Devops Day 2

Docker Installation & Components

Docker Image:

A packaged application with its dependencies.

Commands: docker images, docker pull <image_name>, docker rmi <image_id>

Docker Container:

A running instance of a Docker Image.

Commands: docker run -it <image_name>, docker ps, docker stop <container id>, docker rm <container id>

Dockerfile:

A script to build a Docker Image.

Example: A Node.js app using FROM node:14, COPY, RUN npm install, and CMD ["node", "server.js"]

Docker Volumes:

Used for persistent storage.

Commands: docker volume create <volume_name>, docker volume ls, docker run -v <volume>:/path -it ubuntu

Example Project: Python "Hello, World!" in Docker

Install Docker and Docker Compose.

Create a Flask application (app.py).

Write a Dockerfile for packaging the app.

Define services in docker-compose.yml.

Build and run using docker-compose build and docker-compose up -d.

Test at http://localhost:5000.

Jenkins-Github-Docker-App-Demo

GitHub Setup

Generate a Personal Access Token (PAT) for GitHub access.

Upload the Jenkinsfile to the GitHub repository.

Use git fetch, git pull, git status, git add, git commit, and git push to manage repository updates.

If an error occurs during git push, use:

git push :spAT>@github.com/suser-name>/srepo-name>.git">https://susername>:spAT>@github.com/suser-name>/srepo-name>.git

Jenkins Configuration

groups Jenkins sudo usermod -aG docker Jenkins sudo systemctl restart Jenkins

DockerHub Integration

- Ensure correct Docker Hub username and credentials ID are configured in Jenkins.
- Example placeholders:
 - 1. docker-hub-credentials: Jenkins credential ID for Docker Hub.
 - 2. github-padma: Jenkins credential ID for GitHub.
 - 3. PadmavathyNarayanan: GitHub username.
 - 4. jenkins-docker-demo: GitHub repository name

Final Deployment

- 1. After a successful build, the Docker image is pushed to DockerHub.
- 2. The image can be accessed at

https://hub.docker.com/repository/docker/<user-name>

Screenshots

```
Your name and email address were configured automatically based on your username and hostname. Please check that they are accurate. You can suppress this message by setting them explicitly. Run the following command and follow the instructions in your editor to edit your configuration file:
                       git config --global --edit
  After doing this, you may fix the identity used for this commit with:
                       git commit --amend --reset-author
    4 files changed, 34 insertions(+)
create mode 100644 Dockerfile
create mode 100644 app.py
create mode 100644 docker-compose.yml
create mode 100644 requirements.txt
create mode 100644 requirements.txt

vboxuser@Ubuntu:~/docker-python-app/Devops_$ git push

Username for 'https://github.com': sivakulanthaisamy

Password for 'https://sivakulanthaisamy@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repor information on currently recommended modes of authentication.

fatal: Authentication failed for 'https://github.com/sivakulanthaisamy/Devops_/'

vboxuser@Ubuntu:~/docker-python-app/Devops_$ git push

Username for 'https://github.com': git push https://sivakulanthaisamy:https://drive.google.com/dr
^C
                                                                                                                                    -python-app/Devops_$ git push https://sivakulanthaisamy:ghp_vJGPYaPvqS37X
vboxuser@Ubuntu:~/docker-python-app/Devops_$ git push https:,
sivakulanthaisamy/Devops_
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 919 bytes | 919.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sivakulanthaisamy/Devops_
5341f03..c27de0e main -> main
vboxuser@Ubuntu:~/docker-python-app/Devops_$
  vboxuser@Ubuntu:~
                                                                                                                                   -python-app$ sudo docker build -t test .
Sending build context to Docker daemon 5.12kB Step 1/7: FROM python:3.11
3.11: Pulling from library/python 7cd785773db4: Already exists 091eb8249475: Already exists 255774e0027b: Already exists 353e14e5cc47: Already exists 963091970bc2: Already exists e7235c43f7e3: Already exists
organization of the property o
          ---> 18c0f2265fd9
 Step 2/7: WORKDIR /app
---> Running in c91cda36c106
---> Removed intermediate container c91cda36c106
            --> 629d2b6c177a
 Step 3/7 : COPY requirements.txt .
---> 99bd4d5acee6
---> 99bd4d5acee6

Step 4/7: RUN pip install --no-cache-dir -r requirements.txt
---> Running in 10e50cf5de7f

Collecting flask (from -r requirements.txt (line 1))
Downloading flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)

Collecting Werkzeug>=3.1 (from flask->-r requirements.txt (line 1))
Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)

Collecting Jinja2>=3.1.2 (from flask->-r requirements.txt (line 1))
Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)

Collecting itsdangerous>=2.2 (from flask->-r requirements.txt (line 1))
Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
```

```
thon-app$ sudo docker-compose build
WARN[0000] /home/vboxuser/docker-python-app/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, power it to avoid potential confusion

Compose can now delegate builds to bake for better performance.

To do so, set COMPOSE_BAKE=true.

[+] Building 139.65 (11/11) FINISHED

Source of the performance of the perfor
          [app internal] load build definition from Dockerfile

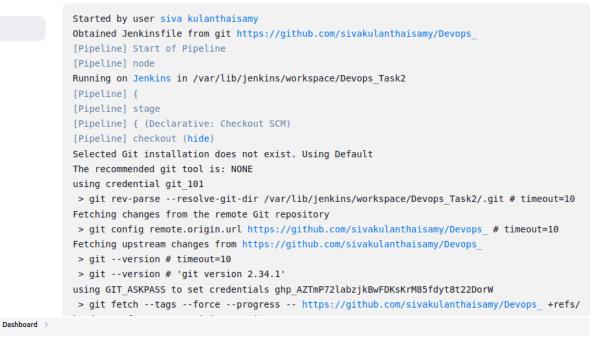
-> transferring dockerfile: 4538
[app internal] load metadata for docker.io/library/python:3.11
[app internal] load .dockerignore
 vboxuser@Ubuntu:~/docker-python-app$ ls
app.py Devops_ docker-compose.yml Dockerfile requirements.txt
vboxuser@Ubuntu:~/docker-python-app$ mv app.py docker-compose.yml Dockerfile requirement.txt Devo
vboxuser@Ubuntu:~/docker-python-app$ mv app.py docker-compose.yml Dockerfile requirements.txt Dev
mv: cannot stat 'app.py': No such file or directory
mv: cannot stat 'docker-compose.yml': No such file or directory
mv: cannot stat 'Dockerfile': No such file or directory
mv: cannot stat 'Dockerfile': No such file or directory
 vboxuser@Ubuntu:~/docker-python-app$ ls
 vboxuser@Ubuntu:~/docker-python-app$ cd Devops
 vboxuser@Ubuntu:~/docker-python-app/Devops_$ ls
 app.py docker-compose.yml Dockerfile README.md requirements.txt vboxuser@Ubuntu:~/docker-python-app/Devops_$ cd ..
 vboxuser@Ubuntu:~/docker-python-app$ cd Devops_
 vboxuser@Ubuntu:~/docker-python-app/Devops_$ ls
app.py docker-compose.yml Dockerfile README.md requirements.txt
vboxuser@Ubuntu:~/docker-python-app/Devops_$ git add --all
vboxuser@Ubuntu:~/docker-python-app/Devops_$ git commit -m "first commit"
 [main c27de0e] first commit
   Committer: vboxuser <vboxuser@Ubuntu.myguest.virtualbox.org>
 Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate. You can suppress this message by setting them explicitly. Run the following command and follow the instructions in your editor to edit
 your configuration file:
              git config --global --edit
 After doing this, you may fix the identity used for this commit with:
              git commit --amend --reset-author
    4 files changed, 34 insertions(+)
    create mode 100644 Dockerfile
    create mode 100644 app.py
    create mode 100644 docker-compose.yml
    create mode 100644 requirements.txt
```

