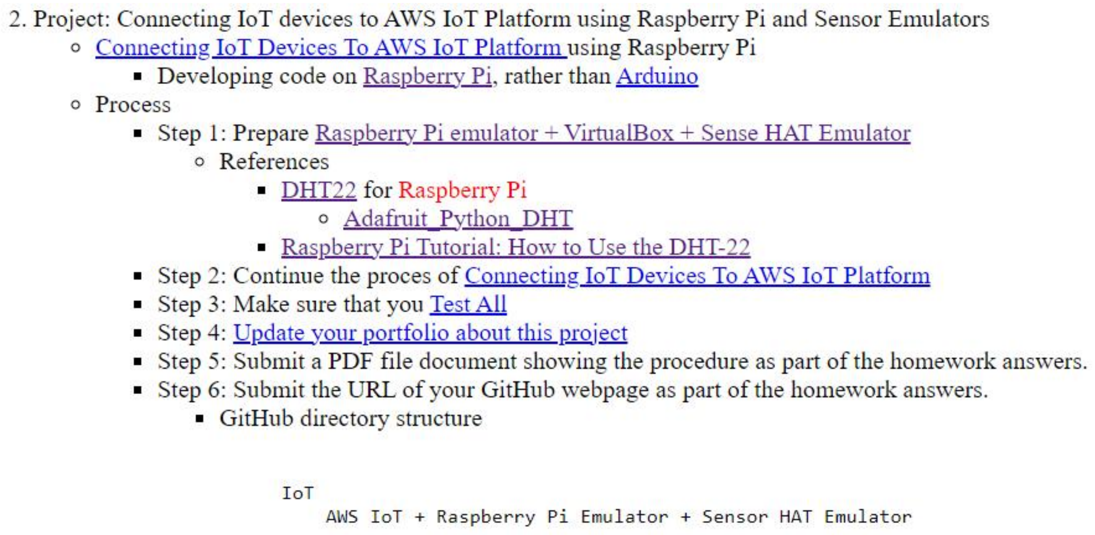
Week 10: Homework 1: Raspberry Pi emulator + VirtualBox + Sense HAT Emulator

Exercises for Learning AWS IoT (sfbu.edu)

Project: Connecting IoT devices to AWS IoT Platform using Raspberry Pi and Sensor Emulators



2. Project: Connecting IoT devices to AWS IoT Platform using Raspberry Pi and Sensor Emulators
   - Connecting IoT Devices To AWS IoT Platform using Raspberry Pi
     - Developing code on Raspberry Pi, rather than Arduino
   - Process
     - Step 1: Prepare Raspberry Pi emulator + VirtualBox + Sense HAT Emulator
       - References
         - DHT22 for Raspberry Pi
           - Adafruit_Python_DHT
         - Raspberry Pi Tutorial: How to Use the DHT-22
     - Step 2: Continue the proces of Connecting IoT Devices To AWS IoT Platform
     - Step 3: Make sure that you Test All
     - Step 4: Update your portfolio about this project
     - Step 5: Submit a PDF file document showing the procedure as part of the homework answers.
     - Step 6: Submit the URL of your GitHub webpage as part of the homework answers.
       - GitHub directory structure

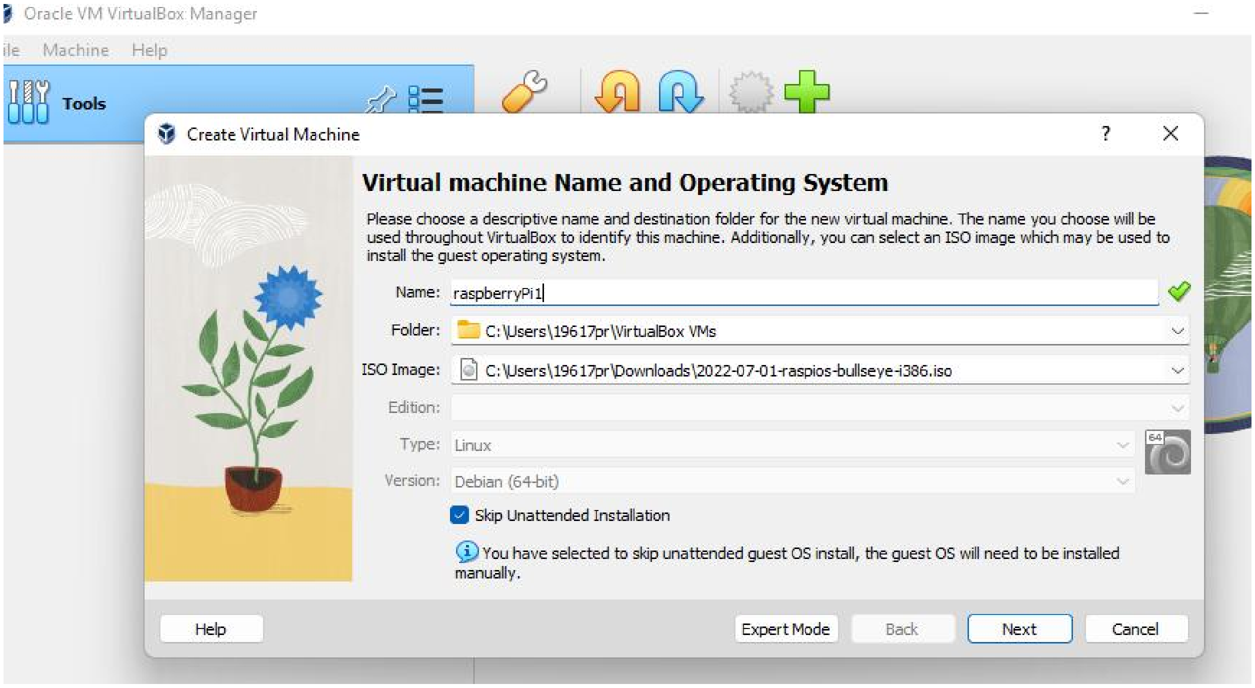
```
IoT
    AWS IoT + Raspberry Pi Emulator + Sensor HAT Emulator
```

Reference:

Running Raspberry Pi Desktop within VirtualBox - Pi My Life Up

Step 1. Prepare Raspberry Pi enulator + VirtualBox + Sense HAT Emulator

Step 1-1. install Oracle VM VirtualBox



Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: raspberryPi1

Folder: C:\Users\19617pr\VirtualBox VMs

ISO Image: C:\Users\19617pr\Downloads\2022-07-01-raspios-bullseye-i386.iso

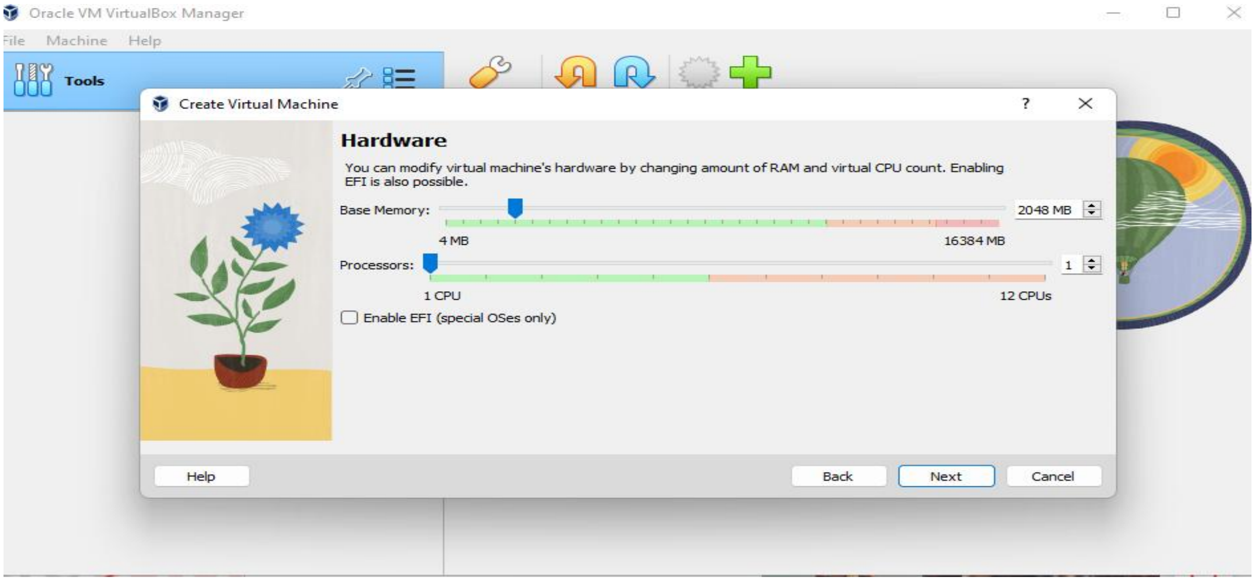
Edition:

Type: Linux

Version: Debian (64-bit)

☑ Skip Unattended Installation

You have selected to skip unattended guest OS install, the guest OS will need to be installed manually.



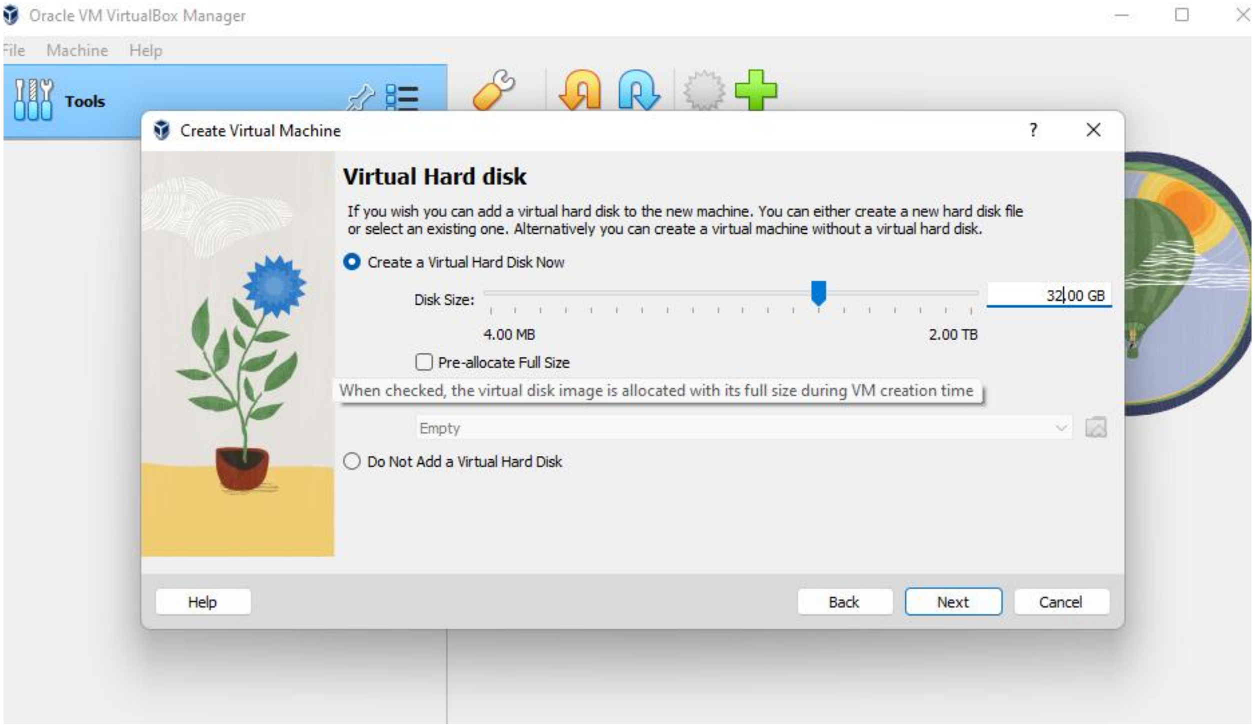
Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory: 2048 MB (4 MB – 16384 MB)

Processors: 1 (1 CPU – 12 CPUs)

☐ Enable EFI (special OSes only)



Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

● Create a Virtual Hard Disk Now

Disk Size: 32.00 GB

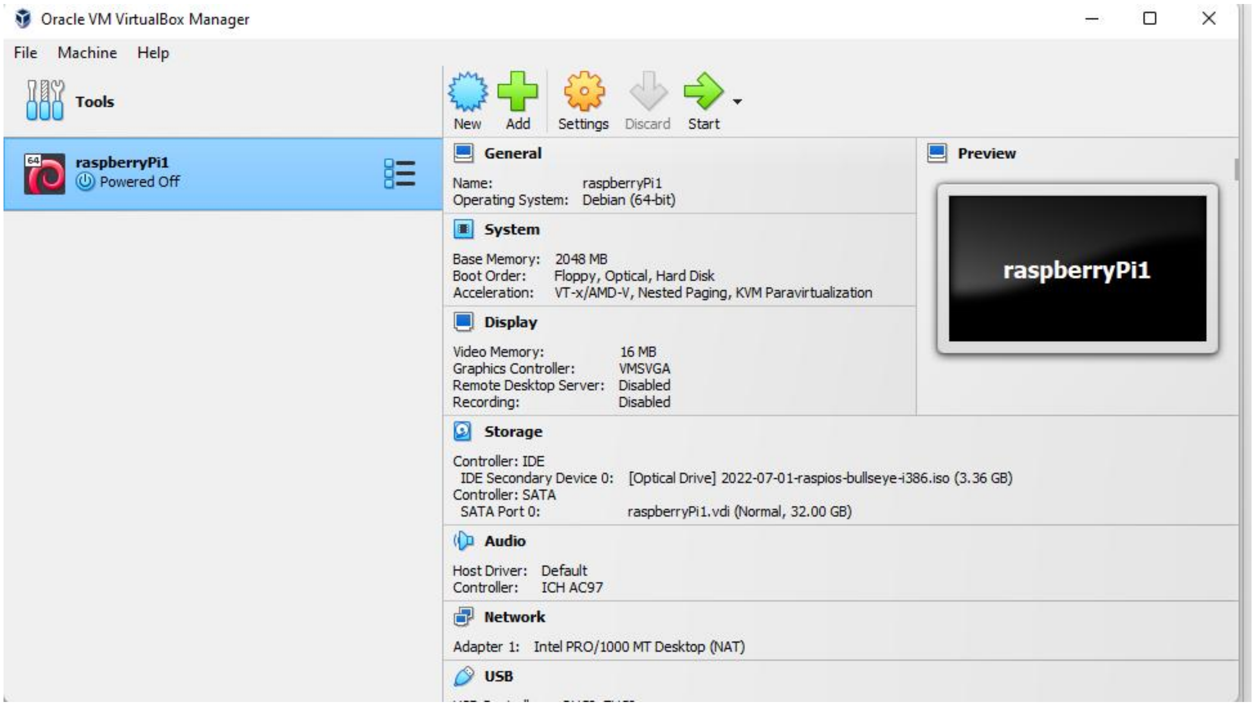
4.00 MB 2.00 TB

□ Pre-allocate Full Size

When checked, the virtual disk image is allocated with its full size during VM creation time

Empty

○ Do Not Add a Virtual Hard Disk



raspberryPi1
Powered Off

General
Name: raspberryPi1
Operating System: Debian (64-bit)

System
Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] 2022-07-01-raspios-bullseye-i386.iso (3.36 GB)
Controller: SATA
SATA Port 0: raspberryPi1.vdi (Normal, 32.00 GB)

Audio
Host Driver: Default
Controller: ICH AC97

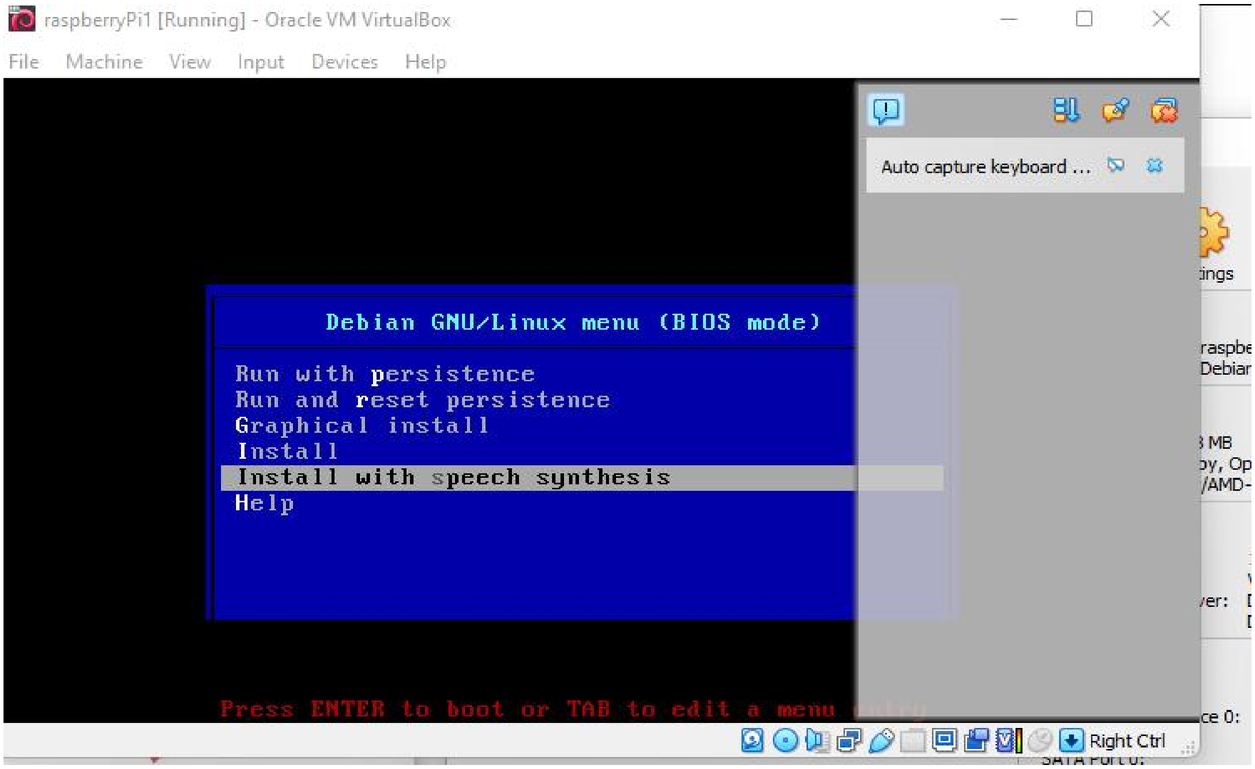
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

Step 1-2: Starting up Raspberry Pi VirtualBox Machine

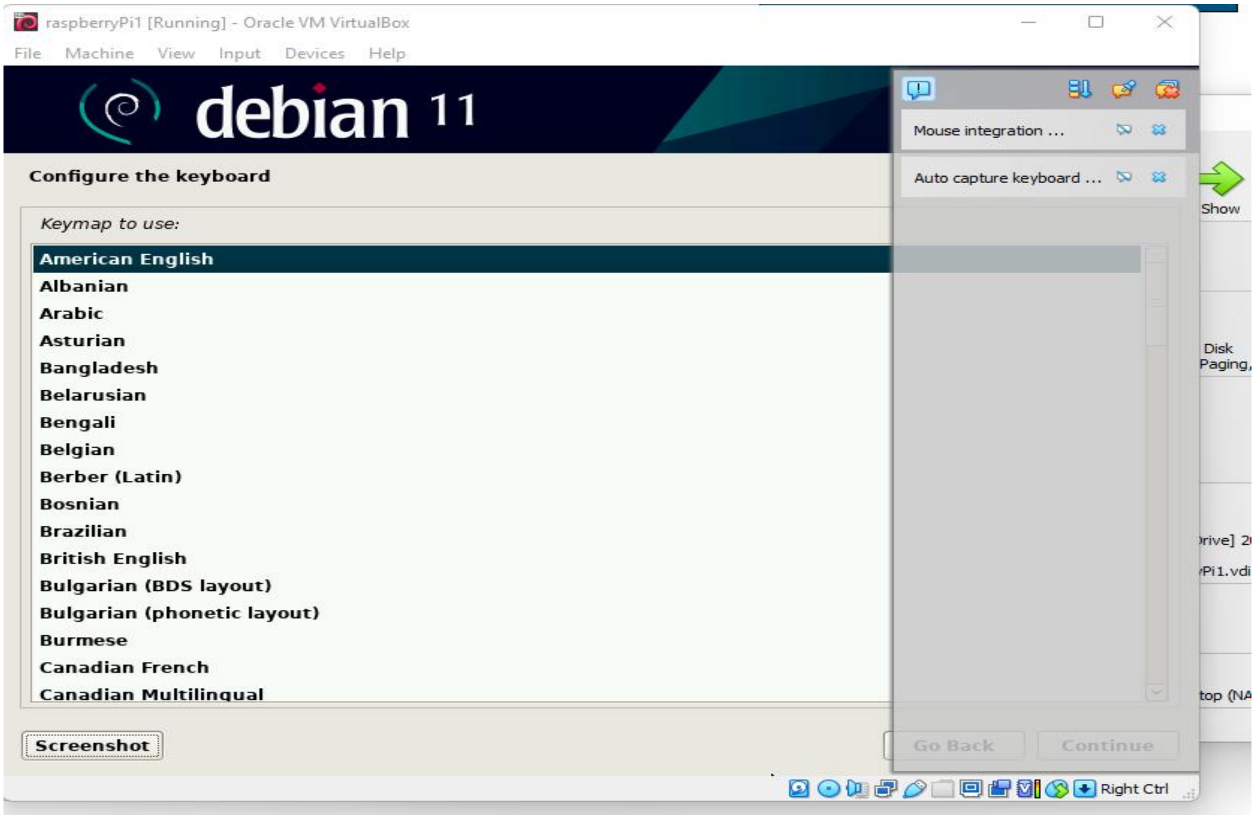
Installing Raspberry Pi Desktop to VirtualBox:

When using these menus, you need to use the ARROW keys to navigate the menus. Use the SPACE key to select an option and the ENTER key to confirm the selection.



raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Auto capture keyboard ...
Right Ctrl

```
Debian GNU/Linux menu (BIOS mode)

Run with persistence
Run and reset persistence
Graphical install
Install
Install with speech synthesis
Help

Press ENTER to boot or TAB to edit a menu entry
```

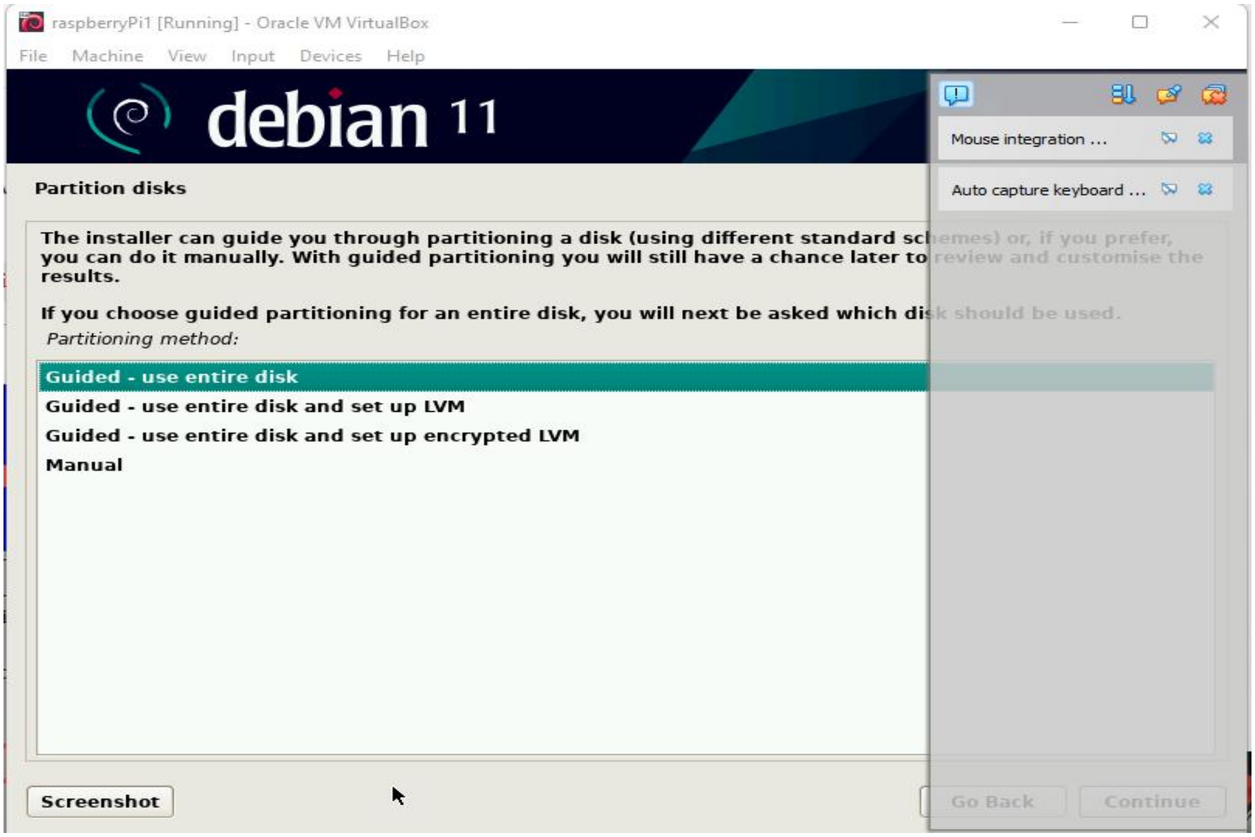


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Mouse integration ...
Auto capture keyboard ...
Screenshot
Go Back
Continue
Right Ctrl

Configure the keyboard

Keymap to use:

American English
Albanian
Arabic
Asturian
Bangladesh
Belarusian
Bengali
Belgian
Berber (Latin)
Bosnian
Brazilian
British English
Bulgarian (BDS layout)
Bulgarian (phonetic layout)
Burmese
Canadian French
Canadian Multilingual


raspberryPi1 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
debian 11
Mouse integration ...
Auto capture keyboard ...

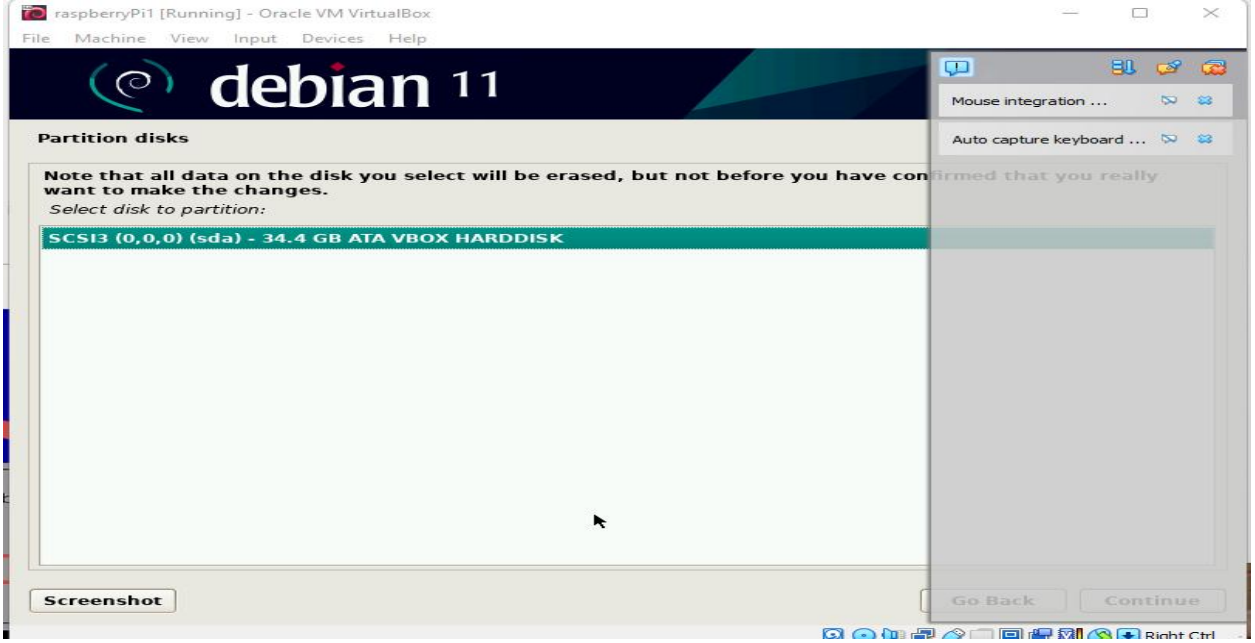
Partition disks

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

- Guided - use entire disk
- Guided - use entire disk and set up LVM
- Guided - use entire disk and set up encrypted LVM
- Manual

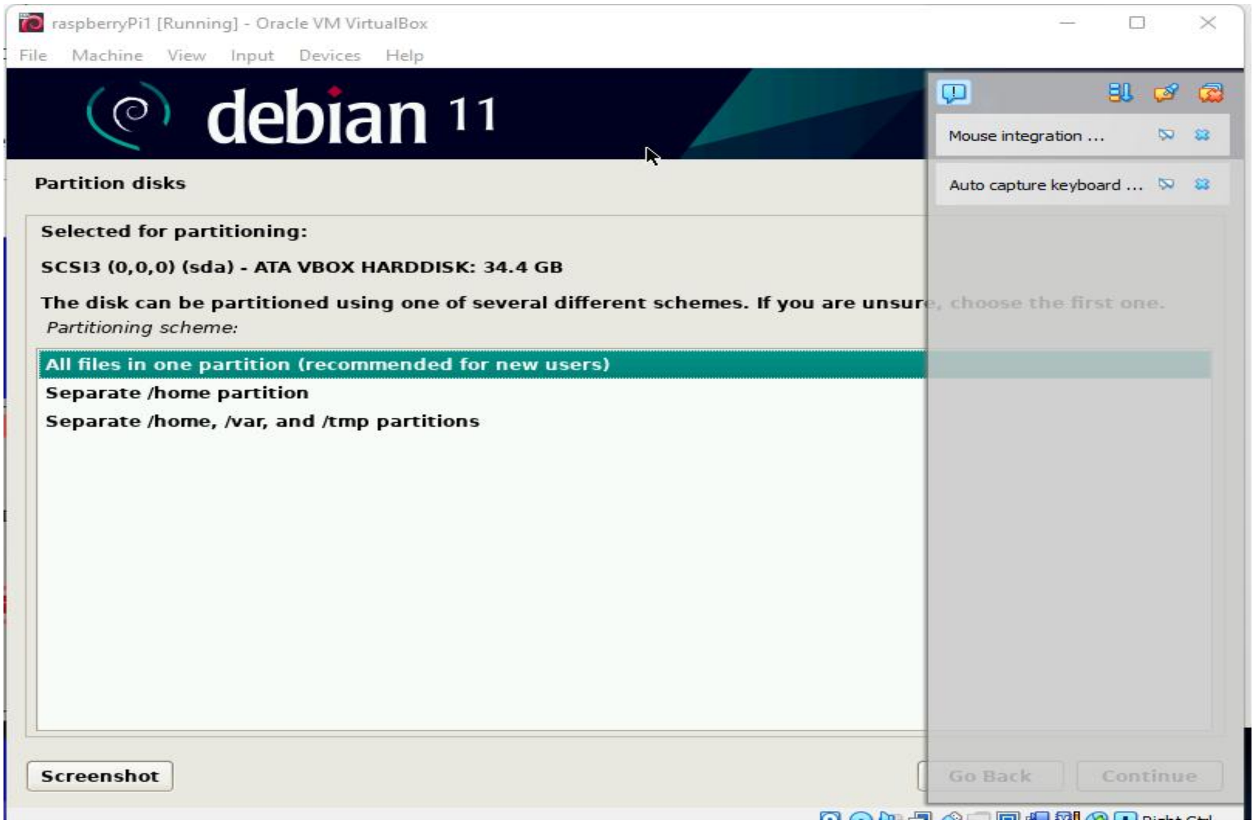

raspberryPi1 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
debian 11
Mouse integration ...
Auto capture keyboard ...

Partition disks

Note that all data on the disk you select will be erased, but not before you have confirmed that you really want to make the changes.

Select disk to partition:

- SCSI3 (0,0,0) (sda) - 34.4 GB ATA VBOX HARDDISK


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Mouse integration ...
Auto capture keyboard ...

Partition disks

Selected for partitioning:

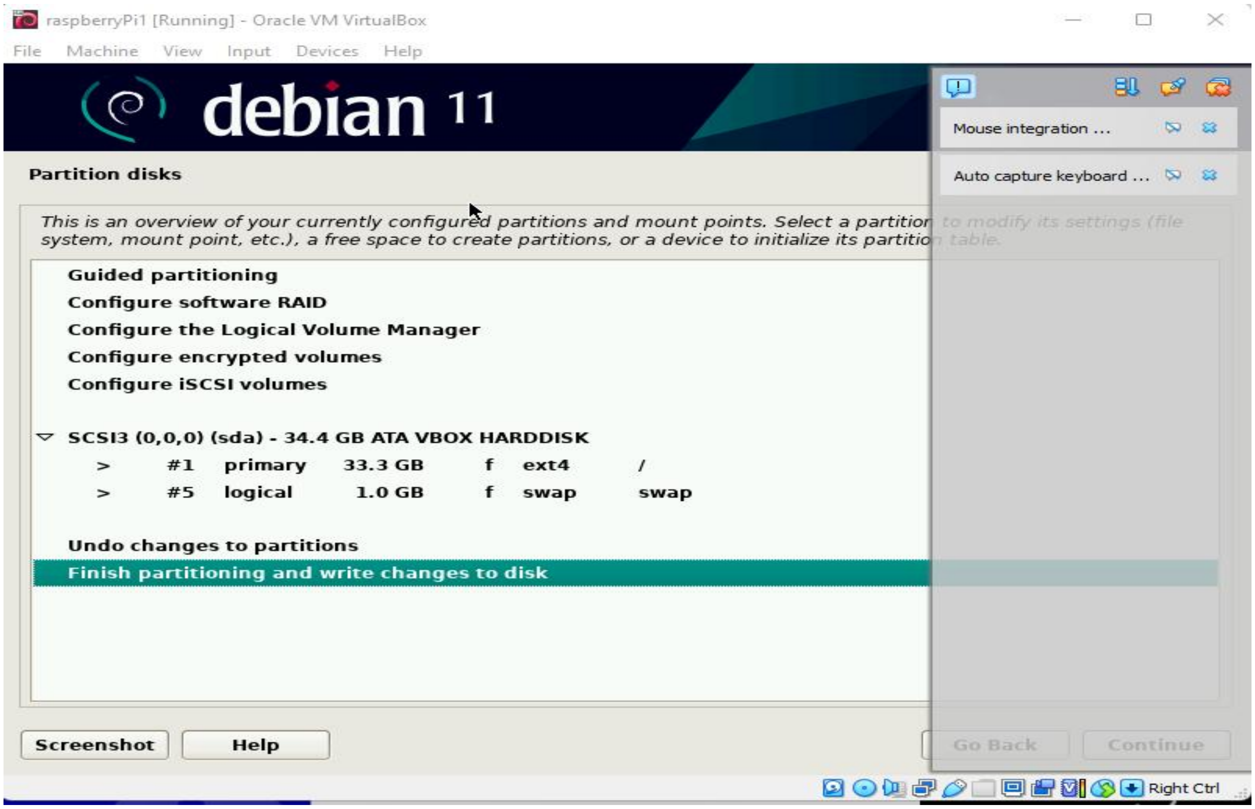
SCSI3 (0,0,0) (sda) - ATA VBOX HARDDISK: 34.4 GB

The disk can be partitioned using one of several different schemes. If you are unsure, choose the first one.

Partitioning scheme:

All files in one partition (recommended for new users)
Separate /home partition
Separate /home, /var, and /tmp partitions

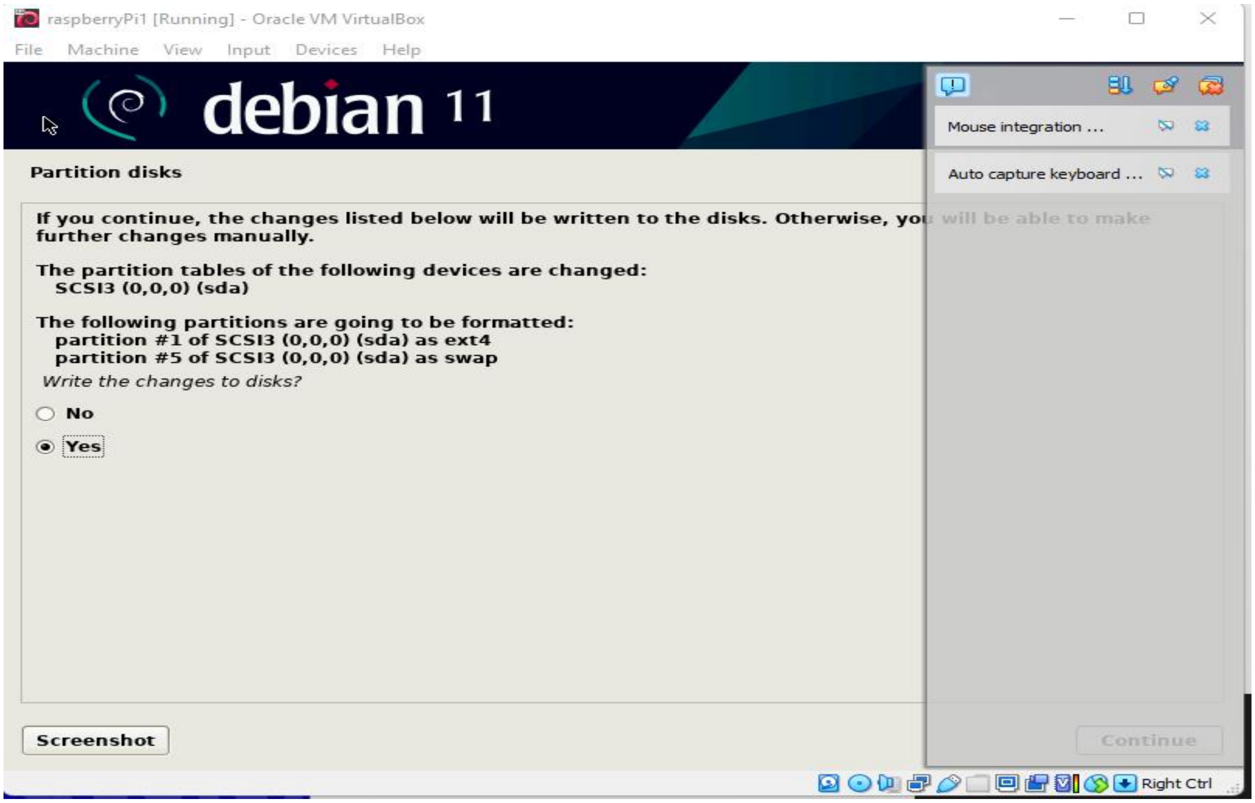
Screenshot
Go Back
Continue


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Mouse integration ...
Auto capture keyboard ...

Partition disks

This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.

```
    Guided partitioning
    Configure software RAID
    Configure the Logical Volume Manager
    Configure encrypted volumes
    Configure iSCSI volumes

▽   SCSI3 (0,0,0) (sda) - 34.4 GB ATA VBOX HARDDISK
        >     #1   primary   33.3 GB    f   ext4      /
        >     #5   logical    1.0 GB    f   swap      swap

    Undo changes to partitions
    Finish partitioning and write changes to disk
```


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Mouse integration ...
Auto capture keyboard ...
Screenshot
Continue
Right Ctrl

Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

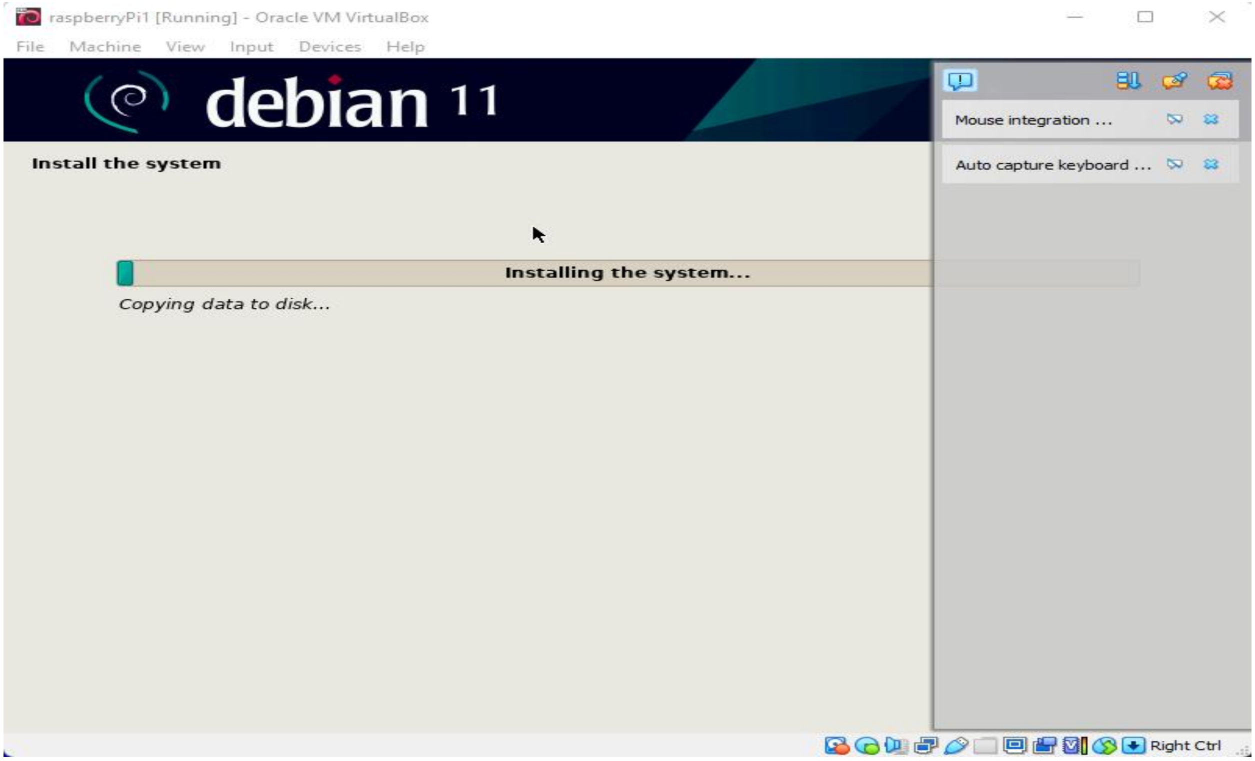
The partition tables of the following devices are changed:
SCSI3 (0,0,0) (sda)

The following partitions are going to be formatted:
partition #1 of SCSI3 (0,0,0) (sda) as ext4
partition #5 of SCSI3 (0,0,0) (sda) as swap

Write the changes to disks?

