Red Hat Developer Hub 1.3

Introduction to plugins

Last Updated: 2025-04-17

Legal Notice

Abstract

The Red Hat Developer Hub (RHDH) application offers a unified platform with various plugins. Using the plugin ecosystem within the Developer Hub application, you can access your development infrastructure and software development tools.

Table of Contents

CHAPTER 1. PLUGINS IN RED HAT DEVELOPER HUB

The Red Hat Developer Hub (RHDH) application offers a unified platform with various plugins. Using the plugin ecosystem within the RHDH application, you can access any kind of development infrastructure or software development tool.

Plugins are modular extensions for RHDH that extend functionality, streamline development workflows, and improve the developer experience. You can add and configure plugins in RHDH to access various software development tools.

Each plugin is designed as a self-contained application and can incorporate any type of content. The plugins utilize a shared set of platform APIs and reusable UI components. Plugins can also retrieve data from external sources through APIs or by relying on external modules to perform the tasks.

RHDH provides both static and dynamic plugins that enhance its functionality. Static plugins are integrated into the core of the RHDH application, while dynamic plugins can be sideloaded into your Developer Hub instance without the need to recompile your code or rebuild the container.

To install or update a static plugin you must update your RHDH application source code and rebuild the application and container image.

To install or update a dynamic plugin, you must restart your RHDH application source code after installing the plugin.

You can also import your own custom-built or third-party plugins or create new features using dynamic plugins.

Dynamic plugins boost modularity and scalability by enabling more flexible and efficient functionality loading, significantly enhancing the developer experience and customization of your RHDH instance.

1.1. DYNAMIC PLUGINS IN RED HAT DEVELOPER HUB

You can use RHDH dynamic plugins in environments where flexibility, scalability, and customization are key. Using dynamic plugins in RHDH provides:

Modularity and extensibility

You can add or modify features without altering the core RHDH application. This modular approach makes it easier to extend functionality as needs evolve.

Customization

You can tailor RHDH to fit specific workflows and use cases, enhancing the overall user experience.

Reduced maintenance and update overhead

You can deploy the updates or new features independently of the main RHDH codebase, reducing the risks and efforts associated with maintaining and updating the platform.

Faster iteration

You can create and test new features more rapidly as plugins, encouraging experimentation and enabling you to quickly iterate based on feedback.

Improved collaboration

You can share plugins across teams or even externally. This sharing can foster collaboration and reduce duplication of effort, as well as help establish best practices across an organization.

Scalability

As organizations grow, their needs become complex. Dynamic plugins enable RHDH to scale alongside such complex needs, accommodating an increasing number of users and services.

Ecosystem growth

Fostering the development of plugins can create a dynamic ecosystem around RHDH. This community can contribute to plugins that cater to different needs, thereby enhancing the platform.

Security and compliance

You can develop plugins with specific security and compliance requirements in mind, ensuring that RHDH installations meet the necessary standards without compromising the core application.

Overall, the use of dynamic plugins in RHDH promotes a flexible, adaptable, and sustainable approach to managing and scaling development infrastructure.

1.2. COMPARING DYNAMIC PLUGINS TO STATIC PLUGINS

Static plugins are built into the core of the RHDH application. Installing or updating a static plugin requires a restart of the application after installing the plugin.

The following table provides a comparison between static and dynamic plugins in RHDH.

Feature	Static plugins	Dynamic plugins
Integration	Built into the core application.	Loaded at runtime, separate from the core.
Flexibility	Requires core changes to add or update features.	Add or update features without core changes.
Development speed	Slower, requires a complete rebuild for new features.	Faster, deploy new functionalities quickly.
Customization	Limited to predefined options.	Easy to tailor platform by loading specific plugins.
Maintenance	More complex due to tightly coupled features.	Enhanced by modular architecture.
Resource use	All features loaded at startup.	Only necessary plugins loaded dynamically.
Innovation	Slower experimentation due to rebuild cycles.	Quick experimentation with new plugins.