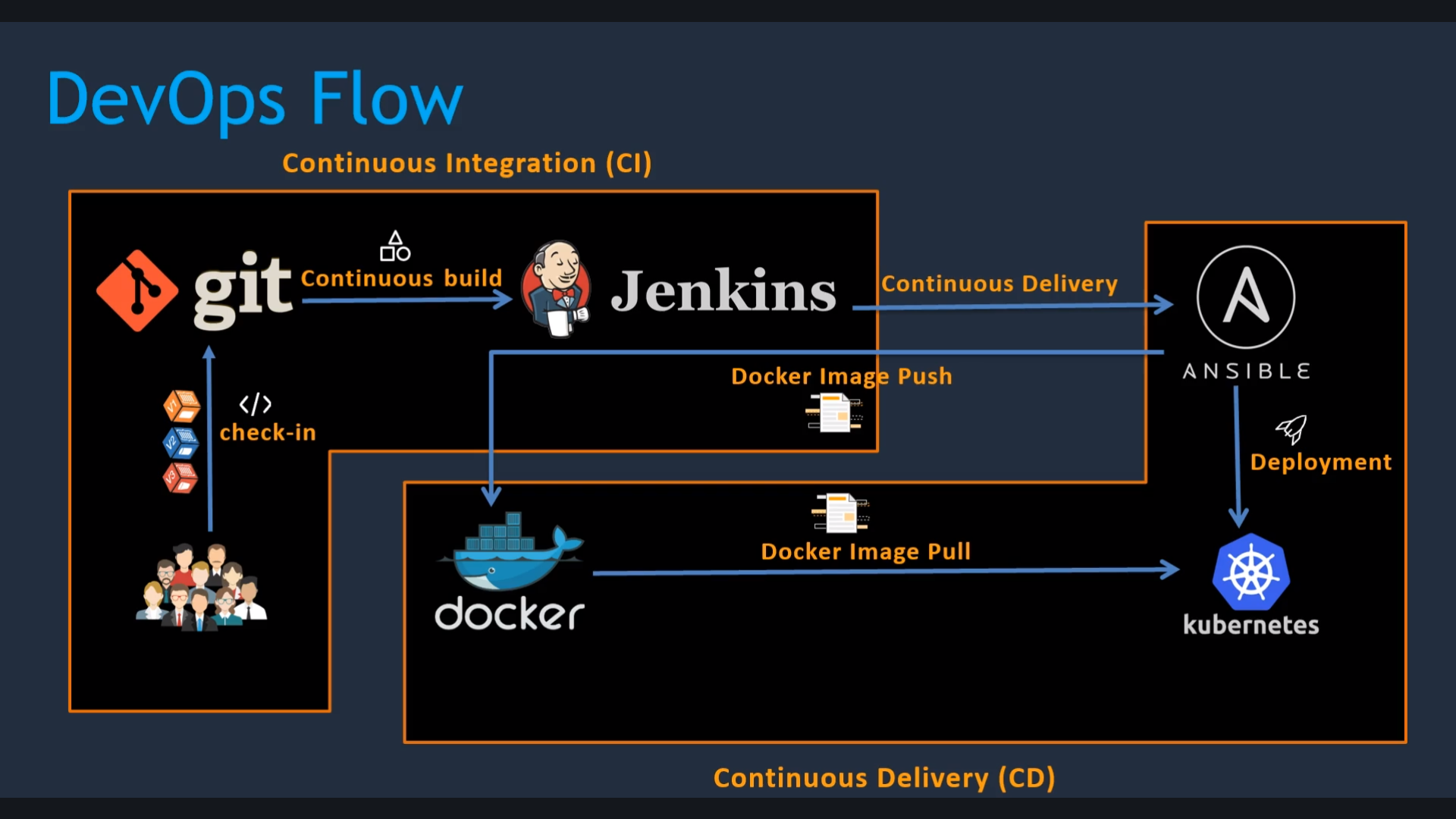
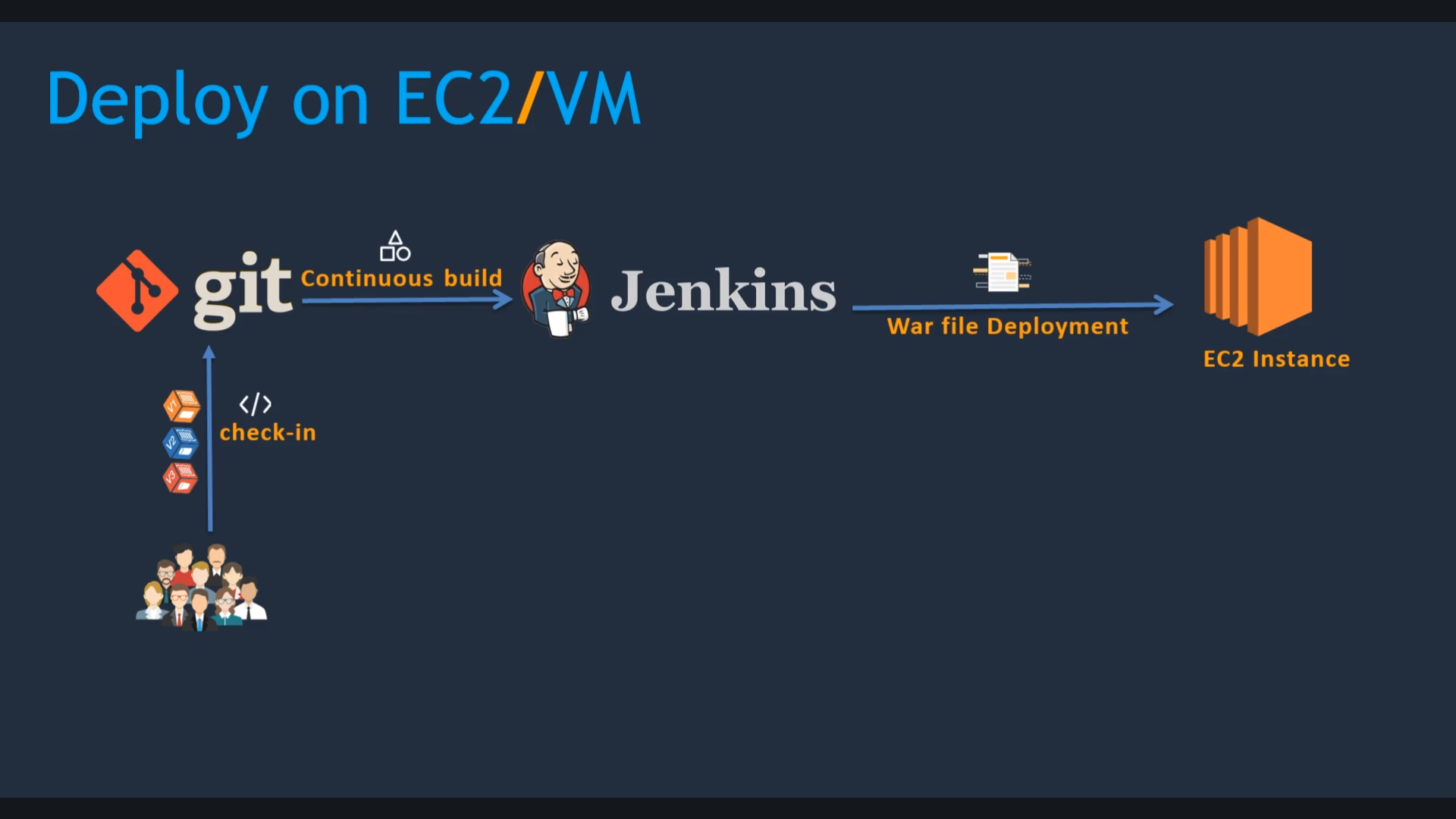
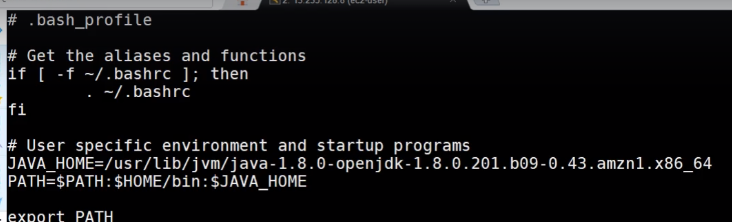
**Create DevOps CI/CD pipelines using Git, Jenkins, Ansible, Docker and Kubernetes on AWS**





