

ACCESS S3 OBJECTS FROM EC2 INSTANCE

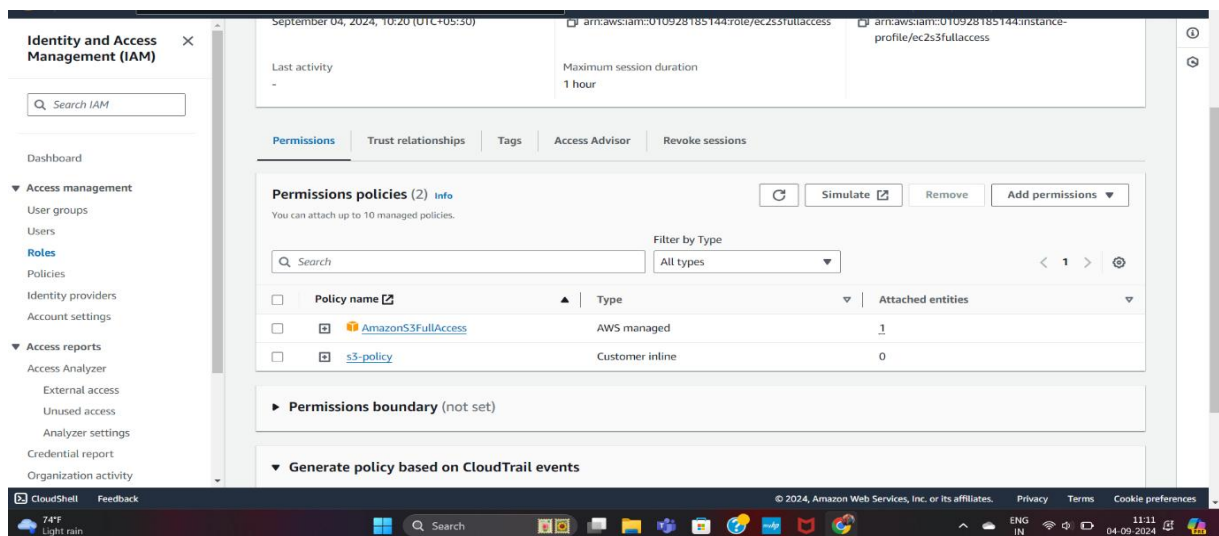
➤ Create an IAM instance profile that grants access to Amazon S3

Complete the following steps:

- Open the **AWS Identity and Access Management (IAM)** console.
- In the navigation pane, under **Access management**, choose **Roles**.
- Choose Create role.
- Under Trusted entity type, choose **AWS service**, and then choose **EC2**.
- Choose Next.
- Create a **custom policy** that provides the minimum required permissions to access your **S3 bucket**.

Note: It's a security best practice to create a policy with the minimum required permissions. However, to allow EC2 access to all your S3 buckets, use the **AmazonS3ReadOnlyAccess** or **AmazonS3FullAccess** managed IAM policy.

- Choose Next.
- Enter a role name, and then choose Create role.



➤ CREATE AMAZON S3 BUCKET

Complete the following steps:

- Open the **Amazon S3**
- Go to the buckets – click on **create bucket**
- **Enabling VERSIONING** is mandatory – create bucket
- After the creation of bucket – click on it – click on upload – add files- upload the files - close

The screenshot shows the 'Create bucket' page in the Amazon S3 console. The breadcrumb navigation at the top reads 'Amazon S3 > Buckets > Create bucket'. The main heading is 'Create bucket' with an 'Info' link. Below this, a note states 'Buckets are containers for data stored in S3.' The 'General configuration' section is active, showing the 'AWS Region' as 'US East (N. Virginia) us-east-1'. Under 'Bucket type', the 'General purpose' option is selected with a radio button, while 'Directory' is unselected. The 'General purpose' description states it is recommended for most use cases and allows a mix of storage classes. The 'Directory' description states it is recommended for low-latency use cases and uses only the S3 Express One Zone storage class. The 'Bucket name' field contains 'swathi-bucket-1' and has an 'Info' link. A note below the field states that the bucket name must be unique and follow naming rules, with a link to 'See rules for bucket naming'. There is a section for 'Copy settings from existing bucket - optional' with a 'Choose bucket' button and a note that only the bucket settings in the configuration are copied. The format 's3://bucket/prefix' is shown. The 'Object Ownership' section is partially visible at the bottom.

