

Project description:

A development project is about to start programming. They ask you to deploy a development environment for them. They have the following requirements:

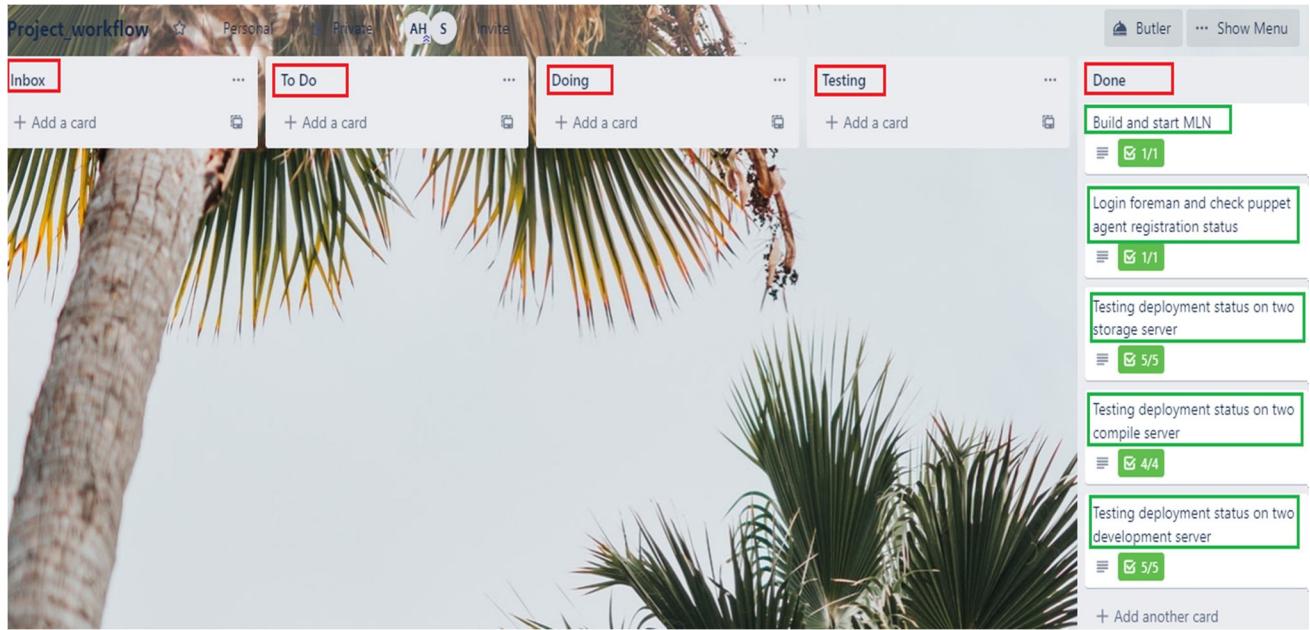
- . - Two storage servers with the GlusterFS server installed
- . - Two servers the developers will use to write code on. They have to have emacs, jed and subversion installed as well as git.
- . - All machines have to have four users with preferred usernames:
 - bob - janet - alice - tim
 - . - All four (*corrected*) users should get root access via sudo
 - . - Tim and janet have to be members of the group «developers». That group has to be created also.
 - . - Two compile servers with gcc, make and binutils installed
 - . - You don't have to configure the GlusterFS service, the project will do that themselves. Just install the packages.

2. Workflow Design

Here I use trello board to map the flow of the work. In the trello board it have 05 cards mentioned below

Trello Card Name	Function
Inbox	The tasks are created here
To Do	The tasks which need to be done
Doing	Task WIP status
Testing	Testing of the work and complete the checklist
Done	Completion of the work.

The tasks moves from left to right till completion . Each tasks has their own checklist for testing. After each task is completed, it is being moved to next card and so on. Screenshot of used trello board is given below where Red marked showing the cards and green marked showing the tasks.



Screenshot of trello board for workflow

Key tasks inside the inbox card are listed following :

2.1 Download necessary script sh file and execute them :

this task will download script sh file and once execute permission provide to the sh file then after execution the sh files will download MLN file and puppet manifests files at proper directory inside the master VM names io12_project , who is already the puppet master of this project. For more details check workflow implementation [download](#)

2.2 Build and start MLN:

this task will build the MLN project file and then start 06 VMs namely storage-server1, storage-server2, compile-server1, compile-server2 & dev-server1, dev-server2 in the openstack who will act as puppet agents of this project. For more info check [build and start](#)

2.3 Login foreman and check puppet agent status:

Login inside foreman with necessary credentials and will check puppet agents hosts and certificates status as well as puppet agents last report status. If the last report status is latest time it indicates the configuration synchronization between puppet master and puppet agents already done and puppets agents are ready for service. For more information check [foreman](#)

Below tasks mainly for testing of configuration at puppet agents-

2.4 Testing deployment status on two storage servers

in this task, it will check user creation status , their sudo privilege status and glusterfs server installation status on two storage servers. More detail check [test storage servers](#)

2.5. Testing deployment status on two compile servers: in this task it will check user creation status , their sudo privilege status and gcc, make and binutils package installation status at two compile servers. For more info check [test compiler server](#)

2.6. Testing deployment status on two development servers: in this task it will check user creation status , their sudo privilege status and emacs, jed and subversion package installation status at two development servers. For more information check [test developer server](#)

3. Workflow implementation:

This is the section where implementation of the workflow is discussed. Tasks given in the trello card above are discussed and explained. There are different tasks namely 6, has different steps to follow within those tasks.

