XenServer 8 Server Certification Kit Guide

Published Mar 2024

V8.4.0 Edition

Table of Contents

Introduction

The Server Certification Kit is an automated test harness for certifying servers, network cards and local storage devices for use with XenServer.

The kit is designed to run automatically once the user has correctly configured their server, and external environment according to the instructions given below.

Whilst we do our best to ensure the kit is bug free, we are still working on improving the kits robustness – if you encounter any issues, then we’d ask that you raise an appropriate bug ticket for us to investigate. Xenserver is committed to improving both the kits quality, and value to both vendors and itself.

A number of vendors have expressed interested in integrating this kit into their own test systems – as much as possible we have designed to kit to be easy to integrate. If you feel there could be modifications made to the kit that would improve its usefulness for you – then please let us know.

If you are interested in contributing improvements to the kit, then please take a look at the project on GitHub: https://github.com/xenserver/auto-cert-kit

> **Note**:
>
> Multicast and SR-IOV tests are optional. Ignore the corresponding test results if you do not want to certify either of the two.

Prerequisites

- **Pool** - Users must set up a pool consisting of two hosts, both running the version of XenServer that is being certified.
- **Network Adaptors** - Each host requires a minimum of two network interfaces. The corresponding network interfaces on each host, are expected to be plugged into the same layer 2 network.

> **Notes**:
>
> - To certify the Multicast feature, please make sure the network device in the test environment supports Multicast.

> - To certify the SR-IOV feature, please make sure both the network adapter and the server support it.

- **Installation** - Installation of the server certification kit must be performed on both the pool coordinator and the pool member machines.
- **VLANs** - For the devices under test, at least on VLAN must be configured on the switch and specified in the network config file.
- **Large MTU** – For network adaptor testing, ports on the switch must be configured correctly in order to allow packets of size 9000 bytes to be passed through without fragmentation.

XenServer Installation

Please download and install XenServer on the two hosts being used for certification. You should then join them in together in a pool using either the CLI or XenCenter.

If you would like to join two hosts that are not identical, or of which CPUs cannot be masked correctly – it is possible to force a pool join. This is acceptable for the operations required by the test kit and can be done by executing the following on the CLI of the pool member host:

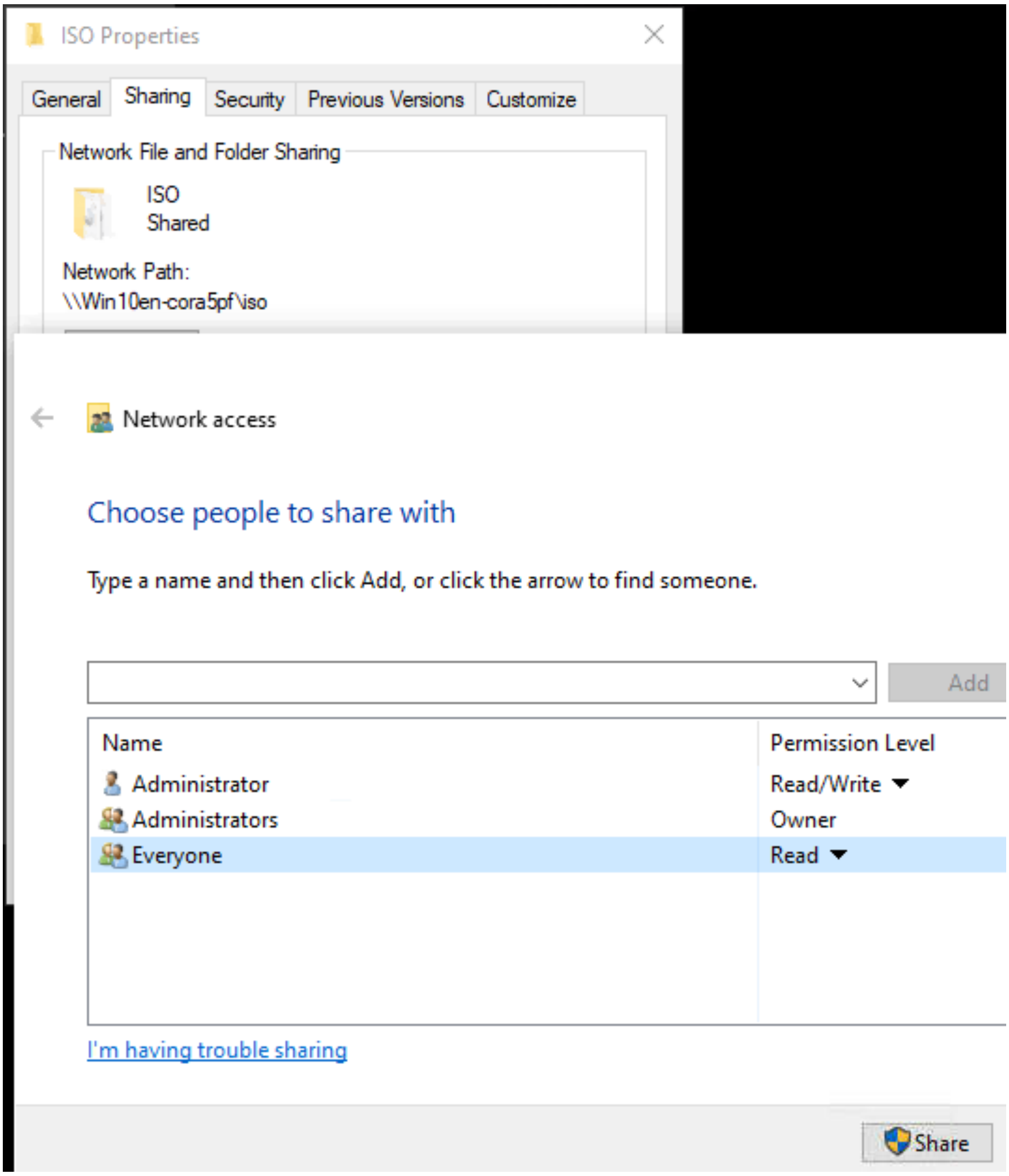
```
xe pool-join master-address=<master-ip> master-
username=root \
master-password=<pass> force=true
```

Then need to create a VM template, which will be used in the test for generating test VM (we call it Droid VM), below is an example for setting up a VM template for Droid VM with XenCenter. (You should already have a Windows Desktop/VM which is running XenCenter on it.)

1.To create a VM template, you should create a VM with using e.g. Rocky 8.6 or 8 latest first as below steps.

(1) Download Rocky 8.x ISO from official web site URL e.g. Rocky-x86_64-minimal.iso and put it onto the same machine of running XenCenter.

(2) On the same machine which is running XenCenter, create a share folder and put above Rocky 8 iso file in it, here as example I will save above iso file in "c:\iso" folder on the windows VM, then share this folder and add “everyone” read access. (Please make sure that your XenServer has network access to this VM which is hosting the shared storage.)



