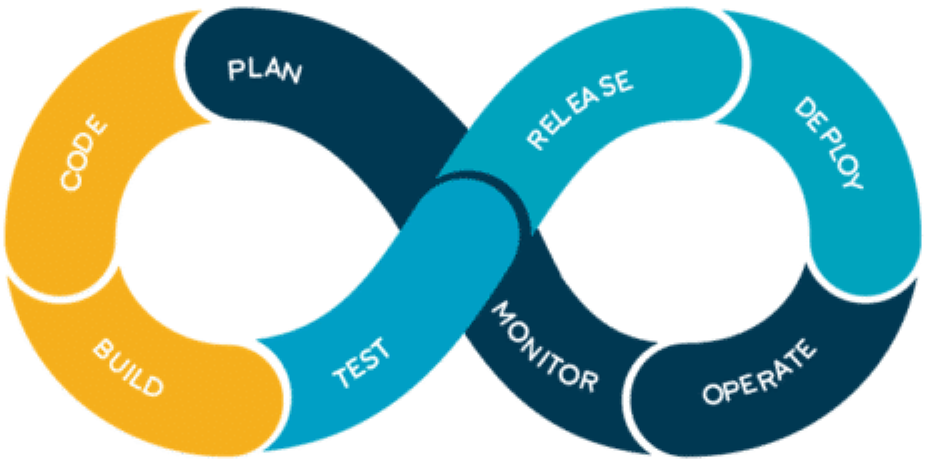
Devops



It is nothing but the practice or methodology of making ‘Developers’ and the ‘Operations’ team work together.

## Why DevOps?

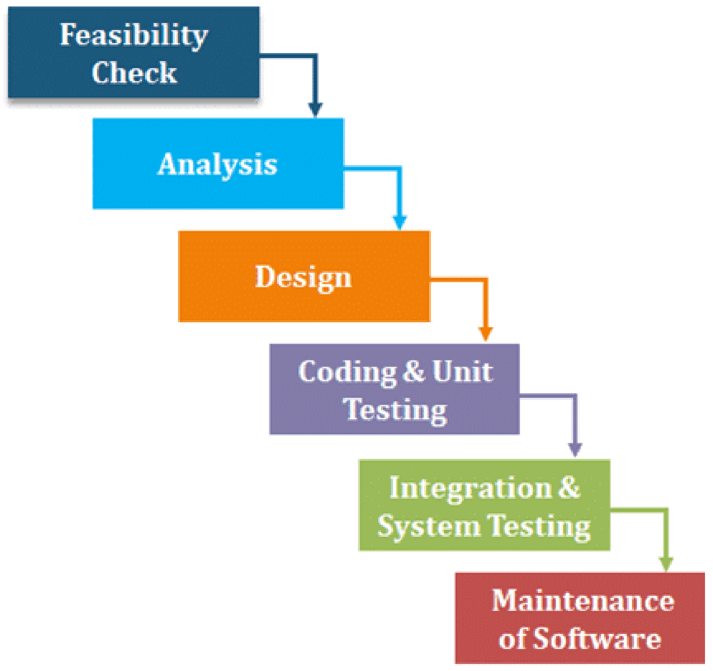
Before understanding the concepts and methodology of DevOps, we need to understand why do we even need DevOps?

Why DevOps? Why not other methods?

Before DevOps came into the picture, the Waterfall model was the earliest SDLC approach that was used for software development. This method, which was used for illustrating SDLC in a sequential flow, was considered to be reliable at first.

### The Workflow of the Waterfall Method

Let’s consider, we’re developing a software using the Waterfall method. Below are the steps that will be included in the SDLC if we’re using this method:



- Feasibility check: The feasibility phase is used for determining whether a particular approach/technique will be feasible enough for developing the software.
- Analysis of the requirements: In this phase, we need to analyze all system and software requirements from customers’ point of view and gather information about these requirements. The requirements will then be captured in the software requirement specification (SRS) document to avoid the incompleteness of the product.
- Design: The goal of this phase is to transform the requirements listed in the SRS document into an ordered structure that is appropriate for their implementation in programming.
- Coding and unit testing: The design that is created in the previous stage is supposed to get converted into the source code in this stage, and then every design module is coded and checked individually.

- Integration and system testing: After the design of each module has been coded, the integration of these modules is carried out appropriately. Then, these integrated modules are tested individually. After this, the acceptance testing is carried out in which the product is delivered to and tested by the customer for checking whether to accept it or reject it.
- Maintenance of the software: Maintenance is that phase of the software development life cycle where 60 percent of the entire effort is spent. Several maintenance operations are performed in this phase such as corrective maintenance, perfective maintenance, and adaptive maintenance, where error corrections and functionality enhancement, along with trying the software on new environments and operating systems, are done.

## Advantages

- This method is easy and simple to use
- Easy to manage due to its rigidity
- Each phase has a review process making it less vulnerable to errors
- The one-at-a-time process phases do not overlap each other
- Reliable for small projects

## Disadvantages

- While the application is in the testing stage, it is really difficult to go back and make changes relating to any issue that happened in the previous steps due to miscommunication or lack of knowledge
- It is a risky process as it is difficult to diagnose and to provide feedback
- Its main focus is to help internal teams work efficiently. It excludes end-users/clients, due to which the majority of people do not trust this methodology

- There will be delays in the testing process because this method insists teams to wait until the process reaches its 4th or 6th stage

## Agile Software Development

Agile involves an incremental approach like the Waterfall model but with an iterative perspective, along with focusing on customer feedback, incorporating small rapid changes, and speeding up releases. It basically breaks the product into smaller divisions and finally integrates them for the testing process.

Now, let’s take a look at its advantages and disadvantages.

### Advantages

- The Agile methodology considers customer feedback throughout the project, which gives enough time to the team for making decisions
- It welcomes making changes but at great expense
- It has the ability to scale
- There is continuous attention to technical excellence and good designs
- This method prioritizes and schedules the most valuable features for implementation, decreasing the risk of having unusable resources
- Small and dedicated teams are involved with a high degree of involvement and coordination

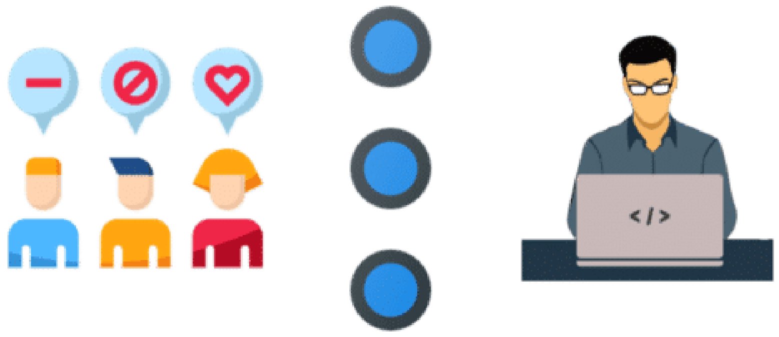
### Disadvantages

- There is less predictability in Agile
- It requires more time and commitment from every stakeholder. Testers, developers, and customers must interact with each other constantly and should agree to each other’s decisions in order to get the task done, and hence Agile is time-consuming