3 .1 Download necessary script sh file and execute them :

There are different steps that should be followed within this task given in the inbox card of the trello board. These steps will login to master VM and download necessary files for creating environment at puppet master for the upcoming puppet agents.

`ssh ubuntu@128.39.121.241 :`

This step helps to login into the master VM where the necessary environment is setup for the project. With ssh, command the floating IP of the master VM's instance is attached inorder to login into the VM and download download.sh and whitelist.sh as root user in root user home directory.

`sudo su`

`cd /root/`

`wget https://git.cs.hioa.no/s340023/puppet_test/raw/master/download.sh :`

This command is used to download the necessary files that are included in the download.sh file from group Github account into the root directory of the master VM. Inside the download.sh file there are other commands which are necessary to download other necessary MLN & puppet manifest files required to setup the environment for the project. The commands inside the download.sh file are given below:

```

root@master:~# cat download.sh
#!/bin/bash
curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/project.mln > /root/project.mln
sleep 2
curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/compile_packages.pp > /etc/puppet/environments/production/manifests/compile_packages.pp
sleep 2
curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/dev_packages.pp > /etc/puppet/environments/production/manifests/dev_packages.pp
sleep 2
curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/gluster.pp > /etc/puppet/environments/production/manifests/gluster.pp
sleep 2
curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/privilege.pp > /etc/puppet/environments/production/manifests/privilege.pp
sleep 2
curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/user.pp > /etc/puppet/environments/production/manifests/user.pp
sleep 2
curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/site.pp > /etc/puppet/environments/production/manifests/site.pp
root@master:~#

```

Screenshot of download.sh file configuration

Description of the commands who are inside of download.sh file is given below

→ curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/project.mln > /root/project.mln:

This curl command is used to download the project.mln file from the github and then put that file in the root directory of the master VM. project.mln file contains all the necessary script that is required to create the numbers of VM and manage the large network.

→ curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/compile_packages.pp > /etc/puppet/environments/production/manifests/compile_packages.pp:

This curl command is used to download the compile_packages.pp file from the github and then put that file in the /etc/puppet/environments/production/manifests directory of the master VM. This pp file consists of puppet command purpose to install gcc , make, binutils packages in compile servers.

Screenshot of the code of **compile_packages.pp** is giving here:

```

root@master:/etc/puppet/environments/production/manifests# cat compile_packages.pp
class compile_packages {
package { "gcc" :
ensure => installed,
}
package { "make" :
ensure => installed,
}
package { "binutils" :
ensure => installed,
}
}
root@master:/etc/puppet/environments/production/manifests#

```

Screenshot of compile_packages.pp file in terminal , red marked is package and green mark is path

```
→ curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/dev_packages.pp >
/etc/puppet/environments/production/manifests/dev_packages.pp
```

This curl command is used to download the `dev_packages.pp` file from the github and then put that file in the `/etc/puppet/environments/production/manifests` directory of the master VM. This pp file consists of puppet command purpose to install emacs, jed, subversion,git packages in dev_servers.

Screenshot of the code of `dev_packages.pp` is giving here:

```
root@master:/etc/puppet/environments/production/manifests# cat dev_packages.pp
class dev_packages {
  package { "emacs" :
    ensure => installed,
  }
  package { "jed" :
    ensure => installed,
  }
  package { "subversion" :
    ensure => installed,
  }
  package { "git" :
    ensure => installed,
  }
}
root@master:/etc/puppet/environments/production/manifests#
```

[Screenshot of dev_packages.pp file in terminal](#)

```
→ curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/gluster.pp >
/etc/puppet/environments/production/manifests/gluster.pp
```

This curl command is used to download the `gluster.pp` file from the github and then put that file in the `/etc/puppet/environments/production/manifests` directory of the master VM. this pp file consists of puppet command purpose to install glusterfs server packages in storage_servers.

Screenshot of the code of `gluster.pp` is giving here:

```
root@master:/etc/puppet/environments/production/manifests# cat gluster.pp
class gluster{
  package { 'glusterfs-server' :
    ensure => installed,
  }
}
root@master:/etc/puppet/environments/production/manifests#
```

[Screenshot of gluster.pp file in terminal](#)

```
→ curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/privilege.pp >
/etc/puppet/environments/production/manifests/privilege.pp
```

This curl command is used to download the `privilege.pp` file from the github and then put that file in the `/etc/puppet/environments/production/manifests` directory of the master VM. This pp file consists of puppet exec command purpose to give sudo privilege for 04 users in all servers unless the users already do not have sudo privilege.

