**ANSIBLE AUTOMATION**

1. Dynamic inventory

The inventory was to be set to /opt/ansible/dynamic/inventory and timeout to 60. Once you correctly configured it show group of hosts majori, bunnies, newbunnies.

vim /etc/ansible/ansible.conf

inventory = /opt/ansible/dynamic/inventory

timeout = 60

1. Create /home/ansible/playbook/httpd.yml
   1. which should install http service
   2. Copy configuration file using get\_url from link <http://classroom.example.com/content/httpd.conf> to /etc/httpd/conf/httpd.conf
   3. Change /var/www/html/index.html file using lineinfile module so that it's content should be "Hello from nodeX". If the file exists it should not overwrite.
   4. Start the service of httpd and enable it.

It may not show the output, may show permission denied.

1. Create a /home/ansible/playbook/httpd-auth.yml file such that in all the nodes
   1. It creates /var/www/html/private directory
   2. copy a file from http://classroom.example.com/content/htaccess-custom to /var/www/html/private/.htaccess file
   3. copy a file from http://classroom.example.com/content/htpasswd to /var/www/html/private/.htpasswd
   4. Implement appropriate permission so that apache can read them
   5. Replace all instances of 'AllowOverride none' with 'Allowoverride AuthConfig' in /etc/httpd/conf/httpd.conf using replace module
   6. Create /var/www/html/private/index.html to read as this node nodeX is protected

You can test you configuration using username ansible and password **ansiansi**. It may not show the output, may show permission denied on one or two systems.

1. Create a script file /home/ansible/playbook/adhoc.sh
   1. This should copy the file /usr/local/bin/runme.sh to /usr/bin/runme.sh on all nodes.
   2. This should run as user ansible.
   3. The file should have execute permissions
2. Create custom groups named production and backup such that node1 and node2 belongs to production and node3 and node4 belongs to backup. Create a tag.yml which should be able to do the following:-
   1. Create a tarball /var/tmp/production.tar.gz for the production group containing the /var/www/html directory contents when run with tag production.
   2. Create a tarball /var/tmp/backup.tar.gz for the backup group containing the /var/log/httpd directory contents when run with tag backup.
3. Create /home/ansible/playbook/inaccess.yml
   1. If the url http://classroom.example.com/content/access.html is accessible it should copy the file to /var/www/html/
   2. You must use block groups rescue group
   3. Otherwise it should create a file noaccess.html with the content as “I have no access."

1. Create a file hostinfo.yml which should store the following system information in /root/hostinfo.txt in a single line nodeX processorcount defaultNICIPV4address defaultNICmacaddress
2. Create ping.yml so that

a) It creates /var/www/html/tag1.html in node 1 & 3 with content "I am tag1" when run as ansible-playbook --tags tag1 ping.yml

b) It creates /var/www/html/tag2.html in node 2 & 4 with content "I am tag2" when run as ansible-playbook --tags tag2 ping.yml

1. Create noshutdown.yml that start httpd service based on all nodes.
   1. shutdown.yml should stop httpd service on both node1 and node 4.
   2. It should not process node2 and node 3

1. Create a file saveabort.yml that
   1. Should create a file /etc/saveabort.txt with content "I am there".
   2. If this file exists, then it should not allow either overwriting the contents of the file or modifying the file.

**Answer: [the script should not modify if files already exists]**

a) Set the role path to /etc/ansible/roles and timeout to be 60

b) Create a requirement file role2install.yml for installing role named examfun from URL <http://classroom.example.com/content/examfun.tar.gz>

1. Create a file createuser.yml
   1. create users in staff variable in examfun role and makes them members of groups staff and web clients
   2. create users in guests in examfun role and makes them members of guests and web clients groups
   3. delete users in revoke in examfun role
2. Create unarchive.yml
   1. It will unzip http://classroom.example.com/content/vault.zip using a password protected file **(password=drone)** stored in encrypt.yml by executing command uzip -o -P {{password}} to /var/www/html
   2. Create encrypt.yml, which is encrypted with password **ansiansi** stored in .vaultpwd
3. create a playbook "failweb.yml" to Install ex407-server, start service ex407-server
4. you must show the output of installation and output of service command
5. if installation fails then you should provide message " Install Failed"
6. If Service fails then it should show that "service start failed"
7. if both fails it should print a message "playbook failed"