

EPAM Systems, 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		

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

Developing custom modules and filters. Learning by doing.

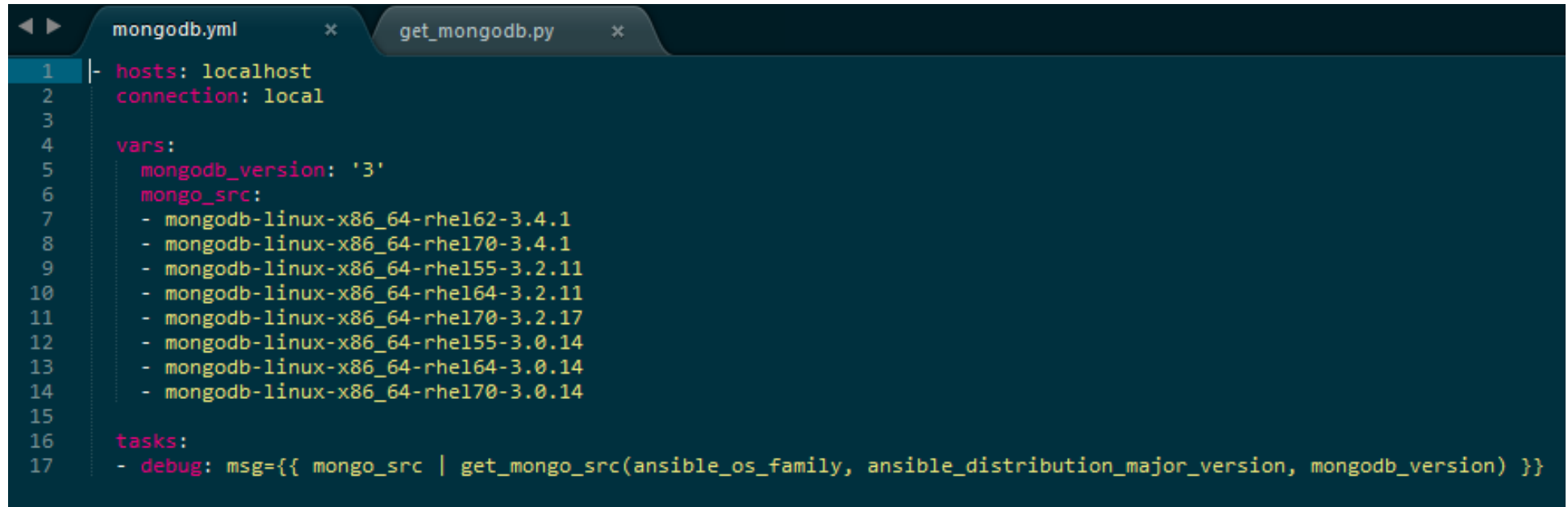
Task

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



```
1 |from __future__ import (absolute_import, division, print_function)
2 |__metaclass__ = type
3 |from ansible import errors
4 |import re
5 |
6 |dict={'RedHat':'rhel','Debian':'debian','Ubuntu':'ubuntu'}
7 |
8 |class FilterModule(object):
9 |
10 |    def filters(self):
11 |        return {
12 |            'get_mongo_src': get_mongo_src,
13 |        }
14 |
15 |def get_mongo_src(mongo_src, os_family, major_version, mongodb_version):
16 |    result = []
17 |
18 |    for i in range(len(mongo_src)):
19 |        if ((dict[os_family] + major_version) in mongo_src[i]) and (str(mongodb_version) in mongo_src[i]):
20 |            result.append(mongo_src[i])
21 |    return "Recomended MongoDB versions for {} distro are: {}".format(os_family, result)
22 |
```



```
1  hosts: localhost
2  connection: local
3
4  vars:
5      mongodb_version: '3'
6      mongo_src:
7          - mongodb-linux-x86_64-rhel62-3.4.1
8          - mongodb-linux-x86_64-rhel70-3.4.1
9          - mongodb-linux-x86_64-rhel55-3.2.11
10         - mongodb-linux-x86_64-rhel64-3.2.11
11         - mongodb-linux-x86_64-rhel70-3.2.17
12         - mongodb-linux-x86_64-rhel55-3.0.14
13         - mongodb-linux-x86_64-rhel64-3.0.14
14         - mongodb-linux-x86_64-rhel70-3.0.14
15
16  tasks:
17      - debug: msg={{ mongo_src | get_mongo_src(ansible_os_family, ansible_distribution_major_version, mongodb_version) }}
```

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 x
1 |#!/bin/sh
2 |#$1 - First parameter - path to vagrantfile
3 |#$2 - Second parameter - state: started, stopped, destroyed
4 |
5 |### check input vars
6 |source $1
7 |
8 |if [ -z "$path" ]; then
9 |    printf '{"failed": true, "msg": "Missing required arguments: path"}'
10 |    exit 1
11 |fi
12 |if [ -z "$state" ]; then
13 |    printf '{"failed": true, "msg": "Missing required arguments: state"}'
14 |    exit 1
15 |fi
16 |
17 |### check if Vagrantfile exists
18 |if [ ! -f "$path" ]; then
19 |    printf '{"failed": true, "msg": "Missing Vagrantfile"}'
20 |    exit 1
21 |fi
22 |###
23 |
24 |
25 |function get_vars
26 |{
27 |    ip=$(vagrant ssh-config | grep HostName | awk '{print $2}')
28 |    port=$(vagrant ssh-config | grep Port | awk '{print $2}')
29 |    user=$(vagrant ssh-config | grep -w "User" | awk '{print $2}' 2>/dev/null)
30 |    key=$(vagrant ssh-config | grep IdentityFile | awk '{print $2}' 2>/dev/null)
31 |    status=$(vagrant status | grep default | awk '{print $2}')
32 |    os_name=$(vagrant ssh -c "cat /etc/redhat-release" 2>/dev/null)
33 |    ram=$(vagrant ssh -c "cat /proc/meminfo | grep MemTotal | awk '{print \$2}'" 2>/dev/null)
34 |}
35 |
36 |
```

```
37 ▼ function vagrant_up
38     {
39         status=$(vagrant status | grep default | awk '{print $2}')
40 ▼     if [ "$status" == "running" ]; then
41         get_vars
42         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" "$ram"
43         exit 0
44 ▼     else
45         vagrant up &>/dev/null
46         get_vars
47         printf '{"failed": false, "changed": true, "ip": "%s", "port": "%s", "user": "%s", "key": "%s", "status": "%s", "os_name": "%s", "ram": "%s"}' "$ip" "$port" "$user" "$key" "$status" "$os_name" "$ram"
48     fi
49 }
50
51
52 function vagrant_halt
53 ▼ { status=$(vagrant status | grep default | awk '{print $2}')
54 ▼     if [ "$status" == "running" ]; then
55         vagrant halt
56         status=$(vagrant status | grep default | awk '{print $2}')
57         printf '{"failed": false, "changed": true, "status": "%s"}' "$status"
58         exit 0
59 ▼     else
60         changed="false"
61         failed="false"
62         printf '{"failed": false, "changed": true, "status": "%s"}' "$status"
63     fi
64 }
65
66
67 function vagrant_destroy
68 ▼ { status=$(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
1  |- name: stack
2     hosts: localhost
3
4     vars:
5         state: started
6
7     tasks:
8         - name: vagrant provision
9           prov: path=Vagrantfile state={{state}}
10          register: vagrant_vars
11         - debug: msg={{vagrant_vars}}
12         - add_host:
13             name: websvr
14             ansible_host: "{{vagrant_vars.ip}}"
15             ansible_port: "{{vagrant_vars.port}}"
16             ansible_connection: ssh
17             ansible_user: "{{vagrant_vars.user}}"
18             ansible_ssh_private_key_file: "{{vagrant_vars.key}}"
19             when: vagrant_vars.status == "running"
20
21     - name: tomcat site
22       hosts: websvr
23       roles:
24         - { role: java}
25         - { role: tomcat}
26         - { role: nginx}
27         - { role: java_test}
28         - { role: tomcat_test}
29         - { role: nginx_test}
30
```

```
[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/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 [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
```