



StarFive  
赛昉科技

# Run Docker on VisionFive 2

Application Note

Version: 1.1

Date: 2023/06/05

Doc ID: VisionFive 2-ANEN-008

# Legal Statements

Important legal notice before reading this documentation.

## PROPRIETARY NOTICE

Copyright©Shanghai StarFive Technology Co., Ltd., 2023. All rights reserved.

Information in this document is provided "as is," with all faults. Contents may be periodically updated or revised due to the product development. Shanghai StarFive Technology Co., Ltd.(hereinafter "StarFive") reserves the right to make changes without further notice to any products herein.

StarFive expressly disclaims all warranties, representations, and conditions of any kind, whether express or implied, including, but not limited to, the implied warranties or conditions of merchantability, fitness for a particular purpose and non-infringement.

StarFive does not assume any liability rising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages.

All material appearing in this document is protected by copyright and is the property of StarFive. You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. StarFive authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services.

## Contact Us

Address: Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China  
Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China

Website: <http://www.starfivetech.com> <http://www.starfivetech.com>

Email: [sales@starfivetech.com](mailto:sales@starfivetech.com)(sales) , [support@starfivetech.com](mailto:support@starfivetech.com)(support)

# Preface

About this guide and technical support information.

## About this document

This application note provides steps to run Docker on VisionFive 2.






## Version History

Table 0-1 Version History

Version	Released	Revision
1.1	2023/06/05	Updated the steps in <a href="#">Building Kernel Image (on page 8)</a> .
1.0	2023/05/11	The first official release.

## Notes and notices

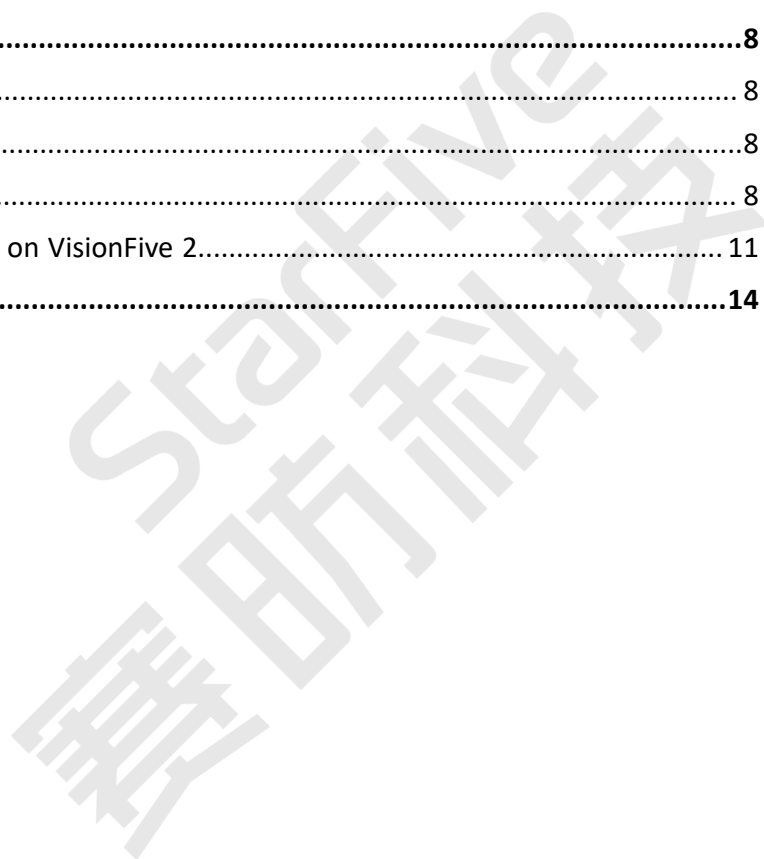
The following notes and notices might appear in this guide:

-  **Tip:**  
Suggests how to apply the information in a topic or step.
-  **Note:**  
Explains a special case or expands on an important point.
-  **Important:**  
Points out critical information concerning a topic or step.
-  **CAUTION:**  
Indicates that an action or step can cause loss of data, security problems, or performance issues.
-  **Warning:**  
Indicates that an action or step can result in physical harm or cause damage to hardware.

---

# Contents

List of Tables.....	5
List of Figures.....	6
Legal Statements.....	ii
Preface.....	iii
<b>1. Introduction.....</b>	<b>7</b>
<b>2. Preparation.....</b>	<b>8</b>
2.1. Preparing Hardware.....	8
2.2. Prepare Kernel Image.....	8
2.2.1. Building Kernel Image.....	8
2.2.2. Updating Kernel Image on VisionFive 2.....	11
<b>3. Boot Docker on VisionFive 2.....</b>	<b>14</b>



# List of Tables

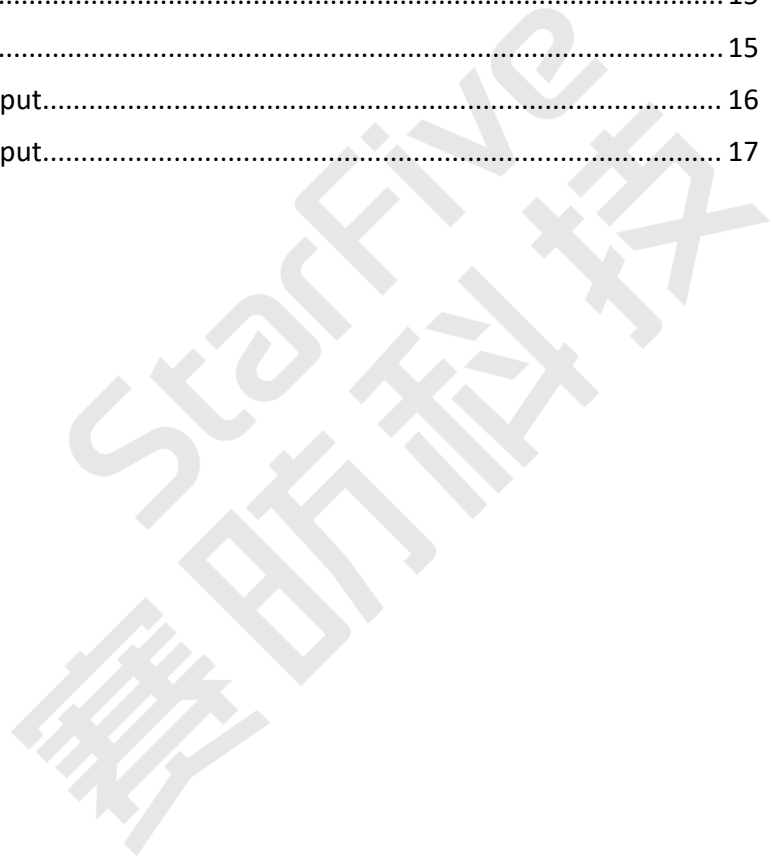
Table 0-1 Version History..... iii

Table 2-1 Hardware Preparation..... 8



# List of Figures

Figure 2-1 Example Running Process.....	10
Figure 2-2 VisionFive 2 Startup Output Example.....	11
Figure 2-3 Example Output.....	11
Figure 3-1 Example Output.....	14
Figure 3-2 Example Output.....	15
Figure 3-3 Example Output.....	15
Figure 3-4 Example Command and Output.....	16
Figure 3-5 Example Command and Output.....	17



---

# 1. Introduction

Docker is an open-source application container engine. Developers can use it to package applications and dependency packages into a light and portable container, and then publish them to any version of Linux devices. It will make the devices to utilize system resources more efficiently and ensure a consistent operating environment. It not only enables continuous delivery and deployment, but also enables easier migration, maintenance and expansion in the later stage.

This application note provides steps to run Docker on VisionFive 2.



## 2. Preparation

Make sure you have prepared the following before running Docker.

1. [Preparing Hardware \(on page 8\)](#)
2. [Prepare Kernel Image \(on page 8\)](#)

### 2.1. Preparing Hardware

Before executing the demo program, make sure you prepare the following:

Table 2-1 Hardware Preparation

Type	M/O*	Item	Notes
General	M	VisionFive 2 Board	-
General	M	<ul style="list-style-type: none"><li>• 32 GB (or more) micro-SD card</li><li>• Micro-SD card reader</li><li>• Computer (Windows/Mac OS/Linux)</li><li>• USB to serial converter (3.3 V I/O)</li><li>• Ethernet cable</li><li>• Power adapter (5 V / 3 A)</li><li>• USB Type-C Cable</li></ul>	These items are used for flashing Debian OS into a Micro-SD card.



**Note:**

\*: M: Mandatory, O: Optional

### 2.2. Prepare Kernel Image

Perform the following steps to prepare the kernel image:

1. [Building Kernel Image \(on page 8\)](#)
2. [Updating Kernel Image on VisionFive 2 \(on page 11\)](#)

#### 2.2.1. Building Kernel Image

Perform the following steps to prepare the kernel image:



## 1. Pull the kernel code:

```
$ git clone https://github.com/starfive-tech/VisionFive2.git
```

- For more instructions, please refer to <https://github.com/starfive-tech/VisionFive2>.
- The Debian image version used in this step is starfive-jh7110-VF2\_515\_v2.5.0-69.

Link: <https://debian.starfivetech.com/>

The corresponding software version is VisionFive2 Software v2.5.0.

Link: [https://github.com/starfive-tech/VisionFive2/releases/tag/VF2\\_v2.5.0](https://github.com/starfive-tech/VisionFive2/releases/tag/VF2_v2.5.0)

## 2. Compile code:

- a. Enter the following directory:

```
/VisionFive2/linux/arch/riscv/configs
```

- b. Use the following file to replace starfive\_visionfive2\_defconfig.

[docker\\_config\\_20230215](#)



**Note:**

To download this file, you need to register or login to [RVspace](#).

- c. Run `./build_kernel_vf2.sh` script under `/VisionFive2/work/linux` directory.

**Note:**

- Script download address: [build\\_kernel\\_vf2.sh](#)

**Note:**

To download this file, you need to register or login to [RVspace](#).

- You need to change the target path of the script file to your own file path.

**Figure 2-1 Example Running Process**

```

yzx@ubuntu: ~/work/VisionFive2/linux
File Edit View Search Terminal Help
#!/bin/bash
CORES=$(nproc)
declare -x INSTALL_PATH=/home/yzx/work/VisionFive2/work/boot/
declare -x INSTALL_MOD_PATH=/home/yzx/work/VisionFive2/work/boot/
declare -x LOCALVERSION="-starfive"
declare -x CROSS_COMPILE=/home/yzx/work/VisionFive2/work/buildroot_initramfs/host/bin/riscv64-buildroot-linux-gnu-
declare -x ARCH=riscv
touch .scmversion
make starfive_visionfive2_defconfig
make olddefconfig nice
make -j "${CORES}" || exit 1
mkdir -p ${INSTALL_PATH}
make zinstall modules_install || exit 2
ls -l ${INSTALL_PATH}
echo "Kernel installed in ${INSTALL_PATH} and modules in ${INSTALL_MOD_PATH}/lib/modules/"
make -j "${CORES}" bindeb-pkg || exit 3

```

**3. Confirm the generated files after compilation:**

- Under `./VisionFive2/work/boot` path:
  - `config-5.15.0-starfive-dirty`
  - `System.map-5.15.0-starfive-dirty`
  - `vmlinuz-5.15.0-starfive-dirty`

**Note:**

To simplify the later operations, rename the generated files as the following:

- `config-5.15.0-starfive`
- `System.map-5.15.0-starfive`
- `vmlinuz-5.15.0-starfive`
- Under `./VisionFive2/work` path:
  - `linux-headers-5.15.0-starfive_5.15.0-starfive-1_riscv64.deb`
  - `linux-image-5.15.0-starfive_5.15.0-starfive-1_riscv64.deb`
  - `linux-libc-dev_5.15.0-starfive-1_riscv64.deb`

## 2.2.2. Updating Kernel Image on VisionFive 2

Perform the following steps to update the kernel image on VisionFive 2:

1. Flashing Debian image to SD card, it is mandatory to start VisionFive 2 normally once.

**i** **Tip:**

- Flash Debian OS into a Micro-SD card as described in the *Flashing OS to a Micro-SD Card* section in the [VisionFive 2 Single Board Computer Quick Start Guide](#).
- The Debian image version used in this step is `starfive-jh7110-VF2_515_v2.5.0-69`.

Link: <https://debian.starfivetech.com/>

The corresponding software version is VisionFive2 Software v2.5.0.

Link: [https://github.com/starfive-tech/VisionFive2/releases/tag/VF2\\_v2.5.0](https://github.com/starfive-tech/VisionFive2/releases/tag/VF2_v2.5.0)

Figure 2-2 VisionFive 2 Startup Output Example

```

COM4 - Tera Term VT
文件(F) 编辑(E) 设置(S) 控制(O) 窗口(W) 帮助(H)

starfive login: root
Password: [ 34.486286] mipi_0p9: disabling

Linux starfive 5.15.0-starfive #1 SMP Mon Dec 19 07:56:37 EST 2022 riscv64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@starfive:~# [ 42.208911] systemd-journald[250]: Oldest entry in /var/log/journal/bd887d4196a5463aa4c948eb57286362/system.journal is older than the configured file retention duration (1month), suggesting rotation.
[ 42.227798] systemd-journald[250]: /var/log/journal/bd887d4196a5463aa4c948eb57286362/system.journal: Journal header limits reached or header out-of-date, rotating.
root@starfive:~# █

```

2. Insert an SD card and perform mount operations under Ubuntu development environment:
  - a. Perform the following command to list devices:

```
fdisk -l
```

Figure 2-3 Example Output

```

Device      Start      End      Sectors  Size Type
/dev/sdc1   2048       34815   32768    16M Linux filesystem
/dev/sdc2   34816     239615  204800   100M EFI System
/dev/sdc3   239616  124669918 124430303 59.3G Linux filesystem
yzzx@ubuntu:~$

```

b. Execute the following command to mount:

```
sudo mount /dev/sdc2 /media/<Username>/root/boot
```

**Example command:**

```
sudo mount /dev/sdc2 /media/yzx/root/boot
```

3. Execute the following command and copy the generated kernel files to /media/<Username>/root/boot/boot:

```
sudo cp boot/vmlinuz-5.15.0-starfive /media/<Username>/root/boot/boot
&& sync
sudo cp boot/config-5.15.0-starfive /media/<Username>/root/boot/boot &&
sync
sudo cp
boot/System.map-5.15.0-starfive /media/<Username>/root/boot/boot &&
sync
```

4. Execute the following command and copy the generated deb image packages to /media/<Username>/root/usr:

```
sudo cp
linux-image-5.15.0-starfive_5.15.0-starfive-1_riscv64.deb /
media/<<Username>/ root/usr/ && sync
sudo cp
linux-libc-dev_5.15.0-starfive-1_riscv64.deb /media/<Username>/
root/usr/ && sync
sudo cp
linux-headers-5.15.0-starfive_5.15.0-starfive-1_riscv64.deb /
media/<Username>/root/usr/ && sync
```

5. Download Docker offline install package and copy it to /media/<username>/root/usr.



**Tip:**

Install package download path: [https://github.com/carlosedp/riscv-bringup/releases/download/v1.0/docker-v20.10.2-dev\\_riscv64.deb](https://github.com/carlosedp/riscv-bringup/releases/download/v1.0/docker-v20.10.2-dev_riscv64.deb)

6. Insert the SD card on VisionFive 2 and restart it to enter the emergency mode. Then enter the password `starfive` to login.

7. Unplug the SD card first and then reinsert it on VisionFive 2. After successful login, enter the /usr directory to install the deb package.



**Note:**

To install the package, execute the following commands in order.

```
dpkg -i linux-headers-5.15.0-starfive_5.15.0-starfive-1_riscv64.deb  
dpkg -i linux-libc-dev_5.15.0-starfive-1_riscv64.deb  
dpkg -i linux-image-5.15.0-starfive_5.15.0-starfive-1_riscv64.deb
```

8. Restart the system to enter the normal login interface, and enter the account and password.
9. Execute the following commands to install Docker related dependency packages and offline packages:

```
apt install libip6tc2 libyajl2 contrack ebtables ethtool iptables  
socat libyajl-dev  
dpkg -i docker-v20.10.2-dev_riscv64.deb
```

## 3. Boot Docker on VisionFive 2

Perform the following steps to boot Docker on VisionFive 2:

1. Execute the following commands in order:

```
dockerd
systemctl start docker
```

2. Execute the following command:

```
systemctl status docker
```

**Figure 3-1 Example Output**

```
root@starfive:/usr# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/etc/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Tue 2023-04-11 08:47:30 UTC; 12s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 7132 (dockerd)
      Tasks: 9
     Memory: 18.9M
           CPU: 1.414s
    CGroup: /system.slice/docker.service
            └─7132 /usr/local/bin/dockerd -H fd:// --containerd=/run/containerd
```



**Note:**

If the output results are not enabled, Active: active (running), StarFive recommends you to restart the system and execute this step again.

3. Execute the following command:

```
docker info
```

Figure 3-2 Example Output

```

root@starfive:/usr# docker info
Client:
 Context:      default
 Debug Mode:  false

Server:
 Containers:  0
  Running:    0
  Paused:     0
  Stopped:    0
 Images:      0
 Server Version:  dev
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: false
 userxattr: false
 Logging Driver: json-file
 Cgroup Driver: systemd
 Cgroup Version: 2

```

4. Execute the following command to run hello-world:

```
docker run --rm hello-world
```

Figure 3-3 Example Output

```

root@starfive:/usr# docker run --rm hello-world
[ 1460.997698] docker0: port 1(veth69587cd) entered blocking state
[ 1461.003723] docker0: port 1(veth69587cd) entered disabled state
[ 1461.009999] device veth69587cd entered promiscuous mode
[ 1462.529926] eth0: renamed from veth8822f9d
[ 1462.619834] docker0: port 1(veth69587cd) entered blocking state
[ 1462.625792] docker0: port 1(veth69587cd) entered forwarding state

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (riscv64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

```

5. Download Alpine Docker image of the RISC-V version and run the following:

```

docker pull riscv64/alpine:edge
docker run -it alpine:edge

```

**Figure 3-4 Example Command and Output**

```

root@starfive:/usr# docker pull riscv64/alpine:edge
edge: Pulling from riscv64/alpine
1c9566da74e4: Pull complete
Digest: sha256:3894e4e3ea0345d0627776199362bf3e68a057a2786b253d8588cf68220f7de3
Status: Downloaded newer image for riscv64/alpine:edge
docker.io/riscv64/alpine:edge
root@starfive:/usr# docker run -it alpine:edge
Unable to find image 'alpine:edge' locally
edge: Pulling from library/alpine
Digest: sha256:2d01a16bab53a8405876cec4c27235d47455a7b72b75334c614f2fb0968b3f90
Status: Downloaded newer image for alpine:edge
[ 1582.673732] docker0: port 1(veth6116cd8) entered blocking state
[ 1582.679750] docker0: port 1(veth6116cd8) entered disabled state
[ 1582.686311] device veth6116cd8 entered promiscuous mode
[ 1584.340633] eth0: renamed from veth31f1453
[ 1584.390567] docker0: port 1(veth6116cd8) entered blocking state
[ 1584.396526] docker0: port 1(veth6116cd8) entered forwarding state
/ # ls
bin      etc      lib      mnt      proc     run      srv      tmp      var
dev      home    media    opt      root    /sbin    sys      usr
/ #

```

6. Execute the following commands in Docker image to verify:

```

cat /etc/os-release
ifconfig
arp -a
ping <Website>

```

**Tip:**

Before executing the commands, you need to replace <Website> with any URL.



**Figure 3-5 Example Command and Output**

```
/ # cat /etc/os-release
NAME="Alpine Linux"
ID=alpine
VERSION_ID=3.18_alpha20230329
PRETTY_NAME="Alpine Linux edge"
HOME_URL="https://alpinelinux.org/"
BUG_REPORT_URL="https://gitlab.alpinelinux.org/alpine/aports/-/issues"
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:AC:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/ # arp -a
/ # ping www.baidu.com
PING www.baidu.com (180.101.50.242): 56 data bytes
64 bytes from 180.101.50.242: seq=0 ttl=51 time=7.881 ms
64 bytes from 180.101.50.242: seq=1 ttl=51 time=7.536 ms
```

As shown in the above output, you have successfully run Docker on VisionFive 2!