

Getting To The Tutorial Environment

For this tutorial, we have configured docker containers with the current binary release of CCNx installed. For those who'd like to participate, we'll give you login credentials. You will need to log in to our server, run some docker commands to create and log in to your private docker container, and then 'git clone' the tutorial code sample repository.

NOTE: The server will be available through the end of the conference. At that time, all contents will vanish - so be sure to copy anything you want to keep off of the server. The public github repository will remain available.

Log in to the server with the credentials given to you.

```
> ssh <userNN>@54.215.238.109
> passwd: <passwd>
```

At this point, you have a shell on the VM. But you are NOT in your private workspace. The next step is to start your docker container. We use the convention of naming the container <username>.cid so, for example, if your username is user10, your container will be named user10.cid.

Start a new private docker container:

```
> /usr/local/bin/createDockerContainer.sh
```

At this point, you have a docker container running in the background. You should see a line of output with an IPAddress in it:

```
"IPAddress": "172.17.0.XX", <-- This is your docker container's IP address
```

If you need to get your container's IPAddress again later, you can run:

```
> /usr/local/bin/getDockerIPAddress.sh
```

Now that you know its address, you can ssh into it as 'user'. **Not your userNN, but just 'user'.** **This only works from shells on the server.**

SSH in to your container from the same window:

```
> ssh user@<your IP address>
passwd: user
```

At this point, you have a shell into your private docker container. The ccnx binary distribution is installed in /usr/local/ccnx. Note that you can ssh in to your container as many times as you'd like, you just need to ssh in to the primary server (54.215.238.109) first and then to your container's IP address.

To download the tutorial code examples, you'll need to clone them from github.

Download the tutorial sample code from Github:

```
> git clone https://github.com/PARC/ccnx-tutorial-icn-2015.git
```

You should now have the tutorial sample code in the ccnx-tutorial-icn-2015 directory.

Running The Forwarder

There are two CCNx 1.0 forwarders available, Metis and Athena. Metis is more mature and faster, while Athena is better suited for experimenting. Both should be in your path. For this tutorial, we will be using Athena.

To start Athena:

```
> athena &
```

You can configure Athena via the athenactl command. You use athenactl to set up links, add routes, and change the log level. Before you can use it, however, you need a key.

Creating a key:

```
> parc_publickey -c <key filename> <passwd> <subject>  
e.g. parc_publickey -c keystore my_password tutorial
```

Now you can use that key when running athenactl:

To get usage help:

```
> athenactl -f keystore -p my_password -help
```

To add a link:

```
> athenactl -f keystore -p my_password add link tcp://<ip or hostname>:9695/name=linkname
```

To add a route:

```
> athenactl -f keystore -p my_password add route linkname lci:/name/to/route
```

To list routes:

```
> athenactl -f keystore -p my_password list routes
```

To list links

```
> athenactl -f keystore -p my_password list links
```

Athena is verbose by default. To reduce the log messages:

```
> athenactl -f keystore -p my_password set level off
```

Other levels are available:

```
> athenactl -f keystore -p my_password set level [off | debug | error | info | all ]
```

Here's an example of adding a route from one forwarder to another. It assumes the other forwarder is running at <other IP>, port 9695.

First, add the link:

```
> athenactl -f keystore -p my_password add link tcp://<other IP>:9695/name=other_machine
```

Then, add the route using the name of the link.

```
> athenactl -f keystore -p my_password add route other_machine lci:/ccnx/tutorial/something
```