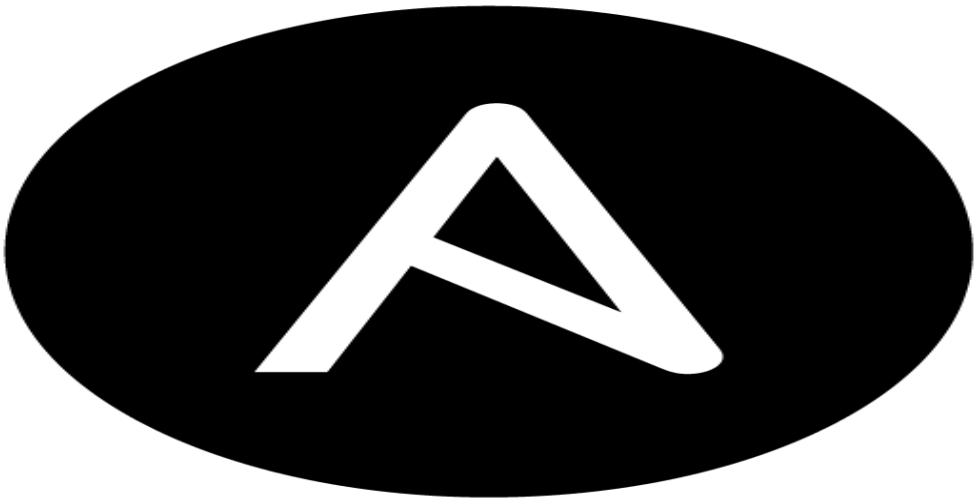
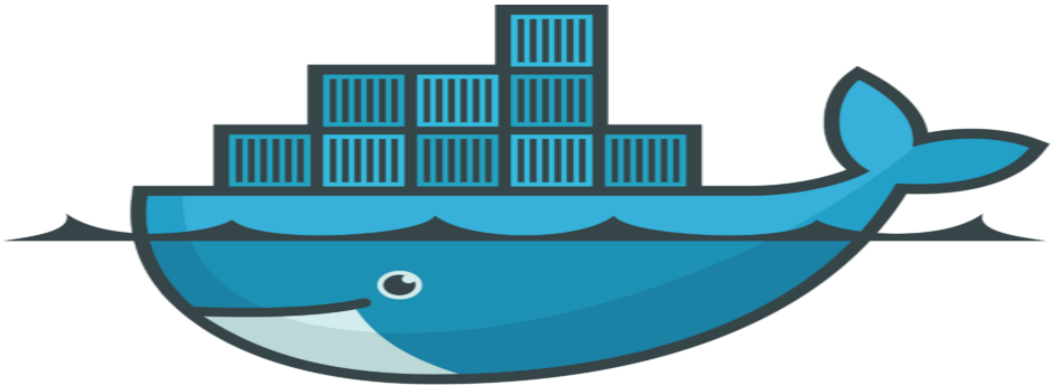


ANSIBLE



docker

kubernetes

TASK 1:-

Write an Ansible PlayBook that does the following operations in the managed nodes:

- Configure Docker
- Start and enable Docker services
- Pull the httpd server image from the Docker Hub
- Run the httpd container and expose it to the public
- Copy the html code in /var/www/html directory and start the web server

