

IVS MySQL and MongoDB Backup Script & Steps

A. MongoDB

Steps

1. Create the file:

```
1 nano /home/pcs-ivs-server1/bin/ivs_mongo_backup.sh
```

2. ivs_mysql_backup.sh file:

```
1#!/bin/bash
2# MongoDB automatic backup script - runs every 6 hours
3# Creates one dump per day, compresses it, and deletes older ones
4
5BACKUP_DIR="/home/pcs-ivs-server1/IVS/mongo_backups"
6CONTAINER_NAME="mongodb_container"          # Your MongoDB container name (from docker ps)
7DB_NAME="ivs_auto_serving"
8USER="root"
9PASS="Rootmongo@123"
10AUTH_DB="admin"                         # Use 'admin' if MONGO_AUTH_DB_NAME is empty
11DATE=$(date +'%Y-%m-%d')
12DUMP_FILE="$BACKUP_DIR/ivs-mongo-dump-$DATE.gz"
13
14mkdir -p "$BACKUP_DIR"
15
16# Avoid duplicate backups for same day
17if [ -f "$DUMP_FILE" ]; then
18    echo "$(date): Backup already exists for today ($DUMP_FILE). Skipping new dump."
19    exit 0
20fi
21
22echo "$(date): Starting MongoDB backup..."
23
24# Run mongodump inside the Docker container
25docker exec "$CONTAINER_NAME" mongodump \
26    --host mongodb \
27    --port 27017 \
28    --db "$DB_NAME" \
29    --username "$USER" \
30    --password "$PASS" \
31    --authenticationDatabase "$AUTH_DB" \
32    --archive --gzip > "$DUMP_FILE"
33
34# Check result
35if [ $? -eq 0 ]; then
36    echo "$(date): Backup successful: $DUMP_FILE"
37    # Delete older backups, keep only today's
38    find "$BACKUP_DIR" -type f -name "ivs-mongo-dump-*.*" ! -name "ivs-mongo-dump-$DATE.*" -delete
39else
40    echo "$(date): Backup failed!"
41    rm -f "$DUMP_FILE"
42fi
43
```

3. Execute the file:

```
1 chmod +x /home/pcs-ivs-server1/bin/ivs_mongo_backup.sh
```

4. Add cron job (every 6 hours)

```
1 crontab -e
```

```
1 0 */6 * * * /home/pcs-ivs-server1/bin/ivs_mongo_backup.sh >> /home/pcs-ivs-server1/IVS/mongo_backups/ivs-mongo-backup.log 2>&1
```

5. Test it manually

```
1 sudo bash /home/pcs-ivs-server1/bin/ivs_mongo_backup.sh
```

6. Check Logs:

```
1 ls /home/pcs-ivs-server1/IVS/mongo_backups/
2 cat /home/pcs-ivs-server1/IVS/mongo_backups/ivs-mongo-backup.log
```

B. MySQL

Steps:

1. Create credentials file:

```
1 nano ~/.my.cnf
```

2. Paste this:

```
1 [client]
2 user = root
3 password = Rootmysql@123
4 host = 127.0.0.1
```

3. Secure the file

```
1 chmod 600 ~/.my.cnf
```

4. Make backup and bin directories

```
1 mkdir -p /home/pcs-ivs-server1/IVS/mysql_backups
2 mkdir -p /home/pcs-ivs-server1/bin
```

5. Create the backup script

```
1 nano /home/pcs-ivs-server1/bin/ivs_mysql_backup.sh
```

6. Paste the following script

```
1 #!/bin/bash
2 # MySQL automatic backup script for Docker - runs every 6 hours
3 # Creates one dump per day and deletes older ones
4
5 BACKUP_DIR="/home/pcs-ivs-server1/IVS/mysql_backups"
6 DB_NAME="ivs_auto_serving"
7 CONTAINER_NAME="mysql_container"    # ← Replace with your actual container name
8 DATE=$(date '+%Y-%m-%d')
9 DUMP_FILE="$BACKUP_DIR/ivs-mysql-dump-$DATE.sql"
10
11 # Create backup directory if not exists
12 mkdir -p "$BACKUP_DIR"
13
14 # Check if today's backup already exists
15 if [ -f "$DUMP_FILE" ]; then
16     echo "$(date): Backup already exists for today ($DUMP_FILE). Skipping new dump."
17     exit 0
18 fi
19
20 # Run mysqldump inside the Docker container
21 docker exec "$CONTAINER_NAME" mysqldump -u root -pRootmysql@123 -h mysql "$DB_NAME" > "$DUMP_FILE"
22
23 # Check the result of the dump command
24 if [ $? -eq 0 ]; then
25     echo "$(date): Backup successful: $DUMP_FILE"
26     # Delete previous backups except today's
27     find "$BACKUP_DIR" -type f -name "ivs-mysql-dump-*sql" ! -name "ivs-mysql-dump-$DATE.sql" -delete
28 else
29     echo "$(date): Backup failed!"
30     rm -f "$DUMP_FILE"
31 fi
```

7. Make it executable

```
1 chmod +x /home/pcs-ivs-server1/bin/ivs_mysql_backup.sh
```

8. Test Backup Manually

```
1 bash /home/pcs-ivs-server1/bin/ivs_mysql_backup.sh
```

9. Add Cron Job (Every 6 Hours)

```
1 crontab -e
```

```
1 0 */6 * * * /home/pcs-ivs-server1/bin/ivs_mysql_backup.sh >> /home/pcs-ivs-server1/IVS/mysql_backups/ivs-mysql-backup.log 2>&1
```