Java Spring Boot CI/CD Pipeline with Docker

Tech Stack used:

Java (Spring Boot)

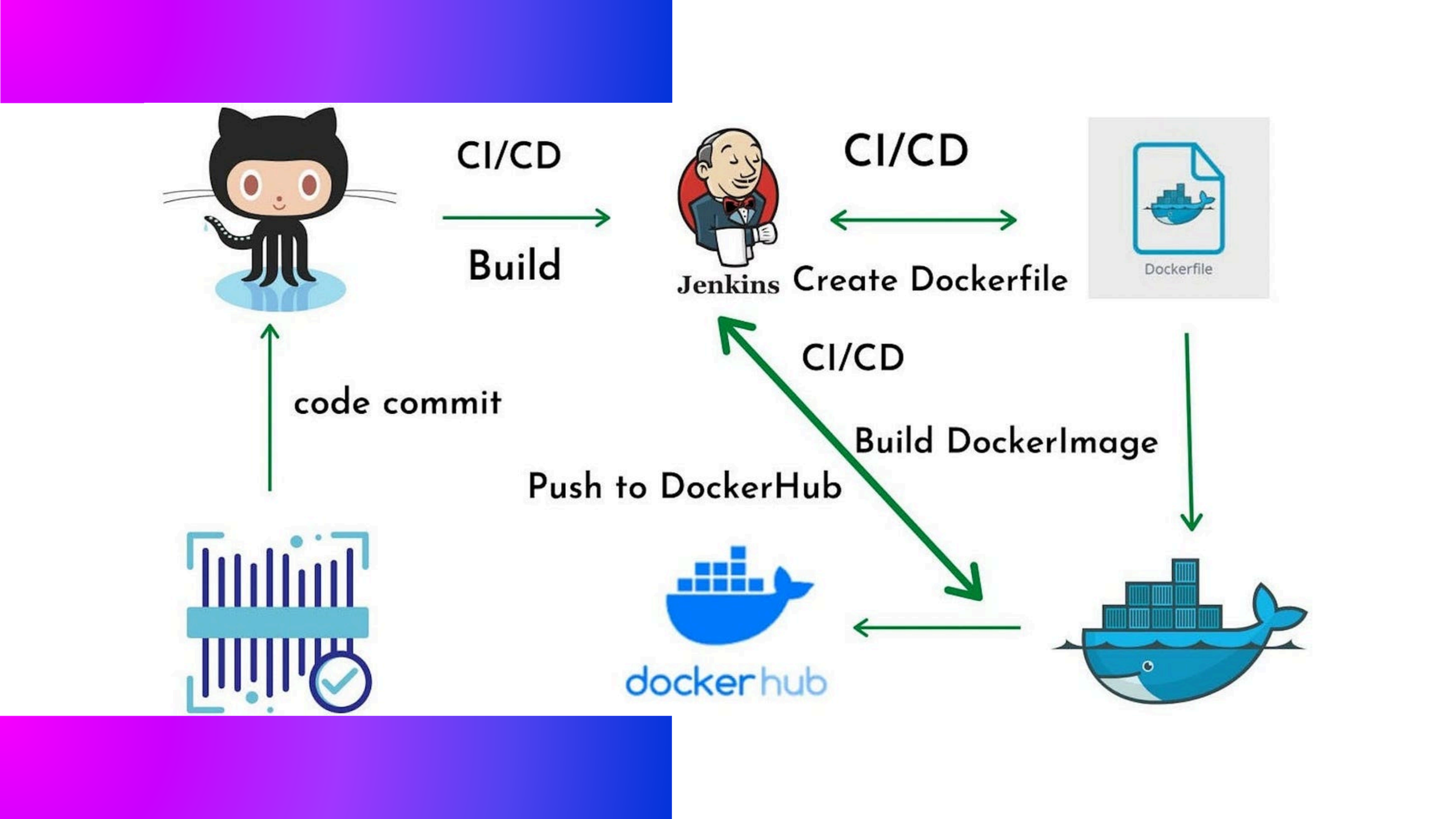
Maven

GitHub

Jenkins

Docker

Docker Hub


CI/CD
Build
Jenkins
CI/CD
Create Dockerfile
Dockerfile
code commit
CI/CD
Build DockerImage
Push to DockerHub
dockerhub

