

# Question

Start: 3/30/2025, 11:00:00 AM

End: 3/30/2025, 2:00:00 PM

## Problem Statement: Automate a Lightweight Log Archival Pipeline

### Background

Your company runs multiple lightweight applications on EC2 instances. These applications generate local log files daily (e.g., `/var/log/app.log`). To simplify storage, centralize access, and reduce costs, your team wants to automate the archival of these logs to Amazon S3 at the end of each day.

The goal is to build a minimum viable setup where:

- Log files from an EC2 instance are uploaded to an S3 bucket daily.
- The EC2 instance should not store AWS credentials locally.
- The uploaded logs should be organized by date in S3 (e.g., `s3://your-bucket-name/logs/2025-03-30/app.log`).
- All S3 uploads should be versioned and securely stored.

You've been asked to set up and demonstrate this pipeline on a single EC2 instance first, before scaling it out to other instances.

### Your Task

Set up a basic log archival pipeline using the following AWS services:

#### EC2:

- Launch an Amazon Linux 2 instance.
- Simulate application logs by writing to `/var/log/app.log` using a cron job or script.
- Install necessary tools (AWS CLI, etc.).
- Secure the instance:
  - Only allow SSH from your IP.
  - Use an IAM role, not access keys, to grant permissions to S3.

#### S3:

- Create a private bucket.
- Enable versioning and optionally server-side encryption.
- Organize uploaded logs under `logs/YYYY-MM-DD/` structure.

### ✅ IAM:

- Create an EC2 instance role with least privilege access to your S3 bucket:
  - s3:PutObject, s3:GetObject, s3:ListBucket only on your bucket.

### ✅ Automation:

- Write a script (Shell or Python) that:
  - Compresses the log file (gzip or similar).
  - Uploads it to S3 with the correct path format.
- Configure it to run once per day using cron or systemd timers.
- Log the upload result locally for auditing.

### ✅ Deliverables:

- A GitHub repo or zip with:
  - Your upload script.
  - A sample IAM policy you attached to the EC2 instance.
  - (Optional) A user-data script that bootstraps the instance.
  - A README documenting:
    - The steps you followed.
    - How to test the upload.
    - Where to find the uploaded logs in S3.
- A screenshot or CLI output showing:
  - The file in S3 under the correct path.
  - The S3 versioning in effect.

Submit here: <https://forms.gle/ER387znmXfN4MvzdA>

All the best. You got this!