Red Hat OpenShift Service on AWS 4

クラスターセキュリティー

Red Hat OpenShift Service on AWS クラスターのセキュリティー保護

Last Updated: 2022-11-21

Red Hat OpenShift Service on AWS 4 クラスターセキュリティー

Red Hat OpenShift Service on AWS クラスターのセキュリティー保護

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本書では、Red Hat OpenShift Service on AWS (ROSA) クラスターのセキュリティー保護について説明します。

目次

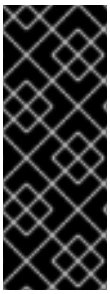
第1章 SSC (SECURITY CONTEXT CONSTRAINTS) の管理

1.1. SCC (SECURITY CONTEXT CONSTRAINTS) について

RBAC リソースがユーザーアクセスを制御するのと同じ方法で、管理者は **SCC (Security Context Constraints)** を使用して Pod のパーミッションを制御できます。これらのパーミッションには、Pod が実行できるアクションおよび Pod がアクセスできるリソースが含まれます。SCC を使用して、Pod がシステムに受け入れられるために必要な Pod の実行に関する条件の一覧を定義することができます。

管理者は SCC (Security Context Constraints) で、以下を制御できます。

- Pod が **allowPrivilegedContainer** フラグが付いた特権付きコンテナーを実行できるかどうか
- Pod が **allowPrivilegeEscalation** フラグで制約されているかどうか
- コンテナーが要求できる機能
- ホストディレクトリーのボリュームとしての使用
- コンテナーの SELinux コンテキスト
- コンテナーのユーザー ID
- ホストの namespace とネットワークの使用
- Pod ボリュームを所有する **FSGroup** の割り当て
- 許可される補助グループの設定
- コンテナーが root ファイルシステムへの書き込みアクセスを必要とするかどうか
- ボリュームタイプの使用
- 許可される **seccomp** プロファイルの設定



重要

Red Hat OpenShift Service on AWS の namespace には **openshift.io/run-level** ラベルを設定しないでください。このラベルは、Kubernetes API サーバーや OpenShift API サーバーなどのメジャー API グループの起動管理に、内部の Red Hat OpenShift Service on AWS コンポーネントで使用されます。**openshift.io/run-level** ラベルが設定される場合には、対象の namespace の Pod に SCC が適用されず、その namespace で実行されるワークロードには高度な特権が割り当てられます。

1.1.1. デフォルトの SCC (Security Context Constraints)

クラスターには、以下の表で説明されているように、デフォルトの SCC (Security Context Constraints) が複数含まれます。追加の SCC は、Operator または他のコンポーネントの Red Hat OpenShift Service on AWS へのインストール時にインストールされる可能性があります。

重要

デフォルトの SCC は変更しないでください。デフォルトの SCC をカスタマイズすると、プラットフォーム Pod のデプロイまたは Red Hat OpenShift Service on AWS のアップグレード時に問題が発生する可能性があります。Red Hat OpenShift Service on AWS の一部のバージョン間のアップグレード時に、デフォルトの SCC 値はデフォルト値にリセットされ、これらの SCC に対するカスタマイズはすべて破棄されます。

表1.1 デフォルトの SCC (Security Context Constraints)

SCC (Security Context Constraints)	説明
anyuid	SCC のすべての機能が **restricted** で提供されますが、ユーザーは任意の UID と GID で実行できます。
hostaccess	ホストの全 namespace にアクセスできますが、対象の namespace に割り当てられた UID および SELinux コンテキストで Pod を実行する必要があります。 **警告** この SCC で、ホストは namespace、ファイルシステム、および PID にアクセスできます。信頼できる Pod だけがこの SCC を使用する必要があります。付与には注意が必要です。
hostmount-anyuid	SCC のすべての機能を **restricted** で提供しますが、ホストのマウントとシステム上の任意の UID および GID として実行できます。 **警告** この SCC は、UID 0 を含む任意の UID としてホストファイルシステムにアクセスできます。付与には注意が必要です。

SCC (Security Context Constraints)	説明
hostnetwork	ホストのネットワークおよびホストポートを使用できますが、対象の namespace に割り当てられた UID および SELinux コンテキストで Pod を実行する必要があります。 **警告** 追加のワークロードをコントロールプレーンホストで実行する場合は、**hostnetwork** へのアクセスを割り当てるときに注意してください。コントロールプレーンホストで **hostnetwork** を実行するワークロードにはクラスター上で root アクセスを持つユーザーと同じ機能があるため、適切に信頼されている必要があります。
hostnetwork-v2	**ホストネットワーク** SCC と似ていますが、次の違いがあります。 • **ALL** 機能がコンテナーから削除されます。 • **NET_BIND_SERVICE** 機能を明示的に追加できます。 • **seccompProfile** はデフォルトで **runtime/default** に設定されています。 • セキュリティーコンテキストでは、**allowPrivilegeEscalation** を設定解除するか、**false** に設定する必要があります。
node-exporter	Prometheus ノードエクスポーターに使用されます。 **警告** この SCC は、UID 0 を含む任意の UID としてホストファイルシステムにアクセスできます。付与には注意が必要です。
nonroot	SCC のすべての機能が **restricted** で提供されますが、ユーザーは root 以外の UID で実行できます。ユーザーは UID を指定するか、コンテナーランタイムのマニフェストに指定する必要があります。

SCC (Security Context Constraints)	説明
nonroot-v2	**nonroot** SCC と同様ですが、次の違いがあります。 ● **ALL** 機能がコンテナーから削除されます。 ● **NET_BIND_SERVICE** 機能を明示的に追加できます。 ● **seccompProfile** はデフォルトで **runtime/default** に設定されています。 ● セキュリティーコンテキストでは、**allowPrivilegeEscalation** を設定解除するか、**false** に設定する必要があります。

SCC (Security Context Constraints)	説明
privileged	すべての特権およびホスト機能にアクセスでき、任意のユーザー、任意のグループ、FSGroup、および任意の SELinux コンテキストで実行できます。 **警告** これは最も制限の少ない SCC であり、クラスター管理にのみ使用してください。付与には注意が必要です。 **privileged** SCC は以下を許可します。 ● ユーザーによる特権付き Pod の実行 ● Pod によるホストディレクトリーのボリュームとしてのマウント ● Pod の任意ユーザーとしての実行 ● Pod の MCS ラベルの使用による実行 ● Pods によるホストの IPC namespace の使用 ● Pod によるホストの PID namespace の使用 ● Pod による FSGroup の使用 ● Pod による補助グループの使用 ● Pod による seccomp プロファイルの使用 ● Pod による機能の要求 **注記** Pod の仕様で **privileged: true** を設定しても、**privileged** SCC が選択されるとは限りません。ユーザーに使用権限がある場合に、**allowPrivilegedContainer: true** が指定されており、優先順位が最も高い SCC が選択されます。

SCC (Security Context Constraints)	説明
restricted	すべてのホスト機能へのアクセスが拒否され、Pod を UID および namespace に割り当てられる SELinux コンテキストで実行する必要があります。 **restricted** SCC は以下を実行します。 ● Pod が特権付きで実行されないようにします。 ● Pod がホストディレクトリーボリュームをマウントできないようにします。 ● Pod が事前に割り当てられた UID 範囲のユーザーとして実行されることを要求します。 ● Pod が事前に割り当てられた MCS ラベルで実行されることを要求します。 ● Pod が FSGroup を使用することを許可します。 ● Pod が補助グループを使用することを許可します。 Red Hat OpenShift Service on AWS 4.10 以前からアップグレードされたクラスターでは、すべての認証済みユーザーがこの SCC を使用できます。アクセスが明示的に付与されない限り、新しい Red Hat OpenShift Service on AWS 4.11 インストールのユーザーは **restricted** SCC を使用できなくなります。
restricted-v2	**restricted** SCC と同様ですが、次の違いがあります。 ● **ALL** 機能がコンテナーから削除されます。 ● **NET_BIND_SERVICE** 機能を明示的に追加できます。 ● **seccompProfile** はデフォルトで **runtime/default** に設定されています。 ● セキュリティーコンテキストでは、**allowPrivilegeEscalation** を設定解除するか、**false** に設定する必要があります。 これは新規インストールで提供され、デフォルトで認証済みユーザーに使用される最も制限の厳しい SCC です。  **注記** **restricted-v2** SCC は、システムにデフォルトで含まれている SCC の中で最も制限が厳しいものです。ただし、さらに制限の厳しいカスタム SCC を作成できます。たとえば、**readOnlyRootFilesystem** を **true** に制限する SCC を作成できます。

