**PROBLEM STATEMENT**

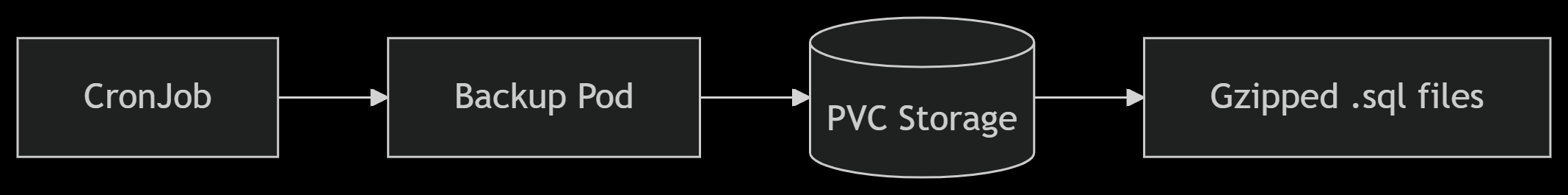
This project addresses the challenge of automating MySQL database backups in Kubernetes by implementing a scheduled CronJob that securely dumps, compresses, and stores backups in a PersistentVolumeClaim (PVC), ensuring data integrity with verification logs while overcoming limitations like manual intervention and storage scalability—though it currently relies on Minikube uptime and lacks cloud integration or alerting for production-grade deployment.

**AIM**

To automate nightly MySQL database backups in Kubernetes using:

* CronJobs for scheduling
* PersistentVolumeClaims (PVC) for storage
* mysqldump + gzip for compressed backups

**PROCEDURE**



**Key Components Visualized:**

1. Automation Path: CronJob → Backup Pod → Script Execution
2. Data Flow: MySQL → mysqldump → Compression → PVC Storage

**TOOLS:**

1. Kubernetes (Minikube for local cluster)
2. kubectl (Kubernetes CLI)
3. Docker (Containerization)
4. MySQL 5.7 (Database)
5. mysqldump (Backup utility)
6. gzip (Compression)

**PROJECT STRUCTURE:**

mysql-backup/

├── backup.sh # Backup script (mysqldump + gzip)

├── Dockerfile # Builds the backup image

├── mysql-backup-cronjob.yaml # Kubernetes CronJob definition

├── mysql-deployment.yaml # MySQL database deployment

├── mysql-pvc.yaml # PersistentVolumeClaim for backups

└── secrets/ # (Optional) Store credentials securely

└── mysql-secrets.yaml # Kubernetes Secrets for DB credentials

**IMPLMENTATION:**

**COMMANDS:**

# Apply configurations

kubectl apply -f mysql-pvc.yaml

kubectl apply -f mysql-deployment.yaml

kubectl apply -f mysql-backup-cronjob.yaml

# Verify

kubectl get pods

kubectl get cronjobs

kubectl describe pvc mysql-pv-claim

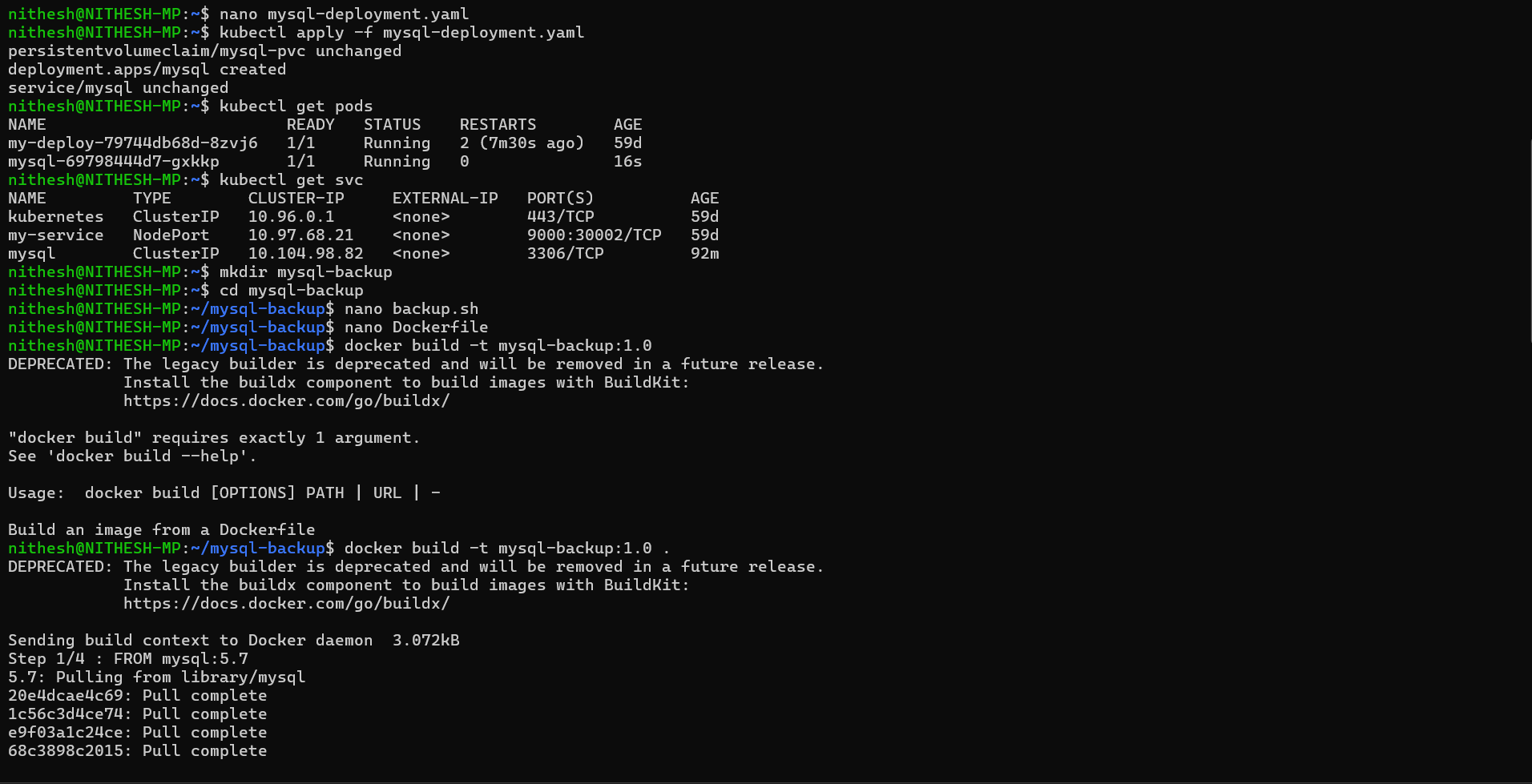
**RESULT**

The project successfully automates MySQL backups in Kubernetes, with CronJobs creating nightly compressed (gzip) backups stored in PersistentVolumeClaims (PVC). Logs confirm successful execution (e.g., Backup successful: /backup/all-databases-2025-05-22.sql.gz), and retention policies delete files older than 7 days.

**CONCLUSION**

This solution provides a **reliable, scalable backup system** for Kubernetes-hosted MySQL databases. While limited to Minikube’s uptime, it demonstrates a production-ready pattern extensible to cloud clusters (EKS/GKE) and cloud storage (S3/GCS). Future enhancements could include backup verification and alerting.

**SCREENSHOTS**

****

