Standalone

1) To create a server and connect to slave node

Step 1: Server Name : ubuntu-server.
No of executors : 0
Remote root direct : /home/nsis/jenkins
Launch method : Launch agent by connecting to the controller

Save

Step 2:
In slave system, enter the ip addres of Master and go the server and copy the curl command and
i) In terminal execute curl command
ii) and java code with sudo

Server should be connected.

In Dashboard
i) create new item
item name : basic-php (standalone)
↳ restrict to (server name) ubuntu-server

↳ Git (paste the repo) code basic php
↳ Build step
Execute shell - Inside paste the sh file code

build

Save

Step -3

Now test in slave

give the ipadress : 8080

create another item name with docker.

i) Restrict to server

ii) Git : repo basic php

iii) Build step : Docker build and execute

↳ i) give the repo name

tag : version (V_1)

Registry credential

(docker name)

Login to Docker-hub

email

password - sahanaviney

i) create repo

↳ save

Same

build

copy

ii) and add execute shell

on build step.

```
Sudo docker run -- name
project -it -d -p
4001:80 (repo name)
:V1
```

Manage Jenkins
↳ Global Tool Configuration.
version 3.9.6 save.

maven
i) Config - maven - add-maven save.

Step 1 : item_name - clean_compile
i) restrict (ubuntu server)
2) Git repo (maven_java)
3) Build step
↳ invoke top-level
i) maven version : ( )
Goals : clean compile

save
Build

Step 2 item_name : ~~clean~~ code-review
i) restrict (ubuntu server)
2) Git repo (maven java)
3) Build step.
↳ invoke top-level
i) maven version : ~~code-vali~~ ( )
ii) validate (goals

again add build step
↳ invoke top-level
Goals : ~~0~~ -P metrics pmd :
pmd checkstyle : checkstyle
findbugs : findbugs

Post build step.
add Tools : checkstyle
add Tool : Findbugs
add Tool : PMD
Save build

Step 3:

item_name : maven_Test

i) restrict

ii) Git repo

iii) Build step : invoke top level

maven version :

Goals : Test validate

Save

build

Step 3: item_name : install-

i) restrict

ii) GIT repo

iii) Build step

invoke top level

Goal : install packages

Save

Build

Now add the Post build steps for all clean-compile, code-review, maven-Test, maven-install.

↓

Configure

In Clean compile

↓

Configure

↓

Post build step.

↳ Build other project

project_to_build = code-review

Save.

In code-review

↳ Build other project

project_to_build = maven-Test

Save

In maven-Test.

↳ Build other project

project_to_build = maven-install

save.

Now in Dashboard add new view

CICD_Test

↳ Build Pipeline view

create

upstream / downstream config

i) select initial Job.

clean-compile

~~Build~~ restrict triggers

. No

OK without restrict localhost:80

Next In slave : ~~localhost:80~~

(IP of 172.16.51:80

slave ~~node~~ 26)

Kube

New item = Kube Test

1) ~~Git~~ restrict

2) Git (php)

3) Build step

i) Deploy to Kubernets

↳ Kub Config: Kube-demo

↳ Config file: bookalbum/deployment, Bookalbum-service.yaml

In Git go to Yaml file

copy the name of

yaml deployment & service.

Save.

Build

In Terminal

ssh chefserver@172.16.51.53 (Give master IP)

password: chef@123

Sudo ~~g~~ kubectl get pods

Sudo kubectl get svc

Sudo kubectl describe pods

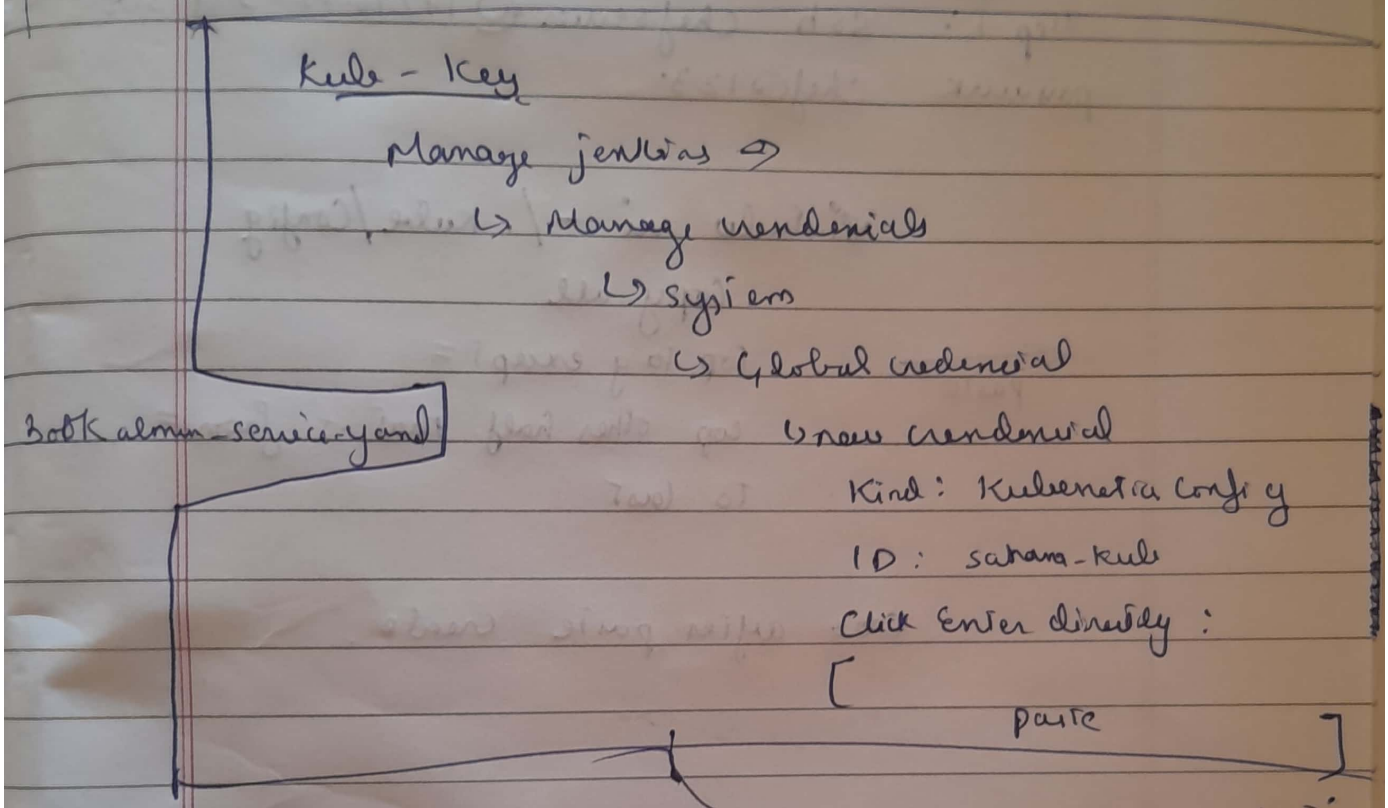
Kubernetes master IP in

Node: 172.16.51.58

Port: 80:32098/TCP

O/P 172.16.51.50 : 32098/index.php

IP of Master



In Terminal:

login to chef server

Step 1: ssh chefserver@172.16.53.53 } Sir

password chef@123.

sudo vi ~/.kube/config

copy all

paste ← top to g except =

cop other half that is from =

to last

after paste create.