NEC

iStorage V100/V110/V300/V310/V310F

NEC Storage Plug-in for Containers Quick Reference Guide



iStorage

IV-UG-402-005-10

目次

第 1 章 概要 1

1.1 NEC Storage Plug-in for Containers について 1

1.2 環境設定タスクについて 2

1.3 要件 2

1.3.1 サポート対象のコンテナオーケストレーター 2

1.3.2 サーバー要件 3

1.3.3 ストレージ要件 3

1.3.4 ネットワーク要件 3

1.4 インストール前のタスク 4

1.4.1 サーバーのインストール前のタスク 4

1.4.1.1 Device Mapper Multipath 設定 4

1.4.2 ストレージのインストール前のタスク 6

1.4.2.1 iSCSI ターゲットの命名規則 7

第 2 章 インストール 8

2.1 OpenShift へのインストール 8

2.2 Storage Plug-in for Containers インスタンスの設定 9

2.3 環境変数 10

第 3 章 使用法 12

3.1 Secret の設定 12

3.2 StorageClass の設定 13

3.3 PersistentVolumeClaim の設定 14

3.4 Pod の設定 16

3.5 コマンド例 17

3.6 Raw Block Volume 19

3.7 ReadWriteMany 20

3.8 ReadOnlyMany 21

3.9 ストレージリソースの分割 22

3.10 Static provisioning 25

3.10.1 Static provisioning を利用する場合の要件 25

3.10.2 Secret および StorageClass を作成する 26

3.10.3 PV を作成する 26

3.10.4 PVC を作成する 30

3.10.5 PVC のクローン作成 31
3.10.6 PVC のスナップショット作成 33
3.10.7 PVC のストレージボリュームの拡張 37
3.10.8 Storage Plug-in for Containers によって作成されたリソースの削除 38
3.10.9 Static provisioning 利用時のトラブルシューティング 39
第 4 章 アップグレード 41
4.1 OpenShift でのアップグレード 41
第 5 章 再作成 42
5.1 Storage Plug-in for Containers を削除する 42
5.2 Storage Plug-in for Containers を作成する 42
第 6 章 アンインストール 43
6.1 OpenShift でのアンインストール 43
第 7 章 トラブルシューティング 44
7.1 障害発生時に収集する情報 44
7.1.1 障害対応窓口への連絡時に必要な情報 44
7.1.2 Storage Plug-in for Containers 情報の収集 45
7.1.3 ストレージシステム情報の収集 46
7.2 PersistentVolume の情報確認 46
7.3 Pod を強制削除する際の注意事項 46
7.4 PersistentVolumeClaim の多重作成、多重削除 47
7.5 ホストグループの設定 47
7.6 Pod 作成または Pod 削除時のタイムアウトエラー 48
7.7 Openshift Pod のスケジュールまたは作成ができない 48
付録 A. 既存の LUN 構成の編集 50
付録 B. `StorageClass` の既存ポートの更新 51

はじめに

NEC Storage Plug-in for Containers によってストレージシステムのボリュームが動的に作成され、コンテナの永続ボリュームとして使用することが可能です。これにより、コンテナ内でステートフルなアプリケーションを実行できます。このマニュアルでは、実装の概要を説明し、Storage Plug-in for Containers の使用要件、インストール、および設定について説明します。

対象読者

このマニュアルは、NEC Storage Plug-in for Containers をインストール、設定、および実行するシステム管理者を対象にしています。

また、次の知識をお持ちであることを前提にしています。

- コンテナ化された環境とその基本的な機能
- ストレージシステム

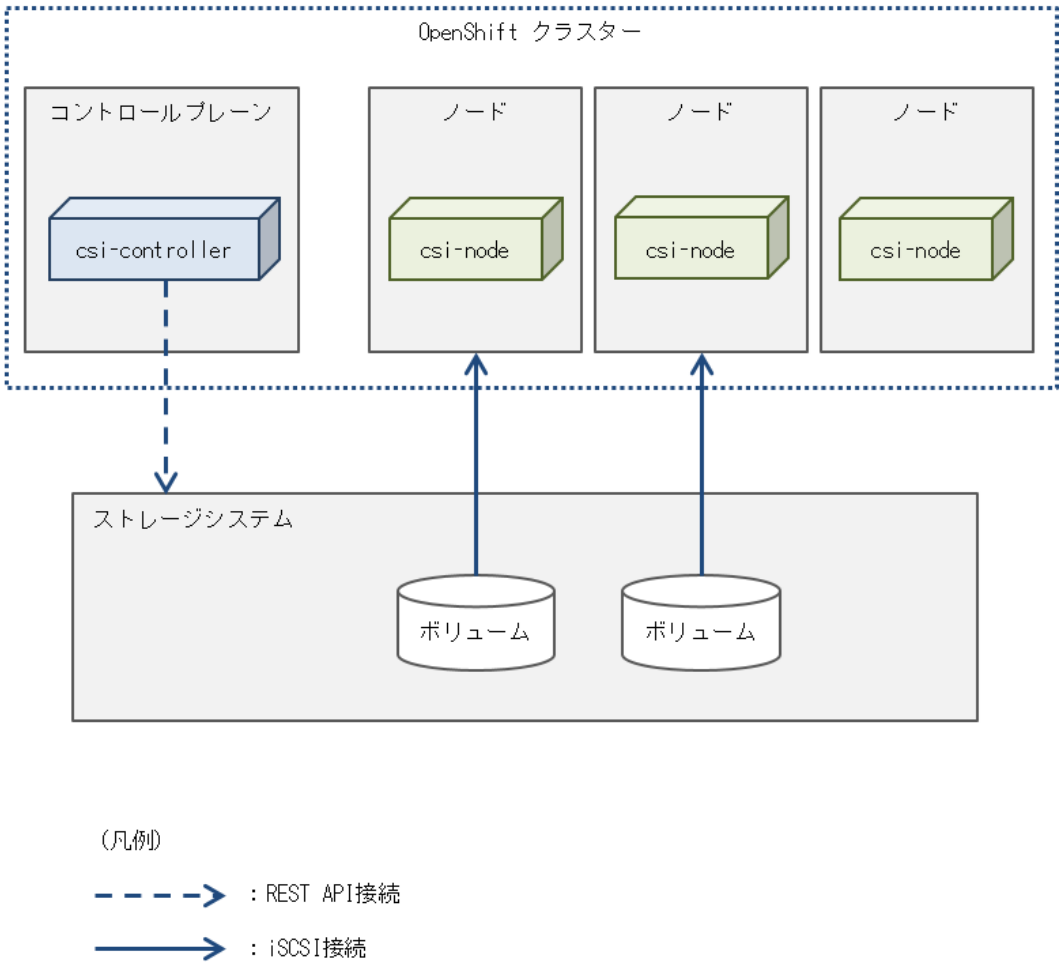
第 1 章
概要

Storage Plug-in for Containers は、OpenShift 環境で NEC ストレージの永続ボリュームを作成し管理するためのソフトウェアです。Storage Plug-in for Containers により、OpenShift 上で動作するステートフルなアプリケーションから、NEC ストレージのボリュームを利用できます。永続ボリュームに対しては、作成および削除のほかにも、スナップショットやクローンなどのオペレーションもサポートされています。

1.1　NEC Storage Plug-in for Containers について

Storage Plug-in for Containers は、CSI (Container Storage Interface)を利用し、OpenShift 環境とストレージシステムのインテグレーションを実現します。

次の図に、Storage Plug-in for Containers がデプロイされているコンテナ環境の例を示します。



Storage Plug-in for Containers を構成するコンポーネントを次の表に示します。

コンポーネント	用途
csi-controller	主に REST API によるストレージ操作用の CSI コントローラーサービスを実装します。 このコンポーネントは Deployment としてデプロイされ、コントロールプレーンで起動します。コントロールプレーンで起動できない場合、ノードで起動することがあります。
csi-node	主に各ノードのボリュームを管理する CSI ノードサービスを実装します。 このコンポーネントは DaemonSet として展開され、すべてのノードにこのコンポーネントが必要です。
ストレージシステム	コンテナにストレージを提供します。

1.2　環境設定タスクについて

Storage Plug-in for Containers を使用すると、コンテナ使用時にストレージシステムを動的に操作できます。Storage Plug-in for Containers を使用するには、インストール前のタスクを完了する必要があります。

操作手順

1. Storage Plug-in for Containers、OpenShift をインストールするサーバー、ストレージシステム、および OpenShift の要件を確認して適用します。
2. インストール前のタスクを実行します。
   a. OpenShift の環境を設定します。
   b. ストレージシステムを設定します。
3. Storage Plug-in for Containers をインストールします。

1.3　要件

Storage Plug-in for Containers をインストールする前に、システム要件が次の最小要件を満たしていることを確認します。

1.3.1　サポート対象のコンテナオーケストレーター

コンテナオーケストレーター	サポートバージョン	備考
Red Hat OpenShift Container Platform	4.18、4.19、および 4.20	–

サポートされているバージョンの詳細については、リリースノートを参照してください。

1.3.2 サーバー要件

コンポーネント	要件
CPU	x86_64
オペレーティングシステム	詳細については、リリースノートを参照してください。
インターフェイス	ベアメタルサーバー：iSCSI 仮想マシン：iSCSI

1.3.3 ストレージ要件

コンポーネント	要件
モデルおよびファームウェア	詳細については、リリースノートを参照してください。
インターフェイス	iSCSI
ユーザーアカウント	ビルトイングループである Storage Administrator (View & Modify)に所属するユーザー、またはそれと同等のロールを持つグループに所属するユーザー
ライセンス	次のライセンスが必要です。 • Dynamic Provisioning • Snapshot • (iStorage V110, V310/V310F 向け)Snapshot Advanced
SVP	シングルおよびデュアルの SVP 構成がサポートされています。 **メモ** SVP は iStorage V110, V310/V310F ストレージシステムでは必要ありません。

1.3.4 ネットワーク要件

Storage Plug-in for Containers では次のネットワーク要件があります。

- Storage Plug-in for Containers は以下のポートを使用します。ファイアウォール設定の参考にしてください。

コンポーネント	ポート	使用法	備考
ストレージシステム	80 または 443	REST API の接続	なし

- Storage Plug-in for Containers は IPv6 をサポートしていません。IPv4 を使用してください。

1.4　インストール前のタスク

Storage Plug-in for Containers をインストールする前に、サーバーとストレージのインストール前の要件を確認して適用します。

1.4.1　サーバーのインストール前のタスク

サーバーコンポーネントごとに、インストール前のタスクの概要を次の表に示します。

コンポーネント	タスク
ハイパーバイザー	仮想マシンを使用する場合は、ハイパーバイザーをセットアップします。 **メモ** Storage Plug-in for Containers は VMware vSphere 7.0/8.0 でテストしています。
iSCSI	ストレージシステムと iSCSI 接続を実施するノードに iSCSI イニシエーターのソフトウェアがインストールされていることを確認します。インストールされていない場合は、次の URL を参照してインストールしてください。 https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/managing_storage_devices/configuring-an-iscsi-initiator_managing-storage-devices#creating-an-iscsi-initiator_configuring-an-iscsi-initiator **メモ** Storage Plug-in for Containers は大文字を含む IQN をサポートしていません。
マルチパス機能	Device Mapper Multipath を使用してください。Device Mapper Multipath の設定については、1.4.1.1　Device Mapper Multipath 設定（4 ページ）を参照してください。

1.4.1.1　Device Mapper Multipath 設定

Device Mapper Multipath を有効にし、`user_friendly_names` オプションに `yes` が設定されていることを確認します。

例：

```
defaults {
        user_friendly_names yes
        find_multipaths yes
}
blacklist {
}
```

メモ

設定値は環境によって異なる場合がありますので、各 OS のドキュメントも参照してください。

- Red Hat Enterprise Linux7：https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/dm_multipath/mpio_setup
- Red Hat Enterprise Linux8：https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_device_mapper_multipath/configuring-dm-multipath_configuring-device-mapper-multipath
- Red Hat Enterprise Linux 9：https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/configuring_device_mapper_multipath/configuring-dm-multipath_configuring-device-mapper-multipath

---

OpenShift の場合は、MachineConfig YAML ファイルを使用する必要があります。詳細については公式ドキュメントを参照してください。https://docs.openshift.com/container-platform/latest/machine_configuration/index.html

手順の例を次に示します。

操作手順

1. `multipath-machineconfig-sample.yaml` を取得します。

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: multipath-machineconfig-sample
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-8; base64,ZGVmYXVsdHMgewp1c2VyX2ZyaWVuZGx5X25hbWVzIHllcwpmaW5kX211bHRpcGF0aHMgeWVzCn0KYmxhY2tsaXN0IHsKfQo=
          verification: {}
        filesystem: root
        mode: 400
        path: /etc/multipath.conf
```

2. 必要に応じて、`multipath-machineconfig-sample.yaml` のマルチパスの設定を変更します。

   次のデフォルトのマルチパスの設定は、`multipath-sample.conf` に記載されています。このファイルを base64 でエンコードした文字列が `multipath-machineconfig-sample.yaml` に記載されています。

```
defaults {
user_friendly_names yes
find_multipaths yes
```

```
}
blacklist {
}
```

a. multipath-sample.conf を取得します。

b. multipath-sample.conf を編集して、マルチパスの設定を変更します。

c. 次のコマンドを実行し、multipath-sample.conf を base64 でエンコードして取得します。

```
# cat multipath-sample.conf | base64 -w0
```

d. multipath-machineconfig-sample.yaml の spec.config.storage.files.contents.source の設定を変更します。

spec.config.storage.files.contents.source には、マルチパスの設定が base64 でエンコードされた文字列が記載されています。この文字列を multipath-sample.conf から base64 でエンコードして取得した文字列に置き換えてください。

3. 次のコマンドを実行します。

```
# oc apply -f multipath-machineconfig-sample.yaml
```

メモ

この MachineConfig は、コンピュートノードにのみ反映されます。MachineConfig が作成されると、すべてのコンピュートノードが 1 つずつ再起動され、すべてのコンピュートノードで/etc/multipath.conf が作成されます。

4. 各コンピュートノードで/etc/multipath.conf を開いて設定が反映されていることを確認します。

メモ

設定が適用されるまでに時間がかかる場合があります。

1.4.2　ストレージのインストール前のタスク

ストレージコンポーネントごとに、完了しておく必要があるインストール前のタスクの概要を、次の表に示します。

コンポーネント	タスク
プール	DP プールを作成します。 Dynamic Tiering はサポートされていません。
iSCSI 接続	Storage Navigator を使用して、ポートセキュリティーを有効にします。 Storage Plug-in for Containers は自動的に次のアクションを実行します。 • iSCSI ターゲットがない場合は、ホストごとに iSCSI ターゲットを作成します。

コンポーネント	タスク
	既存の iSCSI ターゲットを使用する場合は、命名規則に従って iSCSI ターゲット名を変更します（「1.4.2.1　iSCSI ターゲットの命名規則（7 ページ）」を参照してください）。 • OpenShift クラスターに参加する各ホストに対応する iSCSI ターゲットに IQN を追加します。 • 各ホストの iSCSI ターゲットにログインします。 CHAP を使用する場合は、以下の手順を実施します。 • iSCSI ターゲットを作成します（「1.4.2.1　iSCSI ターゲットの命名規則（7 ページ）」を参照してください）。 • ポートおよび iSCSI ターゲットに CHAP を設定します。 • CHAP 認証で iSCSI ターゲットにログインします。ログインは各ホストから実行してください。 **メモ** 既存の iSCSI ターゲットにホストモードオプションが設定されていても、Storage Plug-in for Containers はホストモードオプションを上書きします。

1.4.2.1　iSCSI ターゲットの命名規則

Storage Plug-in for Containers は、iSCSI ターゲットの名前を特定のルールに基づいて検索します。

既存の iSCSI ターゲットを使用する場合は、ストレージ接続に応じて、iSCSI ターゲットの命名規則を参照してください。

iSCSI ターゲットの命名規則

Storage Plug-in for Containers は特定の命名規則で iSCSI ターゲットを検索します。iSCSI ターゲットを見つけられない場合、Storage Plug-in for Containers は iSCSI ターゲット"spc-<*ハッシュ化された IQN*>"を自動的に作成します。すでに iSCSI ターゲットがある場合、それらの iSCSI ターゲットを削除するか、次の命名規則に従って iSCSI ターゲット名を変更する必要があります。

"spc-<*任意の文字列*>"

第 2 章
インストール

ここでは、Storage Plug-in for Containers をインストールする方法について説明します。

2.1 OpenShift へのインストール

Storage Plug-in for Containers は、OperatorHub からインストールできる Operator を使用して、OpenShift に簡単に展開できます。Storage Plug-in for Containers をインストールするには、次の手順に従います。

メモ

- Storage Plug-in for Containers の古いバージョンがインストールされている場合、インストール手順を実行する前にアンインストールしてください。
- インターネットにアクセスできない OpenShift Container Platform 環境に Storage Plug-in for Containers をインストールしたい場合は、事前に certified-operators カタログをミラーリングしてください。手順の詳細については、https://docs.openshift.com/container-platform/latest/disconnected/mirroring/installing-mirroring-installation-images.html#olm-mirror-catalog_installing-mirroring-installation-images を参照してください。

  例えば、OpenShift Container Platform バージョン 4.10 の場合、certified-operators カタログの Index image は、registry.redhat.io/redhat/certified-operator-index:v4.10 です。詳細については、https://docs.openshift.com/container-platform/latest/operators/understanding/olm-rh-catalogs.html を参照してください。

操作手順

1. OpenShift の Web コンソールを使用して OperatorHub にアクセスします。
2. NEC Storage Plug-in for Containers を検索し、Operator をインストールします。

   メモ

   Operator Subscription で次の設定を選択します。

   - Installation mode: **A specific namespace on the cluster** を選択し、任意の namespace を指定します。
   - Update approval: **Manual** を選択し、Install Plan を承認します（https://docs.openshift.com/ を参照してください）。
3. Operator のステータスが **Succeeded** であることを確認します。
4. Operator Pod のステータスが **Running** であることを確認します。
5. Operator Details で **Create Instance** をクリックします。

6. **Create** をクリックします。高度な設定を実施する場合は、「2.2　Storage Plug-in for Containers インスタンスの設定（9 ページ）」を参照してください。
7. 次のコマンドを使用して、ステータス READY が **true** であることを確認します。

```
# oc get nspc -n <Storage Plug-in for Containers の namespace>
NAME    READY    AGE
nspc    true     30s
```

2.2　Storage Plug-in for Containers インスタンスの設定

CustomResource YAML ファイルを編集することで、Storage Plug-in for Containers を設定できます。CustomResource YAML ファイルのパラメーターを次に示します。

パラメーター	説明
spec.imagePullSecrets	イメージを取得する場合に Secret が必要なときは指定します。
spec.controller.containers.name	nspc-csi-controller Pod 内に設定する Storage Plug-in for Containers の名前です。 たとえば、nspc-csi-driver や、csi-provisioner などは、nspc-csi-controller 内のコンテナ名のキーとなります。 コンテナ名を取得するためには、**oc describe deployment nspc-csi-controller -n <SPC_NAMESPACE の値>**コマンドを使用します。
spec.controller.containers.image	nspc-csi-controller のイメージ名です。
spec.controller.containers.imagePullPolicy	nspc-csi-controller のイメージ取得ポリシーです。デフォルト値は、IfNotPresent です。
spec.controller.containers.env	nspc-csi-controller コンテナに設定する環境変数のリストです。「2.3　環境変数（10 ページ）」を参照してください。
spec.controller.containers.args	nspc-csi-controller のエントリーポイントに対する引数です。この引数は、デプロイされている nspc-csi-controller コンテナの spec.template.spec.containers.args にあるすべてのパラメーターを置き換えます。
spec.controller.tolerations	nspc-csi-controller を実行している Pod の Toleration を指定します。Kubernetes と同じ形式を使用します。詳細については、https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/ を参照してください。
spec.controller.affinity.nodeAffinity	nspc-csi-controller を実行している Pod の Node Affinity を指定します。Kubernetes と同じ形式を使用します。詳細については、https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes-using-node-affinity/を参照してください。
spec.node.containers.name	nspc-csi-node Pod 内に設定するコンテナの名前です。

パラメーター	説明
	たとえば、nspc-csi-driver や、liveness-probe などは、nspc-csi-node 内のコンテナ名のキーとなります。 コンテナ名を取得するためには、**oc describe daemonset nspc-csi-node -n <SPC_NAMESPACE の値>**コマンドを使用します。
spec.node.containers.image	nspc-csi-node のイメージ名です。
spec.node.containers.imagePullPolicy	nspc-csi-node のイメージ取得ポリシーです。デフォルト値は、IfNotPresent です。
spec.node.containers.env	nspc-csi-node コンテナ内に設定する環境変数のリストです。
spec.node.containers.args	nspc-csi-node のエントリーポイントに対する引数です。この引数は、デプロイされている nspc-csi-node コンテナの spec.template.spec.containers.args にあるすべてのパラメーターを置き換えます。
spec.node.tolerations	nspc-csi-node を実行している Pod の Toleration を指定します。Kubernetes と同じ形式を使用します。詳細については、https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/を参照してください。
spec.node.affinity.nodeAffinity	nspc-csi-node を実行している Pod の Node Affinity を指定します。Kubernetes と同じ形式を使用します。詳細については、https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes-using-node-affinity/を参照してください。

2.3　環境変数

npsc-csi-controller の nspc-csi-driver の環境変数を次に示します。

環境変数名	説明
SPC_VERIFY_CERTIFICATE	true を指定した場合、ストレージシステムとの HTTPS 接続時に TLS 証明書が検証されます。デフォルト値は、false です。
TZ	ログ取得時のタイムゾーンを指定します（例：Asia/Tokyo）。デフォルト値は、UTC です。

nspc-csi-driver の証明書を検証できるようにする例を以下に示します。

1. 次のコマンドで、現在の設定を確認します。

```
# oc get deployment -n <SPC_NAMESPACE の値> nspc-csi-controller -o yaml
<…>
    - name: nspc-csi-driver
      env:
      - name: CSI_ENDPOINT
        value: unix:///csi/csi-controller.sock
      - name: KUBE_NODE_NAME
        valueFrom:
          fieldRef:
```

