

Backup and Restore Cisco ONP Database

- Backup Database, on page 1
- Restore database using script, on page 5

Backup Database

The database backup configuration on the server, set by the system administrators, allows backups of the entire database to be taken daily, weekly, or monthly. Only system administrators have access to the backup folders and can perform restorations. Backup archives are stored on a highly durable NFS share. The available backup archives include:

- Daily archives for the last 7 days
- Weekly archives for the last 4 weeks
- Monthly archives for the last 6 months

Note You can change the duration for monthly archives.

Setup Backup

Before you begin

- Ensure that a highly durable 2 TB NFS storage is available to store database archives.
- Ensure that the storage is accessible from the Cisco ONP environment.
- Set up rules as defined by the Cisco ONP network and configure the appropriate Access Control List (ACL) to permit or deny access to the storage as needed.

Procedure

Step 1 Log into the Cisco ONP server, where you have installed Cisco ONP by using the tar archive.

Step 2 Create a folder for backup. Ideally it should be mounted on a reliable object storage such as NFS.

Step 3 Copy `restore_mongo.sh` from `/ONP/images/`.

Perform manual backup

Follow these steps to back up the Cisco ONP database.

Procedure

Step 1 Browse to the location where the backup_mongo.sh script is stored.

Step 2 Use the `bash backup_mongo.sh` command to execute the script.

Example:

```
user@server:~/cnp_backups$ bash backup_mongo.sh
```

The backup file is stored in the daily backup folder (backup.daily) with the current time stamp.

Schedule a backup

You can schedule a backup in either of these two ways:

- Schedule a backup using script, on page 2
- Schedule a backup, on page 3

Schedule a backup using script

Follow these steps to schedule the backup operation using a script.

Procedure

Step 1 Log in to the Cisco ONP server.

Step 2 Use the command `bash rotation.sh` to execute the rotation.sh script.

Example:

```
user@server:~/cnp_backups$ bash rotation.sh
```

This command creates the necessary folder structure.

Step 3 Use the command `bash backup_postgres.sh` to execute the backup_postgres.sh script.

Example:

```
user@server:~/cnp_backups$ backup_postgres.sh
```

The backup file is stored in the daily backup folder (backup.daily) with the current time stamp.

Step 4 Enter the path at the user prompt to store the file.

Backup files are stored in your required file path as shown in this example.

Example:

```
/user-defined path/backups/backup.daily
```

Schedule a backup

Follow these steps to schedule the backup operation using Crontab.

Procedure

Step 1 Log in to the Cisco ONP server.

Step 2 Use the command `sudo crontab -e` to configure the Crontab job.

```
user@server~/cnp_backups$ sudo crontab -e
```

Step 3 Update the `Crontab.sh`script with the required cron job entries using the Syntax to schedule the backup using Crontab, on page 5, to schedule backup operations.

Example:

This example shows the scheduling of the backup operation at 3:00 a.m daily.

```
0 3 * * * /home/ciscocnp/cnp_backups/backup_mongo.sh
```

Syntax to schedule the backup using Crontab

Use this syntax to schedule a backup in Crontab:

```
m h dom mon dow
```

Where:

- m: Minutes (0–59)
- h: Hours (1–23)
- dom: Day of the month (1–31)
- mon: Month of the year (1–12)
- dow: Day of the week (0–6, where 0 represents Sunday)

To specify the time, you can provide exact values for the minute (m), hour (h), day of the month (dom), month (mon), and day of the week (dow). Alternatively, you can use the wildcard character `*` in these fields to indicate "any value."

Note The backup jobs are initiated according to the system daemon's understanding of time and time zones as defined by the cron scheduler.

Restore the database manually

Follow these steps to restore the database manually.

Procedure

Step 1 Check the Cisco ONP Database.

Example:

```
$ mongo CnpDB

MongoDB shell version: 2.6.10
 connecting to: CnpDB
 Server has startup warnings:
 2019-01-31T12:17:39.354+0000 I STORAGE [initandlisten]
 2019-01-31T12:17:39.354+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is
strongly recommended with the WiredTiger storage engine.
 2019-01-31T12:17:39.354+0000 I STORAGE [initandlisten] ** See
http://dochub.mongodb.org/core/prodnotes-filesystem
 2019-01-31T12:17:40.013+0000 I CONTROL [initandlisten]
 2019-01-31T12:17:40.013+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled
for the database.
 2019-01-31T12:17:40.013+0000 I CONTROL [initandlisten] ** Read and write access to data and
configuration is unrestricted.
 2019-01-31T12:17:40.013+0000 I CONTROL [initandlisten]
 > show collections
 NetworkEntity
 SystemLog
 BOM
 fs.files
 Users
 fs.chunks
 ErrorInfo
 NetworkObject
 UserGroups
 TaskInfo
 ActionGroups
 Roles
 >
 ```
```

**Step 2** Remove the Cisco ONP back-end and front-end Docker services,"docker service rm cnp_cnp cnp_cnp_frontend" and check again:

**Example:**

```
```

$ mongo CnpDB

 MongoDB shell version: 2.6.10
 connecting to: CnpDB
 Server has startup warnings:
 2019-01-31T12:17:39.354+0000 I STORAGE [initandlisten]
 2019-01-31T12:17:39.354+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS file system is
strongly recommended with the WiredTiger storage engine.
 2019-01-31T12:17:39.354+0000 I STORAGE [initandlisten] ** See
http://dochub.mongodb.org/core/prodnotes-filesystem
 2019-01-31T12:17:40.013+0000 I CONTROL [initandlisten]
 2019-01-31T12:17:40.013+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled
for the database.
```

```
 2019-01-31T12:17:40.013+0000 I CONTROL [initandlisten] ** Read and write access to data and
configuration is unrestricted.
 2019-01-31T12:17:40.013+0000 I CONTROL [initandlisten]
 > show collections
 >
 ```
```

**Step 3** Copy the database to the folder that is mapped to MongoDB container's volume:

**Example:**

```
 $ sudo cp backups/2019-02-01-04-29.archive /var/lib/mongodb_cnp/
```

Step 4 Restore the database and check.

Example:

```
 $ docker exec -t cnp_mongo.1.$(docker service ps -f 'name=cnp_mongo.1' -f 'desired-state=running'
cnp_mongo -q) bash -c "mongorestore -u \$(cat \$MONGO_INITDB_ROOT_USERNAME_FILE) -p \$(cat
\$MONGO_INITDB_ROOT_PASSWORD_FILE) --authenticationDatabase admin --port 27017 -h localhost --db
CnpDB --gzip --archive=/data/db/2019-02-01-04-29.archive"
```

Restore database using script

You can restore the database using a script, which is available at this location:

`Images/DB_backup/restore_mongo.sh`

Follow these steps to restore the database using a script:

Procedure

Use the command `sudo bash restore_mongo.sh` to restore the database

Example:

```
 $ sudo bash restore_mongo.sh DB backup archive file path.
$ sudo bash restore_mongo.sh /backups/backup.daily/2019-05-05-23-58.archive
```

The database backup is created, and a confirmation prompt appears that asks whether to delete the database. `(y/n)`.

Enter `y` to delete the database.

Enter `n` to not delete the database.

Enter `No` to not delete the database and apply only missing data(s) from backup file.

Syntax to schedule the backup using Crontab

Use this syntax to schedule a backup in Crontab:

```
m h dom mon dow
```

Where:
```

- m: Minutes (0-59)
- h: Hours (1-23)
- dom: Day of the month (1-31)
- mon: Month of the year (1-12)
- dow: Day of the week (0-6, where 0 represents Sunday)

To specify the time, you can provide exact values for the minute (m), hour (h), day of the month (dom), month (mon), and day of the week (dow); alternatively, you can use the wildcard character * in these fields to indicate "any value."



Note The backup jobs are initiated according to the system daemon's understanding of time and time zones as defined by the cron scheduler.