

APV 10.2 Installation Guide for Docker

For ArrayOS APV 10.2 Release

March 20, 2018

Chapter 1 Prerequisite

1.1 Server Software Requirement

The server hosting APV 10.2 must meet the following software requirements:

- Operating System: CentOS 7.3
- Docker Version: Docker 1.10 or later

1.2 Server Hardware Requirement

The server hosting APV 10.2 must meet the following hardware requirements:

Hardware	Requirement
CPU	No special requirement
Memory	At least 2 GB physical memory reserved for system use
Disk	At least 10 GB disk space allocated to the APV instance
NIC Card	Supported Models: Intel 82599, Intel XL710, Intel E1000, Virtio, VMware E1000, VMXNET 3
SSL Card	Supported Model: Cavium NITROX III

1.3 Deployment File Preparation

A list of files and templates are prepared by Array Networks in advance for deployment usage. Please contact Array Network Customer Support to get the “apv_docker_10_2.tar.gz” file before the deployment.

The “apv_docker_10_2.tar.gz” file includes the following:

- **docker.image.tar:** the APV 10.2 image for Docker.
- **loader.conf:** the configuration file that controls the load behavior of the system. In the loader.conf file, the administrator needs to configure memory size (mandatory) and CPU core mask (optional) for APV instance:
 - Memory size: Mandatory. This can be configured in parameter “rte_mem_size”. The configuration of memory size will affect the hugepage used by APV instance. For detailed information, please see section 2.1 Configuring Linux Hugepage.

- CPU core mask: Optional. This can be configured in parameter “core_mask”. If no CPU core mask is specified, any available CPU core(s) will be used. Once the CPU core mask is configured, only the specified CPU core(s) will be used. For more information on how to specify the CPU core mask, please contact Array Networks Customer Support.
- **dpgk-devbind.py**: the script used for the binding between the hardware and the kernel modules.

Chapter 2 Deployment Procedure

2.1 Configuring Linux Hugepage

The configuration of the memory size determines the hugepage used by Docker container (APV instance). For example, the configuration “rte_mem_size=1024” means all the deployed Docker container(s) will use 1024 MB hugepage in total. If only one container is deployed, all the 1024 MB hugepage will be used up by this container. If two containers are deployed, each one will use 512 MB hugepage.

Example:

The following example is provided for the scenario wherein only one APV instance is deployed. To allocate 2 GB hugepage to APV instance, please make sure the configuration “rte_mem_size=2048” is already added in the loader.conf file. Perform the following procedure to configure the hugepage:

1. Add “hugepagesz=2M hugepages=1024” (hugepagesz indicates the size of one hugepage and hugepages indicates the hugepage number, which can be adjusted when hugepage changes) to the “/boot/grub2/grub.cfg” file in the line starting with “linux16” as the example shown below:

```
#vim /boot/grub2/grub.cfg
linux16 /vmlinuz-3.10.0-514.el7.x86_64 root=/dev/mapper/centos-roott ro rd.lvm.lv=centos/swap
vconsole.font=latacyrheb-sun16 rd.lvm.lv=centos/root crashkernel=auto vconsole.keymap=us
rhgb quiet LANG=en_US.UTF-8 console=ttyS0 hugepagesz=2M hugepages=1024
```

2. Reboot the system to take effect.

2.2 Creating the Hugepage Directory for APV

Create a directory for the APV running on Docker using the following command:

```
mkdir -p /dev/hugepages/docker/apv
```

2.3 Preparing VFIO Kernel Module

In Virtual Function I/O (VFIO) kernel module preparation, please identify whether the deployment environment is a physical machine or a virtual machine. Considering the I/O Memory Management Unit (IOMMU) is not supported by virtual machines but inherently supported by physical machines, administrators need to perform the preparation work according to its actual deployment environment.

➤ **For Virtual Machine**

Enable the No-IOMMU mode for a virtual machine using the following commands:

```
#modprobe vfio enable_unsafe_noiommu_mode=Y
#modprobe vfio-pci
```

Check whether the No-IOMMU mode has been enabled using the following command:

```
#cat /sys/module/vfio/parameters/enable_unsafe_noiommu_mode
Y
```

The output “Y” indicates that the No-IOMMU mode has been enabled.

➤ **For Physical Machine**

Perform the VFIO kernel module preparation work for a physical machine using the following commands:

```
#modprobe vfio
#modprobe vfio-pci
```

2.4 Binding Hardware to the Kernel Modules

After loading the required modules, administrators need to bind the NIC card and SSL card to the kernel modules. Following is an example of binding the NIC card to VFIO kernel module.

To bind the NIC card to the VFIO kernel module, perform the following procedure:

1. Search the PCI address of the NIC card, using the following command:

➤ **For Physical Machine**

```
#lspci | grep 82599ES
04:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network
Connection (rev 01)
```

➤ **For Virtual Machine**

For VMXNET NIC in VMware virtual machines, please use the following command:

```
#lspci | grep VMXNET
```

For Virtio NIC in KVM virtual machines, please use the following command:

```
#lspci | grep virtio
```

2. Display the NIC binding status using the following command:

```
#!/dppk-devbind.py --status
Network devices using DPDK-compatible driver
=====
<none>

Network devices using kernel driver
=====
```

```
0000:04:00.0 '82599ES 10-Gigabit SFI/SFP+ Network Connection' if=enp4s0f0 drv=ixgbe
unused=vfio-pci
```

Other network devices

=====

<none>

As shown above, the VF of the NIC “82599ES 10-Gigabit” is still unused and is available for the binding. And its related BUS INFO is “0000:04:00.0”, which could be utilized for the binding operation.



