Yunfei Zhou

s3598797

USAP assignment 2

Report

Contents

[Prepare the environment for this project 2](#_Toc495671241)

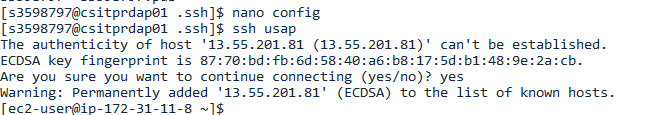
[Task implements 5](#_Toc495671242)

# Prepare the environment for this project

1. Login to RMIT titan.csit.rmit.edu.au
2. In the case, I need a SSH key for login to AWS instance, then I create a SSH key by ssh-keygen -t rsa -b 4096 which provide in <http://titan.csit.rmit.edu.au/~e20925/usap/AWS/setup1.mp4>
3. Go to AWS and import the key that we just generated (I marked the key name as my student id that is s3598797).
4. Launch an instance for the assignment
   1. Instance will be Red Hat Enterprise Linux 7.4.
   2. The instance type is “t2.micro”.
   3. Leave the storage as 10 GB.
   4. Create a security group and expose the port 22, 80,443 to the public.
      1. expose the port 22, 80,443 means that I can access the SSH service, http/web service and https service from any machine in the world.
   5. Launch the instance.
   6. Go to Elastic IP and allocate new address for my instance. The new address is 13.55.201.81, then I associated the address with my instance i-0a6a603ec4d1449ec
5. Modify my SSH host setting by following code

host usap

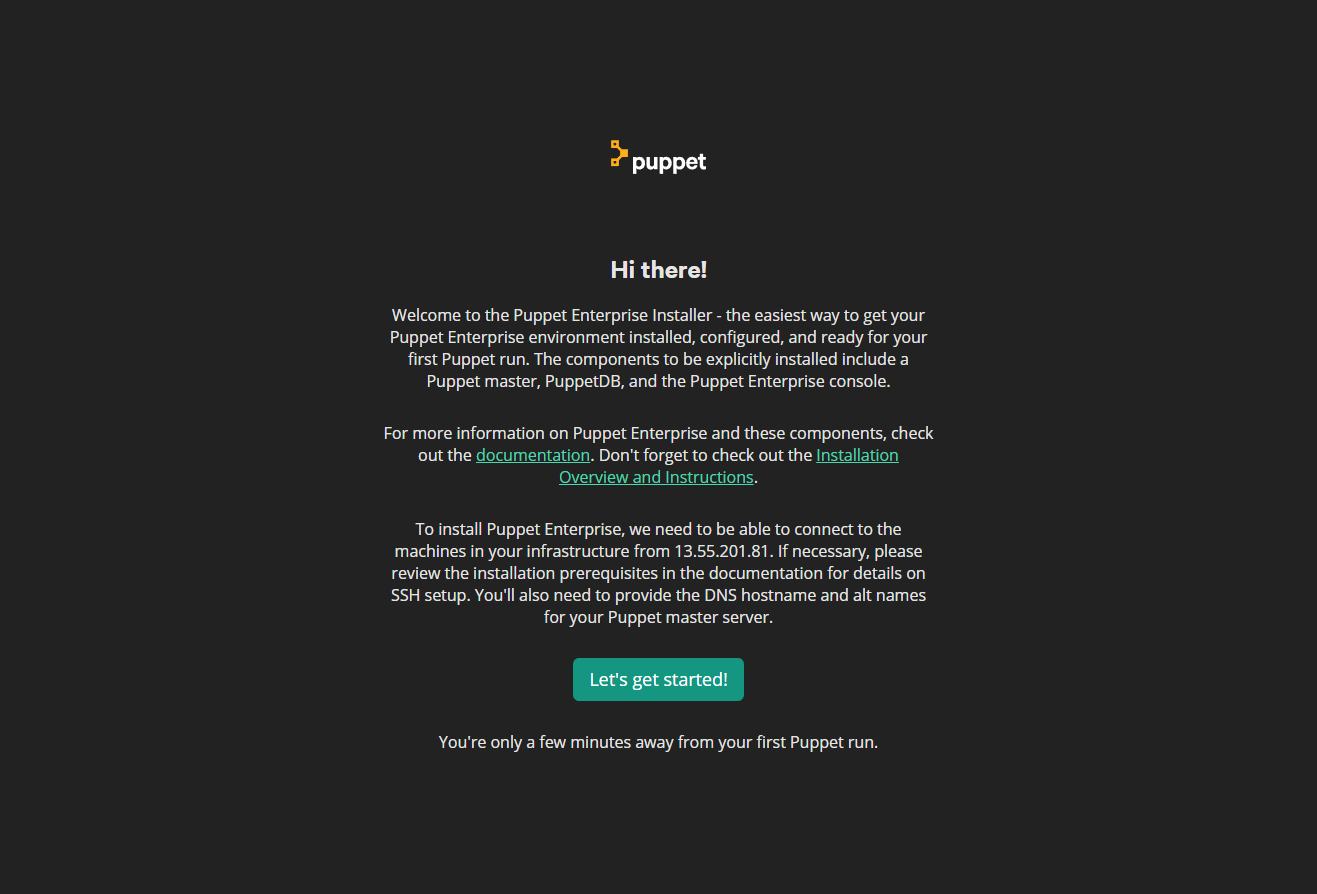
HostName 13.55.201.81  
 User ec2-user  
 IdentityFile ~/.ssh/s3598797



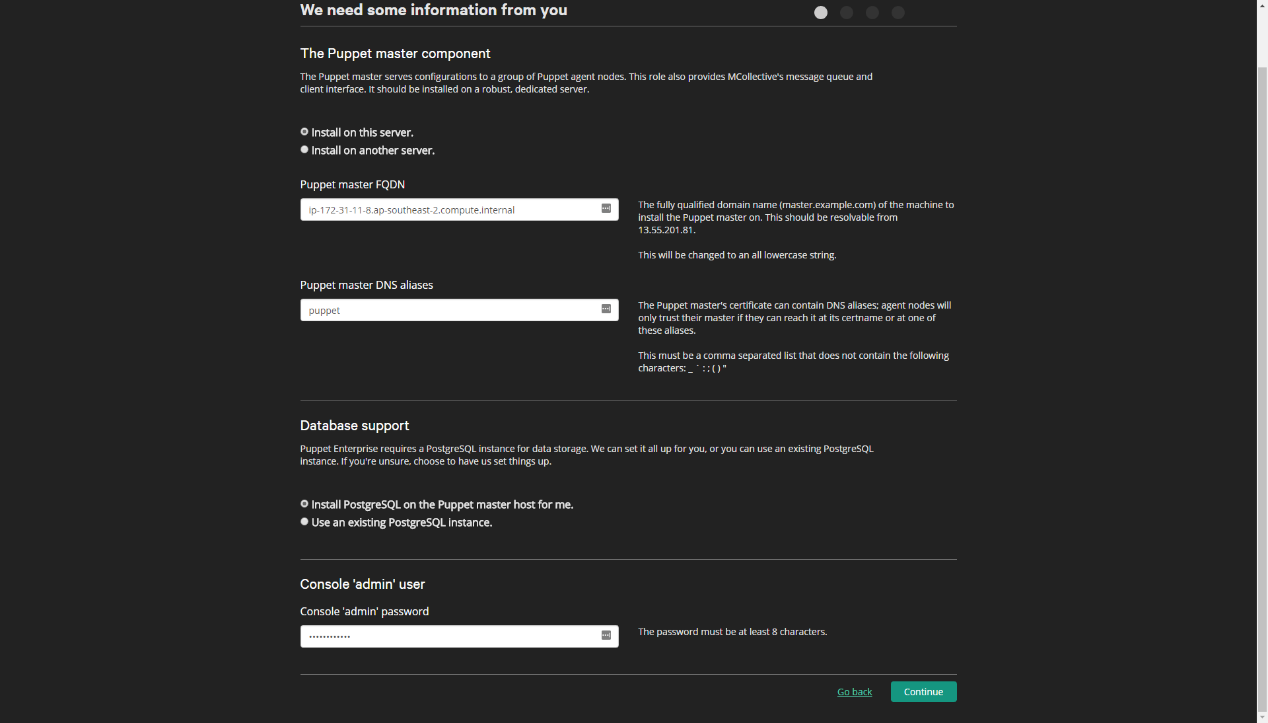
1. SSH to my AWS instance with “ssh usap” command
2. Use “sudo yum -y update”
   1. It will update all current package and installed ‘application’ to latest version.
3. Install wget
   1. I need to download the puppet2017 from RMIT serve, but I get the error from the system. It means that the system does come with wget.
   2. sudo yum install -y wget
   3. It will download and install wget from RHEL server.



