EPAM Systems, RD Dep., RD Dep.

MTN.*NIX.11 Automated Environment Configuration Management

Ansible. 3

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

Developing custom modules and filters. Learning by doing.

Task

On Host Node (Control Machine):

1. Create folder ~/cm/ansible/day-3. All working files are supposed to be placed right there.
2. Develop custom filter to select an url to download mongodb depends on OS name and S/W version from https://www.mongodb.org/dl/linux/
   Requirements:
   - Write a playbook (name: **mongodb.yml**) to prove that this module works
   - At least 9 versions of MongoDB for 3 different Linux distributives (list with links)
   - Filter should process a list of urls and takes 3 options: os_family (discovered by ansible, variable, produced by setup module), os release number and mongodb_version (set in play vars)
   See example in Appendix A

mongodb.yml

```yaml
- hosts: localhost
  connection: local

  vars:
    mongodb_version: '3'
    mongo_src:
    - mongodb-linux-x86_64-rhel62-3.4.1
    - mongodb-linux-x86_64-rhel70-3.4.1
    - mongodb-linux-x86_64-rhel55-3.2.11
    - mongodb-linux-x86_64-rhel64-3.2.11
    - mongodb-linux-x86_64-rhel70-3.2.17
    - mongodb-linux-x86_64-rhel55-3.0.14
    - mongodb-linux-x86_64-rhel64-3.0.14
    - mongodb-linux-x86_64-rhel70-3.0.14

  tasks:
  - debug: msg={{ mongo_src | get_mongo_src(ansible_os_family, ansible_distribution_major_version, mongodb_version) }}

```

mongodb.yml × get_mongodb.py ×

```python
from __future__ import (absolute_import, division, print_function)
__metaclass__ = type
from ansible import errors
import re

dict={'RedHat':'rhel','Debian':'debian','Ubuntu':'ubuntu'}

class FilterModule(object):
    def filters(self):
        return {
            'get_mongo_src': get_mongo_src,
        }

def get_mongo_src(mongo_src, os_family, major_version, mongodb_version):
    result = []
    for i in range(len(mongo_src)):
        if ((dict[os_family] + major_version) in mongo_src[i]) and (str(mongodb_version) in mongo_src[i]):
            result.append(mongo_src[i])
    return "Recomended MongoDB versions for {} distro are: {}".format(os_family, result)

```

3. Develop custom module to manage VirtualBox:
   Arguments:
   - path to vagrantfile
   - state: started, stopped, destroyed
   Return values:
   - state: running, stopped, not created
   - ip address, port
   - path to ssh key file
   - username to connect to VM
   - os_name
   - RAM size
   Errors:
   - file doesn’t exists
   - failed on creation
   - etc

prov

```
#!/bin/sh
#$1 - First parameter - path to vagrantfile
#$2 - Second parameter - state: started, stopped, destroyed

### check input vars
source $1

if [ -z "$path" ]; then
    printf '{"failed": true, "msg": "Missing required arguments: path"}'
    exit 1
fi
if [ -z "$state" ]; then
    printf '{"failed": true, "msg": "Missing required arguments: state"}'
    exit 1
fi

### check if Vagrantfile exists
if [ ! -f "$path" ]; then
    printf '{"failed": true, "msg": "Missing Vagrantfile"}'
    exit 1
fi
###


function get_vars
    {
    ip=$(vagrant ssh-config | grep HostName | awk '{print $2}')
    port=$(vagrant ssh-config | grep Port | awk '{print $2}')
    user=$(vagrant ssh-config | grep -w "User" | awk '{print $2}' 2>/dev/null)
    key=$(vagrant ssh-config | grep IdentityFile | awk '{print $2}' 2>/dev/null)
    status=$(vagrant status | grep default | awk '{print $2}')
    os_name=$(vagrant ssh -c "cat /etc/redhat-release" 2>/dev/null)
    ram=$(vagrant ssh -c "cat /proc/meminfo | grep MemTotal | awk '{print \$2}'" 2>/dev/null)
}


```

```
function vagrant_up
    {
    status=$(vagrant status | grep default | awk '{print $2}')
    if [ "$status" == "running" ]; then
        get_vars
        printf '{"failed": false, "changed": false, "ip": "%s", "port": "%s", "user": "%s", "key": "%s", "status": "%s", "os_name": "%s",
            "ram": "%s"}' "$ip" "$port" "$user" "$key" "$status" "$os_name" "$ram"
        exit 0
    else
        vagrant up &>/dev/null
        get_vars
        printf '{"failed": false, "changed": true, "ip": "%s", "port": "%s", "user": "%s", "key": "%s", "status": "%s", "os_name": "%s",
            "ram": "%s"}' "$ip" "$port" "$user" "$key" "$status" "$os_name" "$ram"
    fi
}


function vagrant_halt
{   status=$(vagrant status | grep default | awk '{print $2}')
    if [ "$status" == "running" ]; then
        vagrant halt
        status=$(vagrant status | grep default | awk '{print $2}')
        printf '{"failed": false, "changed": true, "status": "%s"}' "$status"
        exit 0
    else
        changed="false"
        failed="false"
        printf '{"failed": false, "changed": true, "status": "%s"}' "$status"
    fi
}

```

```
66
67  function vagrant_destroy
68  {   status=$(vagrant status | grep default | head -n1 | awk '{print $2}')
69      if [ "$status" == "running" ] || [ "$status" == "poweroff" ]; then
70          vagrant destroy --force
71          status=$(vagrant status | grep default | head -n1 | awk '{print $2 " " $3}')
72          printf '{"failed": false, "changed": true, "status": "%s"}' "$status"
73          exit 0
74      else
75          status=$(vagrant status | grep default |  head -n1 | awk '{print $2 " " $3}')
76          changed="false"
77          failed="false"
78          printf '{"failed": false, "changed": true, "status": "%s"}' "$status"
79      fi
80  }
81
82  case $state in
83      started)
84          vagrant_up
85      ;;
86      stopped)
87          vagrant_halt
88      ;;
89      destroyed)
90          vagrant_destroy
91      ;;
92      *)
93          printf '{"failed": true, "msg": "invalid state selected {started | stopped | destroyed}"}'
94          exit 1
95      ;;
96  esac
97
98  exit 0
99
```

4. Create a playbook (name: **stack.yml**) to provision Tomcat stack (nginx + tomcat) on VirtualBox VM
   Requirements:
   - 2 Plays: provision VM, roll out Tomcat stack (using roles from previous lab work)
   - 2nd play should work with dynamically composed Inventory (connection settings to VM), http://docs.ansible.com/ansible/add_host_module.html

stack.yml

```yaml
- name: stack
  hosts: localhost

  vars:
    state: started

  tasks:
  - name: vagrant provision
    prov: path=Vagrantfile state={{state}}
    register: vagrant_vars
  - debug: msg={{vagrant_vars}}
  - add_host:
      name: websvr
      ansible_host: "{{vagrant_vars.ip}}"
      ansible_port: "{{vagrant_vars.port}}"
      ansible_connection: ssh
      ansible_user: "{{vagrant_vars.user}}"
      ansible_ssh_private_key_file: "{{vagrant_vars.key}}"
    when: vagrant_vars.status == "running"

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

```

```
[student@epbyminw2473 day-3]$ ansible-playbook stack.yml -i localhost, -c local -vv
No config file found; using defaults

PLAYBOOK: stack.yml ************************************************************
2 plays in stack.yml

PLAY [stack] *******************************************************************

TASK [setup] *******************************************************************
ok: [localhost]

TASK [vagrant provision] *******************************************************
task path: /home/student/cm/ansible/day-3/stack.yml:8
changed: [localhost] => {"changed": true, "failed": false, "ip": "127.0.0.1", "key": "/home/student/cm/ansib
le/day-3/.vagrant/machines/default/virtualbox/private_key", "os_name": "CentOS release 6.8 (Final)", "port":
 "2222", "ram": "630788", "status": "running", "user": "vagrant"}

TASK [debug] *******************************************************************
task path: /home/student/cm/ansible/day-3/stack.yml:11
ok: [localhost] => {
    "msg": {
        "changed": true,
        "failed": false,
        "ip": "127.0.0.1",
        "key": "/home/student/cm/ansible/day-3/.vagrant/machines/default/virtualbox/private_key",
        "os_name": "CentOS release 6.8 (Final)",
        "port": "2222",
        "ram": "630788",
        "status": "running",
        "user": "vagrant"
    }
}

TASK [add_host] ****************************************************************
task path: /home/student/cm/ansible/day-3/stack.yml:12
creating host via 'add_host': hostname=websvr
changed: [localhost] => {"add_host": {"groups": [], "host_name": "websvr", "host_vars": {"ansible_connection
": "ssh", "ansible_host": "127.0.0.1", "ansible_port": "2222", "ansible_ssh_private_key_file": "/home/studen
```

```
TASK [nginx_test : debug] ******************************************************
task path: /home/student/cm/ansible/day-3/roles/nginx_test/tasks/main.yml:10
ok: [websvr] => {
    "msg": "nginx status: STARTED"
}

TASK [nginx_test : get start time] *********************************************
task path: /home/student/cm/ansible/day-3/roles/nginx_test/tasks/main.yml:14
changed: [websvr] => {"changed": true, "cmd": "echo Now: `date`\n pid=`ps aux | grep
: 0, "start": "2017-03-26 18:35:47.288600", "stderr": "", "stdout": "Now: Sun Mar 26
00"], "warnings": []}

RUNNING HANDLER [nginx : restart nginx] ****************************************
changed: [websvr] => {"changed": true, "name": "nginx", "state": "started"}

PLAY RECAP *********************************************************************
localhost                  : ok=4    changed=2    unreachable=0    failed=0
websvr                     : ok=22   changed=18   unreachable=0    failed=0

[student@epbyminw2473 day-3]$
```

Second run of playbook

```
TASK [nginx_test : debug] ******************************************************
task path: /home/student/cm/ansible/day-3/roles/nginx_test/tasks/main.yml:10
ok: [websvr] => {
    "msg": "nginx status: STARTED"
}

TASK [nginx_test : get start time] *********************************************
task path: /home/student/cm/ansible/day-3/roles/nginx_test/tasks/main.yml:14
changed: [websvr] => {"changed": true, "cmd": "echo Now: `date`\n pid=`ps aux | grep
: 0, "start": "2017-03-26 18:38:46.560145", "stderr": "", "stdout": "Now: Sun Mar 26
00"], "warnings": []}

PLAY RECAP *********************************************************************
localhost                  : ok=4    changed=1    unreachable=0    failed=0
websvr                     : ok=21   changed=8    unreachable=0    failed=0

[student@epbyminw2473 day-3]$
```

5. Verification Procedure: playbook will be checked by instructor's CI system as follows:
   5.1 Connect to student's host by ssh (username "student") with own ssh key.
   5.2 Go into the folder mentioned in point 1
   5.3 Destroy: vagrant destroy
   5.4 Execute VM provisioning: ansible-playbook stack.yml -i localhost, -c local -vv
   5.5 If previous steps are done successfully, instructor will check report (pdf-file)
6. Feedback: report issues/problems you had during the development of playbook and time spent for development.

APPENDIX A.

Playbook:

```
- hosts: localhost
  connection: local

  vars:
    mongo_src:
    - mongodb-linux-x86_64-rhel62-3.4.1
    - mongodb-linux-x86_64-rhel70-3.4.1
    - mongodb-linux-x86_64-rhel55-3.2.11
    - mongodb-linux-x86_64-rhel64-3.2.11
    - mongodb-linux-x86_64-rhel70-3.2.17
    - mongodb-linux-x86_64-rhel55-3.0.14
    - mongodb-linux-x86_64-rhel64-3.0.14
    - mongodb-linux-x86_64-rhel70-3.0.14

  tasks:
  - debug: msg={{ mongo_src | get_mongo_src("rhel", "7", "3.2" ) }}
  - debug: msg={{ mongo_src | get_mongo_src("rhel", "6", "3.0" ) }}
  - debug: msg={{ mongo_src | get_mongo_src("rhel", "7", "3.4" ) }}
```

Run:

```
$ ansible-playbook test1.yml -vv
Using /Users/sbeliakou/.ansible.cfg as config file
 [WARNING]: Host file not found: /etc/ansible/hosts
```

```
 [WARNING]: provided hosts list is empty, only localhost is available


PLAYBOOK: test1.yml ************************************************************
1 plays in test1.yml

PLAY [localhost] ***************************************************************

TASK [setup] *******************************************************************
ok: [localhost]

TASK [debug] *******************************************************************
task path: /private/tmp/test1.yml:19
ok: [localhost] => {
    "msg": "mongodb-linux-x86_64-rhel70-3.2.17"
}

TASK [debug] *******************************************************************
task path: /private/tmp/test1.yml:20
ok: [localhost] => {
    "msg": "mongodb-linux-x86_64-rhel64-3.0.14"
}

TASK [debug] *******************************************************************
task path: /private/tmp/test1.yml:21
```

```
ok: [localhost] => {
    "msg": "mongodb-linux-x86_64-rhel70-3.4.1"
}

PLAY RECAP *********************************************************************
localhost                  : ok=4    changed=0    unreachable=0    failed=0
```