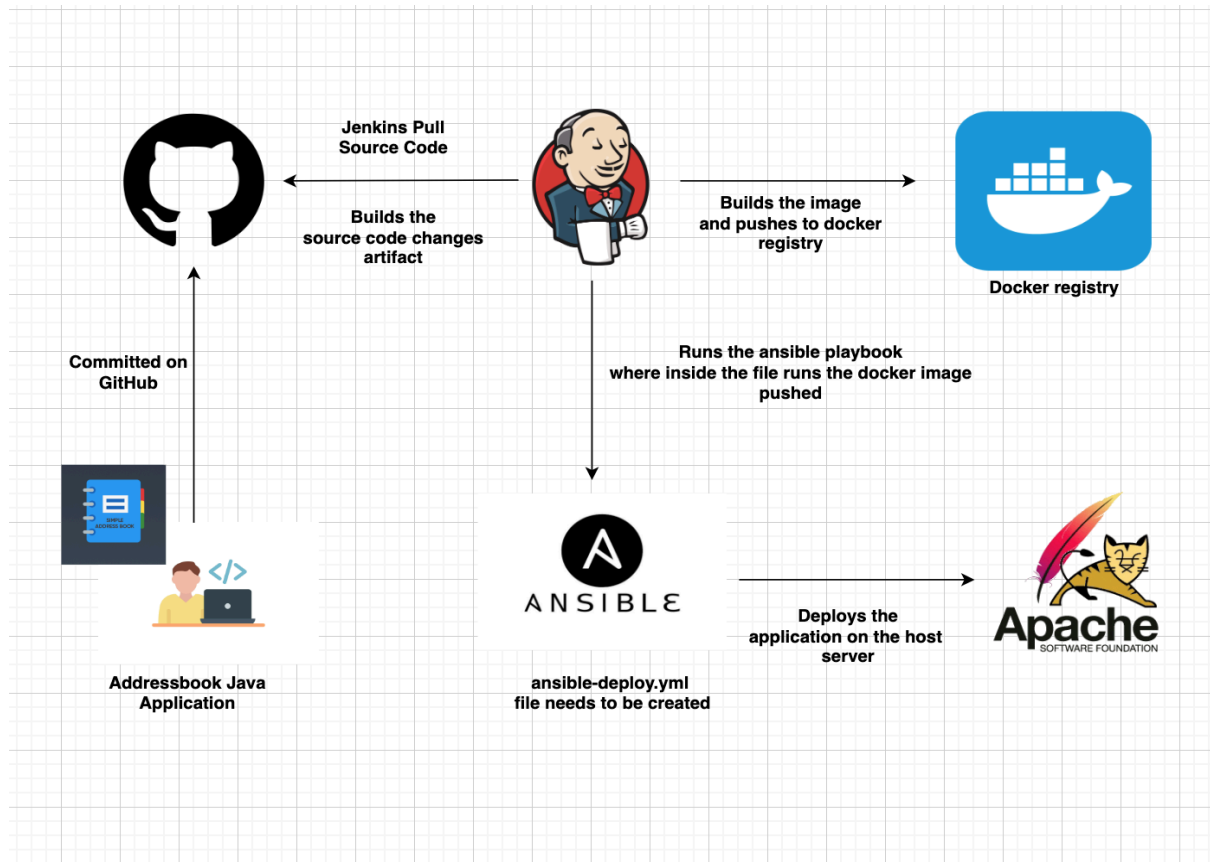
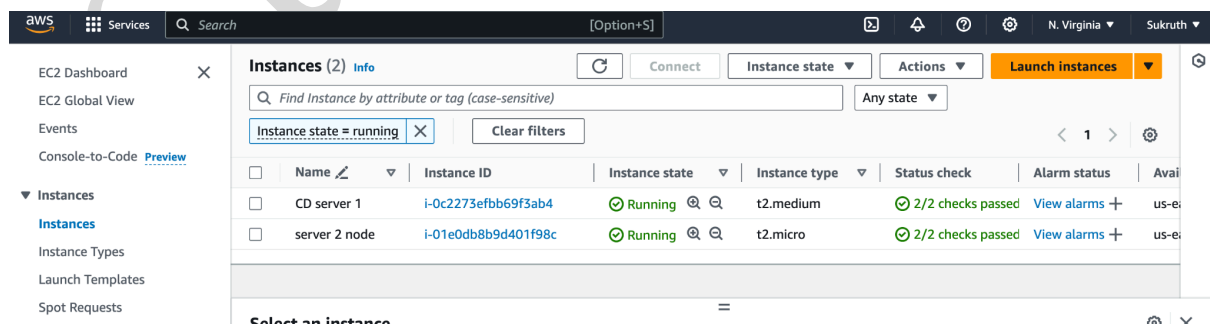


Continuous Integration and Continuous Deployment Pipeline with Jenkins, Docker, and Ansible! Refer to the Hands-On Automation!



