

1. How do you stop SSH service with systemd and make sure it won't start during system bootup? Do this task and verify it by booting the system. (Important: do not do this if you are using VLE environment, but rather only write how you would do this!)

```
user@AC4892-Ubuntu:~$ sudo systemctl disable ssh
```

2. Create a scheduled task with systemd which writes date command output into the file date.txt on user's home directory every two minutes on current day each week. For example if now is tuesday, date information will be written on every tuesday on every week.

```
user@AC4892-Ubuntu:~$ systemd-run --user --on-calendar 'tue *-*-* 00:02:00' /bin/sh -c "date >> ~/date.txt"
Running timer as unit: run-r2c8fdd49dc66457cbb3c7ed3e5a2a960.timer
Will run service as unit: run-r2c8fdd49dc66457cbb3c7ed3e5a2a960.service
user@AC4892-Ubuntu:~$
```

3. Check help for systemctl command. How can you only list enabled unit files that are services?

```
user@AC4892-Ubuntu:~$ systemctl list-unit-files *.service
```

UNIT FILE	STATE
sshd.service	disabled

UNIT FILE	STATE	VENDOR PRESET
accounts-daemon.service	enabled	enabled
apparmor.service	enabled	enabled
apport-autoreport.service	static	enabled
apport-forward@.service	static	enabled
apport.service	generated	enabled
apt-daily-upgrade.service	static	enabled
apt-daily.service	static	enabled
atd.service	enabled	enabled
autovt@.service	enabled	enabled
blk-availability.service	enabled	enabled
bolt.service	static	enabled
clean-mount-point@.service	static	enabled
cloud-config.service	enabled	enabled
cloud-final.service	enabled	enabled
cloud-init-hotplugd.service	static	enabled
cloud-init-local.service	enabled	enabled
cloud-init.service	enabled	enabled
console-getty.service	disabled	disabled
console-setup.service	enabled	enabled
container-getty@.service	static	enabled
cron.service	enabled	enabled
cryptdisks-early.service	masked	enabled
cryptdisks.service	masked	enabled
dbus-org.freedesktop.hostname1.service	static	enabled
dbus-org.freedesktop.locale1.service	static	enabled
dbus-org.freedesktop.login1.service	static	enabled
dbus-org.freedesktop.ModemManager1.service	enabled	enabled

4. Install Apache2 web server to your Ubuntu with apt package management.

With what name was this service installed as systemd service? Web server is started during system bootup by default and listens to port 80. Change systemd setting for this service so that it will not be started automatically during system bootup. Reboot the system and verify that web server has not been started (port 80 is not listened).

```
user@AC4892-Ubuntu:~$ sudo apt install apache2
```

```
user@AC4892-Ubuntu:~$ sudo nano etc/apache2/ports.conf
```

```
Listen 8080
```

```
user@AC4892-Ubuntu:~$ sudo systemctl disable apache2.service
```

```
user@AC4892-Ubuntu:~$ sudo systemctl reboot apache2.SERVICE
```

```
user@AC4892-Ubuntu:~$ sudo systemctl status apache2
[sudo] password for user:
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: enabled)
  Active: inactive (dead)
  Docs: https://httpd.apache.org/docs/2.4/
user@AC4892-Ubuntu:~$
```

5. Create a directory called backup to your home directory. Now create a regular backup with systemd so that the following files from /var/log directory will be added to *backup.tar.gz* archive every day (you may choose the time this occurs daily) and saved into the backup directory inside your home directory:

- auth.log
- kern.log
- faillog

```
user@AC4892-Ubuntu:~$ mkdir backup
```

```
user@AC4892-Ubuntu:~$ crontab -e
```

```
user@AC4892-Ubuntu:~$ crontab -e
no crontab for user - 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
4. /bin/ed

```
user@AC4892-Ubuntu:~$ systemd-run --user --on-calendar '* * * 00:00:00' tar -cvz home/user/backup/backup.tar.gz var/log/auth.log var/log/kern.log var/log/fail.log
```