



使用 **Astra** Astra Control Center

NetApp
November 21, 2023

目录

- 使用 Astra 1
 - 开始管理应用程序 1
 - 保护应用程序 4
 - 监控应用程序和集群运行状况 35
 - 管理您的帐户 37
 - 管理存储分段 48
 - 管理存储后端 50
 - 使用Cloud Insights 和Fluentd连接监控基础架构 55
 - 取消管理应用程序和集群 62
 - 升级 Astra 控制中心 63
 - 卸载 Astra 控制中心 75

使用 Astra

开始管理应用程序

你先请 ["将集群添加到 Astra Control 管理中"](#)、您可以在集群上安装应用程序(在Astra Control之外)、然后转到Astra Control中的应用程序页面开始管理应用程序及其资源。

有关详细信息，请参见 ["应用程序管理要求"](#)。

支持的应用程序安装方法

Astra Control 支持以下应用程序安装方法：

- * 清单文件 *：Astra Control 支持使用 kubectl 从清单文件安装的应用程序。例如：

```
kubectl apply -f myapp.yaml
```

- * Helm 3*：如果使用 Helm 安装应用程序，则 Astra Control 需要 Helm 版本 3。完全支持管理和克隆随 Helm 3 安装的应用程序（或从 Helm 2 升级到 Helm 3）。不支持管理随 Helm 2 安装的应用程序。
- 操作员部署的应用程序：Astra Control支持使用命名空间范围的操作员安装的应用程序、这些应用程序通常采用"按价值传递"而不是"按参考传递"架构设计。操作员及其安装的应用程序必须使用相同的命名空间；您可能需要为操作员修改部署 .yaml 文件，以确保情况确实如此。

以下是一些遵循这些模式的操作员应用程序：

- ["Apache K8ssandra"](#)



对于 K8ssandra，支持原位还原操作。要对新命名空间或集群执行还原操作，需要关闭应用程序的原始实例。这是为了确保传输的对等组信息不会导致跨实例通信。不支持克隆应用程序。

- ["Jenkins CI"](#)
- ["Percona XtraDB 集群"](#)

Astra Control可能无法克隆使用"按参考传递"架构设计的运算符(例如CockroachDB运算符)。在这些类型的克隆操作期间，克隆的操作员会尝试引用源操作员提供的 Kubernetes 机密，尽管在克隆过程中他们拥有自己的新机密。克隆操作可能会失败，因为 Astra Control 不知道源运算符中的 Kubernetes 密钥。

在集群上安装应用程序

你先请 ["已添加集群"](#) 对于Astra Control、您可以在集群上安装应用程序或管理现有应用程序。可以管理范围仅限于单个命名空间的任何应用程序。

管理应用程序

在Astra Control发现集群上的命名空间后、您可以定义要管理的应用程序。您可以选择 ["将整个命名空间作为一个应用程序进行管理、或者单独管理命名空间中的一个或多个应用程序"](#)。这一切都可以细化到数据保护操作所

需的粒度级别。

虽然您可以使用Astra Control单独管理层次结构的两个级别(命名空间和该命名空间中的应用程序)、但最佳做法是选择一个或另一个。如果在命名空间和应用程序级别同时执行操作,则在Astra Control中执行的操作可能会失败。



例如、您可能希望为"Maria"设置一个每周节奏的备份策略、但您可能需要比该策略更频繁地备份"MariaDB"(位于同一命名空间中)。根据这些需求、您需要单独管理这些应用程序、而不是作为单命名空间应用程序来管理。

您需要的内容

- 已将Kubernetes集群添加到Astra Control中。
- 集群上安装的一个或多个应用程序。 [阅读有关支持的应用程序安装方法的更多信息。](#)
- 一个或多个活动Pod。
- 在已添加到Astra Control的Kubernetes集群上指定的命名空间。
- (可选)任何上的Kubernetes标签 ["支持的Kubernetes资源"](#)。



标签是一个键 / 值对,您可以将其分配给 Kubernetes 对象进行标识。通过标签,可以更轻松地对象进行排序,组织和查找。要了解有关 Kubernetes 标签的更多信息, "[请参见 Kubernetes 官方文档](#)"。

开始之前、您还应了解相关信息 ["管理标准命名空间和系统命名空间"](#)。

有关如何使用 Astra Control API 管理应用程序的说明,请参见 ["Astra Automation 和 API 信息"](#)。

应用程序管理选项

- [\[定义要作为应用程序进行管理的资源\]](#)
- [\[定义要作为应用程序进行管理的命名空间\]](#)

其他应用程序管理选项

- [\[取消管理应用程序\]](#)

定义要作为应用程序进行管理的资源

您可以指定 ["构成应用程序的Kubernetes资源"](#) 要使用Astra Control进行管理的。通过定义应用程序、您可以将Kubernetes集群中的元素分组到一个应用程序中。此Kubernetes资源集合按命名空间和标签选择器标准进行组织。

通过定义应用程序、您可以更精细地控制要包含在Astra Control操作中的内容、包括克隆、快照和备份。



定义应用程序时、请确保不在具有保护策略的多个应用程序中包含Kubernetes资源。Kubernetes资源上重叠的保护策略可能会发生原因 数据冲突。 [阅读有关最佳实践的更多信息。](#)

步骤

1. 从应用程序页面中、选择*定义*。
2. 在*定义应用程序*窗口中、输入应用程序名称。

3. 在*集群*下拉列表中选择运行应用程序的集群。
4. 从*命名空间*下拉列表中选择应用程序的命名空间。



只能在单个集群上的指定命名空间中定义应用程序。Astra Control不支持应用程序跨越多个命名空间或集群。

5. 输入应用程序和命名空间的标签。您可以指定单个标签或标签选择器条件(查询)。



要了解有关 Kubernetes 标签的更多信息，["请参见 Kubernetes 官方文档"](#)。

6. 选择*定义*后、根据需要对其他应用程序重复此过程。

定义完应用程序后、该应用程序将显示在应用程序页面上的应用程序列表中。现在、您可以克隆它并创建备份和快照。



您刚刚添加的应用程序在 "受保护 " 列下可能会显示一个警告图标，表示它尚未备份，并且尚未计划备份。



要查看特定应用程序的详细信息，请选择应用程序名称。

定义要作为应用程序进行管理的命名空间

您可以通过将命名空间的资源定义为应用程序来将命名空间中的所有Kubernetes资源添加到Astra Control管理中。如果您要以类似的方式并以通用间隔管理和保护特定命名空间中的所有资源、则此方法比单独定义应用程序更好。

步骤

1. 从集群页面中、选择一个集群。
2. 选择*命名空间*选项卡。
3. 选择包含要管理的应用程序资源的命名空间的"Actions"菜单、然后选择*定义为应用程序*。



如果要管理多个命名空间、请选择这些命名空间、然后选择左上角的*操作*按钮并选择*管理*。



选中*显示系统命名空间*复选框以显示默认情况下在应用程序管理中不使用的系统命名空间。

☐ Show system namespaces

["阅读更多内容"](#)。

此过程完成后、与此命名空间关联的应用程序将显示在中 Associated applications 列。

取消管理应用程序

如果您不再需要备份，创建快照或克隆某个应用程序，则可以停止对其进行管理。



如果取消管理某个应用程序，则先前创建的任何备份或快照都将丢失。

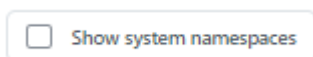
步骤

1. 从左侧导航栏中，选择 * 应用程序 *。
2. 选择应用程序。
3. 从菜单的*操作*列中、选择*取消管理*。
4. 查看相关信息。
5. 键入 "unmanage" 进行确认。
6. 选择 * 是，取消管理应用程序 *。

系统命名空间如何？

Astra Control还会发现Kubernetes集群上的系统命名空间。默认情况下、我们不会向您显示这些系统命名空间、因为您很少需要备份系统应用程序资源。

通过选中*显示系统命名空间*复选框、您可以从选定集群的命名空间选项卡中显示系统命名空间。



Astra Control 本身不是一个标准应用程序，而是一个 " 系统应用程序 "。您不应尝试管理 Astra Control 本身。默认情况下，用于管理的 Astra Control 本身不会显示。

示例：不同版本的单独保护策略

在此示例中、DevOps团队正在管理"金丝利"版本部署。该团队的集群中有三个Pod运行nginx。其中两个 Pod 专用于稳定版本。第三个 POD 适用于加那利版本。

开发运营团队的Kubernetes管理员会添加此标签 `deployment=stable` 稳定释放Pod。该团队将添加此标签 `deployment=canary` 加那利释放POD。

该团队的稳定版本要求每小时创建一次快照，每天进行备份。金那利版本更短暂、因此他们希望为任何标记的对象创建一个不太积极的短期保护策略 `deployment=canary`。

为了避免可能发生的数据冲突、管理员将创建两个应用程序：一个用于"加那利"版本、一个用于"稳定"版本。这样就可以使两组 Kubernetes 对象的备份，快照和克隆操作分开。

了解更多信息

- ["使用 Astra Control API"](#)

保护应用程序

保护概述

您可以使用 Astra 控制中心为应用程序创建备份，克隆，快照和保护策略。备份应用程序可帮助您的服务和关联数据尽可能地可用；在灾难情形下，从备份还原可以确保应用程序及其关联数据的完全恢复，而不会造成任何中断。备份，克隆和快照有助于防止常见威胁，例如勒索软件，意外数据丢失和环境灾难。 ["了解 Astra 控制中心提供的保护类型以及何时使用"](#)。

此外、您还可以将应用程序复制到远程集群、以便为灾难恢复做好准备。

应用程序保护工作流

您可以使用以下示例工作流开始保护应用程序。

[一个] 保护所有应用程序

要确保您的应用程序立即受到保护，"[为所有应用程序创建手动备份](#)"。

[两个] 为每个应用程序配置一个保护策略

要自动执行未来备份和快照，"[为每个应用程序配置一个保护策略](#)"。例如，您可以从每周备份和每日快照开始，这两种备份均保留一个月。强烈建议使用保护策略自动执行备份和快照，而不是手动备份和快照。

[三个] 调整保护策略

随着应用程序及其使用模式的变化，根据需要调整保护策略以提供最佳保护。

[四个] 将应用程序复制到远程集群

"[复制应用程序](#)" 使用 NetApp SnapMirror 技术连接到远程集群。Astra Control 可将快照复制到远程集群、从而提供异步灾难恢复功能。

[五个] 发生灾难时、请使用最新备份或复制功能将应用程序还原到远程系统

如果发生数据丢失，您可以通过进行恢复 "[还原最新备份](#)" 每个应用程序的第一个。然后，您可以还原最新的快照（如果可用）。或者、您也可以使用复制到远程系统。

通过快照和备份保护应用程序

通过使用自动保护策略或临时创建快照和备份来保护所有应用程序。您可以使用 Astra UI 或 "[Astra Control API](#)" 保护应用程序。

如果您使用 Helm 部署应用程序，则 Astra 控制中心需要 Helm 版本 3。完全支持管理和克隆使用 Helm 3 部署的应用程序（或从 Helm 2 升级到 Helm 3）。不支持使用 Helm 2 部署的应用程序。

在 OpenShift 集群上创建用于托管应用程序的项目时，系统会该项目（或 Kubernetes 命名空间）分配一个 SecurityContext UID。要使 Astra 控制中心能够保护您的应用程序并将应用程序移动到 OpenShift 中的其他集群或项目，您需要添加策略，使应用程序能够作为任何 UID 运行。例如，以下 OpenShift 命令行界面命令会为 WordPress 应用程序授予相应的策略。

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

您可以执行以下与保护应用程序数据相关的任务：

- [\[配置保护策略\]](#)
- [\[创建快照\]](#)
- [\[创建备份\]](#)

- [查看快照和备份]
- [删除快照]
- [取消备份]
- [删除备份]

配置保护策略

保护策略通过按定义的计划创建快照，备份或这两者来保护应用程序。您可以选择每小时，每天，每周和每月创建快照和备份，并且可以指定要保留的副本数。例如，保护策略可能会创建每周备份和每日快照，并将备份和快照保留一个月。创建快照和备份的频率以及保留时间取决于组织的需求。

步骤

1. 选择 * 应用程序 *，然后选择应用程序的名称。
2. 选择 * 数据保护 *。
3. 选择 * 配置保护策略 *。
4. 通过选择每小时，每天，每周和每月保留的快照和备份数量来定义保护计划。

您可以同时定义每小时，每天，每周和每月计划。在设置保留级别之前，计划不会变为活动状态。

以下示例将为快照和备份设置四个保护计划：每小时，每天，每周和每月。

Hourly

Every hour on the 0th minute, keep the last 4 snapshots

Daily

Daily at 02:00 (UTC), keep the last 15 snapshots

Weekly

Weekly on Mondays at 02:00 (UTC), keep the last 26 snapshots

Monthly

Every 1st of the month at 02:00 (UTC), keep the last 12 backups

Hourly

Daily

Weekly

Monthly

Select Weekday(s) (optional)

Monday X

Time (UTC) (optional)

02:00

Snapshots to keep

26

Backups to keep

0

Bucket

ntp-nautilus-bucket-10 - ntp-nautilus-bucket-10

Default

Overview

Schedule and retention

Define a policy to continuously protect your application on a schedule and configure a retention count to get started.

For select stateful applications, expect I/O to pause for a short time during a backup or snapshot operation.

Read more in [Protection policies](#)

Cancel

Review →

- 选择 * 审阅 *。
- 选择 * 设置保护策略。 *

结果

Astra 控制中心通过使用您定义的计划和保留策略创建和保留快照和备份来实施数据保护策略。

创建快照

您可以随时创建按需快照。

步骤

1. 选择 * 应用程序 *。
2. 从所需应用程序的 * 操作 * 列的选项菜单中，选择 * 快照 *。
3. 自定义快照的名称，然后选择 * 审阅 *。
4. 查看快照摘要并选择 * 快照 *。

结果

快照过程开始。如果在 * 数据保护 * > * 快照 * 页面的 * 操作 * 列中的状态为 * 可用 *，则快照将成功。

创建备份

您也可以随时备份应用程序。



Astra 控制中心中的 S3 存储分段不会报告可用容量。在备份或克隆由 Astra 控制中心管理的应用程序之前，请检查 ONTAP 或 StorageGRID 管理系统中的存储分段信息。

步骤

1. 选择 * 应用程序 *。
2. 从所需应用程序的 * 操作 * 列的选项菜单中，选择 * 备份 *。
3. 自定义备份的名称。
4. 选择是否从现有快照备份应用程序。如果选择此选项，则可以从现有快照列表中进行选择。
5. 通过从存储分段列表中选择来选择备份的目标。
6. 选择 * 审阅 *。
7. 查看备份摘要并选择 * 备份 *。

结果

Astra 控制中心创建应用程序的备份。



如果网络发生中断或异常缓慢，备份操作可能会超时。这会导致备份失败。



无法停止正在运行的备份。如果需要删除备份，请等待备份完成，然后按照中的说明进行操作 [\[删除备份\]](#)。删除失败的备份，["使用 Astra Control API"](#)。



在执行数据保护操作（克隆，备份，还原）并随后调整永久性卷大小后，在 UI 中显示新卷大小之前，最长会有 20 分钟的延迟。数据保护操作将在几分钟内成功完成，您可以使用存储后端的管理软件确认卷大小的更改。

查看快照和备份

您可以从数据保护选项卡查看应用程序的快照和备份。

步骤

1. 选择 * 应用程序 * ，然后选择应用程序的名称。
2. 选择 * 数据保护 * 。

默认情况下会显示快照。

3. 选择 * 备份 * 可查看备份列表。

删除快照

删除不再需要的计划快照或按需快照。



您不能删除当前正在复制的Snapshot副本。

步骤

1. 选择 * 应用程序 * ，然后选择应用程序的名称。
2. 选择 * 数据保护 * 。
3. 从选项菜单的 * 操作 * 列中为所需快照选择 * 删除快照 * 。
4. 键入单词 "delete" 确认删除，然后选择 * 是，删除 snapshot* 。

结果

Astra 控制中心会删除快照。

取消备份

您可以取消正在进行的备份。



要取消备份、备份必须处于运行状态。您不能取消处于"Pending"状态的备份。

步骤

1. 选择 * 应用程序 * ，然后选择应用程序的名称。
2. 选择 * 数据保护 * 。
3. 选择 * 备份 * 。
4. 从选项菜单中的*操作*列中为所需备份选择*取消*。
5. 键入单词"cancel"以确认删除、然后选择*是、取消备份*。

删除备份

删除不再需要的计划备份或按需备份。



无法停止正在运行的备份。如果需要删除备份，请等待备份完成，然后按照以下说明进行操作。删除失败的备份，["使用 Astra Control API"](#)。

步骤

1. 选择 * 应用程序 *，然后选择应用程序的名称。
2. 选择 * 数据保护 *。
3. 选择 * 备份 *。
4. 从选项菜单的 * 操作 * 列中为所需备份选择 * 删除备份 *。
5. 键入单词 "delete" 确认删除，然后选择 * 是，删除备份 *。

结果

Astra 控制中心删除备份。

还原应用程序

Astra Control 可以从快照或备份还原应用程序。将应用程序还原到同一集群时，从现有快照进行还原的速度会更快。您可以使用 Astra Control UI 或 ["Astra Control API"](#) 还原应用程序。

关于此任务

- 强烈建议在还原应用程序之前为其创建快照或备份。这样、您可以在还原失败时从快照或备份克隆。
- 如果您使用 Helm 部署应用程序，则 Astra 控制中心需要 Helm 版本 3。完全支持管理和克隆使用 Helm 3 部署的应用程序（或从 Helm 2 升级到 Helm 3）。不支持使用 Helm 2 部署的应用程序。
- 如果要还原到其他集群，请确保此集群使用相同的永久性卷访问模式（例如 ReadWriteMany）。如果目标永久性卷访问模式不同，还原操作将失败。
- 任何按命名空间名称 /ID 或命名空间标签限制命名空间的成员用户都可以将应用程序克隆或还原到同一集群上的新命名空间或其组织帐户中的任何其他集群。但是，同一用户无法访问新命名空间中的克隆或还原应用程序。通过克隆或还原操作创建新命名空间后，帐户管理员 / 所有者可以编辑成员用户帐户并更新受影响用户的角色约束，以授予对新命名空间的访问权限。
- 在 OpenShift 集群上创建用于托管应用程序的项目时，系统会该项目（或 Kubernetes 命名空间）分配一个 SecurityContext UID。要使 Astra 控制中心能够保护您的应用程序并将应用程序移动到 OpenShift 中的其他集群或项目，您需要添加策略，使应用程序能够作为任何 UID 运行。例如，以下 OpenShift 命令行界面命令会为 WordPress 应用程序授予相应的策略。

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

步骤

1. 选择 * 应用程序 *，然后选择应用程序的名称。
2. 选择 * 数据保护 *。
3. 如果要从快照还原，请保持选中 * 快照 * 图标。否则，请选择 * 备份 * 图标以从备份中还原。
4. 从要还原的快照或备份的 * 操作 * 列的选项菜单中，选择 * 还原应用程序 *。

5. * 还原详细信息 *：指定已还原应用程序的详细信息。默认情况下，将显示当前集群和命名空间。保留这些值不变，以便原位还原应用程序，从而将应用程序还原到其自身的早期版本。如果要还原到其他集群或命名空间，请更改这些值。
- 输入应用程序的名称和命名空间。
 - 选择应用程序的目标集群。
 - 选择 * 审阅 *。



如果还原到先前已删除的命名空间、则在还原过程中会创建一个同名的新命名空间。任何有权管理先前删除的命名空间中的应用程序的用户都需要手动还原对新重新创建的命名空间的权限。

6. * 还原摘要 *：查看有关还原操作的详细信息，键入 "restore"，然后选择 * 还原 *。

结果

Astra 控制中心会根据您提供的信息还原应用程序。如果您已原位还原应用程序，则任何现有永久性卷的内容将替换为还原应用程序中的永久性卷的内容。



在执行数据保护操作(克隆、备份、还原)并随后调整永久性卷大小后、在Web UI中显示新卷大小之前、最多会有20分钟的延迟。数据保护操作将在几分钟内成功完成，您可以使用存储后端的管理软件确认卷大小的更改。

使用SnapMirror技术将应用程序复制到远程系统

使用Astra Control、您可以使用NetApp SnapMirror技术的异步复制功能、以低RPO (恢复点目标)和低RTO (恢复时间目标)为应用程序构建业务连续性。配置完成后、应用程序便可将数据和应用程序更改从一个集群复制到另一个集群。

有关备份/还原与复制之间的比较、请参见 ["数据保护概念"](#)。

您可以在不同情形下复制应用程序、例如以下仅限内部部署、混合和多云情形：

- 内部站点A到内部站点B
- 使用Cloud Volumes ONTAP 从内部部署到云
- 采用Cloud Volumes ONTAP 的云到内部部署
- 采用Cloud Volumes ONTAP 的云到云(在同一云提供商的不同区域之间或不同云提供商之间)

Astra Control可以跨内部集群、内部到云(使用Cloud Volumes ONTAP)或云之间(Cloud Volumes ONTAP到Cloud Volumes ONTAP)复制应用程序。



您可以同时按相反方向复制另一个应用程序(在另一个集群或站点上运行)。例如、应用程序A、B、C可以从数据中心1复制到数据中心2；应用程序X、Y、Z可以从数据中心2复制到数据中心1。

使用Astra Control、您可以执行以下与复制应用程序相关的任务：

- [\[设置复制关系\]](#)
- [\[在目标集群上使复制的应用程序联机\(故障转移\)\]](#)

- [\[重新同步故障转移复制\]](#)
- [\[反向复制应用程序\]](#)
- [\[将应用程序故障恢复到原始源集群\]](#)
- [\[删除应用程序复制关系\]](#)

复制前提条件

请参见 ["复制前提条件"](#) 开始之前。

设置复制关系

设置复制关系涉及到构成复制策略的以下内容；

- 选择Astra Control创建应用程序快照的频率(包括应用程序的Kubernetes资源以及应用程序每个卷的卷快照)
- 选择复制计划(包括Kubernetes资源以及永久性卷数据)
- 设置创建快照的时间

步骤

1. 从Astra Control左侧导航栏中、选择*应用程序*。
2. 在应用程序页面中、选择*数据保护*>*复制*选项卡。
3. 在Data Protection > Replication选项卡中、选择*配置复制策略*。或者、从应用程序保护框中、选择操作选项并选择*配置复制策略*。
4. 输入或选择以下信息：
 - 目标集群
 - 目标存储类：选择或输入在目标ONTAP 集群上使用配对SVM的存储类。
 - 复制类型："异步"是当前唯一可用的复制类型。
 - 目标命名空间：为目标集群输入新的或现有的目标命名空间。



选定命名空间中任何冲突的资源都将被覆盖。

- 复制频率：设置Astra Control创建Snapshot并将其复制到目标的频率。
 - 偏移：设置从Astra Control创建快照的小时数开始的分钟数。您可能希望使用偏移量、以便它不会与其他计划的操作保持一致。例如、如果要从10：02开始每5分钟创建一次Snapshot、请输入"02"作为偏移分钟数。结果为10：02、10：07、10：12等
5. 选择*下一步*、查看摘要、然后选择*保存*。



首先、在执行第一个计划之前、状态将显示"app-mirror"。

Astra Control会创建用于复制的应用程序Snapshot。

6. 要查看应用程序Snapshot状态、请选择*应用程序*>*快照*选项卡。

Snapshot名称使用"replication-schedule-`<string>`"格式。Astra Control会保留用于复制的最后一个

Snapshot。成功完成复制后、所有较早的复制Snapshot都会被删除。

结果

这将创建复制关系。

建立关系后、Astra Control将完成以下操作：

- 在目标上创建命名空间(如果不存在)
- 在目标命名空间上创建与源应用程序的PVC对应的PVC。
- 创建初始应用程序一致的Snapshot。
- 使用初始Snapshot为永久性卷建立SnapMirror关系。

"数据保护"页面将显示复制关系的状态和状态：<运行状况>|<关系生命周期状态>

例如：normal | established.

在下了解有关复制状态和状态的更多信息。

在目标集群上使复制的应用程序联机(故障转移)

使用Astra Control、您可以将复制的应用程序"故障转移"到目标集群。此操作步骤 将停止复制关系并使应用程序在目标集群上联机。如果应用程序正常运行、则此操作步骤 不会停止源集群上的应用程序。

步骤

1. 从Astra Control左侧导航栏中、选择*应用程序*。
2. 在应用程序页面中、选择*数据保护*>*复制*选项卡。
3. 在"数据保护">"复制"选项卡的"操作"菜单中、选择*故障转移*。
4. 在故障转移页面中、查看相关信息并选择*故障转移*。

结果

故障转移操作步骤 会执行以下操作：

- 在目标集群上、应用程序将根据最新复制的Snapshot启动。
- 源集群和应用程序(如果运行正常)不会停止、并且将继续运行。
- 复制状态将更改为"故障转移"、然后在完成后更改为"故障转移"。
- 根据故障转移时源应用程序上的计划、源应用程序的保护策略将复制到目标应用程序。
- Astra Control会在源集群和目标集群上显示应用程序及其各自的运行状况。

重新同步故障转移复制

重新同步操作将重新建立复制关系。您可以选择关系的源、以便在源或目标集群上保留数据。此操作将重新建立SnapMirror关系、以便按所选方向启动卷复制。

此过程会在重新建立复制之前停止新目标集群上的应用程序。



在重新同步过程中、生命周期状态将显示为"正在建立"。

步骤

1. 从Astra Control左侧导航栏中、选择*应用程序*。
2. 在应用程序页面中、选择*数据保护*>*复制*选项卡。
3. 在"Data Protection">"Replication"选项卡中、从"Actions"菜单中选择*重新同步*。
4. 在重新同步页面中、选择包含要保留的数据的源或目标应用程序实例。



请仔细选择重新同步源、因为目标上的数据将被覆盖。

5. 选择*重新同步*以继续。
6. 键入"resync-"进行确认。
7. 选择*是、重新同步*以完成。

结果

- 复制页面将显示"正在建立"作为复制状态。
- Astra Control将停止新目标集群上的应用程序。
- Astra Control使用SnapMirror重新同步功能按选定方向重新建立永久性卷复制。
- 复制页面将显示已更新的关系。

反向复制应用程序

这是一项计划内操作、用于将应用程序移动到目标集群、同时继续复制回原始源集群。Astra Control会先停止源集群上的应用程序并将数据复制到目标、然后再将应用程序故障转移到目标集群。

在这种情况下、您将交换源和目标。原始源集群将成为新的目标集群、而原始目标集群将成为新的源集群。

步骤

1. 从Astra Control左侧导航栏中、选择*应用程序*。
2. 在应用程序页面中、选择*数据保护*>*复制*选项卡。
3. 在"Data Protection">"Replication"选项卡中、从"Actions"菜单中选择*反向复制*。
4. 在反向复制页面中、查看相关信息并选择*反向复制*以继续。

结果

反向复制会执行以下操作：

- 将为原始源应用程序的Kubernetes资源创建Snapshot。
- 通过删除原始源应用程序的Kubernetes资源(保留PVC和PV)、可以正常停止原始源应用程序的Pod。
- 关闭Pod后、将创建并复制应用程序卷的快照。
- SnapMirror关系将中断、从而使目标卷做好读/写准备。
- 应用程序的Kubernetes资源会使用在原始源应用程序关闭后复制的卷数据从预关闭的Snapshot进行还原。

- 反向重新建立复制。

将应用程序故障恢复到原始源集群

使用Astra Control、您可以通过以下操作序列在"故障转移"操作后实现"故障恢复"。在此恢复原始复制方向的工作流中、Astra Control会将所有应用程序更改复制(重新同步)回原始源集群、然后再反转复制方向。

此过程从已完成故障转移到目标的关系开始、涉及以下步骤：

- 从故障转移状态开始。
- 重新同步此关系。
- 反转复制。

步骤

1. 从Astra Control左侧导航栏中、选择*应用程序*。
2. 在应用程序页面中、选择*数据保护*>*复制*选项卡。
3. 在"Data Protection">"Replication"选项卡中、从"Actions"菜单中选择*重新同步*。
4. 对于故障恢复操作、请选择故障转移应用程序作为重新同步操作的源(保留故障转移后写入的任何数据)。
5. 键入"resync-"进行确认。
6. 选择*是、重新同步*以完成。
7. 重新同步完成后、在"Data Protection">"Replication"选项卡中、从"Actions"菜单中选择*反向复制*。
8. 在反向复制页面中、查看相关信息并选择*反向复制*。

结果

这将合并"重新同步"和"反向关系"操作的结果、以便在复制恢复到原始目标集群的情况下使应用程序在原始源集群上联机。

删除应用程序复制关系

删除此关系会导致出现两个独立的应用程序、它们之间没有任何关系。

步骤

1. 从Astra Control左侧导航栏中、选择*应用程序*。
2. 在应用程序页面中、选择*数据保护*>*复制*选项卡。
3. 在"数据保护">"复制"选项卡的"应用程序保护"框或关系图中、选择*删除复制关系*。

结果

删除复制关系后会执行以下操作：

- 如果已建立此关系、但此应用程序尚未在目标集群上联机(故障转移)、则Astra Control将保留初始化期间创建的PVC、在目标集群上保留一个"空"受管应用程序、并保留目标应用程序以保留可能已创建的任何备份。
- 如果应用程序已在目标集群上联机(故障转移)、则Astra Control会保留PVC和目标应用程序。源应用程序和目标应用程序现在被视为独立的应用程序。备份计划会同时保留在两个应用程序上、但不会彼此关联。

复制关系运行状况和关系生命周期状态

Astra Control显示关系的运行状况以及复制关系的生命周期状态。

复制关系运行状况

以下状态指示复制关系的运行状况：

- 正常：此关系正在建立或已建立、并且已成功传输最新的Snapshot。
- 警告：此关系正在进行故障转移或已进行故障转移(因此不再保护源应用程序)。
- * 严重 *
 - 此关系正在建立或故障转移、上次协调尝试失败。
 - 已建立此关系、上次尝试协调添加新PVC失败。
 - 已建立此关系(因此已成功复制Snapshot、并且可以进行故障转移)、但最近的Snapshot无法复制或无法复制。

复制生命周期状态

以下状态反映了复制生命周期的不同阶段：

- 正在建立：正在创建新的复制关系。Astra Control会根据需要创建命名空间、在目标集群上的新卷上创建永久性卷声明(PVC)、并创建SnapMirror关系。此状态还可以指示复制正在重新同步或反转复制。
- 已建立：存在复制关系。Astra Control会定期检查PVC是否可用、检查复制关系、定期创建应用程序的Snapshot并确定应用程序中的任何新源PVC。如果是、则Astra Control会创建资源以将其包括在复制中。
- 故障转移：Astra Control中断SnapMirror关系、并从上次成功复制的应用程序Snapshot还原应用程序的Kubernetes资源。
- 故障转移：Astra Control停止从源集群复制、在目标上使用最新(成功)复制的应用程序Snapshot、并还原Kubernetes资源。
- 正在重新同步：Astra Control使用SnapMirror重新同步将重新同步源上的新数据重新同步到重新同步目标。此操作可能会根据同步方向覆盖目标上的某些数据。Astra Control会停止在目标命名空间上运行的应用程序、并删除Kubernetes应用程序。在重新同步过程中、状态将显示为正在建立。
- 正在反转：是指在继续复制回原始源集群的同时将应用程序移动到目标集群的计划操作。Astra Control会停止源集群上的应用程序、将数据复制到目标、然后将应用程序故障转移到目标集群。在反向复制期间、状态显示为"正在 建立"。
- 正在删除：
 - 如果已建立复制关系、但尚未进行故障转移、则Astra Control会删除复制期间创建的PVC、并删除目标受管应用程序。
 - 如果复制已失败、则Astra Control会保留PVC和目标应用程序。

克隆和迁移应用程序

克隆现有应用程序以在同一个 Kubernetes 集群或另一个集群上创建重复的应用程序。当 Astra 控制中心克隆应用程序时，它会为您的应用程序配置和永久性存储创建一个克隆。

如果您需要将应用程序和存储从一个 Kubernetes 集群移动到另一个集群，则克隆可以助您一臂之力。例如，您

可能希望通过 CI/CD 管道以及在 Kubernetes 命名空间之间移动工作负载。您可以使用 Astra UI 或 "[Astra Control API](#)" 克隆和迁移应用程序。

您需要的内容

要将应用程序克隆到其他集群，您需要一个默认存储分段。添加第一个存储分段时，它将成为默认存储分段。

关于此任务

- 如果您部署的应用程序明确设置了 StorageClass，并且需要克隆该应用程序，则目标集群必须具有最初指定的 StorageClass。将显式设置了 StorageClass 的应用程序克隆到不具有相同 StorageClass 的集群将失败。
- 如果克隆操作员部署的 Jenkins CI 实例，则需要手动还原永久性数据。这是应用程序部署模式的一个限制。
- Astra 控制中心中的 S3 存储分段不会报告可用容量。在备份或克隆由 Astra 控制中心管理的应用程序之前，请检查 ONTAP 或 StorageGRID 管理系统中的存储分段信息。
- 在应用程序备份或应用程序还原期间，您可以选择指定存储分段 ID。但是，应用程序克隆操作始终使用已定义的默认分段。没有选项可用于更改克隆的分段。如果要控制使用哪个存储分段，您可以选择 "[更改存储分段默认值](#)" 或者执行 "[backup](#)" 后跟 A "[还原](#)" 请单独使用。
- 任何按命名空间名称 /ID 或命名空间标签限制命名空间的成员用户都可以将应用程序克隆或还原到同一集群上的新命名空间或其组织帐户中的任何其他集群。但是，同一用户无法访问新命名空间中的克隆或还原应用程序。通过克隆或还原操作创建新命名空间后，帐户管理员 / 所有者可以编辑成员用户帐户并更新受影响用户的角色约束，以授予对新命名空间的访问权限。

OpenShift 注意事项

- 如果您在集群之间克隆应用程序，则源集群和目标集群必须是 OpenShift 的同一分发版。例如，如果从 OpenShift 4.7 集群克隆应用程序，请使用同时也是 OpenShift 4.7 的目标集群。
- 在 OpenShift 集群上创建用于托管应用程序的项目时，系统会该项目（或 Kubernetes 命名空间）分配一个 SecurityContext UID。要使 Astra 控制中心能够保护您的应用程序并将应用程序移动到 OpenShift 中的其他集群或项目，您需要添加策略，使应用程序能够作为任何 UID 运行。例如，以下 OpenShift 命令行界面命令会为 WordPress 应用程序授予相应的策略。

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

步骤

1. 选择 * 应用程序 *。
2. 执行以下操作之一：
 - 在 * 操作 * 列中选择所需应用程序的选项菜单。
 - 选择所需应用程序的名称，然后选择页面右上角的状态下拉列表。
3. 选择 * 克隆 *。
4. * 克隆详细信息 *：指定克隆的详细信息：
 - 输入名称。
 - 输入克隆的命名空间。
 - 选择克隆的目标集群。
 - 选择是要从现有快照还是备份创建克隆。如果不选择此选项，则 Astra 控制中心将根据应用程序的当前

状态创建克隆。

5. * 源 *：如果选择从现有快照或备份克隆，请选择要使用的快照或备份。
6. 选择 * 审阅 *。
7. * 克隆摘要 *：查看有关克隆的详细信息并选择 * 克隆 *。

结果

Astra 控制中心会根据您提供的信息克隆该应用程序。如果新应用程序克隆位于中、则克隆操作将成功 Available 状态。



在执行数据保护操作（克隆，备份，还原）并随后调整永久性卷大小后，在 UI 中显示新卷大小之前，最长会有 20 分钟的延迟。数据保护操作将在几分钟内成功完成，您可以使用存储后端的管理软件确认卷大小的更改。

管理应用程序执行挂钩

执行挂钩是一种自定义操作、您可以将其配置为与受管应用程序的数据保护操作结合运行。例如，如果您有一个数据库应用程序，则可以使用执行挂钩在快照之前暂停所有数据库事务，并在快照完成后恢复事务。这样可以确保应用程序一致的快照。

执行挂钩的类型

Astra Control支持以下类型的执行挂钩、具体取决于何时可以运行：

- 预快照
- 快照后
- 预备份
- 备份后
- 还原后

有关自定义执行挂钩的重要注意事项

在为应用程序规划执行挂钩时，请考虑以下几点。

- 执行挂钩必须使用脚本执行操作。许多执行挂钩可以引用同一个脚本。
- Astra Control要求执行挂钩使用的脚本以可执行Shell脚本的格式写入。
- 脚本大小限制为96 KB。
- Astra Control使用执行挂钩设置和任何匹配条件来确定哪些挂钩适用于快照、备份或还原操作。
- 所有执行挂钩故障均为软故障；即使某个挂钩发生故障、仍会尝试执行其他挂钩和数据保护操作。但是，如果挂机发生故障，则会在 * 活动 * 页面事件日志中记录一个警告事件。
- 要创建，编辑或删除执行挂钩，您必须是具有所有者，管理员或成员权限的用户。
- 如果执行挂机运行时间超过 25 分钟，则此挂机将失败，从而创建返回代码为不适用的事件日志条目。任何受影响的快照都将超时并标记为失败，并会生成一个事件日志条目，用于记录超时情况。
- 对于临时数据保护操作、所有挂机事件都会生成并保存在*活动*页面事件日志中。但是、对于计划的数据保

护操作、事件日志中仅会记录挂钩故障事件(计划的数据保护操作本身生成的事件仍会记录下来)。



由于执行挂钩通常会减少或完全禁用其所运行的应用程序的功能，因此您应始终尽量缩短自定义执行挂钩运行所需的时间。如果使用关联的执行挂钩启动备份或快照操作、但随后将其取消、则在备份或快照操作已开始时、仍允许运行这些挂钩。这意味着、备份后执行挂钩不能假定备份已完成。

执行顺序

运行数据保护操作时、执行钩事件按以下顺序发生：

1. 任何适用的自定义操作前执行挂钩都会在相应的容器上运行。您可以根据需要创建和运行任意数量的自定义操作前挂钩、但操作前这些挂钩的执行顺序既不能保证也不可配置。
2. 执行数据保护操作。
3. 任何适用的自定义操作后执行挂钩都会在相应的容器上运行。您可以根据需要创建和运行任意数量的自定义操作后挂机、但这些挂机在操作后的执行顺序既不能保证也不可配置。

如果创建多个相同类型的执行挂钩(例如、预快照)、则无法保证这些挂钩的执行顺序。但是、可以保证不同类型的挂钩的执行顺序。例如、具有所有五种不同类型的挂钩的配置的执行顺序如下所示：

1. 已执行备份前的挂钩
2. 已执行预快照挂钩
3. 已执行后快照挂钩
4. 已执行备份后挂钩
5. 已执行还原后挂机

您可以从中的表中的第2种情形中查看此配置的示例 [\[确定挂钩是否会运行\]](#)。



在生产环境中启用执行钩脚本之前，应始终对其进行测试。您可以使用 "kubectl exec" 命令方便地测试脚本。在生产环境中启用执行挂钩后、请测试生成的快照和备份、以确保它们一致。为此、您可以将应用程序克隆到临时命名空间、还原快照或备份、然后测试应用程序。

确定挂钩是否会运行

使用下表帮助确定是否会为您的应用程序运行自定义执行挂钩。

请注意、所有高级应用程序操作都包括运行快照、备份或还原的基本操作之一。根据具体情况、克隆操作可能由这些操作的各种组合组成、因此克隆操作运行时的执行挂钩将会有所不同。

原位还原操作需要现有快照或备份、因此这些操作不会运行快照或备份挂钩。

如果启动并取消包含快照的备份、并且存在关联的执行挂钩、则某些挂钩可能会运行、而其他挂钩则可能不会运行。这意味着、备份后执行挂钩不能假定备份已完成。对于已取消的备份以及关联的执行挂钩、请记住以下几点：



- 备份前和备份后的挂钩始终处于运行状态。
- 如果备份包含新快照且快照已启动、则会运行预快照和后快照挂钩。
- 如果在快照启动之前取消了备份、则不会运行预快照和后快照挂钩。

场景	操作	现有快照	现有备份	命名空间	集群	快照挂钩运行	备份挂钩运行	Restore Hooks run
1.	克隆	不包括	不包括	新增	相同	Y	不包括	Y
2.	克隆	不包括	不包括	新增	不同	Y	Y	Y
3.	克隆或还原	Y	不包括	新增	相同	不包括	不包括	Y
4.	克隆或还原	不包括	Y	新增	相同	不包括	不包括	Y
5.	克隆或还原	Y	不包括	新增	不同	不包括	Y	Y
6.	克隆或还原	不包括	Y	新增	不同	不包括	不包括	Y
7.	还原	Y	不包括	现有	相同	不包括	不包括	Y
8.	还原	不包括	Y	现有	相同	不包括	不包括	Y
9	Snapshot	不适用	不适用	不适用	不适用	Y	不适用	不适用
10	备份	不包括	不适用	不适用	不适用	Y	Y	不适用
11.	备份	Y	不适用	不适用	不适用	不包括	Y	不适用

查看现有执行挂钩

您可以查看应用程序的现有自定义执行挂钩。

步骤

1. 转到 * 应用程序 * ，然后选择受管应用程序的名称。
2. 选择 * 执行挂钩 * 选项卡。

您可以在显示的列表中查看所有已启用或已禁用的执行挂钩。您可以查看挂机的状态、源以及运行时间(操作前或操作后)。要查看与执行挂钩相关的事件日志，请转到左侧导航区域中的 * 活动 * 页面。

查看现有脚本

您可以查看已上传的现有脚本。您还可以在此页面上查看正在使用哪些脚本以及正在使用哪些挂钩。

步骤

1. 转到*帐户*。

2. 选择*脚本*选项卡。

您可以在此页面上查看已上传的现有脚本列表。*使用者*列显示了使用每个脚本的执行挂钩。

添加脚本

您可以添加一个或多个可供执行挂钩引用的脚本。许多执行挂钩可以引用同一个脚本；这样、您就可以通过仅更改一个脚本来更新多个执行挂钩。

步骤

1. 转到*帐户*。
2. 选择*脚本*选项卡。
3. 选择 * 添加 *。
4. 执行以下操作之一：
 - 上传自定义脚本。
 - i. 选择 * 上传文件 * 选项。
 - ii. 浏览到文件并上传。
 - iii. 为脚本指定一个唯一名称。
 - iv. （可选）输入其他管理员应了解的有关该脚本的任何注释。
 - v. 选择*保存脚本*。
 - 从剪贴板粘贴到自定义脚本中。
 - i. 选择*粘贴或类型*选项。
 - ii. 选择文本字段并将脚本文本粘贴到字段中。
 - iii. 为脚本指定一个唯一名称。
 - iv. （可选）输入其他管理员应了解的有关该脚本的任何注释。
5. 选择*保存脚本*。

结果

新脚本将显示在*脚本*选项卡的列表中。

删除脚本

如果不再需要某个脚本、并且任何执行挂钩都不使用该脚本、则可以将其从系统中删除。

步骤

1. 转到*帐户*。
2. 选择*脚本*选项卡。
3. 选择要删除的脚本、然后在*操作*列中选择菜单。
4. 选择 * 删除 *。



如果该脚本与一个或多个执行挂钩关联、则*删除*操作将不可用。要删除此脚本、请先编辑关联的执行挂钩、然后将其与其他脚本关联。

创建自定义执行挂钩

您可以为应用程序创建自定义执行挂钩。请参见 ["执行钩示例"](#) 有关挂机示例。要创建执行挂钩，您需要拥有所有者，管理员或成员权限。



创建用作执行挂钩的自定义Shell脚本时、请务必在文件开头指定适当的Shell、除非您正在运行特定命令或提供可执行文件的完整路径。

步骤

1. 选择 * 应用程序 * ，然后选择受管应用程序的名称。
2. 选择 * 执行挂钩 * 选项卡。
3. 选择 * 添加 * 。
4. 在*挂机详细信息*区域中、通过从*操作*下拉菜单中选择操作类型来确定挂机应在何时运行。
5. 输入此挂钩的唯一名称。
6. （可选）输入执行期间传递到挂机的任何参数，在输入的每个参数之后按 Enter 键以记录每个参数。
7. 在 * 容器映像 * 区域中，如果此挂钩应针对应用程序中包含的所有容器映像运行，请启用 * 应用于所有容器映像 * 复选框。如果该挂钩只能作用于一个或多个指定的容器映像，请在 * 要匹配的容器映像名称 * 字段中输入容器映像名称。
8. 在 * 脚本 * 区域中，执行以下操作之一：
 - 添加新脚本。
 - i. 选择 * 添加 * 。
 - ii. 执行以下操作之一：
 - 上传自定义脚本。
 - I. 选择 * 上传文件 * 选项。
 - II. 浏览到文件并上传。
 - III. 为脚本指定一个唯一名称。
 - IV. （可选）输入其他管理员应了解的有关该脚本的任何注释。
 - V. 选择*保存脚本*。
 - 从剪贴板粘贴到自定义脚本中。
 - I. 选择*粘贴或类型*选项。
 - II. 选择文本字段并将脚本文本粘贴到字段中。
 - III. 为脚本指定一个唯一名称。
 - IV. （可选）输入其他管理员应了解的有关该脚本的任何注释。
 - 从列表选择一个现有脚本。

这将指示执行挂钩使用此脚本。

9. 选择 * 添加挂钩 *。

检查执行挂钩的状态

在快照、备份或还原操作运行完毕后、您可以检查在该操作中运行的执行挂钩的状态。您可以使用此状态信息来确定是要保持执行状态、修改执行状态还是删除执行状态。

步骤

1. 选择 * 应用程序 *，然后选择受管应用程序的名称。
2. 选择*数据保护*选项卡。
3. 选择*快照*可查看正在运行的快照、选择*备份*可查看正在运行的备份。

*挂机状态*显示操作完成后执行挂机运行的状态。有关详细信息、可以将鼠标悬停在状态上。例如、如果在快照期间发生执行挂机故障、则将鼠标悬停在该快照的挂机状态上可显示失败的执行挂机列表。要查看每次失败的原因、您可以查看左侧导航区域中的*活动*页面。

查看脚本使用情况

您可以在Astra Control Web UI中查看哪些执行挂钩使用特定脚本。

步骤

1. 选择 * 帐户 *。
2. 选择*脚本*选项卡。

脚本列表中的*使用者*列包含有关列表中每个脚本使用哪些挂钩的详细信息。

3. 在*使用者*列中选择您感兴趣的脚本的信息。

此时将显示一个更详细的列表、其中包含正在使用此脚本的挂钩的名称以及这些挂钩配置为运行的操作类型。

禁用执行挂钩

如果要暂时阻止执行挂钩在应用程序快照之前或之后运行，可以禁用执行挂钩。要禁用执行挂钩，您需要拥有所有者，管理员或成员权限。

步骤

1. 选择 * 应用程序 *，然后选择受管应用程序的名称。
2. 选择 * 执行挂钩 * 选项卡。
3. 在 * 操作 * 列中选择要禁用的挂机的选项菜单。
4. 选择 * 禁用 *。

删除执行挂钩

如果您不再需要执行挂钩，则可以将其完全移除。要删除执行挂钩，您需要拥有所有者，管理员或成员权限。

步骤

1. 选择 * 应用程序 * ，然后选择受管应用程序的名称。
2. 选择 * 执行挂钩 * 选项卡。
3. 在 * 操作 * 列中选择要删除的挂机的选项菜单。
4. 选择 * 删除 * 。

执行钩示例

使用以下示例了解如何构建执行挂钩。您可以将这些挂钩用作模板或测试脚本。

简单的成功示例

这是一个简单的钩子示例，它成功地将消息写入标准输出和标准错误。

```
#!/bin/sh

# success_sample.sh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
```

```

# $* - The message to write
#
error() {
    msg "ERROR: $" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.sh"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

简单成功示例（**bash** 版本）

这是一个简单的钩子示例，该钩子成功地将消息写入标准输出和标准错误，并写入 **bash**。

```

#!/bin/bash

# success_sample.bash
#
# A simple noop success hook script for testing purposes.
#
# args: None

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#

```

```

info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.bash"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

简单成功示例（zsh 版本）

这是一个简单的钩子示例，该钩子成功地将消息写入标准输出和标准错误，并写入 Z shell。

```

#!/bin/zsh

# success_sample.zsh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

```

```

}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.zsh"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

成功使用参数示例

以下示例演示了如何在挂机中使用 args 。

```

#!/bin/sh

# success_sample_args.sh
#
# A simple success hook script with args for testing purposes.
#
# args: Up to two optional args that are echoed to stdout
#
# Writes the given message to standard output

```

```

#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample_args.sh"

# collect args
arg1=$1
arg2=$2

# output args and arg count to stdout
info "number of args: $#"
```

```

info "arg1 ${arg1}"
info "arg2 ${arg2}"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

以下示例演示了如何对快照前和快照后挂钩使用同一脚本。

```
#!/bin/sh

# success_sample_pre_post.sh
#
# A simple success hook script example with an arg for testing purposes
# to demonstrate how the same script can be used for both a prehook and
# posthook
#
# args: [pre|post]

# unique error codes for every error case
ebase=100
eusage=$((ebase+1))
ebadstage=$((ebase+2))
epre=$((ebase+3))
epost=$((ebase+4))

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
```

```

error() {
    msg "ERROR: $" 1>&2
}

#
# Would run prehook steps here
#
prehook() {
    info "Running noop prehook"
    return 0
}

#
# Would run posthook steps here
#
posthook() {
    info "Running noop posthook"
    return 0
}

#
# main
#

# check arg
stage=$1
if [ -z "${stage}" ]; then
    echo "Usage: $0 <pre|post>"
    exit ${eusage}
fi

if [ "${stage}" != "pre" ] && [ "${stage}" != "post" ]; then
    echo "Invalid arg: ${stage}"
    exit ${ebadstage}
fi

# log something to stdout
info "running success_sample_pre_post.sh"

if [ "${stage}" = "pre" ]; then
    prehook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during prehook"
    fi
fi

```

```

        fi
    fi

    if [ "${stage}" = "post" ]; then
        posthook
        rc=$?
        if [ ${rc} -ne 0 ]; then
            error "Error during posthook"
        fi
    fi
fi

exit ${rc}

```

故障示例

以下示例演示了如何处理挂机故障。

```

#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

```



```

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}

```

详细故障示例

以下示例演示了如何处理挂机故障，并提供更详细的日志记录。

```

#!/bin/sh

# failure_sample_verbose.sh
#
# A simple failure hook script with args for testing purposes.
#
# args: [The number of lines to output to stdout]

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

```

```

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_verbose.sh"

# output arg value to stdout
linecount=$1
info "line count ${linecount}"

# write out a line to stdout based on line count arg
i=1
while [ "$i" -le ${linecount} ]; do
    info "This is line ${i} from failure_sample_verbose.sh"
    i=$(( i + 1 ))
done

error "exiting with error code 8"
exit 8

```

退出代码示例失败

以下示例显示了一个连接失败并显示退出代码。

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"
```

```
argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

失败后成功示例

以下示例显示了首次运行时发生故障的挂钩，但在第二次运行后仍会成功。

```
#!/bin/sh

# failure_then_success_sample.sh
#
# A hook script that fails on initial run but succeeds on second run for
# testing purposes.
#
# Helpful for testing retry logic for post hooks.
#
# args: None
#
#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
```

```
# $* - The message to write
#
error() {
    msg "ERROR: $" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_success sample.sh"

if [ -e /tmp/hook-test.junk ] ; then
    info "File does exist. Removing /tmp/hook-test.junk"
    rm /tmp/hook-test.junk
    info "Second run so returning exit code 0"
    exit 0
else
    info "File does not exist. Creating /tmp/hook-test.junk"
    echo "test" > /tmp/hook-test.junk
    error "Failed first run, returning exit code 5"
    exit 5
fi
```

监控应用程序和集群运行状况

查看应用程序和集群运行状况摘要

选择 *** 信息板 *** 可查看应用程序，集群，存储后端及其运行状况的高级视图。

这些数字或状态不仅仅是静态数字或状态，您可以逐层查看。例如，如果应用程序未得到完全保护，您可以将鼠标悬停在图标上以确定哪些应用程序未得到完全保护，这包括原因。

应用程序区块

"* 应用程序 * 图块可帮助您确定以下内容：

- 您当前使用 Astra 管理的应用程序数量。
- 这些受管应用程序是否运行正常。
- 应用程序是否受到完全保护（如果有最新备份可用，则会对其进行保护）。
- 已发现但尚未管理的应用程序的数量。

理想情况下，此数字为零，因为您可能会在发现应用程序后对其进行管理或忽略。然后，您将监控信息板上发现的应用程序的数量，以确定开发人员何时向集群添加新应用程序。

集群图块

"* 集群 *" 图块提供了有关使用 Astra 控制中心管理的集群运行状况的类似详细信息，您可以像使用应用程序一样深入查看以获取更多详细信息。

存储后端图块

"Storage Backends*" 图块提供的信息可帮助您确定存储后端的运行状况，其中包括：

- 管理的存储后端数量
- 这些受管后端是否运行正常
- 后端是否受到完全保护
- 已发现但尚未管理的后端数量。

查看集群的运行状况和详细信息

添加要由 Astra 控制中心管理的集群后，您可以查看有关集群的详细信息，例如集群的位置，工作节点，永久性卷和存储类。

步骤

1. 在 Astra 控制中心 UI 中，选择 * 集群 *。
2. 在 * 集群 * 页面上，选择要查看其详细信息的集群。



如果集群位于中 `removed` 状态虽然集群和网络连接运行状况良好(外部尝试使用Kubernetes API访问集群成功)、但您提供给Astra Control的kubeconfig可能不再有效。这可能是由于集群上的证书轮换或到期造成的。要更正此问题描述，请使用在 Astra Control 中更新与集群关联的凭据 ["Astra Control API"](#)。

3. 查看 * 概述 *，* 存储 * 和 * 活动 * 选项卡上的信息，找到您要查找的信息。
 - * 概述 *：有关工作节点的详细信息，包括其状态。
 - * 存储 *：与计算关联的永久性卷，包括存储类和状态。
 - * 活动 *：显示与集群相关的活动。



您还可以从 Astra 控制中心 * 信息板 * 开始查看集群信息。在 * 资源摘要 * 下的 * 集群 * 选项卡上，您可以选择受管集群，此操作将转到 * 集群 * 页面。进入 * 集群 * 页面后，请按照上述步骤进行操作。

查看应用程序的运行状况和详细信息

开始管理某个应用程序后，Astra 会提供有关该应用程序的详细信息，您可以通过这些详细信息来确定其状态（是否运行正常），保护状态（是否在发生故障时受到全面保护），Pod，永久性存储等。

步骤

1. 在 Astra 控制中心 UI 中，选择 * 应用程序 *，然后选择应用程序的名称。
2. 查找您需要的信息：

应用程序状态

提供反映应用程序在 Kubernetes 中的状态的状态。例如，Pod 和永久性卷是否联机？如果某个应用程序运行状况不正常，您需要查看 Kubernetes 日志，对集群上的问题描述进行故障排除。Astra 不会提供任何信息来帮助您修复损坏的应用程序。

应用程序保护状态

提供应用程序受保护程度的状态：

- * 完全保护 *：应用程序具有一个活动备份计划，并且备份成功完成不到一周
- * 部分保护 *：应用程序具有活动备份计划，活动快照计划或成功备份或快照
- * 未受保护 *：既不受完全保护也不受部分保护的应用程序。

You can't be Fully protected until you have a recent backup。这一点非常重要，因为备份存储在对象存储中，而不是永久性卷。如果发生故障或意外事件会擦除集群及其永久性存储，则需要备份才能恢复。快照无法让您恢复。

概述

与应用程序关联的 Pod 的状态信息。

数据保护

用于配置数据保护策略以及查看现有快照和备份。

存储

显示应用程序级别的永久性卷。从 Kubernetes 集群的角度来看，永久性卷的状态。

Resources

用于验证正在备份和管理哪些资源。

活动

显示了与应用程序相关的活动。



您还可以从 Astra 控制中心 * 信息板 * 开始查看应用程序信息。在 * 资源摘要 * 下的 * 应用程序 * 选项卡上，您可以选择受管应用程序，此操作将转到 * 应用程序 * 页面。进入 * 应用程序 * 页面后，请按照上述步骤进行操作。

管理您的帐户

管理用户

您可以使用 Astra Control UI 邀请，添加，删除和编辑 Astra Control Center 安装的用户。您可以使用 Astra Control UI 或 "[Astra Control API](#)" 以管理用户。

您还可以使用LDAP对选定用户执行身份验证。

使用 LDAP

LDAP是一种用于访问分布式目录信息的行业标准协议、也是企业身份验证的常见选择。您可以将Astra控制中心连接到LDAP服务器、以便对选定的Astra用户执行身份验证。从较高层面来看、该配置涉及将Astra与LDAP集成、并定义与LDAP定义对应的Astra用户和组。请参见 ["LDAP 身份验证"](#) 有关详细信息 ...

邀请用户

客户所有者和管理员可以邀请新用户访问 Astra 控制中心。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择 * 用户 * 选项卡。
3. 选择 * 邀请用户 *。
4. 输入用户的名称和电子邮件地址。
5. 选择具有适当系统权限的用户角色。

每个角色都提供以下权限：

- * 查看器 * 可以查看资源。
 - " 成员 * " 具有 " 查看器 " 角色权限，可以管理应用程序和集群，取消管理应用程序以及删除快照和备份。
 - * 管理员 * 具有成员角色权限，可以添加和删除除所有者之外的任何其他用户。
 - * 所有者 * 具有管理员角色权限，可以添加和删除任何用户帐户。
6. 要为具有成员或查看器角色的用户添加约束，请启用 * 将角色限制为约束条件 * 复选框。

有关添加约束的详细信息，请参见 ["管理角色"](#)。

7. 选择 * 邀请用户 *。

用户会收到一封电子邮件，告知他们已受邀访问 Astra 控制中心。此电子邮件包含临时密码，需要在首次登录时更改此密码。

添加用户

帐户所有者和管理员可以向 Astra 控制中心安装添加更多用户。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择 * 用户 * 选项卡。
3. 选择 * 添加用户 *。
4. 输入用户的名称，电子邮件地址和临时密码。

用户需要在首次登录时更改密码。

5. 选择具有适当系统权限的用户角色。

每个角色都提供以下权限：

- * 查看器 * 可以查看资源。
- " 成员 * " 具有 " 查看器 " 角色权限，可以管理应用程序和集群，取消管理应用程序以及删除快照和备份。
- * 管理员 * 具有成员角色权限，可以添加和删除除所有者之外的任何其他用户。
- * 所有者 * 具有管理员角色权限，可以添加和删除任何用户帐户。

6. 要为具有成员或查看器角色的用户添加约束，请启用 * 将角色限制为约束条件 * 复选框。

有关添加约束的详细信息，请参见 ["管理角色"](#)。

7. 选择 * 添加 * 。

管理密码

您可以在 Astra 控制中心管理用户帐户的密码。

更改密码

您可以随时更改用户帐户的密码。

步骤

1. 选择屏幕右上角的用户图标。
2. 选择 * 配置文件 * 。
3. 从选项菜单的 * 操作 * 列中选择 * 更改密码 * 。
4. 输入符合密码要求的密码。
5. 再次输入密码进行确认。
6. 选择 * 更改密码 * 。

重置其他用户的密码

如果您的帐户具有管理员或所有者角色权限，则可以重置其他用户帐户以及您自己的帐户的密码。重置密码时，您需要分配一个临时密码，用户必须在登录时更改此密码。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 * 。
2. 选择 * 操作 * 下拉列表。
3. 选择 * 重置密码 * 。
4. 输入符合密码要求的临时密码。
5. 再次输入密码进行确认。



用户下次登录时，系统将提示用户更改密码。

6. 选择 * 重置密码 *。

更改用户的角色

具有所有者角色的用户可以更改所有用户的角色，而具有管理员角色的用户可以更改具有管理员，成员或查看器角色的用户的角色。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择 * 操作 * 下拉列表。
3. 选择 * 编辑角色 *。
4. 选择一个新角色。
5. 要对角色应用约束，请启用 * 将角色限制为约束条件 * 复选框，然后从列表选择一个约束条件。

如果没有限制，您可以添加限制。有关详细信息，请参见 ["管理角色"](#)。

6. 选择 * 确认 *。

结果

Astra 控制中心会根据您选择的新角色更新用户的权限。

删除用户

具有所有者或管理员角色的用户可以随时从帐户中删除其他用户。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 在 * 用户 * 选项卡中，选中要删除的每个用户所在行中的复选框。
3. 从选项菜单的 * 操作 * 列中，选择 * 删除用户 / 秒 *。
4. 出现提示时，键入单词 "remove" 并选择 * 是，删除用户 * 以确认删除。

结果

Astra 控制中心从帐户中删除用户。

管理角色

您可以通过添加命名空间限制并将用户角色限制为这些限制来管理角色。这样，您就可以控制对组织内资源的访问。您可以使用 Astra Control UI 或 ["Astra Control API"](#) 以管理角色。

向角色添加命名空间限制

管理员或所有者用户可以添加命名空间约束。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择 * 用户 * 选项卡。

3. 在 * 操作 * 列中，为具有成员或查看器角色的用户选择菜单按钮。
4. 选择 * 编辑角色 *。
5. 启用 * 将角色限制为约束条件 * 复选框。

此复选框仅适用于 " 成员 " 或 " 查看器 " 角色。您可以从 * 角色 * 下拉列表中选择其他角色。

6. 选择 * 添加约束 *。

您可以按命名空间或命名空间标签查看可用约束的列表。

7. 在 * 约束类型 * 下拉列表中，根据命名空间的配置方式选择 * Kubernetes 命名空间 * 或 * Kubernetes 命名空间标签 *。
8. 从列表中选择一个或多个命名空间或标签，以构成一个限制，将角色限制为这些命名空间。
9. 选择 * 确认 *。

"* 编辑角色 *" 页面将显示您为此角色选择的约束列表。

10. 选择 * 确认 *。

在 * 帐户 * 页面上，您可以在 * 角色 * 列中查看任何成员或查看器角色的限制。



如果为某个角色启用了限制并选择了 * 确认 * 而未添加任何限制，则该角色将被视为具有完全限制（该角色将被拒绝访问分配给命名空间的任何资源）。

从角色中删除命名空间限制

管理员或所有者用户可以从角色中删除命名空间限制。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择 * 用户 * 选项卡。
3. 在 * 操作 * 列中，为具有成员或查看器角色且具有活动约束的用户选择菜单按钮。
4. 选择 * 编辑角色 *。

"* 编辑角色 " 对话框显示角色的活动约束。

5. 选择需要删除的约束右侧的 * X *。
6. 选择 * 确认 *。

有关详细信息 ...

- ["用户角色和命名空间"](#)

查看和管理通知

操作完成或失败时，Astra 会向您发出通知。例如，如果应用程序的备份成功完成，您将

看到通知。

您可以从界面右上角管理这些通知：



步骤

1. 选择右上角的未读通知数量。
2. 查看通知，然后选择 * 标记为已读 * 或 * 显示所有通知 *。

如果选择 * 显示所有通知 *，则会加载通知页面。

3. 在 * 通知 * 页面上，查看通知，选择要标记为已读的通知，选择 * 操作 * 并选择 * 标记为已读 *。

添加和删除凭据

随时从您的帐户中添加和删除本地私有云提供商的凭据，例如 ONTAP S3，使用 OpenShift 管理的 Kubernetes 集群或非受管 Kubernetes 集群。Astra 控制中心使用这些凭据来发现 Kubernetes 集群和集群上的应用程序，并代表您配置资源。

请注意，Astra 控制中心中的所有用户都共享相同的凭据集。

添加凭据

您可以在管理集群时向 Astra 控制中心添加凭据。要通过添加新集群来添加凭据，请参见 ["添加 Kubernetes 集群"](#)。



创建自己的 kubeconfig file 中、您只能定义 * 一 * 上下文元素。请参见 ["Kubernetes 文档"](#) 有关创建的信息 kubeconfig 文件。

删除凭据

随时从帐户中删除凭据。您只能在之后删除凭据 ["取消管理所有关联集群"](#)。



您添加到 Astra 控制中心的第一组凭据始终在使用中，因为 Astra 控制中心使用这些凭据向备份存储分段进行身份验证。最好不要删除这些凭据。

步骤

1. 选择 * 帐户 *。
2. 选择 * 凭据 * 选项卡。
3. 在 * 状态 * 列中选择要删除的凭据的选项菜单。
4. 选择 * 删除 *。
5. 键入单词 "remove" 确认删除，然后选择 * 是，删除凭据 *。

结果

Astra 控制中心将从帐户中删除凭据。

监控帐户活动

您可以在 Astra Control 帐户中查看有关活动的详细信息。例如，邀请新用户时，添加集群时或创建快照时。您还可以将帐户活动导出到 CSV 文件。



如果您从 Astra Control 管理 Kubernetes 集群、并且 Astra Control 连接到 Cloud Insights、则 Astra Control 会将事件日志发送到 Cloud Insights。日志信息(包括 POD 部署和 PVC 附件的相关信息)将显示在 Astra Control Activity 日志中。使用此信息确定您所管理的 Kubernetes 集群上的任何问题。

在 **Astra Control** 中查看所有帐户活动

1. 选择 * 活动 *。
2. 使用筛选器缩小活动列表的范围，或者使用搜索框准确查找所需内容。
3. 选择 * 导出到 CSV* 将您的帐户活动下载到 CSV 文件。

查看特定应用程序的帐户活动

1. 选择 * 应用程序 *，然后选择应用程序的名称。
2. 选择 * 活动 *。

查看集群的帐户活动

1. 选择 * 集群 *，然后选择集群的名称。
2. 选择 * 活动 *。

采取措施解决需要关注的事件

1. 选择 * 活动 *。
2. 选择需要关注的事件。
3. 选择 * 执行操作 * 下拉选项。

从此列表中，您可以查看可能采取的更正操作，查看与问题描述 相关的文档，并获得支持以帮助解决问题描述。

更新现有许可证

您可以将评估版许可证转换为完整许可证，也可以使用新许可证更新现有评估版许可证或完整许可证。如果您没有完整的许可证，请与 NetApp 销售联系人联系以获取完整的许可证和序列号。您可以使用 Astra UI 或 ["Astra Control API"](#) 更新现有许可证。

步骤

1. 登录到 ["NetApp 支持站点"](#)。
2. 访问 Astra 控制中心下载页面，输入序列号，然后下载完整的 NetApp 许可证文件（NLF）。
3. 登录到 Astra 控制中心 UI。
4. 从左侧导航栏中，选择 * 帐户 * > * 许可证 *。
5. 在 * 帐户 * > * 许可证 * 页面中，选择现有许可证的状态下拉菜单，然后选择 * 替换 *。

6. 浏览到您下载的许可证文件。
7. 选择 * 添加 *。
 - 帐户 * > * 许可证 * 页面显示许可证信息，到期日期，许可证序列号，帐户 ID 和使用的 CPU 单元。

有关详细信息 ...

- ["Astra 控制中心许可"](#)

管理存储库连接

您可以将存储库连接到Astra Control、以用作软件包安装映像和项目的参考。导入软件包时、Astra Control会引用映像存储库中的安装映像以及项目存储库中的二进制文件和其他项目。

您需要的内容

- 安装了 Astra 控制中心的 Kubernetes 集群
- 一个正在运行的Docker存储库、您可以访问该存储库
- 可访问的正在运行的项目存储库(如Artifactory)

连接Docker映像存储库

您可以连接Docker映像存储库以保存软件包安装映像、例如用于Astra数据存储的安装映像。安装软件包时、Astra Control会从映像存储库导入软件包映像文件。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择*连接*选项卡。
3. 在* Docker映像存储库*部分中、选择右上角的菜单。
4. 选择 * 连接 *。
5. 添加存储库的URL和端口。
6. 输入存储库的凭据。
7. 选择 * 连接 *。

结果

存储库已连接。在* Docker映像存储库*部分中、存储库应显示已连接状态。

断开Docker映像存储库的连接

如果不再需要与Docker映像存储库的连接、您可以将其删除。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择*连接*选项卡。
3. 在* Docker映像存储库*部分中、选择右上角的菜单。

4. 选择*断开连接*。
5. 选择*是、断开Docker映像存储库*。

结果

存储库已断开连接。在* Docker映像存储库*部分中、存储库应显示已断开连接状态。

连接项目存储库

您可以将项目存储库连接到主机项目、例如软件包二进制文件。安装软件包时、Astra Control会从映像存储库导入软件包的项目。

步骤

1. 在* 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择*连接*选项卡。
3. 在*项目存储库*部分中、选择右上角的菜单。
4. 选择 * 连接 *。
5. 添加存储库的URL和端口。
6. 如果需要身份验证、请启用*使用身份验证*复选框并输入存储库的凭据。
7. 选择 * 连接 *。

结果

存储库已连接。在*项目存储库*部分中、存储库应显示已连接状态。

断开项目存储库的连接

如果不再需要与项目存储库的连接、您可以将其删除。

步骤

1. 在* 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择*连接*选项卡。
3. 在*项目存储库*部分中、选择右上角的菜单。
4. 选择*断开连接*。
5. 选择*是、断开项目存储库*。

结果

存储库已断开连接。在*项目存储库*部分中、存储库应显示已连接状态。

了解更多信息

- ["管理软件包"](#)

管理软件包

NetApp 通过可从 NetApp 支持站点下载的软件包为 Astra Control Center 提供更多功能。连接Docker和项目存

储库后、您可以上传并导入软件包、以便将此功能添加到Astra控制中心。您可以使用命令行界面或Astra控制中心Web UI管理软件包。

您需要的内容

- 安装了 Astra 控制中心的 Kubernetes 集群
- 一个连接的Docker映像存储库、用于存放软件包映像。有关详细信息，请参见 ["管理存储库连接"](#)。
- 一个连接的项目存储库、用于存放软件包二进制文件和项目。有关详细信息，请参见 ["管理存储库连接"](#)。
- NetApp 支持站点提供的软件包

将软件包映像上传到存储库

Astra控制中心引用已连接存储库中的软件包映像和项目。您可以使用命令行界面将映像和项目上传到存储库。

步骤

1. 从NetApp 支持站点 下载软件包、并将其保存在具有的计算机上 `kubectl` 已安装实用程序。
2. 提取压缩的软件包文件、然后将目录更改为Astra Control软件包文件的位置(例如、`acc.manifest.yaml`)。
3. 将软件包映像推送到Docker存储库。进行以下替换：
 - 将`bundle_file`替换为Astra Control捆绑包文件的名称(例如、`acc.manifest.yaml`)。
 - 将`my_regRegistry`替换为Docker存储库的URL。
 - 将`my_registry_user`替换为用户名。
 - 将`my_registry_token`替换为注册表的授权令牌。

```
kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY -u MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
```

4. 如果软件包包含项目、请将这些项目复制到项目存储库。将`bundle_file`替换为Astra Control捆绑包文件的名称、将`network_location`替换为将项目文件复制到的网络位置：

```
kubectl astra packages copy-artifacts -m BUNDLE_FILE -n NETWORK_LOCATION
```

添加软件包

您可以使用Astra Control Center捆绑包文件导入软件包。这样将安装该软件包并使该软件可供Astra控制中心使用。

使用Astra Control Web UI添加软件包

您可以使用Astra控制中心Web UI添加已上传到已连接存储库的软件包。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。

2. 选择*软件包*选项卡。
3. 选择*添加*按钮。
4. 在文件选择对话框中、选择上传图标。
5. 在中选择Astra Control捆绑包文件 .yaml 格式、上传。
6. 选择 * 添加 *。

结果

如果捆绑包文件有效、并且软件包映像和项目位于已连接的存储库中、则软件包将添加到Astra控制中心。当*状态*列中的状态更改为*可用*时、您可以使用软件包。您可以将鼠标悬停在软件包的状态上以获取详细信息。



如果在存储库中未找到某个软件包的一个或多个映像或项目、则会显示该软件包的错误消息。

使用命令行界面添加软件包

您可以使用命令行界面导入已上传到已连接存储库的软件包。为此、您首先需要记录Astra控制中心帐户ID和API令牌。

步骤

1. 使用Web浏览器登录到Astra控制中心Web UI。
2. 从信息板中、选择右上角的用户图标。
3. 选择* API访问*。
4. 记下屏幕顶部附近的帐户ID。
5. 选择*生成API令牌*。
6. 在显示的对话框中、选择*生成API令牌*。
7. 记下生成的令牌、然后选择*关闭*。在CLI中、将目录更改为的位置 .yaml 提取的软件包内容中的软件包文件。
8. 使用捆绑包文件导入软件包、进行以下替换：
 - 将bundle_file替换为Astra Control捆绑包文件的名称。
 - 将Server替换为Astra Control实例的DNS名称。
 - 将account_id和token替换为先前记录的帐户ID和API令牌。

```
kubectl astra packages import -m BUNDLE_FILE -u SERVER -a ACCOUNT_ID  
-k TOKEN
```

结果

如果捆绑包文件有效、并且软件包映像和项目位于已连接的存储库中、则软件包将添加到Astra控制中心。



如果在存储库中未找到某个软件包的一个或多个映像或项目、则会显示该软件包的错误消息。

删除软件包

您可以使用Astra控制中心Web UI删除先前在Astra控制中心导入的软件包。

步骤

1. 在 * 管理帐户 * 导航区域中，选择 * 帐户 *。
2. 选择*软件包*选项卡。

您可以在此页面上查看已安装软件包的列表及其状态。

3. 在软件包的*操作*列中、打开操作菜单。
4. 选择 * 删除 *。

结果

该软件包将从Astra控制中心删除、但该软件包的映像和项目仍保留在存储库中。

了解更多信息

- ["管理存储库连接"](#)

管理存储分段

如果要备份应用程序和永久性存储，或者要跨集群克隆应用程序，则对象存储分段提供程序至关重要。使用Astra 控制中心，添加一个对象存储提供程序作为应用程序的集群外备份目标。

如果要应用程序配置和永久性存储克隆到同一集群，则不需要存储分段。

使用以下 Amazon Simple Storage Service （ S3 ） 存储分段提供商之一：

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Microsoft Azure
- 通用 S3



Amazon Web Services (AWS)和Google Cloud Platform (GCP)使用通用S3存储分段类型。



虽然 Astra 控制中心支持将 Amazon S3 作为通用 S3 存储分段提供商，但 Astra 控制中心可能不支持声称支持 Amazon S3 的所有对象存储供应商。

存储分段可以处于以下状态之一：

- Pending ： 存储分段已计划进行发现。
- Available ： 存储分段可供使用。
- Removed ： 当前无法访问此存储分段。

有关如何使用 Astra Control API 管理存储分段的说明，请参见 ["Astra Automation 和 API 信息"](#)。

您可以执行以下与管理存储分段相关的任务：

- ["添加存储分段"](#)
- [\[编辑存储分段\]](#)
- [\[轮换或删除存储分段凭据\]](#)
- [\[删除存储分段\]](#)



Astra 控制中心中的 S3 存储分段不会报告可用容量。在备份或克隆由 Astra 控制中心管理的应用程序之前，请检查 ONTAP 或 StorageGRID 管理系统中的存储分段信息。

编辑存储分段

您可以更改存储分段的访问凭据信息，并更改选定存储分段是否为默认存储分段。



添加存储分段时，请选择正确的存储分段提供程序，并为该提供程序提供正确的凭据。例如，UI 接受 NetApp ONTAP S3 作为类型并接受 StorageGRID 凭据；但是，这将发生原因使使用此存储分段执行所有未来应用程序备份和还原失败。请参见 ["发行说明"](#)。

步骤

1. 从左侧导航栏中、选择*分段*。
2. 从选项菜单的 * 操作 * 列中，选择 * 编辑 *。
3. 更改存储分段类型以外的任何信息。



您无法修改存储分段类型。

4. 选择 * 更新 *。

轮换或删除存储分段凭据

Astra Control使用存储分段凭据获取访问权限、并为S3存储分段提供机密密钥、以便Astra控制中心可以与存储分段进行通信。

轮换存储分段凭据

如果要轮换凭据、请在维护窗口中没有正在进行的备份(计划备份或按需备份)时轮换凭据。

编辑和轮换凭据的步骤

1. 从左侧导航栏中、选择*分段*。
2. 从选项菜单的 * 操作 * 列中，选择 * 编辑 *。
3. 创建新凭据。
4. 选择 * 更新 *。

删除存储分段凭据

只有在已将新凭据应用于存储分段或存储分段不再处于活动状态时、才应删除存储分段凭据。



添加到 Astra Control 的第一组凭据始终处于使用状态，因为 Astra Control 使用这些凭据对备份存储分段进行身份验证。如果存储分段正在使用中、请勿删除这些凭据、因为这会导致备份失败和备份不可用。



如果删除了活动存储分段凭据、请参见 ["对删除存储分段凭据进行故障排除"](#)。

有关如何使用 Astra Control API 删除 S3 凭据的说明、请参见 ["Astra Automation 和 API 信息"](#)。

删除存储分段

您可以删除不再使用或运行状况不佳的存储分段。您可能需要执行此操作以使对象存储配置简单且最新。



您不能删除默认存储分段。如果要删除此存储分段，请先选择另一个存储分段作为默认存储。

您需要的内容

- 开始之前，应检查以确保此存储分段没有正在运行或已完成的备份。
- 您应进行检查，以确保存储分段未在任何活动保护策略中使用。

如果存在，您将无法继续。

步骤

1. 从左侧导航栏中，选择 * 分段器 *。
2. 从 * 操作 * 菜单中，选择 * 删除 *。



Astra Control 可首先确保没有使用存储分段进行备份的计划策略，并且要删除的存储分段中没有活动备份。

3. 键入 "remove" 确认此操作。
4. 选择 * 是，删除存储分段 *。

了解更多信息

- ["使用 Astra Control API"](#)

管理存储后端

通过将 Astra Control 中的存储集群作为存储后端进行管理，您可以在永久性卷（PV）和存储后端之间建立链接，并获得其他存储指标。您可以监控存储容量和运行状况详细信息，包括当 Astra 控制中心连接到 Cloud Insights 时的性能。

有关如何使用 Astra Control API 管理存储后端的说明，请参见 ["Astra Automation 和 API 信息"](#)。

您可以完成以下与管理存储后端相关的任务：

- ["添加存储后端"](#)
- [\[查看存储后端详细信息\]](#)

- [\[取消管理存储后端\]](#)
- [更新Astra Data Store存储后端许可证](#)
- [升级Astra Data Store存储后端](#)
- [\[删除存储后端\]](#)
- [\[将节点添加到存储后端集群\]](#)
- [\[从存储后端集群中删除节点\]](#)

查看存储后端详细信息

您可以从信息板或后端选项查看存储后端信息。

在存储后端详细信息页面中、对于Astra数据存储、您可以看到以下信息：

- Astra数据存储集群
 - 吞吐量、IOPS和延迟
 - 已用容量与总容量之比
- 用于每个Astra Data Store集群卷
 - 已用容量与总容量之比
 - 吞吐量

从信息板查看存储后端详细信息

步骤

1. 从左侧导航栏中选择 * 信息板 *。
2. 查看存储后端部分，其中显示了以下状态：
 - * 运行状况不正常 *：存储未处于最佳状态。这可能是由于延迟问题描述或应用程序因容器问题描述等原因而降级。
 - * 所有运行状况均正常 *：存储已进行管理并处于最佳状态。
 - * 已发现 *：存储已被发现，但未由 Astra Control 管理。

从后端选项查看存储后端详细信息

查看有关后端运行状况，容量和性能（IOPS 吞吐量和 / 或延迟）的信息。

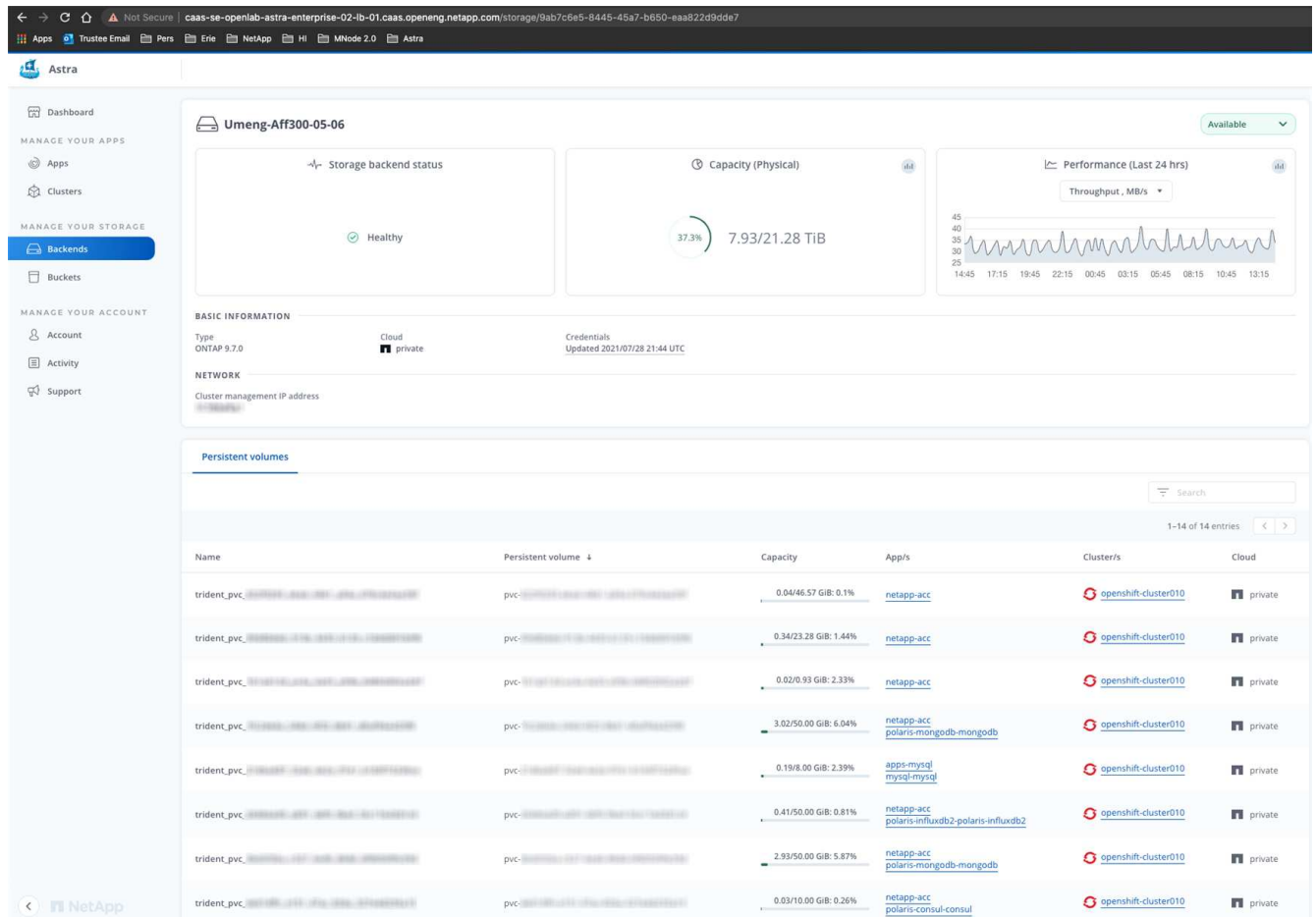
您可以查看Kubernetes应用程序正在使用的卷、这些卷存储在选定的存储后端。使用Cloud Insights 、您可以看到追加信息。请参见 ["Cloud Insights 文档"](#)。

步骤

1. 在左侧导航区域中，选择 * 后端 *。
2. 选择存储后端。



如果您连接到 NetApp Cloud Insights ，则 Cloud Insights 中的数据摘录将显示在后端页面上。



3. 要直接转到 Cloud Insights，请选择指标图像旁边的 * Cloud Insights * 图标。

取消管理存储后端

您可以取消管理后端。

步骤

1. 从左侧导航栏中，选择 * 后端 *。
2. 选择存储后端。
3. 从选项菜单的 * 操作 * 列中，选择 * 取消管理 *。
4. 键入 "unmanage" 确认此操作。
5. 选择 * 是，取消管理存储后端 *。

删除存储后端

您可以删除不再使用的存储后端。您可能需要执行此操作，以使您的配置简单且最新。



如果要删除 Astra Data Store 后端，则 vCenter 不能创建它。

您需要的内容

- 确保存储后端未受管。

- 确存储后端没有与 Astra Data Store 集群关联的任何卷。

步骤

1. 从左侧导航栏中，选择 * 后端 *。
2. 如果管理后端，请取消管理它。
 - a. 选择 * 受管 *。
 - b. 选择存储后端。
 - c. 从 * 操作 * 选项中，选择 * 取消管理 *。
 - d. 键入 "unmanage" 确认此操作。
 - e. 选择 * 是，取消管理存储后端 *。
3. 选择 * 已发现 *。
 - a. 选择存储后端。
 - b. 从 * 操作 * 选项中，选择 * 删除 *。
 - c. 键入 "remove" 确认此操作。
 - d. 选择 * 是，删除存储后端 *。

更新Astra Data Store存储后端许可证

您可以更新 Astra Data Store 存储后端的许可证，以支持更大规模的部署或增强功能。

您需要的内容

- 已部署和管理的 Astra Data Store 存储后端
- Astra Data Store 许可证文件（请联系您的 NetApp 销售代表以购买 Astra Data Store 许可证）

步骤

1. 从左侧导航栏中，选择 * 后端 *。
2. 选择存储后端的名称。
3. 在*基本信息*下、您可以看到安装的许可证类型。

如果将鼠标悬停在许可证信息上，则会显示一个弹出窗口，其中包含更多信息，例如到期时间和授权信息。

4. 在 * 许可证 * 下，选择许可证名称旁边的编辑图标。
5. 在*更新许可证*页面中、执行以下操作之一：

许可证状态	Action
至少已向Astra数据存储添加一个许可证。	从列表选择一个许可证。
尚未向Astra数据存储添加任何许可证。	<ol style="list-style-type: none"> a. 选择*添加*按钮。 b. 选择要上传的许可证文件。 c. 选择*添加*以上传许可证文件。

6. 选择 * 更新 *。

升级Astra Data Store存储后端

您可以从Astra控制中心升级Astra数据存储后端。为此、您必须先上传升级包；Astra控制中心将使用此升级包升级Astra数据存储。

您需要的内容

- 受管Astra Data Store存储后端
- 已上传Astra Data Store升级包(请参见 ["管理软件包"](#))

步骤

1. 选择*后端*。
2. 从列表中选择Astra Data Store存储后端、然后在*操作*列中选择相应的菜单。
3. 选择*升级*。
4. 从列表中选择升级版本。

如果存储库中有多个不同版本的升级包、您可以打开下拉列表以选择所需版本。

5. 选择 * 下一步 *。
6. 选择*开始升级*。

结果

在升级完成之前、*后端*页面的*状态*列将显示*正在升级*状态。

将节点添加到存储后端集群

您可以向 Astra Data Store 集群添加节点，最多可添加为 Astra Data Store 安装的许可证类型所支持的节点数。

您需要的内容

- 已部署并获得许可的 Astra Data Store 存储后端
- 您已在 Astra 控制中心中添加 Astra 数据存储软件包
- 要添加到集群的一个或多个新节点

步骤

1. 从左侧导航栏中，选择 * 后端 *。
2. 选择存储后端的名称。
3. 在 " 基本信息 " 下，您可以查看此存储后端集群中的节点数。
4. 在 * 节点 * 下，选择节点数旁边的编辑图标。
5. 在 * 添加节点 * 页面中，输入有关新节点的信息：
 - a. 为每个节点分配一个节点标签。
 - b. 执行以下操作之一：
 - 如果希望 Astra 数据存储始终根据您的许可证使用最大可用节点数，请启用 * 始终使用最多允许的最

大节点数 * 复选框。

- 如果您不希望 Astra 数据存储始终使用最大可用节点数，请选择所需的要使用的节点总数。

c. 如果您部署的 Astra 数据存储启用了保护域，请将新节点分配给保护域。

6. 选择 * 下一步 *。

7. 输入每个新节点的 IP 地址和网络信息。为一个新节点输入一个 IP 地址，为多个新节点输入一个 IP 地址池。

如果 Astra 数据存储可以使用部署期间配置的 IP 地址，则无需输入任何 IP 地址信息。

8. 选择 * 下一步 *。

9. 查看新节点的配置。

10. 选择 * 添加节点 *。

从存储后端集群中删除节点

您可以从 Astra 数据存储集群中删除节点。这些节点可以是运行状况良好的节点、也可以是发生故障的节点。

从 Astra 数据存储集群中删除节点会将其数据移至集群中的其他节点、并从 Astra 数据存储中删除此节点。

此过程需要满足以下条件：

- 其他节点必须有足够的可用空间来接收数据。
- 集群中必须有4个或更多节点。

步骤

1. 从左侧导航栏中，选择 * 后端 *。
2. 选择存储后端的名称。
3. 选择*节点*选项卡。
4. 从操作菜单中、选择*删除*。
5. 输入"remove"确认删除。
6. 选择*是、删除节点*。

了解更多信息

- ["使用 Astra Control API"](#)

使用Cloud Insights 和Fluentd连接监控基础架构

您可以配置多种可选设置来增强您的 Astra 控制中心体验。要监控和深入了解整个基础架构，请与 NetApp Cloud Insights 建立连接。要从 Astra 控制中心监控的系统收集 Kubernetes 事件，请添加 Fluentd 连接。

如果运行 Astra Control Center 的网络需要一个代理来连接到 Internet（将支持包上传到 NetApp 支持站点或建立与 Cloud Insights 的连接），则应在 Astra Control Center 中配置一个代理服务器。

您还可以从 Astra 控制中心存储后端页面监控 Astra Data Store 存储后端吞吐量、IOPS 和容量。请参见 ["管理存储"](#)

后端”。

添加代理服务器以连接到 **Cloud Insight** 或 **NetApp** 支持站点

如果运行 Astra Control Center 的网络需要一个代理来连接到 Internet（将支持包上传到 NetApp 支持站点或建立与 Cloud Insights 的连接），则应在 Astra Control Center 中配置一个代理服务器。



Astra 控制中心不会验证您为代理服务器输入的详细信息。请确保输入正确的值。

步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 *。
3. 从下拉列表中选择 * 连接 * 以添加代理服务器。



HTTP PROXY

Configure Astra Control to send traffic through a proxy server.

Disconnected



Connect

4. 输入代理服务器名称或 IP 地址以及代理端口号。
5. 如果代理服务器需要身份验证，请选中此复选框，然后输入用户名和密码。
6. 选择 * 连接 *。

结果

如果您输入的代理信息已保存，则 * 帐户 * > * 连接 * 页面的 * HTTP 代理 * 部分将指示它已连接，并显示服务器名称。



HTTP PROXY ?

Server: proxy.example.com:8888

Authentication: Enabled

Connected



编辑代理服务器设置

您可以编辑代理服务器设置。

步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 *。
3. 从下拉列表中选择 * 编辑 * 以编辑连接。

4. 编辑服务器详细信息和身份验证信息。
5. 选择 * 保存 *。

禁用代理服务器连接

您可以禁用代理服务器连接。在禁用之前，系统会警告您可能会对其他连接造成中断。

步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 *。
3. 从下拉列表中选择 * 断开连接 * 以禁用连接。
4. 在打开的对话框中，确认操作。

连接到 Cloud Insights

要监控和深入了解整个基础架构，请将 NetApp Cloud Insights 与您的 Astra 控制中心实例连接起来。Cloud Insights 包含在您的 Astra 控制中心许可证中。

Cloud Insights 应可从 Astra 控制中心使用的网络访问，也可通过代理服务器间接访问。

当 Astra 控制中心连接到 Cloud Insights 时，将创建采集单元 POD。此 POD 从由 Astra 控制中心管理的存储后端收集数据并将其推送到 Cloud Insights。此 POD 需要 8 GB RAM 和 2 个 CPU 核。

此外、如果您在 Astra Control (连接到 Cloud Insights) 上管理 Astra 数据存储集群、则会在 Astra 数据存储上为每个 Astra 数据存储集群创建一个采集单元 POD、并将指标从 Astra 数据存储发送到配对的 Cloud Insights 系统。每个 POD 需要 8 GB RAM 和 2 个 CPU 核。



启用 Cloud Insights 连接后，您可以在 * 后端 * 页面上查看吞吐量信息，并在选择存储后端后从此处连接到 Cloud Insights。您还可以在 "Cluster" 部分的 * 信息板 * 中找到相关信息，并从该处连接到 Cloud Insights。

您需要的内容

- 具有 * 管理 / 所有者 * 权限的 Astra 控制中心帐户。
- 有效的 Astra Control Center 许可证。
- 如果运行 Astra 控制中心的网络需要使用代理连接到 Internet，则为代理服务器。



如果您是 Cloud Insights 的新用户，请熟悉其特性和功能。请参见 "[Cloud Insights 文档](#)"。

步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 *。
3. 选择 * 连接 *，其中下拉列表中显示 * 已断开连接 * 以添加连接。

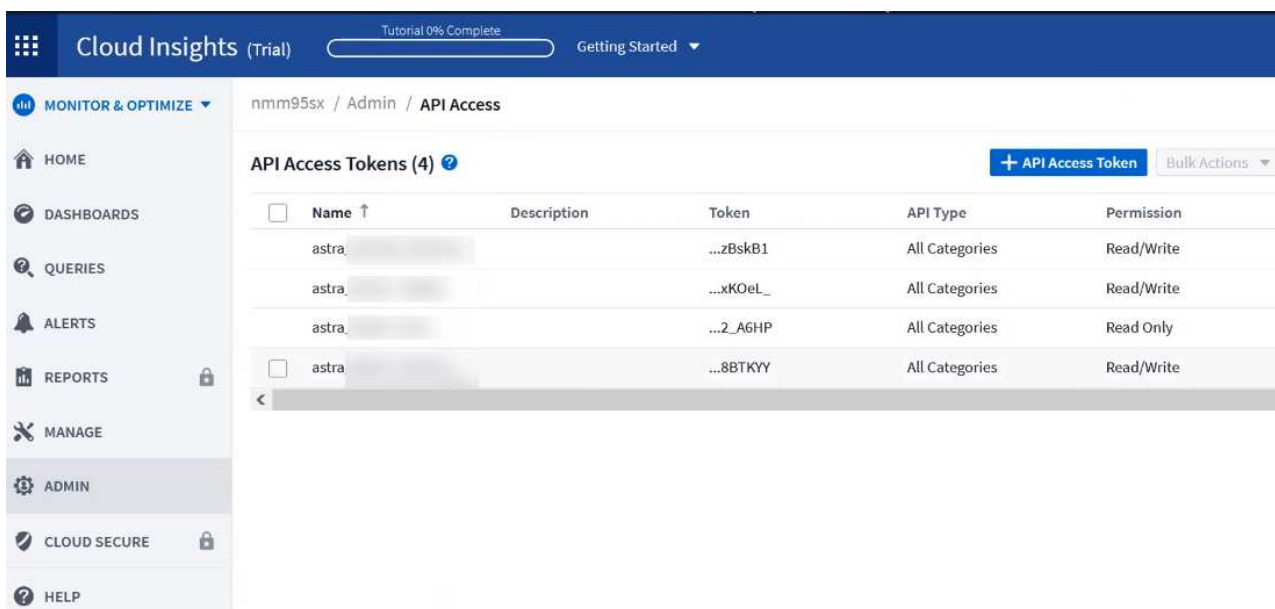


4. 输入 Cloud Insights API 令牌和租户 URL 。例如，租户 URL 采用以下格式：

```
https://<environment-name>.c01.cloudinsights.netapp.com/
```

获取 Cloud Insights 许可证后，您将获得租户 URL 。如果您没有租户 URL ，请参见 ["Cloud Insights 文档"](#)。

- a. 以获取 ["API 令牌"](#)，登录到您的 Cloud Insights 租户 URL 。
- b. 在 Cloud Insights 中，单击 **管理** > **API 访问** 以生成 **读 / 写** 和 **只读** API 访问令牌。



- c. 复制 **只读** 密钥。您需要将其粘贴到 Astra 控制中心窗口中以启用 Cloud Insights 连接。对于读取 API 访问令牌密钥权限，请选择：资产，警报，采集单元和数据收集。
- d. 复制 **读 / 写** 密钥。您需要将其粘贴到 Astra 控制中心 **连接 Cloud Insights** 窗口中。对于读 / 写 API 访问令牌密钥权限，请选择：Assets ， Data Ingestion ， Log ingestion ， Acquisition Unit ， 和数据收集。



建议您生成 **只读** 密钥和 **读 / 写** 密钥，不要将同一密钥用于这两种用途。默认情况下，令牌到期期限设置为一年。我们建议您保留默认选择，以便为令牌提供到期前的最长持续时间。如果令牌过期，遥测将停止。

- e. 将从 Cloud Insights 复制的密钥粘贴到 Astra 控制中心。

5. 选择 **连接** 。



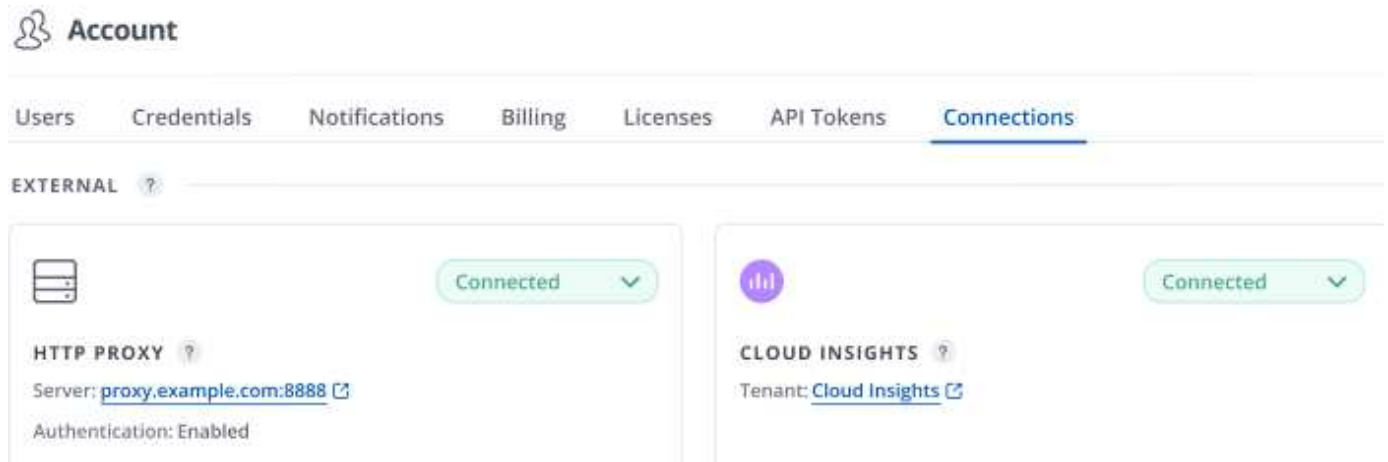
选择 * 连接后，在 Cloud Insights * 帐户 * > * 连接 * 页面的 * 连接 * 部分中，连接状态将更改为 * 待定 *。可以在几分钟内启用连接并将状态更改为 * 已连接 *。



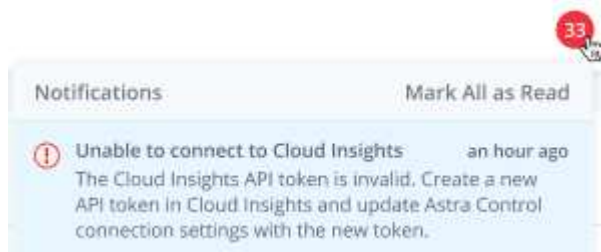
要在 Astra 控制中心和 Cloud Insights UI 之间轻松来回切换，请确保您已登录这两个。

在 Cloud Insights 中查看数据

如果连接成功，则 * 帐户 * > * 连接 * 页面的 * Cloud Insights * 部分将指示已连接，并显示租户 URL。您可以访问 Cloud Insights 以查看成功接收和显示的数据。



如果连接因某种原因失败，则状态将显示 * 失败 *。您可以在用户界面右上角的 * 通知 * 下找到失败的原因。



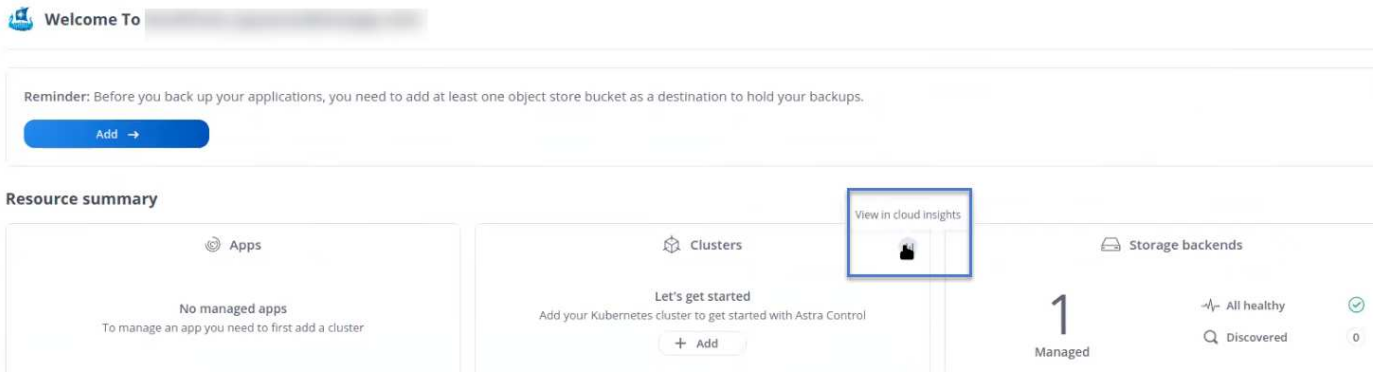
您还可以在 * 帐户 * > * 通知 * 下找到相同的信息。

在 Astra 控制中心中，您可以在 * 后端 * 页面上查看吞吐量信息，并在选择存储后端后从此处连接到 Cloud Insights



要直接转到 Cloud Insights，请选择指标图像旁边的 * Cloud Insights * 图标。

您还可以在 * 信息板 * 上找到相关信息。



启用 Cloud Insights 连接后，如果删除在 Astra 控制中心添加的后端，后端将停止向 Cloud Insights 报告。

编辑 Cloud Insights 连接

您可以编辑 Cloud Insights 连接。



您只能编辑 API 密钥。要更改 Cloud Insights 租户 URL，我们建议您断开 Cloud Insights 连接并使用新 URL 进行连接。

步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 *。
3. 从下拉列表中选择 * 编辑 * 以编辑连接。
4. 编辑 Cloud Insights 连接设置。
5. 选择 * 保存 *。

禁用 Cloud Insights 连接

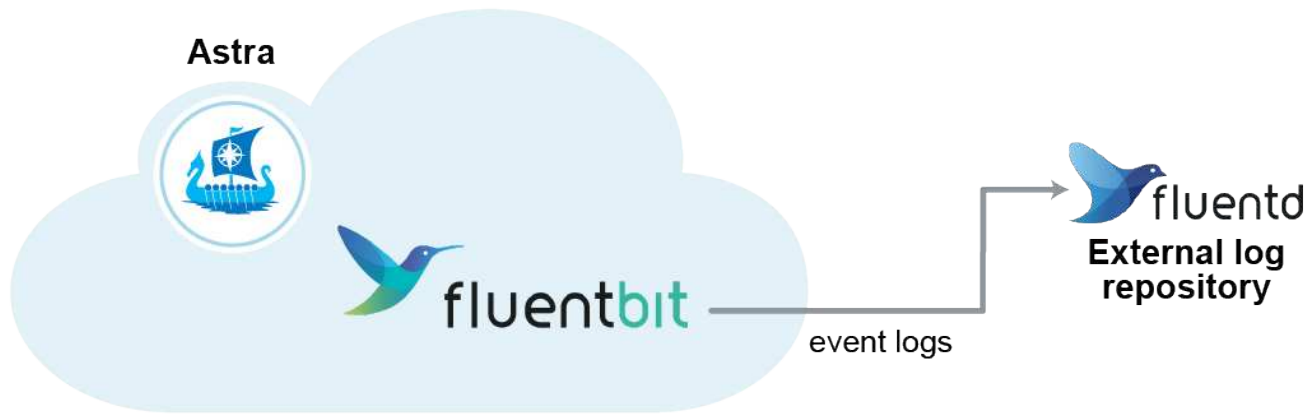
您可以为由 Astra 控制中心管理的 Kubernetes 集群禁用 Cloud Insights 连接。禁用 Cloud Insights 连接不会删除已上传到 Cloud Insights 的遥测数据。


步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 *。
3. 从下拉列表中选择 * 断开连接 * 以禁用连接。
4. 在打开的对话框中，确认操作。确认此操作后，在 * 帐户 * > * 连接 * 页面上，Cloud Insights 状态将更改为 * 待定 *。要将状态更改为 * 已断开连接 *，需要几分钟的时间。

连接到 Fluentd

您可以将日志（Kubernetes 事件）从 Astra 控制中心发送到 Fluentd 端点。默认情况下，Fluentd 连接处于禁用状态。



 只有受管集群中的事件日志才会转发到 Fluentd 。

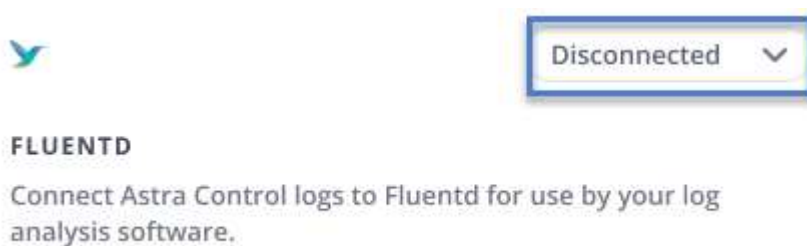
您需要的内容

- 具有 * 管理 / 所有者 * 权限的 Astra 控制中心帐户。
- 已在 Kubernetes 集群上安装并运行 Astra Control Center 。

 Astra 控制中心不会验证您为 Fluentd 服务器输入的详细信息。请确保输入正确的值。

步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 * 。
3. 从显示 * 已断开连接 * 的下拉列表中选择 * 连接 * 以添加连接。



4. 输入 Fluentd 服务器的主机 IP 地址，端口号和共享密钥。
5. 选择 * 连接 * 。

结果

如果您为 Fluentd 服务器输入的详细信息已保存，则 * 帐户 * > * 连接 * 页面的 * 流量 * 部分将指示它已连接。现在，您可以访问已连接的 Fluentd 服务器并查看事件日志。

如果连接因某种原因失败，则状态将显示 * 失败 * 。您可以在用户界面右上角的 * 通知 * 下找到失败的原因。

您还可以在 * 帐户 * > * 通知 * 下找到相同的信息。



如果您在收集日志时遇到问题、应登录到工作节点并确保日志在中可用
`/var/log/containers/`。

编辑 Fluentd 连接

您可以编辑与 Astra Control Center 实例的 Fluentd 连接。

步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 *。
3. 从下拉列表中选择 * 编辑 * 以编辑连接。
4. 更改 Fluentd 端点设置。
5. 选择 * 保存 *。

禁用 Fluentd 连接

您可以禁用与 Astra Control Center 实例的 Fluentd 连接。

步骤

1. 使用具有 * 管理员 / 所有者 * 权限的帐户登录到 Astra 控制中心。
2. 选择 * 帐户 * > * 连接 *。
3. 从下拉列表中选择 * 断开连接 * 以禁用连接。
4. 在打开的对话框中，确认操作。

取消管理应用程序和集群

从 Astra 控制中心删除不再需要管理的任何应用程序或集群。

取消管理应用程序

从 Astra 控制中心停止管理不再需要备份，快照或克隆的应用程序。

- 所有现有备份和快照都将被删除。
- 应用程序和数据始终可用。

步骤

1. 从左侧导航栏中，选择 * 应用程序 *。
2. 选中不再需要管理的应用程序对应的复选框。
3. 从 * 操作 * 菜单中，选择 * 取消管理 *。
4. 键入 "unmanage" 进行确认。
5. 确认要取消管理这些应用程序，然后选择 * 是，取消管理应用程序 *。

结果

Astra 控制中心停止管理应用程序。

取消管理集群

从 Astra 控制中心取消管理不再需要管理的集群。

- 此操作将停止由 Astra 控制中心管理集群。它不会对集群的配置进行任何更改，也不会删除集群。
- 不会从集群中卸载 Trident。 ["了解如何卸载 Trident"](#)。



在取消管理集群之前，您应取消管理与集群关联的应用程序。

步骤

1. 从左侧导航栏中，选择 * 集群 *。
2. 选中不再希望在 Astra 控制中心中管理的集群对应的复选框。
3. 从选项菜单的 * 操作 * 列中，选择 * 取消管理 *。
4. 确认要取消管理集群，然后选择 * 是，取消管理集群 *。

结果

集群状态将更改为 * 正在删除 *，之后，集群将从 * 集群 * 页面中删除，并且不再由 Astra 控制中心管理。



如果 Astra 控制中心和 Cloud Insights 未连接 *，则取消管理集群将删除为发送遥测数据而安装的所有资源。如果已连接 Astra 控制中心和 Cloud Insights *，则取消管理集群将仅删除 fluentbit 和 event-exporter POD。

升级 Astra 控制中心

要升级 Astra Control Center，请从 NetApp 支持站点下载安装包，然后按照以下说明升级环境中的 Astra Control Center 组件。您可以使用此操作步骤在互联网连接或通风环境中升级 Astra 控制中心。

您需要的内容

- ["开始升级之前，请确保您的环境仍满足 Astra Control Center 部署的最低要求"](#)。
- 确保所有集群操作员均处于运行状况良好且可用。

```
kubectl get clusteroperators
```

- 确保所有 API 服务均处于运行状况良好且可用。

```
kubectl get apiservices
```

- 从 Astra 控制中心注销。

关于此任务

Astra 控制中心升级过程将指导您完成以下高级步骤：

- [下载 Astra Control Center 捆绑包](#)
- [\[打开软件包的包装并更改目录\]](#)
- [\[将映像添加到本地注册表\]](#)
- [安装更新后的 Astra 控制中心操作员](#)
- [升级 Astra 控制中心](#)
- [\[升级第三方服务（可选）\]](#)
- [\[验证系统状态\]](#)
- [\[设置传入以进行负载平衡\]](#)



请勿在整个升级过程中执行以下命令、以避免删除所有Astra控制中心Pod： `kubectl delete -f astra_control_center_operator_deploy.yaml`



如果计划，备份和快照未运行，请在维护窗口中执行升级。



如果您使用的是 Red Hat 的 Podman 而不是 Docker 引擎，则可以使用 Podman 命令代替 Docker 命令。

下载 Astra Control Center 捆绑包

1. 下载Astra Control Center升级包 (`astra-control-center-[version].tar.gz`)^
mysupport.netapp.com/site/products/all/details/astra-control-center/downloads-tab[NetApp]。
2. （可选）使用以下命令验证捆绑包的签名：

```
openssl dgst -sha256 -verify AstraControlCenter-public.pub -signature
astra-control-center-[version].tar.gz.sig astra-control-center-
[version].tar.gz
```

打开软件包的包装并更改目录

1. 提取映像：

```
tar -vxzf astra-control-center-[version].tar.gz
```

将映像添加到本地注册表

1. 为容器引擎完成相应的步骤顺序：

Docker

1. 更改为Astra目录：

```
cd acc
```

2. [substep_image_local_registry_push]]将Astra控制中心映像目录中的软件包映像推送到本地注册表。
在运行命令之前、请执行以下替换操作：

- 将bundle_file替换为Astra Control捆绑包文件的名称(例如、acc.manifest.yaml)。
- 将my_regRegistry替换为Docker存储库的URL。
- 将my_registry_user替换为用户名。
- 将my_registry_token替换为注册表的授权令牌。

```
kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY  
-u MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
```

Podman

1. 登录到注册表：

```
podman login [your_registry_path]
```

2. 运行以下脚本、按照注释中的说明替换<your_registry>：

```
# You need to be at the root of the tarball.
# You should see these files to confirm correct location:
#   acc.manifest.yaml
#   acc/

# Replace <YOUR_REGISTRY> with your own registry (e.g
registry.customer.com or registry.customer.com/testing, etc..)
export REGISTRY=<YOUR_REGISTRY>
export PACKAGENAME=acc
export PACKAGEVERSION=22.08.1-26
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
    # Load to local cache
    astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //'')

    # Remove path and keep imageName.
    astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')

    # Tag with local image repo.
    podman tag ${astraImage} ${REGISTRY}/netapp/astra/${PACKAGENAME}
/${PACKAGEVERSION}/${astraImageNoPath}

    # Push to the local repo.
    podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

安装更新后的 **Astra** 控制中心操作员

1. 更改目录：

```
cd manifests
```

2. 编辑Astra控制中心操作员部署YAML (astra_control_center_operator_deploy.yaml)以引用您的本地注册表和密钥。

```
vim astra_control_center_operator_deploy.yaml
```

- a. 如果您使用的注册表需要身份验证、请替换的默认行 `imagePullSecrets: []` 使用以下命令：

```
imagePullSecrets:  
- name: <name_of_secret_with_creds_to_local_registry>
```

- b. 更改 [your_registry_path]。 kube-rbac-proxy 将映像推送到注册表路径中 [上一步](#)。
- c. 更改 [your_registry_path]。 acc-operator-controller-manager 将映像推送到注册表路径中 [上一步](#)。
- d. 将以下值添加到 env 部分。

```
- name: ACCOP_HELM_UPGRADE_TIMEOUT  
  value: 300m
```

```

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
        - args:
            - --secure-listen-address=0.0.0.0:8443
            - --upstream=http://127.0.0.1:8080/
            - --logtostderr=true
            - --v=10
          image: [your_registry_path]/kube-rbac-proxy:v4.8.0
          name: kube-rbac-proxy
          ports:
            - containerPort: 8443
              name: https
        - args:
            - --health-probe-bind-address=:8081
            - --metrics-bind-address=127.0.0.1:8080
            - --leader-elect
          command:
            - /manager
          env:
            - name: ACCOP_LOG_LEVEL
              value: "2"
            - name: ACCOP_HELM_UPGRADE_TIMEOUT
              value: 300m
          image: [your_registry_path]/acc-operator:[version x.y.z]
          imagePullPolicy: IfNotPresent
      imagePullSecrets: []

```

3. 安装更新后的 Astra 控制中心操作员：

```
kubectl apply -f astra_control_center_operator_deploy.yaml
```

响应示例：

```
namespace/netapp-acc-operator unchanged
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.astra.
netapp.io configured
role.rbac.authorization.k8s.io/acc-operator-leader-election-role
unchanged
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role
configured
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
unchanged
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role unchanged
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding unchanged
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding configured
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding unchanged
configmap/acc-operator-manager-config unchanged
service/acc-operator-controller-manager-metrics-service unchanged
deployment.apps/acc-operator-controller-manager configured
```

4. 验证Pod是否正在运行：

```
kubectl get pods -n netapp-acc-operator
```

升级 Astra 控制中心

1. 编辑Astra控制中心自定义资源(CR) (astra_control_center_min.yaml)并更改Astra版本 (astraVersion 在中 Spec)编号为最新值：

```
kubectl edit acc -n [netapp-acc or custom namespace]
```



注册表路径必须与中推送映像的注册表路径匹配 [上一步](#)。

2. 在中添加以下行 additionalValues 在中 Spec 在Astra控制中心CR中：

```
additionalValues:
  nautilus:
    startupProbe:
      periodSeconds: 30
      failureThreshold: 600
```

3. 执行以下操作之一：

- a. 如果您没有自己的IngressController或入口、并且一直使用带有Traefik网关的Astra控制中心作为负载均衡器类型的服务、并且希望继续进行此设置、请指定另一个字段 `ingressType` (如果尚不存在)并将其设置为 `AccTraefik`。

```
ingressType: AccTraefik
```

- b. 如果您要切换到默认的Astra控制中心通用传入部署、请提供您自己的IngressController/传入设置(采用TLS终止等)、打开指向Astra控制中心的路由并进行设置 `ingressType` to `Generic`。

```
ingressType: Generic
```



如果省略此字段，则此过程将成为通用部署。如果您不希望使用通用部署，请务必添加此字段。

4. (可选) 验证 Pod 是否终止并重新可用：

```
watch kubectl get po -n [netapp-acc or custom namespace]
```

5. 等待 Astra 状态条件指示升级已完成且准备就绪：

```
kubectl get -o yaml -n [netapp-acc or custom namespace]
astracontrolcenters.astra.netapp.io astra
```

响应：


```

conditions:
  - lastTransitionTime: "2021-10-25T18:49:26Z"
    message: Astra is deployed
    reason: Complete
    status: "True"
    type: Ready
  - lastTransitionTime: "2021-10-25T18:49:26Z"
    message: Upgrading succeeded.
    reason: Complete
    status: "False"
    type: Upgrading

```

6. 重新登录并验证所有受管集群和应用程序是否仍然存在并受到保护。
7. 如果操作员未更新证书管理器，请接下来升级第三方服务。

升级第三方服务（可选）

在先前的升级步骤中，不会升级第三方服务 Traefik 和 Cert-manager。您可以选择使用此处所述的操作步骤对其进行升级，也可以在系统需要时保留现有服务版本。

- *** 任务期限 ***：默认情况下，Astra 控制中心负责管理任务期限部署的生命周期。正在设置 ...
externalTraefik to false (默认值)表示系统中不存在外部Traefik、并且Astra控制中心正在安装和管理Traefik。在这种情况下、externalTraefik 设置为 false。

另一方面、如果您有自己的Traefik部署、请设置 externalTraefik to true。在这种情况下、您需要维护部署、并且Astra控制中心不会升级CRD、除非 shouldUpgrade 设置为 true。

- **证书管理器**：默认情况下、除非您设置、否则Astra控制中心将安装证书管理器(和CRD)
externalCertManager to true。设置 shouldUpgrade to true 要让Astra控制中心升级CRD。

如果满足以下任一条件，则升级 Traefik：

- externalTraefik: false
- externalTraefik : true , shouldUpgrade : true 。

步骤

1. 编辑 acc CR:

```
kubectl edit acc -n [netapp-acc or custom namespace]
```

2. 更改 externalTraefik 字段和 shouldUpgrade 字段中的任意一个 true 或 false 根据需要。

```
crds:
  externalTraefik: false
  externalCertManager: false
  shouldUpgrade: false
```

验证系统状态

1. 登录到 Astra 控制中心。
2. 验证所有受管集群和应用程序是否仍存在并受到保护。

设置传入以进行负载均衡

您可以设置 Kubernetes 入口对象，用于管理对服务的外部访问，例如集群中的负载均衡。

- 默认升级使用通用传入部署。在这种情况下，您还需要设置入口控制器或入口资源。
- 如果您不希望使用入口控制器、但希望保留现有控制器、请设置 `ingressType` to `AccTraefik`。



有关 "loadbalancer" 服务类型和入口的其他详细信息，请参见 ["要求"](#)。

根据您使用的入口控制器类型，步骤会有所不同：

- nginx 入口控制器
- OpenShift 入口控制器

您需要的内容

- 在 CR 规范中，
 - 条件 `crd.externalTraefik` 存在、应将其设置为 `false` 或
 - 条件 `crd.externalTraefik` 为 `true`，`crd.shouldUpgrade` 也应如此 `true`。
- 所需 ["入口控制器"](#) 应已部署。
- ["入口类"](#) 应已创建与入口控制器对应的。
- 您使用的是介于 v1.19 和 v1.21 之间的 Kubernetes 版本，包括 v1.19 和 v1.21 。

nginx 入口控制器的步骤

1. 使用现有密钥 `secure-testing-cert` 或者创建类型的密钥[kubernetes.io/tls]中的TLS专用密钥和证书 `netapp-acc` (或自定义命名的)命名空间、如中所述 ["TLS 密钥"](#)。
2. 在中部署传入资源 `netapp-acc` 已弃用或新模式的(或自定义命名的)命名空间：
 - a. 对于已弃用的模式，请遵循以下示例：

```
apiVersion: extensions/v1beta1
kind: IngressClass
metadata:
  name: ingress-acc
  namespace: [netapp-acc or custom namespace]
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  tls:
    - hosts:
        - <ACC address>
      secretName: [tls secret name]
  rules:
    - host: [ACC address]
      http:
        paths:
          - backend:
              serviceName: traefik
              servicePort: 80
            pathType: ImplementationSpecific
```

b. 对于新模式，请遵循以下示例：

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
      - path:
        backend:
          service:
            name: traefik
            port:
              number: 80
        pathType: ImplementationSpecific

```

OpenShift 入口控制器的步骤

1. 获取证书并获取密钥，证书和 CA 文件，以供 OpenShift 路由使用。
2. 创建 OpenShift 路由：

```

oc create route edge --service=traefik
--port=web -n [netapp-acc or custom namespace]
--insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem

```

验证入口设置

您可以先验证入口设置，然后再继续操作。

1. 确保已将Traefik更改为 clusterIP 从负载均衡器：

```

kubectl get service traefik -n [netapp-acc or custom namespace]

```

2. 验证 Traefik 中的路由：

```
kubectl get ingressroute ingressroutetls -n [netapp-acc or custom namespace]
-o yaml | grep "Host("
```



结果应为空。

卸载 Astra 控制中心

如果要从试用版升级到完整版本的产品，您可能需要删除 Astra Control Center 组件。要删除 Astra 控制中心和 Astra 控制中心操作员，请按顺序运行此操作步骤中所述的命令。

如果您在卸载时遇到任何问题，请参见 [\[对卸载问题进行故障排除\]](#)。

您需要的内容

- 使用 Astra 控制中心 UI 取消全部管理 **"集群"**。

步骤

1. 删除 Astra 控制中心。以下命令示例基于默认安装。如果已进行自定义配置，请修改命令。

```
kubectl delete -f astra_control_center_min.yaml -n netapp-acc
```

结果

```
astracontrolcenter.astra.netapp.io "astra" deleted
```

2. 使用以下命令删除 netapp-acc 命名空间：

```
kubectl delete ns netapp-acc
```

结果

```
namespace "netapp-acc" deleted
```

3. 使用以下命令删除 Astra 控制中心操作员系统组件：

```
kubectl delete -f astra_control_center_operator_deploy.yaml
```

结果

```
namespace/netapp-acc-operator deleted
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.astra.
netapp.io deleted
role.rbac.authorization.k8s.io/acc-operator-leader-election-role deleted
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role deleted
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
deleted
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role deleted
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding deleted
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding deleted
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding deleted
configmap/acc-operator-manager-config deleted
service/acc-operator-controller-manager-metrics-service deleted
deployment.apps/acc-operator-controller-manager deleted
```

对卸载问题进行故障排除

使用以下解决方法解决卸载 Astra 控制中心时出现的任何问题。

卸载 **Astra** 控制中心无法清理受管集群上的监控操作员 **POD**

如果在卸载 Astra Control Center 之前未取消管理集群，则可以使用以下命令手动删除 netapp-monitoring 命名空间和命名空间中的 Pod：

步骤

1. 删除 acc-monitoring 代理：

```
kubectl delete agents acc-monitoring -n netapp-monitoring
```

结果

```
agent.monitoring.netapp.com "acc-monitoring" deleted
```

2. 删除命名空间：

```
kubectl delete ns netapp-monitoring
```

结果

```
namespace "netapp-monitoring" deleted
```

3. 确认已删除资源：

```
kubectl get pods -n netapp-monitoring
```

结果

```
No resources found in netapp-monitoring namespace.
```

4. 确认已删除监控代理：

```
kubectl get crd|grep agent
```

示例结果：

```
agents.monitoring.netapp.com                2021-07-21T06:08:13Z
```

5. 删除自定义资源定义（CRD）信息：

```
kubectl delete crds agents.monitoring.netapp.com
```

结果

```
customresourcedefinition.apiextensions.k8s.io  
"agents.monitoring.netapp.com" deleted
```

卸载 Astra 控制中心无法清理 Traefik CRD

您可以手动删除 Traefik CRD 。CRD 是全局资源，删除它们可能会影响集群上的其他应用程序。

步骤

1. 列出集群上安装的 Traefik CRD：

```
kubectl get crds |grep -E 'traefik'
```

响应

<code>ingressroutes.traefik.containo.us</code>	<code>2021-06-23T23:29:11Z</code>
<code>ingressroutetcps.traefik.containo.us</code>	<code>2021-06-23T23:29:11Z</code>
<code>ingressrouteudps.traefik.containo.us</code>	<code>2021-06-23T23:29:12Z</code>
<code>middlewares.traefik.containo.us</code>	<code>2021-06-23T23:29:12Z</code>
<code>middlewareetcps.traefik.containo.us</code>	<code>2021-06-23T23:29:12Z</code>
<code>serverstransports.traefik.containo.us</code>	<code>2021-06-23T23:29:13Z</code>
<code>tlsoptions.traefik.containo.us</code>	<code>2021-06-23T23:29:13Z</code>
<code>tlsstores.traefik.containo.us</code>	<code>2021-06-23T23:29:14Z</code>
<code>traefikservices.traefik.containo.us</code>	<code>2021-06-23T23:29:15Z</code>

2. 删除 CRD :

```
kubectl delete crd ingressroutes.traefik.containo.us
ingressroutetcps.traefik.containo.us
ingressrouteudps.traefik.containo.us middlewares.traefik.containo.us
serverstransports.traefik.containo.us tlsoptions.traefik.containo.us
tlsstores.traefik.containo.us traefikservices.traefik.containo.us
middlewareetcps.traefik.containo.us
```

了解更多信息

- ["卸载的已知问题"](#)

版权信息

版权所有 © 2023 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。