JOB SCHEDULING IN LINUX

Linux is an open-source kernel for operating systems. It is widely used in operating systems such as Ubuntu, Debian, etc. One of the unique features of Linux is that it allows the user to have a higher degree of control over the machine. We can execute and program a vast number of things into our computer using simple commands.

Scheduling processes and commands is one of these things. If we wish to schedule a particular script to run next Monday, how would we do so? This is what we will look at in our article. There are 2 basic commands that we can use for this - at and crontab.

At Command

The **at** command can be used to schedule a job to run at a specific time. We can send reminders, execute system updates, and take backups at our desired time. We can specify a specific time and date (such as 12th October at 12 AM) or an interval (such as after 3 hours).

Installation:

at needs to be installed first. We can use **apt-get** to install it. At is not always installed by default on all Unix-like systems. Run the command:

sudo apt-get install at

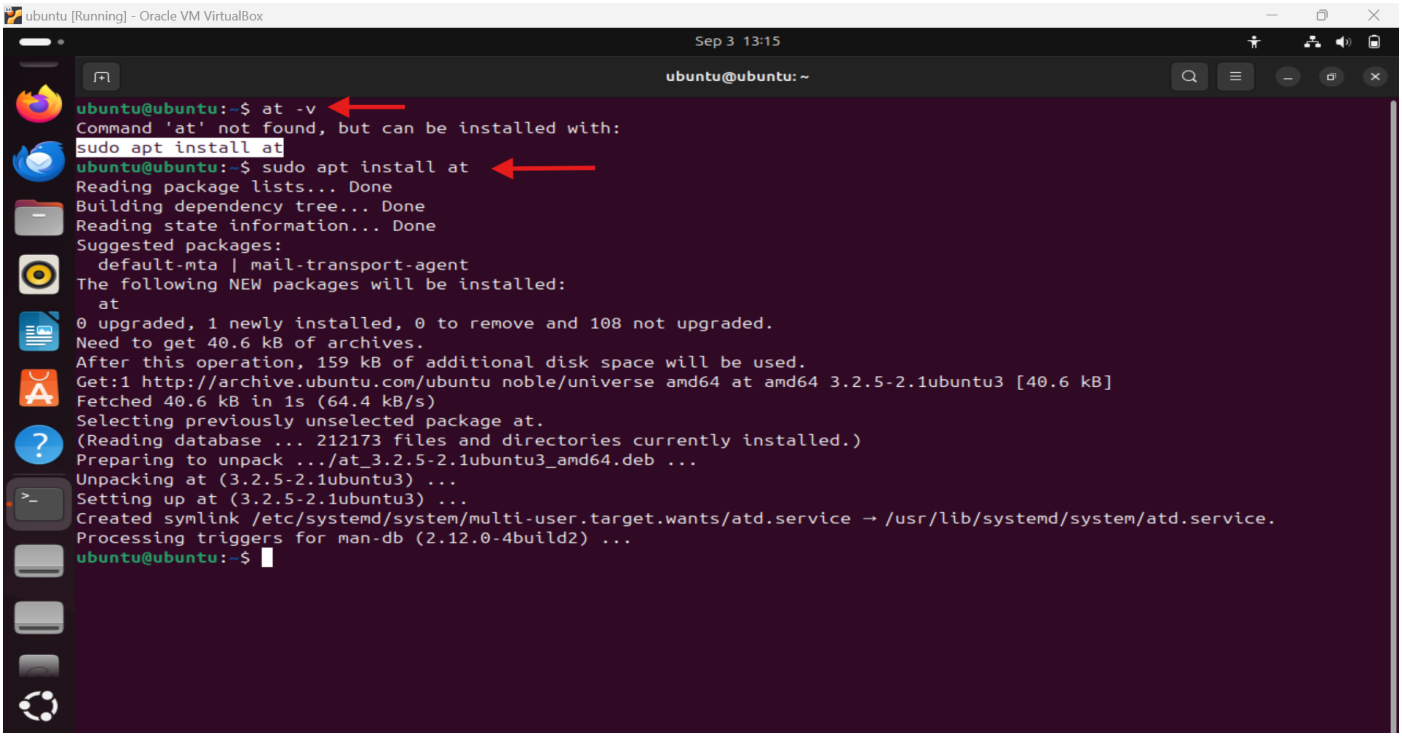
atq : To list all the jobs currently pending

