Lab Work Task. Web Server Provisioning

Review

Using Ansible v2.2.1 for provisioning nginx + tomcat application stack.

Learning by doing.

Task

On Host Node (Control Machine):

1. Create folder ~/cm/ansible/day-2. All working files are supposed to be placed right there.
2. Spin up clear CentOS6 VM using vagrant (repo with vagrantfile). Verify connectivity to the host using ssh keys (user: vagrant)
3. Create ansible inventory file (name: **inventory**) with remote host connection details:
   - Remote VM hostname/ip/port
   - Remote ssh log in username
   - Connection type
4. Develop a playbook (name: **site.yml**) which is supposed to run against any host (specified in inventory)

   4.1 Develop roles:

   - **java**(installs java)

```
---
# Java Support
- name: Needed java version.
  yum:
    name: java-{{java_version}}-openjdk
    state: present
  become: yes
  become_user: root
```

   - **java_test** (does only checks that java installed and running properly)

```
---
# tasks file for java_test
- name: Verify java exists
  shell: |
    javav=$(java -version 2>&1 | grep version | awk '{print $3}' |
sed 's/"//g' | cut -c-5)
    if [ "$javav" == "{{java_version}}" ]; then echo JAVA VERSION
IS OK; else echo fail; fi
  register: java_version_debug
- name: debug
  debug:
   msg: "{{java_version_debug.stdout}}"
```

   - **tomcat** (installs tomcat)

```
# Created group
- name: Creating group.
  group:
    name: "{{ tomcat_group }}"
    state: present
  become: yes
  become_user: root
# Created user
```

```
  - name: Creating user.
    user:
      name: "{{ tomcat_user }}"
      groups: "{{ tomcat_group }}"
      state: present
      append: yes
      createhome: yes
      home: "{{ tomcat_home }}"
    become: yes
    become_user: root
  # Getting tomcat
  - name: Get tomcat.
    unarchive:
      src: "http://archive.apache.org/dist/tomcat/tomcat-
  8/v{{tomcat_version}}/bin/apache-tomcat-
  {{tomcat_version}}.tar.gz"
      dest: "{{ tomcat_home }}"
      remote_src: True
    become: yes
    become_user: "{{ tomcat_user }}"
  # Symlink
  - name: Creating link to tomcat
    file:
      src: "{{tomcat_home}}apache-tomcat-{{tomcat_version}}"
      dest: "{{tomcat_home}}/current"
      state: link
    notify:
      - restart tomcat
    become: yes
    become_user: "{{ tomcat_user }}"
  # Tomcat init script
  - name: Init script
    become: yes
    become_user: root
    template:
      src: tomcat
      dest: /etc/init.d/tomcat
      mode: 0755
    notify:
      - restart tomcat
  # Adding service to startup
  - name: start tomcat
    service:
      name: tomcat
      state: started
    become: yes
    become_user: root
```

- **tomcat_test**(does only checks thattomcat installed and running properly)

```
---
# Check curl responcr
- name: Tomcat is runnning and responding
  uri:
   url: "http://localhost:8080/"
   status_code: 200
  register: result
  until: result.status == 200
  retries: 5
  delay: 1
# Check tomcat for uptime
- name: check uptime
  become: yes
  become_user: root
  shell: ls -al /proc | grep tomcat
```

- **nginx** (installs nginx)

```
---
# Checking nginx
- name: Ensuring nginx is exists
  yum:
    name: nginx
    state: installed
  become: yes
  become_user: root
# Config nginx
- name: Configure nginx
  become: yes
  become_user: root
  template:
    src: default.conf
    dest: /etc/nginx/conf.d/default.conf
  notify:
   - restart nginx
# starting nginx
- name: Starting nginx
  become: yes
  become_user: root
  service:
   name: nginx
   state: started
```

- **nginx_test**(does only checks that nginx installed and running properly)

```
---
# Nginx curl check
- name: Check nginx responce
  uri:
   url: "http://localhost:80/"
   status_code: 200
  register: result
  until: result.status == 200
  retries: 5
  delay: 1
```

4.2 Playbook should consist of **2** Plays:

- Installation
- Verification

```
---
# INSTALLATION PLAY
- name: Installation
  hosts: all
  vars_files:
  - variables.yml
  roles:
  - java
  - nginx
  - tomcat
#VERIFICATION PLAY
- name: Verification
  vars_files:
  - variables.yml
  hosts: all
  roles:
  - java_test
  - nginx_test
  - tomcat_test
```

4.3 Use **handlers** to manage tomcat/nginx configuration changes

4.4 Use module **debug** to check configuration during the installation

4.5 Define play/roles variables (at least):

- **tomcat_version**
- **tomcat_home**
- **tomcat_user**
- **tomcat_group**
- **java_version**

```
---
tomcat_version: 8.5.9
tomcat_home: /opt/tomcat/
tomcat_user: tomcat_as
tomcat_group: tomcat_as_group
java_version: 1.8.0
```

4.6 Every task/handler should have a name section with details of task purpose.

5. Software installation requirements:
    - Tomcat AS should be installed from sources (tar.gz) – download from the official site (http://archive.apache.org/dist/tomcat/).
    - Tomcat AS should be owned (and run) by user specified in variable (default: tomcat_as:tomcat_as_group).
    - Tomcat AS version should be 7.x, 8.x (at least 5 versions), exact version to be installed is taken from appropriate variable.
    - Tomcat installation folder (CATALINA_HOME) is /opt/tomcat/**$version**, where **$version** is the version of tomcat defined in playbook.
    - Java can be installed from CentOS Repositories
    - Use module **yum** to install Nginx
    - Use module **template** for management of nginx cofigs
    - Tomcat home page should be available on port 80 (accessible from Control Machile) via nginx.
6. Verification Procedure: playbook will be checked by instructor'sCI system as follows:
    6.1 Connect to student's host by ssh (username "student") with own ssh key.
    6.2 Go into the folder mentioned in point 1
    6.3 Destroy/Launch VM: vagrant destroy && vagrant up
    6.4 Execute VM provisioning: ansible-playbook site.yml -i inventory -vv
    6.5 If previous steps are done successfully, instructor will check report (pdf-file)
7. Feedback: report issues/problems you had during the development of playbook and time spent for development.