Step 1: Setup the environments

- Install Jenkins, Docker, and Ansible on the same server node on the CD server 1
- By default, Jenkins runs with the user "Jenkins"
- And in the deployment server node make sure to install Apache Tomcat and Docker for successful deployment.



Step 2: Execute the scripts to install or refer to the documentation manually to install required tools on server node 1. Ensure to install Java as a prerequisite for Jenkins.

Documentation links Jenkins: <https://www.jenkins.io/doc/book/installing/linux/>

Documentation links Docker: <https://docs.docker.com/engine/install/ubuntu/> [Needs to be installed on both servers]

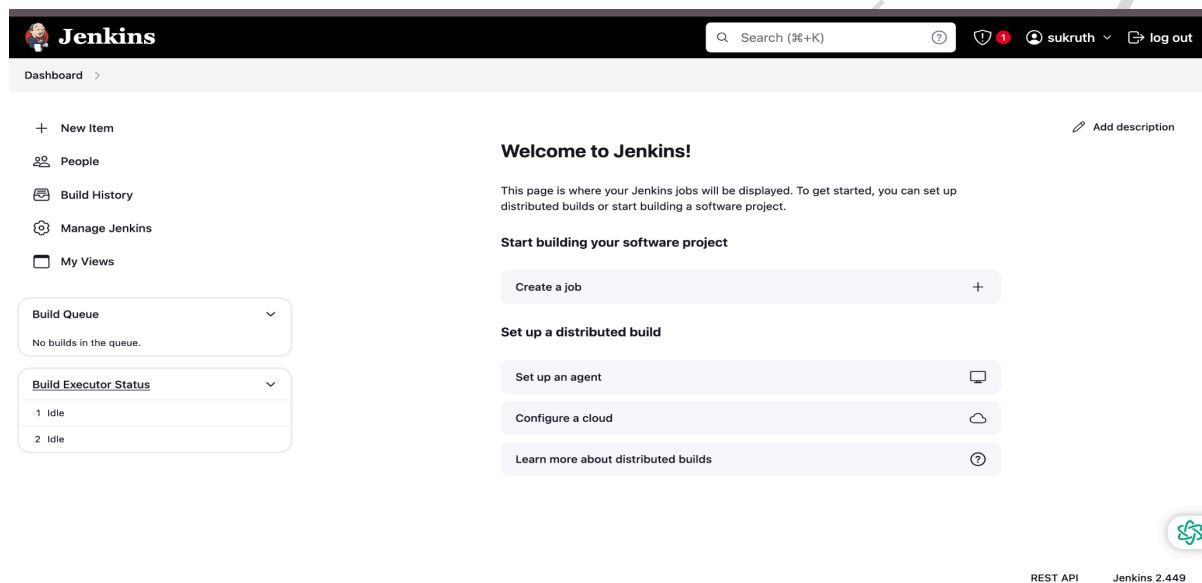
Documentation link Ansible:

https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu

Documentation link for tomcat: <https://tomcat.apache.org/download-10.cgi>

[Needs to be installed on deployment server node]

Successful installation:



Docker

```
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-48-112:~$ docker --version
Docker version 25.0.4, build 1a576c5
ubuntu@ip-172-31-48-112:~$
```

Ansible

```
ansible [core 2.16.4]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
```

Step 3: Create a user named 'Jenkins' and a group named 'Jenkins' in the deployment server node.

```
ubuntu@ip-172-31-23-77:~$ sudo groupadd jenkins
ubuntu@ip-172-31-23-77:~$ sudo useradd jenkins -m -d /home/jenkins -s /bin/bash -g jenkins
ubuntu@ip-172-31-23-77:~$
```

Step 4: Set up SSH keys and generate a key pair for the Jenkins user on server node 1.