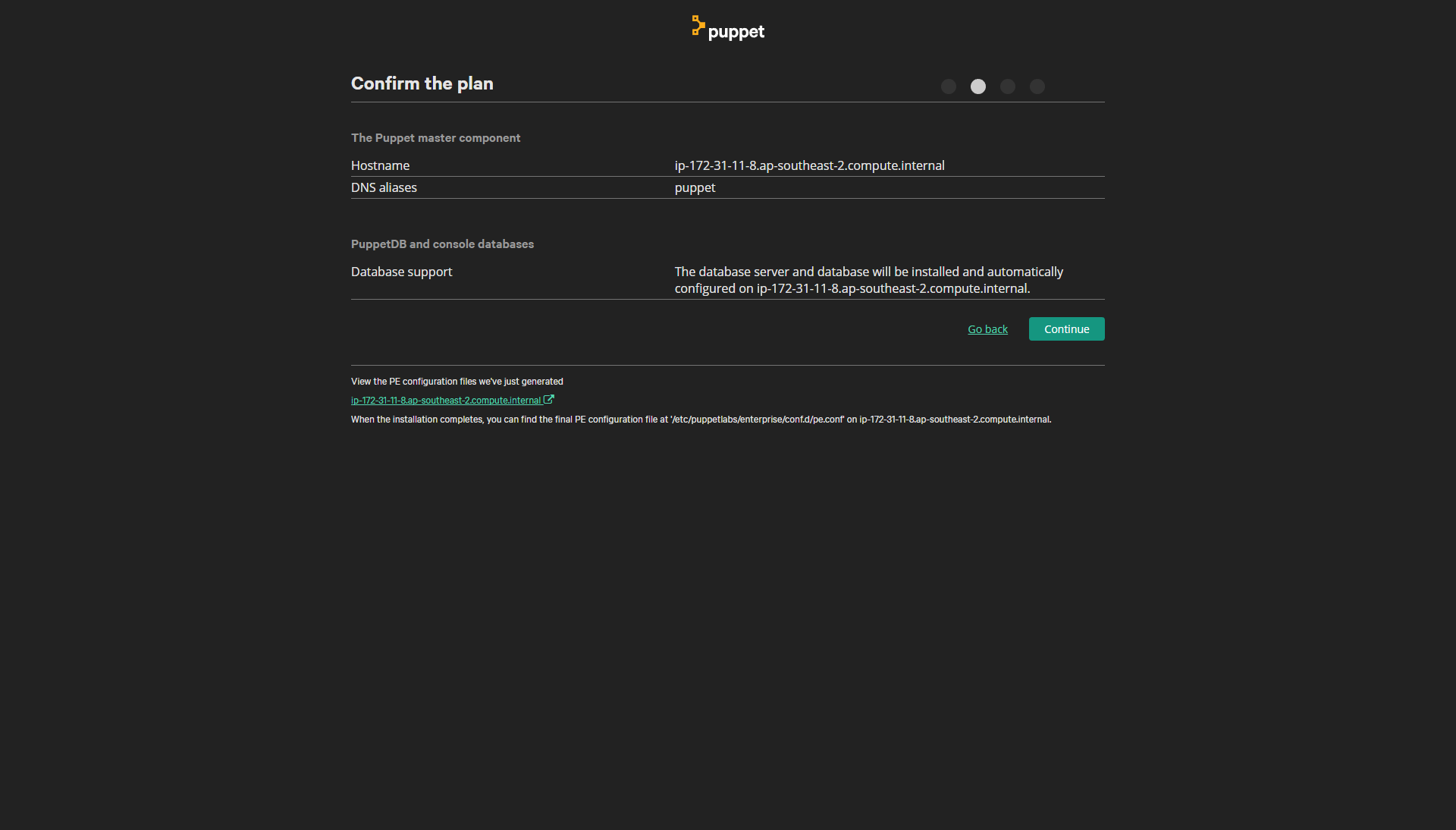
1. Download and install puppet 2017
   1. Download the install package by using “wget <http://titan.csit.rmit.edu.au/~e20925/usap/puppet2017.tar.gz>” command (in face, RMIT server was too slow, so I download the package from the official server by using “wget <https://s3.amazonaws.com/pe-builds/released/2017.2.4/puppet-enterprise-2017.2.4-el-7-x86_64.tar.gz>” command )
   2. Use “tar -zxvf puppet-enterprise-2017.2.4-el-7-x86\_64.tar.gz” to unzip the installation package.
2. Install puppet
   1. cd puppet-enterprise-2017.2.4-el-7-x86\_64
   2. sudo ./puppet-enterprise-installer
   3. proceed with “Guided install” mode (I select 1)
   4. go to security groups and expose the port 3000 for guide installation.
   5. Open a web browser and access <https://13.55.201.81:3000/>



* 1. Input all the basic setting
     1. Puppet master FQDN is ip-172-31-14-227.us-west-2.compute.internal
     2. Puppet master DNS aliases is: puppet
     3. Console 'admin' password: 2015Love1005!!



* 1. Confirm the plan



* 1. Install agent
     1. wget -O - -q --no-check-certificate --secure-protocol=TLSv1 https://35.163.156.5:8140/packages/current/install.bash | sudo bash

# Task implements

1. Task 1
   1. Set a config file that name is “users\_management.pp”

Before everything start, I need install the csh for Fred, then I created group sysadmin, cars, trucks and ambulances.

#groups management

group { 'sysadmin':

ensure => present,

}

group { 'cars':

ensure => present,

}

group { 'trucks':

ensure => present,

}

group { 'ambulances':

ensure => present,

}

Just in case, some of the Linux system set the users’ home under ‘/User/’, then I created the Becca, freed and Wilma’s home directory

file { '/home/Becca':

ensure => directory,

owner => becca,

}

file { '/home/fred':

ensure => directory,

owner => fred,

}

file { '/home/wilma':

ensure => directory,

owner => wilma,

}

* 1. Add password for each users
     1. By using openssl -1 to generate the encrypted password (the password is: ‘password’)
     2. Set password as ‘password => '$1$fAVb8Rx5$dPRNhLgso7wtdG6b8QaF4/',’
     3. Set groups and password for each users

#users Management

user { 'becca':

ensure => present,

home => '/home/becca',

uid => '10018797',

shell => '/bin/bash',

password => '$1$fAVb8Rx5$dPRNhLgso7wtdG6b8QaF4/',

groups => ['sysadmin','cars'], # set groups for becca

}

user { 'fred':

ensure => present,

home => '/home/fred',

uid => '10028797',

shell => '/bin/csh',

password => '$1$fAVb8Rx5$dPRNhLgso7wtdG6b8QaF4/',

groups => ['trucks','cars'], # set groups for fred

}

user { 'wilma':

ensure => present, # make sure this user will be created by puppet

home => '/home/wilma', # set home directory

uid => '10038797', # default user id 1003+ my last 4 student id

password => '$1$fAVb8Rx5$dPRNhLgso7wtdG6b8QaF4/', # managed the user's password

groups => ['trucks','cars','ambulances'], # set groups for wilma

}

* 1. set the generated SSH key for wilam
     1. create .ssh folder for Wilma and inserted private key into the

file { '/home/wilma/.ssh':

ensure => directory,

owner => wilma,

}

ssh\_authorized\_key{ 'wilma\_SSH':