```
            apiVersion: v1
            fieldPath: spec.nodeName
<…>
```

2. Storage Plug-in for Containers マニフェストに、env: SPC_VERIFY_CERTIFICATE を追加します。

```
apiVersion: csi.nec.com/v1
kind: NSPC
metadata:
  name: nspc
  namespace: <SPC_NAMESPACE の値>
spec:
  controller:
    containers:
      - name: nspc-csi-driver
        env:
        - name: SPC_VERIFY_CERTIFICATE
          value: "true"
```

3. Storage Plug-in for Containers をアンインストールして再度インストールします。
Storage Plug-in for Containers のアンインストールおよび再インストールの方法については、「第 2 章　インストール（8 ページ）」および「第 6 章　アンインストール（43 ページ）」を参照してください。

4. 変更内容を確認します。

```
# oc get deployment -n <SPC_NAMESPACE の値> nspc-csi-controller -o yaml
<…>
    - name: nspc-csi-driver
      env:
      - name: CSI_ENDPOINT
        value: unix:///csi/csi-controller.sock
      - name: KUBE_NODE_NAME
        valueFrom:
          fieldRef:
            apiVersion: v1
            fieldPath: spec.nodeName
      - name: SPC_VERIFY_CERTIFICATE
        value: "true"
<…>
```

第 3 章
使用法

ここでは、Storage Plug-in for Containers で使用される各コンポーネントの設定とコマンド例について説明します。

3.1 Secret の設定

Secret ファイルには、Storage Plug-in for Containers がご利用の環境で動作するために必要なストレージの URL、ユーザー名、およびパスワードが含まれています。Secret の例を次に示します。

secret-sample.yaml のパラメーターリファレンス

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-sample             #(1)
type: Opaque
data:
  url: aHR0cDovLzE3Mi4xNi4xLjE=   #(2)
  user: VXNlcjAx                  #(3)
  password: UGFzc3dvcmQwMQ==      #(4)
  hostModeOptions: ODgsODENCg==   #(5)
```

凡例：

(1) Secret 名

(2) base64 でエンコードされたストレージの URL

iStorage V100、V300 シリーズの場合、ストレージコントローラーの IP アドレスを使用してください。

iStorage V110、V310/V310F の場合、サービス IP アドレスを使用してください。

例：

```
echo -n "http://172.16.1.1" | base64
```

(3) base64 でエンコードされたストレージのユーザー名。

例：

```
echo -n "User01" | base64
```

(4) base64 でエンコードされたストレージのパスワード。

例：

```
echo -n "Password01" | base64
```

（5）base64 エンコードされたホストモードオプション。

例：

```
echo -n "88,81" | base64
```

メモ

デフォルトでは、ホストモードオプションは 2、22、25、68、および 91 に設定されています。追加のホストモードオプションを追加する必要がある場合は、`secret.yaml` ファイル内の `hostMode Options` パラメーターで指定する必要があります。ユーザー定義のホストモードオプションを適用するには、Pod を削除して再作成する必要があります。

3.2 StorageClass の設定

StorageClass ファイルには、Storage Plug-in for Containers がご利用の環境で動作するために必要なストレージの設定が含まれています。StorageClass の例を次に示します。

メモ

StorageClass および PVC を作成したあとで StorageClass を再作成した場合、作成済みの既存 PVC には設定が反映されません。

sc-sample.yaml のパラメーターリファレンス

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-sample                          #(1)
  annotations:
    kubernetes.io/description: NEC Storage Plug-in for Containers
provisioner: nspc.csi.nec.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:
  serialNumber: "600001"                   #(2)
  poolID: "1"                              #(3)
  portID : CL1-A,CL2-A                     #(4)
  connectionType: iscsi                    #(5)
  storageEfficiency: "CompressionDeduplication"      #(6)
  storageEfficiencyMode: "PostProcess"               #(7)
  csi.storage.k8s.io/fstype: ext4                    #(8)
  csi.storage.k8s.io/node-publish-secret-name: "secret-sample"         #(9)
  csi.storage.k8s.io/node-publish-secret-namespace: "default"          #(10)
  csi.storage.k8s.io/provisioner-secret-name: "secret-sample"          #(9)
  csi.storage.k8s.io/provisioner-secret-namespace: "default"           #(10)
  csi.storage.k8s.io/controller-publish-secret-name: "secret-sample"   #(9)
  csi.storage.k8s.io/controller-publish-secret-namespace: "default"    #(10)
```

```
  csi.storage.k8s.io/node-stage-secret-name: "secret-sample"           #(9)
  csi.storage.k8s.io/node-stage-secret-namespace: "default"            #(10)
  csi.storage.k8s.io/controller-expand-secret-name: "secret-sample"    #(9)
  csi.storage.k8s.io/controller-expand-secret-namespace: "default"     #(10)
```

凡例：

(1) StorageClass 名

(2) ストレージのシリアル番号

(3) DP プール ID

(4) ポート ID。マルチパスの場合はコンマで区切って指定します。

(5) ストレージとノード間の接続タイプ。iscsi がサポートされています。必ず記載してください。

(6) 容量削減機能の有効化。"Compression"、"CompressionDeduplication"、および"Disabled"がサポートされています。デフォルトは"Disabled"で、"Disabled"の場合は容量削減機能は無効になります。

メモ

iStorage V110、V310/V310F の場合、"Disabled"はサポートされていません。デフォルトは"CompressionDeduplication"になります。

(7) 容量削減機能の実行モード。storageEfficiency が"Compression"または"CompressionDeduplication"の場合に指定できるパラメーターで、"Inline"および"PostProcess"がサポートされています。storageEfficiencyMode の指定がない場合、"Inline"で動作します。パラメーターの詳細については、『システム構築ガイド』の容量削減機能の説明を参照してください。

⚠ 注意

Storage Plug-in for Containers で作成された LDEV の場合、容量削減機能に関するパラメーターは変更しないでください。

(8) ファイルシステムタイプ。ext4 と xfs がサポートされています。csi.storage.k8s.io/fstype の指定がない場合、ext4 が設定されます。

(9) Secret 名

(10) Secret namespace

3.3　PersistentVolumeClaim の設定

このセクションでは、Storage Plug-in for Containers によってストレージシステムに新規ボリュームを動的に作成するために必要な PersistentVolumeClaim を設定します。

PersistentVolumeClaim ファイルには、Storage Plug-in for Containers が PersistentVolume を作成するために使用するボリューム情報が含まれています。PersistentVolumeClaim の例を次に示します。

メモ

- ストレージシステムの既存ボリュームを PersistentVolumeClaim として利用したい場合は、3.10　Static provisioning（25 ページ）を参照してください。
- このセクションで設定する PersistentVolumeClaim と Static provisioning の機能を同時に使用する場合、3.10.3　PV を作成する（26 ページ）の手順で作成した静的な PV は、3.10.4　PVC を作成する（30 ページ）の手順を実施して PVC と適切に関連づける必要があります。
  3.10.4　PVC を作成する（30 ページ）の手順を実施していないときは、このセクションの PersistentVolumeClaim の設定を実施する前に、次の手順を実施してください。

  1. 関連づけが完了していない PV を確認します。

     ```
     oc get pv
     ```

     `STATUS` が `Available` の PV は関連づけが完了していません。

  2. 関連づけが完了していない PV に対し、`claimRef` が設定されているかどうかを確認します。

     ```
     oc get <PV-name> -o yaml
     ```

  3. `claimRef` が設定されていない PV がある場合、次のどちらかの手順を実施します。
     - 静的な PV を再作成して `claimRef` を設定し、3.10.4　PVC を作成する（30 ページ）の手順を実施する。
     - PV が不要な場合、PV を削除する。

`claimRef` が設定されていない PV がある場合にこのセクションの PersistentVolumeClaim の設定を実施すると、Storage Plug-in for Containers によって動的に PV が作成されず、関連づけが完了していない静的な PV と関連づけられるおそれがあります。

pvc-sample.yaml のパラメーターリファレンス

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-sample                #(1)
spec:
  accessModes:
  - ReadWriteOnce                 #(2)
  resources:
    requests:
      storage: 1Gi                #(3)
  storageClassName: sc-sample     #(4)
```

凡例：

(1) PersistentVolumeClaim 名

(2) ReadWriteOnce または ReadOnlyMany を指定してください。ReadOnlyMany を使用する場合は、「3.8　ReadOnlyMany（21 ページ）」を参照してください。

(3) ボリュームの容量

(4) StorageClass 名

PersistentVolumeClaim の使用制限

- PersistentVolumeClaim の作成時に障害が発生した場合、PersistentVolumeClaim オブジェクトは PersistentVolume なしで作成されます。この場合、次のコマンド `oc delete pvc <PVC-name>`を使用して PersistentVolumeClaim オブジェクトを削除します。
- PersistentVolumeClaim の削除時に障害が発生した場合、PersistentVolumeClaim オブジェクトは削除されますが、PersistentVolume オブジェクトは残り、PersistentVolume オブジェクトに関連付けられているストレージリソースも残る可能性があります。この場合、「7.2　PersistentVolume の情報確認（46 ページ）」を参照して、ストレージのボリューム ID を取得してください。その後、`oc delete pv <PV-name>`を実行して PersistentVolume を削除してください。最後にストレージシステムのボリュームを削除してください。詳細については、ストレージシステムのマニュアルを参照してください。

3.4　Pod の設定

Pod ファイルには、コンテナが使用するボリューム情報が含まれています。Storage Plug-in for Containers は、これらの情報を基にボリュームのマウントを行います。Pod の例を次に示します。

pod-sample.yaml のパラメーターリファレンス

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-sample              #(1)
spec:
  containers:
    - name: my-busybox
      image: busybox
      volumeMounts:
      - mountPath: "/data"      #(2)
        name: sample-volume
      command: ["sleep", "1000000"]
      imagePullPolicy: IfNotPresent
  volumes:
    - name: sample-volume
      persistentVolumeClaim:
        claimName: pvc-sample   #(3)
```

凡例：

(1) Pod 名

(2) ボリュームをマウントするコンテナ内のパス

(3) PersistentVolumeClaim 名

3.5 コマンド例

コマンドを使用して Secret、StorageClass、PersistentVolumeClaim、Pod を作成および削除する例を次に示します。

メモ

OpenShift CLI の詳細については、OpenShift の CLI リファレンスを参照してください。

Secret、StorageClass、PersistentVolumeClaim、および Pod の作成

```
# oc create -f secret-sample.yaml
secret/secret-sample created

# oc get secret
NAME            TYPE     DATA   AGE
secret-sample   Opaque   3      34s

# oc create -f sc-sample.yaml
storageclass.storage.k8s.io/sc-sample created

# oc get sc
NAME        PROVISIONER        AGE
sc-sample   nspc.csi.nec.com   21s

# oc create -f pvc-sample.yaml
persistentvolumeclaim/pvc-sample created

# oc get pvc
NAME         STATUS   VOLUME                                     CAPACITY
 ACCESS MODES   STORAGECLASS   AGE
pvc-sample   Bound    pvc-cf8c6089-0386-4c39-8037-e1520a986a7d   1Gi
 RWO            sc-sample      28s

