Red Hat Developer Hub 1.8

Red Hat Developer Hub release notes

Release notes for Red Hat Developer Hub 1.8

Last Updated: 2026-01-22

Red Hat Developer Hub 1.8 Red Hat Developer Hub release notes

Release notes for Red Hat Developer Hub 1.8

Legal Notice

Abstract

Red Hat Developer Hub (Developer Hub) 1.8 is now generally available. Developer Hub is a fully supported, enterprise-grade productized version of upstream Backstage 1.42.5. This document contains release notes for the Red Hat Developer Hub 1.8.

Table of Contents

PREFACE

Red Hat Developer Hub (Developer Hub) 1.8 is now generally available. Developer Hub is a fully supported, enterprise-grade productized version of upstream Backstage, compatible with version 1.42.5. Plugins might be compatible with a newer Backstage version. You can access and download the Red Hat Developer Hub application from the Red Hat Customer Portal or from the Ecosystem Catalog.

CHAPTER 1. NEW FEATURES

This section highlights new features in Red Hat Developer Hub 1.8.

1.1. BUILT-IN MONITORING FOR RED HAT DEVELOPER HUB OPERATOR

This update introduces built-in monitoring for the Red Hat Developer Hub Operator. By enabling **spec.monitoring.enabled: true** in your Backstage custom resource, the Red Hat Developer Hub Operator will automatically manage service monitor resources for Prometheus metrics collection on the **/metrics** endpoint. This simplifies the monitoring process, eliminating the need for manual service monitor setup, and enhances the user experience on OpenShift and Kubernetes clusters with Prometheus Operator installed.

Additional resources

- Enabling metrics monitoring in a Red Hat Developer Hub Operator installation on an OpenShift Container Platform cluster

1.2. ENHANCED PACKAGE MANAGEMENT IN THE EXTENSIONS PAGE

You can manage installed packages more easily under the reorganized **Extensions** navigation. The former **Installed Plugins** tab is renamed to **Installed Packages** and includes new row actions that allow you to edit, enable, and disable packages directly from the list.

1.3. INTRODUCING LOCALIZATION SUPPORT IN CORE PLUGINS, AND FRENCH LOCALIZATION

This update introduces localization support and French localization to these core plugins:

- Adoption insights
- AI Integrations
- Bulk Import
- Extensions (Marketplace)
- Lightspeed
- Orchestrator
- QuickStart
- RBAC
- ScoreCard
- Topology
- Global header
- Homepage

- Tekton
- ArgoCD

This enhancement allows Red Hat Developer Hub to display content in French, improving accessibility for users who speak French. AI/Cursor automation ensures a seamless translation process, enhancing the user experience by providing content in their preferred language, preparing Red Hat Developer Hub for use in multilingual environments and fostering a more inclusive developer community.

Additional resources

- Selecting the language for your Developer Hub instance

1.4. LOCALIZATION SUPPORT FOR STRINGS DEFINED IN RED HAT DEVELOPER HUB CONFIGURATION FILES

With this update, localization support is introduced for strings defined in Red Hat Developer Hub configuration files such as **app-config.yaml** and **dynamic-plugins.default.yaml**. This enables users to customize the interface in their preferred language, providing a consistent multilingual interface across these components:

- Entity Tabs Configuration
- Global Header
- QuickStart
- Sidebar Menu Items
- Floating Action Button (FAB) labels and tooltips

This localization support ensures a more inclusive and user-friendly experience for a diverse user base, improving user experience and supporting global users.

Additional resources

- Enabling Quickstart localization in RHDH

1.5. PLUGINS LOCALIZATION SUPPORT

With this update, Red Hat Developer Hub integrates the Backstage localization framework, enabling users to load translations provided by their plugins. The selected language will persist according to the user settings persistence configuration. Additionally, users can load translations from an external JSON file, allowing them to override existing translations or add translations for existing translation keys.

Additional resources

- Localization support for plugins

1.6. ENHANCED BULK IMPORT WITH SCAFFOLDER TEMPLATES

With this update, users can enhance the Bulk Import plugin by importing repositories using scaffolder templates. This automates and optimizes the process by integrating with existing Backstage templates and Orchestrator workflows. Users can select their preferred pre-ingestion workflow and incorporate various scaffolder actions into their bulk import process, resulting in a more efficient and flexible Bulk Import experience.

