

# jFed Tutorial notes

## Table of Contents

1. Introduction.....	1
2. Starting up jFed .....	1
3. First basic experiment .....	3
3.1. Define the experiment .....	3
3.2. Run the experiment.....	6
3.3. Login on the nodes .....	7
3.4. Install and run iPerf .....	7
3.5. Learning more on your experiment and more functionalities.....	8
3.6. How to get info on your experiment on the node itself .....	9
3.7. Experiment management.....	10
3.8. Let's break things.....	10
3.9. Ending the experiment.....	10
3.10. If something goes wrong ? .....	10
4. Going to a multi-rack experiment and up-scaling .....	11
5. Upscale your experiment .....	12
5.1. Share the LAN.....	12
5.2. Upscale .....	13
6. More advanced.....	14

## 1. Introduction

In this tutorial we will learn how to use the jFed tool (<http://jfed.iminds.be>) to do experiments on the GENI (and compatible) infrastructure. We will start from a very basic experiment and gradually will build up to more advanced topics.

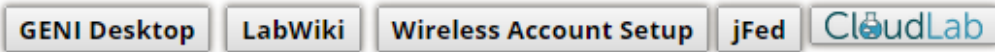
## 2. Starting up jFed

You need to have a recent oracle java 8 (openjdk does not work) running. If not, please install from <http://www.java.com/en/download>. For linux, see: [http://jfed.iminds.be/java8\\_on\\_linux/](http://jfed.iminds.be/java8_on_linux/)

Verify with <http://www.java.com/verify> that you run version 8.

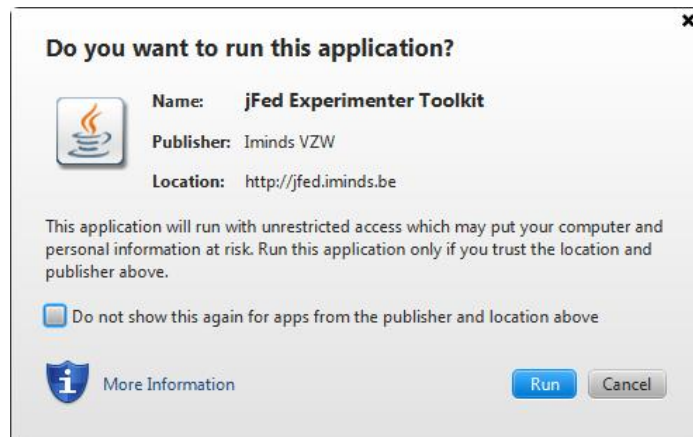
Now go to the Geni portal (<http://portal.geni.net>) and login. Scroll down till you see the Tools section and click the jFed button. (**Note: best results to start jFed from the Geni portal are with Firefox on all platforms.**)

## Tools

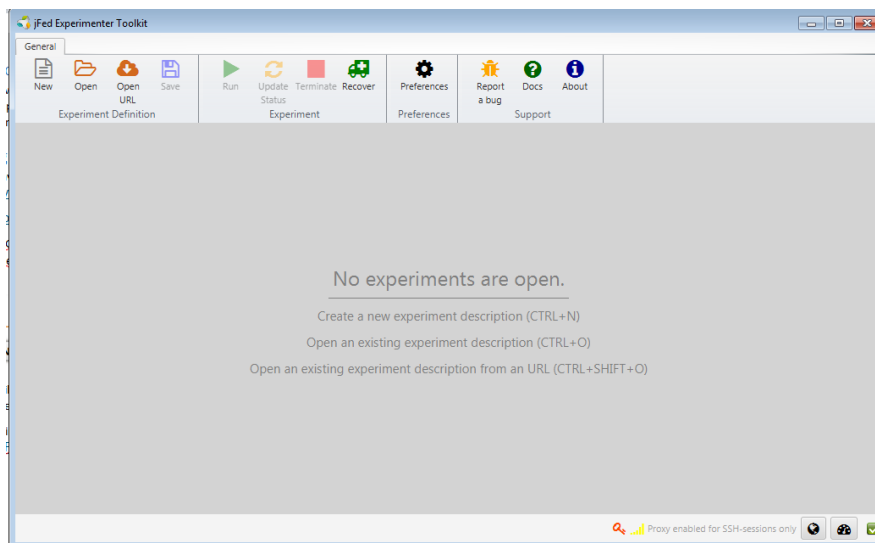


A dialog box will pop up saying that the Java application is signed by iMinds. Click **Run** and optionally you can tick the box 'Do not show this again'.

Note: when using Apple OS X, you may need to change security settings if the above does not work (see [Starting jFed on OS X](#)).



You should now be logged in automatically and see the canvas as shown below:



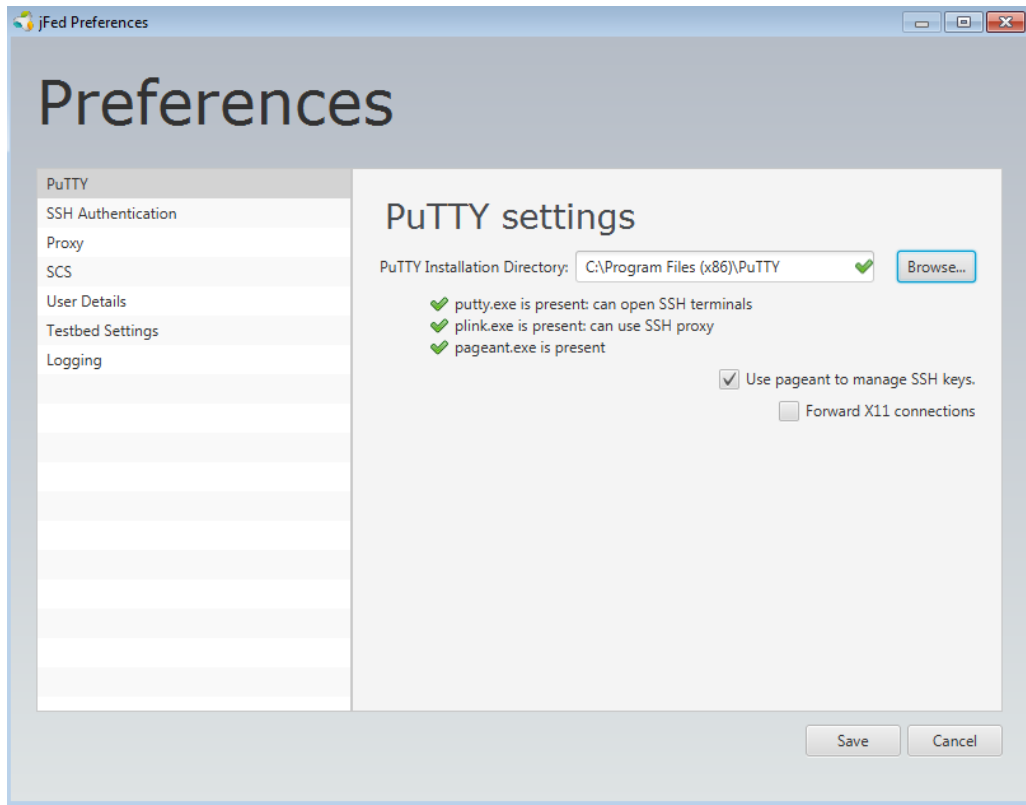
First we will check if our preferences for SSH terminal are okay. Go to 'Preferences' (button on the top). Depending on the platform, the preferences might be opened automatically the first time you run jFed.

For **Windows users**, the SSH terminal that will be used is PuTTY and jFed should have found it as shown below (all checks should be green). If not, then please install PuTTY from the following link:

<http://the.earth.li/~sgtatham/putty/latest/x86/putty-0.63-installer.exe>

For **MAC/Linux users**: the terminal should be automatically chosen.

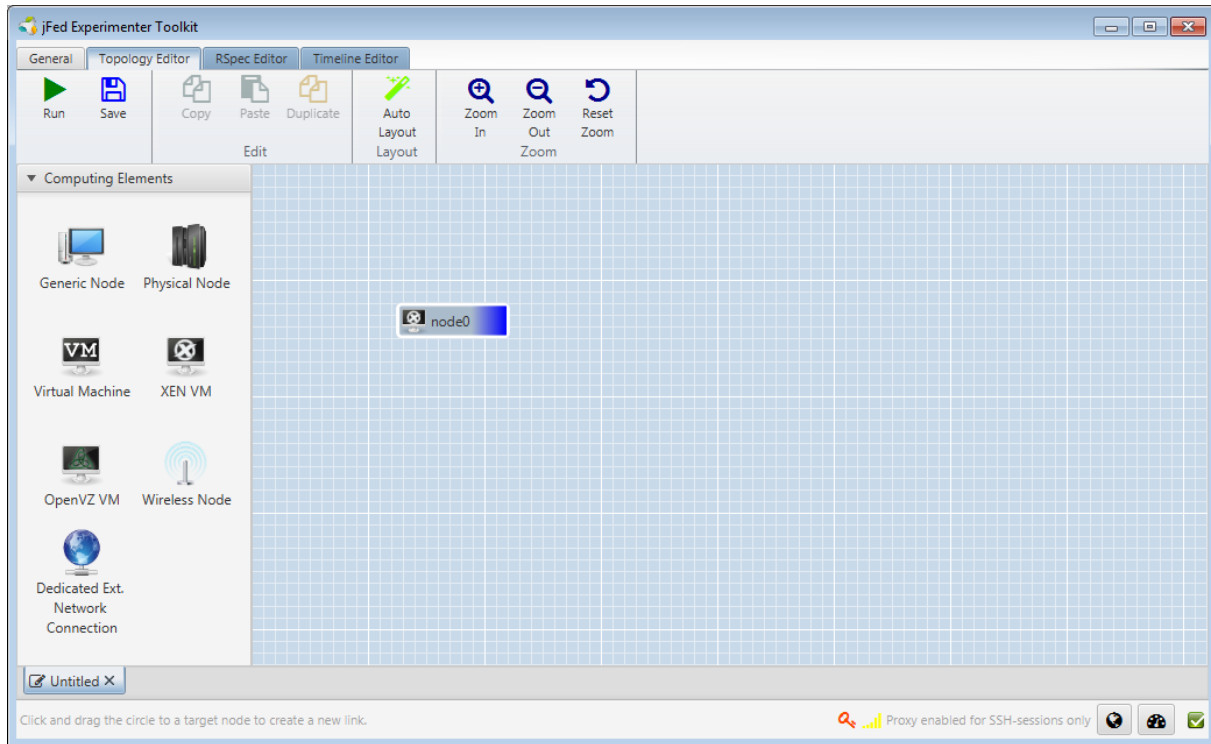
**Note:** check that 'Use pageant/ssh agent to manage SSH keys' is checked.



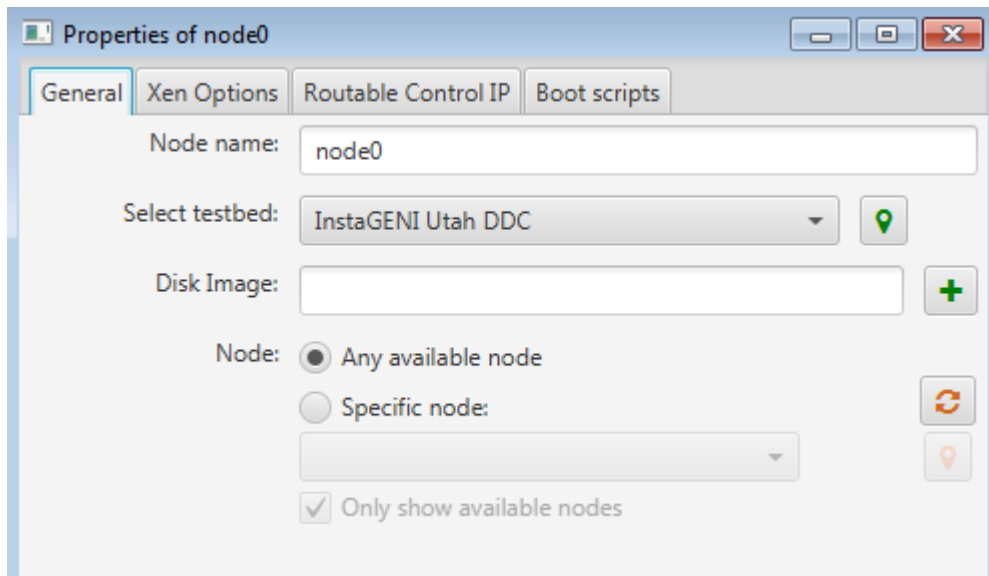
### 3. First basic experiment

#### 3.1. Define the experiment

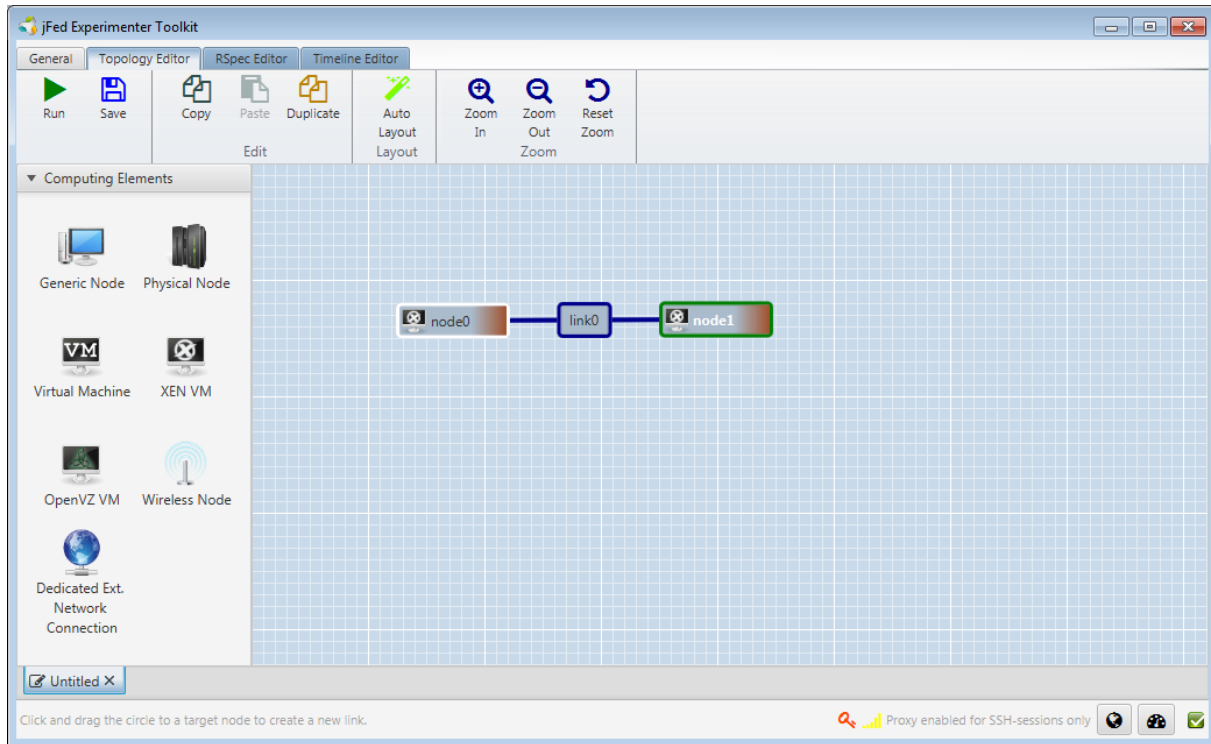
When you click **New**, you get a blank canvas where you can draw your experiment. Let's drag in a XEN VM from the left side to the canvas. For more specific experiments you can right click and configure the node, but for now let it in the default settings.



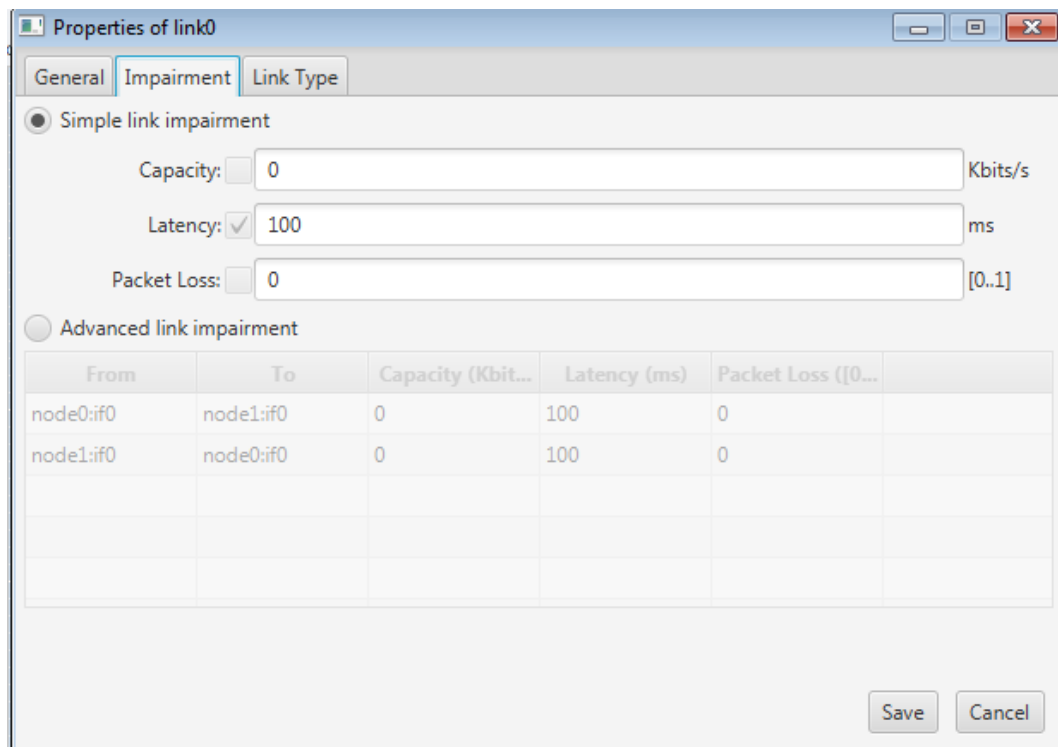
Now right click the XEN VM, and choose the rack that you have been appointed and save. You will see that the color on the right side of the node changes depending on the rack you chose.



