Arm Virtual Hardware Getting Started

Version 1.0

Knowledge Base Article

Non-Confidential

Copyright © 2022 Arm Limited (or its affiliates).
All rights reserved.

Issue 01
107660_0100_01_en

Arm Virtual Hardware Getting Started

Knowledge Base Article

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
0100-01	21 September 2022	Non-Confidential	First release

Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws

and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm’s trademark usage guidelines at https://www.arm.com/company/policies/trademarks.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Feedback

Arm® welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Contents

1. Quickstart....7
1.1 Quickstart for AVH - Raspberry Pi4....7
1.1.1 Creating your device....7
1.2 Quickstart for STM32U5 IoT Discovery Kit....11
1.2.1 Introduction....11
1.2.2 Creating your device....11
1.2.3 Running the example....15
1.2.4 Next steps....16
1.3 Quickstart for iMX8m Arm Cortex Complex....16
1.3.1 Introduction....16
1.3.2 Creating your device....17
1.3.3 Next steps....21
1.4 Quickstart for the API Docs....21

2. Supported Devices and Components....28
2.1 Supported Hardware Components for the Raspberry Pi 4....28
2.1.1 High-Level Summary....28
2.2 Supported Hardware Components for the STM32U5....29
2.2.1 High-Level Summary....29
2.2.2 Detailed List of Components....29
2.3 Supported Hardware Components for the i.M X8M....32
2.3.1 High-Level Summary....32
2.3.2 Detailed List of Components....33

3. Next Steps....37
3.1 Taking and restoring snapshots....37
3.1.1 What are snapshots?....37
3.1.2 Taking snapshots....37
3.2 Quick Connect....42
3.2.1 Note About Your Network Firewall Configuration....42
3.2.2 Step 1: Create a New Public-Private Key Pair....42
3.2.3 Step 2: Add Your SSH Public Key to Your Project....45
3.2.4 Step 3: Create Your Device....46

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3.2.5 Step 4: Use Quick Connect....46
3.2.6 Specifying the Private Key File for SSH Authentication (Optional)....48
3.3 Connecting to the VPN....49
3.3.1 Why Use the VPN?....49
3.3.2 How to Connect with Mac OS....49
3.3.3 How to Connect with Windows OS....53
3.4 Connect via SSH....58
3.5 Connect to the Console....60
3.5.1 Next Steps....62

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

1. Quickstart

Quickstart guides:

- Quickstart for AVH - Raspberry Pi4
- Quickstart for STM32U5 IoT Discovery Kit
- Quickstart for iMX8m Arm Cortex Complex
- Quickstart for the API Docs

1.1 Quickstart for AVH - Raspberry Pi4

This example shows you how to create a Raspberry Pi device along with some basic functionalities of the device's interface.

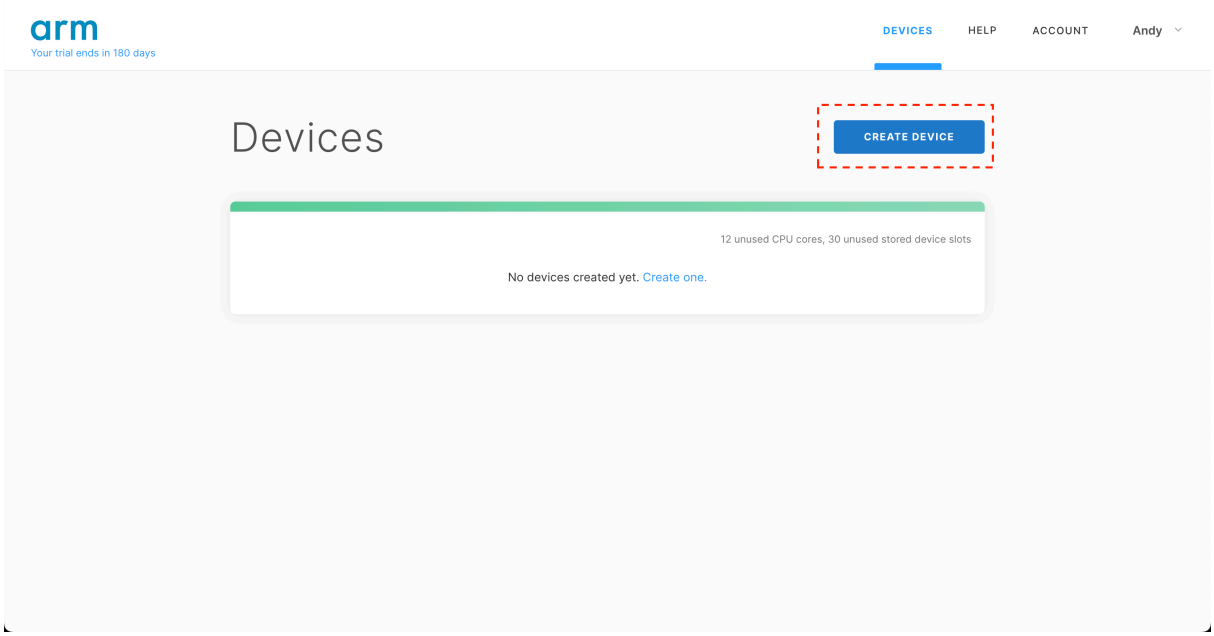
1.1.1 Creating your device

To create your device, do the following:

1. Log in to AVH with your Arm user account at https://app.avh.arm.com/.

   If you do not have an Arm account register at Arm Developer.
2. When you reach the AVH Devices page, click CREATE DEVICE.

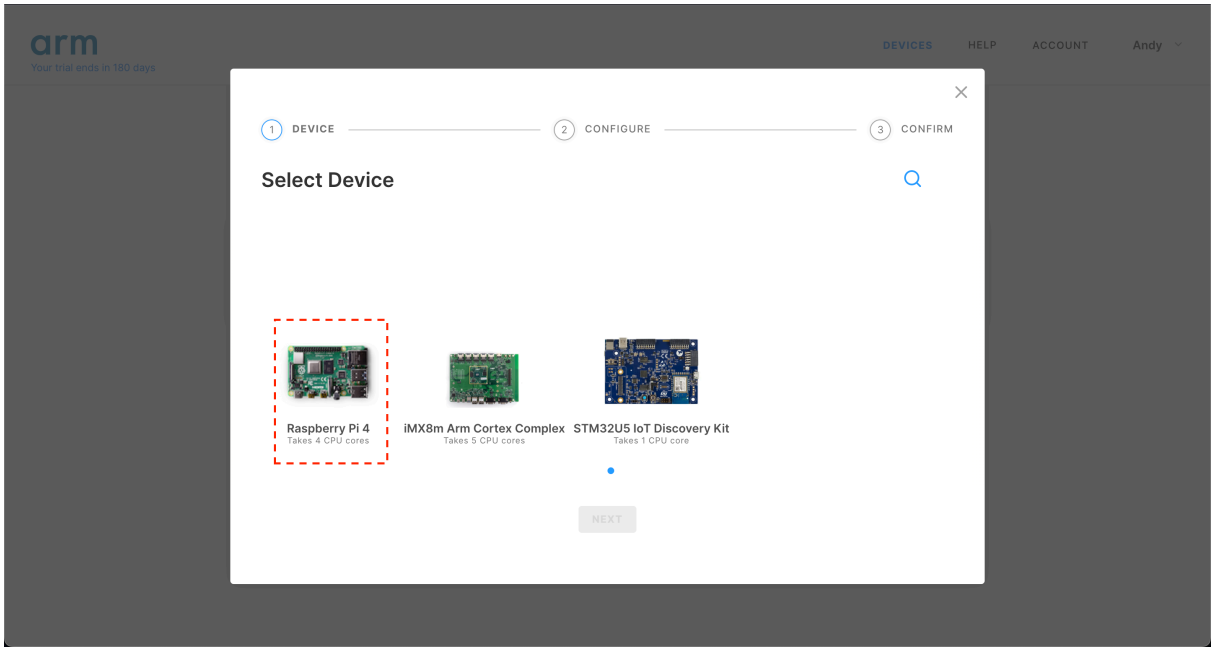
Figure 1-1: Create device



Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

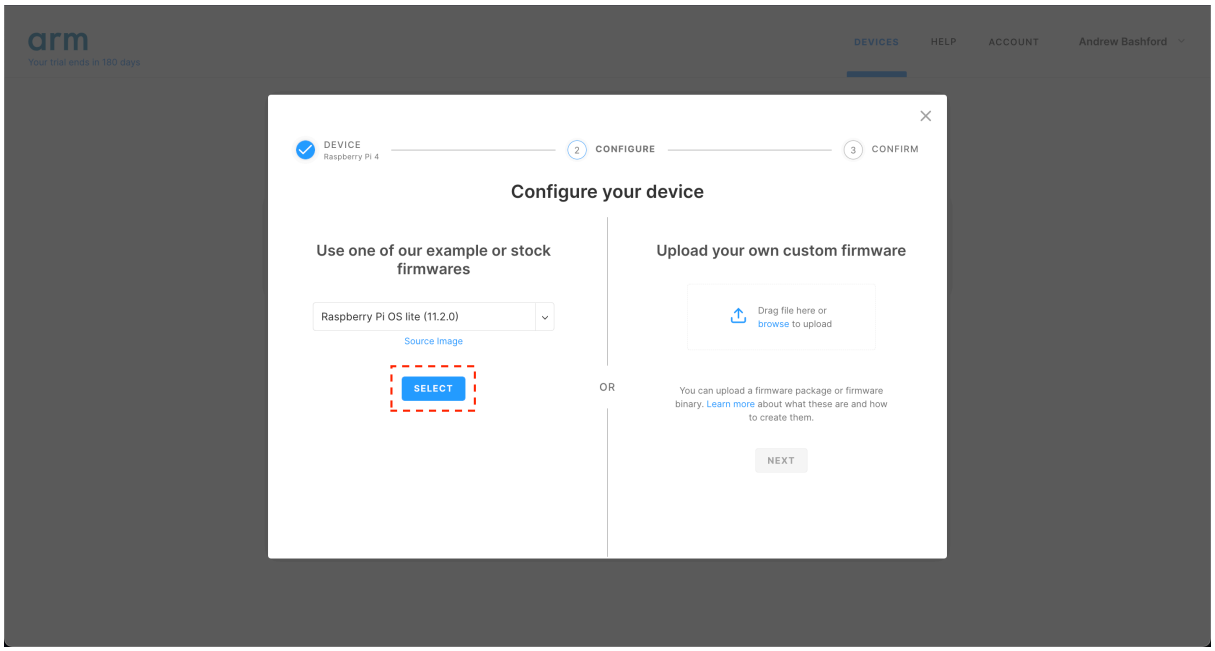
3. Select the Raspberry Pi 4 device, then click NEXT.

Figure 1-2: Select Raspberry Pi 4



4. On the Configure Your Device page, you will see a dropdown with stock firmware options. You can upload your own custom firmware. For this example, choose Raspberry Pi OS lite (11.2.0) and click SELECT.

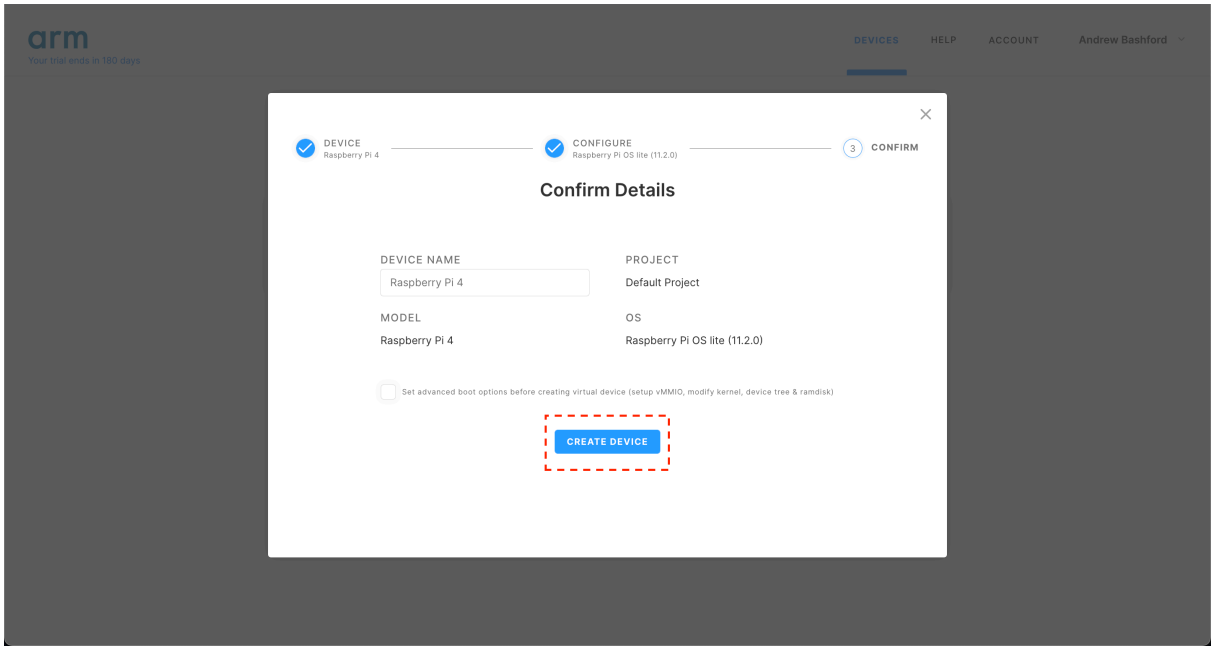
Figure 1-3: Configure your device



Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

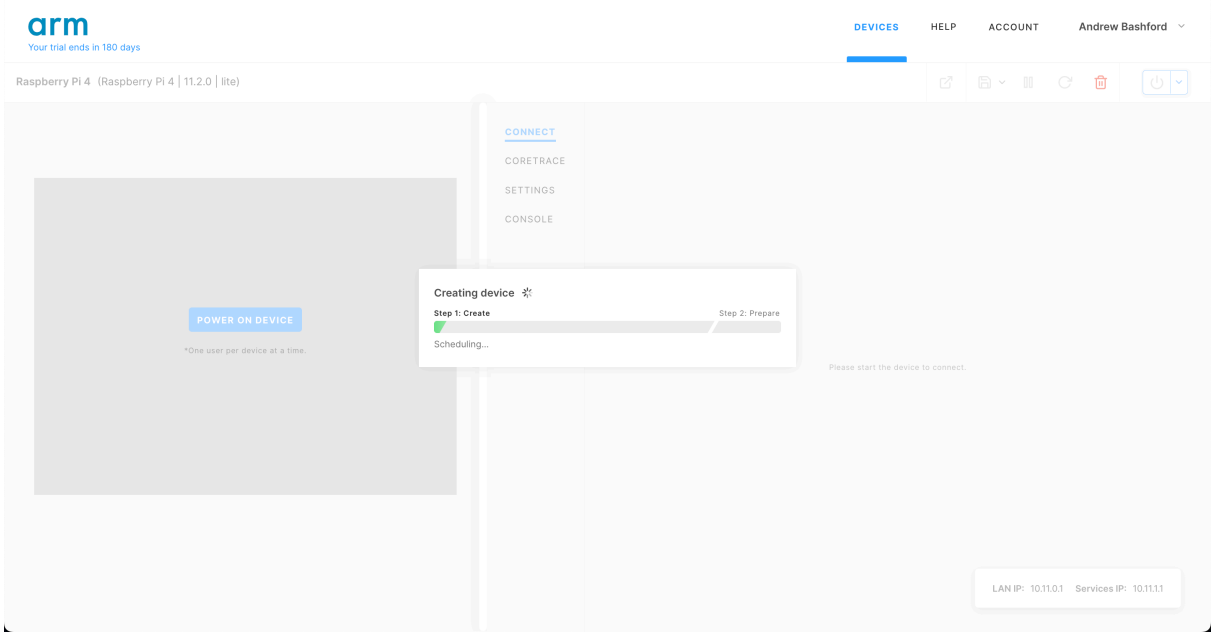
5. On the Confirm Details page, verify your selections then click CREATE DEVICE. Leave the device name as the default and do not set advanced boot options.

Figure 1-4: Confirm details



6. Please be patient while the virtualized device loads. It may take a few minutes.

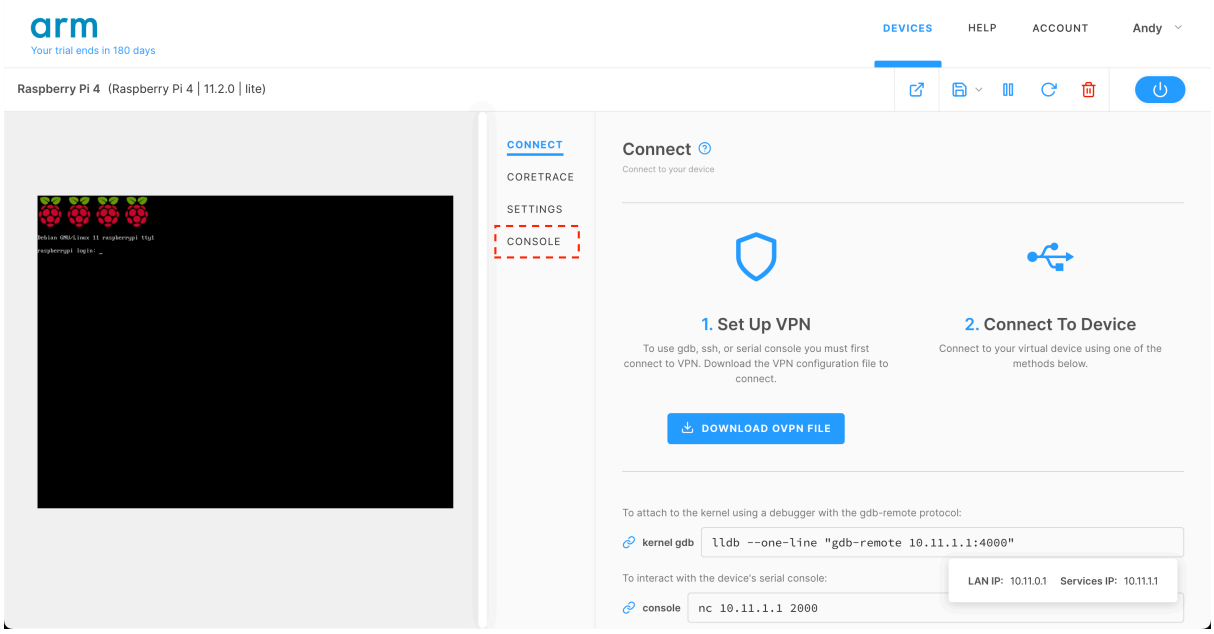
Figure 1-5: Virtualized device loading



7. Your Raspberry Pi device is now ready to go! Next, let's log in. Click CONSOLE in the devices menu.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

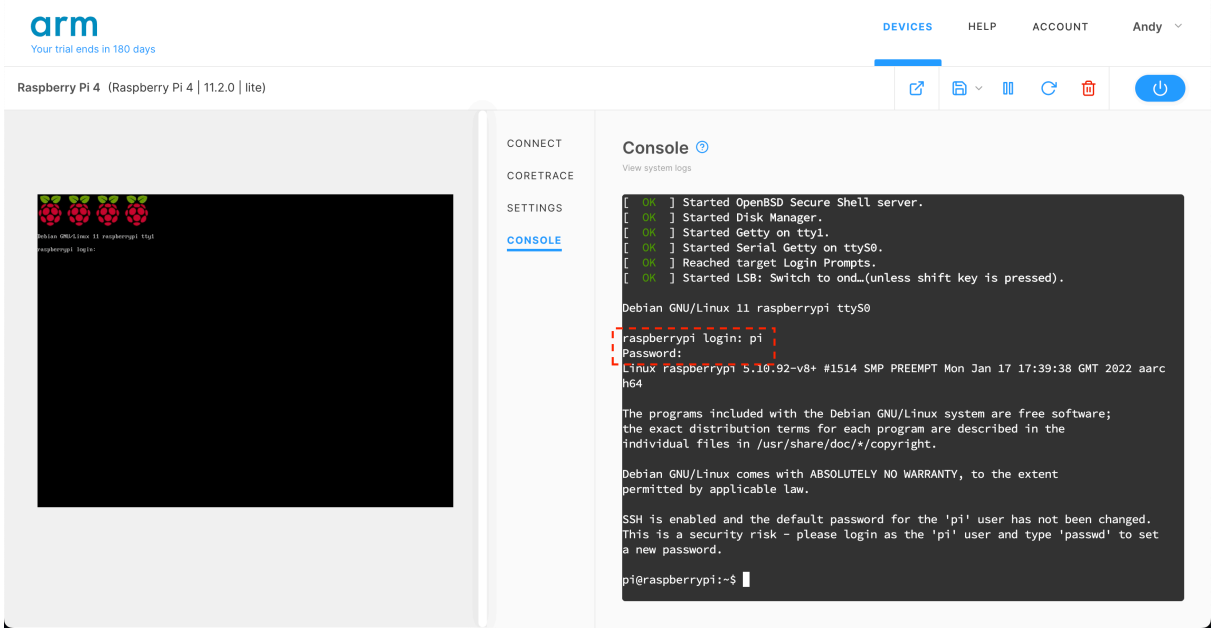
Figure 1-6: Log in



8. Type in the default username: `pi` and password: `raspberry`, then hit `return`. When you see the display name change to `pi@rasberrypi`, you are logged in.

   Alternatively, you could use the graphical console on the right to log in.

Figure 1-7: Console



9. Your virtualized Raspberry Pi 4 device is now ready to use!

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

You can test out your new Raspberry Pi 4 device by creating and restoring Snapshots in the next tutorial, Taking and restoring snapshots.

1.2 Quickstart for STM32U5 IoT Discovery Kit

Set up a STM32U5 IoT Discovery Kit device and run the `IOT_HTTP_WebServer` example.

1.2.1 Introduction

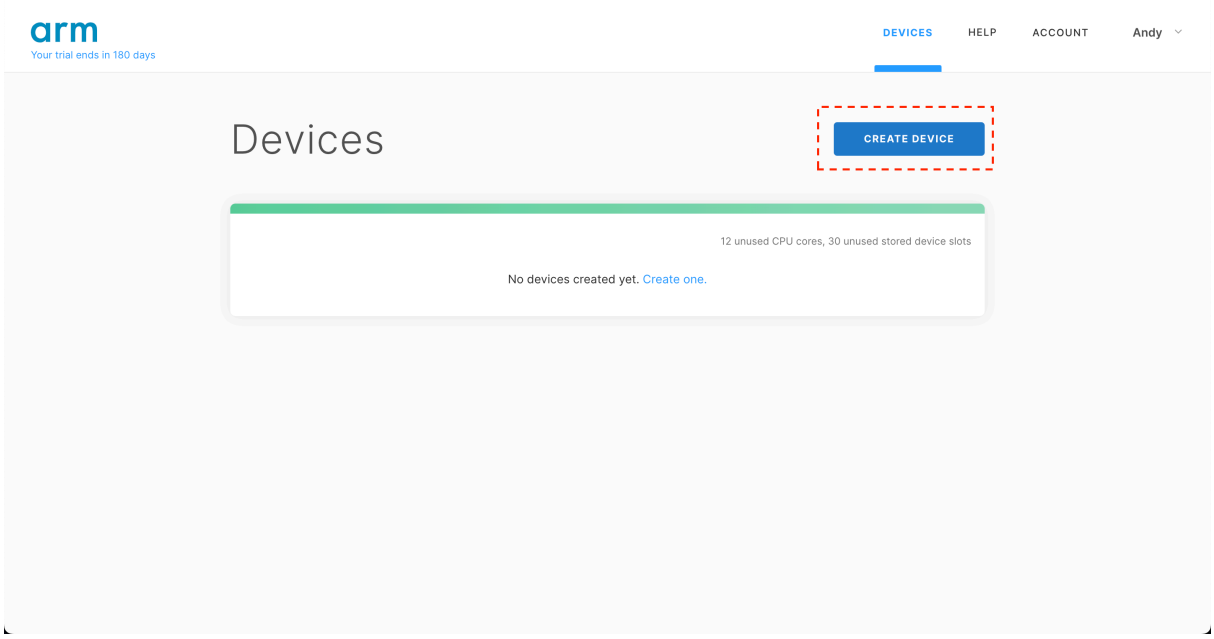
This example demonstrates how to create a STM32U5 IoT Discovery Kit device with an example firmware. In this example kernel component in the archive is an `IOT_HTTP_WebServer.elf` firmware binary produced by STMCubeIDE. Find out more at the ST website.

