

# Lab Zero: A First Experiment Using GENI and Jacks Tools

Adapted from: <http://tinyurl.com/geni-labzero>

## Overview

---

This is a simple starting experiment on GENI.

### Contents

- [Design and Setup](#)
- [Execution](#)
- [Finishing](#)

Over this exercise, you will:

1. Complete a full GENI experiment.
2. Become familiar with the terms: experiment, project, slice, resource, and aggregate.
3. Use the GENI experimenter portal, Jacks, and an SSH client.

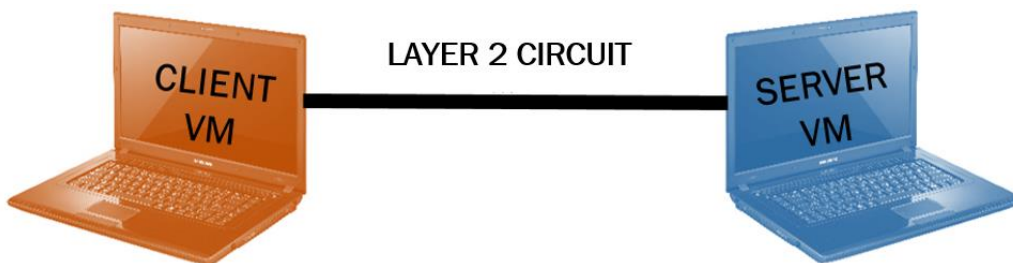
Specifically, you will:

- Use your GENI account for the first time and do some one time setup
  - Create your own ssh keys
  - Download and setup your ssh keys
- Learn how to reserve, login to, and release resources in GENI
  - Create and renew a slice
  - Generate and reserve your own topology of GENI resources using Jacks
  - Login to your resources using your own ssh keys
  - Delete resources
- Use these resources to do a simple experiment
  - Examine the difference between control and data plane interfaces on each node
  - Use `ping` and `ifconfig`

## Design and Setup

---

### 1. Design the Experiment



You will reserve two virtual machines at a single InstaGENI aggregate and connect them.

### 2. Establish the Environment

- a. Log into the GENI portal using the username/password that you did in the pre-work.

- b. You will need to generate and download an SSH keypair. Navigate to your name in the top right hand corner and click *SSH Keys* from the menu.



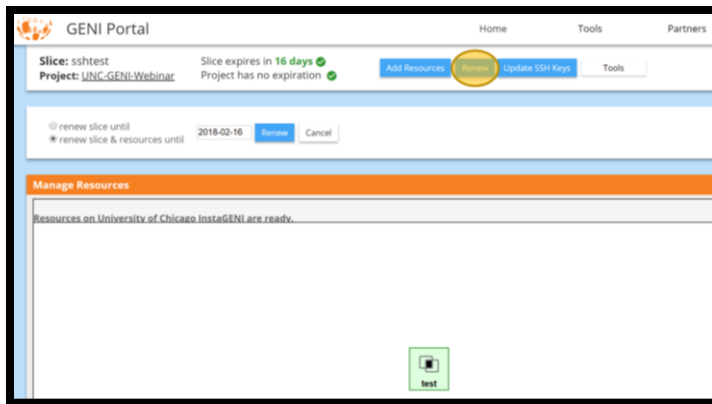
- c. You will need your own key set. You can have the GENI portal create a set for you or upload your own set if you so desire.  
*Recommended* – Have the GENI portal create a set of keys for you. On the SSH Keys page, select the **Generate and Download an SSH Keypair** button. Then enter a chosen passphrase (you will need this later) and click **Generate SSH Private Key**.  
*Alternative* – You may instead upload a public key if you already have a key set you would like to use. Make sure you have the corresponding private key and passphrase.
- d. Now you need to download the private key to your personal computer.  
*PuTTY Users* – Click **Download PuTTY Key** and save the key file.  
*Other* – Click **Download Private Key** and save the key file. If using a terminal, change the key permissions as you did in the pre-work using the following command: `chmod 700 <key>`. Optionally add the key to your ssh agent.

### 3. Obtain Resources

- a. Create a slice.  
i. Go to the *Home* tab and select **Slices**. Click **+ New Slice**.

