Linux Directory Structure (File System Structure) Explained with Examples

**RBAC(ROLE BASED Access Control)=>LDAP(Username and Password, users and groups)**

Have you wondered why certain programs are located under /bin, or /sbin, or /usr/bin, or /usr/sbin?

For example, less command is located under /usr/bin directory. Why not /bin, or /sbin, or /usr/sbin? What is the different between all these directories?

In this article, let us review the Linux filesystem structures and understand the meaning of individual high-level directories.



1. / – Root

* Every single file and directory starts from the root directory.
* Only root user has write privilege under this directory.
* Please note that /root is root user’s home directory, which is not same as /.

2. /bin – User Binaries

* Contains binary executables.
* Common Linux commands you need to use in single-user modes are located under this directory.
* Commands used by all the users of the system are located here.
* For example: ps, ls, ping, grep, cp.

3. /sbin – System Binaries

* Just like /bin, /sbin also contains binary executables.
* But, the Linux commands located under this directory are used typically by system administrator, for system maintenance purpose.
* For example: iptables, reboot, fdisk, ifconfig, swapon

4. /etc – Configuration Files

* Contains configuration files required by all programs.
* This also contains startup and shutdown shell scripts used to start/stop individual programs.
* For example: /etc/resolv.conf, /etc/logrotate.conf

5. /dev – Device Files

* Contains device files.
* These include terminal devices, usb, or any device attached to the system.
* For example: /dev/tty1, /dev/usbmon0

6. /proc – Process Information

* Contains information about system process.
* This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
* This is a virtual filesystem with text information about system resources. For example: /proc/uptime

7. /var – Variable Files

* var stands for variable files.
* Content of the files that are expected to grow can be found under this directory.
* This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

8. /tmp – Temporary Files

* Directory that contains temporary files created by system and users.
* Files under this directory are deleted when system is rebooted.

9. /usr – User Programs

* Contains binaries, libraries, documentation, and source-code for second level programs.
* /usr/bin contains binary files for user programs. If you can’t find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
* /usr/sbin contains binary files for system administrators. If you can’t find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
* /usr/lib contains libraries for /usr/bin and /usr/sbin
* /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

10. /home – Home Directories

* Home directories for all users to store their personal files.
* For example: /home/john, /home/nikita

11. /boot – Boot Loader Files

* Contains boot loader related files.
* Kernel initrd, vmlinux, grub files are located under /boot
* For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

12. /lib – System Libraries

* Contains library files that supports the binaries located under /bin and /sbin
* Library filenames are either ld\* or lib\*.so.\*
* For example: ld-2.11.1.so, libncurses.so.5.7

13. /opt – Optional add-on Applications

* opt stands for optional.
* Contains add-on applications from individual vendors.
* add-on applications should be installed under either /opt/ or /opt/ sub-directory.

14. /mnt – Mount Directory

* Temporary mount directory where sysadmins can mount filesystems.

15. /media – Removable Media Devices

* Temporary mount directory for removable devices.
* For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer

16. /srv – Service Data

* srv stands for service.
* Contains server specific services related data.
* For example, /srv/cvs contains CVS related data.

**At** => command is for single Schedule

**yum install dnf -y =>** This is a package installer tool

**yum install tree -y =>** To install Tree structures.

cat command;

cd..;

How do I determine RHEL version?

To determine RHEL version, type: **cat /etc/redhat-release**

Execute command to find RHEL version: more /etc/issue

Show RHEL version using command line, rune:  **less /etc/os-release**

RHEL 7.x or above user can use the **hostnamectl** command to get RHEL version

To display the RHEL kernel version, type:

**uname -r**

**uname -mrs**

How to confirm RedHat Enterprise Linux version using hostnamectl

**hostnamectl**

Other commands to find out RHEL version

One can use the rpm command:

rpm --query redhat-release-server

Another option is to run lsb\_release command. However, you need to install a package named redhat-lsb-core using the yum command as follows

**sudo yum install redhat-lsb-core**

## now get RHEL version ##

**lsb\_release -d**

**lsb\_release -a**

Ls Command in Linux (List Files and Directories)

**ls [OPTIONS] [FILES]**

**ls**

**ls /etc**

**ls /etc /var /etc/passwd**

**ls /root**

The default output of the ls command shows only the names of the files and directories, which is not very informative.

The -l ( lowercase L) option causes ls to print files in a long listing format.

When the long listing format is used, the ls command will display the following file information:

The file type

The file permissions

Number of hard links to the file

File owner

File group

File size

Date and Time

File name

Consider the following example:

**ls -l /etc/hosts**

To display all files including the hidden files use the -a option:

**ls -la ~/**

**ls -ltr /var**

ls -R

Create a User

**sudo useradd tomcatadm**

Swith to User

**sudo su tomcatadm**

this is only can Testing team person.no we will only Devops team to switch to this User,as its Our Group.

go to Present directory

**pwd**

go to Root/ home directory

cd /

ls -a

cd .ssh

ls

cat

view known\_hosts

to exit from edit or view mode

esc :q!

**drive =file system,**

**software =package**

**folder=directory**

**adding=mounting**

Schedule a job in Linux

**at**  => command for Single Schedule

**What Is a Cron Job? cron is a Linux utility which schedules a command or script on your server to**

**run automatically at a specified time and date.**

**A cron job is the scheduled task itself. Cron jobs can be very useful to automate repetitive tasks.**

media => is for Media Files (MP3,Movie files)

mnt => when ever plug in pendrive in the system.it will show up here .Mounting

opt => programs and files / User Programs

proc => process Id .it contain Process Details.

cat => is for display Data from file

**cat cpuinfo** => this will show CPU information.

cat cpuinfo | grep "cpu cores" wc -l

**cat cpuinfo | grep "cpu cores"**

**[root@ip-172-31-30-86 proc]# id**

**uid=0(root) gid=0(root) groups=0(root) context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023**

**[root@ip-172-31-30-86 proc]# id docker**

**uid=1002(docker) gid=1002(docker) groups=1002(docker)**

**[root@ip-172-31-30-86 proc]# cat /etc/passwd**

**root:x:0:0:root:/root:/bin/bash**

rpm=> rpm is to package installation cmd

wget ,curl = It is invoked HTTP and FTP protocol

**[root@ip-172-31-30-86 proc]# dnf install -y wget**

**[root@ip-172-31-30-86 proc]# dnf install -y curl**

**To install RPM File in Linux System.**

**[root@ip-172-31-30-86 opt]# rpm -ivh chef-15.4.45-1.el7.x86\_64.rpm**

**Whether it is installed or not {qa is to querry}**

**[root@ip-172-31-30-86 opt]# rpm -qa chef**

**chef-15.4.45-1.el7.x86\_64**

**this is an Uninstallation cmd**

**[root@ip-172-31-30-86 opt]# rpm -e chef**

**error: package chef is not installed**

**To get all packages info**

**[root@ip-172-31-30-86 opt]# rpm -qa**

**[root@ip-172-31-30-86 opt]# rpm -qa |wc -l**

**544**

GNU Wget is a computer program that retrieves content from web servers. It is part of the GNU Project.

Its name derives from World Wide Web and get. It supports downloading via HTTP,

HTTPS, and FTP. Its features include recursive download, conversion of links for offline viewing of local HTML,

and support for proxies.

cURL is a computer software project providing a library and command-line tool for transferring data using various protocols.

It was first released in 1997.

The name stands for "Client URL". The original author and lead developer is the Swedish developer Daniel Stenberg

**Switching one User to Another User**

**[ec2-user@ip-172-31-30-86 opt]$ sudo su - root**

**Last login: Sun Nov 3 15:19:24 UTC 2019 on pts/0**

**this cmd will help 1000 cmd history**

**[root@ip-172-31-30-86 opt]# history**

**this is going to install Wget**

**[root@ip-172-31-30-86 opt]# yum install wget**

**This is Apache Web Server**

**[root@ip-172-31-30-86 opt]# yum install httpd**

**Apache HTTP Server**

**[root@ip-172-31-30-86 ~]# dnf install -y httpd**

NGINX | High Performance Load Balancer, Web Server ...

**How to install Nginx on CentOS 7 or RHEL 7**

**sudo yum update  
sudo yum install nginx**

**yum repo Configure for Secure network**

[root@ip-172-31-30-86 opt]# **cd /etc/yum.repos.d/**

[root@ip-172-31-30-86 yum.repos.d]# ls

redhat-rhui-beta.repo.disabled redhat-rhui-client-config.repo redhat-rhui.repo

[root@ip-172-31-30-86 yum.repos.d]#

**yum or rpm : systemctl**

[root@ip-172-31-30-86 yum.repos.d]# **systemctl status httpd**

● httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)

Active: inactive (dead)

Docs: man:httpd.service(8)

[root@ip-172-31-30-86 yum.repos.d**]# systemctl start httpd**

[root@ip-172-31-30-86 yum.repos.d]# **systemctl status httpd**

● httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)

Active: active (running) since Sun 2020-01-19 10:25:56 UTC; 11s ago

Docs: man:httpd.service(8)

Main PID: 23403 (httpd)

Status: "Running, listening on: port 80"

Tasks: 213 (limit: 4998)

Memory: 29.6M

CGroup: /system.slice/httpd.service

├─23403 /usr/sbin/httpd -DFOREGROUND

├─23404 /usr/sbin/httpd -DFOREGROUND

├─23405 /usr/sbin/httpd -DFOREGROUND

├─23406 /usr/sbin/httpd -DFOREGROUND

└─23407 /usr/sbin/httpd -DFOREGROUND

Jan 19 10:25:56 ip-172-31-30-86.us-east-2.compute.internal systemd[1]: Starting The Apache HTTP Server...

