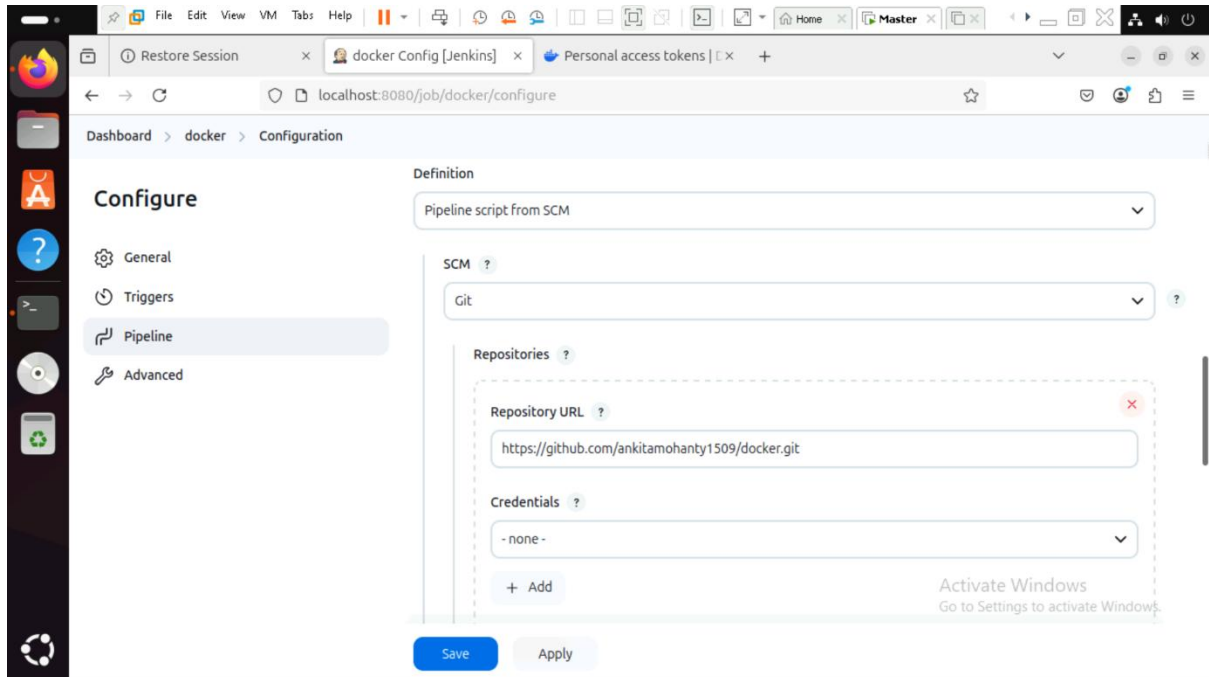
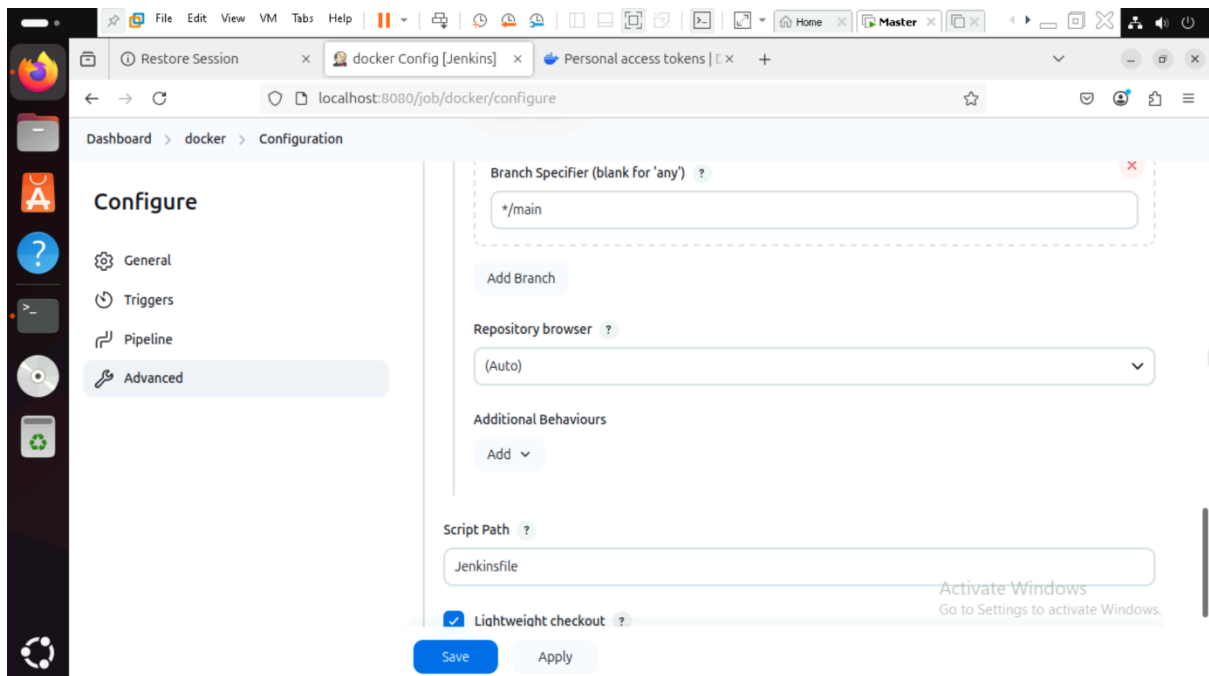


Deploy a Python Flask App using Docker

Step 1: Jenkins Pipeline



Step 2 : Save the Pipeline



Step 3: Build and run the docker container

The screenshot shows the Jenkins web interface in a browser window. The left sidebar contains navigation options like 'Dashboard', 'Changes', 'Build Now', 'Configure', 'Delete Pipeline', 'Full Stage View', 'Favorite', 'Open Blue Ocean', 'Stages', 'Rename', and 'Pipeline Syntax'. The main area displays the 'Stage View' for a pipeline named 'docker'. It shows a sequence of stages: 'Declarative: Checkout SCM' (1s), 'Clone Repository' (1s), 'Build Docker Image' (12s), and 'Push Docker Image' (8s). Below these, a table shows the execution details for each stage, including the commit hash, start time, and duration. The 'Push Docker Image' stage is highlighted in red, indicating a failure. A tooltip for the failed stage shows the message 'failed'.

Stage	Declarative: Checkout SCM	Clone Repository	Build Docker Image	Push Docker Image
Average stage times: (full run time: ~39s)	1s	1s	12s	8s
#4 18:06 No Changes	967ms	1s	9s	25s
#3 18:04 1 commit	1s	1s	10s	5s

Step 4: Push the docker files to Github

The screenshot shows a GitHub repository page for a user named 'ankitamohanty1509'. The repository is named 'docker' and is public. The page displays the commit history, showing a recent commit 'Update Jenkinsfile' by 'ankitamohanty1509' 5 minutes ago. The commit details show that the 'Jenkinsfile' was updated. The repository also lists files: 'Dockerfile', 'Jenkinsfile', 'app.py', and 'requirements.txt'. The right sidebar contains information about the repository, including the number of stars (0), forks (0), and releases (0). There are also links to 'Create a new release' and 'Publish your first package'.