



Jenkins

Plugin Management

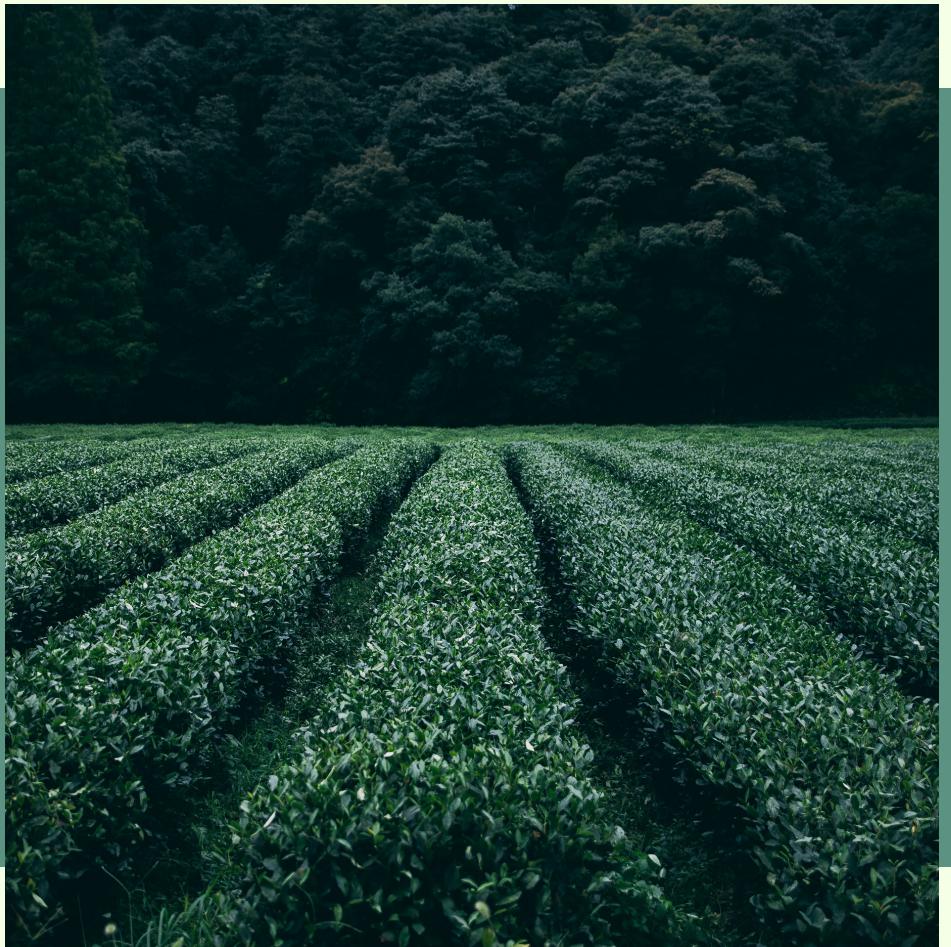
JENKINS

BY ALOK SRIVASTAVA - NETWORK NUTS

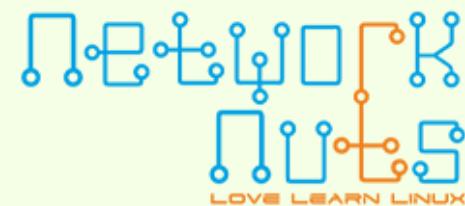
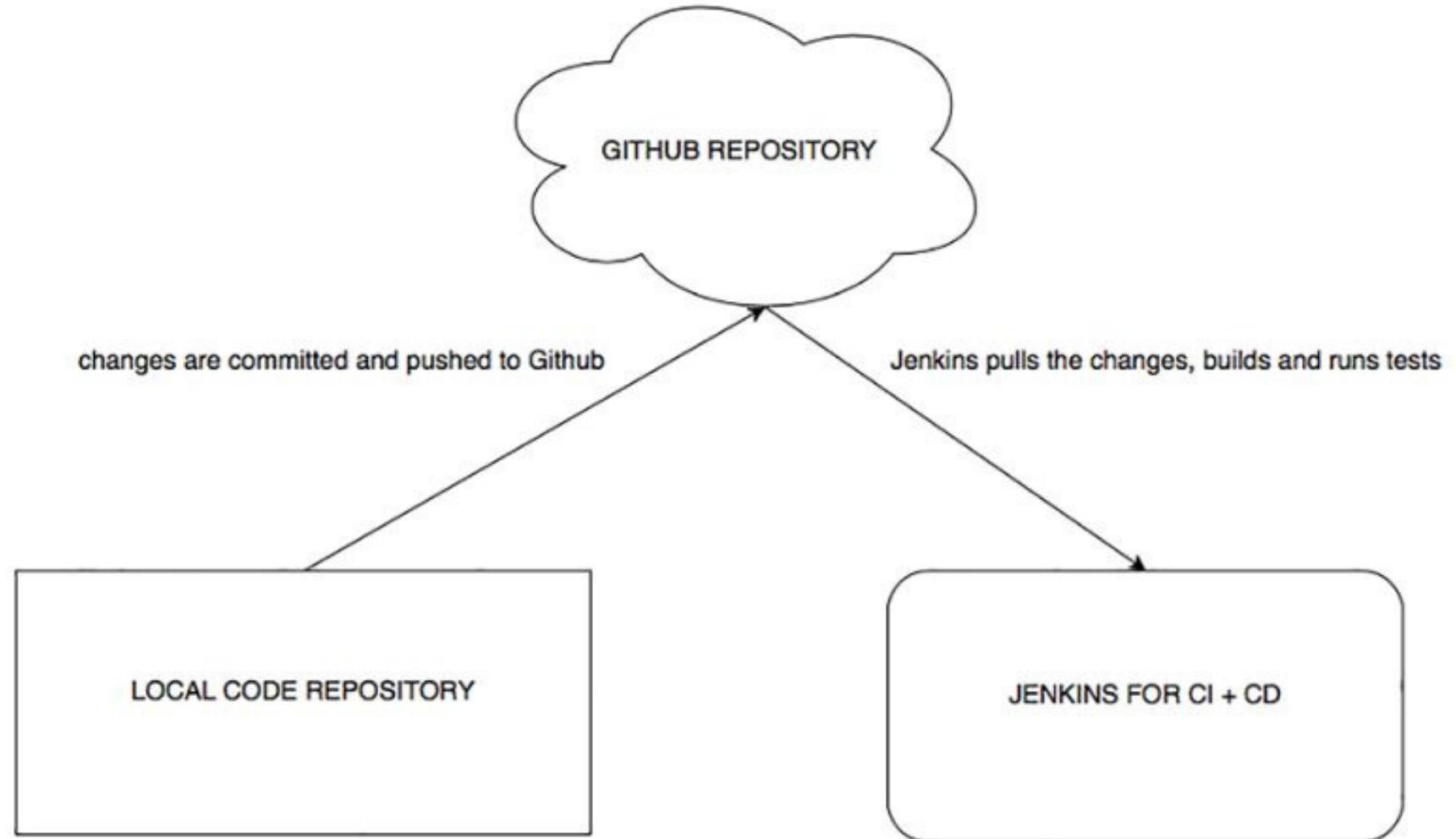


PLUGIN MANAGEMENT

As we build on our Continuous Delivery pipeline, we'll start with a simple CI build. This is to help us get used to the process. Moreover, a number of very crucial principles & guides will be covered, giving us a dynamic view of CI pipelines and builds. This knowledge is applicable to other tools & processes, too.



Here's a quick glance at what we want to achieve:



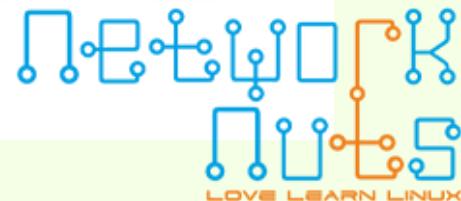
Workflow basically involves the following:

#1 - When code is pushed to GitHub, however little, we can pull the codebase from Jenkins, run the tests, and if they pass, the changes can be merged if a pull request was raised. If they fail, Jenkins will notify us through the notification channels that have been defined. To achieve this, we would normally require a few tools and services.

