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

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

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

sudo apt-get install libreoffice

**1. Virl**

The installation instructions are here:

<http://virl-dev-innovate/VIRL_docs/>

And the OVA files are available on application.

For Cisco internal use the OVA files are here:

<http://wwwin-drrc.cisco.com/virl/download/>

and

[http://10.48.37.151/virl/download](http://10.48.37.151/virl/download/)

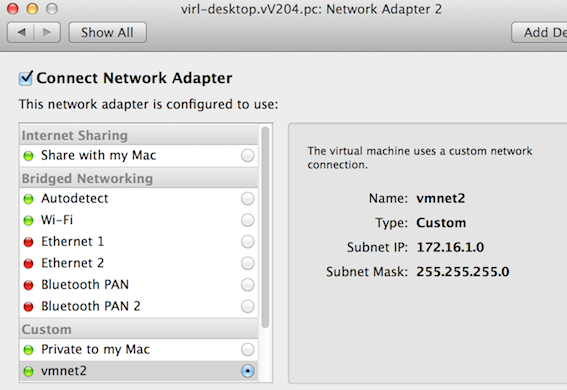
Typical installation: download file virl-desktop.vV204.pc.ova and import into VMware Fusion 6.0.4.

When booting the Ubuntu VM it may prompt you regarding available upgrades - choose the upgrade.

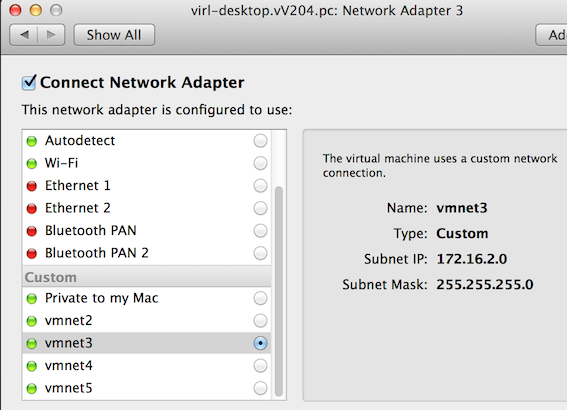
The user/password for the Virl VM is virl/VIRL.

In the VMware Settings (the spanner icon) set each Network Adapter X to Custom vmnetX and enable, where X is 2, 3, 4 and 5.

Screenshot for Network Adapter 2:



Screenshot for Network Adapter 3:



…continue for 4 and 5.

The configuration steps described above plus other necessary steps are described in detail on the Virl website (URL provided at the top of this section).

**2. XR Image**

There is an XR image in Virl but it may not support Netconf.

Load a specific XR image into Virl, which *does* have Netconf.

Obtain the image by applying to devnet-samples@cisco.com.

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

wget http://172.23.29.110/xrvr-images/5.1.1.53U.vmdk

Sample output:

--2014-12-03 18:11:18-- http://172.23.29.110/xrvr-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

**3. VM Maestro**

VMMaestro is already installed in the Virl/Ubuntu VM. VMMaestro also runs on OSX desktop and others. It can be installed from the download page of your VIRL server:

http://<VirlServer>/download

When VM Maestro is first started, use the credentials: username ‘guest’, password ‘guest’.

If VMMaestro stops responding or otherwise goes wrong, delete directory vmmaestro from the home directory. All topologies are stored there so preserve what you need prior to deletion.

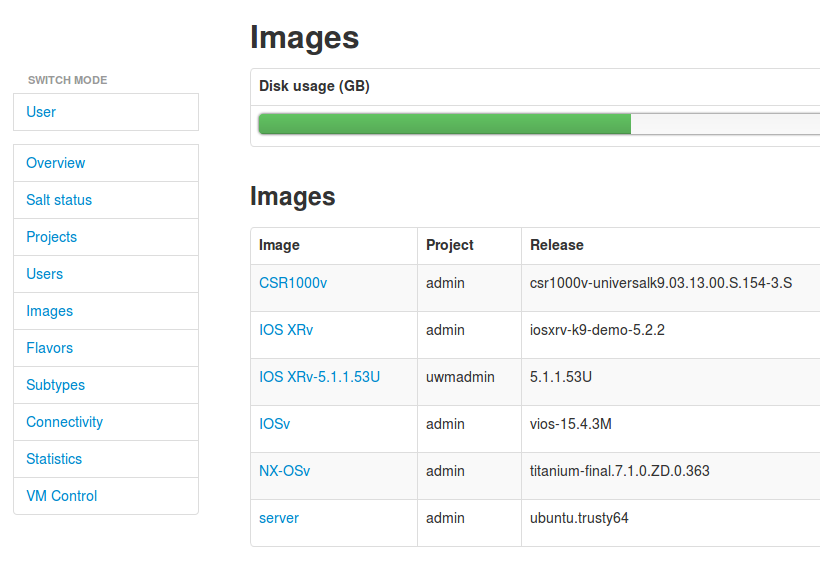
**4. UWM**

To load the image file into Virl use UWM as per the instructions here:

<http://virl-dev-innovate.cisco.com/assets/user_workspace_management.pdf>

As per the PDF, to login, the username is ‘uwmadmin’, password is ‘password’. The web page to login is http://<VirlServer>:19400/user/login. Obviously you can use <http://localhost:19400/user/login> from a web browser running on the Virl VM.

Select option “Images” from the menu on the left of the web page. Here is a screen shot of the resulting web page:



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

If the required image is not displayed then it must be loaded, using the “Add” button to the right of the web page (not shown in the screenshot, above).

Enter the fields that describe the image to add. The sub-type is “IOS XRv” and the fields Name and Release should contain the information contained in the name of the downloaded file, for example 5.1.1.53U. Field Image Path is the location of the file downloaded in the prior step (XR Image). For example /home/virl/5.1.1.53U.vmdk.

Note about Salt: in the Virl documentation the instructions regarding Salt are for paying customers who are provided with keys. For Cisco internal use the Salt ID is your email address with the ampersand replaced with a period. That is, for [john.doe@cisco.com](mailto:john.doe@cisco.com) the Salt ID is john.doe.cisco.com. For the list of master servers, enter them using a comma as a delimiter. For example, “virl-salt-master-3.cisco.com,virl-salt-master-4.cisco.com”. Do not enter any public or private keys. This information is relevant to the UWM menu option “Salt status” which can be seen on the left in the screenshot, above.

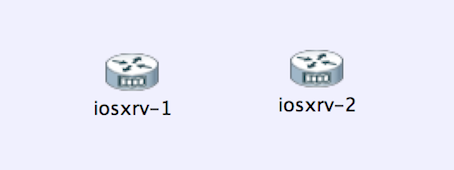
**5. Create and Run Topology in VM Maestro**

In VMMaestro:

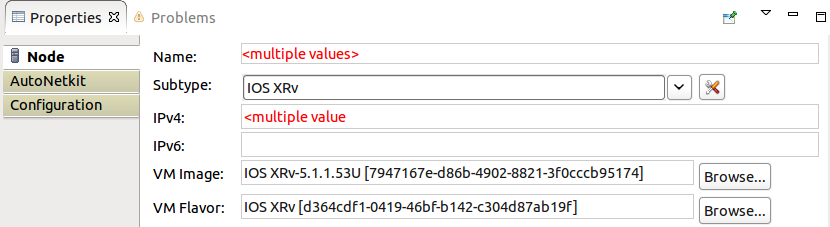
You may need to expand the list of Node Subtypes. To do so, use the VMMaestro user interface: File -> Preferences -> Node Subtypes -> Fetch from Server -> <confirm> -> Ok

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 resulting topology should contain one or more XRv nodes. For example:



Select the XR node(s) and assign values to the fields as shown in the screenshot, below:



Note that multiple nodes were selected, which explains why some fields show “<multiple values>” in some two fields.

\* 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

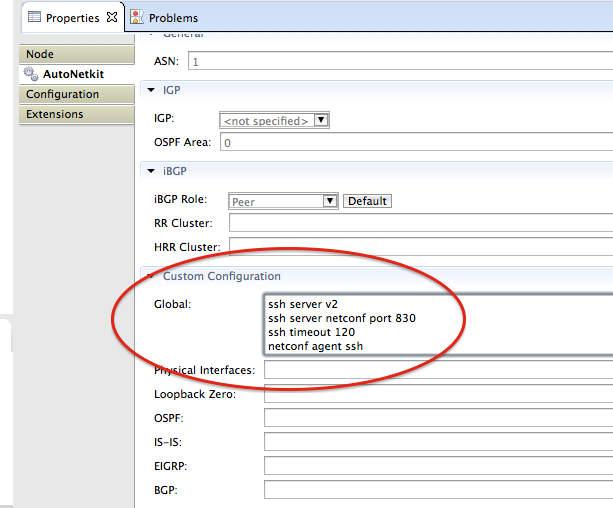
ssh server netconf port 830

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:



To start the simulation:

\* Run menu -> Update Router Configurations -> do not view AutoNetKit output

\* Run menu -> Launch Simulation -> Simulation Perspective

\* Wait for 10 minutes at least. Install the COSC controller while you wait (see below).

\* Connect to the console of each XRv device. Authenticate with credentials cisco/cisco. Note that prior to Virl VM version 204 the credentials were test/test.

You can open the console from the command line using telnet. Or, using VMMaestro, the context menu (right click) of the entry (for each device) in the Simulations tree view (Simulations perspective). Note that it may take longer than you imagine for the XRv device to become available, so be prepared for this not to work at first.

* On each virtual device, apply the command below and accept the defaults:

crypto key generate dsa

\* Get the management IP address with this command:

sh ip int br

\* Test device from command shell (e.g. bash), using the actual management IP address for MgmtEth0/0/CPU0/0 from the previous command.

ssh -p 830 cisco@172.16.1.57 -s netconf

The expected response, after confirming the server key and supplying the password “cisco”, is an XML document starting with “<hello xmlns="urn:ietf .

If you see "no hostkey alg”, the you forgot to enter the "crypto key generate dsa” command described above.

Note: the simulation will not succeed if installation of the Virl VM is incomplete. For example, from Virl VM version 204 you must configure the Salt ID. See the Virl VM documentation as per the URL provided at the top of this document.

**6. Cisco Open SDN Controller (COSC)**

The controller is a service that can run in the Virl VM or a different VM.

The controller is not pre-installed in the Virl/Ubuntu VM.

\* Download:

The download link (a URL of a tar file) is available on request from devnet-samples@cisco.com.

For Cisco internal use, look for artifacts in directory:

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

such as:

<http://codehub-one-fw-ci:8085/auth/view/cdl-eft-throttle/job/cdl-eft-autorelease/lastSuccessfulBuild/artifact/integration/distributions/cdl/target/distribution-karaf-1.0.0-00003.tar.gz>

Note: the URL above is for version 3 of CDL. At the time of writing only version 2 has been used for the configuration described by this document.

Note: prior to the release of COSC the controller was ODL version Helium: <http://www.opendaylight.org/software/downloads>

\* Unpack:

Choose a directory to contain the controller. For example:

mkdir ~/cosc

cd ~/cosc

wget <url-of-tar-ball>

Unpack tar ball and cd to bin sub-directory.

tar xvf distribution-karaf-1.0.0-00003.tar.gz

cd bin

ls

Expected output:

client instance.bat setenv shell.bat status stop.bat

client.bat karaf setenv.bat start status.bat

instance karaf.bat shell start.bat stop

\* Configure

Modify the environment with the command:

source setenv

\* Run

Run the script named start in the sub-directory ‘bin’:

./start

\* Verify

To verify the controller from a web browser:

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

… and expect a HTML page. Replace ‘localhost’ with machine running the controller, if appropriate. The COSC username and password are both ‘admin’.

Verify advanced features of the controller with URL:

<http://localhost: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 - [http://localhost:8181/restconf/config/opendaylight-inventory:nodes](http://localhost:8181/restconf/config/opendaylight-inventory:nodespopd)

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

**7. Python**

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

sudo apt-get upgrade python

Install required packages:

sudo easy\_install lxml

sudo easy\_install requests

sudo easy\_install ipaddress

**8. 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

There is no anonymous “git clone” for project cosc-learning-labs, so you must login, as yourself, to:

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

Note: authentication is via keys, therefore your public key must be uploaded to the website. See this helpful web page for instructions:

<https://help.github.com/articles/generating-ssh-keys>

Download:

mkdir ~/git

cd ~/git

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

(where <you> is your username as registered with the website)

cd cosc-learning-labs/src/learning\_lab

Set environment variable NETWORK\_PROFILE to ‘learning\_lab’ or any other setting.

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 top level directory ‘src’.

Run scripts (in directory src/learning\_lab):

./01\_connected.py

./01\_mount\_device.py

etc

**7. iPython/Jupyter Notebook Server**

\* Install Required Python Packages

sudo easy\_install --upgrade tornado

sudo easy\_install --upgrade pyzmq

\* Install iPython/Jupyter notebook server:

sudo easy\_install ipython

or, on Ubuntu

sudo apt-get install ipython-notebook

\* 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.

\* Certificate

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

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.

\* Create iPython Profile:

ipython profile create learning\_lab

Sample output:

[ProfileCreate] Generating default config file: u'/home/virl/.config/ipython/profile\_learning\_lab/ipython\_config.py'

[ProfileCreate] Generating default config file: u'/home/virl/.config/ipython/profile\_learning\_lab/ipython\_notebook\_config.py'

\* 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/virl/virlvm.pem'

c.NotebookApp.port = 8888

… 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>

\* Run the iPython server:

cd ~/git/cosc-learning-labs/src/learning\_lab/

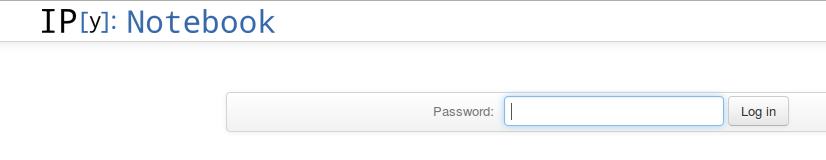
ipython notebook --profile=learning\_lab

A web page will open in a browser.

Verify that the URL begins with https:// (not http://) and has a SSL padlock symbol.

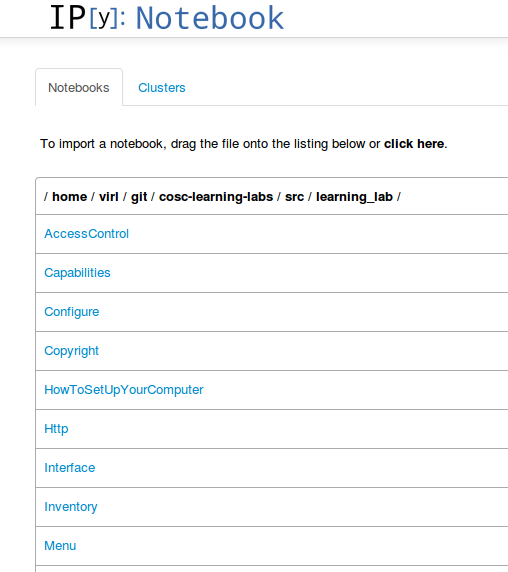
If the certificate you provided is untrusted then instruct the browser to accept it.

When prompted for the password, as per the screen shot below, enter “VIRL” (don’t enter the quotes).



\* Run the COSC Learning Lab:

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:



**8. Static documentation**

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

ipython nbconvert --to html \*.ipynb

or

ipython nbconvert --to markdown \*.ipynb

**9. Mininet**

\* Install

sudo apt-get install mininet

sudo service openvswitch-controller stop

sudo update-rc.d openvswitch-controller disable

\* Test

sudo mn --test pingall

\* Start

Choose *one* of the following commands:

* sudo mn --mac --controller remote,ip=127.0.0.1
* sudo mn --mac --controller remote,ip=192.168.255.205 --topo tree,2 --switch ovsk,protocols=OpenFlow13
* sudo mn --mac --controller remote,ip=192.168.255.205 --topo single --switch ovsk,protocols=OpenFlow13

\* Documentation

<http://mininet.org/download>

To build the documentation locally:

git clone git://github.com/mininet/mininet

mininet/util/install.sh –fw

**10. Eclipse**

On Ubuntu:

sudo apt-get install eclipse

Update site:

pydev - <http://pydev.sf.net/updates/>

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”.

