**Installing and starting the PADS Virtual Machine**

1. Insert the flash drive in your laptop and copy the zip folder to your computer
2. Unzip the folder
3. If VirtualBox is not already installed on the machine, install the OS-specific installer from the Virtualbox directory on the flash drive. Additionally, install the Extensions of the VirtualBox.
4. Run VirtualBox
5. Next, choose File/Import Appliance and navigate to the PADS VM Image
6. Check the box to refresh the MAC Address and import the VM
7. Start the PADS VM.

**Hands-on Exercises on the PADS Framework**

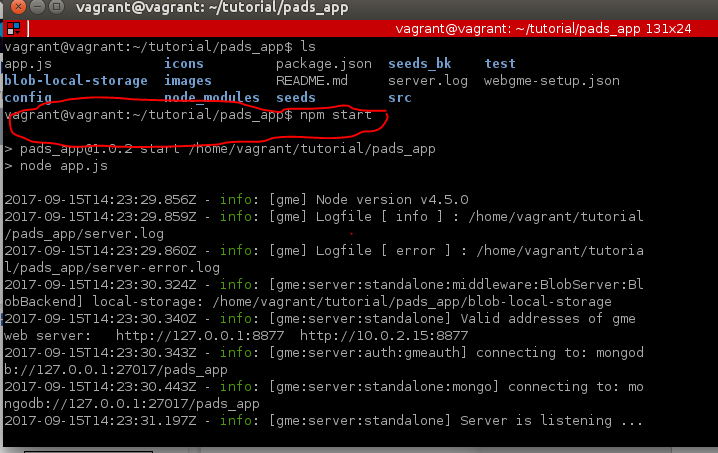
**Objectives and goals of these exercises**

The first exercise illustrates the role of a student wanting to learn and play with a distributed systems algorithm. In this case, it is assumed that an instructor has created the complete metamodel along with the appropriate plugins for the desired algorithm. The student’s responsibilities are to use the metamodel provided by the DSML to model an instance of the distributed algorithm under study by experimenting it under different configurations. Thus, in this exercise the attendee will use the supplied metamodel to model an instance of a publish/subscribe communication paradigm along with brokers, and experiment with this setup in an emulated networked environment.

The second exercise is tailored towards the instructor wherein the instructor is responsible for developing the metamodel and the appropriate plugin for the distributed algorithm that the instructor wants to teach. Due to time limit in this tutorial, in this exercise we have supplied an almost complete metamodel for the MapReduce algorithm and require the attendee to complete it by supplying the missing pieces. The plugins are already supplied.

**Exercise 1: (Student Role)**

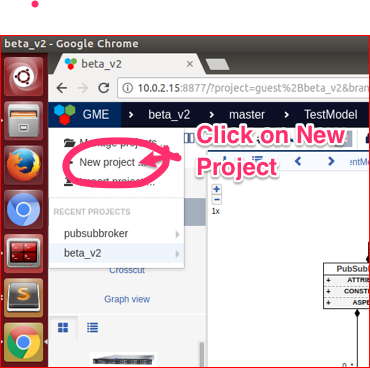
In this exercise we shall create a simple topology that demonstrates the Publish Subscribe communication pattern using broker(s) in distributed systems.



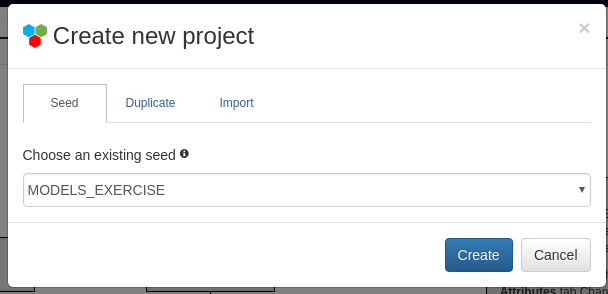
1. First lets start our PADS app. Using the terminal prompt issue following command as shown below.

**cd ~/tutorial/pads\_app**

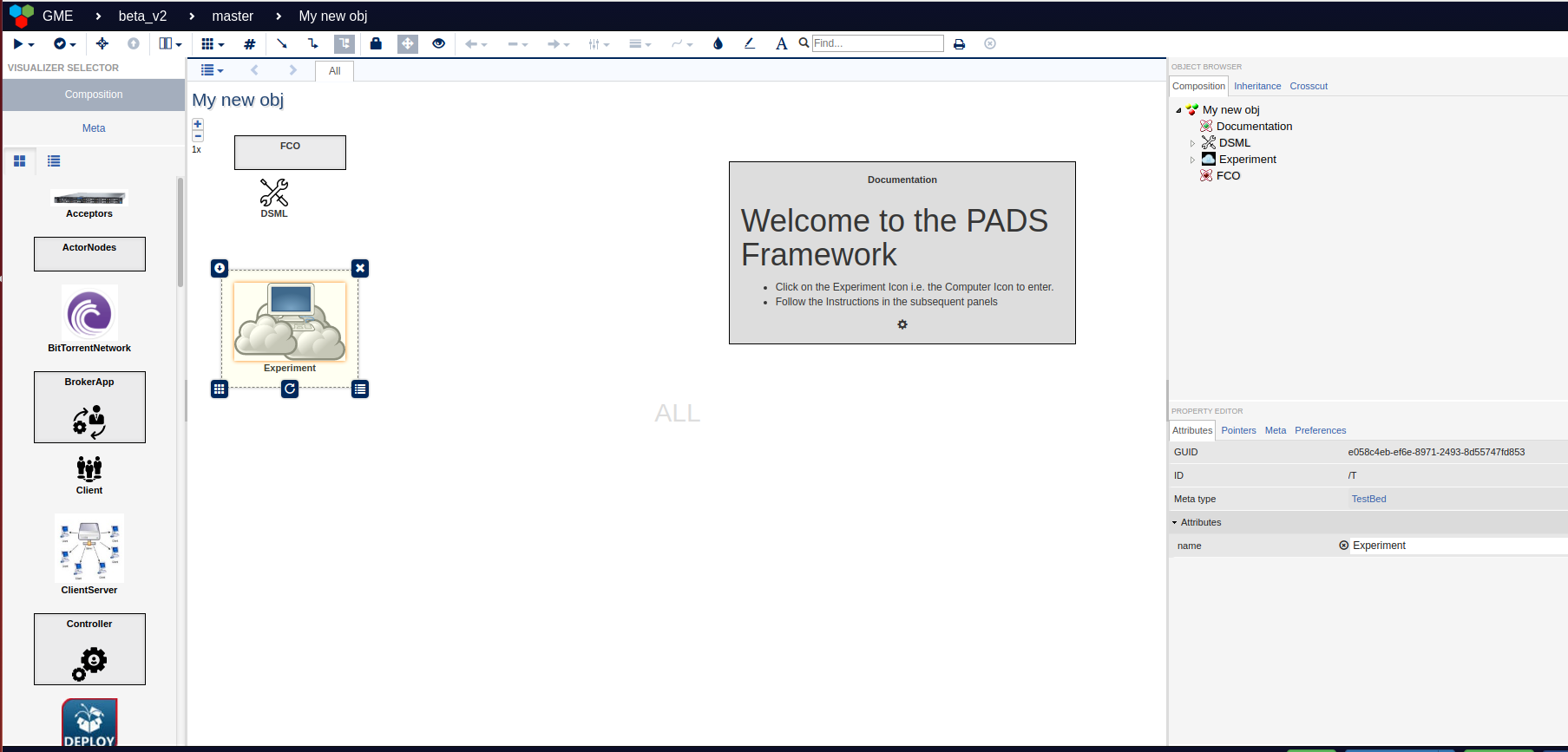
**npm start**

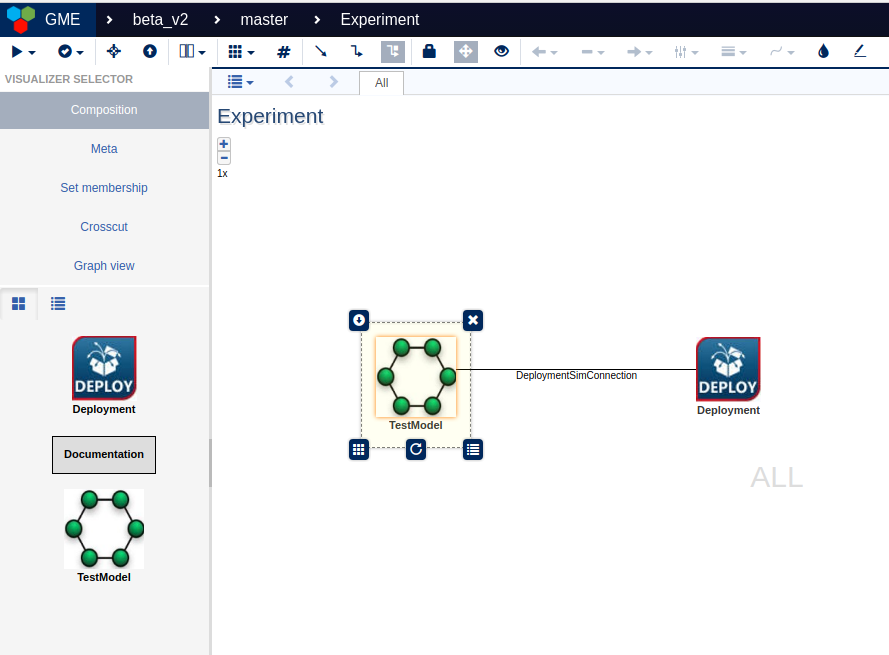
1. Create a new project called **Exercise1\_pubsubbroker** using the initial seed named MODELS\_EXCERCISE. See the screenshot below for guidance.

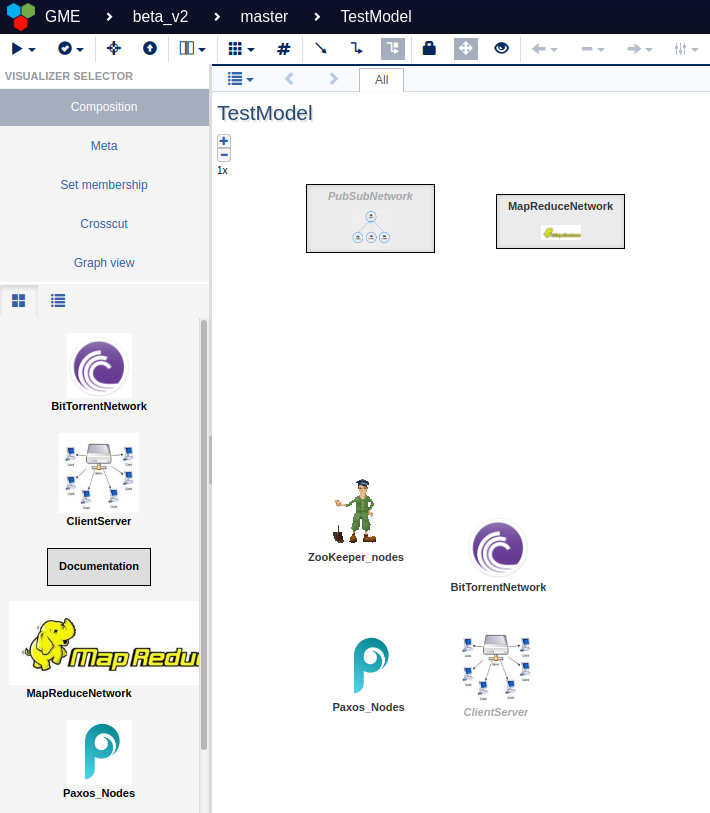


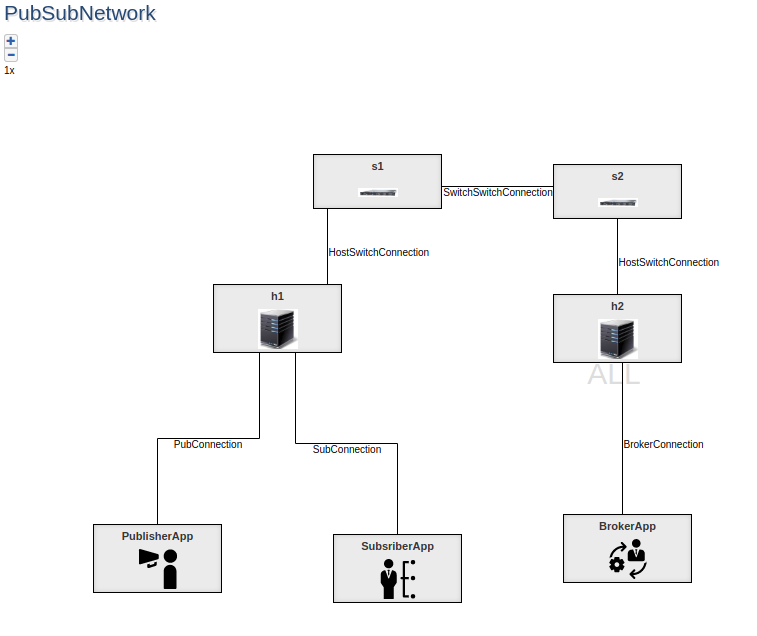


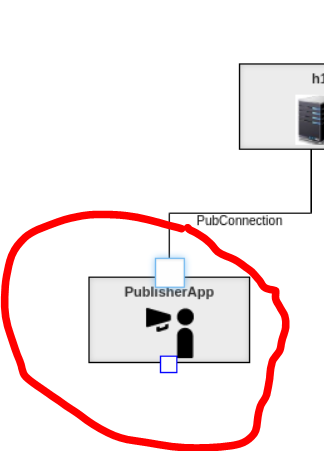
1. Double click on Experiment as shown in the screenshot below. Repeat the actions for both TestModel and then for PubSubNetwork (see screenshots below)

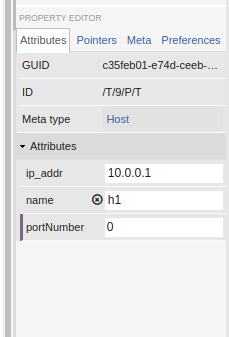
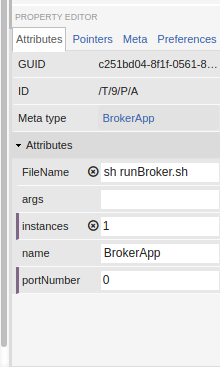
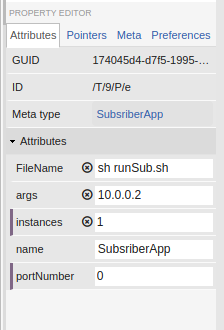
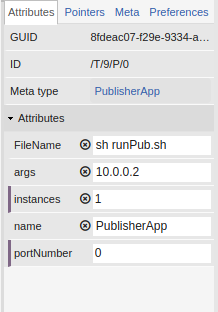


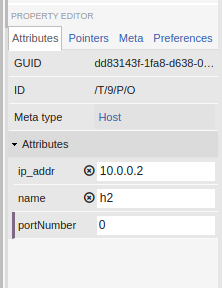




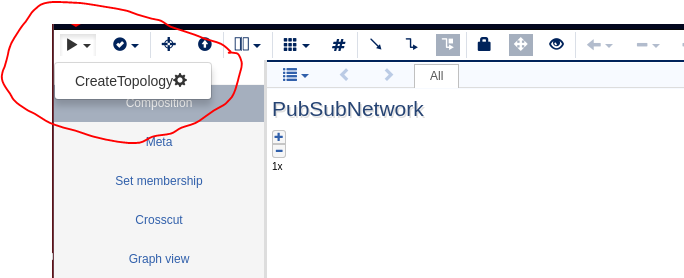
1. Create a network topology in the blank screen as shown below by dragging and dropping icons from the left panel. Also, configure the attributes of the individual nodes using the Property editor on the lower right side panel as shown below. Set the Name and the IP address properties of host h1
2. To create connection between the PublisherApp and the h1, hover the mouse over the PublisherApp icon, this will pop blue box. Click on the blue box and connect the other end to the h1







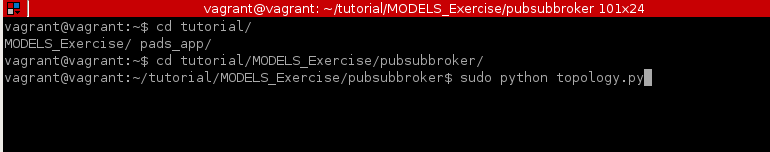
1. Once these steps are completed with exact names as shown above, we are now ready to run the plugin that will generate the code.
2. Execute the plugin **CreateTopology**



1. Click on the Green completion notification panel and click on the **GeneratedFiles.zip**



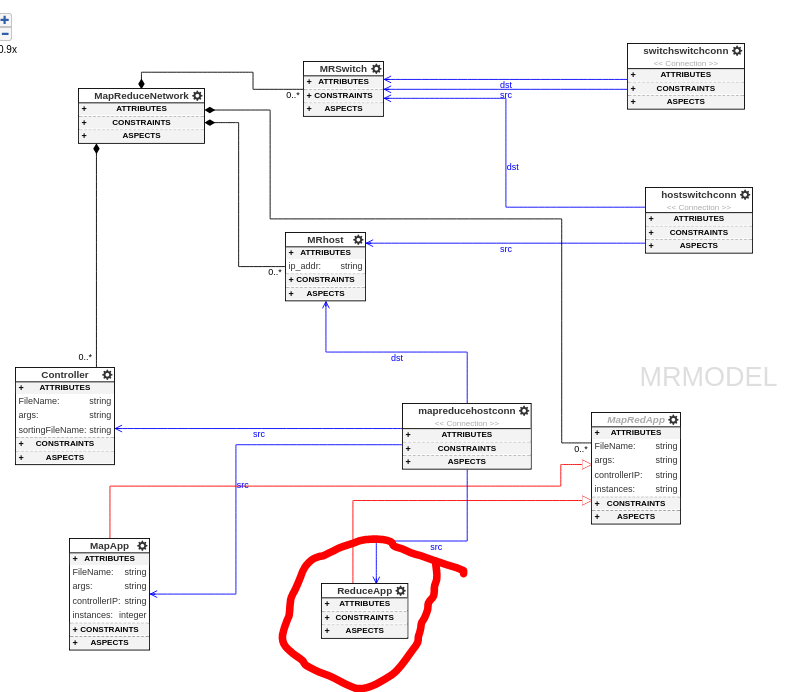
1. This will download the files and will save it in the Downloads Folder.
2. Using the File Manager, unzip the files that were just downloaded
3. Copy all the three generated files: topology.py, command.txt, metadata.json into the pubsubbroker directory
4. Open a terminal program and navigate to the **~/tutorial/MODELS\_Exercise/pubsubbroker** directory
5. Execute the following command in the shell prompt. Ctrl-Alt-t opens up a shell.
6. sudo python topology.py



1. This will instantiate the topology and execute command.txt files commands one by one in the mininet environment
2. After some time the program execution will stop by cleaning all the networks created and exits
3. One can see the results in the results directory

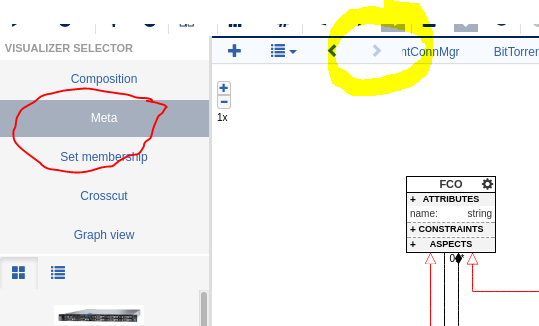
**Exercise 2: Instructor Role**

In this exercise we shall extend our DSML which will support the execution of the popular MapReduce paradigm of distributed computing.

