In this document, we explain how to acquire resources needed to run Performance and Vertical scale tests. It also explains how to use infrastructure preparation scripts to configure vCenter at various stages. This document is not intended to be a cookbook, but rather an outline of steps to help experience engineer to navigate through the stages. Also, it explains how different scripts are connected.

There are several scripts.

1. Python scripts to request and return resources from and to OneCloud.

2. PowerCLI scripts to automate vCenter infrastructure operations.

3. Python scripts to run IXIA performance tests.

4. Set of UNIX shell scripts to test Vertical scale.

You are required to perform manual steps to do NSX Host preparation, Security Policies and Group configuration and partner solution installation.

We assume you already have the appropriate Base vApp provisioned and knowledge on NSX solution operations.

Let’s begin with the steps on how to run performance and vertical scale tests.

1. Open PyCharm  by double clicking on the icon on desktop.

2. On PyCharm, click on the File menu and open certClient.py (This may be already open. If not, follow the steps in the appendix)

2.1 Open certClient.py, locate the line self.base\_url and make sure it is set to:

self.base\_url = **'http://10.148.254.1:8080/devops/webapi'**

3. Open the testif.py file.

4. Change the user\_name and cert\_name as appropriate.

5. Run testif.py.

6. Press 3 to select “Request IXIA resources”.

7. As a result of the request placed in the previous step, there will be a job id generated. It will take several minutes for provisioning the performance test infrastructure.

Press 6 ‘Wait for Job Finish’ and provide the job id from step 6 above. This will poll the backend and inform us when performance resources are fully allocated.

8. Open the vCenter and go to ‘Hosts and Clusters’.

9. Locate “Cert-DC” Data center.

11. Go to F:\scriptRepo. This is already loaded with the latest scripts. To ensure you have the latest scripts, you can pull them by opening the ‘GitHub Desktop’ client (shortcut is available on the desktop) and clicking on Repository->Pull, from the Menu bar.

13. cd to infrascripts-master. There are two directories. perfScripts and scaleScripts.

14. Let’s assume we are running the performance test first.

15. cd to perfScripts directory

16. select perf\_vert\_scale.ps1, right click and select “Run with PowerShell”. This will open a PowerShell window.

17. In the PowerShell window, you will see the following menu

================ Menu ================

1: Enter '1' To Perform Infrastructure Setup

2: Enter '2' To Power up IXIA also move to the Port Group

3: Enter '3' To PowerOff IXIA and move out from Port Group

4: Enter '4' To Remove IXIA VM from the VC inventory

5: Enter '5' To Add vertical Scale VM and enable infra for vertical testing

6: Enter '6' To Add interfaces to Linux VM

7: Enter '7' To Power on Linux VM

8: Enter '8' To Configure IP Address on Linux

9: Enter '9' To Power Off Linux VMS

10: Enter '10' To Remove Linux VMS from the VC inventory

11: Enter '11' To Clear Infrastructure Setup

Q: Enter 'Q' to quit.

Please make a selection:

Option 1 has to be done, once at the start. It is not needed to be repeated for Performance or Vertical tests.

Similarly, Option 11 has to be run only once, after completing all the tests.

Options 2,3,4 belong to Performance tests

Options 5,6,7,8,9,10 belong to Vertical Scale tests.

18. From the menu, enter 1.

This will perform a series of operations: Add ‘esx-perf’ host to the cluster, exit it from maintenance mode, create a new vDS and distributed port groups, scan for storage devices and validate for its presence.

Pay attention to the messages on the screen. Generally, they should go through. If issues (especially if you see messages in red, you may have to manually intervene to fix them).

After completion of the preparation process, original menu is displayed.

19. Go to vCenter and select ‘Networking’ tab.

Note the two vDS relavent to vertical and performance tests. They are

VDS\_performance -> pg is VDS\_performance\_pg

VDS\_vertical\_scale -> pg is VDS\_vertical\_scale\_pg

These VDS and PGs are needed when creating the security profiles for NSX.