1.1.2. SCC (Security Context Constraints) の設定

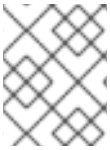
SCC (Security Context Constraints) は、Pod がアクセスできるセキュリティー機能を制御する設定およびストラテジーで設定されています。これらの設定は以下のカテゴリーに分類されます。

カテゴリー	説明
ブール値による制御	このタイプのフィールドはデフォルトで最も制限のある値に設定されます。たとえば、**AllowPrivilegedContainer** が指定されていない場合は、**false** に常に設定されます。
許可されるセットによる制御	このタイプのフィールドがセットに対してチェックされ、その値が許可されることを確認します。
ストラテジーによる制御	値を生成するストラテジーを持つ項目は以下を提供します。 ● 値を生成するメカニズム ● 指定された値が許可される値のセットに属するようにするメカニズム

CRI-O には、Pod の各コンテナーについて許可されるデフォルトの機能一覧があります。

- **CHOWN**
- **DAC_OVERRIDE**
- **FSETID**
- **FOWNER**
- **SETGID**
- **SETUID**
- **SETPCAP**
- **NET_BIND_SERVICE**
- **KILL**

コンテナーはこのデフォルト一覧から機能を使用しますが、Pod マニフェストの作成者は追加機能を要求したり、デフォルト動作の一部を削除して一覧を変更できます。**allowedCapabilities**、**defaultAddCapabilities**、および **requiredDropCapabilities** パラメーターを使用して、Pod からのこのような要求を制御します。これらのパラメーターを使用して、(各コンテナーに追加する必要のある機能や、各コンテナーから禁止または破棄する必要のあるものなど) 要求できる機能を指定できます。



注記

requiredDropCapabilities パラメーターを **ALL** に設定すると、すべての capabilites をコンテナーから取り除くことができます。これは、**restricted-v2** SCC の機能です。

1.1.3. SCC (Security Context Constraints) ストラテジー

RunAsUser

- **MustRunAs**: **runAsUser** が設定されることを要求します。デフォルトで設定済みの **runAsUser** を使用します。設定済みの **runAsUser** に対して検証します。

- **MustRunAsRange**: 事前に割り当てられた値を使用していない場合に、最小および最大値が定義されることを要求します。デフォルトでは最小値を使用します。許可される範囲全体に対して検証します。
- **MustRunAsNonRoot**: Pod がゼロ以外の **runAsUser** で送信されること、または **USER** ディレクティブをイメージに定義することを要求します。デフォルトは指定されません。
- **RunAsAny**: デフォルトは指定されません。**runAsUser** の指定を許可します。

SELinuxContext

- **MustRunAs**: 事前に割り当てられた値を使用していない場合に **seLinuxOptions** が設定されることを要求します。デフォルトとして **seLinuxOptions** を使用します。**seLinuxOptions** に対して検証します。
- **RunAsAny**: デフォルトは指定されません。**seLinuxOptions** の指定を許可します。

SupplementalGroups

- **MustRunAs**: 事前に割り当てられた値を使用していない場合に、少なくとも 1 つの範囲が指定されることを要求します。デフォルトとして最初の範囲の最小値を使用します。すべての範囲に対して検証します。
- **RunAsAny**: デフォルトは指定されません。**supplementalGroups** の指定を許可します。

FSGroup

- **MustRunAs**: 事前に割り当てられた値を使用していない場合に、少なくとも 1 つの範囲が指定されることを要求します。デフォルトとして最初の範囲の最小値を使用します。最初の範囲の最初の ID に対して検証します。
- **RunAsAny**: デフォルトは指定されません。**fsGroup** ID の指定を許可します。

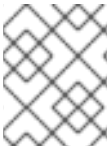
1.1.4. ボリュームの制御

特定のボリュームタイプの使用は、SCC の **volumes** フィールドを設定して制御できます。このフィールドの許容値は、ボリュームの作成時に定義されるボリュームソースに対応します。

- **awsElasticBlockStore**
- **azureDisk**
- **azureFile**
- **cephFS**
- **cinder**
- **configMap**
- **downwardAPI**
- **emptyDir**
- **fc**

- flexVolume
- flocker
- gcePersistentDisk
- gitRepo
- glusterfs
- hostPath
- iscsi
- nfs
- persistentVolumeClaim
- photonPersistentDisk
- portworxVolume
- projected
- quobyte
- rbd
- scaleIO
- secret
- storageos
- vsphereVolume
- * (すべてのボリュームタイプの使用を許可する特殊な値)
- **none** (すべてのボリュームタイプの使用を無効にする特殊な値。後方互換の場合にのみ存在する)

