

# Backup Using Cron Job

## What are cron jobs in Linux?

Any task that you schedule through crons is called a cron job. Cron jobs help us automate our routine tasks, whether they're hourly, daily, monthly, or yearly.

## Basic command For Cronjob

- **crontab -e**: edits crontab entries to add, delete, or edit cron jobs.
- **crontab -l**: list all the cron jobs for the current user.
- **crontab -u username -l**: list another user's crons.
- **crontab -u username -e**: edit another user's crons.

## Basic Syntax For Cronjob

```
* * * * * sh /path/to/script/script.sh
```

Diagram illustrating the cron job syntax fields and their corresponding ranges:

- Min(0-59)
- Hour(0-23)
- Day of the Month(1-31)
- Month of the Year(1-12)
- Day of the Week(0-6)
- Command or Script to Execute

For Fixing a proper time you can visit:-  
<https://crontab.guru/>

## Backup of postgresql Using Cronjob

Step I :- crontab -e

```
GNU nano 4.8 /tmp/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow  command
```

Step II :- 0 5 \* \* 1 pg\_dump "host=localhost port=5432  
user=postgres password=Your\_db\_Password dbname=db\_name"  
> /home/bibek/Desktop/db\_name\_dump.sql

**Explanation** → Taking backup at every week 5 a.m

## Backup of Mysql Using Cronjob

Step I:- create a .my.cnf file in a home directory

Step II:- Edit .my.cnf file like this

```
[client]
user = root
password = "dbpassword"
host = localhost
```

Step III:- create a jobs using crontab -e like this

```
33-37 * * * * mysql_dump "host=localhost user=root password=Carlos@123. customer"
30-33 * * * * mysqldump --routines customer > /home/bibek/Desktop/backup.sql
```

## Dynamic PostgreSQL Backup using Cronjob

Step I:- Create a bash script like this

```
#!/bin/bash

pg_dump "host=localhost port=5432 user=postgres password=Carlos@123. dbname=hr"
> /home/bibek/Desktop/hr_dump$(date +%b_%d_%y_%M).sql
```

Step II:- create a jobs using crontab -e like this

```
32-35 * * * * sh /home/bibek/Desktop/test.sh
```

**Explanation:-** Create a backup at every 32 and 35 min  
using name like `hr_dump_May_05_22_48.sql`

Now files looks like this,

