

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 Copy Download Edit View Source
```

```
1  # Use an official Python runtime as a parent image
2  FROM python:3.9-slim
3
4  # Set the working directory in the container
5  WORKDIR /app
6
7  # Copy the current directory contents into the container at /app
8  COPY . /app
9
10 # Install any needed dependencies
11 RUN pip install --no-cache-dir -r requirements.txt
12
13 # Make port 5000 available to the world outside this container
14 EXPOSE 5000
15
16 # Define environment variable to avoid Python buffering
17 ENV PYTHONUNBUFFERED 1
18
19 # Run app.py when the container launches
20 CMD ["python", "app.py"]
```

Code**Blame**

10 lines (7 loc) · 170 Bytes

```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def hello_world():
7      return 'Hello, World!'
8
9  if __name__ == '__main__':
10     app.run(debug=True)
```

Code**Blame**

29 lines (25 loc) · 746 Bytes

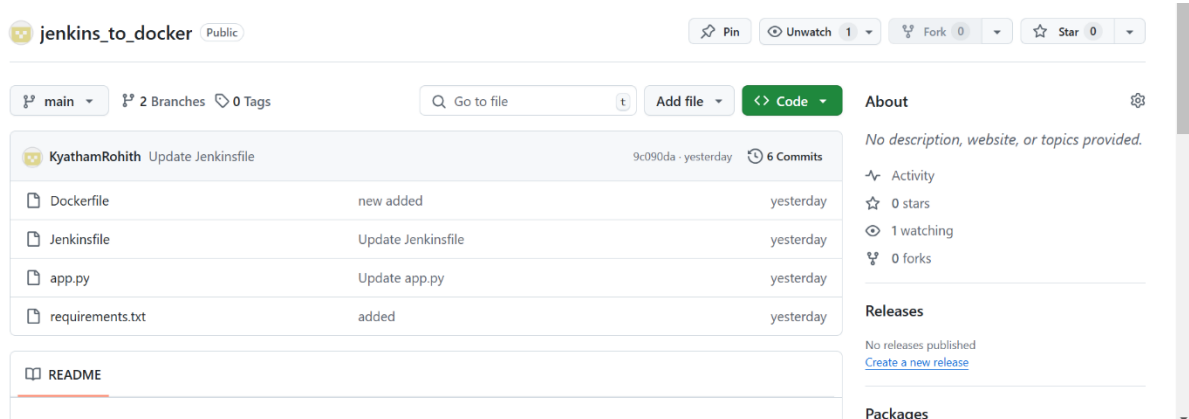
Raw



```
1  pipeline {
2      agent any
3
4      environment {
5          DOCKER_IMAGE = 'pravalikaa18/my_project:latest'
6      }
7
8      stages {
9          stage('Clone Repository') {
10              steps {
11                  git url:'https://github.com/pravalikaa18/jenkins_docker.git',branch: 'main'
12              }
13          }
14
15          stage('Build Docker Image') {
16              steps {
17                  sh 'docker build -t $DOCKER_IMAGE .'
18              }
19          }
20
21          stage('Push Docker Image') {
22              steps {
23                  withDockerRegistry([credentialsId: '395ce1a5-6982-4c78-9994-4a2e8dbce27a', url: 'https://index.docker.io/v1
24                  sh 'docker push $DOCKER_IMAGE'
25              }
26          }
27      }
28  }
```

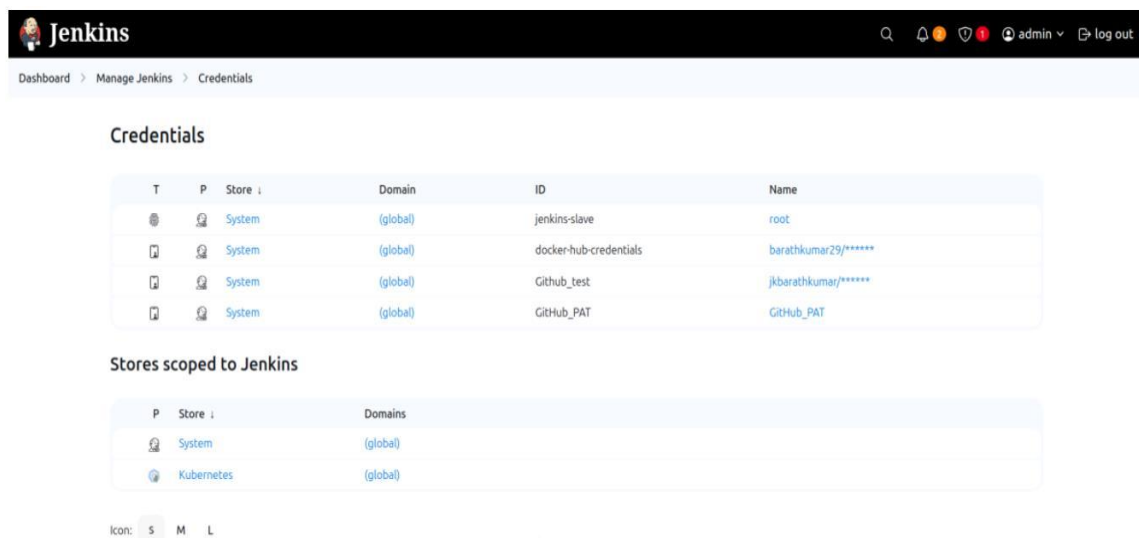
2. Push the Code to GitHub

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



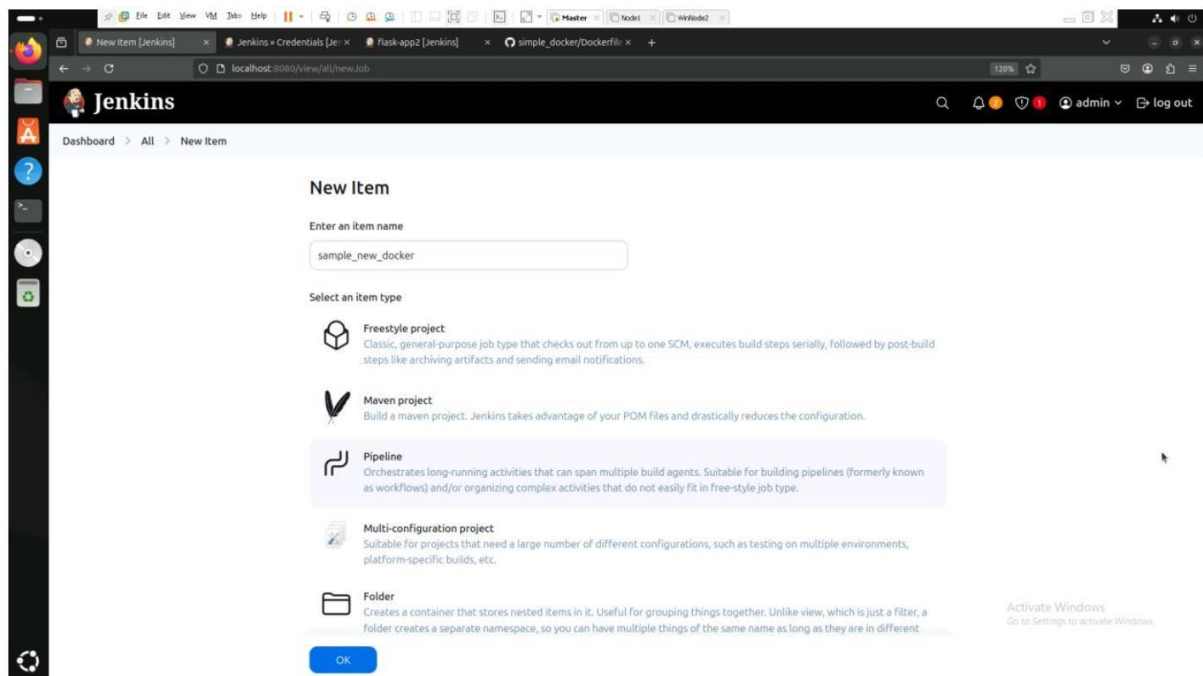
3. Configure Docker Hub Credentials in Jenkins

- 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/myflask-app.git>). Set the branch (typically master or main).
- Click Save

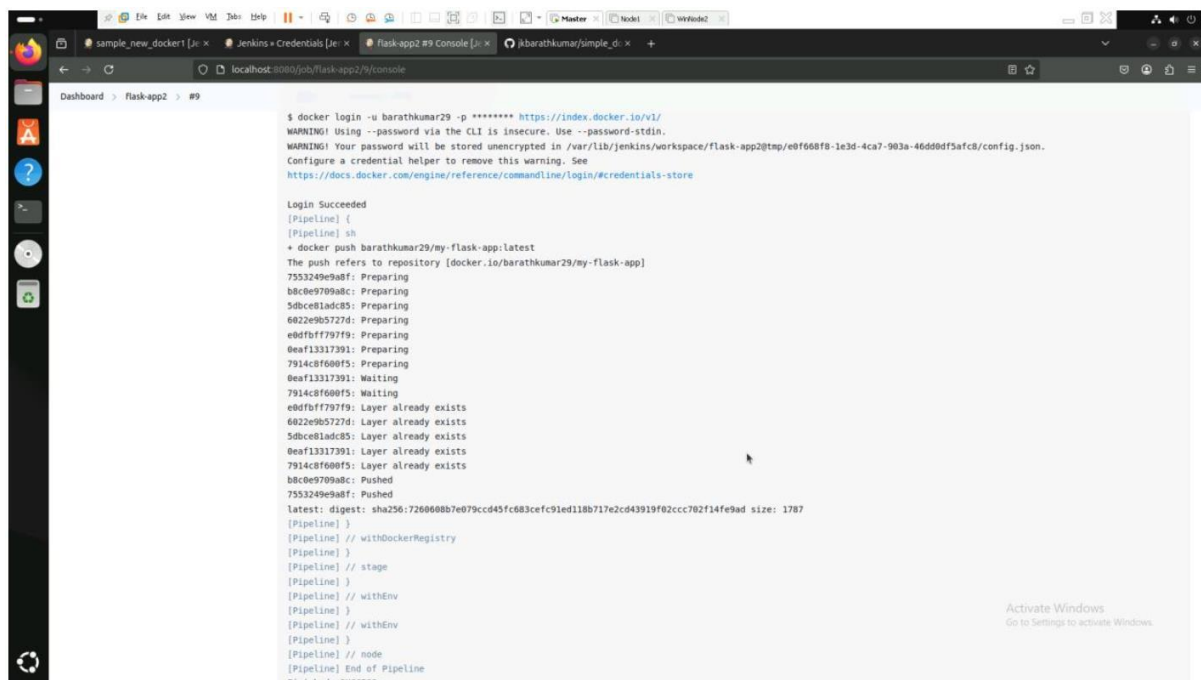
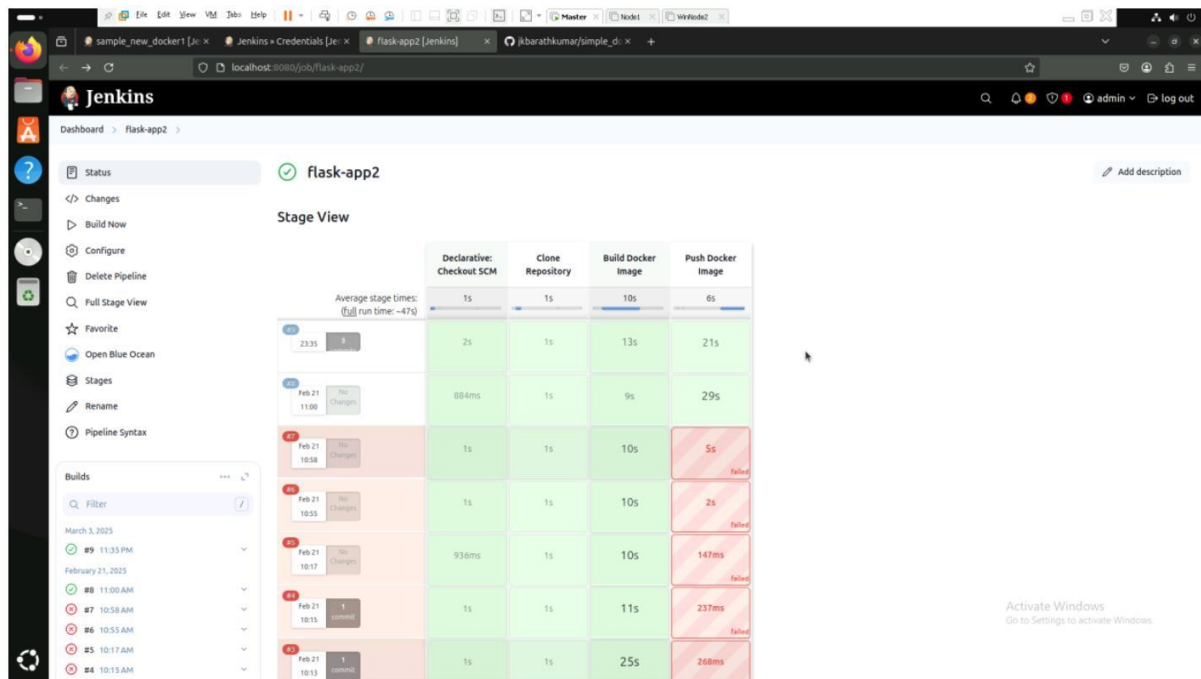


5. Click Build Now

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.



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.

The screenshot shows the Docker Hub interface. The top navigation bar includes 'dockerhub', 'Explore', 'Repositories' (selected), 'Organizations', and 'Usage'. A search bar contains 'Search Docker Hub' and a 'ctrl+K' shortcut. On the right, there are icons for settings, notifications, a refresh button, a grid icon, and a user profile icon labeled 'R'.

Below the navigation bar, the user 'rohith1305' is selected in a dropdown. A search bar contains 'Search by repository name', and a filter dropdown is set to 'All content'. A blue button 'Create a repository' is visible.

The main content area displays a table of repositories:

Name	Last Pushed ↑	Contains	Visibility	Scout
rohith1305/myapp2	about 22 hours ago	IMAGE	Public	Inactive
rohith1305/myapp1	about 23 hours ago	IMAGE	Public	Inactive
rohith1305/myapp	about 23 hours ago	IMAGE	Public	Inactive

Below the table, it shows '1-3 of 3' with navigation arrows.

On the right side, there is a graphic with three padlocks (red, blue, green) connected by dashed lines, with the text 'Create an organization' and a link 'Create and manage users'.

The Windows taskbar at the bottom shows the Start button, a search bar, and several application icons. The system tray on the right shows the language 'ENG US', network status, volume, and the date '05-03-2025' and time '15:59'.