***About cron /crontab***

The cron [daemon](https://kb.iu.edu/d/aiau) is a long-running process that executes commands at specific dates and times. You can use this to schedule activities, either as one-time events or as recurring tasks.

For commands that need to be executed repeatedly (e.g., hourly, daily, or weekly), you can use the crontabcommand. The crontab command creates a crontab file containing commands and instructions for the cron daemon to execute. You can use the crontab command with the following options:

|  |  |
| --- | --- |
| crontab -a filename | Install filename as your crontab file. On many systems, this command is executed simply as crontab filename (i.e., without the -a option). |
| crontab -e | Edit your crontab file, or create one if it doesn't already exist. |
| crontab -l | Display your crontab file. |
| crontab -r | Remove your crontab file. |
| crontab -v | Display the last time you edited your crontab file. (This option is available on only a few systems.) |
| crontab -u user | Used in conjunction with other options, this option allows you to modify or view the crontab file of user. When available, only administrators can use this option. |

Each entry in a crontab file consists of six fields, specifying in the following order:

minute(s) hour(s) day(s) month(s) weekday(s) command(s)

The fields are separated by spaces or tabs. The first five are integer patterns and the sixth is the command to execute. The following table briefly describes each of the fields:

| **Field** | **Value** | **Description** |
| --- | --- | --- |
| minute | 0-59 | The exact minute that the command sequence executes |
| hour | 0-23 | The hour of the day that the command sequence executes |
| day | 1-31 | The day of the month that the command sequence executes |
| month | 1-12 | The month of the year that the command sequence executes |
| weekday | 0-6 | The day of the week that the command sequence executes (Sunday = 0, Monday = 1, Tuesday = 2, and so forth) |
| command | Special | The complete sequence of commands to execute. The command string must conform to Bourne [shell](https://kb.iu.edu/d/agvf) syntax. Commands, executables (such as scripts), or combinations are acceptable. |

Each of the patterns from the first five fields may be either \* (an asterisk), meaning all legal values, or a list of elements separated by commas. An element is either a number or an inclusive range, indicated by two numbers separated by a minus sign (e.g., 10-12).

You can specify days with two fields: day of the month and day of the week. If you specify both of them as a list of elements, cron will observe both of them, for example:

0 0 1,15 \* 1 /mydir/myprogram

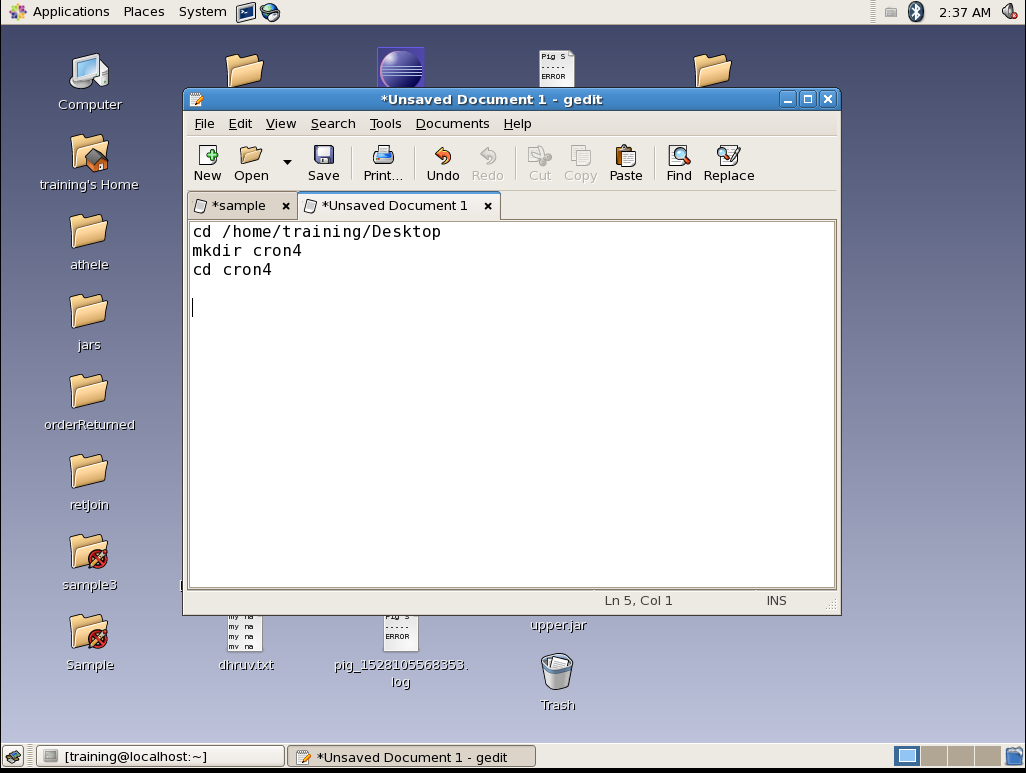
The cron daemon would run the program myprogram in the mydir directory on the first and fifteenth of each month, as well as on every Monday. To specify days by only one field, the other field should be set to \*, for example:

0 0 \* \* 1 /mydir/myprogram

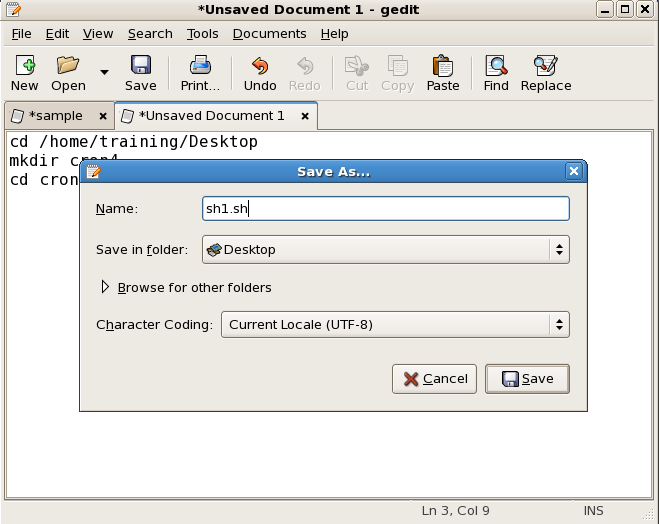
In the above example, the program would run only on Mondays.

***Running a basic cron job in Linux.***

1. To run a cron job we first should have a XXXXX.sh file which should contain commands  that need to be executed on a particular time.
2. Open text editor and create a new file with .sh extension and add the commands need to be executed. (follow the screenshots)



Commands are added to the file and saved with .sh extension



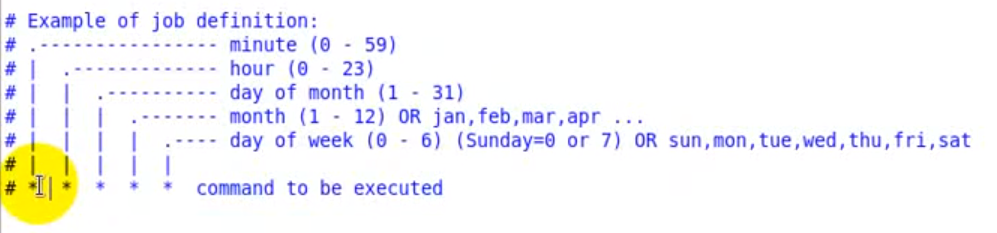
The sh1.sh file is created. Rounded in the figure above.

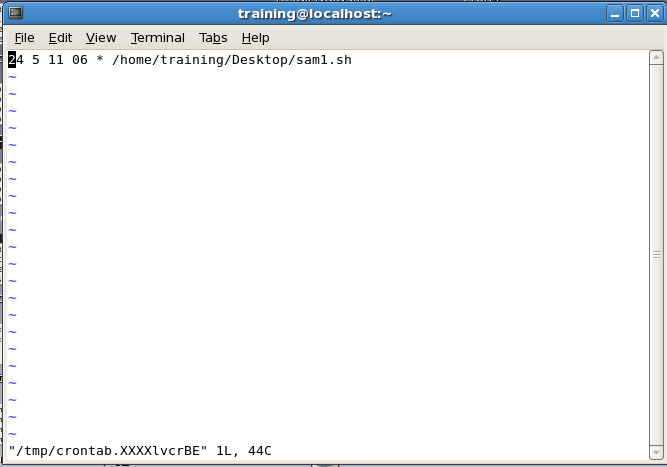
1. For making sh1.sh run on crontab we need to give permission to the file with the command below