VDS\_performance does not have any uplinks i.e., No pNIC attached. It has MTU size of 9000

VDS\_vertical\_scale has one uplink and MTU size is set to 8128.

20. From the menu, enter 2 to “Power on IXIA and move to the Port Group”

21. Follow partner solution integration steps. These may include, but not limited to

Host Preparation on perfCluster, creating Security Groups and Policies. Here, you need to use VDS\_performance\_pg for Performance testing and VDS\_vertical\_scale\_pg for Vertical scale test.

Install the Partner service under NSX Manager > Installation > Service Deployment.

When installing partner solution, it will ask for an IP address. Use IP pool and select ‘SVM\_IP\_pool’.

For the data store, select ‘datastore1’ , which is the local data store on the esx-perf host.

**Note**: For SVM network selection, either select VM Network (‘Specified by host’ option has to be selected here) or VDS\_vertical\_scale, the reason is these have network connectivity. SVMs need management access. VDS\_performance does not have that access. However, protection of traffic happens through Security Policies and there is no bearing on the port group of SVM management interface.

You may need additional steps in different order based on the solution, so please follow partner installation guide and use the port group and cluster information as appropriate.

22. Follow the partner solution instruction and ensure that solution is properly installed. Also ensure that the 6 Ixia\_card VMs deployed as part of the test bed setup are added to the Security Group and are bound to the Security Policy that is using the partner Service Profile. These VMs are the ones to be protected by partner solution and hence this step. Ensure that the rule applied has the traffic flow allowed.

23. Run IXIA performance tests.

PyCharm

In PyCharm, load project “performance” (if not already done).

Make sure H:\ drive is mounted and accessible (Test results will be placed here).

Run loadQuickTestRestApi.py.

Look for any errors on PyCharm console.

At the end of the tests, results are available on H: drive as a PDF file.

NOTE: Ensure that any previously existing PDF files in the H:\ directory are not open. If open, please close them. If Open, the IXIA automation scripts will not copy the files.

Run the test twice. Once, without the partner solution involved and once, with protection from partner solution. This should help us understand the difference in the numbers generated as results, just in case, it comes down to it.

NOTE: When running the test with partner solution, ensure that the policy applied (in PMC) is not blocking any traffic. Ideally, we should create our own policy which has the rules, only to allow traffic and use the same.

24. Once you are satisfied with IXIA tests. Select 3 from the PowerShell Menu. This will power down the IXIA VM and virtual chassis.

25. Select 4 from the PowerShell Menu, to remove IXIA from the vCenter inventory.

Note: On the partner solution, you may also have to remove or mark the IXIA VMs as unprotected before removing from vCenter. Read partner solution instructions.

Next is to prepare for Vertical scale tests. In summary, we perform the following steps.

1. Add 32 Linux-VMs.
2. Add additional 7 interfaces to each Linux-VM to make it 8 interfaces in total.
3. Power on VMs.
4. Configure IP addresses on the new interfaces. Please see section: ”Vertical Scale setup IP Addresses” for IP address structure for the interfaces.
5. Run the ping test for connectivity across all 256 interfaces.

26. Go to PowerShell window that is running infrastructure program and enter 5. This will add 32 Linux VMs. When prompted enter 1 for Starting VM and 32 for Ending VM.

27. After completion of step 26, enter 6 in the selection menu, to add additional interfaces. Enter 7, this will add 7 new interfaces and it will make 8 In total.28. After completion of step 27 above, enter 7 to power on Linux-VMs.

29. After completion of step 28 above, enter 8 to configure IP addresses on newly added interfaces on Linux VMs.

30. Open Git Bash on the Main windows console. “cd f:\git”.

NOTE: By default, the scripts present in F:\Git are the versions bundled when the vApp was initially built. So, we will need to sync a few scripts from F:\scriptRepo\infrascripts-master\perfscripts (latest scripts directory). This should ensure that we have the correct IP addresses that are being pinged. Else, the ping requests will FAIL.31 Run ./ping\_linux\_vm.sh. This will test connectivity to all Linux-VMs newly added. If there are issues, please fix them.