1.2.2 Creating your device

To create your device, do the following:

1. Log in to AVH with your Arm user account. https://avh.arm.com/ If you do not have create an Arm account by registering at https://developer.arm.com/
2. When logged in you land on the AVH Devices page. Click CREATE DEVICE.

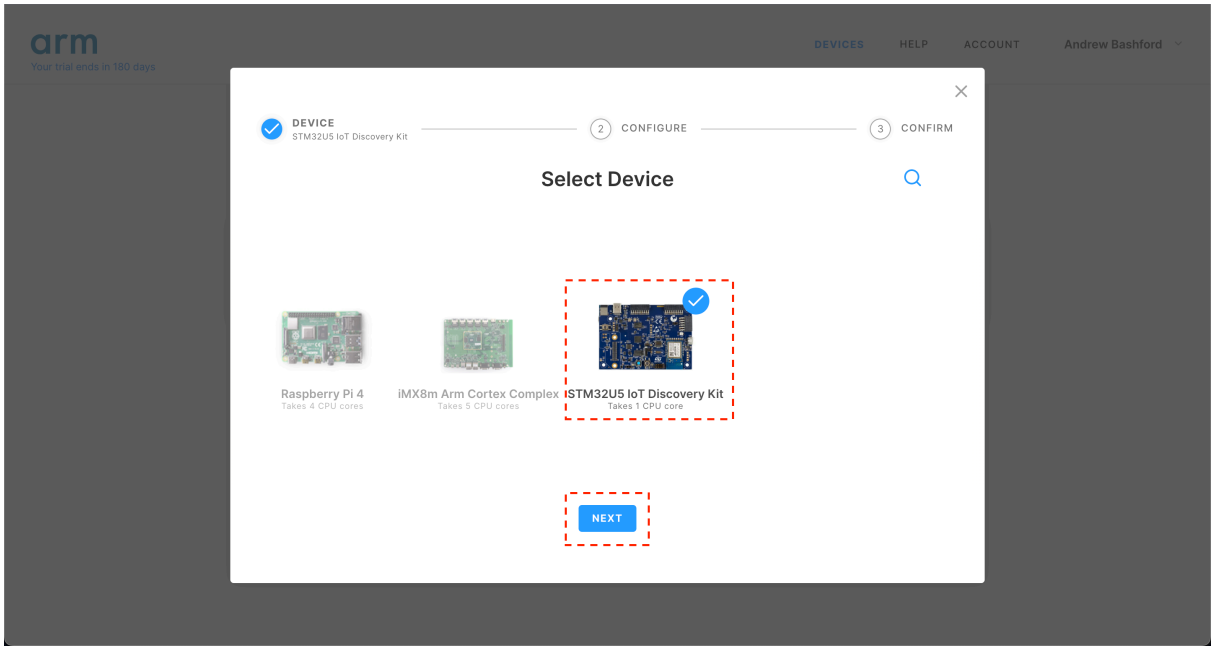
Figure 1-8: Create Device



3. Click on the STM32U5 IoT Discovery Kit to select it. Then click NEXT.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-9: Select STM32U5 IoT Discovery Kit



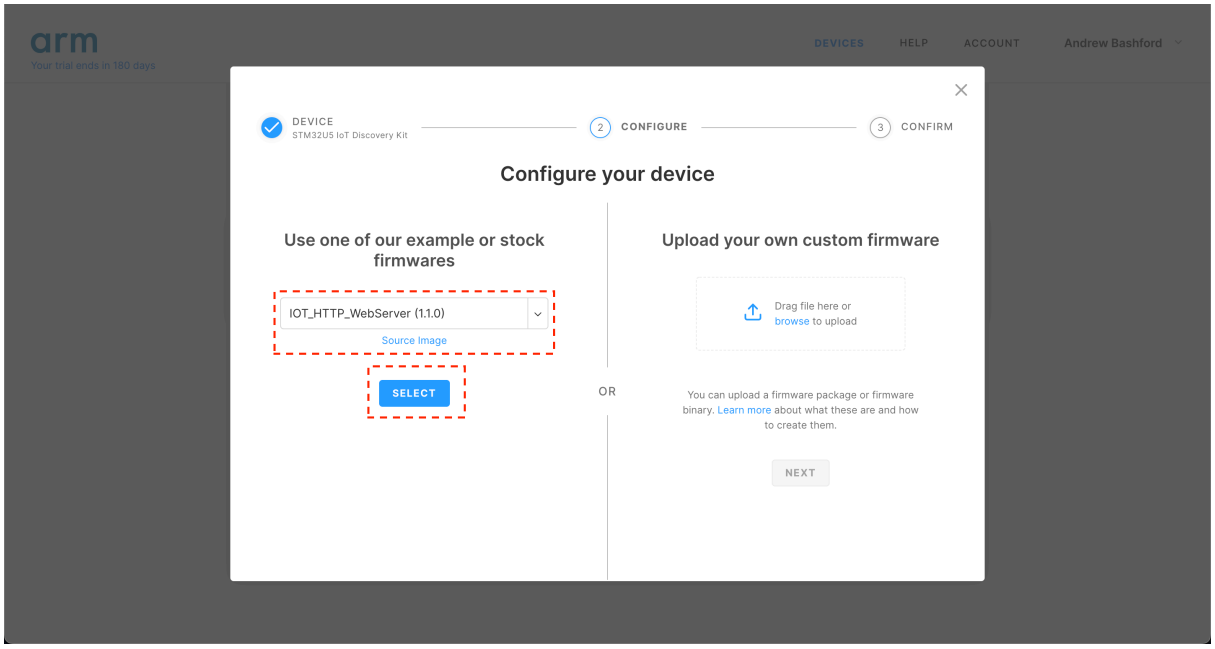
Note

You require 1 free CPU core to build this device or it will be disabled.

4. In this next step, you will see a list of pre-configured examples/stock firmware or you can upload your own custom firmware. In this example, we will use the selection `IOT_HTTP_WebServer (1.1.0)` in the stock firmware section. Click SELECT.

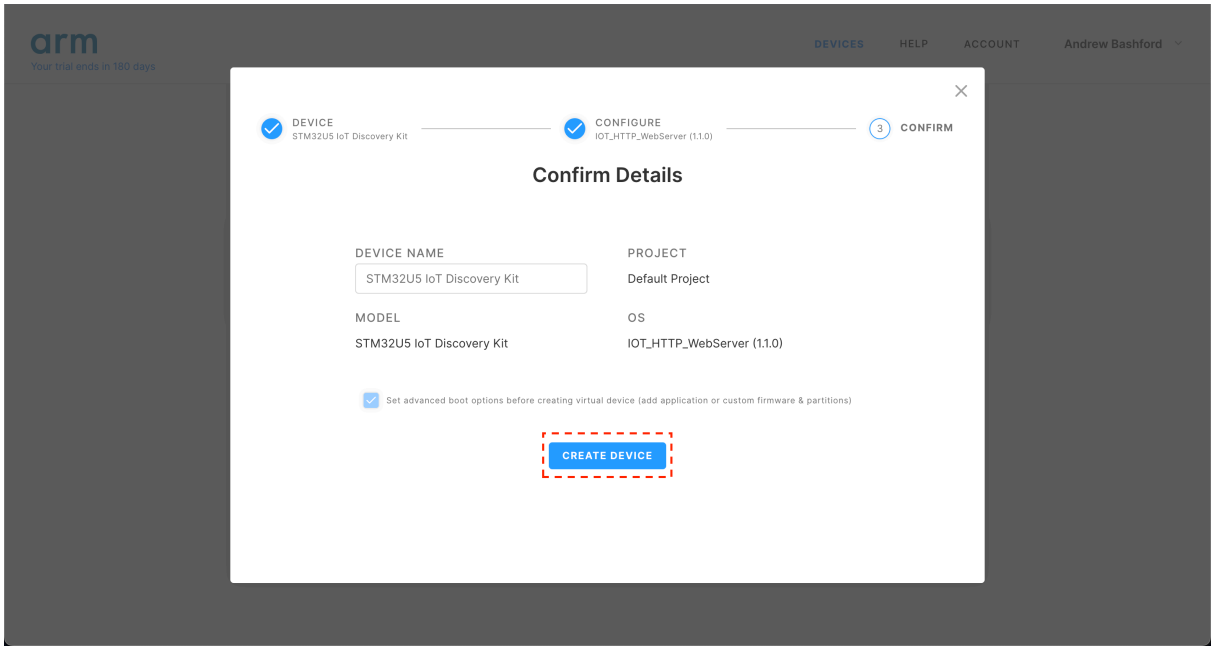
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-10: See a list of pre-configured examples/stock



5. In the confirmation screen, click CREATE DEVICE using the default name and the advance boot options will be checked by default.

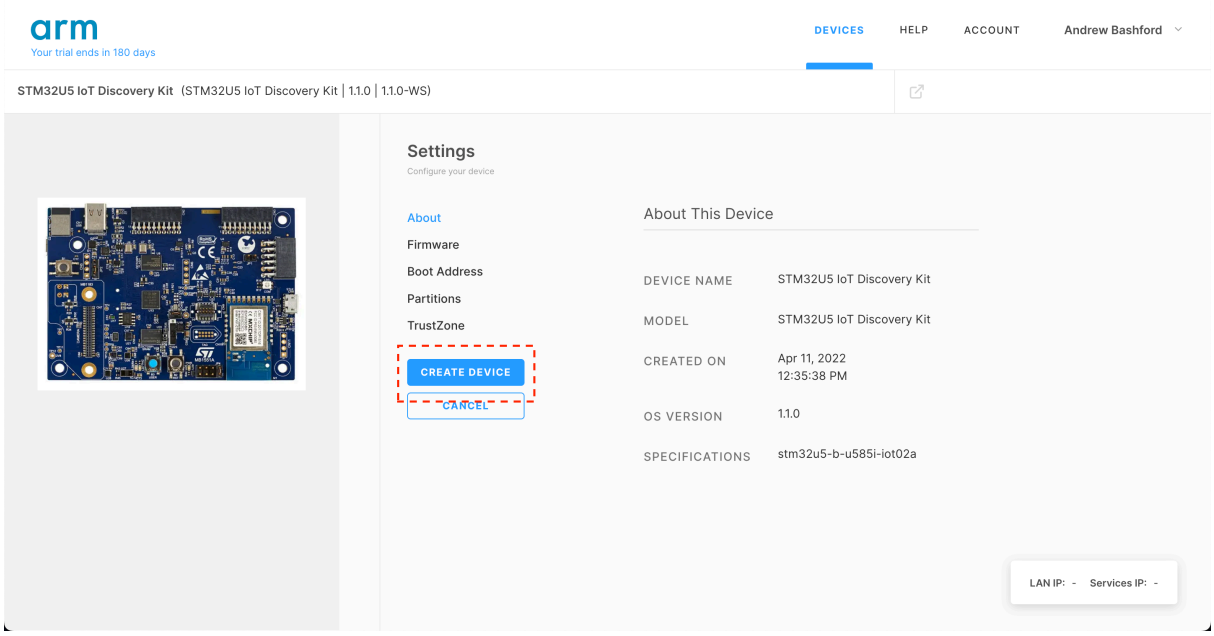
Figure 1-11: Confirmation screen



6. In the advanced boot options, we will just use the default. Click CREATE DEVICE.

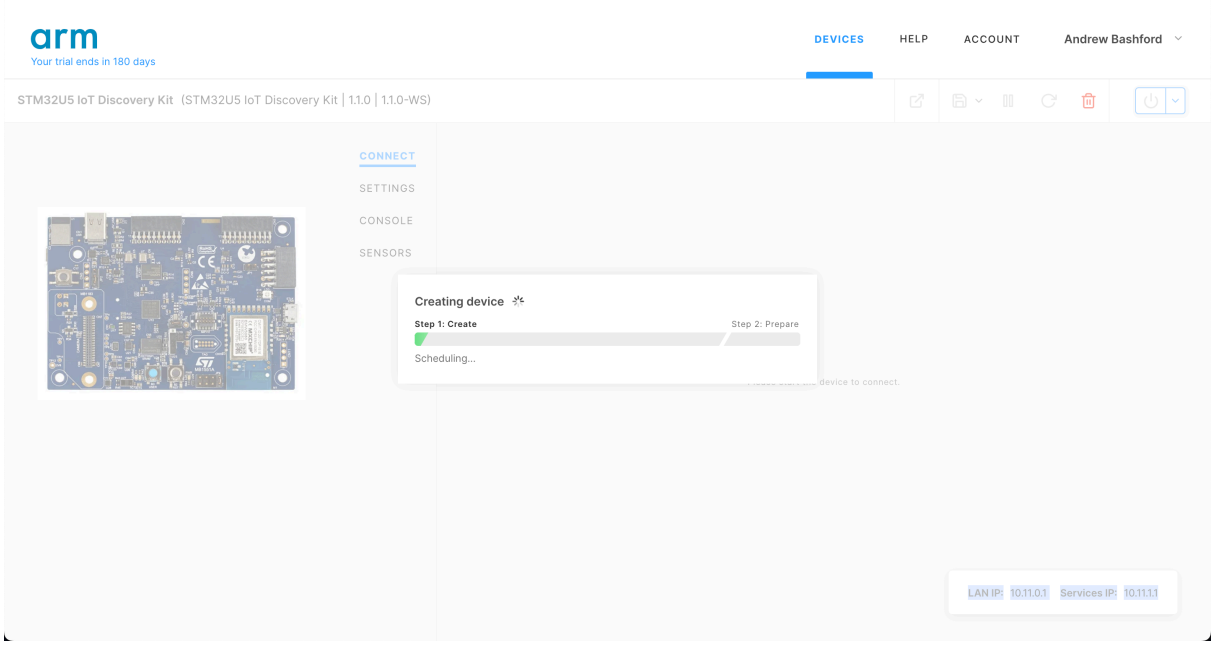
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-12: Advanced boot options



7. The device will then be created. It takes a few moments.

Figure 1-13: Creating devices



8. Your STM32U5 IoT Discovery Kit is now ready to go!

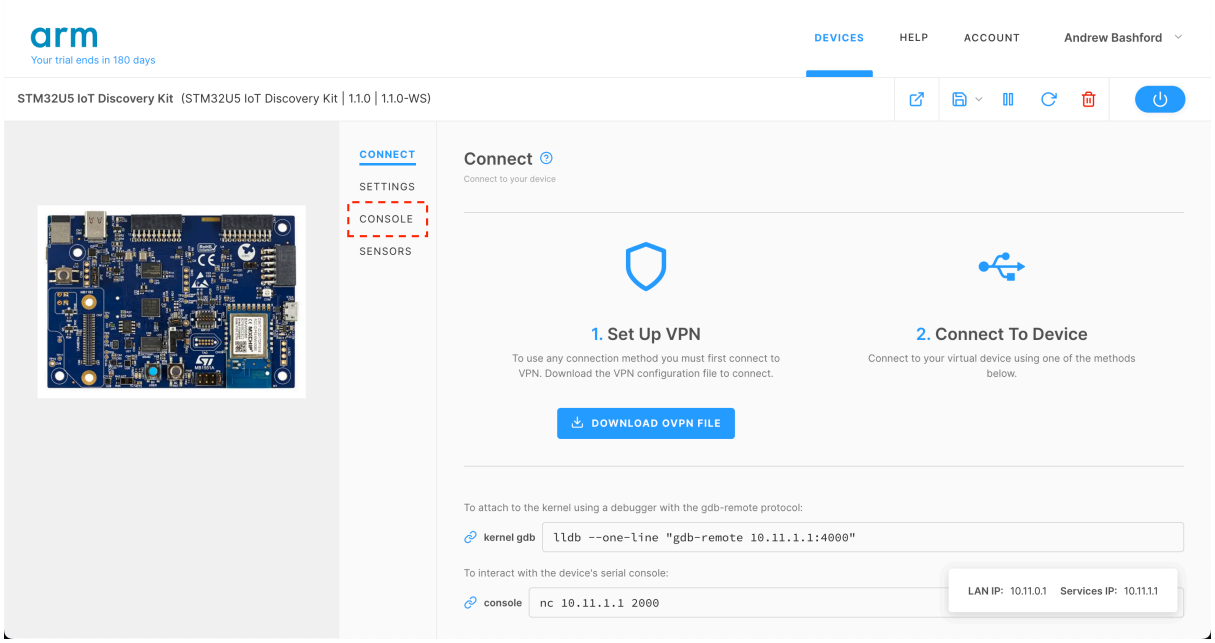
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

1.2.3 Running the example

To run the example, do the following:

1. Click CONSOLE on the devices menu.

Figure 1-14: Running the example



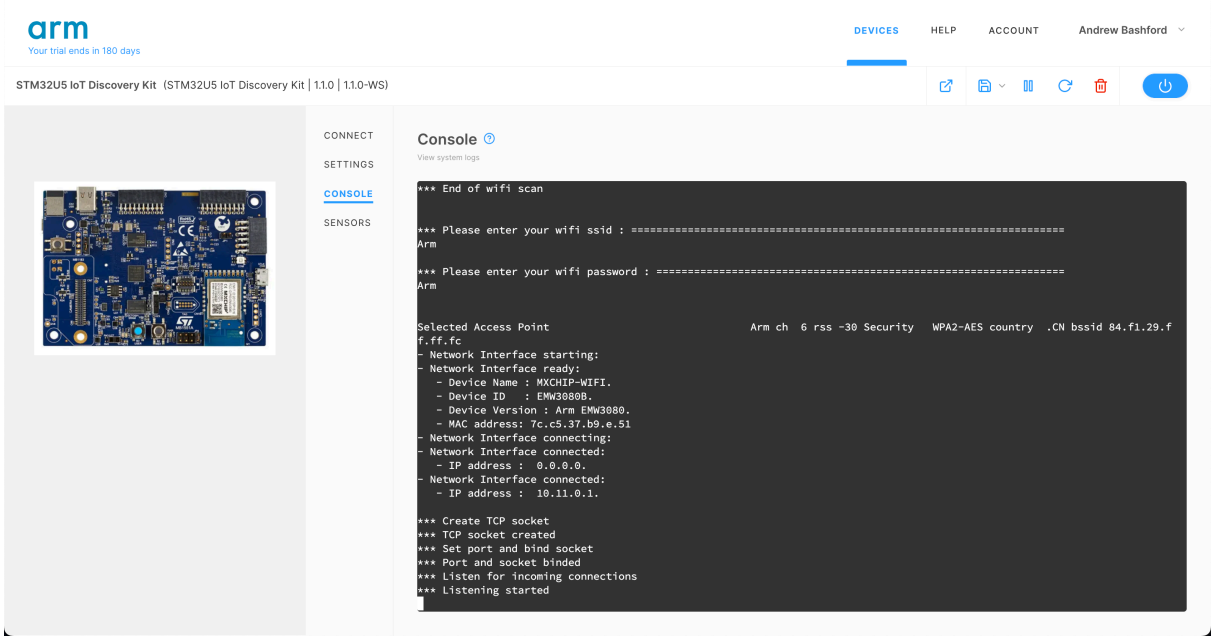
2. Type SSID from the list (`Arm` in this case) and anything as a password (`Arm` in this case).

Note

The SSID and password are case-sensitive.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-15: Type SSID



3. Now the STM32U5 IoT Discovery Kit virtual device is running a sensor web-server available through VPN or other virtual devices (like Raspberry Pi). It is possible to set sensor values using API or from the sensor tab.

1.2.4 Next steps

Further reading on the STM board page.

Other STM32U5 IoT Discovery Kit examples:

- STM32CubeU5 WiFiBasics example
- STM32CubeU5 TFM example

1.3 Quickstart for iMX8m Arm Cortex Complex

Set up an iMX8m Arm Cortex Complex device and log in to the console.

1.3.1 Introduction

This example demonstrates how to create an iMX8m Arm Cortex Complex device and Log in to the Console.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

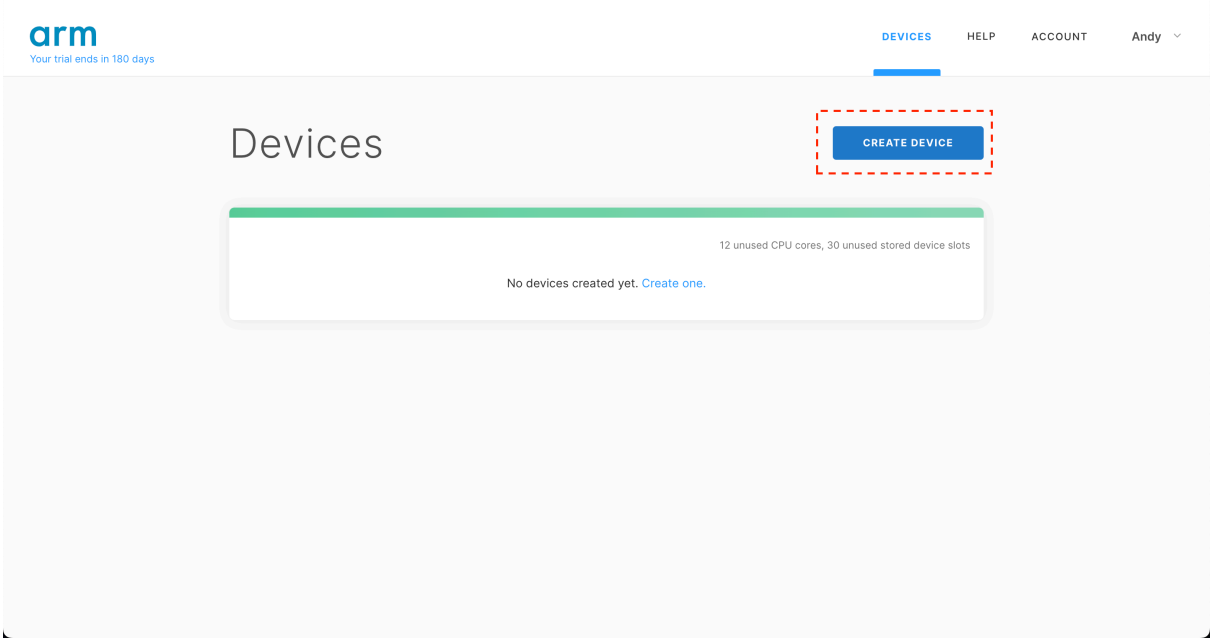
1.3.2 Creating your device

To create your device, do the following:

1. Log in to AVH with your Arm user account. https://avh.arm.com/

   If you do not have or need to create an Arm account register at https://developer.arm.com/
2. When logged in you land on the AVH Devices page. Click CREATE DEVICE.

   Figure 1-16: AVH Devices page



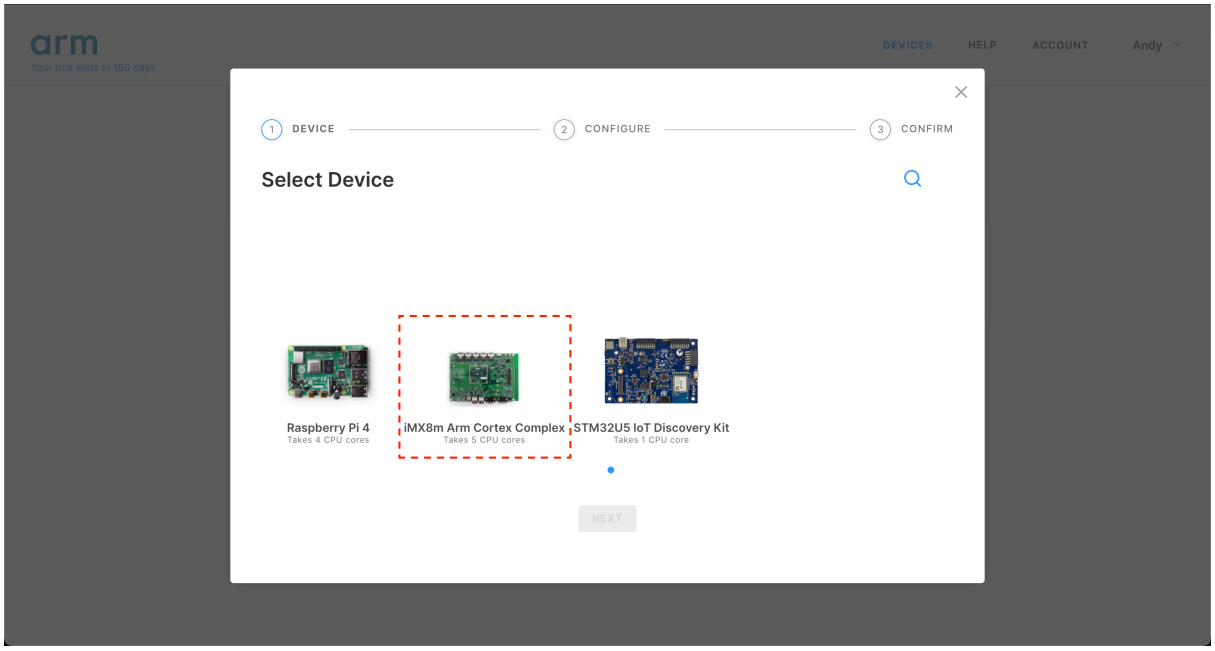
3. Click on the iMX8m Arm Cortex Complex to select it, then click NEXT.

Note

You require 5 free CPU cores to build this device or it will be disabled.

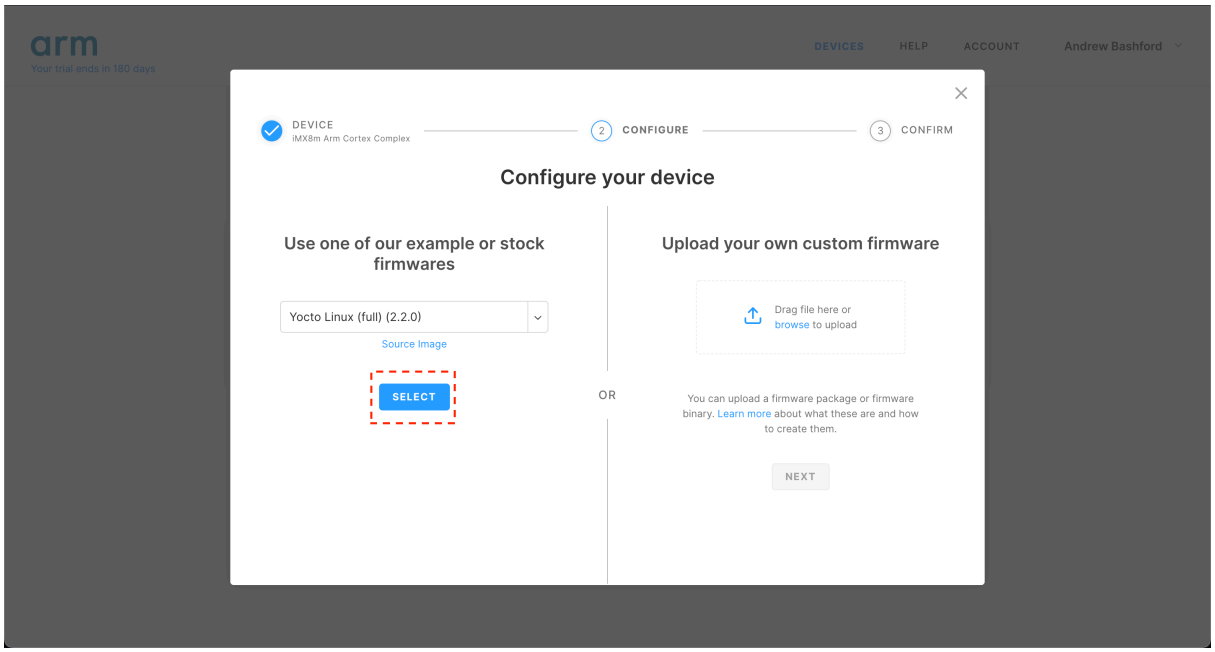
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-17: Select iMX8m Arm Cortex Complex



4. In this next step, you will see a list of pre-configured examples/stock firmware or you can upload your own custom firmware. In this example, we will use the default selection `Yocto Linux (2.2.0)` in the stock firmware section. Click SELECT.

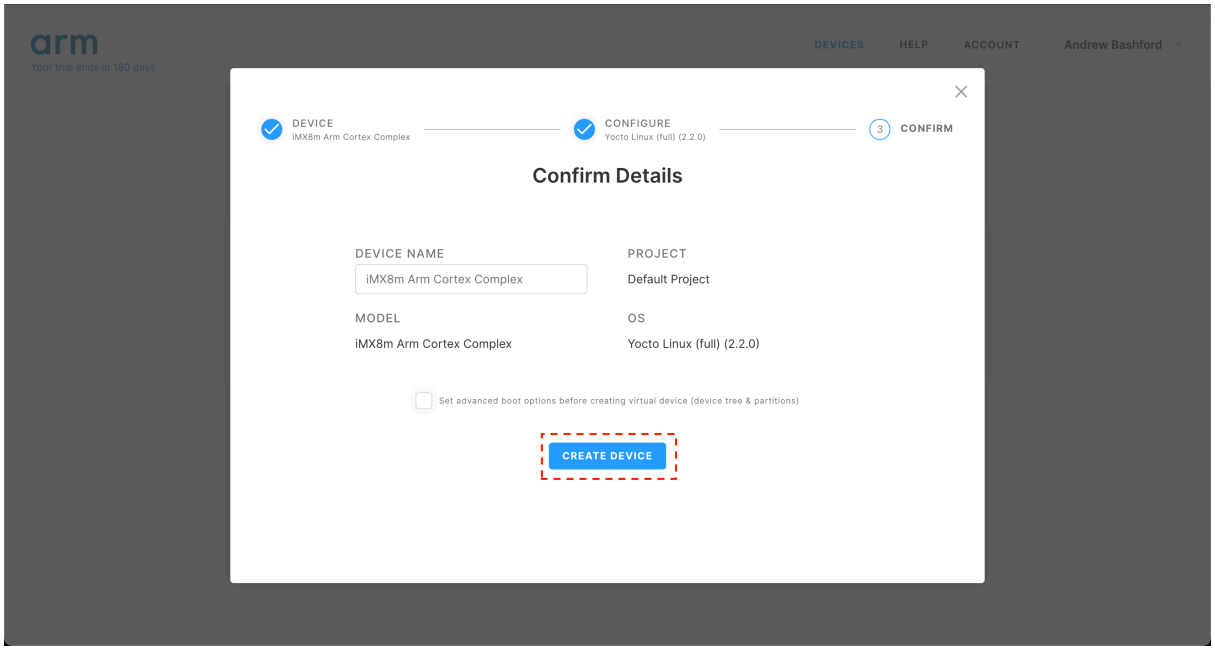
Figure 1-18: Yocto Linux (2.2.0)



5. In the confirmation screen, click CREATE DEVICE using the default name and the advance boot options unchecked.

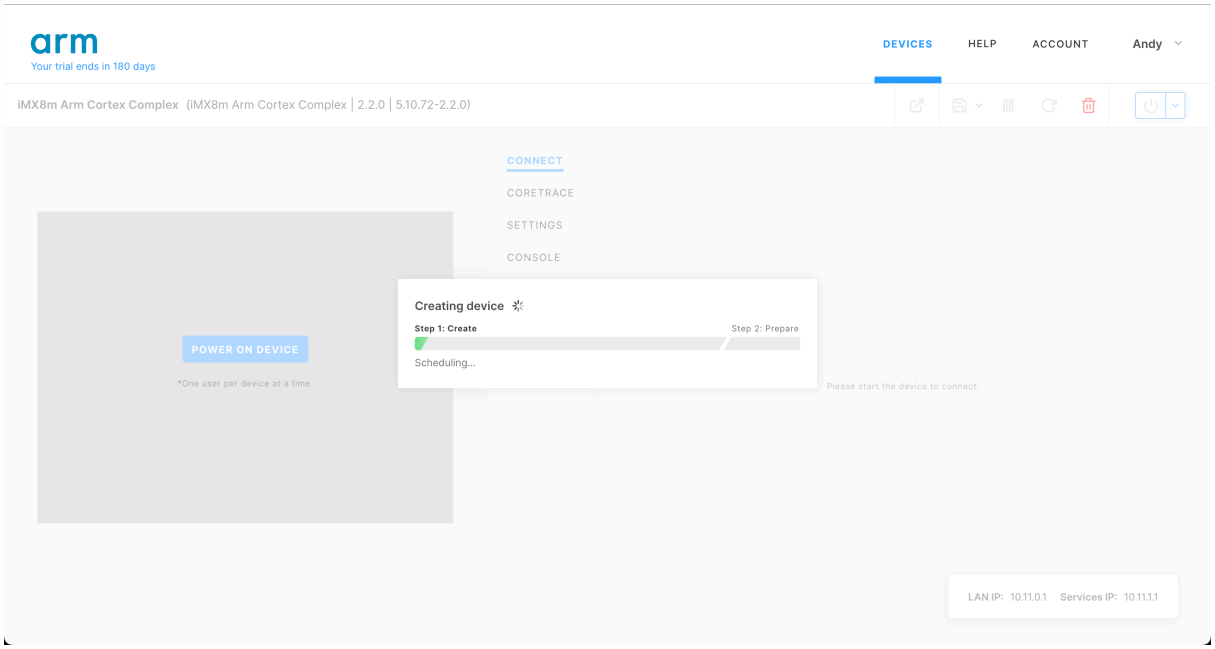
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-19: Confirmation screen



6. The device will then be created. It takes a few moments.

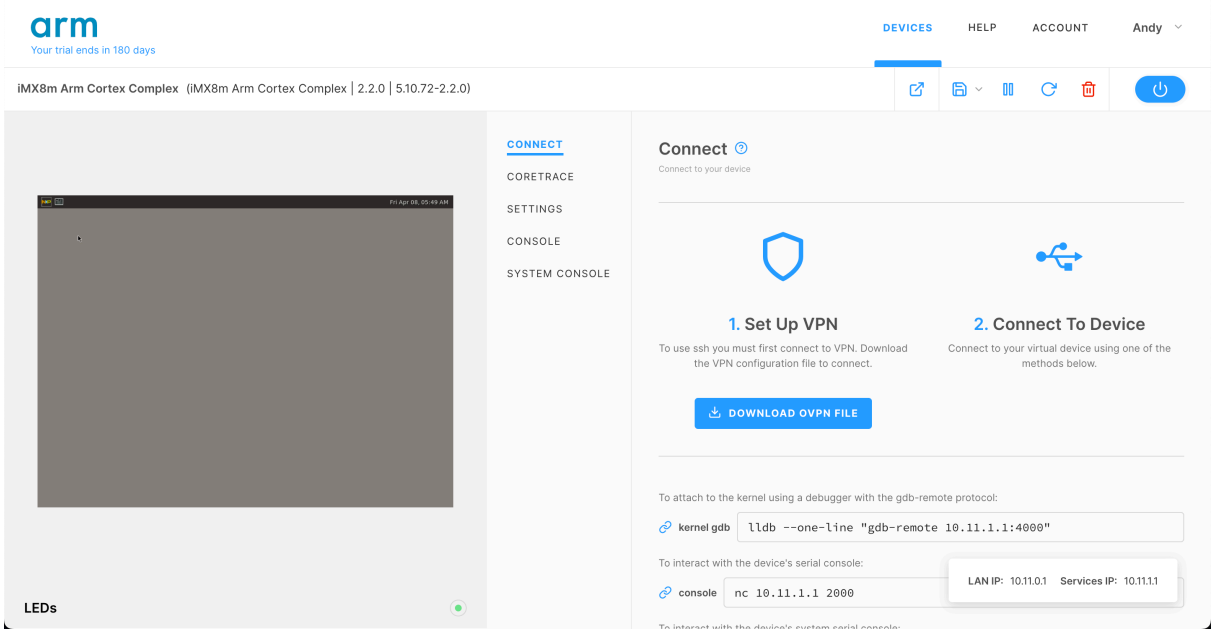
Figure 1-20: Creating device



7. Your iMX8m Arm Cortex Complex is now ready to go!

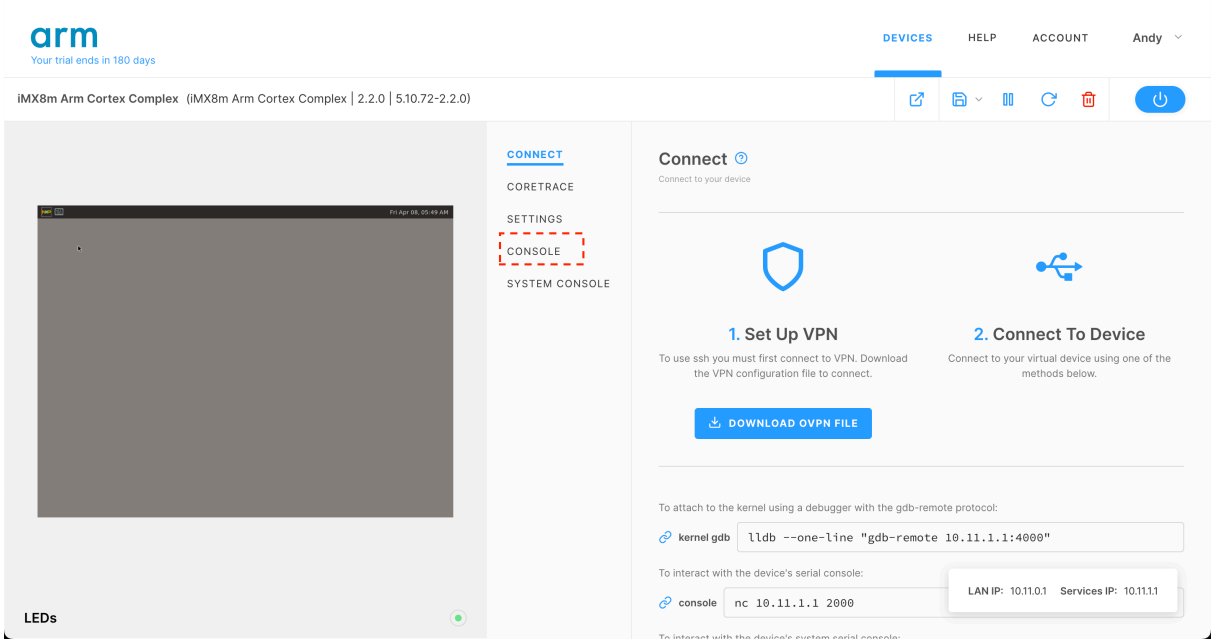
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-21: iMX8m Arm Cortex Complex



8. Next, let's log in. Click CONSOLE on the devices menu.

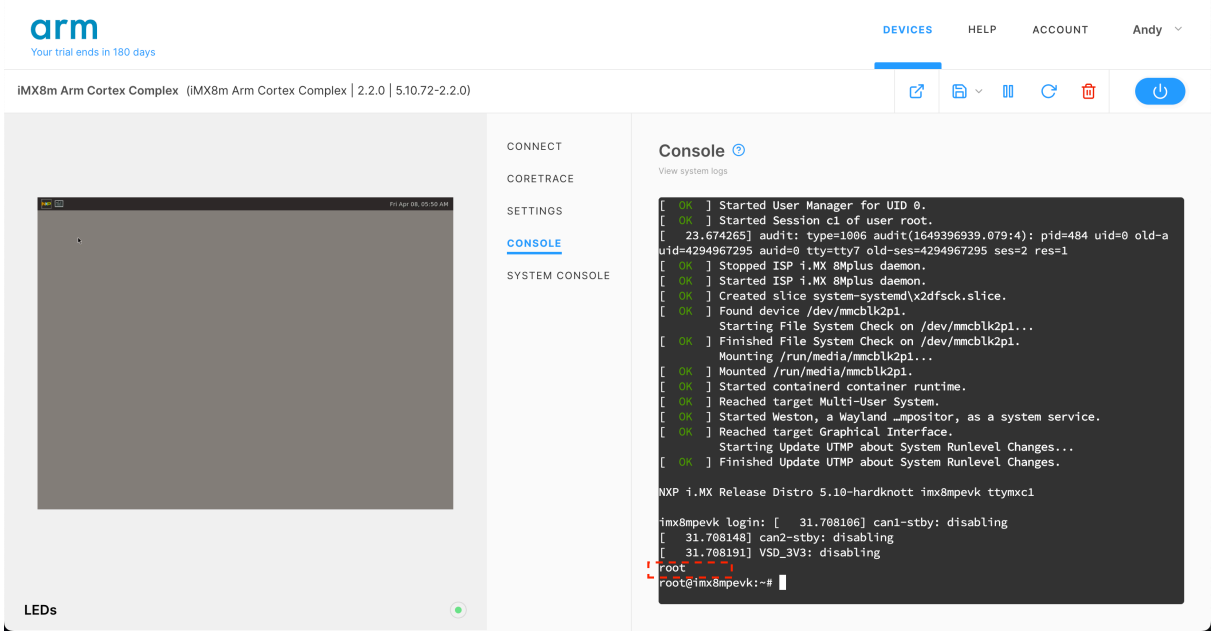
Figure 1-22: Select Console



9. Type into Login `root` then hit `return`, and you are logged in.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-23: Login



Note

The text after the login in the screenshot is normal as this is the device putting the camera to sleep.

10. Now your device is set up, logged in and ready to use.

1.3.3 Next steps

With your new device, you could follow this tutorial to blink the LED.

1.4 Quickstart for the API Docs

This quickstart will demonstrate using the API Docs to create a new device. It will include logging in, setting the API Token, finding your project ID, and finally creating the device.

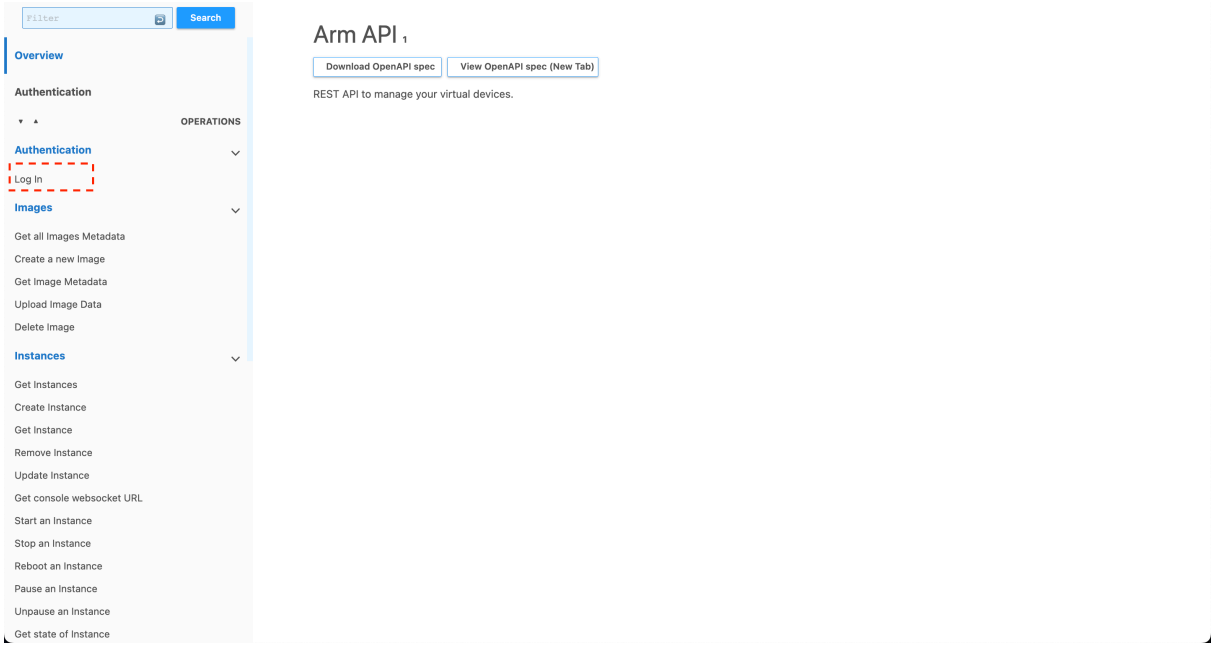
1. Log in to AVH with your Arm user account.

   https://avh.arm.com/ If you do not have or need to create an Arm account register at https://developer.arm.com/
2. To retrieve your API Token, follow the steps in generating an AVH API Token article.
3. Next, go to the AVH API Docs https://app.avh.arm.com/api/docs

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

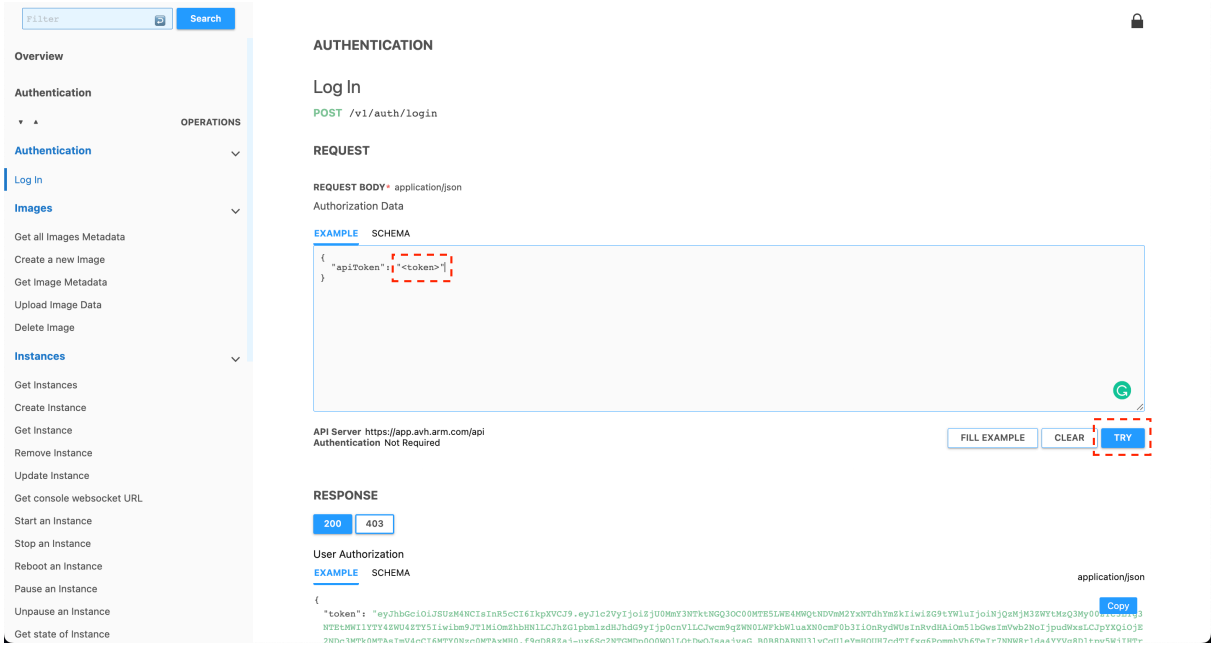
4. Click on “Log In” on the menu.

Figure 1-24: Click “Log In”



5. Paste the API Token you generated into the example request replacing `<token>`, then click “TRY”.

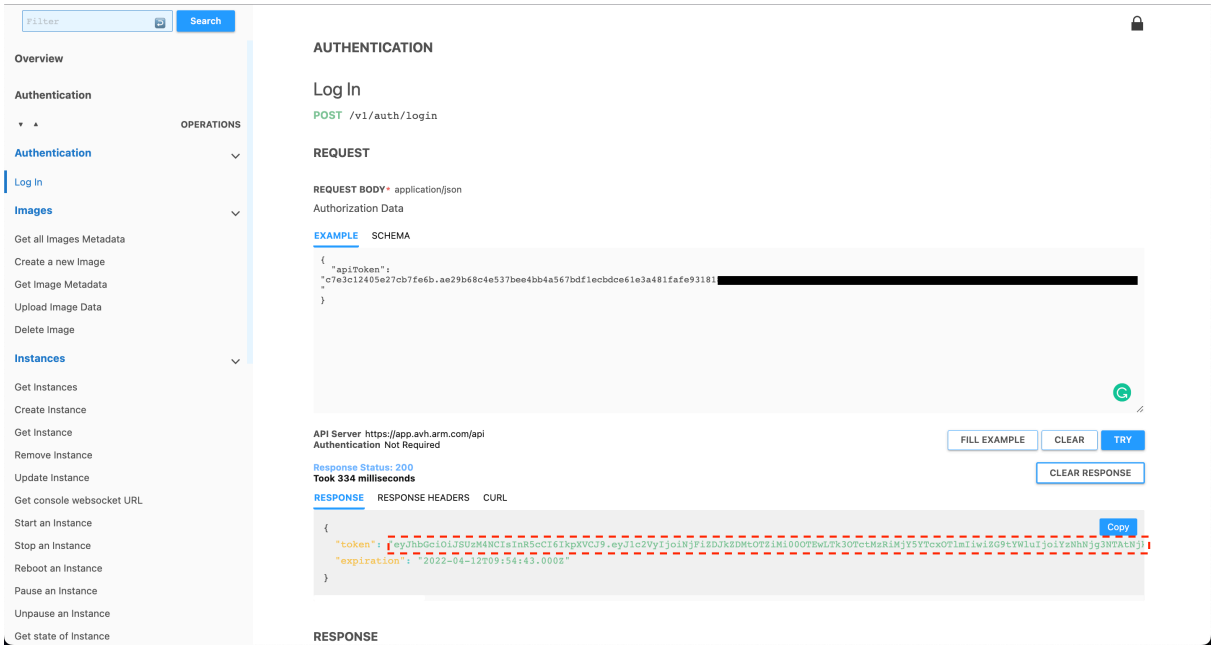
Figure 1-25: Paste the API Token



6. You can now see your auth token in the response. Copy just the value of the token. We are going to use this in the HTTP Bearer.

