Install Ansible package using yum on ansible controller.

```
sudo yum install epel-release -y ; sudo yum install ansible -y
```

For debugging purposes, we would like to configure Ansible to write logs to a different path than default. Configure Ansible to write logs to /var/log/ansible/ansible.log.

log_path = /var/log/ansible/ansible.log

Update `/home/thor/playbooks/ansible.cfg` configuration file with new log path as given below:

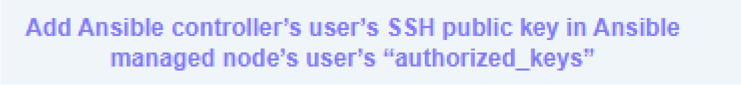
```
cat <<EOF  > /home/thor/playbooks/ansible.cfg
log_path = /var/log/ansible/ansible.log
EOF
```

What type of connection does Ansible establish to manage Unix based hosts ? ssh

Which command is used to generate an SSH key? Ssh-keygen

Among the following options, which file stores the SSH public keys of remote users that are allowed to establish an SSH connection to this host. - ~/.ssh/authorized_keys

We want to establish password-less SSH connectivity for a user from Ansible controller to web1 managed host, what exactly needs to be done. –



Add Ansible controller's user's SSH public key in Ansible managed node's user's "authorized_keys"

On Ansible controller node generate an SSH key with filename ansible under default location (~/.ssh). –

```
ssh-keygen -t rsa -f ~/.ssh/ansible
```

```
[thor@ansible-controller ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/thor/.ssh/id_rsa):
/home/thor/.ssh/ansible
Created directory '/home/thor/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/thor/.ssh/ansible.
Your public key has been saved in /home/thor/.ssh/ansible.pub.
The key fingerprint is:
SHA256:nmOckU8tfhSh2clS4sOmYA4NdOAdItZgIObPkIDmSc0 thor@ansible-controller
```

We would like to establish password-less secure authentication between Ansible controller and web1 node. Use the keys generated in the previous step and do the needful.
User ansible's SSH password for web1 node is ansible and during testing the SSH connection use -i <path-to-your-ssh-private-key> with ssh command.

To add controller key to node .

```
ssh-copy-id -i /home/thor/.ssh/ansible ansible@web1
```

To login to node . specify path for private key .

```
ssh -i /home/thor/.ssh/ansible ansible@web1
```

An inventory file is given at /home/thor/playbooks/inventory. Configure it to use the private ssh key.
Use the ping module to test connectivity through Ansible - ansible -m ping -i inventory web1

```
[thor@ansible-controller playbooks]$ cat inventory
web1 ansible_host=172.20.1.100 ansible_user=ansible
ansible_ssh_private_key_file=/home/thor/.ssh/ansible

[thor@ansible-controller playbooks]$ ansible -m ping -i inventory web1
web1 | SUCCESS => {
   "changed": false,
   "ping": "pong"
}
```

Run the Ansible command to display the version and save the output in /tmp/ansible_version.txt file.

```
ansible --version > /tmp/ansible_version.txt
```

Run the ansible command to gather facts of the localhost and save the output in /tmp/ansible_facts.txt file.

```
ansible -m setup localhost > /tmp/ansible_facts.txt
```

Execute an ad-hoc command to perform a ping connectivity test for the localhost and save the output in /tmp/ansible_ping.txt file.

```
ansible -m ping localhost > /tmp/ansible_ping.txt
```

Run an adhoc command to perform a ping connectivity to all hosts in the /home/thor/playbooks/inventory file and save the output in /tmp/ansible_all.txt file.

```
ansible -m ping -i inventory all > /tmp/ansible_all.txt
```

Run an adhoc command to run a command on host web1 to print the date and save the output in /tmp/ansible_date.txt file on the ansible controller.

Use the command module and argument date. Inventory file is available at /home/thor/playbooks/.

```
ansible -m 'command date' -i inventory web1 > /tmp/ansible_date.txt
```

Create a shell script called host_details.sh under ~/playbooks/ directory and make it executable.

The shell script should run ad-hoc ansible commands to:

Print the hostname of all managed nodes in the inventory file ~/playbooks/inventory

Using copy module copy the /etc/resolv.conf file from ansible controller to /tmp/resolv.conf on node00 host. Use the inventory file ~/playbooks/inventory.

```
[thor@ansible-controller playbooks]$ cat host_details.sh
ansible -a 'hostname' -i inventory all
ansible -m copy -a 'src=/etc/resolv.conf dest=/tmp/resolv.conf' -i inventory node00

[thor@ansible-controller playbooks]$ chmod +x host_details.sh
```

Create a shell script called host_data.sh under ~/playbooks/ directory and make it executable.

The shell script should:

Set ANSIBLE_GATHERING to False

Run an ad-hoc command to print the uptime of all managed nodes in the inventory file ~/playbooks/inventory

Create and run a playbook ~/playbooks/playbook.yml to cat the file /etc/redhat-release on all managed nodes in the inventory file ~/playbooks/inventory. Also please make sure to run this playbook in verbose mode i.e just add -vv at the end of your ansible-playbook command.

```
[thor@ansible-controller playbooks]$ cat playbook.yml
---
- name: 'Print file'
  hosts: all
  tasks:
  - name: 'Print'
    shell: cat /etc/redhat-release

[thor@ansible-controller playbooks]$ cat host_data.sh
export ANSIBLE_GATHERING=False
ansible -a 'uptime' -i inventory all
ansible-playbook -i inventory -vv playbook.yml
```