Getting started with Ansible

Contents

Be sure you connect via ssh to your workshop as the "nsrc" user. Don't become root except where indicated.

Starting with a fresh Ubuntu 14.04 install, we will set up ansible to get the workshop server to configure itself.

1 Install packages

Ubuntu 14.04 already has a sufficiently-recent version of ansible for our purposes [1], so you can just install it directly, along with the 'git' version control system.

```
sudo apt-get install ansible git
```

2 Download the workshop-kit files

This should be done as the non-root (nsrc) user, in your home directory. How you do it depends on whether you have been given full access to the git repository. For purposes of class we will use git over http with passwords.

```
cd
git clone https://git.nsrc.org/nsrc/workshop-kit.git
```

You will see the following prompts:

```
Username:
Password:
```

Enter in the username and password you have been given by your instructors. Once completed you will have the files needed to complete preparing your machine in a workshop-kit directory under your home directory.

3 Configure ansible

`sudo vi /etc/ansible/ansible.cfg` and uncomment the following line:

```
...
host_key_checking = False
...
nocows = 1
```

[1] The ansible PPA provides a more recent version of ansible. If you need it, install as follows:

```
sudo apt-get install software-properties-common  # (12.04: python-software-properties)
sudo add-apt-repository ppa:rquillo/ansible
sudo apt-get update
sudo apt-get install ansible
```

(If you want to see the cows, then leave `nocows` commented out, and also run `sudo apt-get install cowsay`)

Finally, in your checkout's ansible directory, copy `hosts.sample` to `/etc/ansible/hosts`

```
cd
cd workshop-kit/ansible
sudo cp hosts.sample /etc/ansible/hosts
```

4 First run: network reconfiguration

Still inside the ansible directory, run the following commands:

```
ansible-playbook -sK networking.yml --check --diff     # dummy run
ansible-playbook -sK networking.yml                    # live run
```

The flags `-s` and `-K` instruct ansible to use sudo to run actions as root, and to prompt for the sudo password.

With the `--check` flag it will show you what it is going to change; without the `--check` flag it will actually perform the changes. Try both.

The playbook is *idempotent*. This means it is safe to run it repeatedly; anything which is already in the correct state will not be changed. Feel free to run it again.

After the live run, have a look at the following files:

- /etc/hosts
- /etc/network/interfaces
- /etc/iptables/rules.v4

The files configuring the interfaces have now been moved to `/etc/network/interfaces.d/*.cfg`. They contain a more complex configuration with a bridge interfaces for the LAN and WAN connections (br-lan contains eth0, br-wan contains eth1), an IP alias on 10.10.0.254, and NAT rules.

If you are happy with this configuration, reboot your server to activate the new interfaces.

```
sudo reboot
```

You should not need to re-run `networking.yml` again unless you want ansible to reconfigure your network.

4.1 How does it work?

Have a look at the playbook:

```
cd workshop-kit/ansible
cat networking.yml
```

Notice how it contains:

- The host(s) or group(s) on which to run this playbook
- What tasks or roles to apply
- Tags, to allow parts of the playbook to be run selectively

Have a look at the tasks and handlers contained under each role. Handlers are extra actions which are triggered at the end of the run if a task has changed something.

```
cat roles/update_cache/tasks/main.yml
cat roles/ansible_base/tasks/main.yml
cat roles/networking_ubuntu/tasks/main.yml
cat roles/networking_ubuntu/handlers/main.yml
cat roles/networking_ubuntu/templates/interfaces
cat roles/gateway_ubuntu/tasks/main.yml
cat roles/gateway_ubuntu/handlers/main.yml
```

Some of these roles and templates are quite complex - don't worry about the details.

There are also variables which are set per host and group. You can find these in the inventory (hosts) and in files under host_vars/ and group_vars/

```
cat /etc/ansible/hosts
cat host_vars/s1.ws.nsrc.org
cat group_vars/vm_servers
cat group_vars/all
```

In particular, notice that there are variables in `host_vars/s1.ws.nsrc.org` which are used by the `interfaces` template:

```
lan_address: 10.10.0.241
gateway_wan_interfaces: [br-wan, wlan0]
```

If you change these, running ansible will update your `/etc/network/interfaces.d/br-lan.cfg` file.

5 Second run: platform configuration

```
ansible-playbook -sK vm_servers.yml -t platform
```

This installs a few basic packages. If your machine is a Mac Mini then it will install packages specific to that platform, e.g. macfanctld.

`-t platform` means only to run those tasks labelled with tag "platform".

If you see this step hang for more than a few seconds:

```
TASK: [install macmini packages] **********************************************
```

then login through another console or ssh session and run

```
sudo service macfanctld restart
```

(This is a bug in the macfanctld package)

6 Third run: full configuration

Now run the vm_servers.yml playbook again, but without restricting to the platform tag.

```
ansible-playbook -sK vm_servers.yml
```

This will run through a full set of configuration including:

- dns server
- ntp server
- snmp agent
- apt-cacher
- dhcp server
- kvm
- vmbuilder
- dynamips

This may take a fair amount of time to complete.

You can look around and see how your server has changed. For instance, you are now running an snmp server:

```
snmpwalk -v2c -c NetManage localhost ifTable
```

DNS for your private network that you will be using has been configured:

```
dig pc10.ws.nsrc.org
dig sw.ws.nsrc.org
```

You might to consider looking at current running processes, use netstat to see your current routes or open ports, typing ifconig to see all your interfaces, etc... to get a feel for your new workshop server environment.

Finally, you should have functioning DHCP on your LAN network. If you had configured a static IP on your laptop, remove it, and then re-connect to your local network. You should receive an address between 10.10.0.100 and 10.10.0.219 with a gateway of 10.10.0.254, which is your server, a DNS server of 10.10.0.241 (again, your server) and you will have full access to the public internet.

6.1 Refreshing your configuration

Over time, the NSRC ansible configuration will be changed and improved.

To get the latest version, make sure you are inside either the 'workshop-kit' directory or any subdirectory under 'workshop-kit', and type this command:

```
git pull
```

Enter your gitlab username and password as before. Only the items which have changed since you cloned the repo will be fetched.

Now make sure you are inside the workshop-kit/ansible directory:

```
cd
cd workshop-kit/ansible
ansible-playbook -sK vm_servers.yml
```

to apply any changes.

7 Appendix

7.1 Problem building dynamips

If you get a failure during the **dynamips** section that looks like this it may be a problem with the DNS that has just been built by the playbook.

```
TASK: [dynamips | apt_repository repo=ppa:gns3] ******************************
failed: [s1.ws.nsrc.org] => {"failed": true}
msg: Failed to validate the SSL certificate for launchpad.net:443. Use validate_certs=no or

FATAL: all hosts have already failed -- aborting

PLAY RECAP ********************************************************************
           to retry, use: --limit @/home/nsrc/vm_servers.retry
```

Check out the information in the file **dns-issues.md** for a possible fix.

7.2 Using git

Detailed instructions for using git with ssh and configuring it for use for interactive work flow.

This assumes you have an account on git.nsrc.org and your ssh public key has been installed there.

```
cd
git clone git@git.nsrc.org:nsrc/workshop-kit.git
```

If authentication is rejected, you should NOT copy your ssh private key onto your workshop server! Rather you should:

1. Disconnect your ssh session
2. Log back in using ssh and agent forwarding
   - For Linux and OSX:
     - ssh-add
       - enter your passphrase when prompted
     - ssh -oForwardAgent=yes nsrc@x.x.x.x
   - For Windows/putty:
     - run pageant
     - point it to your private key and enter your passphrase
     - connect with agent forwarding enabled

7.2.1 Configure git

Create a file `~/.gitconfig` containing the following. This ensures that any commits you make are labelled with your correct details.

```
[user]
        name = Your Fullname
        email = yourname@yourdomain
[core]
        excludesfile = ~/.gitignore
        #editor = /usr/bin/joe   << or whatever you prefer
[push]
        default = tracking
```

And create `~/.gitignore` as follows: this is to minimise the junk which is picked up.

```
*~
```

8 Obtaining workshop kit files with tarballs

8.1 From a tarball

Alternatively, you may be given a tarball or zipfile containing a snapshot of the repository. Download it using wget, and extract it as appropriate:

```
cd
wget http://...../workshop-kit.tgz
tar -xvzf workshop-kit.tgz

or:
wget http://...../workshop-kit.zip
unzip workshop-kit.zip
```

Either way, you should have a directory called "workshop-kit" which you can cd into.