Varshni D S

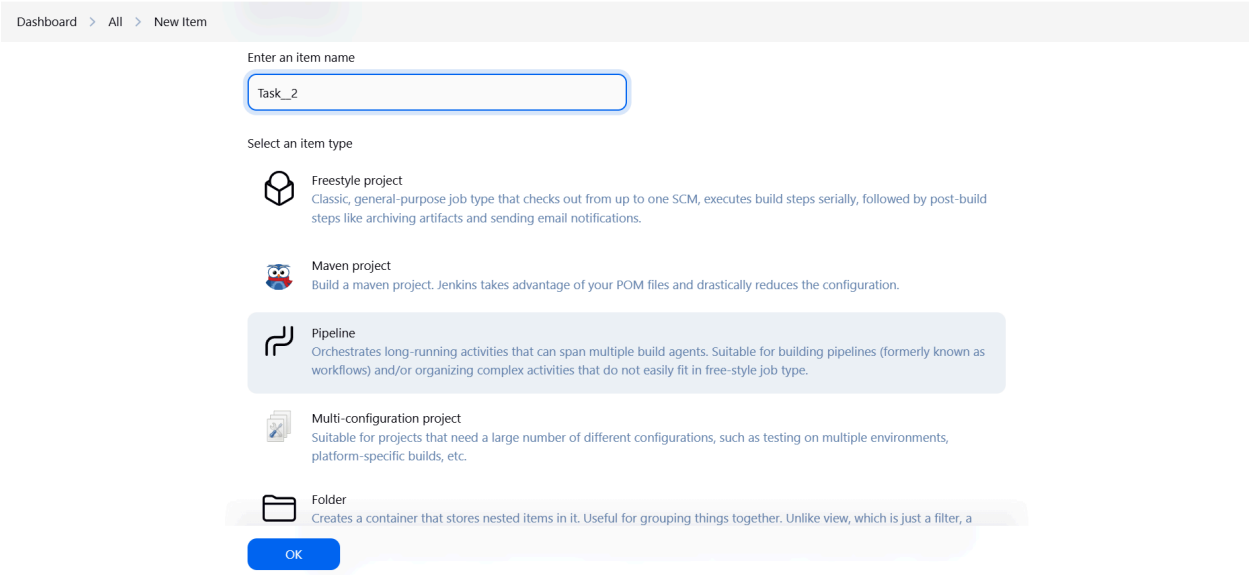
05.02.2025

Review of Jenkins Pipeline, Docker, and GitHub Repository Setup

1. Jenkins Dashboard & Job Creation

The first section displays the Jenkins dashboard, illustrating how to create a new job. The available options include:

- **Freestyle Project**: A flexible choice allowing various build steps and configurations.
- **Maven Project**: Uses POM files for efficient builds.
- **Multi-Configuration Project**: Supports projects requiring multiple configurations, such as matrix builds.
- **Pipeline Project**: Implements CI/CD automation using scripted or declarative pipelines.

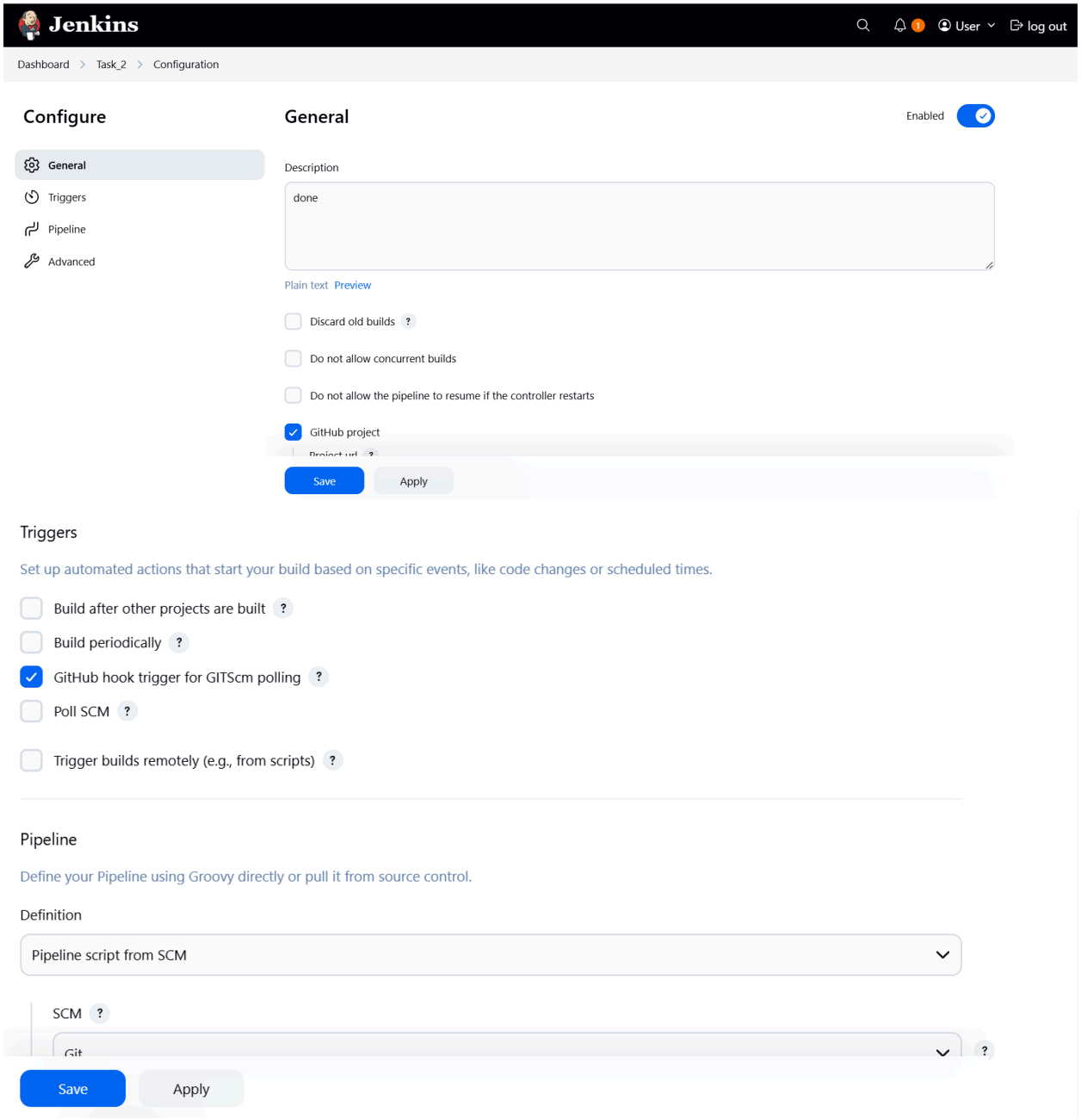


2. Jenkins Configuration Options

The second section highlights the Jenkins pipeline job configuration. Key settings include:

- **Pipeline Speed/Durability Override**: Adjusts execution priority and resource allocation.
- **Preserve Stashes from Completed Builds**: Retains stashes for debugging purposes.

- **GitHub Triggers**: Supports Webhooks and SCM polling for automatic builds.
- **Throttle Builds**: Restricts concurrent builds to optimize resource usage.

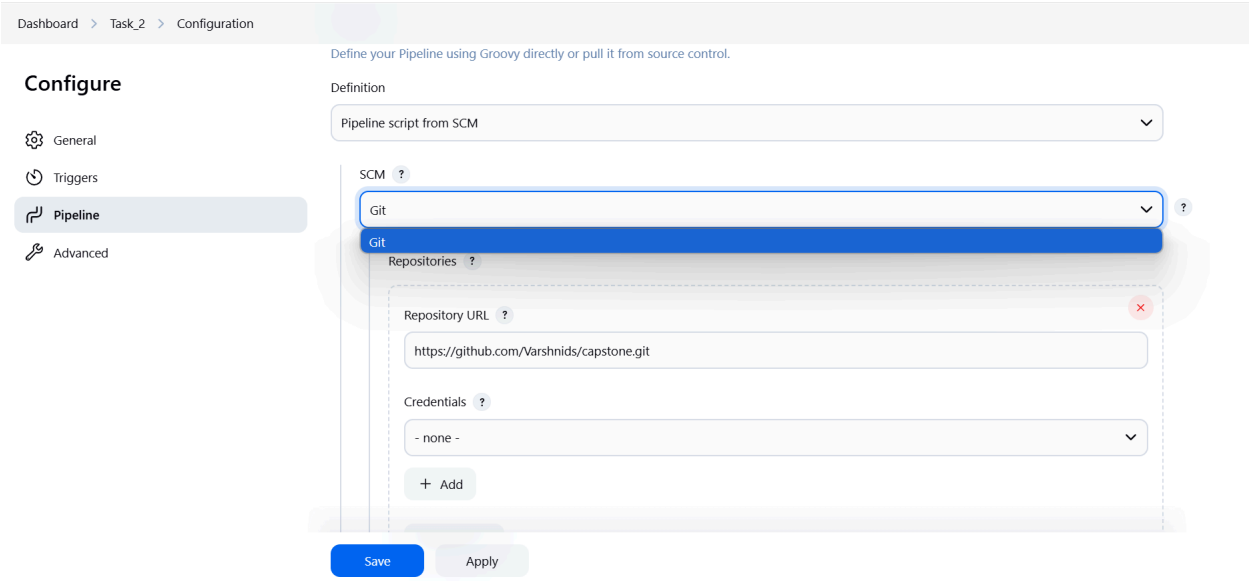


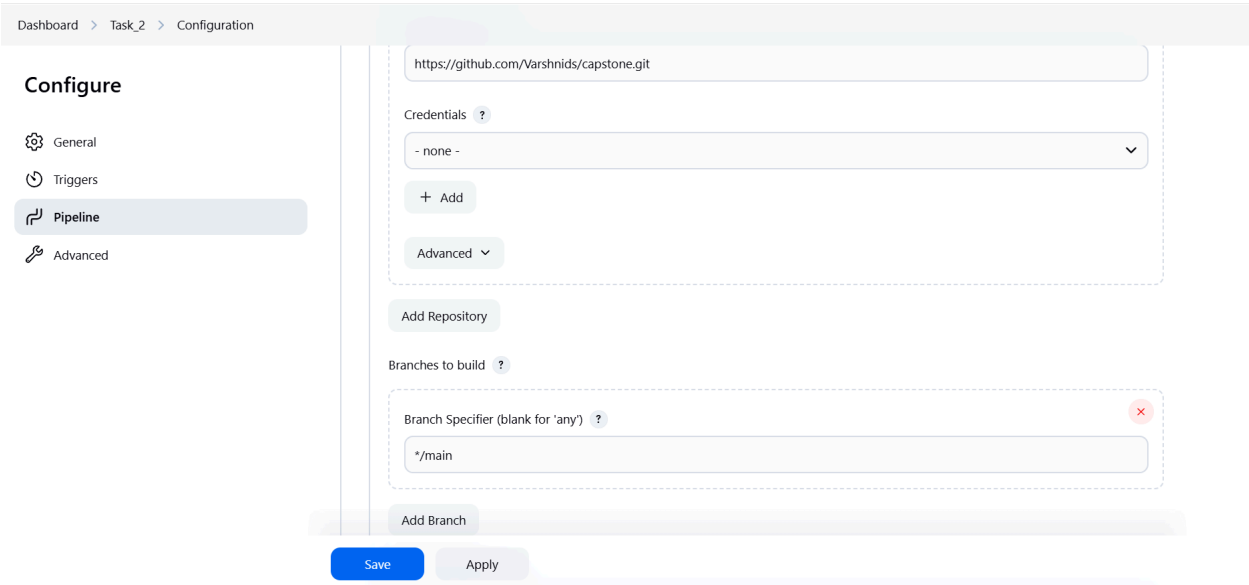
3. Pipeline Script & Source Control Management

This section displays:

- **Pipeline Script from SCM**: Jenkins fetches the pipeline script from a Git repository.
- **Repository URL**: https://github.com/varshnids/capstone.git.
- **Branch Specification**: Configured to main, ensuring Jenkins builds this branch.

- **Jenkinsfile Usage**: Defines pipeline stages and execution steps.





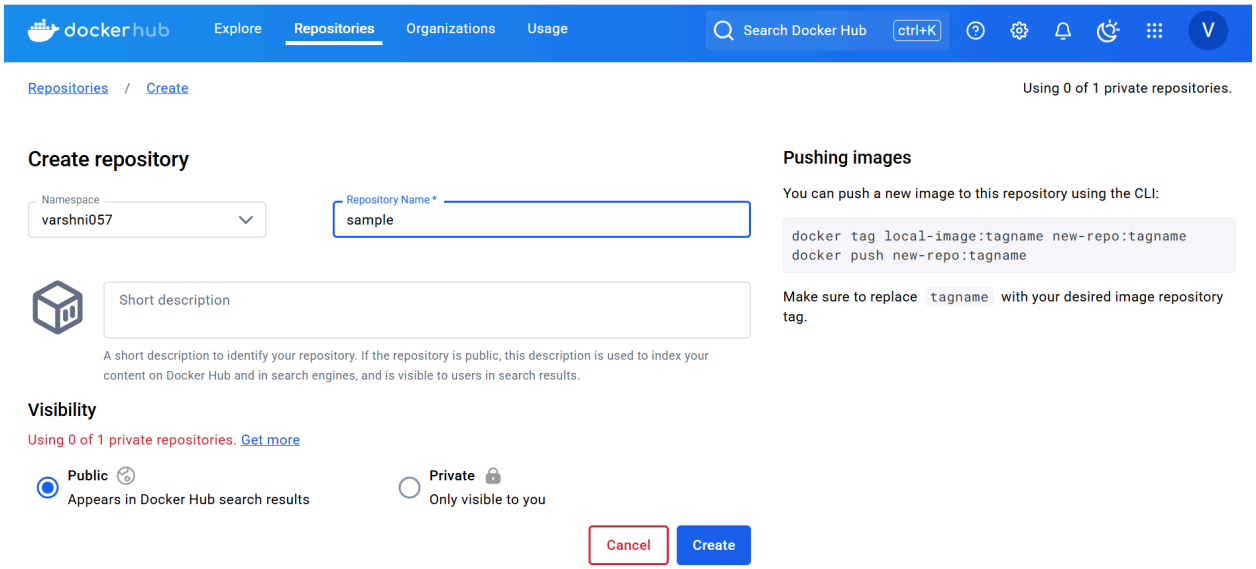
4. Docker Hub Repository Overview

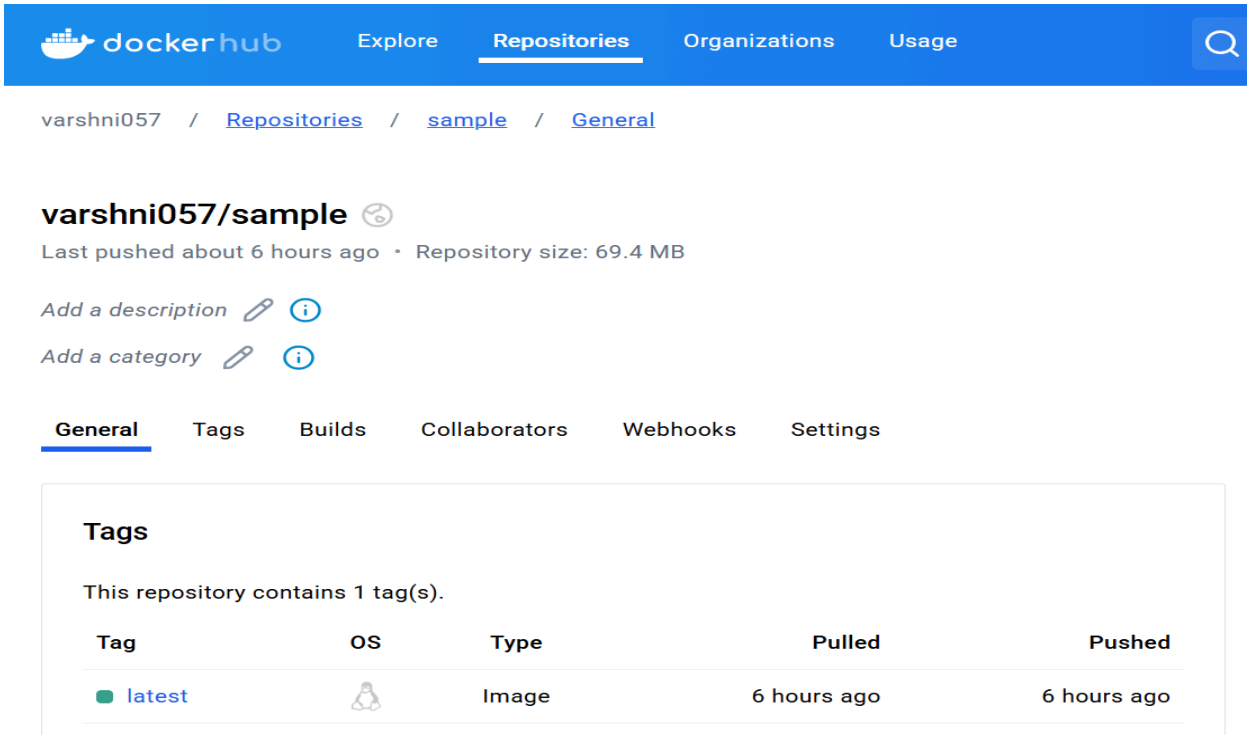
The fourth section provides details about the Docker Hub repository (varshni057/sample). Notable aspects include:

- **Public Repository**: Accessible to all users.

Pushing Docker Images: Uses CLI commands:
 docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname

- **Automated Builds**: Enables GitHub/Bitbucket integration for automatic image builds upon code updates.



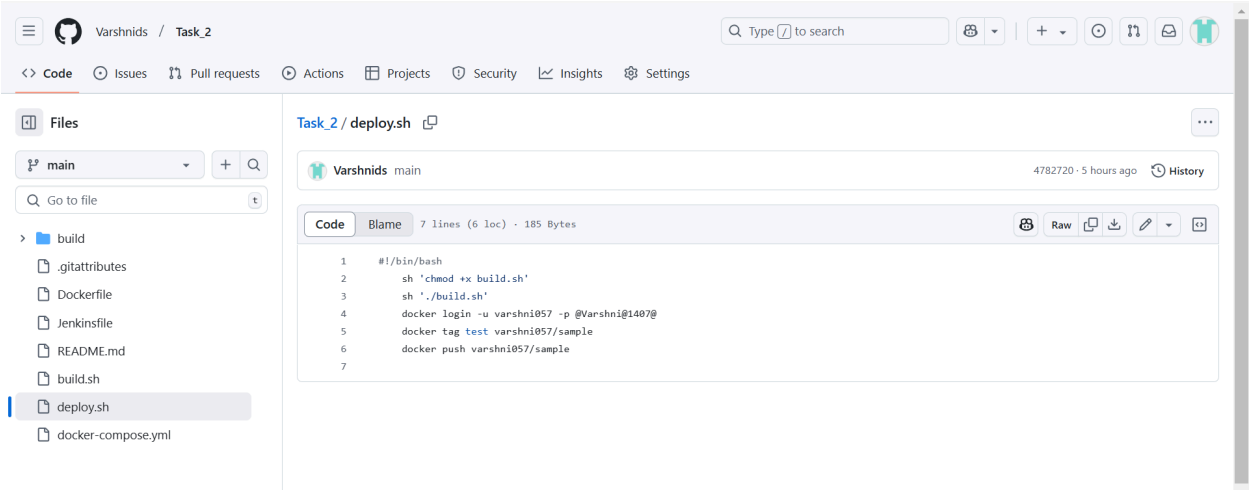


5. GitHub Repository & Deployment Workflow

This section highlights:

- **Repository Files**:
  - Dockerfile: Defines the containerized environment.

- Jenkinsfile: Contains CI/CD pipeline definitions.
- build.sh: Script for project build execution.
- deploy.sh: Script for application deployment.
- docker-compose.yml: Manages multi-service container applications.



Deployment Commands:
docker login -u username -p password
docker push varshni057/sample:tagname

6. Jenkins Build Execution Overview

The final section captures a successfully executed Jenkins pipeline:

- **Build Number**: #1, executed on **Feb 5, 2025**.
- **Repository Revision**: 3744f0a73044a984a5951b7530715b70ccbdees.
- **Execution Duration**: 29 seconds.
- **Console Output**: Displays detailed logs of the build process.

Dashboard > Task_2 > #4

- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#4'
- Timings
- Git Build Data
- Pipeline Overview
- Pipeline Console
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

#4 (05-Feb-2025, 10:51:15 am)

Add description | Keep this build forever

Started by user User

Started 5 hr 48 min ago
Took 20 sec

This run spent:

- 25 ms waiting;
- 20 sec build duration;
- 20 sec total from scheduled to completion.

git

Revision: e3a7577a3e12ca907dac5fd275469c9d530e2e3a
Repository: https://github.com/Varshnids/capstone.git

- refs/remotes/origin/main

No changes.

---

dockerhub | Explore | Repositories | Organizations | Usage | Search Docker Hub ctrl+K | V

varshni057 / Repositories / sample / General

Using 0 of 1 private repositories.

varshni057/sample

Last pushed about 5 hours ago • Repository size: 69.4 MB

Add a description

Add a category

Docker commands

Public view

To push a new tag to this repository:

```
docker push varshni057/sample:tagname
```

General | Tags | Builds | Collaborators | Webhooks | Settings

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	6 hours ago	5 hours ago

See all

Automated builds

Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

Conclusion

The provided images illustrate a structured CI/CD pipeline integrating Jenkins, GitHub, and Docker. The workflow follows these steps:

1. **Defining a Jenkins Pipeline (Jenkinsfile)**.
2. **Managing Source Code in GitHub**.
3. **Using Jenkins for Automated Builds**.
4. **Pushing Docker Images to Docker Hub**.
5. **Executing CI/CD Pipelines Efficiently**.

This setup enhances software development automation and streamlines deployment processes.