Note: When deploying the APV instance with SR-IOV VF, please add “intel_iommu=on iommu=pt” in the kernel argument as the method in 2.1 Configuring Linux Hugepage. And then set the allow_unsafe_interrupts option for the VFIO module using the “**echo Y > /sys/module/vfio_iommu_type1/parameters/allow_unsafe_interrupts**” command.

3. Bind the NIC VF to the VFIO module using the NIC binding script “**dpdk-devbind.py**” (contained in the “apv_docker_10_2.tar.gz” file).

```
#!/dpdk-devbind.py --bind=vfio-pci 0000:04:00.0
```



Note: Please choose an inactive NIC card when performing this step.

4. Find the IOMMU group number of the binding hardware using the following command:

```
#find /sys/devices/ -name "iommu_group" | grep 04:00.0
/sys/devices/pci0000:00/0000:04:00.0/iommu_group/
#readlink -f /sys/devices/pci0000:00/0000:04:00.0/iommu_group/
/sys/kernel/iommu_groups/0
```

As shown above, the IOMMU group number of the binding hardware can be displayed after the binding operation is executed successfully.

5. Check the VFIO device using the following command:

➤ **For Virtual Machine**

```
#ls /dev/vfio/noiommu-<iommu_group_number>
```

➤ **For Physical Machine**

```
#ls /dev/vfio/<iommu_group_number>
```

2.5 Configuring the loader.conf File

The file loader.conf is used to load the default configuration of the system. Please locate the loader.conf file in the “apv_docker_10_2.tar.gz” file, and add a line “pci_whitelist=0000:04:00.0” to the end of the loader.conf to enable the NIC VF.

To add multiple NIC VFs, please use comma “,” to separate the PCI BDF numbers, for example: “pci_whitelist=0000:04:00.0, 0000:05:00.0”.

In the docker environment, a maximum of 4 CPU cores can be allocated to the docker instances. Therefore, please add a line to the end of the loader.conf file to limit the number of CPU core masks. For example, to set the number of CPU core masks to 2, add the line “core_mask=0x3”.



Note: To enable the SSL card after the binding operation, please replace the hardware BUS ID in “pci_whitelist=<PCI_BDF_NUM>”.

2.6 Importing the APV Image

After enabling the Docker service and starting the Docker, please import the “docker.image.tar” image (contained in the “apv_docker_10_2.tar.gz” file) using the following command:

```
#docker load -i docker.image.tar
```

Check the image name of the imported Docker image using the following command:

```
#docker images
```

2.7 Creating the APV

Create the APV instance using the following command:

```
#docker create -i -t --privileged=true --ulimit memlock=-1 -v /dev/hugepages:/mnt/huge/
--name <apv_name> <vfio_arguments> -v /dev/vfio/vfio:/dev/vfio/vfio --device
/dev/vfio/vfio:/dev/vfio/vfio -v /root/loader.conf:/boot/loader.conf -v
/sys/fs/cgroup:/sys/fs/cgroup:ro --cap-add=SYS_ADMIN --cap-add=NET_ADMIN
--net=none <image_name>
```



Note: The “/root/loader.conf” directory should be the path wherein the loader.conf file is decompressed.

The following parameters need to be specified to complete the above command:

- apv_name: indicates the APV instance name.
- image_name: indicates the image name which could be fetched in section 2.6 Importing the APV Image.
- vfio_arguments: indicates the VFIO arguments.

Argument syntax: `--device /dev/vfio/<vfio_device>:/dev/vfio/<vfio_device>`

Fetch the information of “vfio_device” in section 2.4 Binding Hardware to the Kernel Modules. To add more VFIO devices, please arrange multiple VFIO arguments in parallel.

2.8 Starting the Docker

Start the Docker container using the following command:

```
#docker start <apv_name>
```

2.9 Entering the APV CLI

To enter the APV CLI, please perform the following procedure:

1. Disable the selinux permission temporarily using the following command:

```
#setenforce 0
```

2. Log into the APV CLI using the following command:

```
#docker exec -it <apv_name> "/ca/bin/ca_shell"
```

In Docker environment, the following commands are not supported:

Unsupported Commands	Alternative Commands
system update	config backup + Re-import docker image + config restore
system reboot	"#docker stop" + "#docker start"
system shutdown	"#docker stop"



Note: For more information on APV system update, please contact Array Networks Customer Support.

Chapter 3 References

- [1]. [“Use the Docker Command Line”](#). *docs.docker.com*.
- [2]. [“DPDK documentation”](#). *dpgk.org*.
- [3]. Matei Cezar (2016-12-15). [“Installation of CentOS 7.3 Guide”](#). *tecmint.com*.