Top of Form

**Task 13: change system default boot target**

- Create a playbook target.yml that change the current **multi-user.target** default boot target to **graphical.target** boot target

* 

**$ cat target.yml**

* 1. ---
  2. - hosts: all
  3. become: yes
  4. tasks:
  5. - name: change system default target
  6. file:
  7. src: /lib/systemd/system/graphical.target
  8. dest: /etc/systemd/system/default.target
  9. state: link

Question 9: Download and modify file

Create a file in /home/ismat/ansible/ called report.yml. Using this playbook, get a file called report.txt. Download the file from[http://classroom.example.com](https://classroom.example.com/) to all remote hosts at /root/report.txt. Then edit the lines in the file to provide the real information of the hosts. If a disk does not exist then write NONE. The file content of report.txt is

HOST= inventory hostname  
MEMORY=total memory in mb  
BIOS=bios version  
VDA\_DISK\_SIZE=disk size  
VDB\_DISK\_SIZE=disk size

Solution:

---

- name: copy file with custom information

hosts: all

tasks:

- name: get file

get\_url:

url: https://classroom.example.com/report.txt

dest: /root/report.txt

- name: chahge hostname

lineinfile:

path: /root/report.txt

line: HOST={{ ansible\_hostname }}

regex: ^HOST

state: present

- name: chahge memory

lineinfile:

path: /root/report.txt

line: MEMORY={{ ansible\_memtotal\_mb }}

regex: ^MEMORY

state: present

- name: chahge bios version

lineinfile:

path: /root/report.txt

line: BIOS={{ ansible\_bios\_version }}

regex: ^BIOS

state: present

- name: chahge vda

lineinfile:

path: /root/report.txt

line: VDA\_DISK\_SIZE={% if ansible\_devices.vda is defined %}{{ ansible\_devices.vda.size }} {% else %} NONE {% endif %}

regex: ^VDA\_DISK\_SIZE

state: present

- name: chahge vdb

lineinfile:

path: /root/report.txt

line: VDB\_DISK\_SIZE={% if ansible\_devices.sda is defined %}{{ ansible\_devices.sda.size }} {% else %} NONE {% endif %}

regex: ^VDB\_DISK\_SIZE

state: present

#### ... Question 12: Read content from custom web directory

Create a playbook called webdev.yml in /home/ismat/ansible. The playbook will create a directory /webdev on dev host. The permission of the directory is u=rwx,g=rw, other has no permission. Set group id for the folder and owner is webdev. Create a symbolic link from /webdev to /var/www/html/webdev. Serve a file from /webdev/index.html which displays the text “Development”. Curl http://node1.example.com/webdev/index.html to test

Write the playbook webdev.yml

---

- name: web development

hosts: dev

tasks:

- name: create webdev user

user:

name: webdev

state: present

- name: create directory

file:

path: /webdev

state: directory

owner: webdev

mode: u=rwx,g=rw,0=---,g+s

- name: create symbolic link

file:

src: /webdev

path: /var/www/html/webdev

state: link

- name: create index.html

copy:

content: Development

dest: /var/www/html/webdev/index.html

- name: install selinux policy

yum:

name: python3-policycoreutils

state: present

- name: allow httpd from custom directory

sefcontext:

target: '/webdev(/.\*)?'

setype: httpd\_sys\_content\_t

state: present

- name: restore the context

shell: restorecon -vR /webdev

...

Question 14: Write content filtered by host

Create a playbook called issue.yml in /home/ismat/ansible which changes the file /etc/issue on all managed nodes: If the host is a member of dev then write “Development” If the host is a member of test then write “Test” If the host is a member of prod then write “Production”

Solution:

Write a playbook /home/ismat/ansible/issue.yml

---

- name: change issue file

hosts: all

tasks:

- name: change dev hosts issue file

copy:

content: "Development"

dest: /etc/issue

when: '"dev" in group\_names'

- name: change test hosts issue file

copy:

content: "Test"

dest: /etc/issue

when: '"test" in group\_names'

- name: change prod hosts issue file

copy:

content: "Producation"

dest: /etc/issue

when: '"prod" in group\_names'

**Questions about file content –  variables & facts**

**Question: 1**

* + Create a playbook, which can do as shown below:
  + Create a file on all the servers at this path /etc/motd.
  + The file contains “MY SERVER”.

**Answer: 1**

[ali@control ansible]$ vi motd.yml

---

- hosts: all

become: yes

tasks:

- name: Write content in the file

copy:

dest: /etc/motd

content: "MY SERVER"

**Question: 2**

* + Create a playbook, which can do as shown below:
  + Create a file on all the servers at this path /etc/motd.
  + The file contains  “MY SERVER: <Server Name> “.

you replace the server Name with the actual HOSTNAME of the server.

**Question: 3**

* + Create a playbook, which can do as shown below:
  + The existing file (/etc/motd) should be removed ( if any).
  + host group proxy should have the line “This Is PROXY Server”.
  + Host group webservers should have the line “These Are The WEBSERVERS”.
  + Host group database should have the line “This is “DATABASE server

**Answer: 3**

[ali@control ansible]$ cat motd.q3.yml

---

- hosts: all

become: yes

tasks:

- name: Generate the file motd content

file:

path: /etc/mtd

state: absent

- name: Proxy Server file content

copy:

dest: /etc/motd

content: "This Is PROXY Server"

when: "'proxy' in group\_names"

- name: WEBSERVERS file content

copy:

dest: /etc/motd

content: "These Are The WEBSERVERS"

when: "'webservers' in group\_names"

- name: Database Server file content

copy:

dest: /etc/motd

content: "This is DATABASE Server&amp;amp;amp;amp;amp;amp;amp;quot"

when: "'database' in group\_names"

**Question: 4**

* + In this question, you are going to practice the configuration of the SSH Server.
  + Create a playbook, which can do the following tasks:

Set banner to /etc/motd

X11Forwarding is disabled

MaxAuthTries is set to 3

**Answer: 4**

---

- hosts: all

become: yes

tasks:

- name: Set Banner to /etc/motd

lineinfile:

path: /etc/ssh/sshd\_config

regex: "^Banner"

line: "Banner /etc/motd"

- name: Disable X11Forwarding

lineinfile:

path: /etc/ssh/sshd\_config

regex: "^X11Forwarding"

line: "X11Forwarding no"

- name: Set MaxAuthTries to 3

lineinfile:

path: /etc/ssh/sshd\_config

regex: "^MaxAuthTries"

line: "MaxAuthTries 3"

**Question: 5**

* Create a playbook such as servers.yml, which should generate the network file as shown below:
* Servers.yml must use a jinja2 file called servers.j2, and when servers.yml is executed, it creates the file servers.txt on the servers group of webservers and database servers.  
  for example, your file should contain  information as shown below:

<FQDN> <HOSTNAME> <IPADDRESS>

node1.seimaxim.com node1 192.168.1.12

**Answer: 5**

cat servers.yml

---

- hosts: all

become: yes

tasks:

- name: Generating a file by the jinja2 file

template:

src: servers.j2

dest: /root/servers.txt

[ali@control ansible]$ vi servers.j2

{% for host in groups['webservers']%}

{{ hostvars[host]['ansible\_facts']['fqdn']}} {{ hostvars[host]['ansible\_facts']['hostname']}} {{ hostvars[host]['ansible\_facts']['default\_ipv4']['address']}}

{% endfor %}

To check if the files has been generated.

[ali@control ansible]$ ansible all -m shell -a "cat /root/servers\*" -b

node3 | CHANGED |

node2.seimaxim.com node2 10.0.2.15

node3.seimaxim.com node3 10.0.2.15

node4 | CHANGED |

node2.seimaxim.com node2 10.0.2.15

node3.seimaxim.com node3 10.0.2.15

node1 | CHANGED |

node2.seimaxim.com node2 10.0.2.15

node3.seimaxim.com node3 10.0.2.15

node2 | CHANGED |

node2.seimaxim.com node2 10.0.2.15

node3.seimaxim.com node3 10.0.2.15

**Question: 6**

* Create a playbook, which generates a file named “details.txt” file on each server with the following information:  
  The file should be placed in the root directory: /root

HOSTNAME: <hostname>

MEMORY\_TOTAL: <Total actual memory should be presented here>

MEMORY\_FREE: <Free Memory available on the system>

IPV4: <IP address of the system>

FQDN: <fqdn>

SDA\_DISK\_SIZE: </dev/sda1 disk size>

SDC\_DISK\_SIZE: </dev/sdc> #if disk is not attach,it should put NONE

**Answer: 6**

---

- hosts: all

become: yes

tasks:

- name: generate a report

blockinfile:

path: /root/report.txt

create: yes

block: |

MEMORY\_TOTAL: {{ ansible\_facts['memtotal\_mb' ]}}

MEMORY\_FREE: {{ ansible\_facts['memfree\_mb' ]}}

HOSTNAME: {{ ansible\_facts['hostname']}}

IPV4: {{ ansible\_facts['default\_ipv4']['address']}}

FQDN: {{ ansible\_facts['fqdn'] }}

SDA\_DISK\_SIZE: {{ ansible\_facts['devices']['sda']['partitions']['sda1']['size'] }}

SDB\_DISK\_SIZE: {{ ansible\_devices.sdc.size | default('NONE') }}

#To check the report's content run shown below command

[ali@control ansible]$ ansible all -m shell -a "cat /root/report.txt" -b

**Question: 7**

* Create a playbook that checks the file’s existence.  
  print the message “File Exists” if the file exists and print the message “File does not exist”if the file does not exist.

---

- hosts: all

become: yes

tasks:

- name: Check if the file exists

stat:

path: /var/www/html/index.html # or you can put the path of your desired file

register: result

- debug:

msg: "THE FILE EXISTS"

when: result.stat.exists == true

- debug:

msg: "THE FILE DOES'T EXIST"

when: result.stat.exists == false

**Question: 8**

* Create a playbook that performs the shown below actions:
* Change the line “Listen 80” in the /etc/httpd/conf/httpd.conf to “Listen 8080”.
* Allow the port 8080 in firewall and restart the httpd services

**Answer: 8**

# you must have installed httpd or you can put any other file to practice this senario

---

- hosts: webservers

become: yes

tasks:

- name: Modify the content of httpd.conf

replace:

path: /etc/httpd/conf/httpd.conf

regexp: 'Listen 80'

replace: 'Listen 8080'

notify: restart\_httpd

handlers:

- name: restart\_httpd # must be same as notify

service:

name: httpd

state: restarted

**Question: 9**

* Create a playbook that performs the shown below actions:
* Change the default target to the multi-user.target with the file module by using links. don’t use shell command to set-default target.

**Answer: 9**

---

- hosts: database

become: yes

tasks:

- name: Change the default target

file:

src: /usr/lib/systemd/system/multi-user.target

dest: /etc/systemd/system/default.target

state: link

**Question: 10**

* Create a playbook that performs the shown below actions:
* Run this playbook against the webservers group.
* A custom Ansible fact **server\_role=webservers** is created that can be retrieved from **ansible\_local.custom.sample\_exam** when using Ansible setup module.

**Answer: 10**

---

- hosts: webservers

become: yes

tasks:

- name: Ensure the directory exists

file:

path: "/etc/ansible/facts.d/"

state: directory

recurse: yes

- name: Copy content to file in new directory

copy:

content: "[sample\_exam]nserver\_role=webserversn"

dest: "/etc/ansible/facts.d/custom.fact"

**Task 3: File Content**

Create a playbook /home/automation/plays/motd.yml that runs on all inventory hosts and does the following:

* The playbook should replace any existing content of /etc/motd with text. Text depends on the host group.
* On hosts in the **proxy** host group the line should be “Welcome to HAProxy server”.
* On hosts in the **webservers** host group the line should be “Welcome to Apache server”.
* On hosts in the **database** host group the line should be “Welcome to MySQL server”.

I used the [magic variable](https://docs.ansible.com/ansible/latest/reference_appendices/special_variables.html) inventory\_hostname to match the conditions:

---

- name: task 3

hosts: all

become: yes

tasks:

- name: copy content to HAProxy

copy:

content: "Welcome to HAProxy server"

dest: /etc/motd

when: inventory\_hostname in groups["proxy"]

- name: copy content to webservers

copy:

content: "Welcome to Apache server"

dest: /etc/motd

when: inventory\_hostname in groups["webservers"]

- name: copy content to Database

copy:

content: "Welcome to MySQL server"

dest: /etc/motd

when: inventory\_hostname in groups["database"]

**Task 4: Configure SSH Server**

Create a playbook /home/automation/plays/sshd.yml that runs on all inventory hosts and configures SSHD daemon as follows:

* banner is set to /etc/motd
* X11Forwarding is disabled
* MaxAuthTries is set to 3

---

- name: task 4

hosts: all

become: yes

tasks:

- name: configure sshd daemon

lineinfile:

path: /etc/ssh/sshd\_config

regexp: "^Banner"

line: Banner /etc/motd

- name: disable X11Forwarding

lineinfile:

path: /etc/ssh/sshd\_config

regexp: "^X11Forwarding"

line: X11Forwarding no

- name: set MaxAuthTries = 3

lineinfile:

path: /etc/ssh/sshd\_config

regexp: "^MaxAuthTries"

line: MaxAuthTries 3

- name: restart ssh server

service:

name: sshd

state: restarted

enabled: yes

**Task 13: Use Conditionals to Control Play Execution**

Create a playbook /home/automation/plays/sysctl.yml that runs on all inventory hosts and does the following:

* If a server has more than 2048MB of RAM, then parameter **vm.swappiness** is set to 10.
* If a server has less than 2048MB of RAM, then the following error message is displayed:

**Server memory less than 2048MB**

To find the *fact*:

$ ansible ansible2.hl.local -m setup | grep -A10 memory

[...]

"ansible\_memory\_mb": {

"nocache": {

"free": 1447,

"used": 343

},

"real": {

"free": 961,

"total": 1790,

"used": 829

},

"swap": {

[...]

So the variable to use is ansible\_memory\_mb.real.total. And the playbook:

---

- name: task 13

hosts: all

become: true

tasks:

- name: set vm.swappiness to 10 if server has 2GB memory

sysctl:

name: vm.swappiness

value: 10

state: present

when:

- ansible\_memory\_mb.real.total >= 2048

- name: report not enough total memory

debug:

msg: "Server memory less than 2048MB ({{ ansible\_memory\_mb.real.total }}MB)"

when:

- ansible\_memory\_mb.real.total < 2048

- name: Create a tar.gz archive of a single file.

community.general.archive:

path: /path/to/foo/single.file

dest: /path/file.tar.gz

format: gz

force\_archive: true