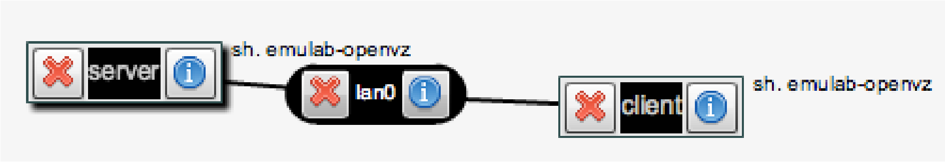
# Introduction to Omni Tutorial

## Experiment Description

In this tutorial we are going to use Omni which is a command line tool for reserving GENI resources. We are going to use a very simple topology that has a server and a client connected by only one link. At install time we are going to install at the server a web and an iperf servers, while the client will issue `http` and `iperf` requests. We will be able to see the performance on the server on a web page.



## Tutorial Instructions

- Part I: Design/Setup
    - Establish Management Environment: Setup your GENI account and join a GENI Project
    - Obtain Resources: Create a slice and reserve resources

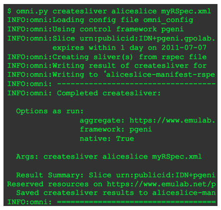
- Part II: Execute
    - Execute Experiment: Login to nodes and test connectivity
    - View Results: See Statistics

- Part III: FInish
    - Teardown Experiment: Delete Resources

# Introduction To Omni Tutorial

Design/Setup Execute Finish

## Configure and Initialize Services:Configure Omni

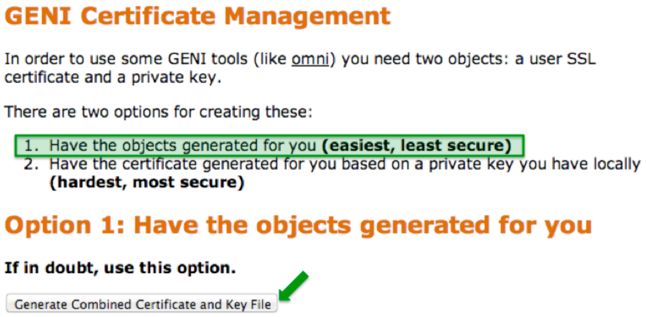


```
$ omni.py createsliver aliceslice myRSpec.xml
INFO:omni:Loading config file omni_config
INFO:omni:Using control framework pgeni
INFO:omni:Slice urn:publicid:IDN+pgeni.gpolab.
          expires within 1 day on 2011-07-07
INFO:omni:Creating sliver(s) from rspec file
INFO:omni:Writing result of createsliver for
INFO:omni:Writing to 'aliceslice-manifest-rspe
INFO:omni: ------------------------------------
INFO:omni: Completed createsliver:

  Options as run:
              aggregate: https://www.emulab.
              framework: pgeni
              native: True

  Args: createsliver aliceslice myRSpec.xml

  Result Summary: Slice urn:publicid:IDN+pgeni
Reserved resources on https://www.emulab.net/p
  Saved createsliver results to aliceslice-man
INFO:omni: ====================================
```

Omni is a tool that will help you reserve resources in GENI, using your GENI credentials. The first step is to load into Omni your information. Omni uses a configuration file, and there is a script that will automatically generate the file with some default options and place it in the default place of `~/.gcf/`.

Follow these steps:

1. In the GENI Portal, go to the `Profile` Tab.
2. Scroll all the way to the end of the page, to the section titled "Configure omni"
3. Click on the "customized configuration data" link
4. If this is the **first time** you do this:
    a. Press the **generate an SSL Cert** link
    b. Make sure you choose the first option
5. In the next page, make sure you choose as your project, the project you just joined and click the "Download omni bundle" button, and select to save it.
6. Open a terminal window
7. Run:
    - `omni-configure.py -f portal`



GENI Certificate Management

In order to use some GENI tools (like omni) you need two objects: a user SSL certificate and a private key.

