How to setup a DNS server by vagrant with puppet using dnsmasq

Jing, mqjing@gmail.com

ETA: 5 mins

Google doc: (view)
Download: lab
vm1: 192.168.50.10 (DNS)
vm2: 192.168.50.11

Simple DNS Working Model

1. Setup static IP for each VM by Vagrantfile
2. Using puppet to setup the A record

Quick

[host] install VirtualBox
[host] install vagrant
    sudo apt-get install vagrant
[host] install vagrant box
    vagrant box add ubuntu/trusty64
[host] install puppet
    sudo apt-get install puppetmaster
[host] install lex-dnsmasq module (detail)
    puppet module install lex-dnsmasq

Download: lab
start vm1
    [host] ~/vm1: vagrant up && vagrant ssh

start vm2 to test
    [host] ~/vm2: vagrant up && vagrant ssh
    [vm2] ~/vm2: sudo vi /etc/resolv.conf --> nameserver 192.168.50.10
        note: for puppet manage the resolv, you can install gdsoperations/resolvconf (ref)

check domain name server (vm1)
    [vm2] ping vm1

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo vi /etc/resolv.conf
vagrant@vagrant-ubuntu-trusty-64:~$ ping vm1
PING vm1 (192.168.50.10) 56(84) bytes of data.
64 bytes from 192.168.50.10: icmp_seq=1 ttl=64 time=0.115 ms
64 bytes from 192.168.50.10: icmp_seq=2 ttl=64 time=0.270 ms
64 bytes from 192.168.50.10: icmp_seq=3 ttl=64 time=0.276 ms
64 bytes from 192.168.50.10: icmp_seq=4 ttl=64 time=0.270 ms
```

Detail

Requirement

- VirtualBox
- Vagrant
  - vagrant box add ubuntu/trusty64
- Puppet
  - sudo apt-get install puppetmaster

Module

puppet module install lex-dnsmasq

ref: https://forge.puppetlabs.com/lex/dnsmasq

File

```
jing@jing-Lenovo-B50-70:~/test/lex-puppet-dnsmasq$ tree
.
├── readme.txt
├── vm1
│   ├── manifests
│   │   └── default.pp
│   └── Vagrantfile
└── vm2
    └── Vagrantfile
```

vm1 (Domain name server)

default.pp

```
# Setup DHCP server
# by Jing.

node default {
```

```
  include dnsmasq
  # Setup DHCP
  dnsmasq::dhcp { 'dhcp':
    paramset   => 'hadoop0' ,
    dhcp_start => '192.168.1.100',
    dhcp_end   => '192.168.1.200',
    netmask    => '255.255.255.0',
    lease_time => '24h',
   }

  # Add A Records
  dnsmasq::address { "vm1": ip  => '192.168.50.10',  }
  dnsmasq::address { "vm2": ip  => '192.168.50.11',  }

}
```

vm1::Vagrantfile

```
...
  config.vm.provision "puppet" do |puppet|
     puppet.module_path ="~/.puppet/modules"
     puppet.options = "--verbose --debug"
  end

  config.vm.network "private_network", ip: "192.168.50.10"
...
```

Test the Domain Name Server

Step 1: Launch vm1 & deploy the DNS server

vagrant up

check

```
vagrant@vagrant-ubuntu-trusty-64:~$ nslookup
> vm1
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   vm1
Address: 192.168.50.10
> vm2
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   vm2
Address: 192.168.50.11
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ ping vm2
PING vm2 (192.168.50.11) 56(84) bytes of data.
64 bytes from 192.168.50.11: icmp_seq=1 ttl=64 time=0.541 ms
64 bytes from 192.168.50.11: icmp_seq=2 ttl=64 time=0.273 ms
^C
--- vm2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.273/0.407/0.541/0.134 ms
```

Step 2: Check the name server from the other vm (vm2)

vm2::Vagrantfile

```
...

   config.vm.network "private_network", ip: "192.168.50.11"
...
```

```
jing@jing-Lenovo-B50-70:~/test/lex-puppet-dnsmasq$ tree
.
├── readme.txt
├── vm1
│   ├── manifests
│   │   └── default.pp
│   └── Vagrantfile
└── vm2
    └── Vagrantfile
```

(a) Create and Launch the vm

[host]~/vm2$ vagrant up && vagrant ssh

(b) Assign the name server

[vm2] sudo vi /etc/resolv.conf

/etc/resolv.conf

```
nameserver 192.168.50.10
```

check

[vm2] ping vm1

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo vi /etc/resolv.conf
vagrant@vagrant-ubuntu-trusty-64:~$ ping vm1
PING vm1 (192.168.50.10) 56(84) bytes of data.
64 bytes from 192.168.50.10: icmp_seq=1 ttl=64 time=0.115 ms
64 bytes from 192.168.50.10: icmp_seq=2 ttl=64 time=0.270 ms
64 bytes from 192.168.50.10: icmp_seq=3 ttl=64 time=0.276 ms
64 bytes from 192.168.50.10: icmp_seq=4 ttl=64 time=0.270 ms
```

Question:
Dynamic DNS update: How do I configure DNSMASQ to use hostnames for computers who get their IP addresses from DHCP.

- Bind (ref)
- DNS ==> hostname <-> mac address?

```
dnsmasq::dhcpstatic { 'example-host':
  mac => 'DE:AD:BE:EF:CA:FE',
  ip  => '192.168.1.10',
}
```

ref: https://forge.puppetlabs.com/lex/dnsmasq

Further Reading

- The best bind tutorial, https://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-private-network-dns-server-on-ubuntu-u14-04
- [puppet, bind] camptocamp/bind, https://forge.puppetlabs.com/camptocamp/bind/readme
- [dnsmsq] 動態更新 dnsmq 的 DNS A record 資料, http://www.lightofdawn.org/wiki/wiki.cgi/LocalDynamicDns
- [bind] Dynamically update DNS A record by DHCP server, http://lani78.com/2012/07/23/make-your-dhcp-server-dynamically-update-your-dns-records-on-ubuntu-12-04-precise-pangolin/
-