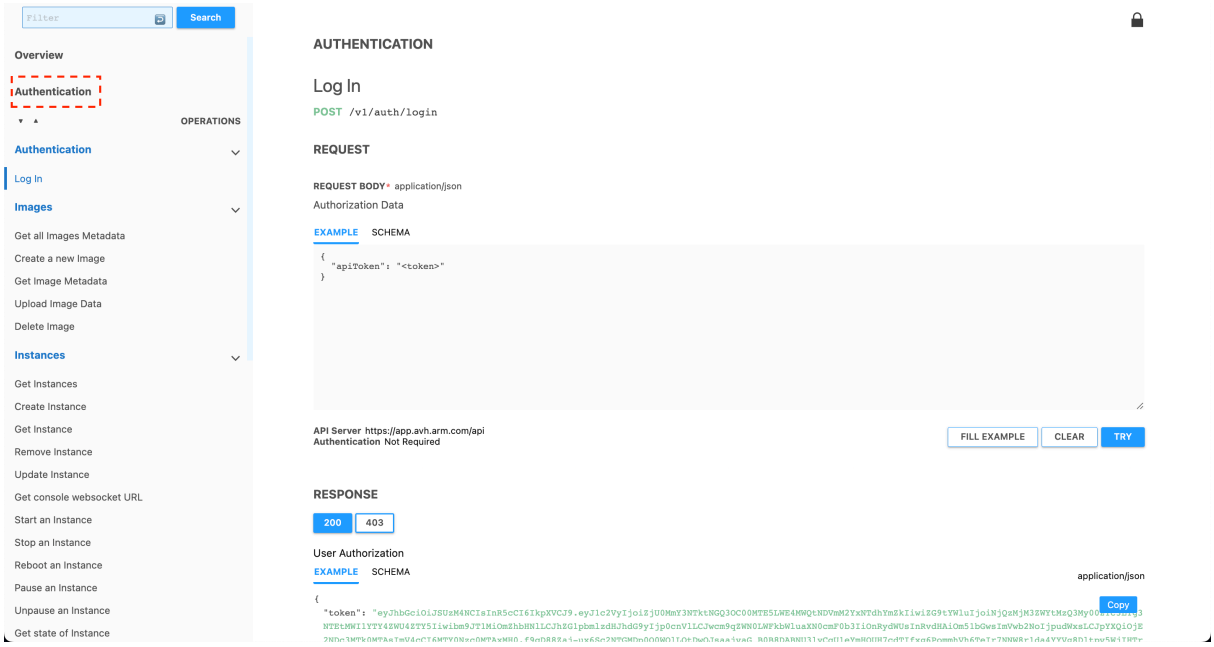
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-26: The auth token



7. Click “Authentication” in the menu.

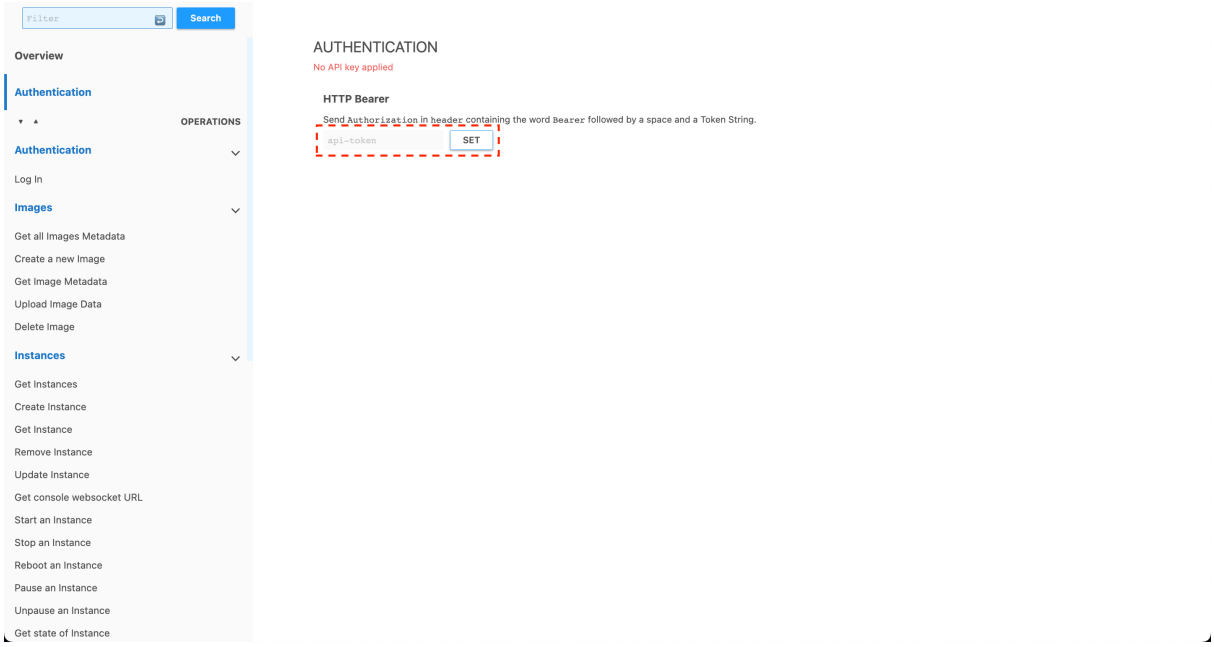
Figure 1-27: Click “Authentication”



8. Paste your auth token in the input and click “SET”

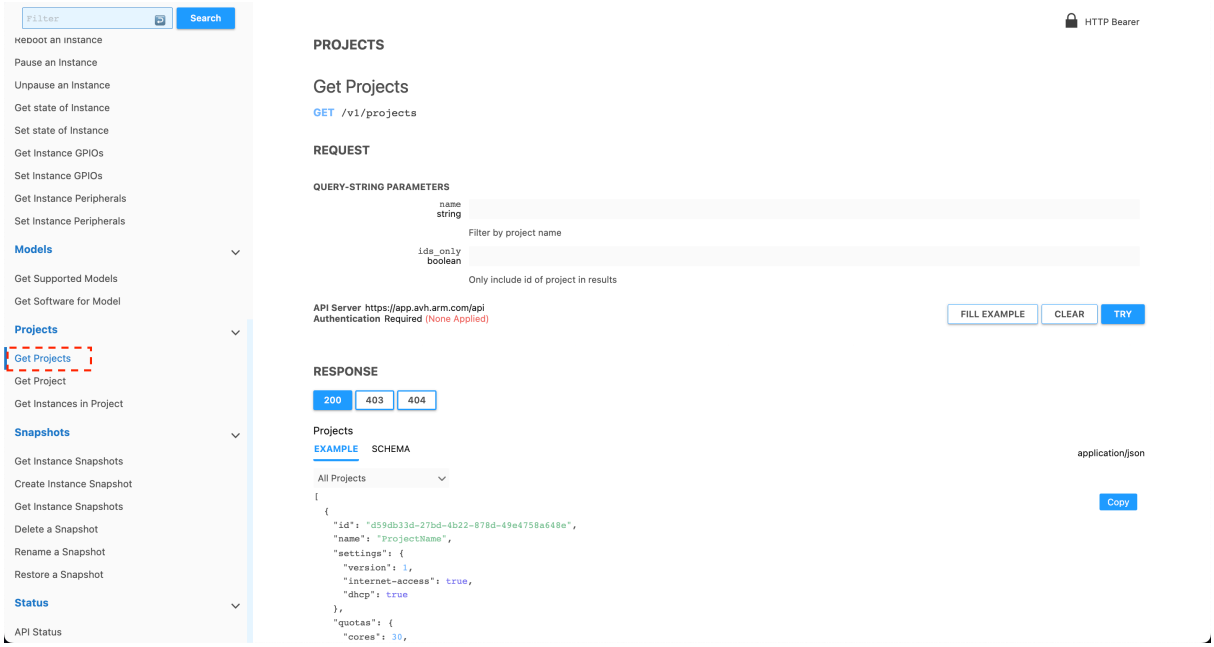
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-28: Paste the auth token



9. Click “Get Projects”

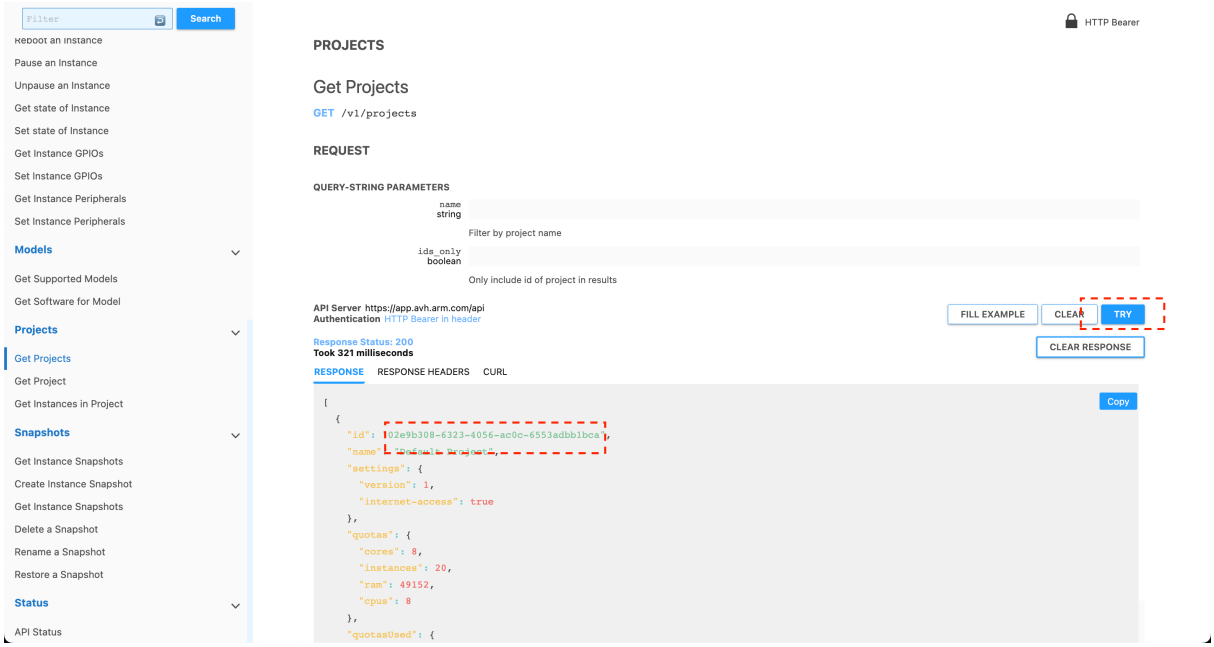
Figure 1-29: Click “Get Projects”



10. Click “TRY” to get your project’s information. Copy the `id` from the JSON, you will need this to create the device.

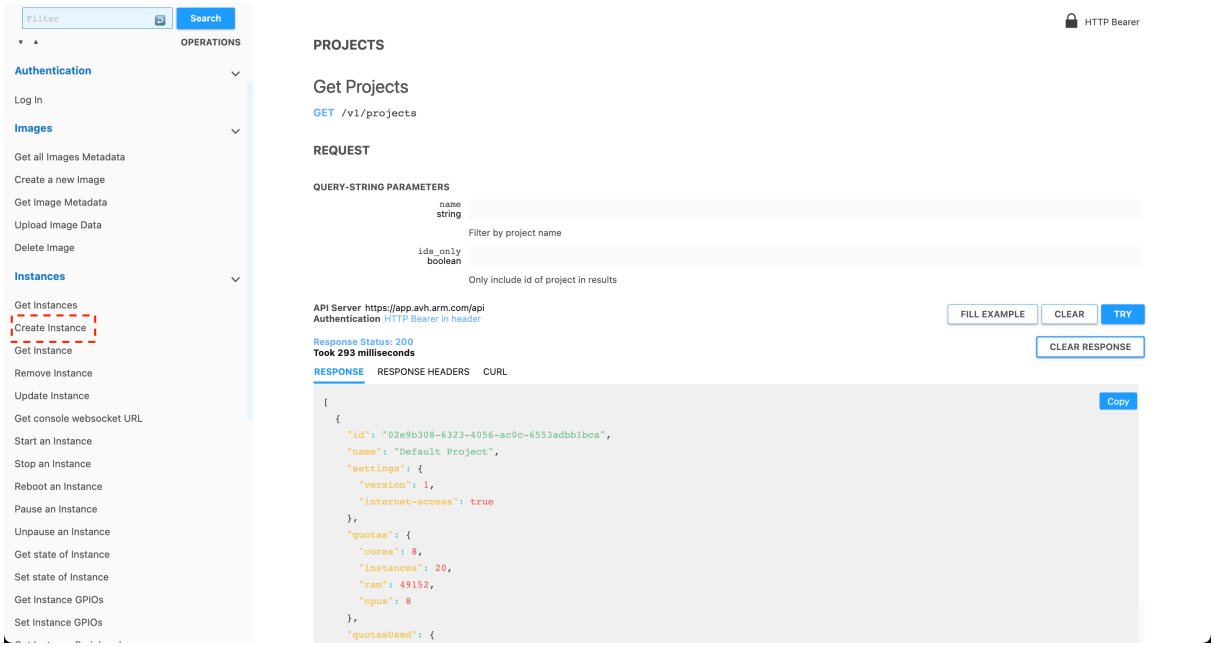
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-30: Copy the id from the JSON



11. In the menu click “Create Instance”

Figure 1-31: Click “Create Instance”



12. We are going to create a STM32U5 IoT Discovery Kit device and name it “STM32U5 created via API”. Paste the object into the example replacing the sample object with your project id.

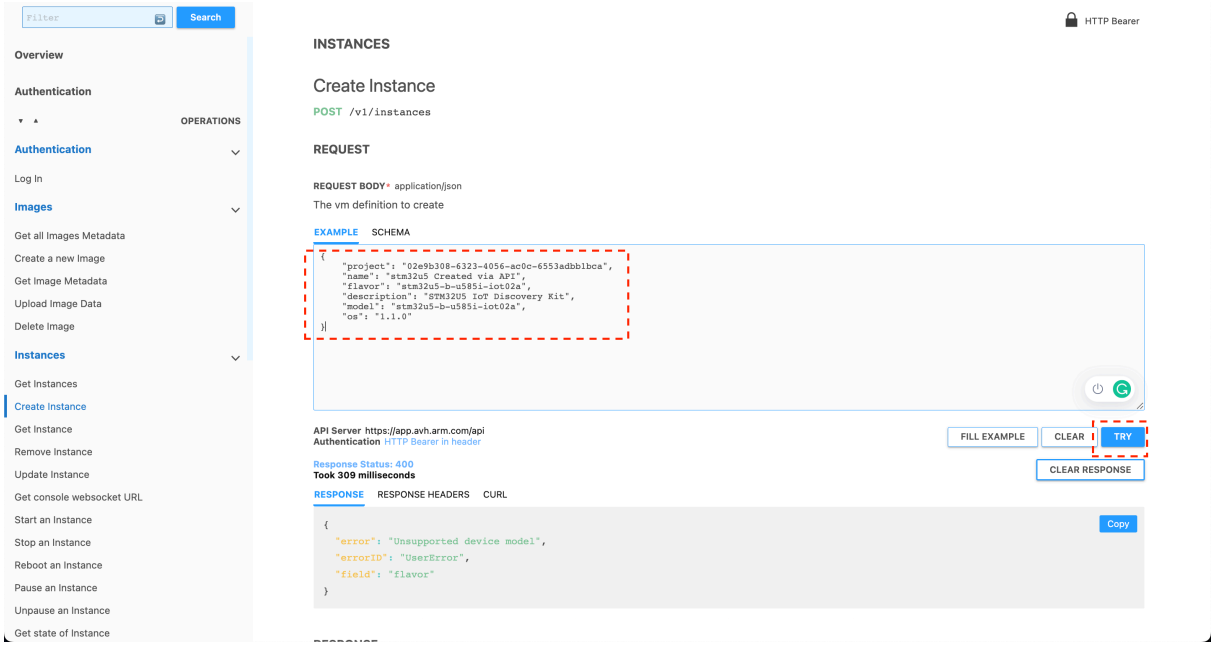
```
{
```

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

```
  "project": "<your_project_id>",
  "name": "stm32u5 Created via API",
  "flavor": "stm32u5-b-u585i-iot02a",
  "description": "STM32U5 IoT Discovery Kit",
  "model": "stm32u5-b-u585i-iot02a",
  "os": "1.1.0"
}
```

Once you have added this click “Try”

Figure 1-32: Click “Try”



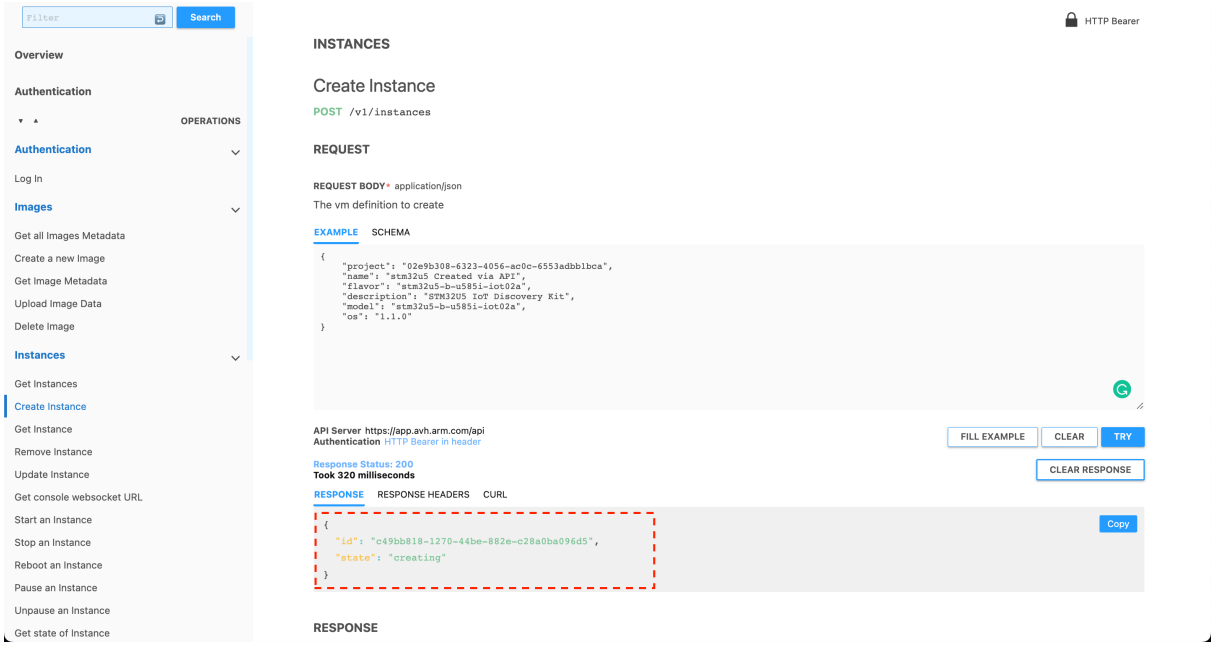
Note

You can find information about the models and the software for the models in the API Doc

13. The response will show that the device status is being created.

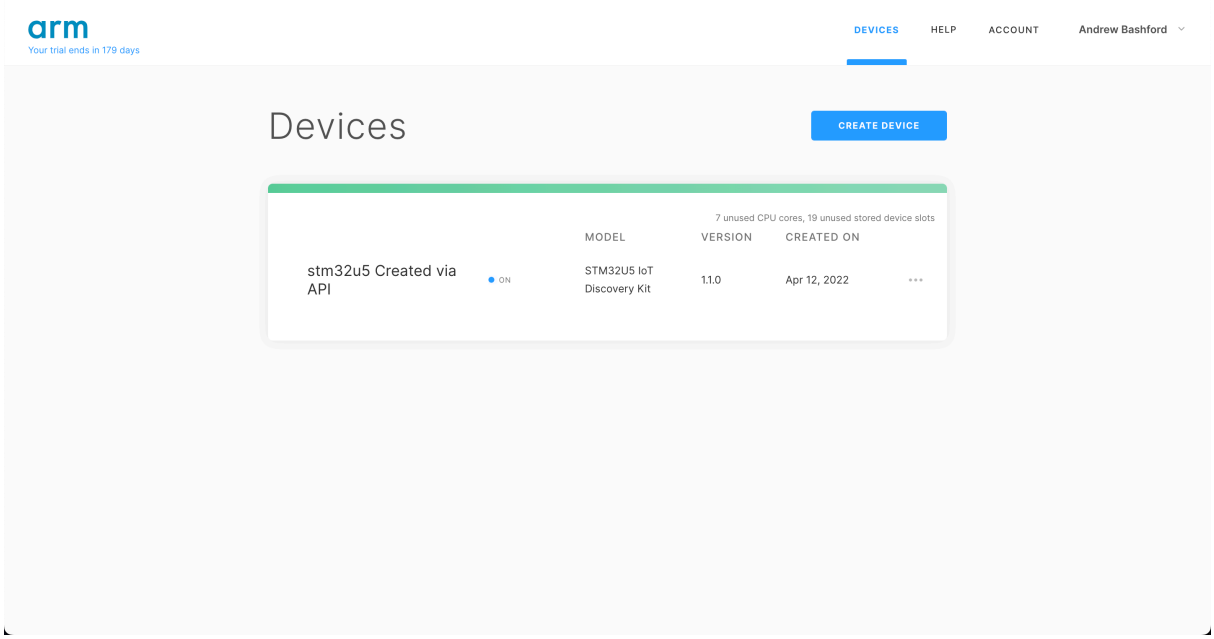
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 1-33: Response showing device status



14. When you go back to AVH the devices list will be the device we created in this quickstart guide.

Figure 1-34: AVH devices list



The API Docs has a rich set of features to programmatically use AVH devices for your use cases.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2. Supported Devices and Components

See which hardware components our virtual devices support:

- Supported Hardware Components for the Raspberry Pi 4
- Supported Hardware Components for the STM32U5
- Supported Hardware Components for the i.M X8M

2.1 Supported Hardware Components for the Raspberry Pi 4

See which hardware components of the Raspberry Pi 4 our virtual devices support.

Note

This information is up-to-date as of 6/6/2022.

2.1.1 High-Level Summary

The virtualized Raspberry Pi 4 board models the following:

- SoC
  - Broadcom BCM2711B0 quad-core A72 (ARMv8-A) 64-bit, runs as Neoverse-N1
- GPU
  - Broadcom VideoCore VI (partially modeled)
- Networking
  - Gigabit Ethernet
- RAM
  - 1GB, 2GB, or 4GB LPDDR4 SDRAM
- GPIO
  - 40-pin GPIO header, populated (user can model using vMMIO)
- Storage
  - microSD
- Ports
  - 2 × micro-HDMI 2.0
  - 3.5 mm analog audio-video jack

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 2 × USB 2.0
- 2 × USB 3.0
- Camera Serial Interface (CSI)

2.2 Supported Hardware Components for the STM32U5

Table of Contents

- High-Level Summary
- Detailed List of Components

Note

This information is up-to-date as of 6/6/2022.

2.2.1 High-Level Summary

The virtualized STM325U5 board models the following:

- 802.11 b/g/n compliant Wi-Fi
- MEMS sensors:
  - 2 digital microphones
  - Relative humidity and temperature sensor
  - 3-axis magnetometer
  - 3D accelerometer and 3D gyroscope
  - Pressure sensor, 260-1260 hPa absolute digital output barometer
- Ambient-light sensor
- 2 user LEDs
- User push-button
- Reset push-button
- USB Type-C

2.2.2 Detailed List of Components

The detailed list of components is as follows:

- CPU

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

    - ICACHE
    - DCACHE
    - RAMCFG
    - EXTI
- Accelerators
    - CORDIC model
    - FMAC model
    - CRC model
- Power
    - PWR model
- Clocks and Watchdogs
    - RCC
    - RTC
    - CRS
    - IWDG
    - WWDG
- Timers
    - TIM6/7
    - TIMx
    - LPTIM
- DMA
    - LPDMA
    - GPDMA
    - DMA2D
        - Stub only since the board has no LCD
- ADC, DAC, Analog
    - ADC4
    - ADC1
    - DAC
    - VREFBUF
    - COMP
    - OPAMP
- Communication, Interconnect, IO
    - USART/UART
    - I2C

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

    - LPUART
    - SPI
    - OTG_FS
    - OCTOSPI
    - FDCAN
    - UCPD
    - GPIO
- Storage
    - FLASH
    - SDMMC
    - FSMC
    - DLYB
- Security
    - GTZC
    - TAMP
    - RNG
    - AES / SAES
    - HASH
    - PKA
- Audio/Video/Touch
    - MDF
    - ADF
    - TSC
- External
    - EMW3080 WLAN
    - TCPP03 USB type-C PD controller
    - M24256 EEPROM
    - IIS2MDCTR magnetometer
    - ISM330DHCX accelerometer and gyro
    - LPS22HH barometer
    - HTS221 thermohydrometer
    - VEML6030 ambient light sensor
    - MX25LM51245G
    - APS6408L-3OBM

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2.3 Supported Hardware Components for the i.M X8M

See which hardware components of the NXP i.M X8M our virtual devices support.

Table of Contents

- High-Level Summary
- Detailed List of Components
  - On-Chip Resources
  - Off-Chip Resources

Note

This information is up-to-date as of 6/6/2022.

2.3.1 High-Level Summary

The virtualized NXP iMX8MPlus board models the following:

- Main CPU Platform
  - Quad Core Arm Cortex-A53 (Trustzone not implemented)
- Low Power, Security CPU
  - Arm Cortex-M4
- External Memory
  - 3 SD/eMMC
- Connectivity and I/O Peripherals
  - MIPI CSI (4-lanes each)
  - 18 I2S TDM (32 b @ 384 kHz), ASRC, 8-ch. PDM DMIC input, eARC
  - 2 USB 2.0/3.0 Type C
  - 2 GbE (1x TSN)
  - UART x4

Note

The following interfaces/IP are not modeled:

- DSP and NPU
- ISP
- PCIe Gen 3

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Raw NAND

2.3.2 Detailed List of Components

In this section read about on-chip and off-chip resources.

2.3.2.1 On-chip resources

In this section read about on-chip resources.

- Security
  - Resource Domain Controller (RDC)
  - CAAM
- Arm Platform and Debug
  - Arm Cortex A53 Platform (A53)
  - Arm Cortex M7 Platform (CM7)
  - Messaging Unit (MU)
  - Semaphore (SEMA4)
  - On-Chip RAM Memory Controller (OCRAM)
  - Shared Peripheral Bus Arbiter (SPBA)
  - System Counter (SYS_CTR)
- Clocks and Power Management
  - Clock Control Module (CCM)
    - PLL and clock root selection modeled for peripherals that use clock frequencies.
  - General Power Controller (GPC)
  - Crystal Oscillator (XTALOSC)
  - Thermal Monitoring Unit (TMU)
    - Returns constant but valid temperature.
- SNVS, Reset, Fuse, and Boot
  - On-Chip OTP Controller (OCOTP_CTRL)
  - Secure Non-Volatile Storage (SNVS)
  - 6.6 Watchdog Timer (WDOG)
    - Watchdog will cause system reboot / power off.
- Interrupts and DMA
  - 7.1 Interrupts and DMA Events
    - GICv3 and NVIC modeled.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

    - Smart Direct Memory Access Controller (SDMA)
        - Documented features modeled. Model includes ROM.
    - Interrupt Request Steering (IRQ_STEER)
- Chip I/O and Pinmux
    - IOMUX Controller (IOMUXC)
        - Necessary features modeled (registers, INITVTOR for M7).
    - General Purpose Input/Output (GPIO)
- External Memory
    - DDR Controller (DDRC)
        - Modeled as block of RAM, no registers.
    - AHB-to-APBH Bridge with DMA (APBH-Bridge-DMA
    - 62BIT Correcting ECC Accelerator (BCH)
        - Modeled for GPMI + APBH_DMA.
    - General Purpose Media Interface (GPMI)
        - Modeled for APBH_DMA with two x8 NAND chips.
- Mass Storage
    - Enhanced Configurable SPI (ECSPI)
    - FlexSPI Controller (FlexSPI)
    - Ultra Secured Digital Host Controller (uSDHC)
- Connectivity
    - HSIO BLK_CTRL
    - Universal Serial Bus Controller (USB)
        - Both host and device modes supported.
    - Universal Serial Bus PHY (USB_PHY)
    - PCI Express (PCIe)
    - PCI Express PHY (PCIe_PHY)
        - PHY model only as needed for PCIe operation.
    - Ethernet MAC (ENET)
        - No QoS support, otherwise complete.
    - Ethernet Quality Of Service (ENET_QOS)
        - No QoS/PTP support, otherwise complete.
    - FlexCAN
- Timers
    - General Purpose Timer (GPT)
    - Pulse Width Modulation (PWM)

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Display, Imaging, and Camera
  - LCD Interface (LCDIF)
    - Supports RGB output modes in this version.
  - MIPI DSI Host Controller (MIPI_DSI)
  - MIPI D-PHY (MIPI_DPHY)
  - LVDS Display Bridge (LDB)
  - HDMI TX Controller
  - HDMI TX PHY
    - PLL modeled, required for audio operation.
  - HDMI TX BLK_CTRL
  - HDMI TX Parallel Audio Interface (HTX_PAI)
    - Minimal register model
  - HDMI TX Parallel Video Interface (HTX_PVI)
    - The HDMI video output is the AVH video output
- Audio
  - AUDIO BLK_CTRL
  - PDM Microphone Interface (MICFIL)
    - The PDM microphone input is the AVH audio input.
    - Stereo input is provided. Other channels are silent.
  - Synchronous Audio Interface (SAI)
  - Enhanced Audio Return Channel (eARC)
  - AUD2HTX HDMI audio output
    - Block is undocumented but required in Linux.
    - The HDMI audio output is the AVH audio output.
  - Low-Speed Communication and Interconnects
    - I2C Controller (I2C)
    - Universal Asynchronous Receiver/Transmitter (UART)

2.3.2.2 Off-chip resources

In this section read about off-chip resources.

- ONFI NAND chips for GPMI
- HDMI monitor (1024×768, 60Hz)
- USB keyboard + mouse
- PCA9450 PMIC

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- PCA6416 GPIO expander
- Wolfson WM8960 audio codec
- PLL frequency computation is required for SAI3 operation.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3. Next Steps

The next steps you can take are as follows:

- Taking and restoring snapshots
- Quick Connect
- Connecting to the VPN
- Connect via SSH
- Connect to the Console

3.1 Taking and restoring snapshots

In this article, we will use the Raspberry Pi 4 device that was created in Quickstart for AVH - Raspberry Pi4 tutorial.

3.1.1 What are snapshots?

Snapshots give you the ability to save the state of a device and then restore or clone that state at a moment's notice. Reduce valuable setup time by instantly returning to a previously saved state, create clones to run multiple tests from the same starting environment, or provide identical starting VMs to all students in a virtual classroom. Snapshots can save you significant time reproducing bugs or repeating tests.

Every virtual device automatically captures a new snapshot right after the initial creation, providing a quick and easy way to revert back to that fresh image. You can take up to five snapshots (in addition to the snapshot from initial creation) per virtual device.

The virtual device must be turned off to take a snapshot.

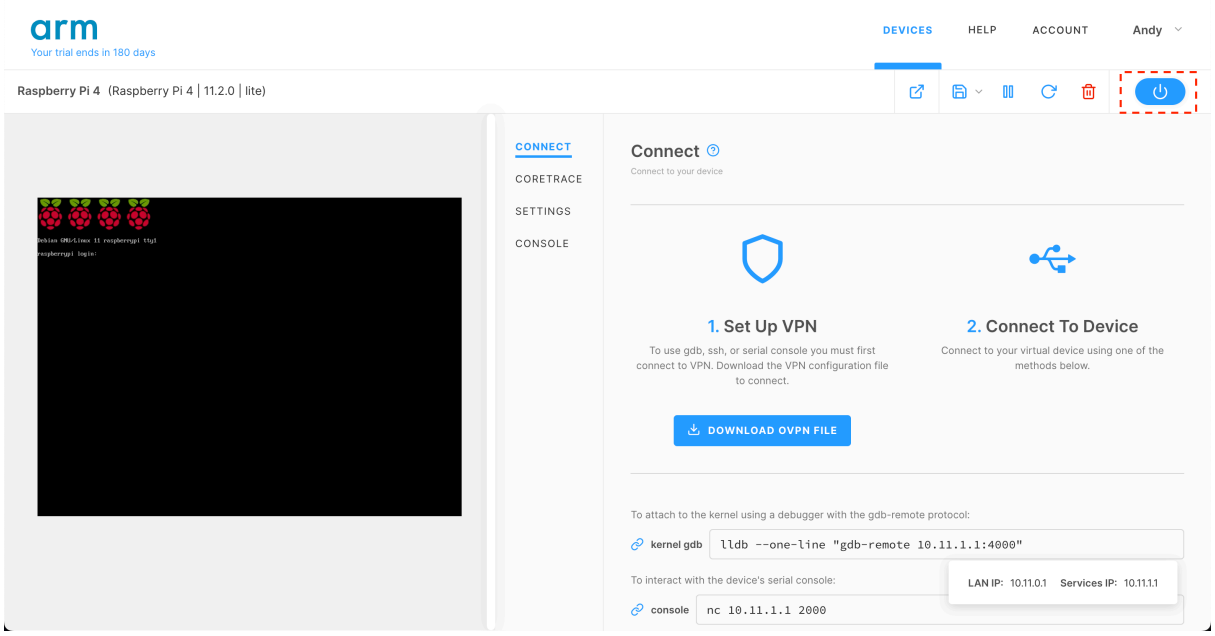
3.1.2 Taking snapshots

This tutorial will demonstrate Snapshots by creating and restoring them.

1. Find the device you created in the Quickstart for AVH - Raspberry Pi4 tutorial.
2. Power off the device by clicking the power symbol on the top-right. Wait until you see POWER ON DEVICE on the left to confirm it is shut down.

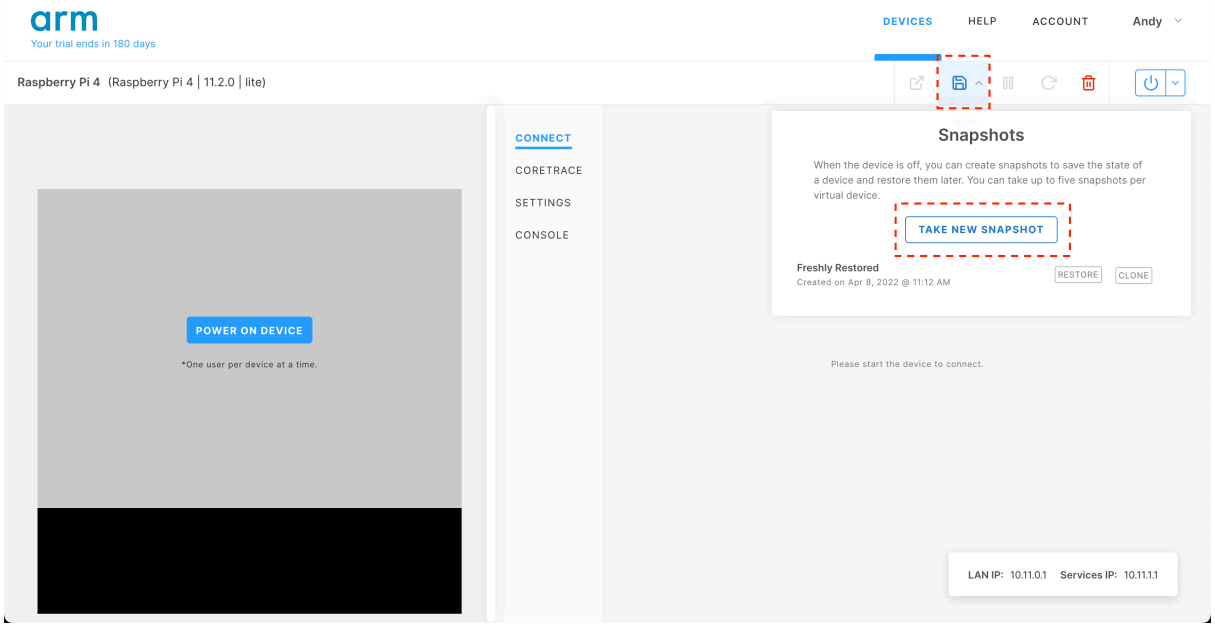
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-1: Power off the device



3. Click the Snapshot icon, then click TAKE NEW SNAPSHOT. You will see Snapshot # 1 appear below the Freshly Restored snapshot.

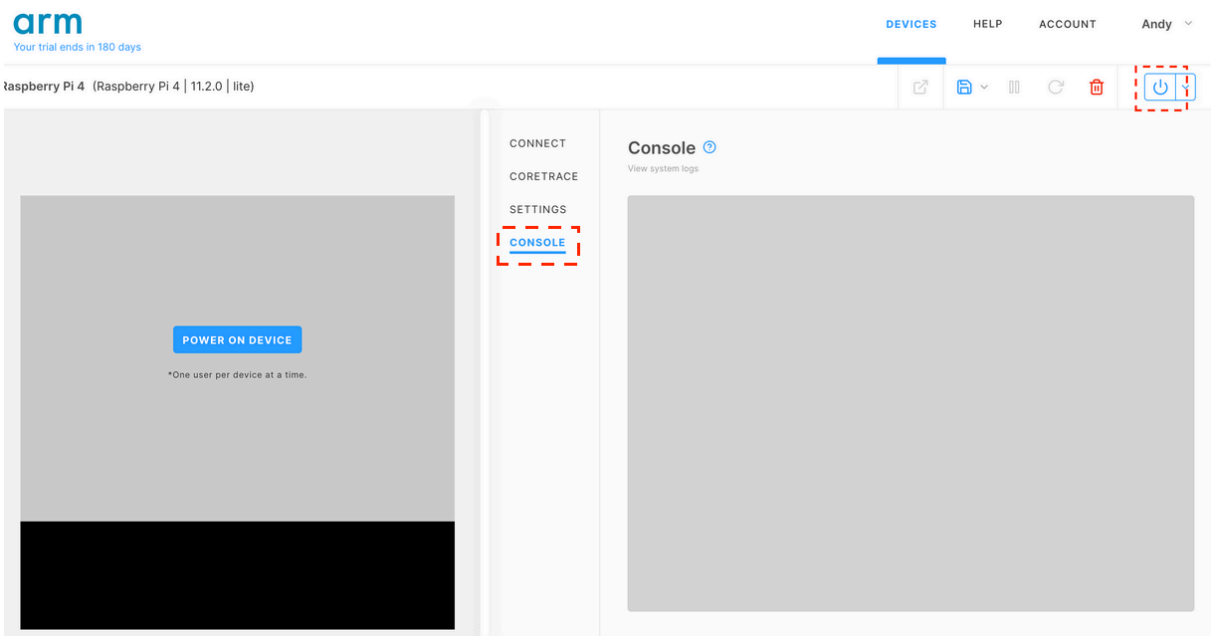
Figure 3-2: Snapshot # 1



4. Now that we saved a snapshot, let's break the device to practice restoring the saved state. Open the CONSOLE and POWER ON the device.

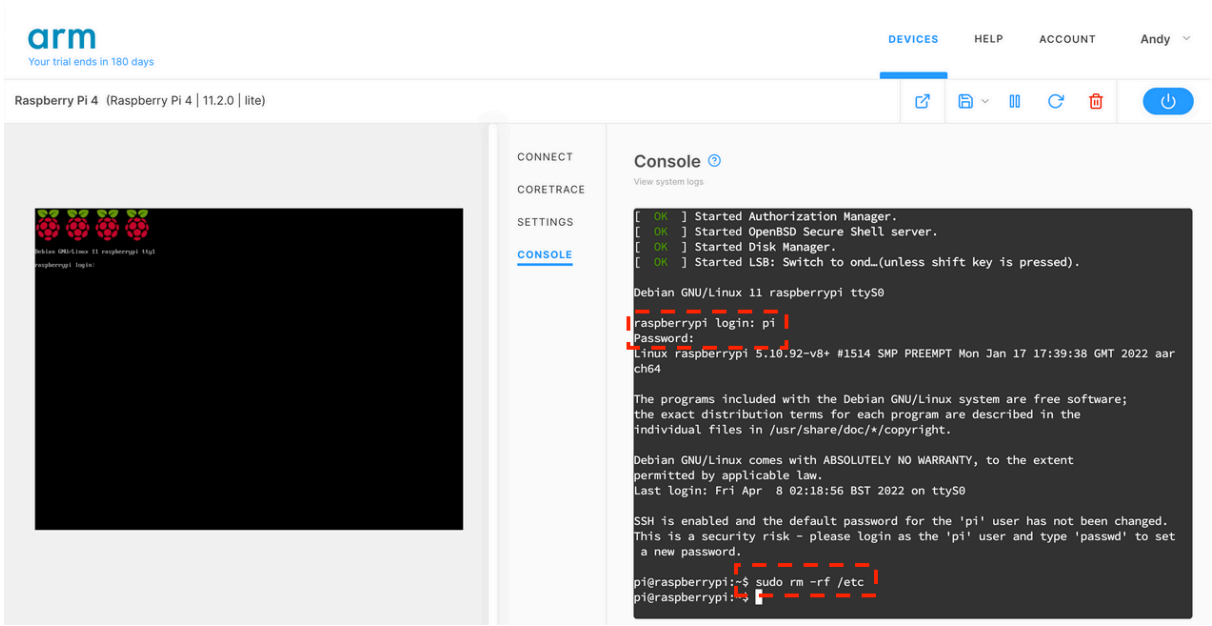
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-3: Open CONSOLE



5. Login with username: `pi` and password:`raspberry`, type `sudo rm -rf /etc`, then hit `Return`.

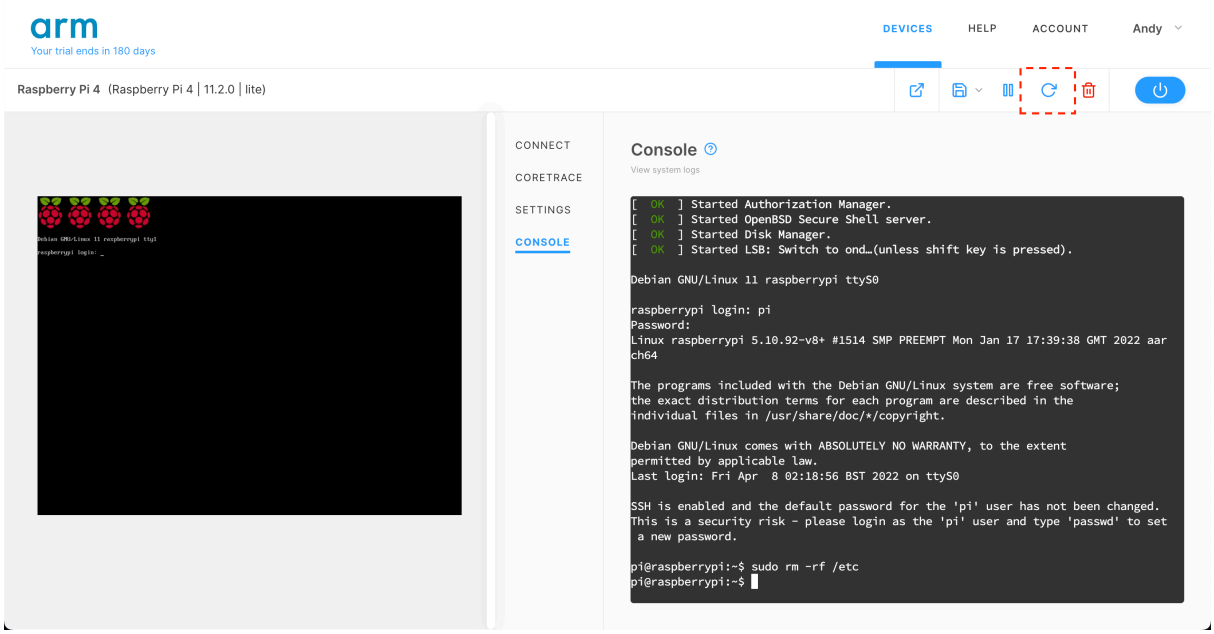
Figure 3-4: Login



6. Click Restart on the top-right.

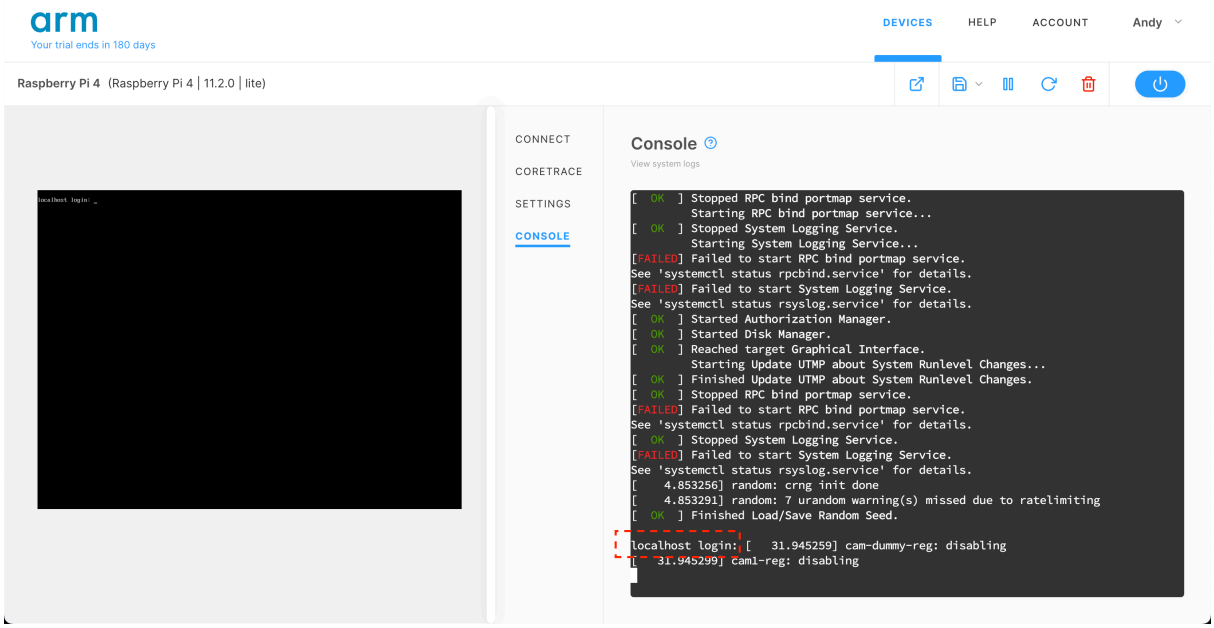
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-5: Restart



7. The Console should display `localhost login:`. This will validate the device is broken, as it is unable to log in normally.

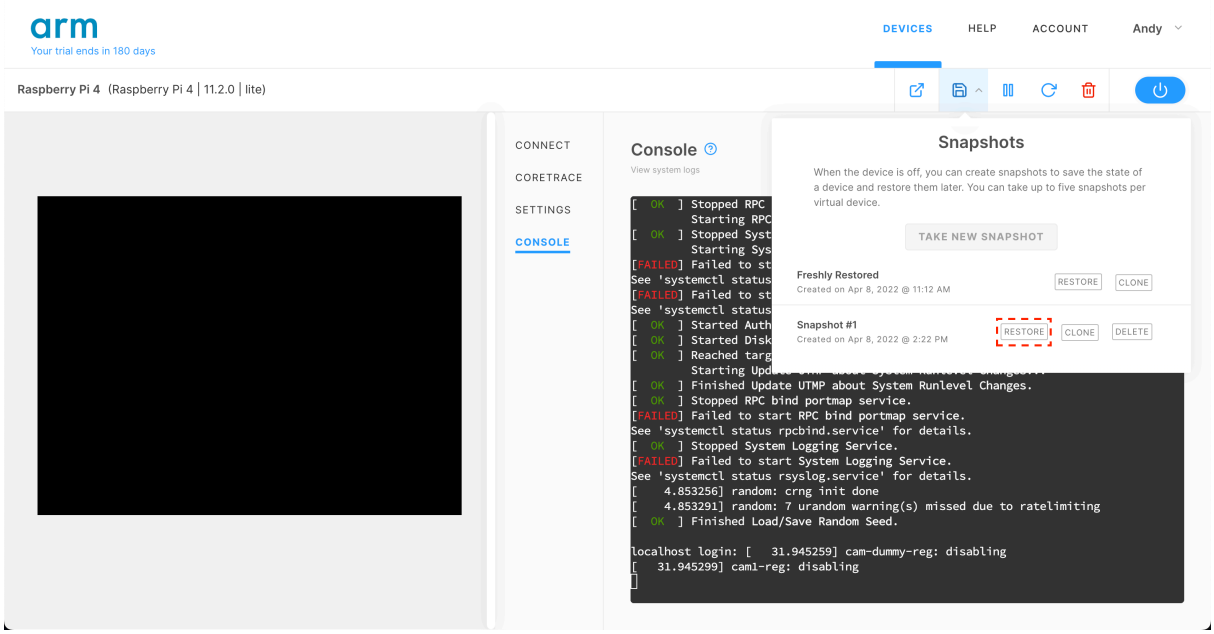
Figure 3-6: Validation



8. We will now restore the snapshot. With the device still powered on, click the Snapshot button then click RESTORE next to Snapshot #1. Wait while the snapshot state is loaded.