->First we will install jenkins

1. Launch the Ec2 instance and open 8080 port
2. Install the java version 1.8\* and remove if any old version of java is there
3. find / -iname java-1.8\* (Find the java jdk file installed path)
4. /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-0.48.amzn1.x86\_64
5. vi .bash\_profile (Set the java home path)



* echo $JAVA\_HOME (To check the java path is configured or not)
* /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-0.48.amzn1.x86\_64 (It should display this path)
* Go to [**https://jenkins.io/download/**](https://jenkins.io/download/)
* sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
* sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
* yum install Jenkins
* **Service Jenkins status** 🡪To check the Jenkins status
* Service jenkins start ->Start the jenkins service
* **35.176.246.5:8080 ->Try the public ip:8080**

cat /var/lib/jenkins/secrets/initialAdminPassword

* Paste the password in the URL
* Change the password once setup is done
* Manage jenkins🡪Global tool configuration -->Add jdk🡪Enter the JAVA\_HOME path

Run Frist Jenkins Job

* GitHub plugin is mandatory
* Install the github plugin and in global configuration tool add github

🡪Maven setup document is there in the github-Simple devops project

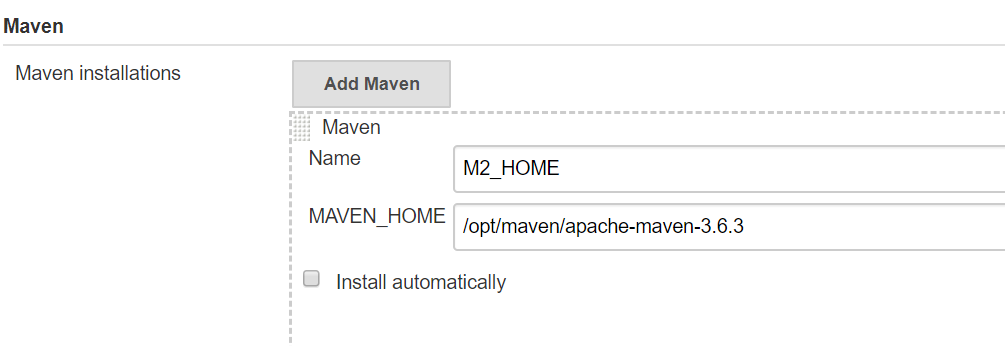
- Search for Maven download <https://maven.apache.org/download.cgi>

-Copy the below binary URL (Binary tar.gz archive apache-maven-3.6.3-bin.tar.gz)

