

How to Install Puppet server/Puppet agent

Jing, mqjing@gmail.com

([閱讀 Google doc 版本](#))

ETA: 10 mins



(The Image is captured from [here](#))

現在不像以前, 安裝系統不再只是一台電腦的事情. 而是在一堆電腦上安裝軟體的問題. 許多系統的正常營運設計, 就是需要將軟體安裝在不同電腦上, 才能達成正常營運的功能. 例如: OpenStack 的 Control node, Compute Node. Ceph 的 Monitor node 就至少需要三台機器, 才符合設計的規範.

安裝軟體不只是談到一台機器內部本身, 有哪個軟體先裝, 哪一個軟體後裝的問題. 現代的某些叢集式系統還牽涉到整個系統 (多台電腦) 必須在某個狀態後, 才能安裝某項功能. 例如: Ceph 的 ObjectStorage 系統, 想安裝 ceph radosgw server, 你的 Ceph OSD 與 Ceph Monitor 都必須裝好. 最後才能安裝 radosgw server.

這些 challenge 都使得以前我們自己寫 script 來管理安裝這檔事, 變得複雜而難以管理. 部署工具的存在, 是必然的發展. 使用正確的部署工具設定安裝系統, 才是對的事情. 大格局使用部署工具, 小事情, 使用 script.

光是 wiki, 就列出 26 種不同的部署工具 [4]. 比較有名的有 puppet, chef, salt, Juju 以及 ansible. 如果你已經有 OpenStack, 而想把系統部署在上面, 你可以選擇 [OpenStack Heat](#) 或者是直接使

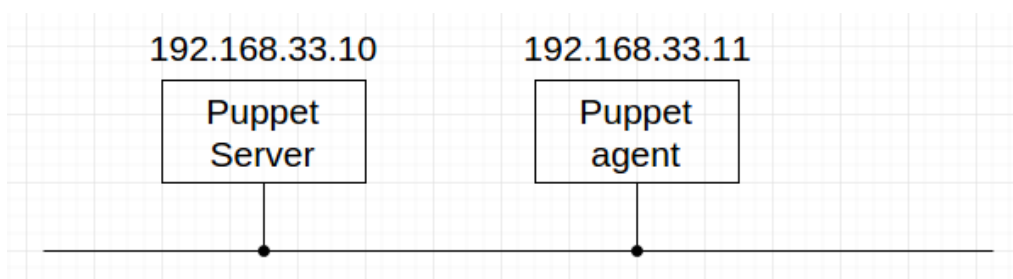
用 [Docker](#) image, 搭配 Docker Registry server, 把整個系統中的 service 一個一個包進 container 裡面, 然後利用 container 管理工具 [Docker Swarm](#) 或 [Kubernetes](#), 把 container deliver 到目標平台上面去。

今天講的是 Puppet. 有名的 OpenStack 的部署工具 fuel, 主要就是使用 Naigun 管理整個安裝流程, 其中 puppet 機制扮演了重要的實際安裝系統角色 (透過安裝在 node 上的 MCollective service). 你也可以用 puppet 來執行安裝自己設計的分散式系統, 只要有 puppet master server, 整個系統基本上就可以視為程式碼. 程式碼就是平台, 平台就是程式碼。

在繼續講下去之前, 我們的習慣是先裝整個系統. 然後再來拿著咖啡 麵包 可樂果 搭配著文件, 來測試說明手冊的內容以及架構. 玩具系統沒起來, 沒有親自實驗手冊講的東西, 心裡就是不會有踏實的感覺。

如何安裝一個最簡單的 Puppet 實驗環境, 在我的 Ubuntu 電腦裡? 10 分鐘自動設定好, 你喝咖啡看電影, 電腦工作。

Puppet Lab



([edit](#))

GitHub ([download](#))

Requirements

- VirtualBox: <https://www.virtualbox.org/>
- Vagrant: <https://www.vagrantup.com/>
- Git:
 - `sudo apt-get install git`

Start your puppet lab

1. `git clone https://github.com/jing-tw/lab-puppet-vagrant.git`
2. `cd ./server-agent-lab`
3. `./up.sh`

Why Puppet Server

Puppet master 是 ruby 的應用程式, 用來處理任意數量 Puppet Agent node 的 configurations. 但是 RMI Ruby runtime 的效能不好.

基於大量 deploy 效率的理由, 設計者設計出 Puppet Server 利用 JRuby 來執行原先 puppet master 的程式碼, 提供與 puppet master 完全相同的 service. 讓整個系統執行在 JVM 上. 在這其中, 有些原本在 puppet master 的 code, 則改用新的程式語言 Clojure 來實現 [1].

基本觀念

Puppet configure 整個目標系統, 主要就是兩個 stage:

- Compile a catalog
- Apply the catalog

而 catalog 文件就是你用來要求目標電腦系統的狀態. [Catalog 列了所有你想要管理的資源, 你想要的狀態以及它們之間被設定的先後順序](#) [2]. 使用 catalog 的好處可以讓 puppet manifest 用邏輯 模板 函式來描述每一台不同 node (機器) 的狀態與相依關係. 每一個 node 不需要瞭解其他 node 的資訊, 只要接收處理自己相關指令即可. 另外, 在 node 上的 agent 不需要執行 catalog 編譯的工作, 自然可以大大減少在 node 端在 deploy 時候, 所付出的額外工作, 把資源放在設定自己的系統工作上. Puppet 還有一種稱為 noop 模式, 允許執行模擬安裝的工作.

Agent/Master 架構

簡單的說, 就是 puppet master server 控制重要的部署設定資訊, 受管理的 agent node 則向 master node 提出 apply 自己相關 catalog 部署. 例如: 執行安裝 apache server 的工作.

在這個模式裡面, 每一個受管理的 node 都有一個在背景執行的 Puppet agent service. 有一個或多個 server 執行 Puppet master application. 而 Puppet master 是一種 Puppet Server 的形式.

整個的流程如下所示 [3]:

Step 1: Puppet agent 每隔一段時間, 就會遞送自己目前狀態 (fact) 給 Puppet master 並 request 關於自己的一份 catalog.

Step 2: Puppet master compile, 然後回傳對應的 catalog

Step 3: 一旦 Puppet agent 收到 catalog 後, 立即檢查每一個列在 catalog 中的 [resource](#), 它的要求狀態是否改變了 (例如: 察覺到某個設定檔的內容現在是 AAA, 或者是某個軟體現在被要求版本是 123 版之類的事情).

檢查結果如果發現有必要改變自己, 以符合來自 puppet master catalog 的描述. Puppet agent 就會列出所有不符合狀態的 resource, 執行修正安裝的工作, 最後使自己完全符合 catalog 的要求為止.

最後, Puppet agent 將送出一份報告給 Puppet master. 結束這次的流程.

Communication/Security

不是隨便在路上找一個 node, 就可以連到我們的 Puppet master, 提出安裝的要求. Puppet master 選擇經過它授權的 node, 接收 request.

因此, Puppet master 有自己的授權單位 (certification authority, CA), 用來處理這檔事. 我們可以在 master server 上, 使用 puppet cert list 指令, 視察目前有哪些新的 node 提出授權需求. 用 puppet cert sign 指令, 對已知的 node 授權. 經過授權的 node, 才可以提出需求, 進行 Puppet master catalog 設計的部署.

Installation Detail

Handle the Name (simple)

```
./hosts_ip_map.sh
```

```
declare -A HostIP
HostIP=( ["master.smb.com"]="192.168.33.10" \
        ["node1.smb.com"]="192.168.33.11")
```

```
./common.sh
```

```
function add_host() {
    HOSTNAME=$1
    IP=$2

    valid_ip ${IP}

    if [[ $? -eq 1 ]]
    then
        echo "Usage: addhost hostname ip"
        return 1
    fi

    HOSTS_LINE="${IP} ${HOSTNAME}"
    if [ -n "$(grep $HOSTNAME /etc/hosts)" ]
    then
        echo "$HOSTNAME already exists : $(grep $HOSTNAME $ETC_HOSTS)"
    else
        echo "Adding $HOSTNAME to your $ETC_HOSTS";
        sudo -- sh -c -e "echo '$HOSTS_LINE' >> /etc/hosts";
    fi
}
```

```

        if [ -n "$(grep $HOSTNAME /etc/hosts)" ]
        then
            echo "$HOSTNAME was added succesfully \n $(grep $HOSTNAME
/etc/hosts)";
        else
            echo "Failed to Add $HOSTNAME, Try again!";
        fi
    fi
}

```

```

function fun_add_hosts(){
    ip=$1
    username=$2
    private_key=$3

    for node in "${!HostIP[@]}"
    do
        ssh -o StrictHostKeyChecking=no -i $private_key $username@$ip \
            "$(< common.sh);add_host $node ${HostIP["$node"]}"
    done
}

```

```

fun_add_hosts 192.168.33.10 vagrant
master/.vagrant/machines/default/virtualbox/private_key
fun_add_hosts 192.168.33.11 vagrant
node1/.vagrant/machines/default/virtualbox/private_key

```

Install Puppet Server

System Requirement

RAM: > 1GB (default 2GB)

Core: 2

Step 1: Reset the puppet server default memory setting

```

./hosts_ip_map.sh

```

```

declare -A HostIP
HostIP=( ["master.smb.com"]="192.168.33.10" \
          ["node1.smb.com"]="192.168.33.11")

```

```
node=master.smb.com
ip=${HostIP["$node"]}
private_key=master/.vagrant/machines/default/virtualbox/private_key

# Reset puppet-server memory to 1GB and then restart the service.
ssh -o StrictHostKeyChecking=no -i $private_key vagrant@$ip "$(< common.sh);
fun_puppet_master_memory"
```

```
variable="-Xms1g -Xmx1g -XX:MaxPermSize=256m"
sudo sed -i "s#^\.*bJAVA_ARGS\b.*\##JAVA_ARGS=\"${variable}\"#g"
/etc/default/puppetserver
```

```
# restart the service
ssh -o StrictHostKeyChecking=no -i $private_key vagrant@$ip sudo service puppetserver
restart
```

```
# start the service when the node reboot
ssh -o StrictHostKeyChecking=no -i $private_key vagrant@$ip sudo
/opt/puppetlabs/bin/puppet resource service puppetserver ensure=running enable=true
```

Step 2: Install the puppet server

```
#!/bin/bash
echo nameserver 192.168.0.1 >> /etc/resolv.conf
echo 127.0.1.1 master.smb.com >> /etc/hosts

ntpdate pool.ntp.org

cd /tmp
#Ref: https://docs.puppet.com/puppetserver/2.4/install\_from\_packages.html#debian
wget https://apt.puppetlabs.com/puppetlabs-release-pc1-trusty.deb
dpkg -i puppetlabs-release-pc1-trusty.deb
apt-get update
apt-get -y --force-yes install puppet-agent # Because puppetserver depends on agent
apt-get -y --force-yes install puppetserver

# setup the puppet server from 2GB to 1GB
variable="-Xms1g -Xmx1g -XX:MaxPermSize=256m"
sudo sed -i "s#^\.*bJAVA_ARGS\b.*\##JAVA_ARGS=\"${variable}\"#g"
/etc/default/puppetserver

#start the puppet server
sudo service puppetserver restart
```

Verify the version

puppet help

```
See 'puppet help <subcommand>' for more information.
See 'puppet help <subcommand>' for more information.
Puppet v4.5.0
vagrant@master:~$
```

Debug

Log

/var/log/puppetlabs/puppetserver

Install Puppet Agent

System Requirement

RAM: no special requirement

Core: no special requirement

Step 1: Install the puppet agent

```
echo nameserver 192.168.0.1 >> /etc/resolv.conf
echo 127.0.1.1 node1.smb.com >> /etc/hosts

# setup ntp service
ntpdate pool.ntp.org

# install puppet agent
cd /tmp
wget https://apt.puppetlabs.com/puppetlabs-release-pc1-trusty.deb
sudo dpkg -i puppetlabs-release-pc1-trusty.deb
sudo apt-get update

sudo apt-get -y --force-yes install puppet-agent
```

Step 2: Start the puppet

```
./hosts_ip_map.sh

declare -A HostIP
HostIP=( ["master.smb.com"]="192.168.33.10" \
```

```
["node1.smb.com"]="192.168.33.11")
```

```
node=node1.smb.com
ip=${HostIP["$node"]}
private_key=node1/.vagrant/machines/default/virtualbox/private_key

# start puppet service
ssh -o StrictHostKeyChecking=no -i $private_key vagrant@$ip \
"sudo /opt/puppetlabs/bin/puppet resource service puppet ensure=running enable=true;
sudo /opt/puppetlabs/bin/puppet agent --test --server master.smb.com"
```

Sign the Request

```
./hosts_ip_map.sh
```

```
declare -A HostIP
HostIP=( ["master.smb.com"]="192.168.33.10" \
        ["node1.smb.com"]="192.168.33.11")
```

```
node=master.smb.com
ip=${HostIP["$node"]}
private_key=master/.vagrant/machines/default/virtualbox/private_key

#Sign node1
# sudo /opt/puppetlabs/bin/puppet cert --sign node1.smb.com
ssh -o StrictHostKeyChecking=no -i $private_key vagrant@$ip \
"sudo /opt/puppetlabs/bin/puppet cert list;
sudo /opt/puppetlabs/bin/puppet cert sign --all"
```