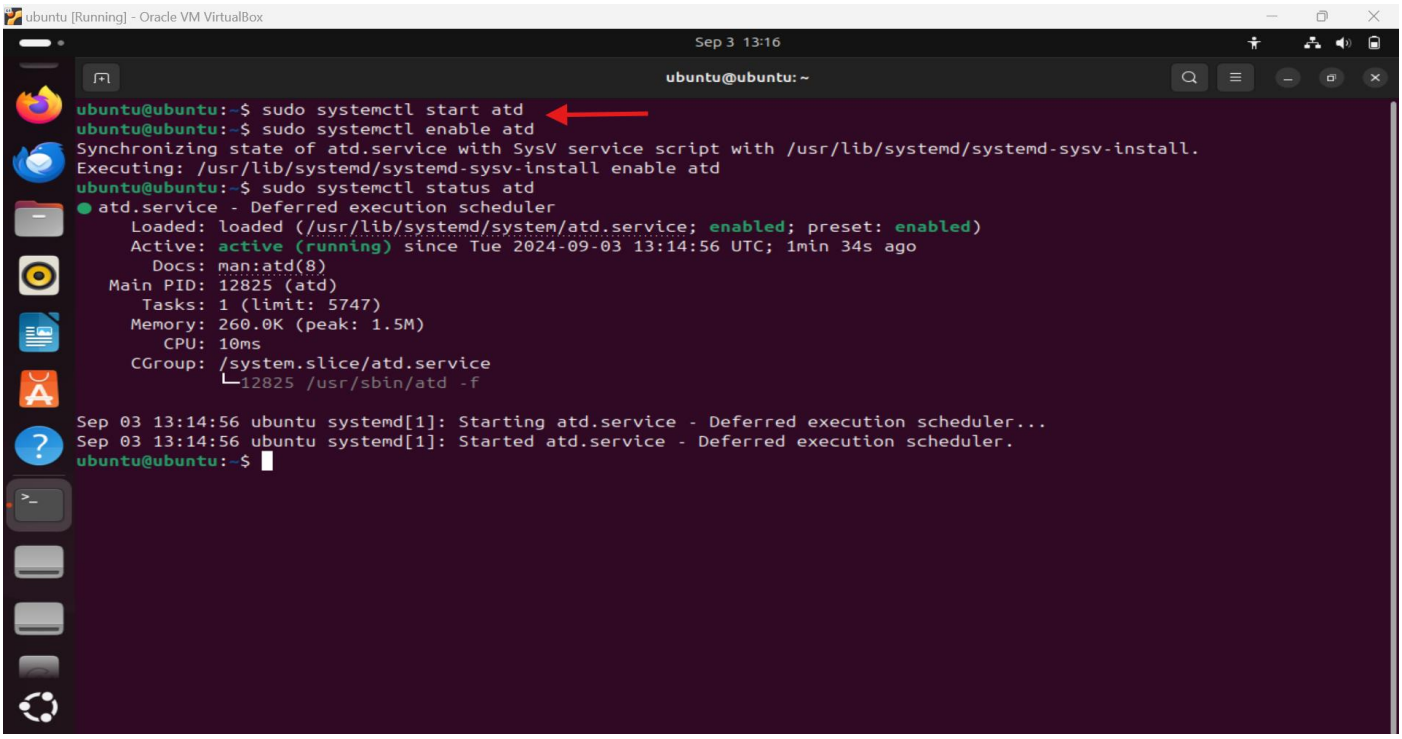
atrm :atrm is used to delete a pending job.

Once it is installed, we must start it and enable it. We can do so by running the commands:

sudo systemctl start atd

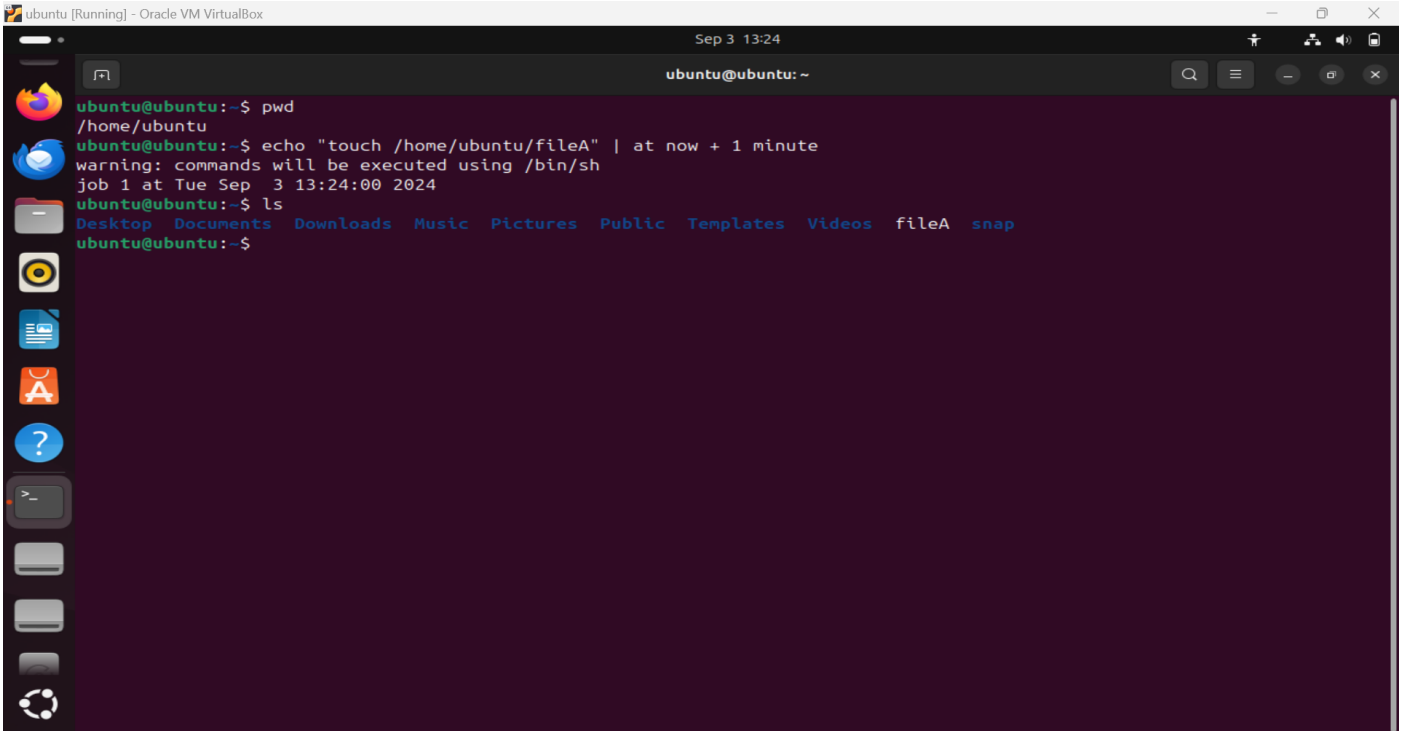
sudo systemctl enable atd





To schedule the creation of an empty file, use the at command. For example, to create the file one minute from now:

- **echo "touch /path/to/yourfile.txt" | at now + 1 minute**
- **“ ls ”** check the file has been created.



Cron

One limitation of the at command is that we can't use it to run recurring tasks. For example, if I want to run a task every Monday, we can't use at to accomplish this. For this, we use crontab. Before we see what crontab is, let us look at the cron daemon first.

Cron Daemon

The cron daemon is a background process that is always running on a Linux machine. It is used to schedule jobs at regular time intervals, at specific recurring intervals, or on a particular date and time.

Cron Tables - known as crontab for short, are where the commands to be run, and the times to run them are stored.

Crontab

A crontab file is a simple plain text file in which each line represents a job. They are usually in the /etc folder (or its subdirectories), and each user has their own crontab file. Crontab files contain all of the scheduled jobs of a particular user. It is a text file that each user can edit and add their tasks to.

A cron job is a scheduled task that runs automatically at specified intervals. To set up a cron job that runs at 9 a.m. every day, you need to edit your crontab file.

Cron is installed by default, so we do not have to install it. However, we may have to create a crontab file for the current user.

Format for Crontab

Each line must have five fields, separated by a space - followed by the command or the script to execute. It looks like this:

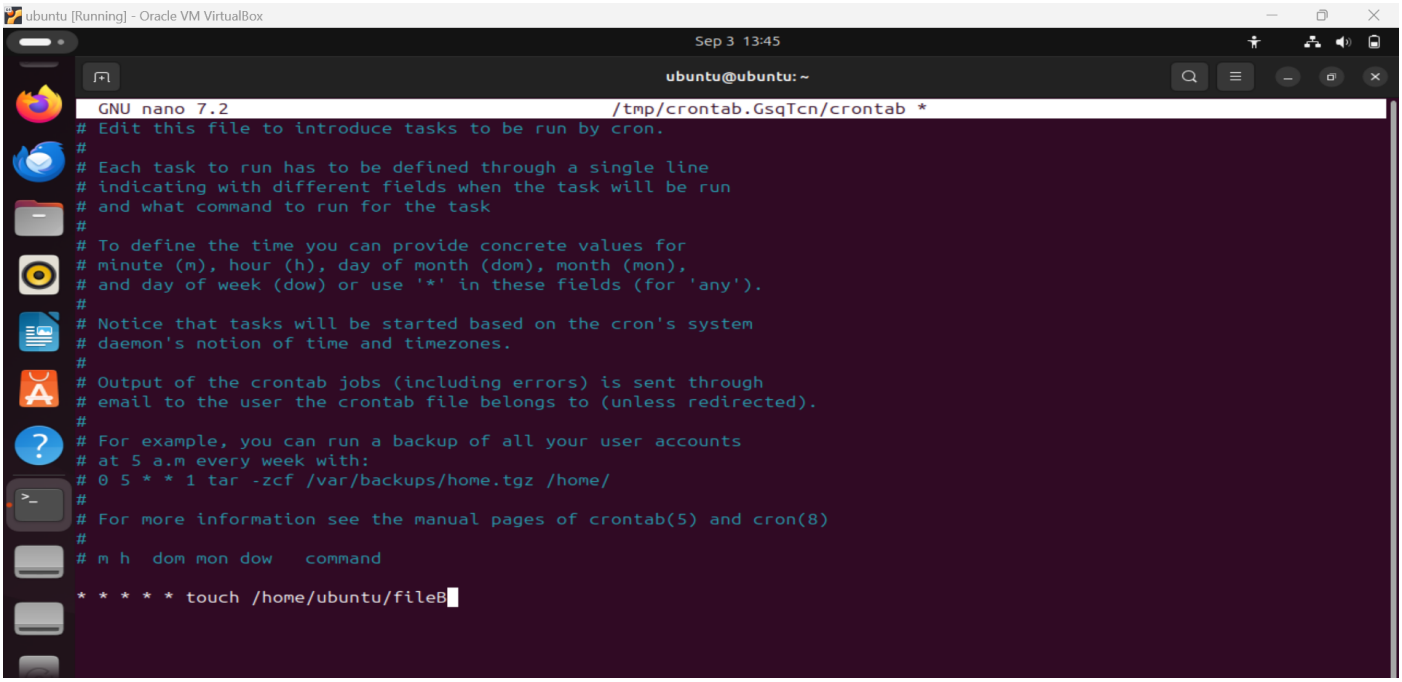
```
# ┌──────────── min (0 - 59)
# │ ┌──────────── hour (0 - 23)
# │ │ ┌──────────── day of month (1 - 31)
# │ │ │ ┌──────────── month (1 - 12)
# │ │ │ │ ┌──────────── day of week (0 - 6) (0 to 6 are Sunday to
# │ │ │ │ │                  Saturday, or use names; 7 is also Sunday)
# │ │ │ │ │
# │ │ │ │ │
# * * * * *  command to execute
```

Open the crontab file by using the command **crontab -e**

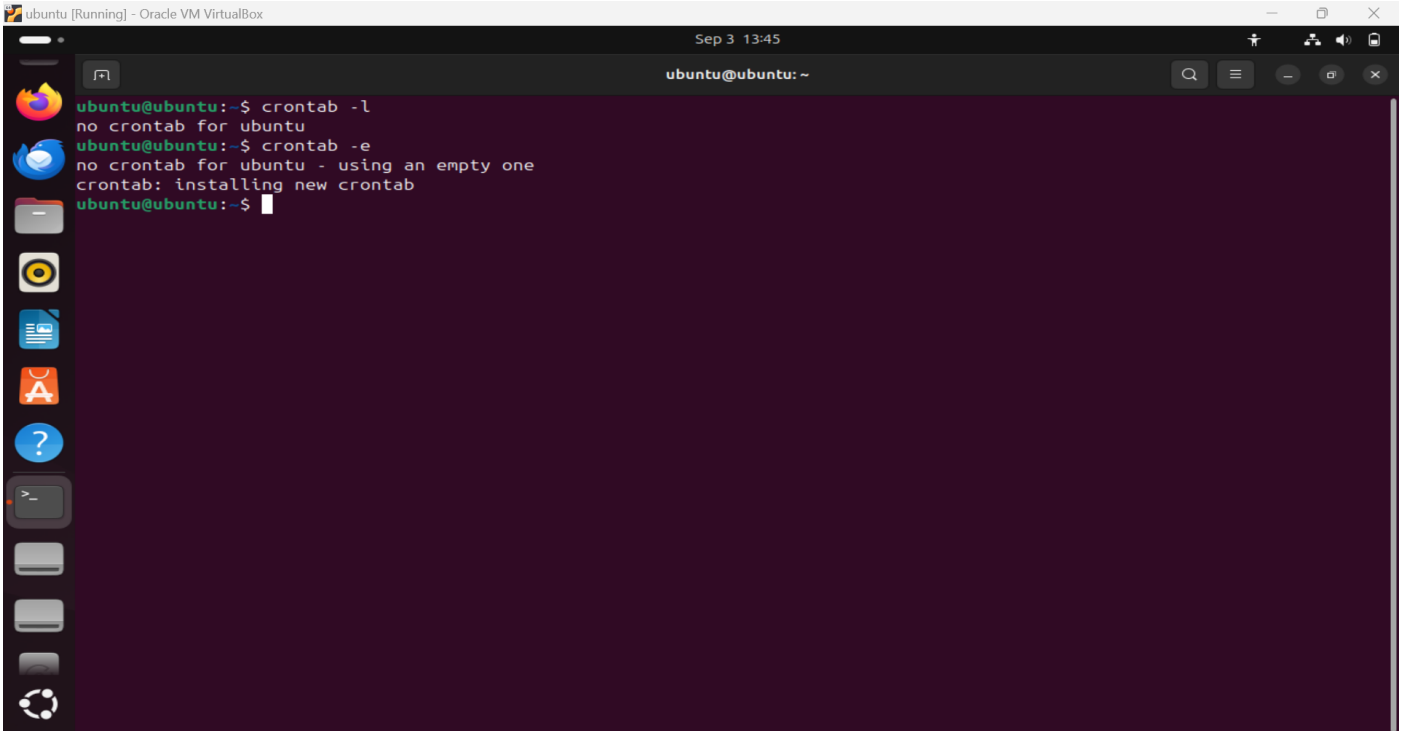
After using the crontab -e it will ask for **editor** to use. Select the editor as per your need.

```
no crontab for admin - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny

Choose 1-3 [1]: 1
```

