

Get the best out of Live Sessions HOW?

e!



Check your Internet Connection

Log in 10 mins before, and check your internet connection to avoid any network issues during the LIVE session.

Speak with the Instructor

By default, you will be on mute to avoid any background noise. However, if required you will be **unmuted by instructor**.



Clear Your Doubts

Feel free to clear your doubts. Use the “Questions” tab on your webinar tool to interact with the instructor at any point during the class.

Let us know if you liked our content

Please share feedback after each class. It will help us to enhance your learning experience.



edureka!



DevOps Certification Training

COURSE OUTLINE

MODULE 04

1. Introduction to DevOps

2. Version Control with Git

3. Git and Jenkins

4. Continuous Integration with Jenkins

5. Configuration Management using Ansible

6. Containerization using Docker Part - I



7. Containerization using Docker Part - II

8. Container Orchestration Using
Kubernetes Part-I

9. Container Orchestration Using
Kubernetes Part-II

10. Monitoring Using Prometheus and
Grafana

11. Provisioning Infrastructure using
Terraform Part - I

12. Provisioning Infrastructure using
Terraform Part - II

edureka!

Module 4 – Continuous Integration With Jenkins

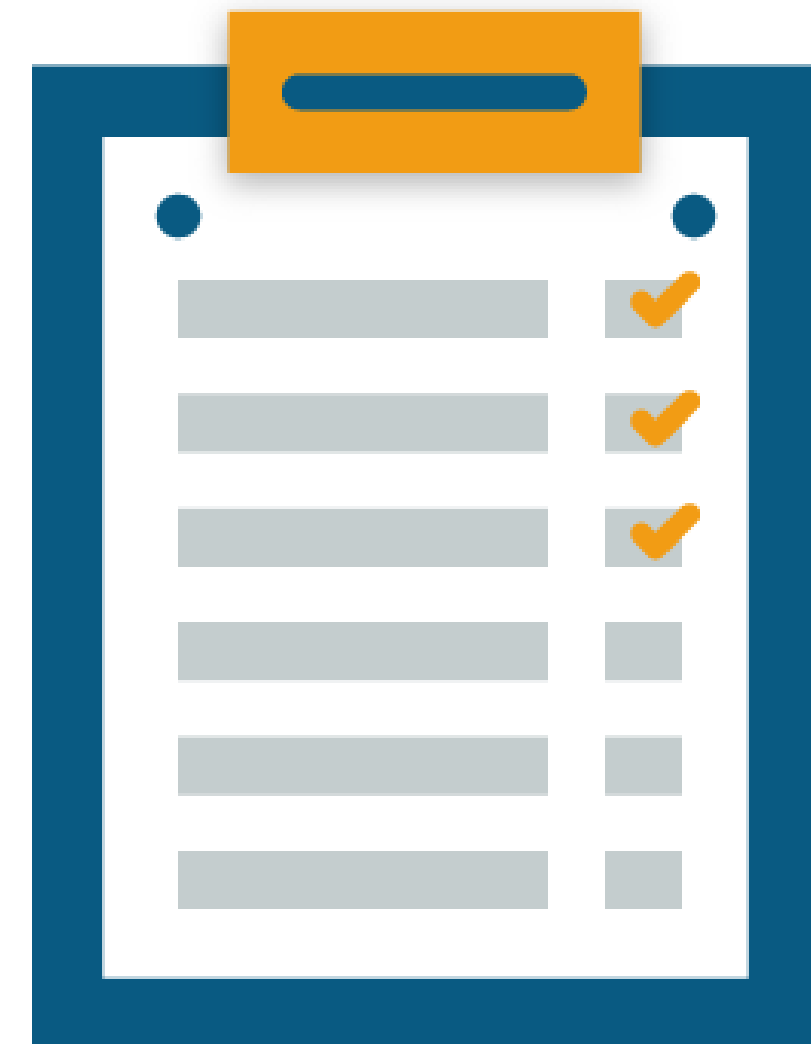
edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Topics

Following are the topics covered in this module:

- Jenkins Architecture
- Plugin Management in Jenkins
- Jenkins Security Management
- Notification in Jenkins
- Jenkins Master slave architecture
- Jenkins Delivery Pipeline
- Jenkins Declarative pipeline



Objectives

After completing this module, you should be able to:

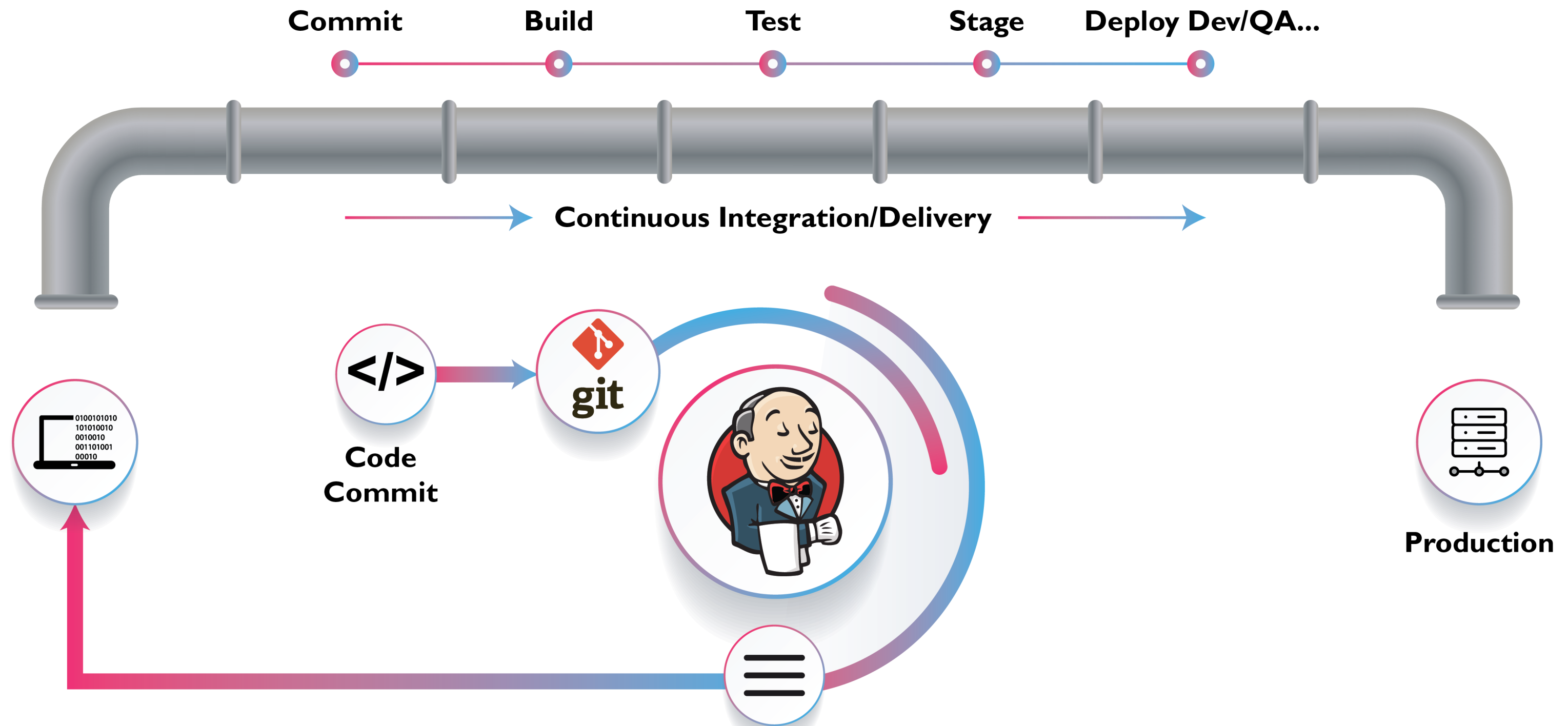
- Managing authorization in Jenkins
- Jenkins notification management
- Master slave architecture in Jenkins
- Adding a slave node to Jenkins master
- Build and deploy codes using Jenkins
- Build pipeline plugin in Jenkins
- Declarative pipeline in Jenkins



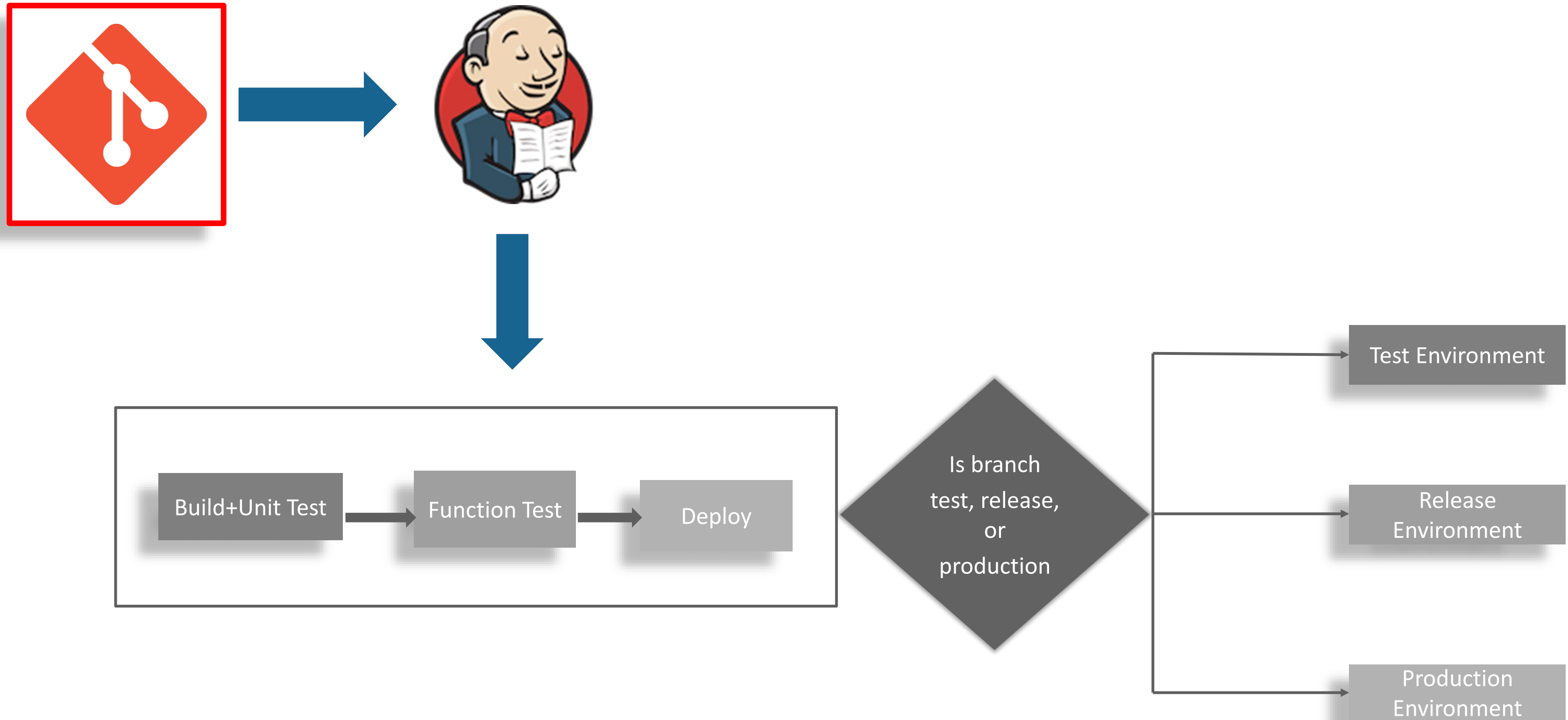


Jenkins Architecture

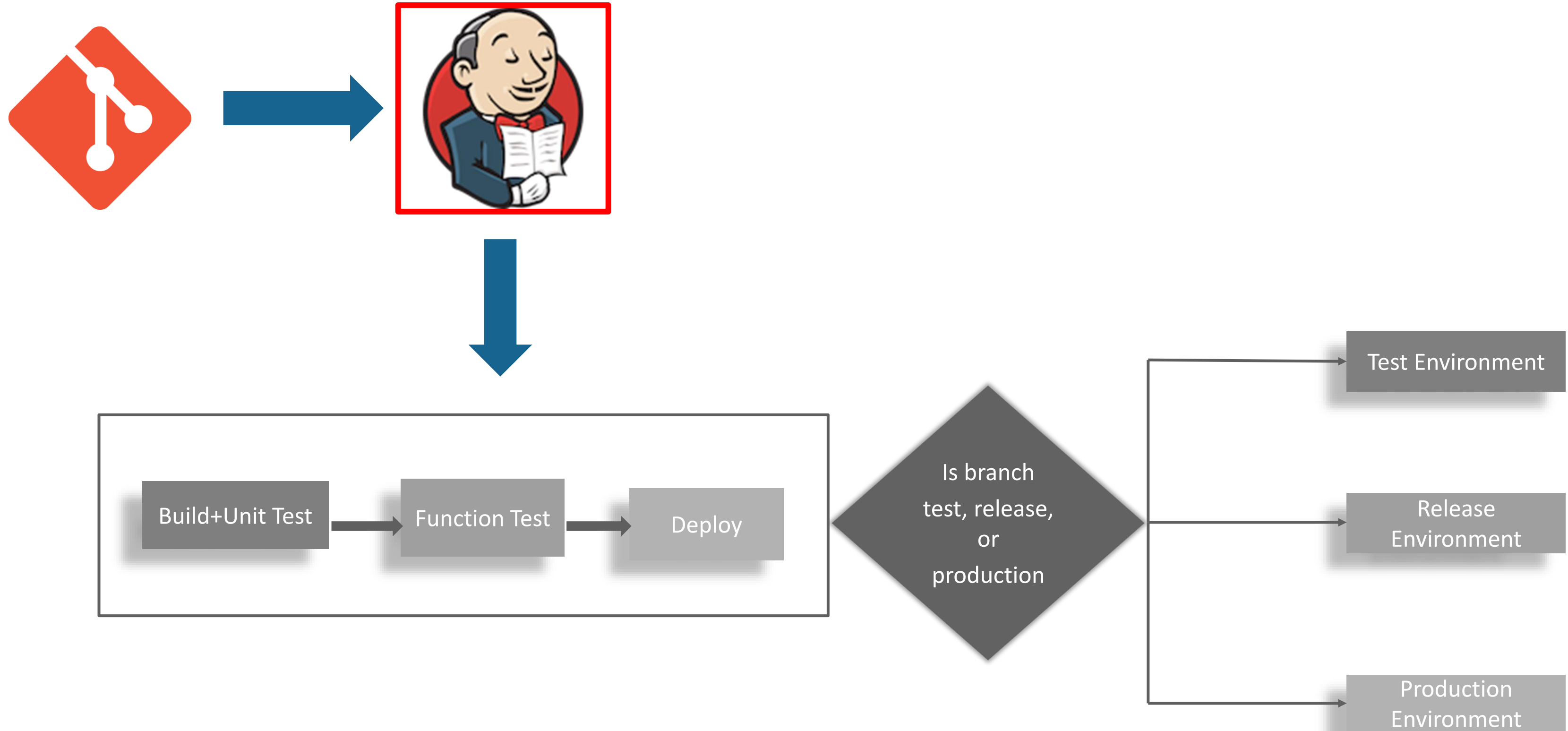
Role of Jenkins in DevOps



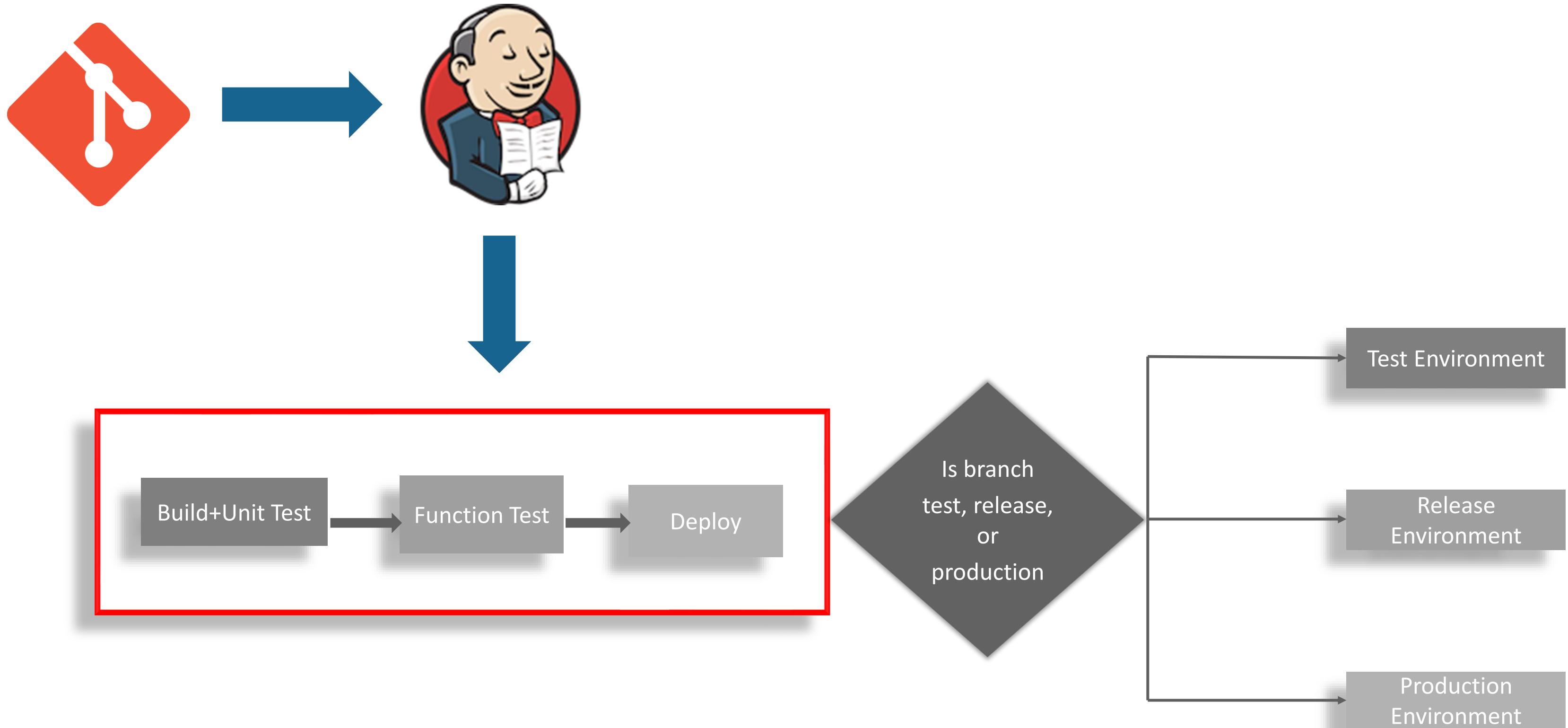
Jenkins Architecture: Source Control Management



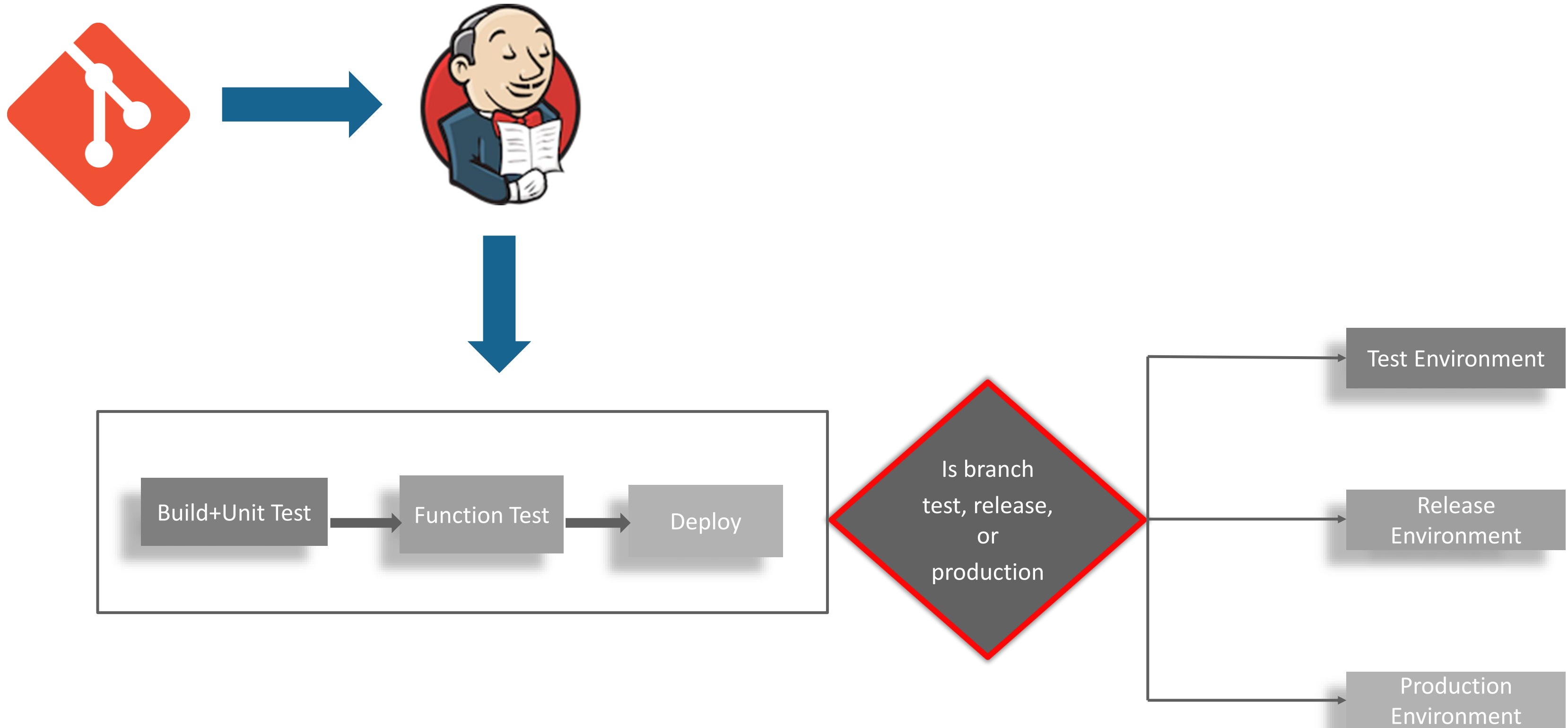
Jenkins Architecture: Jenkins Operation



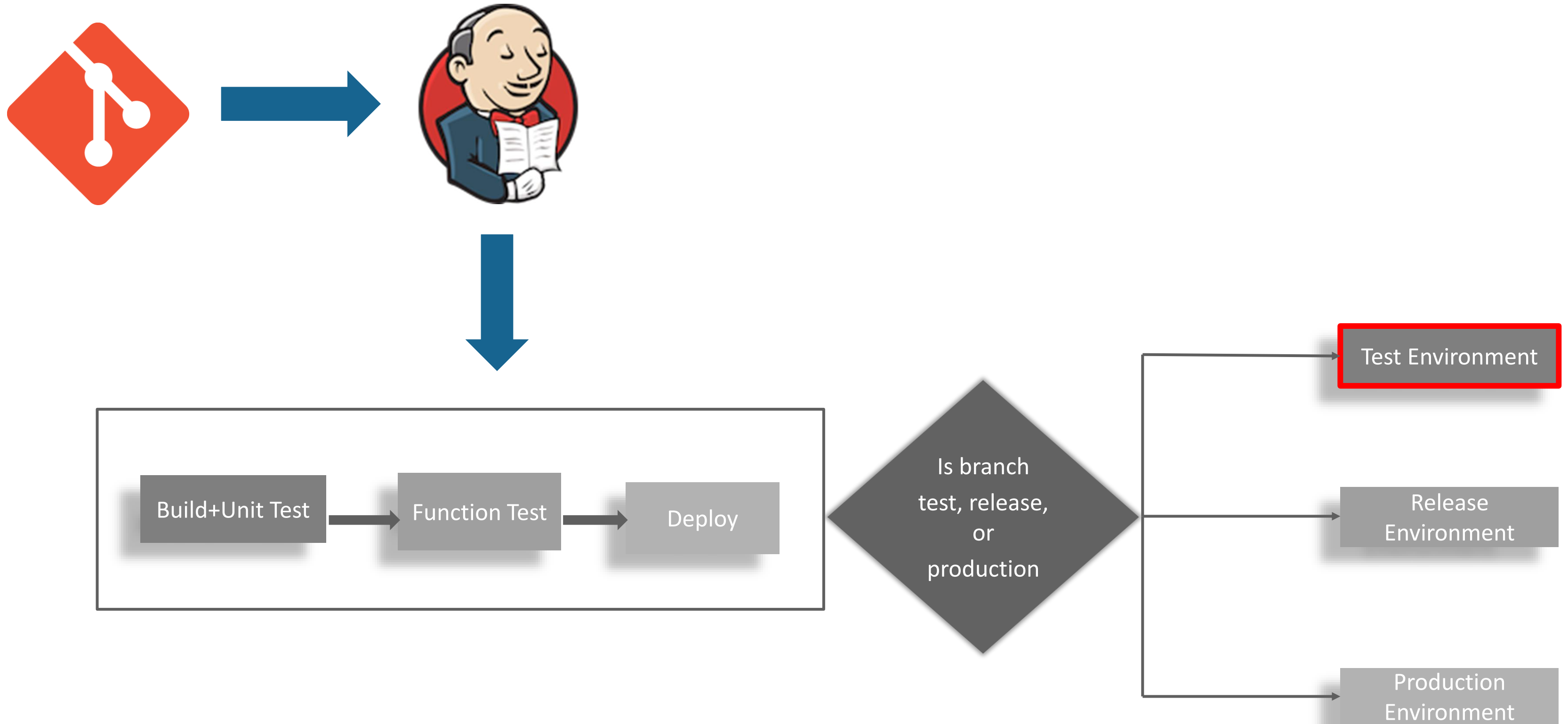
Jenkins Architecture: Build, Test & Deploy



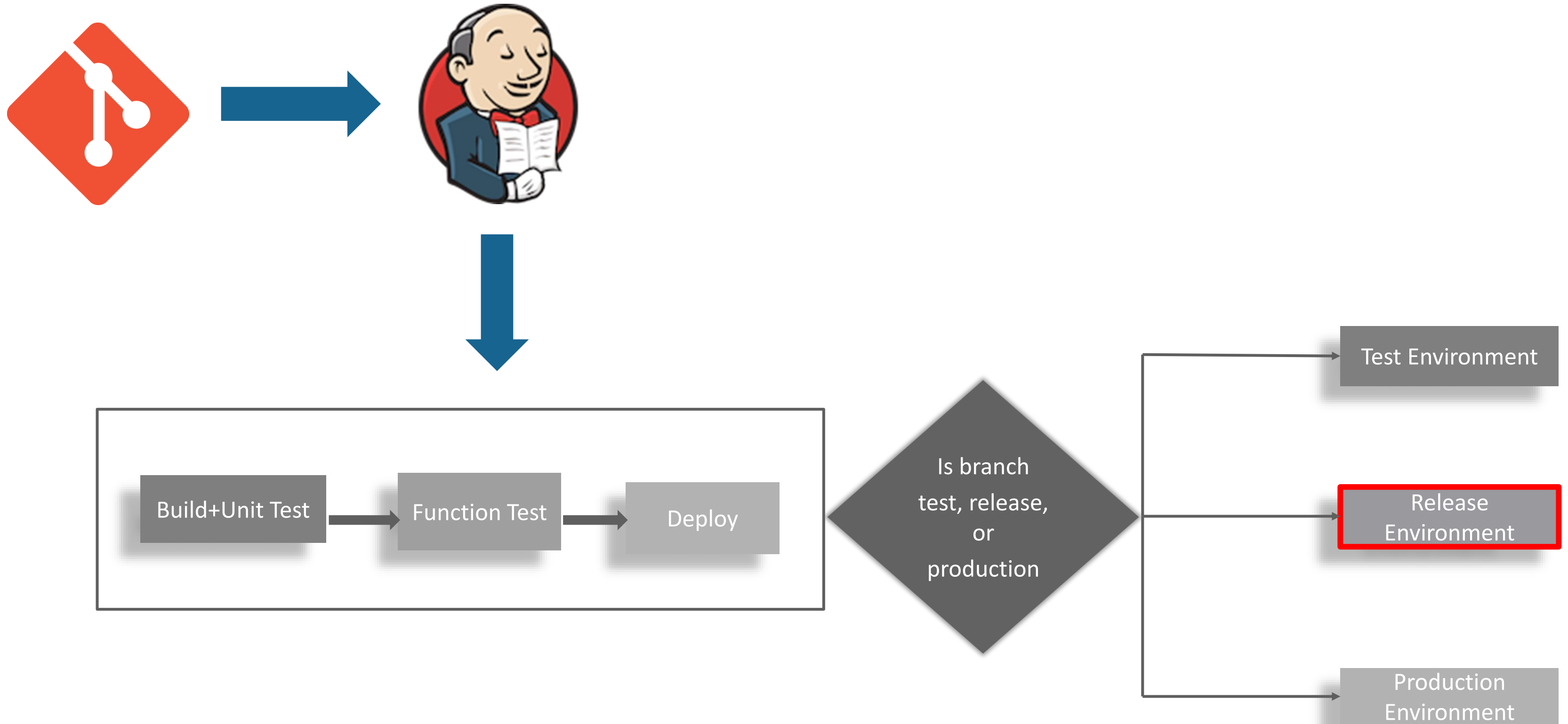
Jenkins Architecture: Condition Check



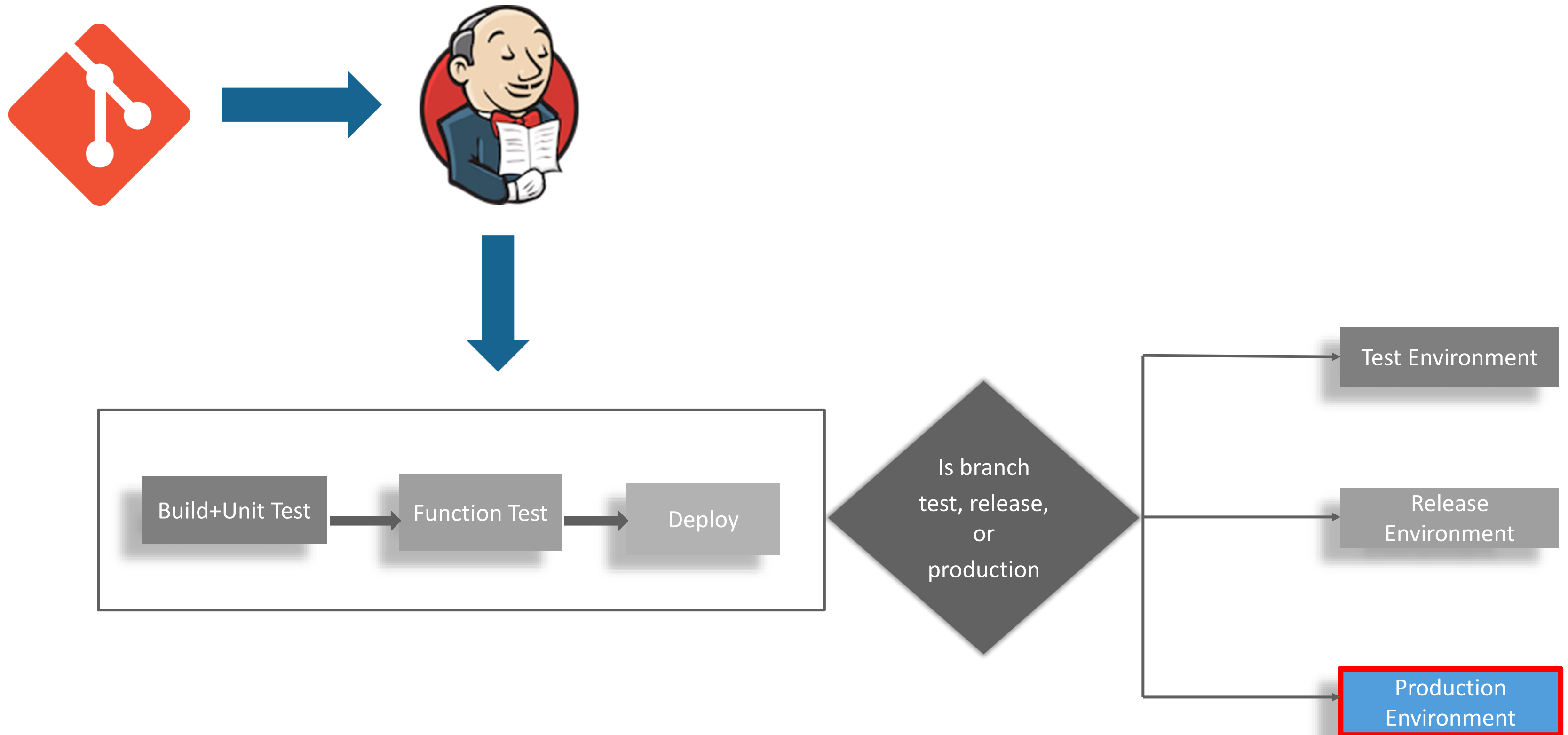
Jenkins Architecture: Deployed for Testing



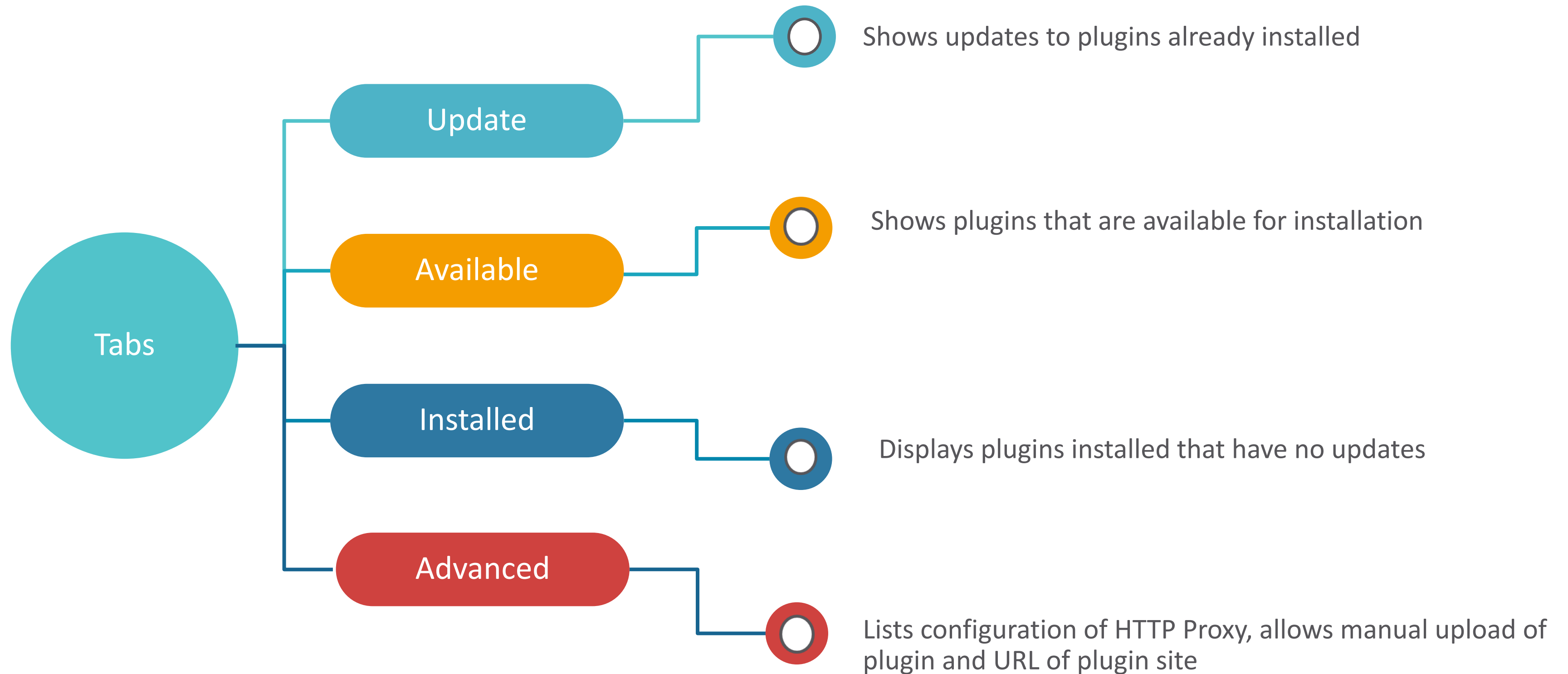
Jenkins Architecture: Deployed For Release



Jenkins Architecture: Deployed For Production



Plugin Management in Jenkins



Jenkins Projects

Free Style Project

This is the central feature of Jenkins. Jenkin will build your project by combining any SCM with any build system.



Multi-configuration Project

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



Github Organization

Scans a Github Organization for all matching repositories..



Pipeline

Suitable for building pipelines or organizing complex activities that do not easily fit in free style



Folder

Creates a container and stores nested items in it. Useful for grouping things together.



Multibranch Pipeline

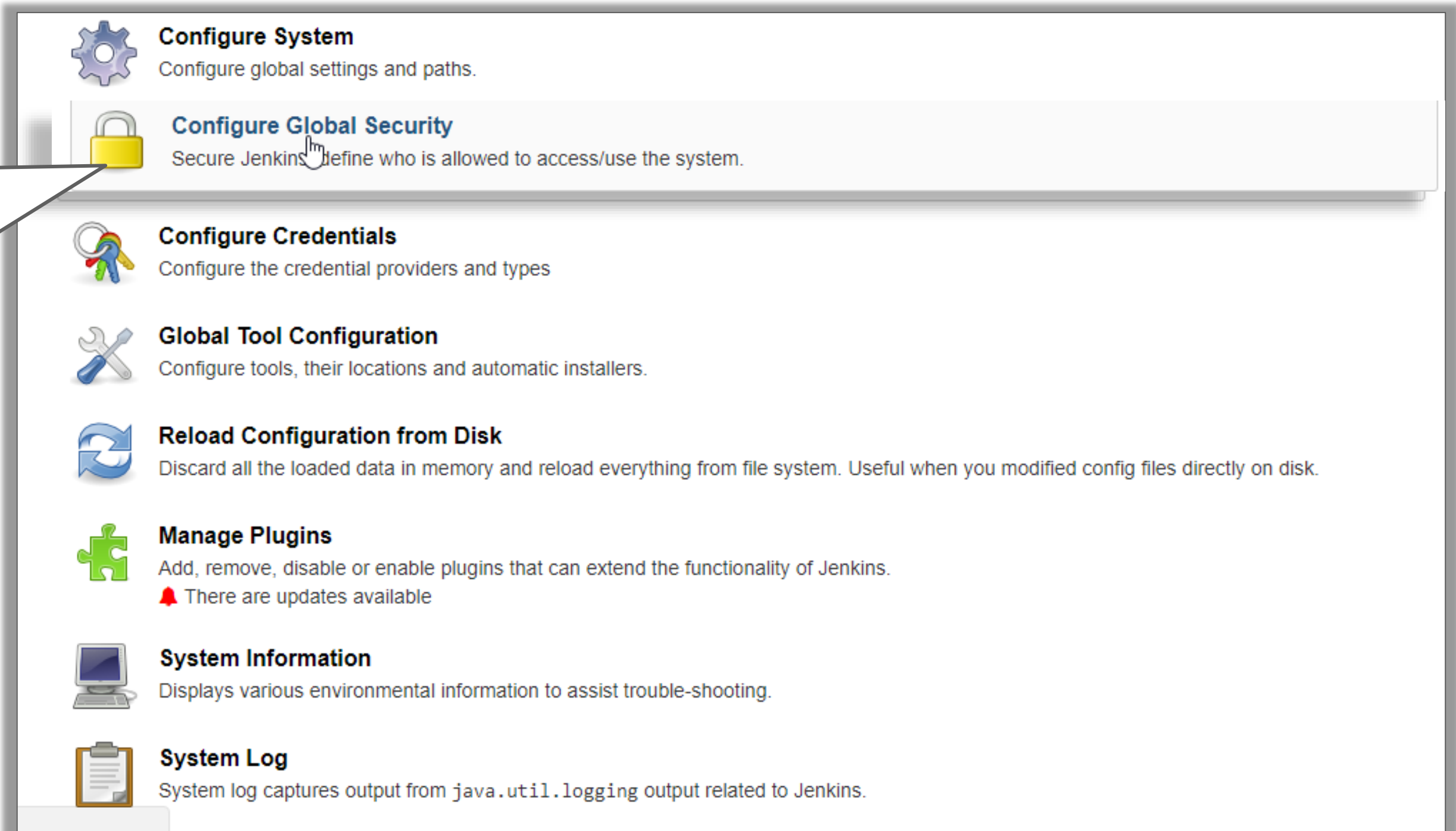
Creates a set of pipeline projects according to detected branches in one SCM repositories.



Jenkins Management: Securing Jenkins

Securing Jenkins: Configure Global Security

One Can manage security level in Jenkins environment and projects through Jenkins Security.



Securing Jenkins: Manage Authorization

Any one can do anything

Configure Global Security

☒ Enable security
Disable remember me ☐
Access Control

Security Realm

- ☐ Delegate to servlet container
- ☒ Jenkins' own user database
 - ☐ Allow users to sign up
- ☐ LDAP

Authorization

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


Markup Formatter

This is the least secure setup allowing anyone to perform any operation on Jenkins Server.

NOTE: Avoid using Authorization of type "Anyone can do anything".

Securing Jenkins: Manage Authorization

Legacy Mode

 **Configure Global Security**

☒ Enable security

Disable remember me ☐

Access Control

Security Realm

- ☐ Delegate to servlet container
- ☒ Jenkins' own user database
 - ☐ Allow users to sign up
- ☐ LDAP

Authorization

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

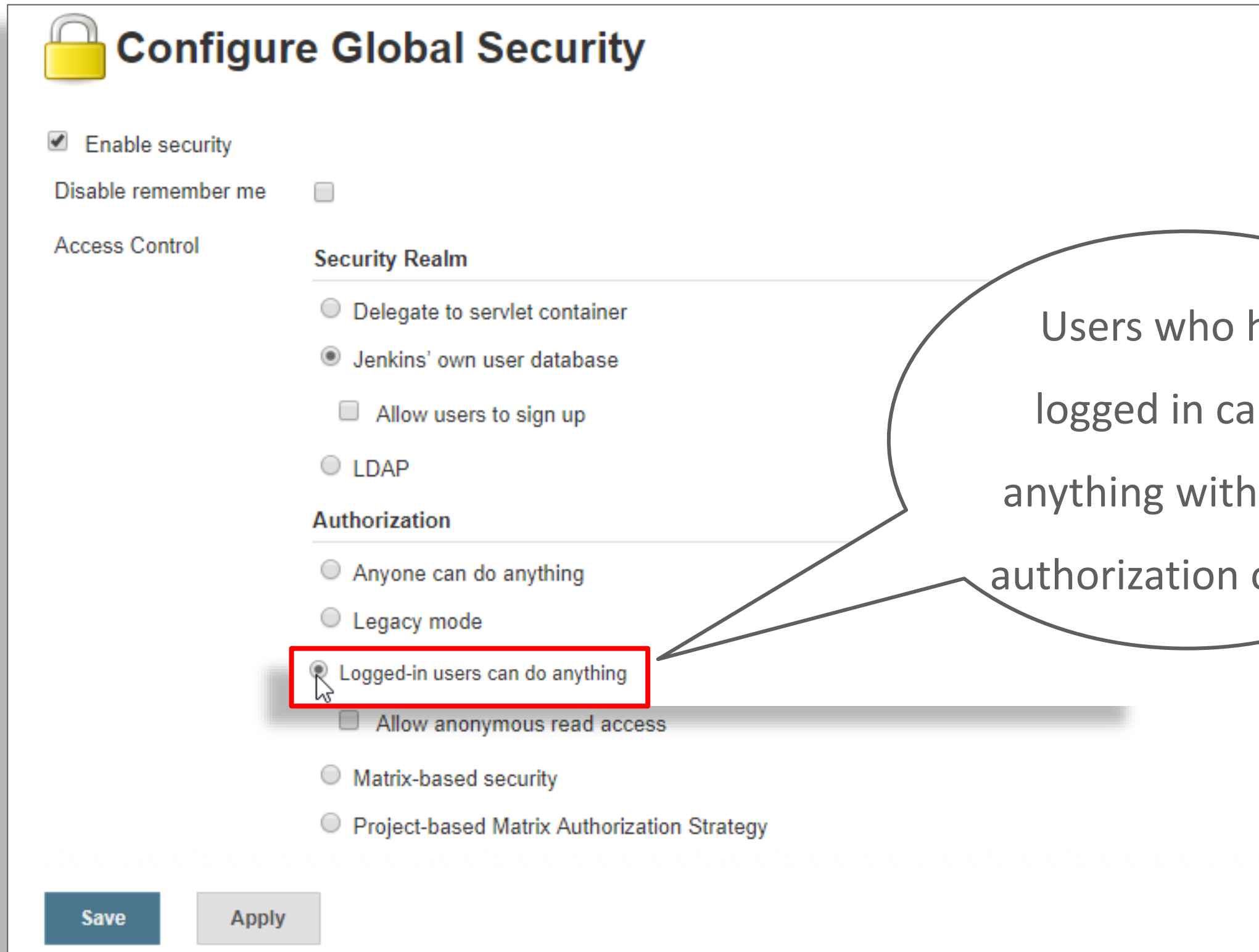
Markup Formatter


Save **Apply**

This authorization
uses existing projects.

Securing Jenkins: Manage Authorization

Logged-in users
can do anything



 **Configure Global Security**

☒ Enable security

Disable remember me ☐

Access Control

Security Realm

- ☐ Delegate to servlet container
- ☒ Jenkins' own user database
 - ☐ Allow users to sign up
- ☐ LDAP

Authorization


- ☐ Anyone can do anything
- ☐ Legacy mode
- ☒ Logged-in users can do anything
- ☐ Allow anonymous read access
- ☐ Matrix-based security
- ☐ Project-based Matrix Authorization Strategy

Save **Apply**

Users who have
logged in can do
anything within this
authorization option.

Matrix – Based Security Mode

Matrix Based Security



Configure Global Security

☒ Enable security
Disable remember me ☐
Access Control

Security Realm

☐ Delegate to servlet container

☒ Jenkins' own user database

☐ Allow users to sign up

☐ LDAP

Authorization

☐ Anyone can do anything

☐ Legacy mode

☐ Logged-in users can do anything

☒ Matrix-based security

User/group	Overall	Credentials	Agent	Job																		
	Administer	Read	Create	Delete	Manage	Domains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Build	Cancel	Configure	Create	Delete	Discover	Move	Force
Anonymous Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

User/group to add:

User1

Add

☐ Project-based Matrix Authorization 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

TCP port for JNLP agents

☐ Fixed :

☐ Random

☒ Disable

Save

Apply

Matrix-based security is the best practice among different modes of Authorization.

Project-Based Matrix Authorization Strategy

Project Based Matrix Authorization Strategy

Configure Global Security

☒ Enable security
Disable remember me Access Control

☐ Security Realm

- ☐ Delegate to servlet container
- ☒ Jenkins' own user database
 - ☐ Allow users to sign up
- ☐ LDAP

Authorization

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

User/group	Overall	Credentials				Agent				Job										
	Administer	Read	Create	Delete	Manage	Domains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Build	Cancel	Configure	Create	Delete	Discover
Anonymous Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

User/group to add:

Markup Formatter

Markup Formatter

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

Agents

This Authorization strategy helps in managing the users access for various task with respect to projects.



Jenkins Management: Notification

Notification In Jenkins

Back to Dashboard

Manage Jenkins

Filter: email exte

UpdatesAvailableInstalledAdvanced

Enabled	Name ↓	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	bouncycastle API Plugin This plugin provides an stable API to Bouncy Castle related tasks.	2.16.2		Uninstall
<input checked="" type="checkbox"/>	Command Agent Launcher Plugin Allows agents to be launched using a specified command.	1.2		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	2.61		Uninstall
<input checked="" type="checkbox"/>	JUnit Plugin Allows JUnit-format test results to be published.	1.23		Uninstall
<input checked="" type="checkbox"/>	Email Publisher Plugin This plugin allows you to configure email notifications for build results	1.20		Uninstall
<input checked="" type="checkbox"/>	Matrix Project Plugin Multi-configuration (matrix) project type.	1.12		Uninstall
<input checked="" type="checkbox"/>	Pipeline: Job			

Jenkins has the Email Extension Plugin which enables Jenkins to send the email to the developer to inform about the job.



Email Notification Setup

Extended E-mail Notification

SMTP server

smtp.gmail.com

Default user E-mail suffix

☒ Use SMTP Authentication

User Name

abc@gmail.com

Password

Use SSL

☒

SMTP port

465

Charset

UTF-8

Default Content Type

Plain Text (text/plain)

☐ Use List-ID Email Header

☐ Add 'Precedence: bulk' Email Header

Default Recipients

Reply To List

Emergency reroute

Excluded Recipients

Default Subject

\$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS!

Maximum Attachment Size

Default Content

\$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS:
Check console output at \$BUILD_URL to view the results.

Save

Apply

1

Smtp server needs to be mentioned

2

User has to enter the email address & password of that email

3

Smtp Port number for gmail server is 465

Email Configuration

Extended E-mail Notification

SMTP server

smtp.gmail.com

Default user E-mail suffix

☒ Use SMTP Authentication

User Name

abc@gmail.com

Password

Use SSL

☒

SMTP port

465

Charset

UTF-8

Default Content Type

Plain Text (text/plain)

☐ Add 'Precedence: bulk' Email Header

Default Recipients

Reply To List

Emergency reroute

Excluded Recipients

Default Subject

\$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS!

Maximum Attachment Size

Default Content

\$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS:
Check console output at \$BUILD_URL to view the results.

Save

Apply

4

Jenkins send the mail in plain text format

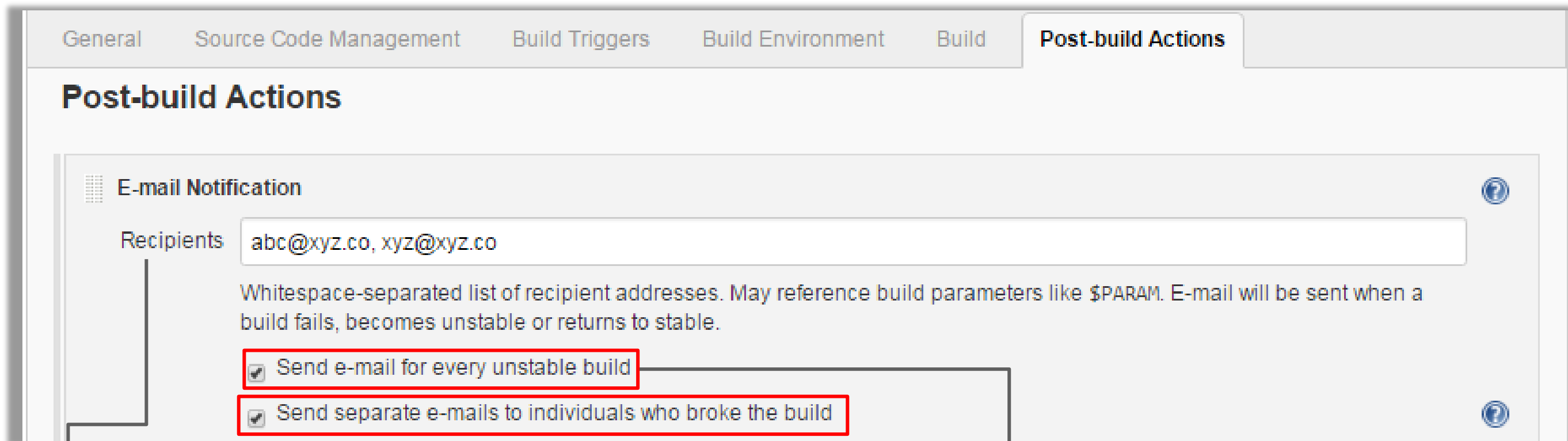
5

Subject in mail is set default in Jenkins

6

Format in which Jenkins will send mail

Post Build Actions



The screenshot shows the Jenkins 'Post-build Actions' configuration page. At the top, there are tabs for 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. The 'Post-build Actions' tab is selected. Below the tabs, the section is titled 'Post-build Actions'. Under this, there is a section for 'E-mail Notification' with a help icon. The 'Recipients' field contains the text 'abc@xyz.co, xyz@xyz.co'. Below this field is a descriptive text: 'Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.' There are two checkboxes: 'Send e-mail for every unstable build' and 'Send separate e-mails to individuals who broke the build'. Both checkboxes are checked and are highlighted with red rectangular boxes. Arrows from these boxes point to explanatory text boxes below the form.

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

Post-build Actions

E-mail Notification ⓘ

Recipients

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ Send e-mail for every unstable build ⓘ

☒ Send separate e-mails to individuals who broke the build ⓘ

Mentioned Email-Id receives mails from Jenkins for different build information.

Clicking on this checkbox enables Jenkins to send separate mail to individuals

Clicking on this checkbox enables Jenkins to send mail for unstable builds



Adding Slave Node To Jenkins

Jenkins Master

- Jenkins Master performs basic installation and handles task related to builds and configuration
- They schedule builds
- They monitor slaves
- Records and presents the build result

Jenkins Slave

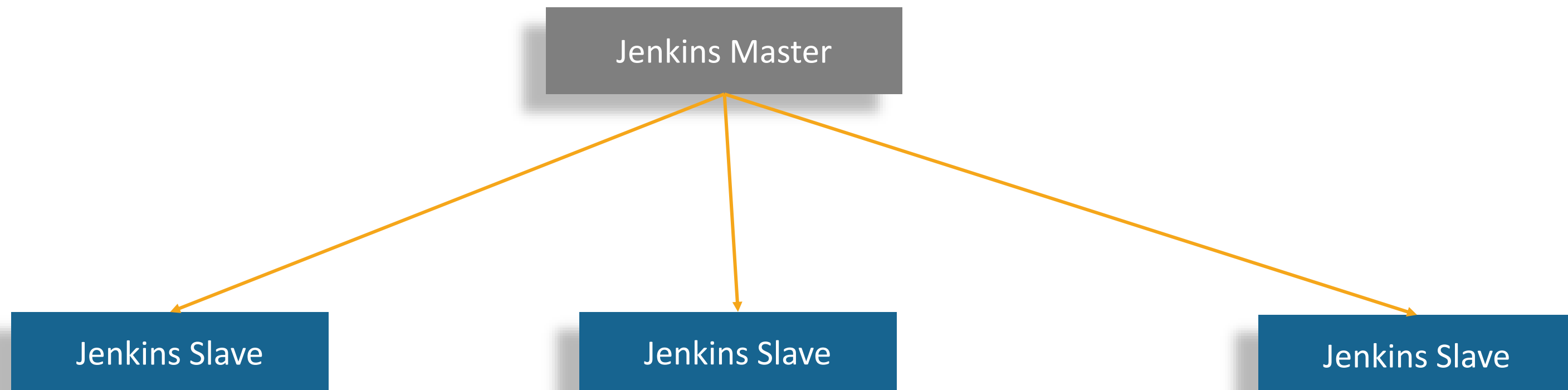
- Slaves are basically set up to offload builds from the master and distribute the workload
- They listen to the master's request
- Slaves can run on a variety of operating systems
- They mainly execute build jobs which are dispatched by Jenkins Master instance



CHARACTERISTICS

Master Slave Architecture In Jenkins

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

How To Setup Slaves On Jenkins

- Using username and password
- Using ssh keys

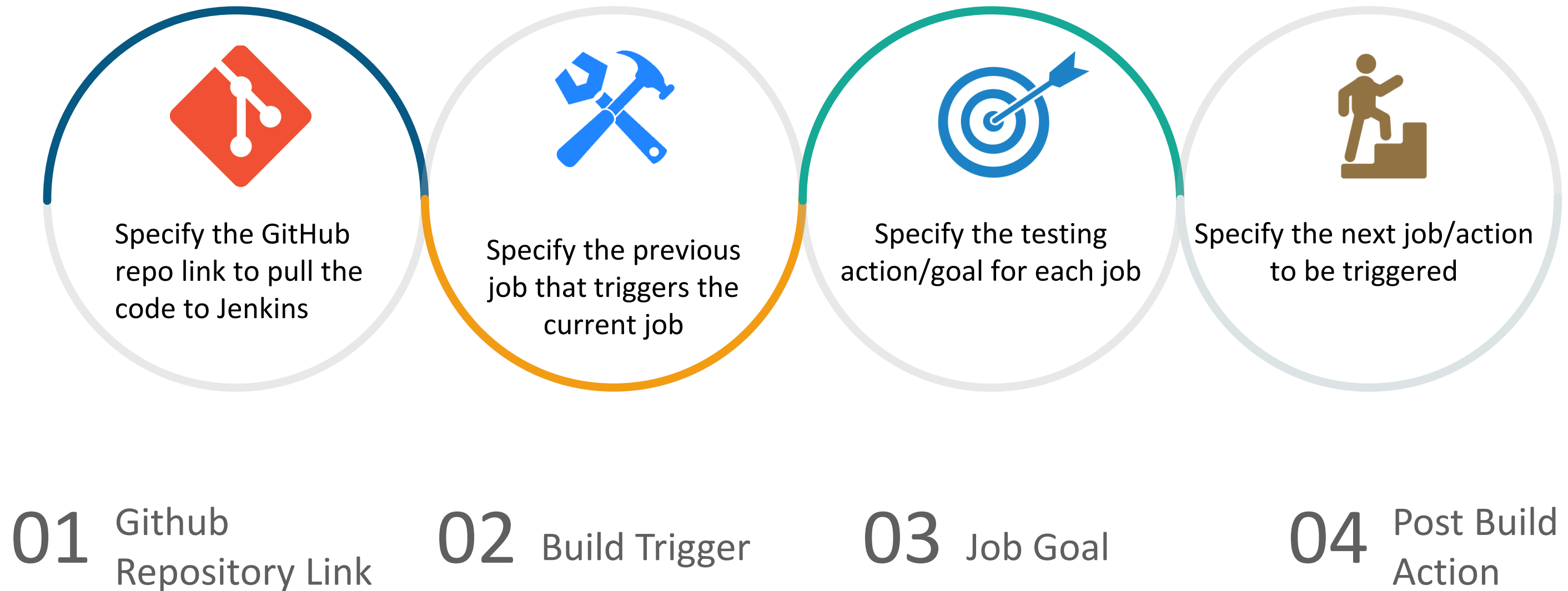
Preparing Slaves using ssh key follow the same steps as setting up of Slave setup using username and password does. The only difference is that instead of username and password for slave, ssh key is used.



Build Delivery Pipeline Using Jenkins

Build A Job

The set of tasks to build the job are listed below:



Continuous Delivery Pipeline

- It enables a constant flow of changes into production via an automated software production line.
- A typical CD pipeline includes the following stages:
 - build automation
 - continuous integration,
 - test automation
 - deployment automation

Build Pipeline Plugin Installation



Filter: Build Pipeline Plugin

Enabled	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	bouncycastle API Plugin This plugin provides an stable API to Bouncy Castle related tasks.	2.16.2		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"/>	jQuery plugin This plugin provides an stable version of jQuery Javascript Library	1.12.4-0		Uninstall
<input checked="" type="checkbox"/>	Parameterized Trigger plugin	2.35.2		Uninstall
<input checked="" type="checkbox"/>	Pipeline: Supporting APIs Common utility implementations to build Pipeline Plugin	2.17		Uninstall



Create Pipeline Build View

Create Pipeline Build View

AllDevOps+

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		DevCompile	9 hr 46 min - #18	19 hr - #5	27 sec	
		DeveCodeReview	9 hr 45 min - #13	N/A	30 sec	
		job1	2 days 17 hr - #1	N/A	0.45 sec	
		job2	2 days 16 hr - #1	N/A	0.12 sec	
		job3	N/A	N/A	N/A	
		job4	22 hr - #5	1 day 0 hr - #3	9.4 sec	
		QAMetricCheck	8 hr 31 min - #14	13 hr - #2	49 sec	
		QAPackage	47 min - #14	N/A	27 sec	
		QAUnitTest	9 hr 45 min - #14	13 hr - #3	27 sec	

Icon: [S](#) [M](#) [L](#)

Legend

[RSS for all](#)

[RSS for failures](#)

[RSS for just latest builds](#)

Click here to create a Pipeline View

Configuring Pipeline View

View name

DevOps Pipeline

☐

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.

☐

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

☐

My View
This view automatically displays all the jobs that the current user has an access to.

OK



Trigger Options

Build Cards

Standard build card

Use the default build cards

Restrict triggers to most recent successful builds

☐ Yes ☒ No

Always allow manual trigger on pipeline steps

☐ Yes ☒ No

Display Options

No Of Displayed Builds

1

Row Headers

Just the pipeline number

Show just the build pipeline number

Column Headers

No header

Do not show any column headers

Refresh frequency (in seconds)

3

URL for custom CSS files

Console Output Link Style

Lightbox

OK

Apply

After setting the initial job and configuring it accordingly, pipeline is ready to run

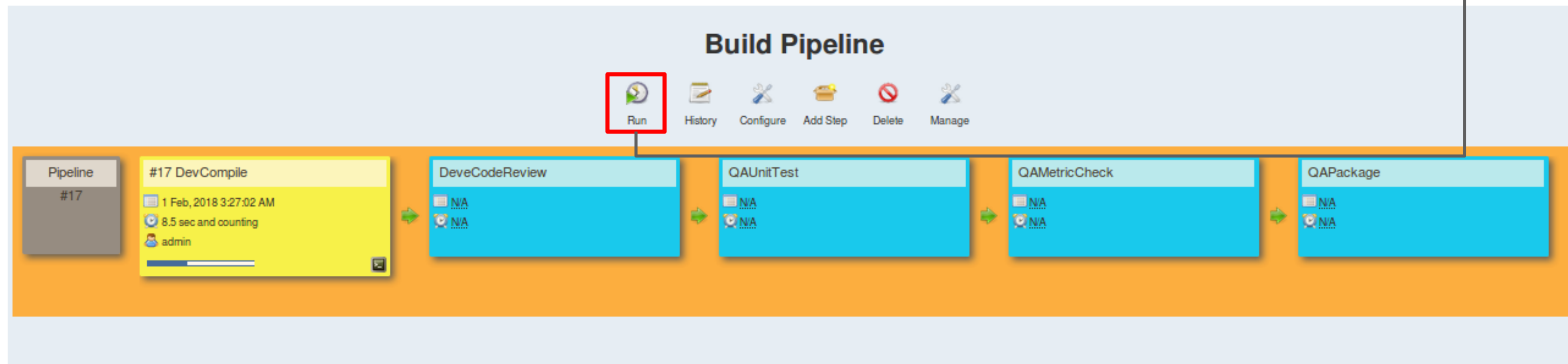
edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

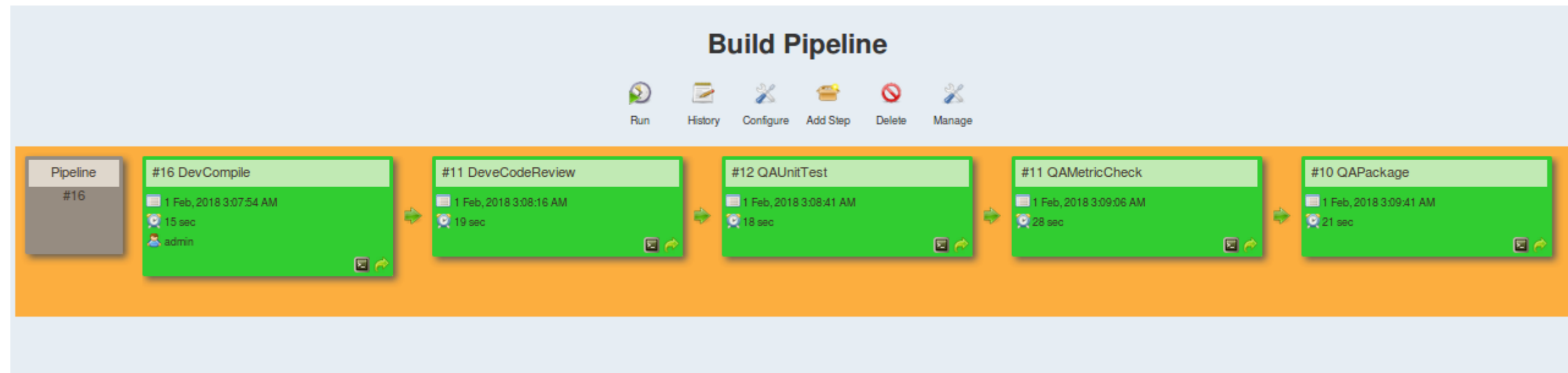
Complete Build Pipeline

- Yellow blocks represent the currently executing job
- Blue blocks represent the jobs waiting for execution
- Green blocks represent the successful build of job
- Red blocks represent failure of building of job

After clicking on the view that is created, you can run your pipeline build



Build Pipeline: Successful Build



Here is the pipeline of successfully
build jobs executed on Jenkins
server

Pipeline as a Code

Pipeline As Code

Pipeline as code provides functionalities like:

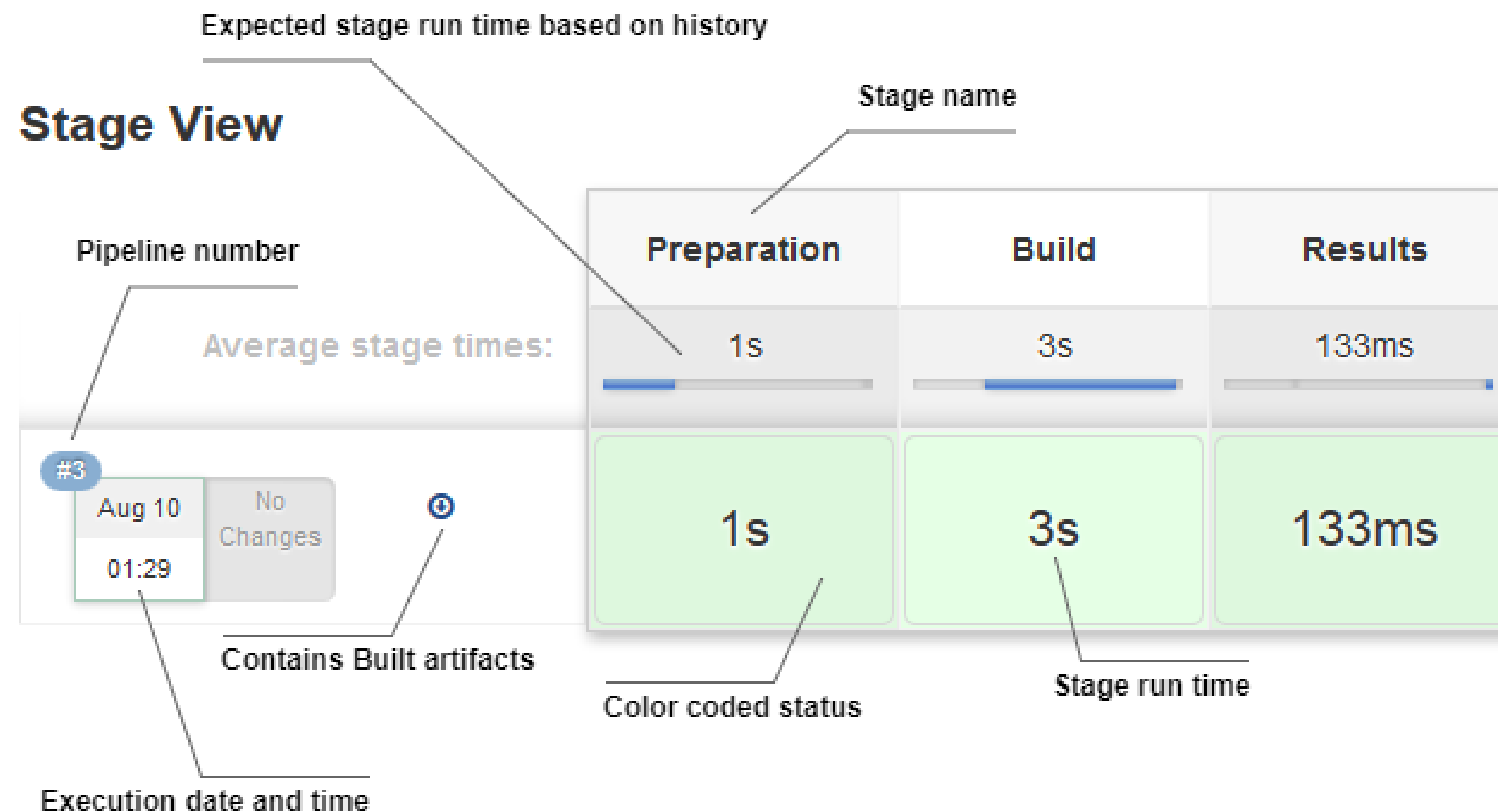
- Defining the pipeline flow into source control
- Suspend or resume executing jobs

Pipeline plugin enables users to implement the whole build, test and deploy pipeline in Jenkins and store their code, treating the pipeline as another piece of code in their source control repository.

Jenkins Pipeline Stage View

Jenkins Stage View helps to visualize the progress of various stages of the Pipeline in real time

The Stage View page will look something like the following screenshot:



Basic Structure of a Declarative Pipeline: Nodes

Nodes

- A node block is the Jenkins agent wherein stage blocks, directives, and steps should run
- The node block structure is as below:

node ('<parameter>') {<constituents>}

Defines: The node where the stage, directives, or steps should run

Constituents: Multiple stage blocks, directives, or steps

Required: Yes

Parameters: Any, label

Basic Structure of a Declarative Pipeline: Stages

Stages

- A stage block is a bundle of closely related steps and directives that have a common objective
- The stage block structure is as below:

stage ('<parameter>') {<constituents>}

Defines: A collection of steps and directives

Constituents: Multiple node blocks, directives, or steps

Required: Yes

Parameters: A string that is the name of the stage (mandatory)

Basic Structure of a Declarative Pipeline: Directives

- The purpose of directives are to suggest the node block, stage block, and steps by providing them with any of the following :
 - Options
 - Parameters
 - Environments
 - triggers, tools

Defines: The node where the stage should run

Constituents: Environments, options, parameters, triggers, tools

Required: No, but every CI/CD Pipeline has it

Parameters: None

Basic Structure of a Declarative Pipeline: Steps

- Fundamental block of a Pipeline
- A step could be:
 - batch script or a shell script, or
 - any other command that is executable
- Steps have various purposes, such as cloning a repository, building code, running tests, uploading artifacts to the repository server, performing static code analysis, and so on

Defines:- It tells Jenkins what to do

Constituents:- Commands, scripts, and so on

Required:- No, but every CI/CD Pipeline has it

Parameters:- None

Pipeline Syntax Overview

Declarative Pipeline fundamentals

In Declarative Pipeline syntax, there is concept of **Pipeline block** which defines all the work done throughout the entire Pipeline

Jenkinsfile (Declarative Pipeline)

```
pipeline {  
  agent any ❶  
  stages {  
    stage('Build') { ❷  
      steps {  
        // ❸  
      }  
    }  
    stage('Test') { ❹  
      steps {  
        // ❺  
      }  
    }  
    stage('Deploy') { ❻  
      steps {  
        // ❼  
      }  
    }  
  }  
}
```

Pipeline Syntax Overview

01

Execute the Pipeline or any of its stages, on any available agent

02

Define the "Build" stage

03

Perform some steps related to the "Build" stage

04

Defines the "Test" stage

Pipeline Syntax Overview

05

Perform some steps related to the "Test" stage

06

Define the "Deploy" stage

07

Perform some steps related to the "Deploy" stage

Pipeline Syntax Overview

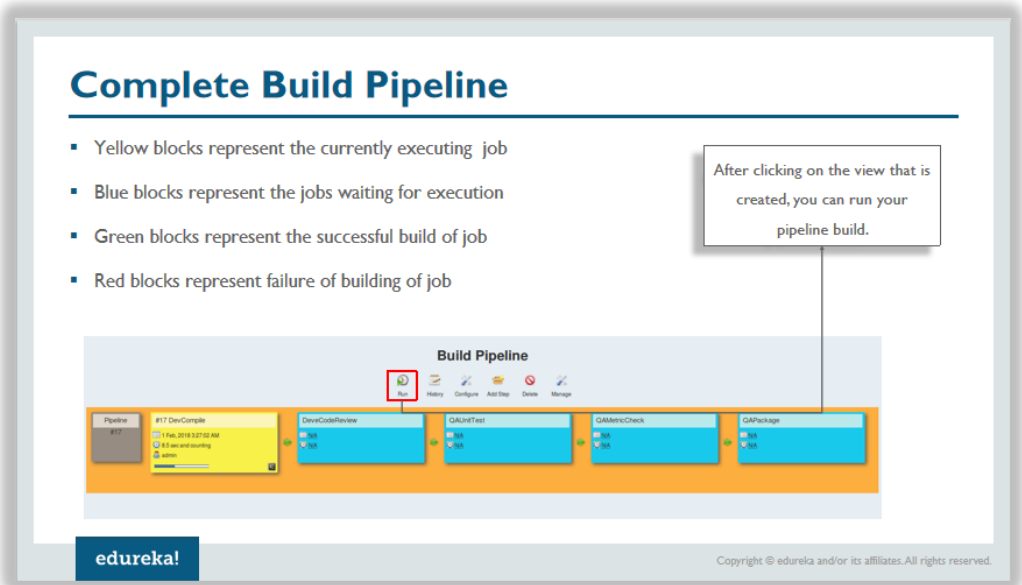
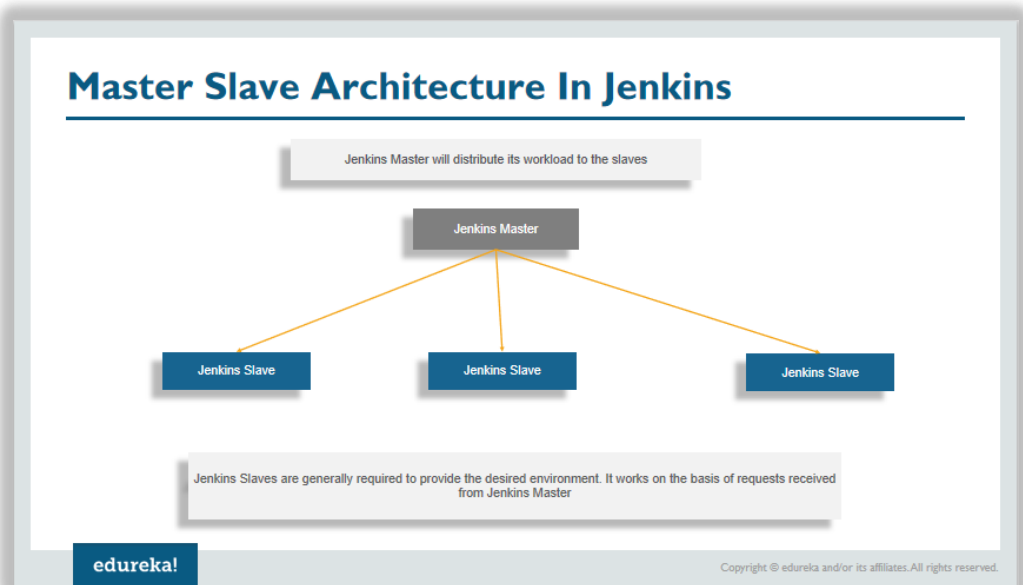
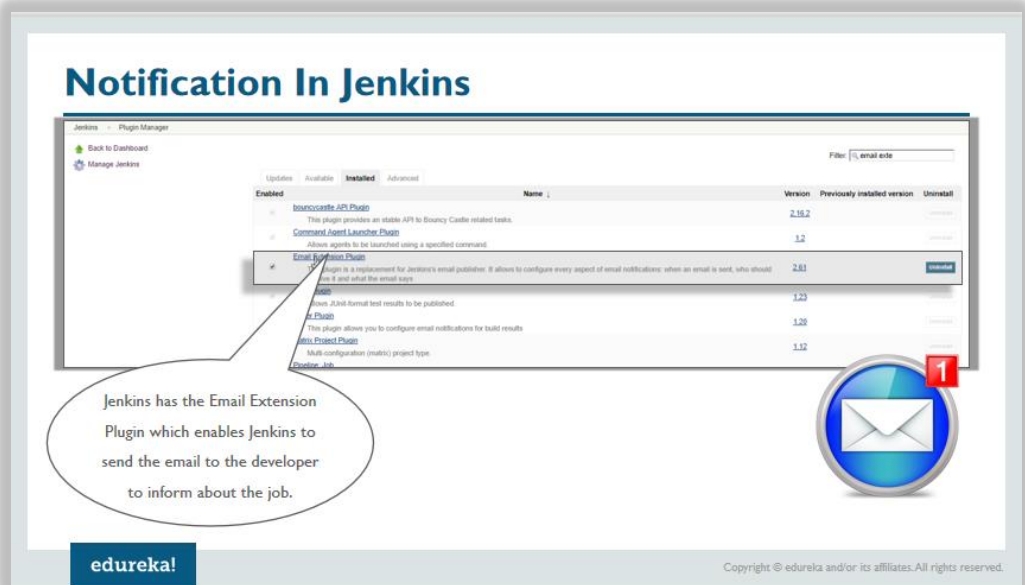
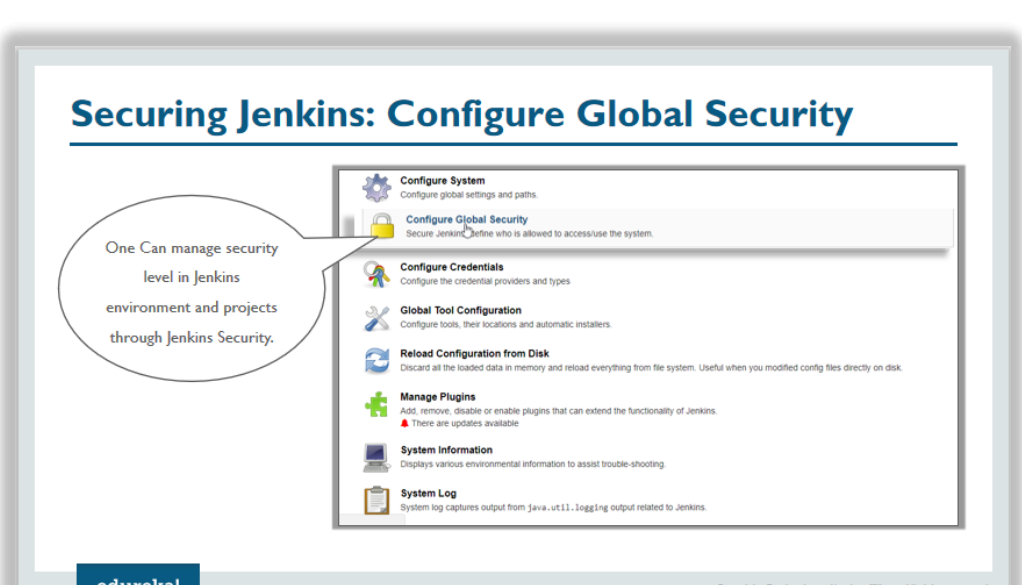
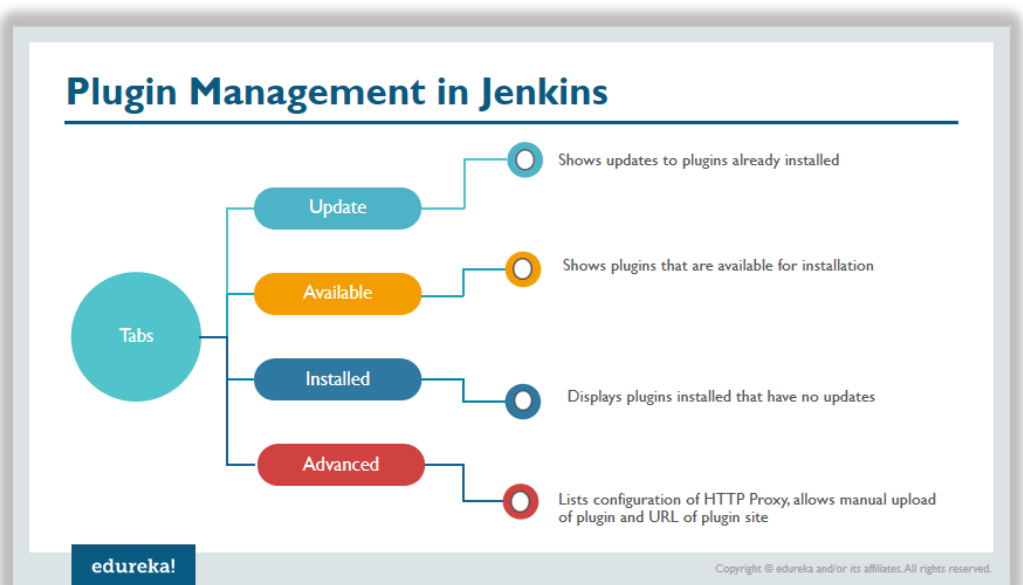
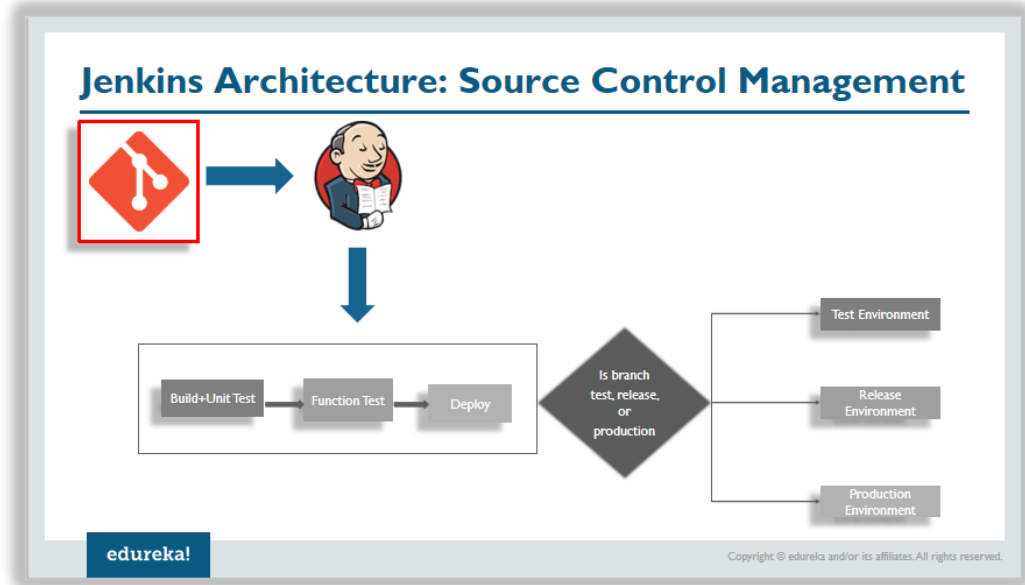
Scripted Pipeline fundamentals

- In a Scripted Pipeline, the node block does core work for the entire Pipeline
- Adding the Pipeline's work inside a node block does two things:
 - Schedules the steps contained within the block to run when an executor is free on a node
 - Creates a workspace where work can be done on files checked out from source control

Jenkinsfile (Scripted Pipeline)

```
node { ❶  
    stage('Build') { ❷  
        // ❸  
    }  
    stage('Test') { ❹  
        // ❺  
    }  
    stage('Deploy') { ❻  
        // ❼  
    }  
}
```


Summary



Questions



FEEDBACK





Thank You

For more information please visit our website
www.edureka.co