

Lab Setup Horizon  
Automated using Selenium

Version 1.2

Document created by:

Joseph Barber

Document created date:

04/12/2015

Contents

[Setup 3](#_Toc437517766)

[Common Failures 5](#_Toc437517767)

[Debug 5](#_Toc437517768)

[Re-run Lab Setup After Failure 5](#_Toc437517769)

[Final Notes – Jira Issue 8](#_Toc437517770)

[Final Notes – TiS build 9](#_Toc437517771)

[Final Notes – Lab\_Setup.sh/Cleanup.sh Version 10](#_Toc437517772)

# Setup

Starting with a FRESH VM:  
  
VirtualBox version: 4.3.20

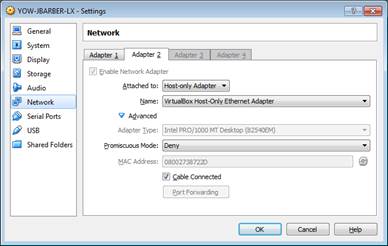
Python version: 2.7.6  
Located: /usr/bin/python

Firefox version: 42.0

VM OS: Linux VM OS in VirtualBox

Modifications to the Linux VM in VirtualBox so that it can access Horizon (https://10.10.10.2):

1. Add Virtual Network Adapter to Linux VM in VirtualBox:



1. Commands to Run in Linux VM:

sudo ip addr (verify eth1 exists)

sudo ip addr add 10.10.10.6/24 dev eth1 (add IP to eth1)

sudo ip link set eth1 up (set eth1 to state up)

ifconfig (verify that eth1 is up and has IP 10.10.10.6)

ping 10.10.10.2 (ensure connection to horizon is working)

1. Once you have that setup properly you can now install the prerequisites:

sudo apt-get install python-pip

sudo pip install selenium

1. Note: If you want to run this in headless mode (I.E no browser showing). You must get:

sudo pip install pyvirtualdisplay

With pyvirtualdisplay now installed you can continue following this guide.

Install WASSP REPO using WASSP Users' Guide

1. Follow this step if you will be running lab setup in Horizon using Selenium on your Linux VM:  
   **See note on page 8 about workaround for Jira issue.**

cd /home/<user>/wassp-repos/testcases/cgcs/selenium\_tests/Horizon\_Automation

python lab\_setup\_horizon.py

If you have installed pyvirtualdisplay to run the lab\_setup\_horizon.py headless mode:

cd /home/<user>/wassp-repos/testcases/cgcs/selenium\_tests/Horizon\_Automation

python lab\_setup\_horizon.py --headless

1. If you are automating this for the purposes of sanity through WASSP follow this step:

When automating through WASSP a pre\_test.frag and a post\_test.frag are required to start and kill a vnc server. Please refer to the Horizon\_Automation folder for an example. A VNC server session is only required if you plan to run the test on a machine without a display.

You can run the test through WASSP without the pre\_test.frag and post\_test.frag using the test\_case.xml if you want to locally run it on your machine through WASSP. The test\_case.xml is where lab\_setup\_horizon.py is called which is the main python file to run the automated Horizon lab setup.

# Common Failures

There are a few common reasons for the script to fail.  
In lab\_setup\_horizon.py every method call is very dependent on the previous method call.

1. Can’t find next HTML element
2. Can’t get URL

# Debug

It is fairly simple to debug the common failures listed above.

If Selenium WebDriver cannot find the next HTML element that means that the previous function that has been performed has failed. **Remember**: In lab\_setup\_horizon.py every method call is very dependent on the previous method call. Sometimes adding extra time for the page to load will fix this issue.

This means that when tracing a trace back error, don’t just look at the line that failed but look at the step that was executed before it.

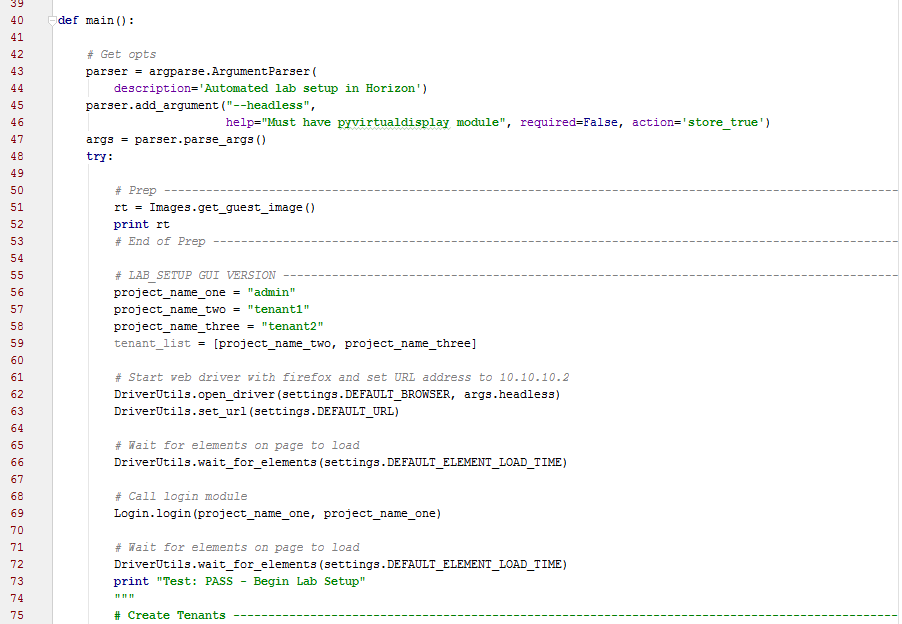
* time.sleep()

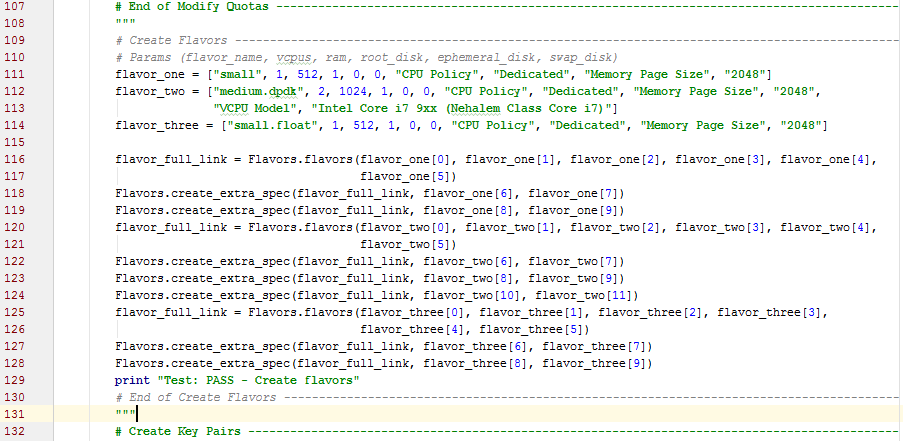
# Re-run Lab Setup after Failure

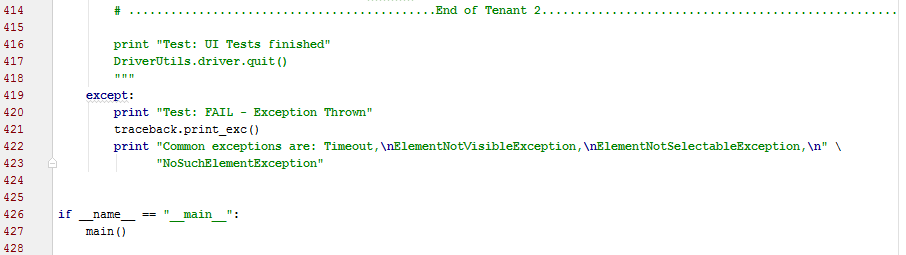
If lab\_setup\_horizon.py has failed during its run and you fix the issue. You can manually cleanup Horizon using the lab\_cleanup.sh script. Or you can run the remainder of the script where the crash happened. You can comment out the section that has already run. Just make sure that you don’t comment out the login section at the beginning!

Use 3 double quotes to start a multiline comment and another 3 double quotes to close it.

For example if you wanted just to login and run create flavors you would do this:



**<Commented out section>**



Don’t forget to comment out the bottom properly not commenting out the entire file.

# Tested Execution Setup - Jira Issue

There is currently a Jira opened that affects this script:  
The image cannot be created from HTTP URL.

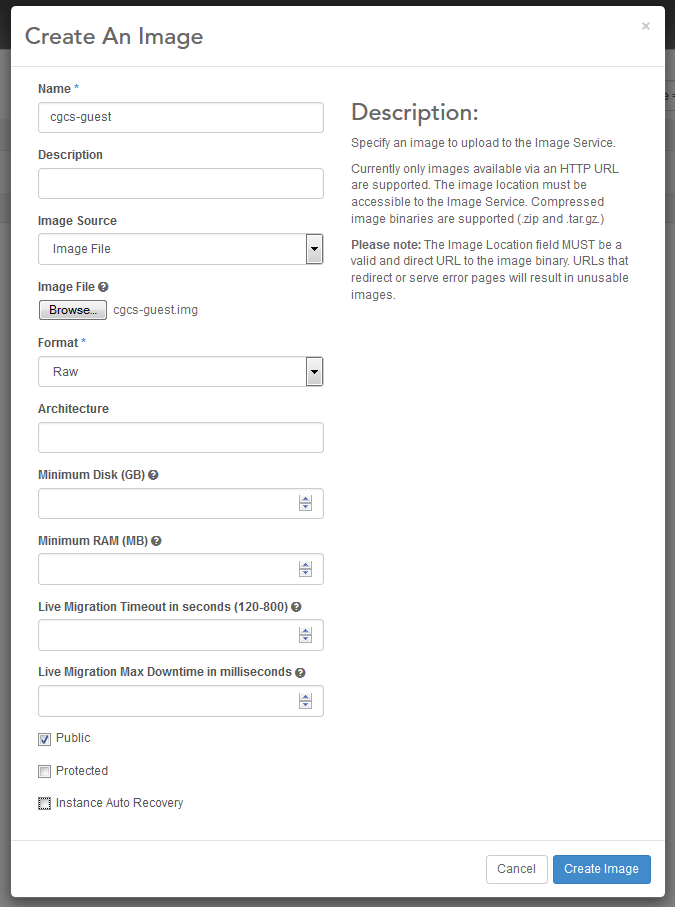
<https://jira.wrs.com:8443/browse/CGTS-3158>

Workaround:

Manually create the image inside Horizon before running lab\_setup\_horizon.py

* Download the cgcs-guest.img locally to your PC from a cgts server
* When creating the image change the ‘Image Source’ from ‘Image Location’ to ‘Image File’ and browse for the cgcs-guest.img saved previously
* Follow the screenshot below

Once this Jira has been fixed, see lab\_setup\_horizon.py and search for CGTS-3158



# Tested Execution Setup – TiS build

Last build used:  
###  
### Wind River Titanium Server  
### Release 15.10  
###  
### Wind River Systems, Inc.  
###  
  
SW\_VERSION="15.10"  
BUILD\_TARGET="Host Installer"  
BUILD\_TYPE="Formal"  
BUILD\_ID="2015-12-06\_22-05-42"  
SRC\_BUILD\_ID="61"  
  
JOB="CGCS\_2.0\_Unified\_Daily\_Build"  
BUILD\_BY="jenkins"  
BUILD\_NUMBER="266"  
BUILD\_HOST="yow-cgts3-lx"  
BUILD\_DATE="2015-12-06 22:05:57 -0500"

# Tested Execution Setup – Lab\_Setup.sh/Cleanup.sh Version

The latest lab\_setup.sh and lab\_cleanup.sh that were used are inside the folder:

/home/<user>/wassp-repos/testcases/cgcs/selenium\_tests/Horizon\_Automation

**Note:** If lab\_setup.sh changes there must be changes made to lab\_setup\_horizon.py to replicate what was changed.

Methods of understanding what has been changed:

1. Check lab\_cleanup.group0.log
   1. Read CLI commands. Each CLI command should correspond to a method call in lab\_setup\_horizon.py
2. Check Horizon after the lab\_setup.sh has run
   1. Screenshot or remember what is different after running lab\_cleanup.sh and then lab\_setup\_horizon.py

Lab\_Cleanup.sh commit: 5116ebaa667a01bf0c8ca783fb5d22a63662343c  
  
Tested Execution Setup – Run on Non-VBox Lab

Lab\_setup\_horizon.py has been run on TiS on Tis (Cumulus cloud) setup successfully.

If you wanted to run lab\_setup\_horizon.py on a non-VBox lab, you need to go into the settings.py file and edit the DEFAULT\_URL constant to the IP of the specific lab.