- Limited documentation often comes as a problem. In the case of fallbacks, there are very less detailed documents so as to cross-check
- The Agile model requires minimal planning at the beginning that makes it easier to develop the project quickly, but there is never a finite end. Due to unexpected functionalities, a clear vision of the project is not available, and mostly the stakeholders are not sure of what their final product would look like

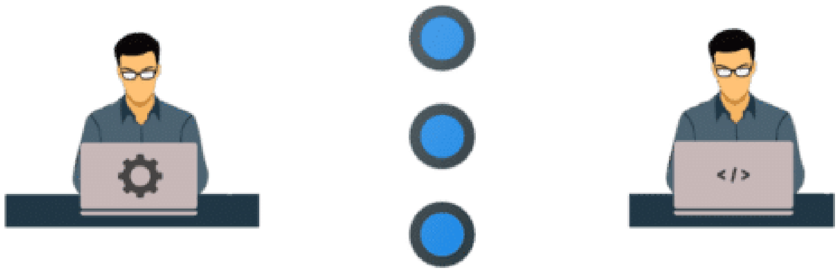
In a nutshell, when the Waterfall model failed to deliver consistency in the result, the Agile methodology came into existence. However, as discussed above, there were many disadvantages to the Agile model as well:

- In the case of the Waterfall model, there was a gap between customers' software requirements and the developers, which was overcome by Agile



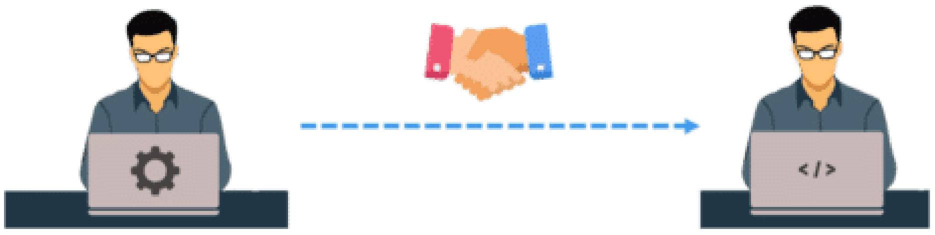
*Gaps Between Customers and Developers*

- While in the case of the Agile method, there was still a gap between the development and operations folks


*Gaps Between the Operations Team and Developers*

*How do you think it was overcome?*

It was in this scenario DevOps was introduced in order to overcome the gap between developers and the operations team.



Now, let's check out the major differences between Agile and DevOps.

## Differences Between Agile and DevOps

| Agile | DevOps |
|---|---|
| Agile majorly focuses on collaboration, customer feedback, and small rapid changes | DevOps brings development and operations teams together |
| It does not focus on automation | It focuses majorly on automation to increase efficiency while deployment |
| The development process is inherent for Agile, making it less focused on testing and implementation processes | DevOps focuses on all development, testing, and implementation phases with equal importance |
| It overcomes the gap between customers and developers | It overcomes the gap between the development and operations folks |

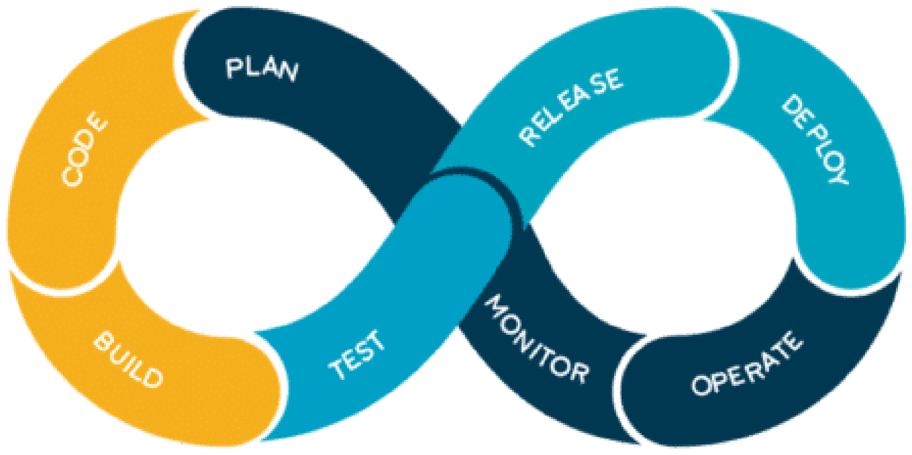
How exactly does DevOps work?

## DevOps Lifecycle

DevOps focuses on bringing all the development, operations, and IT infrastructure guys, including Developers, Testers, System Admins, and QAs, under one roof. Hence, all these people together are called DevOps Engineers.

DevOps Engineers share the end-to-end responsibility of gathering information, setting up the infrastructure, developing, testing, deploying, continuous monitoring, and fetching feedback from end-users. This process of developing, testing, deploying, and monitoring keeps on repeating for better results

You can actually figure it all out from the DevOps diagram illustrated below:



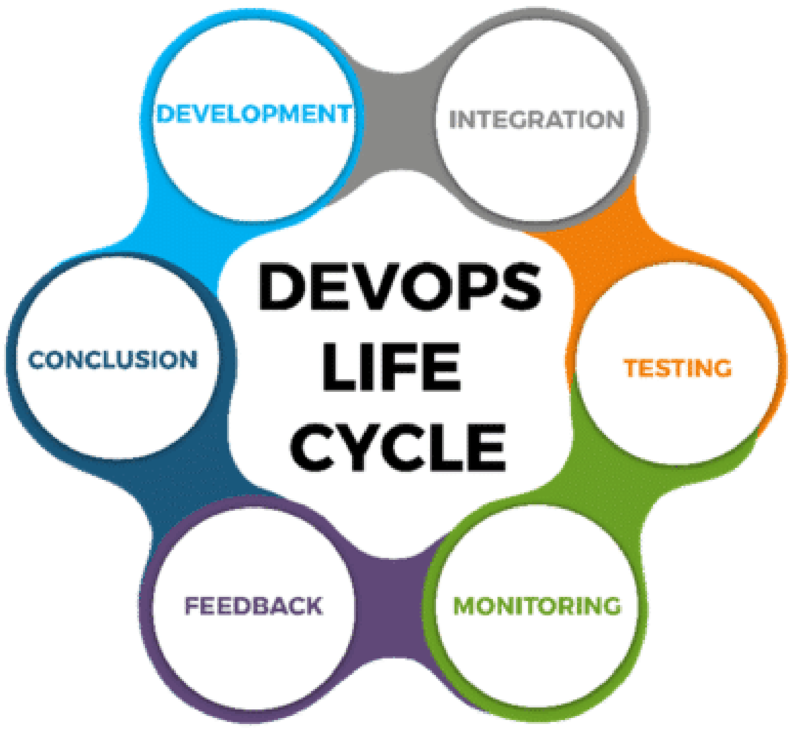
- Code: The first step in the DevOps life cycle is coding, where developers build the code on any platform
- Build: Developers build the version of their program in any extension depending upon the language they are using
- Test: For DevOps to be successful, the testing process must be automated using any automation tool like Selenium
- Release: A process for managing, planning, scheduling, and controlling the build in different environments after testing and before deployment
- Deploy: This phase gets all artifacts/code files of the application ready and deploys/executes them on the server
- Operate: The application is run after its deployment, where clients use it in real-world scenarios.
- Monitor: This phase helps in providing crucial information that basically helps ensure service uptime and optimal performance
- Plan: The planning stage gathers information from the monitoring stage and, as per feedback, implements the changes for better performance

## Different Lifecycle Stages

Now, let’s discuss the different stages in the DevOps life cycle that contributes to the consistent software development life cycle (SDLC):

- Continuous Development
- Continuous Integration
- Continuous Testing
- Continuous Monitoring
- Virtualization and Containerization

These stages are basically the aspects for achieving the DevOps goal.



