

S3 Storage Fundamentals Lab

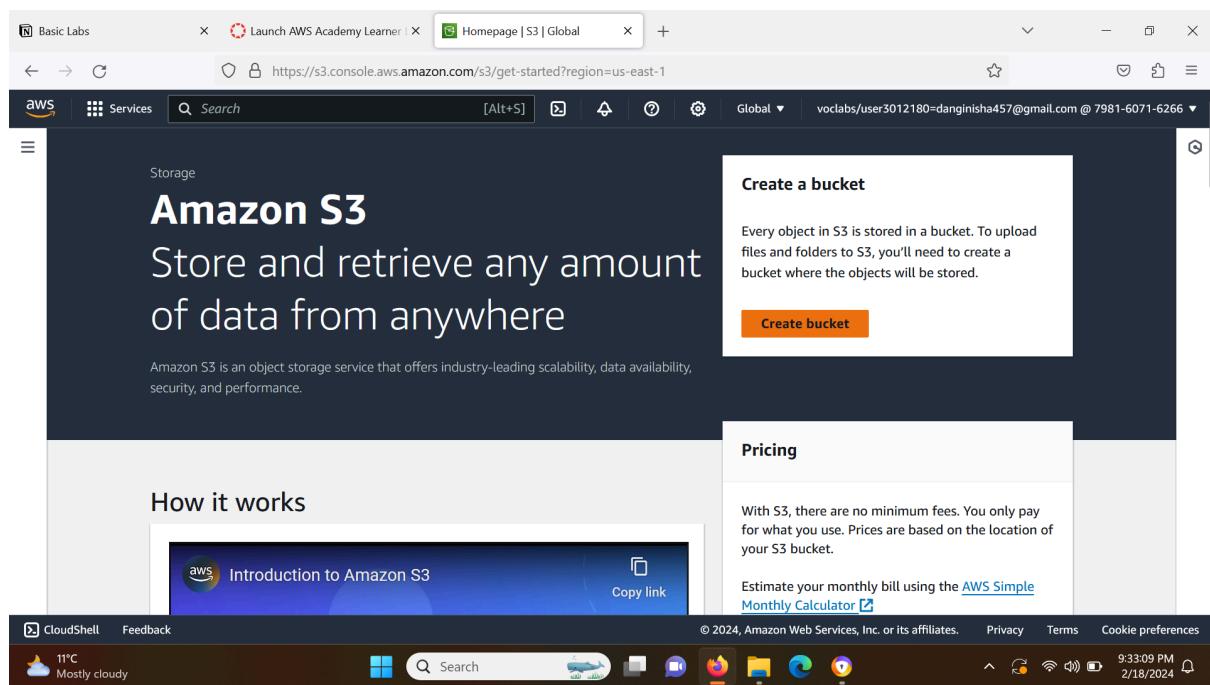
Objective: To gain hands-on experience with Amazon S3 by performing basic storage operations.

Approach: This lab involves creating an S3 bucket, uploading files to it, and setting up bucket policies for access control. Students will explore the S3 management console, learn about object storage, and understand the concepts of buckets and objects.

Goal: Students will understand how to use S3 for storing and managing data, learn about S3 security and permissions, and become familiar with S3's user interface.

Solution:

Step 1: We start by creating a bucket



Step 2: We give a unique name to the bucket.

The screenshot shows the 'Create S3 bucket' wizard on the AWS console. In the 'Bucket type' section, the 'General purpose' option is selected. The 'Bucket name' field contains 'newbucketbootcamp'. Below the name field, a note states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)'.

Step 3: Then, tick the checkbox to block all the public access.

The screenshot shows the 'Create S3 bucket' wizard on the AWS console. In the 'Block Public Access settings for this bucket' section, the 'Block all public access' checkbox is checked. A note below it states: 'Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.' The other three settings are collapsed.

Enable bucket versioning and bucket key. Then click on “Create Bucket” button.

The screenshot shows the 'Bucket Versioning' section of the AWS S3 Bucket creation wizard. It includes a descriptive text about versioning, a note that it's optional, and two radio button options: 'Disable' (unchecked) and 'Enable' (checked). Below this is a 'Tags - optional (0)' section with a note about using tags for costs and organization, and a link to learn more. A 'No tags associated with this bucket.' message is displayed, along with a 'Add tag' button. The top navigation bar shows the URL as <https://s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general>. The bottom navigation bar includes CloudShell, Feedback, and various system icons.

