Table of Contents

[1 Revision History 1](#_Toc529198281)

[2 Test execution workflow 2](#_Toc529198282)

[2.1 Vertical Scale - Test bed setup and test execution 2](#_Toc529198283)

[2.1.1 Vertical Scale setup (IP Addresses) 5](#_Toc529198284)

[2.2 Callback testbed setup and test execution 7](#_Toc529198285)

[2.3 Steps for running Horizontal Scale tests 11](#_Toc529198286)

[Appendix: 19](#_Toc529198287)

[Steps to be followed, to increase the disk size on the Storage VMs 19](#_Toc529198288)

[Steps to add the new disk with size, as needed 20](#_Toc529198289)

[Miscellaneous 22](#_Toc529198290)

[Troubleshooting 26](#_Toc529198291)

# 1 Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Who** | **When** | **Why** | **Status** |
| HCLT Team | Sep-2018 | Consolidated all 3 existing test execution docs into one, maintaining the same test execution workflow as followed currently. | Completed |
| HCLT Team | Nov-2018 | Since Performance test is ‘Not Applicable’ anymore, removed all references to it. | Completed |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 2 Test execution workflow

In general, we start with the Vertical Scale and Callback tests, before moving on to the Horizontal Scale test.

# 2.1 Vertical Scale - Test bed setup and test execution

In this section, we cover the steps on how to acquire resources needed to run Vertical scale tests. It also explains how to use infrastructure preparation scripts to configure vCenter at various stages. This documentation is not intended to be a cookbook, but rather an outline of steps to help the 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. 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.

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 Locate the line self.base\_url and make sure it is set to:

self.base\_url = **'**[**http://10.148.254.1:8080/devops/webapi**](http://10.148.254.1:8080/devops/webapi)**'**

3. Open testif.py file.

4. Change the user\_name and cert\_name as appropriate. By default, they are

user\_name = **"Tissa"**cert\_name = **"netx\_6.3"**

5. Run testif.py. It lists down a set of options to choose and execute.

6. Choose 3 to select “Request IXIA resources” and hit Enter.

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 test infrastructure.

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

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

9. Locate “Cert-DC” Datacenter.

10. Go to F:\scriptRepo. This is loaded with the test 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.

11. Go to F:\scriptRepo\infrascripts\scalescripts. **Right click on the script named "perf\_vert\_scale.ps1" and select Edit. This opens a PowerShell console.**

12. Run it and you will see the below output: (please ensure that you are running it the same way as mentioned here (highlighted in red above), as PowerShell might work differently depending on how the script is run).

================ 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 interfaces From Linux VM

11: Enter '11' To Remove Linux VMS from the VC inventory

12: Enter '12' To Clear Infrastructure Setup

Q: Enter 'Q' to quit.

Please make a selection:

Options 1,12 perform the infrastructure setup and clean up respectively. Ignore the Options 2,3,4.

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

13. From the menu, enter 1.

This will perform a series of operations such as: 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 their presence.

Pay attention to the messages on the screen. Generally, they should go through. If there are 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 again.

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

Note the vDS relevant to the Vertical tests.

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

This VDS and PG is needed when creating the security profiles for NSX.

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

15. Setup the partner solution and configure it accordingly. These may include, but not limited to -

Partner Manager installation, Service registration, Host Preparation on perfCluster, creating Security Groups and Policies etc., Here, you need to use VDS\_vertical\_scale\_pg for Vertical scale test.

Install the Partner service by going to Networking and Security> Installation > Service Deployment in the vSphere client.

When installing the 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, select VDS\_vertical\_scale, the reason is that this has network connectivity. SVMs need management access.

16. You may need additional steps to be performed, depending on the design of the partner solution and hence, follow the partner installation guide and use the port group, cluster information as appropriate.

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

* Add 32 Linux-VMs.
* Add additional 7 interfaces to each Linux-VM to make it 8 interfaces in total.
* Power on these VMs.
* Configure IP addresses on the new interfaces. Please refer section [Vertical Scale setup - IP Addresses](#_2.1.1_Vertical_Scale)**,** for IP address structure for the interfaces.
* Run the ping test for connectivity across all 256 interfaces.

17. Go to the PowerShell window that was triggered in step 12 and enter 5. This will add 32 Linux VMs. When prompted, enter 1 as the index for Starting VM and 32 as the index for Ending VM.

18. After completion of step 17, enter 6 in the selection menu. This will add the additional interfaces.

19. After completion of step 18 above, enter 7 to power on the Linux-VMs.

20. After completion of step 19 above, enter 8 to configure IP addresses on the newly added interfaces on the Linux VMs.

21. 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\perfscripts (latest scripts directory). This should ensure that we have the correct IP addresses that are being pinged. Else, the ping requests will FAIL.

22 Run ./ping\_linux\_vm.sh from Linux-VM1. This will test the basic connectivity to all Linux-VMs newly added. If there are issues, please fix them.

23. If partner solution is not already installed, you may have to install it now. Ensure all the newly added VMs are appearing in the Partner Management Console. Follow the partner instructions and ensure that the correct policies (to allow traffic) are applied to Linux-VMs.

24. 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\_32\_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\_32\_if.sh are also bundled with the scripts that you downloaded from Git hub.

You may scp ping\_test\_32\_if.sh to other Linux-VM machines and perform similar connectivity tests if you desire.

This completes the Vertical Scale test. Back up the **results** file for report generation.

Before cleaning up the infrastructure, ensure that the Callback test is also completed. Please refer the section [Callback testbed setup and test execution](#_2.2_Callback_testbed)for detailed instructions on the same.

Once the Callback test execution is done, undeploy the partner SVM, GI (if applicable), clean-up the Security Group/Policy etc.,). DO NOT UNREGISTER THE SERVICE.

# 2.1.1 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 there are 32 Linux-VMs to give a total of 256 interfaces (filters).

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

Let’s assume there are 32 Linux VMs 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.

Each subnet will have an 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 # in the VM name. In the above example, the # is 2, as the VM involved is Linux-VM2.

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.

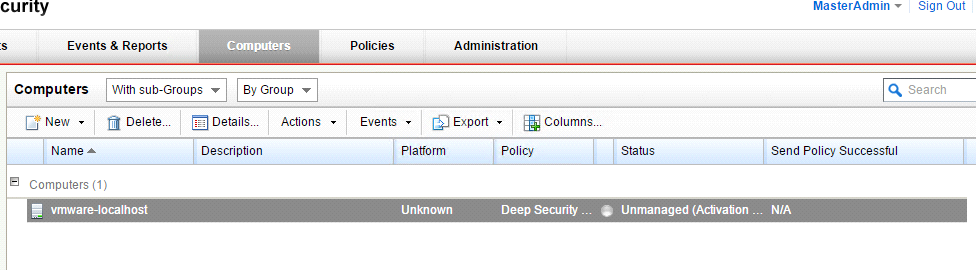
Note: Interface number one is already configured with 172.16.10/23 subnet. Please see scale documentation for the details.

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 it 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 (Main console VM), 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.

**Git Hub access details**

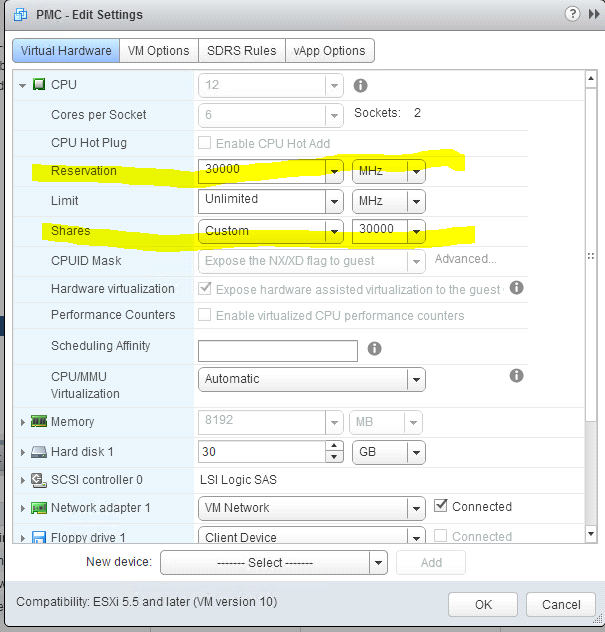
Username: **ecocert**

Password: **VMware1!**

**URLs**

<https://github.com/ecocert/infrascripts>

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



# 2.2 Callback testbed setup and test execution

In this section, we cover the steps on how to run the callback or control scale tests.

There are two scripts.

* Python scripts to setup and clean-up configuration
* powerCLI scripts to get Guest VM IP-address.

As mentioned in **page 5**, Callback test execution is expected to be done, before cleaning up the Vertical Scale test infrastructure.

The steps given below are to be followed for Callback test execution.

* Create a Security Group (SG1) with one of the Linux-VMs as member. Create a Security Policy as given below, for the testing.

A sample Security Policy:

*Create a Security Policy with the following configuration as a template.* ***Note*** *the Services are* ***optional*** *as well. We are showing 5 services here (ICMP -3, Syslog -2) for educational purpose only.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Network Introspection Services** | **Source** | **Destination** | **Action** | **Service** | **Service Name** | **Service Profile** |
| SP1 | Name - inbound | Policy's Security Group | SG1 | Redirect to Service | ICMP Echo | Partner Service Name | Partner Service Profile |
|  |  |  |  |  | ICMP Redirect |  |  |
|  |  |  |  |  | ICMP Destination U |  |  |
|  |  |  |  |  | Syslog (UDP) |  |  |
|  |  |  |  |  | Syslog (TCP) |  |  |
|  | Name - outbound | SG1 | Policy's Security Group | Redirect to Service | ICMP Echo | Partner Service Name | Partner Service Profile |
|  |  |  |  |  | ICMP Redirect |  |  |
|  |  |  |  |  | ICMP Destination U |  |  |
|  |  |  |  |  | Syslog (UDP) |  |  |
|  |  |  |  |  | Syslog (TCP) |  |  |

* Open the powerCLI script

infrascripts\testscripts\CallbackScale362\GetAllVM\_IPAddr.ps1

Make sure that the prefix is matching the Guest VM names

$VMPrefix = "Linux-VM\*"

***PS: Also validate the VC IP address, username/password.***

Get all Guest VM IP-address. Run PowerCLI script

cd infrascripts\testscripts\CallbackScale362\

PS > .\GetAllVM\_IPAddr.ps1

This will generate a file with all valid Guest VMs object-ids.

…\input\gvm\_object\_ids.txt

This will also generate a log file with more details.

…\output\ps\_gvm\_objects.log

* Open PyCharm  by double-clicking on the icon on desktop.
* Open the python script

infrascripts\testscripts\CallbackScale362\constants.py

Make sure that SCALE\_COUNT\_TOTAL & SG-Binding are set to 1000 and 64 respectively.

infrascripts\testscripts\CallbackScale362\constants.py

SCALE\_COUNT\_TOTAL = 1000

#Set SG\_BINDING\_COUNT less than 128

SG\_BINDING\_COUNT = 64

The script will

* create SCALE\_COUNT\_TOTAL Security Groups and Security Policies.
* SG/SP binding is done as per the value provided for SG\_BINDING\_COUNT.

