Lab Work Task. Day 2

Task

A) On Control Machine:

Create folder ~/ansible/day2. All working files should be placed there (Vagrantfile, playbooks, roles folder with custom roles, inventory)

Picture 1

B) Spin up 2 VMs form single Vagranfile. Compose Inventory:

- Specify groups;
- Define group vars;
- Define host vars;
- Define ansible connection settings (common for all hosts/groups in inventory).

Picture 2.1 – Inventory file

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  # create wevserver VM
  config.vm.define "centos-webserver" do |webserver|
    webserver.vm.box = "sbeliakou/centos"
    webserver.vm.box_check_update = false
    webserver.vm.network :private_network, ip: "178.65.0.5"
    webserver.ssh.insert_key = false
    webserver.vm.hostname = "webserver"

    webserver.vm.provider "virtualbox" do |vb|
      vb.name = "centos-webserver-day2"
      vb.memory = "1024"
    end
  end

  # create appserver VM
  config.vm.define "centos-appserver" do |appserver|
    appserver.vm.box = "sbeliakou/centos"
    appserver.vm.box_check_update = false
    appserver.vm.network :private_network, ip: "178.65.0.6"
    appserver.ssh.insert_key = false
    appserver.vm.hostname = "appserver"

    appserver.vm.provider "virtualbox" do |vb|
      vb.name = "centos-appserver-day2"
      vb.memory = "2048"
    end
  end

end
```

Picture 2.2 – Vagrantfile

```
[student@EPBYMINW0910 Ansible]$ cat ./Day02/inventory
[webservers]
httpd           ansible_host=178.65.0.5 ansible_connection=paramiko ansible_user={{ user_name }}

[appservers]
tomcat          ansible_host=178.65.0.6 ansible_connection=paramiko ansible_user={{ user_name }}
[student@EPBYMINW0910 Ansible]$
```

Picture 2.3 – cat inventory file

C) Develop following Roles:

- common (provides common system updates/configuration/so on);
- devops_user (creates “devops:devops” user, updates authorized keys file, configures sudoers, disables requiretty setting for this user);

Picture 3.1

Picture 3.2 – Proof

– java (installs java, verifies java installation; supports java version customization);

Picture 3.3

Picture 3.4

Picture 3.5 – Check Java version on appserver

– tomcat (installs tomcat from sources - war, creates tomcat user with custom uid, creates tomcat user's group with custom gid, configures systemd to start tomcat automatically, verifies tomcat installation; supports tomcat customization: JAVA_OPTS, version, etc; role has dependency on java role);

Picture 3.6

Picture 3.7

Picture 3.8 – JAVA_OPTS

Picture 3.8 – JAVA_HOME

Add JAVA to /role/tomcat/meta/main.yml

```
dependencies:
- role: java
  java_version: 1.8.0
```

Picture 3.8

– httpd (installs apache httpd service, forwards requests to tomcat through custom VirtualHost, verifies http installation and running state).

Picture 3.8

Picture 3.8 – Check proxy settings and service stratus

Picture 3.9 – Check results

D) Make sure that all necessary application state changes are being done with handlers on demand.

Picture 4.1

Picture 4.2

And httpd handlers you may see in /role/httpd/handlers/main.yml

Picture 4.3 – Restart service after change proxy configs

E) All roles should leave facts about configured/provisioned software on the system (Verify system's configuration with ad-hoc command – setup module - after provisioning)

```
student@EPBYMINW0910:~/devosplab/Ansible/Day02/roles/httpd/tasks
student@EPBYMINW0910:~/devosplab/Ansible/Day02/roles/httpd/tasks 73x33
- name: httpd installation
  yum:
    name: httpd
    state: present

- name: httpd service enable
  service:
    name: httpd
    state: started
    enabled: yes
  become: yes
  become_user: root

- package_facts:

- name: create ansible facts folder httpd
  file:
    path: /etc/ansible/facts.d
    state: directory
  become: yes

- name: httpd save deatails
  copy:
    content: |
      { "version:" "{{ ansible_facts.packages.httpd[0].version }}" }
    dest: /etc/ansible/facts.d/httpd.fact
  become: yes
~
~
~
~
~
"main.yml" [noeol] 27L, 507C                    19,5          All
```

Picture 5

F) Develop provision.yml playbook:

– Sections pre_tasks, tasks, roles, post_tasks;

```
TASK [install net-tools] ***********************************************************************************
ok: [httpd] => {"changed": false, "msg": "", "rc": 0, "results": ["net-tools-2.0-0.24.20131004git.el7.x86_64 providing n
et-tools is already installed"]}
ok: [tomcat] => {"changed": false, "msg": "", "rc": 0, "results": ["net-tools-2.0-0.24.20131004git.el7.x86_64 providing
net-tools is already installed"]}

TASK [creating_user : user create] *************************************************************************
```

Picture 6.1 – Pre_tasks install net-tools

```
TASK [debug] ***********************************************************************************************
ok: [httpd] => {
    "msg": "Works at the end"
}
ok: [tomcat] => {
    "msg": "Works at the end"
}

PLAY RECAP *************************************************************************************************
httpd                      : ok=12   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
tomcat                     : ok=18   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[student@EPBYMINW0910 Day02]$
```

Picture 6.1 – Post_tasks debag info

– Specify tomcat version in playbook, inventory (group vars, host vars), extra vars – try different.

Picture 7.1 – Filed “user_name” is hleb

```
$ ansible-playbook -i inventory playbook.yml --extra-vars
"{\"user_name\":\"devops\"}" -u root -k -v
```

Picture 7.2 – Successful trying extra vars

Picture 7.3 – User not changed

This is useful for, among other things, setting the hosts group or the user for the playbook. Difference only in the precedence.

Here is the order of precedence from least to greatest (the last listed variables winning prioritization):

1. command line values (eg “-u user”)
2. role defaults
3. inventory file or script group vars
4. inventory group_vars/all
5. playbook group_vars/all
6. inventory group_vars/*
7. playbook group_vars/*
8. inventory file or script host vars
9. inventory host_vars/*
10. playbook host_vars/*
11. host facts / cached set_facts
12. play vars
13. play vars_prompt
14. play vars_files

15. role vars (defined in role/vars/main.yml)
16. block vars (only for tasks in block)
17. task vars (only for the task)
18. include_vars
19. set_facts / registered vars
20. role (and include_role) params
21. include params
22. extra vars (always win precedence)