



Cisco HyperFlex CSI によるステートフルアプリケーションの展開

- [Cisco HyperFlex CSI によるステートフルアプリケーションの展開の前提条件](#) (1 ページ)
- [管理者ホスト](#) (1 ページ)
- [ステートフルアプリケーションの展開](#) (2 ページ)

Cisco HyperFlex CSI によるステートフルアプリケーションの展開の前提条件

HyperFlex CSI ストレージ インテグレーションを使用してステートフルアプリケーションを展開する前に、次の前提条件を満たしている必要があります。

- Cisco HyperFlex クラスタがインストールされ、HX 5.0(x) 以降を実行している。
- Cisco HyperFlex CSI インテグレーションが展開されました。
- HX Connect の [iSCSI] タブから、iSCSI ネットワークを最初に作成する必要があります。詳細については、『[Cisco HyperFlex Data Platform Administration Guide, Release 5.0](#)』を参照してください。

管理者ホスト

このガイドでは、管理者ホストは kubectl コマンドなどを Kubernetes クラスタに対して実行するための Linux ベースのシステムのことです。これは通常、Kubernetes クラスタの一部ではない別のシステム (VM) ですが、別のシステム (VM) をインストール/管理する必要がない場合は、管理者ホストとして Kubernetes ノードの 1 つを使用できます。

ステートフル アプリケーションの展開

ステートフル アプリケーションを展開するには、次の手順を実行します。

永続的なボリューム要求の作成

永続ボリューム要求は、単にユーザによるストレージの要求です。ユーザは、ストレージ要件、必要なストレージのサイズまたは容量、およびその他のオプションを指定します。関連付けられたストレージクラスに応じて、ストレージ要件は、要求されたストレージをプロビジョニングし、Kubernetes で使用できるようにする適切なプロビジョニング担当者にルーティングされます。



(注) 最大 PVC サイズは 64Ti です。サポートされている最小 PVC サイズは 1 Gi です。



(注) CHAP で保護されたボリュームを作成できます。ターゲットごとに1つのストレージクラスで作成できるボリューム（永続的なボリューム要求）は最大 255 です。

ステップ 1 管理者ホストで、次の内容の「message-board-pvc.yaml」という名前のファイルを作成します。

例：

```
administrator-host:hxcsi$ cat ./message-board-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: message-board-pvc
spec:
  storageClassName: csi-hxcsi-default
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

ステップ 2 管理者ホストで、`kubectl create -f` コマンドを使用して永続ボリューム クレームを作成します。

例：

```
administrator-host:hxcsi$ kubectl create -f ./message-board-pvc.yaml

persistentvolumeclaim/message-board-pvc created
```

ステップ 3 管理者ホストで `kubectl get pvc` コマンドを使用して、永続ボリューム クレームが作成され、永続ボリュームに正常にバインドされていることを確認します。

例：

```
administrator-host:hxcsi$ kubectl get pvc
```

```

NAME          STATUS  VOLUME  CAPACITY  ACCESS  MODES  STORAGECLASS  AGE
message-board-pvc  BOUND  pvc-8069462e-662c-11e9-a163-005056a086d9  10Gi  RWO    csi-hxcsi-default  20s

```

ステートフル Kubernetes ワークロードの展開

Kubernetes ワークロードは、Kubernetes ワークロードのタイプに関係なく、ポッドや展開などのさまざまな形式で提供され、それぞれが Cisco HyperFlex CSI インテグレーションと永続ボリュームクレームを使用して永続ストレージを活用できます。次に、Cisco HyperFlex CSI インテグレーションのテストに使用できる Cisco Message Board と呼ばれるサンプルオープンソースアプリケーションの導入を示します。同じ方法と手順に従って、独自のアプリケーションでテストすることもできます。

ステップ 1 管理者ホストで、展開するワークロードを定義する YAML ファイルを作成します。

例：

以下は、Kubernetes Deployment と NodePort を介して展開された Cisco Message Board アプリケーションへの接続を可能にする Kubernetes Service の両方を作成するサンプルの Cisco Message Board アプリケーションの YAML ファイルを示しています。

(注) Kubernetes Deployment 定義の「ボリューム」セクションで永続ボリュームクレーム名を参照していること。この例では、「message-board-pvc」永続ボリュームクレームにバインドされた永続ボリュームが、「/sqlldb」の場所（パス）にある「message_board:version1」コンテナ内にマウントされます。

```

administrator-host:hxcsi$ cat ./message-board-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: message-board
  labels:
    app: message-board
spec:
  replicas: 1
  selector:
    matchLabels:
      app: message-board
  template:
    metadata:
      labels:
        app: message-board
        name: message-board
    spec:
      volumes:
        - name: demovolume1
          persistentVolumeClaim:
            claimName: message-board-pvc
      containers:
        - name: message-board
          image: michzimm/message_board:version1
          ports:
            - containerPort: 5000

```

```

      volumeMounts:
        - mountPath: "/sqldb"
          name: demovolume1
    ---
  apiVersion: v1
  kind: Service
  metadata:
    name: message-board
    labels:
      name: message-board
    namespace: default
  spec:
    type: NodePort
    ports:
      - port: 5000
        nodePort: 30002
    selector:
      name: message-board

```

ステップ 2 管理者ホストで `kubectl create -f` コマンドを使用して、展開とサービスを作成します。

例：

```

administrator-host:hxcsi$ kubectl create -f ./message-board-deployment.yaml
deployment.apps/message-board created
service/message-board created

```

ステップ 3 管理者ホストで `kubectl get pods` コマンドを使用して、展開されたポッドのステータスを確認します。

例：

```

administrator-host:hxcsi$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
csi-attacher-hxcsi-0                2/2    Running   0           3h51m
csi-attacher-hxcsi-1                2/2    Running   0           3h51m
csi-nodeplugin-hxcsi-9fgsf          2/2    Running   0           3h51m
csi-nodeplugin-hxcsi-qqvwj          2/2    Running   0           3h51m
csi-provisioner-hxcsi-0              2/2    Running   0           3h51m
csi-provisioner-hxcsi-1              2/2    Running   0           3h51m
csi-resizer-hxcsi-0                  2/2    Running   0           3h51m
csi-resizer-hxcsi-1                  2/2    Running   0           3h51m
message-board-6df65d6b59-49xhq      1/1    Running   0           95s

```

例：

ステップ 4 管理者ホストで、`kubectl get services` コマンドを使用して、展開されたサービスのステータスを確認します。

例：

```

root@administrator-host:hxcsi$ kubectl get services
NAME                                TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
csi-attacher-hxcsi                  ClusterIP       10.98.79.159    <none>           12346/TCP        3h53m
csi-provisioner-hxcsi                ClusterIP       10.99.73.185    <none>           12345/TCP        3h53m
kubernetes                           ClusterIP       10.96.0.1        <none>           443/TCP          4h24m
message-board                        NodePort        10.107.227.152  <none>           5000:30002/TCP  2m59s

```

サンプルの Cisco Message Board アプリケーションでは、サービスは「NodePort」とポート「30002」を使用して設定されます。これは、Web ブラウザで任意の Kubernetes ノードの IP アドレスとポート「30002」を指定することにより、アプリケーションが稼働中であることを意味します。例：`http://<k8s-worker1>:30002`

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。