**JENKINS PROCESS FOR FOUNDATION-BACKEND**

**Step 1: Jenkins Setup**

* Open Jenkins and navigate to "Manage Jenkins".
* Click on "Global Tool Configuration" and add Docker.
* Go to "Manage Plugins" and install necessary plugins for Docker integration.

**Step 2: Create a New Job Setup**

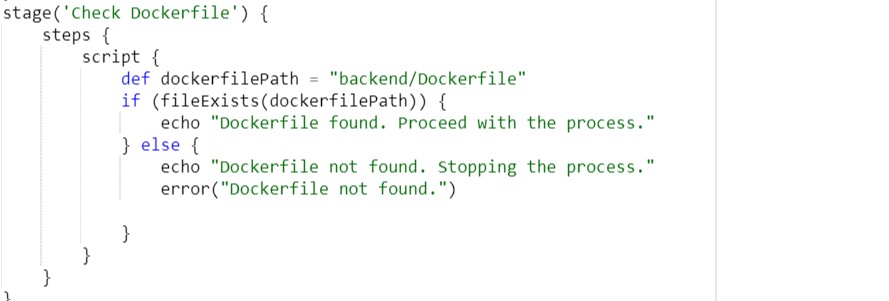
* Open Jenkins and create a new job for the application.
* Select "Pipeline" as the job type and click "OK".
* Navigate to the Jenkins Dashboard and click on the created job to configure it.
* Start writing the pipeline Groovy script.
* After writing the script, click on "Apply" and then "Save".
* Once saved, navigate back to the Jenkins Dashboard and click "Build Now".
* You can view the progress of the build in the Stage View.
* Go to the build history and click on the recent build number to view details

**Step 3: Clone Repository**

* Navigate to the Jenkins Dashboard.
* Open the Jenkins pipeline job for your project.
* Execute the "Clone Repository" stage to fetch the latest code from the repository.
* 

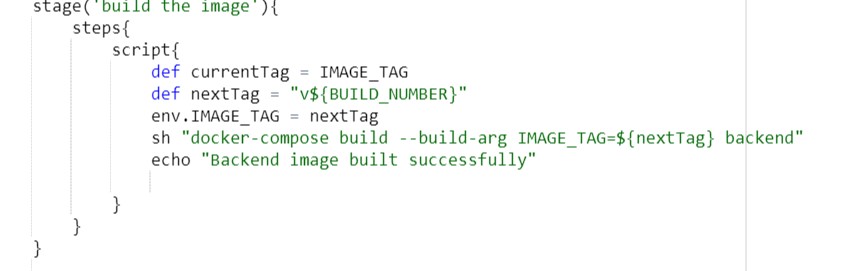
**Step 4: Check Dockerfile**

* Continue from the previous step.
* Execute the "Check Docker file" stage to ensure the Dockerfile for the backend exists.
* If the Dockerfile exists, proceed; otherwise, stop the process.



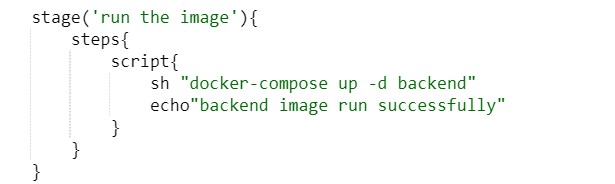
**Step 5: Build the Docker Image**

* Execute the "Build the Docker Image" stage.
* The script will build the Docker image, with tag .



**Step 6: Run the Backend Image**

* Execute the "Run the Backend Image" stage to start the Docker container from the built image.
* The container for the backend will start in detached mode.



**step 7: List Docker Images:**

* This stage lists the Docker images present on the Jenkins agent.
* It uses the docker images command to display the list of images.
* After listing the images, it echoes a message indicating that the Docker images list has been displayed.

