



使用**SnapMirror**复制卷 Astra Trident

NetApp
January 14, 2026

目录

- 使用SnapMirror复制卷 1
 - 复制前提条件 1
 - 创建镜像PVC 1
 - 卷复制状态 4
 - 在计划外故障转移期间提升辅助PVC 5
 - 在计划内故障转移期间提升辅助PVC 5
 - 在故障转移后还原镜像关系 5
 - 其他操作 5
 - 将主PVC复制到新的二级PVC 6
 - 调整镜像、主PVC或二级PVC的大小 6
 - 从PVC中删除复制 6
 - 删除PVC (之前已镜像) 6
 - 删除TMR 6
 - 在ONTAP联机时更新镜像关系 6
 - 在ONTAP脱机时更新镜像关系 6
 - 启用Asta Control配置程序 7

使用SnapMirror复制卷

您可以使用Astra Control配置程序在一个集群上的源卷和对等集群上的目标卷之间创建镜像关系、以便为灾难恢复复制数据。您可以使用具有名称流的自定义资源定义(CRD)执行以下操作：

- 在卷之间创建镜像关系(PVC)
- 删除卷之间的镜像关系
- 中断镜像关系
- 在灾难情况下提升二级卷(故障转移)
- 在集群之间执行应用程序无中断过渡(在计划内故障转移或迁移期间)

复制前提条件

开始之前、请确保满足以下前提条件：

ONTAP 集群

- **Astra**控件配置程序：Astra控件配置程序23.10或更高版本必须位于使用ONTAP作为后端的源和目标Kubernetes集群上。
- 许可证：必须在源和目标ONTAP集群上启用使用数据保护包的ONTAP SnapMirror异步许可证。有关详细信息、请参见 ["ONTAP 中的SnapMirror许可概述"](#)。

对等

- **集群和SVM**：ONTAP存储后端必须建立对等状态。有关详细信息、请参见 ["集群和 SVM 对等概述"](#)。



确保两个ONTAP集群之间的复制关系中使用的SVM名称是唯一的。

- **Astra Control**置备程序和**SVM**：对等远程SVM必须可供目标集群上的Astra Control置备程序使用。

支持的驱动程序

- ONTAP -NAS和ONTAP SAN驱动程序支持卷复制。

创建镜像PVC

按照以下步骤并使用CRD示例在主卷和二级卷之间创建镜像关系。

步骤

1. 在主Kubernetes集群上执行以下步骤：
 - a. 使用参数创建StorageClass对象 `trident.netapp.io/replication: true`。

示例

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "nfs"
  trident.netapp.io/replication: "true"
```

- b. 使用先前创建的StorageClass创建PVC。

示例

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

- c. 使用本地信息创建镜像关系CR。

示例

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: promoted
  volumeMappings:
    - localPVCName: csi-nas
```

Astra Control配置程序会提取卷的内部信息以及卷的当前数据保护(DP)状态、然后填充镜像关系的状态字段。

- d. 获取TridentMirrorRelationship CR以获取PVC的内部名称和SVM。

```
kubectl get tmr csi-nas
```

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
  generation: 1
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
status:
  conditions:
  - state: promoted
    localVolumeHandle:
      "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
    localPVCName: csi-nas
    observedGeneration: 1
```

2. 在二级Kubernetes集群上执行以下步骤：

- a. 使用trident.netapp.io/replication: true参数创建StorageClass。

示例

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/replication: true
```

- b. 使用目标和源信息创建镜像关系CR。

示例

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: established
  volumeMappings:
    - localPVCName: csi-nas
      remoteVolumeHandle:
        "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
```

Asta控件配置程序将使用配置的关系策略名称(或ONTAP的默认策略名称)创建SnapMirror关系并对其进行初始化。

- c. 使用先前创建的StorageClass创建一个PVC以用作二级(SnapMirror目标)。

示例

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
  annotations:
    trident.netapp.io/mirrorRelationship: csi-nas
spec:
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 1Gi
storageClassName: csi-nas
```

Astra Control配置程序将检查是否存在TridentMirrorRelationship CRD、如果此关系不存在、则无法创建卷。如果存在此关系、Astra控件配置程序将确保将新FlexVol卷放置到与镜像关系中定义的远程SVM建立对等关系的SVM上。

卷复制状态

三级镜像关系(TCR)是一种CRD、表示PVC之间复制关系的一端。目标T关系 管理器具有一个状态、该状态会告诉Astra Control配置程序所需的状态是什么。目标T关系 管理器具有以下状态：

- 已建立：本地PVC是镜像关系的目标卷、这是一个新关系。
- 提升：本地PVC可读写并可挂载、当前未建立任何有效的镜像关系。

- 重新建立：本地PVC是镜像关系的目标卷、以前也位于该镜像关系中。
 - 如果目标卷曾经与源卷建立关系、因为它会覆盖目标卷的内容、则必须使用重新建立的状态。
 - 如果卷之前未与源建立关系、则重新建立的状态将失败。

在计划外故障转移期间提升辅助PVC

在二级Kubernetes集群上执行以下步骤：

- 将TridentMirrorRelationship的_spec.state_字段更新到 promoted。

在计划内故障转移期间提升辅助PVC

在计划内故障转移(迁移)期间、执行以下步骤以提升二级PVC：

步骤

1. 在主Kubernetes集群上、创建PVC的快照、并等待创建快照。
2. 在主Kubnetes集群上、创建SnapshotInfo CR以获取内部详细信息。

示例

```
kind: SnapshotInfo
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  snapshot-name: csi-nas-snapshot
```

3. 在二级Kubernetes集群上、将_TridentMirrorRelationship_CR的_spec.state_字段更新为_promoted_和_spec.promotedSnapshotHandle_、以成为快照的内部名称。
4. 在二级Kubernetes集群上、确认Trident镜像 关系的状态(stats.state字段)为已提升。

在故障转移后还原镜像关系

在还原镜像关系之前、请选择要用作新主卷的那一端。

步骤

1. 在二级Kubernetes集群上、确保已更新TudentMirrorRelationship上的_spec.netVolumeHandle_字段的值。
2. 在二级Kubernetes集群上、将Trident镜像 关系的_spec.mirector_字段更新到 reestablished。

其他操作

Asta Control配置程序支持在主卷和二级卷上执行以下操作：

将主PVC复制到新的二级PVC

确保您已有一个主PVC和一个次要PVC。

步骤

1. 从已建立的二级(目标)集群中删除PerbestentVolumeClaim和TridentMirorRelationship CRD。
2. 从主(源)集群中删除TridentMirorRelationship CRD。
3. 在主(源)集群上为要建立的新二级(目标) PVC创建新的TridentMirorRelationship CRD。

调整镜像、主PVC或二级PVC的大小

可以正常调整PVC的大小、如果数据量超过当前大小、ONTAP将自动扩展任何目标flexvol。

从PVC中删除复制

要删除复制、请对当前二级卷执行以下操作之一：

- 删除次要PVC上的镜像关系。此操作将中断复制关系。
- 或者、将spec.state字段更新为_promoted_。

删除PVC (之前已镜像)

ASRA Control配置程序会检查是否存在复制的PVC、并在尝试删除卷之前释放复制关系。

删除TMR

删除镜像关系一端的T磁 还原会导致剩余的T磁 还原在Astra Control配置程序完成删除之前过渡到_promoted状态。如果选定要删除的TMirror已处于_Promote 状态、则不存在现有镜像关系、此时TMirror将被删除、Astra Control配置程序会将本地PVC提升为_ReadWrite。此删除操作将释放ONTAP中本地卷的SnapMirror元数据。如果此卷将来要在镜像关系中使用、则在创建新镜像关系时、它必须使用具有_re设立_卷复制状态的新TMirror。

在ONTAP联机时更新镜像关系

建立镜像关系后、可以随时更新这些关系。您可以使用 state: promoted 或 state: reestablished 字段更新关系。将目标卷提升为常规ReadWrite卷时、可以使用_promotedSnapshotHandle_指定要将当前卷还原到的特定快照。

在ONTAP脱机时更新镜像关系

您可以使用CRD执行SnapMirror更新、而Astra Control不直接连接到ONTAP集群。请参阅以下TridentAction镜像 更新的示例格式：

示例

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

`status.state` 反映TridentAction镜像 更新CRD的状态。它可以从 `_suced_`、`_in Progress_` 或 `_failed` 中获取值。

启用Astra Control配置程序

Trident 23.10及更高版本提供了使用Astra Control配置程序的选项、允许获得许可的Astra Control用户访问高级存储配置功能。除了基于标准Astra三端CSI的功能之外、Astra Control配置程序还提供了此扩展功能。您可以使用此操作步骤启用和安装Astra控件配置程序。

您的Astra Control Service订阅会自动包含Astra Control配置程序使用的许可证。

在即将推出的Astra Control更新中、Astra Control配置程序将取代Astra Trandent作为存储配置程序和流程编排程序、并且Astra Control必须使用它。因此、强烈建议Astra Control用户启用Astra Control配置程序。Astra三元数据将继续保持开源状态、并使用NetApp的新CSI和其他功能进行发布、维护、支持和更新。

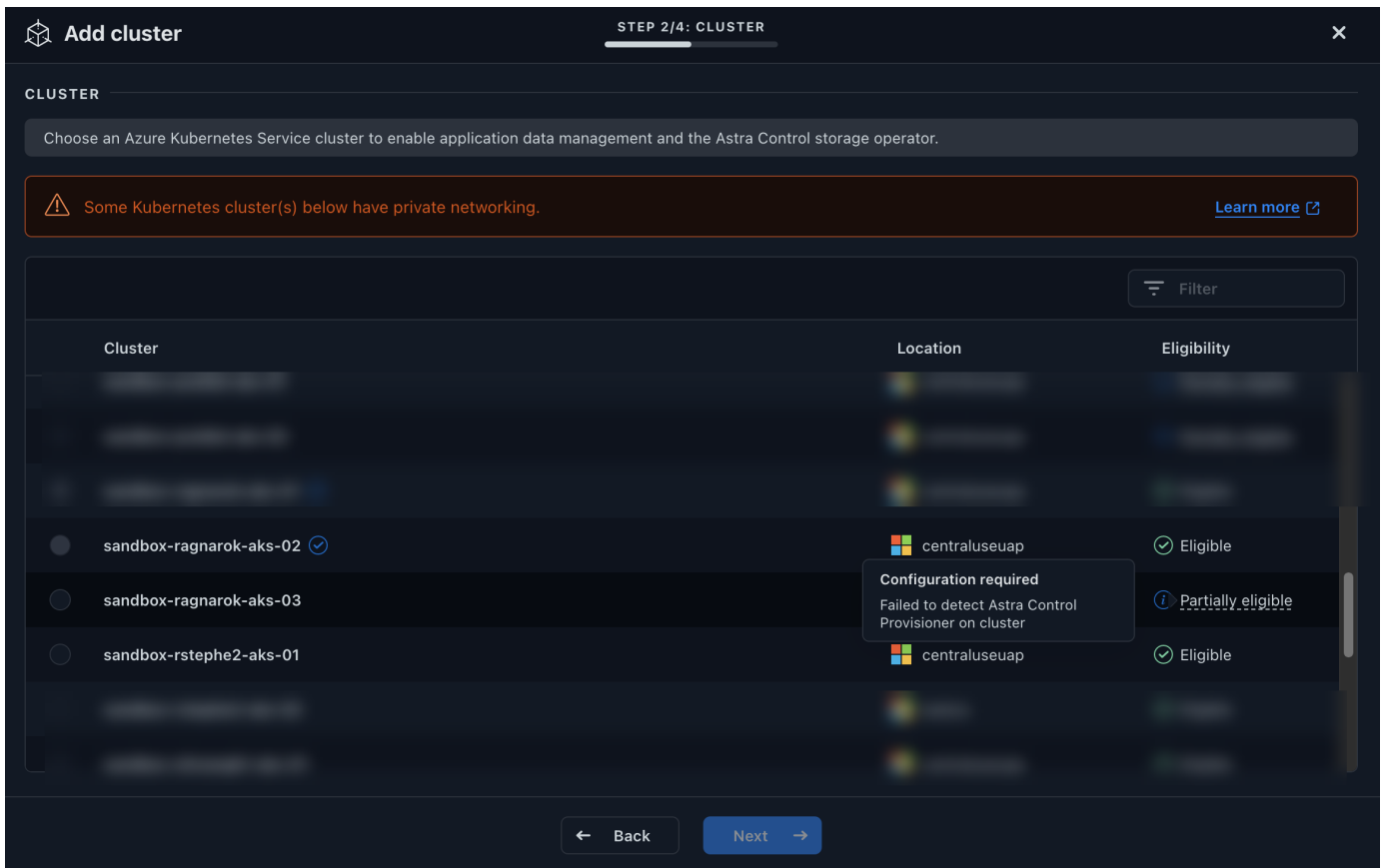
如何知道是否需要启用**Astra Control**配置程序？

如果您向Astra Control Service中添加的集群之前未安装Astra三端磁盘，则该集群将标记为 `Eligible`。之后 "[将集群添加到Astra Control](#)"，A作用 力控制配置程序将自动启用。

如果未标记集群，则会因为以下原因之一而标记该集群 `Eligible Partially eligible`：

- 它使用的是旧版本的Astra三端到子
- 它使用的是A作用 于三端的23.10、但尚未启用配置程序选项
- 此集群类型不允许自动启用

对于 `Partially eligible` 这种情况、请按照以下说明手动为集群启用Astra Control配置程序。



在启用**Astra Control**配置程序之前

如果您现有的Asta Trident没有Astra Control配置程序、但要启用Astra Control配置程序、请首先执行以下操作：

- 如果您安装了**Astra**三端安装程序，请确认其版本在四个版本的窗口内：如果您的Astra三端安装程序在版本24.02的四个版本窗口内，则可以使用Astra Control置备程序直接升级到Astra三端安装程序24.02。例如，您可以直接从Asta三端23.04升级到24.02。
- *确认集群具有一个AMD64*系统架构：Astra Control配置程序映像在amd64和ARM64 CPU架构中都提供，但Astra Control仅支持amd64。

步骤

1. 访问NetApp Astra控件映像注册表：

- 登录到Astra Control Service UI并记录您的Astra Control帐户ID。
 - 选择页面右上角的图图标。
 - 选择* API访问*。
 - 记下您的帐户ID。
- 在同一页面中，选择*Generate API令牌*并将API令牌字符串复制到剪贴板，然后将其保存在编辑器中。
- 使用您的首选方法登录Astra Control注册表：

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

```
crane auth login cr.astra.netapp.io -u <account-id> -p <api-token>
```

2. (仅限自定义注册表)按照以下步骤将图像移动到自定义注册表。如果您不使用注册表，请按照中的“三端”运算符步骤操作 [下一节](#)。



以下命令可以使用Podman、而不是Docker。如果您使用的是Windows环境、建议使用PowerShell。

Docker

- a. 从注册表中提取Asta Control配置程序映像：



提取的映像不支持多个平台、只支持与提取映像的主机相同的平台、例如Linux amd64。

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0  
--platform <cluster platform>
```

示例：

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0  
--platform linux/amd64
```

- b. 标记图像：

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0  
<my_custom_registry>/trident-acp:24.02.0
```

- c. 将映像推送到自定义注册表：

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

起重机

- a. 将Asta Control配置程序清单复制到自定义注册表：

```
crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0  
<my_custom_registry>/trident-acp:24.02.0
```

3. 确定原来的Asta三端安装方法是否使用了。
4. 使用您最初使用的安装方法在Asta Trident中启用Asta Control配置程序：

Asta三端操作员

- a. "下载Asta三端安装程序并解压缩"(英文)
- b. 如果您尚未安装Astra三端安装程序、或者您从初始Astra三端安装程序中删除了操作员、请完成以下步骤:
 - i. 创建客户需求日:

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.1
6.yaml
```

- ii. 创建三端命名空间 (kubectl create namespace trident)或确认三端命名空间仍然存在 (kubectl get all -n trident)。如果已删除此命名空间、请重新创建它。

- c. 将Astra Trident更新到24.02.0:



对于运行Kubernetes 1.24或更早版本的集群, 请使用 bundle_pre_1_25.yaml。
对于运行Kubernetes 1.25或更高版本的集群, 请使用 bundle_post_1_25.yaml。

```
kubectl -n trident apply -f trident-installer/deploy/<bundle-
name.yaml>
```

- d. 验证Astra trident是否正在运行:

```
kubectl get torc -n trident
```

响应:

NAME	AGE
trident	21m

- e. 如果您有一个使用机密的注册表, 请创建一个用于提取Astra Control置备程序映像的密钥:

```
kubectl create secret docker-registry <secret_name> -n trident
--docker-server=<my_custom_registry> --docker-username=<username>
--docker-password=<token>
```

- f. 编辑TridentOrchestrator CR并进行以下编辑:

```
kubectl edit torc trident -n trident
```

- i. 为Asta三端图像设置自定义注册表位置，或从Asta Control注册表或中提取该位置
(tridentImage: <my_custom_registry>/trident:24.02.0 tridentImage:
netapp/trident:24.02.0。
- ii. 启用Asta Control配置程序 (enableACP: true)。
- iii. 设置Asta Control置配置程序映像的自定义注册表位置，或从Asta Control注册表或中提取该映像
(acpImage: <my_custom_registry>/trident-acp:24.02.0 acpImage:
cr.astra.netapp.io/astra/trident-acp:24.02.0。
- iv. 如果您在本过程的前面已建立 [图像拉取密钥](#)，则可以在此处进行设置 (imagePullSecrets:
- <secret_name>)。使用您在前面步骤中创建的相同名称机密名称。

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.02.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.02.0
  imagePullSecrets:
    - <secret_name>
```

- g. 保存并退出文件。部署过程将自动开始。
- h. 验证是否已创建操作员、部署和副本集。

```
kubectl get all -n trident
```



在 Kubernetes 集群中只能有 * 一个操作符实例 *。请勿创建多个部署的Asta三端操作员。

- i. 验证 trident-acp 容器是否正在运行且 acpVersion 24.02.0 状态为 Installed:

```
kubectl get torc -o yaml
```

响应:

```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
status: Installed
```

Tridentctl

- "[下载Astra三端安装程序并解压缩](#)"(英文)
- "[如果您已有Astra Trident、请从托管它的集群中将其卸载](#)"(英文)
- 在启用Astra Control配置程序的情况下安装Astra Trent (--enable-acp=true):

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

- 确认已启用Astra Control配置程序:

```
./tridentctl -n trident version
```

响应:

```
+-----+-----+-----+ | SERVER
VERSION | CLIENT VERSION | ACP VERSION | +-----+
+-----+-----+-----+ | 24.02.0 | 24.02.0 | 24.02.0. |
+-----+-----+-----+ +-----+
```

掌舵

- 如果您安装了Astra Trident 23.07.1或更早版本、"[卸载](#)"则需要操作员和其他组件。
- 如果您的Kubornetes集群运行的是1.24或更早版本、请删除PSP:

```
kubect1 delete psp tridentoperatorpod
```

- 添加Astra Trident Helm存储库:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

d. 更新Helm图表:

```
helm repo update netapp-trident
```

响应:

```
Hang tight while we grab the latest from your chart
repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. ☐Happy Helming!☐
```

e. 列出图像:

```
./tridentctl images -n trident
```

响应:

```
| v1.28.0 | netapp/trident:24.02.0 |
| | docker.io/netapp/trident-
autosupport:24.02 |
| | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0 |
| | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0 |
| | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3 |
| | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3 |
| | registry.k8s.io/sig-storage/csi-node-
driver-registrar:v2.10.0 |
| | netapp/trident-operator:24.02.0 (optional)
```

f. 确保提供了三项运算符24.02.0:

```
helm search repo netapp-trident/trident-operator --versions
```

响应:

NAME	CHART VERSION	APP VERSION	
DESCRIPTION			
netapp-trident/trident-operator	100.2402.0	24.02.0	A

g. 使用 `helm install` 并运行包括这些设置的以下选项之一:

- 部署位置的名称
- Astra三端版本
- Asta Control配置程序映像的名称
- 用于启用配置程序的标志
- (可选)本地注册表路径。如果您使用的是本地注册表、则 " `{\f270 {\f151 {\f270}` " 可以位于一个注册表或不同的注册表中、但所有CSI图像都必须位于同一注册表中。
- 三端名称空间

选项

- 没有注册表的映像

```
helm install trident netapp-trident/trident-operator --version 100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-acp:24.02.0 --set enableACP=true --set operatorImage=netapp/trident-operator:24.02.0 --set tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02 --set tridentImage=netapp/trident:24.02.0 --namespace trident
```

- 一个或多个注册表中的图像

```
helm install trident netapp-trident/trident-operator --version 100.2402.0 --set acpImage=<your-registry>:<acp image> --set enableACP=true --set imageRegistry=<your-registry>/sig-storage --set operatorImage=netapp/trident-operator:24.02.0 --set tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02 --set tridentImage=netapp/trident:24.02.0 --namespace trident
```

您可以使用 `helm list` 查看安装详细信息、例如名称、命名空间、图表、状态、应用程序版本、和修订版号。

如果您在使用Helm部署TRident时遇到任何问题、请运行此命令以完全卸载Asta TRident:

```
./tridentctl uninstall -n trident
```

在尝试重新启用**Astra Control**配置程序之前，请勿 ["完全删除Asta Trident CRD"](#) 作为卸载的一部分。

结果

Asta Control配置程序功能已启用、您可以使用当前运行的版本可用的任何功能。

安装Asta Control配置程序后、在Asta Control UI中托管此配置程序的集群将显示一个 `ACP version` 而非 `Trident version` 字段以及当前安装的版本号。

CLUSTER STATUS

Available

Version	Managed	Kube-system namespace UID	ACP Version
v1.24.9+rke2r2	2024/03/15 17:32 UTC		
Private route identifier	Cloud instance	Default bucket	
	private	astra-bucket1 (inherited)	

Overview

Namespaces

Storage

Activity

有关详细信息 ...

- ["Asta Trident升级文档"](#)

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。