**Lab 08**

**Laboratory Exercise**

**LAB EXERCISE**

**Time to Complete**

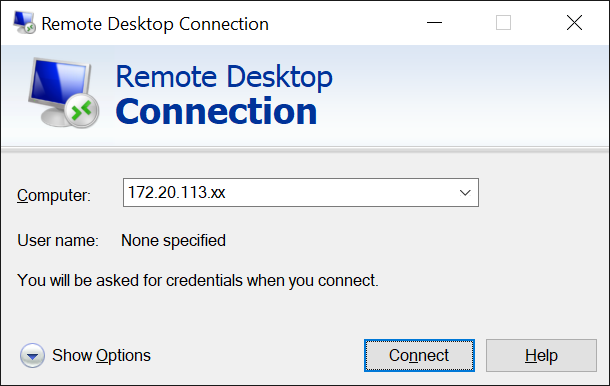
Approximately 45 Minutes

**What You Need**

* You need to complete lab 6 exercises so that 2 new Puppet Clients are setup for use in this lab exercise.

From your machine logged-in to RP VPN, run Remote Desktop Connection to connect to the ubuntu Linux Virtual Machine (VM). Please login based on your assigned VM as shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Name** | **VM** | **IP Address** | **User Name** | **Password** |
| 1 | LEOW TANG QING | PDC2-Ubuntu-03 | 172.20.113.184 | dockeradm | docker!2 |
| 2 | LIM SHEN HUI | PDC2-Ubuntu-02 | 172.20.113.183 | dockeradm | docker!2 |
| 3 | LIN JINGZHOU | PDC2-Ubuntu-04 | 172.20.113.185 | dockeradm | docker!2 |
| 4 | MUHAMMAD FAISAL BIN SHAIK HASSAN | PDC2-Ubuntu-05 | 172.20.113.186 | dockeradm | docker!2 |
| 5 | LYNN LEE QING XIA | PDC2-Ubuntu-06 | 172.20.113.187 | dockeradm | docker!2 |
| 6 | NG CHEE KIONG | PDC2-Ubuntu-07 | 172.20.113.188 | dockeradm | docker!2 |
| 7 | PARAMASIVAM S/O VANNU GOPAL | PDC2-Ubuntu-08 | 172.20.113.189 | dockeradm | docker!2 |
| 8 | SAHLATUL-FARIHAH BINTE M ASARI | PDC2-Ubuntu-09 | 172.20.113.190 | dockeradm | docker!2 |
| 9 | SHAIKH FAID BIN OMAR | PDC2-Ubuntu-10 | 172.20.113.191 | dockeradm | docker!2 |
| 10 | CHIU JING XIONG | PDC2-Ubuntu-11 | 172.20.113.192 | dockeradm | docker!2 |
| 11 | KELLY WONG XUE YEE | PDC2-Ubuntu-12 | 172.20.113.193 | dockeradm | docker!2 |
| 12 | LIM SI YING | PDC2-Ubuntu-13 | 172.20.113.194 | dockeradm | docker!2 |
| 13 | LIN LI YI | PDC2-Ubuntu-14 | 172.20.113.195 | dockeradm | docker!2 |
| 14 | MUHAMMAD MUQTADIR BIN SADIQ BASHA | PDC2-Ubuntu-15 | 172.20.113.196 | dockeradm | docker!2 |
| 15 | NUR HIDAYAH BTE RAMLEE | PDC2-Ubuntu-16 | 172.20.113.197 | dockeradm | docker!2 |
| 16 | NUR NADIA ASHBOLLAH BINTE | PDC2-Ubuntu-17 | 172.20.113.198 | dockeradm | docker!2 |
| 17 | NUR THAQIFAH AQILAH BINTE JURAIMI | PDC2-Ubuntu-18 | 172.20.113.199 | dockeradm | docker!2 |
| 18 | RAUDHATUNNISHA BTE RAMLI | PDC2-Ubuntu-19 | 172.20.113.200 | dockeradm | docker!2 |
| 19 | SITI NUR ALYSHYIA BINTE HASHIM | PDC2-Ubuntu-20 | 172.20.113.201 | dockeradm | docker!2 |
| 20 | TAN TEE BING | PDC2-Ubuntu-21 | 172.20.113.202 | dockeradm | docker!2 |



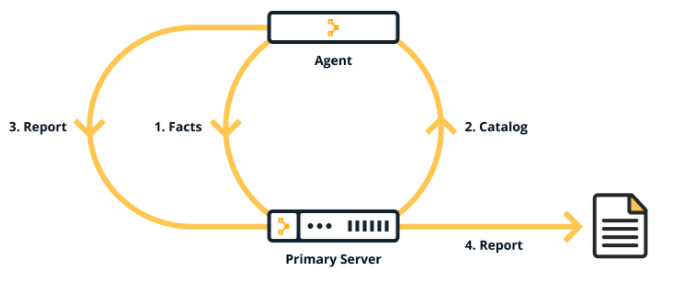
Replace xx with the IP address of the VM that you have been assigned.

**Part 1: Facts and Facter**

1. In this lab, you will use facter and facts for server inventory

* Facter and facts are already in the Puppet master node and Puppet agent nodes.

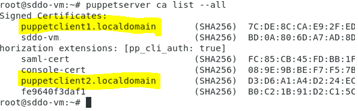
The following shows the relation between puppet master (primary server) and puppet agent.