ISO
Shared

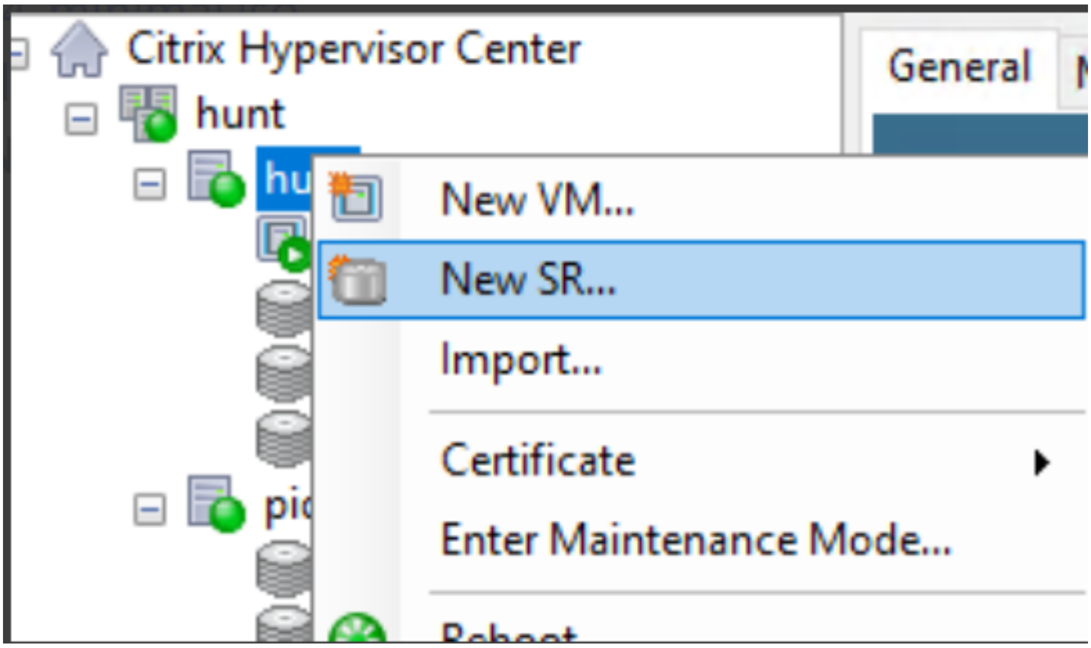
Network Path:
\\Win10en-cora5pf\iso

Choose people to share with

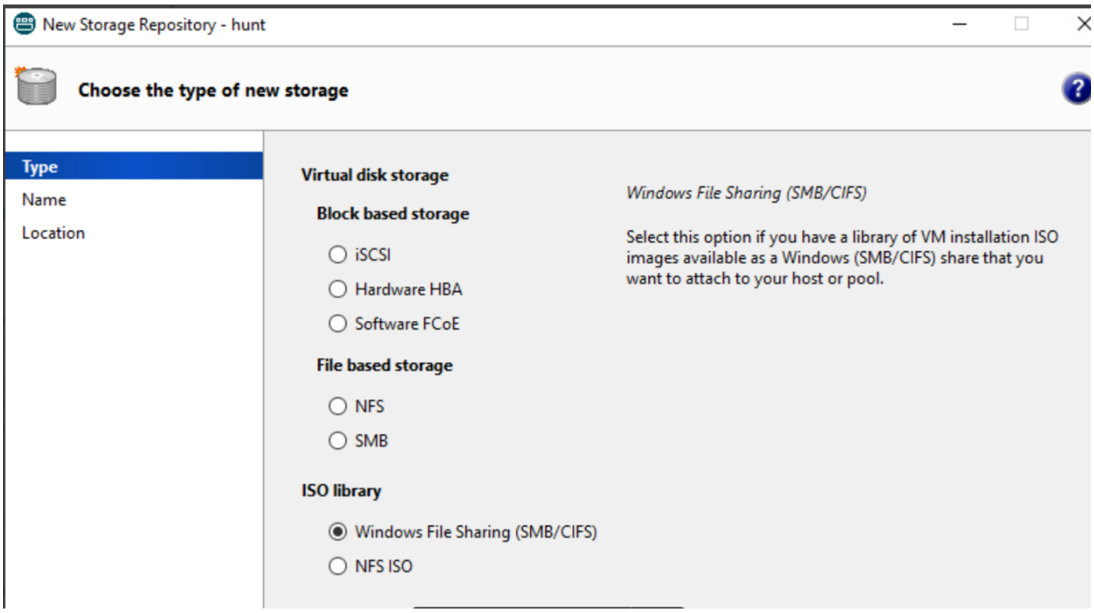
Type a name and then click Add, or click the arrow to find someone.

Name	Permission Level
Administrator	Read/Write ▼
Administrators	Owner
Everyone	Read ▼

(3) Create new SR with above share folder on XenCenter, open XenCenter and add your host, right click your host then click **New SR.**



(4) Select **Windows File Sharing.**



Choose the type of new storage

Virtual disk storage

Block based storage

○ iSCSI

○ Hardware HBA

○ Software FCoE

File based storage

○ NFS

○ SMB

ISO library

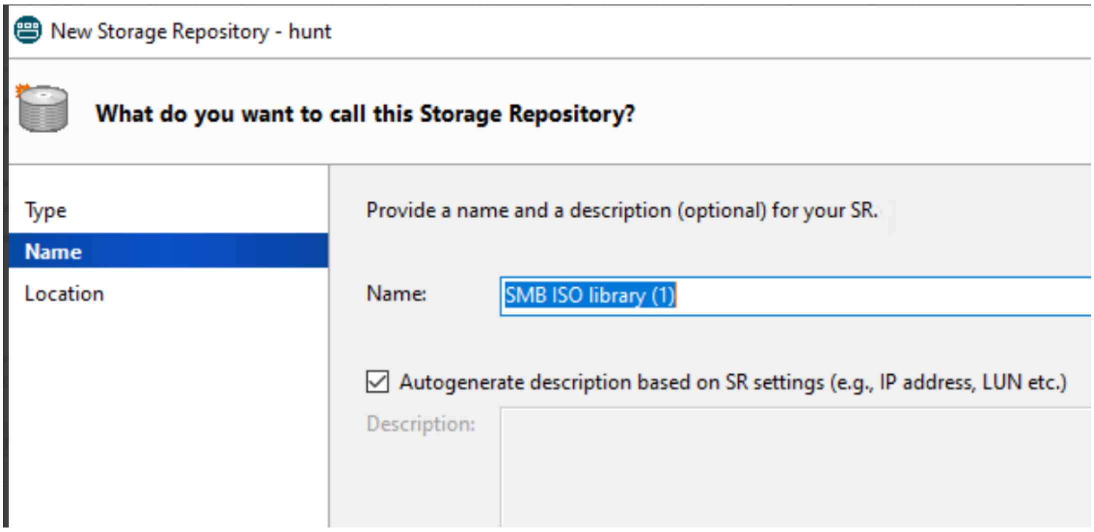
● Windows File Sharing (SMB/CIFS)

○ NFS ISO

Windows File Sharing (SMB/CIFS)

Select this option if you have a library of VM installation ISO images available as a Windows (SMB/CIFS) share that you want to attach to your host or pool.

(5) Click Next and input the name, then click **Next.**



New Storage Repository - hunt

What do you want to call this Storage Repository?

Provide a name and a description (optional) for your SR.

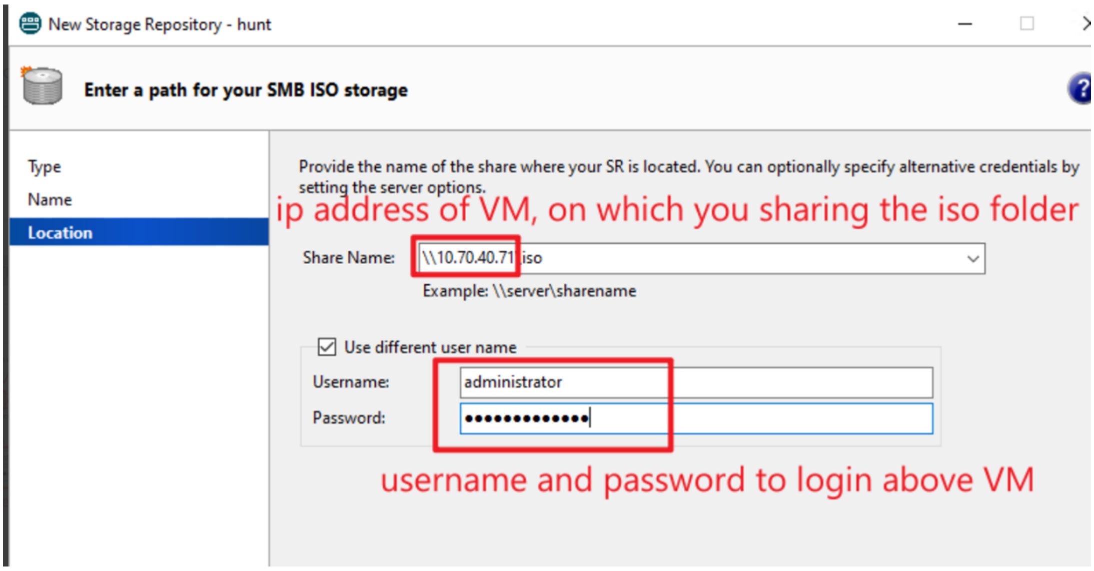
Name: SMB ISO library (1)

☑ Autogenerate description based on SR settings (e.g., IP address, LUN etc.)

Description:

(6) Input the sharing path, username and password, then click **Finish.**

> **Note:**
> In share name should fill in “\\FQDN of SMB server\share folder” or “\\IP Address of SMB server\share folder”, SMB server is the VM on which you created the share folder in step 1.2, here as example, in step 1.2 I have created a share folder on windows VM(it’s IP 10.70.40.71) in “c:\iso”, this windows VM is the SMB server and share folder is ‘iso’, so in share name I fill in “\\10.70.40.71\iso”. Please refer to below screenshot.



New Storage Repository - hunt

Enter a path for your SMB ISO storage

Provide the name of the share where your SR is located. You can optionally specify alternative credentials by setting the server options.

ip address of VM, on which you sharing the iso folder

Share Name: \\10.70.40.71\iso

Example: \\server\sharename

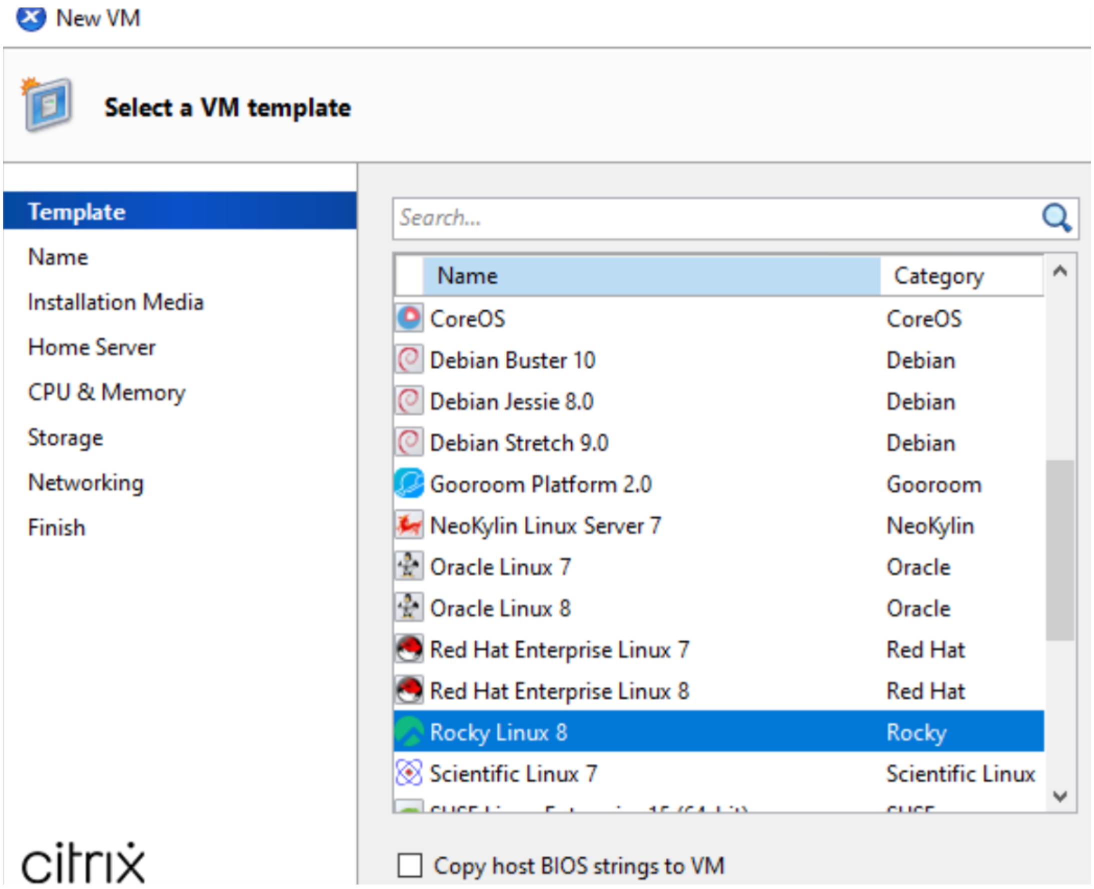
☑ Use different user name

Username: administrator

Password: •••••••••••••

username and password to login above VM

(7) Now you can create new VM in XenCenter, select **Rocky Linux 8.**



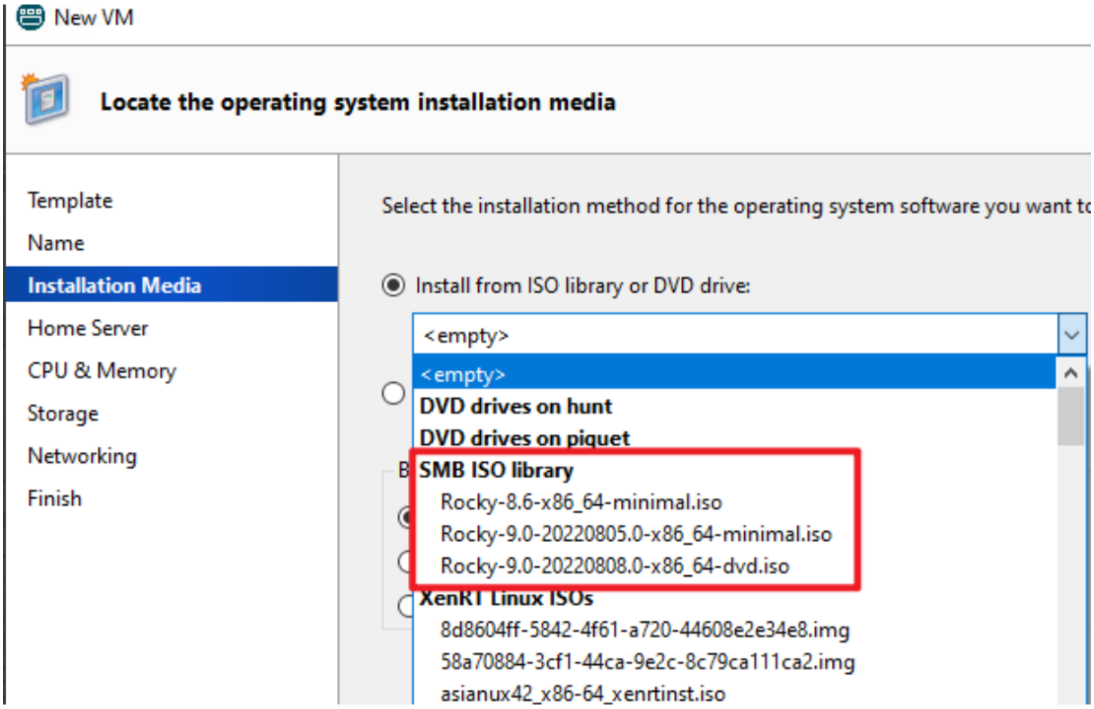
New VM

Select a VM template

Name	Category
CoreOS	CoreOS
Debian Buster 10	Debian
Debian Jessie 8.0	Debian
Debian Stretch 9.0	Debian
Gooroom Platform 2.0	Gooroom
NeoKylin Linux Server 7	NeoKylin
Oracle Linux 7	Oracle
Oracle Linux 8	Oracle
Red Hat Enterprise Linux 7	Red Hat
Red Hat Enterprise Linux 8	Red Hat
Rocky Linux 8	Rocky
Scientific Linux 7	Scientific Linux

□ Copy host BIOS strings to VM

(8) After you created the new SR, you can select the iso file, which you save in the sharing folder.



New VM

Locate the operating system installation media

Select the installation method for the operating system software you want t

● Install from ISO library or DVD drive:

<empty>

Template
Name
Installation Media
Home Server
CPU & Memory
Storage
Networking
Finish
<empty>
DVD drives on hunt
DVD drives on piquet
SMB ISO library
Rocky-8.6-x86_64-minimal.iso
Rocky-9.0-20220805.0-x86_64-minimal.iso
Rocky-9.0-20220808.0-x86_64-dvd.iso
XenRT Linux ISOs
8d8604ff-5842-4f61-a720-44608e2e34e8.img
58a70884-3cf1-44ca-9e2c-8c79ca111ca2.img
asianux42_x86-64_xenrtinst.iso

- **Memory** = 4GB and **Storage** = 10GB is recommended.
- Please set root password = **xenserver**

2.After Rocky 8 Linux installed successful, please Install latest XenServer-LinuxGuestTools first.

- Download: XenServer downloads page.
- Install: Install XenServer VM Tools for Linux.

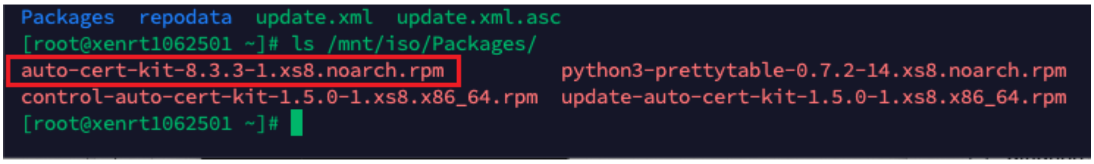
3.You can get scripts from Server Certification Kit ISO file, which can be used for setting up VM, please refer below example for getting and using those scripts.