○ No

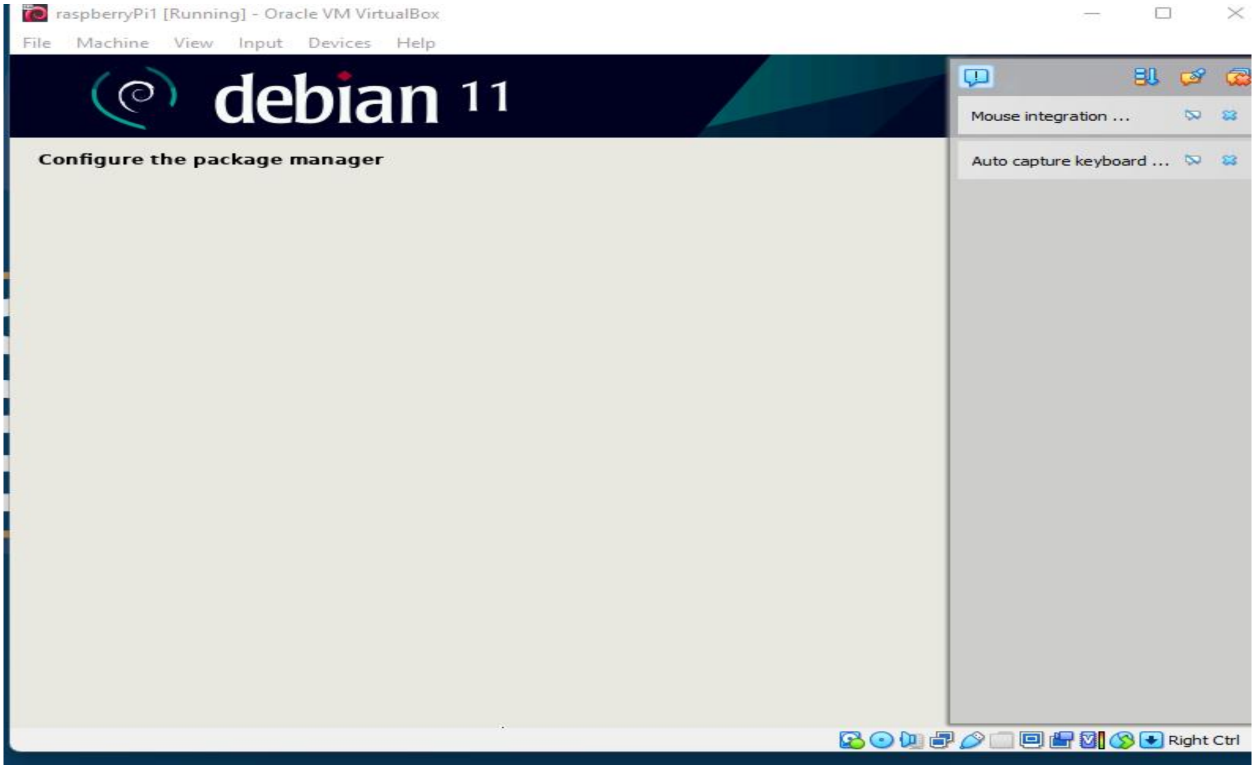
● Yes


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Mouse integration ...
Auto capture keyboard ...
Right Ctrl

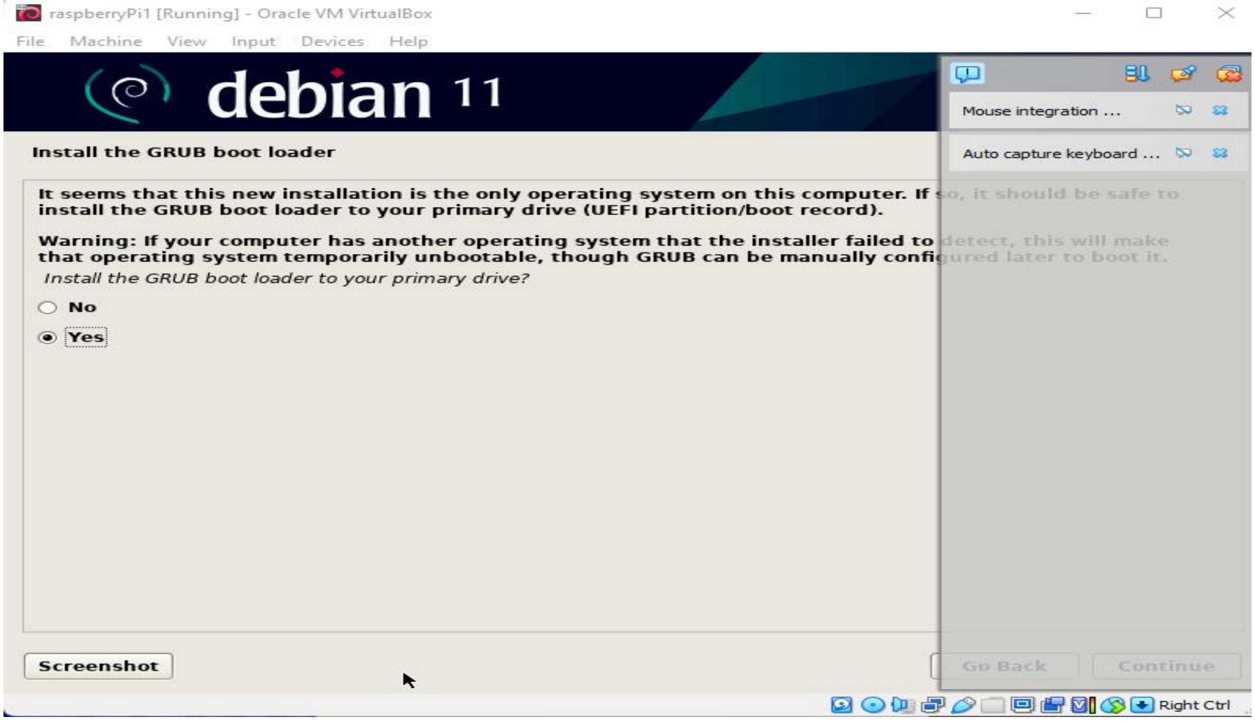
Install the system

Installing the system...

Copying data to disk...



Configure the package manager



Install the GRUB boot loader

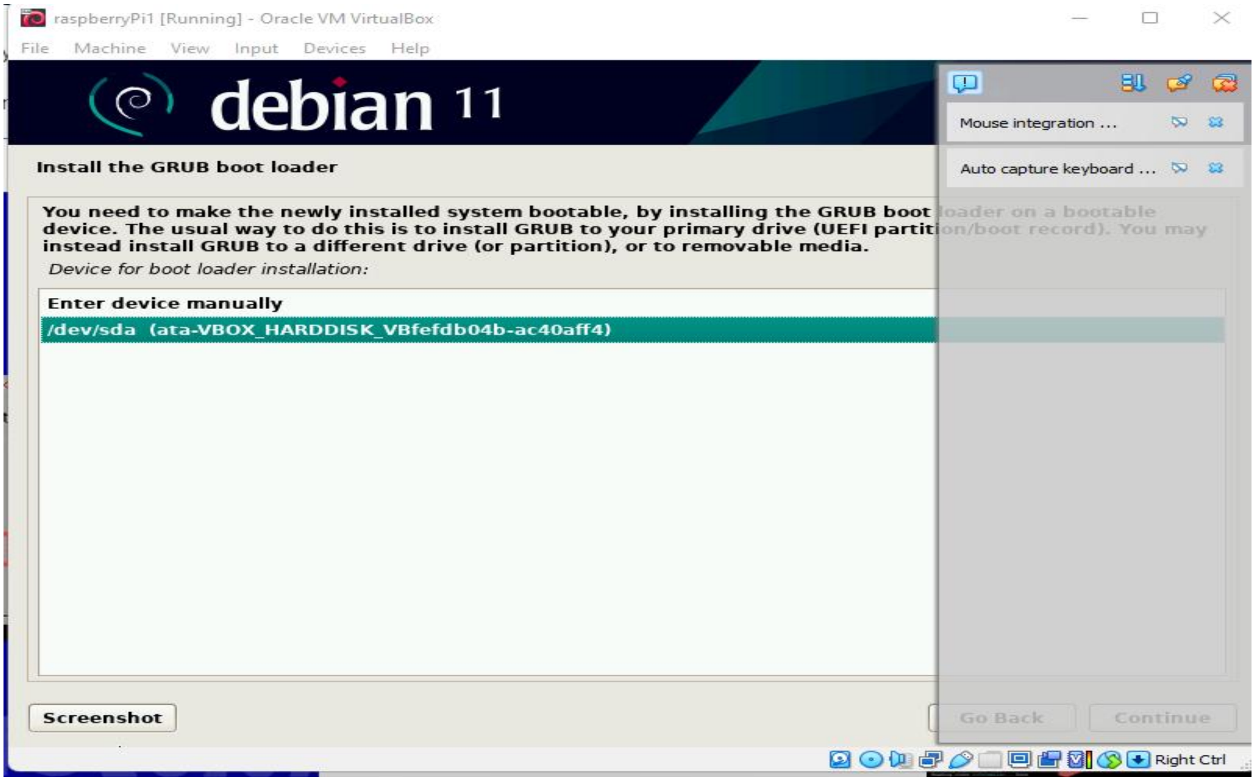
It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to your primary drive (UEFI partition/boot record).

Warning: If your computer has another operating system that the installer failed to detect, this will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to your primary drive?

○ No

● Yes


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Mouse integration ...
Auto capture keyboard ...

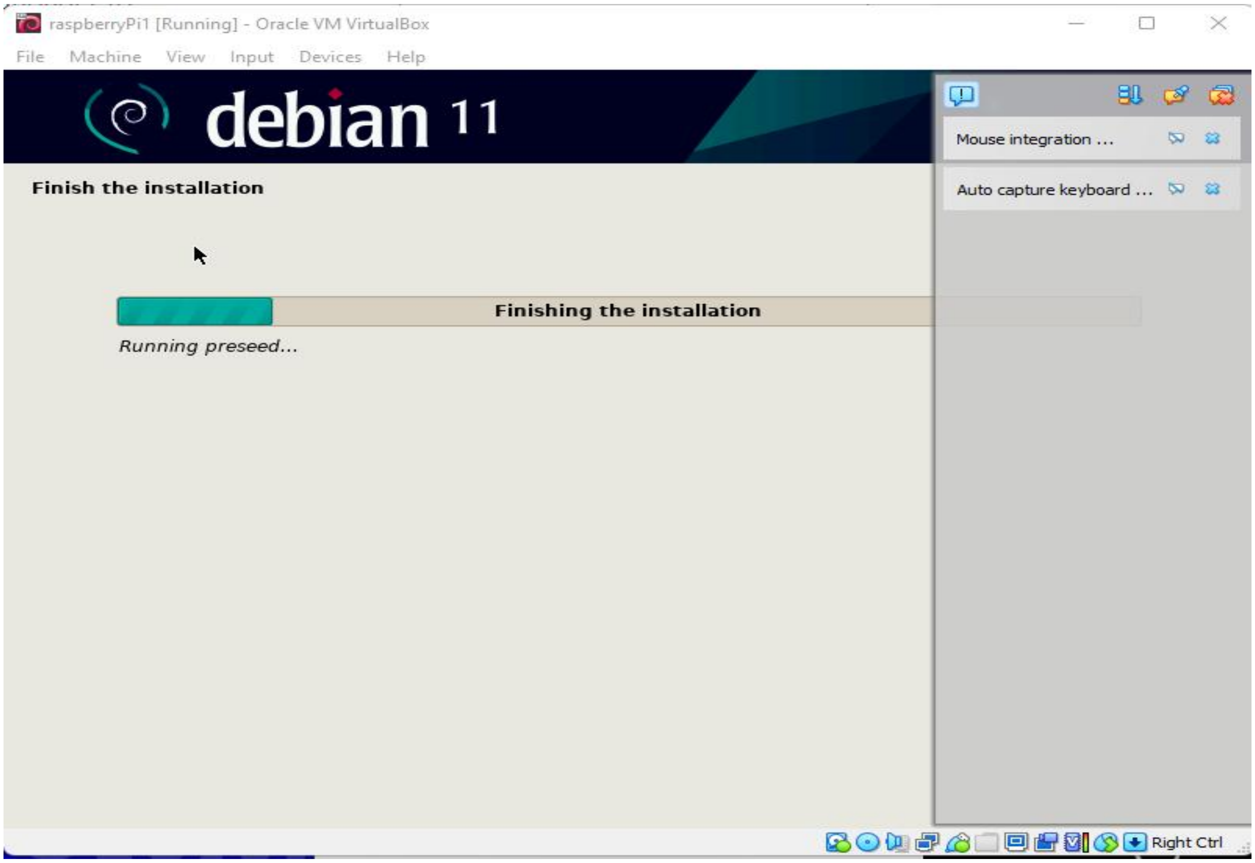
Install the GRUB boot loader

You need to make the newly installed system bootable, by installing the GRUB boot loader on a bootable device. The usual way to do this is to install GRUB to your primary drive (UEFI partition/boot record). You may instead install GRUB to a different drive (or partition), or to removable media.

Device for boot loader installation:

Enter device manually
/dev/sda (ata-VBOX_HARDDISK_VBfefdb04b-ac40aff4)

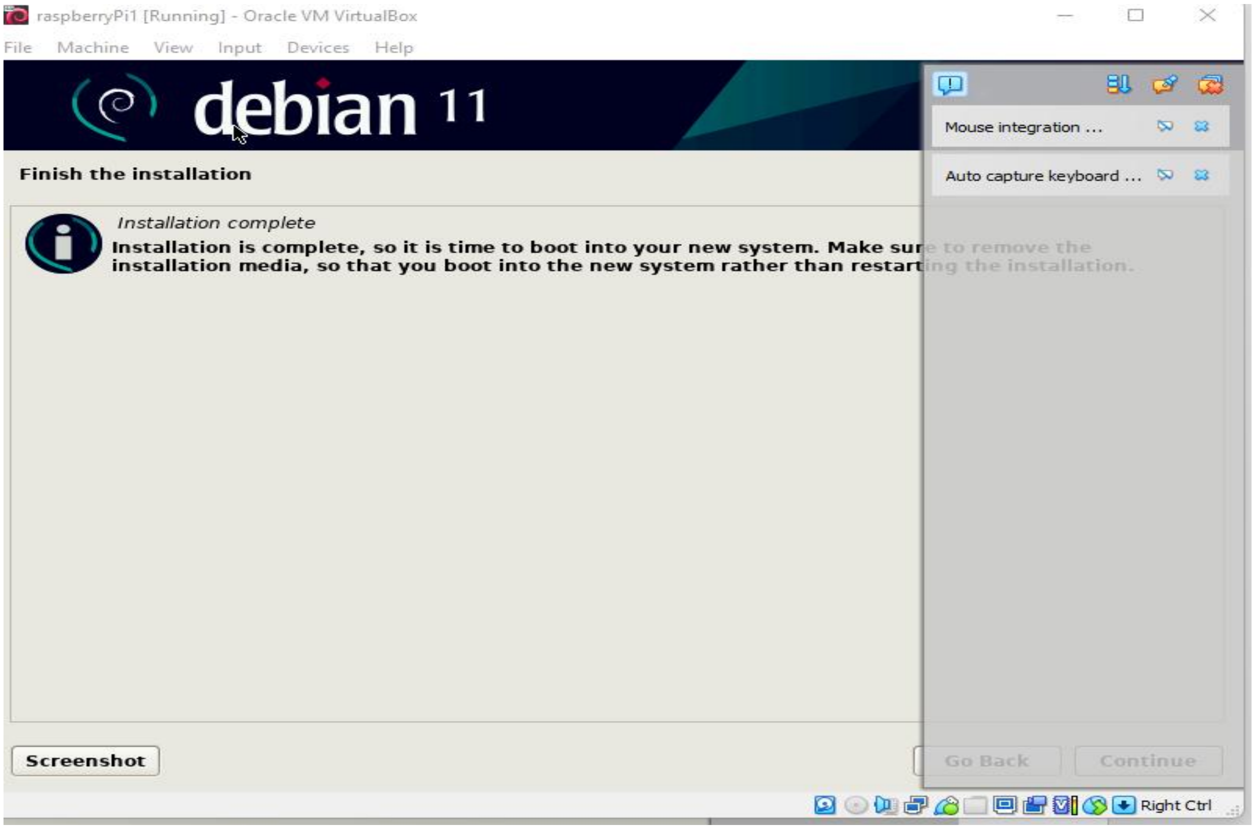
Screenshot
Go Back
Continue
Right Ctrl


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Mouse integration ...
Auto capture keyboard ...

Finish the installation

Finishing the installation

Running preseed...

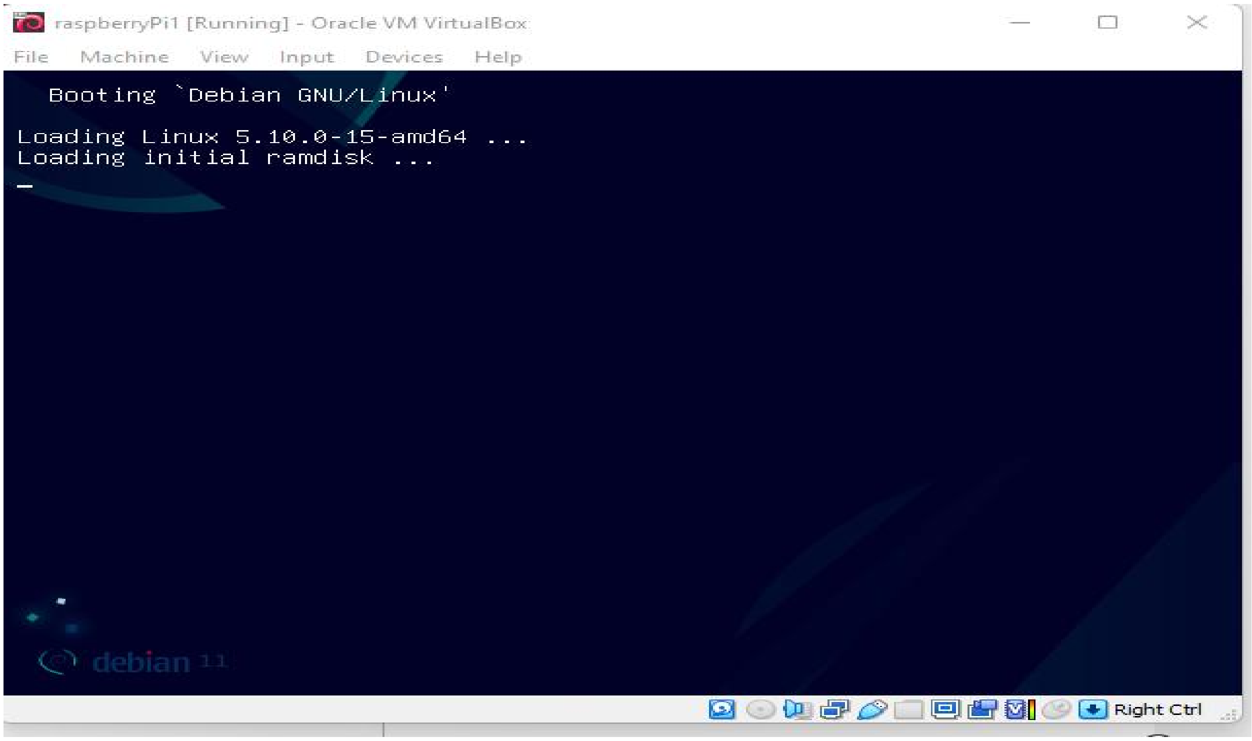


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Mouse integration ...
Auto capture keyboard ...
Screenshot
Go Back
Continue
Right Ctrl

Finish the installation

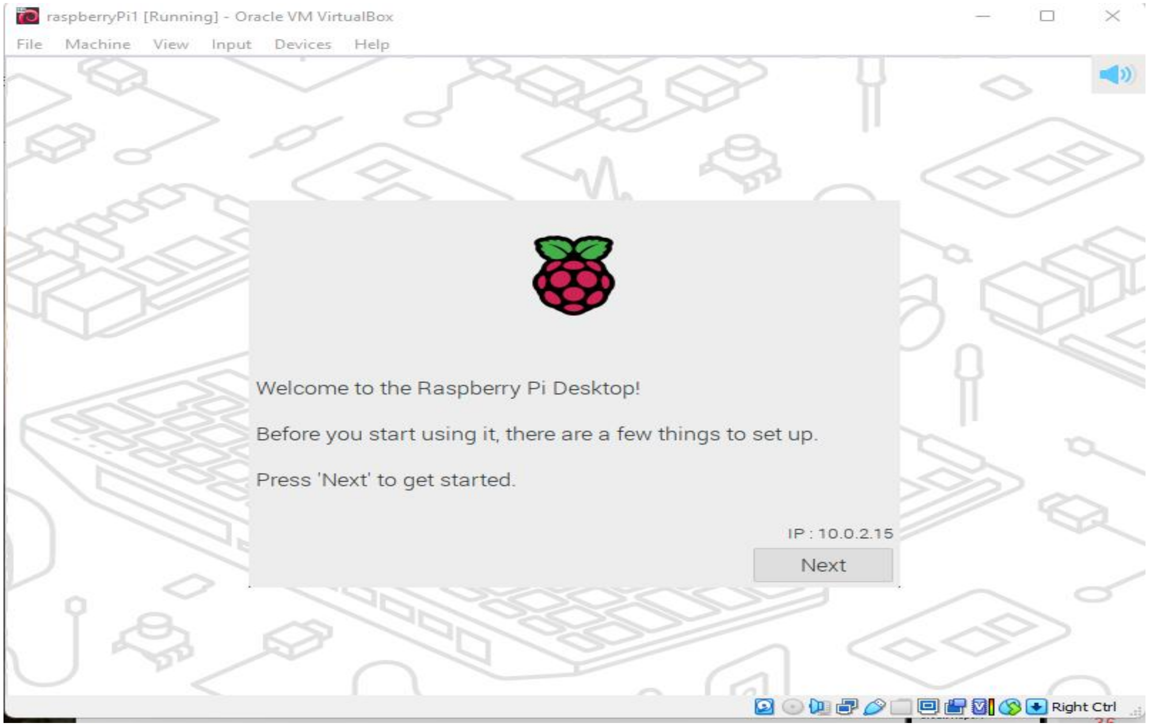
Installation complete

Installation is complete, so it is time to boot into your new system. Make sure to remove the installation media, so that you boot into the new system rather than restarting the installation.



raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Right Ctrl

```
  Booting `Debian GNU/Linux'

Loading Linux 5.10.0-15-amd64 ...
Loading initial ramdisk ...
_
```

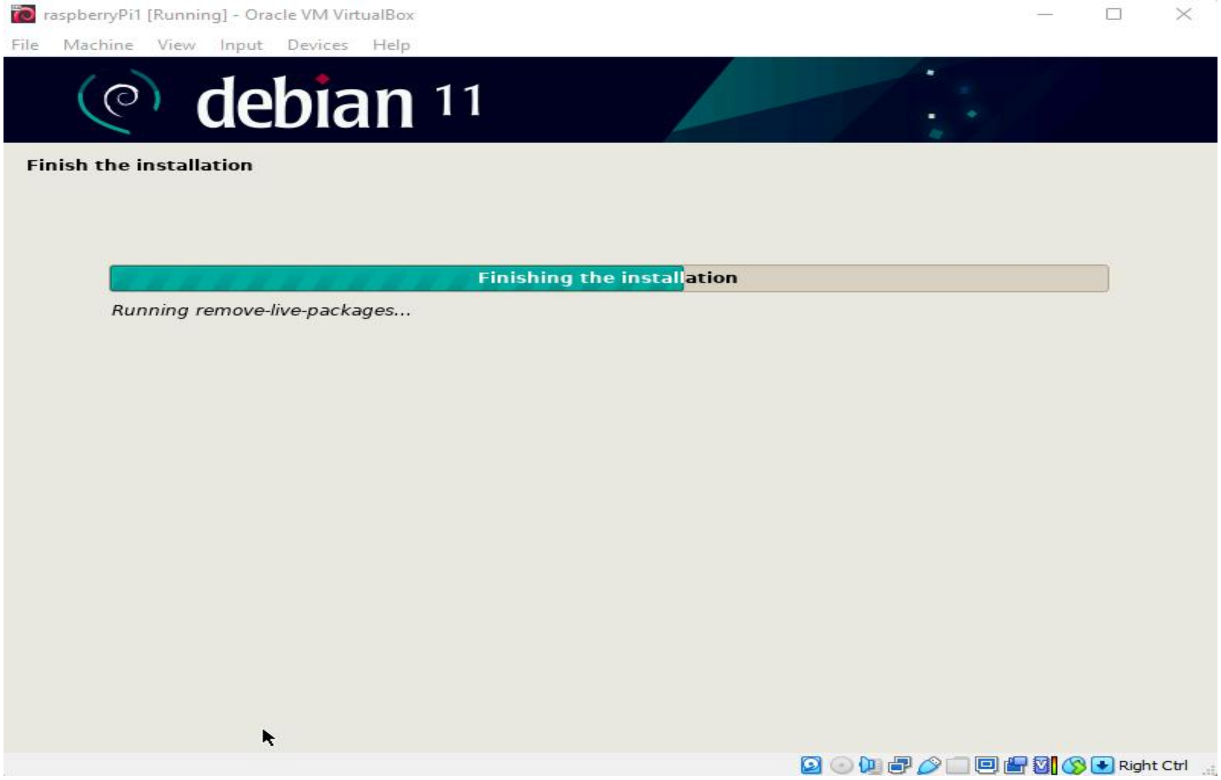

raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

Welcome to the Raspberry Pi Desktop!

Before you start using it, there are a few things to set up.

Press 'Next' to get started.

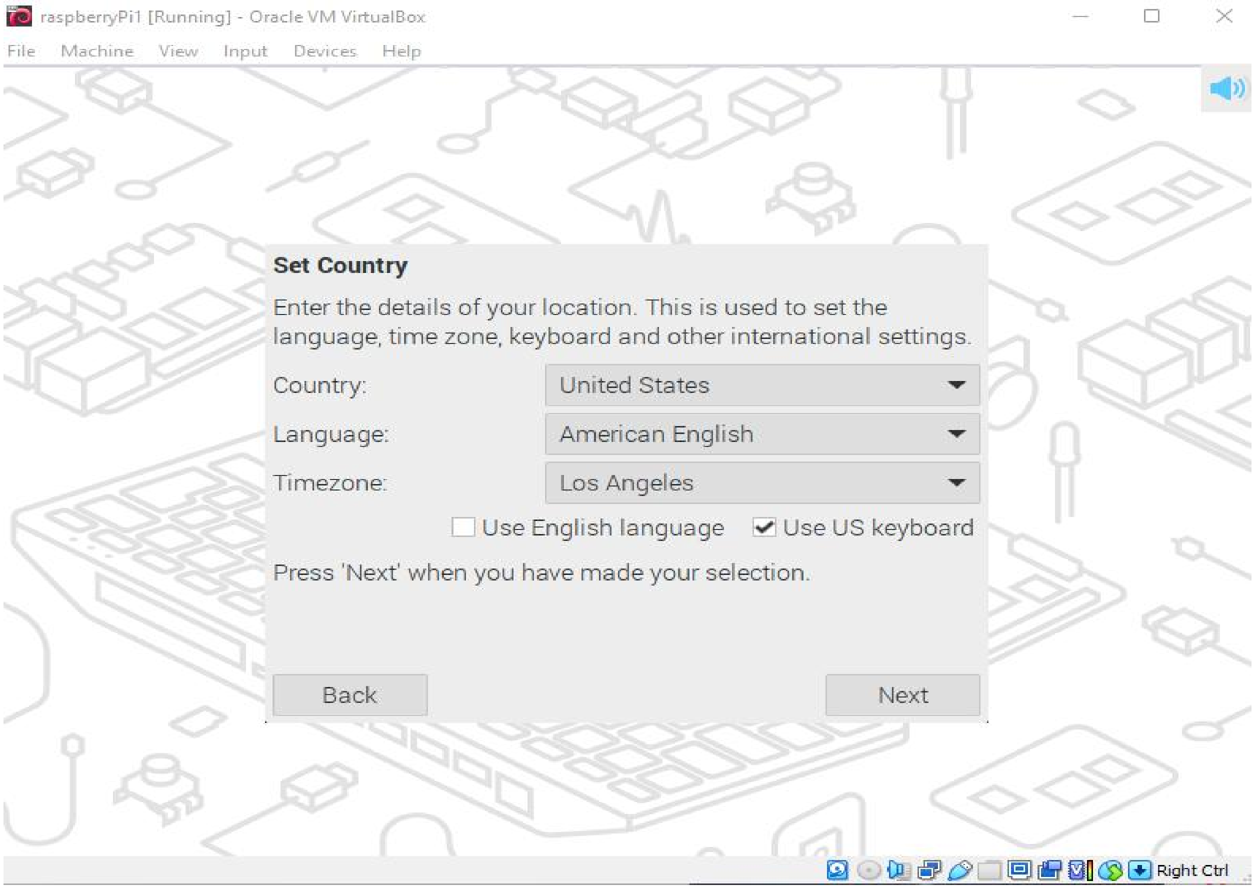
IP : 10.0.2.15


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
debian 11
Right Ctrl

Finish the installation

Finishing the installation

Running remove-live-packages...


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

Set Country

Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

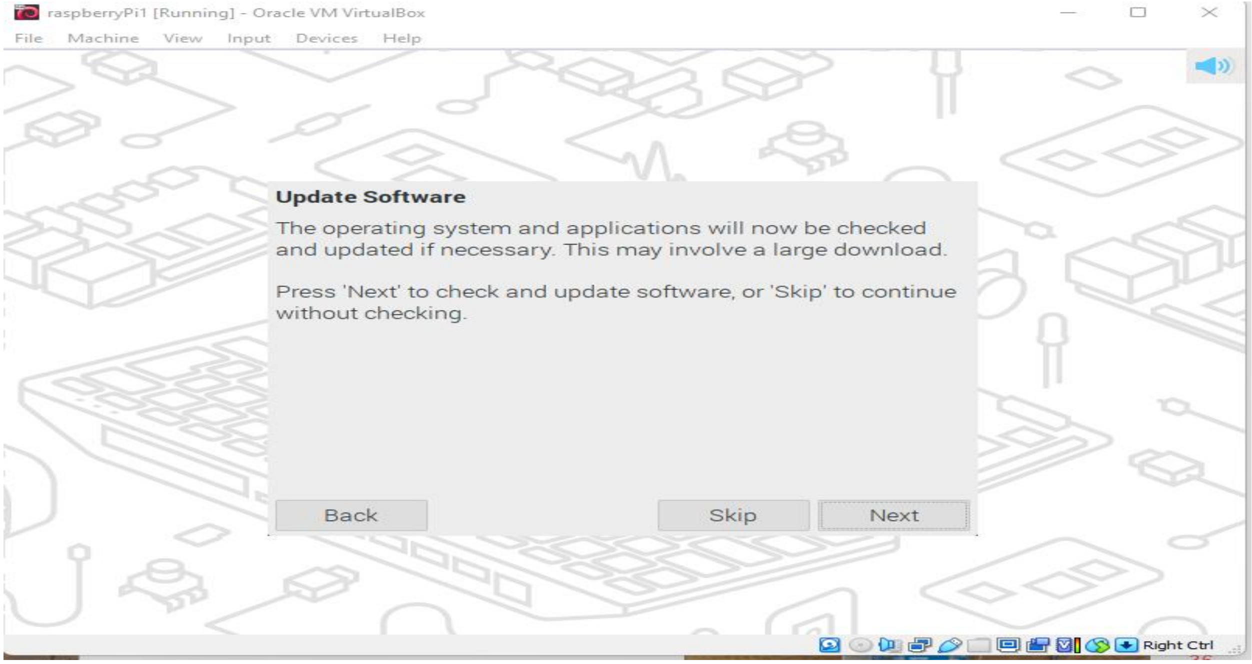
Country: United States

Language: American English

Timezone: Los Angeles

□ Use English language ☑ Use US keyboard

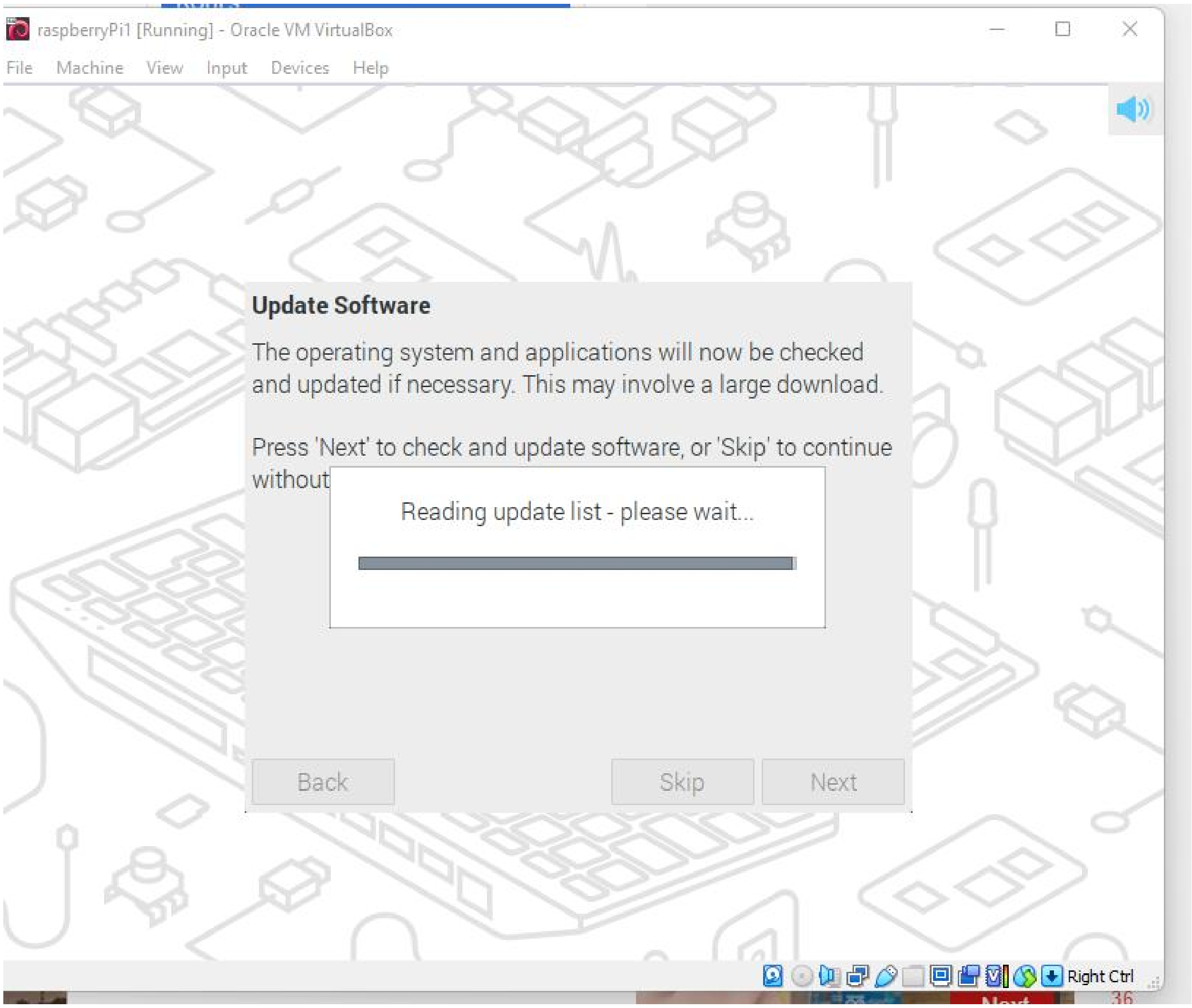
Press 'Next' when you have made your selection.


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' to check and update software, or 'Skip' to continue without checking.



raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

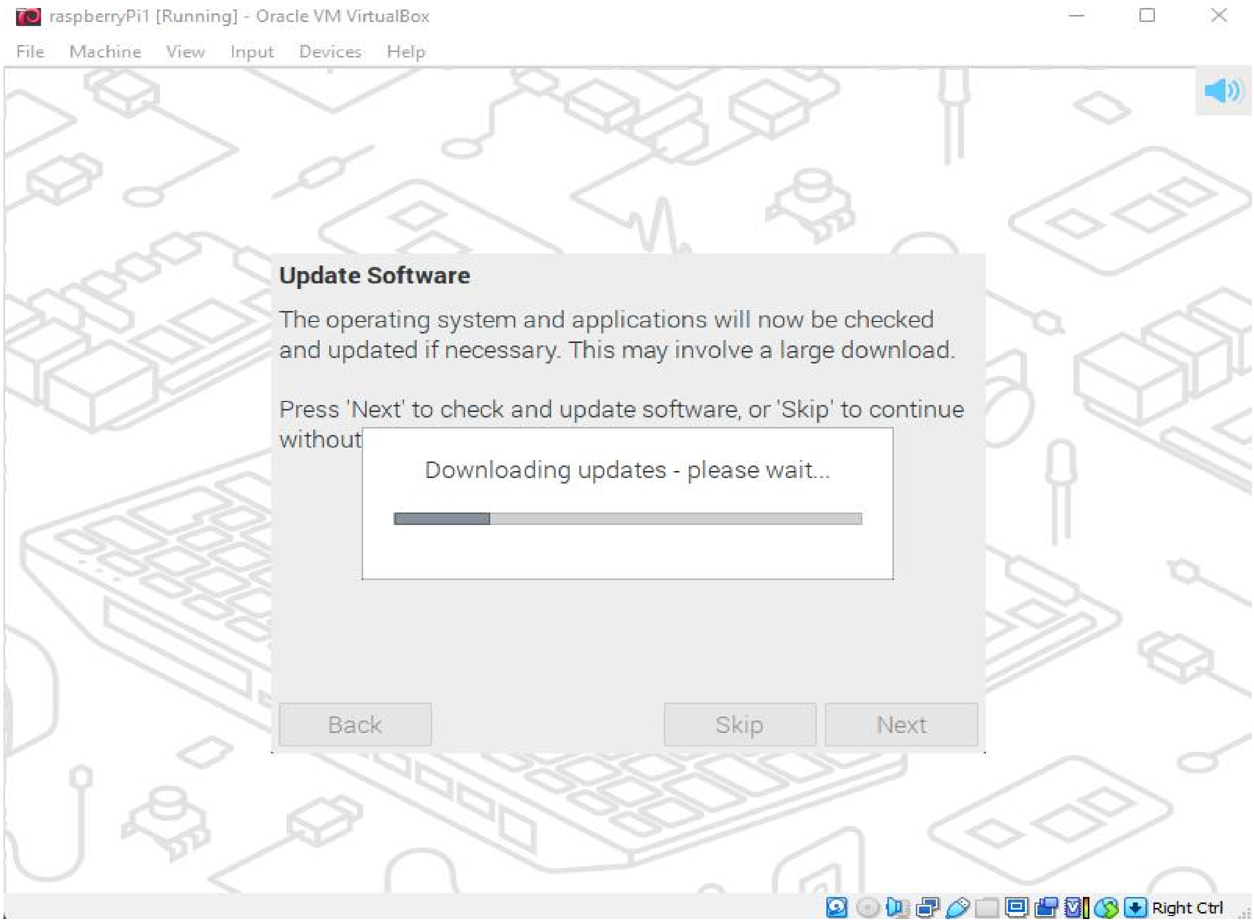
Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' to check and update software, or 'Skip' to continue without

Reading update list - please wait...

Back
Skip
Next
Right Ctrl

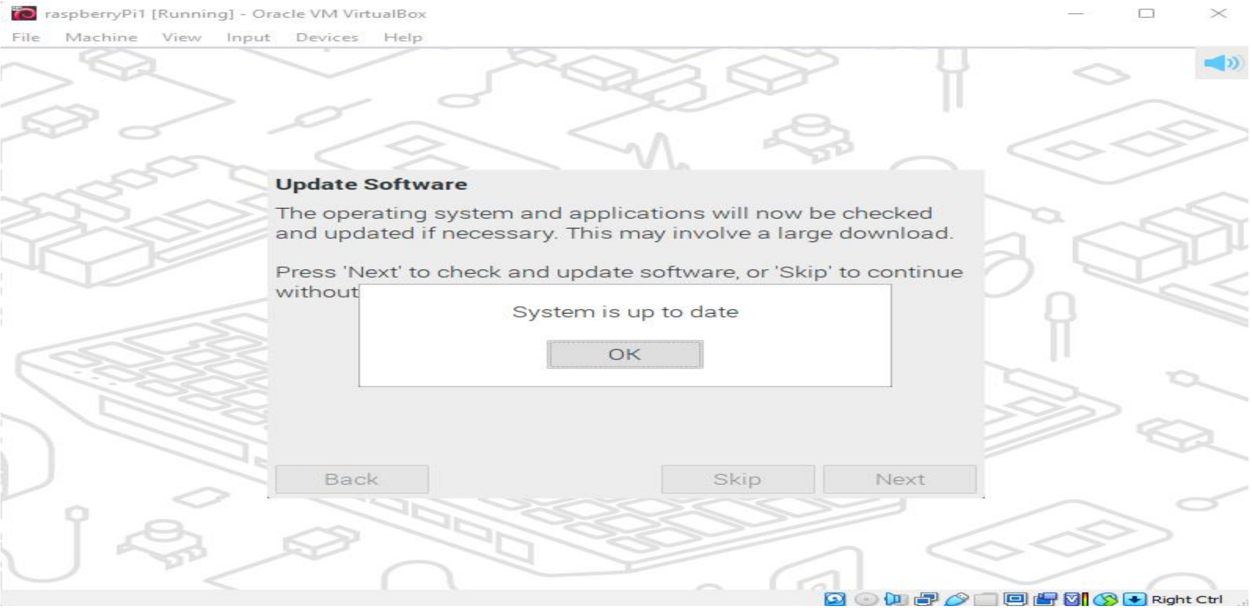


Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' to check and update software, or 'Skip' to continue without

Downloading updates - please wait...

