Ansible EXERCISE 3

1. Install Ansible from Instructions from last session
2. Check versions if ansible installed properly:
3. ansible - -version
4. ansible-playbook - -version
5. ansible-galaxy - -version
6. ansible - -list-hosts all
7. sudo vi /etc/ansible/hosts
8. check default configuration of ansbile in :
9. sudo vi /etc/ansible/ansbile.cfg

now lets create our own configuration from this folder

1. mkdir dev
2. Cd dev
3. vim dev
4. [loadbalancer]
5. lb01
6. [webserver]
7. app01
8. app02
9. [database]
10. db01
11. [control]
12. ansible ansible\_connection=local
13. save and exit
14. ansible -i dev --list-hosts all
15. and lets create a config file
16. vim ansible.cfg
17. [defaults]
18. inventory = ./dev
19. ansible --list-hosts all
20. ansible -m ping all
21. ansible -m command -a "hostname" all
22. now run :
23. ansible –a “hostname” all
24. should be the same result
25. Then I want to ansible it first (try it out)
26. Ansible <ip address> -i inventory -u <user> -m ping –k
27. I get an error like the demo
28. Ssh <user>@<ip-address> from ansible control server
29. Put yes in the prompt of the known hosts file
30. Now do the command from line 9 again
31. It should ok now
32. Now do the same to the other host ip
33. And run: ansible all –I inventory –u <user> -m ping –k
34. And it will on both servers
35. Now lets run verbose mode to see background noise
36. Ansible <ip address> -i inventory -u <user> -m ping -k -vvv
37. ansible all –I inventory –u <user> -m command –a "/usr/sbin/apt-get upgrade –y"
38. now we are going to create our Inventory File

with all the parameters & group:

1. mkdir exercise4
2. cd exercise4
3. vi inventory
4. and add lines:
5. web1 ansible\_ssh\_host=<ip-address of web1> ansible\_ssh\_user=<username> ansible\_ssh\_pass=<password>
6. db1 ansible\_ssh\_host=<ip-address of db1> ansible\_ssh\_user=<username> ansible\_ssh\_pass=<password>
7. [webservers]
8. web1
10. [dbservers]
11. db1
12. [datacenter:children]
13. webservers
14. dbservers
15. Save the file
16. Exit from vi
17. Now lets add variable in the inventory
18. To remove duplicate parameters that are common
19. [datacenter:vars]
    1. ansible\_ssh\_user=<username> ansible\_ssh\_pass=<password>
20. mkdir exercise5
21. mkdir exercise5/production
22. mkdir exercise5/test
23. mkdir exercise5/production/host\_vars
24. mkdir exercise5/production/group\_vars
25. mkdir exercise5/test/host\_vars
26. mkdir exercise5/test/group\_vars
27. cd exercise5/production
28. copy inventory file from exercise3 to production folder
29. cd group\_vars
30. vim all
31. ----

# this is our user

Username: all\_username

1. Save and exit
2. Cd ..
3. Ansible webservers –I inventory\_prod –m user –a "name={{username}} password 123456" –sudo
4. Now let cd group\_vars
5. Now lets create group file webservers
6. Vim webservers
7. ----
8. # this is a good comment
9. username: group\_user
10. Save and exit
11. Cd ..
12. Now run 58 command again
13. Now let go to host level
14. Cd host\_vars
15. Vim web1
16. ----

#this is a comment

Username: web1\_user

Save and exit

1. Now rerun the 58 command again
2. Now add keys to remote hosts to allow ansible to connect without a password in ssh
3. Just follow the instructions from the last presentation
4. Now lets install Webserver in with apt-get modules
5. Ansible webservers –I inventory –m apt-get –a "name=apache2 state=present" –-sudo
6. Ansible-doc service
7. Ansible webservers –I inventory –m service –a "name=apache2 enabled=yes state=started" --sudo
8. Now lets install mysql in db server
9. Ansible dbservers –I inventory –m apt-get –a "name=mysql-server state=present" –-sudo
10. Ansible dbservers –I inventory –m service –a "name=mysqld state=started" --sudo
11. Now lets stop all firewall in all servers
12. Ansible dbservers:webservers –I inventory –m service –a "name=iptables state=stopped" –sudo
13. Now let practice : Using Setup Module:
14. ansible web1 -i inventory -m setup
15. to filter the facts:
16. ansible web1 -i inventory -m setup -a “filter=ansible\_eth\*”
17. ansible web1 -i inventory -m setup -a “filter=ansible\_mounts”
18. now lets check tree in ll hosts
19. ansible all -i inventory -m setup –tree ./setup
20. ls
21. cd setup
22. ls
23. cat <server name>
24. Now lets write a playbook:
25. Create a file in atom under playbooks :
26. loadbalancer.yml
27. ---
28. - hosts: loadbalancer
29. become: true # give sudo privleges
30. tasks:
31. - name: install nginx
32. apt: name=nginx state=present update\_cache=yes

create another file in atom under playbooks:

database.yml

---

- hosts: database

become: true # give sudo privleges

tasks:

- name: install mysql-server

apt: name=mysql-server state=present update\_cache=yes

now lets create webserver playbook:

webserver.yml:

---

- hosts: webserver

become: true # give sudo privleges

tasks:

- name: install web components

apt: name={{item}} state=present update\_cache=yes

with\_items:

- apache2

- libapache2-mod-wsgi

- python-pip

- pyhton-virtualenv

now run each playbook:

ansible-playbooks playbooks/…

after you executed all the playbooks

run them again to seed different results and fast feedback

now lets add service handling in the playbooks

edit loadbalancer.yml in atom:

---

- hosts: loadbalancer

become: true # give sudo privleges

tasks:

- name: install nginx

apt: name=nginx state=present update\_cache=yes

- name: ensure nginx service is running

service: name=nginx state=started enabled=yes

now run this playbook

now check if nginx is serving:

wget -qO- http://lb01 | less

make sure you copy the service in loadbalancer playbook to the others and run them as well

now lets make another in atom under playbooks:

stack\_restart.yml

---

# Bring the stack down

- hosts: loadbalancer

become: true # give sudo privleges

tasks:

- service: name=nginx state=stopped

- hosts: webserver

become: true # give sudo privleges

tasks:

- service: name=apache2 state=stopped

- hosts: database

become: true # give sudo privleges

tasks:

- service: name=mysql state=restarted

# Bring the stack up

- hosts: webserver

become: true # give sudo privleges

tasks:

- service: name=apache2 state=started

- hosts: loadbalancer

become: true # give sudo privleges

tasks:

- service: name=nginx state=started

now lets add handlers so it won’t restart every run :

vim playbooks/webserver.yml

---

- hosts: webserver

become: true # give sudo privleges

tasks:

- name: install web components

apt: name={{item}} state=present update\_cache=yes

with\_items:

- apache2

- libapache2-mod-wsgi

- python-pip

- python-virtualenv

- name: ensure apache2 service is running

service: name=apache2 state=started enabled=yes

- name: ensure mod\_wsgi enabled

apache2\_module: state=present name=wsgi

notify: restart apache2

- name: copy demo app source

copy: src=demo/app/ dest=/var/www/demo mode=8755

notify: restart apache2

handlers:

- name: restart apache2

service: name=apache2 state=restarted

and we added the copy module for copy content to the webserver

make sure you copy demo app to files under playbooks

now lets add another copy task to the webserver playbook of apache config

---

- hosts: webserver

become: true # give sudo privleges

tasks:

- name: install web components

apt: name={{item}} state=present update\_cache=yes

with\_items:

- apache2

- libapache2-mod-wsgi

- python-pip

- python-virtualenv

- name: ensure apache2 service is running

service: name=apache2 state=started enabled=yes

- name: ensure mod\_wsgi enabled

apache2\_module: state=present name=wsgi

notify: restart apache2

- name: copy demo app source

copy: src=demo/app/ dest=/var/www/demo mode=8755

notify: restart apache2

- name: copy apache virtual host config

copy: src=demo/demo.conf dest=/etc/apache2/sites-available mode=8755

notify: restart apache2

handlers:

- name: restart apache2

service: name=apache2 state=restarted

Roles:

Mkdir roles

Cd roles

Ansible-galaxy init control

Ansible-galaxy init nginx

Ansible-galaxy init apache2

Ansible-galaxy init demo\_app

Ansible-galaxy init mysql

Check main.yml

Copy control playbook to main.yml without the hosts – only taks

Name:

…..

Now lets change control.yml to :

---

* hosts: control

become: true

roles:

- control

now lets run ansible-playbook control.yml

ansible –m setup db01