*chmod 777 sh1.sh*

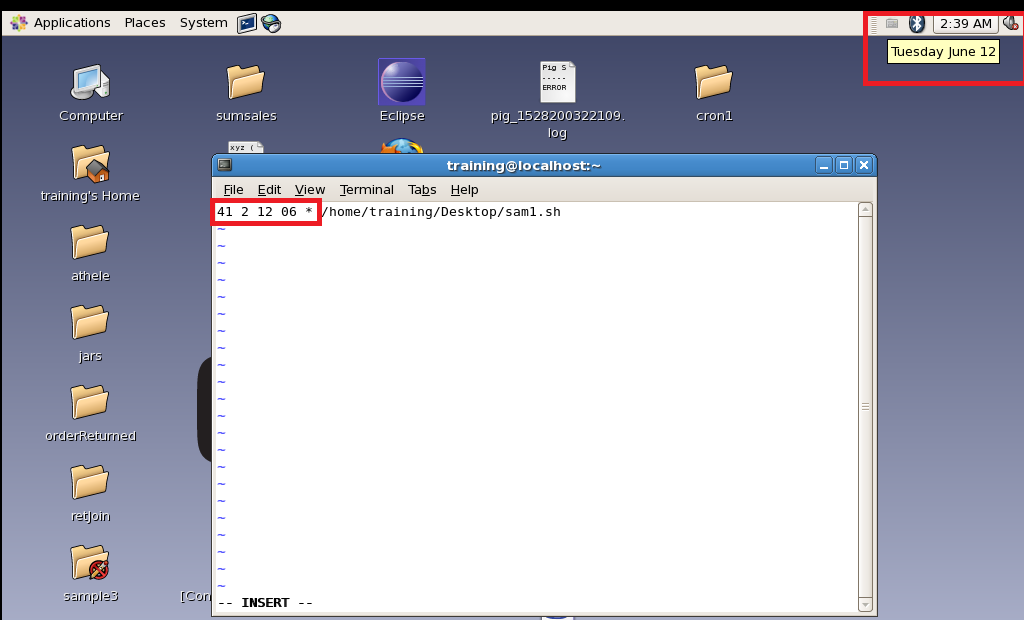


1. Open the crontab, by *crontab -e* command.

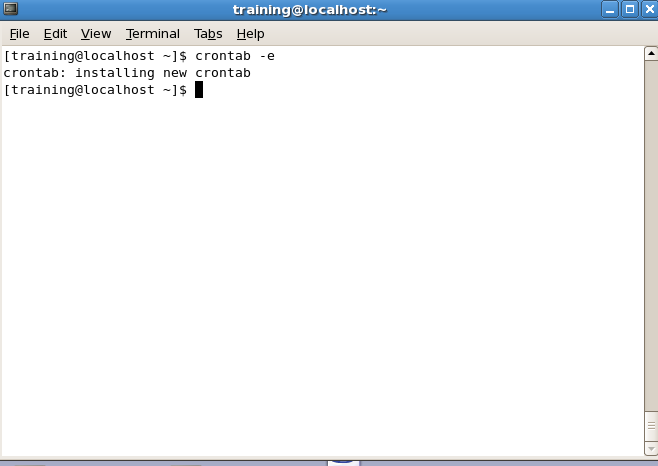




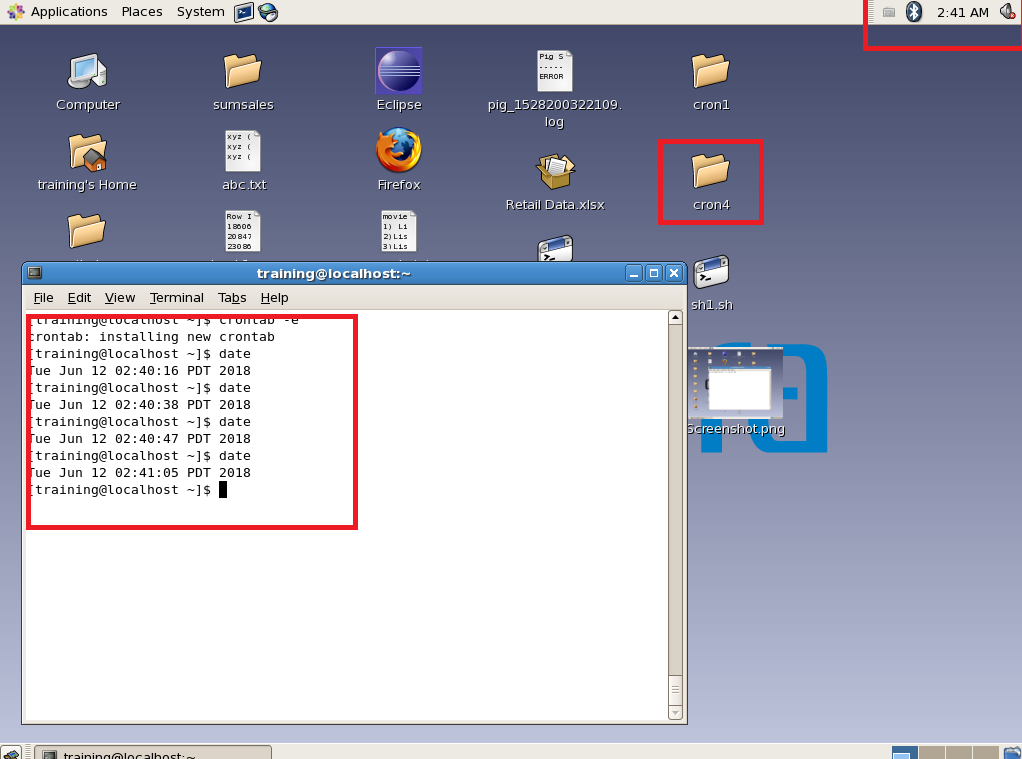
1. Edit the crontab file according to time of your LINUX OS (it can be different from windows OS ) and put the .sh file path(or you can put command directly) which you want to execute. As done below.



1. Make it sure that time you configured should be 2 – 3 mins future than the present time.

