Lab One: Understanding the AM API Using Named Data Networking

Adapted From:
http://groups.geni.net/geni/wiki/GENIExperimenter/Tutorials/jacks/GettingStarted_PartII_ndn/Procedure

Overview

This is a slightly more complicated beginning exercise on GENI. You will:

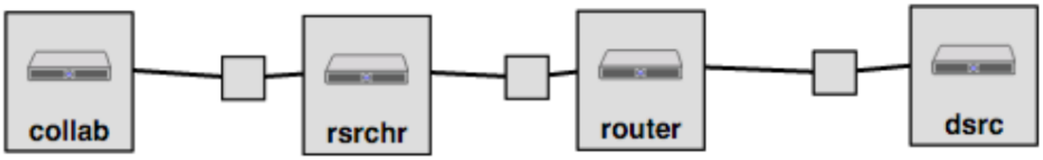
1. Understand more of the behind the scenes of GENI
2. Learn about RSpecs
3. Learn about the Omni experimenter tool

This experiment will focus on the Named Data Networking (NDN) Future Internet Architecture. You will install NDN software on the nodes in your network and run on application on top of the software to fetch content by name. Following the session, you can instrument your application using the GENI Desktop to see graphs of traffic on the network links. This will give you greater insight into when application data is fetched from a cache ad when it comes all the way from a data source.

Sections

Design and Setup

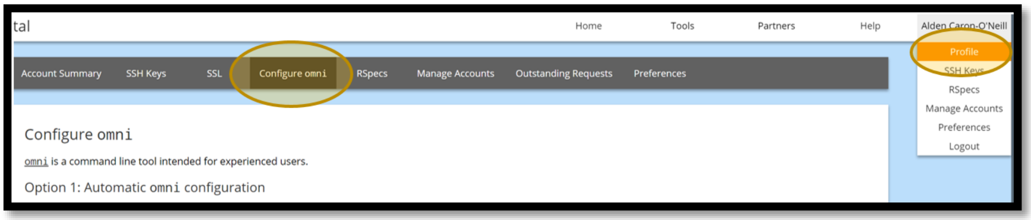
1. Design the Experiment



The experiment will consist of four virtual machines in a single aggregate connected by layer 2 links. For this experiment, select an InstaGENI aggregate.

2. Establish the Environment

a. You will need a GENI account setup with SSH keys (if you have completed lab zero you are all set).
b. Configure Omni
   i. Log into the GENI Portal.
   ii. Click on your name in the upper right hand corner, and then select the "Profile" tab menu item. Click on "Configure omni".

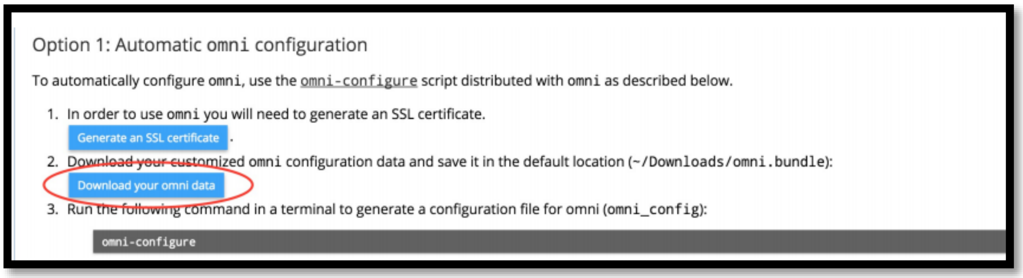


Configure omni

omni is a command line tool intended for experienced users.

Option 1: Automatic omni configuration

iii. Click “Download your omni data”.



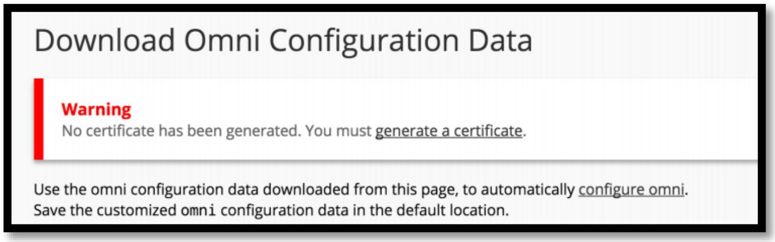
Option 1: Automatic omni configuration

To automatically configure omni, use the omni-configure script distributed with omni as described below.

1. In order to use omni you will need to generate an SSL certificate.
   Generate an SSL certificate .
2. Download your customized omni configuration data and save it in the default location (~/Downloads/omni.bundle):
   Download your omni data
3. Run the following command in a terminal to generate a configuration file for omni (omni_config):

```
omni-configure
```

iv. If you have not already made a GENI certificate, you will need to generate a certificate. Choose the simple option if you are not sure which one. Click ‘Generate Combined Certificate and Key File’ and then ‘Close’ to return to the download page. If the warning persists, reload the page.



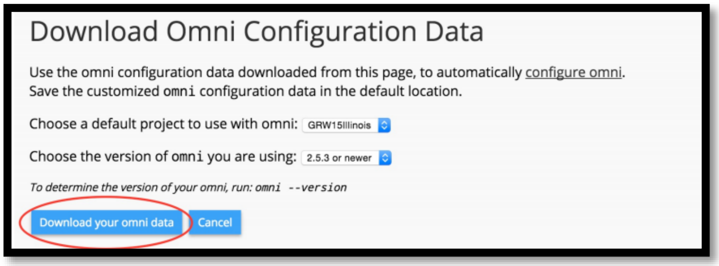
Download Omni Configuration Data

Warning
No certificate has been generated. You must generate a certificate.

Use the omni configuration data downloaded from this page, to automatically configure omni.
Save the customized omni configuration data in the default location.

v. For now, you will want to set your default project to be the one you have joined for this webinar. In the future, you can change the default project from the command line in Omni.

vi. Then click “Download your omni data”.



Download Omni Configuration Data

Use the omni configuration data downloaded from this page, to automatically configure omni.
Save the customized omni configuration data in the default location.