***PS: Also validate the NSX IP address, username/password.***

* Run python script CallbackScaleTests.py

Call Option-1 to do the setup. As part of the Callback test setup, the script creates SCALE\_COUNT\_TOTAL(=1000) Security Groups, and SCALE\_COUNT\_TOTAL(=1000) Security Policies. Every Security Policy includes 5 policy rules as mentioned above. The 5 rules are hard-coded in the script.

infrascripts\testscripts\CallbackScale362\CallbackScaleTests.py

1. CB Scale Config Setup

2. CB Scale Config Cleanup

Enter Your Choice: 1

Started Callback Scale Setup...

SG CREATION : success\_count=1000 failure\_count=0

SP CREATION : success\_count=1000 failure\_count=0

TOTAL SP CREATION time is : 1349.0105699188025 seconds ...

This will also generate a log file with more details.

Sample LOG contents: output\ cb\_script.log

…\output\ cb\_script.log

12-02-2018:18:39:07,35 INFO [CallbackScaleUtils.py:230 :cb\_scale\_sec\_group\_create] CREATED SG Security\_Group\_5001 object-id = securitygroup-971

12-02-2018:18:39:07,278 INFO [CallbackScaleUtils.py:230 :cb\_scale\_sec\_group\_create] CREATED SG Security\_Group\_5002 object-id = securitygroup-972

12-02-2018:18:39:07,499 INFO [CallbackScaleUtils.py:230 :cb\_scale\_sec\_group\_create] CREATED SG Security\_Group\_5003 object-id = securitygroup-973

13-02-2018:14:24:05,489 ERROR [CallbackScaleUtils.py:898 :cb\_scale\_sec\_policy\_apply] APPLYING SP policy-1195 status code = 500

13-02-2018:14:24:05,884 ERROR [CallbackScaleUtils.py:898 :cb\_scale\_sec\_policy\_apply] APPLYING SP policy-1196 status code = 500

14-02-2018:16:28:43,481 INFO [CallbackScaleUtils.py:799 :cb\_scale\_sec\_policy\_delete] DELETED SP policy-499

14-02-2018:16:28:47,365 INFO [CallbackScaleUtils.py:799 :cb\_scale\_sec\_policy\_delete] DELETED SP policy-500

14-02-2018:16:28:50,681 INFO [CallbackScaleUtils.py:799 :cb\_scale\_sec\_policy\_delete] DELETED SP policy-501

* The script will generate CSV files for the time taken.

Security Policy Creation: output\ sp\_creation\_time.csv

Security Policy Apply: output\ sp\_apply\_time.csv

These files can be used to plot the graphs.

* Validate that all Security Groups, and Security Policies are created as expected.

If for some reason, any of the Security Policy’s status is “Failed”, then please manually resolve it.

Click the Policy and in the pop-up hit “Resolve All”.

* Run python script CallbackScaleTests.py

Call Option-2 to do the Clean-up. As part of the Callback test clean-up, the script will use all the object-ids of Security Groups, Security Policies already stored in the /output/ temporary files and does the clean-up.

***PS: Please don’t keep any .../input/ or .../output/ directory files open during the python scripts running time, otherwise the cleanup might not delete those temporary files and you might have issues in next runs.***

infrascripts\testscripts\CallbackScale362\CallbackScaleTests.py

1. CB Scale Config Setup

2. CB Scale Config Cleanup

Enter Your Choice: 2

Started Callback Scale Cleanup...

Completed Callback Scale Cleanup. Done

This will also update the log file with more details.

…\output\ cb\_script.log

Once done with the test execution, **ensure that the results are backed up**. This includes, but not limited to, cb\_script.log, sp\_apply\_time.csv, sp\_creation\_time.csv etc.,

* Tear down the setup.

To do so, go to PowerShell window where we are running the infrastructure script and

1. enter 9, to power off Linux-VMs.
2. enter 10, to remove the additional interfaces from the Linux VMs.
3. enter 11, to remove Linux VMs from the vCenter inventory.
4. enter 12, to clear the remaining configuration from the vCenter.
5. Finally, you must return the resources borrowed from one cloud. Follow the steps below, to do so.
6. Open the PyCharm project and run testif.py (if not already running).
7. Select option 4 “Remove IXIA Resources”.

# 2.3 Steps for running Horizontal Scale tests

**Setting up the horizontal scale test environment**

1. In the main console VM, Open PyCharm  by double clicking on the icon in the desktop.
2. Click on the File menu.
   1. Open certClient.py. Locate the line

self.base\_url = **'**[**http://127.0.0.1:9090/devops/webapi**](http://127.0.0.1:9090/devops/webapi)**'**

and change it to

self.base\_url = **'**[**http://10.148.254.1:8080/devops/webapi**](http://10.148.254.1:8080/devops/webapi)**'**

* 1. Open testif.py.
  2. Change the user\_name and cert\_name as appropriate. By default, they are:

user\_name = **"Tissa"**cert\_name = **"netx\_6.3"**

* 1. Run testif.py.

1. Press 1 to select “Request ESX For Scale”. This will add a new disk to the FREENAS Storage VMs “stgb-01a-1”, “stgb-01a-2”, “stgb-01a-3”, “stgb-01a-4”, which will serve as a storage for partner SVMs. As of now, we are considering SVM size as 20GB and number of SVMs as 256.

20GB\*256 = 5120GB of total disk space for SVMs allocation. 5120GB/4 = 1280GB. On each of the FREENAS VMs, 1280 GB will be added as a new disk in the above operation.

**Please review the Partner SVM size and get the storage extended accordingly, before doing "Request ESX For Scale" operation. If the SVM size is larger than the default SVM size considered (i.e., 20 GB), the disk size on the Storage VMs needs to be increased. Refer the section ”**[Steps to be followed, to increase the disk size on the Storage VMs](#_Steps_to_be)**” , for details on how to get it done.**

1. As a result of the request posted in step 3, a Job ID will be generated. It will take several minutes for provisioning of the Scale infrastructure. Press 6 and provide the Job ID that was generated. This will poll the backend and inform us when the scale resources are fully allocated.
2. There are 64 ESX Horizontal hosts in the vApp. These are named as
   * 1. esx-horizontal-1 to esx-horizontal-32 and
     2. esx-horizontal-1-1 to esx-horizontal-32-1
3. There are 4 nested ESXi hosts running on each of these horizontal hosts. Hence, there are a total of 256 nested ESXi hosts.
4. Go to F:\scriptRepo\infrascripts\scalescripts. **Right click on the script named "setup\_scale\_testbed.ps1" and select Edit. This opens a PowerShell console.** Run it and you will see the below output: (please ensure that you are running it the same way as mentioned here (highlighted in red). PowerShell works differently depending on how the script is run).

================ Test bed setup for Horizontal Scale testing ================

1: Press '1' for checking whether Horizontal hosts are in UP state

2: Press '2' for powering on the scale ESXi hosts

3: Press '3' to add the scale hosts into vCenter

4: Press '4' to register Linux VMs into Scale hosts in the vCenter

5: Press '5' to add the scale hosts into vDS and migrate Linux VM network to VDS

6: Press '6' to perform storage volume, extent and target creation in the FREENAS Servers

7: Press '7' to do the storage scan on the scale hosts

8: Press '8' to remove scale hosts from vDS

9: Press '9' to unregister Linux VMs from vCenter

10: Press '10' to remove scale hosts from vCenter

11: Press '11' to shutdown scale hosts

12: Press '12' to power on linux VMs

13: Press '13' to shutdown linux VMs

14: Press '14' to Prepare Host Clusters for NSX

15: Press '15' to Deploy NSX Guest Introspection or Partner Service

16: Press '16' to Monitor and Resolve alarms raised in Guest Introspection or Partner Service deployment

Q: Press 'Q' to quit.

Please make a selection:

1. Execute the options 1-5, one after the other. After completing step 5, all the Scale hosts will be added to the vCenter and the vDS.
2. The hostnames of the nested ESXi hosts are named as esx-scale-1.corp.local to esx-scale-256.corp.local.
3. Each of these nested ESXi hosts has a Tiny Core Linux Guest VM, which is mounted on the “VM-datastore”. All the Guest VMs can be accessed using **root/VMware1!**
4. After the hosts are added to the vCenter, check if the Guest VMs are mounted properly. If not, execute the action "7: to do storage scan on the scale hosts". This will scan and mount the "VM-datastore" on all the Scale hosts. This will make the Guest VMs visible.

**NOTE**: To check if the Guest VMs are mounted properly, one must go to each cluster and examine Tiny-Linux-VM-(x) where x ranges from 1 to 256. If you see "inaccessible" next to any of the Linux-VMs, run the storage scan.

1. Power ON all the Guest VMs.

Please do the following, before powering on the VMs.

As the ScaleClusters have DRS enabled, vCenter gives migration recommendation for VMs during power ON. To prevent this, set the DRS automation level to "Disabled" for all the virtual machines in all the clusters. The link below will guide you do it for each ScaleCluster.

<https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.resmgmt.doc/GUID-B560341B-B377-4FA7-BF3B-98A4788AAE3A.html>

Select option "12" in the scale test bed menu, for powering ON the Linux VMs.

1. Mounting new storage volume in the FREENAS server for storing partner SVM.
2. Select option '6' to perform storage volume, extent and target creation in the FREENAS Servers.

You will see the following output

* 1. There are 4 FREENAS Servers with the following index-IP mapping,
  2. 1-192.168.110.61, 2-192.168.110.62, 3-192.168.110.63, 4-192.168.110.64

