Lab Work Task. Tomcat AS Provisioning

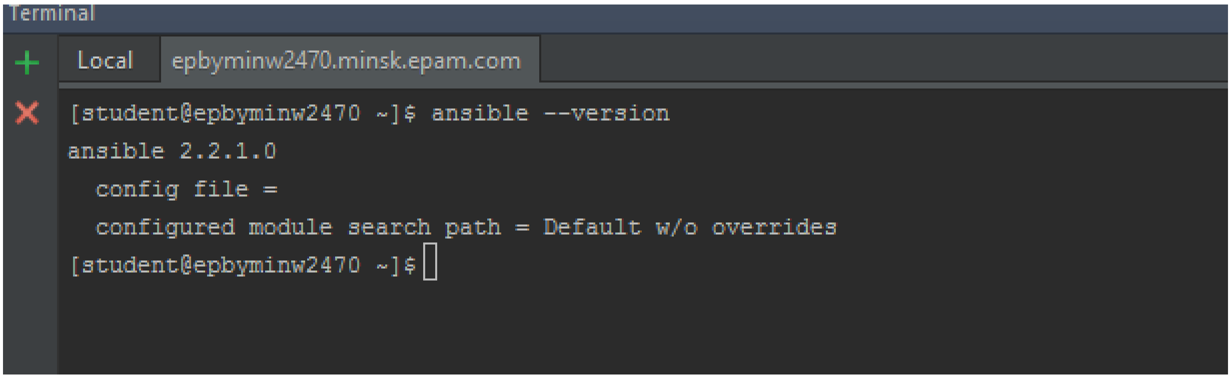
Review

Using Ansible v2.2.1 for provisioning tomcat application stack. Learning by doing.

Task

On Host Node (Control Machine):

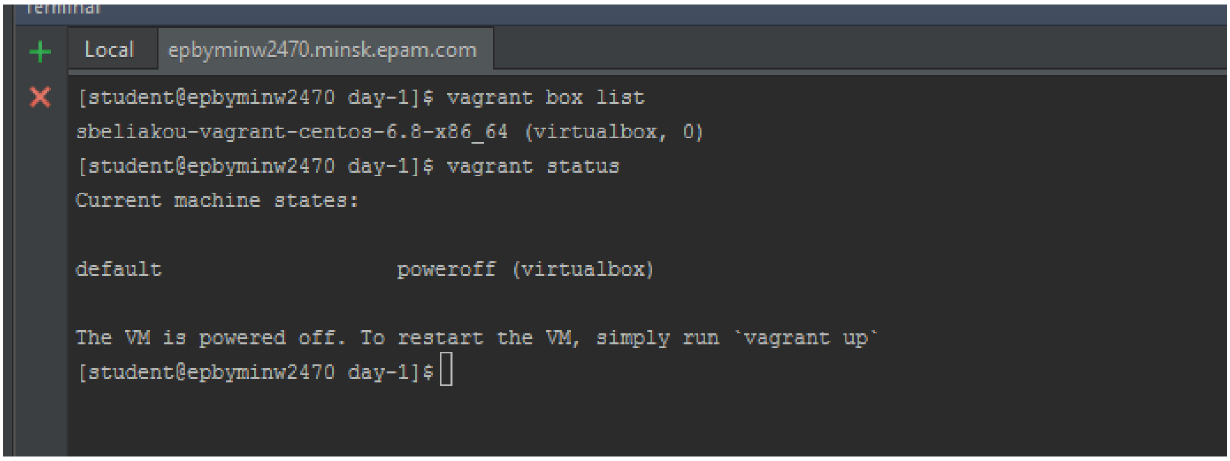
1. Install Ansible v2.2.1 with python pip. Report details where ansible has been installed.



```
[student@epbyminw2470 ~]$ ansible --version
ansible 2.2.1.0
  config file =
  configured module search path = Default w/o overrides
[student@epbyminw2470 ~]$ 
```

Ansible version.

2. Create folder ~/cm/ansible/day-1. All working files are supposed to be placed right there.
3. Spin up clear CentOS6 VM using vagrant (“vagrant init sbeliakou/centos-6.8-x86_64”). Verify connectivity to the host using ssh keys (user: vagrant)

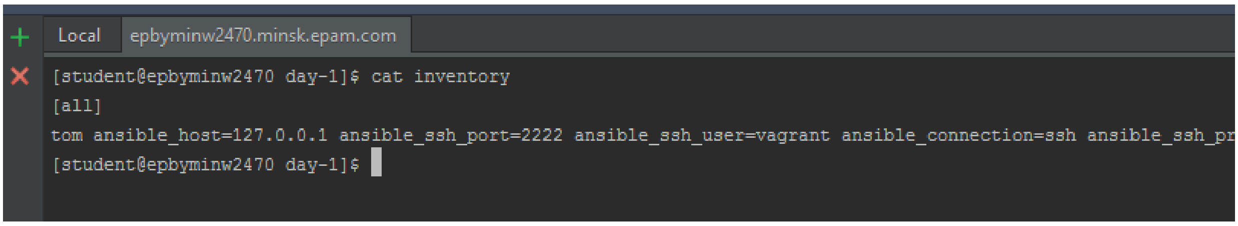


```
[student@epbyminw2470 day-1]$ vagrant box list
sbeliakou-vagrant-centos-6.8-x86_64 (virtualbox, 0)
[student@epbyminw2470 day-1]$ vagrant status
Current machine states:

default                   poweroff (virtualbox)

The VM is powered off. To restart the VM, simply run `vagrant up`
[student@epbyminw2470 day-1]$ 
```

4. Create ansible inventory file (name: **inventory**) with remote host connection details:
   - Remote VM hostname/ip/port
   - Remote ssh log in username
   - Connection type



```
[student@epbyminw2470 day-1]$ cat inventory
[all]
tom ansible_host=127.0.0.1 ansible_ssh_port=2222 ansible_ssh_user=vagrant ansible_connection=ssh ansible_ssh_pr
[student@epbyminw2470 day-1]$ 
```

Created inventory file.

5. Test ansible connectivity to the VM with ad-hoc command:

   $ ansible VM-name -i inventory -m setup

   Find out host details:

   - Number of CPUs
   - Host name
   - Host IP(s)
   - Total RAM

6. Develop a playbook (name: **tomcat_provision.yml**) which is supposed to run against any host (specified in inventory)

   Use following modules (at least):

   - **copy**
   - **file**
   - **get_url**
   - **group**
   - **service**
   - **shell**
   - **unarchive**
   - **user**
   - **yum**

   Define play variables (at least):

   - **tomcat_version**
   - **java_version**

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

   Examples:

   - name: Ensure user student exists
   - name: Fetch artifact form the Shared repository

   Ensure tomcat is up and running properly with module "shell" (at least 3 different checks).

   Second (and other) run(s) of playbook shouldn't interrupt the service – one of checks should show tomcat uptime.

tomcat_provision.yml listing

```
- hosts: all
  vars:
    tomcat_version: 8.5.9
    java_version: 1.8.0
  tasks:
# CREATING GROUP
  - name: Creating group tomcat_as.
    group:
      name: tomcat_as
      state: present
    become: yes
    become_user: root
# CREATING USER
  - name: Creating user tomcat_as.
```

```
        user:
          name: tomcat_as
          groups: tomcat_as
          state: present
          append: yes
        become: yes
        become_user: root
    # CREATING TOMCAT DIRECTORY
      - name: Creating tomcat directory.
        file:
          path: /opt/tomcat
          state: directory
          owner: vagrant
          group: vagrant
        become: yes
        become_user: root
    # DOWNLOADING TOMCAT WITH PREDEF VERSION
      - name: Downloading tomcat server.
        get_url:
          url: "http://archive.apache.org/dist/tomcat/tomcat-8/v{{tomcat_version}}/bin/apache-tomcat-{{tomcat_version}}.tar.gz"
          dest: /opt/tomcat/
    # UNPACKING ARCHIVE
      - name: Unpacking tomcat archive.
        unarchive:
          src: /opt/tomcat/apache-tomcat-{{tomcat_version}}.tar.gz
          creates: /opt/tomcat/apache-tomcat-{{tomcat_version}}
          dest: /opt/tomcat/
          remote_src: yes
    # MV DIRECTORY
      - name: Moving directory.
        command: mv /opt/tomcat/apache-tomcat-{{tomcat_version}} /opt/tomcat/{{tomcat_version}} creates=/opt/tomcat/{{tomcat_version}}
        become: yes
        become_user: root
    # Removing tomcat archive.
      - name: Removing archive file.
        file:
          path: /opt/tomcat/apache-tomcat-{{tomcat_version}}.tar.gz
          state: absent
        become: yes
        become_user: root
    # CHANGING PERMISSIONS
      - name: Changing directory permissions.
        file:
          path: /opt/tomcat/
          recurse: yes
          owner: tomcat_as
          group: tomcat_as
        become: yes
        become_user: root
    # INSTALLING JAVA WITH YUM
      - name: Installing java!
        yum:
          name: java-{{java_version}}-openjdk
          state: present
        become: yes
        become_user: root
    # COPYING STARTUP SCRIPT
      - name: Moving startup script to vm.
        copy:
          src: /home/student/cm/ansible/day-1/tomcat.sh
          dest: /etc/init.d
        become: yes
        become_user: root
    # CHANGE SCRIPT CHMODE
      - name: Change script chomd.
        file:
          path: /etc/init.d/tomcat.sh
          mode: 0755
        become: yes
        become_user: root
    # Starting service
      - name: Adding service to startup
        service:
          name: tomcat.sh
          enabled: yes
          state: started
        become: yes
        become_user: root
    #Running tests
    #Check uptime
      - name: Check uptime
        shell: "ps -eo pid,comm,lstart,args | grep tomcat"
    #Check avaliability
      - name: Check response
        shell: "sleep 15 && curl -iL http://localhost:8080 | grep HTTP"
    #Check service status
      - name: Check status
shell: /etc/init.d/tomcat.sh status
```

7. 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 tomcat_as:tomcat_as_group
   - Tomcat AS version should be 8.x
   - 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

8. Verification Procedure: playbook will be checked by instructor'sCI system as follows:

   8.1 Connect to student's host by ssh (username "student") with own ssh key.

   8.2 Check the version of ansible installed on the system (as mentioned in point 1)

   8.3 Go into the folder mentioned in point 2

   8.4 Destroy/Launch VM: vagrant destroy && vagrant up

   8.5 Execute VM provisioning: ansible-playbook tomcat_provision.yml -i inventory -vv

   8.6 If previous steps are done successfully, instructor will check the report

9. Feedback: report issues/problems you had during the development of playbook and time spent for development.