

# DevOps Foundations: Version Control and CI/CD with Jenkins



## CI/CD Pipeline with Jenkins



# Learning Objectives

By the end of this lesson, you will be able to:

- 🔗 Outline the fundamental concepts of CI/CD to deliver high-quality software consistently, efficiently, and with reduced risk
- 🔗 Identify the appropriate CI/CD tools for optimizing the software development process, ensuring compatibility, scalability, and efficiency
- 🔗 Define the features and benefits of Jenkins for optimizing development process and achieving reliable software delivery
- 🔗 Illustrate the Jenkins architecture and its support for various platforms for optimizing resource management in CI/CD pipelines



# Learning Objectives

By the end of this lesson, you will be able to:

- 🔗 Illustrate how to create new users in Jenkins to enhance security, enable customization, and promote collaboration among team members
- 🔗 Explore the Jenkins UI to effectively utilize its features and administer user accounts and permissions

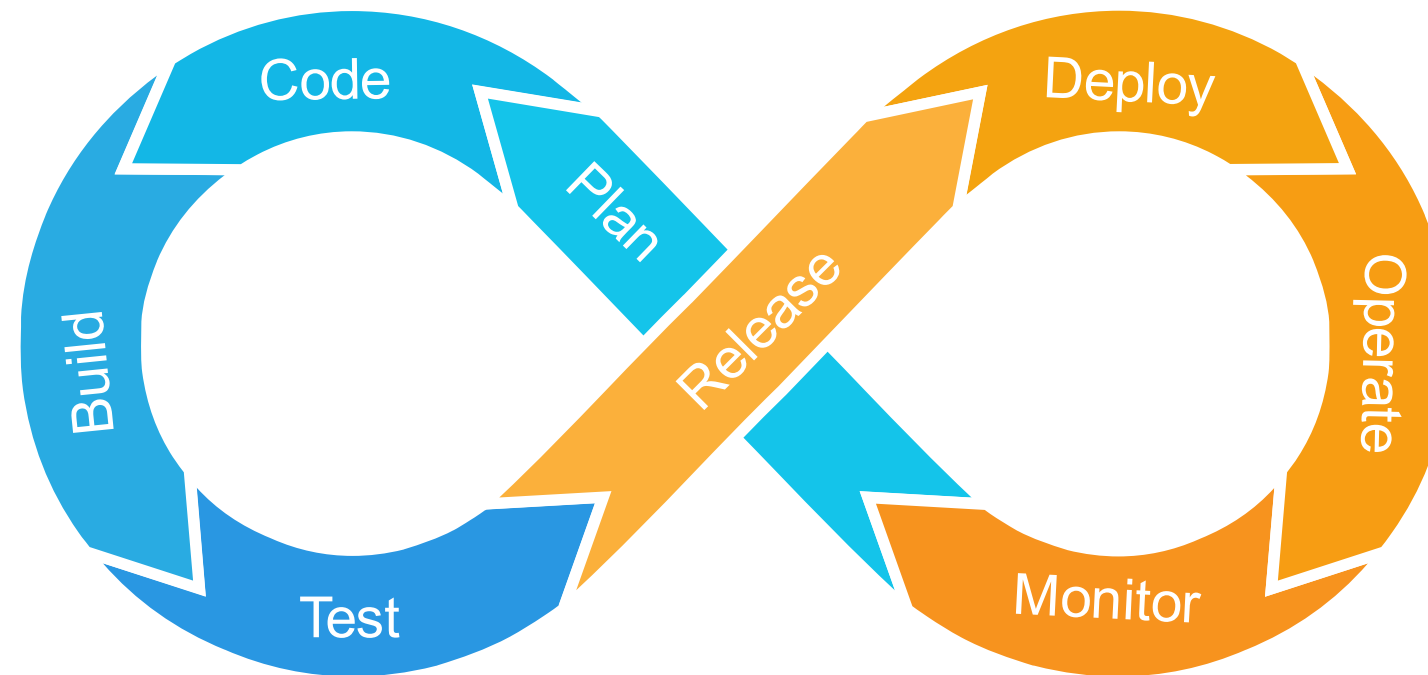




# Getting Started with CI/CD Pipeline

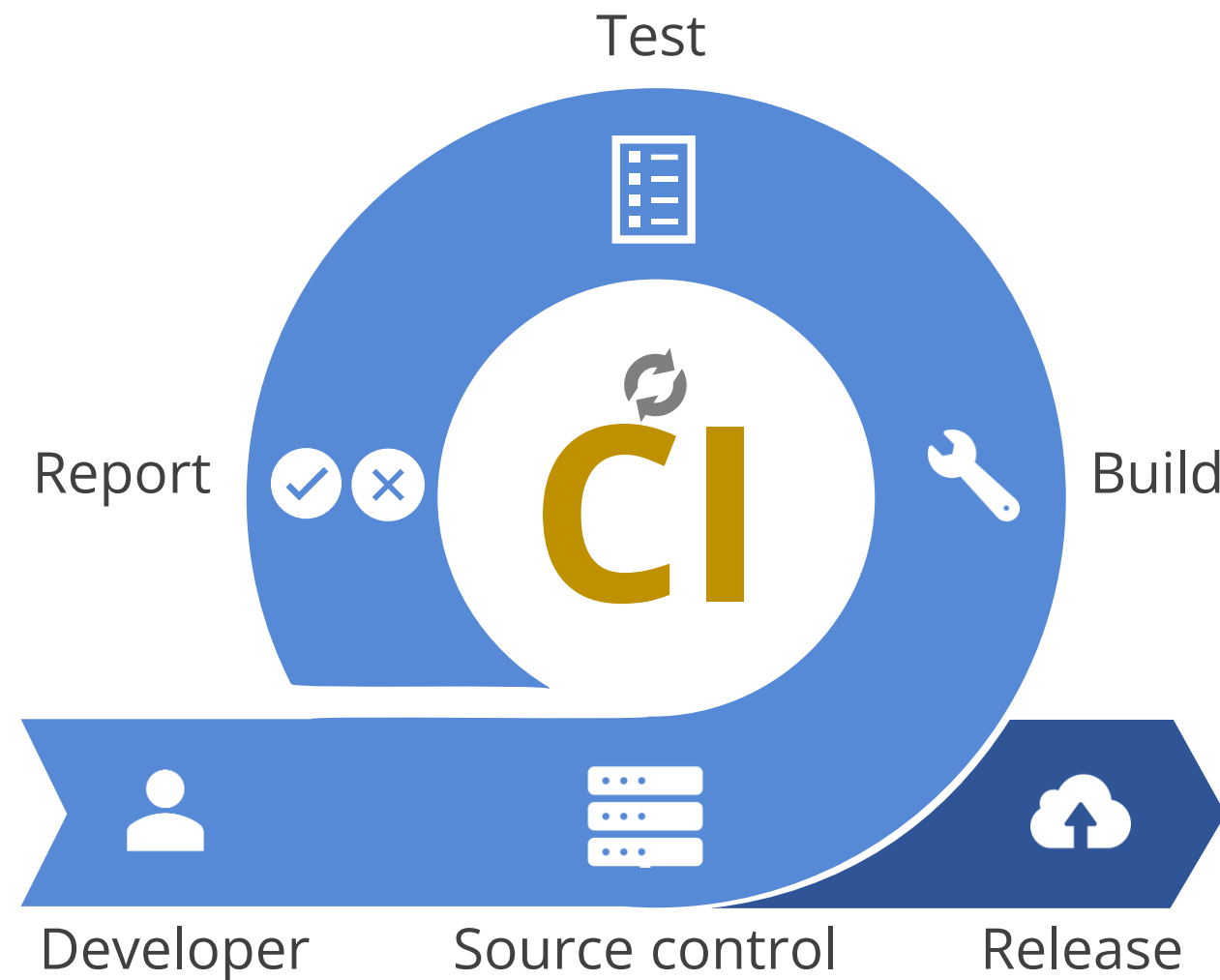
# What Is a CI/CD Pipeline?

CI/CD pipeline stands for continuous integration and continuous delivery pipeline. It is essentially an automated workflow that streamlines the software development process by automating various stages.



# Continuous Integration

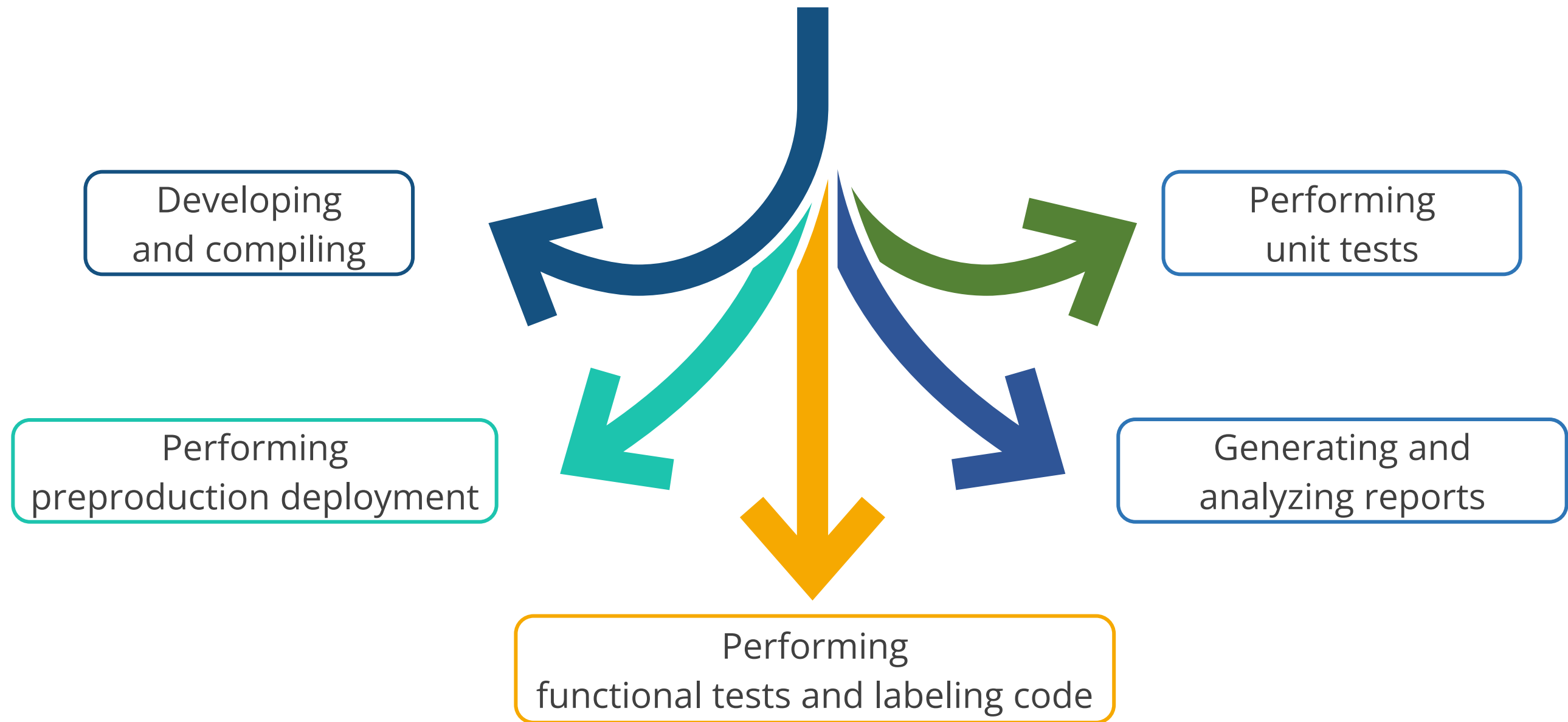
Continuous Integration (CI) is a development practice in which all development work is integrated as early as possible. It emphasizes frequent integration of code changes into a central repository.



It involves automating the build, testing, and verifying the code whenever a change is made.

# Tasks Involved in Continuous Integration

Following is a description of the tasks involved in continuous integration:





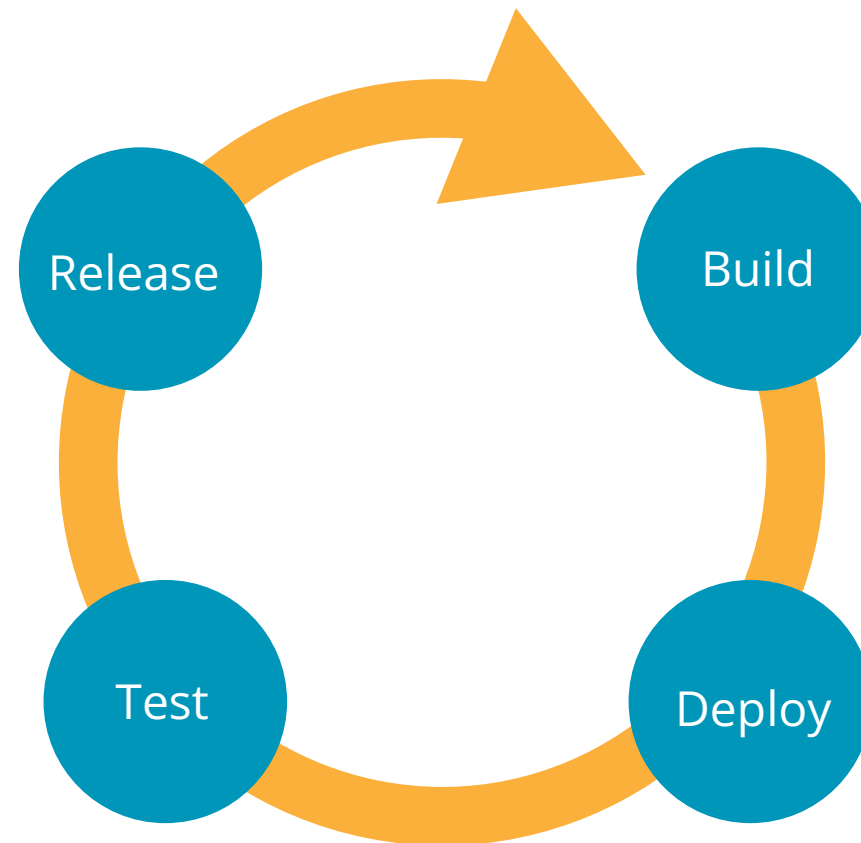
# Advantages of Continuous Integration

Continuous integration offers several advantages, including:

- It gives a clear snapshot of the ongoing development work when automated end-to-end acceptance tests are conducted.
- It results in fewer bugs and quicker delivery when CI tools are used, as they are designed to identify and fix integration and regression issues faster.
- It is the automation of the deployment process that provides testers and end users with quick access to the software.
- It simplifies and accelerates delivery when the deployment process is automated by CI.

# Continuous Delivery

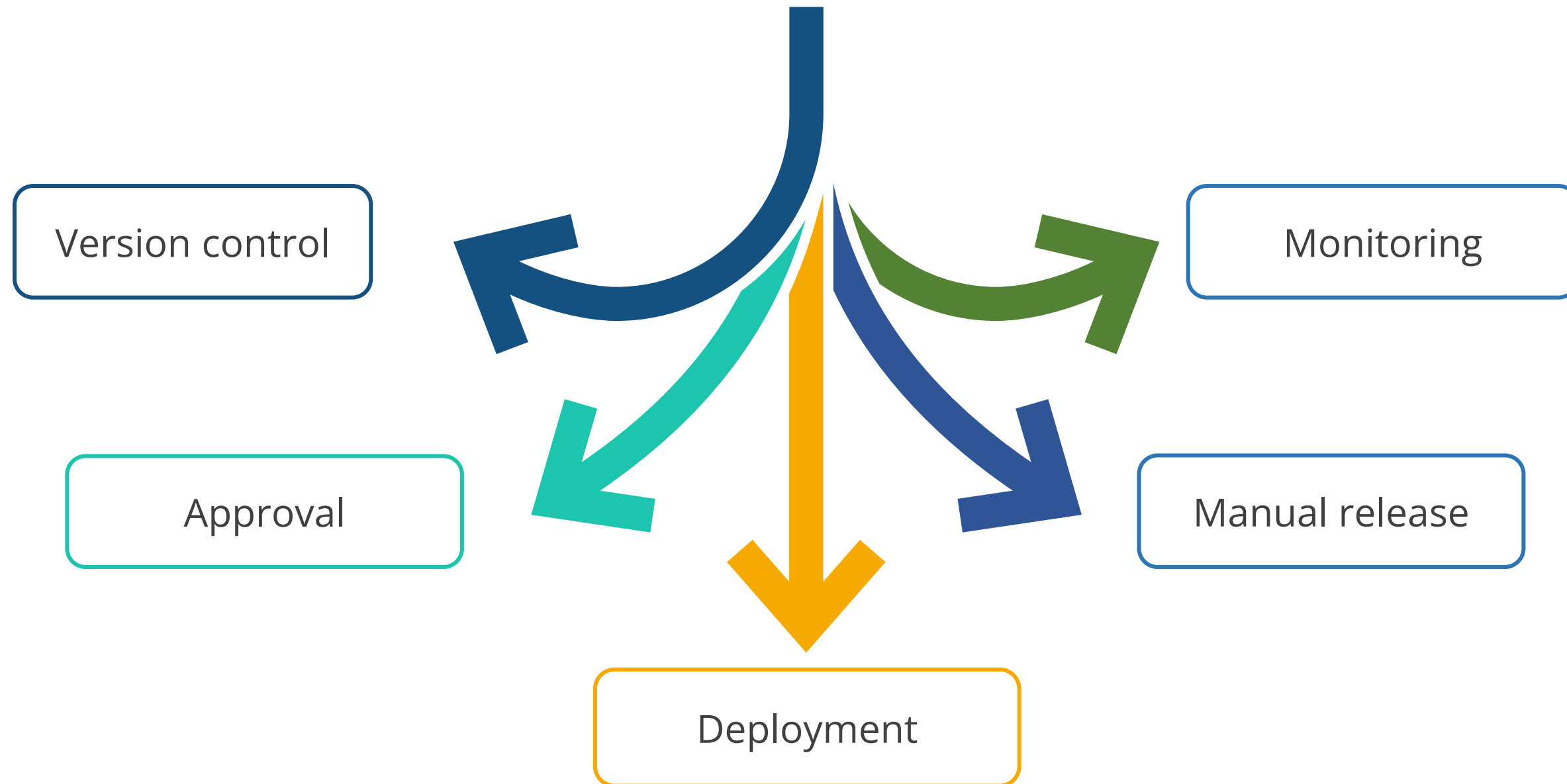
Continuous Delivery (CD) is an extension of continuous integration. It automates the entire software release process ensuring that the code changes are always ready for deployment to production.



This DevOps practice significantly reduces the risk and time associated with the traditional deployments.

# Tasks Involved in Continuous Delivery

The following automated tasks are carried out under continuous delivery that streamlines the software release process:



# Advantages of Continuous Delivery

CD boosts your team's productivity and code quality by automating processes and delivering faster updates to customers. Here are some advantages of continuous delivery:

- It allows developers to develop and deploy high-quality software at a fast pace.
- It helps proactively resolve issues and ensure more stable production releases.
- It makes release processes as efficient and repeatable as possible.
- It boosts DevOps return on investment by making manual processes easily repeatable.

# Continuous Deployment

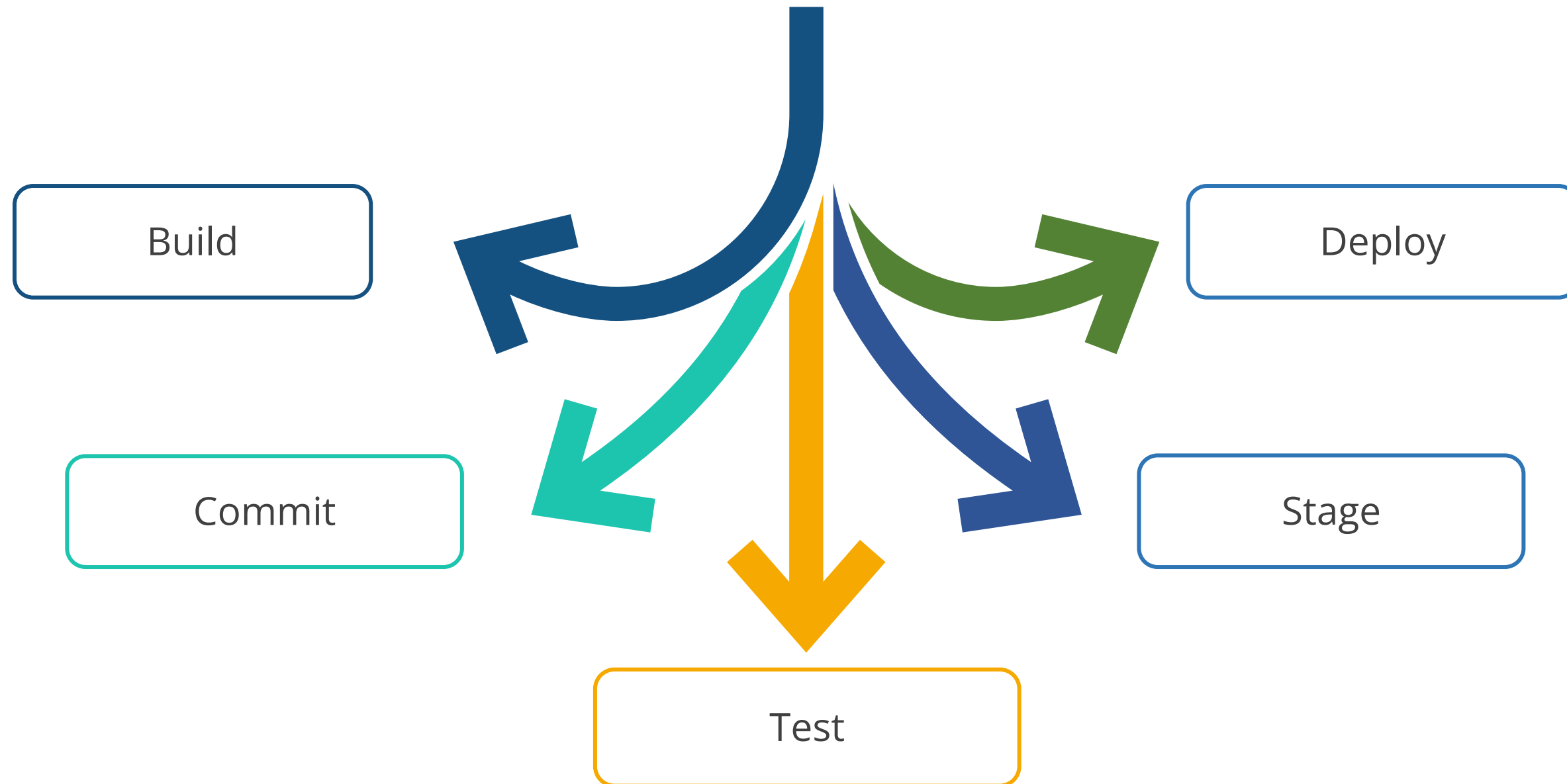
Continuous Deployment (CD) is an extension of continuous integration. It aims to reduce the time between the development team writing a new line of code and its use in production.



It essentially advances continuous delivery by removing the requirement for manual approval before updates are pushed live.

# Tasks Involved in Continuous Deployment

The following tasks are carried out to build a robust and automated pipeline with a strong focus on testing and monitoring to ensure smooth and reliable delivery of code changes to production:



# Advantages of Continuous Deployment

Continuous deployment simplifies the release process and offers the following advantages:

## **Fast delivery**

It automatically deploys the changes right after development.



## **Fast feedback cycle**

It maintains stability by quickly identifying and resolving bugs.

## **Low-risk releases**

It becomes safer and repeatable with daily releases.



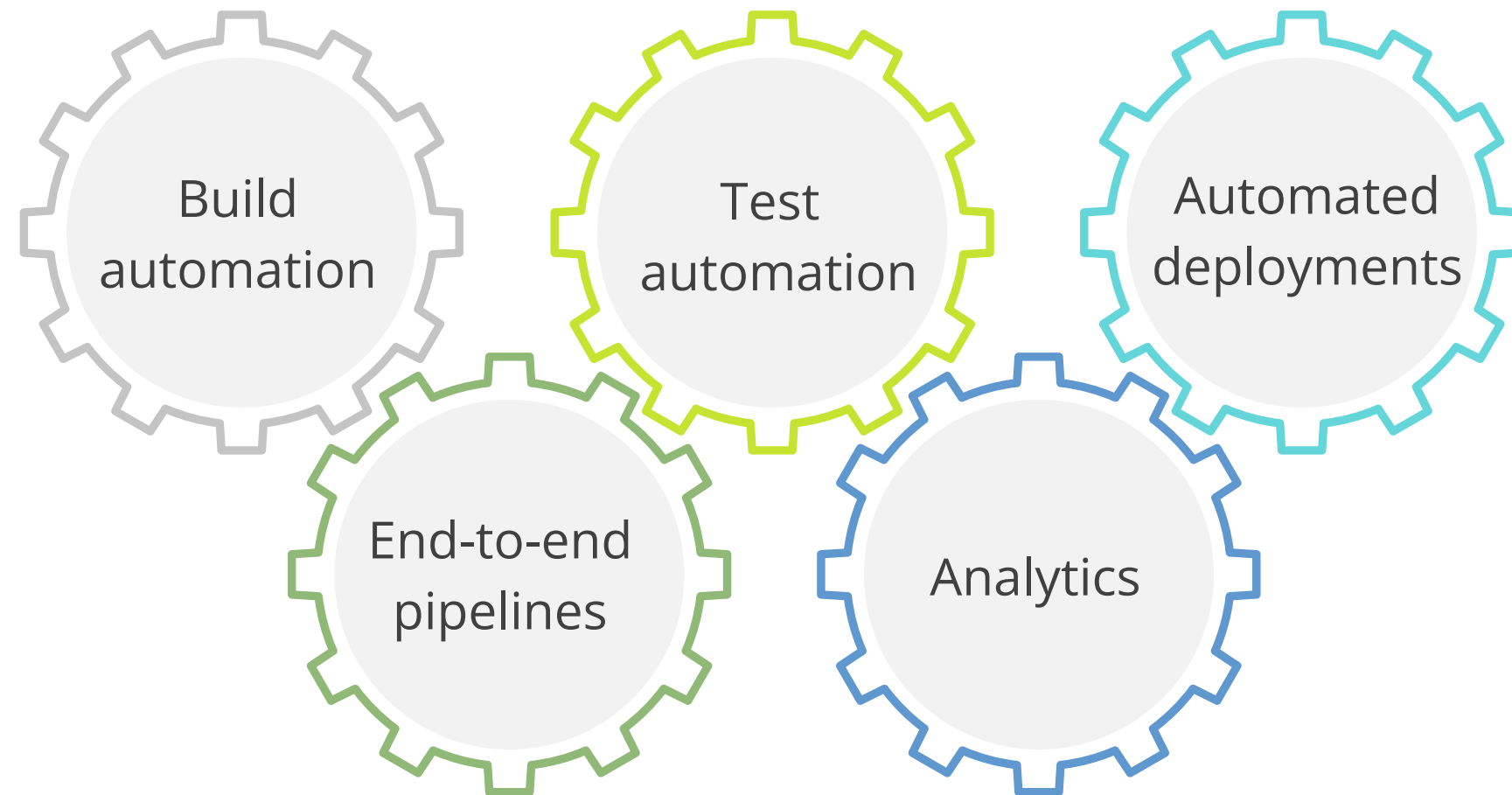
## **Flexible release options**

It enables cost-free, instant software deployment.

# Importance of CI/CD Pipeline

CI/CD enables organizations to develop software quickly and efficiently.

Here are the main benefits of implementing CI/CD pipelines:





# CI/CD Workflow

A CI/CD workflow is essentially an automated process that includes various stages. Following is an outline of the usual workflow:

## Source stage

It entails the organization and preservation of source code in a regulated and versioned way.

## Build stage

It is the process where new code is combined with the existing code to create deployable artifacts.

## Test stage

It refers to the process where automated tests are executed to verify the code and identify any bugs.

## Deploy stage

It refers to the process where the code is deployed to either staging or production environments.

# Elements of a CI/CD Pipeline

CI/CD elements can enhance DevOps workflow and software delivery, spanning development to deployment. The following are the fundamental elements of a CI/CD pipeline:

## A single source repository

A SCM repository that includes all files and scripts for builds, such as source code, libraries, version control, and build scripts

## Frequent check

Frequent merging of small code segments into the main branch, avoiding multiple sub-branches and simultaneous changes

## Automated builds

The process of scripting and automating the retrieval of software code from a repository, compiling it into a binary artifact

## Self-testing builds

The incorporation of pre-build scripts to validate code integrity before initiating builds

# Elements of a CI/CD Pipeline

The following are the fundamental elements of a CI/CD pipeline:

## Frequent iterations

Frequent iterations of small commits reduce conflict hideouts. This allows easy rollbacks in case of issues.

## Stable testing environments

The emphasis is on testing code in a production clone rather than in live environments to identify any missed bugs before deployment.

## Maximum visibility

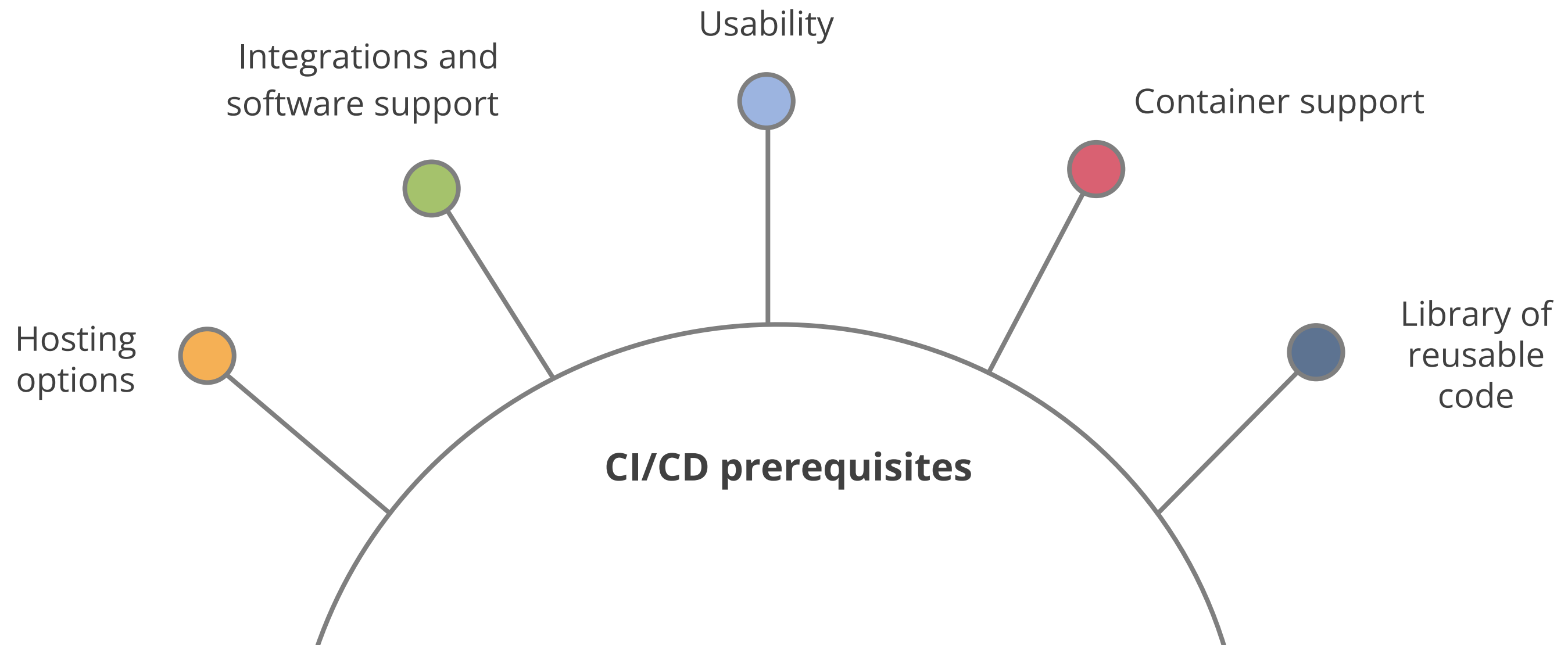
Developers should access the latest executables and repository changes.

## Predictable deployments

Deployments ought to be routine, low-risk, and reliable, allowing for updates at any time.

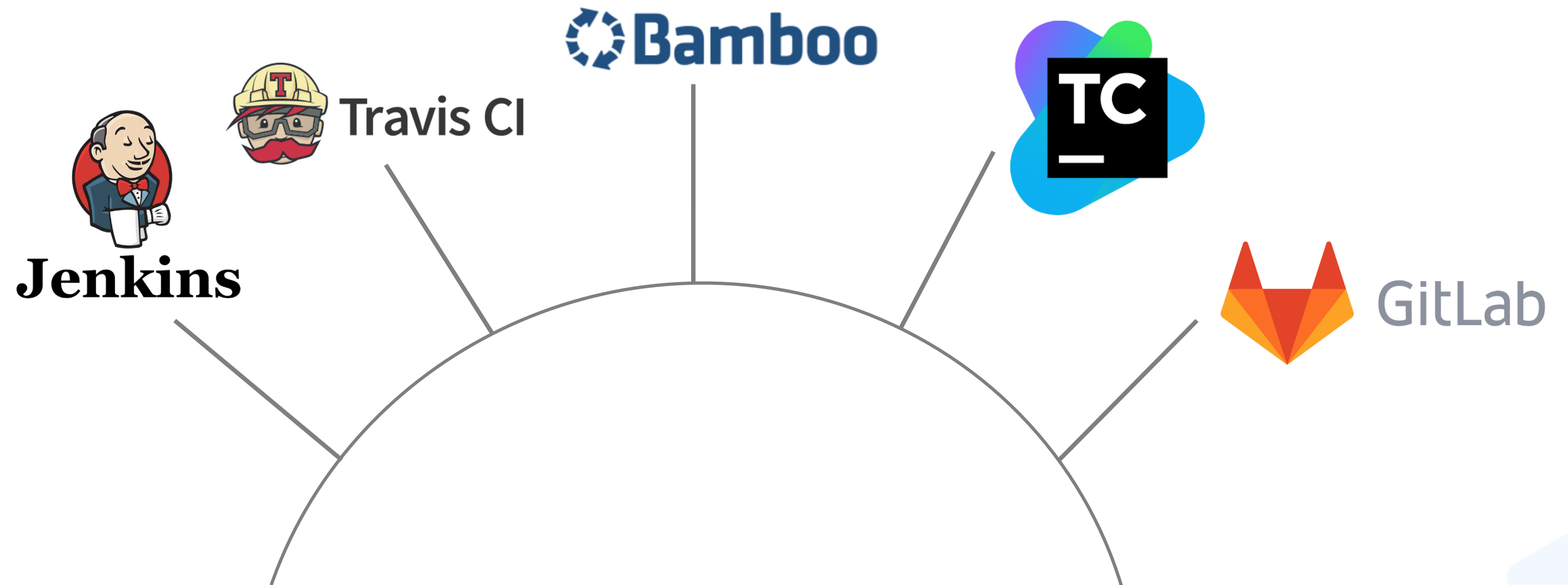
# CI/CD Tool Selection

The following is the list of parameters that must be considered while selecting CI/CD tools:



# CI/CD Tools

Following is the list of the popular tools available for building CI/CD pipelines:



## Quick Check



As a software development team lead, you are tasked with explaining the concept of a CI/CD pipeline to a new developer joining your team. How would you describe a CI/CD pipeline and its significance in the software development process?

- A. A physical pipeline used in construction projects for transporting materials
- B. A sequence of automated processes for building, testing, and deploying software changes
- C. A method of organizing workflow in a development environment
- D. A pipeline transporting oil and gas products across different regions



# Introduction to Jenkins

# What Is Jenkins?

It is an automation tool that provides a flexible and robust platform for building, testing, analyzing, deploying, and monitoring software changes. Here are some key points about Jenkins:



**Jenkins**

- It is an open-source project written in the Java programming language that facilitates continuous integration and continuous delivery (CI/CD) practices.
- It supports Windows, macOS, and other Unix-based operating systems.
- It is free, community-supported, and a popular first-choice tool for continuous integration.
- It is primarily used for on-premises deployment, but it can also be run on cloud servers.



# Why Use Jenkins?

It is used for creating and managing CI/CD pipelines, which includes the following:

## Continuous integration

It instantly tests Git commits, ensuring release readiness and early bug detection.

## Continuous delivery

It simplifies and automates software updates and feature rollouts.

## Build and automation

Its plugins and tool integrations automate tasks such as code testing, app packaging, and deployment.

## Pipeline orchestration

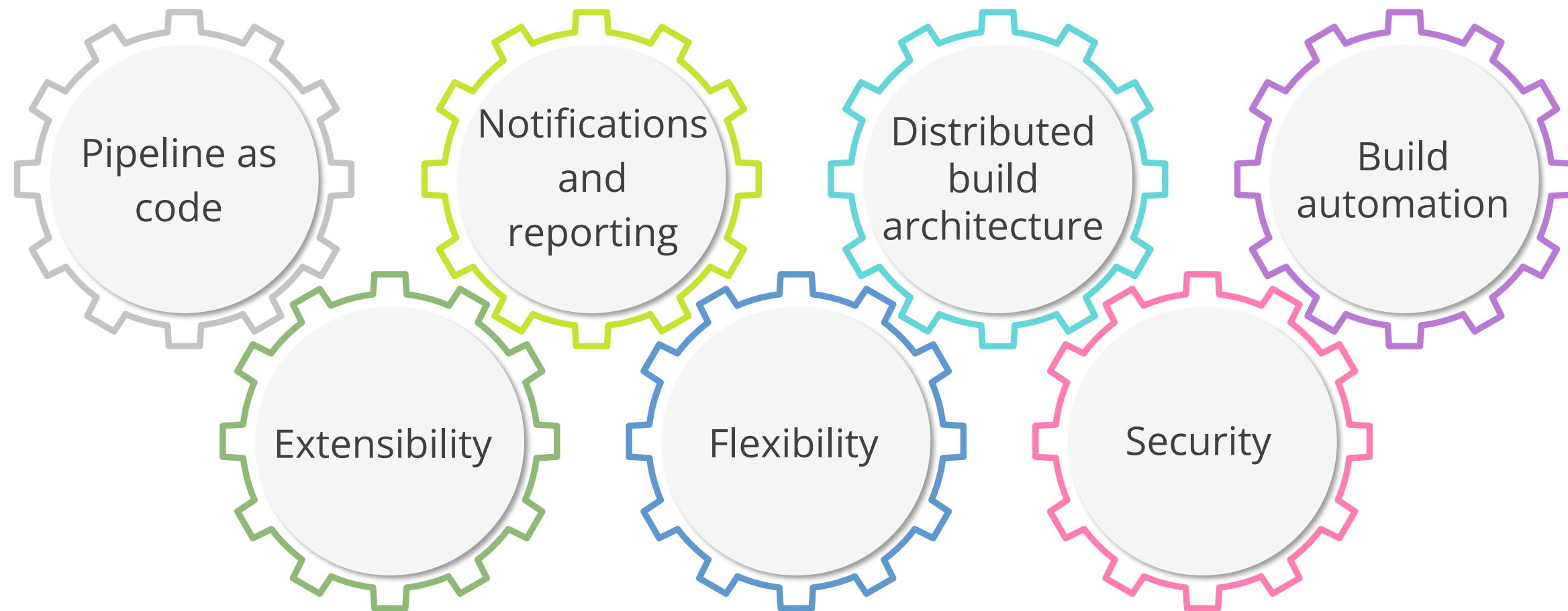
It manages CI/CD pipelines, simplifying software lifecycle oversight.

## Extensibility and customization

Its plugins allow user-specific customizations and tool integrations.

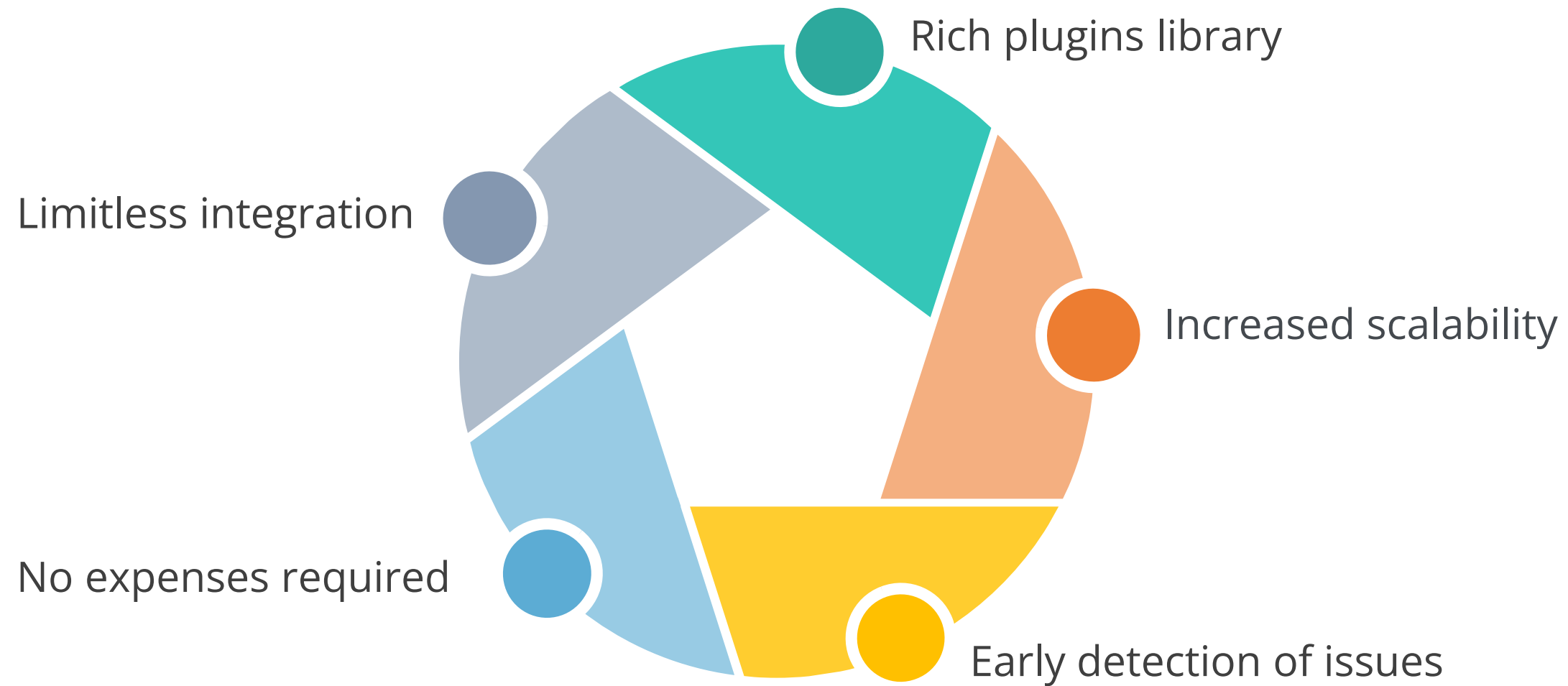
# Jenkins: Features

It is a versatile automation tool that offers a wide range of features, including:



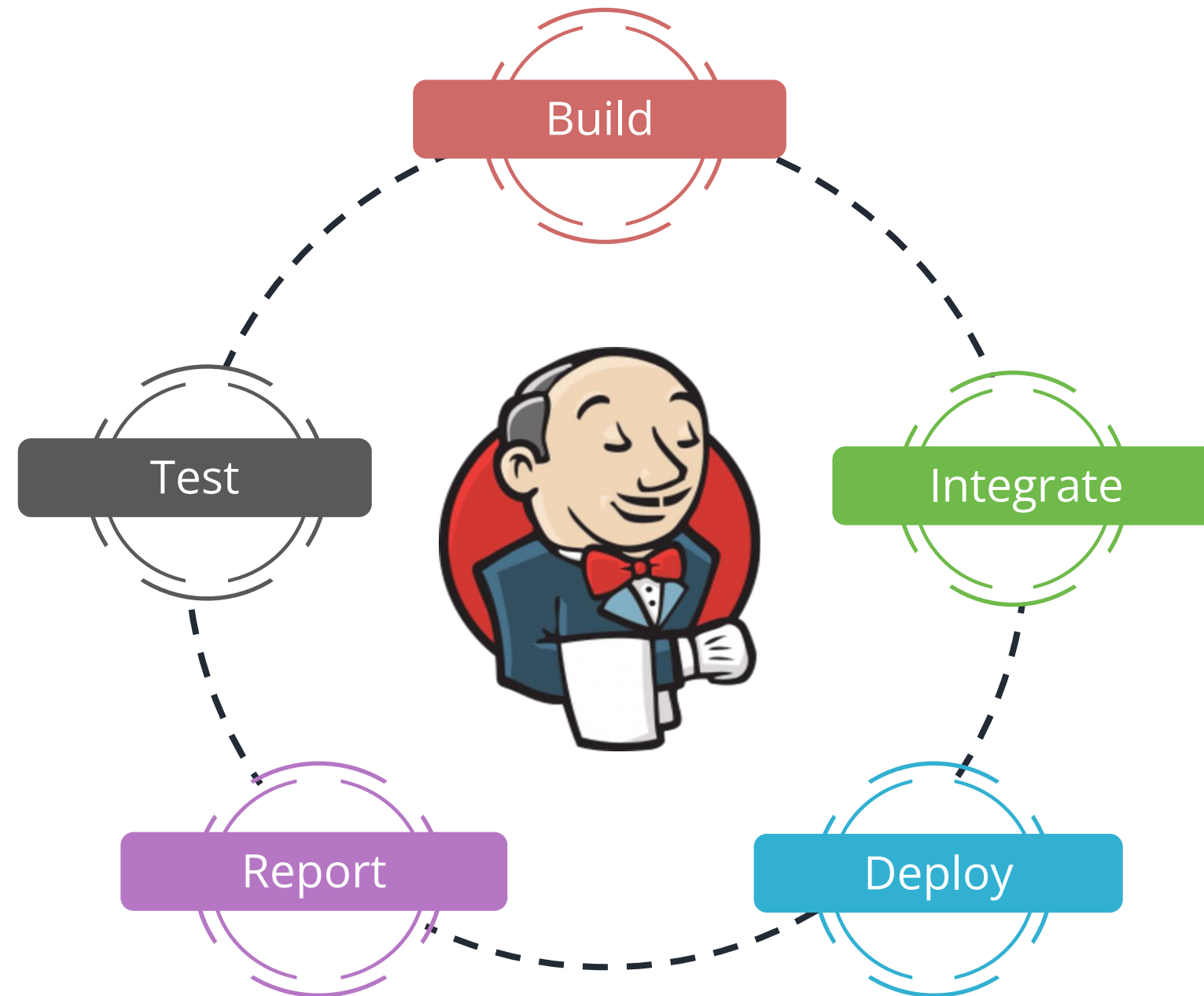
# Benefits of Jenkins

Here are some of the benefits of Jenkins:



# Role of Jenkins in DevOps

It is used to perform different tasks:



# How Does Jenkins Work?

Jenkins working can be disintegrated into the following points:

## **Triggering builds:**

Jenkins triggers a build when a commit is made to the development branch.

## **Integration testing:**

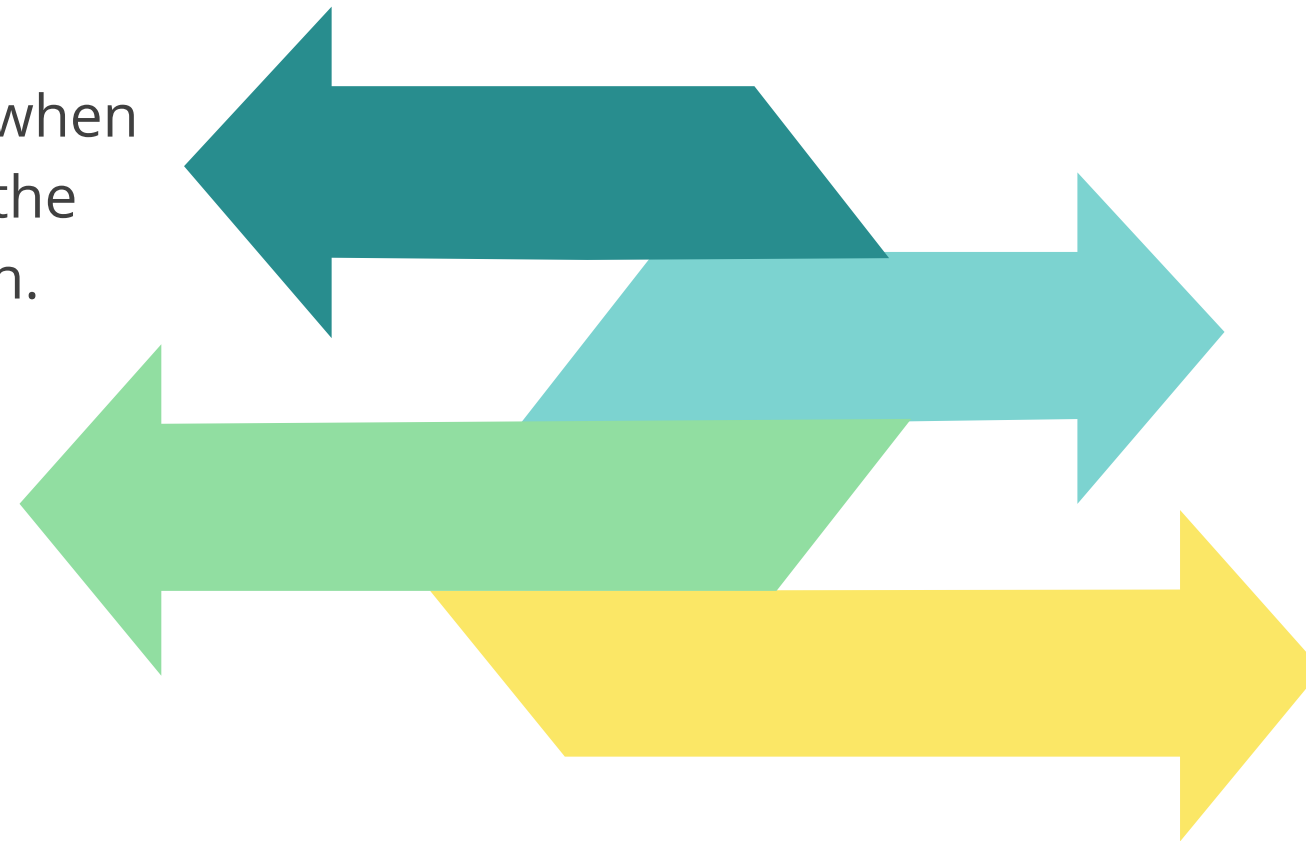
Jenkins automates testing and identifies integration issues.

## **Build steps:**

Jenkins tests code, alerts developers for errors, and moves to integration if successful.

## **User acceptance testing:**

Jenkins automates user testing and deploys code that passes these tests.

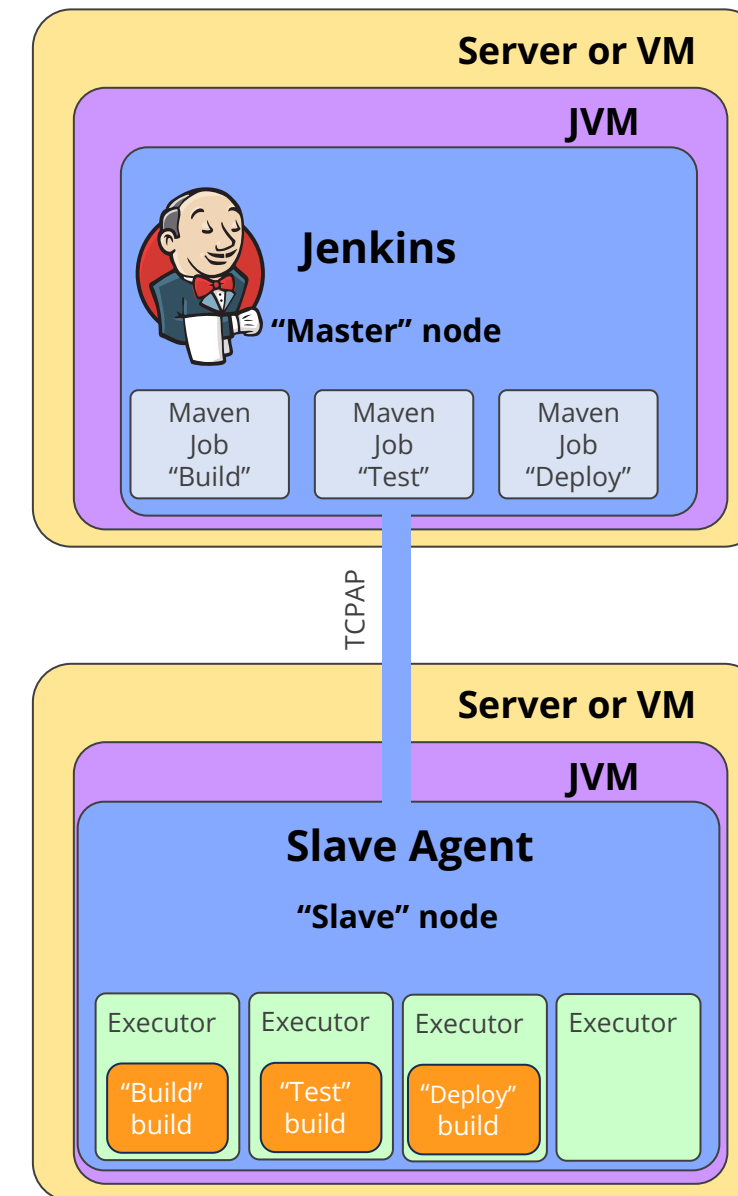


# Jenkins Architecture

Jenkins adheres to the master-slave architecture for managing builds. The slave and master agents communicate via a TCP/IP connection.



# Jenkins



## Quick Check



You are part of a DevOps team responsible for streamlining the software development lifecycle. Which aspect of Jenkins would be most beneficial for automating the build and deployment process of your project?

- A. Using Jenkins' plugins for integrating various tools and technologies
- B. Understanding Jenkins' role in enabling continuous integration and delivery (CI/CD) practices
- C. Exploring Jenkins' distributed architecture for efficient resource utilization
- D. Using Jenkins across different platforms for seamless integration with cloud environments



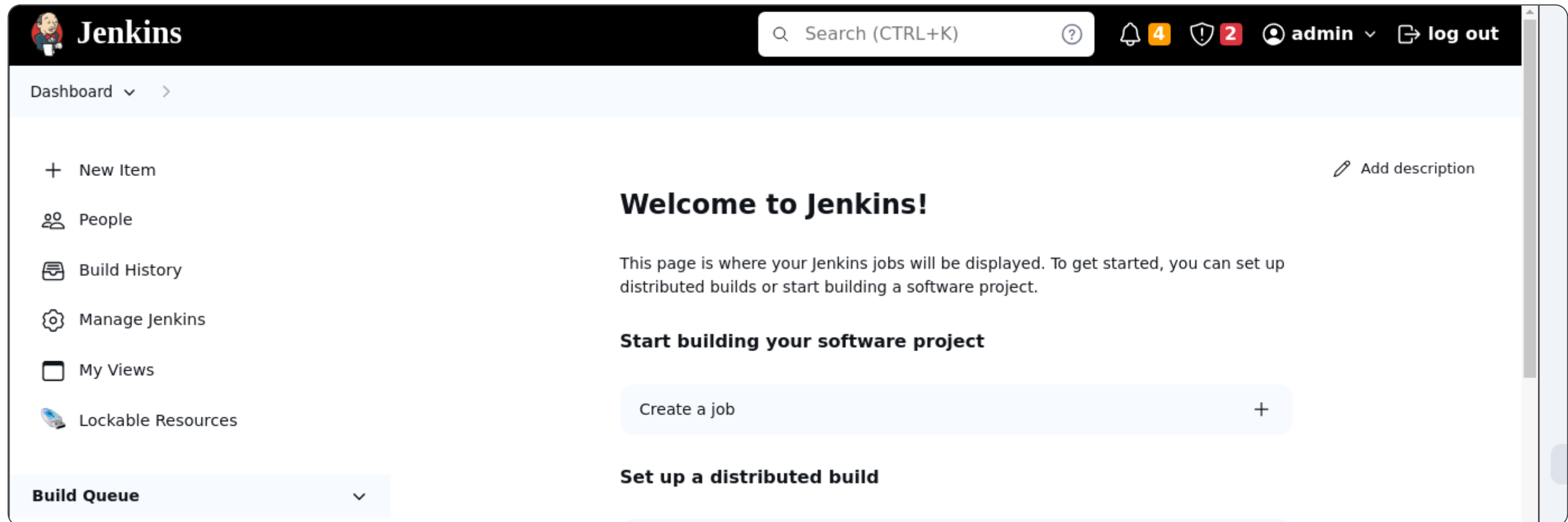
# Exploring Jenkins UI



# Introduction to Jenkins Interface

## Dashboard

It is the main platform for summarizing project statuses.



The screenshot displays the Jenkins Dashboard interface. At the top, a black header bar contains the Jenkins logo on the left, a search bar with the text "Search (CTRL+K)" in the center, and notification icons (a bell with "4", a shield with "2"), the user name "admin" with a dropdown arrow, and a "log out" button on the right. Below the header, a light blue bar shows the breadcrumb "Dashboard" with a dropdown arrow and a right-pointing chevron. The main content area is divided into a left sidebar and a central panel. The sidebar lists several menu items: "New Item" with a plus icon, "People" with a group of people icon, "Build History" with a document icon, "Manage Jenkins" with a gear icon, "My Views" with a list icon, and "Lockable Resources" with a disk icon. At the bottom of the sidebar is a "Build Queue" section with a dropdown arrow. The central panel features a "Welcome to Jenkins!" heading, followed by a paragraph explaining that this is where Jenkins jobs are displayed and providing instructions on how to get started. Below this, there is a section titled "Start building your software project" which contains a light blue button labeled "Create a job" with a plus icon. At the bottom of the central panel, there is a section titled "Set up a distributed build".

Jenkins

Search (CTRL+K)

4 2 admin log out

Dashboard

+ New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

Build Queue

### 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**

Add description

# Introduction to Jenkins Interface

## New item

It refers to the process of creating a new element that aids in the software development lifecycle.

The screenshot displays the Jenkins web interface. At the top, there's a black header with the Jenkins logo, a search bar labeled 'Search (CTRL+K)', and user information 'admin' with a 'log out' link. Below the header, a light blue sidebar on the left contains navigation links: 'Dashboard', '+ New Item' (highlighted with a red box), 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Lockable Resources'. The main content area shows a table of jobs. The first job is 'Demo', which is in a 'Building' state (indicated by a sun icon). The table has columns for 'S' (Status), 'W' (Webhook), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. At the bottom, there's a 'Build Queue' section and a footer with 'Icon: S M L', 'Icon legend', and three Atom feed links.

S	W	Name ↓	Last Success	Last Failure	Last Duration
...	☀	Demo ↓	N/A	N/A	N/A

# Introduction to Jenkins Interface

## People

It provides a list of users who have committed to any Jenkins-configured jobs.

The screenshot displays the Jenkins web interface. At the top, the Jenkins logo and name are on the left, and a search bar with the text 'Search (CTRL+K)' is on the right. To the right of the search bar are notification icons: a bell with '4', a shield with '2', and a user profile labeled 'admin' with a dropdown arrow, followed by a 'log out' button. Below the top bar, a breadcrumb trail shows 'Dashboard' with a dropdown arrow and a right-pointing chevron. On the left sidebar, there are links: '+ New Item', 'People' (highlighted with a red box), 'Build History', 'Manage Jenkins', 'My Views', and 'Lockable Resources'. The main content area shows a table of users. Above the table is a filter button labeled 'All' with a plus sign. The table has columns: 'S' (Status), 'W' (Webhook), 'Name' (with a downward arrow), 'Last Success', 'Last Failure', and 'Last Duration'. There is one row with a blue status icon, a yellow sun icon, the name 'Demo' with a dropdown arrow, and 'N/A' for the last success, failure, and duration. To the right of the table is a green play button icon and a link 'Add description'. At the bottom, there is an 'Icon legend' section with 'S', 'M', and 'L' icons, and three Atom feed links: 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. A 'Build Queue' section is visible at the bottom left.

S	W	Name ↓	Last Success	Last Failure	Last Duration
...	☀	Demo ↓	N/A	N/A	N/A

# Introduction to Jenkins Interface

## Build History

It is a feature that records all executed builds for a specific job.

The screenshot displays the Jenkins web interface. At the top, there's a black header with the Jenkins logo, a search bar labeled 'Search (CTRL+K)', and user information 'admin' with a 'log out' link. Below the header, a light blue sidebar on the left contains navigation links: 'New Item', 'People', 'Build History' (highlighted with a red box), 'Manage Jenkins', 'My Views', and 'Lockable Resources'. The main content area shows the 'Build History' for a job named 'Demo'. It includes a table with columns for status (S, W), name, last success, last failure, and last duration. The 'Demo' job is shown with a status of 'S' (Success) and a last success of 'N/A'. Below the table, there's an 'Icon legend' section with radio buttons for 'S', 'M', and 'L' (selected), and links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. A 'Build Queue' section is visible at the bottom left.

Jenkins

Search (CTRL+K)

admin log out

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

Add description

S	W	Name ↓	Last Success	Last Failure	Last Duration
...	☀	Demo ↓	N/A	N/A	N/A

Icon: S M L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

Build Queue

# Introduction to Jenkins Interface

## Manage Jenkins

It centralizes Jenkins environment management for administrators.

The screenshot displays the Jenkins web interface. At the top, there's a black header with the Jenkins logo, a search bar, and user information. Below this is a light blue sidebar with navigation links. The 'Manage Jenkins' link, represented by a gear icon, is highlighted with a red rectangle. The main content area shows a table of Jenkins items, with the 'Demo' item visible. At the bottom, there's a 'Build Queue' section.

**Jenkins** Search (CTRL+K) 4 2 admin log out

Dashboard >

+ New Item Add description

People

Build History

**Manage Jenkins**

My Views

Lockable Resources

S	W	Name ↓	Last Success	Last Failure	Last Duration
...	☀	Demo ↓	N/A	N/A	N/A

Icon: S M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

Build Queue

# Introduction to Jenkins Interface

## My Views

It is a feature that enables users to craft and oversee their personalized views.

The screenshot displays the Jenkins web interface. At the top, the Jenkins logo is on the left, and a search bar with the text 'Search (CTRL+K)' is on the right. To the right of the search bar are notification icons (a bell with '4', a shield with '2'), the user name 'admin', and a 'log out' button. Below the top bar, a 'Dashboard' dropdown menu is visible. On the left sidebar, the 'My Views' option is highlighted with a red rectangle. The main content area shows a table of views. The table has columns for 'S' (Success), 'W' (Warning), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. A single view named 'Demo' is listed, with 'N/A' for the last success and failure times, and 'N/A' for the last duration. To the right of the table is a green play button icon. Below the table, there is an 'Icon legend' section with three radio buttons labeled 'S', 'M', and 'L'. To the right of the legend are three Atom feed links: 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. At the bottom left, there is a 'Build Queue' dropdown menu.

S	W	Name ↓	Last Success	Last Failure	Last Duration
...	☀	Demo ↓	N/A	N/A	N/A

# Introduction to Jenkins Interface

## Lockable Resources

It is a plugin that defines lockable resources for builds.

The screenshot displays the Jenkins web interface. At the top, the Jenkins logo and name are on the left, and a search bar, notification bell (4), warning shield (2), user profile (admin), and log out button are on the right. Below the header, a sidebar on the left contains navigation links: New Item, People, Build History, Manage Jenkins, My Views, and Lockable Resources (which is highlighted with a red rectangle). The main content area shows a table of lockable resources. The table has columns for S (Status), W (Warning), Name, Last Success, Last Failure, and Last Duration. A single row is visible with a blue status icon, a yellow warning icon, the name 'Demo', and 'N/A' for the last success, failure, and duration. Below the table, there is an 'Icon legend' section with three radio buttons labeled S, M, and L, where L is selected. To the right of the legend are three Atom feed links: 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. At the bottom left, there is a 'Build Queue' section with a dropdown arrow.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Demo	N/A	N/A	N/A

Icon: S M **L** Icon legend

Atom feed for all Atom feed for failures Atom feed for just latest builds

# Introduction to Jenkins Interface

## Build Queue

It shows the jobs that are queued for execution.

The screenshot displays the Jenkins Dashboard interface. On the left sidebar, the 'Build Queue' link is highlighted with a red rectangular box. The main content area shows the 'Start building your software project' section with a 'Create a job' button. Below this is the 'Set up a distributed build' section with buttons for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. The 'Build Queue' section, which is highlighted, shows 'No builds in the queue.' Below it, the 'Build Executor Status' section lists two idle executors. The bottom right corner of the dashboard shows 'REST API' and 'Jenkins 2.426.3'.


Dashboard ▾ >


- Manage Jenkins
- My Views
- Lockable Resources
- Build Queue** ▾  
No builds in the queue.
- Build Executor Status** ▾  
1 Idle  
2 Idle


**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 

REST API Jenkins 2.426.3



# Introduction to Jenkins Interface

## Build Executor Status

It indicates the count of resources available for concurrent build execution.

The screenshot displays the Jenkins Dashboard. On the left sidebar, there are links for 'Manage Jenkins', 'My Views', and 'Lockable Resources'. Below these, the 'Build Queue' section shows 'No builds in the queue.' The 'Build Executor Status' section, highlighted with a red box, shows two idle executors: '1 Idle' and '2 Idle'. The main content area has a header 'Start building your software project' with a 'Create a job' button. Below this, the 'Set up a distributed build' section includes buttons for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. The footer shows 'REST API' and 'Jenkins 2.426.3'.

Dashboard ▾ >

- ⚙️ Manage Jenkins
- 📄 My Views
- 🔒 Lockable Resources

**Build Queue** ▾

No builds in the queue.

**Build Executor Status** ▾

- 1 Idle
- 2 Idle

**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 ?

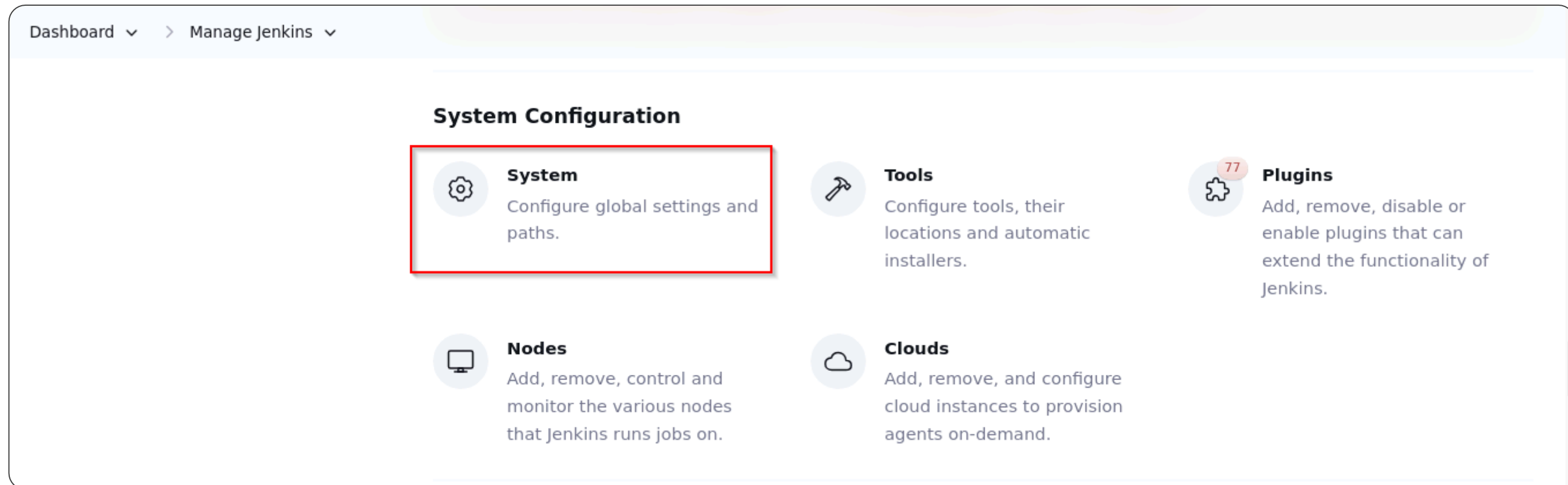
REST API Jenkins 2.426.3

# System Configuration

It is located under the **Manage Jenkins** option in the navigation bar of the Jenkins dashboard, providing functionalities for managing global settings and performing server administration tasks.

## System

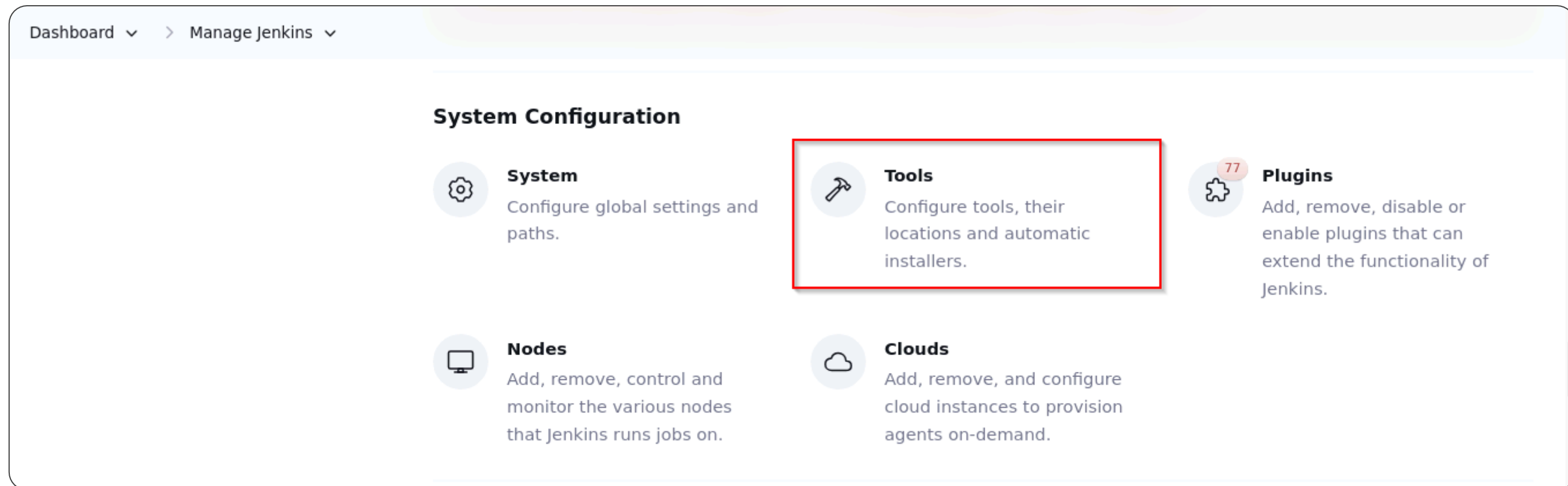
It manages global settings, paths, variables, security, and other configurations.



# System Configuration

## Tools

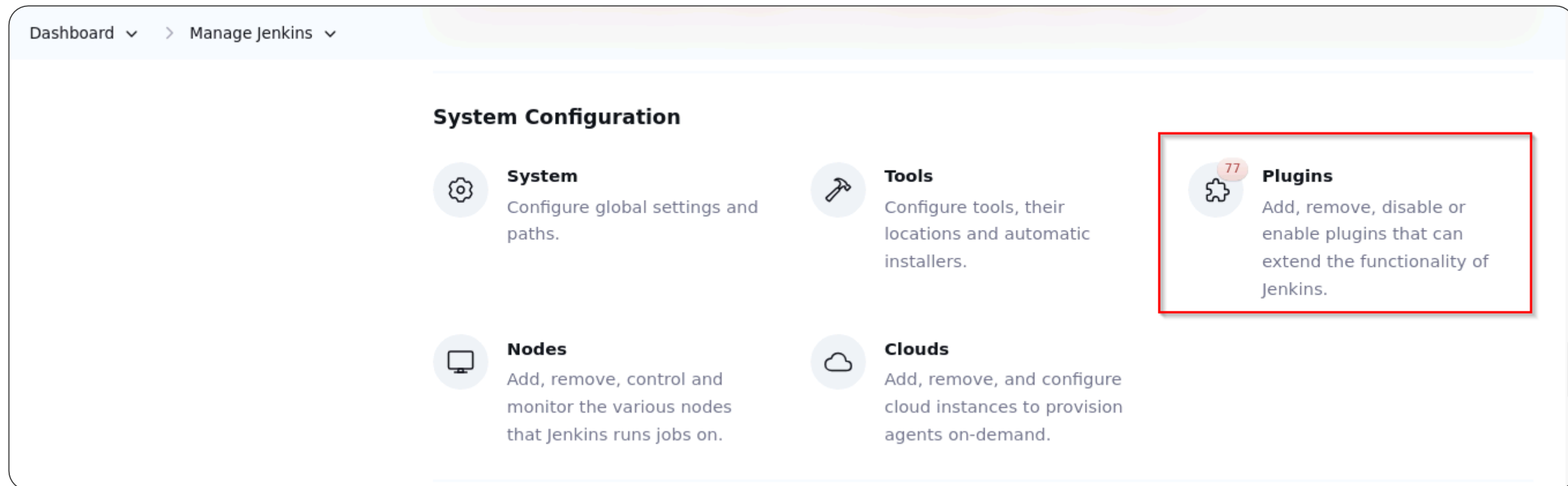
It sets up tools, locations, and installers, like defining JDK or Maven.



# System Configuration

## Plugins

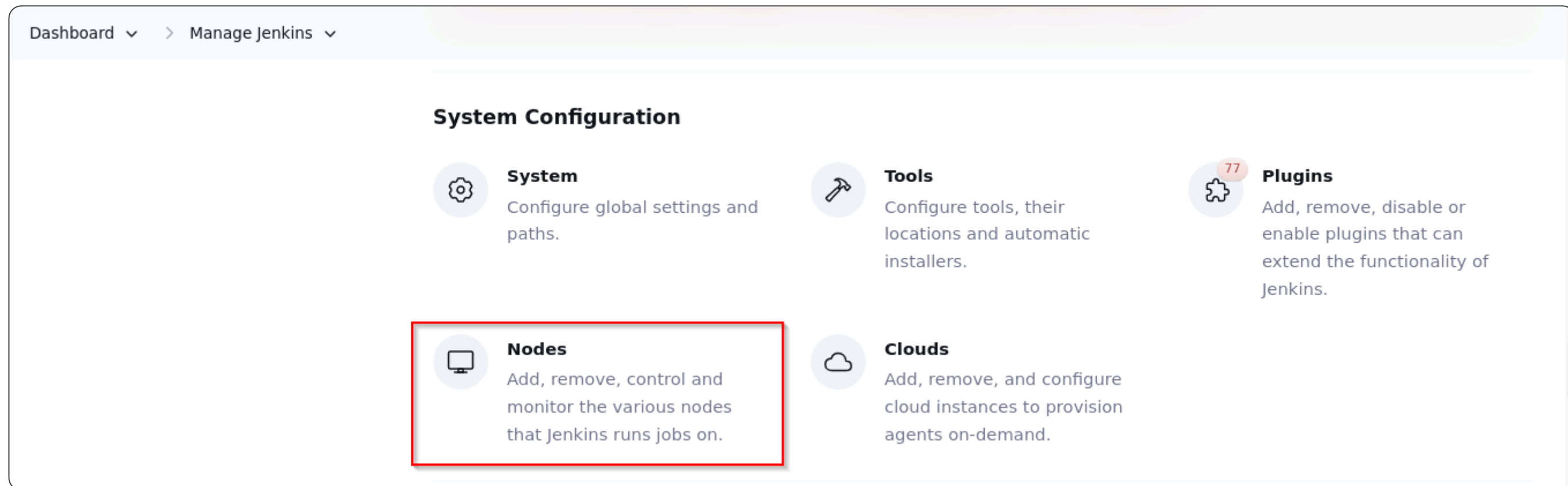
It plays a vital role in extending Jenkins' functionalities and customizing it for your specific needs.



# System Configuration

## Nodes

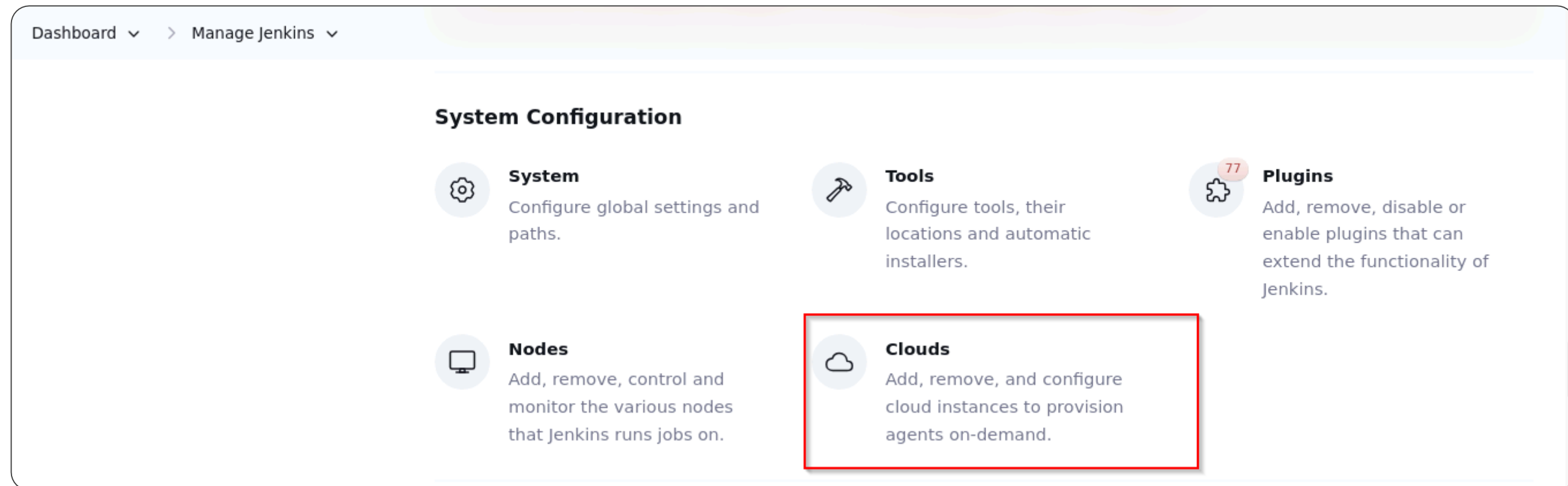
This relates to machines, often called agents, that perform jobs in a Jenkins configuration.



# System Configuration

## Clouds

It refers to a feature that integrates with cloud platforms like AWS, Azure, or GCP.

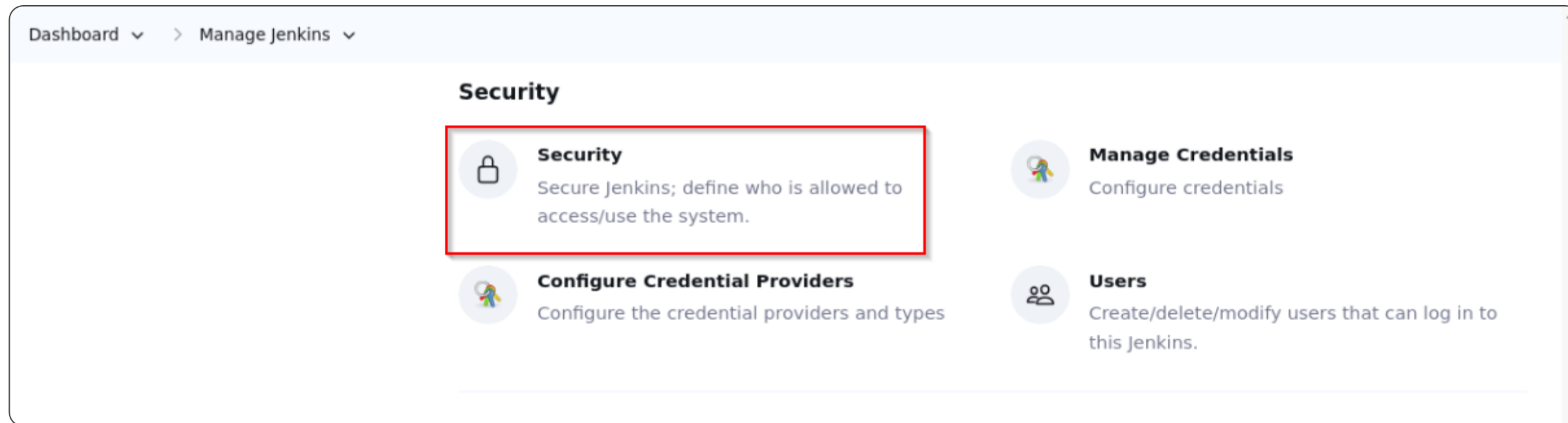


# Security

It enables administrators to set up and control security measures. The following are the main sections:

## Security

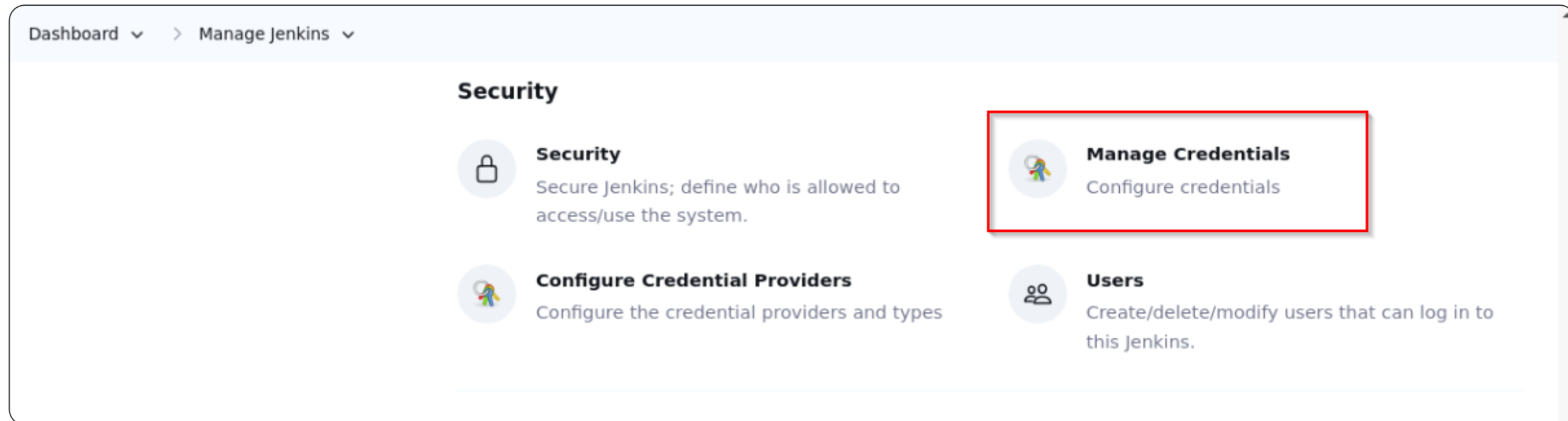
It involves functionalities for user authentication, authorization, and comprehensive server security.



# Security

## Manage Credentials

It is for the secure storage and management of credentials in the build pipelines.

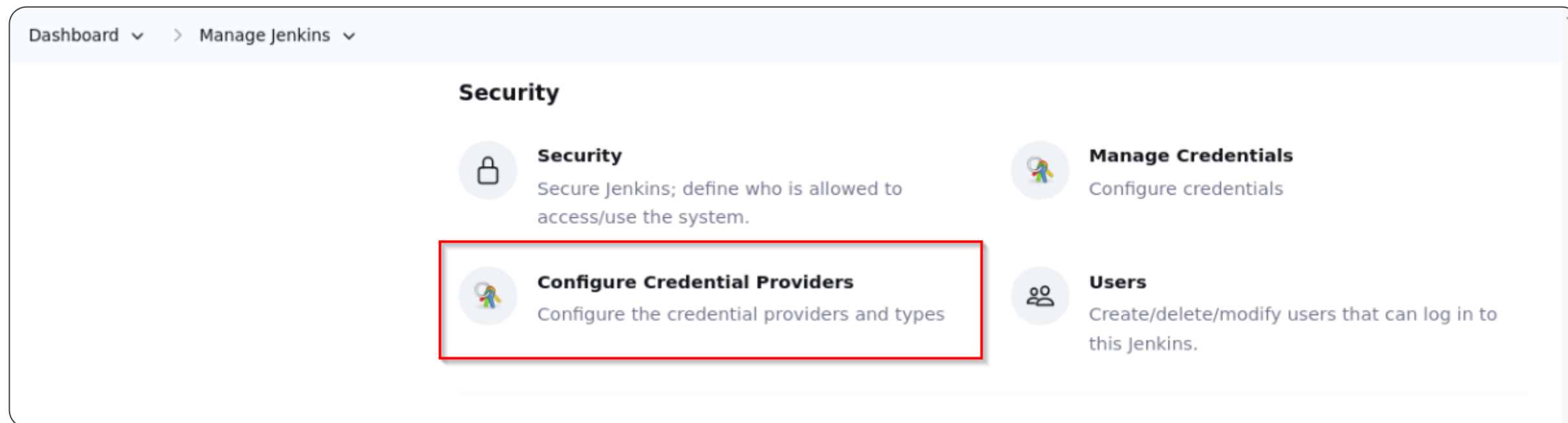




# Security

## Configure Credential Providers

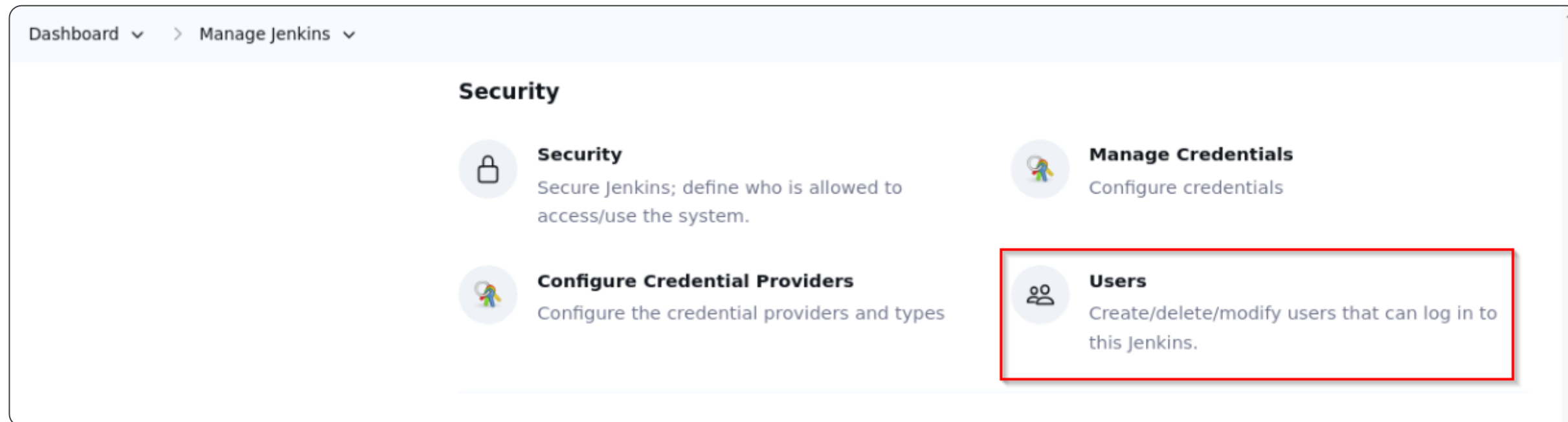
It extends the way Jenkins manages credentials beyond the built-in store.



# Security

## Users

It essentially involves managing user accounts for Jenkins access and interaction.

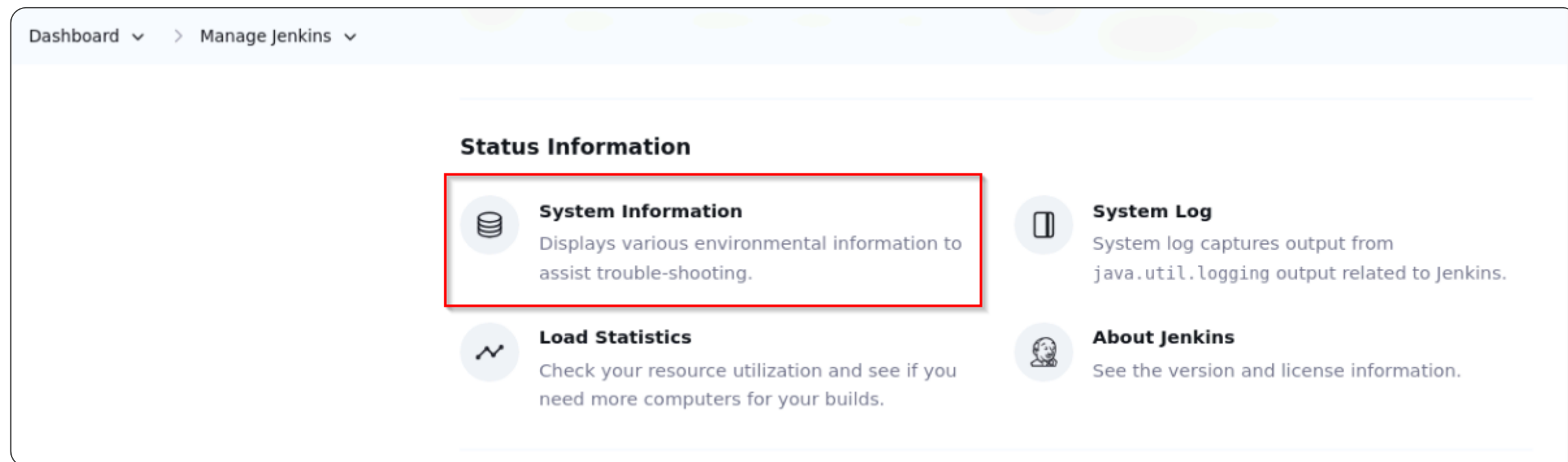


# Status Information

It relates to indicators showing the health and activity of your Jenkins server and its jobs.  
Here is a brief overview:

## System Information

It offers a detailed view of your Jenkins server's setup and resources.





# Status Information


## System Log


It provides a record of all activities happening within the Jenkins server.

Dashboard ▾ > Manage Jenkins ▾

**System Information**  
Displays various environmental information to assist trouble-shooting.

**Load Statistics**  
Check your resource utilization and see if you need more computers for your builds.

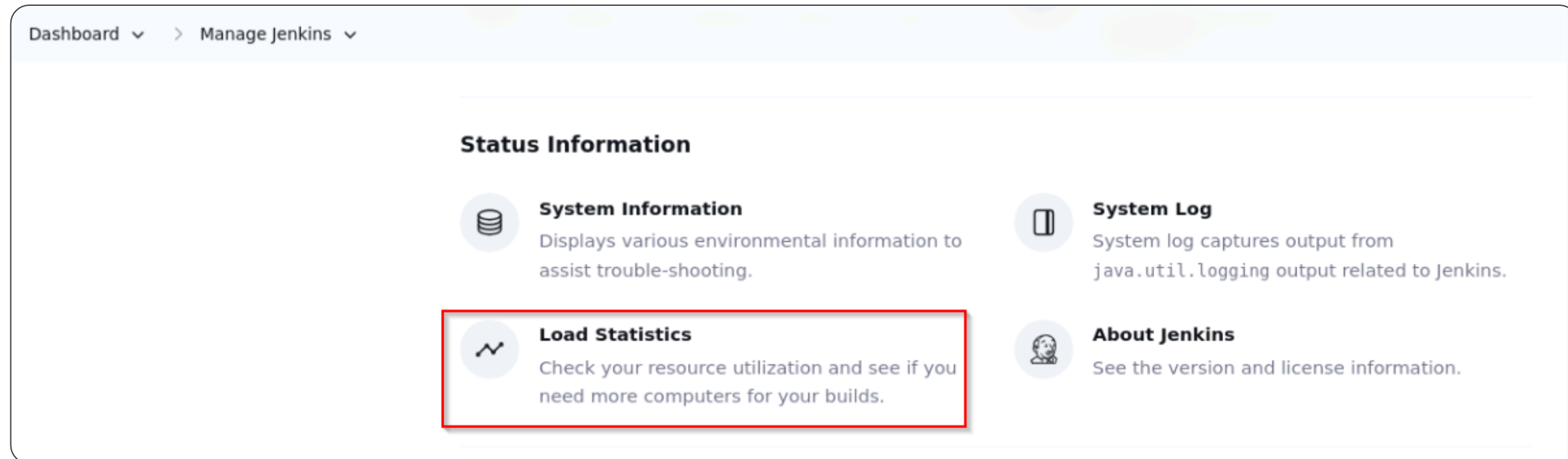
**System Log**  
System log captures output from `java.util.logging` output related to Jenkins.

**About Jenkins**  
See the version and license information.

# Status Information

## Load Statistics

It provides insights into the resource utilization of your Jenkins server and its associated build agents.





# Status Information


## About Jenkins


It offers details about the version and license of the installed Jenkins instance.

Dashboard ▾ > Manage Jenkins ▾

**System Information**  
Displays various environmental information to assist trouble-shooting.

**Load Statistics**  
Check your resource utilization and see if you need more computers for your builds.

**System Log**  
System log captures output from java.util.logging output related to Jenkins.

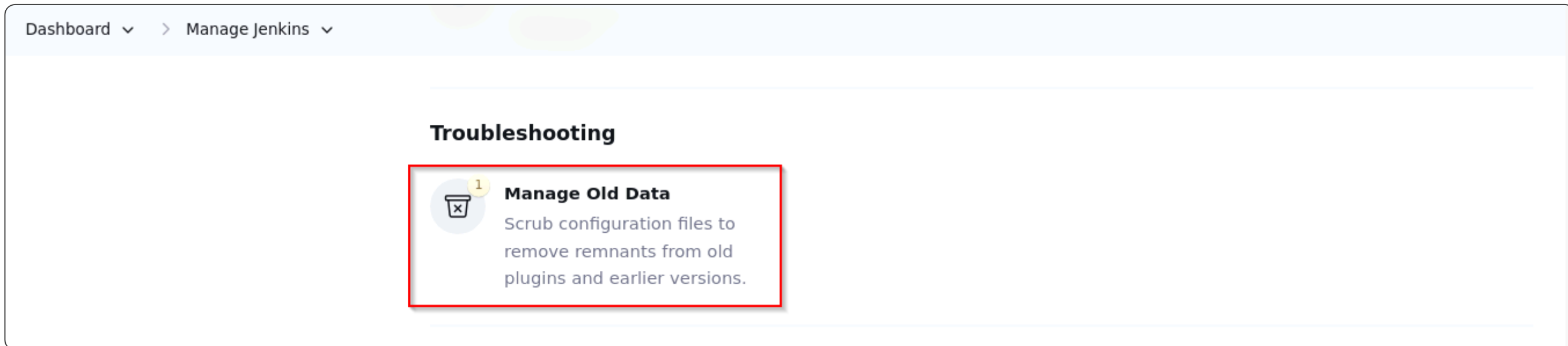
**About Jenkins**  
See the version and license information.

# Troubleshooting

It identifies and resolves issues that prevent effective use of the UI.

## Manage Old Data

It is important to maintain a healthy and efficient Jenkins environment.

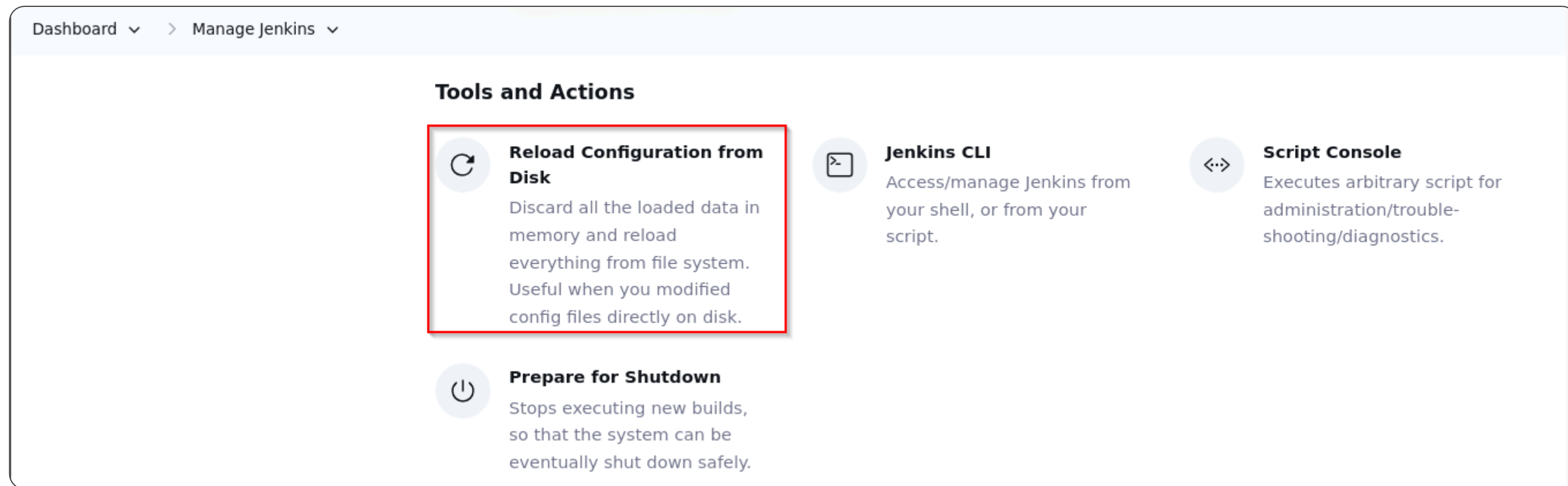


# Tools and Actions

It provides various tools to manage and maintain the Jenkins environment.

## Reload Configuration from Disk

It enables users to discard all data loaded in memory and reload everything from the file system.






# Tools and Actions

## Jenkins CLI


It allows users to access and manage Jenkins from its shell or script.

Dashboard ▾ > Manage Jenkins ▾


**Tools and Actions**

**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/trouble-shooting/diagnostics.

**Prepare for Shutdown**

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


Jenkins 2.426.3

# Tools and Actions


## Script Console

It allows the execution of arbitrary scripts for administration, troubleshooting, and diagnostics.


Dashboard ▾ > Manage Jenkins ▾

**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.

**Prepare for Shutdown**

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

**Jenkins CLI**

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

**Script Console**

Executes arbitrary script for administration/troubleshooting/diagnostics.

Jenkins 2.426.3


# Tools and Actions

## Prepare for Shutdown


It provides an option to stop executing new builds, allowing the system to be shut down safely over time.

Dashboard ▾ > Manage Jenkins ▾


### Tools and Actions

**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/trouble-shooting/diagnostics.

**Prepare for Shutdown**

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

Jenkins 2.426.3

## Assisted Practice



### Creating a new user in Jenkins

Duration: 10 Min.

#### Problem statement:

You have been assigned a task to create a new user in Jenkins for managing access and permissions within the Jenkins environment.

#### Outcome:

By completing this demo, you will be able to create a new user in Jenkins, effectively managing access and permissions within the Jenkins environment.

**Note:** Refer to the demo document for detailed steps

# Assisted Practice: Guidelines



Steps to be followed:

1. Log in to the Jenkins CI tool
2. Access the user management section to manage users

## Quick Check

As a seasoned DevOps engineer managing Jenkins for your organization's CI/CD pipelines, how would you assist a new team member struggling to grasp the Jenkins interface?



- A. Offer a comprehensive walkthrough of the Jenkins interface, emphasizing essential features and functionalities
- B. Encourage exploration of system configuration settings to deepen understanding of Jenkins architecture
- C. Recommend focusing on troubleshooting techniques to address any UI-related issues encountered
- D. Suggest exploring status information and logs to monitor job execution and identify errors or failures

# Organizations Using Jenkins

Notable companies leveraging Jenkins are as follows:

The Netflix logo, consisting of the word "NETFLIX" in a bold, red, sans-serif font.

Jenkins helps Netflix automate the build and test processes, enabling continuous integration and accelerating software development. This improves velocity, user experience, and business outcomes.

The Uber logo, featuring the word "Uber" in a bold, black, sans-serif font.

Uber uses Jenkins to optimize development efficiency, improve reliability through automated deployments, and ensure scalability and security, enabling smooth software updates and global operations.



Salesforce uses Jenkins to automate its diverse and complex deployments, ensuring enhanced efficiency, customization, scalability, and the integration of security and compliance checks into its development pipeline.

# Key Takeaways

- Continuous Integration (CI) is a development practice that emphasizes early integration of all development work, promoting frequent integration of code changes into a central repository.
- Continuous Delivery (CD) extends continuous integration, automating the software release process to ensure code changes are always deployment-ready for production.
- Jenkins is an automated tool offering a versatile platform for developing, testing, analyzing, deploying, and monitoring software changes.
- Jenkins follows a master-slave architecture to manage builds, where communication between the master and slave agents occurs over a TCP/IP connection.
- The Jenkins interface allows users to manage builds, set up pipelines, and administer the environment efficiently.







**Thank You**