Now duplicate the node, by first selecting the node, and the clicking the duplicate button on the top. Then drag a link between the two nodes.

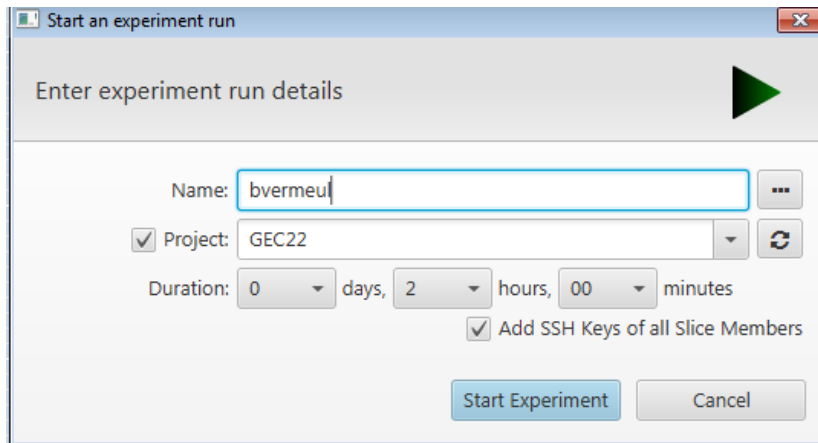


Now double click the link to configure a latency of 100ms on the link in the 'Impairment' tab.



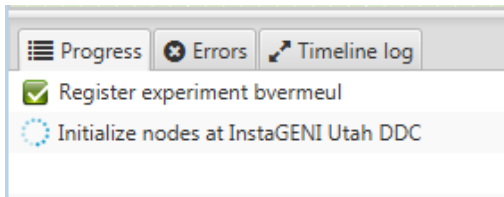
### 3.2. Run the experiment

Now we will run this experiment by clicking the 'Run' button in the top bar. You will see that you need to chose a unique name for your experiment and chose the project and duration (default is 2 hours).

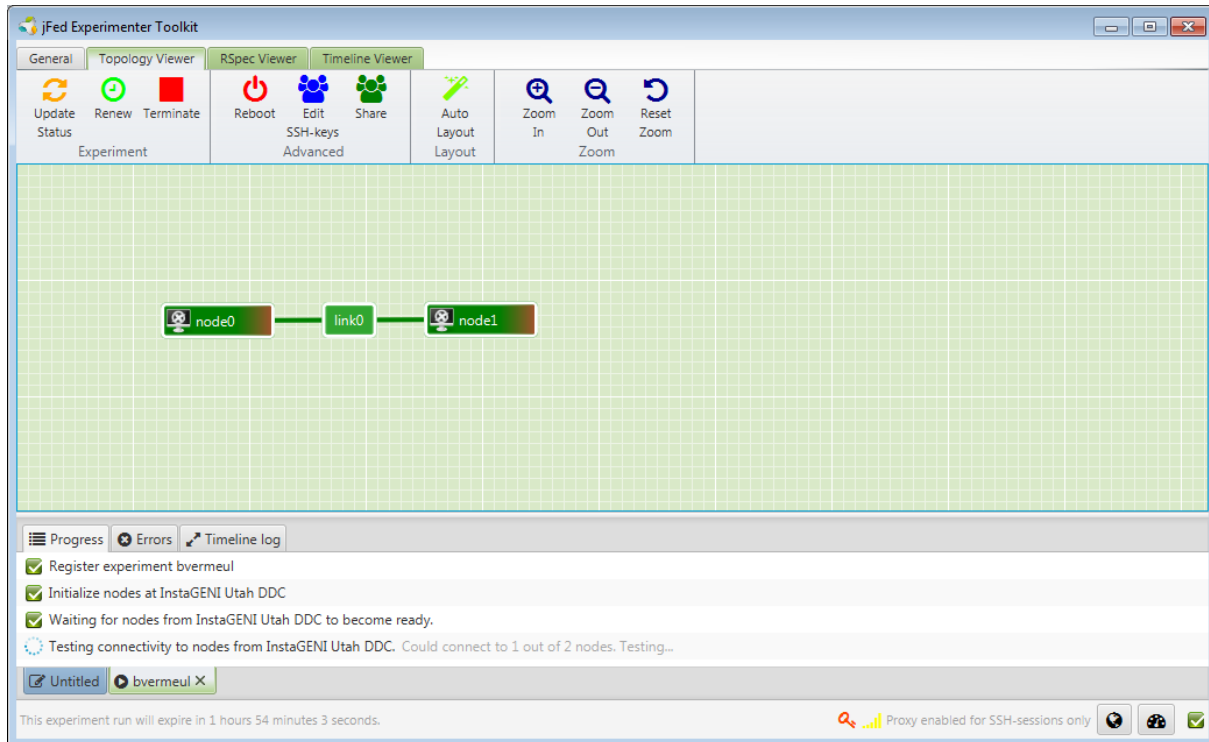


The screenshot shows a dialog box titled "Start an experiment run" with a close button in the top right corner. The main heading is "Enter experiment run details" with a green play button icon to its right. Below this, there are several input fields: a text box for "Name" containing "bvermeul", a dropdown for "Project" set to "GEC22", and a duration selector set to "0 days, 2 hours, 00 minutes". There are also checkboxes for "Add SSH Keys of all Slice Members" and "Start Experiment" and "Cancel" buttons at the bottom.