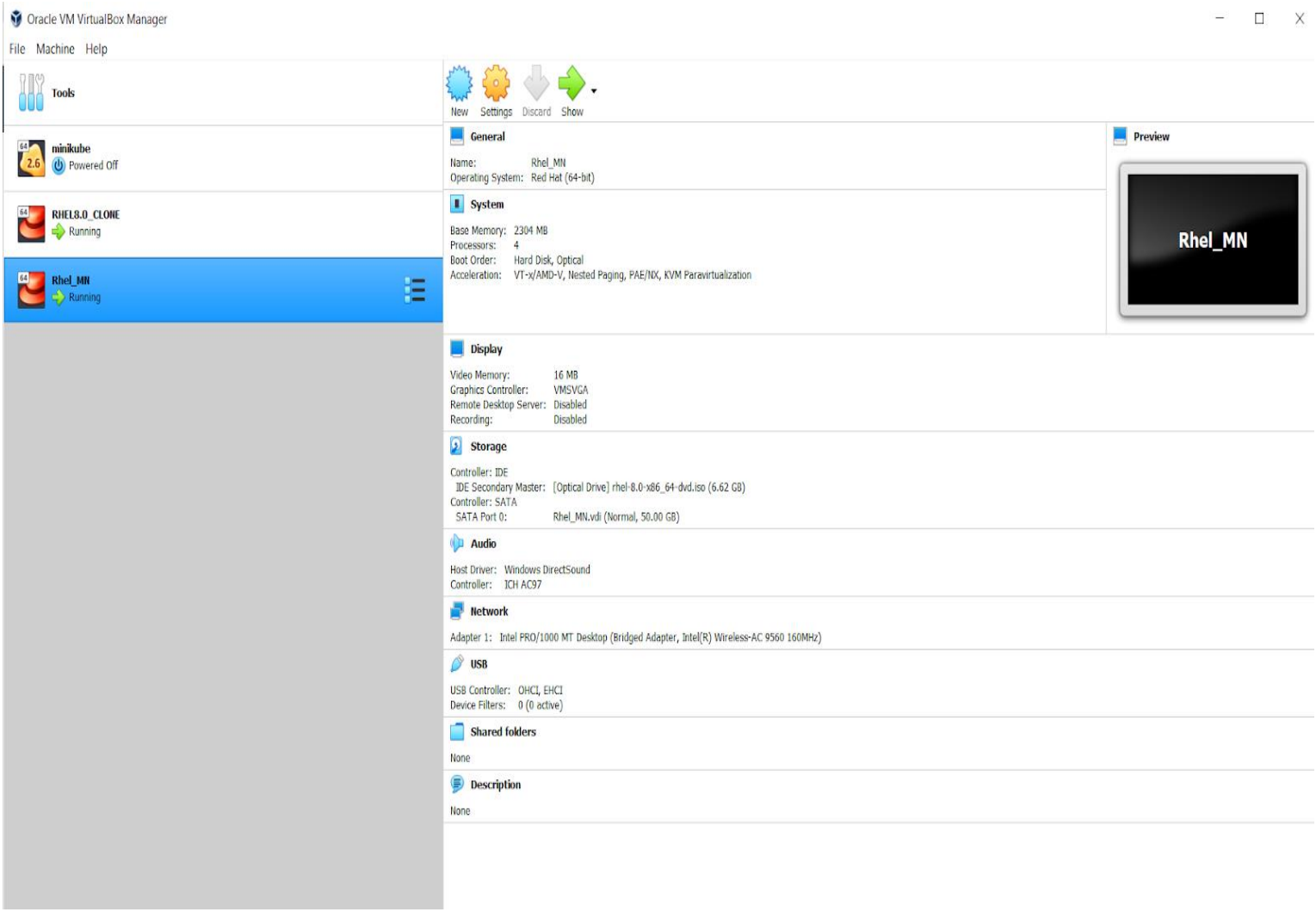
Important Notes:-

1. Ansible is so powerful and strong tool. It automatically come to know about the OS because they internally call the commands of that OS and install the software for us.
2. For configuration of any server we have two ways :
Manual : here we do each and everything manually.
Automation : we can achieve automation in two ways -
• Traditional way : here we have to write perl or shell scrips , they work on imprerative language. Here we have to know what to do and how to do both.
• Intelligent way : they work on declarative language but here we have to only know what to do. Tell your code I want that particular software to be installed and behind the scene they will know how to do.
3. Installing ansible on the top rhel 8 : "we use yum install ansible" but by default rhel 8 dvd does not provide ansible software. But we know developer big ansible use python as language for ansible. And we also know pip is a command to install python software or libraries or packages. So instead of yum you can use "pip3 install ansible" for Installing ansible on top of rhel8.
4. Why we use pip3 ..? Because we know two versions of python , python2 and python3 and rhel8 by default give 3 version of Python.
5.controller node : when we write and run the ansible code in the system .. that system is known as or work as a controller node.
6. Managed node : system whom you have to managed by ansible is known as managed node.
7. Inventory : it is basically a database of managed node IPs which you have to tell to your controller node.
Note :
*** Ansible is so powerful and strong tool. It automatically come to know about the OS because they internally call the commands of that OS and install the software for us.**
*** Ansible by defualt use SSH protocol for connectivity.**
*** Ansible don't know how to install software in any of OS but ansible know how to call the internal command.**
*** ansible all --list-hosts**
Above show us about nodes or inventory.
8. For creating an inventory we have to create one file.. where we have to write the ip of managed nodes, ansible_ssh_user ansible_ssh_pass etc
9. Config file for ansible : pip command does not create config file for us . So we have to create the config file :
*** Create directory " mkdir /etc/ansible".**
*** In this directory create ansible.cfg file.**
*** Then ansible automatically retrieve this file and using this file ansible do everything.**
Note :
*** Tell ansible about the database or inventory in a config file. Write the path of your database or inventory within the [defaults] header.**
*** Ansible is a tool work in a push mechanism.**
10. Push mechanism : when controller node go towards the manage node and install the package that process is come under push mechanism. And we also don't required to install any ansible software at the managed nodes. So ansible is agentless.
11. Pull mechanism : puppet and chef works on pull mechanism. Here manage node go towards the controller node and pull the package and then install. But here we have to install puppet and chef respective software at the managed node also.and in this scenario puppet and chef is considered as agent.