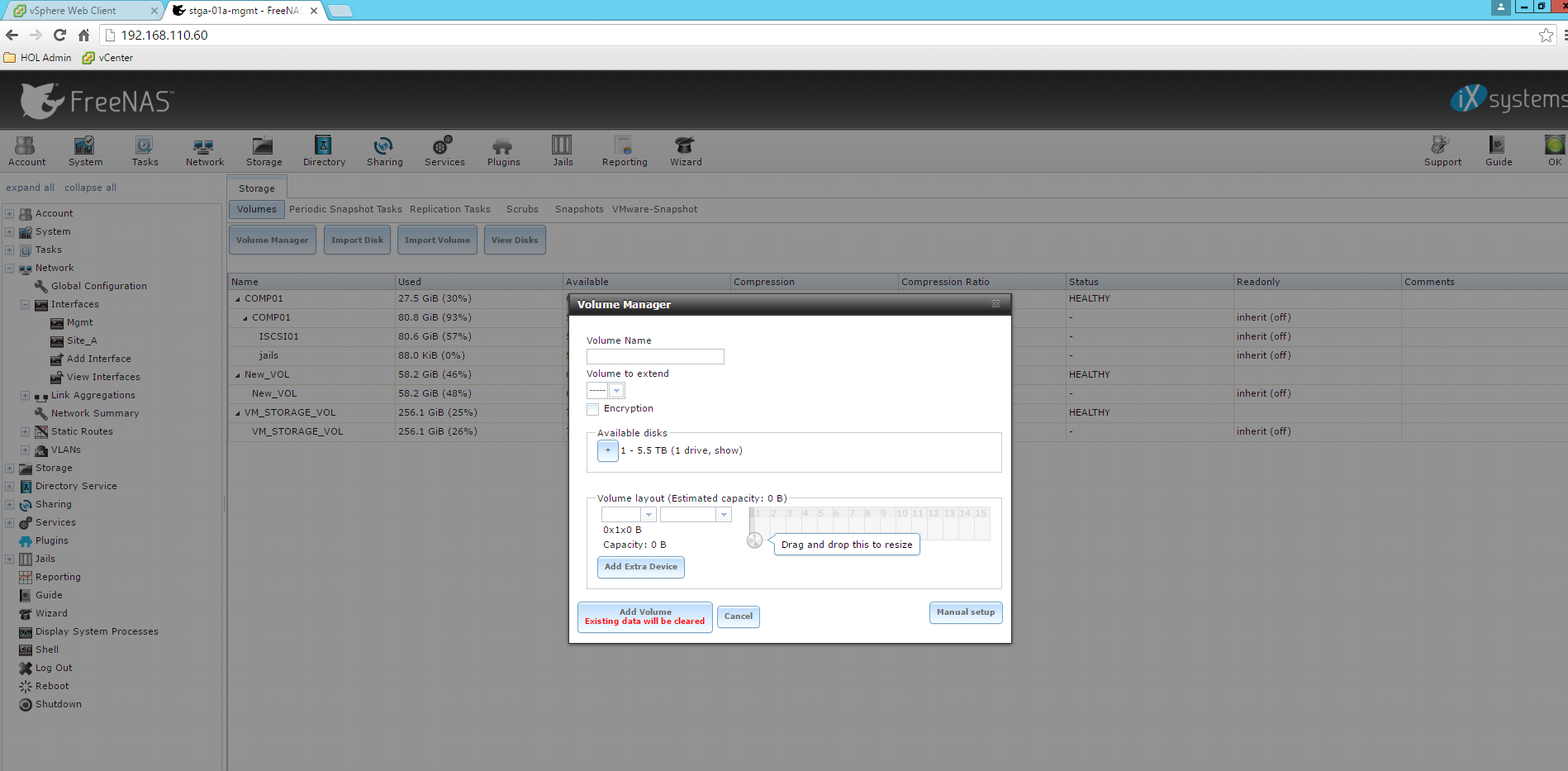
It will ask for start and end FREENAS Server index.

If you enter start as 1 and end as 4, it creates storage volume, extent and target to extent in all the FREENAS Servers.

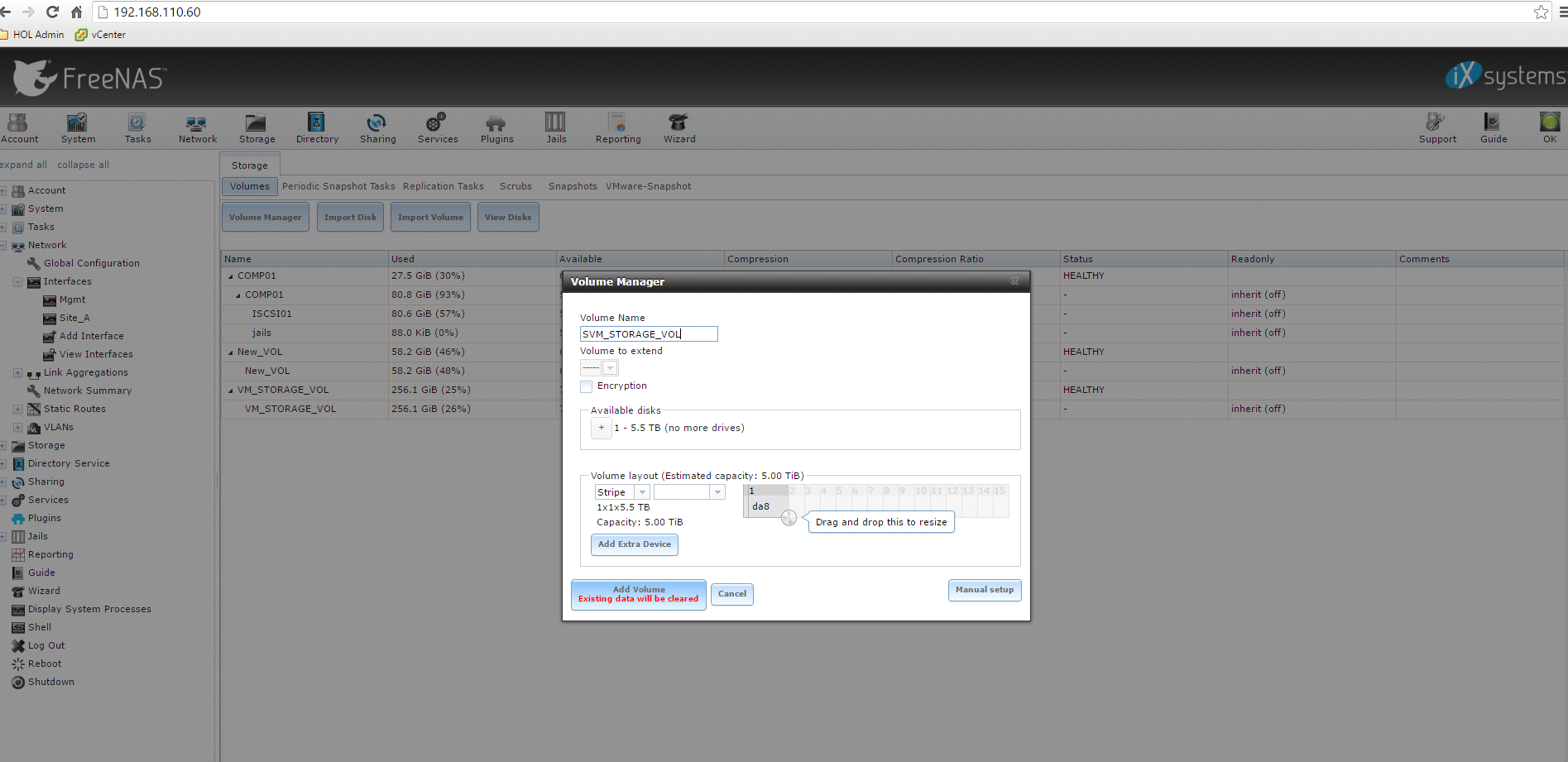
**Please note the below section is a graphical way of carrying out storage volume, extent and target to extent creation operations in the FREENAS Servers. It can be ignored as we are achieving the same in step 14 above.**

Open the first FreeNAS WebUI by typing [http://192.168.110.61/](http://192.168.110.60/) in the web browser. Login with root/VMware1!

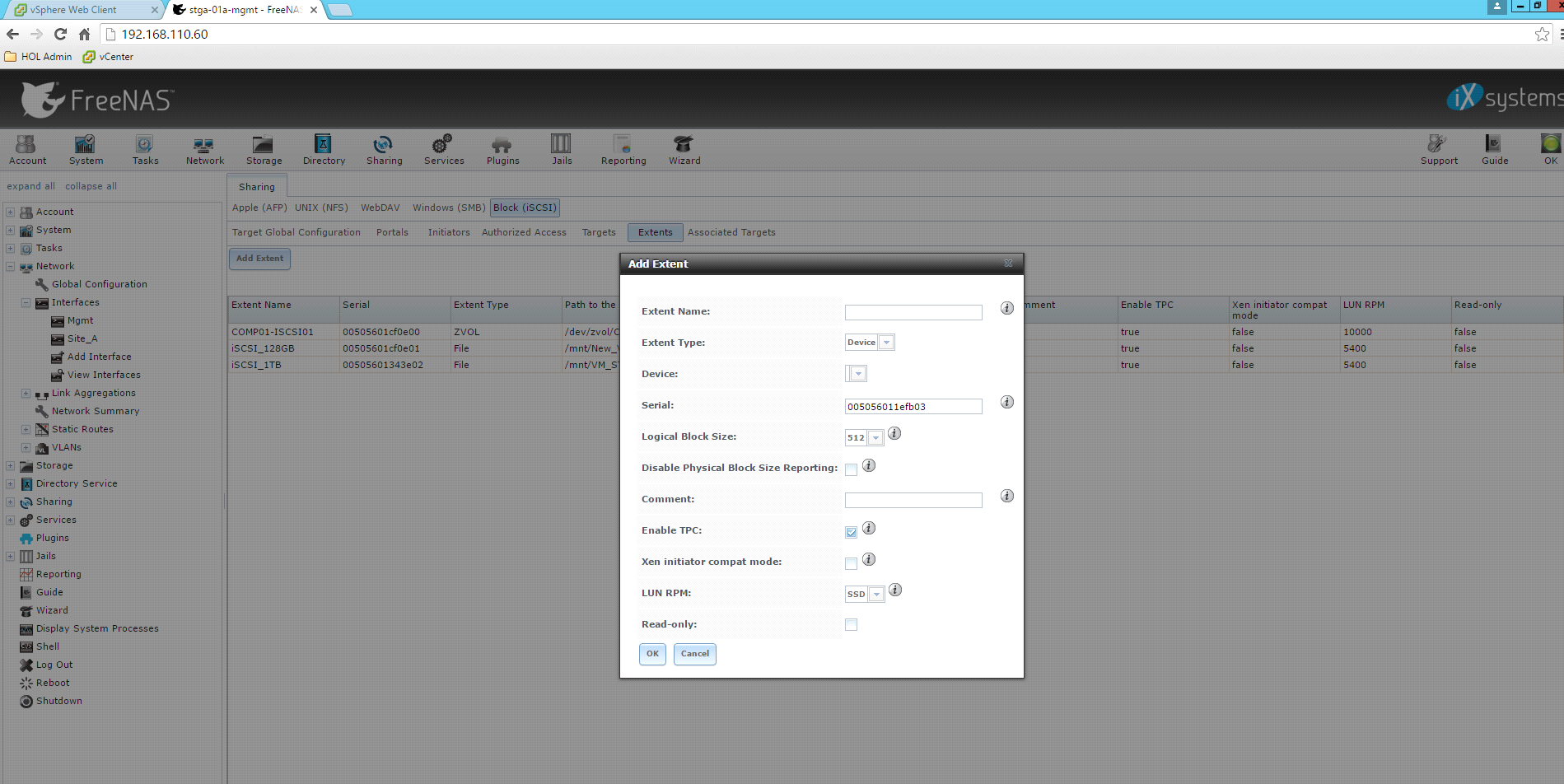
Go to Storage and click on “Volume Manager”. You will see the new disk of size X TB (X range from 1 –4).



Click on + sign under Available disks, give volume name as “SVM\_STORAGE\_VOL” and click on “Add Volume”. This will create a new volume of size XTB.