Click start and at the bottom you will see the progress of the experiment setup (and errors if there would appear some).



After a couple of minutes, the two VMs should be up and running, indicated by the green color of the nodes.



### 3.3. Login on the nodes

We now can simply login on the VMs by double clicking the nodes. Try to ping to the other node and see if the round trip delay (RTT) is 200ms (100ms one way).

```

brechtv@node0: ~
Using username "brechtv".
Authenticating with public key "imported-openssh-key" from agent
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-56-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

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.

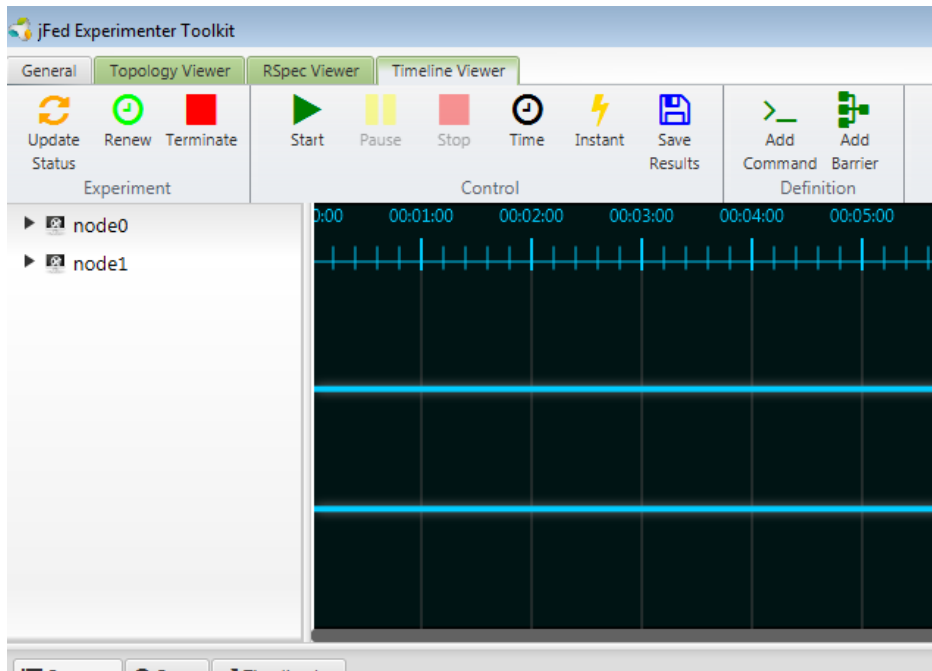
brechtv@node0:~$ ping node1
PING node1-link0 (192.168.0.2) 56(84) bytes of data:
64 bytes from node1-link0 (192.168.0.2): icmp_req=1 ttl=64 time=402 ms
64 bytes from node1-link0 (192.168.0.2): icmp_req=2 ttl=64 time=200 ms
64 bytes from node1-link0 (192.168.0.2): icmp_req=3 ttl=64 time=200 ms
64 bytes from node1-link0 (192.168.0.2): icmp_req=4 ttl=64 time=201 ms
^C
--- node1-link0 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 200.886/251.214/402.009/87.063 ms
brechtv@node0:~$

```

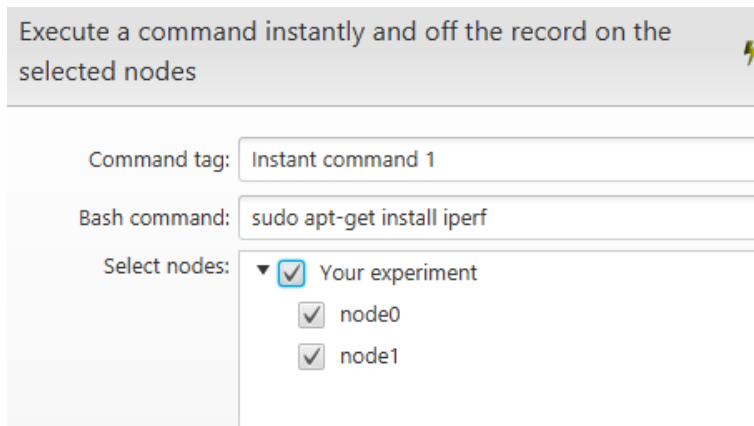
### 3.4. Install and run iPerf

We can install manually iPerf on the nodes, but jFed has a nice feature to install easily software on multiple nodes.

Go to the 'Timeline Viewer' by clicking the tab on the top, and click 'Instant'.



Fill in 'sudo apt-get install iperf' and check both nodes and click ok at the bottom.



Now on one node, run iperf server:

```
iperf -s
```

On the other:

```
iperf -c hostname_other_machine (may depend on which node you are logged in).
```

This will start a 10 second TCP traffic flow.

Which bandwidth do you measure and what did you expect ?