(1) Download Server Certification Kit ISO file from HCL website and put it on Rocky 8 Linux VM in /root.

(2) Mount Server Certification Kit ISO, copy **xenserver-auto-cert-kit-.el7.noarch.rpm** from IOS/ Packages folder to local disk, please refer below example.

```
# mkdir /mnt/iso
# mount -t iso9660 -o loop /root/xenserver-server-cert-
kit-xs8.iso /mnt/iso/
```

Here as example, I used Server Certification Kit ISO version 1.3.13, so if you use another version, should replace 1.3.13-1 to your server certification kit version, you can also check the file as below screenshot and then copy this file to “/root”.



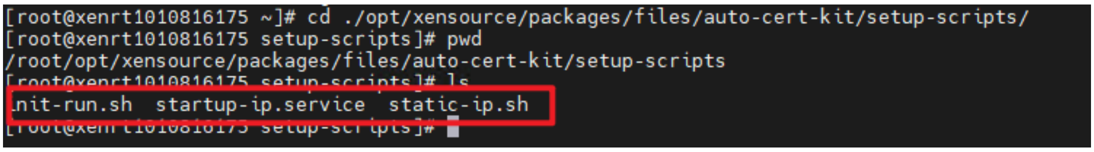
```
Packages  repodata  update.xml  update.xml.asc
[root@xenrt1062501 ~]# ls /mnt/iso/Packages/
auto-cert-kit-8.3.3-1.xs8.noarch.rpm          python3-prettytable-0.7.2-14.xs8.noarch.rpm
control-auto-cert-kit-1.5.0-1.xs8.x86_64.rpm  update-auto-cert-kit-1.5.0-1.xs8.x86_64.rpm
[root@xenrt1062501 ~]#
```

```
    # cp /mnt/iso/Packages/xenserver-auto-cert-
kit-1.3.13-1.el7.noarch.rpm /root
```

(3) Unpack this rpm file. Then can find VM setup scripts in “/root/opt/xensource/packages/files/auto-cert-kit/setup-scripts/”

```
    # rpm2cpio xenserver-auto-cert-
kit-1.3.13-1.el7.noarch.rpm | cpio -ivd
```

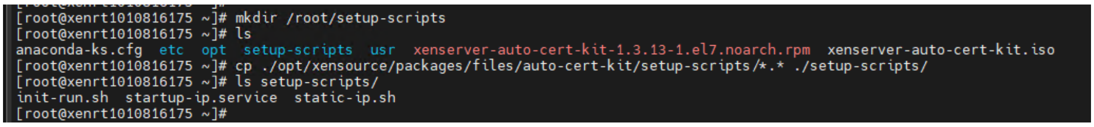
(4) In setup-scripts folder you can find three files as below screenshot shows.



```
[root@xenrt1010816175 ~]# cd ./opt/xensource/packages/files/auto-cert-kit/setup-scripts/
[root@xenrt1010816175 setup-scripts]# pwd
/root/opt/xensource/packages/files/auto-cert-kit/setup-scripts
[root@xenrt1010816175 setup-scripts]# ls
nit-run.sh  startup-ip.service  static-ip.sh
[root@xenrt1010816175 setup-scripts]#
```

(5) Then create folder “/root/setup-scripts” and copy all above three files to this folder.

```
    # mkdir /root/setup-scripts
    # cp ./opt/xensource/packages/files/auto-cert-kit/
setup-scripts/*.* ./setup-scripts/
```

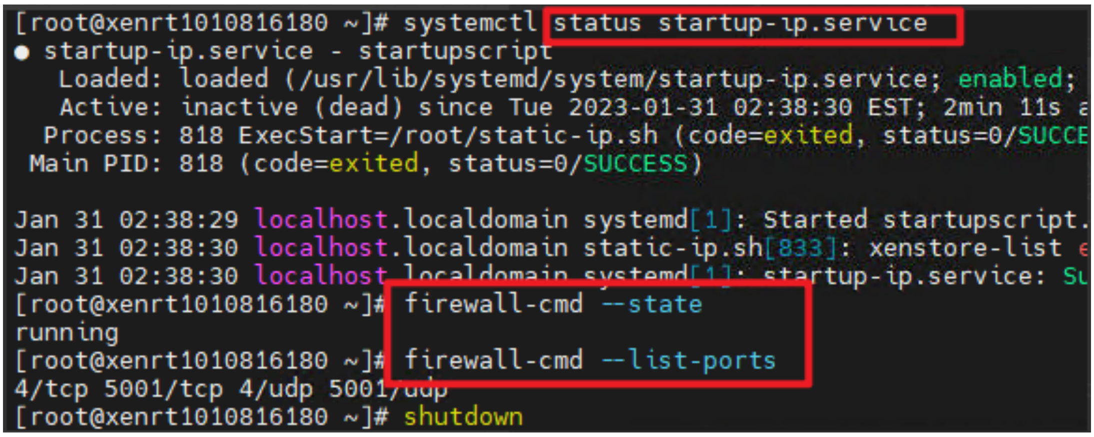


```
[root@xenrt1010816175 ~]# mkdir /root/setup-scripts
[root@xenrt1010816175 ~]# ls
anaconda-ks.cfg  etc  opt  setup-scripts  usr  xenserver-auto-cert-kit-1.3.13-1.el7.noarch.rpm  xenserver-auto-cert-kit.iso
[root@xenrt1010816175 ~]# cp ./opt/xensource/packages/files/auto-cert-kit/setup-scripts/*.* ./setup-scripts/
[root@xenrt1010816175 ~]# ls setup-scripts/
init-run.sh  startup-ip.service  static-ip.sh
[root@xenrt1010816175 ~]#
```

(6) Then can run commands as below to setup VM.

```
    # cd /root/setup-scripts/
    # sh init-run.sh
    # reboot
```

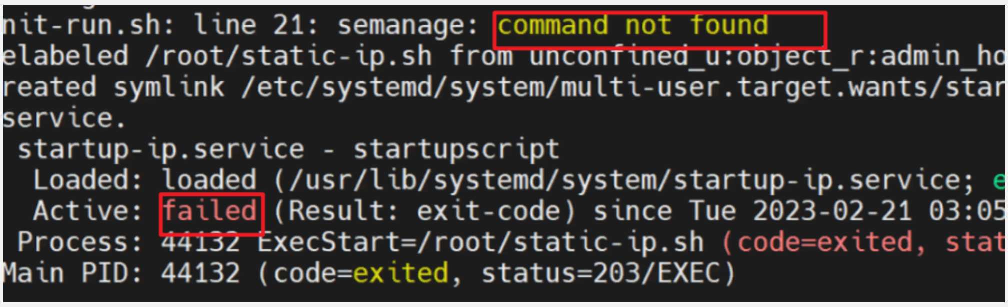
(7) After VM rebooting, service status and firewall rule should be as below..



```
[root@xenrt1010816180 ~]# systemctl status startup-ip.service
● startup-ip.service - startupscript
   Loaded: loaded (/usr/lib/systemd/system/startup-ip.service; enabled;
   Active: inactive (dead) since Tue 2023-01-31 02:38:30 EST; 2min 11s a
  Process: 818 ExecStart=/root/static-ip.sh (code=exited, status=0/SUCCE
 Main PID: 818 (code=exited, status=0/SUCCESS)

Jan 31 02:38:29 localhost.localdomain systemd[1]: Started startupscript.
Jan 31 02:38:30 localhost.localdomain static-ip.sh[833]: xenstore-list e
Jan 31 02:38:30 localhost.localdomain systemd[1]: startup-ip.service: Su
[root@xenrt1010816180 ~]# firewall-cmd --state
running
[root@xenrt1010816180 ~]# firewall-cmd --list-ports
4/tcp 5001/tcp 4/udp 5001/udp
[root@xenrt1010816180 ~]# shutdown
```

> **Note:**
> Scripts works on Rocky 8.6, but you may encounter dependency problems on the newer Rocky 8 version, if you encounter problem as below screenshot.



```
nit-run.sh: line 21: semanage: command not found
elabeled /root/static-ip.sh from unconfined_u:object_r:admin_ho
reated symlink /etc/systemd/system/multi-user.target.wants/star
service.
 startup-ip.service - startupscript
  Loaded: loaded (/usr/lib/systemd/system/startup-ip.service; e
  Active: failed (Result: exit-code) since Tue 2023-02-21 03:05
 Process: 44132 ExecStart=/root/static-ip.sh (code=exited, stat
Main PID: 44132 (code=exited, status=203/EXEC)
```

(8) We can see that can’t find command “semanage”, so need to install “semanage” by manual as below steps.