32. If partner solution is not already installed, you may have to install it now. If you have already installed from Performance tests, ensure all the newly added VMs are appearing in the Partner Management Console. Follow the partner instructions and ensure that the correct policies are applied to Linux-VMs.

33. Run the connectivity test.

Open console to Linux-VM1 from vCenter. (Note you can SSH to address 172.16.11.1, however, if you are testing firewalls, the connectivity may get affected. So, recommend option is to use popup console). The username and password to connect to it, are as follows: vmware/VMware1!

Run the following commands.

./ping\_test\_if.sh > results

tail -f results

Check for the desired results. Based on filters, you may see some end points are not reachable.

ping\_test\_if.sh are also bundled with the scripts that you downloaded from Git hub.

You may scp ping\_test\_if.sh to other Linux-VM machines and perform similar connectivity tests if you desire.

Now, you have completed the tests. Before removing infrastructure, please follow partner uninstall instructions and uninstall it from the vCenter. This includes deletion of Security Group/Policy, Un-deployment of the Service VM, GI (if applicable), Service un-registration etc.,)

34. Go to PowerShell window where we are running the infrastructure script and enter 9, to power off Linux-VMs.

35 After completion of step 34, enter 10 to remove Linux VMs from the vCenter inventory.

36. After completion of step 35, enter 11 to clear the remaining configuration from the vCenter.

37. Finally you have to return the resources borrowed from one cloud. This is done by:

Open the PyCharm project and run certClient.py (if not already running).

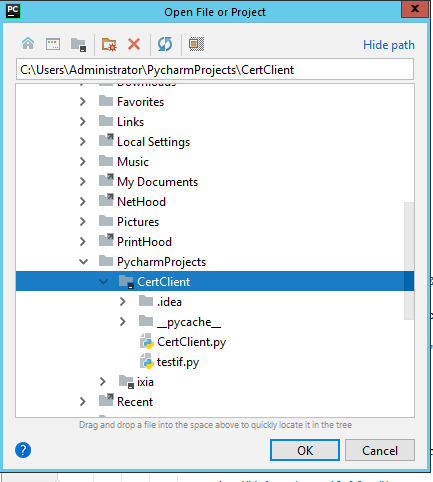
Select option 4 “Remove IXIA Resources”.

Congratulations, you have completed Performance and Vertical Scale tests of NSX scale test plan.

**Appendix:**

Open PyCharm utility

From File menu, select Open and navigate to CertClient project as below.



Click OK.

From the file tab, select testif.py.

From the RUN menu, select RUN and then select testif.py.

For IXIA Performance testing,select Open from the File menu and navigate to IXIA project as below and click OK.



Notes:

When creating vApp template, select **make identical copies** radio button. Please do not use **customize VM** option.

Preparing ixconsole for automation.

Install IXnetworks.

REST APIs are executed through IxNetwork Connection Manager. Hence we want it to be started automatically at boot up (note: Windows at startup, means someone needs to login. For our purpose, it is at boot up. We do not want anyone to login). Option 2 is the recommended way.

Option 1:

Instructions on how to run an application at boot up are listed in below references:

<https://www.howtogeek.com/50786/using-srvstart-to-run-any-application-as-a-windows-service/>

<http://www.rozanski.org.uk/services_quick>

In caseyou need to delete the service, instructions are below.

<https://www.howtogeek.com/howto/windows-vista/how-to-delete-a-windows-service-in-vista-or-xp/>

Option 2:

Note: From IxConnection manager window, you can make it a service and/or remove it as a service. This will place IXConnection manager as a service and above Option 1 is not needed. Included here for historical reasons and information.