Additional resources

- Input parameters for Bulk Import Scaffolder template

1.7. ENABLING SOFTWARE TEMPLATE VERSION UPDATE NOTIFICATIONS

With this update, you can enable notification alerts whenever a Software Template is updated with a new version.

Additional resources

- Enabling Software Template version update notifications in Red Hat Developer Hub

1.8. SOFTWARE TEMPLATE PROVENANCE AND DEPENDENCY TRACKING

With this update, Red Hat Developer Hub supports Software Template provenance and a dedicated dependency view to improve component traceability and lifecycle management across your organization.

1.9. ADDITIONAL RESOURCES

- Tracking component origin and Software Template version

1.10. USERS CAN CUSTOMIZE THEIR HOMEPAGE

With this update, Red Hat Developer Hub users can customize their homepage, empowering personalization and productivity. Users can now move, resize, remove, and add existing cards, fostering a more flexible and adaptable user experience. The customization options are based on the existing settings, and users can reset their configuration to the default. The feature aims to improve the resize and reorder mechanism, and update existing cards to work better on different card sizes.

Additional resources

- Customizing the Home page

1.11. DEVELOPER HUB COMMUNITY PLUGINS UPDATED TO BACKSTAGE 1.42

The Developer Hub community plugins have been updated to Backstage version 1.42.

1.12. QUICK START EXPERIENCE FOR DEVELOPERS LOGGING INTO RED HAT DEVELOPER HUB FOR THE FIRST TIME

With this update, Red Hat Developer Hub includes a guided Quick Start experience tailored for the developer persona. This new feature appears automatically upon a developer's first login to help them get started quickly and accelerate adoption of the platform.

The developer Quick Start provides guided next steps for key features, including:

- Bulk import
- Software Catalog
- Self-service templates
- Learning paths

This feature is integrated with RBAC, allowing platform engineers to configure the Quick Start content and conditionally display it to specific developer groups for personalized onboarding.

Additional resources

- {setting-up-and-configuring-your-first-rhdh-instance-link}[{setting-up-and-configuring-your-first-rhdh-instance-title}]

1.13. TRANSPARENT PLUGIN SUPPORT INDICATORS

With this update, the plugin support model is now transparently aligned with Red Hat's standard release classifications. The **Verified** badge is removed, a clearer tiered support system is implemented, and plugin metadata are enhanced, providing a more informative and user-friendly experience in the **Extensions Catalog**. This change improves visibility into plugin maturity, enhances customer trust, and ensures internal consistency across the catalog.

Release Status

Indicates the support status:

- Generally Available (GA)
- Technology Preview (TP)
- Developer Preview (DP).

Supported By

Indicates the support provider:

- Red Hat
- Partner Name
- Customer Name
- Backstage Community

Backstage Version Supported

Replaces the previous **Version** field.

Author

Indicates the plugin author.

Tags

Indicates the plugin tags.

Category

Indicates the plugin category.

Badges

GA (Green)

For plugins that have the **Generally Available Release Status**, and a non empty **Supported By** value.

Certified (Purple)

For Red Hat partner plugins (unchanged).

Custom (Yellow)

For customer-developed plugins (unchanged).

> NOTE
>
> The **Verified** badge is removed.

Additional resources

- Utilizing plugin indicators and support types in Red Hat Developer Hub

1.14. ENHANCED PACKAGE MANAGEMENT IN THE EXTENSIONS PAGE

With this update, Red Hat Developer Hub includes a reorganized Extensions navigation, enabling the user to manage installed packages more easily. The former **Installed Plugins** tab is renamed to **Installed Packages** and includes new row actions that allow you to edit, enable, and disable packages directly from the list.

1.15. SUPPORT FOR HIGH AVAILABILITY IN GOOGLE KUBERNETES ENGINE

Red Hat Developer Hub now supports high availability setups in Google Kubernetes Engine (GKE). This enhancement allows the deployment to scale beyond a single replica, ensuring the application remains operational and accessible even in the event of failures or disruptions.

Additional resources

- Configuring high availability in Red Hat Developer Hub

1.16. CUSTOMIZABLE CONTAINER DEPLOYMENT IN RED HAT DEVELOPER HUB PODS

Previously, injecting necessary files (**extraFiles**) and environment variables (**extraEnvs**) was restricted to the default **backstage-backend** container.

With this update, you can configure resource injection for any container in the Red Hat Developer Hub pod, including sidecars and system containers. This allows you to complete the job of deploying custom components, such as security agents, log collectors, or configuration managers, that require specific volumes or runtime variables to operate successfully.

Additional resources

- Injecting extra files and environment variables into Backstage containers

CHAPTER 2. BREAKING CHANGES

This section lists breaking changes in Red Hat Developer Hub 1.8.

2.1. ARGO CD, TEKTON, AND TOPOLOGY PLUGINS REQUIRE THE KUBERNETES FRONTEND AND KUBERNETES BACKEND PLUGINS

With this update, the Argo CD, Tekton, and Topology plugins requires the Kubernetes Frontend and Kubernetes Backend plugins. Before this update, these plugins depended only on the Kubernetes Backend plugin, and the feature provided by the Kubernetes Frontend plugin where provided by the removed **@janus-idp/shared-react** package.

Procedure

1. Enable the Kubernetes Frontend and Kubernetes Backend plugins.

```
global:
  dynamic:
    plugins:
      - package: ./dynamic-plugins/dist/backstage-plugin-kubernetes
        disabled: false
      - package: ./dynamic-plugins/dist/backstage-plugin-kubernetes-backend-dynamic
        disabled: false
```

2. The Kubernetes Frontend plugin shows automatically a **Kubernetes** tab for Software Catalog entities with the annotation **backstage.io/kubernetes-id** or **backstage.io/kubernetes-namespace**.
   Optionally, to hide the **Kubernetes** tab in the Software Catalog, disable the feature with this dynamic plugin configuration:

```
global:
  dynamic:
    plugins:
      - package: ./dynamic-plugins/dist/backstage-plugin-kubernetes
        disabled: false
        pluginConfig:
          dynamicPlugins:
            frontend:
              backstage.plugin-kubernetes:
                mountPoints: []
      - package: ./dynamic-plugins/dist/backstage-plugin-kubernetes-backend-dynamic
        disabled: false
```

CHAPTER 3. DEPRECATED FUNCTIONALITIES

This section lists deprecated functionalities in Red Hat Developer Hub 1.8.

3.1. BACKSTAGE CR VERSIONS V1ALPHA1 AND V1ALPHA2

Backstage CR versions **v1alpha1** and **v1alpha2** were deprecated in 1.7 and will be removed in 1.9.

3.2. DEPRECATION OF BUNDLED PLUGIN WRAPPERS

To enhance performance, decrease image size, and reduce maintenance, the method of including wrapped dynamic plugins within the main Red Hat Developer Hub container image was deprecated in RHDH 1.7.0. Red Hat Developer Hub is transitioning to a model where all dynamic plugins will be distributed as independent OCI artifacts.

This is a deprecation notice only; there are no breaking changes in 1.8.0 related to plugins, wrappers, or support for OCI artifacts. All previously bundled plugins will continue to be bundled in this release. However, we encourage customers to prepare for the removal of wrappers in a future release by beginning to use the new OCI artifacts in 1.9.0. Documentation will be updated to guide this migration.

Additional resources

- Loading a plugin packaged as an OCI image

3.3. DEPRECATION OF OCM PLUGINS

The Open Cluster Management (OCM) plugins integrates your Red Hat Developer Hub instance with the MultiClusterHub and MultiCluster engines of OCM. The OCM plugins are deprecated as of RHDH 1.8, and will be removed in a future release.

CHAPTER 4. TECHNOLOGY PREVIEW

This section lists Technology Preview features in Red Hat Developer Hub 1.8.

> **IMPORTANT**
>
> Technology Preview features provide early access to upcoming product innovations, enabling you to test functionality and provide feedback during the development process. However, these features are not fully supported under Red Hat Subscription Level Agreements, may not be functionally complete, and are not intended for production use. As Red Hat considers making future iterations of Technology Preview features generally available, we will attempt to resolve any issues that customers experience when using these features. See: Technology Preview support scope.

4.1. BULK IMPORT GITLAB PROJECTS

With this update, users can bulk import entities from GitLab into Red Hat Developer Hub, enhancing onboarding efficiency.

Additional resources

- Importing multiple GitLab repositories

CHAPTER 5. DEVELOPER PREVIEW

This section lists Developer Preview features in Red Hat Developer Hub 1.8.

IMPORTANT

Developer Preview features are not supported by Red Hat in any way and are not functionally complete or production-ready. Do not use Developer Preview features for production or business-critical workloads. Developer Preview features provide early access to functionality in advance of possible inclusion in a Red Hat product offering. Customers can use these features to test functionality and provide feedback during the development process. Developer Preview features might not have any documentation, are subject to change or removal at any time, and have received limited testing. Red Hat might provide ways to submit feedback on Developer Preview features without an associated SLA.

For more information about the support scope of Red Hat Developer Preview features, see Developer Preview Support Scope.

5.1. DYNAMIC PLUGIN FACTORY TO CONVERT PLUGINS INTO DYNAMIC PLUGINS

You can automate the conversion and packaging of standard Backstage plugins into RHDH dynamic plugins by using the RHDH Dynamic Plugin Factory tool.

The core function of the Dynamic Plugin Factory tool is to streamline the dynamic plugin build process, offering the following capabilities:

5.2. EVENTS MODULE AVAILABLE

With this release, you can use the Events Module together with scheduled updates to make sure your GitHub user or catalog entities are updated whenever changes occur in the external system.

Additional resources

- Configuring Events module

5.3. BUILT-IN TECHDOCS FOR RHDH LOCAL

With this update, RHDH Local includes essential *Getting Started* and *How-To* documentation about RHDH Local itself, embedded as TechDocs. Once running, access this built-in documentation directly in the application.

To learn how to configure RHDH for its supported platforms, rather see Red Hat Developer Hub documentation.

5.4. TECH RADAR AND QUAY PLUGINS IN RHDH LOCAL

With this update, RHDH Local integrates the Tech Radar and Quay plugins, for a better out of the box experience for users. The Tech Radar plugin provides a visually engaging element on the homepage or via navigation, and showcases the dynamic plugins loading mechanism for simple Developer Hub extension. The Quay plugin demonstrates integration with external services and provides a practical example of extending the Developer Hub software catalog.

5.5. RHDH LOCAL DEFAULT HOMEPAGE ENHANCEMENTS

With this update, RHDH Local default homepage integrates a floating action button with quick links for easy access to documentation and information.

5.6. ENHANCED CUSTOMIZATION AND COLLABORATION WITH PRE-LOADED TEMPLATES IN RHDH LOCAL

With this update, RHDH Local integrates pre-loaded essential templates, enabling users to create their own custom plugins and add TechDocs or software components to existing projects within Red Hat Developer Hub. This enhancement demonstrates Developer Hub extensibility, empowers users to extend the platform, and enhances the overall user experience by promoting collaboration and customization.

5.7. CONFIGURABLE KEY PERFORMANCE INDICATORS (KPIS) BY USING THE SCORECARD PLUGIN

With this release, Red Hat Developer Hub integrates the Scorecard plugin, enabling users to visualize software quality and health metrics directly in Developer Hub.

Developers can visualize scorecards with the metrics available to the their role in a tab on component detail pages, allowing them to quickly check the quality of their code and ensure it meets team standards.

Platform engineers can configure Key Performance Indicators (KPIs) from various data sources, with initial support for GitHub open pull requests and Jira open issues. Custom metric providers will allow for the addition of more data sources.

This simplifies the monitoring of applications by offering a single, clear report on health, security, and compliance.

Additional resources

- Understand and visualize Red Hat Developer Hub project health using Scorecards

5.8. OPENSHIFT AI CONNECTOR FOR RED HAT DEVELOPER HUB

You can use OpenShift AI Connector for RHDH to enable users to use Red Hat Developer Hub (RHDH) to surface AI Models and Model Servers from Red Hat OpenShift AI (RHOAI) directly into the RHDH/Backstage Catalog.

Additional resources

- OpenShift AI Connector for Red Hat Developer Hub
- Blogpost on OpenShift AI Connector for RHDH

5.9. INTERACTING WITH MODEL CONTEXT PROTOCOL TOOLS FOR RED HAT DEVELOPER HUB

You can enhance your Red Hat Developer Hub integration by leveraging the Model Context Protocol (MCP) server. This integration enables seamless communication with various Artificial Intelligence (AI) clients, facilitating efficient data exchange and expanding the functionality of the platform.

Additional resources

- Interacting with Model Context Protocol tools for Red Hat Developer Hub
- Blogpost on MCP in Red Hat Developer Hub

5.10. RED HAT DEVELOPER LIGHTSPEED FOR RED HAT DEVELOPER HUB NOW USES LIGHTSPEED CORE (LCORE)

The Developer Lightspeed for RHDH plugin has completed its migration from the Road-Core Service to Lightspeed Core (LCORE). This architectural change provides enhanced stability and prepares the plugin for future feature development during the Developer Preview.

Additional resources

- Red Hat Developer Lightspeed for Red Hat Developer Hub
- What is Llama Stack
- Lightspeed Core

CHAPTER 6. FIXED ISSUES

This section lists issues fixed in Red Hat Developer Hub 1.8.

6.1. FIXED ISSUES IN 1.8.0

6.1.1. Improved startup speed with updated init container image pull policy for Developer Hub Helm Chart

With this update, the pull policy for the init container image of the Developer Hub Helm Chart was changed from **Always** to **IfNotPresent**. This change reduces the repeated download time during startup of the container image, which is approximately 2.5 GB, thereby significantly improving startup speed for users.

Additional resources

- RHDHBUGS-1000

6.1.2. Improved Authentication for Self-Hosted Enterprise SCM Providers

Previously, actions requiring access to a self-hosted enterprise SCM provider failed, returning an error that no authentication provider was available for the specified host.

With this update, the SCM integration correctly identifies and uses the configured authentication provider for the corresponding enterprise host.

Additional resources

- RHDHBUGS-1028

6.1.3. Customizable image names for job and data index services in Developer Hub Helm Chart

Previously, when deploying the Developer Hub Helm Chart with the Orchestrator enabled, it was not possible to customize the image names of the job and data index services, for example in disconnected environments. Setting the **orchestrator.sonataflowPlatform.jobServiceImage** and **orchestrator.sonataflowPlatform.dataIndexImage** would return a schema validation error from Helm. This update fixes this issue.

Additional resources

- RHDHBUGS-2003

6.1.4. Network policy installation fix for Developer Hub RHDH Helm Chart with Orchestrator flavor

Before this update, the Developer Hub RHDH Helm Chart would not install Network Policies when the Orchestrator flavor was deployed with **serverlessLogicOperator** disabled, preventing the Developer Hub Pods from being completely available. This update fixes this situation by installing network policies unconditionally when the orchestrator is enabled.

Additional resources

- RHDHBUGS-2020

6.1.5. Resolved SonataFlow Pod Crash Issue

In the new release, a timing problem during the RHDH 1.7 installation with Orchestrator plugins, affecting SonataFlow database provisioning, has been addressed. This issue caused SonataFlow pods to repeatedly enter the **CrashLoopBackOff** state, leading to delays and potential confusion for users. With this update, SonataFlow pods no longer encounter the **CrashLoopBackOff** state due to the database provisioning delay. This improvement enhances the user experience, as SonataFlow pods now start promptly, eliminating unnecessary wait times.

Additional resources

- RHDHBUGS-2036

6.1.6. Fixed SCM integration failures for self-hosted enterprise SCM providers

Previously, SCM integration failed for self-hosted enterprise SCM providers because the system could not identify the configured host. This resulted in a *No auth provider available* error. With this release, the SCM integration now correctly uses the configured authentication provider for the corresponding enterprise host. As a result, end users can now successfully perform actions requiring enterprise SCM access.

Additional resources

- RHDHBUGS-2249

6.1.7. Fixed incorrect version detection for local plugin updates

Before this update, local plugins within the image were incorrectly detected for updates due to version misidentification. This led to users being unable to update local plugins from wrappers, resulting in outdated functionality. With this release, local plugin updates are now automatically detected, eliminating the need for manual adjustments and ensuring seamless plugin version updates for end users.

Additional resources

- RHDHBUGS-2250

CHAPTER 7. FIXED SECURITY ISSUES

You can view the security issues fixed in Red Hat Developer Hub 1.8 at Red Hat Security Updates .

- For 1.8.0, see Red Hat Security Advisory RHSA-2025:20047 .
- For 1.8.1, see Red Hat Security Advisory RHSA-2025:22861 .
- For 1.8.2, see Red Hat Security Advisory RHSA-2026:0531 .

CHAPTER 8. KNOWN ISSUES

This section lists known issues in Red Hat Developer Hub 1.8.

8.1. EXISTING OPERATOR USERS WITH ORCHESTRATOR 1.7 REQUIRE A MANUAL UPDATE IN THEIR DYNAMIC-PLUGINS CONFIGMAP

If you have an existing Operator-backed instance of Developer Hub with the Orchestrator, you must update your dynamic-plugins ConfigMap to set the version of the Orchestrator plugins to 1.8.2 once the Developer Hub Operator is upgraded to 1.8. Otherwise, the Developer Hub instance will not be upgraded at all.

Example of a dynamic-plugins ConfigMap enabling the Orchestrator plugins in 1.8 for Operator-backed instances

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: dynamic-plugins-rhdh
data:
  dynamic-plugins.yaml: |
    includes:
      - dynamic-plugins.default.yaml
    plugins:
      - package: "@redhat/backstage-plugin-orchestrator@1.8.2"
        disabled: false
      - package: "@redhat/backstage-plugin-orchestrator-backend-dynamic@1.8.2"
        disabled: false
        dependencies:
          - ref: sonataflow
      - package: "@redhat/backstage-plugin-scaffolder-backend-module-orchestrator-dynamic@1.8.2"
        disabled: false
      - package: "@redhat/backstage-plugin-orchestrator-form-widgets@1.8.2"
        disabled: false
```

Additional resources

- RHDHBUGS-2240

8.2. DEPLOYMENT UPDATE ERROR WITH DYNAMIC PLUGIN CONFIGURATION

Updating the deployment configuration using the **values.yaml** to include specific dynamic plugin configurations might cause an error during the deployment process.

When configuring the **dynamicRoutes** for the **red-hat-developer-hub.backstage-plugin-dynamic-home-page** plugin, the use of the placeholder {{firstName}} in a configuration property, for example title, can result in the following fatal deployment error:

"function **firstName** not defined".

This error prevents the cluster from spinning up correctly.

Configuration example

```
dynamicPlugins:
  frontend:
    red-hat-developer-hub.backstage-plugin-dynamic-home-page:
      dynamicRoutes:
        - path: /
          importName: DynamicHomePage
          config:
            props:
              title: &#39;Howdy {{firstName}} or {{displayName}}&#39;
```

Additional resources

- RHDHBUGS-2227

8.3. CLICKING ON THE DEDICATED RHDH LOCAL GUIDE LINK IN THE UI SIDEBAR ALSO HIGHLIGHTS THE CATALOG ITEM

In this update, the RHDH Local default configuration includes built-in TechDocs. However, when selecting the new *RHDH Local Guide* link in the RHDH Local UI sidebar, the *Catalog* link is also highlighted. There is currently no known workaround.

Additional resources

- RHDHBUGS-2132

8.4. HANDLE INSTALLATION DISABLED SCENARIO IN THE INSTALLED PACKAGES PAGE

When installation is disabled, the actions on the Installed Packages page are still shown. Similarly, if the YAML file is misconfigured, the actions appear, but the API call fails with an error. This does not break the UI, the API failure is handled gracefully, and the correct reason for the failure is displayed in the UI.

Additional resources

- RHDHBUGS-2126

8.5. CHANGES TO THE OPERATOR DEFAULT CONFIGURATION DO NOT PERSIST ACROSS OPERATOR UPGRADES

Changes to the Developer Hub Operator default configuration do not persist across operator upgrades. There is no known workaround.

Additional resources

- RHDHBUGS-2102

8.6. ERROR MESSAGE WHEN MANUALLY ACCESSING ACCESSING PLUGINS WITHOUT ASSOCIATED ENTITY YAML

This error occurs when a user tries to access a package or plugin that does not have an associated entity YAML. Users will not encounter this error under normal usage; it only appears if they manually modify the plugin or package name in the URL. This ticket will handle this scenario more gracefully by indicating why access to a particular plugin is not allowed.

Additional resources

- RHDHBUGS-2059

8.7. HIDE PACKAGE FOR ENTITIES MISSING DYNAMICARTIFACT VALUE IN CODE EDITOR

For packages with missing **spec.dynamicArtifact** value in their catalog entity, we currently show - package: ./dynamic-plugins/dis/….

Additional resources

- RHDHBUGS-2058

8.8. QUAY AND ARGO CD REQUIRE THEIR RESPECTIVE BACKEND PLUGINS TO CORRECTLY DISPLAY PERMISSIONS IN THE UI

Example configuration for Quay plugin:

```
plugins:
  - package: ./dynamic-plugins/dist/backstage-community-plugin-quay
    disabled: false
  - package: oci://ghcr.io/redhat-developer/rhdh-plugin-export-overlays/backstage-community-plugin-quay-backend:bs_1.42.5__1.6.0!backstage-community-plugin-quay-backend
    disabled: false
    pluginConfig:
      quay:
        apiUrl: ${QUAY_API_URL}
        apiKey: ${QUAY_API_KEY}
```

Example configuration for Argo CD plugin:

```
plugins:
  - package: ./dynamic-plugins/dist/roadiehq-scaffolder-backend-argocd-dynamic
    disabled: true
  - package: ./dynamic-plugins/dist/backstage-community-plugin-redhat-argocd
    disabled: false
  - package: oci://ghcr.io/redhat-developer/rhdh-plugin-export-overlays/backstage-community-plugin-redhat-argocd-backend:bs_1.42.5__0.10.0!backstage-community-plugin-redhat-argocd-backend
    disabled: false
    pluginConfig:
      argocd:
        username: &#34;${ARGOCD_USERNAME}&#34;
        password: &#34;${ARGOCD_PASSWORD}&#34;
        appLocatorMethods:
          - type: &#39;config&#39;
          instances:
            - name: argoInstance1
              url: &#34;${ARGOCD_INSTANCE1_URL}&#34;
```

For ArgoCD, you will need to update your **app-config.yaml** to add its id to **permission.rbac.pluginsWithPermission**:

```
permission:
  enabled: true
  rbac:
    pluginsWithPermission:
      - argocd
```

Additional resources

- RHDHBUGS-2038

8.9. MUI V5 COMPONENTS MIGHT RENDER WITH BROKEN STYLES IN NON-BUNDLED PLUGINS

Plugins that are not bundled with Red Hat Developer Hub, meaning not part of the Red Hat Developer Hub wrappers, might experience styling issues or broken visuals when using Material UI v5 components.

1. Workaround
   - Add the following code to the plugin’s entry file, such as **workspaces/<pluginId>/plugins/<pluginId>/src/index.ts**:

     ```
     import { unstable_ClassNameGenerator as ClassNameGenerator } from
     &#39;@mui/material/className&#39;;

     ClassNameGenerator.configure(componentName =&gt; {
       return componentName.startsWith(&#39;v5-&#39;)
         ? componentName
         : `v5-${componentName}`;
     });
     ```

Additional resources

- RHDHBUGS-986

8.10. OUT-OF-MEMORY ERRORS IN THE RED HAT DEVELOPER HUB OPERATOR

To resolve Out-Of-Memory errors in the Red Hat Developer Hub Operator, adjust memory settings manually:

1. Edit the deployment:

   ```
   kubectl edit deployment rhdh-operator -n &lt;namespace&gt;
   ```

2. In the editor, set the memory request to 1 GiB and the limit to 2 GiB:

```
spec:
  containers:
  - name: rhdh-operator
    resources:
      requests:
        memory: &#34;1G&#34;
      limits:
        memory: &#34;2G&#34;
```

3. Reapply this change whenever the Operator is actively reconciled. The pod should become healthy, using about 1.07 GiB of memory.

Additional resources

- RHDHBUGS-664