Red Hat build of Cryostat 3

Using the Red Hat build of Cryostat Operator to configure Cryostat

Last Updated: 2024-07-02

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 ® is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js ® 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 ® 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

Red Hat build of Cryostat is a Red Hat offering on OpenShift Container Platform. Use the Using the Red Hat build of Cryostat Operator to configure Cryostat to learn how to use the Red Hat build of Cryostat Operator to configure Cryostat.

Table of Contents

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. RED HAT BUILD OF CRYOSTAT OPERATOR

You can use the Red Hat build of Cryostat Operator to manage and configure your Cryostat instance. The Red Hat build of Cryostat Operator is available on the OpenShift Container Platform (OCP).

1.1. OVERVIEW OF THE RED HAT BUILD OF CRYOSTAT OPERATOR

After you create or update a Cryostat application on the OpenShift Container Platform, the Red Hat build of Cryostat Operator creates and manages the Cryostat application.

Operator level 2 seamless upgrades

The Operator Capability Level for the Red Hat build of Cryostat Operator is set to **Level 2 Seamless Upgrades** on the Operator Lifecycle Manager framework. After you upgrade your Red Hat build of Cryostat Operator, the Red Hat build of Cryostat Operator automatically upgrades Cryostat and its related components. The automatic upgrade operation does not remove any JFR recordings, templates, rules, and other stored components, from your Cryostat instance.

> NOTE
>
> The automatic upgrade operation occurs only for minor releases or patch update releases of Cryostat. For major releases, you might need to re-install the Red Hat build of Cryostat Operator.

Persistent volume claims

You can create persistent volume claims (PVCs) on Red Hat OpenShift with the Red Hat build of Cryostat Operator so that your Cryostat application can store archived recordings on a cloud storage disk.

Operator configuration settings

Additionally, you can make the following changes to the default configuration settings for the Red Hat build of Cryostat Operator:

- Configure the PVC that was created by the Red Hat build of Cryostat Operator, so that your Cryostat application can store archived recordings on a cloud storage disk.
- Configure your Cryostat application to trust TLS certificates from specific applications.
- Disable cert-manager, so that the operator does not need to generate self-signed certificates for Cryostat components.
- Install custom event template files, which are located in ConfigMaps, to your Cryostat instance, so you can use the templates to create recordings when Cryostat starts.

The following configuration options for the Red Hat build of Cryostat Operator are included:

- Resource requirements, which you can use to specify resource requests or limits for the **core**, **datasource**, **grafana**, **storage**, **db**, or **auth-proxy** containers.
- Service customization, so that you can control the services that the Red Hat build of Cryostat Operator creates.
- Sidecar report options, which the Red Hat build of Cryostat Operator can use to provision one or more report generators for your Cryostat application.

Single-namespace or multi-namespace Cryostat instances

The Red Hat build of Cryostat Operator provides a **Cryostat** API that you can use to create Cryostat instances that work in a single namespace or across multiple namespaces. You can control these Cryostat instances by using a GUI that is accessible from the Red Hat OpenShift web console.

NOTE

From Cryostat 3.0, the **Cryostat** API supports the creation of both single-namespace and multi-namespace instances. The **Cluster Cryostat** API that you could use to create multi-namespace instances in Cryostat 2.x releases is deprecated and superseded by the **Cryostat** API in Cryostat 3.x.

Users who can access the multi-namespace Cryostat instance have access to all target applications in any namespace that is visible to that Cryostat instance. Therefore, when you deploy a multi-namespace Cryostat instance, you must consider which namespaces to select for monitoring, which namespace to install Cryostat into, and which users can have access rights.

Prerequisites for configuring the Red Hat build of Cryostat Operator

Before you configure the Red Hat build of Cryostat Operator, ensure that the following prerequisites are met:

- Installed the Red Hat build of Cryostat Operator in a project on Red Hat OpenShift.
- Created a Cryostat instance by using the Red Hat build of Cryostat Operator.

Additional resources

- See Operator Capability Levels (Operator SDK)
- See Installing Cryostat on Red Hat OpenShift using an operator (Installing Cryostat)

1.2. DISABLING CERT-MANAGER

You can disable cert-manager functionality by configuring the **enableCertManager** property of the Red Hat build of Cryostat Operator.

By default, Red Hat build of Cryostat Operator's **enableCertManager** property is set to **true**. This means that the Red Hat build of Cryostat Operator uses the cert-manager **CA** issuer to generate self-signed certificates for your Cryostat components. The Red Hat build of Cryostat Operator uses these certificates to enable HTTPS communication among Cryostat components operating in a cluster.

You can set the **enableCertManager** property to **false**, so that the Red Hat build of Cryostat Operator does not need to generate self-signed certificates for Cryostat components.

IMPORTANT

If you set the **enableCertManager** property to **false**, you could introduce potential security implications from unencrypted internal traffic to the cluster that contains your running Cryostat application.

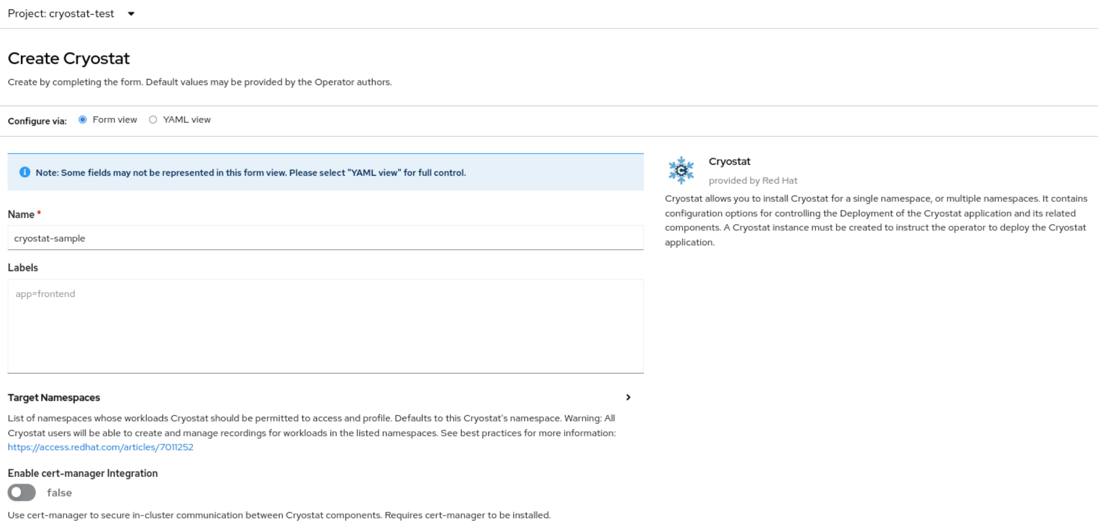
Prerequisites

- Logged in to the OpenShift Container Platform by using the Red Hat OpenShift web console.

Procedure

1. If you want to start creating a Cryostat instance, perform the following steps:
   1. On your Red Hat OpenShift web console, click **Operators** > **Installed Operators**.
   2. From the list of available Operators, select Red Hat build of Cryostat.
   3. On the **Operator details** page, click the **Details** tab.
   4. In the **Provided APIs** section, select Cryostat, and then click **Create instance**.
2. On the Create Cryostat panel, to configure the **enableCertManager** property, choose one of the following options:
   1. If you want to use the Form view:
      1. Click the **Form view** radio button.
      2. Set the **Enable cert-manager Integration** switch to **false**, and then enter a value in the **Name** field.

Figure 1.1. Toggling the Enable cert-manager Integration switch to false



Project: cryostat-test ▾

Create Cryostat

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: ● Form view ○ YAML view

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name *

cryostat-sample

Labels

app=frontend

Target Namespaces

List of namespaces whose workloads Cryostat should be permitted to access and profile. Defaults to this Cryostat's namespace. Warning: All Cryostat users will be able to create and manage recordings for workloads in the listed namespaces. See best practices for more information: https://access.redhat.com/articles/7011252

Enable cert-manager Integration

false

Use cert-manager to secure in-cluster communication between Cryostat components. Requires cert-manager to be installed.

Cryostat

provided by Red Hat

Cryostat allows you to install Cryostat for a single namespace, or multiple namespaces. It contains configuration options for controlling the Deployment of the Cryostat application and its related components. A Cryostat instance must be created to instruct the operator to deploy the Cryostat application.

   2. If you want to use the YAML view:
      1. Click the **YAML view** radio button.
      2. In the **spec:** key set of the YAML file, change the **enableCertManager** property to **false**.

Example of configuring the spec: key set in a YAML file

```
--
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
spec:
  enableCertManager: false
--
```

3. If you want to configure other properties in the custom resource (CR) for this Cryostat instance, see the other sections of this document for more information about these properties.

4. If you want to finish creating this Cryostat instance, click **Create**.

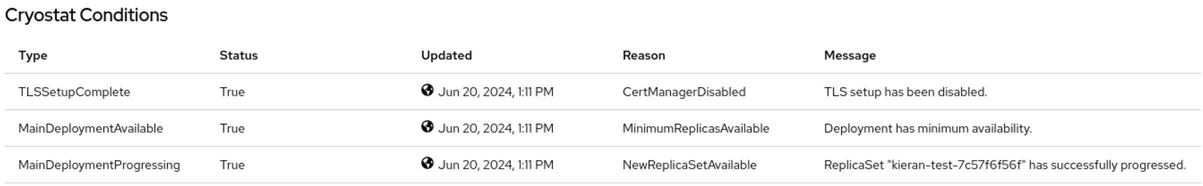
When you click **Create**, this Cryostat instance is available under the **Cryostat** tab on the **Operator details** page. You can subsequently edit the CR properties for a Cryostat instance by clicking the instance name on the **Operator details** page and then select **Edit Cryostat** from the **Actions** drop-down menu.

The Red Hat build of Cryostat Operator automatically restarts your Cryostat application, enabling the application to run with the updated **enableCertManager** property configuration.

Verification

1. Select your Cryostat instance from the **Cryostat** tab on the **Operator details** page.
2. Navigate to the **Cryostat Conditions** table.
3. Verify that the **TLSSetupComplete** condition is set to **true** and that the **Reason** column for this condition is set to **CertManagerDisabled**. This indicates that you have set the **enableCertManager** property to **false**.

Figure 1.2. Example showing the TLSSetupComplete condition set to true



Cryostat Conditions

Type	Status	Updated	Reason	Message
TLSSetupComplete	True	Jun 20, 2024, 1:11 PM	CertManagerDisabled	TLS setup has been disabled.
MainDeploymentAvailable	True	Jun 20, 2024, 1:11 PM	MinimumReplicasAvailable	Deployment has minimum availability.
MainDeploymentProgressing	True	Jun 20, 2024, 1:11 PM	NewReplicaSetAvailable	ReplicaSet "kieran-test-7c57f6f56f" has successfully progressed.

Additional resources

- See the cert-manager documentation
- See Creating a JDK Flight Recorder (JFR) recording (Creating a JFR recording with Cryostat)

1.3. CUSTOMIZING EVENT TEMPLATES

You can configure the **eventTemplates** property of the Red Hat build of Cryostat Operator YAML configuration file to include multiple custom templates. An event template outlines the event recording criteria for your JDK Flight Recording (JFR). You can configure a JFR through its associated event template.

By default, Red Hat build of Cryostat Operator includes some pre-configured event templates. These pre-configured event templates might not meet your needs, so you can use Red Hat build of Cryostat Operator to generate custom event templates for your Cryostat instance and store these templates in ConfigMaps for easier retrieval. You can generate a custom event template in the following ways:

- Use the Red Hat OpenShift web console to upload an event template into a custom resource.
- Edit the YAML file for your Cryostat custom resource on the Red Hat OpenShift web console.

After you store a custom event template in a **ConfigMap**, you can deploy a new Cryostat instance with this custom event template. You can then use your custom event template with JFR to monitor your Java application to meet your needs.

Prerequisites

- Logged in to the OpenShift Container Platform by using the Red Hat OpenShift web console.
- Logged in to your Cryostat web console.

Procedure

1. To download a default event template, navigate to your Cryostat web console and from the **Events** menu, click **Downloads**.

   NOTE

   Event templates are in XML format and have a file name extension of **.jfc**.

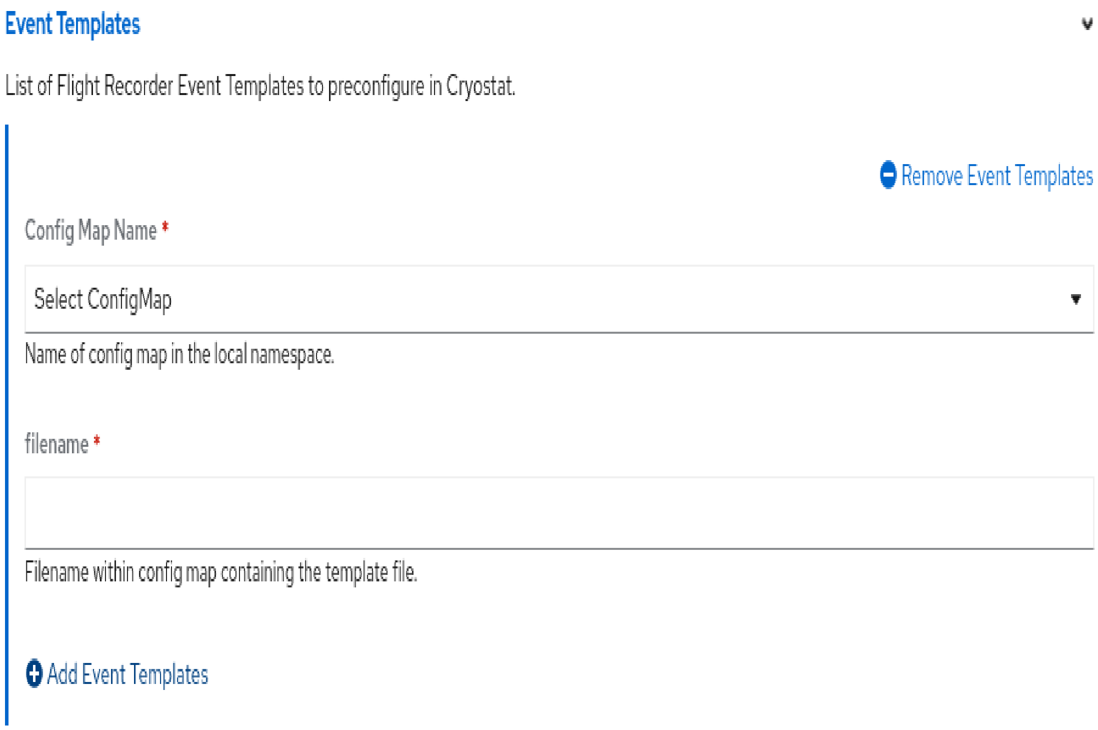
2. *Optional:* If you want a custom event template, edit the downloaded default event template by using a text editor or XML editor to configure the template to meet your needs.
3. Log in to your Red Hat OpenShift web console by entering the **oc login** command in your CLI.
4. Create a **ConfigMap** resource from the event template by entering the following command in your CLI. You must issue the command in the path where you want to deploy your Cryostat application. You can use this resource to store an event template file that is inside the cluster where you run your Cryostat instance.

   ### Example of creating a ConfigMap resource by using the CLI

   ```
   $ oc create configmap <template_name> --from-file=<path_to_custom_event_template>
   ```

5. If you want to start creating a Cryostat instance, perform the following steps:
   a. On your Red Hat OpenShift web console, click **Operators > Installed Operators**.
   b. From the list of available Operators, select Red Hat build of Cryostat.
   c. On the **Operator details** page, click the **Details** tab.
   d. In the **Provided APIs** section, select Cryostat, and then click **Create instance**.
6. On the Create Cryostat panel, to upload an event template in XML format into a resource, choose one of the following options:
   a. If you want to use the Form view:
      i. Click the **Form view** radio button.
      ii. Navigate to the **Event Templates** section of the Cryostat instance.
      iii. From the **Event Templates** menu, click **Add Event Template**. An **Event Templates** section opens on your Red Hat OpenShift console.
      iv. From the **Config Map Name** drop-down list, select the ConfigMap resource that contains your event template.

Figure 1.3. Event Templates option for a Cryostat instance



Event Templates

List of Flight Recorder Event Templates to preconfigure in Cryostat.

Config Map Name *

Select ConfigMap

Name of config map in the local namespace.

filename *

Filename within config map containing the template file.

v. In the **Filename** field, enter the name of the **.jfc** file that is contained within your ConfigMap.

b. If you want to use the YAML view:

i. Click the **YAML view** radio button.

ii. Specify any custom event templates for the **eventTemplates** property. This property points the Red Hat build of Cryostat Operator to your ConfigMap, so that the Red Hat build of Cryostat Operator can read the event template.

Example of specifying custom event templates for the eventTemplates property

```
--
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
spec:
  eventTemplates:
  - configMapName: custom-template1
    filename: my-template1.jfc
  - configMapName: custom-template2
    filename: my-template2.jfc
--
```

IMPORTANT

You must select the name of a ConfigMap, which is associated with your Cryostat or Cluster Cryostat instance, from the **configMapName** drop-down list. Additionally, you must specify a key associated with the ConfigMap in the **filename** field.

7. If you want to configure other properties in the custom resource (CR) for this Cryostat instance, see the other sections of this document for more information about these properties.

8. If you want to finish creating this Cryostat instance, click **Create**.

When you click **Create**, this Cryostat instance is available under the **Cryostat** tab on the **Operator details** page. You can subsequently edit the CR properties for a Cryostat instance by clicking the instance name on the **Operator details** page and then select **Edit Cryostat** from the **Actions** drop-down menu.

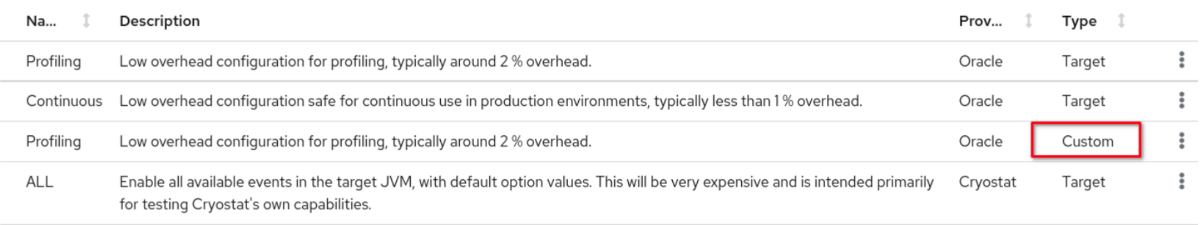
The Red Hat build of Cryostat Operator can now provide the custom event template as an XML file to your Cryostat application. Your custom event template opens alongside default event templates in your Cryostat web console.

Verification

1. On the Cryostat web console, click **Events** from the menu. If an **Authentication Required** window opens on your web console, enter your credentials and click **Save**.

2. Under the **Event Templates** tab, check if your custom event template shows in the list of available event templates.

Figure 1.4. Example of a listed custom event template under the Event Templates tab



Na...	Description	Prov...	Type
Profiling	Low overhead configuration for profiling, typically around 2 % overhead.	Oracle	Target
Continuous	Low overhead configuration safe for continuous use in production environments, typically less than 1 % overhead.	Oracle	Target
Profiling	Low overhead configuration for profiling, typically around 2 % overhead.	Oracle	Custom
ALL	Enable all available events in the target JVM, with default option values. This will be very expensive and is intended primarily for testing Cryostat's own capabilities.	Cryostat	Target

Additional resources

- See Installing Cryostat on OpenShift using an operator (Installing Cryostat)
- See Accessing Cryostat by using the web console (Installing Cryostat)
- See Using custom event templates (Using Cryostat to manage a JFR recording)

1.4. CONFIGURING TLS CERTIFICATES

You can specify the Red Hat build of Cryostat Operator to configure Cryostat to trust TLS certificates from specific applications.

Cryostat attempts to open a JMX connection to a target JVM that uses a TLS certificate. For a successful JMX connection, the Cryostat must pass all its authentication checks on the target JVM certificate.

You can specify multiple TLS secrets in the **trustedCertSecrets** array of the Red Hat build of Cryostat Operator YAML configuration file. You must specify the secret located in the same namespace as your Cryostat application in the **secretName** property of the array. The **certificateKey** property defaults to **tls.crt**, but you can change the value to an X.509 certificate file name.

IMPORTANT

Configuring a TLS certificate is required only for applications that have enabled TLS for remote JMX connections by using the **com.sun.management.jmxremote.registry.ssl=true** attribute.

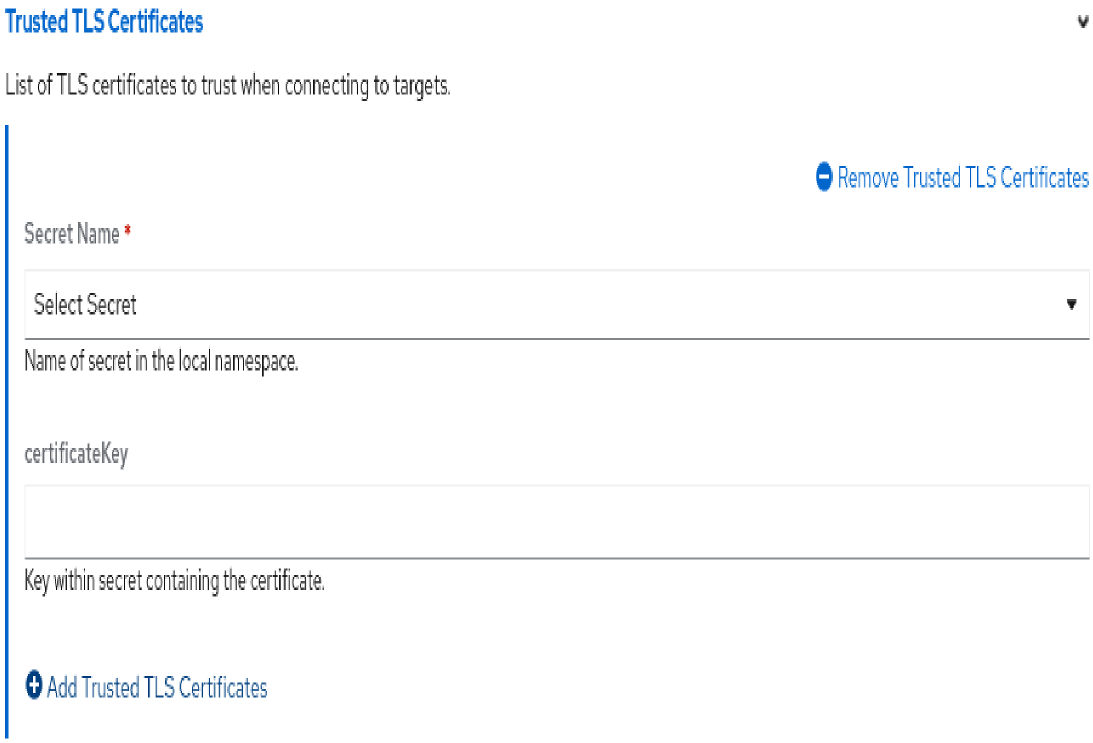
Prerequisites

- Logged in to the OpenShift Container Platform by using the OpenShift web console.
- Logged in to your Cryostat web console.

Procedure

1. If you want to start creating a Cryostat instance, perform the following steps:
   a. On your Red Hat OpenShift web console, click **Operators > Installed Operators**.
   b. From the list of available Operators, select Red Hat build of Cryostat.
   c. On the **Operator details** page, click the **Details** tab.
   d. In the **Provided APIs** section, select Cryostat, and then click **Create instance**.
2. On the Create Cryostat panel, to configure a TLS certificate, choose one of the following options:
   a. If you want to use the Form view:
      i. Click the **Form view** radio button.
      ii. In the **Name** field, specify a name for the instance of Cryostat that you want to create.
      iii. Expand the **Trusted TLS Certificates** option, then click **Add Trusted TLS Certificates**. A list of options displays on your Red Hat OpenShift web console.

Figure 1.5. The Trusted TLS Certificates option



Trusted TLS Certificates

List of TLS certificates to trust when connecting to targets.

Secret Name *

Select Secret

Name of secret in the local namespace.

certificateKey

Key within secret containing the certificate.

iv. Select a TLS secret from the **Secret Name** list. The **Certificate Key** field is optional.

> **NOTE**
>
> You can remove a TLS certificate by clicking **Remove Trusted TLS Certificates**.

b. If you want to use the YAML view:

i. Click the **YAML view** radio button.

ii. Specify your secret, which is located in the same namespace as your Cryostat application, in the **secretName** property of the **trustedCertSecrets** array.

Example of specifying a secret in the **trustedCertSecrets** array

```
--
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
spec:
  trustedCertSecrets:
  - secretName: my-tls-secret
--
```

iii. *Optional:* Change the **certificateKey** property value to the application’s X.509 certificate file name. If you do not change the value, the **certificateKey** property defaults to **tls.crt**.

Example of changing the **certificateKey** property’s value

```
--
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
spec:
  trustedCertSecrets:
  - secretName: my-tls-secret
    certificateKey: ca.crt
--
```

3. If you want to configure other properties in the custom resource (CR) for this Cryostat instance, see the other sections of this document for more information about these properties.

4. If you want to finish creating this Cryostat instance, click **Create**.

When you click **Create**, this Cryostat instance is available under the **Cryostat** tab on the **Operator details** page. You can subsequently edit the CR properties for a Cryostat instance by clicking the instance name on the **Operator details** page and then select **Edit Cryostat** from the **Actions** drop-down menu.

The Red Hat build of Cryostat Operator automatically restarts your Cryostat instance with the configured security settings.

Verification

1. Determine that all your application pods exist in the same OpenShift cluster namespace as your Cryostat pod by issuing the following command in your CLI:

   ```
   $ oc get pods
   ```

2. Log in to the web console of your Cryostat instance.

3. On the **Dashboard** menu for your Cryostat instance, select a target JVM from the **Target** list.

4. In the navigation menu on the Cryostat web console, select **Recordings**. On the **Authentication Required dialog** window, enter your secret's credentials and then select **Save** to provide your credentials to the target JVM.

   NOTE

   If the selected target has password authentication enabled for JMX connections, you must provide the JMX credentials for the target JVM when prompted for a connection.

   Cryostat connects to your application through the authenticated JMX connection. You can now use the **Recordings** and **Events** functions to monitor your application's JFR data.

Additional resources

- See Creating a JDK Flight Recorder (JFR) recording (Creating a JFR recording with Cryostat)

- See Installing Cryostat on Red Hat OpenShift using an operator (Installing Cryostat)

- See Accessing Cryostat by using the web console (Installing Cryostat)

1.5. CHANGING STORAGE VOLUME OPTIONS

You can use the Red Hat build of Cryostat Operator to configure storage volumes for your Cryostat or Cluster Cryostat instance. Cryostat supports persistent volume claim (PVC) and **emptyDir** storage volume types.

By default, Red Hat build of Cryostat Operator creates a PVC for your Cryostat or Cluster Cryostat instance that uses the default **StorageClass** resource with 500 mebibytes (MiB) of allocated storage.

You can create a custom PVC for your Cryostat application on OpenShift Container Platform by choosing one of the following options:

- Navigating to **Storage Options > PVC > Spec**in the **Form view** window, and then customizing your PVC by completing the relevant fields.
- Navigating to the **YAML view** window, and then editing the **storageOptions** array in the **spec: key** set to meet your needs.

NOTE

You can learn more about creating a custom PVC by navigating to Changing storage volume options in the *Using the Red Hat build of Cryostat Operator to configure Cryostat* guide.

You can configure the **emptyDir** storage volume for your Cryostat application on OpenShift Container Platform by choosing one of the following options:

- Enabling the **Empty Dir** setting in **Storage Options** on the **Form view** window.
- Setting the **spec.storageOptions.emptyDir.enabled** to **true** in the **YAML view** window.

Prerequisites

- Logged in to the OpenShift Container Platform by using the Red Hat OpenShift web console.

Procedure

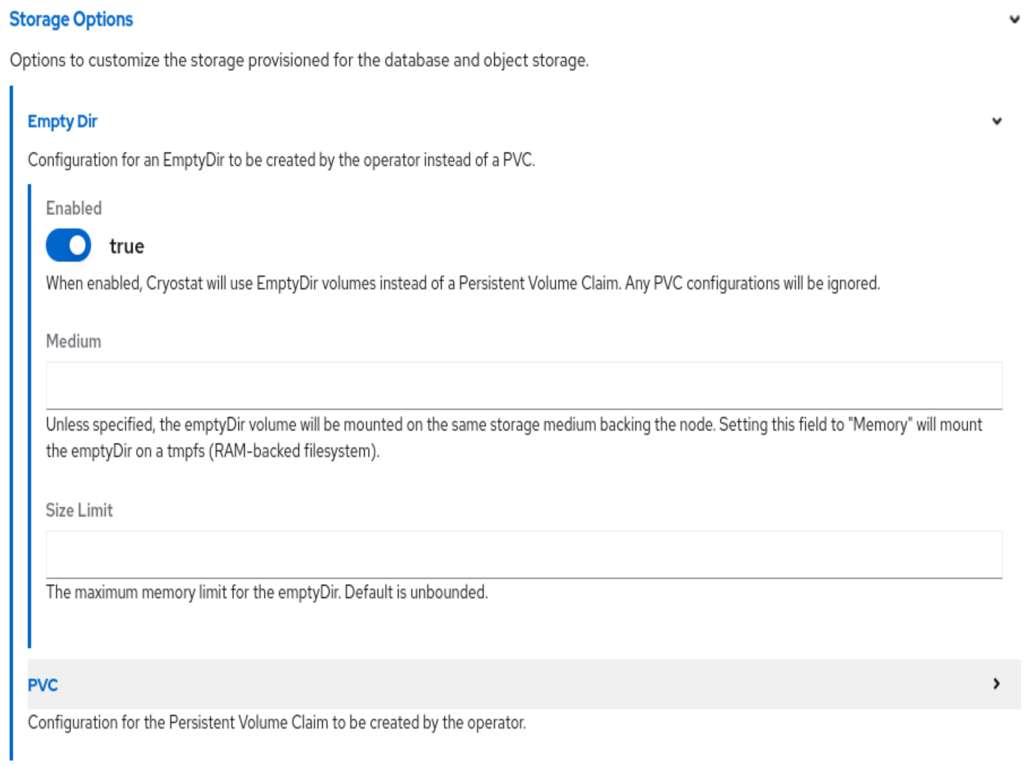
1. If you want to start creating a Cryostat instance, perform the following steps:
   a. On your Red Hat OpenShift web console, click **Operators > Installed Operators**.
   b. From the list of available Operators, select Red Hat build of Cryostat.
   c. On the **Operator details** page, click the **Details** tab.
   d. In the **Provided APIs** section, select Cryostat, and then click **Create instance**.
2. On the Create Cryostat panel, to change storage settings for your Cryostat application, choose one of the following options:
   a. If you want to use the Form view:
      i. Click the **Form view** radio button.

ii. Navigate to the **Storage Options** section, and enter a value in the **Name** field.

iii. Expand **Storage Options** and click **Empty Dir**. An expanded selection of options opens on your Red Hat OpenShift web console.

iv. Set the **Enabled** switch to **true**.

Figure 1.6. Example showing the Empty Dir switch set to true



Storage Options

Options to customize the storage provisioned for the database and object storage.

Empty Dir

Configuration for an EmptyDir to be created by the operator instead of a PVC.

Enabled

true

When enabled, Cryostat will use EmptyDir volumes instead of a Persistent Volume Claim. Any PVC configurations will be ignored.

Medium

Unless specified, the emptyDir volume will be mounted on the same storage medium backing the node. Setting this field to "Memory" will mount the emptyDir on a tmpfs (RAM-backed filesystem).

Size Limit

The maximum memory limit for the emptyDir. Default is unbounded.

PVC

Configuration for the Persistent Volume Claim to be created by the operator.

b. If you want to use the YAML view:

i. Click the **YAML view** radio button.

ii. In the **spec:** key set of the YAML file, add the **storageOptions** definition and set the **emptyDir** property to **true**.

Example showing the emptyDir property set as true

```
--
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
spec:
  storageOptions:
    emptyDir:
      enabled: true
      medium: "Memory"
      sizeLimit: 1Gi
--
```

iii. *Optional:* Set values for the **medium** and **sizeLimit** properties.

3. If you want to configure other properties in the custom resource (CR) for this Cryostat instance, see the other sections of this document for more information about these properties.

4. If you want to finish creating this Cryostat instance, click **Create**.

When you click **Create**, this Cryostat instance is available under the **Cryostat** tab on the **Operator details** page. You can subsequently edit the CR properties for a Cryostat instance by clicking the instance name on the **Operator details** page and then select **Edit Cryostat** from the **Actions** drop-down menu.

The Red Hat build of Cryostat Operator creates an **EmptyDir** volume for storage instead of creating a PVC for your Cryostat instance.

1.6. SCHEDULING OPTIONS FOR CRYOSTAT

From the Red Hat OpenShift web console, you can use the Red Hat build of Cryostat Operator to define policies for scheduling a Cryostat application and its generated reports to nodes.

You can define **Node Selector**, **Affinities**, and **Tolerations** definitions in the YAML configuration file for a Cryostat or Cluster Cryostat custom resource (CR) on Red Hat OpenShift. You must define these definitions under the **spec.SchedulingOptions** property for the Cryostat application and the **spec.ReportOptions.SchedulingOptions** property for the report generator sidecar. By specifying the **SchedulingOptions** property, the Cryostat application and its report generator sidecar pods will be scheduled on nodes that meet the scheduling criteria.

a targeted node application can receive sidecar reports updates from a Cryostat instance.

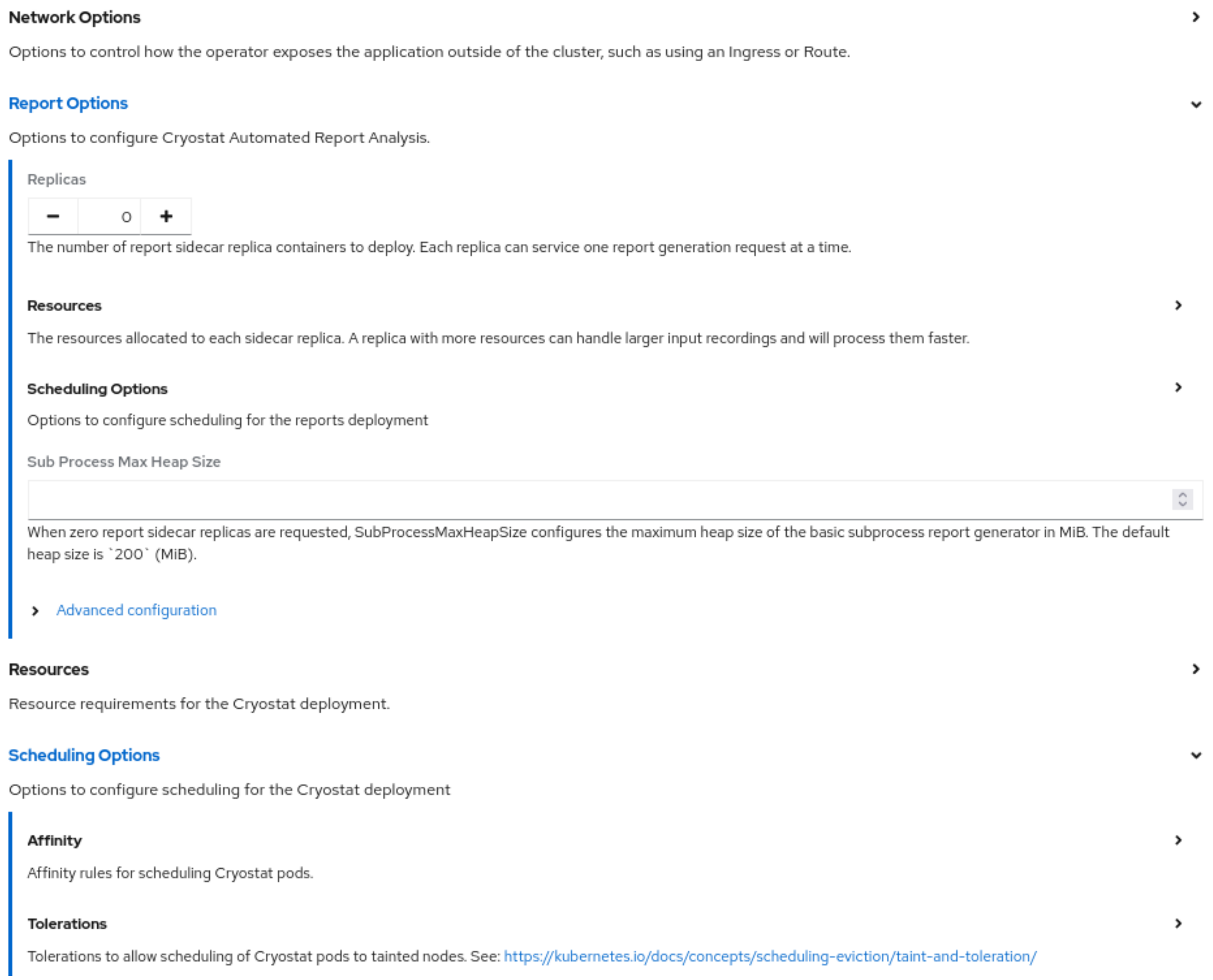
Example that shows the YAML configuration for a Cryostat CR that defines schedule options

```
kind: Cryostat
apiVersion: operator.cryostat.io/v1beta2
metadata:
  name: cryostat
spec:
  schedulingOptions:
    nodeSelector:
      node: good
    affinity:
      nodeAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          nodeSelectorTerms:
          - matchExpressions:
            - key: node
              operator: In
              values:
              - good
              - better
      podAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
        - labelSelector:
            matchLabels:
              pod: good
          topologyKey: topology.kubernetes.io/zone
      podAntiAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
```

```
        - labelSelector:
            matchLabels:
              pod: bad
          topologyKey: topology.kubernetes.io/zone
    tolerations:
    - key: node
      operator: Equal
      value: ok
      effect: NoExecute
  reportOptions:
    replicas: 1
    schedulingOptions:
      nodeSelector:
        node: good
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: node
                operator: In
                values:
                - good
                - better
        podAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchLabels:
                pod: good
            topologyKey: topology.kubernetes.io/zone
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchLabels:
                pod: bad
            topologyKey: topology.kubernetes.io/zone
      tolerations:
      - key: node
        operator: Equal
        value: ok
        effect: NoExecute
```

Alternatively, you can open your Red Hat OpenShift web console, create a Cryostat instance, and then define **Affinities** and **Tolerations** definitions in the **SchedulingOptions** and **reportOptions.SchedulingOptions** options for that Cryostat instance.

Figure 1.7. The Report Options and Scheduling Options panels on the OpenShift web console



Network Options

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

Report Options

Options to configure Cryostat Automated Report Analysis.

Replicas

0

The number of report sidecar replica containers to deploy. Each replica can service one report generation request at a time.

Resources

The resources allocated to each sidecar replica. A replica with more resources can handle larger input recordings and will process them faster.

Scheduling Options

Options to configure scheduling for the reports deployment

Sub Process Max Heap Size

When zero report sidecar replicas are requested, SubProcessMaxHeapSize configures the maximum heap size of the basic subprocess report generator in MiB. The default heap size is `200` (MiB).

Resources

Resource requirements for the Cryostat deployment.

Scheduling Options

Options to configure scheduling for the Cryostat deployment

Affinity

Affinity rules for scheduling Cryostat pods.

Tolerations

Tolerations to allow scheduling of Cryostat pods to tainted nodes. See: https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/

CHAPTER 2. POD SECURITY ADMISSION

Red Hat OpenShift uses Pod Security Admission (PSA) to apply a set of security rules for application pods that are in the same Red Hat OpenShift cluster. In the context of Cryostat, these application pods include a Cryostat pod and a Report sidecar pod. Optionally, you can enable the Report sidecar pod on a Cryostat custom resource (CR). If an application does not meet the policy standards, the application cannot run in your Red Hat OpenShift cluster.

Red Hat OpenShift 4.8 or later no longer supports the **PodSecurityPolicy** API and uses the PSA instead. The PSA provides the following benefits:

- Includes a built-in controller that can enforce pod security standards for your application pods.
- Includes a set of pod security standards that define three different policies: **Privileged**, **Baseline**, and **Restricted**.

On Red Hat OpenShift, you can use the PSA with security context constraints (SCCs) to define policies for an Red Hat OpenShift cluster. By default, the **restricted-v2** SCC aligns with the **Restricted** pod security standard.

NOTE

By default, the security context for a Cryostat pod conforms to the **restricted-v2** SCC, which means that Red Hat OpenShift can admit the pod in namespaces that enforce the **Restricted** pod security standard.

The **Restricted** policy requires that the Red Hat build of Cryostat Operator configures the container security context as follows:

- Drops **ALL** capabilities
- Sets **allowPrivilegeEscaltion** to **false**

The **Restricted** policy requires that the Red Hat build of Cryostat Operator configures the pod security context as follows:

- Sets **runAsNonRoot** to **true**
- Sets the **seccompProfile** to **RuntimeDefault**

Additionally, the Red Hat build of Cryostat Operator defines **fsGroup** in the pod security context for the Cryostat application pod, so that Cryostat can read and write to files in a persistent storage volume on Red Hat OpenShift.

If you have additional requirements beyond conforming to the **Restricted** pod security standard, you can override the default security contexts that Cryostat uses.

2.1. CONFIGURING SECURITY CONTEXTS

You can specify pod and container security contexts in the Cryostat custom resource (CR) on Red Hat OpenShift. The security context applies permissions to the Cryostat pod, the Report sidecar pod (when it is in use), and the containers for each pod.

NOTE

If you change the settings of the CR, these settings override the default security context settings.

A security context applies specific permissions to an application that exists in a pod. The security context cannot change the criteria of an SCC policy. You can create a custom SCC to instruct the Red Hat OpenShift cluster to enforce strict permissions on the pod, such as actions that the pod can perform or resources that the pod can access.

To create a custom SCC you must have cluster administration permissions. You must also create a security context for any pods that operate in the cluster, so that these pods meet the custom SCC requirements.

An SCC enforces changes at the Red Hat OpenShift cluster level and namespace level, so that any pods operating inside this cluster receive policy criteria. By contrast, a security context is unique to a pod.

By default, the Red Hat build of Cryostat Operator conforms to the **restricted-v2** SCC policy for your Cryostat pod.

By default, the Red Hat build of Cryostat Operator creates a service account for Cryostat and its components, such as **jfr-datasource**, **grafana**, **storage**. **database**, and **auth-proxy**.

To enable this service account to use a custom SCC, perform either of the following steps:

- Create a **Role Binding** that binds the Cryostat service account to a role that **uses** your custom SCC.
- Use a **Label Syncer** component to instruct your project's namespace to follow PSA policies.

NOTE

The **Label Syncer** component is outside the scope of this document. You cannot use the **Label Syncer** component on Red Hat OpenShift system namespaces, which are usually prefixed with the **openshift-** tag.

IMPORTANT

Before you configure a security context to apply specific permissions to an application pod, consider the security risks that you might introduce to your cluster on Red Hat OpenShift. The PSA provides three gradient policy levels that typically meet most requirements. Red Hat does not take any responsibility for security context changes that do not align with the Red Hat OpenShift pod security standards.

Prerequisites

- Logged in to the OpenShift Container Platform by using the Red Hat OpenShift web console.
- Installed the Red Hat build of Cryostat Operator in a project on Red Hat OpenShift. See Installing Cryostat on Red Hat OpenShift by using a Red Hat build of Cryostat Operator (Installing Cryostat).
- *Optional:* Read the new PSA and new SCC policies. See Managing security context constraints (OpenShift Container Platform).

- *Optional:* Configured your project to use one of the three polices that the PSA provides.
  - If you want to use a custom SCC to enforce specific policies for your pod, you must configure the SCC to enable your pod’s service account to access it.

Procedure

1. If you want to start creating a Cryostat instance, perform the following steps:
   a. On your Red Hat OpenShift web console, click **Operators > Installed Operators**.
   b. From the list of available Operators, select Red Hat build of Cryostat.
   c. On the **Operator details** page, click the **Details** tab.
   d. In the **Provided APIs** section, select Cryostat, and then click **Create instance**.

   > NOTE
   >
   > The Red Hat build of Cryostat Operator does not create a service account for the Report sidecar pods. Instead, these pods use default service accounts in their own namespaces.

2. On the Create Cryostat panel, to configure a security context, choose one of the following options:
   a. If you want to use the YAML view:
      i. Click the **YAML view** radio button.
      ii. From the **spec:** element, edit the **securityOptions** and **reportOptions** properties to match your security requirements.

Example configuration for a security context

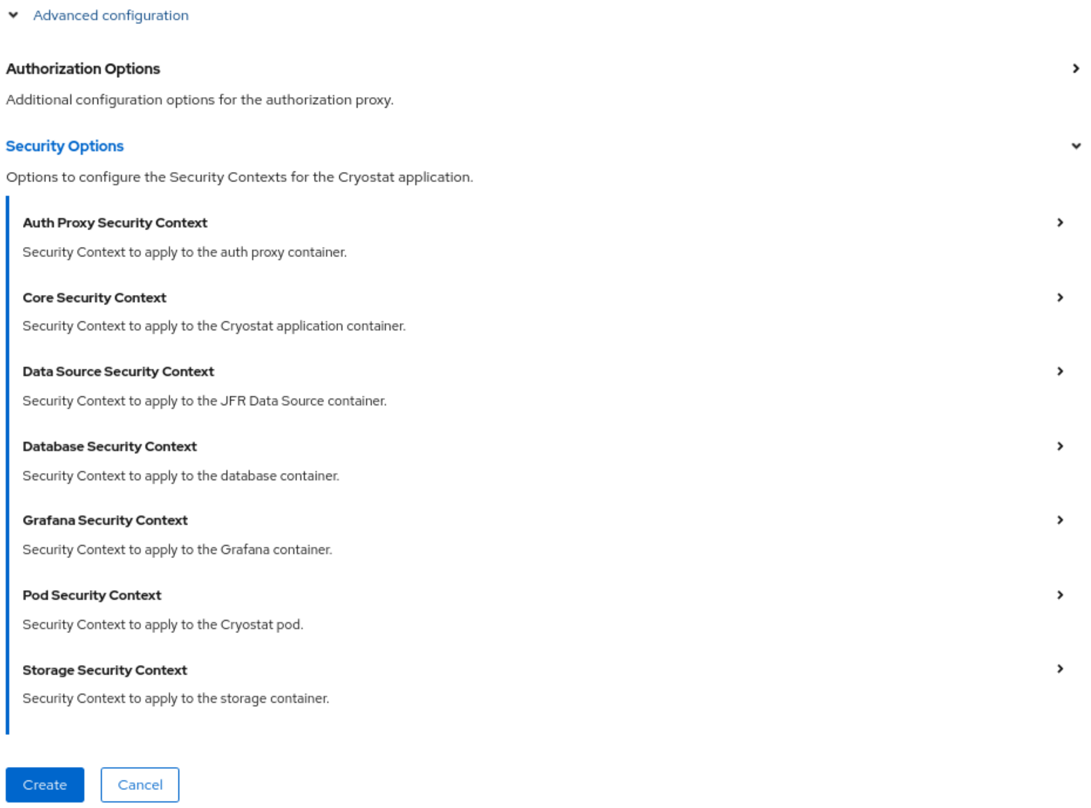
```
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
spec:
  securityOptions:
    podSecurityContext:
      runAsNonRoot: true
      seccompProfile:
        type: RuntimeDefault
    coreSecurityContext:
      allowPrivilegeEscalation: false
      capabilities:
        drop:
          - ALL
      runAsUser: 1001
    dataSourceSecurityContext:
      allowPrivilegeEscalation: false
      capabilities:
        drop:
          - ALL
  grafanaSecurityContext:
    allowPrivilegeEscalation: false
    capabilities:
      drop:
        - ALL
reportOptions:
  replicas: 1
  podSecurityContext:
    runAsNonRoot: true
    seccompProfile:
      type: RuntimeDefault
  reportsSecurityContext:
    allowPrivilegeEscalation: false
    capabilities:
      drop:
        - ALL
    runAsUser: 1001
```

b. If you want to use the Form view:

   i. Click the *Form view" radio button.

   ii. Expand **Advanced Configurations** to open additional options on your Red Hat OpenShift web console.

Figure 2.1. The Advanced configuration menu options



Advanced configuration

Authorization Options

Additional configuration options for the authorization proxy.

Security Options

Options to configure the Security Contexts for the Cryostat application.

Auth Proxy Security Context

Security Context to apply to the auth proxy container.

Core Security Context

Security Context to apply to the Cryostat application container.

Data Source Security Context

Security Context to apply to the JFR Data Source container.

Database Security Context

Security Context to apply to the database container.

Grafana Security Context

Security Context to apply to the Grafana container.

Pod Security Context

Security Context to apply to the Cryostat pod.

Storage Security Context

Security Context to apply to the storage container.

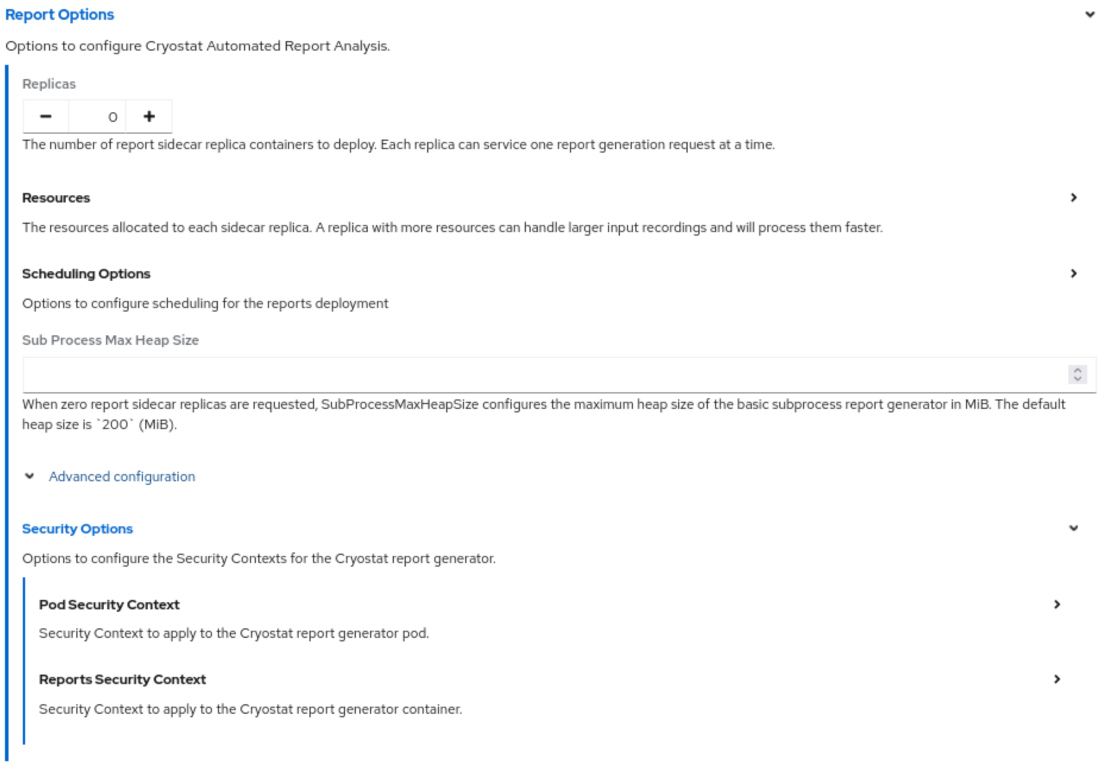
   iii. Expand **Core Security Context**. From the available list of options, define settings for your security context.

   iv. Expand each of the following security contexts in turn and define settings, as appropriate:

- Auth Proxy Security Context
- Data Source Security Context
- Database Security Context
- Grafana Security Context
- Pod Security Context
- Storage Security Context

v. *Optional:* If you are using the **Report Generator** service, you can also configure the security contexts for this service. In this situation, expand **Report Options > Advanced Configuration > Security Options**. Then expand and define **Reports Security Context** and **Pod Security Context** settings, as appropriate.

Figure 2.2. Report Generator security contexts



Report Options

Options to configure Cryostat Automated Report Analysis.

Replicas

− 0 +

The number of report sidecar replica containers to deploy. Each replica can service one report generation request at a time.

Resources

The resources allocated to each sidecar replica. A replica with more resources can handle larger input recordings and will process them faster.

Scheduling Options

Options to configure scheduling for the reports deployment

Sub Process Max Heap Size

When zero report sidecar replicas are requested, SubProcessMaxHeapSize configures the maximum heap size of the basic subprocess report generator in MiB. The default heap size is `200` (MiB).

Advanced configuration

Security Options

Options to configure the Security Contexts for the Cryostat report generator.

Pod Security Context

Security Context to apply to the Cryostat report generator pod.

Reports Security Context

Security Context to apply to the Cryostat report generator container.

3. If you want to configure other properties in the custom resource (CR) for this Cryostat instance, see the other sections of this document for more information about these properties.

4. If you want to finish creating this Cryostat instance, click **Create**.

When you click **Create**, this Cryostat instance is available under the **Cryostat** tab on the **Operator details** page. You can subsequently edit the CR properties for a Cryostat instance by clicking the instance name on the **Operator details** page and then select **Edit Cryostat** from the **Actions** drop-down menu.

Additional resources

- Pod Security Standard policies.

2.2. POD SECURITY STANDARD POLICIES

The Pod Security Admission (PSA) includes three policies that cover security levels related to pod security standards. The following table explains each policy:

Profile	Description
Privileged	An unrestricted policy that provides a wide level of permissions for your Cryostat pod. Consider setting this policy if you need to provide known privilege escalations to your pods.
Baseline	Default policy that restricts known privileged escalations. The **Baseline** policy sets controls where each control defines restricted fields and allowed values.
Restricted	The **Restricted** policy that provides a low level of permissions for your Cryostat pod. This policy sets controls with each control defining restricted fields and allowed values.

CHAPTER 3. CONFIGURING RBAC SETTINGS

When you install Cryostat 3.0 by using either the Cryostat Operator or a Helm chart, Cryostat includes a reverse proxy (**openshift-oauth-proxy** or **oauth2_proxy**) in the pod. All API requests to Cryostat and all users of the Cryostat web console or Grafana dashboard are directed through this proxy, which handles client sessions to control access to the application. When deployed on Red Hat OpenShift, the proxy uses the Cryostat installation namespace to perform RBAC checks for user authentication and authorization by integrating with the Red Hat OpenShift cluster SSO provider.

From Cryostat 3.0 onward, Cryostat 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 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.

NOTE

You can optionally configure the auth proxy with an **htpasswd** file to enable Basic authentication. On Red Hat OpenShift, this enables you to define additional user accounts that can access Cryostat beyond those with Red Hat OpenShift SSO RBAC access.

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 Red Hat OpenShift SSO RBAC permissions for accessing Cryostat.

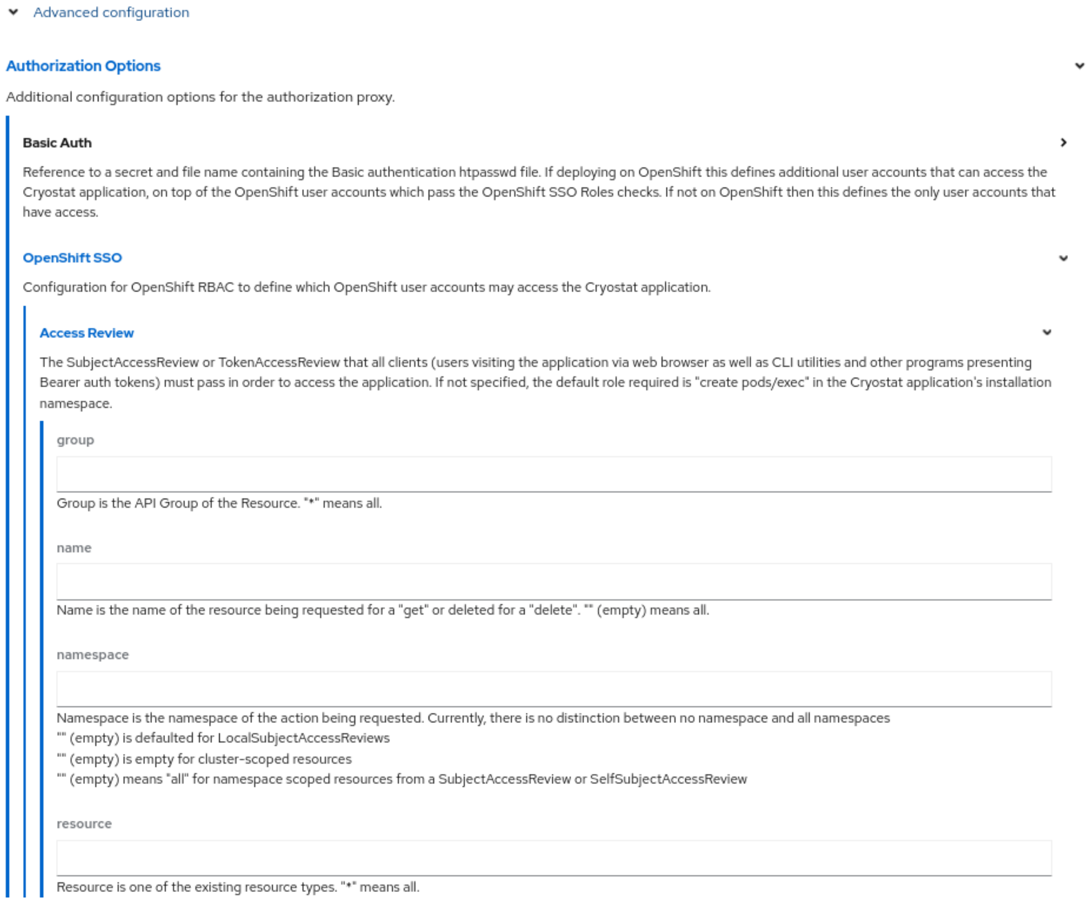
Prerequisites

- Logged in to the OpenShift Container Platform by using the Red Hat OpenShift web console.

Procedure

1. If you want to start creating a Cryostat instance, perform the following steps:
    a. On your Red Hat OpenShift web console, click **Operators > Installed Operators**.
    b. From the list of available Operators, select Red Hat build of Cryostat.
    c. On the **Operator details** page, click the **Details** tab.
    d. In the **Provided APIs** section, select Cryostat and then click **Create instance**.
2. On the Create Cryostat panel, to customize the required SubjectAccessReview or TokenAccessReview for all client access to Cryostat, choose one of the following options:
    a. If you are using the Form view:
        i. Click the **Form view** radio button.
        ii. To open additional options, expand **Advanced Configurations** to open additional options.
        iii. Expand the **Authorization Options > OpenShift SSO > Access Review**section of the Cryostat CR.

Figure 3.1. Access Review properties for a Cryostat instance



Advanced configuration

Authorization Options

Additional configuration options for the authorization proxy.

Basic Auth

Reference to a secret and file name containing the Basic authentication htpasswd file. If deploying on OpenShift this defines additional user accounts that can access the Cryostat application, on top of the OpenShift user accounts which pass the OpenShift SSO Roles checks. If not on OpenShift then this defines the only user accounts that have access.

OpenShift SSO

Configuration for OpenShift RBAC to define which OpenShift user accounts may access the Cryostat application.

Access Review

The SubjectAccessReview or TokenAccessReview that all clients (users visiting the application via web browser as well as CLI utilities and other programs presenting Bearer auth tokens) must pass in order to access the application. If not specified, the default role required is "create pods/exec" in the Cryostat application's installation namespace.

group

Group is the API Group of the Resource. "*" means all.

name

Name is the name of the resource being requested for a "get" or deleted for a "delete". "" (empty) means all.

namespace

Namespace is the namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces
"" (empty) is defaulted for LocalSubjectAccessReviews
"" (empty) is empty for cluster-scoped resources
"" (empty) means "all" for namespace scoped resources from a SubjectAccessReview or SelfSubjectAccessReview

resource

Resource is one of the existing resource types. "*" means all.

iv. Use the following fields to specify any customized RBAC settings that are required for accessing Cryostat:

Field	Details
group	API group of the resource. A wilcard asterisk (*) value represents all groups.
name	Name of the resource being requested for a **get** or deleted for a **delete**. An empty value represents all names.
namespace	Namespace of the action being requested. Currently, there is no distinction between no namespace and all namespaces. Consider the following guidelines: ● An empty value is defaulted for LocalSubjectAccessReviews. ● An empty value represents no cluster-scoped resources. ● An empty value represents all namespace-scoped resources from a SubjectAccessReview or SelfSubjectAccessReview.

Field	Details
resource	An existing resource type. A wildcard asterisk (*****) value represents all resource types.
subresource	An existing resource type. An empty value represents no resource types.
verb	A Kubernetes resource API verb (for example, **get**, **list**, **watch**, **create**, **update**, **delete**, **proxy**). A wildcard asterisk (*****) value represents all verbs.
version	API version of the resource. A wildcard asterisk (*****) value represents all versions.

b. If you are using the YAML view:

i. Click the **YAML view** radio button.

ii. From the **spec:** element, edit the **authorizationOptions:OpenShiftSSO** properties to match your RBAC permission requirements.

Example configuration for RBAC permissions

```
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
  namespace: cryostat-test
spec:
  ...
  authorizationOptions:
    openShiftSSO:
      accessReview:
        group: <API group of resource>
        name: <Name of resource being requested or deleted>
        namespace: <Namespace of action being requested>
        resource: <An existing resource type>
        subresource: <An existig resource type>
        verb: <A Kubernetes resource API verb>
        version: <API version of resource>
  ...
```

3. If you want to configure other properties in the custom resource (CR) for this Cryostat instance, see the other sections of this document for more information about these properties.

4. If you want to finish creating this Cryostat instance, click **Create**.

When you click **Create**, this Cryostat instance is available under the **Cryostat** tab on the **Operator details** page. You can subsequently edit the CR properties for a Cryostat instance by clicking the instance name on the **Operator details** page and then select **Edit Cryostat** from the **Actions** drop-down menu.

Revised on 2024-07-02 13:37:27 UTC