- How to install necessary packages for getting semanage command using the yum command

```
# yum provides /usr/sbin/semanage
```

- From the above sample output, you can see that we need to install policycoreutils-python-utils-2.8-16.1.el8.noarch package to use the semanage command.

```
# yum install policycoreutils-python-utils
```

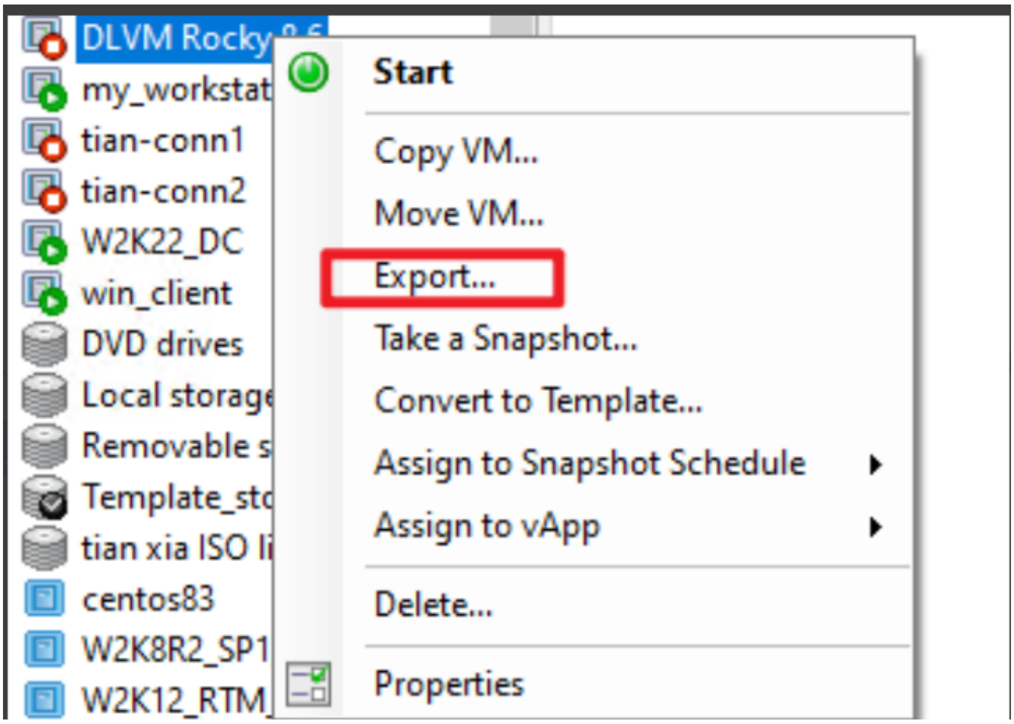
- Now “semanage” command can be used, you can re-run “init-run.sh” and verify the result.

(9) Extend pm_freeze_timeout for VM

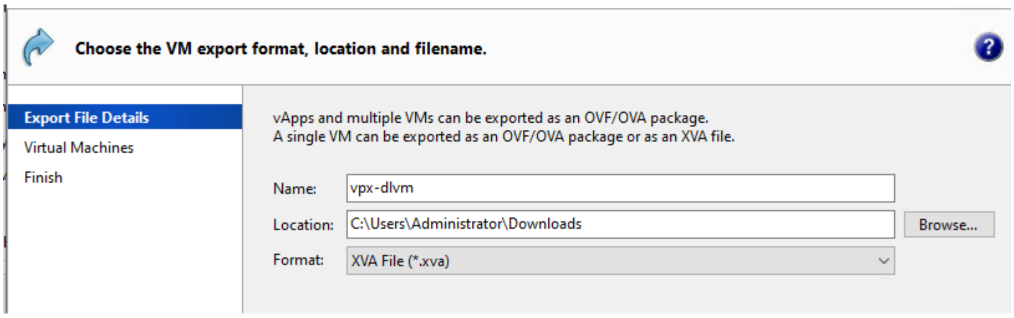
```
# echo 300000 > /sys/power/pm_freeze_timeout
```

4.When above steps done, need to export this VM as “.xva" file.

- Shut down VM, right click VM on XenCenter and select "Export...".



- Input the name=vpx-dlvm, and location and select Format as "XVA file".


Choose the VM export format, location and filename.

vApps and multiple VMs can be exported as an OVF/OVA package.
A single VM can be exported as an OVF/OVA package or as an XVA file.

Name: vpx-dlvm

Location: C:\Users\Administrator\Downloads

Format: XVA File (*.xva)

For the automated certification kit to run successfully, there are currently the following requirements on your XenServer deployment:

Server Certification Kit Installation

The XenServer certification kit needs to be run with the latest version of the corresponding XenServer Release. Make sure that XenServer 8 has been updated to the latest version before testing. Once the above environment has been set up, please download the xenserver-server-cert-kit-xs8.iso supplemental pack as provided by XenServer, and copy the ISO onto the /root directory of the Dom0 filesystem residing on the pool coordinator host. Use the following command to install server certification kit into all hosts in the pool:

```
xe update-upload file-name=/root/xenserver-server-cert-kit-xs8.iso
```

The command returns the update UUID of server certification kit package on successful upload.

```
xe update-apply uuid=<update uuid of server certification kit> --multiple
```

Upload "vpx-dlvm.xva” file to the server certification kit home folder on **all hosts** "/opt/xensource/packages/files/auto-cert-kit" .

Server Certification Kit Operation

To run the certification tests, please run the following commands:

```
cd /opt/xensource/packages/files/auto-cert-kit
./ack_cli.py [options]
```

Unless specified otherwise, the test kit will attempt to execute all of its tests (network, local storage, CPU and operational). For network adaptor certification, only the network tests are a requirement for certification, though it is advisable for the complete test kit to have been run.

For any of the options the user is required to specify a network configuration (ini format) file on the command line:

```
./ack_cli.py -n network.conf
```

There is an example file located in the server certification kit's root directory (networkconf.example). The purpose of this file is to show all available configuration items for both network interfaces and static IP addressing. You can refer to Setting up the network configuration to learn how to construct network.conf file.

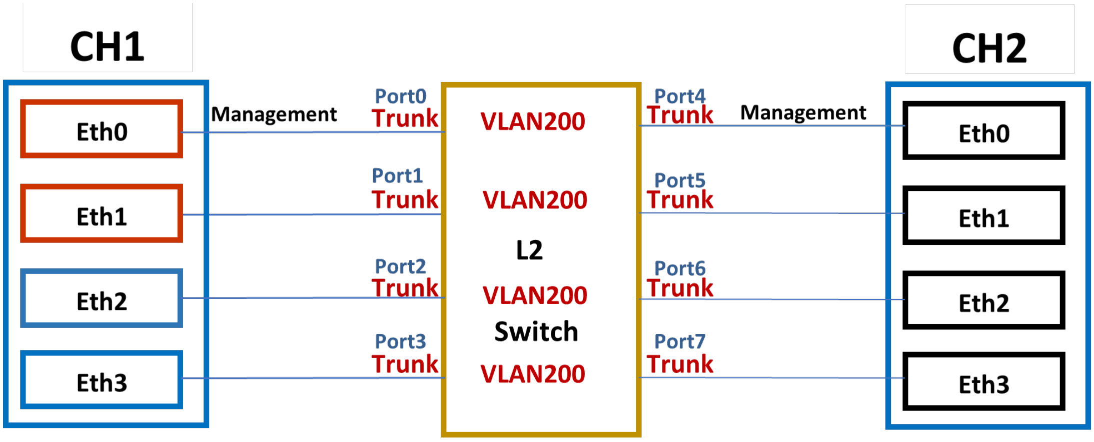
Once executed, the server certification kit will then generate and execute a list of tests for each device on the pool coordinator host that should be certified.

You can also run the kit in debug mode, with the argument -d. This will cause the kit to exit on exception, rather than continue to run the remaining tests.

Setting up the network configuration

If DHCP is available, we would suggest that you configure the test kit to rely on DHCP to hand out IP addresses to test VMs used by the kit. This then requires that you describe in the network configuration (config) file, which interfaces you wish to test, and which L2 networks and VLANs are associated with each. If the interface with SR-IOV feature is to be certified, you must specify the VF driver to be used.

To provide you an example, please consider the following setup:



In the preceding illustration, eth0 is the management interface for both XenServer hosts (it is required that the management interface be the same on both hosts, else the kit will fail to execute properly).

XenServer 1 (XS1) has the hardware that is being certified, so we need to specify in our network config file the devices that can be used for testing, and their properties. In this example, eth0/eth1 and eth2/eth3 are pairs of identical cards.

The VLAN is needed for the test. Switch ports configured as 802.1Q VLAN trunk ports can be used with the XenServer VLAN features to connect guest virtual network interfaces (VIFs) to specific VLANs. In this case, the XenServer server performs the VLAN tagging/untagging functions for the guest, which is unaware of any VLAN configuration.

In the VLAN test case, the network packet from the NICs in CH1 host (e.g., eth0, eth1…shown above) will have both tagged type and untagged type, so the ports on the switch connected to each XenServer host must be configured as the corresponding VLAN and in trunk mode. Then the switch can correctly identify and process the data packets to ensure the tagged packets can be received and sent to the corresponding VLAN ports.

Moreover, the "Native VLAN" is also important for the trunk port and it defaults to “1”. For the tagged packet, if its VLAN id is the same as the “Native VLAN”, the tag will be removed before its forwarding. For the untagged packet, the receiving switch will internally convert it to "Native VLAN" so that the untagged packet will be processed in the same way as the tagged packet.

For more information about configuring VLANs on your switch, see the documentation for your switches. If you use a Cisco switch, you can click here for some reference. If you use switches from other manufacturers, you can go to their official websites for some help.

As the example diagram shown above, the L2 switch is configured to allow VLAN 200 packets to be passed through between all the connected ports in trunk mode. The “Native VLAN” is not changed and defaults to “1”. If all the interfaces support SR-IOV, the user of the kit could construct the following config file:

```
[eth0]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
[eth1]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
[eth2]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
[eth3]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
```

See below for a description of the keys:

- **network_id** - ID of the logical network on layer 2
- **vlan_ids** - VLAN ID used in the test, delimited by comma if multiple IDs present.
- **vf_driver_name** - Driver name of the SR-IOV VF to be used by the Droid VM If the key is specified, server certification kit will write the value, denoted as `<name>`,

  into file /etc/modulesload.d/<name>.conf of the Droid VM. Then the VM is able to load it automatically during booting.If the key is not specified or its value is empty, it indicates that the VF driver is already able to be loaded automatically.

- **vf_driver_pkg** - Name of the driver package (.rpm) for the SR-IOV VF. The package is located under /opt/xensource/packages/files/auto-cert-kit/ of the pool coordinator host. If the key is specified, server certification kit will upload the specified driver to the Droid VM and install. If the key is not specified or its value is empty, it indicates the driver has already been installed in the Droid VM.

> **Note**:
>
> The Droid VM used by server certification kits is /opt/xensource/packages/files/auto-cert-kit/vpxdlvm.xva, which is based on upstream Rocky 8.6 or 8 latest. The specified .rpm package must be applicable to the Droid VM in use.

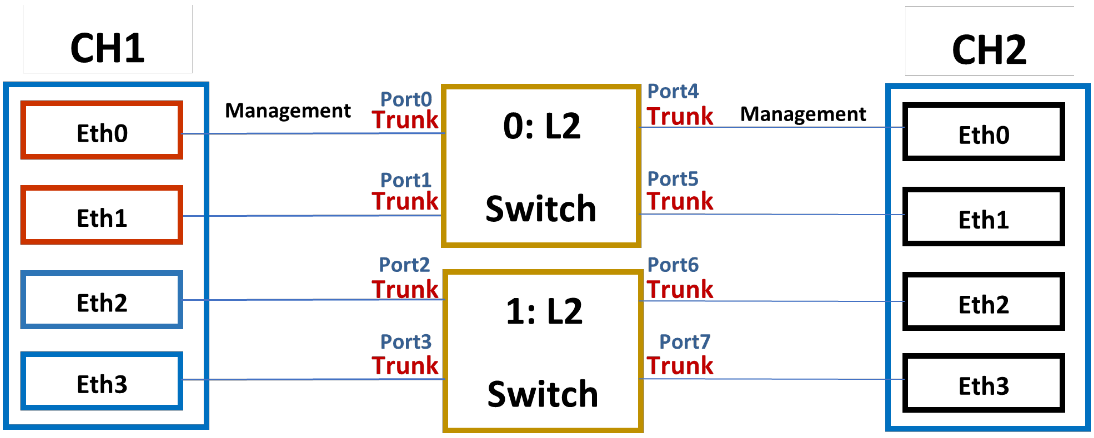
The preceding configuration allows for the server certification kit to use any of the devices that it wished for the various tests constituting the kit – however the same results could be achieved by specifying the following configuration:

```
[eth1]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-
ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
```

```
[eth3]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-
ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
```

This is because eth1 and eth3 represent two different network cards being certified, and are both located on the same Layer 2 physical network. This is required for bonding tests.

If the servers had instead been set up as shown in the following diagram:



Then the network configuration file would need to differentiate between eth0/eth1 and eth2/eth3’s logical network. In the network configuration file, we required that the user specify an integer to identify which interfaces are connected to the same layer 2 network. (The integer has no meaning other than for the purpose of differentiating).

It is important now to consider the following as an example of an **incorrect** configuration file:

```
[eth1]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
[eth3]
network_id = 1
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
```

Whilst the user has correctly differentiated between eth1 and eth3 with logical network IDs ‘0’ and ‘1’, there is now not enough interfaces specified for the kit to create a valid bond. In order for a valid bond to be configured, we must have at least two interfaces defined for the same logical network ID.

This would mean specifying the following configuration:

```
[eth0]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
[eth1]
network_id = 0
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
[eth2]
network_id = 1
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
[eth3]
```

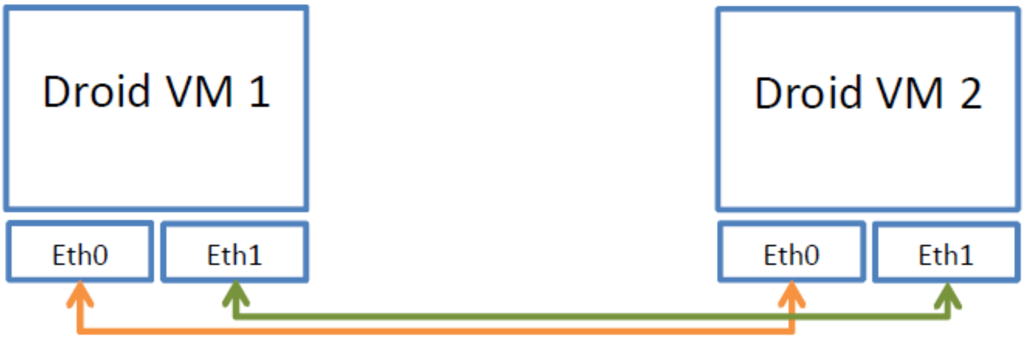
```
network_id = 1
vlan_ids = 200
vf_driver_name = ixgbevf
vf_driver_pkg = kmod-
ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
```

Management Network

When specifying the network configuration file, the user need only to declare the management network in the config file if either it uses a static IP range, or it is required for testing (most likely as part of a bond).

The kit requires that the management network (the one being used by Xenserver) is used in each and every test it runs. This is so as to ensure the kit can communicate with the VM over the network properly.

The result is that each Droid VM will be configured as shown in the following diagram:



In each test, ‘eth1’ on the VM will correlate to the physical interface under test – and instructions run from inside the

VM such as ping and iperf commands will be directed and measured over the eth1 interface.

Configuring the kit to use static IP addresses

In the case where DHCP is not able to be used, a user can configure the kit to use a static address range. The address ranges provided by the user must be exclusively allocated for the purposes of the kit.

Each address range must be specified for the combination of Logical ID (see above) and VLAN. In the simplest case, where each physical adaptor is connected to a port on the same L2 switch, and where VLAN 200 has been defined for each adapter – the following should be added to the configuration file:

```
[static_0_200]
ip_start = 192.168.0.2
ip_end = 192.168.0.10
netmask = 255.255.255.0
gw = 192.168.0.1
```

This indicates that the kit should configure a static IP between the range 192.168.0.2-10 with a default subnet of 255.255.255.0 and a default gateway of 192.168.0.1 for any VM interfaces that are connected to a device on logic ID 0, with VLAN 200.

If only this ini section is specified, then it is assumed that for the default case, DHCP is available – and that it is only when using VLAN 200 that an IP should be allocated.

If a static address range is required for the non-VLAN case, then a user should add another static config section with the VLAN specified as 0 (default VLAN id) like so:

```
[static_0_200]
ip_start = 192.168.0.2
ip_end = 192.168.0.10
netmask = 255.255.255.0
gw = 192.168.0.1
[static_0_0]
ip_start = 192.168.1.2
ip_end = 192.168.1.10
netmask = 255.255.255.0
gw = 192.168.1.1
```

All these sections of “static_<NETWORK>_<VLAN>” is to configure for the network appears in the ini file, and which interface devices will be certified no matter whether they are in management network or test network.

There is another section “static management” supplied, it can be used if intend to allocate static IP for management network while the network is not specified in ini file. E.g., in below configuration, XenServer management interface (default eth0) is not configured to certify, and user needs to allocate static ip for management network, then “static management” section meets the requirement.

```
[eth1]
network_id = 0
vlan_ids = 200
[eth2]
network_id = 0
vlan_ids = 200
[static_0_200]
ip_start = 192.168.0.2
ip_end = 192.168.0.10
netmask = 255.255.255.0
```

```
gw = 192.168.0.1
[static_0_0]
ip_start = 192.168.1.2
ip_end = 192.168.1.10
netmask = 255.255.255.0
gw = 192.168.1.1
[static_management]
ip_start = 192.168.2.2
ip_end = 192.168.2.10
netmask = 255.255.255.0
gw = 192.168.2.1
```

Querying the status of the test run

Depending on the set of tests being executed by the kit, a host reboot may be required. This means that unless you are executing the kit from the host's physical console you will no longer see the progress of the kit after the reboot. If this happens, then you can follow the logs being generated at /var/log/auto-cert-kit.log, however you can also query the test kit status by running the status.py module (located in the server certification kit install directory).

The module will return one of the following results:

```
0:Finished (Passed:[num] Failed:[num] Skipped:[num])
1:Process not running. An error has occurred. (Passed:
[num],
Failed:[num], Skipped:[num], Waiting:[num], Running:[num])
2:Running - [num]% complete (Passed:[num], Failed:[num],
Skipped:[num],
Waiting:[num], Running:[num])
3:Server rebooting... (Passed:[num], Failed:[num],
Skipped:[num],
Waiting:[num], Running:[num])
4:Manifest file has not been created. Have run the kit?
```

```
(Has an error
occured?)
5:An error has occured reading. [testfile]
```

Test Results and Logs

After a test run has been completed, there should be two files created in the /root/ directory on the pool coordinator host:

- ack-submission-[time]-[date].tar.gz
- results.txt

The results.txt file will detail the output of your test run, specifying which tests have passed or failed, along with the features marked as supported for your device.

More specific result information and test exceptions can be found in the XML file generated after each run:

```
/opt/xensource/packages/files/auto-cert-kit/test_run.conf
```

Debug logging is currently printed to stdout, as well as the server certification kit's log file which is found in /var/log/auto-cert-kit.log. This log file is collected automatically as part of a XenServer status report (which is required for submission).

Submitting results and logs

We would obviously appreciate it if vendors could submit log/result files so that we can establish we are collecting the appropriate information concerning your devices, as well as fix any bugs that you may have found during the testing performed on your hardware.

The mechanism for providing us with these files and feedback are via the XenServer ticket tracker. Please see instructions below for creating a new ticket.

In the normal case where the test kit runs to completion, we would ask that the vendor submits the ack-submission package:

- /root/ack-submission-[time]-[date].tar.gz
- xenserver-sr-iov-certification-form , if SR-IOV certification is performed

However, if there is a failure such that the ack-submission package is not created, then we would ask that you submit the following:

- Server Status Report - this can be obtained either by either running the ‘xen-bugtool –y’ command on the coordinator host, or by using XenCenter (Tools - Get Server Status Report).
- The network.conf and test_run.conf file mentioned in the section above.

Current Known Limitations

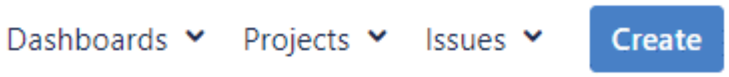
Xenserver is aware of the following limitations in the server certification kit at present:

- SR-IOV tests in server certification kit are able to test the maximum VFs per single port/PF, but not all ports/PFs combined if the network adapter has multiple ports.

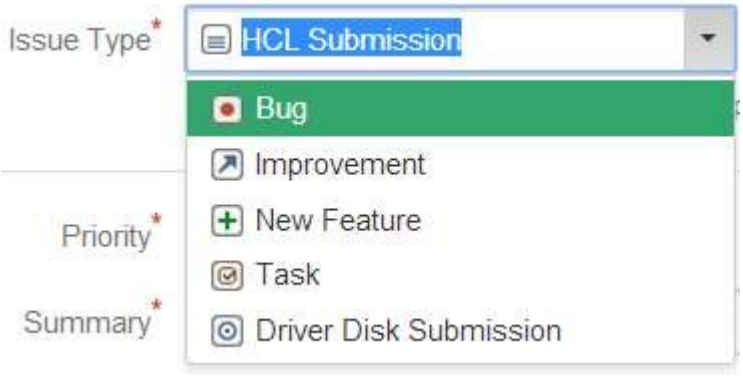
Bug Reports and Feedback

In order for us to collect your feedback on this kit, and improve it for subsequent releases, we would ask that you submit either certifications or bugs on our issue tracking system, ‘tracker’.

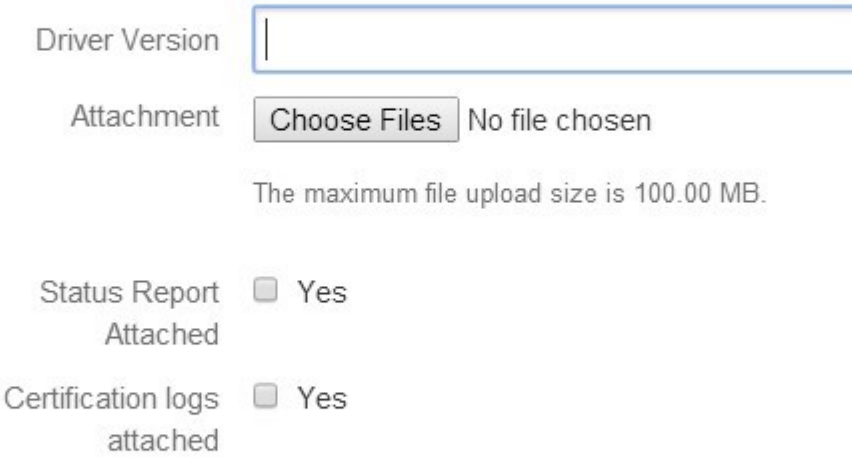
1.Tracker can be found here - https://tracker.citrix.com 2.If your company already has a project open on this system with XenServer, then please raise your issues here. If however, you do not have a specific project for your company, please raise your issues under the **XenServer Hardware Compatibility List (HCL)** project. 3.You can create an issue by click **Create** on the top menu.



4.Select the issue type as **HCL Submission** if you are looking to certify your hardware on HCL. Similarly the issue type needs to be of the type **Driver Disk Submission** in case for Driver Disks and likewise others.


Issue Type* HCL Submission

5.Fill in the details and attach the necessary logs that are required for certification. Driver Version needs to be filled only if your Issue type is of **Driver Disk Submission** else leave it blank.


Driver Version

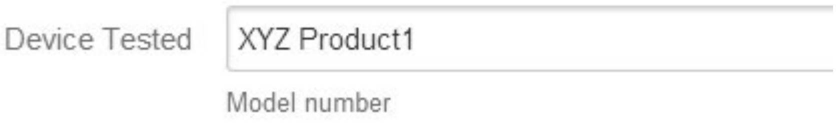
Attachment Choose Files No file chosen

The maximum file upload size is 100.00 MB.

Status Report Attached □ Yes

Certification logs attached □ Yes

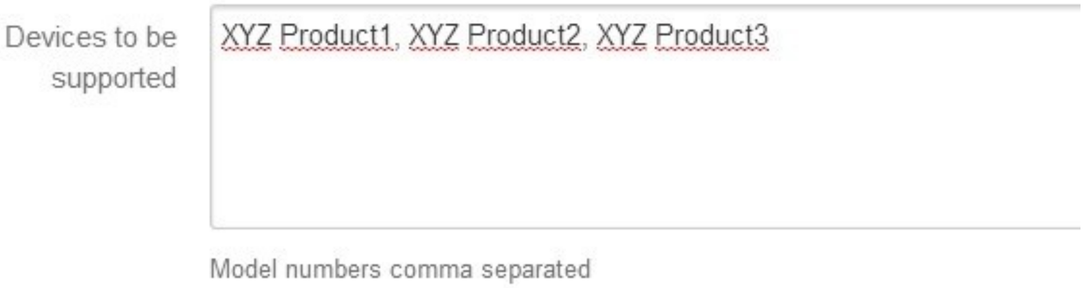
6.Enter the name of your tested product name in **Device Tested** field that matches to the exact way it needs to be updated on HCL.



Device Tested: XYZ Product1

Model number

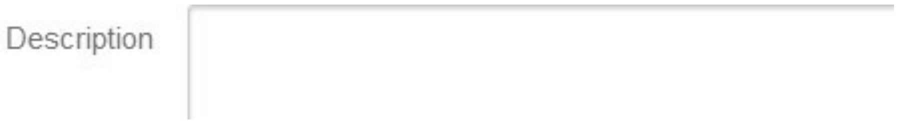
7.You can also have your equivalent devices certified based on the Device Tested product which can be listed in **Device to be supported** field separated by commas.



Devices to be supported: XYZ Product1, XYZ Product2, XYZ Product3

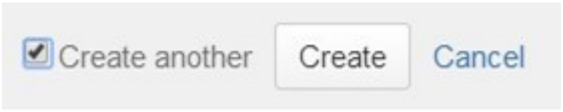
Model numbers comma separated

8.Enter a description of your product which you would like to highlight the reviewer.



Description:

9.That’s it. You can check on **Create another** before you hitting the **Create** in case you want to add more products for certification. Please note that you need to raise only one ticket per product even though you might be testing different interconnects say iSCSI, FC, and so on for a storage controller.



10.The progress of your ticket can be tracked by clicking on **My Unresolved Reported Issues** on top left corner of your Dashboard.

11.We thank you for the submission made and can usually respond you within 2-3 days. In case of emergencies, you can up the priority which should be used with care.

Troubleshoot

This section is designed to capture some of the common issues faced by vendors running the certification kit. Our hope is that in time, we will be able to improve the kit to help users avoid the problems being faced, but most of the issues we see are linked to environmental factors. If you are encountering an issue with kit, please take a look at the following failures and their steps for resolution.

My VLAN test network_tests.VLANTestClass.test_vlan_high_port failed.

Please validate the VLAN ID you configured in your network.conf file. The specified VLAN ID (say [200]) should be already configured on the switch. The switch needs to allow XenServer to tag the packets and for it to pass them onwards to the appropriate ports. Make sure the VLAN ID is configured on the switch for all the ports being used by the servers under test.

My MTU test network_tests.MTUPingTestClass.test_ping fails.

This test is to verify whether or not your NICs can support Jumbo frames. If you know your NIC to already support Jumbo frames, then the most likely cause for failure is simply because your switch has not yet been configured for Jumbo frames. Please check this with your respective IT to get the switch configured properly.

If you believe you have configured the switch appropriately, then please also validate manually that Jumbo frames can be passed through the device under test, and attach the logs in your submission.

My Multicast test network_tests.MulticastTestClass fails.

This test is to verify whether your NICs support Multicast feature, and it’s optional, so you can ignore the test result if you don’t want to certify the feature. If you encounter the test is failed, one possible reason is that other network device, like a switch or a router, in your test environment does not support Multicast.

My SR-IOV test network_tests.InterHostSRIOVTestClass, IntraHostSRIOVTestClass1, or IntraHostSRIOVTestClass2 fails?

This test is to verify whether your NICs can support SR-IOV feature, and it’s optional, so you can ignore the test result if you don’t want to certify the feature. There could be multiple factors that would cause the failure. Use the checklist below to identify the real cause.

- Whether the server (including CPU, BIOS, firmware, and PCI bus) supports SR-IOV?
- Whether correct network adapter driver (for PF) has been installed in XenServer?
- Whether correct VF driver is specified in network.conf, and RPM file is already uploaded to folder /opt/xensource/packages/files/auto-cert-kit/ of pool coordinator host?
- Whether the VF driver is applicable to the Droid VM used by server certification kit? If you are not sure whether the driver is applicable, you can import Droid VM into XenServer, start it, install VF driver manually and perform manual verification test at first before running server certification kit. Below is command to import Droid VM:

```
# cd /opt/xensource/packages/files/auto-cert-kit/
# xe vm-import filename=vpx-dlvm.xva
```

- Whether there is sufficient memory and IP addresses available in your test environment? This is concerned because IntraHostSRIOVTestClass2 for SR-IOV will tests the maximum VFs support of one PF.
  With an example, assume the maximum VFs is 63, which means server certification kit will create 11 Droid VMs and pass through all the 63 VFs to them evenly. It’s calculable that 11GB memories and 74 (63 VF interfaces + 11 management interfaces) IP addresses are required.
  Due to hardware limitations or other reasons, you may

not want to test so many VFs even if your NIC supports it. In this case, you are able to specify the maximum VFs to test using key max_vf_num. In the following example, server certification kit will test only 16 VFs instead of original maximum.

```
  [eth0]
  network_id = 0
  vlan_ids = 200
  vf_driver_name = ixgbevf
  vf_driver_pkg = kmod-
ixgbevf-2.16.1-1.el7.elrepo.x86_64.rpm
  max_vf_num = 16
```

Few tests have failed. I don’t want to run the full kit all over again. Is there a way to run just these failed tests?

We provide the following two methods to customize your re-run:

Method 1:

Run a specific category of tests. Your tests are categorized into several groups. You can select to run just one of these groups with an additional argument “–m <TAG>”. Please refer to the following table for relevant tags.

```
./ack_cli.py -n network.conf –m <Tag>
```

Category	Tag
Network Tests	NET

Category	Tag
Operation Tests	All tests (default)
CPU Tests	CPU
Local Storage /IOZone Test	LSTOR
All tests (default)	ALL

Method 2:

Run a specific test class. You can specify any test classes you want to run with an additional argument ‘–o “run_classes=`<TEST CLASS LIST>`”’, where `<TEST CLASS LIST>` is a list of test classes with space as delimiter, and test class is in format of “`<module name>`.`<class name>`”. All module and class names are available in file result.txt.

Examples:

1.Run only Crash Dump test class

```
# ./ack_cli.py -n network.conf -o \
"run_classes=operations_tests.CrashDumpTestClass"
```

2.Run only Multicast test class

```
# ./ack_cli.py -n network.conf -o \
"run_classes=network_tests.MulticastTestClass"
```

3.Run only all SR-IOV test classes

```
# ./ack_cli.py -n network.conf -o \
"run_classes=network_tests.InterHostSRIOVTestClass \
network_tests.IntraHostSRIOVTestClass1 \
network_tests.IntraHostSRIOVTestClass2"
```

4.Run any test classes you want like below showing

```
    # ./ack_cli.py -n network.conf -o \
    "run_classes=network_tests.MulticastTestClass \
    network_tests.IntraHostSRIOVTestClass2 \
    operations_tests.CrashDumpTestClass"
```

Notice and Disclaimer

The contents of this kit are subject to change without notice.

Copyright © 2024 Cloud Software Group Inc. This kit allows you to test your products for compatibility with XenServer products. Actual compatibility results may vary. The kit is not designed to test for all compatibility scenarios. Should you use the kit, you must not misrepresent the nature of the results to third parties. TO THE EXTENT PERMITTED BY APPLICABLE LAW, XENSERVER MAKES AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, AND XENSERVER SPECIFICALLY DISCLAIMS WITH RESPECT TO THE KIT ANY CONDITIONS OF QUALITY, AVAILABILITY, RELIABILITY, BUGS OR ERRORS, AND ANY IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. YOU ASSUME THE RESPONSIBILITY FOR ANY INVESTMENTS MADE OR COSTS INCURRED TO ACHIEVE YOUR INTENDED RESULTS. TO THE EXTENT PERMITTED BY APPLICABLE LAW, XENSERVER

SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL, PUNITIVE OR OTHER DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF INCOME, LOSS OF OPPORTUNITY, LOST PROFITS OR ANY OTHER DAMAGES), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, AND WHETHER OR NOT FOR NEGLIGENCE OR OTHERWISE, AND WHETHER OR NOT XENSERVER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.