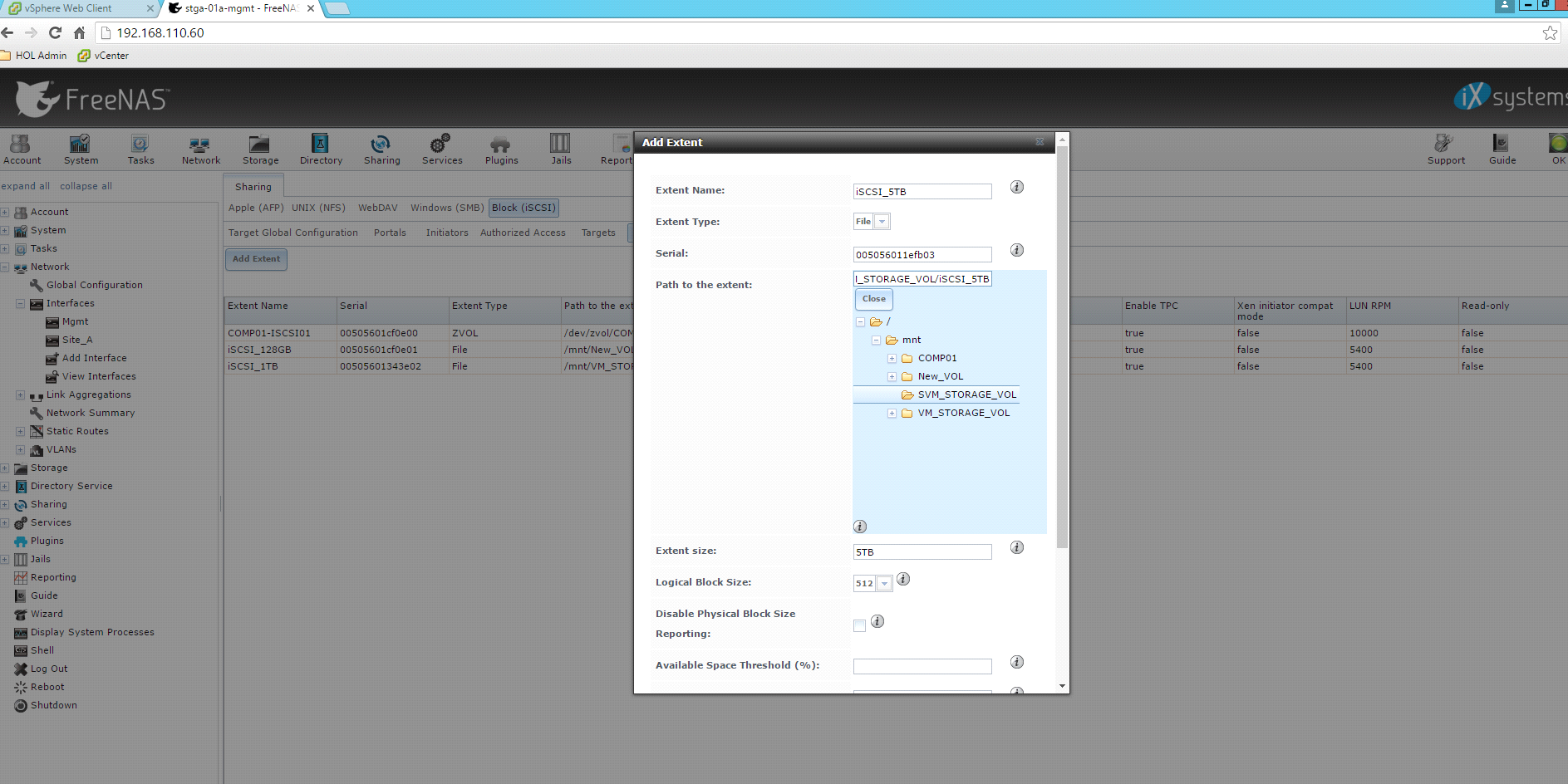


Go to Sharing->Block(iSCSI) -> Extents and click on Add Extent.

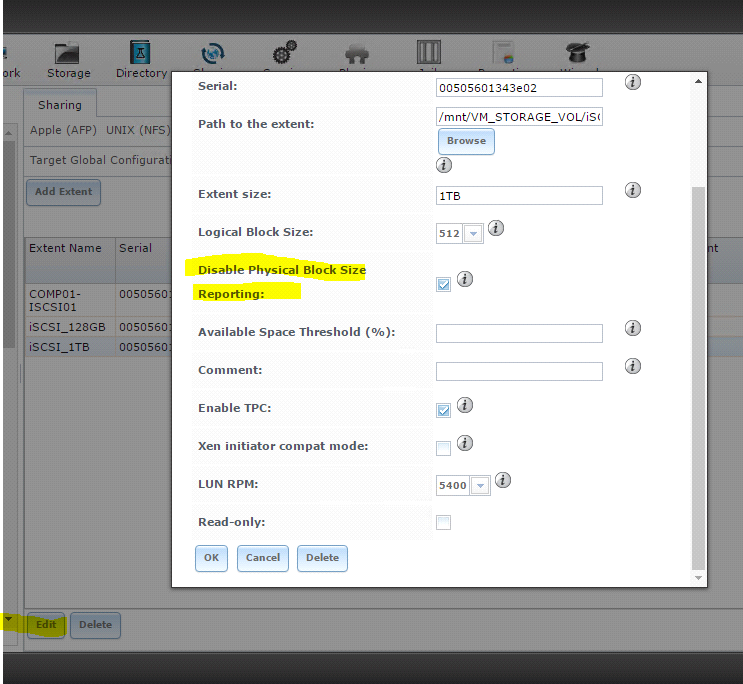


Enter the extent name as “iSCSI\_2TB” (assuming the disk size is 2 TB), select Extent Type “File”, path to extent as “/mnt/SVM\_STORAGE\_VOL/iSCSI\_2TB” and give appropriate extent size, 2TB in this case. Click on “OK”. This will create a new Extent “iSCSI\_2TB”.

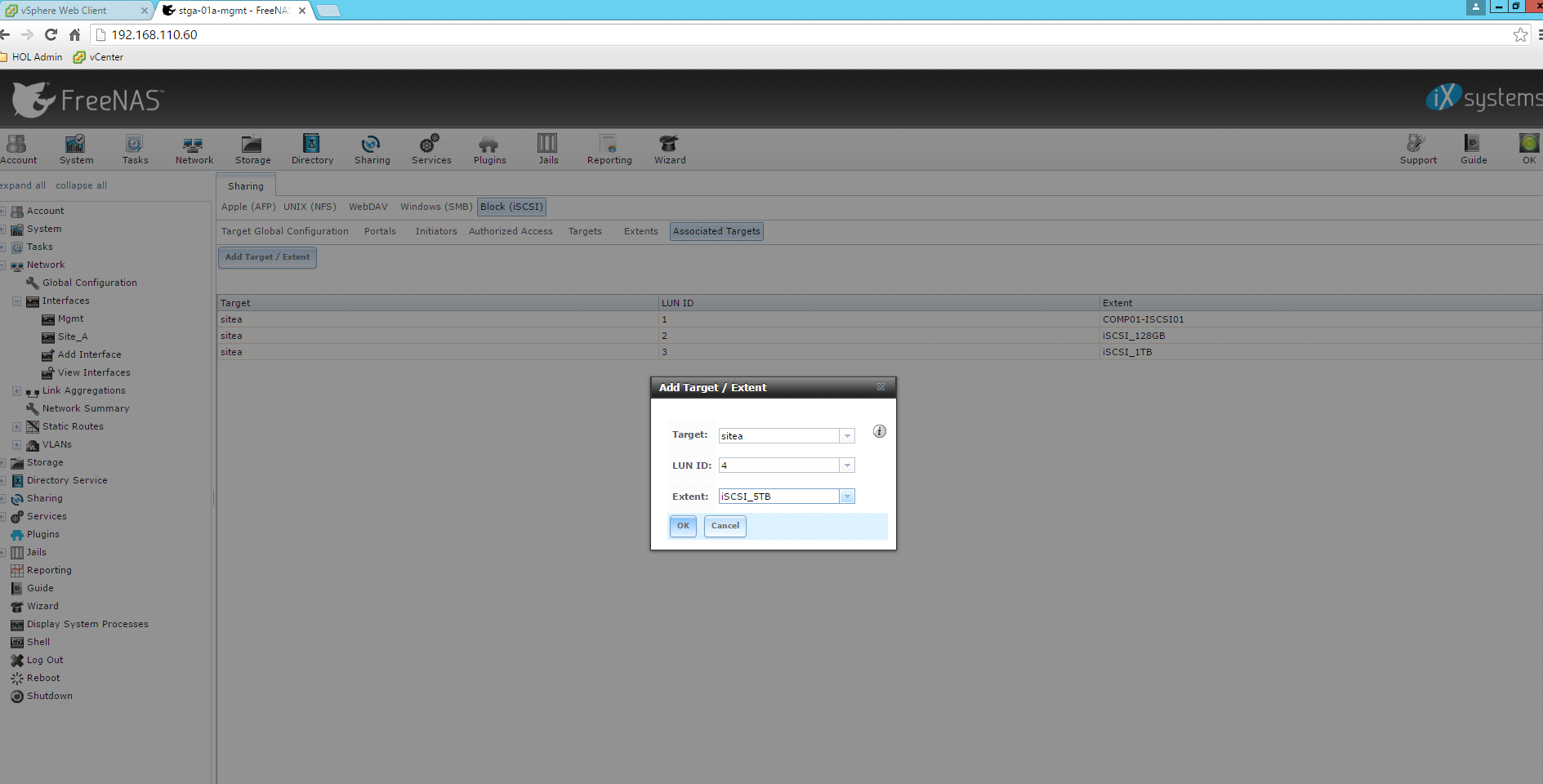
**NOTE**: If you select Browse, you may not see the filename iSCSI\_2TB. Specify the full pathname including the filename. Also, please specify the extent size appropriately. Extent size of auto does not work when the file does not exist. When creating the extent for the first time, the file is not present.



Also **enable** the checkbox for “Disable Physical Block Size Reporting” for each new Add-Extent.