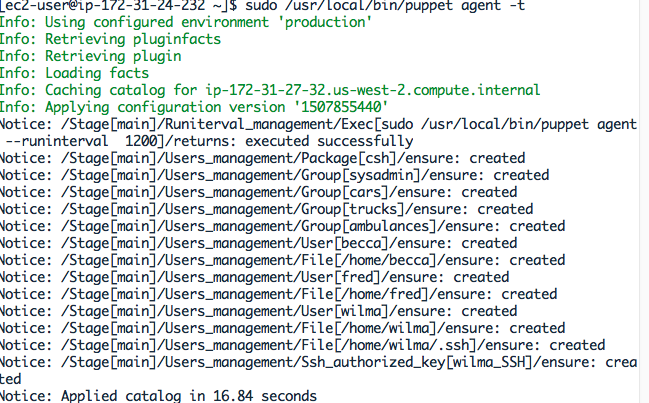
user => 'wilma',

type => 'ssh-rsa',

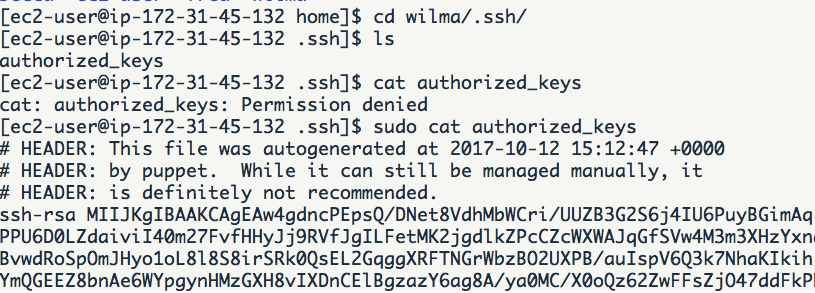
key => 'MIIJKgIBAAKCA',

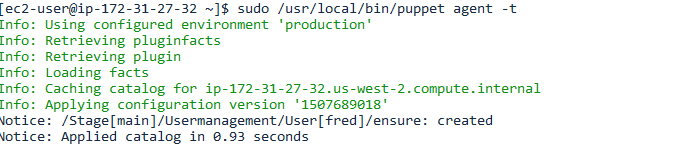
}

**First run results:**

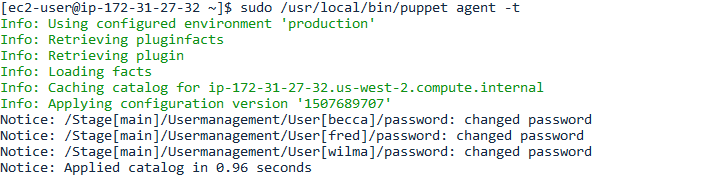
****

**Result for SSH**





**Password change test and result:**



**Test:**

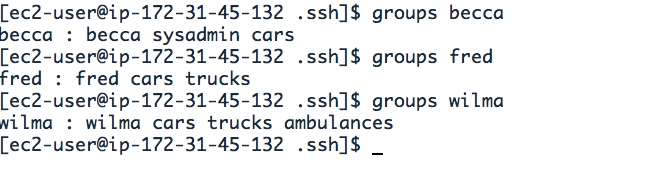
**Input with correct password for Becca**



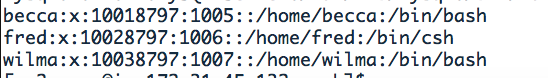
Input with incorrect password for Wilma



**Group set result:**



**User ID and shall result:**

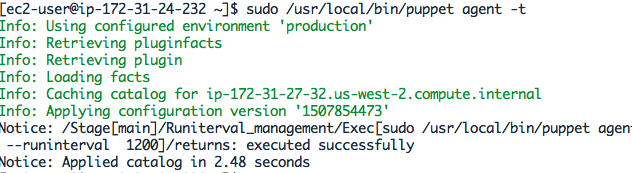


1. Task 2 set the check in time to 20 minutes (3 times in a hour)

exec { 'sudo /usr/local/bin/puppet agent --runinterval 1200':

path => ['/usr/bin', '/usr/sbin','/usr/local/bin'],

**Result:**



1. Task3 install packages and manage the services
   1. Due to the lynx is on optional channel, I need to enable the optional channel for my rest installation before I start.

exec{ 'sudo /usr/bin/yum-config-manager --enable rhui-REGION-rhel-server-extras rhui-REGION-rhel-server-optional && exit':

path => ['/usr/bin', '/usr/sbin',],

}

* 1. start the installation

exec { 'sudo /usr/bin/yum update -y': # command this resource will run

path => ['/usr/bin', '/usr/sbin',],

}

#install the shell packages

package { 'openssh':

ensure => installed,

}

package { 'httpd':

ensure => installed,

}

package { 'php':

ensure => installed,

}

package { 'tigervnc-server':

ensure => installed,

}

package { 'tmux':

ensure => installed,

}

package { 'lynx':

ensure => installed,

}

package { 'gcc':

ensure => installed,

}

package { 'vim':

ensure => installed,

}

package { 'emacs':

ensure => installed,

}

package { 'git':

ensure => installed,

}

package { 'wget':

ensure => installed,

}

(Note: the httpd is apache)