* 1. As **root,** check that the 2 agent containers puppetclient1.localdomain and puppetclient2.localdomain are registered with puppet master

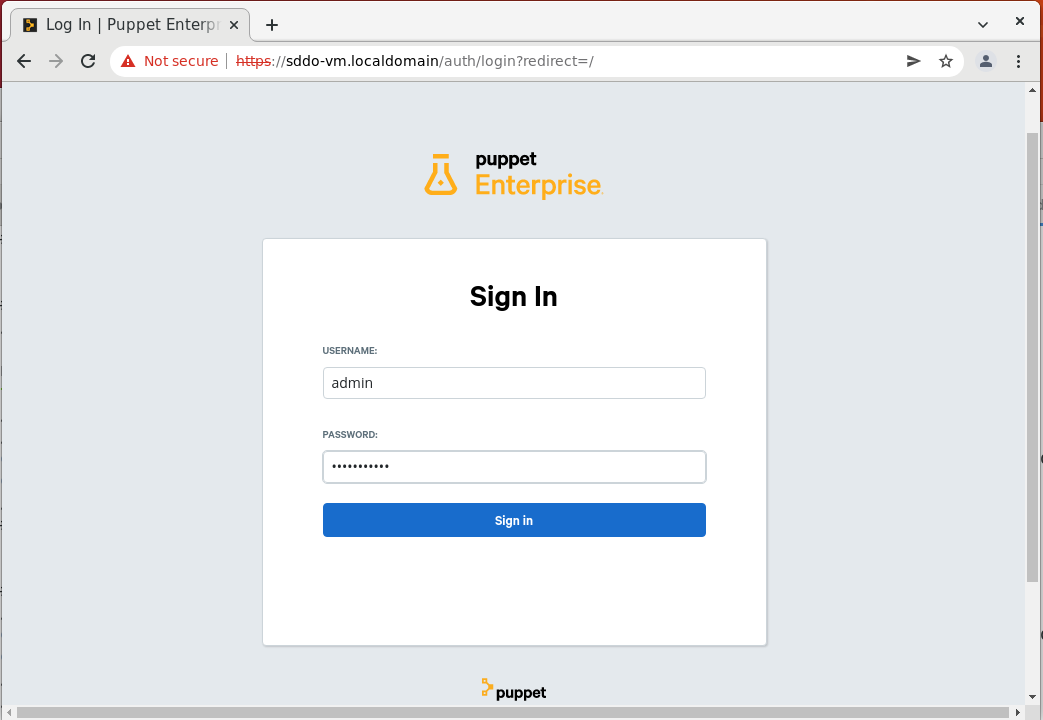
Open **Terminal** and enter the following command to login as a superuser.

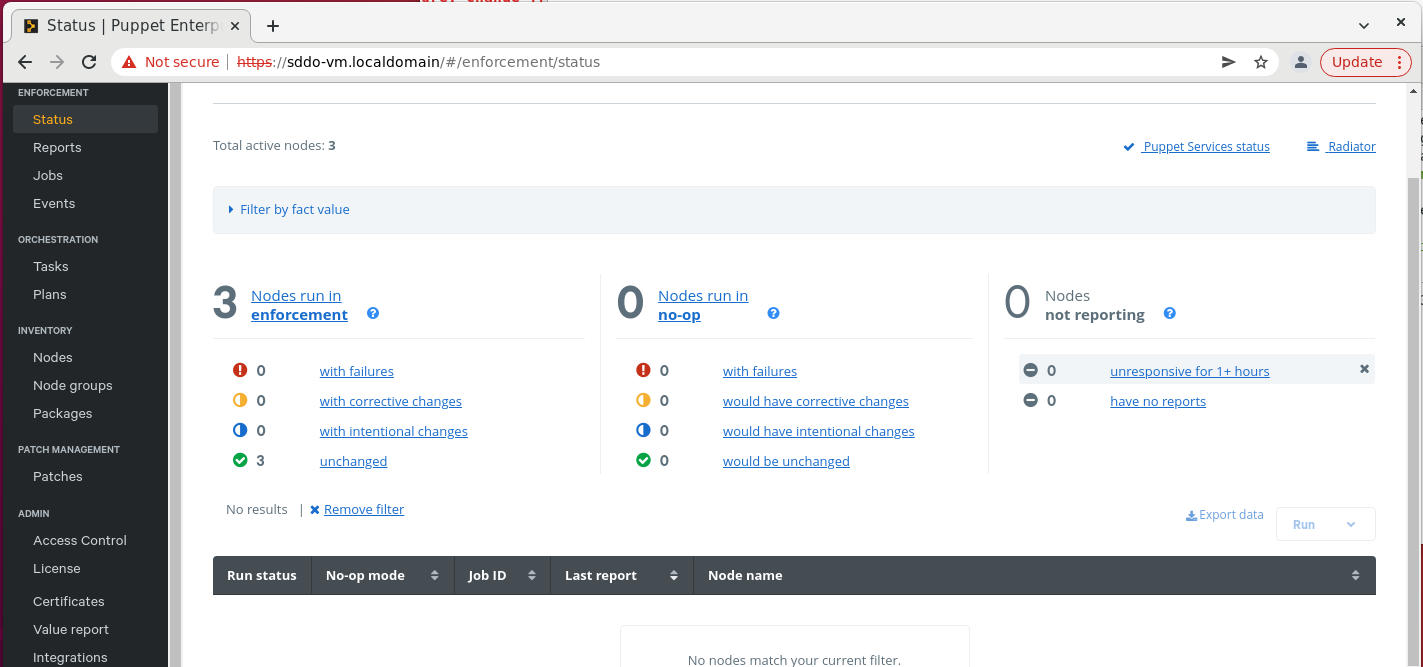
root@sddo-vm: **su - root**

root@sddo-vm: **puppetserver ca list --all  
**The above shows the 2 new agent nodes (created during lab 6) are registered with puppet master.

