

## **Foundation Application :Backend**

### **Introduction**

The Foundation Application is designed to streamline and automate the deployment process using Docker and Jenkins pipelines. This application facilitates continuous integration and delivery, ensuring that your software is built, tested, and deployed efficiently.

### **Features**

- Automated Deployment: Utilizes Jenkins pipelines for automated building and deployment of the application.
- Docker Integration: Leverages Docker for containerization, ensuring consistent environments from development to production.
- Real-Time Updates: Supports continuous integration to apply updates and new features rapidly and reliably.

## **Getting Started**

### **Prerequisites**

Before starting, ensure you have the following installed:

- Docker and Docker Compose
- Jenkins with Pipeline and Docker plugins installed
- Foundation Application
- Git

## **Installation and Setup**

### **Clone the Repository:**

```
def sourceDir = "/home/naveen/Desktop/Projects/Foundation_Backend/"

    def destDir = "/home/naveen/.jenkins/workspace//Foundation_Backend"

    sh "rsync -av --exclude='__pycache__/'
/home/naveen/Desktop/Projects/Foundation_Backend/ ${env.WORKSPACE}"/"cd
FoundationApp
```

### **Jenkins Pipeline Setup:**

Create a new Jenkins pipeline job.

Use the provided Jenkinsfile from the repository as the pipeline script.

## Configuration

### Environment Variables:

- DOCKER\_CMD: The Docker command to use.
- DOCKERFILE\_PATH: Path to the Dockerfile.
- CONTAINER\_PORT: The port number for the Docker container.
- IMAGE\_TAG: Tag for the Docker image.

### API Keys and Secrets:

Store any API keys or secrets as environment variables in Jenkins.

## Usage

The Jenkins pipeline automates several tasks:

- Cloning the Repository: Copies your project to the workspace.
- Dockerfile Check: Ensures the Dockerfile exists before proceeding.
- Config.yaml Check: Verifies the presence of a config.yaml file.
- Building the Image: Constructs the Docker image with a new tag.
- Checking for Port Conflict: Ensures the specified port is available.
- Running the Image: Deploys the application using Docker Compose.
- Listing Docker Images: Displays all available Docker images.

To trigger the pipeline, push changes to your repository or manually start the pipeline job in Jenkins.

 pipeline