* 1. Install special package the not include in the official repository

Download the package from the web site

#install dia2code

exec { 'sudo yum install -y libxml2-devel; wget http://prdownloads.sourceforge.net/dia2code/dia2code-0.8.3.tar.gz ; tar -xvf dia2code-0.8.3.tar.gz; cd dia2code-0.8.3; bash ./configure; sudo make; sudo install; cd ..; rm -rf dia2code-0.8.3; rm dia2code-0.8.3.tar.gz': # command this resource will run

path => ['/usr/bin', '/usr/sbin','/usr/local/bin'],

}

# install mysql-server

exec { 'wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm ; sudo rpm -Uvh mysql57-community-release-el7-11.noarch.rpm ; rm mysql57-community-release-el7-11.noarch.rpm': # command this resource will run

path => ['/usr/bin', '/usr/sbin','/usr/local/bin'],

}

package { 'mysql-server':

ensure => installed,

}

#install the epel for cgdb and sshfs

exec{ 'wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm; rpm -ivh epel-release-latest-7.noarch.rpm; rm epel-release-latest-7.noarch.rpm':

path => ['/usr/bin', '/usr/sbin',],

}

package { 'cgdb':

ensure => installed,

}

package { 'fuse-sshfs':

ensure => installed,

}

* 1. Ensure the services is on boot.

# ensure apache2 service is running

service { 'httpd':

ensure => running,

enable => true, # Make sure it will start on boot

}

# ensure mysql service is running

service { 'mysqld':

ensure => running,

enable => true, # Make sure it will start on boot

}

#ensure openssh is running

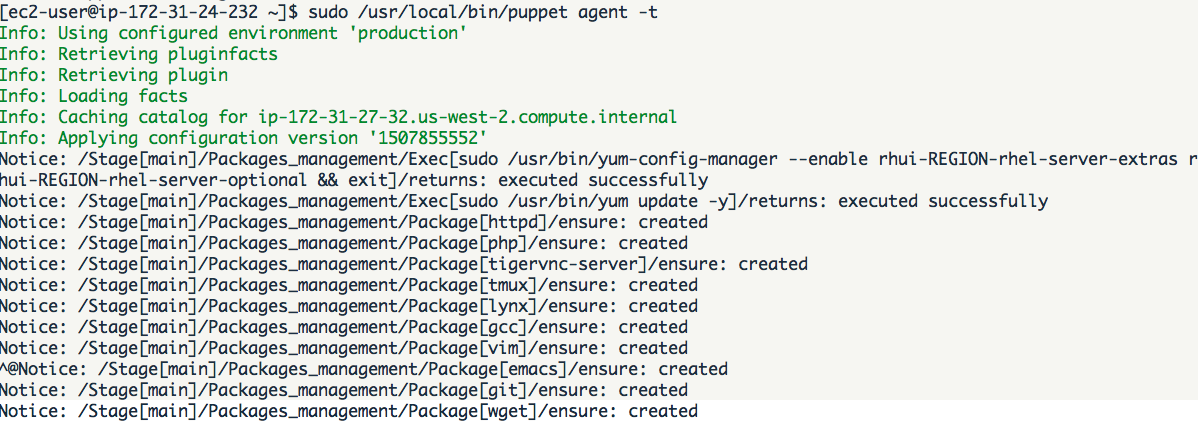
service { 'sshd':

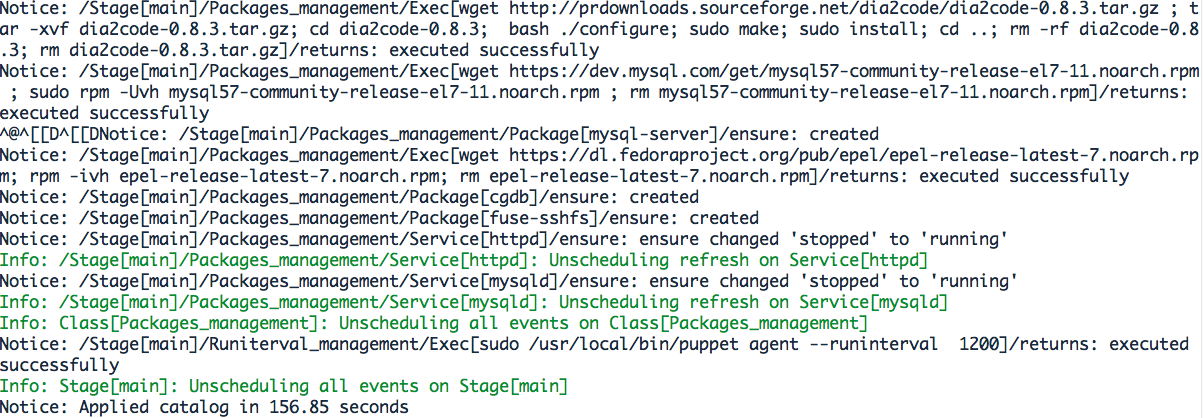
ensure => running,

enable => true, # Make sure it will start on boot

}

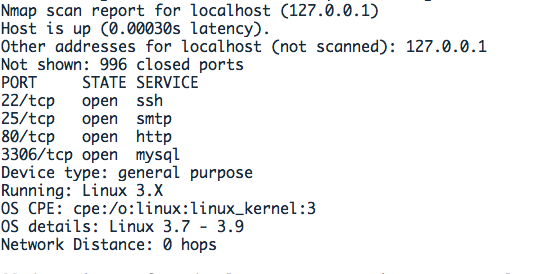
Results:





**Check running services results:**

sudo nmap -sT -O localhost



1. Task 4.
   1. Off-set the root login

Use Augeas function to locate and change the setting in sshd\_config, once the ‘PermitRootLogin’ is set as no, then root will not allow access by ssh

#a. disable root login for ssh

augeas { 'sshd\_config':

context => '/files/etc/ssh/sshd\_config',

changes =>[

'set PermitRootLogin no',

],

}

* 1. Change the apache document

First, I need make sure the ‘s3598797’ directory is existing under /var/www/. After that I am using Augeas to change the DocumentRoot from '/var/www/html' to '/var/www/s3598797'.

#b. disable root login for ssh

# Make sure the directory is exist

file{'/var/www/s3598797':

ensure => directory,

}

augeas { 'httpd\_conf':

context => '/files/etc/httpd/conf/httpd.conf',

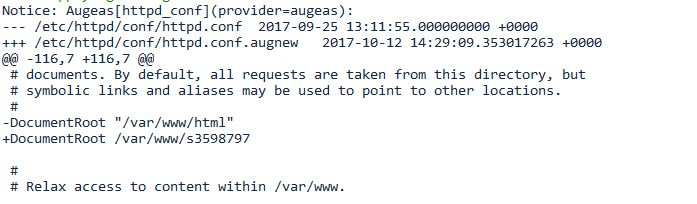
changes => [

'set directive[.="DocumentRoot"] "DocumentRoot"',

"set directive[.='DocumentRoot']/arg '/var/www/s3598797'",

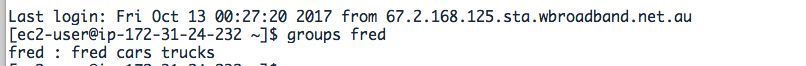
],

}



* 1. Add Ba
  2. Add Fred to sudo (wheel) group

Before the script running, I checked the Fred is not in the sudo (wheel) group



exec{ 'sudo useradd -g fred wheel':

path => ['/usr/bin', '/usr/sbin','/usr/local/bin'],

}

By using the command “sudo usermod -aG wheel fred”, it will put fred into group wheel

Result:

/var/folders/k9/9bq9g06s0rl26dls4k9zd7m80000gn/T/ro.nextwave.Snappy/ro.nextwave.Snappy/D8881C5C-6F7B-449A-8721-1AF06A909960.png

* 1. Ensure /usr/local/bin has been set for everyone in system.

Set the path for /bin , /sbin, /usr/bin, /usr/sbin, and /usr/local/bin, In case the system $PATH was fully replaced by the code.

Exec { path => [ '/bin/', '/sbin/' , '/usr/bin/', '/usr/sbin/','/usr/local/bin' ] }

* 1. mount

1. Task 5: Manage the hosts record to make a shortcut for ssh login without input full URL like titan.csit.rmit.edu.au

Before I do any change, I checked the /etc/hosts file. It looks like:



# create a fully qualified full host entry with an alias

host { 'titan.csit.rmit.edu.au':

ip => '131.170.5.131',

host\_aliases => 'titan',

}

host { 'jupiter.csit.rmit.edu.au':

ip => '131.170.5.135',

host\_aliases => 'jupiter',

}

host { 'saturn.csit.rmit.edu.au':

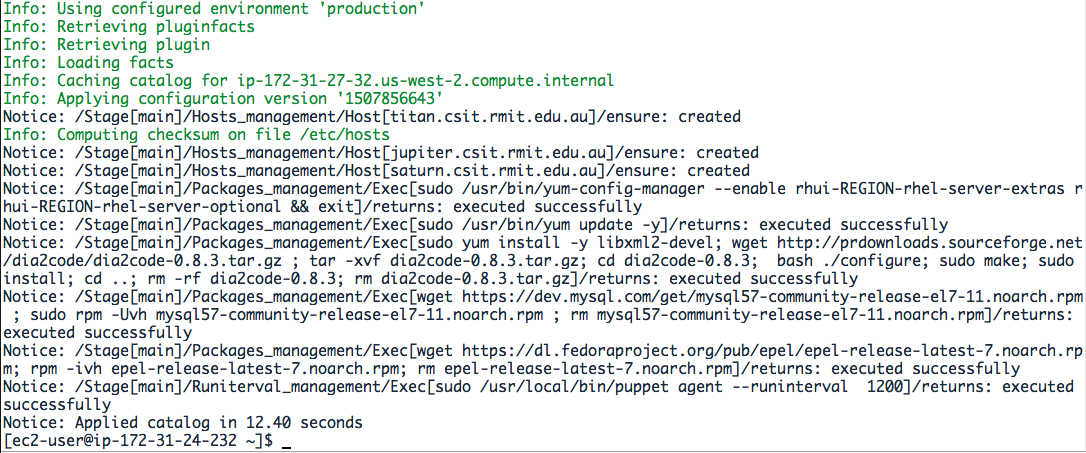
ip => '131.170.5.132',

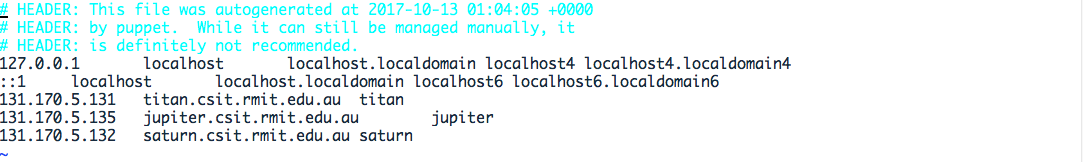
host\_aliases => 'saturn',

}

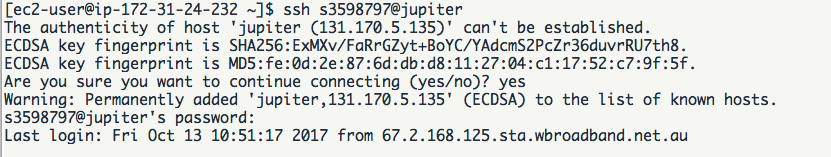
These operate will create the host and host aliases for each host, after that I will able to access these URL without input full address.

**Results:**

****



Succussed login in to core teaching via SSH sxxxx@jupiter



1. Task 6
2. Task 7: Ensure the PTAH environment include /usr/local/bin

Simply put the following code in the class, with will insert the PATH into the environment.

Exec { path => '/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/puppetlabs/bin' }

**Results:**



1. Task 8

# add a notify to the file resource

file { '/etc/ssh/sshd\_config':

notify => Service['sshd'], # this sets up the relationship

mode => '0644',

owner => 'root',

group => 'root',

}

file { '/etc/httpd/conf/httpd.conf':

notify => Service['httpd'], # this sets up the relationship

mode => '0644',

owner => 'root',

group => 'root',

}