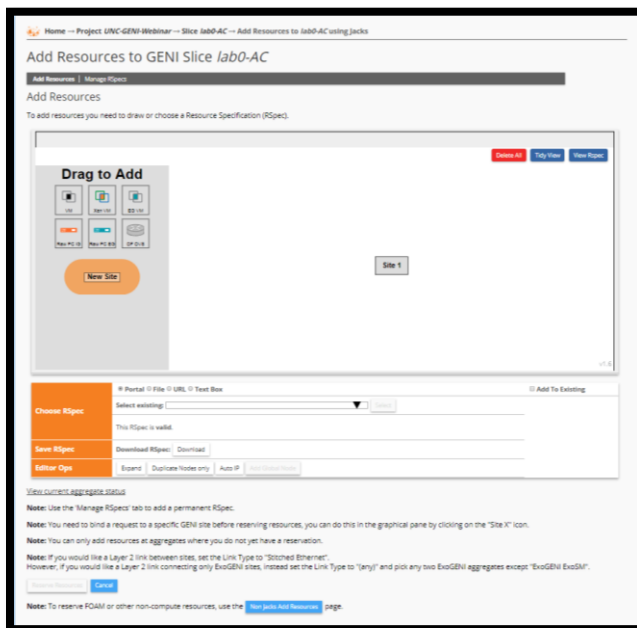


- ii. Make sure the correct project is selected (*UNC-Classroom-Spring2018*). As a slice name, use the slice name from the worksheet (*lab0\_<Initials>*). You can leave the description empty. Click **Create Slice**.
- b. (Optional) Renew your slice. Notice that slices and the resources within them have distinct expiration times. Make sure that you renew both if desired.
- You can renew your slice and resources both from the Home page and from the Slice page. From the Home page, you can only renew slices once they have resources and only for a predetermined amount of time, i.e. 7 days.
  - To renew on the Slice page, select **Renew** and select either to renew just the slice or both the resources and the slice.
  - Click on the date to the right. In the calendar that pops up, select a date in the near future. Press **Renew** to renew the slice.

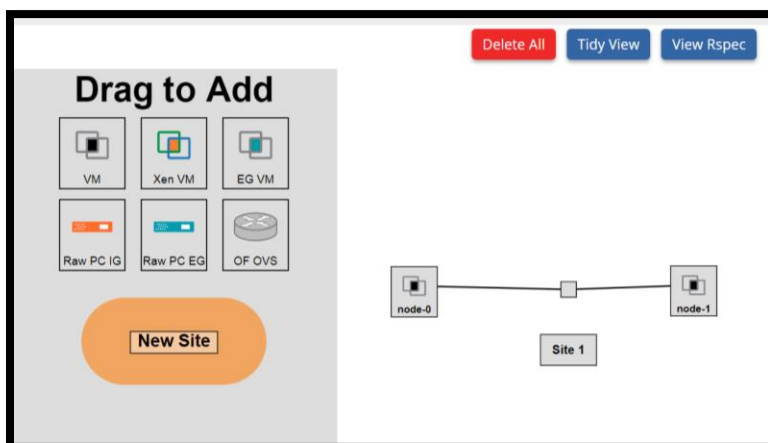


c. Reserve Two Virtual Machines at One Aggregate

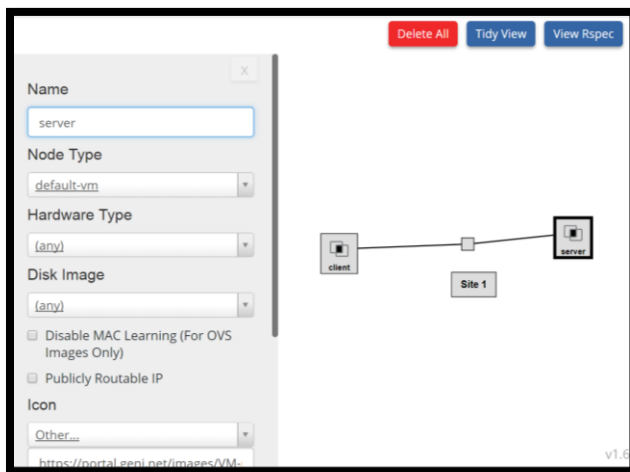
- i. You will use *Jacks*, a tool integrated with the Portal that allows you to draw topologies and then reserve corresponding resources.
- ii. In the middle top of the Slice page, next to the **Renew** button, click **Add Resources** and wait for Jacks to load.



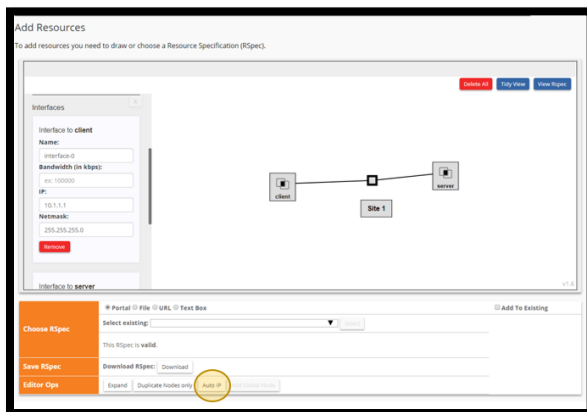
- iii. Drag two of the VMs to the canvas. The VM icon represents a generic default-vm which the aggregate has a well-known default for (for InstaGENI it is a Xen VM and for ExoGENI it is an ExoSmall).
- iv. Click near one of the VM boxes on the canvas and then click and drag towards the other VM. Release when you reach the other VM.



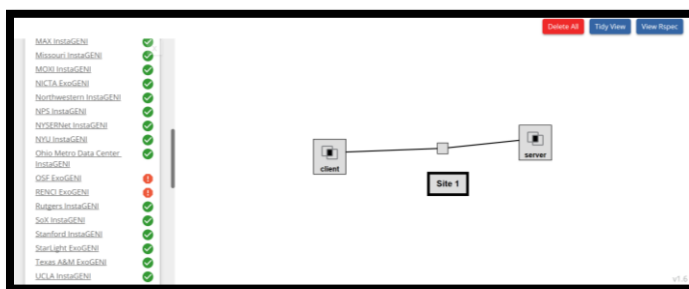
- v. Click on the VM to edit the name. Name one VM **Client** and name the other **Server**.



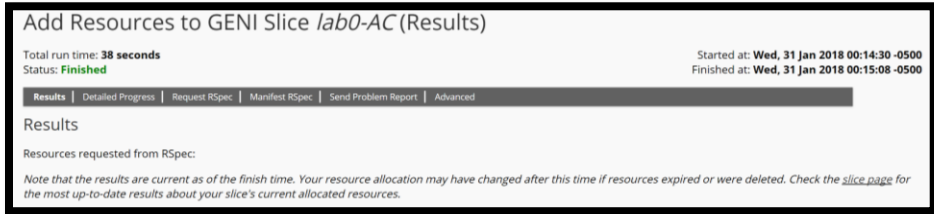
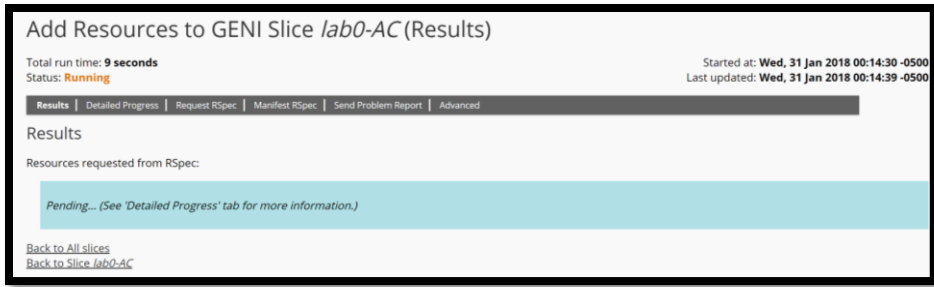
- vi. Assign IP addresses to the interfaces. You can do this by clicking on the empty box in the middle of the link, or more easily by clicking the **Auto IP** button in the **Editor Ops** section at the bottom of the page.



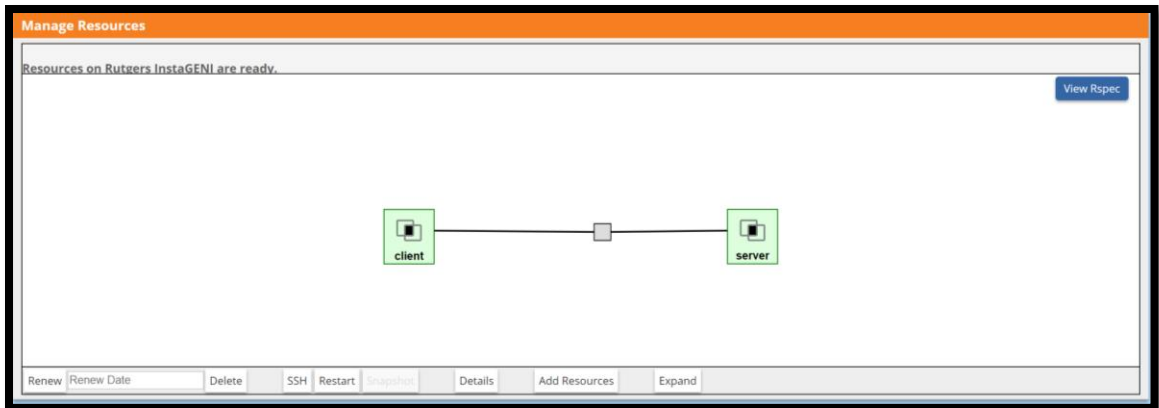
- vii. Next, you need to select an aggregate. Click the Site 1 box and select an *InstaGENI* aggregate from the list. Note that some aggregates may be down (signified with a red exclamation point). Choose one that has a green checkmark. Sometimes sites are busy, if you run into a problem you can select another site.



- viii. After selecting an aggregate, click **Reserve Resources** at the bottom of the page.
- d. Wait for the resources to be ready for use.
- i. After clicking **Reserve Resources**, a status page will open. Wait for the status to change to **Finished**. This means the resources have been reserved, although the nodes may need more time to load.



- ii. Navigate to the **Slice Page**. The nodes will turn green when they are ready for use.

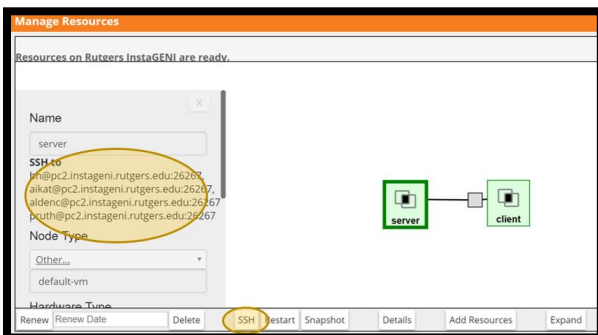


## Execution

### 4. Configure and Initialize

- a. Find the node login information.

There are a number of ways to get the SSH information for each node. On the page where the resources were being reserved, there is a list of the SSH information for each node. Once in the Jacks editor, you can click on a node to see the SSH information, or click the SSH button at the bottom of the editor. Note the hostname and port number. Record these in your worksheet.



- b. Log into both of the nodes. You will do this in the same way you logged into the test node in the pre-work, except using the node information you recorded in step (4.a) and the SSH keys you generated or uploaded. You will need to do this procedure twice, once for the Client node and once for the Server node. It is likely that the only difference between the two is the port number.

Command Line Terminal: Start SSH by typing:

```
$ ssh -i <private key location> <username>@<hostname> -p <port number>
```

Or if you added your keys to your ssh-agent:

```
$ ssh <username>@<hostname> -p <port number>
```

Enter your passphrase when prompted (you should not be prompted for a password).

PutTY: Open PuTTY, enter <username>@<hostname> in the Host Name field and the port number in the Port field. Select SSH as the connection type, navigate to Connection->SSH->Auth and load in your downloaded private PuTTY style key. Open the SSH connection and enter your passphrase when asked.



## 5. Execute Experiment

### a. Send IP Traffic

The first simple experiment that we will run is to verify the IP connectivity between our hosts.

Check the interfaces of your nodes. In each terminal type: `sudo ifconfig`

You should see at least two interfaces:

- The **control interface** is the interface you use to access the node, e.g. ssh into your host. The control interface is mainly used for control traffic, i.e. traffic for controlling the node and the experiment (likely something like 172.17.1.9).
- The **data interface** is the interface that is used for sending experimental traffic; this is the interface that connects to the other hosts of your experiment through GENI. The links between these interfaces are the ones that allow you to run non-IP experiments. **The data interface is the one that has an IP address and mask that match what you configured before you reserved your resources** (likely something like 10.10.1.1).

Fill in the worksheet, noting the name and IP address of the control and data interfaces for each node. Note that the command prompt on each node should say either “client” or “server”

- From the client, ping the server **data plane interface**. From the terminal window that is logged into the client, type:

```
$ ping <server data IP addr> -c 5
```

Example: `ping 10.10.1.2 -c 5`

- Now, ping the server **control plane interface**. From the terminal window that is logged in to the client, type:

```
$ ping <server control IP addr> -c 5
```

Example: `ping 172.17.1.9 -c 5`

### b. Install and use *iperf*

- i. Install the *iperf* software on both nodes: `sudo apt-get install iperf`
- ii. Start an *iperf* server on the server node: `iperf -s`
- iii. Run an *iperf* client via the data plane: `iperf -c <server data IP addr>`  
Example: `iperf -c 10.10.1.2`  
**What is the bandwidth of this link and why?**
- iv. Run an *iperf* client via the control plane:  

```
$ iperf -c <server control IP addr>
```

  
Example: `iperf -c 172.17.2.2`  
**What is the bandwidth of this link and why?**
- v. Type CTRL-C on the server node to stop the *iperf* server.
- c. Bring down the server's data interface
  - i. From the client node, start pinging the server **data plane interface**:  

```
$ ping <server data IP addr>
```
  - ii. From the server node, bring down the **data plane interface** (be careful not to disable the control interface):  

```
$ sudo ifconfig <server data interface name> down
```

  
**After you bring down the data interface, the pings should indicate that the destination is unreachable, why is this?**
- d. Bring down the server's control interface
  - i. From the client node, start pinging the server **control plane interface**:  

```
$ ping <server control IP addr>
```
  - ii. From the server node, bring down the **control plane interface**:  

```
$ sudo ifconfig <server control interface name> down
```

  
**Your ssh session should immediately hang, why? After you bring down the interface, your pings should indicate the destination is unreachable. Why?**

## 6. Analyze Experiment

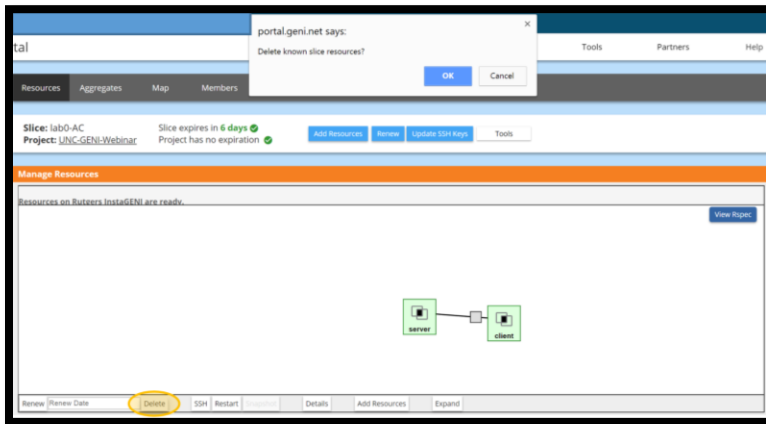
- a. In a typical experiment, this is where you would analyze your results.
  - b. Logout of any nodes you are still logged into (type `exit` into the terminal).
-

# Finish

---

## 7. Teardown Experiment

- Delete your resources so that they are available for other people. Press the **Delete** button at the bottom and then **OK** to the popup window.



## 8. Archive Experiment

Save any needed data (likely none for this experiment).