**Security Context Constraints (SCCs)**

Red Hat OpenShift provides *security context constraints (SCCs)*, a security mechanism that restricts access to resources, but not to operations in OpenShift.

SCCs limit the access from a running pod in OpenShift to the host environment. SCCs control:

* + Running privileged containers.
  + Requesting extra capabilities for a container
  + Using host directories as volumes.
  + Changing the SELinux context of a container.
  + Changing the user ID.

Some containers developed by the community might require relaxed security context constraints to access resources that are forbidden by default, such as file systems, sockets, or to access a SELinux context.

You can run the following command as a cluster administrator to list the SCCs defined by OpenShift:

**oc get scc**

**OpenShift provides eight SCCs:**

**• anyuid**

**• hostaccess**

**• hostmount-anyuid**

**• hostnetwork**

**• node-exporter**

**• nonroot**

**• privileged**

**• restricted**

**Most pods created by OpenShift use the SCC named restricted, which provides limited access to resources external to OpenShift.**

Container images downloaded from public container registries, such as Docker Hub, might fail to run using the **restricted** SCC.

For example, a container image that requires running as a specific user ID can fail because the **restricted** SCC runs the container using a random user ID.

A container image that listens on port 80 or port 443 can fail for a related reason. The random user ID used by the **restricted** SCC cannot start a service that listens on a privileged network port (port numbers less than 1024). Use the **scc-subject-review** subcommand to list all the security context constraints that can overcome the limitations of a container

**oc get pod *podname* -o yaml | oc adm policy scc-subject-review -f –**

For the **anyuid** SCC,

the **run as user** strategy is defined as **RunAsAny**, which means that the pod can run as any user ID available in the container. This strategy allows containers that require a specific user to run the commands using a specific user ID.

**To change the container to run using a different SCC**,

you must create a service account bound to a pod. Use the **oc create serviceaccount** command to create the service account, and use the **-n** option if the service account must be created in a namespace different than the current one.

**oc create serviceaccount *service-account-name***

To associate the service account with an SCC, use the **oc adm policy** command. Use the **-z** option to identify a service account, and use the **-n** option if the service account exists in a namespace different than the current one.

**oc adm policy add-scc-to-user *SCC* -z *service-account***

**Important**

Assigning an SCC to a service account or removing an SCC from a service account

must be performed by a cluster administrator. Allowing pods to run with a less

restrictive SCC can make your cluster less secure. Use with caution.

**Privileged Containers**

Some containers might need to access the runtime environment of the host. For example, the S2I builder containers are a class of privileged containers that require access beyond the limits of their own containers. These containers can pose security risks because they can use any resources on an OpenShift node. Use SCCs to enable access for privileged containers by creating service accounts with privileged access.