**Exercise 6.2 Puppet Setup and Deploy**

We will now pull and run a puppet server and a puppet client in Docker containers.

Open the Cloud Platform Console at [https://console.cloud.google.com](https://console.cloud.google.com/).

Click on the three horizontal bars at the left most side of the blue bar near the top of the browser window. *Select Compute Engine*.

Select *VM Instances*. You should see the virtual machine you created earlier.

Click on the checkbox to the left of the VM name and then select *START*. It will take a few moments to start.

Click on *SSH* to start a terminal window.

**Change the host name to student:** Find the icon that looks like a gear in the upper right-hand corner of this terminal browser window and select *Change Linux User Name*. Enter *student* and *click Change*. Now, notice the prompt that says "student@lab:~$"



Create a Docker network for puppet.

docker network create puppet

Check what has been created.

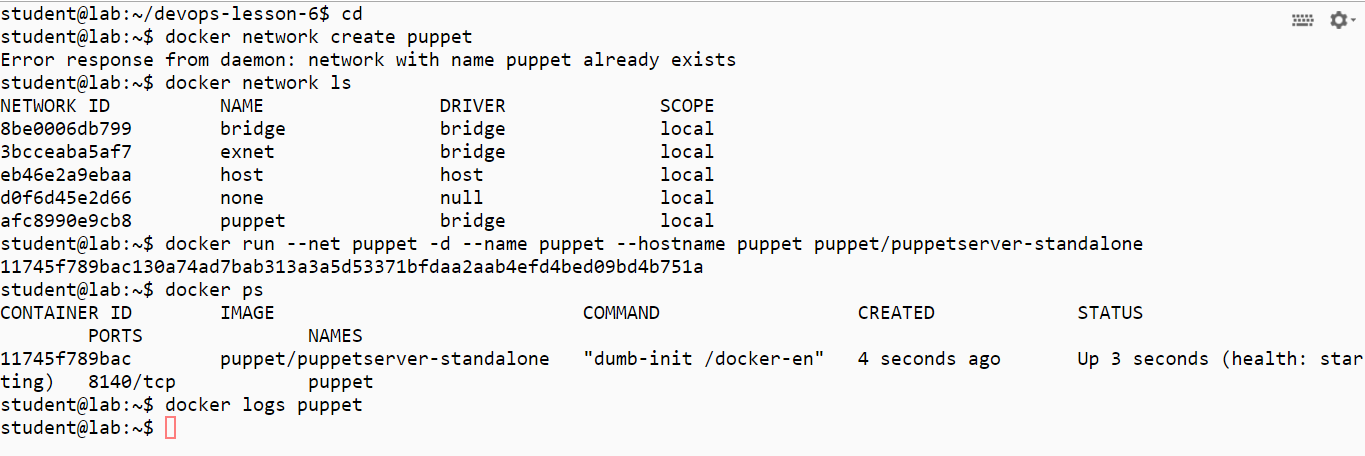
docker network ls

**Step 2**

Start the puppet server.

docker run --net puppet -d --name puppet --hostname puppet puppet/puppetserver-standalone  
docker ps

We need to ensure that the server is running. Extract the logs until you see the message:  
Puppet Server has successfully started and is now ready to handle requests.  
docker logs puppet

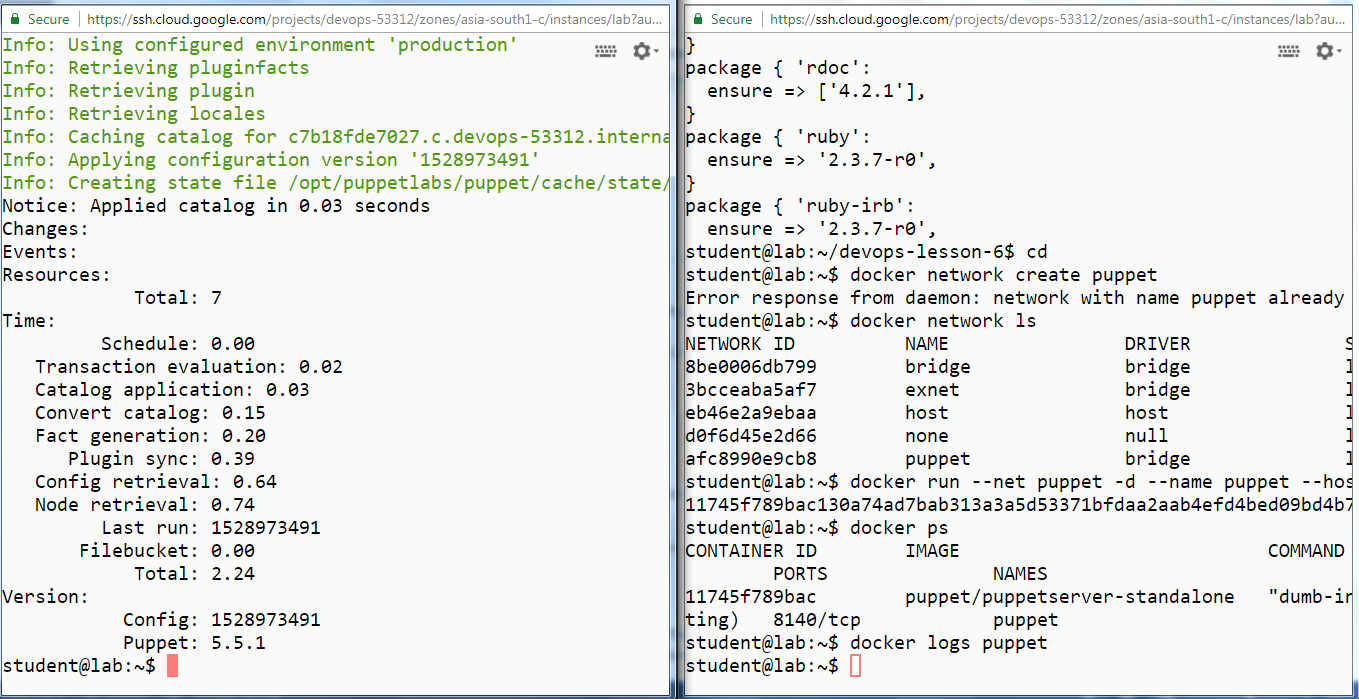


**Step 3**

Open another terminal window using SSH.

Start the puppet agent. It will connect to the server and perform a certificate exchange. Check out the output.

docker run --rm --net puppet puppet/puppet-agent-alpine



**Step 4**

You will see in the puppet master window that the connection from agent is made and agent is pulling the manifest information that the master is holding.

**Step 5**

Tidy up.

docker rm -f puppet

You will need to stop the lab computer at the end of each day to prevent it from accumulating costs during the evening and night.

From the Web UI, you can navigate to the Compute Engine section and select your lab computer. When it is selected, click on the icon representing the "Stop" operation as shown below:

