

Create a user :-

cd

cd apache-tomcat-8.5.76/conf/

ls

cat tomcat-users.xml.

vi tomcat-users.xml.

⇒ Paste the permission from (Pdt).

⇒ Deployment location webapps

**

How to create a service?

⇒ Create a file under '/etc/systemd/system/' directory:

(To config Tomcat)

Now, →

Jenkins?

CI-CD-Job

Jenkins? → manage plugin Jenkins

→ manage plugin

→ Deploy to container

→ download

→ CI-CD-Job

→ Config

→ Post-build

→ Deploy war/ear

**/*.war

→ content path

hello world servlet

→ Containers

Tomcat 8x

→ Add cred

User :- admin

pass :- admin

ID :- Tomcat-cred

Dec :- Tomcat-cred

→ Tomcat URL :-

[Save] [Apply]

→ To reconnect

/etc/init.d/tomcat start

communication

E-mail :-

→ Dashboard

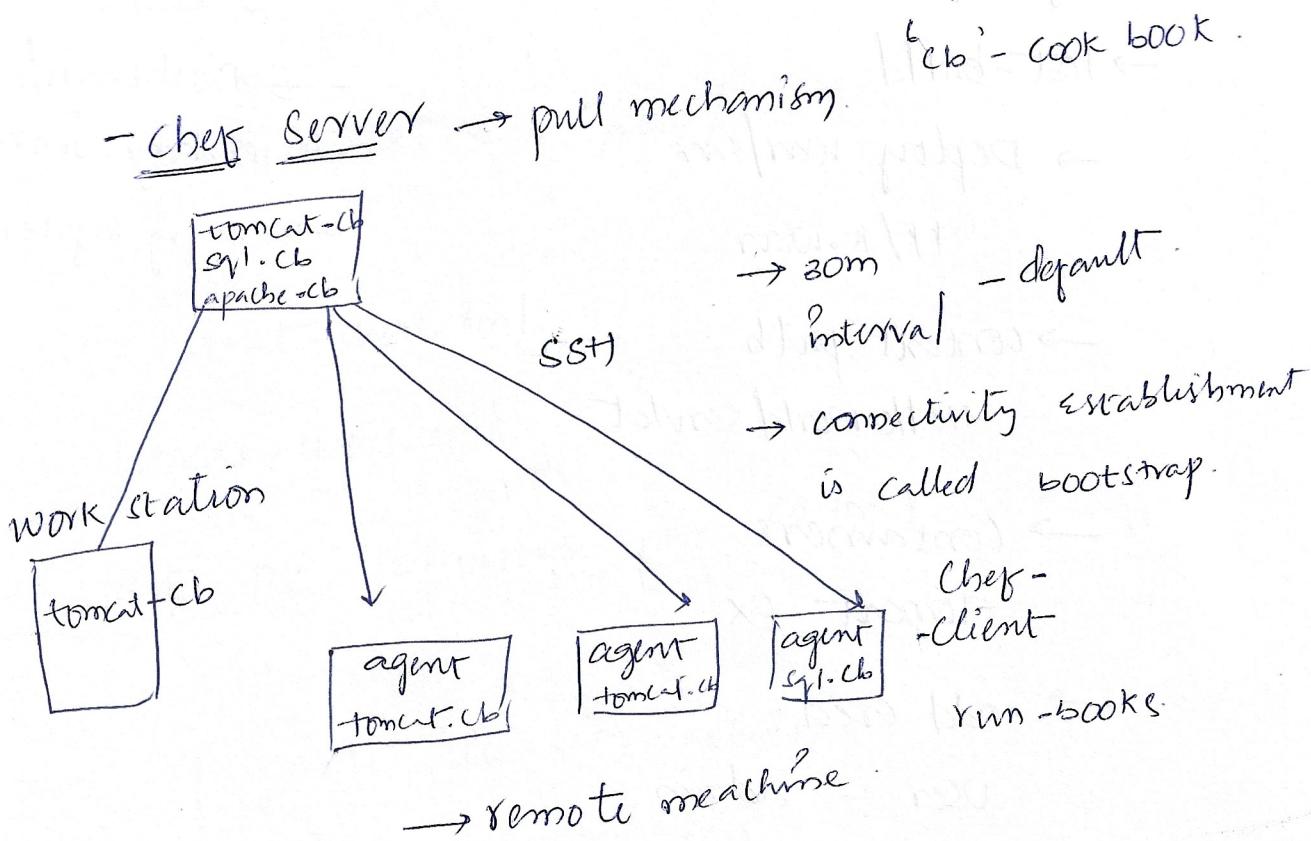
→ manage Jenkins

→ config system

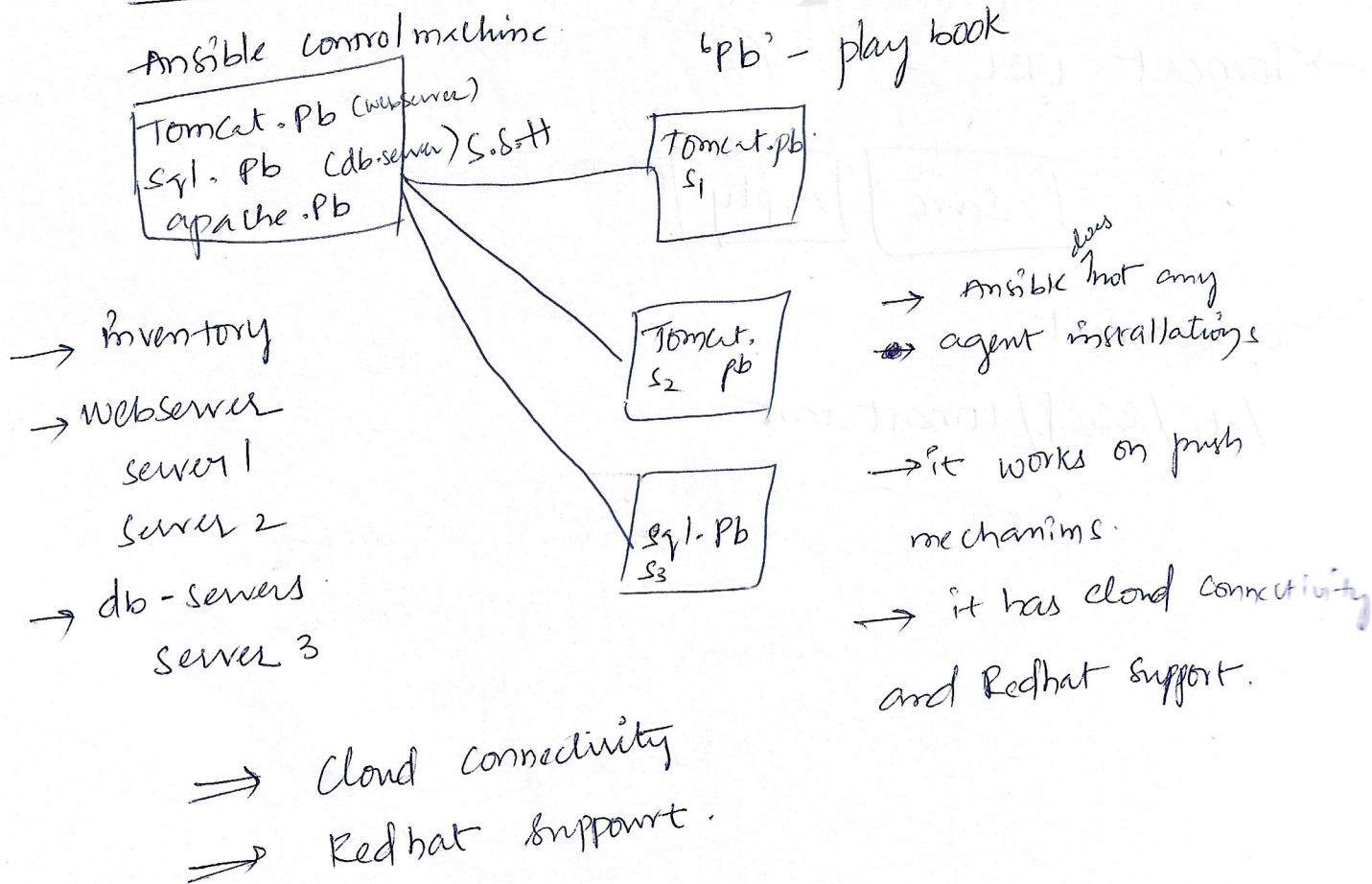
Ansible :-

ANSIBLE :-

10/03/2022



Ansible Server :- Push mechanism.



Mammal Connection :-

→ CM

"control machine" (113) (91)

→ RM

"Remote machine" (44) (166)

→ CM (113) (91)

user add ansible

→ RM (44)

user add ansible

visudo

⇒ root ALL=(ALL) ALL

ansible ALL=(ALL) ALL

ls

→ cm (113) (91)

su ansible

cd

prod

whoami

ssh-keygen

cd .ssh/

ls

cat id_rsa.pub

→ copy the public key

→ RM (44) (166)

su ansible

cd

pwd

whoami

mkdn' .ssh

chmod 700 .ssh/

cd .ssh/

vi authorized_keys

→ Paste the public "ip of (113) CM"

chmod 600 authorized_keys

ls -l

cat authorized_keys

cd

→ CM (113) (91)

ssh ansible@Private I.P. ("RM")

⇒ ssh username@I.P address

→ connection is done ←

exit from "RM"

→ download ansible ←

wget "Path"

ls

→ no permission ←

Go back to root user

git

ls

cd /home/jamisble/

ls

cd

rpm -ivh /home/jamisble/epel-release-latest-7.noarch.rpm

Yum install ansible -y

ansible --version

ls

cd /etc/ansible/

ls

→ hosts file is nothing but inventory file

cat hosts

→ To get help.

vi hosts → To provide 'rm' details to 'cm'

→ remove ## for web servers and add "rm"
private I.p address.

ex:- [web servers]

172.31.22.44

→ by default the vi hosts file is not active we have
to activate it, for that got →

cd /etc/ansible/

ls

vi ansible.cfg

→ [defaults]

inventory = /etc/ansible/hosts

→ inventory = /etc/ansible/hosts

⇒ Now,
check the connection b/w two machines.

Switch to ansible user.

su ansible

cd

prod

whoami

ansible webservers --list-hosts

→ To see the list of host added under webserver group. (list of 'RM')

ansible webservers -m ping → (To check connectivity)

~~imp~~ ansible webservers -m setup } Ansible Adhoc command.

→ To know server information.

ansible webservers -m command -a uptime.

ansible webservers -m command -a hostname.

ansible webserver -m shell -a 'ls -l > temp.txt'

→ Bash shell command.

→ ansible-doc -l

→ list of modules in ansible

ansible webserver -m file -a 'dest=/home/ansible/samkavant'

state=touch mode=600 owner=ansible group=ansible

ansible webserver -m file -a 'dest=/home/ansible/samkav'

state=directory mode=755 owner=ansible group=ansible

→ A module can be used to create a 'file' or

a 'directory'

Email communication :- { After Nexus } -

b/w the teams. to provide the reports of the Build.

- Dashboard
- Manage Jenkins
- Configuration System
- E-mail notification

SMTP Server - smtp.gmail.com

SMTP AUTH

Use SSL . Use SMTP AUTH

User name :- email id

Pass :- email Pass

Use SSL

SMTP Port

465

Test config

Now, Go to Email and check activity

Go to settings and enable 'Less secure app access'

Now, Go to 'Jenkins'

Go to Job ex:- CICD-Job

→ Post-build action

→ E-mail Notification

Add e-mail id

* E-mail will be sent if only error is occurred.

→ whenever there is failure we will receive a mail and next success build mail will also be sent

⇒ [Failure and Success] communication

Remove the existing settings

→ Manage Jenkins

→ config system

→ Extended E-mail Notification

SMTP server :- Smtp.gmail.com

SMTP Port :- 465

Use SSL

→ Add credentials

Username & Password

Username :- mail id

Password :- mail password

ID :- email-cred

Description :- email-cred

Now,

CI-CD-Job

→ Job config

→ Post-build

remove already missing settings

→ Editable email notification

→ Attach build log

→ Advanced settings

Triggers

Add triggers

* Always

→ Advanced

→ Add recipient

Ansible :-

'CM'

Switch to ansible and check connectivity

ansible webserver -m ping

'RM'

root user

visudo

→ copy NOPASSWD = ALL and
paste ansible ALL = [ALL] NOPASSWD = ALL
and save

'CM'

ansible webserver -m user -a 'name=user1 password=user1' --become

→ to create the users

'RM'

cat /etc/passwd

→ user1 is created

'CM'

→ copy module :-

ansible webserver -m copy -a "src=sample.txt dest=/home/ansible/simple.txt"