Todo.java | ToDoListAppl... | TodoControll... | application.... | Jenkinfile

```java
package com.example.to_do.list;

public class Todo {
    private Long id;
    private String task;
    private boolean completed;

    public Todo() {}

    public Todo(Long id, String task, boolean completed) {
        this.id = id;
        this.task = task;
        this.completed = completed;
    }

    // Getters and Setters
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }

    public String getTask() { return task; }
    public void setTask(String task) { this.task = task; }

    public boolean isCompleted() { return completed; }
    public void setCompleted(boolean completed) { this.completed = completed; }
}
```

Todo.java | ToDoListAppl... | TodoControll... | application.... | Jenkinfile

```java
package com.example.to_do.list;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class ToDoListApplication {

//  public class Todo {
//      private Long id;
//      private String task;
//      private boolean completed;
//
//      // constructors, getters, setters
//  }

    public static void main(String[] args) {
        SpringApplication.run(ToDoListApplication.class, args);
    }

}
```

Todo.java | ToDoListAppl... | TodoControll... | application.... | Jenkinfile

```java
package com.example.to_do.list;

import com.example.to_do.list.Todo;
import org.springframework.web.bind.annotation.*;

import java.util.*;

@RestController
@RequestMapping("/todos")
public class TodoController {

    private Map<Long, Todo> todoMap = new HashMap<>();
    private Long idCounter = 1L;

    @GetMapping
    public Collection<Todo> getAllTodos() {
        return todoMap.values();
    }

    @PostMapping
    public Todo createTodo(@RequestBody Todo todo) {
        todo.setId(idCounter++);
        todoMap.put(todo.getId(), todo);
        return todo;
    }

    @GetMapping("/{id}")
    public Todo getTodoById(@PathVariable Long id) {
        return todoMap.get(id);
    }
}
```

Todo.java | ToDoListAppl... | TodoControll... | application.... | Jenkinfile

```
spring.application.name=to-do-list
server.port=8082
```

Todo.java | ToDoListAppl... | TodoControll... | application.... | Dockerfile

```
# Use official OpenJDK image as base (Java 17)
FROM openjdk:17-jdk-alpine

# Set working directory inside the container
WORKDIR /app

# Copy the jar file from your target folder into the container
COPY target/to-do-list-0.0.1-SNAPSHOT.jar app.jar

# Expose the port your app runs on
EXPOSE 8082

# Run the jar file
ENTRYPOINT ["java", "-jar", "app.jar"]
```

ToDoListAppl... | TodoControll... | application.... | Jenkinfile | Dockerfile

```
pipeline {
    agent any

    environment {
        IMAGE_NAME = 'yourdockerhubusername/todo-list-api'
    }

    stages {
        stage('Checkout') {
            steps {
                checkout([
                    $class: 'GitSCM',
                    branches: [[name: '*/main']],
                    userRemoteConfigs: [[
                        url: 'https://github.com/pranjalixxx/Docker_project.git',
                        credentialsId: 'github-credentials'
                    ]]
                ])
            }
        }

        stage('Build with Maven') {
            steps {
                sh './mvnw clean package -DskipTests'
            }
        }

        stage('Build Docker Image') {
            steps {
                script {
                    docker.build("${IMAGE_NAME}")
                }
            }
        }
```

ToDoListAppl... | TodoControll... | application.... | Jenkinfile | Dockerfile

```
                branches: [[name: '*/main']],
                userRemoteConfigs: [[
                    url: 'https://github.com/pranjalixxx/Docker_project.git',
                    credentialsId: 'github-credentials'
                ]]
            ])
        }
    }

    stage('Build with Maven') {
        steps {
            sh './mvnw clean package -DskipTests'
        }
    }

    stage('Build Docker Image') {
        steps {
            script {
                docker.build("${IMAGE_NAME}")
            }
        }
    }

    stage('Push to Docker Hub') {
        steps {
            script {
                docker.withRegistry('https://index.docker.io/v1/', 'dockerhub-credentials') {
                    docker.image("${IMAGE_NAME}").push("latest")
                }
            }
        }
    }
}
```

Todo.java | ToDoListAppli... | TodoControll... | application.... | Jenkinfile | Dockerfile | to-do-list/p...

https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.5.5</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.example</groupId>
    <artifactId>to-do-list</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>to-do-list</name>
    <description>Demo project for Spring Boot</description>
    <url/>
    <licenses>
        <license/>
    </licenses>
    <developers>
        <developer/>
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
```

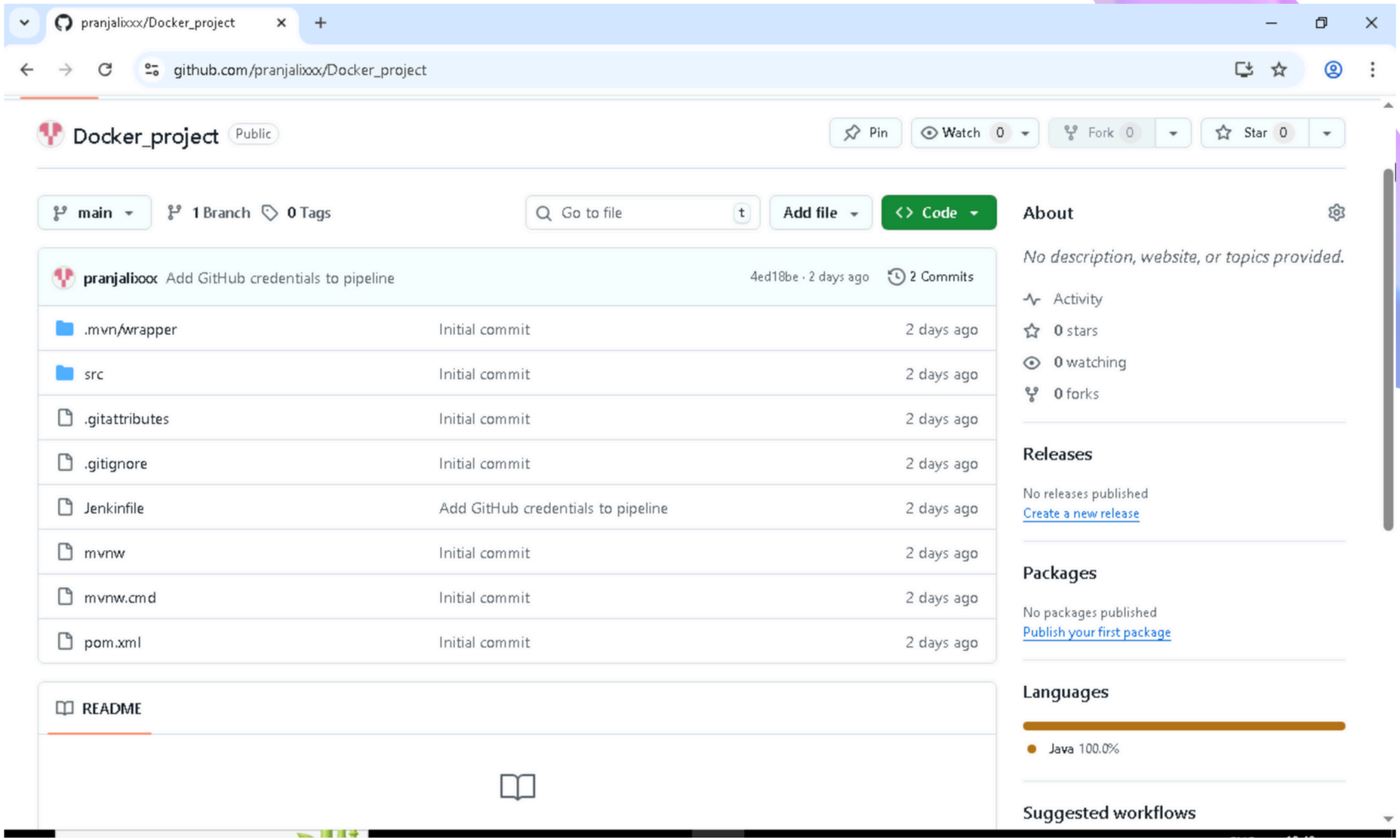
Todo.java | ToDoListAppli... | TodoControll... | application.... | Jenkinfile | Dockerfile

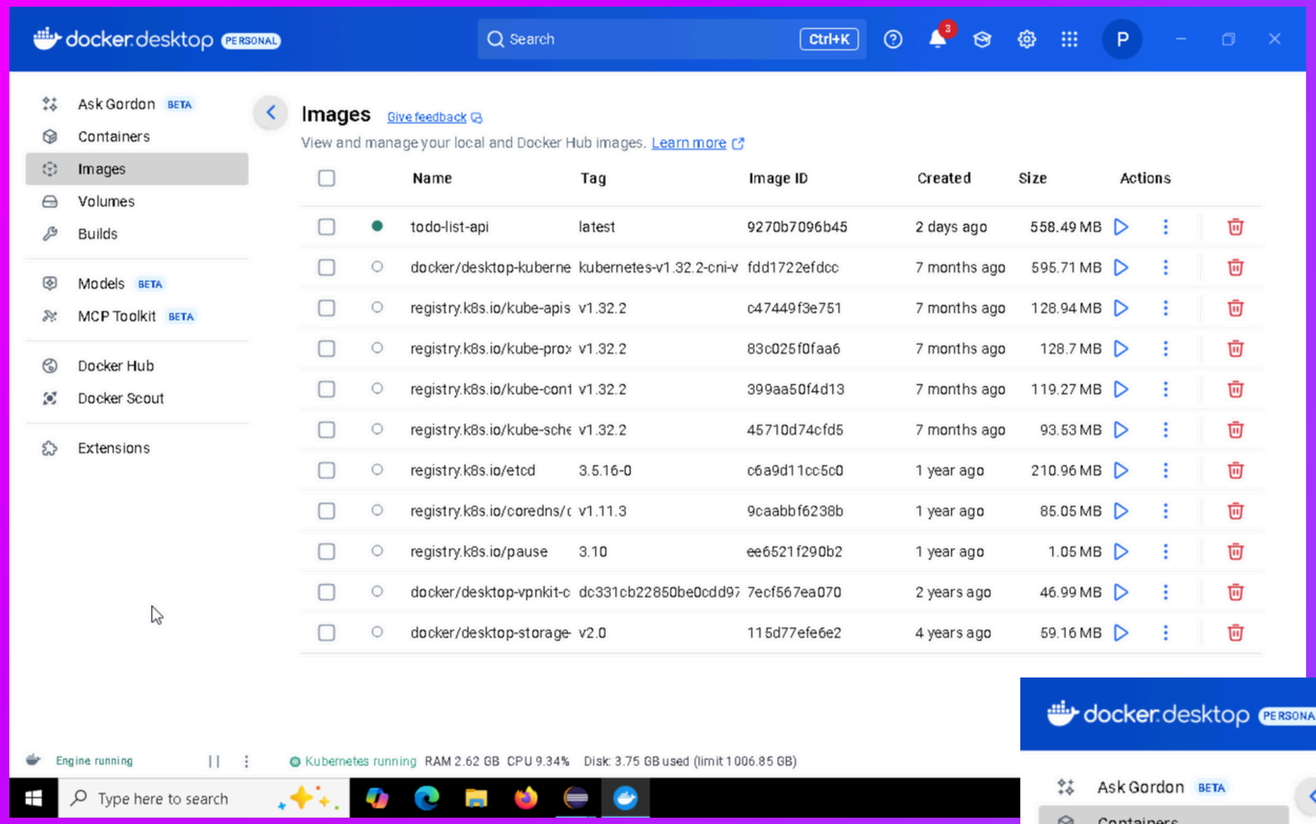
```xml
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
        <tag/>
        <url/>
    </scm>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

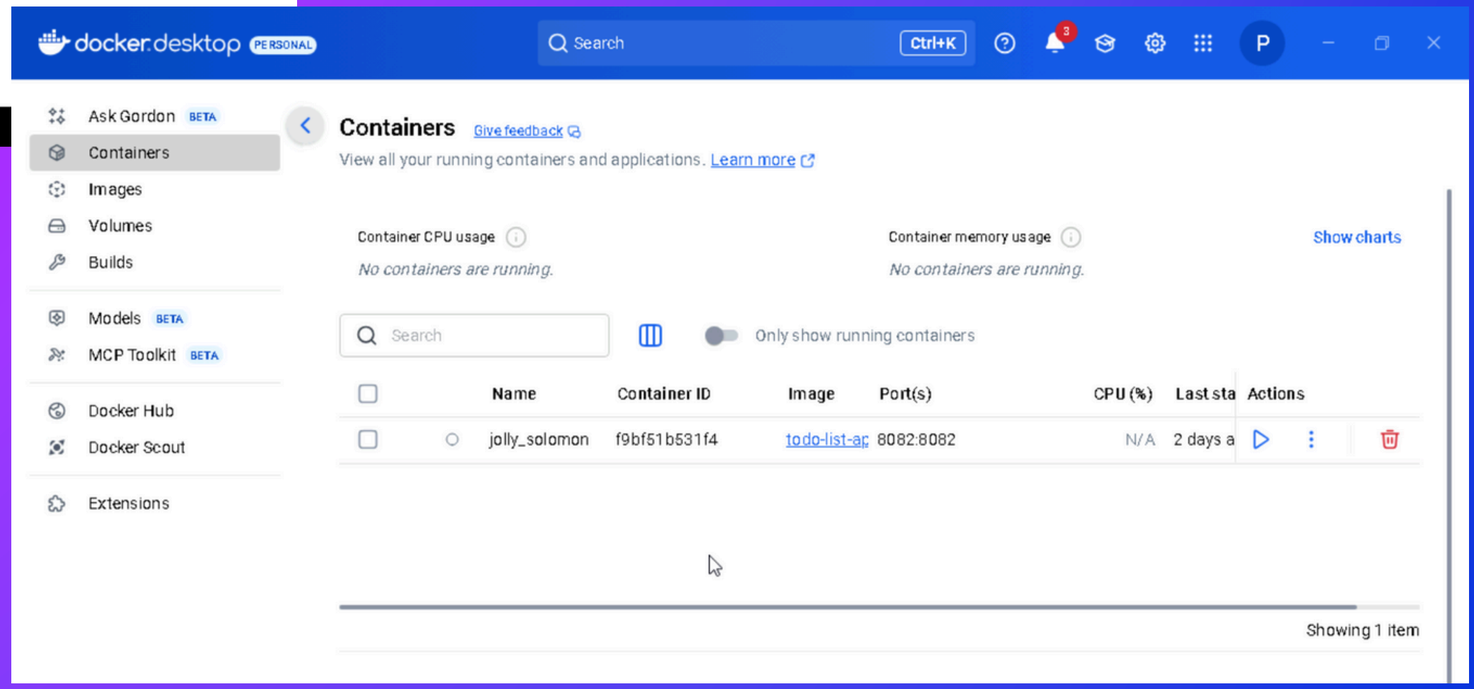
```
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/r73q2gee1u3id4yc4u1o9c19t

C:\Users\Administrator\Downloads\demoDocker>docker image
Usage:  docker image COMMAND

Manage images

Commands:
  build       Build an image from a Dockerfile
  history     Show the history of an image
  import      Import the contents from a tarball to create a filesystem image
  inspect     Display detailed information on one or more images
  load        Load an image from a tar archive or STDIN
  ls          List images
  prune       Remove unused images
  pull        Download an image from a registry
  push        Upload an image to a registry
  rm          Remove one or more images
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.

C:\Users\Administrator\Downloads\demoDocker>docker images
REPOSITORY                                TAG
IMAGE ID       CREATED          SIZE
bubblec/demodocker                        latest
55efaf5d821f   54 seconds ago   819MB
docker/desktop-kubernetes                 kubernetes-v1.32.2-cni-v1.6.0-critools-v1.31.1-cri-dockerd-v0.3.16-1-debian
fdd1722efdcc   6 months ago     596MB
```

```
c002e8f6062: Pushing [==========================>                        ]  92.27MB/203.9MB
ogin Succeeded

c002e8f6062: Pushing [==========================>                        ]  94.37MB/203.9MB
sing default tag: latest
he push refers to repository [docker.io/bubblec/demodocker]
c002e8f6062: Pushing [==========================>                        ]  94.37MB/203.9MB
2218e2d5087: Pushed
262579e8e45: Pushing [=============================================>     ]  40.89MB/44.96MB
eab4e2287a5: Pushed
c002e8f6062: Pushing [=============>                                     ]  54.53MB/203.9MB

c002e8f6062: Pushing [==========================>                        ]  94.37MB/203.9MB



c002e8f6062: Pushing [==========================>                        ]  94.37MB/203.9MB


c002e8f6062: Pushed
atest: digest: sha256:55efaf5d821faa4c8cb2b11b809ee2ec27a0f86a030e22c758b06fad2fcf3421 size: 856

:\Users\Administrator\Downloads\demoDocker>
```

```
7ecf567ea070   2 years ago      47MB
docker/desktop-storage-provisioner        v2.0
115d77efe6e2   4 years ago      59.2MB

C:\Users\Administrator\Downloads\demoDocker>docker run -p
flag needs an argument: 'p' in -p

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run 'docker run --help' for more information

C:\Users\Administrator\Downloads\demoDocker>docker run -p 9090:8083
docker: 'docker run' requires at least 1 argument

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

See 'docker run --help' for more information

C:\Users\Administrator\Downloads\demoDocker>docker run -p 9090:8083 bubblec/demodocker

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.5)
7c002e8f6062: Pushing [==========================>                        ]  91.23MB/203.9MB
7c002e8f6062: Pushing [==========================>                        ]  90.18MB/203.9MB
```


pranjalixxx/Docker_project
github.com/pranjalixxx/Docker_project
Docker_project
Public
Pin
Watch 0
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Add file
Code
pranjalixxx Add GitHub credentials to pipeline
4ed18be · 2 days ago
2 Commits
.mvn/wrapper Initial commit 2 days ago
src Initial commit 2 days ago
.gitattributes Initial commit 2 days ago
.gitignore Initial commit 2 days ago
Jenkinfile Add GitHub credentials to pipeline 2 days ago
mvnw Initial commit 2 days ago
mvnw.cmd Initial commit 2 days ago
pom.xml Initial commit 2 days ago
README
About
No description, website, or topics provided.
Activity
0 stars
0 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Java 100.0%
Suggested workflows


docker desktop PERSONAL
Search Ctrl+K
Ask Gordon BETA
Containers
Images
Volumes
Builds
Models BETA
MCP Toolkit BETA
Docker Hub
Docker Scout
Extensions
Images
Give feedback
View and manage your local and Docker Hub images. Learn more
Name Tag Image ID Created Size Actions
todo-list-api latest 9270b7096b45 2 days ago 558.49 MB
docker/desktop-kuberne kubernetes-v1.32.2-cni-v fdd1722efdcc 7 months ago 595.71 MB
registry.k8s.io/kube-apis v1.32.2 c47449f3e751 7 months ago 128.94 MB
registry.k8s.io/kube-prox v1.32.2 83c025f0faa6 7 months ago 128.7 MB
registry.k8s.io/kube-con1 v1.32.2 399aa50f4d13 7 months ago 119.27 MB
registry.k8s.io/kube-sche v1.32.2 45710d74cfd5 7 months ago 93.53 MB
registry.k8s.io/etcd 3.5.16-0 c6a9d11cc5c0 1 year ago 210.96 MB
registry.k8s.io/coredns/c v1.11.3 9caabbf6238b 1 year ago 85.05 MB
registry.k8s.io/pause 3.10 ee6521f290b2 1 year ago 1.05 MB
docker/desktop-vpnkit-c dc331cb22850be0cdd97 7ecf567ea070 2 years ago 46.99 MB
docker/desktop-storage- v2.0 115d77efe6e2 4 years ago 59.16 MB
Engine running
Kubernetes running RAM 2.62 GB CPU 9.34% Disk: 3.75 GB used (limit 1006.85 GB)
Type here to search


docker desktop PERSONAL
Search Ctrl+K
Ask Gordon BETA
Containers
Images
Volumes
Builds
Models BETA
MCP Toolkit BETA
Docker Hub
Docker Scout
Extensions
Containers
Give feedback
View all your running containers and applications. Learn more
Container CPU usage
No containers are running.
Container memory usage
No containers are running.
Show charts
Search
Only show running containers
Name Container ID Image Port(s) CPU (%) Last sta Actions
jolly_solomon f9bf51b531f4 todo-list-ap 8082:8082 N/A 2 days a
Showing 1 item

Conclusion

Automated and reliable CI/CD setup

Consistent environment using Docker

Code and build traceability via GitHub and Jenkins

Future Scope:

Add Kubernetes for deployment

Add SonarQube for code quality checks

Use a cloud provider (AWS/GCP) for hosting

Thank you