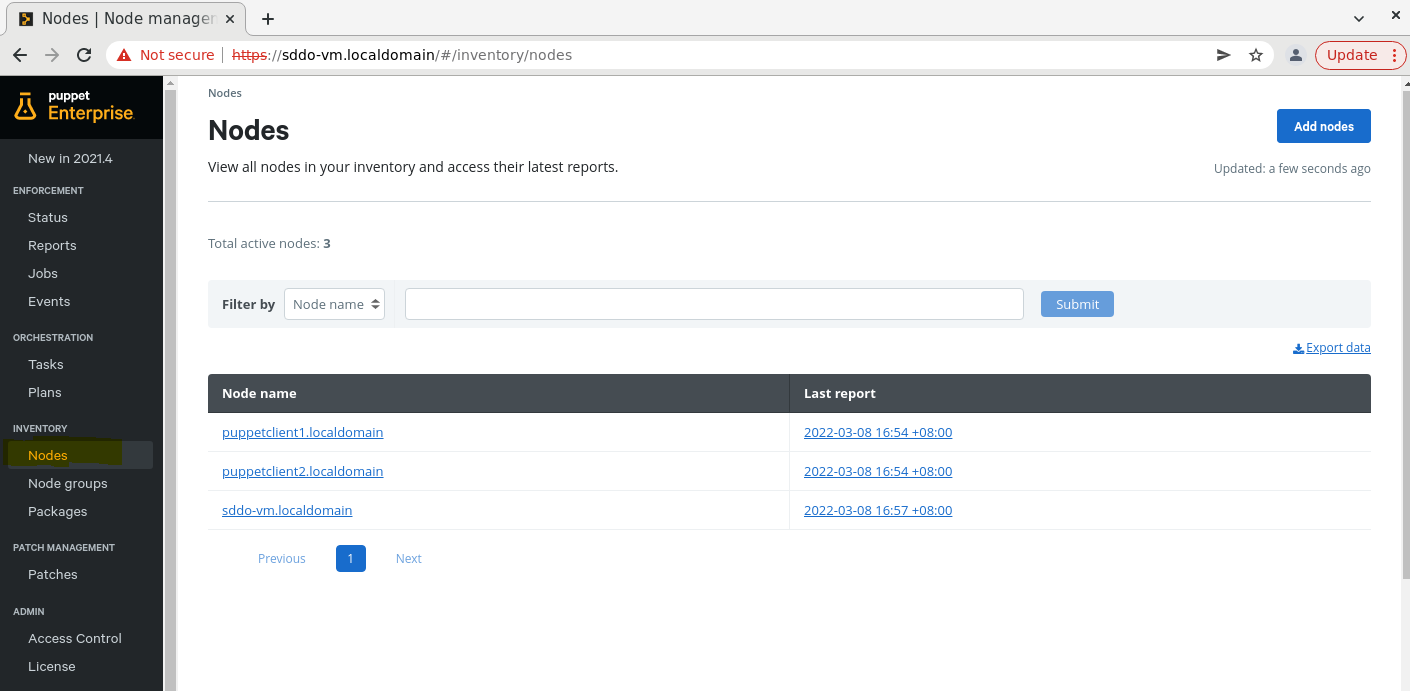
1.2 Accessing the console  
  
The console is the web interface for Puppet Enterprise.

Open a web browser (e.g. Google Chrome), enter **https://localhost**. Puppet Enterprise console will be shown.   
  
Enter username: **admin**, password: **password1234** to sign in.

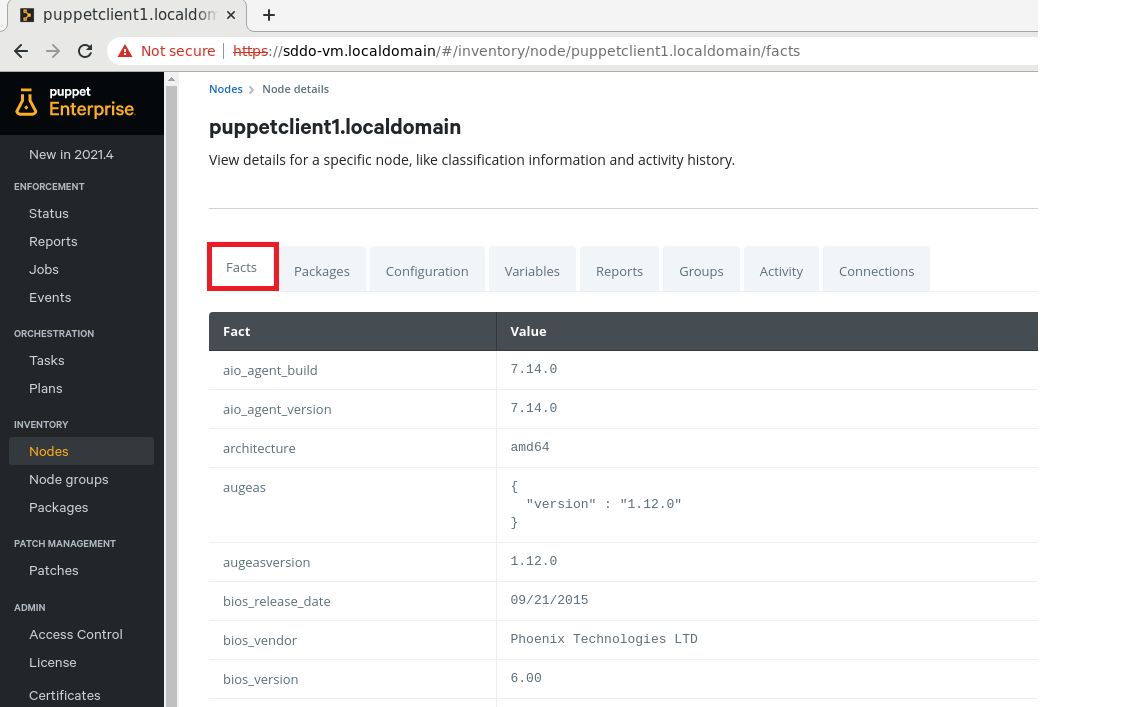




1.3 Click on Nodes



1.4 Click on puppetclient1.localdomain. By default, Facts is selected. Facts collected when puppet agent runs are recorded on puppet master.