The screenshot shows the 'Encryption type' and 'Advanced settings' sections of the AWS S3 Bucket creation wizard. Under 'Encryption type', three options are listed: 'Server-side encryption with Amazon S3 managed keys (SSE-S3)' (selected), 'Server-side encryption with AWS Key Management Service keys (SSE-KMS)' (unchecked), and 'Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)' (unchecked). A note below DSSE-KMS states: 'Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#)'. Under 'Bucket Key', it notes that SSE-KMS reduces encryption costs by lowering calls to AWS KMS. It also states that S3 Bucket Keys aren't supported for DSSE-KMS and provides a link to learn more. Two radio button options are shown: 'Disable' (unchecked) and 'Enable' (checked). Below these sections is a note: 'After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.' At the bottom right are 'Cancel' and 'Create bucket' buttons. The top navigation bar shows the URL as <https://s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general>. The bottom navigation bar includes CloudShell, Feedback, and various system icons.

We can see our bucket is created.

The screenshot shows the AWS S3 console with a green success message at the top: "Successfully created bucket 'newawsbucketbootcamp'". Below it, a note says: "To upload files and folders, or to configure additional bucket settings, choose View details." A link to "View details" is shown. The main table lists one bucket:

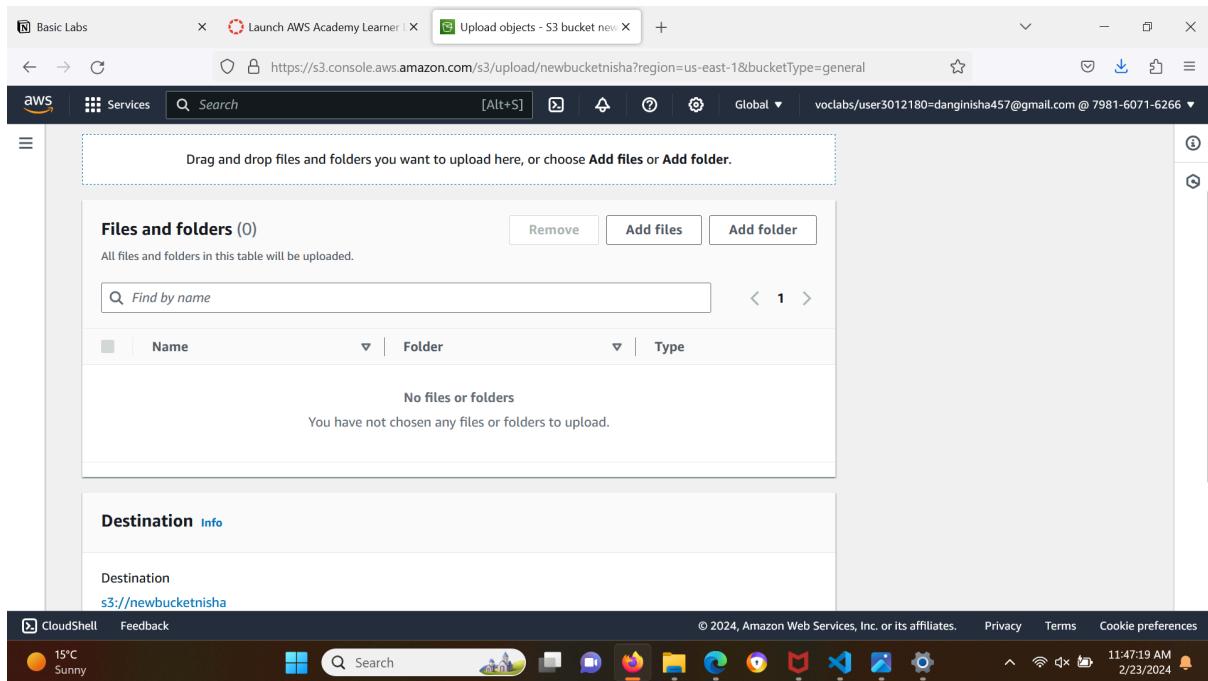
Name	AWS Region	Access	Creation date
newawsbucketbootcamp	US East (N. Virginia) us-east-1	Bucket and objects not public	February 23, 2024, 11:41:02 (UTC+05:45)

Step 4: Now, we can upload files here.

