Chapter : Schedule Bash script being executed automatically at a desired time (using crontab)

==========

You can use the **crontab command** to manage cron jobs on your system. Here's a brief overview of how to use it:

To edit your user's crontab file, run crontab -e. This will open the crontab file in your default text editor.

In the crontab file, each line represents a cron job. Each job has five fields, separated by spaces, that specify when the job should run:

```
* * * * * command-to-be-executed
| | | | |
| | | | └──── day of the week (0 - 6) (Sunday to Saturday)
| | | └─────── month (1 - 12)
| | └──────── day of the month (1 - 31)
| └───────── hour (0 - 23)
└──────── minute (0 - 59)
```

For example, the following line specifies a **job that runs at 2:30 AM every day:**

```
30 2 * * * /path/to/command
```

To add a new cron job, add a new line to the crontab file in the format described above.

- Add and edit crontab : crontab -e
- To view your current cron jobs, run crontab -l.
- To remove all cron jobs, run crontab -r.
- To remove a specific cron job, edit the crontab file to remove the corresponding line.

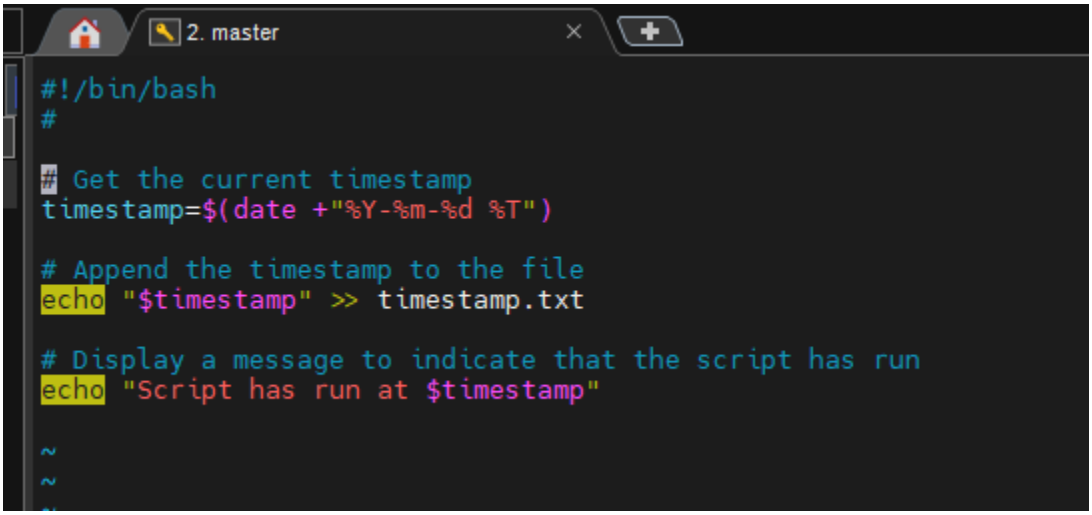
Note that the command-to-be-executed in each cron job can be any valid command, including a path to a shell script or a single command with arguments. If the command requires a specific working directory, you can specify the working directory before the command.

Chapter : Schedule Bash script being executed automatically at a desired time (using crontab)

==========

Made a Simple script that everytime it will run then it will store current timestamp in timestamp.txt

vi abc.sh



date is a command in Unix-like operating systems that prints or sets the system date and time.

+%Y-%m-%d %T is an argument passed to the **date** command that specifies the format in which the date and time should be displayed.

%Y means the year in 4 digits format, **%m** means the month in 2 digits format, **%d** means the day in 2 digits format, **%T** means the time in 24-hour format in the format of **HH:MM:SS**.

The **$(...)** syntax is used to execute the **date** command and store its output in a variable called **timestamp**. The resulting timestamp would be a string in the format of **YYYY-MM-DD HH:MM:SS**.