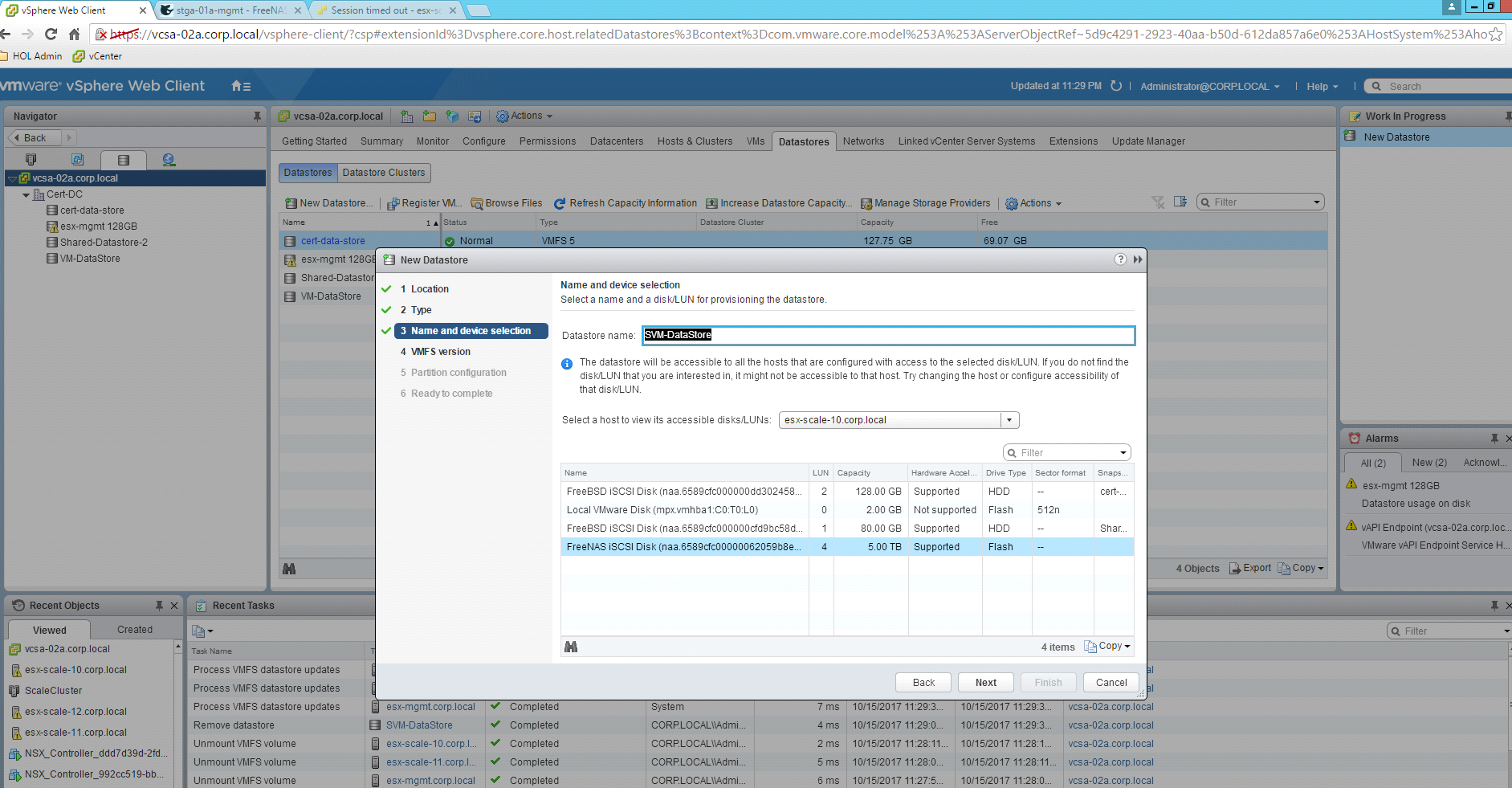


1. Go to Sharing->Block(iSCSI) -> Associated Targets and click on “Add Target/Extent”. Select the target as “sitea”, LUN ID as “1” (select unused LUN ID) and Extent as “iSCSI\_2TB”. Click on “OK”. This will create a new target to the extent created above.



Repeat the steps from 12.1 to 12.6 for the other FREENAS servers by logging into [http://192.168.110.62/](http://192.168.110.60/), [http://192.168.110.63/](http://192.168.110.60/) and [http://192.168.110.64/](http://192.168.110.60/)

Next step is to mount these as data stores in the vCenter. In the vCenter Web UI, go to Storage-> Datastores -> New Datastores. Select location as “ScaleCluster1” and Type as “VMFS”. Give the Datastore name as “SVM-DataStore-1”. In “select a host to view its accessible disks/LUNs:” select any of the esx-scale hosts between 1 and 64. After this, select 1280GB storage volume that was created and click on Next.



Do not change any of the configurations in the subsequent page you get after Next button. Finally click on “Finish”. This will create a “SVM-DataStore-1” of size 1280GB on ScaleCLuster1. Repeat the above steps for mounting the other datastores

"SVM-DataStore-2" on "ScaleCluster2" (select any of the scale hosts between 65 and 128).

"SVM-DataStore-3" on "ScaleCluster3" (select any of the scale hosts between 129 and192).

"SVM-DataStore-4" on "ScaleCluster4" (select any of the scale hosts between 193 and 256).

The above step will make these newly added data stores visible on all the scale hosts. If not, rescan the VMFS volumes in the hosts by executing the action "7: to do storage scan on the scale hosts" in scale\_test\_bed powercli script.

For Partner Management Console VM, use the static IP address as 172.17.11.12/23 and gateway as 172.11.10.1, set the domain name as PMC.corp.local.

Perform the necessary Scale testing with the configured test bed.

1. Select option "14" to do Host Preparation. It will ask for start and end Scale cluster index.
2. If you enter start as 1 and end as 4, it initiates Host Preparation and monitors the status in all the 4 scale clusters, one after the other.
3. Select option "15" to do Guest Introspection or Partner SVM deployment. After selecting this option, it asks for start and end Scale cluster index. It will also ask you to enter the service name of the Partner Service or the Guest Introspection to be deployed. You need to enter the service name within quotes. For Guest introspection, enter "Guest Introspection". The deployment will be done sequentially in all the scale clusters you have entered.

**NOTE**: The above action will also resolve any alarms raised during the deployment.

1. We have also added option "16" which will monitor and resolve alarms raised in the Service VM deployment. This option can be ignored as these operations are already taken care of, in option "15". However, in case, you have initiated the deployment manually and if you are not running option "15", you can select option "16" to monitor the Service VM deployment status.
2. Run ./ping\_linux\_vm.sh from Tiny-Linux-VM-1. This will test the basic connectivity to all the other Tiny-Linux-VMs newly added. If there are issues, please fix them and re-run the same.

Once all the SVMs are successfully deployed and their status look good, we will need to create a Security Group and ensure that all the Tiny-Linux-VMs are added as part of it. We will need to create a Security Policy using the partner Service Profile and bind it with the Security Group created. The Security Policy should be created to ALLOW traffic, by default.

Once the above configuration is done, we can initiate the actual test run.

Run ./ping\_linux\_vm.sh from Tiny-Linux-VM-1. This time we are validating the connectivity to all the other Tiny-Linux-VMs via the SVM, as per the policy applied.

Redirect the test results to a file. For ex., ./ping\_linux\_vm.sh > results. Here, ‘**results’** is the file where the test results are stored and back it up for report generation.

**Winding down the Horizontal scale test environment**

1. Once the testing is done, you can unmount the SVM datastores that we added earlier and delete them all from the vSphere client.
2. After the datastores are deleted from the vSphere web client, we will need to delete the volumes. Refer the section **Cleaning up the test bed (optional)** on **page 24,** for more details on the steps involved here (follow till the step “Go to Storage->Volume Manager, detach “SVM\_STORAGE\_VOL””).

*\*\* The other steps listed in the section* ***Cleaning up the test bed (optional)*** *are not really needed to be done. Those capture the steps for an ideal clean up. Here, we are not looking for an ideal clean up, as the vApp is anyway destroyed after the testing is done*.

1. Once the volumes are detached from the Storage VMs, go to pyCharm and run testif.py.
2. Select option “2. Remove ESX Scale” and press Enter. This will power off all the esx-horizontal-1 to esx-horizontal-32 and esx-horizontal-1-1 to esx-horizontal-32-1 hosts and remove them from the vAPP. This returns a Job ID. You can monitor the job by selecting option 6 and giving the Job ID that was generated.
3. This removes all the Scale ESXi resources from the existing vApp, so other vApps can use them.

**This completes the Horizontal Scale test.**

# Appendix:

## Steps to be followed, to increase the disk size on the Storage VMs

**Pre-requisites:**

The first step is to understand what is the size of one Partner SVM that is deployed. Once we get to know that, we will need to arrive at the disk space required for 200 SVMs of that size. As captured above, we will have 5.12 TB storage allotted for 200 SVM deployment as part of the resource allocation for Horizontal Scale test. Depending on the total disk space required, we need to decide on the extra disk space needed.

**For ex.,** Let’s say the SVM size is 30 GB. We generally use 200 ESXi hosts for testing and hence 200 SVMs will be deployed. The storage needed for 200 SVM deployment is around ~6 TB. As part of the test bed setup, we have 5.12 TB and hence the disk size needed extra is ~1 TB. So, an additional disk space of 250 GB per Storage VM (i.e., 250 GB \*4) is good.

The next step is to ensure that we have enough storage or space left in the org. To do so, we will need to check with the Engineer who has Admin access to the org **us01-5-nsx-certx-d** and know the overall storage available. **Currently, the contact person is Harish Bhat.**

**Once we get the confirmation that we have enough storage available at the org level, we will need to work on the storage extension. We can also power OFF some unwanted resources and save some storage, thus leading to better utilisation.**

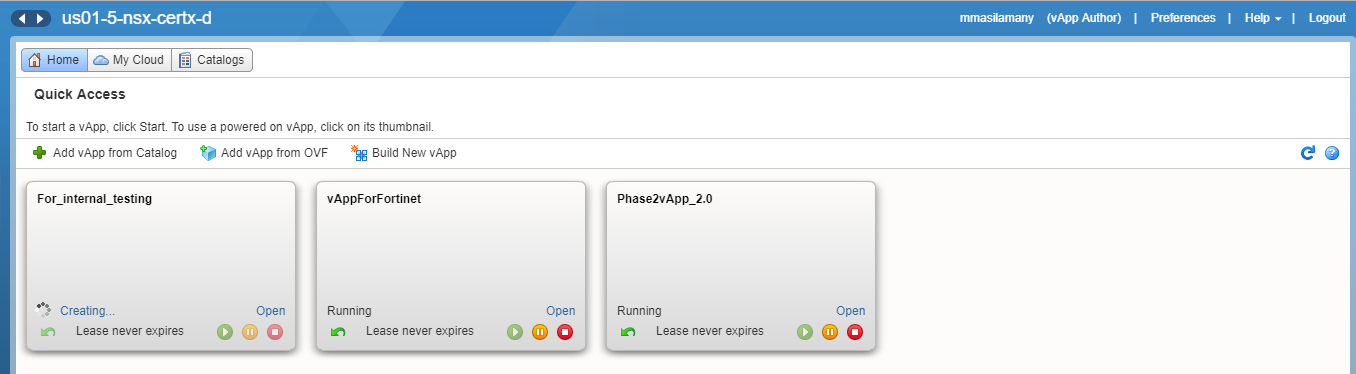
**Given below are the steps to save some storage used by unnecessary resources and using the same for extending the storage.**