Now, let’s discuss each of them in detail.

### Continuous Development

In the Waterfall model, our software product gets broken into multiple pieces or sub-parts for making the development cycles shorter, but in this stage of DevOps, the software is getting developed continuously.

- Tools used: As we code and build in this stage, we can use GIT to maintain different versions of the code. To build/package the code into an executable file, we can use a reliable tool, namely, Maven.

## Continuous Integration

In this stage, if our code is supporting a new functionality, it is integrated with the existing code continuously. As the continuous development keeps on, the existing code needs to be integrated with the latest one 'continuously,' and the changed code should ensure that there are no errors in the current environment for it to work smoothly.

- Tools used: Jenkins is the tool that is used for continuous integration. Here, we can pull the latest code from the GIT repository, of which we can produce the build and deploy it on the test or the production server.

## Continuous Testing

In this stage, our developed software is getting tested continuously to detect bugs using several automation tools.

- Tools used: For the QA/Testing purpose, we can use many automated tools, and the tool used widely for automation testing is Selenium as it lets QAs test the codes in parallel to ensure that there is no error, incompetencies, or flaws in the software.

## Continuous Monitoring

It is a very crucial part of the DevOps life cycle where it provides important information that helps us ensure service uptime and optimal performance. The operations team gets results from reliable monitoring tools to detect and fix the bugs/flaws in the application.

- Tools used: Several tools such as Nagios, Splunk, ELK Stack, and Sensu are used for monitoring the application. They help us monitor our application and servers closely to check their health and whether they are operating actively. Any major issue detected by these tools is forwarded to the development team to fix in the continuous development phase.

Let’s now talk about some of the major DevOps tools in detail.

## DevOps Tools

- Puppet: Puppet is one of the widely-used DevOps tools. It allows delivering and releasing technology changes quickly and frequently. It has features of versioning, automated testing, and continuous delivery.
- Docker: Docker is a high-end DevOps tool that allows building, shipping, and running distributed applications on multiple systems. It helps assemble the applications quickly and is typically suitable for container management.
- Jenkins: Jenkins is one of the most popular DevOps tools that allow monitoring of the execution of repeated jobs. Apart from this, Jenkins lets us integrate the changes and access the results easily and quickly.
- Ansible: This tool helps automate the entire life cycle of an application, and it manages complicated deployments and enhances productivity.
- Nagios: This DevOps tool helps monitor the IT infrastructure. It is capable of determining errors and rectifying them with the help of the standard network, server, and log monitoring systems.

## GIt

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

Continious changes made find out changes and who made change on code it's difficult to maintain previous times.to solve this one git had born.

Source code management(scs)
Version code management(vcs)
Ditstribution control system
Market 90 % of place ocuuping by the git.

Install:
Git environment is not important concept is important.
Download windows.
Google scm git
U will get official git website.
Download for windows/mac
Install that dmg or application
If u r using windows after run that application just .

For windows:Left click on u can see git bash click on that,
If u r using mac open terminal
Type
git –version

if it will show the version means u will successfully installed git.

Note: if u hav a windows system to shift from one drive to other
Type  cd c:

## Now installation is done:

Apart from git all systems are following "centralized system:"
Here we have main system one and each and everytime we are connected to that center system.
1000 action every time interact with server
slow process
need broadband connection
user collobarartion not possible
But git is an distribution model .

Distribution also like centralized system but
Here every system will create a repository apart from main repository
Now speed is high.
It will store in local repository.
User collaboration possible.
After u work is done u can push that code to once.

Git Terminologies

Client,server,branch,master,

Before we dive into git practical check git install on u r local system or not
Coomand: git --version

Creating a repository:
Go to any path and make a directory like mkdir devops and cd devops
Create files using nano comand
 nano 1.txt
nano 2.txt

For nano saving a file is ctrl +x and press enter y + enter

For git init  to intialize a git repository
Initialized empty Git repository in /Users/mac/Desktop/devops/.git/
Know status of file :git status (red is working if green means stagging area and )
Adding a file to stagging ares : git add . (here . means all files it will take)
Saving this file in git file system
 Git commit -m"i am saving my files"
After all check the status
Git status  will be like this:
On branch master
nothing to commit, working tree clean

## Creating a github or pushing from git to git hub:

Login in ur git hub account and go this url :https://github.com/new
Give name and create a repository;
Git remote add origion reponame

```
git remote add origin https://github.com/shivasingam111/Devops.git
```

To push local github server:
git push origin master
Give username
Username for 'https://github.com': shivasingam111@gmail.com
Give password
Password for 'https://shivasingam111@gmail.com@github.com':

If u get status like this means it s success:
master -> master

## Clone reposotory from gitHub to local:

Create a directory;
Mkdir devclone cd movethatfolder
To clone url from server to local :git clone url
git clone https://github.com/shivasingam111/Devops.git

Now in clone folder create one file
nano file4.txt
git add .
git commit -m"cloning data storing in sever"

To push directly to server
git push bcoz when we were clonning automatically the region will add we dont need add again region.

Git Pull

If Multiple developer are working on same repo now i want latest created or updated file means now i need to do pull the code from server

git pull origin master
Diffrence b/w git clone and git pull:
Git clone : if we want copy the whole repo that situation we can use git clone
Git pull: when we are working on same repo when we want the updated file then that time we can use the git pull.

## Git branch:

To create a branch : git branch branchname
And " git branch" command used for the branches what are present in repo
And " git checkout branchname" used for switching from one branch to other branch
And " git branch -D branchname " used for the Deleting a branch
Note: "when the branch is created the files which are present in master it will all are coming to subbranch"

Now create a file in sub branch and commit the file it will be there in subbranch only it wont appear in master or other branches

"Whenever u r creating a file and not commit that file it will not store any branch in will save in working zone only due to that it will visible to all the branches before commiting a file"

If u commit means it will store in that branch only that the reason it will visible in that branch only.

To push branch to github: git push origin gbranch
git push origin branchname

Now our local subbranch and its data will store on severt git hub account

GIT LOG(helaps to see the history of repository)
Command is " git log "

Note: here we are seeing the only logs of whats ever the branch we are in currently either it may be sub or master branch.