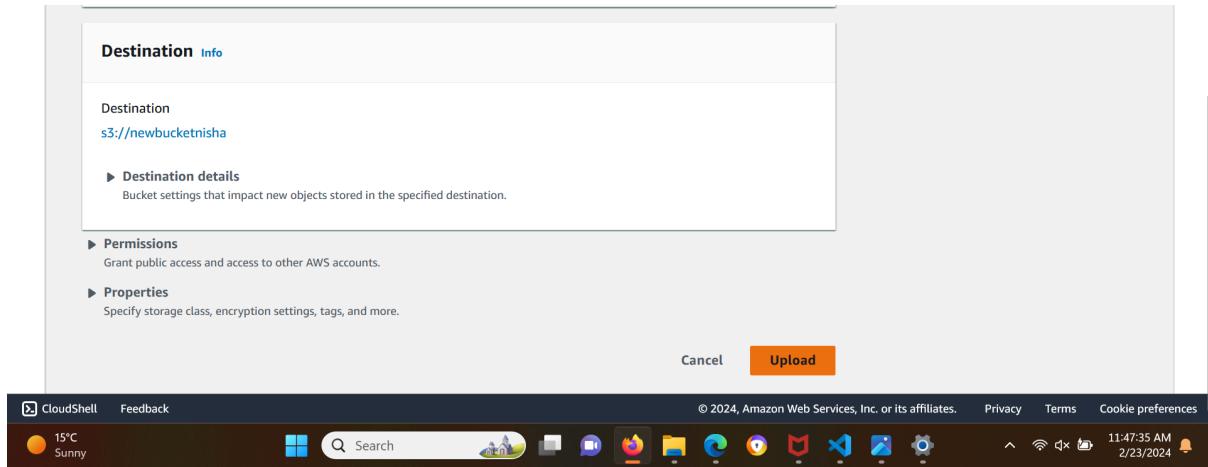
Click the “Upload” button.

The screenshot shows the AWS S3 Objects page for the 'newawsbucketbootcamp' bucket. At the top, there's a toolbar with buttons for Actions, Copy S3 URI, Copy URL, Download, Open, Delete, Create folder, and Upload. The 'Upload' button is highlighted. Below the toolbar, a message states: "Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions." A note below says: "No objects" and "You don't have any objects in this bucket." A large "Upload" button is centered at the bottom of the list area.

Then click on “Add files”



Then, click on the “Upload” button.



Step 5: Then, we can edit the bucket settings. Here, I have blocked public access.

The screenshot shows the AWS S3 Bucket Settings page for a bucket named 'newbucketnisha'. In the 'Block public access (bucket settings)' section, the 'Block all public access' switch is set to 'Off'. Below this, there is a note about individual Block Public Access settings for the bucket. The 'Bucket policy' section is also visible, showing a JSON policy block. The browser interface includes tabs for 'Basic Labs', 'Launch AWS Academy Learner', and 'newbucketnisha - S3 bucket'. The address bar shows the URL: <https://s3.console.aws.amazon.com/s3/buckets/newbucketnisha?region=us-east-1&bucketType=general&tab=perm>. The bottom of the screen shows the Windows taskbar with various pinned icons and the system clock indicating 12:10:36 PM on 2/23/2024.

Step 6: This code allows uploading files to a specific S3 bucket.

The screenshot shows the AWS S3 Bucket Settings page for the same bucket. In the 'Bucket policy' section, a JSON policy document is displayed. The policy allows public read access to objects in the bucket. The policy document is as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::newbucketnisha/*"
    }
  ]
}
```

The browser interface and taskbar are identical to the previous screenshot, showing the same tabs and system information.

Step 7: We can see object details over here. Using this URL, we can view the file uploaded.

The screenshot shows the AWS S3 console interface. A modal window displays detailed information about an uploaded file named "van-gogh-starry-night-min.jpg". The modal includes fields for Owner (awsalbsc0w6977799t1703191035), AWS Region (US East (N. Virginia) us-east-1), Last modified (February 23, 2024, 11:54:23 (UTC+05:45)), Size (93.7 KB), Type (jpg), and Key (van-gogh-starry-night-min.jpg). On the right side of the modal, there are links for S3 URI (s3://newbucketnisha/van-gogh-starry-night-min.jpg), Amazon Resource Name (ARN) (arn:aws:s3:::newbucketnisha/van-gogh-starry-night-min.jpg), Entity tag (Etag) (0626c9a641d0e8147c998dba90f2c8c1), and a green button labeled "Object URL Copied". Below the modal, a message says "Object management overview" and "The following bucket properties and object management configurations impact the behavior of this object." The browser's status bar at the bottom shows the URL https://newbucketnisha.s3.amazonaws.com/van-gogh-starry-night-min.jpg.

Here's the file.

The screenshot shows a web browser window displaying the image "van-gogh-starry-night-min.jpg" from the S3 bucket. The image is a reproduction of Vincent van Gogh's "Starry Night" painting, featuring a dark, swirling sky filled with yellow stars and a prominent, dark, cypress-topped hill in the foreground. The browser's status bar at the bottom shows the URL https://newbucketnisha.s3.amazonaws.com/van-gogh-starry-night-min.jpg.