Choose a default project to use with omni: GRW15Illinois

Choose the version of omni you are using: 2.5.3 or newer

To determine the version of your omni, run: omni --version

vii. The bundle will likely be saved at *~/Downloads/omni.bundle*

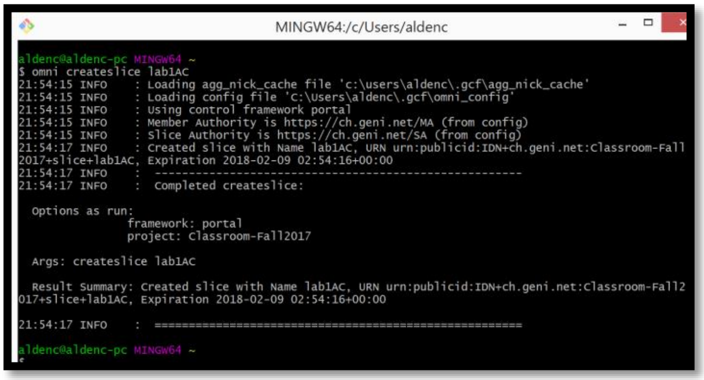
viii. In a terminal window, type

```
$ omni-configure
```

3. Obtain Resources

a. Create a slice using omni and a chosen slice name. As we must have unique slice names within the project, we recommend something like lab1<your_initials>.

```
$ omni createslice SLICENAME
```



MINGW64:/c/Users/aldenc

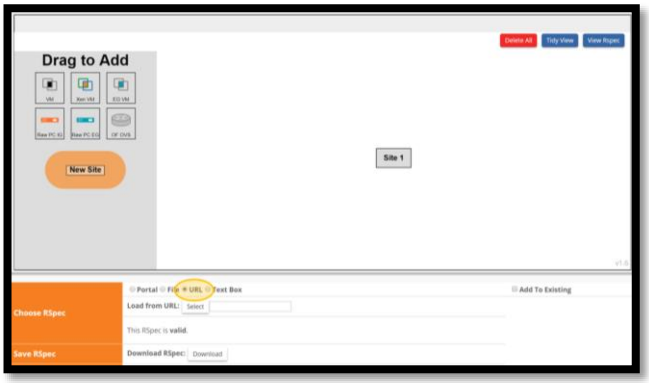
```
aldenc@aldenc-pc MINGW64 ~
$ omni createslice lab1AC
21:54:15 INFO     : Loading agg_nick_cache file 'c:\users\aldenc\.gcf\agg_nick_cache'
21:54:15 INFO     : Loading config file 'C:\Users\aldenc\.gcf\omni_config'
21:54:15 INFO     : Using control framework portal
21:54:15 INFO     : Member Authority is https://ch.geni.net/MA (from config)
21:54:15 INFO     : Slice Authority is https://ch.geni.net/SA (from config)
21:54:17 INFO     : Created slice with Name lab1AC, URN urn:publicid:IDN+ch.geni.net:Classroom-Fall
2017+slice+lab1AC, Expiration 2018-02-09 02:54:16+00:00
21:54:17 INFO     :  ------------------------------------------------------
21:54:17 INFO     :  Completed createslice:

  Options as run:
                framework: portal
                project: Classroom-Fall2017

  Args: createslice lab1AC

  Result Summary: Created slice with Name lab1AC, URN urn:publicid:IDN+ch.geni.net:Classroom-Fall2
017+slice+lab1AC, Expiration 2018-02-09 02:54:16+00:00

21:54:17 INFO     :  ======================================================

aldenc@aldenc-pc MINGW64 ~
$
```

b. Load a simple topology in Jacks

i. Navigate to the GENI Portal and open the Slice page for your created slice.

ii. Press **Add Resources** to open up Jacks.

iii. From the **Choose RSpec** menu, select the URL button.



○ Portal ○ File ● URL ○ Text Box

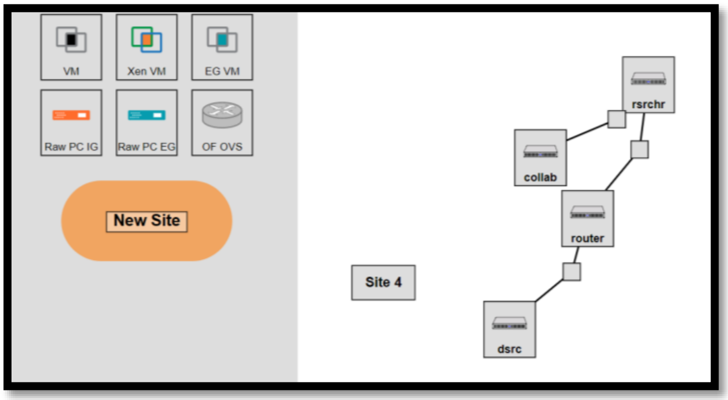
□ Add To Existing

Load from URL: Select

This RSpec is valid.

Download RSpec: Download

iv. Type (or copy paste) in: `http://www.gpolab.bbn.com/experiment-support/LabOne/ccnx/lab1ccn-line-PARTIAL.xml`
Click ‘Select’ and a topology should appear with four nodes as so:



c. Modify the RSpec to automatically install and execute CCNX software.
We need to install the following software on the nodes:

i. The CCNX software (ccnx-0.6.2.tar.gz)

ii. Scripts that set up the CCNX software (ccnx-setup.tar.gz)

iii. Scripts used to pull atmospheric precipitation data using the CCNX protocol (ccnx.atmos.tar.gz)

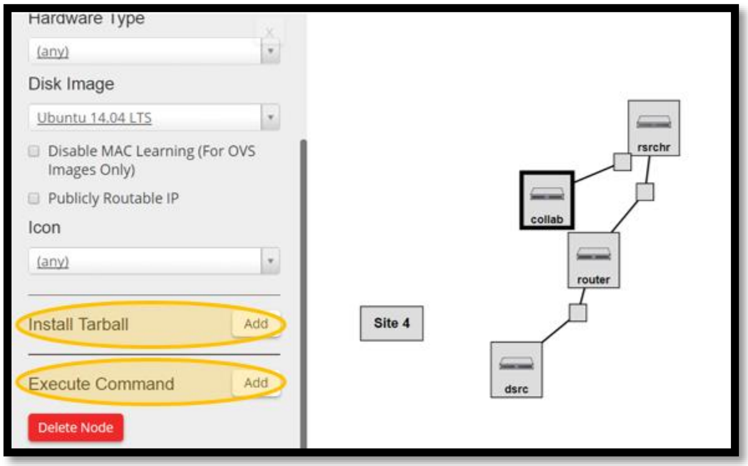
When the nodes start up, we need the following scripts to execute:

i. Script that sets up the node (node-setup)

ii. Script that sets up the ccnx-protocol (ccnx-setup)

iii. Script that sets up ccnx protocol routes (add-precip-routes)

d. Click the **collab node** to manipulate details of the node configuration.
Notice the two sections where we can “Install Tarball” or “Execute Command”.



Hardware Type

(any)

Disk Image

Ubuntu 14.04 LTS

□ Disable MAC Learning (For OVS Images Only)

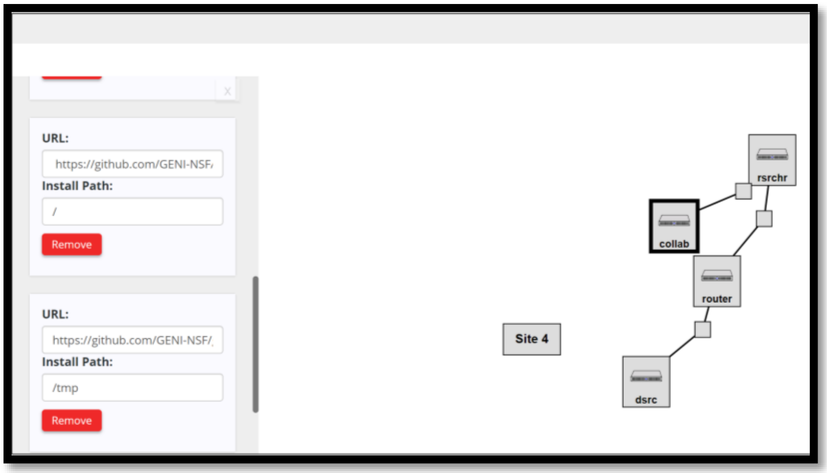
□ Publicly Routable IP

Icon

(any)

i. You will add three install scripts.

1. **URL:** `https://github.com/GENI-NSF/geni-tutorials/raw/master/OldTutorials/ContentCentricNetworking/ccnx-0.6.2.tar.gz`
   Install Path: /
2. **URL:** `https://github.com/GENI-NSF/geni-tutorials/raw/master/OldTutorials/ContentCentricNetworking/ccnx-atmos.tar.gz`
   Install Path: /
3. **URL:** `https://github.com/GENI-NSF/geni-tutorials/raw/master/OldTutorials/ContentCentricNetworking/ccnx-setup.tar.gz`
   Install Path: `/tmp`

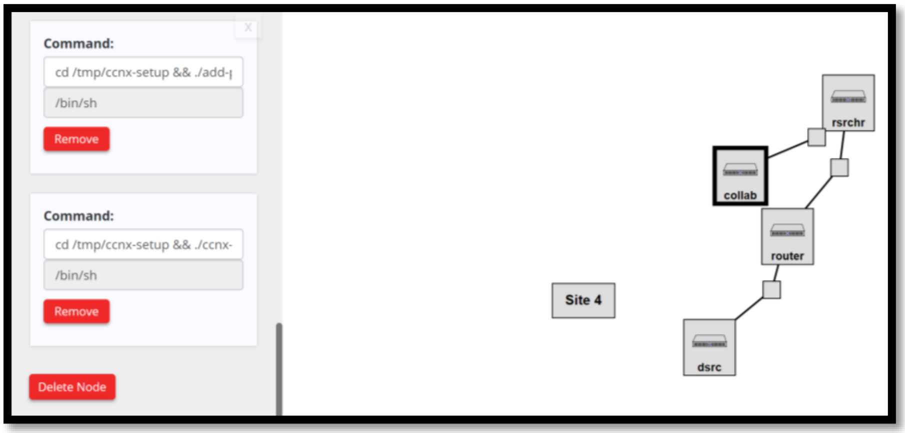


URL: https://github.com/GENI-NSF/
Install Path: /

URL: https://github.com/GENI-NSF/
Install Path: /tmp

ii. In a similar manner, use the “Add” button under Execute Scripts to add three execute services on this node.

1. **`Script:`** `cd /tmp/ccnx-setup && ./node-setup`
2. **`Script:`** `cd /tmp/ccnx-setup && ./add-precip-routes rsrchr`
3. **`Script:`** `cd /tmp/ccnx-setup && ./ccnx-setup router 4`



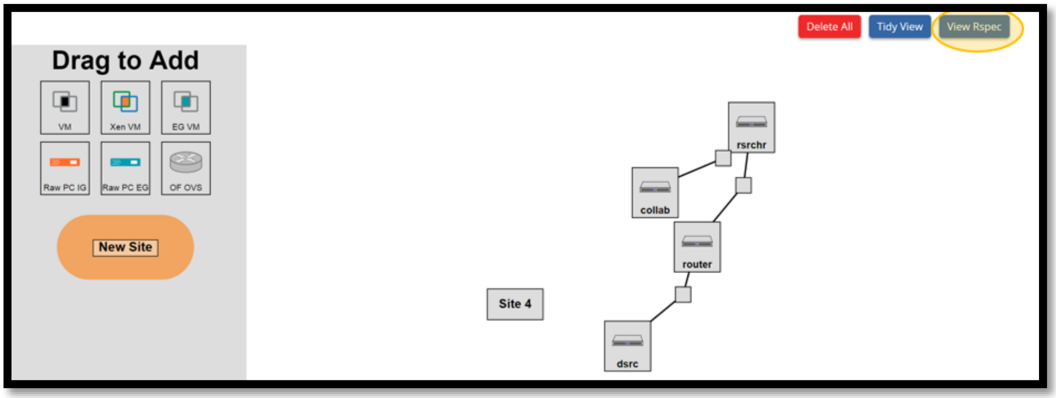
Command: cd /tmp/ccnx-setup && ./add-|
/bin/sh

Command: cd /tmp/ccnx-setup && ./ccnx-
/bin/sh

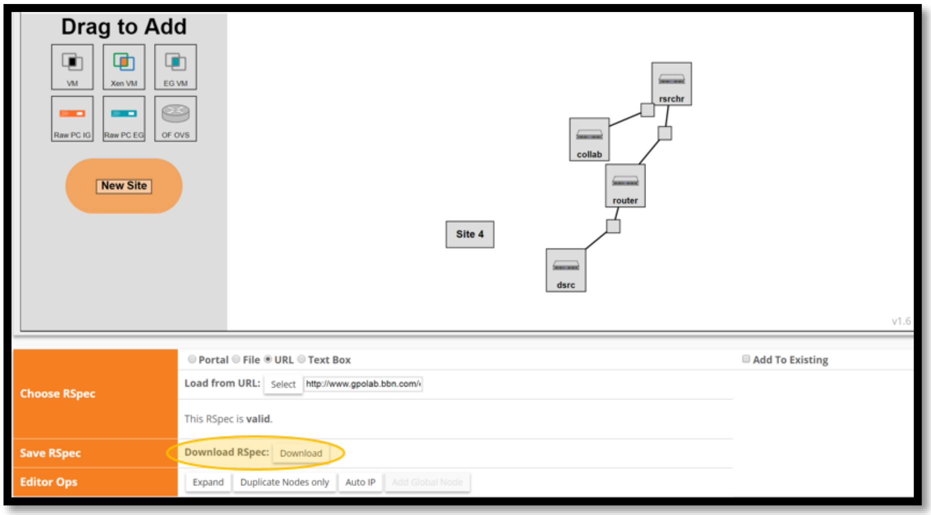
iii. The other nodes already have their script installs specified. You can check this by clicking on the other nodes.

e. Export the modified request RSpec.
Now we will pull back some of the covers and inspect exactly what Jacks has been doing for us when preparing the RSpecs for the experiments we design. Each node and link has a corresponding element in the RSpec, and the details of the component configuration (such as the install and execute services we requested above) are specified with attributes, or sometimes child elements, within those portions of the document.

i. From the Add Resources window, select the View RSpec button on the top right hand side. This will bring up a pane showing the current RSpec -- please take a moment to inspect it. The `<node>` and `<link>` elements contain the specification for the components we will request, and if you look closely, you will be able to see the install and execute service elements you added earlier. Click on the "View RSpec" button to return to the main screen.



ii. Use the 'Download' button in the lower left part of the screen next to Save RSpec to make a local copy of your RSpec with the name `SLICENAME_request_rspec.xml`



○ Portal ○ File ● URL ○ Text Box   □ Add To Existing

Load from URL: http://www.gpolab.bbn.com/

This RSpec is valid.

Download RSpec:

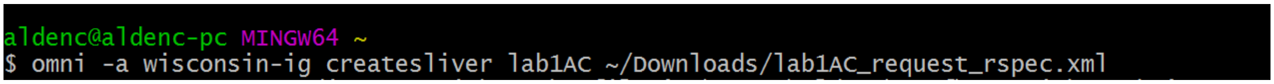
f. Instantiate the new experiment using Omni

Now switch back to the terminal and use the command line Omni client.

Enter the command:

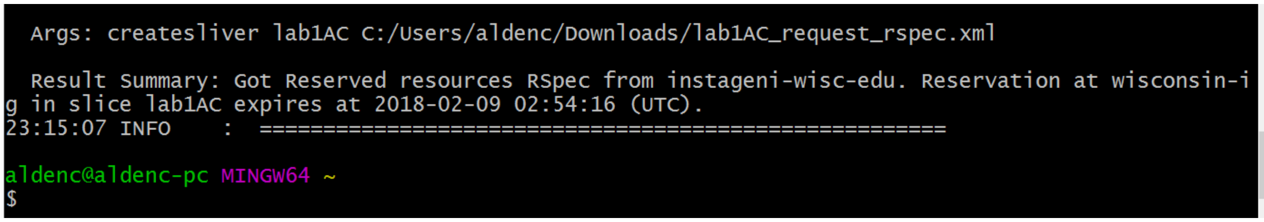
```
$ omni -a AM_NICKNAME createsliver SLICENAME RSPEC_FILE
```

Make sure to replace AM_NICKNAME with the nickname of an InstaGENI aggregate (use `omni nicknames` to find one), replace SLICENAME with your slice name, and RSPEC_FILE with the path to your downloaded file.



```
aldenc@aldenc-pc MINGW64 ~
$ omni -a wisconsin-ig createsliver lab1AC ~/Downloads/lab1AC_request_rspec.xml
```

If all goes well, after a few minutes and lots of text you should see:



```
  Args: createsliver lab1AC C:/Users/aldenc/Downloads/lab1AC_request_rspec.xml

  Result Summary: Got Reserved resources RSpec from instageni-wisc-edu. Reservation at wisconsin-i
g in slice lab1AC expires at 2018-02-09 02:54:16 (UTC).
23:15:07 INFO     : ====================================================

aldenc@aldenc-pc MINGW64 ~
$
```

This indicates you have successfully reserved the resources detailed in your RSpec!

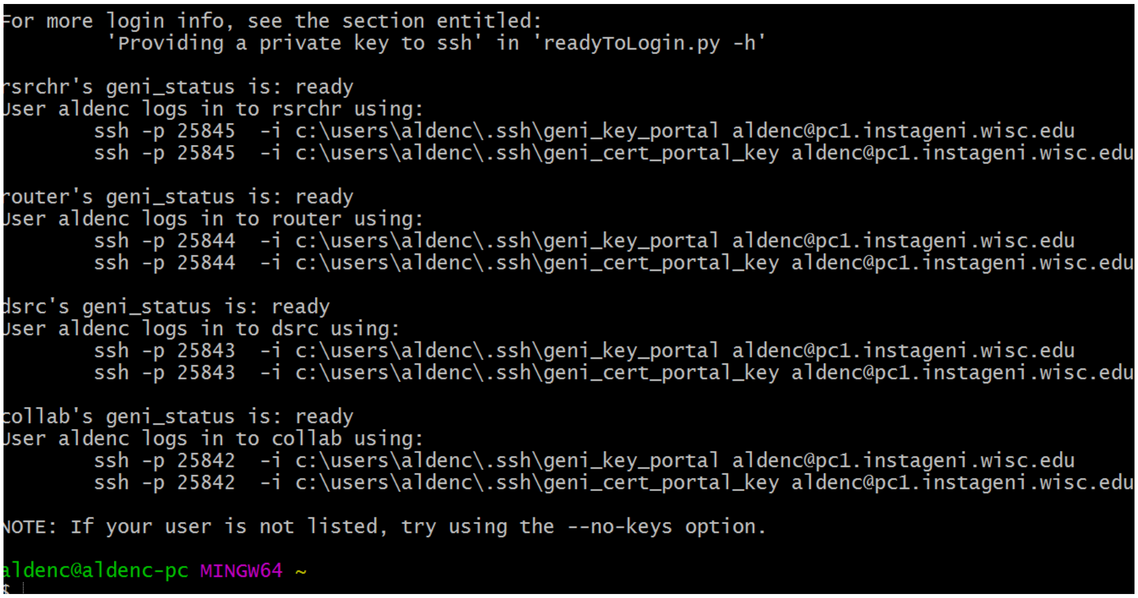
Execution

4. Wait for resources to be ready

Use the following command to check if your nodes are ready for use:

```
$ readyToLogin -a AM_NICKNAME SLICENAME
```

