Installation Instructions for the

Citrix Hypervisor 8.1 Automated Certification Kit

Version 1.3.4

# Introduction

The Automated Certification Kit (ACK) is an automated test harness for certifying Servers, Network Cards and Local Storage devices for use with Citrix Hypervisor.

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. Citrix 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>

**Notes**:

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

# Prerequisites

In order for the automated certification kit to run successfully, there are currently the following requirements on your Citrix Hypervisor deployment:

* **Pool** - Users must set up a pool consisting of two hosts, both running the version of Citrix Hypervisor 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 ACK must be performed on both the master and the slave 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.

# Citrix Hypervisor Installation

Please download and install Citrix Hypervisor 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 slave:

xe pool-join master-address=<master-ip> master-username=root \

master-password=<pass> force=true

# ACK Installation

Once the above environment has been set up, please download the ch-auto-cert-kit.iso supplemental pack as provided by Citrix, and copy the ISO onto the /root directory of the Dom0 file-system residing on the master. Use the following command to install ACK into all hosts in the pool:

xe update-upload file-name=”/root/ch-auto-cert-kit.iso” --multiple

<it will return update uuid of ACK package on successful upload>

xe update-apply uuid=<update uuid of ACK>

# ACK Operation

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

cd /opt/xensource/packages/files/auto-cert-kit

python 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:

python ack\_cli.py -n network.conf

There is an example file located in the ACK'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.

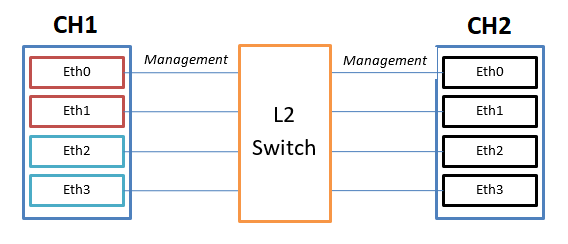
Once executed, the ACK will then generate and execute a list of tests for each device on the master host that should be certified.

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

Citrix Hypervisor 1 (CH1) 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.

If the L2 switch is configured to allow VLAN 200 packets to be passed through between all the connected ports, and all interfaces support SR-IOV, then 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, ACK will write the value, denoted as <name>, into file **/etc/modules-load.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 master host.

If the key is specified, ACK 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 ACK is **/opt/xensource/packages/files/auto-cert-kit/vpx-dlvm.xva**, which is based on upstream CentOS 7.2 X86\_64 and kernel version is 3.10.0. The specified .**rpm** package must be applicable to the Droid VM in use.

The configuration above would allow for the ACK 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

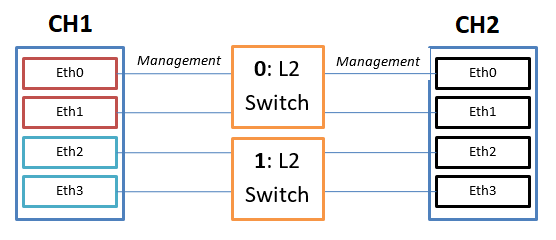
vlan\_ids = 200

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 like this:



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:

[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 Citrix Hypervisor) 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 like so:

Droid VM 1

Eth0

Eth1

Droid VM 2

Eth0

Eth1

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

# 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 ACK 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 completed, there should be two files created in the /root/ directory on the master:

* 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 ACK's log file which is found in /var/log/auto-cert-kit.log. This log file is collected automatically as part of a Citrix Hypervisor 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 Citrix Hypervisor 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
* CitrixHypervisor\_SR-IOV\_Certification\_Form.docx,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 master 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

Citrix is aware of the following limitations in the ACK at present:

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

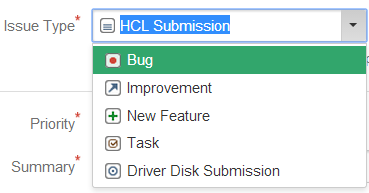
# Bug Reports and Feedback

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’.

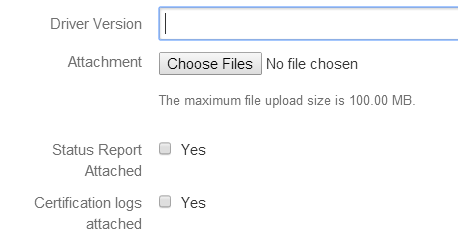
1. Tracker can be found here - <https://tracker.vmd.citrix.com>
2. If your company already has a project open on this system with Citrix, then please raise your issues here. If however, you do not have a specific project for your company, please raise your issues under the ‘**Citrix Hypervisor Hardware Compatibility List (HCL)**’ project.
3. You can get going to create an issue by click “Create Issue” on the Top as shown below



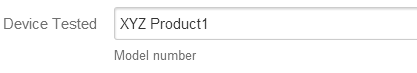
1. Select the issue type as “HCL Submission” if you are looking to certify your hardware on HCL. Similarly the issue type needs to be of the type “Driver Disk Submission” in case for Driver Disks and likewise others.



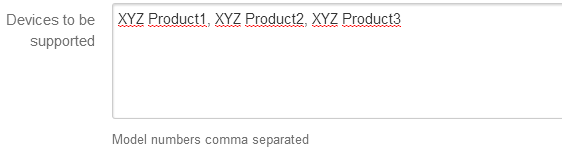
1. Please fill in the details and attach the necessary logs that are required for certification. Driver Version needs to be filled only if your Issue type is of “Driver Disk Submission” else leave it blank.



1. 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.



1. You can also have your equivalent devices certified based on the Device Tested product which can be listed in “Device to be supported” field with comma separated.



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



1. 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 etc for a storage controller.



1. Tickets’ progress can be tracked by clicking on “My Unresolved Reported Issues” on top left corner of your Dashboard.
2. 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 utilised 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.

1. **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 Citrix Hypervisor 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.*

1. **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 Jumo frames can be passed through the device under test, and attach the logs in your submission.*

1. **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.*

1. **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 Citrix Hypervisor?*
* *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 master host*?*
* *Whether the VF driver is applicable to the Droid VM used by ACK?*

*If you are not sure whether the driver is applicable, you can import Droid VM into Citrix Hypervisor, start it, install VF driver manually and perform manual verification test at first before running ACK. Below is command to import Driod 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 ACK 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, ACK 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

1. **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:*

* *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 the table below for relevant tags.*

python ack\_cli.py -n network.conf –m <Tag>

|  |  |
| --- | --- |
| **Category** | **Tag** |
| Network Tests | NET |
| Operation Tests | OPS |
| CPU Tests | CPU |
| Local Storage /IOZone Test | LSTOR |
| All tests (default) | ALL |

* *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*

# python ack\_cli.py -n network.conf –o "run\_classes=operations\_tests.CrashDumpTestClass"

1. *Run only Multicast test class*

# python ack\_cli.py -n network.conf –o "run\_classes=network\_tests.MulticastTestClass"

1. *Run only all SR-IOV test classes*

# python ack\_cli.py -n network.conf -o "run\_classes=network\_tests.InterHostSRIOVTestClass network\_tests.IntraHostSRIOVTestClass1 network\_tests.IntraHostSRIOVTestClass2"

1. *Run any test classes you want like below showing*

# python ack\_cli.py -n network.conf -o "run\_classes=network\_tests.MulticastTestClass network\_tests.IntraHostSRIOVTestClass2 operations\_tests.CrashDumpTestClass"