



トラブルシューティング

- [トラブルシューティング \(1 ページ\)](#)
- [HXCSI ポッドの展開時の ImagePullBackOff ステータスエラー \(1 ページ\)](#)
- [ボリューム削除の失敗 \(2 ページ\)](#)
- [ノード削除中の ContainerCreating 状態のアプリケーション ポッド \(2 ページ\)](#)
- [終了中または ContainerCreating 状態のアプリケーション ポッド スタッド \(3 ページ\)](#)
- [削除されたポッドが同じノードに戻るようスケジュールされている \(4 ページ\)](#)

トラブルシューティング

次のセクションでは、HyperFlex CSI 統合をインストールして使用するときに見られる一般的な問題について説明します。提供される情報には、問題の診断に役立つ症状と、問題を解決するための解決策が含まれています。

HXCSI ポッドの展開時の ImagePullBackOff ステータスエラー

- **症状 1** : コマンド「`kubectl get pods [-n <namespace>]`」を実行すると、HXCSI ポッドのステータスが「ImagePullBackOff」であると表示されます。
- **症状 2** : コマンド「`kubectl description pod`」の実行<csi-pod_name>」には、「Error : ErrImagePull」および「Back-off pulling image...」というエラーを含むメッセージが表示されます。

解決方法 :

- **解決策 1** : `hxcsi-setup` スクリプトに指定された HXCSI コンテナイメージ名が正しいことを確認します。
- **解決策 2** : HXCSI コンテナイメージが、各 Kubernetes ワーカーノードの Docker 内に直接存在するか、またはローカルコンテナイメージレジストリに存在することを確認します。

- **解決策 3** : `hxcsi-setup` スクリプトによって生成される次の YAML ファイルの「`imagePullPolicy`」行が「`IfNotPresent`」に設定されていることを確認します。
`csi-attacher-hxcsi.yaml`、`csi-nodeplugin-hxcsi.yaml`、`csi-provisioner-hxcsi .yaml`
- **解決策 4** : 次のイメージが各 Kubernetes ノードのローカルコンテナ イメージ レジストリにロードされていることを確認します。`csi-attacher-3.2.1-cisco1.tar`、`csi-node-driver-registrar-2.2.0-cisco1.tar`、`csi-resizer-1.2.0-cisco1.tar`、`csi-provisioner-2.2.1-cisco1`

ボリューム削除の失敗

`NodeUnpublish` が成功し、ボリュームがマウント解除された後でも、古いボリューム接続が存在するため、ボリュームの削除は失敗します。これは、`etcd` リーダーの選択中に `delete volumeattachment kubernetes api` が失われた場合に発生します。`nodeUnpublish` が完了し、ボリュームがノードから正常にアンマウントされた後でも、ボリュームの削除は失敗します。

`external-provisioner` のログは次のように表示されます。

ボリュームの削除に失敗しました : `persistentvolume <pv-name>` はまだノード `<node-name>` に接続されています。

`external-attacher` のログは次のように表示されます。

`<Volume-attachment>` はすでに接続されています。

解決方法 :

次のコマンドを使用して、古いボリューム接続を削除します。

```
kubectl delete volumeattachments <VA-name>
```

プロビジョニング担当者が再試行すると、数秒後に `pv` が削除されます。

または、次のコマンドを使用して手動で削除することもできます。

```
kubectl delete pv <pv-name>
```

ノード削除中の **ContainerCreating** 状態のアプリケーションポッド

ノード削除中の `ContainerCreating` 状態のアプリケーションポッドまたはまたはマルチアタッチエラー状態でスタックし、ボリュームをマウントできません。これは、クラスタから `k8s` ワーカー ノードを削除または削除するときが発生することがあり、ポッドは新しいワーカーノードに移行します。

`K8` ワーカー ノードを削除する推奨方法は、次のコマンドを使用することです。

```
kubectl drain <node-name>
kubectl delete node <node-name>
```

詳細については、「[Kubernetes からノードを正常に削除するには \(How to gracefully remove a node from Kubernetes?\)](#)」を参照してください。

終了中または **ContainerCreating** 状態のアプリケーションポッドスタッド

アプリケーションポッドは、**Terminating** または **ContainerCreating** 状態で表示される場合があります。一般的な理由には、VMの再起動が完了せず、ハング状態になっていることが含まれます。これは、次の例に示すように、**systemd** に禁止ロックを設定するプロセスがあり、そのプロセスがシャットダウン前のタスクを完了せず、**systemd** がプロセスの終了に失敗した場合に発生します。

```
ubuntu@m5-k8-3:~$ systemctl-inhibit --list
Who: Unattended Upgrades Shutdown (UID 0/root, PID 778/unattended-upgr)
What: shutdown
Why: Stop ongoing upgrades or perform upgrades before shutdown
Mode: delay

1 inhibitors listed.
ubuntu@m5-k8-3:~$ ps aux | grep 778
root 778 0.0 0.1 185948 20028 ? Ssl May13 0:00 /usr/bin/python3
/usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
ubuntu@m5-k8-3:~$
```

iscsid が正常にシャットダウンできなかった場合、VMがハング状態のままになることがあります。シャットダウンプロセスは **iscsid** を強制終了する前に最大時間待機する場合がありますが、VMがリブートしない場合があります。次の例に示すように、報告された iSCSI 接続エラーを確認します。

```
Jun 09 19:12:46 m5-k19-3 iscsid[967]: Kernel reported iSCSI connection 2:0 error
(1010 - ISCSI_ERR_BAD_ITT: Received invalid initiator task tag from target) state (3)
Jun 09 19:12:48 m5-k19-3 iscsid[967]: connection2:0 is operational after recovery (1
attempts)
Jun 11 15:17:27 m5-k19-3 iscsid[967]: Kernel reported iSCSI connection 2:0 error (1010
-
ISCSI_ERR_BAD_ITT: Received invalid initiator task tag from target) state (3)
Jun 11 15:17:29 m5-k19-3 iscsid[967]: connection2:0 is operational after recovery (1
attempts)
```

ソリューション

VM コンソールにログインし、VMが正しくリブートし、**Kubelet** が応答していることを確認します。

- ワーカーノードは **Ready** 状態である必要があります。VMが完全にシャットダウンしていない場合は、再起動します。
- VMの再起動中に **iscsid** が正常にシャットダウンしない場合は、HXDP ターゲットへの iSCSI データパス接続（接続エラー、MTU、再試行など）を確認します。

VMの再起動またはリブートが実行された後、アプリケーションポッドが **Terminating** または **ContainerCreating** 状態で表示されることがあります。根本的な原因としては、VMがハングし、

シャットダウンプロセスが完了せず、Kubelet が応答しない状態になっていることが考えられます。デフォルトでは、API コントローラ マネージャは `kubernetes` ノードが起動するまで 5 分間待機し、その後アプリケーションポッドを別の場所で削除または再作成することを決定します。

この状態から回復するには、VM を再起動またはリセットし、VM が起動することを確認します。

削除されたポッドが同じノードに戻るようスケジュールされている

実行中 (**Running**) 状態のポッドが `kubectl delete pod` コマンドを使用して削除された後に再作成され、名前空間を削除すると、終了中 (**Terminating**) 状態でスタックしました。

実行中のポッドで `kubectl delete pod` コマンドを使用する代わりに、次のベストプラクティスの方法が推奨されます。

1. 削除するポッドが実行されているノード名をメモします。

```
kubectl get pods -o wide --all-namespaces
```

2. ポッドが実行されているノードのコードンをオフにします。

```
kubectl cordon <node-name>
```

3. ポッドを削除します。

```
kubectl delete pod <pod-name>
```

4. 削除されたポッドが別のノードでスケジュールされていることを次を使用して確認します。

```
kubectl get pods -o wide --all-namespaces
```

5. ノードのコードンを外します。

```
kubectl uncordon <node-name>
```

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。