→ err. because no source, so, create a source

touch sample.txt

install module :-

Apache with default
Launch page

1) install apache :- Yum install apache

2) start apache :- service httpd start

Installation of git

using ansible :-

3) default page :- deploy into deployment location
(var/www/html)

ansible webservers -m yum -a "name=git state=present" --become

To delete the package :-

ansible webservers -m yum -a "name=git state=absent" --become

To install apache :-

ansible webservers -m yum -a "name=httpd state=present" --become

To start apache :-

ansible webservers -m yum -a "name=httpd state=started" --become

Ansible Playbook :- Triggering a bunch of modules.
A playbook is set of plays and a play is set of modules.
Contains 3 Sections :-

- - -

Host :- based on ansible inventory file

Variable :- optional. (inputs)

Task :- running modules will be defined under task section

→ This is written in 'Yml' language → "YAML"

Create a directory called playbooks.

mkdir playbooks

cd playbooks /

ls

→ To create a file

vi filecreate.Yml

→ paste the code

ls

→ To check any syntax error

ansible-playbook <playbook name> --syntax-check

ex:- ansible-playbook filecreate.Yml --syntax-check

→ To execute the play book

ansible-playbook (playbook name)

ex:- ansible-playbook filecreate.Yml

To create a directory

vi dircreate.yml

→ Paste the code ←

ansible-playbook dircreate.yml

To create multiple directories :- [loop]

Whenever working

vi multidir.yml

→ Paste the code ←

ansible-playbook multidir.yml

with the loop we

can use 'items'

→ To check the directories created

ansible webservers -m command -a 'ls -l'

To find and replace a word :-

[sed command
Linux].

vi findreplace.yml

→ Paste the code ←

ansible-playbook findreplace.yml

→ It will fail ←

ansible webservers -m copy -a 'src=/etc/passwd dest=/home
/ansible/passwd' --become

→ To check

ansible webservers -m command -a 'ls -l'

ansible-playbook findreplace.yml

Now, check.

ansible webserver -m command -a 'cat /etc/passwd'

→ To compress (or) archive the data :-

vi archive.yml

→ Paste the code ←

ansible-playbook archive.yml

Check list

ansible webserver -m command -a 'ls -l'

→ multiple modules in playbook :-

Apache setup

vi apache.yml

→ Paste the code ←

ansible-playbook apache.yml

Installations :- (using Playbooks)

Maven :-

cd playbooks/

vi mm.yml

→ Paste the playbook from Pdf.

copy module :- src/copy src/dist

line module :- To append content

get module :- To download content from internet.

get-url : to download content from internet.

module = wget

get-ur = playbooks ansible.

Tomecat :- 'Service Tomcat status'

vi tomcat.yml

module = Yum

→ Create a file tomcat-users.xml

→ Deployment area is 'webapps'

on remote server ① way

cd /var/share/tomcat

cd conf

cat tomcat-users.xml (copy content).

on control server

check if you are on /home/ansible / location

Create the tomcat-users.xml and (paste the content).

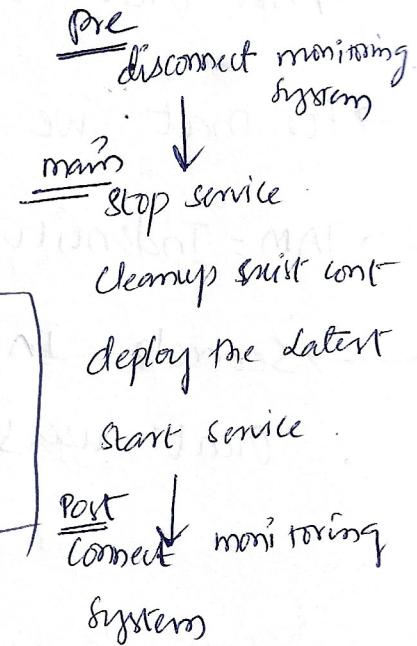
and add the user from pdf (Tomcat).

cd playbooks /

ansible-playbook tomcat.xml

Tasks:-

Pre-tasks, Post-tasks and tags.



Tags:-

To execute a particular play from play book there we

use Tags.

debug = echo

vi prepost.yml

→ To check if any tags present in playbook

ansible-playbook 2tag name > --list-tags

ex: ansible-playbook prepost.yml --list-tags

→ for particular tags

ansible -playbook prepost.yml --tags Samkar

→ To skip particular tag

ansible-playbook prepost.yml --skip-tags Samkar