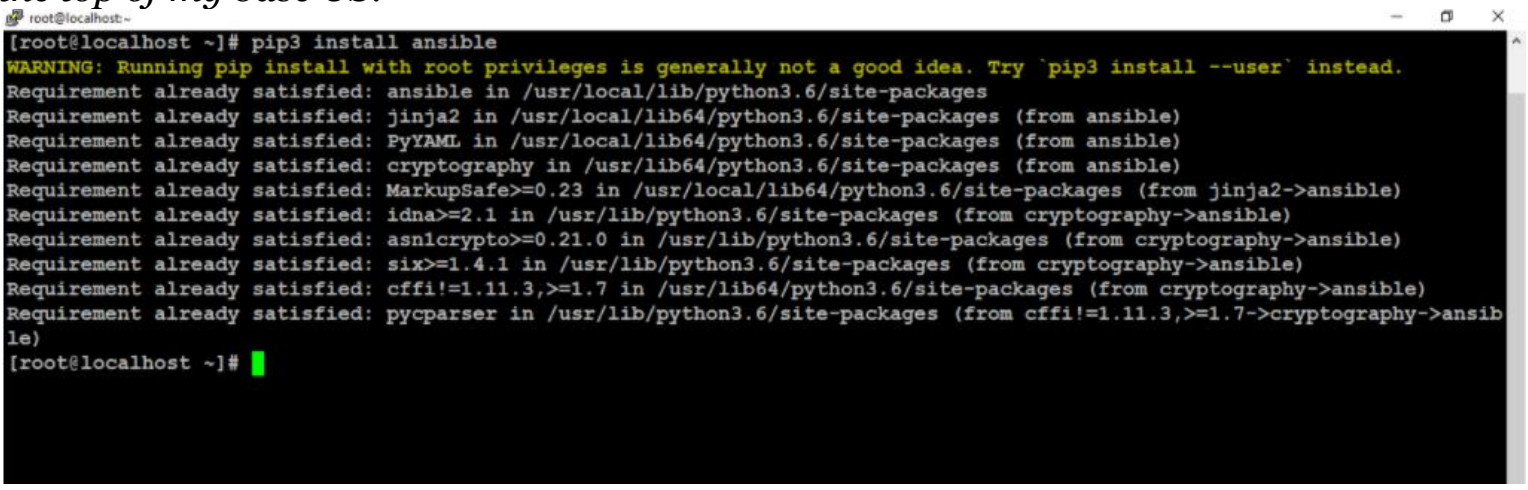
13. ansible automatically check the current state if the software is not installed then it go to the desired state to install the package this concept is called as idempotance.
14. For doing ssh between one VM and other VM you have to install sshpass software using yum install sshpass.
15. Modules : ansible core power is because of modulus. Ansible command is not intelligent,ansible intelligency comes from ansible module. in ansible if you manage the software or packages ansible has module available named as package module, if you if you want to copy one file from one system to other system ansible has module available named as copy and if you want to start,stop,reload,restart the program ansible has module available named as service.
16. For configuring the web server we have to know the following steps :
*** first step : we have to know which software has to be installed for a web server such as nginx, iis, httpd.**
*** Second step : we have to copy the web pages from one system to other.**
*** Third step : we have to start the services or program.**
17. We tell ansible to do something in two ways :
*** Commands : when we write ansible command on the terminal such as ansible all -m package -a "name=firefox state=present".**
This command is basically known as ad-hoc commands.
*** Program : In a program basically we have to write a script and that script is known as a program file but in ansible world this program file is known as a playbook. And the language that ansible used to create the playbook is known as yaml to format the code and we all know that it is a declarative language one file we can write the multiple play that is why it is known as a play book.for running and playbook we have a following command.**
ansile-playbook web.yml
Here web.yml is my program file basically known as ansible-playbook.
Note :
*** Inside the playbook we have to write the hosts which is talking about the inventory and we have to also write the tasks that tell what we have to do.**
*** Ansible always try to sync with what you need they also check the content not only look the file name.**
18. Ansible gives a very simple way that we can learn only one command. Because ansible create modules for us. they hide the command or resources of different operating system under one layer known as resource abstraction layer.
19. Remote copy are of two types :
*** Copy module - it works for static file and it never check the variable inside the file.**
*** Template module - it works for dynamic file and it always check the variable inside the file.**
20. We can easily give names of the tasks by name keyword
21. For managing the commands we have one module which is also known as command module. We can use it as :
tasks:
- command : pip3 install docker
22. We can take input from the user using prompt i.e var_prompt keyword. It will secure our value we can also set the prompt name by default private value is yes but we can set it as no by private keyword.
23. Two RHEL OS on your system with yum configuration.