Screenshot of the code of `privilege.pp` is giving here

```

root@master:/etc/puppet/environments/production/manifests# cat privilege.pp
class privilege {
exec { 'sudotim':
  command    => '/usr/sbin/usermod -aG sudo tim',
  unless => '/usr/bin/sudo -l -U tim | grep ALL',
}
exec { 'sudojanet':
  command    => '/usr/sbin/usermod -aG sudo janet',
  unless => '/usr/bin/sudo -l -U janet | grep ALL',
}
exec { 'sudoalice':
  command    => '/usr/sbin/usermod -aG sudo alice',
  unless => '/usr/bin/sudo -l -U alice | grep ALL',
}
exec { 'sudobob':
  command    => '/usr/sbin/usermod -aG sudo bob',
  unless => '/usr/bin/sudo -l -U bob | grep ALL',
}
}
root@master:/etc/puppet/environments/production/manifests#

```

Screenshot of privilege.pp file

```

→curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/user.pp >
/etc/puppet/environments/production/manifests/user.pp

```

Curl command is used to download the **user.pp** file from the github and then put that file in the **/etc/puppet/environments/production/manifests** directory of the master VM. This pp file consists of puppet command purpose to create users namely alice,bob,janet & tim , as well as create a group called developers and put tim & janet to that group in all servers. Screenshot of the code of **user.pp is giving here**

```

class user {
group { 'developers':
  ensure => 'present',
}

user { 'bob':
  ensure => present,
  home => '/home/bob',
  managehome => true,
  password => pw_hash('bob', 'SHA-512', 'mysalt'),
}

user { 'janet':
  ensure => present,
  home => '/home/janet',
  managehome => true,
  groups => 'developers',
  password => pw_hash('janet', 'SHA-512', 'mysalt'),
}

user { 'alice':

```

```

user { 'alice':
  ensure => present,
  home => '/home/alice',
  managehome => true,
  password => pw_hash('alice', 'SHA-512', 'mysalt'),
}
user { 'tim':
  ensure => present,
  home => '/home/tim',
  managehome => true,
  groups => 'developers',
  password => pw_hash('tim', 'SHA-512', 'mysalt'),
}
}

```

Screenshot of user.pp file

```

→ curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/site.pp >
/etc/puppet/environments/production/manifests/site.pp

```

Curl command is used to download the `site.pp` file from the github and then put that file in the `/etc/puppet/environments/production/manifests` directory of the master VM. **this site.pp file's purpose is to start compiling a catalog with above mentioned manifest files and apply them to corresponding puppet agents. This starting point is called the *site manifest*.**

```

# site.pp must exist (puppet #15106, foreman #1708)
node "compile-server1" {
  include user
  include privilege
  include compile_packages
}

node "compile-server2" {
  include user
  include privilege
  include compile_packages
}

node "dev-server1" {
  include user
  include privilege
  include dev_packages
}

node "dev-server2" {
  include user
  include privilege
  include dev_packages
}

```

Screenshot of site.pp file

```

wget https://git.cs.hioa.no/s340023/puppet_test/raw/master/whitelist.sh:

```

This command is used to download the `whitelist.sh` file where it's purpose is to configure upcoming puppet agents server hostname in to master VM puppet master in `/etc/puppet/autosign.conf` file thus the puppet agents can register with puppet master without manual intervention.

```
#!/bin/bash
echo "compile-server1" >> /etc/puppet/autosign.conf
sleep 1
echo "compile-server2" >> /etc/puppet/autosign.conf
sleep 1
echo "dev-server1" >> /etc/puppet/autosign.conf
sleep 1
echo "dev-server2" >> /etc/puppet/autosign.conf
sleep 1
echo "storage-server1" >> /etc/puppet/autosign.conf
sleep 1
echo "storage-server2" >> /etc/puppet/autosign.conf
```

Screenshot of `whitelist.pp` file

Below command is to give necessary execute permission to sh files and execute them thus they can download the dependent files for the work , files function were mentioned above descriptions.

```
chmod +x download.sh
```

```
chmod +x whitelist.sh
```

```
./download.sh
```

```
./whitelist.sh
```

3.2 Build and start MLN

In the above section it is already discussed what necessary files need to download and the function of each file. In this section it will briefly describe about the MLN script file in this project called `project.mln` file which is used to build , start VMs and apply some additional features to those VMs when they start up.

As general practice `mln build -f project.mln` is used to build MLN

Then `mln start -p deployment -s 20` fired up the VMs namely **storage-server1, storage-server2, compile-server1, compile-server2 & dev-server1, dev-server2** for the purpose of `project1` by taking 20seconds interval between fire up each VM. Each of this VM will act as a puppet agent and our master VM is puppet master.

