

# AdvLinuxTU

User Manual

## V4.0

**ADVANTECH**

*Enabling an Intelligent Planet*

---

## Contents

1.	About This Manual .....	1
2.	AdvLinuxTU Overview .....	2
2.1.	Hardware Support List .....	2
2.2.	Main Features.....	2
2.3.	AdvLinuxTU Screenshot .....	3
3.	Installation Guide.....	4
3.1.	System Requirements.....	4
3.2.	Prepare for Installation.....	4
3.3.	Install from USB disk .....	4
3.4.	AdvLinuxTU inactively auto upgrade.....	6
3.5.	Raid mode & Installation Steps.....	7
3.5.1.	Setting RAID Mode .....	7
3.5.2.	Create RAID Volume .....	7
3.6.	Boot Mode & Installation Steps .....	9
3.6.1.	Legacy Boot Mode Installation Steps .....	9
3.6.2.	UEFI Boot Mode Installation Steps .....	12
4.	Advantech device driver .....	34
5.	GPIO .....	35
5.1.	Sysfs Interface.....	35
5.1.1.	UNO-238 .....	35
5.1.2.	UNO-420 .....	36
5.1.3.	UNO-137 .....	36
5.1.4.	UNO-148 .....	37
5.1.5.	UNO-348 .....	37
5.1.6.	UNO-2484G_V2.....	38
5.1.7.	UNO-127 / UNO-2271G_V2 / AMAX-5570 .....	38
5.1.8.	UNO-238_V2.....	39
5.1.9.	UNO-137_V2.....	39
5.1.10.	AMAX-5570S .....	40
5.2.	Libgpio.....	40
6.	EAPI(Embedded API) Solution.....	40
7.	Introduction of service .....	40
7.1.	SSH .....	40
7.2.	FTP .....	41
7.3.	Watchdog.....	43
7.4.	WISE-Agent.....	44
7.5.	Firewall .....	44
8.	Full disk encryption .....	46
8.1.	Install .....	46
8.2.	Change keys .....	47
9.	Read-Only system setting .....	49
9.1.	Install .....	49
9.2.	Changes for read-only setting .....	51

---

9.3.	Uninstall.....	52
10.	Backup & Restore.....	54
10.1.	Backup .....	54
10.2.	Restore .....	56

# 1. About This Manual

Thank you for choosing Advantech.

AdvLinuxTU v4.0 is based on Ubuntu Desktop 22.04.5, its kernel is 5.15.

AdvLinuxTU-IOTG v4.0 is based on Ubuntu Intel IoT, its kernel is 5.15.

## 2. AdvLinuxTU Overview

AdvLinuxTU is an embedded Linux system designed for Advantech embedded devices, and it is especially optimized for TPC/UNO/PPC series.

### 2.1. Hardware Support List

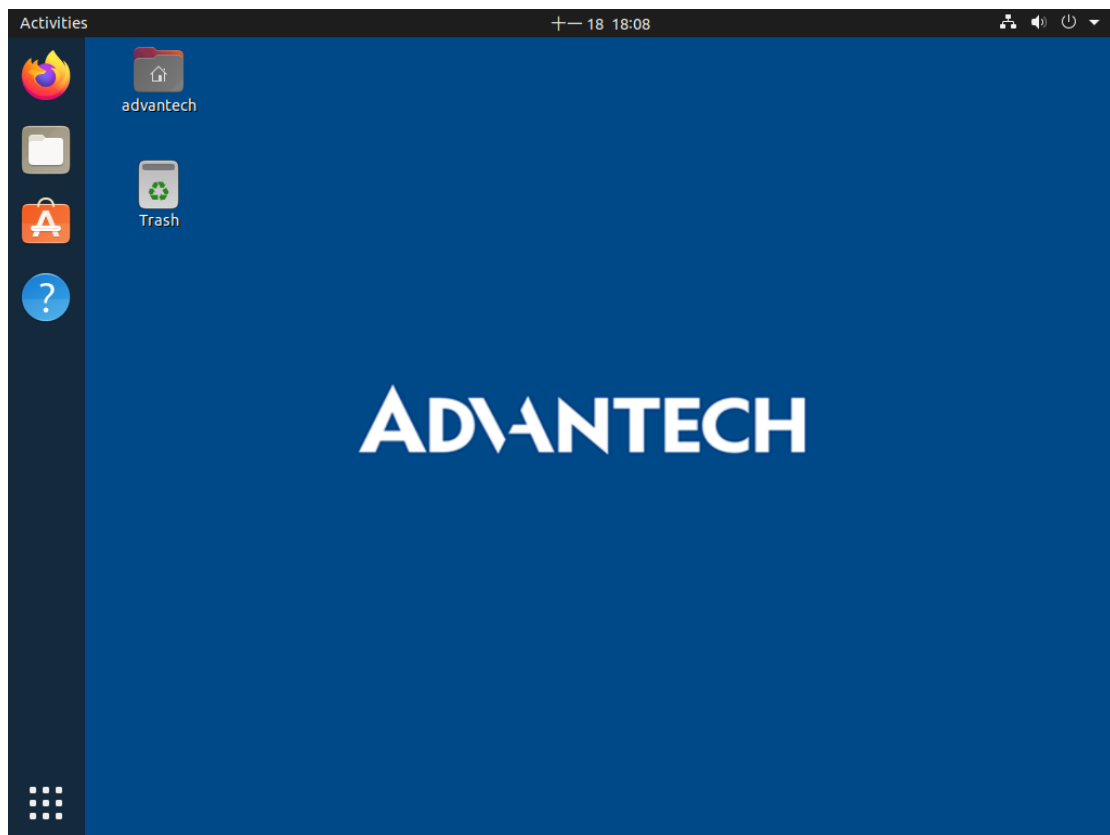
AdvLinuxTU can be installed on all Advantech devices theoretically.

### 2.2. Main Features

AdvLinuxTU has many important features coming for industrial customers. Here we list some of them and give a brief introduction. We will explain them in detail in later sections.

- **Provide Embedded QT runtime environment**  
The user can run embedded QT application.
- **USB installation support**  
The user can burn the AdvLinuxTU ISO to a USB disk, and then the user can install AdvLinuxTU to Advantech device from the USB disk.
- **Hardware auto detection**  
When installing AdvLinuxTU, the installation will detect the hardware automatically and install all drivers it needs. Therefore, once the installation is completed, no more drivers will be needed unless the user adds some other devices such as Advantech data collecting card to the embedded device.
- **OEM Installation**  
User can install OEM mode and ship to customers, they can create them language, keyboard and account of Ubuntu.
- **Online Installation and Update**  
You can use apt-get command to install and update software, extending AdvLinuxTU functions.

## 2.3. AdvLinuxTU Screenshot



## 3. Installation Guide

In this chapter, we will introduce the installation step by step.

### 3.1. System Requirements

Recommended requirements:

Storage size:

32 G or more

Memory size:

2 G or more

### 3.2. Prepare for Installation

We suggest that the user makes a copy of data in storage medium in order to prevent data lost in case of improper operation during installation.

AdvLinuxTU supports installing from USB disk.

Prepare:

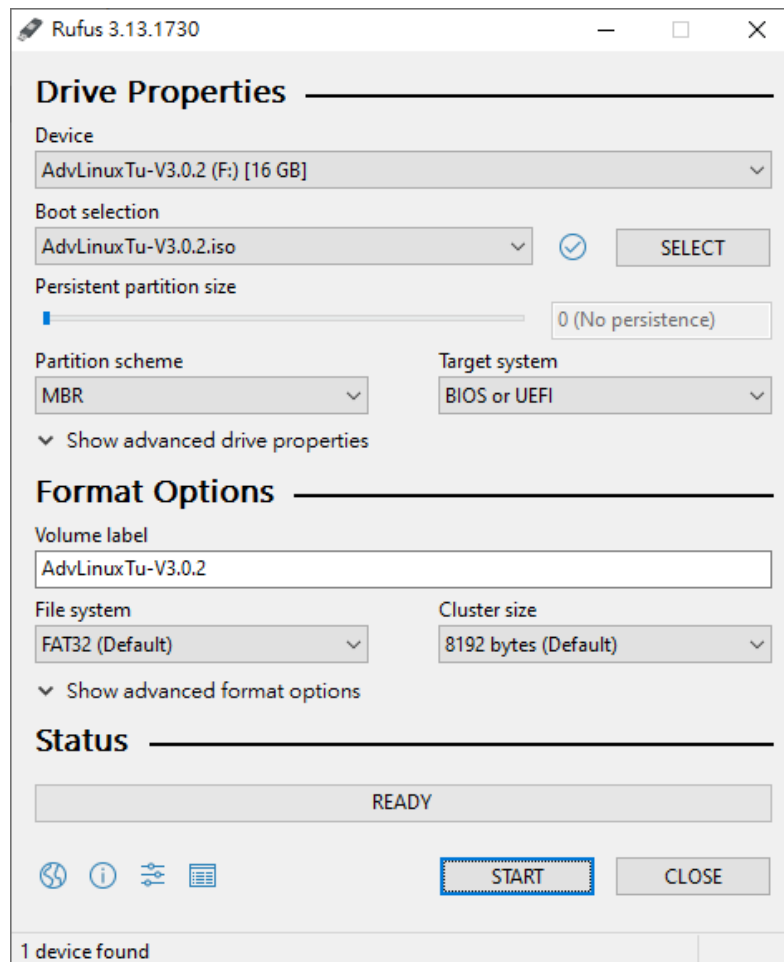
- ✓ Rufus (Rufus is open source and is 100% Free Software (GPL v3), You can download from <https://rufus.ie/>)
- ✓ USB Disk (storage size is more than 4G)
- ✓ AdvLinuxTU ISO

### 3.3. Install from USB disk

We use Rufus tool to burn AdvLinuxTU ISO file to USB disk, Rufus is a utility that helps format and create bootable USB flash drives.

#### **System Requirements for Rufus:**

Windows 7 or later, 32 or 64 bit doesn't matter. Once downloaded, the application is ready to use.



Picture 3-3-1 Select the AdvLinuxTU ISO File and start to burn

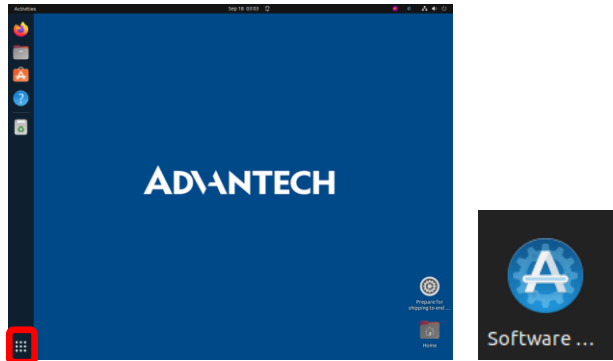
Burn the ISO as follows:

- a) Insert USB disk to windows PC
- b) Start Rufus tool
- c) Select AdvLinuxTU ISO
- d) Click START to start burn

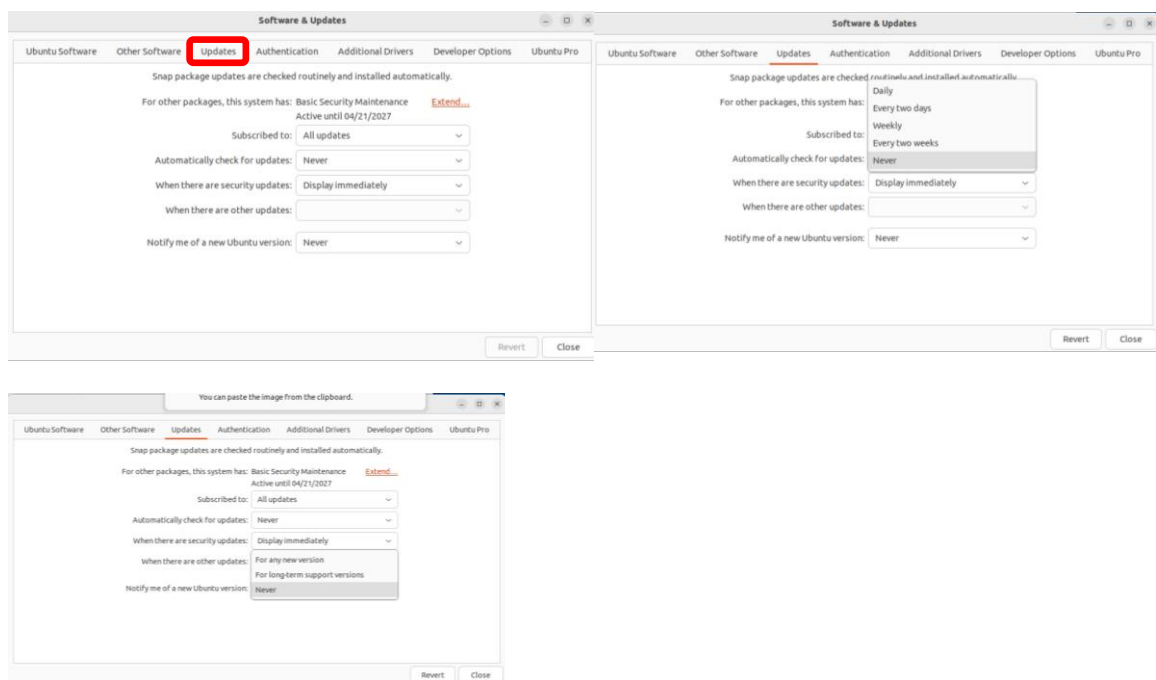


## 3.4. AdvLinuxTU inactively auto upgrade

AdvLinuxTU disabled automatic update by default. If you want enable automatic update Find Software & Updates then open the application.



Select “Updates”, then you can select your strategy



Update the package lists

```
$ sudo systemctl unmask apt-daily.timer
```

```
$ sudo systemctl enable apt-daily.timer
```

```
$ sudo systemctl start apt-daily.timer
```

## 3.5. Raid mode & Installation Steps

UNO-2483G, UNO-2484G, UNO-3283G, TPC-B500...etc support raid mode.

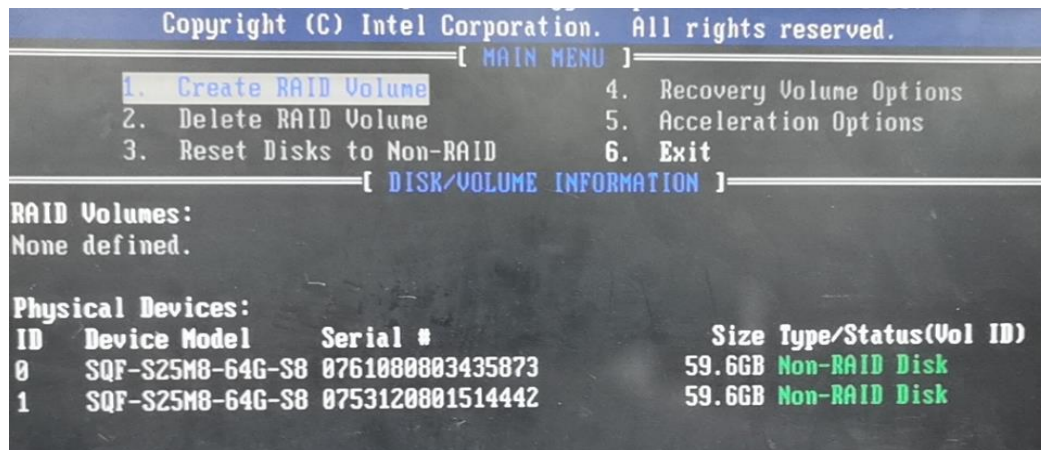
**Note:** AdvLinuxTU v4.x not support temporary

### 3.5.1. Setting RAID Mode

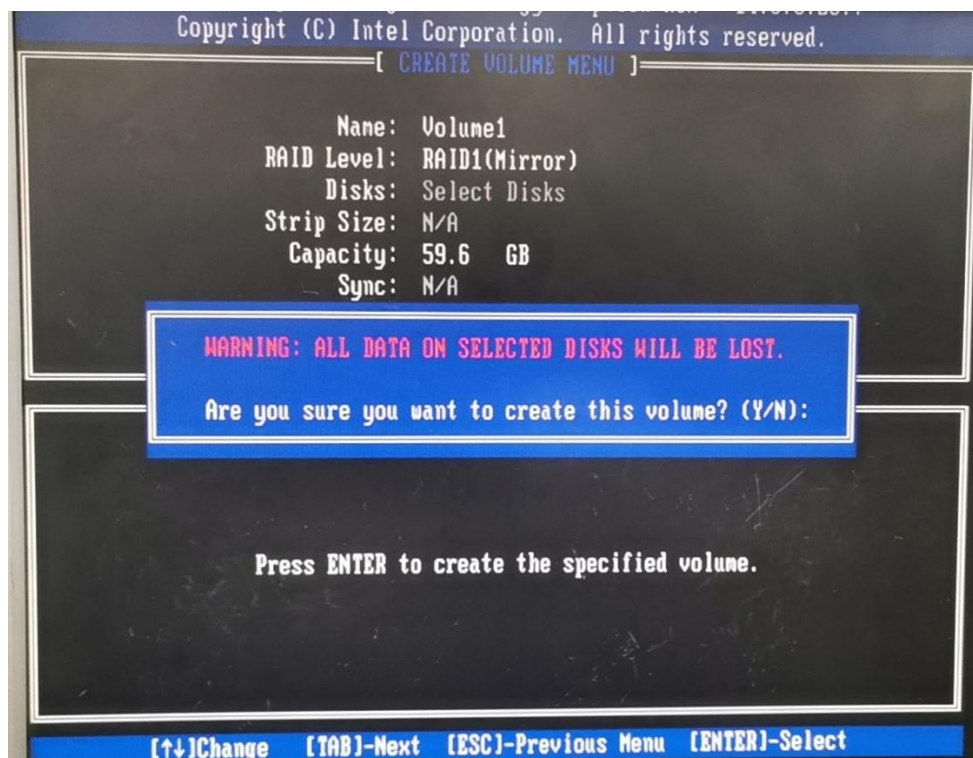
Enter the BIOS interface, set the hard disk mode to RAID mode.

### 3.5.2. Create RAID Volume

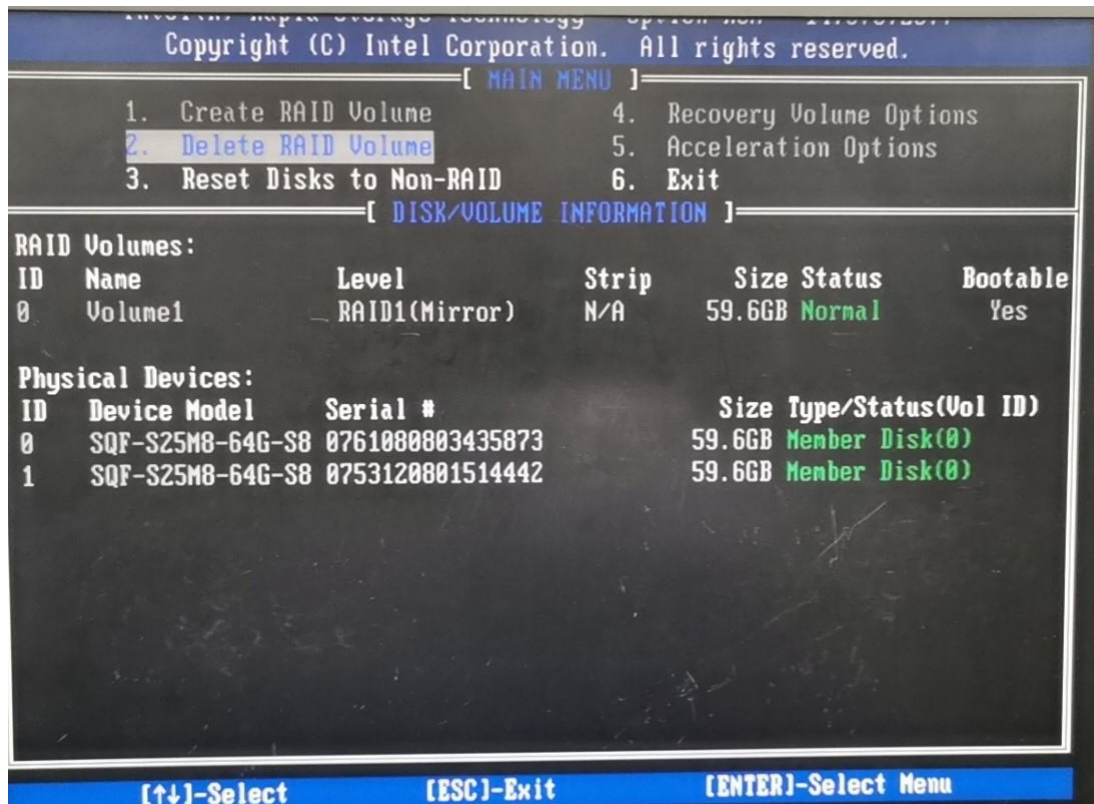
Enter “ctr + i” will show the RAID interface, choose “Create RAID Volume” and press enter.



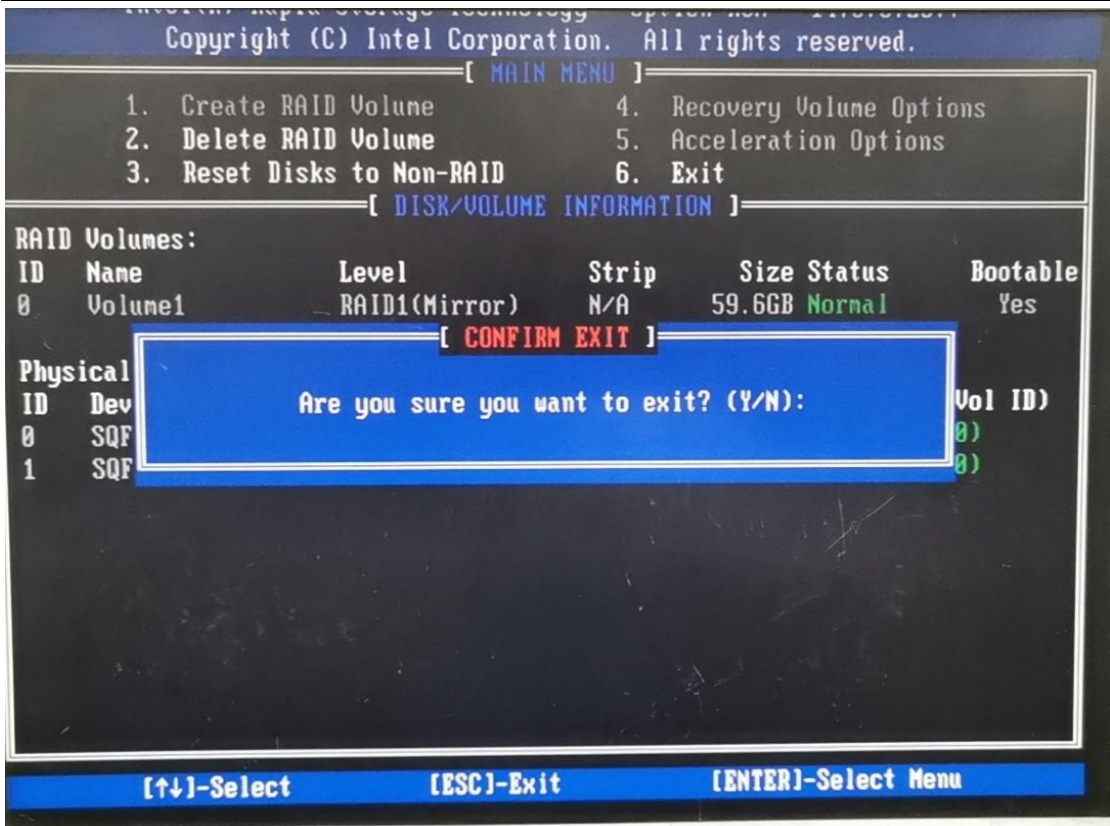
Enter “Shift + y” to Create Raid Volume. At present, RAID1 is mainly supported. The specific steps are as follows



Don't modify the default "Volume1", otherwise AdvlinuxTU will install failed.  
You can see Volume1 show up.



Enter "Shift + y" to exit.



### 3.6. Boot Mode & Installation Steps

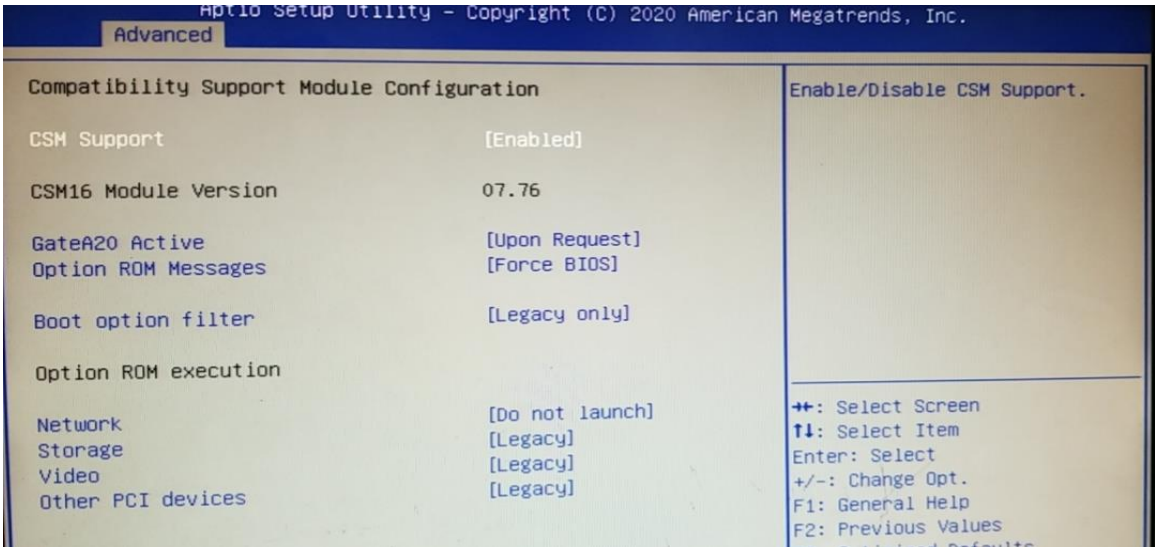
AdvLinuxTU supports two boot modes: Legacy and UEFI (recommend).

#### 3.6.1. Legacy Boot Mode Installation Steps

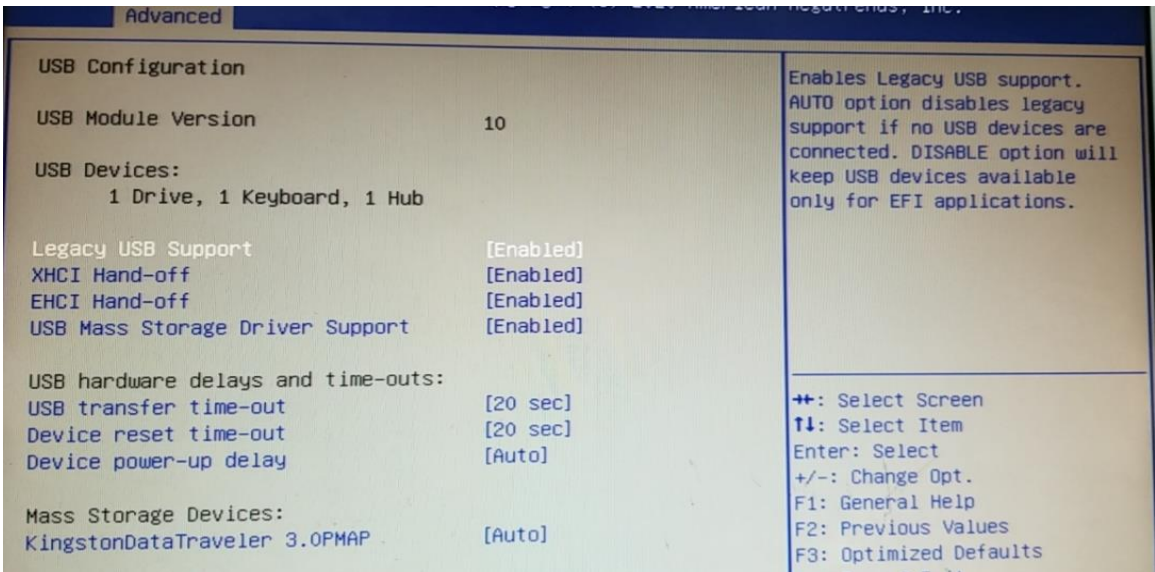
##### 3.6.1.1. BIOS Configuration

Step1: CSM configuration

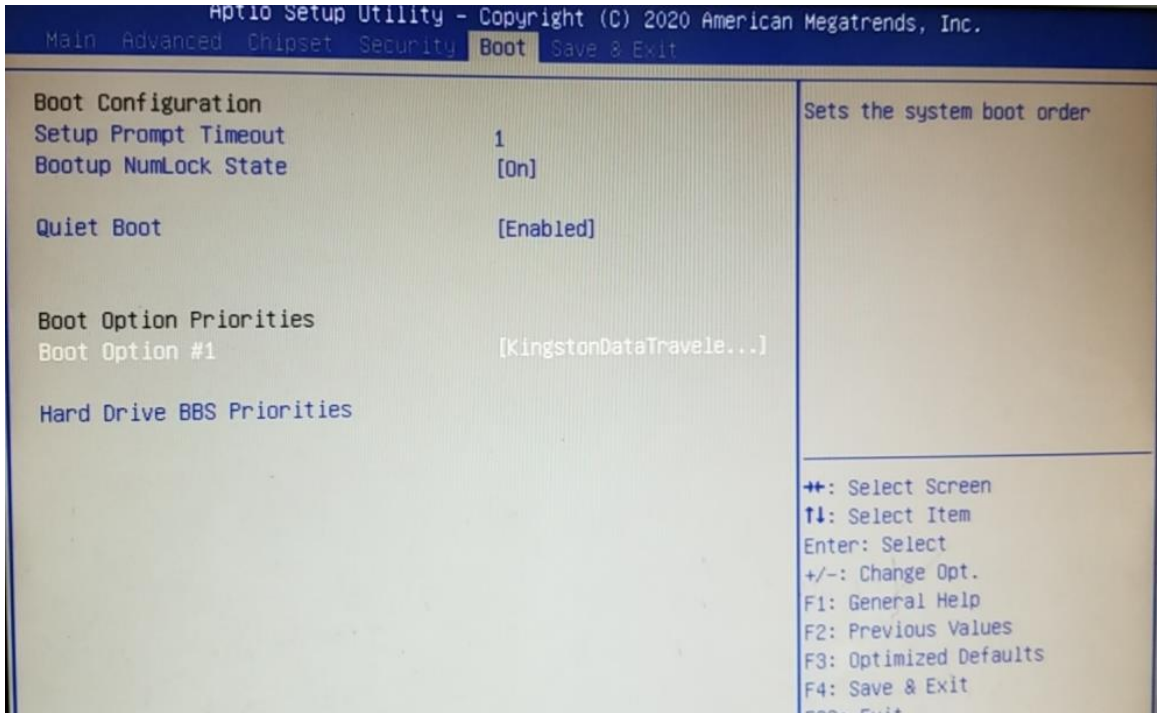




Step2: USB configuration

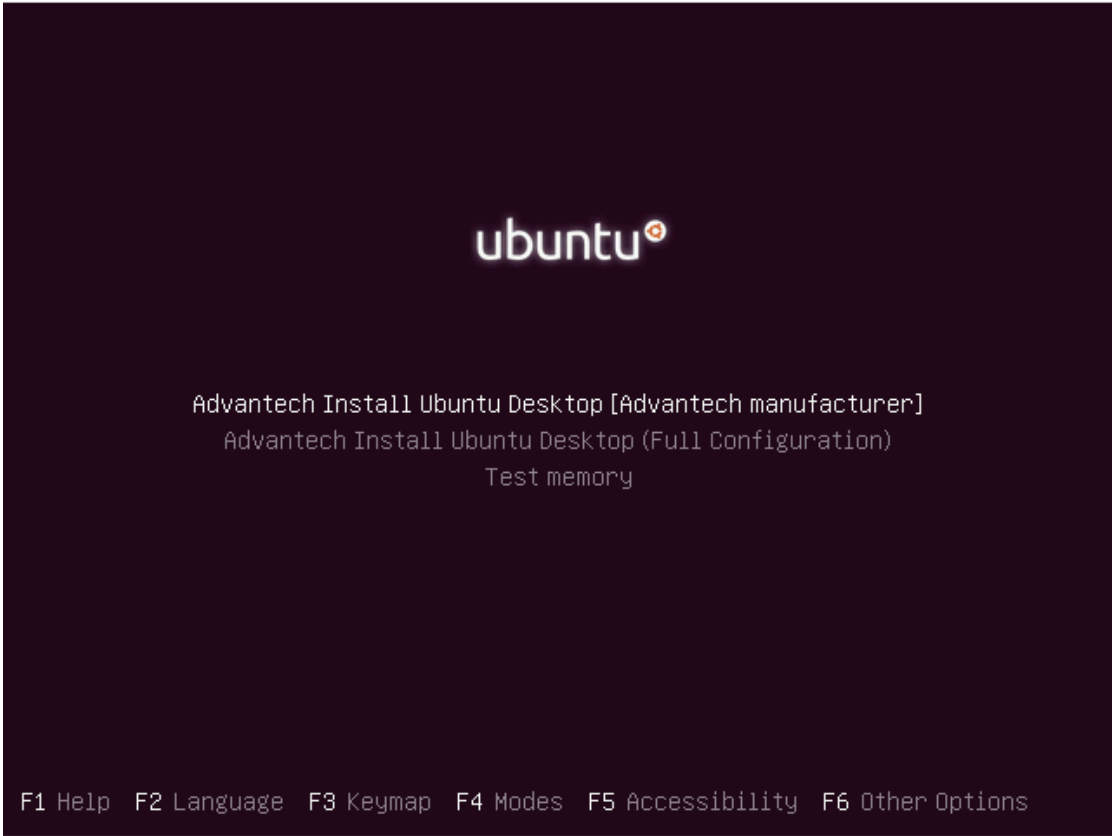


Step3: Boot Option #1 selection

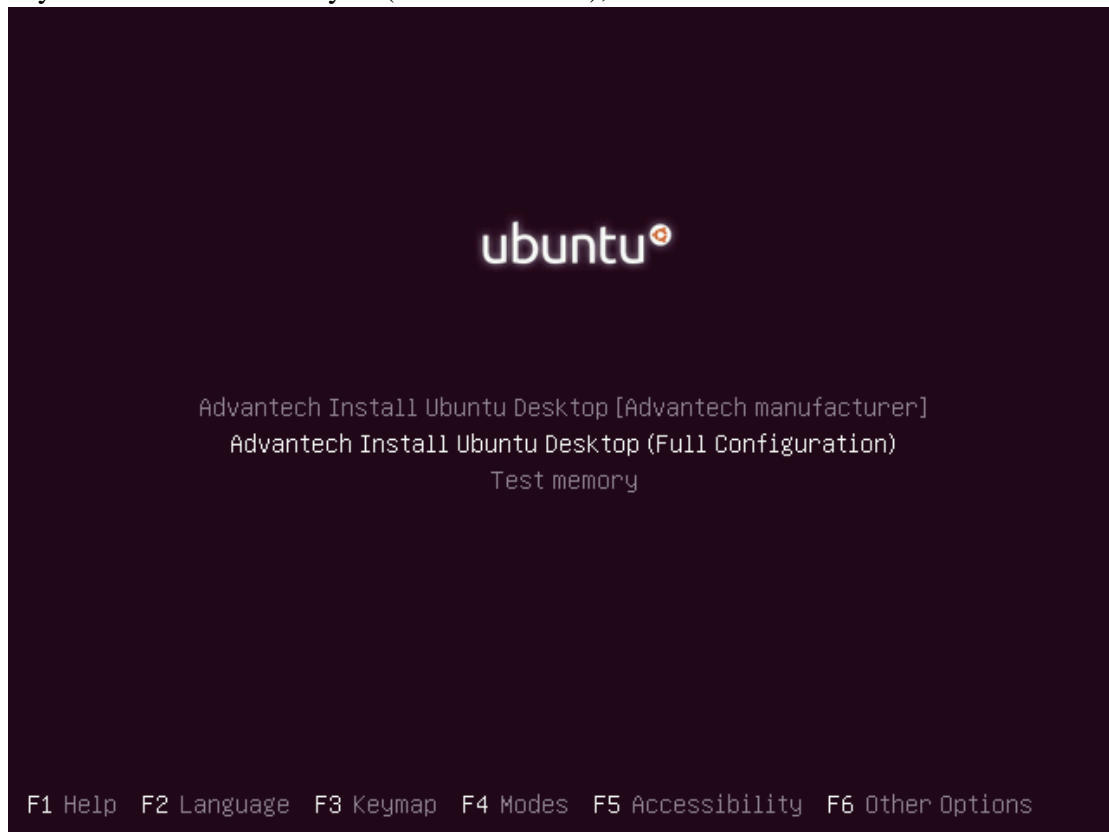


**3.6.1.2. Install Mode Selection**

If you want to install for your end-user (OEM mode), select the first one (default).



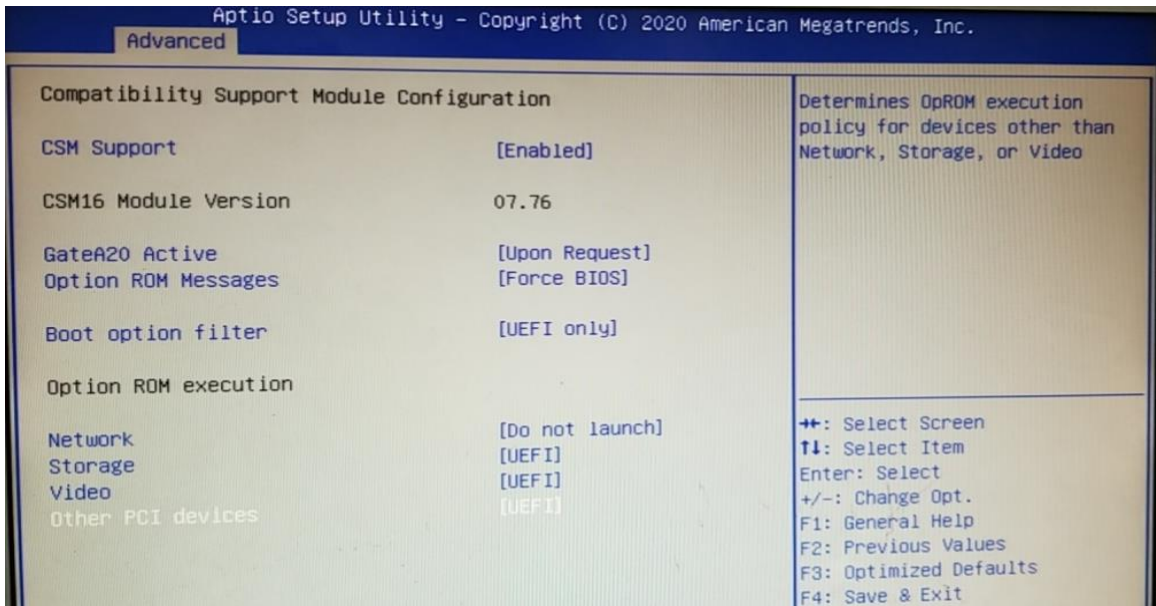
If you want to install for you (End-user mode), select the second one.



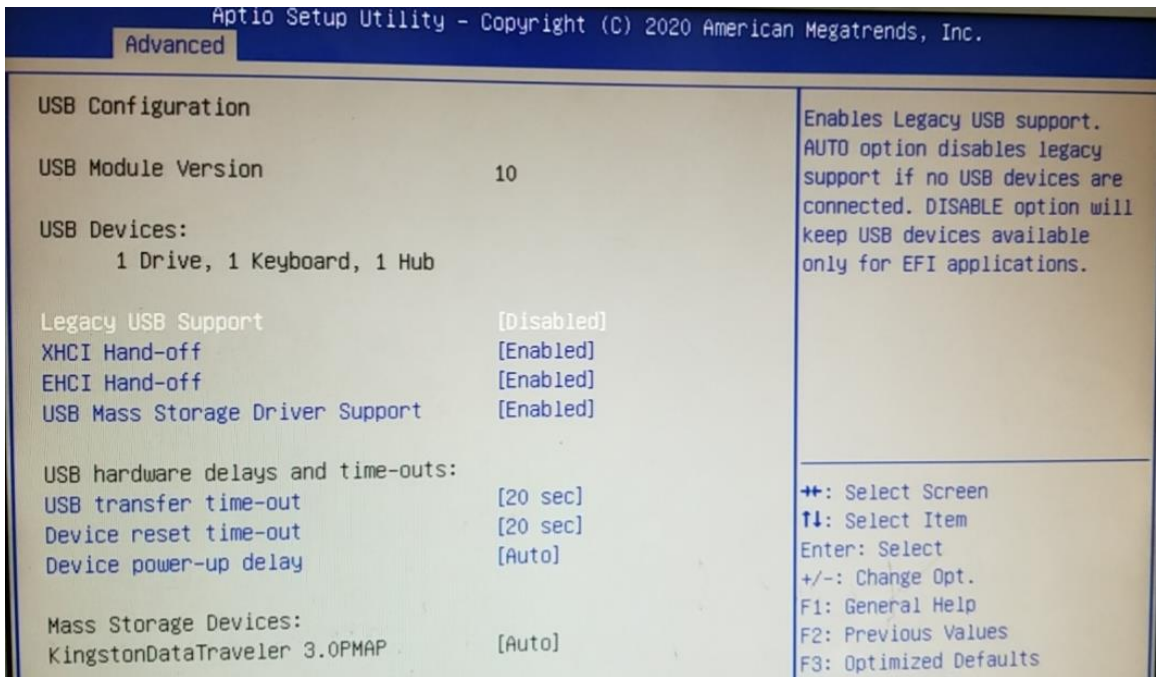
## **3.6.2. UEFI Boot Mode Installation Steps**

### **3.6.2.1. BIOS Configuration**

Step1: CSM configuration

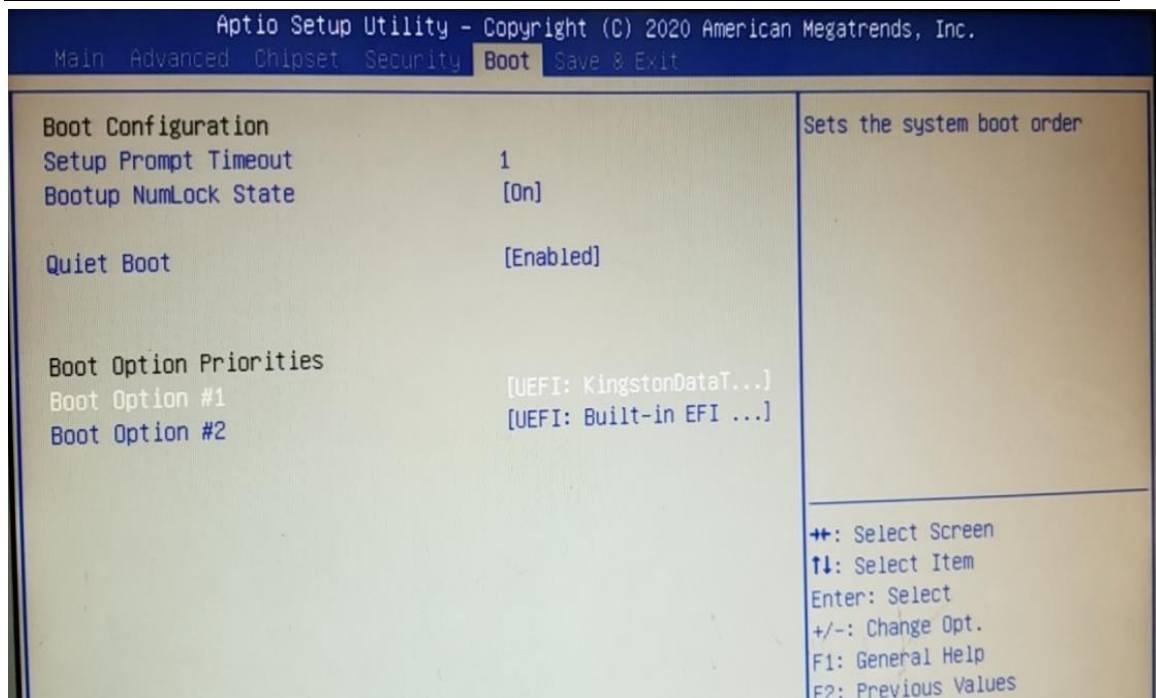


Step2: USB configuration



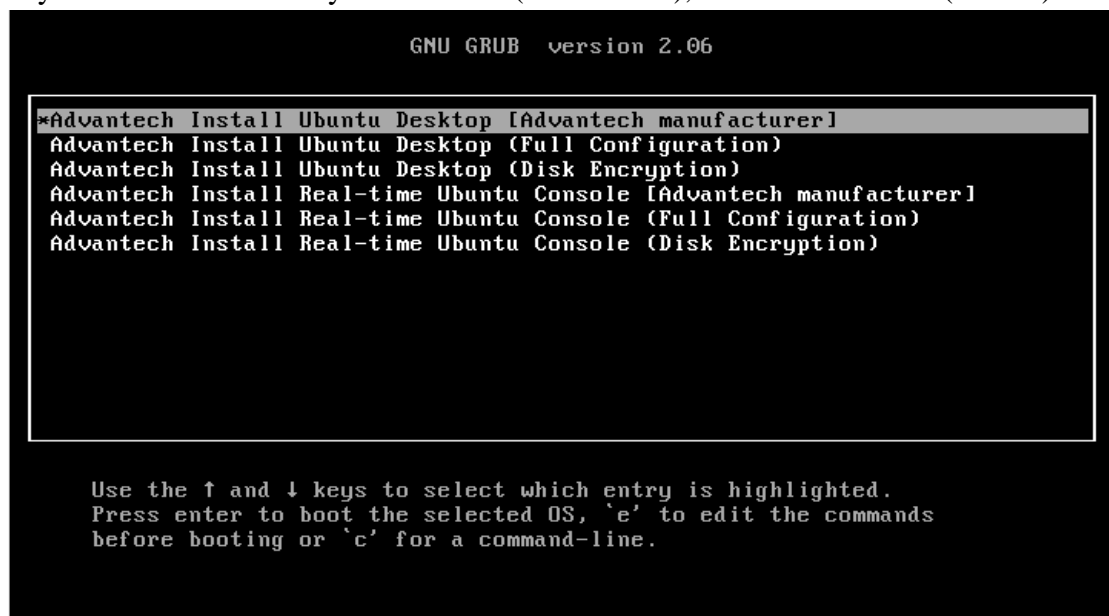
Step3: Boot Option #1 selection





### 3.6.2.2. Install Mode Selection

If you want to install for your end-user (OEM mode), select the first one (default).



