# **Ansible**

Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy.

It supports configuration management with examples as below.

- Configuration of servers

- Application deployment

- Continuous testing of already install application

- Automation of tasks

- Learning is easy, no need to learn a special language

- Syntaxes used by Ansible is YAML

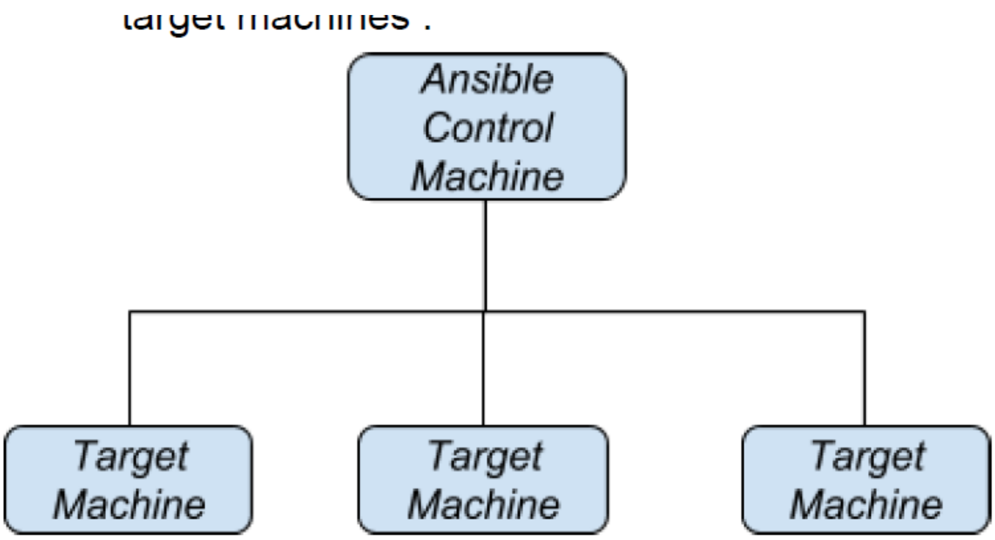
- Ansible internally uses python for processing YAML templates

- Ansible is agentless

- Ansible is highly durable & scalable

- SSH is the communication channel used by ansible to controlling

target machines .



Ansible control machine is the machine installed with Ansible, from this machine we run ansible code(Playbooks)

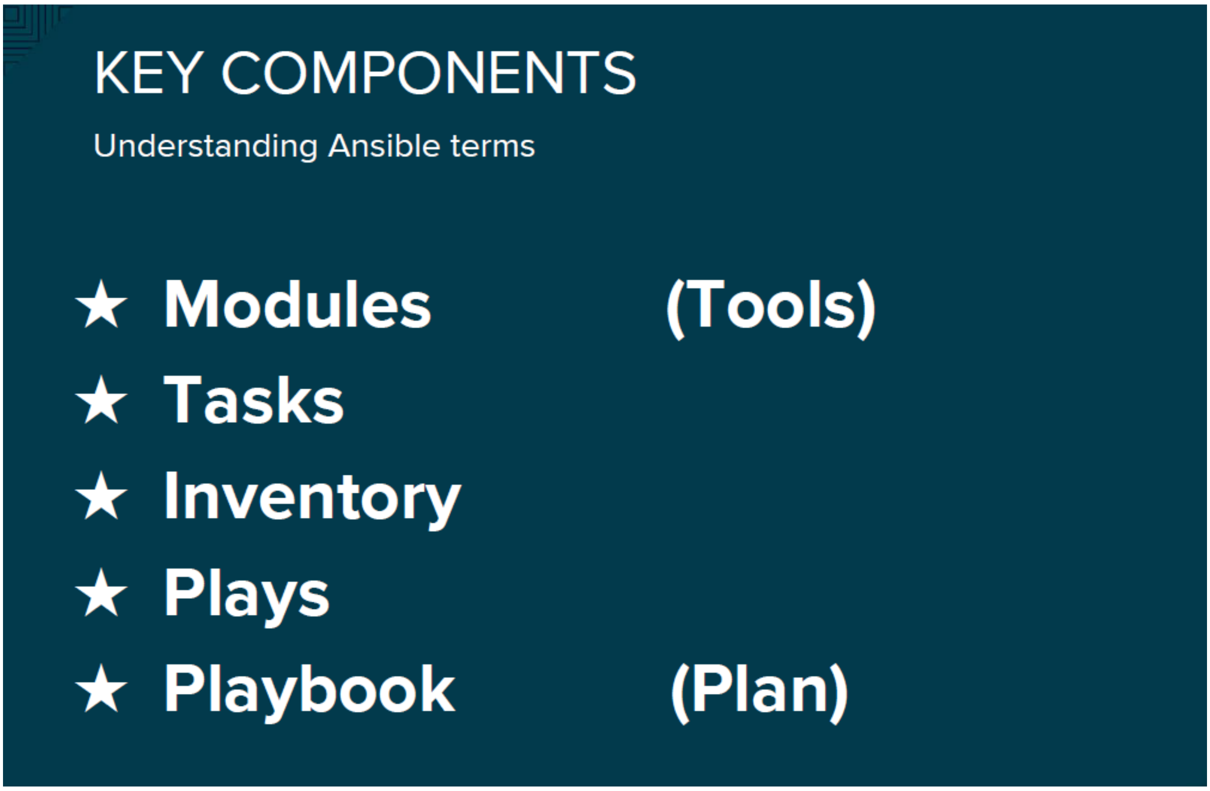
Target machine is the server controlled by ansible, target machine can be webserver, database server or any machine.

Note: In case of chef, puppet we need to install agents on the target machine.

## YAML

- YAML is used as configuration files
- YAML is data representation.

## Key Components of Ansible:



KEY COMPONENTS

Understanding Ansible terms

★ Modules (Tools)

★ Tasks

★ Inventory

★ Plays

★ Playbook (Plan)

## Installation of Ansible:

Step 1: First we need to take 3 ec2-instances , one for Ansible control master and two for nodes.

Step 2: Login into Ansible control master after do below steps

1. Create one user that name is called ansible ( useradd ansible)
2. Assign password to ansible ( passwd ansible )
3. edit file vi /etc/ssh/sshd_config to comment out

   Permitrootlogin =yes (remove #)

   PasswordAuthentication = Yes (remove #)

4. Add ansible user under file vi /etc/sudoers (like below)

```
ansible     ALL=(ALL)           NOPASSWD: ALL
```

5. Start ssh service

   Service sshd start

6. Switch user as ansible

   Su – ansible

7. Generate sshkey from master to nodes

   1. ssh-keygen -t rsa (enter 4 steps)

      restart the ssh service

      sudo service sshd restart

   2. ssh-copyid node1 IP
   3. ssh-copyid node2 IP

8. Install ansible by using below command

   sudo amazon-linux-extras install ansible2 -y

9. modify file

   sudo vi /etc/ansible/ansible.cfg (remove # (comment) infront of inventory file.

10. Edit the file below

    sudo vi /etc/ansible/hosts

    comments out for webservers and provide node1 and node2 private ips.

**Node 1 setup:**

Step 1: Login into Node1 after do below steps

1. Create one user that name is called ansible ( useradd ansible)
2. Assign password to ansible ( passwd ansible )
3. edit file vi /etc/ssh/sshd_config to comment out

   Permitrootlogin =yes (remove #)

   PasswordAuthentication = Yes (remove #)
4. Add ansible user under file vi /etc/sudoers (like below)

   ansible ALL=(ALL) NOPASSWD: ALL
5. Start ssh service

   Service sshd start
6. Switch user as ansible

   Su – ansible
7. Generate sshkey from master to nodes
   1. ssh-keygen -t rsa (enter 4 steps)
8. restart the ssh service

   sudo service sshd restart

**Node 2 setup:**

Step 1: Login into Node1 after do below steps

1. Create one user that name is called ansible ( useradd ansible)
2. Assign password to ansible ( passwd ansible )
3. edit file vi /etc/ssh/sshd_config to comment out

   Permitrootlogin =yes (remove #)

   PasswordAuthentication = Yes (remove #)
4. Add ansible user under file vi /etc/sudoers (like below)

   ansible ALL=(ALL) NOPASSWD: ALL
5. Start ssh service

   Service sshd start

6. Switch user as ansible

   Su – ansible

7. Generate sshkey from master to nodes

   1. ssh-keygen -t rsa (enter 4 steps)

8. restart the ssh service

   sudo service sshd restart

Finally Ansible setup complted from master and server.

You will check below command from master

**Sudo ansible all -m ping**

Then you will get ping response from Nodes.

All ansible commands from master you need to write.

**Ansible adhoc-commands**

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

above command  all means webservers

m stands for module , yum is the module , that module name is git and state is present means install

i.e git installation for both the nodes

```
 ansible all -m yum -a "name=git state=present" --become
 ansible all -m yum -a "name=httpd state=present"
ansible all -m yum -a "name=httpd state=present" --become
ansible all -m service -a "name=httpd state=started" --become
ansible all -m yum -a "name=httpd state=absent" –become
ansible all -m copy -a "src=/home/ansible/123 dest=/home/ansibl"e --become
ansible all -m command -a 'df -h'
ansible all -m command -a 'free -m'
ansible all -m command -a 'uname -a'
ansible all -m command -a 'ifconfig'
ansible all -m command -a 'ifconfig' --become
ansible-doc -l
```

**Playbooks:**

Playbook contains list of play or tasks

Tasks contains modules

Modules will execute the entire playbook.