When your nodes are ready, you should get a screen response like below. It may take a few minutes, and in the meantime, the nodes may say their status is “changing”.



```
For more login info, see the section entitled:
         'Providing a private key to ssh' in 'readyToLogin.py -h'

rsrchr's geni_status is: ready
User aldenc logs in to rsrchr using:
        ssh -p 25845  -i c:\users\aldenc\.ssh\geni_key_portal aldenc@pc1.instageni.wisc.edu
        ssh -p 25845  -i c:\users\aldenc\.ssh\geni_cert_portal_key aldenc@pc1.instageni.wisc.edu

router's geni_status is: ready
User aldenc logs in to router using:
        ssh -p 25844  -i c:\users\aldenc\.ssh\geni_key_portal aldenc@pc1.instageni.wisc.edu
        ssh -p 25844  -i c:\users\aldenc\.ssh\geni_cert_portal_key aldenc@pc1.instageni.wisc.edu

dsrc's geni_status is: ready
User aldenc logs in to dsrc using:
        ssh -p 25843  -i c:\users\aldenc\.ssh\geni_key_portal aldenc@pc1.instageni.wisc.edu
        ssh -p 25843  -i c:\users\aldenc\.ssh\geni_cert_portal_key aldenc@pc1.instageni.wisc.edu

collab's geni_status is: ready
User aldenc logs in to collab using:
        ssh -p 25842  -i c:\users\aldenc\.ssh\geni_key_portal aldenc@pc1.instageni.wisc.edu
        ssh -p 25842  -i c:\users\aldenc\.ssh\geni_cert_portal_key aldenc@pc1.instageni.wisc.edu

NOTE: If your user is not listed, try using the --no-keys option.

aldenc@aldenc-pc MINGW64 ~
$
```

5. Try out the CCN protocol

The install and execute services requested in our RSpec have already started, and nodes in our experiment should be running the CCN protocol. Our experiment consists of:

i. A data source node `dsrc1` that holds precipitation data from the US National Oceanic and Atmospheric Administration (NOAA)

ii. A researcher node `rsrchr` that gets data from the data source

iii. A collaborator node `collab` that gets data from the researcher

In the CCN protocol, data is accessed by name. In our case, we use a program called client to get precipitation data by date range (e.g. precipitation between 1902/01/01 and 1902/01/02). All nodes cache data for a certain period. When a node receives a request for data, it checks its local cache. If the data is in its cache, it returns that data. Otherwise, it forwards it on to its neighbor.

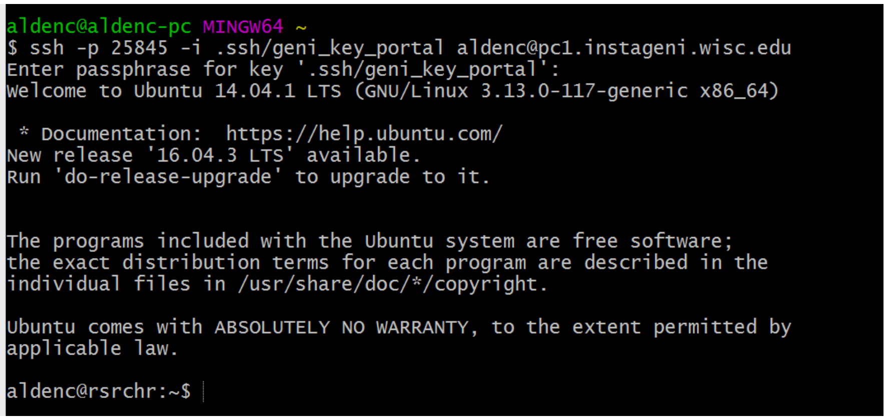
We verify this caching behavior by logging into the researcher node and using the client program to get precipitation data for a certain data range. Then the client displays how long it took to get the data. We repeat the same request and compare the time; it should be faster because the data is coming out of a cache. We can repeat this experiment with a variety of data ranges. You can repeat these steps on the collaborator node as well.

An optional section uses GENI Desktop to visualize the data requests. We will not be going through this part in the webinar, but if so interested, do not delete your setup at the end and run through the instructions by yourself (feel free to email me with any questions!). They can be found at the end of the file.

Desktop Instructions

Now we are going to run the CCN application.

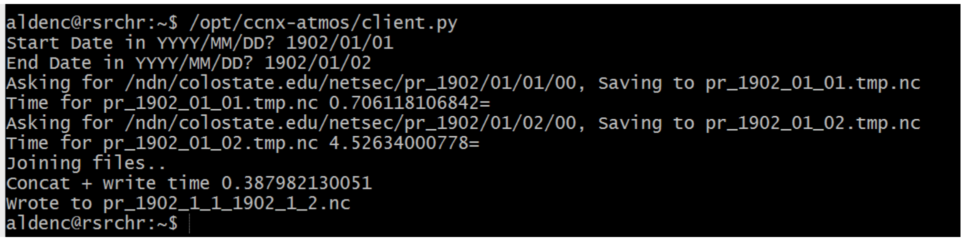
a. Log into the `rsrchr` node using the `SSH` command returned by `readyToLogin.` (Note that the login may be slightly different depending on your terminal, for example the path to your ssh key may need to be adjusted).



```
aldenc@aldenc-pc MINGW64 ~
$ ssh -p 25845 -i .ssh/geni_key_portal aldenc@pc1.instageni.wisc.edu
Enter passphrase for key '.ssh/geni_key_portal':
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-117-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

aldenc@rsrchr:~$
```

b. Once logged in, ask for precipitation data from 1 Jan 1902 to 2 Jan 1902:

```
$ /opt/ccnx-atmos/client.py
Start Date in YYYY/MM/DD? 1902/01/01
End Date in YYYY/MM/DD? 1902/01/02
```



```
aldenc@rsrchr:~$ /opt/ccnx-atmos/client.py
Start Date in YYYY/MM/DD? 1902/01/01
End Date in YYYY/MM/DD? 1902/01/02
Asking for /ndn/colostate.edu/netsec/pr_1902/01/01/00, Saving to pr_1902_01_01.tmp.nc
Time for pr_1902_01_01.tmp.nc 0.706118106842=
Asking for /ndn/colostate.edu/netsec/pr_1902/01/02/00, Saving to pr_1902_01_02.tmp.nc
Time for pr_1902_01_02.tmp.nc 4.52634000778=
Joining files..
Concat + write time 0.387982130051
Wrote to pr_1902_1_1_1902_1_2.nc
aldenc@rsrchr:~$
```

c. Repeat the above command. Compare the times between the two. Notice how much faster the second data retrieval is.

Finish

If you are not planning to do the GENI Desktop part of this experiment later, then go ahead and delete your resources to free them up for other users.

```
$ omni -a AM_NICKNAME deletesliver SLICENAME
```

You have now completed the exercise!

Lab One: Visualizing Your Content Centric Networking (CCN) Application Using the GENI Desktop

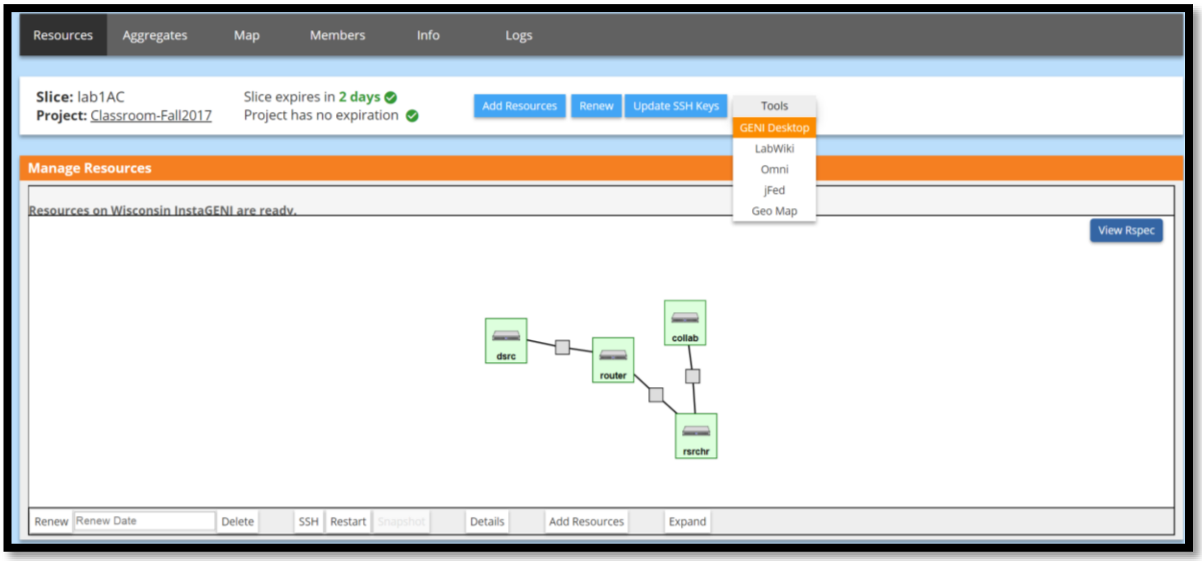
Adapted from:
http://groups.geni.net/geni/wiki/GENIExperimenter/Tutorials/GettingStarted_PartII_ccn/Procedure/Execute/GEMINI

Overview

This is an optional part of the Omni CCN lab. You should have completed that set up before beginning this section. This will use GENI Desktop to allow you to visualize what is happening in the data transfers.

Setup

1. Navigate to your slice, select "Tools", and in the dropdown menu, select "GENI Desktop".

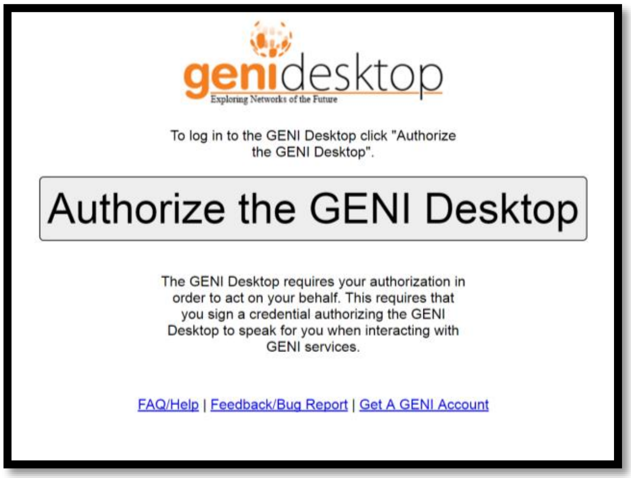


Slice: lab1AC
Project: Classroom-Fall2017

Slice expires in 2 days
Project has no expiration

Resources on Wisconsin InstaGENI are ready.

2. You will be asked to authorize the GENI Desktop, click to do so.



genidesktop
Exploring Networks of the Future

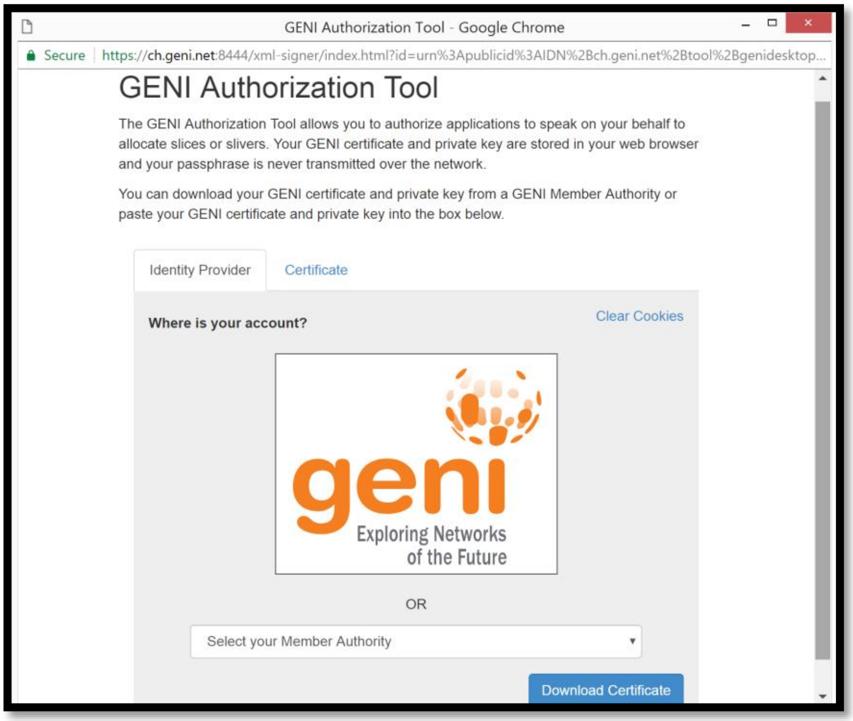
To log in to the GENI Desktop click "Authorize the GENI Desktop".

Authorize the GENI Desktop

The GENI Desktop requires your authorization in order to act on your behalf. This requires that you sign a credential authorizing the GENI Desktop to speak for you when interacting with GENI services.

FAQ/Help | Feedback/Bug Report | Get A GENI Account

3. The GENI Authorization Tool will pop up, select GENI as your account identity provider.



GENI Authorization Tool - Google Chrome

Secure | https://ch.geni.net:8444/xml-signer/index.html?id=urn%3Apublicid%3AIDN%2Bch.geni.net%2Btool%2Bgenidesktop...

GENI Authorization Tool

The GENI Authorization Tool allows you to authorize applications to speak on your behalf to allocate slices or slivers. Your GENI certificate and private key are stored in your web browser and your passphrase is never transmitted over the network.

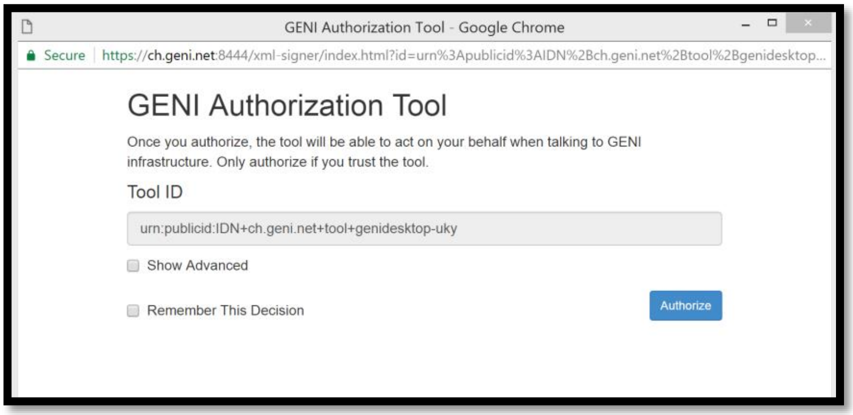
You can download your GENI certificate and private key from a GENI Member Authority or paste your GENI certificate and private key into the box below.

Where is your account?

OR

Select your Member Authority

4. Then click “Authorize” to finish authorization.



GENI Authorization Tool - Google Chrome

Secure | https://ch.geni.net:8444/xml-signer/index.html?id=urn%3Apublicid%3AIDN%2Bch.geni.net%2Btool%2Bgenidesktop...

GENI Authorization Tool

Once you authorize, the tool will be able to act on your behalf when talking to GENI infrastructure. Only authorize if you trust the tool.

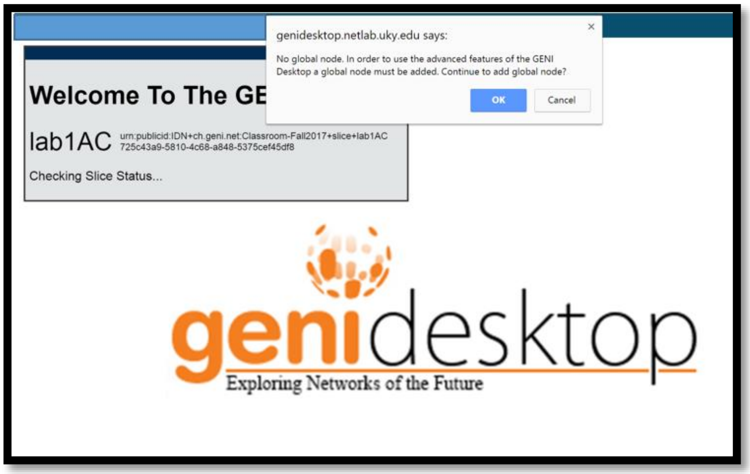
Tool ID

urn:publicid:IDN+ch.geni.net+tool+genidesktop-uky

□ Show Advanced

□ Remember This Decision

5. Now click “OK” to allow GENI Desktop to add a global node.



genidesktop.netlab.uky.edu says:

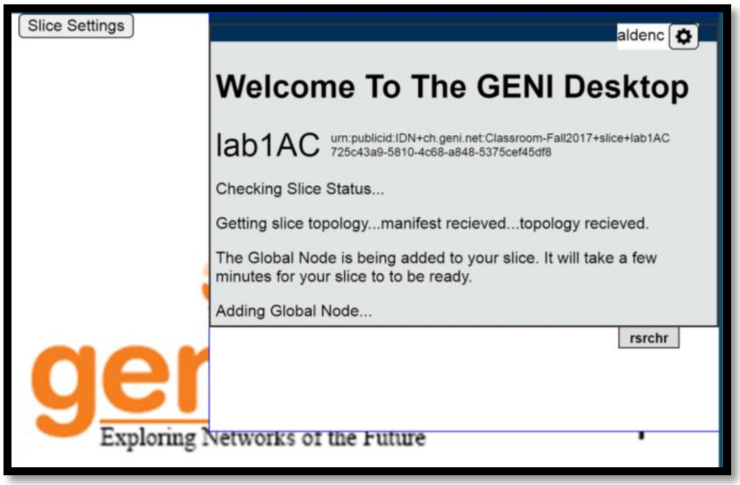
No global node. In order to use the advanced features of the GENI Desktop a global node must be added. Continue to add global node?

Welcome To The GE

lab1AC urn:publicid:IDN+ch.geni.net:Classroom-Fall2017+slice+lab1AC 725c43a9-5810-4c68-a848-5375cef45df8

Checking Slice Status...

6. Then the desktop will get your slice ready, initialized, and instrumentized. This may take 5-20 minutes for this experiment or longer for a larger more complicated experiment.



Welcome To The GENI Desktop

lab1AC urn:publicid:IDN+ch.geni.net:Classroom-Fall2017+slice+lab1AC
725c43a9-5810-4c68-a848-5375cef45df8

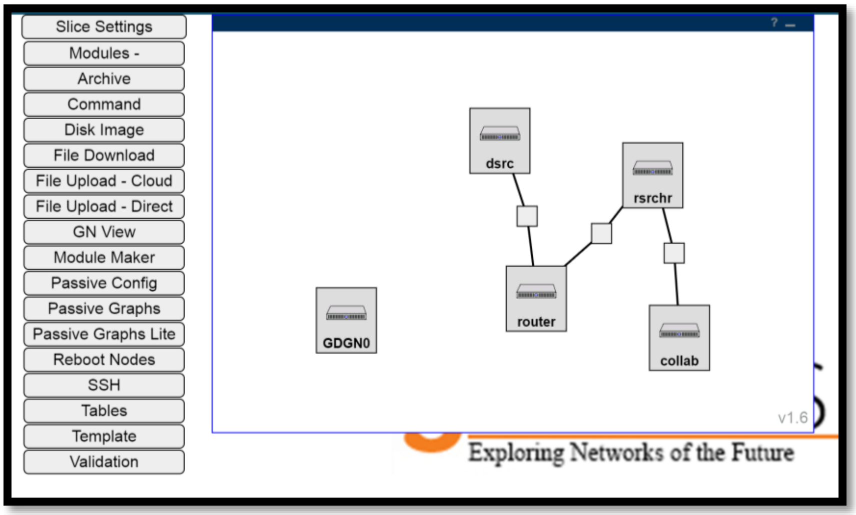
Checking Slice Status...

Getting slice topology...manifest recieved...topology recieved.

The Global Node is being added to your slice. It will take a few minutes for your slice to to be ready.

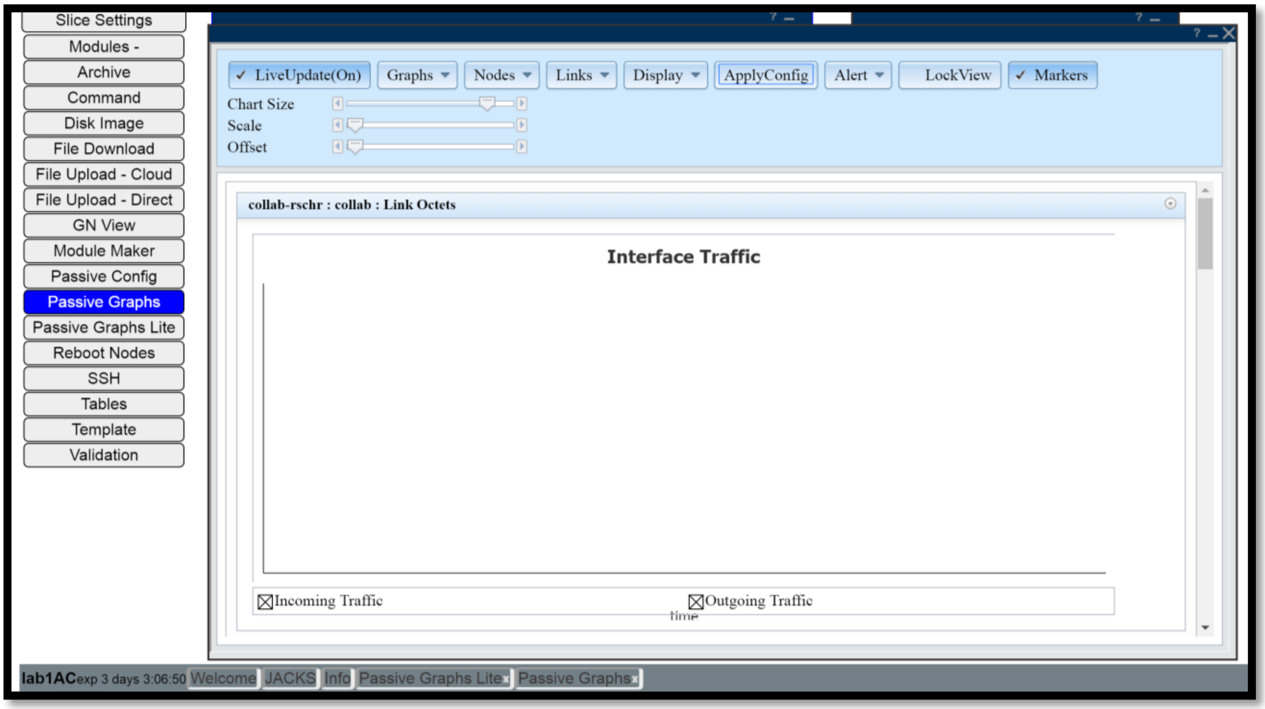
Adding Global Node...

7. Once your slice is ready, it will look as so:

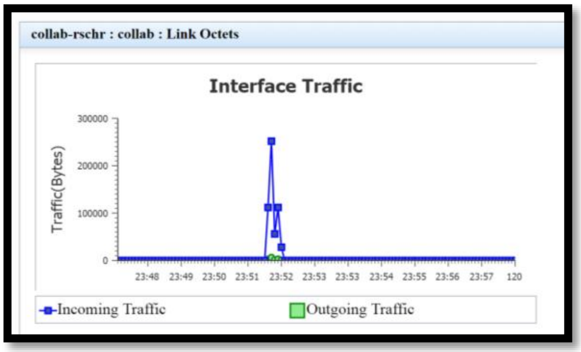


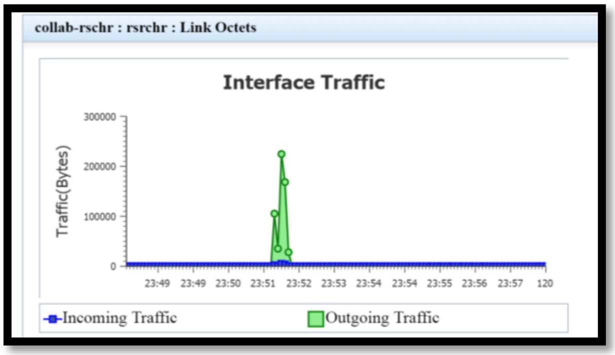
Visualization

1. Select “Passive Graphs”



2. In the menus, under Graphs, select “linkoctets” only. We will use this to see the byte traveling along the links. Under links, select all three links. Turn on LiveUpdate and click ApplyConfig.
3. Go back to your SSH window and rerun the data grabbing experience. Watch the different links; on the first run of the data you should see activity on the links, but not on the second. ***Why?***





For instance, in these images, the spike is the first call for the data. A second request for the same data was made, but no activity appeared.

Finish

Remember to delete your resources when finished!