

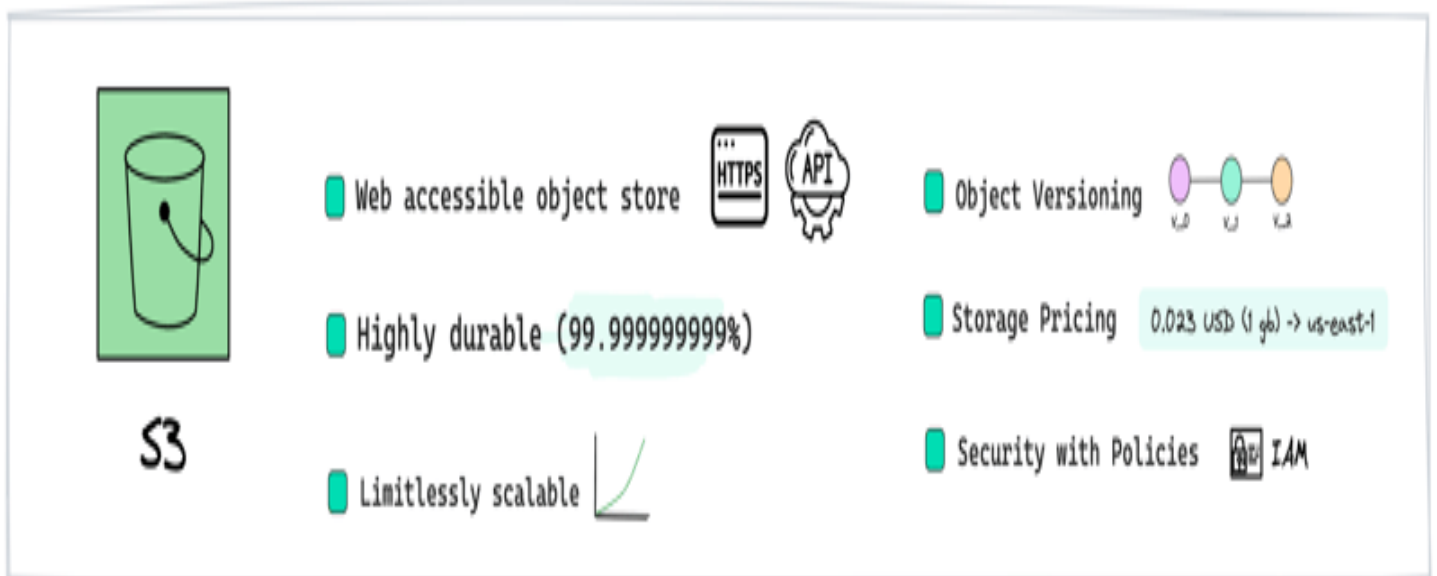
### Working with Cloud Storage Services

This experiment explores setting up an AWS S3 storage bucket, uploading data, verifying its successful storage, and downloading it. By utilizing cloud storage, data management becomes scalable, secure, and easily accessible. This guide walks through configuring an S3 bucket, performing data transfers, and ensuring successful retrieval.

#### What is AWS S3?

Amazon S3 (Simple Storage Service) is an object storage service that offers scalability, security, and high availability. It allows users to store and retrieve any amount of data at any time, making it ideal for backups, website hosting, and data lakes.

- Secure and highly available storage solution
- Supports various storage classes for cost optimization
- Used for website hosting, backup, and disaster recovery
- Provides strong access control and encryption options
- Integrates seamlessly with other AWS services



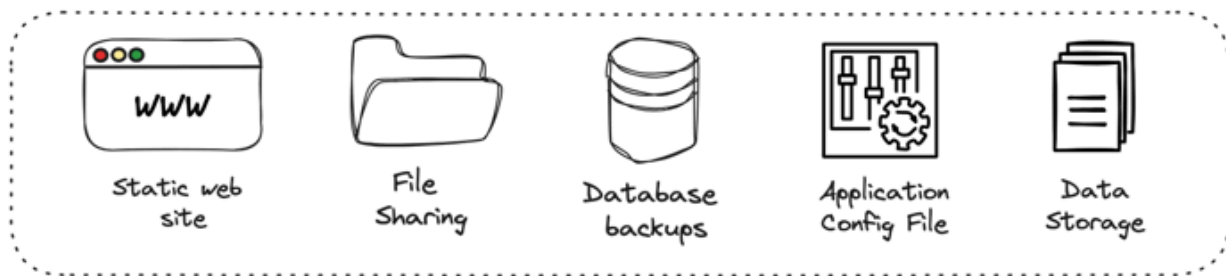
#### Use Cases for S3

Amazon S3 is widely used across industries for a variety of purposes, including storing media files, hosting websites, and serving as a data lake for big data analytics.

- Storing images, videos, and static website assets

- Hosting static websites and content delivery
- Backup and disaster recovery solutions
- Log storage for analytics and monitoring
- Machine learning and big data storage







### Multiple use cases



## S3 Storage Types

Amazon S3 provides different storage classes to meet various needs, balancing cost and performance. Each class is designed for a specific use case, from frequent access to long-term archival.

- **S3 Standard**: High durability and performance for frequently accessed data
- **S3 Intelligent-Tiering**: Automatically moves data to lower-cost tiers when not accessed
- **S3 Standard-IA (Infrequent Access)**: Lower cost for data that is accessed less frequently
- **S3 Glacier & Glacier Deep Archive**: Cost-effective long-term archival storage
- **S3 One Zone-IA**: Lower-cost storage for infrequently accessed data in a single availability zone

Storage Type		
Milliseconds Access		Standard
		Intelligent Tiering
		Standard-IA
		One Zone-IA
Minutes or hours		Glacier
12 or 48 Hours		Glacier Deep Archive
		Infrequent

# Step-by-Step Guide

---

## Step 1: Start LocalStack

Run the following command to start LocalStack:

```
localstack start
```

Alternatively, use Docker:

```
docker run --rm -it -p 4566:4566 localstack/localstack
```

### Start Docker Desktop

- Launch Docker Desktop and wait until it indicates that "Docker is running."
- LocalStack will simulate AWS services on port 4566 , allowing local cloud development without an actual AWS account.

## Step 2: Creating an S3 Bucket in LocalStack

### Commands Breakdown:

#### 1. List Existing Buckets:

```
aws --endpoint-url=http://localhost:4566 --region us-east-1 s3 ls
```

- **Purpose:** This command is used to list the existing S3 buckets in your LocalStack environment, running at localhost:4566 .
- **Output:** No output is shown initially, meaning there are no buckets created at that point.

#### 2. Create a New S3 Bucket:

```
aws --endpoint-url=http://localhost:4566 --region  
us-east-1 s3 mb s3://my-test-bucket
```

- **Purpose:** This command creates a new S3 bucket named my-test-bucket in the LocalStack environment.
- **Output:** make\_bucket: my-test-bucket — This confirms the successful creation of the my-test-bucket .

#### 3. List Buckets Again:

```
aws --endpoint-url=http://localhost:4566 --region us-east-1 s3 ls
```

- **Purpose:** This command lists the existing S3 buckets again, this time after creating the new bucket.
- **Output:**

```
2025-01-23 13:09:41 my-test-bucket
```

— This confirms that the `my-test-bucket` is now listed and was successfully created.

## Detailed Explanation:

### 1. LocalStack's Purpose:

- LocalStack emulates AWS services locally, providing an environment for developers to simulate AWS without needing an actual AWS account or incurring costs. It mimics APIs for various services such as S3, EC2, Lambda, etc., making it a useful tool for development and testing.

### 2. CLI Commands and Output:

- The AWS CLI interacts directly with LocalStack through the endpoint `localhost:4566`, which is where the LocalStack service is running. Every time you run commands like `aws s3 mb` or `aws s3 ls`, LocalStack processes these commands as if they were sent to the actual AWS cloud.
- After running the `aws s3 mb` command, LocalStack creates the `my-test-bucket` bucket in its emulated environment. The successful creation is confirmed by the output: `make_bucket: my-test-bucket`.
- When you list the buckets with `aws s3 ls`, it shows the newly created bucket as `my-test-bucket`, confirming the operation was successful.

### 3. Why the `localhost:4566` Page Remains Empty:

- `localhost:4566` is not intended to serve a graphical user interface (GUI) like the AWS Management Console. Instead, it acts as an endpoint for API requests from tools like the AWS CLI or SDKs.
- LocalStack emulates AWS services by providing HTTP-based API endpoints. These APIs handle requests and respond accordingly, but LocalStack does not serve a web-based dashboard. Therefore, accessing `localhost:4566` via a browser will result in a blank or empty page, which is completely normal.
- To interact with LocalStack, users should rely on the AWS CLI, SDKs, or other infrastructure management tools (like Terraform). These tools send API calls to LocalStack, which then

processes and responds with the requested information or performs actions such as creating S3 buckets.

#### 4. Why the S3 Bucket Is Created:

- The `aws s3 mb` command in LocalStack simulates the creation of an S3 bucket, just as it would in AWS. In your case, the bucket `my-test-bucket` was created locally within LocalStack.
- Even though LocalStack is emulating AWS S3 locally, the functionality is designed to be very similar to the actual AWS S3 service. Therefore, the `aws s3 ls` command shows the bucket because LocalStack has processed the request successfully and maintains an internal record of the bucket.