Key configuration of `project.mln` is mentioned below:

Base: Ubuntu16.04 OS

flavor m1.small

Network netsys_net

And additional features as follows:

- 1. Namespace is given according to function of the hosts**

2. Install puppet agents
3. Download then apply master VM root ssh key by puppet apply master_key.pp file
4. Download and then configure /etc/hosts file with master VM IP & hostname by puppet apply hosts_entry.pp file.
5. Configure /etc/puppet/default file and /etc/puppet/puppet.conf file accordingly
6. And finally restart puppet service

Screenshot of project MLN fie key config parts is giving below:

```
host storage_server1 {
    superclass common
}

host storage_server2 {
    superclass common
}

host dev_server1 {
    superclass common
}

host dev_server2 {
    superclass common
}

host compile_server1 {
    superclass common
}

host compile_server2 {
    superclass common
}root@master:~#
```

```
superclass common {
  openstack {
    image Ubuntu16.04
    flavor m1.small
  }
}

user_data {
  curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/ssh_master_key.pp > /home/ubuntu/master_key.pp
  sleep 2
  ls -lrt /home/ubuntu/master_key.pp
  sleep 1
  sudo wget -O /home/ubuntu/puppetlabs-release-trusty.deb https://apt.puppetlabs.com/puppetlabs-release-trusty.deb
  sleep 2
  dpkg -i /home/ubuntu/puppetlabs-release-trusty.deb
```

```

sleep 2
apt-get update && apt-get -y install puppet
sleep 10
puppet --version
sleep 2
puppet resource user root
sleep 1
puppet apply /home/ubuntu/master_key.pp
sleep 5
cp -v /home/ubuntu/.ssh/authorized_keys /root/.ssh/
sleep 2
touch /etc/default/puppet
sleep 2
echo "START = yes" > /etc/default/puppet
sleep 2
curl https://git.cs.hioa.no/s340023/puppet_test/raw/master/hosts_entry.pp > /home/ubuntu/hosts_ent
'y.pp
sleep 3
ls -lrt /home/ubuntu/hosts_entry.pp
sleep 1
sudo puppet apply /home/ubuntu/hosts_entry.pp
sleep 5
sudo puppet config set --section agent server master.openstacklocal
sleep 2
sudo puppet config set --section agent interval 180
sleep 2
sudo puppet config set --section agent runinterval 5m
sleep 2
sudo service puppet restart
sleep 2
sudo puppet agent --enable

```

Screenshot of project MLN file key config parts from master VM

In above screenshot of MLN script file it is shown that two additional puppet manifest files are being downloaded and config at puppet agents at user_data section for each VM. Brief about those two manifests file is given following.

hosts_entry.pp file is used to configure master VM IP & fqdn at puppet agents' /etc/hosts file

 **hosts_entry.pp** 139 Bytes 

```

1 host {
2     'master.openstacklocal':
3         ip => '10.0.31.180',
4         host_aliases => 'master',
5         ensure => 'present',
6 }

```

Screenshot of hosts_entry.pp file configuration

ssh_master_key.pp file purpose is to apply master VM root ssh user key to puppet agents thus from master VM we can login puppet agents as root user

```

1 ssh_authorized_key { "master_root" :
2   key => 'AAAAB3NzaC1yc2EAAAQABAAQD0C/q2
3   user => ubuntu,
4   ensure => present,
5   type => rsa,
6 }

```

Screenshot of ssh_master_key.pp file configuration

3.3 Login foreman and check puppet agent registration status and configuration sync status

Login Foreman <https://128.39.121.241>

Infrastructure -> Smart Proxies -> Click Edit drop down menu -> Certificates

Here check all the puppet agents certificates showing or not

Then wait around 10minutes and check puppet agents status ok or not at [Hosts -> All Hosts](#) on the foreman

As long as the node last report status latest time is showing there it means puppet agent already sync configuration with puppet master

Certificate Name	State
compile-server1	valid
compile-server2	valid
dev-server1	valid
dev-server2	valid
master.openstacklocal	valid
project-puppet-vm2	valid
storage-server1	valid
storage-server2	valid

Screenshot showing puppet agent registration status at Foreman

Name	Last report
compile-server1	3 minutes ago
compile-server2	1 minute ago
dev-server1	2 minutes ago
dev-server2	2 minutes ago
master.openstacklocal	21 minutes ago
project-puppet-vm2.openstack	1 minute ago
storage-server1	4 minutes ago
storage-server2	2 minutes ago