### [3.5. Learning more on your experiment and more functionalities](#)

- Going more in detail:



- Go back to the design mode (blue tab), and go to the rspec editor: you can also manually change things, save the RSpec, etc
- If you right click a node and configure it, you can also select images
- For XEN VMs you can configure RAM, extra disk, routable control IP. Also Exogeni can be selected under virtual machine and configured. (we will not start such an experiment)
- Go to the running experiment (bottom green tab), in RSpec view and verify details on nodes, login, RSpec manifest
- Information on options for a running node:
  - Node reload = reload the image for that node (=reformat the node)
  - Node info = detailed ssh info + interface info
  - Node reboot = simple reboot of the node
  - create image = will take an image of your node that you can use in new experiments (we won't do this now)

### 3.6. How to get info on your experiment on the node itself

The previous section described how to get information about nodes and your experiment from jFed. Now we will try to get this same information on the node itself (e.g. to use in scripts you want to use)

We use the 'geni-get' tool for this.

Geni-get supports the following commands:

```
geni-get commands
{
  "client_id": "Return the experimenter-specified client_id for this node",
  "commands": "Show all available commands",
  "control_mac": "Show the MAC address of the control interface on this node",
  "geni_user": "Show user accounts and public keys installed on this node",
  "getversion": "Report the GetVersion output of the aggregate manager that allocated this node",
  "manifest": "Show the manifest rspec for the local aggregate sliver",
  "slice_email": "Retrieve the e-mail address from the certificate of the slice containing this node",
  "slice_urn": "Show the URN of the slice containing this node",
  "sliverstatus": "Give the current status of this sliver (AM API v2)",
  "status": "Give the current status of this sliver (AM API v3)",
  "user_email": "Show the e-mail address of this sliver's creator",
  "user_urn": "Show the URN of this sliver's creator"
}
```

You can try some, e.g.

`geni-get slice_email`

Now, we will demonstrate a more advanced script which parses the manifest RSpec and shows you th info on your node:

```
wget http://doc.ilabt.iminds.be/ilabt-documentation/_downloads/geni-get-info.py
chmod u+x geni-get-info.py
./geni-get-info.py
```

And you should see info on interfaces and IP addresses and so.

### 3.7. Experiment management

In the GENI framework, you can combine multiple tools as the tools mention the information about experiments in the portal.

- Verify on geni portal that your experiment is listed and verify the expiration date
- Renew in jFed can extend this (for all slivers), or you can renew in the portal
- In jFed, you can Edit ssh keys on the nodes if you want to add other users ('edit ssh keys button')
- If you close jFed, or have created an experiment with another tool, you can 'recover' this experiment
- In jFed you can share the experiment with other people in your project, to make others member of the slice and be able to recover it in jFed e.g.

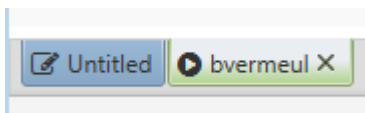
### 3.8. Let's break things

- Verify that eth0 is your control interface (where you are logged in)
- Turn off: 'ifconfig eth0 down', your ssh connection will be lost
- The next steps might not work for everyone, but you can try them:
  - Now, in jFed right click the node and click reboot
  - Right click and 'show console' to show you how it boots
  - After that you can access the node again

### 3.9. Ending the experiment

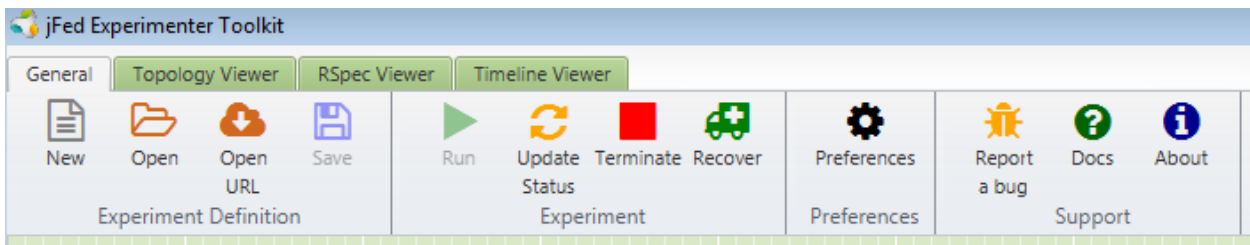
To release your resources before the endtime of your experiment, you can click the `Terminate` button at the top in jFed. After that the nodes will become black and if your ssh connection is still open, you can see that the node will be shutdown.

At the bottom of the jFed window, you see multiple tabs: the blue ones are experiment definitions, the green/red ones are experiment runs. You can click to switch tabs.



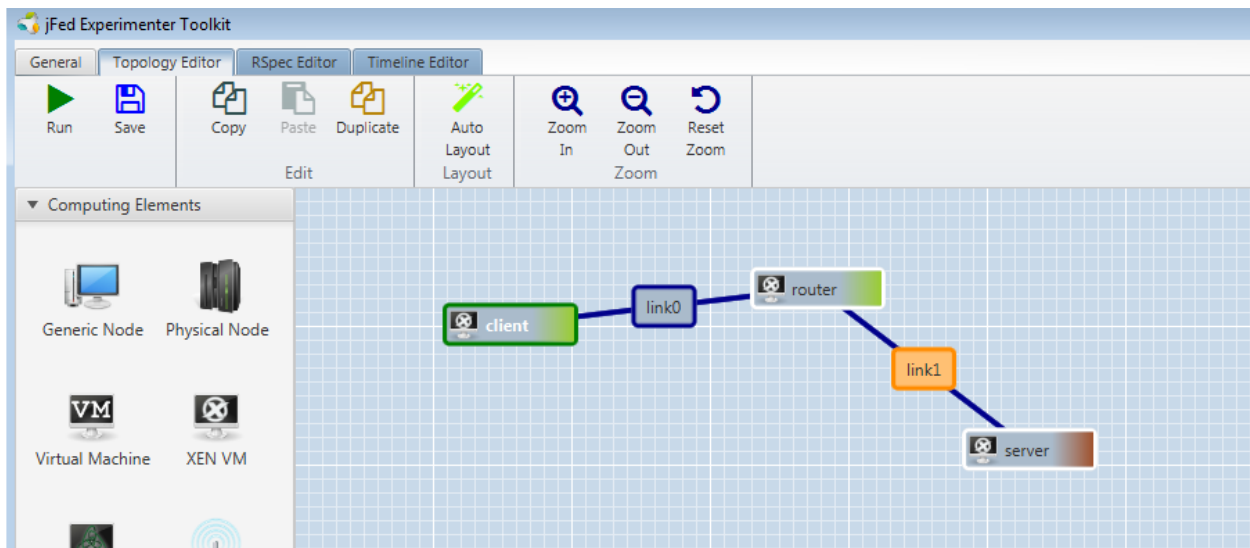
### 3.10. If something goes wrong ?

**Push the bugreport button on the general tab and report what went wrong.**



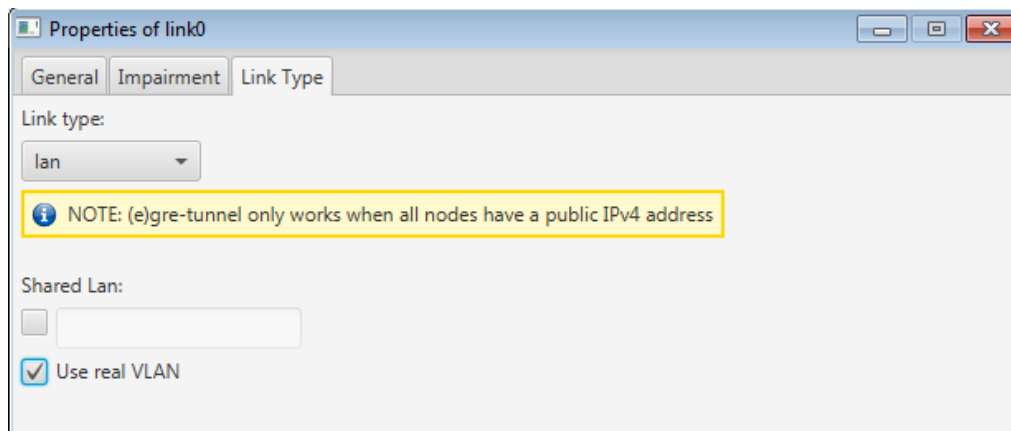
## 4. Going to a multi-rack experiment and up-scaling

Now we will do a more advanced experiment using VMs on multiple racks. You should have received a paper with 2 different racks mentioned on it. Design your experiment as follows:



So, we define a server in one rack, and then a router and client on the other rack. As you can see, you can define names for the routers (do you find how?). Verify the IP addresses that jFed has chosen by double clicking the links.

**Very important for the next part of the tutorial: check for the link between client and router that in the link properties – link type, the 'Use real vlan' check is marked. This will force the two VMs to be installed on different physical hosts.**

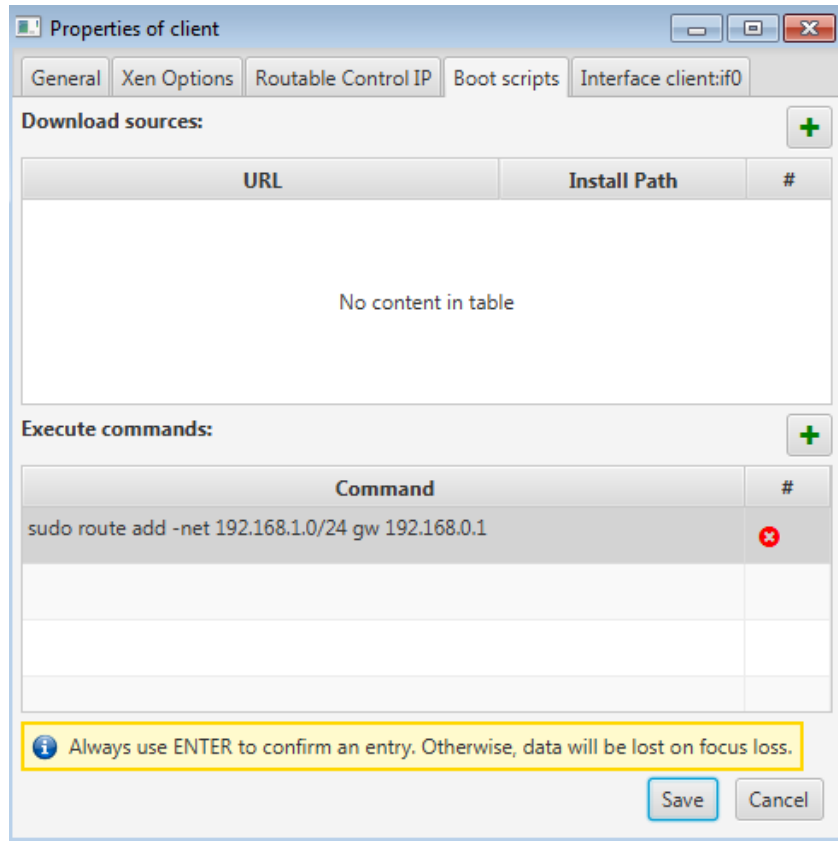


Now run the experiment, verify that you can ping from the server to the router (watch the latency).

