**Crontab Set-up**

**Crontab**

**Crontab command is used to schedule commands to be executed periodically. To see what crontabs are currently running on your system, you can open a terminal and run:**

sudo crontab -l

**To edit the list of cronjobs you can run:**

sudo crontab -e

**Cronjobs are written in the following format:**

\* \* \* \* \* /bin/execute/this/script.sh

**Scheduling explained**

**There are 5 asterisks. The asterisks represent different date parts in the following order:**

1. minute (from 0 to 59)

2. hour (from 0 to 23)

3. day of month (from 1 to 31)

4. month (from 1 to 12)

5. day of week (from 0 to 6) (0=Sunday)

**Execute every minute**

**To execute every minute leave all asterisks**

\* \* \* \* \* /bin/execute/this/script.sh

every minute, of every hour, of every day of the month, of every month, and every day in the week

**Execute specific time**

**To schedule the script to run at 1AM every Friday, we would need the following cronjob:**

0 1 \* \* 5 /bin/execute/this/script.sh

1. minute: 0

2. of hour: 1

3. of day of month: \* (every day of month)

4. of month: \* (every month)

5. and weekday: 5 (=Friday)

**To schedule the script to Monday till Friday at 1 AM, we would need the following cronjob:**

0 1 \* \* 1-5 /bin/execute/this/script.sh

**10 past after every hour on the 1st of every month:**

10 \* 1 \* \* /bin/execute/this/script.sh

**To to run the script every 10 minutes:**

0,10,20,30,40,50 \* \* \* \* /bin/execute/this/script.sh

**But crontab allows you to do this as well:**

\*/10 \* \* \* \* /bin/execute/this/script.sh

**Special words**

**It is possible to use special words:**

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Definition** | **Value** |
| @reboot | Run once, at startup |  |
| @yearly | Run once a year | 0 0 1 1 \* |
| @annually | (same as @yearly) |  |
| @monthly | Run once a month | 0 0 1 \* \* |
| @weekly | Run once a week | 0 0 \* \* 0 |
| @daily | Run once a day | 0 0 \* \* \* |
| @midnight | (same as @daily) |  |
| @hourly | Run once an hour | 0 \* \* \* \* |

**Storing the crontab output**

**To save the output in a separate logfile:**

Linux can report on different levels. There's standard output (STDOUT) and standard errors (STDERR). STDOUT is marked 1, STDERR is marked 2.

So the following statement tells Linux to store STDERR in STDOUT as well, creating one datastream for messages & errors: 2>&1.

\*/10 \* \* \* \* /bin/execute/this/script.sh 2>&1 >> /var/log/script\_output.log

**Shell script tips**

**Important**

**To improve Shell script file security add at start:**

#!/bin/bash

**Java variables set-up**

**To run Jmeter \*.jmx file in the Shell script file \*.sh Java variables should be exported including Java location path: JAVA\_HOME=/usr/java/home/**

**export JAVA\_HOME**

**export PATH=$PATH:/usr/java/home/bin/**

**Jmeter script running**

**To make Jmeter file executed by Shell script file \*.sh full paths of jmeter and executed \*.jmx script should be indicated:**

/execute/bin/jmeter -n -t /execute/bin/example.jmx -S /common/properties\_name.properties -Jthreads\_quantity=100

Where:

-n = non gui mode

-t ./..\*jmx = test plan (case)

-S **/** = properties file

-J = (without space) property name & property value