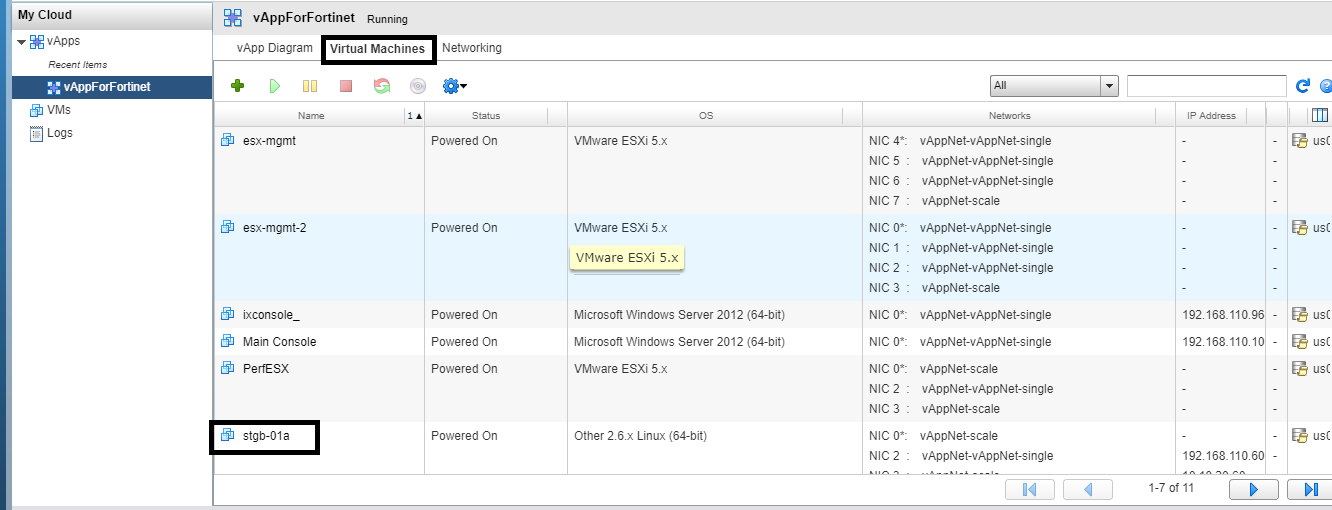
1. By default, when we setup the test bed for Horizontal test, there are 256 esx-scale hosts deployed out of 64 esx-horizontal hosts (4 nested esx-scale hosts per esx-horizontal host). We generally use 200 esx-scale hosts for our testing. The remaining 56 are powered ON and are not being used.
2. Given #1, we can save some storage by powering OFF the unused resources. Here’s how we did it for one of our partners whose SVM size was around 36 GB.
3. We powered OFF 10 out of the 64 **esx-horizontal** hosts at the vApp level. This means that 40 **nested esx-scale hosts** (4 esx-scale hosts present in each esx-horizontal host) are powered OFF, in turn.
4. After #3, we checked with Harish on the storage available at the org-level and found that there was around 7 TB available.
5. By default, we have 1.28 TB allocated for each of the Storage VMs. We have 4 Storage VMs and hence the total disk space available is 5.12 TB.
6. For the deployment of 200 partner SVMs of size 36 GB each, we needed 7.2 TB (36\*200). Let’s say 8 TB. So, we needed an additional storage of ~3 TB (i.e., 5.12 + 2.88 = 8).
7. We deleted the 1.28 TB disks (updation is not possible) on each of the Storage VMs and added a new disk of 2 TB (on each of the Storage VMs).
8. We mounted these volumes as datastores in the vSphere client and noted that the change in disk size was reflected.
9. This meant that we had 8 TB allocated for 200 SVMs deployment. Once done, we could deploy 200 SVMs successfully.

The section below covers the instructions on how to increase the disk size of these storage VMs after our analysis on the extra disk space required for 200 SVMs deployment.

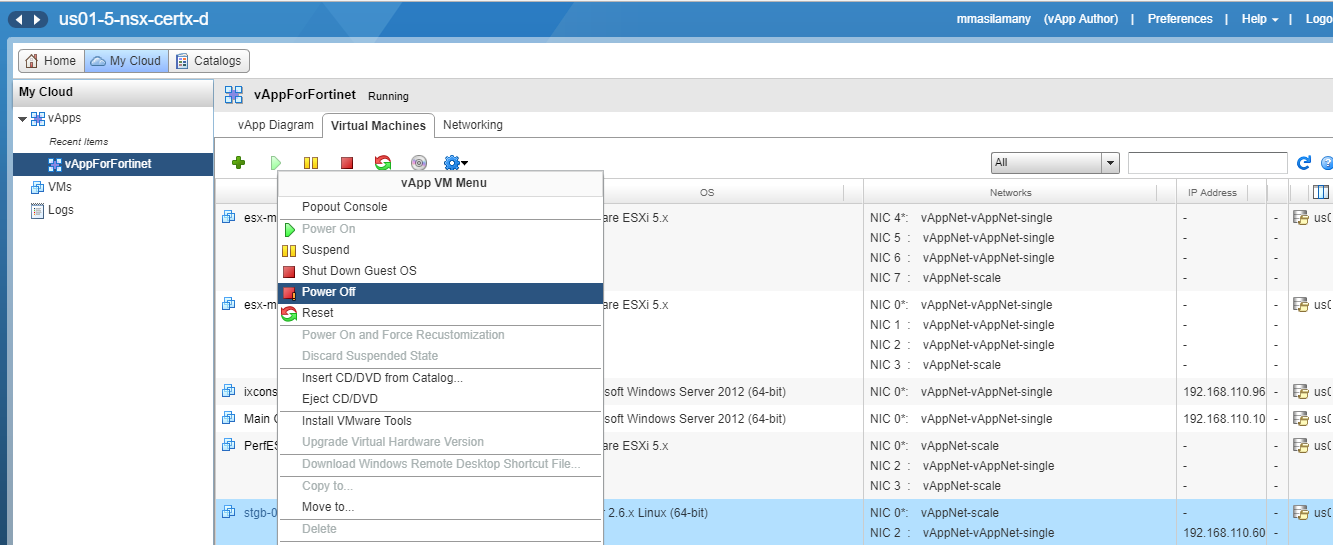
### Steps to add the new disk with size, as needed

1. Go to the URL <https://vcore5-us01.oc.vmware.com/cloud/org/us01-5-nsx-certx-d>.
2. On the Home page, click on the vApp where you want to get the disk space increased. Click on the **Virtual Machines** tab and that should list all the pre-built VMs available in the vApp.



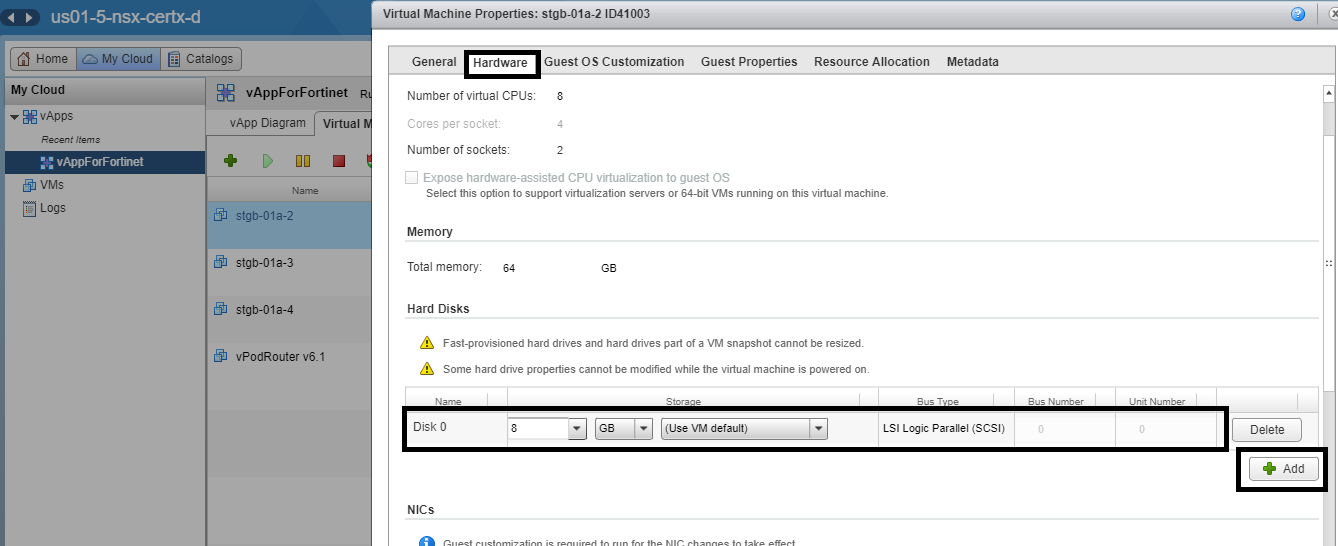


1. Look out for the VMs named **stgb-01a, stgb-02a, stgb-03a and stgb-04a**. These are the Storage VMs.
2. Power OFF these VMs, so we can edit the default configuration.

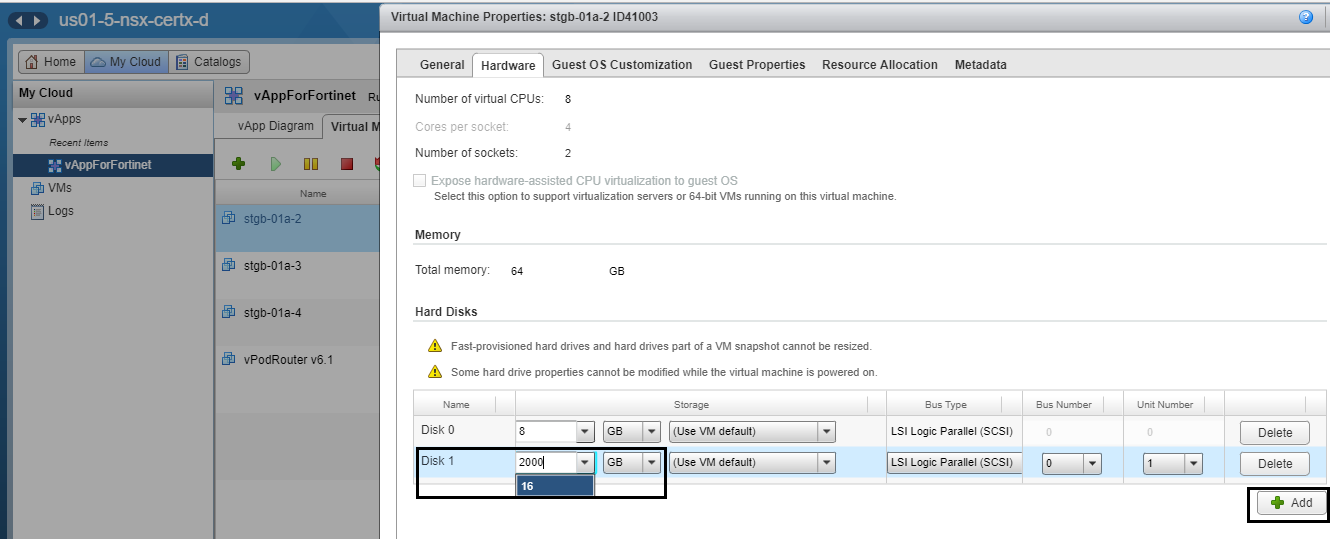


1. Once all these VMs are Powered OFF, right click on each of the VMs and click on **Properties**. Let’s say, we start with VM **stgb-01a**. Right click on it and choose Properties.
2. This opens the **Virtual Machine Properties** page. It shows all the configuration such as Hardware, Guest OS, Resource allocation etc.,
3. Click on the **Hardware** tab. Under **Hard Disks**, there will be a couple of disks shown.
4. By default, these VMs have one disk of size 8 GB.
5. When we provision the Horizontal test bed, there will be a second disk added with the default disk size set to 1.28 TB.
6. We need to delete the one that shows the disk size as 1280 GB. This is generally Disk 1. Click on the Delete button next to it and that deletes it.

DO NOT delete the other disk (Disk 0). NOTE that we are deleting the disk, as it’s not possible for us to edit the size on the existing disk.



1. Once the disk is deleted, add a new disk and provide the new size. Let’s say 2TB.
2. To add a new disk, click on the Add button at the bottom right of the Hard Disks section. Edit the size and click on OK. This adds the new disk, with the disk size as mentioned.



1. Repeat the steps 5-9 for the other Storage VMs as well. Once done, we need to power the VMs back ON.
2. To do so, select the VM, right click on it and choose **Power On**.
3. Wait for all the VMs to power ON completely and then verify if the disk added newly, is reflected. To verify, right click on the VM and click on **Properties**. Click on the Hardware tab and verify the disk size under **Hard Disks**. The new disk should show the disk size that was updated earlier (2GB in the above example).

We are all set to use these disks for Horizontal Scale test execution.