Jan 19 10:25:56 ip-172-31-30-86.us-east-2.compute.internal systemd[1]: Started The Apache HTTP Server.

Jan 19 10:25:56 ip-172-31-30-86.us-east-2.compute.internal httpd[23403]: Server configured, listening on: port 80

**http://ec2-3-135-229-220.us-east-2.compute.amazonaws.com:80**

**systemctl stop httpd**

**systemctl restart httpd**

**ps -ef => cmd to check all running Process**

**ps -ef |grep "httpd"**

**downloading through Wget Topmcat**

[root@ip-172-31-30-86 opt]# wget https://www.apache.org/dist/tomcat/tomcat-9/v9.0.30/bin/apache-tomcat-9.0.30.tar.gz.sha512

**[root@ip-172-31-30-86 etc]# ls -lh**

**sudo dnf install gzip**

**Unziping the file with gunzip**

**[root@ip-172-31-30-86 opt]# gunzip apache-tomcat-9.0.27.tar.gz**

[gzip (Compress) syntax]

# gzip [Files To Be Compressed]

[gzip (Decompress) syntax]

# gunzip [FileName.gz]

[gzip (Decompress) syntax]

# gzip -d [FileName.gz]

1. Install tar on CentOS/RHEL, run: **sudo yum install tar**
2. Upgrade tar on CentOS/RHEL 6/7/8 Linux, run: **sudo yum update tar**

tar [options] file

### Extract a tar.gz archive

tar -xvzf wordpress.tar.gz

* **x** – Extract files.
* **v** – Verbose outputs.
* **z** – Work on compressed files
* **f** – Extract file named wordpress.tar.gz

tar -xvzf wordpress.tar.gz -C /tmp/

[root@ip-172-31-30-86 opt]# tar xf apache-tomcat-9.0.30.tar

[root@ip-172-31-30-86 opt]# sudo dnf install -y java

**>/dev/null nullify the display window**

**yum install -y java-1.8.0-openjdk >/dev/null**

**To Clear the Screen**

**[root@ip-172-31-30-86 bin]# clear**

**[root@ip-172-31-30-86 bin]# ./version.sh**

**[root@ip-172-31-30-86 bin]# ./startup.sh**

**by default http uses 80 port**

**and HTTPs Uses 443**

[**http://ec2-3-135-229-220.us-east-2.compute.amazonaws.com:8080/**](http://ec2-3-135-229-220.us-east-2.compute.amazonaws.com:8080/)

**[root@ip-172-31-30-86 bin]# ps -ef |grep "tomcat"**

**Killing the Process**

**[root@ip-172-31-30-86 bin]# kill -9 3423**

**[root@ip-172-31-30-86 bin]# ps -ef |grep "tomcat"**

**root 3469 1480 0 18:00 pts/0 00:00:00 grep --color=auto tomcat**

**[root@ip-172-31-30-86 bin]#**

This is an Envirment Variable

**[root@ip-172-31-30-86 ~]# env**

**Printing MAIL Envirment Variable**

**[root@ip-172-31-30-86 ~]# echo $MAIL**

**/var/spool/mail/root**

**To set Env Variable it can be access All through the Apps Lifecycles.**

[root@ip-172-31-30-86 ~]# export NAME="Utpal Maiti"

[root@ip-172-31-30-86 ~]# echo $NAME

Utpal Maiti

**To find out bash profile Where we can set env variable (Hidden)**

[root@ip-172-31-30-86 ~]# ls -a

**Bash Profile Editing**

**[root@ip-172-31-30-86 ~]# vim .bash\_profile**

**Esc>:q!<enter>**

**To reload a file Use . + filename**

**[root@ip-172-31-30-86 ~]# . .bash\_profile**

**To view path Variable**

**[root@ip-172-31-30-86 ~]# echo $PATH**

**/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/root/bin**

**Setting Up PATH Variable**

**[root@ip-172-31-30-86 bin]# export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/root/bin:/opt/apache-tomcat-9.0.27/bin**

**Mounting Devices ,It will add on Boottime**

**[root@ip-172-31-30-86 etc]# cat fstab**

[root@ip-172-31-30-86 ~]# cat /etc/fstab

#

# /etc/fstab

# Created by anaconda on Tue Jun 18 17:03:37 2019

#

# Accessible filesystems, by reference, are maintained under '/dev/disk/'.

# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.

#

# After editing this file, run 'systemctl daemon-reload' to update systemd

# units generated from this file.

#

UUID=a727b695-0c21-404a-b42b-3075c8deb6ab / xfs defaults 0 0

[root@ip-172-31-30-86 ~]#

**[root@ip-172-31-30-86 etc]# cat hosts**

To display HostName

**[root@ip-172-31-30-86 etc]# cat hostname**

**ip-172-31-30-86.us-east-2.compute.internal**

It will display All Group Names

**[root@ip-172-31-30-86 etc]# view group**

**To Bootstrap OS Message Place in ,motd file .then connect new session**

**[root@ip-172-31-30-86 etc]# vi motd**

**It gives the Kernnel Related Info.**

**[root@ip-172-31-30-86 etc]# uname -a**

Linux ip-172-31-30-86.us-east-2.compute.internal 4.18.0-80.4.2.el8\_0.x86\_64 #1 SMP Fri Jun 14 13:20:24 UTC 2019 x86\_64 x86\_64 x86\_64 GNU/Linux

**[root@ip-172-31-30-86 etc]# cat os-relese**

To view all User Password

**[root@ip-172-31-30-86 etc]# view passwd**

To view Local User Password

**[root@ip-172-31-30-86 etc]# view shadow**

**To change Local User Password**

[root@ip-172-31-30-86 etc]# **passwd jenkins**

Changing password for user jenkins.

New password:**jenkins**

BAD PASSWORD: The password is shorter than 8 characters

Retype new password:**jenkins**

passwd: all authentication tokens updated successfully.

[root@ip-172-31-30-86 etc]# **view shadow**

[root@ip-172-31-30-86 etc]# **passwd** **tomcatadmUtpal**

Changing password for user tomcatadmUtpal.

New password: **tomcatadmUtpal**

BAD PASSWORD: The password contains the user name in some form

Retype new password: **tomcatadmUtpal**

passwd: all authentication tokens updated successfully.

[root@ip-172-31-30-86 etc]#

Dns Related info.

[root@ip-172-31-30-86 etc]# cat resolv.conf

# Generated by NetworkManager

search us-east-2.compute.internal

nameserver 172.31.0.2

HostName OR A fully qualified domain address (FQDA) is a string forming an Internet e-mail address

**To get HostName Info**

**[root@ip-172-31-30-86 etc]# hostname -f**

**ip-172-31-30-86.us-east-2.compute.internal**

**[root@ip-172-31-30-86 etc]# hostname -i**

**172.31.30.86**

**[root@ip-172-31-30-86 etc]# ifconfig**

**[root@ip-172-31-30-86 etc]# cat /etc/passwd | awk -F ":"'{print $l}'**

**[root@ip-172-31-30-86 etc]# cat /etc/passwd | awk -F ":"'{print $l}' |sort**

**[root@ip-172-31-30-86 etc]# cat /etc/passwd | awk -F ":"'{print $l}' |sort -r**

**[root@ip-172-31-30-86 etc]# cat /etc/passwd | awk -F ":"'{print $l}' |sort -r |wc -l**

**[root@ip-172-31-30-86 etc]# cat /etc/passwd | awk -F ":"'{print $NF}'**

**[root@ip-172-31-30-86 etc]# cat /etc/passwd | awk -F ":"'{print $NF}' |uniq**

**[root@ip-172-31-30-86 etc]# cat /etc/passwd | awk -F ":"'{print $NF}' |uniq -c**

**Find method to find a file or Directory**

**[root@ip-172-31-30-86 opt]# find /opt -type f -name "\*.rpm"**

**File size is More than 10MB**

**[root@ip-172-31-30-86 opt]# find /opt -type f -name "\*.rpm" -size +10M**

**it will check older 10 days data.**

**[root@ip-172-31-30-86 opt]# find /opt -type f -name "\*.rpm" -size +10M -mtime -10**

**GIT Essentials**

**[root@ip-172-31-30-86 opt]# clear**

**[root@ip-172-31-30-86 opt]# cd /opt**

**[root@ip-172-31-30-86 opt]# sha1sum helloworld.txt**

**329cd12216419c4181bf0b5823cc08fe6c8da4aa helloworld.txt**

**[root@ip-172-31-30-86 opt]# cp helloworld.txt abc.txt**

**cp: overwrite 'abc.txt'? y**

**[root opt]# sha1sum abc.txt**

**811a15e85b92e02ab965dd1ac8098c0e0d65debe abc.txt**

**[root@ip-172-31-30-86 opt]# md5sum abc.txt**

**7db316c5c213cdba7d2b5d2aebb36785 abc.txt**

**[root@ip-172-31-30-86 opt]# sha256sum abc.txt**

**335ad96fca20d5da4df54bcaf0282f0a705f9f9df7709773b6b8b54bf8a5a2d8 abc.txt**

**[root opt]# sudo dnf install git**

**[root@ip-172-31-32-109 opt]# rpm -qa git**

**[root@ip-172-31-32-109 opt]# which git**

**/usr/bin/git**

**[root@ip-172-31-32-109 opt]# git --version**

**git version 2.18.2**

**git config --global user.name**

**[root opt]# git config --global user.name**

**[root opt]# git config --global user.name "Utpal Maiti"**

**[root opt]# git config --global user.email "Utpalmaiti@gmail.com"**

**[root opt]# git config --global --list**

**[root@ip-172-31-32-109 opt]# cd /root**

**[root@ip-172-31-32-109 ~]# pwd**

**/root**

**[root@ip-172-31-32-109 ~]# cd /home**

**[root@ip-172-31-32-109 home]# pwd**

**/home**

**[root@ip-172-31-32-109 home]# ls**

**ansible docker ec2-user git github tomcatadm**

**[root@ip-172-31-32-109 home]# ls -a**

**. .. ansible docker ec2-user git github tomcatadm**

**[root@ip-172-31-32-109 home]# cd /root**

**[root@ip-172-31-32-109 ~]# ls -a**

**. .. anaconda-ks.cfg .ansible .bash\_history .bash\_logout .bash\_profile .bashrc .cshrc .gitconfig original-ks.cfg .ssh .tcshrc .viminfo**

