THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

Automated CI/CD Pipeline Implementation Using Jenkins on RHEL

Description:

This project involved setting up and configuring a full Jenkins-based Continuous Integration/Continuous Deployment (CI/CD) pipeline in a Red Hat Enterprise Linux (RHEL) environment. Jenkins, an open-source automation server, was used to automate the software development lifecycle by enabling efficient integration and delivery of code changes.

I started by installing and configuring Jenkins on a RHEL server and then proceeded to explore the Jenkins dashboard thoroughly. I created multiple freestyle jobs for different stages of the build and deployment process. Each job was configured with appropriate build triggers, such as GitHub webhooks and periodic polling, to ensure that code changes pushed to the repository were detected automatically.

Build steps were added to pull code from a GitHub repository, compile it, and perform automated tasks. Post-build actions were configured to notify stakeholders via email and archive artifacts. I created and organized views within Jenkins to manage and monitor related jobs more effectively, allowing for better project structure and visual clarity.

I installed several essential plugins such as Git, Pipeline, and Email Extension to extend Jenkins capabilities. Jenkins' console output logs were reviewed regularly to trace job execution, debug errors, and confirm successful builds. Additionally, I explored the use of credentials management and secure parameter passing, which is critical in real-world CI/CD use cases.

Throughout the project, I became familiar with job chaining, build dependencies, Jenkins' security model, and how to manage plugin updates. This project helped me gain a deep, practical understanding of how Jenkins supports continuous integration and delivery workflows, allowing development teams to deliver software faster and more reliably.

THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- **Check Jenkins status**

```
thisara@localhost:~
File  Edit  View  Search  Terminal  Help
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2025-07-10 10:26:21 PDT; 14s ago
 Main PID: 8038 (java)
    Tasks: 65 (limit: 48845)
   Memory: 621.9M
   CGroup: /system.slice/jenkins.service
           └─8038 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=
```

- **Creating a project**

New Item

Enter an item name

Jobtest1

Select an item type

Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

- **Create build steps**

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell ?

Command

See the list of available environment variables

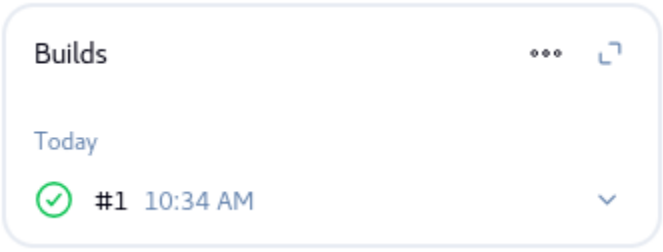
```
echo "hello world"
```

THISARA KANDAGE

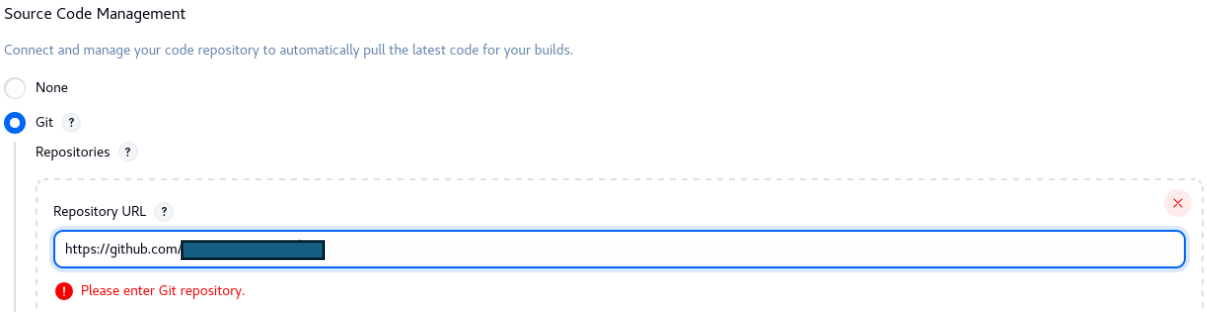
UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