****

1.5 Access the other puppet node i.e. puppetclient2.localdomain and view the facts of the node

1. **Work on facts using *facter* command**

Facts are information collected about a node and reported to the Puppet master on each run. Facts are collections of system information such as timezone, IP address, hostname, cpu temperature, uptime, disk space, installed applications, running daemons or services, and many other things.

Facter is Puppet's system inventory tool. Facter discovers facts intrinsic to a node (such as its hostname, network interfaces and operating system, timezone etc.).Facter has a large number of built-in facts.

2.1 Access a puppet node e.g. puppetclient1.localdomain

How do you access the container puppetclient1.localdomain?

You may refer to lab 6.

2.2 Issue **puppet facts** command on a puppet node e.g. puppetclient1.localdomain

root@puppetclient1.localdomain: **puppet facts**

Paste the output of the command.

2.3 Issue **puppet facts** command for a particular fact

root@puppetclient1.localdomain: **puppet facts os**

Paste the output of the command.

2.4 Issue **facter** command

root@puppetclient1.localdomain: **facter**

Paste the output of the command.

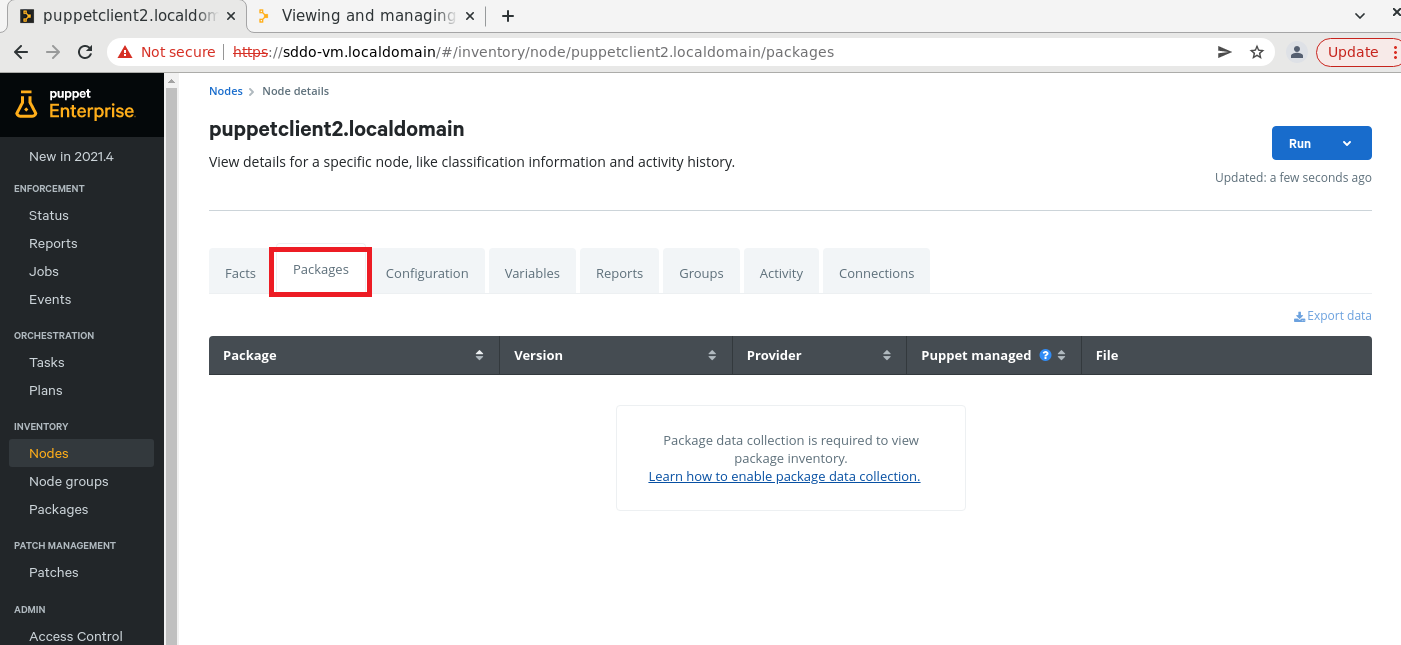
2.5 Issue **facter** command for a particular fact

root@puppetclient1.localdomain: **facter os**

Paste the output of the command.

