Juanson, Aliya Dane P.
CPE212-CPE31S2

Tools Needed:
1. VM with Ubuntu, CentOS and Ansible installed 2. Web browser

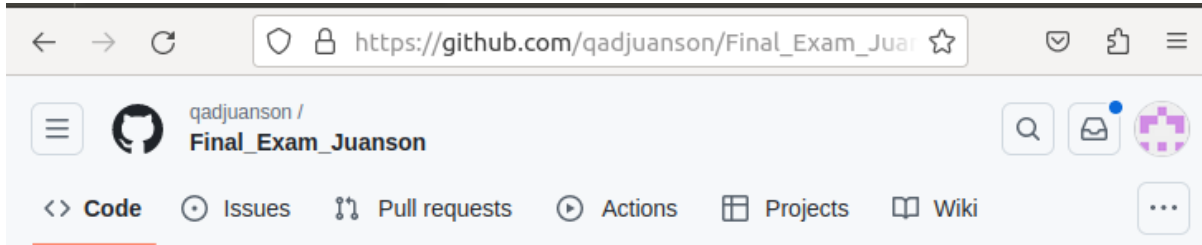| Procedure: |
|---|
| 1. Create a repository and label it as "Final_Exam_Surname"

2. Clone your new repository in your VM

3. Create an Ansible playbook that does the following with an input of a config.yaml file and structure inventory file.

3.1 Install and configure one enterprise service that can be installed in Debian and Centos servers

3.2 Install and configure one monitoring tool that can be installed in Debian and Centos servers (if it is a stack there should be option of different host)

4.4 Change Motd as "Ansible Managed by <username>"

4. Push and commit your files in GitHub

5. Make sure to show evidence of input (codes) process (codes successfully running) and output (evidence of installation)

5. For your final exam to be counted, please paste your repository link as an answer in this exam.

Note: Extra points if you will implement the said services via containerization. |

Screenshots:

1. Create a repository and clone it.

← → C   ○ 🔒 https://github.com/qadjuanson/Final_Exam_Jua ☆   ♡ ⑂ ≡

≡   qadjuanson /
    Final_Exam_Juanson   Q   ⊡

<> **Code**   ⊙ Issues   ⇅ Pull requests   ⊙ Actions   ⊞ Projects   📖 Wiki   ···

```
qadjuanson@workstation:~$ git clone git@github.com:qadjuanson/Final_Exam_Juanso
n.git
Cloning into 'Final_Exam_Juanson'...
warning: You appear to have cloned an empty repository.
```

2. Create a file named *inventory* and *ansible.cfg.*

```
qadjuanson@workstation:~/Final_Exam_Juanson$ nano inventory
qadjuanson@workstation:~/Final_Exam_Juanson$ cat inventory
[Ubuntu]
192.168.56.116
[Centos]
192.168.56.115
qadjuanson@workstation:~/Final_Exam_Juanson$ nano ansible.cfg
qadjuanson@workstation:~/Final_Exam_Juanson$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = qadjuanson
host_key_checking = true
deprecation_warnings = false
```

3. Create a playbook named *config.yaml.* It must contain installation and configuration of one enterprise service (Apache) and one monitoring tool (Nagios), that can be installed in Debian and Centos servers.

```yaml
---
- hosts: all
  become: true
  gather_facts: yes

  tasks:
  - name: install updates
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for Ubuntu Servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"

  - name: Install required dependencies on Ubuntu
    apt:
      name:
        - gcc
        - libc6
        - make
        - wget
        - unzip
        - apache2
        - php
        - libgd-dev
        - openssl
        - libssl-dev
        - autoconf
        - bc
        - gawk
        - dc
        - build-essential
        - snmp
        - libnet-snmp-perl
```

```yaml
          - gettext
      state: present
    when: ansible_distribution == "Ubuntu"

  - name: Install required dependencies on CentOS
    yum:
      name:
          - gcc
          - glibc
          - glibc-common
          - wget
          - unzip
          - httpd
          - php
          - gd
          - gd-devel
          - perl
          - postfix
          - openssl
          - openssl-devel
          - make
          - autoconf
      state: present
    when: ansible_distribution == "CentOS"

  - name: Download Nagios Core source code
    get_url:
      url: "https://assets.nagios.com/downloads/nagioscore/releases/
nagios-4.5.6.tar.gz"
      dest: /tmp/nagios-4.5.6.tar.gz

  - name: Extract Nagios source code
    unarchive:
      src: /tmp/nagios-4.5.6.tar.gz
      dest: /tmp
      remote_src: yes

  - name: Download Nagios Plugins
    get_url:
      url: "https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz"
      dest: /tmp/nagios-plugins-2.4.11.tar.gz

  - name: Extract Nagios Plugins
    unarchive:
      src: /tmp/nagios-plugins-2.4.11.tar.gz
      dest: /tmp
      remote_src: yes

  - name: Create Nagios group
    group:
      name: nagios

  - name: Create Nagios user and group
    user:
      name: nagios
      group: nagios
```

```yaml
  - name: Create nagcmd group
    group:
      name: nagcmd

  - name: Add nagios and apache/httpd users to nagcmd group
    user:
      name: "{{ item }}"
      groups: nagcmd
      append: yes
    loop:
      - nagios
      - "{{ 'www-data' if ansible_os_family == 'Debian' else 'apache' }}"

  - name: Compile and install Nagios Core
    shell: |
      cd /tmp/nagios-4.5.6
      ./configure --with-command-group=nagcmd
      make all
      make install
      make install-init
      make install-commandmode
      make install-config
      make install-webconf
    args:
      creates: /usr/local/nagios/bin/nagios

  - name: Install Nagios Plugins
    shell: |

      cd /tmp/nagios-plugins-2.4.11
      ./configure --with-nagios-user=nagios --with-nagios-group=nagios
      make
      make install
    args:
      creates: /usr/local/nagios/libexec/check_http

  - name: Set Nagios admin password
    command: htpasswd -b -c /usr/local/nagios/etc/htpasswd.users
nagios_qadjuanson "deng"

  - name: Enable and start Apache/Httpd service on Ubuntu
    service:
      name: apache2
      enabled: yes
      state: started
    when: ansible_distribution == "Ubuntu"

  - name: Enable and start Apache/Httpd service on CentOS
    service:
      name: httpd
      enabled: yes
      state: started
    when: ansible_distribution == "CentOS"

  - name: Enable and start Nagios service
    service:
      name: nagios
```

```yaml
- name: Enable and start Nagios service
  service:
    name: nagios
    enabled: yes
    state: started

- name: Enable external command execution in Nagios
  lineinfile:
    path: /usr/local/nagios/etc/nagios.cfg
    regexp: '^#?check_external_commands='
    line: 'check_external_commands=1'

- name: Restart Nagios service to apply changes
  service:
    name: nagios
    state: restarted

- name: Restart Apache/Httpd to apply changes on Ubuntu
  service:
    name: apache2
    state: restarted
  when: ansible_distribution == "Ubuntu"

- name: Restart Apache/Httpd to apply changes on CentOS
  service:
    name: httpd
    state: restarted
  when: ansible_distribution == "CentOS"
```

4. Run the playbook.

```
qadjuanson@workstation:~/Final_Exam_Juanson$ ansible-playbook config.yaml --ask
-become-pass
BECOME password:

PLAY [all] *******************************************************************
*

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.115]
ok: [192.168.56.116]

TASK [install updates] *******************************************************
*
skipping: [192.168.56.116]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] **********************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.116]

TASK [install apache and php for Ubuntu Servers] *****************************
*
skipping: [192.168.56.115]
ok: [192.168.56.116]

TASK [install apache and php for CentOS servers] *****************************
*
```

```
*
skipping: [192.168.56.116]
ok: [192.168.56.115]

TASK [Install required dependencies on Ubuntu] *******************************
*
skipping: [192.168.56.115]
ok: [192.168.56.116]

TASK [Install required dependencies on CentOS] *******************************
*
skipping: [192.168.56.116]
ok: [192.168.56.115]

TASK [Download Nagios Core source code] **************************************
*
ok: [192.168.56.115]
changed: [192.168.56.116]

TASK [Extract Nagios source code] ********************************************
*
changed: [192.168.56.116]
ok: [192.168.56.115]

TASK [Download Nagios Plugins] ***********************************************
*
ok: [192.168.56.115]
changed: [192.168.56.116]
```

```
TASK [Extract Nagios Plugins] *********************************************
*
changed: [192.168.56.116]
ok: [192.168.56.115]

TASK [Create Nagios group] ***********************************************
*
ok: [192.168.56.116]
ok: [192.168.56.115]

TASK [Create Nagios user and group] **************************************
*
ok: [192.168.56.116]
ok: [192.168.56.115]

TASK [Create nagcmd group] ***********************************************
*
ok: [192.168.56.116]
ok: [192.168.56.115]

TASK [Add nagios and apache/httpd users to nagcmd group] *****************
*
ok: [192.168.56.116] => (item=nagios)
ok: [192.168.56.116] => (item=www-data)
ok: [192.168.56.115] => (item=nagios)
ok: [192.168.56.115] => (item=apache)

TASK [Compile and install Nagios Core] ***********************************
*
```

```
..
ok: [192.168.56.116]
ok: [192.168.56.115]

TASK [Install Nagios Plugins] ********************************************
*
ok: [192.168.56.116]
ok: [192.168.56.115]

TASK [Set Nagios admin password] *****************************************
*
changed: [192.168.56.116]
changed: [192.168.56.115]

TASK [Enable and start Apache/Httpd service on Ubuntu] *******************
*
skipping: [192.168.56.115]
ok: [192.168.56.116]

TASK [Enable and start Apache/Httpd service on CentOS] *******************
*
skipping: [192.168.56.116]
ok: [192.168.56.115]

TASK [Enable and start Nagios service] ***********************************
*
ok: [192.168.56.116]
ok: [192.168.56.115]
```

```
TASK [Enable external command execution in Nagios] ****************************
*
ok: [192.168.56.116]
ok: [192.168.56.115]

TASK [Restart Nagios service to apply changes] *******************************
*
changed: [192.168.56.116]
changed: [192.168.56.115]

TASK [Restart Apache/Httpd to apply changes on Ubuntu] ***********************
*
skipping: [192.168.56.115]
changed: [192.168.56.116]

TASK [Restart Apache/Httpd to apply changes on CentOS] ***********************
*
skipping: [192.168.56.116]
changed: [192.168.56.115]

PLAY RECAP *******************************************************************
*
192.168.56.115             : ok=20    changed=3    unreachable=0    failed=0
skipped=5    rescued=0    ignored=0
192.168.56.116             : ok=20    changed=7    unreachable=0    failed=0
skipped=5    rescued=0    ignored=0
```

```
qadjuanson@server1:~$ sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2024-12-04 09:02:33 +08; 14min ago
  Process: 13987 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCE
  Process: 13993 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC
 Main PID: 13997 (apache2)
    Tasks: 6 (limit: 2318)
   CGroup: /system.slice/apache2.service
           ├─13997 /usr/sbin/apache2 -k start
           ├─14002 /usr/sbin/apache2 -k start
           ├─14003 /usr/sbin/apache2 -k start
           ├─14004 /usr/sbin/apache2 -k start
           ├─14005 /usr/sbin/apache2 -k start
           └─14006 /usr/sbin/apache2 -k start

Dec 04 09:02:33 server1 systemd[1]: Starting The Apache HTTP Server...
Dec 04 09:02:33 server1 apachectl[13993]: AH00558: apache2: Could not reliably
Dec 04 09:02:33 server1 systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

qadjuanson@localhost:~ — sudo systemctl status httpd

```
[qadjuanson@localhost ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di>
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Wed 2024-12-04 09:02:35 PST; 17min ago
       Docs: man:httpd.service(8)
   Main PID: 16963 (httpd)
     Status: "Total requests: 3; Idle/Busy workers 100/0;Requests/sec: 0.00282;>
      Tasks: 177 (limit: 10962)
     Memory: 33.8M
        CPU: 1.052s
     CGroup: /system.slice/httpd.service
             ├─16963 /usr/sbin/httpd -DFOREGROUND
             ├─16964 /usr/sbin/httpd -DFOREGROUND
             ├─16965 /usr/sbin/httpd -DFOREGROUND
             ├─16966 /usr/sbin/httpd -DFOREGROUND
             └─16967 /usr/sbin/httpd -DFOREGROUND

Dec 04 09:02:35 localhost.localdomain systemd[1]: Starting The Apache HTTP Serv>
Dec 04 09:02:35 localhost.localdomain httpd[16963]: AH00558: httpd: Could not r>
Dec 04 09:02:35 localhost.localdomain systemd[1]: Started The Apache HTTP Serve>
Dec 04 09:02:35 localhost.localdomain httpd[16963]: Server configured, listenin>
lines 1-22/22 (END)
```

```
qadjuanson@server1:~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.6
     Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset:
     Active: active (running) since Wed 2024-12-04 09:02:30 +08; 18min ago
       Docs: https://www.nagios.org/documentation
    Process: 13952 ExecStopPost=/bin/rm -f /usr/local/nagios/var/rw/nagios.cmd (c
    Process: 13951 ExecStop=/bin/kill -s TERM ${MAINPID} (code=exited, status=0/S
    Process: 13954 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/et
    Process: 13953 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios
   Main PID: 13955 (nagios)
      Tasks: 6 (limit: 2318)
     CGroup: /system.slice/nagios.service
             ├─13955 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios
             ├─13956 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/
             ├─13957 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/
             ├─13958 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/
             ├─13959 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/
             └─13990 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios

Dec 04 09:02:30 server1 nagios[13955]: qh: echo service query handler registere
Dec 04 09:02:30 server1 nagios[13955]: qh: help for the query handler registere
Dec 04 09:02:30 server1 nagios[13955]: wproc: Successfully registered manager a
Dec 04 09:02:30 server1 nagios[13955]: wproc: Registry request: name=Core Worke
Dec 04 09:02:30 server1 nagios[13955]: wproc: Registry request: name=Core Worke
Dec 04 09:02:30 server1 nagios[13955]: wproc: Registry request: name=Core Worke
Dec 04 09:02:30 server1 nagios[13955]: wproc: Registry request: name=Core Worke
Dec 04 09:02:30 server1 nagios[13955]: SERVICE FLAPPING ALERT: localhost;Curren
Dec 04 09:02:31 server1 nagios[13955]: Successfully launched command file worke
lines 1-27
```

```
[qadjuanson@localhost ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: d▷
     Active: active (running) since Wed 2024-12-04 09:02:30 PST; 19min ago
       Docs: https://www.nagios.org/documentation
    Process: 16773 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagi▷
    Process: 16774 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/▷
   Main PID: 16775 (nagios)
      Tasks: 6 (limit: 10962)
     Memory: 24.2M
        CPU: 631ms
     CGroup: /system.slice/nagios.service
             ├─16775 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagi▷
             ├─16776 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va▷
             ├─16777 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va▷
             ├─16778 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va▷
             ├─16779 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va▷
             └─16783 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagi▷

Dec 04 09:02:30 localhost.localdomain nagios[16775]: qh: Socket '/usr/local/nag▷
Dec 04 09:02:30 localhost.localdomain nagios[16775]: qh: core query handler reg▷
Dec 04 09:02:30 localhost.localdomain nagios[16775]: qh: echo service query han▷
Dec 04 09:02:30 localhost.localdomain nagios[16775]: qh: help for the query han▷
Dec 04 09:02:30 localhost.localdomain nagios[16775]: wproc: Successfully regist▷
lines 1-23...skipping...
```

5. Change Motd as "Ansible Managed by <qadjuanson>".

```
- name: Set MOTD
  copy:
    content: "Ansible Manage by qadjuanson"
    dest: /etc/motd
    owner: root
    group: root
    mode: 0644
```

```
qadjuanson@server1:~$ cat /etc/motd
Ansible Manage by qadjuansonqadjuanson@server1:~$
```

6. Push and commit your files in GitHub.

```
qadjuanson@workstation:~/Final_Exam_Juanson$ git add *
qadjuanson@workstation:~/Final_Exam_Juanson$ git commit -m "Final Exam Done"
[master (root-commit) ea2efd0] Final Exam Done
 3 files changed, 209 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.yaml
 create mode 100644 inventory
qadjuanson@workstation:~/Final_Exam_Juanson$ git push
Counting objects: 5, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.60 KiB | 1.60 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To github.com:qadjuanson/Final_Exam_Juanson.git
 * [new branch]      master -> master
```

qadjuanson Final Exam Done		ea2efd0 · 3 minutes ago
ansible.cfg	Final Exam Done	3 minutes ago
config.yaml	Final Exam Done	3 minutes ago
inventory	Final Exam Done	3 minutes ago

https://github.com/qadjuanson/Final_Exam_Juanson

Conclusion:

In this practical exam, I am able to install one enterprise service and one monitoring tool. I chose Apache for my enterprise service and while for the monitoring tool I chose Nagios. These two can be installed in Debian and Centos servers.