#### 5. Key Takeaways:

- **LocalStack is functional:** The successful creation and listing of the `my-test-bucket` confirm that LocalStack is operating correctly and the AWS CLI is interacting with it as expected.
- **No graphical interface is provided at `localhost:4566`:** LocalStack is designed to work through API calls, not through a web dashboard.
- **Use CLI for interaction:** For operations like creating buckets, listing them, or interacting with other AWS services, the AWS CLI or other tools should be used to send requests to LocalStack's API endpoints.

## Step 3: Steps to Store and Access Images & CSV in S3 (LocalStack)

### 1. Upload Files to S3 Bucket

Navigate to the directory where your images and CSV file are located:

```
cd "C:\Users\rawat\Documents\8 SEMESTER\
Cloud Computing\Lab\Experiment 3\Items"
```

Run the following command to upload all files to the S3 bucket:

```
aws --endpoint-url=http://localhost:4566 --region
us-east-1 s3 cp . s3://my-test-bucket/ --recursive
```

- **Purpose:** This uploads all files (images & CSV) from your local directory to `my-test-bucket`.
- **Output:** Each file's upload confirmation.

### 2. List Files in S3 Bucket

To confirm that the files were uploaded successfully, list them:

```
aws --endpoint-url=http://localhost:4566 --region  
us-east-1 s3 ls s3://my-test-bucket/
```

- **Expected Output:**

```
2025-02-18 10:30:00    12345 image1.jpg  
2025-02-18 10:30:01    67890 image2.png  
2025-02-18 10:30:02     45678 data.csv
```

### 3. Access & Download Files

- **Download CSV File:**

```
aws --endpoint-url=http://localhost:4566 --region  
us-east-1 s3 cp s3://my-test-bucket/data.csv .
```

- **Purpose:** Retrieves `data.csv` from S3 to your local machine.
- **Output:** download: `s3://my-test-bucket/data.csv` to `./data.csv`

- **Download Images:**

```
aws --endpoint-url=http://localhost:4566 --region us-east-1  
s3 cp s3://my-test-bucket/image1.jpg .  
aws --endpoint-url=http://localhost:4566 --region  
us-east-1 s3 cp s3://my-test-bucket/image2.png .
```

- **Purpose:** Downloads the images locally for viewing.

### Download Directly to a Specific Path

To download the file directly to `C:\Users\rawat\Downloads\`, use the following command:

```
aws --endpoint-url=http://localhost:4566 --region us-east-1 s3  
cp s3://my-test-bucket/Sample_Housing_CSV_File.csv  
"C:\Users\rawat\Downloads\Sample_Housing_CSV_File.csv"
```

### Explanation:

- `aws s3 cp` → Copies the file from S3 to your local machine.
- `s3://my-test-bucket/Sample_Housing_CSV_File.csv` → Specifies the file location in the S3 bucket.
- `C:\Users\rawat\Downloads\Sample_Housing_CSV_File.csv` → Specifies the target directory for the downloaded file.

## Verify Download

After running the command, check your `Downloads` folder. To confirm in Command Prompt, run:

```
dir "C:\Users\rawat\Downloads\Sample_Housing_CSV_File.csv"
```

## 4. Print CSV Content in Command Line

After downloading the CSV, print its content using:

```
type data.csv
```

OR (For Linux/macOS):

```
cat data.csv
```

- **Expected Output (Example):**

```
Name, Age, City
Alice, 25, New Delhi
Bob, 30, Mumbai
```

## 5. Open Images for Viewing

For Windows:

```
start image1.jpg
start image2.png
```

For macOS:

```
open image1.jpg
open image2.png
```

For Linux (if you have an image viewer installed):

```
xdg-open image1.jpg
xdg-open image2.png
```

## Useful Resources for Amazon S3

---

## 1. AWS S3 Documentation

- Official AWS guide covering S3 concepts, APIs, and best practices.
- [Read here](#)

## 2. AWS S3 CLI Reference

- Detailed command-line instructions for managing S3 buckets and objects.
- [Read here](#)

## 3. AWS S3 Security Best Practices

- A guide on securing your S3 buckets and preventing unauthorized access.
- [Read here](#)

## 4. LocalStack - AWS Services on Local Machine

- Learn how to simulate AWS S3 locally using LocalStack.
- [Read here](#)

## 5. AWS S3 Pricing & Free Tier Details

- Understand S3 pricing models and how to optimize costs.
- [Read here](#)