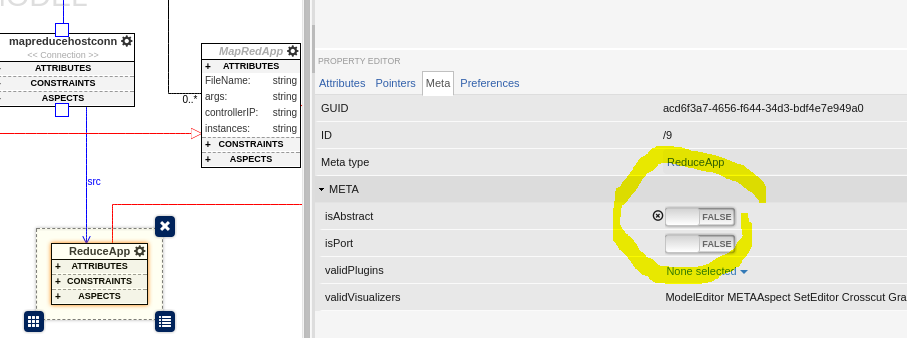


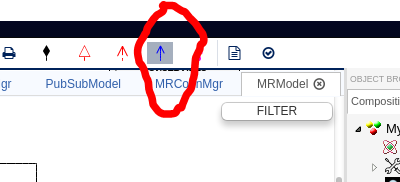
Double Click on the MapReduce meta model.

