Astra Control Service 文档

Astra Control Service

NetApp
April 24, 2024

目录

Astra Control Service 文档

发行说明

Astra Control Service的新增功能

NetApp 会定期更新 Astra Control Service ，为您提供新功能，增强功能和错误修复。

2024年3月14日

(技术预览)声明性Kubarnetes工作流

此Astra Control Center版本包含声明性Kubarnetes功能、可用于从本机Kubarnetes自定义资源(CR)执行数据管理。

此功能仅在Astra Control Service早期采用者计划(EAP)实例中可用。有关加入NetApp的信息、请联系您的EAP销售代表。

安装后 "Asta连接器" 在要管理的集群上、您将能够在UI或CR中执行以下基于CR的集群操作：

- "使用自定义资源定义应用程序"
- "定义存储分段"
- "保护整个集群"
- "备份应用程序"
- "创建快照"
- "为快照或备份创建计划"
- "从快照或备份还原应用程序"

2023年11月7日

新增功能和支持

- 使用由**ONTAP**驱动程序提供支持的存储后端为应用程序提供备份和还原功能：为启用备份和还原操作 `ontap-nas-economy` 和一些 "简单的步骤"。
- **Astra Control Service**支持内部**Red Hat OpenShift**容器平台集群

  "添加集群"

- 不可改变的备份：Astra Control现在支持 "不可更改的只读备份" 作为抵御恶意软件和其他威胁的额外安全层。
- * Astra Control配置程序简介*

  在23.10版中、Astra Control引入了一个新的软件组件、称为Astra Control配置程序、该组件可供所有获得许可的Astra Control用户使用。Asta Control配置程序提供对Asta三元数据以外的一组高级管理和存储配置功能的访问。所有Astra Control客户均可免费使用这些功能。

  - 开始使用**Astra Control**配置程序
    您可以 "启用Asta Control配置程序" 如果您已安装并配置环境以使用Asta Trident 23.10。
  - **Astra Control**配置程序功能

Astra Control配置程序23.10版提供了以下功能：

  - 通过**Kerberos 5**加密增强存储后端安全性：您可以通过提高存储安全性 "正在启用加密" 托管集群和存储后端之间的流量。Astra Control配置程序支持通过NFSv4.1连接从Red Hat OpenShift集群到Azure NetApp Files和内部ONTAP卷进行Kerberos 5加密。
  - 使用快照恢复数据：Astra Control配置程序可使用从快照快速原位还原卷 `TridentActionSnapshotRestore` (TSR) CR。
  - 使用为应用程序提供备份和还原功能 **`ontap-nas-economy`** 驱动程序支持的存储后端：如上所述以上。

- *Astra Control Service支持在AWS (ROSA)集群上运行Red Hat OpenShift Service *

  "添加集群"

- 支持管理使用**NVMe/TCP**存储的应用程序
  Astra Control现在可以管理由使用NVMe/TCP连接的永久性卷提供支持的应用程序。
- 默认情况下，执行挂钩处于关闭状态：从此版本开始，执行挂钩功能可以是 "enabled" 或禁用以提高安全性(默认情况下处于禁用状态)。如果尚未创建用于Astra Control的执行挂钩、则需要 "启用执行挂钩功能" 开始创建挂钩。如果您在此版本之前创建了执行挂钩、则执行挂钩功能将保持启用状态、您可以像往常一样使用挂钩。

2023年10月2日

新增功能和支持

这是一个小错误修复版本。

2023年7月27日

新增功能和支持

- 现在、克隆操作仅支持实时克隆(托管应用程序的当前状态)。要从快照或备份克隆、请使用还原工作流。

  "还原应用程序"

2023年6月26日

新增功能和支持

- Azure Marketplace订阅现在按小时计费、而不是按分钟计费

  "设置计费"

2023年5月30日

新增功能和支持

- 支持专用Amazon EKS集群

  "通过Astra Control Service管理专用集群"

- 支持在还原或克隆操作期间选择目标存储类

"还原应用程序"

2023年5月15日

新增功能和支持

这是一个小错误修复版本。

2023年4月25日

新增功能和支持

- 支持专用Red Hat OpenShift集群

  "通过Astra Control Service管理专用集群"

- 支持在还原操作期间包括或排除应用程序资源

  "还原应用程序"

- 支持管理纯数据应用程序

  "开始管理应用程序"

2023年1月17日

新增功能和支持

- 增强的执行挂钩功能以及其他筛选选项

  "管理应用程序执行挂钩"

- 支持将NetApp Cloud Volumes ONTAP 用作存储后端

  "了解Astra Control"

2022年11月22日

新增功能和支持

- 支持跨多个命名空间的应用程序

  "定义应用程序"

- 支持在应用程序定义中包括集群资源

  "定义应用程序"

- 增强了备份、还原和克隆操作的进度报告功能

  "监控正在运行的任务"

- 支持管理已安装兼容版本的Astra Trident的集群

  "从 Astra Control Service 开始管理 Kubernetes 集群"

- 支持在一个Astra Control Service帐户中管理多个云提供商订阅

  "管理云实例"

- 支持将公共云环境中托管的自管理Kubernetes集群添加到Astra Control Service

  "从 Astra Control Service 开始管理 Kubernetes 集群"

- 现在、Astra控制服务的计费按命名空间计费、而不是按应用程序计费

  "设置计费"

- 支持通过AWS Marketplace订阅Astra Control Service基于期限的服务

  "设置计费"

已知问题和限制

- "此版本的已知问题"
- "此版本的已知限制"

2022年9月7日

此版本为Astra Control Service基础架构提供了稳定性和故障恢复能力增强功能。

2022年8月10日

此版本包含以下新增功能和增强功能：

- 改进的应用程序管理工作流改进的应用程序管理工作流提高了定义由Astra Control管理的应用程序的灵活性。

  "管理应用程序"

- 支持Amazon Web Services集群Astra Control Service现在可以管理在Amazon Elastic Kubernetes Service托管的集群上运行的应用程序。您可以将集群配置为使用Amazon Elastic Block Store或Amazon FSx for NetApp ONTAP 作为存储后端。

  "设置Amazon Web Services"

- 增强的执行挂钩除了快照前和快照后执行挂钩之外、您现在还可以配置以下类型的执行挂钩：

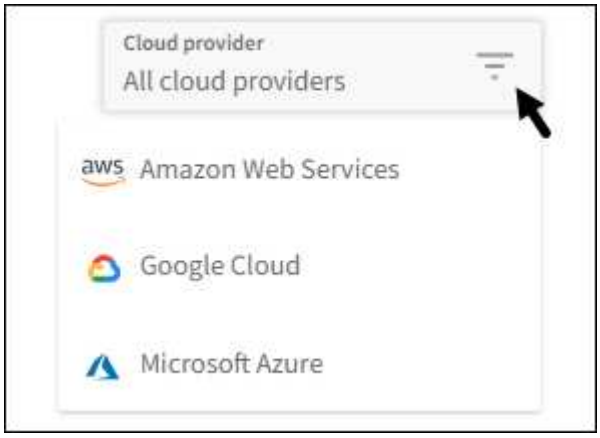
  - 预备份
  - 备份后
  - 还原后

  除了其他改进之外、Astra Control现在还支持对多个执行挂钩使用同一个脚本。

> 此版本已删除NetApp为特定应用程序提供的默认快照前和快照后执行挂钩。如果您不为快照提供自己的执行挂钩、则Astra控制服务将仅从2022年8月4日开始创建崩溃状态一致的快照。请访问 "NetApp Verda GitHub存储库" 示例执行钩脚本、您可以根据环境进行修改。

"管理应用程序执行挂钩"

- Azure Marketplace支持您现在可以通过Azure Marketplace注册到Astra Control Service。
- 选择云提供商阅读Astra Control Service文档时、您现在可以选择页面右上角的云提供商。您将看到仅与您选择的云提供商相关的文档。



2022年4月26日

此版本包含以下新增功能和增强功能：

- 命名空间基于角色的访问控制(RBAC) Astra控制服务现在支持向成员或查看器用户分配命名空间约束。

  "命名空间基于角色的访问控制（RBAC）"

- Azure Active Directory支持Astra控制服务支持使用Azure Active Directory进行身份验证和身份管理的AKS集群。

  "从 Astra Control Service 开始管理 Kubernetes 集群"

- 支持专用AKS集群现在、您可以管理使用专用IP地址的AKS集群。

  "从 Astra Control Service 开始管理 Kubernetes 集群"

- 从Astra Control中删除存储分段现在、您可以从Astra Control Service中删除存储分段。

  "删除存储分段"

2021年12月14日

此版本包含以下新增功能和增强功能：

- 新的存储后端选项

- 原位应用程序还原现在、您可以通过还原到同一集群和命名空间来原位还原应用程序的快照、克隆或备份。

  "还原应用程序"

- 使用执行挂钩的脚本事件Astra Control支持自定义脚本、您可以在为应用程序创建快照之前或之后运行这些脚本。这样，您就可以执行暂停数据库事务等任务，以使数据库应用程序的快照保持一致。

  "管理应用程序执行挂钩"

- 操作员部署的应用程序Astra Control支持一些与操作员一起部署的应用程序。

  "开始管理应用程序"

- 具有资源组范围的服务主体Astra控制服务现在支持使用资源组范围的服务主体。

  "创建 Azure 服务主体"

2021 年 8 月 5 日

此版本包含以下新增功能和增强功能：

- Astra 控制中心
  Astra Control 现在可采用新的部署模式。_Asta Control Center_是一款自行管理的软件、您可以在数据中心安装和运行该软件、以便管理内部Kubernetes集群的Kubernetes应用程序生命周期管理。

  了解更多信息。 "转至Astra控制中心文档"。

- 自带存储分段现在、您可以通过添加其他存储分段以及更改云提供商中Kubernetes集群的默认存储分段来管理Astra用于备份和克隆的存储分段。

  "管理存储分段"

2021 年 6 月 2 日

此版本包含错误修复以及 Google Cloud 支持的以下增强功能。

- 支持共享VPC现在、您可以使用共享VPC网络配置管理GCP项目中的GKEE集群。
- 现在、如果使用CVS服务类型Astra Control Service、则CVS服务类型的永久性卷大小将在使用CVS服务类型时创建最小大小为300 GiB的永久性卷。

  "了解 Astra 控制服务如何使用适用于 Google Cloud 的 Cloud Volumes Service 作为永久性卷的存储后端"。

- 现在、GKE-工作节点支持容器优化操作系统容器优化操作系统。这是对 Ubuntu 支持的补充。

  "了解有关 GKEE 集群要求的更多信息"。

2021 年 4 月 15 日

此版本包含以下新增功能和增强功能：

- 现在、支持AKS集群Astra控制服务可以管理Azure Kubernetes Service (AKS)中受管Kubernetes集群上运行的应用程序。

  "了解如何开始使用"。

- REST API Astra Control REST API现在可供使用。API 基于现代技术和当前最佳实践。

  "了解如何使用 REST API 自动执行应用程序数据生命周期管理"。

- 每年订阅Astra Control Service现在可提供_Premium订阅_。

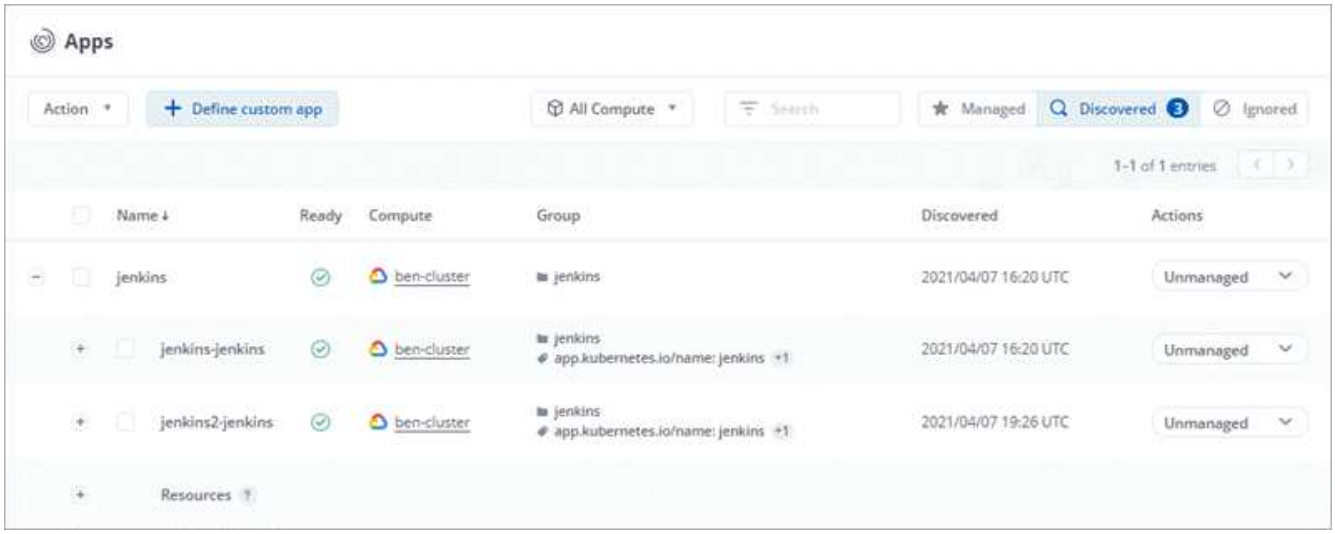
  按折扣价预付费，每年订阅一次，您可以在每个应用程序软件包中管理多达 10 个应用程序。请联系 NetApp 销售部门，根据您的组织需要购买任意数量的软件包—例如，从 Astra Control Service 购买 3 个软件包来管理 30 个应用程序。

  如果您管理的应用程序超过年度订阅所允许的数量，则每个应用程序的超额费用为每分钟 0.005 美元（与高级 PayGo 相同）。

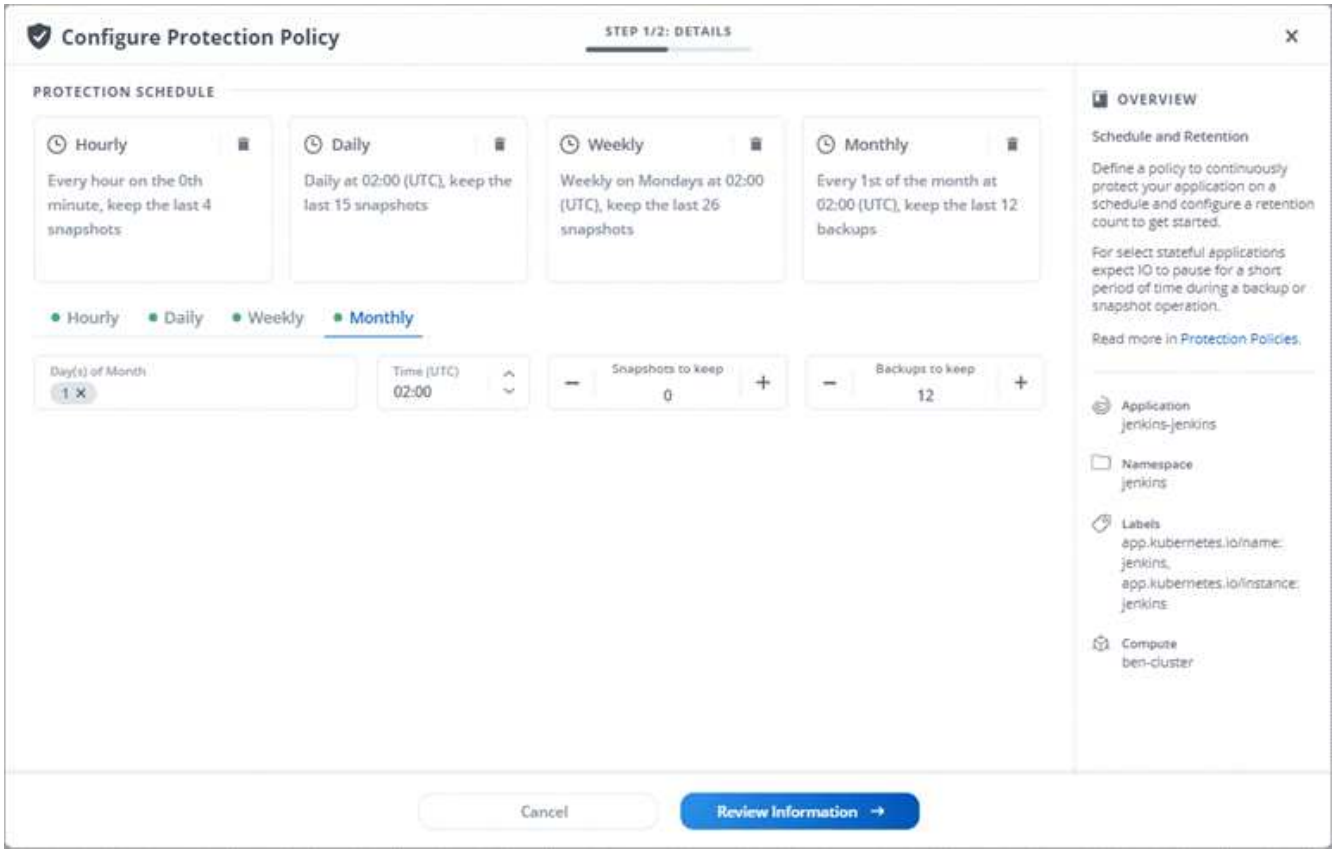
  "了解有关 Astra Control 服务定价的更多信息"。

- 命名空间和应用程序可视化我们改进了"发现的应用程序"页面、以更好地显示命名空间和应用程序之间的层次结构。只需展开一个命名空间即可查看该命名空间中包含的应用程序。

  "了解有关管理应用程序的更多信息"。



- 用户界面增强功能数据保护向导已进行了增强、易于使用。例如，我们优化了保护策略向导，以便在定义保护计划时更轻松地查看该计划。



- 活动增强功能我们可以更轻松地在您的Astra Control帐户中查看有关活动的详细信息。
  - 按受管应用程序，严重性级别，用户和时间范围筛选活动列表。
  - 将您的 Astra Control 帐户活动下载到 CSV 文件中。
  - 选择集群或应用程序后，直接从集群页面或应用程序页面查看活动。

    "了解有关查看帐户活动的更多信息"。

2021年3月1日

Astra Control Service 现在支持 "*CVS 服务类型*" 借助适用于 Google Cloud 的 Cloud Volumes Service 。这是对 *cvs-Performance* 服务类型的补充。请注意， Astra 控制服务使用适用于 Google Cloud 的 Cloud Volumes Service 作为永久性卷的存储后端。

此增强功能意味着， Astra Control Service 现在可以管理在 *any* 中运行的 Kubernetes 集群的应用程序数据 "支持 Cloud Volumes Service 的 Google 云区域"。

如果您可以灵活地在 Google Cloud 区域之间进行选择，则可以根据性能要求选择 CVS 或 CVS-Performance 。"了解有关选择服务类型的更多信息"。

2021年1月25日

我们很高兴地宣布， Astra 控制服务现已全面上市。我们采纳了从测试版收到的许多反馈，并进行了一些其他显著的改进。

- 现在，您可以通过计费从免费计划过渡到高级计划。 "了解有关计费的更多信息"。

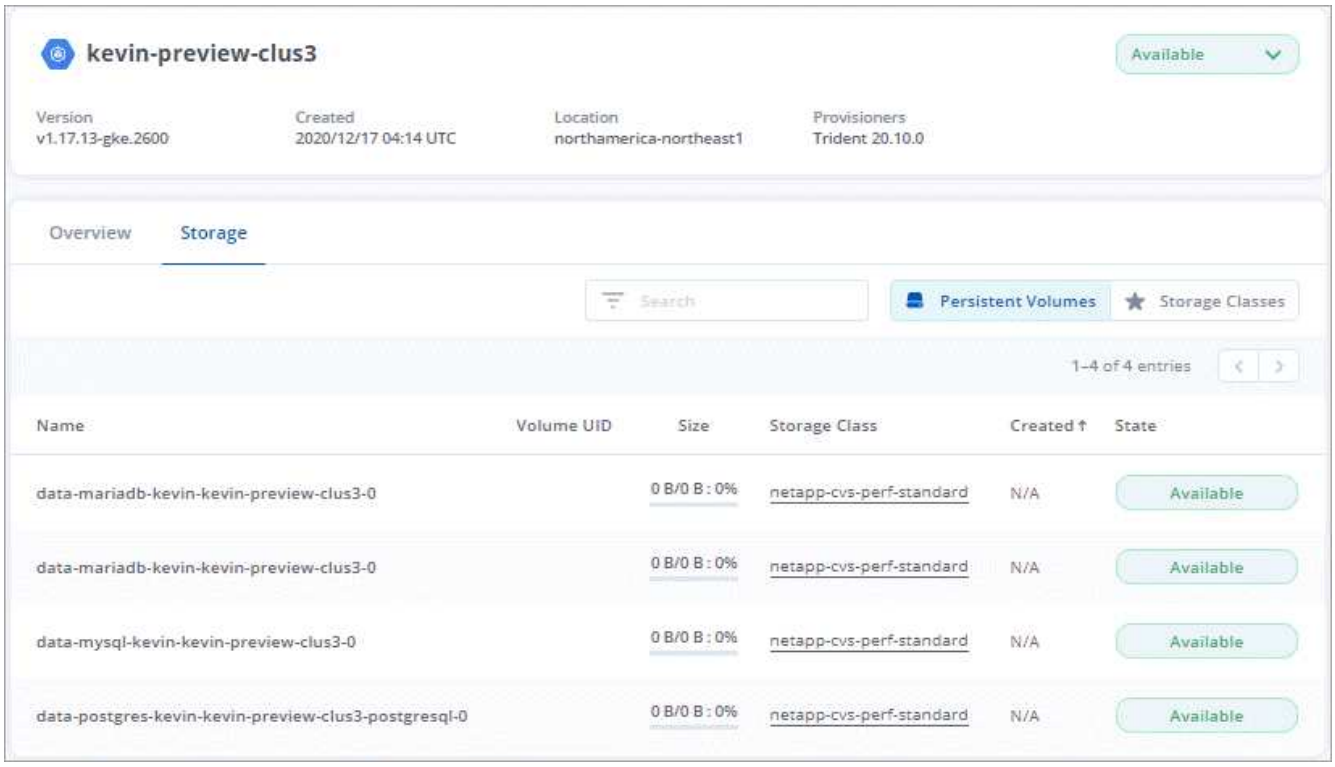
- 现在，使用 CVS-Performance 服务类型时， Astra Control Service 会创建最小大小为 100 GiB 的永久性卷。
- Astra Control Service 现在可以更快地发现应用程序。
- 现在，您可以自行创建和删除帐户。
- 当 Astra 控制服务无法再访问 Kubernetes 集群时，我们改进了通知功能。

  这些通知非常重要，因为 Astra Control Service 无法管理已断开连接的集群的应用程序。

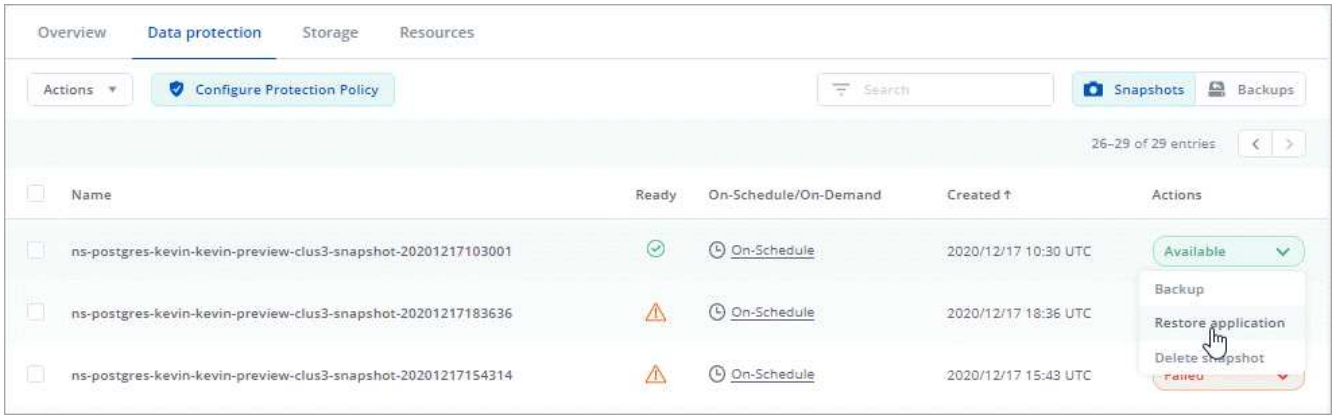
2020年12月17日(测试版更新)

我们主要关注错误修复以改善您的体验，但我们还进行了一些其他显著的改进：

- 当您将第一个 Kubernetes 计算添加到 Astra Control Service 时，现在将在集群所在的地理位置创建对象存储。
- 现在，当您在计算级别查看存储详细信息时，可以查看有关永久性卷的详细信息。



- 我们添加了一个选项，用于从现有快照或备份还原应用程序。



- 如果删除了 Astra Control Service 正在管理的 Kubernetes 集群，则该集群现在将显示为 * 已删除 * 状态。然后，您可以从 Astra Control Service 中删除此集群。
- 现在，帐户所有者可以修改为其他用户分配的角色。
- 我们添加了一个计费部分，该部分将在发布 Astra 控制服务以实现通用可用性（GA）时启用。

已知问题

已知问题可确定可能妨碍您成功使用此版本产品的问题。

以下已知问题会影响当前版本：

应用程序

- [无法在已删除并重新创建的命名空间上定义应用程序]

备份，还原和克隆

- 使用特定版本的 PostgreSQL 时应用程序克隆失败
- [如果在管理集群后添加了volumesnapshotclass、则应用程序备份和快照将失败]
- 对ONTAP NAS经济型存储类的原位还原操作失败
- 使用Kerberos传输中加密时从备份还原可能会失败
- [对于保留策略已过期的存储分段、删除后备份数据仍会保留在存储分段中]

其他问题

- 当 Astra Trident 脱机时，应用程序数据管理操作失败，并显示内部服务错误（500）

无法在已删除并重新创建的命名空间上定义应用程序

如果使用命名空间定义应用程序、删除命名空间、然后在同一命名空间中重新安装此应用程序、则此操作将失败、并显示409错误代码。要使用重新创建的命名空间定义应用程序、请先删除旧应用程序实例。

使用特定版本的 **PostgreSQL** 时应用程序克隆失败

使用 BitNami PostgreSQL 11.5.0 图表时，同一集群中的应用程序克隆始终会失败。要成功克隆，请使用图表的早期或更高版本。

如果在管理集群后添加了volumesnapshotclass、则应用程序备份和快照将失败

在这种情况下、备份和快照失败、并显示UI 500错误。作为临时解决策 、刷新应用程序列表。

对ONTAP NAS经济型存储类的原位还原操作失败

如果您对应用程序执行原位还原(将应用程序还原到其原始命名空间)、并且应用程序的存储类使用 `ontap-nas-economy` 驱动程序、如果未隐藏快照目录、则还原操作可能会失败。在原位还原之前、请按照中的说明进行操作 "为ONTAP NAS经济型操作启用备份和还原" 以隐藏快照目录。

使用Kerberos传输中加密时从备份还原可能会失败

将应用程序从备份还原到使用Kerberos传输中加密的存储后端时、还原操作可能会失败。此问题描述不会影响从快照还原或使用NetApp SnapMirror复制应用程序数据。

> 在对NFSv4卷使用Kerberos传输中加密时、请确保NFSv4卷使用正确的设置。请参阅的NetApp NFSv4域配置一节(第13页) "《NetApp NFSv4增强功能和最佳实践指南》"。

对于保留策略已过期的存储分段、删除后备份数据仍会保留在存储分段中

如果在存储分段的保留策略过期后删除应用程序的不可更改备份、则备份将从Astra Control中删除、而不是从存储分段中删除。此问题描述将在即将发布的版本中予以修复。

当 Astra Trident 脱机时，应用程序数据管理操作失败，并显示内部服务错误（ 500 ）

如果应用程序集群上的 Astra Trident 脱机（并恢复联机），并且在尝试应用程序数据管理时遇到 500 个内部服务错误，请重新启动应用程序集群中的所有 Kubernetes 节点以还原功能。

已知限制

已知限制确定了本产品版本不支持的平台、设备或功能、或者这些平台、设备或功能无法与产品正确交互操作。仔细审查这些限制。

一般限制

以下限制会影响 Astra Control Service 在任何受支持的 Kubernetes 部署中对 Kubernetes 集群的管理。

与 Postgres Pod 的现有连接导致故障

在 Postgres Pod 上执行操作时，不应直接在 Pod 中连接以使用 psql 命令。Astra 控制服务需要使用 psql 访问权限来冻结和解冻数据库。如果已建立连接，则快照，备份或克隆将失败。

"Activity"页面最多可显示100、000个事件

Astra Control Activity页面最多可显示100、000个事件。要查看所有记录的事件、请使用检索这些事件 "Astra Control REST API"。

GKEE 集群管理的限制

以下限制适用于在 Google Kubernetes Engine （ GKEE ）中管理 Kubernetes 集群。

应用程序管理限制

以下限制会影响 Astra Control Service 对应用程序的管理。

使用同一命名空间的多个应用程序不能一起还原到不同的命名空间

如果您通过在Astra Control中创建多个应用程序定义来管理多个使用同一命名空间的应用程序、则无法将所有应用程序还原到另一个命名空间。您需要将每个应用程序还原到其自己单独的命名空间。

Astra Control不会自动为云实例分配默认分段

Astra Control不会自动为任何云实例分配默认分段。您需要手动设置云实例的默认存储分段。如果未设置默认分段、您将无法在两个集群之间执行应用程序克隆操作。

不支持对使用证书管理器的应用程序执行原位还原操作

此版本的Astra控制服务不支持使用证书管理器原位还原应用程序。支持将还原操作还原到其他命名空间和克隆操作。

使用设置的存储类部署应用程序后，应用程序克隆将失败

在使用显式设置的存储类（例如， `helm install …-set global.storageClass=netapp-cvs-perf-至 至至` ）部署应用程序后，后续克隆应用程序的尝试要求目标集群具有最初指定的存储类。将具有显式设置的存储类的应用程序克隆到没有相同存储类的集群将失败。此情况下没有恢复步骤。

使用 **Ppass by reference operators** 安装的应用程序克隆可能会失败

Astra Control 支持使用命名空间范围的运算符安装的应用程序。这些操作员通常采用 " 按价值传递 " 架构，而不是 " 按参考传递 " 架构。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"

请注意、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。

> 在克隆操作期间、需要IngressClass资源或webhooks才能正常运行的应用程序不能在目标集群上定义这些资源。

基于角色的访问控制（**Role-Based Access Control**， **RBAC**）限制

以下限制适用于 Astra Control 限制用户访问资源或功能的方式。

具有命名空间 **RBAC** 限制的用户可以添加和取消管理集群

不应允许具有命名空间 RBAC 限制的用户添加或取消管理集群。由于当前的限制， Astra 不会阻止此类用户取消管理集群。

具有命名空间约束的成员用户无法访问克隆或还原的应用程序，除非管理员用户将命名空间添加到此限制中

任意 `member` 使用命名空间名称/ID限制RBAC的用户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。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 `member` 受影响用户的用户帐户和更新角色约束、以授予对新命名空间的访问权限。

使用某些**Snapshot**控制器版本的**Kubernetes 1.25**或更高版本集群的快照可能会失败

如果在运行1.25或更高版本的Kubernetes集群上安装了v1beta1版本的快照控制器API、则该集群的快照可能会失败。

作为临时解决策 、在升级现有Kubernetes 1.25或更高版本的安装时、请执行以下操作：

1. 删除任何现有的Snapshot CRD和任何现有的Snapshot控制器。
2. "卸载 Astra Trident"。
3. "安装快照CRD和快照控制器"。
4. "安装最新版本的Astra Trdent"。
5. "创建卷快照类"。

入门

了解Astra Control

Astra Control 是 Kubernetes 应用程序数据生命周期管理解决方案，可简化有状态应用程序的操作。轻松保护，备份和迁移 Kubernetes 工作负载，并即时创建有效的应用程序克隆。

功能

Astra Control 为 Kubernetes 应用程序数据生命周期管理提供了关键功能：

- 自动管理永久性存储
- 创建应用程序感知型按需快照和备份
- 自动执行策略驱动的快照和备份操作
- 将应用程序和数据从一个 Kubernetes 集群迁移到另一个集群
- 使用NetApp SnapMirror技术(Astra Control Center)将应用程序复制到远程系统
- 将应用程序从暂存克隆到生产
- 直观显示应用程序运行状况和保护状态
- 使用Web UI或API实施备份和迁移工作流

部署模式

Astra Control 有两种部署模式：

- * Astra Control Service*：NetApp管理的服务、可为多个云提供商环境中的Kubernetes集群以及自管理Kubernetes集群提供应用程序感知型数据管理。
- * Astra Control Center* ：自管理软件，可为内部环境中运行的 Kubernetes 集群提供应用程序感知型数据管理。Astra控制中心还可以安装在多个云提供商环境中、并具有一个NetApp Cloud Volumes ONTAP 存储后端。

	Astra 控制服务	**Astra** 控制中心
如何提供?	作为 NetApp 提供的一项完全托管的云服务	作为可下载、安装和管理的软件
它托管在何处?	基于 NetApp 选择的公有云	在您自己的Kubernetes集群上
如何更新?	由 NetApp 管理	您可以管理任何更新

	Astra 控制服务	**Astra** 控制中心

支持哪些**Kubednetes**分发版?	• 云提供商
	  ◦ Amazon Web Services
	  ▪ Amazon Elelic Kubelnetes Service (EKS)
	  ◦ Google Cloud
	  ▪ Google Kubernetes Engine （GKEE）
	  ◦ Microsoft Azure
	  ▪ Azure Kubernetes Service （AKS）
	• 自管理集群
	  ◦ Kubnetes (上游)
	  ◦ Rancher Kubernetes Engine （RKE）
	  ◦ Red Hat OpenShift 容器平台
	• 内部集群
	  ◦ Red Hat OpenShift容器平台内部部署
		• 基于Azure堆栈HCI的Azure Kubnetes Service
		• Google Anthos
		• Kubnetes (上游)
		• Rancher Kubernetes Engine （RKE）
		• Red Hat OpenShift 容器平台

	Astra 控制服务	Astra 控制中心
支持哪些存储后端?	• 云提供商 ◦ Amazon Web Services ▪ Amazon EBS ▪ 适用于 NetApp ONTAP 的 Amazon FSX ▪ "Cloud Volumes ONTAP" ◦ Google Cloud ▪ Google 持久磁盘 ▪ NetApp Cloud Volumes Service ▪ "Cloud Volumes ONTAP" ◦ Microsoft Azure ▪ Azure受管磁盘 ▪ Azure NetApp Files ▪ "Cloud Volumes ONTAP" • 自管理集群 ◦ Amazon EBS ◦ Azure受管磁盘 ◦ Google 持久磁盘 ◦ "Cloud Volumes ONTAP" ◦ NetApp MetroCluster ◦ "Longhorn" • 内部集群 ◦ NetApp MetroCluster ◦ NetApp ONTAP AFF 和 FAS 系统 ◦ NetApp ONTAP Select ◦ "Cloud Volumes ONTAP" ◦ "Longhorn"	• NetApp ONTAP AFF 和 FAS 系统 • NetApp ONTAP Select • "Cloud Volumes ONTAP" • "Longhorn"

Astra 控制服务的工作原理

Astra Control Service 是一种由 NetApp 管理的云服务，它始终处于启用状态，并使用最新功能进行更新。它利用多个组件实现应用程序数据生命周期管理。

从较高的层面来看， Astra Control Service 的工作原理如下：

- 您可以通过设置云提供商并注册 Astra 帐户开始使用 Astra Control Service 。

+*对于GKEE集群、Astra Control Service使用 "适用于 Google Cloud 的 NetApp Cloud Volumes Service" 或 Google Persistent Disk 作为永久性卷的存储后端。

+*对于AKS集群、Astra Control Service使用 "Azure NetApp Files" 或Azure受管磁盘作为永久性卷的存储后端。

+*对于Amazon EKS集群、Astra Control Service使用 "Amazon Elastic Block Store" 或 "适用于 NetApp ONTAP 的 Amazon FSX" 作为永久性卷的存储后端。

- 您可以将第一个 Kubernetes 计算添加到 Astra Control Service 中。然后， Astra 控制服务将执行以下操作：
  - 在云提供商帐户中创建一个对象存储，该帐户是备份副本的存储位置。

+在Azure中、Astra Control Service还会为Blob容器创建资源组、存储帐户和密钥。

- 在集群上创建新的管理员角色和 Kubernetes 服务帐户。
- 使用此新管理员角色在集群上安装link./概念/architution#Astra-control-components [Astra Control置备程序]、并创建一个或多个存储类。
- 如果您使用NetApp云服务存储产品作为存储后端、Astra控制服务将使用Astra控制配置程序为应用程序配置永久性卷。如果您使用Amazon EBS或Azure托管磁盘作为存储后端、则需要安装特定于提供商的CSI驱动程序。中提供了安装说明 "设置Amazon Web Services" 和 "使用 Azure 受管磁盘设置 Microsoft Azure"。
  - 此时、您可以从集群定义应用程序。永久性卷将通过新的默认存储类在存储后端配置。
  - 然后，您可以使用 Astra Control Service 管理这些应用程序，并开始创建快照，备份和克隆。

Astra Control的免费计划支持您管理帐户中多达10个命名空间。如果您要管理10个以上的命名空间、则需要通过从"免费计划"升级到"高级计划"来设置计费。

Astra 控制中心的工作原理

Astra 控制中心在您自己的私有云中本地运行。

Astra控制中心支持Kuburenet集群、其中Astra控制配置程序配置了存储类、并具有ONTAP存储后端。

Astra 控制中心完全集成到 AutoSupport 和 Active IQ 生态系统中，可为用户和 NetApp 支持提供故障排除和使用信息。

您可以使用 90 天评估许可证试用 Astra Control Center 。评估版可通过电子邮件和社区选项获得支持。此外，您还可以从产品支持信息板访问知识库文章和文档。

要安装和使用 Astra 控制中心，您需要满足特定的要求 "要求"。

从较高的层面来看， Astra 控制中心的工作原理如下：

- 您可以在本地环境中安装 Astra Control Center 。详细了解如何操作 "安装 Astra 控制中心"。
- 您可以完成一些设置任务，例如：
  - 设置许可
  - 添加第一个集群。
  - 添加在添加集群时发现的存储后端。

- 添加用于存储应用程序备份的对象存储分段。

详细了解如何操作 "设置 Astra 控制中心"。

您可以向集群添加应用程序。或者、如果要管理的集群中已有一些应用程序、则可以使用Astra控制中心来管理它们。然后、使用Astra控制中心创建快照、备份、克隆和复制关系。

有关详细信息 ...

- "NetApp Asta产品系列文档"
- "Astra 控制中心文档"
- "Astra Control API文档"
- "Astra Trident 文档"
- "ONTAP 文档"

支持的 Kubernetes 部署

Astra控制服务可以管理Amazon Elastic Kubernetes Service (EKS)中受管Kubernetes集群上运行的应用程序以及您自己管理的集群。

Astra控制服务可以管理Google Kubernetes Engine (GKEE)中受管Kubernetes集群上运行的应用程序以及您自己管理的集群。

Astra控制服务可以管理Azure Kubernetes Service (AKS)中受管Kubernetes集群上运行的应用程序以及您自己管理的集群。

- "了解如何为Astra Control Service设置Amazon Web Services"。
- "了解如何为 Astra Control Service 设置 Google Cloud"。
- "了解如何使用适用于 Astra 控制服务的 Azure NetApp Files 设置 Microsoft Azure"。
- "了解如何使用 Azure 托管磁盘为 Astra Control Service 设置 Microsoft Azure"。
- "了解如何在将自管理集群添加到Astra Control Service之前对其进行准备"。

Astra 控制服务快速入门

此页面简要概述了开始使用 Astra 控制服务所需完成的步骤。每个步骤中的链接将转到一个页面，其中提供了更多详细信息。

[一个] 设置您的云提供商

1. Google Cloud
    - 查看 Google Kubernetes Engine 集群要求。
    - 从 Google 云市场购买适用于 Google Cloud 的 Cloud Volumes Service 。
    - 启用所需的 API 。

- 创建服务帐户和服务帐户密钥。
- 设置从 VPC 到适用于 Google Cloud 的 Cloud Volumes Service 的网络对等关系。

  "了解有关 Google Cloud 要求的更多信息"。

2. Amazon Web Services：
   - 查看Amazon Web Services集群要求。
   - 创建Amazon帐户。
   - 安装Amazon Web Services CLI。
   - 创建IAM用户。
   - 创建并附加权限策略。
   - 保存IAM用户的凭据。

     "了解有关Amazon Web Services要求的更多信息"。

3. Microsoft Azure
   - 查看您计划使用的存储后端的 Azure Kubernetes Service 集群要求。

     "了解有关 Microsoft Azure 和 Azure NetApp Files 要求的更多信息"。

"了解有关 Microsoft Azure 和 Azure 托管磁盘要求的更多信息"。

如果您要管理自己的集群、而集群不是由云提供商托管的、请查看自管理集群的要求。
"了解有关自管理集群要求的更多信息"。

[两个] 完成 Astra Control 注册

1. 创建 "NetApp BlueXP" 帐户。
2. 在创建Astra Control帐户时、请指定您的NetApp BlueXP电子邮件ID "从Astra Control产品页面"。

"了解有关注册过程的更多信息"。

[三个] 将集群添加到 Astra Control

登录后，选择 * 添加集群 * 以开始使用 Astra Control 管理集群。

"了解有关添加集群的更多信息"。

设置您的云提供商

设置Amazon Web Services

要使用Astra Control Service管理Amazon Elastic Kubernetes Service (EKS)集群、需要执行几个步骤来准备Amazon Web Services项目。

快速开始设置Amazon Web Services

按照以下步骤快速入门，或者向下滚动到其余部分以了解完整详细信息。

[一个] 查看适用于Amazon Web Services的Astra Control Service要求

确保集群运行状况良好并运行受支持的Kubernetes版本、工作节点处于联机状态并运行Linux或Windows等。 了解有关此步骤的更多信息。

[两个] 创建Amazon帐户

如果您还没有Amazon帐户、则需要创建一个帐户、以便可以使用EKS。 了解有关此步骤的更多信息。

[三个] 安装Amazon Web Services CLI

安装AWS命令行界面、以便从命令行管理AWS。 按照分步说明进行操作。

[四个] 可选：创建IAM用户

创建Amazon身份和访问管理(IAM)用户。您也可以跳过此步骤并将现有IAM用户与Astra Control Service结合使用。

阅读分步说明。

[五个] 创建并附加权限策略

创建具有所需权限的策略、以使Astra Control Service能够与您的AWS帐户进行交互。

阅读分步说明。

[六个] 保存IAM用户的凭据

保存IAM用户的凭据、以便将这些凭据导入到Astra Control Service中。

阅读分步说明。

EKS集群要求

Kubernetes 集群必须满足以下要求，才能通过 Astra Control Service 发现和管理它。

Kubernetes 版本
: 集群运行的Kubbernetes版本必须介于1.25到1.28之间。

映像类型
: 每个工作节点的映像类型必须为Linux。

集群状态
: 集群必须运行状况良好，并且至少有一个联机辅助节点，并且没有处于故障状态的辅助节点。

Asta Control配置程序
: 使用存储后端执行操作需要Astra Control配置程序和外部Snapshot控制器。要启用这些操作、请执行以下操作：

1. "安装快照CRD和快照控制器"。
2. "启用Asta Control配置程序"。
3. "创建卷快照类"。

适用于**Amazon Elastic Block Store (EBS)**的**CSI**驱动程序

如果您使用Amazon EBS存储后端、则需要为EBS安装容器存储接口(Container Storage Interface、CSI)驱动程序(它不会自动安装)。

有关安装CSI驱动程序的说明、请参阅相关步骤。

安装外部快照程序

如果您尚未执行此操作， "安装快照CRD和快照控制器"。

将**CSI**驱动程序作为**Amazon EKS**加载项进行安装

1. 为服务帐户创建Amazon EBS CSI驱动程序IAM角色。按照说明进行操作 "在Amazon文档中"、使用说明中的AWS命令行界面命令。
2. 使用以下AWS命令行界面命令添加Amazon EBS CSI加载项、将括号<>中的信息替换为特定于您的环境的值。将<driver_role>替换为您在上一步中创建的EBS CSI驱动程序角色的名称：

```
aws eks create-addon \
  --cluster-name <CLUSTER_NAME> \
  --addon-name aws-ebs-csi-driver \
  --service-account-role-arn
arn:aws:iam::<ACCOUNT_ID>:role/<DRIVER_ROLE>
```

配置**EBS**存储类

1. 将Amazon EBS CSI驱动程序GitHub存储库克隆到您的系统。

```
git clone https://github.com/kubernetes-sigs/aws-ebs-csi-
driver.git
```

2. 导航到动态配置示例目录。

```
cd aws-ebs-csi-driver/examples/kubernetes/dynamic-provisioning/
```

3. 从清单目录部署EBS SC存储类和EBS声明永久性卷声明。

```
kubectl apply -f manifests/storageclass.yaml
kubectl apply -f manifests/claim.yaml
```

4. 描述EBS-SC存储类。

```
kubectl describe storageclass ebs-sc
```

您应看到描述存储类属性的输出。

创建**Amazon**帐户

如果您还没有Amazon帐户、则需要创建一个帐户来为Amazon EKS启用计费。

步骤

1. 转至 "Amazon主页" 、选择右上角的*登录*、然后选择*从此处开始*。
2. 按照提示创建帐户。

安装Amazon Web Services CLI

安装AWS命令行界面、以便从命令行管理AWS资源。

步骤

1. 转至 "AWS命令行界面入门" 并按照说明安装CLI。

可选：创建IAM用户

创建IAM用户、以便您可以使用和管理AWS服务和资源、并提高安全性。您也可以跳过此步骤、并将现有IAM用户与Astra Control Service结合使用。

步骤

1. 转至 "创建IAM用户" 并按照说明创建IAM用户。

创建并附加权限策略

创建具有所需权限的策略、以使Astra Control Service能够与您的AWS帐户进行交互。

步骤

1. 创建一个名为`policy.json`的新文件。
2. 将以下JSON内容复制到文件中：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "fsx:DescribeVolumes",
                "ec2:DescribeRegions",
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:GetObject",
                "iam:SimulatePrincipalPolicy",
                "s3:ListAllMyBuckets",
                "eks:DescribeCluster",
                "eks:ListNodegroups",
                "eks:DescribeNodegroup",
                "eks:ListClusters",
                "iam:GetUser",
                "s3:DeleteObject",
                "s3:DeleteBucket",
                "autoscaling:DescribeAutoScalingGroups"
            ],
            "Resource": "*"
        }
    ]
}
```

3. 创建策略：

```
POLICY_ARN=$(aws iam create-policy  --policy-name <policy-name> --policy
-document file://policy.json  --query='Policy.Arn' --output=text)
```

4. 将策略附加到 IAM 用户。将`<IAM用户名>`替换为您创建的IAM用户或现有IAM用户的用户名：

```
aws iam attach-user-policy --user-name <IAM-USER-NAME> --policy-arn
=$POLICY_ARN
```

保存IAM用户的凭据

保存IAM用户的凭据、以便让Astra Control Service能够识别该用户。

步骤

1. 下载凭据。将`<IAM用户名>`替换为要使用的IAM用户的用户名：

```
aws iam create-access-key --user-name <IAM-USER-NAME> --output json >
credential.json
```

结果

此时将创建`credential.json`文件、您可以将凭据导入到Astra Control Service中。

设置 Google Cloud

在使用 Astra Control Service 管理 Google Kubernetes Engine 集群之前，需要执行一些步骤来准备 Google Cloud 项目。

> 如果您不开始使用 Google Cloud Volumes Service for Google Cloud 作为存储后端，而打算稍后再使用，则应完成必要的步骤立即配置 Google Cloud Volumes Service for Google Cloud 。稍后创建服务帐户意味着您可能无法访问现有存储分段。

快速开始设置 Google Cloud

按照以下步骤快速入门，或者向下滚动到其余部分以了解完整详细信息。

[一个] 查看 Google Kubernetes Engine 的 Astra Control Service 要求

确保集群运行状况良好并运行受支持的Kubernetes版本、工作节点处于联机状态并运行受支持的映像类型等。了解有关此步骤的更多信息。

[两个] （可选）：购买适用于 Google Cloud 的 Cloud Volumes Service

如果您计划使用适用于 Google Cloud 的 Cloud Volumes Service 作为存储后端，请转到 Google 云市场中的 NetApp Cloud Volumes Service 页面，然后选择购买。 了解有关此步骤的更多信息。

[三个] 在 Google Cloud 项目中启用 API

启用以下 Google Cloud API：

- Google Kubernetes 引擎
- 云存储
- 云存储 JSON API
- 服务使用情况
- Cloud Resource Manager API
- NetApp Cloud Volumes Service

    - Cloud Volumes Service for Google Cloud 必需
    - Google Persistent Disk 的可选（但建议）
- 服务使用者管理 API
- 服务网络 API
- 服务管理 API

按照分步说明进行操作。

[四个] 创建具有所需权限的服务帐户

创建具有以下权限的 Google Cloud 服务帐户：

- Kubernetes 引擎管理员
- NetApp Cloud Volumes 管理员
    - Cloud Volumes Service for Google Cloud 必需
    - Google Persistent Disk 的可选（但建议）
- 存储管理员
- 服务使用情况查看器
- 计算网络查看器

阅读分步说明。

[五个] 创建服务帐户密钥

为服务帐户创建密钥，并将密钥文件保存在安全位置。 按照分步说明进行操作。

[六个] （可选）：为 **VPC** 设置网络对等

如果您计划使用适用于 Google Cloud 的 Cloud Volumes Service 作为存储后端，请设置从 VPC 到适用于 Google Cloud 的 Cloud Volumes Service 的网络对等关系。 按照分步说明进行操作。

GKEE 集群要求

Kubernetes 集群必须满足以下要求，才能通过 Astra Control Service 发现和管理它。请注意，只有当您计划使用适用于 Google Cloud 的 Cloud Volumes Service 作为存储后端时，其中某些要求才适用。

Kubernetes 版本

集群运行的Kubbernetes版本必须介于1.26到1.28之间。

映像类型

每个工作节点的映像类型必须为 `COS_CONTAINERD`。

集群状态

集群必须运行状况良好，并且至少有一个联机辅助节点，并且没有处于故障状态的辅助节点。

Google Cloud 地区

如果您计划使用 Cloud Volumes Service for Google Cloud 作为存储后端，则集群必须在中运行 "支持 Cloud Volumes Service for Google Cloud 的 Google 云区域。" 请注意， Astra 控制服务支持两种服务类型： CVS 和 CVS-Performance 。作为最佳实践，您应选择一个支持适用于 Google Cloud 的 Cloud Volumes Service 的区域，即使您不将其用作存储后端也是如此。这样，如果您的性能要求发生变化，将来可以更轻松地将 Cloud Volumes Service for Google Cloud 用作存储后端。

网络

如果您计划使用适用于 Google Cloud 的 Cloud Volumes Service 作为存储后端，则集群必须位于与适用于 Google Cloud 的 Cloud Volumes Service 建立对等关系的 VPC 中。 下面介绍了此步骤。

专用集群

如果集群为专用集群，则会显示 "授权网络" 必须允许 Astra 控制服务 IP 地址：

52.188.218.166/32

GKEE 集群的操作模式

您应使用标准操作模式。目前尚未测试自动驾驶模式。 "了解有关操作模式的更多信息"。

存储池

如果将NetApp Cloud Volumes Service用作CVS服务类型的存储后端、则需要先配置存储池、然后才能配置卷。请参见 "GKE- 集群的服务类型，存储类和 PV 大小" 有关详细信息 …

可选：购买适用于**Google Cloud**的**Cloud Volumes Service**

Astra 控制服务可以使用适用于 Google Cloud 的 Cloud Volumes Service 作为永久性卷的存储后端。如果您计划使用此服务，则需要从 Google 云市场购买适用于 Google Cloud 的 Cloud Volumes Service ，以便为永久性卷开票。

步骤

1. 转至 "NetApp Cloud Volumes Service 页面" 在 Google Cloud Marketplace 中，选择 * 购买 * ，然后按照提示进行操作。

   "按照 Google Cloud 文档中的分步说明购买并启用此服务"。

在项目中启用 **API**

您的项目需要访问特定 Google Cloud API 的权限。API 用于与 Google 云资源进行交互，例如 Google Kubernetes Engine （ GKEE ）集群和 NetApp Cloud Volumes Service 存储。

步骤

1. "使用 Google Cloud 控制台或 gcloud CLI 启用以下 API"：
   - Google Kubernetes 引擎
   - 云存储
   - 云存储 JSON API
   - 服务使用情况
   - Cloud Resource Manager API

- NetApp Cloud Volumes Service （适用于 Google Cloud 的 Cloud Volumes Service 所需）
- 服务使用者管理 API
- 服务网络 API
- 服务管理 API

以下视频显示了如何从 Google Cloud 控制台启用 API 。

► https://docs.netapp.com/zh-cn/astra-control-service/media/get-started/video-enable-gcp-apis.mp4 *(video)*

创建服务帐户

Astra Control Service 使用 Google Cloud 服务帐户为您的 Kubernetes 应用程序数据管理提供便利。

步骤

1. 转到 Google Cloud ，然后 "使用 console ， gcloud 命令或其他首选方法创建服务帐户"。
2. 为服务帐户授予以下角色：
   - * Kubernetes Engine Admin* —用于列出集群并创建管理员访问权限以管理应用程序。
   - * NetApp Cloud Volumes Admin* —用于管理应用程序的永久性存储。
   - * 存储管理员 * —用于管理用于备份应用程序的存储分段和对象。
   - * 服务使用情况查看器 * - 用于检查是否已启用所需的 Cloud Volumes Service for Google Cloud API 。
   - * 计算网络查看器 * - 用于检查 Kubernetes VPC 是否允许访问适用于 Google Cloud 的 Cloud Volumes Service 。

如果您要使用 gcloud ，可以从 Astra Control 界面中执行相关步骤。选择 * 帐户 > 凭据 > 添加凭据 * ，然后选择 * 说明 * 。

如果您要使用 Google Cloud 控制台，以下视频将介绍如何从控制台创建服务帐户。

► https://docs.netapp.com/zh-cn/astra-control-service/media/get-started/video-create-gcp-service-

account.mp4 *(video)*

为共享 **VPC** 配置服务帐户

要管理驻留在一个项目中但使用不同项目（共享 VPC ）中的 VPC 的 GKEE 集群，您需要将 Astra 服务帐户指定为具有 * 计算网络查看器 * 角色的主机项目的成员。

步骤

1. 从 Google Cloud 控制台中，转到 * IAM & Admin* 并选择 * 服务帐户 * 。
2. 找到已有的 Astra 服务帐户 "所需权限" 然后复制此电子邮件地址。
3. 转到您的主机项目，然后选择 * IAM & Admin* > * IAM * 。
4. 选择 * 添加 * 并为服务帐户添加一个条目。
   a. * 新成员 *： 输入服务帐户的电子邮件地址。
   b. * 角色 *： 选择 * 计算网络查看器 * 。
   c. 选择 * 保存 * 。

结果

使用共享 VPC 添加 GKEE 集群将完全适用于 Astra 。

创建服务帐户密钥

您将在添加第一个集群时提供服务帐户密钥，而不是向 Astra Control Service 提供用户名和密码。Astra 控制服务使用服务帐户密钥来建立您刚刚设置的服务帐户的身份。

服务帐户密钥是以 JavaScript 对象表示法（ JSON ）格式存储的纯文本。其中包含有关您有权访问的 GCP 资源的信息。

您只能在创建密钥时查看或下载 JSON 文件。但是，您可以随时创建新密钥。

步骤

1. 转到 Google Cloud ，然后 "使用 console ， gcloud 命令或其他首选方法创建服务帐户密钥"。
2. 出现提示时，将服务帐户密钥文件保存在安全位置。

以下视频显示了如何从 Google Cloud 控制台创建服务帐户密钥。

► https://docs.netapp.com/zh-cn/astra-control-service/media/get-started/video-create-gcp-service-account-

key.mp4 *(video)*

可选：为**VPC**设置网络对等

如果您计划将 Cloud Volumes Service for Google Cloud 用作存储后端服务，则最后一步是设置从 VPC 到 Cloud Volumes Service for Google Cloud 的网络对等关系。

设置网络对等关系的最简单方法是直接从 Cloud Volumes Service 获取 gcloud 命令。在创建新文件系统时，可以从 Cloud Volumes Service 访问这些命令。

步骤

1. "转至NetApp BlueXP全球区域地图" 并确定要在集群所在的 Google Cloud 区域中使用的服务类型。

   Cloud Volumes Service 提供两种服务类型： CVS 和 CVS-Performance 。 "详细了解这些服务类型"。

2. "转到 Google Cloud Platform 中的 Cloud Volumes"。
3. 在 * 卷 * 页面上，选择 * 创建 * 。
4. 在 * 服务类型 * 下，选择 * CVS* 或 * CVS-Performance* 。

   您需要为 Google Cloud 区域选择正确的服务类型。这是您在步骤 1 中确定的服务类型。选择服务类型后，页面上的区域列表将更新为支持该服务类型的区域。

   完成此步骤后，您只需输入网络信息即可获取命令。

5. 在 * 区域 * 下，选择您的区域和分区。
6. 在 * 网络详细信息 * 下，选择您的 VPC 。

   如果尚未设置网络对等，您将看到以下通知：

7. 选择按钮以查看 network peering set up 命令。
8. 复制命令并在 Cloud Shell 中运行。

   有关使用这些命令的详细信息，请参见 "适用于 GCP 的 Cloud Volumes Service 的快速入门"。

   "了解有关配置私有服务访问和设置网络对等的更多信息"。

9. 完成后，您可以在 * 创建文件系统 * 页面上选择取消。

   我们开始创建此卷只是为了获取用于建立网络对等关系的命令。

使用 **Azure NetApp Files** 设置 **Microsoft Azure**

在使用 Astra Control Service 管理 Azure Kubernetes Service 集群之前，需要执行一些步骤来准备 Microsoft Azure 订阅。如果您计划使用 Azure NetApp Files 作为存储后端，请按照以下说明进行操作。

快速开始设置 **Azure**

按照以下步骤快速入门，或者向下滚动到其余部分以了解完整详细信息。

[一个**]** 查看 **Azure Kubernetes Service** 的 **Astra Control Service** 要求

确保集群运行状况良好并运行受支持的Kubernetes版本、节点池处于联机状态并运行Linux等。 了解有关此步骤的更多信息。

[两个**]** 注册 **Microsoft Azure**

创建 Microsoft Azure 帐户。 了解有关此步骤的更多信息。

[三个**]** 对于 **Azure NetApp Files** ，

注册 NetApp 资源提供商。 了解有关此步骤的更多信息。

[四个**] Create a NetApp account**

转到 Azure 门户中的 Azure NetApp Files 并创建 NetApp 帐户。 了解有关此步骤的更多信息。

[五个**]** 设置容量池

为永久性卷设置一个或多个容量池。 了解有关此步骤的更多信息。

[六个**] Delegate a subnet to Azure NetApp Files**

将子网委派给 Azure NetApp Files ，以便 Astra 控制服务可以在该子网中创建永久性卷。 了解有关此步骤的更多信息。

[七个**]** 创建 **Azure** 服务主体

创建具有 " 贡献者 " 角色的 Azure 服务主体。 了解有关此步骤的更多信息。

[八个**]** 可选：为**Azure**备份存储分段配置冗余

默认情况下、Astra Control Service用于存储Azure Kubernetes Service备份的存储分段使用本地冗余存储(LRS)冗余选项。作为一个可选步骤、您可以为Azure存储分段配置更持久的冗余级别。 了解有关此步骤的更多信息。

Azure Kubernetes Service 集群要求

Kubernetes 集群必须满足以下要求，才能通过 Astra Control Service 发现和管理它。

Kubernetes 版本

集群必须运行Kubnetes 1.26至1.28版。

映像类型

所有节点池的映像类型必须为 Linux 。

集群状态

集群必须运行状况良好，并且至少有一个联机辅助节点，并且没有处于故障状态的辅助节点。

Azure 区域

集群必须位于 Azure NetApp Files 可用的区域。 "按区域查看 Azure 产品"。

订阅。

集群必须位于启用了 Azure NetApp Files 的订阅中。您可以选择订阅 注册 Azure NetApp Files。

vNet

请考虑以下 vNet 要求：

- 集群必须位于可直接访问 Azure NetApp Files 委派子网的 vNet 中。 了解如何设置委派子网。
- 如果您的 Kubernetes 集群位于与另一个 vNet 中的 Azure NetApp Files 委派子网建立对等关系的 vNet 中，则对等连接的两端必须处于联机状态。
- 请注意， vNet （包括对等 VNet ）与 Azure NetApp Files 中使用的 IP 数量的默认限制为 1 ， 000 。 "查看 Azure NetApp Files 资源限制"。

  如果您即将达到限制，您有两种选择：

  - 您可以 "提交增加限制的请求"。如需帮助，请联系您的 NetApp 代表。
  - 创建新的 Amazon Kubernetes Service （ AKS ）集群时，请为此集群指定一个新网络。创建新网络后，配置新子网并将子网委派给 Azure NetApp Files 。

注册 **Microsoft Azure**

如果您没有 Microsoft Azure 帐户，请先注册 Microsoft Azure 。

步骤

1. 转至 "Azure 订阅页面" 订阅 Azure 服务。
2. 选择一个计划并按照说明完成订阅。

对于 **Azure NetApp Files** ，

注册 NetApp 资源提供商即可访问 Azure NetApp Files 。

步骤

1. 登录 Azure 门户。

2. "按照 Azure NetApp Files 文档注册 NetApp 资源提供商"。

Create a NetApp account

在 Azure NetApp Files 中创建 NetApp 帐户。

步骤

1. "按照 Azure NetApp Files 文档从 Azure 门户创建 NetApp 帐户"。

Set up a capacity pool

需要一个或多个容量池，这样 Astra 控制服务才能在容量池中配置永久性卷。Astra Control Service 不会为您创建容量池。

在为 Kubernetes 应用程序设置容量池时，请考虑以下事项：

- 需要在将使用 Astra Control Service 管理 AKS 集群的同一 Azure 区域中创建容量池。
- 容量池可以具有 " 超 " ， " 高级 " 或 " 标准 " 服务级别。其中每个服务级别都是为满足不同的性能需求而设计的。Astra 控制服务支持所有这三项功能。

  您需要为要在 Kubernetes 集群中使用的每个服务级别设置一个容量池。

  "详细了解 Azure NetApp Files 的服务级别"。

- 在为要使用 Astra Control Service 保护的应用程序创建容量池之前，请为这些应用程序选择所需的性能和容量。

  配置适当的容量可确保用户可以根据需要创建永久性卷。如果容量不可用，则无法配置永久性卷。

- Azure NetApp Files 容量池可以使用手动或自动 QoS 类型。Astra 控制服务支持自动 QoS 容量池。不支持手动 QoS 容量池。

步骤

1. "按照 Azure NetApp Files 文档设置自动 QoS 容量池"。

Delegate a subnet to Azure NetApp Files

您需要将子网委派给 Azure NetApp Files ，以便 Astra 控制服务可以在该子网中创建永久性卷。请注意，通过 Azure NetApp Files ，您只能在一个 vNet 中拥有一个委派子网。

如果您使用的是对等 VNets ，则对等连接的两端必须处于联机状态： Kubernetes 集群所在的 vNet 和已委派 Azure NetApp Files 子网的 vNet 。

步骤

1. "按照 Azure NetApp Files 文档将子网委派给 Azure NetApp Files"。

完成后

等待大约 10 分钟，然后发现在委派子网中运行的集群。

创建 **Azure** 服务主体

Astra 控制服务需要分配有贡献者角色的 Azure 服务主体。Astra 控制服务使用此服务主体来代表您促进 Kubernetes 应用程序数据管理。

服务主体是指专为应用程序，服务和工具而创建的身份。为服务主体分配角色将限制对特定 Azure 资源的访问。

按照以下步骤使用 Azure 命令行界面创建服务主体。您需要将输出保存在 JSON 文件中，并稍后将其提供给 Astra Control Service 。 "有关使用 CLI 的详细信息，请参见 Azure 文档"。

以下步骤假定您有权创建服务主体，并且计算机上已安装 Microsoft Azure SDK （AZ 命令）。

要求

- 服务主体必须使用常规身份验证。不支持证书。
- 必须为服务主体授予对您的 Azure 订阅的贡献者或所有者访问权限。
- 您为范围选择的订阅或资源组必须包含 AKS 集群和您的 Azure NetApp Files 帐户。

步骤

1. 确定 AKS 集群所在的订阅和租户 ID （这些集群是您要在 Astra Control Service 中管理的集群）。

   ```
   az configure --list-defaults
   az account list --output table
   ```

2. 根据您使用的是整个订阅还是资源组，执行以下操作之一：
   - 创建服务主体，分配 " 贡献者 " 角色，并指定集群所在的整个订阅的范围。

     ```
     az ad sp create-for-rbac --name service-principal-name --role
     contributor --scopes /subscriptions/SUBSCRIPTION-ID
     ```

   - 创建服务主体，分配 " 贡献者 " 角色，并指定集群所在的资源组。

     ```
     az ad sp create-for-rbac --name service-principal-name --role
     contributor --scopes /subscriptions/SUBSCRIPTION-
     ID/resourceGroups/RESOURCE-GROUP-ID
     ```

3. 将生成的 Azure 命令行界面输出存储为 JSON 文件。

   您需要提供此文件，以便 Astra Control Service 能够发现您的 AKS 集群并管理 Kubernetes 数据管理操作。 "了解如何在 Astra Control Service 中管理凭据"。

4. 可选：将订阅 ID 添加到 JSON 文件中，以便 Astra 控制服务在您选择此文件时自动填充此 ID 。

   否则，您需要在出现提示时在 Astra Control Service 中输入订阅 ID 。

- 示例 *

```
{
  "appId": "0db3929a-bfb0-4c93-baee-aaf8",
  "displayName": "sp-example-dev-sandbox",
  "name": "http://sp-example-dev-sandbox",
  "password": "mypassword",
  "tenant": "011cdf6c-7512-4805-aaf8-7721afd8ca37",
  "subscriptionId": "99ce999a-8c99-99d9-a9d9-99cce99f99ad"
}
```

5. 可选：测试您的服务主体。根据您的服务主体使用的范围，从以下示例命令中进行选择。

   订阅范围

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az group list --subscription SUBSCRIPTION-ID
az aks list --subscription SUBSCRIPTION-ID
az storage container list --account-name STORAGE-ACCOUNT-NAME
```

   资源组范围

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az aks list --subscription SUBSCRIPTION-ID --resource-group RESOURCE-
GROUP-ID
```

可选：为**Azure**备份存储分段配置冗余

您可以为Azure备份存储分段配置更持久的冗余级别。默认情况下、Astra Control Service用于存储Azure Kubernetes Service备份的存储分段使用本地冗余存储(LRS)冗余选项。要对Azure存储分段使用更持久的冗余选项、您需要执行以下操作：

步骤

1. 使用创建使用所需冗余级别的Azure存储帐户 "这些说明"。
2. 使用在新存储帐户中创建Azure容器 "这些说明"。
3. 将此容器作为分段添加到Astra Control Service中。请参见 "添加一个额外的存储分段"。
4. (可选)要使用新创建的存储分段作为Azure备份的默认存储分段、请将其设置为Azure的默认存储分段。请参见 "更改默认分段"。

使用 **Azure** 受管磁盘设置 **Microsoft Azure**

在使用 Astra Control Service 管理 Azure Kubernetes Service 集群之前，需要执行一些步

骤来准备 Microsoft Azure 订阅。如果您计划使用 Azure 托管磁盘作为存储后端，请按照以下说明进行操作。

快速开始设置 **Azure**

按照以下步骤快速入门，或者向下滚动到其余部分以了解完整详细信息。

[一个] 查看 **Azure Kubernetes Service** 的 **Astra Control Service** 要求

确保集群运行状况良好并运行受支持的Kubernetes版本、节点池处于联机状态并运行Linux等。 了解有关此步骤的更多信息。

[两个] 注册 **Microsoft Azure**

创建 Microsoft Azure 帐户。 了解有关此步骤的更多信息。

[三个] 创建 **Azure** 服务主体

创建具有 " 贡献者 " 角色的 Azure 服务主体。 了解有关此步骤的更多信息。

[四个] 配置容器存储接口（ **CSI** ）驱动程序详细信息

您需要配置 Azure 订阅和集群以使用 CSI 驱动程序。 了解有关此步骤的更多信息。

[五个] 可选：为**Azure**备份存储分段配置冗余

默认情况下、Astra Control Service用于存储Azure Kubernetes Service备份的存储分段使用本地冗余存储(LRS)冗余选项。作为一个可选步骤、您可以为Azure存储分段配置更持久的冗余级别。 了解有关此步骤的更多信息。

Azure Kubernetes Service 集群要求

Kubernetes 集群必须满足以下要求，才能通过 Astra Control Service 发现和管理它。

Kubernetes 版本

集群必须运行Kubnetes 1.26至1.28版。

映像类型

所有节点池的映像类型必须为 Linux 。

集群状态

集群必须运行状况良好，并且至少有一个联机辅助节点，并且没有处于故障状态的辅助节点。

Azure 区域

作为最佳实践，您应选择一个支持 Azure NetApp Files 的区域，即使您不将其用作存储后端也是如此。这样，如果性能要求发生变化，将来可以更轻松地将 Azure NetApp Files 用作存储后端。 "按区域查看 Azure 产品"。

CSI 驱动程序

集群必须安装适当的 CSI 驱动程序。

注册 **Microsoft Azure**

如果您没有 Microsoft Azure 帐户，请先注册 Microsoft Azure 。

步骤

1. 转至 "Azure 订阅页面" 订阅 Azure 服务。
2. 选择一个计划并按照说明完成订阅。

创建 **Azure** 服务主体

Astra 控制服务需要分配有贡献者角色的 Azure 服务主体。Astra 控制服务使用此服务主体来代表您促进 Kubernetes 应用程序数据管理。

服务主体是指专为应用程序，服务和工具而创建的身份。为服务主体分配角色将限制对特定 Azure 资源的访问。

按照以下步骤使用 Azure 命令行界面创建服务主体。您需要将输出保存在 JSON 文件中，并稍后将其提供给 Astra Control Service 。 "有关使用 CLI 的详细信息，请参见 Azure 文档"。

以下步骤假定您有权创建服务主体，并且计算机上已安装 Microsoft Azure SDK （AZ 命令）。

要求

- 服务主体必须使用常规身份验证。不支持证书。
- 必须为服务主体授予对您的 Azure 订阅的贡献者或所有者访问权限。
- 您为范围选择的订阅或资源组必须包含 AKS 集群和您的 Azure NetApp Files 帐户。

步骤

1. 确定 AKS 集群所在的订阅和租户 ID （这些集群是您要在 Astra Control Service 中管理的集群）。

```
az configure --list-defaults
az account list --output table
```

2. 根据您使用的是整个订阅还是资源组，执行以下操作之一：
   - 创建服务主体，分配 " 贡献者 " 角色，并指定集群所在的整个订阅的范围。

     ```
     az ad sp create-for-rbac --name service-principal-name --role
     contributor --scopes /subscriptions/SUBSCRIPTION-ID
     ```

   - 创建服务主体，分配 " 贡献者 " 角色，并指定集群所在的资源组。

     ```
     az ad sp create-for-rbac --name service-principal-name --role
     contributor --scopes /subscriptions/SUBSCRIPTION-
     ID/resourceGroups/RESOURCE-GROUP-ID
     ```

3. 将生成的 Azure 命令行界面输出存储为 JSON 文件。

   您需要提供此文件，以便 Astra Control Service 能够发现您的 AKS 集群并管理 Kubernetes 数据管理操作。"了解如何在 Astra Control Service 中管理凭据"。

4. 可选：将订阅 ID 添加到 JSON 文件中，以便 Astra 控制服务在您选择此文件时自动填充此 ID 。

   否则，您需要在出现提示时在 Astra Control Service 中输入订阅 ID 。

   - 示例 *

```
{
  "appId": "0db3929a-bfb0-4c93-baee-aaf8",
  "displayName": "sp-example-dev-sandbox",
  "name": "http://sp-example-dev-sandbox",
  "password": "mypassword",
  "tenant": "011cdf6c-7512-4805-aaf8-7721afd8ca37",
  "subscriptionId": "99ce999a-8c99-99d9-a9d9-99cce99f99ad"
}
```

5. 可选：测试您的服务主体。根据您的服务主体使用的范围，从以下示例命令中进行选择。

   订阅范围

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az group list --subscription SUBSCRIPTION-ID
az aks list --subscription SUBSCRIPTION-ID
az storage container list --account-name STORAGE-ACCOUNT-NAME
```

   资源组范围

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az aks list --subscription SUBSCRIPTION-ID --resource-group RESOURCE-
GROUP-ID
```

配置容器存储接口（ **CSI** ）驱动程序详细信息

要将Azure受管磁盘与Astra Control Service结合使用、您需要安装所需的CSI驱动程序。

在 **Azure** 订阅中启用 **CSI** 驱动程序功能

在安装 CSI 驱动程序之前，您需要在 Azure 订阅中启用 CSI 驱动程序功能。

步骤

1. 打开 Azure 命令行界面。
2. 运行以下命令以注册驱动程序：

```
az feature register --namespace "Microsoft.ContainerService" --name
"EnableAzureDiskFileCSIDriver"
```

3. 运行以下命令以确保更改已传播：

```
az provider register -n Microsoft.ContainerService
```

   您应看到类似于以下内容的输出：

```
{
"id": "/subscriptions/b200155f-001a-43be-87be-
3edde83acef4/providers/Microsoft.Features/providers/Microsoft.ContainerSer
vice/features/EnableAzureDiskFileCSIDriver",
"name": "Microsoft.ContainerService/EnableAzureDiskFileCSIDriver",
"properties": {
   "state": "Registering"
},
"type": "Microsoft.Features/providers/features"
}
```

在 **Azure Kubernetes Service** 集群中安装 **Azure** 托管磁盘 **CSI** 驱动程序

您可以安装 Azure CSI 驱动程序以完成准备工作。

步骤

1. 转至 "Microsoft CSI 驱动程序文档"。
2. 按照说明安装所需的 CSI 驱动程序。

可选：为**Azure**备份存储分段配置冗余

您可以为Azure备份存储分段配置更持久的冗余级别。默认情况下、Astra Control Service用于存储Azure Kubernetes Service备份的存储分段使用本地冗余存储(LRS)冗余选项。要对Azure存储分段使用更持久的冗余选项、您需要执行以下操作：

步骤

1. 使用创建使用所需冗余级别的Azure存储帐户 "这些说明"。
2. 使用在新存储帐户中创建Azure容器 "这些说明"。
3. 将此容器作为分段添加到Astra Control Service中。请参见 "添加一个额外的存储分段"。
4. (可选)要使用新创建的存储分段作为Azure备份的默认存储分段、请将其设置为Azure的默认存储分段。请参

见 "更改默认分段"。

注册 Astra Control Service 帐户

要使用Astra控制服务、您需要一个与您的NetApp BlueXP帐户关联的Astra控制服务帐户。完成Astra Control Service注册过程、然后、如果您还没有BlueXP帐户、请注册BlueXP以访问Astra Control Service。

注册 Astra Control 帐户

在登录到 Astra 控制服务之前，您需要完成注册过程才能获取 Astra 控制服务帐户。

使用 Astra Control Service 时，您可以从帐户中管理应用程序。帐户包括可以查看和管理帐户中的应用程序以及您的计费详细信息的用户。

步骤

1. "转到BlueXP上的Asta Control页面"。
2. 选择*注册免费版*。
3. 在表单中提供所需信息。

   填写表单时需要注意的一些重要事项：

   - 您的企业名称和地址必须准确无误，因为我们会对其进行验证，以满足全球贸易合规性的要求。
   - * Astra 帐户名称 * 是您企业的 Astra Control Service 帐户的名称。您将在 Astra Control Service 用户界面中看到此名称。请注意，如果需要，您可以创建其他帐户（最多 5 个）。
   - 在*企业电子邮件地址*字段中，如果您有NetApp BlueXP帐户，请在此处输入用于该帐户的电子邮件。如果您还没有NetApp BlueXP帐户、请使用注册BlueXP时在此处输入的电子邮件地址。
4. 选择 * 创建帐户 * 。

注册到BlueXP

Astra控制服务集成在NetApp BlueXP的身份验证服务中。您可以使用BlueXP或NetApp NetApp 支持站点 凭据登录到BlueXP。如果您还没有NetApp BlueXP或NetApp 支持站点 帐户、请注册BlueXP、以便访问Astra Control服务和NetApp的其他云服务。如果您已有BlueXP或NetApp 支持站点 帐户并已完成注册、则可以访问 "Astra 控制服务" 直接使用BlueXP或NetApp 支持站点 凭据。

> 您还可以使用公司目录中的凭据(联合身份)通过单点登录登录到BlueXP。要了解更多信息，请转到 "帮助中心" 然后选择 * Cloud Central 登录选项 * 。

步骤

1. 转至 "NetApp BlueXP"。
2. 在右上角，选择*Get started*。
3. 选择 * 注册 * 。
4. 填写表单。

确保您在此处输入的电话号码和电子邮件地址与您在前一个Astra Control注册表中使用的地址相同。

5. 选择 * 注册 * 。

> 您在这些表单中输入的电子邮件地址是您的NetApp BlueXP用户ID的电子邮件地址。当您注册新的Astra Control帐户或Astra Control管理员邀请您加入现有Astra Control帐户时、请使用此BlueXP用户ID。

6. 等待NetApp BlueXP发送的电子邮件。此电子邮件来自地址 saas.support@netapp.com ，可能需要几分钟才能收到。请务必检查您的垃圾邮件文件夹。
7. 电子邮件到达后，选择电子邮件中的链接以验证您的电子邮件地址。

结果

现在、您已拥有有效的BlueXP用户登录名。

注册后、您可以直接从使用BlueXP凭据访问ASRA Control https://astra.netapp.io。

将集群添加到Asta Control Service

设置环境后，您可以创建 Kubernetes 集群，然后将其添加到 Astra Control Service 。这样、您就可以使用Astra Control Service来保护集群上的应用程序。

根据您需要添加到Astra Control Service的集群类型、您需要使用不同的步骤来添加集群。

- "将公共提供商管理的集群添加到Astra Control Service"：使用以下步骤添加具有公共IP地址且由云提供商管理的集群。您需要云提供商的服务主体帐户、服务帐户或用户帐户。
- "将私有提供商管理的集群添加到Asta Control Service"：使用以下步骤添加具有专用IP地址且由云提供商管理的集群。您需要云提供商的服务主体帐户、服务帐户或用户帐户。
- "将公共自管理集群添加到Astra Control Service"：使用以下步骤添加具有公共IP地址且由您的组织管理的集群。您需要为要添加的集群创建一个kubeconfigfile文件。
- "将私有自管理集群添加到Asta Control Service"：使用以下步骤添加具有专用IP地址且由您的组织管理的集群。您需要为要添加的集群创建一个kubeconfigfile文件。

安装Asta Connector以管理集群

Asta Connector是一款位于受管集群上的软件、可促进受管集群与Asta Control之间的通信。对于使用Asta Control Service管理的集群、Asta Connector有两个可用版本：

- *Astra Connecter*的先前版本： "安装旧版本的Asta Connector" 如果您计划使用非Kubnetes本机工作流管理集群、请在集群上执行此操作。
- 声明性**Kubernetes Astra Connecter**： "为使用声明性Kubarnetes工作流管理的集群安装Asta Connector" 如果您计划使用声明性Kubarnetes工作流管理集群、请在集群上执行此操作。在集群上安装Asta Connector后、该集群将自动添加到Asta Control中。

> 声明性Kubarnetes Asta Connector仅作为Asta Control早期采用者计划(EAP)的一部分提供。有关加入NetApp的信息、请联系您的EAP销售代表。

安装旧版本的**Asta Connector**

Astra Control Service使用以前版本的Astra Connector在Astra Control Service和使用非Kubnetes本机工作流管理的专用集群之间实现通信。您需要在要使用非Kubbernetes本机工作流管理的专用集群上安装Astra Connector。

早期版本的Astra Connector支持使用非Kubenetes本机工作流管理的以下类型的专用集群：

- Amazon Elelic Kubelnetes Service (EKS)
- Azure Kubernetes Service （AKS）
- Google Kubernetes Engine （GKEE）
- 基于AWS的Red Hat OpenShift服务(ROSA)
- 具有AWS PrivateLink的罗莎
- Red Hat OpenShift容器平台内部部署

关于此任务

- 执行这些步骤时、请对要使用Astra Control Service管理的专用集群执行这些命令。
- 如果使用的是Bastion主机、请从Bastion主机的命令行对这些命令执行问题描述。

开始之前

- 您需要访问要使用Astra Control Service管理的专用集群。
- 要在集群上安装Asta Connector操作员、您需要具有Kubbernetes管理员权限。

步骤

1. 在要使用非Kubbernetes本机工作流管理的专用集群上安装先前的Astra Connector运算符。运行此命令时、命名空间 `astra-connector-operator` 创建并将配置应用于命名空间：

```
kubectl apply -f https://github.com/NetApp/astra-connector-operator/releases/download/23.07.0-202310251519/astraconnector_operator.yaml
```

2. 确认操作员已安装并准备就绪：

```
kubectl get all -n astra-connector-operator
```

3. 从Asta Control获取API令牌。请参见 "Astra Automation文档" 有关说明，请参见。
4. 创建Astra-connector命名空间：

```
kubectl create ns astra-connector
```

5. 创建Astra Connector CR文件并将其命名为 `astra-connector-cr.yaml`。更新方括号<>中的值以匹配您的Astra Control环境和集群配置：

- **Astra**：控制服务的Web UI <ASTRA_CONTROL_SERVICE_URL>。例如：

```
https://astra.netapp.io
```

- **Astra**：您在上一步中获得的<ASTRA_CONTROL_SERVICE_API_TOKEN>标记。
- **AzAzure Kubernet**：(仅限集群)—专用<PRIVATE_AKS_CLUSTER_NAME>集群的集群名称。只有在添加专用AKS集群时、才会取消注释并填充此行。
- **Astra**：从<ASTRA_CONTROL_ACCOUNT_ID> Web UI获取。选择页面右上角的图图标，然后选择*API access*。

```yaml
apiVersion: netapp.astraconnector.com/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  natssync-client:
    cloud-bridge-url: <ASTRA_CONTROL_SERVICE_URL>
  imageRegistry:
    name: theotw
    secret: ""
  astra:
    token: <ASTRA_CONTROL_SERVICE_API_TOKEN>
    #clusterName: <PRIVATE_AKS_CLUSTER_NAME>
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    acceptEULA: yes
```

6. 在您填充之后 `astra-connector-cr.yaml` 使用正确值的文件、应用CR：

```
kubectl apply -f astra-connector-cr.yaml
```

7. 验证Asta Connector是否已完全部署：

```
kubectl get all -n astra-connector
```

8. 验证集群是否已注册到Astra Control：

```
kubectl get astraconnector -n astra-connector
```

您应看到类似于以下内容的输出：

```
NAME                REGISTERED   ASTRACONNECTORID
STATUS
astra-connector     true         be475ae5-1511-4eaa-9b9e-712f09b0d065
Registered with Astra
```

记下ASTRACONNECTRID；将集群添加到Astra Control时将需要此ID。

下一步是什么?

现在您已安装Astra Connector、可以将私有集群添加到Astra Control Service了。

- "将私有提供商管理的集群添加到Asta Control Service"：使用以下步骤添加具有专用IP地址且由云提供商管理的集群。您需要云提供商的服务主体帐户、服务帐户或用户帐户。
- "将私有自管理集群添加到Asta Control Service"：使用以下步骤添加具有专用IP地址且由您的组织管理的集群。您需要为要添加的集群创建一个kubeconfigfile文件。

有关详细信息 ...

- "添加集群"

(技术预览)安装声明性Kubarnetes Asta Connector

使用声明性Kubarnetes工作流管理的集群使用Astra Connector在受管集群和Astra Control之间实现通信。您需要在要使用声明性Kubarnetes工作流管理的所有集群上安装Astra Connector。

您可以使用Kubarnetes命令和自定义资源(Custom Resource、CR)文件安装声明性Kubarnetes Asta Connector。

关于此任务

- 执行这些步骤时、请在要使用Astra Control进行管理的集群上执行这些命令。
- 如果使用的是Bastion主机、请从Bastion主机的命令行对这些命令执行问题描述。

开始之前

- 您需要访问要使用Astra Control管理的集群。
- 要在集群上安装Asta Connector操作员、您需要具有Kubbernetes管理员权限。

如果为集群配置了强制实施Pod安全接入(这是Kubernetes 1.25及更高版本集群的默认设置)、则需要对相应的卷空间启用PSA限制。请参见 "使用Astra Control准备用于集群管理的环境" 有关说明，请参见。

步骤

1. 在要使用声明性Kubarnetes工作流管理的集群上安装A作用 连接器操作员。运行此命令时、命名空间 `astra-connector-operator` 创建并将配置应用于命名空间：

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/24.02.0-
202403151353/astraconnector_operator.yaml
```

2. 确认操作员已安装并准备就绪：

```
kubectl get all -n astra-connector-operator
```

3. 从Asta Control获取API令牌。请参见 "Astra Automation文档" 有关说明，请参见。
4. 使用令牌创建密钥。将<API_TOKEN>替换为您从Astra Control收到的令牌：

```
kubectl create secret generic astra-token \
--from-literal=apiToken=<API_TOKEN> \
-n astra-connector
```

5. 创建Docker密钥以提取Astra Connector映像。将括号<>中的值替换为您环境中的信息：

> 您可以在Astra Control Web UI中找到<ASTRA_CONTROL_ACCOUNT_ID>。在Web UI中，选择页面右上角的图图标，然后选择*API access*。

```
kubectl create secret docker-registry regcred \
--docker-username=<ASTRA_CONTROL_ACCOUNT_ID> \
--docker-password=<API_TOKEN> \
-n astra-connector \
--docker-server=cr.astra.netapp.io
```

6. 创建Astra Connector CR文件并将其命名为 `astra-connector-cr.yaml`。更新方括号<>中的值以匹配您的Astra Control环境和集群配置：
   - <ASTRA_CONTROL_ACCOUNT_ID>：在上一步中从Astra Control Web UI获取。
   - <CLUSTER_NAME>：应在Asta Control中分配此集群的名称。
   - <ASTRA_CONTROL_URL>：Asta Control的Web UI URL。例如：

```
https://astra.control.url
```

```yaml
apiVersion: astra.netapp.io/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  astra:
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    clusterName: <CLUSTER_NAME>
    #Only set `skipTLSValidation` to `true` when using the default self-signed
    #certificate in a proof-of-concept environment.
    skipTLSValidation: false #Should be set to false in production environments
    tokenRef: astra-token
  natsSyncClient:
    cloudBridgeURL: <ASTRA_CONTROL_HOST_URL>
  imageRegistry:
    name: cr.astra.netapp.io
    secret: regcred
```

7. 在您填充之后 `astra-connector-cr.yaml` 使用正确值的文件、应用CR：

```
kubectl apply -n astra-connector -f astra-connector-cr.yaml
```

8. 验证Asta Connector是否已完全部署：

```
kubectl get all -n astra-connector
```

9. 验证集群是否已注册到Astra Control：

```
kubectl get astraconnectors.astra.netapp.io -A
```

您应看到类似于以下内容的输出：

```
NAMESPACE          NAME               REGISTERED   ASTRACONNECTORID
STATUS
astra-connector    astra-connector    true         00ac8-2cef-41ac-8777-
ed0583e   Registered with Astra
```

10. 验证该集群是否显示在Astra Control Web UI的*集群*页面上的受管集群列表中。

添加由提供程序管理的集群

将公共提供商管理的集群添加到**Astra Control Service**

设置云环境后、您可以创建Kubennetes集群、然后将其添加到Astra Control Service中。

- 创建 Kubernetes 集群
- 将集群添加到Asta Control Service
- [更改默认存储类]

创建 **Kubernetes** 集群

如果您还没有集群、则可以创建满足要求的集群 "Amazon Elastic Kubernetes Service (EKS)的Astra Control Service要求"。如果您还没有集群、则可以创建满足要求的集群 "Google Kubernetes Engine （ GKEE ） 的 Astra Control Service 要求"。如果您还没有集群、则可以创建满足要求的集群 "采用 Azure NetApp Files 的 Azure Kubernetes Service （ AKS ） 的 Astra 控制服务要求" 或 "采用 Azure 受管磁盘的 Azure Kubernetes Service （ AKS ） 的 Astra Control Service 要求"。

> Astra控制服务支持使用Azure Active Directory (Azure AD)进行身份验证和身份管理的AKS集群。创建集群时、请按照中的说明进行操作 "正式文档" 将集群配置为使用Azure AD。您需要确保集群满足AKS管理的Azure AD集成的要求。

将集群添加到**Asta Control Service**

登录到 Astra Control Service 后，第一步是开始管理集群。在将集群添加到Astra Control Service之前、您需要执行特定任务并确保集群满足特定要求。

在管理Azure Kubenetes Service和Google Kubenetes Engine集群时、请注意、Astra Control配置程序安装和生命周期管理有两种选择：

- 您可以使用Astra Control Service自动管理Astra Control置管程序的生命周期。要执行此操作、请确保未在要使用Astra Control Service管理的集群上安装Astra Trent、并且未启用Astra Control配置程序。在这种情况下、Astra Control Service会在您开始管理集群时自动启用Astra Control配置程序、并自动处理Astra Control配置程序升级。
- 您可以自行管理Astra Control配置程序的生命周期。为此、请先在集群上启用Asta Control配置程序、然后再使用Asta Control Service管理集群。在这种情况下、Astra Control Service检测到Astra Control配置程序已启用、不会重新安装它或管理Astra Control配置程序升级。请参见 "启用Asta Control配置程序" 有关步骤、请启用Asta Control配置程序。

在使用Asta Control Service管理Amazon Web Services集群时、如果需要只能与Asta Control配置程序一起使用的存储后端、则需要先在集群上手动启用Asta Control配置程序、然后再使用Asta Control Service进行管理。请参见 "启用Asta Control配置程序" 了解启用Astra Control配置程序的步骤。

开始之前

Amazon Web Services

- 您应拥有包含创建集群的IAM用户凭据的JSON文件。 "了解如何创建IAM用户"。
- Amazon FSx for NetApp ONTAP需要Astra Control配置程序。如果您计划使用Amazon FSx for NetApp ONTAP作为EKS集群的存储后端、请参阅中的Astra Control配置程序信息 "EKS集群要求"。
- (可选)如果需要提供 `kubectl` 集群对非集群创建者的其他IAM用户的命令访问权限、请参见中的说明 "在Amazon EKS中创建集群后、如何为其他IAM用户和角色提供访问权限？"。
- 如果您计划使用NetApp Cloud Volumes ONTAP 作为存储后端、则需要将Cloud Volumes ONTAP 配置为使用Amazon Web Services。请参见Cloud Volumes ONTAP "设置文档"。

Microsoft Azure

- 您应拥有包含在创建服务主体时Azure命令行界面输出的JSON文件。 "了解如何设置服务主体"。

  如果未将 Azure 订阅 ID 添加到 JSON 文件中，您也需要此 ID 。

- 如果您计划使用NetApp Cloud Volumes ONTAP 作为存储后端、则需要将Cloud Volumes ONTAP 配置为与Microsoft Azure配合使用。请参见Cloud Volumes ONTAP "设置文档"。

Google Cloud

- 您应拥有具有所需权限的服务帐户的服务帐户密钥文件。 "了解如何设置服务帐户"。
- 如果您计划使用NetApp Cloud Volumes ONTAP 作为存储后端、则需要将Cloud Volumes ONTAP 配置为与Google Cloud配合使用。请参见Cloud Volumes ONTAP "设置文档"。

步骤

1. (可选)如果要添加Amazon EKS集群或要自行管理Astra Control配置程序的安装和升级、请在此集群上启用Astra Control配置程序。请参见 "启用Asta Control配置程序" 了解支持步骤。
2. 在浏览器中打开Asta Control Service Web UI。
3. 在信息板上，选择 * 管理 Kubernetes 集群 * 。

   按照提示添加集群。

4. 提供商：选择您的云提供商、然后提供创建新云实例所需的凭据或选择要使用的现有云实例。
5. * Amazon Web Services*：上传JSON文件或从剪贴板粘贴JSON文件的内容、以提供有关Amazon Web Services IAM用户帐户的详细信息。

   JSON文件应包含创建集群的IAM用户的凭据。

6. * Microsoft Azure* ：通过上传 JSON 文件或从剪贴板粘贴此 JSON 文件的内容来提供有关 Azure 服务主体的详细信息。

   JSON 文件应包含创建服务主体时 Azure 命令行界面的输出。它还可以包含您的订阅 ID ，以便自动添加到 Astra 。否则，您需要在提供 JSON 后手动输入 ID 。

7. * Google Cloud Platform* ：通过上传文件或粘贴剪贴板中的内容来提供服务帐户密钥文件。

   Astra 控制服务使用此服务帐户发现在 Google Kubernetes Engine 中运行的集群。

8. 其他：此选项卡仅适用于自行管理的集群。
   a. 云实例名称：为要在添加此集群时创建的新云实例提供一个名称。了解更多信息 "云实例"。
   b. 选择 * 下一步 * 。

      Astra Control Service会显示一个集群列表、您可以从中进行选择。

   c. 集群：从列表中选择要添加到Astra Control Service的集群。

      > 从集群列表中选择时，请注意*Eligiblity*列。如果集群"不符合条件"或"部分符合条件"、请将鼠标悬停在状态上方以确定集群是否具有问题描述。例如，它可能会标识集群没有工作节点。

   d. 选择 * 下一步 * 。
   e. (可选)存储：(可选)选择默认情况下希望部署到此集群中的Kubernetes应用程序使用的存储类。
9. 要为集群选择新的默认存储类，请启用*Assign a new default storage class*复选框。
10. 从列表中选择新的默认存储类。

    > 每个云提供商存储服务都会显示以下价格、性能和弹性信息：
    >
    > - Cloud Volumes Service for Google Cloud：价格、性能和弹性信息
    > - Google Persistent Disk：没有价格、性能或弹性信息
    > - Azure NetApp Files ：性能和弹性信息
    > - Azure受管磁盘：无可用的价格、性能或弹性信息
    > - Amazon Elastic Block Store：没有价格、性能或弹性信息
    > - 适用于NetApp ONTAP 的Amazon FSX：没有价格、性能或弹性信息
    > - NetApp Cloud Volumes ONTAP ：没有价格、性能或弹性信息

    每个存储类均可使用以下服务之一：

    - "适用于 Google Cloud 的 Cloud Volumes Service"
    - "Google 持久磁盘"
      - "Azure NetApp Files"
      - "Azure 受管磁盘"
      - "Amazon Elastic Block Store"
      - "适用于 NetApp ONTAP 的 Amazon FSX"
      - "NetApp Cloud Volumes ONTAP"

      了解更多信息 "Amazon Web Services集群的存储类"。了解更多信息 "AKS 集群的存储类"。了解更多信息 "GKE 集群的存储类"。

      a. 选择 * 下一步 * 。
      b. 审核和批准：审核配置详细信息。

c. 选择*Add*将集群添加到Astra Control Service。

结果

如果这是您为此云提供程序添加的第一个集群、Astra Control Service将为此云提供程序创建一个对象存储、用于备份在符合条件的集群上运行的应用程序。(在为此云提供程序添加后续集群时、不会再创建其他对象存储。)如果指定了默认存储类、则Astra控制服务将设置您指定的默认存储类。对于在Amazon Web Services或Google Cloud Platform中管理的集群、Astra Control Service还会在集群上创建管理员帐户。这些操作可能需要几分钟时间。

更改默认存储类

您可以更改集群的默认存储类。

使用**Astra Control**更改默认存储类

您可以在Astra Control中更改集群的默认存储类。如果集群使用先前安装的存储后端服务、则可能无法使用此方法更改默认存储类(不能选择*设置为默认值*操作)。在这种情况下、您可以 [使用命令行更改默认存储类]。

步骤

1. 在 Astra 控制服务 UI 中，选择 * 集群 * 。
2. 在*集群*页面上、选择要更改的集群。
3. 选择 * 存储 * 选项卡。
4. 选择*存储类*类别。
5. 选择要设置为默认值的存储类的*操作*菜单。
6. 选择*设置为默认值*。

使用命令行更改默认存储类

您可以使用Kubernetes命令更改集群的默认存储类。无论集群的配置如何、此方法都有效。

步骤

1. 登录到Kubernetes集群。
2. 列出集群中的存储类：

```
kubectl get storageclass
```

3. 从默认存储类中删除默认指定。将<SC_NAME> 替换为存储类的名称：

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. 将其他存储类标记为默认值。将<SC_NAME> 替换为存储类的名称：

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. 确认新的默认存储类：

```
kubectl get storageclass
```

将私有提供商管理的集群添加到**Asta Control Service**

您可以使用Astra Control Service管理专用Google Kubbernetes Engine (GKE)集群。以下说明假定您已创建专用AKS或OpenShift集群、并准备了一种安全方法来远程访问该集群；有关创建和访问专用AKS或OpenShift集群的详细信息、请参阅以下文档：

- "适用于私有AKS集群的Azure文档"
- "适用于私有OpenShift集群的Azure文档"

您可以使用Astra Control Service管理私有Azure Kubbernetes Service (AKS)集群以及AKS中的私有Red Hat OpenShift集群。以下说明假定您已创建专用AKS或OpenShift集群、并准备了一种安全方法来远程访问该集群；有关创建和访问专用AKS或OpenShift集群的详细信息、请参阅以下文档：

- "适用于私有AKS集群的Azure文档"
- "适用于私有OpenShift集群的Azure文档"

您可以使用Astra Control Service管理专用Amazon Elastic Kubelnetes Service (EKS)集群。以下说明假定您已创建一个专用EKS集群、并已准备好一种安全方法来远程访问该集群；有关创建和访问专用EKS集群的详细信息、请参阅 "Amazon EKS文档"。

要将专用集群添加到Astra Control Service、您需要执行以下任务：

1. 安装A作用 连接器
2. [设置永久性存储]
3. 将私有提供商管理的集群添加到Asta Control Service

安装**A**作用 连接器

在添加专用集群之前、您需要在此集群上安装Astra Connector、以便Astra Control可以与其通信。请参见 "为使用非Kubnetes本机工作流管理的专用集群安装以前版本的Astra Connector" 有关说明，请参见。

设置永久性存储

为集群配置永久性存储。有关配置永久性存储的详细信息、请参见入门文档：

- "使用 Azure NetApp Files 设置 Microsoft Azure"
- "使用 Azure 受管磁盘设置 Microsoft Azure"
- "设置Amazon Web Services"

- "设置 Google Cloud"

将私有提供商管理的集群添加到**Asta Control Service**

现在、您可以将专用集群添加到Astra Control Service。

在管理Azure Kubenetes Service和Google Kubenetes Engine集群时、请注意、Astra Control配置程序安装和生命周期管理有两种选择：

- 您可以使用Astra Control Service自动管理Astra Control置管程序的生命周期。要执行此操作、请确保未在要使用Astra Control Service管理的集群上安装Astra Trent、并且未启用Astra Control配置程序。在这种情况下、Astra Control Service会在您开始管理集群时自动启用Astra Control配置程序、并自动处理Astra Control配置程序升级。
- 您可以自行管理Astra Control配置程序的生命周期。为此、请先在集群上启用Asta Control配置程序、然后再使用Asta Control Service管理集群。在这种情况下、Astra Control Service检测到Astra Control配置程序已启用、不会重新安装它或管理Astra Control配置程序升级。请参见 "启用Asta Control配置程序" 有关步骤、请启用Asta Control配置程序。

在使用Asta Control Service管理Amazon Web Services集群时、如果需要只能与Asta Control配置程序一起使用的存储后端、则需要先在集群上手动启用Asta Control配置程序、然后再使用Asta Control Service进行管理。请参见 "启用Asta Control配置程序" 了解启用Astra Control配置程序的步骤。

开始之前

Amazon Web Services

- 您应拥有包含创建集群的IAM用户凭据的JSON文件。 "了解如何创建IAM用户"。
- Amazon FSx for NetApp ONTAP需要Astra Control配置程序。如果您计划使用Amazon FSx for NetApp ONTAP作为EKS集群的存储后端、请参阅中的Astra Control配置程序信息 "EKS集群要求"。
- (可选)如果需要提供 `kubectl` 集群对非集群创建者的其他IAM用户的命令访问权限、请参见中的说明 "在Amazon EKS中创建集群后、如何为其他IAM用户和角色提供访问权限？"。
- 如果您计划使用NetApp Cloud Volumes ONTAP 作为存储后端、则需要将Cloud Volumes ONTAP 配置为使用Amazon Web Services。请参见Cloud Volumes ONTAP "设置文档"。

Microsoft Azure

- 您应拥有包含在创建服务主体时Azure命令行界面输出的JSON文件。 "了解如何设置服务主体"。

  如果未将 Azure 订阅 ID 添加到 JSON 文件中，您也需要此 ID 。

- 如果您计划使用NetApp Cloud Volumes ONTAP 作为存储后端、则需要将Cloud Volumes ONTAP 配置为与Microsoft Azure配合使用。请参见Cloud Volumes ONTAP "设置文档"。

Google Cloud

- 您应拥有具有所需权限的服务帐户的服务帐户密钥文件。 "了解如何设置服务帐户"。
- 如果集群为专用集群，则会显示 "授权网络" 必须允许 Astra 控制服务 IP 地址：

  52.188.218.166/32

- 如果您计划使用NetApp Cloud Volumes ONTAP 作为存储后端、则需要将Cloud Volumes ONTAP 配置为与Google Cloud配合使用。请参见Cloud Volumes ONTAP "设置文档"。

步骤

1. (可选)如果要添加Amazon EKS集群或要自行管理Astra Control配置程序的安装和升级、请在此集群上启用Astra Control配置程序。请参见 "启用Asta Control配置程序" 了解支持步骤。
2. 在浏览器中打开Asta Control Service Web UI。
3. 在信息板上，选择 * 管理 Kubernetes 集群 * 。

   按照提示添加集群。

4. 提供商：选择您的云提供商、然后提供创建新云实例所需的凭据或选择要使用的现有云实例。
5. * Amazon Web Services*：上传JSON文件或从剪贴板粘贴JSON文件的内容、以提供有关Amazon Web Services IAM用户帐户的详细信息。

   JSON文件应包含创建集群的IAM用户的凭据。

6. * Microsoft Azure* ：通过上传 JSON 文件或从剪贴板粘贴此 JSON 文件的内容来提供有关 Azure 服务主体的详细信息。

   JSON 文件应包含创建服务主体时 Azure 命令行界面的输出。它还可以包含您的订阅 ID ，以便自动添加到 Astra 。否则，您需要在提供 JSON 后手动输入 ID 。

7. * Google Cloud Platform* ：通过上传文件或粘贴剪贴板中的内容来提供服务帐户密钥文件。

   Astra 控制服务使用此服务帐户发现在 Google Kubernetes Engine 中运行的集群。

8. 其他：此选项卡仅适用于自行管理的集群。
   a. 云实例名称：为要在添加此集群时创建的新云实例提供一个名称。了解更多信息 "云实例"。
   b. 选择 * 下一步 * 。

      Astra Control Service会显示一个集群列表、您可以从中进行选择。

   c. 集群：从列表中选择要添加到Astra Control Service的集群。

      > 从集群列表中选择时，请注意*Eligiblity*列。如果集群"不符合条件"或"部分符合条件"、请将鼠标悬停在状态上方以确定集群是否具有问题描述。例如，它可能会标识集群没有工作节点。

9. 选择 * 下一步 * 。
10. (可选)存储：(可选)选择默认情况下希望部署到此集群中的Kubernetes应用程序使用的存储类。
    a. 要为集群选择新的默认存储类，请启用*Assign a new default storage class*复选框。
    b. 从列表中选择新的默认存储类。

每个云提供商存储服务都会显示以下价格、性能和弹性信息：

- Cloud Volumes Service for Google Cloud：价格、性能和弹性信息
- Google Persistent Disk：没有价格、性能或弹性信息
- Azure NetApp Files ：性能和弹性信息
- Azure受管磁盘：无可用的价格、性能或弹性信息
- Amazon Elastic Block Store：没有价格、性能或弹性信息
- 适用于NetApp ONTAP 的Amazon FSX：没有价格、性能或弹性信息
- NetApp Cloud Volumes ONTAP ：没有价格、性能或弹性信息

每个存储类均可使用以下服务之一：

- "适用于 Google Cloud 的 Cloud Volumes Service"
- "Google 持久磁盘"
- "Azure NetApp Files"
- "Azure 受管磁盘"
- "Amazon Elastic Block Store"
- "适用于 NetApp ONTAP 的 Amazon FSX"
- "NetApp Cloud Volumes ONTAP"

  了解更多信息 "Amazon Web Services集群的存储类"。了解更多信息 "AKS 集群的存储类"。了解更多信息 "GKE 集群的存储类"。

c. 选择 * 下一步 * 。

d. 审核和批准：审核配置详细信息。

e. 选择*Add*将集群添加到Astra Control Service。

结果

如果这是您为此云提供程序添加的第一个集群、Astra Control Service将为此云提供程序创建一个对象存储、用于备份在符合条件的集群上运行的应用程序。(在为此云提供程序添加后续集群时、不会再创建其他对象存储。)如果指定了默认存储类、则Astra控制服务将设置您指定的默认存储类。对于在Amazon Web Services或Google Cloud Platform中管理的集群、Astra Control Service还会在集群上创建管理员帐户。这些操作可能需要几分钟时间。

更改默认存储类

您可以更改集群的默认存储类。

使用**Astra Control**更改默认存储类

您可以在Astra Control中更改集群的默认存储类。如果集群使用先前安装的存储后端服务、则可能无法使用此方法更改默认存储类(不能选择*设置为默认值*操作)。在这种情况下、您可以 [使用命令行更改默认存储类]。

步骤

1. 在 Astra 控制服务 UI 中，选择 * 集群 * 。
2. 在*集群*页面上、选择要更改的集群。
3. 选择 * 存储 * 选项卡。
4. 选择*存储类*类别。
5. 选择要设置为默认值的存储类的*操作*菜单。
6. 选择*设置为默认值*。

使用命令行更改默认存储类

您可以使用Kubernetes命令更改集群的默认存储类。无论集群的配置如何、此方法都有效。

步骤

1. 登录到Kubernetes集群。
2. 列出集群中的存储类：

```
kubectl get storageclass
```

3. 从默认存储类中删除默认指定。将<SC_NAME> 替换为存储类的名称：

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. 将其他存储类标记为默认值。将<SC_NAME> 替换为存储类的名称：

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. 确认新的默认存储类：

```
kubectl get storageclass
```

添加自管理集群

将公共自管理集群添加到**Astra Control Service**

设置环境后，您可以创建 Kubernetes 集群，然后将其添加到 Astra Control Service 。

自管理集群是指直接配置和管理的集群。Astra Control Service支持在公共云环境中运行的自管理集群。您可以通过上传将自管理集群添加到Astra Control Service `kubeconfig.yaml` 文件您需要确保集群满足此处所述的要求。

支持的Kubnetes分发版

您可以使用Astra Control Service管理以下类型的公共自管理集群：

Kubbernetes分发	支持的版本
Kubnetes (上游)	1.27至1.29
Rancher Kubernetes Engine （RKE）	RKE 1：版本1.24.17、1.25.13、1.26.8、带RANcher Manager 2.7.9 RKE 2：版本1.23.16和1.24.13、带Randcher Manager 2.6.13 RKE 2：版本1.24.17、1.25.14、1.26.9、带RANcher Manager 2.7.9
Red Hat OpenShift 容器平台	4.12至4.14

以下说明假定您已创建一个自行管理的集群。

- 将集群添加到Asta Control Service
- [更改默认存储类]

将集群添加到Asta Control Service

登录到 Astra Control Service 后，第一步是开始管理集群。在将集群添加到Astra Control Service之前、您需要执行特定任务并确保集群满足特定要求。

开始之前

自管理集群是指直接配置和管理的集群。Astra Control Service支持在公共云环境中运行的自管理集群。您的自行管理集群可以使用Astra控件配置程序与NetApp存储服务连接、也可以使用容器存储接口(CSI)驱动程序与Amazon Elastic Block Store (EBS)、Azure托管磁盘和Google持久磁盘连接。

Astra控制服务支持使用以下Kubernetes分发版的自管理集群：

- Red Hat OpenShift 容器平台
- Rancher Kubernetes引擎
- 上游Kubernetes

您的自管理集群需要满足以下要求：

- 集群必须可通过Internet访问。
- 如果您正在使用或计划使用已启用CSI驱动程序的存储、则必须在集群上安装相应的CSI驱动程序。有关使用CSI驱动程序集成存储的详细信息、请参阅存储服务的文档。
- 您可以访问仅包含一个上下文元素的集群kubeconfigfile文件。请遵循 "这些说明" 生成kubeconfig.文件。
- 如果要使用引用私有证书颁发机构(CA)的kubeconfigfile文件添加集群、请将以下行添加到 `cluster` kubeconfig"文件的部分。这样、Astra Control便可添加集群：

  ```
  insecure-skip-tls-verify: true
  ```

- *仅Rancher *：在Rancher环境中管理应用程序集群时、请修改Rancher提供的kubeconfig文件中的应用程序集群默认上下文、以使用控制平面上下文、而不是Rancher API服务器上下文。这样可以减少Rancher API 服务器上的负载并提高性能。
- **Astra Control**配置程序要求：要管理集群、您应正确配置Astra Control配置程序(包括其Astra三项功能组件)。
  - 查看**Astra**三端环境要求：在安装或升级Astra Control配置程序之前、请查看 "支持的前端、后端和主机配置"。
  - 启用**Astra Control**配置程序功能：强烈建议您安装Astra Trident 23.10或更高版本并启用 "Asta Control配置程序高级存储功能"。在未来版本中、如果Asta Control配置程序未启用、则Asta Control将不支持Asta Trent。
  - 配置存储后端：必须至少有一个存储后端 "已在Asta Trdent中配置" 在集群上。
  - 配置存储类：必须至少有一个存储类 "已在Asta Trdent中配置" 在集群上。如果配置了默认存储类，请确保该存储类是具有默认标注的*Only"存储类。
  - 配置卷快照控制器并安装卷快照类： "安装卷快照控制器" 以便可以在Astra Control中创建快照。 "创建" 至少一个 `VolumeSnapshotClass` 使用Asta三端到功能。

步骤

1. 在信息板上，选择 * 管理 Kubernetes 集群 * 。

   按照提示添加集群。

2. 提供程序：选择*其他*选项卡以添加有关自行管理的集群的详细信息。
   a. 其他：通过上传提供有关自管理集群的详细信息 `kubeconfig.yaml` 文件或粘贴的内容 `kubeconfig.yaml` 文件。

      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。

3. 凭据名称：提供要上传到Astra Control的自管理集群凭据的名称。默认情况下，凭据名称会自动填充为集群的名称。
4. 专用路由标识符：此字段仅适用于专用集群。
5. 选择 * 下一步 * 。
6. (可选)存储：(可选)选择默认情况下希望部署到此集群中的Kubernetes应用程序使用的存储类。
   a. 要为集群选择新的默认存储类，请启用*Assign a new default storage class*复选框。
   b. 从列表中选择新的默认存储类。

      每个云提供商存储服务都会显示以下价格、性能和弹性信息：

      - Cloud Volumes Service for Google Cloud：价格、性能和弹性信息
      - Google Persistent Disk：没有价格、性能或弹性信息
      - Azure NetApp Files ：性能和弹性信息
      - Azure受管磁盘：无可用的价格、性能或弹性信息
      - Amazon Elastic Block Store：没有价格、性能或弹性信息
      - 适用于NetApp ONTAP 的Amazon FSX：没有价格、性能或弹性信息
      - NetApp Cloud Volumes ONTAP ：没有价格、性能或弹性信息

      每个存储类均可使用以下服务之一：

      - "适用于 Google Cloud 的 Cloud Volumes Service"
      - "Google 持久磁盘"
        - "Azure NetApp Files"
        - "Azure 受管磁盘"
        - "Amazon Elastic Block Store"
        - "适用于 NetApp ONTAP 的 Amazon FSX"
        - "NetApp Cloud Volumes ONTAP"

          了解更多信息 "Amazon Web Services集群的存储类"。了解更多信息 "AKS 集群的存储类"。了解更多信息 "GKE 集群的存储类"。

   c. 选择 * 下一步 * 。
   d. 审核和批准：审核配置详细信息。
   e. 选择*Add*将集群添加到Astra Control Service。

您可以更改集群的默认存储类。

使用**Astra Control**更改默认存储类

您可以在Astra Control中更改集群的默认存储类。如果集群使用先前安装的存储后端服务、则可能无法使用此方法更改默认存储类(不能选择*设置为默认值*操作)。在这种情况下、您可以 [使用命令行更改默认存储类]。

步骤

1. 在 Astra 控制服务 UI 中，选择 * 集群 * 。
2. 在*集群*页面上、选择要更改的集群。
3. 选择 * 存储 * 选项卡。
4. 选择*存储类*类别。
5. 选择要设置为默认值的存储类的*操作*菜单。
6. 选择*设置为默认值*。

使用命令行更改默认存储类

您可以使用Kubernetes命令更改集群的默认存储类。无论集群的配置如何、此方法都有效。

步骤

1. 登录到Kubernetes集群。
2. 列出集群中的存储类：

```
kubectl get storageclass
```

3. 从默认存储类中删除默认指定。将<SC_NAME> 替换为存储类的名称：

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. 将其他存储类标记为默认值。将<SC_NAME> 替换为存储类的名称：

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. 确认新的默认存储类：

```
kubectl get storageclass
```

将私有自管理集群添加到**Asta Control Service**

设置环境后，您可以创建 Kubernetes 集群，然后将其添加到 Astra Control Service 。

自管理集群是指直接配置和管理的集群。Astra Control Service支持在公共云环境中运行的自管理集群。您可以通过上传将自管理集群添加到Astra Control Service `kubeconfig.yaml` 文件您需要确保集群满足此处所述的要求。

支持的**Kubnetes**分发版

您可以使用Astra Control Service管理以下类型的专用自管理集群：

Kubbernetes分发	支持的版本
Kubnetes (上游)	1.27至1.29
Rancher Kubernetes Engine （RKE）	RKE 1：版本1.24.17、1.25.13、1.26.8、带RANcher Manager 2.7.9 RKE 2：版本1.23.16和1.24.13、带Randcher Manager 2.6.13 RKE 2：版本1.24.17、1.25.14、1.26.9、带RANcher Manager 2.7.9
Red Hat OpenShift 容器平台	4.12至4.14

以下说明假定您已创建私有集群并准备好了远程访问该集群的安全方法。

要将专用集群添加到Astra Control Service、您需要执行以下任务：

1. 安装A作用 连接器
2. [设置永久性存储]
3. 将专用自管理集群添加到Asta Control Service

安装**A**作用 连接器

在添加专用集群之前、您需要在此集群上安装Astra Connector、以便Astra Control可以与其通信。请参见 "为使用非Kubnetes本机工作流管理的专用集群安装以前版本的Astra Connector" 有关说明，请参见。

设置永久性存储

为集群配置永久性存储。有关配置永久性存储的详细信息、请参见入门文档：

- "使用 Azure NetApp Files 设置 Microsoft Azure"
- "使用 Azure 受管磁盘设置 Microsoft Azure"
- "设置Amazon Web Services"
- "设置 Google Cloud"

将专用自管理集群添加到**Asta Control Service**

现在、您可以将专用集群添加到Astra Control Service。

开始之前

自管理集群是指直接配置和管理的集群。Astra Control Service支持在公共云环境中运行的自管理集群。您的自行管理集群可以使用Astra控件配置程序与NetApp存储服务连接、也可以使用容器存储接口(CSI)驱动程序与Amazon Elastic Block Store (EBS)、Azure托管磁盘和Google持久磁盘连接。

Astra控制服务支持使用以下Kubernetes分发版的自管理集群：

- Red Hat OpenShift 容器平台
- Rancher Kubernetes引擎
- 上游Kubernetes

您的自管理集群需要满足以下要求：

- 集群必须可通过Internet访问。
- 如果您正在使用或计划使用已启用CSI驱动程序的存储、则必须在集群上安装相应的CSI驱动程序。有关使用CSI驱动程序集成存储的详细信息、请参阅存储服务的文档。
- 您可以访问仅包含一个上下文元素的集群kubeconfigfile文件。请遵循 "这些说明" 生成kubeconfig.文件。
- 如果要使用引用私有证书颁发机构(CA)的kubeconfigfile文件添加集群、请将以下行添加到 `cluster` kubeconfig"文件的部分。这样、Astra Control便可添加集群：

  ```
  insecure-skip-tls-verify: true
  ```

- *仅Rancher *：在Rancher环境中管理应用程序集群时、请修改Rancher提供的kubeconfig文件中的应用程序集群默认上下文、以使用控制平面上下文、而不是Rancher API服务器上下文。这样可以减少Rancher API 服务器上的负载并提高性能。
- **Astra Control**配置程序要求：要管理集群、您应正确配置Astra Control配置程序(包括其Astra三项功能组件)。
  - 查看**Astra**三端环境要求：在安装或升级Astra Control配置程序之前、请查看 "支持的前端、后端和主机配置"。
  - 启用**Astra Control**配置程序功能：强烈建议您安装Astra Trident 23.10或更高版本并启用 "Asta Control配置程序高级存储功能"。在未来版本中、如果Asta Control配置程序未启用、则Asta Control将不支持Asta Trent。
  - 配置存储后端：必须至少有一个存储后端 "已在Asta Trdent中配置" 在集群上。
  - 配置存储类：必须至少有一个存储类 "已在Asta Trdent中配置" 在集群上。如果配置了默认存储类，请确保该存储类是具有默认标注的*Only"存储类。
  - 配置卷快照控制器并安装卷快照类： "安装卷快照控制器" 以便可以在Astra Control中创建快照。 "创建" 至少一个 `VolumeSnapshotClass` 使用Asta三端到功能。

步骤

1. 在信息板上，选择 * 管理 Kubernetes 集群 * 。

   按照提示添加集群。

2. 提供程序：选择*其他*选项卡以添加有关自行管理的集群的详细信息。
3. 其他：通过上传提供有关自管理集群的详细信息 `kubeconfig.yaml` 文件或粘贴的内容 `kubeconfig.yaml` 文件。

> 创建自己的 `kubeconfig` file中、您只能定义*一*上下文元素。请参见 "这些说明" 有关创建的信息 `kubeconfig` 文件。

4. 凭据名称：提供要上传到Astra Control的自管理集群凭据的名称。默认情况下，凭据名称会自动填充为集群的名称。
5. 专用路由标识符：输入专用路由标识符，您可以从Astra Connector获取该标识符。如果您通过查询Astra Connector `kubectl get astraconnector -n astra-connector` 命令中、专用路由标识符称为 `ASTRACONNECTORID`。

> 专用路由标识符是与Asta Connector关联的名称、Asta可通过Asta管理专用Kubnetes集群。在这种情况下、专用集群是指不向Internet公开其API服务器的Kubbernetes集群。

6. 选择 * 下一步 * 。
7. (可选)存储：(可选)选择默认情况下希望部署到此集群中的Kubernetes应用程序使用的存储类。
   a. 要为集群选择新的默认存储类，请启用*Assign a new default storage class*复选框。
   b. 从列表中选择新的默认存储类。

> 每个云提供商存储服务都会显示以下价格、性能和弹性信息：
>
> - Cloud Volumes Service for Google Cloud：价格、性能和弹性信息
> - Google Persistent Disk：没有价格、性能或弹性信息
> - Azure NetApp Files ：性能和弹性信息
> - Azure受管磁盘：无可用的价格、性能或弹性信息
> - Amazon Elastic Block Store：没有价格、性能或弹性信息
> - 适用于NetApp ONTAP 的Amazon FSX：没有价格、性能或弹性信息
> - NetApp Cloud Volumes ONTAP ：没有价格、性能或弹性信息

每个存储类均可使用以下服务之一：

- "适用于 Google Cloud 的 Cloud Volumes Service"
- "Google 持久磁盘"
- "Azure NetApp Files"
- "Azure 受管磁盘"
- "Amazon Elastic Block Store"
- "适用于 NetApp ONTAP 的 Amazon FSX"
- "NetApp Cloud Volumes ONTAP"

了解更多信息 "Amazon Web Services集群的存储类"。了解更多信息 "AKS 集群的存储类"。了解更

多信息 "GKE 集群的存储类"。

c. 选择 * 下一步 * 。

d. 审核和批准：审核配置详细信息。

e. 选择*Add*将集群添加到Astra Control Service。

更改默认存储类

您可以更改集群的默认存储类。

使用**Astra Control**更改默认存储类

您可以在Astra Control中更改集群的默认存储类。如果集群使用先前安装的存储后端服务、则可能无法使用此方法更改默认存储类(不能选择*设置为默认值*操作)。在这种情况下、您可以 [使用命令行更改默认存储类]。

步骤

1. 在 Astra 控制服务 UI 中，选择 * 集群 * 。
2. 在*集群*页面上、选择要更改的集群。
3. 选择 * 存储 * 选项卡。
4. 选择*存储类*类别。
5. 选择要设置为默认值的存储类的*操作*菜单。
6. 选择*设置为默认值*。

使用命令行更改默认存储类

您可以使用Kubernetes命令更改集群的默认存储类。无论集群的配置如何、此方法都有效。

步骤

1. 登录到Kubernetes集群。
2. 列出集群中的存储类：

```
kubectl get storageclass
```

3. 从默认存储类中删除默认指定。将<SC_NAME> 替换为存储类的名称：

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. 将其他存储类标记为默认值。将<SC_NAME> 替换为存储类的名称：

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. 确认新的默认存储类：

```
kubectl get storageclass
```

检查**Asta**三端安装版本

要添加使用Asta Control置备程序或Asta三端对存储服务使用的自管理集群、请确保已安装的Asta三端对等版本为23.10或最新版本。

步骤

1. 确定您正在运行的Astra三项目标版本：

```
kubectl get tridentversions -n trident
```

如果安装了Astra Trident、则会显示类似于以下内容的输出：

```
NAME       VERSION
trident    24.02.0
```

如果未安装Astra Trident、您将看到类似于以下内容的输出：

```
error: the server doesn't have a resource type "tridentversions"
```

2. 执行以下操作之一：
   - 如果您运行的是Asta三端凹凸版23.01或更早版本、请使用这些版本 "说明" 在升级到Asta Control配置程序之前、升级到Asta三端到最新版本。您可以 "执行直接升级" 如果您的Astra三端存储在版本24.02的四个版本的窗口中、则将Astra Control配置程序更新为24.02。例如、您可以直接从Asta三端23.04升级到Asta Control配置程序24.02。
   - 如果您运行的是Astra Trdent 23.10或更高版本、请验证Astra Control配置程序是否已启用 "enabled" 。Asta Control配置程序不能用于23.10之前的Asta Control Center版本。 "升级Astra Control配置程序" 以便它与您要升级的Astra Control Center版本相同、以访问最新功能。
3. 确保Pod正在运行：

```
kubectl get pods -n trident
```

4. 检查存储类是否正在使用受支持的Astra Trident驱动程序。配置程序名称应为

csi.trident.netapp.io。请参见以下示例：

```
kubectl get sc
```

响应示例：

```
NAME                    PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
ontap-gold (default)    csi.trident.netapp.io          Delete
Immediate           true                   5d23h
```

创建**kubeconfig**文件

您可以使用kubeconfig"文件将集群添加到Astra Control Service。根据要添加的集群类型、您可能需要使用特定步骤为集群手动创建kubeconfigfile文件。

- 为Amazon EKS集群创建kubeconfig.文件
- 为AWS (ROSA)集群上的Red Hat OpenShift Service创建一个kubeconfigfile文件
- [为其他类型的集群创建kubeconfig.文件]

为**Amazon EKS**集群创建**kubeconfig.**文件

按照以下说明为Amazon EKS集群创建kubeconfigfile文件和永久令牌密钥。EKS中托管的集群需要永久令牌密钥。

步骤

1. 按照亚马逊文档中的说明生成kubeconfig:

   "为Amazon EKS集群创建或更新kubeconconfig文件"

2. 按如下所示创建服务帐户：

   a. 创建名为的服务帐户文件 astracontrol-service-account.yaml。

      根据需要调整服务帐户名称。命名空间 kube-system 这些步骤需要。如果您在此处更改了服务帐户名称、则应在以下步骤中应用相同的更改。

```
<strong>astracontrol-service-account.yaml</strong>
```

+

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astra-admin-account
  namespace: kube-system
```

3. 应用服务帐户：

```
kubectl apply -f astracontrol-service-account.yaml
```

4. 创建 `ClusterRoleBinding` 文件已调用 `astracontrol-clusterrolebinding.yaml`。

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astra-admin-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astra-admin-account
  namespace: kube-system
```

5. 应用集群角色绑定：

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

6. 创建名为的服务帐户令牌机密文件 `astracontrol-secret.yaml`。

```
<strong>astracontrol-secret.yaml</strong>
```

```
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: astra-admin-account
  name: astra-admin-account
  namespace: kube-system
type: kubernetes.io/service-account-token
```

7. 应用令牌密钥：

```
kubectl apply -f astracontrol-secret.yaml
```

8. 检索令牌密钥：

```
kubectl get secret astra-admin-account -n kube-system -o
jsonpath='{.data.token}' | base64 -d
```

9. 更换 `user` 部分的AWS EKS kubeconfigconfig文件以及令牌、如以下示例所示：

```
user:
    token: k8s-aws-
v1.aHR0cHM6Ly9zdHMudXMtd2VzdC0yLmFtYXpvbmF3cy5jb20vP0FjdGlvbj1HZXRDYWxsZ
XJJZGVudGl0eSZWZXJzaW9uPTIwMTEtMDYtMTUmWC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQ
y1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBM1JEWDdKU0haWU9LSEQ2SyUyRjIwMjMwN
DAzJTJGdXMtd2VzdC0yJTJGc3RzJTJGYXdzNF9yZXF1ZXN0JlgtQW16LURhdGU9MjAyMzA0M
DNUMjA0MzQwWiZYLUFtei1FeHBpcmVzPTYwJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCUzQ
ngtazhzLWF3cy1pZCZYLUFtei1TaWduYXR1cmU9YjU4ZWM0NzdiM2NkZGYxNGRhNzU4MGI2Z
WQ2zY2NzI2YWIwM2UyNThjMjRhNTJjNmVhNjc4MTRlNjJkOTg2Mg
```

为AWS (ROSA)集群上的Red Hat OpenShift Service创建一个kubeconfigfile文件

按照以下说明为Red Hat OpenShift Service on AWS (ROSA)集群创建kubeconfigTM文件。

步骤

1. 登录到ROSA集群。
2. 创建服务帐户：

```
oc create sa astracontrol-service-account
```

3. 添加集群角色：

```
oc adm policy add-cluster-role-to-user cluster-admin -z astracontrol-service-account
```

4. 使用以下示例、创建一个服务帐户机密配置文件：

```
<strong>secret-astra-sa.yaml</strong>
```

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-account"
type: kubernetes.io/service-account-token
```

5. 创建密钥：

```
oc create -f secret-astra-sa.yaml
```

6. 编辑您创建的服务帐户、并将Astra Control服务帐户机密名称添加到中 `secrets` 部分。

```
oc edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-dvfcd
kind: ServiceAccount
metadata:
  creationTimestamp: "2023-08-04T04:18:30Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "169770"
  uid: 965fa151-923f-4fbd-9289-30cad15998ac
secrets:
- name: astracontrol-service-account-dockercfg-dvfcd
- name: secret-astracontrol-service-account ####ADD THIS ONLY####
```

7. 列出服务帐户密码、替换 `<CONTEXT>` 使用适用于您的安装的正确环境：

```
kubectl get serviceaccount astracontrol-service-account --context
<CONTEXT> --namespace default -o json
```

输出的结尾应类似于以下内容：

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-dvfcd"},
{ "name": "secret-astracontrol-service-account"}
]
```

中每个元素的索引 `secrets` 阵列以0开头。在上面的示例中、是的索引 `astracontrol-service-account-dockercfg-dvfcd` 将为0、并为创建索引 `secret-astracontrol-service-account` 将为1。在输出中、记下服务帐户密钥的索引编号。在下一步中、您将需要此索引编号。

8. 按如下所示生成 kubeconfig：
   a. 创建 `create-kubeconfig.sh` 文件替换 `TOKEN_INDEX` 在以下脚本的开头、使用正确的值。

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)
```

```
# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token
-user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

b. 获取用于将其应用于 Kubernetes 集群的命令。

```
source create-kubeconfig.sh
```

9. (可选)将kubeconfig重命名为集群的有意义名称。

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

为其他类型的集群创建**kubeconfig.**文件

按照以下说明为然彻集群、上游Kubernetes集群和Red Hat OpenShift集群创建有限或扩展的角色kubeconconfig文件。

对于使用kubeconfig"管理的集群、您可以选择为Astra Control Service创建有限权限或扩展权限管理员角色。

如果您适用场景的环境发生以下任一情况、则此操作步骤可帮助您创建一个单独的kubeconfig:

- 您希望限制Astra Control对其管理的集群的权限
- 您使用多个环境、并且不能使用在安装期间配置的默认Asta Control kubeconfig,否则在您的环境中使用单一环境的有限角色将不起作用

开始之前

在完成操作步骤 步骤之前、请确保您对要管理的集群具有以下信息：

- 答 "支持的版本" 已安装kubeck.
- 对要使用Astra Control Service添加和管理的集群的kubect访问权限

> 对于此操作步骤、您不需要对运行Astra控制服务的集群进行kubect访问。

- 要使用活动环境的集群管理员权限管理的集群的活动kubeconfig

步骤

1. 创建服务帐户：
   a. 创建名为的服务帐户文件 `astracontrol-service-account.yaml`。

```
<strong>astracontrol-service-account.yaml</strong>
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astracontrol-service-account
  namespace: default
```

   b. 应用服务帐户：

```
kubectl apply -f astracontrol-service-account.yaml
```

2. 创建以下具有足够权限的集群角色之一、以使集群由Astra Control管理：

集群角色受限

此角色包含由Asta Control管理集群所需的最低权限：

a. 创建 `ClusterRole` 文件、例如、 `astra-admin-account.yaml`。

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:

# Get, List, Create, and Update all resources
# Necessary to backup and restore all resources in an app
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - create
  - patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
  - deployments
```

```
    - horizontalpodautoscalers
    - ingresses
    - jobs
    - namespaces
    - networkpolicies
    - persistentvolumeclaims
    - poddisruptionbudgets
    - pods
    - podtemplates
    - replicasets
    - replicationcontrollers
    - replicationcontrollers/scale
    - rolebindings
    - roles
    - secrets
    - serviceaccounts
    - services
    - statefulsets
    - tridentmirrorrelationships
    - tridentsnapshotinfos
    - volumesnapshots
    - volumesnapshotcontents
    verbs:
    - delete

  # Watch resources
  # Necessary to monitor progress
  - apiGroups:
    - ""
    resources:
    - pods
    - replicationcontrollers
    - replicationcontrollers/scale
    verbs:
    - watch

  # Update resources
  - apiGroups:
    - ""
    - build.openshift.io
    - image.openshift.io
    resources:
    - builds/details
    - replicationcontrollers
    - replicationcontrollers/scale
    - imagestreams/layers
```

```
  - imagestreamtags
  - imagetags
  verbs:
  - update
```

b. (仅适用于OpenShift集群)在末尾附加以下内容 `astra-admin-account.yaml` 文件：

```
# OpenShift security
- apiGroups:
  - security.openshift.io
  resources:
  - securitycontextconstraints
  verbs:
  - use
  - update
```

c. 应用集群角色：

```
kubectl apply -f astra-admin-account.yaml
```

已扩展集群角色

此角色包含要由Asta Control管理的集群的扩展权限。如果您使用多个环境，并且无法使用在安装期间配置的默认Asta Control kubeconfig,则可以使用此角色，否则在您的环境中，只使用一个环境的有限角色将不起作用：

> 以下内容 `ClusterRole` 步骤是一个常规Kubbernetes示例。有关特定于您的环境的说明、请参见Kubennetes分发版的文档。

a. 创建 `ClusterRole` 文件、例如、 `astra-admin-account.yaml`。

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

b. 应用集群角色：

```
kubectl apply -f astra-admin-account.yaml
```

3. 为集群角色创建与服务帐户的集群角色绑定：

a. 创建 `ClusterRoleBinding` 文件已调用 `astracontrol-clusterrolebinding.yaml`。

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: astra-admin-account
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

b. 应用集群角色绑定：

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. 创建并应用令牌密钥：
   a. 创建名为的令牌机密文件 `secret-astracontrol-service-account.yaml`。

```
<strong>secret-astracontrol-service-account.yaml</strong>
```

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  namespace: default
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-
account"
type: kubernetes.io/service-account-token
```

   b. 应用令牌密钥：

```
kubectl apply -f secret-astracontrol-service-account.yaml
```

5. 通过将令牌密钥名称添加到、将其添加到服务帐户 `secrets` 数组(以下示例中的最后一行)：

```
kubectl edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-48xhx
kind: ServiceAccount
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{},
"name":"astracontrol-service-account","namespace":"default"}}
  creationTimestamp: "2023-06-14T15:25:45Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "2767069"
  uid: 2ce068c4-810e-4a96-ada3-49cbf9ec3f89
secrets:
- name: astracontrol-service-account-dockercfg-48xhx
<strong>- name: secret-astracontrol-service-account</strong>
```

6. 列出服务帐户密码、替换 `<context>` 使用适用于您的安装的正确环境：

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

输出的结尾应类似于以下内容：

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]
```

中每个元素的索引 `secrets` 阵列以0开头。在上面的示例中、是的索引 `astracontrol-service-account-dockercfg-48xhx` 将为0、并为创建索引 `secret-astracontrol-service-account` 将为1。在输出中、记下服务帐户密钥的索引编号。在下一步中、您将需要此索引编号。

7. 按如下所示生成 kubeconfig：
   a. 创建 `create-kubeconfig.sh` 文件
   b. 替换 `TOKEN_INDEX` 在以下脚本的开头、使用正确的值。

```
<strong>create-kubeconfig.sh</strong>
```

```
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  *-o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
```

```
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token-
user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

c. 获取用于将其应用于 Kubernetes 集群的命令。

```
source create-kubeconfig.sh
```

8. (可选)将kubeconfig重命名为集群的有意义名称。

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

下一步是什么?

现在、您已登录并向Astra Control添加了集群、可以开始使用Astra Control的应用程序数据管理功能了。

- "开始管理应用程序"
- "保护应用程序"
- "克隆应用程序"
- "设置计费"
- "邀请和管理用户"
- "管理云提供商凭据"
- "管理通知"
- "部署Asta Control的自管理实例"

Astra Control Service 视频

查看NetApp TV、了解有关Astra Control Service的最新视频内容。NetApp TV中的视频演

示了Astra Control Service的某些功能、或者向您展示了如何完成某些常见任务。

"Astra Control Service 视频"

概念

架构和组件

Asta Control是一款Kubennet应用程序数据生命周期管理解决方案、可简化有状态应用程序的操作、并帮助您在混合和多云环境之间存储、保护和移动Kubennet工作负载。

功能

Astra Control 为 Kubernetes 应用程序数据生命周期管理提供了关键功能：

存储：

- 为容器化工作负载动态配置存储
- 对从容器到永久性卷的数据进行传输中加密
- 跨区域、跨区域复制

保护：

- 自动发现整个应用程序及其数据并提供应用程序感知型保护
- 根据组织需求从任何Snapshot版本即时恢复应用程序
- 跨区域、区域和云提供商实现快速故障转移

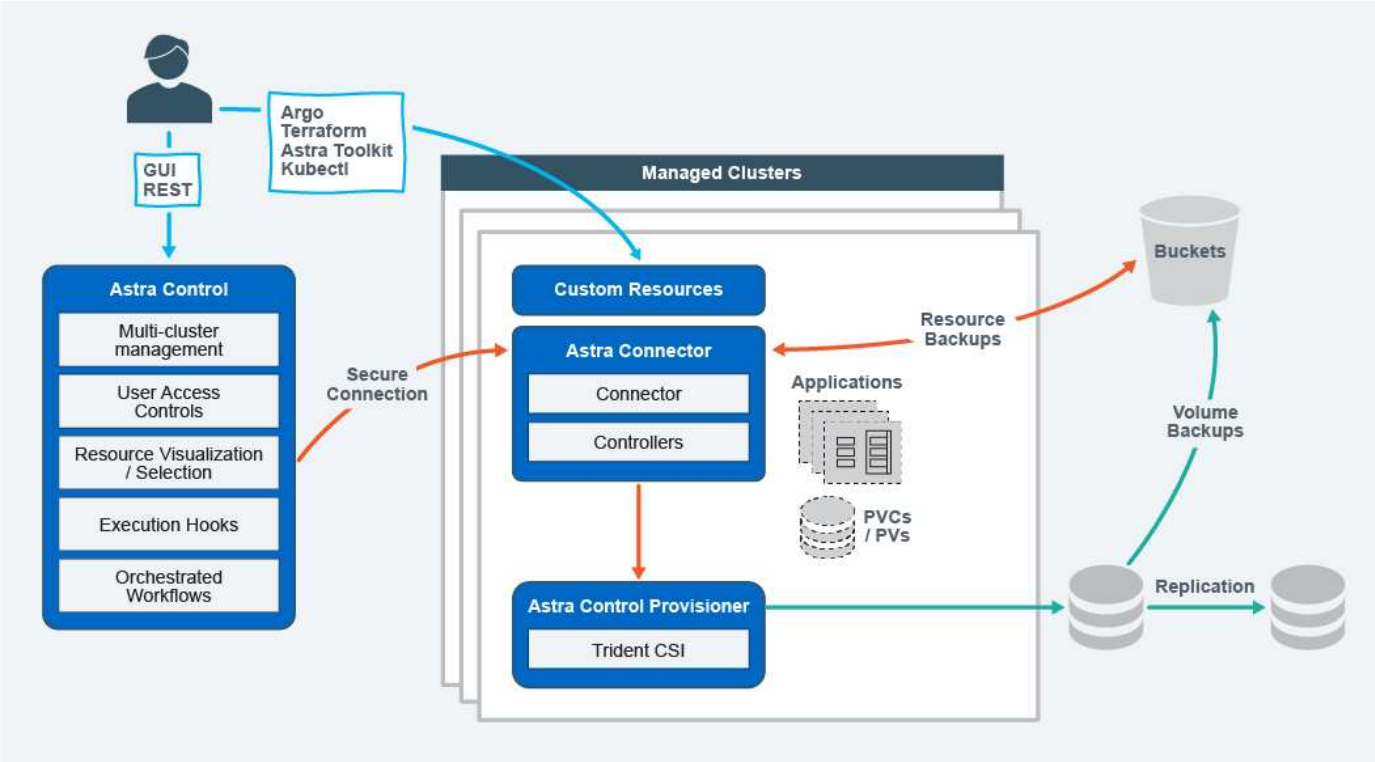
移动：

- 在Kubbernetes集群和云之间实现全面的应用程序和数据移动
- 即时克隆整个应用程序和数据
- 通过一致的Web UI和API一键迁移应用程序

架构

Asta Control的架构支持IT提供高级数据管理功能、以增强Kubbernetes应用程序的功能和可用性、简化容器化工作负载在公有云和内部环境之间的管理、保护和移动。 并通过REST API和SDK提供自动化功能、支持编程访问、以便与现有工作流无缝集成。

Astra Control是Kub联网 原生版本、支持利用自定义资源的数据保护工作流、同时保持与现有API和SDK的向后兼容性。Kubornetes原生数据保护具有显著优势；通过与Kubornetes API和资源无缝集成、数据保护可以通过组织的现有CI/CD和/或GitOps工具成为应用程序生命周期的固有组成部分。

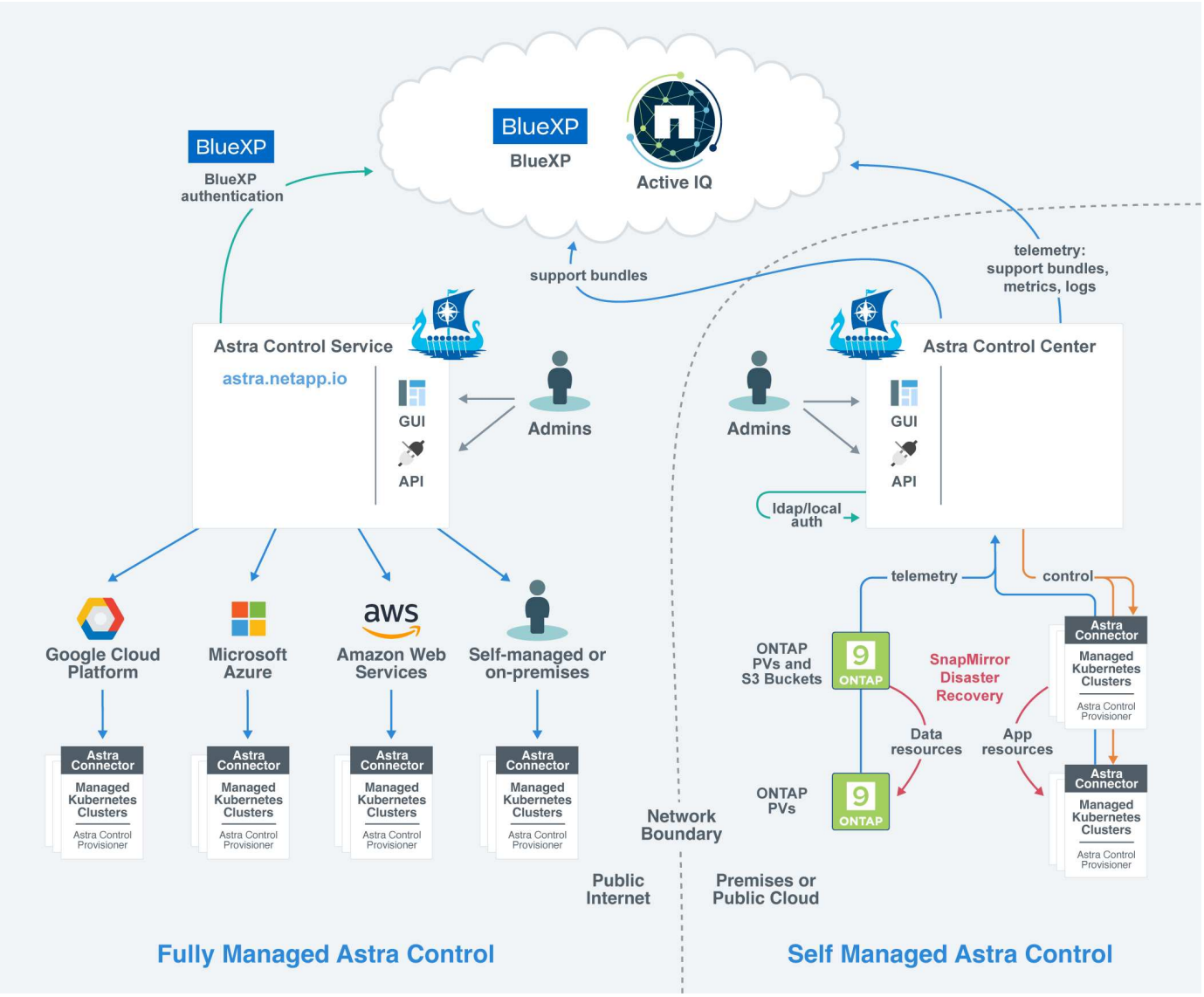


Astra Control基于四个互补组件构建：

- **Astra Control**：Astra Control是适用于所有托管集群的集中式管理服务，可提供协调的工作负载，以实现云和内部环境中的应用程序保护和移动性以及以下功能：
  - 多个集群和云的组合视图
  - 保护协调一致的工作流
  - 精细的资源可视化和选择
- **Astra Connecter**：Astra Connector与Astra Control相结合、可提供与每个受管集群的安全连接、无论连接状态如何、均可在本地执行计划的操作、并具有以下功能：
  - 本地执行计划的操作、而不管连接状态如何
  - 在集群之间分布和优化A作用 的系统资源使用的本地操作
  - 本地安装、允许对集群进行最低权限访问以提高安全性
- **Astra Control**置备程序：Astra Control置备程序提供核心CSI配置功能和高级存储管理功能，以增加安全性和灾难恢复配置，以及以下功能：
  - 为容器化工作负载动态配置存储
  - 高级存储管理：
    - 对从容器到PV的数据进行传输中加密
    - SnapMirror Cloud功能支持跨区域、跨区域复制
- **Astra**自定义资源：每个集群上使用的自定义资源提供了一种Kubernetes本机方法来在本地运行操作、简化了与其他Kubernetes友好工具和自动化的集成、并提供了以下功能：
  - 直接的生态系统工具集成和自动化工作流
  - 用于启用自定义工作流的较低级别的基本功能

部署模式

Asta Control有两种部署模式。



- * Astra Control Service*：NetApp管理的服务、可为多个云提供商环境中的Kubernetes集群以及自管理Kubernetes集群提供应用程序感知型数据管理。

  "Astra Control Service 文档"

- * Astra Control Center*：自管理软件，可为内部环境中运行的 Kubernetes 集群提供应用程序感知型数据管理。Astra控制中心还可以安装在多个云提供商环境中、并具有一个NetApp Cloud Volumes ONTAP 存储后端。

  "Astra 控制中心文档"

	Astra 控制服务	**Astra** 控制中心
如何提供?	作为 NetApp 提供的一项完全托管的云服务	作为可下载、安装和管理的软件

	Astra 控制服务	**Astra** 控制中心
它托管在何处?	基于 NetApp 选择的公有云	在您自己的Kubernetes集群上
如何更新?	由 NetApp 管理	您可以管理任何更新
支持哪些**Kubednetes**分发版?	• 云提供商 ◦ Amazon Web Services ▪ Amazon Elelic Kubelnetes Service (EKS) ◦ Google Cloud ▪ Google Kubernetes Engine （GKEE） ◦ Microsoft Azure ▪ Azure Kubernetes Service （AKS） • 自管理集群 ◦ Kubnetes (上游) ◦ Rancher Kubernetes Engine （RKE） ◦ Red Hat OpenShift 容器平台 • 内部集群 ◦ Red Hat OpenShift容器平台内部部署	• 基于Azure堆栈HCI的Azure Kubnetes Service • Google Anthos • Kubnetes (上游) • Rancher Kubernetes Engine （RKE） • Red Hat OpenShift 容器平台

	Astra 控制服务	Astra 控制中心

支持哪些存储后端?	• 云提供商
	  ◦ Amazon Web Services
	  ▪ Amazon EBS
	  ▪ 适用于 NetApp ONTAP 的 Amazon FSX
	  ▪ "Cloud Volumes ONTAP"
	  ◦ Google Cloud
	  ▪ Google 持久磁盘
	  ▪ NetApp Cloud Volumes Service
	  ▪ "Cloud Volumes ONTAP"
	  ◦ Microsoft Azure
	  ▪ Azure受管磁盘
	  ▪ Azure NetApp Files
	  ▪ "Cloud Volumes ONTAP"
	• 自管理集群
	  ◦ Amazon EBS
	  ◦ Azure受管磁盘
	  ◦ Google 持久磁盘
	  ◦ "Cloud Volumes ONTAP"
	  ◦ NetApp MetroCluster
	  ◦ "Longhorn"
	• 内部集群
	  ◦ NetApp MetroCluster
	  ◦ NetApp ONTAP AFF 和 FAS 系统
	  ◦ NetApp ONTAP Select
	  ◦ "Cloud Volumes ONTAP"
	  ◦ "Longhorn"
		• NetApp ONTAP AFF 和 FAS 系统
		• NetApp ONTAP Select
		• "Cloud Volumes ONTAP"
		• "Longhorn"

有关详细信息 ...

- "Astra Control Service 文档"
- "Astra 控制中心文档"
- "Astra Trident 文档"
- "Astra Control API"
- "Cloud Insights 文档"

- "ONTAP 文档"

数据保护

了解Astra控制服务中可用的数据保护类型、以及如何以最佳方式使用它们来保护您的应用程序。

快照，备份和保护策略

快照和备份均可保护以下类型的数据：

- 应用程序本身
- 与应用程序关联的任何永久性数据卷
- 属于应用程序的任何资源项目

snapshot 是应用程序的时间点副本，它与应用程序存储在同一个已配置卷上。通常速度较快。您可以使用本地快照将应用程序还原到较早的时间点。快照对于快速克隆很有用；快照包括应用程序的所有 Kubernetes 对象，包括配置文件。快照对于克隆或还原同一集群中的应用程序非常有用。

_backup_基于快照。它存储在外部对象存储中、因此、与本地快照相比、创建速度可能会较慢。您可以将应用程序备份还原到同一集群，也可以通过将应用程序备份还原到其他集群来迁移应用程序。您还可以选择较长的备份保留期限。由于备份存储在外部对象存储中，因此在发生服务器故障或数据丢失时，备份通常比快照提供更好的保护。

保护策略 _ 是一种通过根据您为应用程序定义的计划自动创建快照和 / 或备份来保护应用程序的方法。此外、您还可以通过保护策略选择要在计划中保留多少个快照和备份、并设置不同的计划粒度级别。使用保护策略自动执行备份和快照是确保每个应用程序根据组织的需求和服务级别协议(Service Level Agreement、SLA)要求进行保护的最佳方式。

> *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关联的永久性存储，则需要备份才能恢复。快照无法让您恢复。

> 如果您执行快照或备份、但操作失败并显示错误"The resource was n't created because of an internal server问题描述 "、请检查以确保您正在使用的存储后端安装了正确的驱动程序。某些存储后端需要容器存储接口(CSI)驱动程序、而其他存储后端则需要外部快照控制器。

不可配置的备份

不可变备份是指在指定时间段内无法更改或删除的备份。在创建不可更改的备份时、Astra Control会检查以确保您使用的存储分段是一次写入多次读取(Write on时 读取多次、WORM)存储分段、如果是、则会确保备份在Astra Control中不可更改。
Astra Control Service支持使用以下平台和存储分段类型创建不可配置的备份：

- Amazon Web Services使用配置了S3对象锁定的Amazon S3存储分段
- Microsoft Azure使用已配置保留策略的Azure存储分段
- 使用配置了保留策略的Google Cloud Storage存储分段的Google Kub并 配置了保留策略

- 使用配置了S3对象锁定的S3存储分段的NetApp StorageGRID

使用不可配置备份时、请注意以下事项：

- 如果备份到不受支持的平台中的WORM存储分段或备份到不受支持的存储分段类型、则可能会出现无法预测的结果、例如、即使已过保留时间、备份删除也会失败。
- Astra Control不支持数据生命周期管理策略、也不支持手动删除用于不可变备份的存储分段上的对象。确保存储后端未配置为管理Astra Control快照或备份数据的生命周期。

克隆

_cloner_是应用程序、其配置及其永久性数据卷的精确副本。您可以在同一个 Kubernetes 集群或另一个集群上手动创建克隆。如果需要将应用程序和存储从一个 Kubernetes 集群移动到另一个 Kubernetes 集群，则克隆应用程序非常有用。

AWS集群的存储类和性能

Astra控制服务可以使用Amazon Elastic Block Store (EBS)、Amazon FSx for NetApp ONTAP 或NetApp Cloud Volumes ONTAP 作为Amazon Elastic Kubelnetes Service (EKS)集群的存储后端。

Amazon Elastic Block Store (EBS)

您的集群可以使用容器存储接口(Container Storage Interface、CSI)驱动程序与EBS建立接口。使用EBS作为EKS集群的存储后端时、您可以配置一些存储类参数。有关参数的含义以及如何配置参数的详细信息、请参见 "Kubernetes文档"。

您可以在EBS中使用多种不同类型的卷：

- 固态驱动器(SSD)
- 硬盘驱动器(HDD)
- 上一代产品

有关每种卷类型及其性能的详细信息、请参见 "Amazon EBS文档"。有关定价信息、请参见 "Amazon EBS定价"。

适用于 NetApp ONTAP 的 Amazon FSX

使用适用于NetApp ONTAP 的FSX作为AWS集群的存储后端时、I/O性能取决于文件系统的配置以及工作负载的特征。有关适用于NetApp ONTAP 性能的FSX的具体信息、请参见 "适用于NetApp ONTAP 性能的Amazon FSX"。有关定价信息、请参见 "适用于NetApp ONTAP 的Amazon FSX定价"。

NetApp Cloud Volumes ONTAP

有关配置NetApp Cloud Volumes ONTAP 的具体信息、包括性能建议、请访问 "NetApp Cloud Volumes ONTAP 文档"。

AKS 集群的存储类和 PV 大小

Astra控制服务支持使用Azure NetApp Files 、Azure托管磁盘或NetApp Cloud Volumes ONTAP 作为Azure Kubornetes Service (AKS)集群的存储后端。

Azure NetApp Files

Astra 控制服务支持将 Azure NetApp Files 作为 Azure Kubernetes Service （AKS）集群的存储后端。您应了解选择存储类和永久性卷大小如何帮助您实现性能目标。

服务级别和存储类

Azure NetApp Files 支持三种服务级别：超存储，高级存储和标准存储。其中每个服务级别都是为满足不同的性能需求而设计的：

超存储

每 1 TiB 最多可提供 128 MiB/ 秒吞吐量。

高级存储

每 1 TiB 可提供高达 64 MiB/ 秒的吞吐量。

标准存储

每 1 TiB 可提供多达 16 MiB/ 秒的吞吐量。

这些服务级别是容量池的一个属性。您需要为要在 Kubernetes 集群中使用的每个服务级别设置一个容量池。 "了解如何设置容量池"。

Astra 控制服务会将这些服务级别用作永久性卷的存储类。将 Kubernetes 集群添加到 Astra Control Service 时，系统会提示您选择 "ule" ， "Premium" 或 "Standard" 作为默认存储类。存储类的名称包括 *netapp-anf-perf-ulum* ， *netapp-anf-perf-prem* 和 *netapp-anf-perf-standard* 。

"有关这些服务级别的详细信息，请参见 Azure NetApp Files 文档"。

永久性卷大小和性能

如上所述，每个服务级别的吞吐量均为已配置容量的 1 TiB 。这意味着，较大的卷可以提供更好的性能。因此，在配置卷时，应同时考虑容量和性能需求。

最小卷大小

Astra 控制服务使用最小卷大小 100 GiB 来配置永久性卷，即使 PVC 要求的卷大小更小也是如此。例如，如果 Helm 图表中的 PVC 要求提供 6 GiB ，则 Astra 控制服务会自动配置 100 GiB 卷。

应用程序备份

如果备份Azure NetApp Files 存储上的应用程序、Astra控制服务会自动临时扩展容量池。备份完成后、Asta Control Service会将容量池缩减到其先前的大小。根据您的Azure订阅、在这种情况下、您可能会产生存储费用。您可以在*Activity*页面事件日志中查看容量池调整大小事件的历史记录。

如果在调整大小操作期间、容量池超过Azure订阅允许的最大大小、则备份操作将失败、并会从Azure API触发

警告。

Azure 受管磁盘

Astra控制服务可以使用容器存储接口(Container Storage Interface、CSI)驱动程序作为存储后端与Azure受管磁盘进行连接。此服务可提供由 Azure 管理的块级存储。

"了解有关 Azure 受管磁盘的更多信息"。

NetApp Cloud Volumes ONTAP

有关配置NetApp Cloud Volumes ONTAP 的具体信息、包括性能建议、请访问 "NetApp Cloud Volumes ONTAP 文档"。

GKE- 集群的服务类型，存储类和 PV 大小

Astra控制服务支持将NetApp Cloud Volumes Service for Google Cloud、Google持久磁盘或NetApp Cloud Volumes ONTAP作为永久性卷的存储后端选项。

适用于 Google Cloud 的 Cloud Volumes Service

Astra 控制服务可以使用适用于 Google Cloud 的 Cloud Volumes Service 作为永久性卷的存储后端。您应了解选择服务类型，存储类和永久性卷大小如何帮助您实现性能目标。

概述

Cloud Volumes Service for Google Cloud 提供两种服务类型： *CVS* 和 *CVS-Performance* 。特定 Google Cloud 地区支持这些服务类型。 "转至NetApp BlueXP全球区域地图" 确定集群所在的 Google Cloud 区域支持的服务类型。

如果您的 Kubernetes 集群必须位于特定区域，则您将使用该区域支持的服务类型。

但是，如果您可以灵活地在 Google Cloud 区域之间进行选择，则我们会根据您的性能要求建议您执行以下操作：

- 对于具有中到高性能存储需求的 K8s 应用程序，请选择支持 CVS-Performance 并使用高级或至尊存储类的 Google Cloud 区域。此类工作负载包括 AI/ML 管道， CI/CD 管道，介质处理以及包括关系， NoSQL ，时间序列等在内的数据库
- 对于具有中低端存储性能需求的 K8s 应用程序（ Web 应用程序，通用文件存储等），请选择支持 CVS 或 CVS-Performance 且具有标准存储类的 Google Cloud 区域。

> 如果将CVS服务类型与Asta Control配置程序结合使用、则需要先配置存储池、然后才能配置卷。如果在未配置存储池的情况下配置卷、则卷配置将失败。请参见 "Cloud Volumes Service文档" 有关创建卷的详细信息、请参见。

下表快速比较了此页面上所述的信息。

服务类型	用例	支持的区域	存储类	最小卷大小
CVS 性能	具有中到高存储性能需求的应用程序	"查看支持的 Google Cloud 区域"	• netapp-cvs-perf-standard • netapp-cvs-perf-Premium • netapp-vcs-perf-至 高性能	100 GiB
CVS	具有中低端存储性能需求的应用程序	"查看支持的 Google Cloud 区域"	netapp-cvs-standard	300 GiB

CVS-Performance 服务类型

在选择存储类和创建永久性卷之前，请了解有关 CVS-Performance 服务类型的更多信息。

存储类

CVS-Performance 服务类型支持三种服务级别：标准，高级和极速。将集群添加到 Astra Control Service 时，系统会提示您选择标准，高级或极速作为永久性卷的默认存储类。其中每个服务级别都针对不同的容量和带宽需求而设计。

存储类的名称包括 *netapp-cvs-perf-standard* ， _netapp-cvs-perf-premium _ 和 _netapp-cvs-perf-Extreme 。

"有关这些服务级别的详细信息，请参见 Cloud Volumes Service for Google Cloud 文档"。

永久性卷大小和性能

"如 Google Cloud 文档所说明"，每个服务级别允许的带宽均为所配置容量的每 GiB 。这意味着，较大的卷将提供更好的性能。

请务必通读上述链接的 Google Cloud 页面。其中包括成本比较和示例，可帮助您更好地了解如何将服务级别与卷大小结合使用以实现性能目标。

最小卷大小

Astra 控制服务使用最小卷大小 100 GiB 和 CVS-Performance 服务类型来配置永久性卷，即使 PVC 请求的卷大小较小也是如此。例如，如果 Helm 图表中的 PVC 要求提供 6 GiB ，则 Astra 控制服务会自动配置 100 GiB 卷。

CVS 服务类型

在选择存储类和创建永久性卷之前，请了解有关 CVS 服务类型的更多信息。

存储类

CVS 服务类型支持一个服务级别：标准。在管理支持 CVS 服务类型的区域中的集群时， Astra 控制服务会使用标准服务级别作为永久性卷的默认存储类。此存储类名为 *netapp-cvs-standard* 。

"有关标准服务级别的详细信息，请参见 Cloud Volumes Service for Google Cloud 文档"。

永久性卷大小和性能

CVS 服务类型允许的带宽是按配置容量的每 GiB 计算的。这意味着，较大的卷将提供更好的性能。

最小卷大小

Astra 控制服务使用最小卷大小 300 GiB 和 CVS 服务类型来配置永久性卷，即使 PVC 要求的卷大小更小也是如此。例如，如果请求 20 GiB ，则 Astra 控制服务会自动配置 300 GiB 卷。

由于限制，如果 PVC 请求的卷介于 700-999 GiB 之间，则 Astra 控制服务会自动配置卷大小 1000 GiB 。

Google 持久磁盘

Astra控制服务可以使用容器存储接口(Container Storage Interface、CSI)驱动程序作为存储后端与Google Persistent Disk建立接口。此服务可提供由 Google 管理的块级存储。

"了解有关 Google 持久磁盘的更多信息"。

"详细了解 Google 持久磁盘的不同性能级别"。

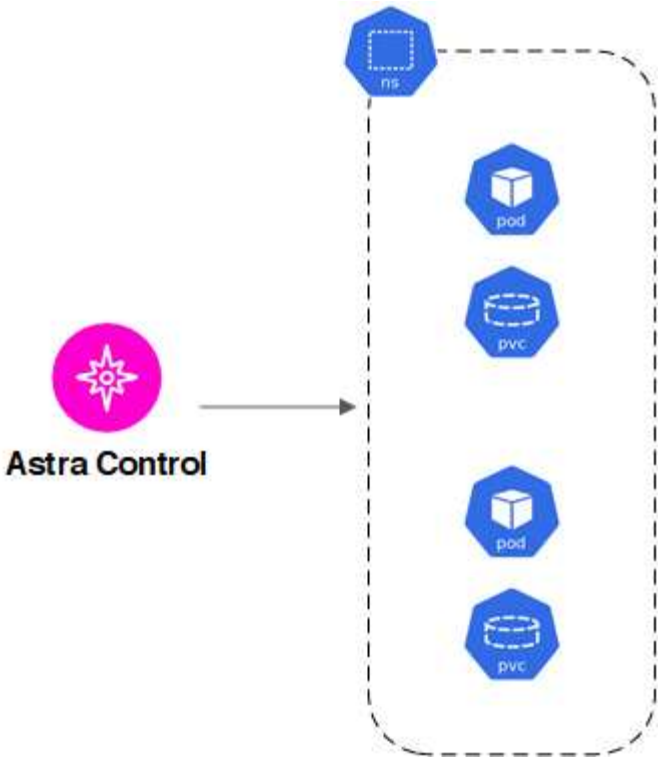
NetApp Cloud Volumes ONTAP

有关配置NetApp Cloud Volumes ONTAP 的具体信息、包括性能建议、请访问 "NetApp Cloud Volumes ONTAP 文档"。

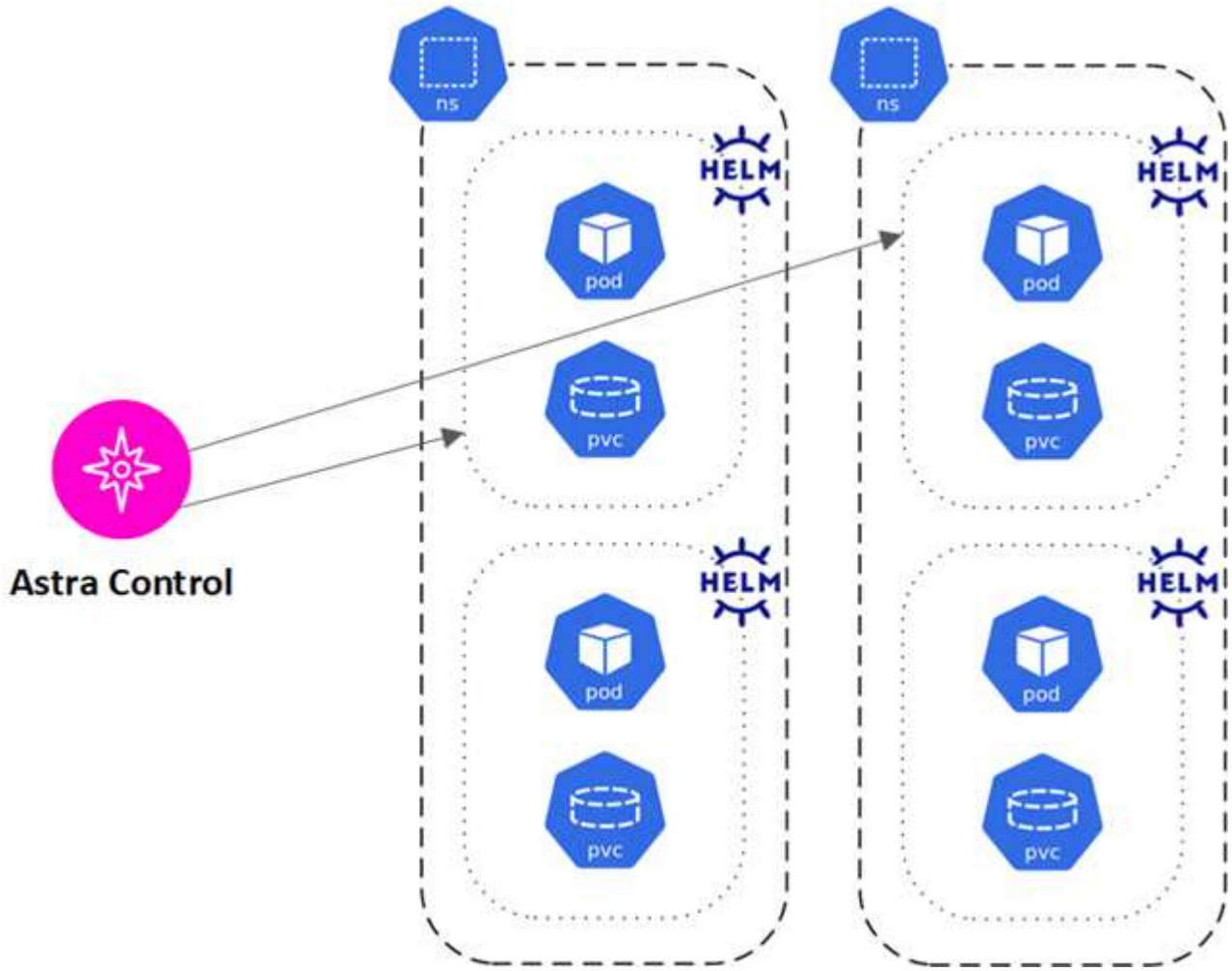
应用程序管理

当Astra Control发现集群时、这些集群上的应用程序将不受管理、直到您选择要如何管理它们为止。Astra Control 中的受管应用程序可以是以下任一项：

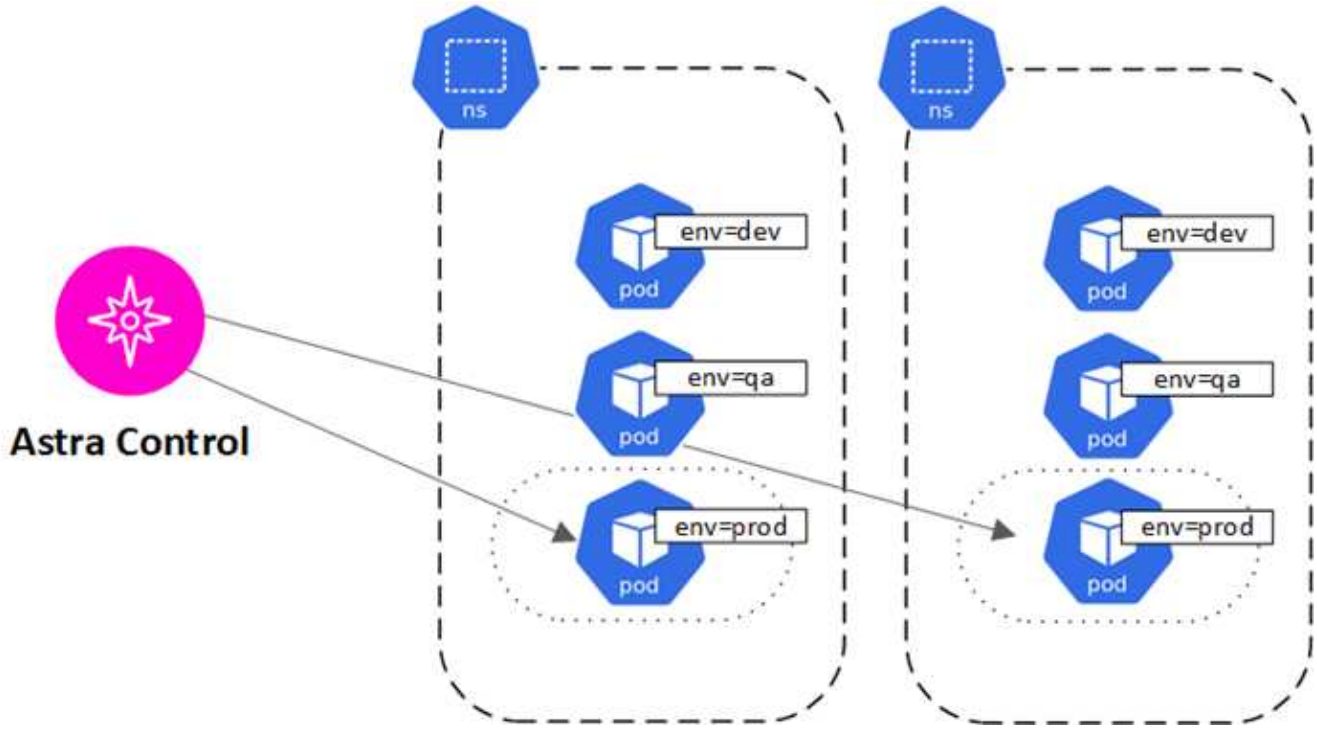
- 命名空间，包括该命名空间中的所有资源



- 部署在一个或多个命名空间中的单个应用程序(此示例使用Helm 3)



- 一组通过一个或多个命名空间中的Kubernetes标签标识的资源



用户角色和命名空间

了解 Astra Control 中的用户角色和命名空间，以及如何使用它们控制对组织中资源的访问。

用户角色

您可以使用角色控制用户对 Astra Control 资源或功能的访问权限。以下是 Astra Control 中的用户角色：

- * 所有者 * 具有管理员权限，可以删除帐户。
- * 管理员 * 具有成员权限，可以邀请其他用户。
- * 成员 * 可以全面管理应用程序和集群。
- * 查看器 * 可以查看资源。

您可以向 " 成员 " 或 " 查看器 " 用户添加限制，以将用户限制为一个或多个 [命名空间]。

命名空间

命名空间是指您可以分配给由 Astra Control 管理的集群中的特定资源的范围。将集群添加到 Astra Control 时，Astra Control 会发现集群的命名空间。发现后，可以将命名空间作为约束分配给用户。只有有权访问该命名空间的成员才能使用该资源。您可以使用命名空间来控制对资源的访问，方法是采用对您的组织有意义的模式；例如，按公司内的物理区域或部门进行访问。向用户添加约束时，您可以将该用户配置为可以访问所有命名空间或仅访问一组特定命名空间。您还可以使用命名空间标签分配命名空间约束。

了解更多信息

- "管理角色"

使用 Astra 控制服务

登录到 Astra 控制服务

通过访问，可通过基于 SaaS 的用户界面访问 Astra Control Service https://astra.netapp.io。

> 您可以使用单点登录使用公司目录中的凭据（联合身份）登录。要了解更多信息，请转到 "帮助中心" 然后选择 * Cloud Central 登录选项 * 。

开始之前

- "BlueXP用户ID"。
- "新的 Astra Control 帐户" 或 "邀请现有帐户"。
- 支持的 Web 浏览器。

  Astra 控制服务支持最新版本的 Firefox ， Safari 和 Chrome ，最小分辨率为 1280 x 720 。

步骤

1. 打开 Web 浏览器并转到 https://astra.netapp.io。
2. 使用您的NetApp BlueXP凭据登录。

管理和保护应用程序

开始管理应用程序

你先请 "将 Kubernetes 集群添加到 Astra Control"、您可以在集群上安装应用程序(在Astra Control之外)、然后转到Astra Control中的应用程序页面来定义应用程序。

您可以定义和管理包含存储资源且运行Pod的应用程序、或者包含存储资源且未运行任何Pod的应用程序。没有运行Pod的应用程序称为纯数据应用程序。

应用程序管理要求

Astra Control 具有以下应用程序管理要求：

- 许可：要管理10个以上的名称空间，您需要订阅Astra Control。
- 命名空间：可以使用Astra Control在单个集群上的一个或多个指定命名空间内定义应用程序。一个应用程序可以包含跨越同一集群中多个命名空间的资源。Astra Control不支持在多个集群之间定义应用程序。
- 存储类：如果您安装的应用程序明确设置了存储类、并且需要克隆该应用程序、则克隆操作的目标集群必须具有最初指定的存储类。将具有显式设置的存储类的应用程序克隆到没有相同存储类的集群将失败。
- * Kubernetes Resources* ：使用非 Astra Control 收集的 Kubernetes 资源的应用程序可能没有完整的应用程序数据管理功能。Astra Control 收集以下 Kubernetes 资源：

ClusterRole	ClusterRoleBinding	ConfigMap
CronJob	CustomResourceDefinition	CustomResource
DaemonSet	DeploymentConfig	HorizontalPodAutoscaler
Ingress	MutatingWebhook	NetworkPolicy
PersistentVolumeClaim	Pod	PodDisruptionBudget
PodTemplate	ReplicaSet	Role
RoleBinding	Route	Secret
Service	ServiceAccount	StatefulSet
ValidatingWebhook		

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。可以管理范围限定为一个或多个命名空间的任何应用程序。

只有当存储位于Astra Control支持的存储类上时、Astra Control才会管理有状态应用程序。Asta Control Service支持Asta Control配置程序或通用CSI驱动程序支持的任何存储类。

- "了解 GKEE 集群的存储类"
- "了解 AKS 集群的存储类"
- "了解AWS集群的存储类"

定义应用程序

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 管理跨越一个或多个命名空间的应用程序 或 将整个命名空间作为一个应用程序进行管理。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或跨命名空间)、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。

开始之前

- 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。
- 已添加到Astra Control的Kubernetes集群上的现有命名空间。
- (可选) Any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 Kubernetes 对象进行排序，组织和查找。要了解有关 Kubernetes 标签的更多信息， "请参阅Kubernetes官方文档"。

关于此任务

- 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"。
- 如果您计划在Astra Control中对应用程序使用多个命名空间、请考虑 "修改具有命名空间限制的用户角色" 定义应用程序之前。
- 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]

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发生原因 数据冲突。

阅读有关将集群范围的资源添加到应用程序命名空间的更多信息。

> 除了自动包含的Astra Control之外、您还可以导入与命名空间资源关联的集群资源。您可以添加一个规则、该规则将包含特定组的资源、种类、版本以及标签(可选)。如果存在Astra Control不会自动包含的资源、您可能需要执行此操作。
>
> 您不能排除Astra Control自动包含的任何集群范围的资源。
>
> 您可以添加以下内容 `apiVersions` (这些组与API版本结合使用)：
>
> | 资源种类 | **apiVersions (组+版本)** |
> |---|---|
> | `ClusterRole` | rbac.authorization.k8s.io/v1 |
> | `ClusterRoleBinding` | rbac.authorization.k8s.io/v1 |
> | `CustomResource` | apiextensions.k8s.io/v1、apiextensions.k8s.io/v1bea1 |
> | `CustomResourceDefinition` | apiextensions.k8s.io/v1、apiextensions.k8s.io/v1bea1 |
> | `MutatingWebhookConfiguration` | 可批准registration.K8s.IO/v1 |
> | `ValidatingWebhookConfiguration` | 可批准registration.K8s.IO/v1 |

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*下拉列表中为应用程序选择一个命名空间。

   > 可以使用Astra Control在单个集群上的一个或多个指定命名空间中定义应用程序。一个应用程序可以包含跨越同一集群中多个命名空间的资源。Astra Control不支持在多个集群之间定义应用程序。

5. (可选)为每个命名空间中的Kubernetes资源输入一个标签。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， "请参阅Kubernetes官方文档"。

6. (可选)通过选择*添加命名空间*并从下拉列表中选择命名空间来为应用程序添加其他命名空间。
7. (可选)为您添加的任何其他命名空间输入单个标签或标签选择器条件。
8. (可选)要在Astra Control自动包含的资源之外还包括集群范围的资源、请选中*包括其他集群范围的资源*并完成以下操作：

a. 选择*添加包含规则*。

b. 组：从下拉列表中、选择API资源组。

c. 种类：从下拉列表中、选择对象架构的名称。

d. 版本：输入API版本。

e. 标签选择器：也可以包括要添加到规则中的标签。此标签仅用于检索与此标签匹配的资源。如果不提供标签、则Astra Control将收集为该集群指定的所有资源类型的实例。

f. 查看根据条目创建的规则。

g. 选择 * 添加 * 。

> 您可以根据需要创建任意数量的集群范围资源规则。这些规则将显示在"定义应用程序摘要"中。

9. 选择 * 定义 * 。

10. 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中 `Healthy` 在应用程序页面上的应用程序列表中的状态。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。

要查看添加到此应用程序的资源、请选择*资源*选项卡。在资源列中选择资源名称后面的数字、或者在搜索中输入资源名称、以查看包含的其他集群范围资源。

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。如果您需要单独定义应用程序、则最好使用此方法 "打算管理和保护特定命名空间中的所有资源" 以类似的方式并按通用间隔执行。

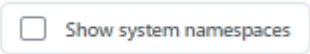
步骤

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*。

> 如果要定义多个应用程序、请从命名空间列表中进行选择、然后选择左上角的*操作*按钮并选择*定义为应用程序*。这将在各个命名空间中定义多个单独的应用程序。有关多命名空间应用程序、请参见 [定义要作为应用程序进行管理的资源]。

> 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。
> 
>
> "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`Associated applications`列中。

[技术预览]使用Kubernetes自定义资源定义应用程序

您可以通过使用自定义资源(CR)将要使用Asta Control管理的Kubbernetes资源定义为应用程序来指定这些资源。如果要单独管理某个命名空间中的资源、或者要以类似方式并按相同间隔管理和保护某个特定命名空间中的所有资源、则可以添加集群范围的资源。

步骤

1. 创建自定义资源(CR)文件并将其命名为(例如、 `astra_mysql_app.yaml`）。
2. 在中命名应用程序 `metadata.name`。
3. 定义要管理的应用程序资源：

spec.includedClusterScopedResources

除了Asta Control自动包含的资源类型之外、还包括集群范围的资源类型：

- * spec.includedClusterScopedResources*:_(可选)_要包含的集群范围资源类型列表。
  - **groupVersion Kind**：_(可选)_明确标识一种类型。
    - **group**：_(如果使用groupVersion Kind、则为必需项)_要包含的资源的API组。
    - 版本：_(如果使用groupVersionKind、则为必需项)_要包含的资源的API版本。
    - **kind**：_(如果使用groupVersion Kind、则为必填项)_要包含的资源种类。
  - **labelSelecter**：_(可选)_一组资源的标签查询。它仅用于检索与标签匹配的资源。如果不提供标签、则Astra Control将收集为该集群指定的所有资源类型的实例。将对“对等标签”和“对等显示”的结果进行AND运算。
    - 匹配标签：_(可选)_个｛key、value｝对的映射。在匹配标签映射中，单个{key,value}相当于m匹 配表达式的元素，该元素的键字段为“key”，运算符为“in”，值数组仅包含“value”。这些要求是AND。
    - 不符合要求：_(可选)_标签选择器要求列表。这些要求是AND。
      - **key**:_(如果使用了“对等表达式”，则为必需项)_与标签选择器关联的标签键。
      - 运算符：_(如果使用的是对等表达式则为必需项)_表示键与一组值的关系。有效运算符为 `In`， `NotIn`， `Exists` 和 `DoesNotExist`。
      - **values**：_(如果使用的是匹配备用)_字符串值数组。如果运算符为 `In` 或 `NotIn`，值数组必须为空。如果运算符为 `Exists` 或 `DoesNotExist`，值数组必须为空。

spec.includedNamespaces

在应用程序中的这些资源中包括名称和资源：

- 。**spec.includedNamespaces**:_(必需)_用于定义命名空间和可选的资源选择筛选器。
  - 命名空间：_(必需)_包含要使用Astra Control管理的应用程序资源的命名空间。
  - **labelSelecter**：_(可选)_一组资源的标签查询。它仅用于检索与标签匹配的资源。如果不提供标签、则Astra Control将收集为该集群指定的所有资源类型的实例。将对“对等标签”和“对等显示”的结果进行AND运算。
    - 匹配标签：_(可选)_个｛key、value｝对的映射。在匹配标签映射中，单个{key,value}相当于m匹 配表达式的元素，该元素的键字段为“key”，运算符为“in”，值数组仅包含“value”。这些要求是AND。
    - 不符合要求：_(可选)_标签选择器要求列表。 `key` 和 `operator` 为必填项。这些要求是AND。
      - **key**:_(如果使用了“对等表达式”，则为必需项)_与标签选择器关联的标签键。
      - 运算符：_(如果使用的是对等表达式则为必需项)_表示键与一组值的关系。有效运算符为 `In`， `NotIn`， `Exists` 和 `DoesNotExist`。
      - 值：_(如果使用匹配备则为必需项)_字符串值数组。如果运算符为 `In` 或 `NotIn`，值数组必须为空。如果运算符为 `Exists` 或 `DoesNotExist`，值数组必须为空。

YAML示例：

```yaml
apiVersion: astra.netapp.io/v1
kind: Application
metadata:
  name: astra_mysql_app
spec:
  includedNamespaces:
    - namespace: astra_mysql_app
      labelSelector:
        matchLabels:
          app: nginx
          env: production
        matchExpressions:
          - key: tier
            operator: In
            values:
              - frontend
              - backend
```

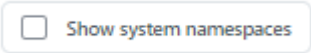
4. 在您填充之后 `astra_mysql_app.yaml` 使用正确值的文件、应用CR：

```
kubectl apply -f astra_mysql_app.yaml -n astra-connector
```

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。

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应用程序。您可以使用 Astra UI 或 "Astra Control API" 保护应用程序。

了解更多信息 "Astra Control中的数据保护"。

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]
- 为ONTAP NAS经济型操作启用备份和还原
- [创建不可还原的备份]
- [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。您可以使用Astra Control Web UI或自定义资源(CR)文件定义保护策略。

如果您需要备份或快照的运行频率高于每小时一次，则可以 "使用 Astra Control REST API 创建快照和备份"。

> 如果要定义一个保护策略来创建不可变备份以写入一次多次读取(Write 1机会 读取、WORM)分段、请确保备份的保留时间不短于为分段配置的保留期限。

> 偏移备份和复制计划以避免计划重叠。例如、在每小时的前几个小时执行备份、并计划复制、以5分钟的偏移和10分钟的间隔开始。

使用**Web UI**配置保护策略

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，每周和每月计划要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。

   在为备份设置保留级别时，您可以选择要将备份存储到的存储分段。

   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。

   [示例配置策略的屏幕截图，您可以选择每小时，每天，每周或每月创建快照和备份。]

5. [技术预览]从存储分段列表中为备份或快照选择目标分段。
6. 选择 * 审阅 * 。
7. 选择 * 设置保护策略。 *

[技术预览**]**使用**CR**配置保护策略

步骤

1. 创建自定义资源(CR)文件并将其命名为 `astra-control-schedule-cr.yaml`。更新方括号<>中的值、以匹配您的Astra Control环境、集群配置和数据保护需求：
   - <CR_NAME>：此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - <APPLICATION_NAME>：要备份的应用程序的KubeNet名称。
   - <APPVAULT_NAME>：应存储备份内容的AppVault的名称。
   - <BACKUPS_RETAINED>：要保留的备份数。零表示不应创建任何备份。
   - <SNAPSHOTS_RETAINED>：要保留的快照数量。零表示不应创建任何快照。
   - <GRANULARITY>：计划应运行的频率。可能值以及必需的关联字段：
     - `hourly` (需要您指定 `spec.minute`)
     - `daily` (需要您指定 `spec.minute` 和 `spec.hour`)
     - `weekly` (需要您指定 `spec.minute`， `spec.hour`，和 `spec.dayOfWeek`)
     - `monthly` (需要您指定 `spec.minute`， `spec.hour`，和 `spec.dayOfMonth`)
   - <DAY_OF_MONTH>：_(可选)_计划应运行的日期(1 - 31)。如果粒度设置为、则此字段为必填字段 `monthly`。
   - <DAY_OF_WEEK>：_(可选)_计划应运行的日期(0 - 7)。值0或7表示星期日。如果粒度设置为、则此字段为必填字段 `weekly`。
   - <HOUR_OF_DAY>：_(可选)_计划应运行的时间(0 - 23)。如果粒度设置为、则此字段为必填字段 `daily`， `weekly`或 `monthly`。
   - <MINUTE_OF_HOUR>：_(可选)_计划应运行的分钟(0 - 59)。如果粒度设置为、则此字段为必填

字段 hourly， daily， weekly`或 `monthly。

```
apiVersion: astra.netapp.io/v1
kind: Schedule
metadata:
  namespace: astra-connector
  name: <CR_NAME>
spec:
  applicationRef: <APPLICATION_NAME>
  appVaultRef: <APPVAULT_NAME>
  backupRetention: "<BACKUPS_RETAINED>"
  snapshotRetention: "<SNAPSHOTS_RETAINED>"
  granularity: <GRANULARITY>
  dayOfMonth: "<DAY_OF_MONTH>"
  dayOfWeek: "<DAY_OF_WEEK>"
  hour: "<HOUR_OF_DAY>"
  minute: "<MINUTE_OF_HOUR>"
```

2. 在您填充之后 `astra-control-schedule-cr.yaml` 使用正确值的文件、应用CR：

```
kubectl apply -f astra-control-schedule-cr.yaml
```

结果

Astra Control 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。

关于此任务

Astra Control支持使用以下驱动程序支持的存储类创建快照：

- `ontap-nas`
- `ontap-san`
- `ontap-san-economy`

> 如果您的应用使用由支持的存储类 `ontap-nas-economy` 驱动程序、无法创建快照。为快照使用备用存储类。

使用**Web UI**创建快照

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*下一步*。
4. [技术预览]从存储分段列表中选择快照的目标分段。
5. 查看快照摘要并选择 * 快照 * 。

[技术预览**]**使用**CR**创建快照

步骤

1. 创建自定义资源(CR)文件并将其命名为 `astra-control-snapshot-cr.yaml`。更新方括号<>中的值以匹配您的Astra Control环境和集群配置：
   - <CR_NAME>：此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - <APPLICATION_NAME>：要创建快照的应用程序的KubeNet名称。
   - <APPVAULT_NAME>：应存储快照内容的AppVault的名称。
   - <RECLAIM_POLICY>：_(可选)_定义删除快照CR时快照会发生什么情况。有效选项：
     - `Retain`
     - `Delete` (默认)

```
apiVersion: astra.netapp.io/v1
kind: Snapshot
metadata:
  namespace: astra-connector
  name: <CR_NAME>
spec:
  applicationRef: <APPLICATION_NAME>
  appVaultRef: <APPVAULT_NAME>
  reclaimPolicy: <RECLAIM_POLICY>
```

2. 在您填充之后 `astra-control-snapshot-cr.yaml` 使用正确值的文件、应用CR：

```
kubectl apply -f astra-control-snapshot-cr.yaml
```

结果

快照过程开始。如果在*数据保护*>*快照*页面的*状态*列中、快照状态为*运行状况*、则快照将成功。

创建备份

您也可以随时备份应用程序。

请注意在备份Azure NetApp Files 存储上托管的应用程序时如何处理存储空间。请参见 "应用程序备份" 有关详细信息 …

Astra Control支持使用以下驱动程序支持的存储类创建备份：

- `ontap-nas`
- `ontap-nas-economy`
- `ontap-san`
- `ontap-san-economy`

关于此任务

Astra Control中的存储分段不报告可用容量。在备份或克隆Asta Control管理的应用程序之前、请检查相应存储管理系统中的存储分段信息。

如果您的应用使用由支持的存储类 `ontap-nas-economy` 驱动程序、您需要这样做 启用备份和还原 功能。请确保您已定义 `backendType` 中的参数 "Kubbernetes存储对象" 值为 `ontap-nas-economy` 在执行任何保护操作之前。

使用**Web UI**创建备份

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. [技术预览]从存储分段列表中选择备份的目标分段。
6. 选择 * 下一步 * 。
7. 查看备份摘要并选择*备份*。

[技术预览**]**使用**CR**创建备份

步骤

1. 创建自定义资源(CR)文件并将其命名为 `astra-control-backup-cr.yaml`。更新方括号<>中的值以匹配您的Astra Control环境和集群配置：
   - <CR_NAME>：此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - <APPLICATION_NAME>：要备份的应用程序的KubeNet名称。
   - <APPVAULT_NAME>：应存储备份内容的AppVault的名称。

   ```
   apiVersion: astra.netapp.io/v1
   kind: Backup
   metadata:
     namespace: astra-connector
     name: <CR_NAME>
   spec:
     applicationRef: <APPLICATION_NAME>
     appVaultRef: <APPVAULT_NAME>
   ```

2. 在您填充之后 `astra-control-backup-cr.yaml` 使用正确值的文件、应用CR：

   ```
   kubectl apply -f astra-control-backup-cr.yaml
   ```

结果

Astra Control 会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。
- 如果需要取消正在运行的备份、请按照中的说明进行操作 [取消备份]。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]。
- 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。

为**ONTAP NAS**经济型操作启用备份和还原

Asta Control配置程序提供了备份和还原功能、可为使用的存储后端启用这些功能 `ontap-nas-economy` 存储类。

开始之前

- 您已启用Astra Control配置程序或Astra Trdent。
- 您已在Astra Control中定义了一个应用程序。在您完成此操作步骤之前、此应用程序的保护功能将受限。
- 您已拥有 `ontap-nas-economy` 已选择作为存储后端的默认存储类。

展开以了解配置步骤

1. 在ONTAP存储后端执行以下操作：

   a. 查找托管的SVM `ontap-nas-economy`应用程序的基于卷。

   b. 登录到连接到创建卷的ONTAP的终端。

   c. 隐藏SVM的Snapshot目录：

   > 此更改会影响整个SVM。隐藏的目录将继续可访问。

```
nfs modify -vserver <svm name> -v3-hide-snapshot enabled
```

+

> 验证ONTAP存储后端上的Snapshot目录是否已隐藏。如果未能隐藏此目录、可能会导致无法访问您的应用程序、尤其是在使用NFSv3的情况下。

2. 在Asta Control配置程序或Asta Trdent中执行以下操作：

   a. 为每个基于ONTAP NAS经济型且与应用程序关联的PV启用Snapshot目录：

```
tridentctl update volume <pv name> --snapshot-dir=true --pool
-level=true -n trident
```

   b. 确认已为每个关联PV启用Snapshot目录：

```
tridentctl get volume <pv name> -n trident -o yaml | grep
snapshotDir
```

   响应：

```
snapshotDirectory: "true"
```

3. 在Astra Control中、启用所有关联的快照目录后刷新应用程序、以便Astra Control识别更改后的值。

结果

该应用程序已准备好使用Astra Control进行备份和还原。每个PVC还可供其他应用程序用于备份和恢复。

创建不可还原的备份

只要存储不可变备份的存储分段上的保留策略禁止、就无法修改、删除或覆盖该备份。您可以通过将应用程序备份到配置了保留策略的存储分段来创建不可配置的备份。请参见 "数据保护" 了解有关使用不可配置备份的重要

信息。

开始之前

您需要使用保留策略配置目标存储分段。根据您使用的存储提供程序、执行此操作的方式会有所不同。有关详细信息、请参见存储提供程序文档：

- **Amazon Web Services**： "创建存储分段时启用S3对象锁定、并设置默认保留模式"g监管"和默认保留期限"。
- **Google Cloud**： "使用保留策略配置存储分段并指定保留期限"。
- *Microsoft Azure"： "在容器级别范围上使用基于时间的保留策略配置Blob存储分段"。
- * NetApp StorageGRID *： "创建存储分段时启用S3对象锁定、并将默认保留模式设置为"Compliance (合规性)"和默认保留期限"。

> Astra Control中的存储分段不报告可用容量。在备份或克隆Asta Control管理的应用程序之前、请检查相应存储管理系统中的存储分段信息。

> 如果您的应用使用由支持的存储类 `ontap-nas-economy` 驱动程序、请确保您已定义 `backendType` 中的参数 "Kubbernetes存储对象" 值为 `ontap-nas-economy` 在执行任何保护操作之前。

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. 从存储分段列表中为备份选择一个目标分段。一次写入、多次读取(WORM)存储分段的状态在存储分段名称旁边显示为"已锁定"。

   > 如果存储分段类型不受支持、则在将鼠标悬停在存储分段上或选择存储分段时会指示此情况。

6. 选择 * 下一步 * 。
7. 查看备份摘要并选择*备份*。

结果

Astra Control可为应用程序创建不可移动的备份。

> - 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。
> - 如果您尝试同时为同一应用程序创建两个不可变备份到同一存储分段、Astra Control会阻止第二个备份启动。等待第一个备份完成、然后再启动另一个备份。
> - 您无法取消正在运行的不可更改备份。
> - 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。

> 不可还原备份会在其所使用的存储分段旁边显示状态为"已锁定"。

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 * 。

   默认情况下会显示快照。

3. 选择*备份*以引用备份列表。

删除快照

删除不再需要的计划快照或按需快照。

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 * 。
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 Control 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、备份必须位于中 `Running` 状态。您无法取消中的备份 `Pending` 状态。

> 您无法取消正在运行的不可更改备份。

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"以确认操作、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。

> 如果需要取消正在运行的备份、请按照中的说明进行操作 [取消备份]。要删除备份、请等待备份完成、然后按照以下说明进行操作。

> 在保留期限到期之前、您不能删除不可更改的备份。

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 Control 会删除备份。

[技术预览]保护整个集群

您可以为集群上的任何或所有非受管卷创建计划的自动备份。这些工作流由NetApp以Kubbernetes服务帐户、角色绑定和cron作业的形式提供、并使用Python脚本进行编排。

工作原理

在配置和安装完整集群备份工作流时、cron作业会定期运行、并保护尚未管理的任何命名空间、从而根据您在安装期间选择的计划自动创建保护策略。

如果您不希望使用完整集群备份工作流保护集群上的每个非受管命名空间、则可以改用基于标签的备份工作流。基于标签的备份工作流也会使用cron任务、但它不会保护所有非受管命名库、而是通过您提供的标签来标识命名库、以根据铜牌、银牌或金牌备份策略保护命名库。

在所选工作流范围内创建新命名空间时、该命名空间会自动受到保护、无需任何管理员操作。这些工作流是按集群实施的、因此不同的集群可以根据集群的重要性使用任一工作流、并具有独特的保护级别。

示例：完全集群保护

例如、在配置和安装完整集群备份工作流时、任何命名空间中的任何应用程序都将定期进行管理和保护、而无需管理员进一步努力。安装工作流时、命名空间不必存在；如果将来添加命名空间、它将受到保护。

示例：基于标签的保护

要获得更精细的粒度、您可以使用基于标签的工作流。例如、您可以安装此工作流、并告诉用户根据所需的保护级别、将多个标签之一应用于要保护的任何命名区域。这样、用户就可以使用其中一个标签创建命名空间、而无需通知管理员。它们的新命名空间以及其中的所有应用程序都会自动受到保护。

为所有的名段创建计划备份

您可以使用完整集群备份工作流为集群上的所有名空间创建计划备份。

步骤

1. 将以下文件下载到可通过网络访问集群的计算机：
    - "Components.YAML CRD文件"
    - "protectCluster.py Python脚本"
2. 要配置和安装此工具包、请执行以下步骤： "按照附带的说明进行操作"。

为特定的名段创建计划备份

您可以使用基于标签的备份工作流按标签为特定命名库创建计划备份。

步骤

1. 将以下文件下载到可通过网络访问集群的计算机：
    - "Components.YAML CRD文件"
    - "protectCluster.py Python脚本"
2. 要配置和安装此工具包、请执行以下步骤： "按照附带的说明进行操作"。

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。

> 如果将命名空间筛选器添加到在还原或克隆操作之后运行的执行挂钩、并且还原或克隆源和目标位于不同的命名空间中、则命名空间筛选器仅会应用于目标命名空间。

开始之前

- 首先保护您的应用程序：强烈建议您在恢复应用程序之前为其创建快照或备份。这样、如果还原失败、您就可以从快照或备份克隆。
- 检查目标卷：如果要还原到其他存储类、请确保该存储类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。例如、如果源永久性卷使用rwx访问模式、请选择无法提供rwx的目标存储类、例如Azure托管磁盘、AWS EBS、Google持久磁盘或 `ontap-san`，发生原因则还原操作是否会失败。有关永久性卷访问模式的详细信息、请参阅 "Kubernetes" 文档。
- 规划空间需求：对使用NetApp ONTAP 存储的应用程序执行原位还原时、还原的应用程序使用的空间可能会增加一倍。执行原位还原后、从还原的应用程序中删除所有不需要的快照以释放存储空间。
- 支持的存储类驱动程序：Astra Control支持使用以下驱动程序支持的存储类还原备份：
    - `ontap-nas`
    - `ontap-nas-economy`
    - `ontap-san`
    - `ontap-san-economy`
- **(**仅限**ONTA-NAS**经济型驱动程序**)**备份和还原：备份或还原使用由备份的存储类的应用程序之前 `ontap-nas-economy` 驱动程序、请验证 "ONTAP存储后端上的Snapshot目录处于隐藏状态"。如果未能隐藏此目录、可能会导致无法访问您的应用程序、尤其是在使用NFSv3的情况下。

> 在与其他应用程序共享资源的应用程序上执行原位还原操作可能会产生意外结果。对其中一个应用程序执行原位还原时、这些应用程序之间共享的任何资源都会被替换。

步骤

1. 选择 * 应用程序 * ，然后选择应用程序的名称。
2. 从“操作”列的“选项”菜单中，选择*Restore*。
3. 选择还原类型：
   - 还原到原始命名空间：使用此操作步骤 将应用程序原位还原到原始集群。
     i. 选择要用于原位还原应用程序的快照或备份、这会将应用程序还原到其自身的早期版本。
     ii. 选择 * 下一步 * 。

     > 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。

   - 还原到新命名空间：使用此操作步骤 将应用程序还原到另一个集群或使用与源不同的命名空间。您还可以使用此操作步骤将应用程序迁移到其他存储类。
     i. 指定已还原应用程序的名称。
     ii. 为要还原的应用程序选择目标集群。
     iii. 为与应用程序关联的每个源命名空间输入目标命名空间。

     > 作为此还原选项的一部分、Astra Control会创建新的目标命名空间。指定的目标命名空间不能已存在于目标集群上。

     iv. 选择 * 下一步 * 。
     v. 选择用于还原应用程序的快照或备份。
     vi. 选择 * 下一步 * 。
     vii. 选择以下选项之一：
       - 使用原始存储类还原：除非目标集群上不存在、否则应用程序将使用最初关联的存储类。在这种情况下、将使用集群的默认存储类。
       - 使用其他存储类还原：选择目标集群上的存储类。在还原过程中、所有应用程序卷(无论其最初关联的存储类是什么)都将迁移到此不同的存储类。
     viii. 选择 * 下一步 * 。
4. 选择要筛选的任何资源：
   - 恢复所有资源：恢复与原始应用程序关联的所有资源。
   - 过滤资源：指定规则以还原原始应用程序资源的子集：
     i. 选择在已还原的应用程序中包括或排除资源。
     ii. 选择*添加包含规则*或*添加排除规则*，并配置规则以在应用程序恢复期间过滤正确的资源。您可以编辑或删除规则、然后重新创建规则、直到配置正确为止。

要了解有关配置包含和排除规则的信息、请参见 [在应用程序还原期间筛选资源]。

5. 选择 * 下一步 * 。
6. 请仔细查看有关还原操作的详细信息，键入“restore”(如果出现提示)，然后选择*Restore*。

[技术预览]使用自定义资源从备份中恢复(CR)

您可以使用自定义资源(CR)文件将备份中的数据还原到其他命名空间或原始源命名空间。

使用**CR**从备份还原

步骤

1. 创建自定义资源(CR)文件并将其命名为 `astra-control-backup-restore-cr.yaml`。更新方括号<>中的值以匹配您的Astra Control环境和集群配置：
   - <CR_NAME>：此CR操作的名称；为您的环境选择一个合理的名称。
   - <APPVAULT_NAME>：存储备份内容的AppVault的名称。
   - <BACKUP_PATH>：AppVault中存储备份内容的路径。例如：

     ```
     ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
     20231213023800_94347756-9d9b-401d-a0c3
     ```

   - <SOURCE_NAMESPACE>：还原操作的源命名空间。
   - <DESTINATION_NAMESPACE>：还原操作的目标命名空间。

     ```
     apiVersion: astra.netapp.io/v1
     kind: BackupRestore
     metadata:
       name: <CR_NAME>
       namespace: astra-connector
     spec:
       appVaultRef: <APPVAULT_NAME>
       appArchivePath: <BACKUP_PATH>
       namespaceMapping: [{"source": "<SOURCE_NAMESPACE>",
     "destination": "<DESTINATION_NAMESPACE>"}]
     ```

<stdin>中未解析的指令- include：..//_include/Selectic-Restore-cr.adoc[]

1. 在您填充之后 `astra-control-backup-restore-cr.yaml` 使用正确值的文件、应用CR：

   ```
   kubectl apply -f astra-control-backup-restore-cr.yaml
   ```

使用**CR**从备份还原到原始命名空间

步骤

1. 创建自定义资源(CR)文件并将其命名为 `astra-control-backup-ipr-cr.yaml`。更新方括号<>中的值以匹配您的Astra Control环境和集群配置：
   - <CR_NAME>：此CR操作的名称；为您的环境选择一个合理的名称。
   - <APPVAULT_NAME>：存储备份内容的AppVault的名称。
   - <BACKUP_PATH>：AppVault中存储备份内容的路径。例如：

```
ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
20231213023800_94347756-9d9b-401d-a0c3
```

```yaml
apiVersion: astra.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  name: <CR_NAME>
  namespace: astra-connector
spec:
  appVaultRef: <APPVAULT_NAME>
  appArchivePath: <BACKUP_PATH>
```

<stdin>中未解析的指令- include：..//_include/Selectic-Restore-cr.adoc[]

1. 在您填充之后 `astra-control-backup-ipr-cr.yaml` 使用正确值的文件、应用CR：

```
kubectl apply -f astra-control-backup-ipr-cr.yaml
```

[技术预览]使用自定义资源从快照恢复(CR)

您可以使用自定义资源(CR)文件从快照将数据还原到其他命名空间或原始源命名空间。

使用**CR**从快照还原

步骤

1. 创建自定义资源(CR)文件并将其命名为 `astra-control-snapshot-restore-cr.yaml`。更新方括号<>中的值以匹配您的Astra Control环境和集群配置：
   - <CR_NAME>：此CR操作的名称；为您的环境选择一个合理的名称。
   - <APPVAULT_NAME>：存储备份内容的AppVault的名称。
   - <BACKUP_PATH>：AppVault中存储备份内容的路径。例如：

     ```
     ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
     20231213023800_94347756-9d9b-401d-a0c3
     ```

   - <SOURCE_NAMESPACE>：还原操作的源命名空间。
   - <DESTINATION_NAMESPACE>：还原操作的目标命名空间。

     ```
     apiVersion: astra.netapp.io/v1
     kind: SnapshotRestore
     metadata:
       name: <CR_NAME>
       namespace: astra-connector
     spec:
       appArchivePath: <BACKUP_PATH>
       appVaultRef: <APPVAULT_NAME>
       namespaceMapping: [{"source": "<SOURCE_NAMESPACE>",
     "destination": "<DESTINATION_NAMESPACE>"}]
     ```

<stdin>中未解析的指令- include：..//_include/Selectic-Restore-cr.adoc[]

1. 在您填充之后 `astra-control-snapshot-restore-cr.yaml` 使用正确值的文件、应用CR：

   ```
   kubectl apply -f astra-control-snapshot-restore-cr.yaml
   ```

使用**CR**从快照还原到原始命名空间

步骤

1. 创建自定义资源(CR)文件并将其命名为 `astra-control-snapshot-ipr-cr.yaml`。更新方括号<>中的值以匹配您的Astra Control环境和集群配置：
   - <CR_NAME>：此CR操作的名称；为您的环境选择一个合理的名称。
   - <APPVAULT_NAME>：存储备份内容的AppVault的名称。
   - <BACKUP_PATH>：AppVault中存储备份内容的路径。例如：

```
ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
20231213023800_94347756-9d9b-401d-a0c3
```

```yaml
apiVersion: astra.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  name: <CR_NAME>
  namespace: astra-connector
spec:
  appArchivePath: <BACKUP_PATH>
  appVaultRef: <APPVAULT_NAME>
```

<stdin>中未解析的指令- include：..//_include/Selectic-Restore-cr.adoc[]

1. 在您填充之后 `astra-control-snapshot-ipr-cr.yaml` 使用正确值的文件、应用CR：

```
kubectl apply -f astra-control-snapshot-ipr-cr.yaml
```

结果

Astra Control 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、则现有永久性卷的内容将替换为已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、备份或还原)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。

> 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。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。

在应用程序还原期间筛选资源

您可以向添加筛选器规则 "还原" 此操作将指定要从还原的应用程序中包括或排除的现有应用程序资源。您可以根据指定的命名空间、标签或GVK (GroupVersion Kind)包括或排除资源。

阅读有关包含和排除方案的更多信息

- 选择包含原始命名空间的规则(就地还原)：您在规则中定义的现有应用程序资源将被删除，并替换为用于还原的选定快照或备份中的应用程序资源。未在包含规则中指定的任何资源将保持不变。
- 选择包含新名称空间的规则：使用此规则在还原的应用程序中选择所需的特定资源。未在包含规则中指定的任何资源将不会包含在已还原的应用程序中。
- 选择包含原始名称空间的排除规则(就地恢复)：您指定要排除的资源将不会还原、并且保持不变。未指定排除的资源将从快照或备份中还原。如果筛选的资源中包含相应的状态集、则永久性卷上的所有数据都将被删除并重新创建。
- 选择包含新名称空间的排除规则：使用此规则可选择要从还原的应用程序中删除的特定资源。未指定排除的资源将从快照或备份中还原。

规则可以是包含类型、也可以是排除类型。不提供组合使用资源包含和排除的规则。

步骤

1. 选择筛选资源并在恢复应用程序向导中选择包含或排除选项后，选择*添加包含规则*或*添加排除规则*。

   您不能排除Asta Control自动包含的任何集群范围的资源。

2. 配置筛选器规则：

   必须至少指定一个命名空间、标签或GVK。确保在应用筛选器规则后保留的任何资源足以使已还原的应用程序保持运行状况良好。

   a. 为规则选择特定命名空间。如果不进行选择、则会在筛选器中使用所有名称空间。

      如果您的应用程序最初包含多个名称空间、而您将其还原到新的名称空间、则会创建所有名称空间、即使它们不包含资源也是如此。

   b. (可选)输入资源名称。

   c. (可选)标签选择器：包括A "标签选择器" 以添加到规则中。标签选择器用于仅筛选与选定标签匹配的资源。

   d. (可选)选择*使用GVK (GroupVersion Kind)设置来筛选资源*以获取其他筛选选项。

      如果使用GVK筛选器、则必须指定版本和种类。

      i. (可选)组：从下拉列表中选择Kubernetes API组。

      ii. **KND**：从下拉列表中选择要在筛选器中使用的Kubernetes资源类型的对象模式。

      iii. 版本：选择Kubernetes API版本。

3. 查看根据条目创建的规则。

4. 选择 * 添加 * 。

   您可以根据需要创建任意数量的资源包含和排除规则。这些规则将显示在启动操作之前的还原应用程序摘要中。

克隆和迁移应用程序

您可以克隆现有应用程序、以便在同一个Kubernetes集群或另一个集群上创建重复的应用程序。当 Astra Control 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。

> 如果将命名空间筛选器添加到在还原或克隆操作之后运行的执行挂钩、并且还原或克隆源和目标位于不同的命名空间中、则命名空间筛选器仅会应用于目标命名空间。

开始之前

- 检查目标卷：如果克隆到其他存储类、请确保该存储类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、则克隆操作将失败。例如、如果源永久性卷使用rwx访问模式、请选择无法提供rwx的目标存储类、例如Azure托管磁盘、AWS EBS、Google持久磁盘或 `ontap-san`，发生原因将使克隆操作失败。有关永久性卷访问模式的详细信息、请参阅 "Kubernetes" 文档。
- 要将应用程序克隆到其他集群、您需要确保已为包含源集群的云实例分配默认分段。如果源云实例未设置默认分段、则跨集群克隆操作将失败。
- 在克隆操作期间、需要IngressClass资源或webhooks才能正常运行的应用程序不能在目标集群上定义这些资源。

克隆限制

- 显式存储类：如果部署的应用程序已明确设置存储类、并且需要克隆此应用程序、则目标集群必须具有最初指定的存储类。将具有显式设置的存储类的应用程序克隆到没有相同存储类的集群将失败。
- **UNONTAP NAS**经济型应用程序：如果应用程序的存储类由提供支持、则无法使用克隆操作 `ontap-nas-economy` 驱动程序。但是、您可以 "为ONTAP NAS经济型操作启用备份和还原"。
- 克隆和用户约束：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。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。
- 克隆使用默认分段：
  - 在应用程序备份或应用程序还原期间、您可以指定要使用的存储分段。在跨集群克隆时、需要指定默认分段、但在同一集群内克隆时、指定分段是可选的。
  - 在集群间克隆时、包含克隆操作的源集群的云实例必须设置默认分段。
  - 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。
- 使用**Jenkins CI**：如果克隆操作员部署的Jenkins CI实例、则需要手动还原持久数据。这是应用程序部署模式的一个限制。

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. 指定克隆的详细信息：
   - 输入名称。
   - 选择克隆的目标集群。
   - 输入克隆的目标命名空间。与应用程序关联的每个源命名空间都会映射到一个目标命名空间。

     在克隆操作中、Astra Control会创建新的目标命名空间。指定的目标命名空间不能已存在于目标集群上。

   - 选择 * 下一步 * 。
   - 选择将原始存储类与应用程序保持关联、或者选择其他存储类。

     您可以将应用程序的存储类迁移到本机云提供商存储类或其他受支持的存储类、也可以将应用程序从支持的存储类迁移 `ontap-nas-economy` 存储类 `ontap-nas` 在同一集群上、或者将应用程序复制到存储类由支持的另一集群 `ontap-nas-economy` 驱动程序。

   如果您选择了其他存储类、但在还原时此存储类不存在、则会返回错误。

5. 选择 * 下一步 * 。
6. 查看有关克隆的信息、然后选择*克隆*。

结果

Astra Control会根据您提供的信息克隆应用程序。当新应用程序克隆处于中时、克隆操作成功 `Healthy` 状态。

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。

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 Service支持以下类型的执行挂钩、具体取决于它们可以运行的时间：

- 预快照
- 快照后
- 预备份
- 备份后
- 还原后

执行钩筛选器

向应用程序添加或编辑执行挂钩时，可以向执行挂钩添加过滤器，以管理挂钩将匹配的容器。对于在所有容器上使用相同容器映像的应用程序、筛选器非常有用、但可能会将每个映像用于不同的用途(例如Elasticsearch)。通过筛选器、您可以创建执行挂钩在某些容器上运行的方案、但不一定是所有相同的容器上运行的方案。如果为单个执行钩创建多个筛选器、则这些筛选器将与逻辑运算符和运算符结合使用。每个执行连接最多可以有10个活动筛选器。

添加到执行挂钩中的每个筛选器都会使用一个正则表达式来匹配集群中的容器。当某个挂钩与某个容器匹配时、该挂钩将在该容器上运行其关联脚本。筛选器的正则表达式使用正则表达式2 (RE2)语法、不支持创建从匹配列表中排除容器的筛选器。有关Astra Control在执行挂钩筛选器中支持正则表达式语法的信息、请参见 "正则表达式2 (RE2)语法支持"。

> 如果将命名空间筛选器添加到在还原或克隆操作之后运行的执行挂钩、并且还原或克隆源和目标位于不同的命名空间中、则命名空间筛选器仅会应用于目标命名空间。

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。

> 由于执行挂钩通常会减少或完全禁用其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、备份后执行挂钩中使用的逻辑不能假定备份已完成。

- 默认情况下、对于新的Astra Control部署、执行挂钩功能处于禁用状态。
  - 您需要先启用执行挂钩功能、然后才能使用执行挂钩。
  - 所有者或管理员用户可以为当前Astra Control帐户中定义的所有用户启用或禁用执行挂钩功能。请参见 [启用执行挂钩功能] 和 [禁用执行挂钩功能] 有关说明，请参见。
  - 在Astra Control升级期间、功能启用状态会保留下来。
- 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，所有钩子事件都会生成并保存在*Activity*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)。

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、具有所有不同类型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

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**
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	不包括	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	不包括	不适用

执行钩示例

请访问 "NetApp Verda GitHub项目" 为Apache Cassandra和Elasticsearch等常见应用程序下载真正的执行挂钩。您还可以查看示例并了解如何构建自己的自定义执行挂钩。

启用执行挂钩功能

如果您是所有者或管理员用户、则可以启用执行挂钩功能。启用此功能时、此Astra Control帐户中定义的所有用户都可以使用执行挂钩并查看现有执行挂钩和挂钩脚本。

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。
3. 选择*启用执行挂钩*。

   出现*Account*>*Feature settings (功能设置)*选项卡。

4. 在*执行挂钩*窗格中，选择设置菜单。
5. 选择 * 启用 * 。
6. 注意出现的安全警告。
7. 选择*是，启用执行挂钩*。

禁用执行挂钩功能

如果您是所有者或管理员用户、则可以对此Astra Control帐户中定义的所有用户禁用执行挂钩功能。您必须先删除所有现有的执行挂钩、然后才能禁用执行挂钩功能。请参见 [删除执行挂钩] 有关删除现有执行挂钩的说明。

步骤

1. 进入*Account*，然后选择*Feature settings (功能设置)*选项卡。
2. 选择 * 执行挂钩 * 选项卡。

3. 在*执行挂钩*窗格中，选择设置菜单。
4. 选择 * 禁用 * 。
5. 注意出现的警告。
6. Type `disable` 确认要为所有用户禁用此功能。
7. 选择*是，禁用*。

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。您可以查看挂钩的状态、匹配的容器数量、创建时间以及运行时间(操作前或操作后)。您可以选择 + 此挂机名称旁边的图标可展开要运行它的容器列表。要查看与此应用程序的执行挂钩相关的事件日志、请转到*活动*选项卡。

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

每个执行挂钩都必须使用脚本执行操作。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、您只需更改一个脚本、即可更新多个执行挂钩。

步骤

1. 确保执行钩子功能为 enabled。
2. 转到*帐户*。
3. 选择*脚本*选项卡。
4. 选择 * 添加 * 。
5. 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。
6. 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、并将其添加到Astra Control中。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. 确保执行钩子功能为 enabled。
2. 选择 * 应用程序 * ，然后选择受管应用程序的名称。
3. 选择 * 执行挂钩 * 选项卡。
4. 选择 * 添加 * 。
5. 在*挂机详细信息*区域中：
    a. 从*操作*下拉菜单中选择操作类型、以确定何时应运行挂钩。
    b. 输入此挂钩的唯一名称。

c. （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。

6. (可选)在*挂机筛选器详细信息*区域中、您可以添加筛选器来控制执行挂机运行在哪些容器上：
   a. 选择*添加筛选器*。
   b. 在*挂机筛选器类型*列中、从下拉菜单中选择要筛选的属性。
   c. 在*正则表达式*列中、输入要用作筛选器的正则表达式。Astra Control使用 "正则表达式2 (RE2)正则表达式语法"。

   > 如果在正则表达式字段中筛选某个属性的确切名称(例如Pod名称)而不包含其他文本、则会执行子字符串匹配。要匹配确切的名称以及仅匹配该名称、请使用精确的字符串匹配语法(例如、 `^exact_podname$`）。

   d. 要添加更多筛选器、请选择*添加筛选器*。

   > 一个执行钩的多个筛选器与一个逻辑运算符和运算符结合使用。每个执行连接最多可以有10个活动筛选器。

7. 完成后、选择*下一步*。
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。
   - 从列表中选择一个现有脚本。

     这将指示执行挂钩使用此脚本。

9. 选择 * 下一步 * 。
10. 查看执行钩配置。
11. 选择 * 添加 * 。

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。

编辑执行挂钩

如果要更改执行挂钩的属性、筛选器或所使用的脚本、您可以编辑该执行挂钩。要编辑执行挂钩、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*列中选择要编辑的挂钩的选项菜单。
4. 选择 * 编辑 * 。
5. 完成每个部分后、选择*下一步*进行所需的更改。
6. 选择 * 保存 * 。

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 * 。
5. 在显示的对话框中、键入"delete"进行确认。
6. 选择*是、删除执行钩*。

有关详细信息 ...

- "NetApp Verda GitHub项目"

查看应用程序和计算运行状况

查看应用程序和集群运行状况摘要

单击 * 信息板 * 可查看应用程序，集群及其运行状况的详细视图。

" 应用程序 " 图块可帮助您确定以下内容：

- 您当前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）。

请注意，这些不仅仅是数字或状态，您可以从其中的每种状态深入了解。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。

" 集群 " 图块提供了有关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。

查看集群的运行状况和详细信息

将 Kubernetes 集群添加到 Astra Control 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 控制服务 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。

> 如果集群处于 `removed` 状态，而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），则您为 Astra Control 提供的 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：与集群相关的活动。

> 您还可以从 Astra 控制服务 * 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

开始管理应用程序后、Astra Control会提供有关该应用程序的详细信息、您可以通过这些信息确定其通信状态(Astra Control是否可以与该应用程序通信)、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. 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Asta Control是否可以与应用程序通信的状态。

   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

与应用程序相关的 Astra Control 活动。

管理存储分段

您可以管理Astra用于备份和克隆的存储分段。您可以在云实例中添加其他分段、删除现有分段以及更改Kubernetes集群的默认分段。

只有所有者和管理员才能管理存储分段。

Astra Control 如何使用存储分段

当您开始管理云实例的第一个Kubernetes集群时、Astra Control Service会为此创建初始存储分段 "云实例"。

您可以手动将存储分段指定为云实例的默认存储分段。如果您这样做、则Astra控制服务会默认对您在该云实例中的任何受管集群上创建的备份和克隆使用此存储分段(您可以选择其他存储分段进行备份)。如果您将应用程序从云实例中的任何受管集群实时克隆到另一个集群、则Astra Control Service会使用源云实例的默认分段来执行克隆操作。

您可以为多个云实例设置与默认分段相同的分段。

您可以在创建保护策略或启动临时备份时从任何存储分段中进行选择。

在启动备份或克隆之前， Astra 控制服务会检查目标存储分段是否可访问。

查看现有存储分段

查看可供Astra控制服务使用的分段列表、以确定其状态并确定云实例的默认分段(如果已定义)。

存储分段可以具有以下任一状态：

待定

添加存储分段后、它将以待定状态启动、而Astra Control会发现它。

可用

此存储分段可供 Astra Control 使用。

已删除

存储分段目前无法正常运行。将鼠标悬停在状态图标上可确定问题所在。

如果某个存储分段处于 "Removed" 状态，您仍可以将其设置为默认存储分段并将其分配给保护计划。但是，如果数据保护操作开始时存储分段未处于可用状态，则该操作将失败。

步骤

1. 转至*分段器*。

   此时将显示可供Astra控制服务使用的分段列表。

添加一个额外的存储分段

您可以随时添加其他分段。这样、您可以在创建保护策略或启动临时备份时在存储分段之间进行选择、并可以更改云实例使用的默认存储分段。

您可以添加以下类型的存储分段：

- Amazon Web Services
- 通用 S3
- Google 云平台
- Microsoft Azure
- NetApp ONTAP S3
- NetApp StorageGRID S3

开始之前

- 确保您知道现有存储分段的名称。
- 确保您拥有为Astra Control提供管理存储分段所需权限的存储分段凭据。
- 如果存储分段位于Microsoft Azure中：
  - 此存储分段必须属于名为_Astra-backup-rg_的资源组。
  - 如果Azure存储帐户实例性能设置为"Premium"、则"Premium account type"设置必须设置为"Block blobs"。

步骤

1. 转至*分段器*。
2. 选择 * 添加 * ，然后按照提示添加存储分段。
   - * 类型 * ： 选择您的云提供商。
   - * 现有存储分段名称 * ： 输入存储分段的名称。
   - * 问题描述 * ： 也可以输入存储分段的问题描述。
     - 存储帐户(仅限Azure)：输入Azure存储帐户的名称。此存储分段必须属于名为_Astra-backup-rg_的资源组。
     - * S3服务器名称或IP地址*(仅限AWS和S3存储分段类型)：输入与您所在地区对应的S3端点的完全限定域名、而不输入 `https://`。请参见 "Amazon文档" 有关详细信息 ...

- 选择凭据：输入为Astra控制服务提供管理存储分段所需权限的凭据。您需要提供的信息因存储分段类型而异。
  a. 选择 * 添加 * 以添加存储分段。

结果

Astra Control Service添加了存储分段。现在、您可以在创建保护策略或执行临时备份时选择此存储分段。您也可以将此分段设置为云实例的默认分段。

更改默认分段

您可以更改云实例的默认存储分段。默认情况下、Astra Control Service会将此存储分段用于备份和克隆。每个云实例都有自己的默认存储分段。

> Astra Control不会自动为任何云实例分配默认分段。在两个集群之间执行应用程序克隆操作之前、您需要手动为云实例设置默认分段。

步骤

1. 转至*云实例*。
2. 在*操作*列中选择要编辑的云实例的配置菜单。
3. 选择 * 编辑 * 。
4. 在存储分段列表中、选择要用作此云实例的默认存储分段的存储分段。
5. 选择 * 更新 * 。

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。
> - 在"一次写入、多次读取"(WORM)分段的云提供程序保留期限到期之前、您不能删除该分段。WORm分段名称旁用"已锁定"表示。

开始之前

- 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，以确保存储分段未用于任何计划的备份。

如果存在，您将无法继续。

步骤

1. 转至*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 * 。

[技术预览]使用自定义资源管理存储分段

您可以使用应用程序集群上的Astra Control自定义资源(CR)添加存储分段。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，则必须添加对象存储分段提供程序。Astra Control 会将这些备份或克隆存储在您定义的对象存储分段中。如果使用的是自定义资源方法、则应用程序快照功能需要一个存储分段。

如果您要将应用程序配置和永久性存储克隆到同一集群、则无需在Astra Control中使用存储分段。

Asta Control的存储分段自定义资源称为AppVault。此CR包含在保护操作中使用存储分段所需的配置。

开始之前

- 确保您有一个可从Astra Control Center管理的集群访问的存储分段。
- 确保您具有此存储分段的凭据。
- 确保存储分段为以下类型之一：
  - 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。

步骤

1. 创建自定义资源(CR)文件并将其命名为(例如、 `astra-appvault.yaml`）。
2. 配置以下属性：
   - * metadata.name*:_(必需)_ AppVault自定义资源的名称。
   - **spec.prefix**：_(可选)_一个路径、该路径前缀为存储在AppVault中的所有实体的名称。
   - 。**spec.providerConfig**:_(必需)_用于存储使用指定提供程序访问AppVault所需的配置。
   - * spec.providerCredentials*:_(必需)_存储使用指定提供程序访问AppVault所需的任何凭据的引用。
     - * spec.providerCredentials.valueFromSecret*:_(可选)_表示凭据值应来自机密。
       - **key**:_(如果使用了valueFroMSecret)密钥的有效密钥_。
       - **name**:_(如果使用valueF物品 密钥，则为必需项)_包含此字段值的机密的名称。必须位于同一命名空间中。
   - * spec.providerType*:_(必需)_用于确定提供备份的内容；例如、NetApp ONTAP S3或Microsoft Azure。

     YAML示例：

```yaml
apiVersion: astra.netapp.io/v1
kind: AppVault
metadata:
  name: astra-appvault
spec:
  providerType: generic-s3
  providerConfig:
    path: testpath
    endpoint: 192.168.1.100:80
    bucketName: bucket1
    secure: "false"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        name: s3-creds
        key: accessKeyID
    secretAccessKey:
      valueFromSecret:
        name: s3-creds
        key: secretAccessKey
```

3. 在您填充之后 `astra-appvault.yaml` 使用正确值的文件、应用CR：

```
kubectl apply -f astra-appvault.yaml -n astra-connector
```

> 添加存储分段时、Astra Control会使用默认存储分段指示符标记一个存储分段。您创建的第一个存储分段将成为默认存储分段。添加分段时、您可以稍后决定添加 "设置另一个默认存储分段"。

了解更多信息

- "使用 Astra Control API"

监控正在运行的任务

您可以在Astra Control中查看有关过去24小时内已完成、失败或已取消的正在运行的任务和任务的详细信息。例如、您可以查看正在运行的备份、还原或克隆操作的状态、并查看完成百分比和估计剩余时间等详细信息。您可以查看已运行的已计划操作或手动启动的操作的状态。

查看正在运行或已完成的任务时、您可以展开任务详细信息以查看每个子任务的状态。对于正在进行的或已完成的任务、任务进度条为绿色、对于已取消的任务、任务进度条为蓝色、对于因错误而失败的任务、任务进度条为红色。

> 对于克隆操作、任务子任务由快照和快照还原操作组成。

要查看有关失败任务的详细信息、请参见 "监控帐户活动"。

步骤

1. 在任务运行期间、转到*应用程序*。
2. 从列表中选择应用程序的名称。
3. 在应用程序的详细信息中、选择*任务*选项卡。

   您可以查看当前或过去任务的详细信息、并按任务状态进行筛选。

> 任务将在*任务*列表中保留长达24小时。您可以使用配置此限制以及其他任务监控器设置 "Astra Control API"。

管理您的帐户

设置计费

您可以使用多种方法管理Astra Control Service帐户计费。如果您使用的是Azure或Amazon AWS、则可以通过Microsoft Azure Marketplace或AWS Marketplace订阅Astra Control服务计划。执行此操作时、您可以通过Marketplace管理您的计费详细信息。或者、您也可以直接向NetApp订阅。如果您直接向NetApp订阅、则可以通过Astra Control Service管理您的计费详细信息。如果您在未订阅的情况下使用Astra控制服务、则系统会自动订阅免费计划。

您可以通过Astra Control Service Free Plan管理帐户中多达10个命名空间。如果您要管理10个以上的命名空间、则需要通过从免费计划升级到高级计划来设置计费、或者通过Azure Marketplace或AWS Marketplace进行订阅。

计费概述

使用 Astra Control Service 会产生两种成本：由 NetApp 为 Astra Control Service 收取费用，由云提供商为永久性卷和对象存储收取费用。

Astra Control Service 计费

Astra 控制服务提供三个计划：

免费计划
: 免费管理多达10个命名空间。

高级 **PayGo**
: 按特定速率管理每个命名空间的数量不限。

高级订阅
: 按年订阅的折扣价预付费用、使您能够管理每个_命名 空间包_最多20个命名空间。请联系NetApp销售部

门、根据您的组织需要购买任意数量的软件包。例如、购买3个软件包、可从Asta Control Service管理60个空间。如果您管理的命名空间超过年度订阅所允许的数量、则会按每个额外命名空间的订阅相关超额使用率向您收取费用。如果您还没有 Astra Control 帐户，则购买高级订阅会自动为您创建一个 Astra Control 帐户。如果您已有免费计划，则会自动转换为高级订阅。

创建Astra Control帐户时、您会自动订阅免费计划。Astra Control的信息板显示了您当前在10个可用命名空间中管理的命名空间数量。在管理包含命名空间的第一个应用程序时、开始为某个命名空间计费；在管理包含此命名空间的最后一个应用程序时、该命名空间的计费将停止。

如果您尝试管理第11个命名空间、则Astra Control会通知您已达到免费计划的限制。然后，系统会提示您从免费计划升级到高级计划。您将按每个额外命名空间的订阅相关超额使用率付费。

您可以随时升级到高级计划。升级后、Astra Control开始为帐户中的_all_命名空间收费。前10个命名空间不会保留在免费计划中。

Google Cloud 计费

永久性卷由NetApp Cloud Volumes Service提供支持、应用程序的备份存储在Google云存储分段中。

- "查看 Cloud Volumes Service 的定价详细信息"。

  请注意， Astra 控制服务支持所有服务类型和服务级别。您使用的服务类型取决于 "Google Cloud 地区"。

- "查看 Google Cloud 存储分段的定价详细信息"。

Microsoft Azure 计费

永久性卷由Azure NetApp Files提供支持、应用程序的备份存储在Azure Blb容器中。

- "查看 Azure NetApp Files 的定价详细信息"。
- "查看 Microsoft Azure Blob 存储的定价详细信息"。
- "在Azure Marketplace中查看A作用 力控制服务计划和定价"

> Astra Control Service的Azure计费率为每小时一次、在使用时间超过29分钟后开始新的计费时间。

Amazon Web Services计费

永久性卷由EBS或FSx for NetApp ONTAP提供支持、应用程序的备份存储在AWS存储分段中。

- "查看Amazon Web Services的定价详细信息"。

在Azure Marketplace中订阅Astra Control服务

您可以使用Azure Marketplace订阅Astra控制服务。您的帐户和计费详细信息通过Marketplace进行管理。

> 要观看Azure Marketplace订阅流程的视频演练、请访问 "NetApp TV"。

步骤

1. 转至 "Azure Marketplace"。

2. 选择*立即获取*。
3. 按照说明订阅计划。

在AWS Marketplace中订阅Astra Control服务

您可以使用AWS Marketplace订阅Astra控制服务。您的帐户和计费详细信息通过Marketplace进行管理。

步骤

1. 转至 "AWS Marketplace"。
2. 选择*查看购买选项*。
3. 如果系统提示您这样做、请登录到AWS帐户或创建新帐户。
4. 按照说明订阅计划。

直接向NetApp订阅Astra控制服务

您可以在Astra控制服务UI中订阅Astra控制服务、也可以联系NetApp销售部门。

从免费计划升级到高级 PayGo 计划

随时升级您的计费计划、通过按需购买从Astra Control开始管理10多个命名空间。您只需要一张有效的信用卡即可。

步骤

1. 选择 * 帐户 * ，然后选择 * 账单 * 。
2. 在 * 计划 * 下，转到 * 高级 PayGo* 并选择 * 立即升级 * 。
3. 提供有效信用卡的付款详细信息，然后选择 * 升级到高级计划 * 。

> 如果信用卡即将到期， Astra Control 将通过电子邮件向您发送电子邮件。

结果

现在、您可以管理10个以上的命名空间。Astra Control开始为您当前管理的_all_命名空间收费。

从免费计划升级到高级订阅

请联系 NetApp 销售部门，以折扣价按年订阅预付费用。

步骤

1. 选择 * 帐户 * ，然后选择 * 账单 * 。
2. 在 * 计划 * 下，转到 * 高级订阅 * 并选择 * 联系销售人员 * 。
3. 向销售团队提供详细信息以启动此流程。

结果

NetApp 销售代表将与您联系以处理您的采购订单。订单完成后、Astra Control将在*计费*选项卡上反映您的当前计划。

查看当前成本和计费历史记录

Astra Control可按命名空间显示当前的每月成本以及详细的计费历史记录。如果您通过Marketplace订阅了计划、则不会显示计费历史记录(但您可以通过登录到Marketplace来查看此历史记录。)

步骤

1. 选择 * 帐户 * ，然后选择 * 账单 * 。

   您的当前成本将显示在计费概述下。

2. 要按命名空间查看计费历史记录、请选择*计费历史记录*。

   Astra Control可为您显示每个命名空间的使用分钟数和成本。使用分钟数是Astra Control在计费期间管理您的命名空间的分钟数。

3. 选择下拉列表以选择上个月。

更改 **Premium PayGo** 的信用卡

如果需要，您可以更改 Astra Control 已记录的用于计费的信用卡。

步骤

1. 选择 * 帐户 > 计费 > 付款方式 * 。
2. 选择配置图标。
3. 修改信用卡。

重要注意事项

- 您的计费计划按 Astra Control 帐户制定。

  如果您有多个帐户，则每个帐户都有自己的计费计划。

- 您的Astra Control费用包括命名空间管理费用。您的云提供商会单独为永久性卷的存储后端付费。

  "了解有关 Astra Control 定价的更多信息"。

- 每个计费周期都在一个月的最后一天结束。
- 您不能从高级版计划降级到免费版计划。

邀请和删除用户

邀请用户加入您的 Astra Control 帐户，并删除不应再访问此帐户的用户。

邀请用户

帐户所有者和管理员可以邀请其他用户加入 Astra Control 帐户。

步骤

1. 确保用户具有 "BlueXP登录"。

2. 选择 * 帐户 * 。
3. 在 * 用户 * 选项卡中，选择 * 邀请 * 。
4. 输入用户的名称，电子邮件地址及其角色。

   请注意以下事项：

   - 电子邮件地址必须与用户用于注册BlueXP的电子邮件地址匹配。
   - 每个角色都提供以下权限：
     - * 所有者 * 具有管理员权限，可以删除帐户。
     - * 管理员 * 具有成员权限，可以邀请其他用户。
     - * 成员 * 可以全面管理应用程序和集群。
     - * 查看器 * 可以查看资源。
5. 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"。

6. 要邀请其他用户，请选择 * 添加其他用户 * 并输入新用户的信息。

   一次最多可以邀请 10 个用户。您可以在 * 邀请用户 * 对话框左侧邀请的用户之间导航。

7. 选择 * 邀请用户 * 。

结果

用户将收到一封电子邮件，邀请他们加入您的帐户。

更改用户的角色

帐户所有者可以更改所有用户的角色，而帐户管理员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. 选择 * 帐户 * 。
2. 在 * 用户 * 选项卡的 * 操作 * 列中为用户选择菜单。
3. 选择 * 编辑角色 * 。
4. 选择一个新角色。
5. 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"。

6. 选择 * 确认 * 。

结果

Astra Control 会根据您选择的新角色更新用户的权限。

删除用户

具有所有者角色的用户可以随时从此帐户中删除其他用户。

步骤

1. 选择 * 帐户 * 。
2. 在 * 用户 * 选项卡中，选择要删除的用户。
3. 在 * 操作 * 列中选择菜单，然后选择 * 删除用户 * 。
4. 出现提示时，键入 "remove" 确认删除，然后选择 * 是，删除用户 * 。

结果

Astra Control 会将用户从帐户中删除。

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"

添加和删除凭据

随时在您的帐户中添加和删除云提供商凭据。Astra Control 使用这些凭据来发现 Kubernetes 集群，集群上的应用程序，并代表您配置资源。

请注意， Astra Control 中的所有用户都共享相同的凭据集。

添加凭据

向 Astra Control 添加凭据的最常见方法是管理集群，但您也可以从帐户页面添加凭据。然后，在管理其他 Kubernetes 集群时，您可以选择这些凭据。

开始之前

- 对于Amazon Web Services、您应具有用于创建集群的IAM帐户凭据的JSON输出。 "了解如何设置IAM用户"。
- 对于 GKE- ，您应该拥有具有所需权限的服务帐户的服务帐户密钥文件。 "了解如何设置服务帐户"。
- 对于 AKS ，您应具有包含创建服务主体时 Azure 命令行界面输出的 JSON 文件。 "了解如何设置服务主体"。

  如果未将 Azure 订阅 ID 添加到 JSON 文件中，您也需要此 ID 。

步骤

1. 选择 * 帐户 > 凭据 * 。
2. 选择 * 添加凭据 * 。

3. 选择* Microsoft Azure*。
4. 选择* Google Cloud Platform*。
5. 选择* Amazon Web Services*。
6. 在 Astra Control 中输入凭据名称，以便将其与其他凭据区分开。
7. 提供所需的凭据。
8. * Microsoft Azure* ：通过上传 JSON 文件或从剪贴板粘贴 JSON 文件的内容，为 Astra Control 提供有关 Azure 服务主体的详细信息。

   JSON 文件应包含创建服务主体时 Azure 命令行界面的输出。它还可以包括您的订阅 ID ，以便自动添加到 Astra Control 。否则，您需要在提供 JSON 后手动输入 ID 。
9. * Google Cloud Platform* ：通过上传文件或粘贴剪贴板中的内容来提供 Google Cloud 服务帐户密钥文件。
10. * Amazon Web Services*：通过上传文件或粘贴剪贴板中的内容来提供Amazon Web Services IAM用户凭据。
11. 选择 * 添加凭据 * 。

结果

现在，您可以在向 Astra Control 添加集群时选择这些凭据。

删除凭据

随时从帐户中删除凭据。您只能在之后删除凭据 "取消管理所有集群"、除非您正在轮换凭据(请参见 [轮换凭据] ）。

> 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 > 凭据 * 。
2. 在 * 状态 * 列中选择要删除的凭据的下拉列表。
3. 选择 * 删除 * 。
4. 键入凭据名称以确认删除，然后选择 * 是，删除凭据 * 。

结果

Astra Control 会从帐户中删除凭据。

轮换凭据

您可以轮换帐户中的凭据。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。

步骤

1. 按照中的步骤删除现有凭据 [删除凭据]。
2. 按照中的步骤添加新凭据 [添加凭据]。

3. 更新所有分段以使用新凭据：
    a. 从左侧导航栏中、选择*分段*。
    b. 在 * 操作 * 列中选择要编辑的存储分段的下拉列表。
    c. 选择 * 编辑 * 。
    d. 在*选择凭据*部分中、选择添加到Astra Control的新凭据。
    e. 选择 * 更新 * 。
    f. 对系统上的任何剩余存储分段重复步骤* b*到* e*。

结果

Astra Control开始使用新的云提供商凭据。

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。

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 * 。

查看和管理通知

操作完成或失败时， Astra Control 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。

未读通知的数量显示在界面右上角。

您可以查看这些通知并将其标记为已读（如果您希望像我们一样清除未读通知，则可以使用此功能）。

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 * 。

关闭您的帐户

如果您不再需要 Astra Control 帐户，可以随时关闭该帐户。

> 关闭帐户后， Astra Control 自动创建的分段将自动删除。

步骤

1. "取消管理所有应用程序和集群"。
2. "从 Astra Control 中删除凭据"。
3. 选择 * 帐户 > 计费 > 付款方式 * 。
4. 选择 * 关闭帐户 * 。
5. 输入您的帐户名称并确认关闭该帐户。

管理云实例

云实例是云提供商中的一个唯一域。您可以为每个云提供商创建多个云实例、每个云实例都有自己的名称、凭据和关联集群。

在向Astra Control添加新集群时、您可以创建一个云实例。您可以使用Astra Control UI编辑云实例以更改其名称或默认存储分段、并使用Astra Control API对云实例执行其他操作。

添加云实例

在向Astra Control添加新集群时、您可以添加新的云实例。请参见 "从 Astra Control Service 开始管理 Kubernetes 集群" 有关详细信息 …

编辑云实例

您可以修改云提供商的现有云实例。

步骤

1. 转至*云实例*。
2. 在云实例列表中、选择要编辑的云实例的*操作*菜单。
3. 选择 * 编辑 * 。

   在此页面上、您可以更新云实例的名称和默认存储分段。

   > Astra Control中的每个云实例都必须具有唯一的名称。

轮换云实例的凭据

您可以使用Astra Control API轮换云实例的凭据。了解更多信息。 "转到 Astra 自动化文档"。

删除云实例

您可以使用Astra Control API从云提供商中删除云实例。了解更多信息。 "转到 Astra 自动化文档"。

启用Asta Control配置程序

Astra 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Asta三端CSI的功能之外、Asta 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Asta Control用户启用Asta Control配置程序。Asta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。

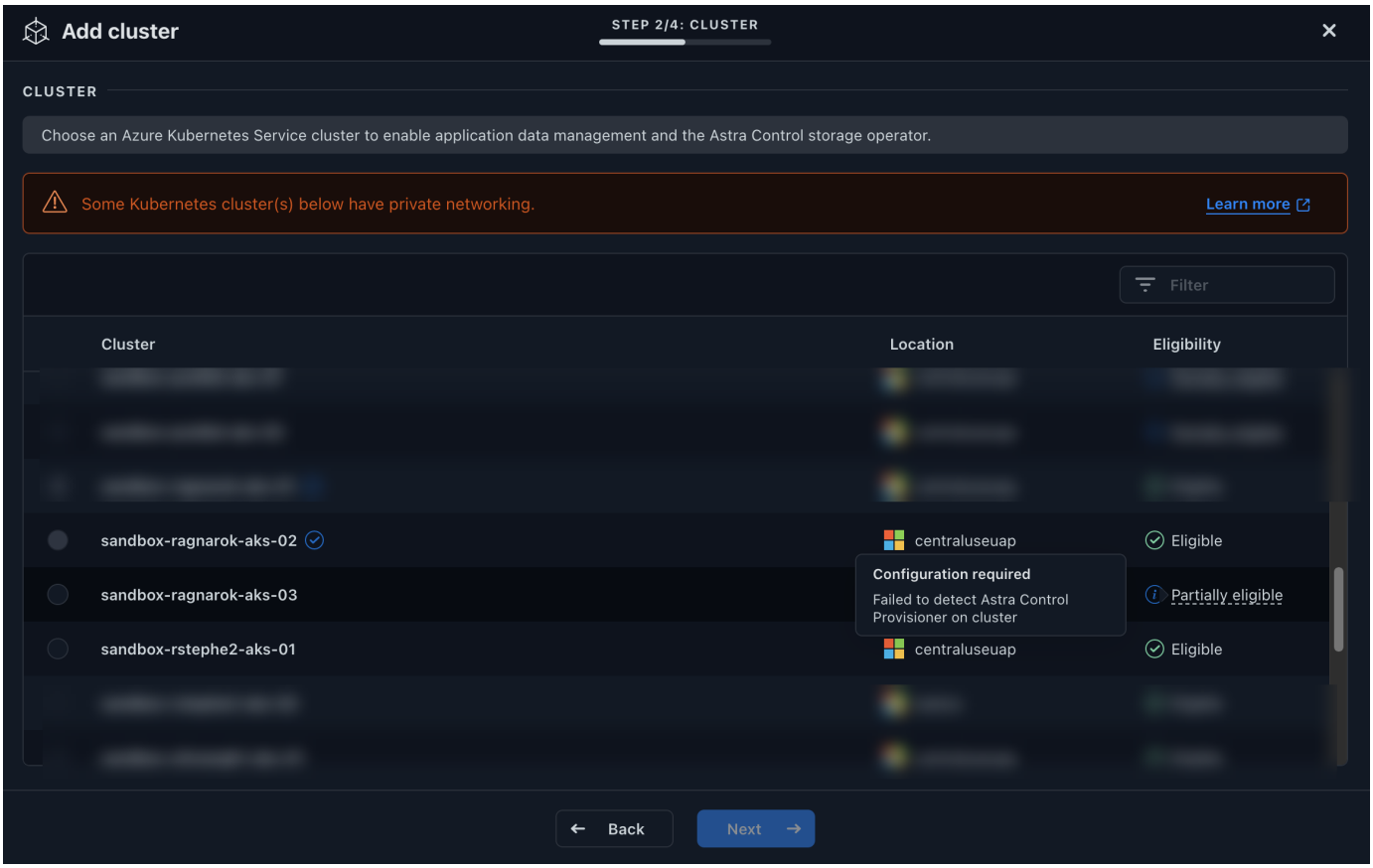
如何知道是否需要启用**Astra Control**配置程序?

如果您向Astra Control Service添加的集群之前未安装Astra三项功能、则此集群将标记为 `Eligible`。你先请 "将集群添加到Asta Control"，A作用 控制配置程序将自动启用。

集群未标记 `Eligible`，将被标记 `Partially eligible` 原因之一：

- 它使用的是旧版本的Asta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

适用于 `Partially eligible` 在某些情况下、请按照以下说明手动为集群启用Asta 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Asta Control配置程序、但要启用Asta 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：
   a. 登录到Astra Control Service UI并记录您的Astra Control帐户ID。
      i. 选择页面右上角的图图标。
      ii. 选择* API访问*。
      iii. 记下您的帐户ID。
   b. 在同一页面中，选择*Generate API t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。
   c. 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、请按照中的三端修复操作符步骤进行操作 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 Trdent安装方法是否使用。
4. 使用您最初使用的安装方法在Asta Trdent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"。

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. 创建三项命名空间 (`kubectl create namespace trident`)或确认三项命名空间仍然存在（`kubectl get all -n trident`）。如果已删除此命名空间、请重新创建它。

c. 将Astra Trdent更新到24.02.0：

> 对于运行Kubornetes 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Astra三端映像设置自定义注册表位置或从Astra Control注册表中提取该映像（`tridentImage: <my_custom_registry>/trident:24.02.0` 或 `tridentImage: netapp/trident:24.02.0`）。

ii. 启用Asta Control配置程序（`enableACP: true`）。

iii. 设置Asta Control配置程序映像的自定义注册表位置或将其从Asta Control注册表中提取（`acpImage: <my_custom_registry>/trident-acp:24.02.0` 或 `acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0`）。

iv. 如果您已建立 图像拉取密钥 在本操作步骤的前面部分、您可以在此处设置它们（`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` 容器正在运行 `acpVersion` 为 `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

a. "下载Asta三端安装程序并解压缩"。

b. "如果您已有Asta Trident、请从托管它的集群中将其卸载"。

c. 在启用Asta Control配置程序的情况下安装Asta Trent (`--enable-acp=true`）：

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

d. 确认已启用Asta 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

a. 如果您安装了Astra Trident 23.07.1或更早版本、 "卸载" 操作员和其他组件。

b. 如果您的Kubornetes集群运行的是1.24或更早版本、请删除PSP：

```
kubectl delete psp tridentoperatorpod
```

c. 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` 并运行以下选项之一、其中包括这些设置：

- 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、则为 " {\f270 {\f151 {\f270} " 可以位于一个注册表或不同的注册表中、但所有CSI映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：

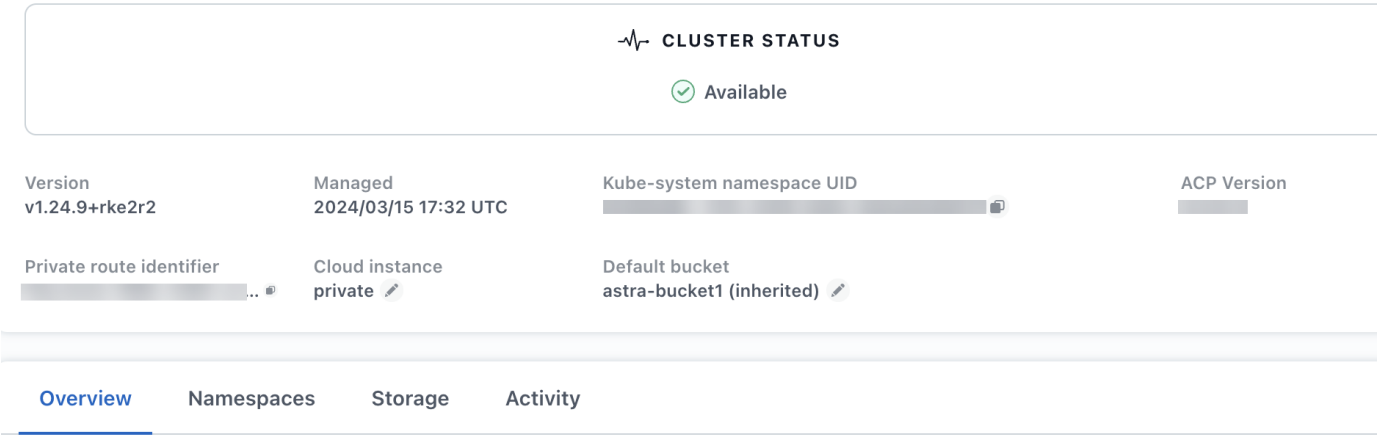
```
./tridentctl uninstall -n trident
```

请勿 "完全删除Asta Trdent CRD" 在尝试重新启用Astra Control配置程序之前、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 `ACP version` 而不是 `Trident version` 字段和当前安装的版本号。



有关详细信息 ...

- "Asta Trdent升级文档"

取消管理应用程序和集群

从 Astra Control 中删除不再需要管理的任何应用程序或集群。

停止管理应用程序

停止管理不再需要从 Astra Control 备份，创建快照或克隆的应用程序。

取消管理应用程序时：

- 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. 选择应用程序。
3. 从选项菜单的操作列中、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 * 。

结果

Astra Control 停止管理应用程序。

停止管理集群

停止从 Astra Control 管理您不再需要管理的集群。

> 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。

作为最佳实践，我们建议您先从 Astra Control 中删除集群，然后再通过 GCP 将其删除。

取消管理集群时：

- 此操作将停止由 Astra Control 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。
- Astra Control配置程序或Astra三端存储不会从集群中卸载。 "了解如何卸载 Astra Trident"。

步骤

1. 选择 * 集群 * 。
2. 选中不再需要管理的集群对应的复选框。
3. 从“操作”列的选项菜单中，选择“取消管理”。
4. 确认要取消管理此集群、然后选择*是、取消管理*。

结果

集群状态将更改为*正在删除*。之后，该群集将从*Clusters页中删除，并且不再由Astra Control管理。

从云提供商中删除集群

在删除持久性卷（ PV ）驻留在 NetApp 存储类上的 Kubernetes 集群之前，您需要先按照以下方法之一删除永久性卷声明（ PVC ）。在删除集群之前删除PVC和PV可确保您不会收到云提供商的意外账单。

- * 方法 1*： 从集群中删除应用程序工作负载命名空间。请勿删除 Trident 命名空间。
- * 方法 2*： 删除 PVC 和 Pod ，或者删除挂载 PV 的部署。

当您从Astra Control管理Kubernetes集群时、该集群上的应用程序会使用您的云提供商作为永久性卷的存储后端。如果从云提供商中删除集群而未先删除 PV ，则后端卷将与集群一起被删除。

使用上述方法之一将从集群中删除相应的 PV 。在删除集群之前，请确保集群上的 NetApp 存储类中不存在任何 PV 。

如果在删除集群之前未删除永久性卷、则需要手动从云提供程序中删除后端卷。

部署Asta Control的自管理实例

如果您需要位于网络内部的Asta Control自管理实例、则可以直接从Asta Control Service部署Asta Control Center。

步骤

1. 在仪表板的Getting Started区域中，选择*Deploy a self-managed instance*。
2. 执行以下操作之一：
   - 通过选择*Gener成*生成新的API令牌。
   - 粘贴现有A作用 力控制REST API令牌。请参见 "Astra Automation文档" 有关生成API令牌的指导。
3. 按照*Deploy Astra Control Center*窗口中的说明进行操作。

使用Astra Control配置程序

配置存储后端加密

通过使用Astra Control配置程序、您可以对受管集群和存储后端之间的流量启用加密、从而提高数据访问安全性。

Astra Control配置程序支持对两种类型的存储后端进行Kerberos加密：

- 内部部署**Kubernet**—控制配置程序支持通过从Red ONTAP OpenShift和上游Kubernetes集群到内部ONTAP卷的NFS3和NFSv4连接进行Kerberos加密。
- **NFSv**-控件配置程序支持通过从上游Azure NetApp Files集群到Azure NetApp Files卷的NFSv4.1连接进行Kerberos加密。

您可以创建、删除、调整大小、创建快照、克隆、 只读克隆、并导入使用NFS加密的卷。

为内部**ONTAP**卷配置传输中的**Kerberos**加密

您可以对受管集群与内部ONTAP存储后端之间的存储流量启用Kerberos加密。

> 仅支持使用对使用内部ONTAP存储后端的NFS流量进行Kerberos加密 `ontap-nas` 存储驱动程序。

开始之前

- 确保您已安装 "已启用Asta Control配置程序" 在受管集群上。
- 确保您可以访问 `tridentctl` 实用程序。
- 确保您对ONTAP存储后端具有管理员访问权限。
- 确保您知道要从ONTAP存储后端共享的一个或多个卷的名称。
- 确保已准备好ONTAP Storage VM以支持NFS卷的Kerberos加密。请参见 "在数据 LIF 上启用 Kerberos" 有关说明，请参见。
- 确保已正确配置使用Kerberos加密的任何NFSv4卷。请参阅的NetApp NFSv4域配置一节(第13页) "《NetApp NFSv4增强功能和最佳实践指南》"。

添加或修改**ONTAP**导出策略

您需要向现有ONTAP导出策略添加规则、或者创建新的导出策略、以便对ONTAP Storage VM根卷以及与上游Kubornetes集群共享的任何ONTAP卷支持Kerberos加密。您添加的导出策略规则或创建的新导出策略需要支持以下访问协议和访问权限：

访问协议

使用NFS、NFSv3和NFSv4访问协议配置导出策略。

访问详细信息

您可以根据卷的需求配置以下三种不同版本的Kerberos加密之一：

- **Kerberos 5**-(身份验证和加密)
- **Kerberos 5i**-(身份验证和加密与身份保护)
- **Kerberos 5p**-(身份验证和加密、具有身份和隐私保护功能)

使用适当的访问权限配置ONTAP导出策略规则。例如、如果集群要挂载混合使用Kerberos 5i和Kerberos 5p加密的NFS卷、请使用以下访问设置：

Type	只读访问	读/写访问	超级用户访问
"unix"	enabled	enabled	enabled
Kerberos 5i	enabled	enabled	enabled
Kerberos 5p	enabled	enabled	enabled

有关如何创建ONTAP导出策略和导出策略规则、请参见以下文档：

- "创建导出策略"
- "向导出策略添加规则"

创建存储后端

您可以创建包含Kerberos加密功能的A作用 力控制配置程序存储后端配置。

关于此任务

在创建用于配置Kerberos加密的存储后端配置文件时、您可以使用指定三个不同版本的Kerberos加密之一 `spec.nfsMountOptions` 参数：

- `spec.nfsMountOptions: sec=krb5` (身份验证和加密)
- `spec.nfsMountOptions: sec=krb5i` (身份验证和加密以及身份保护)
- `spec.nfsMountOptions: sec=krb5p` (身份验证和加密以及身份和隐私保护)

请仅指定一个Kerberos级别。如果在参数列表中指定多个Kerberos加密级别、则仅会使用第一个选项。

步骤

1. 在受管集群上、使用以下示例创建存储后端配置文件。将括号<>中的值替换为您环境中的信息：

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-ontap-nas-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  storageDriverName: "ontap-nas"
  managementLIF: <STORAGE_VM_MGMT_LIF_IP_ADDRESS>
  dataLIF: <PROTOCOL_LIF_FQDN_OR_IP_ADDRESS>
  svm: <STORAGE_VM_NAME>
  username: <STORAGE_VM_USERNAME_CREDENTIAL>
  password: <STORAGE_VM_PASSWORD_CREDENTIAL>
  nasType: nfs
  nfsMountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
  qtreesPerFlexvol:
  credentials:
    name: backend-ontap-nas-secret
```

2. 使用您在上一步中创建的配置文件创建后端：

```
tridentctl create backend -f <backend-configuration-file>
```

如果后端创建失败，则后端配置出现问题。您可以运行以下命令来查看日志以确定发生原因：

```
tridentctl logs
```

确定并更正配置文件中的问题后，您可以再次运行 create 命令。

创建存储类。

您可以创建存储类来配置采用Kerberos加密的卷。

关于此任务

创建存储类对象时、您可以使用指定三个不同版本的Kerberos加密之一 `mountOptions` 参数：

- `mountOptions: sec=krb5` (身份验证和加密)
- `mountOptions: sec=krb5i` (身份验证和加密以及身份保护)
- `mountOptions: sec=krb5p` (身份验证和加密以及身份和隐私保护)

请仅指定一个Kerberos级别。如果在参数列表中指定多个Kerberos加密级别、则仅会使用第一个选项。如果您在存储后端配置中指定的加密级别与您在存储类对象中指定的加密级别不同、则存储类对象优先。

步骤

1. 使用以下示例创建StorageClass Kubenetes对象：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-sc
provisioner: csi.trident.netapp.io
mountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
parameters:
  backendType: "ontap-nas"
  storagePools: "ontapnas_pool"
  trident.netapp.io/nasType: "nfs"
allowVolumeExpansion: True
```

2. 创建存储类：

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. 确保已创建存储类：

```
kubectl get sc ontap-nas-sc
```

   您应看到类似于以下内容的输出：

```
NAME            PROVISIONER             AGE
ontap-nas-sc    csi.trident.netapp.io   15h
```

配置卷

创建存储后端和存储类后、您现在可以配置卷。请参阅以下说明 "配置卷"。

为Azure NetApp Files卷配置传输中的Kerberos加密

您可以对受管集群与单个Azure NetApp Files存储后端或Azure NetApp Files存储后端虚拟池之间的存储流量启用Kerberos加密。

开始之前

- 确保已在受管Red Hat OpenShift集群上启用Asta Control配置程序。请参见 "启用Asta Control配置程序" 有关说明，请参见。
- 确保您可以访问 `tridentctl` 实用程序。
- 请注意中的要求并按照中的说明、确保您已为Kerberos加密准备好Azure NetApp Files存储后端 "Azure NetApp Files 文档"。
- 确保已正确配置使用Kerberos加密的任何NFSv4卷。请参阅的NetApp NFSv4域配置一节(第13页) "《NetApp NFSv4增强功能和最佳实践指南》"。

创建存储后端

您可以创建包含Kerberos加密功能的Azure NetApp Files存储后端配置。

关于此任务

在创建配置Kerberos加密的存储后端配置文件时、您可以对其进行定义、使其应用于以下两个可能的级别之一：

- 使用的*存储后端级别* `spec.kerberos` 字段
- 使用的*虚拟池级别* `spec.storage.kerberos` 字段

在虚拟池级别定义配置时、系统会使用存储类中的标签来选择该池。

在任一级别、您都可以指定以下三种不同版本的Kerberos加密之一：

- `kerberos: sec=krb5` (身份验证和加密)
- `kerberos: sec=krb5i` (身份验证和加密以及身份保护)
- `kerberos: sec=krb5p` (身份验证和加密以及身份和隐私保护)

步骤

1. 在受管集群上、根据需要定义存储后端的位置(存储后端级别或虚拟池级别)、使用以下示例之一创建存储后端配置文件。将括号<>中的值替换为您环境中的信息：

存储后端级别示例

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-anf-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-anf-secret
```

虚拟池级别示例

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-anf-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  storage:
    - labels:
        type: encryption
      kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-anf-secret
```

2. 使用您在上一步中创建的配置文件创建后端：

```
tridentctl create backend -f <backend-configuration-file>
```

如果后端创建失败，则后端配置出现问题。您可以运行以下命令来查看日志以确定发生原因：

```
tridentctl logs
```

确定并更正配置文件中的问题后，您可以再次运行 create 命令。

创建存储类。

您可以创建存储类来配置采用Kerberos加密的卷。

步骤

1. 使用以下示例创建StorageClass Kubenetes对象：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "nfs"
  selector: "type=encryption"
```

2. 创建存储类：

```
kubectl create -f sample-input/storage-class-anf-sc-nfs.yaml
```

3. 确保已创建存储类：

```
kubectl get sc anf-sc-nfs
```

您应看到类似于以下内容的输出：

```
NAME          PROVISIONER              AGE
anf-sc-nfs     csi.trident.netapp.io    15h
```

配置卷

创建存储后端和存储类后、您现在可以配置卷。请参阅以下说明 "配置卷"。

使用快照恢复卷数据

Asta Control配置程序可使用从快照快速原位还原卷 `TridentActionSnapshotRestore` (TSR) CR。此CR用作要务Kubbernetes操作、在操作完成后不会持久保留。

Asta Control配置程序支持在上执行快照还原 `ontap-san`， `ontap-san-economy`， `ontap-nas`， `ontap-`

`nas-flexgroup`， `azure-netapp-files`， `gcp-cvs`，和 `solidfire-san` 驱动程序。

开始之前

您必须具有绑定的PVC和可用的卷快照。

- 验证PVC状态是否已绑定。

```
kubectl get pvc
```

- 确认卷快照已准备就绪、可以使用。

```
kubectl get vs
```

步骤

1. 创建TSR CR。此示例将为PVC创建CR `pvc1` 和卷快照 `pvc1-snapshot`。

```
cat tasr-pvc1-snapshot.yaml

apiVersion: v1
kind: TridentActionSnapshotRestore
metadata:
  name: this-doesnt-matter
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. 应用CR以从快照还原。此示例将从Snapshot还原 `pvc1`。

```
kubectl create -f tasr-pvc1-snapshot.yaml

tridentactionsnapshotrestore.trident.netapp.io/this-doesnt-matter
created
```

结果

Asta Control配置程序从快照还原数据。您可以验证快照还原状态。

```
kubectl get tasr -o yaml

apiVersion: v1
items:
- apiVersion: trident.netapp.io/v1
  kind: TridentActionSnapshotRestore
  metadata:
    creationTimestamp: "2023-04-14T00:20:33Z"
    generation: 3
    name: this-doesnt-matter
    namespace: trident
    resourceVersion: "3453847"
    uid: <uid>
  spec:
    pvcName: pvc1
    volumeSnapshotName: pvc1-snapshot
  status:
    startTime: "2023-04-14T00:20:34Z"
    completionTime: "2023-04-14T00:20:37Z"
    state: Succeeded
kind: List
metadata:
  resourceVersion: ""
```

- 在大多数情况下、如果发生故障、Asta Control配置程序不会自动重试此操作。您需要再次执行此操作。
- 没有管理员访问权限的Kubbernetes用户可能必须获得管理员授予的权限、才能在其应用程序命名空间中创建TSR CR。

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。请参见 "ONTAP 中的SnapMirror许可概述" 有关详细信息 …

对等

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

> 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Kubbernetes集群上执行以下步骤：
   a. 使用创建StorageClass对象 `trident.netapp.io/replication: true` 参数。

      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TridentMirorRelationship 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Kubbernetes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TridentMirorRelationship 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Kubbernetes集群上执行以下步骤：

- 将TridentMirorRelationship的_spec.state_字 段更新为 `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Kubbernetes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_TridentMirorRelationship_ 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TundentMirorRelationship上的_spic.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flevxvol。

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**TTr**

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` 用于更新关系的字段。
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。

使用 Astra Control REST API 实现自动化

Astra Control 具有一个 REST API ，可用于使用编程语言或 Curl 等实用程序直接访问 Astra Control 功能。您还可以使用 Ansible 和其他自动化技术管理 Astra Control 部署。

了解更多信息。 "转到 Astra 自动化文档"。

知识和支持

注册以获得支持

在设置帐户时， Astra Control 会尝试自动注册您的帐户以获得支持。如果无法注册，您可以手动注册以获得支持。要获得 NetApp 技术支持的帮助，需要注册支持。

验证您的支持注册

Astra Control 包含一个支持状态字段，您可以通过此字段确认您的支持注册。

步骤

1. 选择 * 支持 * 。
2. 请查看支持状态字段。

   支持状态从 " 未注册 " 开始，但在完成后将变为 " 正在进行 " ，最后变为 " 已注册 " 。

   此支持注册状态每 15 分钟轮询一次。新的NetApp客户可能需要下一个工作日才能完成加盟和支持注册。如果序列号在 48 小时内未显示 " 已注册 " ，您可以使用 astra.feedback@netapp.com 联系 NetApp 或从手动注册 https://register.netapp.com。

获取序列号

注册帐户时， Astra Control 会使用您提供的有关贵公司的信息生成一个 20 位数的 NetApp 序列号，此序列号以 "941" 开头。

NetApp 序列号表示您的 Astra Control 帐户。打开 Web 服务单时，您需要使用此序列号。

您可以从 * 支持 * 页面的 Astra Control 界面中找到序列号。

激活支持授权

如果 Astra Control 无法自动注册您的帐户以获得支持，则必须注册与 Astra Control 关联的 NetApp 序列号才能激活支持授权。我们提供 2 种支持注册选项：

1. 使用现有 NetApp 支持站点（ NSS ） SSO 帐户的当前 NetApp 客户
2. 新的 NetApp 客户，没有现有的 NetApp 支持站点（ NSS ） SSO 帐户

选项 **1** ： 使用现有 **NetApp** 支持站点（ **NSS** ）帐户的当前 **NetApp** 客户

步骤

1. 导航到 "云数据服务支持注册" 页面。
2. 选择 * 我已注册为 NetApp 客户 * 。
3. 输入您的 NetApp 支持站点凭据以登录。

   此时将显示现有客户注册页面。

4. 在表单上填写所需信息：
    a. 输入您的姓名，公司和电子邮件地址。
    b. 选择*Astra Control Service*作为产品线。
    c. 选择计费提供商。
    d. 输入序列号。
    e. 选择 * 提交 * 。

结果

您应重定向到 " 已成功提交注册 " 页面。与您的注册关联的电子邮件地址将在几分钟内收到一封电子邮件，指出 " 您的产品现在有资格获得支持 " 。

此为适用序列号的一次性支持注册。

选项 **2**： 新的 **NetApp** 客户，没有现有的 **NetApp** 支持站点（ **NSS** ）帐户

步骤

1. 导航到 "云数据服务支持注册" 页面。
2. 选择 * 我不是 NetApp 注册客户 * 。

   此时将显示"New Customer Registration"(新客户注册)页面。

3. 在表单上填写所需信息：
    a. 输入您的姓名、公司信息和详细联系信息。
    b. 选择*Astra Control Service*作为产品线。
    c. 选择计费提供商。
    d. 输入序列号。
    e. 输入验证码值。
    f. 选中此复选框以确认您已阅读NetApp隐私政策。
    g. 选择 * 提交 * 。

       您将收到一封来自已提交注册的确认电子邮件。如果未发生错误，系统将重新定向到 " 已成功提交注册 " 页面。您还将在一小时内收到一封电子邮件，指出 " 您的产品现在有资格获得支持 " 。

   此为适用序列号的一次性支持注册。

4. 作为 NetApp 的新客户，您还需要创建一个 NetApp 支持站点（ NSS ）用户帐户，以供将来激活支持以及访问支持门户以进行技术支持聊天和 Web 服务单。

   转至 "NetApp 支持注册站点" 以执行此任务。您可以提供新注册的 Astra Control 序列号来加快此过程。

故障排除

了解如何解决您可能遇到的一些常见问题。

https://kb.netapp.com/Cloud/Astra/Control

有关详细信息 ...

- "故障排除"

获取帮助

NetApp 以多种方式为 Astra Control 提供支持。全天候提供广泛的免费自助支持选项、例如知识库(KB)文章和不和渠道。您的 Astra Control 帐户包括通过 Web 服务单提供的远程技术支持。

您必须先执行此操作 "激活对您的 NetApp 序列号的支持" 以便使用这些非自助服务支持选项。聊天和 Web 服务单以及案例管理需要使用 NetApp 支持站点（ NSS ） SSO 帐户。

您可以从 Astra Control UI 访问支持选项，方法是从主菜单中选择 * 支持 * 选项卡。

自助支持

这些选项全天候免费提供：

- "知识库"

  搜索与 Astra Control 相关的文章，常见问题解答或中断修复信息。

- 文档。

  这是您当前正在查看的文档站点。

- "通过不和获得帮助"

  转到"Pub类别"中的Astra、与同行和专家建立联系。

- 反馈电子邮件

  发送电子邮件至 astra.feedback@netapp.com ，告知我们您的想法，想法或顾虑。

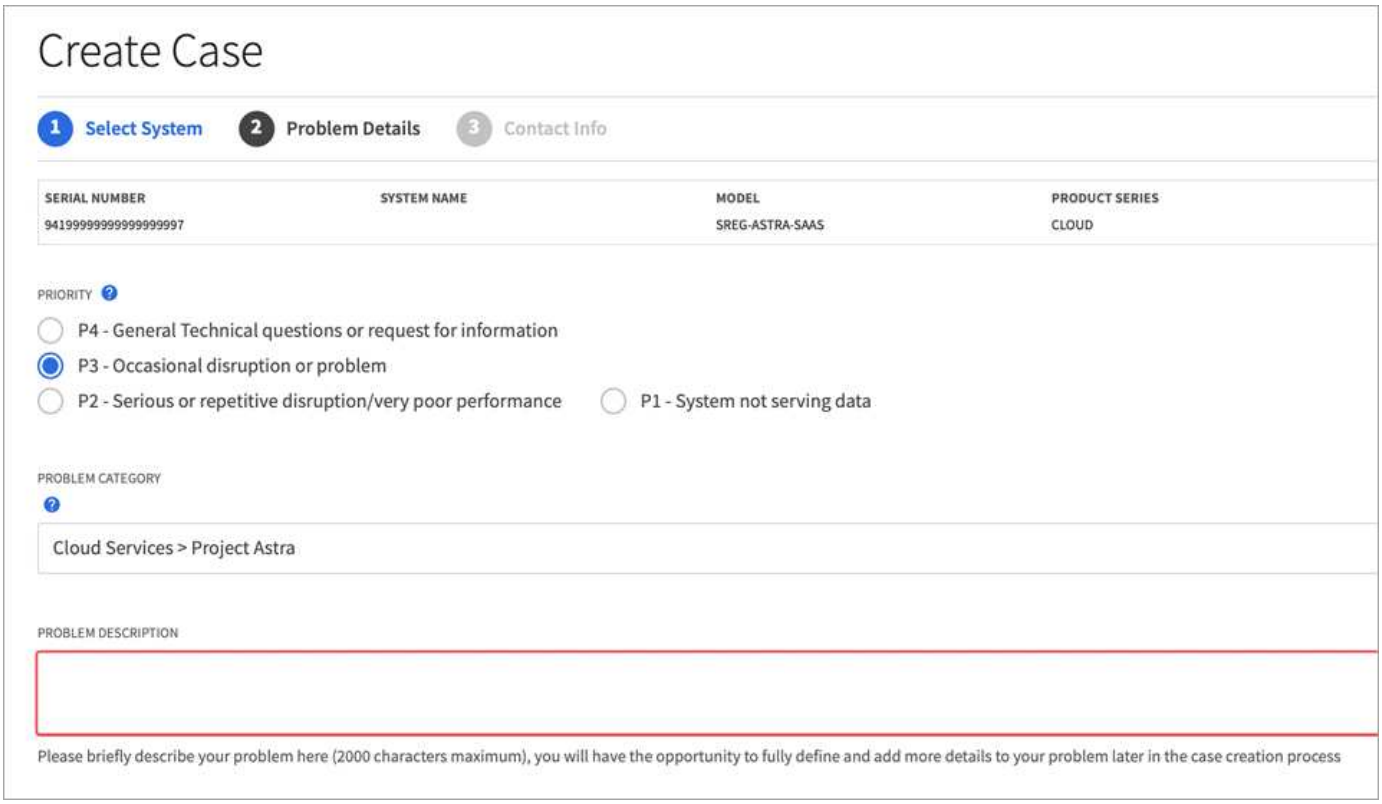
订阅支持

除了上述自助支持选项之外，您还可以与 NetApp 支持工程师一起解决任何问题 "激活对您的 NetApp 序列号的支持"。

激活 Astra Control 序列号后，您可以通过创建来访问 NetApp 技术支持资源 "支持服务单"。

选择 * 云数据服务 > Astra * 。

使用您的 "941" 序列号打开 Web 服务单。 "详细了解您的序列号"。


Create Case
1 Select System
2 Problem Details
3 Contact Info
SERIAL NUMBER
9419999999999999997
SYSTEM NAME
MODEL
SREG-ASTRA-SAAS
PRODUCT SERIES
CLOUD
PRIORITY
P4 - General Technical questions or request for information
P3 - Occasional disruption or problem
P2 - Serious or repetitive disruption/very poor performance
P1 - System not serving data
PROBLEM CATEGORY
Cloud Services > Project Astra
PROBLEM DESCRIPTION
Please briefly describe your problem here (2000 characters maximum), you will have the opportunity to fully define and add more details to your problem later in the case creation process

常见问题解答

如果您只是想快速了解问题解答，此常见问题解答会很有帮助。

概述

Astra Control旨在简化Kubbernetes原生应用程序的应用程序数据生命周期管理操作。Astra控制服务支持在多个云提供商环境中运行的Kubernetes集群。

以下各节将为您在使用 Astra Control 时可能遇到的其他一些问题提供解答。如有任何其他澄清，请联系 astra.feedback@netapp.com

访问 Astra Control

注册**Astra Control**时、为什么需要提供这么多详细信息？

Astra Control 在注册时需要准确的客户信息。要进行全球贸易合规性（ GTC ）检查，必须提供此信息。

为什么在注册**Astra Control**时收到“注册失败”错误？

Astra Control 要求您在入职部分提供准确的客户信息。如果您提供的信息不正确，则会收到 "Registration Failed" 错误。您所属的其他帐户也会被锁定。

什么是**Astra Control Service URL**？

您可以通过访问 Astra 控制服务 https://astra.netapp.io。

我向一位同事发送了一封电子邮件邀请函、但他们尚未收到邀请函。我该怎么办？

要求他们检查其垃圾邮件文件夹中是否有来自 do-not-reply@netapp.com 的电子邮件，或者在收件箱中搜索 " 邀请函 " 。 您也可以删除此用户并尝试重新添加它们。

我从免费版升级到**Premium PayGO**计划。前**10**个名字会收取费用吗？

是的。升级到高级计划后、Astra Control开始为您的帐户中的所有受管命名空间收取费用。

我在一个月中升级到**Premium PayGO**计划。是否会收取整个月的费用？

否从升级到超值计划时开始计费。

我正在使用免费计划、是否会因永久性卷索赔而收费？

是的、您需要为云提供商的集群使用的永久性卷付费。

注册 Kubernetes 集群

在将集群添加到**Astra Control Service**之前、是否需要在集群上安装**CSI**驱动程序？

否将集群添加到Astra Control后、该服务将自动在Kubbernetes集群上安装Astra三端容器存储接口(CSI)驱动程序。此CSI驱动程序用于为云提供商支持的集群配置永久性卷。

在添加到**Astra Control Service**后、我需要向集群添加工作节点。我该怎么办？

可以将新的工作节点添加到现有池中、也可以创建新池、但前提是它们是 `COS_CONTAINERD` 映像类型。这些信息将由 Astra Control 自动发现。如果新节点在 Astra Control 中不可见，请检查新工作节点是否正在运行受支持

的映像类型。您还可以使用验证新工作节点的运行状况 `kubectl get nodes` 命令：

注册Elastic Kubernetes Service (EKS)集群

是否可以将专用**EKS**集群添加到**Astra Control Service**?

可以、您可以将专用EKS集群添加到Astra Control Service。要添加专用EKS集群、请参见 "从 Astra Control Service 开始管理 Kubernetes 集群"。

注册Azure Kubernetes Service (AKS)集群

是否可以将专用**AKS**集群添加到**Astra Control Service**?

可以，您可以将专用 AKS 集群添加到 Astra Control Service 。要添加专用AKS集群、请参见 "从 Astra Control Service 开始管理 Kubernetes 集群"。

是否可以使用**Active Directory**管理**AKS**集群的身份验证?

可以、您可以将AKS集群配置为使用Azure Active Directory (Azure AD)进行身份验证和身份管理。创建集群时、请按照中的说明进行操作 "正式文档" 将集群配置为使用Azure AD。您需要确保集群满足AKS管理的Azure AD集成的要求。

注册Google Kubernetes Engine (GKEE)集群

是否可以将专用**GKE**集群添加到**Astra Control Service**?

可以，您可以将专用 GKE- 集群添加到 Astra Control Service 中。要添加专用GKE集群、请参见 "从 Astra Control Service 开始管理 Kubernetes 集群"。

专用GKE集群必须具有 "授权网络" 设置为允许 Astra Control IP 地址：

52.188.218.166/32

我的**GKE**集群是否可以驻留在共享**VPC**上?

是的。Astra Control可以管理共享VPC中的集群。 "了解如何为共享 VPC 配置设置 Astra 服务帐户"。

在哪里可以找到我在**GCP**上的服务帐户凭据?

登录到后 "Google Cloud Console"，您的服务帐户详细信息将显示在 * IAM 和管理 * 部分中。有关详细信息，请参见 "如何为 Astra Control 设置 Google Cloud"。

我想从不同的**GCP**项目添加不同的**GKE**集群。**Astra Control**是否支持这一点?

不支持，此配置不受支持。仅支持一个 GCP 项目。

删除集群

如何正确取消注册、关闭集群以及删除关联的卷?

1. "从 Astra Control 取消管理应用程序"。
2. "从 Astra Control 中取消注册集群"。
3. "删除永久性卷声明"。

4. 删除集群。

从**Asta Control**中删除集群后、我的应用程序和数据会发生什么情况？

从 Astra Control 中删除集群不会对集群的配置（应用程序和永久性存储）进行任何更改。对该集群上的应用程序执行的任何 Astra Control 快照或备份都将无法还原。存储在存储后端的卷快照数据不会被删除。由 Astra Control 创建的永久性存储备份将保留在云提供商的对象存储中，但无法还原。

> 在通过 GCP 删除集群之前，请始终从 Astra Control 中将其删除。如果在集群仍由 Astra Control 管理时从 GCP 中删除集群，则可能会对您的 Astra Control 帐户产生发生原因问题。

取消管理**Astra Control**配置程序时、它是否会自动从集群中卸载？

从Astra Control Center取消管理集群时、Astra Control配置程序或Astra三项功能不会自动从集群中卸载。要卸载Asta Control配置程序及其组件或Asta Trandent、您需要 "请按照以下步骤卸载包含Astra Control配置程序服务的Astra Trdent实例"。

管理应用程序

Astra Control是否可以部署应用程序？

Astra Control 不会部署应用程序。应用程序必须部署在 Astra Control 之外。

我没有看到应用程序的任何**PVC**绑定到**GCP CVS**。出什么问题了？

在成功添加到 Astra Control 后， Astra Trident 运算符会将默认存储类设置为 `netapp-cvs-perf-Premium` 。如果应用程序的 PVC 未绑定到适用于 Google Cloud 的 Cloud Volumes Service ，您可以执行以下几个步骤：

- 运行 `kubectl get SC` 并检查默认存储类。
- 检查用于部署应用程序的 YAML 文件或 Helm 图表，查看是否定义了其他存储类。
- GKE1.24及更高版本不支持基于Docker的节点映像。检查以确保GKEE中的工作节点映像类型为 `COS_CONTAINERD` NFS挂载成功。

停止从**Asta Control**管理应用程序后、应用程序会发生什么情况？

任何现有备份或快照都将被删除。应用程序和数据始终可用。数据管理操作不适用于非受管应用程序或属于该应用程序的任何备份或快照。

数据管理操作

Astra Control在何处创建对象存储分段？

第一个受管集群的地理位置决定了对象存储的位置。例如，如果您添加的第一个集群位于欧洲区域，则会在同一地理位置创建存储分段。如果需要，您可以 "添加其他存储分段"。

我的帐户中有我未创建的快照。他们来自哪里？

在某些情况下， Astra Control 会在执行其他过程时自动创建快照。如果这些快照的使用时间超过几分钟，您可以安全地将其删除。

我的应用程序使用多个**PV.Astra Control**是否会为所有这些**PVC**创建快照和备份？

是的。Astra Control对应用程序执行的快照操作包括绑定到应用程序PVC的所有PV的快照。

我是否可以直接通过云提供商管理**Asta Control**拍摄的快照？

否Asta Control创建的快照和备份只能使用Asta Control进行管理。

Asta Control配置程序

Astra Control配置程序的存储配置功能与**Astra Trident**中的存储配置功能有何不同?

Asta Control配置程序作为Asta Control的一部分、支持一组超群的存储配置功能、这些功能在开源Asta三元数据中不可用。这些功能是对开放源码的三元数据可用的所有功能的补充。

Astra Control配置程序是否正在取代**Astra Trent**?

Asta Control配置程序已取代Asta Trandent、成为Asta Control架构中的存储配置程序和流程编排程序。Asta Control用户应执行此操作 "启用Asta Control配置程序" 使用A作用 力控制。此版本仍支持Asta三项功能、但未来版本不支持此功能。Asta三元数据将保持开源状态、并使用NetApp的新CSI和其他功能进行发布、维护、支持和更新。但是、只有包含A作用 力三项CSI功能以及扩展存储管理功能的A作用 力控制配置程序才能用于即将推出的A作用 力控制版本。

我是否必须为**Asta**三端存储付费?

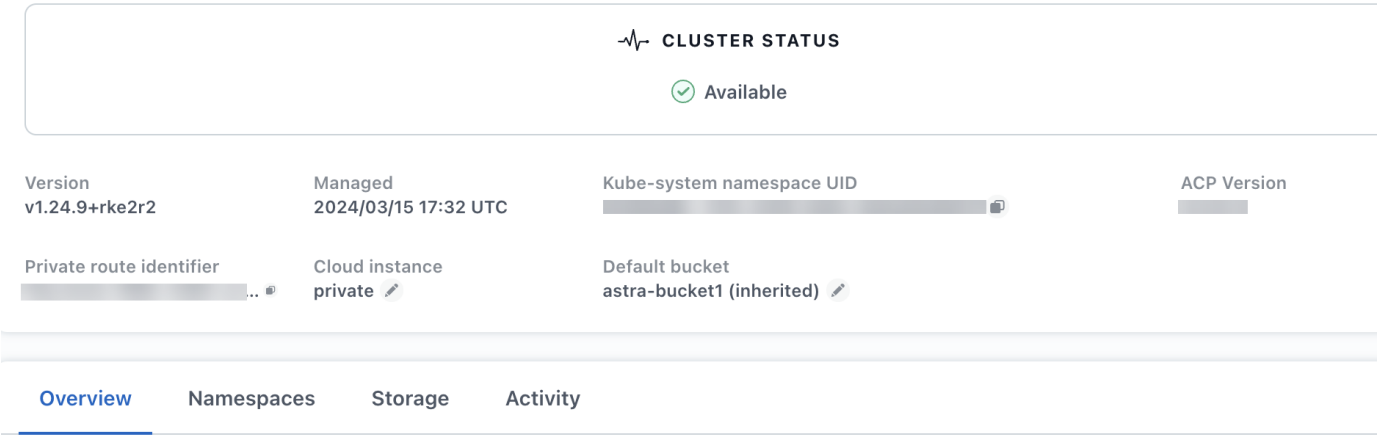
否Asta三端技术将继续采用开源方式、并可免费下载。现在、使用Astra Control配置程序功能需要Astra Control许可证。

是否可以在不安装和使用所有**Astra Control**的情况下使用**Astra Control**中的存储管理和配置功能?

可以。即使您不想使用Astra Control数据管理功能的完整功能集、也可以升级到Astra Control配置程序并使用其功能。

如何知道**Astra Control**配置程序是否已取代了集群上的**Astra Trident**?

安装Asta Control配置程序后、Asta Control UI中的主机集群将显示 `ACP version` 而不是 `Trident version` 字段和当前安装的版本号。



如果您无权访问此UI、则可以使用以下方法确认安装成功:

Asta三端操作员

验证 `trident-acp` 容器正在运行 `acpVersion` 为 `23.10.0` 或更高版本、状态为 `Installed`：

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

响应：

```
status:
  acpVersion: 23.10.0
  currentInstallationParams:
    ...
    acpImage: <my_custom_registry>/trident-acp:v23.10.0
    enableACP: "true"
    ...
  ...
  status: Installed
```

Tridentctl

确认已启用Asta Control配置程序：

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

响应：

```
+----------------+----------------+-------------+ | SERVER VERSION |
CLIENT VERSION | ACP VERSION | +----------------+----------------
+-------------+ | 23.10.0 | 23.10.0 | 23.10.0. | +----------------
+----------------+-------------+
```

法律声明

法律声明提供对版权声明、商标、专利等的访问。

版权

"https://www.netapp.com/company/legal/copyright/"

商标

NetApp 、 NetApp 徽标和 NetApp 商标页面上列出的标记是 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。

"https://www.netapp.com/company/legal/trademarks/"

专利

有关 NetApp 拥有的专利的最新列表，请访问：

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

隐私政策

"https://www.netapp.com/company/legal/privacy-policy/"

开放源代码

通知文件提供有关 NetApp 软件中使用的第三方版权和许可证的信息。

"有关 Astra 的通知"

Astra Control API 许可证

https://docs.netapp.com/us-en/astra-automation/media/astra-api-license.pdf