EPAM Systems, RD Dep., RD Dep.

MTN.*NIX Automated Environment Configuration Management

Ansible. 2

REVISION HISTORY					
Ver.	**Description of Change**	**Author**	**Date**	**Approved**	
				Name	**Effective Date**
<1.0>	Initial revision	Siarhei Beliakou	17-Mar-2017		

Lab Work Task. Web Server Provisioning

Review

Using Ansible v2.3.1 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

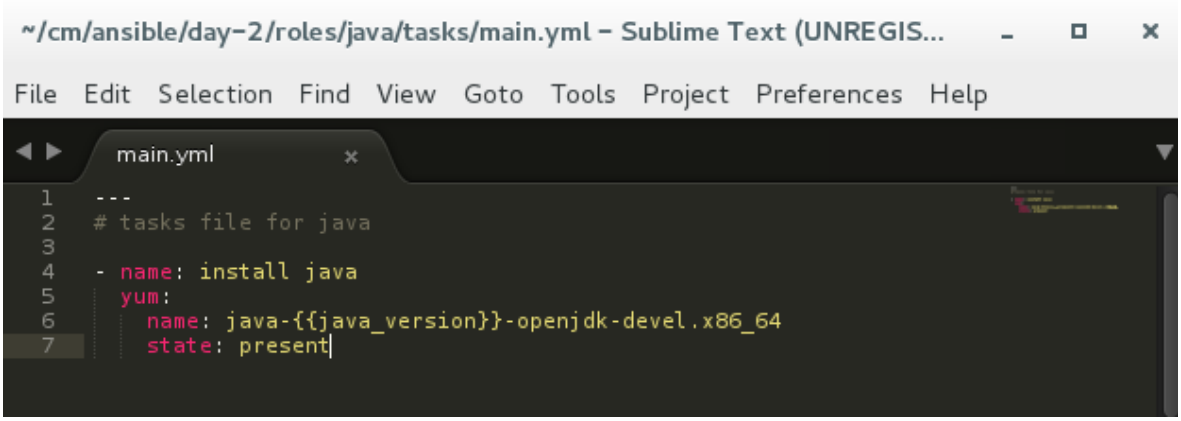
4. Develop a playbook (name: **site.yml**) which is supposed to run against any host (specified in inventory)

site.yml | main.yml — tomcat_test/tasks | main.yml

```
- name: Installation
  hosts: tomcat
  become: yes
  become_method: sudo

  pre_tasks:
   - debug: msg="Let's install Nginx, Tomcat"

  roles:
   - tomcat
   - nginx



- name: Verification local
  hosts: tomcat
  become: yes
  become_method: sudo

  pre_tasks:
   - debug: msg="Let's test Java, Tomcat, Nginx"

  roles:
   - java_test
   - tomcat_test
   - nginx_test

- name: Verification remote
  hosts: localhost

  pre_tasks:
   - debug: msg="Let's test Java, Tomcat, Nginx from localhost"

  tasks:

    - name: Check response from Tomcat service
      uri:
         url: http://192.168.56.10:80
         return_content: yes
      register: webpage

    - name: Fail if Tomcat is not in the page content
      assert:
       that:
         - "'Tomcat' in webpage.content"
       msg: 'Fail Tomcat is not in the page content'
```

4.1 Develop roles:

- **java** (installs java)



Open main.yml ~/cm/ansible/day-2/roles/java/defaults

```
---
# defaults file for java

java_version: 1.8.0
```

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

Open main.yml ~/cm/ansible/day-2/roles/java_test/tasks Save

```
---
# tasks file for java_test

- name: check Java version
  shell: java -version 2>&1 | grep 'version' | awk '{print $3}'
  register: out

- debug: var=out.stdout

- name: check if java has port 8080 open
  shell: ss -tp state listening sport = :8080 | grep java
  become: yes
  register: java_out

- debug: var=java_out.stdout
```

- **tomcat** (installs tomcat)

Open main.yml ~/cm/ansible/day-2/roles/tomcat/tasks Save

```
- name: be sure group tomcat exist
  group:
     name: '{{tomcat_group}}'
     state: present

- name: be sure user tomcat exist
  user:
     name: '{{tomcat_user}}'
     group: '{{tomcat_group}}'
     state: present

- name: be sure tomcat_home directory exist
  file:
      path: '{{tomcat_home}}'
      state: directory
      recurse: yes

- name: be sure tomcat downloaded
  get_url:
     url: '{{tomcat_url_download}}'
     dest: /home/vagrant/apache-tomcat-{{tomcat_version}}.
{{tomcat_subversion}}.tar.gz
     force: no

- name: be sure tomcat archive exist
  unarchive:
     src: /home/vagrant/apache-tomcat-{{tomcat_version}}.
{{tomcat_subversion}}.tar.gz
     dest: /home/vagrant/
     remote_src: yes
     creates: /home/vagrant/apache-tomcat-{{tomcat_version}}.
{{tomcat_subversion}}
  notify:
     - copy file

- name: force  notify copy file
  meta: flush_handlers


- name: be sure tomcat.service not changed
  template:
      src: tomcat.service.j2d
      dest: /etc/systemd/system/tomcat.service
  notify:
     - systemctl daemon
     - restart tomcat

- service:
      name: tomcat
      enabled: yes
      state: started
```

Open main.yml ~/cm/ansible/day-2/roles/tomcat/handlers Save

```
- name: systemctl daemon
  shell: systemctl daemon-reload

- name: restart tomcat
  service:
     name: tomcat
     state: restarted

- name : copy file
  shell: cp -R /home/vagrant/apache-tomcat-{{tomcat_version}}.
{{tomcat_subversion}}/* '{{tomcat_home}}' && chown -R
'{{tomcat_user}}':'{{tomcat_group}}' /opt/tomcat/
```

main.yml
~/cm/ansible/day-2/roles/tomcat/defaults

```
tomcat_version: 8
tomcat_subversion: 0.45
tomcat_user: tomcat_as
tomcat_group: tomcat_as_group
```

main.yml
~/cm/ansible/day-2/roles/tomcat/meta

main.yml

```
dependencies:
- {role: java}
```

tomcat.service.j2d
~/cm/ansible/day-2/roles/tomcat/templates

```
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target

[Service]
Type=forking

Environment=CATALINA_PID={{tomcat_home}}/temp/tomcat.pid
Environment=CATALINA_HOME={{tomcat_home}}
Environment=CATALINA_BASE={{tomcat_home}}

ExecStart={{tomcat_home}}/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID

User={{tomcat_user}}
Group={{tomcat_group}}
UMask=0007

[Install]
WantedBy=multi-user.target
```

main.yml
~/cm/ansible/day-2/roles/tomcat/vars

```
tomcat_url_download: http://archive.apache.org/dist/tomcat/tomcat-
{{tomcat_version}}/v{{tomcat_version}}.{{tomcat_subversion}}/bin/apache-
tomcat-{{tomcat_version}}.{{tomcat_subversion}}.tar.gz
tomcat_home: /opt/tomcat/{{tomcat_version}}.{{tomcat_subversion}}
```

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

Open main.yml ~/cm/ansible/day-2/roles/tomcat_test/tasks Save

```
---
# tasks file for tomcat_test
- name: Check response status
  shell: curl -Is http://127.0.0.1:8080 | head -n 1 | awk '{print $2}'
  register: response

- debug: var=response.stdout

- name: stat /opt/tomcat
  stat: path=/opt/tomcat
  register: st

- debug: msg="Path /opt/tomcat exists and is a directory"
  when: st.stat.isdir is defined and st.stat.isdir

- name: ownership /opt/tomcat
  stat: path=/opt/tomcat
  register: owner

- debug: msg="Ownership /opt/tomcat correct user=tomcat_as,
group=tomcat_as_group"
  when: owner.stat.pw_name == 'tomcat_as' and owner.stat.gr_name ==
'tomcat_as_group'

- name: Check if tomcat is running
  shell: ps aux | grep tomcat | grep -v grep
  ignore_errors: yes
  changed_when: false
  register: service_tomcat_status

- debug: var=service_tomcat_status.stdout_lines
```

- **nginx** (installs nginx)

Open main.yml ~/cm/ansible/day-2/roles/nginx/tasks

```
---
# tasks file for nginx

- name: be sure Nginx is installed
  yum:
    name: nginx
    state: present

- name: be sure nginx.conf not changed
  template:
      src: nginx.conf.j2d
      dest: /etc/nginx/nginx.conf
  notify:
    - restart nginx

- name: be sure Nginx service started
  service:
    name: nginx
    enabled: yes
    state: started
```

Open main.yml ~/cm/ansible/day-2/roles/nginx/handlers

```
---
# handlers file for nginx

- name: restart nginx
  service:
     name: nginx
     state: restarted
```

Open nginx.conf.j2d ~/cm/ansible/day-2/roles/nginx/templates Save

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;


include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local]
"$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;


    include /etc/nginx/conf.d/*.conf;

    server {
        listen       80 default_server;
        listen       [::]:80 default_server;
        server_name  _;
        root         /usr/share/nginx/html;

     location / {
     proxy_pass http://127.0.0.1:8080;
        }

        error_page 404 /404.html;
            location = /40x.html {
        }
```

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

Open main.yml ~/cm/ansible/day-2/roles/nginx_test/tasks

```
---
# tasks file for nginx_test

- name: Check if Nginx is running
  shell: ps aux | grep nginx | grep -v grep
  ignore_errors: yes
  changed_when: false
  register: service_nginx_status

- name: Report status of NGINX
  fail:
    msg: |
      Service nginx is not running.
      Return code from `grep`:
      {{ service_nginx_status.rc }}
  when: service_nginx_status.rc != 0

- name: Check response status
  shell: curl -Is http://127.0.0.1:80
  register: response

- debug: var=response.stdout
```

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

- Installation
- Verification

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**

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's CI 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.