Demo for Product deployment with CI/CD pipeline in a Demo environment

Script ?

```
1 → pipeline {
2
        agent any
3
4 +
        stages {
5 +
            stage('Checkout') {
6 *
                steps {
 7
                    git branch: 'product-demo', url: 'https://github.com/moosasharieff/group2-global-green-init.git'
 8
9
10
11 *
            stage('Build Docker image') {
12 +
                steps {
13 *
                    script {
14
                        // Change directory to the client folder
15 *
                        dir('client') {
                            // Build Docker image using the Dockerfile inside the client folder
16
17
                            sh 'docker build -t green .'
18
19
20
21
22
23 *
            stage('Run Docker container') {
24 +
                steps {
25 *
                    script {
26
                        // Run Docker container
27
                        sh 'docker run -it -d -p 81:80 green'
28
29
30
31
32
33 *
        post {
34 +
            success {
35
                echo 'Pipeline executed successfully!'
36
37
38
```

This is a Continuous Integration/Continuous Deployment (CI/CD) pipeline configured to deploy a product demo in a demo environment. The pipeline is defined using Jenkinsfile syntax, allowing for automation of the deployment process.

Stages:

Checkout:

This stage checks out the '**product-demo**' branch from the specified GitHub repository (https://github.com/moosasharieff/group2-global-green-init.git).

Build Docker image:

In this stage, the Docker image is built using the Dockerfile located inside the 'client' folder of the project repository. The Docker image is tagged as 'green'.

Run Docker container:

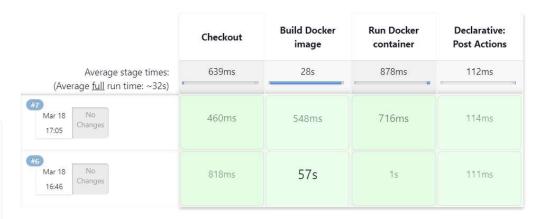
The Docker container is run with the 'green' image. It is started in interactive mode (-it), detached mode (-d), and mapped to port 81 on the host system (-p 81:80). This stage launches the application associated with version V2 at the URL https://v2.globalgreeninit.world/.

Post-build Actions:

If the pipeline execution is successful, a message indicating successful execution is echoed.

```
Successfully built f5e566f43058
Successfully tagged green:latest
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run Docker container)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker run -it -d -p 81:80 green
0098956a2ec9964f9a32084c68a1dc10e19e383b72ba2d3ee32e3cf35ce08295
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline executed successfully!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Stage View



Permalinks

- Last build (#7), 6 hr 36 min ago
- Last stable build (#7), 6 hr 36 min ago
- Last successful build (#7), 6 hr 36 min ago
- Last failed build (#5), 7 hr 2 min ago
- Last unsuccessful build (#5), 7 hr 2 min ago
- Last completed build (#7), 6 hr 36 min ago

This pipeline demonstrates the automated deployment process of a product demo using a CI/CD pipeline, from checking out the code repository to building and running the Docker container in a demo environment.