Linux & AWS EC2 Web Server with Automated S3 Backup

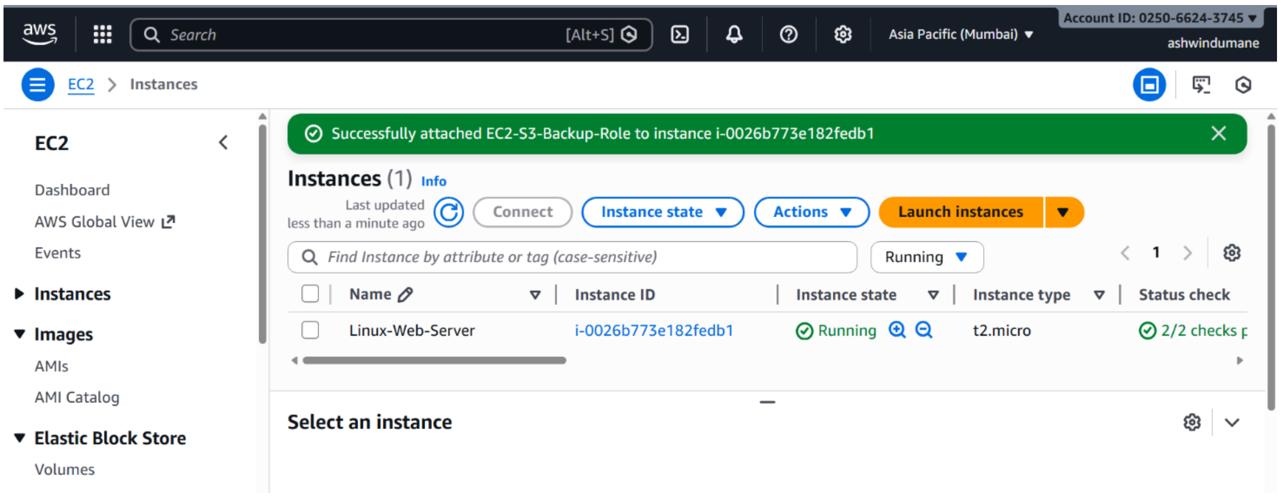
Project Description:

This project focuses on hands-on practice with Linux and AWS services. Students will deploy a Linux-based web server on an AWS EC2 instance and host a simple web application. Application data will be backed up automatically to Amazon S3 using shell scripting and cron jobs. The project helps students understand Linux server management, AWS EC2, S3 storage, IAM roles, and basic automation.

Services Used:

- **Operating System:** Amazon Linux 2023
- **Cloud Platform:** Amazon Web Services (AWS)
- **Compute Service:** Amazon EC2
- **Storage Service:** Amazon S3
- **Identity & Access Management:** AWS IAM (IAM Role for EC2)
- **Web Server:** Apache HTTP Server
- **Automation & Scripting:** Bash (Shell Scripting)
- **Command Line Tools:** AWS CLI

Step 1: AWS EC2 Instance Created



Successfully attached EC2-S3-Backup-Role to instance i-0026b773e182fedb1

Name	Instance ID	Instance state	Instance type	Status check
Linux-Web-Server	i-0026b773e182fedb1	Running	t2.micro	2/2 checks p

This screenshot shows the creation of an Amazon EC2 instance using Amazon Linux. The instance is configured to host a Linux-based web server & showing the EC2 instance is in a running state and has been assigned a public IPv4 address.

Step 2: EC2 User Data Script for Automated Web Server Setup and S3 Backup

EC2 USER DATA SHELL SCRIPT:

```
#!/bin/bash

# Update system

yum update -y


# Install Apache Web Server

yum install httpd -y


# Start and enable Apache

systemctl start httpd

systemctl enable httpd


# Create sample website

cat <<EOF > /var/www/html/index.html

<h1>Linux & AWS EC2 Web Server</h1>

<p>Automated S3 Backup using Shell Script & Cron</p>

EOF


# Install AWS CLI

yum install awscli -y


# Create backup script

cat <<'EOF' > /home/ec2-user/s3_backup.sh

#!/bin/bash


DATE=$(date +%F)

BACKUP_DIR="/var/www/html"

S3_BUCKET="s3://linux-ec2-backup-yourbucketname"


tar -czf /tmp/web-backup-$DATE.tar.gz $BACKUP_DIR

aws s3 cp /tmp/web-backup-$DATE.tar.gz $S3_BUCKET

rm -f /tmp/web-backup-$DATE.tar.gz

EOF
```

