## Exercise 1 - List cron jobs

The -l option of the crontab command prints the current crontab.

```
crontab -l
```

You may get a message `no crontab` if your crontab is empty.

## Exercise 2 - Add a job in the crontab file

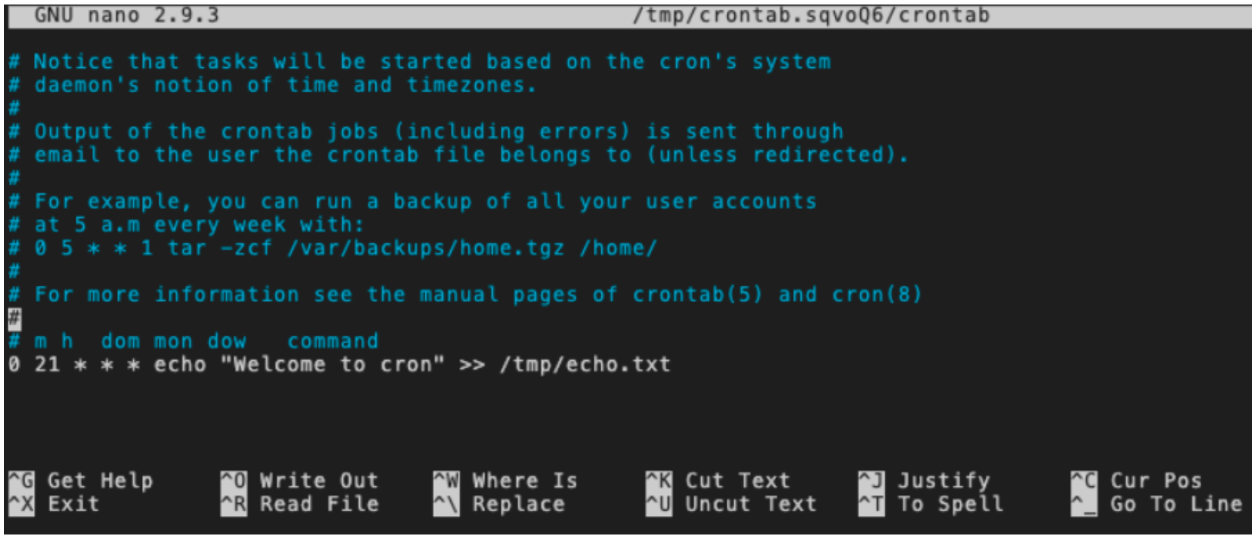
### 3.1. Add a job to crontab.

To add a cron job, run the command below

```
crontab -e
```

Add the below line at the end of the crontab file.

```
0 21 * * * echo "Welcome to cron" >> /tmp/echo.txt
```



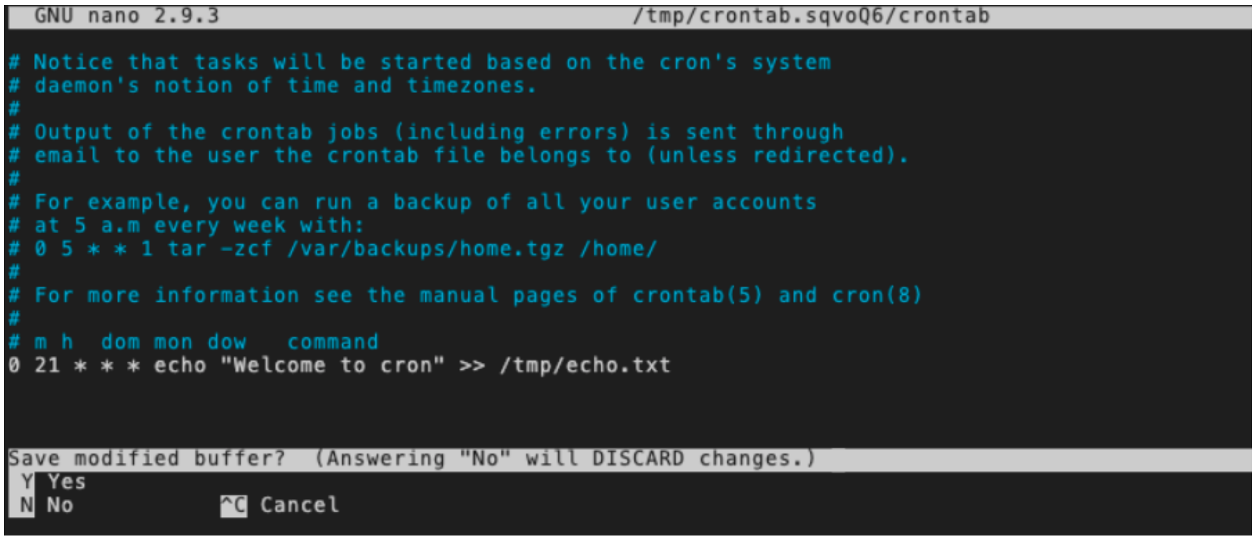
```
  GNU nano 2.9.3                              /tmp/crontab.sqvoQ6/crontab

# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 21 * * * echo "Welcome to cron" >> /tmp/echo.txt
```

The above job specifies that the echo command should run when the minute is 0 and the hours is 21. It effectively means the job runs at 9.00 p.m every day.

