

Learning Lab: Cisco Open Software-Defined-Network Controller (COSC)

Preface

This document resides in the git project:

https://devhub.cisco.com/sf/sfmain/do/viewProject/projects.cosc_learning_labs

<https://gerrit-open1.cisco.com/gerrit/#/admin/projects/cosc-learning-labs>

Access to the git project is described below. The path to this document is:

<git>/cosc-learning-labs/doc/guide/virl.docx

<http://gerrit-open1.cisco.com/gerrit/gitweb?p=cosc-learning-labs.git;a=blob;f=doc/guide/virl.docx>

On Ubuntu, you need LibreOffice to read this document. To install:

```
sudo apt-get install libreoffice
```

A copy of this document is available in the following additional formats:

- <git>/cosc-learning-labs/doc/guide/virl.pdf
- <http://gerrit-open1.cisco.com/gerrit/gitweb?p=cosc-learning-labs.git;a=blob;f=doc/guide/virl.pdf>

Table of Contents

Learning Lab: Cisco Open Software-Defined-Network Controller (COSC)	1
Preface	1
Table of Contents	1
Overview	2
Fast Track	2
Instructions	2
1. Connect to a Virl VM	2
2. VM Maestro	2
3. Create and Run a Topology in VM Maestro	3
4. Cisco Open SDN Controller (COSC)	7
5. Client Applications and Sample Code	9
6. iPython/Jupyter Notebook Server	12
7. Eclipse IDE	15
Appendices	17
Alternative Image of XR Device	17
Virl VM on Rack Space	19
Download OVPN certificate file from Virl VM Server	19
About the image of the Netconf enabled XR device	21
Custom Topology	24

Overview

This document contains instructions for running a network simulation, consisting of:

- Network Devices – Netconf enabled, two or more.
- COSC Server - Ubuntu.
- COSC Client – Ubuntu.

Each component in the list above is configured by the instructions below. The Virl VM server is also configured.

Fast Track

- Connect to Virl VM
- Install image of Netconf enabled XR device
- Run simulation of the topology
- Configure each network device (router)
- Configure COSC Server
- Configure COSC Client

Instructions

1. Connect to a Virl VM

Obtain a network connection to a Virl VM. The provisioning of the Virl VM is outside the scope of this document. Some of the appendices are related to this topic.

For the remainder of this document it is assumed that the IP address of the Virl VM is 172.16.1.1. This is for illustration purposes only. Please adjust all the following instructions accordingly.

2. VM Maestro






Run the application VM Maestro on your computer. Obtain VM Maestro from Virl VM. Assuming that the IP address of the Virl VM is 172.16.1.1, the URL is:

<http://172.16.1.1/download>

You should see a package for your platform, as per the screen-shot below:



Index of /download

	Name	Last modified	Size	Description
	Parent Directory	-		
	VMMaestro-dev-1.2.0-91-linux.gtk.x86_64.zip	2015-02-05 11:21	83M	
	VMMaestro-dev-1.2.0-91-setup_32.exe	2015-02-05 11:21	82M	
	VMMaestro-dev-1.2.0-91-setup_64.exe	2015-02-05 11:21	82M	
	VMMaestro-dev-1.2.0-91.dmg	2015-02-05 11:22	99M	

Apache/2.4.7 (Ubuntu) Server at 172.16.1.1 Port 80

When VM Maestro is first started, you will be prompted for credentials. The username is 'guest' and the password is 'guest'.

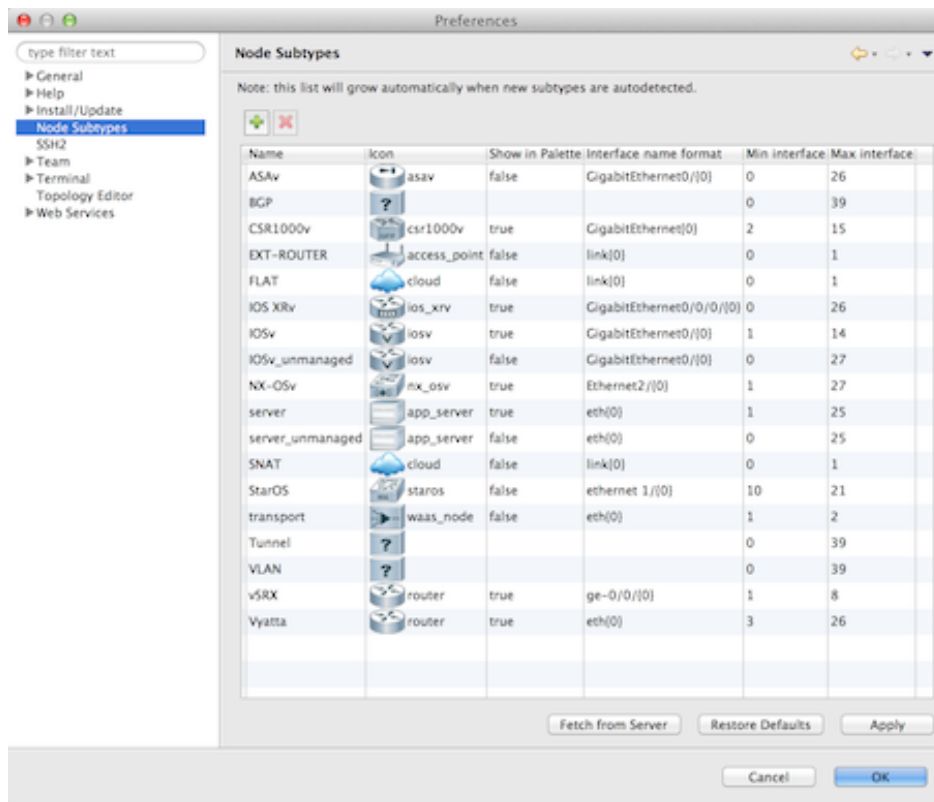
If VM Maestro stops responding, or otherwise goes wrong, delete directory `vmmaestro` from your home directory. *All topologies are stored there so preserve what you need prior to deletion.*

3. Create and Run a Topology in VM Maestro

In VMMAestro:

You may need to expand the list of Node Subtypes. To do so, use the VMMAestro user interface: File -> Preferences -> Node Subtypes -> Fetch from Server -> <confirm> -> Ok

The screenshot, below, shows the Node Subtypes:



Nathan John Sowatskey 27/2/2015 17:34

Comment [1]: We really just want to open up the Git view, import the project and use the existing topology here. We can put these directions for how to create a topology in an appendix.

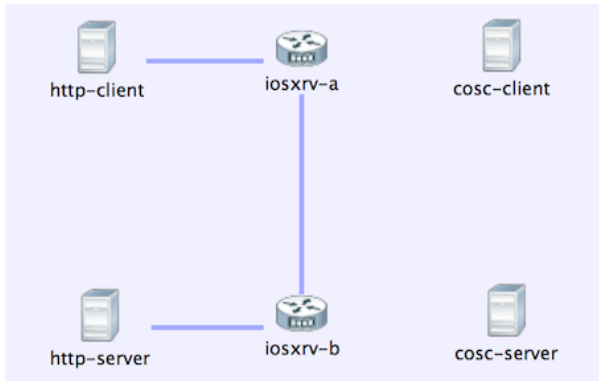
Create your own topology or import an existing topology file. There are topologies saved in the project `cosc-learning-labs`. You may need to download that project - see below for git clone instructions. The topology files are in directory 'topology' (top level of that project). The topology used in the remainder of this document is:

```
<git>/cosc-learning-labs/topology/standard.virl
```

Nathan John Sowatskey 3/3/2015 10:29

Comment [2]: These instructions need appear first.

The screen-shot, below, is the topology named above:



To start the simulation:

1. Configuration Menu -> Build Initial Configurations

Note: you may answer "No" to the pop-up question: Would you like to see the router configuration changes made by AutoNetkit?

Note: you may answer "No" to the pop-up question: Would you like to open the AutoNetkit Visualization?

2. Simulation menu -> Launch Simulation -> Simulation Perspective
3. Wait for 10 minutes, at least. Do not attempt to connect to a virtual device prematurely or you may compromise the initialisation process.
4. Connect to the console of each XRv device. Authenticate with credentials cisco/cisco. Note that older versions of Virl VM (prior to version 204) used the credentials test/test.

You can open the console from the command line using telnet. Or, in VM Maestro, use the context menu (right click) on a device in the Simulations view (Simulations perspective).

5. On each virtual device, apply the commands below and accept the suggested default values:

```
netconf-yang agent ssh
crypto key generate dsa
```

Note: if you are familiar with IOS devices you may be accustomed to entering commands such as 'enable' or 'conf t' (configure true). These commands are not necessary at this time.

Sample of console (pressing Enter to accept default value of 1024):

```
Tue Feb 10 14:48:22.374 UTC
The name for the keys will be: the_default
Choose the size of your DSA key modulus. Modulus size can be
512, 768, or 1024
bits. Choosing a key modulus
```

```
How many bits in the modulus [1024]:
Generating DSA keys ...
Done w/ crypto generate keypair
[OK]
```

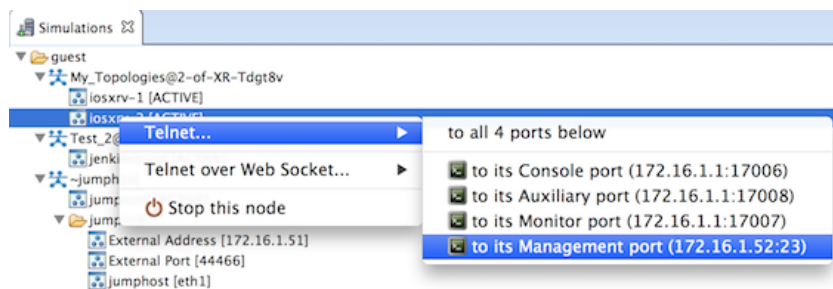
- Get the management IP address with this command:

```
sh ip int br
```

Sample output, highlighting the information sought, 172.16.1.52:

Interface	IP-Address	Status
Protocol		
Loopback0	192.168.0.2	Up
Up		
MgmtEth0/0/CPU0/0	172.16.1.52	Up
Up		
GigabitEthernet0/0/0/0	unassigned	Up
Up		
GigabitEthernet0/0/0/1	unassigned	Up

The IP address of the management interface is also displayed on the context menu of VM Maestro, as highlighted in the screen-show below:



- Test device from command shell (e.g. bash), using the management IP address obtained in the step above:

```
ssh -p 830 cisco@172.16.1.52 -s netconf
```

Note: Netconf runs over ssh and the standard port is 830.

The expected response, after confirming the server key and supplying the password "cisco", is an XML document starting with:

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

If you see "no hostkey alg" then you skipped the command "crypto key generate dsa" command, as described above.

After following the instructions 1 to 7, above, each virtual network device:

- is running in the network simulation
- has Netconf enabled
- has cryptography enabled
- has a management interface IP address
- can be accessed via ssh

4. Cisco Open SDN Controller (COSC)

The Controller is a service that runs on a computer that can *reach* the virtual network devices. In these instructions, the Controller is installed on the virtual server named "cosc-server" in the topology diagram, [above](#).

Login to the cosc-server, using the IP address assigned when the simulation was run. In the example below this is 172.16.1.60:

```
ssh -o StrictHostKeyChecking=no cisco@172.16.1.60
```

Sample successful response:

```
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-45-generic x86_64)

* Documentation:  https://help.ubuntu.com/
Last login: Tue Feb 10 10:51:16 2015 from 172.16.1.21
```

Copy the COSC distribution file to the cosc-server using its URL:

```
mkdir ~/cosc
cd ~/cosc
wget <url>
```

Note for Cisco [internal use](#): look in directory:

<http://codehub-one-fw-ci:8085/auth/view/cdl-eft-throttle/job/cdl-eft-autorelease/lastSuccessfulBuild/artifact/integration/distributions/cdl/target>

Confirm the integrity of the [file](#):

```
md5sum distribution-karaf-1.0.0-00003.tar.gz
```

Correct response:

```
28b6573955bcff93403c33e8d6a641f9  distribution-karaf-1.0.0-00003.tar.gz
```

Unpack the distribution:

```
tar xf distribution-karaf-1.0.0-00003.tar.gz
ls -l
```

Directory listing:

```
total 154596
drwxrwxr-x 9 vir1 vir1      4096 Feb 10 11:39 distribution-karaf-1.0.0-00003
-rw-r--r-- 1 vir1 vir1 158298322 Feb  6 17:59 distribution-karaf-1.0.0-00003.tar.gz
```

Install [Java](#):

```
sudo apt-get update
sudo apt-get install openjdk-7-jdk
```

Alternative:

<http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html>

Nathan John Sowatskey 3/3/2015 10:41

Comment [3]: It is also possible to run the controller on your own laptop, for example, which is explained in an appendix below.

Nathan John Sowatskey 3/3/2015 10:44

Comment [4]: This is required as the first attempt to use this command will create an entry in your ~/.ssh/known_hosts file with the fingerprint of the server that was instantiated the first time you do this. Subsequent attempts with the same topology, under the same conditions, will be to a server with the same IP address but, as it will probably be a different server instance, the fingerprint will be different, and so ssh will fail.

Nathan John Sowatskey 3/3/2015 10:44

Comment [5]: Something missing here?

Nathan John Sowatskey 3/3/2015 10:45

Comment [6]: We need to start using the OVA version of COSC.

Nathan John Sowatskey 3/3/2015 10:47

Comment [7]: We will need to refer to a download page with the MD5 hash, and move away from using this specific file. Check with Meenakshi.

Nathan John Sowatskey 3/3/2015 10:48

Comment [8]: We should be using a pre-prepared VM with all tools installed.

See what version of Java was installed:

```
java -version
```

Need at least 1.7.0_55:

```
java version "1.7.0_75"  
OpenJDK Runtime Environment (IcedTea 2.5.4) (7u75-2.5.4-1~trustyl)  
OpenJDK 64-Bit Server VM (build 24.75-b04, mixed mode)
```

Run the controller service:

```
cd distribution-karaf-1.0.0-00003/bin  
source setenv  
nohup ./start
```

Check the status:

```
./status
```

Successful output:

```
Running ...
```

The COSC username and password are both 'admin'.
To verify the controller from a web browser:

<http://172.16.1.1:8181/apidoc/explorer/index.html>

... and expect a HTML page.

Verify advanced features of the controller with URL:

<http://172.16.1.1:8181/restconf/config/opendaylight-inventory:nodes/node/controller-config/yang-ext:mount/config:modules>

Expect status code 200 and response content in XML format.

To test the controller from the command line:

```
wget --user=admin --password=admin -O - --progress=dot  
http://172.16.1.1:8181/restconf/config/opendaylight-inventory:nodes
```

While starting:

```
Connecting to 172.16.1.1:8181... connected.  
HTTP request sent, awaiting response... 503 Service Unavailable
```

When ready:

```
Connecting to 172.16.1.1:8181... connected.  
HTTP request sent, awaiting response... 200 OK
```

Or:

```
curl -u=admin:admin http://  
172.16.1.1:8181/restconf/config/opendaylight-inventory:nodes
```


If, during normal usage, the controller becomes unresponsive then re-install, as above, from the tar `ball`.

5. Client Applications and Sample Code

The client applications and sample code run on a computer that connects to the COSC service. Suitable computers are your own laptop or the `cosc-server`. The following instructions apply to whichever computer you choose as the *COSC Client*.

a. Python

To see the version of Python:

```
python --version
```

Upgrade to version 2.7.x if lower. For example, on Ubuntu:

```
sudo apt-get update
sudo apt-get upgrade python
```

Upgrade python-dev too. For example, on Ubuntu:

```
sudo apt-get install python-dev
```

Test if `easy_install` is available. For example, on Ubuntu:

```
which easy_install
```

Desired output:

```
/usr/bin/easy_install
```

If there is no output then `easy_install` must be installed:

```
sudo apt-get -y install libxslt-dev
sudo apt-get -y install zlib1g-dev
sudo apt-get -y install python-setuptools
```

Test if `pip` is available. For example, on Ubuntu:

```
which pip
```

Desired output:

```
/usr/local/bin/pip
```

If there is no output then `pip` must be installed:

```
sudo easy_install pip
```

b. Virtual Environment

The use of a virtual environment is `optional`. If an entire computer is dedicated to running the client application then a virtual environment is not necessary. If a virtual computer is used then it is already a virtual environment.

If multiple Python applications run on the same computer then use a separate virtual

Nathan John Sowatskey 3/3/2015 10:49

Comment [9]: Remembering to delete the previous install contents entirely.

Nathan John Sowatskey 3/3/2015 10:50

Comment [10]: We should be using the `cosc-client` which has the code already cloned and Eclipse JEE with PyDev pre-installed.

Nathan John Sowatskey 3/3/2015 10:50

Comment [11]: All of this should be pre-packaged in the `cosc-client dev` VM. We can move this to an appendix.

Nathan John Sowatskey 3/3/2015 10:51

Comment [12]: `Pip` is generally preferred to `easy_install` as I understand it?

Nathan John Sowatskey 3/3/2015 10:53

Comment [13]: The section above needs to be re-written with `pip` and also tested on OSX.

Nathan John Sowatskey 3/3/2015 10:53

Comment [14]: Move this to an appendix.

Nathan John Sowatskey 3/3/2015 10:54

Comment [15]: Relevant to running code in one's own environment.

environment for each. There are multiple tools that provide a virtual environment. The example below uses virtualenv. See also: venv, pyenv, pythonz.

On Ubuntu, OSX, Linux:

```
pip install virtualenv
cd ~/git/cosc-learning-labs
virtualenv -p python2.7 env
source env/bin/activate
```

The final command, above, enters a *shell* or *mode*. Eventually, when you want to exit:

```
deactivate
```

Before you deactivate the virtualenv shell:

```
pip install -e src
```

To run the test suite:

```
cd ~/git/cosc-learning-labs/src
pytest -t ../test
```

or

```
python setup.py test -a ../test
```

c. Git project cosc-learning-labs

Project cosc-learning-labs contains:

- this document
- a Python utility library
- sample scripts that can run from the command line
- iPython Notebooks

You must have git installed on the client computer. There are many git client applications to choose from, as listed on the git web-site:

<http://git-scm.com/downloads/guis>

Ubuntu:

```
sudo apt-get install git
```

A sample console follows for a Linux-style computer:

```
cd ~
mkdir git
cd git
git clone http://gerrit-open1.cisco.com/gerrit/cosc-learning-labs
```

If you do not want to be anonymous:

```
git clone ssh://<you>@gerrit-open1.cisco.com:29418/cosc-learning-labs
```

Sample output:

```
Cloning into 'cosc-learning-labs'...
remote: Counting objects: 773, done
remote: Finding sources: 100% (773/773)
```

Nathan John Sowatskey 3/3/2015 10:55

Comment [16]: We should do this first, else, for example, the standard.virl won't be available.

Nathan John Sowatskey 3/3/2015 10:55

Comment [17]: This is anonymous.

Nathan John Sowatskey 3/3/2015 10:56

Comment [18]: This only works if one has added a ssh key. We need a screenshot of that for DevHub.

```
remote: Total 773 (delta 375), reused 773 (delta 375)
Receiving objects: 100% (773/773), 4.68 MiB | 0 bytes/s, done.
Resolving deltas: 100% (375/375), done.
Checking connectivity... done.
```

d. Configure the client

Set environment variable NETWORK_PROFILE to 'learning_lab' or any other setting you prefer.

```
export NETWORK_PROFILE=learning_lab
```

The settings are read from Python module with path:

```
~/git/cosc-learning-labs/src/settings/${NETWORK_PROFILE}.py
```

... which is a module in the 'settings' package, which is in the project's top level directory 'src'.

To display the settings:

```
cat ~/git/cosc-learning-labs/src/settings/${NETWORK_PROFILE}.py
```

Output:

```
config = {
  'network_device': {'xrvr-1': {
    'address': '172.16.1.53',
    'port': 830,
    'password': 'cisco',
    'username': 'cisco'},
    'xrvr-2': {
    'address': '172.16.1.52',
    'port': 830,
    'password': 'cisco',
    'username': 'cisco'},
    'xrvr-999': {
    'address': '172.16.1.999',
    'port': 830,
    'password': 'cisco',
    'username': 'cisco'}},
  'odl_server': {'address': '172.16.1.1',
    'port': 8181,
    'password': 'admin',
    'username': 'admin'}}
```

e. Python package 'learning_lab'

Change directory:

```
cd cosc-learning-labs/src/learning_lab
```

Run a Python script:

```
./01_connected.py
```

Output:

```
Detect NETWORK_PROFILE: sandbox
Config loaded from module <module 'settings.sandbox' from
```

Nathan John Sowatskey 3/3/2015 10:56

Comment [19]: If this does not happen then "Milan" is the default, and we need to change that to be "default", or make "learning_lab" the default.

Nathan John Sowatskey 3/3/2015 10:58

Comment [20]: We need a section in this document that explains the project code structure with "basics" explained.

```
'../../settings/sandbox.py'>
Usage: from settings.config import config
connected ['controller-config']
```

6. iPython/Jupyter Notebook Server

This step is only necessary if you want to edit and run the sample code using the dynamic web-page notebooks. Sub-steps 4, 5 and 6 are only necessary if the web-server and web-browser are on different computers.

1. Install Required Python Packages

```
sudo easy_install --upgrade tornado
sudo apt-get -y install libzmq-dev
sudo easy_install --upgrade pyzmq
```

2. Install iPython/Jupyter notebook server:

```
sudo easy_install ipython
```

or, on **Ubuntu**:

```
sudo apt-get -y install ipython-notebook
```

3. Check if port number 8888 is in use:

```
netstat -antu | grep 8888
```

There is no output if the port is free, otherwise expect output like:

```
tcp        0      0 127.0.0.1:8888      0.0.0.0:*
LISTEN
```

If port 8888 is not free then find another port and use it in the configuration file as explained below.

4. Certificate

A certificate is required to enable SSL on the iPython server. If a certificate is not available then create an untrusted certificate:

```
cd ~
openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout
virlvm.pem -out virlvm.pem
```

The certificate file name and location are at your discretion. In the example above the file name is **virlvm.pem** in the current directory.

5. Create iPython Profile:

```
ipython profile create learning_lab
```

Note the directory where the config files are created. For example:

```
[ProfileCreate] Generating default config file:
u'/home/virl/.config/ipython/profile_learning_lab/ipython_config.py'
```

Nathan John Sowatskey 3/3/2015 12:05

Comment [21]: This should ideally be a separate document explaining what iPython is for.

Nathan John Sowatskey 3/3/2015 10:58

Comment [22]: We should use pip?

Nathan John Sowatskey 3/3/2015 10:59

Comment [23]: By implication, the above then applies on OSX?

Nathan John Sowatskey 3/3/2015 11:00

Comment [24]: Why?

Nathan John Sowatskey 3/3/2015 11:00

Comment [25]: Where is this used?

```
[ProfileCreate] Generating default config file:
u'/home/virl/.config/ipython/profile_learning_lab/ipython_notebook_config.py'
```

6. Edit iPython Notebook Profile:

Edit the profile file that was created by the previous step. For example:

```
vi ~/.config/ipython/profile_learning_lab/ipython_notebook_config.py
```

Insert the following lines into the profile file:

```
c.NotebookApp.password =
u'sha1:39e0c1c8eeac:7eebe244c251d04c38b75582ddfd14bde8a55c5b'
c.NotebookApp.ip = '*'
c.NotebookApp.certfile = u'/home/cisco/virlvm.pem'
c.NotebookApp.port = 8888
c.NotebookApp.open_browser = False
```

... after line 3 which contains:

```
c = get_config()
```

Adjust the file name and path of the certificate file to suit the way you provided/created the certificate.

Change port 8888 to the free port as per the relevant instruction, above.

The password hash is for "VIRL" (without the quotes).

Note: the elegant way to edit the configuration file is to uncomment lines instead of inserting them, as instructed above. It is up to **you**.

All the instructions above are explained in detail here:

http://ipython.org/ipython-doc/2/notebook/public_server.html

7. Run the iPython server:

```
cd ~/git/cosc-learning-labs/src/notebook
ipython notebook --profile=learning_lab
```

Sample output from starting iPython on the remote computer 172.16.1.1:

```
2015-02-10 16:36:30.248 [NotebookApp] Using existing profile dir:
u'/home/virl/.ipython/profile_learning_lab'
2015-02-10 16:36:30.253 [NotebookApp] Using system MathJax
2015-02-10 16:36:30.259 [NotebookApp] Serving notebooks from local
directory: /home/virl/git/cosc-learning-labs/src/notebook
2015-02-10 16:36:30.259 [NotebookApp] The IPython Notebook is running
at: https://[all ip addresses on your system]:8888/
2015-02-10 16:36:30.259 [NotebookApp] Use Control-C to stop this
server and shut down all kernels (twice to skip confirmation).
2015-02-10 16:36:30.259 [NotebookApp] WARNING | No web browser found:
could not locate runnable browser.
```

If the computer is headless, as above, you will need to open a browser on a different computer. The URL to open **is**:

<https://172.16.1.1:8888>

Nathan John Sowatskey 3/3/2015 12:06

Comment [26]: Where did the password come from?

Nathan John Sowatskey 3/3/2015 12:07

Comment [27]: Do we have a bundled configuration sample file?

Nathan John Sowatskey 3/3/2015 12:07

Comment [28]: Where does the IP address come from?

Verify that the URL begins with <https://> (not <http://>) and has a SSL padlock symbol. If the certificate you provided is untrusted then instruct the browser to accept it. When prompted for the password, as per the screen shot below, enter "VIRL" (don't enter the quotes).

IP[y]: Notebook

Password:

Log in

8. Choose the COSC Learning Lab Menu:

You should see a list of notebooks, including one named Menu, which you should select to open. In the screenshot below you can see "Menu" at the bottom, which is the intended starting point:

IP[y]: Notebook

Notebooks

Clusters

To import a notebook, drag the file onto the listing below or [click here](#).

/ home / viri / git / cosc-learning-labs / src / learning_lab /

[AccessControl](#)

[Capabilities](#)

[Configure](#)

[Copyright](#)

[HowToSetUpYourComputer](#)

[Http](#)

[Interface](#)

[Inventory](#)

[Menu](#)

9. Static iPython Notebooks

Pre-requisites:

```
sudo apt-get install pandoc
```

```
pip install pygments sphinx
```

```
sudo apt-get install pandoc texlive-latex-base texlive-latex-recommended  
texlive-latex-extra texlive-fonts-recommended
```

To convert the 'active' iPython pages to static html or markdown:

```
ipython nbconvert --to html --template full *.ipynb  
or  
ipython nbconvert --to markdown *.ipynb
```

7. Eclipse IDE

This step is only necessary if you want to edit and run the sample code using Eclipse IDE.

On Ubuntu:

```
sudo apt-get install eclipse
```

Update site:

```
pydev - http://pydev.sf.net/updates/
```

Relevant discussion of pydev and Eclipse:

<http://ntraft.com/eclipse-with-pydev-and-virtualenv>

The best place to set the environment variable NETWORK_PROFILE is:

```
Eclipse -> Preferences... -> PyDev -> Interpreters -> Python Interpreter ->  
Environment (tab) -> New... (button in lower panel)
```

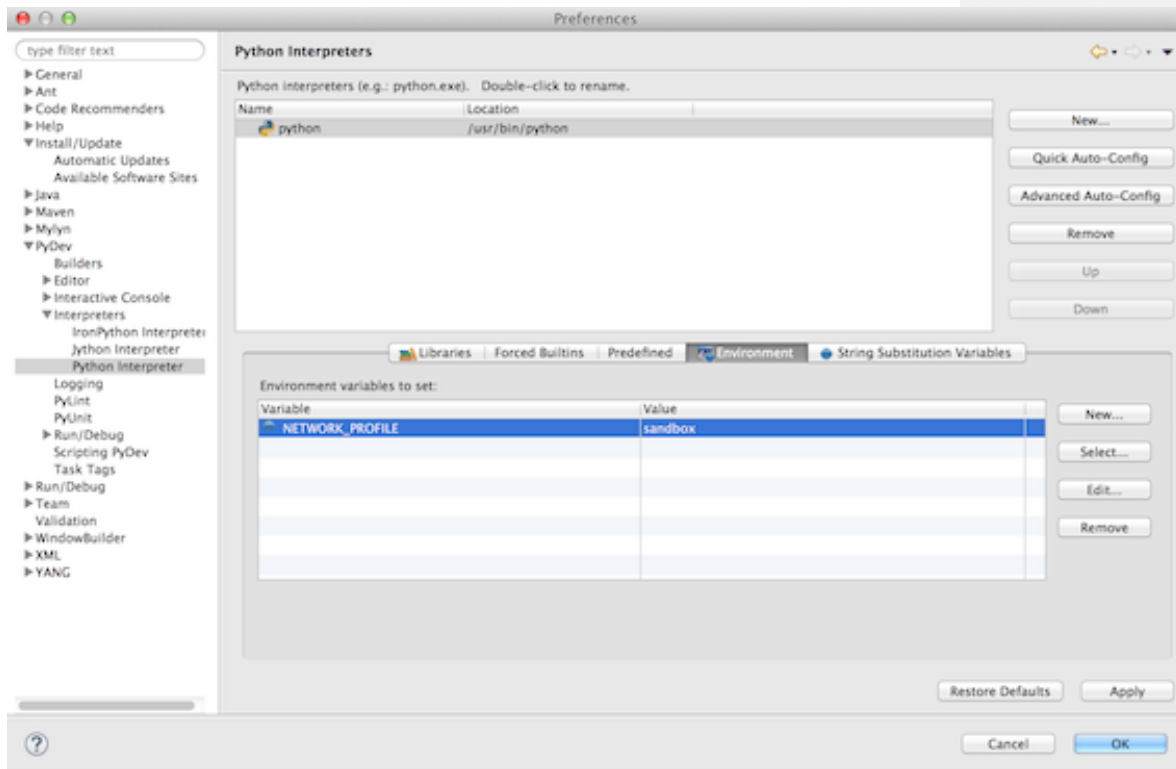
In the screen-shot, below, the environment variable NETWORK_PROFILE is highlighted and has the value "sandbox".

Nathan John Sowatskey 3/3/2015 12:07

Comment [29]: Separate document.

Nathan John Sowatskey 3/3/2015 12:08

Comment [30]: Should be pre-installed on the dev client VM in the technology pack. Use Eclipse JEE.



Import the project into Eclipse:

File -> Import ... -> General -> Existing Projects into Workspace -> Next (button)
-> Select root directory:

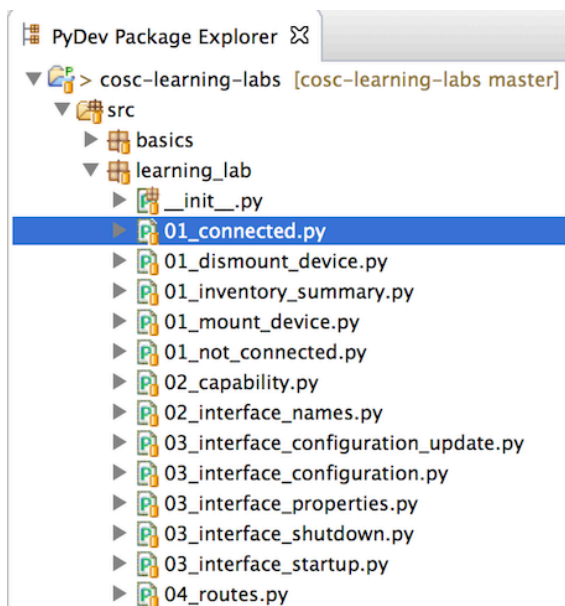
Enter the path to the directory that you previously cloned from git, named:

cosc-learning-labs

Open the project in the Explorer view and expand directories:

cosc-learning-labs -> src -> learning_lab

Select a script, as shown below in the screen-shot:



Run a script using the toolbar button (green, looks like a 'play' symbol) or use the context menu:

Run As -> Python Run

The Console View will contain the output of the selected script, as shown below in the screen shot:



The green 'Relaunch' button in the toolbar of the Console View, above, will re-run the same script. It is especially convenient when repeatedly editing and running.

Appendices

Each appendix contains miscellaneous information related to Viri VM or COSC.

Alternative Image of XR Device

There is a XR image on the Viri VM but you may want to simulate a different image. An alternative image can be uploaded to the Viri VM UWM.

From a Linux-style computer, assuming that UWM is listening on 172.16.1.1:19400:

```
curl -F "subtype=IOS XRv" -F "version=5.1.1.53U" -F
"release=5.1.1.53U" -F "image-
url=http://10.176.1.75/images/5.1.1.53U.vmdk" -u uwmadmin
http://172.16.1.1:19400/rest/images
```

A successful response is shown below. Note that you must enter the password "password" when prompted (without the " "):

```
Enter host password for user 'uwmadmin':
{
  "disk-usage": {
    "percentage-usage": "77.31",
    "total-disk-space-GB": "28.25",
    "used-disk-space-GB": "21.84"
  },
  "image": {
    "_owned": true,
    "checksum": "219c8e678480fdca3dcfb14522b7628b",
    "container_format": "bare",
    "created_at": "2015-02-09T19:48:24",
    "deleted": false,
    "deleted_at": null,
    "disk_format": "qcow2",
    "id": "e7d549d3-cd18-4ef3-8f7d-35d218b3add9",
    "is_public": true,
    "min_disk": 3,
    "min_ram": 0,
    "name": "IOS XRv-5.1.1.53U",
    "owner": "77db3c34bff849569550c1582181c7fe",
    "properties": {
      "config_disk_type": "cdrom",
      "hw_disk_bus": "ide",
      "hw_vif_model": "virtio",
      "release": "5.1.1.53U",
      "serial": "3",
      "subtype": "IOS XRv",
      "version": "5.1.1.53U"
    },
    "protected": false,
    "size": 599785472,
    "status": "active",
    "updated_at": "2015-02-09T19:48:26",
    "virtual_size": null
  }
}
```

If you prefer to break the command above into two steps then copy the image file to the Viri VM:

```
scp 5.1.1.53U.vmdk viri1@172.16.1.1:5.1.1.53U.vmdk
```

and then add the image to UWM:

```
curl -F "subtype=IOS XRv" -F "version=5.1.1.53U" -F
"release=5.1.1.53U" -F "image-path=/home/viri1/5.1.1.53U.vmdk" -u
uwmadmin http://172.16.1.1:19400/rest/images
```

The new XR image is confirmed visually by visiting the web page:

<http://172.16.1.1:19400/admin/images>

Nathan John Sowatskey 12/3/2015 07:54

Comment [31]: We seem to missing the step about using a local file and/or how the file became local such that we could scp.

The username is 'uwmadmin' and the password is 'password'.

In the screenshot, below, you can see the menu item "Images" on the left and the Netconf version of XR named IOS XRv-51.1.53U.

SWITCH MODE

User

Overview

Salt status

Projects

Users

Images

Flavors

Subtypes

Connectivity

Statistics

VM Control

Images

Disk usage (GB)

Images

Image	Project	Release
CSR1000v	admin	csr1000v-universalk9.03.13.00.S.154-3.S
IOS XRv	admin	iosxrv-k9-demo-5.2.2
IOS XRv-5.1.1.53U	uwmadmin	5.1.1.53U
IOSv	admin	vios-15.4.3M
NX-OSv	admin	titanium-final.7.1.0.ZD.0.363
server	admin	ubuntu.trusty64

VirI VM on Rack Space

Follow the instructions on the Sandbox web page:

<https://developer.cisco.com/site/devnet/sandbox>

To gain access to a remote computer you must *reserve* it for a particular start time and duration. The remote computer consumes the first 30 minutes of the reserved time in preparation.

You will receive a file by email when the remote machine is ready. The attached file extension is ".ovpn". If you do not receive this file by email then obtain it from the remote machine by FTP, as explained below.

Use the ".ovpn" file with an application running on your local computer to establish a connection with the remote computer. This IP address of the remote computer is determined by this application and is typically 172.16.1.1. There is a list of suitable applications on the Sandbox web site. For example, OpenVPN and TunnelBlick.

Download OVPN certificate file from VirI VM Server

To obtain the ".ovpn" file from the remote computer you must have:

1. The public IP address of the remote computer
2. A key-pair, consisting of two files, one private key and one public key.

The key-pair is stored in the git repository with website:

Nathan John Sowatskey 3/3/2015 12:09

Comment [32]: Separate document.

<https://gerrit-open1.cisco.com/gerrit/#/admin/projects/cosc-learning-labs>

An ssh public key can be added at this website:

https://devhub.cisco.com/sf/scm/do/listRepositories/projects.cosc_learning_labs/scm

The command to download a copy of the project onto your local computer is:

```
git clone http://gerrit-open1.cisco.com/gerrit/cosc-learning-labs
```

The path to the key-pair is:

```
cosc-learning-labs/etc/virl_key*
```

A sample console follows for a Linux-style local computer:

```
cd ~
mkdir git
cd git
git clone http://gerrit-open1.cisco.com/gerrit/cosc-learning-labs

Cloning into 'cosc-learning-labs'...
remote: Counting objects: 773, done
remote: Finding sources: 100% (773/773)
remote: Total 773 (delta 375), reused 773 (delta 375)
Receiving objects: 100% (773/773), 4.68 MiB | 0 bytes/s, done.
Resolving deltas: 100% (375/375), done.
Checking connectivity... done.

ls -l cosc-learning-labs/etc/virl_key*
-rwxrwxr-x 1 virl virl 1679 Feb  9 17:24 cosc-learning-
labs/etc/virl_key
-rwxrwxr-x 1 virl virl 406 Feb  9 17:24 cosc-learning-
labs/etc/virl_key.pub
```

Download the ".ovpn" file from the remote computer using the key-pair. In the sample command, below, the public IP address of the remote computer is 104.130.21.16.

Linux command:

```
sftp -i ~/git/cosc-learning-labs/etc/virl_key
virl@104.130.21.16:/home/virl/vpn-client.ovpn
```

The first time you connect to the remote computer, using the public IP address, the following message will appear on the console and you should respond with "yes" at the prompt, as shown:

```
The authenticity of host '104.130.21.16 (104.130.21.16)' can't be
established.
RSA key fingerprint is
7a:72:d9:b0:2b:ed:83:16:a2:fb:0a:de:8b:b5:55:57.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '104.130.21.16' (RSA) to the list of known
hosts.
```

A sample of a successful download of file: vpn-client.ovpn

```
Connected to 104.130.21.16.
Fetching /home/virl/vpn-client.ovpn to vpn-client.ovpn
```

```
/home/virl/vpn-client.ovpn          100% 6311      6.2KB/s
00:00

ls -l *ovpn
-rw-r--r--  1 user  staff   6311   9 Feb 17:36 vpn-client.ovpn
```

About the image of the Netconf enabled XR device

To login to the Virl VM using a key pair:

```
ssh -o StrictHostKeyChecking=no -i ~/git/cosc-learning-
labs/etc/virl_key virl@172.16.1.1
```

The first time you connect to the Virl VM, the following message will appear on the console and you should respond with "yes" at the prompt, as shown:

```
The authenticity of host 172.16.1.1 (172.16.1.1) can't be
established.
RSA key fingerprint is
7a:72:d9:b0:2b:ed:83:16:a2:fb:0a:de:8b:b5:55:57.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.1.1' (RSA) to the list of known
hosts.
```

Successful response:

```
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-45-generic x86_64)

* Documentation:  https://help.ubuntu.com/
Last login: Mon Feb  9 17:49:59 2015 from 2.220.163.215
```

Alternatively, the public IP address of the remote computer can be used. In the example below it is 104.130.21.16.

```
ssh -i ~/git/cosc-learning-labs/etc/virl_key virl@104.130.21.16
```

Upload the XR Netconf image as follows:

```
virl_uwm_client image-create --subtype "IOS XRv" --version 5.1.1.53U -
-release 5.1.1.53U --image-url
http://10.176.1.75/images/5.1.1.53U.vmdk
```

Successful response:

```
INFO      2015-02-09 17:58:28,269 virl.uwm.client Client.image_create
called args=('IOS XRv',
'5.1.1.53U',
'5.1.1.53U',
None,
None,
'http://10.176.1.75/images/5.1.1.53U.vmdk',
None) kargs={}
INFO      2015-02-09 17:59:25,747 virl.uwm.client image_create POST on
URL "http://localhost:19400/rest/images"
INFO      2015-02-09 17:59:25,747 virl.uwm.client image_create response
200 to POST on URL "http://localhost:19400/rest/images"
{'u'disk-usage': {'u'percentage-usage': u'67.92',
                  u'total-disk-space-GB': u'28.25',
                  u'used-disk-space-GB': u'19.18'},
 u'image': {'u'_owned': True,
            u'checksum': u'219c8e678480fdca3dcfb14522b7628b',
```

```

u'container_format': u'bare',
u'created_at': u'2015-02-09T17:59:20',
u'deleted': False,
u'deleted_at': None,
u'disk_format': u'qcow2',
u'id': u'1f431476-cabc-4669-b5a3-2964240920d7',
u'is_public': True,
u'min_disk': 3,
u'min_ram': 0,
u'name': u'IOS XRv-5.1.1.53U',
u'owner': u'c2ea4aleee104008a9a8fa3a1879af70',
u'properties': {u'config_disk_type': u'cdrom',
                 u'hw_disk_bus': u'ide',
                 u'hw_vif_model': u'virtio',
                 u'release': u'5.1.1.53U',
                 u'serial': u'3',
                 u'subtype': u'IOS XRv',
                 u'version': u'5.1.1.53U'},
u'protected': False,
u'size': 599785472,
u'status': u'active',
u'updated_at': u'2015-02-09T17:59:24',
u'virtual_size': None}}

```

The commands above assume that the image is available in the Sandbox at URL:

```
http://10.176.1.75/images/5.1.1.53U.vmdk
```

If the image is not located at the URL above then it can be obtained elsewhere.

To request by email:

```
devnet-samples@cisco.com
```

If you are attending a Cisco educational course you will receive instructions to download the image file from the Cisco intranet. The Virl VM does not have access to the Cisco intranet. In this situation the image file must be copied to your local computer and then copied to the remote computer.

For Cisco internal use: download the image file, into the current directory, using wget from a bash shell, as follows:

```
wget http://172.23.29.110/xrvm-images/5.1.1.53U.vmdk
```

Your computer must have 600 MB of disk space available. Sample output:

```

--2014-12-03 18:11:18--  http://172.23.29.110/xrvm-
images/5.1.1.53U.vmdk
Connecting to 172.23.29.110:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 599654400 (572M)
Saving to: '5.1.1.53U.vmdk'

```

The command to verify:

```
md5sum 5.1.1.53U.vmdk
```

Output:

```
0cab7bd5dc40d3a31a032b0c6e2dacef
```

Copy the image from your local computer to the remote computer using secure-file-transfer-protocol (SFTP). In the example below:

- The IP address 172.16.1.1 is the remote computer. The public IP address can also be used.
- The local file 5.1.1.53U.vmdk is the XR image

From the command line, such as bash:

```
sftp -i ~/git/cosc-learning-labs/etc/virl_key virl@172.16.1.1
```

Output:

```
Connected to 172.16.1.1.
```

The file to transfer should be in the current directory:

```
sftp> ll
```

You will see one or more files, including:

```
5.1.1.53U.vmdk
```

Initiate the transfer:

```
sftp> put 5.1.1.53U.vmdk
```

Wait for 100% complete:

```
Uploading 5.1.1.53U.vmdk to /home/cisco/5.1.1.53U.vmdk
5.1.1.53U.vmdk          100% 572MB 73.0KB/s
2:13:41
```

Confirm the file listing on the remote computer:

```
sftp> ls -l
```

Output:

```
-rw-r--r--  1 virl  virl  599654400 Feb  5 19:45 5.1.1.53U.vmdk
```

Close the FTP session:

```
sftp> quit
```

Display the checksum of the remote file:

```
ssh -i ~/git/cosc-learning-labs/etc/virl_key virl@172.16.1.1 md5sum
5.1.1.53U.vmdk
```

Output:

```
0cab7bd5dc40d3a31a032b0c6e2dacef 5.1.1.53U.vmdk
```

Compare the checksum of the remote file to the local file. In this example the local computer is OSX so the command is `md5` not `md5sum`:

```
md5 5.1.1.53U.vmdk
```

Output:

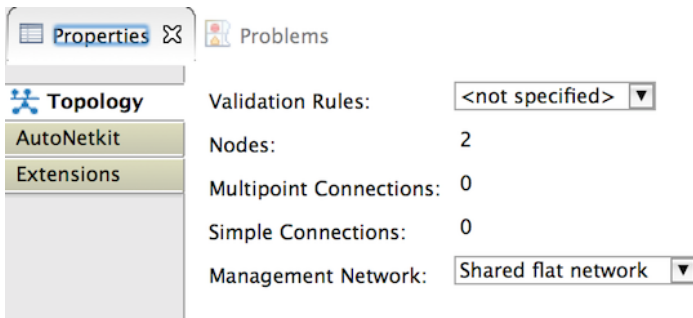
```
MD5 (5.1.1.53U.vmdk) = 0cab7bd5dc40d3a31a032b0c6e2dacef
```

The image file is now uploaded to the remote computer and can be added to the Viri VM. The following command is similar to the command shown earlier in this section but it references a file on the remote server instead of a URL. As before, the password is "password". The expected response is similar to that shown earlier for the similar command.

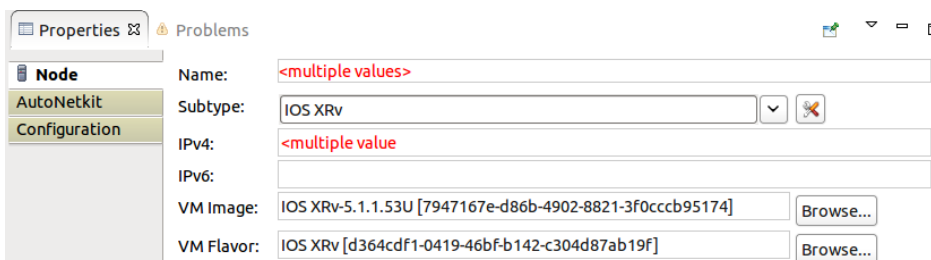
```
curl -F "subtype=IOS XRv" -F "version=5.1.1.53U" -F  
"release=5.1.1.53U" -F "image-path=/home/virl/5.1.1.53U.vmdk" -u  
uwadmin http://localhost:19400/rest/images
```

Custom Topology

In the properties for the topology, ensure that field 'Management Network' has the value "shared flat network", as shown in the screen-shot, below:



The XR node sub-type must be Netconf enabled. Select the XR nodes and assign values to the fields 'VM Image' and 'VM Flavor' as shown in the screen-shot, below:



Note that multiple nodes were selected, which explains why some fields show "<multiple values>".

Nathan John Sowatskey 3/3/2015 12:10

Comment [33]: We have documented this already. It should be moved to a separate document.

The following configuration is necessary on each XR device in the topology. If you select multiple devices you can apply this configuration to them all.

```
ssh server v2
ssh server netconf port 830
ssh timeout 120
netconf agent ssh
```

The path to the property is Properties window -> AutoNetkit tab -> Custom Configuration group -> Global.

The instructions, above, produce the screenshot:

