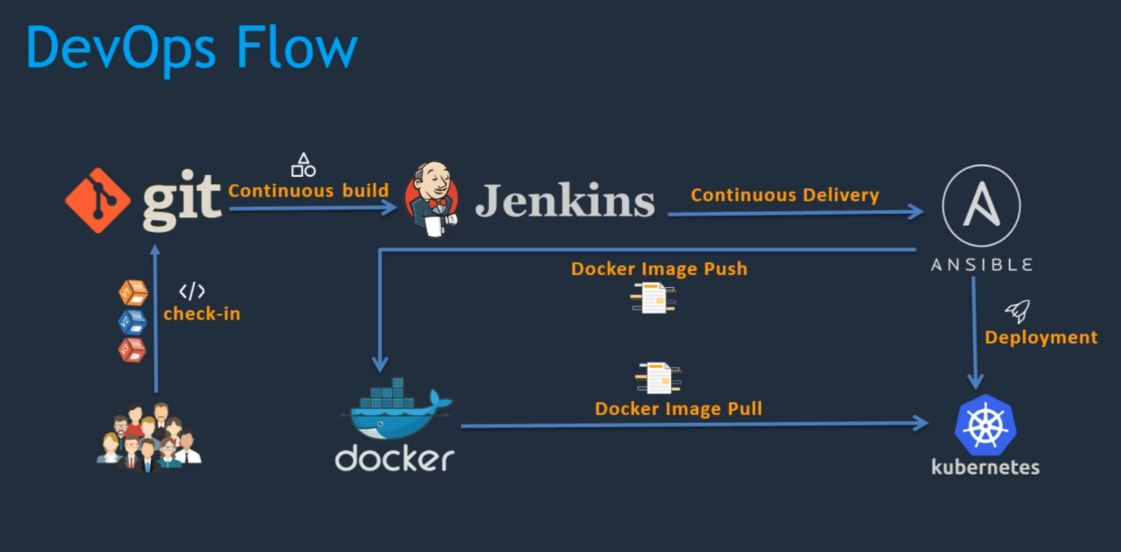
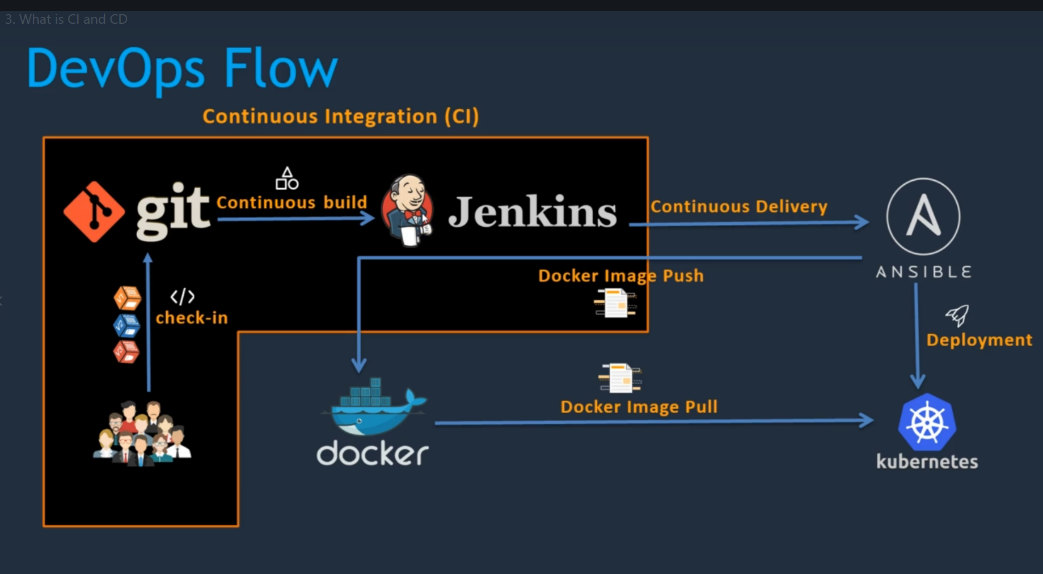
**CI/CD Course**

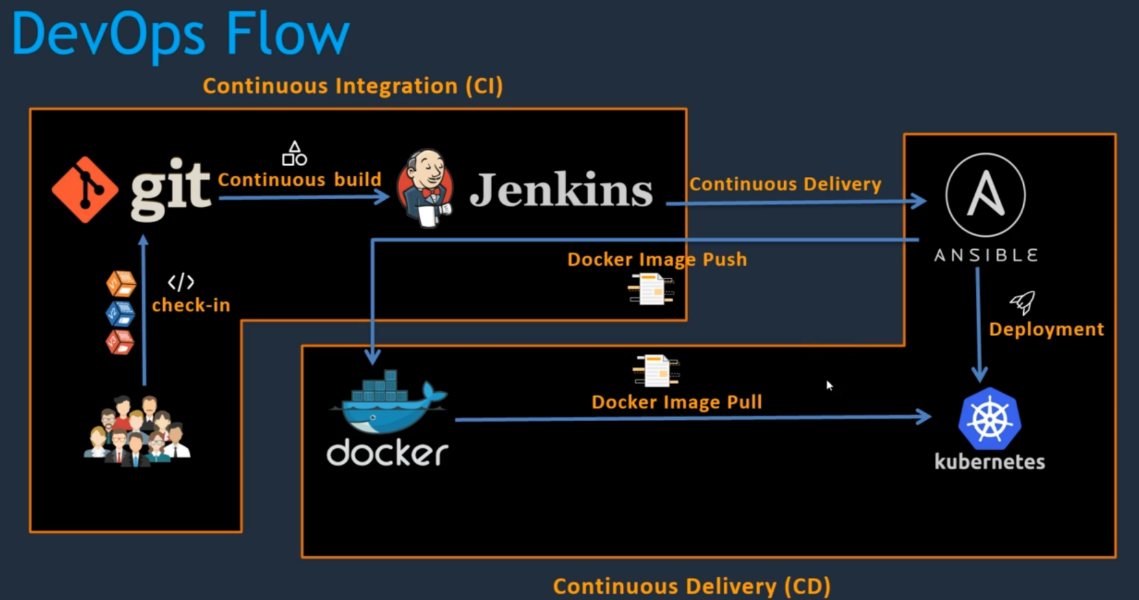
1. **URL :** <https://www.udemy.com/course/valaxy-devops>
2. **CI CD Flow :**

****

1. **Vid3: What is CI and CD:**
   1. **CI Flow: Continuous Integration**

****

* 1. **CD Flow: Continuous Delivery and Continuous Deployment**

****

1. **Vid4: Resources and CI/CD:**
   1. **Github Proj:** <https://github.com/yankils/hello-world>
   2. **Github DevOps Proj :** <https://github.com/yankils/Simple-DevOps-Project>
2. **Vid6 : Jenkins Installations:**
   1. **1- Create EC2 instance** 
      1. Add tag - Name : JenkinsServer
      2. Security Group ->
         1. Name : DevOps\_SG
         2. Allow Http 8080 - http rule
      3. Download the SecurityKey
      4. Launch it
      5. Copy IP address
   2. **2- Install and Luanch MobaXterm**
      1. Session -> SSH
      2. Use the SecurityKey (down in the bottom)
      3. use **ec2-user** -> Connect
   3. **3- Linux**
      1. Go to the root -> **sudo su** -
      2. java -version -> 1.7 => (Remove it) -> yum remove java-1.7.\* ->(clear)
      3. Install java 1.8 -> yum install java-1.8.\*
   4. **4-Setting Java home Path:**
      1. Go to home directory >cd ~
      2. Use this command to get the JVM dir path
         1. **>** find /usr/lib/jvm/java-1.8\* | head -n 3
      3. Copy the address like “/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.252.b09-2.51.amzn1.x86\_64/jre”
      4. Open the “.bash\_profile” file **>** vi .bash\_profile - (use Insert key)
         1. JAVA\_HOME= <paste above path>
         2. PATH =$PATH:$HOME/bin:$JAVA\_HOME
      5. **Note:** You need to logout and login to have >echo $JAVA\_HOME effect
      6. Logout > exit
      7. Login >sudo su –
      8. Now you can do echo $JAVA\_HOME
   5. **Download and Install Jenkins:**
      1. First create Jenkins Repo on local with following command. Find it on Jenkins site > RedHat Linux

> sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

* + 1. And then import a download repo key for Jenkins with following command

> sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

* + 1. Then run **> yum install jenkins**
  1. **Check the Jenkins Service & Open the page**
     1. **> service jenkins status**
     2. **> service jenkins start**
     3. Now go to EC ip address with port 8080 on your local browser
        1. <http://3.83.65.37:8080/>
     4. Cat the password file : /var/lib/jenkins/secrets/initialAdminPassword
  2. **Setting JAVA\_HOME path :**
     1. **Manage Jenkins 🡪 Global Tool Configuration**
     2. **JDK 🡪** 
        1. **Name: JAVA\_HOME**
        2. JAVA\_HOME : /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.252.b09-2.51.amzn1.x86\_64
     3. Save it

1. **Vid8: Git Installation and Plugin:**
   1. **Change the hostname:**  change the hostname and exit and login back again as sudo su -
      1. **>hostname** Jenkins
   2. **>** yum install git -y
   3. Go to Jenkins 🡪 Manage Plugins 🡪 **github**
   4. Configure github in Global Tool Configurations
      1. Specify git path
2. **Vid9 : Maven Setup:**
   1. **Copy the link from Maven site :** For file [apache-maven-3.6.3-bin.tar.gz](https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz) copy the link address
   2. **Goto EC2 console 🡪 cd /opt**
   3. **> wget <copy/paste the above link>**
   4. Unzip it using command **> tar -xvzf** apache-maven-3.6.3.bin.tar.gz
   5. Move it to unzipped dir to “maven” dir
      1. **>mv** apache-maven-3.6.3.x maven
   6. Set the M2 and M2\_HOME path in .bash\_profile file
      1. > vi ~/.bash\_profile
      2. **M2\_HOME**=/opt/maven
      3. **M2**=/opt/maven/bin
      4. Also add them in the path > Path:$M2:$M2\_HOME
   7. Now **install the following two Jenkins plugins**
      1. Maven Integration
      2. Maven Invoker
   8. Configure Maven in Global Tool Configuration using M2\_HOME
3. **Vid10: Create First Maven Jobs :**
   1. **Create New Maven Job(Item)**
   2. **Use Hello world project github:**
      1. **URL :** [**https://github.com/yankils/hello-world.git**](https://github.com/yankils/hello-world.git)
   3. **Goals**
      1. 🡪 **clean install package**
   4. Run the job
   5. To check the workspace on linux you can go to
      1. cd **/var/lib/Jenkins/workspace/**
      2. You’ll find “My\_first\_Mave\_Build” ..

**Section4 : Integration Docker in CI/CD piepline**

1. **Vid14 : Docker Setup:**
   1. **Create another EC2 Server as above:**
      1. Using same key and Security Group permissions
      2. Login to it using MobaXterm with same key
      3. >hostname docker-host
   2. Install Docker:
      1. > yum install docker -y
      2. **>**  service docker status
      3. **>**  service docker start
   3. **Docker hub:**
      1. **Note** : Go to docker hub and search for tomcat
      2. **>** docker **pull** **tomcat:latest**
      3. **>** docker **images** 🡪 to check the images
   4. **Creating & Running Container:**
      1. **>** **docker** **run --name devop-container -p 8080:8080 tomcat**
      2. Now check on the local browser 🡪 http:// 54.174.247.119:8080
      3. **Note: If you see 404 error then webapp at following location is empty**
         1. **To get into container >** docker exec -it <container> /bin/bash
         2. **/usr/local/tomcat/webapps**
         3. **>** cd /usr/local/tomcat/webapps.dist 🡪 this dist file has all the files
         4. **> cp -R \* ../webapps 🡪 To copy all files from webapp.dist to webapp folder**
      4. Now check on the local browser 🡪 http:// 54.174.247.119:8080
2. **Vid16 : Integrating Dockers with Jenkins:**
   1. **Note:**  In this section we’ll use Jenkins SSH plugin to copy .war file to docker host
   2. **Install “Publish Over SSH” Jenkins plugin:**
   3. **Create user on docker-host:**
      1. **>** useradd dockeradmin
      2. **>**  passwd dockeradmin
      3. **This user must be docker group:**  Add it to docker group
         1. **> cat**  /etc/group 🡪 it will show all the groups including “docker”
         2. **>**  usermod -aG docker dockeradmin
         3. **>** id dockeradmin
         4. **> ip addr**  🡪 To give it in Jenkins System Config (use the No2 ip address)
      4. **Go to Jenkins 🡪 Configure System:**
         1. **SSH Servers:**
            1. **Name :**  docker-host
            2. **Hostname:**  172.31.90.44
            3. **Click on Use password**
            4. **Password :**  <dockeradmin password>
            5. **Test Configuration 🡪** It will fail due to sshd\_config has not given passwordAuthentication
         2. **Go to docker-host for password Authentication access permission:**
            1. **We need to edit sshd\_config**

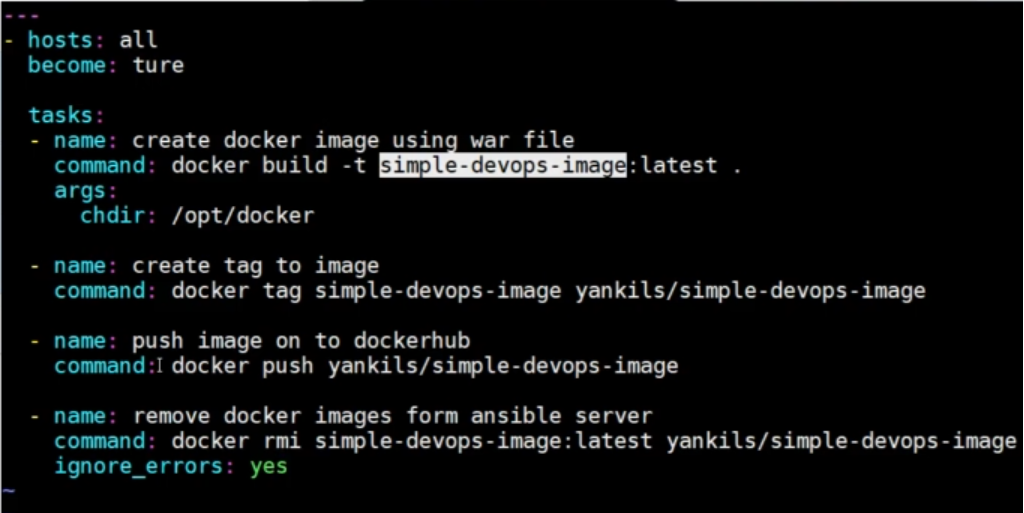
> vi /etc/ssh/sshd\_config

edit 🡪 passwordAuthenticaton yes 🡪 save file

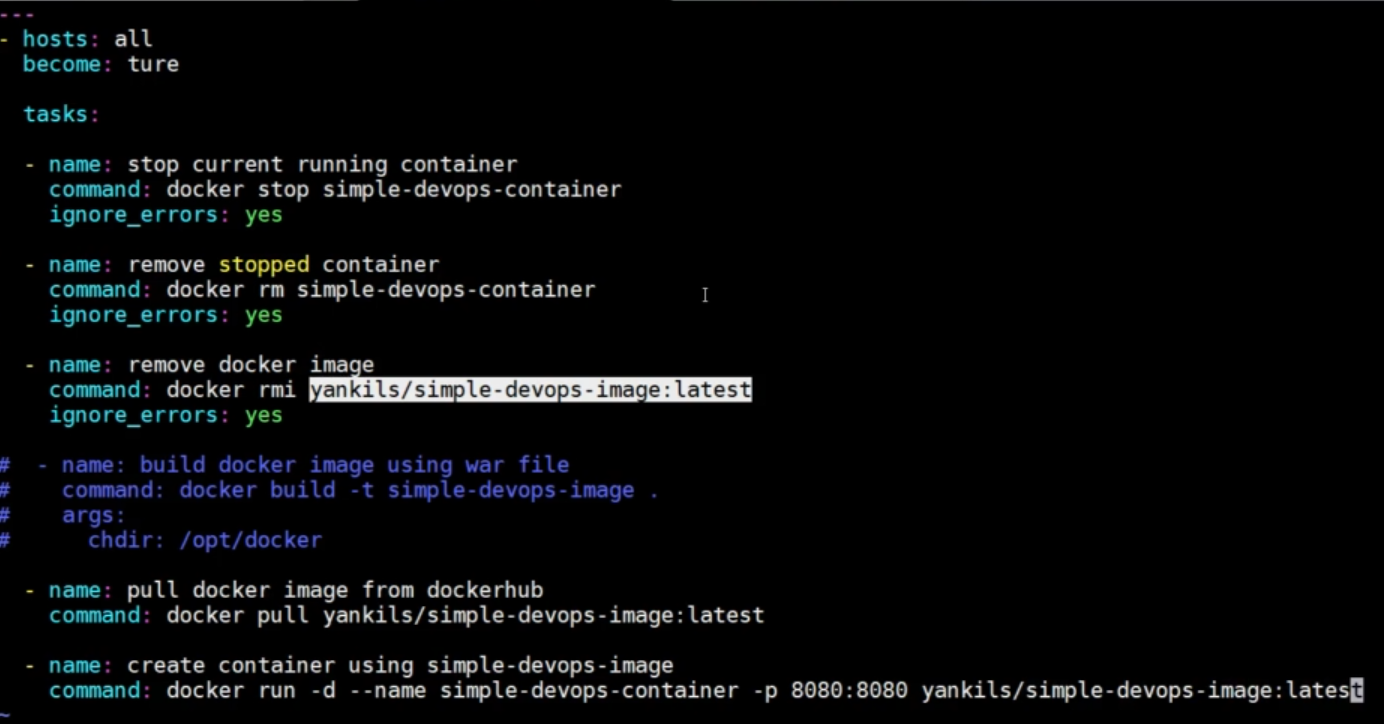
> service sshd reload

**Now test the ssh config in Jenkins 🡪** will pass

1. **Vid17: Creating Jenkins Job : Deploy\_on\_Docker:**
   1. **Copy from previous job 🡪**  docker\_on\_tomcat\_server
      1. Remove Polling
      2. Remove Deploy war to Container post step
      3. **Add Post-build step : “Send build artifacts over SSH”**
         1. **SSH Server :**  docker-host <pre-populated>
         2. **Transfer:**
            1. **Source file** : /webapp/target/\*.war
            2. **Remove prefix:**  /webapp/target
            3. **Remote directory:**  .
      4. **Now run the build**  🡪 You’ll see **webapp.war** is copied on **/home/dockeradmin** folder
2. **Vid18: Creating a Dockerfile :**
   1. Creating docker file at user “dockeradmin” home dire ie. “/home/dockeradmin”
   2. **>** vi **Dockerfile**
      1. FROM tomcat:latest
      2. MAINTAINER KSyed
      3. COPY ./webapp.war **/usr/local/tomcat**
      4. <Save>
   3. > docker build -t devops-image .
   4. > docker images 🡪 you’ll see “devops-image” present
   5. > docker run -d --name devops-container -p 8080:8080 devops-image
   6. Now check if the tomcat is running using browser 🡪 <http://54.174.247.119:8080/webapp/> 🡪 it will show you app
   7. Now delete it as we’ll be doing all these steps from Jenkins
   8. > docker rm <container-name/id>
   9. >docker rmi <image-name>
   10. > docker rmi $(docker images -a -q) 🡪 deleting/removing all the docker images
3. **Vid 19 :** 
   1. **Create another job “Deploy\_on\_Docker\_Container” out of last job “Deploy\_on\_Docker”**
   2. **Add Post-build step : “Send build artifacts over SSH”**
      1. **Exec command :** cd /home/dockeradmin; docker build -t devops-image . ;docker run -d --name docker-container -p 8080:8080 devops-image
      2. <Save the job>
   3. **Now Build the job:**  After build success go to 🡪<http://54.174.247.119:8080/webapp/> 🡪 you’ll see the app
   4. **Note :** If you try to build second times then it will fail as continer is already working. For that we need to use ansible environment
4. **Vid19: Ansible Setup:**
   1. **Create EC2 as previously:**
      1. **>** hostname ansible-control-node
   2. **Installations and User Creations:**
      1. **Install Python and Python-pip:**
         1. **>** yum install python -y
         2. **>**  yum install python-pip -y
      2. **Install ansible**
         1. **>**pip install ansible
      3. **Create Ansible User:**
         1. **>** useradd ansadmin
         2. **>** passwd ansadmin 🡪Set the new password
      4. **Grant Sudo access “ansadmin”**
         1. Editing visudo
            1. >visudo 🡪 ansadmin ALL=(ALL) NOPASSWD=ALL
         2. Or using following command- Adding “ansadmin” to “sudoers” file
            1. >echo “ansadmin ALL=(ALL) NOPASSWD: ALL” >> /etc/sudoers
   3. **Install Docker:**
      1. **>** yum install docker -y
      2. **>**  service docker start
      3. **>** service docker status
      4. **Adding “ansadmin” to “docker” group**
         1. **>** usermod -aG docker ansadmin
         2. **>** id ansadmin 🡪 to check the user group
   4. **SSHD\_Config Password Permission:**
      1. **Need to edit sshd\_config file at /etc/ssh/sshd\_config:**
         1. **>** vi /etc/ssh/sshd\_config
            1. Edit “passwordAuthentication yes” 🡪 save file
         2. > service sshd reload
            1. It will pass the test in Jenkins “Configure System”
   5. **Create “ansadmin” on “docker-host” EC2**
      1. Same steps as we did for ansible-remote-node with same password
   6. **Generate and Copy SSH Key pair:**
      1. **Generate Key pair as “ansadmin”:**
         1. **>** su – ansadmin 🡪 switch to ansadmin account
         2. > ssh-keygen 🡪 to generate key
            1. 2 keys(private & public) will be created at /home/ansadmin/.ssh/id\_rsa & id\_rsa.pub
      2. **Copy Key to Target(docker-host) system:**
         1. **>**  ssh-copy-id ansadmin@<docker-hostIP>
         2. **Question ? yes**
         3. Provide password of ansadmin 🡪 you’ll be loggedin to docker-host
      3. **Copy Key to Localhost:**
         1. **>**  ssh-copy-id ansadmin@localhost
   7. **Ping Test:**
      1. **Create hosts file at /etc/ansible**
         1. /etc/ansible>sudo vi hosts
            1. <docker-hostIP>
            2. Localhost 🡪 save file
      2. **>** ansible all -m ping 🡪 it will ping to both “docker-host” and “localhost”
5. **Vid21: Jenkins Integration:**
   1. **Create a directory in /opt/docker**
      1. **>** sudo mkdir docker
      2. **Make “ansadmin” an owner**
         1. **>** sudo chown -R ansadmin:ansadmin **/opt/docker**
   2. **Create Docker file as we did in docker-file**
      1. Create Docker file at /opt/docker by copy/pasting same contents.
   3. **Add another SSH Server in Jenkkins Configure System:**
      1. **Name :**  ansible-server
      2. **Hostname:** <ipadd>
      3. **Username:** ansadmin
      4. **Password :** <password> 🡪 test the configuration
   4. **Create Job “Deploy\_on\_Ansible”**
      1. **In SSH Publisher post step:**
         1. Select ansible-host
         2. All are same apart from “Remote directory”
            1. //opt//docker
   5. **Run Job and Check the site:**
6. **Vid22/24/25/26/27/28:**
   1. **Creating hosts file** under /opt/docker 🡪 same hosts file as we did for /etc/ansible
   2. **Write Ansible palaybood to Create Docker Image**
      1. Create docker image
      2. Tag docker image with docker hub user prefix
      3. Push docker hub image
      4. Remove both images

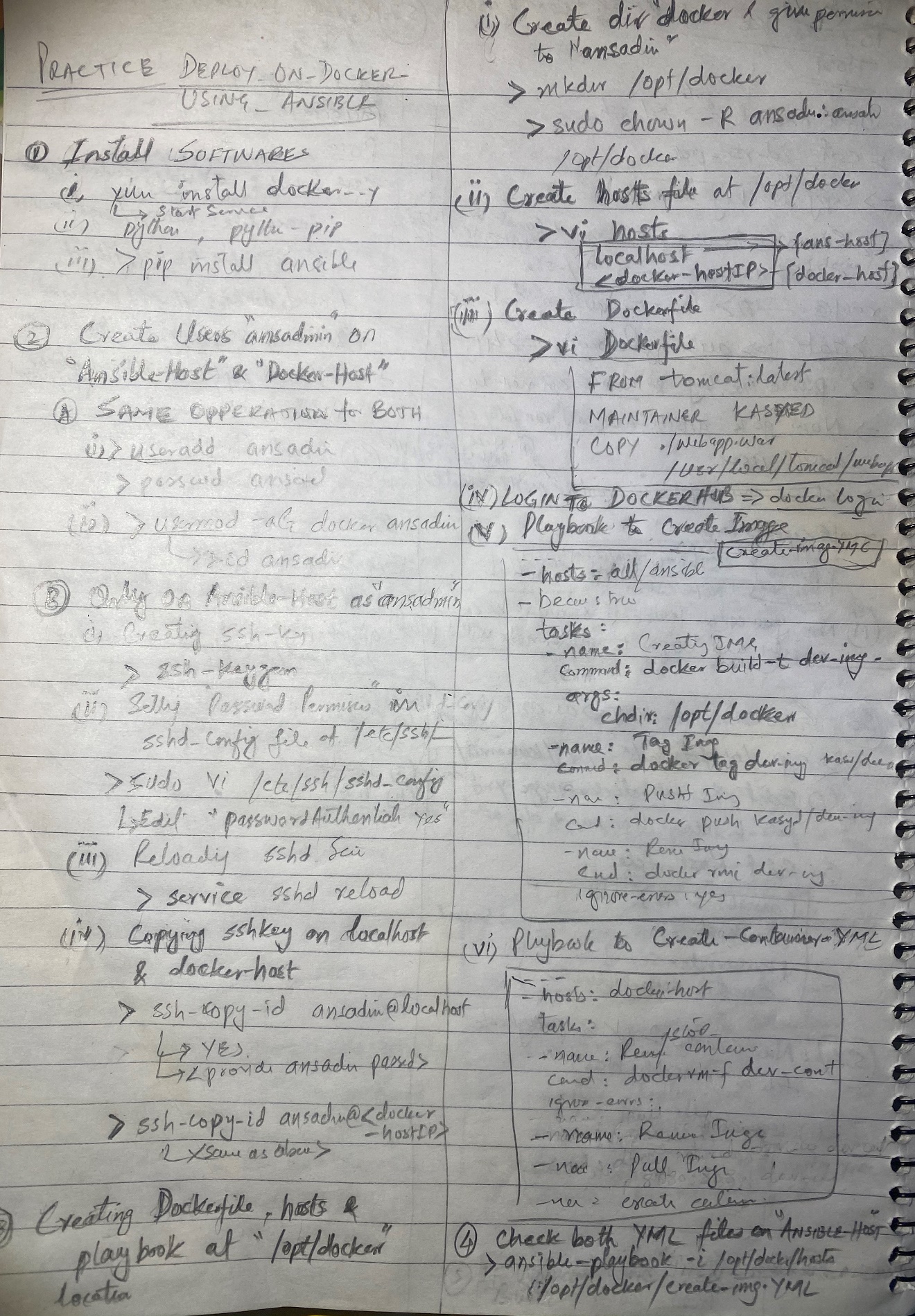


* + 1. **Note:** You need to remove “become: true” otherwise this will throw error.
  1. **Write Ansible playbook to Create Docker Container:**
     1. Stop docker container
     2. Remove docker container
     3. Remove docker hub image
     4. Pull docker-hub image
     5. Create/Run docker container from docker-hub image



* 1. **Check the Docker Image file by running it with following command:**
     1. **>**ansible-playbook -i /opt/docker/hosts “<docker-image.yml> --limit localhost
     2. **Check the image is pushed on docker-hub**
  2. **Check the Docker Container file by running it with following command:**
     1. **>**ansible-playbook -i /opt/docker/hosts “<docker-container.yml> --limit <docker-hostIP>
     2. **Check the container is up and running in docker-host**
     3. **Also check the webapp using** browser http://<dockerhostIP>:8080/webapp

1. **Simple Steps To Deploy WAR File on Docker Container using Ansible Playbook:**
   1. **Following are the Straight forward steps:**

****

1. **Vid29: Jenkins job update:**
   1. **Create another job “Deploy\_on\_AnsiblePlaybook” out of Vid21 Job ie. “Deploy\_on\_Ansible”**
   2. **SSH-Publisher:**
      1. **Transfer:**
         1. **Source file :** webapp/target/\*.war
         2. **Remove prefix :**  webapp/target
         3. **Remote directory:**  //opt//docker//
   3. **Just add Exec Command in SSH Publisher**
      1. **For Ansible-host/node:**
         1. ansible-playbook -i /opt/docker/hosts /opt/docker/create-docker-image.yml --limit localhost;
      2. **For docker-host:**
         1. ansible-playbook -i /opt/docker/hosts /opt/docker/create-docker-container.yml --limit 172.31.90.44;
   4. **Run the job and check following**
      1. **New Docker-hub image**
      2. **New container running on docker-host**
      3. **Also check the webapp using** browser http://<dockerhostIP>:8080/webapp