We are writing playbooks by using YAML script

Playbook name extension must be .yml

We will create one directory for writing all playbooks

That is sudo mkdir /opt/playbooks

cd /opt/playbooks

=================================================

Example playbook

Sudo vi file.yml

```
---
 - hosts: localhost
   become: yes
   tasks:
     - name: create empty file
       file:
         path: /etc/foo.conf
         owner: ec2-user
         group: ec2-user
         mode: 0644
         state: touch
```

Above playbook is the creating file in ACM.

How to run playbook for bellow commands

ansible-playbook file.yml –syntax-check → for syntax checking ansible-playbook file.yml –check --→ for Dry run mode

ansible-playbook file.yml  for running playbook.

**Jenkins Integrate with Ansible:**

**Note: Before going to Jenkins integrate with Ansible,**

**You need to install tomcat on Node 1 or Node 2**

Step 1: First we need to take one ec2-instances  for Jenkins

Step 2: Login into Jenkins do below steps

1. Create one user that name is called ansible ( useradd ansible)
2. Assign password to ansible ( passwd ansible )
3. edit file vi /etc/ssh/sshd_config to comment out

   Permitrootlogin =yes (remove #)

   PasswordAuthentication = Yes (remove #)

4. Add ansible user under file vi /etc/sudoers (like below)

```
ansible     ALL=(ALL)          NOPASSWD: ALL
```

5. Start ssh service

   Service sshd start

6. Switch user as ansible

   Su – ansible

7. Generate sshkey from master to nodes

   1. ssh-keygen -t rsa (enter 4 steps)

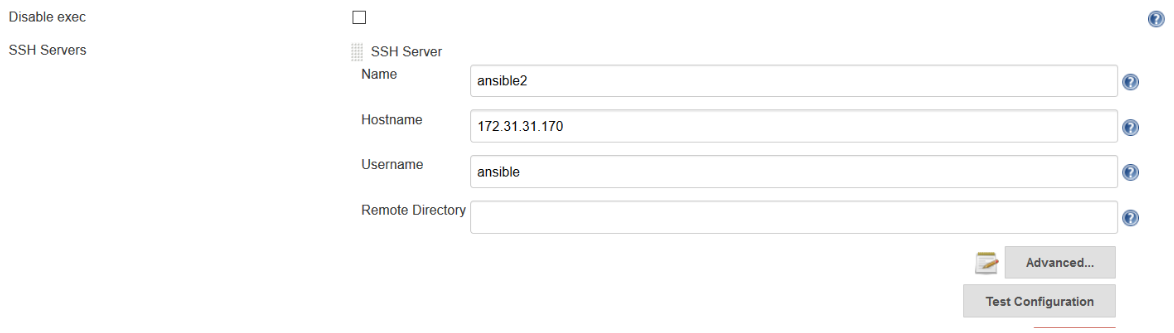
      restart the ssh service

      sudo service sshd restart

   2. ssh-copyid Ansible Control master IP

**Step2:** First you need to install **PublishOverssh** plugin in Jenkins

Step 3: In Jenkins go to manage Jenkins and after that click on configure system and provide ACM ansible user credentials   like below.



Disable exec □

SSH Servers

SSH Server

Name: ansible2

Hostname: 172.31.31.170

Username: ansible

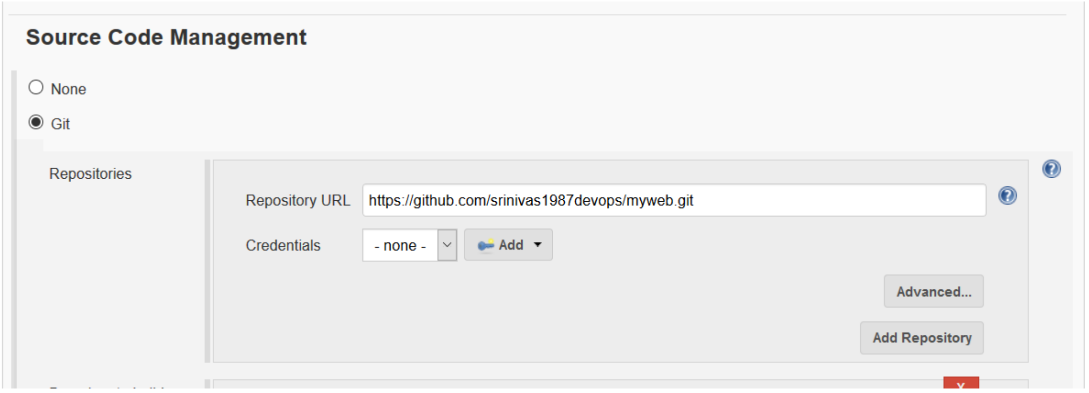
Remote Directory:

Step3: write the playbook in copy.yml under /opt/playbooks

**Note: Don’t run the playbook in master**

Step 4:  Create a Jenkins job under Freestyle  like below steps

1.Under SCM section



Source Code Management

○ None

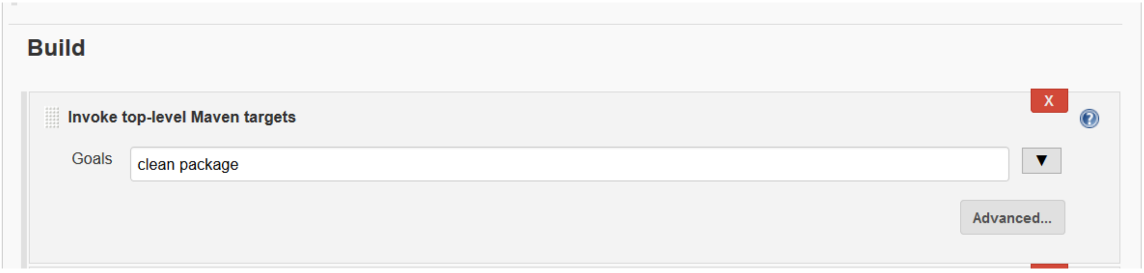
● Git

Repositories

Repository URL: https://github.com/srinivas1987devops/myweb.git

Credentials: - none -

2.Under Build section

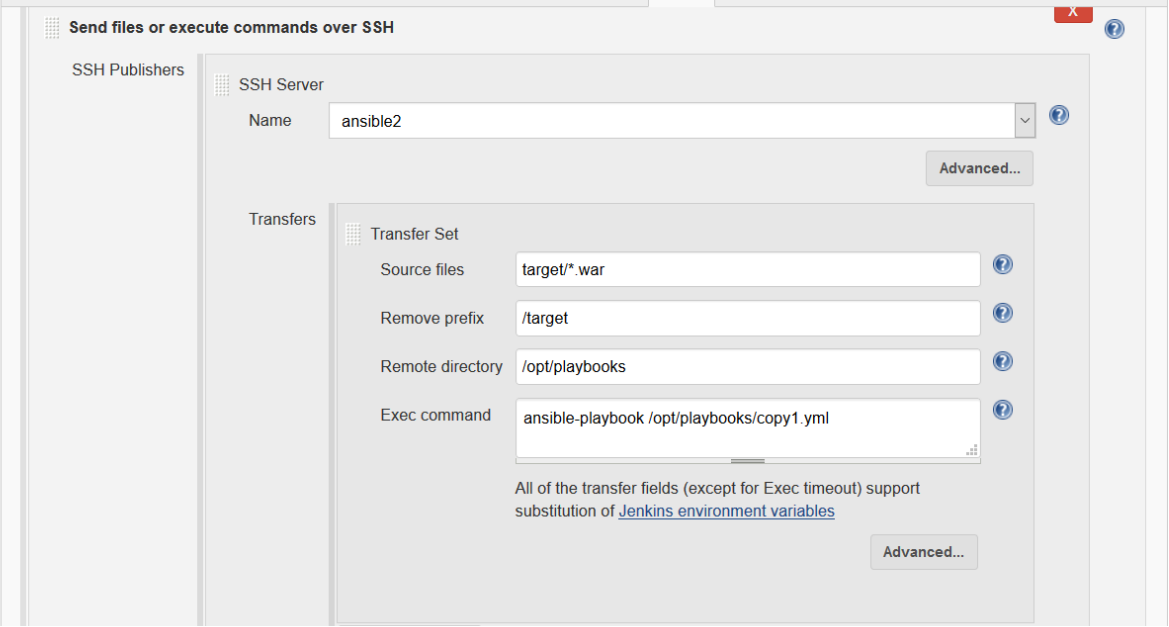


Build

Invoke top-level Maven targets

Goals: clean package

3.One more step add in Build section select send files or execute commands over ssh



Send files or execute commands over SSH

SSH Publishers

SSH Server

Name: ansible2

Transfers

Transfer Set

Source files: target/*.war

Remove prefix: /target

Remote directory: /opt/playbooks

Exec command: ansible-playbook /opt/playbooks/copy1.yml

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

Step5 : Finally Build the Jenkins Job.