**Lab 04**

**Laboratory Exercise**

**LAB EXERCISE**

**Time to Complete**

Approximately 60 Minutes

**What You Need in this lab**

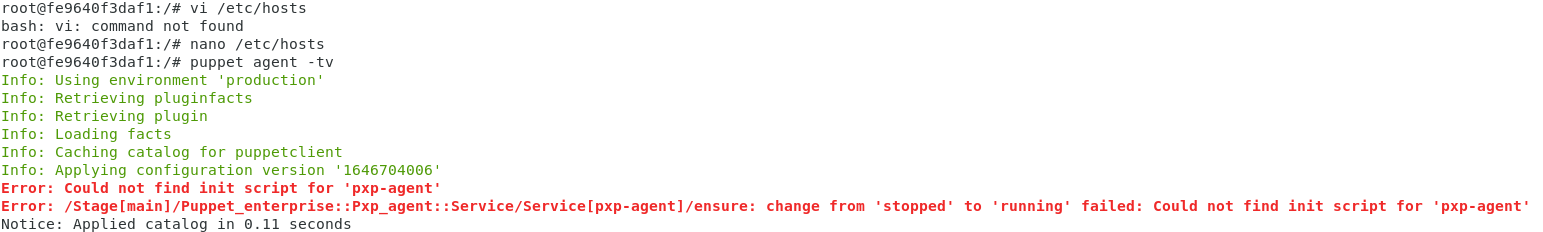
* Establish the communication between Puppet master and agent
* Create a node declaration for your agent
* Work with Bash Aliases

|  |
| --- |
| **Part 1: Master and Agent** |

The main objective in this part of the lab is to commit the container’s file changes or settings into a new image and run the image as a privileged mode container in Docker.

It is a known issue that Puppet agent runs can fail if the Systemd is disabled or removed in a Docker container.  
  
Source:   
<https://puppet.com/docs/puppet/7/known_issues_puppet.html#known_issues_puppet-pa-2028>

<https://tickets.puppetlabs.com/browse/PA-2028>



This can overcome by running a privileged mode container in Docker.

**Steps/Hints**:

1. Switch to root.

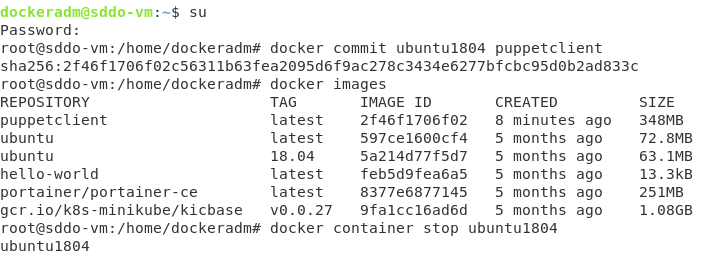
**# su**

1. Issue the following command to commit the file changes or settings of the ubuntu1804 container into a new image.

**# docker commit ubuntu1804 puppetclient**

1. Stop the existing container.

**# docker container stop ubuntu1804**

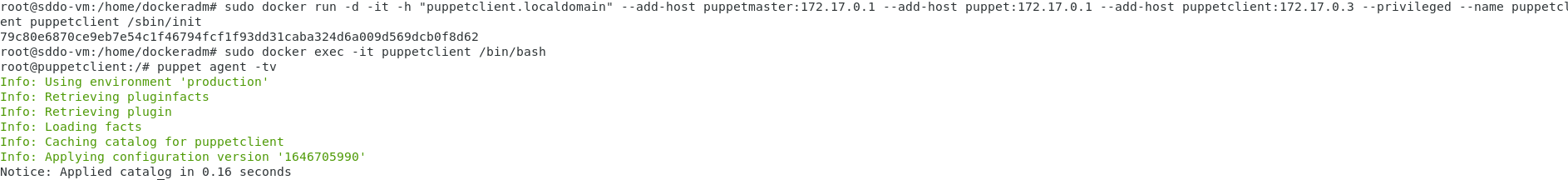


1. Run a new container in privileged mode and test the connection between Puppet master and agent.

**# sudo docker run -d -it -h “puppetlcient.localdomain” –add-host puppetmaster:172.17.0.1 –add-host puppet:172.17.0.1 –add-host puppetclient:172.17.0.3 --privileged --name puppetclient puppetclient /sbin/init**

**# sudo docker exec -it puppetclient /bin/bash**

**# puppet agent -tv**



|  |
| --- |
| **Part 2: Node Definition (NTP)** |

The main objective in this part of the lab is to create a new module and put a node definition in the main manifest to assign specific configurations to specific node.

Puppet begins compiling a catalog with either a single manifest file or a directory of manifest files that are interpreted as if they were a single file. The **primary manifest**, often known as the **site manifest**, is the beginning point.

The manifest you pass as a parameter on the command line is used by the **puppet apply** command:

***puppet apply /etc/puppetlabs/code/environments/production/manifests/site.pp***

You can give Puppet a **single.**pp file or a **directory of.pp files** to apply. Puppet apply uses the manifest you supply it, not the manifest of an environment.

Puppet server utilizes the main manifest set by the current node's environment. By **default**, the main manifest for an environment is **<ENVIRONMENTS DIRECTORY>/<ENVIRONMENT>/manifests**, which is for example **/etc/puppetlabs/code/environments/production/manifests**.

1. Use the following command to show the **modulepath**.

**# puppet config print manifest --section server --environment production**

|  |
| --- |
| <Insert screen capture of results> |

|  |
| --- |
| <Insert screen capture of results> |

**Question:** What is the command to show the **certname**?

|  |
| --- |
| <Insert screen capture of results> |

1. The **puppet apply** command and the primary server service both get the majority of their content from modules in one or more folders. The **modulepath** is a list of directories where Puppet looks for modules. The current node's environment determines the **modulepath**. **# puppet config print | grep -i module**

|  |
| --- |
| <Insert screen capture of results> |

1. Navigate to the module path. **#cd /etc/puppetlabs/code/environments/production/modules**

**# ls -lrt**

|  |
| --- |
| <Insert screen capture of results> |

1. The command to perform all puppet module related operation is puppet module.

# **puppet module --help**

|  |
| --- |
| <Insert screen capture of results> |

1. Create a new module named **ntpmodule**.



1. Before declaring this class in the main manifest, ensure no syntax error.

**# puppet parser validate init.pp**

|  |
| --- |
| <Insert screen capture of results> |

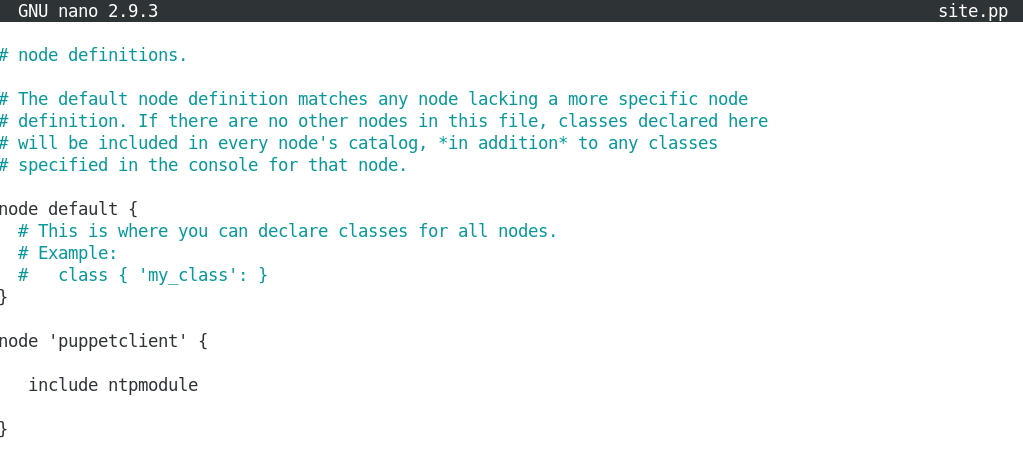
1. Find the agent managed by master and create a node definition and declare the class there.

**# sudo /opt/puppetlabs/bin/puppetserver ca list –all**

|  |
| --- |
| <Insert screen capture of results> |

1. Navigate to main manifest directory and declare a **ntpmodule** class inside **site.pp** for the node identified in the previous step. **# cd /etc/puppetlabs/code/environments/production/manifests**

**# nano site.pp**



1. Perform a syntax on the **site.pp** file.

**# puppet parser validate site.pp**

|  |
| --- |
| <Insert screen capture of results> |

1. On the agent (node), verify that the chrony service is not installed yet.

**# systemctl status chrony**

|  |
| --- |
| <Insert screen capture of results> |

1. Perform a smoke test.

**# puppet agent -tv –noop**

|  |
| --- |
| <Insert screen capture of results> |

1. Perform a puppet run.

**# puppet agent -tv**

|  |
| --- |
| <Insert screen capture of results> |

|  |
| --- |
| **Do it yourself** |

1. Create a manifest **ngix** to deploy the package to the puppet agent (node) and set the content of the **index.html** to “**Hello DV1C04!**”.

|  |
| --- |
| <Insert screen capture of results> |

|  |
| --- |
| **Part 2: Working with Bash Aliases** |

The main objective in this part of the lab is to create bash aliases so you can be more productive on the command line.

When working on the command line, bash aliases are essentially shortcuts that can save you from having to remember long commands and eliminate a lot of typing. For example, you could set the alias **tgz** to be a shortcut for the **tar -xvfz** command.

1. Simply run the alias command to see a list of defined aliases on your profile.

**# alias**

|  |
| --- |
| <Insert screen capture of results> |

1. Set a new alias **cdpp**.

**# alias cdpp ="cd $(puppet config print manifest)"**

|  |
| --- |
| <Insert screen capture of results> |

1. Try out the new alias.

**# cdpp**

|  |
| --- |
| <Insert screen capture of results> |

1. To make the alias persistent you need to declare it in the ~/.bash\_profile or ~/.bashrc file.

|  |
| --- |
| <Insert screen capture of results> |

**References**<https://puppet.com/docs/puppet/7/dirs_manifest.html#:~:text=By%20default%2C%20the%20main%20manifest,the%20default%20for%20all%20environments>.

<https://twpower.github.io/178-run-container-as-privileged-mode-en>

Puppet 6 Essentials By Andrew Mallett

**End of Lab Exercise --**