# Miscellaneous

The Horizontal hosts esx-horizontal-1 to esx-horizontal-32 have

* Management IP addresses starting from 192.168.110.21 to 192.168.110.52. All of them have mask 255.255.255.0
* VMKernel(vmk1) IP addresses accessing iSCSI Storage starting from 10.10.21.21 to 10.10.21.52. All of them have mask 255.255.254.0

The Horizontal hosts esx-horizontal-1-1 to esx-horizontal-32-1 have

* Management IP addresses starting from 192.168.110.101 to 192.168.110.132. All of them have mask 255.255.255.0
* VMKernel(vmk1) IP addresses for accessing iSCSI Storage starting from 10.10.21.61 to 10.10.21.92. All of them have mask 255.255.254.0

The Scale Host esx-scale-1 has

* Management IP address 172.17.11.1, mask 255.255.254.0
* VMKernel(vmk1) IP address for accessing iSCSI storage is 10.10.20.1

The Scale hosts from esx-scale-2 to esx-scale-254 have

* Management IP address starting from 172.17.10.2 to 172.17.10.254. All of them have mask 255.255.254.0
* VMKernel(vmk1) IP addresses for accessing iSCSI Storage starting from 10.10.20.2 to 10.10.20.253(except esx-scale-55, esx-scale-60, esx-scale-253 and esx-scale-254). All of them have mask 255.255.254.0
* esx-scale-55 has vmk1 IP address as 10.10.20.253, mask 255.255.254.0
* esx-scale-60 has vmk1 IP address as 10.10.20.254, mask 255.255.254.0
* esx-scale-253 has vmk1 IP address as 10.10.21.1, mask 255.255.254.0
* esx-scale-254 has vmk1 IP address as 10.10.21.2, mask 255.255.254.0

The Scale Host esx-scale-255 has

* Management IP address 172.17.11.2, mask 255.255.254.0
* VMKernel(vmk1) IP address for accessing iSCSI storage is 10.10.21.3, mask 255.255.254.0

The Scale Host esx-scale-256 has

* Management IP address 172.17.11.3, mask 255.255.254.0
* VMKernel(vmk1) IP address for accessing iSCSI storage is 10.10.21.4, mask 255.255.254.0

The esx-mgmt Host has

* Management IP address 192.168.110.55, mask 255.255.255.0
* VMKernel(vmk1) IP address for accessing iSCSI storage is 10.10.20.55, mask 255.255.254.0

The esx-mgmt-2 Host has

* Management IP address 192.168.110.56, mask 255.255.255.0
* VMKernel(vmk1) IP address for accessing iSCSI storage is 10.10.21.5, mask 255.255.254.0

There are 5 FREENAS Servers

* Stgb-01a VM has Management IP address as 192.168.110.60/24 and IP for accessing iSCSI is 10.10.20.60/23
* Stgb-01a-1 VM has Management IP address as 192.168.110.61/24 and IP for accessing iSCSI is 10.10.21.161/23
* Stgb-01a-2 VM has Management IP address as 192.168.110.62/24 and IP for accessing iSCSI is 10.10.21.162/23
* Stgb-01a-3 VM has Management IP address as 192.168.110.63/24 and IP for accessing iSCSI is 10.10.21.163/23
* Stgb-01a-4 VM has Management IP address as 192.168.110.64/24 and IP for accessing iSCSI is 10.10.21.164/23

MTU is set to 8000 in the servers as well as all of the VMK.

There are 256 Linux Guest VMs, IP addresses are

* Linux-VM1 has 172.16.11.1/23
* Linux-VM2 to Linux-VM254 has IP address ranging from 172.16.10.2 to 172.16.10.254
* Linux-VM255 has IP 172.16.11.2/23, Linux-VM256 has IP 172.16.11.3/23

There are 256 Tiny core Linux Guest VMs, IP addresses are

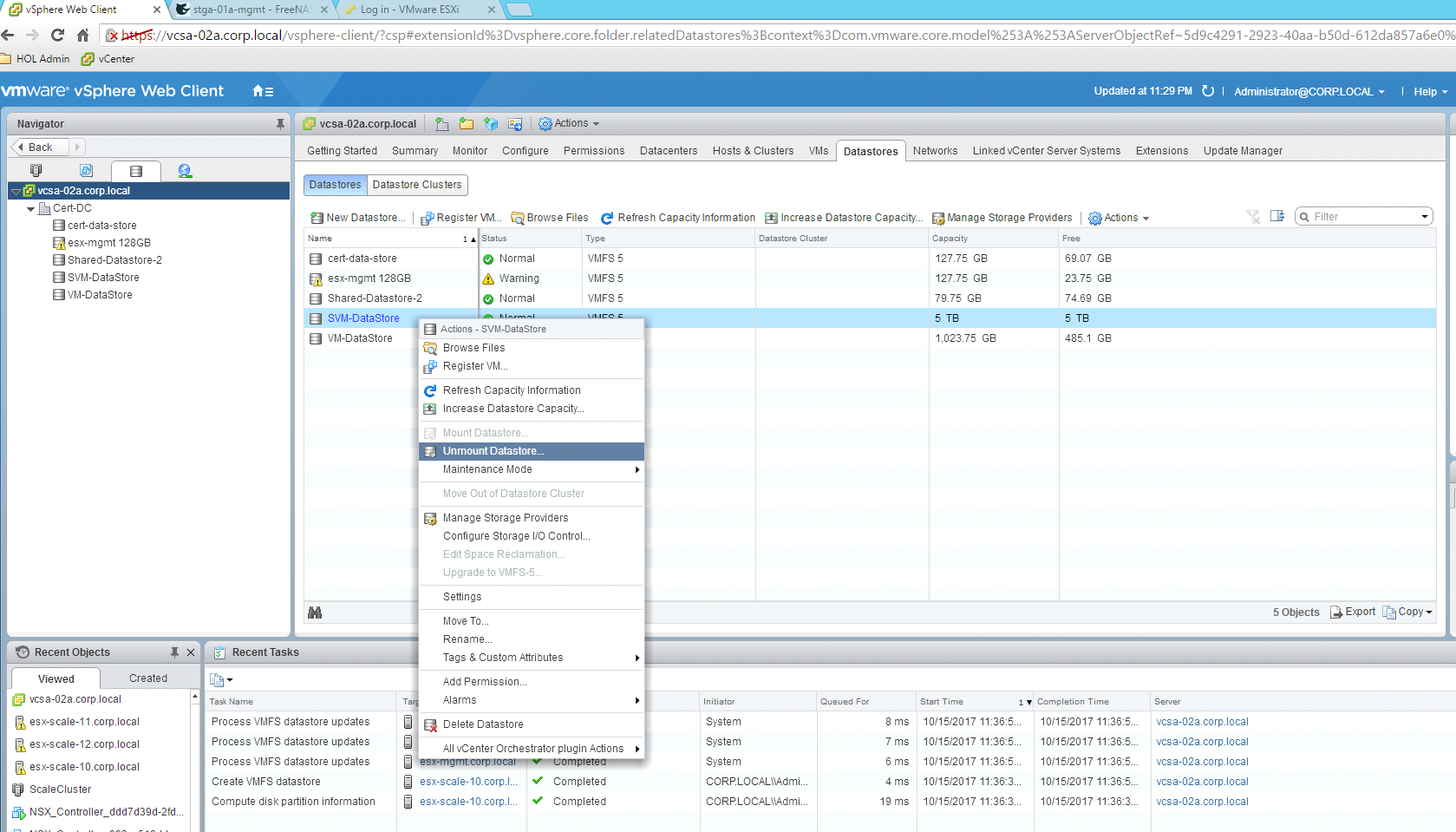
* Tiny-Linux-VM-1 to Tiny-Linux-VM-255 has IP address ranging from 172.18.10.1/16 to 172.18.10.255/16
* Tiny-Linux-VM-256 - IP 172.18.11.1/16

ESX-perf host is used for Vertical Scale testing.

* IP address is 192.168.110.134 statically configured. Storage address 10.10.21.165/24.
* MTU is 8000 on the VMK adapter. VSwitch is storage\_v\_switch on esx-perf and MTU is 8000 on the vSwitch.
* **NOTE**: if you are going to have nested ESX on esx-perf and need storage access at 8000 MTU, then the MTU of the vSwitch needs to be increased.

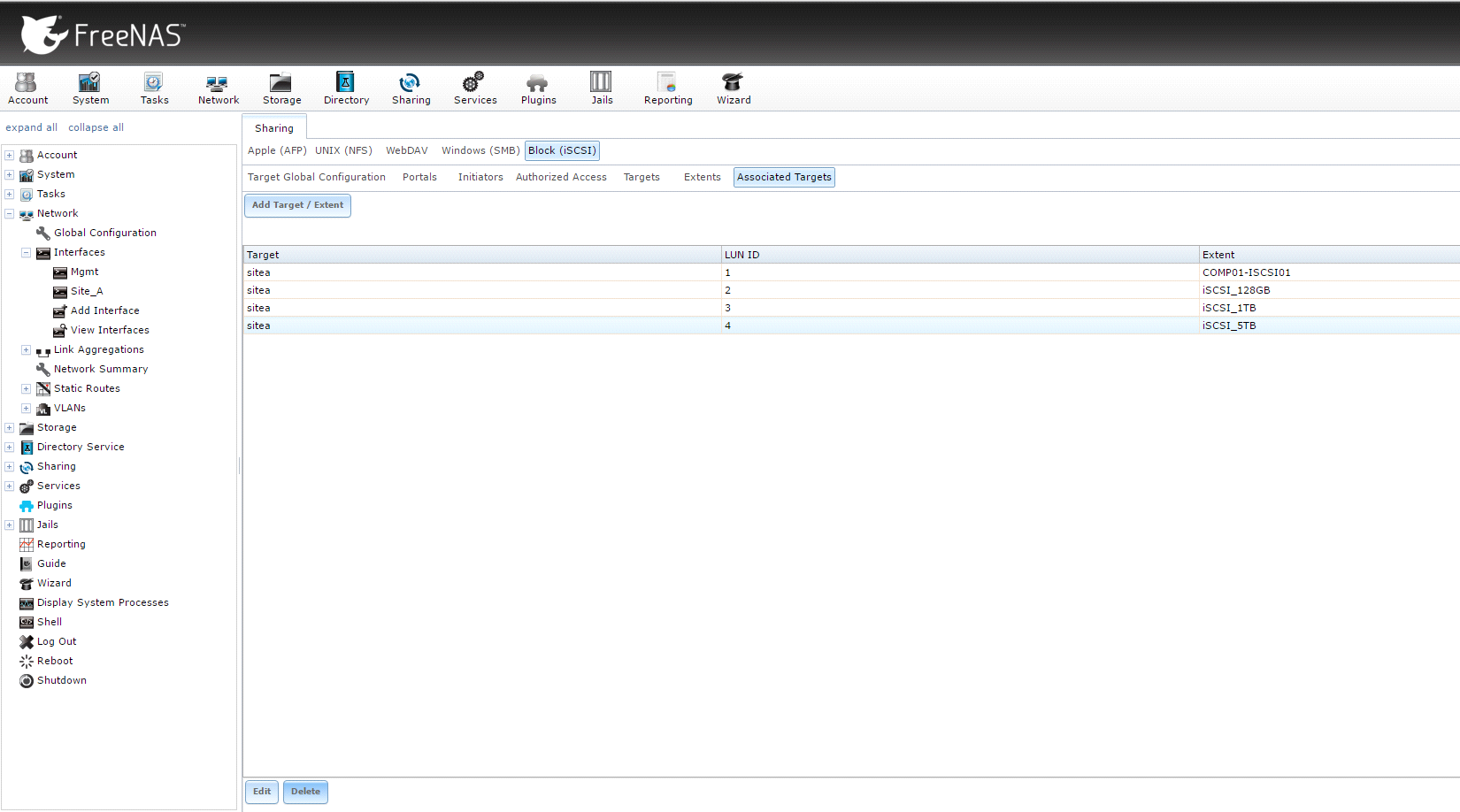
**Cleaning up the test bed (optional):**

* Unmount the “SVM-DataStore-1”, "“SVM-DataStore-2”, “SVM-DataStore-3” and “SVM-DataStore-4” in the vCenter on all the hosts.