- **Controller Node**
- **Managed Node**

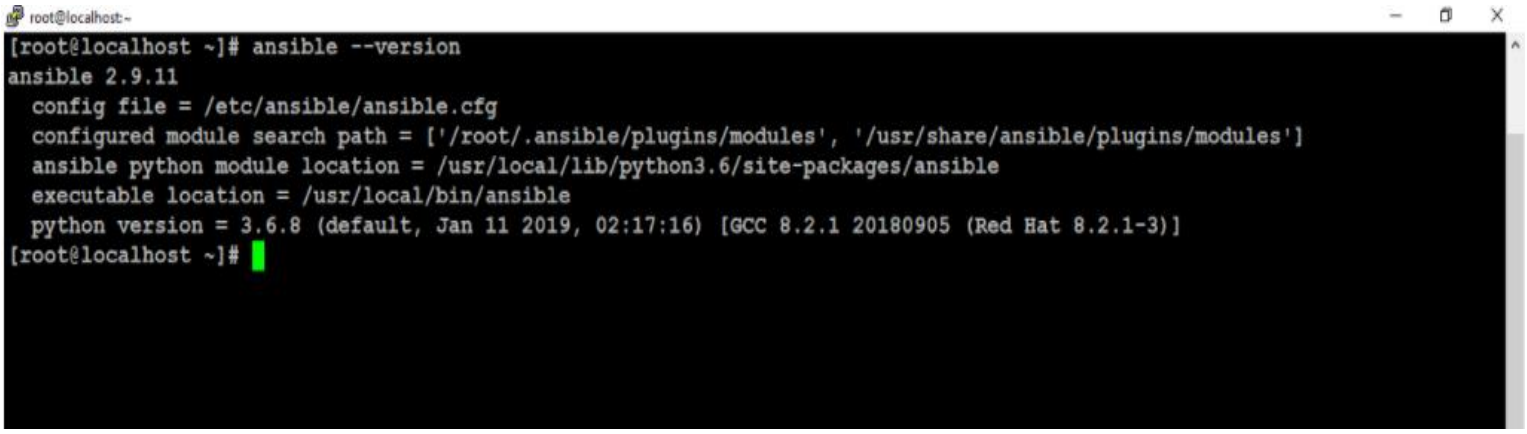
Screenshot:-



I am using oracle virtual box for deploying my controller node and managed node on the top of my base OS.

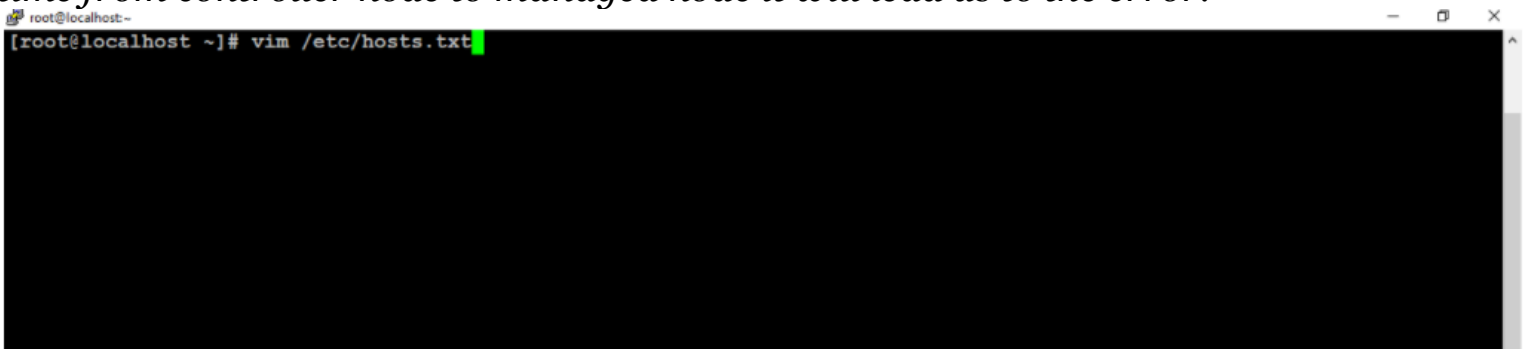


Installing the ansible in the controller node we have to use one command : "pip3 install ansible"



for checking the version of ansible you have to use one command : " ansible -- version"

Here you can see the config file of ansible where i mentioned the location of my inventory and i have written host_key_checking=false because when we do ssh first time from controller node to managed node it will lead us to the error.



```
[dockerhost]
192.168.43.219
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/hosts.txt" 3L, 74C                                      2,13          All
```

This is my inventory file where we can mention the details about managed or target node IP, username and password and also group the host as a dockerhost.