Screenshot showing puppet agents last report time for config sync at Foreman

4. Testing of the workflow

4.1 Testing of MLN project deployment status

mln status -p deployment

All VM status is showing up and running

```
root@master:~# mln status -p deployment
deployment host dev_server2 up - | Running | netsys_net=10.0.35.45
deployment host storage_server1 up - | Running | netsys_net=10.0.35.43
deployment host dev_server1 up - | Running | netsys_net=10.0.35.42
deployment host storage_server2 up - | Running | netsys_net=10.0.35.44
deployment host compile_server2 up - | Running | netsys_net=10.0.35.41
deployment host compile_server1 up - | Running | netsys_net=10.0.35.46
root@master: #
```

Screenshot : project all VM status is showing up and running

4.2 Testing deployment status on two compile server :

Login in compile-server1 & compile-server2 puppet agents from master VM as root user , IP you can take from foreman . below screenshot is an example for compile-server1
All Screenshot is this section taken from compile-server1 as example

```
root@master:~# ssh root@10.0.35.46
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-83-generic x86_64)

System restart required
Last login: Sat Feb 29 08:41:35 2020 from 10.0.31.180
root@compile-server1:~#
```

Screenshot of Login compile-server1(10.0.35.46) from master VM(10.16.31.180) as root user

Check Compiled catalog status from puppet agents is finished or not.

```
root@compile-server1:~# cat /var/log/syslog | grep catalog
Feb 28 12:02:59 compile-server1 cloud-init[1232]: #033[mNotice: Compiled catalog for compile-server1 in environment production in 0.10 seconds#033[0m
Feb 28 12:03:00 compile-server1 cloud-init[1232]: #033[mNotice: Finished catalog run in 0.06 seconds#033[0m
Feb 28 12:03:18 compile-server1 cloud-init[1232]: #033[mNotice: Compiled catalog for compile-server1 in environment production in 0.11 seconds#033[0m
Feb 28 12:03:19 compile-server1 cloud-init[1232]: #033[mNotice: Finished catalog run in 0.07 seconds#033[0m
Feb 28 12:04:17 compile-server1 puppet-agent[4260]: Finished catalog run in 26.49 seconds
Feb 28 12:08:45 compile-server1 puppet-agent[9054]: Finished catalog run in 0.27 seconds
Feb 28 12:13:44 compile-server1 puppet-agent[9248]: Finished catalog run in 0.18 seconds
Feb 28 12:18:44 compile-server1 puppet-agent[9435]: Finished catalog run in 0.18 seconds
Feb 28 12:23:44 compile-server1 puppet-agent[9611]: Finished catalog run in 0.23 seconds
```

Screenshot of Compiled catalog finish status from compile-server1

Check user bob, janet ,alice, tim creation status. If following command has output that means user already created

```
grep bob /etc/passwd
grep janet /etc/passwd
grep alice /etc/passwd
grep tim /etc/passwd
```

```
root@compile-server1:~# grep bob /etc/passwd
bob:x:1001:1002::/home/bob:
root@compile-server1:~# grep janet /etc/passwd
janet:x:1002:1003::/home/janet:
root@compile-server1:~# grep alice /etc/passwd
alice:x:1004:1005::/home/alicex:
root@compile-server1:~# grep tim /etc/passwd
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
tim:x:1003:1004::/home/tim:
root@compile-server1:~#
```

Screenshot of user creation status from compile-server1

Check user bob, janet ,alice,tim have sudo privilege or not. if the command has output then it means users have sudo privilege

```
/usr/bin/sudo -l -U tim
/usr/bin/sudo -l -U janet
/usr/bin/sudo -l -U alice
/usr/bin/sudo -l -U bob
```

```
root@compile-server1:~# /usr/bin/sudo -l -U tim
sudo: unable to resolve host compile-server1
Matching Defaults entries for tim on compile-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/us
User tim may run the following commands on compile-server1:
    (ALL : ALL) ALL
root@compile-server1:~# /usr/bin/sudo -l -U janet
sudo: unable to resolve host compile-server1
Matching Defaults entries for janet on compile-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/us
User janet may run the following commands on compile-server1:
    (ALL : ALL) ALL
root@compile-server1:~# /usr/bin/sudo -l -U alice
sudo: unable to resolve host compile-server1
Matching Defaults entries for alice on compile-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/us
User alice may run the following commands on compile-server1:
    (ALL : ALL) ALL
root@compile-server1:~# /usr/bin/sudo -l -U bob
sudo: unable to resolve host compile-server1
Matching Defaults entries for bob on compile-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/us
User bob may run the following commands on compile-server1:
    (ALL : ALL) ALL
root@compile-server1:~#
```

Screenshot of users' sudo privilege status from compile-server1

Check tim & janet are belong to developers group or not. if the command has output that means tim & janet belongs to developers group, it also shows bob & alice are not in the developer group.

```
grep developers /etc/group | awk -F: '{print $4}'
root@compile-server1:~# grep developers /etc/group | awk -F: '{print $4}'
janet,tim
root@compile-server1:~#
```

Screenshot of janet & tim belongs to developer group

Check make ,gcc, binutils version with below command respectively . if version information shows that means make, gcc and binutils already installed.

```
make --version
gcc --version
ld -v
```

```
root@compile-server1:~# make --version
GNU Make 4.1
Built for x86_64-pc-linux-gnu
Copyright (C) 1988-2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
root@compile-server1:~# gcc --version
gcc (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

root@compile-server1:~# ld -v
GNU ld (GNU Binutils for Ubuntu) 2.26.1
root@compile-server1:~#
```

Screenshot of make , gcc & binutils package installation status from compile-server1

4.2 Testing deployment status on two storage server

Login in storage-server1 & storage-server2 as root user from master , IP you can take from foreman.

