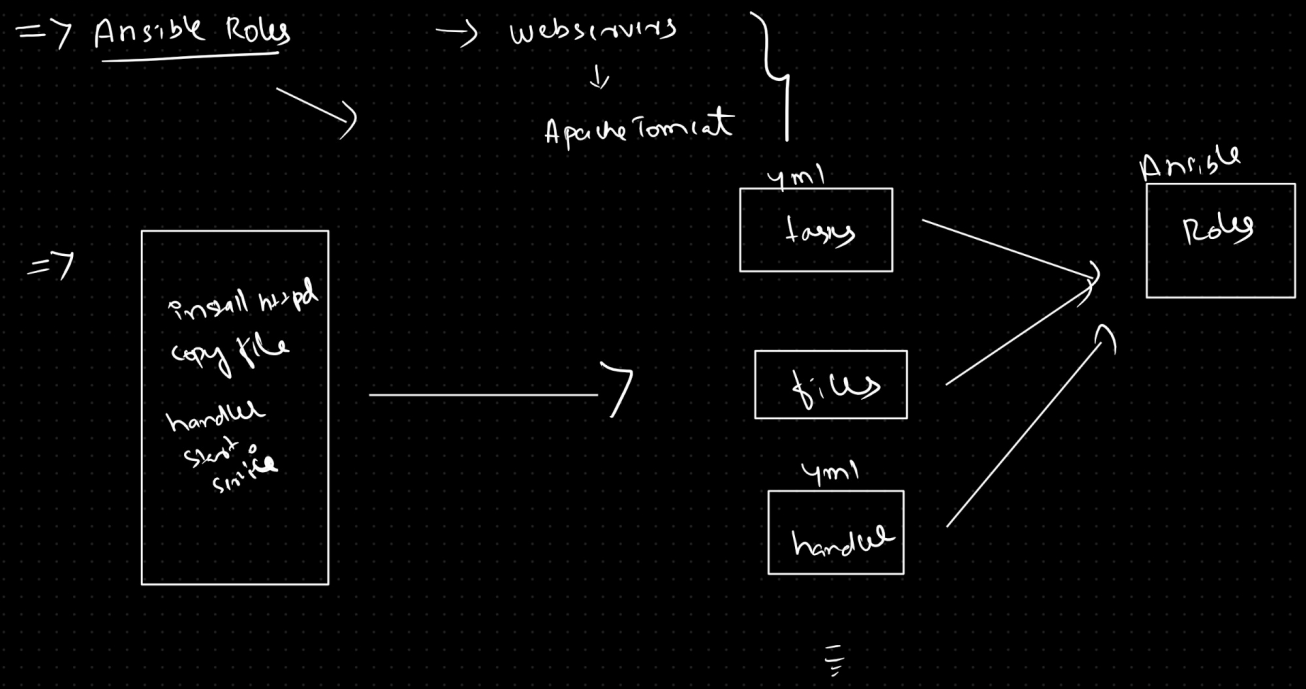


Whenever there are more tasks or more functionalities in our playbooks, it becomes difficult to manage and maintain.
Ansible Roles allows us to break down a complex playbook into separate , smaller chunks that can be co-ordinated by a central entry point.

Practicals with Ansible Roles

sudo su ansible

cd ~

pwd --> /home/ansible

mkdir roles

cd roles

ansible-galaxy init apache ---> apache role is created

(install tree)
tree apache

cd apache

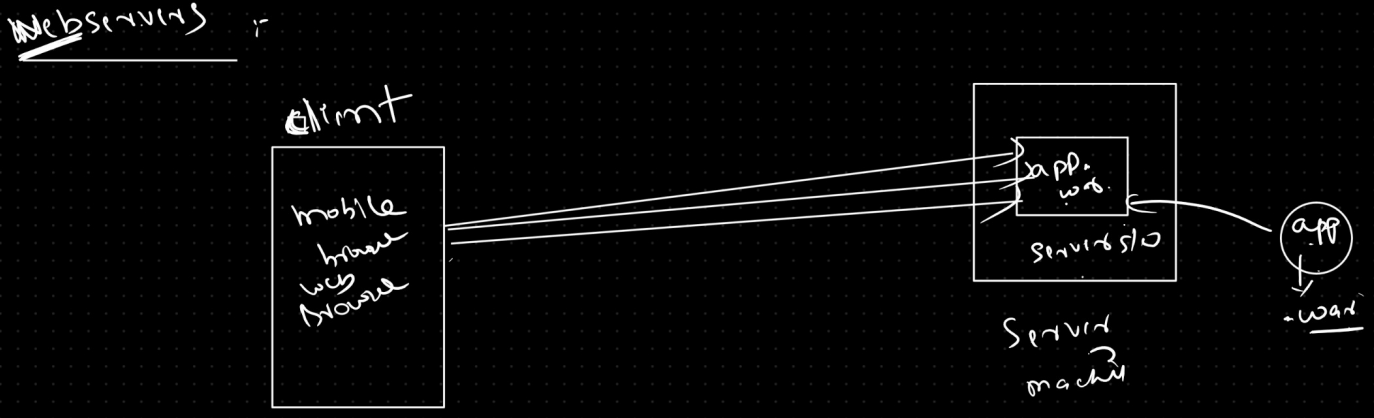
vi tasks/main.yml

```
---
# tasks file for apache
- name: httpd
  yum:
   name: httpd
   state: present
- name: copy index.html
  copy:
   src: index.html
   dest: /var/www/html/index.html
  notify:
    - restart apache
...
```

```
vi files/index.html

<h1>data</h1>
---
# handlers file for apache

- name: restart apache
  service:
   name: httpd
   state: restarted
...

cd ..
cd ..

vi runsetup.yml

---
- hosts: webservers
  become: true
  roles:
    - apache
...


 ansible-playbook runsetup.yml
```

Ansible Tower
==========
Ansible Tower in a paid s/w managed by Red Hat which provides web based UI to create/schedule/manage/execute playbook

---



Webservers
=========
Server Software is used to run our web application and User across the globe with internet can access our web applications by sending request to server

Users will use client s/w or web browsers to send request to server

Server is responsible to handle requests and response

There are many server softwares in the market to run our web apps

Tomcat, JBoss, Glassfish, WebLogic, IIS, WebSphere .....

To Run our web app server is mandatory

The Process of executing web applications by using sever is called deployment

Tomcat Server
=========
Its free and open source s/w which is developed by Apache organization

Tomcat server is developed using Java language to run tomcat , Java also has to be installed

Tomcat supports multiple operating systems

By Default tomcat runs on 8080 port number (If needed we can change the port number of Tomcat)

Tomcat Server Directory Structure
bin
Contains scripts to start and stop the server.

Windows: startup.bat, shutdown.bat

Linux: startup.sh, shutdown.sh

conf
Contains Tomcat configuration files.

Examples: server.xml, tomcat-users.xml

webapps
Deployment directory where .war files are placed for deployment.

logs
Stores all Tomcat server log messages.

lib
Contains required libraries (.jar files) for the Tomcat server.

temp
Temporary files generated by the server.

These can be safely deleted if needed.

Web Application Deployment Process
Create a Maven web application in the ec2-user home directory.

Package the project into a .war file using the Maven goal:

mvn clean package
Copy the generated .war file to the Tomcat webapps directory:

$ cp <app/target/warfile> <tomcat/webapps>
Start the Tomcat server using the script from the bin directory:

$ sh startup.sh
Note: Ensure port 8080 is allowed in the EC2 VM's Security Group (Inbound Rules).

Access your web application using the browser:

Tomcat Admin Console
Changing Default Port Number