

Jenkins Pipeline

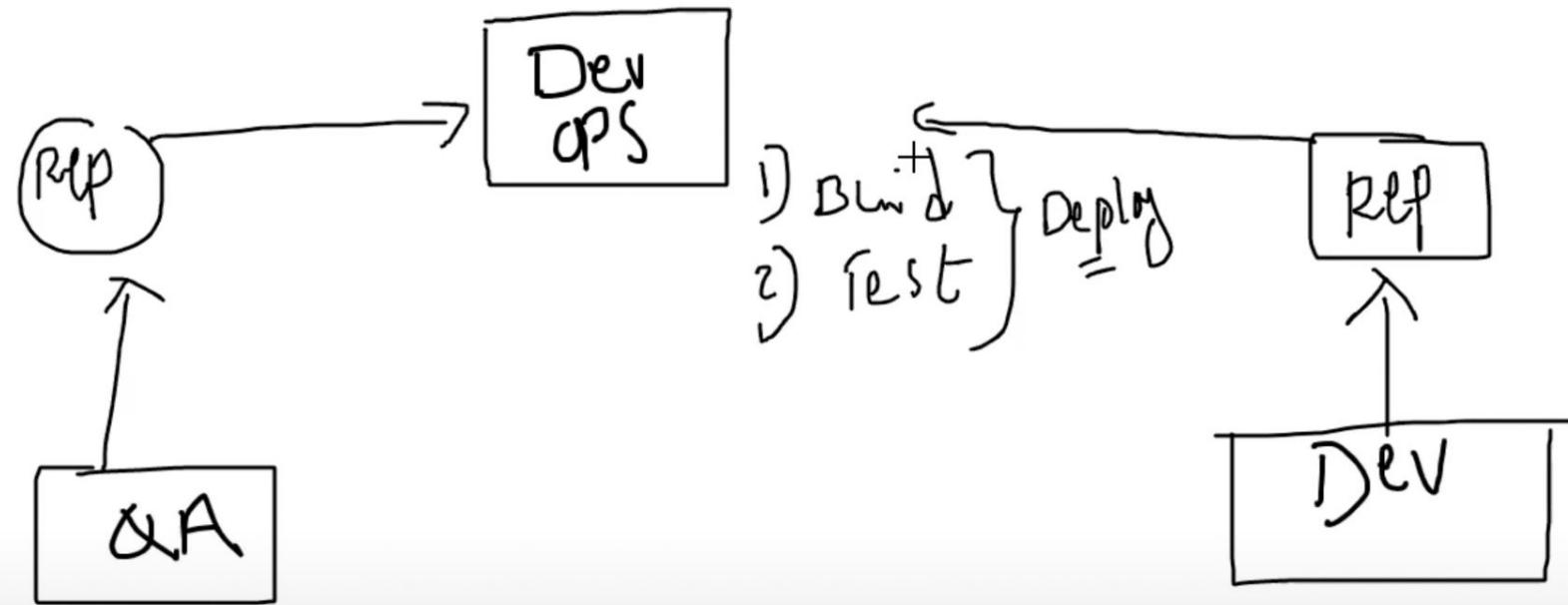
Overview on Continuous Integration (CI) & Continuous Deployment (CD)

What is Jenkins Pipeline

Jenkins Pipeline Jobs

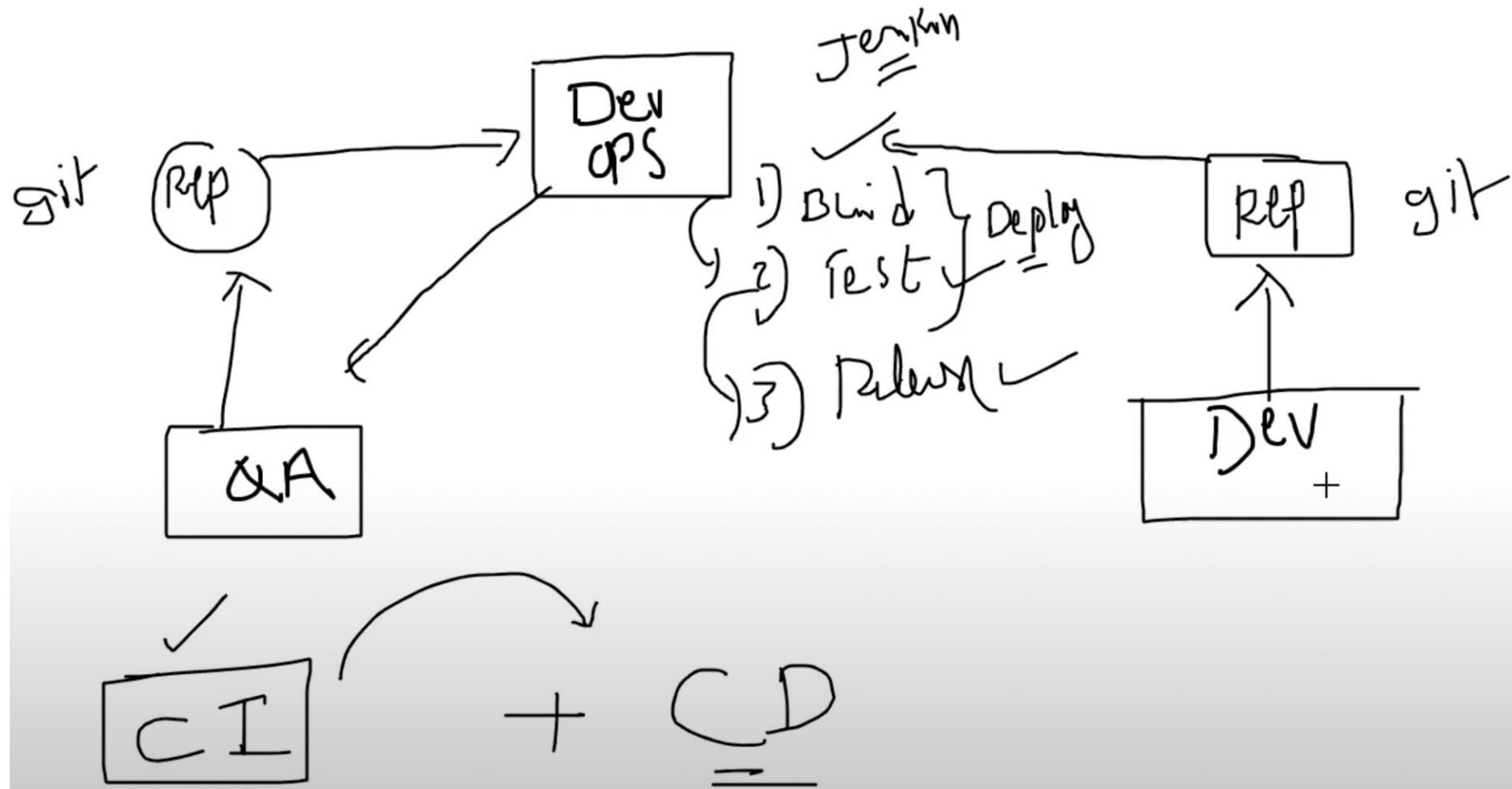
How to Create Pipeline in Jenkins

How to View Pipeline

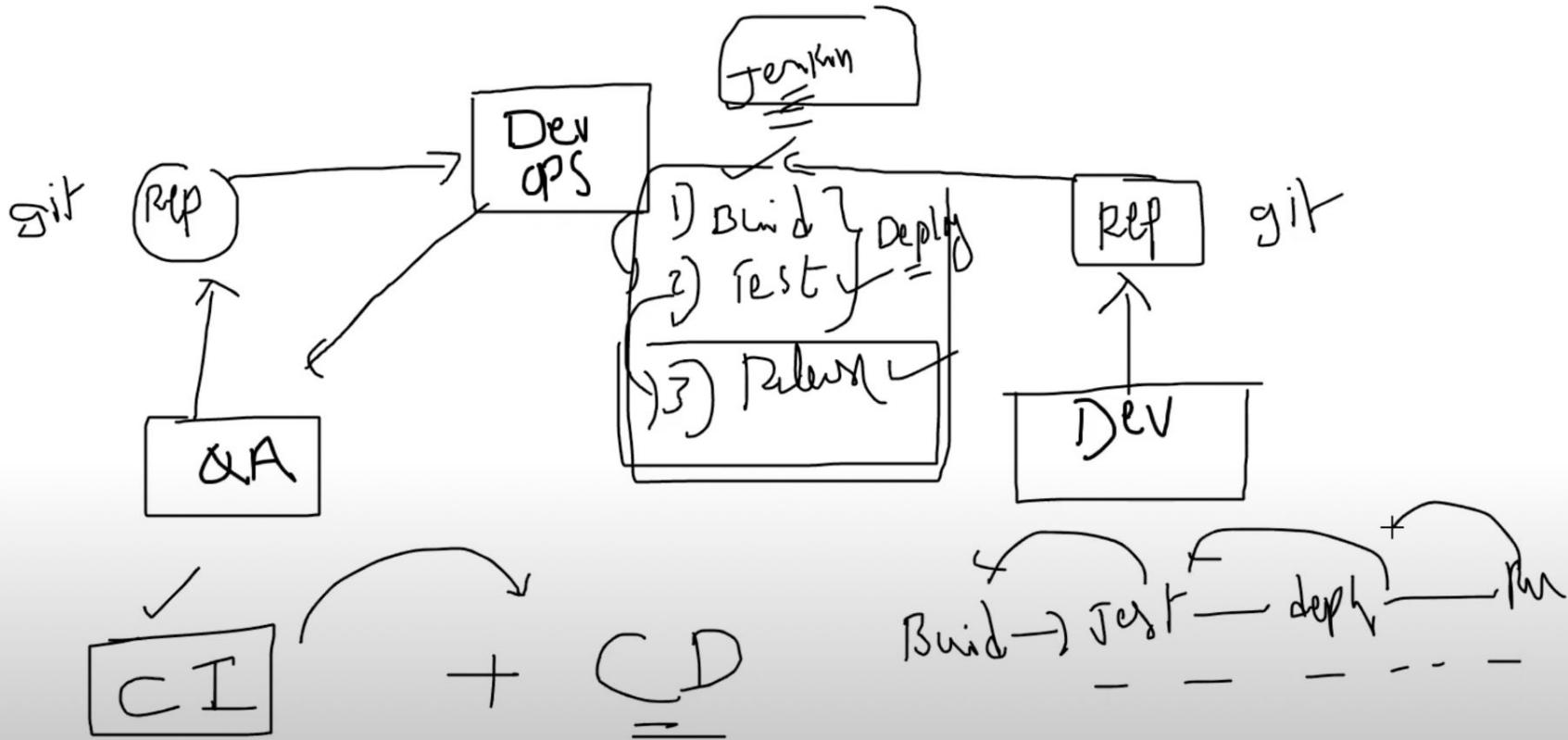


Continuous Integration

1. Developing the code [developers give the programs at the end of the day]
2. Build and test will happen next day build will be ready
3. Build will be ready and deployment will happen to qa environment
4. This process continuously called continuous integration
5. We will get the code from common repository



1. To do above task they use the certain tool called jenkins
2. Build will include certain activity include lot of jobs
3. Test will be include certain number of jobs
4. Deployment will include certain number of jenkins jobs
5. Release will include certain number of jobs
6. Using jenkins we can automate CI and CD Process

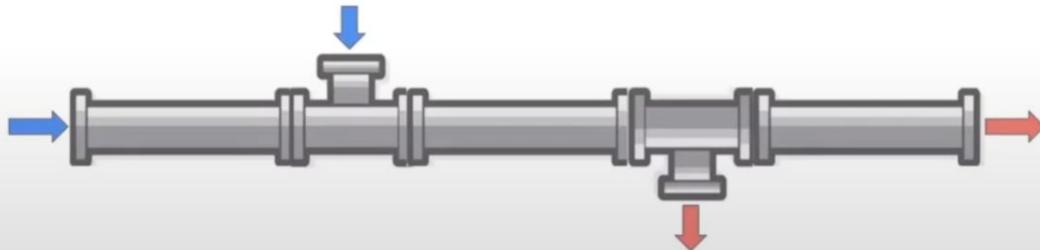


What is Pipeline?

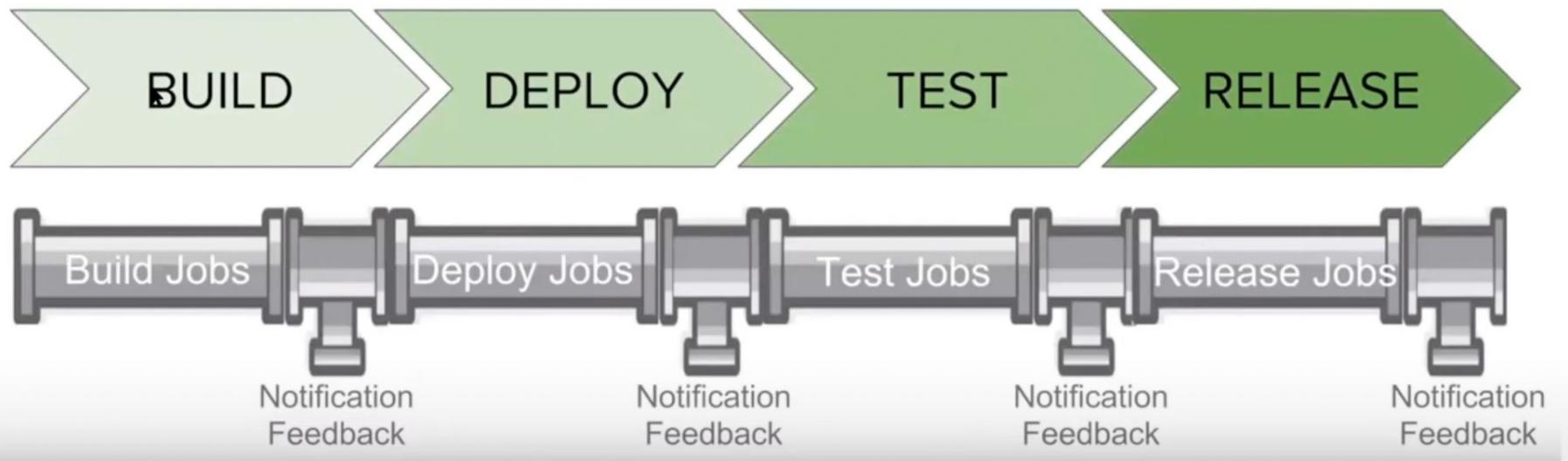
- In Jenkins, a pipeline is a group of events or jobs which are interlinked with one another in a sequence.



Pipe

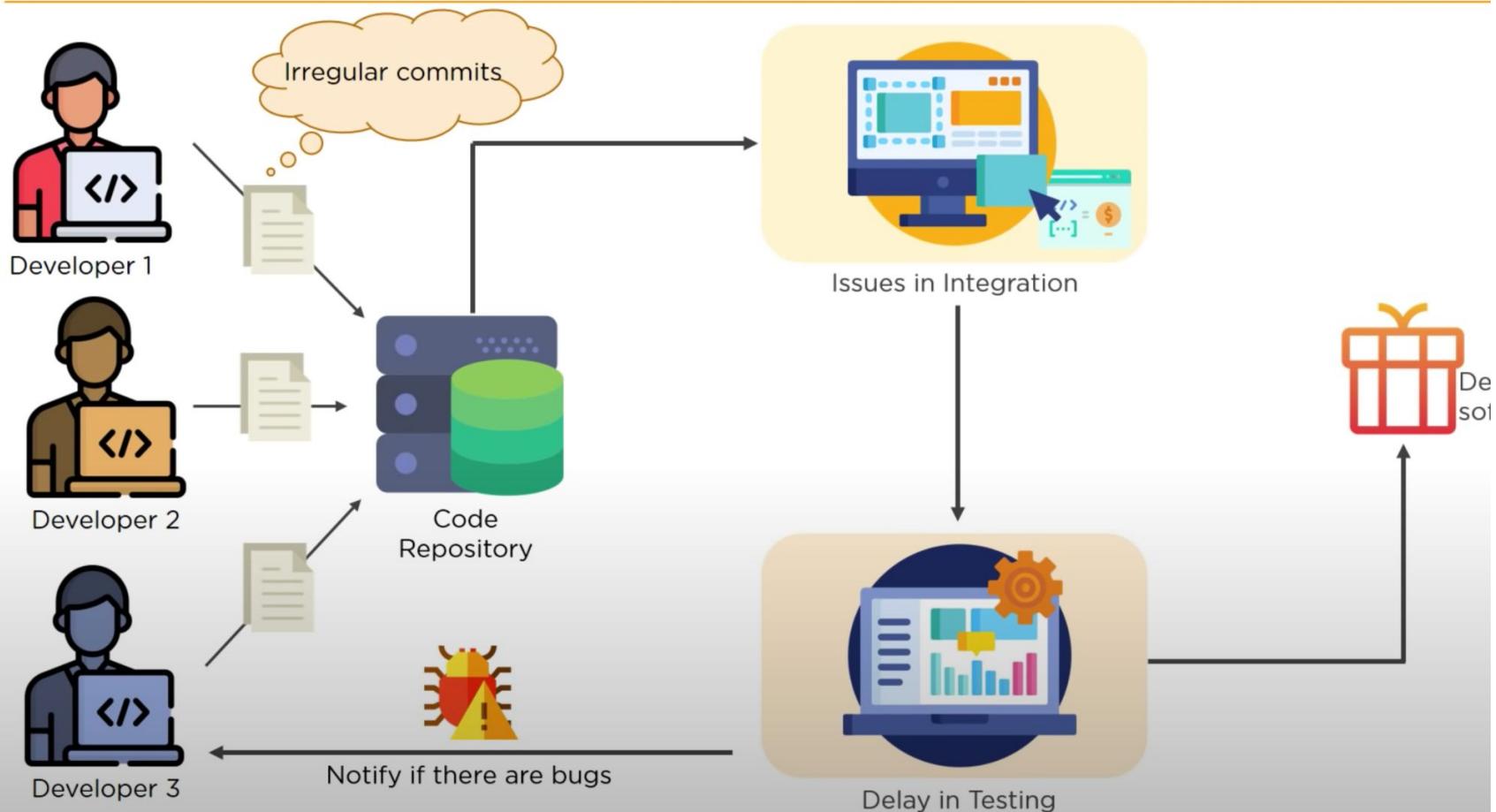


Pipeline



1. From the development they will get the code
2. Deployment
3. Test
4. Release-will release software to customers

Before Jenkins

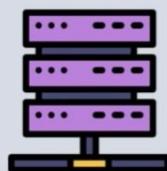


What is Jenkins?

Jenkins is a Continuous Integration tool that allows continuous development, test and deployment of newly created codes



All the codes will be pulled only at night



All the changes made to the code are build together



Commit changes to the source code

Nightly build and integration



Code is pulled whenever there is a commit made to the source code

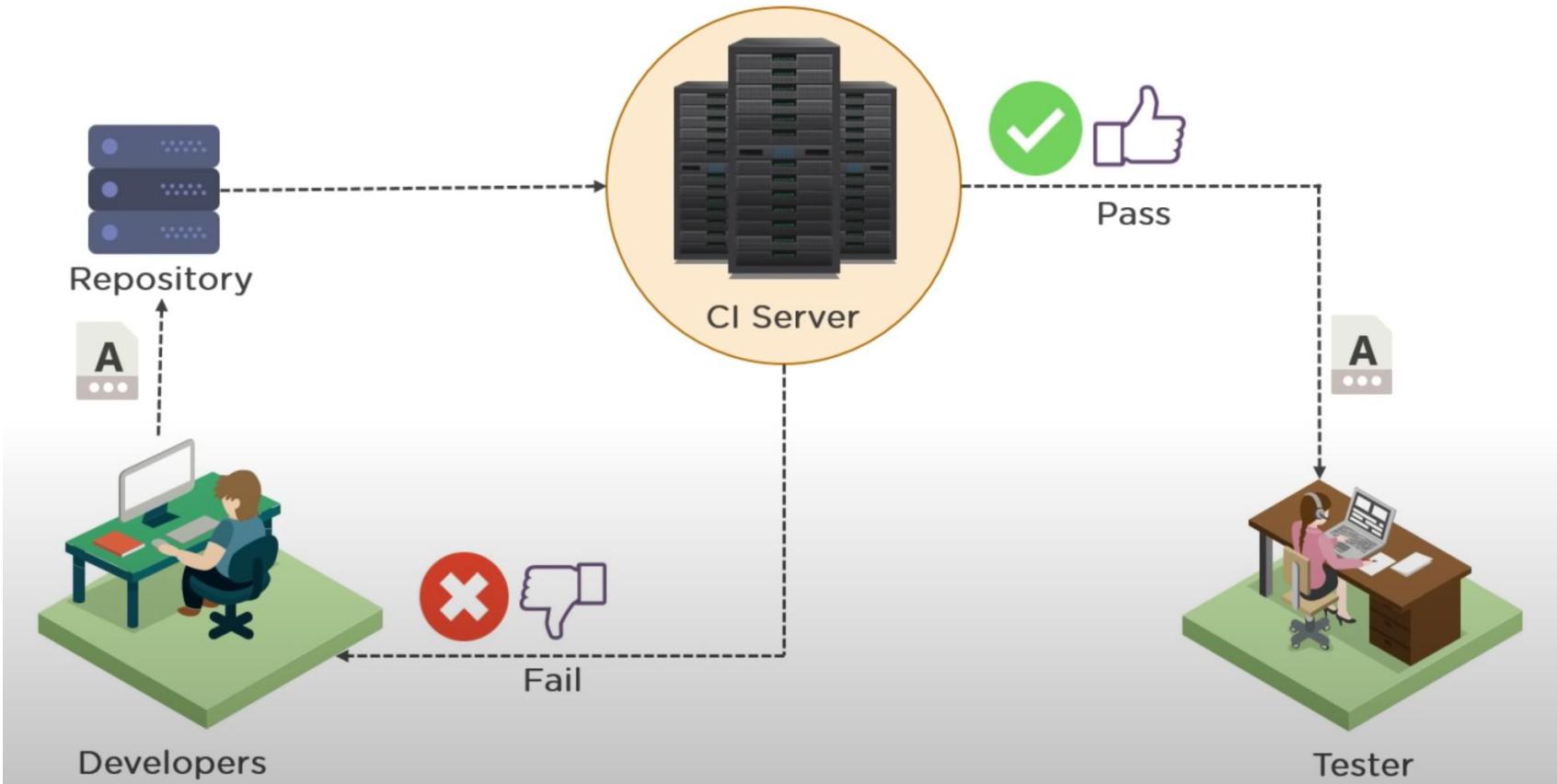


All the changes made to the source code is build continuously

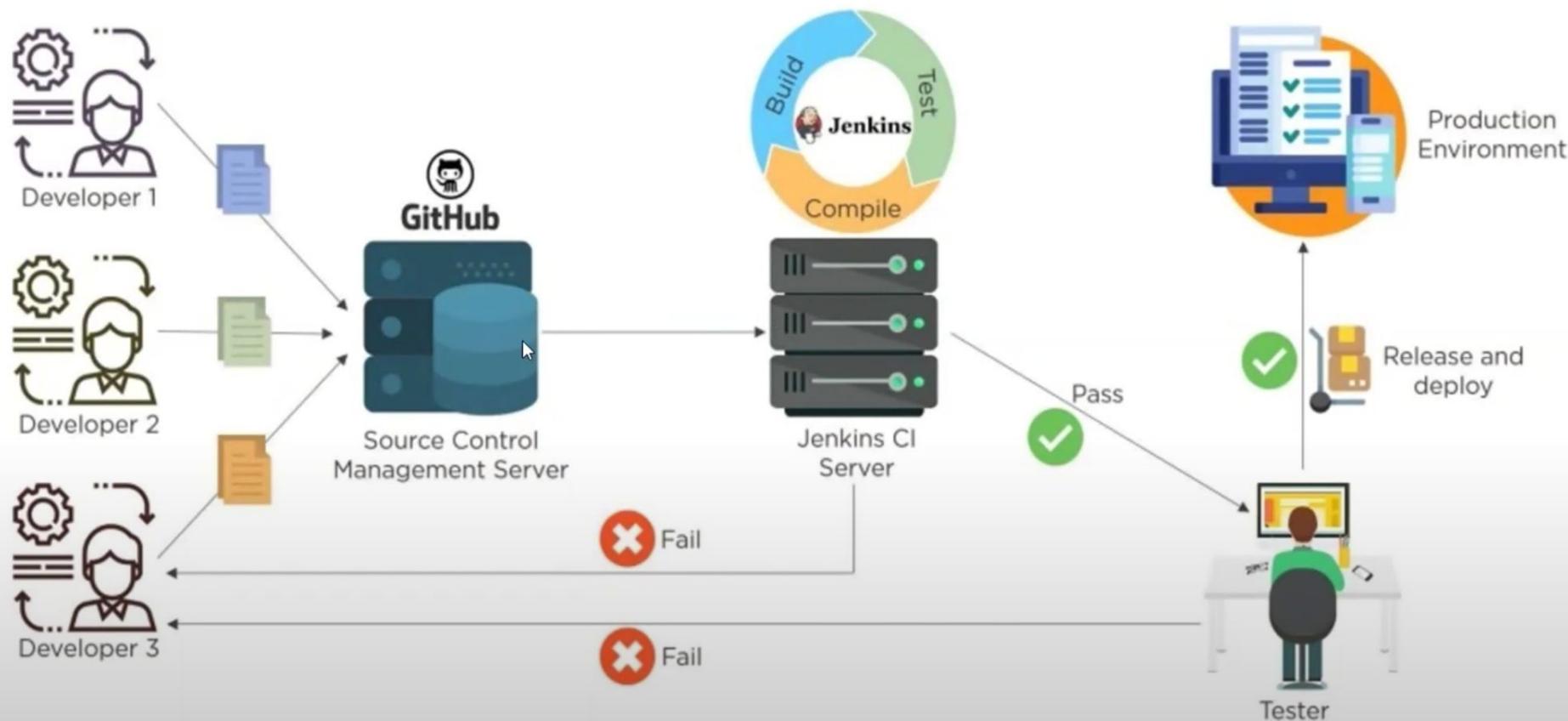


Continuous build and Integration

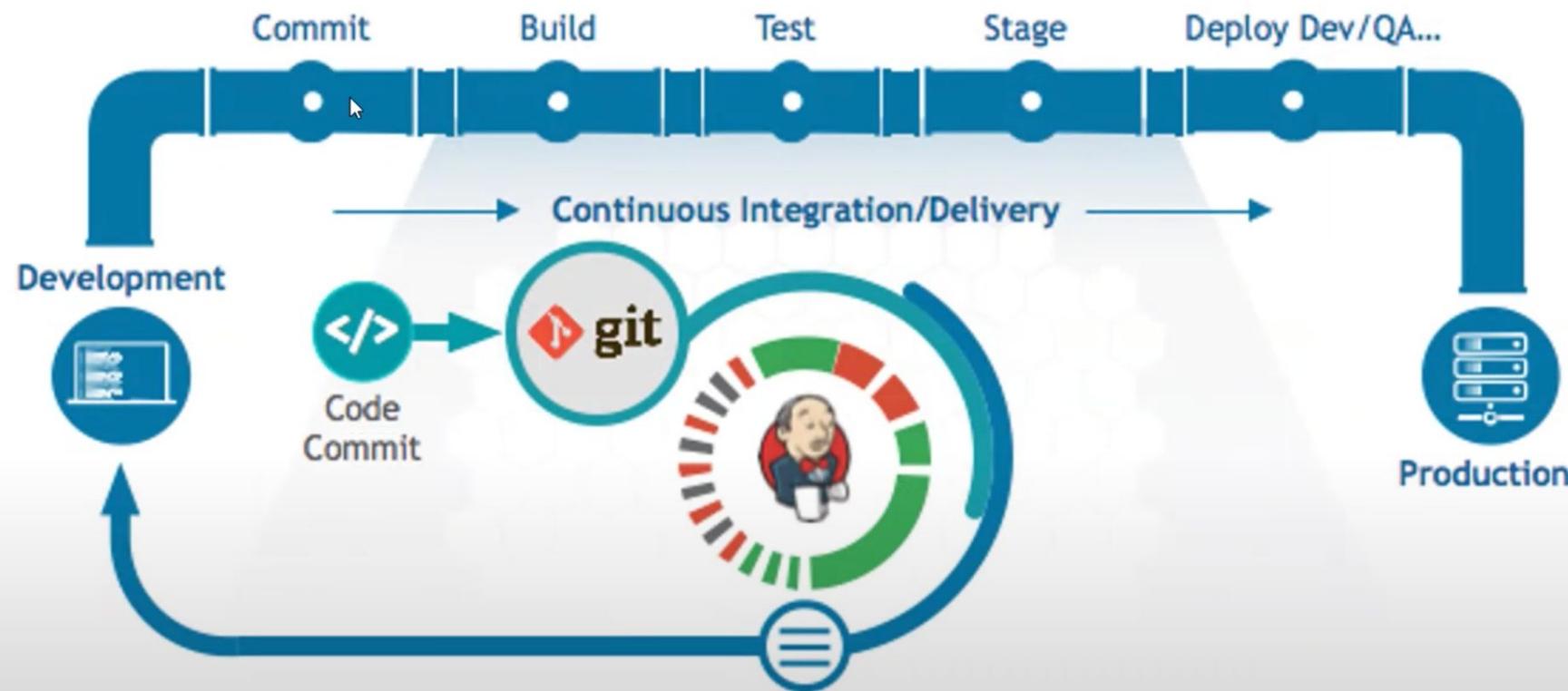
What is Continuous Integration?



CI & CD



Jenkins Pipeline



How many Ways we can create Pipeline

- We can Create Jenkins Pipeline in 2 Ways
- 1) Using **Build And Delivery Pipeline Plugins**
- 2) Using **Groovy Script on the Fly**(Here we use Jenkins file)
 - Scripted
 - Declarative

We will get one job in each and every section

Enter an item name
SampleBuildJob » Required field

 **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

 **Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

 **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **External Job**
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

 **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

 **Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

 **Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

If or to create a new item from other existing, you can use this option:

- GitHub Branches
- GitHub Pull Requests
- GitHub hook trigger for GITScm polling
- Poll SCM



Build Environment

- Create Delivery Pipeline version
- With Ant



Build

Execute shell

Command

```
<input type="text" value="" style="width: 100%; height: 150px; border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"/>

See the list of available environment variables


```



- Build after other projects are built 
- Build periodically 
- GitHub Branches 
- GitHub Pull Requests 
- GitHub hook trigger for GITScm polling 
- Poll SCM 

Build Environment

- Create Delivery Pipeline version 
- With Ant 

Build

 Execute shell X 

Command `date
echo "Build process successfully done"`

See [the list of available environment variables](#)

Advanced...

Subversion



Build Triggers

- Build after other projects are built
- Build periodically
- GitHub Branches
- GitHub Pull Requests
- GitHub hook trigger for GITScm polling
- Poll SCM



Build Environment

- Create Delivery Pipeline version
- With Ant



Build

Execute shell



Command

```
date  
echo "Deploy job Successfully executed...."
```

See [the list of available environment variables](#)

Advanced...

Subversion

Build Triggers

- Build after other projects are built
- Build periodically
- GitHub Branches
- GitHub Pull Requests
- GitHub hook trigger for GITScm polling
- Poll SCM



Build Environment

- Create Delivery Pipeline version
- With Ant



Build

Execute shell

Command
`date
echo "Test| job Successfully executed....."`



See [the list of available environment variables](#)

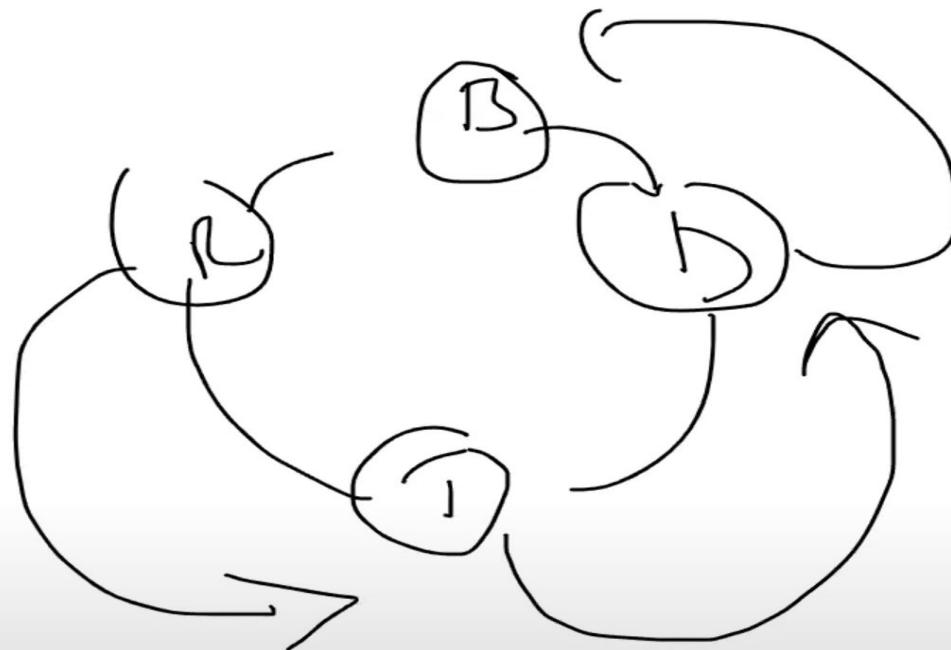
Advanced...

Build Queue						
		Name ↓	Last Success	Last Failure	Last Duration	
		CucumberProject	1 mo 9 days - #1	N/A	4 min 32 sec	
		Docker_Selenium_Grid_Project	3 days 2 hr - #3	3 days 2 hr - #2	1 min 20 sec	
		nopCommerce_maven_project	4 mo 20 days - #1	N/A	1 min 17 sec	
		nopCommerce_V001_Cucumber	1 mo 6 days - #1	N/A	3 min 24 sec	
		nopCommerce_V002_Github	N/A	1 mo 22 days - #2	13 sec	
		nopcommerce_v002_MavenProject_June_20	1 mo 22 days - #21	1 mo 22 days - #23	50 sec	
		nopCommerceV002_FreeStyleProjec_20_June	1 mo 23 days - #2	N/A	55 sec	
		nopCommerceV1_python_github	N/A	4 mo 14 days - #1	2.2 sec	
		phonselenium	4 mo 14 days - #2	N/A	49 sec	
		RestAssyred	N/A	3 mo 0 days - #1	34 sec	
		SampleBuildJob	N/A	N/A	N/A	
		SampleDeployJob ▾	N/A	N/A	N/A	
		SampleReleaseJob	N/A	N/A	N/A	
		SampleTestJob	N/A	N/A	N/A	

Icon:

Legend RSS for all RSS for failures RSS for just latest builds

Upstream and Downstream Job



General

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

 Disable this project Execute concurrent builds if necessary[Advanced...](#)**Source Code Management** None Git Subversion**Build Triggers** Build after other projects are builtProjects to watch **No such project 'Sampl'. Did you mean 'SampleTestJob'?** Trigger only if build is stable Trigger even if the build is unstable Trigger even if the build fails Build periodically GitHub Branches GitHub Pull Requests GitHub hook trigger for GITScm polling Poll SCM**Build Environment** Create Delivery Pipeline version

With Ant

**Save****Apply**

 Back to Dashboard

 Status

 Changes

 Workspace

 Build Now

 Delete Project

 Configure

 Rename

 Build History trend =

X

 #1 X
(pending—Finished waiting)

 RSS for all  RSS for failures

Project SampleDeployJob



[Workspace](#)



[Recent Changes](#)

Upstream Projects

 [SampleBuildJob](#)

Downstream Projects

 [SampleTestJob](#)

Permalinks



How to setup BUILD PIPELINE in Jenkins

Step1: Chain required jobs in sequence Add upstream/downstream jobs

Step2: Install *Build Pipeline Plugin*

Step3: Add

 Build Pipeline View

 Configure the view Step

Step4: Run and Validate

Enabled	Name ↓	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Ant Adds Apache Ant support to Jenkins	1.7		Uninstall
<input checked="" type="checkbox"/>	Apache HttpComponents Client 4.x API Plugin Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.	4.5.5-3.0	Downgrade to 4.5.3-2.1	Uninstall
<input checked="" type="checkbox"/>	bouncycastle API This plugin provides an stable API to Bouncy Castle related tasks.	2.16.2		Uninstall
<input checked="" type="checkbox"/>	Branch API This plugin provides an API for multiple branch based projects.	2.5.2		Uninstall
<input checked="" type="checkbox"/>	Build Pipeline Plugin This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.	1.5.8		Uninstall
<input checked="" type="checkbox"/>	Command Agent Launcher Allows agents to be launched using a specified command.	1.2		Uninstall
<input checked="" type="checkbox"/>	Conditional BuildStep A buildstep wrapping any number of other buildsteps, controlling their execution based on a defined condition (e.g. BuildParameter).	1.3.6		Uninstall
<input checked="" type="checkbox"/>	Credentials This plugin allows you to store credentials in Jenkins.	2.1.18	Downgrade to 2.1.16	Uninstall
<input checked="" type="checkbox"/>	Delivery Pipeline Plugin This plugin visualize Delivery Pipelines (Jobs with upstream/downstream dependencies)	1.4.0		Uninstall
<input checked="" type="checkbox"/>	Display URL API Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.2.0		Uninstall
<input checked="" type="checkbox"/>	Durable Task Plugin Library offering an extension point for processes which can run outside of Jenkins yet be monitored.	1.30		Uninstall
<input checked="" type="checkbox"/>	Email Extension Plugin This plugin is a replacement for Jenkins's email publisher. It allows to configure every aspect of email notifications: when an email is sent, who should receive it and what the email says.	2.66	Downgrade to 2.66	Uninstall

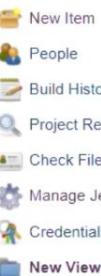
	Updates	Available	Installed	Advanced			
Enabled			Name ↓		Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>			Ant	Adds Apache Ant support to Jenkins	1.7		Uninstall
<input checked="" type="checkbox"/>			Apache HttpComponents Client 4.x API Plugin	Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.	4.5.5-3.0	Downgrade to 4.5.3-2.1	Uninstall
<input checked="" type="checkbox"/>			bouncycastle API	This plugin provides an stable API to Bouncy Castle related tasks.	2.16.2		Uninstall
<input checked="" type="checkbox"/>			Branch API	This plugin provides an API for multiple branch based projects.	2.5.2		Uninstall
<input checked="" type="checkbox"/>			Build Pipeline Plugin	This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.	1.5.8		Uninstall
<input checked="" type="checkbox"/>			Command Agent Launcher	Allows agents to be launched using a specified command.	1.2		Uninstall
<input checked="" type="checkbox"/>			Conditional BuildStep	A buildstep wrapping any number of other buildsteps, controlling their execution based on a defined condition (e.g. BuildParameter).	1.3.6		Uninstall
<input checked="" type="checkbox"/>			Credentials	This plugin allows you to store credentials in Jenkins.	2.1.18	Downgrade to 2.1.16	Uninstall
<input checked="" type="checkbox"/>			Delivery Pipeline Plugin	This plugin visualize Delivery Pipelines (Jobs with upstream/downstream dependencies)	1.4.0		Uninstall
<input checked="" type="checkbox"/>			Display URL API	Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.2.0		Uninstall
<input checked="" type="checkbox"/>			Durable Task Plugin	Library offering an extension point for processes which can run outside of Jenkins yet be monitored.	1.30		Uninstall
<input checked="" type="checkbox"/>			Email Extension Plugin	This plugin is a replacement for Jenkins's email publisher. It allows to configure every aspect of email notifications: when an email is sent,	2.66	Downgrade to 2.66	Uninstall

Enabled	Name ↓	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Ant Adds Apache Ant support to Jenkins	1.7		Uninstall
<input checked="" type="checkbox"/>	Apache HttpComponents Client 4.x API Plugin Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.	4.5.5-3.0	Downgrade to 4.5.3-2.1	Uninstall
<input checked="" type="checkbox"/>	bouncycastle API This plugin provides an stable API to Bouncy Castle related tasks.	2.16.2		Uninstall
<input checked="" type="checkbox"/>	Branch API This plugin provides an API for multiple branch based projects.	2.5.2		Uninstall
<input checked="" type="checkbox"/>	Build Pipeline Plugin This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.	1.5.8		Uninstall
<input checked="" type="checkbox"/>	Command Agent Launcher Allows agents to be launched using a specified command.	1.2		Uninstall
<input checked="" type="checkbox"/>	Conditional BuildStep A buildstep wrapping any number of other buildsteps, controlling their execution based on a defined condition (e.g. BuildParameter).	1.3.6		Uninstall
<input checked="" type="checkbox"/>	Credentials This plugin allows you to store credentials in Jenkins.	2.1.18	Downgrade to 2.1.16	Uninstall
<input checked="" type="checkbox"/>	Delivery Pipeline Plugin This plugin visualize Delivery Pipelines (Jobs with upstream/downstream dependencies)	1.4.0		Uninstall
<input checked="" type="checkbox"/>	Display URL API Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.2.0		Uninstall
<input checked="" type="checkbox"/>	Durable Task Plugin Library offering an extension point for processes which can run outside of Jenkins yet be monitored.	1.30		Uninstall
<input checked="" type="checkbox"/>	Email Extension Plugin This plugin is a replacement for Jenkins's email publisher. It allows to configure every aspect of email notifications: when an email is sent, who should receive it and what the email says.	2.66	Downgrade to 2.66	Uninstall

What is the reason for creating multiple users?

In large organization there are multiple team to manage Jenkins, run jobs, monitor jobs and configure jobs. So to manage these without proper permissions and assigned roles would be very difficult in Jenkins.

Role Strategy Plugin help in assign roles to different users.



View name

Tel

Build Pipeline View

Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

Delivery Pipeline View

Continuous Delivery pipelines, perfect for visualization on information radiators. Shows one or more delivery pipeline instances, based on traditional Jenkins jobs with upstream/downstream dependencies.

Delivery Pipeline View for Jenkins Pipelines

Continuous Delivery pipelines, perfect for visualization on information radiators. Shows one or more delivery pipeline instances, based on Jenkins pipelines (created using the Pipeline or Workflow plugin).

List View

Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

OK

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Queue

Jobs in the queue.

Executor Status**Layout****Based on upstream/downstream relationship**

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream config

Select Initial Job

SampleBuildJob

Trigger Options**Build Cards****Standard build card**

Use the default build cards

Restrict triggers to most recent successful builds Yes NoAlways allow manual trigger on pipeline steps Yes No**Display Options****No Of Displayed Builds**

1
1
2
3
5
10
20
50
100
200
500

Row Headers**Column Headers****Refresh frequency (in seconds)****URL for custom CSS files****Console Output Link Style**

Lightbox

OK**Apply**

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

**Manage Nodes and Clouds**

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

**Configure Global Security**

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

**Manage Credentials**

Configure credentials

**Configure Credential Providers**

Configure the credential providers and types

**Manage Users**

Create/delete/modify users that can log in to this Jenkins

**In-process Script Approval**

Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions.

 Back to Dashboard Manage Jenkins Create User

Create User

Username:

Password:

kkjava

Confirm password:

user

Full name:

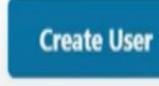
blah

E-mail address:

foo

kishan.javatrainer@gmail.com

kishan.cs2018@gmail.com

 Create User

None

Authorization

Strategy

Authorization

Anyone can do anything

?

Legacy mode

?

Logged-in users can do anything

?

Matrix-based security

?

Project-based Matrix Authorization Strategy

?

Role-Based Strategy

?



Reload Configuration from Disk

Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.



Jenkins CLI

Access/manage Jenkins from your shell, or from your script.



Script Console

Executes arbitrary script for administration/troubleshooting/diagnostics.



Prepare for Shutdown

Stops executing new builds, so that the system can be eventually shut down safely.

Uncategorized



Manage and Assign Roles

Handle permissions by creating roles and assigning them to users/groups



Managed files

e.g. settings.xml for maven, central managed scripts, custom files, ...



Editable Email Notification Templates

Configure global templates for Editable Email Notification Plugin

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Lockable Resources
- New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle



Assign Roles

Global roles

User/group	admin
Raj Sharma	<input checked="" type="checkbox"/>
Anonymous	<input type="checkbox"/>

User/group to add

Add



Item roles

User/group
Anonymous

User/group to add

Add



» Required field**Freestyle project**

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**GitHub Organization**

Scans a GitHub organization (or user account) for all repositories matching some defined markers.

**Multibranch Pipeline**

Creates a set of Pipeline projects according to detected branches in one SCM repository.

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

Parameterized Trigger

[Build Tools](#) [Build Triggers](#)

The Jenkins Plugins Parent POM Project

2.41 4 mo 17 days ago

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

Handy Uri Templates 2.x API

[api-plugin](#)

2.1.8-1.0 2 yr 0 mo ago

Bundles Handy Uri Templates 2.x and allows it to be used by Jenkins plugins

JAXB

[api-plugin](#)

2.3.0.1 2 yr 7 mo ago

JAXB packaging for more transparent Java 9+ compatibility

Role-based Authorization Strategy

[Security](#) [Authentication and User Management](#)

3.2.0 2 mo 22 days ago

Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions.

Favorite

[Edit view columns](#) [Miscellaneous](#) [User Interface](#) [Authentication and User Management](#)

2.2.2 8 mo 7 days ago

<https://plugins.jenkins.io/role-strategy/>

Dashboard > Configure Global Security

Security Realm

- Delegate to servlet container
- Jenkins' own user database
 - Allow users to sign up
- LDAP
- None

Authorization

- Anyone can do anything
- Legacy mode
- Logged-in users can do anything
- Matrix-based security
- Project-based Matrix Authorization Strategy
- Role-Based Strategy

Markup Formatter

Markup Formatter

- Plain text

Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.

Agents

Users

These users can log into Jenkins. This is a sub set of [this list](#) which also contains auto-created users who **really** just made some commits on some **projects** and have no **direct** Jenkins access.

User ID	Name	
 admin2	admin2	
 dev	dev	
 tester 	tester	 

<https://www.javatpoint.com/create-and-manage-users-in-jenkins>

 John Doe
Project Manager

Manage Roles

Global roles

Role to add

tester

Add

Item roles

Role	Pattern	Credentials	Job	Run	SCM	Lockable Resources
		Create Delete ManageDomains Update View Build Cancel Configure	Create Delete Discover Move Read Workspace Delete Replay Update Tag Reserve Unlock View			

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Repository URL ?

Credentials ?

- none - Add ▾

Advanced...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') X ?

Add Branch

Repository browser ?

Additional Behaviours

Add ▾

Build Triggers



(Auto)



Additional Behaviours

 Check out to a sub-directoryLocal subdirectory for repo [?](#)

D:\Learning\localgitrepo

[Add ▾](#)

Build Triggers

- Trigger builds remotely (e.g., from scripts) [?](#)
- Build after other projects are built [?](#)
- Build periodically [?](#)
- GitHub hook trigger for GITScm polling [?](#)
- Poll SCM [?](#)

Build Environment

- Delete workspace before build starts [?](#)
- Use secret text(s) or file(s) [?](#)
- Abort the build if it's stuck [?](#)
- Add timestamps to the Console Output [?](#)
- Inspect build log for published Gradle build scans [?](#)
- With Ant [?](#)

Build



Manage Roles

Global roles

Role	Overall				Credentials				Agent								Job								Run		View	
	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	Configure	Create
admin	<input checked="" type="checkbox"/>																											
developer	<input type="checkbox"/>																											
tester	<input type="checkbox"/>																											
read user	<input type="checkbox"/>																											

Role to add

read user

Add

Item roles

Role Pattern	Credentials	Job	Run	SCM	Lockable Resources
--------------	-------------	-----	-----	-----	--------------------



Assign Roles

Global roles

User/group	admin	developer	read user	tester	
admin2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dev	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
tester	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Role : developer
User : dev

User/group to add

read



Add

Item roles

User/group
 Anonymous

User/group to add



Add

Node roles

User/group
 Anonymous

[Plain text] [Preview](#)

- Discard old builds ?
- GitHub project ?
- This build requires lockable resources ?
- This project is parameterised ?
- Throttle builds ?
- Disable this project ?
- Execute concurrent builds if necessary ?

[Advanced...](#)

Source Code Management

None

Git ?

Repositories ?

Repository URL

✖ Please enter Git repository.

Credentials

- NONE -

[+ Add +](#)

[Advanced...](#)

[Add Repository](#)

Build Jenkins Job Automatically on GitHub Commit

Poll SCM - periodically polls the SCM to check whether changes were made (i.e. new commits) and builds the project if new commits were pushed since the last build.

Build Periodically - whereas build periodically builds the project periodically even if nothing has changed.

Build Triggers - Poll SCM



* * * * *

Use default maven global settings

JDK

JDK installations

Add JDK

List of JDK installations on this system

Git

Git installations



Name

Default

Path to Git executable [?](#)

git.exe

There's no such executable git.exe in PATH: C:/Program Files/Microsoft/jdk-11.0.12.7-hotspot/bin, C:/Windows/system32, C:/Windows, C:/Windows/System32/Wbem, C:/Windows/System32/WindowsPowerShell/v1.0/, C:/Windows/System32/OpenSSH/, C:/Program Files/nodejs/, C:/Program Files/PuTTY/, C:/Users/india/AppData/Local/Microsoft/WindowsApps, C:/Users/india/AppData/Local/Programs/Microsoft VS Code/bin, C:/Users/india/AppData/Roaming/npm, C:/Program Files/heroku/bin.

Install automatically [?](#)

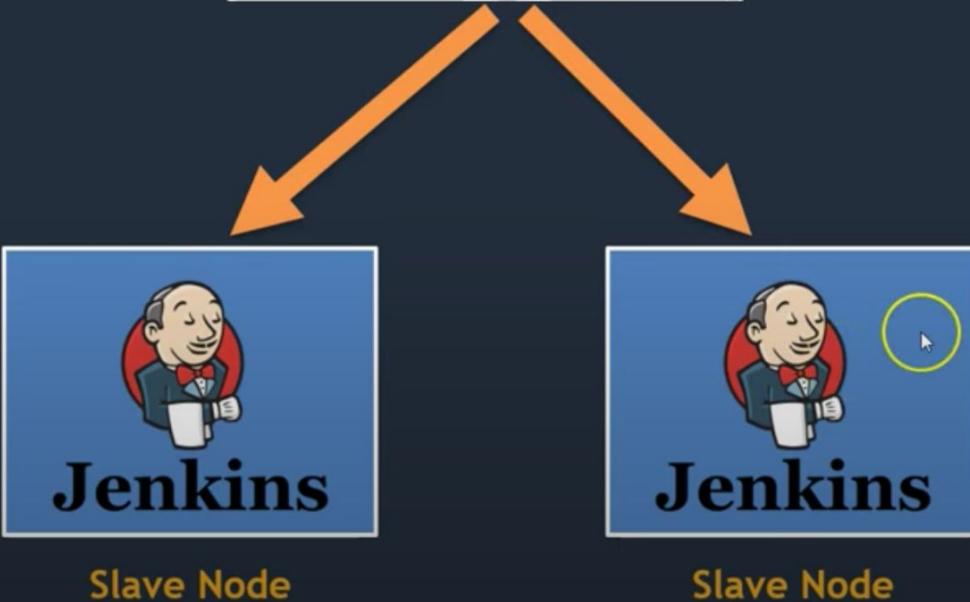
Delete Git

Automatic builds/

<https://netcorecloud.com/tutorials/install-and-configure-email-notifications-in-jenkins/>

<https://netcorecloud.com/tutorials/install-and-configure-email-notifications-in-jenkins/>

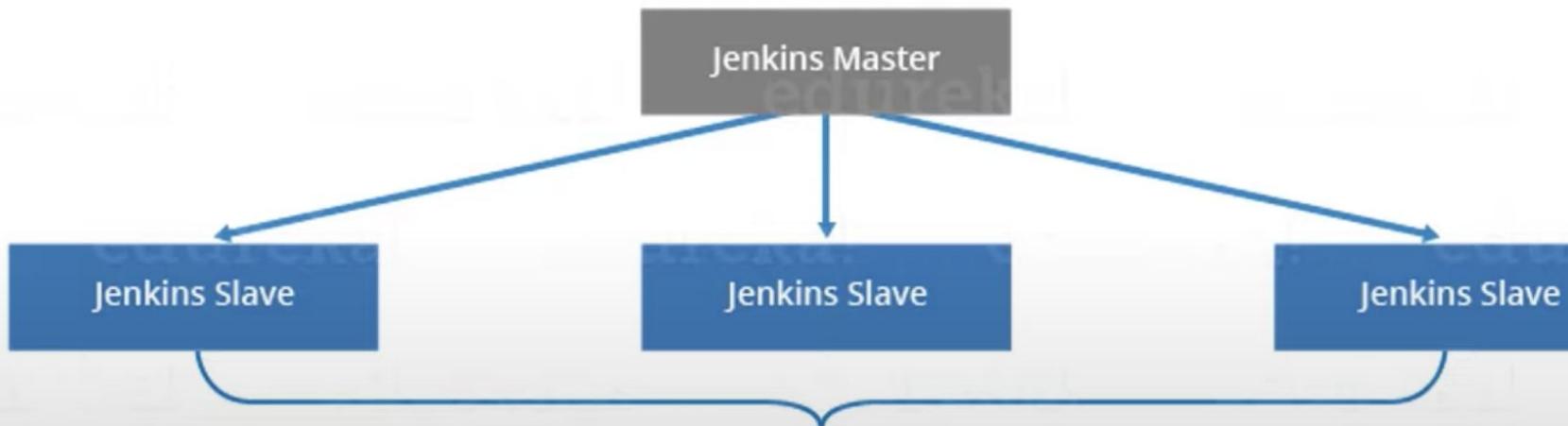
Jenkins Master



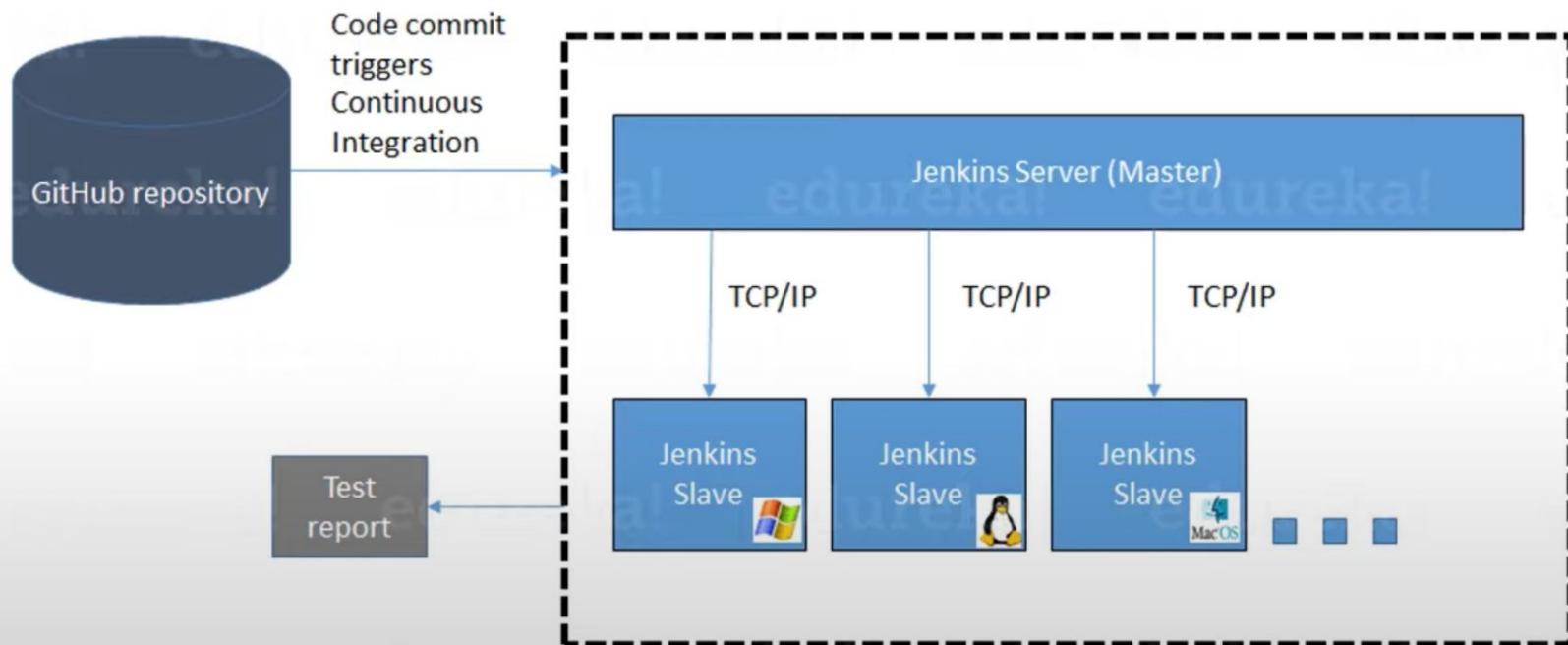
Slave Node

Slave Node

Jenkins Master will distribute its workload to the Slaves



Jenkins Slaves are generally required to provide the desired environment.
It works on the basis of requests received from Jenkins Master.



@gmail.com

Use SMTP Authentication [?](#)

User Name

shanswadi@gmail.com

Password

.....

Use SSL [?](#)

Use TLS

SMTP Port [?](#)

465

Reply-To Address

shanswadi@gmail.com

SMTP Port [?](#)

465

Reply-To Address

shanswadi@gmail.com

Charset

UTF-8

Test configuration by sending test e-mail

Test e-mail recipient

shanswadi@gmail.com

Email was successfully sent

[Test configuration](#)

Build Triggers

- Build after other projects are built [?](#)
- Build periodically [?](#)
- GitHub Branches
- GitHub Pull Requests [?](#)
- GitHub hook trigger for GITScm polling [?](#)
- Poll SCM [?](#)

Schedule [?](#)

```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Friday, February 10, 2023 at 1:03:25 PM Coordinated Universal Time; would next run at Friday, February 10, 2023 at 1:03:25 PM Coordinated Universal Time.

Configure

General

 Source Code Management

 Build Triggers

 Build Environment

 Build Steps

 Post-build Actions

Throttle builds 

Execute concurrent builds if necessary 

[Advanced...](#)

Source Code Management

 None

 Git 

Build Triggers

Build after other projects are built 

Build periodically 

Schedule 

 Do you really mean "every minute" when you say "*****"? Perhaps you meant "H *****" to poll once per hour

 Jenkins[Jenkins](#) > [Nodes](#) >[Back to Dashboard](#)[Manage Jenkins](#)[New Node](#)[Configure Clouds](#)[Node Monitoring](#)[Build Queue](#)

No builds in the queue.

[Build Executor Status](#)

1 Idle

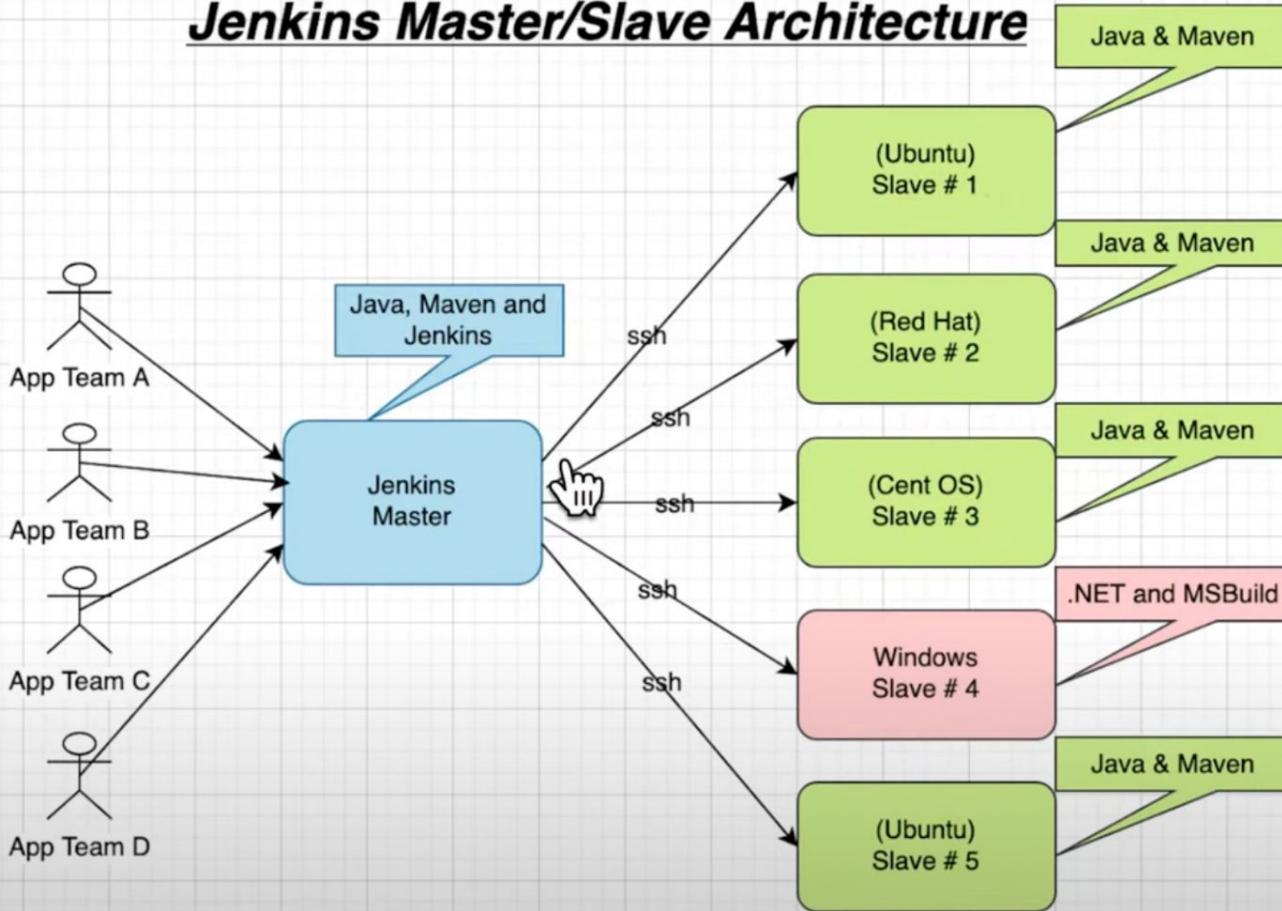
2 Idle

Name	<input type="text" value="Slave1"/>
Description	<input type="text" value="Jenkins Slave"/>
# of executors	<input type="text" value="1"/> ▼
Remote root directory	<input type="text"/> ✖ Remote directory is mandatory
Labels	<input type="text"/>
Usage	<input type="text" value="Use this node as much as possible"/> ▼
Launch method	<input type="text" value="Launch agent agents via SSH"/> ▼
Host	<input type="text"/> ✖ The Host must be specified
Credentials	<input type="text" value="- none -"/> ▼  Add ?
Host Key Verification Strategy	<input type="text" value="Known hosts file Verification Strategy"/> ▼ ?

✖ The selected credentials cannot be found



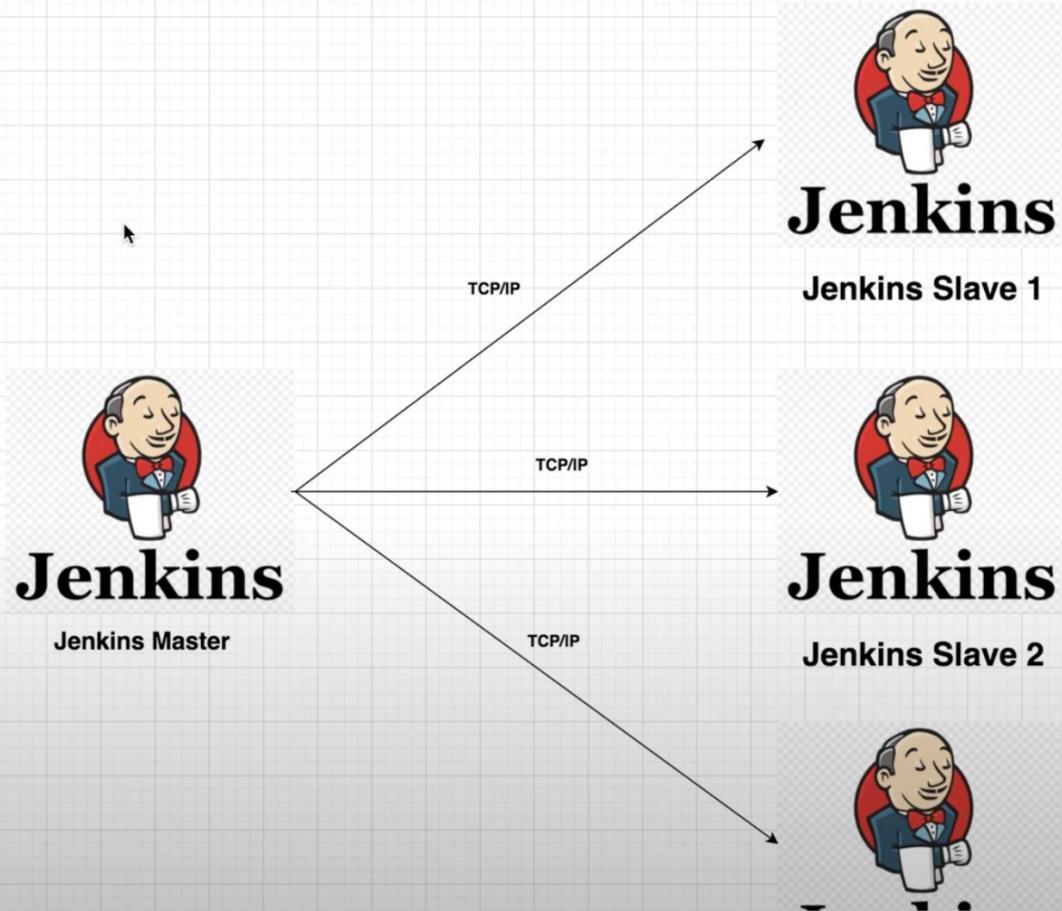
Jenkins Master/Slave Architecture



Advantages

1. Distributed Builds
2. Faster throughput
3. Quicker Feedback
4. Scalable architecture

Jenkins Distributed Architecture



Jenkins Master

Your main Jenkins server is the Master. The Master's job is to handle:

- Scheduling build jobs.
- Dispatching builds to the slaves for the actual execution.
- Monitor the slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master instance of Jenkins can also execute build jobs directly.

Jenkins Slave

A Slave is a Java executable that runs on a remote machine. Following are the characteristics of Jenkins Slaves:

- It hears requests from the Jenkins Master instance.
- Slaves can run on a variety of operating systems.
- The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.
- You can configure a project to always run on a particular Slave machine or a particular type of Slave machine, or simply let Jenkins pick the next available Slave.

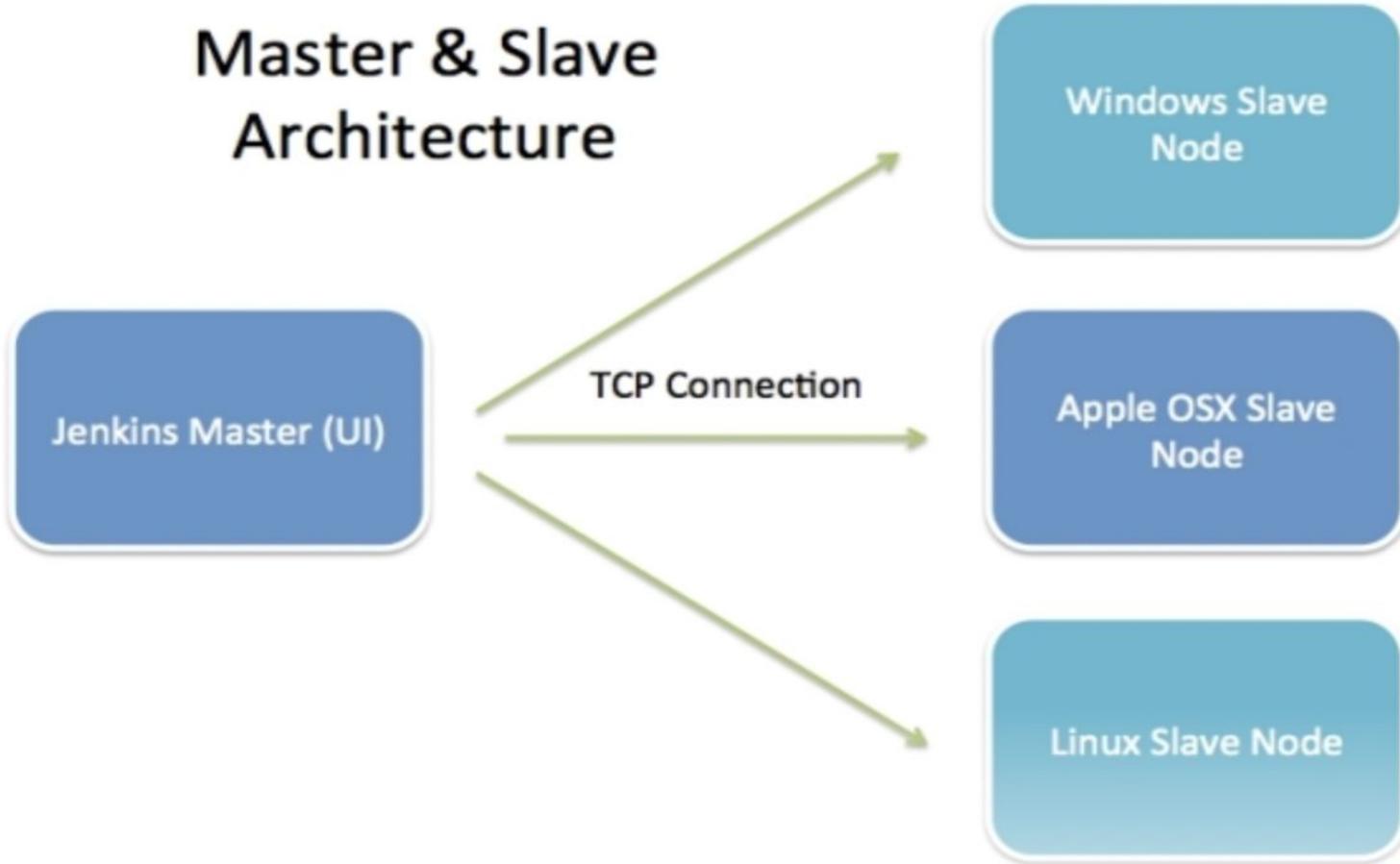
The above image represents the following functions:

- Jenkins checks the Git repository at periodic intervals for any changes made in the source code.
- Each build requires a different testing environment which is not possible for a single Jenkins server. In order to perform testing in different environments, Jenkins uses various Slaves as shown in the diagram.
- Jenkins Master requests these Slaves to perform testing and to generate test reports.

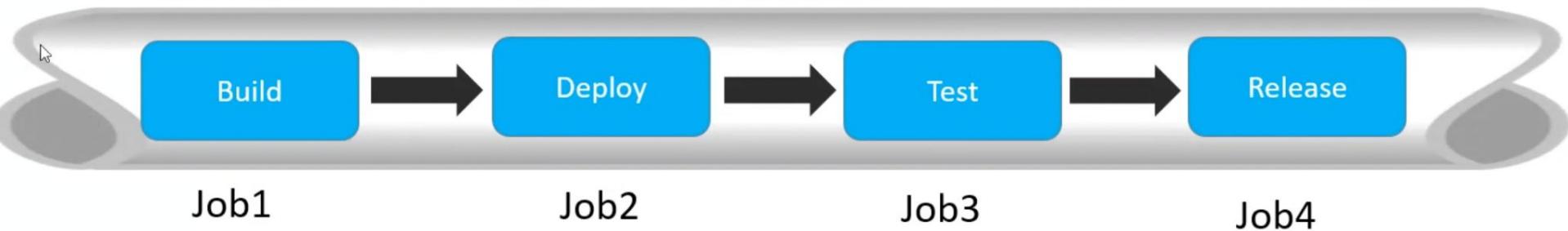
A standalone Jenkins instance can grow fairly quickly into a disk-munching, CPU-eating monster. To prevent this from happening, we can scale Jenkins by implementing a *slave* node architecture, which can help us offload some of the responsibilities of the master Jenkins instance. Let's clarify this concept. A Jenkins slave node is simply a device configured to act as an automation executor on behalf of the master. The Jenkins master simply represents the base installation of Jenkins. The master will continue to perform basic operations and serve the user interface, while the slaves do the heavy lifting.

This distributed computing model will allow the Jenkins master to remain responsive to users, while offloading automation execution to the connected slave(s). To illustrate the concept of a master, and slave mode architecture let's look at an example. *Figure 2-1* shows a Jenkins master and three slave nodes of varying OS types:

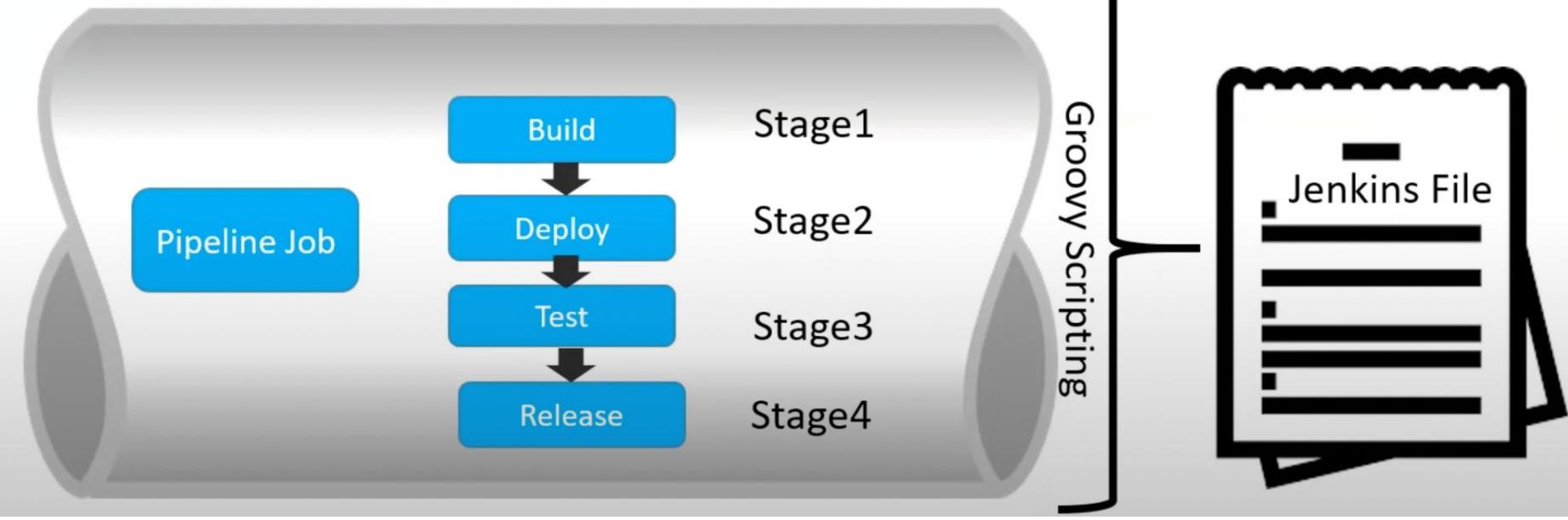
Master & Slave Architecture



Build And Delivery Pipeline Plugins



Pipeline Job with Scripting



1. Instead of having job approach we will have one approach is pipeline approach
2. Every job will be consider has stages
3. We will create only one pipeline job inside this we will have multiple activity
4. All this can be run using the single pipeline
5. Groovy script is simple piece of code
6. We can create single pipeline using single pipeline we can run multiple jobs
7. Pipeline creation itself is only one job

Pipeline concepts

- **Pipeline**
- A Pipeline is a user-defined model of a CD pipeline. A Pipeline's code defines your entire build process, which typically includes stages for building an application, testing it and then delivering it.
- Also, *a **pipeline block** is a key part of Declarative Pipeline syntax.*
- **Node**
- A node is a machine which is part of the Jenkins environment and is capable of executing a Pipeline.
- Also, *a **node block** is a key part of Scripted Pipeline syntax.*
- **Stage**
- A stage block defines a conceptually distinct subset of tasks performed through the entire Pipeline (e.g. "Build", "Test" and "Deploy" stages), which is used by many plugins to visualize or present Jenkins Pipeline status/progress.
- **Step**
- A single task. Fundamentally, a step tells Jenkins what to do at a particular point in time (or "step" in the process). For example, to execute the shell command make use the sh step: sh 'make'. When a plugin extends the Pipeline DSL, that typically means the plugin has implemented a new step.

Node will be very complex is the big organization

Every stage will contain multiple tasks

Pipeline concepts

- **Pipeline**
- A Pipeline is a user-defined model of a CD pipeline. A Pipeline's code defines your entire build process, which typically includes stages for building an application, testing it and then delivering it.
- Also, *a **pipeline block** is a key part of Declarative Pipeline syntax.*
- **Node**
- A node is a machine which is part of the Jenkins environment and is capable of executing a Pipeline.
- Also, *a **node block** is a key part of Scripted Pipeline syntax.*
- **Stage**
- A stage block defines a conceptually distinct subset of tasks performed through the entire Pipeline (e.g. "Build", "Test" and "Deploy" stages), which is used by many plugins to visualize or present Jenkins Pipeline status/progress.
- **Step**
- A single task. Fundamentally, a step tells Jenkins what to do at a particular point in time (or "step" in the process). For example, to execute the shell command make use the sh step: sh 'make'. When a plugin extends the Pipeline DSL, that typically means the plugin has implemented a new step.

When we have single local server in the jenkins than you can go for scripted pipeline

Scripted Pipeline

Jenkinsfile (Scripted Pipeline)

```
node {  
    stage('Build') {  
        //  
    }  
    stage('Test') {  
        //  
    }  
    stage('Deploy') {  
        //  
    }  
}
```

- ➊ Execute this Pipeline or any of its stages, on any available agent.
- ➋ Defines the "Build" stage. `stage` blocks are optional in Scripted Pipeline syntax. However, implementing `stage` blocks in a Scripted Pipeline provides clearer visualization of each `stage's subset of tasks/steps in the Jenkins UI.
- ➌ Perform some steps related to the "Build" stage.
- ➍ Defines the "Test" stage.
- ➎ Perform some steps related to the "Test" stage.
- ➏ Defines the "Deploy" stage.
- ➐ Perform some steps related to the "Deploy" stage.

Enter an item name

Jenkins_scriptedPipe

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

GitHub Pull Requests GitHub hook trigger for GITScm polling Poll SCM Disable this project Quiet period

Advanced Project Options

[Advanced...](#)

Pipeline

Definition

 Pipeline script Pipeline script Pipeline script from SCM Use Groovy Sandbox[Pipeline Syntax](#)[Save](#)[Apply](#)

node {

```
stage('Build')
{
    echo "Building the Project....."
}
```

```
stage('Test')
{
    echo "Testing the Project....."
}
```

```
stage('Deploy')
{
    echo "Deploying the Project....."
}
}
```

Node represent the current system

Pipeline script is used for scripted pipeline

Pipline Script from scm which is used for declarative scripted pipeline

Inside stage we can write steps also

```
pipeline{

    agent any

    stages

    {

        stage('Build')

        {

            steps{

                echo "Building the Project....."

            }

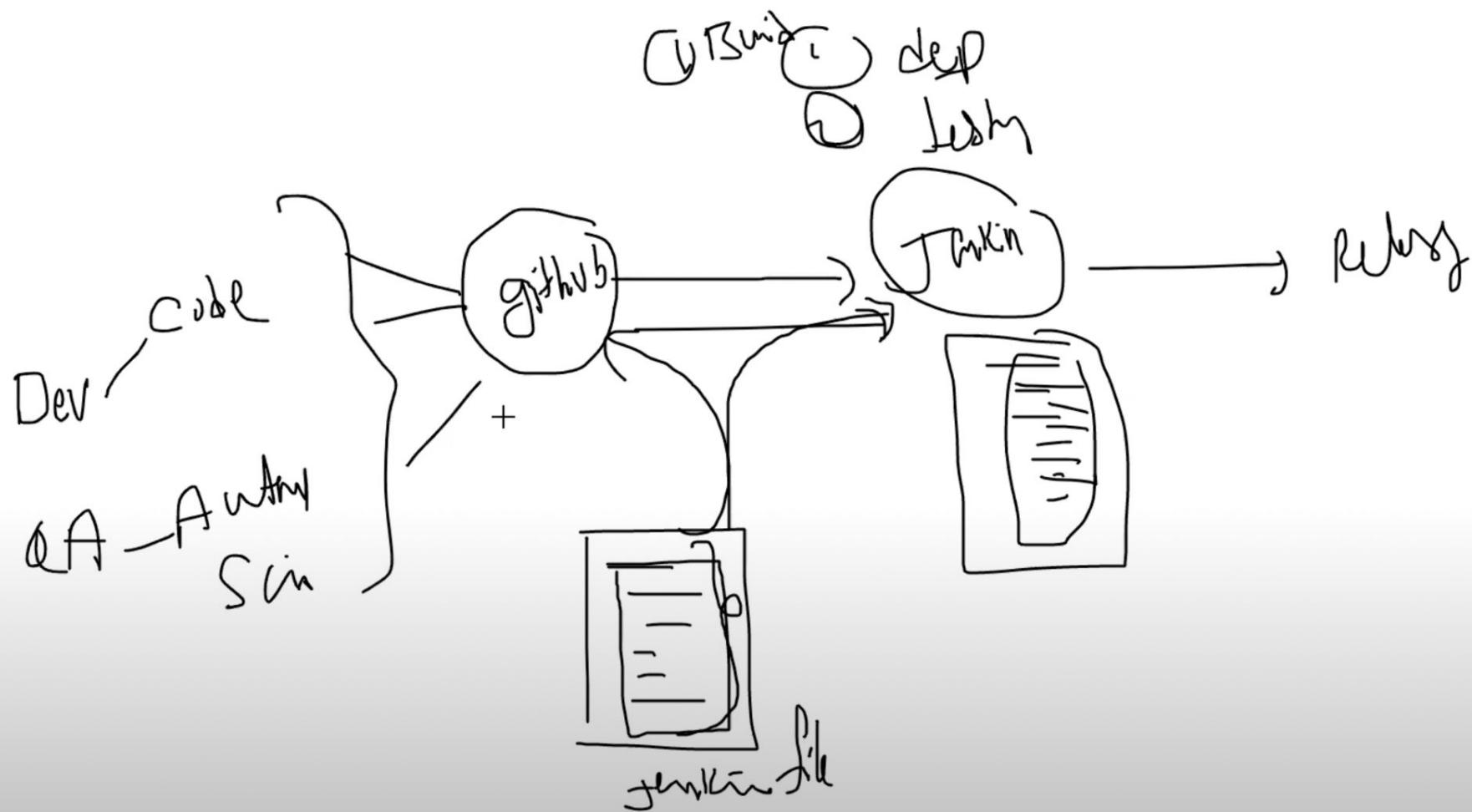
        }

        stage('Test')

    }

}
```

Declarative approach we need to create jenkinsfile



- 1) 1st approach we will write jenkins code
- 2) 2nd approach we will create jenkinsfile in the github

Jenkins file

- Jenkinsfile is a text file that contains the definition of a Jenkins Pipeline and is checked into source control.
- <https://jenkins.io/doc/book/pipeline/jenkinsfile/>

<https://www.jenkins.io/doc/pipeline/tour/hello-world/>

General Build Triggers Advanced Project Options Pipeline

- Build periodically
- Build whenever a SNAPSHOT dependency is built
- GitHub Branches
- GitHub Pull Requests
- GitHub hook trigger for GITScm polling
- Poll SCM
- Disable this project
- Quiet period

Advanced Project Options

[Advanced...](#)

Pipeline

Definition

SCM

Script Path

Lightweight checkout

[Pipeline Syntax](#)

[General](#)[Build Triggers](#)[Advanced Project Options](#)[Pipeline](#)

- Disable this project
- Quiet period



Advanced Project Options

[Advanced...](#)

Pipeline

Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

m/pavanoltraining/jenkinspipeline.git

Please enter Git repository.

Credentials

- none -

Add

[Advanced...](#)[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any')

*/master

[Add Branch](#)

Repository browser

(Auto)



Additional Behaviours

Add

General Build Triggers Advanced Project Options **Pipeline**

Definition Pipeline script from SCM

SCM Git

Repositories

Repository URL <https://github.com/pavanoltraining/jenkinsfile>

Credentials - none - Add Advanced... Add Repository

Branches to build

Branch Specifier (blank for 'any') */master

Add Branch

Repository browser (Auto)

Additional Behaviours Add

Script Path Jenkinsfile

Lightweight checkout

[Pipeline Syntax](#)

Save Apply

Branch: master ▾

JenkinsPipelineDemoProject / jenkinsfile

[Find file](#) [Copy path](#) pavanoltraining Create jenkinsfile

66618ed 2 hours ago

1 contributor

34 lines (33 sloc) 348 Bytes

[Raw](#) [Blame](#) [History](#)   

```
1 pipeline{
2   agent any
3   stages
4   {
5     stage('Build')    ↴
6     {
7       steps{
8         echo "Building the Code....."
9         bat "mvn clean"
10      }
11    }
12    stage('Test')
13    {
14      steps{
15        echo "Testing the Code....."
16        bat "mvn test"
17      }
18    }
19    stage('Compile')
20    {
21      steps{
22        echo "Compiling the Project....."
23        bat "mvn compile"
24      }
25    }
26    stage('Deploy')
27    {
28      steps{
29        echo "Deploying the Project....."
30      }
31    }
32  }
33 }
```

 Back to Dashboard Manage Jenkins New Node Configure Clouds Node Monitoring

Build Queue ^

No builds in the queue.

Build Executor Status ^

1 Idle

2 Idle

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time	
	Built-In Node  Data obtained	Linux (amd64)	In sync	3.87 GB	 0 B	3.87 GB	0ms	

[Refresh status](#)

Jenkins

Dashboard > Nodes >

Back to Dashboard

Manage Jenkins

New Node

Configure Clouds

Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Node name

java-node

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

OK

[Back to Dashboard](#)[Manage Jenkins](#)[New Node](#)[Configure Clouds](#)[Node Monitoring](#)

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Name

java-node



Description

this node helps to build java applications



Number of executors

1



Remote root directory

**Remote directory is mandatory**

Labels



Usage

 Use this node as much as possible

Launch method



Executors is how many parallel jobs we can run at a time is the parallel threads

Dashboard > Nodes >

[Back to Dashboard](#) [Manage Jenkins](#) [New Node](#) [Configure Clouds](#) [Node Monitoring](#)

Build Queue ^
No builds in the queue.

Build Executor Status ^
1 Idle
2 Idle

Name: java-node

Description: this node helps to build java applications

Number of executors: 1 (highlighted with a yellow circle)

Remote root directory: **Remote directory is mandatory**

Labels:

Usage: Use this node as much as possible

Launch method:

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

2

Remote root directory



/opt/build

Labels



java-build-node

Usage



Use this node as much as possible

Use this node as much as possible

Only build jobs with label expressions matching this node



Launch agent by connecting it to the controller

 Disable WorkDir

Custom WorkDir path



Internal data directory



remoting

 Fail if workspace is missing Use WebSocket

Either WebSocket mode is selected, or the TCP port for inbound agents must be enabled

Labels [?](#)

java-build-node

Usage [?](#)

Use this node as much as possible

Launch method [?](#)

Launch agent by connecting it to the controller

 Disable WorkDir [?](#)Custom WorkDir path [?](#)Internal data directory [?](#)remoting Fail if workspace is missing [?](#) Use WebSocket [?](#)

1. Whatever the job having that labels are executed in the slave system if we are using the lables
- 2.

Build Executor Status

1 Idle

2 Idle

Labels

java-build-node

Usage

Use this node as much as possible

Launch method

Launch agent by connecting it to the controller

Launch agent by connecting it to the controller

Launch agent via execution of command on the controller

Let Jenkins control this Windows agent as a Windows service

Internal data directory

remoting

Fail if workspace is missing

Use WebSocket

Either WebSocket mode is selected, or the TCP port for inbound agents must be enabled

Availability

Advanced...

Execute command on the slave to connect to master

Execute command on the master to connect to slave

Status

Delete Agent

Configure

Build History

Load Statistics

Log

Build Executor Status

Agent java-node-slave (this node helps to build java applications)

[Mark this node temporarily offline](#)

Run from agent command line:

```
curl -sO http://3.238.37.89:8080/jnlpJars/agent.jar  
java -jar agent.jar -jnlpUrl http://3.238.37.89:8080/manage/computer/java%2Dnode%2Dslave/jenkins-agent.jnlp -secret  
22f743ad58ffcd515232cb1939a4ce2a4116280e0128b2c0903fede604ad8ae2 -workDir "/opt/build"
```

Or run from agent command line, with the secret stored in a file:

```
echo 22f743ad58ffcd515232cb1939a4ce2a4116280e0128b2c0903fede604ad8ae2 > secret-file  
curl -sO http://3.238.37.89:8080/jnlpJars/agent.jar  
java -jar agent.jar -jnlpUrl http://3.238.37.89:8080/manage/computer/java%2Dnode%2Dslave/jenkins-agent.jnlp -secret @secret-  
file -workDir "/opt/build"
```

Note: PowerShell users must use curl.exe instead of curl because curl is a default PowerShell cmdlet alias for Invoke-WebRequest.

Labels

java-build-node

Projects tied to java-node-slave

None

 Status

 Delete Agent

 Configure

 Build History

 Load Statistics

 Script Console

 Log

 System Information

 Disconnect

Agent java-node-slave (this node helps to build java applications)

[Mark this node temporarily offline](#)

Agent is connected.

Labels

 java-build-node

Projects tied to java-node-slave

None

Build Executor Status



1 Idle

Last login: Thu Feb 16 17:01:52 2023 from ec2-18-206-107-27.compute-1.amazonaws.com

```
_|_ _|_)  
_| ( / Amazon Linux 2 AMI  
__|\_\_|
```

<https://aws.amazon.com/amazon-linux-2/>

[ec2-user@ip-172-31-59-237 ~]\$

[ec2-user@ip-172-31-59-237 ~]\$

[ec2-user@ip-172-31-59-237 ~]\$

[ec2-user@ip-172-31-59-237 ~]\$ java -jar agent.jar -jnlpUrl http://3.238.37.89:8080/manage/computer/java%2Dnode%2Dslave/jenkins-agent.jnlp -secret 22f743ad58ffcd515232cb1939a4ce2a4116280e0128b2c0903fede604ad8ae2 -workDir "/home/ec2-user"

Feb 16, 2023 5:02:11 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir

INFO: Using /home/ec2-user/remoting as a remoting work directory

Feb 16, 2023 5:02:11 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging

INFO: Both error and output logs will be printed to /home/ec2-user/remoting

Feb 16, 2023 5:02:11 PM hudson.remoting.jnlp.Main createEngine

INFO: Setting up agent: java-node-slave

Feb 16, 2023 5:02:11 PM hudson.remoting.Engine startEngine

INFO: Using Remoting version: 3077.vd69cf116da_6f

Feb 16, 2023 5:02:11 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir

INFO: Using /home/ec2-user/remoting as a remoting work directory

Feb 16, 2023 5:02:12 PM hudson.remoting.jnlp.Main\$CuiListener status

INFO: WebSocket connection open

Feb 16, 2023 5:02:12 PM hudson.remoting.jnlp.Main\$CuiListener status

INFO: Connected

^C[ec2-user@ip-172-31-59-237 ~]\$ uptime

17:02:36 up 4:36, 2 users, load average: 0.06, 0.03, 0.00

[ec2-user@ip-172-31-59-237 ~]\$ █

Jenkins
Master

Jenkins
Slave

QA

Jenkins
Slave

SIT

Jenkins
Slave

DEV