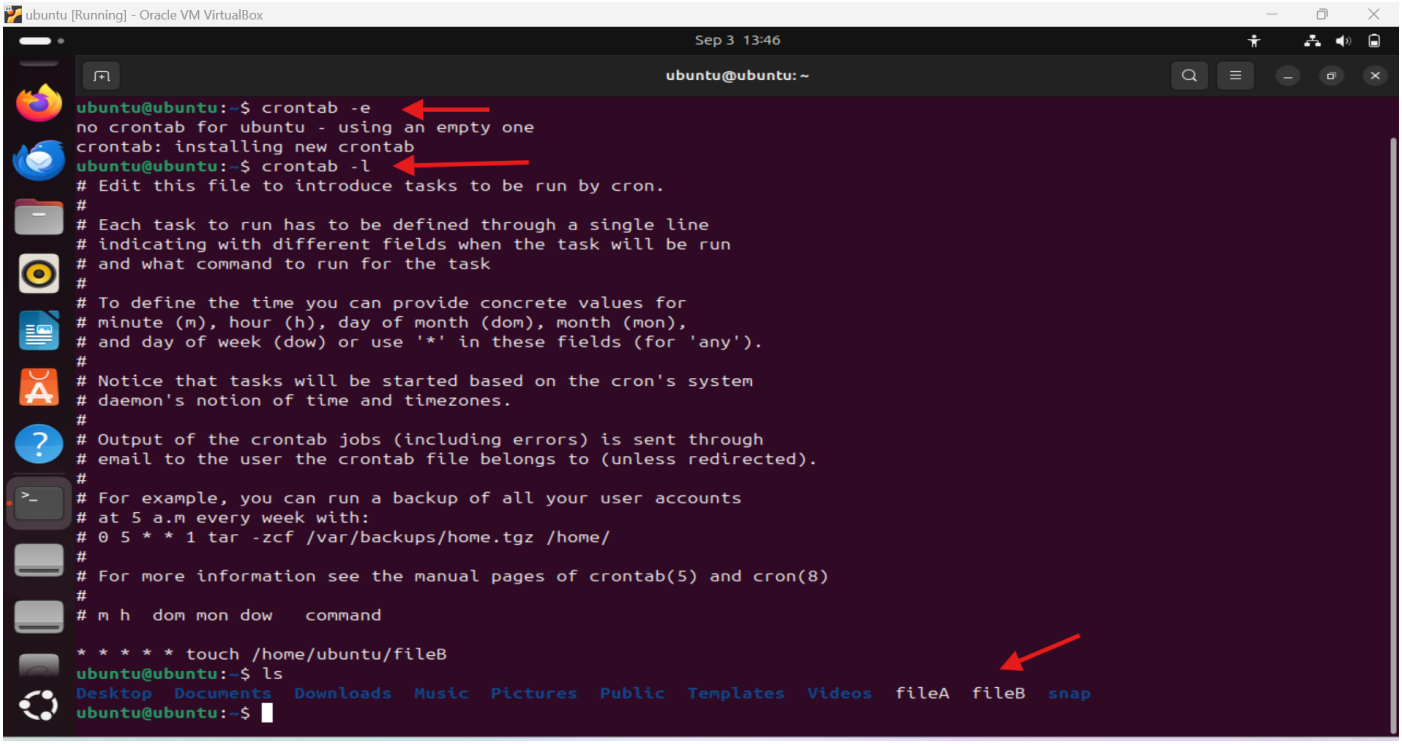
*** * * * * touch /home/ubuntu/fileB** this command will run every minute of every hour of every day of every month, and every day of the week to create the file.



This will create a file called fileB in our home directory.



The file just created should be visible. In home directory, check if the file has been created.



Whether you're managing server tasks or developing scripts, mastering cron and at can significantly enhance your productivity.