If you want to install for you (End-user mode), select the second one.

```

GNU GRUB  version 2.06

Advantech Install Ubuntu Desktop [Advantech manufacturer]
*Advantech Install Ubuntu Desktop (Full Configuration)
Advantech Install Ubuntu Desktop (Disk Encryption)
Advantech Install Real-time Ubuntu Console [Advantech manufacturer]
Advantech Install Real-time Ubuntu Console (Full Configuration)
Advantech Install Real-time Ubuntu Console (Disk Encryption)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.

```

If you want to install for you (real-time kernel), select the fourth to sixth.

**Note:** real-time Ubuntu is console mode

```

GNU GRUB  version 2.06

Advantech Install Ubuntu Desktop [Advantech manufacturer]
Advantech Install Ubuntu Desktop (Full Configuration)
Advantech Install Ubuntu Desktop (Disk Encryption)
*Advantech Install Real-time Ubuntu Console [Advantech manufacturer]
Advantech Install Real-time Ubuntu Console (Full Configuration)
Advantech Install Real-time Ubuntu Console (Disk Encryption)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.

```

If you want set to desktop mode, you can execute command as follow

```
$ sudo systemctl set-default graphical.target
```

```
$ reboot
```

If you want back to console mode, you can execute command as follow

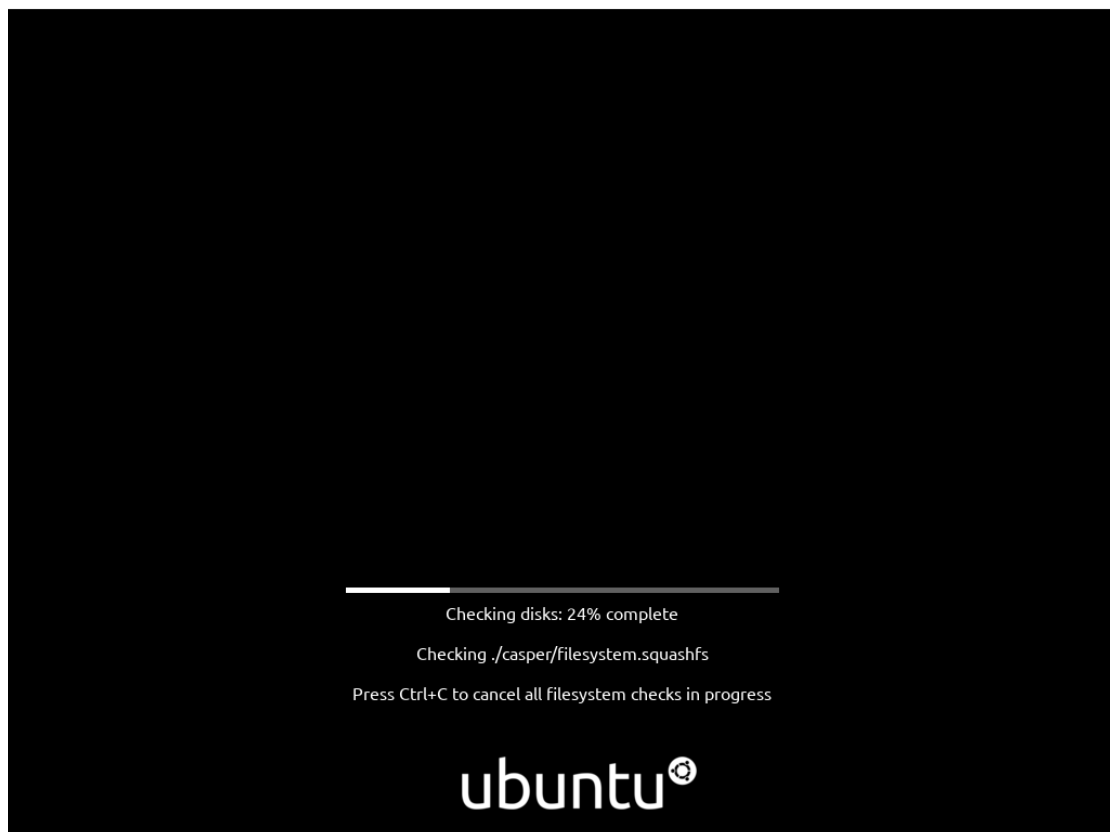
```
$ sudo systemctl set-default multi-user.target
```

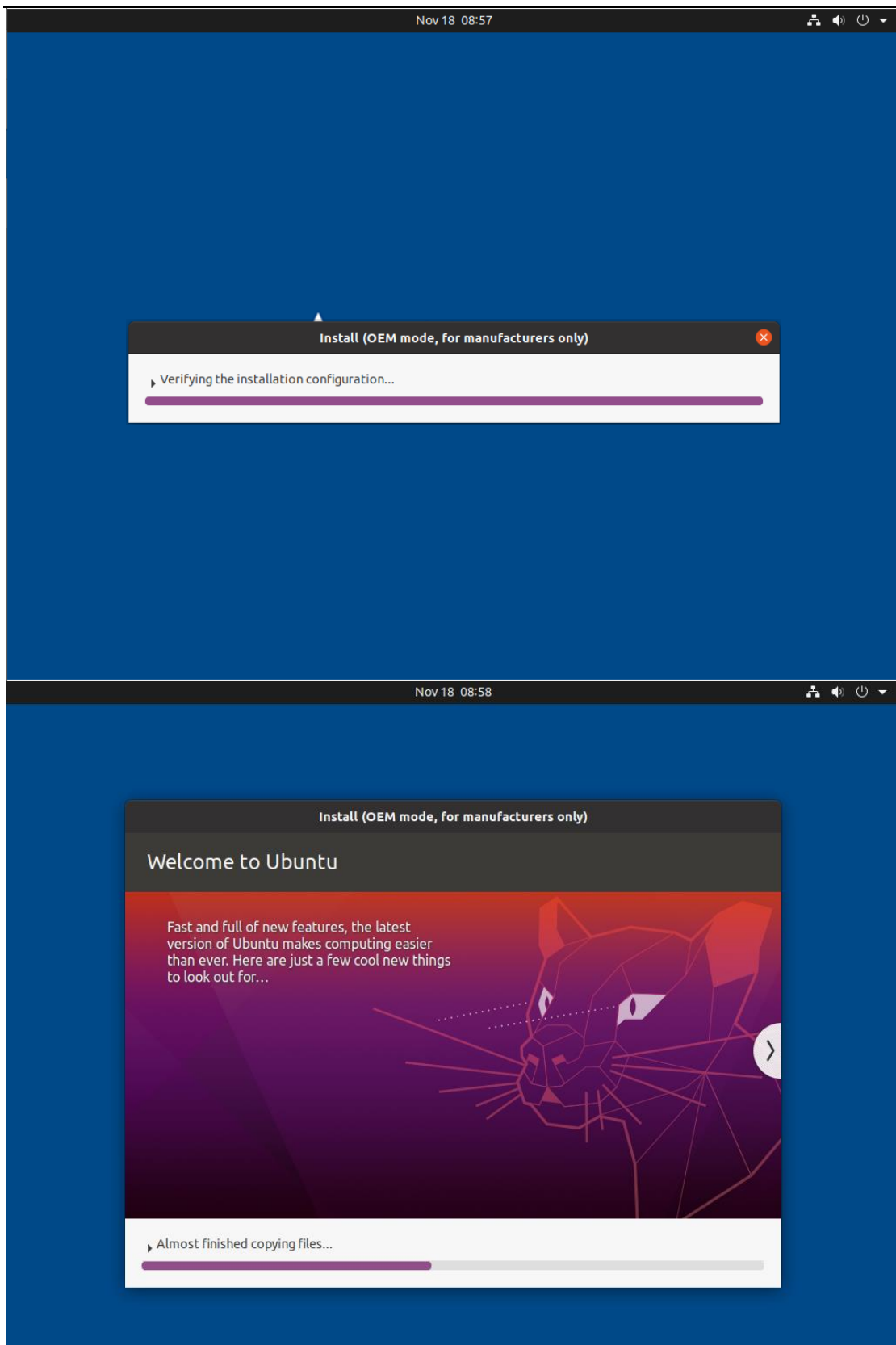
```
$ reboot
```

## OS Installation

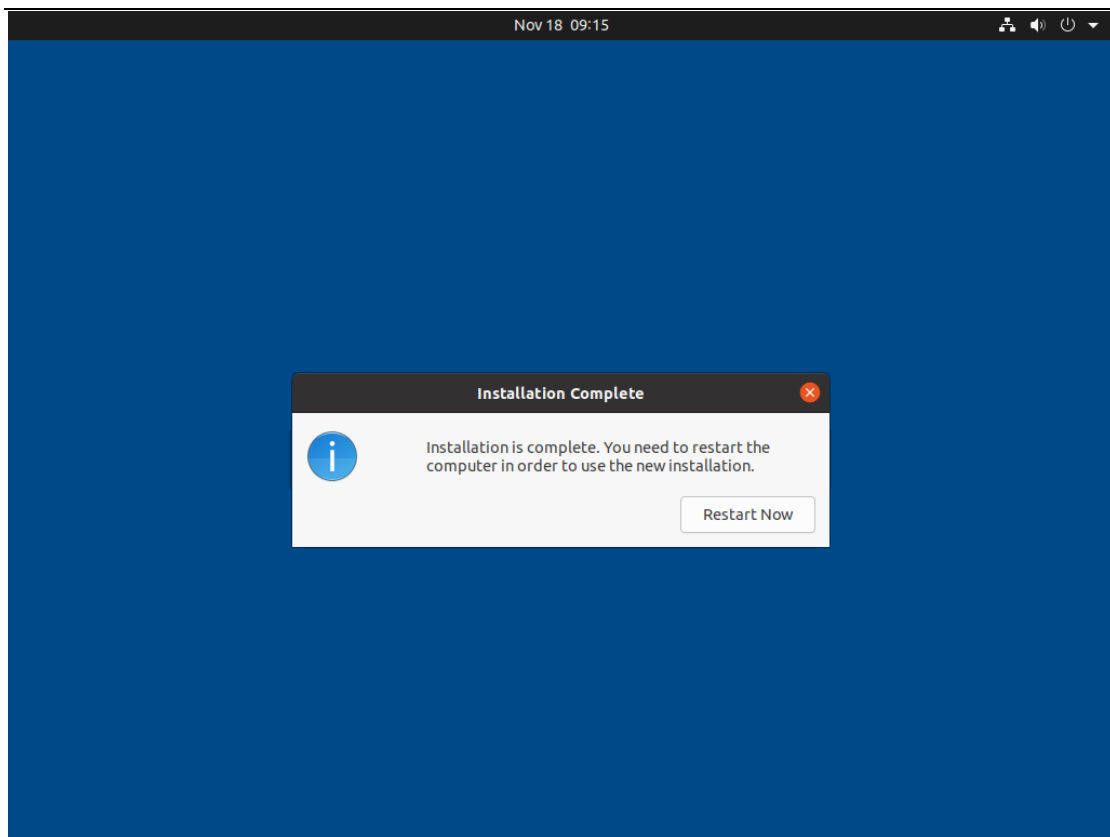
### 3.6.2.3. OEM mode

In OEM mode, the installation will start by itself and the screen will show the pictures in below.

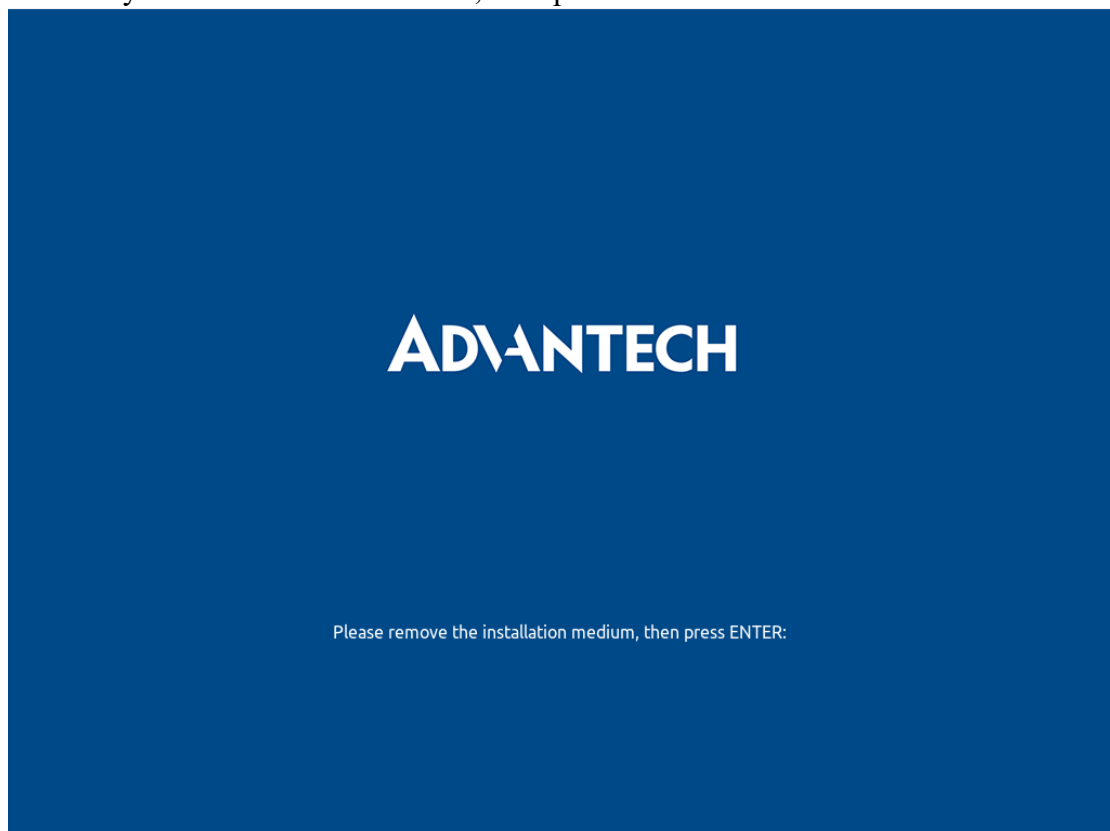




Click "Restart Now" in Installation Complete dialog.



Remove your USB disk from device, then press “ENTER”.



After reboot, the device will initialize itself.

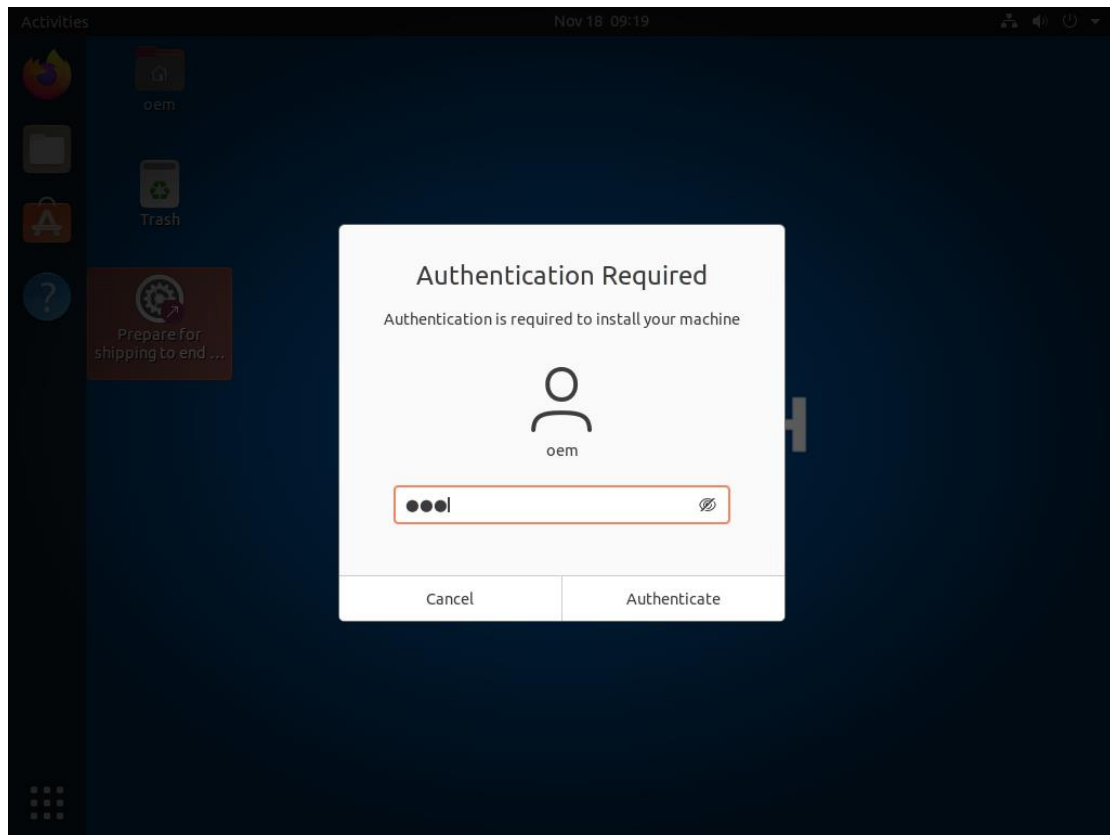


After initialization, the device will login automatically (Login User: oem, Password: oem).

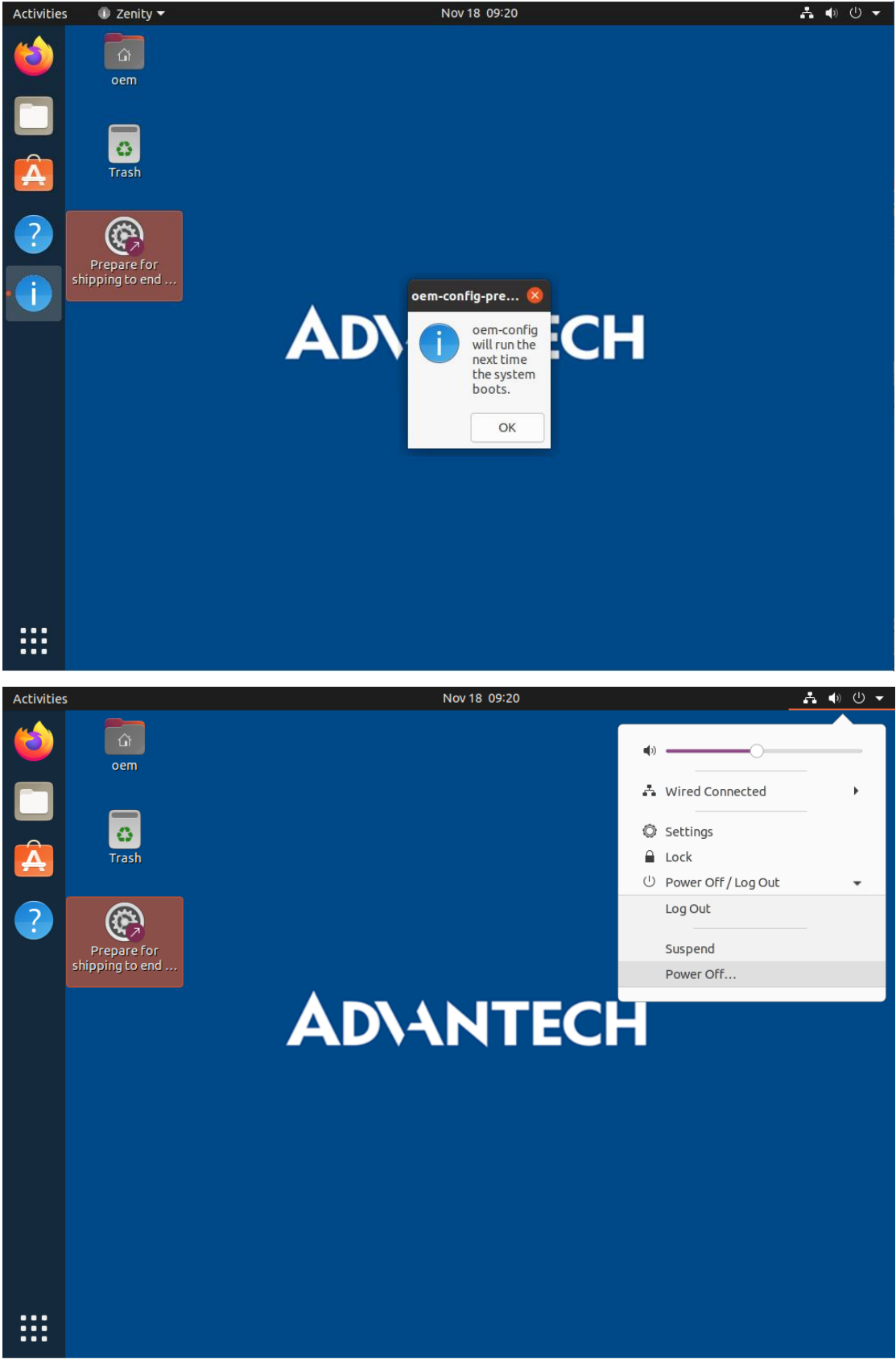


Now you can check anything in the state.