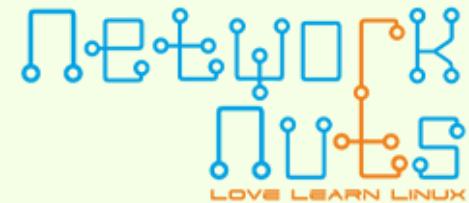
#2 - Use of Git, which is a tool that helps version code collaboratively. If you are running Git Bash, run the following command to confirm that Git is running as expected:

```
[root@jenkins ~]# git
usage: git [--version] [--help] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]
```

The most commonly used git commands are:



#3 - We also need Jenkins plugins. Plugins are resources that enable us to utilize services, including third-party applications, and achieve more operations on Jenkins.

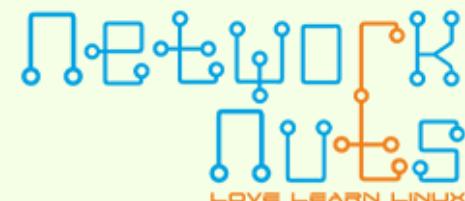


Jenkins Plugin Management

Plugin management is a skill and a requirement that's crucial for anyone willing to unleash Jenkins' full potential. On its own, the Jenkins server will not always fit everyone's requirements.

With plugins, Jenkins is able to become resourceful and efficient, still giving anyone a chance to add custom plugins to fit any project requirement. Plugins can achieve a lot regarding what people want to set up, for example:

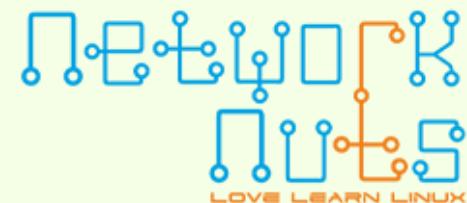
You want to send a notification after every build? There's a plugin for that.



Principles of Plugin Usage

Plugins in the Jenkins environment are used to improve the functionality and meet user requirements. Here are a few key points to remember:

- Always read through a plugin's documentation and guide to understand whether it achieves your intended goals.
- Before installing, check the usage statistics and update frequency.
- Will it work with your Jenkins server version?
- Understanding how it works will have a great impact on how much help the tool provides, allowing you to maximize the potential.



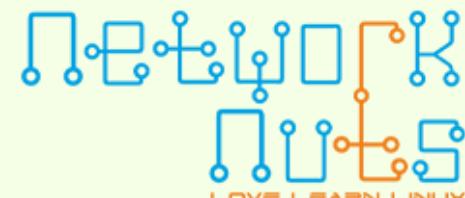
Few Types of Plugins in Jenkins

Administration Plugins

These plugins are related but not limited to the following:

Administration plugins will help customize access control and the overall management of the host.

- Service authentication: Introduction of new ways to access the host and its services, for example, LDAP.
- Audit trail and general security: Follow up on who did what; limit access to services and various operations
- Node and job-related management: Allows a variety of node-related operations, including the support of multiple operating system requirements.

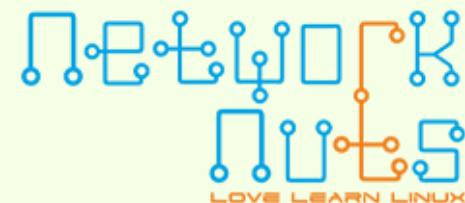


Few Types of Plugins in Jenkins

User Interface (UI) Plugins

UI plugins that help customize the Jenkins UI may provide the following:

- Customizing the view tabs, menu, and dropdowns
- Formatting text, and even images
- Email templates

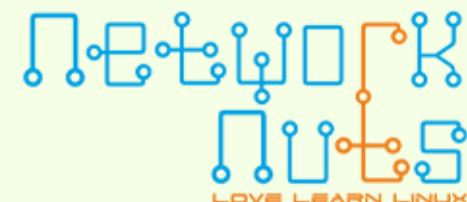


Few Types of Plugins in Jenkins

Source Code Management (SCM) Plugins

SCM plugins are what help integrate version control services. They provide the following:

- Allow Jenkins to run version control systems such as Git, Mercurial, and SCM.
- Allow Jenkins to pull code from version control hosts, such as GitHub, Bitbucket, GitLab, and so on.
- Authenticate Jenkins to pull from both private and public version control hosts.

