This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Setting up Genesys Multicloud CX Private Edition

OpenShift security settings

2/6/2026

Contents

Learn how OpenShift uses security context constraints (SCCs) to control pod permissions and how you can use arbitrary user IDs (UIDs) to enhance security against permissions escalation in the OpenShift environment. For general information on private edition security, see the security overview.

Related documentation:

-
-

RSS:

- For private edition

OpenShift uses *security context constraints* (SCCs) to control pod permissions. The goal is to provide an appropriate set of permissions that are limited—as much as possible, anyway—to a select group of users responsible for deploying and managing the private edition software. This is essentially a form of role-based access control (RBAC).

OpenShift comes with eight predefined SCCs with varying levels of access control. By default, all pods and containers in the private edition environment use the **restricted** SCC, which is the most restrictive of these predefined SCCs.

Arbitrary UIDs

OpenShift uses arbitrary, or randomly assigned, user IDs (UIDs) to increase access security. This means that the IDs of the users accessing the pods and containers and running the application processes are unspecified and unpredictable.

By default, the securityContext settings exposed in the **values.yaml** files of the respective services specify the user and group IDs for the **genesys** user (500:500:500). You must modify the Helm charts to enable use of arbitrary UIDs.

Modify Helm charts

Override the **securityContext** settings in **values.yaml** so that no specific IDs are defined for users and the group (the group defaults to root). You must do this for all of the levels at which the **securityContext** is specified in the Helm chart, for example:

```
podSecurityContext:
  fsGroup: null
  runAsUser: null
  runAsGroup: 0
  runAsNonRoot: true

securityContext:
  fsGroup: null
```

```
runAsUser: null
runAsGroup: 0
runAsNonRoot: true
```

Check the ServiceAccount:

- If your service uses the **default** ServiceAccount, which uses the **restricted** SCC, no further action is required, as this SCC does not enforce a UID/GID range.
- If your service uses a custom ServiceAccount, you must verify that it has not been associated with an SCC that enforces a UID/GID range, such as the **genesys-restricted** SCC, which has been deprecated.

  For example, to identify if any services are set to use the **genesys-restricted** SCC, execute the following command:

  ```
  oc adm policy who-can use scc genesys-restricted -n
  ```

  If any services have been deployed with the **genesys-restricted** SCC, Genesys recommends that you redeploy the service so that you use arbitrary UIDs.