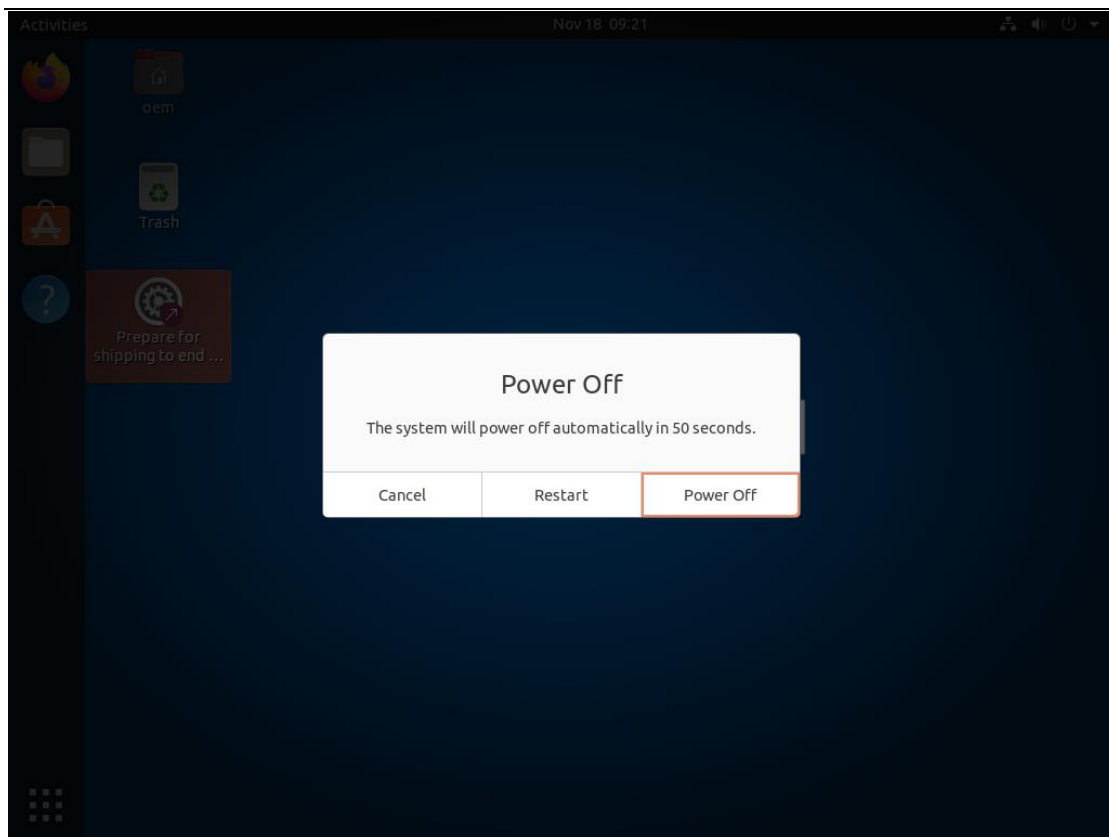
Before shipping to end-user, you need click the icon in desktop.



Click “OK” and poweroff







If the device is UNO-420 or UNO-2271G, the default is console mode. Before shipping to end-user, you need input the command:

```
$ sudo oem-config-prepare
$ poweroff
```

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Thu Nov 18 08:21:45 2021
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

oem@ubuntu:~$ sudo oem-config-prepare
[sudo] password for oem:
Created symlink /etc/systemd/system/oem-config.target.wants/oem-config.service - /lib/systemd/system/oem-config.service.
The unit files have no installation config (WantedBy=, RequiredBy=, Also=,
Alias= settings in the [Install] section, and DefaultInstance= for template
units). This means they are not meant to be enabled using systemctl.

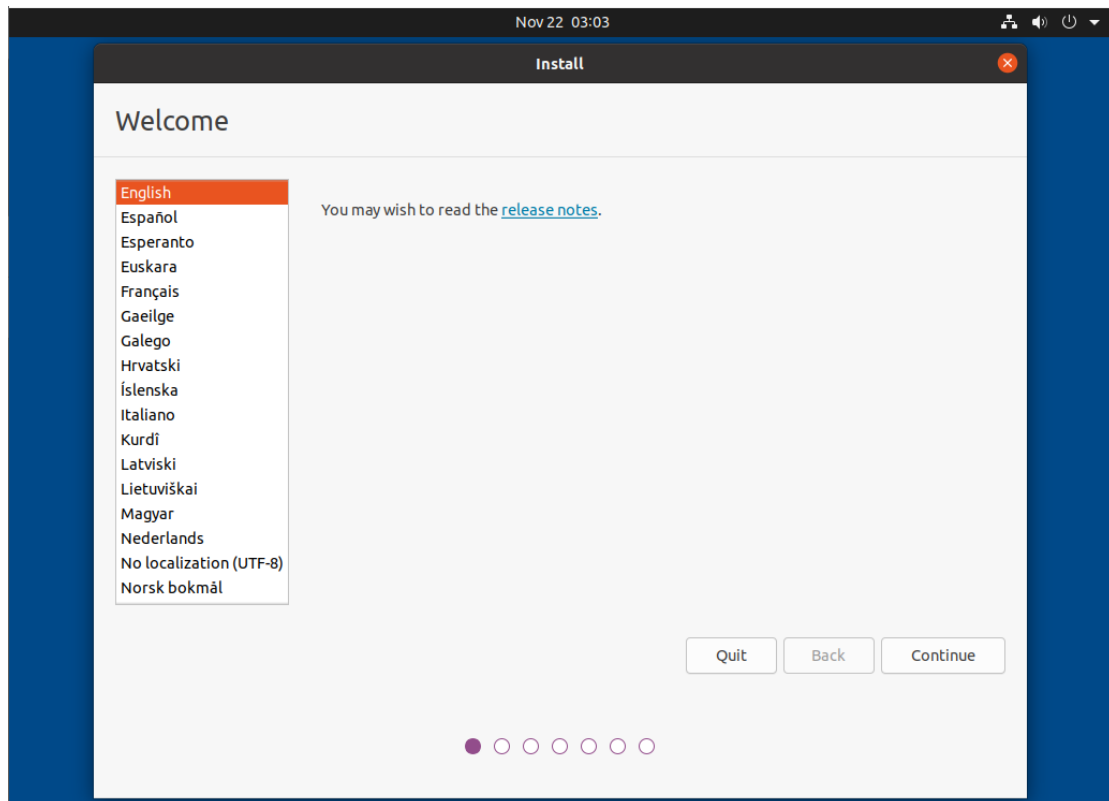
Possible reasons for having this kind of units are:
* A unit may be statically enabled by being symlinked from another unit's
  .wants/ or .requires/ directory.
* A unit's purpose may be to act as a helper for some other unit which has
  a requirement dependency on it.
* A unit may be started when needed via activation (socket, path, timer,
  D-Bus, udev, scripted systemctl call, ...).
* In case of template units, the unit is meant to be enabled with some
  instance name specified.
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target - /lib/systemd/system/oem-config.target.
oem-config will run the next time the system boots.
oem@ubuntu:~$ poweroff
```

After the steps, the device is shutdown. Now you can ship the device to end-user.

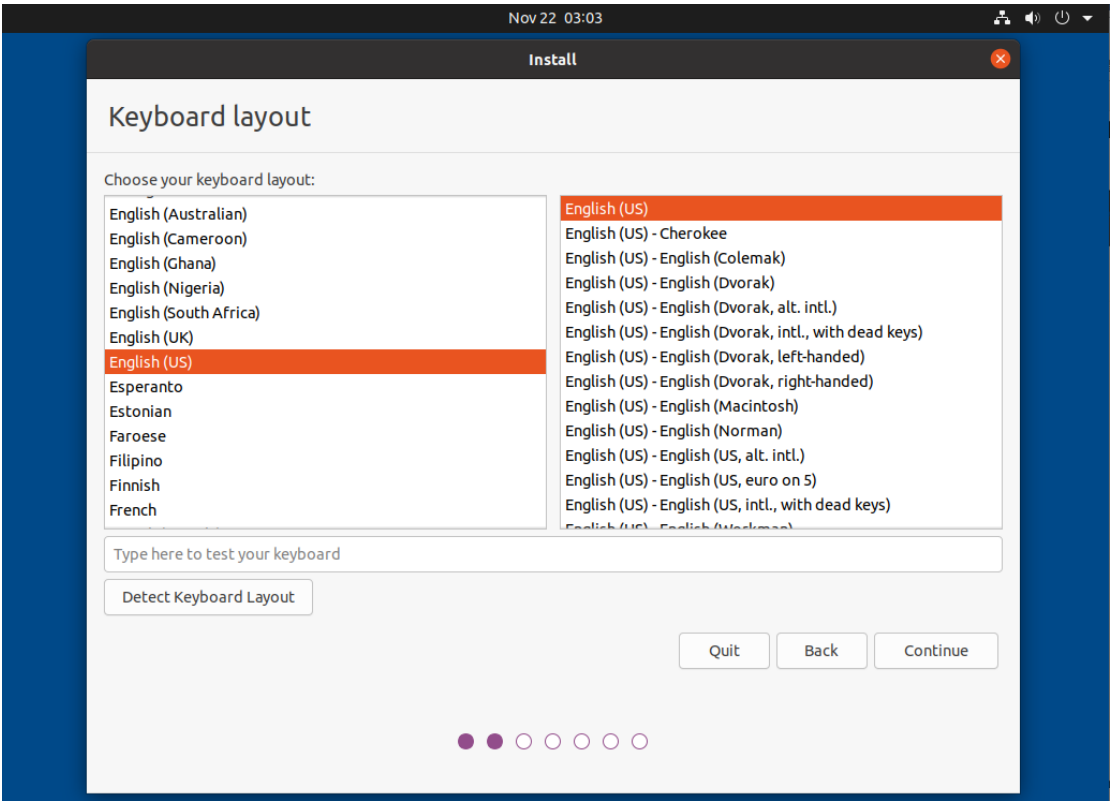
## 3.6.2.4. End-user mode

In End-user mode, you need choose settings in installation.

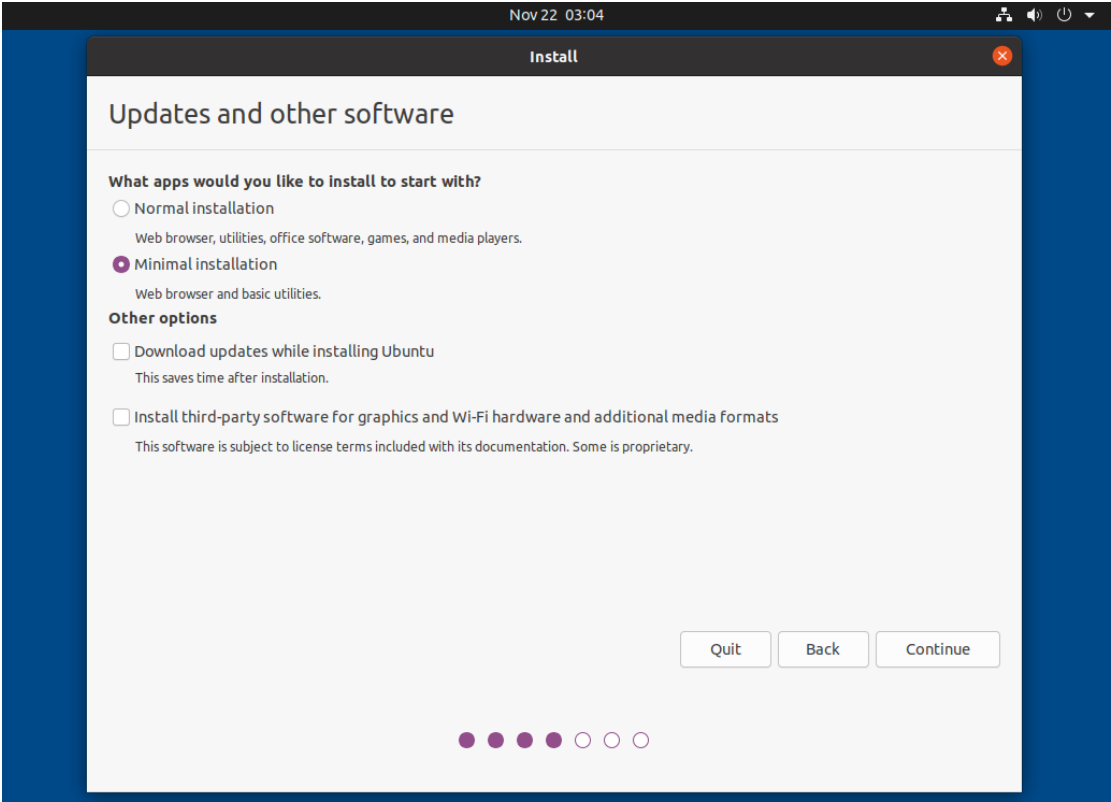
Select installation language you want.



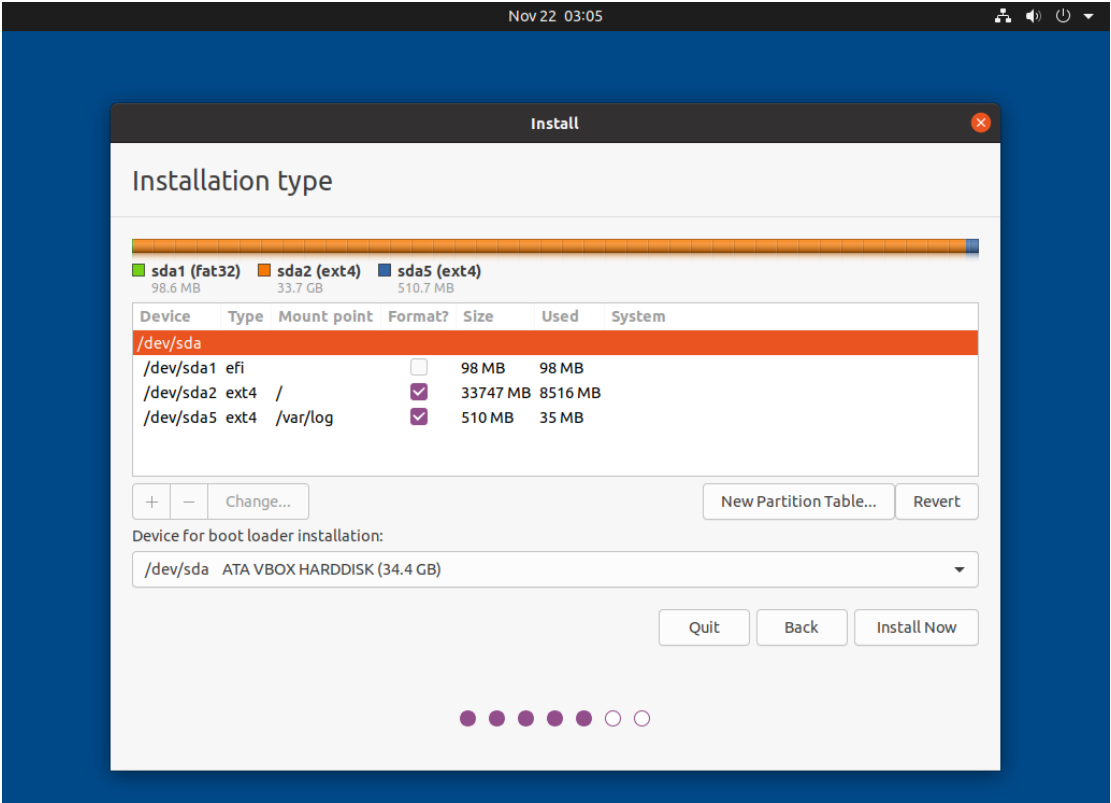
Select current keyboard layout.



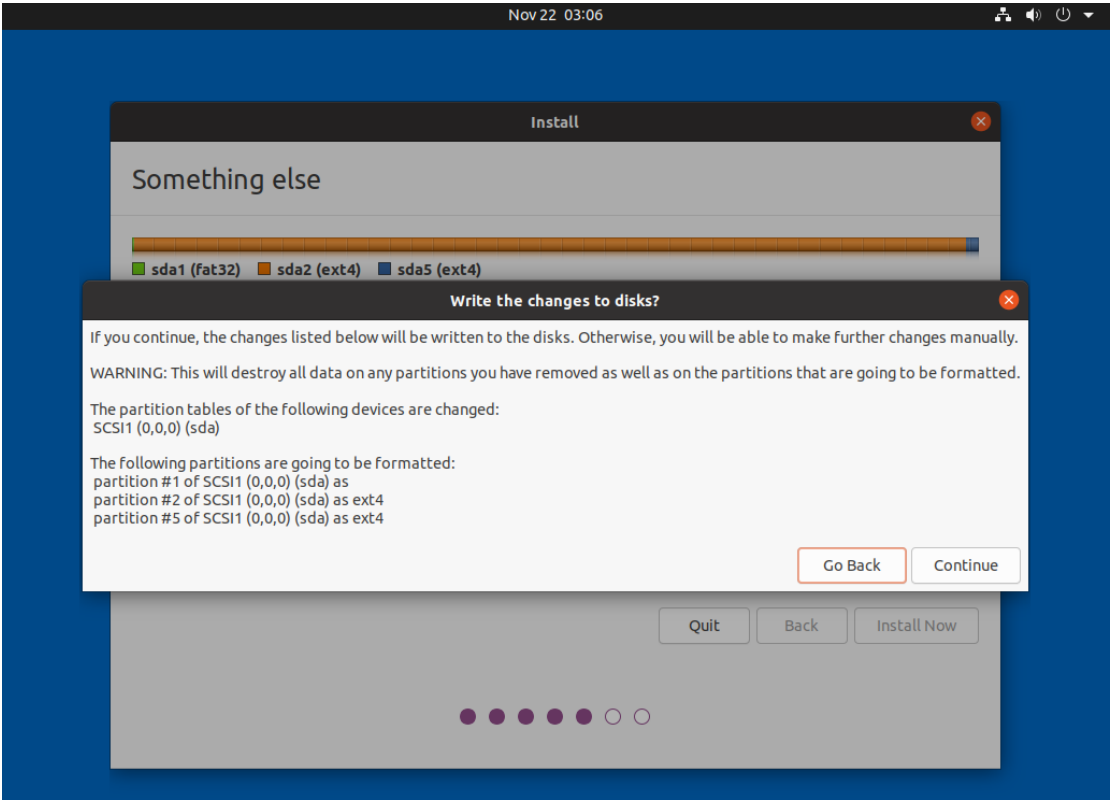
We recommend minimal installation, you can change it.



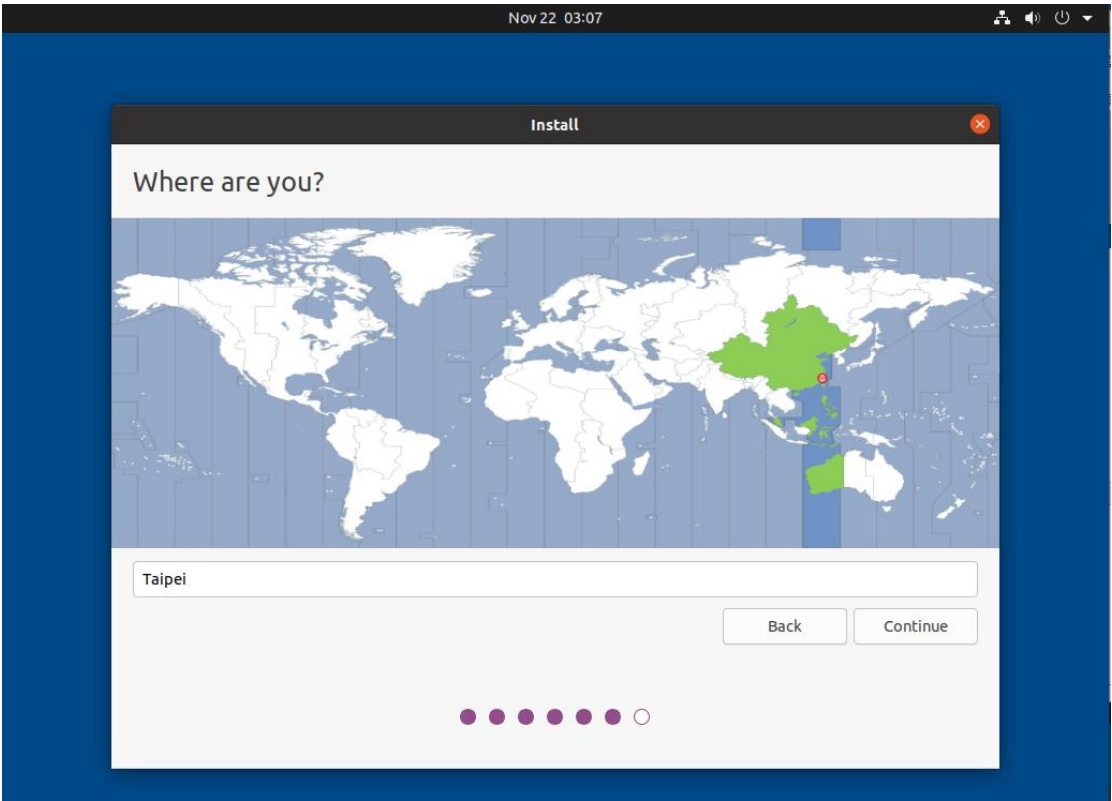
This is recommending for preset partitions, you can change it.



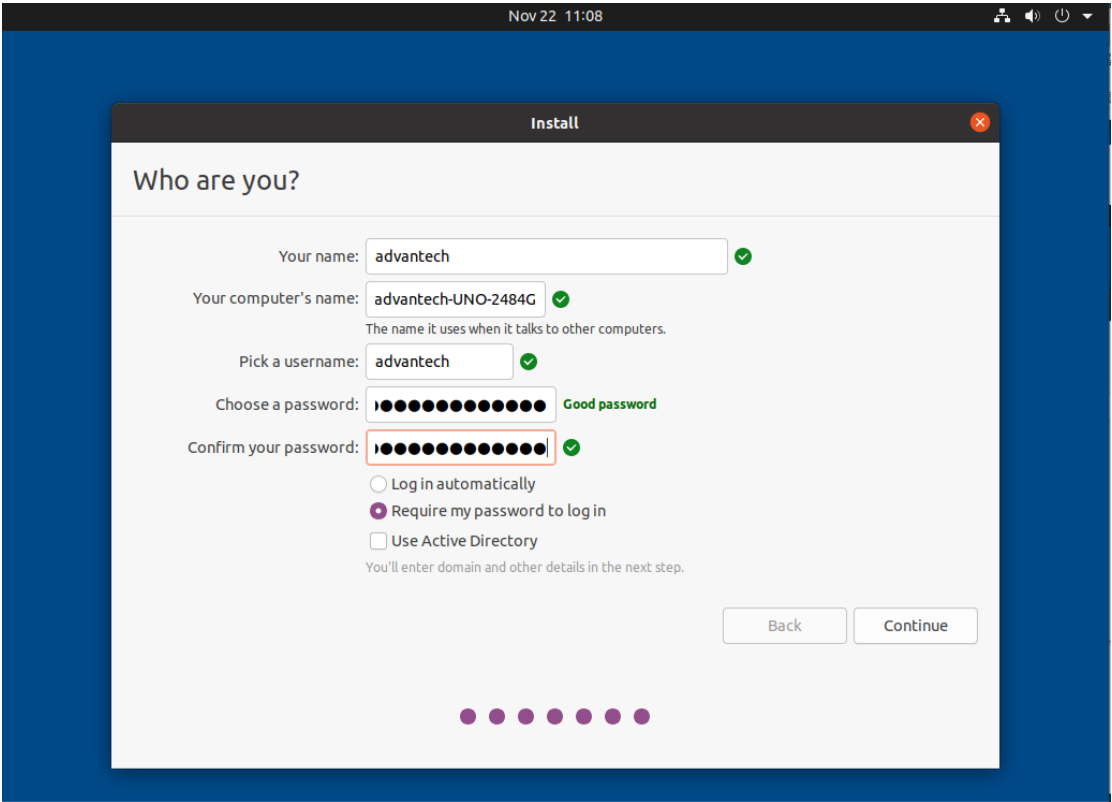
Reconfirm the settings of disk. You can click “Continue” to next step.



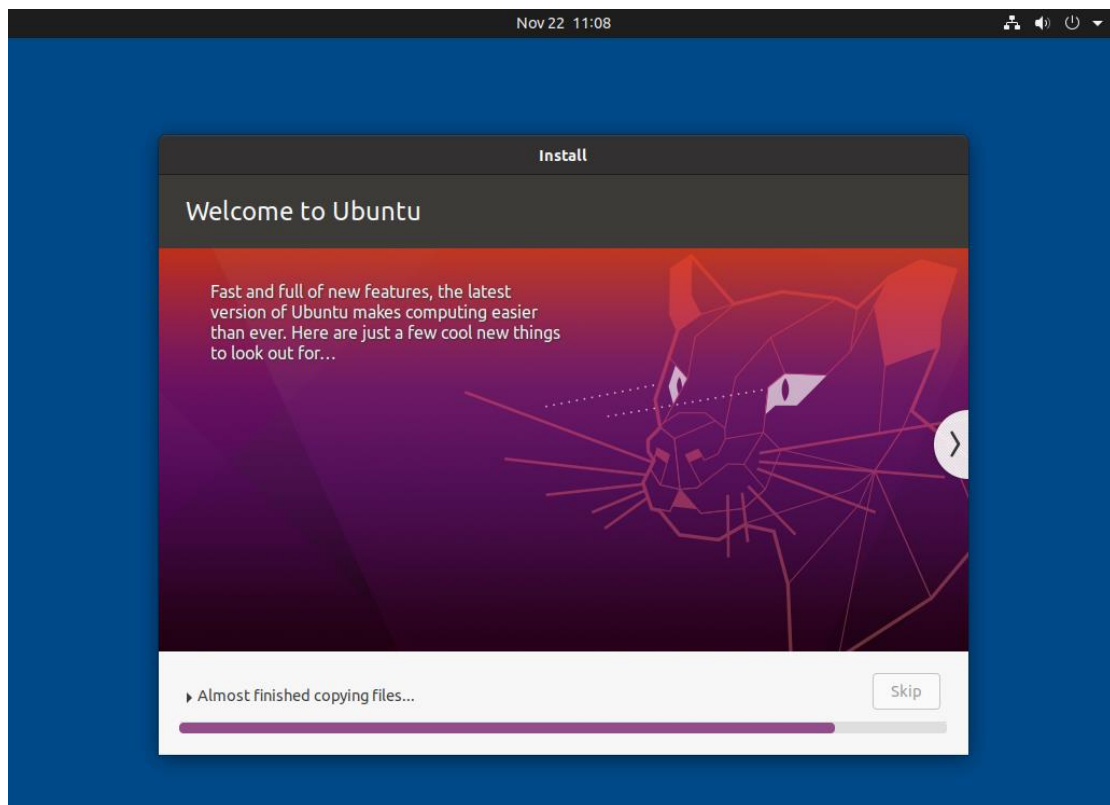
Choose your location.



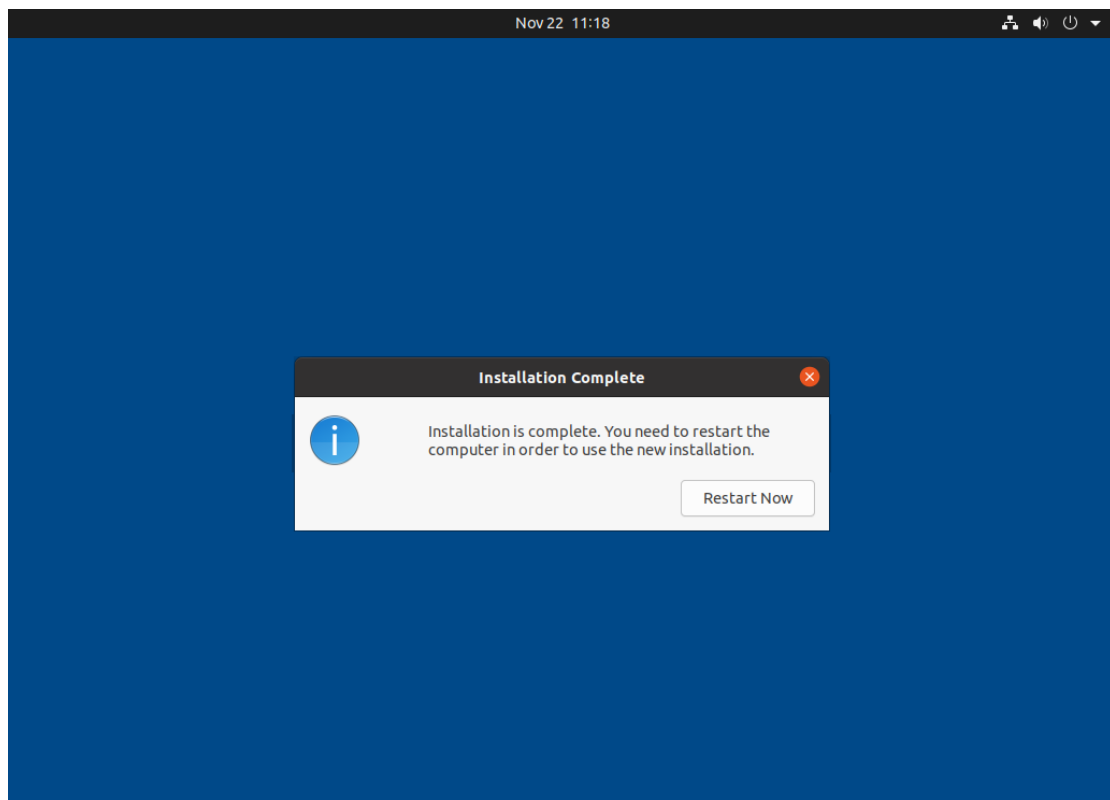
Setting the device information and your account.



After settings, the installation will start. It needs take some times.



Click “Restart Now” in Installation Complete dialog.



Remove your USB disk from device, then press “ENTER”.



# ADVANTECH

Please remove the installation medium, then press ENTER:

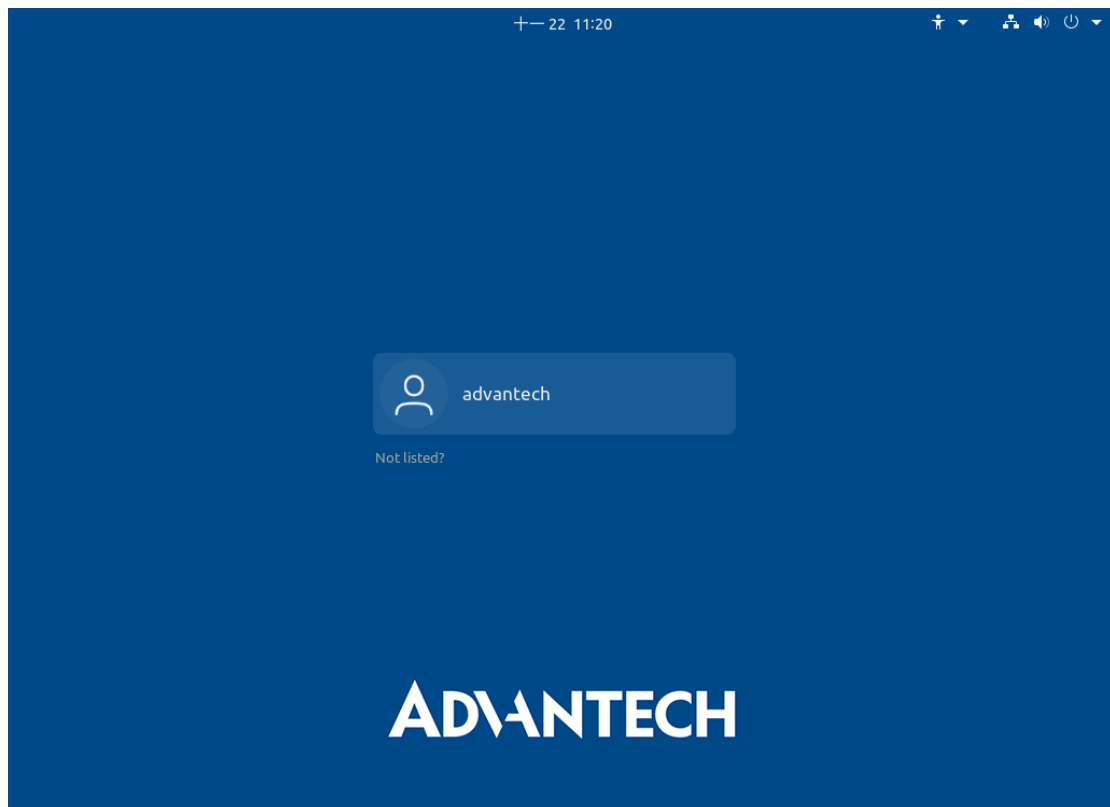
After reboot, the device will initialize itself.



# ADVANTECH

  
Device is initializing...Wait

After initialization, you can login it.



If the device is UNO-420 or UNO-2271G, the default is console mode.

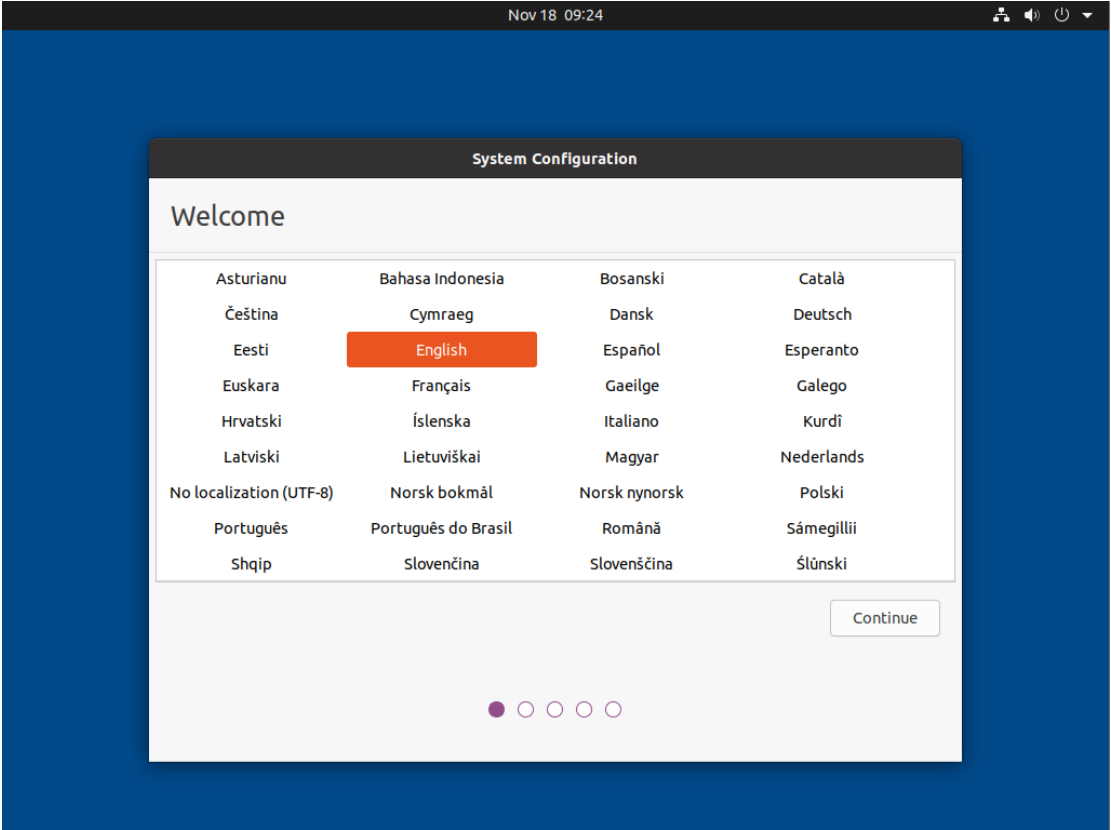
```
Ubuntu 20.04.3 LTS advantech-UNO-420 tty1
advantech-UNO-420 login: _
```

### 3.6.2.5. Shipping from OEM

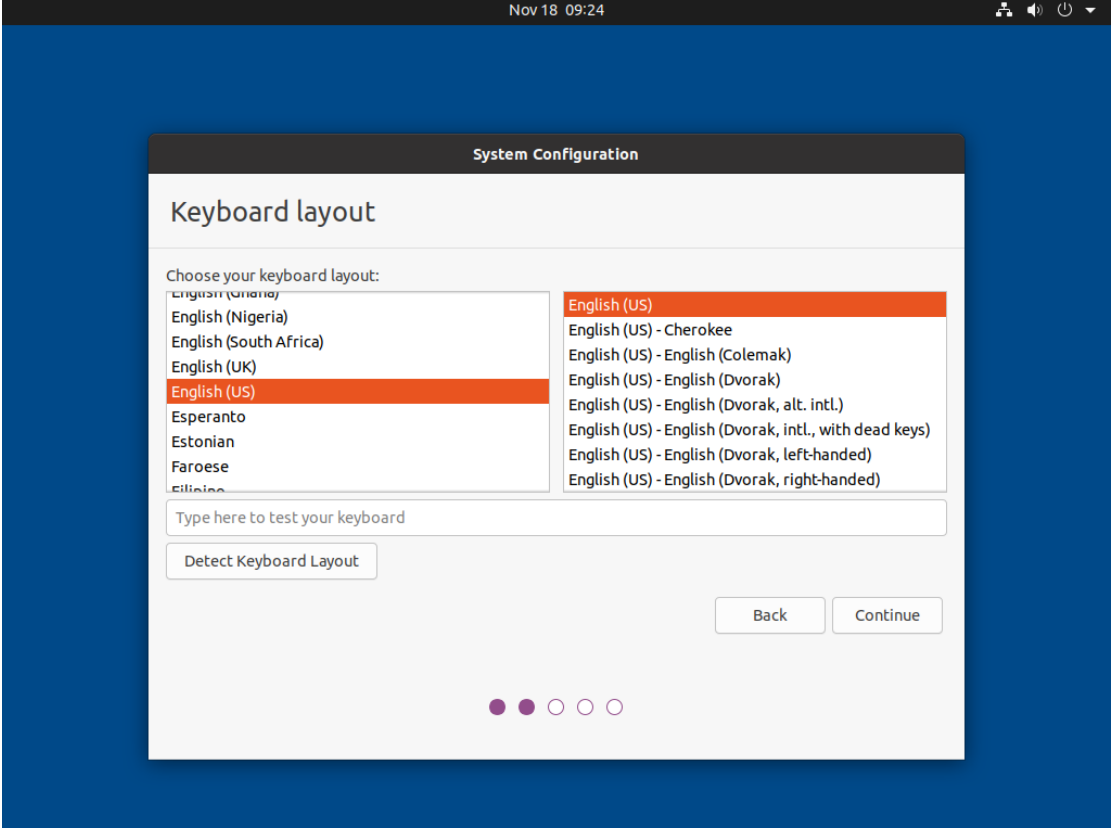
If you are not install OS to the device, you can power on the device and setting the device you want.



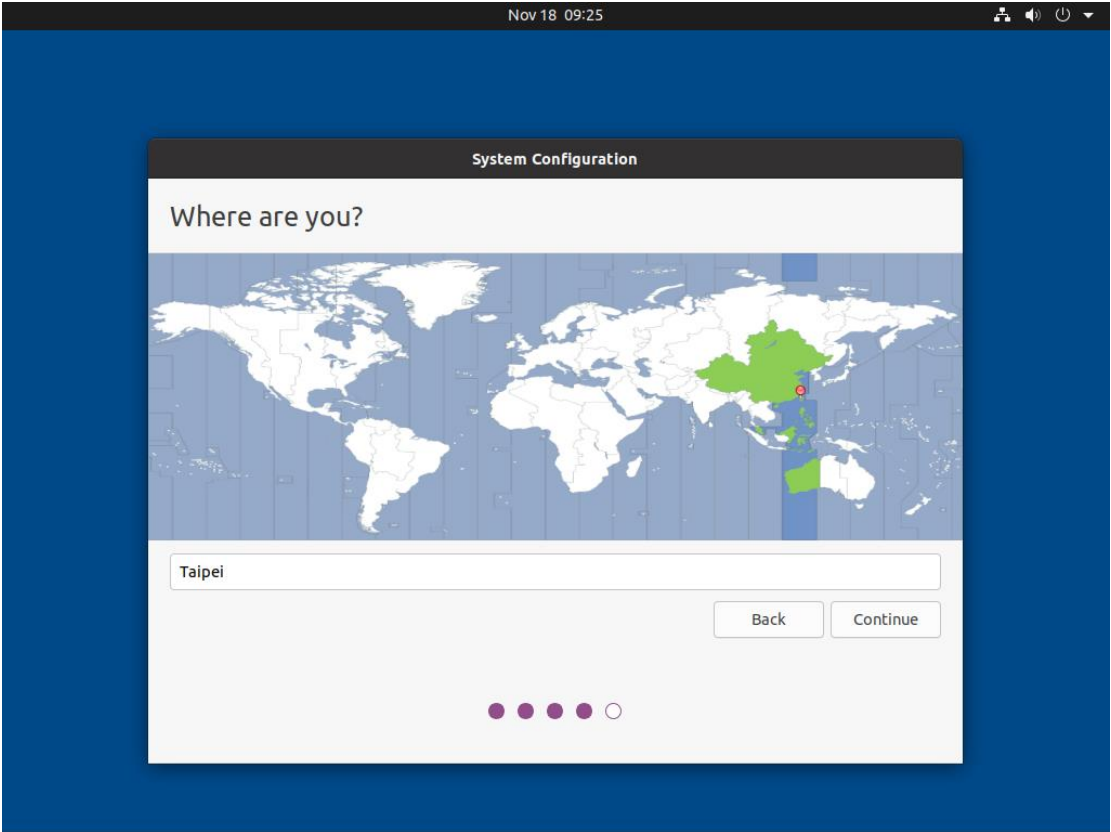
Select your Language.



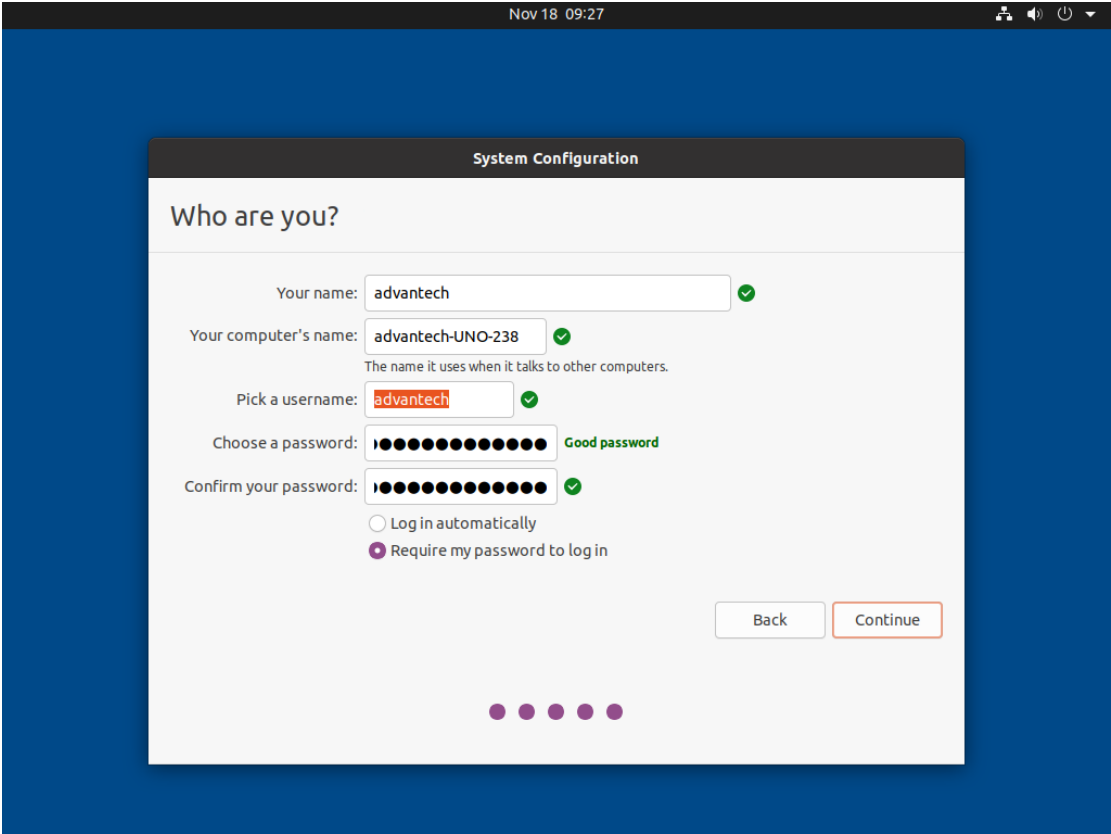
Select current keyboard layout.



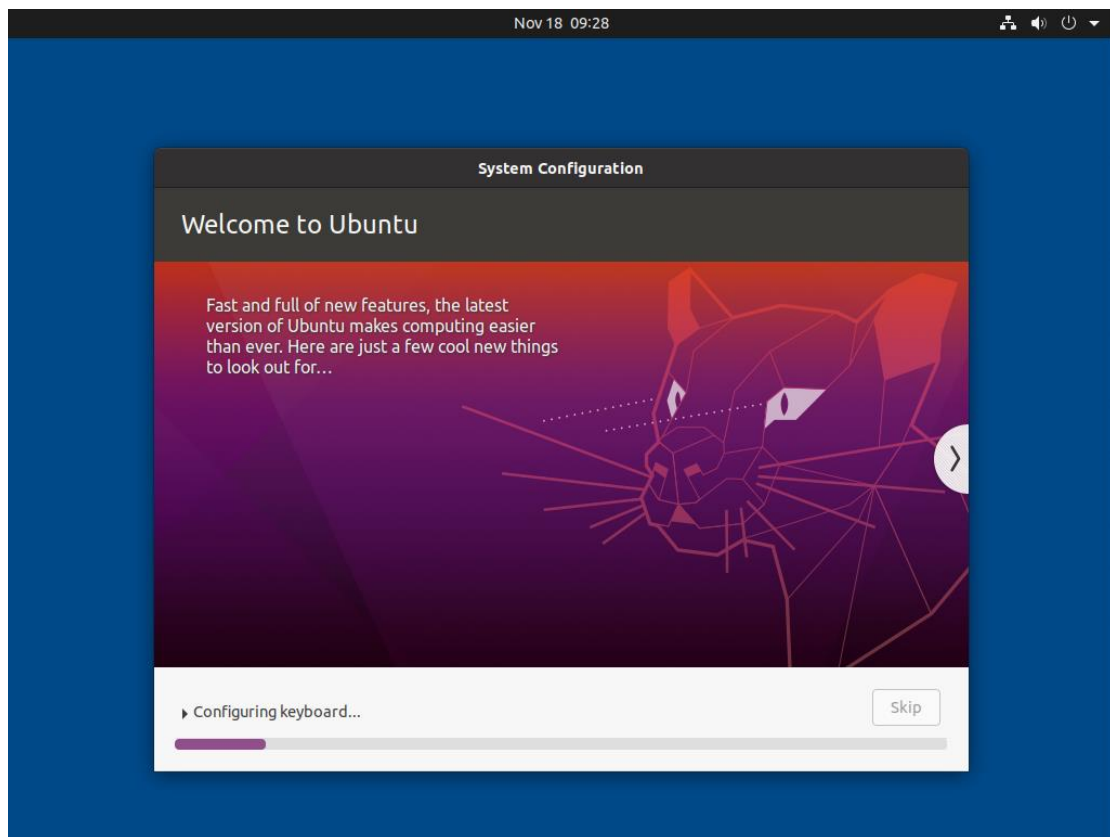
Confirm your location.



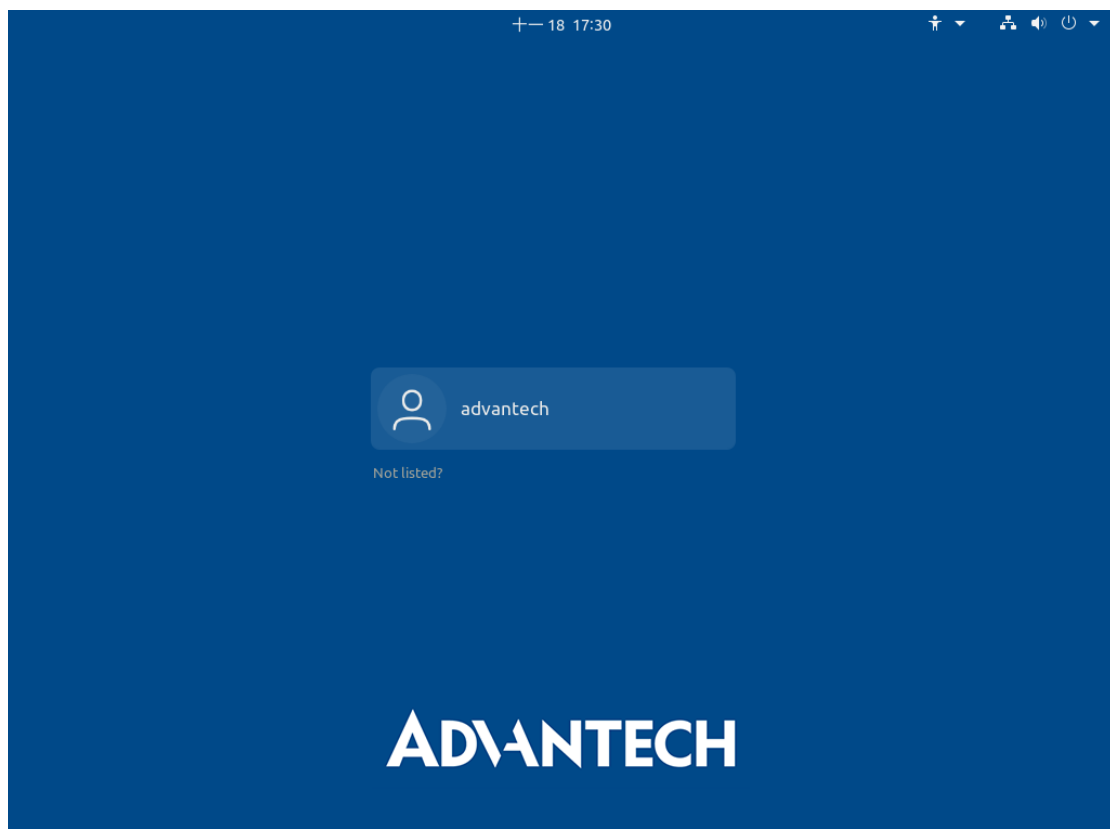
Setting the device information and your account.



It will configure your settings

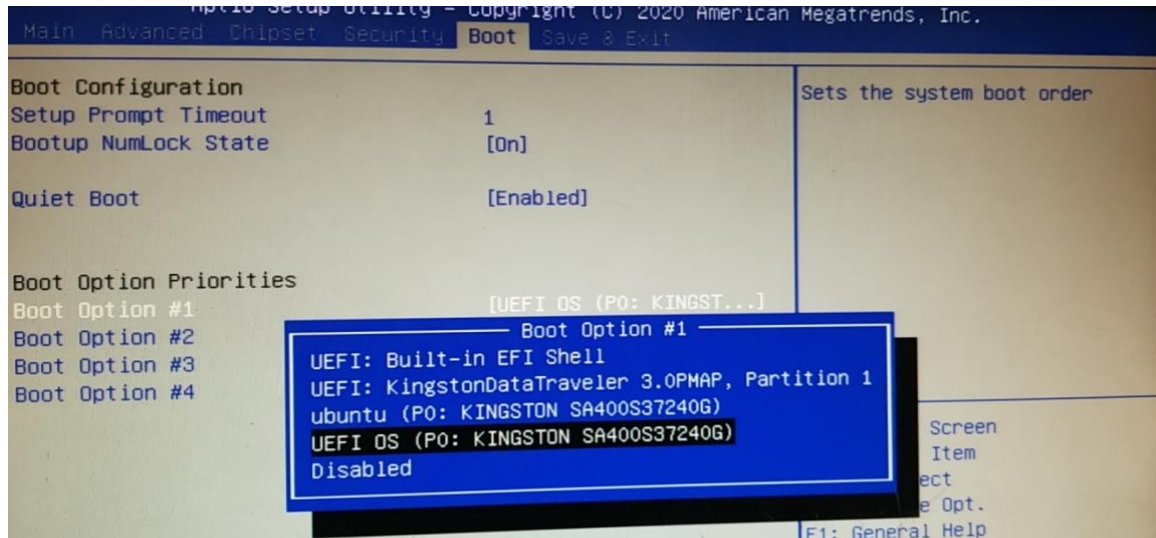


After configuration, you can login it.



### 3.6.2.6. Change Boot Option #1

Enter Reboot and go into BIOS, You can see there are two UEFI Boot hard disk. Ubuntu (P0: KINGSTON SA400S37240G) and UEFI OS (P0: KINGSTON SA400S37240G). You can choose both of them.



## 4. Advantech device driver

AdvLinuxTU supports Advantech products, and the related drivers, the related drivers will be installed in `/usr/src/advantech` and auto loaded during the booting process. Here are some examples or demos in `/usr/src/advantech/${driver name}/example/`.

If you want use Advantech device driver, you need to check secure boot is disable in BIOS.

## 5. GPIO

Our devices default use GPIO Sysfs Interface to control GPIOs. AdvLinuxTU 4.x provide two method to control gpio:

1. [Sysfs Interface](#)
2. [libgpio](#)

### 5.1. Sysfs Interface

You can get more detail from

<https://www.kernel.org/doc/Documentation/gpio/sysfs.txt>.

We list the mapping tables of devices as below.

For example, for GPIO0 of UNO-420.(kernel:5.11)

You can get the gpio pin direction:

```
$ cat /sys/class/gpio/gpio504/direction
```

The output is “in” as DI or “out” as DO.

You can set the gpio pin direction:

```
$ echo in > /sys/class/gpio/gpio504/direction //set as DI
```

```
$ echo out > /sys/class/gpio/gpio504/direction //set as DO
```

You can get the status of gpio pin:

```
$ cat /sys/class/gpio/gpio504/value
```

The output is “0” as low or “1” as high.

You can set the status of gpio pin when direction is DO

```
$ echo 0 > /sys/class/gpio/gpio504/value //set to low
```

```
$ echo 1 > /sys/class/gpio/gpio504/value //set to high
```

#### 5.1.1. UNO-238

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
GPIO0	/sys/class/gpio/gpio192	/sys/class/gpio/gpio704
GPIO1	/sys/class/gpio/gpio193	/sys/class/gpio/gpio705
GPIO2	/sys/class/gpio/gpio194	/sys/class/gpio/gpio706
GPIO3	/sys/class/gpio/gpio195	/sys/class/gpio/gpio707

GPIO4	/sys/class/gpio/gpio196	/sys/class/gpio/gpio708
GPIO5	/sys/class/gpio/gpio197	/sys/class/gpio/gpio709
GPIO6	/sys/class/gpio/gpio198	/sys/class/gpio/gpio710
GPIO7	/sys/class/gpio/gpio199	/sys/class/gpio/gpio711

## 5.1.2. UNO-420

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
GPIO0	/sys/class/gpio/gpio504	/sys/class/gpio/gpio1016
GPIO1	/sys/class/gpio/gpio505	/sys/class/gpio/gpio1017
GPIO2	/sys/class/gpio/gpio506	/sys/class/gpio/gpio1018
GPIO3	/sys/class/gpio/gpio507	/sys/class/gpio/gpio1019
GPIO4	/sys/class/gpio/gpio508	/sys/class/gpio/gpio1020
GPIO5	/sys/class/gpio/gpio509	/sys/class/gpio/gpio1021
GPIO6	/sys/class/gpio/gpio510	/sys/class/gpio/gpio1022
GPIO7	/sys/class/gpio/gpio511	/sys/class/gpio/gpio1023

## 5.1.3. UNO-137

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
DI0	/sys/class/gpio/gpio451	/sys/class/gpio/gpio963
DI1	/sys/class/gpio/gpio453	/sys/class/gpio/gpio964
DI2	/sys/class/gpio/gpio454	/sys/class/gpio/gpio965
DI3	/sys/class/gpio/gpio455	/sys/class/gpio/gpio966
DI4	/sys/class/gpio/gpio456	/sys/class/gpio/gpio967
DI5	/sys/class/gpio/gpio457	/sys/class/gpio/gpio968
DI6	/sys/class/gpio/gpio458	/sys/class/gpio/gpio969
DI7	/sys/class/gpio/gpio451	/sys/class/gpio/gpio970
DO0	/sys/class/gpio/gpio452	/sys/class/gpio/gpio971
DO1	/sys/class/gpio/gpio459	/sys/class/gpio/gpio972
DO2	/sys/class/gpio/gpio460	/sys/class/gpio/gpio973
DO3	/sys/class/gpio/gpio461	/sys/class/gpio/gpio974
DO4	/sys/class/gpio/gpio462	/sys/class/gpio/gpio975
DO5	/sys/class/gpio/gpio463	/sys/class/gpio/gpio976
DO6	/sys/class/gpio/gpio464	/sys/class/gpio/gpio977
DO7	/sys/class/gpio/gpio465	/sys/class/gpio/gpio978

## 5.1.4. UNO-148

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
DI0	/sys/class/gpio/gpio424	/sys/class/gpio/gpio936
DI1	/sys/class/gpio/gpio425	/sys/class/gpio/gpio937
DI2	/sys/class/gpio/gpio426	/sys/class/gpio/gpio938
DI3	/sys/class/gpio/gpio427	/sys/class/gpio/gpio939
DI4	/sys/class/gpio/gpio428	/sys/class/gpio/gpio940
DI5	/sys/class/gpio/gpio429	/sys/class/gpio/gpio941
DI6	/sys/class/gpio/gpio430	/sys/class/gpio/gpio942
DI7	/sys/class/gpio/gpio431	/sys/class/gpio/gpio943
DO0	/sys/class/gpio/gpio314	/sys/class/gpio/gpio826
DO1	/sys/class/gpio/gpio315	/sys/class/gpio/gpio827
DO2	/sys/class/gpio/gpio325	/sys/class/gpio/gpio837
DO3	/sys/class/gpio/gpio326	/sys/class/gpio/gpio838
DO4	/sys/class/gpio/gpio327	/sys/class/gpio/gpio839
DO5	/sys/class/gpio/gpio328	/sys/class/gpio/gpio840
DO6	/sys/class/gpio/gpio329	/sys/class/gpio/gpio841
DO7	/sys/class/gpio/gpio330	/sys/class/gpio/gpio842
Programmable LED	/sys/class/gpio/gpio65	

## 5.1.5. UNO-348

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
GPIO0	/sys/class/gpio/gpio40	
GPIO1	/sys/class/gpio/gpio41	
GPIO2	/sys/class/gpio/gpio42	
GPIO3	/sys/class/gpio/gpio43	
GPIO4	/sys/class/gpio/gpio44	
GPIO5	/sys/class/gpio/gpio45	
GPIO6	/sys/class/gpio/gpio46	
GPIO7	/sys/class/gpio/gpio47	
Programmable LED	/sys/class/gpio/gpio65	



## 5.1.6. UNO-2484G\_V2

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
GPIO0	/sys/class/gpio/gpio81	
GPIO1	/sys/class/gpio/gpio82	
GPIO2	/sys/class/gpio/gpio42	
GPIO3	/sys/class/gpio/gpio43	
GPIO4	/sys/class/gpio/gpio44	
GPIO5	/sys/class/gpio/gpio45	
GPIO6	/sys/class/gpio/gpio46	
GPIO7	/sys/class/gpio/gpio47	

## 5.1.7. UNO-127 / UNO-2271G\_V2 / AMAX-5570

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
Programmable LED	/sys/class/gpio/gpio964	

## 5.1.8. UNO-238\_V2

	Kernel: older 5.15
GPIO0	/sys/class/gpio/gpio648
GPIO1	/sys/class/gpio/gpio649
GPIO2	/sys/class/gpio/gpio650
GPIO3	/sys/class/gpio/gpio651
GPIO4	/sys/class/gpio/gpio652
GPIO5	/sys/class/gpio/gpio653
GPIO6	/sys/class/gpio/gpio654
GPIO7	/sys/class/gpio/gpio655

## 5.1.9. UNO-137\_V2

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
DI0	/sys/class/gpio/gpio30	
DI1	/sys/class/gpio/gpio31	
DI2	/sys/class/gpio/gpio32	
DI3	/sys/class/gpio/gpio33	
DI4	/sys/class/gpio/gpio34	
DI5	/sys/class/gpio/gpio35	
DI6	/sys/class/gpio/gpio36	
DI7	/sys/class/gpio/gpio37	
DO0	/sys/class/gpio/gpio40	
DO1	/sys/class/gpio/gpio41	
DO2	/sys/class/gpio/gpio42	
DO3	/sys/class/gpio/gpio43	
DO4	/sys/class/gpio/gpio44	
DO5	/sys/class/gpio/gpio45	
DO6	/sys/class/gpio/gpio46	
DO7	/sys/class/gpio/gpio47	

## 5.1.10. AMAX-5570S

	Kernel: older 5.15	Kernel: 5.15-1026 & 5.19
DO0	/sys/class/gpio/gpio40	
DO1	/sys/class/gpio/gpio41	
DO2	/sys/class/gpio/gpio42	
DO3	/sys/class/gpio/gpio43	
DI4	/sys/class/gpio/gpio44	
DI5	/sys/class/gpio/gpio45	
DI6	/sys/class/gpio/gpio46	
DI7	/sys/class/gpio/gpio47	
Programmable LED	/sys/class/gpio/gpio964	

## 5.2. Libgpio

libgpiod tool: <https://git.kernel.org/pub/scm/libs/libgpiod/libgpiod.git/tree/README>

API Ref: <https://libgpiod.readthedocs.io/en/latest/files.html>

API example: <https://git.kernel.org/pub/scm/libs/libgpiod/libgpiod.git/tree/examples>

If you want use libgpio API, you should unexport sysfs GPIO.

```
$ sudo /usr/src/advantech/sysfs-gpio/adv-gpio.sh stop
```

If you want use sysfs you can reboot or export sysfs GPIO as follow:

```
$ sudo /usr/src/advantech/sysfs-gpio/adv-gpio.sh start
```

## 6. EAPI(Embedded API) Solution

AdvLinuxTU supports EAPI solution. if you want to know the detail, please read *Embedded API (EAPI) Developer Guide V1.4*.

## 7. Introduction of service

### 7.1. SSH

The default SSH service is enable. You can check status in command:

```
$ systemctl status sshd.service
```

```
advantech@advantech-UNO-2271G-E2xAE:~$ systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-11-19 09:01:57 CST; 6h ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 673 (sshd)
     Tasks: 1 (limit: 4514)
    Memory: 4.7M
   CGroup: /system.slice/ssh.service
           └─673 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

+-- 19 09:01:56 advantech-UNO-2271G-E2xAE systemd[1]: Starting OpenBSD Secure Shell server...
+-- 19 09:01:57 advantech-UNO-2271G-E2xAE sshd[673]: Server listening on 0.0.0.0 port 22.
+-- 19 09:01:57 advantech-UNO-2271G-E2xAE sshd[673]: Server listening on :: port 22.
+-- 19 09:01:57 advantech-UNO-2271G-E2xAE systemd[1]: Started OpenBSD Secure Shell server.
+-- 19 14:58:07 advantech-UNO-2271G-E2xAE sshd[2305]: Accepted password for advantech from 172.16.12.71 port 50564 ssh2
+-- 19 14:58:07 advantech-UNO-2271G-E2xAE sshd[2305]: pam_unix(sshd:session): session opened for user advantech by (uid=0)
```

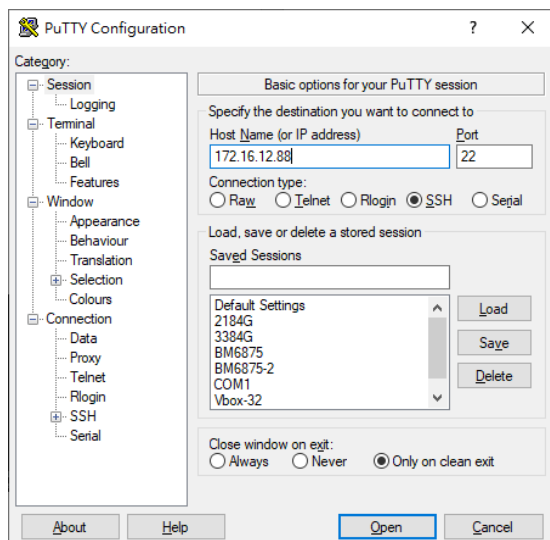
If you want to enable/disable, you can try in commands:

```
$ sudo systemctl enable sshd.service
```

```
$ sudo systemctl disable sshd.service
```

The system will start/stop in next boot.

If service started, you can use ssh client (like putty) to connect the device.



## 7.2. FTP

The default FTP service is enable. You can check status in command:

```
$ systemctl status ssh.service
```

```
oem@ubuntu:~$ systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/systemd/ssh.service; enabled; vendor preset: en>
   Active: active (running) since Mon 2022-12-26 06:40:21 UTC; 2min 52s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 748 (sshd)
     Tasks: 1 (limit: 4370)
    Memory: 4.1M
    CGroup: /system.slice/ssh.service
            └─748 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Dec 26 06:40:21 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
Dec 26 06:40:21 ubuntu sshd[748]: Server listening on 0.0.0.0 port 22.
Dec 26 06:40:21 ubuntu sshd[748]: Server listening on :: port 22.
Dec 26 06:40:21 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
Dec 26 06:42:49 ubuntu sshd[7246]: Accepted password for oem from 172.16.12.55 >
Dec 26 06:42:49 ubuntu sshd[7246]: pam_unix(sshd:session): session opened for u>
```

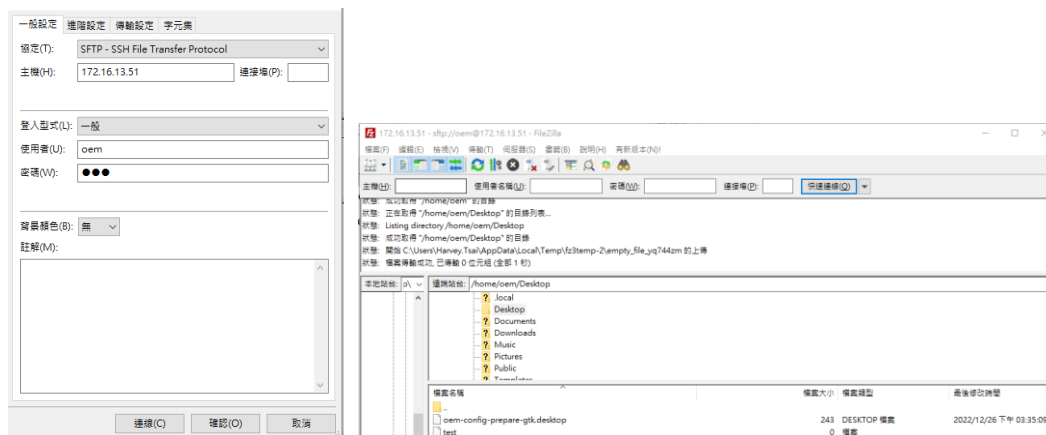
If you want to enable/disable, you can try in commands:

```
$ sudo systemctl enable ssh.service
```

```
$ sudo systemctl disable ssh.service
```

The system will start/stop in next boot.

If service started, you can use ftp client (like FileZilla) to connect the device by sftp.



## 7.3. Watchdog

The default Watchdog service is disable. You can check status in command:

```
$ systemctl status watchdog.service
```

```

root@ubuntu:~# systemctl status watchdog.service
● watchdog.service - watchdog daemon
   Loaded: loaded (/lib/systemd/system/watchdog.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-12-02 06:17:17 UTC; 1min 21s ago
     Process: 1719 ExecStartPre=/bin/sh -c [ -z "$(watchdog_module)" ] || [ "$(watchdog_module)" = "none" ] || /sbin/modprobe $watchdog_module (code=exited, status=0/SUCCESS)
    Main PID: 1728 (watchdog)
       Tasks: 1 (limit: 4418)
      Memory: 588.0K
    CGroup: /system.slice/watchdog.service
            └─1728 /usr/sbin/watchdog

Dec 02 06:17:17 ubuntu watchdog[1728]: interface: no interface to check
Dec 02 06:17:17 ubuntu watchdog[1728]: temperature: no sensors to check
Dec 02 06:17:17 ubuntu watchdog[1728]: no test binary files
Dec 02 06:17:17 ubuntu watchdog[1728]: no repair binary files
Dec 02 06:17:17 ubuntu watchdog[1728]: error retry time-out = 60 seconds
Dec 02 06:17:17 ubuntu watchdog[1728]: repair attempts = 1
Dec 02 06:17:17 ubuntu watchdog[1728]: alive=/dev/watchdog heartbeat=[none] to=root no_act=no force=no
Dec 02 06:17:17 ubuntu watchdog[1728]: watchdog now set to 60 seconds
Dec 02 06:17:17 ubuntu watchdog[1728]: hardware watchdog identity: wdat_wdt
Dec 02 06:17:17 ubuntu systemd[1]: Started watchdog daemon.

```

You can enable/disable it in command:

```
$ sudo systemctl enable watchdog.service
```

```
$ sudo systemctl disable watchdog.service
```

In “/etc/watchdog.conf”, you can add “watchdog-timeout” for specific timeout and more detail from [WebSite](#).

If you want to test if the hardware watchdog is working, you can try in command: (make sure that the status of watchdog service is disable)

```
$ sudo watchdog -v -F
```

The system will reboot after kill watchdog (Ctrl+C).

Or make kernel crash in command: (make sure that the status of watchdog service is enable)

```
$ echo c | sudo tee /proc/sysrq-trigger
```

The command will make system hang and hardware watchdog will reboot system after it's timeout.

If your image is not AdvLinuxTU, you should insert module yourself.

You should insert different watchdog driver according to different device as flow:

```
$ sudo modprobe <driver name>
```

**Superio driver(w83627hf\_wdt):**

UNO-127, UNO-148, UNO-247, UNO-2484G\_V2, UNO-348, UNO-348\_H, UNO-348\_W,  
PPC-3??-7706, TPC-B610?-A\*

**Intel watchdog driver(wdat\_wdt):**

UNO-137\_V2, UNO-2271G\_V2, UNO-2372G\_V2, AMAX-5570S

**Mio driver(watchdog-ahc0):**

UNO-238, UNO-238\_V2

**Ec driver(adv\_wdt\_drv):**

AMAX-5570, AMAX-5580, UNO-1372G, UNO-137, UNO-2372G, UNO-2473G, UNO-2484G, UNO-410, UNO-420, UNO-430

## 7.4. WISE-Agent

The Image is pre-installed WISE-Agent which is a service of Advantech to connect WISE-DeviceOn. The default WISE-Agent service is disable.

You can enable/disable it in command:

```
$ sudo systemctl enable saagent.service
$ sudo systemctl disable saagent.service
```

You can get more detail from `/usr/src/advantech/wise-agent/bin/Install` manual and [WebSite](#).

## 7.5. Firewall

The default Firewall of the image is active and only allow ssh. You can check status in command:

```
$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
```

You can enable or disable

```
$ sudo ufw enable
Firewall is active and enabled on system startup
$ sudo ufw disable
Firewall stoped and disabled on system startup
```

You can add protocol or port connection to allow list

```
$ sudo ufw allow ssh
```

or

```
$ sudo ufw allow 22
```

you can delete protocol or port connection from allow list

```
$ sudo ufw delete allow ssh
```

or

```
$ sudo ufw delete allow 22
```

you can reset firewall

```
$ sudo ufw reset
```

After reset, default is deny all connection when firewall enable.

You can allow all connection

```
$ sudo ufw default allow
```

You can deny all connection

```
$ sudo ufw default deny
```



## 8. Full disk encryption

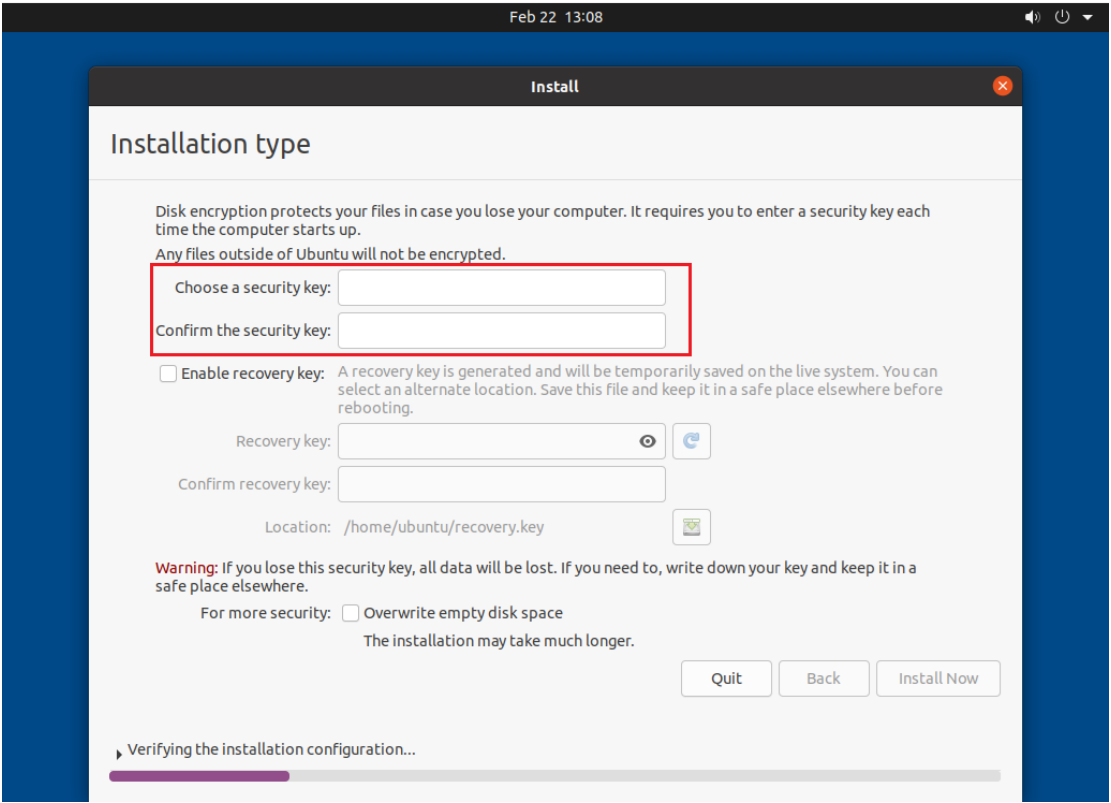
AdvLinuxTU supports enabling disk encryption during installation. Disk encryption protects the data on a block device by encrypting it. To access the device's decrypted contents, user need to provide a passphrase or key as authentication. AdvLinuxTU integrate the key with TPM, so you do not need to input the passphrase during running. The function of Disk encryption is like the bitlocker function of Windows Embedded.

### 8.1. Install

You need to choose [Advantech Install Ubuntu Desktop (Disk Encryption)] in grub menu.

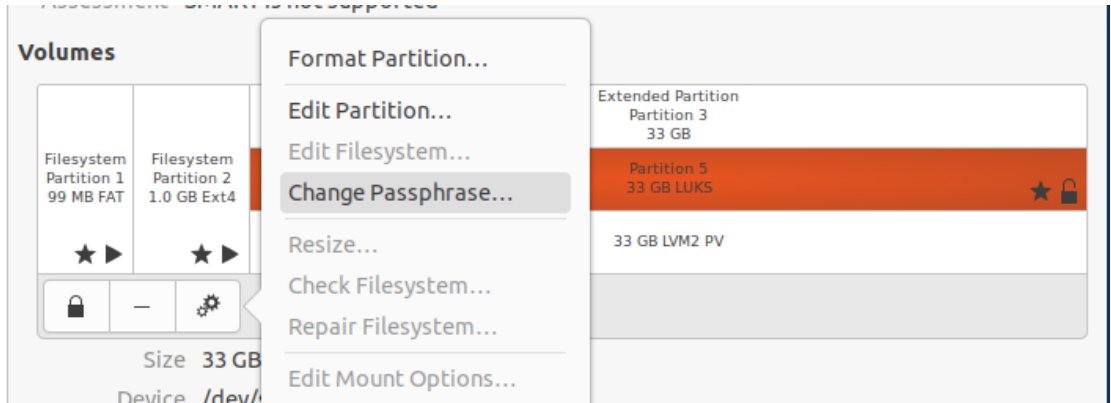


You need to input a security key and remember it. The security key can unlock disk encryption when you boot device if you change device, update BIOS or clear TPM. We also create a random key in TPM to boot without any input.



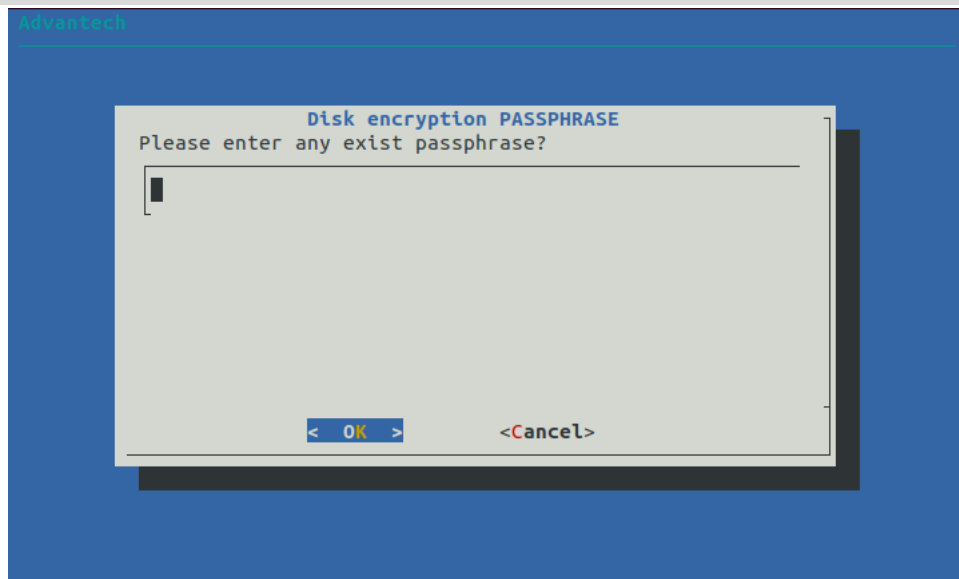
## 8.2. Change keys

You can change the security key after boot. [Show Applications → Disks]

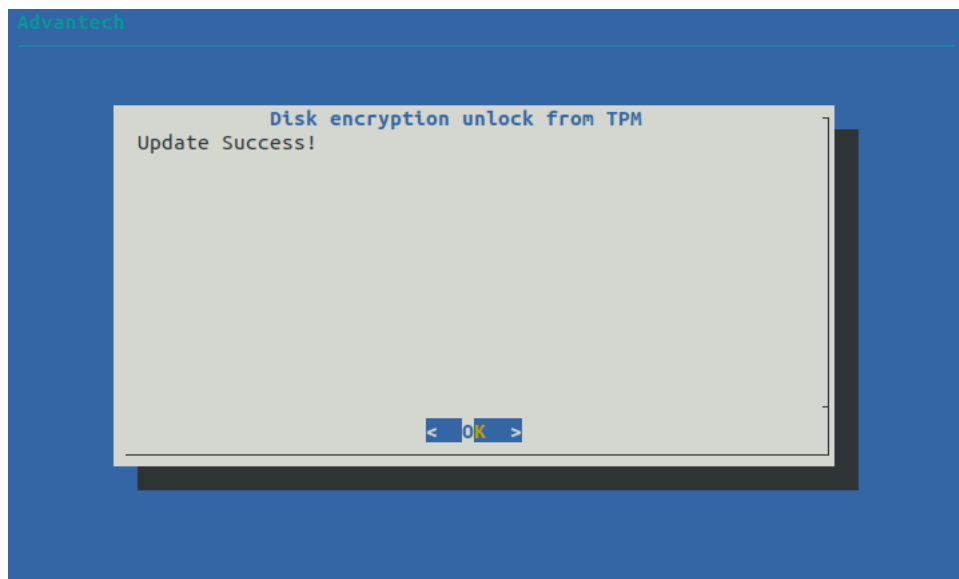


You can change the TPM key for disk encryption.

```
$ sudo atcc.fde -tpm-update
```



You need input the security key created in the image installation for create a new random key in TPM.



## 9. Read-Only system setting

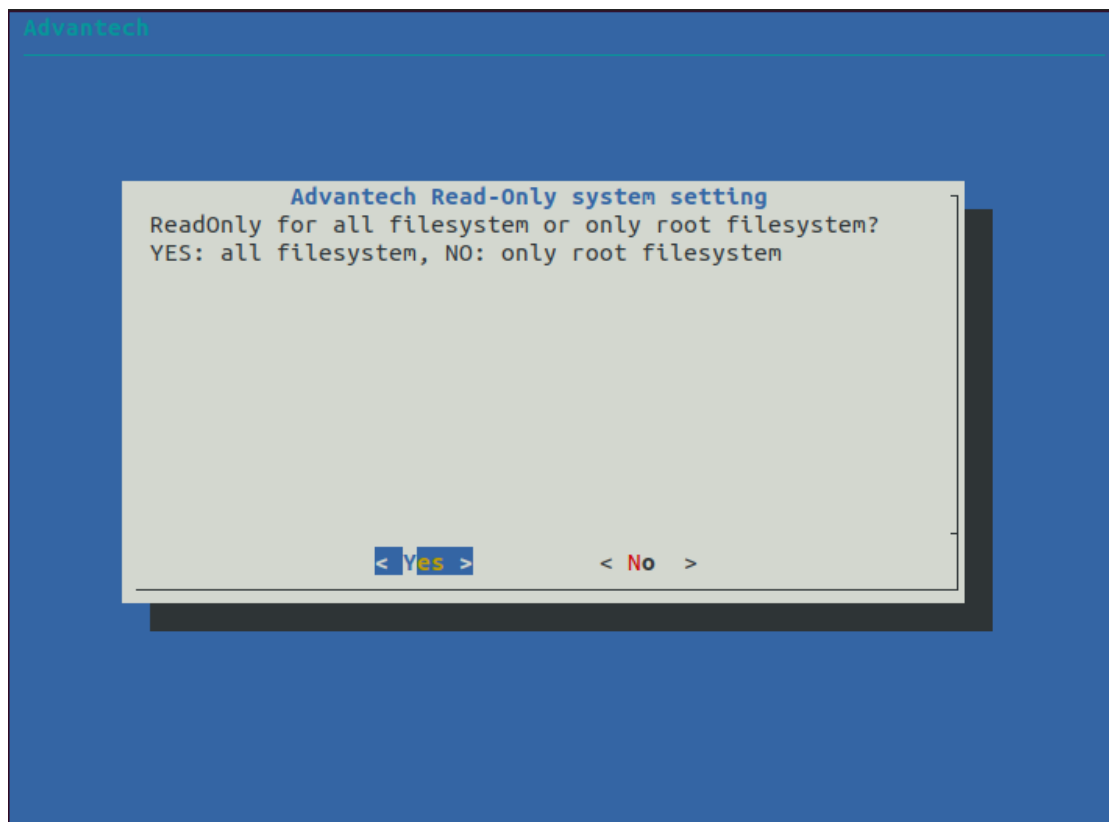
AdvLinuxTU supports read-only system setting and any changes will be lost after reboot. It has two modes for read-only system setting. User can choose read-only setting in all file system or root file system only.

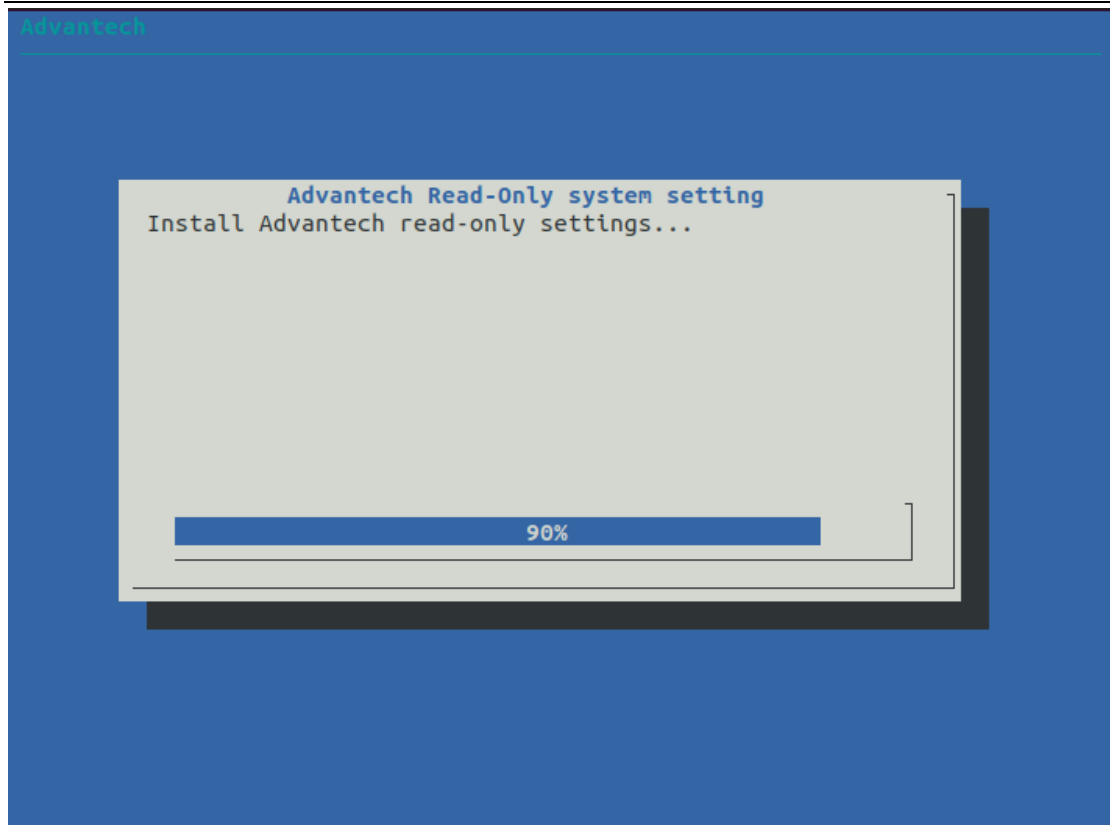
### 9.1. Install

Read-only system setting is not install in default and you need to install if you want.

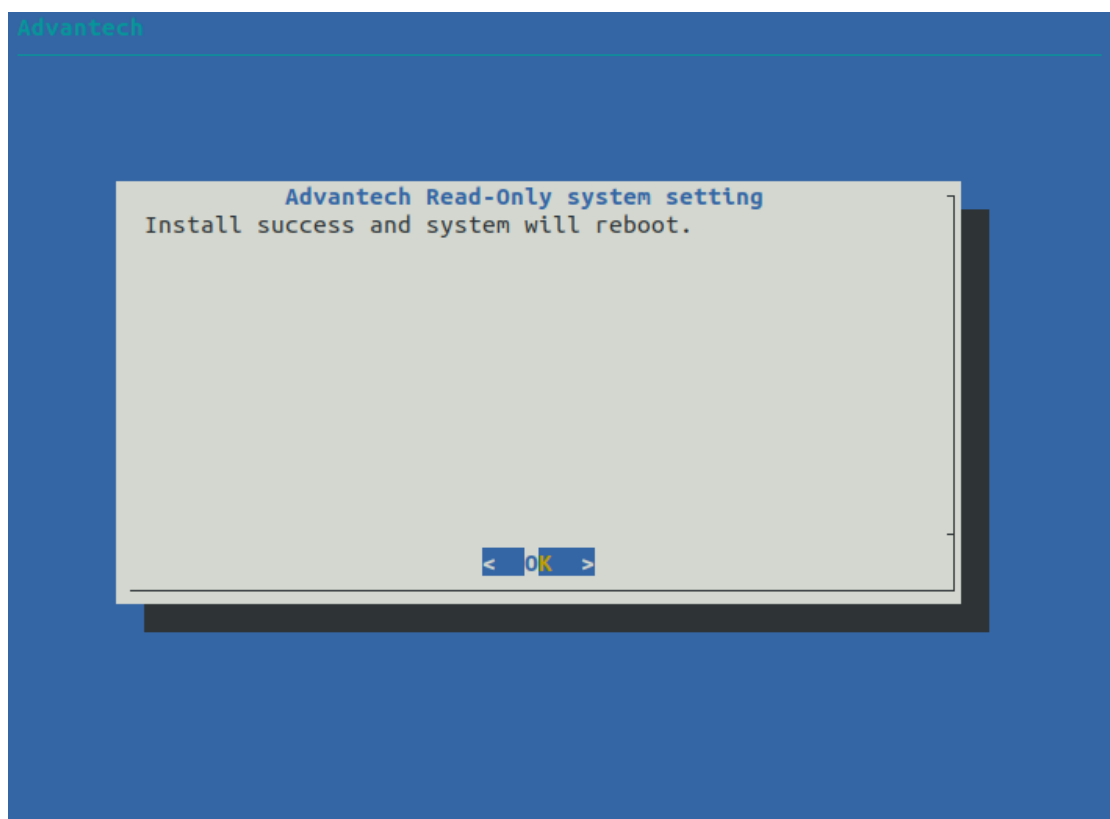
```
$ sudo atcc.rofs -install
```

You can choose read-only in all file system or root file system only. You can read-write or do changes in non-root file system if you choose read-only in root file system only.





After install, the system will reboot and it boot in read-only system by default.



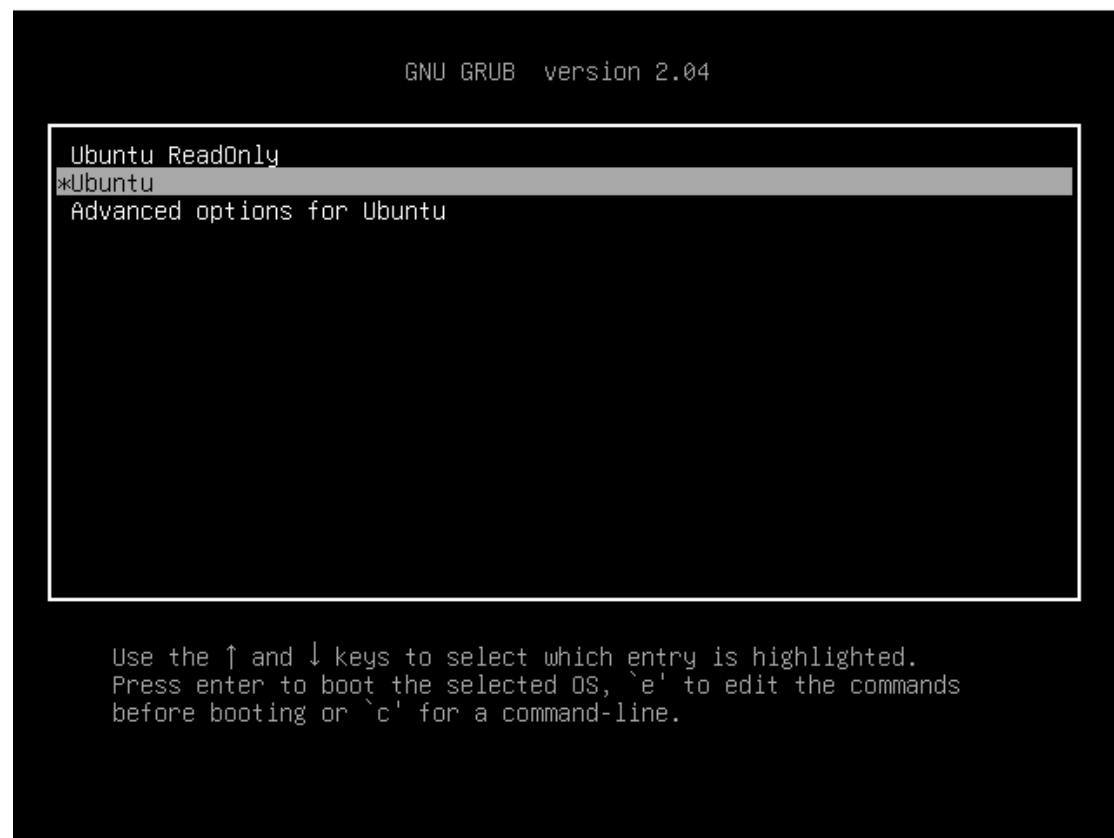
you can check setting status.