AutoTest

```
./hosts_ip_map.sh
```

```
declare -A HostIP
HostIP=( ["master.smb.com"]="192.168.33.10" \
        ["node1.smb.com"]="192.168.33.11")
```

```
# copy the test file to puppet server
TEST_FILE=./auto-test/test_site.pp
master_node=master.smb.com
master_ip=${HostIP["$master_node"]}
```

```
private_key=master/.vagrant/machines/default/virtualbox/private_key
scp -i $private_key $TEST_FILE vagrant@$master_ip:~/test_site.pp
ssh -o StrictHostKeyChecking=no -i $private_key vagrant@$master_ip sudo cp test_site.pp
/etc/puppetlabs/code/environments/production/manifests/site.pp
```

test deploy

```
node=node1.smb.com
```

```
ip=${HostIP["$node"]}
```

```
private_key=node1/.vagrant/machines/default/virtualbox/private_key
```

```
# sudo /opt/puppetlabs/bin/puppet agent --enable;
```

```
ssh -o StrictHostKeyChecking=no -i $private_key vagrant@$ip \
```

```
"sudo /opt/puppetlabs/bin/puppet agent --test --server master.smb.com"
```

verify

```
ssh -o StrictHostKeyChecking=no -i $private_key vagrant@$ip cat /tmp/temp1.txt
```

References

- <https://www.digitalocean.com/community/tutorials/how-to-install-puppet-4-in-a-master-agent-setup-on-ubuntu-14-04>
- Install puppet Server, https://docs.puppet.com/puppetserver/2.4/install_from_packages.html
- Install puppet agent, https://docs.puppet.com/puppet/4.5/reference/install_linux.html

Appendix

- Puppetmaster vs. puppet server ([ref](#))
- [debug, **syntax** test] How to test the puppet code syntax ([view](#))
- [puppet, add **user**] How to create a user and setup the password ([view](#))
- [puppet, add user **sudo**] Create a user with **passwordless** sudo privilege ([view](#))
- [puppet, passwd-less] How to Enable Password-Less SSH ([view](#))
- [puppet, **variable**] A simple example on puppet variable ([view](#))
- [node, **definition**] How to define a node for assign configuration ([view](#))
- [class, definition] How to define a class ([view](#))
- [module, **install/uninstall**] How to **install/uninstall** puppet module ([view](#))
- [module, **use**] How to use module ([view](#))
- [module, erb, dynamic content] How to define a module, erb example ([view](#))
- [custom type, **define and declare**] How to **define** and **declare** a resource type ([view](#))
- [custom type, **create**] How to create a custom type ([view](#)) -- pure ruby
- [custom type, **provider**] How to implement your custom type using provider ([view](#)) -- pure

ruby

- [datatype, **declaration**] How to use declare a datatype in puppet ([view](#))
- [storeconfig, **exported resource**] What is <| |> -- collection of the resource ([view](#))
- [resource, **dependence**] How to manage the resource **dependence** ([view](#))
 - [resource, relation, **order**] How to setup resource order ([view](#))
- [debug, **message**] How debug message on puppet master/client (view)
- [puppet, **create file**] The experimental environment as code by vagrant + puppet ([view](#))
- [puppet, file, **symbolic link**] How to create symbolic link file ([view](#))
- [puppet, **bash**] How to run a bash script command from puppet ([view](#))
- [puppet, **ppa**] How to setup ppa ([view](#))
- [puppet, **require**] How to setup a task requiring a specified service ready ([view](#))
- [puppet, **dnsmasq**] How to setup a DNS server, dnsmasq ([view](#))
- [datatype, hash] How to use hash datatype in puppet (view)
- [exec, **create**] How to write an exec with create attribute ([view](#))
- [puppet, adrien/**network ip, exec**] How to use automatic setup network interface using adrien/network module with puppet ([view](#))
- [puppet, **gluster**] Install gluster with a specified version ([view](#))
- [puppet, storage] <https://forge.puppetlabs.com/puppetlabs/lvm>
- http://finninday.net/wiki/index.php/Zero_to_puppet_in_one_day
- [puppet, **hostname**] How to setup hostname ([view](#))
- [puppet, **zfs**, installation] How to install **zfs** for Ubuntu ([view](#))
- [playbooks, installation] How to use playbook ([ref](#))