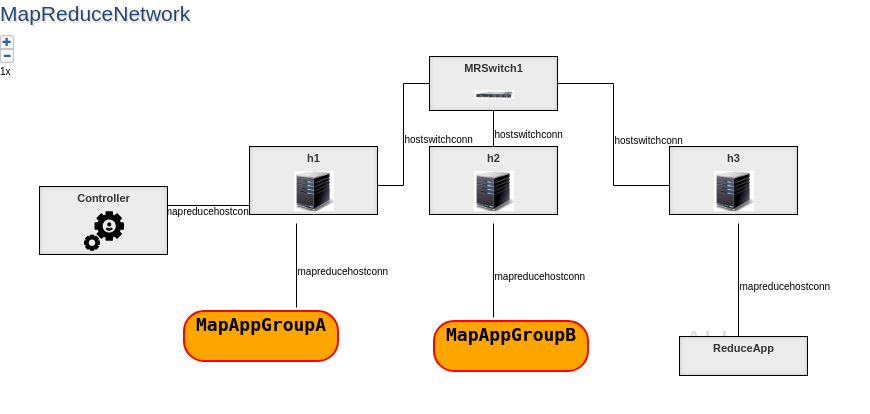
1. First go to the META view

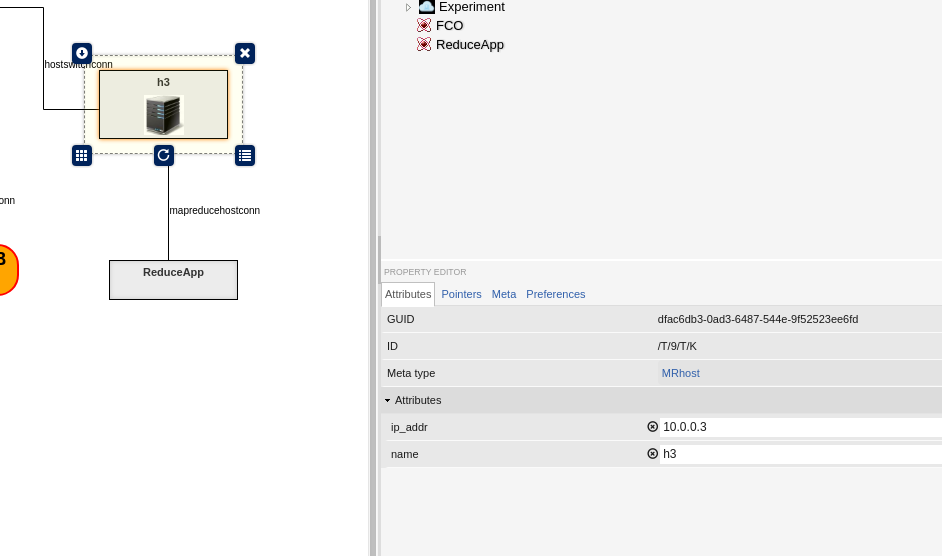


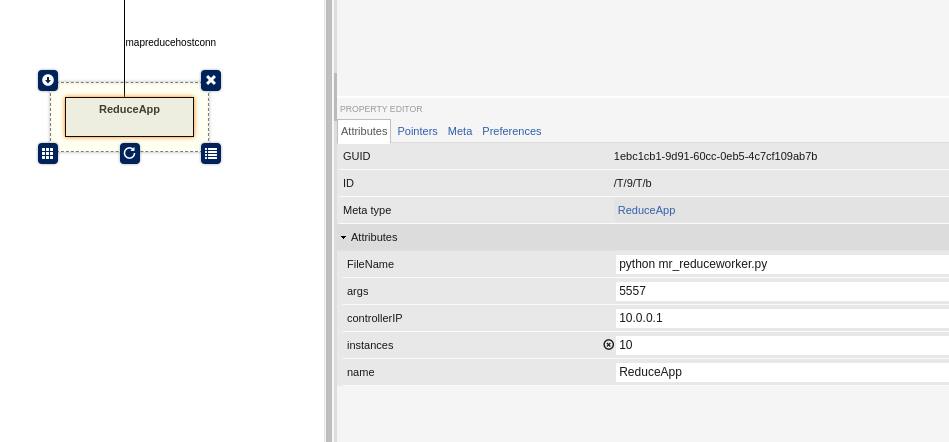
1. Go to the MRModel
2. Add an **MapRedApp** element from the left part browser
3. Name the **MapRedApp** element as **ReduceApp**
4. From the property editor make **isAbstract** from the Meta tab Property as **FALSE** as shown below.



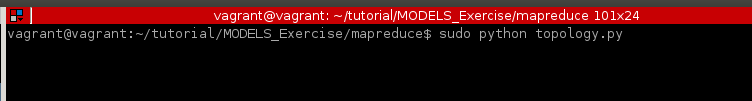
1. Next, for our model we are going to place the **ReduceApp on MRHost machine.**
2. Thus we need to create a connection between the **ReduceApp** and **MRHost** object. Click on the Blue pointer on the top panel as shown below and connect FROM **mapreducehostconn** to **ReduceApp** and name it as **src pointer.**
3. 
4. **This completes our DSML for the MAPREDUCE**
5. Next, we go back to the **COMPOSITION** view by clicking on the left hand panel
6. We then add **MRHost** from the left panel **partbrowser,** name it as **h3**.
7. Set the ip address of the **h3** as **10.0.0.3**
8. We also add ReduceApp from the left panel partbrowser and set the number of **instances to 10**
9. Follow the naming convention as shown in the figure below







1. Once we have our model ready, then we execute a plugin that will generate the necessary topology code.
2. Execute the plugin **mrMininetTopo**
3. Click on the Green completion notification panel and click on the **GeneratedFiles.zip**
4. This will download the files and will save it in the Downloads Folder.
5. Using the File Manager, unzip the files just downloaded
6. Copy all the three files generated: topology.py, command.txt, metadata.json into the **mapreduce** directory
7. Open a terminal program and navigate to the **~/tutorial/MODELS\_Exercise/mapreduce** directory



1. Execute the following command in the shell prompt
2. sudo python topology.py
3. This will instantiate the topology and execute command.txt files commands one by one in the mininet environment
4. After some time, the program execution will stop by cleaning all the networks created and exits
5. One can see the results in the result.csv and the results directory