Configuration Management:

Environment: Ubuntu 20 LTS, ansible 2.9.16, puppet 4.5

1. Which ansible command can display all ansible\_ configuration for a host.

Answer: ansible -m setup your\_hostname

2. To configure a cron job that runs logrotate on all machines every 10 minutes between 2h - 4h, you can use an Ansible playbook. Here's the playbook:

---

- name: Configure logrotate cron job

hosts: all

become: yes

tasks:

- name: Add logrotate cron job

cron:

name: "Run logrotate"

minute: "\*/10"

hour: "2-3"

job: "/usr/sbin/logrotate /etc/logrotate.conf"

3. To deploy ntpd package to the following 3 servers:

app-vm1.fra1.internal (192.168.0.2)

db-vm1.fra1.db (192.168.0.3)

web-vm1.fra1.web (192.168.0.4)

with the custom config file attached you can use the ansible palybook below:

---

- name: Deploy ntpd with custom configuration

hosts:

- app-vm1.fra1.internal

- db-vm1.fra1.db

- web-vm1.fra1.web

become: yes

tasks:

- name: Install ntpd package

package:

name: ntp

state: present

- name: Copy custom ntpd configuration

copy:

content: |

tinker panic 0

restrict default kod nomodify notrap nopeer noquery

restrict -6 default kod nomodify notrap nopeer noquery

restrict 127.0.0.1

restrict -6 ::1

server 192.168.0.252 minpoll 4 maxpoll 8

server 192.168.0.251 minpoll 4 maxpoll 8

server 192.168.0.0 # local clock

fudge 192.168.0.0 stratum 10

driftfile /var/lib/ntp/drift

keys /etc/ntp/keys

dest: /etc/ntp.conf

owner: root

group: root

mode: '0644'

notify: Restart ntpd

- name: Ensure ntpd service is enabled and started

service:

name: ntpd

state: started

enabled: yes

handlers:

- name: Restart ntpd

service:

name: ntpd

state: restarted

Edit your /etc/ansible/hosts file as below to execute the playbook to the mentioned hosts:

[ntp\_servers]

app-vm1.fra1.internal ansible\_host=192.168.0.2

db-vm1.fra1.db ansible\_host=192.168.0.3

web-vm1.fra1.web ansible\_host=192.168.0.4

To deploy the monitoring templates for the specified machines onto your Nagios server "monitoring.fra1.internal", you'll need to create configuration files based on the templates you've provided. Here's how you can do this using an Ansible playbook:

---

- name: Deploy Nagios configurations

hosts: monitoring.fra1.internal

become: yes

tasks:

- name: Ensure Nagios config directory exists

file:

path: /etc/nagios/conf.d

state: directory

mode: '0755'

- name: Deploy host configurations

template:

src: host\_template.cfg.j2

dest: "/etc/nagios/conf.d/{{ item.name }}\_host.cfg"

mode: '0644'

loop:

- { name: 'app-vm1.fra1.internal', ip: '192.168.0.2' }

- { name: ' db-vm1.fra1.db', ip: '192.168.0.3' }

- { name: 'web-vm1.fra1.web', ip: '192.168.0.4' }

- name: Deploy service configurations

template:

src: service\_template.cfg.j2

dest: "/etc/nagios/conf.d/{{ item }}\_service.cfg"

mode: '0644'

loop:

- 'app-vm1.fra1.internal'

- 'db-vm1.fra1.db'

- 'web-vm1.fra1.web'

- name: Restart Nagios

service:

name: nagios

state: restarted

You have to save this yml file and then need to create a template file for the host configuration named host\_template.cfg.j2:

define host {

host\_name {{ item.name }}

address {{ item.ip }}

check\_command check­\_ping

active\_checks\_enabled 1

passive\_checks\_enabled 1

}

Also create a template file for the service configuration named service\_template.cfg.j2:

define service {

service\_description ntp\_process

host\_name {{ item }}

check\_command check\_ntp

check\_interval 10

}

Make sure your Ansible inventory includes the Nagios server.