# oc create -f pod-sample.yaml
pod/pod-sample created

# oc get pod
NAME         READY   STATUS    RESTARTS   AGE
pod-sample   1/1     Running   0          20s
```

⚠ 注意

Storage Plug-in for Containers で作成した LDEV の場合、ニックネームは変更しないでください。

Storage Plug-in for Containers によって作成された PersistentVolume 情報の確認

```
# oc get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAI
M POLICY   STATUS   CLAIM                STORAGECLASS   REASON   AGE
pvc-36c13883-b920-4ef8-af15-c388723ddb13   1Gi        RWO            Delete
           Bound    default/pvc-sample   sc-sample               45s

# oc describe pv pvc-36c13883-b920-4ef8-af15-c388723ddb13
Name:            pvc-36c13883-b920-4ef8-af15-c388723ddb13
Labels:          <none>
Annotations:     pv.kubernetes.io/provisioned-by: nspc.csi.nec.com
                 volume.kubernetes.io/provisioner-deletion-secret-name: sec
ret-sample
                 volume.kubernetes.io/provisioner-deletion-secret-namespace
: default
Finalizers:      [kubernetes.io/pv-protection external-attacher/nspc-csi-ne
c-com]
StorageClass:    sc-sample
Status:          Bound
Claim:           default/pvc-sample
Reclaim Policy:  Delete
Access Modes:    RWO
VolumeMode:      Filesystem
Capacity:        1Gi
Node Affinity:   <none>
Message:
Source:
    Type:              CSI (a Container Storage Interface (CSI) volume sour
ce)
    Driver:            nspc.csi.nec.com
    FSType:            ext4
    VolumeHandle:      01--scsi--934000600110--44--spc-702526a5dd
    ReadOnly:          false
    VolumeAttributes:      connectionType=iscsi
                           hostModeOption=
                           ldevIDDec=44
                           ldevIDHex=00:2C
                           nickname=spc-702526a5dd
                           portIPs=
                           ports=CL1-A,CL2-A
                           size=1Gi
                           storage.kubernetes.io/csiProvisionerIdentity=168
5678634724-8081-nspc.csi.nec.com
Events:                <none>
```

Secret、StorageClass、PersistentVolumeClaim、および Pod の削除

```
# oc get pod
NAME         READY   STATUS    RESTARTS   AGE
pod-sample   1/1     Running   0          30s
```

```
# oc delete pod pod-sample
pod "pod-sample" deleted

# oc get pvc
NAME         STATUS   VOLUME                                     CAPACITY
 ACCESS MODES   STORAGECLASS   AGE
pvc-sample   Bound    pvc-cf8c6089-0386-4c39-8037-e1520a986a7d   1Gi
 RWO            sc-sample      46s

# oc delete pvc pvc-sample
persistentvolumeclaim "pvc-sample" deleted

# oc get sc
NAME        PROVISIONER         AGE
sc-sample   nspc.csi.nec.com   53s

# oc delete sc sc-sample
storageclass.storage.k8s.io "sc-sample" deleted

# oc get secret
NAME            TYPE     DATA   AGE
secret-sample   Opaque   3      74s

# oc delete secret secret-sample
secret "secret-sample" deleted
```

3.6 Raw Block Volume

OpenShift は、ファイルシステムに加え、Raw Block Volume もサポートします。ここでは Raw Block Volume を適用する方法について説明します。

前提条件

この機能には StorageClass が必要です。

pvc-sample-block.yaml のパラメーターリファレンス

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-sample-block          #(1)
spec:
  accessModes:
  - ReadWriteOnce                 #(2)
  volumeMode: Block
  resources:
    requests:
      storage: 1Gi                #(3)
  storageClassName: sc-sample     #(4)
```

凡例：

(1) PersistentVolumeClaim 名

(2) `ReadWriteOnce` または `ReadWriteMany` を指定します。

(3) ボリュームサイズ

(4) StorageClass 名

pod-sample-block.yaml のパラメーターリファレンス

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-sample-block                #(1)
spec:
  containers:
    - name: my-busybox
      image: busybox
      volumeDevices:
      - devicePath: "/block"            #(2)
        name: sample-volume
      command: ["sleep", "1000000"]
      imagePullPolicy: IfNotPresent
  volumes:
    - name: sample-volume
      persistentVolumeClaim:
        claimName: pvc-sample-block     #(3)
```

凡例：

(1) Pod 名

(2) パス（ボリュームがコンテナ内にマウントされているパス）

(3) PersistentVolumeClaim 名

コマンド例

- raw ブロックボリュームの PersistentVolumeClaim を作成します。

```
# oc create -f pvc-sample-block.yaml
```

- raw ブロックボリュームの Pod を作成します。

```
# oc create -f pod-sample-block.yaml
```

3.7 ReadWriteMany

ご利用の OpenShift クラスター内の一つまた は複数のノードにボリュームをマウントして、読み取り/書き込みのオペレーションを実行できます。

メモ

OpenShift Virtualization は ReadWriteMany（複数読み取り/書き込み）アクセスモードをサポートしています。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-sample-block    #(1)
spec:
  accessModes:
  - ReadWriteMany    #(2)
  volumeMode: Block
  resources:
    requests:
      storage: 7Gi    #(3)
  storageClassName: sc-sample    #(4)
```

凡例:

（1）PersistentVolumeClaim 名

（2） `ReadWriteMany` を指定します。

（3）ボリュームの容量

（4）StorageClass 名

3.8　ReadOnlyMany

ご利用の OpenShift クラスターの 1 つまたは多数のノードに読み込み専用のボリュームとしてマウントできます。

ReadOnlyMany の PersistentVolumeClaim を作成するには、既存の PVC から PersistentVolumeClaim を作成する必要があります。

「3.10.5　PVC のクローン作成（31 ページ）」を参照して、PersistentVolumeClaim のマニフェストファイルに ReadOnlyMany を指定します。次に指定例を示します。

メモ

この機能は iStorage V110、V310/V310F ではサポートしません。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-rox-sample
spec:
  dataSource:
    name: pvc-sample
    kind: PersistentVolumeClaim
    apiGroup: ""
  accessModes:
  - ReadOnlyMany # Specify "ReadOnlyMany" here.
```

```
  resources:
    requests:
      storage: 1Gi
  storageClassName: sc-sample
```

3.9　ストレージリソースの分割

この機能では、ストレージシステムのリソースを OpenShift クラスターごとに分割できます。

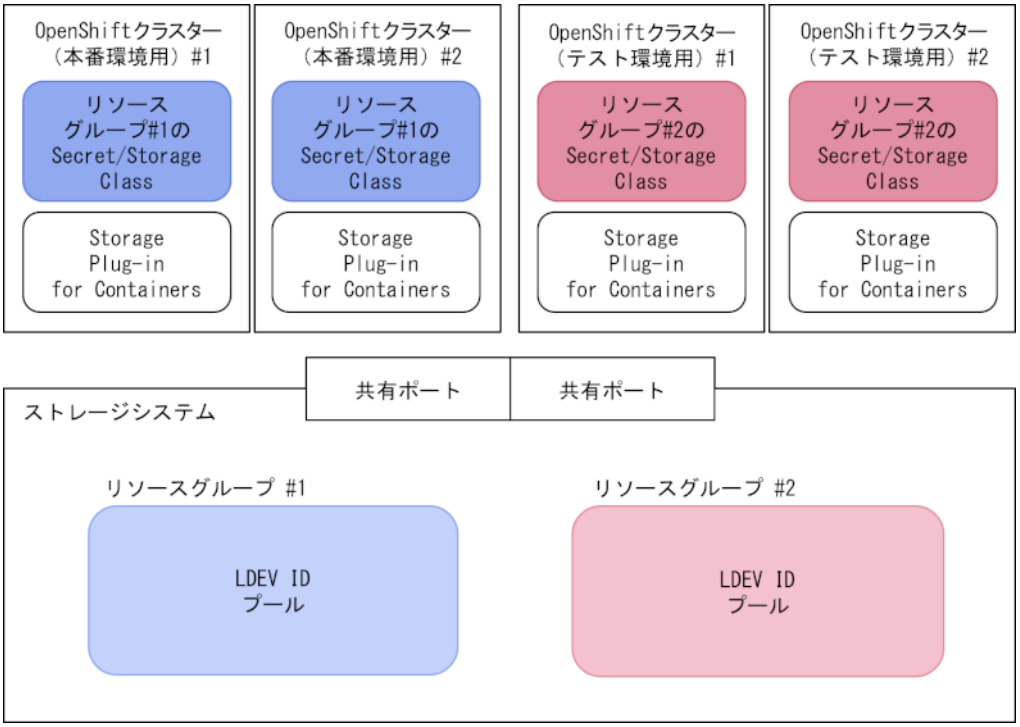
リソース分割の例を次に示します。

- 特定の OpenShift クラスターのリソースグループに追加する LDEV ID の範囲を制限できます。
- OpenShift クラスター間の影響を分離できます。

リソース分割機能を使用する前に、ストレージシステムの設定、Secret および StorageClass の設定が必要です。

サポートする構成

ストレージリソースを分割する構成の例を次に示します。



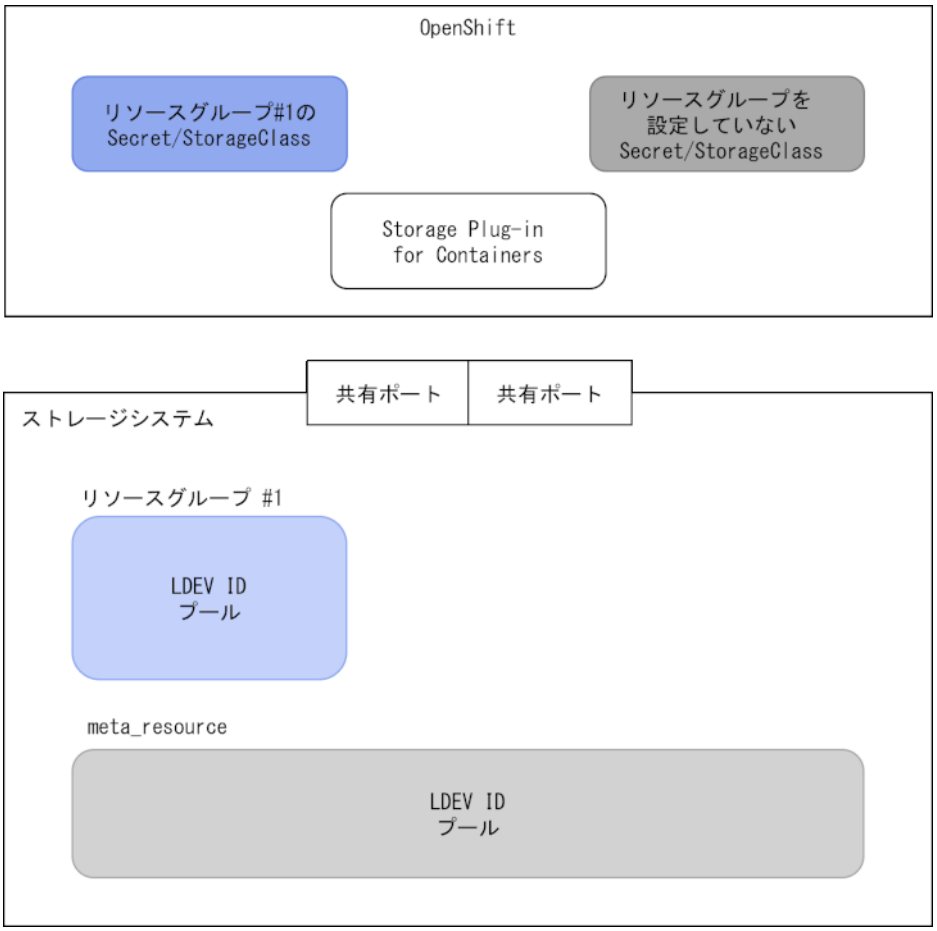
サポートしない構成

次のような構成はサポートしません。

例 1

同じ OpenShift クラスターに次の構成を両方とも含めることはできません。

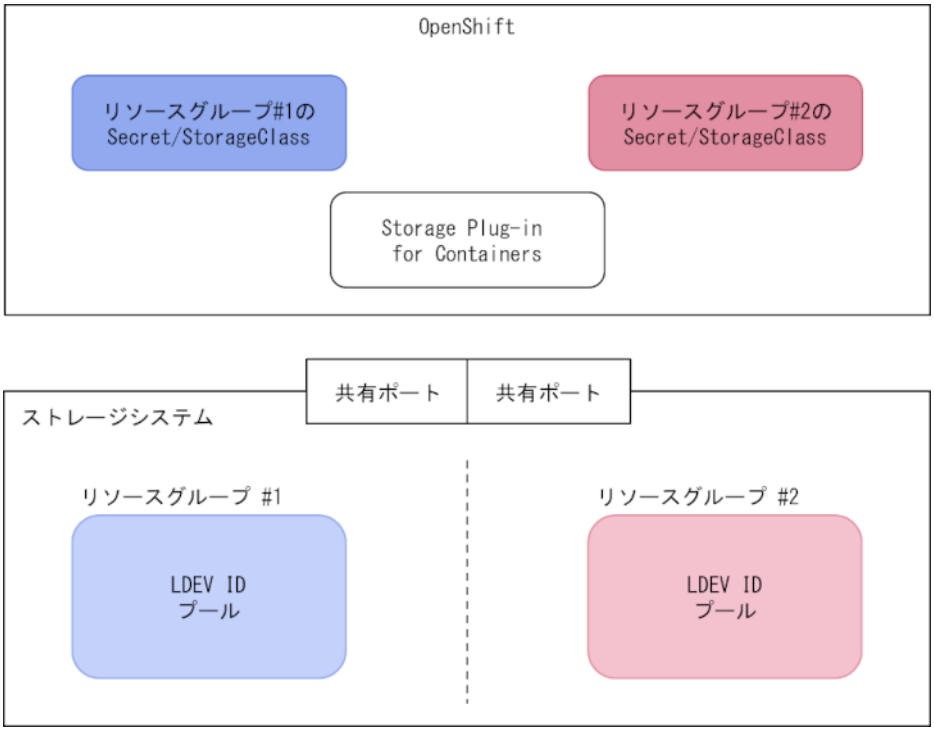
- リソースグループ用に構成された StorageClass および Secret
- メタリソースで使用するために一般的に構成された StorageClass および Secret



例 2

1 つのストレージシステムに対し複数のリソースグループが構成されている場合、それぞれのリソースグループを同じ OpenShift クラスターのリソースグループに対応させることはできません。

OpenShift クラスターには、ストレージシステムごとに 1 つのリソースグループ（storageClass および Secret を含む）だけを構成できます。



ストレージシステムの要件、設定

次の要件に合うようストレージシステムの設定を行ってください。

ストレージリソース	説明
リソースグループ	1 つの OpenShift クラスターに対して、複数のリソースグループを使用することはできません。仮想ストレージマシンはサポートしていません。
ストレージシステムのユーザーグループおよびユーザー	ストレージシステムのユーザーには、作成したリソースグループに対するアクセス権限だけを付与し、ほかのリソースグループにはアクセスできないようにしてください。
プール	作成したリソースグループ内のプールボリュームからプールを作成してください。
LDEV	必要な数の未使用の LDEV ID をリソースグループに割り当ててください。 容量削減機能を有効にすると、重複排除用システムデータボリュームが作成されます。このボリュームの割り当てに必要な LDEV ID を各リソースグループに対して登録します。登録が必要な LDEV ID の数については、ストレージシステムのマニュアルを参照してください。
ホストグループ	StorageClass に定義したストレージシステムのポート 1 つにつき、ホスト数と同じ数のホストグループ ID を準備してください。例えば、ホスト数が 3 でポート数が 2 の場合、計 6 個のホストグループ ID が必要です。ストレージシステム のポートごとに、準備したホストグループ ID をリソースグループに割り当て てください。

Secret の設定

ストレージシステムのリソースグループ ID を指定します。

Secret の設定例：

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-sample
type: Opaque
data:
  url: aHR0cDovLzE3Mi4xNi4xLjE=
  user: VXNlcjAx
  password: UGFzc3dvcmQwMQ==
stringData:
  resourceGroupID: "1"    # Specify resource group ID
```

StorageClass の設定

ストレージシステムのポートの IP アドレスを任意の順に指定します。

StorageClass の設定例：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-sample
provisioner: nspc.csi.nec.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:
  serialNumber: "600001"
  poolID: "1"
  portID : CL1-A,CL2-A
  connectionType: iscsi
  portIP: "192.168.10.10, 192.168.10.11"    # Specify iSCSI Port IP Address
es.
<...>
```

3.10 Static provisioning

ストレージシステムの既存ボリュームを、コンテナオーケストレーターで PVC として利用できるようにする機能です。この機能を利用すると、既存ボリュームに対し、Storage Plug-in for Containers で動的にプロビジョニングした PVC と同様に操作できます。

3.10.1 Static provisioning を利用する場合の要件

Static provisioning を利用する場合の要件を示します。

ボリュームが次の要件を満たしていることを確認してください。

- DP 属性の LDEV であること。

  iStorage V110、V310/V310F の場合、DRS 属性の LDEV であること。

  ヒント

  DRS 属性はデータ削減共有ボリュームを示します。

- ニックネームの形式が spc-<*16 進数で 10 けたの数値*>であること。

  <*16 進数で 10 けたの数値*>は、LDEV ごとにユニークな値である必要があります。

  ニックネームがほかの LDEV と重複した場合は、Storage Plug-in for Containers でサポートされている機能が正常に動作しないおそれがあります。

  ニックネーム用のユニークな文字列は、コマンドでも生成できます。コマンドの例を次に示します。

```
# echo spc-$(cat /dev/urandom | tr -dc a-f0-9 | head -c 10)
```

- LDEV がポートにマッピングされていないこと。
- ペアが形成されていないこと。

その他の要件については、1.3　要件（2 ページ）を参照してください。

LDEV が特定のリソースグループに割り当てられている場合は、3.9　ストレージリソースの分割（22 ページ）にあるストレージシステムの要件も参照してください。

3.10.2　Secret および StorageClass を作成する

PV および PVC を作成する際に指定する Secret および StorageClass を作成します。

Secret および StorageClass の YAML ファイルの設定については、3.1　Secret の設定（12 ページ）および3.2　StorageClass の設定（13 ページ）を参照してください。ボリュームが特定のリソースグループに割り当てられている場合は、3.9　ストレージリソースの分割（22 ページ）にある Secret の設定および StorageClass の設定も参照してください。

StorageClass に指定するパラメーターは、対象のボリュームの状態に合わせた値を指定してください。もし、パラメーターに指定した値が実際のボリュームの状態と不一致である場合、Storage Plug-in for Containers でサポートされている機能が正常に動作しないおそれがあります。

3.10.3　PV を作成する

PVC に関連づけるための PV を作成します。

メモ

- 1 つのボリュームに対して PV は 1 つだけ作成できます。

1 つのボリュームに対して複数の PV を作成した場合、Storage Plug-in for Containers でサポートされている機能が正常に動作しないおそれがあります。

- PV のパラメーターの指定が誤っている場合、想定外のエラーメッセージが出力されるおそれがあります。

操作手順

1. YAML ファイルを作成します。

   PV に指定するパラメーターは、対象のボリュームの状態、および作成した StorageClass の設定に合わせた値を指定してください。もし、パラメーターに指定した値が実際のボリュームの状態と不一致である場合、Storage Plug-in for Containers でサポートされている機能が正常に動作しないおそれがあります。

   YAML ファイルの例：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: static-pv
  annotations:
    pv.kubernetes.io/provisioned-by: nspc.csi.nec.com
spec:
  persistentVolumeReclaimPolicy: Delete
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 1Gi
  volumeMode: Filesystem
  csi:
    fsType: ext4
    volumeAttributes:
      connectionType: fc
      ports: CL1-A, CL2-A
    volumeHandle: 01--scsi--934000070010--50000--spc-c3d46c5a71
    driver: nspc.csi.nec.com
    controllerExpandSecretRef:
      name: secret-sample
      namespace: default
    controllerPublishSecretRef:
      name: secret-sample
      namespace: default
  storageClassName: sc-sample
  claimRef:
    name: static-pvc
    namespace: default
```

   パラメーター：

パラメーター	説明	必須または任意
metadata.name	PV 名を指定します。	必須

パラメーター	説明	必須または任意
metadata.annotations.pv.kubernetes.io/provisioned-by	nspc.csi.nec.com を指定します。	persistentVolumeReclaimPolicy に Delete を指定した場合は必須です。
spec.persistentVolumeReclaimPolicy	再クレームポリシーを指定します。 Retain を指定した場合、PVC を削除しても PV および LDEV は削除されません。このとき、PV は released 状態のため、LDEV を再利用したい場合は PV を再作成する必要があります。詳細については、https://kubernetes.io/docs/concepts/storage/persistent-volumes/#retain を参照してください。 Delete を指定した場合、PVC を削除すると、PV および LDEV は削除されます。 デフォルト値は Retain です。	任意
spec.accessModes	アクセスモードを指定します。 サポートしているアクセスモードについては、3.3　PersistentVolumeClaim の設定（14 ページ）を参照してください。	必須
spec.capacity.storage	LDEV サイズを指定します。	必須
spec.volumeMode	Filesystem または Block を指定します。 対象の LDEV を raw ブロックボリュームとして使用していた場合は、必ず Block を指定してください。もし、LDEV を raw ブロックボリュームで使用していたにもかかわらず、Filesystem を指定すると、ファイルシステムでフォーマットされ、既存のデータが消去されます。 デフォルト値は Filesystem です。	任意
spec.csi.fsType	対象の LDEV のファイルシステムタイプを指定します。 デフォルト値は ext4 です。 volumeMode に Block を指定した場合、このパラメーターは無効になります。 サポートしているファイルシステムタイプについては、3.2　StorageClass の設定（13 ページ）を参照してください。	任意
spec.csi.volumeAttributes.connectionType	ストレージシステムとノード間の接続タイプを指定します。 サポートしている接続タイプについては、3.2　StorageClass の設	必須

パラメーター	説明	必須または任意
	定（13 ページ）を参照してください。	
spec.csi.volumeAttributes.ports	ストレージポート ID を指定します。 マルチパス構成の場合は、ストレージポート ID をコンマで区切ってください。	必須
spec.csi.volumeAttributes.portIPs	ストレージポート IP アドレスを指定します。 マルチパス構成の場合は、ストレージポート IP アドレスをコンマで区切ってください。	connectionType に iscsi を指定し、LDEV が特定のリソースグループに割り当てられている場合は必須です。
spec.csi.volumeHandle	次のフォーマットで指定します。 01--<*IO プロトコル*>--<*ストレージデバイス ID*>--<*LDEV ID*>--<*LDEV ニックネーム*> ***<IO プロトコル>*** scsi を指定します。 ***<ストレージデバイス ID>*** 各ストレージモデルの『REST API リファレンスガイド』を参照して確認してください。ストレージデバイス ID は 12 けたの値で、形式は次のとおりです。 *<ストレージモデルごとの固定値（6 けた）><シリアル番号（6 けた）>* 例えば、 iStorage V100 の場合、固定値は 934000 です。 ***<LDEV ID>*** ストレージシステムの管理ソフトウェアを使用して確認してください。 値は 10 進数で指定してください。 ***<LDEV ニックネーム>*** ストレージシステムの管理ソフトウェアを使用して確認してください。	必須
spec.csi.driver	nspc.csi.nec.com を指定します。	必須
spec.csi.controllerExpandSecretRef.name	Secret 名を指定します。	必須
spec.csi.controllerExpandSecretRef.namespace	Secret の namespace を指定します。	必須
spec.csi.controllerPublishSecretRef.name	Secret 名を指定します。	必須

パラメーター	説明	必須または任意
spec.csi.controllerPublishSecretRef.namespace	Secret の namespace を指定します。	必須
spec.storageClassName	StorageClass 名を指定します。	必須
spec.claimRef.name	3.10.4　PVC を作成する（30 ページ）で作成する PVC 名を指定します。	必須 このパラメーターを指定しないと、PV が意図しない PVC に関連づけられ、Storage Plug-in for Containers の機能が正常に動作しないおそれがあります。
spec.claimRef.namespace	3.10.4　PVC を作成する（30 ページ）で作成する PVC の namespace を指定します。	必須

2. YAML ファイルをデプロイします。

```
# oc apply -f <YAML ファイル名>
```

3.10.4　PVC を作成する

PVC を作成し、Storage Plug-in for Containers の機能が使用できる状態にします。

操作手順

1. YAML ファイルを作成します。

   YAML ファイルの例：

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: static-pvc
  namespace: default
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: sc-sample
  volumeMode: Filesystem
  volumeName: static-pv
```

   パラメーター：

パラメーター	説明	必須または任意
metadata.name	PVC 名を指定します。	必須
metadata.namespace	PVC の namespace を指定します。	任意
spec.accessModes	PV の accessModes と同じ値を指定します。	必須

パラメーター	説明	必須または任意
spec.resources.requests.storage	PV の capacity.storage と同じ値を指定します。	必須
spec.storageClassName	PV の storageClassName と同じ値を指定します。	必須
spec.volumeName	PV 名を指定します。	必須 このパラメーターを指定しないと、PVC が意図しない PV に関連づけられ、Storage Plug-in for Containers の機能が正常に動作しないおそれがあります。
spec.volumeMode	PV で volumeMode を指定した場合は、PV と同じ値を指定します。 デフォルト値は Filesystem です。	PV の volumeMode に Block を指定した場合は必須です。

2. YAML ファイルをデプロイします。

```
# oc apply -f <YAML ファイル名>
```

3. PVC の STATUS が Bound であることを確認します。

   PVC が Bound 状態になれば、Storage Plug-in for Containers がサポートしている機能が利用できるようになります。

3.10.5　PVC のクローン作成

この機能では、既存ボリュームのクローンとして複製を作成できます。クローンは、通常のボリュームと同様に使用できます。

メモ

- クローンの操作対象のボリュームが拡張されている場合は、拡張が完了したことを確認してからこの機能を実行してください。詳細については、「3.10.7　PVC のストレージボリュームの拡張（37 ページ）」を参照してください。
- データの整合性を確保するため、クローン作成前にデータをフラッシュしてください。例えば、一時的に Pod を削除してください。

前提条件

この機能には、次のリソースが必要です。

- StorageClass
- PersistentVolumeClaim

pvc-from-pvc-sample.yaml のパラメーターリファレンス

この YAML ファイルは既存のボリューム"pvc-sample"からクローンを作成するためのマニフェストファイルです。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-pvc-sample      #(1)
spec:
  dataSource:
    name: pvc-sample             #(2)
    kind: PersistentVolumeClaim
    apiGroup: ""
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi               #(3)
  storageClassName: sc-sample    #(4)
```

凡例：

(1) クローンの PersistentVolumeClaim 名

(2) ソースの PersistentVolumeClaim 名

(3) ソースボリュームのサイズを指定します。`oc get pv <PV-name> -o yaml` コマンドを使用してサイズを取得します。サイズは、**size** に表示されます。

メモ

ボリュームが拡張されている場合、または Static Provisioning で作成された PersistentVolume の場合は、`oc get pv <PV-name>`コマンドを使用してサイズを取得します。サイズは、**CAPACITY** パラメータに表示されます。

(4) dataSource に使用したのと同じ StorageClass 名を指定します。

コマンド例

- クローン用の PersistentVolumeClaim を作成します。

```
# oc create -f pvc-from-pvc-sample.yaml
```

- クローンから Pod を作成する：

```
example yaml file

kind: Pod

apiVersion: v1

metadata:
```

```
  name: pod-clone-pvc

  namespace: <name of the namespace>

spec:

  containers:  # Must be a list

    - name: my-busybox  # Add "-" to indicate a list item

      image: busybox

      volumeMounts:

        - mountPath: "/data"  #(2)

          name: sample-volume

      command: ["sleep", "1000000"]

      imagePullPolicy: IfNotPresent

  volumes:  # Must be a list

    - name: sample-volume

      persistentVolumeClaim:

        claimName: pvc-from-pvc-sample
```

3.10.6 PVC のスナップショット作成

この機能では、ボリュームのポイントインタイムイメージであるスナップショットを作成できます。スナップショットは、既存ボリュームの以前の状態を復元または複製するために使用できます。

メモ

- iStorage V110、V310/V310F ストレージシステムでは、スナップショットからクローンが作成されると、スナップショットは PVC に変換されます。
- ボリュームが拡張された場合は、ボリュームスナップショットを作成する前に、オペレーションが正常に完了したことを確認してください。詳細については、「3.10.7 PVC のストレージボリュームの拡張（37 ページ）」を参照してください。
- データの整合性を保つため、スナップショットを作成する前に、スナップショットを作成する前にデータをフラッシュしてください。例えば、一時的に Pod を削除してください。

前提条件

この機能には、次のリソースが必要です。

- StorageClass

- PersistentVolumeClaim

volumesnapshotclass-sample.yaml のパラメーターリファレンス

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: snapshotclass-sample    #(1)
driver: nspc.csi.nec.com
deletionPolicy: Delete
parameters:
  poolID: "1"                   #(2)
  retentionPeriod: "1"              #(3)
  csi.storage.k8s.io/snapshotter-secret-name: "secret-sample" #(4)
  csi.storage.k8s.io/snapshotter-secret-namespace: "default"  #(5)
```

凡例：

(1) VolumeSnapshotClass 名

(2) StorageClass と同じ poolID

(3)（オプション）保持期間：1～12,288 時間の保持期間を指定し、その間変更できないスナップショットを作成します。

スナップショットデータの保持期間が設定されると、指定された期間が終了するまで、スナップショットデータは更新から保護されます。保持期間が終了すると、ペアのスナップショットデータ保持設定は自動的に無効になります。

メモ

- `retentionPeriod` が `VolumeSnapshotClass` で指定されている場合、定義された保持期間中はスナップショットを削除またはクローン作成することはできません。
- イミュータブルスナップショットは現在、iStorage V110, V310/V310F ストレージシステムでのみサポートされています。

(4) StorageClass と同じ Secret 名

(5) StorageClass と同じ Secret namespace

volumesnapshot-sample.yaml のパラメーターリファレンス

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: snapshot-sample                                  #(1)
spec:
  volumeSnapshotClassName: snapshotclass-sample          #(2)
  source:
    persistentVolumeClaimName: pvc-sample                #(3)
```

凡例：

(1) VolumeSnapshot 名

(2) VolumeSnapshotClass 名

(3) スナップショットの取得対象の PersistentVolumeClaim 名

pvc-from-snapshot-sample.yaml のパラメーターリファレンス

メモ

スナップショットからの Persistent Volume の作成は、iStorage V110, V310/V310F ではサポートされていません。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snapshot-sample    #(1)
spec:
  dataSource:
    name: snapshot-sample           #(2)
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi                  #(3)
  storageClassName: sc-sample       #(4)
```

凡例：

(1) PersistentVolumeClaim 名

(2) VolumeSnapshot 名

(3) ソースボリュームのサイズを指定します。`oc get pv <PV-name> -o yaml` コマンドを使用してサイズを取得します。サイズは、**size** に表示されます。

メモ

ボリューム拡張済みのボリュームの場合、または StaticProvisioning で作成された PV（PersistentVolume）の場合は、`oc get pv <PV-name>`コマンドを使用してサイズを取得します。サイズは、**CAPACITY** で表示されます。

(4) dataSource に使用したのと同じ StorageClass 名を指定します。

コマンド例

- VolumeSnapshotClass を作成します。

```
# oc create -f volumesnapshotclass-sample.yaml
```

- VolumeSnapshot を作成します。

```
# oc create -f volumesnapshot-sample.yaml
```

- readyToUse が true であることを確認します。true の場合、VolumeSnapshot の作成が完了しています。

```
# oc get volumesnapshot -o yaml
```

メモ

readyToUse が false の場合は、次のステップを実行して原因と解決方法を確認してください。

1. 次のコマンドを使用して、boundVolumeSnapshotContentName を取得します： oc get volumesnapshot -o yaml
2. 次のコマンドを使用して、エラーメッセージを確認します：**oc describe volumesnapshotcontent <VolumeSnapshotContentName>**

- スナップショットから PersistentVolumeClaim を作成します。

```
# oc create -f pvc-from-snapshot-sample.yaml
```

メモ

iStorage V110、V310/V310F ストレージシステムでは、スナップショットから作成できる PVC は 1 つだけです。PVC が作成されると、スナップショットは無効になり、再利用できなくなります。

- Pod を作成します：

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-sample            #(1)
spec:
  containers:
    - name: my-busybox
      image: busybox
      volumeMounts:
      - mountPath: "/data"    #(2)
        name: sample-volume
      command: ["sleep", "1000000"]
```

```
    imagePullPolicy: IfNotPresent
  volumes:
    - name: sample-volume
      persistentVolumeClaim:
        claimName: pvc-from-snapshot-sample  #(3)
```

3.10.7　PVC のストレージボリュームの拡張

この機能では、既存ボリュームの容量を拡張できます。ボリュームを拡張するために Pod を削除して作成し直す必要はありません。

⚠ 注意

`oc get pvc` コマンドで、**CAPACITY** に表示されるボリューム拡張の完了を確認してください。ボリュームの拡張が完了する前に、OS をシャットダウンしたりノードの drain 操作をしたりしないでください。

前提条件

この機能には次のリソースが必要です。

- StorageClass
- PersistentVolumeClaim

メモ

ボリュームの拡張には、次の制限があります。

- iStorage V110、V310/V310F ストレージシステムでは、クローンされた PVC を親ボリュームより大きい容量に拡張することはできません。
- ボリューム拡張の最低追加サイズは、1 GiB です。
- ボリューム拡張の最大追加サイズは、7 TiB またはプール容量の警告しきい値を超えない値です。7 TiB を超えるサイズを追加する場合は、コマンドをもう一度実行してください。
- ボリューム容量を縮小することはできません。
- StorageClass の `allowVolumeExpansion` が `false` に設定されている場合、この設定で作成された PersistentVolume は拡張できません。
- `oc get pv <PV-name> -o yaml` コマンドで取得されたサイズは、ボリュームの拡張後に更新されません。ボリュームが拡張された場合は、`oc get pv <PV-name>`コマンドを使用してサイズを取得してください。サイズは **CAPACITY** に表示されます。

コマンド例

- 既存ボリューム pvc-sample の容量を 5GiB に拡張します。

```
# oc patch pvc pvc-sample --patch \
'{"spec":{"resources":{"requests":{"storage": "5Gi"}}}}'
```

- CAPACITY でボリュームの拡張が完了したことを確認します。

```
# oc get pv <PV-name>
NAME            CAPACITY     ACCESS MODES      RECLAIM POLICY
STATUS CLAIM                 STORAGECLASS      REASON AGE
<PV-name>       5Gi          RWO               Delete
Bound default/pvc-sample     sc-sample         35s
```

⚠ 注意

- Storage Plug-in for Containers で作成された LDEV のサイズを変更する場合、ストレージシステムの管理ソフトウェアで変更せずに、oc コマンドで変更してください。
- ボリューム拡張のためのパッチを適用すると、ボリューム容量が増加します。ただし、PVC が Pod に接続されていない場合、PVC の容量は更新されない可能性があります。ノード内からファイルシステムを拡張する必要があります。

3.10.8 Storage Plug-in for Containers によって作成されたリソースの削除

Storage Plug-in for Containers によって作成されたリソースを削除できます。

Pod を削除するには、次のように入力します：

```
oc delete pod <pod-name>
```

PVC を削除するには、次のように入力します：

```
oc delete pvc <pvc-name>
```

メモ

iStorage V110, V310/V310F ストレージシステムでは、PVC がクローンされている場合、親 PVC を削除する前にクローンされた PVC を最初に削除する必要があります。

スナップショットを削除するには、次のように入力します：

```
oc delete vs <snapshot-name>
```

3.10.9　Static provisioning 利用時のトラブルシューティング

Pod の削除後に VolumeAttachment が残る

PV の YAML ファイルのパラメーター形式に不正がある場合、またはパラメーターに指定した値が不正である場合、Pod の作成に失敗します。この Pod を削除すると、VolumeAttachment が残ることがあります。

Pod の作成に失敗する例を次に示します。

- `volumeHandle` のフォーマット不正（エラーコード：`0x0000c002`）
- LDEV に対する権限不足（エラーコード：`0x00001007`）
- `controllerPublishSecretRef` が未指定（エラーコード：`0x0000c00f`）

VolumeAttachment を削除するには、次のコマンドを実行してください。

```
# oc patch volumeattachments <VolumeAttachment 名> --type merge -p '{"metada
ta":{"finalizers":null}}'
```

PVC の削除後に PV が残る

PV の YAML ファイルのパラメーター形式に不正がある場合、またはパラメーターに指定した値が不正である場合、`persistentVolumeReclaimPolicy` に `Delete` を指定していても、PVC の削除後に PV が残ることがあります。

PVC の削除後に PV が残る例を次に示します。

- `volumeHandle` のフォーマット不正（エラーコード：`0x0000c002`）
- LDEV に対する権限不足（エラーコード：`0x00001007`）

PV を削除するには、次の手順を実行してください。

1. ボリュームの削除が必要な場合、`volumeHandle` の情報を利用して対象のボリュームを確認します。

   ```
   # oc get pv <PV 名> -o yaml
   ```

2. PV を削除します。

   ```
   # oc delete pv <PV 名>
   ```

3. ボリュームの削除が必要な場合、ストレージシステムの管理ソフトウェアを使用して削除します。

PVC の削除後にボリュームが残る

`volumeHandle` の<*LDEV-ID*>で指定した LDEV に対して、<*LDEV-nickname*>の値に不正がある場合、`persistentVolumeReclaimPolicy` に `Delete` を指定すると、PVC の削除後に PV は削除されますが、ボリュームは削除されません。

ボリュームを削除するには、ストレージシステムの管理ソフトウェアを使用して削除してください。

第 4 章
アップグレード

ここでは、Storage Plug-in for Containers をアップグレードする方法について説明します。

メモ

- OpenShift の古いバージョンを使用している場合、最新の Storage Plug-in for Containers が動作しないおそれがあります。最新の Storage Plug-in for Containers を使用したい場合は、OpenShift のアップグレードを先に実施してください。
- Storage Plug-in for Containers をアップグレードする前に、2.2　Storage Plug-in for Containers インスタンスの設定（9 ページ）で設定した内容をバックアップしてください。バックアップした設定内容は、新しいバージョンの Storage Plug-in for Containers でも使用できる可能性があります。設定時に使用した `nspc_v1_nspc.yaml` が残っていない場合は、`oc get nspc -A -o yaml` を実行し設定内容を取得してください。

4.1　OpenShift でのアップグレード

OpenShift の Web コンソールを使用して Storage Plug-in for Containers をアップグレードできます。

操作手順

1. Operator Details の Storage Plug-in for Containers タブで NSPC を削除します。
2. Storage Plug-in for Containers の Operator をアンインストールします。
3. 新しい Storage Plug-in for Containers をインストールします。「2.1　OpenShift へのインストール（8 ページ）」を参照してください。

第 5 章
再作成

ここでは、Storage Plug-in for Containers を再作成する方法について説明します。Storage Plug-in for Containers を再作成するには、Storage Plug-in for Containers を削除し、作成し直してください。

5.1　Storage Plug-in for Containers を削除する

OpenShift の Web コンソールにアクセスし、Storage Plug-in for Containers のインスタンスを削除します。

5.2　Storage Plug-in for Containers を作成する

OpenShift の Web コンソールにアクセスし、Storage Plug-in for Containers のインスタンスを作成します。

詳細は、2.1　OpenShift へのインストール（8 ページ）を参照してください。

第 6 章
アンインストール

ここでは、Storage Plug-in for Containers をアンインストールする方法について説明します。この手順には、PersistentVolumeClaim、PersistentVolume、StorageClass、Storage Plug-in for Containers、およびその他の要素の削除が含まれます。

6.1　OpenShift でのアンインストール

OpenShift の Web コンソールを使用して Storage Plug-in for Containers をアンインストールできます。

操作手順

1. Storage Plug-in for Containers によって作成された PersistentVolumeClaim を使用しているすべての Pod を削除します。
2. Storage Plug-in for Containers に関連する、作成した VolumeSnapshotClass、VolumeSnapshot、PersistentVolumeClaim、StorageClass、Secret などをすべて削除します。
3. Operator Details の Storage Plug-in for Containers タブで、**NSPC** を削除します。
4. Storage Plug-in for Containers の Operator をアンインストールします。

第 7 章
トラブルシューティング

ここでは、障害発生時に取得する情報、およびエラーが発生するケースと対処方法について説明します。

7.1　障害発生時に収集する情報

Storage Plug-in for Containers で障害が発生した場合は、次の情報を採取してください。採取した情報は、お問い合わせの際に障害対応窓口にお渡しください。

7.1.1　障害対応窓口への連絡時に必要な情報

障害対応窓口に連絡する場合、Storage Plug-in for Containers およびストレージシステムの情報を収集してください。詳細については次の表を参照してください。

項目	収集手順
コマンドの実行ログ	実行したコマンドと、そのコマンドの実行結果を取得してください。
操作対象リソースの `oc describe` コマンドの実行結果	操作を実施したリソースに対して、次のコマンドを実行してください。 `# oc describe <resource> -n <Storage Plug-in for Containers の namespace> <resource-name>`
クラスターの情報	次のコマンドを実行してください。 `# oc cluster-info dump -A > dump.txt`
Pod の情報	7.1.2　Storage Plug-in for Containers 情報の収集（45 ページ）の手順 1 で実行したコマンドと、そのコマンドの実行結果を取得してください。
Operator のログ	「7.1.2　Storage Plug-in for Containers 情報の収集（45 ページ）」を参照してください。
CSI コントローラーのログ	「7.1.2　Storage Plug-in for Containers 情報の収集（45 ページ）」を参照してください。
CSI ノードのログ	「7.1.2　Storage Plug-in for Containers 情報の収集（45 ページ）」を参照してください。
PVC 関連のマニフェスト	StorageClass、Secret、および PersistentVolumeClaim の YAML ファイルを取得してください。
Snapshot 関連のマニフェスト	VolumeSnapshotClass、Secret、および VolumeSnapshot の YAML ファイルを取得してください。
Snapshot 関連のログ	「3.10.6　PVC のスナップショット作成（33 ページ）」でインストールした Snapshot Controller のログを取得してください。

項目	収集手順
アプリケーションのマニフェスト	Storage Plug-in for Containers の PVC を使用しているアプリケーションの YAML ファイルを取得してください。
ストレージシステムのログ	「7.1.3　ストレージシステム情報の収集（46 ページ）」を参照してください。

7.1.2　Storage Plug-in for Containers 情報の収集

`oc logs` コマンドを使用することで実行中のコンテナのログを取得できます。Storage Plug-in for Containers のログを収集するには、Operator、CSI コントローラー、CSI ノードのログを収集する必要があります。

メモ

必要に応じてクラスターのログレベルを設定してください。

https://kubernetes.io/docs/concepts/cluster-administration/logging/

1. ログを取得する前に、次のコマンドを実行して Pod の名前を確認します。

```
# oc get pod -A -o wide
```

2. 次のコマンドを実行して、ログを取得します。

   - Operator

```
# oc logs -n <Storage Plug-in for Containers の namespace> nspc-operator-controller-manager-<id>
```

   - CSI コントローラー

```
# oc logs -n <Storage Plug-in for Containers の namespace> nspc-csi-controller-<id> -c nspc-csi-driver
```

メモ

`-c nspc-csi-driver` を必ず指定して、コマンドを実行してください。

   - CSI ノード

```
# oc logs -n <Storage Plug-in for Containers の namespace> nspc-csi-node-<id> -c nspc-csi-driver
```

メモ

- `-c nspc-csi-driver` を必ず指定して、コマンドを実行してください。
- CSI ノードは DaemonSet として展開されるため、複数の CSI ノードの Pod が表示されます。 すべての Pod のログを収集してください。

3. 取得したログは、古いログがローテーションされて削除されているおそれがあるため、次の手順でディレクトリーを取得します。

a. 対象ノードを確認します。

   手順 1 の実行結果から、NAME の値が nspc-csi-controller を含む名称になっている行を探し、同じ行の NODE の値を確認してください。

b. 手順 a で確認したノードの/var/log/pods 以下に格納されているディレクトリーを取得します。

7.1.3　ストレージシステム情報の収集

SVP を使用している場合は、通常ダンプファイルを採取してください。SVP を使用しない構成の場合は、Maintenance Utility からシステムダンプを採取してください。これらのストレージシステムのダンプファイルの採取方法については、『システム管理者ガイド』または『HA Device Manager - Storage Navigator ユーザガイド』を参照してください。

7.2　PersistentVolume の情報確認

Storage Plug-in for Containers でボリュームが動的に作成された場合、PersistentVolume の spec.csi.volumeAttributes に作成されたボリュームの情報が設定されます。oc get pv <*PV-name*> -o yaml を実行することで、この情報を確認できます。

設定されるボリュームの情報は内部的な用途で使用されますが、障害発生時に有用な情報も格納されます。次の表で格納される情報について説明します。

プロパティ	説明
ldevIDDec	10 進数の LDEV ID
ldevIDHex	16 進数の LDEV ID
size	ボリュームの容量 **メモ** このプロパティに表示される容量は、ボリューム作成時に使用した容量です。

7.3　Pod を強制削除する際の注意事項

特定のノードから Pod を強制削除した場合、削除した Pod と Pod に関連づく PVC の情報が該当するノードに残り、予期せぬエラーが発生するおそれがあります。

これらの情報を適切に削除するには、該当するノードを再度利用する前にノードの再起動を必ず実施してください。

7.4　PersistentVolumeClaim の多重作成、多重削除

PersistentVolumeClaim の作成と削除を同時に実行すると、ストレージシステムに負荷がかかり、0x0000100b、0x0000100f、0x0000101a、または 0x0000f007 のエラーが発生することがあります。この問題は、csi-provisioner コンテナに--worker-threads 引数を指定することで軽減できます。この引数によって、作成と削除の同時実行数は制限されます。デフォルト値は 100 です。

次の例に、--worker-threads の数値を 10 に減らす方法を示します。YAML の設定については、「2.2　Storage Plug-in for Containers インスタンスの設定（9 ページ）」を参照してください。

```
apiVersion: csi.nec.com/v1
kind: NSPC
metadata:
  name: nspc
  namespace: <SPC_NAMESPACE の値>
spec:
  imagePullSecrets:
  - regcred-redhat-com
  controller:
    containers:
      - name: csi-provisioner
        args:
        - --csi-address=/csi/csi-controller.sock
        - --timeout=300s
        - --v=5
        - --worker-threads=10
        - --default-fstype=ext4
```

問題が解決しない場合は、障害対応窓口に連絡してください。

7.5　ホストグループの設定

エラー 0x00001023 が発生した場合は、ストレージ内のホストグループを変更する必要があります。Storage Plug-in for Containers は、命名規則に基づいて"spc-<*wwn1*>-<*wwn2*>-<*wwn3*>"という名前のホストグループを検索します（1.4.2.1　iSCSI ターゲットの命名規則（7 ページ）を参照してください）。このエラーは、ホストグループの名前が"spc-<*wwn1*>-<*wwn2*>-<*wwn3*>"の命名形式に従っていないために発生した可能性があります。この問題を解決するには、エラーメッセージに表示されたホストグループを削除し、ホスト WWN を持つホストグループの名前を変更します。

1. Storage Plug-in for Containers を削除します。

   詳細は 5.1　Storage Plug-in for Containers を削除する（42 ページ）を参照してください。

2. エラーメッセージで指定されているホストグループを削除します。
3. 各ホストの WWN を持つホストグループを検索し、それらを削除するか、名前を"spc-<*wwn1*>-<*wwn2*>-<*wwn3*>"に変更します。

4. Storage Plug-in for Containers を作成します。

   詳細については、5.2　Storage Plug-in for Containers を作成する（42 ページ）を参照してください。

7.6　Pod 作成または Pod 削除時のタイムアウトエラー

ストレージシステムの状態やホスト OS の状態によっては、Storage Plug-in for Containers の処理が遅くなり、タイムアウトエラーが発生するおそれがあります。

Pod 作成や Pod 削除の際にタイムアウトエラーが続く場合、第 5 章　再作成（42 ページ）を参照して、Storage Plug-in for Containers のインスタンスを再作成してください。タイムアウトエラーが解消しない場合、タイムアウトエラーが発生している Pod が割り当てられているノードを再起動してください。ノードの再起動は、ほかの環境に影響を与えるおそれがあります。このため、影響範囲を十分に確認してから、OpenShift のドキュメントを参照し、手順を一つ一つ確認しながら慎重に再起動してください。

7.7　Openshift Pod のスケジュールまたは作成ができない

マルチパス対応のブロックデバイスで構成された PersistentVolumes を使用する場合、Openshift Pod をスケジュールまたは作成することはできません。

これは、マルチパスサービスまたはモジュールが正しく構成されていない場合、または Pod がスケジュールされているノードで実行されていない場合に発生する可能性があります。ストレージデバイス（通常は SAN ディスク）がコンテナランタイムによって検出されないか、利用できません。

問題を解決するには、次の手順を実行してください：

1. マルチパスサービスのステータスを確認します。:

   ```
   “systemctl status multipathd”
   ```

   `multipathd` デーモンがアクティブで実行中であるかどうかを確認します。デーモンが実行中の状態でない場合は、手順 2 に進みます。

2. マルチパスサービスを有効にして起動します。:

   ```
   “systemctl enable --now multipathd”
   ```

   このコマンドは `multipathd` サービスを直ちに開始し、ノードの起動時に確認します。

3. Device Mapper Multipath に必要なマルチパスカーネルモジュールをロードします。:

```
“modprobe dm-multipath”
```

4. 検出されたマルチパスデバイスを確認します。:

```
“multipath -ll”
```

メモ

- マルチパス設定を構成した後、ノードを再起動する必要があります。
- PersistentVolume によって使用されるストレージデバイスが一覧表示されていることを確認します。

付録 A. 既存の LUN 構成の編集

LUN 構成を更新するには、SPC_LUN_MAX パラメーターを変更します。デフォルトでは、256 個の LUN がサポートされています。SPC_LUN_MAX パラメーターには、最大 2048 までの値を指定することができます。

⚠ 警告

SPC_LUN_MAX パラメーターの値を 2048 より大きく設定しないでください。

メモ

SPC_LUN_MAX の環境値を変更する場合は、LUN を検出およびサポートするために、HBA ドライバに基づいた適切なカーネルレベルの構成も必要です。設定を確認するには、ベンダー固有のガイドを参照してください。

Storage Plug-in for ContainersCSI ドライバイメージをパッチするには、次の手順を実行します：

操作手順

1. Storage Plug-in for Containers がインストールされているクラスタに移動します。
2. コントローラノードにログオンします。
3. （オプション）Storage Plug-in for Containers が Operator 経由でデプロイされている場合、Opetrator のレプリカ数を 0 にスケールダウンします。:

```
oc scale deployment nspc-operator-controller-manager --replicas=0 -n <Storage-Plug-in-for-Containers-namespace>
```

4. Storage Plug-in for Containers の DaemonSet YAML ファイルを編集します。:

```
oc edit daemonset nspc-csi-node -n <Storage-Plug-in-for-Containers-namespace>
```

5. nspc-csi-driver **コンテナの** env セクションで、SPC_LUN_MAX パラメーター値を更新します。:

```
- env:

- name: SPC_LUN_MAX

value: “2048”
```

6. 変更を保存します。

操作結果

新しい構成が反映され、Pod が再デプロイされます。

付録 B. StorageClass の既存ポートの更新

StorageClass の既存ポートを更新することができます。StorageClass は一度作成すると、直接編集することができません。

前提条件

以下のシークレットキーが利用可能であることを確認します：

- portID
- iSCSI の場合：portIP

Stretched PVC の場合、以下の詳細が利用可能であることを確認します：

- primaryPortID
- secondaryPortID
- iSCSI 用：primaryPortIP および secondaryPortIP

StorageClass の既存ポートを更新するには、次のいずれかの手順を実行します。

- StorageClass と PersistentVolume の変更（51 ページ）
- Secret の変更（52 ページ）

メモ

Secret を使用してポートを設定する場合、次の制約があります：

- 永続ボリューム（PV）は作成後に変更することができません。一度作成されると、プロビジョニング時に定義された元のポート値が保持されます。PV にマッピングされたポートは、Secret で定義されたポートに対応しています。

StorageClass と PersistentVolume の変更

1. ポート移行が必要な Pod を特定します。
2. その Pod が使用している StorageClass、PV、PVC を特定します。
3. StorageClass からポート情報を取得します。
4. Pod を削除します。
5. 削除した Pod に関連付けられていた PV を編集し、persistentVolumeReclaimPolicy パラメータを変更します。
6. パラメータ値を Delete から Retain に更新します。
7. PV.yaml ファイルをバックアップします。
8. PVC を削除します。

9. PV を削除します。
10. StorageClass を削除します。

メモ

StorageClass の名前と構成パラメータをメモしておきます。

11. 必要なポートを含む新しい StorageClass.yaml ファイルを作成します。

メモ

ポートの変更を除いて、StorageClass の名前と構成パラメータが同じであることを確認します。

12. バックアップした PV.yaml ファイルを、新しい StorageClass.yaml ファイルで定義された新しいポート情報で更新します。
13. 修正した PV.yaml ファイルを使用して PV を再作成します。
14. 静的 PV 用の PVC を作成します。

メモ

PVC 名と構成パラメータが以前と同じであることを確認します。

15. Pod を作成します。

Secret の変更

操作手順

1. StorageClass を作成します。（3.2 StorageClass の設定（13 ページ）を参照してください）。
2. PVC を作成します。（3.10.4 PVC を作成する（30 ページ）を参照してください）。
3. Pod を作成します。（3.4 Pod の設定（16 ページ）を参照してください）。

   マッピングされたポートが StorageClass で定義されているものと同じであることを確認します。
4. Pod を削除します。

   これによりボリュームがアンマウントされ、ポートのマッピングが解除されます。
5. ポートを設定または変更するために secret.yaml ファイルを更新します。:

メモ

data の下のすべての値は、base64 エンコードされた値を記載する必要があります。

```
portID: Q0wyLUc=      # CL2-G

portIP: MTcyLjI1LjU5LjE5Mg==  # 172.25.59.192
```

また、stringData の下に設定をプレーンテキストで追加することもできます。この場合、Openshift は自動的に値を base64 に変換します。例：

```
apiVersion: v1

stringData:

  portID: CL2-G

  portIP: 172.25.59.192

data:

  password: SGl0YWNoaTE=

  url: aHR0cHM6Ly8xNzIuMjUuNDcuNTU=

  user: dWNwYQ==

kind: Secret
```

6. Pod を再作成します。

   Storage Plug-in for Containers は、StorageClass の代わりに Secret によって更新されたポートを使用するようになります。

7. Pod が実行中であることを確認します。:

```
oc get pods
```

iStorage V100/V110/V300/V310/V310F
NEC Storage Plug-in for Containers
Quick Reference Guide

IV-UG-402-005-10

2026 年 3 月 第 10 版 発行

日本電気株式会社

© NEC Corporation 2021-2026