The output of the command should be sent to a file /tmp/echo.txt.

Press Control + X to save the changes.

Press 'Y' to confirm.



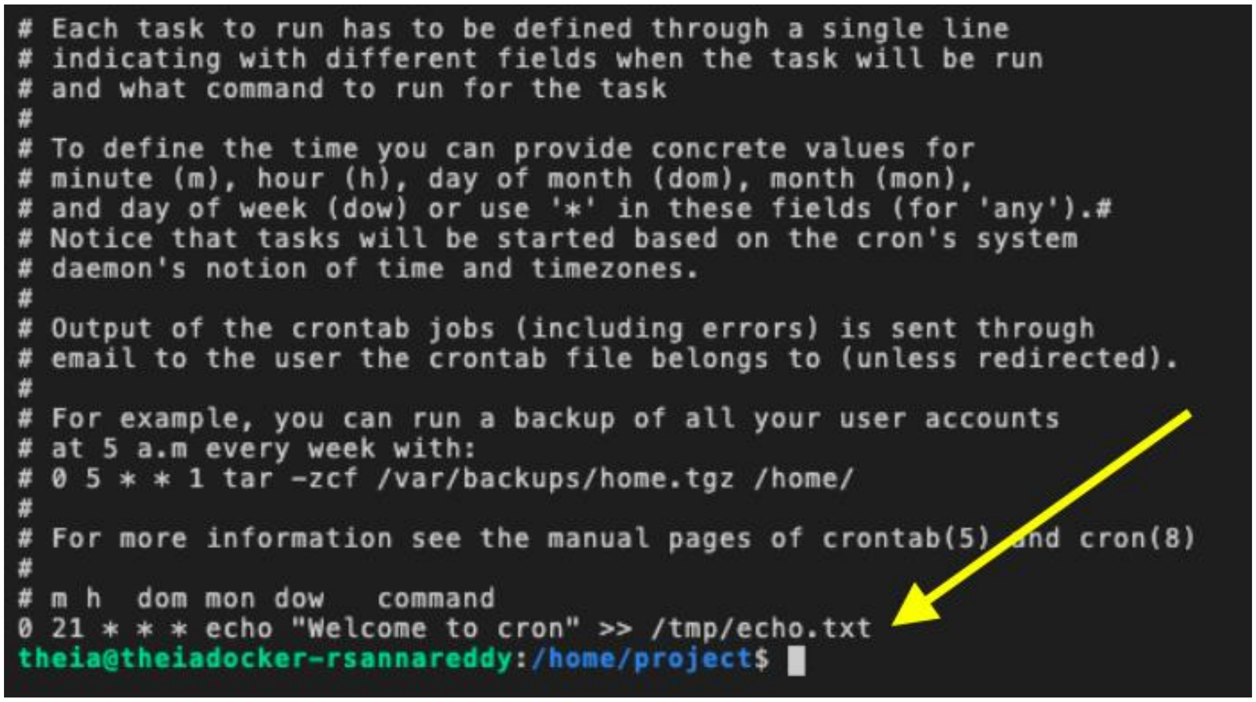
```
  GNU nano 2.9.3                                    /tmp/crontab.sqvoQ6/crontab

# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 21 * * * echo "Welcome to cron" >> /tmp/echo.txt


Save modified buffer?  (Answering "No" will DISCARD changes.)
 Y Yes
 N No           ^C Cancel
```

Press Enter to come out of the editor.

Check if the job is added to the crontab by running the following command.

`crontab -l`

You should see the newly added job in the output.



```
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 21 * * * echo "Welcome to cron" >> /tmp/echo.txt
theia@theiadocker-rsannareddy:/home/project$
```

## 3.2. Schedule a shell script.

Let us create a simple shell script that prints the current time and the current disk usage statistics.

Step 1: Create script called “mycron.sh”

Step 2: add this into the script.

```
#! /bin/bash
# print the current date time
date
# print the disk free statistics
df -h
```

Step 3: Save the file.

Step 4: Verify that the script is working:

```
chmod u+x diskusage.sh
./diskusage.sh
```

The script should print the current timestamp and the disk usage stats.

Let us schedule this script to be run everyday at midnight 12:00 (when the hour is 0 on the 24 hour clock). We want the output of this script to be appended to `/home/project/diskusage.log`.

Edit the crontab.

```
crontab -e
```

Add the following line to the end of the file:

```
0 0 * * * /home/project/disksusage.sh >>/home/project/diskusage.log
```

Press Control + X to save the changes.

Press 'Y' to confirm.

Press Enter to come out of the editor.

Check if the job is added to the crontab by running the following command.

```
crontab -l
```

You should see the newly added job in the output.

## Exercise 4 - Remove the current crontab

The -r option causes the current crontab to be removed.

Caution: This removes all your cron jobs. Be extra cautious when you use this command on a production server.

```
crontab -r
```

Verify if your crontab is removed.

```
crontab -l
```