SYSTEMS AND SITES RELIABILITY ENGINEER HOMEWORK

Simple Web Application

Nodes:

a. Controller Node
   - Hostname : control-station. aulialabs.com
   - IP Address : 192.168.230.128
   - User : ansible

b. Managed Node (as Webserver)
   - Hostname : webserver.aulialabs.com
   - IP Address : 192.168.230.131
   - User : ansible

Controller Node

IP address:

```
[root@control-station ~]# ip a | grep ens33 -A1
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:8a:62:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.230.128/24 brd 192.168.230.255 scope global noprefixroute dynamic ens33
       valid_lft 1123sec preferred_lft 1123sec
[root@control-station ~]#
```

Hostname:

```
[root@control-station ~]# hostnamectl
   Static hostname: control-station.aulialabs.com
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 91f4c1f64fec44a7bb728dc19254dd10
           Boot ID: 5a6ce5e483d846bbaf69bdfbd53f380f
    Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-1062.el7.x86_64
      Architecture: x86-64
[root@control-station ~]#
```

Managed Node (Webserver)

IP address:

```
[root@webserver ~]# ip a | grep ens33 -A1
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:6b:ce:d7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.230.131/24 brd 192.168.230.255 scope global noprefixroute dynamic ens33
       valid_lft 1486sec preferred_lft 1486sec
[root@webserver ~]#
```

Hostname:

```
[root@webserver ~]# hostnamectl
   Static hostname: webserver.aulialabs.com
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 002f200525a949399a6d5b579fc5bbc3
           Boot ID: 4ee1591c03fb48be93aa20d00d98ea2e
    Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-1062.el7.x86_64
      Architecture: x86-64
[root@webserver ~]#
```

Prerequisite Tasks:

a. Ansible package and it dependencies already installed on Controller Node

b. Connection to the managed node should be passwordless

c. This task will be executed by the user "ansible"

On the Controller Node

1. Create user “ansible”

```
# useradd ansible
# passwd ansible
```

```
[root@control-station ~]# useradd ansible
[root@control-station ~]# cat /etc/passwd | grep ansible
ansible:x:1001:1001::/home/ansible:/bin/bash
[root@control-station ~]#
```

2. Grant user “ansible” as sudoers

```
# vim /etc/sudoers.d/ansible
  ansible ALL=(ALL) NOPASSWD:ALL
```

```
[root@control-station ~]# vim /etc/sudoers.d/ansible
[root@control-station ~]# ls -l /etc/sudoers.d/ansible
-rw-r--r--. 1 root root 31 Mar  2 04:59 /etc/sudoers.d/ansible
[root@control-station ~]#
```

3. Setup SSH passwordless as user “ansible”

```
# su ansible
$ ssh-keygen
$ ssh-copy-id localhost
$ ssh-copy-id
```

On the Managed Node

1. Create user “ansible”

```
# useradd ansible
# passwd ansible
```

```
[root@webserver ~]# useradd ansible
[root@webserver ~]# cat /etc/passwd | grep ansible
ansible:x:1001:1001::/home/ansible:/bin/bash
[root@webserver ~]#
```

2. Grant user “ansible” as sudoers

```
# vim /etc/sudoers.d/ansible
```

```
ansible ALL=(ALL) NOPASSWD:ALL
```

```
[root@webserver ~]# vim /etc/sudoers.d/ansible
[root@webserver ~]# cat /etc/sudoers.d/ansible
ansible ALL=(ALL) NOPASSWD:ALL
[root@webserver ~]#
```

3. Setup SSH passwordless

```
# su ansible
$ ssh-keygen
$ ssh-copy-id localhost
$ ssh-copy-id
```

Setup the Web Server

1. Create an inventory file at /home/ansible/inventory

```
$ vi /home/ansible/inventory
```

```
[control]
control-station.aulialabs.com

[webserver]
webserver.aulialabs.com
```

2. Test ping to All nodes (Controler & Managed Node) from Controller Node to ensure the connection successfully established.

```
$ ansible -i /home/ansible/inventory -m ping all
```

```
[ansible@control-station ~]$ ansible -i /home/ansible/inventory -m ping all
webserver.aulialabs.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
control-station.aulialabs.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[ansible@control-station ~]$
```

3. Create a Jinja2 template file index.html.j2 is used to create the file /webserver/index.html

```
$ vi index.html.j2
```

```
The IP address of '{{ ansible_hostname }}' server is: '{{
ansible_default_ipv4.address }}'
```

```
[ansible@control-station ~]$ ls -lh
total 8.0K
-rw-rw-r--. 1 ansible ansible 99 Mar  2 04:20 index.html.j2
-rw-rw-r--. 1 ansible ansible 77 Mar  2 03:43 inventory
[ansible@control-station ~]$ cat index.html.j2
The IP address of '{{ ansible_hostname }}' server is: '{{ ansible_default_ipv4.address }}'
[ansible@control-station ~]$
```

4. Creating a playbook /home/ansible/webapp.yaml that runs on webserver node.

```
$ vi /home/ansible/webapp.yaml

---
- hosts: webserver
  become: yes
  vars:
    webdir: /webserver
    web_linked: /var/www/html/webserver
    pkgs: httpd, firewalld, mod_ssl
    jinja2_temp: index.html.j2
    all_svc:
      - httpd
      - firewalld
  tasks:
    - name: Install all packages needed
      yum :
        name: "{{ pkgs }}"
        state: latest

    - name: Enable and start the all service needed
      service:
        name: "{{ item }}"
        state: started
        enabled: yes
      loop: "{{ all_svc }}"

    - name: Allow all port needed for services to communicate
      firewalld:
        port: "{{ item }}"
        immediate: true
        permanent: true
        state: enabled
      loop:
        - 80/tcp
        - 443/tcp

    - name : Create a web directory
      file:
        state: directory
        dest: "{{ webdir }}"
        owner: apache
        group: apache
        mode: u=rwx,g=rx,o=rx

    - name: Install jinja2 template for a web content
      template:
        src: "{{ jinja2_temp }}"
```

```
        dest: "{{ webdir }}/index.html"
        owner: apache
        group: apache
        mode: u=rwx,g=rx,o=rx

    - name : Create symbolically linked from {{ webdir }} to {{
web_linked }}
      file:
        src: "{{ webdir }}"
        dest: "{{ web_linked }}"
        state: link

    - name: Set a selinux permission to {{ webdir }} and {{ web_linked
}}
      sefcontext:
        target: "{{ item }}"
        setype: httpd_sys_content_t
        state: present
      with_items:
        - "{{ web_linked }}(/.*)?"
        - "{{ webdir }}(/.*)?"

    - name: Re-write the selinux permission to {{ webdir }} and {{
web_linked }}
      command: "{{ item }}"
      loop:
        - "restorecon -vvRF {{ webdir }}"
        - "restorecon -vvRF {{ web_linked }}"
      notify: restart httpd

  handlers:
    - name: restart httpd
      service:
        name: httpd
        state: restarted
...
```

```yaml
---
- hosts: webserver
  become: yes
  vars:
    webdir: /webserver
    web_linked: /var/www/html/webserver
    pkgs: httpd, firewalld, mod_ssl
    jinja2_temp: index.html.j2
    all_svc:
      - httpd
      - firewalld
  tasks:
    - name: Install all packages needed
      yum :
        name: "{{ pkgs }}"
        state: latest

    - name: Enable and start the all service needed
      service:
        name: "{{ item }}"
        state: started
        enabled: yes
      loop: "{{ all_svc }}"

    - name: Allow all port needed for services to communicate
      firewalld:
        port: "{{ item }}"
        immediate: true
        permanent: true
        state: enabled
      loop:
        - 80/tcp
        - 443/tcp
    - name : Create a web directory
      file:
        state: directory
        dest: "{{ webdir }}"
        owner: apache
        group: apache
        mode: u=rwx,g=rx,o=rx

    - name: Install jinja2 template for a web content
      template:
        src: "{{ jinja2_temp }}"
        dest: "{{ webdir }}/index.html"
        owner: apache
        group: apache
        mode: u=rwx,g=rx,o=rx

    - name : Create symbolically linked from {{ webdir }} to {{ web_linked }}
      file:
        src: "{{ webdir }}"
        dest: "{{ web_linked }}"
        state: link

    - name: Set a selinux permission to {{ webdir }} and {{ web_linked }}
      sefcontext:
        target: "{{ item }}"
        setype: httpd_sys_content_t
        state: present
      with_items:
        - "{{ web_linked }}(/.*)?"
        - "{{ webdir }}(/.*)?"

    - name: Re-write the selinux permission to {{ webdir }} and {{ web_linked }}
      command: "{{ item }}"
      loop:
        - "restorecon -vvRF {{ webdir }}"
        - "restorecon -vvRF {{ web_linked }}"

  handlers:
    - name: restart httpd
      service:
        name: httpd
        state: restarted
...
```

5. Run playbook /home/ansible/webapp.yaml

```
$ ansible-playbook -i /home/ansible/inventory /home/ansible/webapp.yaml
```

```
[ansible@control-station ~]$ ansible-playbook -i /home/ansible/inventory /home/ansible/webapp.yaml

PLAY [webserver] ***************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [webserver.aulialabs.com]

TASK [Install all packages needed] *********************************************************************************
changed: [webserver.aulialabs.com]

TASK [Enable and start the all service needed] *********************************************************************
changed: [webserver.aulialabs.com] => (item=httpd)
ok: [webserver.aulialabs.com] => (item=firewalld)

TASK [Allow all port needed for services to communicate] ***********************************************************
changed: [webserver.aulialabs.com] => (item=80/tcp)
changed: [webserver.aulialabs.com] => (item=443/tcp)

TASK [Create a web directory] **************************************************************************************
changed: [webserver.aulialabs.com]

TASK [Install jinja2 template for a web content] *******************************************************************
changed: [webserver.aulialabs.com]

TASK [Create symbolically linked from /webserver to /var/www/html/webserver] ***************************************
changed: [webserver.aulialabs.com]

TASK [Set a selinux permission to /webserver and /var/www/html/webserver] ******************************************
changed: [webserver.aulialabs.com] => (item=/var/www/html/webserver(/.*)?)
changed: [webserver.aulialabs.com] => (item=/webserver(/.*)?)

TASK [Re-write the selinux permission to /webserver and /var/www/html/webserver] ***********************************
changed: [webserver.aulialabs.com] => (item=restorecon -vvRF /webserver)
changed: [webserver.aulialabs.com] => (item=restorecon -vvRF /var/www/html/webserver)

RUNNING HANDLER [restart httpd] ************************************************************************************
changed: [webserver.aulialabs.com]

PLAY RECAP *********************************************************************************************************
webserver.aulialabs.com    : ok=10   changed=9    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ansible@control-station ~]$
```

Here is the result after running the playbook: