# Callback or Control scale testing steps

In this document, we explain how to run callback or control scale tests. This document 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 two scripts.

1. Python scripts to setup and cleanup configuration
2. powerCLI scripts to get Guest VM IP-address.

This document is on top of the steps required for Vertical Scale test, as mentioned in the other document (instructions running performance testbed.doc).

Let’s begin:

1. 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.
2. cd to infrascripts-master. There are three directories. perfscripts, scalescripts & testscripts.
3. Follow the steps for Vertical Scale tests from document (instructions running performance testbed.doc) & run the options 5,7,8. **This will bring up 32 Guest VMs with 1 vNIC and ipv4-address configured.**
4. **If option # 6 was already done, the Guest VMs will have 8 vNICs added and so, we will need to manually remove the other 7 interfaces. If #6 was not done, we don’t have to do anything specific. We can continue with the next steps.** Validate the connectivity. run ./ping\_linux\_vm.sh from each Guest VM. This will test connectivity to all Linux-VMs newly added. If issues, please fix them.
5. Create only **one** Security Policy as required 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) |  |  |

1. 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

1. Open PyCharm  by double-clicking on the icon on desktop.
2. 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.***

1. 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

1. 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.

1. 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”.

1. Run python script CallbackScaleTests.py

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

***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

1. Tear down the setup. Follow the steps for Vertical Scale tests from document (instructions running performance testbed.doc) & run the Options 9,10,11. This will power off 32 Guest VMs and remove them from the vCenter inventory.

Congratulations, you have completed Callback Scale test of NSX scale test plan.