**[root@ip-172-31-32-109 ~]#**

**git clone** [**https://github.com/git/git**](https://github.com/git/git)

**home directory of root user is /root**

**normal User home directory is /home**

**[root@ip-172-31-32-109 git]# cd /root**

**[root@ip-172-31-32-109 ~]# ls**

**anaconda-ks.cfg original-ks.cfg**

**[root@ip-172-31-32-109 ~]# ls -a**

**. .. anaconda-ks.cfg .ansible .bash\_history .bash\_logout .bash\_profile .bashrc .cshrc .gitconfig original-ks.cfg .ssh .tcshrc .viminfo**

**[root@ip-172-31-32-109 ~]# vim .gitconfig**

**[root@ip-172-31-32-109 ~]#**

**[root ~]# cd /opt**

**[root opt]# ls**

**[root opt]# mkdir myjavaapp**

**[root opt]# ls**

**[root opt]# ls -a**

**[root opt]# ls -l**

**[root opt]# rm -rf .git**

**[root opt]# cd myjavaapp/**

**[root myjavaapp]# git init**

**[root myjavaapp]# ls -a**

**[root myjavaapp]# vi Hello.java**

**(please type some content)**

**[root myjavaapp]# git status**

**[root myjavaapp]# git add Hello.java**

**[root@ip-172-31-32-109 myjavaapp]# git commit -m "Thi si smy First Commit"**

**[root@ip-172-31-32-109 myjavaapp]# vim Readme.MD**

**[root@ip-172-31-32-109 myjavaapp]# git status**

**[root myjavaapp]# git diff**

**[root myjavaapp]# git diff --staged**

**[root@ip-172-31-32-109 myjavaapp]# git commint -m "Adding Readme File"**

**[root@ip-172-31-32-109 myjavaapp]# git log --stat**

**[root@ip-172-31-32-109 myjavaapp]# git log --oneline**

**[root@ip-172-31-32-109 myjavaapp]# git log -n 2**

**[root@ip-172-31-32-109 myjavaapp]# git log --author="Utpal"**

**[root@ip-172-31-32-109 myjavaapp]# git log --author="Utpal" -n 2**

**[root@ip-172-31-32-109 myjavaapp]# git log --grep="Utpal"**

**[root@ip-172-31-32-109 myjavaapp]# git log --grep="Thi"**

**[root@ip-172-31-32-109 myjavaapp]# git diff HEAD~1**

**[root@ip-172-31-32-109 myjavaapp]# git diff HEAD~2**

**[root@ip-172-31-32-109 myjavaapp]# git log --oneline**

**66ebd4a (HEAD -> master) Adding Readme File**

**f80605d Thi si smy First Commit**

**[root@ip-172-31-32-109 myjavaapp]# git diff f80605d 66ebd4a**

**[root@ip-172-31-32-109 myjavaapp]# git diff f80605d 66ebd4a Hello.java**

**[root@ip-172-31-32-109 myjavaapp]# git checkout HEAD~1 >> all Reverted to previous Stg**

**[root@ip-172-31-32-109 myjavaapp]# git checkout HEAD~1 Hello.java**

**>> Only Hello.java Reverted to previous Stg**

**OR**

**[root@ip-172-31-32-109 myjavaapp]# git checkout f80605d >> all Reverted to previous Stg**

**[root@ip-172-31-32-109 myjavaapp]# git checkout f80605d Hello.java**

**>> Only Hello.java Reverted to previous Stg**

**[root myjavaapp]# git reset HEAD Hello.java >> Unstg the file from stg area**

**[root@ip-172-31-32-109 myjavaapp]# git commit -amend -m "restored the file to f80605d"**

**[root@ip-172-31-32-109 myjavaapp]# echo 'app info ' > access.log**

**[root@ip-172-31-32-109 myjavaapp]# touch .gitignore**

**[root myjavaapp]# git add .**

**[root myjavaapp]# git commit -m "Added ignore file to "**

**[root@ip-172-31-32-109 myjavaapp]# git log --oneline**

**[root@ip-172-31-32-109 myjavaapp]# git tag 1.2.3 49771c3**

**[root@ip-172-31-32-109 myjavaapp]# git log --stat**

**commit 49771c3ca62b85bc41ab7b0f0231d6cb8d14eb5a (HEAD -> master, tag: 1.2.3)**

**use HTTPS URL**

[**https://github.com/UtpalMaiti/myjavaappgit.git**](https://github.com/UtpalMaiti/myjavaappgit.git)

### …or create a new repository on the command line

echo "# myjavaappgit" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/UtpalMaiti/myjavaappgit.git

git push -u origin master

### …or push an existing repository from the command line

git remote add origin https://github.com/UtpalMaiti/myjavaappgit.git

git push -u origin master

**[root myjavaapp]# git remote -v**

**Please make sure you have the correct access rights and the repository exists.**

**[root myjavaapp]# git remote add mycentral https://github.com/UtpalMaiti/myjavaappgit.git**

**[root myjavaapp]# git remote -v**

**mycentral https://github.com/UtpalMaiti/myjavaappgit.git (fetch)**

**mycentral https://github.com/UtpalMaiti/myjavaappgit.git (push)**

**[root myjavaapp]# git status**

**[root myjavaapp]# git branch -a**

**\* master**

**[root myjavaapp]# git push mycentral master**

**[root myjavaapp]# git remote rm mycentral**

**[root myjavaapp]# git remote -v**

**[root myjavaapp]# git remote rm mycentralHTTPS**

**[root myjavaapp]# git remote add mycentral https://github.com/UtpalMaiti/myjavaapp.git**

**[root myjavaapp]# git remote -v**

**[root opt]# cd /root**

**[root ~]# git clone** [**https://github.com/UtpalMaiti/myjavaapp.git**](https://github.com/UtpalMaiti/myjavaapp.git)

**[root myjavaapp]# git branch -a**

**[root myjavaapp]# git remote -v**

**[root myjavaapp]# git pull origin master**

**[root myjavaapp]# git branch feature-abc >> Create a Baranch**

**[root myjavaapp]# git branch -a >> to view all Branch**

**[root myjavaapp]# git checkout feature-abc >> Switched to branch 'feature-abc'**

[**https://bitbucket.org/product**](https://bitbucket.org/product)

[**https://github.com/UtpalMaiti/myjavaappgit**](https://github.com/UtpalMaiti/myjavaappgit)

**Branching Strategies**

**[root@ip-172-31-32-109 myjavaappgit]# git branch feature-fix12 [root@ip-172-31-32-109 myjavaappgit]# git branch -a**

**[root@ip-172-31-32-109 myjavaappgit]# git checkout feature-fix12**

**[root@ip-172-31-32-109 myjavaappgit]# git branch -a**

**[root@ip-172-31-32-109 myjavaappgit]# ls**

**[root@ip-172-31-32-109 myjavaappgit]# git add .**

**[root@ip-172-31-32-109 myjavaappgit]# git commit -m "Added new code for Feature-fix12"**

**[root@ip-172-31-32-109 myjavaappgit]# git remote -v**

**[root@ip-172-31-32-109 myjavaappgit]# git push origin feature-fix12**

#### **CREATE A pull requests to marge the code to master from another branch**

**Assignments**

#### Clone with SSH

[git@github.com:UtpalMaiti/myjavaappgit.git](mailto:git@github.com:UtpalMaiti/myjavaappgit.git)

**[root myjavaapp]# git remote add mycentral https://github.com/UtpalMaiti/myjavaapp.git**

**[root@ip-172-31-32-109 myjavaappgit]# git remote -v**

#### **Maven(JAVA Build Tool)**

#### [**https://search.maven.org/**](https://search.maven.org/)

#### **java-1.8.0-openjdk-devel**

**Maven only for JAVA Build Tool**

**Cmake is Only for C Language Build Tool**

**MSBuild is for Dotnet Build Tool**

**JAVA Based Applications Salesforce, ServiceNow**

#### [**https://openjdk.java.net/install/**](https://openjdk.java.net/install/)

#### **[root@ip-172-31-30-86 ~]# yum install -y java-1.8.0-openjdk-devel**

**[root@ip-172-31-30-86 ~]# java -version**

[**https://maven.apache.org/download.cgi**](https://maven.apache.org/download.cgi)

