mysql -u root -p

##############################################################################################################################################################

using this cmd we will sync our directory to remote directory

rsync -avhe ssh --chown=USER:GROUP /tmp/\*.sh user@remote-host:/tmp/

here we will exciute the pre-requests script using jenkins with all necessary file exciution permissionns

after exicuting this

the code is depolyed

after the post script is exciuted

after that here we will exicute the cmd like removing particular directory and files in it

rm -rf /tmp/\*.sh

#############################################################################################################################################

################################# crontab format ##########################

\* \* \* \* \* /path/to/the/script

| | | | | \_ \_ \_ \_ \_ \_ Day of the week (0-7)

| | | | \_ \_ \_ \_ \_ \_ \_ Day of the year (1-12)

| | |\_ \_ \_ \_ \_ \_ \_ Day of the month ( 1-31)

| |\_ \_ \_ \_ \_ \_ \_ Hour (0 -23 )

|\_ \_ \_ \_ \_ \_ \_ \_ Minute (0 -59)

How to Add/Modify Crontab

User can edit their crontab jobs with the help of following crontab command:

$ crontab -u -e

The above command will open the personal crontab configuration of your computer system, which can be edited by using your default text editor.

There is no need to restart your crontab as it will pick up your changes automatically when you use following command.

$ crontab -l

To remove your crontab tasks, use the following command.

$ crontab -r

To add or update job in crontab, use below given command.

crontab -e

Command to edit other user’s crontab

crontab -u username -e

How to List Crontab

Command to view crontab entries of current user

crontab -l

Command to view crontab entries of a specific user:

crontab -u username -l

Important Crontab Examples

Here, are some important examples of Crontab

Description Command

Cron command to do the various scheduling jobs.

1) Below given command execute at 7 AM and 5 PM daily.

0 7,17 \* \* \* /scripts/script.sh

2) Command to execute a cron after every 5 minutes.

\*/5\* \* \* \* \* /scripts/script.sh

3) Cron scheduler command helps you to execute the task on every Monday at 5 AM. This command is helpful for doing weekly tasks like system clean-up.

0 5 \* \* mon /scripts/script.sh

4) Command run your script on 3 minutes interval.

\*/3 \* \* \* \* /scripts/monitor.sh

5) Command to schedule a cron to which executes for a specific month.

This command to run tasks run in Feb, June and September months. Sometimes we need to schedule a task to execute a select monthly task.

\* \* \* feb,jun,sep \* /script/script.sh

6) Command to execute on selected days. This example will run each Monday and Wednesday at 5 PM.

0 17 \* \* mon,wed /script/script.sh

7) This command allows cron to execute on first Saturday of every month.

0 2 \* \* sat [ $(date +%d) -le 06 ] && /script/script.sh

8) Command to run a script for 6 hours interval so it can be configured like below.

0 \*/6 \* \* \* /scripts/script.sh

9) This command schedule a task to execute twice on Monday and Tuesday. Use the following settings to do it.

0 4,17 \* \* mon,tue /scripts/script.sh

10) Command schedule a cron to execute after every 15 Seconds.

\* \* \* \* \* /scripts/script.sh

\* \* \* \* \* sleep 15; /scripts/script.sh

11) Command to schedule tasks on a yearly basis.

@yearly timestamp is= to “0 0 5 1 \*”. This executes the task in the fifth minute of every year. You can use it to send for new year greetings.

@yearly /scripts/script.sh

12) Command tasks to execute on a monthly basis.

@monthly timestamp is similar to “0 0 1 \* \*”. This command expression allows the execution of a task in the first minute of the month.

@monthly /scripts/script.sh

Command to execute multiple tasks using a single cron.

\* \* \* \* \* /scripts/script.sh; /scripts/scrit2.sh

Command to schedule tasks to execute on a weekly basis.

@weekly timestamp is similar to “0 0 4 \* sun”. This is used to perform the weekly tasks like the system cleanup etc.

@weekly /bin/script.sh

Task will be scheduled to execute on a daily basis.

@daily timestamp is similar to “0 2 \* \* \*”. It executes the task in the second minute of every day.

@daily /scripts/script.sh

Allows tasks to execute on an hourly.

@hourly timestamp is similar to “0 \* \* \* \*”. This command executes a task in the first minute of every hour.

@hourly /scripts/script.sh

Allows tasks to execute on system reboot.

@reboot expression is useful for those tasks that the system wants to run on your system startup. This is helpful to begin tasks background automatically.

@reboot /scripts/script.sh]

Summary:

Cron is named after Greek word “Chronos” that is used for time.

Cronjobs help OS to take a scheduled backup of log files or database.

Linux system pack has a useful task scheduler named crontab that can be scheduled to run an automated process as root.

Crontab of Linux has six fields. The first five fields define the time and date of execution, and the 6’th field is used for command execution.