

Run Docker on VisionFive 2

Application Note Version: 1.2 Date: 2024/12/05 Doc ID: VisionFive 2-ANEN-008

Legal Statements

Important legal notice before reading this documentation.

PROPRIETARY NOTICE

Copyright©Guangdong StarFive Technology Co., Ltd., 2024. 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. Guangdong 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. StarFiveauthorizes 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 S201, Zone A, No. 2, Haoyang Road, Yunlu Community, Daliang Subdistrict, Shunde District, Foshan, Guangdong, China, 528300

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

Email: sales@starfivetech.com(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	
1.2	2024/12/05	Added the following sections:	
		• Environment Requirements (on page 8)	
		• Preparing Software (on page 8)	
		• Compile and Replace Kernel (on page 9)	
		• Install Docker <i>(on page 14)</i>	
		Deleted Prepare Kernel Image chapter.	
1.1	2023/06/05	Updated the steps in <i>Building Kernel Image</i> .	
1.0	2023/05/11	The first official release.	

Notes and notices

1

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 Tables5				
List of Figures				
Legal Statements2				
Preface				
1. Introduction				
2. Preparation				
2.1. Environment Requirements				
2.2. Preparing Hardware				
2.3. Preparing Software				
2.3.1. Flash Image				
2.3.2. Compile and Replace Kernel9				
2.3.3. Install Docker14				
3. Boot Docker on VisionFive 2				

List of Tables

Table 0-1 Version History	;
Table 2-1 Hardware Preparation	;

List of Figures

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.

- Preparing Hardware (on page 8)
- Environment Requirements (on page 8)
- Preparing Software (on page 8)

2.1. Environment Requirements

The environment requirements are as follows:

- Linux Kernel: Linux 5.15
- OS: Debian 12
- SBC: VisionFive 2
- SoC: JH-7110

2.2. Preparing Hardware

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

Туре	M/0*	Item	Notes
General	М	VisionFive 2 Board	-
General	M	 32 GB (or more) micro-SD card Micro-SD card reader Computer (Windows/Mac OS/Linux) USB to serial converter (3.3 V I/O) Ethernet cable Power adapter (5 V / 3 A) USB Type-C Cable 	These items are used for flashing Debian OS into a Micro-SD card.

Table 2-1 Hardware Preparation

Note: *: M: Mandatory, O: Optional

2.3. Preparing Software

This section mainly introduces the required software operations from the following 3 aspects:

- Flash Image (on page 8)
- Compile and Replace Kernel (on page 9)
- Install Docker (on page 14)

2.3.1. Flash Image

Confirm to proceed with the following steps:

- 1. According to the "Flashing OS to a Micro-SD Card" chapter in <u>VisionFive 2 Single Board Computer Quick Start Guide</u>, flash the Debian OS to the Micro-SD card.
- 2. Log in to Debian and ensure that VisionFive 2 is connected to the internet. For detailed instructions, please refer to the "Using SSH over Ethernet" or "Using USB to Serial Converter" chapter in <u>VisionFive 2 Single Board Computer Quick Start</u> <u>Guide</u>.
- 3. To expand partitions on Debian, please refer to the "*Extend Partition on SD Card or eMMC*" chapter in <u>VisionFive 2</u> <u>Single Board Computer Quick Start Guide</u>.

2.3.2. Compile and Replace Kernel

StarFivedefault Debian kernel does not support CGROUP, BRIDGE and other functions required by Docker. If the relevant configurations in the kernel are not turned on, various errors may occur when installing or running Docker.

2.3.2.1. Required Kernel Options

The Docker open source team has provided a script that can detect whether the kernel configuration meets Docker's running requirements. The source address is: <u>https://github.com/moby/moby/blob/master/contrib/check-config.sh</u>.

1. Get script and run the following command:

\$./check-config.sh

The following is the example output:

Figure 2-1 Example Output:

user@starfive:~\$./check-config.sh info: reading kernel config from /proc/config.gz ... Generally Necessary: cgroup hierarchy: cgroupv2 Controllers: - cpu: available cpuset: missing - io: missing - memory: missing pids: missing CONFIG_NAMESPACES: enabled CONFIG_NET_NS: enabled CONFIG_PID_NS: enabled CONFIG_IPC_NS: enabled CONFIG_UTS_NS: enabled CONFIG_CGROUPS: enabled CONFIG_CGROUP_CPUACCT: missing CONFIG_CGROUP_DEVICE: missing CONFIG_CGROUP_FREEZER: missing CONFIG_CGROUP_SCHED: enabled CONFIG_CPUSETS: missing CONFIG_MEMCG: missing CONFIG_KEYS: enabled CONFIG_VETH: missing CONFIG_BRIDGE: missing CONFIG_BRIDGE_NETFILTER: missing CONFIG_IP_NF_FILTER: enabled CONFIG_IP_NF_MANGLE: missing CONFIG_IP_NF_TARGET_MASQUERADE: enabled CONFIG_NETFILTER_XT_MATCH_ADDRTYPE: missing CONFIG_NETFILTER_XT_MATCH_CONNTRACK: enabled CONFIG_NETFILTER_XT_MATCH_IPVS: missing CONFIG_NETFILTER_XT_MARK: enabled CONFIG_IP_NF_NAT: enabled CONFIG_NF_NAT: enabled CONFIG_POSIX_MQUEUE: enabled CONFIG_CGROUP_BPF: enabled

The output results are mainly divided into 2 parts:

- **Generally Necessary:** Indicates the necessary configuration. The missing options in the figure need to be opened in the kernel configuration to recompile and burn the kernel to support Docker.
- Optional Features: Indicates optional configuration, and can be opened as needed.
- 2. From the example output, we can see that there are many missing options in the necessary configuration. You can refer to this link to pull codes, switch branches, and apply the default kernel options. After applying the default options, configure the corresponding kernel options:

• To enable **CONFIG_CGROUP_***, **CONFIG_CPUSETS**, **CONFIG_MEMCG** and itms under **Controller**, you need to enable following options under **General setup > Control Group support**:

Figure 2-2 Configure Kernel Options

	Control Group support
[*]	Memory controller
[*]	IO controller
[*]	CPU controller>
[*]	PIDs controller
[]	RDMA controller
[*]	Freezer controller
[]	HugeTLB controller
[*]	Cpuset controller
[*]	<pre>Include legacy /proc/<pid>/cpuset file</pid></pre>
[*]	Device controller
[*]	Simple CPU accounting controller
[]	Perf controller
[*]	Support for eBPF programs attached to cgroups
[]	Misc resource controller
[]	Debug controller

 To enable CONFIG_VETH, you need to enable Virtual ethernet pair device options under Device Drivers > Network device support:

Figure 2-3 Virtual ethernet pair device

<*> Virtual ethernet pair device

 To enable CONFIG_BRIDGE, you need to enable 802.1d Ethernet Bridging option under Networking support > Networking options:

Figure 2-4 802.1d Ethernet Bridging

*> 802.1d Ethernet Bridging

 To enable CONFIG_BRIDGE_NETFILTER, you need to enable Bridged IP/ARP packets filtering option under Networking support > Networking options > Network packet filtering framework (Netfilter):

Figure 2-5 Bridged IP/ARP packets filtering

Bridged IP/ARP packets filtering

 To enable CONFIG_NETFILTER_XT_MATCH_ADDRTYPE, you need to enable "addrtype" address type match support option under Networking support > Networking options > Network packet filtering framework (Netfilter) > Core Netfilter Configuration:

Figure 2-6 "addrtype" address type match support

```
> "addrtype" address type match support
```

 To enable CONFIG_IP_NF_MANGLE, you need to enable Packet mangling option under Networking support > Networking options > Network packet filtering framework (Netfilter) > IP: Netfilter Configuration:

Figure 2-7 Packet mangling



• To enable CONFIG_NETFILTER_XT_MATCH_IPVS, you need to enable the following 2 options:

• Enable IP virtual server support option under Networking support > Networking options > Network packet filtering framework (Netfilter):

Figure 2-8 IP virtual server support & "ipvs" match support

<*> IP virtual server support --->

• Enable "ipvs" match support option under Networking support > Networking options > Network packet filtering framework (Netfilter) > Core Netfilter Configuration:

Figure 2-9 "ipvs" match support



 To enable CONFIG_IP6_NF_IPTABLES, you need to enable IP6 tables support (required for filtering) option under Networking support > Networking options > Network packet filtering framework (Netfilter) > IPV6: Netfilter Configuration:

Figure 2-10 IP6 tables support (required for filtering)

IP6 tables support (required for filtering)

 To enable CONFIG_NF_TABLES_BRIDGE, you need to enable Ethernet Bridge nf_tables support (NEW) option under Networking support > Networking options > Network packet filtering framework (Netfilter):

Figure 2-11 Ethernet Bridge nf_tables support (NEW)

> Ethernet Bridge nf_tables support (NEW) ---

After selecting the above options, you still need to enable the following options under Networking support
 Networking options > Network packet filtering framework (Netfilter) > Core Netfilter Configuration, otherwise, Docker will not start properly:

Figure 2-12 Netfilter nf_tables_support
-*- Network Address Translation support
<*> Netfilter nf_tables support
[] Netfilter nf_tables mixed IPv4/IPv6 tables support
<pre>[] Netfilter nf_tables netdev tables support</pre>
<pre>< > Netfilter nf_tables number generator module</pre>
<pre><*> Netfilter nf_tables conntrack module</pre>
<pre><*> Netfilter nf_tables counter module</pre>
<pre>< > Netfilter nf_tables connlimit module</pre>
<pre>< > Netfilter nf_tables log module</pre>
<pre>< > Netfilter nf_tables limit module</pre>
<pre>< > Netfilter nf_tables masquerade support</pre>
<pre>< > Netfilter nf_tables redirect support</pre>
<pre><*> Netfilter nf_tables nat module</pre>
<pre>< > Netfilter nf_tables tunnel module</pre>
<pre>< > Netfilter nf_tables stateful object reference module</pre>
<pre>< > Netfilter nf_tables queue module</pre>
<pre>< > Netfilter nf_tables quota module</pre>
<pre>< > Netfilter nf_tables reject support</pre>
<pre><*> Netfilter x_tables over nf_tables module</pre>
<pre>< > Netfilter nf_tables hash module</pre>
< > Netfilter nf_tables socket match support
<pre>< > Netfilter nf_tables passive OS fingerprint support</pre>
<pre>< > Netfilter nf_tables tproxy support</pre>
<pre>< > Netfilter nf_tables SYNPROXY expression support</pre>
< > Netfilter flow table module
<pre>-*- Netfilter Xtables support (required for ip_tables)</pre>
*** Xtables combined modules ***
-*- nfmark target and match support

2.3.2.2. Compile and Replace Kernel

Click on the following link to compile and replace kernel files, and modify startup items:

- <u>Compile Kernel</u>
- <u>Replace Kernel</u>

After replacing and starting the corresponding kernel, follow the method in <u>Required Kernel Options (on page 9)</u> to check Docker kernel options again:

Figure 2-13 Check Docker Kernel Options

Gé	enerally Necessary
- 00	caroup hierarchy: caroupy?
	Controllers'
	- cnu: available
	- chuset, available
	- io: available
	- memory; available
	- nids' available
_	CONFIG NAMESPACES, enabled
_	CONFIG NET NS: enabled
_	CONFIG PID NS: enabled
_	CONFIG TPC NS: enabled
_	CONFIG UTS NS: enabled
_	CONFIG_CGROUPS: enabled
_	CONFIG CGROUP CPUACCT: enabled
_	CONFIG CGROUP DEVICE: enabled
_	CONFIG CGROUP ERFEZER: enabled
_	CONFIG CGROUP SCHED: enabled
_	CONFIG CPUSETS: enabled
-	CONFIG MEMCG: enabled
_	CONFIG KEYS: enabled
_	CONFIG VETH: enabled
_	CONFIG BRIDGE: enabled
-	CONFIG BRIDGE NETFILTER: enabled
-	CONFIG IP NF FILTER: enabled
-	CONFIG_IP_NF_MANGLE: enabled
-	CONFIG_IP_NF_TARGET_MASQUERADE: enabled
-	CONFIG_NETFILTER_XT_MATCH_ADDRTYPE: enabled
-	CONFIG_NETFILTER_XT_MATCH_CONNTRACK: enable
-	CONFIG_NETFILTER_XT_MATCH_IPVS: enabled
-	CONFIG_NETFILTER_XT_MARK: enabled
-	CONFIG_IP_NF_NAT: enabled
-	CONFIG_NF_NAT: enabled
-	CONFIG_POSIX_MQUEUE: enabled
-	CONFIG_CGROUP_BPF: enabled

Result:

All kernel options that display missing in the required kernel options have been updated to enabled, indicating that all required kernel options are turned on.

2.3.3. Install Docker

Follow the steps below to install Docker:

1. Run the following command to install Docker:

\$ sudo apt install docker.io -y

2. After installation, run the following command to reboot the system:

\$ sudo reboot

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



Note:

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

3. Execute the following command:

docer info

```
Figure 3-2 Example Output
```

```
root@starfive:/usr# docker info
Client:
             default
Context:
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. Download Alpine Docker image of the RISC-V version and run the following:

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

. .

.

. . .

Figure 3-3 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
/ # 1s
bin etc lib mnt proc run srv tmp var
dev home media opt root sbin sys usr

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



1 Tip:

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

Figure 3-4 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)
10
         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!