//Capital O

**[root@ip-172-31-30-86 opt]# curl -O https://www-us.apache.org/dist/maven/maven-3/3.6.2/binaries/apache-maven-3.6.2-bin.tar.gz**

**[root@ip-172-31-32-109 Maven]# curl -O** [**https://www-eu.apache.org/dist/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz**](https://www-eu.apache.org/dist/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz)

**[root@ip-172-31-32-109 Maven]# tar Zxf apache-maven-3.6.3-bin.tar.gz**

**[root@ip-172-31-32-109 apache-maven-3.6.3]# cd conf**

**[root@ip-172-31-32-109 conf]# ls**

**logging settings.xml toolchains.xml**

**[root@ip-172-31-32-109 apache-maven-3.6.3]# cd bin**

**[root@ip-172-31-32-109 bin]# ls**

**m2.conf mvn mvn.cmd mvnDebug mvnDebug.cmd mvnyjp**

**[root@ip-172-31-32-109 bin]# .mvn --version**

**-bash: .mvn: command not found**

**[root@ip-172-31-32-109 bin]# ./mvn --version**

**[root@ip-172-31-32-109 bin]# pwd**

**/opt/Maven/apache-maven-3.6.3/bin**

**[root@ip-172-31-32-109 bin]# export =$PATH:/opt/Maven/apache-maven-3.6.3/bin**

**-bash: export: `=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/opt/Maven/apache-maven-3.6.3/bin': not a valid identifier**

**[root@ip-172-31-32-109 bin]# export PATH=$PATH:/opt/Maven/apache-maven-3.6.3/bin**

**[root@ip-172-31-32-109 opt]# git clone https://github.com/gkdevops/maven-sample-code.git**

**[root@ip-172-31-32-109 maven-sample-code]# tree**

[root@ip-172-31-32-109 maven-sample-code]# vim src/main/java/com/flipkart/App.java [root@ip-172-31-32-109 maven-sample-code]# vim src/test/java/com/flipkart/AppTest.java

**JAR: It’s a basic file with just sample java classes**

**WAR: When we use the serverlets, JSP’s and Web Content**

**EAR: EJB,Hibernate,Multiple WAR nad JAR’s**

**Source code => Build Tool =>Artifacts**

**BUILD:**

**1.Downloads dependency files for the source code.**

**2.Downloads files for the maven runtime.**

**[root@ip-172-31-32-109 maven-sample-code]# mvn compile**

**[root@ip-172-31-32-109 maven-sample-code]# cd ~**

**[root@ip-172-31-32-109 ~]# ls**

**anaconda-ks.cfg myjavaappgit original-ks.cfg**

**[root@ip-172-31-32-109 ~]# ls -a**

**. .. anaconda-ks.cfg .ansible .bash\_history .bash\_logout .bash\_profile .bashrc .cshrc .gitconfig .m2 myjavaappgit original-ks.cfg .ssh .tcshrc .viminfo**

**[root@ip-172-31-32-109 ~]# cd .m2**

**[root@ip-172-31-32-109 .m2]# ls**

**repository**

**[root@ip-172-31-32-109 .m2]# cd repository/**

**[root@ip-172-31-32-109 repository]# ls**

**backport-util-concurrent com commons-io junit org**

**[root@ip-172-31-32-109 repository]# pwd**

**/root/.m2/repository**

**[root@ip-172-31-32-109 repository]#**

**Maven phases(Default LifeCycle)**

**Compile=>test-compile=>test=>package=>integration-test=>verify=>install=>deploy**

**[root@ip-172-31-32-109 maven-sample-code]# mvn compile**

**[root@ip-172-31-32-109 maven-sample-code]# mvn test-compile**

**[root@ip-172-31-32-109 maven-sample-code]# mvn test**

**[root@ip-172-31-32-109 maven-sample-code]# vim pom.xml**

**[root@ip-172-31-32-109 maven-sample-code]# mvn package =>by default it will create jar file**

**[root@ip-172-31-32-109 maven-sample-code]# mvn clean**

**[root@ip-172-31-32-109 maven-sample-code]# vi pom.xml**

**[root@ip-172-31-32-109 maven-sample-code]# mvn clean package**

**<package>war/ear/jar</package>**

#### **Ansible(Automation Language)(CD)**

**(Continues Delivery) (Chef and Puppet)**

**Centos 7 for Master and Ansible installation.**

**https://releases.ansible.com/ansible/rpm/release/epel-7-x86\_64/ansible-2.8.6-1.el7.ans.noarch.rpm**

**[root@ip-172-31-41-83 ~]# curl -O https://releases.ansible.com/ansible/rpm/release/epel-7-x86\_64/ansible-2.8.6-1.el7.ans.noarch.rpm**

**[root@ip-172-31-41-83 ~]# yum install ansible-2.8.6-1.el7.ans.noarch.rpm -y**

**[root@ip-172-31-41-83 ~]# ansible –version**

**[root@ip-172-31-41-83 ansible]# pwd**

**/etc/ansible**

**[root@ip-172-31-41-83 ansible]# vim ansible.cfg**

**Changing Configuration (Ansible confing File)**

**[root@ip-172-31-41-83 ansible]# vim inventory.txt**

**[root@ip-172-31-41-83 ansible]# ssh-keygen**

**Generating public/private rsa key pair.**

**Enter file in which to save the key (/root/.ssh/id\_rsa):**

**/root/.ssh/id\_rsa already exists.**

**Overwrite (y/n)? y**

**Enter passphrase (empty for no passphrase):**

**Enter same passphrase again:**

**Your identification has been saved in /root/.ssh/id\_rsa.**

**Your public key has been saved in /root/.ssh/id\_rsa.pub.**

**The key fingerprint is:**

**SHA256:GbMNQhRg+uxXixTKARWE6AJv4zsd7IaPvlNTYgBY6UA root@ip-172-31-41-83.us-east-2.compute.internal**

**The key's randomart image is:**

**+---[RSA 2048]----+**

**|+Eo=\*++. |**

**|= ++ . |**

**|o+... o + |**

**|..=+oo.o B |**

**|.o +=o. S . |**

**| ..=. o . |**

**| \*.oo . |**

**| =.+. |**

**| .+\*. |**

**+----[SHA256]-----+**

**Don’t pass any info on prompt. On hit enter key three times**

**[root@ip-172-31-41-83 ansible]# cd /root**

**[root@ip-172-31-41-83 ~]# ls -a**

**. anaconda-ks.cfg ansible-2.8.6-1.el7.ans.noarch.rpm .bash\_logout .bashrc original-ks.cfg .ssh .viminfo**

**.. .ansible .bash\_history .bash\_profile .cshrc .pki .tcshrc**

**[root@ip-172-31-41-83 ~]# cd .ssh**

**[root@ip-172-31-41-83 .ssh]# ls**

**authorized\_keys id\_rsa id\_rsa.pub known\_hosts**

**[root@ip-172-31-41-83 .ssh]# ls lh**

**ls: cannot access lh: No such file or directory**

**[root@ip-172-31-41-83 .ssh]# ls -l**

**total 16**

**-rw-------. 1 root root 544 Nov 17 07:45 authorized\_keys**

**-rw-------. 1 root root 1679 Jan 21 15:09 id\_rsa**

**-rw-r--r--. 1 root root 429 Jan 21 15:09 id\_rsa.pub**

**-rw-r--r--. 1 root root 219 Nov 17 14:28 known\_hosts**

**[root@ip-172-31-41-83 .ssh]#**

**//Client Server RHEL**

**[root@ip-172-31-32-109 ~]# /etc/ssh/sshd\_config**

**-bash: /etc/ssh/sshd\_config: Permission denied**

**[root@ip-172-31-32-109 ~]# sudo vim /etc/ssh/sshd\_config**

**[root@ip-172-31-32-109 ~]# pwd**

**/root**

**(Enable Password Authentication edit config file)**

**[root@ip-172-31-32-109 ~]# passwd**

**Changing password for user root.**

**New password:** Password

**BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word**

**Retype new password:** Password

**passwd: all authentication tokens updated successfully.**

**Set a password and Restart**

**[root@ip-172-31-32-109 ~]# systemctl restart sshd**

**[root@ip-172-31-32-109 ~]# view /etc/passwd**

**From Master server (Coying SSH Certification Key )**

**[root@ip-172-31-41-83 .ssh]# ssh-copy-id root@ip-172-31-32-109.us-east-2.compute.internal**

**/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id\_rsa.pub"**

**/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed**

**/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys**

**root@ip-172-31-32-109.us-east-2.compute.internal's password:** Password

**Number of key(s) added: 1**

**Now try logging into the machine, with: "ssh 'root@ip-172-31-32-109.us-east-2.compute.internal'"**

**and check to make sure that only the key(s) you wanted were added.**

[**https://docs.ansible.com/ansible/latest/user\_guide/intro\_adhoc.html**](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html)

# Introduction to ad-hoc commands

[root@ip-172-31-41-83 ansible]# ansible webservers -m ping

[WARNING]: Unable to parse /opt/inventory.txt as an inventory source

[WARNING]: No inventory was parsed, only implicit localhost is available

[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

[WARNING]: Could not match supplied host pattern, ignoring: webservers

[root@ip-172-31-41-83 ansible]# vi inventory.txt

[root@ip-172-31-41-83 ansible]# cp inventory.txt /opt

[root@ip-172-31-41-83 ansible]# ls

ansible-code Helloworld.yml inventory.txt log

[root@ip-172-31-41-83 ansible]# cd ..