There are two options for creating these:

1. Have the objects generated for you **(easiest, least secure)**
2. Have the certificate generated for you based on a private key you have locally **(hardest, most secure)**

Option 1: Have the objects generated for you

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

The output should look like:

```
geni@GEC16-Tutorials:~$ omni-configure.py -f portal
{'portal_bundle': '~/Downloads/omni-bundle.zip', 'verbose': False, 'cert': '
configfile': '~/.gcf/omni_config', 'plkey': '~/.ssh/geni_pl_key', 'prkey': '
INFO:omniconfig:Using configfile: /home/geni/.gcf/omni_config
INFO:omniconfig:Using portal bundle /home/geni/Downloads/omni-bundle.zip
INFO:omniconfig:Private key stored at: /home/geni/.ssh/geni_key
INFO:omniconfig:Wrote omni configuration file at: /home/geni/.gcf/omni_confi
geni@GEC16-Tutorials:~$ omni-configure.py -f portal
```

### Look around the omni_config file

Open the file `~/.gcf/omni_config` using either vim or emacs. Close to the top of the file you will see two parameters called `default_cf` and `users`. Your username should be at least listed in the user section. Look for the sections in the file that are named `[pgch]` and `[<username>]`.

In the `[<username>]` section, the information need for logging-in to reserved compute resources are provided. It includes your unique user URN and a public key that would be uploaded to the hosts that you reserve.

In the `[pgch]` section you configure Omni to use your personal information. The cert and the key attribute point to files that we have manually downloaded from the portal.

Another interesting section to look at is the `[aggregate-nicknames]` sections. Flack already knows the URL for all the AMs and present you a list of AMs to choose from using a short, descriptive name. In Omni a user is required to pass the URL for each call to the GENI AM API. In this section the user gets a chance to provide short descriptive names to the URLs that are easier to memorize and use.

### Test Omni setup

In order to test that our configuration is correct, you can issue a `getversion` command. For this step we have used ProtoGENI's AM, but you can use any Aggregate Manager(AM).

i. In the terminal type:

```
omni.py getversion -a ig-gpo
```

In GENI each AM can be contacted using a specific URL. In order to make it easier to talk to AMs, Omni uses nicknames; i.e. it maps each URL into an easy to remember nickname. Nicknames are defined in each user's omni_config file.

For the rest of the tutorial we are going to use aggregate nicknames.

## Obtain Resources

In this step we are going to setup a sample GENI experiment, in order to get familiarized with most of the Omni commands.

### Create and renew a slice

The first thing to do when preparing to run a GENI experiment is to create a slice.

i. To create a slice type, use as `<slicename>` omniXXX, where XXX are your initials. :

```
  omni.py createslice <slicename>
```

ii. To verify that your slice has been created and in general to keep track of your slices, use the `listmyslices`. You can figure out your username under the `Profile` tab. Type:

```
  omni.py listmyslices <username>
```

iii. Renew your slice. To extend the lifetime of your slice type:

```
  omni.py renewslice <slicename> YYYYMMDD
```

### Reserve resources, a.k.a create a sliver

The next step in a GENI experiment is to actually reserve resources. For this experiment we are going to use the Aggregate manager of ProtoGENI in Utah.

i. In Omni in order to see what each AM offers you can use the `listresources` command.

Type:

```
omni.py listresources -a pg-utah -o
```

The `-o` option will save the output to a file. The filename is chosen by Omni and printed as part of the output. The output will look like :

```
geni@GEC16-Tutorials:~$ omni.py listresources -a pg-utah -o
INFO:omni:Loading config file /home/geni/.gcf/omni_config
INFO:omni:Using control framework portal
INFO:omni:Substituting AM nickname pg-utah with URL https://www.emulab.
INFO:omni:Saving output to a file.
INFO:omni:Substituting AM nickname pg-utah with URL https://www.emulab.
INFO:omni:Listed advertised resources at 1 out of 1 possible aggregates
INFO:omni:Substituting AM nickname pg-utah with URL https://www.emulab.
INFO:omni: (PG log url - look here for details on any failures: https:/
INFO:omni:Writing to 'rspec-www-emulab-net-protogeniv2.xml'
INFO:omni: ---------------------------------------------------------------
INFO:omni: Completed listresources:

  Options as run:
                aggregate: ['pg-utah']
                framework: portal
                output: True
                project: SampleClass

  Args: listresources

  Result Summary: Queried resources from 1 of 1 aggregate(s).
Wrote rspecs from 1 aggregate(s) to 1 file(s)
Saved listresources RSpec at 'unspecified_AM_URN' (url 'https://www.emu
INFO:omni: ===============================================================
```

In the last line of the output Omni will tell you the name of the file that output is saved at. In the example above this would be `rspec-www-emulab-net-protogeniv2.xml`. Open the file that Omni saved and just take a look to see how an advertisement RSpec looks like. In order to see only available resources type

```
omni.py listresources -a pg-utah --available -o
```

ii. To be able to reserve resources you will need to craft a request rspec. For this example we have created the rspec for you, use this rspec URL :

http://www.gpolab.bbn.com/experiment-support/HelloGENI/hellogeni.rspec . Type :

```
omni.py createsliver -a pg-utah <slicename> <rspec_url>
```

i. Look at your reserved resources. Type:

```
omni.py listresources -a pg-utah <slicename>
```

ii. Extend the lifetime of your sliver:

```
omni.py renewsliver -a pg-utah <slicename> YYYYMMDD
```

iii. Check the status of your resources. Type:

```
omni.py sliverstatus -a pg-utah <slicename>
```

The `sliverstatus` command reports the status of your overall GENI slice. When the status is ready we are ready to continue to the next step.