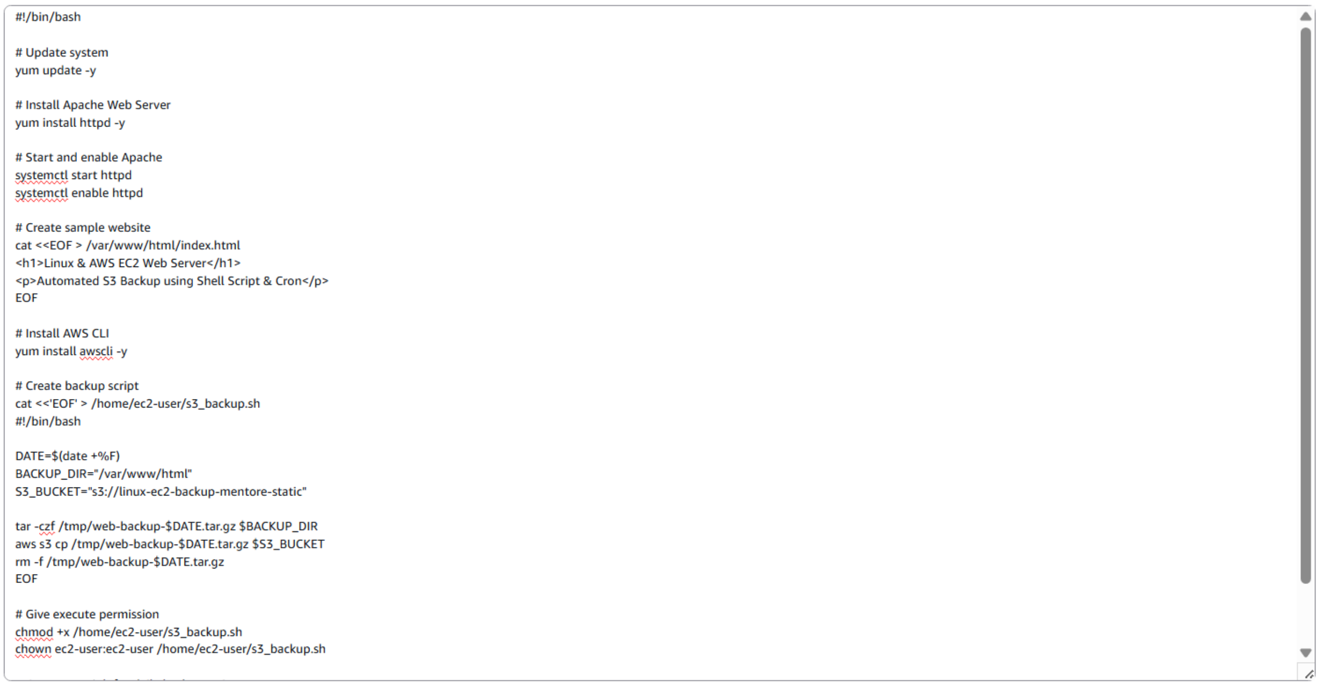
# Give execute permission

chmod +x /home/ec2-user/s3_backup.sh

chown ec2-user:ec2-user /home/ec2-user/s3_backup.sh

# Create cron job for daily backup at 2 AM

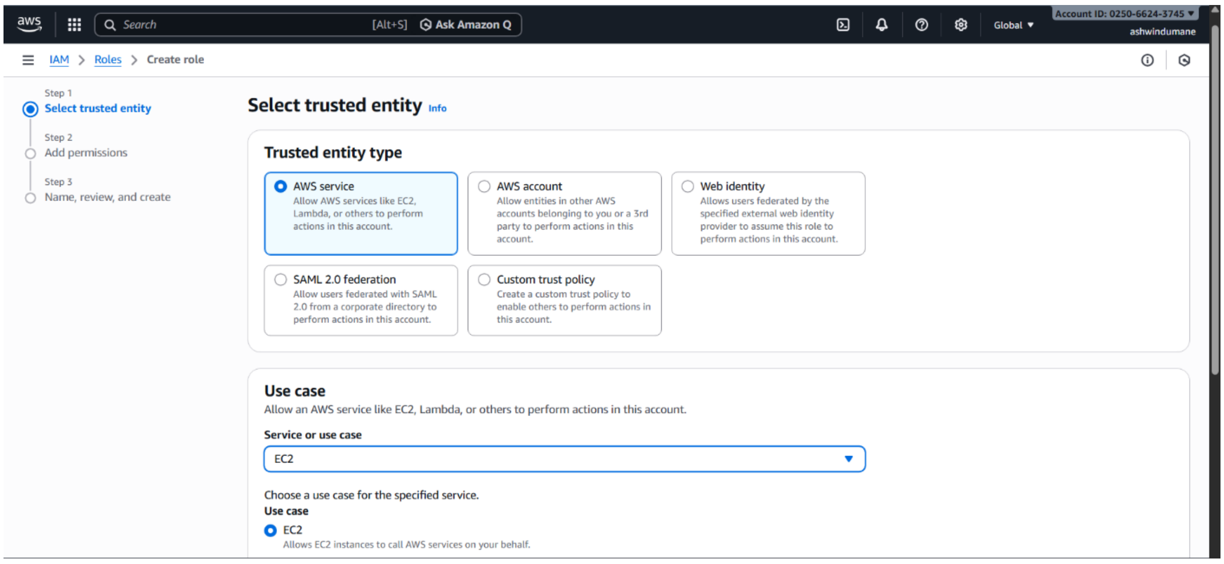
echo "0 2 * * * /home/ec2-user/s3_backup.sh" | crontab -u ec2-user -



```
#!/bin/bash

# Update system
yum update -y

# Install Apache Web Server
yum install httpd -y

# Start and enable Apache
systemctl start httpd
systemctl enable httpd

# Create sample website
cat <<EOF > /var/www/html/index.html
<h1>Linux & AWS EC2 Web Server</h1>
<p>Automated S3 Backup using Shell Script & Cron</p>
EOF

# Install AWS CLI
yum install awscli -y

# Create backup script
cat <<'EOF' > /home/ec2-user/s3_backup.sh
#!/bin/bash

DATE=$(date +%F)
BACKUP_DIR="/var/www/html"
S3_BUCKET="s3://linux-ec2-backup-mentore-static"

tar -czf /tmp/web-backup-$DATE.tar.gz $BACKUP_DIR
aws s3 cp /tmp/web-backup-$DATE.tar.gz $S3_BUCKET
rm -f /tmp/web-backup-$DATE.tar.gz
EOF

# Give execute permission
chmod +x /home/ec2-user/s3_backup.sh
chown ec2-user:ec2-user /home/ec2-user/s3_backup.sh
```

It shows the EC2 *User Data* shell script used during instance creation. The script automatically updates the system, installs and starts the Apache web server, deploys a sample web page, installs the AWS CLI, creates an S3 backup script, and sets the required permissions to enable automated website backups.

Step 3: IAM Role Creation – Trusted Entity Selection



Select trusted entity Info

Trusted entity type

- [✓ AWS service] Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation: Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy: Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case: EC2

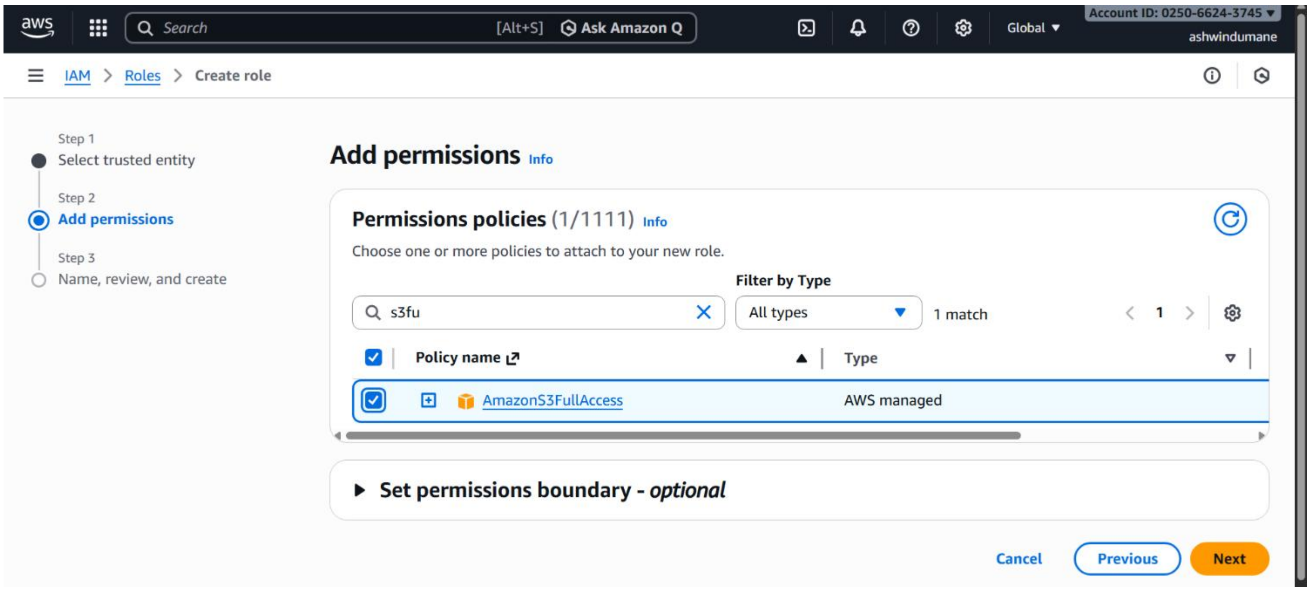
Choose a use case for the specified service.

Use case

- [✓ EC2] Allows EC2 instances to call AWS services on your behalf.

This screenshot shows the creation of an IAM role by selecting **AWS service** as the trusted entity and choosing **EC2** as the use case. This allows the EC2 instance to securely access other AWS services, such as Amazon S3, using the assigned IAM role without storing access keys.

Step 4: Attach S3 Permissions to IAM Role



Add permissions Info

Permissions policies (1/1111) Info

Choose one or more policies to attach to your new role.

Filter by Type

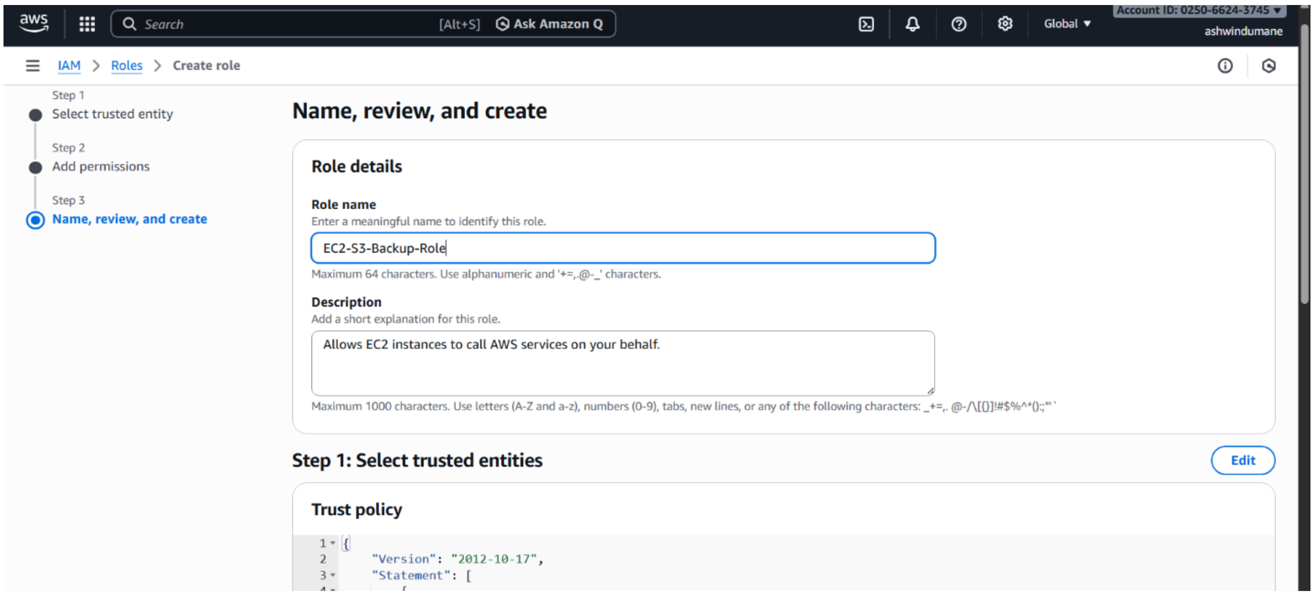
s3fu | All types | 1 match

	Policy name	Type
✓	AmazonS3FullAccess	AWS managed

▶ Set permissions boundary - *optional*

This screenshot shows the attachment of the **AmazonS3FullAccess** managed policy to the IAM role. This permission allows the EC2 instance to upload, list, and manage backup files in the specified Amazon S3 bucket.

Step 5: Name, Review, and Create IAM Role



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

EC2-S3-Backup-Role

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();:"'`

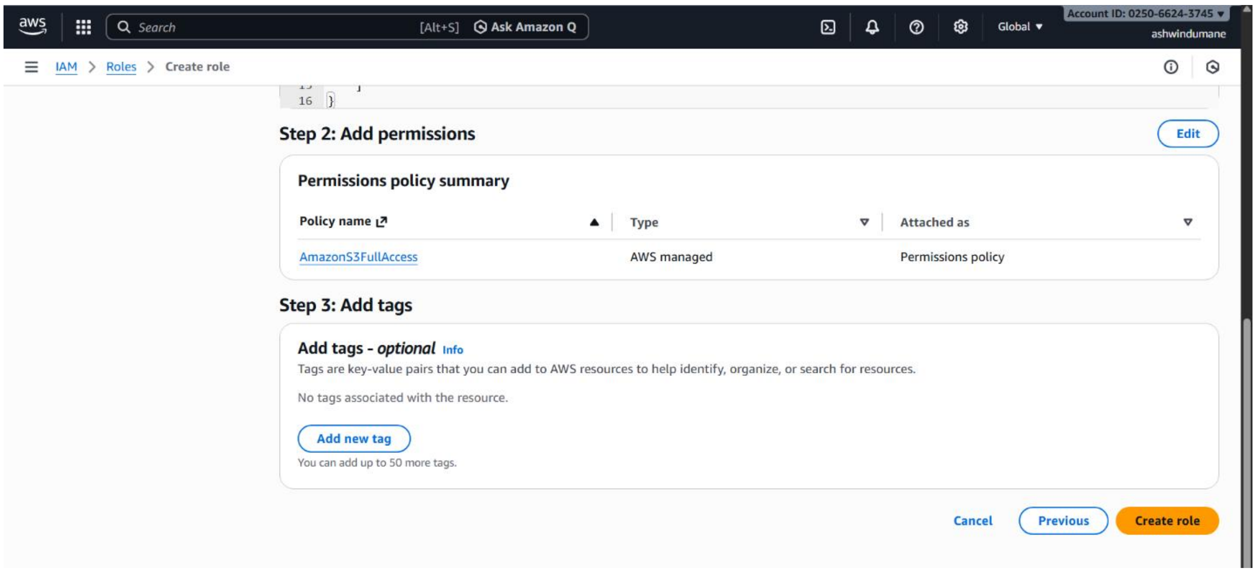
Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
```

This screenshot shows the final step of IAM role creation, where the role is named **EC2-S3-Backup-Role** and reviewed before creation. This role is then ready to be attached to the EC2 instance to enable secure access to Amazon S3 for automated backups.

Step 6: Review Permissions and Create IAM Role



Step 2: Add permissions

Edit

Permissions policy summary

Policy name	Type	Attached as
AmazonS3FullAccess	AWS managed	Permissions policy

Step 3: Add tags

Add tags - *optional* Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

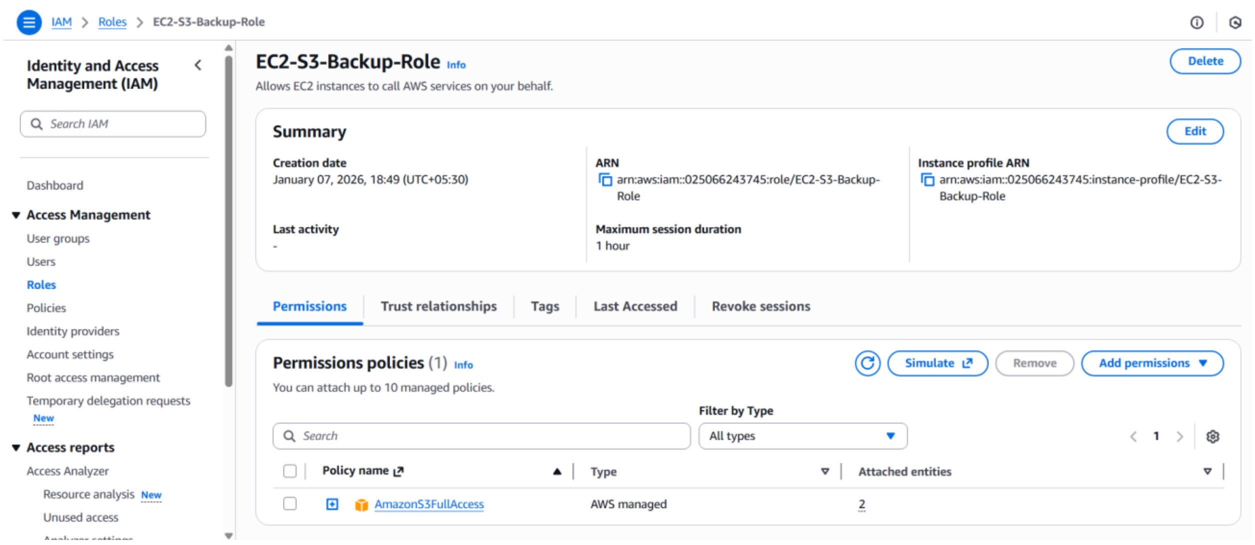
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

This screenshot shows the final review of the attached permissions and optional tags before creating the IAM role. After confirmation, the role is created with the required S3 access and is ready to be assigned to the EC2 instance.

Step 7: IAM Role Successfully Created and Verified



EC2-S3-Backup-Role Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
January 07, 2026, 18:49 (UTC+05:30)

ARN
arn:aws:iam::025066243745:role/EC2-S3-Backup-Role

Instance profile ARN
arn:aws:iam::025066243745:instance-profile/EC2-S3-Backup-Role

Last activity
-

Maximum session duration
1 hour

Permissions policies (1) Info

You can attach up to 10 managed policies.

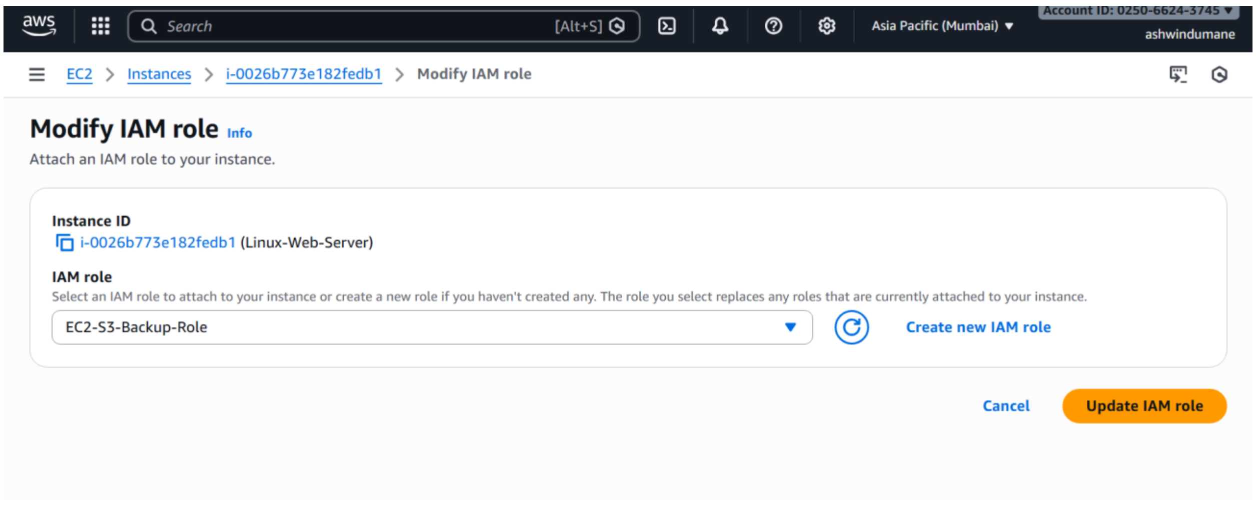
Filter by Type: All types

Policy name	Type	Attached entities
AmazonS3FullAccess	AWS managed	2

This screenshot shows the successfully created IAM role **EC2-S3-Backup-Role** with the **AmazonS3FullAccess** policy attached. It confirms that the role is active and ready to be associated with the EC2 instance to enable secure S3 access for automated backups.

Step 8: Attach IAM Role to EC2 Instance.

This shows the process of attaching the **EC2-S3-Backup-Role** IAM role to the running EC2 instance. By associating this role, the EC2 instance gains secure permission to access Amazon S3 for performing automated backup operations


Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 0250-6624-3745
ashwindumane
EC2 > Instances > i-0026b773e182fedb1 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

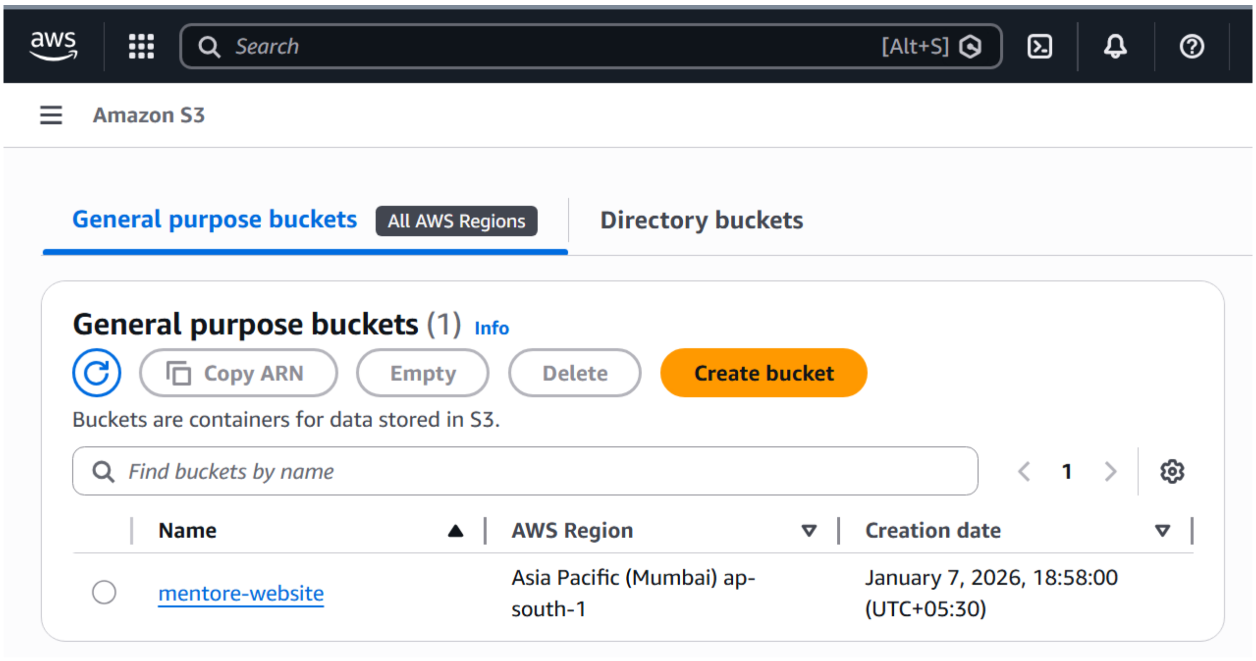
i-0026b773e182fedb1 (Linux-Web-Server)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

EC2-S3-Backup-Role

Step 9: Amazon S3 Bucket Creation



General purpose buckets (1) Info

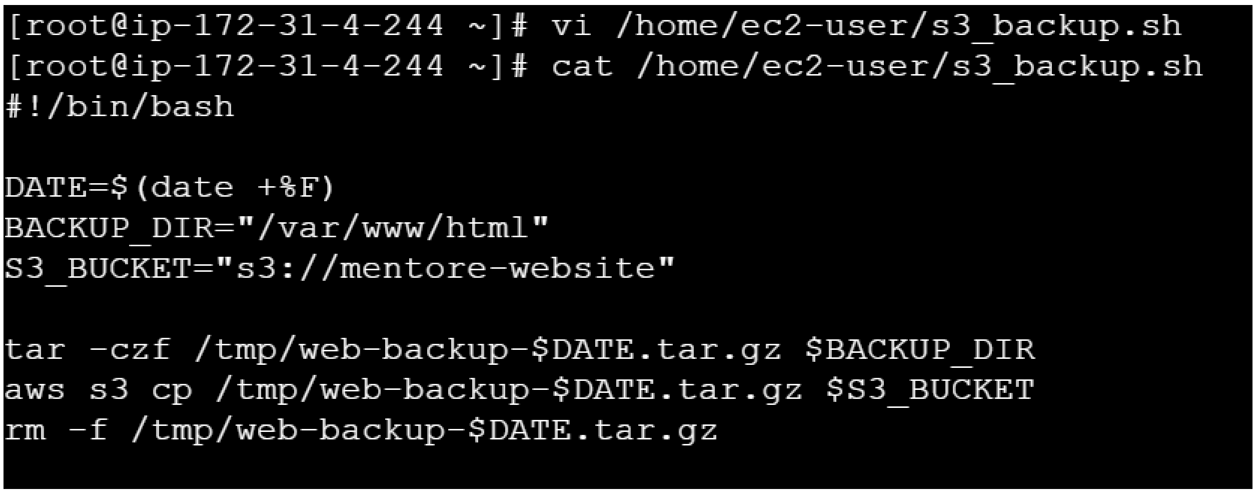
Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
mentore-website	Asia Pacific (Mumbai) ap-south-1	January 7, 2026, 18:58:00 (UTC+05:30)

This screenshot shows the successful creation of the Amazon S3 bucket named **mentore-website** in the Asia Pacific (Mumbai) region. This bucket is used to store automated backup files generated from the EC2 web server.

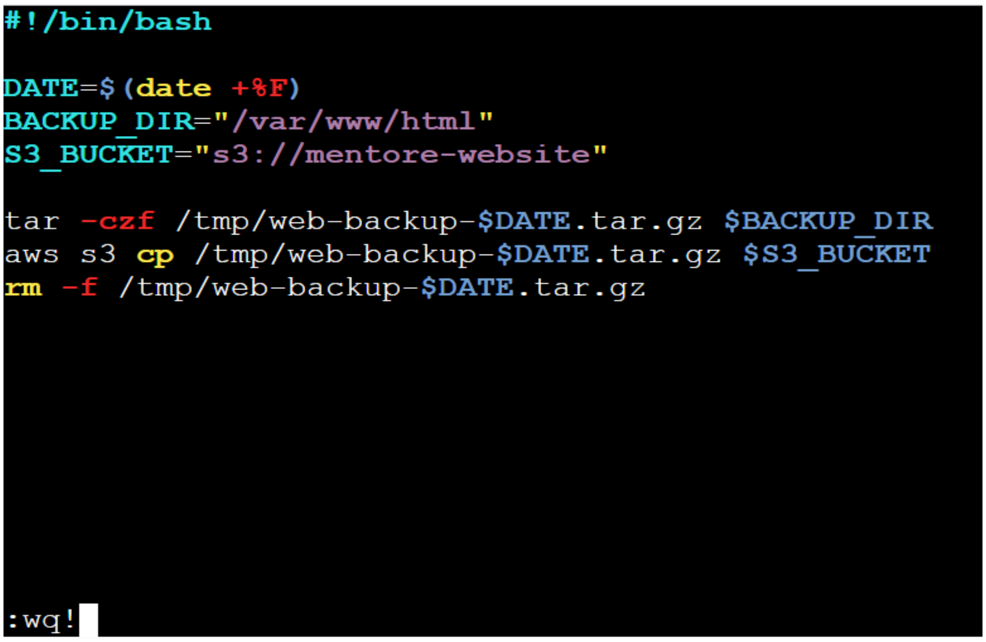
Step 10: S3 Backup Shell Script Configuration

This screenshot shows the creation and verification of the shell script (s3_backup.sh) on the EC2 instance. The script compresses the web server files from /var/www/html and uploads the backup archive to the Amazon S3 bucket **mentore-website**, enabling automated data backup.



```
[root@ip-172-31-4-244 ~]# vi /home/ec2-user/s3_backup.sh
[root@ip-172-31-4-244 ~]# cat /home/ec2-user/s3_backup.sh
#!/bin/bash

DATE=$(date +%F)
BACKUP_DIR="/var/www/html"
S3_BUCKET="s3://mentore-website"

tar -czf /tmp/web-backup-$DATE.tar.gz $BACKUP_DIR
aws s3 cp /tmp/web-backup-$DATE.tar.gz $S3_BUCKET
rm -f /tmp/web-backup-$DATE.tar.gz
```

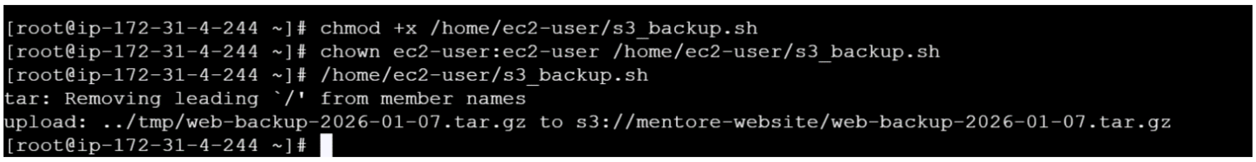
Step 11: Backup Script Finalization and Save



```
#!/bin/bash

DATE=$(date +%F)
BACKUP_DIR="/var/www/html"
S3_BUCKET="s3://mentore-website"

tar -czf /tmp/web-backup-$DATE.tar.gz $BACKUP_DIR
aws s3 cp /tmp/web-backup-$DATE.tar.gz $S3_BUCKET
rm -f /tmp/web-backup-$DATE.tar.gz









:wq!
```

This screenshot shows the finalized s3_backup.sh shell script after configuration and saving. The script defines backup variables, compresses the web server directory, uploads the archive to the Amazon S3 bucket, and removes the temporary file, completing the automated backup setup.

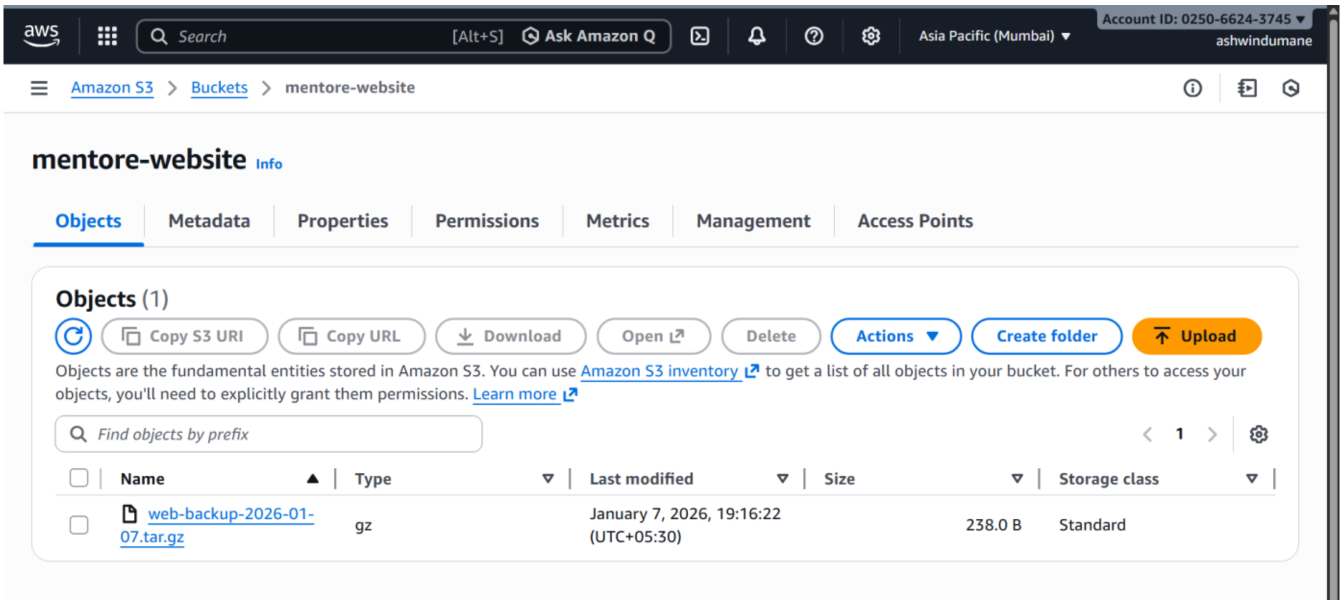
Step 12: Backup Script Execution and Successful Upload to S3



```
[root@ip-172-31-4-244 ~]# chmod +x /home/ec2-user/s3_backup.sh
[root@ip-172-31-4-244 ~]# chown ec2-user:ec2-user /home/ec2-user/s3_backup.sh
[root@ip-172-31-4-244 ~]# /home/ec2-user/s3_backup.sh
tar: Removing leading `/' from member names
upload: ../tmp/web-backup-2026-01-07.tar.gz to s3://mentore-website/web-backup-2026-01-07.tar.gz
[root@ip-172-31-4-244 ~]#
```

This screenshot shows the execution of the backup script after assigning executable permissions and ownership. It confirms that the website backup was successfully created and uploaded to the Amazon S3 bucket, validating the automated backup process.

Step 13: Verification of Backup File in Amazon S3



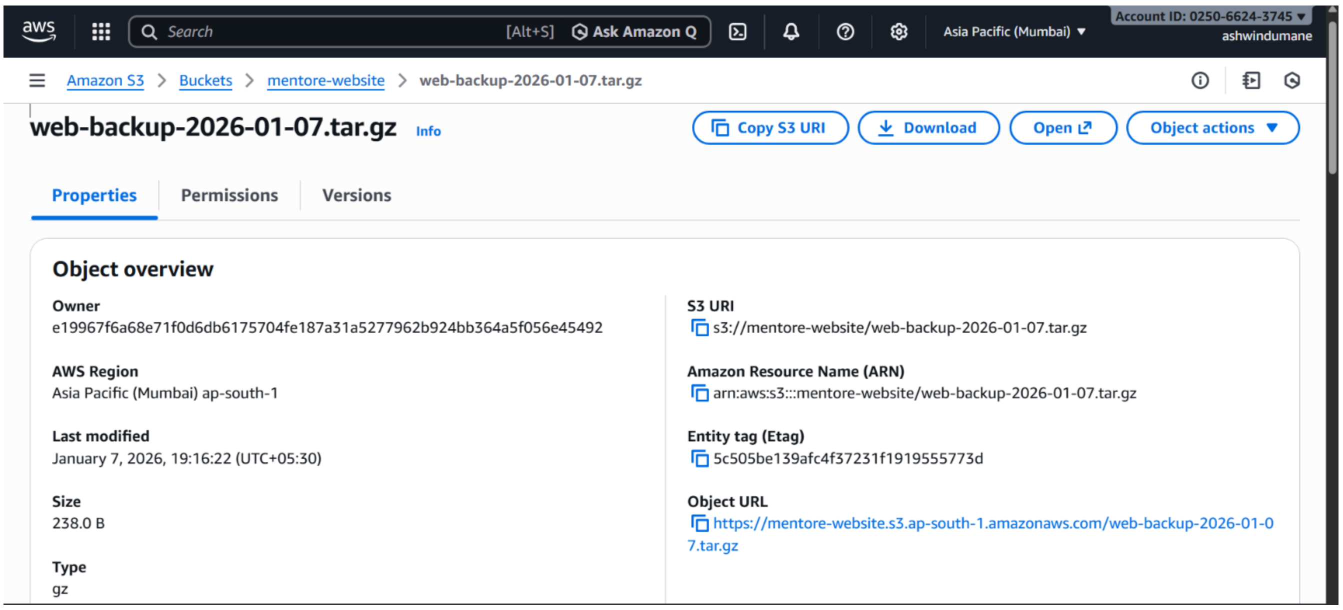
Objects (1)

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. Learn more

Name	Type	Last modified	Size	Storage class
web-backup-2026-01-07.tar.gz	gz	January 7, 2026, 19:16:22 (UTC+05:30)	238.0 B	Standard

This screenshot shows the Amazon S3 bucket **mentore-website** containing the uploaded backup file (web-backup-YYYY-MM-DD.tar.gz). It confirms that the automated backup from the EC2 web server was successfully stored in S3.

Step 14: Backup Object Details in Amazon S3


aws
Search [Alt+S] Ask Amazon Q
Asia Pacific (Mumbai)
Account ID: 0250-6624-3745
ashwindumane
Amazon S3 > Buckets > mentore-website > web-backup-2026-01-07.tar.gz
web-backup-2026-01-07.tar.gz Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions

Object overview

Owner
e19967f6a68e71f0d6db6175704fe187a31a5277962b924bb364a5f056e45492

AWS Region
Asia Pacific (Mumbai) ap-south-1

Last modified
January 7, 2026, 19:16:22 (UTC+05:30)

Size
238.0 B

Type
gz

S3 URI
s3://mentore-website/web-backup-2026-01-07.tar.gz

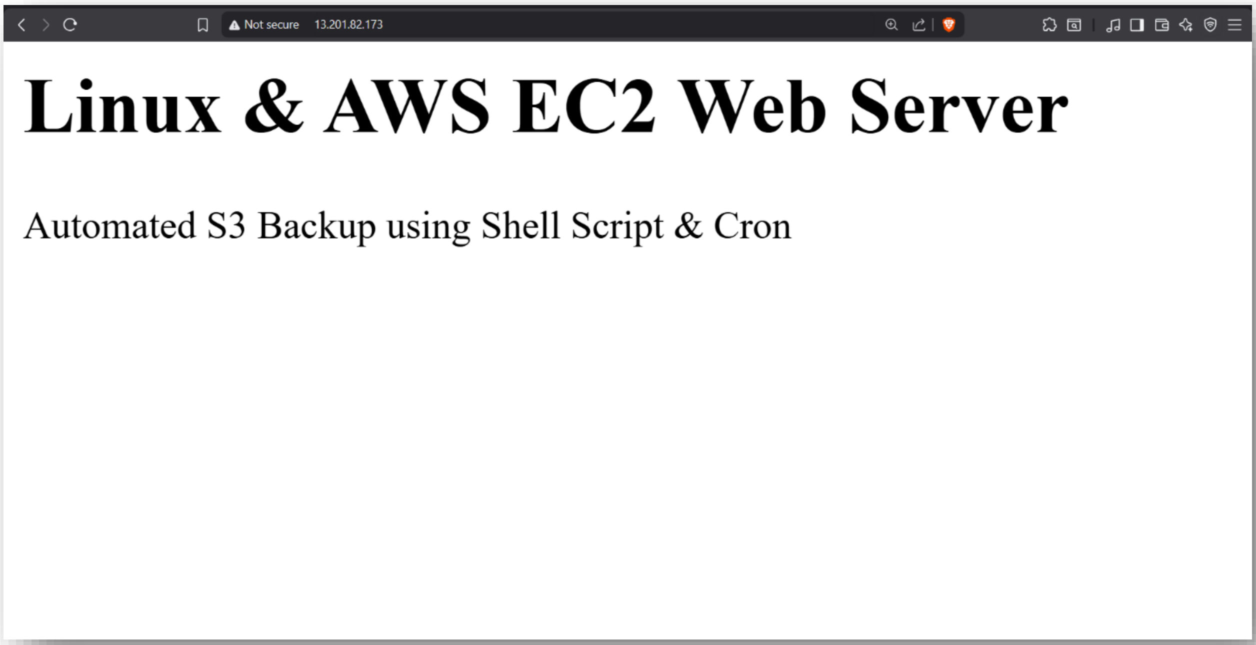
Amazon Resource Name (ARN)
arn:aws:s3:::mentore-website/web-backup-2026-01-07.tar.gz

Entity tag (Etag)
5c505be139afc4f37231f1919555773d

Object URL
https://mentore-website.s3.ap-south-1.amazonaws.com/web-backup-2026-01-07.tar.gz

This screenshot displays the detailed properties of the backup file stored in Amazon S3, including the object name, size, region, and S3 URI. It confirms that the backup file was successfully uploaded and is accessible within the S3 bucket.

Step 15: Web Application Successfully Deployed on EC2



Linux & AWS EC2 Web Server

Automated S3 Backup using Shell Script & Cron

This screenshot shows the live web application accessed through the EC2 public IP address in a browser. It confirms that the Linux-based web server is running correctly and serving the hosted web page, completing the project implementation.