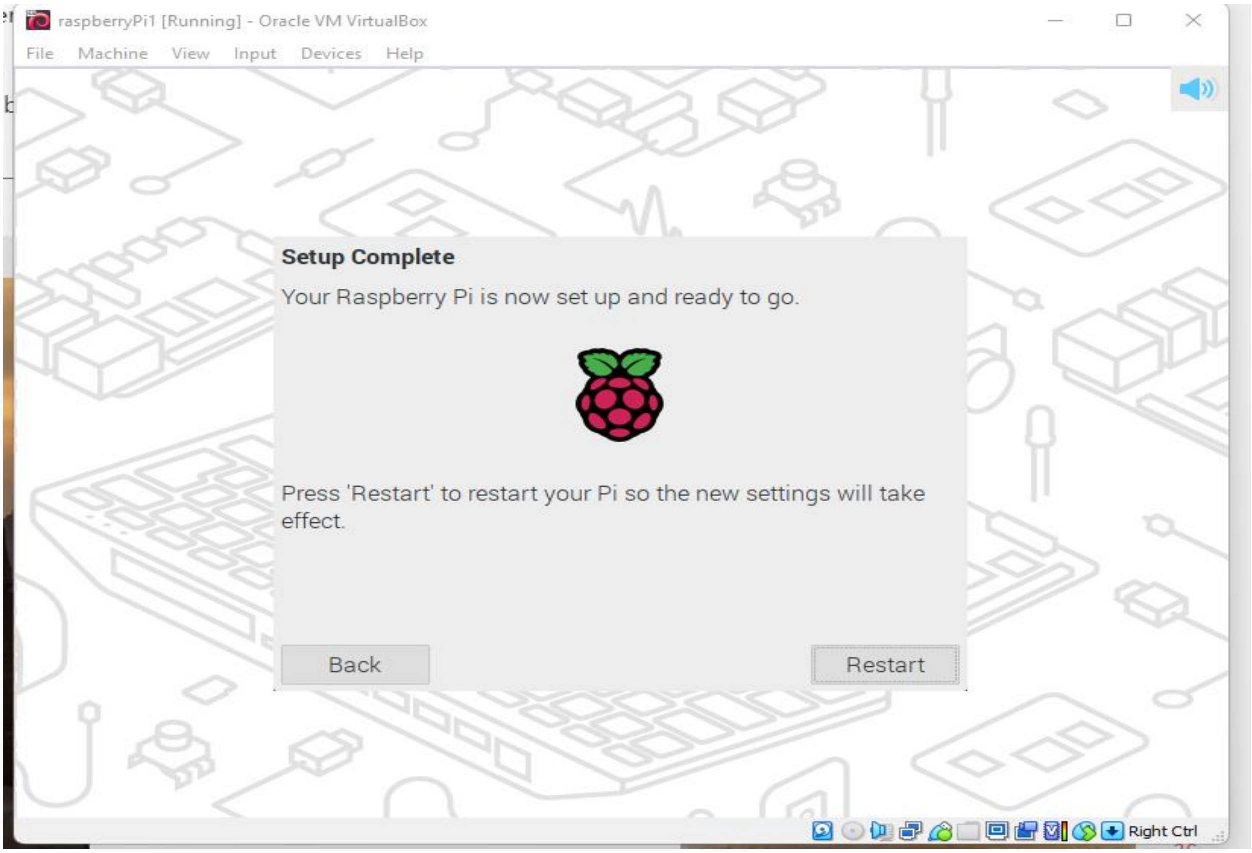


Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' to check and update software, or 'Skip' to continue without

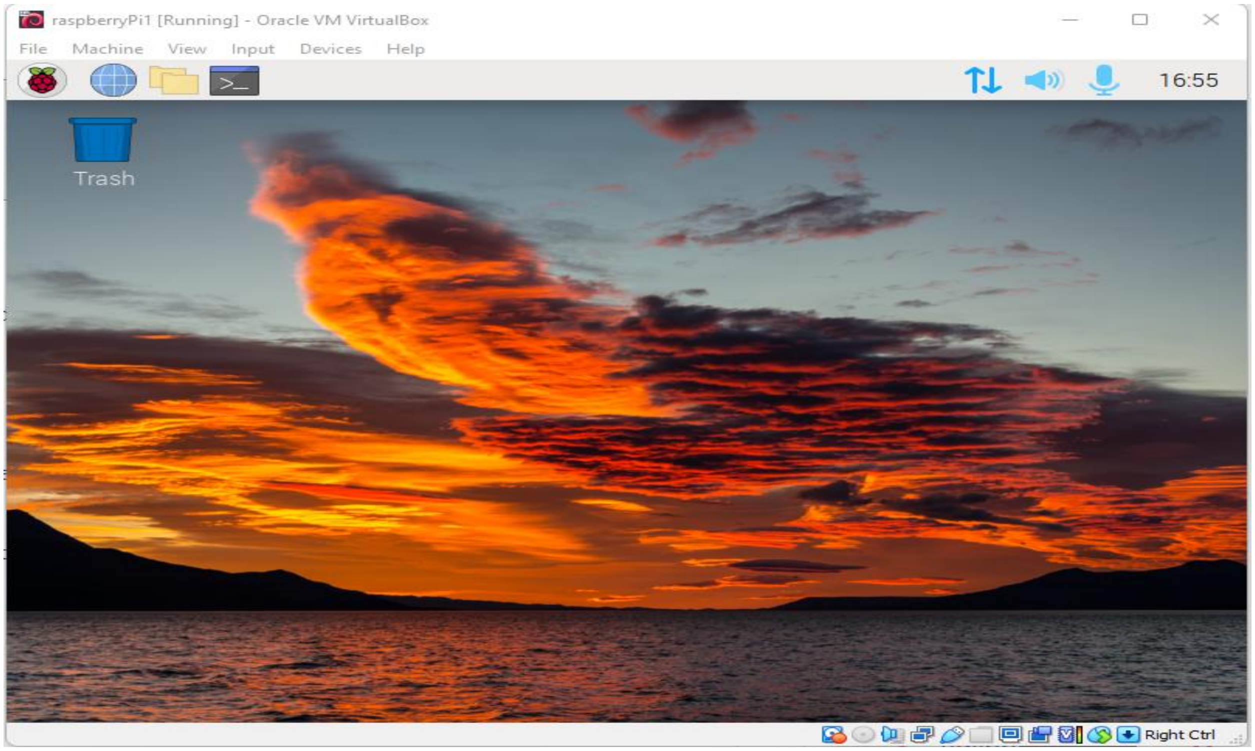
System is up to date


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Back
Restart
Right Ctrl

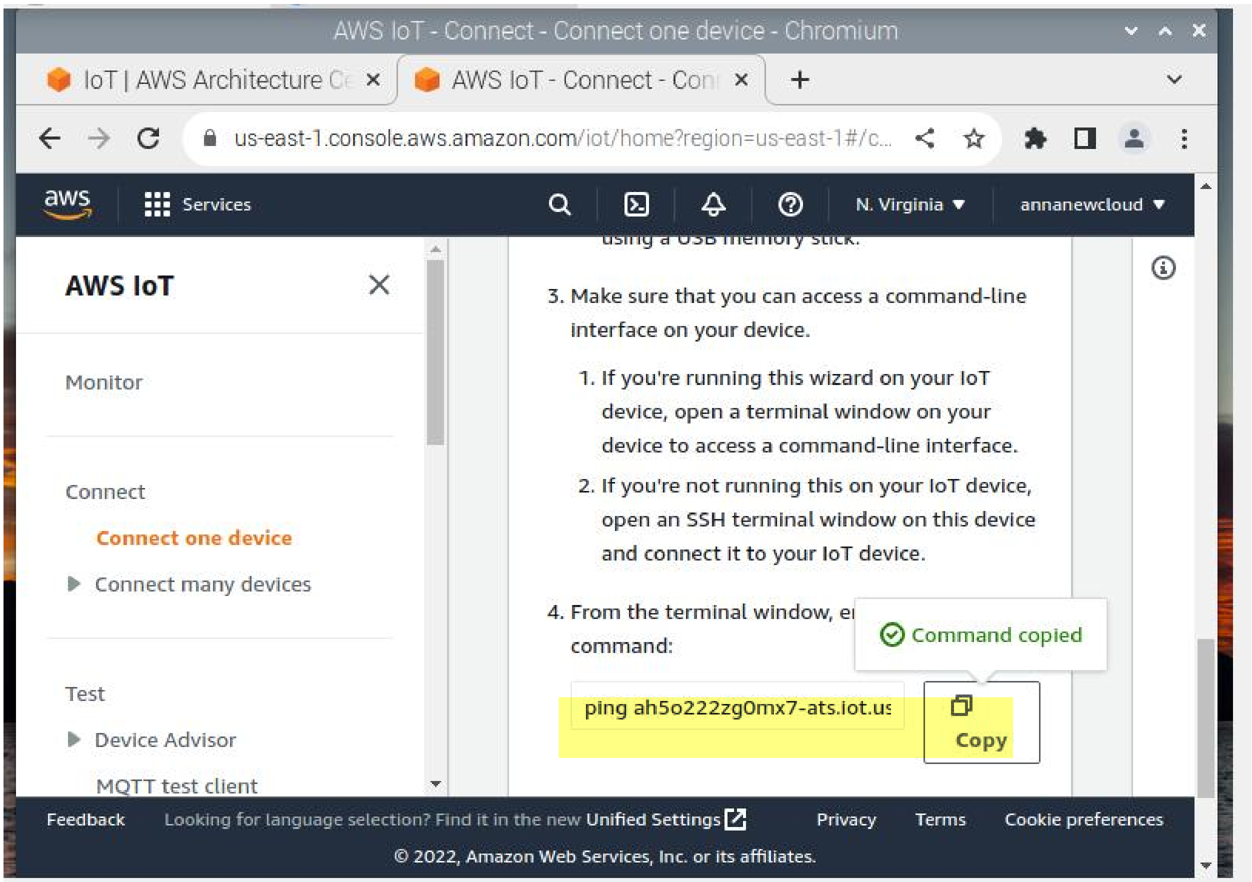
Setup Complete

Your Raspberry Pi is now set up and ready to go.

Press 'Restart' to restart your Pi so the new settings will take effect.


raspberryPi1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
16:55
Trash
Right Ctrl

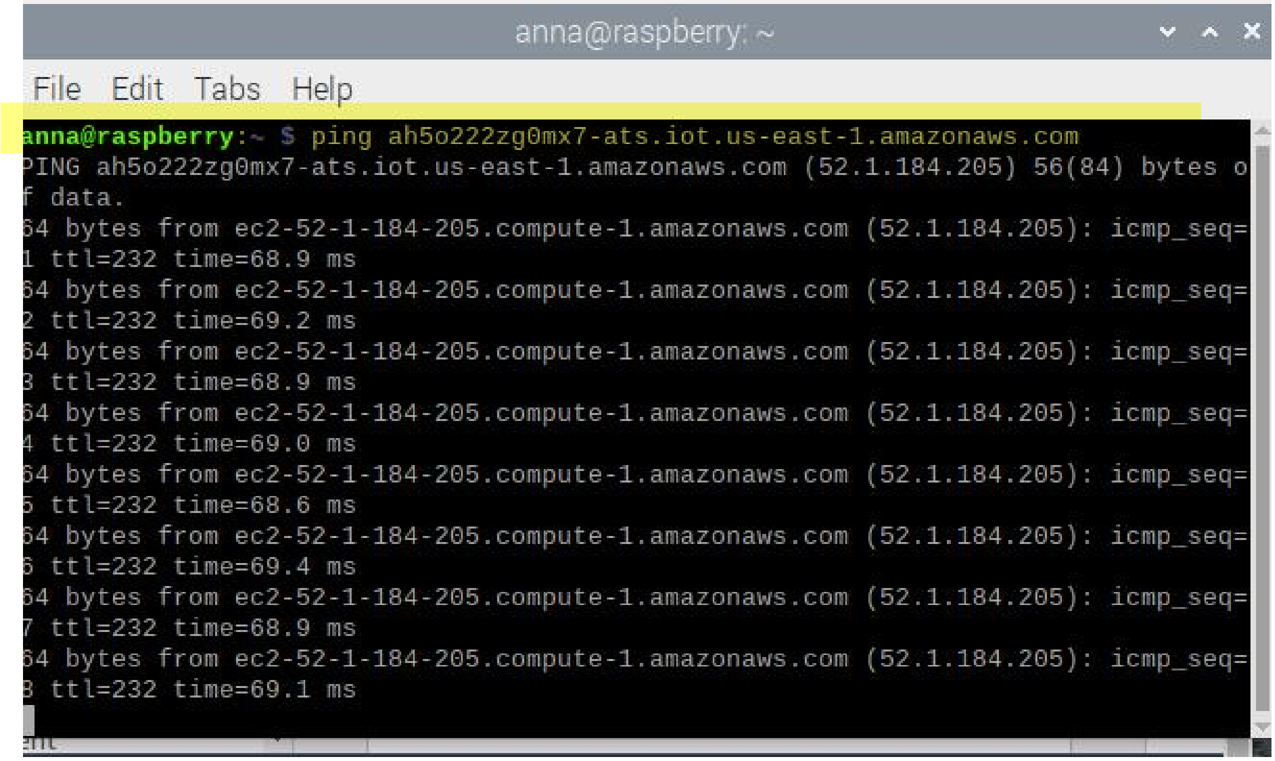
Step 2: Connecting IoT Devices to AWS IoT



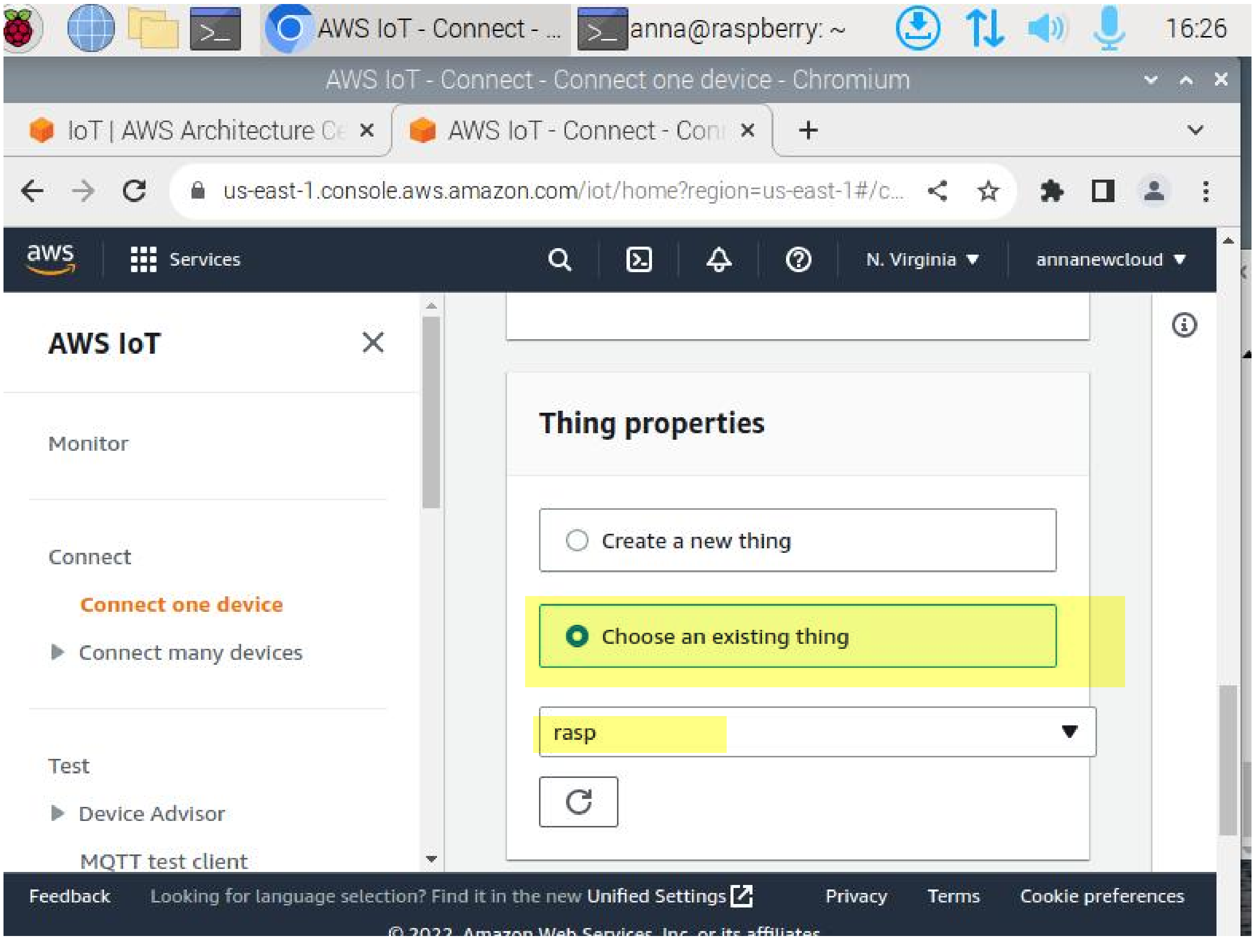
using a USB memory stick.

3. Make sure that you can access a command-line interface on your device.
   1. If you're running this wizard on your IoT device, open a terminal window on your device to access a command-line interface.
   2. If you're not running this on your IoT device, open an SSH terminal window on this device and connect it to your IoT device.
4. From the terminal window, e[illegible] command:

```
ping ah5o222zg0mx7-ats.iot.us
```



```
anna@raspberry:~ $ ping ah5o222zg0mx7-ats.iot.us-east-1.amazonaws.com
PING ah5o222zg0mx7-ats.iot.us-east-1.amazonaws.com (52.1.184.205) 56(84) bytes o
f data.
64 bytes from ec2-52-1-184-205.compute-1.amazonaws.com (52.1.184.205): icmp_seq=
1 ttl=232 time=68.9 ms
64 bytes from ec2-52-1-184-205.compute-1.amazonaws.com (52.1.184.205): icmp_seq=
2 ttl=232 time=69.2 ms
64 bytes from ec2-52-1-184-205.compute-1.amazonaws.com (52.1.184.205): icmp_seq=
3 ttl=232 time=68.9 ms
64 bytes from ec2-52-1-184-205.compute-1.amazonaws.com (52.1.184.205): icmp_seq=
4 ttl=232 time=69.0 ms
64 bytes from ec2-52-1-184-205.compute-1.amazonaws.com (52.1.184.205): icmp_seq=
5 ttl=232 time=68.6 ms
64 bytes from ec2-52-1-184-205.compute-1.amazonaws.com (52.1.184.205): icmp_seq=
6 ttl=232 time=69.4 ms
64 bytes from ec2-52-1-184-205.compute-1.amazonaws.com (52.1.184.205): icmp_seq=
7 ttl=232 time=68.9 ms
64 bytes from ec2-52-1-184-205.compute-1.amazonaws.com (52.1.184.205): icmp_seq=
8 ttl=232 time=69.1 ms
```



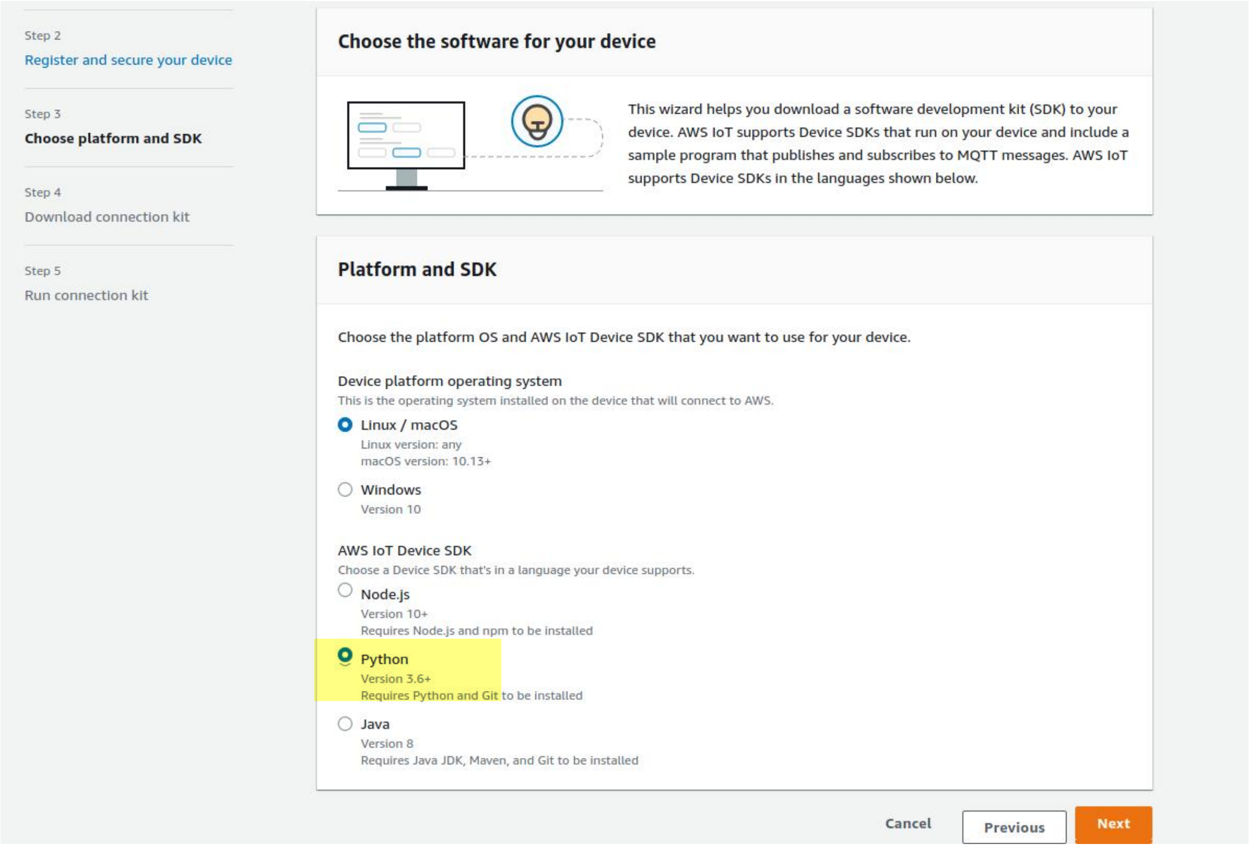
Thing properties

○ Create a new thing

● Choose an existing thing

rasp

AWS IoT - Connect - ...
anna@raspberry: ~
16:26
AWS IoT - Connect - Connect one device - Chromium
IoT | AWS Architecture Ce
AWS IoT - Connect - Con
us-east-1.console.aws.amazon.com/iot/home?region=us-east-1#/c...
aws
Services
N. Virginia ▼
annanewcloud ▼
AWS IoT
Monitor
Connect
Connect one device
Connect many devices
Test
Device Advisor
MQTT test client
Feedback
Looking for language selection? Find it in the new Unified Settings
Privacy
Terms
Cookie preferences
© 2022, Amazon Web Services, Inc. or its affiliates.



Choose the software for your device

This wizard helps you download a software development kit (SDK) to your device. AWS IoT supports Device SDKs that run on your device and include a sample program that publishes and subscribes to MQTT messages. AWS IoT supports Device SDKs in the languages shown below.

