



Chapter 4

Lab 4.3

Objective: Deploying a Java application with Ansible infrastructure automation

Introduction to Ansible

Ansible is a configuration management tool with a different concept than Puppet or Chef. With Ansible, you work from a management node, which then uses SSH to log into your managed nodes to make the configuration changes. Other than SSH keys, Ansible requires no configuration in the client nodes - it is agentless. The management node can be your workstation, but ideally, Ansible is run as close to the deployed system as possible. For smaller installations, it may be applicable to use your workstation. In this example, you will build two systems: the Ansible management machine and the client application node.

Setting Up the Client Node

Step 1. Launch a second Ubuntu virtual machine on Vagrant:

1. Use the following command:

```
$ vagrant init ubuntu/trusty64
```

2. Set up port forwarding from `guest:8080` to `host:8083`.
3. Add an IP to the Vagrant box; under `config.vm.box = "ubuntu/trusty64"` add this line:
`config.vm.network "private_network", ip: "192.168.50.4" - vagrant up.`

All Vagrant configuration is done below. The "2" in `vagrant.configure` configures the configuration version (we support older styles for backwards compatibility). Please do not change it unless you are proficient in this topic.

Example of client `Vagrantfile`:

```
# -*- mode: ruby -*-  
# vi: set ft=ruby :  
  
Vagrant.configure("2") do |config|  
  #setup base box  
  config.vm.box = "ubuntu/trusty64"
```

```
#set IP
config.vm.network "private_network", ip: "192.168.50.4"

#port forwarding
config.vm.network "forwarded_port", guest: 8080, host: 8083

end
```

The Ansible control system is ready to test from the management node.

Building the Ansible Management Node

Step 1. On your Ubuntu virtual machine containing Artifactory install and configure Ansible

1. The VM needs its own IP similar to the client. In the `Vagrantfile` put the following contents above the forwarded ports:

```
$ config.vm.network "private_network", ip: "192.168.50.5"
```

2. To apply these changes, use the command presented below:

```
$ vagrant reload --provision
```

3. Install Ansible dependencies:

```
$ sudo apt-get install software-properties-common
```

4. Add the personal package archive for Ansible:

```
$ sudo apt-add-repository ppa:ansible/ansible
```

5. Update the package list:

```
$ sudo apt-get update
```

6. Install Ansible:

```
$ sudo apt-get install ansible
```

Step 2. Create a keypair for the ssh connections between nodes

1. Generate a new keypair `ssh-keygen -t rsa -b 4096 -C 'ansible'`.
2. Copy the public key onto the client node `scp ~/.ssh/id_rsa.pub 192.168.50.4:~/.ssh/authorized_keys`. If a password is requested, the default is `"vagrant"`.

3. Edit the `/etc/ansible/hosts` file and add the following:

```
[app]
192.168.50.4
```

Testing Ansible

Step 1. The following command can be used from the management node to control the client:

```
$ ansible app -s -m ping
```

The output should return the following:

```
192.168.50.4 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Note: It is important to give your Ansible control user the same name across all client nodes. You can specify a different user at the Ansible command line, but it is far easier to omit this.

Build the Ansible Playbook to Install Petclinic

Ansible configuration scripts are called “playbooks.” You will build a collection of playbooks similar to the Chef and Puppet solutions. Again, for simplicity’s sake, you are building these playbooks specifically for your Ubuntu environment.

Step 1. On your control node, you will create a working directory and build playbooks inside it

1. Create a `playbook` directory:

```
$ mkdir -p playbooks/roles
```

2. Navigate to the new directory:

```
$ cd playbooks/roles
```

3. Initialize a Java role from the Ansible community:

```
$ ansible-galaxy init java
```

4. Edit the `/roles/java/tasks/main.yml` file to match the following:

```
- name: ensure required packages are installed for Java 7
  apt: name={{item}} state=latest update_cache=yes
```

```

with_items:
  - python-software-properties

- name: Add Java repository to sources
  action: apt_repository repo='ppa:webupd8team/java'

- name: Autoaccept license for Java
  action: shell echo oracle-java7-installer
shared/accepted-oracle-license-v1-1 select true | sudo
/usr/bin/debconf-set-selections

- name: Update APT package cache
  action: apt update_cache=yes

- name: Install Java 7
  action: apt pkg=oracle-java7-installer state=latest install_recommends=yes

- name: Set Java 7 Env
  action: apt pkg=oracle-java7-set-default state=latest
install_recommends=yes

```

5. Create a `tomcat` role (from the roles directory):

```
$ ansible-galaxy init tomcat
```

6. Edit the `/roles/tomcat/tasks/main.yml` file to match the following:

```

# tasks file for tomcat

- name: Install tomcat
  apt: name=tomcat7

- name: Fix permissions on tomcat user homedir
  file:
    path: /usr/share/tomcat7
    state: directory
    owner: tomcat7
    group: tomcat7
    mode: 0755

```

7. Edit the `/roles/tomcat/handlers/main.yml` file to match the following:

```

# handlers file for tomcat

- name: restart tomcat
  service: name=tomcat7 state=restarted

```

8. Create a `petclinic` role (from the `roles` directory):

```
$ ansible-galaxy init petclinic
```

9. Edit the `/roles/petclinic/tasks/main.yml` file to match the following:

```
# tasks file for petclinic

#dependencies:
# - { role: tomcat }

- name: petclinic
  get_url:
    url:
'http://192.168.50.5:8081/<artifactory_path_to_petclinic>/<name_of_the_petcli
nic_artifact>'
    dest: /var/lib/tomcat7/webapps/petclinic.war
    owner: tomcat7
    group: tomcat7
  notify:
    - restart tomcat
```

10. Finally, in the `playbook` directory create a `site.yml` file with the following:

```
- hosts: app
  become: yes
  become_method: sudo

  roles:
    - java
    - tomcat
    - petclinic
```

With those files created, you can simply run the playbooks with the following command:

```
$ ansible-playbook site.yml
```

To see the running `petclinic`, browse to `http://localhost:8083/petclinic`.