Contents

[Project Name: 1](#_Toc67921152)

[**Description**: 1](#_Toc67921153)

[Flow Diagram: 1](#_Toc67921154)

[DevOps Tools Used 1](#_Toc67921155)

[Steps to Implement 2](#_Toc67921156)

[Jenkins Server 2](#_Toc67921157)

[Ansible Server 3](#_Toc67921158)

[Validation test 5](#_Toc67921159)

[Tomcat Configuration 5](#_Toc67921160)

[Output: 6](#_Toc67921161)

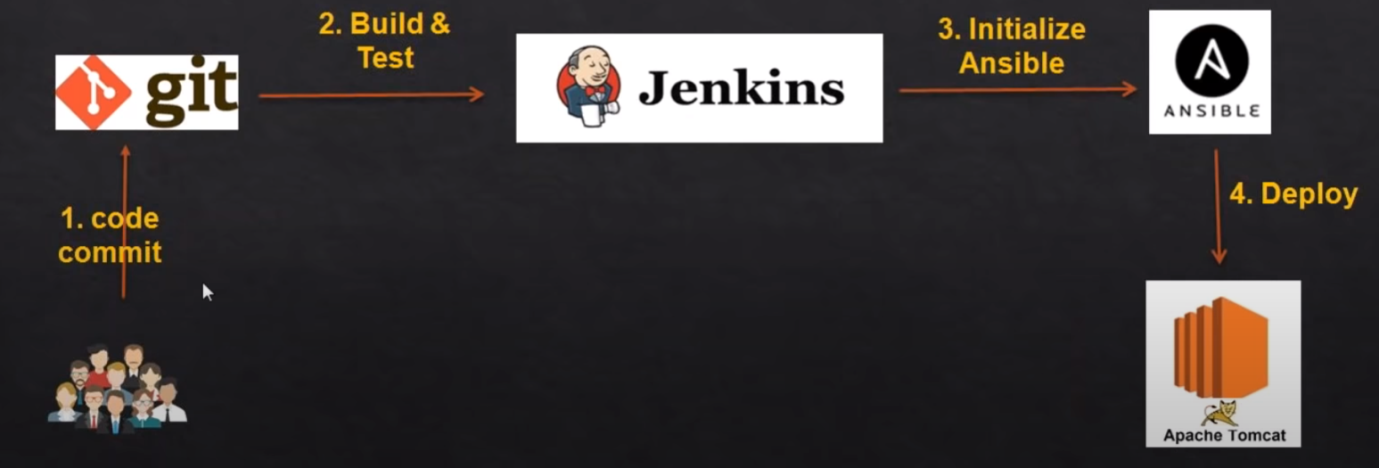
# Project Name:

Continuous Integration to deploy the applications in Tomcat using Ansible

# **Description**:

Create a Continuous Integration Job that Pulls the Code from GitHub repository, Builds the application with Maven, Copies the application package (in war/jar format) into Tomcat Server using Ansible.

# Flow Diagram:



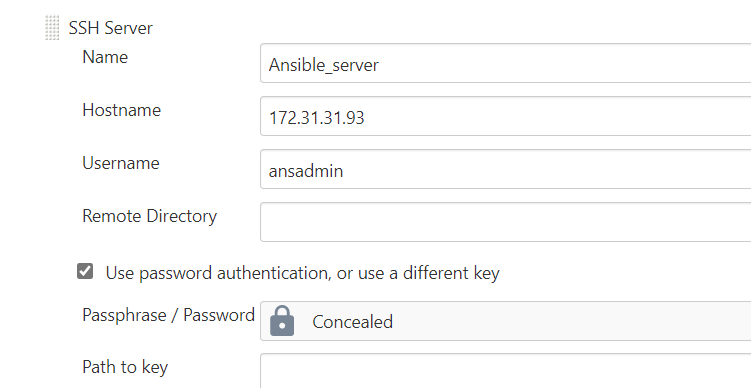
# DevOps Tools Used

* Git
* Maven
* Jenkins
* Ansible

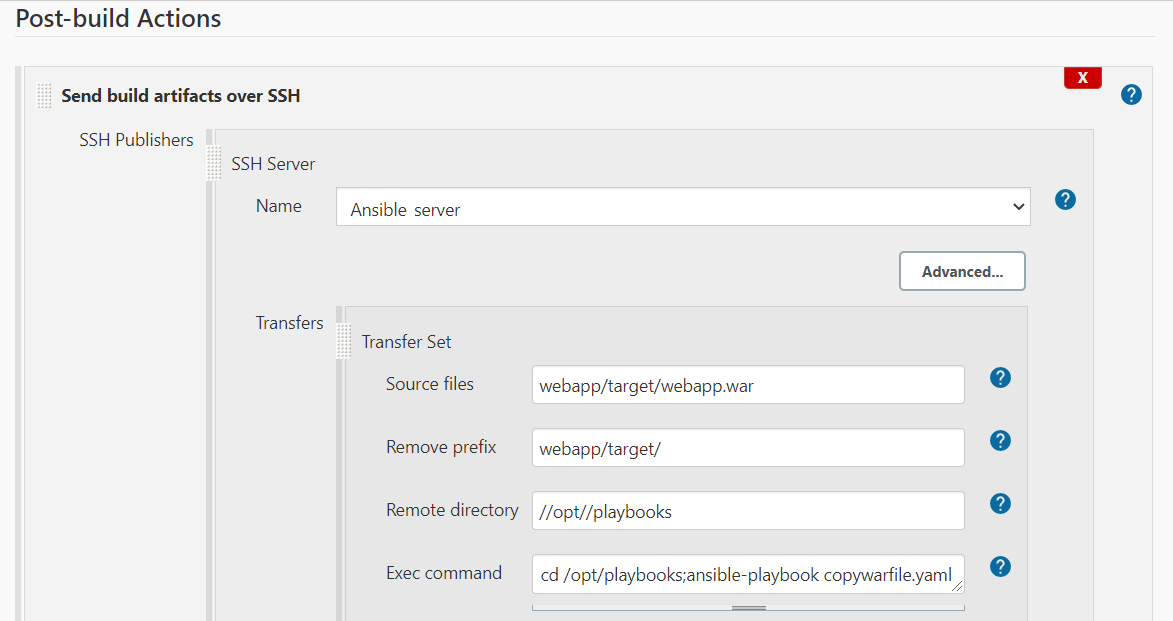
# Steps to Implement

## Jenkins Server

* Login to AWS Console
* Create EC2 Instance with Redhat Enterprise Linux 8 AMI
* Login to EC2 Instance from your local system using Git Bash or any other tools
* Install Git
* Install Maven
* Install Jenkins
  + Once Jenkins is installed, Open the http://<Public-IP of the Server>:8080 in a browser
  + Unlock the Jenkins by following the instructions
  + Install Suggested Plugins
  + Create Admin user
  + Then once you logged into Jenkins
    - Go to Manage Jenkins 🡪 Manage Plugins 🡪 Available Tab
    - Search for Publish Over SSH then install the Plugin
  + Mange Jenkins 🡪 Configure System
    - Add SSH Server
    - Configure Ansible Server Details
      * Hostname: Private IP of Ansible server
      * Username
      * Password



* + Integrate Git, Some of the Git Projects to be considered
    - <https://github.com/daticahealth/java-tomcat-maven-example.git>
  + Integrate Maven
  + Add Post Build Action to Integrate with Ansible



* Trigger the Job

## Ansible Server

1. Install Python latest version (on Control node and Managed host)

yum install python3 -y

1. By default, python3 is the command to run python commands. to use just python, use "alternatives" command. (on Control node and Managed host)

alternatives --set python /usr/bin/python3

1. Check for Python version

python --version

1. Install python-pip package manager (on Control node)

yum -y install python3-pip

1. Create a new user for ansible administration & grant admin access to the user (on Control node and Managed host)
2. useradd ansadmin

passwd ansadmin

1. Below command adds ansadmin to sudoers file. But we strongly recommended using "visudo" command if you are aware vi or nano editor. (on Control node and Managed host)

echo "ansadmin ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

1. Using key-based authentication is advised. If you are still at the learning stage use password-based authentication (on Control node and Managed host)
2. # sed command replaces "PasswordAuthentication no to yes" without editing file
3. sed -ie 's/PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd\_config

sudo service sshd reload

**Install Ansible as a ansadmin user (on Control node)**

su - ansadmin

pip3 install ansible --user

Note: Ansible must be installed as a user (here ansadmin)

1. check for ansible version

ansible --version

1. Log in as a ansadmin user on master and generate ssh key (on Control node)

ssh-keygen

1. Copy keys onto all ansible managed hosts (on Control node)

ssh-copy-id <target-server>

### Validation test

1. Create a directory /etc/ansible and create an inventory file called "hosts" add Tomcat server node IP address in it.
2. Run ansible command as ansadmin user it should be successful (Master)

ansible all -m ping

1. Write a playbook to copy jar/war file on to tomcat server
   1. cd /opt/playbooks
   2. the File named copywarfile.yaml

**# copywarfile.yml**

**---**

**- hosts: all**

**become: true**

**tasks:**

**- name: copy jar/war onto tomcat servers**

**copy:**

**src: /opt/playbooks/webapp.war**

**dest: /opt/apache-tomcat-8.5.64/webapps**

## Tomcat Configuration

1. Tomcat configuration in Linux System

* Create EC2 Instance (Red Hat EL 8 OS)
* Login to EC2 from Git Bash
* sudo yum install java-1.8\*
* java -version
* Edit /etc/profile to set JAVA\_HOME for all users
  + Export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-2.el8\_3.x86\_64
* sudo yum install wget -y
* cd /opt
* wget https://mirrors.estointernet.in/apache/tomcat/tomcat-8/v8.5.64/bin/apache-tomcat-8.5.64.tar.gz
* tar -xvzf apache-tomcat-8.5.64.tar.gz
* chmod +x /opt/apache-tomcat-8.5.64/bin/startup.sh
* chmod +x /opt/apache-tomcat-8.5.64/bin/shutdown.sh
* Create ansadmin user
  + adduser ansadmin
  + passwd ansadmin
* Enable password less authentication
  + Add user in to sudoers file by executing
    - visudo
      * Add **ansadmin ALL=(ALL) NOPASSWD:ALL**
  + **Vi /etc/ssh/sshd\_config**
    - **Change the PasswordAuthentication yes**
    - **Service sshd restart**

# Output:

* Once the Build Job is executed successfully from Jenkins
  + The war file gets copied from Jenkins Server to Ansible Server
  + Ansible Server copies the war file into Tomcat server and starts the Tomcat
  + Copy Public IP of Tomcat Server and access it from Browser to see if you can access the application