



# Red Hat build of Cryostat 3

## Release notes for the Red Hat build of Cryostat 3.0



# Red Hat build of Cryostat 3 Release notes for the Red Hat build of Cryostat 3.0

---

## Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

The Release notes for the Red Hat build of Cryostat 3.0 document provides an overview of new features in Cryostat 3.0 and a list of potential known issues and possible workarounds.

---

## Table of Contents

<b>PREFACE</b> .....	<b>3</b>
<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>4</b>
<b>CHAPTER 1. SUPPORT POLICY FOR CRYOSTAT</b> .....	<b>5</b>
<b>CHAPTER 2. NEW FEATURES</b> .....	<b>6</b>
Cryostat DB container	6
Cryostat Storage container	6
Reverse proxy architecture	6
Support for customizing the route host name	6
Dynamic attachment to the JVM	7
Support for launching the Cryostat agent as a standalone process	8
<b>CHAPTER 3. FEATURE ENHANCEMENTS</b> .....	<b>9</b>
Cryostat container enhancements	9
Cryostat API support for creating multi-namespace Cryostat instances	9
RBAC enhancements for accessing Cryostat	9
Cryostat CR validation enhancements	9
Cryostat Helm chart configuration enhancements	9
Cryostat agent embedded web server	10
Cryostat agent port enhancement	10
<b>CHAPTER 4. UNSUPPORTED AND DEPRECATED FEATURES</b> .....	<b>11</b>
Target TLS certificate upload	11
JMX target credentials passed through API requests	11
Cryostat self-discovery	11
Cryostat Operator installation for a single namespace	11
Cluster Cryostat API	11
<b>CHAPTER 5. KNOWN ISSUES</b> .....	<b>13</b>
Subviews such as /recordings/create are not accessible when visited directly in URL paths	13
Active Recordings table fails to update when restarting a recording with the replace=always parameter	13
<b>CHAPTER 6. ADVISORIES RELATED TO THIS RELEASE</b> .....	<b>14</b>

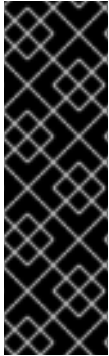


---

## PREFACE

The Red Hat build of Cryostat is a container-native implementation of JDK Flight Recorder (JFR) that you can use to securely monitor the Java Virtual Machine (JVM) performance in workloads that run on an OpenShift Container Platform cluster. You can use Cryostat 3.0 to start, stop, retrieve, archive, import, and export JFR data for JVMs inside your containerized applications by using a web console or an HTTP API.

Depending on your use case, you can store and analyze your recordings directly on your Red Hat OpenShift cluster by using the built-in tools that Cryostat provides or you can export recordings to an external monitoring application to perform a more in-depth analysis of your recorded data.



### IMPORTANT

Red Hat build of Cryostat is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).



## CHAPTER 1. SUPPORT POLICY FOR CRYOSTAT

Red Hat supports a major version of Cryostat for a minimum of 6 months. Red Hat bases this figure on the time that the product gets released on the Red Hat Customer Portal.

You can install and deploy Cryostat on Red Hat OpenShift Container Platform 4.12 or a later version that runs on an x86\_64 or ARM64 architecture.

### Additional resources

- For more information about the Cryostat life cycle policy, see [Red Hat build of Cryostat](#) on the Red Hat OpenShift Container Platform Life Cycle Policy web page.

## CHAPTER 2. NEW FEATURES

This section describes new features that the Crioostat 3.0 release provides.

### Crioostat DB container

From Crioostat 3.0 onward, the [Red Hat Ecosystem Catalog](#) also includes a Crioostat DB container image (**criostat-db**). When you install Crioostat by using either the Crioostat Operator or a Helm chart, the **criostat-db** container is also automatically deployed.

The **criostat-db** container provides a lightly customized Postgres database. Crioostat now uses this database for storing information such as the encrypted target JMX credentials keyring, Automated Rules definitions, discovered targets, and discovery plug-ins.

In previous releases, Crioostat stored information such as the JMX credentials keyring in a simple H2 file-based database, and other information such as Automated Rules definitions were stored directly as files on disk. With the introduction of the **criostat-db** container, Crioostat can now store different types of information in the same Postgres database.

### Crioostat Storage container

From Crioostat 3.0 onward, the [Red Hat Ecosystem Catalog](#) also includes a Crioostat Storage container image (**criostat-storage**). When you install Crioostat by using either the Crioostat Operator or a Helm chart, the **criostat-storage** container is also automatically deployed.

The **criostat-storage** container provides a lightly customized SeaweedFS storage solution that acts as an S3-compatible storage provider.

In previous releases, Crioostat used direct storage of files on disk for archived Flight Recordings and custom Event Templates. With the introduction of the **criostat-storage** container, Crioostat no longer needs to use direct file-system access for this type of information.

### Reverse proxy architecture

When you install Crioostat 3.0 by using either the Crioostat Operator or a Helm chart, Crioostat now includes a reverse proxy (**openshift-oauth-proxy** or **oauth2\_proxy**) in the pod. Only this proxy is exposed to cluster traffic through a service. This means that all API requests to Crioostat and all users of the Crioostat web console or Grafana dashboard are directed through the proxy.

The proxy handles user sessions to control access to the application, providing unified access control and user sessions for both the Crioostat web console and Grafana dashboard. Both of these user interfaces are accessible through the same route and present the same TLS certificate.

When deployed on Red Hat OpenShift, the proxy uses the Crioostat installation namespace to perform role-based access control (RBAC) checks for user authentication and authorization by integrating with the Red Hat OpenShift cluster SSO provider. You can optionally configure the auth proxy with an **htpasswd** file to enable Basic authentication. On Red Hat OpenShift, this allows for defining additional user accounts that can access Crioostat beyond those with Red Hat OpenShift SSO RBAC access.

### Support for customizing the route host name

By default, Red Hat OpenShift Container Platform automatically assigns a host name, based on the cluster's default ingress domain name, for any routes that do not specify a host. Depending on your requirements, you might want to use a particular host name for the route that the Crioostat Operator creates for your Crioostat deployment. In Crioostat 3.0, you can use a new **.spec.networkOptions.coreConfig.externalHost** property in the Crioostat custom resource (CR) to specify a custom host name for the Crioostat route.

In the Red Hat OpenShift console, you can access this property when creating your Crioostat CR:

Project: cryostat-test ▾

**Network Options** ▾

Options to control how the operator exposes the application outside of the cluster, such as using an Ingress or Route.

**Core Config** ▾

Specifications for how to expose the Cryostat service, which serves the Cryostat application.

**External Host**

cryostat.example.com

Externally routable host to be used to reach this Cryostat service. Used to define a Route's host on OpenShift when it is first created. On Kubernetes, define this using "spec.ingressSpec".

**Ingress Spec** >

Configuration for an Ingress object. Currently subpaths are not supported, so unique hosts must be specified (if a single external IP is being used) to differentiate between ingresses/services.

Alternatively, you can create your Cryostat CR in YAML format. For example:

```
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
spec:
  networkOptions:
    coreConfig:
      externalHost: cryostat.example.com
```

Once a route is created in Red Hat OpenShift Container Platform, you cannot change the route's host name. If you need to change the route's host name after you have already created your Cryostat CR, you must delete the Cryostat CR and create a new CR with the modified host name.

## Dynamic attachment to the JVM

From Cryostat 3.0 onward, the Cryostat agent can attach dynamically to an application JVM that is already running without requiring an application restart. This dynamic attachment feature has the following requirements:

- You must ensure that the agent's JAR file is copied to the JVM's file system (for example, by using the **oc cp** command).
- You must be able to run the agent as a separate process on the same host or within the same application (for example, by using the **oc exec** command).

Dynamic attachment supports ad hoc one-time profiling or troubleshooting workflows where you might not need the agent to be attached every time the JVM starts. Dynamic attachment also suits situations where you cannot or do not want to reconfigure your application for the sole purpose of attaching the agent. Because the agent can attach to a running JVM without requiring an application restart, this also means there is no application downtime.



## NOTE

In previous releases, your only option was to enable your application's JVM to load and initialize the Cryostat agent at JVM startup. This requires that you configure the application to pass the **-javaagent** JVM flag with the path to the Cryostat agent's JAR file. Depending on your requirements, you can continue to use this type of static attachment to the JVM.

## Support for launching the Cryostat agent as a standalone process

From Cryostat 3.0 onward, if you want the Cryostat agent to attach dynamically to an application JVM that is already running, you can launch the agent as a standalone Java process. This feature requires that you have already copied the agent's JAR file to the JVM's file system (for example, by using the **oc cp** command).

To launch the agent, you can run the following command, where **<agent\_jar\_file>** represents the agent's JAR file name and **<pid>** represents the process ID (PID) of the JVM:

```
$ java -jar target/<agent_jar_file> <pid>
```

For example:

```
$ java -jar target/cryostat-agent-0.4.0.jar 1234
```

The agent process uses its Attach providers to look up the specified PID. If the specified PID is found, the agent process attaches to this PID and attempts to load the agent's JAR file into this JVM, which then bootstraps into the normal agent launch process.

You can also specify additional late-binding configuration options to the agent launcher by using command-line options. For example:

```
$ java -jar target/cryostat-agent-0.4.0.jar \  
-Dcryostat.agent.baseuri=http://cryostat.local \  
--smartTrigger=[ProcessCpuLoad>0.2]~profile \  
@/deployment/app/moreAgentArgs \  
1234
```

For more information about the available options and their behavior, run the **java -jar target/cryostat-agent-0.4.0.jar -h** help command. System properties that you specify with **-D** are set onto the host JVM before the injected agent attempts to read the configuration values. This has the same effect as setting these system properties or equivalent environment variables on the host JVM process itself.

## CHAPTER 3. FEATURE ENHANCEMENTS

Cryostat 3.0 includes feature enhancements that build upon the Cryostat 2.4 offerings.

### Cryostat container enhancements

In this release, the main Cryostat container (**cryostat**) has been reimplemented by using Quarkus. In previous releases, the **cryostat** container was built directly by using Eclipse Vert.x.

This enhancement allows Cryostat to take advantage of the Quarkus framework's strengths and features, which enables Cryostat to provide higher performance access to your JDK Flight Recorder data as well as better data integrity and security.

### Cryostat API support for creating multi-namespace Cryostat instances

From Cryostat 3.0 onward, the **Cryostat** API supports the creation of both single-namespace and multi-namespace Cryostat instances. When you install a Cryostat instance by using the Cryostat Operator, the **Cryostat** API now enables you to specify an optional list of target namespaces. This supersedes the behavior in previous releases where the **Cryostat** API supported the creation of single-namespace instances only.



#### NOTE

In previous releases, you could use the **Cluster Cryostat** API to create multi-namespace Cryostat instances. Cryostat 3.0 no longer provides a separate **Cluster Cryostat** API.

### RBAC enhancements for accessing Cryostat

Cryostat now applies the same role-based access control (RBAC) permission check to all users for the purpose of permitting or denying access to the product. By default, the required RBAC role in the Cryostat application's installation namespace is **create pods/exec**.

Any Red Hat OpenShift user accounts that are assigned the required RBAC role now have full access to the Cryostat web console and all Cryostat features. If a Red Hat OpenShift account does not have the required RBAC role, this user is blocked from accessing Cryostat.

When installing a Cryostat instance by using the Cryostat Operator, you can optionally use the **.spec.authorizationOptions.openShiftSSO.accessReview** field in the Cryostat custom resource (CR) to customize the required RBAC permissions for accessing Cryostat.

This enhancement supersedes the behavior in previous releases where you could configure different levels of authorization for different user accounts.

### Cryostat CR validation enhancements

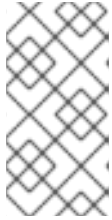
The Cryostat Operator now performs additional validation checks against Cryostat CR objects before accepting these objects for processing. One noteworthy validation check is that a user who creates a Cryostat CR with a list of target namespaces must have sufficient permissions to create single-namespace Cryostat CRs in these target namespaces.

### Cryostat Helm chart configuration enhancements

You can now set the following configuration parameters for the Cryostat Helm chart:

- **authentication.openshift.enabled**

This property enables the deployment of **openshift-oauth-proxy** and is disabled by default. If this property is disabled, **oauth2\_proxy** is deployed instead.



## NOTE

You can configure both **openshift-oauth-proxy** and **oauth2-proxy** to enable basic authentication by using the **authentication.basicAuth property**. However, for users who are deploying Cryostat on Red Hat OpenShift, the **openshift-oauth-proxy** also supports integration with the Red Hat OpenShift cluster SSO.

- **authentication.basicAuth**

This property configures basic authentication on the auth proxy. If you enable the deployment of **openshift-oauth-proxy**, this basic authentication is in addition to the Red Hat OpenShift SSO. If you enable the deployment of **oauth2\_proxy**, this basic authentication is the only out-of-the-box supported user authentication mechanism.

- **openshiftOauthProxy.accessReview**

This property configures the **SubjectAccessReview** for testing client access to Cryostat through Red Hat OpenShift SSO.

For a full list of configuration parameters, see the [Cryostat Helm Chart readme file](#) .

## Cryostat agent embedded web server

At Cryostat agent startup, the agent starts an embedded web server, which is used to service requests from the Cryostat server. The embedded web server secures itself by using basic authentication.

In previous releases, the basic user name was always **user**, and the randomly generated password consisted of 24 ASCII characters. In Cryostat 3.0, the default user name is **user** and the default password length is 24 characters, but the user name and the password length are both configurable. In this release, the randomly generated password is also based on a larger character set compared to previous releases.

## Cryostat agent port enhancement

When configuring your applications to use the Cryostat agent, the agent base URI now uses port 4180 by default. This supersedes the behavior in previous releases where the agent base URI used port 8181.

This enhancement is due to the introduction of the [reverse proxy architecture](#) in Cryostat 3.0. Port 4180 is the HTTP port of the auth proxy, which passes authorized requests to Cryostat. You must therefore configure the Cryostat agent to send requests to port 4180 rather than directly to port 8181, because the Cryostat HTTP port is now hidden behind the proxy.

## CHAPTER 4. UNSUPPORTED AND DEPRECATED FEATURES

Cryostat 3.0 removes some features because of their high maintenance costs, low community interest, and better alternative solutions.

### Target TLS certificate upload

The Security view of the Cryostat web console no longer offers a way to upload SSL/TLS certificates directly into the Cryostat server truststore. The Security view now only displays a list of certificates that have already been loaded.

From Cryostat 3.0 onward, new certificates must be added to the storage volume that Cryostat reads at startup. You can configure any new certificates by using the existing **TrustedCertSecrets** property in the Cryostat CR.

### JMX target credentials passed through API requests

The **X-JMX-Authorization** header is no longer supported. This means that Cryostat no longer accepts API requests from target applications to allow Cryostat to authenticate itself and store credentials in memory for the duration of a JMX connection to an application.

From Cryostat 3.0 onward, JMX credentials for target applications are always stored in an encrypted database that is stored on a persistent volume claim (PVC) on Red Hat OpenShift. The Settings view of the Cryostat web console also no longer offers an advanced configuration for selecting which authentication mechanism to use.

### Cryostat self-discovery

When you install Cryostat by using the Cryostat Operator or a Helm chart, Cryostat no longer discovers itself as a target application by default. In previous releases, Cryostat exposed a JMX port on a Kubernetes service and the Cryostat Operator generated credentials and assigned a TLS certificate to help secure this port.

From Cryostat 3.0 onward, the JMX port exposed by Cryostat is disabled and the corresponding Service port is removed, which means that Cryostat can no longer discover itself as a connectable target. This also means that Cryostat no longer appears in the target selection list or in the Topology view of the Cryostat web console.



#### NOTE

If you want to connect Cryostat to itself to check performance, you can create a Custom Target with the URL value **localhost:0**. This value instructs the JVM to open a local JMX connection to itself, without exposing a port to the network, which means that additional authentication and TLS encryption is unnecessary.

### Cryostat Operator installation for a single namespace

Support is no longer provided for installing the Cryostat Operator in a single namespace or subset of cluster namespaces.

From Cryostat 3.0 onward, the Cryostat Operator can only be installed on a cluster-wide basis. Cluster-wide installation is the preferred mode for the Operator Lifecycle Manager and per-namespace installations are a deprecated feature.

### Cluster Cryostat API

The **Cluster Cryostat** API is no longer supported. In this release, when installing a Cryostat instance by using the Cryostat Operator, you can no longer select a **Cluster Cryostat** option in the the Provided APIs section of the **Details** tab.

From Crio 3.0 onward, you can use the **Crio** API to create both single-namespace and multi-namespace Crio instances. When you install a Crio instance by using the Crio Operator, the **Crio** API now enables you to specify an optional list of target namespaces.



## CHAPTER 5. KNOWN ISSUES

Sometimes a Cryostat release might contain an issue or issues that Red Hat acknowledges and might fix at a later stage during the product's development. Review each known issue for its description and its resolution.

### Subviews such as `/recordings/create` are not accessible when visited directly in URL paths

#### Description

The Cryostat web console is not accessible from a web browser when you attempt to access the console through a URL path that includes a subview such as `/topology/create-custom-target`, `/rules/create`, or `/recordings/create`.

For example, if you enter a URL path such as `https://my_cryostat_domain/recordings/create`, the Cryostat console shows a blank page.

#### Workaround

Do not specify subviews in URL paths.

For example, if you want to access `https://my_cryostat_domain/recordings/create`, enter a URL path of `https://my_cryostat_domain/recordings` in your web browser, and then click **Create** in the Cryostat web console.

### Active Recordings table fails to update when restarting a recording with the `replace=always` parameter

#### Description

If a client sends a request that includes the `replace=always` parameter to recreate an existing recording, the Active Recordings table in the Cryostat web console is not updated to show details of the new recording. Even though a **Recording created** notification is displayed, the new recording does not automatically appear in the Active Recordings table.

#### Workaround

Reload the Active Recordings page or navigate away from then back to the current page. The Active Recordings table then correctly displays the new recording.

## CHAPTER 6. ADVISORIES RELATED TO THIS RELEASE

The following advisory has been issued to document bug fixes and CVE fixes included in the Cryostat 3.0 release:

- [RHSA-2024:4173](#)

*Revised on 2024-07-02 13:31:29 UTC*