ANSIBLE

Ansible – configuration management tool

If we want to configure certain functions in remote servers like if we want to install tomcat on 1000servers then manually it takes lot of time, in that scenario configuration management takes place.

Wherever the servers be, if we give instructions in the tool then configuration management tool makes sure to install in all the servers irrespective of location.

Advantages:

1. Resource saving
2. Provisioning of servers
3. DR (disaster recovery)
4. Snow flake server effect
5. Idempotent

Popular configuration management tools:

* Ansible
* Chef
* Puppet
* Saltstack

Machine in which Ansible is installed is called controller. Nodes under controller are known as Managed nodes or managed hosts.

For working of Ansible we must have passwordless connection between controller and managed nodes/hosts.

In Ansible we have push methodology, it means pushing of resources from controller to node servers with the help of passwordless SSH connection.

**ANSIBLE Installation:**

* Create 3 instances, naming them as controller and server-1&2.
* connect to server-1, check the username and set a password.
* After setting the password, restart the service.
* Repeat it on all managed nodes you have.
* Connect to the controller, generate keygen. It will create two keys which are private and public.
* Now we will copy the generated keys into server 1 & 2, Ssh copyid@ubuntu private IP address of server1.
* update repository, as Jenkins is dependent on java, Ansible also dependent on Software Properties Common.
* Install software properties common.
* By default while updating repository we have an older version of Ansible in the repository. Before installing latest version of Ansible we first have to add the latest version of ansible into repository. Sudo apt-add-repository ppa:ansible/ansible.
* Now once again update repository and install ansible.
* Inventory is the file where we store the ip addresses of nodes.
* Inventory file will be present in /etc/ansible/ folder. Now open that inventory(hosts) file. And copy the ip addresses of those two servers.
* Linux command – date to see the current date.
* Now if we want to see the date of manager nodes, **ansible all –a ‘date’**
* To see all the files along with hidden in the servers, **ansible all –a ‘ls –la’**
* Now create a Redhat Linux instance and connect with controller where ansible is installed.

Ansible is created using a scripting language Python. Ansible performs configurations in 3 different ways:

1. **Adhoc commands**
2. **Playbooks**
3. **Roles**

In organizations we mostly work on Playbooks and Roles. First we will know how to use python modules in Adhoc commands, Playbooks and Roles.

Important modules in Ansible:

1. Command – running Linux commands on remote servers.
2. Shell – if we have python script and want to execute it in all the servers then shell module is used.
3. Ping – this module is used to check whether the remote servers are pingable or not.
4. User – use to perform user administration. User creation, user deletion etc.
5. File – in Linux we use touch command to create files and mkdir to create directories. To remove files we use rm command. To change permision we use chmod command, and to change ownership we use chown Linux command. To change group ownership chgrp command is used. These Linux commands are used only on one server, File module is used to perform all these operations in all the servers at same time.
6. Copy – in Jenkins we use scp to copy files from source location to destination using passwordless connection. But to copy for 100 servers we have to change ip address for 100 times, but in ansible Using COPY module we can do it in hundreds or thousands of servers at same time.
7. Fetch – this is reverse of your copy module. Copy files from managed nodes to controller.
8. Apt – installing software on multiple servers at same time.
9. Yum – same like apt, but apt is used in Ubuntu whereas yum is used in centos and Redhat Linux.
10. Git – used to perform git operations in all the servers at same time.
11. get\_url – similar to wget. But wget is used on one server whereas get\_url is used on multiple servers.
12. Service – service module is used to restart the service same time at multiple servers.
13. Uri – it is used to check whether remote server is reachable or not.
14. Debug – used to print in ansible
15. Replace
16. Pause – if you want to stop execution of a playbook for some time then this module is used.
17. Include – used to call child playbook from parent playbook.
18. Stat – used to store files and folders in remote servers.
19. Docker\_container – used to get detailed information about container.
20. Docker\_images – used to get detailed info about image.

**Syntax of Adhoc command:**

Ansible all/ip\_address/group\_name –i path\_of\_inventory -m module name -a “ argument“

Let’s connect to controller

check the date of managed nodes**: ansible all –i /etc/ansible/hosts –m command –a ‘date’**

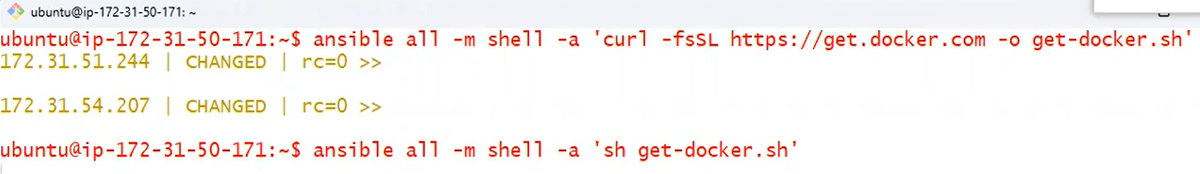
checking memory in nodes: **ansible all –i /etc/ansible/hosts –m command –a ‘free -m’**

For inventory we can create an inventory and make use of it. Let’s choose one ip address, **cat > myinventory** paste the address of one server. Now we created inventory with only 1 IP address in it.

No need to mention –i if we are working on default inventory. If we create any inventory then we should use –i.

It is okay even if you don’t give –m and module name while you are working on command module. If you don’t enter any module by default it considers as command module.

Installing docker using ansible.



User creation using ansible



If we mention user ID and didn’t give group ID then group id will be same as user id.

Creating files:



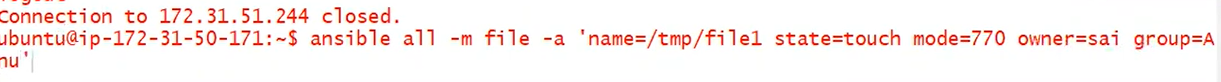
Creating directory:

While creating directory it is same as creating file but the difference is state = directory.

Deleting directory:

State = absent

Changing group ownership and permissions of user for a file:



**Copy module:**

Create a file in controller with some data in it.



If the output comes in yellow then it describes that there are some changes made in it. When it comes in green colour then it describes successful and required changes are already made in it.

**Apt module:**

In this we have three states, present for installation, Absent for uninstallation, latest for upgrading.



If the requirement is updating apt and installing tomcat9 on managed nodes.



If you want to uninstall any application and delete it’s configuration files,



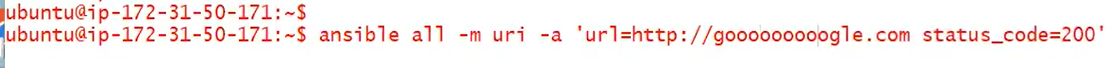
Restarting service in all server nodes.



Here we use service module, and in service module we have 3 states, restarted. Started. Stopped.

**URI module:**

This module is used to check whether specific service is accessible from manage nodes or not. In this we have only two conclusions. -1 means failed, 200 means success.



**Get\_url module:**



**Replace module:**

replacing specific part of data can be done using replace module.

In tomcat we have server.xml where we have connector port.



Git cloning in all manager nodes:



**Fetch module:**



**Disadvantages:**

We can call only one module at a time.

We can run only one argument at a time.

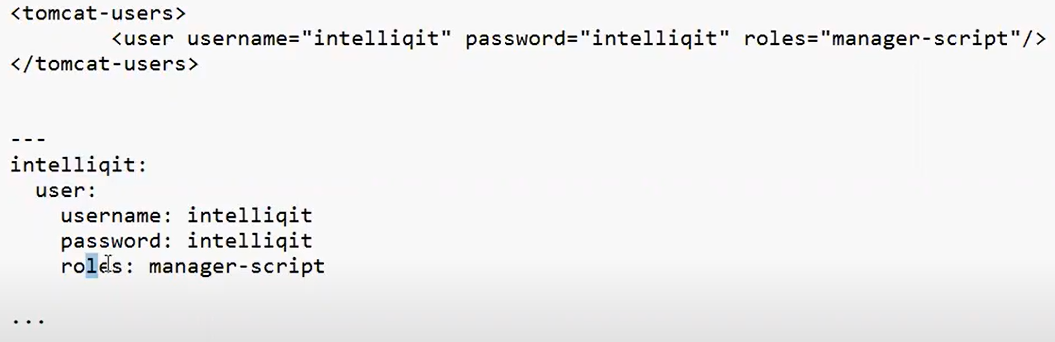
Reusability wise playbook is more advantageous than Adhoc commands.

Install apache2 on all manage nodes by updating repository. After that in apache2 home page we want to give content.

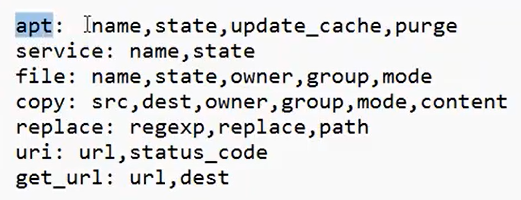


Now we will restart apache2 service.

In **playbook** we can execute multiple modules.



Top one is xml file and bottom one is yml file in ansible.



**ANSIBLE PLAYBOOKS:**

Connect to controller.

Create a folder playbooks.

Playbook to create user:



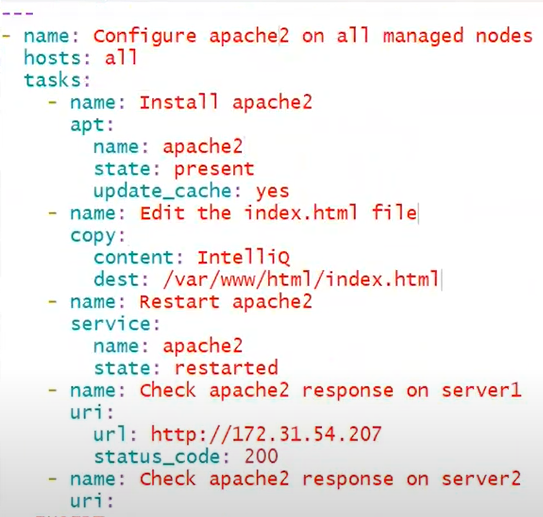
To check whether the yml script created is syntactically correct or not.

Ansible-playbook playbook1.yml --syntax –check

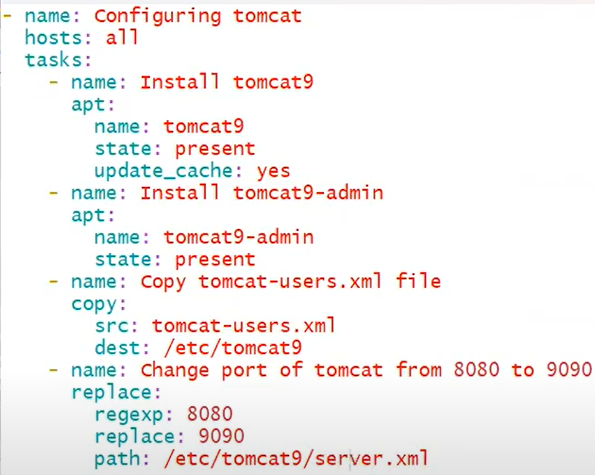
If the yml script is correct it will return the playbook name, if there is any mistake then it shows where the mistake is.

To run a playbook 🡪 ansible-playbook playbook\_name

Configuring apache2 on all managed nodes:



Installing Tomcat9 and Tomcat9-admin,copy the tomcat-users.xml file into all managed nodes.



