S3 Backup Job

**📌 Attach an IAM Role to EC2 with S3 Access**

1. Go to the **IAM console** in AWS.
2. Create a new role:
   * **Trusted entity**: EC2
   * **Attach policy**: AmazonS3FullAccess *(or a custom policy with limited permissions)*
3. Attach this role to your EC2 instance:
   * **EC2 Console → Instances → Select your instance → Actions → Security → Modify IAM Role**

**📂 Kubernetes Configuration (in k8s folder)**

**1. persistent-volume.yml**

*apiVersion: v1*

*kind: PersistentVolume*

*metadata:*

*name: mysql-backup-pv*

*spec:*

*capacity:*

*storage: 1Gi*

*accessModes:*

*- ReadWriteOnce*

*hostPath:*

*path: "/mnt/data/mysql-backups"*

**2. persistent-volume-claim.yml**

*apiVersion: v1*

*kind: PersistentVolumeClaim*

*metadata:*

*name: mysql-backup-pvc*

*spec:*

*accessModes:*

*- ReadWriteOnce*

*resources:*

*requests:*

*storage: 1Gi*

🖥️ **Apply the PV and PVC:**

*kubectl apply -f persistent-volume.yml*

*kubectl apply -f persistent-volume-claim.yml*

**🕓 Check CronJob Schedule Pattern**

Use the website [CrontabGuru](https://crontab.guru) to validate your cron expression.

**3. cronjob.yml**

*apiVersion: batch/v1*

*kind: CronJob*

*metadata:*

*name: mysql-backup-upload*

*spec:*

*schedule: "0 3 \* \* \*" # Daily at 3 AM*

*jobTemplate:*

*spec:*

*template:*

*spec:*

*containers:*

*- name: backup-upload*

*image: amazonlinux:2023*

*env:*

*- name: MYSQL\_PWD*

*value: "root" # Consider a Kubernetes Secret!*

*command:*

*- /bin/bash*

*- -c*

*- |*

*echo "[+] Installing tools..."*

*yum install -y mariadb105 gzip awscli > /dev/null*

*TIMESTAMP=$(date +%F-%H-%M)*

*FILE=/mnt/backup/quantumsoft-$TIMESTAMP.sql.gz*

*echo "[+] Dumping MySQL database..."*

*mysqldump -h mysql-service -u root quantumsoft | gzip > "$FILE"*

*if [ -f "$FILE" ]; then*

*echo "[+] Uploading $FILE to S3..."*

*aws s3 cp "$FILE" s3://mysql-backups-quantumsoft/mysql/*

*echo "[✓] Backup uploaded successfully."*

*else*

*echo "[!] Backup file missing: $FILE" >&2*

*exit 1*

*fi*

*volumeMounts:*

*- name: backup-volume*

*mountPath: /mnt/backup*

*restartPolicy: OnFailure*

*volumes:*

*- name: backup-volume*

*persistentVolumeClaim:*

*claimName: mysql-backup-pvc*

🖥️ **Apply the CronJob:**

*kubectl apply -f cronjob.yml*

**🧪 Test the CronJob Manually**

*kubectl create job --from=cronjob/mysql-backup-job mysql-backup-test*

Where:

* mysql-backup-job = name of the CronJob
* mysql-backup-test = name of the new one-time Job created from it

**✅ Useful Validation Commands**

*kubectl get pods*

*kubectl get pv*

*kubectl get pvc*

*kubectl describe pod <pod-name>*

**☁️ Verify on S3**

* Go to your S3 bucket
* Check if the .sql.gz file was uploaded under mysql/ directory

If it's not uploaded:

* Check logs using:

*kubectl logs <pod-name>*

Resolve any missing dependencies or upload issues.