

# Understanding the GENI AM API through a simple experiment

Understanding the GENI AM API through a simple experiment  
Overview  
What you will learn  
Experiment Topology and Setup  
Procedure

## Overview

Understand how resources are described and reserved in GENI. Learn how to customize those resources using standard mechanisms. Use multiple tools to act on a slice and reserve resources.

While doing this experiment, use and understand the following terms: resource specification (RSpec), sliver, and aggregate manager API (AM API)

## What you will learn

At the end of this exercise, you will be able to:

- Use and understand the following terms:
  - Resource Specification (aka RSpec)
  - sliver
  - Aggregate Manager API (AM API)
- Customize the resources they reserve using standard mechanisms:
  - understand that a resource specification (RSpec) is an XML file
  - understand that resources are reserved using the aggregate manager API (AM API)
  - understand how to load a topology using an RSpec
  - have edited an RSpec using a tool (Flack) as well as by hand
  - have reserved and deleted resources using a command line tool (omni)
  - know how to specify the image to load on a compute node
  - know how to have software automatically loaded and run on a compute node
  - have run a simple layer 2 experiment
- Use multiple tools to act on a slice and reserve resources
  - understand that all tools use common mechanisms to describe resources (RSspecs) and to reserve resources (the AM API)
- Understand the strengths of different types of GENI racks

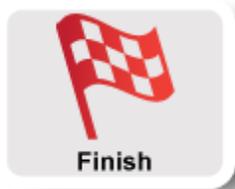
## Experiment Topology and Setup



Use the Omni GENI client to reserve two VMs connected by a Layer 2 circuit.

Then arrange for automatic installation and execution of software on these nodes (performing basic traffic measurement tests), and manually log in to inspect the results.

## Procedure



- **Part I: Design/Setup**

- Step 1: Design Experiment
- Step 2: Establish Management Environment
- Step 3: Obtain Resources

- **Part II: Execute**

- Step 4: Configure and Initialize Services
- Step 5: Execute Experiment
- Step 6: Analyze and Visualize Experiment

- **Part III: Finish**

- Step 7: Teardown Experiment
  - Step 8: Archive Experiment
-

# Understanding the AM API



## 1. Design the Experiment

1. In today's experiment you will use resources at the aggregate listed on the worksheet.

## 2. Establish the Environment

### 2.1 Pre-work: Create a GENI account

1. Go to <https://portal.geni.net> and press the **Use GENI** button
2. From the Drop Down menu select your institution. If you got an account through the GENI Identity Provider, please select **GENI Project Office**.



Start typing the name of your institution and see the list become smaller.

## GENI

*Please sign in using your account at one of our partners:*

Enter your college, university, or organization's name

  
[GENI Project Office in a list](#) [Get Help](#)

*No account? [Contact GENI Help](#)*



3. You will be transferred to the Login Page of your institution. Fill in your username and password.
4. Complete the form that appears after you have successfully logged in and press **Continue**.
5. You will be transferred to an **Activation Page**.

**Figure 2-1** Logging into the GENI Experimenter Portal.

Make sure both checkboxes are checked and then press **Activate**.

Congratulations, you have successfully created a GENI account.

## 2.2 Pre-work: Project lead adds you to a project

In order to use the portal to reserve resources, you must join a project.

1. The tutorial organizers should have added you to the project for this tutorial. On the home page, you should see that you are a member of at least one project.



If you are not doing this exercise as part of an in-person tutorial please sign up for a [GENI account](#) and join a project.

## 2.3 Pre-work: Ensure SSH keys are setup

Access to compute resources in GENI is provided through ssh key pairs and thus the portal needs a public key to upload to compute resources. For the purposes of the tutorial we will have the portal create an SSH key pair for you. (However, if you prefer to use your personal public key you can choose to upload it.) Verify that you have at least one public key associated with your account. To do that, after you login to the portal check under your Profile, under the `SSH keys` tab. If you do not have SSH keys associated yet, please follow [these instructions](#).

## 2.4 Configure Omni

1. Login to the [GENI Portal](#)
2. Click on the `Profile tab` link on the top of your screen. Then click on the `Configure omni` tab under `PROFILE`.



3. Click on the `Download your omni data` button under step 2.

### Option 1: Automatic omni configuration

To configure omni, use the [omni-configure](#) script distributed with omni as described below.

1. In order to use omni or other command line tools 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.zip):

[Download your omni data](#)

3. Generate an omni\_config by running the following command in a terminal:

```
omni-configure
```

4. If this is the first time you try to access your GENI certificate you will have to generate one. Click on the `generate a certificate` link.

## 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](#).

Choose a default project to use with omni:

5. Unless you really understand how SSL certificates work, choose the simple option. Click on the `Generate Combined Certificate and Key File` button and then click on `Close`. You will be taken back to the download page with the warning. Reload the page to enable the download button.

### GENI Certificate Management

In order to use some GENI tools (like [omni](#)) you need a signed SSL user certificate.

There are two options for creating this:

1. Have it generated for you. This is the easiest option. **If in doubt, use this option.**
2. Have the SSL certificate generated for you based on a private key you keep locally. This is the most secure

#### Simple Option: Have the SSL certificate generated for you

**If in doubt, use this option.**

*An SSL certificate always has a corresponding SSL private key. This option will generate one file which contains different from your SSH private key.)*

Remember, in order to use this, you will need to have the downloaded combination certificate/private key file.

6. If you are a member of more than one project, select which project you would like to be the default one for running experiments in GENI. You can always change the project that is used by the `-r` command line option of Omni. Then click on `Download your omni data`.

## DOWNLOAD OMNI CONFIGURATION DATA

Use the omni configuration data downloaded from this page, to automatically [configure omni](#).

Choose a default project to use with omni:

7. The bundle will be saved at `~/Downloads/omni-bundle.zip`
8. Open a terminal window and type:

```
omni-configure
```

The cert and key files you need will be installed in the appropriate folders.

## 3. Obtain Resources

Now that you are a member of a project, you can create a slice and reserve resources.

### 3.1 Create a slice

1. Go to the *Home* tab.
2. Press the **Create Slice** button for this project.
3. As a slice name use the slice name on your worksheet, you can leave the description empty and press

[Home](#) → [Project tutorial](#) → [Create Slice in project tutorial](#)

## CREATE NEW SLICE

A GENI slice is a container for reserving and managing a set of GENI resources.

<b>Project name</b>	tutorial
<b>Slice name</b>	<input type="text" value="geni01"/> -- Required
<b>Slice description</b>	<input type="text"/>

**Note: Slice name is public**

## Create Slice **Figure 3-1** Create a new slice.

Now that you are a member of a project, you can create a slice and reserve resources.

### 3.2. Import a simple experiment

For this exercise, we will start with a prepared rspec file.

You should load this topology into Flack:

1. From the "Import" menu (see figure), select the "Import from the web" item.
2. Enter the URL corresponding to the rack type you are using and open it:



**InstaGENI**

<http://www.gpolab.bbn.com/experiment-support/UnderstandAMAPI/ig/understandAMAP>



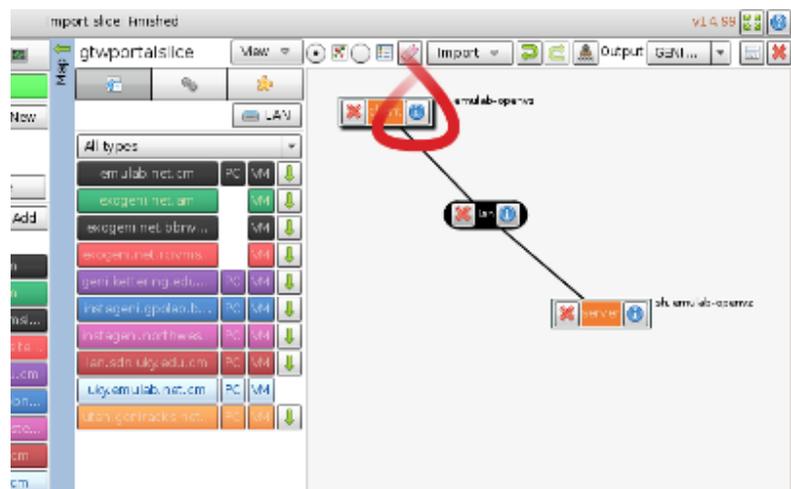
**ExoGENI**

<http://www.gpolab.bbn.com/experiment-support/UnderstandAMAPI/eg/understandAMAP>

3. Because this file contains a generic description of resources and does not specify any particular manager, Flack will report "There were resources detected without a manager selected, please select a manager you would like to use." Select "OK", and then choose the manager listed on your works list. Then select "Accept", and a network topology (which might look familiar if you have completed tutorials) should appear on the canvas.

