

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
```

Diagram illustrating the fields of a cron job line:

- Field 1: day of the week (0 - 6) (Sunday to Saturday)
- Field 2: month (1 - 12)
- Field 3: day of the month (1 - 31)
- Field 4: hour (0 - 23)
- Field 5: 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.

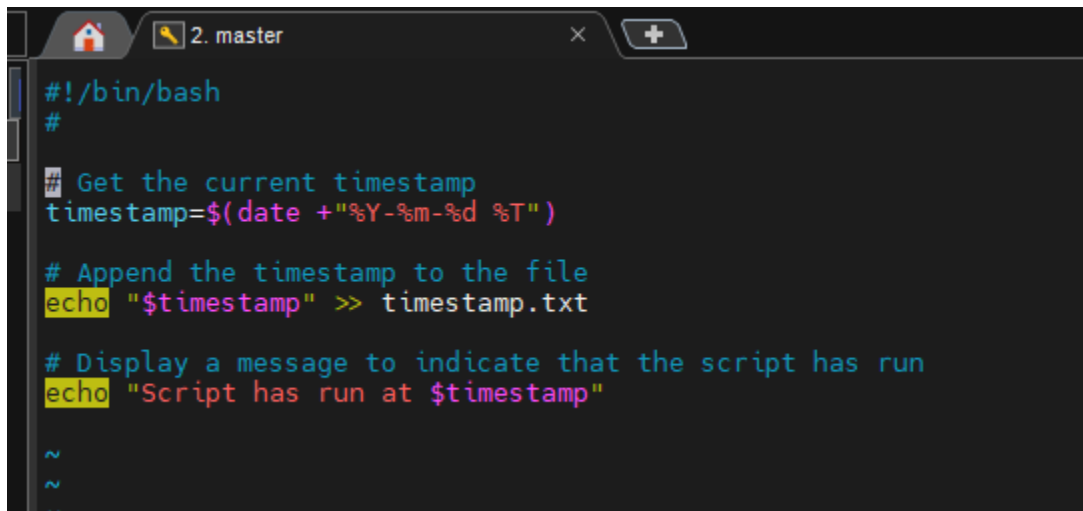
Note that the command-to-be-executed in each cron job can be any valid command, including a path to a shell script or a single command with arguments. If the command requires a specific working directory, you can specify the working directory before the command.

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



```
#!/bin/bash
#
# Get the current timestamp
timestamp=$(date +"%Y-%m-%d %T")

# Append the timestamp to the file
echo "$timestamp" >> timestamp.txt

# Display a message to indicate that the script has run
echo "Script has run at $timestamp"
~
~
~
```

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 :

```
2. master
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# vi abc.sh
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# ls
abc.sh
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# crontab -e
```

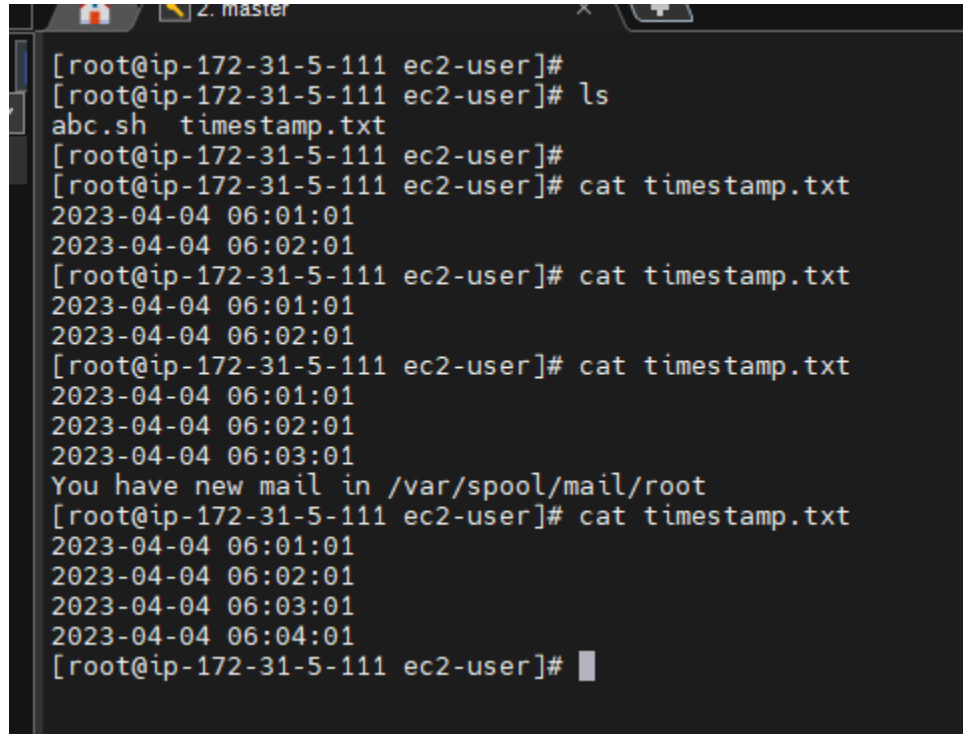
```
2. master
# run abc.sh script at every minute
* * * * * cd /home/ec2-user/ && bash abc.sh
~
~
~
~
```

Save the file

```
2. master
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# vi abc.sh
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# ls
abc.sh
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]#
[root@ip-172-31-5-111 ec2-user]# crontab -l
# run abc.sh script at every minute
* * * * * cd /home/ec2-user/ && bash abc.sh
[root@ip-172-31-5-111 ec2-user]#
```

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

=====