### Git Stash:

Its used for multi tasking .

Ex: i am in another branch i am in midlle of the file coding work if i directly checkout that branch and i shifted to any other branch or master the files what we changed in other branch that code will reflect on other branches.

So to overcome above issue we use stash concept
By using this whereever branch we r in middle of the work we can stash or temp save that file and revert back to other branches that data will be wont visible .bcoz we used stash

“ git stash “ to stash a repo
“ git stash pop “ to comeback the previous stash code place.

After save and commit a file then task is done.

## Git Revert:

In git log report whatever the position u want revertback we can use the git log take commitid

“ git revert commitid paste”
After typing this command that file will revert from there

Git Diff:
It s used to find outt the difference between two files. Or two commits

# Ansible

----------------------------------------------------------------------------------------------------------------
Ansible Real time redhat theory with practical
Login to aws console.
Launch 2 ec2 REDHAT linux instances and security groups allow all icmp and tcp .launch 2 instances.
And give 2 instance name as ex: ansible server or master 2.ansible node server1.

sudo su
vi /etc/hosts
private ip address  server1.com server1
Private ip address node1.com node1

Do the same operations in both server and node.

After going to aws console check server and node actions click on reboot.
sudo su

cd
pwd(now u r in root directory to install softwares)
yum install wget   (its for download web packages in linux)
yum info wget     (info command used to know complete information about that package)
cat /etc/redhat-release (to find version of redhat)
rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm (its for Upgrade package we downloaded that epel repo)
yum install epel-release -y   --- its for downloading the epel package in our linux machine
yum install python  (ansible develop by python so we need python if python not available means we will go with python 3)
If python not install means it will give suggestions like this below
There are following alternatives for "python": python2, python36, python38

Now we go with python3.
yum install git python3 python3-pip ansible openssl -y  --- using this command we at a time installing the python and ansible)

Now it will download all the packeges.
To check python version --- python3 --version
ansible --version (for ansible shows the version)
from above steps the ansible configuration is done.
Now we have to make ssh connection between the node and server.
cd .ssh
ssh-keygen (it will generate keys)
before generate ssh-keygen go to
vi /etc/ssh/sshd_config
In this file
Permit root login will be yes &
PasswordAuthentication yes
systemctl restart sshd (its used for the restart a ssh)
Now in the position of .ssh type ssh-keygen
Ls now u wil see the public key and private key.
vi public key (copy that one)
After login into node server
Sudo su
Cd
Cd .ssh
vi authorized_keys
To go for end of ssh file type shift +a
Paste the public key of ansible server in node..
Save the file.
In node go
vi /etc/ssh/sshd_config
In this file
Permit root login will be yes &
PasswordAuthentication yes

systemctl restart sshd (its used for the restart a ssh)

Afeter ssh restart on node go to main server console.
and sudo su
ssh give here nodeserver ipaddress.
type yes
Now it will login on node server.
-----------------------------------------------------------for this succesfully we made ssh connection between server to node-------------
Master server inventory containn all the ip address of nodes.
cd /etc/ansible/
vi hosts
[webserver]
172.31.38.220 (its is a private ip address of node like how many nodes present give step by step)
Adhoc commands (quick task if we wanna perform)

```
ex:   cd /etc/ansible/
     ansible -m ping all
172.31.38.220 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```

Like this we will ping the all node from server using adhoc commands.

## AdHoc Coomands

***ansible private ip of node -m yum -a "name=httpd state =installed"

```
 ansible webserver -m yum -a 'name=httpd state=installed'
   ansible webserver -m copy -a "src=/etc/ansible/index.html dest=/var/www/html"
  ansible webserver -m service -a 'name=httpd state=started'
```

## Ansible PlayBooks

After create a folder inside create a sample index.html file.
Write something on this file

```
mkdir ansible playbooks
touch index.html
```

After create copysample.yml file

```
---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
    - name: install httpd
      yum: name=httpd state=installed
    - name: copy index.html file
      copy: src=index.html dest=/var/www/html
    - name: restart service
      service: name=httpd state=started
```

Save file and check the syntax using below command

ansible-playbook ymlfilename --syntax
If no error means
Ansible-playbook ymlfilename
Then it will show ok.

## Ansible variables:

3 sections
Target section
Variable section
Task section

Example:

```
[root@ip-172-31-41-84 playbooks]# vi variable.yml
```

vi variable.yml

```
---
- hosts: webserver
  remote_user: root
  become: yes
  vars:
   pkg: httpd
  tasks:
   - name: install httpd
     yum: name={{pkg}} state=installed
   - name: creates index.html file
     copy: src=index.html dest=/var/www/html
   - name: restart httpd
     service: name={{pkg}} state=started
```

Save
ansible-playbook variable.yml --syntax (if no errors means run below code)
ansible-playbook variable.ym
l

## Ansible files:

Previously we store variables inside a playbook but now we are storing variables in a file.

Create fileabc.yml file paste below code on this.

```
pkg: httpd
```

After create files.yml (paste below code)

```
---
- hosts: webserver
  remote_user: root
  become: yes
  vars_files:
   - fileabc.yml
  tasks:
   - name: install httpd
     yum: name={{pkg}} state=installed
   - name: creates index.html file
     copy: src=index.html dest=/var/www/html
   - name: restart httpd
     service: name={{pkg}} state=started
```

## Checking syntax:

```
ansible-playbook files.yml --syntax
ansible-playbook files.yml
```

Ansible Handler

It will notify when something is happen

Ex:if any changes is there means we will restart the server otherwise no

```
vi handler.yml
---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
    - name: install httpd
      yum: name=httpd state=installed
    - name: copy index.html file
      copy: src=index.html dest=/var/www/html
      notify: restart httpd
    - name: restart service
      service: name=httpd state=started
  handlers:
    - name : restart the server
      service: name=httpd state=restarted
Note :

Must u have to check the syntax:
ansible-playbook handler.yml --syntax
ansible-playbook handler.yml
```

Ignore errors

```
Create a vi ignore.yml file
Below code i am giving wrong name 4 index.html
---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
    - name: install httpd
      yum: name=httpd state=installed
    - name: copy index.html file
      copy: src=indgjx.html dest=/var/www/html
      Ignore_errors: yes
    - name: restart service
      service: name=httpd state=started
```

Check synax and run

## Ansible Template

If u want get dynamic data from all the nodes we use template.
Is a file which contain configuration parameters dynamic value.

It will follow jinza tool extension which support python

```
vi template.yml
---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
   - name: create xyz file
     template: src=xyz.j2 dest=/root/xyz
```

vi xyz.j2 (paste below code on that file)

```
system name is {{ansible_nodename}}
```

By using this we will get all our node names

```
ansible-playbook template.yml --syntax
ansible-playbook template.yml
```

If its success means login to u r node.

```
sudo su
Cd
Ls
```
now check below the xyz file is created.

