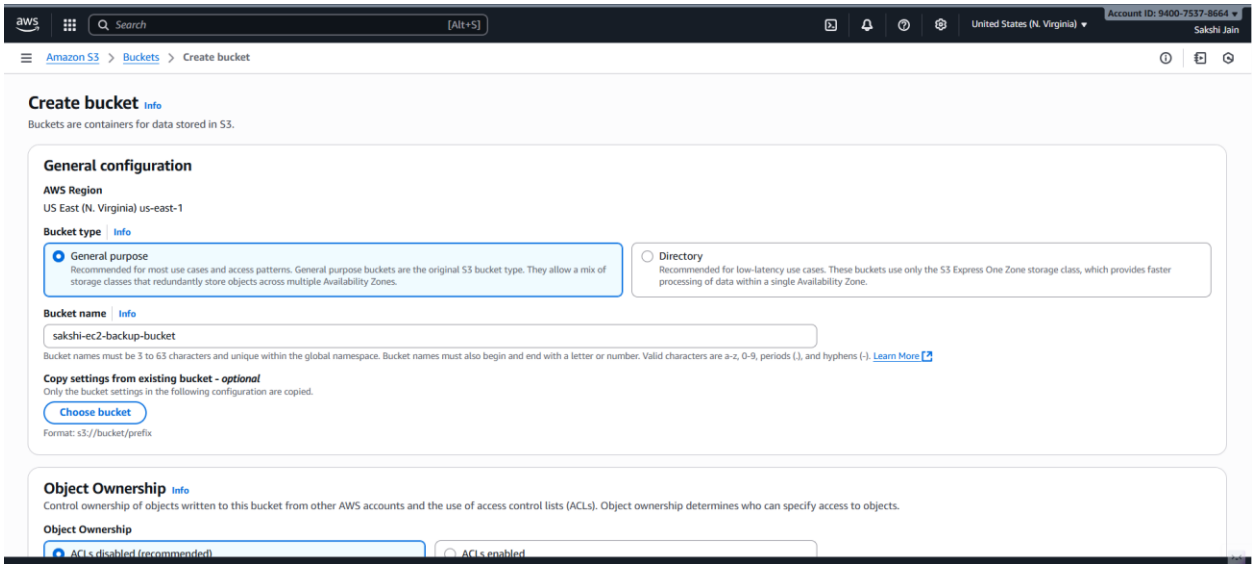
Step 1: Create an S3 Bucket

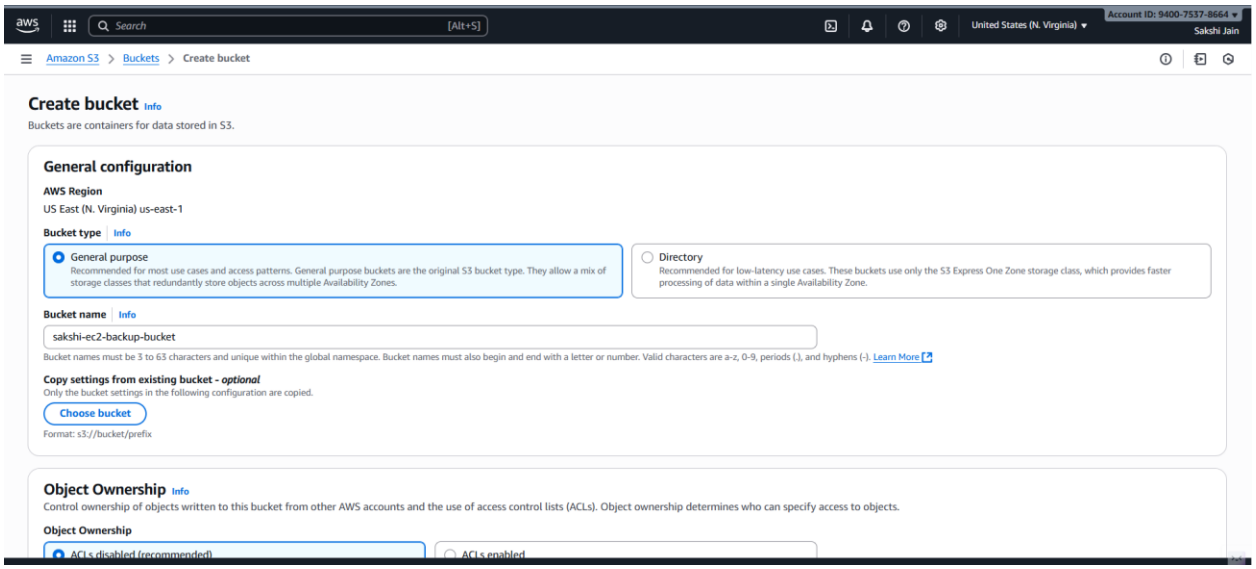
1. Go to **AWS Management Console → S3 → Create bucket**
2. Give a unique name like:
3. sakshi-ec2-backup-bucket
4. Choose your region (same as EC2 region).
5. Keep other settings default → **Create bucket**.

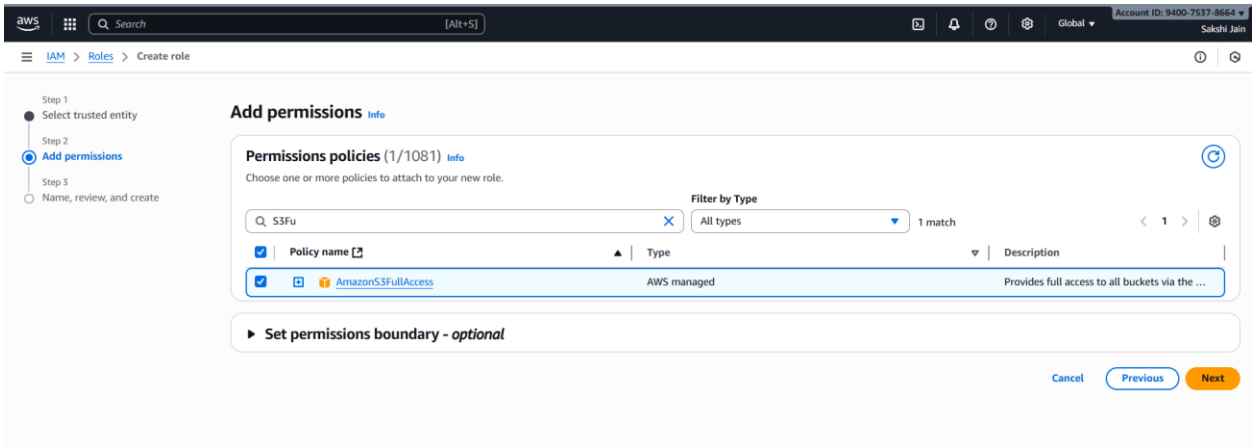


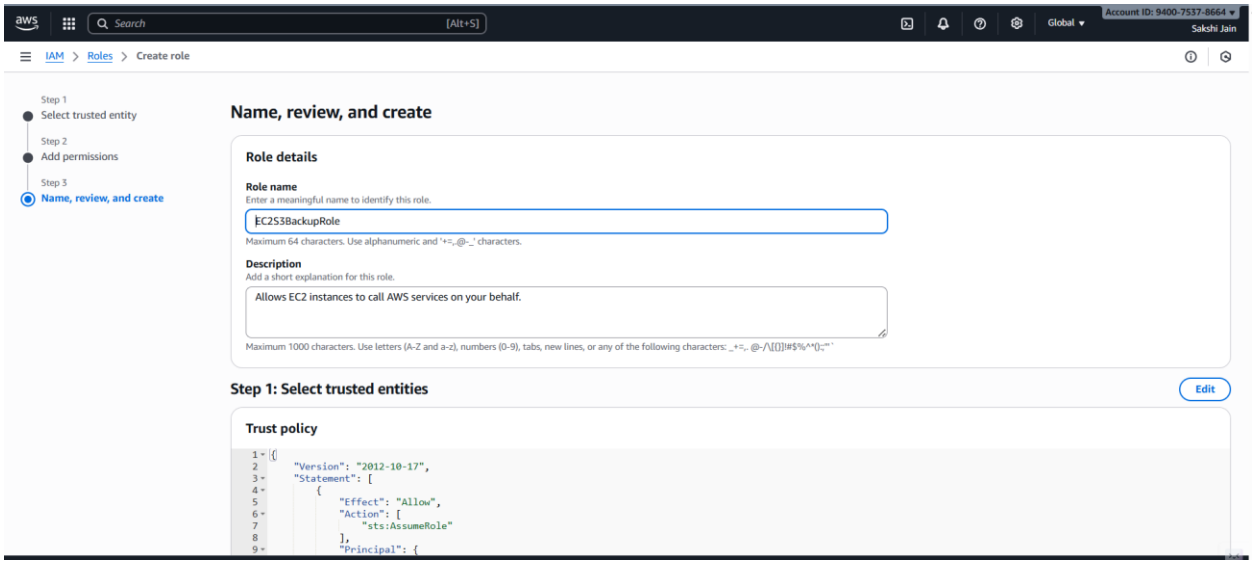
Step 2: Create an IAM Role for EC2

1. Go to **IAM → Roles → Create Role**
2. Choose **AWS Service → EC2**
3. Click **Next: Permissions**
4. Attach the following policy:
    - **AmazonS3FullAccess** (for simplicity in learning)
    - In real projects, use a **custom policy** to limit access to only your backup bucket.
5. Name the role:
6. EC2S3BackupRole

7. Click **Create Role**

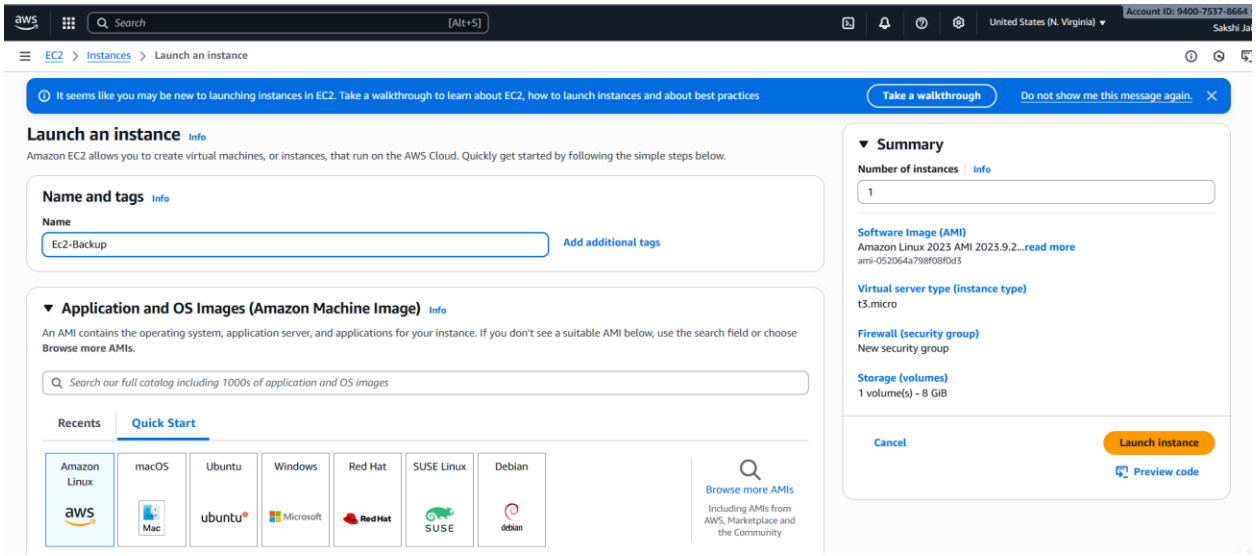


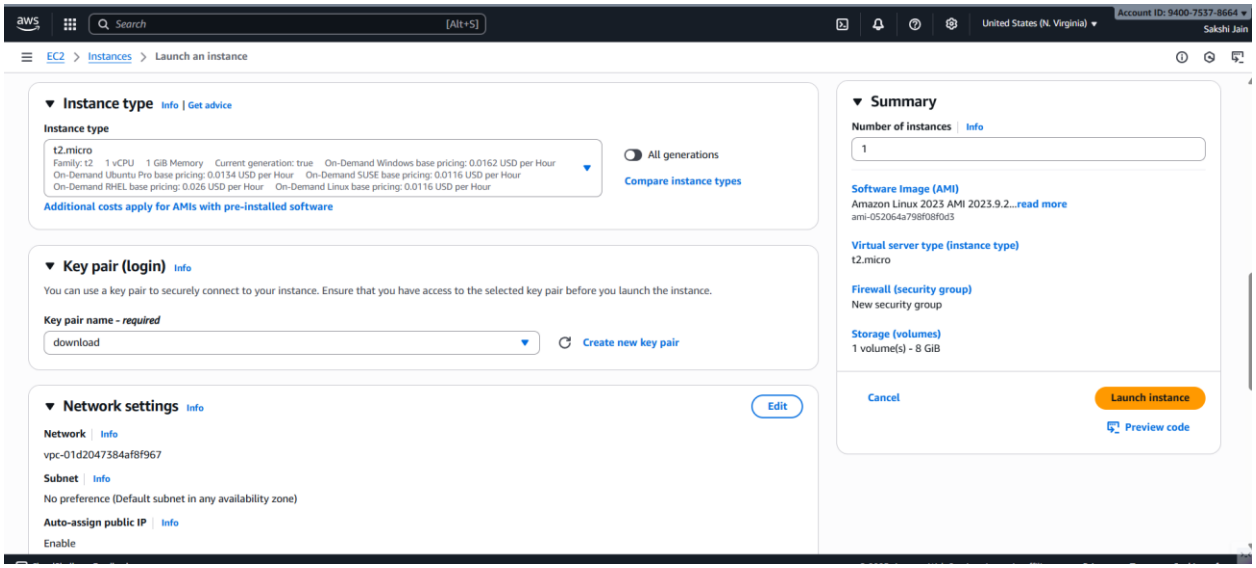




Step 3: Launch an EC2 Instance

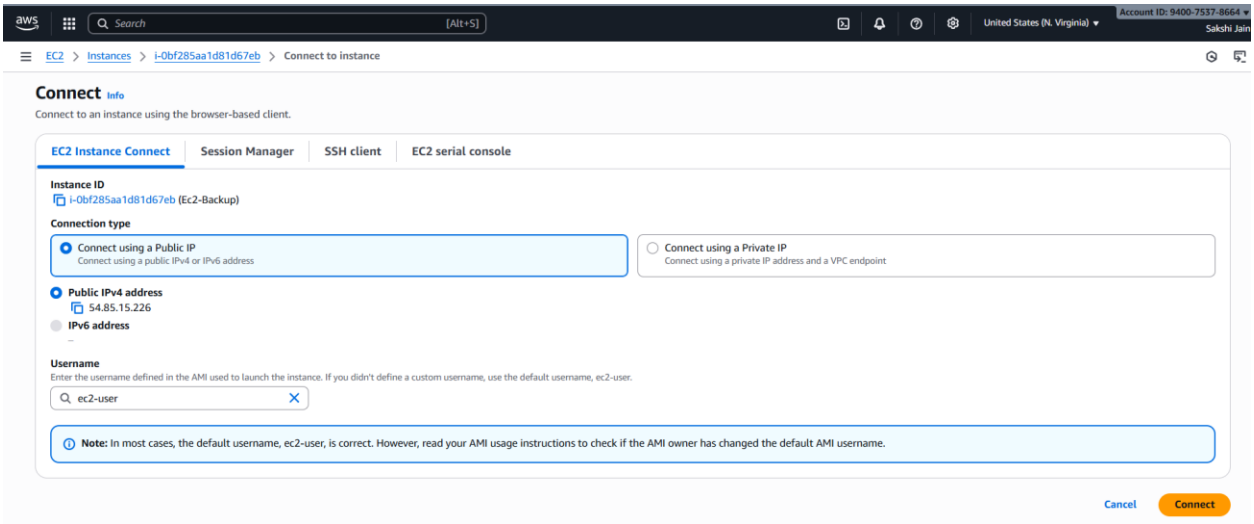
1. Go to **EC2 → Launch Instance**
2. Select:
    - Amazon Linux 2 or Ubuntu (Recommended)
    - t2.micro (Free Tier)
3. Under **IAM Role**, choose:
4. EC2S3BackupRole
5. Add key pair → Launch instance





Step 4: Connect to EC2 via SSH

Use your terminal (Linux/Mac) or PowerShell (Windows):



Step 5: Install AWS CLI

Check if AWS CLI is installed:

aws --version

If not installed (Ubuntu example):

sudo apt update

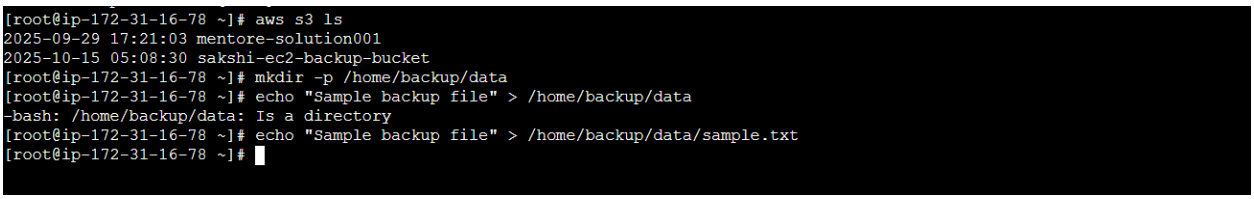
sudo apt install awscli -y

Step 6: Verify IAM Role Access

Run:

aws s3 ls

You should see your S3 buckets — no need for manual key configuration because the EC2 instance has the IAM role attached.

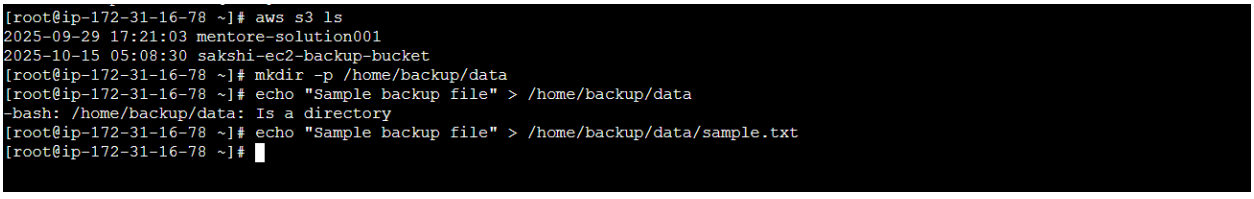


🗂️ **Step 7: Create Backup Directory**

Example: You want to back up /home/backup/data

mkdir -p /home/ubuntu/data

echo "Sample backup file" > /home/backup/data/sample.txt



⚙️ Step 8: Create Backup Script

Create a script file:

sudo vim /home/backup/backup_to_s3.sh

Add the following code:

#!/bin/bash

Variables

SOURCE_DIR="/home/backup/data"

S3_BUCKET="s3://sakshi-ec2-backup-bucket"

DATE=$(date +'%Y-%m-%d-%H-%M')

BACKUP_FILE="/tmp/backup-$DATE.tar.gz"

Create backup archive

tar -czf $BACKUP_FILE $SOURCE_DIR

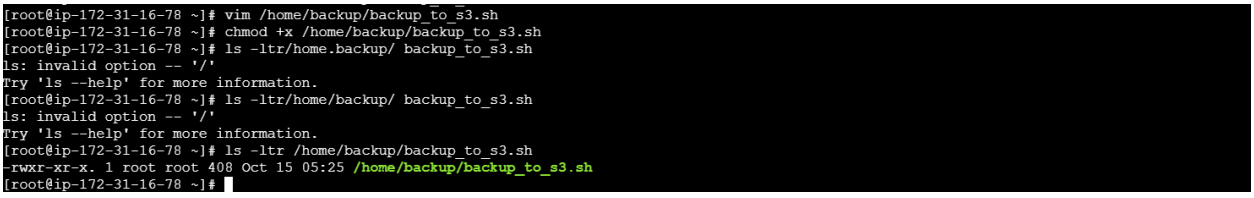
Upload to S3

aws s3 cp $BACKUP_FILE $S3_BUCKET/

Remove local temp backup file

rm $BACKUP_FILE

echo "Backup completed successfully at $(date)" >> /home/backup/backup_log.txt



Make Script Executable

chmod +x /home/backup/backup_to_s3.sh

```
[root@ip-172-31-16-78 ~]# vim /home/backup/backup_to_s3.sh
[root@ip-172-31-16-78 ~]# chmod +x /home/backup/backup_to_s3.sh
[root@ip-172-31-16-78 ~]# ls -ltr/home.backup/ backup_to_s3.sh
ls: invalid option -- '/'
Try 'ls --help' for more information.
[root@ip-172-31-16-78 ~]# ls -ltr/home/backup/ backup_to_s3.sh
ls: invalid option -- '/'
Try 'ls --help' for more information.
[root@ip-172-31-16-78 ~]# ls -ltr /home/backup/backup_to_s3.sh
-rwxr-xr-x. 1 root root 408 Oct 15 05:25 /home/backup/backup_to_s3.sh
[root@ip-172-31-16-78 ~]#
```

Test It Manually

/home/backup/backup_to_s3.sh

Check S3:

aws s3 ls s3://sakshi-ec2-backup-bucket

You should see your backup file uploaded ✅

```
[root@ip-172-31-16-78 ~]# /home/backup/backup_to_s3.sh
tar: Removing leading `/' from member names
pload: ../tmp/backup-2025-10-15-05-33.tar.gz to s3://sakshi-ec2-backup-bucket/backup-2025-10-15-05-33.tar.gz
[root@ip-172-31-16-78 ~]# aws s3 ls s3://sakshi-ec2-backup-bucket
2025-10-15 05:33:03        178 backup-2025-10-15-05-33.tar.gz
```

⏰ Step 9: Automate with Cron Job

To schedule the backup **daily at midnight:**

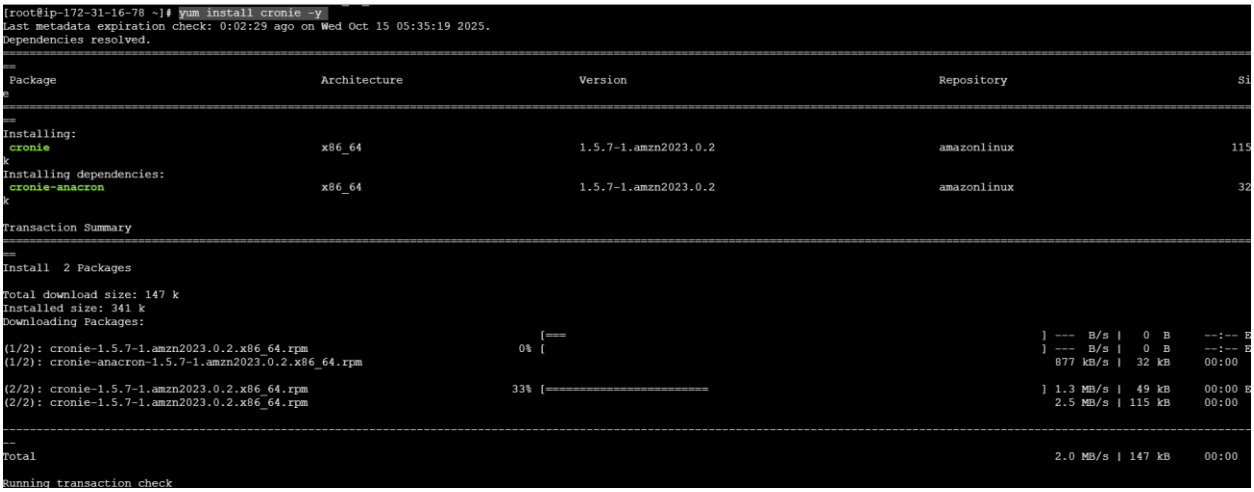
Open cron:

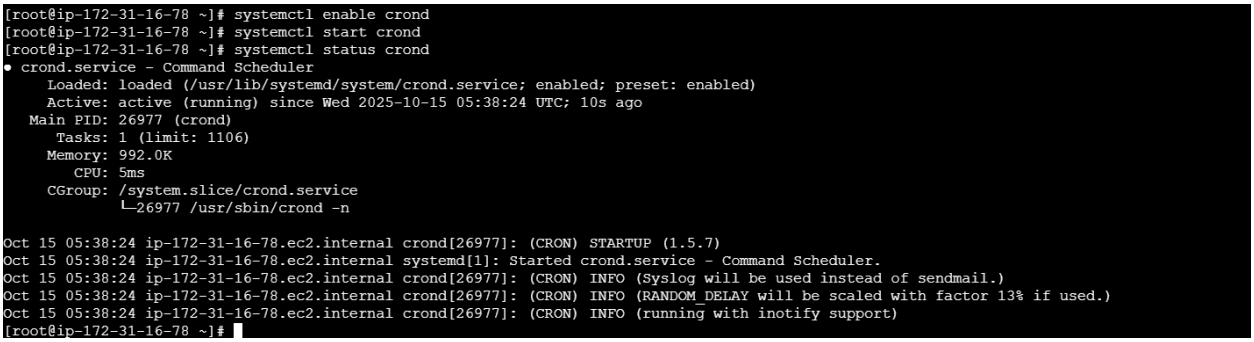
crontab -e

Add the line:

20 11 * * * /home/backup/backup_to_s3.sh

✅ This runs your backup script every day at **12:00 AM**.





🔄 Step 10: Create Restore Script

Create another script file:

sudo vim /home/backup/restore_from_s3.sh

Add:

#!/bin/bash

Variables

S3_BUCKET="s3://sakshi-ec2-backup-bucket"

RESTORE_DIR="/home/backup/restore"

LATEST_BACKUP=$(aws s3 ls $S3_BUCKET/ | sort | tail -n 1 | awk '{print $4}')

Create restore directory

mkdir -p $RESTORE_DIR

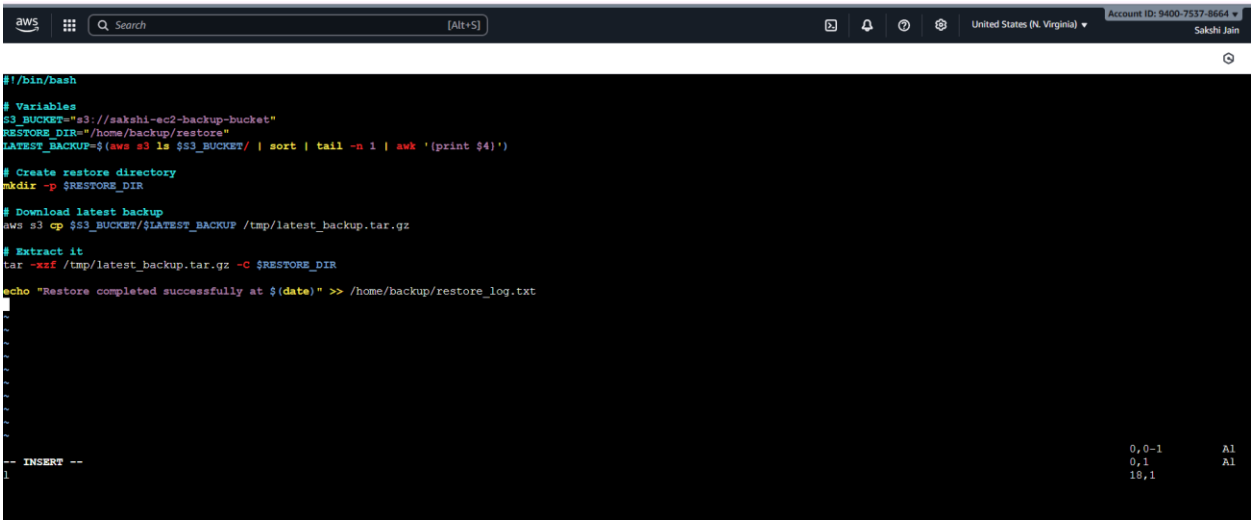
Download latest backup

aws s3 cp $S3_BUCKET/$LATEST_BACKUP /tmp/latest_backup.tar.gz

Extract it

tar -xzf /tmp/latest_backup.tar.gz -C $RESTORE_DIR

echo "Restore completed successfully at $(date)" >> /home/backup/restore_log.txt



Make Script Executable

chmod +x /home/backup/restore_from_s3.sh

```
[ec2-user@ip-172-31-16-78 ~]$ sudo vim /home/backup/restore_from_s3.sh
[ec2-user@ip-172-31-16-78 ~]$ chmod +x /home/backup/restore_from_s3.sh
chmod: changing permissions of '/home/backup/restore_from_s3.sh': Operation not permitted
[ec2-user@ip-172-31-16-78 ~]$ sudo chmod +x /home/backup/restore_from_s3.sh
[ec2-user@ip-172-31-16-78 ~]$ sudo /home/backup/restore_from_s3.sh
download: s3://sakshi-ec2-backup-bucket/backup-2025-10-15-05-33.tar.gz to ../../tmp/latest_backup.tar.gz
[ec2-user@ip-172-31-16-78 ~]$
```

Run Restore Script

/home/backup/restore_from_s3.sh

Your data will be restored under /home/backup/restore/.