CI/CD Core

- 1. CI/CD Core Overview
- 2. CMDB (Configuration Management Database)
- 3. Pipelines
- 4. Integrations

1. CI/CD Core Overview

Continuous Integration (CI) and Continuous Deployment (CD) are practices that help you quickly and reliably deliver software updates to customers. CI/CD Core module provides an interface to connect with external tools (e.g., Jenkins) for automating builds, tests, and deployments.

Use Cases

1. **Set Up a Jenkins Integration:** Navigate to CI/CD → Integrations → + Add, enter "Jenkins" as the name, add the Jenkins server URL, and provide credentials (API token, username, secret key).
2. **Check Pipelines:** Click Pipelines, select Jenkins, and view existing Jenkins jobs or runs for better tracking.
3. **Monitor Builds:** As Jenkins triggers automated builds/tests, the system continuously updates and displays the run statuses for real-time monitoring.

The following actions are available in the CI/CD Core module:

1. CMDB
2. Integrations
3. Pipelines

The CI/CD Core module links your system with external DevOps pipelines. By setting up an integration (e.g., Jenkins), you can authenticate and manage your continuous integration/deployment processes directly from within the system. If your organization uses other CI/CD platforms, our system can integrate with those—just provide the required credentials and endpoints.

2. CMDB (Configuration Management Database)

The CMDB module provides a centralized view of configuration items (instances) grouped by project. It allows teams to manage, search, and organize technical targets and environments that are used across security, CI/CD, and operational workflows.

CMDB is project-aware: all instances are created and viewed within the context of a selected project.

Layout Overview

The CMDB screen is divided into two main areas:

Settings by project
Displays a horizontal list of available projects. Selecting a project updates the visible configuration items (instances) below.
A search field allows quick filtering of projects by name.

Instances
Displays configuration items (targets) associated with the selected project.
Includes:

1. Search field to find instances by name
2. **+ Add** button to create a new instance
3. Table view for managing existing instances

If no instances exist for the selected project, the system shows **“No data to display.”**

Instances Table

Each row in the Instances table represents a configuration item (CI) or target.

Columns:
Target – Name or identifier of the configuration item
Actions – Edit or manage the instance

The table updates dynamically based on the selected project.

Adding an Instance

To create a new configuration item:

1. Select a project from **Settings by project**.
2. In the **Instances** section, click **+ Add**.
3. Enter the required instance details (target name and related configuration).
4. Click **Save**.

The new instance will appear in the table under the selected project.

Searching Instances

Use the **Search** field in the Instances section to quickly locate configuration items by name. Search is applied only within the currently selected project.

Project-Based Scope

1. Instances belong to a single project.
2. Switching projects immediately updates the Instances list.
3. Each project maintains its own isolated CMDB scope.

This ensures clear separation between environments, applications, or customers.

3. Pipelines

Use the **Pipelines** page to connect your external CI/CD systems (like Jenkins) and view build runs alongside your security reports.

Use Cases

1. **Accessing Pipelines**
   Navigate to **CI/CD → Pipelines** from the left menu. Select an existing integration (e.g., Jenkins) from the **Automation** dropdown to manage builds and workflows.
2. **Tracking Runs**
   Once your integration is set up, build runs or jobs from your external CI system (e.g., Jenkins) will appear in **Runs**, displaying execution history, statuses, and timestamps.
3. **Future Expansions**
   If integrating another CI/CD platform (e.g., GitLab CI, Travis CI), simply provide the **API keys and URLs** to extend platform compatibility and manage builds seamlessly.

Automation

Through this section, you can manage and select integrations that you've configured for your product—enabling streamlined automation and efficient control over connected systems.

1. **Automation dropdown**: lists all CI/CD integrations you've set up.
2. Select **Jenkins** (or another tool) to start pulling in build data.

Runs

Once an integration is selected, this table populates with your pipeline executions:
Name ⇅: the identifier of each pipeline or job.
Status: success, failure, in-progress, etc.
Start / Finish Timestamps: when each run began and ended.
Duration: total build time.

If you haven't run any builds yet, you'll see:

> *No data to display*

In the provided screenshots, no runs are currently displayed ("No data to display"), but in a real scenario, you'd see a list of pipeline executions, their status, timestamps, and other

> relevant data.

Getting Started

1. **Configure** your CI/CD server (e.g. Jenkins) with its API endpoint and access token in **Integrations**.
2. **Select** that integration from the **Automation** dropdown on this page.
3. **Trigger** a build in Jenkins (or your chosen system).
4. **Refresh** the **Pipelines → Runs** table here—your new build appears with full details.

Future Expansions

To integrate additional platforms (GitLab CI, Travis CI, etc.), simply add their API credentials under **Integrations**. They’ll then appear in the same **Automation** dropdown for seamless management.

4. Integrations

The **CI/CD (Continuous Integration / Continuous Deployment Management System)** is designed to streamline the process of integrating, deploying, and managing automated pipelines.

Use Cases

1. **Accessing Integrations**
   Navigate to **CI/CD → Integrations** from the left-hand menu to view a list of configured integrations (e.g., Jenkins).
2. **Adding or Editing an Integration**
   Click **+ Add** to create a new integration or select the **Edit (pencil)** icon to modify an existing one. Enter the **name/label**, **URLs**, and **credentials** (API tokens, usernames, secret keys), then click **Save** to apply changes.
3. **Why Integrations Matter**
   The system uses valid credentials to authenticate with your CI/CD tool (e.g., Jenkins), enabling automated pipeline triggers and monitoring. Incorrect credentials or URLs will prevent successful integration.

> **Important**: Integrations with unsupported CI/CD tools will not work. Ensure that only pre-approved and compatible systems are configured.

1. Accessing Integrations
In the left-hand menu, expand CI/CD and click Integrations.

> A list of configured integrations appears (e.g., “Jenkins”).

2. Adding or Editing an Integration
Click + Add (if visible) to create a new integration, or click the Edit (pencil) icon on an existing one.

1. **Name / Label**: Provide a friendly name for the integration (e.g., “Jenkins”).
2. **URLs**: Add one or more URLs and corresponding keys (e.g., apiUrl: https://...).
3. **Credentials**: Input tokens, usernames, secret keys, or other authentication data required by your CI/CD tool.

Click **Save** to store your changes.

3. Why Integrations Matter
The system can authenticate with your CI/CD tool (e.g., Jenkins) and trigger or monitor pipelines/jobs by entering valid credentials.

> Invalid credentials or URLs will prevent successful integration.

Key Points to Remember

1. **Only pre-approved integrations work** – attempting to connect unsupported systems will result in errors.
2. **Correct credentials are required** for successful integration.
3. **Deployment pipelines appear automatically** after integration with a CI/CD tool.
4. **New integrations can be added** based on business needs, but require prior discussion and approval.