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):

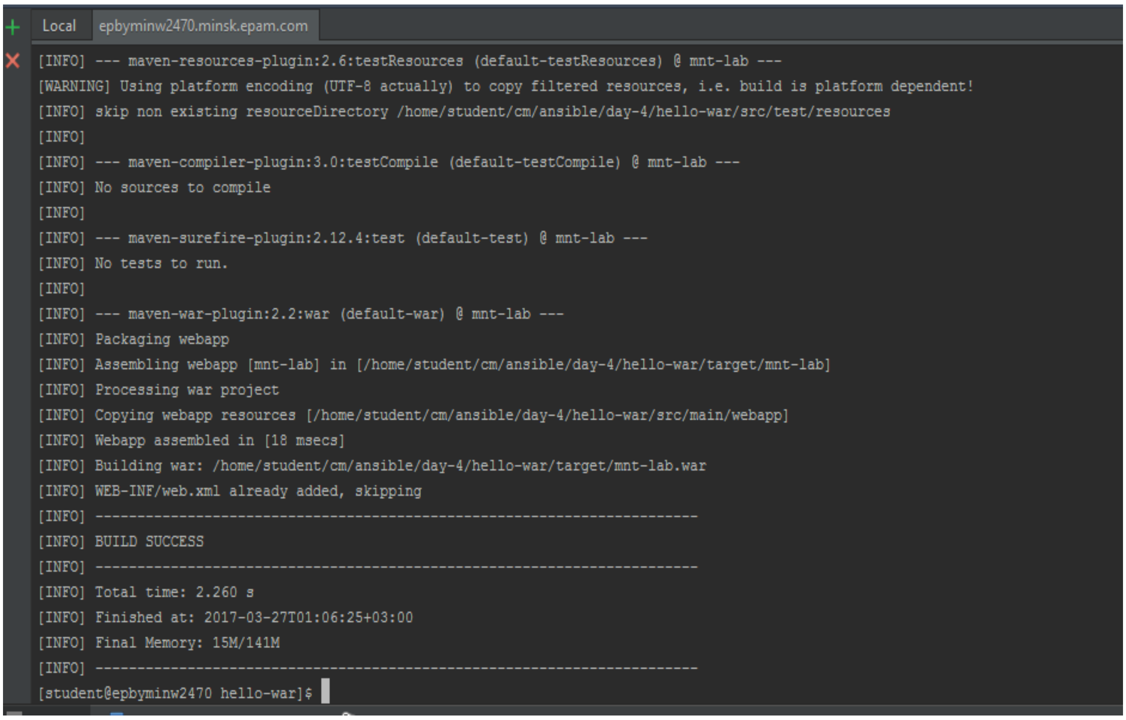
1. Create folder ~/cm/ansible/day-4. All working files are supposed to be placed right there.
2. Configure Ansible to use roles from ~/cm/ansible/day-3 folder

```
[defaults]
roles_path = /home/student/cm/ansible/day-3/roles
host_key_checking = False

callback_plugins = callback_plugins

stdout_callback = human
callback_whitelist = human
```

Add ansible.cfg

3. Build java sources with maven form here:https://github.com/sbeliakou/hello-war (MANUALLY)
   Requirements:
   Update file src/main/resources/build-info.txt with following details:
   - Build time
   - Build Machine Name
   - Build User Name

   Build command:
   $ mvnclean package -DbuildNumber=$VERSION



```
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ mnt-lab ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /home/student/cm/ansible/day-4/hello-war/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.0:testCompile (default-testCompile) @ mnt-lab ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ mnt-lab ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-war-plugin:2.2:war (default-war) @ mnt-lab ---
[INFO] Packaging webapp
[INFO] Assembling webapp [mnt-lab] in [/home/student/cm/ansible/day-4/hello-war/target/mnt-lab]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/student/cm/ansible/day-4/hello-war/src/main/webapp]
[INFO] Webapp assembled in [18 msecs]
[INFO] Building war: /home/student/cm/ansible/day-4/hello-war/target/mnt-lab.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.260 s
[INFO] Finished at: 2017-03-27T01:06:25+03:00
[INFO] Final Memory: 15M/141M
[INFO] ------------------------------------------------------------------------
[student@epbyminw2470 hello-war]$
```

Develop a module for deploying resulted war file to Tomcat AS.
Example:
- deploy: url=... war=… username=… password=…

```
#!/bin/bash

source $1

if [ -z "$war" ]; then
    printf '{"failed": true, "msg": "missing argument: war"}'
    printf "\n"
    exit 1
fi
if [ -z "$url" ]; then
    printf '{"failed": true, "msg": "missing argument: url"}'
    printf "\n"
    exit 1
fi

if [ -z $pwd ]; then
   printf '{"failed": true, "msg": "missing argument: password"}'
   printf "\n"
   exit 1
fi

if [ -z $usr ]; then
   printf '{"failed": true, "msg": "missing argument: username"}'
   printf "\n"
   exit 1
fi

if [ -f $war ]; then
   printf '{"failed": true, "msg": "missing war file."}'
   printf "\n"s
   exit 1
fi

warfile=`basename $war`
filepath=${warfile%.*}

output= $(curl --silent -T "$war" -u $usr:$pwd
"$url/manager/text/deploy?path=/$filepath&update=true" 2>&1 | python
-c "import json,sys; print json.dumps(sys.stdin.read())")

if [ $? -ne 0 ];then
    printf '{"failed": true, "msg" : "deployment error", "output":
%s}' "$output"
    exit 1
else
    printf '{"changed": true, "failed": false, "msg" : "Well done"}'
    exit 0
fi
```

Requirements:
Afterdeployment you should update file /var/lib/tomcat/webapps/deploy-info.txt
  Add following details:
  - Deployment time
  - Deploy User

4. Develop a playbook (name: **deploy.yml**) to deploy war file (role!), pass war-file as an extra-argument to playbook.
   Example:
   $ ansible-playbook deploy.yml … -e war=…

   Consider: Playbook should deploy war file and test that deployment was successful.

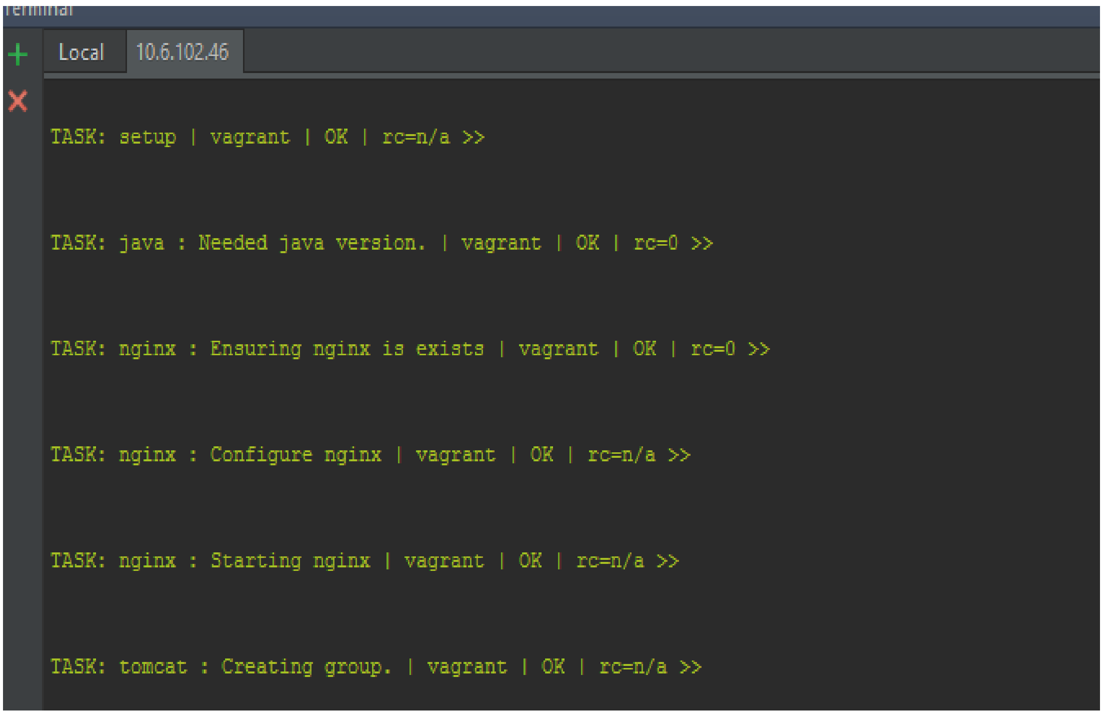
```
# Creating directory
- name: Created Dir
  shell: 'mkdir -p /var/lib/tomcat/webapps'
- name: get current time
  shell: date
  register: currentdate
# Adding to file
- name: Add data to file
  lineinfile:
    dest: /var/lib/tomcat/webapps/deploy-info.txt
    regexp: '^Deployment\stime.+'
    line: 'Deployment time: {{currentdate.stdout}}'
    create: yes
# Adding to file
- name: Add data to file
  lineinfile:
    dest: /var/lib/tomcat/webapps/deploy-info.txt
    regexp: '^Deploy\suser.+'
    line: 'Deploy user: {{user}}'
# Testing
- name: Test deployment
  uri: url={{output.appurl}} return_content=yes
  register: app_webpage
  delegate_to: 127.0.0.1
  no_log: yes
- name: result of app test
  debug: msg="Well done bro!"
  when: "'{{currentdate.stdout}}' in app_webpage.content"
```

5. Develop custom callback plugin to make playbook output more human readable.
   Example: Appendix A.

```
from __future__ import (absolute_import, division, print_function)
__metaclass__ = type

from ansible.plugins.callback import CallbackBase


class CallbackModule(CallbackBase):

    '''
    This is the default callback interface, which simply prints
messages
    to stdout when new callback events are received.
    '''

    CALLBACK_VERSION = 2.0
    CALLBACK_TYPE = 'stdout'
    CALLBACK_NAME = 'deploy'

    def show(self, task, host, result, caption):
        buf = "{0} | {1} | {2} | rc={3} >>\n".format(task, host,
caption,
result.get('rc', 'n/a'))
        buf += result.get('stdout', '')
        buf += result.get('stderr', '')
        buf += result.get('msg', '')
        print(buf + "\n")

    def v2_runner_on_failed(self, result, ignore_errors=False):
        self.show(result._task, result._host.get_name(),
result._result, "FAILED")

    def v2_runner_on_ok(self, result):
        self.show(result._task, result._host.get_name(),
result._result, "OK")

    def v2_runner_on_skipped(self, result):
        self.show(result._task, result._host.get_name(),
result._result, "SKIPPED")

    def v2_runner_on_unreachable(self, result):
        self.show(result._task, result._host.get_name(),
result._result, "UNREACHABLE")
```

```
[defaults]
roles_path = /home/student/cm/ansible/day-3/roles
host_key_checking = False

callback_plugins = callback_plugins

stdout_callback = human
callback_whitelist = human
```

Config ansible



```
TASK: setup | vagrant | OK | rc=n/a >>


TASK: java : Needed java version. | vagrant | OK | rc=0 >>


TASK: nginx : Ensuring nginx is exists | vagrant | OK | rc=0 >>


TASK: nginx : Configure nginx | vagrant | OK | rc=n/a >>


TASK: nginx : Starting nginx | vagrant | OK | rc=n/a >>


TASK: tomcat : Creating group. | vagrant | OK | rc=n/a >>
```

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 deploy.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.