Automate docker built and push using Jenkinsfile

Setup a Simple Flask App Project

Structure my-flask-app

- app.py
- requirements.txt
- Dockerfile
- Jenkinsfile

Code | Blame | 20 lines (14 loc) · 536 Bytes | Raw

```
# Use an official Python runtime as a parent image
FROM python:3.9-slim

# Set the working directory in the container
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Make port 5000 available to the world outside this container
EXPOSE 5000

# Define environment variable to avoid Python buffering
ENV PYTHONUNBUFFERED 1

# Run app.py when the container launches
CMD ["python", "app.py"]
```

Code Blame 10 lines (7 loc) · 170 Bytes

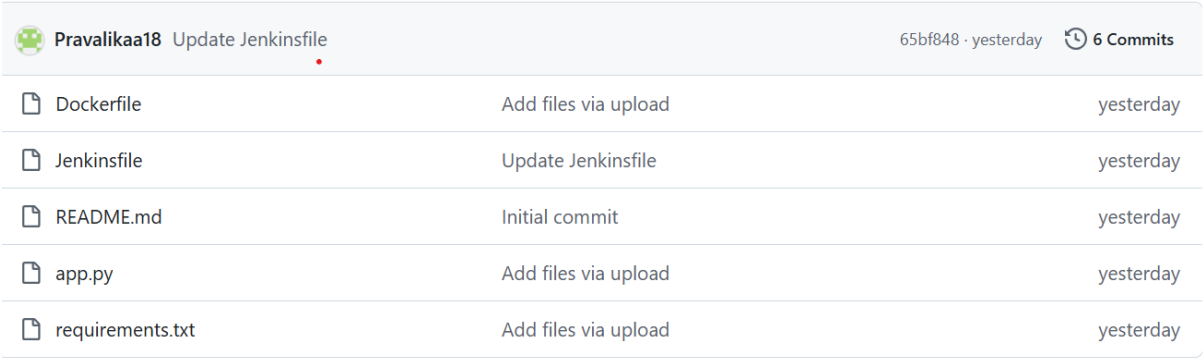
```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(debug=True)
```

Code Blame 29 lines (25 loc) · 746 Bytes Raw

```
pipeline {
    agent any

    environment {
        DOCKER_IMAGE = 'pravalikaa18/my_project:latest'
    }

    stages {
        stage('Clone Repository') {
            steps {
                git url:'https://github.com/pravalikaa18/jenkins_docker.git',branch: 'main'
            }
        }

        stage('Build Docker Image') {
            steps {
                sh 'docker build -t $DOCKER_IMAGE .'
            }
        }

        stage('Push Docker Image') {
            steps {
                withDockerRegistry([credentialsId: '395ce1a5-6982-4c78-9994-4a2e8dbce27a', url: 'https://index.docker.io/v1
                    sh 'docker push $DOCKER_IMAGE'
                }
            }
```

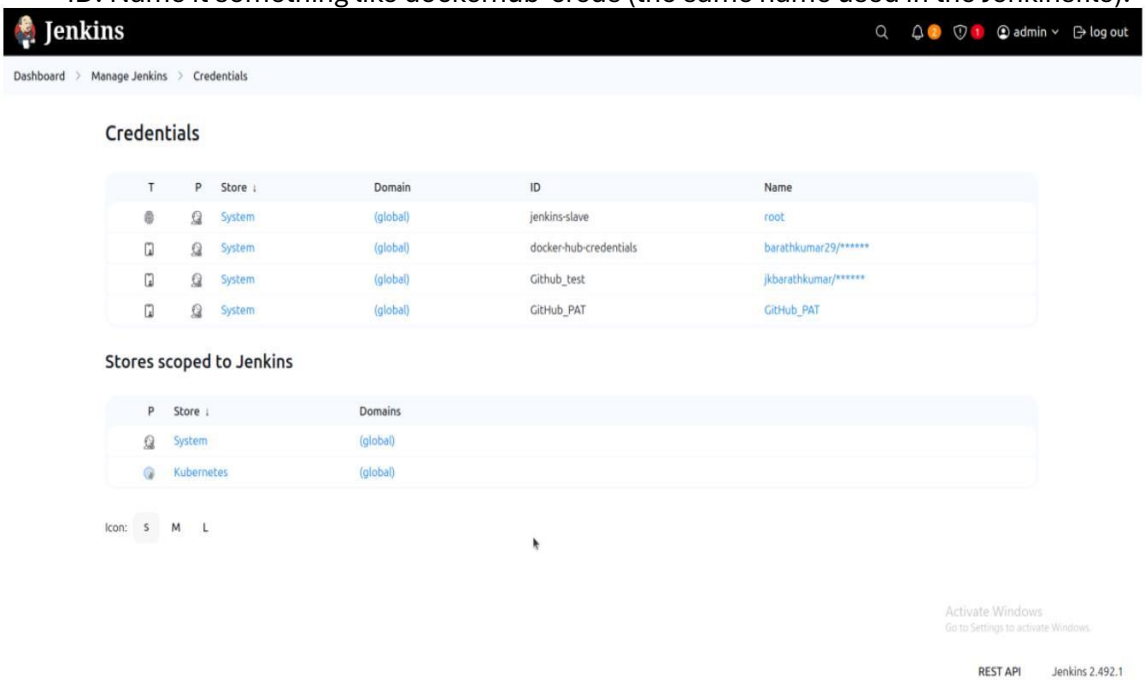
2. Push the Code to GitHub

- **Make sure you have a GitHub repository created for the project.**
- **Push all the files (app.py, requirements.txt, Dockerfile, Jenkinsfile) to the GitHub repositor**



3. Configure Docker Hub Credentials in Jenkins

- Go to Jenkins > Manage Jenkins > Manage Credentials.
- Add new credentials: o Username: Your Docker Hub username.
- Password: Your Docker Hub password (or token).
- ID: Name it something like dockerhub-creds (the same name used in the Jenkinsfile).



4.Create a New Pipeline in Jenkins

• In Jenkins, click New Item > Pipeline.

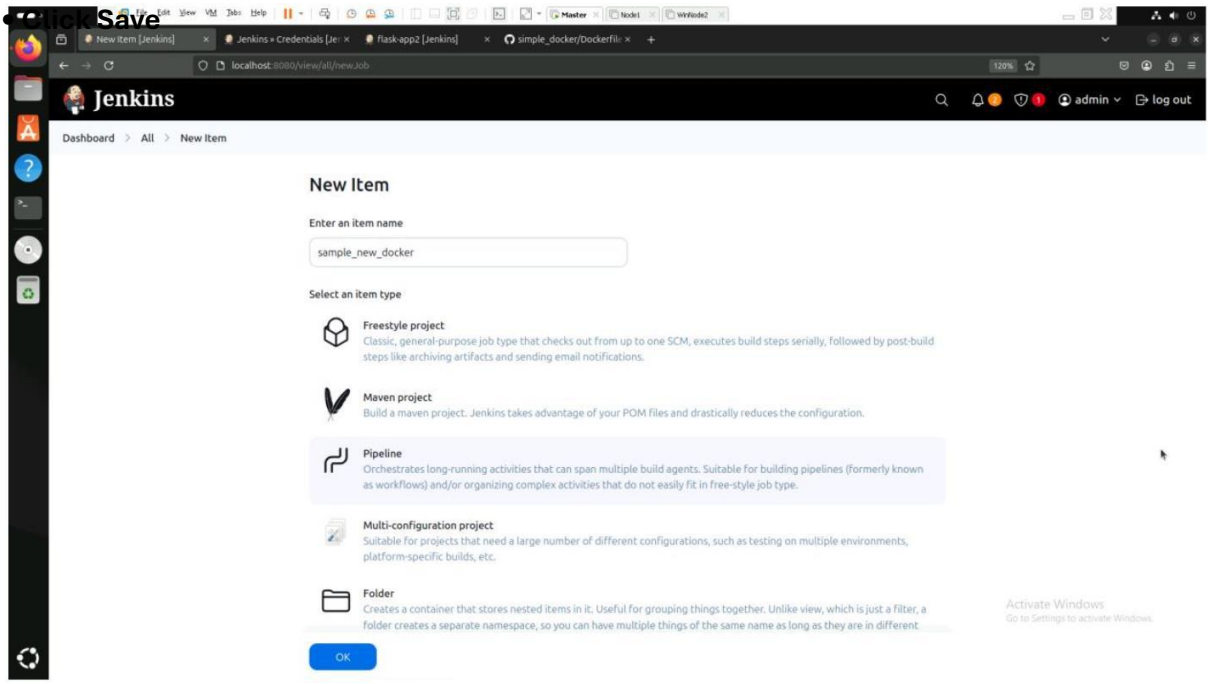
• Enter a name for the pipeline

. • Under Pipeline Definition, select Pipeline script from SCM.

Select Git as the SCM.

Enter the GitHub repository URL (https://github.com/your-username/my flask-app.git). Set the branch (typically master or main).

• Click Save

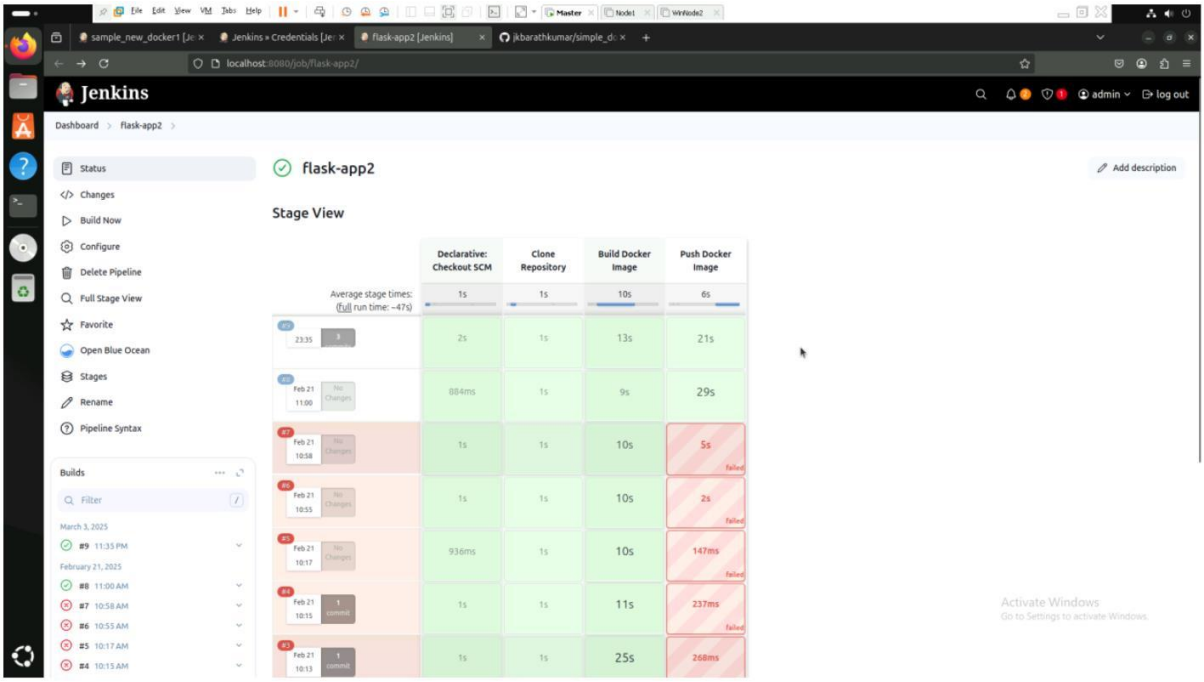


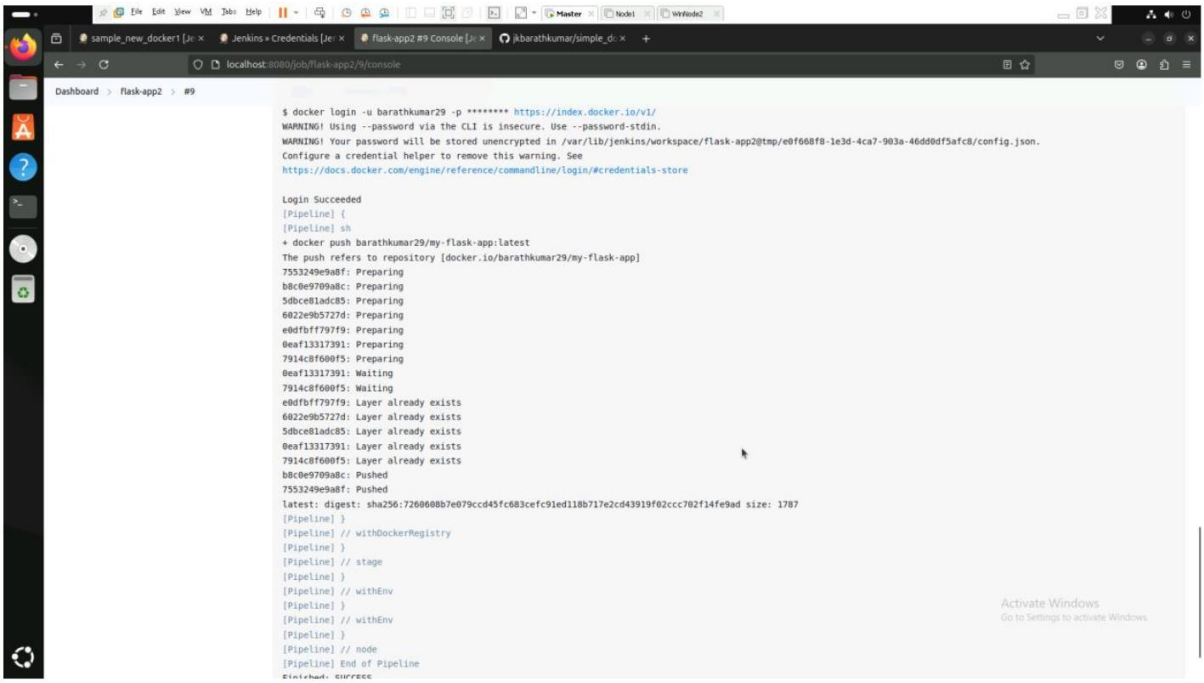
5. Click Build Now

Click Build Now in Jenkins to trigger the build.

Jenkins will:

- **Checkout the code from GitHub.**
- **Build the Docker image.**
- **Push the image to Docker Hub.**


Jenkins
admin
log out
Dashboard
flask-app2
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Favorite
Open Blue Ocean
Stages
Rename
Pipeline Syntax
Builds
Filter
flask-app2
Add description
Stage View
Declarative: Checkout SCM
Clone Repository
Build Docker Image
Push Docker Image
Average stage times:
(full run time: ~47s)
1s
1s
10s
6s
2s
1s
13s
21s
884ms
1s
9s
29s
1s
1s
10s
5s
failed
1s
1s
10s
2s
failed
936ms
1s
10s
147ms
failed
1s
1s
11s
237ms
failed
1s
1s
25s
268ms
Activate Windows
Go to Settings to activate Windows.


Dashboard
flask-app2
#9
$ docker login -u barathkumar29 -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/flask-app2@tmp/e0f668f8-1e3d-4ca7-903a-46dd0df5afc8/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 push barathkumar29/my-flask-app:latest
The push refers to repository [docker.io/barathkumar29/my-flask-app]
7553249e9a8f: Preparing
b8c0e9709a8c: Preparing
5dbce81adc85: Preparing
6022e9b5727d: Preparing
e0dfbff797f9: Preparing
0eaf13317391: Preparing
7914c8f600f5: Preparing
0eaf13317391: Waiting
7914c8f600f5: Waiting
e0dfbff797f9: Layer already exists
6022e9b5727d: Layer already exists
5dbce81adc85: Layer already exists
0eaf13317391: Layer already exists
7914c8f600f5: Layer already exists
b8c0e9709a8c: Pushed
7553249e9a8f: Pushed
latest: digest: sha256:7260608b7e079ccd45fc683cefc91ed118b717e2cd43919f02ccc702f14fe9ad size: 1787
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Activate Windows
Go to Settings to activate Windows.

6. Verify Docker Image on Docker Hub

- **After the build finishes, log into your Docker Hub account.**
- **You should see the my-flask-app image under Repositories with the latest tag.**