Automated S3 Backup and Restore using EC2 & IAM

---

1. Introduction

In today's digital world, data is one of the most valuable assets for any organization. Protecting it from accidental loss, corruption, or system failure is crucial.
This project demonstrates an automated data backup and restore solution using **Amazon Web Services (AWS)** — specifically **EC2, S3, IAM, and Bash scripting**.

The system automatically takes daily backups of selected directories from an **EC2 instance** and uploads them to an **S3 bucket**. When needed, the system can restore the latest backup from S3 to the EC2 instance with a simple script.

---

2. Necessity of the Project

- Manual backups are **time-consuming and error-prone**.
- System crashes or data corruption can lead to **data loss** without a proper backup plan.
- Businesses need **automated, reliable, and scalable** backup solutions.
- Cloud platforms like AWS offer **cost-effective** and **highly available** data storage services.

Hence, automation of backup and restore using AWS ensures **data safety, reliability, and ease of recovery**.

---

3. Motivation for the Project

The motivation behind this project comes from the need to ensure **data continuity and disaster recovery** in both academic and enterprise environments.
By learning to use AWS automation tools, we not only protect data but also gain practical skills in:

- **Cloud computing**
- **Bash scripting**
- **System administration**
- **Data management**

This project reflects real-world DevOps and cloud operations used in IT industries.

---

4. Objective

The main objective is to **automate the process of data backup and restoration** using AWS cloud resources.

Specific objectives:

1. Create an EC2 instance with an IAM role that allows S3 access.
2. Develop a Bash script to automatically back up specific directories.
3. Upload backups to S3 daily using **AWS CLI and cron jobs**.
4. Create a restore script to download and extract the latest backup from S3.
5. Ensure logging and error handling for reliability.

---

5. Literature Survey

Previous studies and practices in data management highlight that:

- Cloud-based storage (e.g., AWS S3, Azure Blob, Google Cloud Storage) provides **durable, redundant, and scalable** data storage.
- Automation using shell scripts or configuration management tools (e.g., Ansible, Terraform) simplifies **data backup management**.
- Research papers emphasize that automated cloud backup solutions reduce **human error** and improve **business continuity**.
- AWS documentation and blogs provide step-by-step tutorials for using **IAM roles**, **S3 buckets**, and **AWS CLI** for automation tasks.

Thus, this project integrates best practices from both research and real-world cloud implementation.

---

6. Research Question

“How can we build a reliable, secure, and automated backup system on AWS that allows effortless restoration of data in case of loss or system failure?”

---

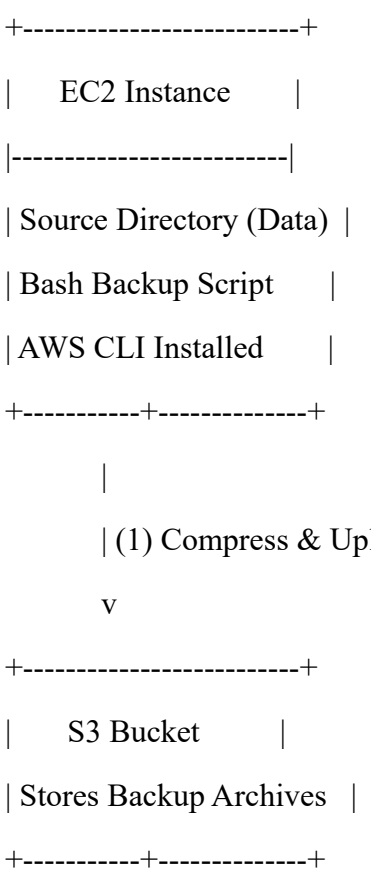
7. System Structure

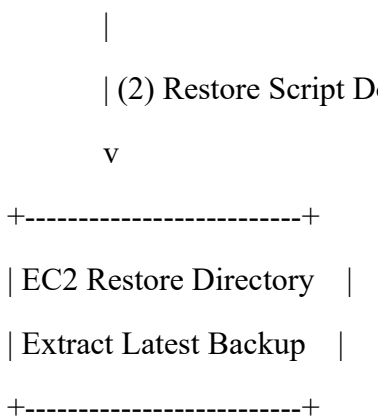
The system consists of the following major components:

1. **AWS EC2 Instance:**
   Hosts application data and executes backup scripts.
2. **IAM Role:**
   Grants EC2 secure permission to access the S3 bucket.
3. **S3 Bucket:**
   Stores compressed backup archives.
4. **Backup Script:**
   Automates backup creation and upload to S3.
5. **Restore Script:**
   Downloads and extracts data from S3 to EC2.
6. **Cron Job:**
   Schedules daily automatic backups.

---

8. System Design Framework (Flow Diagram)





---

9. Methodology

Step 1: Create an S3 bucket for backup storage.

Step 2: Create an IAM role with AmazonS3FullAccess and attach it to the EC2 instance.

Step 3: Launch an EC2 instance and install AWS CLI.

Step 4: Write and test the backup script to compress and upload data.

Step 5: Write and test the restore script to download and extract data.

Step 6: Schedule the backup script using a cron job for daily automation.

Step 7: Maintain log files for tracking backup and restore operations.

---

10. Proposed Learning Strategy

Through this project, the learning approach focuses on **hands-on AWS practice** and **automation scripting**:

- Understand AWS resource interaction (IAM ↔ EC2 ↔ S3)
- Apply **Bash scripting** concepts (loops, conditionals, variables)
- Learn **AWS CLI commands**
- Implement **cron scheduling**
- Manage logs for debugging and monitoring

This approach builds real-world skills for **DevOps and Cloud Engineer** roles.

---

11. Requirement Specification

Category	Details
Software	AWS account, Amazon Linux/Ubuntu (EC2), AWS CLI, Bash
Hardware	EC2 t2.micro instance (1 vCPU, 1GB RAM)
AWS Services	S3, EC2, IAM
Scripts	backup_to_s3.sh, restore_from_s3.sh
Network	Internet access for AWS API communication
Tools	Terminal/SSH client, Cron scheduler

---

12. Result and Discussion

- The **backup script** successfully compressed and uploaded selected directories to the S3 bucket.
- The **restore script** automatically fetched the latest backup and extracted it to the EC2 instance.
- **Logs** were maintained for both backup and restore operations.
- The entire process can be **automated** and **repeated daily** without manual intervention.

This demonstrates a **robust, low-cost, and scalable** solution for small-scale cloud data management.

---

13. Advantages

✅ Fully automated daily backups
✅ Secure IAM-based access control
✅ Reliable and scalable cloud storage (S3)
✅ Easy data restoration process
✅ Cost-effective (uses free-tier resources)
✅ Real-world DevOps use case

---

14. Disadvantages

⚠️ Requires internet access (no offline backup)
⚠️ AWS CLI and IAM setup can be complex for beginners
⚠️ S3 storage cost increases with large data volumes
⚠️ Backup frequency fixed by cron schedule (manual edit needed for changes)

---

15. Applications

- Server and website data backups
- Educational institutions storing student records
- Cloud-based project environments
- DevOps pipelines for automatic configuration backup
- Disaster recovery solutions for small businesses

---

16. Conclusion

This project successfully demonstrates how AWS cloud services can be leveraged to automate backup and restoration processes.
By combining **EC2**, **S3**, **IAM roles**, **Bash scripting**, and **cron jobs**, we created a practical solution for **data protection and disaster recovery**.

The skills learned — cloud configuration, automation scripting, and AWS CLI usage — are directly applicable to real-world DevOps and cloud operations.

In conclusion, **Automated S3 Backup and Restore** ensures data safety, simplifies management, and reflects a professional-level understanding of cloud automation.

Project Members:
Sakshi Jain
Umesh Chimankar
Jay Soni
Nikita Binnar
Vaishavi Kumbhar
Kaveri Kanawade