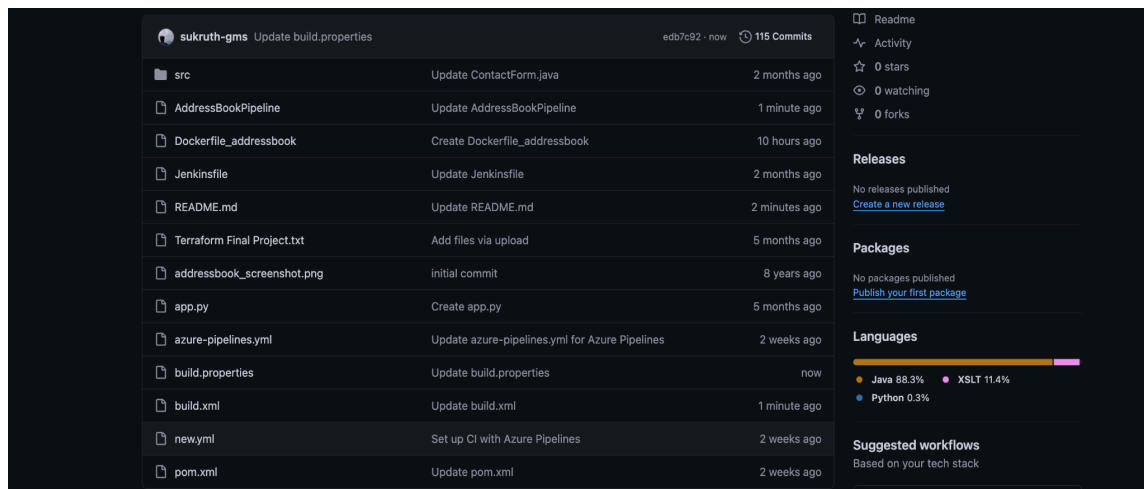
```
ubuntu@ip-172-31-48-112:~$ sudo su - jenkins
jenkins@ip-172-31-48-112:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/jenkins/.ssh/id_rsa):
Created directory '/var/lib/jenkins/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/id_rsa
Your public key has been saved in /var/lib/jenkins/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kyK/WBJUzq09i14ObCmaI7I3WcQjbd3IL2laeotxcKA jenkins@ip-172-31-48-112
The key's randomart image is:
+---[RSA 3072]-----+
|
|      .
|      + .
| oE+. o .
| .*. * o .
| ..=ooo S
|  .*= + +
|  oO= O o
| o BO.B =
| o+++o.o .
+---[SHA256]-----+
jenkins@ip-172-31-48-112:~$
```

i-0c2273efbb69f3ab4 (CD server 1)

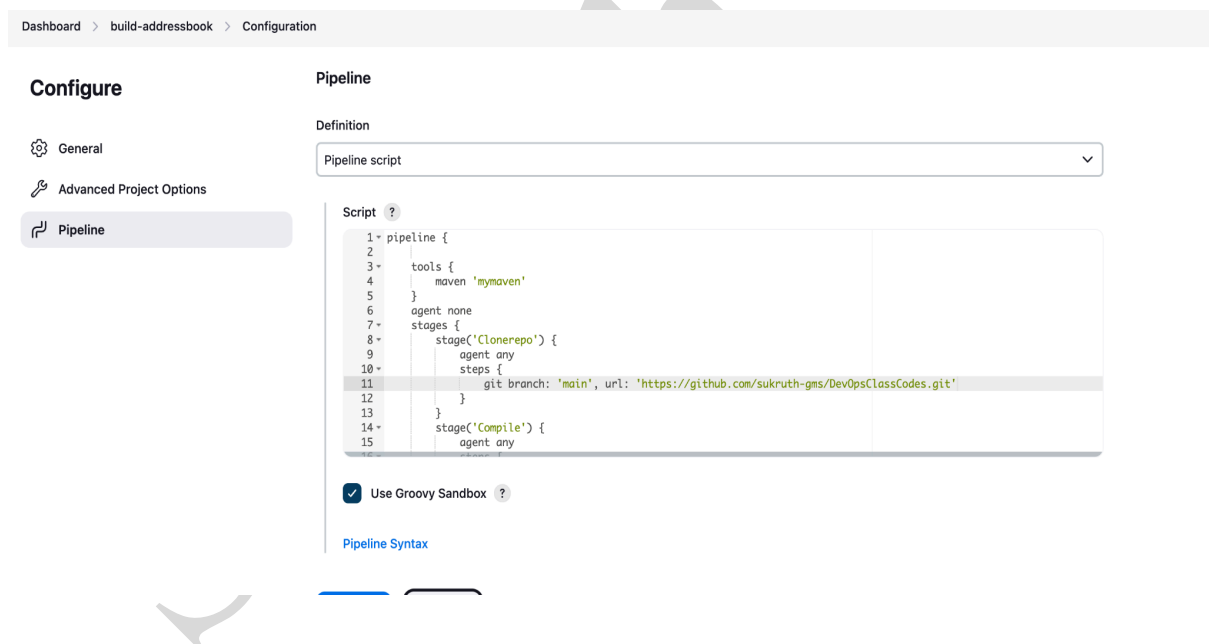
Copy the public key of server node 1 to the deployment server for the Jenkins user.

```
ubuntu@ip-172-31-23-77:~$ sudo passwd jenkins
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-23-77:~$ sudo su jenkins
jenkins@ip-172-31-23-77:/home/ubuntu$ cd ~/.ssh
jenkins@ip-172-31-23-77:~/.ssh$ ls
id_rsa  id_rsa.pub
jenkins@ip-172-31-23-77:~/.ssh$ vi authorized_keys
jenkins@ip-172-31-23-77:~/.ssh$ cd ..
jenkins@ip-172-31-23-77:~$ chmod 700 ~/.ssh
jenkins@ip-172-31-23-77:~$
```

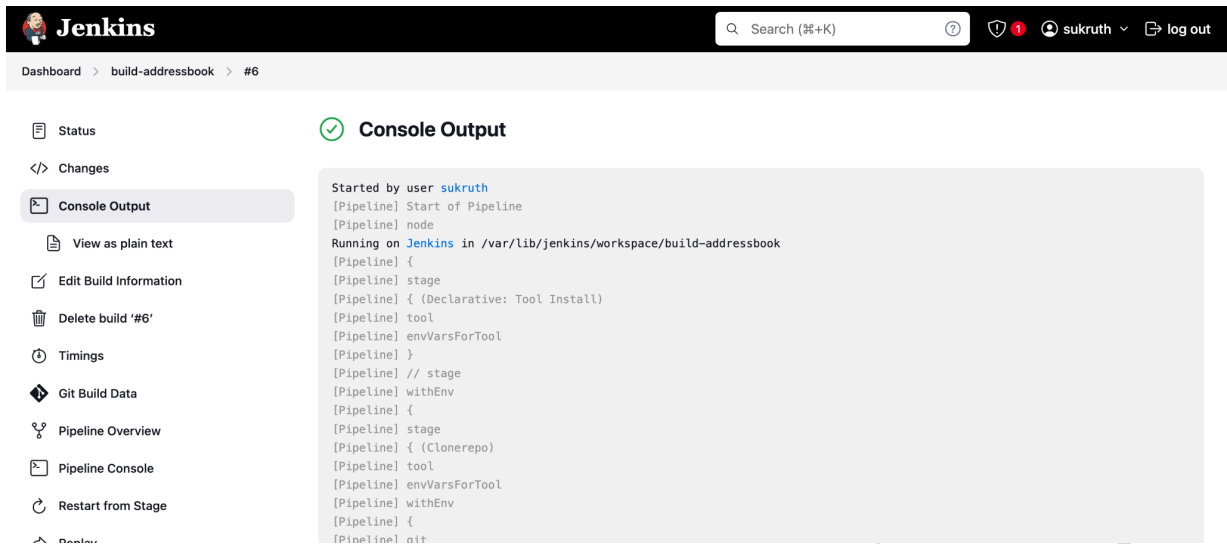
Step 5: Navigate to your GitHub repository's source code and review its contents to ensure it includes the necessary Dockerfile, Jenkinsfile, and pom.xml for the build and deployment process



Step 6: Access Jenkins work items and create a declarative pipeline tasked with performing build, test, and compile operations on the source code using Maven.



Verify the output console of the Jenkins pipeline for a successful build. Additionally, ensure proper indentation in the Jenkins pipeline configuration.



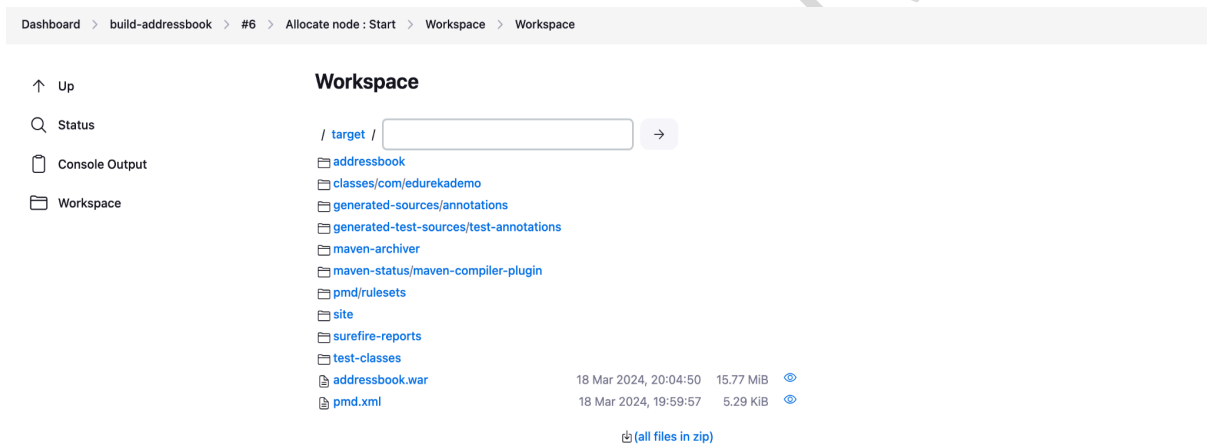
The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information (sukruth). The breadcrumb trail is "Dashboard > build-addressbook > #6". On the left sidebar, the "Console Output" option is selected. The main area displays the console output for build #6, which is a Jenkins Pipeline. The output shows the pipeline starting, running on Jenkins in the workspace directory, and executing various stages and tools. The output is as follows:

```

Started by user sukruth
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/build-addressbook
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clonerepo)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git

```

Navigate to the workspace directory in the output console of the pipeline and inspect the target folder to access the build artifacts.




The screenshot shows the Jenkins workspace view for build #6. The breadcrumb trail is "Dashboard > build-addressbook > #6 > Allocate node : Start > Workspace > Workspace". The left sidebar shows the "Workspace" option selected. The main area displays the workspace contents, including a list of files and folders. The files and folders are as follows:

| File/Folder | Modified | Size | Icon |
|---|-----------------------|-----------|--------|
| addressbook | | | Folder |
| classes/com/edurekademo | | | Folder |
| generated-sources/annotations | | | Folder |
| generated-test-sources/test-annotations | | | Folder |
| maven-archiver | | | Folder |
| maven-status/maven-compiler-plugin | | | Folder |
| pmd/rulesets | | | Folder |
| site | | | Folder |
| surefire-reports | | | Folder |
| test-classes | | | Folder |
| addressbook.war | 18 Mar 2024, 20:04:50 | 15.77 MiB | File |
| pmd.xml | 18 Mar 2024, 19:59:57 | 5.29 KiB | File |


Below the table, there is a link to download all files in a zip format: [\(all files in zip\)](#).

Ensure to identify the paths of the built WAR file and the Dockerfile for configuring the next Jenkins job.


Step 6: Create a new freestyle Jenkins Job for deployment. Execute a shell script within the job to build the Docker image using the build artifact and push it to the Docker registry.


 **Jenkins**


Dashboard > Addressbook_deployment >


 Status


Addressbook_deployment


 Changes

 Workspace

 Build Now

 Configure

 Delete Project

 Rename

Permalinks

Step 7: Before configuring the job, navigate to your Ansible project directory, locate the host file, and add the IP address of the deployment server as a host.

```
ubuntu@ip-172-31-48-112:~$ cd /etc/ansible/
ubuntu@ip-172-31-48-112:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-48-112:/etc/ansible$
```

```
# Ex 3: A collection of database servers in the 'dbse

### [dbservers]
###
### db01.intranet.mydomain.net
### db02.intranet.mydomain.net
### 10.25.1.56
### 10.25.1.57

# Ex4: Multiple hosts arranged into groups such as 'De

### [Debian]
### alpha.example.org
### beta.example.org

### [openSUSE]
### green.example.com
### blue.example.com

34.227.59.107
```

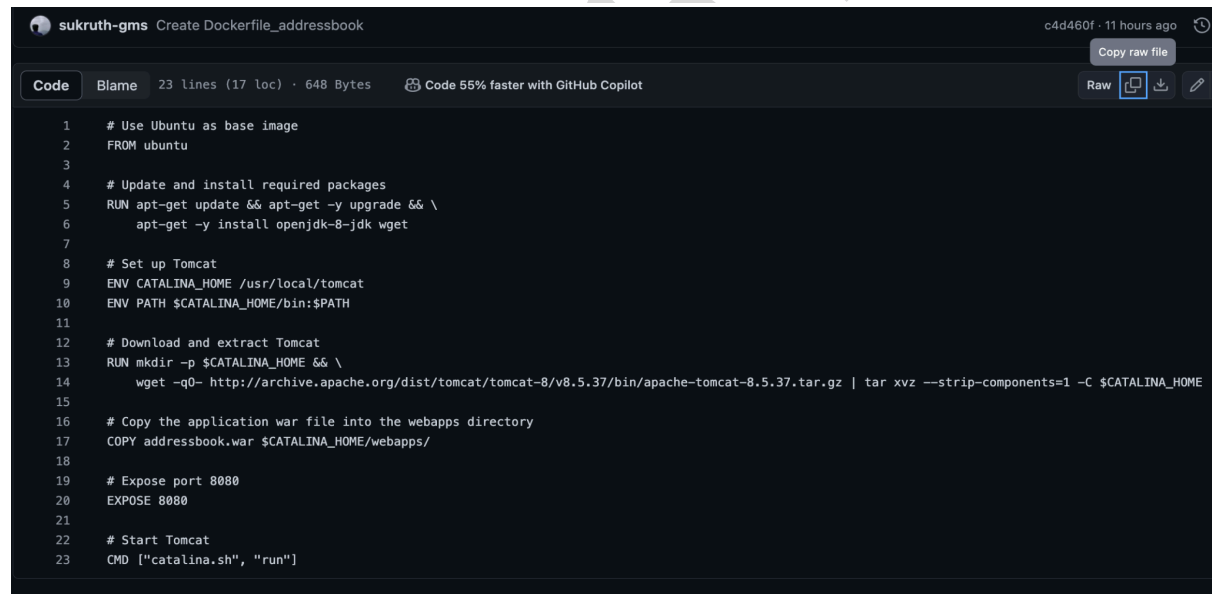
Subsequently, create an **Ansible playbook** with either a .yml or .yaml extension. Within the playbook, define the task configurations that need to be executed.

Addressbook-deploy. yaml [Path to be in addressbook_deployment workspace] for running

```
ubuntu@ip-172-31-48-112:/etc/ansible$ cat addressbook-deploy.yaml
---
- hosts: all
  become: yes
  tasks:
    - name: Deploy Address Book Docker Container
      command: docker run -d -P sukruth17/addressbook:v2
      register: out
    - debug: var=out
```

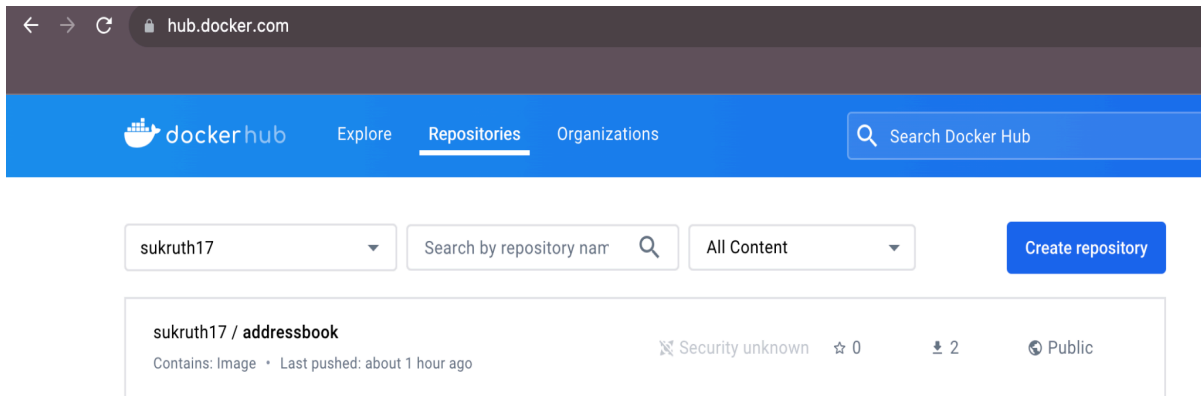
Ensure that your **Dockerfile** is properly configured. Verify that the Dockerfile is set up to create a Docker image capable of running a Java web application using Apache Tomcat.

Dockerfile: [Path to be in addressbook_deployment workspace] for running



```
sukruth-gms Create Dockerfile_addressbook c4d460f · 11 hours ago
Copy raw file
Code Blame 23 Lines (17 loc) · 648 Bytes Code 55% faster with GitHub Copilot Raw
1 # Use Ubuntu as base image
2 FROM ubuntu
3
4 # Update and install required packages
5 RUN apt-get update && apt-get -y upgrade && \
6     apt-get -y install openjdk-8-jdk wget
7
8 # Set up Tomcat
9 ENV CATALINA_HOME /usr/local/tomcat
10 ENV PATH $CATALINA_HOME/bin:$PATH
11
12 # Download and extract Tomcat
13 RUN mkdir -p $CATALINA_HOME && \
14     wget -qO- http://archive.apache.org/dist/tomcat/tomcat-8/v8.5.37/bin/apache-tomcat-8.5.37.tar.gz | tar xvz --strip-components=1 -C $CATALINA_HOME
15
16 # Copy the application war file into the webapps directory
17 COPY addressbook.war $CATALINA_HOME/webapps/
18
19 # Expose port 8080
20 EXPOSE 8080
21
22 # Start Tomcat
23 CMD ["catalina.sh", "run"]
```

Step 8: Navigate to Docker Registry (hub.docker.com) and create a repository with the application name.



Step 9: Configure your *Jenkins Job*: Build Step for Addressbook_Deployment

```
1 #!/bin/bash
2
3 # Build Tomcat Docker image with application war file
4 sudo docker build -t addressbookapp:v2 /var/lib/jenkins/workspace/Addressbook_deployment
5
6 # Check if Docker build was successful
7 if [ $? -ne 0 ]; then
8     echo "Error: Docker build failed"
9     exit 1
10 fi
11
12 # Login to Docker Hub
13 sudo docker login -u sukru17 -p sukru17
14
15 # Check if Docker login was successful
16 if [ $? -ne 0 ]; then
17     echo "Error: Docker login failed"
18     exit 1
19 fi
20
21 # Tag the Docker image
22 sudo docker tag addressbookapp:v2 sukru17/addressbook:v2
23
24 # Push the Docker image to Docker Hub
25 sudo docker push sukru17/addressbook:v2
26
27 # Check if Docker push was successful
28 if [ $? -ne 0 ]; then
29     echo "Error: Docker push failed"
30     exit 1
31 fi
32
33 # Change directory to Ansible playbook location
34 cd /etc/ansible
35
36 # Run Ansible playbook for deployment
37 ansible-playbook -i hosts addressbook-deploy.yaml
38
39 # Check if Ansible playbook execution was successful
40 if [ $? -ne 0 ]; then
41     echo "Error: Ansible playbook execution failed"
42     exit 1
43 fi
44
45 # Exit with success status
46 exit 0
```


Monitor the **Jenkins job console** for output and confirm the **successful deployment**.

```
Dashboard > Addressbook_deployment > #17 > Console Output

"d3d97175c3be: Pulling fs layer",
"d3d97175c3be: Waiting",
"8ea325196cf2: Verifying Checksum",
"8ea325196cf2: Download complete",
"23828d760c7b: Verifying Checksum",
"23828d760c7b: Download complete",
"d3d97175c3be: Verifying Checksum",
"d3d97175c3be: Download complete",
"82521fbf8a19: Verifying Checksum",
"82521fbf8a19: Download complete",
"23828d760c7b: Pull complete",
"82521fbf8a19: Pull complete",
"8ea325196cf2: Pull complete",
"d3d97175c3be: Pull complete",
"Digest: sha256:25389a271f7ffef726ac14011f7bf71debbbaeb1bd994562211eab7bc5e1c1",
"Status: Downloaded newer image for sukruth17/addressbook:v2"
],
"stdout": "3f7aacc4840c2743c60e903769f99c877c0644fd23f4b473e75b6a33985c3525",
"stdout_lines": [
  "3f7aacc4840c2743c60e903769f99c877c0644fd23f4b473e75b6a33985c3525"
]
}
}
}

PLAY RECAP *****
34.227.59.107      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

Finished: SUCCESS
```

On the **deployment server**, verify the containers that are currently up and **running**.

```
ubuntu@ip-172-31-23-77:~$ sudo su jenkins
jenkins@ip-172-31-23-77:/home/ubuntu$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
3f7aacc4840c   sukruth17/addressbook:v2           "catalina.sh run"       About an hour ago    Up 33 minutes   0.0.0.0:32768->8080/tcp, :::32768->8080/tcp
ondescending_noyce
```

Both Jenkins work items have been **executed successfully**.

Search (⌘+K)

sukruth log out

Dashboard

+ New Item

People

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All

+

| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|------------------------|----------------|---------------|---------------|
| | | Addressbook_deployment | 38 min #17 | 44 min #16 | 24 sec |
| | | build-addressbook | 1 hr 49 min #7 | 2 hr 1 min #5 | 22 sec |

Icon: S M L

Icon legend

Atom feed for all

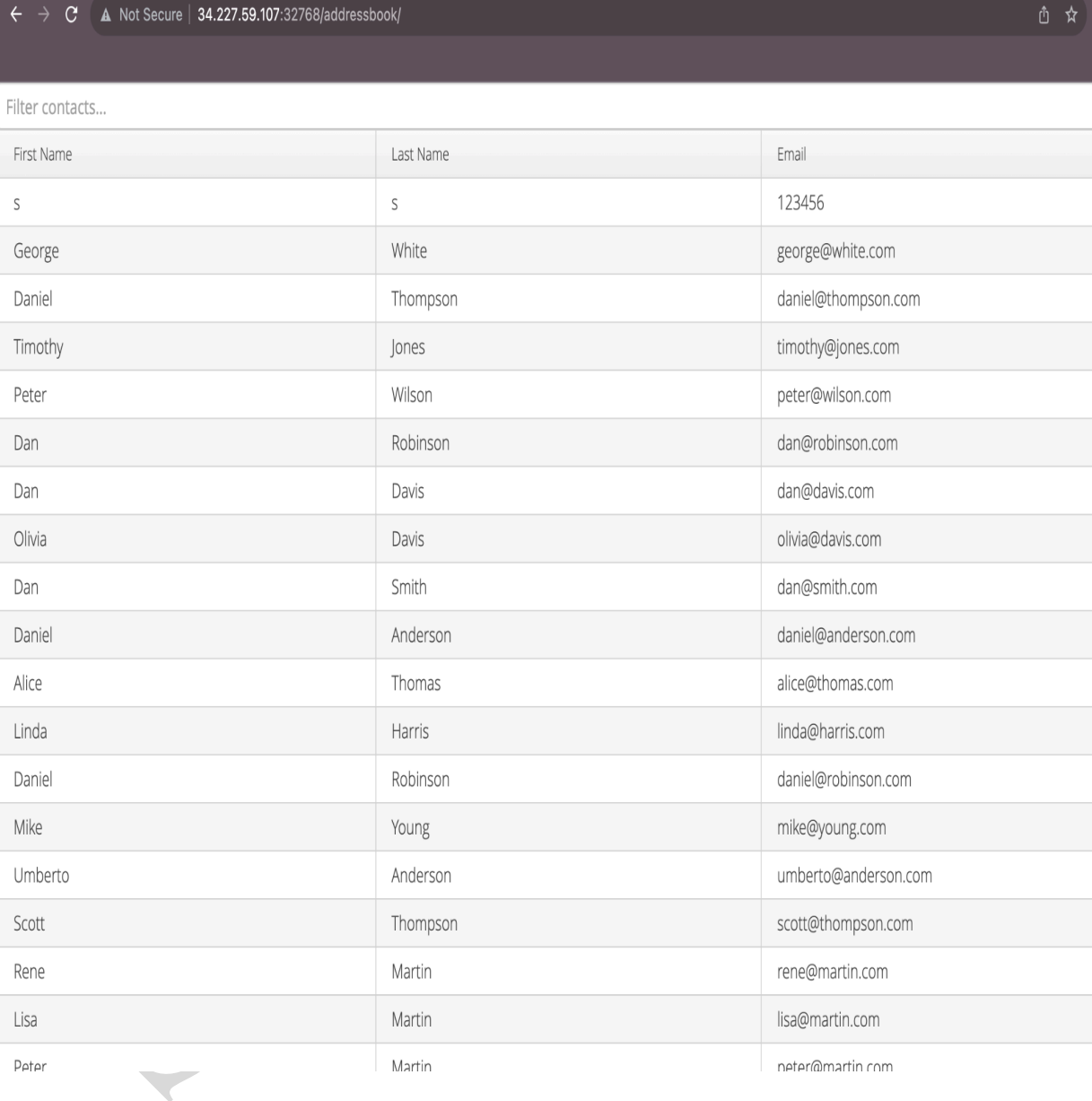
Atom feed for failures

Atom feed for just latest builds

Add description

Application

Step 10: Navigate to the port opening indicated in the container details and access the host URL for the deployed application.



| First Name | Last Name | Email |
|------------|-----------|----------------------|
| s | s | 123456 |
| George | White | george@white.com |
| Daniel | Thompson | daniel@thompson.com |
| Timothy | Jones | timothy@jones.com |
| Peter | Wilson | peter@wilson.com |
| Dan | Robinson | dan@robinson.com |
| Dan | Davis | dan@davis.com |
| Olivia | Davis | olivia@davis.com |
| Dan | Smith | dan@smith.com |
| Daniel | Anderson | daniel@anderson.com |
| Alice | Thomas | alice@thomas.com |
| Linda | Harris | linda@harris.com |
| Daniel | Robinson | daniel@robinson.com |
| Mike | Young | mike@young.com |
| Umberto | Anderson | umberto@anderson.com |
| Scott | Thompson | scott@thompson.com |
| Rene | Martin | rene@martin.com |
| Lisa | Martin | lisa@martin.com |
| Peter | Martin | peter@martin.com |

Addressbook application