## Next: Run Experiment

# Introduction To Omni Tutorial

Design/Setup Execute Finish

## Execute Experiment: Login to nodes and test connectivity

Now that resources are ready we can start running our experiment. We will first start by logging in to our reserved nodes.

### Logging Into the nodes -- Omni Scripting

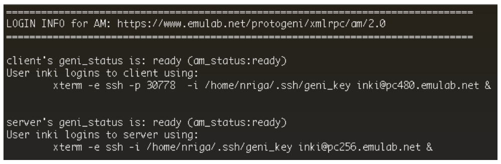
Depending on the Aggregate the login information are either part of the `sliverstatus` call, or as part of the manifest rspec, i.e. returned from the `listresources` call. Omni comes with a script that makes it easy to gather all this information.

i. In the terminal run:

```
readyToLogin.py -a pg-utah <slicename>
```

The script will return the actual command that you would need to use for logging in.

ii. Login only to the client host by copy-paste the command from the output of the script.



```
================================================================================
LOGIN INFO for AM: https://www.emulab.net/protogeni/xmlrpc/am/2.0
================================================================================

client's geni_status is: ready (am_status:ready)
User inki logins to client using:
        xterm -e ssh -p 30778  -i /home/nriga/.ssh/geni_key inki@pc480.emulab.net &


server's geni_status is: ready (am_status:ready)
User inki logins to server using:
        xterm -e ssh -i /home/nriga/.ssh/geni_key inki@pc256.emulab.net &
```

### Ping between your hosts

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

i. From the client ping the server. From the xterm that is logged in to the client type :

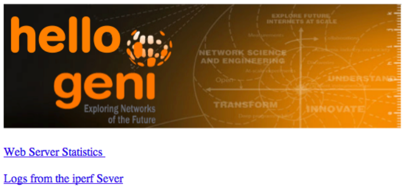
```
    ping server -c 5
```

**Note**: You can use the name that is assigned to the host to directly ping the host, you can also ping using the IP of the data interface of the node. An example output should look like :

```
[[inki@r2 ~]$ ping server -c 5
PING 10.10.2.1 (10.10.2.1) 56(84) bytes of data.
64 bytes from 10.10.2.1: icmp_seq=1 ttl=64 time=1.95 ms
64 bytes from 10.10.2.1: icmp_seq=2 ttl=64 time=2.27 ms
64 bytes from 10.10.2.1: icmp_seq=3 ttl=64 time=2.04 ms
64 bytes from 10.10.2.1: icmp_seq=4 ttl=64 time=1.95 ms
64 bytes from 10.10.2.1: icmp_seq=5 ttl=64 time=1.72 ms

--- 10.10.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.728/1.992/2.278/0.179 ms
```

## View results: See statistics

For this example experiment we used the install script facility to automatically install the necessary software and kick-off the experiment. In this very simple setup, we have installed and launched a web server as well as an iperf server, on the server host. On the client, we have started some processes to test both of these services. To view the results of this experiment:



Web Server Statistics

Logs from the iperf Sever

1. Back in the portal go to the `Slices` Tab
2. The slice that you created with Omni should appear in the list.
3. In the specific slice toolbar, press the button **Launch Flack** . Flack is a graphical, interactive reservation tool for GENI. This will open a new tab where Flack will open.
4. Once the flack page stops loading, press of the (i) button next to the name of the server host.
5. In the new tab that will open press the **Visit** button.
6. Click the **webserver statistics** link to look at statistics. Refresh the page a couple of times to see how the statistics change as the client requests documents.
7. Click the **iperf logs** link to see the statistics from the iperf transfers.

## Next: Teardown Experiment

# Introduction To Omni Tutorial

Design/Setup Execute Finish

## Cleanup resources

Although all your reservations, have expiration times, you should always release your resources once you have completed your experiment to make them available to other experimenters.

i. Logout from your hosts.
ii. In the terminal, where you have been running your omni commands do:

```
omni.py deletesliver -a  pg-utah <slicename>
```