[root@ip-172-31-41-83 opt]# ls

ansible inventory.txt log

[root@ip-172-31-41-83 opt]# ansible webservers -m ping

ip-172-31-32-109.us-east-2.compute.internal | SUCCESS => {

"ansible\_facts": {

"discovered\_interpreter\_python": "/usr/libexec/platform-python"

},

"changed": false,

"ping": "pong"

}

[root@ip-172-31-41-83 opt]#

**[root@ip-172-31-41-83 opt]# ansible webservers -m user -a "name=ansible state=present"**

ip-172-31-32-109.us-east-2.compute.internal | SUCCESS => {

"ansible\_facts": {

"discovered\_interpreter\_python": "/usr/libexec/platform-python"

},

"append": false,

"changed": false,

"comment": "",

"group": 1001,

"home": "/home/ansible",

"move\_home": false,

"name": "ansible",

"shell": "/bin/bash",

"state": "present",

"uid": 1001

}

[root@ip-172-31-41-83 opt]#

[**https://docs.ansible.com/ansible/latest/modules/user\_module.html#user-module**](https://docs.ansible.com/ansible/latest/modules/user_module.html#user-module)

**playbook syntax Checking**

[root@ip-172-31-41-83 opt]# ansible-playbook Helloworld.yml --syntax-check

**Running Ansible Playbooks**

**[root@ip-172-31-41-83 opt]# ansible-playbook Helloworld.yml**

[root@ip-172-31-41-83 opt]# dnf install git -y >/dev/null

[root@ip-172-31-41-83 opt]# git clone <https://github.com/gkdevops/ansible-code.git>

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_hosts\_vars.yml --syntax-check**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_hosts\_vars.yml**

[root@ip-172-31-41-83 playbooks]# ansible-playbook vars\_simple.yml

[root@ip-172-31-32-109 opt]# vim helloworld.txt (This is Client Server Only)

[root@ip-172-31-32-109 opt]# stat abc.txt

**[root@ip-172-31-41-83 playbooks]# cat myvariables.yml**

**[root@ip-172-31-41-83 playbooks]# cat vars\_files.yml**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_files.yml**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_task.yml**

**[root@ip-172-31-41-83 playbooks]# ansible -i /opt/ansible-code/production-inv.yml webservers -m setup**

**[root@ip-172-31-41-83 playbooks]# ansible -i /opt/ansible-code/production-inv.yml webservers -m ping**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_extra.yml --extra-vars "mysoftware=unzip"**

**[root@ip-172-31-32-109 opt]# yum repolist**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_prec.yml**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_prec.yml --extra-vars "mysoftware=elinks"**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_simple.yml --tags=install**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_simple.yml --tags "install,configure"**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml vars\_simple.yml --skip-tags "install,configure"**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml forloop-1.yml --syntax-check**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml forloop-1.yml**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml array.yml**

**[root@ip-172-31-32-109 opt]# rm file1.txt -f**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml array.yml**

**-vvvv too more details for debug purpose**

**-v for minimum details for debug purpose.**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml array.yml -vvvv**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml multi-play.yml -vv**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml service.yml**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml command-1.yml**

**[Utpal@ip-172-31-41-83 playbooks]# $cd downloads/; ssh -i "Ansible.pem" centos@ec2-3-136-157-182.us-east-2.compute.amazonaws.com**

[**https://docs.ansible.com/ansible/latest/modules/list\_of\_commands\_modules.html**](https://docs.ansible.com/ansible/latest/modules/list_of_commands_modules.html)

**[root@ip-172-31-41-83 playbooks]# vim command-1.yml [root@ip-172-31-41-83 playbooks]# vim /opt/script.sh**

**[root@ip-172-31-41-83 playbooks]# chmod +x /opt/script.sh**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml command-1.yml**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml command-2.yml --syntax-check**

**[root@ip-172-31-41-83 playbooks]# ls**

**[root@ip-172-31-41-83 playbooks]# echo $?**

**127 (Return Code ,0 is success other than o failure)**

**[root@ip-172-31-41-83 playbooks]# ls 2>/dev/null**

**[root@ip-172-31-41-83 playbooks]# ls 1>/dev/null**

**{Run From localUser}**

**[root@ip-172-31-41-83 playbooks]# find / -type f -name "\*.log"**

**[root@ip-172-31-41-83 playbooks]# find / -type f -name "\*.log" 2>/dev/null**

[**https://docs.ansible.com/ansible/latest/user\_guide/playbooks\_filters.html#filters-for-formatting-data**](https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters.html#filters-for-formatting-data)

**[root@ip-172-31-41-83 playbooks]# ansible webservers -i /opt/ansible-code/production-inv.yml -m setup > /opt/facts.txt**

**[root@ip-172-31-41-83 playbooks]# vim /opt/facts.txt**

**[root@ip-172-31-41-83 playbooks]# vim facts-1.yml**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml facts-1.yml**

**[root@ip-172-31-41-83 playbooks]# vim facts-2.yml**

**[root@ip-172-31-41-83 playbooks]# ansible-playbook -i /opt/ansible-code/production-inv.yml facts-2.yml**

**Ansible Roles Structure**

[**https://docs.ansible.com/ansible/latest/user\_guide/playbooks\_reuse\_roles.html**](https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html)

**[root@ip-172-31-41-83 opt]# ansible-galaxy init apache**

**[root@ip-172-31-41-83 apache]# tree**

**[root@ip-172-31-41-83 apache]# ls**

**- name: To update the apache tomcat config copy:**

**src: /opt/server.xml**

**dest: /opt/apache-tomcat-8.5.45/conf/server.xml owner: tomcat**

**group: tomcatadm**

**ignore\_errors: yes**

**[root@ip-172-31-41-83 tasks]# cat ../handlers/main.yml**

**[root@ip-172-31-41-83 tasks]# cat configure.yml**

[**https://jinja.palletsprojects.com/en/2.10.x/**](https://jinja.palletsprojects.com/en/2.10.x/)

**[root@ip-172-31-32-109 opt]# sudo dnf remove httpd**

**[root@ip-172-31-41-83 production]# vim /etc/ansible/ansible.cfg**

[**https://galaxy.ansible.com/**](https://galaxy.ansible.com/)

**[root@ip-172-31-41-83 production]# vim /opt/ansible-code/roles/production/apache/tasks/main.yml -r [root@ip-172-31-41-83 production]# ansible-playbook -i /opt/ansible-code/production-inv.yml apache\_kelly.yml**

**[root@ip-172-31-14-27 ~]# yum install -y https://releases.ansible.com/ansible/rpm/release/epel-7-x86\_64/ansible-2.9.4-1.el7.ans.noarch.rpm**

[**https://releases.ansible.com/ansible-tower/**](https://releases.ansible.com/ansible-tower/)

**[root@ip-172-31-14-27 opt]# curl -O** [**https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-3.6.2-1.tar.gz**](https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-3.6.2-1.tar.gz)

**[root@ip-172-31-14-27 opt]# tar zxf ansible-tower-setup-3.6.2-1.tar.gz**

**[root@ip-172-31-14-27 ansible-tower-setup-3.6.2-1]# yum install -y tree vim dnf**

**[root@ip-172-31-14-27 ansible-tower-setup-3.6.2-1]# vim inventory**

**[root@ip-172-31-14-27 ansible-tower-setup-3.6.2-1]# ./setup.sh**

[**https://ec2-3-135-201-114.us-east-2.compute.amazonaws.com/**](https://ec2-3-135-201-114.us-east-2.compute.amazonaws.com/)

[**https://www.ansible.com/license**](https://www.ansible.com/license)

Verification ID: 121eeeb5

Message ID: e1c31e8c-2c83-4825-875e-c81187473de7

#### **Docker(Container)**

[**https://hub.docker.com/u/utpalmaiti/content/sub-9bdbc30c-b30f-4f9d-8123-b1cde9950dd0**](https://hub.docker.com/u/utpalmaiti/content/sub-9bdbc30c-b30f-4f9d-8123-b1cde9950dd0)

[**https://storebits.docker.com/ee/centos/sub-9bdbc30c-b30f-4f9d-8123-b1cde9950dd0/centos/7/x86\_64/stable-19.03/Packages/**](https://storebits.docker.com/ee/centos/sub-9bdbc30c-b30f-4f9d-8123-b1cde9950dd0/centos/7/x86_64/stable-19.03/Packages/)

**[root@ip-172-31-41-83 ~]# yum install wget curl tree vim dnf -y**

**[root@ip-172-31-41-83 opt]# wget https://storebits.docker.com/ee/centos/sub-9bdbc30c-b30f-4f9d-8123-b1cde9950dd0/centos/7/x86\_64/stable-19.03/Packages/containerd.io-1.2.6-3.3.el7.x86\_64.rpm**

**[root@ip-172-31-41-83 opt]# wget https://storebits.docker.com/ee/centos/sub-9bdbc30c-b30f-4f9d-8123-b1cde9950dd0/centos/7/x86\_64/stable-19.03/Packages/docker-ee-19.03.5-3.el7.x86\_64.rpm**