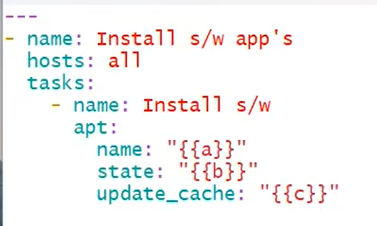
VARIABLES:

There are three types of variables:

* Global scope variables
* Play scope variables
* Host scope variables

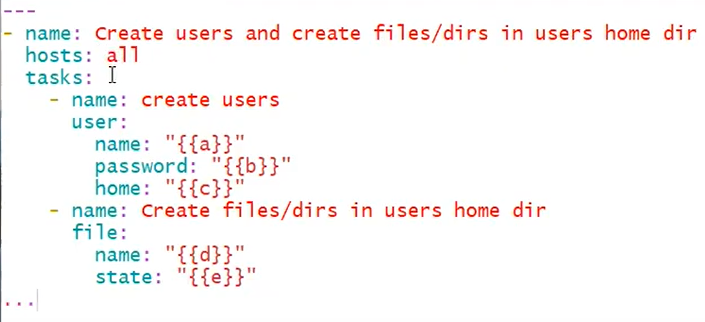
Global Scope Variables:

create a yml file.





Creating users and files using global scope variables:





**Play Scope variable:**

We can give the variables inside the yml file.



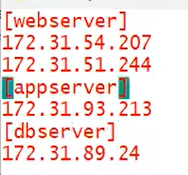


In playbook we have given instruction as installing tomcat9 but in variables, global scope variables are more prioritized than play scope variables. So even though we given variables in yml file the variables we declared as global scope variables gets executed and it uninstall tree.

HOST SCOPE VARIABLES: classified into two types:

* Variables to work on single host
* Variables to work on group of hosts

We can group them in hosts file where we saved the ip addresses of servers. The group name is written in square brackets.



One machine can be in one or more groups.

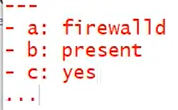
Creating user in one group of hosts.



We can combine two groups as a single group.

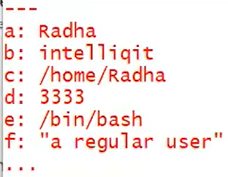
Create a directory in the folder where you have all the playbooks available.

Enter into the directory and create a file, create a file with group name.



Variables working on single host:

Make the server IP address as file name. store some data in it.



This file is created in host\_vars folder. Now come out of that folder and create a playbook.

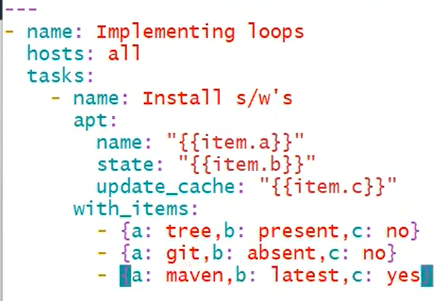
Priority wise first comes global scope variables, then Host scope variables and at last play scope variables.

**LOOPS**:

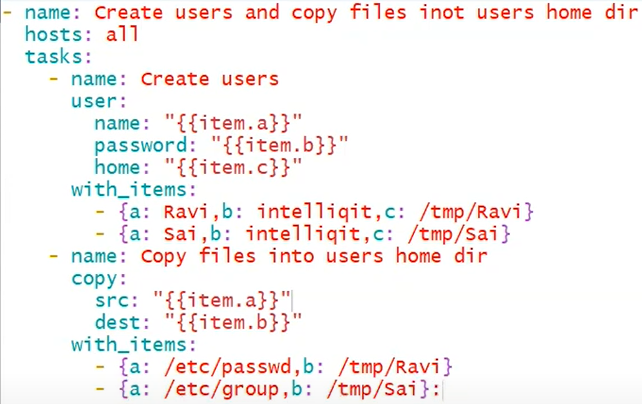
Way through which a module can be implemented multiple times.



In case of installing one application and unistalling one application.



Create users and copy files:



**TAGS**:

In a playbook if we want to selectively choose something to execute and remaining part to skip then we use tags in playbook. Tags create an alias name by which we can execute.