Debugging:

When IXConnection manager is running as a service, you can access its console using <http://localhost:8008> or <http://ixconsole:8008> or 192.168.110.96:8008.

Login using local account : username **debug** and password **VMware1!**

We have placed ixia.ini file in C:\, please refer to that for configuration.

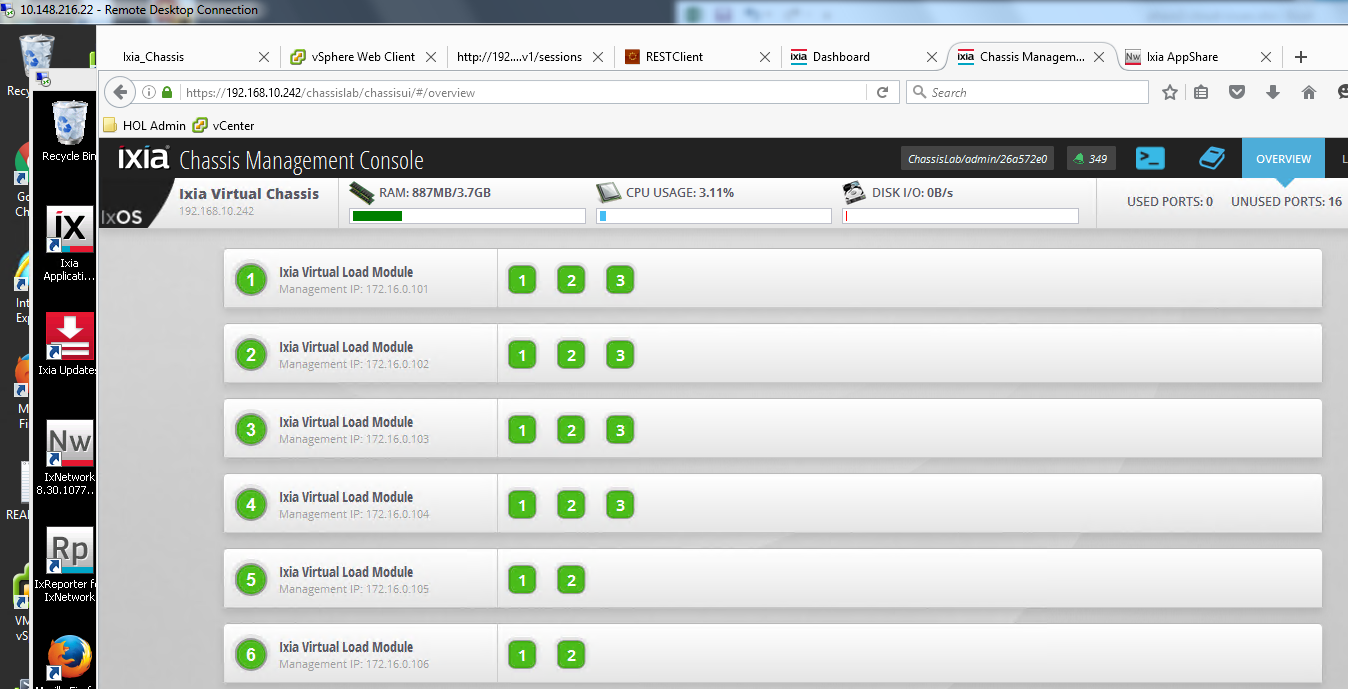
Also create a backdoor local account debug. You can login to this with ixconsole\debug as username. Password is VMware1!

IXIA seems to export results to local C:\ drive.

We write in to the C:\Results directory. However, we are running automation from Main console. We want people to get results without logging in to ixconsole machine. Hence export(share) C:\Results directory. On the main console, mount this as a remote H:\ drive.

Make sure all the ports are up on IXIA chassis.

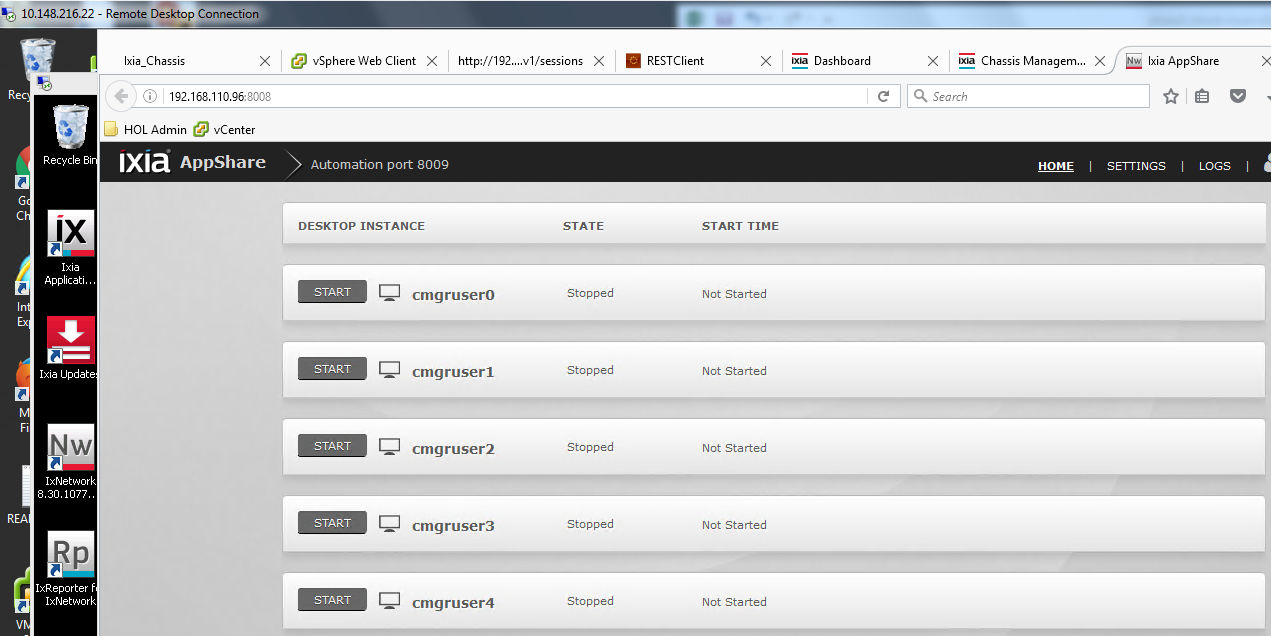
Login details for IXIA Chassis – 192.168.10.242 - admin / admin



Make sure all connections are in START mode, as shown below.

Login to IXIA Chassis – 192.168.110.96:8008 - debug / VMware1!

We will be using only the first connection manager.



Additional IXIA notes

10. (optional) Open remote desktop session to ixconsole.corp.local and login as CORP\Administrator, and use password VMware1!

11. (optional) Minimize the remote desktop connection to ixconsole

12 (optional) From the original remote desktop session to the main console Verify that the ixconnection manager is up and running by cutting and pasting the following URL

<http://192.168.110.96:11009/api/v1/sessions>

This will give the output of sessions that are available on the IX Connection manager.

To manually start a session: (below is just for reference you do not need to do manual steps)

POST [http://192.168.110.96:11009/api/v1/sessions/<ID](http://192.168.110.96:11009/api/v1/sessions/%3cID)>

POST [http://192.168.110.96:11009/api/v1/sessions/<ID>/operations/start](http://192.168.110.96:11009/api/v1/sessions/%3cID%3e/operations/start)

STOP

POST [http://192.168.110.96:11009/api/v1/sessions/<ID>/operations/stop](http://192.168.110.96:11009/api/v1/sessions/%3cID%3e/operations/stop)

Delete a session

DELETE [http://192.168.110.96:11009/api/v1/sessions/<ID](http://192.168.110.96:11009/api/v1/sessions/%3cID)>

ID : 8020-8025

13. On PyCharm and load the project “performance”.

14: Make sure H:\ drive is mounted and accessible (Your test results will be here).

14. Run loadQuickTestRestApi.py.

15. Look for any errors and results should be available on H: drive.

**Vertical Scale setup IP Addresses**

Vertical scale setup is intended to validate number of filters (interfaces) that a given SVM can support in a single host.

Each Linux VM will have 8 interfaces and tshere are 32 Linux-VMs to give a total of 256 interfaces (filters).

Let’s assume there are 8 interfaces for each host and that they are numbered from 1-8 and denoted by (i).

Let’s assume there are 32 hosts and they are numbered from 1-32. Let’s assume it is denoted by (h).

There is a separate subnet for each interface. The subnet is 172.20.(i).0/24.

Host in each subnet is will have address of 172.20.(i).(h).

As an example, consider Linux-VM2. It will have following IP addresses

For card 2, 172.20.1.**2**

For card 3, 172.20.2.**2**

For card 4, 172.20.3.**2**.

and so on – last field is the host id.

On Linux-VM3,

For card 2, 172.20.1.**3**

For card 3, 172.20.2.**3**

For card 4, 172.20.3.**3**

and so on – last field is the host id.

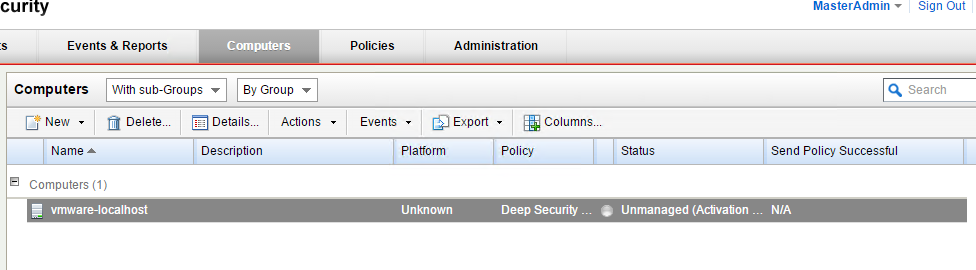
Note: Interface number one is already configured with 172.16.10/23 subnet. Please see scale documentation for the details.

Linux-VM1 is 172.16.11.1, Linux-VM2-32 are 172.16.10.2 to 172.16.10.32.

Note: When creating the clusters, please enable DRS and set to manual.

This will create ESX Agent folder and locate Service Agents such as GI and SVM in the folder. Easier to manage and navigate GUI. Idea here is these agents are non vMotionable.

If you are installing TrendMicro on control center, after installation please remove/deactivate control center from protected computer. Otherwise you will not be able to access vCenter through GUI.



Note: After the interfaces are added and new IPs configured, it updates the .vmdk file in the VM-datastore. After this, even if you delete the VM from the inventory and add it back, you will have the configured IP.

Running CPU bench mark:

Performance numbers on virtual platform depends on the power of underlying resources. Any number to be meaningful, the results have to be published with the performance data, so readers can relate to their platforms with easily available meaningful set of scores.

We propose to use Geekbench <https://www.geekbench.com>

Create a Windows VM. Allocate 14 CPU and 2 sockets.

Make minimum CPU reservation to 30,000.

Select Expose Hardware Virtualization to Host.

Select Allocate all Host Memory.

From Advanced options, set Latency sensitivity to High.

Install Geekbench and run it.

Note down the Score for single core and multi core.

Also, detail score report is archived an accessible through browser. Example URL is here

<https://browser.geekbench.com/v4/cpu/5874391>

Please review the screen shot from CPU configuration. See Reservation is set to 30,000 Mhz, and also shares are set to 30,000.

Git Hub access details

Username: ecocert

Password: VMware1!

**URLs**

https://github.com/ecocert/infrascripts

https://github.com/ecocert/infrascripts.git