```
cat xyz
system name is ip-172-31-38-220.ap-south-1.compute.internal
```

Above its displayning the node name

## Dry Run:

Before we run playbook in production environment we will check fine or not

It will explain what will happen if u execute this playbook
Change index.html
And run
Ansible-playbook handler.yml --check
Now it will only display the flow what will happen if we execute this playbook.
For that testing or getting information purpose we are using dryrun.

## Loops playbook ansible

For suppose i want install multi packages in nodes or at a time we install 100 packages at a time.

```
vi loop.yml
---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
   - name: installing multiple packages
     yum: name={{item}} state=installed
     with_items:
        - httpd
        - curl
        - wget
```

```
ansible-playbook loop.yml --syntax
```

Above if we observe withitems place we need to give whatever the packages we need to install.

## Ansible Security or vault

```
---
- hosts: webserver
  remote_user: root
  become: yes
  vars:
   password: raj
  tasks:
   - name: install httpd
```

```
    yum: name=httpd state=installed
  - name: creates index.html file
    copy: src=index.html dest=/var/www/html
  - name: restart httpd
    service: name=httpd state=started
```

To run this

```
ansible-vault encrypt vault.yml
```

Above command will convert the plain text to other diffrent hidden format

It will ask u r playbook password give it 2 times now that file will be encrypted
Now if u type cat password.yml its not in human readable launguage.

```
ansible-playbook vault.yml --ask-vault-pass
```

Above command is asked before ur playbook will execute

```
ansible-vault edit vault.yml (coomand for edit the encrypt file)
ansible-vault rekey vault.yml (we can modify the password)
ansible-vault decrypt vault.yml (we can decrypt the ansible file)
```

## Register

Register module we used to capture the output

Debug is used to display the output

Vi register.yml

```
---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
   - name: install httpd
     yum: name=httpd state=installed
   - name: creates index.html file
     copy: src=index.html dest=/var/www/html
   - name: restart httpd
     service: name=httpd state=started
     register: output
   - debug:
       msg: "{{output}}"
```

```
ansible-playbook register.yml --syntax
ansible-playbook register.yml
```

## Tags:

In play book if u want perform a particular task for ex i have 3 task i want to run 2 task that time i just give 2 tagname is enough to run that task.

```
vi tags.yml
---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
   - name: install httpd
     yum: name=httpd state=installed
     tags:
      - install
   - name: creates index.html file
     copy: src=index.html dest=/var/www/html
     tags:
      - configure
   - name: restart httpd
     service: name=httpd state=started
     tags:
      - service
save:
```

Execute below commands:

Method1:

```
ansible-playbook tags.yml --tag  "install,configure"
ansible-playbook tags.yml --skip-tags "install,configure"
```

Method2:

```
ansible-playbook tags.yml --start-at-task="creates index.html file"
```

Method3:

```
ansible-playbook tags.yml --step  (this step ask permission 2 execute)
```

## Ansible Include:

By using this we are including the yaml file or playbooks

```
vi include.yml

---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
   - include: pavani.yml
   - include: config.yml
   - include: service.yml
```

## Ansible Role Concept

Role is like a section can not be executed we can use in inside the playbook.

There is noway to execute role we can execute inside playbook

Roles having the below section
Tasks
Handlers
Defaults
Vars
Files
Templates
Meta

Cd etc
Cd ansible
cd roles
check any roles are there are not
To create a roles
ansibel-galaxy init webserver(or give any name) + enter
We will get webserver was created successufully. Galsxy is like repository we will find all roles
Now ls -la we will see weberserver
Cd webserver
type ls we will see all below sections

Tasks
Handlers
Defaults
Vars
Files
Templates
Meta

Cd tasks
Ls here main.yml is the default one of role
vi main.yml
Remove all the data in that file paste below code

```
  - name: install httpd
     yum: name=httpd state=installed
     tags:
      - install
   - name: creates index.html file
```

```
     copy: src=index.html dest=/var/www/html
     tags:
      - configure
   - name: restart httpd
     service: name=httpd state=started
Save
Cd ..
Cd handlers
vi main.yml
- name: restart service
  service: name=httpd state=restarted
Save
Cd ..
Cd files
vi index.html

Hai all

Save

Now go to this path
Cd /etc/ansible/playbook
vi role.yml paste below code

---
- hosts: webserver
  remote_user: root
  become: yes
  roles:
   - webserver(here webserver is the role name)
Save.
```

Before execute role.yml if in ur playbook u have index.html remove this file .the reason because roles already we have index.html i.e it will through error.

Now run

```
ansible-playbook role.yml --syntax
ansible-playbook role.yml
```

Ansible tower is a gui
Its a very high price

---

Red Hat Ansible Configuration Commandwise linux terminal:

Login in with 2 server
vi /etc/hosts
private ipaddress  server1.com serever1
Private ipaddress node1.com node1

```
 1  cd /etc
    2  nano hosts
    3  ls
    4  nano hosts
    5  vi hosts
    6  cat hosts
    7  pwd
    8  cd
    9  exit
   10  pwd
   11  cd
   12  pwd
   13  cd /
   14  cd
   15  ls
   16  cd /
   17  ls
   18  cd root
   19  pwd
   20  clear
   21  clear
   22  exit
   23  clear
   24  cd
   25  yum install wget
   26  yum -y install wget
   27  yum search wget
   28  yum info wget
   29  clear
   30  cat /etc/redhat-release
   31  cat /etc/centos-release
   32  clear
   34  rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
   35  yum install epel-release -y
   36  yum install python
   37  yum install git python3 python3-pip ansible openssl -y
   38  python
   39  python --version
   40  python3
   41  clear
   42  ansible --version
   44.passwd root
   45  clear
   46  cd
   47  pwd
   48  ls -a
   49  pwd
```

```
50  cd .ssh
51  ls
52  ssh-keygen
53  ls
54  vi /etc/ssh/sshd_config
55  systemctl restart sshd
56  cat id_rsa.pub
57  cat id_rsa.pub
58  ssh 172.31.46.46
59  clear
60  history
61  clear
62  cd
63  cd /etc/ansible/
64  ls
65  vi hosts
66  clear
67  ansible -m ping all
68  ansible -m ping webserver
```

Adhoc commands (quick task if we wanna perform)
ex:   cd /etc/ansible/
      ansible -m ping all
I want to install httpd package using adhoc command
For reference of adhoc commands below link is useful.
https://www.middlewareinventory.com/blog/ansible-ad-hoc-commands/#Example_13_ad_hoc_command_to_Install_a_package_using_yum_command

ansible private ip of node -m yum -a "name=httpd state =installed"

```
ansible 172.31.46.46  -m service -a "name=httpd state=installed"
```

To start httpd using adhoc

```
ansible 172.31.46.46  -m service -a "name=httpd state=started"
```

Now from server we are creating index.html file in node
ansible nodeip -m copy -a 'src=/etc/ansible/index.html dest=/var/www/html'
ansible webserver -m copy -a 'src=/etc/ansible/index.html dest=/var/www/html'

```
ansible 172.31.46.46  -m service -a "name=httpd state=started"
```

Above we did install httpd and we create a index.html from server using ansible adhoc

After create a folder inside create a sample index.html file.
Write something on this file

Create one more file ends with yml
Ex:webserver.yml