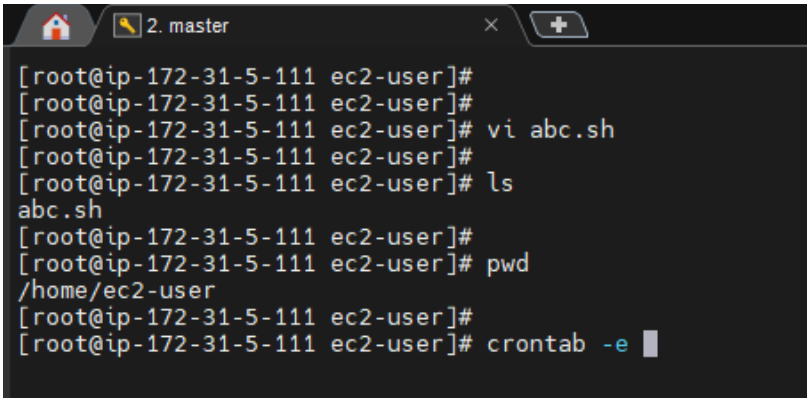
So the overall effect of **timestamp=$(date +"%Y-%m-%d %T")** is to generate a string containing the current date and time in the desired format and store it in the **timestamp** variable for later use.

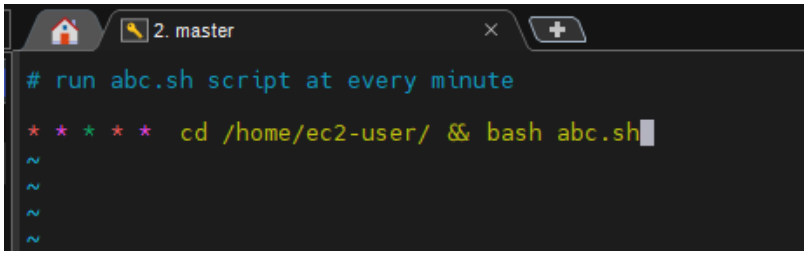
You can then use the **crontab** utility to schedule this script to run specified time, as described in my previous answer.

Chapter : Schedule Bash script being executed automatically at a desired time (using crontab)

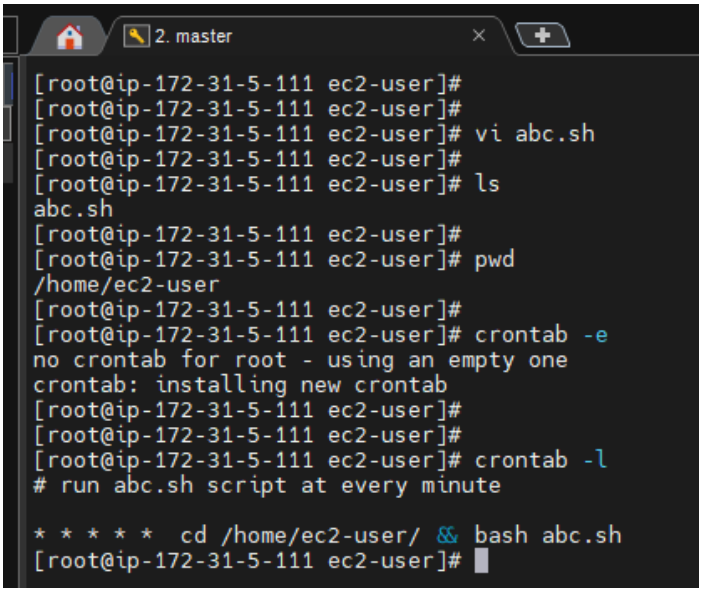
==========

Scheduling script using crontab command :