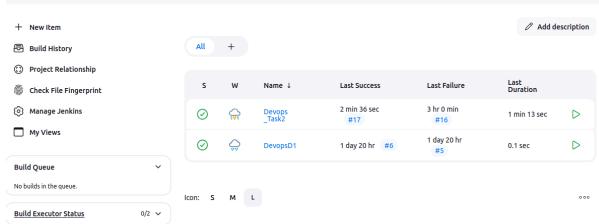
```
vboxuser@Ubuntu:~/docker-python-app$ nano app.py
vboxuser@Ubuntu:~/docker-python-app$ nano requirements.txt
vboxuser@Ubuntu:~/docker-python-app$ nano Dockerfile
vboxuser@Ubuntu:~/docker-python-app$ nano docker-compose.yml
vboxuser@Ubuntu:~/docker-python-app$ sudo docker-compose build
 ---> 81d5a890d1cb
Successfully built 81d5a890d1cb
Successfully tagged test:latest
vboxuser@Ubuntu:~/docker-python-app$ sudo docker run -p 5000:5000 test
 * Serving Flask app 'app'
* Debug mode: off
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
172.17.0.1 - - [19/Mar/2025 05:18:40] "GET / HTTP/1.1" 200 - 172.17.0.1 - - [19/Mar/2025 05:18:41] "GET /favicon.ico HTTP/1.1" 404 - ^Cvboxuser@Ubuntu:~/docker-python-app$ git status fatal: not a git repository (or any of the parent directories): .git
vboxuser@Ubuntu:~/docker-python-app$ ^C
vboxuser@Ubuntu:~/docker-python-app$ git init
hint: Using 'master' as the name for the initial branch. This default branch name hint: is subject to change. To configure the initial branch name to use in all
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
Initialized empty Git repository in /home/vboxuser/docker-python-app/.git/
vboxuser@Ubuntu:~/docker-python-app$ ls
app.py docker-compose.yml Dockerfile requirements.txt
vboxuser@Ubuntu:~/docker-python-app$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
```





View as plain text





```
GNU nano 6.2
pipeline {
agent any
environment {
                                                                                                              Jenkinsfile
             DOCKER_IMAGE = "siva182004/docker-app:latest" // Docker image name
CONTAINER_NAME = "docker-running-app"
REGISTRY_CREDENTIALS = "docker-hub-credentials" // Jenkins credentials ID for Docker Hub
      stages {
    stage('Checkout Code') {
    stage('checkout Code') {
}
                    steps {
    withCredentials([usernamePassword(credentialsId: 'demo', usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN>
    git url: "https://$GIT_USER:$GIT_TOKEN@github.com/Surya-2k4/Jenkins-Docker-Demo.git", branch: 'main'
                          git url:
}
             stage('Build Docker Image') {
    steps {
        script {
            sh 'docker build -t ${DOCKER_IMAGE} .' // Ensure the image is built with the correct tage...
             stage('Docker Login') {
                    steps {
   withcredentials([usernamePassword(credentialsId: 'docker-hub-credentials', usernameVariable: 'DOCKER_USER', passworb's h 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin'
   Activities

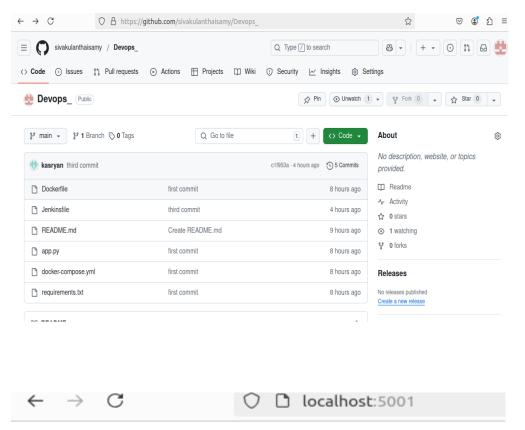
    Terminal

                                                                                                                          Mar 19 19:25
                                                                                                                                                                                                                Q = -
                                                                                               vboxuser@Ubuntu: ~/docker-python-app/Devops_
            vboxuser@Ubuntu:~$ docker --version
Docker version 26.1.3, build 26.1.3-Oubuntu1~22.04.1
vboxuser@Ubuntu:~$ jenkins --version
2.492.2
             Documents Music Public Templates
docker-python-app Downloads Pictures snap Videos
vboxuser@Ubuntu:-$ cd docker-python-app
vboxuser@Ubuntu:-/docker-python-app$ ls
Devops
             VBOXASET@UDUNTEL. PACKET PYTHON BOPY/

[Sudo] password for vboxuser:

vboxuser@Ubuntu:-/docker-python-app/Devops_$ sudo systemctl restart jenkins

vboxuser@Ubuntu:-/docker-python-app/Devops_$
```



Hello, Flask is running in Docker!