### 3.3. Modify the experiment to automatically install and execute network test software

1. For this sample experiment, we'd like some additional software (Apache, iperf, and a couple of custom scripts) loaded onto the VMs to perform traffic measurements. In Flack, you can manipulate the details of the request for a node or link with the "i" button on the relevant component on the canvas (see figure). Do this now on the client node.



2. When you open the node information, you should see options like those in the figure. The controls available to you will vary depending on the types of the aggregate manager and the component, but

many features are widely available throughout GENI.

3. The way we will request installation of the proper software and execution of our experiment is to add *install* and *execute* services. First, select "+ Add Install Service" as shown. Flack will now prompt you for an "Archive URL" and a location for installation. For this experiment, please enter the appropriate URL:



```
http://www.gpolab.bbn.com/experiment-support/UnderstandAMAPI/ig/underAMAPI_ig.t
```



```
http://www.gpolab.bbn.com/experiment-support/UnderstandAMAPI/eg/underAMAPI_eg.t
```

and ask for this to be installed in:

```
/local
```

4. Similarly, please use the "+ Add Execute Service" button to request that the appropriate command:



```
sudo /local/install-script.sh
```



```
/local/install-client.sh
```

(client node only)

```
/local/install-server.sh
```

(server node only)

be executed using

```
sh
```



Be very careful when entering this information -- these commands will not be executed yet, so it will be some time before you will see any relevant error messages if there is a mistake here.

5. Once both the "install" and "execute" services are specified, please choose "Apply". Then repeat the procedure to add the same "install" and "execute" services on the server node.

6. Also on the server node, click on the "link" tab and select "Routable Control IP" as shown in the figure.

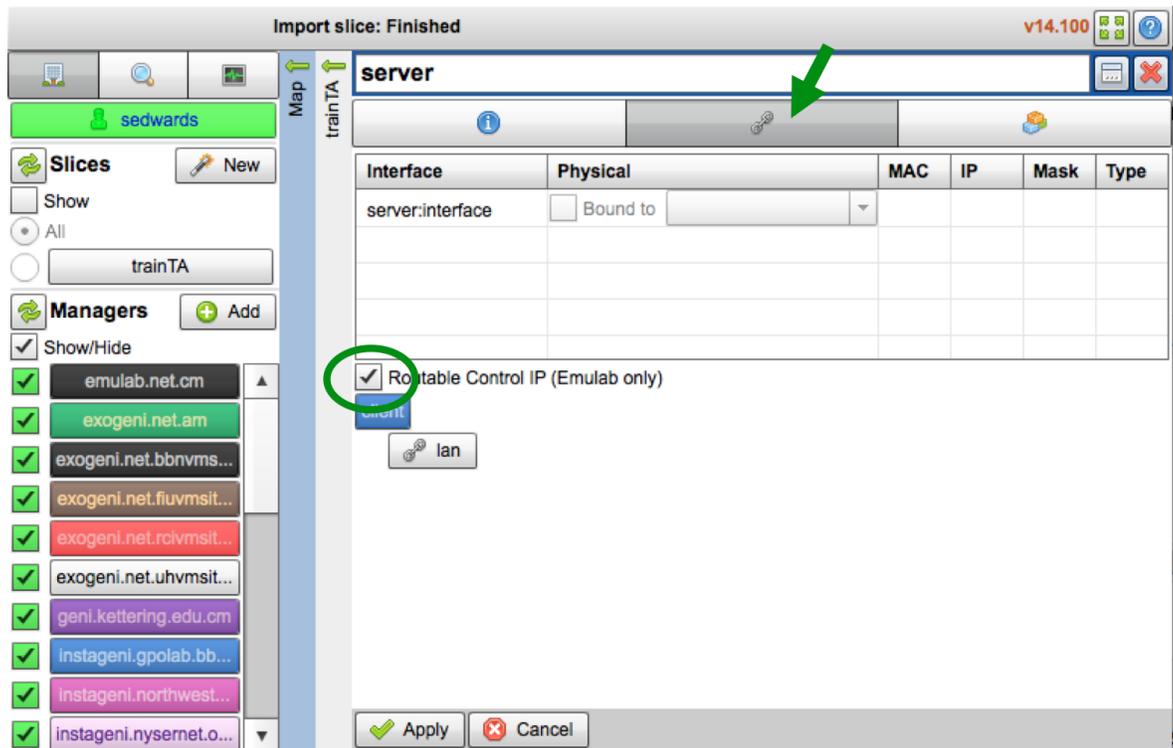
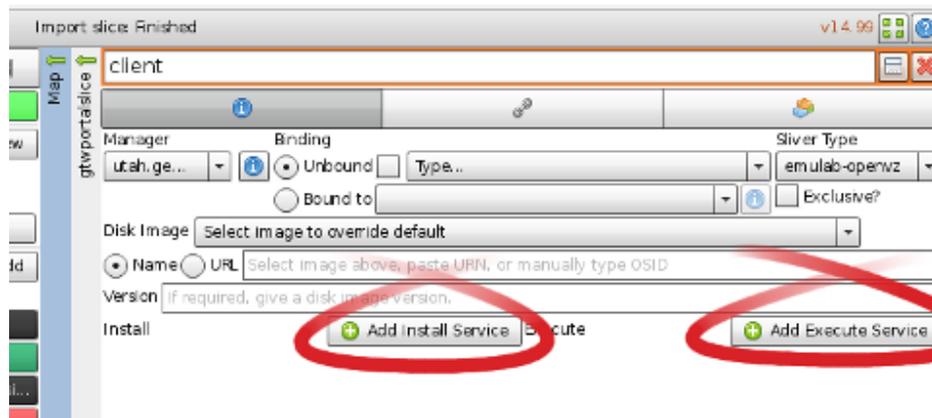


Figure 3-5 On the *Link* tab, select the "Routable Control IP" button.

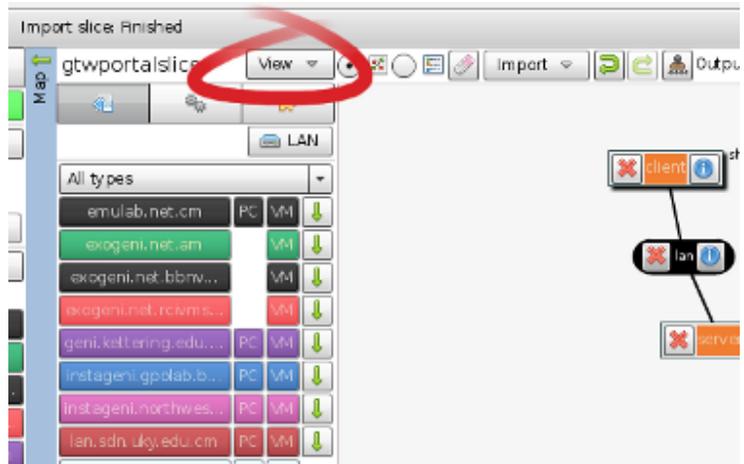
If you have time, it would be instructive to download the "underAMAPI\_\*.tar.gz" file yourself, and inspect the contents. You will see the `install-script.sh` file referred to in the execute service, and this is a good example of how you can ask for GENI components to perform tasks for you without any manual intervention. This is a very useful facility for large experiments!



### 3.4. Export the modified request RSpec

Now we will pull back some of the covers and inspect exactly what Flack 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.

1. From the "View" menu (see figure), select the "Preview request document(s)" item. This will bring up a window showing the current rspec -- please take a moment to inspect it. The XML elements under the "flack" namespace were added for Flack's internal use (containing information about the canvas layout, editing history, etc.), and are ignored by aggregate managers. 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.
2. Use the "Save to file" button (in the upper left) to make a local copy of your rspec. We'll use this in the next step to demonstrate how other client tools also use rspec files to communicate requests to aggregate managers.



### 3.5. Instantiate the new experiment using Omni

For this step, we'll change the approach a bit and switch to a new client tool: the command line Omni client. From a terminal, please enter the command:

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

where `AM_NICKNAME` is the nickname for your assigned aggregate manager and `SLICENAME` is the name of the slice you created at the portal (both of these are given on your worksheet). `RSPEC_FILE` should be replaced with the filename of the rspec you saved in step 4.

If all is well, Omni should give you a number of informational messages, such as:

```
INFO:omni:Loading config file /home/geni/.gcf/omni_config
```

It should quickly proceed to the point where it makes the request to the remote manager:

```
INFO:omni:Creating sliver(s) from rspec file /home/geni/Downloads/experiments.r
```

This step can sometimes be time-consuming, so please be patient. If it succeeds, within a few minutes Omni should report:

```
INFO:omni: Completed createsliver:
```

and your sliver is complete!

---

**Introduction**

**Next: Execute**

## Understanding the AM API



### 4 Wait for experiment setup

1. Please use the command:

```
readyToLogin --no-keys -a AM_NICKNAME SLICENAME
```

where (as before) `AM_NICKNAME` and `SLICENAME` are your aggregate manager nickname and your slice name (both found on your worksheet).

2. If it reports that the sliver is not yet ready (for example, it might say that the status is "changing"), then please wait a minute or two and try again. Once everything is complete, `readyToLogin` will give output that should look something like this:

```
...
server's geni_status is: ready (am_status:ready)
User example logs in to server using:
  ssh -p 32768 example@pc1.utah.geniracks.net
User example logs in to client using:
  ssh -p 32769 example@pc1.utah.geniracks.net
...
```



If you didn't previously complete the Flask tutorial (or are not running an `ssh` agent), then your `ssh` client might not be set up to log in with above commands. Try re-running `readyToLogin` without the `--no-keys` option, and it will give you one or more `ssh` commands to choose from (which should work, although might require the key passphrase).

### 5 Log in to client node

1. Copy and paste the `ssh` command lines directly into your terminal to log in to either of your hosts. While you're welcome to inspect either one, for the purpose of this experiment, the `client` host is the one running the `iperf` tests and collecting all the logs, so please use the `client` `ssh` command now.

You may get a warning from `ssh` complaining that the authenticity of the host cannot be established. This is just because your `ssh` client has never accessed this VM before, and so does not yet recognise its key. Say "yes", you do want to continue connecting, and you should see a shell prompt from the remote end:

```
[example@client ~]$
```

The `install` and `execute` services requested in our `rspec` have already started, and measurements are now being collected.

You can verify that things are working by entering the hostname of the server node in your browser. It should bring up a webpage of statistics from your experiment.

(In addition, you can inspect the `/local` directory on each host, and looking for the appropriate processes with a command like `ps ax`. If you do not see the proper files and processes, please double-check the `rspec` you used in the previous step.)

You can verify that things are working by entering the hostname of the server node in your browser. It should bring up a webpage of statistics from your experiment.

(In addition, you can inspect the `/local` directory on each host, and looking for the appropriate processes with a command like `ps ax`. If you do not see the proper files and processes, please double-check the `rspec` you used in the previous step.)



[Web Server Statistics](#)

[Logs from the iperf Sever](#)

**Figure 5-1** Enter the hostname of the *server* node in your browser to see statistics

2. The client machine is saving all the test results in the `/tmp/iperf-logs` directory. Files with timestamps in the names will gradually appear there (there are 100 tests overall, and it may take 20 minutes for all of them to complete if you want to wait for them).

Each log file corresponds to one test with some number of simultaneous TCP connections over the VLAN link you requested between the two hosts. Later tests gradually include more concurrent connections, so the throughput of each individual connection will decrease, but the aggregate throughput (the `[SUM]` line at the end of each file) should remain approximately consistent.

## 6. Analyze Experiment

For a real experiment, of course, this step would be the most important and collection, analysis and archival of the results would be critical, but for now, play around as necessary to satisfy your curiosity and then continue.

---

**Introduction**

**Next: Finish**

## Understanding the AM API



## 7 Cleaning up

As in previous tutorials, it is always good practice to clean up slivers right away so that resources can be released and reused by other experimenters. While other tools (such as Flack and the GENI Portal) can also be used to clean up, no matter which software was used to establish the sliver in the first place, for this example we will demonstrate the clean up procedure using Omni.

The command to use is:

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

where once again `AM_NICKNAME` is the aggregate manager nickname and `SLICENAME` is the name of your slice (both found on your worksheet).

A minute or so later, Omni should respond with:

```
INFO:omni: Completed deletesliver:
```

and some details of exactly what was deleted.

## 8. Archive Experiment

Congratulations! You have completed the exercise.

---

## Introduction