All Screenshot is this section taken from storage-server1 as example

```
root@master:~# ssh root@10.0.35.43
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-83-generic x86_64)

Last login: Sat Feb 29 09:45:02 2020 from 10.0.31.180
root@storage-server1:~#
```

Screenshot of Login storage-server1(10.0.35.43) from master VM(10.16.31.180) as root user

Check Compiled catalog status from puppet agents is finished or not.

```
root@storage-server1:~# cat /var/log/syslog | grep catalog
Feb 28 12:00:16 storage-server1 cloud-init[1234]: #033[mNotice: Compiled catalog for storage-server1 in environment production in 0.10 seconds#033[0m
Feb 28 12:00:17 storage-server1 cloud-init[1234]: #033[mNotice: Finished catalog run in 0.07 seconds#033[0m
Feb 28 12:00:35 storage-server1 cloud-init[1234]: #033[mNotice: Compiled catalog for storage-server1 in environment production in 0.11 seconds#033[0m
Feb 28 12:00:35 storage-server1 cloud-init[1234]: #033[mNotice: Finished catalog run in 0.07 seconds#033[0m
Feb 28 12:06:49 storage-server1 puppet-agent[4278]: Finished catalog run in 45.36 seconds
Feb 28 12:11:01 storage-server1 puppet-agent[10160]: Finished catalog run in 0.20 seconds
Feb 28 12:16:07 storage-server1 puppet-agent[10344]: Finished catalog run in 7.86 seconds
Feb 28 12:21:09 storage-server1 puppet-agent[10627]: Finished catalog run in 9.17 seconds
Feb 28 12:26:05 storage-server1 puppet-agent[10808]: Finished catalog run in 0.19 seconds
```

Screenshot of Compiled catalog finish status from storage-server1

Check user bob, janet ,alice,tim creation status. If following command has output that means user already created

```
grep bob /etc/passwd
grep janet /etc/passwd
grep alice /etc/passwd
grep tim /etc/passwd
```

```
root@storage-server1:~# grep bob /etc/passwd
bob:x:1001:1002::/home/bob:
root@storage-server1:~# grep janet /etc/passwd
janet:x:1002:1003::/home/janet:
root@storage-server1:~# grep alice /etc/passwd
alice:x:1004:1005::/home/alice:
root@storage-server1:~# grep tim /etc/passwd
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
tim:x:1003:1004::/home/tim:
```

Screenshot of user creation status from storage-server1

Check user bob, janet ,alice,tim have sudo privilege or not. if the command has output then it means users have sudo privilege

```
/usr/bin/sudo -l -U tim
/usr/bin/sudo -l -U janet
/usr/bin/sudo -l -U alice
/usr/bin/sudo -l -U bob
```

```
root@storage-server1:~# /usr/bin/sudo -l -U tim
sudo: unable to resolve host storage-server1
Matching Defaults entries for tim on storage-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loc
```

User tim may run the following commands on storage-server1:

```
(ALL : ALL) ALL
```

```
root@storage-server1:~# /usr/bin/sudo -l -U janet
sudo: unable to resolve host storage-server1
Matching Defaults entries for janet on storage-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loc
```

User janet may run the following commands on storage-server1:

```
(ALL : ALL) ALL
```

```
root@storage-server1:~# /usr/bin/sudo -l -U alice
sudo: unable to resolve host storage-server1
Matching Defaults entries for alice on storage-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loc
```

User alice may run the following commands on storage-server1:

```
(ALL : ALL) ALL
```

```
root@storage-server1:~# /usr/bin/sudo -l -U bob
sudo: unable to resolve host storage-server1
Matching Defaults entries for bob on storage-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loc
```

User bob may run the following commands on storage-server1:

```
(ALL : ALL) ALL
```

```
root@storage-server1:~#
```

Screenshot of users' sudo privilege status from storage-server1

Check tim & janet are belong to developers group and bob & alice not in this group. if the command has output that means tim & janet belongs to developers group

```
grep developers /etc/group | awk -F: '{print $4}'
```

```
root@storage-server1:~# grep developers /etc/group | awk -F: '{print $4}'  
janet,tim  
root@storage-server1:~#
```

Screenshot of janet & tim belongs to developer group from storage-server1

Check glusterfs server version . if version information showing that means glusterfs already installed.

```
glusterfs --version
```

```
root@storage-server1:~# gluster --version  
glusterfs 3.7.6 built on Dec 25 2015 20:50:46  
Repository revision: git://git.gluster.com/glusterfs.git  
Copyright (c) 2006-2011 Gluster Inc. <http://www.gluster.com>  
GlusterFS comes with ABSOLUTELY NO WARRANTY.  
You may redistribute copies of GlusterFS under the terms of the GNU General Public License.  
root@storage-server1:~#
```

Screenshot of glusterfs server package installation status from storage-server1

4.3 Testing deployment status on two development server

Login in dev-server1 & dev-server2 as root user from master , IP you can take from foreman. below screenshot is an example for dev_server1

All Screenshot is this section taken from dev-server1 as example

```
root@master:~# ssh root@10.0.35.42  
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-83-generic x86_64)  
  
Last login: Fri Feb 28 12:13:38 2020 from 10.0.31.180  
root@dev-server1:~#
```

Screenshot of Login dev-server1(10.0.35.42) from master VM(10.16.31.180) as root user

Check Compiled catalog status from puppet agents is finished or not.

```
root@dev-server1:~# cat /var/log/syslog | grep catalog  
Feb 28 11:59:39 dev-server1 cloud-init[1224]: #033[mNotice: Compiled catalog for dev-server1 in environment production in 0.10 seconds#033[0m  
Feb 28 11:59:40 dev-server1 cloud-init[1224]: #033[mNotice: Finished catalog run in 0.07 seconds#033[0m  
Feb 28 11:59:58 dev-server1 cloud-init[1224]: #033[mNotice: Compiled catalog for dev-server1 in environment production in 0.11 seconds#033[0m  
Feb 28 11:59:59 dev-server1 cloud-init[1224]: #033[mNotice: Finished catalog run in 0.06 seconds#033[0m  
Feb 28 12:07:23 dev-server1 puppet-agent[4275]: Finished catalog run in 101.30 seconds  
Feb 28 12:10:24 dev-server1 puppet-agent[9680]: Finished catalog run in 0.18 seconds  
Feb 28 12:15:24 dev-server1 puppet-agent[9958]: Finished catalog run in 0.18 seconds  
Feb 28 12:20:25 dev-server1 puppet-agent[10149]: Finished catalog run in 0.21 seconds  
Feb 28 12:25:25 dev-server1 puppet-agent[10330]: Finished catalog run in 0.18 seconds  
Feb 28 12:30:27 dev-server1 puppet-agent[10509]: Finished catalog run in 0.19 seconds
```

Screenshot of Compiled catalog finish status from storage_server1

Check user bob, janet ,alice,tim creation status. If following command has output that means user already created

```
grep bob /etc/passwd
```

```
grep janet /etc/passwd
grep alice /etc/passwd
grep tim /etc/passwd
root@dev-server1:~# grep bob /etc/passwd
bob:x:1001:1002::/home/bob:
root@dev-server1:~# grep janet /etc/passwd
janet:x:1002:1003::/home/janet:
root@dev-server1:~# grep alice /etc/passwd
alice:x:1004:1005::/home/alice:
root@dev-server1:~# grep tim /etc/passwd
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
tim:x:1003:1004::/home/tim:
root@dev-server1:~#
```

Screenshot of user creation status from dev-server1

Check user bob, janet ,alice,tim have sudo privilege or not. if the command has output then it means users have sudo privilege

```
/usr/bin/sudo -l -U tim
/usr/bin/sudo -l -U janet
/usr/bin/sudo -l -U alice
/usr/bin/sudo -l -U bob
```

```
root@dev-server1:~# /usr/bin/sudo -l -U tim
sudo: unable to resolve host dev-server1
Matching Defaults entries for tim on dev-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:,/
User tim may run the following commands on dev-server1:
    (ALL : ALL) ALL
root@dev-server1:~# /usr/bin/sudo -l -U janet
sudo: unable to resolve host dev-server1
Matching Defaults entries for janet on dev-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:,/
User janet may run the following commands on dev-server1:
    (ALL : ALL) ALL
root@dev-server1:~# /usr/bin/sudo -l -U alice
sudo: unable to resolve host dev-server1
Matching Defaults entries for alice on dev-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:,/
User alice may run the following commands on dev-server1:
    (ALL : ALL) ALL
root@dev-server1:~# /usr/bin/sudo -l -U bob
sudo: unable to resolve host dev-server1
Matching Defaults entries for bob on dev-server1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:,/
User bob may run the following commands on dev-server1:
    (ALL : ALL) ALL
```

Screenshot of users' sudo privilege status from dev-server1

Check tim & janet are belong to developers group and bob & alice not in this group. if the command has output that means tim & janet belongs to developers group
grep developers /etc/group | awk -F: '{print \$4}'

```
root@dev-server1:~# grep developers /etc/group | awk -F: '{print $4}'
janet,tim
root@dev-server1:~#
```

Screenshot of janet & tim belongs to developer group from dev-server1

Check emacs,jed ,subversion ,git version with below command respectively . if version information showing that means emacs, jed, subversion and git already installed.

```
emacs --version
jed --version
svn --version
git --version
```

```

root@dev-server1:~# emacs --version
GNU Emacs 24.5.1
Copyright (C) 2015 Free Software Foundation
GNU Emacs comes with ABSOLUTELY NO WARRANTY
http://www.gnu.org/licenses/
root@dev-server1:~# jed --version
jed version: 0.99.19/unix
Compiled with GNU C 4.9
S-Lang version: 2.3.0
root@dev-server1:~# svn --version
svn, version 1.9.3 (r1718519)
compiled Jul 26 2019, 15:51:17 on x86_64-pc-linux-gnu
Copyright (C) 2015 The Apache Software Foundation
This software consists of contributions made by many
people; see the NOTICE file for more information.
Subversion is open source software, see http://subversion.apache.org

```

```

root@dev-server1:~# git --version
git version 2.7.4
root@dev-server1:~

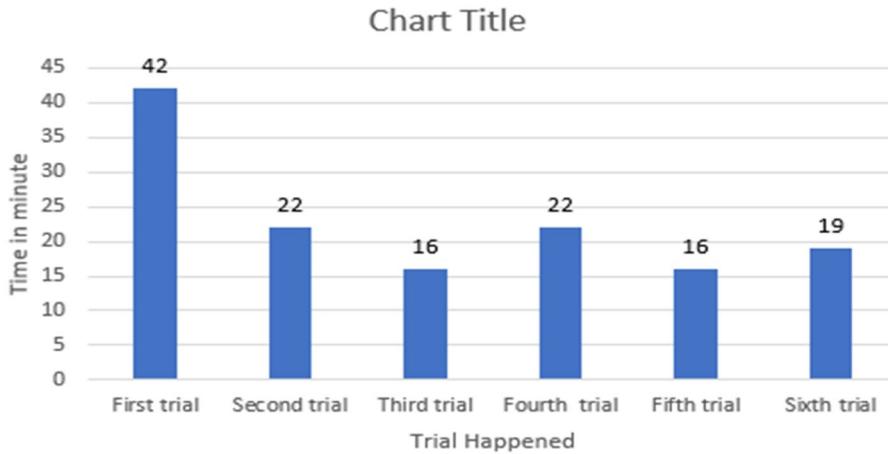
```

Screenshot of emacs, jed , svn & git package installation status from dev-server1

5. Implementation and Testing Reflection

In the group io12, we two members did implementation and testing 06 times where each guy did it 03 times. The first trial took 42 minutes and then the next implementation time reduced gradually and also had some ups and downs. The minimum implementation time duration was 16 minutes. The statistics of the time duration of the implementation and the testing is shown in table and graph (picture 1)

Trial	Time duration in minute
First trial	42
Second trial	22
Third trial	16
Fourth trial	22
Fifth trial	16
Sixth trial	19



Picture 1: Number of trial and time duration of each trial

6. Junior workflows

Thes project of the “A development environment” was tested with two other users other then the group io12 members for the testing and calculating the time duration by the user to deploy the environment designed by the group io12. Each of the users tested the project in different times and the time duration of the deployment of the environment also varies with a small amount of time. The testing was done in the same manner as the project designer so that the time duration can be measured. The first junior user took 27 minutes to deploy the development environment where the second user took 30 minute of time to do the same task. The two members of the group io12 took an average of 26 minutes and 19 minutes respectively from the first three trials and the last three trials, where the other two users took 27 and 30 minutes of time to implement and test the designed project.

7. Rollback

Below are the rollback part to remove the environment from master and shutoff the VM agents and revoke their certificates and delete the hosts from foreman. Thus allow us to create a new development environment. To do this we have the following key steps:

1. Stop MLN: `mln stop -> mln stop -p deployment`
2. Delete project VM agent instances from openstack
3. Delete MLN project directory : `opt/mln/projects/root/deployment` directory \rightarrow `rm -rf /opt/mln/projects/root/deployment`
4. Revoke certificates from the foreman : `Infrastructure -> Smart Proxies -> Click Edit drop down menu -> Certificates -> Revoke`
5. Delete the puppet agents from foreman : `Hosts -> All Hosts -> Delete hosts`

6. Delete all downloaded files from master VM :
`-->rm /root/project.mln download.sh whitelist.sh`

```
-->rm /etc/puppet/environments/production/manifests/*
```

7. clear autosign.conf file at master VM `nano /etc/puppet/autosign.conf` and delete the content, save and exit.

8.Conclusion

It was a very good experience implementing this project. We became know how to manage several VMs from a central configuration management system. From this project work we try to gain knowledge regarding MLN, puppet master , puppet agents , Foreman . Besides, documenting the work and applying the work multiple times by different users give us idea about how to ensure idempotency.