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type [Info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

swathi-bucket-1

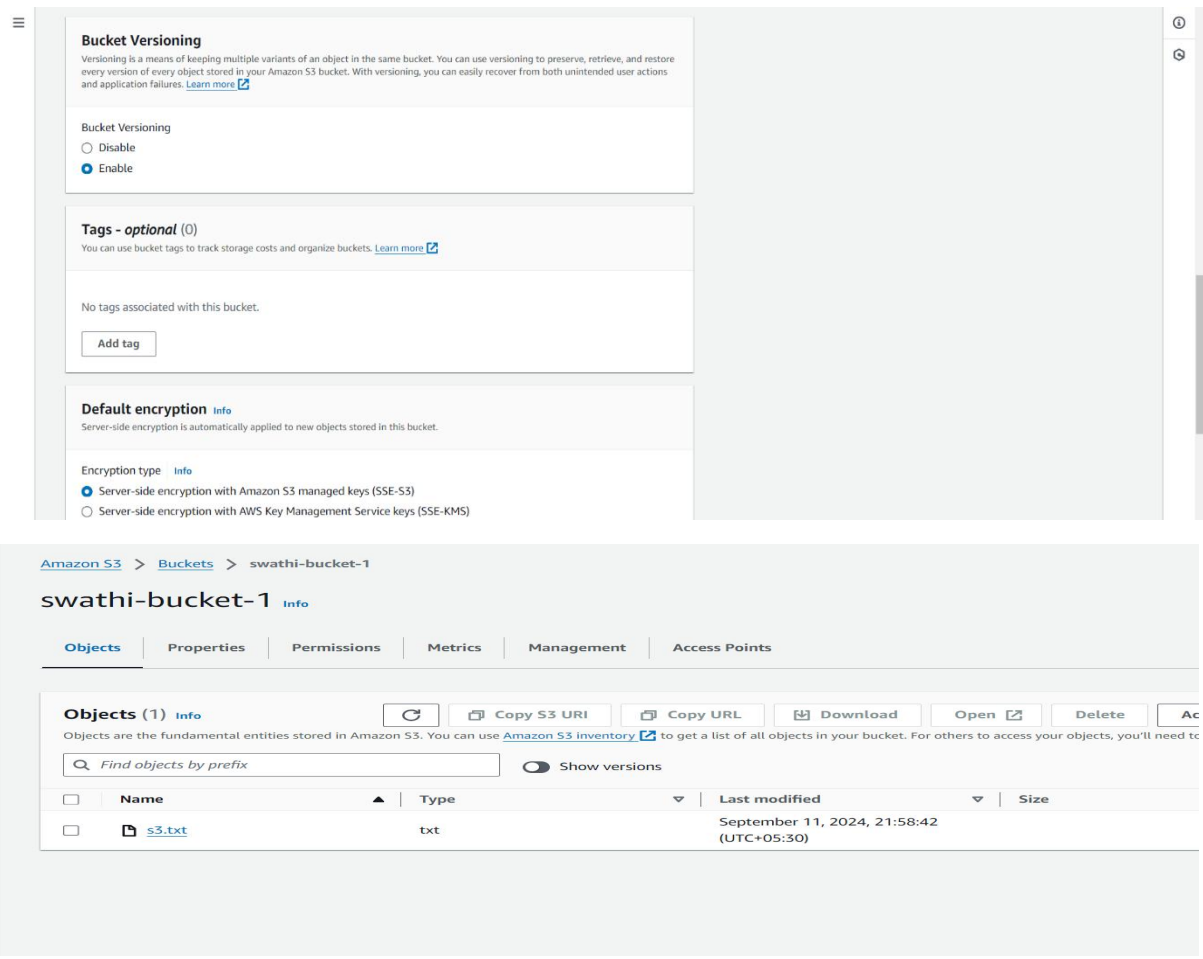
Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Object Ownership [Info](#)



➤ CREATE EC2 INSTANCE & Attach the IAM ROLE TO THE EC2 INSTANCE

Complete the following steps:

- Open the **Amazon EC2** console.
- In the navigation pane, choose **Instances**.
- Select the instance that you want to **attach the IAM role** to.
- Choose the Actions tab, and then choose Security.
- Choose Modify **IAM role**.
- Select the IAM role, and then choose Save. The IAM role is assigned to your EC2 instance

The screenshot shows the AWS Management Console interface. The top navigation bar includes 'EC2 Dashboard', 'EC2 Global View', and 'Events'. The left sidebar lists various services: 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity', 'Reservations', 'Images', 'AMIs', 'AMI Catalog', 'Elastic Block Store', 'Volumes', 'Snapshots', and 'Lifecycle Manager'. The main content area displays a table of EC2 instances. Two instances are listed: 'web1' (ID: i-0849fba3374b5efca) and 'web2' (ID: i-02163b27b4ba0ceec), both in a 'Running' state. A context menu is open for instance 'web1', showing options like 'Connect', 'View details', 'Manage instance state', 'Instance settings', 'Networking', 'Security', 'Image and templates', and 'Monitor and troubleshoot'. The 'Modify IAM role' option is highlighted. Below the table, the details for instance 'web1' are shown, including its public IPv4 address (54.151.23.150), private IPv4 address (172.31.0.222), and instance state (Running). The bottom section shows the 'Modify IAM role' dialog for instance 'i-0ce91e208fc8cae03'. It prompts the user to select an IAM role to attach to the instance. A search bar is present, and a list of roles is shown, with 'ec2-s3fullaccess' selected and marked with a checkmark. The dialog also includes 'Cancel' and 'Update IAM role' buttons.

➤ VERIFY ACCESS TO THE S3 BUCKET

Complete the following steps:

Note: If you receive errors when you run AWS Command Line Interface (AWS CLI) commands, then see [Troubleshoot AWS CLI errors](#). Also, make sure that you're using the most recent AWS CLI version.

- Install the **AWS CLI** on your EC2 instance.
- Run the following command to verify access to your S3 buckets:

Aws s3 ls s3://DOC-EXAMPLE-BUCKET

```
aws Services Search [Alt+S] N. California usha @ 0109-2818-5144
inflating: aws/dist/docutils/writers/s5_html/themes/default/outline.css
inflating: aws/dist/docutils/writers/s5_html/themes/medium-white/framing.css
inflating: aws/dist/docutils/writers/s5_html/themes/medium-white/pretty.css
inflating: aws/dist/docutils/writers/s5_html/themes/big-black/framing.css
inflating: aws/dist/docutils/writers/s5_html/themes/big-black/pretty.css
inflating: aws/dist/docutils/writers/s5_html/themes/big-black/_base
inflating: aws/dist/docutils/writers/s5_html/themes/medium-black/_base
inflating: aws/dist/docutils/writers/s5_html/themes/medium-black/pretty.css
inflating: aws/dist/docutils/writers/s5_html/themes/big-white/framing.css
inflating: aws/dist/docutils/writers/s5_html/themes/big-white/pretty.css
inflating: aws/dist/docutils/writers/html4css1/template.txt
inflating: aws/dist/docutils/writers/html4css1/html4css1.css
inflating: aws/dist/docutils/writers/pep_html/pep.css
inflating: aws/dist/docutils/writers/pep_html/template.txt
inflating: aws/dist/docutils/writers/odf_odt/styles.odt
You can now run: /usr/local/bin/aws --version
root@ip-172-31-10-231:~# ^C
root@ip-172-31-10-231:~# ls
aws awscli v2.zip snap
root@ip-172-31-10-231:~# aws s3 ls
2024-09-04 04:37:23 s3-bucket-usha
2024-09-04 04:31:55 swathi-bucket-1
root@ip-172-31-10-231:~# ls
aws awscli v2.zip snap
root@ip-172-31-10-231:~# touch usha
root@ip-172-31-10-231:~# aws s3 cp usha s3://swathi-bucket-1
upload: ./usha to s3://swathi-bucket-1/usha
root@ip-172-31-10-231:~#
```

i-0004627893e76548e (my s3-ec2)
PublicIPs: 54.177.33.249 PrivateIPs: 172.31.10.231

-----SUCCESSFULLY ACCESSED S3 OBJECTS FROM EC2 INSTANCE-----