```
$ atcc.rofs -show
```

```
advantech@advantech-device:~$ atcc.rofs -show
at-uwf:      Installed
Current:     ReadOnly enable in root filesystem only
```

## 9.2. Changes for read-only setting

You want to do changes in read-only file system. you can reboot device and choose “Ubuntu” in GRUB menu.

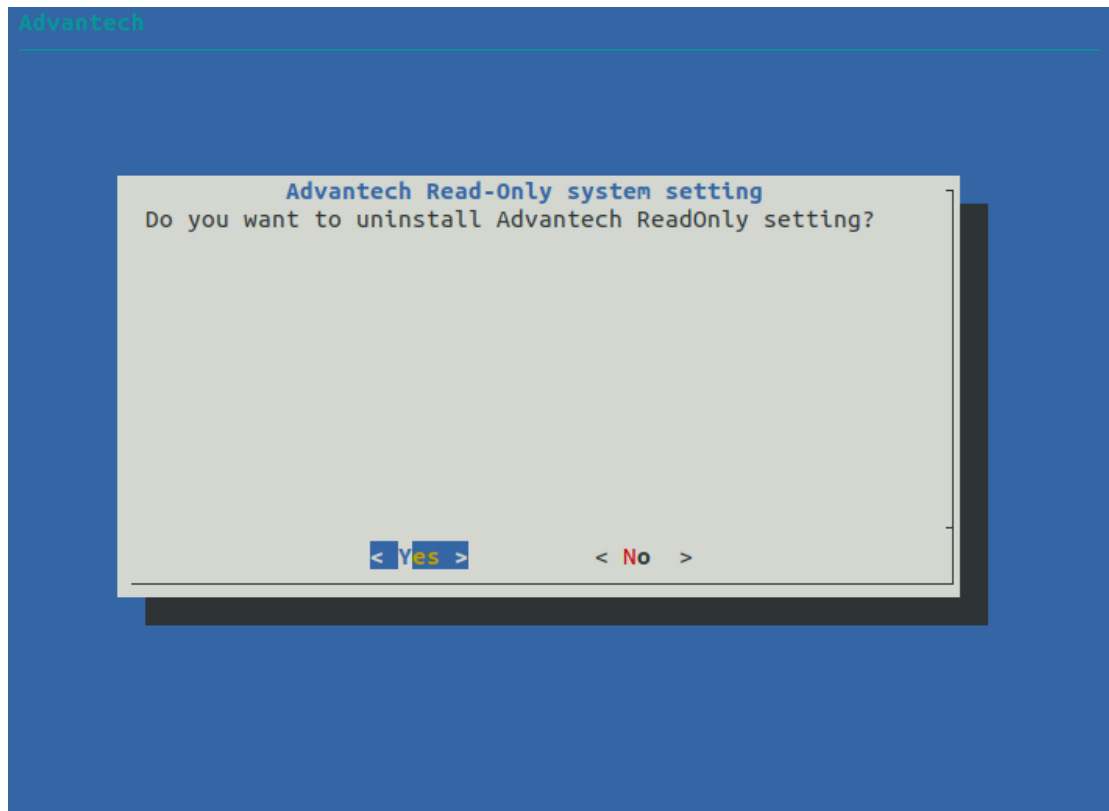


***You must reboot and back into read-only system after changes be done.***

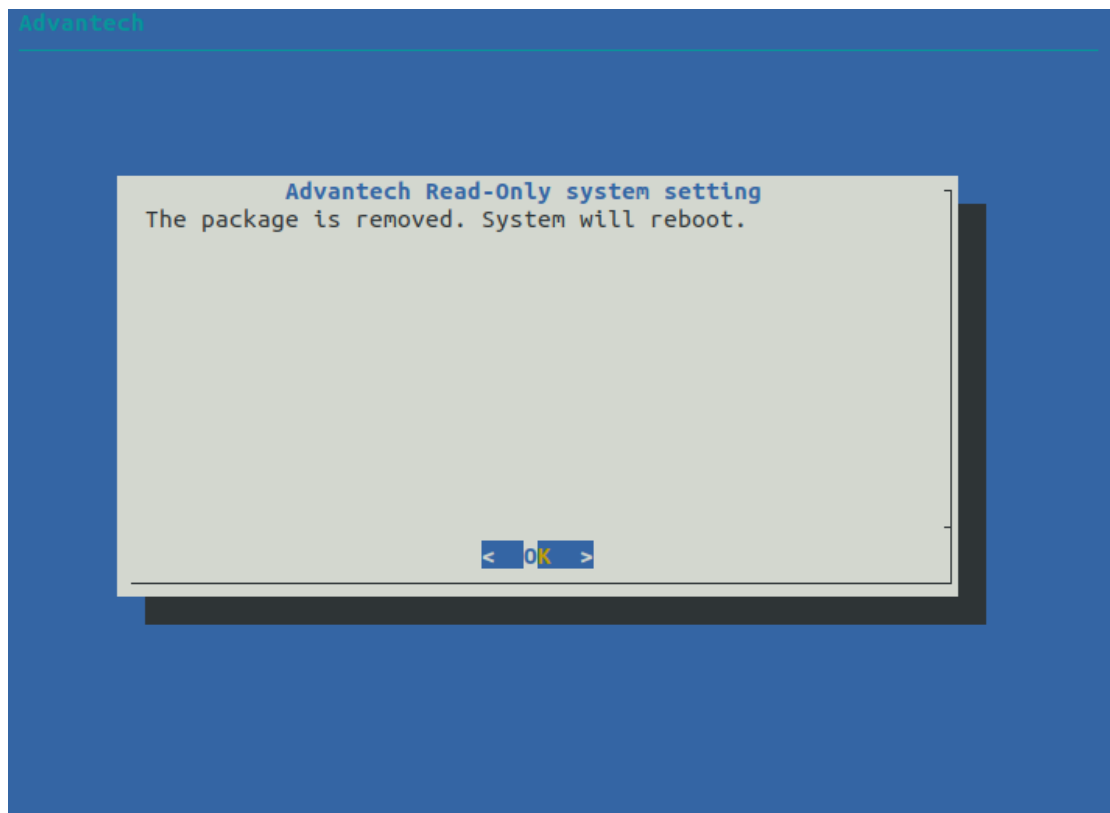
## 9.3. Uninstall

You want to uninstall read-only system setting. you must reboot and choose “Ubuntu” in GRUB menu.

```
$ sudo atcc.rofs -uninstall
```



After uninstall, the system will reboot.





## 10. Backup & Restore

AdvLinuxTU supports disk backup and restore. You can use USB disk to backup and use the USB disk to restore. The USB disk size should be the same or more then the disk size to make sure everything is ok.

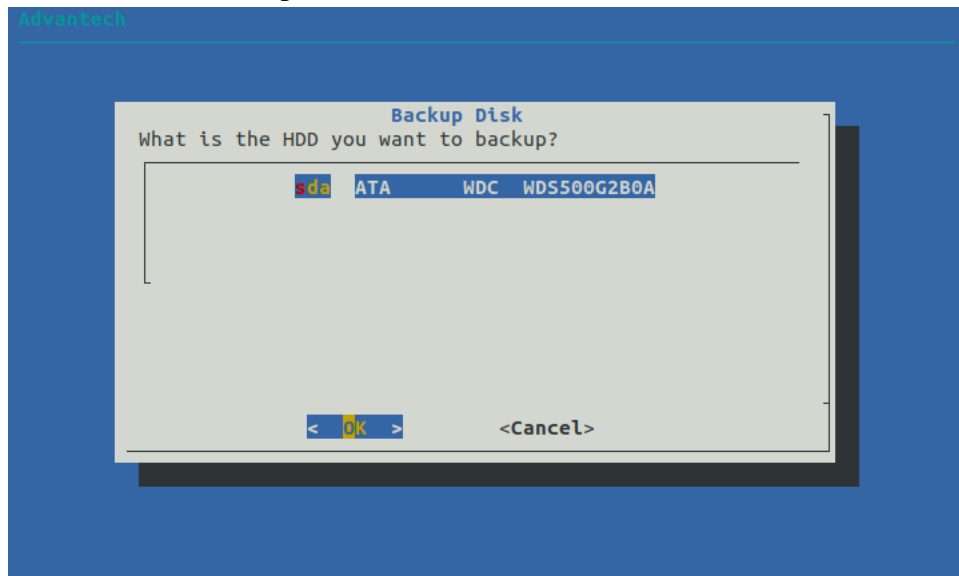
AdvLinuxTu backup and restore process cannot support secure boot enable in BIOS.

### 10.1. Backup

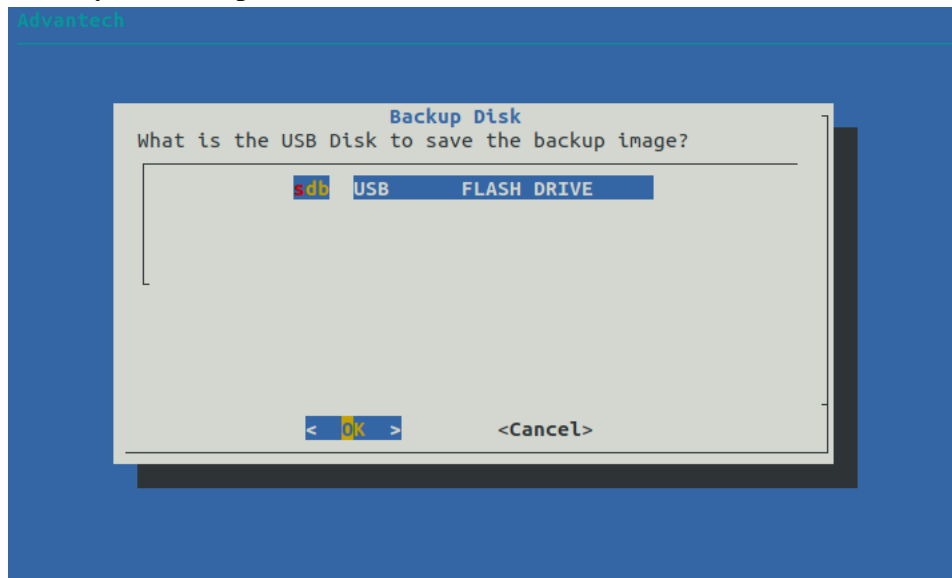
You can use a USB disk to backup.

```
$ sudo atcc.backup
```

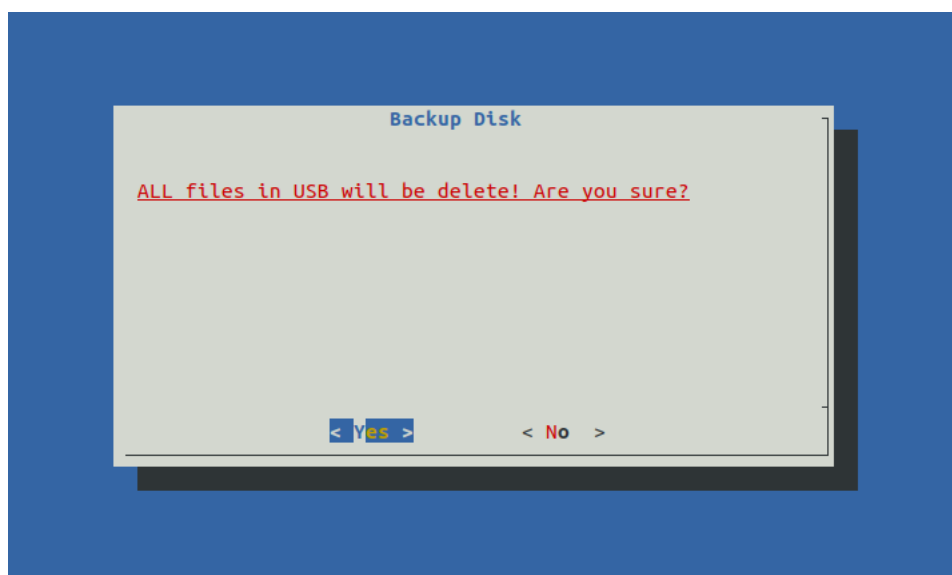
Select HDD to backup



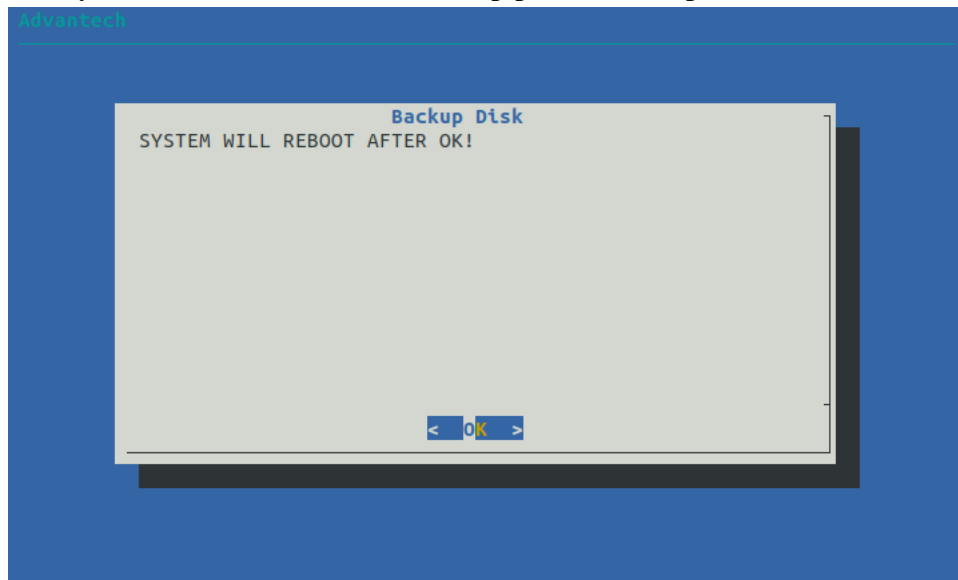
Select your backup USB disk.



Make sure to delete files in USB disk.



The system will reboot and run backup process after press [OK].

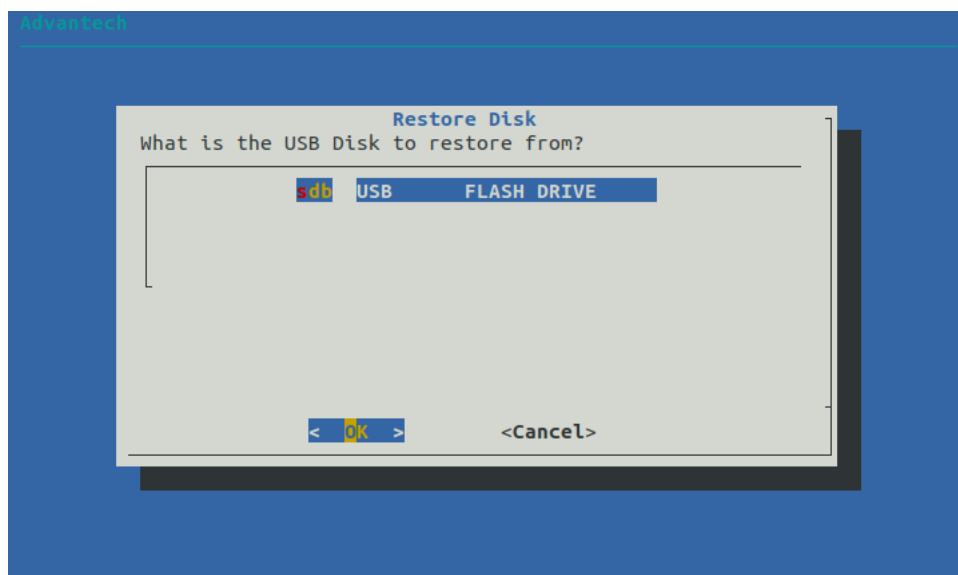


## 10.2. Restore

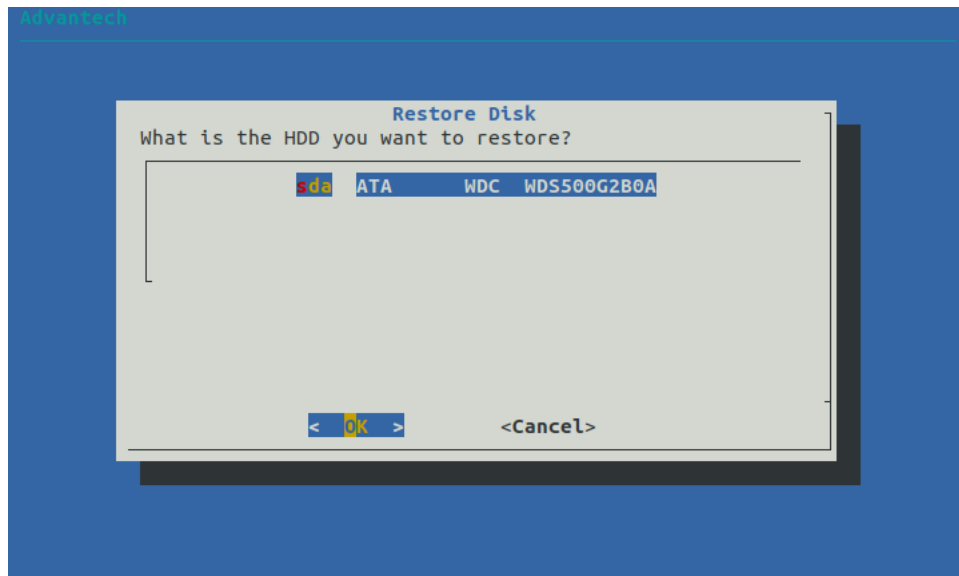
If AdvLinuxTu can boot into system, you can

```
$ sudo atcc.restore
```

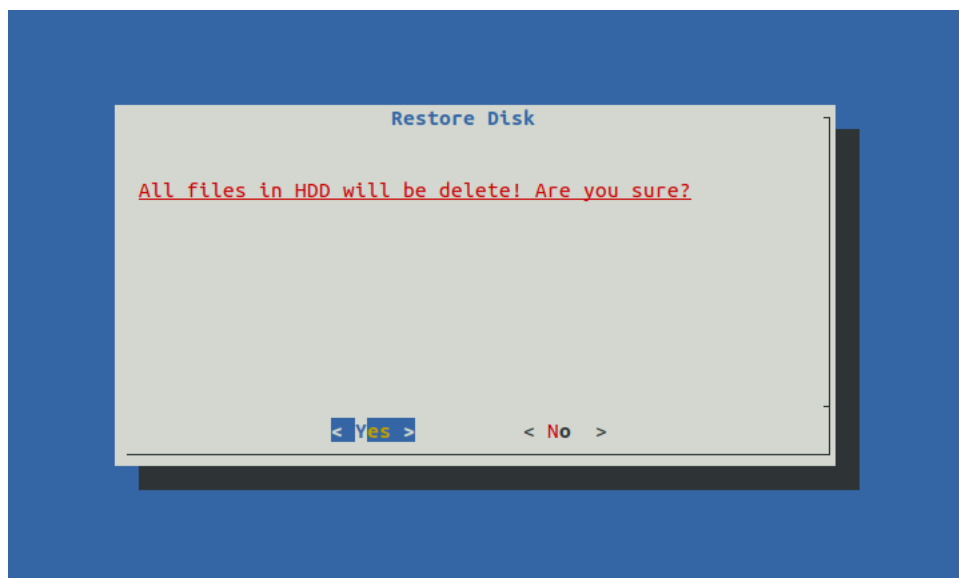
Select restore USB disk.



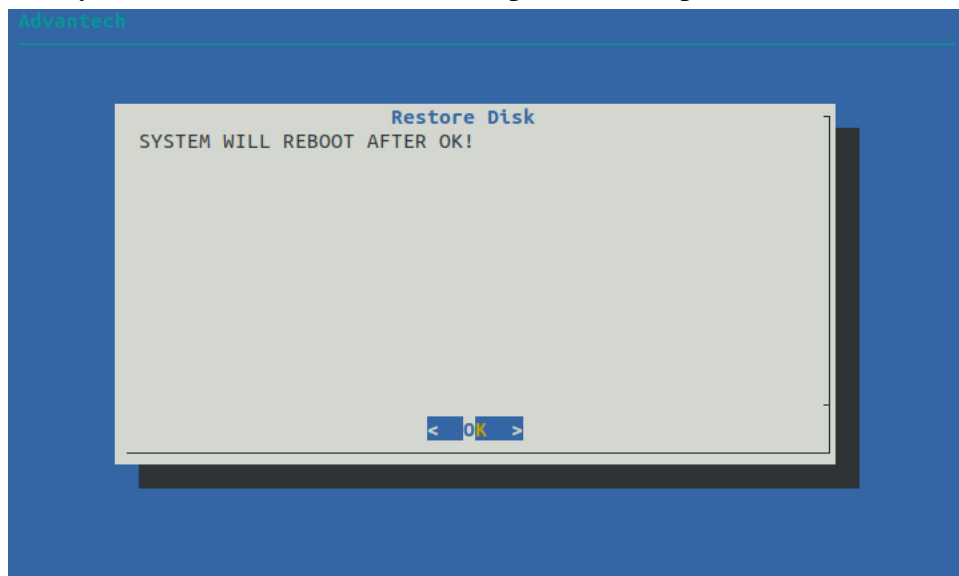
Select HDD to restore.



Make sure to delete all files in HDD



The system will reboot and run restore process after press [OK].



If you cannot boot AdvLinuxTu, you can boot from the restore USB via BIOS to run restore process.