- Right click **apache-maven-3.6.3-bin.tar.gz** on and copy the URL

- Follow the document in Github

- tar -xvzf apache-maven-3.6.1-bin.tar.gz

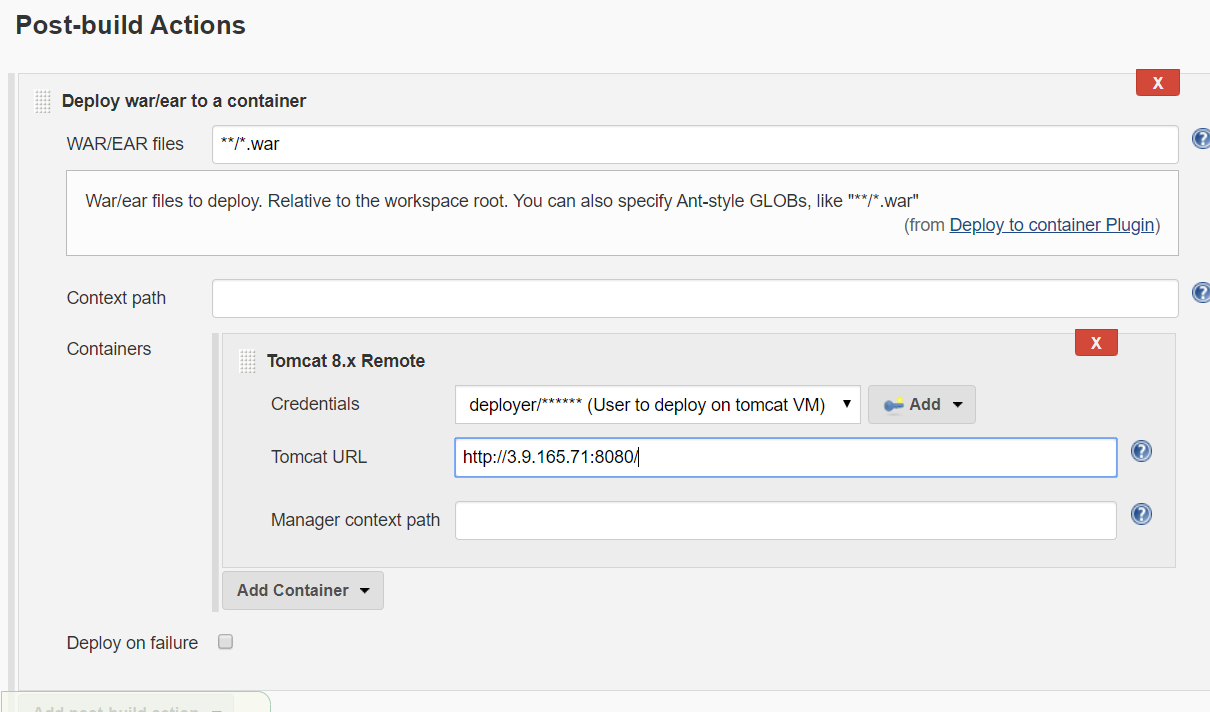
* Check the env variables whether configured correct or not using -echo $M2 ,echo $M2\_HOME
* After that check the maven version mvn –version
* Install the Maven plugins in Jenkins
* **Maven integration & Maven invoker** plugin
* Manage Jenkins > Global Tool Configuration > Maven
* 
* IN Jenkins while running the jobs Code will copy into **Workspace directory**
* Workspace🡪Webapp/Targets -🡪 our outcome of Build will be stored
* **/var/lib/jenkins/workspace**/ (Path in CLI)

**Tomcat Server setup**

* <https://github.com/sanjaykumarsj/Simple-DevOps-Project/blob/master/Tomcat/tomcat_installation.MD>
* find / -iname context.xml
* vi /opt/tomcat/webapps/manager/META-INF/context.xml
* comment the values line

🡪Install **Deploy to container plugin** on Jenkins

🡪Post build actions select **Deploy to war/ear to a container**



* <http://3.9.165.71:8080/webapp/> (TomcatURL/webapp)
* Try to access this file to view the webpage

**🡪Build triggers🡪poll SCM 🡪 \* \* \* \* \* (Runs Every minute)**

* **cd webapp/src/main/webapp/index.jsp (To change the Webapp code)**
* change the code->git status->git add.->git commit-> **git push origin master**



🡪<https://github.com/sanjaykumarsj/Simple-DevOps-Project/blob/master/Docker/Docker_Installation_Steps.MD>

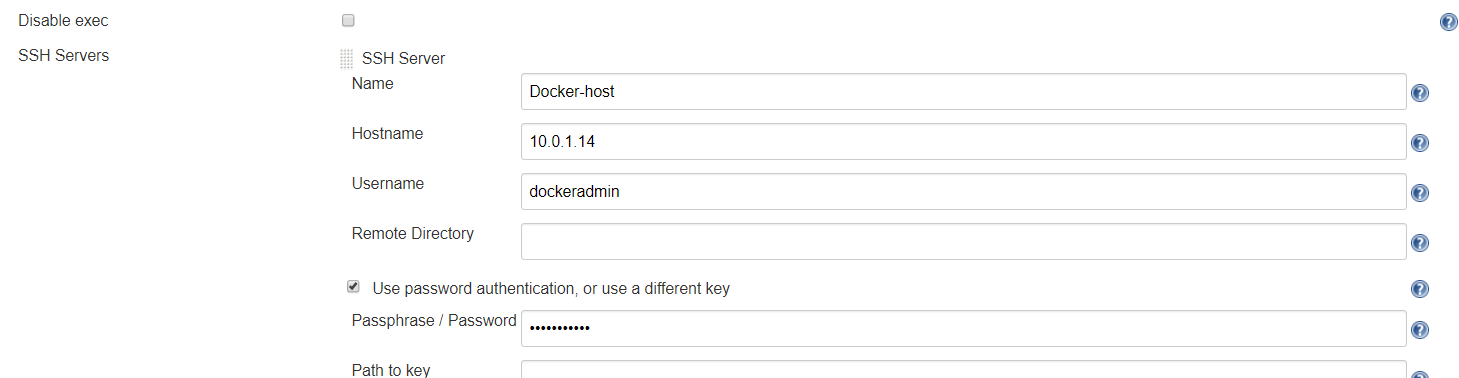
* **service docker status** (to check the status of docker)
* **service docker start**
* **docker ps**
* **Search for docker hub**
* Outside all users keep their docker files in private in docker hub
* Always use official images
* **docker pull tomcat:latest**
* **docker image ls / docker images (to list the docker images)**
* 
* **docker run --name tomcat-container -p 8080:8080 tomcat:latest**
* <http://dockerip:8080/>
* Docker ps -a
* Docker rm <container id> (to remove the docker)
* **docker run -d --name tomcat-container -p 8080:8080 tomcat:latest (in wont show in the frontend)**
* **docker exec -it tomcat-container /bin/bash (Login to Tomcat container)**
* After login check webapps folder
* If you face http error go to one version old of tomcat
* **docker run -d --name tomcat-8 -p 8081:8080 tomcat:9.0**
* **docker image rm <container id> (delete the docker image)**
* Install [Publish Over SSH](https://plugins.jenkins.io/publish-over-ssh) plugin in jenkins
* Create a user (Dockeradmin) in docker server
* **cat /etc/group (This user must be in docker group)**
* **id dockeradmin**
* **usermod -aG docker dockeradmin (add the dockeradmin to docker group)**

Output

[root@ip-10-0-1-14 ec2-user]# id dockeradmin

uid=501(dockeradmin) gid=501(dockeradmin) groups=501(dockeradmin),497(docker)

* **Jenkins->manager jenkins->configure system->Publish over ssh->add SSH servers->copy the ip address( Docker server Eth0 )-> Test the connection**



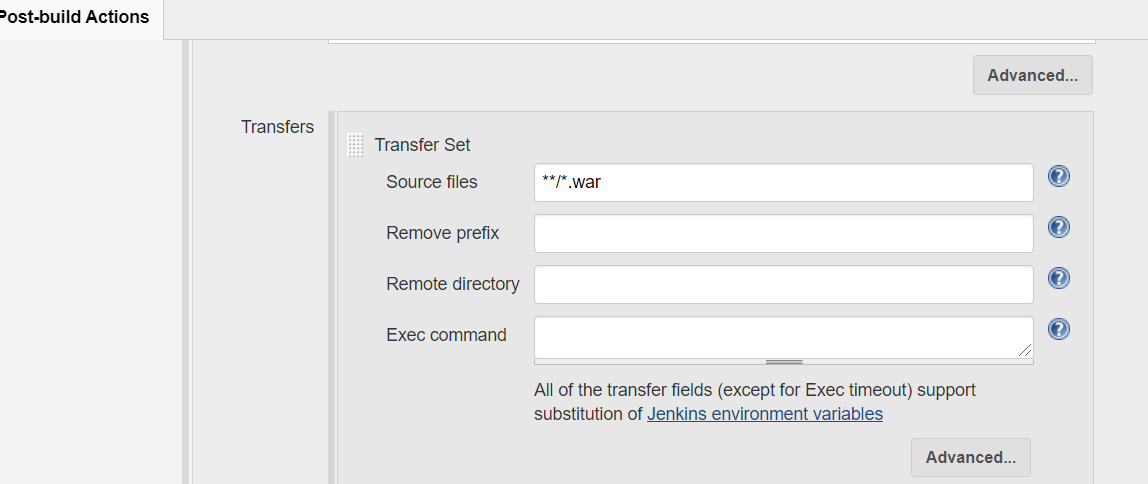
* **vi /etc/ssh/sshd\_config -> enable Password authentication to Yes**