```
[root@ip-172-31-5-111 ec2-user]#  
[root@ip-172-31-5-111 ec2-user]# ls  
abc.sh  timestamp.txt  
[root@ip-172-31-5-111 ec2-user]#  
[root@ip-172-31-5-111 ec2-user]# cat timestamp.txt  
2023-04-04 06:01:01  
2023-04-04 06:02:01  
[root@ip-172-31-5-111 ec2-user]# cat timestamp.txt  
2023-04-04 06:01:01  
2023-04-04 06:02:01  
[root@ip-172-31-5-111 ec2-user]# cat timestamp.txt  
2023-04-04 06:01:01  
2023-04-04 06:02:01  
2023-04-04 06:03:01  
You have new mail in /var/spool/mail/root  
[root@ip-172-31-5-111 ec2-user]# cat timestamp.txt  
2023-04-04 06:01:01  
2023-04-04 06:02:01  
2023-04-04 06:03:01  
2023-04-04 06:04:01  
[root@ip-172-31-5-111 ec2-user]#
```

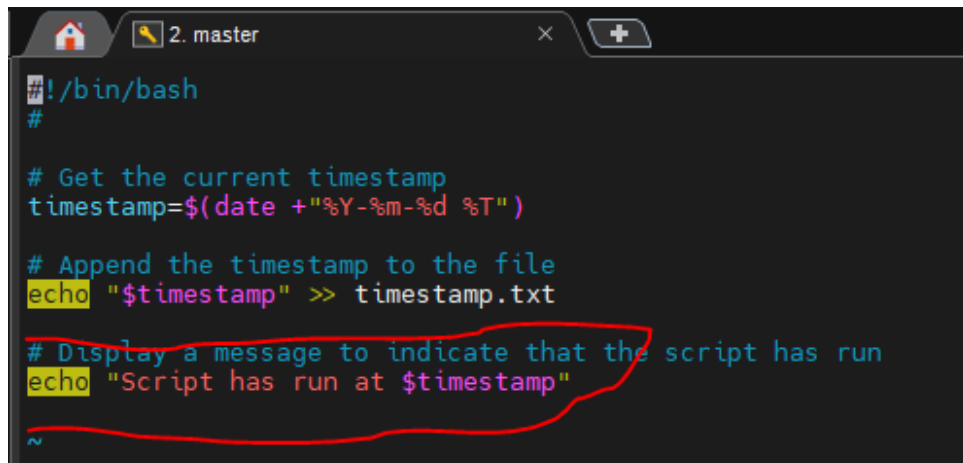
Its working.. :)

.....

crontab logs?

You can check in /var/spool/mail/

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



```
#!/bin/bash  
#  
# Get the current timestamp  
timestamp=$(date +"%Y-%m-%d %T")  
# Append the timestamp to the file  
echo "$timestamp" >> timestamp.txt  
# Display a message to indicate that the script has run  
echo "Script has run at $timestamp"
```

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

```
=====

From root@ip-172-31-5-111.ap-south-1.compute.internal Tue Apr  4 06:01:01 2023
Return-Path: <root@ip-172-31-5-111.ap-south-1.compute.internal>
X-Original-To: root
Delivered-To: root@ip-172-31-5-111.ap-south-1.compute.internal
Received: by ip-172-31-5-111.ap-south-1.compute.internal (Postfix, from userid 0
)
    id 45CC74ABBCC; Tue,  4 Apr 2023 06:01:01 +0000 (UTC)
From: "(Cron Daemon)" <root@ip-172-31-5-111.ap-south-1.compute.internal>
To: root@ip-172-31-5-111.ap-south-1.compute.internal
Subject: Cron <root@ip-172-31-5-111> cd /home/ec2-user/ && bash abc.sh
Content-Type: text/plain; charset=ANSI_X3.4-1968
Auto-Submitted: auto-generated
Precedence: bulk
X-Cron-Env: <XDG_SESSION_ID=801>
X-Cron-Env: <XDG_RUNTIME_DIR=/run/user/0>
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/root>
X-Cron-Env: <PATH=/usr/bin:/bin>
X-Cron-Env: <LOGNAME=root>
X-Cron-Env: <USER=root>
Message-Id: <20230404060101.45CC74ABBCC@ip-172-31-5-111.ap-south-1.compute.inter
nal>
Date: Tue,  4 Apr 2023 06:01:01 +0000 (UTC)
Script has run at 2023-04-04 06:01:01

From root@ip-172-31-5-111.ap-south-1.compute.internal Tue Apr  4 06:02:01 2023
Return-Path: <root@ip-172-31-5-111.ap-south-1.compute.internal>
X-Original-To: root
Delivered-To: root@ip-172-31-5-111.ap-south-1.compute.internal
Received: by ip-172-31-5-111.ap-south-1.compute.internal (Postfix, from userid 0
)
    id 49AE14ABBCC; Tue,  4 Apr 2023 06:02:01 +0000 (UTC)
From: "(Cron Daemon)" <root@ip-172-31-5-111.ap-south-1.compute.internal>
To: root@ip-172-31-5-111.ap-south-1.compute.internal
Subject: Cron <root@ip-172-31-5-111> cd /home/ec2-user/ && bash abc.sh
Content-Type: text/plain; charset=ANSI_X3.4-1968
Auto-Submitted: auto-generated
Precedence: bulk
X-Cron-Env: <XDG_SESSION_ID=803>
X-Cron-Env: <XDG_RUNTIME_DIR=/run/user/0>
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/root>
X-Cron-Env: <PATH=/usr/bin:/bin>
X-Cron-Env: <LOGNAME=root>

1,1 Top
```