

Build Cronjob

Bash Script to Export and Upload DB Backup

```
#!/bin/bash

# MySQL Backup and S3 Upload Script with s3cmd
# Usage: ./mysql_backup_and_upload.sh

# Configuration Variables (can be replaced by environment variables)
MYSQL_HOST="${MYSQL_HOST:-mysql-service}"
MYSQL_USER="${MYSQL_USER:-root}"
MYSQL_PASSWORD="${MYSQL_PASSWORD:-your-mysql-password}"
BACKUP_DIR="${BACKUP_DIR:-/mnt/introvesia_pvc01/dbs/backups}"
S3_BUCKET="${S3_BUCKET:-your-s3-bucket}"
RETENTION_DAYS="${RETENTION_DAYS:-7}"

# Logging function
log() {
    echo "[$(date +%Y-%m-%d %H:%M:%S)] $*"
}

# Ensure backup directory exists
mkdir -p "${BACKUP_DIR}"

# Generate timestamp for backup file
TIMESTAMP=$(date +%Y%m%d_%H%M%S)
BACKUP_FILE="${BACKUP_DIR}/mysql_backup_${TIMESTAMP}.sql"
COMPRESSED_BACKUP_FILE="${BACKUP_FILE}.gz"

# Create s3cmd configuration file
```

```
S3CMD_CONFIG="/tmp/s3cfg"
cat << EOF > "${S3CMD_CONFIG}"
[default]
access_key = ${AWS_ACCESS_KEY_ID}
secret_key = ${AWS_SECRET_ACCESS_KEY}
host_base = s3.amazonaws.com
host_bucket = %(bucket)s.s3.amazonaws.com
use_https = True
EOF

# Perform MySQL Backup
log "Starting MySQL backup..."
if ! mysqldump -h "${MYSQL_HOST}" -u "${MYSQL_USER}" -p"${MYSQL_PASSWORD}" --all-databases >
"${BACKUP_FILE}"; then
    log "ERROR: MySQL backup failed"
    exit 1
fi

# Compress the backup
log "Compressing backup file..."
if ! gzip "${BACKUP_FILE}"; then
    log "ERROR: Backup compression failed"
    exit 1
fi

# S3 Upload using s3cmd
log "Uploading backup to S3..."
# Create S3 path with daily subdirectory
S3_PATH="s3://${S3_BUCKET}/mysql-backups/$(date +"%Y-%m-%d")/"

# Upload to S3
if ! s3cmd -c "${S3CMD_CONFIG}" put "${COMPRESSED_BACKUP_FILE}" "${S3_PATH}"; then
    log "ERROR: S3 upload failed"
    exit 1
fi

# Cleanup: Remove local backups older than RETENTION_DAYS
log "Cleaning up old local backups..."
find "${BACKUP_DIR}" -name "*.sql.gz" -type f -mtime +${RETENTION_DAYS} -delete
```

```
# Log success
log "Backup and upload completed successfully"

# Remove temporary s3cmd config
rm "${S3CMD_CONFIG}"
```

Create Cronjob

```
apiVersion: v1
kind: Secret
metadata:
  name: mysql-backup-credentials
  namespace: db
type: Opaque
stringData:
  MYSQL_HOST: mysql-service
  MYSQL_USER: root
  MYSQL_PASSWORD: your-secure-password
  BACKUP_DIR: /mnt/introvesia_pvc01/dbs/backups
  S3_BUCKET: your-s3-bucket-name
  AWS_ACCESS_KEY_ID: your-aws-access-key
  AWS_SECRET_ACCESS_KEY: your-aws-secret-key
```

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: mysql-backup-and-upload
  namespace: db
spec:
  schedule: "0 2 * * *" # Run daily at 2 AM
  jobTemplate:
    spec:
      template:
        spec:
          restartPolicy: OnFailure
          containers:
```

```
- name: mysql-backup
image: mysql:8.0
command:
- /bin/sh
- -c
- |
# Install dependencies
apt-get update && apt-get install -y wget python3-pip

# Install s3cmd
pip3 install s3cmd

# Copy backup script to container
cp /scripts/mysql_backup_and_upload.sh /tmp/mysql_backup_and_upload.sh
chmod +x /tmp/mysql_backup_and_upload.sh

# Run backup script
/tmp/mysql_backup_and_upload.sh
env:
- name: MYSQL_HOST
valueFrom:
secretKeyRef:
name: mysql-backup-credentials
key: MYSQL_HOST
- name: MYSQL_USER
valueFrom:
secretKeyRef:
name: mysql-backup-credentials
key: MYSQL_USER
- name: MYSQL_PASSWORD
valueFrom:
secretKeyRef:
name: mysql-backup-credentials
key: MYSQL_PASSWORD
- name: BACKUP_DIR
valueFrom:
secretKeyRef:
name: mysql-backup-credentials
key: BACKUP_DIR
- name: S3_BUCKET
valueFrom:
```

```
    secretKeyRef:
      name: mysql-backup-credentials
      key: S3_BUCKET
  - name: AWS_ACCESS_KEY_ID
    valueFrom:
      secretKeyRef:
        name: mysql-backup-credentials
        key: AWS_ACCESS_KEY_ID
  - name: AWS_SECRET_ACCESS_KEY
    valueFrom:
      secretKeyRef:
        name: mysql-backup-credentials
        key: AWS_SECRET_ACCESS_KEY
  volumeMounts:
  - name: backup-storage
    mountPath: /mnt/introvesia_pvc01/dbs/backups
  - name: backup-script
    mountPath: /scripts
  volumes:
  - name: backup-storage
    hostPath:
      path: /mnt/introvesia_pvc01/dbs/backups
      type: Directory
  - name: backup-script
    configMap:
      name: mysql-backup-script
      defaultMode: 0755
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-backup-script
  namespace: db
data:
  mysql_backup_and_upload.sh: |
    #!/bin/bash

    # MySQL Backup and S3 Upload Script
    # (Same script as in the previous artifact)
```

... (paste the entire script from the previous artifact here)

Revision #1

Created 7 December 2024 09:29:19 by Ahmad

Updated 7 December 2024 14:23:25 by Ahmad