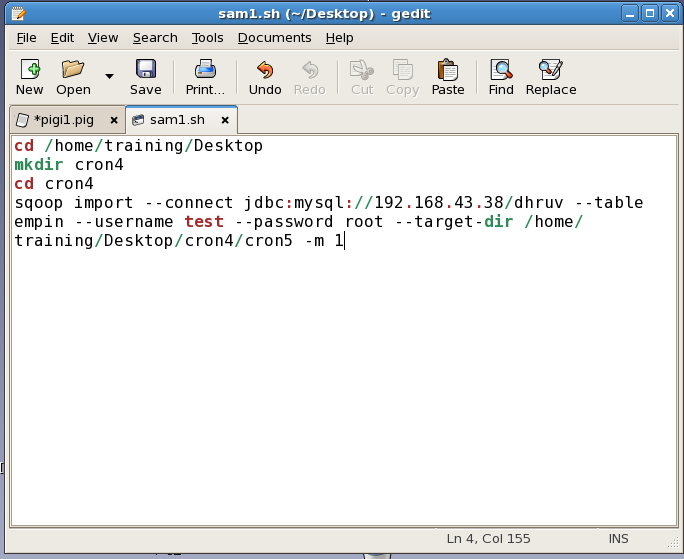


1. After saving your crontab file you will get a message “installing new crontab”.
2. Now wait till the time you have given in the crontab and check the changes according to your commands.



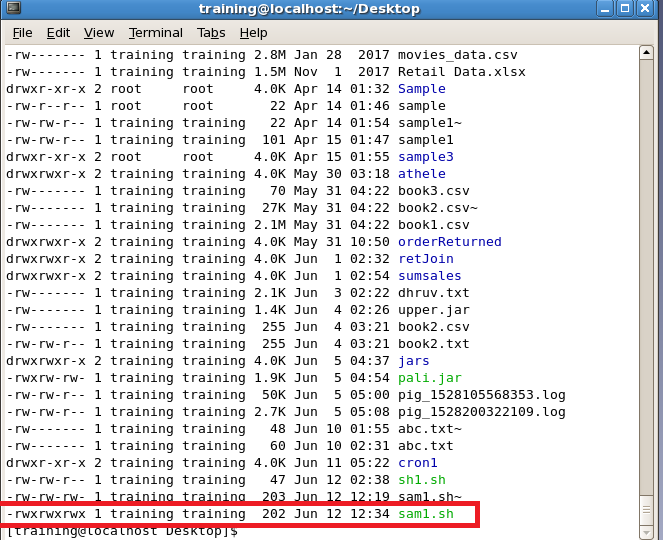
***Cron job for SQOOP***

1. for making a SQOOP cron job we just have to follow similar steps as done for making basic cron job.
2. Make the .sh file with Sqoop commands for ex. See the screenshot.

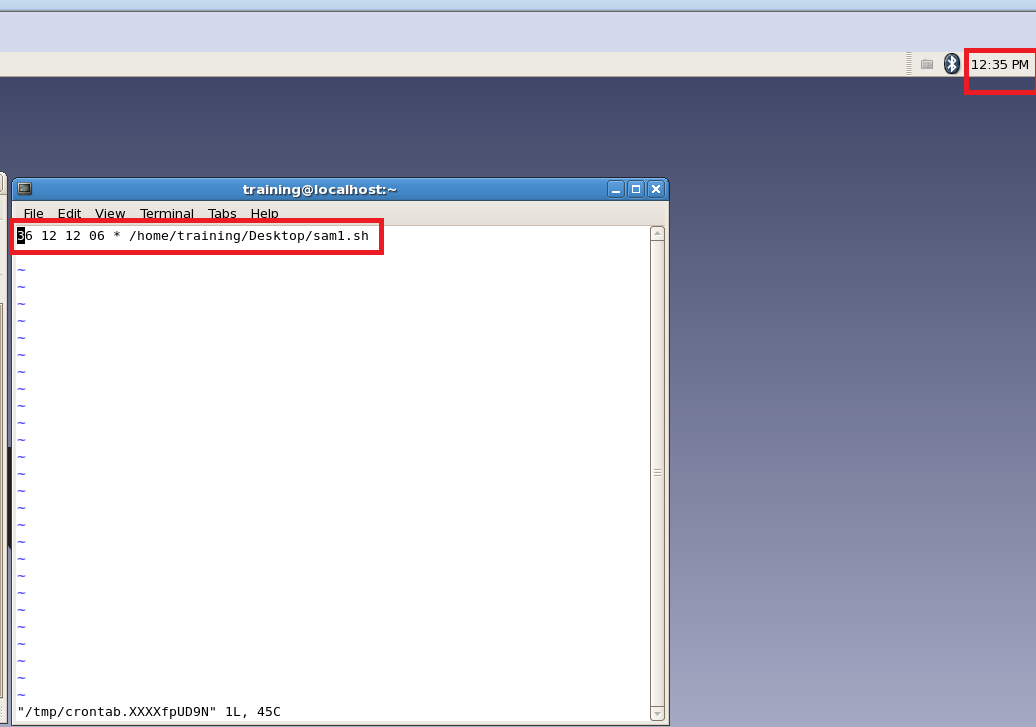


1. For making sam1.sh run on crontab we need to give permission to the file with the command below

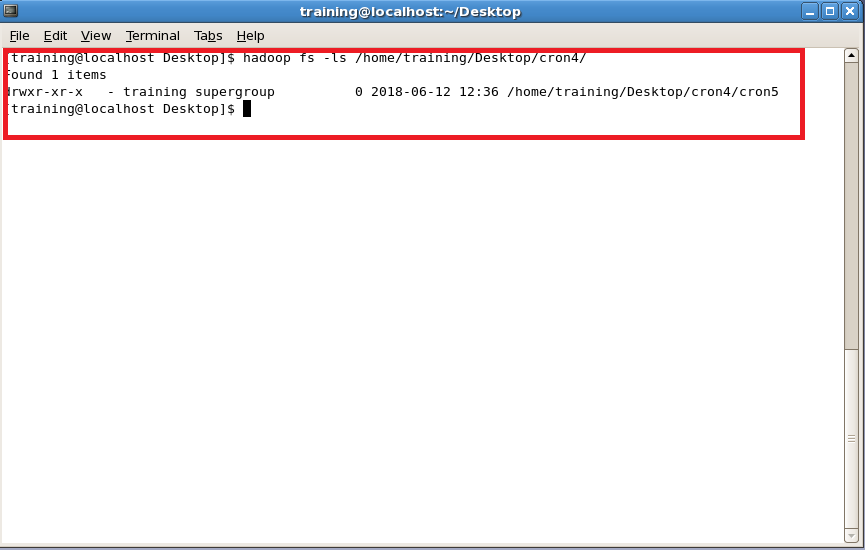
*chmod 777 sam1.sh*

**

1. Now follow the same steps(4,5,6,7,8) and after the job time you can check the results by listing the files in the sqoop imported directory.(kindly go through the below screenshots)



Crontab is edited for the next scheduled job for sam1.sh



In the above screenshot you can see the result of the Sqoop import is saved in cron4 folder.