**[root@ip-172-31-41-83 opt]# wget https://storebits.docker.com/ee/centos/sub-9bdbc30c-b30f-4f9d-8123-b1cde9950dd0/centos/7/x86\_64/stable-19.03/Packages/docker-ee-cli-19.03.5-3.el7.x86\_64.rpm**

**[root@ip-172-31-41-83 opt]# yum install \*.rpm -y**

**[root@ip-172-31-41-83 opt]# systemctl status docker**

**[root@ip-172-31-41-83 opt]# systemctl start docker**

**[root@ip-172-31-41-83 opt]# systemctl enable docker**

**[root@ip-172-31-41-83 opt]# docker image ls --help**

**[root@ip-172-31-41-83 opt]# docker pull httpd:latest**

**[root@ip-172-31-41-83 opt]# docker pull registry1.abc.com/httpd:latest**

**[root@ip-172-31-41-83 opt]# cd /var/lib/docker**

**[root@ip-172-31-41-83 docker]# ls**

**[root@ip-172-31-41-83 opt]# docker container**

**[root@ip-172-31-41-83 opt]# docker container run -d -p 21000:80 --name apache-1 httpd:latest**

**[root@ip-172-31-41-83 opt]# docker container ls**

[**http://ec2-3-15-175-201.us-east-2.compute.amazonaws.com:21000/**](http://ec2-3-15-175-201.us-east-2.compute.amazonaws.com:21000/)

**[root@ip-172-31-41-83 opt]# ps -ef | grep docker**

**[root@ip-172-31-41-83 opt]# docker container run -d -p 7777:80 --name apache-2 httpd:latest**

[**http://ec2-3-15-175-201.us-east-2.compute.amazonaws.com:7777/**](http://ec2-3-15-175-201.us-east-2.compute.amazonaws.com:7777/)

**[root@ip-172-31-41-83 opt]# docker container stop 55b4170cda3c**

**[root@ip-172-31-41-83 opt]# docker container ls**

**[root@ip-172-31-41-83 opt]# docker container ls -a**

**[root@ip-172-31-41-83 opt]# docker container kill apache-2**

**[root@ip-172-31-41-83 opt]# docker container ls -a**

**[root@ip-172-31-41-83 opt]# docker container start apache-1 apache-2**

**[root@ip-172-31-41-83 opt]# docker container pause apache-1**

**[root@ip-172-31-41-83 opt]# docker container ls**

**[root@ip-172-31-41-83 opt]# docker container unpause apache-1**

**[root@ip-172-31-41-83 opt]# docker container ls**

**[root@ip-172-31-41-83 opt]# free -h**

**[root@ip-172-31-41-83 opt]# docker container stats 55b4170cda3c**

**[root@ip-172-31-41-83 opt]# docker container top apache-1**

**[root@ip-172-31-41-83 opt]# docker container logs apache-1**

**[root@ip-172-31-41-83 opt]# cat /etc/os-release**

**[root@ip-172-31-41-83 opt]# docker container exec -it apache-1 /bin/bash**

**root@55b4170cda3c:/usr/local/apache2# hostname**

**root@55b4170cda3c:/usr/local/apache2# ls**

**root@55b4170cda3c:/usr/local/apache2# hostname -i**

**172.17.0.2**

**root@55b4170cda3c:/usr/local/apache2# pwd**

**/usr/local/apache2**

**root@55b4170cda3c:/usr/local/apache2/htdocs# echo "<html><body><h1>It works! This is Docker Engine ,It is Running in Container....</h1></body></html>" > index.html**

**root@55b4170cda3c:/usr/local/apache2/htdocs# cat /etc/os-release**

**root@55b4170cda3c:/usr/local/apache2/htdocs# exit**

**exit**

**[root@ip-172-31-41-83 opt]# docker container ls**

**[root@ip-172-31-41-83 opt]# docker container rm apache-1**

**Error response from daemon: You cannot remove a running container 55b4170cda3c3393d2019f47c2d56892e935c8fc06e54e2278d531724cfad54a. Stop the container before attempting removal or force remove**

**[root@ip-172-31-41-83 opt]# docker container rm apache-1 -f**

**apache-1**

**[root@ip-172-31-41-83 opt]# docker container stop apache-2**

**apache-2**

**[root@ip-172-31-41-83 opt]# docker container rm apache-2**

**apache-2**

**[root@ip-172-31-41-83 opt]# docker container ls**

**CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES**

**[root@ip-172-31-41-83 opt]#**

**[root@ip-172-31-41-83 helloworld]# docker image build -t helloworld:1.0.0 . (Same Location)**

**[root@ip-172-31-41-83 helloworld]# docker image build -t helloworld:1.0.0 -f /path (To Spacify Location)**

**[root@ip-172-31-41-83 helloworld]# docker image ls**

**[root@ip-172-31-41-83 helloworld]# docker container run --name hello1 helloworld:1.0.0**

**[root@ip-172-31-41-83 apache]# touch dockerfile [root@ip-172-31-41-83 apache]# vim dockerfile [root@ip-172-31-41-83 apache]# utpal; date**

**[root@ip-172-31-41-83 apache]# utpal && date && echo "Hello"**

**[root@ip-172-31-41-83 apache]# docker image build -t myapache:1.0.0 .**

**[root@ip-172-31-41-83 apache]# docker image history 5243a76af43f**

**[root@ip-172-31-41-83 apache]# docker container ls -a**

**[root@ip-172-31-41-83 apache]# docker container run -d -p 21000:80 --name apache1 myapache:1.0.0**

#### **Kubernetes(Container Management)**

[**https://kubernetes.io/**](https://kubernetes.io/)

[**https://kubernetes.io/docs/setup/**](https://kubernetes.io/docs/setup/)

<https://aws.amazon.com/eks/>

RDS

<https://azure.microsoft.com/en-in/services/kubernetes-service/>

<https://cloud.google.com/kubernetes-engine>

<https://eksworkshop.com/>

<https://dzone.com/articles/deploying-a-kubernetes-cluster-with-amazon-eks>

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

<https://docs.docker.com/install/linux/docker-ce/centos/>

**[root@ip-172-31-10-173 ~]# sudo yum install -y tree vim git tree**

**sudo yum install -y yum-utils \**

**device-mapper-persistent-data \**

**lvm2**

sudo **yum-config-manager \**

**--add-repo \**

**https://download.docker.com/linux/centos/docker-ce.repo**

**[root@ip-172-31-14-27 opt]# yum repolist**

**[root@ip-172-31-14-27 opt]# sudo yum install docker-ce docker-ce-cli containerd.io -y**

**[root@ip-172-31-14-27 opt]# systemctl status docker**

**[root@ip-172-31-14-27 opt]# systemctl start docker**

**[root@ip-172-31-14-27 opt]# systemctl status docker**

**[root@ip-172-31-14-27 opt]# swapoff -a (to DIsbale Swap Memory)**

**[root@ip-172-31-14-27 kubernets]# vim install.sh**

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

EOF

*# Set SELinux in permissive mode (effectively disabling it)*

setenforce 0

sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

systemctl enable --now kubelet

**[root@ip-172-31-14-27 kubernets]# cp install install.sh**

**[root@ip-172-31-14-27 kubernets]# sh install.sh**

<https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Addresses:public-ip=3.130.182.208>

**Elastic IP address allocated.**

Elastic IP address 3.130.182.208

Associate this Elastic IP address

1. [EC2](https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Home:)
2. [Elastic IP addresses](https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Addresses:)
3. [Associate Elastic IP address](https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2)

Top of Form

Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (3.130.182.208)

Bottom of Form

**sudo yum install -y yum-utils \ device-mapper-persistent-data \ lvm2 sudo yum-config-manager \ --add-repo \ https://download.docker.com/linux/centos/docker-ce.repo sudo yum install docker-ce docker-ce-cli containerd.io -y systemctl status docker systemctl start docker systemctl status docker swapoff -a cat <<EOF > /etc/yum.repos.d/kubernetes.repo [kubernetes] name=Kubernetes baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64 enabled=1 gpgcheck=1 repo\_gpgcheck=1 gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg EOF # Set SELinux in permissive mode (effectively disabling it) setenforce 0 sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes systemctl enable --now kubelet**

**[root@ip-172-31-14-27 kubernets]# sh install.sh**

**sudo kubeadm init**

**kubeadm join 172.31.10.173:6443 --token 3620l6.go7yx2yr72j4qos4 \**

**--discovery-token-ca-cert-hash sha256:f2227876906ce1ef04ae679bf188e9400710cb616c14f4ffaa3d078e762bef41**

**[root@ip-172-31-2-222 ~]# exit**

**[centos@ip-172-31-10-173 ~]$ sudo mkdir -p $HOME/.kube**

**udo cp -[centos@ip-172-31-10-173 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config**

**sudo chown[centos@ip-172-31-10-173 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config**

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown **$(**id -u**)**:**$(**id -g**)** $HOME/.kube/config

**[centos@ip-172-31-10-173 ~]$ sudo yum install curl wget -y**

<https://kubernetes.io/docs/tasks/tools/install-kubectl/>

**[centos@ip-172-31-10-173 ~]$ sudo curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stab le.txt`/bin/linux/amd64/kubectl**

