

# DevOps Project -02

## Prerequisites:

1. Ansible server
2. Jenkins Server
3. Tomcat Server

## Objective:

Build will happen through Jenkins and deployment through ansible on tomcat server.

## Steps:

1. Developers will commit the changes to the code using git on github account
2. Jenkins will build and test the project for every commit made by developers and create distributable jar or war file.
3. Finally, ansible will deploy war file on tomcat server residing on ec2 instance.

## Project Workflow:



## Part-01 Integration Steps

Install "publish Over SSH"

- Manage Jenkins > Manage Plugins > Available > Publish over SSH

Enable connection between Ansible and Jenkins

- Manage Jenkins > Configure System > Publish Over SSH > SSH Servers
  - SSH Servers:
    - Hostname:<ServerIP>
    - username: ansadm
    - password: \*\*\*\*\*

Test the connection "Test Connection"

## Part-02 - Execute job to connect

create a copywarfile.yml on Ansible under /opt/playbooks

```
# copywarfile.yml
---
- hosts: all
  become: true
  tasks:
    - name: copy jar/war onto tomcat servers
      copy:
        src: /op/playbooks/wabapp/target/webapp.war
        dest: /opt/apache-tomcat-8.5.32/webapps
```

Add tomcat server details to /etc/ansible/hosts (if you are using other hosts file update server info there)

```
echo "<server_IP>" >> /etc/ansible/hosts
```

Create *Jenkins job*, Fill the following details,

- *Source Code Management:*
  - Repository: <https://github.com/ValaxyTech/hello-world.git>
  - Branches to build : \*/master
- *Build:*
  - Root POM:pom.xml
  - Goals and options : clean install package

- *Add post-build steps*
  - Send files or execute commands over SSH
    - SSH Server : ansible\_server
    - Source fiels: webapp/target/\*.war
    - Remote directory: //opt//playbooks
- *Add post-build steps*
  - Send files or execute commands over ssH
    - SSH Server : ansible\_server
    - Exec command ansible-playbook /opt/playbooks/copywarfile.yml

Execute job and you should be able to seen build has been deployed on Tomcat server.