**Part 3: Packages**

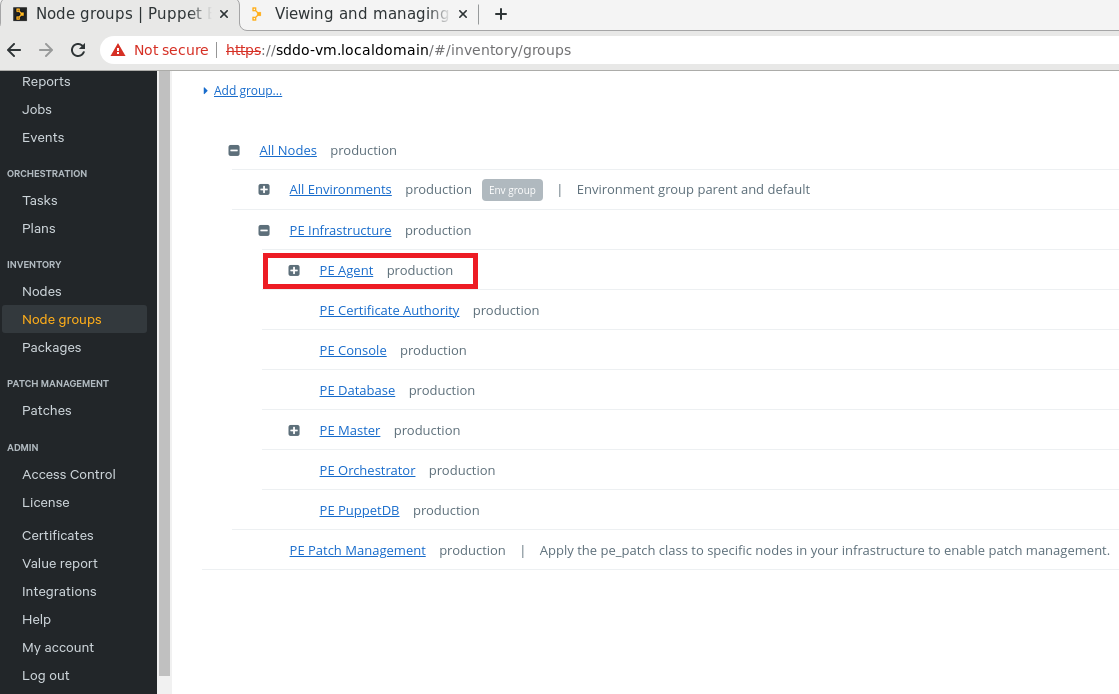
3. Package data collection is disabled by default, so the **Packages** page in the console initially appears blank.

****

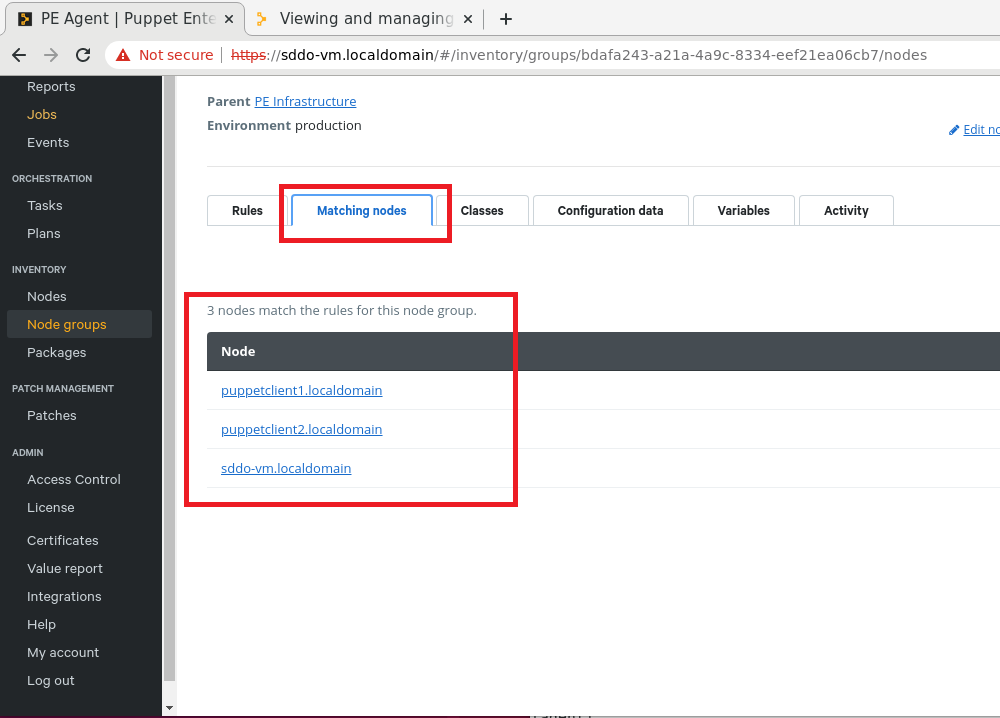
3.1 In order to view a node's current package inventory, enable package data collection.

You can choose to collect package data on all your nodes, or just a subset. Package inventory reporting is available on nodes.

