**GitHub to Jenkins to Docker Image Pipeline**

**Summary:** Code is first stored in the GitHub repository. Jenkins, on demand, will pull the code from the repository and package the code into a Docker Image. The Docker Image will then be pushed into a private Dockery registry. The Image can be downloaded by those who have access to the registry. This will allow Docker Images to get the newest code in a pre-configured Docker Image when needed.

**Steps for GitHub to Jenkins Pipeline: Docker Image creation**

The GitHub repository code is pulled from

A screenshot of a computer

AI-generated content may be incorrect.

The JenkinsFile, which is the pipeline script in Jenkins, will tell the pipeline where to get the application code from. The JenkinsFile will also build the Docker Image through Docker and push the Image to the specified Docker Repository.

A screenshot of a computer code

AI-generated content may be incorrect.

A DockerFile is needed in the code repository, so Docker knows how to build the Docker Image, after receiving the Dockerfile from Jenkins.

The Dockerfile pulls a base image from the official Docker Registry hub. In this scenario a Linux image is pulled from the Registry. The Dockerfile file updates the Linux instance from Docker, installs the GitHub application, so code can be pulled from the GitHub repository. The DockeFile then copies GitHub files over to the following directory: /usr/var/www/

A screenshot of a computer program

AI-generated content may be incorrect.

Port 80 is set to accept incoming web requests. The last command of the Dockerfile sets the apache web server to be the main process running on this image when it loads.

**Jenkins build output data:**

Started by user [admin](http://18.204.238.203:8080/user/admin)

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Jenkins](http://18.204.238.203:8080/computer/(built-in)/) in /var/lib/jenkins/workspace/ultimate-cd-cd-pipeline

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Tool Install)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Source Control Management)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] withEnv

[Pipeline] {

[Pipeline] git

The recommended git tool is: git

No credentials specified

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/ultimate-cd-cd-pipeline/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url <https://github.com/lthompsonINTL/Reis-website/> # timeout=10

Fetching upstream changes from <https://github.com/lthompsonINTL/Reis-website/>

> git --version # timeout=10

> git --version # 'git version 2.43.0'

> git fetch --tags --force --progress -- <https://github.com/lthompsonINTL/Reis-website/> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git rev-parse refs/remotes/origin/main^{commit} # timeout=10

Checking out Revision facf02daefc05bb59f4407504b5c517cb9e14eb5 (refs/remotes/origin/main)

> git config core.sparsecheckout # timeout=10

> git checkout -f facf02daefc05bb59f4407504b5c517cb9e14eb5 # timeout=10

> git branch -a -v --no-abbrev # timeout=10

> git branch -D main # timeout=10

> git checkout -b main facf02daefc05bb59f4407504b5c517cb9e14eb5 # timeout=10

Commit message: "Update Dockerfile"

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Build Docker Image & Push)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] withEnv

[Pipeline] {

[Pipeline] script

[Pipeline] {

[Pipeline] withDockerRegistry

$ docker login -u lthompsonintl -p \*\*\*\*\*\*\*\* <https://index.docker.io/v1/>

WARNING! Using --password via the CLI is insecure. Use --password-stdin.

WARNING! Your password will be stored unencrypted in /var/lib/jenkins/workspace/ultimate-cd-cd-pipeline@tmp/4c535e38-8c66-4cb6-910c-e17f4b0974fd/config.json.

Configure a credential helper to remove this warning. See

<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

[Pipeline] {

[Pipeline] sh

+ docker build -t lthompsonintl/jenkins-cicd-pipeline:latest .

DEPRECATED: The legacy builder is deprecated and will be removed in a future release.

Install the buildx component to build images with BuildKit:

<https://docs.docker.com/go/buildx/>

Sending build context to Docker daemon 19.72MB

Step 1/7 : FROM httpd:2.4

---> 0de612e99135

Step 2/7 : RUN apt-get update && apt-get upgrade -y

---> Using cache

---> 8113224734ea

Step 3/7 : RUN apt install git -y

---> Using cache

---> d93c1aeea652

Step 4/7 : RUN git clone <https://github.com/lthompsonINTL/Reis-website.git> /usr/var/www/

---> Using cache

---> c760969abd79

Step 5/7 : WORKDIR /usr/var/www/

---> Using cache

---> 872490f0f41e

Step 6/7 : EXPOSE 80

---> Using cache

---> 2357bade9ab6

Step 7/7 : CMD ["httpd-foreground"]

---> Using cache

---> 807d99fa4ebd

Successfully built 807d99fa4ebd

Successfully tagged lthompsonintl/jenkins-cicd-pipeline:latest

[Pipeline] sh

+ docker push lthompsonintl/jenkins-cicd-pipeline:latest

The push refers to repository [docker.io/lthompsonintl/jenkins-cicd-pipeline]

9dce97d39ff9: Preparing

8a1ea9ddef9a: Preparing

54e1d73088d3: Preparing

ce8b199d6616: Preparing

f45ccb4c91cd: Preparing

bfbb785ffbb6: Preparing

5f70bf18a086: Preparing

ae4d5d766e6e: Preparing

5f1ee22ffb5e: Preparing

bfbb785ffbb6: Waiting

5f70bf18a086: Waiting

5f1ee22ffb5e: Waiting

54e1d73088d3: Layer already exists

9dce97d39ff9: Layer already exists

ce8b199d6616: Layer already exists

f45ccb4c91cd: Layer already exists

8a1ea9ddef9a: Layer already exists

5f70bf18a086: Layer already exists

ae4d5d766e6e: Layer already exists

bfbb785ffbb6: Layer already exists

5f1ee22ffb5e: Layer already exists

latest: digest: sha256:888a8fd34ad2d27d333c3f287049325a65b140403b0d4096fbe6a728daa00515 size: 2208

[Pipeline] }

[Pipeline] // withDockerRegistry

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS

**Docker Repository**

**A screenshot of a computer

AI-generated content may be incorrect.**