Checking whether docker setup is present in the managed node or not ...?

```
[root@localhost ~]# cd /etc/yum.repos.d/
[root@localhost yum.repos.d]# ls
redhat.repo  yum.repo
[root@localhost yum.repos.d]# _
```

Here you can see yum is not configured for docker that is there is no docker repo and i have only two repository : yum.repo is for my dvd and redhat.repo is already created when i have installed RHEL8 on the top of Virtualbox.

```
[root@localhost ~]# rpm -q docker-ce
package docker-ce is not installed
[root@localhost ~]# _
```

Confirming docker is installed in your system by using one command that is ' rpm -q docker-ce ' .

Now i am creating one playbook script for deploying docker setup on the top of managed node.

```
root@localhost:~
[root@localhost ~]# vim docker.yml


[root@localhost ~]# cd /etc/yum.repos.d/
[root@localhost yum.repos.d]# ls
docker.repo        epel-playground.repo  epel-testing-modular.repo  redhat.repo
epel-modular.repo  epel.repo             epel-testing.repo          yum.repo
[root@localhost yum.repos.d]# cat docker.repo
[docker]
baseurl = https://download.docker.com/linux/centos/7/x86_64/stable/
gpgcheck = 0
name = docker yum repo

[root@localhost yum.repos.d]#
```

```
[root@localhost html]# mkdir /dvd
mkdir: cannot create directory '/dvd': File exists
[root@localhost html]# cd /dvd
[root@localhost dvd]# ls
AppStream  EFI   extra_files.json  images     media.repo               RPM-GPG-KEY-redhat-release
BaseOS     EULA  GPL              isolinux   RPM-GPG-KEY-redhat-beta  TRANS.TBL
[root@localhost dvd]# yum repolist
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manager t
o register.
Repository docker-ce-edge is listed more than once in the configuration
Repository docker-ce-edge-debuginfo is listed more than once in the configuration
Repository docker-ce-edge-source is listed more than once in the configuration
Repository docker-ce-nightly is listed more than once in the configuration
Repository docker-ce-nightly-debuginfo is listed more than once in the configuration
Repository docker-ce-nightly-source is listed more than once in the configuration
Repository docker-ce-stable is listed more than once in the configuration
Repository docker-ce-stable-debuginfo is listed more than once in the configuration
Repository docker-ce-stable-source is listed more than once in the configuration
Repository docker-ce-test is listed more than once in the configuration
Repository docker-ce-test-debuginfo is listed more than once in the configuration
Repository docker-ce-test-source is listed more than once in the configuration
Last metadata expiration check: 0:55:10 ago on Sat 01 Aug 2020 03:30:10 AM IST.
repo id                                 repo name                                                status
AppStream                               Local_appstream                                          4,672
BaseOS                                  Local_baseos                                             1,658
docker-ce-stable                        Docker CE Stable - x86_64                                   71
[root@localhost dvd]# _
```

```
[root@localhost yum.repos.d]# rpm -q docker-ce
docker-ce-18.09.1-3.el7.x86_64
[root@localhost yum.repos.d]# rpm -q docker-py
package docker-py is not installed
[root@localhost yum.repos.d]# rpm -q docker
package docker is not installed
[root@localhost yum.repos.d]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2020-08-01 01:41:55 IST; 2h 46min ago
     Docs: https://docs.docker.com
 Main PID: 19795 (dockerd)
    Tasks: 36
   Memory: 180.1M
   CGroup: /system.slice/docker.service
           ├─19638 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 6379 -container-ip >
           ├─19643 containerd-shim -namespace moby -workdir /var/lib/docker/containerd/daemon/io.co>
           ├─19795 /usr/bin/dockerd -H fd://
           └─19812 containerd --config /var/run/docker/containerd/containerd.toml --log-level info

Aug 01 04:02:09 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:09.442663634+05:30" le>
Aug 01 04:02:09 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:09.452571952+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.669146341+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.827848638+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.842182658+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.845291112+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.978294017+05:30" le>
Aug 01 04:04:03 localhost.localdomain dockerd[19795]: time="2020-08-01T04:04:03.570485355+05:30" le>
Aug 01 04:27:32 localhost.localdomain dockerd[19795]: time="2020-08-01T04:27:32.092990270+05:30" le>
Aug 01 04:27:32 localhost.localdomain dockerd[19795]: time="2020-08-01T04:27:32.103602041+05:30" le>
lines 1-23/23 (END)
```

```
[root@localhost /]# cd /etc/yum.repos.d
[root@localhost yum.repos.d]# ls
docer-ce.repo  docker-ce.repo  dvd1.repo  dvd2.repo  redhat.repo
[root@localhost yum.repos.d]# cat dvd1.repo
[BaseOS]
baseurl = file:///dvd/BaseOS
gpgcheck = 0
name = Local_baseos

[root@localhost yum.repos.d]# cat dvd2.repo
[AppStream]
baseurl = file:///dvd/AppStream
gpgcheck = 0
name = Local_appstream

[root@localhost yum.repos.d]# _
```

```
[root@localhost html]# mkdir /dvd
mkdir: cannot create directory ‘/dvd’: File exists
[root@localhost html]# cd /dvd
[root@localhost dvd]# ls
AppStream  EFI   extra_files.json  images    media.repo              RPM-GPG-KEY-redhat-release
BaseOS     EULA  GPL               isolinux  RPM-GPG-KEY-redhat-beta  TRANS.TBL
[root@localhost dvd]# yum repolist
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manager t
o register.
Repository docker-ce-edge is listed more than once in the configuration
Repository docker-ce-edge-debuginfo is listed more than once in the configuration
Repository docker-ce-edge-source is listed more than once in the configuration
Repository docker-ce-nightly is listed more than once in the configuration
Repository docker-ce-nightly-debuginfo is listed more than once in the configuration
Repository docker-ce-nightly-source is listed more than once in the configuration
Repository docker-ce-stable is listed more than once in the configuration
Repository docker-ce-stable-debuginfo is listed more than once in the configuration
Repository docker-ce-stable-source is listed more than once in the configuration
Repository docker-ce-test is listed more than once in the configuration
Repository docker-ce-test-debuginfo is listed more than once in the configuration
Repository docker-ce-test-source is listed more than once in the configuration
Last metadata expiration check: 0:55:10 ago on Sat 01 Aug 2020 03:30:10 AM IST.
repo id                                repo name                                              status
AppStream                              Local_appstream                                        4,672
BaseOS                                 Local_baseos                                           1,658
docker-ce-stable                       Docker CE Stable - x86_64                                 71
[root@localhost dvd]# _
```

```
[root@localhost yum.repos.d]# rpm -q docker-ce
docker-ce-18.09.1-3.el7.x86_64
[root@localhost yum.repos.d]# rpm -q docker-py
package docker-py is not installed
[root@localhost yum.repos.d]# rpm -q docker
package docker is not installed
[root@localhost yum.repos.d]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2020-08-01 01:41:55 IST; 2h 46min ago
     Docs: https://docs.docker.com
 Main PID: 19795 (dockerd)
    Tasks: 36
   Memory: 180.1M
   CGroup: /system.slice/docker.service
           ├─19638 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 6379 -container-ip >
           ├─19643 containerd-shim -namespace moby -workdir /var/lib/docker/containerd/daemon/io.co>
           ├─19795 /usr/bin/dockerd -H fd://
           └─19812 containerd --config /var/run/docker/containerd/containerd.toml --log-level info

Aug 01 04:02:09 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:09.442663634+05:30" le>
Aug 01 04:02:09 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:09.452571952+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.669146341+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.827848638+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.842182658+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.845291112+05:30" le>
Aug 01 04:02:10 localhost.localdomain dockerd[19795]: time="2020-08-01T04:02:10.978294017+05:30" le>
Aug 01 04:04:03 localhost.localdomain dockerd[19795]: time="2020-08-01T04:04:03.570485355+05:30" le>
Aug 01 04:27:32 localhost.localdomain dockerd[19795]: time="2020-08-01T04:27:32.092990270+05:30" le>
Aug 01 04:27:32 localhost.localdomain dockerd[19795]: time="2020-08-01T04:27:32.103602041+05:30" le>
lines 1-23/23 (END)
```

```
[root@localhost yum.repos.d]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
httpd               latest              9d2a0c6e5b57        9 days ago          166MB
[root@localhost yum.repos.d]# docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED             STATUS
 PORTS                    NAMES
b256a2abcdbe        httpd               "httpd-foreground"   24 minutes ago      Up 24 minutes
 0.0.0.0:6379->80/tcp   myhttpd
[root@localhost yum.repos.d]# docker exec -it myhttpd bash
root@b256a2abcdbe:/usr/local/apache2# cd /usr/local/apache2/htdocs/
root@b256a2abcdbe:/usr/local/apache2/htdocs# ls
home.html
```

-Configuring yum

- *Creation of a folder and mounting the cdrom*
- *Creation of a repo dvd.repo and dvd1.repo*

```
- name: task_1
  gather_facts: No hosts:
  server1 tasks:
  - name: create_dvd_folder file:
      path: /dvd state:
      directory mode:
      "0755"
  - name: mount_dvd
    mount:
      path: /dvd/
      src: /dev/cdrom fstype:
      iso9660 opts: ro,loop
      state: mounted
  - name: yum_repo_BaseOS
    yum_repository:
      name: BaseOS description:
      Local_baseos file: dvd1
      baseurl: file:///dvd/BaseOS
      gpgcheck: no
   - name: yum_repo_AppStream
      yum_repository:
        name: AppStream description:
        Local_appstream


        file: dvd2
        baseurl: file:///dvd/AppStream
        gpgcheck: no
```

Removing unwanted package precreated if any

```
  - name: Remove docker if installed from CentOS repo yum:
      name:
      - docker
      - docker-client
      - docker-client-latest
      - docker-common
      - docker-latest
      - docker-latest-logrotate
      - docker-logrotate
```

- docker-engine state:
removed

- ***Installing dependencies***

- name: Install yum utils

yum:
name: yum-utils state:
latest

- name: Install device-mapper-persistent-data yum:
name: device-mapper-persistent-data state:
latest

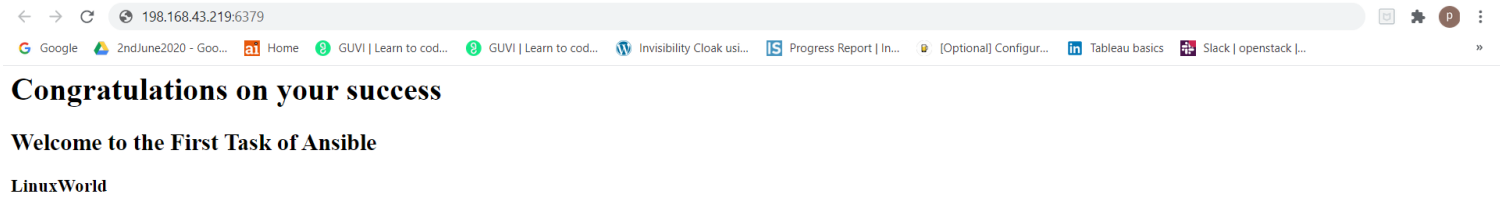
- name: Install lvm2 yum:
name: lvm2
state: latest

Configure a docker-ce.repo and installing docker using shell as well as using ansible

```
- name: Add Docker repo get_url:
    url: https://download.docker.com/linux/centos/docker-ce.repo dest:
    /etc/yum.repos.d/docker-ce.repo
  become: yes

 - name: Enable Docker Edge repo ini_file:
    dest: /etc/yum.repos.d/docker-ce.repo section:
    'docker-ce-edge'
    option: enabled value: 0
  become: yes

 - name: Enable Docker Test repo ini_file:
    dest: /etc/yum.repos.d/docker-ce.repo section:
    'docker-ce-test'
    option: enabled value: 0
  become: yes
 - name: Install Docker
  shell: "dnf install --nobest docker-ce -y"

    Or
 - name: Install Docker package:
    name: docker-ce state: latest
  become: yes

  -  Starting docker service and installation req packages
- name: Start Docker service service:
    name: docker state: started
  enabled: yes become: yes

 - pip3:
    name:  "docker-py"  executable:
    pip2.7 state: present
Or

 - name: docker-py
  shell: "pip3 install docker"
```

Pulling the docker image and creating the container starting the services ,mounting the data and exporting the port

```
- name: pull an image
    docker_image:
      name: httpd source: pull
 - name: creating a httpd container
    docker_container:
      name: myhttpd image: httpd
      state: started published_ports:
      - "6379:80"
      volumes:
      - "/var/www/html/:/usr/local/apache2/htdocs/:ro"
```



Practical Work:-using kubernetes.

Configure yum for docker :

```
- hosts: all
  tasks:
    - name: docker repo
      yum_repository:
        name: docker
        baseurl: https://download.docker.com/linux/centos/7/x86_64/stable/
        description: my docker repo

        gpgcheck: 0
```

Configure yum for k8s :

```
 - name: kube repo
      yum_repository:
        name: kube
        baseurl: https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
        description: my kube repo

        gpgcheck: 0
```

Installing k8s Commands :

```
 - name: installing kubeadm,kubectl and kubelet
      package:
        name:
         - "kubeadm"
         - "kubectl"
         - "kubelet"

        state: present
```

Set Se-Linux to permissive mode :

```
- name: permissive mode of se-linux
      selinux:
        state: permissive

        policy: targeted
```

Installing Docker and Start Docker services :

```
- name: installing docker
      package:
         name: "docker-ce-3:18.09.1-3.el7"
         state: present


    - name: starting and enabling services of docker
      service:
        name: "docker"
        state: started
```

```
    enabled: yes
```

Copying daemon.json file :

```
- name: copying daemon.json file
  copy:

  src: daemon.json
```

Enabling the system reload :

```
- name: for enabling system reload
  systemd:

  daemon_reload: yes
```

Restarting the docker services and disabling the swap :

```
- name: restarting docker services
  service:
    name: "docker"
    state: restarted


- name: disabling the swap
  command: swapoff -a


- name: for disabling the swap permanently
  replace:
    path: /etc/fstab
    regexp: '^([^#].*?\sswap\s+sw\s+.*)$'
    replace: '#\1'
```

Installing Iproute-tc :

```
- name: installing iproute-tc
  package:
    name: "iproute-tc"
    state: present
```

Starting and enabling the kubelet program :

```
- name: starting and enabling kubelet program
  service:
    name: "kubelet"
    state: started
    enabled: yes
```

Initializing kubeadm and provide range of IPs :

```
- hosts: masternode
  tasks:
    - name: initialising kubeadm and providing a range of Ips
      shell: kubeadm init --v=5 --pod-network-cidr=10.10.1.0/16
     ignore_errors: yes
```

Creating .kube directory , copying admin.conf to .kube/config and change the user.

```
 - name: creating .kube directory
       become: yes
       file:
        path: $HOME/.kube
        state: directory
        mode: 0755


    - name: copying
      shell: cp -i /etc/kubernetes/admin.conf $HOME/.kube/config


    - name: changing the user
      shell: chown $(id -u):$(id -g) $HOME/.kube/config
```

Installing Flannel Plugin i.e Overlay Network in the master node:

```
root@localhost ~]# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documen
ation/kube-flannel.yml
odsecuritypolicy.policy/psp.flannel.unprivileged created
lusterrole.rbac.authorization.k8s.io/flannel created
lusterrolebinding.rbac.authorization.k8s.io/flannel created
erviceaccount/flannel created
onfigmap/kube-flannel-cfg created
aemonset.apps/kube-flannel-ds-amd64 created
aemonset.apps/kube-flannel-ds-arm64 created
aemonset.apps/kube-flannel-ds-arm created
aemonset.apps/kube-flannel-ds-ppc64le created
aemonset.apps/kube-flannel-ds-s390x created
root@localhost ~]# _
```

Now my intention is to launch one pod using ansible playbook and for this i have created one pod.yml file

```
 - hosts : masternode
   task:
    - name: creating pod
      command: kubectl run mypod -n kube-system --image=vimal13/apache-webserver-php
```

```
NAME      STATUS    ROLES     AGE     VERSION
master    Ready     master    28m     v1.18.4
slave1    Ready     <none>    2m2s    v1.18.4
```

```
[root@localhost ~]# kubectl get ns
NAME              STATUS   AGE
default           Active   104m
kube-node-lease   Active   104m
kube-public       Active   104m
kube-system       Active   104m
[root@localhost ~]#
```

root@localhost:~

```
[root@localhost ~]# kubectl get pods -n kube-system
NAME                                          READY   STATUS    RESTARTS   AGE
coredns-66bff467f8-ld9zz                      1/1     Running   1          134m
coredns-66bff467f8-mgmph                      1/1     Running   1          134m
etcd-localhost.localdomain                    1/1     Running   2          134m
kube-apiserver-localhost.localdomain          1/1     Running   3          134m
kube-controller-manager-localhost.localdomain 1/1     Running   12         134m
kube-flannel-ds-amd64-2752b                   1/1     Running   2          60m
kube-proxy-pbxjs                              1/1     Running   1          134m
kube-scheduler-localhost.localdomain          1/1     Running   12         134m
mypod                                         0/1     Pending   0          2m50s
[root@localhost ~]#
```

Thanks for reading!!!

-