EPAM Systems, RD Dep., RD Dep.

MTN.*NIX.11 Automated Environment Configuration Management

Ansible. 4

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

Legal Notice This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM Systems.

Lab Work Task. Web Server Provisioning

Review

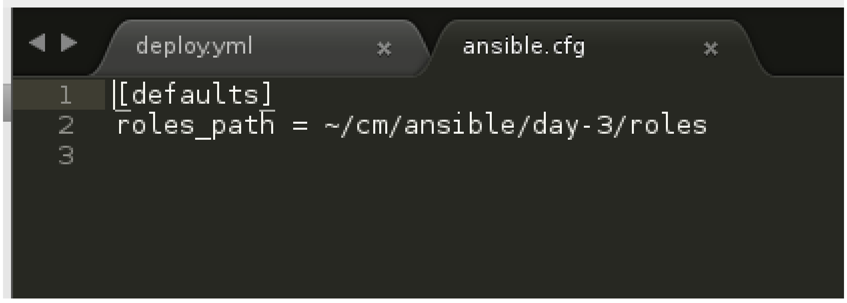
Using Ansible v2.2.1 for provisioning nginx + tomcat application stack.

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



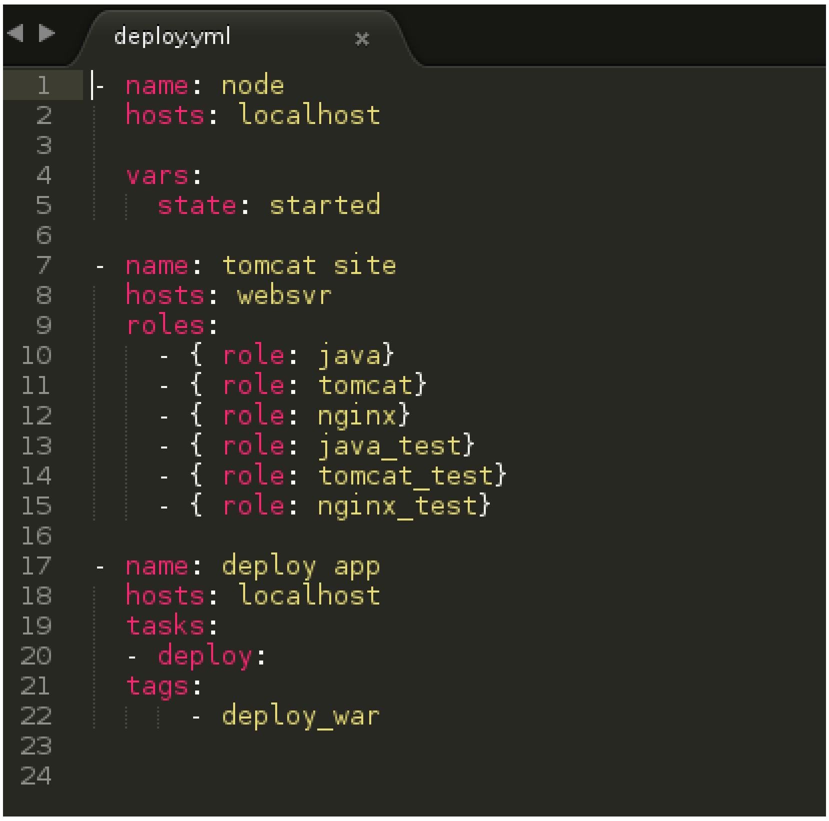
```
[defaults]
roles_path = ~/cm/ansible/day-3/roles
```

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:
   $ mvn clean package -DbuildNumber=$VERSION
4. Develop a module for deploying resulted war file to Tomcat AS.
   Example:
   - deploy: url=... war=… username=… password=…

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

5. 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.



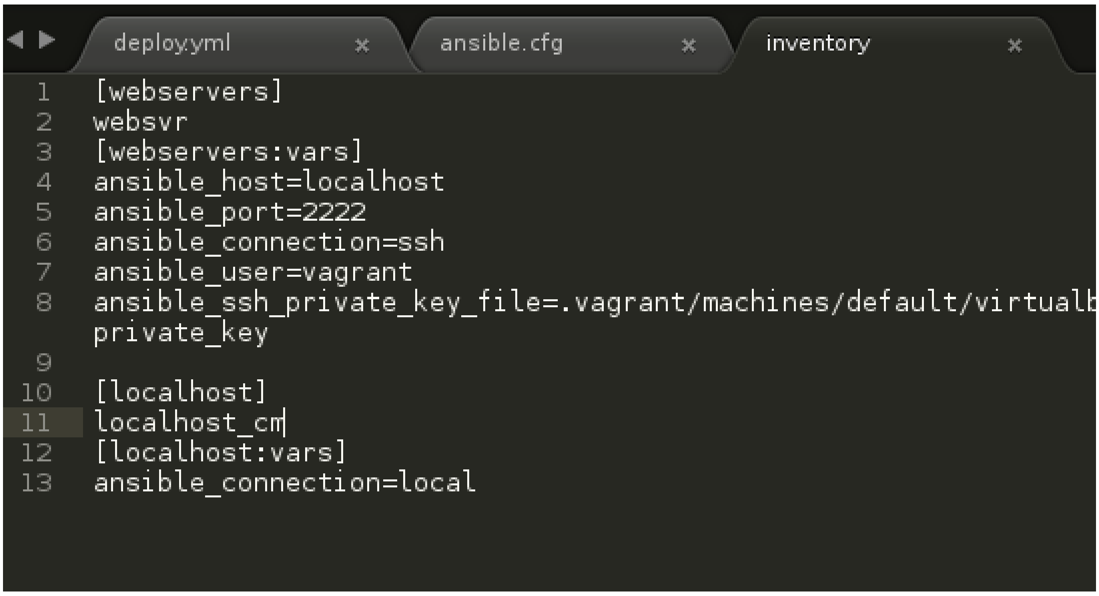
```
- name: node
  hosts: localhost

  vars:
    state: started

- name: tomcat site
  hosts: websvr
  roles:
    - { role: java}
    - { role: tomcat}
    - { role: nginx}
    - { role: java_test}
    - { role: tomcat_test}
    - { role: nginx_test}

- name: deploy app
  hosts: localhost
  tasks:
  - deploy:
  tags:
      - deploy_war


```



```
[webservers]
websvr
[webservers:vars]
ansible_host=localhost
ansible_port=2222
ansible_connection=ssh
ansible_user=vagrant
ansible_ssh_private_key_file=.vagrant/machines/default/virtualb
private_key

[localhost]
localhost_cm
[localhost:vars]
ansible_connection=local
```

6. Develop custom callback plugin to make playbook output more human readable.
   Example: Appendix A.
7. Verification Procedure: playbook will be checked by instructor’s CI system as follows:
   7.1 Connect to student’s host by ssh (username “student”) with own ssh key.
   7.2 Go into the folder mentioned in point 1
   7.3 Destroy/Launch VM: vagrant destroy && vagrant up
   7.4 Execute VM provisioning: ansible-playbook deploy.yml -i inventory -vv
   7.5 If previous steps are done successfully, instructor will check report (pdf-file)
8. Feedback: report issues/problems you had during the development of playbook and time spent for development.

APPENDIX A

Playbook:

```
$ cat demo.yml
- hosts: localhost
  connection: local

  tasks:
    - name: Greetings
      shell: echo Hello

    - name: Skip
      debug: msg="Hello again"
      when: False

    - name: Fail and continue
      fail: msg="What can go wrong?"
      ignore_errors: True

    - name: Fail
      fail: msg="OH SHI--"
```

Run:

```
1 plays in demo.yml
TASK: setup | localhost | OK | rc=n/a >>

TASK: Greetings | localhost | OK | rc=0 >>
Hello

TASK: Skip | localhost | SKIPPED | rc=n/a >>
```

```
TASK: Fail and continue | localhost | FAILED | rc=n/a >>
What can go wrong?

TASK: Fail | localhost | FAILED | rc=n/a >>
OH SHI--
```

© EPAM Systems, 2017