```
---
- hosts: webserver
  remote_user: root
  become: yes
  tasks:
    - name: install httpd
      yum: name=httpd state=installed
    - name: copy index.html file
      copy: src=index.html dest=/var/www/html
    - name: restart service
      service: name=httpd state=started
```

After creating yml file we need to check the ansible syntax
ansible-playbook webserver.yml --syntx
If u dont have any errors go with
ansible-playbook webserver.yml

Note:without index.html if u mention it will through an error

## Ansible variables:

Play book having 3 section
Target section
Variable section
Task section

```
---
- hosts: webserver
  remote_user: root
  become: yes
  vars:
   pkg: httpd
  tasks:
    - name: install httpd
      yum: name=((pkg)) state=installed
    - name: copy index.html file
      copy: src=index.html dest=/var/www/html
    - name: restart service
      service: name=((pkg)) state=started
```

Now again start httpd services

-----------------------------------------------------------------------------------------------

Aws linux2 ansible configuration

Login to aws console.
Launch 2 ec2 linux instamces and security groups allow all icmp and tcp .launch 2 instances.
And give 2 instance name as ex: ansible serevr or master 2.ansible node server1.
Login into putty or mac console.
For reference install ansible & python if its not there
https://medium.com/@khandelwal12nidhi/ansible-setup-on-aws-ec2-instance-d83fac41fcc8
Login to master ansible server and sudo su

Note: for ansible python version must be 2.7 and higher
For more ansible infomartion go for https://docs.ansible.com/

Follow the below procedure to install ansible.

```
 1  yum -y install wget
 2  wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
 3  ls
 4  rpm -ivh epel-release-latest-7.noarch.rpm
 5  yum -y update
 6  yum -y install python ansible python-pip open-ssl
 7  python --version
 8  ansible --version
 9  sudo yum-config-manager --enable epel
10  yum repolist
11  sudo yum-config-manager --enable epel
12   yum install ansible
13   ansible --version
```

Above lines description:--------------------above 13 lines description which line whar doing----
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
Note :epel repo nothing but the where we will get all latest package to download.

Its in rpm fomat we need to use that one then type below command
rpm -ivh rpm file name
Ex: rpm -ivh epel-release-latest-7.noarch.rpm
Now we need to update bcoz whenever we download from web we need to update it.
yum -y update
yum -y install python ansible python-pip open-ssl

Note :epel repo nothing but the where we will get all latest package to download.

Its in rpm fomat we need to use that one then type below command

rpm -ivh rpm file name
Ex: rpm -ivh epel-release-latest-7.noarch.rpm

In epple repository ansible and python all packages will be there.its a free repository.

Now we need to update bcoz whenever we download from web we need to update it.
yum -y update

Now we observe python packge will update bcoz ansible develop by python.
Finally it will show complete after.

Now we downloaded the packages now we have to install on our machines.

yum -y install python ansible python-pip open-ssl

After it will show complete.

After install check ansible version install or not.
ansible --version
If ansible version not showing means follow below procedure.

```
sudo yum-config-manager --enable epel
 yum repolist
 sudo yum-config-manager --enable epel
 yum install ansible
 ansible --version
```

Reference :
https://medium.com/@khandelwal12nidhi/ansible-setup-on-aws-ec2-instance-d83fac41fcc8

whereis ansible
The above command will show the path of the ansible.

Now we have successfully installed ansible.
-------------------------------------------------------------------------------------------------
Login into master server of ansible
Sudo su
Create a user
useradd ansadmin
passwd ansadmin
give password like:ansadmin@123

passwd: all authentication tokens updated successfully.

After visudo
last line paste this code (adding user )
Below of #includedir /etc/sudoers.d  copy below code
ansadmin ALL=(ALL)       NOPASSWD: ALL

Then login into all nodes follow the same above procedure create ansadminuser and give permission using visudo

Sudo su
Create a user
useradd ansadmin
passwd ansadmin
give password like:ansadmin@123
passwd: all authentication tokens updated successfully.

After visudo
Insertmode and save the file
last line paste this code (adding user )
Below of #includedir /etc/sudoers.d  copy below code
ansadmin ALL=(ALL)       NOPASSWD: ALL

Go to cd /etc/ssh
nano sshd_config
# EC2 uses keys for remote access
PasswordAuthentication yes
Save and contine
service sshd restart
Login in to node

Go to cd /etc/ssh
nano sshd_config
# EC2 uses keys for remote access
PasswordAuthentication yes
Save and contine
service sshd restart

Go to Master ansible server:
sudo su ansadmin
Cd
ls -a
Here we dont have .ssh we need to create ssh-keygen
ssh-keygen
And type 3 times enter.
Now we have ls -a .ssh file we have

Cd .ssh
Inside we have public and private key

```
[ansadmin@ip-172-31-13-245 ~]$ cd .ssh
[ansadmin@ip-172-31-13-245 .ssh]$ ls
```

ssh-copy-id 13.127.180.4(ssh-copy-id node public ipaddress)
Type yes
Give ansadmin password.

Now status will be like this

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

ssh 13.127.180.4 (tologin into node server)

Now it will show

```
Now try logging into the machine, with:   "ssh '13.127.180.4'"
and check to make sure that only the key(s) you wanted were added.

[ansadmin@ip-172-31-13-245 .ssh]$ ssh 13.127.180.4

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
--------
```

Exit
Cd (form ansible server)
-- cd /etc/anisible
sudo vi hosts
Paste the node public ip address at top of file and save it

To check all conectivity
ansible all -m ping
Now we can see the status of node files like below

```
13.127.180.4 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

--------------------passwordless authentication done to node from server------------------------------------
------------------------------------------------------------------------------------------------

## Now we have to configure inventory files

cd /etc
Now in etc directory we have ansible folder is there.
cd ansible
Now we can see this

```
ansible.cfg  hosts  roles
```

Now go to hosts

This hosts is for server level
Vi hosts
# Ex 3: A collection of database servers in the 'dbservers' group
[webserver]
172.31.1.188  (it is a node server 1 private ip address)
Paste code as it is.
Save

Now we have to establish ansible connection

In ansible we are making connection between server to node using ssh (so that we need to change on file)

Type cd
Ls -a (here u can see that .ssh file)
Cd .ssh

Here we have authorization key will be there (remember after key will generate in ansible it will change)

Cd
Cd /etc

Cd ssh
ls -al
Here u can see
sshd_config

Vi sshd_config

Under authentication
Permit root login (before remove the #)
PermitRootLogin yes

And
# EC2 uses keys for remote access
PasswordAuthentication yes
Default password auth wiil be no make it as a yes.

Save the file and exit

Cd..
Cd ansible
Type  “ssh-keygen”

And enter 3 times enter thats it the key will generate to check key
Cd
Cd .ssh/
path(/home/ec2-user/.ssh)
Ls

```
authorized_keys  id_rsa  id_rsa.pub
```

Now u can see apart from auth keys we have id_rsa plublic key and private key.