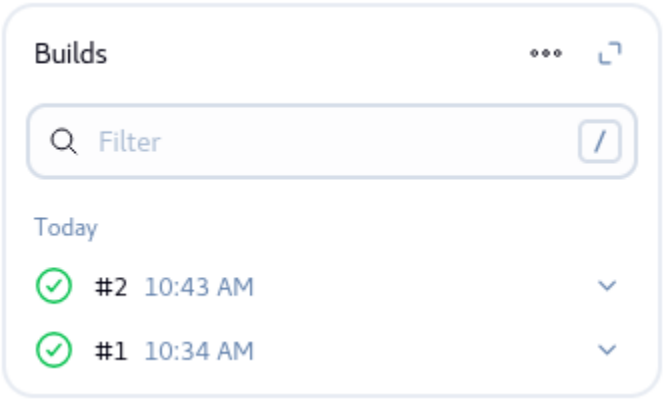
- **Build is done**



- **Pull a GitHub repo**



- **Again, built the project**



- **Console output**

```
First time build. Skipping changelog.
[Jobtest1] $ /bin/sh -xe /tmp/jenkins18104106786377823949.sh
+ echo 'hello world'
hello world
Finished: SUCCESS
```

- **If need to build happen again and again in fix interval of time.**
- **For that create another job**

New Item

Enter an item name

Jobtest2

Select an item type

Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

None

Git ?

Repositories ?

Repository URL ?

https://github.com/amrjeet/certproj.git

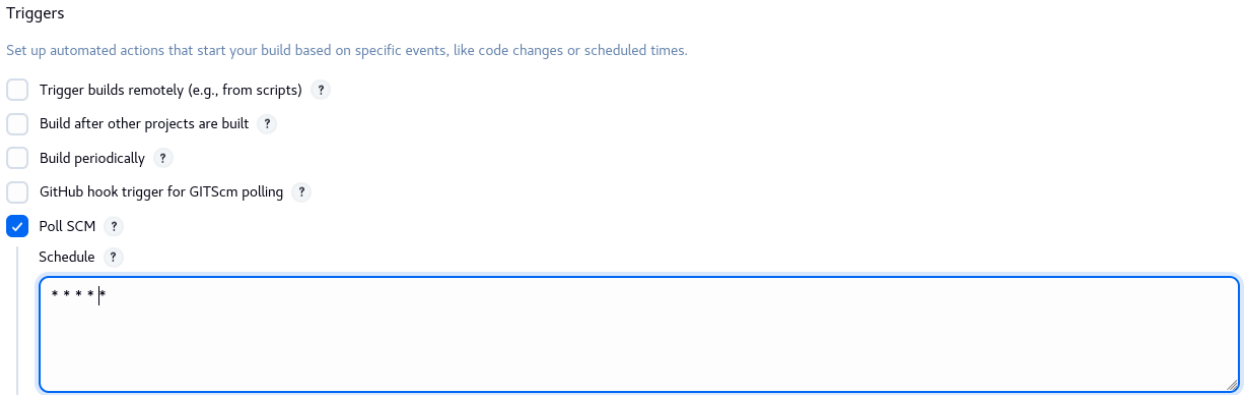
Please enter Git repository.

THISARA KANDAGE

UNDERGRADUATE - SLIIT

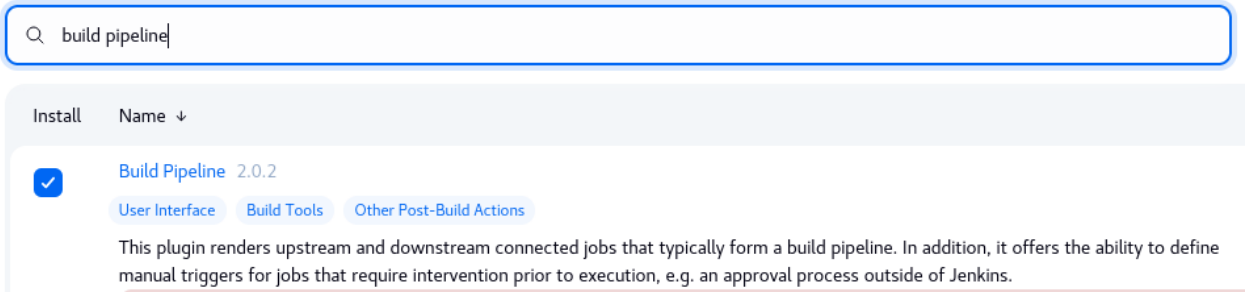
E-mail LinkedIn GitHub Website

- **Add Build triggers**



(5 starts for build for in every single minute)

- **Install required plugins**

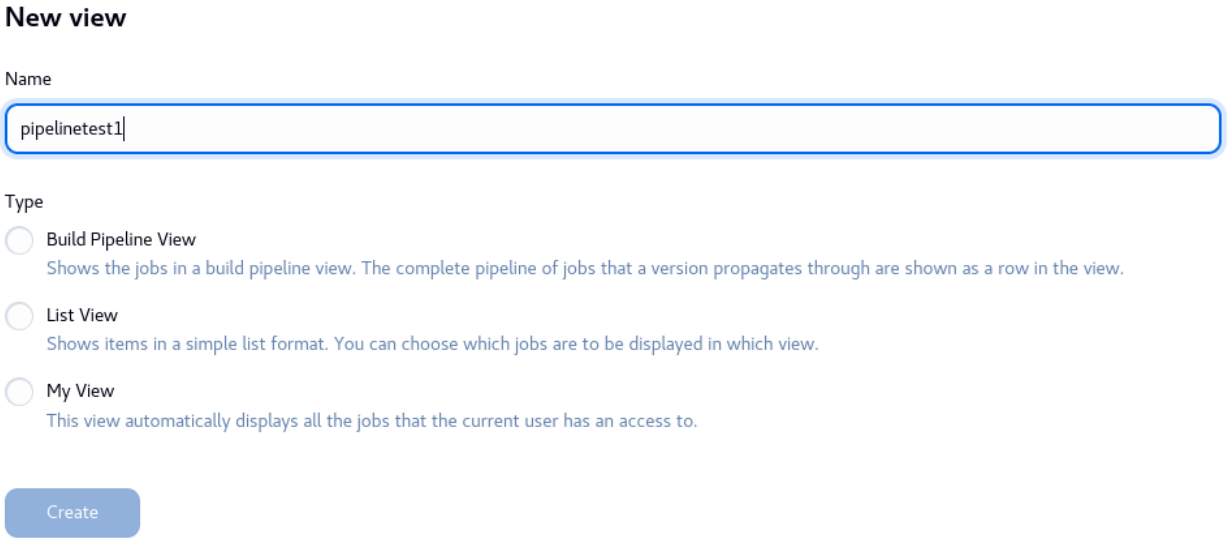


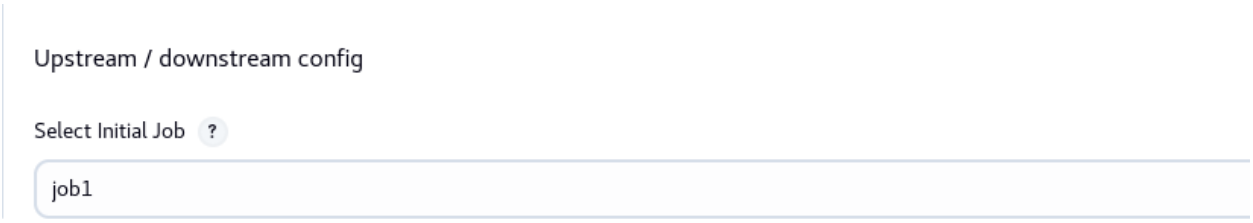
THISARA KANDAGE

UNDERGRADUATE - SLIIT

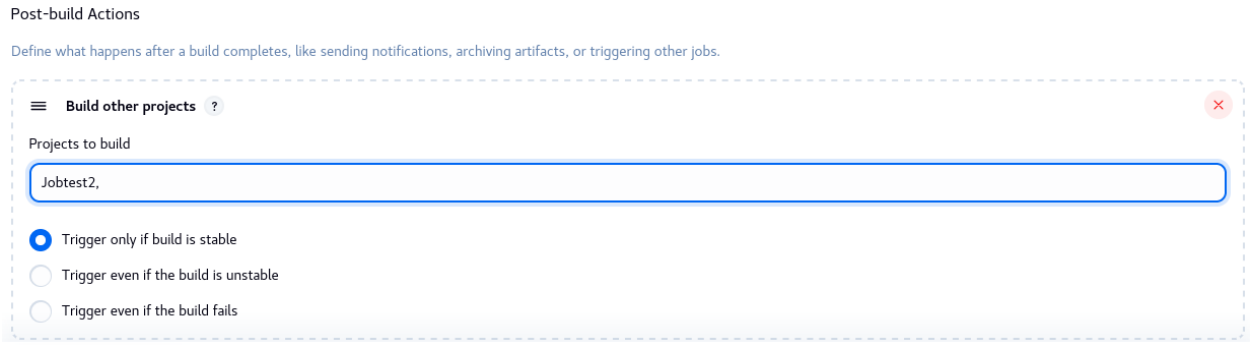
E-mail LinkedIn GitHub Website

- **Create new view for jobs**





- **Add post-build actions for job 1**



THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

- ## Add build triggers for job 2

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

Trigger builds remotely (e.g., from scripts) ?

Build after other projects are built ?

Projects to watch

Jobtest1,

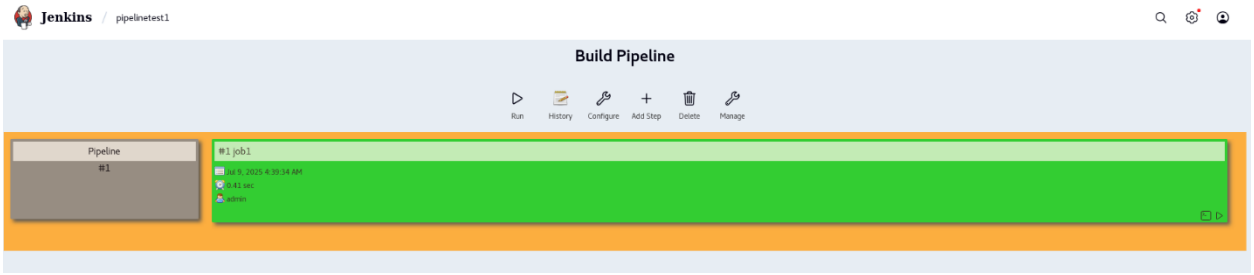
Trigger only if build is stable

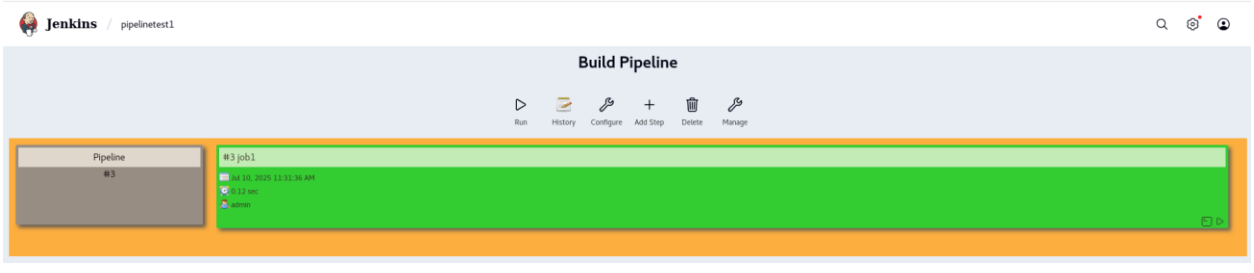
Trigger even if the build is unstable

Trigger even if the build fails

Always trigger, even if the build is aborted

- ## Build pipeline view





THISARA KANDAGE

UNDERGRADUATE - SLIIT

E-mail LinkedIn GitHub Website

Summary– What I Learned

- Installed and configured Jenkins on a RHEL-based system.
- Created and managed multiple freestyle jobs for automation tasks.
- Integrated GitHub repositories for continuous code fetching.
- Configured build triggers (poll SCM, webhook triggers).
- Defined and executed build steps and post-build actions.
- Installed and managed Jenkins plugins for enhanced functionality.
- Reviewed Jenkins console output logs for debugging and verification.
- Managed views for better job organization and monitoring.
- Gained experience with Jenkins dashboard, job configuration, and credentials handling.
- Understood key concepts of CI/CD pipelines and how Jenkins facilitates automation.