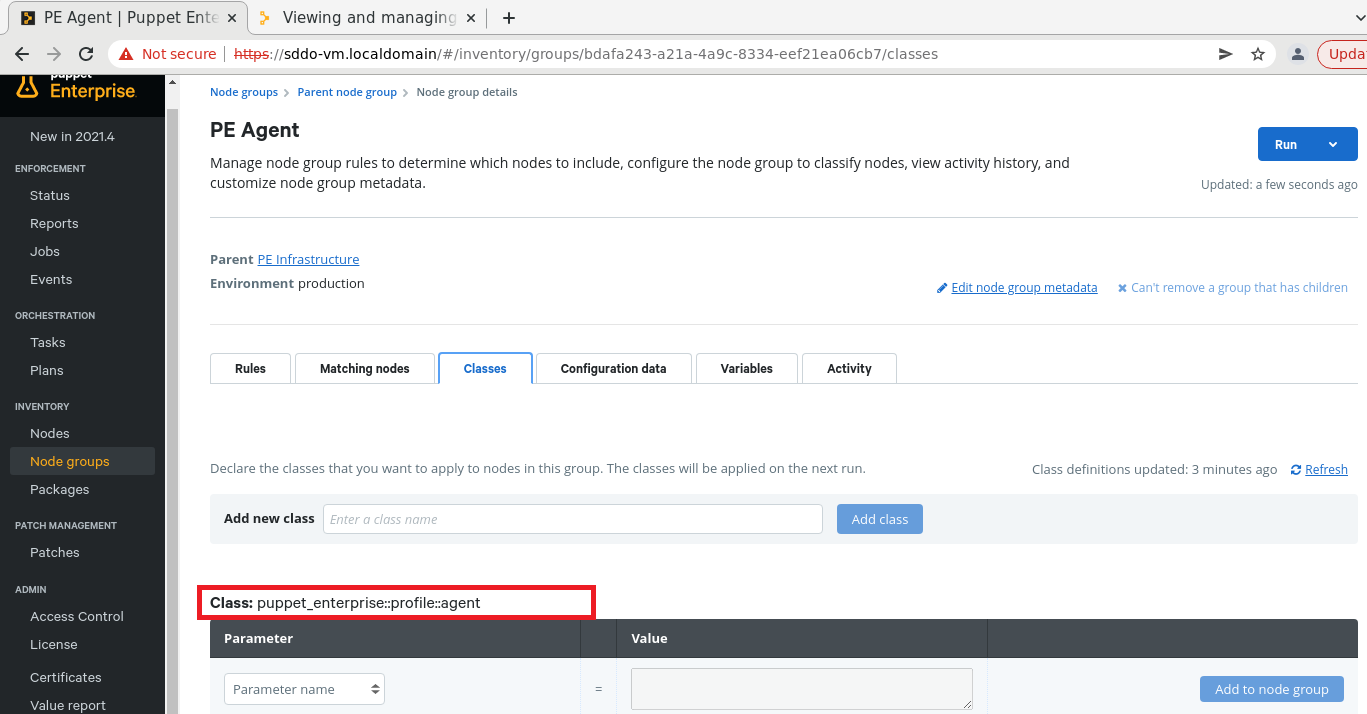
* + 1. In the console, click **Node groups**.
  + If you want to collect package data on all your nodes, click the **PE Agent** node group.

****

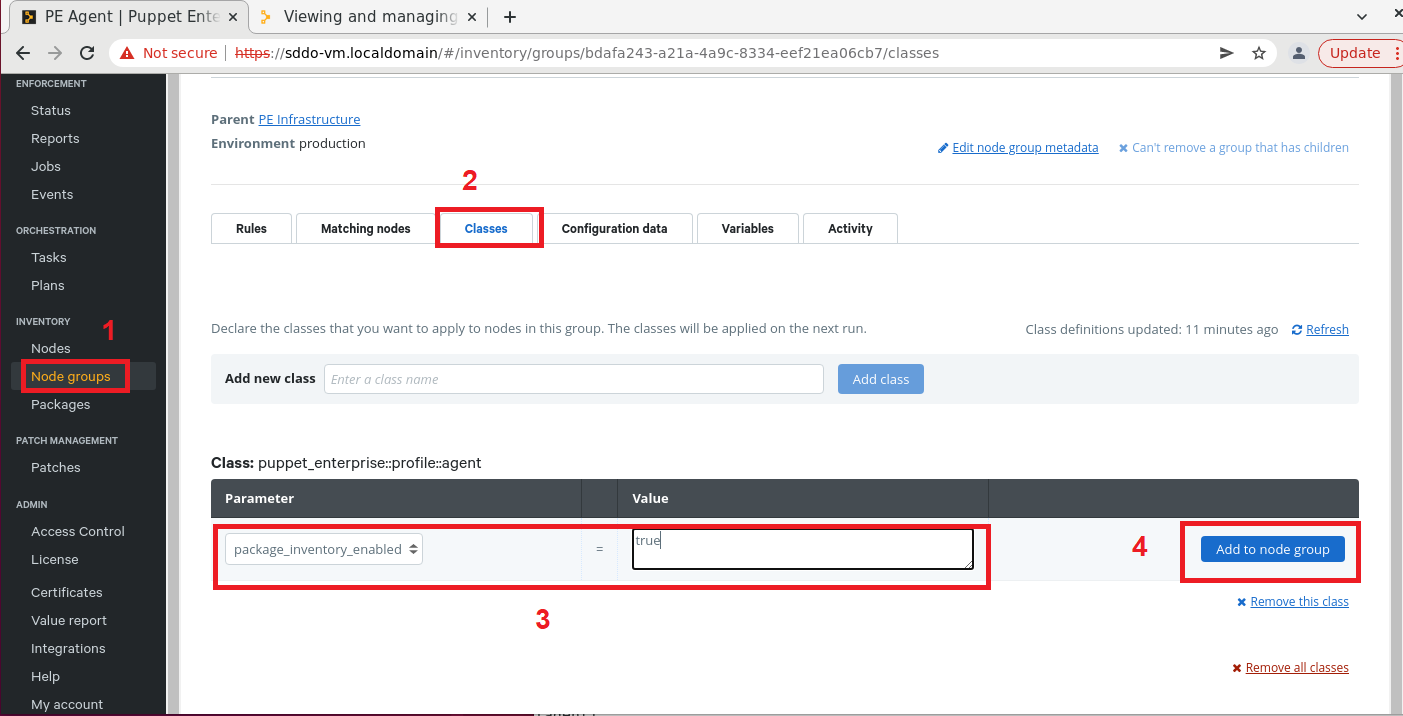
* + 1. Click on Matching Nodes and check all the agent nodes are there.