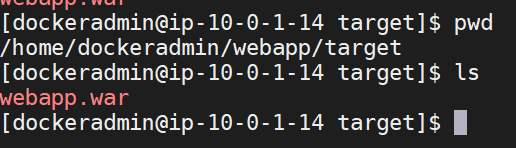
# EC2 uses keys for remote access

* **PasswordAuthentication yes**
* **service sshd reload**
* Then test the configuration on jenkins

-------------------

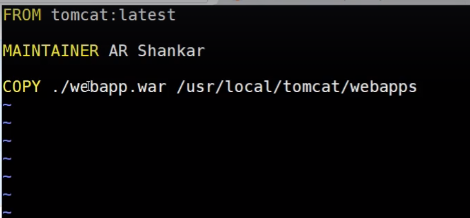
Jenkins Job to copy artifacts on to DockerHost

* --> Create a job in Jenkins->copy from old job->Remove poll SCM->Remove postbuild actions
* 🡪Postbuild actions->Send build artifacts over ssh ->
* 



* /home/dockeradmin/webapp/target/webapp.war (Artifacts location)

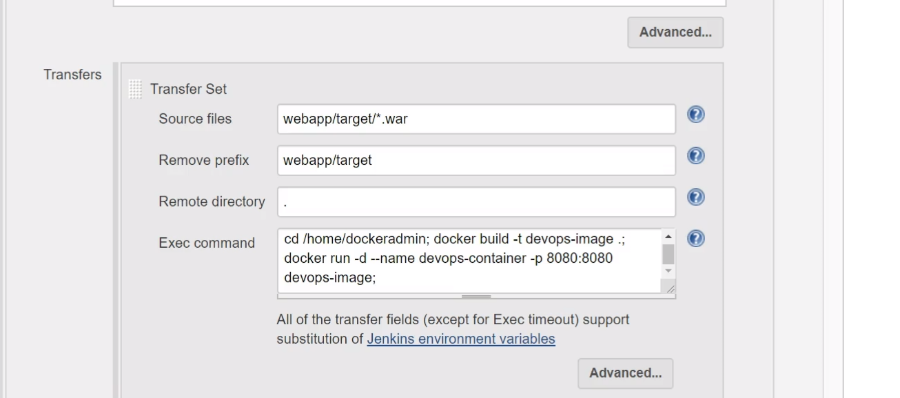
**Create a simple docker file**



* **/usr/local/tomcat/webapps (default path in containers)**
* **docker build -t devops-project . (to build the docker image t=tags .=check in current directory)**
* After that create a container out of it
* **docker run -d --name devops-container -p 8080:8080 devops-project**
* **docker run -d --name <name of the container> -p 8080:8080 <image name>**
* <http://Docker-IP:8080/webapp/> (to access the URL)

Deploy a war file on Docker container using Jenkins

* -> Jenkins server->add project-> Deploy\_on\_Container ->Copy from deploy on docker->Post build actions->
* cd /home/dockeradmin; docker build -t devops-image .; docker run -d --name docker-container -p 8080:8080 devops-image;



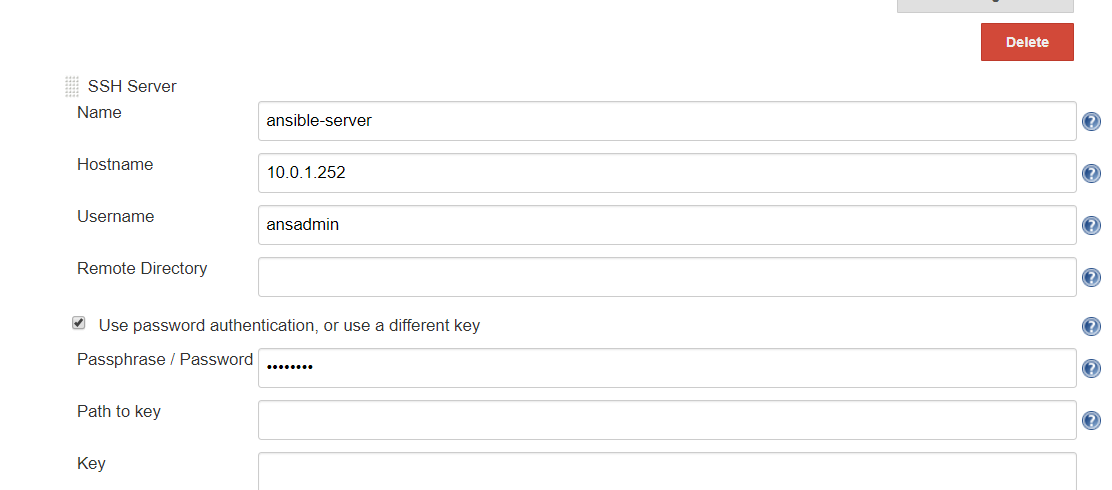
* 🡪 <http://35.177.51.42:8080/webapp/>
* If you build again you will get errors. To overcome we have to use **ansible**

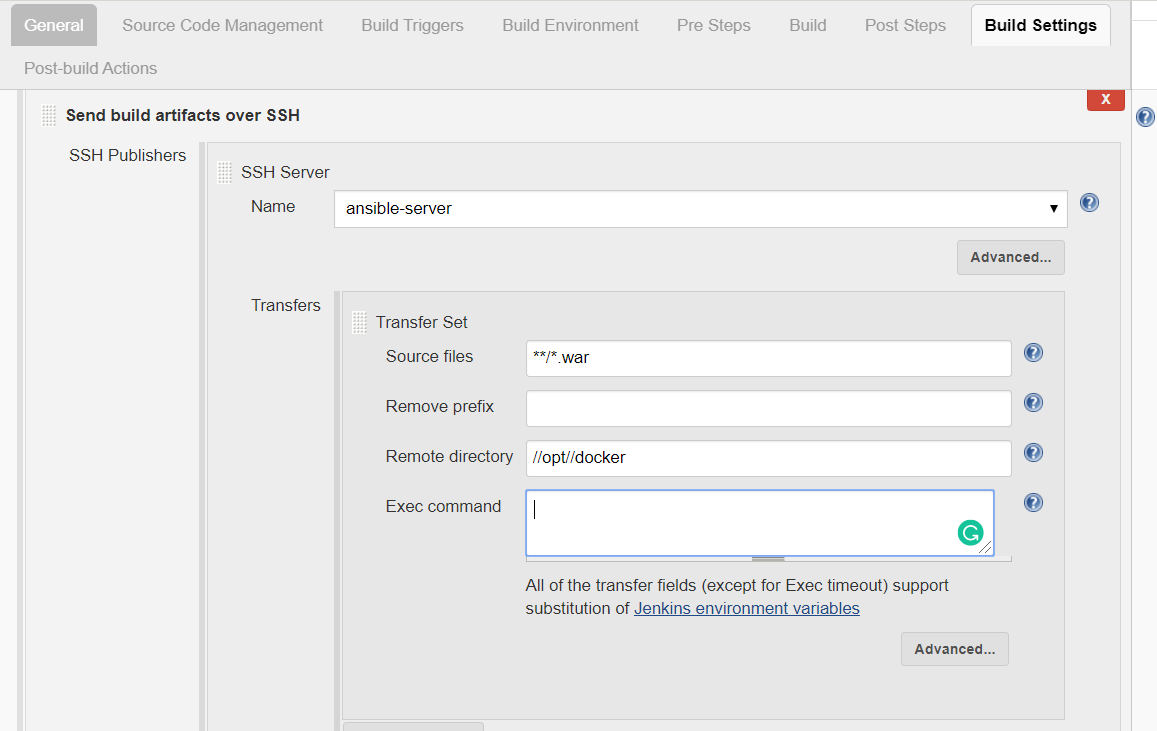
**Ansible Setup**

* <https://github.com/sanjaykumarsj/Simple-DevOps-Project/blob/master/Ansible/Ansible_installation.MD>
* Mkdir /etc/ansible
* Install docker on ansible server
* Start docker services
* Vi /etc/ssh/sshd\_config 🡪 Password authentication Yes
* Service sshd reload
* /home/ansadmin/.ssh
* Go to docker server->Create a user(ansadmin)->
* Goto ansbible server->change to ansadmin user->
* ssh-copy-id ansadmin@<Docker-server ip eth0>
* check the connection ssh <server ip>
* cd /etc/ansible
* Create hosts file🡪add ip address of docker,localhost
* **ansible all -m ping** (ping and check )

**Integrate ansible with jenkins**

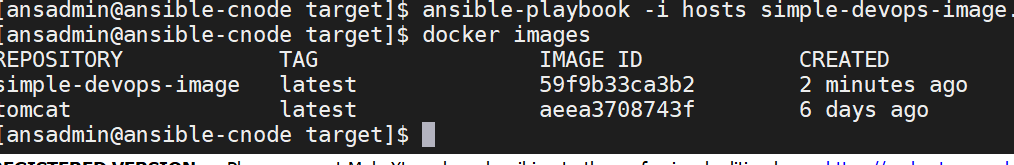
🡪 Manage Jenkins-> configure system->publish over ssh🡪 add



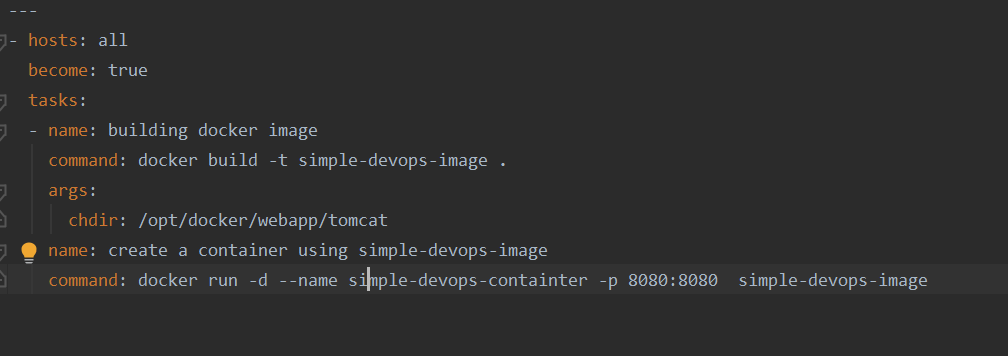
* <https://github.com/sanjaykumarsj/Simple-DevOps-Project/tree/master/Jenkins_Jobs>
* <https://github.com/sanjaykumarsj/Simple-DevOps-Project/blob/master/Jenkins_Jobs/Deploy_on_Container_using_Ansible.MD>
* Go to Jenkins server-> create a job-> Deploy\_on\_Container\_using\_ansible🡪Copy from deploy on container->Post build actions-> ssh server->ansible server
* Create a directory docker in ansible server (cd /opt🡪 mkdir docker) give full rights to ansadmin
* /opt/docker (Using this directory to save the artifacts)
* 
* Save->build now-> Check in ansible server webapp.war file is created or not
* [root@ansible-cnode target]# pwd
* /opt/docker/webapp/target
* [root@ansible-cnode target]# ls
* webapp.war

Creating an ansible playbook

* Go to ansible server -> vim Dockerfile 🡪 Lines updated in the documentation -> create a docker image out of this 🡪 for that we need to automate using ansible -> vim simple-devops-image.yml 🡪write the code🡪
* Create a hosts file in same directory ( /opt/docker/webapp/target )
* **ansible-playbook -i hosts simple-devops-image - -check** (to check if any errors)
* **ansible-playbook -i hosts simple-devops-image.yml**
* After check the **docker images**



* Make docker container out of this image
* vim simple-devops-project.yml



* docker ps -a