The key have in encrypted format

## Now we have to made ssh connection in node1 also.

Login node
Sudo su
Cd /etc

```
 cd /etc
      ls
      ls -la | grep ssh
     cd ssh
      ls -la
      vi sshd_config
```

Permit root login (before remove the #)

```
PermitRootLogin yes
```

Restart a ssh server
** https://www.cyberciti.biz/faq/howto-restart-ssh/
And

```
# EC2 uses keys for remote access
PasswordAuthentication yes
```

Default password auth wiil be no make it as a yes.

Save and exit the file

Now login in to u r master server and cd .ssh ls and id_public key copy the code and
Again login in node server .ssh ls vi authentication keys fle open
And paste the public key of masyter and :wq and save

Note:by using this server will comminicate to node.
------------------------------------------------------------------------------------------------
Ping to server to node code :aws ansible
After sudo su.
cd /etc   after cd vi hosts
Go to insert mode type like this.

server1 private ipaddress sevrer1.com server1
Server2 node ipaddress   server2node.com server2

Ex: 11.657.545 sevrer1.com server1
      11.347.45 server2node.com server2
Save it after go to login with other server2 putty login.
Sudo su and cd etc/  and vi hosts paste the above code & save it.

      11.657.545 sevrer1.com server1
      11.347.45 server2node.com server2
     Save and continue.
GO TO AWS CONSOLE CHOOSE 2 SERVERS ACTION→ REBOOT THE SERVER.
Login server1 and type.
Ping server2 (if its ping means successufully we hosted the servers)...

---------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------

# Docker

Docker is used for containerization

Aws launch console using redhat enterprise edition above 6 .

```
sudo su
Cd
yum update -y
yum install -y https://download.docker.com/linux/centos/7/x86_64/stable/Packages/containerd.io-1.2.6-3.3.el7.x86_64.rpm
yum install docker-ce
 docker -v
 systemctl status docker
  sudo systemctl enable --now docker
   systemctl is-active docker
   systemctl is-enabled docker
    systemctl status docker
```

------------installation complete---

Docker :installation 2 nd way and easy way:

https://www.linuxtechi.com/install-docker-ce-centos-8-rhel-8/

From bro

```
docker run -d --name myhttpd -p 80:80 httpd
```

```
  111  docker ps -a
  112  ifconfig
  113  history
  114  docker search httpd
  115  docker search git
  116  docker pull alpine/git
  117  docker images
   118  netstat -tunlp
```

----------------------------------
To start docker
systemctl start docker
systemctl status docker

Now we have to create an image
We have 3 ways to create a image

1. from dockerhub
2. from existing images / containers
3. From files

тo check images we can use
 docker images
To check container
docker ps -a

Note : if u want to remove image first u need to remove the container and next u have 2 remove the image

1)Now we are going to pull the image from docker hub:
docker pull image name
docker pull centos:6

Now if u check docker images means the images will there in images
docker images

To remove image docker remove image name or id
docker rmi imgid
2.docker search imagename
ex: docker search clock
choose highest start name and download the image

docker pull imagename

docker images

Now after downloading images from git hub if we want to run use this command

docker run -it --name myname hereimagenamepsteit
I --input
T - terminal
--name -- mycutom name

Now it will run

3)Now we have to create a container to store all images (below command we create a container and logged in)
docker run -it --name nameofimg image id
docker run -it --name centos d0343456

Now check u ls -la
Ur in inside of container

** docker ps -a will show u the all containers but
docker ps will show u the only running containers
to exit from the container we have to use exit command

docker start containerid
docker stop containerid
docker kill containerid
Procedure:

docker images
ps -a
docker start containerid

Now to check which container is running
docker ps

Creating image from container
docker commit containerid

---------------

Docker port fotrwarding

Login into container

docker run it --name dammyname containername

yum install httpd

After complete

Now we are creating image using above container httpd

docker commit conainerid imagename

-------------------------------------------------------

Docker container lifecycle:

Docker hub ---><----pull/push ---> docker engine → Docker Images →

1.run(container)

2.stop

3.delete.

Common docker operations:

Make yourself sudo su

docker --version

docker pull ubuntu

docker images

docker run -it -d ubuntu(image name)

docker ps

docker ps -a

Working with containers

docker exec -it ec7424fa964c(container id) bash

Above command runs after u will in inside container

Inside container i am updating or doing anything

apt- get update

Note: docker not installed in inside container.

exit (used to exit from container)

sudo docker stop containerid

sudo docker kill containerid

docker rmi imageid
docker rm -f containerids

Creating a docker Hub account:

And comeback to terminal
Execute docker container

docker exec -it containerid bash
ls
Now u r inside a container
Make a folder
Mkdir app
Cd app
exit
Saving Changes inside the container
docker commit containerid nameforimagenew

docker images
docker run -it -d imagename
Docker exec runidabove bash

Ls check previous container will be there.

-----------
Run ubuntu container instal apache and ubuntu container with apache
clear / exit

To remove all containers at a time
docker rm -f $(sudo docker ps -a -q)

docker run -it -d ubuntu
docker exec -it aab4e338d2121e1d801174e9cb6b9e0ad609f921402710e886f11402ab7fb256 bash
apt-get install  apache2
service apache2 status
service apache2 start

Exit and save this container
Port forwarding:

sudo docker run -it -p 82:80  -d newapche
docker exec -it 694dbc9cfd33 bash
service apche2 start

Go to aws console copy ip and end of url :82

[http://13.126.249.211:82/](http://13.126.249.211:82/)
Now apache installing inside our container.

For testing stop container
docker stop container id

Pushing to docker hub:

docker login
  docker login
Give username and password

docker push imagename
docker push newapche

Go to hub check u r repo

Docker file
From
Add
Run
Cmd
Entrypoint
Env

From Ubuntu
RUN apt-get update
ADD . /var/www//html/

Goto putty ip

Mkdir dockerfiles
Cd dockerfiles
vi  dockerfile

```
FROM ubuntu
MAINTAINER Romin Irani
RUN apt-get update
RUN apt-get install -y nginx
COPY index.html /usr/share/nginx/html/
ENTRYPOINT [“/usr/sbin/nginx”,”-g”,”daemon off;”]
EXPOSE 80
```

docker build . t new_dockerfile .

Docker Swarm:

Kubernates:

Container Management tool in cluster system.
https://github.com/ValaxyTech/DevOpsDemos/blob/master/Kubernetes/k8s-setup.md

In security groups allow 22,80,8080,443,
------------------------------------------------------------------------------------------------------------------------------------------------------------

# Jenkins

Go to aws console and take redhat linux with security group :8080 and launch the instance.

```
 yum update -y
  sudo yum install java-1.8.0-openjdk-devel
  which java
  cd /bin
  ls
 which java
 ls -la | grep jav
 cd
 java -version
curl --silent --location http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo | sudo tee /etc/yum.repos.d/jenkins.repo
 sudo rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key
 yum -y install jenkins
 systemctl status jenkins
 systemctl start jenkins
 systemctl status jenkins
 systemctl enable jenkins
 cat /var/lib/jenkins/secrets/initialAdminPassword
```

Type password
And next click  on installed plugins.

Create username raj
Password : raj@95157

Click on save and continue.

Click on start using jenkins ready

** to remove jenkins
sudo yum remove jenkins

--------------
Jenkins default path

```
cd /var/lib/jenkins/
```

---------------------instalation completed--------------------------------

In dashboard → new item we can create a job inside a job u can configure git also
Job is nothing but a task

Note: vvimp: default jenkins path
/var/lib
Ls -la u can see jenkins folder over there
cd jenkins
Here we have all plugins ,nodes,updates,etc,

To check available plugins

Manage jenkins → manage plugins → available plugins.

In advanced we can upload the plugin from outside download
Note: never use advanced bcoz ur jenkins version and plugin install must be match otherwise we may face issue.

Before direct upload plugin install in sandbox and test it if everything is fine we will upload main server.

Lets practical:
Create a new item → jobname + freestyle project → ok
Now we see general and other tabs
General:
Description about job
Discard old bill -> to remove old bill
Github project → to have github project:

Source code management:
Here we can add source code management .

Build:
Chhose execute shell → in command type date
Apply & save

Go to dashboard click on build now.

Build History we can see build numbers : show green success or fail red

Integrating github to jenkins

Go to terminal check git is there or not

*** yum -y install git

manage jenkins -> manage plugins → install (check git installed or not) -->here git plugin will be there default

Create one job .
Free style + name is git job → ok + Source code management -->click on git → paste the github url repo + apply and save → build now → click on success we can see success message..

Now code is there we have to build the code.

We can build the code using maven.

By using maven we can conver source code into object code.
Dependency management tool (it will include all dependencies which is needed)
Documentation tool
Project management tool

Configure maven in jenkins

Manage jenkins → Global tool configuration → down side above docker maven is there → click on add maven → give name maven and click on install automatically → choose version → apply and save .

To check maven installed or not
manage jenkins -> manage plugins → available (Maven Integration) file will there -> check on this one → click on install without restart.

Now go and check installed its there or not
manage jenkins -> manage plugins → installed .(Maven Integration plugin) will be there.

Create one more job → maven name freestyle + ok → source code → git paste github url repo → Build → click on invoke top level maven targets → maven version choose maven name & goal type maven install + apply and save → build.

```
cd /opt

  wget https://www-eu.apache.org/dist/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
   sudo tar xzf apache-maven-3.6.3-bin.tar.gz
   sudo ln -s apache-maven-3.6.3 maven
   sudo vi /etc/profile.d/maven.sh
   source /etc/profile.d/maven.sh
  mvn -version
   mvn --version
   ls
   export M2_HOME=/opt/apache-maven-3.6.3/
   export PATH=$PATH:$M2_HOME/bin
  mvn --version
```

**Result will be:**

```
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /opt/apache-maven-3.6.3
Java version: 1.8.0_252, vendor: Oracle Corporat
```

After showing result from root

```
vi .bash_profile
    export M2_HOME=/opt/apache-maven-3.6.3/
   export PATH=$PATH:$M2_HOME/bin
```

Paste below export and save it.

**Maven git need to check from aws.
Create a maven job with maven project .
Git add https://github.com/shivasingam111/simple-java-maven-app.git
Github also add https://github.com/shivasingam111/simple-java-maven-app.git this url

Build pom.xml below type compile

Save and build now

Shows success message

Check style plugin:
Manage jenkins → manage plugins → available check (check style plugin is there or not)

Checkstyle + intstall without restart .

Take a new freestyle job

To get report of the code we can use this one….

User management in jenkins:

Create users in jenkins:
Manage jenkins → manage users → create user. + save

Now we have to give the permission.

Manage jenkins -> configure global secuirity →
In authorization tab → check with Matrix-based security

Here add user or group :(username) give name and give permission + save.

Continious deployment:

Deplyment server :

Tomcat server on same jenkins server:
ls -la
[]cd apache-tomcat-8.5.54
[]cd conf
vi server.xml (change port no 8080 to 80)

```
<Connector port="80" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

vi tomcat-users.xml (add roll name)

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<role rolename="admin"/>
<user username="admin" password="admin" roles="admin,manager-gui,manager-script"/>
```

Cd bin
Cd ./startup.sh

***if tomcat manager not open means go and below procedure
/webapps/manager/META-INF/context.xml:

```
<Context antiResourceLocking="false" privileged="true"> <!-- <Valve
className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> --> </Context>
```

Comment the value tag above code

----------

***Note :whenever u r starting a apache server u need to start bin file cd bin /  ./startup.sh
Take aws ec2 new server and allow all trafic + launch.

Terminal :

```
sudo dnf install wget
wget https://downloads.apache.org/tomcat/tomcat-8/v8.5.54/bin/apache-tomcat-8.5.54.tar.gz
tar -xvzf apache-tomcat-8.5.54.tar.gz
ls -la
[]cd apache-tomcat-8.5.54
[]cd conf
vi tomcat-users.xml
```

Paste the below code

```
<role rolename="manager-script"/>
  <role rolename="manager-gui"/>
  <user username="tomcat" password="tomcat" roles="manager-script,manager-gui"/>
```

Now we have to run or start the tomcat server.

```
cd ..
cd bin
sudo yum install java-1.8.0-openjdk-devel
 ./startup.sh
```

The result will be:

```
Using CATALINA_BASE:   /root/apache-tomcat-8.5.54
Using CATALINA_HOME:   /root/apache-tomcat-8.5.54
Using CATALINA_TMPDIR: /root/apache-tomcat-8.5.54/temp
Using JRE_HOME:        /
Using CLASSPATH:       /root/apache-tomcat-8.5.54/bin/bootstrap.jar:/root/apache-tomcat-8.5.54/bin/tomcat-juli.jar
```

After startup.sh the tomcat server is going to start.
Now apache tomcat server is open.

```
Cd apache-tomcat-8.5.54/webapps/manager/META-INF
vi context.xml
```

Comment this value.

```
<!--
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
  <Manager
sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalina\.filters\.Cs
rfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
-->
```

Now Again we need to restart the tomcat server.

```
cd bin
./startup.sh
```

In url type : http://13.127.5.66:8080/manager/html

One popup will open just u have to give the username ans password account;
Username : tomcat
Password: tomcat

## Integrating Jenkins With Tomcat(valaxy technologies)

Go To jenkins and manage plugins → available → search (deploy to container) → install without restart .

Step1:
Create a freesty project -> git url :https://github.com/shivasingam111/webapp.git → Poll SCM * * * * * → Build (top level maven project) choose maven project & goals : package apply and save → build now.

Step 2:  go to previous project and post actions choose deploy/war file :
war/ear files give exact path : target/mvn-hello-world.war → conext path any name → containers choose ur tomcat version and credentials give ur tomcat credentials → give tomcat url → apply and save .

Build now the file ear directly deploy to our tomcat server.

Check with:
ipaddreess /projectname
ex:http://35.154.103.68/mvn-hello-world/

Jenkins backup plugin
Used for taking backup from server.

## Jenkins Upstream and DownStream: