



Continuous Integration

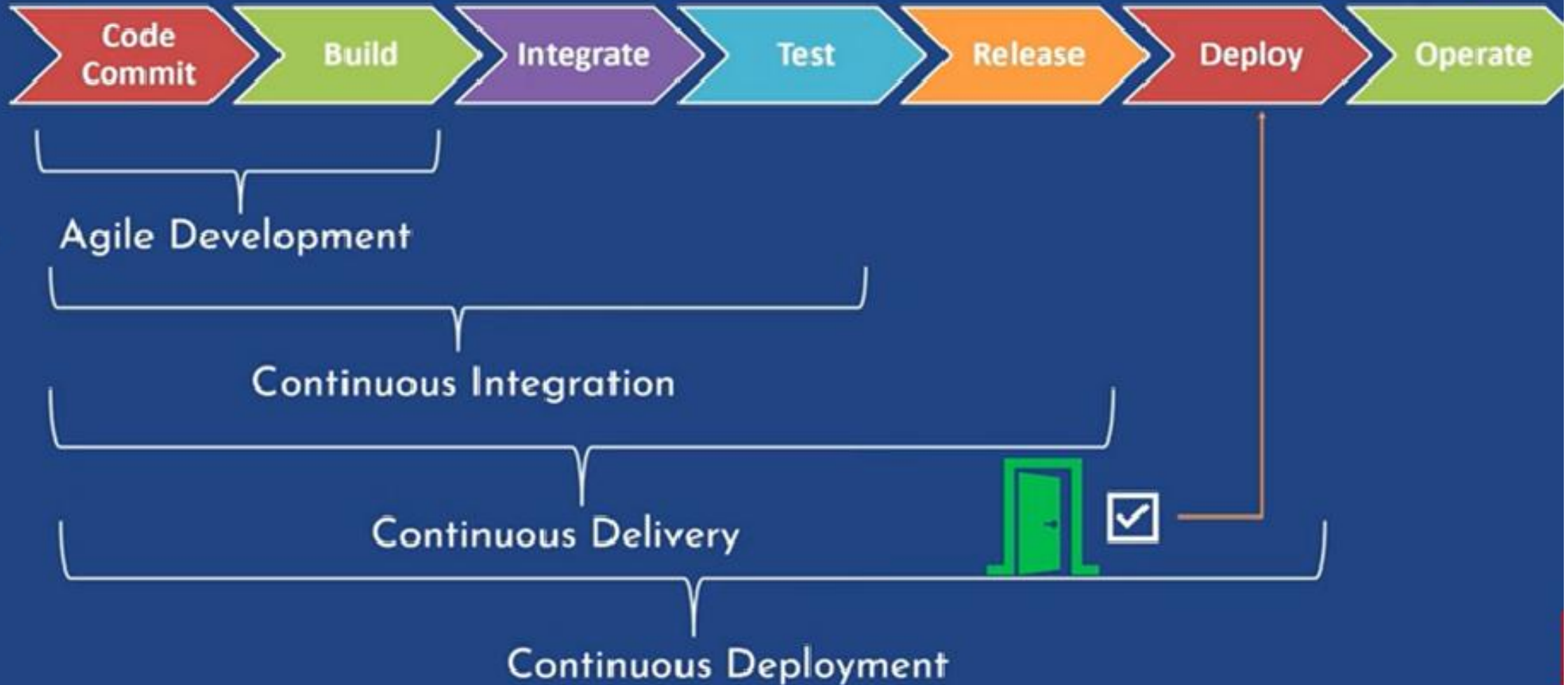
Introduction to Jenkins



Introduction to Jenkins

AGILE AND CONTINUOUS

Continuous Feedback at all stages



Aspect	Continuous Delivery	Continuous Deployment
Manual Approval	Yes	No
Automation Level	High, but final step is manual	Fully automated
Deployment Frequency	On-demand, after manual review & approval	Immediately, following each successful build
Risk Level	Lower, due to human oversight	Higher, mitigated by extensive automation
Typical Use Cases	Regulated industries, critical systems	Fast-paced, agile teams, non-critical updates

Introduction to Jenkins

- Jenkins is an open-source automation server used for –
 - continuous integration (CI) and
 - continuous delivery (CD) in software development.
- It helps developers automate the process of building, testing, and deploying applications.

Key Features of Jenkins

- **Automation:** Automates repetitive tasks like code compilation, testing, and deployment.
- **CI/CD Pipeline:** Supports building and integrating code frequently to detect issues early.
- **Plugins Support:** Over 1,800 plugins for integrating with various tools like Git, Docker, Kubernetes, Maven, etc.
- **Distributed Builds:** Can run builds on multiple machines to speed up execution.

2

Jenkins Installation

Jenkins Installation on Windows – setup

- The simplest way to install Jenkins on Windows is to use the Jenkins Windows installer (.msi file).
- That program will install Jenkins as a service using a 64-bit JVM chosen by the user.
- Keep in mind that to run Jenkins as a service, the account that runs Jenkins must have permission to login as a service.

Jenkins Installation on Windows – setup

- Prerequisites
 - Minimum hardware requirements:
 - 256 MB of RAM
 - 1 GB of drive space (although 10 GB is a recommended minimum if running Jenkins as a Docker container)
 - Recommended hardware configuration for a small team:
 - 4 GB+ of RAM
 - 50 GB+ of drive space

Jenkins Installation on Windows – setup

- Prerequisites
 - Software requirements:
 - Java (Java 17 or Java 21)
 - Any Web browser (Updated)

Jenkins Installation on Windows – setup

1

<https://www.jenkins.io/download/>

Blog Success St

Download and deploy

The Jenkins project produces two release lines: Stable (LTS) and weekly. Depending on your organi recommendations about the release lines.

Stable (LTS)

Long-Term Support (LTS) release baselines are chosen every 12 weeks from the stream of regular releases. Every 4 weeks we release stable releases which include bug and security fix backports.

[Learn more...](#)

2

Kubernetes

Ubuntu/Debian

Red Hat Enterprise Linux and derivatives

Fedora

Windows

openSUSE

Click here to download the .msi file for installation of Jenkins.

Running Jenkins using PowerShell to install default plugins

localhost:8090/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

1

Copy the given path and paste it in file explorer to open a Notepad file wherein a password for login is present.

2

Copy the password and paste it here.

Running Jenkins using PowerShell to install default plugins

localhost:8090

Getting Started

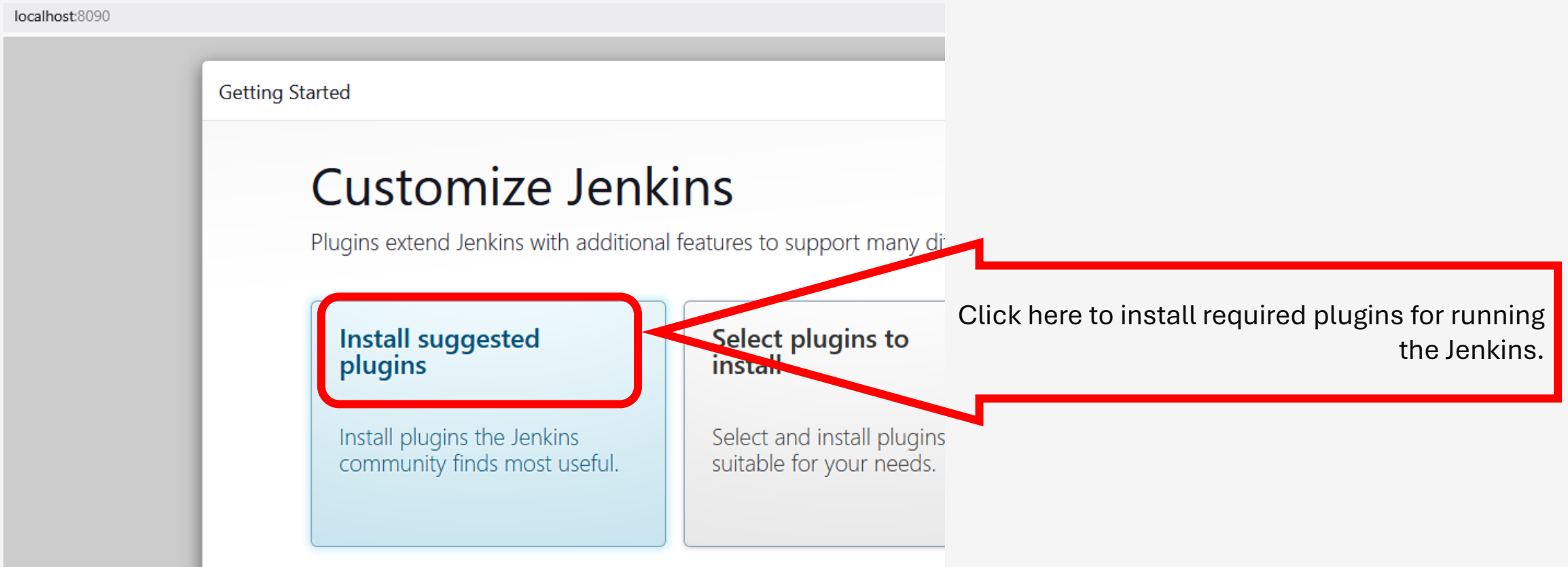
Customize Jenkins

Plugins extend Jenkins with additional features to support many different use cases.

Install suggested plugins
Install plugins the Jenkins community finds most useful.

Select plugins to install
Select and install plugins suitable for your needs.

Click here to install required plugins for running the Jenkins.

The image shows a screenshot of the Jenkins web interface at localhost:8090. The page is titled 'Getting Started' and 'Customize Jenkins'. It explains that plugins extend Jenkins with additional features. There are two main options: 'Install suggested plugins' (highlighted with a red box) and 'Select plugins to install'. A red callout box points to the 'Install suggested plugins' button with the text 'Click here to install required plugins for running the Jenkins.'

Running Jenkins using PowerShell to install default plugins

localhost:8090

Getting Started


Getting Started

<input checked="" type="checkbox"/> Folders	<input checked="" type="checkbox"/> OWASP Markup Formatter	<input type="checkbox"/> Build Timeout	<input type="checkbox"/> Credentials Binding	<div>** Ionicons API Folders OWASP Markup Formatter ** ASM API ** JSON Path API ** Struts</div>
<input type="checkbox"/> Timestampers	<input type="checkbox"/> Workspace Cleanup	<input type="checkbox"/> Ant	<input type="checkbox"/> Gradle	
<input type="checkbox"/> Pipeline	<input type="checkbox"/> GitHub Branch Source	<input type="checkbox"/> Pipeline: GitHub Groovy Libraries	<input type="checkbox"/> Pipeline Graph View	
<input type="checkbox"/> Git	<input type="checkbox"/> SSH Build Agents	<input type="checkbox"/> Matrix Authorization Strategy	<input type="checkbox"/> PAM Authentication	
<input type="checkbox"/> LDAP	<input type="checkbox"/> Email Extension	<input type="checkbox"/> Mailer	<input type="checkbox"/> Dark Theme	

Jenkins Dashboard

Dashboard [Jenkins]

localhost:8090

 Jenkins

Admin At FAMT

log out

Dashboard >

+ New Item

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

A red circle containing the white number 3, indicating the third step in a sequence.

Setting up a CI/CD pipeline for a web development

Setting up a demo pipeline

The screenshot illustrates the Jenkins web interface for creating a new pipeline. It is divided into three numbered steps:

- Step 1:** The user is on the Jenkins Dashboard. A red box highlights the **+ New Item** button in the left-hand navigation menu.
- Step 2:** The **New Item** dialog is open. A red box highlights the **Enter an item name** field, which contains the text **demoPipeline**. Another red box highlights the **Pipeline** item type, described as "Orchestrates long-running activities that can span multiple build agents, workflows) and/or organizing complex activities that do not easily fit in freestyle projects." A third red box highlights the **OK** button at the bottom of the dialog.
- Step 3:** The **Configure** page for the new pipeline is shown. The **General** tab is selected in the left sidebar. On the right, under the **General** section, there are checkboxes for **Discard old builds**, **Do not allow**, **Do not allow**, **GitHub project**, and **Pipeline**.

Setting up a demo pipeline

Scroll down the
“Configure” page up to
4 “Pipeline” section to
write a script for the
demo pipeline.

Scroll down to click
5 “Save” button to save
the pipeline.

Script ?

```
1 pipeline {  
2   agent any // Defines where the pipeline runs  
3  
4   stages {  
5     stage('Build') { // Defines a step in the pipeline  
6       steps {  
7         echo 'Building the project...' // Print message to console  
8       }  
9     }  
10  
11    stage('Test') {  
12      steps {  
13        echo 'Running tests...'  
14      }  
15    }  
16  
17    stage('Deploy') {  
18      steps {  
19        echo 'Deploying the application...'  
20      }  
21    }  
22  }  
23  
24  post {  
25    success {  
26      echo 'Pipeline completed successfully!' // Runs if the pipeline is successful  
27    }  
28    failure {  
29      echo 'Pipeline failed!' // Runs if any stage fails  
30    }  
31  }  
32 }
```

Setting up a demo pipeline

← → ↻ localhost:8090/job/demoPipeline/

Jenkins

Dashboard > demoPipeline >

6

Status demoPipeline

</> Changes

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

Permalinks

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

If you want to edit pipeline again then use "Configure"

7

Builds

Filter

Today

✓	#2	5:59 PM	▼
✓	#1	5:59 PM	▼

Setting up a demo pipeline

- </> Changes
- ▶ Build Now
- ⚙️ Configure
- 🗑️ Delete Pipeline
- 📁 Stages
- ✎️ Rename
- ❓ Pipeline Syntax

Builds

Filter

Today

- ✓ #2 5:59 PM
- ✓ #1 5:59 PM

9

- </> Changes
- 📄 Console Output
- ✎️ Edit Build Information
- 🗑️ Delete build '#2'
- 🕒 Timings
- 🔗 Pipeline Overview
- 📄 Pipeline Console
- 🔄 Restart from Stage
- 🔄 Replay
- 📋 Pipeline Steps
- 📁 Workspaces

8

- 📄 Status
- </> Changes
- 📄 Console Output
- ✎️ Edit Build Information
- 🗑️ Delete build '#2'
- 🕒 Timings
- 🔗 Pipeline Overview
- 📄 Pipeline Console
- 🔄 Restart from Stage
- 🔄 Replay
- 📋 Pipeline Steps
- 📁 Workspaces
- ← Previous Build

✓ Console Output

```
Started by user Admin At FANT
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\demoPipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Building the project...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] echo
Running tests...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] echo
Deploying the application...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline completed successfully!
[Pipeline] }
[Pipeline] // stage
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```