

MitziCom: 30- to 40-hour proof-of-concept (POC) using Red Hat
Ansible Tower
Advanced Deployment with Red Hat Ansible Automation - Final Lab

version 0.0

Author: ir. Jan Gerrit Kootstra

24th June 2021

Contents

1	Introduction	2
1.1	Business Use Case	2
1.2	a list of environment variables and their values needed in the whole document	2
1.3	needed items	2
I	The proof of Concept	3
2	Create Lab Environment	4
2.1	Set Up Lab Environment	4
2.2	Review Notes About Ansible Tower Installation	5
2.3	Connect to the Ansible Tower Home Lab environment	5
2.3.1	Set Up Control host	5
2.3.2	internal network and subnet	7
2.3.3	external network and subnet	8
2.3.4	router	9
2.3.5	security groups	10
2.3.6	flavor	11
2.3.7	key pairs	11
II	After the Proof of Concept	12
3	Review Use Case	13

List of Tables

2.1	credential for tower webUI	5
2.2	Internal network details	7
2.3	Internal subnet details	7
2.4	External network details	8
2.5	External subnet details	8
2.6	Router details	9
2.7	Security Group and Rules for apps instances	10
2.8	Security Group and Rules for frontend instance	10
2.9	Security Group and Rules for DB instance	10
2.10	Details for flavor	11
2.11	Key Pairs details	11

List of Figures

Chapter 1

Introduction

the document is split in two parts:

- part 1 describes the Proof of Concept which meant to review the following Business use case as written down in section [1.1](#)
- part 2 will discuss the implementation of a production and QA environment after the successful POC

1.1 Business Use Case

The purpose of the POC is to determine the feasibility of using Ansible Tower as a CI/CD tool for automating continuous deployment of an internal three-tier application on QA and production environments. Note that this requires maintaining several instances of the application.

1.2 a list of environment variables and their values needed in the whole document

```
export TOWER_GUID="21dc"  
export OSP_GUID="9a1f"  
export OSP_DOMAIN="dynamic.opentlc.com"  
export OPENTLC_ID="jangerrit.kootstra_kpn.com"  
export MAIL_ID="jangerrit.kootstra@kpn.com"  
export OPENTLC_PASSWORD="Jgk671221!"  
export GITHUB_REPO="https://github.com/jgkootstra/nextgen_ansible_advanced_homework.git"  
export JQ_REPO_BASE="http://www.opentlc.com/download/ansible_bootcamp"  
export REGION="us-east-1"
```

1.3 needed items

a laptop or desktop to login to the lab environments.

We will use the prompt starting with [laptop] to indicate that a cli command is started on the laptop or desktop of the reader.

Part I

The proof of Concept

Chapter 2

Create Lab Environment

We will need 3 different environments to be able to run our setup:

- an Ansible tower cluster with 3 nodes
we use the Ansible Advanced - Homework environment for this.
- an Red Hat OpenStack Platform with 1 node
we use Ansible Advanced NG - OpenStack
- a three tier application setup to simulate the customers main application.
we use Ansible Advanced - Three Tier App

Reserve 45 minutes to deploy each of these enviroments in parallel.

2.1 Set Up Lab Environment

In this section, you provision the lab environments needed to develop the POC. Playbooks are provided to help you set up the SSH keys required to connect to the Red Hat OpenStack Platform (RHOSP) environment.

First, you provision the Ansible Advanced - Homework and Ansible Advanced NG - OpenStack lab environments. You provision the Ansible Advanced - Three Tier App environment in a later section.

1. Navigate to the OPENTLC lab portal and log in using your OPENTLC credentials.
If you have forgotten your credentials, visit the OPENTLC Account Management page.
2. Navigate to Services Catalogs All Services OPENTLC Automation.
Select Ansible Advanced - Homework and click Order.
Select Training - as part of course for Purpose.
Click Submit.
3. Navigate to Services Catalogs All Services OPENTLC Automation.
Select Ansible Advanced NG - OpenStack and click Order.
Select check box Check this box to confirm that you understand and know your runtime and expiration dates.
Select Training - as part of course for Purpose.
Click Submit.

4. Check your email for messages from Red Hat describing how to connect to the environments, including your GUIDs (unique identifiers) and environment details.

Further details are provided in the lab instructions.

2.2 Review Notes About Ansible Tower Installation

- Ansible Tower is deployed in the Ansible Advanced - Homework environment.
- The Ansible Tower isolated node is deployed on workstation- OSP_{GUID} .OSP_DOMAIN in the Ansible Advanced NG - OpenStack environment

The credentials are based on the table [2.1](#)

Name	Details
URL	https://tower(1-3).TOWER_GUID.example.opentlc.com
Username	admin
Password	As provided in Email

Table 2.1: credential for tower webUI

2.3 Connect to the Ansible Tower Home Lab environment

```
[laptop]$ ssh <user-id>@control.<guid>.example.opentlc.com
password: <Enter password as provided>
[user@control ~]$ sudo -i
[root@control ~]#
```

2.3.1 Set Up Control host

Copy your OPENTLC private key to the Ansible Advanced - Homework control host under the root users home directory.

The path to your OPENTLC key is /root/.ssh/mykey.pem.

Set the permissions of the /root/.ssh/mykey.pem file to 0400 and set the owner as root:root.

We do not expect all students to have Linux or Apple laptops. This is why the lab must be performed from the control host, and why you must copy the OPENTLC key to the control host. The OPENTLC key is the private key that you use to connect to the lab environments from your laptop.

Fork the https://github.com/redhat-gpte-devopsautomation/nextgen_ansible_advanced_homework.git repository in your GitHub account.

The forked repository acts as the skeleton of the project.

After forking the repository, clone the forked repository from your account onto the control host of the Ansible Advanced - Homework lab environment under the /root directory.

The repository contains playbooks configured to help you with your POC.

The README file in the repository contains the list of playbooks and details of each.

Ansible Playbooks are provided to provision the network, security groups, flavor, and key pairs on OpenStack.

From the cloned repo copy labrc file in your home dir and edit to setup env variables which we will be using in the whole lab.

```
[root@control ~]# git clone https://github.com/<github handler>/nextgen_ansible_advanced_homework
[root@control ~]# cd nextgen_ansible_advanced_homework
[root@control ~/nextgen_ansible_advanced_homework]# cp labrc ~/
[root@control ~/nextgen_ansible_advanced_homework]# cat ~/labrc
```

the content of labrc is shown in [section 1.2](#) on [page 2](#)

```
[root@control ~]# source ~/labrc
```

Please make sure you have copied the labrc file to your homedir before editing it as it contains sensitive information.

From the cloned repository, run the site-setup-prereqs.yaml playbook to provision the network, flavor, security groups, and SSH keys:

```
[root@control ~/nextgen_ansible_advanced_homework]# mv ~/ansible-tower-setup-*/ ~/ansible-tower-setup/
[root@control ~/nextgen_ansible_advanced_homework]# cp /etc/ansible/hosts ~/ansible-tower-setup/
[root@control ~/nextgen_ansible_advanced_homework]# chmod 0400 /root/.ssh/openstack.pem
[root@control ~/nextgen_ansible_advanced_homework]# ansible-playbook site-setup-prereqs.yaml
```

Password:;copy your OSP workstation password from email;

The playbook sets up openstack.pem as a private key and openstack.pub as a public key for connecting to the workstation host from the control host.

The playbook also sets up workstation- $\{\text{OSP_GUI}\}$. $\{\text{OSP_DOMAIN}\}$ as an isolated node.\\

Verify a successful connection between the control and workstation hosts using the openstack.pem key:

```
[root@control ~/nextgen_ansible_advanced_homework]# ssh -i /root/.ssh/openstack.pem cloud-user@workstation
[cloud-user@workstation ~]$ exit
From your web browser, connect to tower1. $\{\text{TOWER\_GUID}\}$ .example.opentlc.com to verify
```

Navigate to Instance Groups from the side panel and select osp.

The isolated node helps you in provisioning OpenStack instances and deploying a three-tier application on RHOSP instances.

You do not have direct access to RHOSP instances because they are running behind a firewall.

OpenStack components are provisioned for you and are created by the site-setup-prereqs.yaml playbook that you ran earlier.

RHOSP-hosted network details are provided here:

2.3.2 internal network and subnet

Parameter	value
cloud	openstack
state	present
name	int_network
external	false
admin_state_up	yes
provider_network_type	null
provider_physical_network	null

Table 2.2: Internal network details

Parameter	value
cloud	openstack
state	present
name	int_subnet
network_name	int_network
cidr	172.16.0.0/24
ip_version	4
gateway_ip	null
enable_dhcp	yes
allocation_pool_start	null
allocation_pool_end	null
dns_nameservers	8.8.8.8

Table 2.3: Internal subnet details

The private network is a non-routable network.

2.3.3 external network and subnet

Following are details are for external network and subnet.

Parameter	value
cloud	openstack
state	present
name	ext_network
external	true
admin_state_up	yes
provider_network_type	flat
provider_physical_network	datacentre

Table 2.4: External network details

Parameter	value
cloud	openstack
state	present
name	ext_subnet
network_name	ext_network
cidr	192.0.2.0/24
ip_version	4
gateway_ip	192.0.2.254
enable_dhcp	no
allocation_pool_start	192.0.2.150
allocation_pool_end	192.0.2.200
dns_nameservers	8.8.8.8

Table 2.5: External subnet details

2.3.4 router

Following are details for router to connect between the internal network and external physical network.

Parameter	value
cloud	openstack
state	present
name	router
network_name	ext_network
interfaces	int_subnet

Table 2.6: Router details

2.3.5 security groups

Parameter	value
security_group	apps
port	22/tcp
port	8080/tcp
remote_ip_prefix	0.0.0.0/0

Table 2.7: Security Group and Rules for apps instances

Parameter	value
security_group	frontend
port	22/tcp
port	80/tcp
port	443/tcp
remote_ip_prefix	0.0.0.0/0

Table 2.8: Security Group and Rules for frontend instance

Parameter	value
security_group	db
port	22/tcp
port	5432/tcp
remote_ip_prefix	0.0.0.0/0

Table 2.9: Security Group and Rules for DB instance

2.3.6 flavor

Parameter	value
cloud	openstack
state	present
name	m1.medium
ram	2048
vcpus	1
disk	20

Table 2.10: Details for flavor

2.3.7 key pairs

Parameter	value
name	ansible_ssh

Table 2.11: Key Pairs details

A Red Hat Enterprise Linux image named **rhel-guest** for provisioning RHOSP instances.

Part II

After the Proof of Concept

Chapter 3

Review Use Case