Platform and SDK

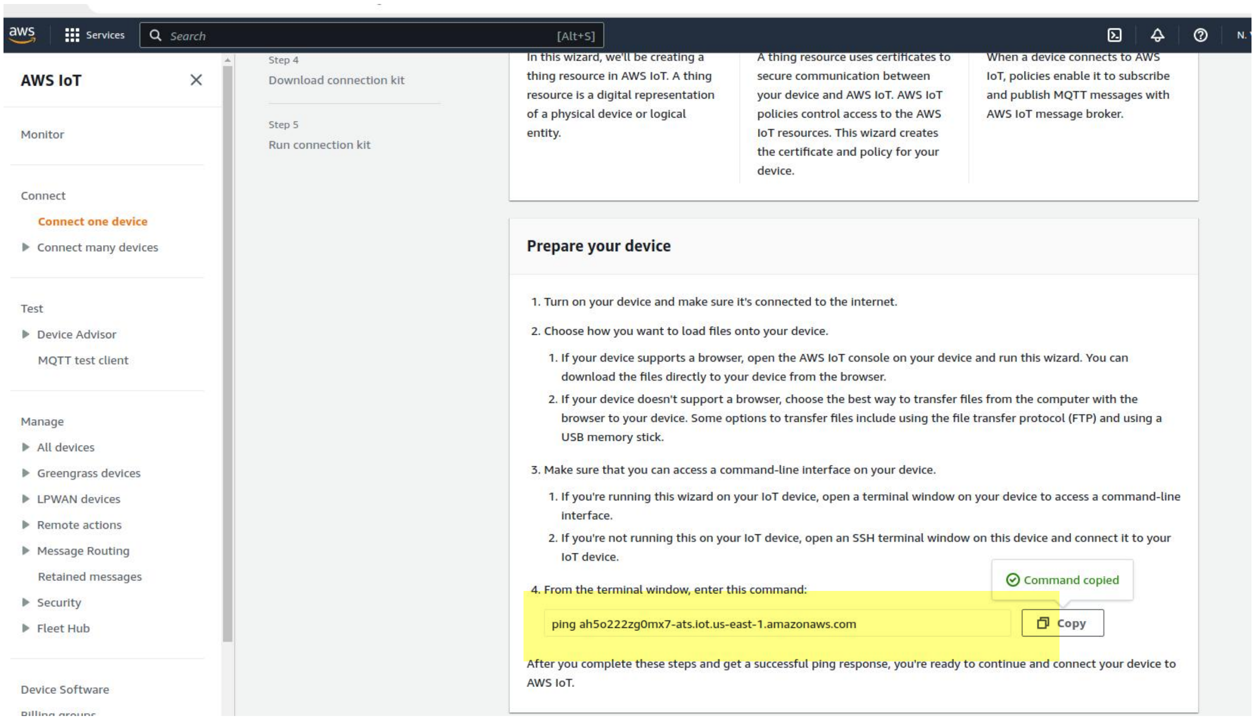
Choose the platform OS and AWS IoT Device SDK that you want to use for your device.

Device platform operating system
This is the operating system installed on the device that will connect to AWS.

- ● Linux / macOS
  Linux version: any
  macOS version: 10.13+
- ○ Windows
  Version 10

AWS IoT Device SDK
Choose a Device SDK that's in a language your device supports.

- ○ Node.js
  Version 10+
  Requires Node.js and npm to be installed
- ● Python
  Version 3.6+
  Requires Python and Git to be installed
- ○ Java
  Version 8
  Requires Java JDK, Maven, and Git to be installed



In this wizard, we'll be creating a thing resource in AWS IoT. A thing resource is a digital representation of a physical device or logical entity.

A thing resource uses certificates to secure communication between your device and AWS IoT. AWS IoT policies control access to the AWS IoT resources. This wizard creates the certificate and policy for your device.

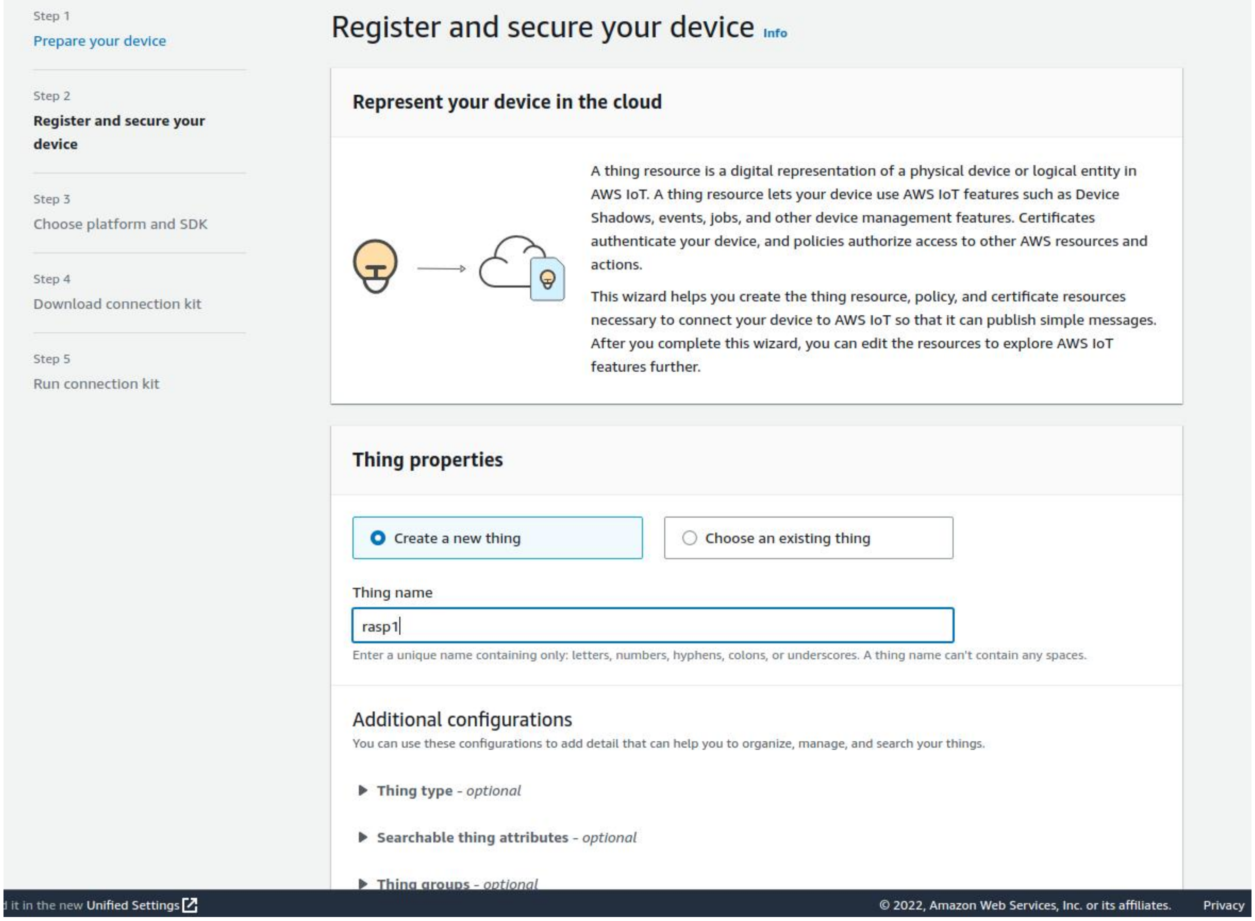
When a device connects to AWS IoT, policies enable it to subscribe and publish MQTT messages with AWS IoT message broker.

Prepare your device

1. Turn on your device and make sure it's connected to the internet.
2. Choose how you want to load files onto your device.
   1. If your device supports a browser, open the AWS IoT console on your device and run this wizard. You can download the files directly to your device from the browser.
   2. If your device doesn't support a browser, choose the best way to transfer files from the computer with the browser to your device. Some options to transfer files include using the file transfer protocol (FTP) and using a USB memory stick.
3. Make sure that you can access a command-line interface on your device.
   1. If you're running this wizard on your IoT device, open a terminal window on your device to access a command-line interface.
   2. If you're not running this on your IoT device, open an SSH terminal window on this device and connect it to your IoT device.
4. From the terminal window, enter this command:

```
ping ah5o222zg0mx7-ats.iot.us-east-1.amazonaws.com
```

After you complete these steps and get a successful ping response, you're ready to continue and connect your device to AWS IoT.



Register and secure your device

Represent your device in the cloud

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. A thing resource lets your device use AWS IoT features such as Device Shadows, events, jobs, and other device management features. Certificates authenticate your device, and policies authorize access to other AWS resources and actions.

This wizard helps you create the thing resource, policy, and certificate resources necessary to connect your device to AWS IoT so that it can publish simple messages. After you complete this wizard, you can edit the resources to explore AWS IoT features further.

Thing properties

● Create a new thing ○ Choose an existing thing

Thing name

rasp1

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

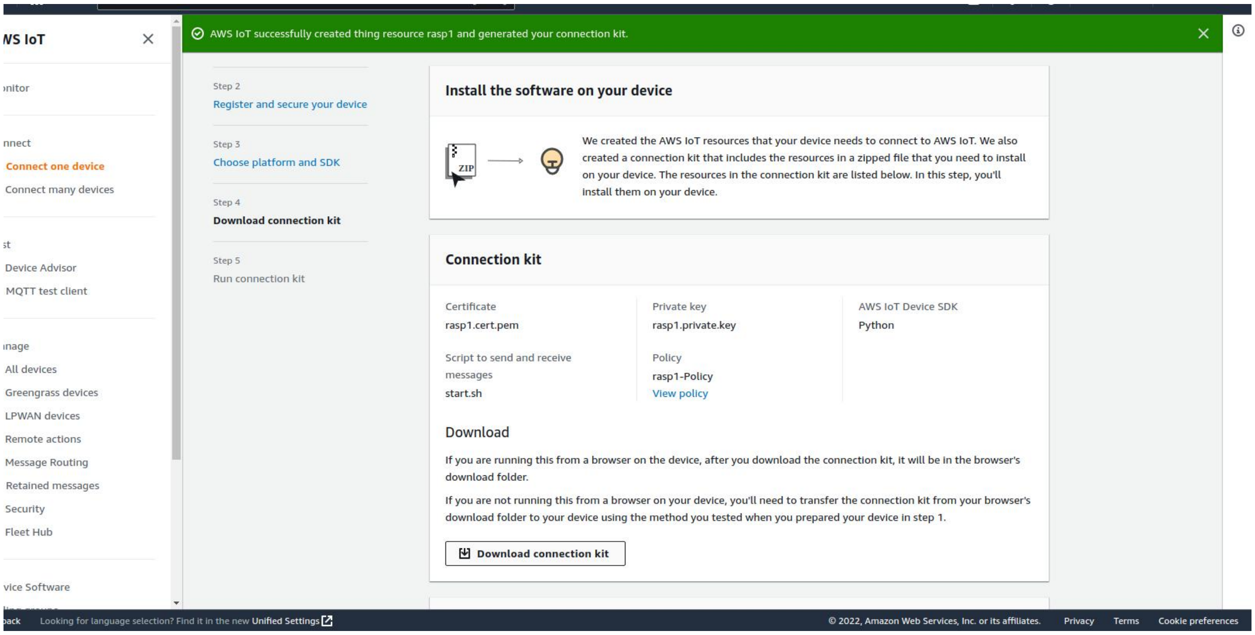
Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

▶ **Thing type** - *optional*

▶ **Searchable thing attributes** - *optional*

▶ **Thing groups** - *optional*



AWS IoT successfully created thing resource rasp1 and generated your connection kit.

Install the software on your device

We created the AWS IoT resources that your device needs to connect to AWS IoT. We also created a connection kit that includes the resources in a zipped file that you need to install on your device. The resources in the connection kit are listed below. In this step, you'll install them on your device.

Connection kit

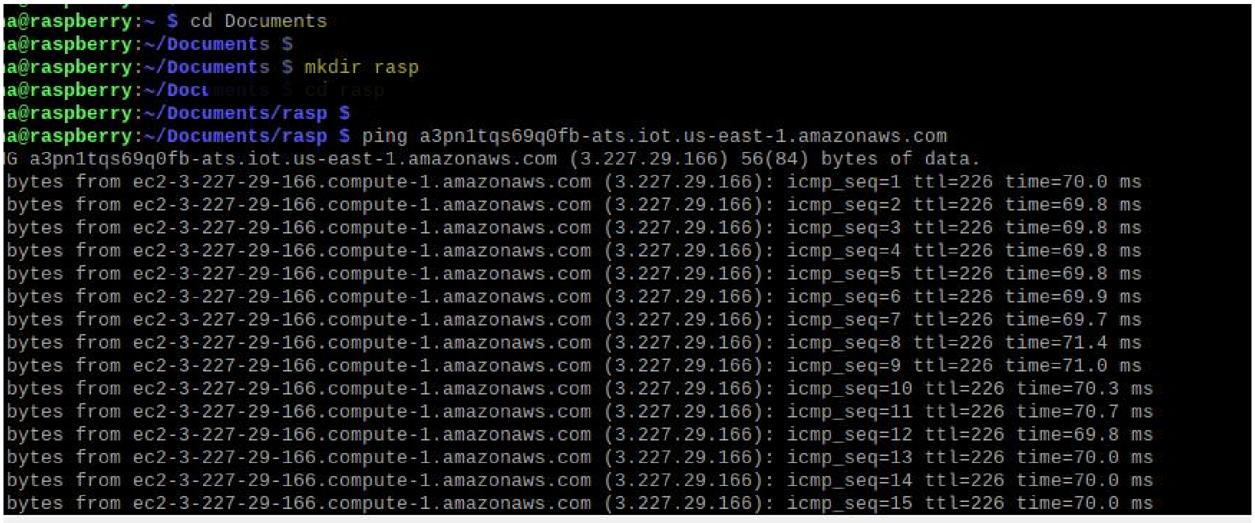
Certificate rasp1.cert.pem	Private key rasp1.private.key	AWS IoT Device SDK Python
Script to send and receive messages start.sh	Policy rasp1-Policy View policy	

Download

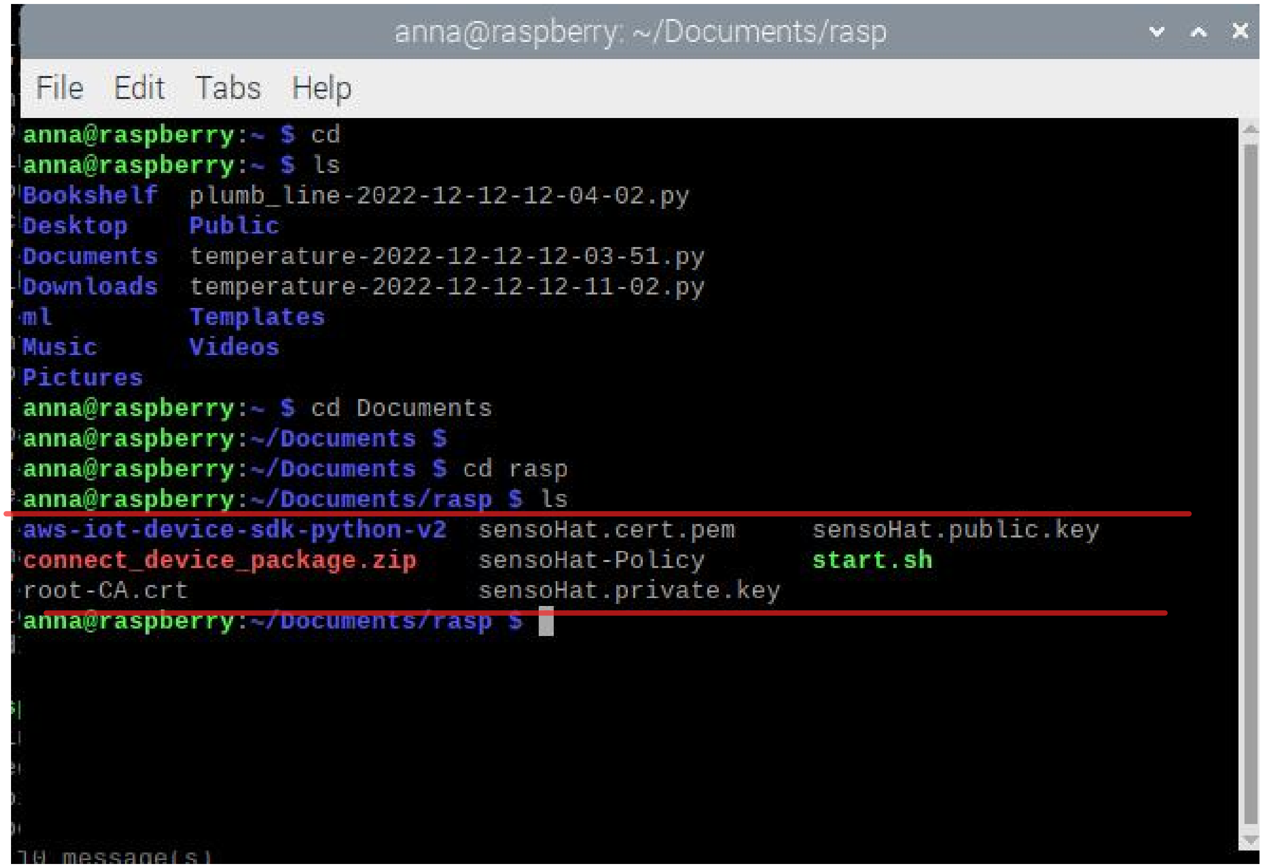
If you are running this from a browser on the device, after you download the connection kit, it will be in the browser's download folder.

If you are not running this from a browser on your device, you'll need to transfer the connection kit from your browser's download folder to your device using the method you tested when you prepared your device in step 1.

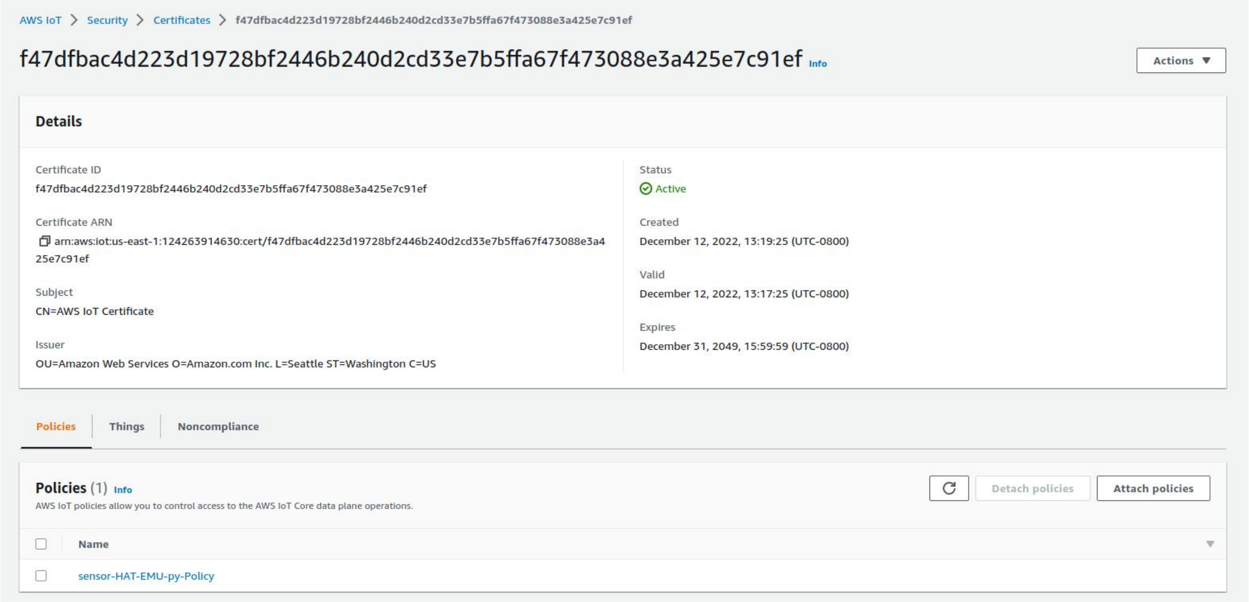
Download connection kit



```
a@raspberry:~ $ cd Documents
a@raspberry:~/Documents $
a@raspberry:~/Documents $ mkdir rasp
a@raspberry:~/Documents $ cd rasp
a@raspberry:~/Documents/rasp $
a@raspberry:~/Documents/rasp $ ping a3pn1tqs69q0fb-ats.iot.us-east-1.amazonaws.com
G a3pn1tqs69q0fb-ats.iot.us-east-1.amazonaws.com (3.227.29.166) 56(84) bytes of data.
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=1 ttl=226 time=70.0 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=2 ttl=226 time=69.8 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=3 ttl=226 time=69.8 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=4 ttl=226 time=69.8 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=5 ttl=226 time=69.8 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=6 ttl=226 time=69.9 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=7 ttl=226 time=69.7 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=8 ttl=226 time=71.4 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=9 ttl=226 time=71.0 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=10 ttl=226 time=70.3 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=11 ttl=226 time=70.7 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=12 ttl=226 time=69.8 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=13 ttl=226 time=70.0 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=14 ttl=226 time=70.0 ms
bytes from ec2-3-227-29-166.compute-1.amazonaws.com (3.227.29.166): icmp_seq=15 ttl=226 time=70.0 ms
```



```
anna@raspberry:~ $ cd
anna@raspberry:~ $ ls
Bookshelf  plumb_line-2022-12-12-12-04-02.py
Desktop    Public
Documents  temperature-2022-12-12-12-03-51.py
Downloads  temperature-2022-12-12-12-11-02.py
ml         Templates
Music      Videos
Pictures
anna@raspberry:~ $ cd Documents
anna@raspberry:~/Documents $
anna@raspberry:~/Documents $ cd rasp
anna@raspberry:~/Documents/rasp $ ls
aws-iot-device-sdk-python-v2  sensoHat.cert.pem     sensoHat.public.key
connect_device_package.zip    sensoHat-Policy       start.sh
root-CA.crt                   sensoHat.private.key
anna@raspberry:~/Documents/rasp $
```



AWS IoT > Security > Certificates > f47dfbac4d223d19728bf2446b240d2cd33e7b5ffa67f473088e3a425e7c91ef
Actions ▼
Info
Policies
Things
Noncompliance
Detach policies
Attach policies

f47dfbac4d223d19728bf2446b240d2cd33e7b5ffa67f473088e3a425e7c91ef

Details

Certificate ID
f47dfbac4d223d19728bf2446b240d2cd33e7b5ffa67f473088e3a425e7c91ef

Certificate ARN
arn:aws:iot:us-east-1:124263914630:cert/f47dfbac4d223d19728bf2446b240d2cd33e7b5ffa67f473088e3a425e7c91ef

Subject
CN=AWS IoT Certificate

Issuer
OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US

Status
Active

Created
December 12, 2022, 13:19:25 (UTC-0800)

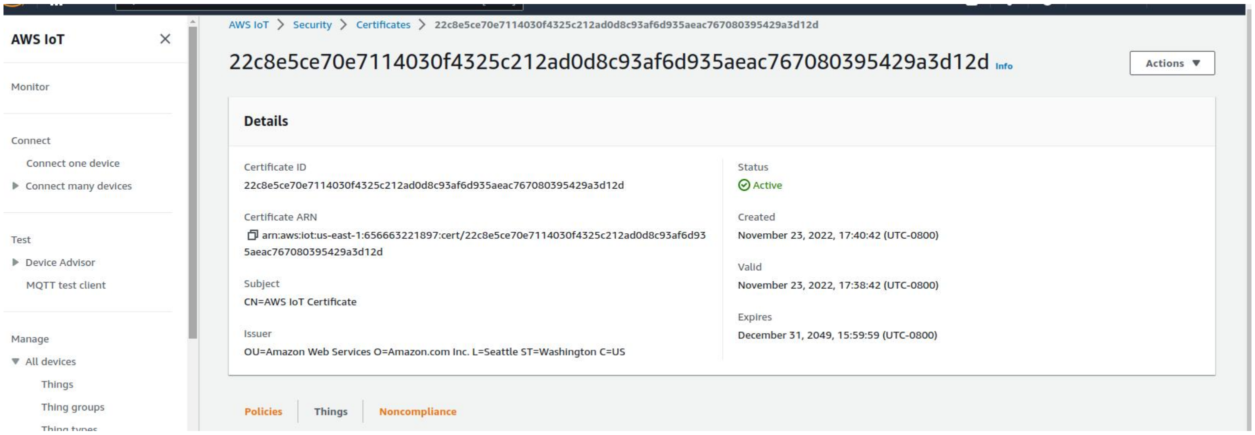
Valid
December 12, 2022, 13:17:25 (UTC-0800)

Expires
December 31, 2049, 15:59:59 (UTC-0800)

Policies (1)

AWS IoT policies allow you to control access to the AWS IoT Core data plane operations.

□	Name
□	sensor-HAT-EMU-py-Policy



22c8e5ce70e7114030f4325c212ad0d8c93af6d935aeac767080395429a3d12d

Details

Certificate ID
22c8e5ce70e7114030f4325c212ad0d8c93af6d935aeac767080395429a3d12d

Certificate ARN
arn:aws:iot:us-east-1:656663221897:cert/22c8e5ce70e7114030f4325c212ad0d8c93af6d935aeac767080395429a3d12d

Subject
CN=AWS IoT Certificate

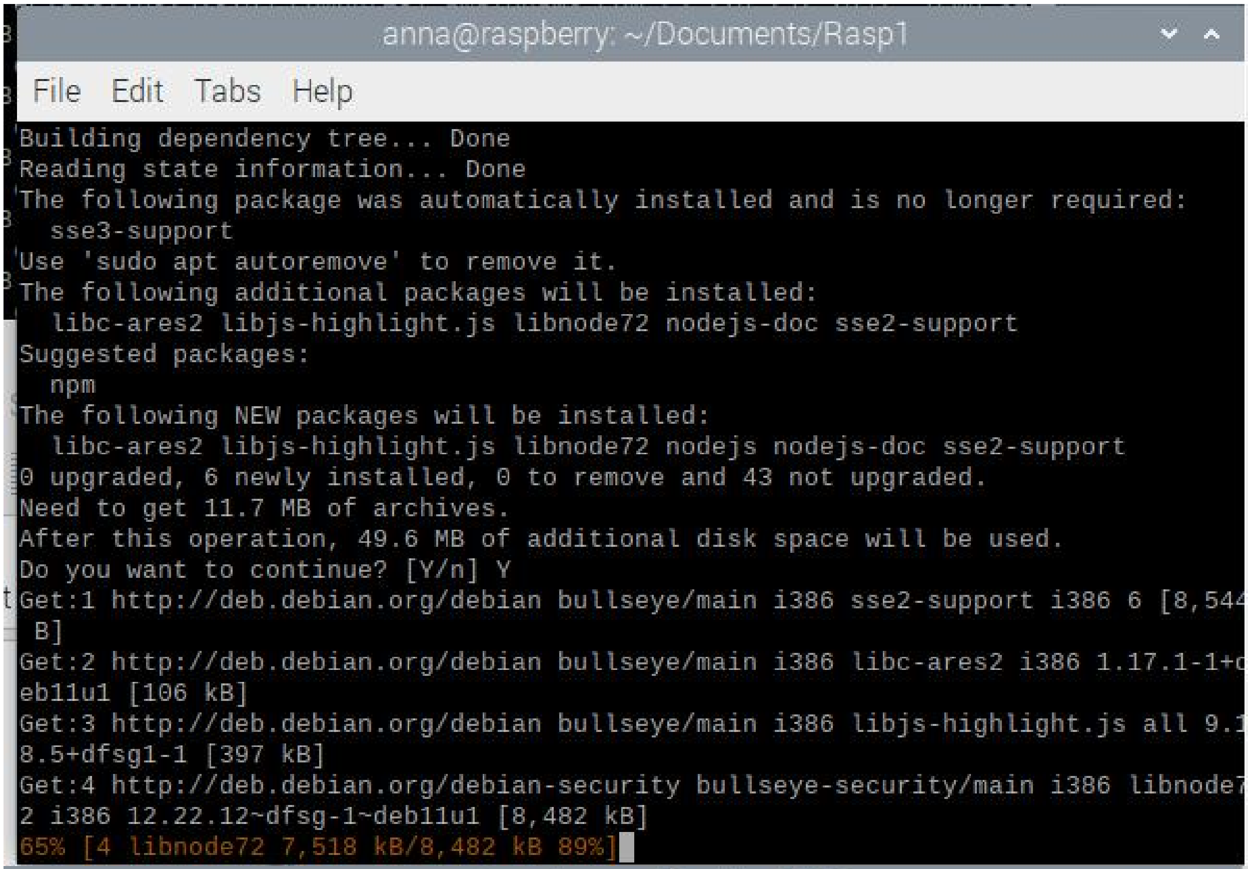
Issuer
OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US

Status
Active

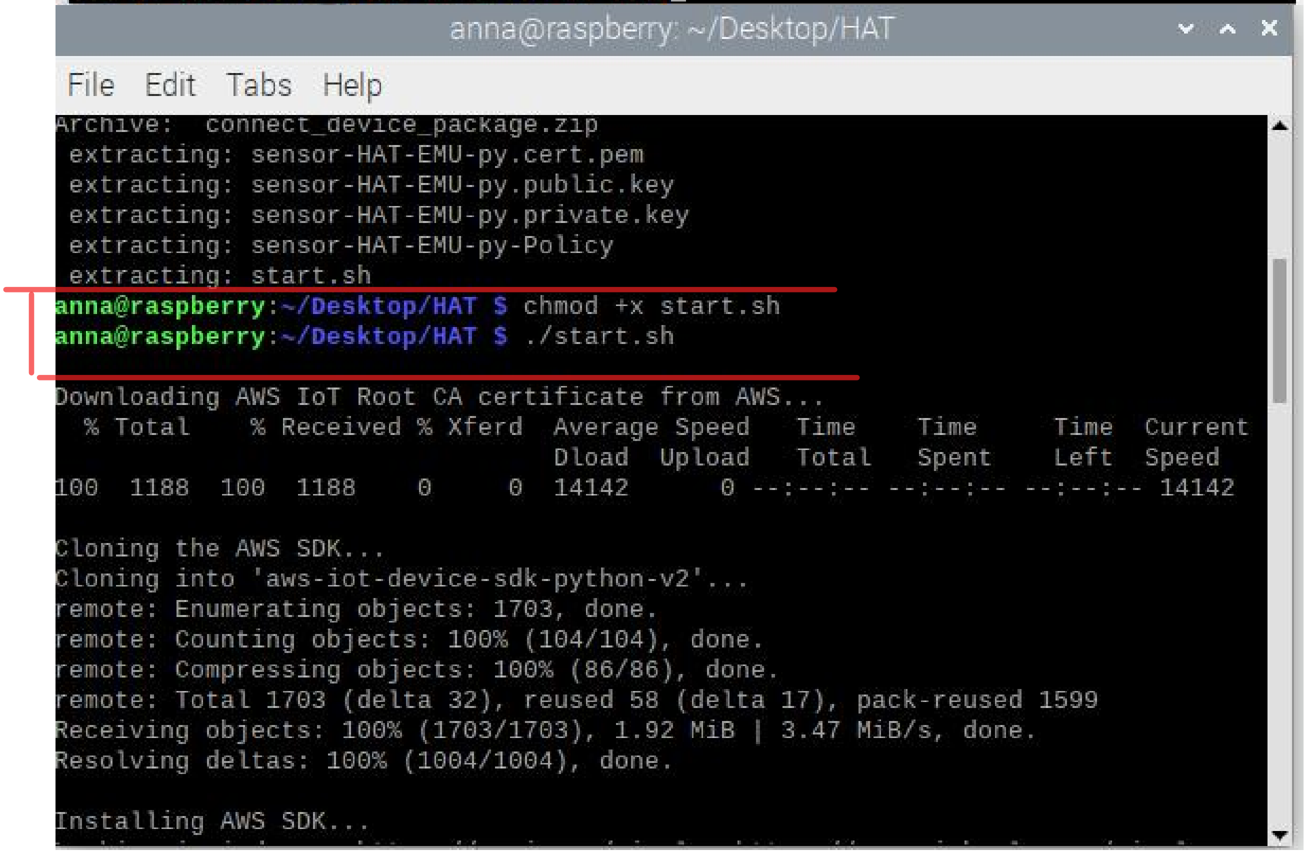
Created
November 23, 2022, 17:40:42 (UTC-0800)

Valid
November 23, 2022, 17:38:42 (UTC-0800)

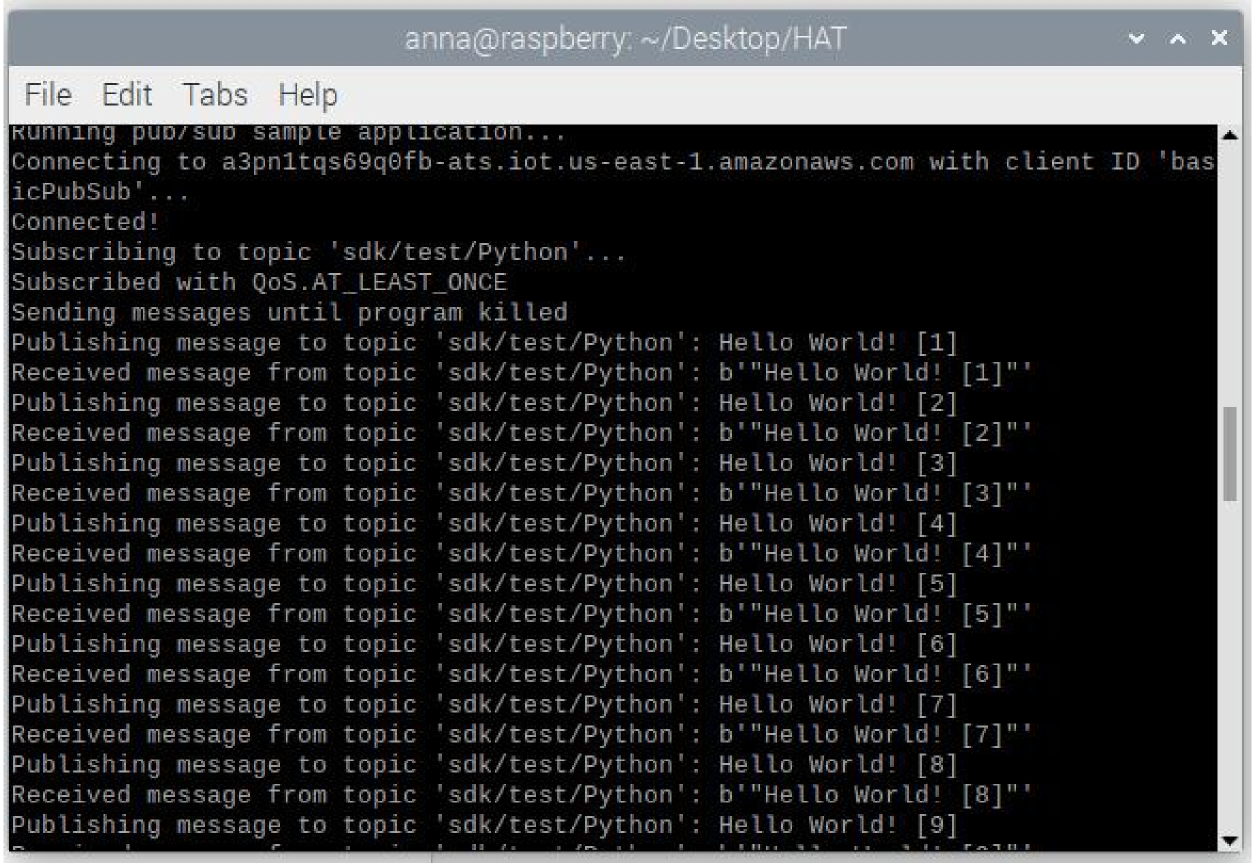
Expires
December 31, 2049, 15:59:59 (UTC-0800)



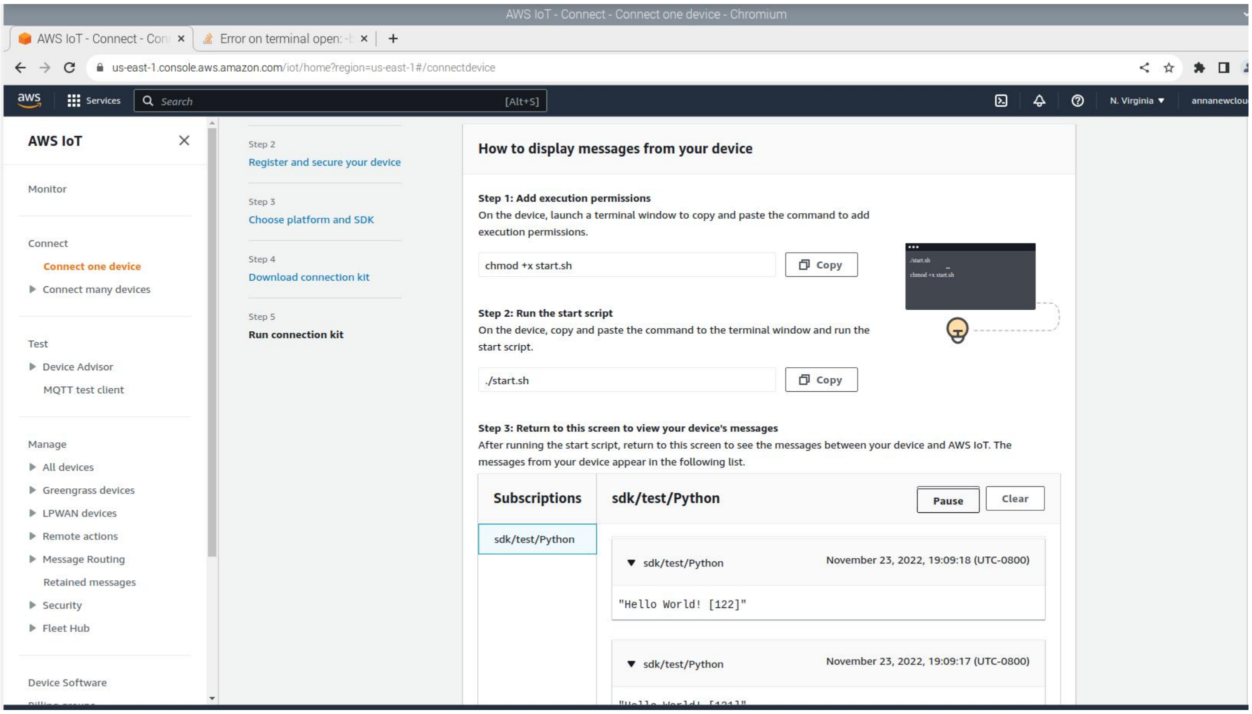
```
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  sse3-support
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libc-ares2 libjs-highlight.js libnode72 nodejs-doc sse2-support
Suggested packages:
  npm
The following NEW packages will be installed:
  libc-ares2 libjs-highlight.js libnode72 nodejs nodejs-doc sse2-support
0 upgraded, 6 newly installed, 0 to remove and 43 not upgraded.
Need to get 11.7 MB of archives.
After this operation, 49.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://deb.debian.org/debian bullseye/main i386 sse2-support i386 6 [8,544
 B]
Get:2 http://deb.debian.org/debian bullseye/main i386 libc-ares2 i386 1.17.1-1+d
eb11u1 [106 kB]
Get:3 http://deb.debian.org/debian bullseye/main i386 libjs-highlight.js all 9.1
8.5+dfsg1-1 [397 kB]
Get:4 http://deb.debian.org/debian-security bullseye-security/main i386 libnode7
2 i386 12.22.12~dfsg-1~deb11u1 [8,482 kB]
65% [4 libnode72 7,518 kB/8,482 kB 89%]
```



```
Archive:  connect_device_package.zip
 extracting: sensor-HAT-EMU-py.cert.pem
 extracting: sensor-HAT-EMU-py.public.key
 extracting: sensor-HAT-EMU-py.private.key
 extracting: sensor-HAT-EMU-py-Policy
 extracting: start.sh
anna@raspberry:~/Desktop/HAT $ chmod +x start.sh
anna@raspberry:~/Desktop/HAT $ ./start.sh

Downloading AWS IoT Root CA certificate from AWS...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1188  100  1188    0     0  14142      0 --:--:-- --:--:-- --:--:-- 14142

Cloning the AWS SDK...
Cloning into 'aws-iot-device-sdk-python-v2'...
remote: Enumerating objects: 1703, done.
remote: Counting objects: 100% (104/104), done.
remote: Compressing objects: 100% (86/86), done.
remote: Total 1703 (delta 32), reused 58 (delta 17), pack-reused 1599
Receiving objects: 100% (1703/1703), 1.92 MiB | 3.47 MiB/s, done.
Resolving deltas: 100% (1004/1004), done.

Installing AWS SDK...
```



```
Running pub/sub sample application...
Connecting to a3pn1tqs69q0fb-ats.iot.us-east-1.amazonaws.com with client ID 'bas
icPubSub'...
Connected!
Subscribing to topic 'sdk/test/Python'...
Subscribed with QoS.AT_LEAST_ONCE
Sending messages until program killed
Publishing message to topic 'sdk/test/Python': Hello World! [1]
Received message from topic 'sdk/test/Python': b'"Hello World! [1]"'
Publishing message to topic 'sdk/test/Python': Hello World! [2]
Received message from topic 'sdk/test/Python': b'"Hello World! [2]"'
Publishing message to topic 'sdk/test/Python': Hello World! [3]
Received message from topic 'sdk/test/Python': b'"Hello World! [3]"'
Publishing message to topic 'sdk/test/Python': Hello World! [4]
Received message from topic 'sdk/test/Python': b'"Hello World! [4]"'
Publishing message to topic 'sdk/test/Python': Hello World! [5]
Received message from topic 'sdk/test/Python': b'"Hello World! [5]"'
Publishing message to topic 'sdk/test/Python': Hello World! [6]
Received message from topic 'sdk/test/Python': b'"Hello World! [6]"'
Publishing message to topic 'sdk/test/Python': Hello World! [7]
Received message from topic 'sdk/test/Python': b'"Hello World! [7]"'
Publishing message to topic 'sdk/test/Python': Hello World! [8]
Received message from topic 'sdk/test/Python': b'"Hello World! [8]"'
Publishing message to topic 'sdk/test/Python': Hello World! [9]
```



How to display messages from your device

Step 1: Add execution permissions
On the device, launch a terminal window to copy and paste the command to add execution permissions.

`chmod +x start.sh` Copy

Step 2: Run the start script
On the device, copy and paste the command to the terminal window and run the start script.

`./start.sh` Copy

Step 3: Return to this screen to view your device's messages
After running the start script, return to this screen to see the messages between your device and AWS IoT. The messages from your device appear in the following list.

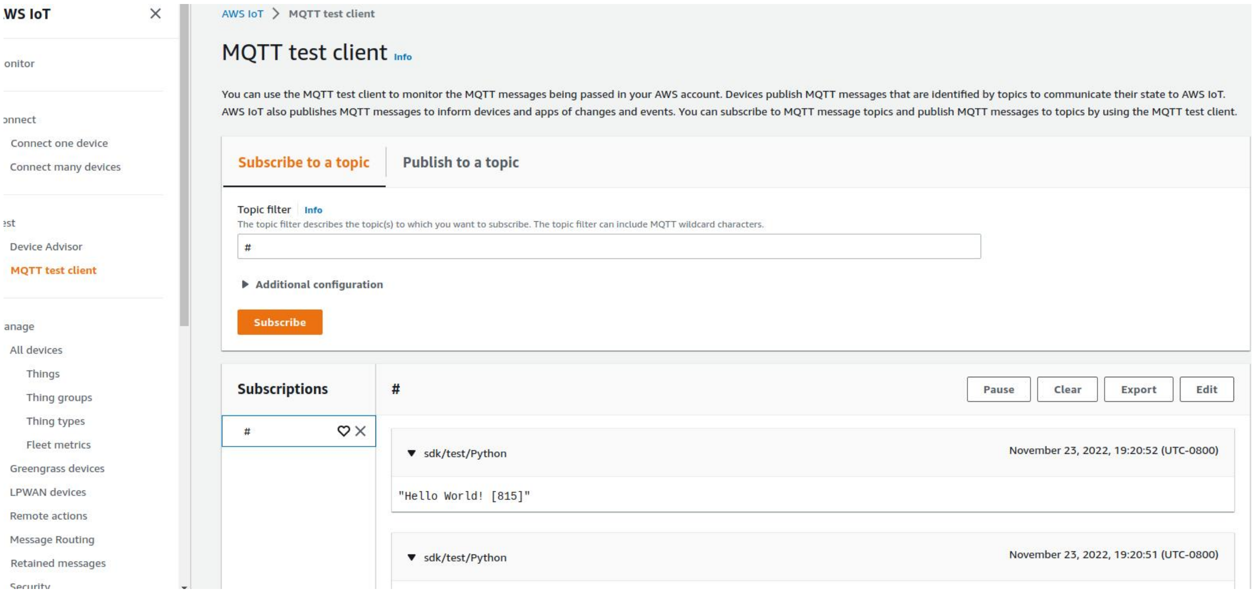
Subscriptions | **sdk/test/Python** | Pause | Clear

sdk/test/Python

▼ sdk/test/Python — November 23, 2022, 19:09:18 (UTC-0800)

```
"Hello World! [122]"
```

▼ sdk/test/Python — November 23, 2022, 19:09:17 (UTC-0800)



MQTT test client

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Topic filter

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

#

Subscriptions: #

sdk/test/Python   November 23, 2022, 19:20:52 (UTC-0800)

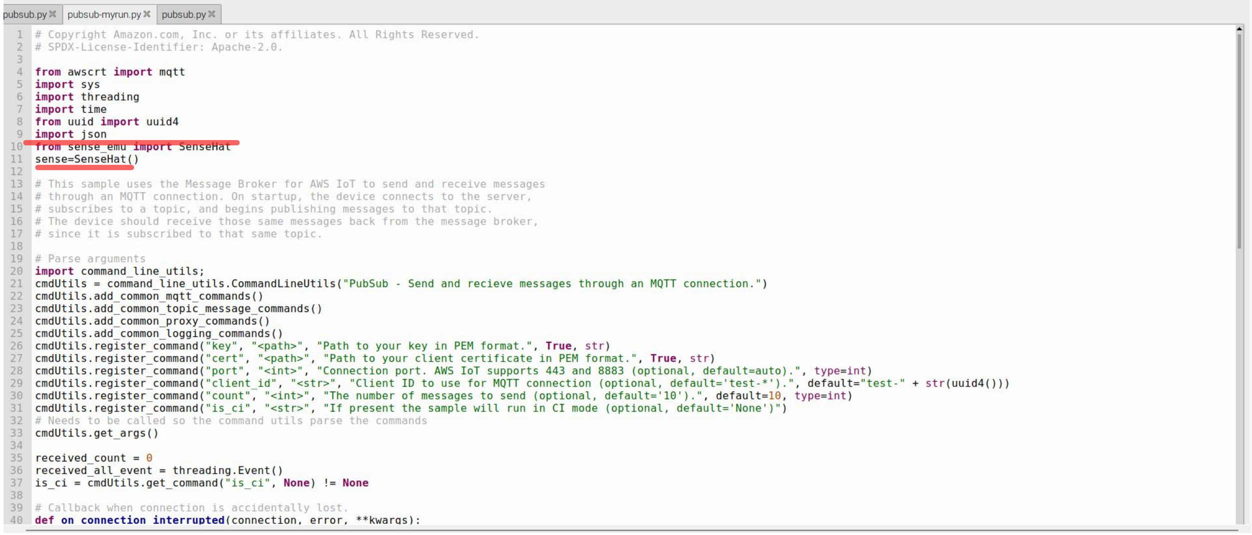
"Hello World! [815]"

sdk/test/Python   November 23, 2022, 19:20:51 (UTC-0800)

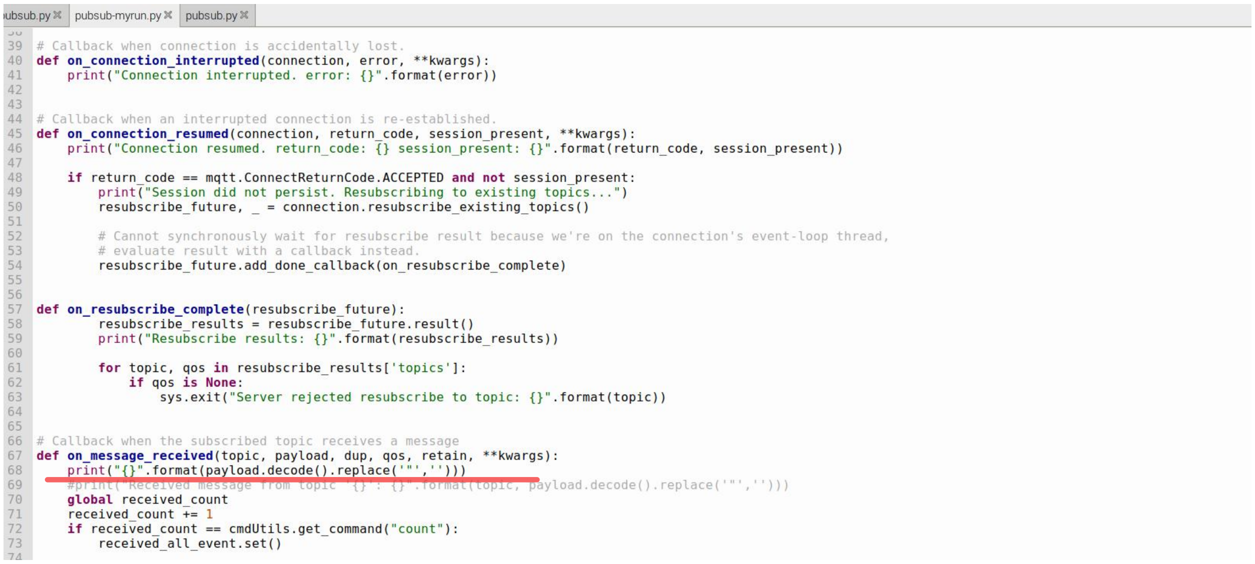
Step 3: Modify code:

Go to aws-iot-device-sdk-python-v2 -> samples -> pubsub.py

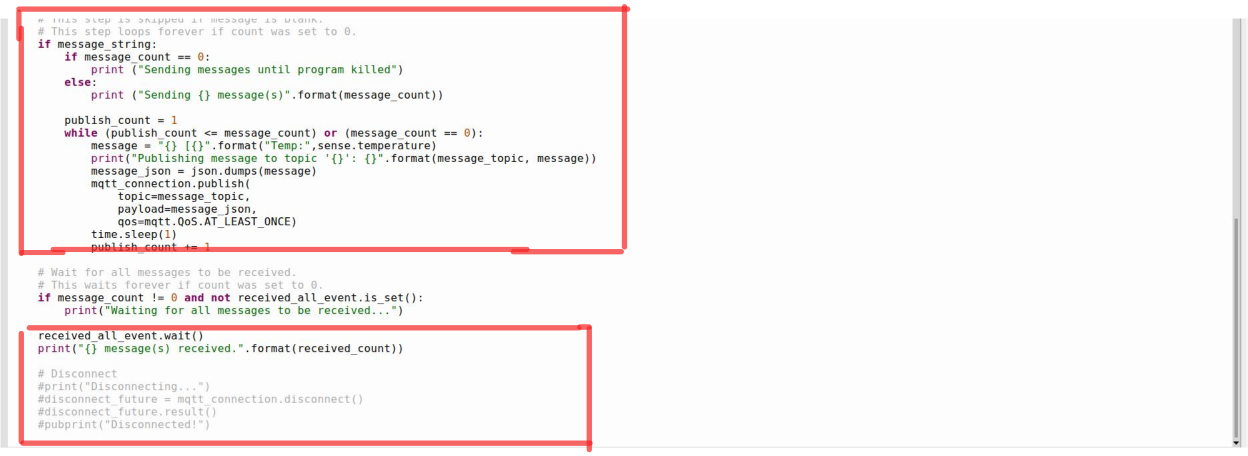
Remame the pubsub.py to pubsub-myrun.py



```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0.

from awscrt import mqtt
import sys
import threading
import time
from uuid import uuid4
import json
from sense_emu import SenseHat
sense=SenseHat()

# This sample uses the Message Broker for AWS IoT to send and receive messages
# through an MQTT connection. On startup, the device connects to the server,
# subscribes to a topic, and begins publishing messages to that topic.
# The device should receive those same messages back from the message broker,
# since it is subscribed to that same topic.

# Parse arguments
import command_line_utils;
cmdUtils = command_line_utils.CommandLineUtils("PubSub - Send and recieve messages through an MQTT connection.")
cmdUtils.add_common_mqtt_commands()
cmdUtils.add_common_topic_message_commands()
cmdUtils.add_common_proxy_commands()
cmdUtils.add_common_logging_commands()
cmdUtils.register_command("key", "<path>", "Path to your key in PEM format.", True, str)
cmdUtils.register_command("cert", "<path>", "Path to your client certificate in PEM format.", True, str)
cmdUtils.register_command("port", "<int>", "Connection port. AWS IoT supports 443 and 8883 (optional, default=auto).", type=int)
cmdUtils.register_command("client_id", "<str>", "Client ID to use for MQTT connection (optional, default='test-*').", default="test-" + str(uuid4()))
cmdUtils.register_command("count", "<int>", "The number of messages to send (optional, default='10').", default=10, type=int)
cmdUtils.register_command("is_ci", "<str>", "If present the sample will run in CI mode (optional, default='None')")
# Needs to be called so the command utils parse the commands
cmdUtils.get_args()

received_count = 0
received_all_event = threading.Event()
is_ci = cmdUtils.get_command("is_ci", None) != None

# Callback when connection is accidentally lost.
def on_connection_interrupted(connection, error, **kwargs):
```

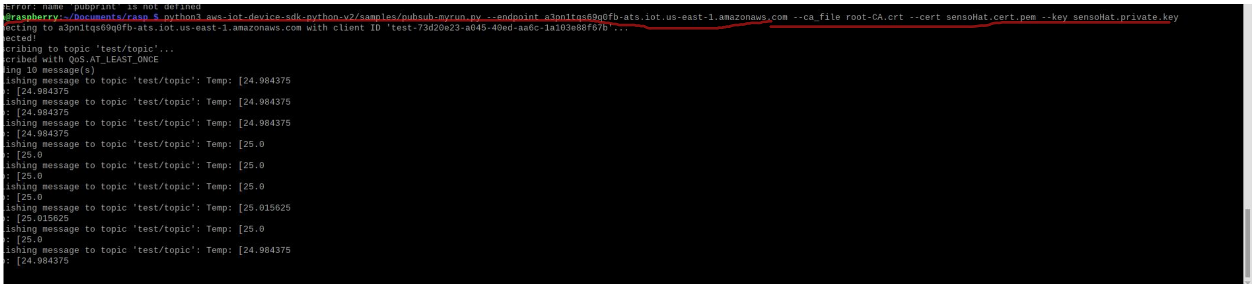


```
# Callback when connection is accidentally lost.
def on_connection_interrupted(connection, error, **kwargs):
    print("Connection interrupted. error: {}".format(error))


# Callback when an interrupted connection is re-established.
def on_connection_resumed(connection, return_code, session_present, **kwargs):
    print("Connection resumed. return_code: {} session_present: {}".format(return_code, session_present))

    if return_code == mqtt.ConnectReturnCode.ACCEPTED and not session_present:
        print("Session did not persist. Resubscribing to existing topics...")
        resubscribe_future, _ = connection.resubscribe_existing_topics()

        # Cannot synchronously wait for resubscribe result because we're on the connection's event-loop thread,
        # evaluate result with a callback instead.
        resubscribe_future.add_done_callback(on_resubscribe_complete)


def on_resubscribe_complete(resubscribe_future):
        resubscribe_results = resubscribe_future.result()
        print("Resubscribe results: {}".format(resubscribe_results))

        for topic, qos in resubscribe_results['topics']:
            if qos is None:
                sys.exit("Server rejected resubscribe to topic: {}".format(topic))


# Callback when the subscribed topic receives a message
def on_message_received(topic, payload, dup, qos, retain, **kwargs):
    print("{}".format(payload.decode().replace('"','')))
    #print("Received message from topic '{}': {}".format(topic, payload.decode().replace('"','')))
    global received_count
    received_count += 1
    if received_count == cmdUtils.get_command("count"):
        received_all_event.set()
```

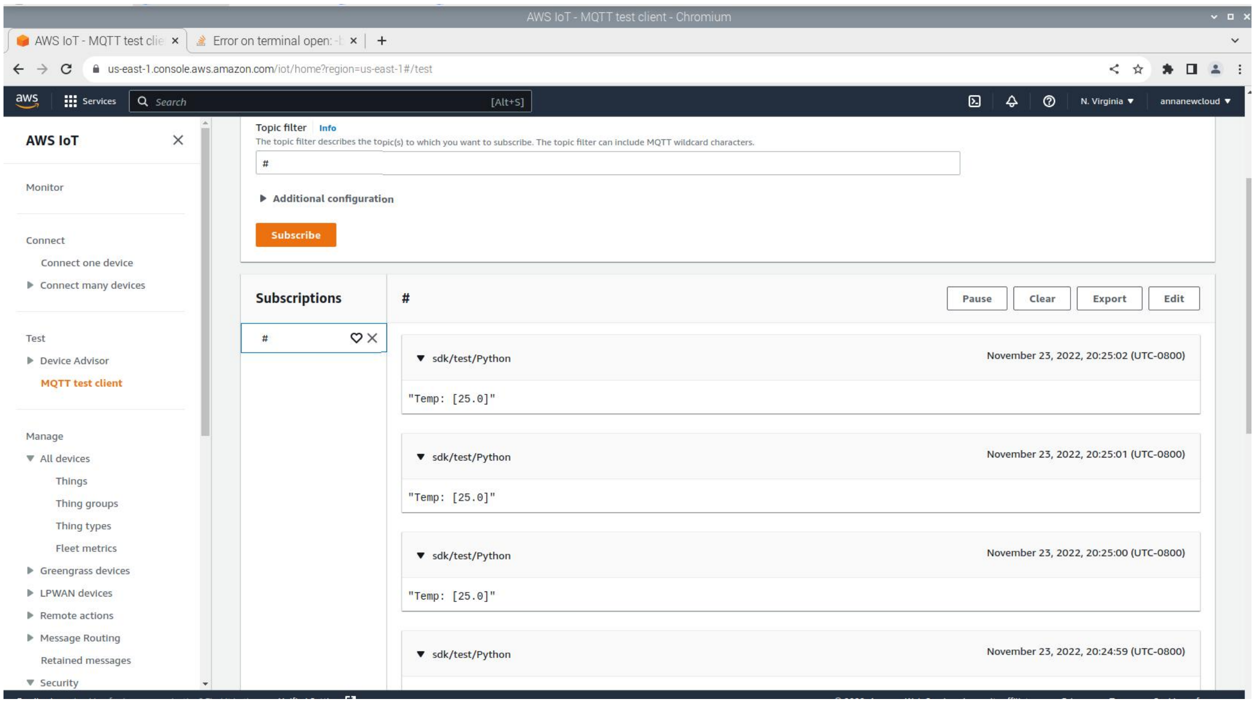


```
# This step is skipped if message is blank.
# This step loops forever if count was set to 0.
if message_string:
    if message_count == 0:
        print ("Sending messages until program killed")
    else:
        print ("Sending {} message(s)".format(message_count))

    publish_count = 1
    while (publish_count <= message_count) or (message_count == 0):
        message = "{} [{}".format("Temp:",sense.temperature)
        print("Publishing message to topic '{}': {}".format(message_topic, message))
        message_json = json.dumps(message)
        mqtt_connection.publish(
            topic=message_topic,
            payload=message_json,
            qos=mqtt.QoS.AT_LEAST_ONCE)
        time.sleep(1)
        publish_count += 1

# Wait for all messages to be received.
# This waits forever if count was set to 0.
if message_count != 0 and not received_all_event.is_set():
    print("Waiting for all messages to be received...")

received_all_event.wait()
print("{} message(s) received.".format(received_count))

# Disconnect
#print("Disconnecting...")
#disconnect_future = mqtt_connection.disconnect()
#disconnect_future.result()
#pubprint("Disconnected!")
```

Run the code and result:



```
eError: name 'pubprint' is not defined
a@raspberry:~/Documents/rasp $ python3 aws-iot-device-sdk-python-v2/samples/pubsub-myrun.py --endpoint a3pn1tqs69q0fb-ats.iot.us-east-1.amazonaws.com --ca_file root-CA.crt --cert sensoHat.cert.pem --key sensoHat.private.key
necting to a3pn1tqs69q0fb-ats.iot.us-east-1.amazonaws.com with client ID 'test-73d20e23-a045-40ed-aa6c-1a103e88f67b'...
nected!
scribing to topic 'test/topic'...
scribed with QoS.AT_LEAST_ONCE
ding 10 message(s)
lishing message to topic 'test/topic': Temp: [24.984375
p: [24.984375
lishing message to topic 'test/topic': Temp: [24.984375
p: [24.984375
lishing message to topic 'test/topic': Temp: [24.984375
p: [24.984375
lishing message to topic 'test/topic': Temp: [25.0
p: [25.0
lishing message to topic 'test/topic': Temp: [25.0
p: [25.0
lishing message to topic 'test/topic': Temp: [25.0
p: [25.0
lishing message to topic 'test/topic': Temp: [25.015625
p: [25.015625
lishing message to topic 'test/topic': Temp: [25.0
p: [25.0
lishing message to topic 'test/topic': Temp: [24.984375
p: [24.984375
```



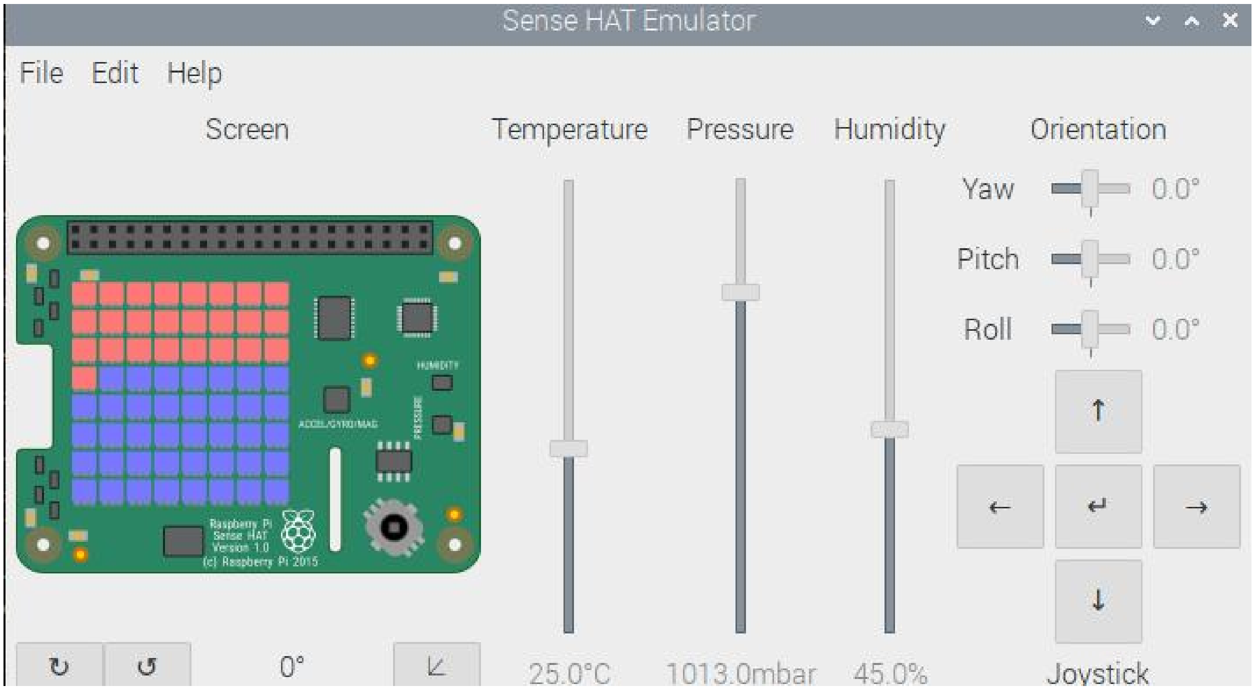
Topic filter

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

#

Subscriptions: #

```
sdk/test/Python                November 23, 2022, 20:25:02 (UTC-0800)
"Temp: [25.0]"

sdk/test/Python                November 23, 2022, 20:25:01 (UTC-0800)
"Temp: [25.0]"

sdk/test/Python                November 23, 2022, 20:25:00 (UTC-0800)
"Temp: [25.0]"

sdk/test/Python                November 23, 2022, 20:24:59 (UTC-0800)
```



Yaw 0.0°
Pitch 0.0°
Roll 0.0°

0°

25.0°C
1013.0mbar
45.0%