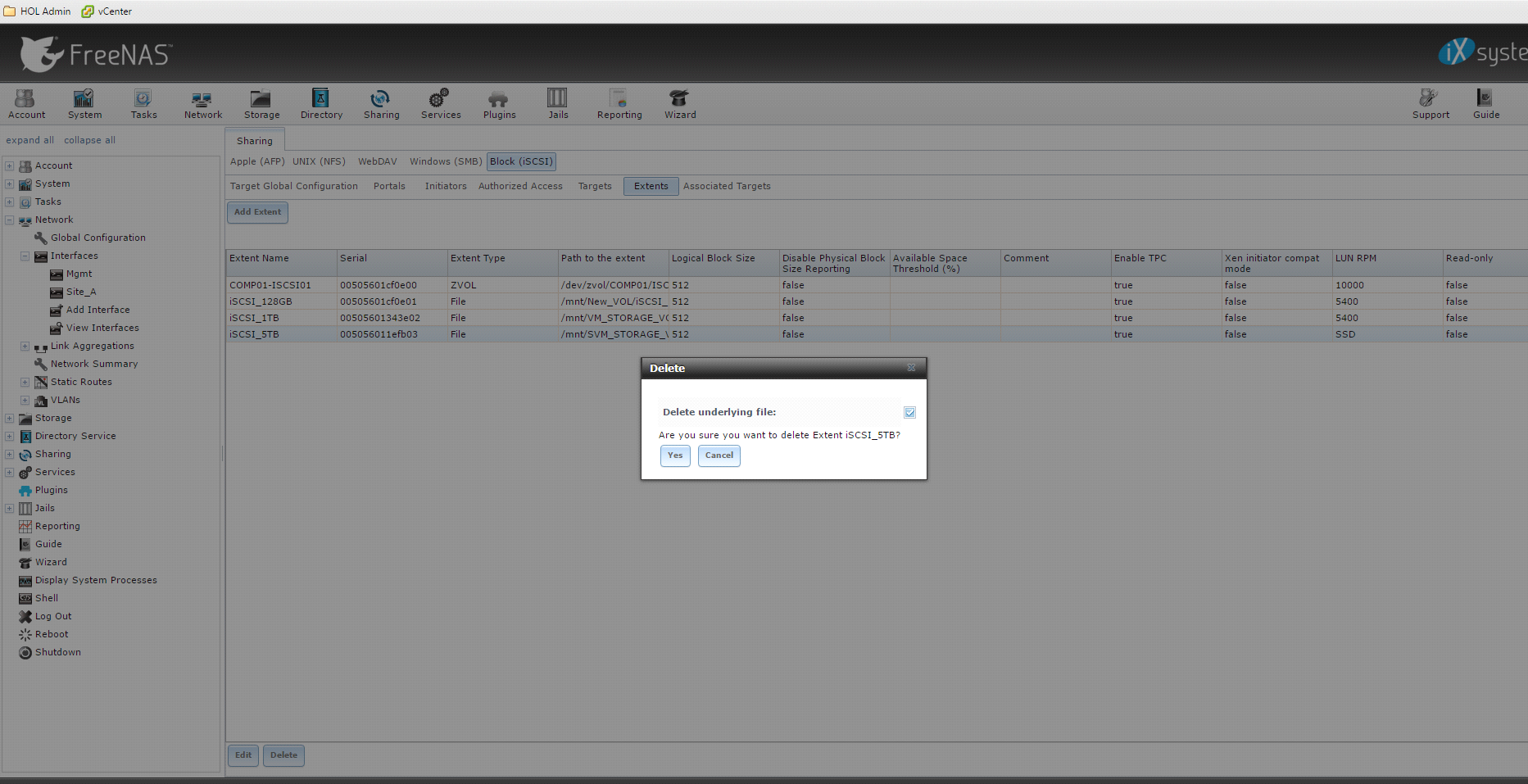


* After the datastores are unmounted, select “Delete Datastore” option. This will delete the datastores on all the hosts.
* We need to remove the storage volume in the FreeNAS server. Follow the steps below in all the 4 FREENAS Servers:

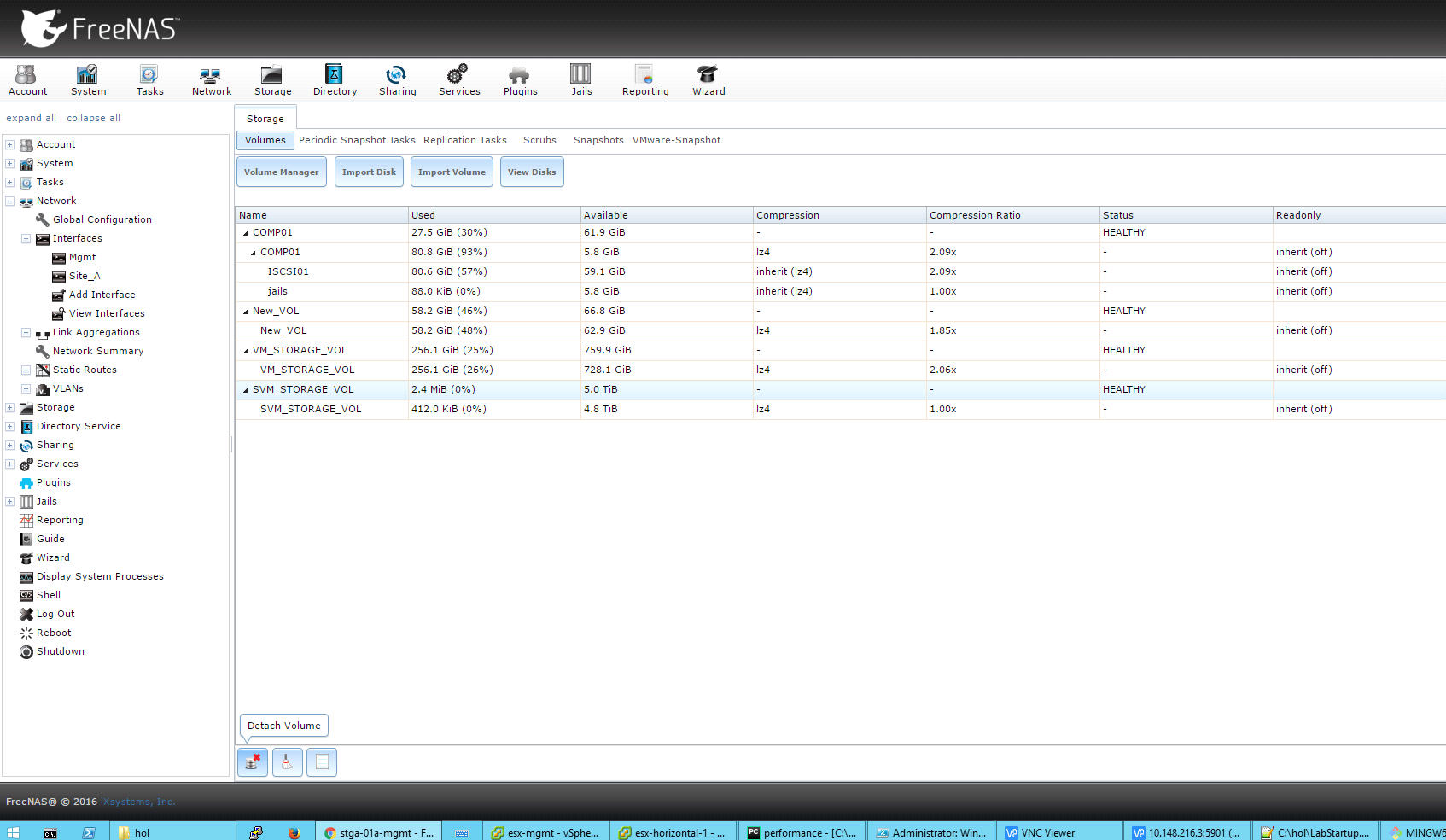
In the FreeNAS UI, go to Sharing->Block(iSCSI) -> Associated Targets. Delete iSCSI\_5TB target



* Go to Sharing->Block(iSCSI)->Extents, delete iSCSI\_5TB extent. Mark the checkbox “Delete underlying file”



* Go to Storage->Volume Manager, detach “SVM\_STORAGE\_VOL”



* Shutdown all the Guest VMs by executing "option 13: to shutdown Linux VMs "
* Detach the scale ESXi hosts from the vDS by executing the action "8: remove scale hosts from vDS" in scale\_test\_bed powercli script.
* Unregister all the ESXi hosts, VMs using the option "9: un-register VMs from the scale hosts" in scale\_test\_bed powercli script.
* Remove all the scale ESXi hosts from vCenter by executing the action "10: remove scale hosts from vCenter" in scale\_test\_bed powercli script.
* Shutdown all the scale ESXi hosts by executing the action "11: shutdown the scale hosts" in scale\_test\_bed powercli script. This step is not required as we are executing the Horizontal test at the last. The vApp will be deleted after this test.
* In the pyCharm which is running testif.py, select option “2. Remove ESX Scale” and press Enter. This will power off all the esx-horizontal-1 to esx-horizontal-32 and esx-horizontal-1-1 hosts and remove them from the vAPP. Again, you can monitor the job by selecting option 6 and giving Job ID.

# Troubleshooting

**vApp level issues:**

1. After creating the vApp, if the remote desktop connection to it doesn’t work, disable the firewall on the vApp.

To do so, go to vCloud Director, open the vApp, click on the 'Networking' tab and disable the checkbox for firewall.

2. If internet access doesn’t work in the Main Console VM, ensure that the default gateway is configured correctly. It is supposed to point to 192.168.110.10.

3. If the OVF URL generated via HFS server is not accessible from vCenter (i.e., if the SVM deployment fails), disable the Windows firewall on the Main Console VM.

**Horizontal Scale Test – Issues and their resolution:**

**Issue**: While running testif.py, for option ‘1. Request ESXi for Scale’. If the job fails in between (by adding only few ESXi Hosts to the vCloud App).

**Fix**: Please delete the unused resources/vApps if any. This could happen due to in-sufficient resources available to add all 64 ESXi hosts. Once done, run ‘Remove ESXi for Scale’ option before running ‘Request ESXi for Scale’. This is to ensure that we clean up the previously added resources.

**Callback Test -** **Issues and their resolution:**

If the creation of a Security Group/Policy fails (via test script), ensure that the previously created Security Groups/Policies are cleaned up and initiate another run.