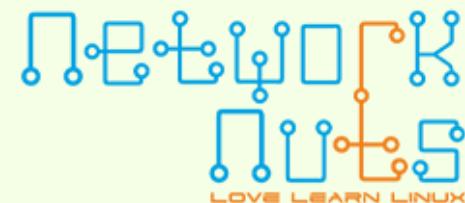


Few Types of Plugins in Jenkins

Build Management Plugins

Build Management are plugins that are involved in any build step. A point to consider here is that if you ever need a feature or process done that is not readily available on Jenkins, be sure to check the plugin index. Consider the following:

- Allow Jenkins to trigger notifications on build failure or pass
- Manage build artifacts
- Trigger deploys or other custom build steps



 **Manage Jenkins**

 **My Views**

 **Lockable Resources**

 **Credentials**

 **New View**

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle



Configure System

Configure global settings and paths.



Configure Global Security

Secure Jenkins; define who is allowed to access/use the system



Configure Credentials

Configure the credential providers and types



Global Tool Configuration

Configure tools, their locations and automatic installers.



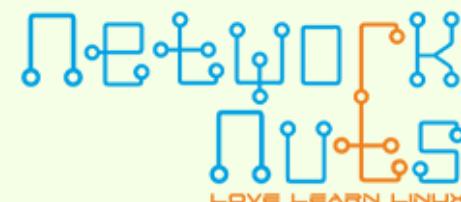
Reload Configuration from Disk

Discard all the loaded data in memory and reload everything from modified config files directly on disk.



Manage Plugins

Add, remove, disable or enable plugins that can extend the func





Updates	Available	Installed	Advanced
Install	Name ↓	Version	
			No updates

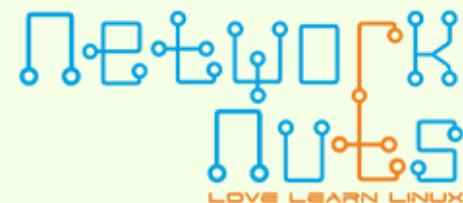
Update information obtained: 22 hr ago

[Check now](#)

Select: [All](#), [None](#)

This page lists updates to the plugins you currently use.

- Updates: Check for updates on currently installed plugins.
- Available: Check for available plugins directly from the plugin index.
- Installed: All currently installed plugins, both pre-and post-installation of the plugins.
- Advanced: The final tab allows us to add a custom plugin to Jenkins.



Filter: git

Updates

Available

Installed

Advanced

Install

Name

Version

[GitHub Authentication](#)

- GitHub Authentication plugin using GitHub OAuth to provide authentication and authorization capabilities for GitHub and GitHub Enterprise. 0.31

[Gitlab Authentication](#)

- This is the an authentication plugin using gitlab OAuth. 1.4

[Gitcolony Build Notification](#)

- This plugin updates live branch build status in [Gitcolony](#). 1.1

[GitHub Issues](#)

- This plugin creates GitHub issues when builds fail, and automatically closes the issue when the build starts passing again. 1.2.4

[Pipeline GitHub Notify Step](#)

- Plugin that provides a GitHub status notification step 1.0.4

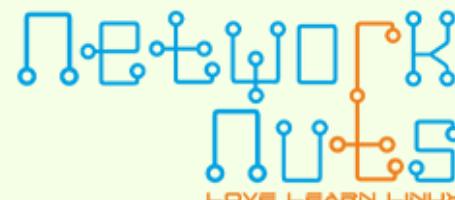
[Git Parameter](#)

- n o 1.0

Install without restart

Download now and install after restart

Update information obtained: 18 sec

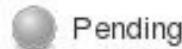


Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity

GitHub Authentication



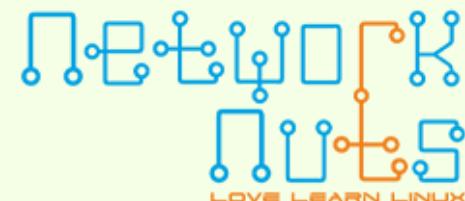
Pending

[Go back to the top page](#)

(you can start using the installed plugins right away)



Restart Jenkins when installation is complete and no jobs are running



Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

GitHub Authentication



Success

[Go back to the top page](#)

→ (you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

We are currently not running any jobs on our server, so it's OK to restart the host. Please don't do this on a production server, it might interfere with someone else's build or, a scheduled job.