**[centos@ip-172-31-10-173 ~]$ sudo curl -LO** [**https://storage.googleapis.com/kubernetes-**](https://storage.googleapis.com/kubernetes-)**release/release/v1.17.0/bin/linux/amd64/kubectl**

**[centos@ip-172-31-10-173 ~]$ sudo chmod +x ./kubectl**

**[centos@ip-172-31-10-173 ~]$ sudo mv ./kubectl /usr/local/bin/kubectl [centos@ip-172-31-10-173 ~]$ kubectl version --client Client Version: version.Info{Major:"1", Minor:"17", GitVersion:"v1.17.2", GitCommit:"59603c6e503c87169aea6106f57b9f242f64df89", GitTreeState:"clean", BuildDate:"2020-01-18T23:30:10Z", GoVersion:"go1.13.5", Compiler:"gc", Platform:"linux/amd64"}**

**[root@ip-172-31-10-173 ~]# export KUBECONFIG=/etc/kubernetes/admin.conf**

kubectl get -h

kubectl get pods –namespace kube-syatem

sudo sysctl net.bridge.bridge-nf-call-iptables=1

kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=**$(**kubectl version | base64 | tr -d '\n'**)**"

kubectl get pods -–namespace kube-syatem

kubctl get nodes

docker ps | grep kube-apiserver

**[centos@ip-172-31-10-173 ~]$ kubectl cluster-info**

**Kubernetes master is running at https://172.31.10.173:6443**

**KubeDNS is running at** [**https://172.31.10.173:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy**](https://172.31.10.173:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy)

**kubectl cluster-info dump**

**[centos@ip-172-31-10-173 ~]$ kubectl get nodes -o wide**

### **Installing a Pod network add-on**

[Weave Net](https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/#tabs-pod-install-5)

**[root@ip-172-31-10-173 ~]# systemctl daemon-reload**

**[root@ip-172-31-10-173 ~]# systemctl restart kubelet**

**[centos@ip-172-31-10-173 ~]$ kubectl get pods --namespace kube-system**

**[centos@ip-172-31-10-173 ~]$ sudo sysctl net.bridge.bridge-nf-call-iptables=1**

**[centos@ip-172-31-10-173 ~]$ sudo kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"**

**[centos@ip-172-31-10-173 ~]$ kubectl get pods --namespace kube-system**

**[centos@ip-172-31-10-173 ~]$ kubectl get pods**

**NAME READY STATUS RESTARTS AGE**

**nginx-579579fb7c-2d72v 0/1 Pending 0 23m**

**[centos@ip-172-31-10-173 ~]$ kubectl get nodes**

**NAME STATUS ROLES AGE VERSION**

**ip-172-31-10-173.us-east-2.compute.internal Ready master 66m v1.17.2**

**[centos@ip-172-31-10-173 ~]$ kubectl get pods --namespace kube-system**

**[centos@ip-172-31-10-173 ~]$ kubectl get nodes**

**After install shell scripts run this commands**

**[centos@ip-172-31-10-173 ~]$ sudo sysctl net.bridge.bridge-nf-call-iptables=1**

**kubeadm join 172.31.10.173:6443 --token h5uykq.k2xyozl9ql3ry21g \**

**--discovery-token-ca-cert-hash sha256:6dd0c769a95e533ad4cbc609f8dc8479d308f606cdd2edcab7a5cb9a18b289bb**

**[root@ip-172-31-10-173 ~]# kubectl get namespace**

**[centos@ip-172-31-10-173 ~]$ kubectl apply -f pod.yml**

**[root@ip-172-31-10-173 ~]# kubectl get all**

**[root@ip-172-31-10-173 ~]# kubectl get pods**

**[root@ip-172-31-10-173 ~]# kubectl get pods -o wide**

**[root@ip-172-31-10-173 ~]# docker container ls**

**[root@ip-172-31-10-173 ~]# kubectl describe pod sample-pod**

**[root@ip-172-31-10-173 ~]# kubectl exec -it sample-pod /bin/bash**

**[centos@ip-172-31-10-173 ~]$ kubectl apply -f pod.yml**

**[centos@ip-172-31-10-173 ~]$ kubectl delete -f pod.yml**

**[centos@ip-172-31-10-173 ~]$ kubectl delete pod pod.yml**

[**https://github.com/gkdevops/kubernetes-code**](https://github.com/gkdevops/kubernetes-code)

[**https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/**](https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/)

**[centos@ip-172-31-10-173 ~]$ vim replicaset.yaml**

**[centos@ip-172-31-10-173 ~]$ kubectl apply -f replicaset.yaml**

**[root@ip-172-31-10-173 ~]# kubectl get pods**

**[root@ip-172-31-10-173 ~]# kubectl get all**

**[root@ip-172-31-10-173 ~]# kubectl get replicaset**

**[root@ip-172-31-10-173 ~]# kubectl get pods**

**[root@ip-172-31-10-173 ~]# kubectl get nodes**

**[root@ip-172-31-10-173 ~]# systemctl status kubectl**

**[root@ip-172-31-10-173 ~]# systemctl start kubectl**

**[root@ip-172-31-10-173 ~]# systemctl start docker**

**[root@ip-172-31-10-173 ~]# systemctl start kubectl**

**[root@ip-172-31-10-173 ~]# kubectl get pods**

**[root@ip-172-31-10-173 ~]# kubectl get all**

**[centos@ip-172-31-10-173 ~]$ kubectl apply -f** **deployment.yaml**

**[root@ip-172-31-10-173 ~]# kubectl get deployments**

**[root@ip-172-31-10-173 ~]# kubectl get replicates**

**[root@ip-172-31-10-173 ~]# kubectl get pods**

**[root@ip-172-31-10-173 ~]# kubectl rollout status deployment.v1.apps/nginx-deployment**

[**https://hub.docker.com/\_/nginx?tab=description**](https://hub.docker.com/_/nginx?tab=description)

**[centos@ip-172-31-10-173 ~]$ kubectl apply -f deployment.yaml**

**[root@ip-172-31-10-173 ~]# kubectl rollout status deployment.v1.apps/nginx-deployment**

**[root@ip-172-31-10-173 ~]# kubectl get deployments**

**[root@ip-172-31-10-173 ~]# kubectl get replicasets**

**[root@ip-172-31-10-173 ~]# kubectl get pods**

**[root@ip-172-31-10-173 ~]# vim deployment.yaml**

**[root@ip-172-31-10-173 ~]# kubectl describe deployment nginx-deployment**

**[root@ip-172-31-10-173 ~]# kubectl describe namespace default**

[**https://kubernetes.io/docs/concepts/policy/resource-quotas/**](https://kubernetes.io/docs/concepts/policy/resource-quotas/)

**[root@ip-172-31-10-173 ~]# kubectl describe namespace default**

**[root@ip-172-31-10-173 ~]# kubectl get nodes**

**[root@ip-172-31-10-173 ~]# systemctl start kubectl**

**[root@ip-172-31-10-173 ~]# systemctl start docker**

**[root@ip-172-31-10-173 ~]# kubectl get nodes**

**[root@ip-172-31-10-173 ~]# ls**

**[centos@ip-172-31-10-173 ~]$ kubectl delete -f resourceQuota.yml**

**[centos@ip-172-31-10-173 ~]$ kubectl delete -f limitRange.yml**

**[root@ip-172-31-10-173 ~]# kubectl describe namespace default**

**[centos@ip-172-31-10-173 ~]$ kubectl get pods**

**[centos@ip-172-31-10-173 ~]$ kubectl delete pod sample-pod**

**[centos@ip-172-31-10-173 ~]$ kubectl describe node ip-172-31-10-173.ec2.internal**

**[centos@ip-172-31-10-173 ~]$ kubectl taint nodes ip-172-31-10-173.ec2.internal hardware=slow:NoSchedule**

**[centos@ip-172-31-10-173 ~]$ kubectl describe node ip-172-31-10-173.ec2.internal | grep “Taint”**

[**https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/**](https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/)

kubectl taint nodes node1 key1=value1:NoSchedule

kubectl taint nodes node1 key1=value1:NoExecute

kubectl taint nodes node1 key2=value2:NoSchedule

**[centos@ip-172-31-10-173 ~]$ cp deployment.yaml taints.yaml**

**[centos@ip-172-31-10-173 ~]$ kubctl delete -f taints.yaml**

**[centos@ip-172-31-10-173 ~]$ kubctl apply -f taints.yaml**

**[centos@ip-172-31-10-173 ~]$ kubctl get pods -o wide**

**[centos@ip-172-31-10-173 ~]$ kubctl apply -f deployment-live-ready.yaml**

**[root@ip-172-31-10-173 ~]# kubectl get replicasets**

**[centos@ip-172-31-10-173 ~]$ cp deployment.yaml deamonset.yaml**

[**https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/**](https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/)

