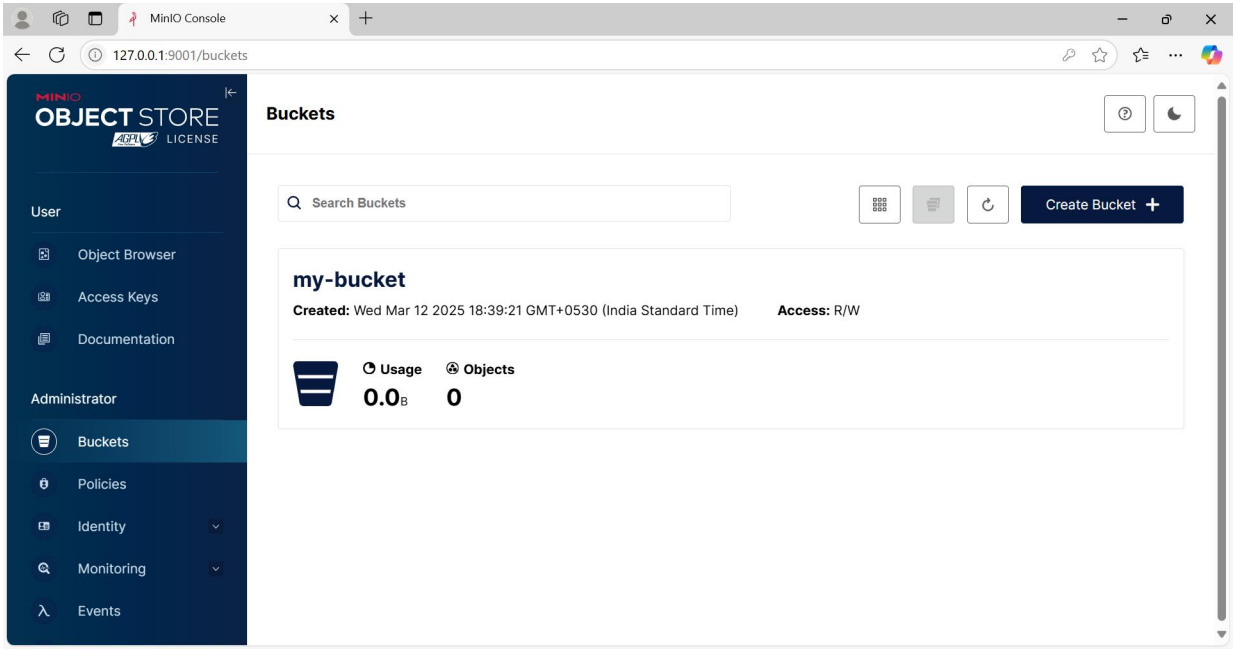


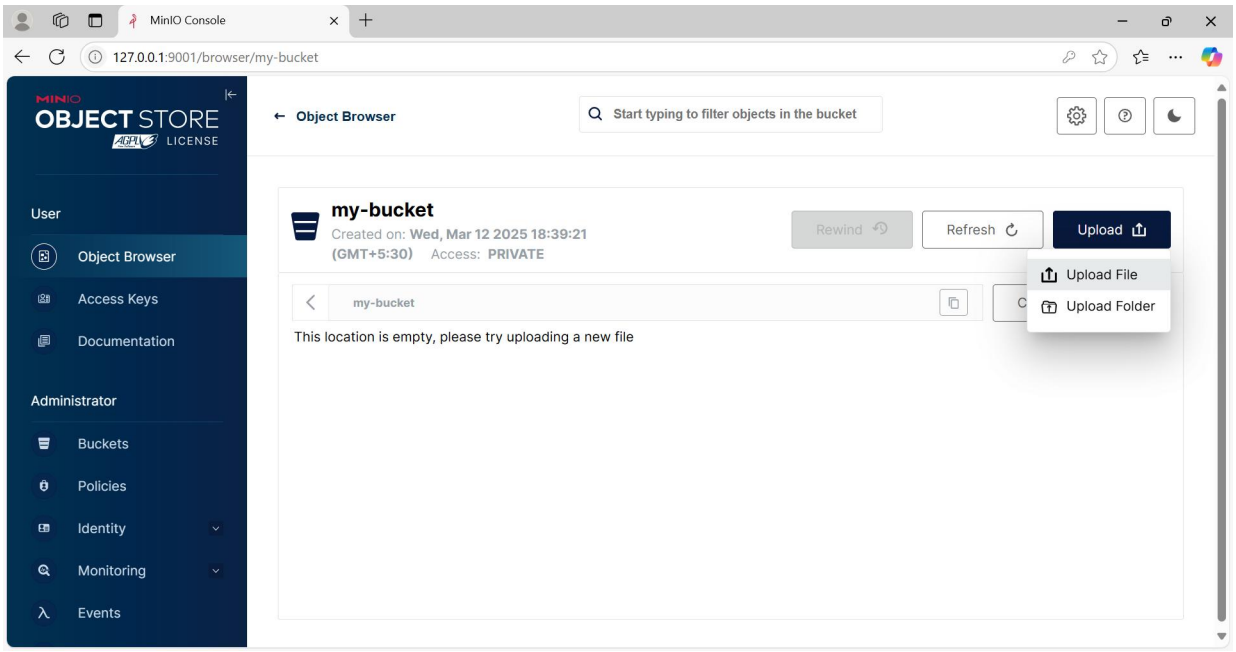


Step 5:

Upload a File

1. Open the newly created bucket (my-bucket).
2. Click **"Upload Files"** and select any file from your computer.
3. The file will now be stored in MinIO.



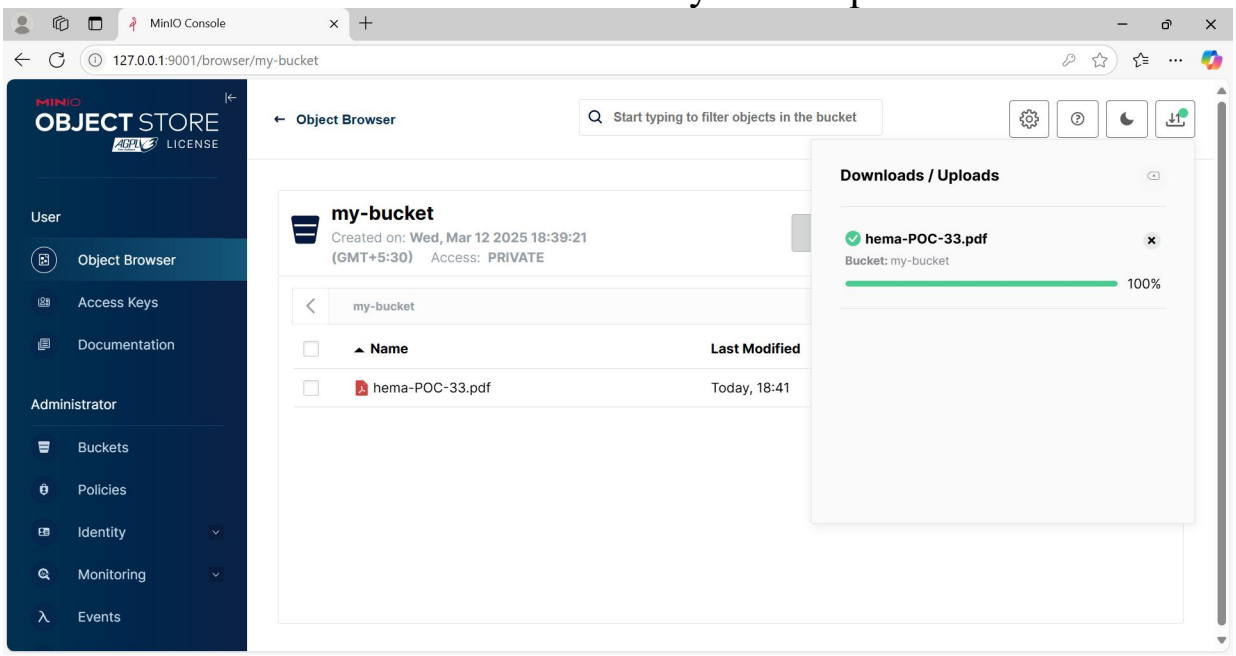


Step 6:

Download a File

1. Click on the uploaded file inside the bucket.

2. Click **"Download"** to save it back to your computer.



Step 7:

1. Press **Ctrl+C** to exit .

2. Stop the Running MinIO Container

docker stop minio

3. Remove the MinIO Container

docker rm minio

4. Remove MinIO Data (Optional)

docker volume prune

```
INFO: Exiting on signal: INTERRUPT

C:\Users\DELL>docker stop minio
minio

C:\Users\DELL>docker rm minio
minio

C:\Users\DELL>docker volume prune
WARNING! This will remove anonymous local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
d7cd32cc69a02ffb06a371186c5b34e45509a82c43b6da501c80405ef40b1a2d

Total reclaimed space: 4.738MB
```

Outcomes

By completing this **MinIO POC**, you will:

1. **Understand S3-Compatible Object Storage** – Gain hands-on experience with MinIO, an open-source alternative to AWS S3, for storing and retrieving objects efficiently.

2. **Deploy and Manage MinIO Using Docker** – Learn how to run MinIO inside a Docker container, exposing the necessary ports and configuring access credentials dynamically.

3. **Access MinIO via Web Interface & API** – Explore MinIO's web UI for managing storage buckets and interact with its API for automation and scripting purposes.

4. **Perform Basic Storage Operations** – Create, upload, download, and delete objects in MinIO buckets using both the web console and command-line tools.

5. **Work with Environment Variables in Docker** – Learn how to configure MinIO dynamically using MINIO_ROOT_USER and MINIO_ROOT_PASSWORD environment variables during container deployment.