Configure the proper routing on client and server. This depends on your exact topology, but the commands are like this:

```
sudo route add -net 192.168.1.0/24 gw 192.168.0.1
```

In the future, you can also configure this before running your experiment, by using the node boot scripts option. Double click the node, go to the boot scripts:



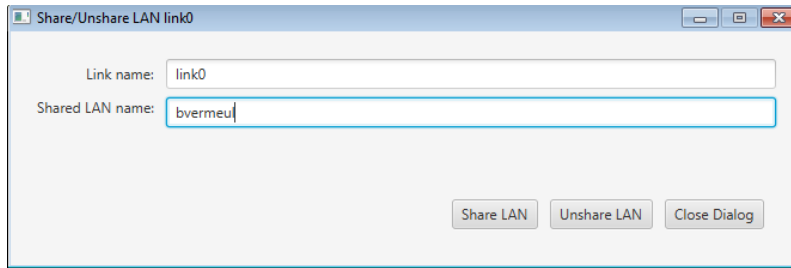
Note: Stitching bandwidth is by default set to 10Mb/s in jFed. You can change this in your experiment definition by clicking the link and changing the link impairment – bandwidth.

## 5. Upscale your experiment

### 5.1. Share the LAN

Now we want to add more clients. We will open up the network between client and router ('sharing the vlan') by right clicking the green link and choosing 'share lan':

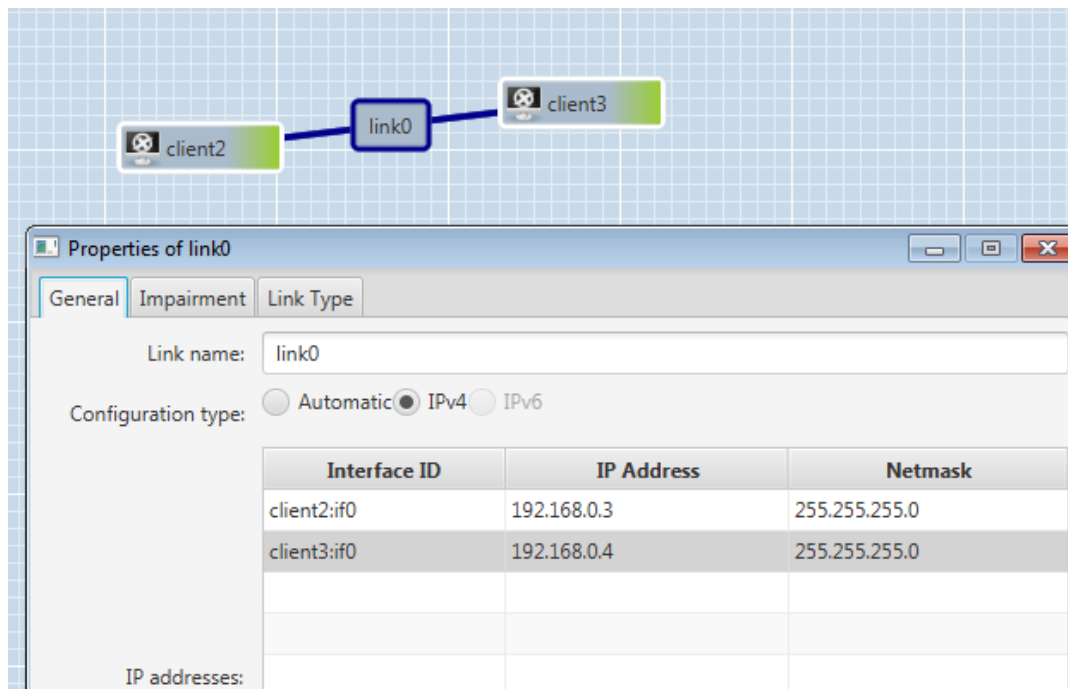
You have to choose a unique name for the link (and remember it) and type it in the 'Shared LAN name' box/



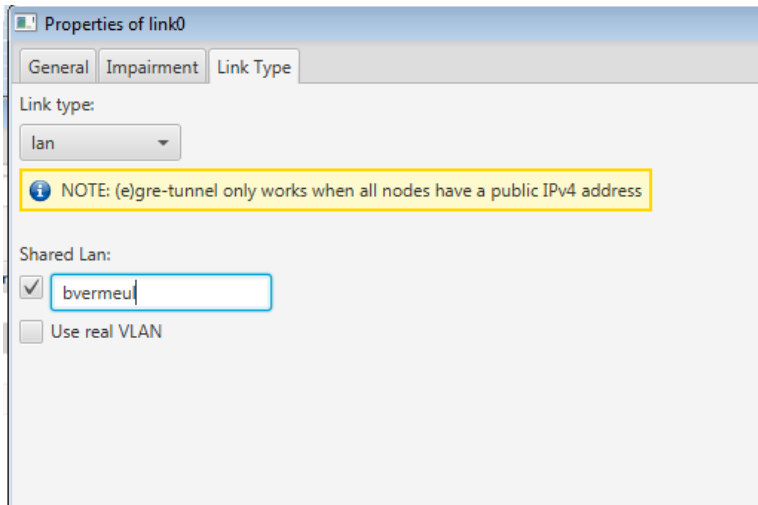
## 5.2. Upscale

We will create now a new experiment which will contain two nodes that will be added to the client vlan that we opened up in the previous section. **Of course, they should be on the same rack as the client and router in the previous part.**

We have to edit the IP addresses so the new nodes do not clash with the existing ones, and they should be in the same subnet. Easiest to do this, is double click the link.



To link this new network with the existing one, go to the 'Link type' tab and select and fill in the LAN unique shared lan name you used in the previous section.



Verify after login in on the nodes that you can ping the existing IP addresses. You might need to add a route to reach the server.

It should be straight forward from this tutorial that you can easily scale up further:

- Use the duplicate button in jFed
- Use bash scripting to create RSpecs
- Use geni-lib to make upscaled RSpecs

## 6. More advanced

In this last part, have a look into more advanced preferences in jFed you can configure:

- Ssh authentication: add your own key
- Proxy: in case of firewall problems or to access IPv6 nodes
- Configure SCS
- Testbed settings: exosm setting to chose the central exogeni broker or go to the specific rack controllers
- Ssh agent forwarding to login from node to node automatically: login on a node, and then ssh to another IP address in your topology