**[root@ip-172-31-10-173 ~]# !systemctl**

**[root@ip-172-31-10-173 ~]# systemctl start kubectl**

**[root@ip-172-31-10-173 ~]# systemctl start docker**

**[root@ip-172-31-10-173 ~]# cd services**

**[root@ip-172-31-10-173 ~]# vim ClusterIP.yaml**

**[root@ip-172-31-10-173 ~]# vim deployment-services.yaml**

**[root@ip-172-31-10-173 ~]# kubectl get all**

**[root@ip-172-31-10-173 ~]# kubectl apply -f deployment-services.yaml**

**[root@ip-172-31-10-173 ~]# kubectl get services**

**[root@ip-172-31-10-173 ~]# kubectl apply -f ClusterIP.yaml**

**[root@ip-172-31-10-173 ~]# kubectl describe service clusterip-services**

**[centos@ip-172-31-10-173 ~]$ kubectl get pods -o wide**

**[centos@ip-172-31-10-173 ~]$ kubectl apply -f pod.yaml**

**[centos@ip-172-31-10-173 ~]$ kubectl get pods -o wide**

**[centos@ip-172-31-10-173 ~]$ kubectl exec -it sample-pod /bin/bash**

**[centos@ip-172-31-10-173 ~]$ kubectl apt-get update -y >/dev/null**

**[centos@ip-172-31-10-173 ~]$ kubectl apt-get install curl wget -y >/dev/null**

**[centos@ip-172-31-10-173 ~]$ curl** [**http://10.96.67.48:80**](http://10.96.67.48:80)

**[centos@ip-172-31-10-173 ~]$ exit**

**[centos@ip-172-31-10-173 ~]$ curl** [**http://10.96.67.48:80**](http://10.96.67.48:80)

**[centos@ip-172-31-10-173 ~]$ exit**

**[centos@ip-172-31-10-173 ~]$ cd –**

**[centos@ip-172-31-10-173 ~]$ kubectl delete -f ClusterIP.yaml**

**[centos@ip-172-31-10-173 ~]$ vim NodePort.yaml**

**[root@ip-172-31-10-173 ~]# kubectl apply -f NodePort.yaml**

**[root@ip-172-31-10-173 ~]# kubectl get services**

**[root@ip-172-31-10-173 ~]# kubectl describe service NodePort -services**

**ec2-3-130-182-208.us-east-2.compute.amazonaws.com:30036**

**[centos@ip-172-31-10-173 ~]$ kubectl get pods -o wide**

**ec2-3-130-182-208.us-east-2.compute.amazonaws.com:30036**

[**https://metallb.universe.tf/**](https://metallb.universe.tf/)

**[centos@ip-172-31-10-173 ~]$ cp ClusterIP.yaml LoadBalancer.yaml**

**[centos@ip-172-31-10-173 ~]$ vim LoadBalancer.yaml**

[**https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LoadBalancers:sort=loadBalancerName**](https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LoadBalancers:sort=loadBalancerName)

#### **Jenkins(Container Delivery)**

[**https://jenkins.io/**](https://jenkins.io/)

**[root@ip-172-31-10-173 opt]# yum install -y dnf tree wget unzip >/dev/null**

**[root@ip-172-31-10-173 opt]# wget** [**http://mirrors.jenkins.io/war-stable/latest/jenkins.war**](http://mirrors.jenkins.io/war-stable/latest/jenkins.war)

<https://hub.docker.com/r/jenkinsci/jenkins>

**[root@ip-172-31-10-173 opt]# dnf install -y java-1.8.0-openjdk-devel**

**[root@ip-172-31-10-173 opt]# java -version**

**[root@ip-172-31-10-173 opt]# java -jar jenkins.war**

**Running from: /opt/jenkins.war**

**webroot: $user.home/.jenkins**

<http://3.130.182.208:8080/login?from=%2F>

**[root@ip-172-31-10-173 opt]# wget** [**https://www-us.apache.org/dist/tomcat/tomcat-9/v9.0.30/bin/apache-tomcat-9.0.30.tar.gz**](https://www-us.apache.org/dist/tomcat/tomcat-9/v9.0.30/bin/apache-tomcat-9.0.30.tar.gz)

**[root@ip-172-31-10-173 opt]# tar -zxf apache-tomcat-9.0.30.tar.gz**

<https://devblogs.microsoft.com/dotnet/introducing-net-5/>

**[root@ip-172-31-10-173 webapps]# cp /opt/jenkins.war .**

**[root@ip-172-31-10-173 bin]# ./startup.sh**

<http://3.130.182.208:8080/jenkins/login?from=%2Fjenkins%2F>

**[root@ip-172-31-10-173 bin]# cd /home/centos/.jenkins**

**[root@ip-172-31-10-173 bin]# cat /root/.jenkins/secrets/initialAdminPassword**

**2d35133baa1248198fe6132a3fb88597**

**utpal**

**utpal**

**utpal**

**utpal**

[**example@gmail.com**](mailto:example@gmail.com)

[**http://3.130.182.208:8080/jenkins/**](http://3.130.182.208:8080/jenkins/)

**[root@ip-172-31-32-109 opt]# vim /etc/ssh/sshd\_config**

**//PasswordAuthentication set to Yes**

**PermitRootLogin yes**

**[root@ip-172-31-32-109 opt]# passwd Changing password for user root. New password: utpal BAD PASSWORD: The password is shorter than 8 characters Retype new password: utpal passwd: all authentication tokens updated successfully.**

**[root@ip-172-31-10-173 ~]# passwd Changing password for user root. New password: utpal BAD PASSWORD: The password is shorter than 8 characters Retype new password: utpal passwd: all authentication tokens updated successfully.**

**[root@ip-172-31-32-109 opt]# systemctl restart sshd**

**[root@ip-172-31-32-109 opt]# mkdir jenkins**

**[root@ip-172-31-10-173 bin]# ./startup.sh**

**Login to Jenkins Portal**

<https://plugins.jenkins.io/>

Download From Here

<https://updates.jenkins.io/download/plugins/deploy/>

**[root@ip-172-31-10-173 /]# cd /root/.jenkins/**

**[root@ip-172-31-10-173 .jenkins]# pwd**

**/root/.jenkins**

**[root@ip-172-31-10-173 .jenkins]# cd plugins/**

**Place your plugin Files Here**

<http://3.130.182.208:8080/jenkins/restart>

**//For Restart Only**

**tuning Jenkins**

[**https://www.google.com/search?q=tuning+Jenkins&rlz=1C1SQJL\_enIN875IN875&oq=tuning+Jenkins&aqs=chrome..69i57j0l3.10778j0j7&sourceid=chrome&ie=UTF-8**](https://www.google.com/search?q=tuning+Jenkins&rlz=1C1SQJL_enIN875IN875&oq=tuning+Jenkins&aqs=chrome..69i57j0l3.10778j0j7&sourceid=chrome&ie=UTF-8)

**[root@ip-172-31-10-173 opt]# vim helloworld.java**

**[root@ip-172-31-10-173 .jenkins]# pwd**

**/root/.jenkins**

**[root@ip-172-31-10-173 FS\_Helloworld]# pwd**

**/opt/jenkins/workspace/FS\_Helloworld**

[**https://jenkins.io/doc/book/pipeline/**](https://jenkins.io/doc/book/pipeline/)

### **Declarative versus Scripted Pipeline syntax**

<https://groovy-lang.org/>

1.How to write assign Variable

2.How to write conditions

3.How to Define functions

[**https://www.tiobe.com/tiobe-index/**](https://www.tiobe.com/tiobe-index/)

[**https://www.sonarqube.org/**](https://www.sonarqube.org/)

[**https://jfrog.com/artifactory/**](https://jfrog.com/artifactory/)

<https://www.sonatype.com/>

**manage tomcat through systemd**

**[root@ip-172-31-41-83 ~]# cat /etc/fstab**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd | awk -F ":" '{print $1}’**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd | awk -F ":" '{print $1}' |sort**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd | awk -F ":" '{print $1}' |sort -nrk 1**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd | awk -F ":" '{print $3}'**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd | awk -F ":" '{print $NF}'**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd | awk -F ":" '{print $NF}'|sort**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd | awk -F ":" '{print $NF}'|sort|uniq -c**

**[root@ip-172-31-41-83 ~]# cat /etc/passwd | awk -F ":" '{print $NF}'|sort|uniq**

**[root@ip-172-31-41-83 ~]# echo $?**

**[root@ip-172-31-41-83 ~]# sed**

# HP ALM(Quality Center)

<http://www.agilemodeling.com/artifacts/userStory.htm>

<https://www.atlassian.com/agile/project-management/user-stories>

<https://www.microfocus.com/en-us/products/static-code-analysis-sast/overview>

<https://www.qualys.com/>

<https://jfrog.com/>

[**https://jfrog.com/**](https://jfrog.com/)

**[root@ip-172-31-41-83 ~]# docker pull docker.bintray.io/jfrog/artifactory-oss:latest**

**[root@ip-172-31-41-83 ~]# docker image ls**

**[root@ip-172-31-41-83 ~]# docker container run --name artifactory -d -v artifactory\_data:/var/opt/jfrog/ -p 8081:8081 docker.bintray.io/jfrog/artifactory-oss:latest**

**[root@ip-172-31-41-83 ~]# docker container start artifactory**

**[root@ip-172-31-41-83 ~]# docker container ls -a**

<http://ec2-18-189-195-10.us-east-2.compute.amazonaws.com:8081/>

username : admin

password : password

<https://www.jfrog.com/confluence/display/RTF/Working+With+Pipeline+Jobs+in+Jenkins>

<https://jenkins.io/doc/pipeline/steps/workflow-basic-steps/>

**[root@ip-172-31-6-223 sonarQube]# docker image pull sonarqube:lts**

**[root@ip-172-31-6-223 sonarQube]# docker container run -d -p 9000:9000 --name sonarqube sonarqube:lts**

**Username : admin**

**Password : admin**

<https://www.sonarlint.org/>