新規 SCC について許可されるボリュームの推奨される最小セットは、**configMap**、**downwardAPI**、**emptyDir**、**persistentVolumeClaim**、**secret**、および **projected** です。



注記

Red Hat OpenShift Service on AWS の各リリースに新しいタイプ追加されるため、この許可されるボリュームタイプ一覧がすべて網羅しているわけではありません。

注記

後方互換性を確保するため、**allowHostDirVolumePlugin** の使用は **volumes** フィールドの設定をオーバーライドします。たとえば、**allowHostDirVolumePlugin** が false に設定されていて、**volumes** フィールドで許可されている場合は、**volumes** から **hostPath** 値が削除されます。

1.1.5. 受付制御

SCC が設定された **受付制御** により、ユーザーに付与された機能に基づいてリソースの作成に対する制御が可能になります。

SCC の観点では、これは受付コントローラーが、SCC の適切なセットを取得するためにコンテキストで利用可能なユーザー情報を検査できることを意味します。これにより、Pod はその運用環境についての要求を行ったり、Pod に適用する一連の制約を生成したりする権限が与えられます

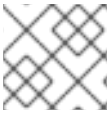
受付が Pod を許可するために使用する SCC のセットはユーザーアイデンティティーおよびユーザーが属するグループによって決定されます。さらに、Pod がサービスアカウントを指定する場合は、許可される SCC のセットに、サービスアカウントでアクセスできる制約が含まれます。

受付は以下の方法を使用して、Pod の最終的なセキュリティーコンテキストを作成します。

1. 使用できるすべての SCC を取得します。
2. 要求に指定されていないセキュリティーコンテキストに、設定のフィールド値を生成します。
3. 利用可能な制約に対する最終的な設定を検証します。

制約の一致するセットが検出される場合は、Pod が受け入れられます。要求が SCC に一致しない場合は、Pod が拒否されます。

Pod はすべてのフィールドを SCC に対して検証する必要があります。以下は、検証する必要のある 2 つのフィールドのみについての例になります。



注記

これらの例は、事前に割り当てられた値を使用するストラテジーに関連します。

MustRunAs の FSGroup SCC ストラテジー

Pod が **fsGroup** ID を定義する場合、その ID はデフォルトの **fsGroup** ID に等しくなければなりません。そうでない場合は、Pod が SCC で検証されず、次の SCC が評価されます。

SecurityContextConstraints.fsGroup フィールドに値 **RunAsAny** があり、Pod 仕様が **Pod.spec.securityContext.fsGroup** を省略すると、このフィールドは有効とみなされます。検証時に、他の SCC 設定が他の Pod フィールドを拒否し、そのため Pod を失敗させる可能性があることに注意してください。

MustRunAs の SupplementalGroups SCC ストラテジー

Pod 仕様が 1 つ以上の **supplementalGroups** ID を定義する場合、Pod の ID は namespace の **openshift.io/sa.scc.supplemental-groups** アノテーションの ID のいずれかに等しくなければなりません。そうでない場合は、Pod が SCC で検証されず、次の SCC が評価されます。

SecurityContextConstraints.supplementalGroups フィールドに値 **RunAsAny** があり、Pod 仕様が **Pod.spec.securityContext.supplementalGroups** を省略する場合、このフィールドは有効とみなされます。検証時に、他の SCC 設定が他の Pod フィールドを拒否し、そのため Pod を失敗させる可能性があることに注意してください。

1.1.6. SCC (Security Context Constraints) の優先度設定

SCC(Security Context Constraints) には優先度フィールドがあり、受付コントローラーの要求検証を試行する順序に影響を与えます。優先度の高い SCC は並び替える際にセットの先頭に移動します。利用可能な SCC の完全なセットが判別されると、それらは以下に戻づいて順序付けられます。

1. 優先度が高い順。 nil は優先度 0 とみなされます。
2. 優先度が等しい場合、SCC は最も制限の多いものから少ないものの順に並べ替えられます。
3. 優先度と制限のどちらも等しい場合、SCC は名前順に並べ替えられます。

デフォルトで、クラスター管理者に付与される **anyuid** SCC には SCC セットの優先度が指定されます。これにより、クラスター管理者は Pod の **SecurityContext** で **RunAsUser** を指定しなくても Pod を任意のユーザーとして実行できます。管理者は、希望する場合は依然として **RunAsUser** を指定できます。

1.2. 事前に割り当てられる SCC (SECURITY CONTEXT CONSTRAINTS) 値について

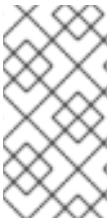
受付コントローラーは、これが namespace の事前に割り当てられた値を検索し、Pod の処理前に SCC (Security Context Constraints) を設定するようにトリガーする SCC (Security Context Constraint) の特定の条件を認識します。各 SCC ストラテジーは他のストラテジーとは別に評価されます。この際、(許可される場合に) Pod 仕様の値と共に集計された各ポリシーの事前に割り当てられた値が使用され、実行中の Pod で定義される各種 ID の最終の値が設定されます。

以下の SCC により、受付コントローラーは、範囲が Pod 仕様で定義されていない場合に事前に定義された値を検索できます。

1. 最小または最大値が設定されていない **MustRunAsRange** の **RunAsUser** ストラテジーです。受付は **openshift.io/sa.scc.uid-range** アノテーションを検索して範囲フィールドを設定します。
2. レベルが設定されていない **MustRunAs** の **SELinuxContext** ストラテジーです。受付は **openshift.io/sa.scc.mcs** アノテーションを検索してレベルを設定します。
3. **MustRunAs** の **FSGroup** ストラテジーです。受付は、**openshift.io/sa.scc.supplemental-groups** アノテーションを検索します。
4. **MustRunAs** の **SupplementalGroups** ストラテジーです。受付は、**openshift.io/sa.scc.supplemental-groups** アノテーションを検索します。

生成フェーズでは、セキュリティーコンテキストのプロバイダーが Pod にとくに設定されていないパラメーター値をデフォルト設定します。デフォルト設定は選択されるストラテジーに基づいて行われます。

1. **RunAsAny** および **MustRunAsNonRoot** ストラテジーはデフォルトの値を提供しません。Pod がパラメーター値 (グループ ID など) を必要とする場合は、値を Pod 仕様内に定義する必要があります。
2. **MustRunAs** (単一の値) ストラテジーは、常に使用されるデフォルト値を提供します。たとえば、グループ ID の場合、Pod 仕様が独自の ID 値を定義する場合でも、namespace のデフォルトパラメーター値が Pod のグループに表示されます。
3. **MustRunAsRange** および **MustRunAs** (範囲ベース) ストラテジーは、範囲の最小値を提供します。単一値の **MustRunAs** ストラテジーの場合のように、namespace のデフォルト値は実行中の Pod に表示されます。範囲ベースのストラテジーが複数の範囲で設定可能な場合、これは最初に設定された範囲の最小値を指定します。

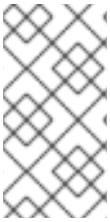


注記

FSGroup および **SupplementalGroups** ストラテジーは、**openshift.io/sa.scc.supplemental-groups** アノテーションが namespace に存在しない場合に **openshift.io/sa.scc.uid-range** アノテーションにフォールバックします。いずれも存在しない場合は、SCC が作成されません。

注記

デフォルトで、アノテーションベースの **FSGroup** ストラテジーは、自身をアノテーションの最小値に基づく単一の範囲で設定します。たとえば、アノテーションが **1/3** を読み取ると、**FSGroup** ストラテジーは **1** の最小値および最大値で自身を設定します。追加のグループを **FSGroup** フィールドで許可する必要がある場合は、アノテーションを使用しないカスタム SCC を設定することができます。



注記

openshift.io/sa.scc.supplemental-groups アノテーションは、**<start>/<length** または **<start>-<end>** 形式のコンマ区切りのブロックの一覧を受け入れます。**openshift.io/sa.scc.uid-range** アノテーションは単一ブロックのみを受け入れます。

1.3. SCC (SECURITY CONTEXT CONSTRAINTS) の例

以下の例は、SCC (Security Context Constraints) 形式およびアノテーションを示しています。

注釈付き privileged SCC

```
allowHostDirVolumePlugin: true
allowHostIPC: true
allowHostNetwork: true
allowHostPID: true
allowHostPorts: true
allowPrivilegedContainer: true
allowedCapabilities: 1
- '*'
apiVersion: security.openshift.io/v1
defaultAddCapabilities: [] 2
fsGroup: 3
  type: RunAsAny
groups: 4
- system:cluster-admins
- system:nodes
kind: SecurityContextConstraints
metadata:
  annotations:
    kubernetes.io/description: 'privileged allows access to all privileged and host
      features and the ability to run as any user, any group, any fsGroup, and with
      any SELinux context.  WARNING: this is the most relaxed SCC and should be used
      only for cluster administration. Grant with caution.'
  creationTimestamp: null
  name: privileged
priority: null
readOnlyRootFilesystem: false
```

```
requiredDropCapabilities: 5
- KILL
- MKNOD
- SETUID
- SETGID
runAsUser: 6
  type: RunAsAny
seLinuxContext: 7
  type: RunAsAny
seccompProfiles:
- '*'
supplementalGroups: 8
  type: RunAsAny
users: 9
- system:serviceaccount:default:registry
- system:serviceaccount:default:router
- system:serviceaccount:openshift-infra:build-controller
volumes:
- '*'
```

1. Pod が要求できる機能の一覧です。特殊な記号 * は任意の機能を許可しますが、一覧が空の場合は、いずれの機能も要求できないことを意味します。
2. Pod に含める追加機能の一覧です。
3. セキュリティーコンテキストの許可される値を定める **FSGroup** ストラテジータイプです。
4. この SCC へのアクセスを持つグループです。
5. Pod から取り除く機能の一覧です。または、**ALL** を指定してすべての機能をドロップします。
6. セキュリティーコンテキストの許可される値を定める **runAsUser** ストラテジータイプです。
7. セキュリティーコンテキストの許可される値を定める **seLinuxContext** ストラテジータイプです。
8. セキュリティーコンテキストの許可される補助グループを定める **supplementalGroups** ストラテジーです。
9. この SCC にアクセスできるユーザーです。

SCC の **users** および **groups** フィールドは SCC にアクセスできるユーザー制御します。デフォルトで、クラスター管理者、ノードおよびビルドコントローラーには特権付き SCC へのアクセスが付与されます。認証されたすべてのユーザーには **restricted-v2** SCC へのアクセスが付与されます。

明示的な runAsUser 設定を使用しない場合

```
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo
spec:
  securityContext: 1
  containers:
  - name: sec-ctx-demo
    image: gcr.io/google-samples/node-hello:1.0
```

-

1. コンテナーまたは Pod が実行時に使用するユーザー ID を要求しない場合、有効な UID はこの Pod を作成する SCC よって異なります。**restricted-v2** SCC はデフォルトですべての認証ユーザーに付与されるため、ほとんどの場合はすべてのユーザーおよびサービスアカウントで利用でき、使用されます。**restricted-v2** SCC は、**securityContext.runAsUser** フィールドの使用できる値を制限し、これをデフォルトに設定するために **MustRunAsRange** ストラテジーを使用します。受付プラグインではこの範囲を指定しないため、現行プロジェクトで **openshift.io/sa.scc.uid-range** アノテーションを検索して範囲フィールドにデータを設定します。最終的にコンテナーの **runAsUser** は予測が困難な範囲の最初の値と等しい値になります。 予測が困難であるのはすべてのプロジェクトにはそれぞれ異なる範囲が設定されるためです。

明示的な runAsUser 設定を使用する場合

```
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo
spec:
  securityContext:
    runAsUser: 1000 1
  containers:
    - name: sec-ctx-demo
      image: gcr.io/google-samples/node-hello:1.0
```

1. 特定のユーザー ID を要求するコンテナーまたは Pod が Red Hat OpenShift Container Platform on AWS で受け入れられるのは、サービスアカウントまたはユーザーに、そのユーザー ID を許可するように、SCC へのアクセスが付与されている場合のみです。SCC は、任意の ID や特定の範囲内にある ID、または要求に固有のユーザー ID を許可します。

この設定は、SELinux、fsGroup、および Supplemental Groups について有効です。

1.4. SCC (SECURITY CONTEXT CONSTRAINTS) の作成

OpenShift CLI (**oc**) を使用して SCC (Security Context Constraints) を作成することができます。

前提条件

- OpenShift CLI (**oc**) をインストールしている。
- **cluster-admin** ロールを持つユーザーとしてクラスターにログインしている。

手順

1. **scc_admin.yaml** という名前の YAML ファイルで SCC を定義します。

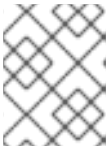
   SecurityContextConstraints オブジェクト定義

   ```
   kind: SecurityContextConstraints
   apiVersion: security.openshift.io/v1
   metadata:
     name: scc-admin
   allowPrivilegedContainer: true
   ```

```
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
fsGroup:
  type: RunAsAny
supplementalGroups:
  type: RunAsAny
users:
- my-admin-user
groups:
- my-admin-group
```

オプションとして、**requiredDropCapabilities** フィールドに必要な値を設定して、SCC の特定の機能を取り除くことができます。指定された機能はコンテナーからドロップされます。すべてのケイパビリティーを破棄するには、**ALL** を指定します。たとえば、**KILL** 機能、**MKNOD** 機能、および **SYS_CHROOT** 機能のない SCC を作成するには、以下を SCC オブジェクトに追加します。

```
requiredDropCapabilities:
- KILL
- MKNOD
- SYS_CHROOT
```



注記

allowedCapabilities と **requiredDropCapabilities** の両方に、機能を追加できません。

CRI-O は、Docker ドキュメント に記載されている同じ一連の機能の値をサポートし ます。

2. ファイルを渡して SCC を作成します。

```
$ oc create -f scc_admin.yaml
```

出力例

```
securitycontextconstraints "scc-admin" created
```

検証

- SCC が作成されていることを確認します。

```
$ oc get scc scc-admin
```

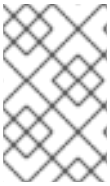
出力例

```
NAME        PRIV    CAPS    SELINUX    RUNASUSER   FSGROUP    SUPGROUP
PRIORITY   READONLYROOTFS   VOLUMES
scc-admin   true    []      RunAsAny   RunAsAny    RunAsAny   RunAsAny   <none>    false
```

```
[awsElasticBlockStore azureDisk azureFile cephFS cinder configMap downwardAPI
emptyDir fc flexVolume flocker gcePersistentDisk gitRepo glusterfs iscsi nfs
persistentVolumeClaim photonPersistentDisk quobyte rbd secret vsphere]
```

1.5. SCC (SECURITY CONTEXT CONSTRAINTS) へのロールベースのアクセス

SCC は RBAC で処理されるリソースとして指定できます。これにより、SCC へのアクセスのスコープを特定プロジェクトまたはクラスター全体に設定できます。ユーザー、グループ、またはサービスアカウントを SCC に直接割り当てると、クラスター全体のスコープが保持されます。



注記

SCC をデフォルト namespace (**default**、**kube-system**、**kube-public**、**openshift-node**、**openshift-infra**、および **openshift**) のいずれかに作成します。これらの namespace は Pod またはサービスの実行に使用しないでください。

ロールの SCC へのアクセスを組み込むには、ロールの作成時に **scc** リソースを指定します。

```
$ oc create role <role-name> --verb=use --resource=scc --resource-name=<scc-name> -n
<namespace>
```

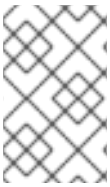
これにより、以下のロール定義が生成されます。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
...
  name: role-name 1
  namespace: namespace 2
...
rules:
- apiGroups:
  - security.openshift.io 3
  resourceNames:
  - scc-name 4
  resources:
  - securitycontextconstraints 5
  verbs: 6
  - use
```

1. ロールの名前。
2. 定義されたロールの namespace。指定されていない場合は、**default** にデフォルト設定されます。
3. **SecurityContextConstraints** リソースを含む API グループ。**scc** がリソースとして指定される場合に自動的に定義されます。
4. アクセスできる SCC の名前のサンプル。
5. ユーザーが SCC 名を **resourceNames** フィールドに指定することを許可するリソースグループの名前。

6 ロールに適用する動詞の一覧。

このようなルールを持つローカルまたはクラスターロールは、ロールバインディングまたは クラスターロールバインディングでこれにバインドされたサブジェクトが **scc-name** というユーザー定義の SCC を使用することを許可します。



注記

RBAC はエスカレーションを防ぐように設計されているため、プロジェクト管理者であっても SCC へのアクセスを付与することはできません。デフォルトでは、**restricted-v2** SCC を含め、SCC リソースで動詞 **use** を使用することは許可されていません。

1.6. SCC (SECURITY CONTEXT CONSTRAINTS) コマンドのリファレンス

OpenShift CLI (**oc**) を使用して、インスタンスの SCC (Security Context Constraints) を通常の API オブジェクトとして管理できます。

1.6.1. SCC (Security Context Constraints) の表示

SCC の現在の一覧を取得するには、以下を実行します。

```
$ oc get scc
```

出力例

```
NAME                              PRIV    CAPS                    SELINUX     RUNASUSER          FSGROUP
SUPGROUP    PRIORITY     READONLYROOTFS   VOLUMES
anyuid                            false   <no value>              MustRunAs   RunAsAny           RunAsAny
RunAsAny    10           false
["configMap","downwardAPI","emptyDir","persistentVolumeClaim","projected","secret"]
hostaccess                        false   <no value>              MustRunAs   MustRunAsRange     MustRunAs
RunAsAny    <no value>   false
["configMap","downwardAPI","emptyDir","hostPath","persistentVolumeClaim","projected","secret"]
hostmount-anyuid                  false   <no value>              MustRunAs   RunAsAny           RunAsAny
RunAsAny    <no value>   false
["configMap","downwardAPI","emptyDir","hostPath","nfs","persistentVolumeClaim","projected","secret"]

hostnetwork                       false   <no value>              MustRunAs   MustRunAsRange     MustRunAs
MustRunAs   <no value>   false
["configMap","downwardAPI","emptyDir","persistentVolumeClaim","projected","secret"]
hostnetwork-v2                    false   ["NET_BIND_SERVICE"]    MustRunAs   MustRunAsRange
MustRunAs   MustRunAs    <no value>   false
["configMap","downwardAPI","emptyDir","persistentVolumeClaim","projected","secret"]
node-exporter                     true    <no value>              RunAsAny    RunAsAny           RunAsAny
RunAsAny    <no value>   false        ["*"]
nonroot                           false   <no value>              MustRunAs   MustRunAsNonRoot   RunAsAny
RunAsAny    <no value>   false
["configMap","downwardAPI","emptyDir","persistentVolumeClaim","projected","secret"]
nonroot-v2                        false   ["NET_BIND_SERVICE"]    MustRunAs   MustRunAsNonRoot
RunAsAny    RunAsAny     <no value>   false
["configMap","downwardAPI","emptyDir","persistentVolumeClaim","projected","secret"]
privileged                        true    ["*"]                   RunAsAny    RunAsAny           RunAsAny    RunAsAny
<no value>   false        ["*"]
```

```
restricted                false   <no value>          MustRunAs   MustRunAsRange     MustRunAs
RunAsAny    <no value>   false
["configMap","downwardAPI","emptyDir","persistentVolumeClaim","projected","secret"]
restricted-v2             false   ["NET_BIND_SERVICE"]   MustRunAs   MustRunAsRange
MustRunAs   RunAsAny    <no value>   false
["configMap","downwardAPI","emptyDir","persistentVolumeClaim","projected","secret"]
```

1.6.2. SCC (Security Context Constraints) の検証

特定の SCC についての情報 (SCC が適用されるユーザー、サービスアカウントおよびグループを含む) を表示できます。

たとえば、**restricted** SCC を検査するには、以下を実行します。

```
$ oc describe scc restricted
```

出力例

```
Name:                         restricted
Priority:                     <none>
Access:
  Users:                      <none> (1)
  Groups:                     <none> (2)
Settings:
  Allow Privileged:               false
  Allow Privilege Escalation:     true
  Default Add Capabilities:       <none>
  Required Drop Capabilities:     KILL,MKNOD,SETUID,SETGID
  Allowed Capabilities:           <none>
  Allowed Seccomp Profiles:       <none>
  Allowed Volume Types:
configMap,downwardAPI,emptyDir,persistentVolumeClaim,projected,secret
  Allowed Flexvolumes:            <all>
  Allowed Unsafe Sysctls:         <none>
  Forbidden Sysctls:              <none>
  Allow Host Network:             false
  Allow Host Ports:               false
  Allow Host PID:                 false
  Allow Host IPC:                 false
  Read Only Root Filesystem:      false
  Run As User Strategy: MustRunAsRange
    UID:                          <none>
    UID Range Min:                <none>
    UID Range Max:                <none>
  SELinux Context Strategy: MustRunAs
    User:                         <none>
    Role:                         <none>
    Type:                         <none>
    Level:                        <none>
  FSGroup Strategy: MustRunAs
    Ranges:                       <none>
  Supplemental Groups Strategy: RunAsAny
    Ranges:                       <none>
```

1. SCC が適用されるユーザーとサービスアカウントを一覧表示します。
2. SCC が適用されるグループを一覧表示します。