****

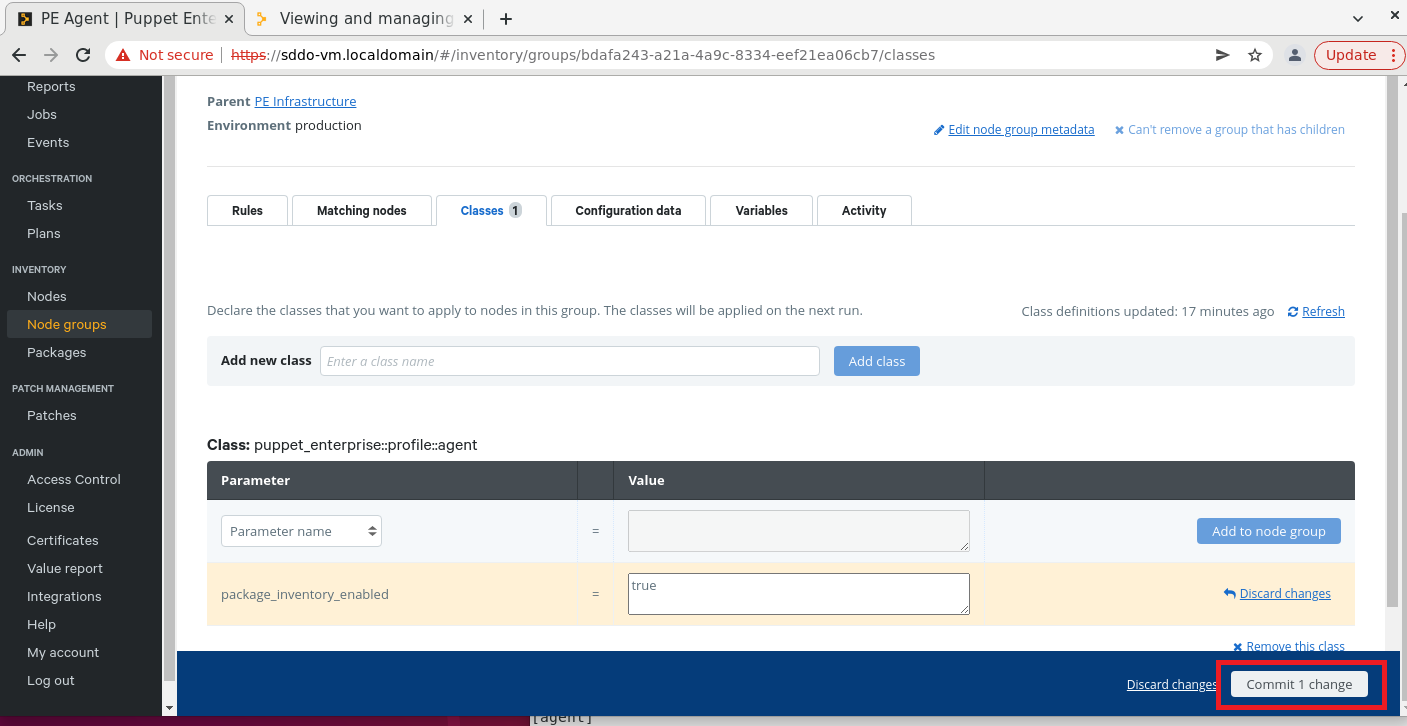
* + 1. Click **Classes**. In the **Add new class** field, select **puppet\_enterprise::profile::agent** and click **Add class**.



* + 1. In the **puppet\_enterprise::profile::agent** class, set the **Parameter** to **package\_inventory\_enabled** and the **Value** to true. Click **Add parameter**, and commit changes.

****

Click on “Commit 1 change”

****

3.1.5 On the node's inventory page, click **Facts** and locate **puppet\_inventory\_metadata** in the list.

The fact value looks something like:

{

"packages" : {

"collection\_enabled" : true,

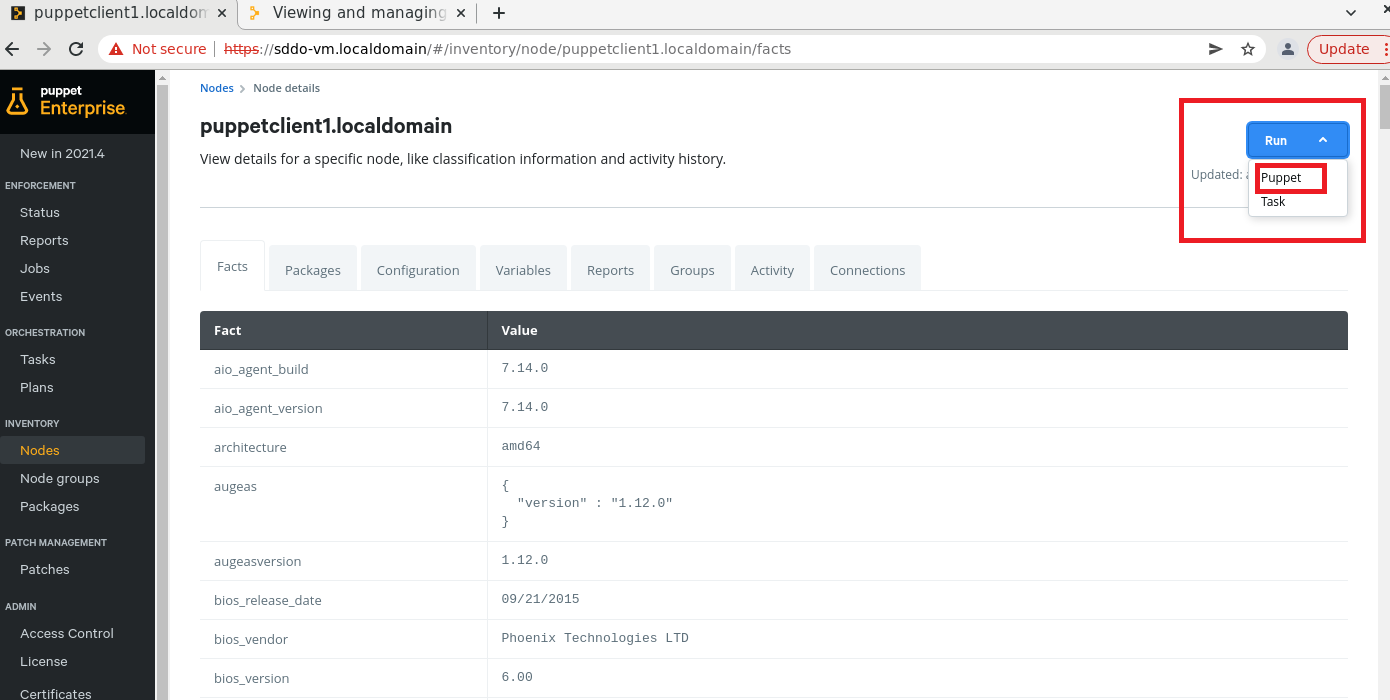
"last\_collection\_time" : "1.8121s"

}

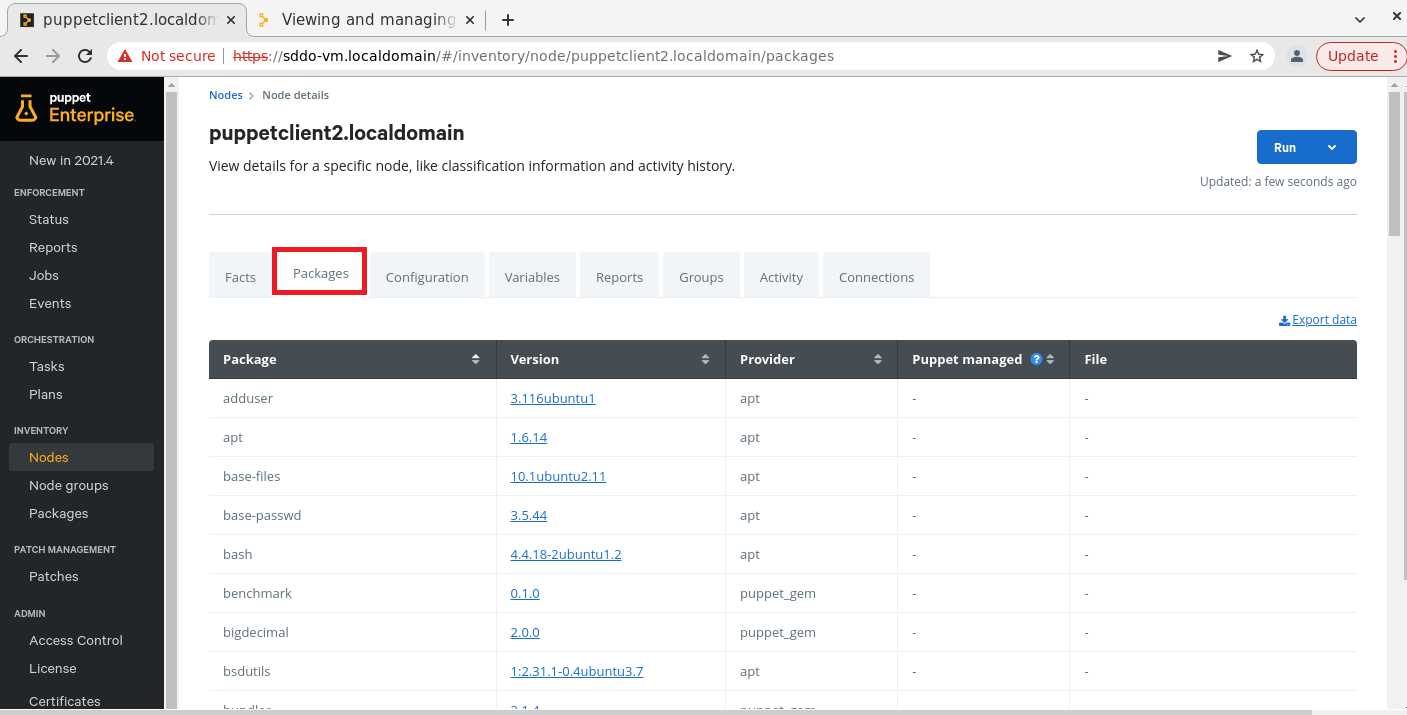
}

* + 1. Run Puppet to apply these changes to the nodes in your node group.

Puppet enables package inventory collection on this Puppet run, and begins collecting package data and reporting it on the **Packages** page on each subsequent Puppet run.

****

* + 1. Run Puppet a second time to begin collecting package data, then click **Packages**.

****

1. **Work on packages using *puppet resource package* command**
   1. Access bash shell of puppet node e.g. puppetclient1.localdomain

(How do you access the container puppetclient1.localdomain?

You may refer to lab 6.)

* 1. On a puppet node, to list the installed package

root@puppetclient1: puppet resource package

*(you may use* ***apt list - -install*** *to check the packages*)

Paste your screenshot of above command output:

* 1. On a puppet node, to list one particular package

root@puppetclient1: puppet resource package < package name>

*(you may use* ***apt list - -install | grep <package name>*** *to check the package*)

Paste your screenshot of command output for **puppet resource package vim**

* 1. On a puppet node, to remove one particular package

root@puppetclient1: puppet resource package < package name> ensure=absent

*(you may use* ***apt list - -install | grep <package name>*** *to check the package*)

Paste your screenshot of command output for **puppet resource package vim ensure=absent**

* 1. On a puppet node, to install one particular package

root@puppetclient1: puppet resource package < package name> ensure=present

*(you may use* ***apt list - -install | grep <package name>*** *to check the package*)

Paste your screenshot of command output for **puppet resource package vim ensure=present**

**--End of Lab Exercise --**