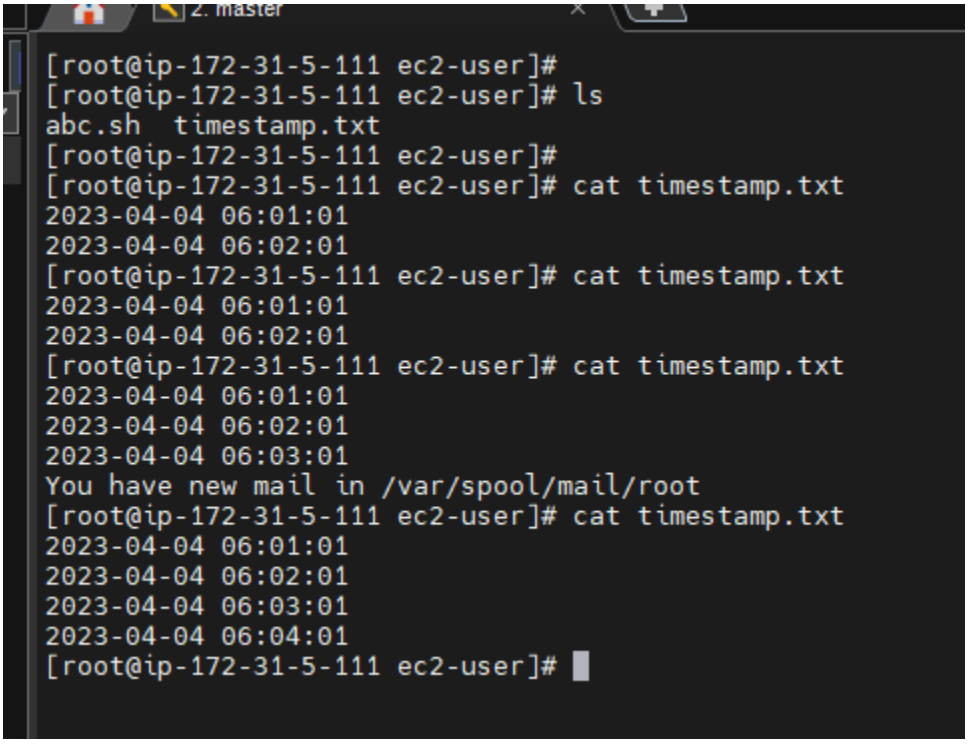


Save the file



Chapter : Schedule Bash script being executed automatically at a desired time (using crontab)

==========



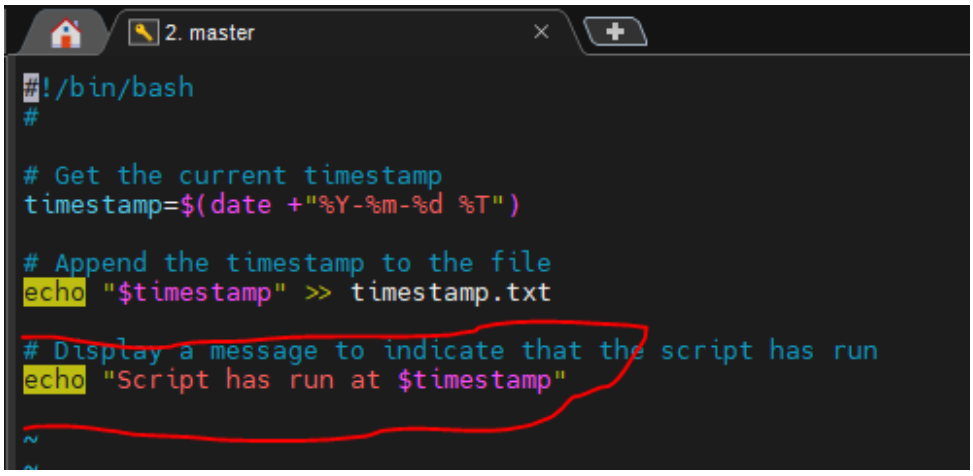
Its working.. :)

.....

crontab logs?

You can check in /var/spool/mail/

In my case it is /var/spool/mail/root



Chapter : Schedule Bash script being executed automatically at a desired time (using crontab)

==========