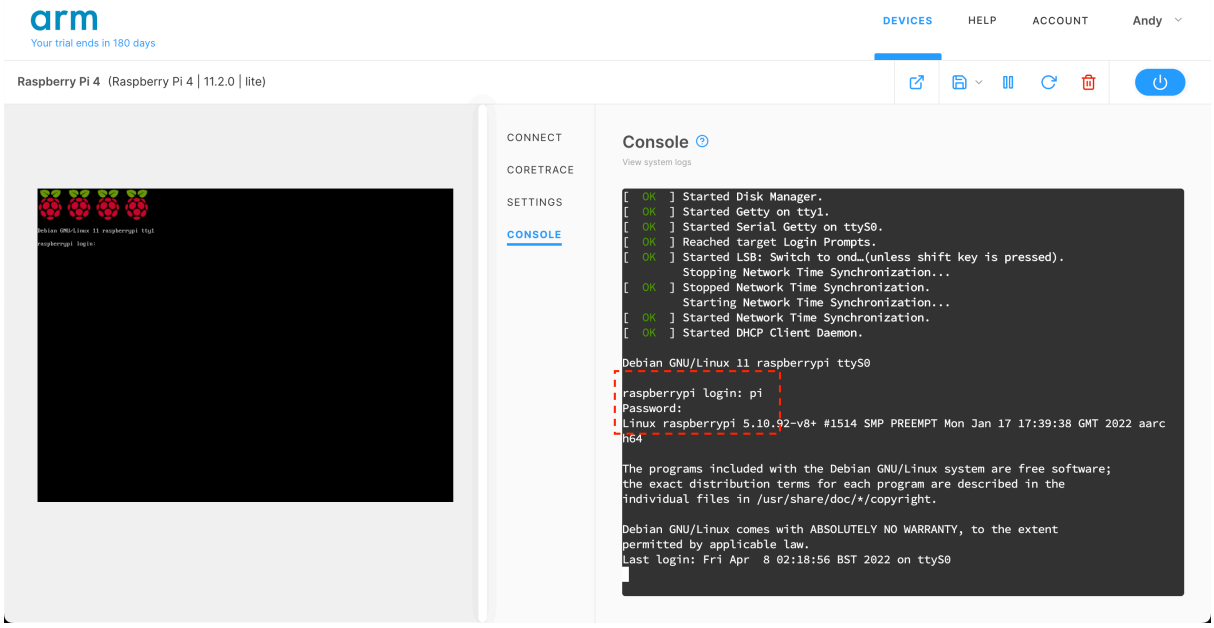
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-7: Restore snapshots



9. After the device boots up, validate that the device is working by logging in with username: `pi` and password: `raspberry`.

Figure 3-8: Validate the device



In the next tutorial, Connecting to the VPN, we will connect our device to a VPN.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3.2 Quick Connect

Learn how to use the Quick Connect feature that enables you to connect without a VPN via an SSH Tunnel.

Table of Contents

- Note About Your Network Firewall Configuration
- Step 1: Create a New Public-Private Key Pair
  - Generate a Key Pair with Terminal
  - Generate a Key Pair with PowerShell
- Step 2: Add Your SSH Public Key to Your Project
- Step 3: Create Your Device
- Step 4: Use Quick Connect
- Specifying the Private Key File for SSH Authentication (Optional)

AVH's Quick Connect feature allows you to connect to your virtual devices and use features without requiring a VPN connection.

This "Quick Connect" feature uses standard SSH, which comes pre-installed on Mac, Windows, and Linux.

Quick Connect is available on Raspberry Pi4 and iMX8m Arm Cortex Complex at present.

No more downloading a VPN client and installing the .ovpn profile to gain root access!

3.2.1 Note About Your Network Firewall Configuration

Your network firewall must allow outbound connections on Port 22 to AVH's proxy server.

You can find the address for the proxy server in the `SSH -J ... or SSH -M ...` command under your device's "Connect -> Quick Connect" section (for example `proxy.avh.com`).

3.2.2 Step 1: Create a New Public-Private Key Pair

AVH supports the use of standard SSH keys, which make use of public-key cryptography for authentication when connecting your local computer with virtual devices.

We will utilize the ssh-keygen tool, which is installed by default on all three major operating systems, and the Ed25519 signature scheme.

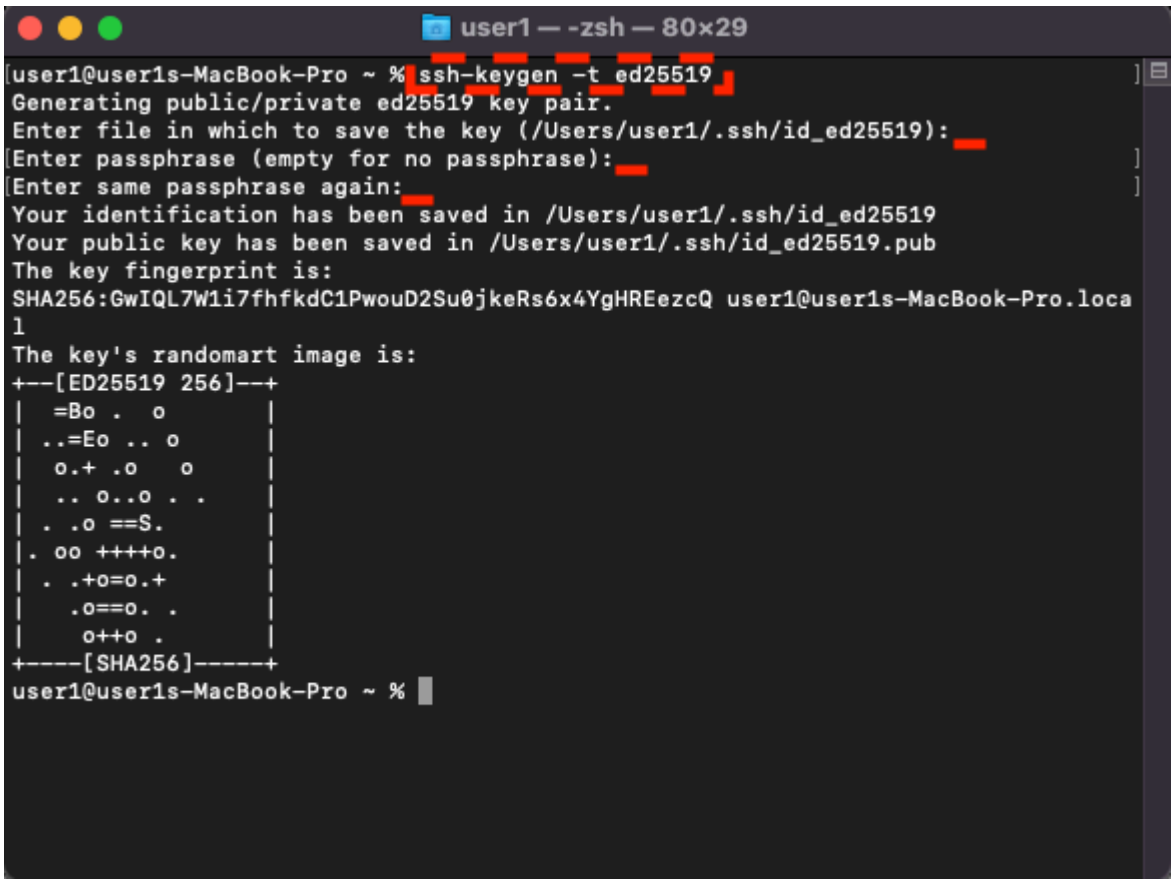
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3.2.2.1 Generate a Key Pair with Terminal

To generate a key pair with Terminal, do the following:

1. Open Terminal and enter `ssh-keygen -t ed25519`.
2. Use the default file location (`/Users/<user>/.ssh/id_ed25519`) by pressing the `return` key.
3. Enter an optional passphrase for added security. In this example, we will leave the passphrase field blank.
4. Confirm your optional passphrase.
5. The terminal will show the location of your identification (private key) and public key files, the key fingerprint, and the randomart image.

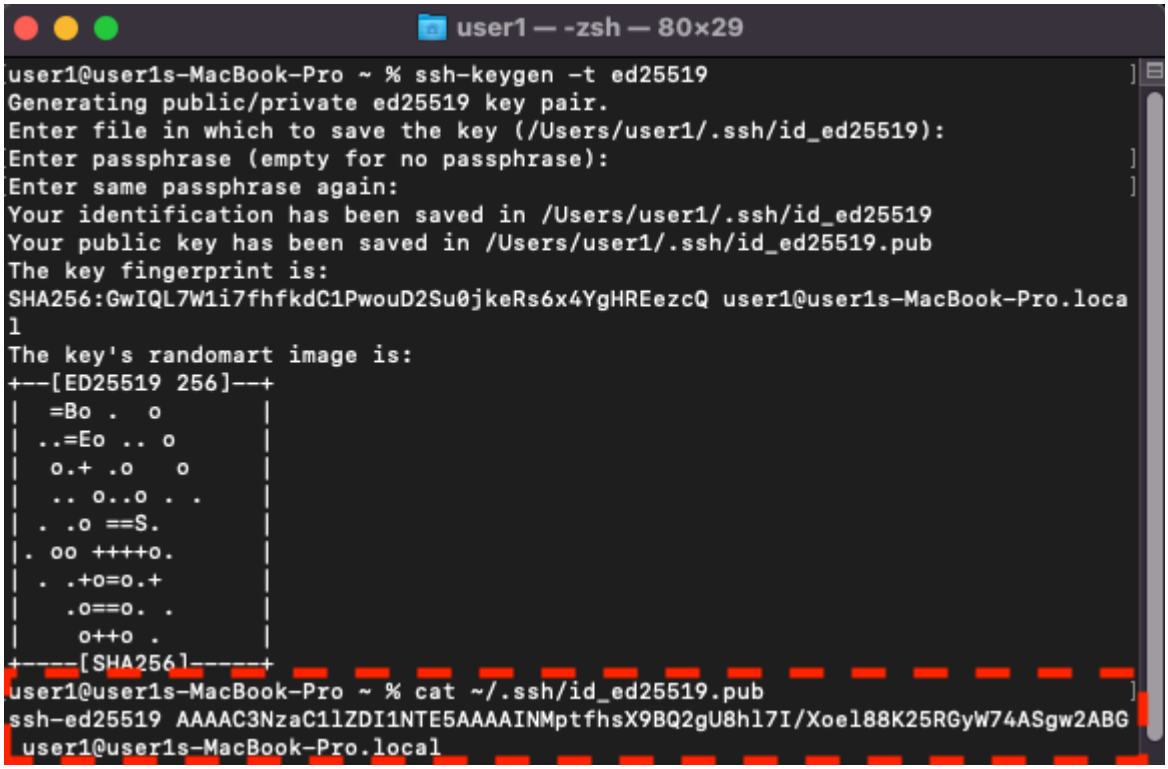
Figure 3-9: ssh-keygen output



6. View the contents of the file using `cat ~/.ssh/id_ed25519.pub`.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-10: File contents



7. Copy the output to use later in Step 2. In our example, we will copy the following:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINMptfhsX9BQ2gU8hl7I/Xoel88K25RGyW74ASgw2ABG
 user1@user1s-MacBook-Pro.local
```

3.2.2.2 Generate a Key Pair with PowerShell

To generate a key pair with PowerShell, do the following:

1. Open PowerShell and enter `cd ~/.ssh`.
2. Start the key generation process using `ssh-keygen -t ed25519`.
3. Use the default file location (`c:\Users\<user>\.ssh\id_ed25519`) by pressing the `Enter` key.
4. Enter an optional passphrase for added security. In this example, we will leave the passphrase field blank.
5. Confirm the passphrase.
6. PowerShell will show the location of your identification (private key) and public key files, the key fingerprint, and the random art image.
7. View the contents of the file using `cat .\id_ed25519.pub`.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

8. Copy the output, which we will use in Step 2. In our example, we will copy:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIURGEJFM5/YvlWysZT4nSnT5bjovGK/Uj8d6fUBsBUp
 <user>@Lenovo-Laptop
```

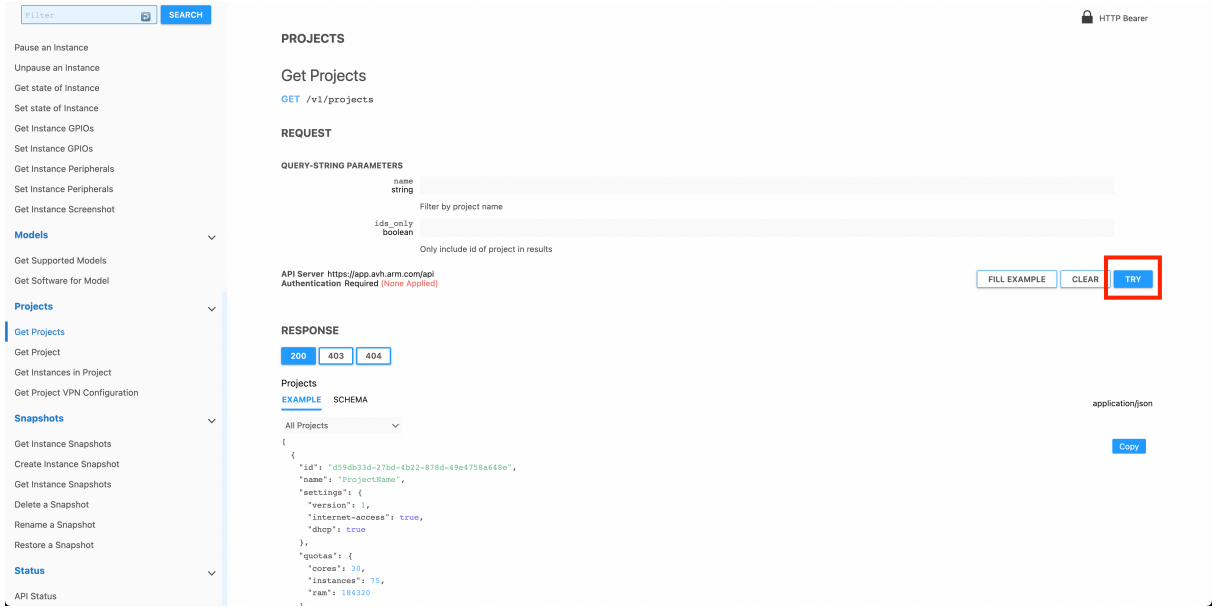
3.2.3 Step 2: Add Your SSH Public Key to Your Project

Once added this SSH key will be included in each new device created.

The AVH REST API supports adding SSH keys to your project using the command line.

1. Authenticate to the Rest API with your API Token. See Quickstart for the API Docs for guidance.
2. Go to the Get Projects in the API and click “Try”.

Figure 3-11: Get Projects



3. Copy the ID of the project named “Default Project” in the response.
4. Go to “Add Project Authorized Key”
5. Add the Project ID you have just copied into the “project Id string” input and add the SSH key you created in Step 1 into the “key” value pair key in the JSON. Now click “TRY”.

   The response will show the newly created key. Please note you can only have one SSH key per project.

   You can check the current SSH key with “Get Project Authorized Keys”.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3.2.4 Step 3: Create Your Device

After adding you add the public key to your project, all devices you create under that project will automatically include the public key.

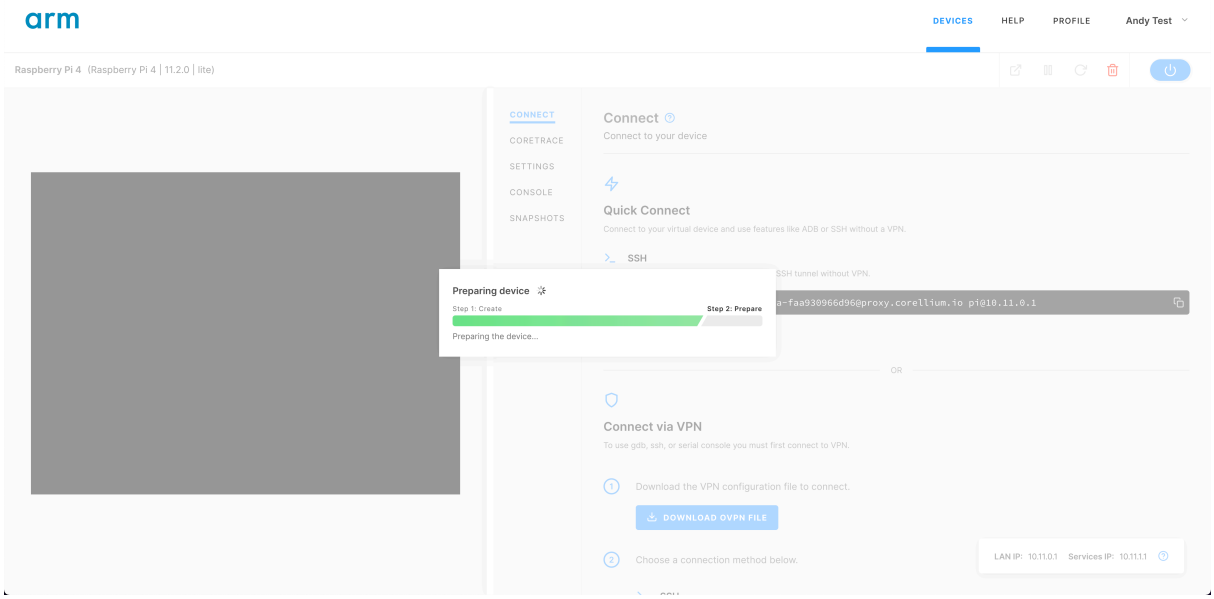
Note

You cannot add SSH keys to an existing device.

---

1. Create a new device and wait for it to boot. In this instance, we will create a Raspberry Pi.

Figure 3-12: New Raspberry Pi device



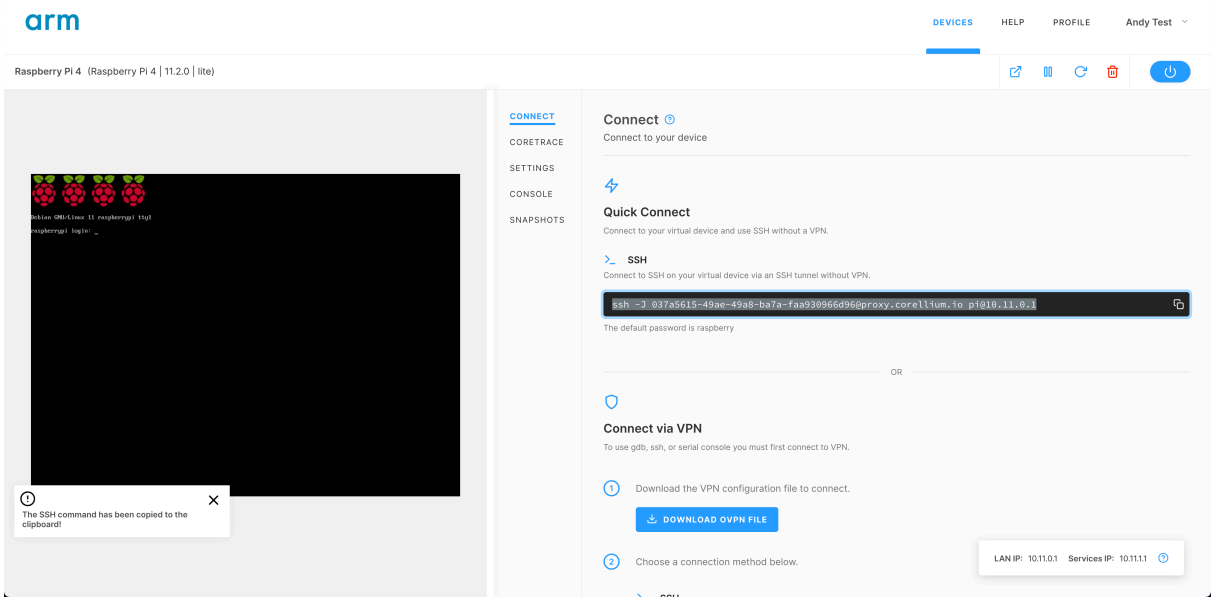
3.2.5 Step 4: Use Quick Connect

To use Quick Connect, do the following:

1. Once booted, copy the Quick Connect SSH command.

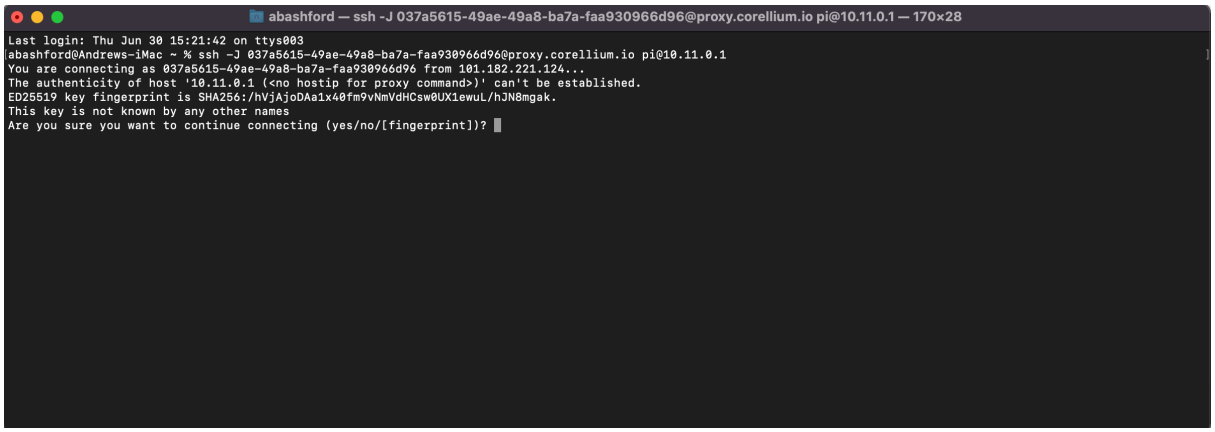
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-13: Quick Connect SSH command



2. Paste and run the copied command into Terminal or Powershell.

Figure 3-14: Paste and run the command



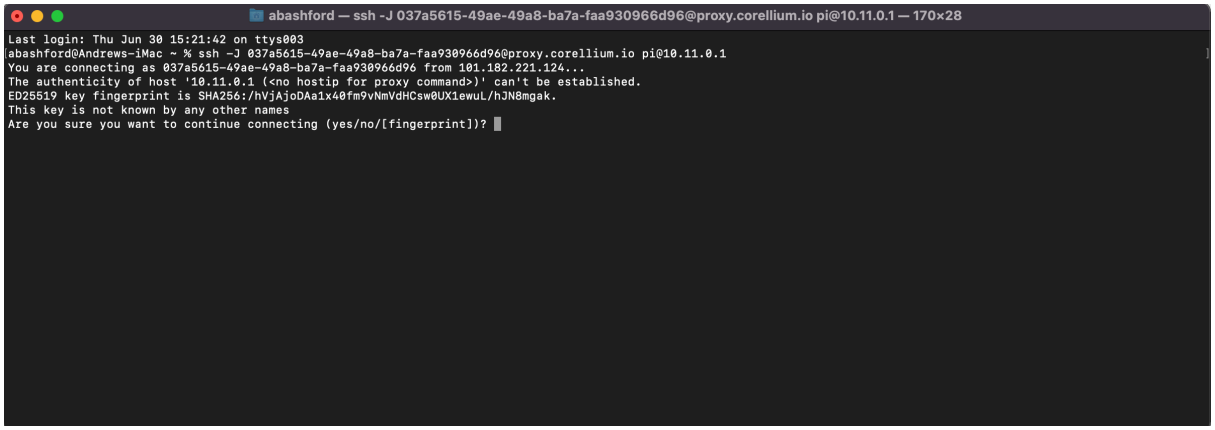
Note

Please note that if you would like to specify the private key file to use for SSH authentication rather than use the default filename, refer to Specifying the Private Key File for SSH Authentication (Optional) at the end of the article.

3. The first time you connect, you will need to confirm the proxy host fingerprint. Type `yes` when asked about connecting to the proxy server (for example `proxy.avh.com`). This step only happens once. You should see a `You are connecting as ...` confirmation message.

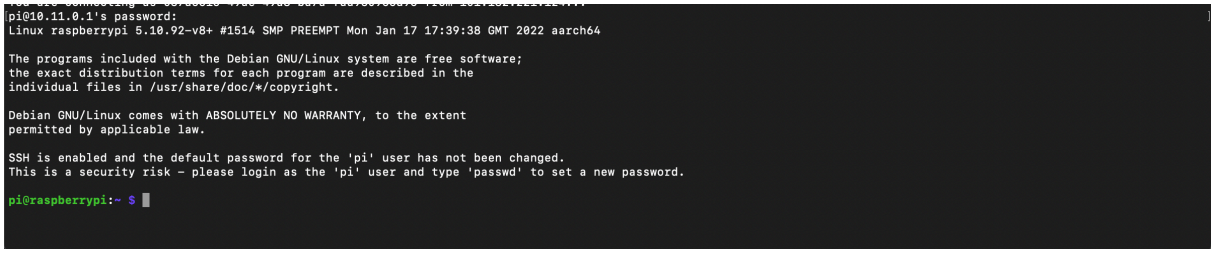
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-15: Confirm the proxy host fingerprint



4. If there is a password required you can enter it now.

Figure 3-16: Enter the password



5. Now you are connected without the need for a VPN.

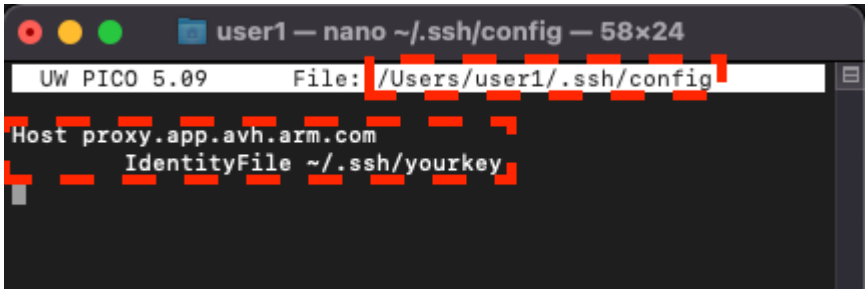
3.2.6 Specifying the Private Key File for SSH Authentication (Optional)

As an advanced option, if you normally use a "named" ssh file key using the `-i` argument, as opposed to using the default filename `id_ed25519` detailed in this article, you have two options:

- Include `-i` arguments in the ssh command. If you plan to use separate keys for the jump proxy host and for the target destination, you will need to include two `-i` arguments. You can read more about this method here.
- Specify the `IndentityFile` in your `~/.ssh/config`. (The private key file must exist in `~/.ssh`, and the associated public key will also need to be added to your AVH project.)

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-17: Specify the IndentityFile



3.3 Connecting to the VPN

In this article, we will use the Raspberry Pi 4 device that was created in Quickstart for AVH - Raspberry Pi4 tutorial.

- How to Connect with Mac OS
- How to Connect with Windows OS

3.3.1 Why Use the VPN?

Using a VPN will put your Virtual Device and the computer you are accessing AVH from (typically a desktop or laptop computer) on the same network.

Common use cases for this are:

- SSH access to the device
- Access your application via various TCP/IP or UDP ports and protocols from your desktop/laptop computer.
- Kernel debugging
- Run a vmmio peripheral over TCP/IP on your local computer to feed stimuli data through a peripheral running locally on your computer into the Virtual Device for application and algorithm validation.

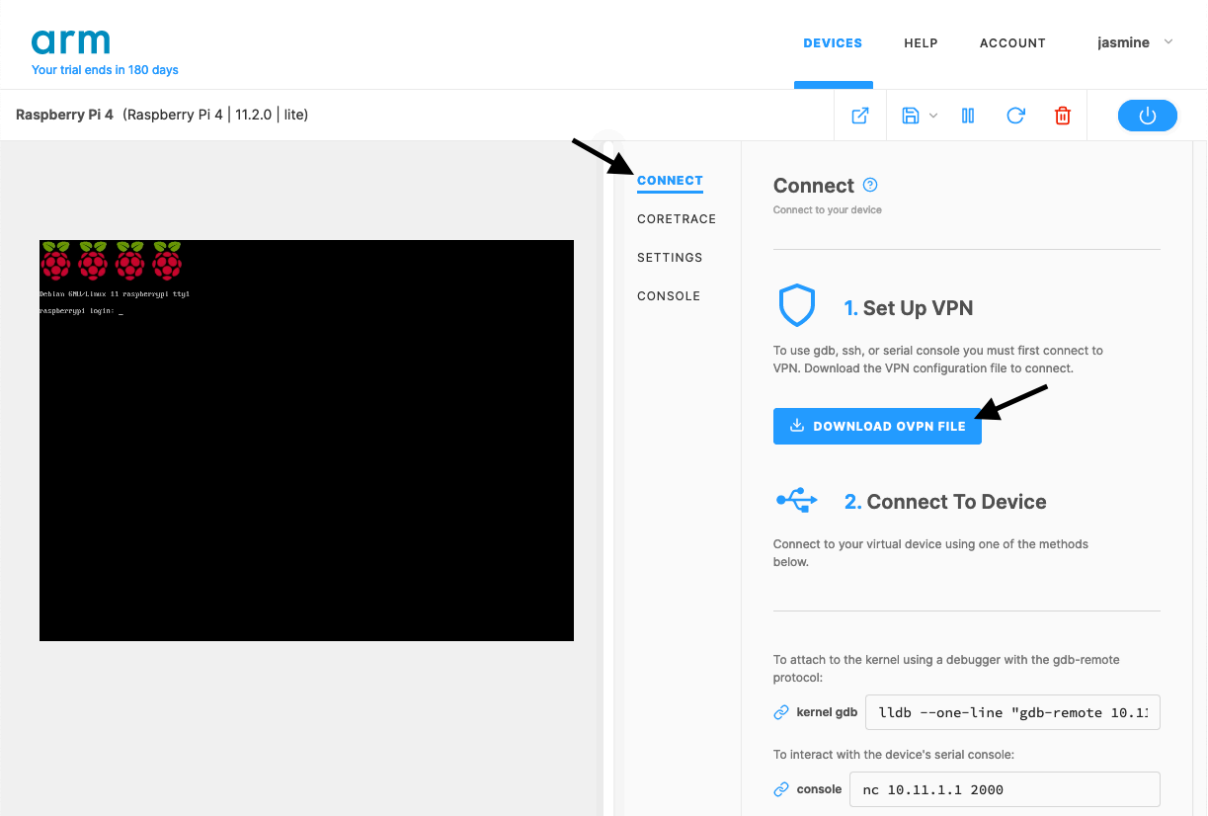
3.3.2 How to Connect with Mac OS

To connect with Mac OS, do the following:

1. Go to the device created in the Quickstart for AVH - Raspberry Pi4 tutorial.
2. First, click CONNECT and then DOWNLOAD OVPN FILE.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

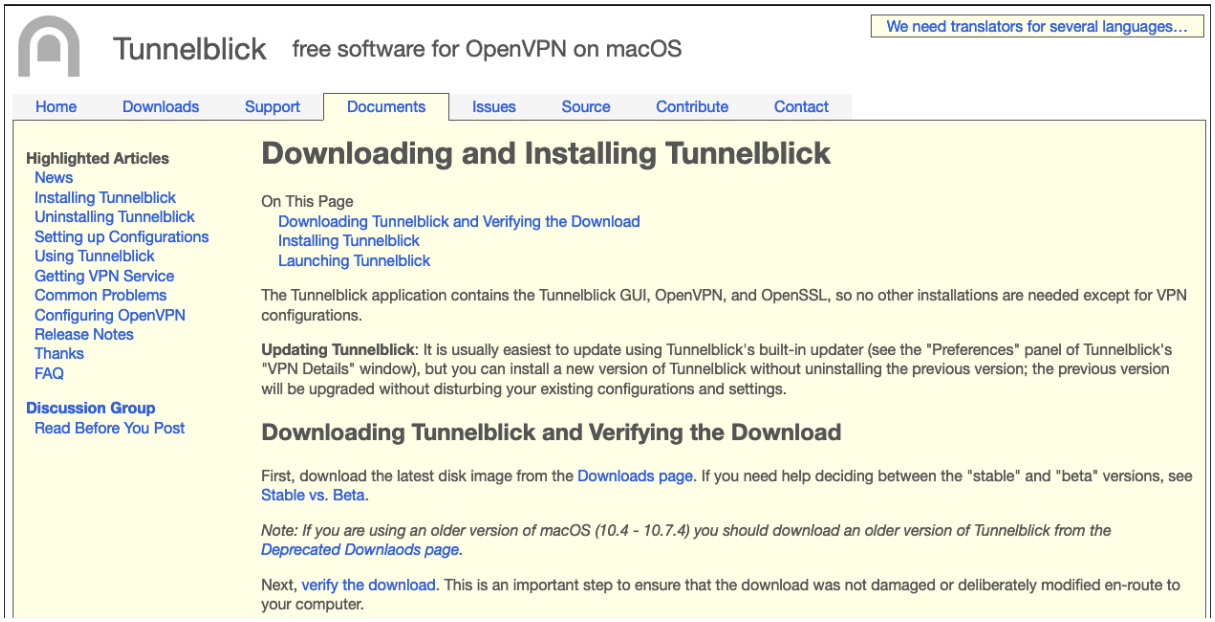
Figure 3-18: DOWNLOAD OVPN FILE



3. Download the appropriate version of Tunnelblick for your Machine from this URL
   - If you are on a new version of MacOS you need to use a patched version of Tunnelblick as MacOS no longer allows kext loading. That version is available from https://github.com/sbingner/Tunnelblick/releases/download/v3.8.8beta03%2Biouserethernet1/Tunnelblick.dmg
4. Navigate to this URL and follow the instructions to install Tunnelblick.

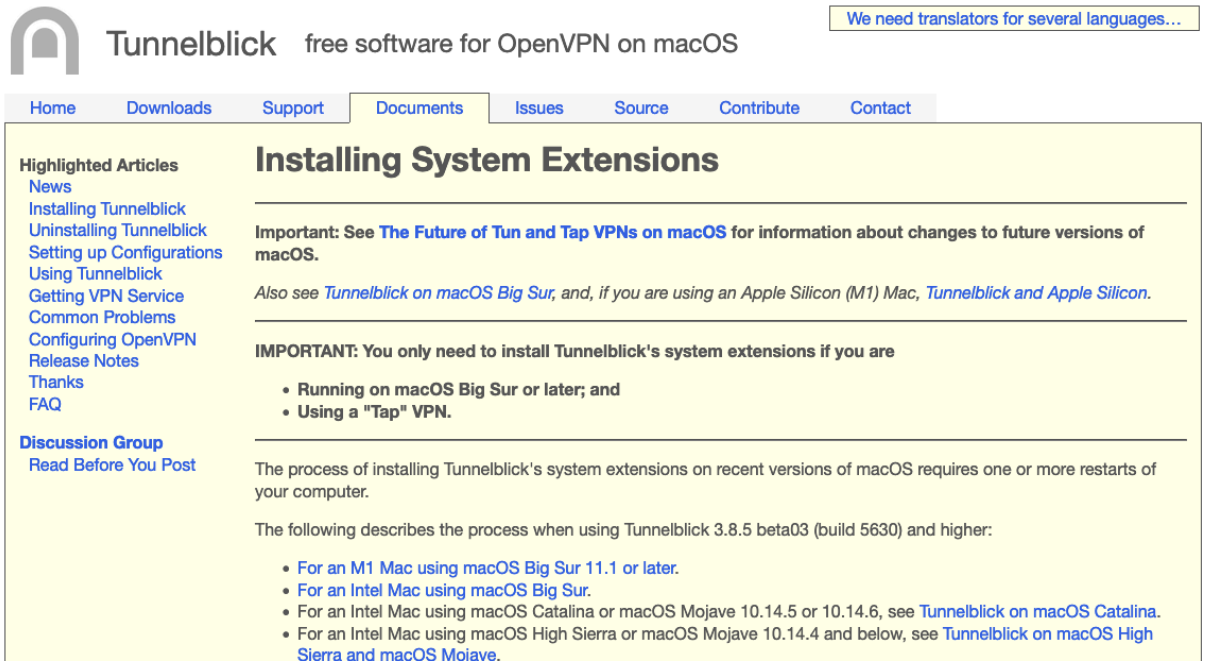
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-19: Downloading and installing Tunnelblick



5. Navigate to this URL to install the system extensions for Tunnelblick.

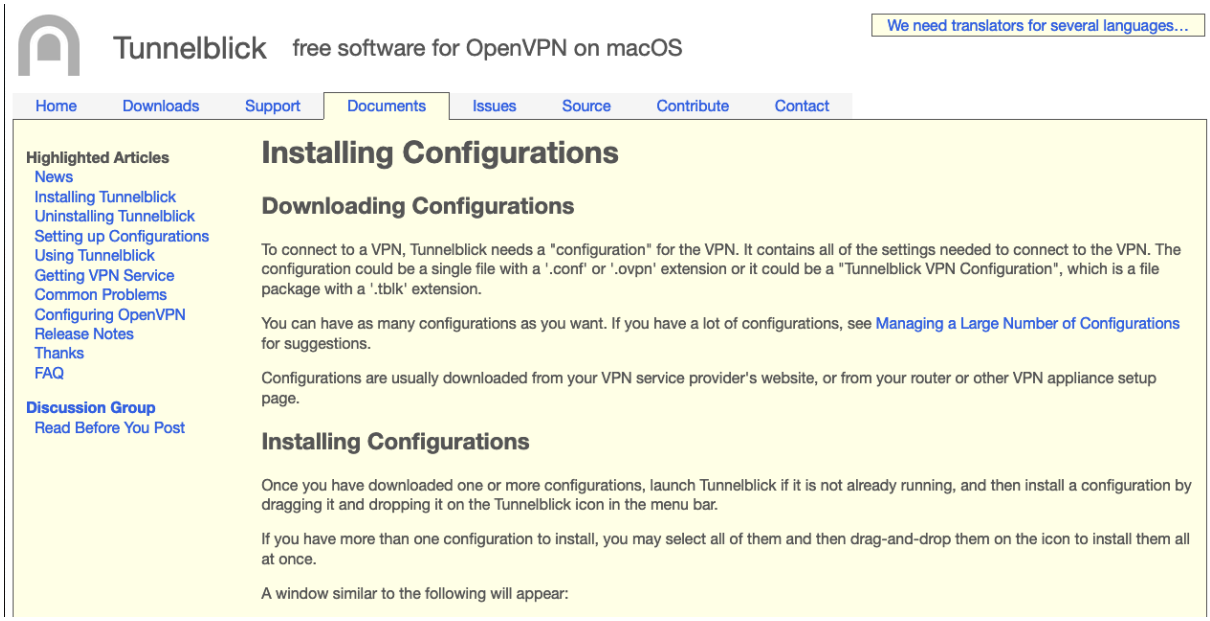
Figure 3-20: Install System Extensions



6. Add the VPN Configuration file to Tunnelblick by dragging the file over the Tunnelblick Application Icon. Navigate to this URL for detailed instructions.

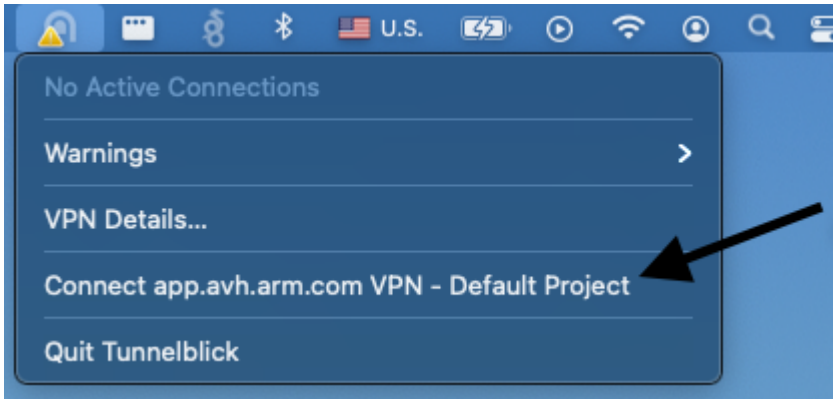
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-21: Installing Configurations



7. Click the TunnelBlick icon, then Connect ....

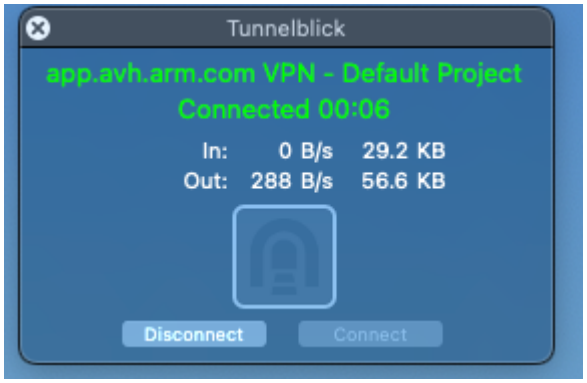
Figure 3-22: Connect



8. The text will turn green and you are now connected to the VPN!

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-23: Connected message

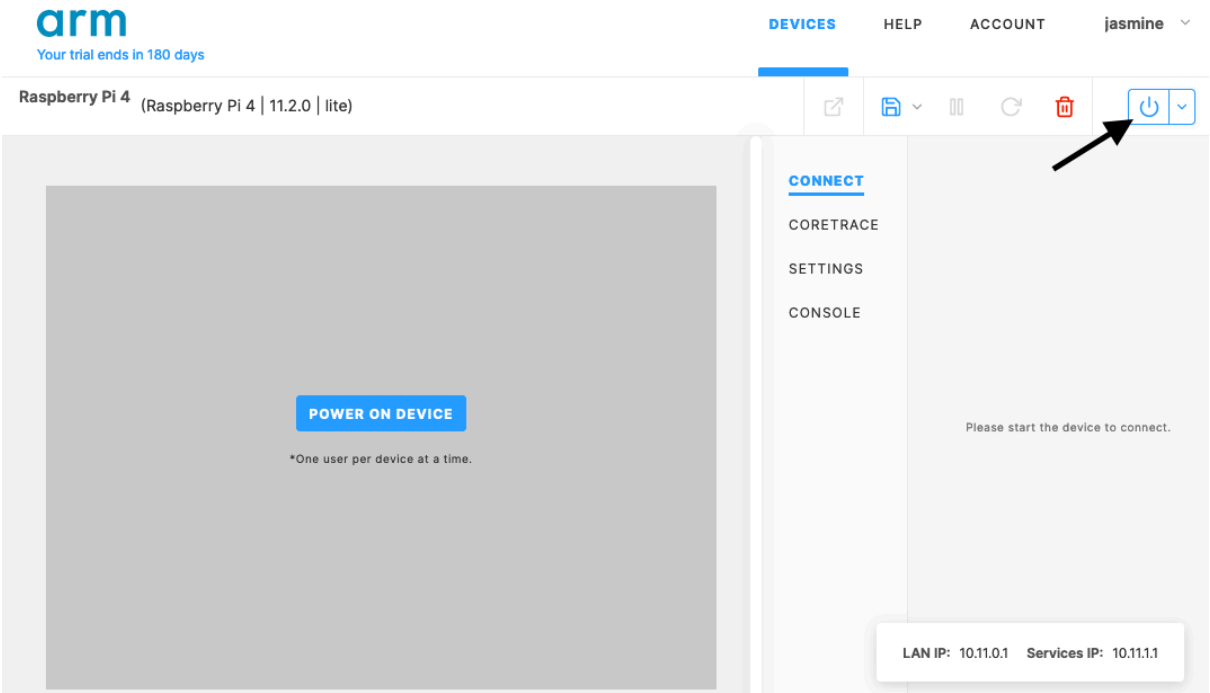


3.3.3 How to Connect with Windows OS

To connect with Windows OS, do the following:

1. If the device is off, power it on.

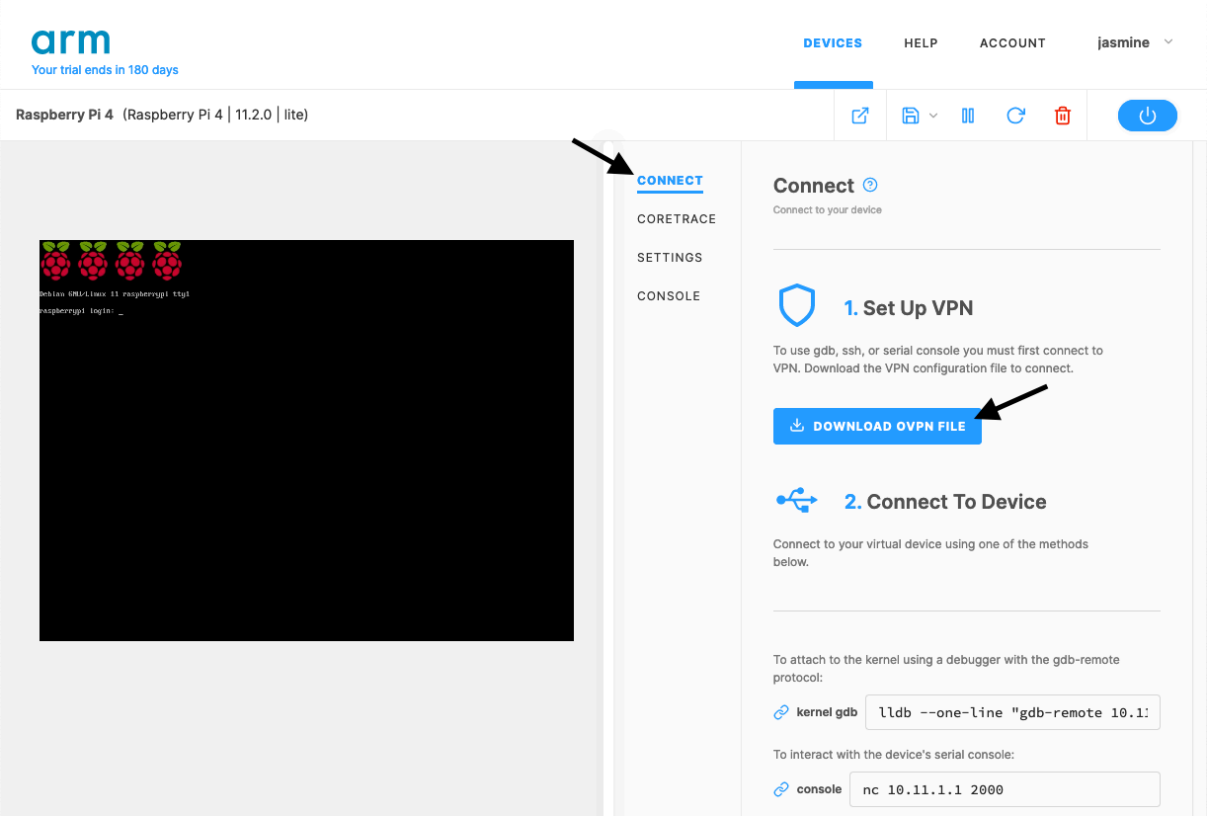
Figure 3-24: Turn on the device



2. Click CONNECT and then DOWNLOAD OVPN FILE.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

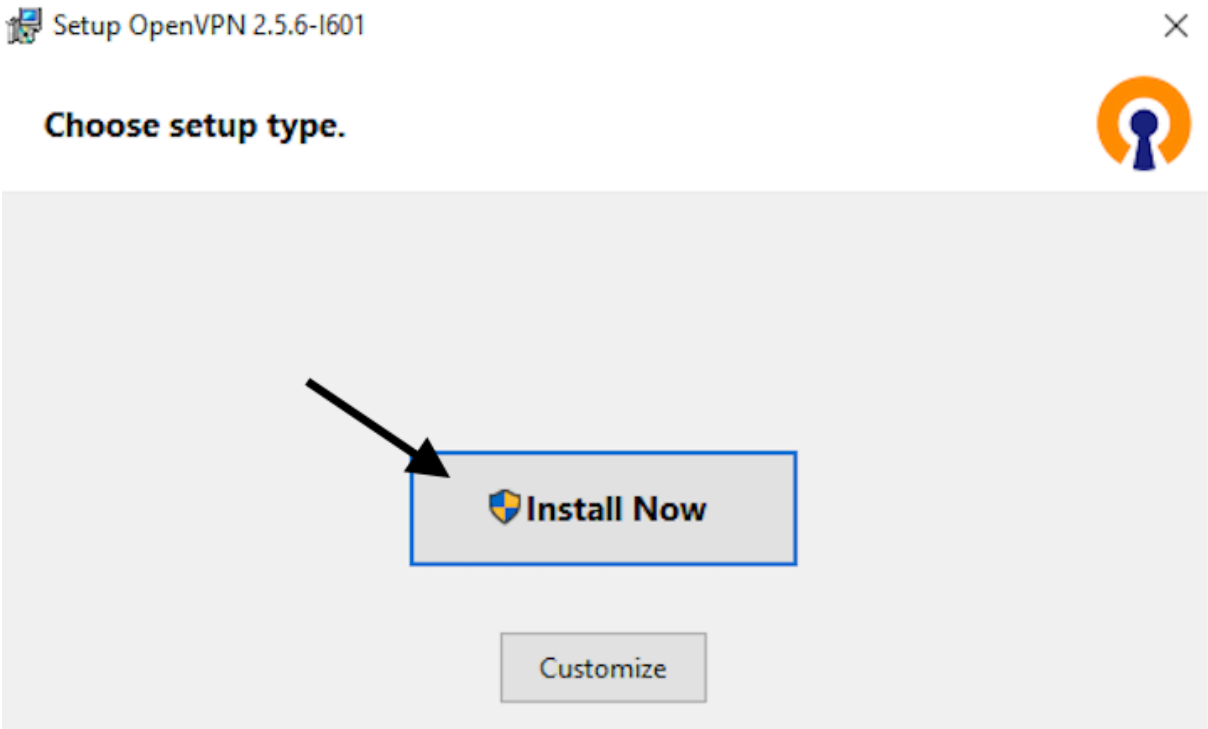
Figure 3-25: DOWNLOAD OVPN FILE



3. Install the appropriate version of OpenVPN for your machine from this URL.
4. Double click the downloaded file, then click Install Now.

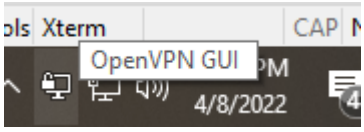
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-26: Install now



5. The OpenVPN GUI icon should appear in the taskbar by the clock.

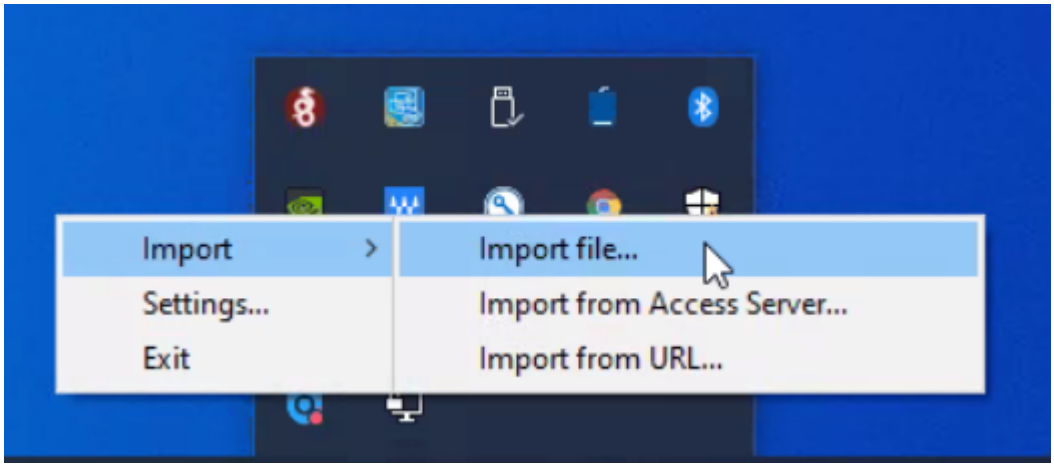
Figure 3-27: OpenVPN GUI



6. Right-click on the OpenVPN GUI icon in the taskbar and select Import → Import File.

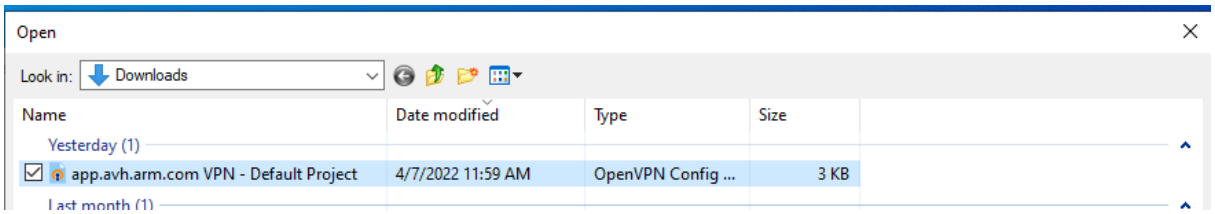
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-28: Imort file



7. Then, select the downloaded OVPN file.

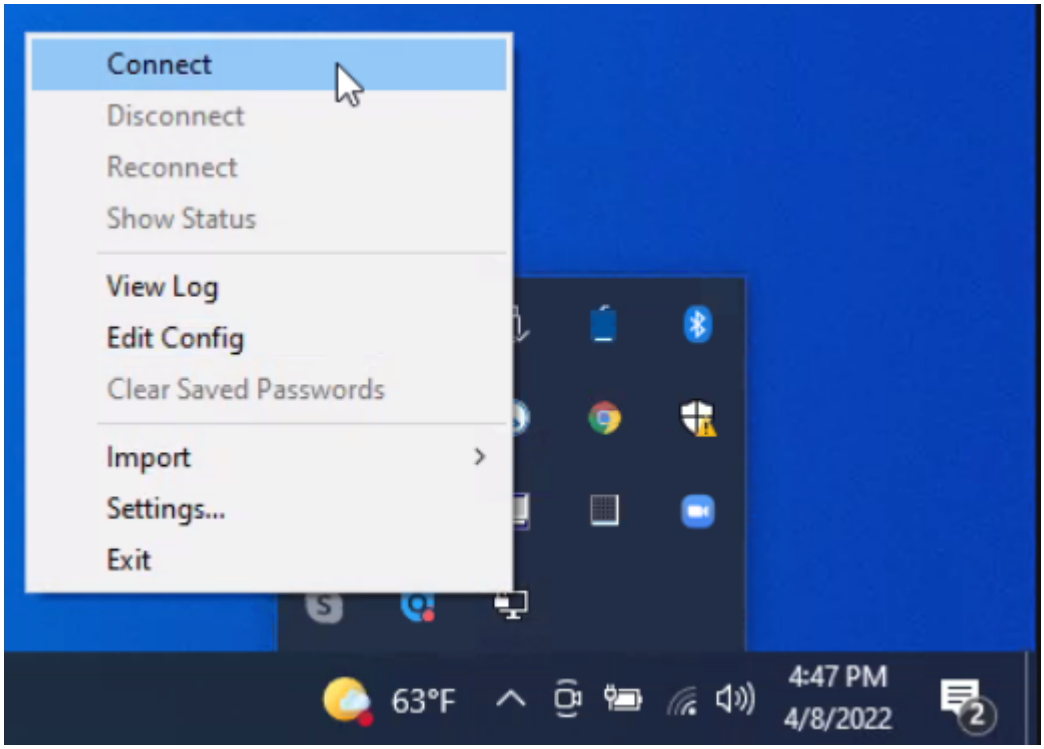
Figure 3-29: Select the OVPN file



8. Right-click on the OpenVPN GUI icon in the taskbar and select Connect.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

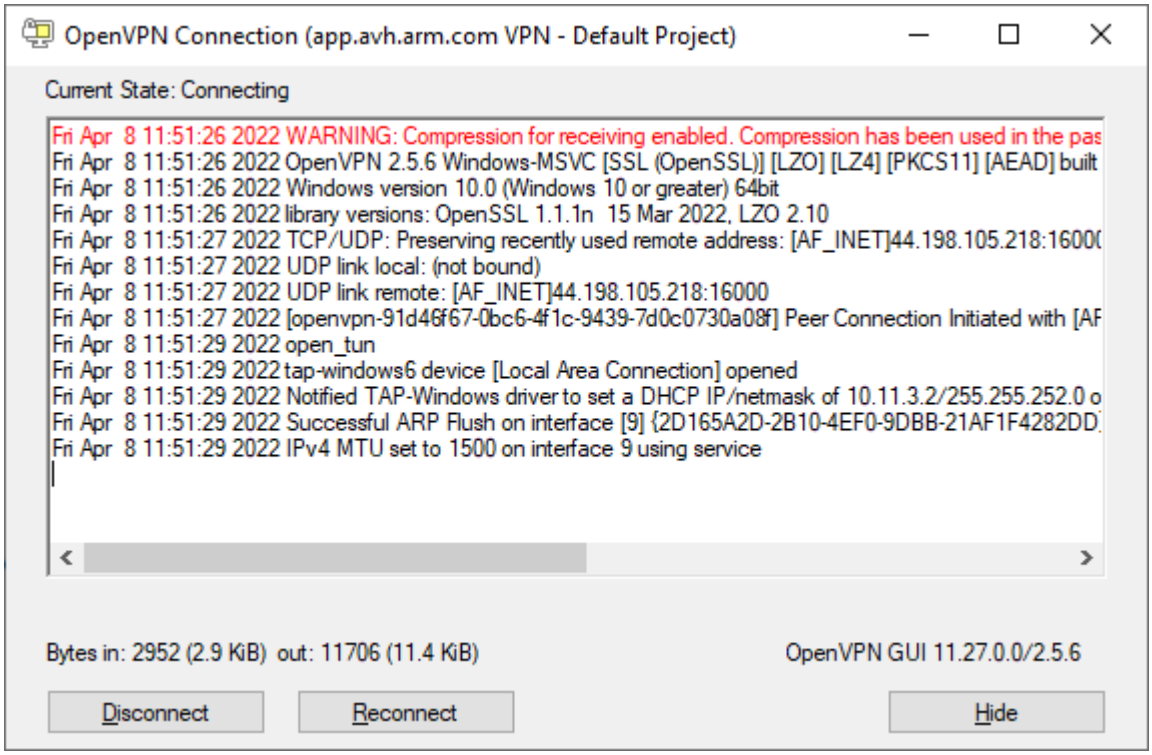
Figure 3-30: Connect



9. This window will appear and it will close when OpenVPN has connected to the device.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-31: Open VPN Connection



10. So now you have connected the VPN to your own host network you can SSH to the device or access the webserver.

Next try connecting to your local machine with SSH with the Connect via SSH tutorial.

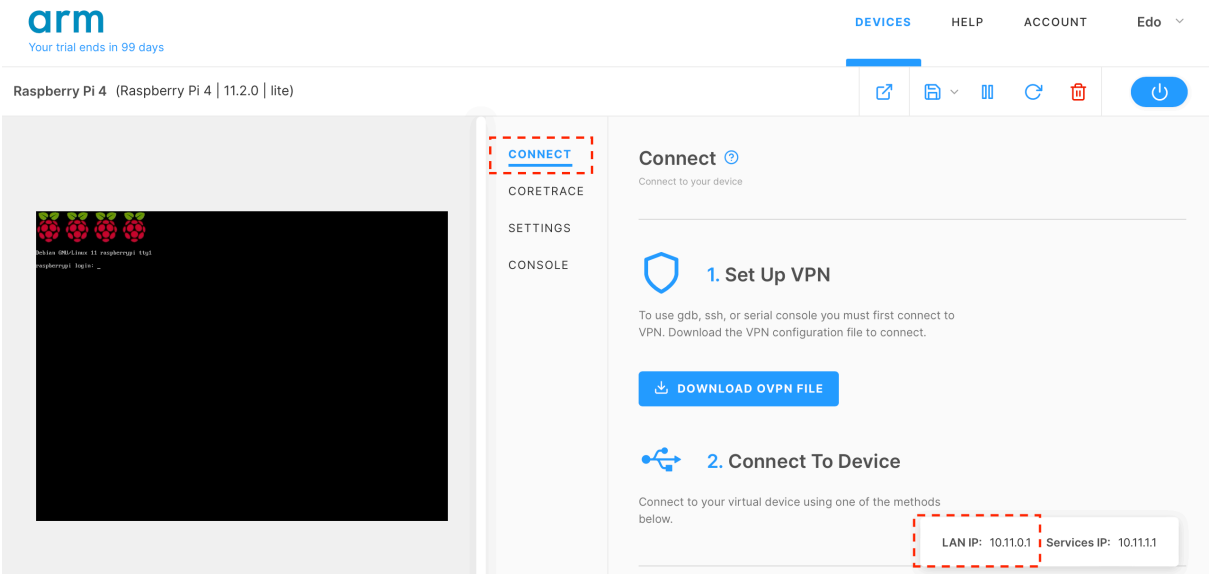
3.4 Connect via SSH

This tutorial will demonstrate how to connect a device with SSH.

1. To follow this tutorial we need a device set up with a VPN. If you haven't got a device set up, start with the Quickstart for AVH - Raspberry Pi4 tutorial then follow the Connecting to the VPN tutorial. Otherwise use a device with the VPN already set up.
2. On your device page, click on "CONNECT" and copy the "LAN IP" address from the bottom right. In this example, it is `10.11.0.1`.

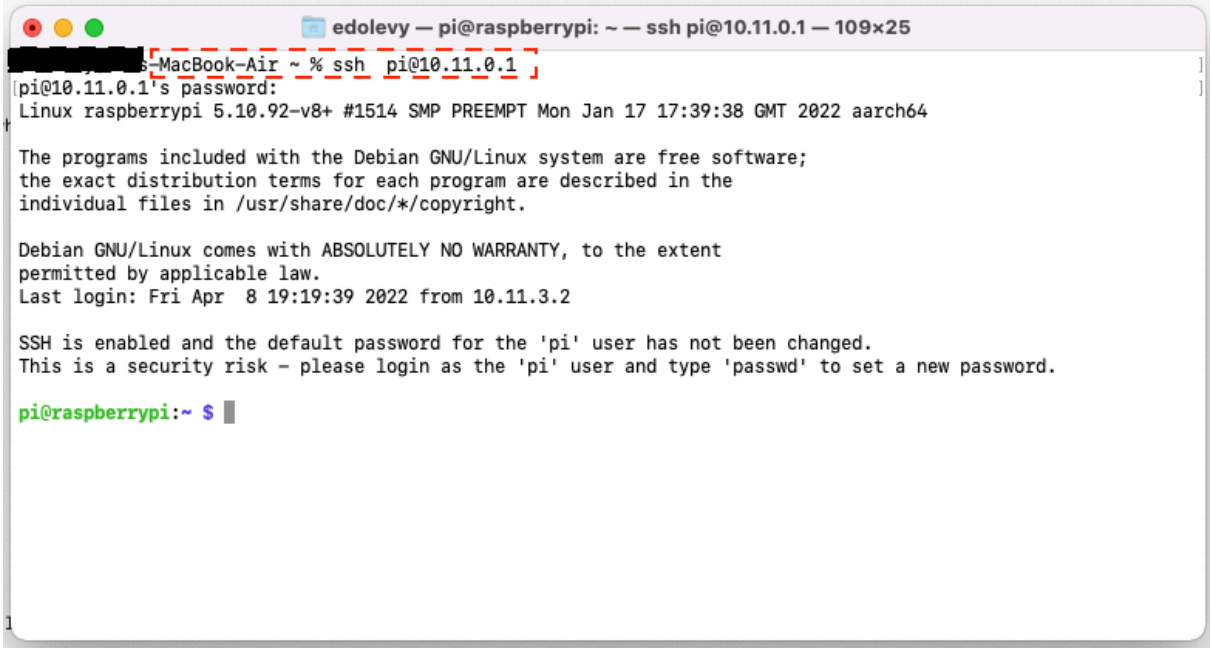
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-32: LAN IP address



3. Open a local terminal and paste the following command to connect: `ssh <Username>@<LAN IP>` In our example: `ssh pi@10.11.0.1` Then use your password to complete the connection, in this example the password is `raspberry`

Figure 3-33: Connect to SSH



4. You have now connected your virtual device to SSH.

Next, try Connect to the Console.

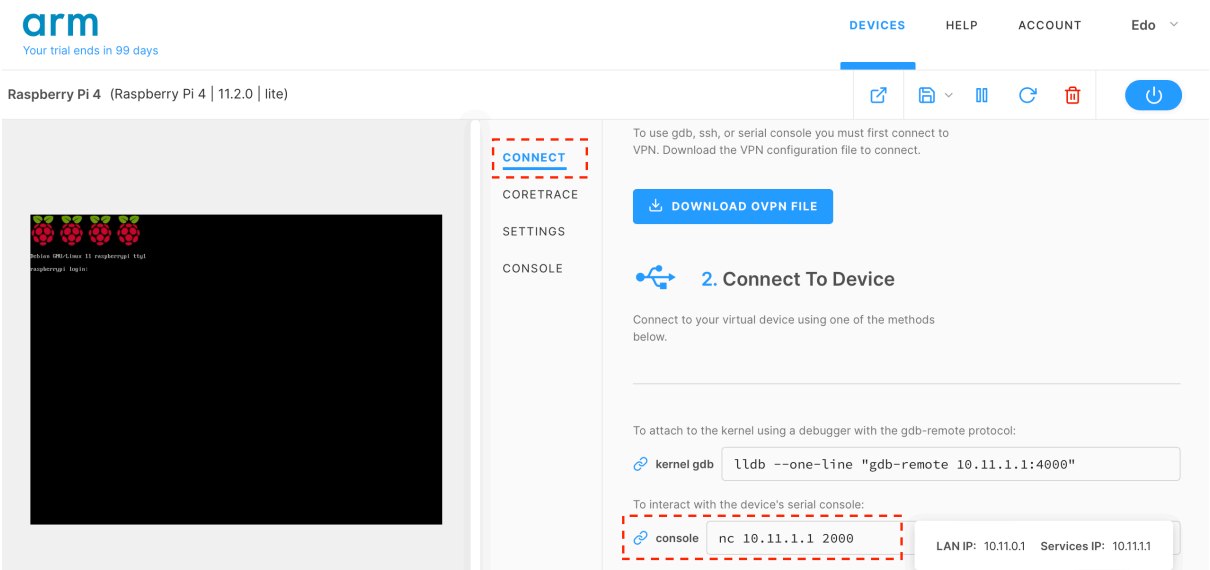
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3.5 Connect to the Console

This tutorial will demonstrate how to connect to the console with a device with the VPN setup.

1. To follow this tutorial we need a device set up with a VPN. If you haven't created a device yet, start with the Quickstart for AVH - Raspberry Pi4 tutorial then follow the Connecting to the VPN tutorial. Otherwise use a device with the VPN already set up.
2. On your device page, click on CONNECT copy the `console command: nc <device IP> <port>` in the example below it is `nc 10.11.1.1 2000`

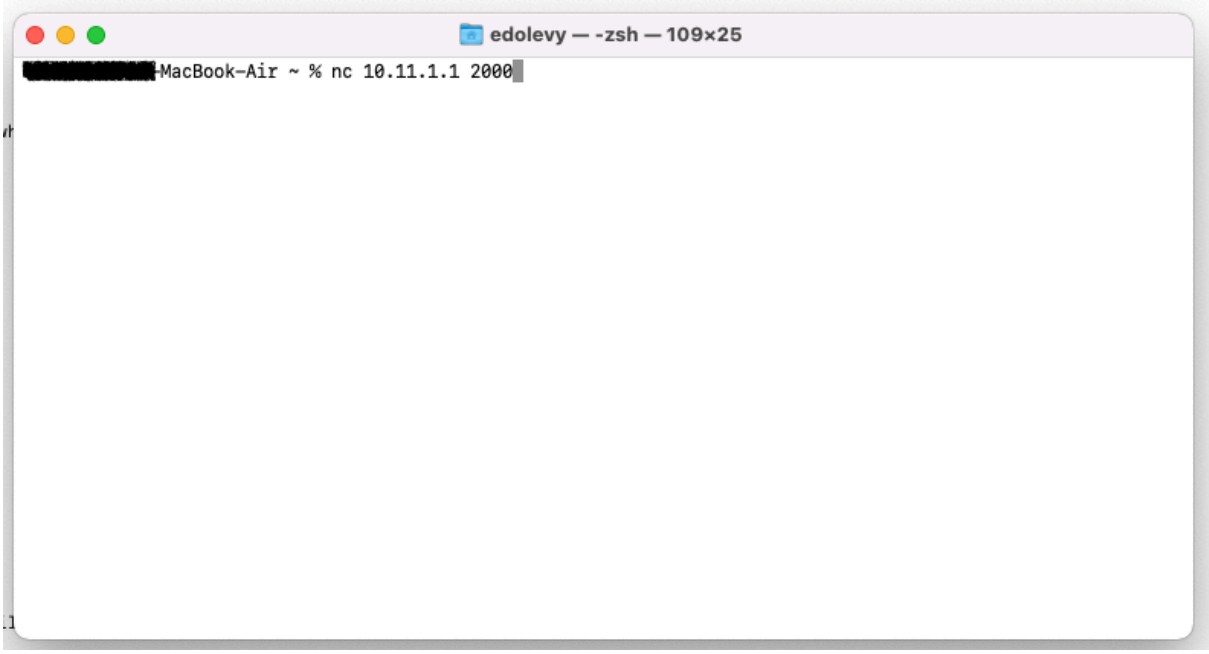
Figure 3-34: Connect



3. Paste the command into your shell (or shell application) and hit `return`.

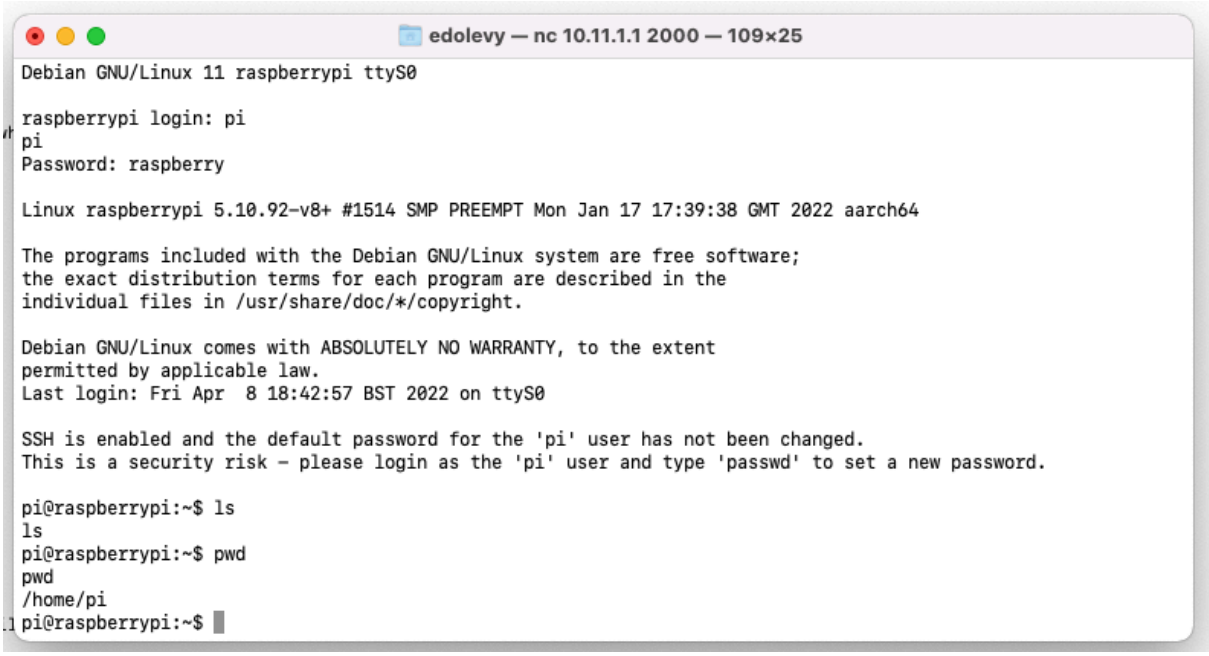
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-35: Paste the command



4. The first time logging (and after reboot) in you will need to enter the login credentials. In this example, we are using a Raspberry Pi4 device with the default credentials. User: `pi` Password: `raspberry`
5. You are now logged in to the device's console!

Figure 3-36: Console



Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3.5.1 Next Steps

You can explore the AVH examples within this knowledgebase.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential