采用NetApp Astra和适用于Red Hat OpenShift的Cisco Intersight的FlexPod混合云

FlexPod

NetApp
March 25, 2024

This PDF was generated from https://docs.netapp.com/zh-cn/flexpod/hybrid-cloud/flexpod-rho-cvo-solution-overview.html on March 25, 2024. Always check docs.netapp.com for the latest.

目录

采用NetApp Astra和适用于Red Hat OpenShift的Cisco Intersight的FlexPod 混合云 . . . 1
TR-4936：采用NetApp Astra和适用于Red Hat OpenShift的Cisco Intersight的FlexPod 混合云 . . . 1
解决方案组件 . . . 2
安装和配置 . . . 10
解决方案验证 . . . 31
结论 . . . 52

采用NetApp Astra和适用于Red Hat OpenShift的Cisco Intersight的FlexPod 混合云

TR-4936：采用NetApp Astra和适用于Red Hat OpenShift的Cisco Intersight的FlexPod 混合云

Abhinav Singh

简介

随着容器和Kubernetes成为开发、部署、运行、管理和扩展容器化应用程序的实际选择、企业越来越多地在这些应用程序上运行业务关键型应用程序。业务关键型应用程序在很大程度上依赖于状态。有状态应用程序具有关联的状态、数据和配置信息、并依靠先前的数据事务来执行其业务逻辑。在Kubernetes上运行的业务关键型应用程序仍然像传统应用程序一样具有可用性和业务连续性要求。服务中断可能会严重影响公司的收入、工作效率和声誉损失。因此、在集群、内部数据中心和混合云环境内部和之间快速轻松地保护、恢复和移动Kubernetes工作负载至关重要。企业已经认识到将业务转变为混合云模式的优势、将应用程序现代化到云原生外形规格的优势也在他们的市场中居高不下。

本技术报告将NetApp Astra控制中心与基于FlexPod 融合基础架构解决方案 的Red Hat OpenShift容器平台结合在一起、并扩展到Amazon Web Services (AWS)以构成混合云数据中心。以熟悉为基础 "FlexPod 和Red Hat OpenShift"本文档介绍了NetApp Astra控制中心、从安装、配置、应用程序保护工作流以及内部和云之间的应用程序迁移开始。同时、还讨论了在将NetApp Astra控制中心用于在Red Hat OpenShift上运行的容器化应用程序时、应用程序感知型数据管理功能(例如备份和恢复、业务连续性)的优势。

下图显示了解决方案 概述。

[错误：缺少图形映像]

audience

本文档的目标受众包括首席技术官(CTO)、应用程序开发人员、Cloud解决方案 架构师、站点可靠性工程师(SRE)、开发运营工程师、ITOps以及专注于设计、托管和管理容器化应用程序的专业服务团队。

NetApp Astra Control—主要用例

NetApp Astra Control旨在为使用Cloud原生 微服务的客户简化应用程序保护：

- *使用快照的时间点(PIT)应用程序表示。*借助Astra Control、您可以为容器化应用程序创建端到端快照、其中包括Kubernetes上运行的应用程序的配置详细信息以及相关的永久性存储。发生意外事件时、可以单击按钮将应用程序还原到已知正常状态。
- *完整副本应用程序备份。*借助Astra Control、您可以按预定义的计划执行完整的应用程序备份、该计划可用于自动将应用程序还原到同一K8s集群或另一个K8s集群。
- *应用程序可移植性和克隆迁移。*借助Astra Control、您可以将整个应用程序及其数据从一个Kubernetes集群克隆到另一个集群或同一个K8s集群中。无论集群位于何处、此功能还有助于在K8s集群之间移植或迁移应用程序(只需在克隆后删除源应用程序实例即可)。
- *自定义应用程序一致性。*借助Astra Control、您可以利用执行挂钩来控制应用程序暂停状态的定义。将‘pre’和‘post '执行挂钩置于快照和备份工作流中、在创建快照或备份之前、您的应用程序将以您自己的方式暂停。

- *自动执行应用程序级灾难恢复(DR)。*借助Astra Control、您可以为容器化应用程序配置业务连续性灾难恢复(Business Continuity Disaster Recovery、BCDr)计划。后端使用NetApp SnapMirror、并自动完成灾难恢复工作流的实施。

解决方案 拓扑

本节介绍解决方案 的逻辑拓扑。

下图显示了解决方案 拓扑结构、其中包括运行OpenShift容器平台集群的FlexPod 内部环境以及采用NetApp Cloud Volumes ONTAP 、Cisco Intersight和NetApp Cloud Manager SaaS平台的AWS上的自管理OpenShift容器平台集群。

[错误：缺少图形映像]

第一个OpenShift容器平台集群是在FlexPod 上安装的裸机集群、第二个OpenShift容器平台集群部署在FlexPod 上运行的VMware vSphere上、第三个OpenShift容器平台集群部署为 "专用集群" 作为一个自助管理基础架构、迁移到AWS上的现有虚拟私有云(VPC)中。

在此解决方案 中、FlexPod 通过站点到站点VPN连接到AWS、但是、客户也可以使用直接连接实施扩展到混合云。Cisco Intersight用于管理FlexPod 基础架构组件。

在此解决方案 中、Astra控制中心负责管理在FlexPod 和AWS上运行的OpenShift容器平台集群上托管的容器化应用程序。Astra控制中心安装在FlexPod 上运行的OpenShift裸机实例上。Astra Control与主节点上的Kube-API进行通信、并持续监控Kubernetes集群是否发生更改。添加到K8s集群中的任何新应用程序都会自动发现并提供管理。

容器化应用程序的PIT表示可以使用Astra控制中心捕获为快照。可以通过计划的保护策略或按需触发应用程序快照。对于Astra支持的应用程序、快照的崩溃状态是一致的。应用程序快照构成永久性卷中应用程序数据的快照、以及与该应用程序关联的各种Kubernetes资源的应用程序元数据。

可以使用使用预定义的备份计划或按需使用Astra Control创建应用程序的完整副本备份。对象存储用于存储应用程序数据的备份。NetApp ONTAP S3、NetApp StorageGRID 以及任何通用S3实施均可用作对象存储。

"接下来：解决方案 组件。"

解决方案组件

"上一步：解决方案概述。"

FlexPod

FlexPod 是一组定义的硬件和软件，可为虚拟化和非虚拟化解决方案奠定集成基础。FlexPod 包括NetApp ONTAP 存储、Cisco Nexus网络、Cisco MDS存储网络、Cisco统一计算系统(Cisco UCS)。该设计非常灵活、可以将网络、计算和存储安装到一个数据中心机架中、也可以根据客户的数据中心设计进行部署。端口密度允许网络组件容纳多种配置。

Astra Control

Astra Control为公有 云和内部环境中托管的云原生应用程序提供应用程序感知型数据保护服务。Astra Control可为在Kubernetes上运行的容器化应用程序提供数据保护、灾难恢复和迁移功能。

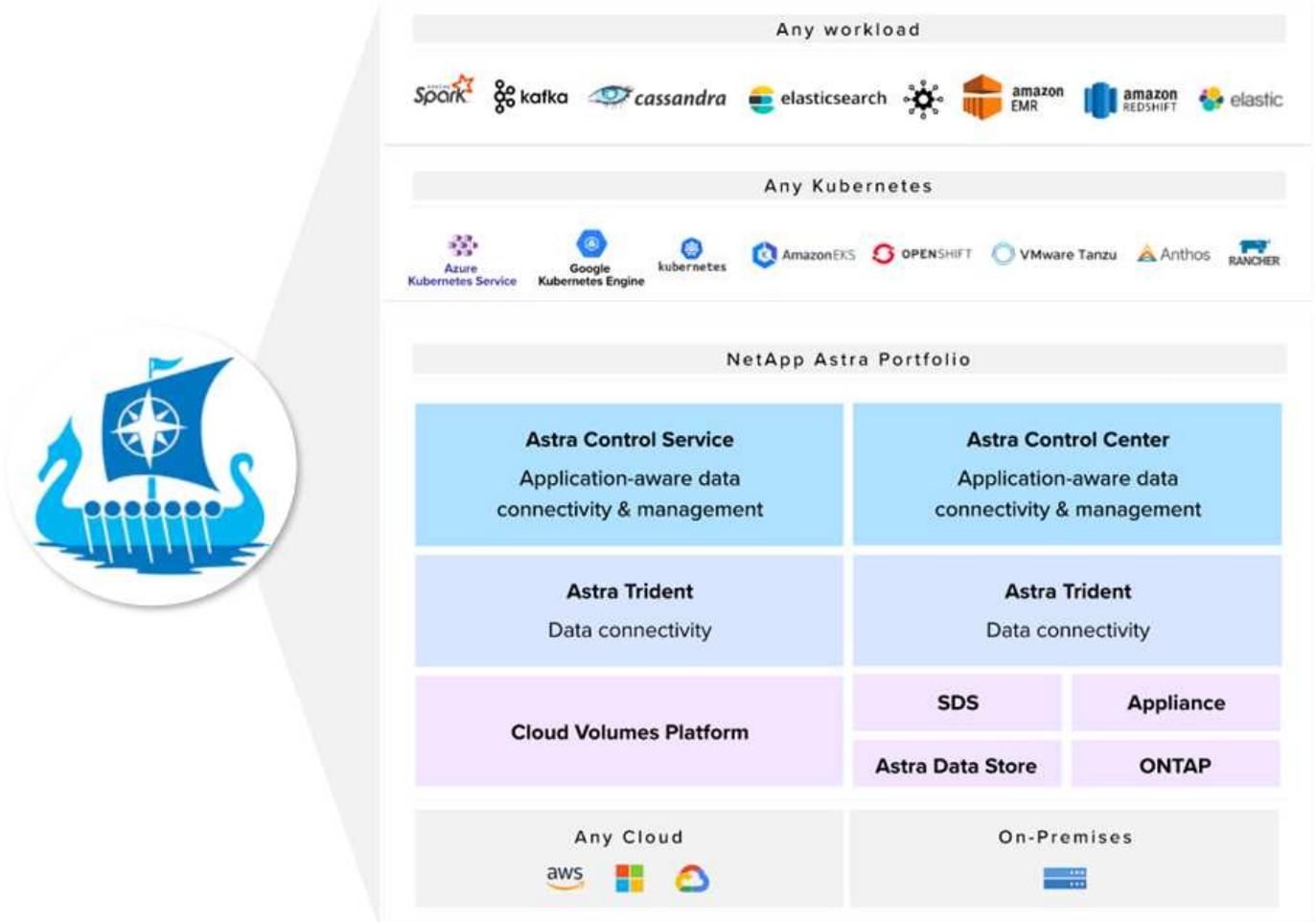
功能

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
- 创建应用程序一致的按需快照和备份
- 自动执行策略驱动的快照和备份操作
- 在混合云环境中将应用程序和关联数据从一个Kubernetes集群迁移到另一个集群
- 将应用程序克隆到同一个K8s集群或另一个K8s集群
- 直观显示应用程序保护状态
- 提供图形用户界面和完整的REST API列表、用于利用现有内部工具实施所有保护工作流。

Astra Control为容器化应用程序提供了一个单一的管理平台可视化视图、其中包括对在Kubernetes集群上创建的相关资源的深入了解。您可以使用一个门户查看所有云或所有数据中心中的所有集群、所有应用程序。您可以在所有环境(内部或公有 云)中使用Astra Control API来实施数据管理工作流。

下图显示了Astra Control功能。



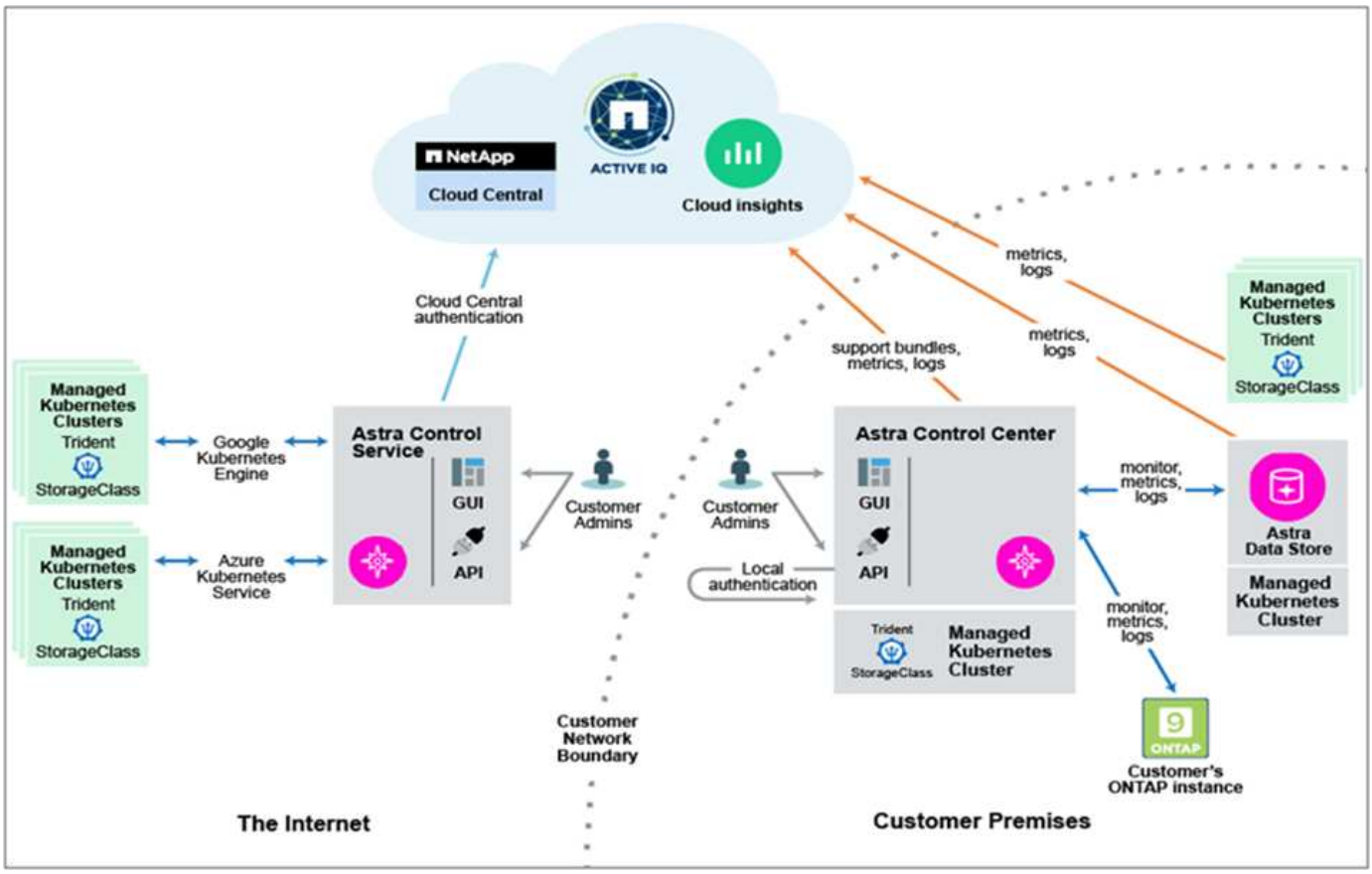
Astra Control消费模式

Astra Control有两种消费模式：

- * Astra Control Service。* NetApp托管的一项完全托管服务、可在Google Kubernetes Engine (GKEE) Azure Kubernetes Service (AKS)中为Kubernetes集群提供应用程序感知型数据管理。
- * Astra Control Center。*自管理软件、可为内部和混合云环境中运行的Kubernetes集群提供应用程序感知型数据管理。

本技术报告利用Astra控制中心管理Kubernetes上运行的云原生应用程序。

下图显示了Astra Control架构。



Astra Trident

Astra Trident是一款完全受支持的开源存储编排程序、适用于容器和Kubernetes分发版。它从一开始就经过精心设计、可帮助您使用行业标准界面(如)满足容器化应用程序的持久性需求 "容器存储接口(CSI)"。借助Astra Trident、微服务和容器化应用程序可以利用NetApp存储系统产品组合提供的企业级存储服务。

Astra Trident作为Pod部署在Kubernetes集群上、并为您的Kubernetes工作负载提供动态存储编排服务。它可以帮助您的容器化应用程序快速轻松地使用NetApp广泛的产品组合中的永久性存储、其中包括NetApp ONTAP (NetApp AFF 、NetApp FAS 、NetApp ONTAP Select 、云、 和适用于NetApp ONTAP 的Amazon FSx) 、NetApp Element 软件(NetApp SolidFire)以及Azure NetApp Files 服务、Google Cloud上的云卷服务和AWS上的云卷服务。在FlexPod 环境中、Astra Trident用于为容器动态配置和管理永久性卷、这些容器由NetApp FlexVol 卷和ONTAP 存储平台(例如NetApp AFF 和FAS 系统以及Cloud Volumes ONTAP)上托管的LUN提供支持。Trident在实施Astra Control提供的应用程序保护方案方面也发挥着关键作用。有关Astra Trident的详细信息、请参见 "Astra Trident文档。"

存储后端

要使用Astra Trident、您需要受支持的存储后端。Trident后端定义了Trident与存储系统之间的关系。它会告诉Trident如何与该存储系统通信、以及Trident如何从该存储系统配置卷。Trident将自动从后端提供符合存储类定义的要求的存储池。

- ONTAP AFF 和 FAS 存储后端。作为存储软件和硬件平台、ONTAP 可提供核心存储服务、支持多个存储访问协议以及存储管理功能、例如NetApp Snapshot副本和镜像。
- Cloud Volumes ONTAP 存储后端
- "Astra 数据存储" 存储后端

NetApp Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP 是一款软件定义的存储产品、可为文件和块工作负载提供高级数据管理。借助Cloud Volumes ONTAP 、您可以优化云存储成本并提高应用程序性能、同时增强数据保护、安全性和合规性。

主要优势包括：

- 利用内置的重复数据删除，数据压缩，精简配置和克隆功能最大限度地降低存储成本。
- 在云环境发生故障时，确保企业级可靠性和持续运行。
- Cloud Volumes ONTAP 利用NetApp行业领先的复制技术SnapMirror将内部数据复制到云中、因此可以轻松地为多种使用情形提供二级副本。
- Cloud Volumes ONTAP 还与 Cloud Backup Service 集成，提供备份和还原功能，以保护云数据并对其进行长期归档。
- 按需在高性能和低性能存储池之间切换，而无需使应用程序脱机。
- 使用NetApp SnapCenter 确保Snapshot副本的一致性。
- Cloud Volumes ONTAP 支持数据加密，并提供防病毒和勒索软件保护。
- 与 Cloud Data sense 集成有助于您了解数据环境并识别敏感数据。

Cloud Central

Cloud Central提供了一个中央位置、用于访问和管理NetApp云数据服务。通过这些服务、您可以在云中运行关键应用程序、创建自动化灾难恢复站点、备份数据以及跨多个云有效迁移和控制数据。有关详细信息，请参见 "Cloud Central。"

Cloud Manager

Cloud Manager是一款基于SaaS的企业级管理平台、IT专家和云架构师可以利用NetApp的云解决方案集中管理其混合多云基础架构。它提供了一个集中式系统、用于查看和管理内部和云存储、支持混合、多个云提供商和客户。有关详细信息，请参见 "Cloud Manager"。

连接器

Connector是一个实例、可使Cloud Manager管理公有 云环境中的资源和流程。要使用Cloud Manager提供的许多功能、需要使用Connector。连接器可以部署在云或内部网络中。

连接器在以下位置受支持：

- AWS
- Microsoft Azure
- Google Cloud
- 在您的内部环境中

要了解有关Connector的更多信息、请参见 "此链接。"

NetApp Cloud Insights

Cloud Insights 是一款NetApp云基础架构监控工具、可用于监控由Astra控制中心管理的Kubernetes集群的性能和利用率。Cloud Insights 将存储使用量与工作负载相关联。在 Astra 控制中心中启用 Cloud Insights 连接后，遥测信息将显示在 Astra 控制中心 UI 页面中。

NetApp Active IQ Unified Manager

借助NetApp Active IQ Unified Manager 、您可以通过一个经过重新设计的直观界面监控ONTAP 存储集群、该界面可通过社区智慧和AI分析提供智能。它可以全面洞察存储环境及其运行的虚拟机(VM)的运行状况、性能和主动式能力。当存储基础架构发生问题描述 时、Unified Manager可以通知您问题描述 的详细信息、以帮助您确定根发生原因。通过VM信息板、您可以查看VM的性能统计信息、以便调查从VMware vSphere主机到网络最后再到存储的整个I/O路径。某些事件还提供了可用于更正问题描述 的补救措施。您可以为事件配置自定义警报、以便在发生问题时、通过电子邮件和SNMP陷阱通知您。Active IQ Unified Manager 可以预测容量和使用趋势、以便在出现问题之前主动采取行动、从而防止做出长期可能导致其他问题的被动短期决策、从而为用户的存储需求进行规划。

Cisco Intersight

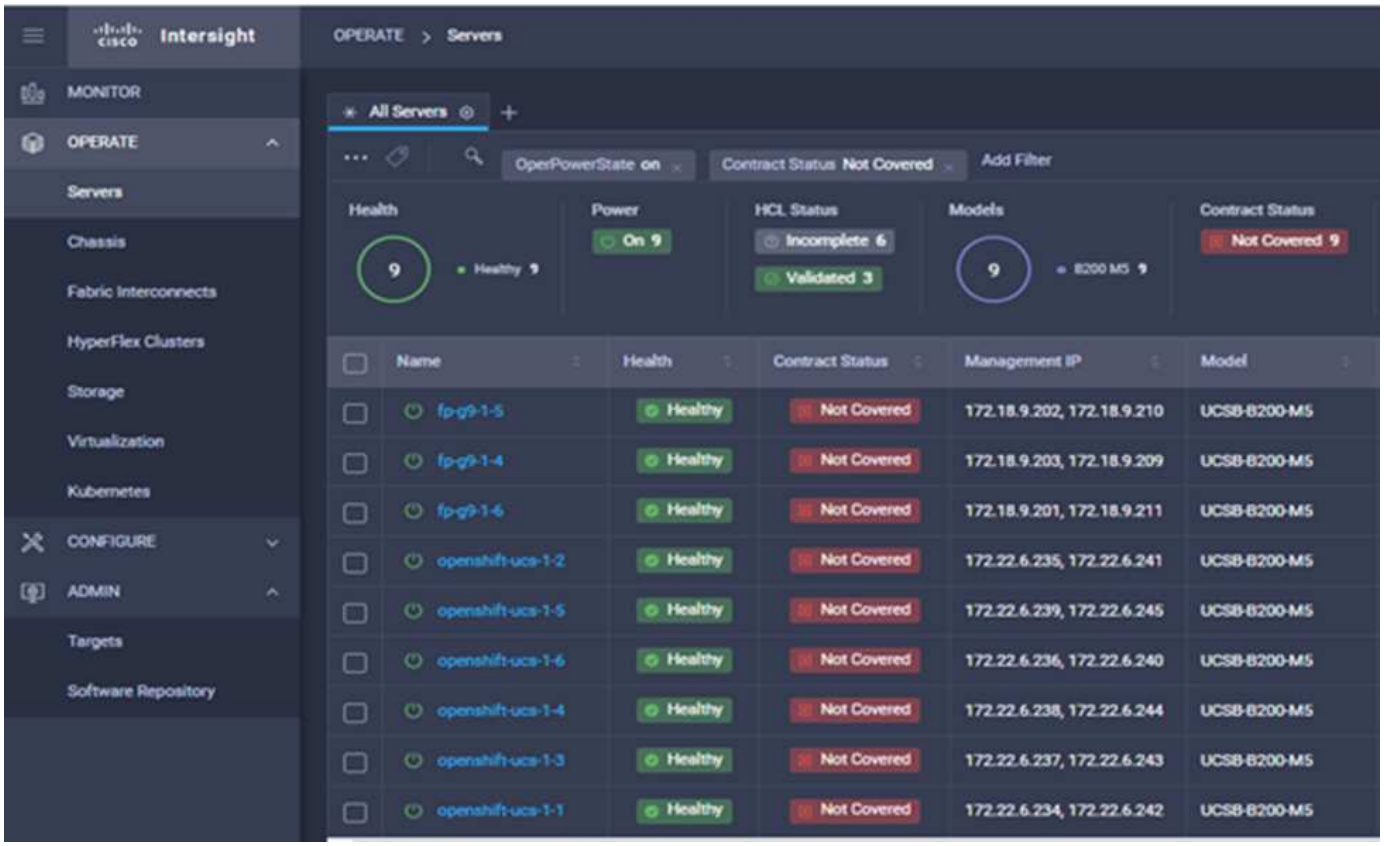
Cisco Intersight是一个SaaS平台、可为传统和云原生应用程序和基础架构提供智能自动化、可观察性和优化功能。该平台可帮助IT团队推动变革、并提供专为混合云设计的运营模式。

Cisco Intersight具有以下优势：

- *交付速度更快。*由于采用基于敏捷性的软件开发模式、可从云或客户数据中心以服务的形式交付、并经常进行更新和持续创新。这样、客户就可以专注于加快业务部门的交付速度。
- *简化操作。*通过使用一个安全的SaaS交付工具以及通用清单、身份验证和API在整个堆栈和所有位置之间运行、简化操作、消除团队之间的孤岛。从内部管理物理服务器和虚拟机管理程序到虚拟机、K8s、无服务器、自动化、 在内部和公有 云之间实现优化和成本控制。
- *持续优化。*利用Cisco Intersight在每一层提供的智能以及Cisco TAC持续优化您的环境。这种智能功能可转换为建议的可自动操作、因此您可以实时适应每个变化：从移动工作负载和监控物理服务器的运行状况到自动调整K8s集群大小、再到您所使用的公有 云的成本降低建议。

Cisco Intersight支持两种管理操作模式：UCSM受管模式(Umm)和Intersight受管模式(IMM)。在首次设置互联阵列期间、您可以为光纤连接的Cisco UCS系统选择原生 Umm或IMM。在此解决方案 中、使用原生 Umm。

下图显示了Cisco Intersight信息板。

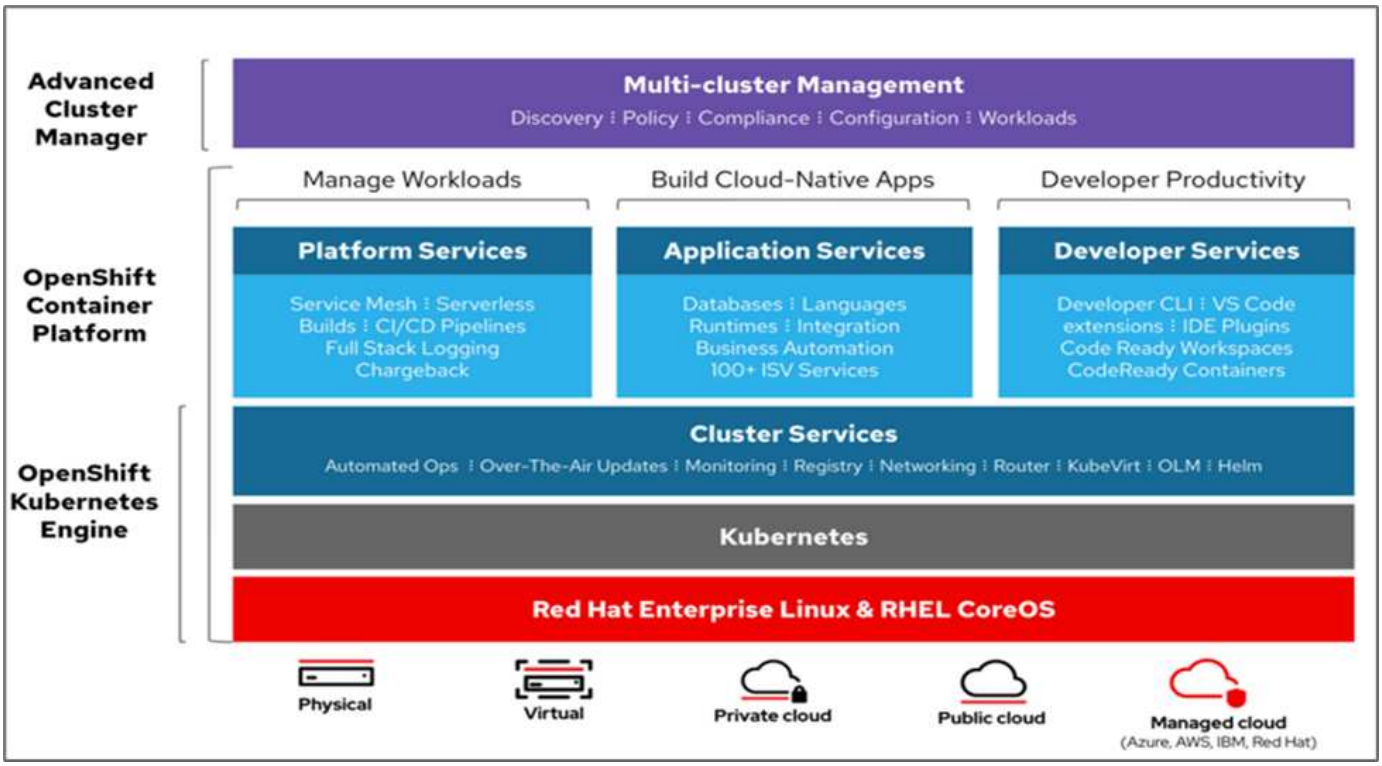


Red Hat OpenShift 容器平台

Red Hat OpenShift容器平台是一个容器应用程序平台、它将CRI-O和Kubernetes汇集在一起、并提供一个API和Web界面来管理这些服务。CRI-O是Kubernetes容器运行时接口(CRI)的实施、可使用与Open Container Initiative (OCI)兼容的运行时。它是使用Docker作为Kubernetes运行时的轻型替代方案。

OpenShift容器平台允许客户创建和管理容器。容器是独立的进程、在其自身环境中运行、独立于操作系统和底层基础架构。OpenShift容器平台有助于开发、部署和管理基于容器的应用程序。它提供了一个自助服务平台、可按需创建、修改和部署应用程序、从而加快开发和发布生命周期。OpenShift容器平台具有一个基于微服务的架构、该架构包含可协同工作的小型分离单元。它在Kubernetes集群上运行、其中包含有关对象的数据存储在etcd中、etcd是一个可靠的集群模式密钥值存储。

下图概述了Red Hat OpenShift容器平台。



Kubernetes基础架构

在OpenShift容器平台中、Kubernetes跨一组CRI-O运行时主机管理容器化应用程序、并提供部署、维护和应用程序扩展机制。CRI-O服务软件包、实例化和运行容器化应用程序。

Kubernetes集群由一个或多个主节点和一组工作节点组成。此解决方案 设计包括硬件和软件堆栈中的高可用性(HA)功能。Kubernetes集群设计为在HA模式下运行、其中包含三个主节点和至少两个辅助节点、以帮助确保集群不会出现单点故障。

Red Hat核心操作系统

OpenShift容器平台使用Red Hat Enterprise Linux CoreOS (RHCOS)、这是一种面向容器的操作系统、它结合了CoreOS和Red Hat Atomic Host操作系统的一些最佳特性和功能。RHCOS专门为从OpenShift容器平台运行容器化应用程序而设计、可与新工具配合使用、以实现快速安装、基于操作员的管理和简化的升级。

RHCOS包括以下功能：

- IGNITION、OpenShift容器平台将其用作首次启动系统配置、用于首次启动和配置计算机。
- CRI-O是一种Kubernetes原生 容器运行时实施、它与操作系统紧密集成、可提供高效且优化的Kubernetes体验。CRI-O可提供运行、停止和重新启动容器的功能。它完全取代了OpenShift容器平台3中使用的Docker容器引擎。
- Kubernetes的主节点代理Kubelet负责启动和监控容器。

VMware vSphere 7.0

VMware vSphere是一个虚拟化平台、可将大量基础架构(包括CPU、存储和网络等资源)作为一个无缝、多功能且动态的操作环境进行全面管理。与管理单个计算机的传统操作系统不同、VMware vSphere可将整个数据中心的基础架构聚合在一起、从而创建一个具有资源的动力中心、这些资源可以快速动态地分配给任何需要的应用程序。

有关详细信息，请参见 "VMware vSphere"。

VMware vSphere vCenter

VMware vCenter Server可通过一个控制台统一管理所有主机和VM、并对集群、主机和VM进行聚合性能监控。通过VMware vCenter Server、管理员可以深入了解计算集群、主机、虚拟机、存储、子操作系统、 虚拟基础架构的其他关键组件。VMware vCenter可管理VMware vSphere环境中提供的丰富功能。

硬件和软件版本

可以将此解决方案 扩展到运行中定义的受支持软件、固件和硬件版本的任何FlexPod 环境 "NetApp 互操作性表工具" 和 "Cisco UCS硬件兼容性列表。" OpenShift集群以裸机方式安装在FlexPod 和VMware vSphere上。

要管理多个OpenShift (K8s)集群、只需要一个Astra控制中心实例、而每个OpenShift集群上都安装了Trident CSI。Astra控制中心可以安装在其中任何一个OpenShift集群上。在此解决方案 中、Astra控制中心安装在OpenShift裸机集群上。

下表列出了OpenShift的FlexPod 硬件和软件版本。

组件	产品	**version**
计算	Cisco UCS互联阵列6454	4.1 (3c)
	Cisco UCS B200 M5服务器	4.1 (3c)
网络	Cisco Nexus 9336C-x2 NX-OS	9.3 (8)
存储	NetApp AFF A700	9.11.1
	NetApp Astra 控制中心	22.04.0
	NetApp Astra Trident CSI插件	22.04.0
	NetApp Active IQ Unified Manager	9.11
软件	VMware ESXi nenic 以太网驱动程序	1.0.35.0
	vSphere ESXi	7.0 (U2)
	VMware vCenter设备	7.0 U2b
	Cisco Intersight Assist虚拟设备	1.0.9-342
	OpenShift容器平台	4.9.
	OpenShift容器平台主节点	RHCOS 4.9
	OpenShift容器平台工作节点	RHCOS 4.9

下表列出了AWS上OpenShift的软件版本。

组件	产品	**version**
计算	主实例类型：m5.xlarge	不适用
	员工实例类型：m5.large	不适用
网络	虚拟私有云传输网关	不适用

组件	产品	**version**
存储	NetApp Cloud Volumes ONTAP	9.11.1
	NetApp Astra Trident CSI插件	22.04.0
软件	OpenShift容器平台	4.9.
	OpenShift容器平台主节点	RHCOS 4.9
	OpenShift容器平台工作节点	RHCOS 4.9

"接下来：适用于OpenShift容器平台4的FlexPod 裸机安装。"

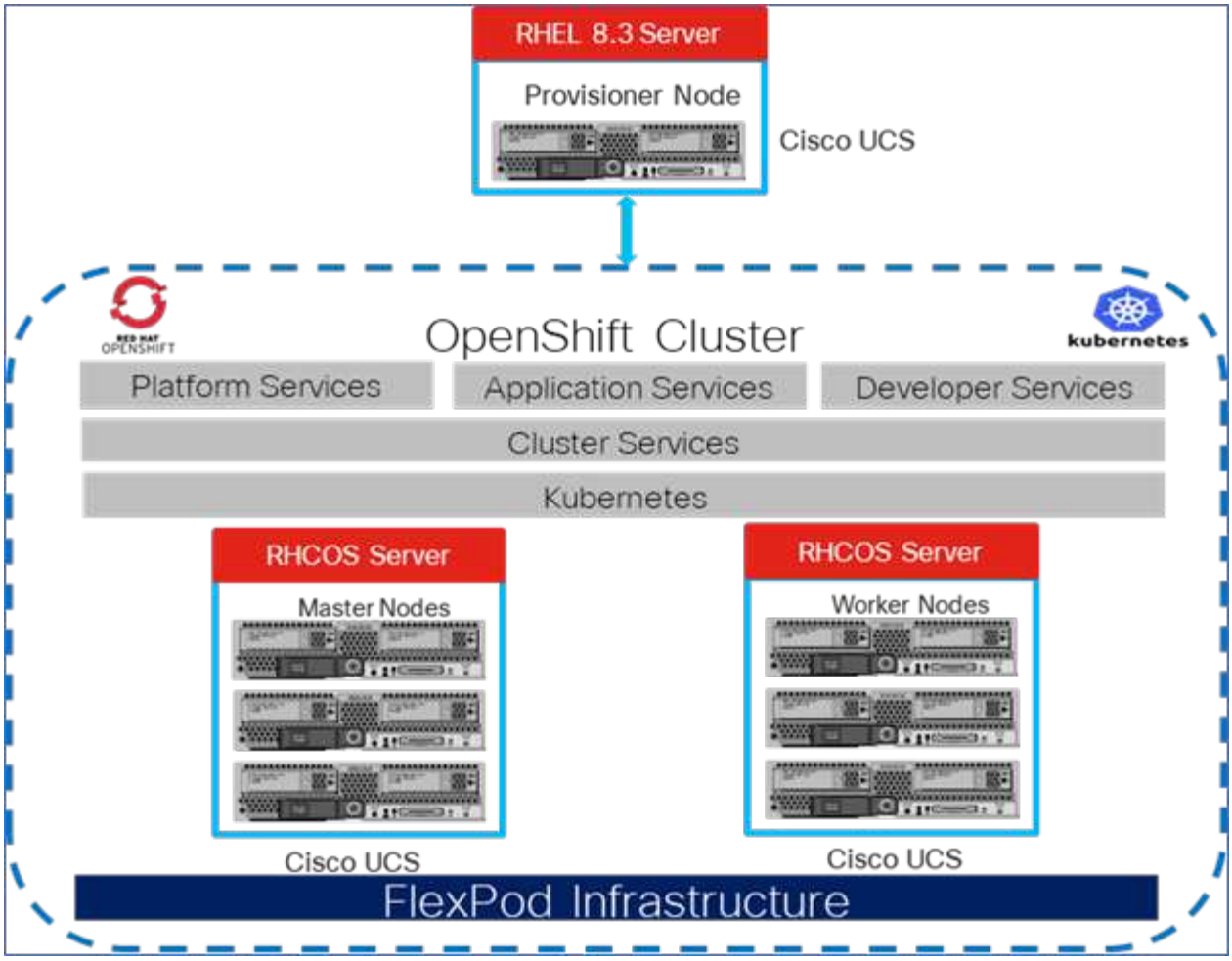
安装和配置

适用于**OpenShift**容器平台**4**的**FlexPod** 裸机安装

"先前版本：解决方案 组件。"

要了解适用于OpenShift容器平台4的FlexPod 裸机设计、部署详细信息以及NetApp Astra Trident安装和配置、请参见 "采用OpenShift的FlexPod Cisco经验证的设计和部署指南(CVD)"。此CVD涵盖使用Ansible的FlexPod 和OpenShift容器平台部署。CVD还提供了有关准备工作节点、Astra Trident安装、存储后端和存储类配置的详细信息、这些配置是部署和配置Astra控制中心的几个前提条件。

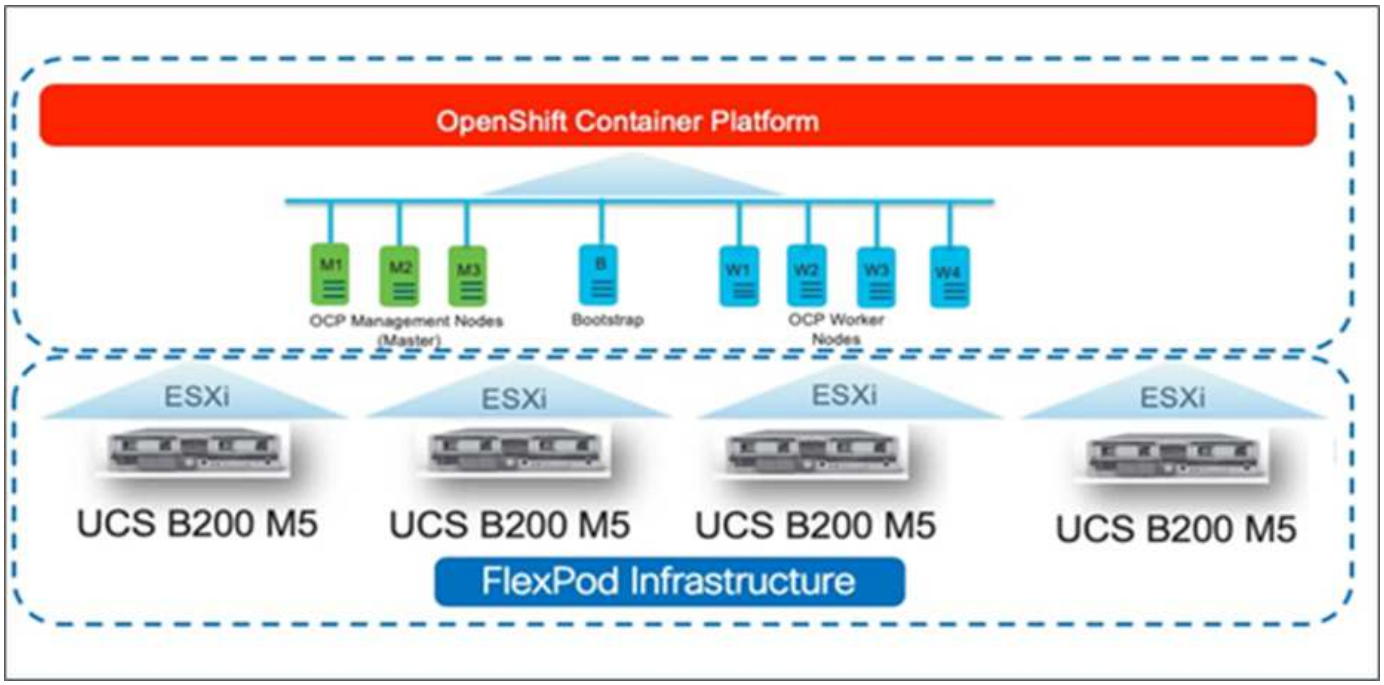
下图显示了FlexPod 上的OpenShift容器平台4裸机。



VMware安装上的适用于OpenShift容器平台4的FlexPod

有关在运行VMware vSphere的FlexPod 上部署Red Hat OpenShift容器平台4的详细信息、请参见 "适用于OpenShift容器平台的FlexPod 数据中心4."。

下图显示了vSphere上适用于OpenShift容器平台4的FlexPod。

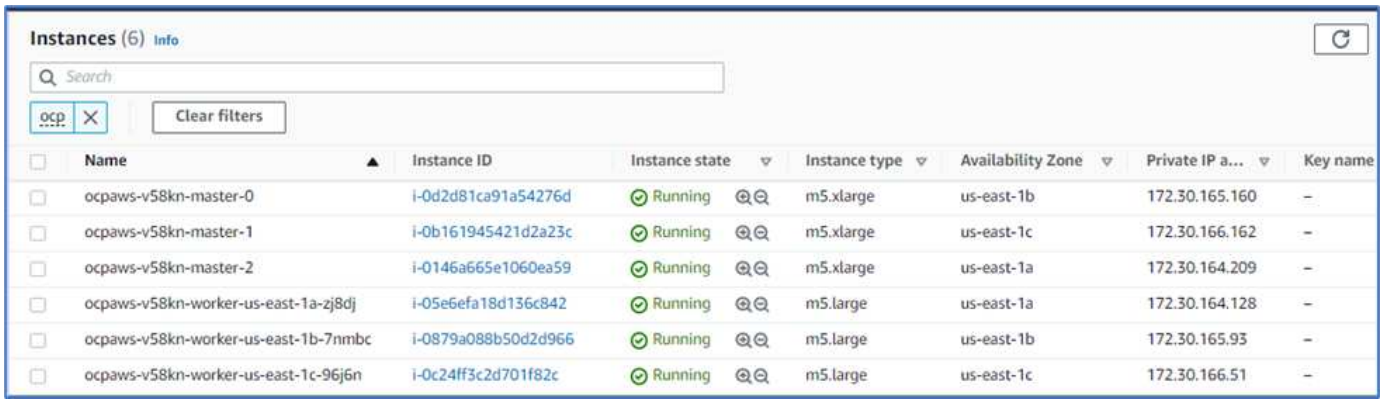


"接下来：在AWS上运行Red Hat OpenShift。"

AWS上的Red Hat OpenShift

"先前版本：适用于OpenShift容器平台4的FlexPod 裸机安装。"

AWS上会部署一个单独的自管理OpenShift容器平台4集群作为灾难恢复站点。主节点和工作节点跨越三个可用性区域、以实现高可用性。



Instances (6) Info

Search

ocp | Clear filters

Name	Instance ID	Instance state	Instance type	Availability Zone	Private IP a...	Key name
ocpaws-v58kn-master-0	i-0d2d81ca91a54276d	Running	m5.xlarge	us-east-1b	172.30.165.160	–
ocpaws-v58kn-master-1	i-0b161945421d2a23c	Running	m5.xlarge	us-east-1c	172.30.166.162	–
ocpaws-v58kn-master-2	i-0146a665e1060ea59	Running	m5.xlarge	us-east-1a	172.30.164.209	–
ocpaws-v58kn-worker-us-east-1a-zj8dj	i-05e6efa18d136c842	Running	m5.large	us-east-1a	172.30.164.128	–
ocpaws-v58kn-worker-us-east-1b-7nmbc	i-0879a088b50d2d966	Running	m5.large	us-east-1b	172.30.165.93	–
ocpaws-v58kn-worker-us-east-1c-96j6n	i-0c24ff3c2d701f82c	Running	m5.large	us-east-1c	172.30.166.51	–

```
[ec2-user@ip-172-30-164-92 ~]$ oc get nodes
NAME                           STATUS   ROLES    AGE   VERSION
ip-172-30-164-128.ec2.internal   Ready    worker   29m   v1.22.8+f34b40c
ip-172-30-164-209.ec2.internal   Ready    master   36m   v1.22.8+f34b40c
ip-172-30-165-160.ec2.internal   Ready    master   33m   v1.22.8+f34b40c
ip-172-30-165-93.ec2.internal    Ready    worker   30m   v1.22.8+f34b40c
ip-172-30-166-162.ec2.internal   Ready    master   36m   v1.22.8+f34b40c
ip-172-30-166-51.ec2.internal    Ready    worker   28m   v1.22.8+f34b40c
```

OpenShift部署为 "专用集群" 添加到AWS上的现有VPC中。专用OpenShift容器平台集群不会公开外部端点、只能从内部网络访问、并且在Internet上不可见。使用NetApp Cloud Manager部署单节点NetApp Cloud Volumes ONTAP 、它为Astra Trident提供存储后端。

有关在AWS上安装OpenShift的详细信息、请参见 "OpenShift文档"。

"下一步：NetApp Cloud Volumes ONTAP。"

NetApp Cloud Volumes ONTAP

"先前版本：AWS上的Red Hat OpenShift。"

NetApp Cloud Volumes ONTAP 实例部署在AWS上、并用作Astra Trident的后端存储。在添加Cloud Volumes ONTAP 工作环境之前、必须先部署连接器。如果您尝试在没有连接器的情况下创建首个Cloud Volumes ONTAP 工作环境、Cloud Manager将提示您。要在AWS中部署Connector、请参见 "创建连接器"。

要在AWS上部署Cloud Volumes ONTAP 、请参见 "AWS快速入门"。

部署Cloud Volumes ONTAP 后、您可以在OpenShift容器平台集群上安装Astra Trident并配置存储后端和Snapshot类。

"接下来：在OpenShift容器平台上安装Astra Control Center。"

在OpenShift容器平台上安装Astra Control Center

"先前版本：NetApp Cloud Volumes ONTAP。"

您可以将Astra控制中心安装在FlexPod 上运行的OpenShift集群上、也可以安装在具有Cloud Volumes ONTAP 存储后端的AWS上。在此解决方案 中、Astra控制中心部署在OpenShift裸机集群上。

可以使用所述的标准过程安装Astra控制中心 "此处" 或从Red Hat OpenShift OperatorHub获取。Astra控制操作员是Red Hat认证的操作员。在此解决方案 中、使用Red Hat OperatorHub安装Astra控制中心。

环境要求

- Astra控制中心支持多个Kubernetes分发版；对于Red Hat OpenShift、支持的版本包括Red Hat OpenShift容器平台4.8或4.9。
- 除了环境和最终用户的应用程序资源要求之外、Astra控制中心还需要以下资源：

组件	要求
存储后端容量	至少500 GB可用
工作节点	至少3个辅助节点、每个节点具有4个CPU核心和12 GB RAM
完全限定域名(FQDN)地址	Astra 控制中心的 FQDN 地址
Astra Trident	已安装并配置 Astra Trident 21.04 或更高版本

组件	要求
入口控制器或负载平衡器	配置入口控制器以使用URL或负载平衡器公开Astra控制中心、以提供IP地址、并将其解析为FQDN

- 您必须具有可将Astra Control Center构建映像推送到的现有私有映像注册表。您需要提供用于上传映像的映像注册表的URL。

> 执行某些工作流时会提取某些映像、并在必要时创建和销毁容器。

- Astra 控制中心要求创建一个存储类并将其设置为默认存储类。Astra 控制中心支持由 Astra Trident 提供的以下 ONTAP 驱动程序：
  - ontap-NAS
  - ontap-nas-flexgroup
  - ontap-san
  - ontap-san-economy.

> 我们假设已部署的OpenShift集群安装了Astra Trident并配置了ONTAP 后端、同时还定义了默认存储类。

- 对于OpenShift环境中的应用程序克隆、Astra Control Center需要允许OpenShift挂载卷并更改文件所有权。要修改ONTAP 导出策略以允许执行这些操作、请运行以下命令：

```
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -superuser sys
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -anon 65534
```

> 要将第二个OpenShift操作环境添加为托管计算资源、请确保已启用Astra Trident卷快照功能。要使用Astra Trident启用和测试卷快照、请参见官方信息 "Astra Trident说明"。

- 答 "VolumeSnapClass" 应在管理应用程序的所有Kubernetes集群上进行配置。这也可能包括安装了Astra控制中心的K8s集群。Astra控制中心可以管理运行该控制中心的K8s集群上的应用程序。

应用程序管理要求

- *许可。*要使用Astra控制中心管理应用程序、您需要获得Astra控制中心许可证。
- *命名空间。*命名空间是可由Astra控制中心作为应用程序进行管理的最大实体。您可以选择根据现有命名空间中的应用程序标签和自定义标签筛选出组件、并将部分资源作为应用程序进行管理。
- *存储类。*如果您安装的应用程序明确设置了StorageClass、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StorageClass。将显式设置了StorageClass的应用程序克隆到不具有相同StorageClass的集群失败。
- * Kubernetes Resources。*使用非Astra Control捕获的Kubernetes资源的应用程序可能无法提供完整的应用程序数据管理功能。Astra Control可以捕获以下Kubernetes资源：

Kubernetes资源		
ClusterRole	ClusterRoleBinding.	配置映射
自定义资源定义	自定义资源	cronjob
DemonSet	HorizontalPodAutoscaler	传入
DeploymentConfig	MutatingWebhook	PersistentVolumeClaim
POD	PodDisruptionBuget	播客模板
网络策略	ReplicaSet	Role
RoleBinding.	路由	机密
验证 Webhook		

使用 **OpenShift OperatorHub** 安装 **Astra** 控制中心

以下操作步骤 将使用Red Hat OperatorHub安装Astra控制中心。在此解决方案 中、Astra控制中心安装在运行于FlexPod 上的裸机OpenShift集群上。

1. 从下载 Astra 控制中心捆绑包（`Astra-control-center-[version].tar.gz`） "NetApp 支持站点"。
2. 从下载Astra控制中心证书和密钥的.zip文件 "NetApp 支持站点"。
3. 验证捆绑包的签名。

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

4. 提取Astra映像。

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

5. 更改为 Astra 目录。

```
cd astra-control-center-[version]
```

6. 将映像添加到本地注册表。

```
For Docker:
docker login [your_registry_path]OR
For Podman:
podman login [your_registry_path]
```

7. 使用相应的脚本加载映像、标记映像并将其推送到本地注册表。

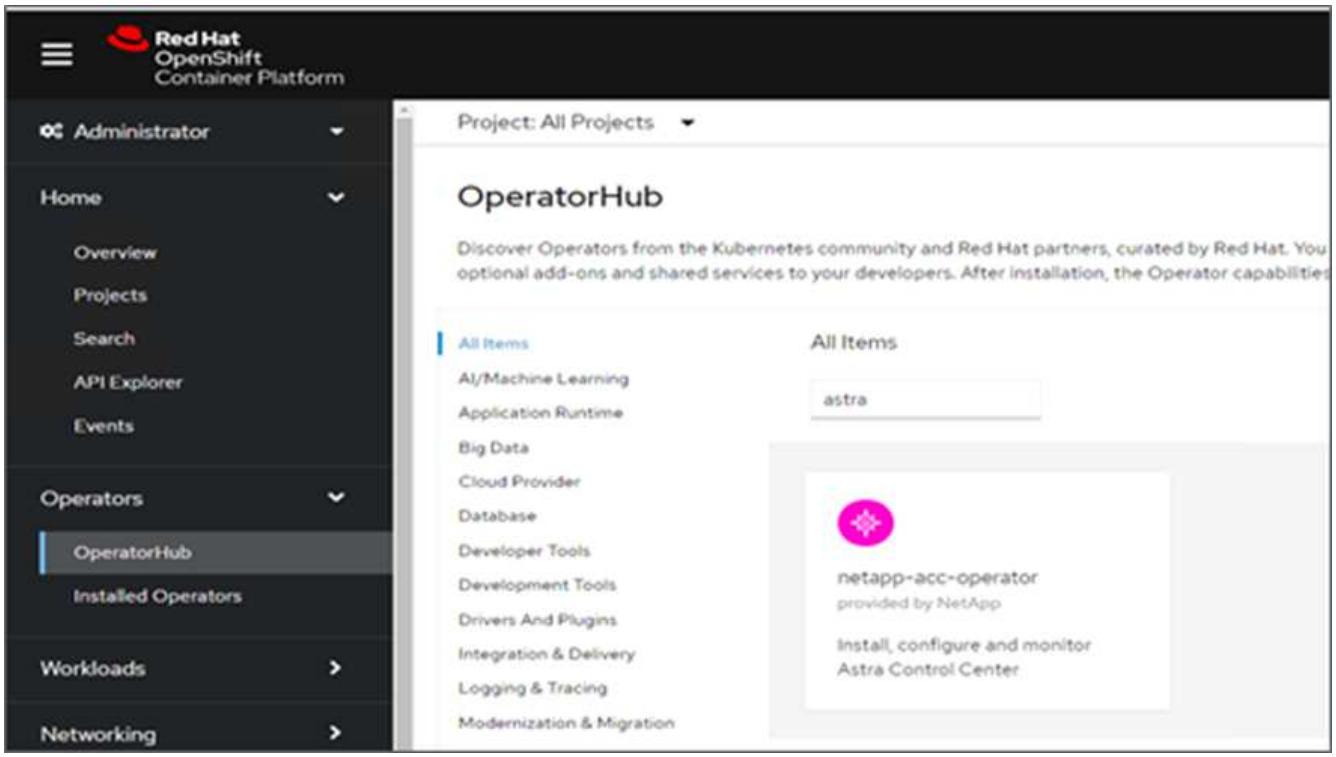
对于 Docker：

```
export REGISTRY=[Docker_registry_path]
for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image trimming
the 'Loaded images: '
  astraImage=$(docker load --input ${astraImageFile} | sed 's/Loaded
image: //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  docker tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  docker push ${REGISTRY}/${astraImage}
done
```

对于 Podman：

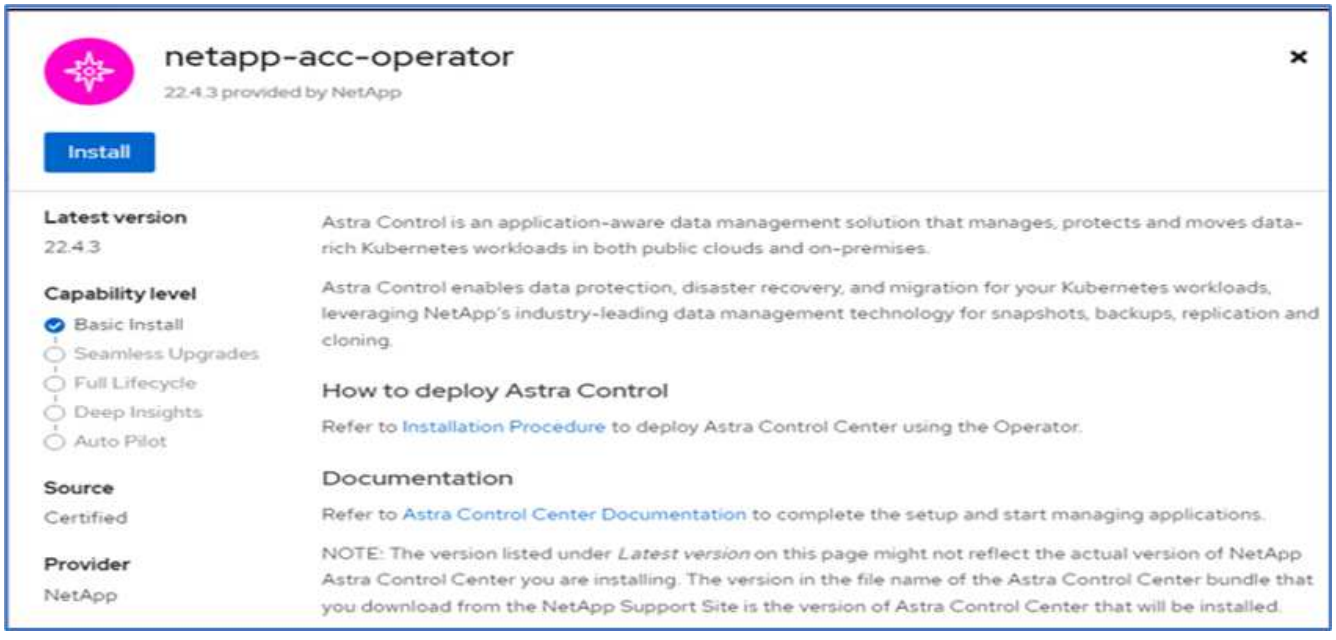
```
export REGISTRY=[Registry_path]
for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image trimming
the 'Loaded images: '
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  podman push ${REGISTRY}/${astraImage}
done
```

8. 登录到裸机OpenShift集群Web控制台。从侧面菜单中、选择Operators > OperatorHub。输入`Astra`以列出`NetApp-Acc-operator`。

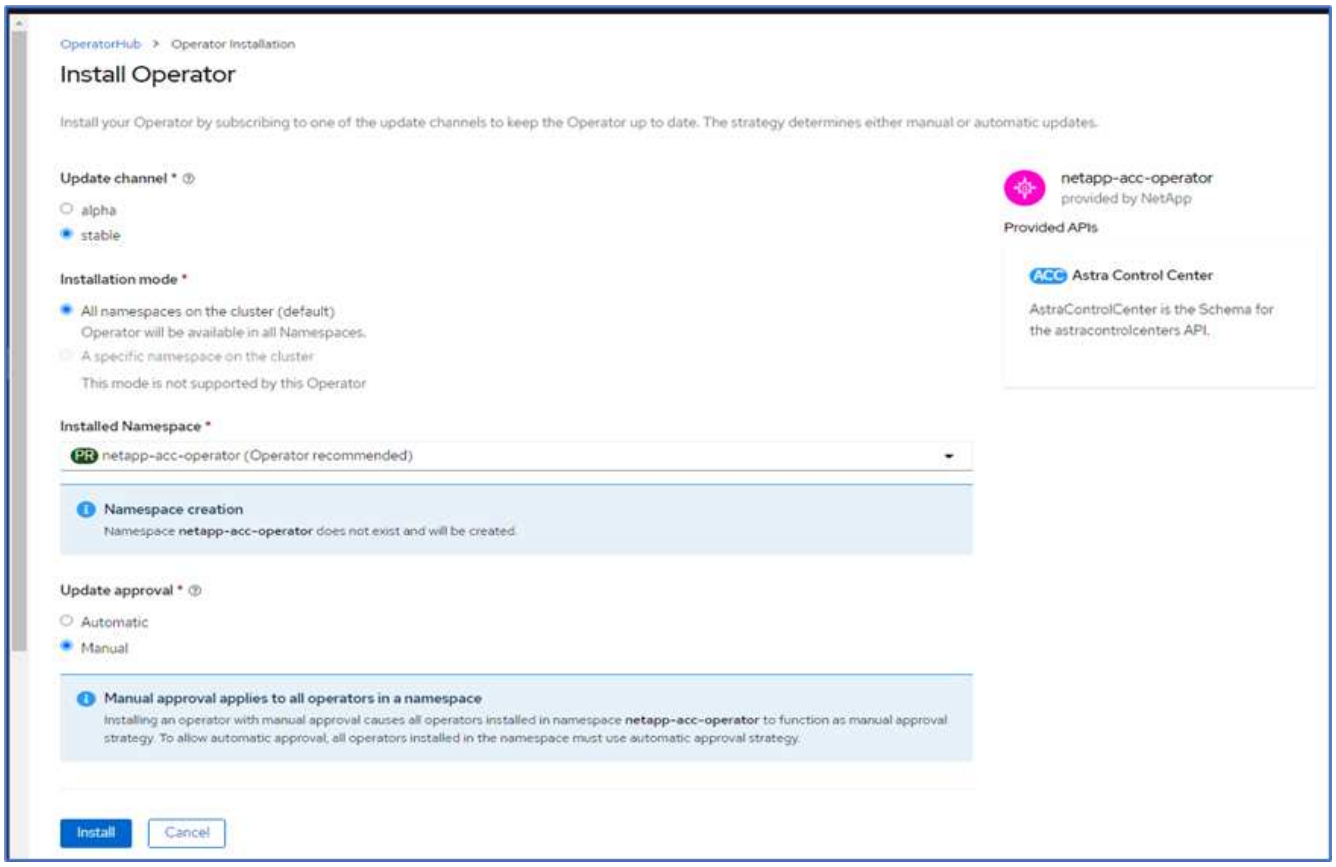


`NetApp-Acc-operator`是一个经过认证的Red Hat OpenShift操作员、并列在OperatorHub目录下。

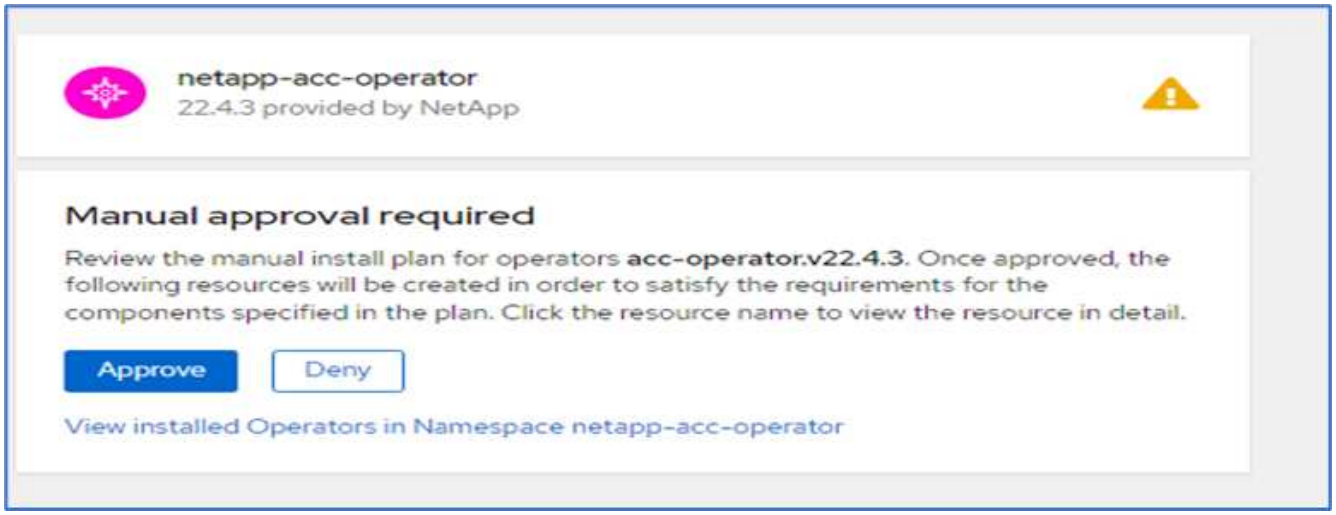
9. 选择`NetApp-Acc-operator`、然后单击安装。



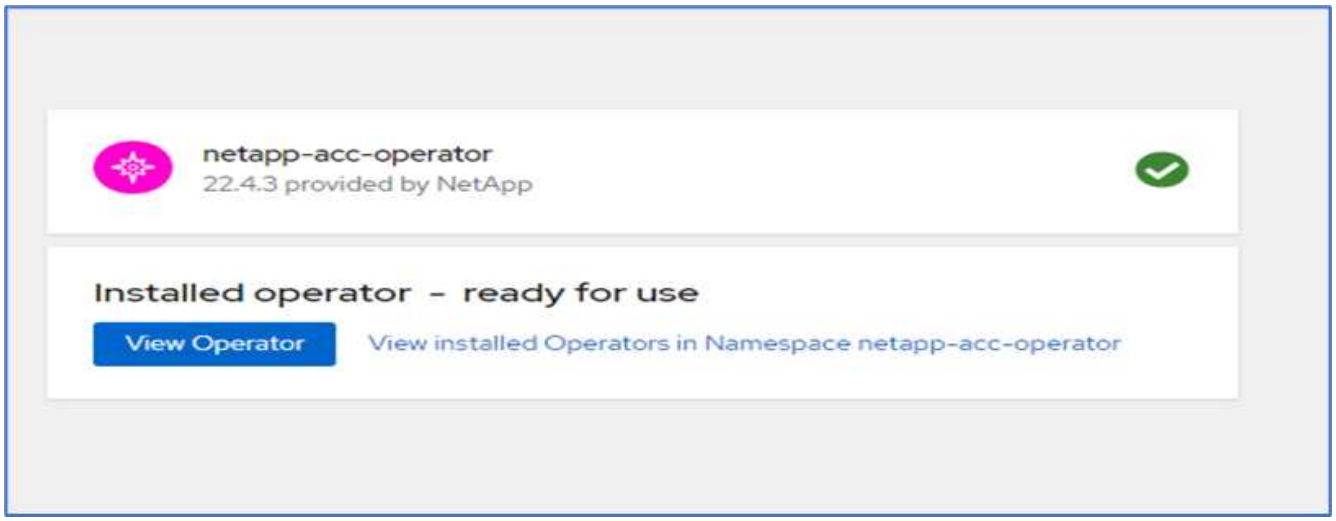
10. 选择相应的选项、然后单击安装。



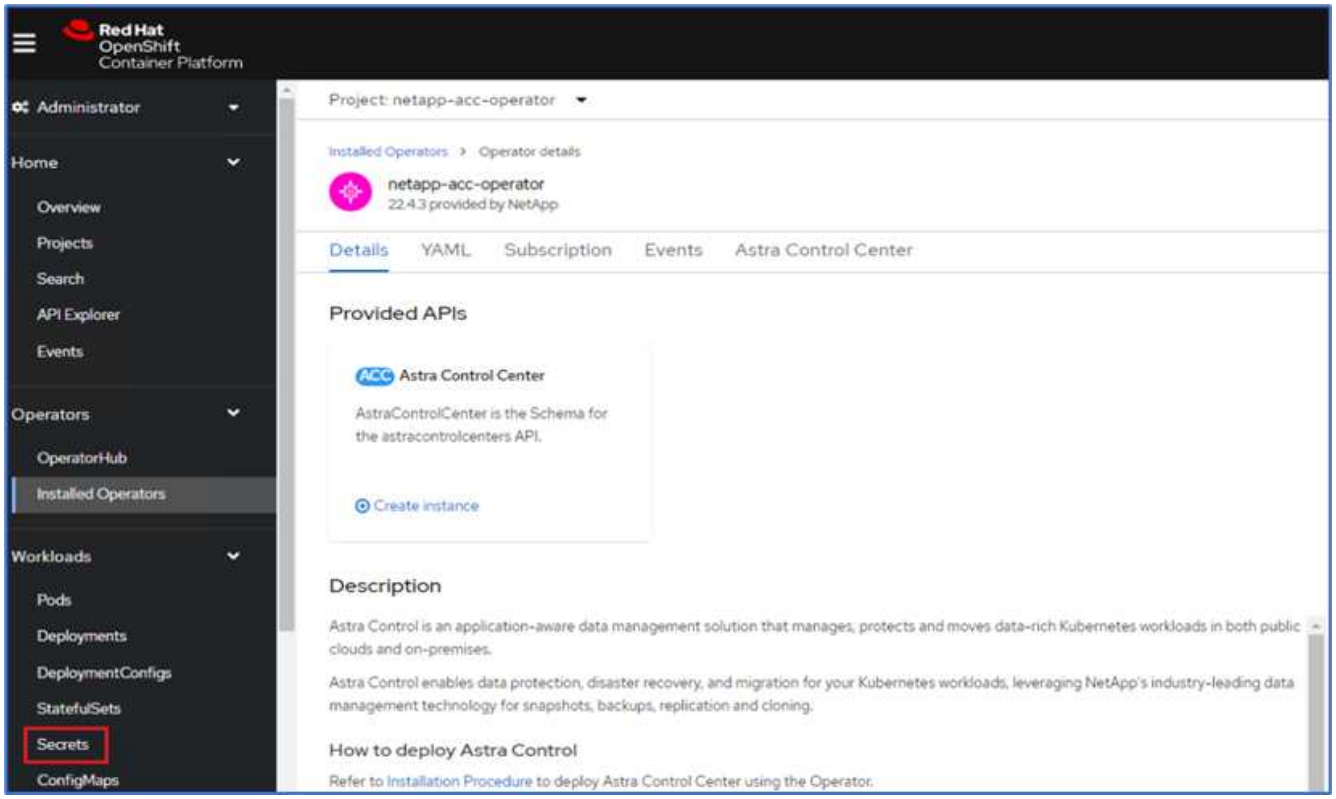
11. 批准安装并等待操作员安装。



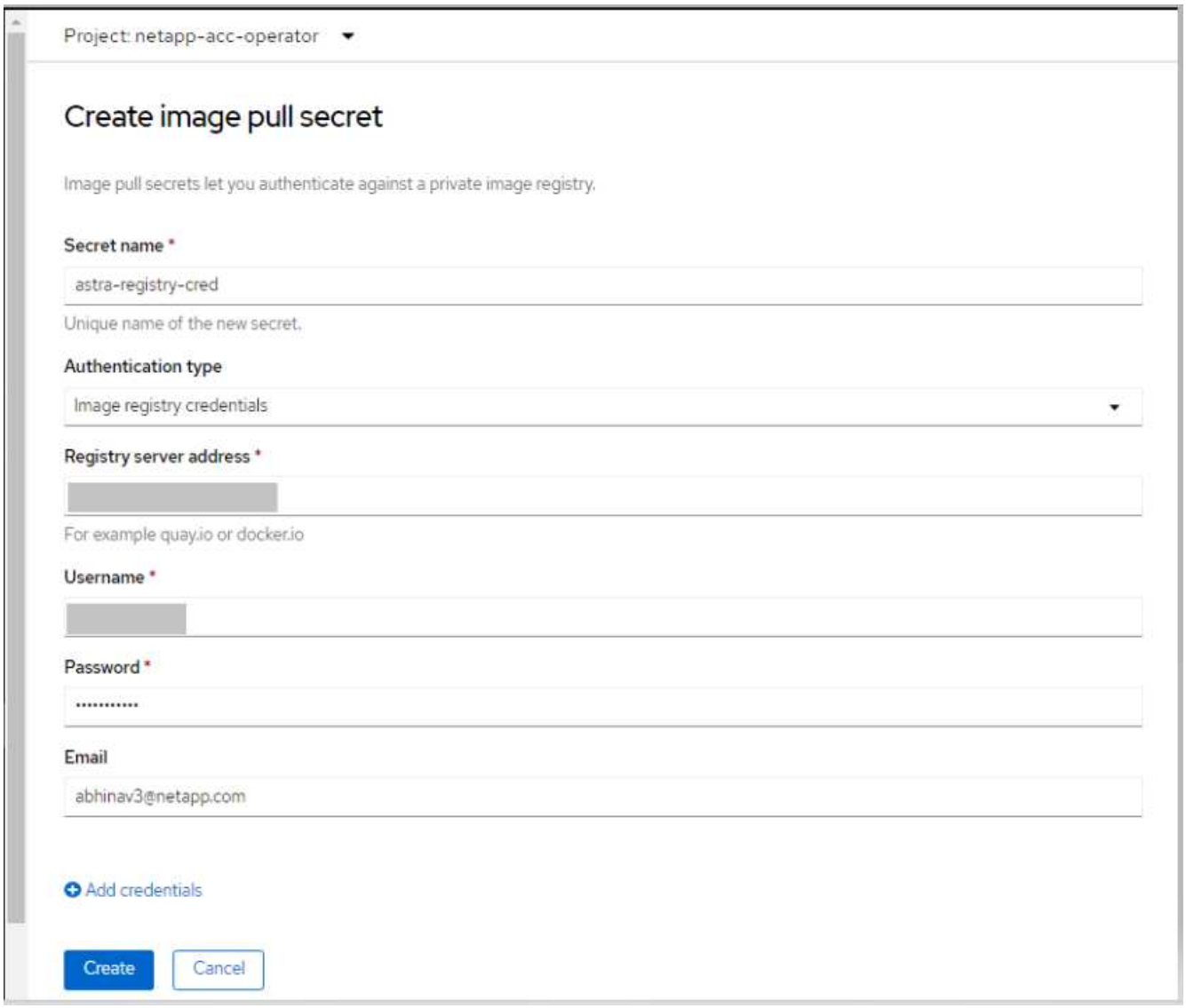
12. 在此阶段、操作员已成功安装并准备就绪、可以使用。单击View Operator开始安装Astra Control Center。



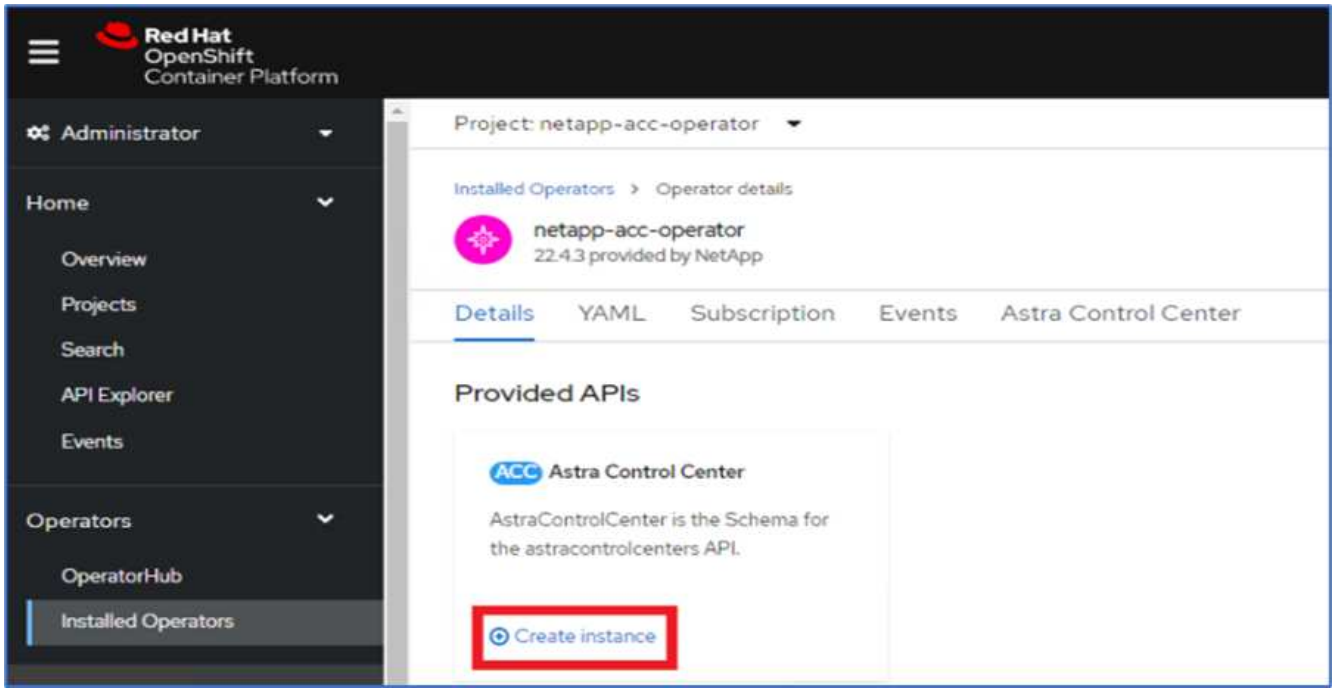
13. 在安装Astra控制中心之前、请创建Pull密钥、以便从先前推送的Docker注册表下载Astra映像。



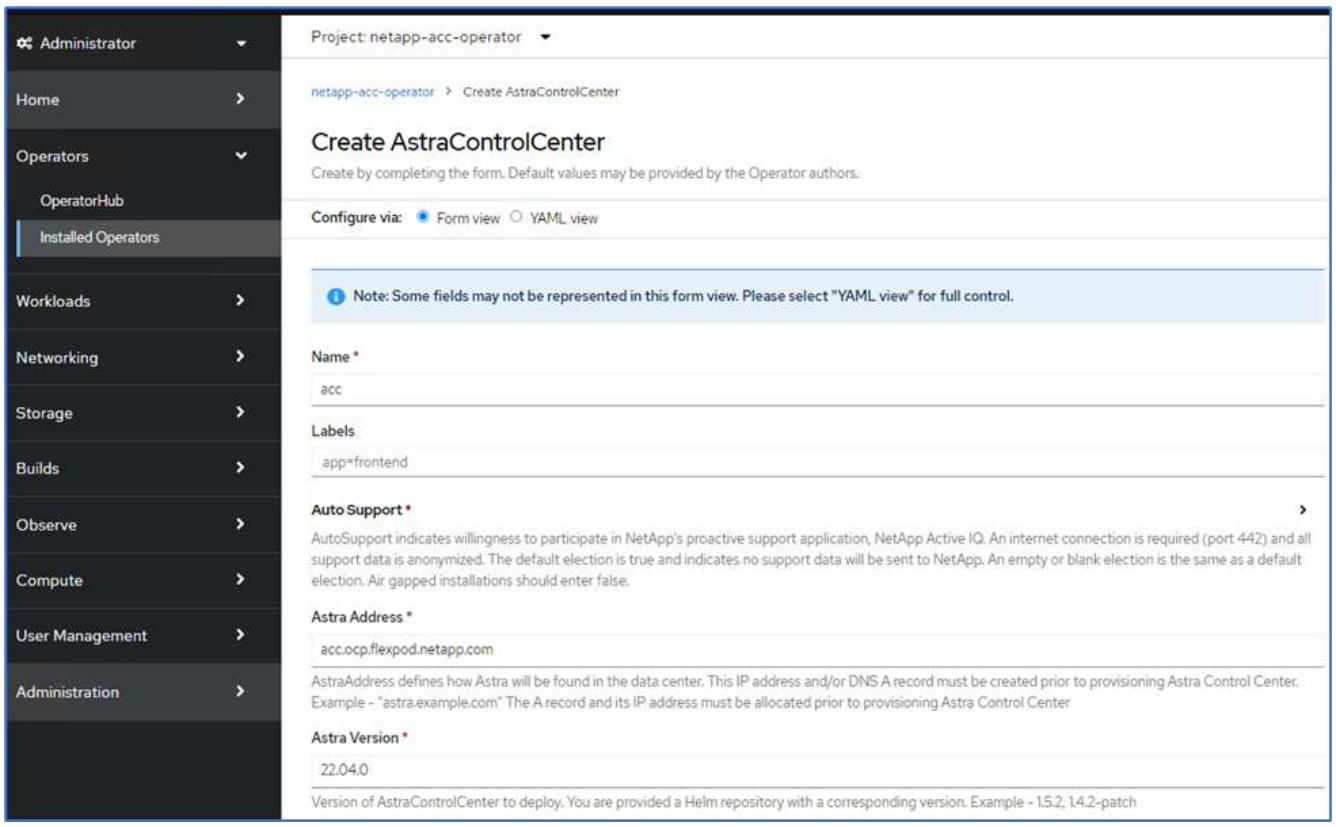
14. 要从Docker私有repod中提取Astra控制中心映像、请在`NetApp-Acc-operator`命名空间中创建一个密钥。此机密名称将在Astra控制中心YAML清单中稍后提供。



15. 从侧面菜单中、选择Operators > Installed Operators、然后单击提供的API部分下的Create Instance。

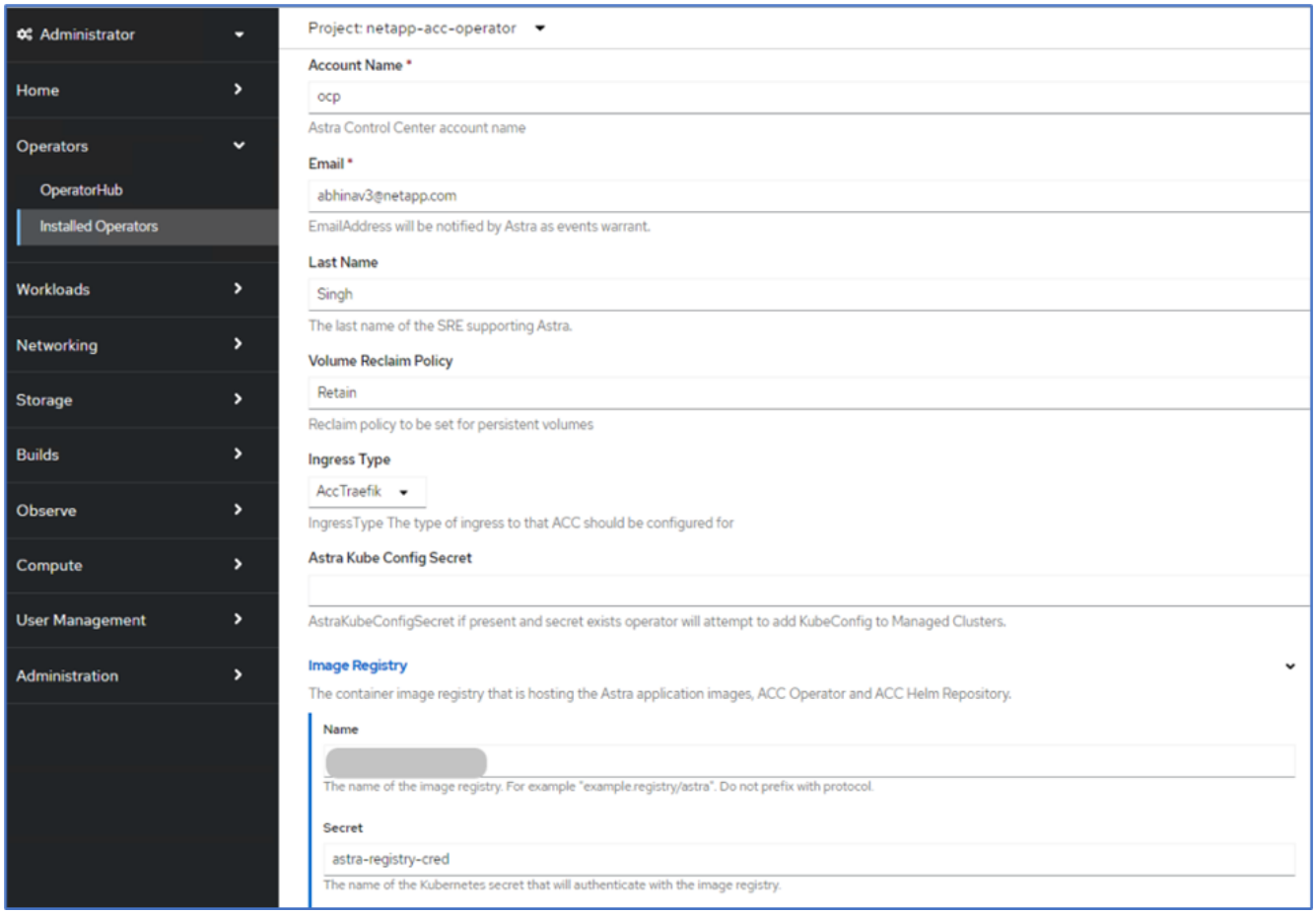


16. 填写创建AstraControlCenter表单。提供名称、Astra地址和Astra版本。



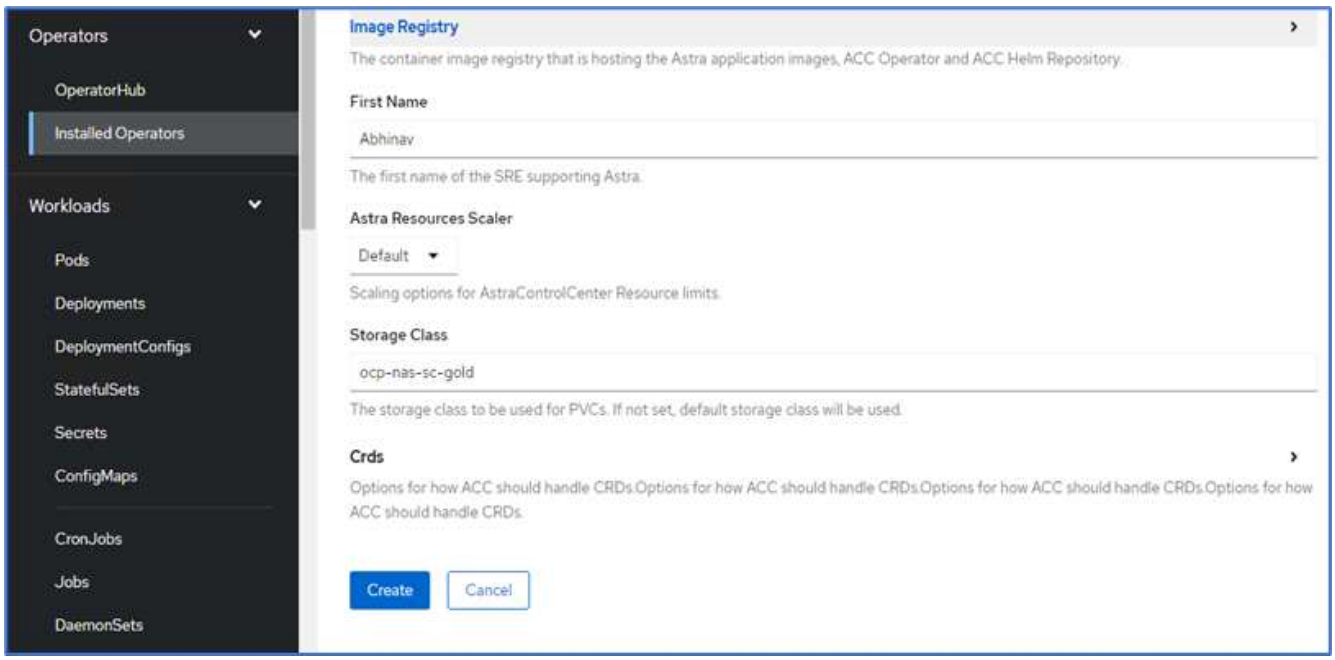
在Astra Address下、提供Astra控制中心的FQDN地址。此地址用于访问Astra控制中心Web控制台。FQDN还应解析为可访问的IP网络、并应在DNS中进行配置。

17. 输入帐户名称、电子邮件地址、管理员姓氏、并保留默认卷回收策略。如果使用的是负载平衡器、请将"传入类型"设置为`AccTraefik`。否则、请为`In防护.Controller`选择Generic。在映像注册表下、输入容器映像注册表路径和密钥。

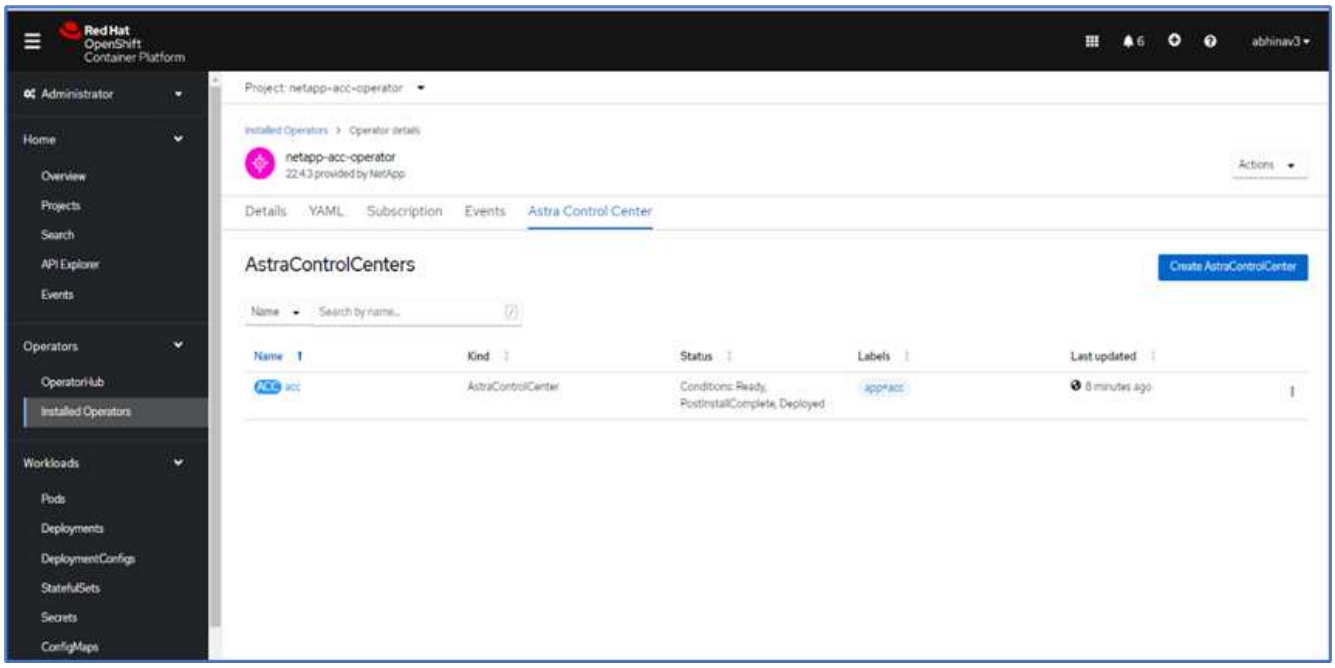


在此解决方案 中、使用了金属负载平衡器。因此、入口类型为AccTraefik。这会将Astra控制中心traefik网关公开为loadbalancer类型的Kubernetes服务。

18. 输入管理员的名字、配置资源扩展并提供存储类。单击创建。



Astra控制中心实例的状态应从"Deploying (部署)"更改为"Ready (就绪)"。



19. 确认所有系统组件均已成功安装、并且所有Pod均已运行。

```
root@abhinav-ansible# oc get pods -n netapp-acc-operator
NAME                                               READY   STATUS
RESTARTS   AGE
acc-helm-repo-77745b49b5-7zg2v                     1/1     Running   0
10m
acc-operator-controller-manager-5c656c44c6-tqnmn   2/2     Running   0
13m
activity-589c6d59f4-x2sfs                          1/1     Running   0
6m4s
api-token-authentication-4q5lj                     1/1     Running   0
5m26s
api-token-authentication-pzptd                     1/1     Running   0
5m27s
api-token-authentication-tbtg6                     1/1     Running   0
5m27s
asup-669df8d49-qps54                               1/1     Running   0
5m26s
authentication-5867c5f56f-dnpp2                    1/1     Running   0
3m54s
bucketservice-85495bc475-5zcc5                     1/1     Running   0
5m55s
cert-manager-67f486bbc6-txhh6                      1/1     Running   0
9m5s
cert-manager-cainjector-75959db744-4l5p5           1/1     Running   0
9m6s
cert-manager-webhook-765556b869-g6wdf              1/1     Running   0
```

```
9m6s
cloud-extension-5d595f85f-txrfl                    1/1     Running   0
5m27s
cloud-insights-service-674649567b-5s4wd            1/1     Running   0
5m49s
composite-compute-6b58d48c69-46vhc                 1/1     Running   0
6m11s
composite-volume-6d447fd959-chnrt                  1/1     Running   0
5m27s
credentials-66668f8ddd-8qc5b                       1/1     Running   0
7m20s
entitlement-fd6fc5c58-wxnmh                        1/1     Running   0
6m20s
features-756bbb7c7c-rgcrm                          1/1     Running   0
5m26s
fluent-bit-ds-278pg                                1/1     Running   0
3m35s
fluent-bit-ds-5pqc6                                1/1     Running   0
3m35s
fluent-bit-ds-8l7cq                                1/1     Running   0
3m35s
fluent-bit-ds-9qbft                                1/1     Running   0
3m35s
fluent-bit-ds-nj475                                1/1     Running   0
3m35s
fluent-bit-ds-x9pd8                                1/1     Running   0
3m35s
graphql-server-698d6f4bf-kftwc                     1/1     Running   0
3m20s
identity-5d4f4c87c9-wjz6c                          1/1     Running   0
6m27s
influxdb2-0                                        1/1     Running   0
9m33s
krakend-657d44bf54-8cb56                           1/1     Running   0
3m21s
license-594bbdc-rghdg                              1/1     Running   0
6m28s
login-ui-6c65fbbbd4-jg8wz                          1/1     Running   0
3m17s
loki-0                                             1/1     Running   0
9m30s
metrics-facade-75575f69d7-hnlk6                    1/1     Running   0
6m10s
monitoring-operator-65dff79cfb-z78vk               2/2     Running   0
3m47s
nats-0                                             1/1     Running   0
```

```
10m
nats-1                                                1/1     Running   0
9m43s
nats-2                                                1/1     Running   0
9m23s
nautilus-7bb469f857-4hlc6                             1/1     Running   0
6m3s
nautilus-7bb469f857-vz94m                             1/1     Running   0
4m42s
openapi-8586db4bcd-gwwvf                              1/1     Running   0
5m41s
packages-6bdb949cfb-nrq8l                             1/1     Running   0
6m35s
polaris-consul-consul-server-0                        1/1     Running   0
9m22s
polaris-consul-consul-server-1                        1/1     Running   0
9m22s
polaris-consul-consul-server-2                        1/1     Running   0
9m22s
polaris-mongodb-0                                     2/2     Running   0
9m22s
polaris-mongodb-1                                     2/2     Running   0
8m58s
polaris-mongodb-2                                     2/2     Running   0
8m34s
polaris-ui-5df7687dbd-trcnf                           1/1     Running   0
3m18s
polaris-vault-0                                       1/1     Running   0
9m18s
polaris-vault-1                                       1/1     Running   0
9m18s
polaris-vault-2                                       1/1     Running   0
9m18s
public-metrics-7b96476f64-j88bw                       1/1     Running   0
5m48s
storage-backend-metrics-5fd6d7cd9c-vcb4j              1/1     Running   0
5m59s
storage-provider-bb85ff965-m7qrq                      1/1     Running   0
5m25s
telegraf-ds-4zqgz                                     1/1     Running   0
3m36s
telegraf-ds-cp9x4                                     1/1     Running   0
3m36s
telegraf-ds-h4n59                                     1/1     Running   0
3m36s
telegraf-ds-jnp2q                                     1/1     Running   0
```

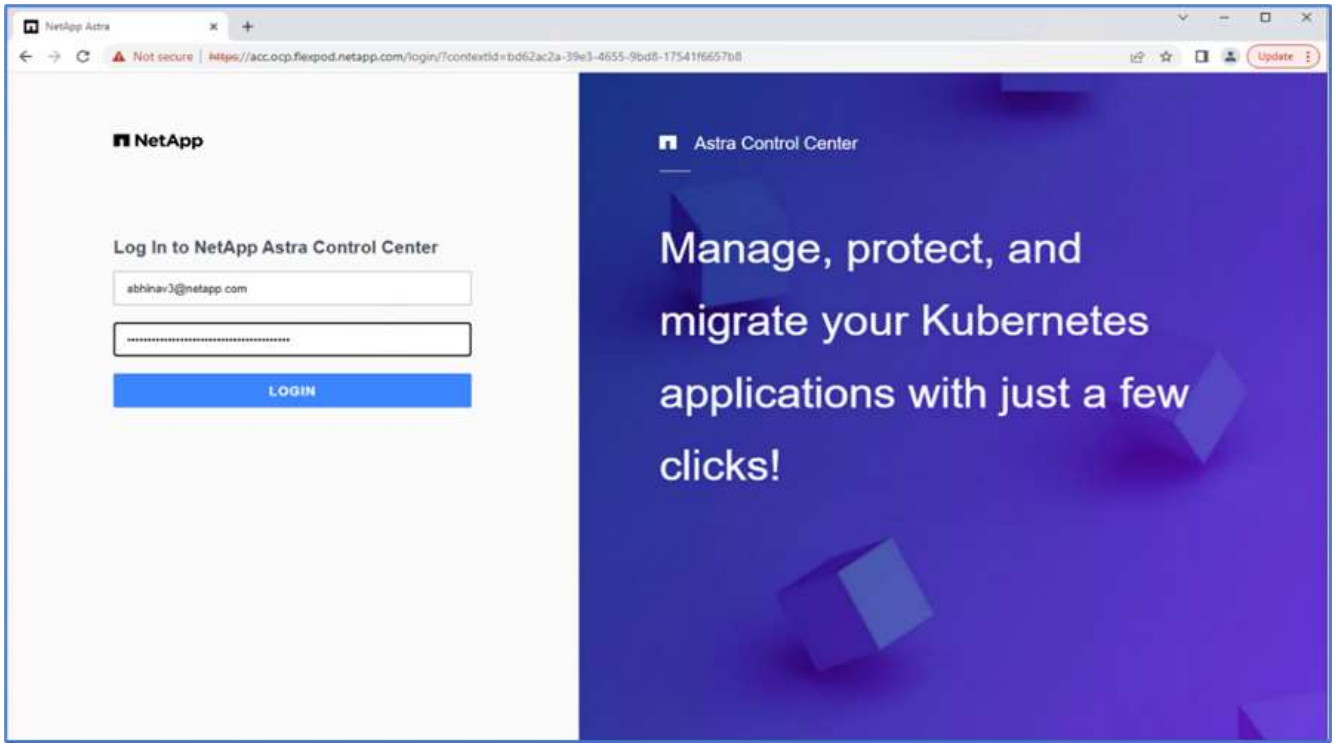
```
3m36s
telegraf-ds-pdz5j                                 1/1     Running   0
3m36s
telegraf-ds-znqtp                                 1/1     Running   0
3m36s
telegraf-rs-rt64j                                 1/1     Running   0
3m36s
telemetry-service-7dd9c74bfc-sfkzt                1/1     Running   0
6m19s
tenancy-d878b7fb6-wf8x9                           1/1     Running   0
6m37s
traefik-6548496576-5v2g6                          1/1     Running   0
98s
traefik-6548496576-g82pq                          1/1     Running   0
3m8s
traefik-6548496576-psn49                          1/1     Running   0
38s
traefik-6548496576-qrkfd                          1/1     Running   0
2m53s
traefik-6548496576-srs6r                          1/1     Running   0
98s
trident-svc-679856c67-78kbt                       1/1     Running   0
5m27s
vault-controller-747d664964-xmn6c                 1/1     Running   0
7m37s
```

每个POD的状态应为"runned"。部署系统Pod可能需要几分钟的时间。

20. 所有Pod运行时、运行以下命令以检索一次性密码。在输出的YAML版本中、检查`status.deploymentState`字段中的已部署值、然后复制`status.uuid`值。密码为`Acc-`、后跟UUID值。(Acc-UUID)。

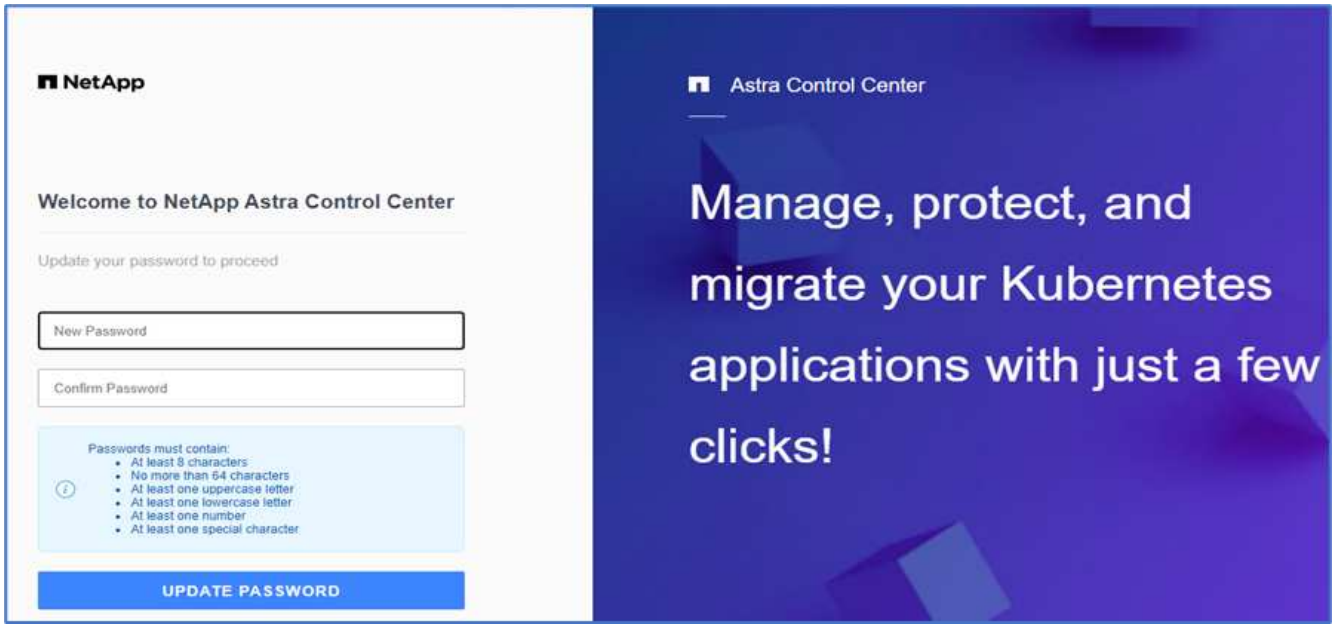
```
root@abhinav-ansible# oc get acc -o yaml -n netapp-acc-operator
```

21. 在浏览器中、使用您提供的FQDN导航到URL。
22. 使用默认用户名(即安装期间提供的电子邮件地址)和一次性密码Acc-UUID登录。



如果您输入的密码三次不正确、则管理员帐户将锁定15分钟。

23. 更改密码并继续。

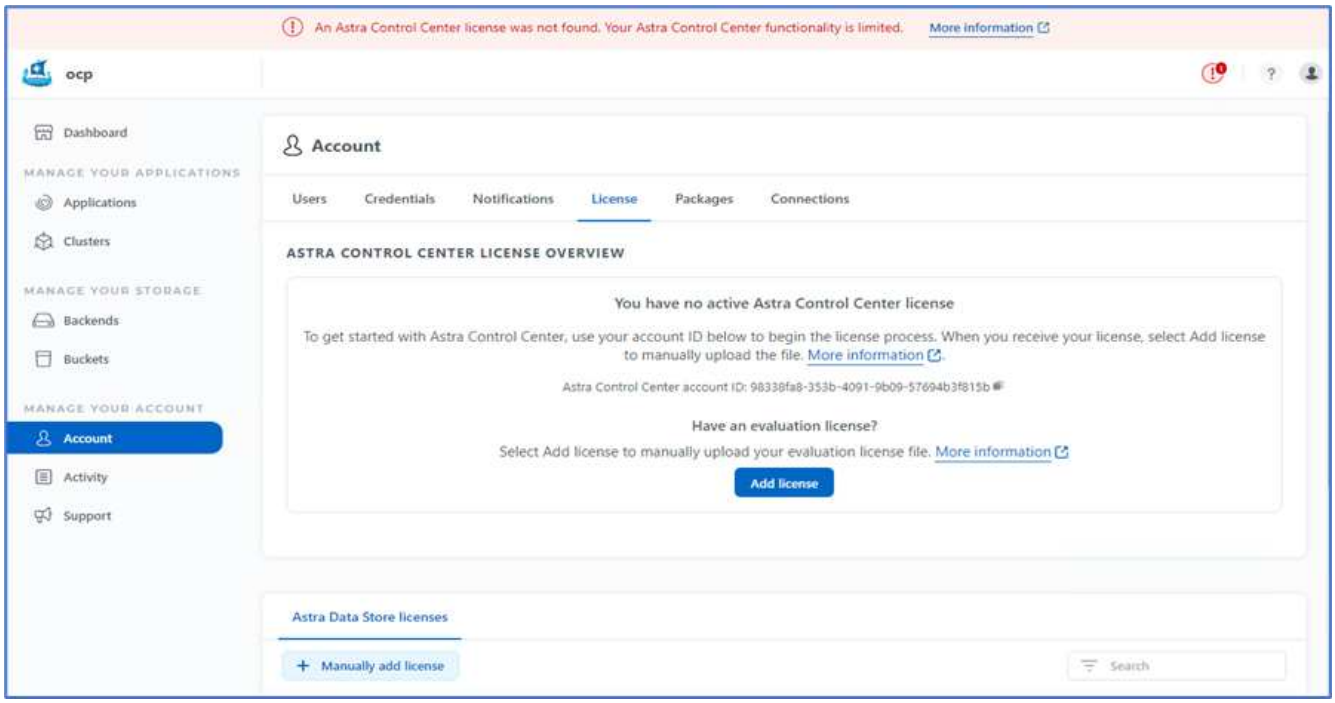


有关Astra控制中心安装的详细信息、请参见 "Astra控制中心安装概述" 页面。

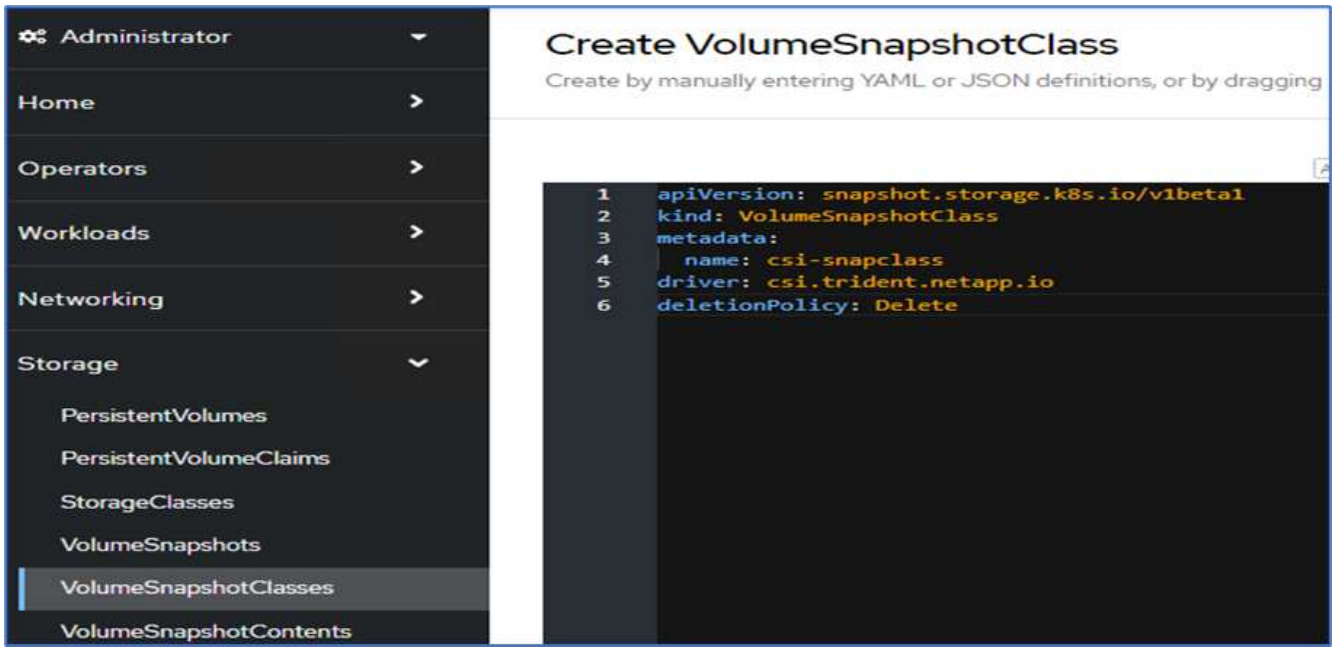
设置 **Astra** 控制中心

安装Astra控制中心后、登录到用户界面、上传许可证、添加集群、管理存储以及添加存储分段。

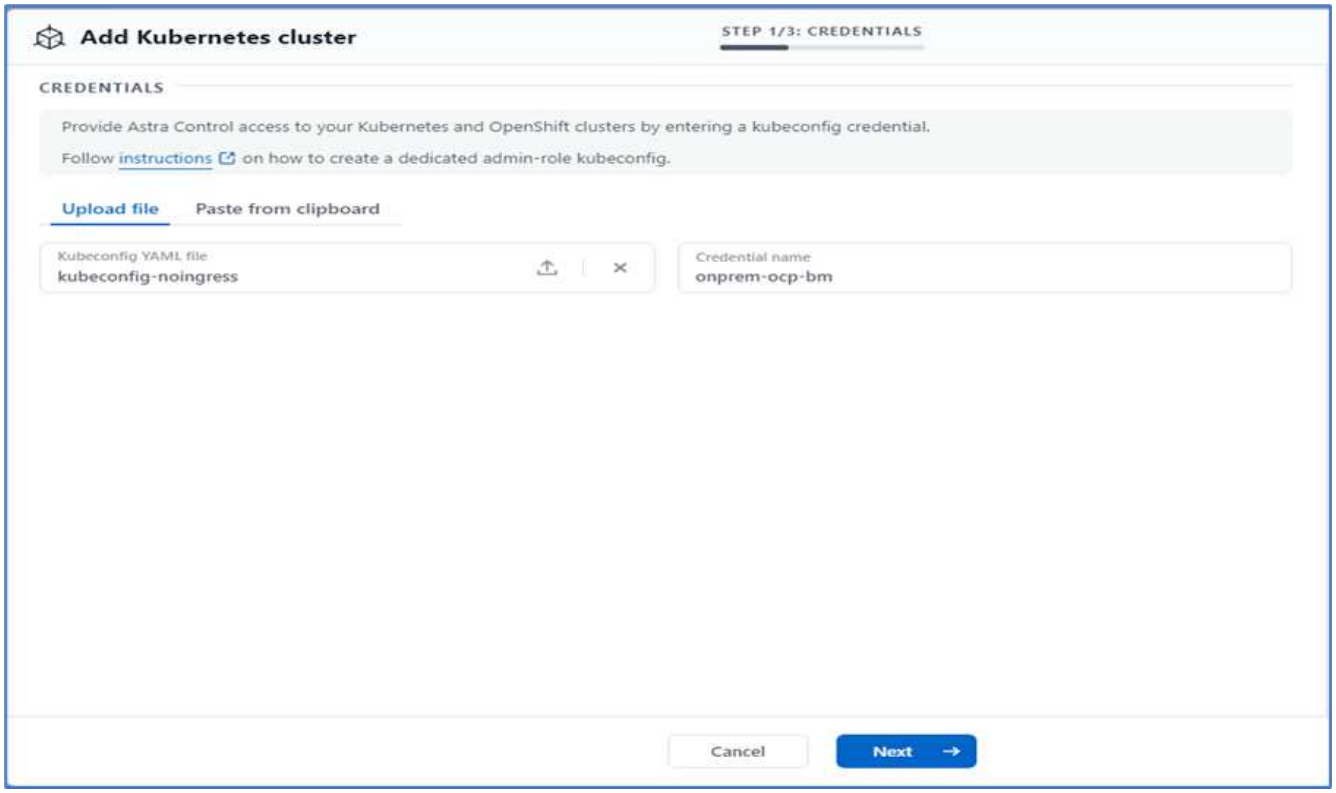
1. 在主页上的Account下、转到License选项卡并选择Add License以上传Astra许可证。



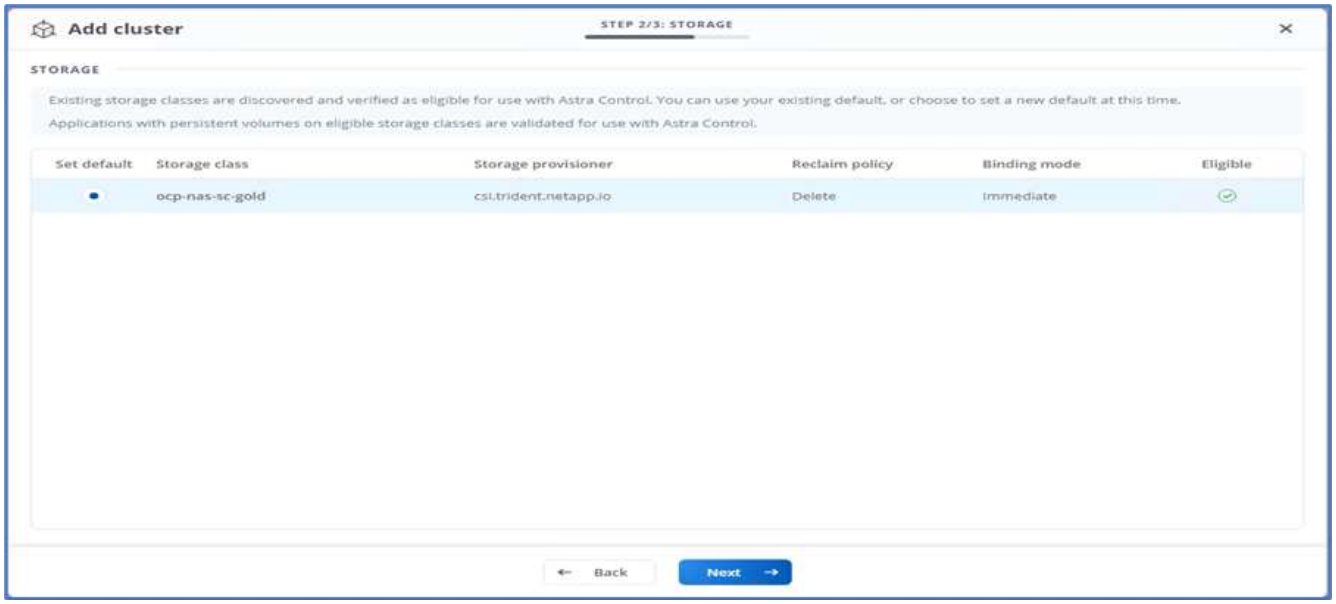
2. 在添加OpenShift集群之前、请从OpenShift Web控制台创建Astra Trident卷快照类。卷快照类配置了`csi.trident.netapp.io`驱动程序。



3. 要添加Kubernetes集群、请转到主页上的Clusters、然后单击Add Kubernetes Cluster。然后上传集群的`kubeconfig`文件并提供凭据名称。单击下一步。



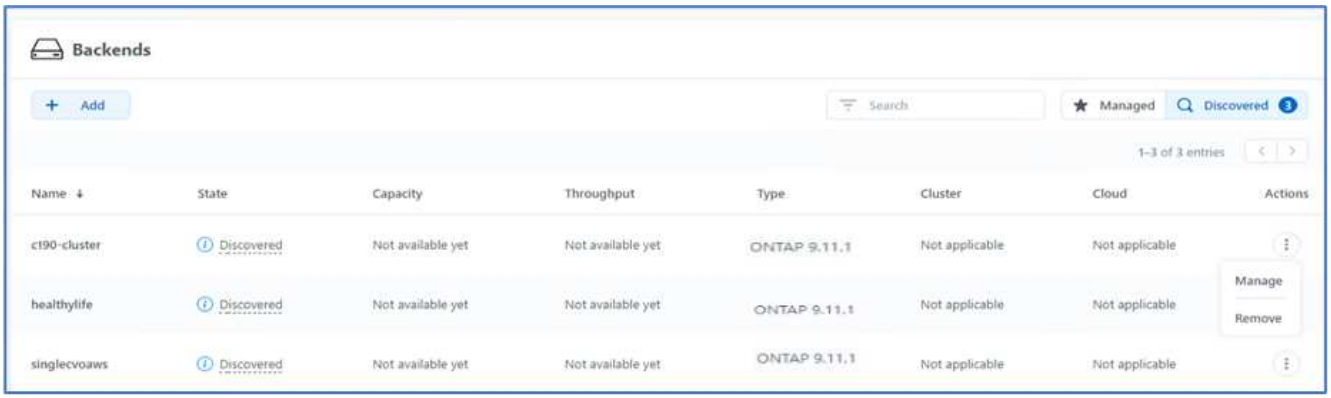
4. 系统会自动发现现有存储类。选择默认存储类、单击下一步、然后单击添加集群。



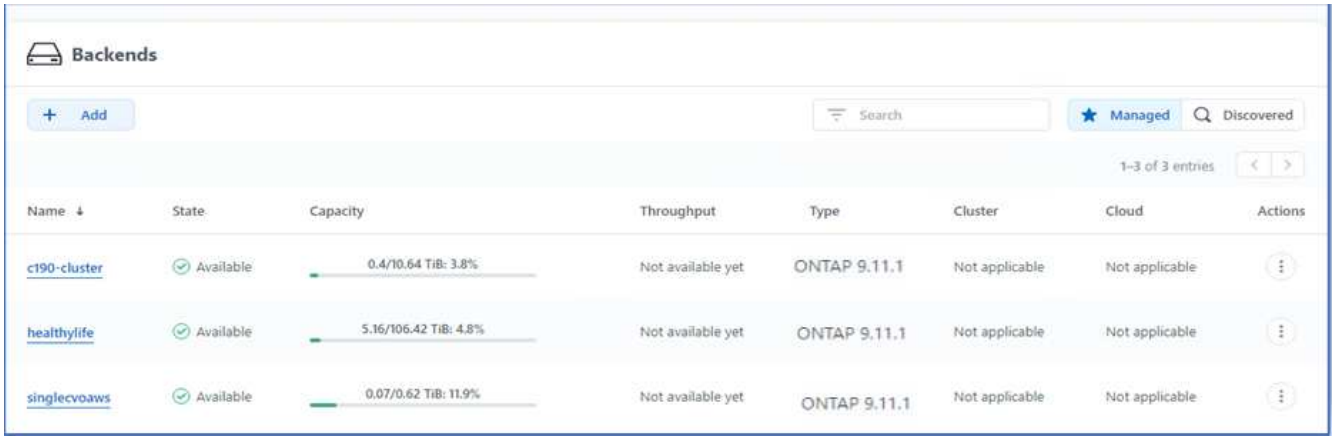
5. 只需几分钟即可添加集群。要添加其他OpenShift容器平台集群、请重复步骤1–4。

> 要将其他OpenShift操作环境添加为托管计算资源、请确保使用Astra Trident "VolumeSnapshotClass对象" 已定义。

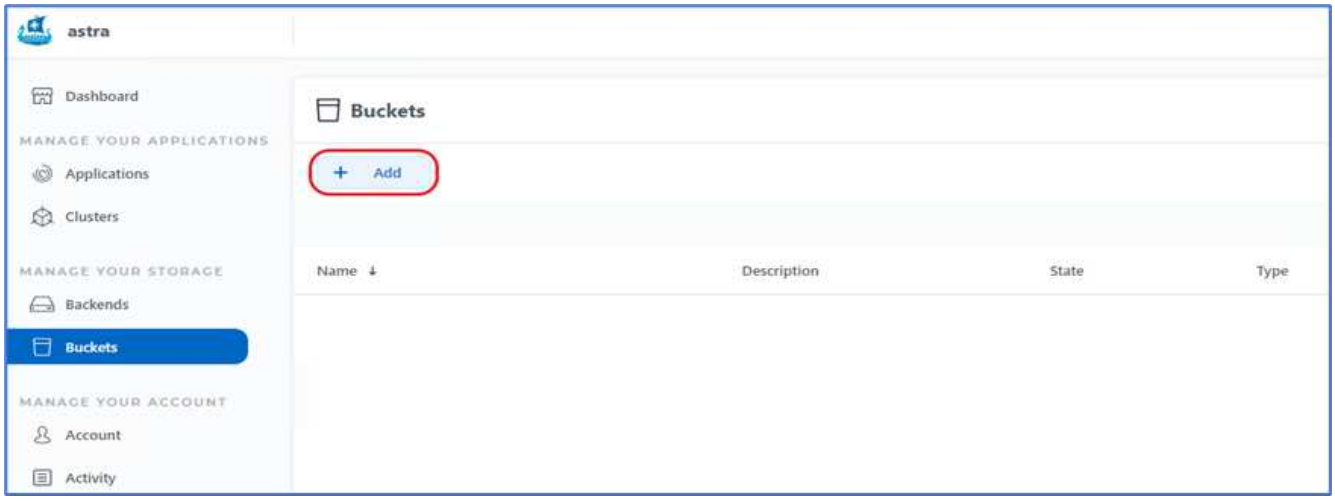
6. 要管理存储、请转至后端、单击"Actions against the backend that you would like to manage"下的三个点。单击Manage。



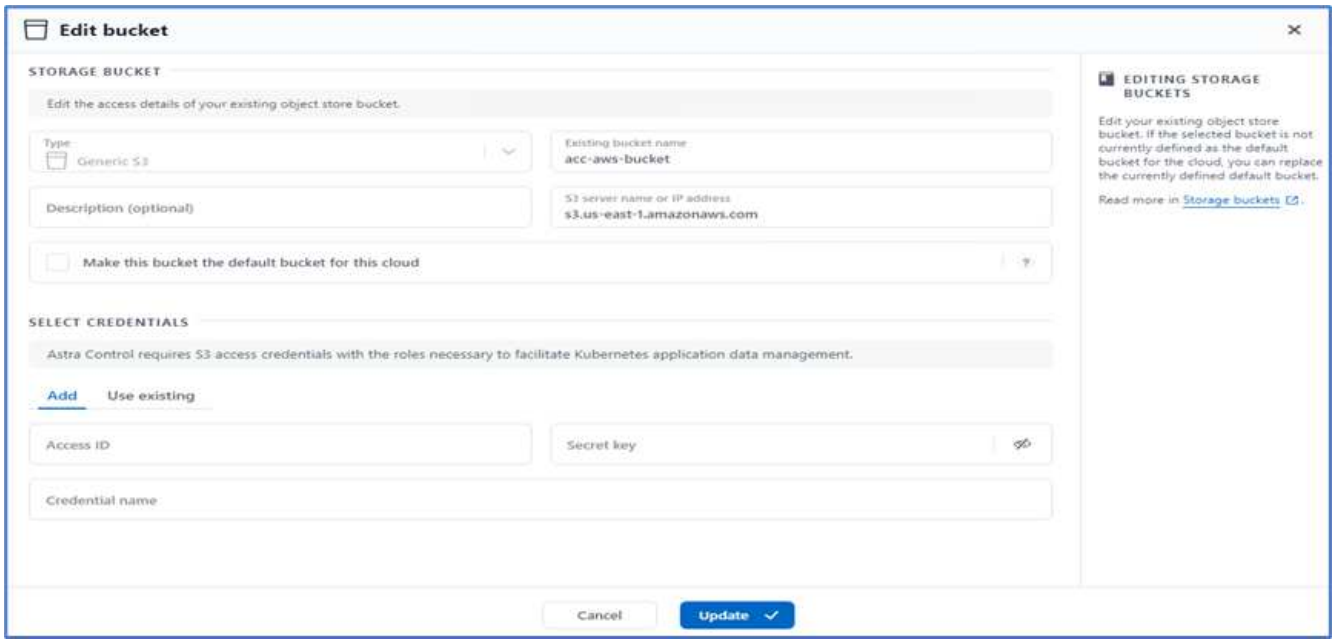
7. 提供ONTAP 凭据、然后单击下一步。查看相关信息、然后单击受管。后端应类似于以下示例。



8. 要将存储分段添加到Astra Control、请选择Bucket并单击Add。

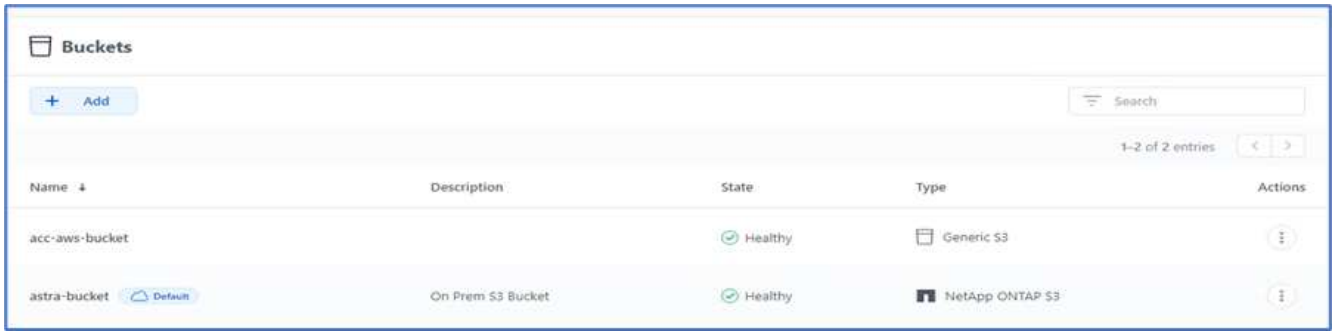


9. 选择存储分段类型并提供存储分段名称、S3服务器名称或IP地址和S3凭据。单击更新。



在此解决方案 中、AWS S3和ONTAP S3存储分段均已使用。您也可以使用StorageGRID。

存储分段状态应为运行状况良好。



作为向Astra控制中心注册Kubernetes集群以实现应用程序感知型数据管理的一部分、Astra Control会自动创建角色绑定和NetApp监控命名空间、以便从应用程序Pod和工作节点收集指标和日志。将一个受支持的基于ONTAP的存储类设置为默认值。

你先请 "将集群添加到 Astra Control 管理中"、您可以在集群上安装应用程序(在Astra Control之外)、然后转到Astra Control中的应用程序页面来管理这些应用程序及其资源。有关使用Astra管理应用程序的详细信息、请参见 "应用程序管理要求"。

"接下来：解决方案 验证概述。"

解决方案验证

概述

"先前版本：在OpenShift容器平台上安装Astra Control Center。"

在本节中、我们将回顾解决方案 以及一些使用情形：

- 将有状态应用程序从远程备份还原到云中运行的另一个OpenShift集群。
- 将有状态应用程序还原到OpenShift集群中的同一命名空间。
- 通过从一个FlexPod 系统(OpenShift容器平台裸机)克隆到另一个FlexPod 系统(VMware上的OpenShift容器平台)来实现应用程序移动性。

值得注意的是、此解决方案 仅验证了少数使用情形。此验证并不代表Astra控制中心的全部功能。

"接下来：使用远程备份恢复应用程序。"

通过远程备份恢复应用程序

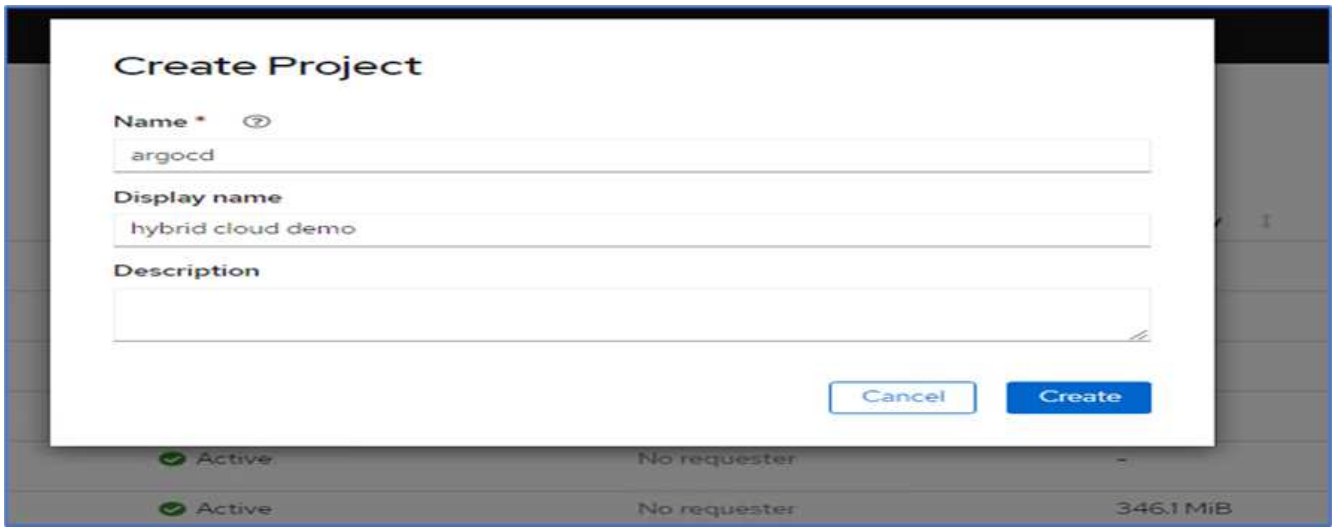
"上一步：解决方案 验证概述。"

借助Astra、您可以进行完整的应用程序一致的备份、用于将应用程序及其数据还原到在内部数据中心或公有 云中运行的不同Kubernetes集群。

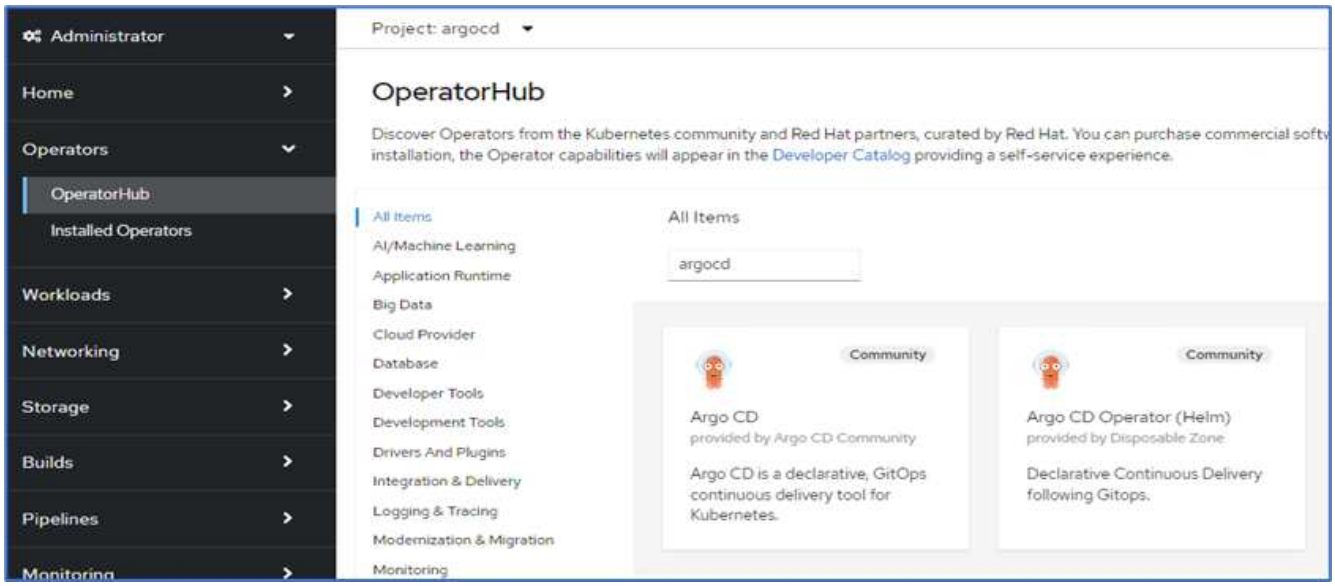
要验证应用程序恢复是否成功、请模拟FlexPod 系统上运行的应用程序的内部故障、并使用远程备份将应用程序还原到云中运行的K8s集群。

此示例应用程序是一个使用MySQL作为数据库的价目表应用程序。为了实现部署自动化、我们使用了 "Argo CD" 工具。Argo CD是一款适用于Kubernetes的声明性GitOps持续交付工具。

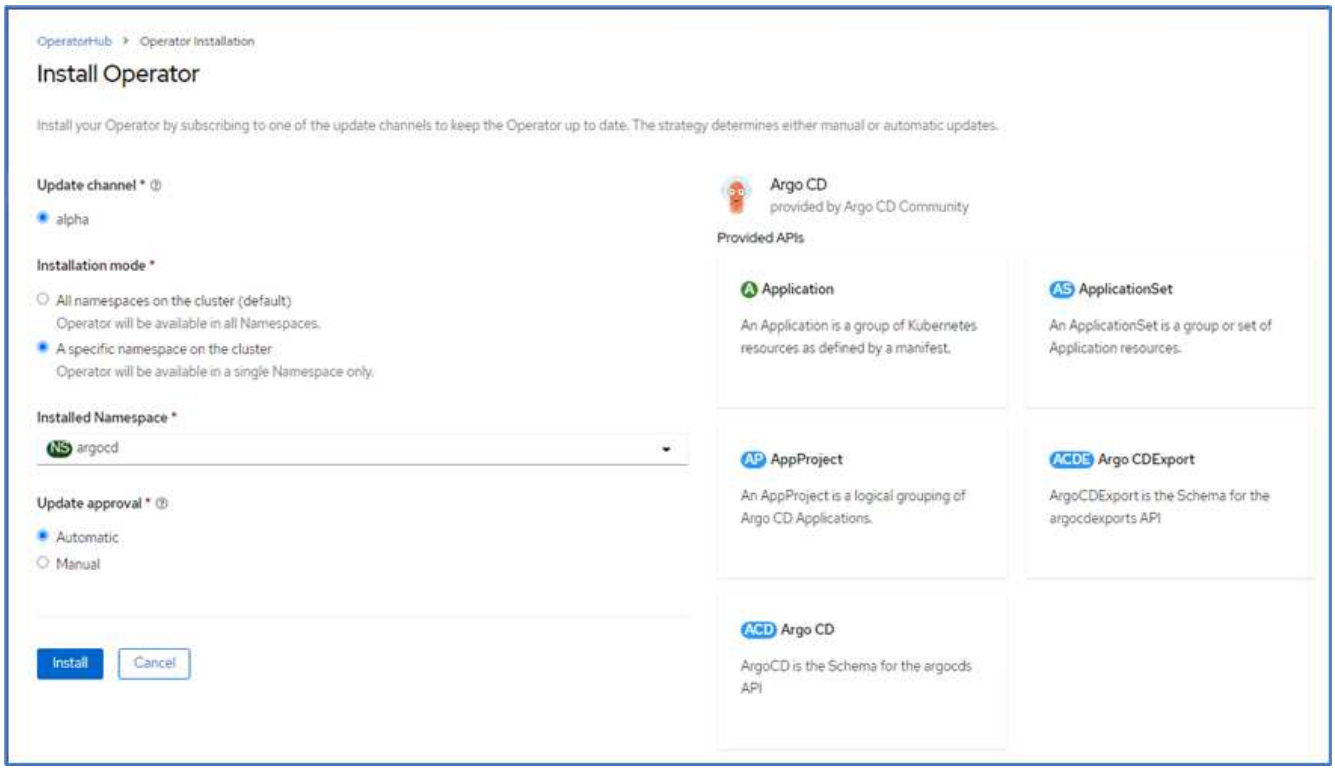
1. 登录到内部OpenShift集群并创建一个名为`argocd`的新项目。



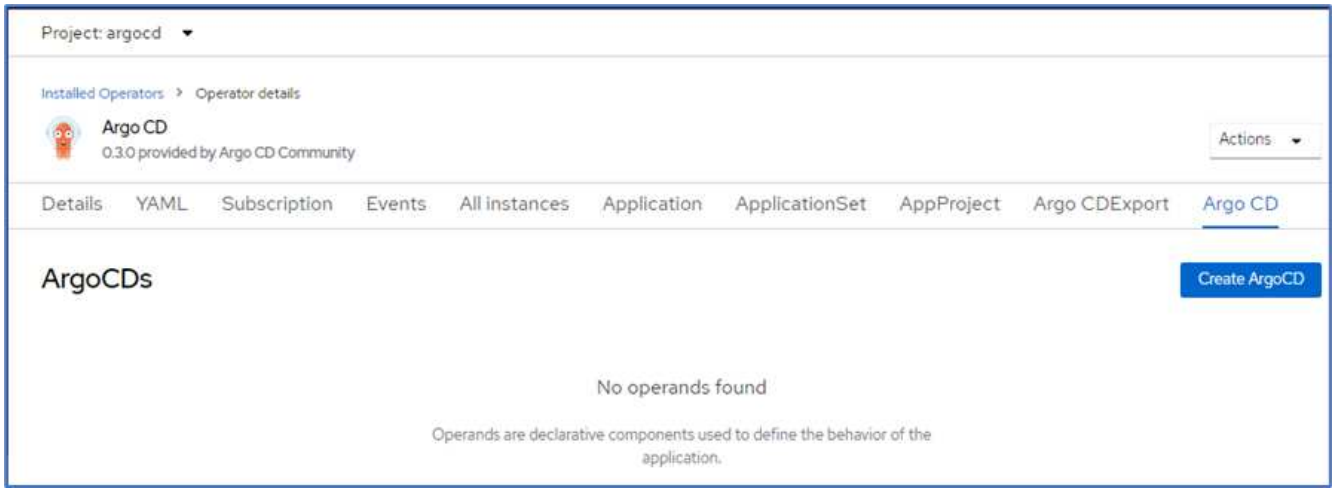
2. 在OperatorHub中、搜索`argocd`并选择Argo CD operator。



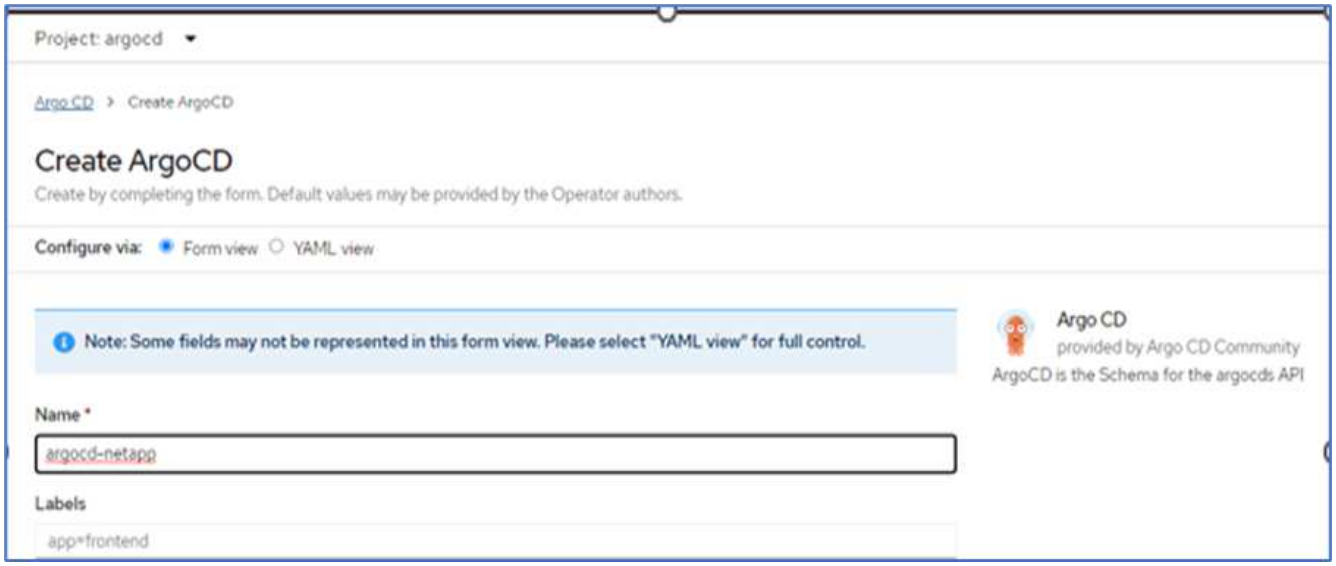
3. 在`argocd`命名空间中安装操作符。



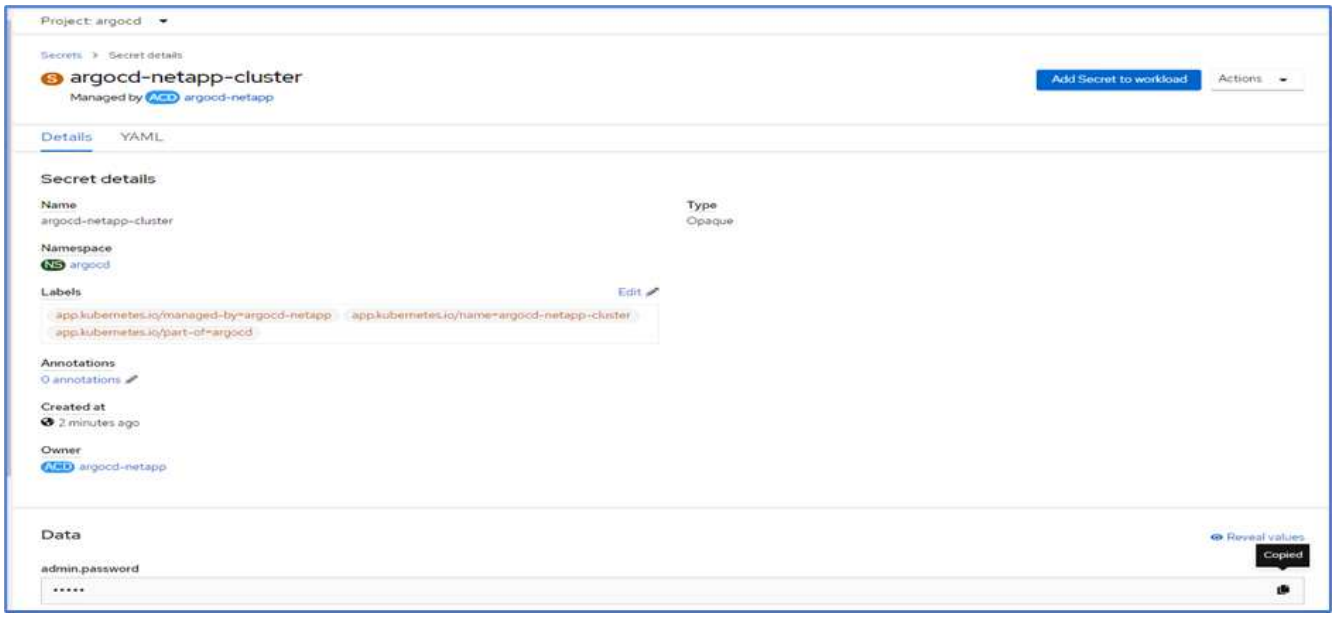
4. 转到运算符并单击创建ArgoCD。



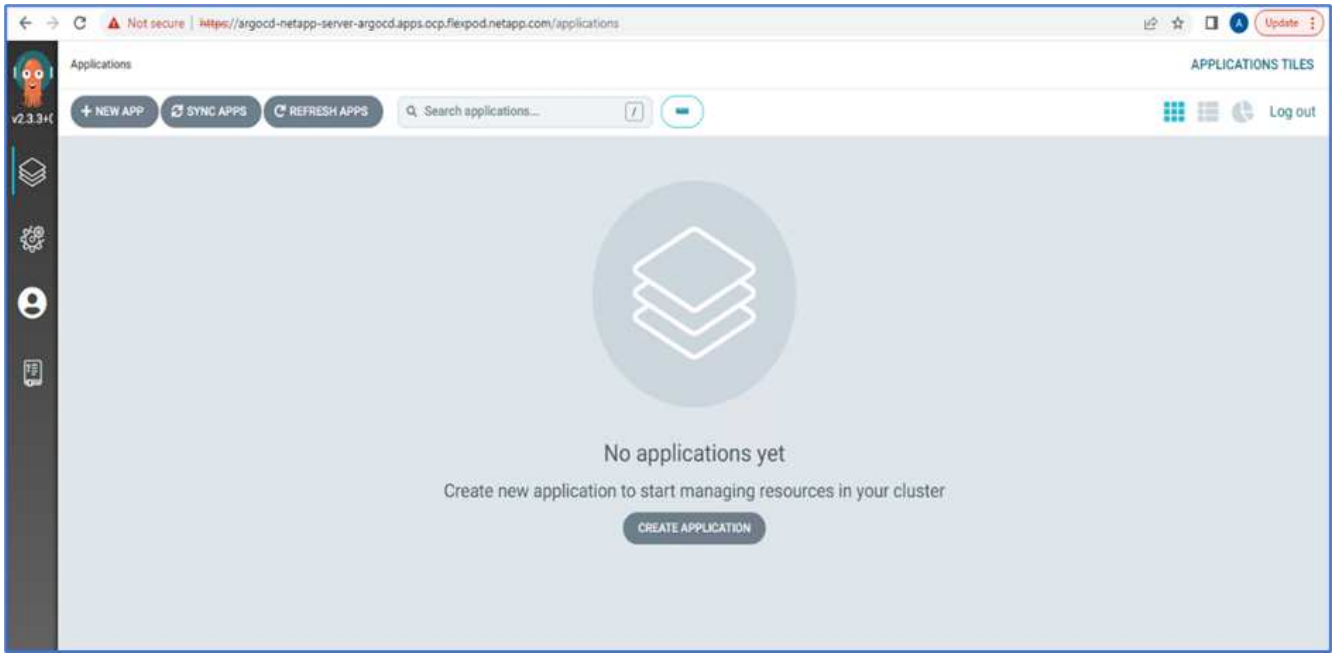
5. 要在`argocd`项目中部署Argo CD实例、请提供一个名称、然后单击创建。



6. 要登录到Argo CD、默认用户为admin、密码位于名为`argocd-netapp-cluster`的机密文件中。



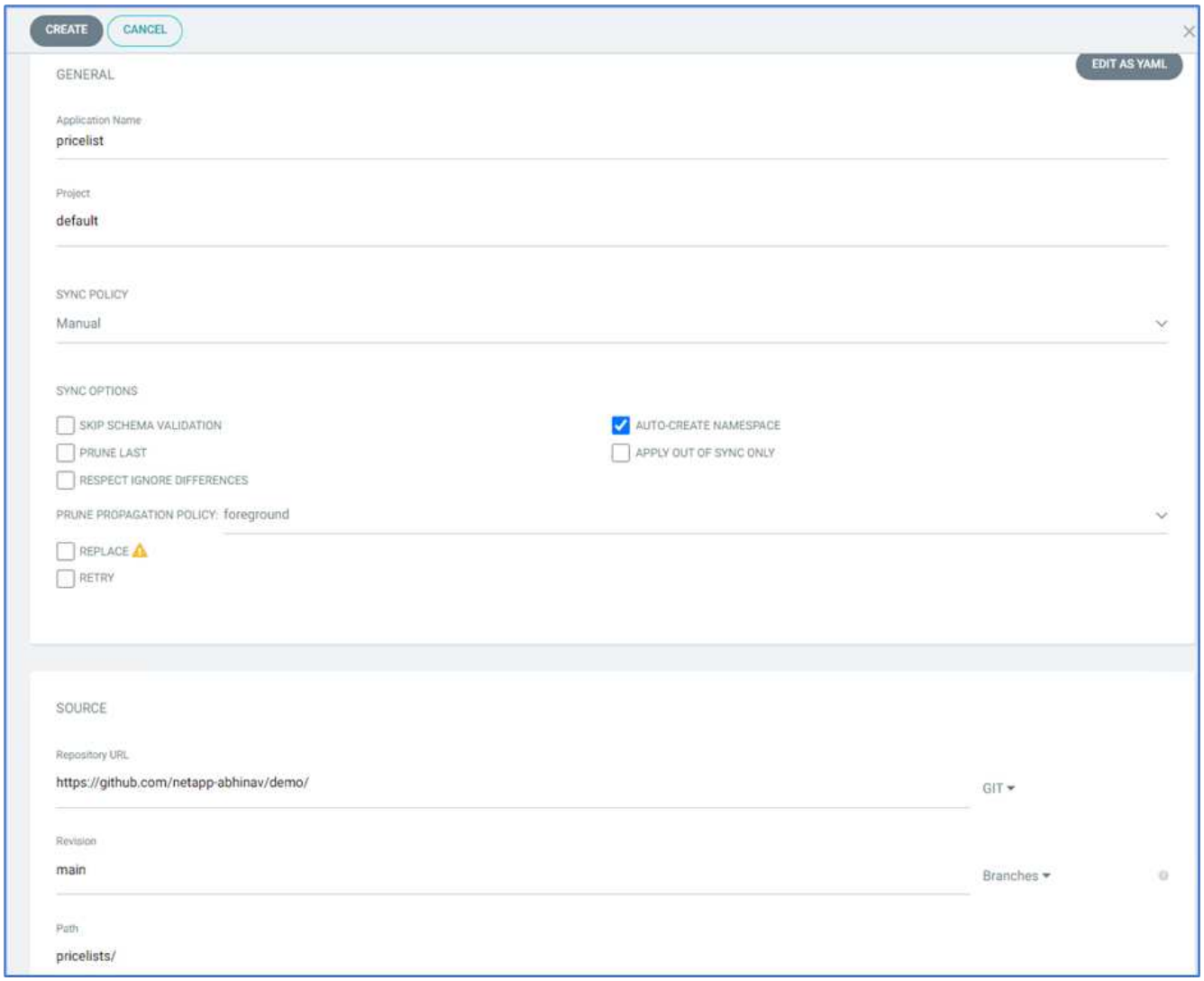
7. 从侧面菜单中、选择路由>位置、然后单击`argocd` routes的URL。输入用户名和密码。



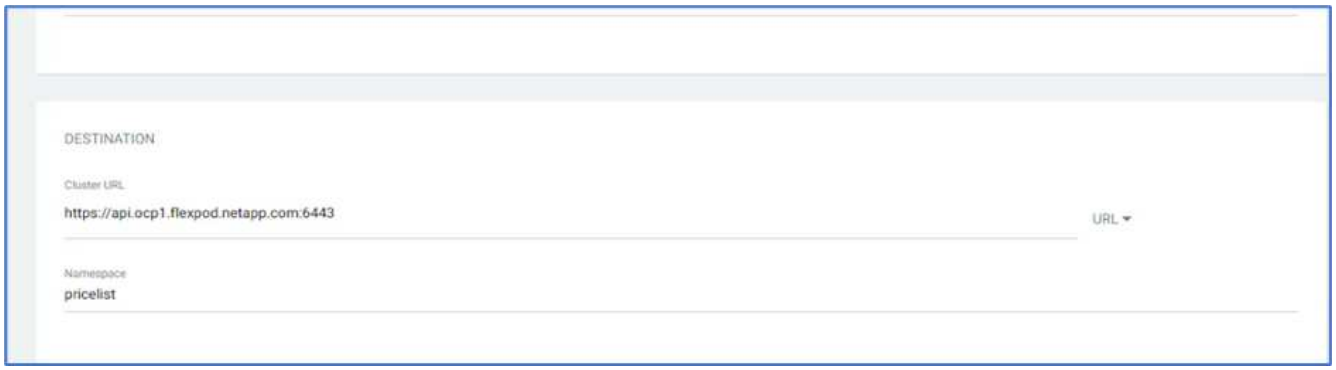
8. 通过CLI将内部OpenShift集群添加到Argo CD。

```
####Login to Argo CD####
abhinav3@abhinav-ansible$ argocd-linux-amd64 login argocd-netapp-server-
argocd.apps.ocp.flexpod.netapp.com --insecure
Username: admin
Password:
'admin:login' logged in successfully
Context'argocd-netapp-server-argocd.apps.ocp.flexpod.netapp.com' updated
####List the On-Premises OpenShift cluster####
abhinav3@abhinav-ansible$ argocd-linux-amd64 cluster add
ERRO[0000] Choose a context name from:
CURRENT  NAME
CLUSTER              SERVER
*        default/api-ocp-flexpod-netapp-com:6443/abhinav3
api-ocp-flexpod-netapp-com:6443
https://api.ocp.flexpod.netapp.com:6443
         default/api-ocp1-flexpod-netapp-com:6443/abhinav3
api-ocp1-flexpod-netapp-com:6443
https://api.ocp1.flexpod.netapp.com:6443
####Add On-Premises OpenShift cluster###
abhinav3@abhinav-ansible$ argocd-linux-amd64 cluster add default/api-
ocp1-flexpod-netapp-com:6443/abhinav3
WARNING: This will create a service account `argocd-manager` on the
cluster referenced by context `default/api-ocp1-flexpod-netapp-
com:6443/abhinav3` with full cluster level admin privileges. Do you want
to continue [y/N]? y
INFO[0002] ServiceAccount "argocd-manager" already exists in namespace
"kube-system"
INFO[0002] ClusterRole "argocd-manager-role" updated
INFO[0002] ClusterRoleBinding "argocd-manager-role-binding" updated
Cluster 'https://api.ocp1.flexpod.netapp.com:6443' added
```

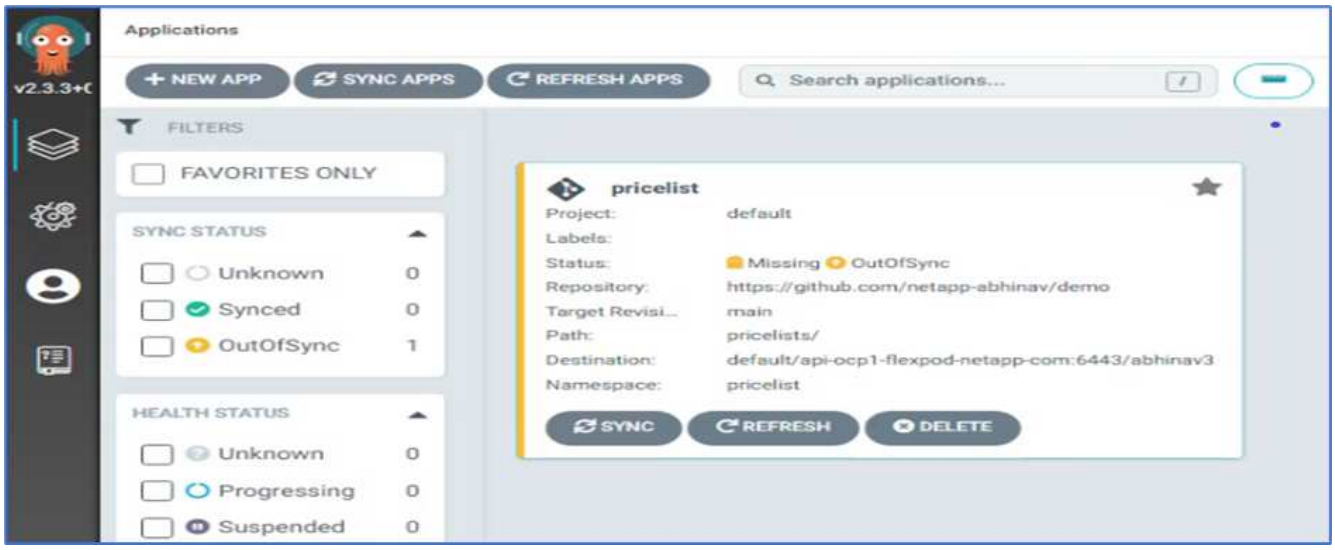
9. 在ArgoCD UI中、单击新应用程序并输入有关应用程序名称和代码存储库的详细信息。



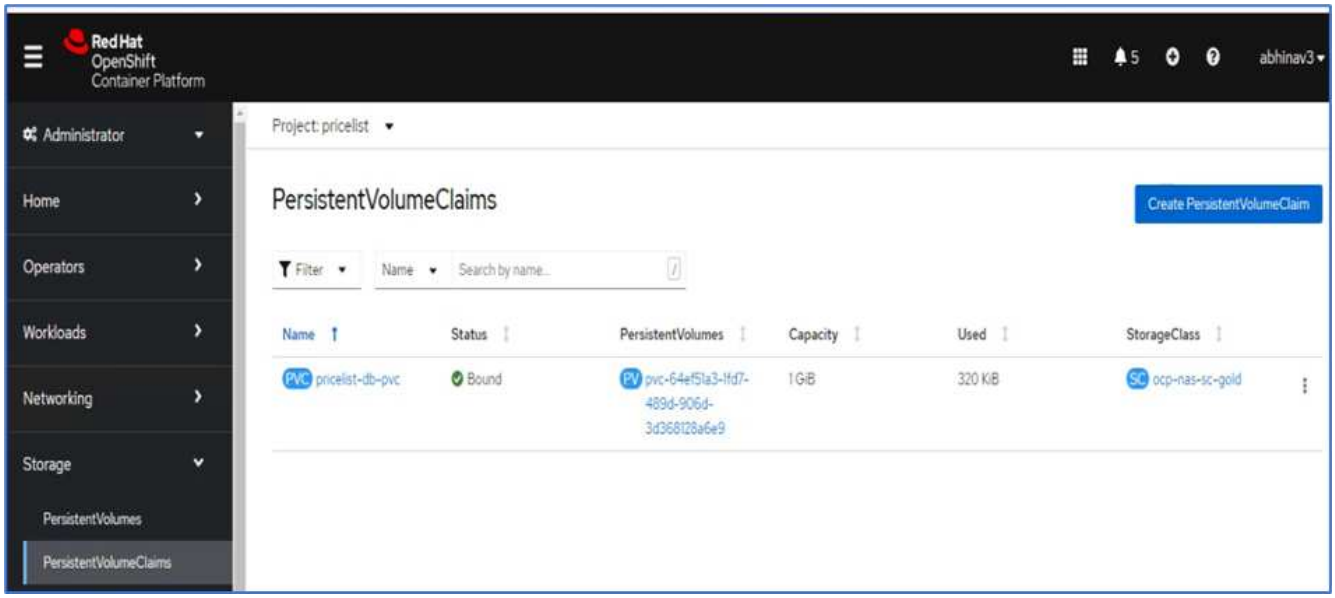
10. 输入要随命名空间一起部署应用程序的OpenShift集群。



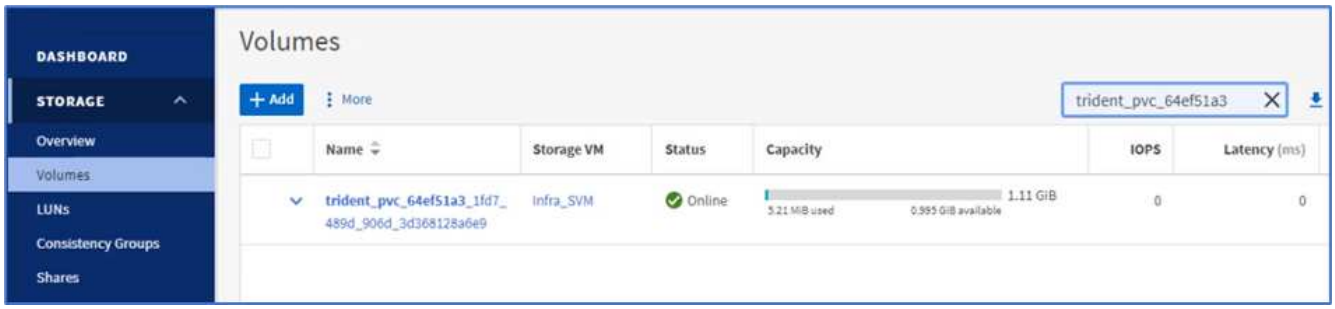
11. 要在内部OpenShift集群上部署此应用、请单击同步。



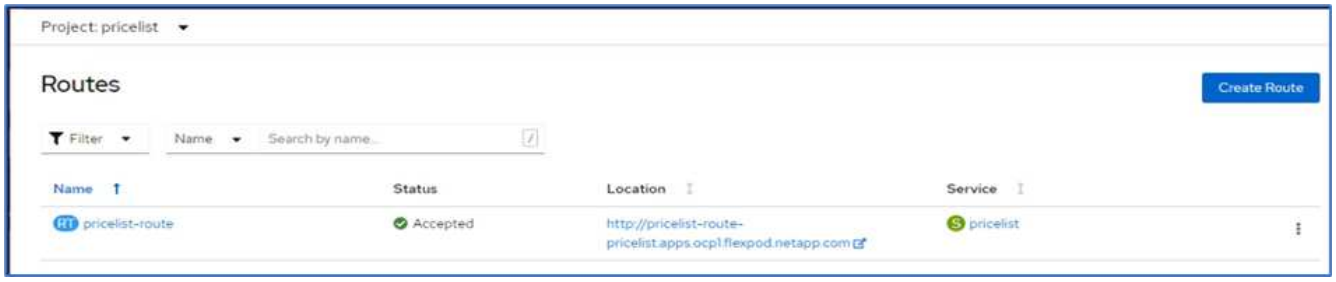
12. 在OpenShift容器平台控制台中、转至项目价目表、然后在存储下验证PVC的名称和大小。



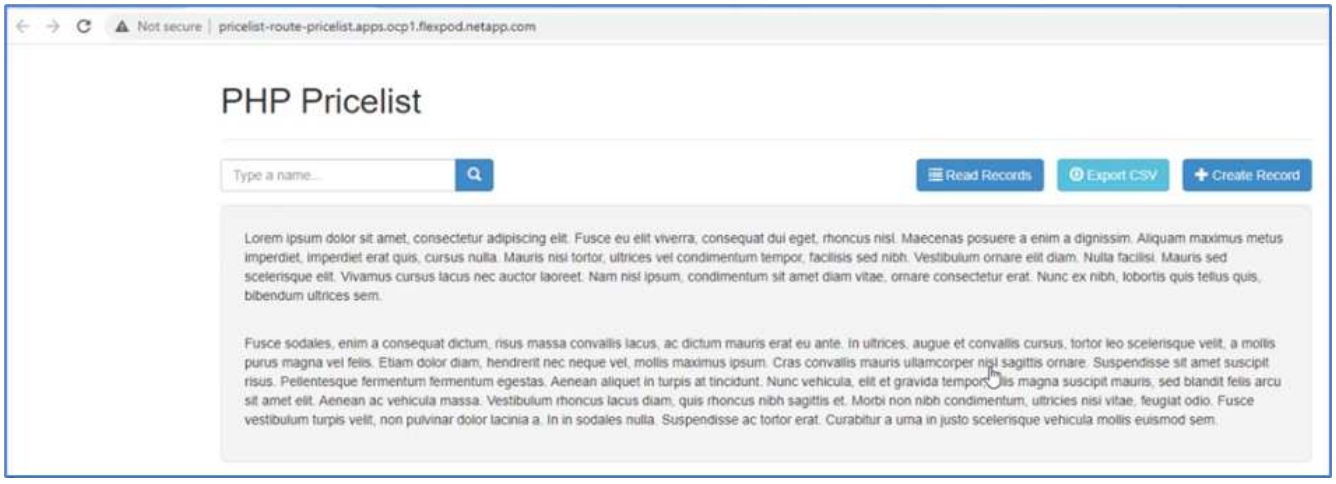
13. 登录到System Manager并验证PVC。



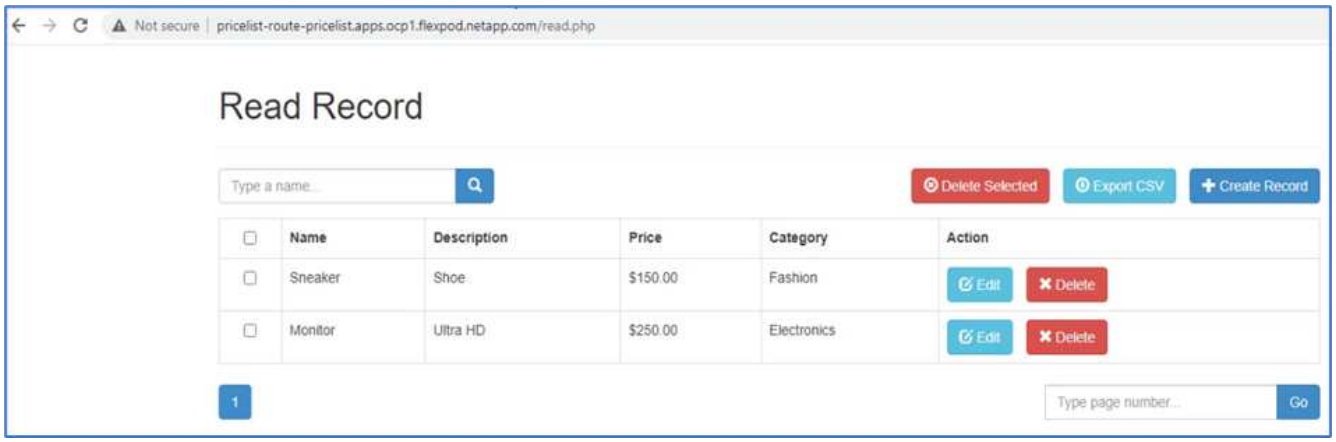
14. Pod运行后、从侧面菜单中选择Networking > routes、然后单击Location下的URL。



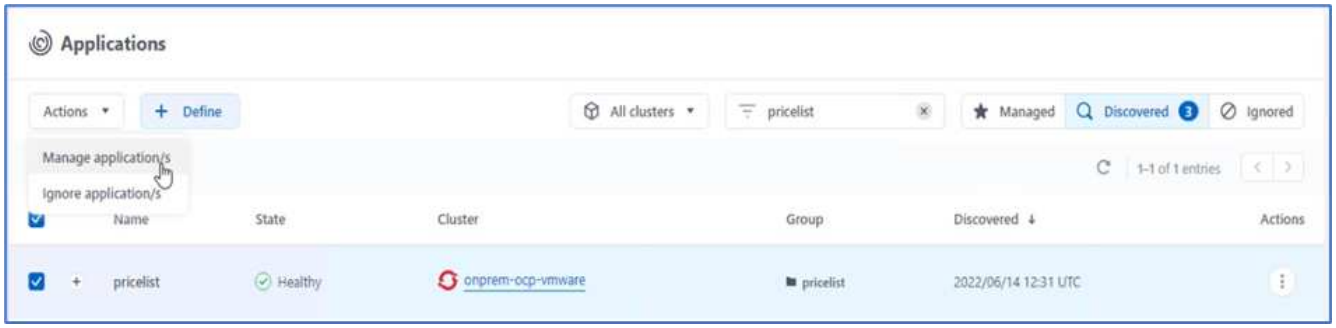
15. 此时将显示Pricelist应用程序主页。



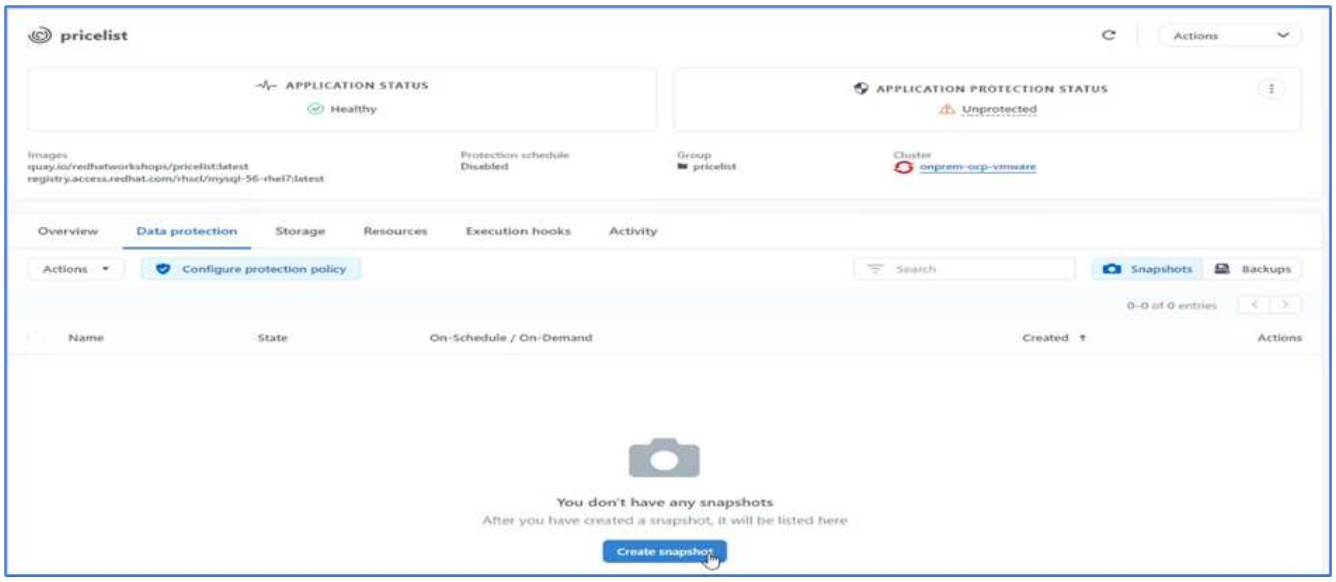
16. 在网页上创建一些记录。



17. 此应用程序会在Astra控制中心中发现。要管理此应用程序、请转到"应用程序">"已发现"、选择"价目表"应用程序、然后单击"操作"下的"管理应用程序"。

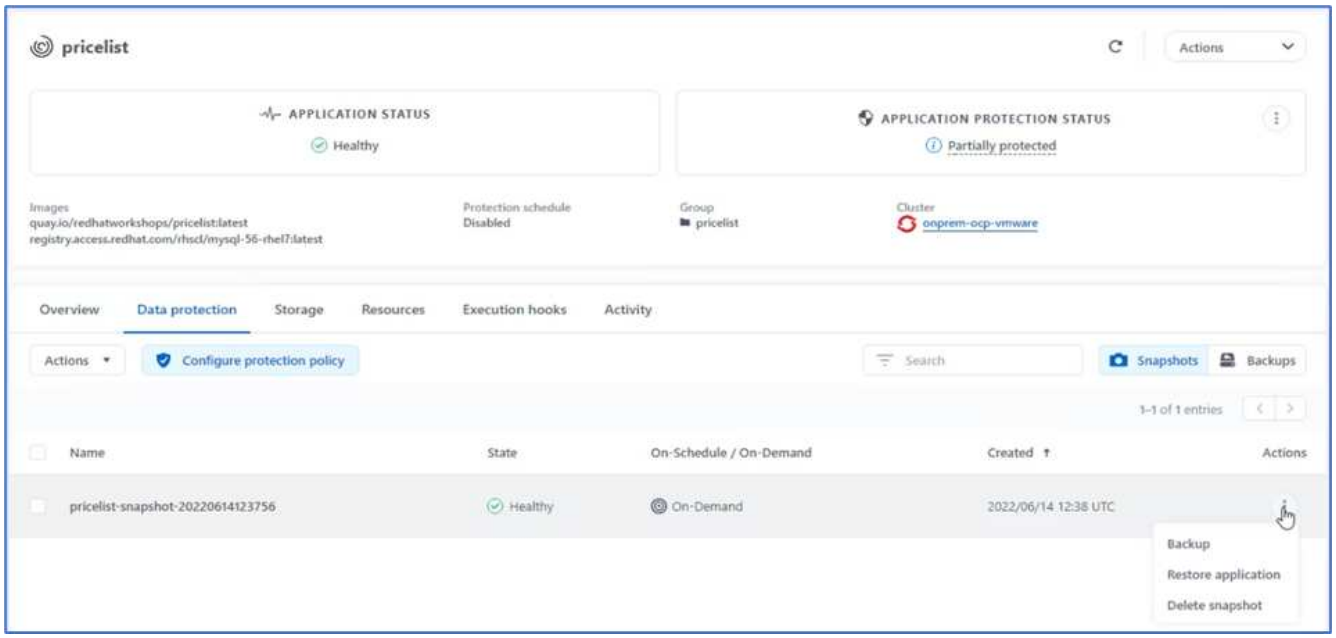


18. 单击Pricelist应用程序、然后选择Data Protection。此时、不应存在快照或备份。单击Create Snapshot以创建按需快照。

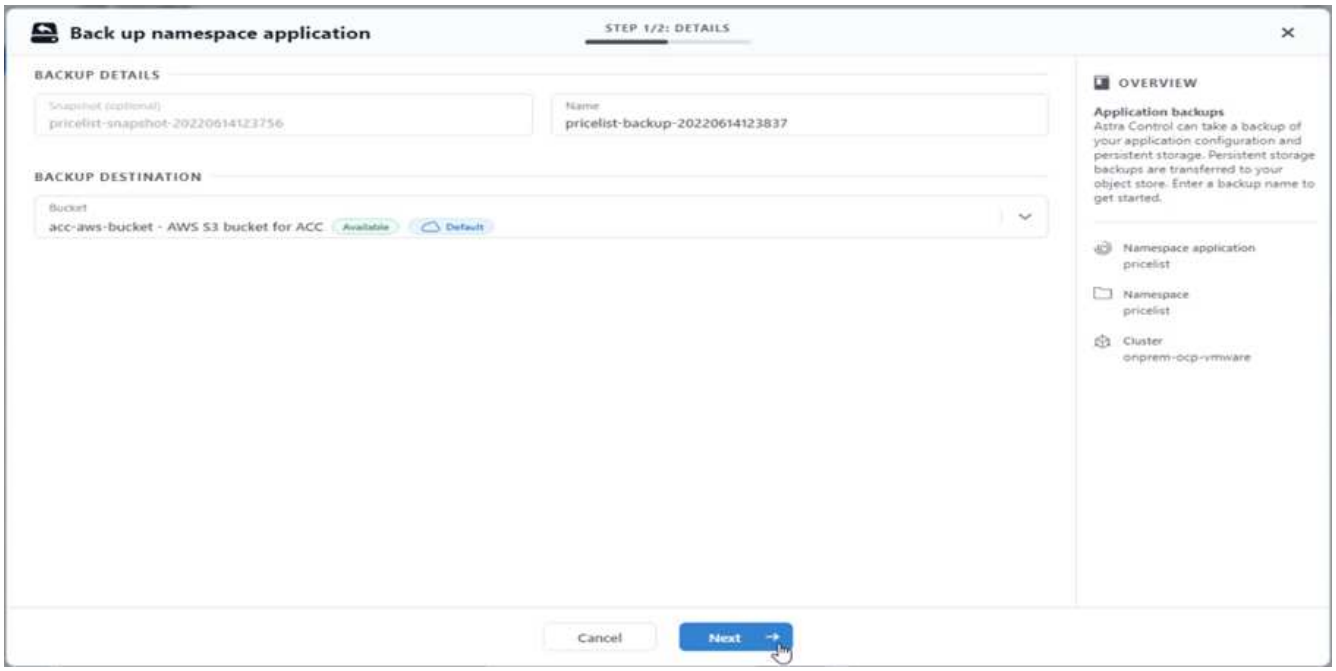


NetApp Astra控制中心既支持按需快照、也支持计划快照和备份。

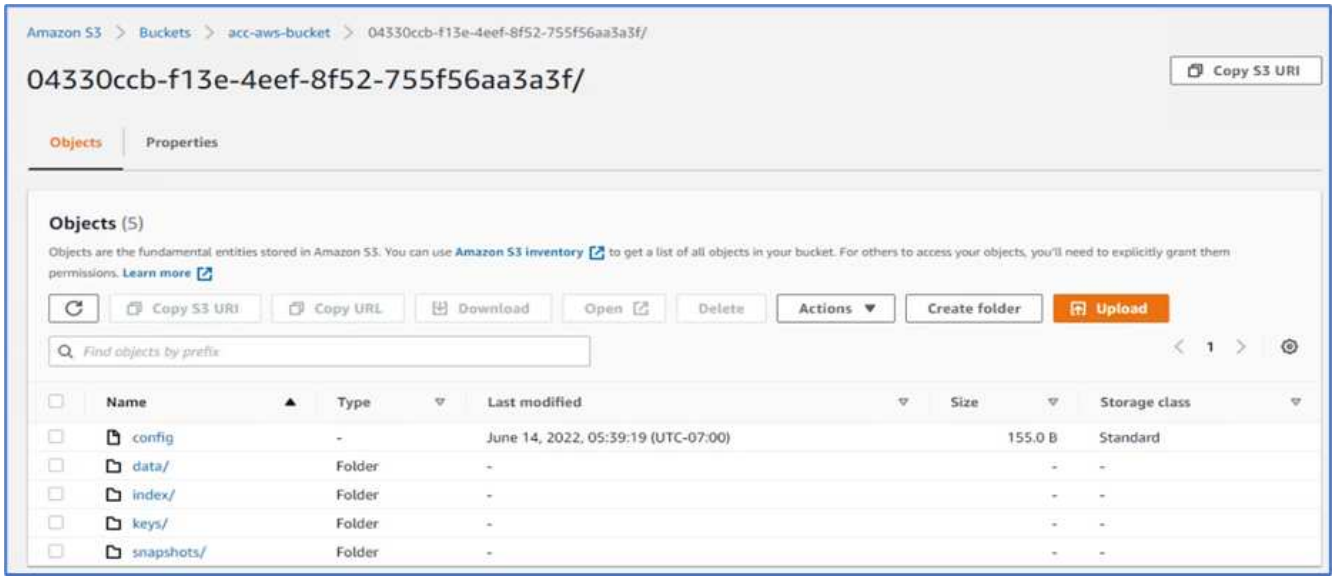
19. 创建快照并使其运行状况良好后、使用该快照创建远程备份。此备份存储在S3存储分段中。



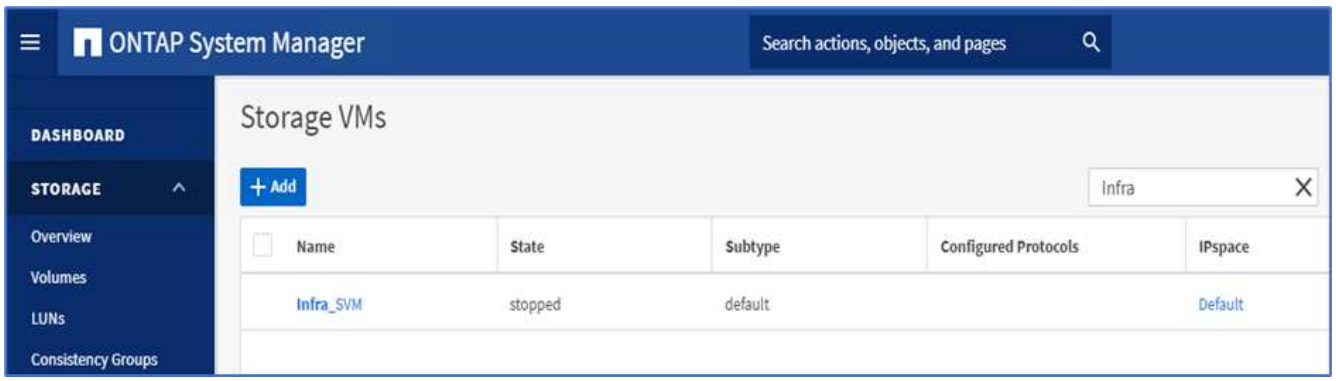
20. 选择AWS S3存储分段并启动备份操作。



21. 备份操作应在AWS S3存储分段中创建一个包含多个对象的文件夹。



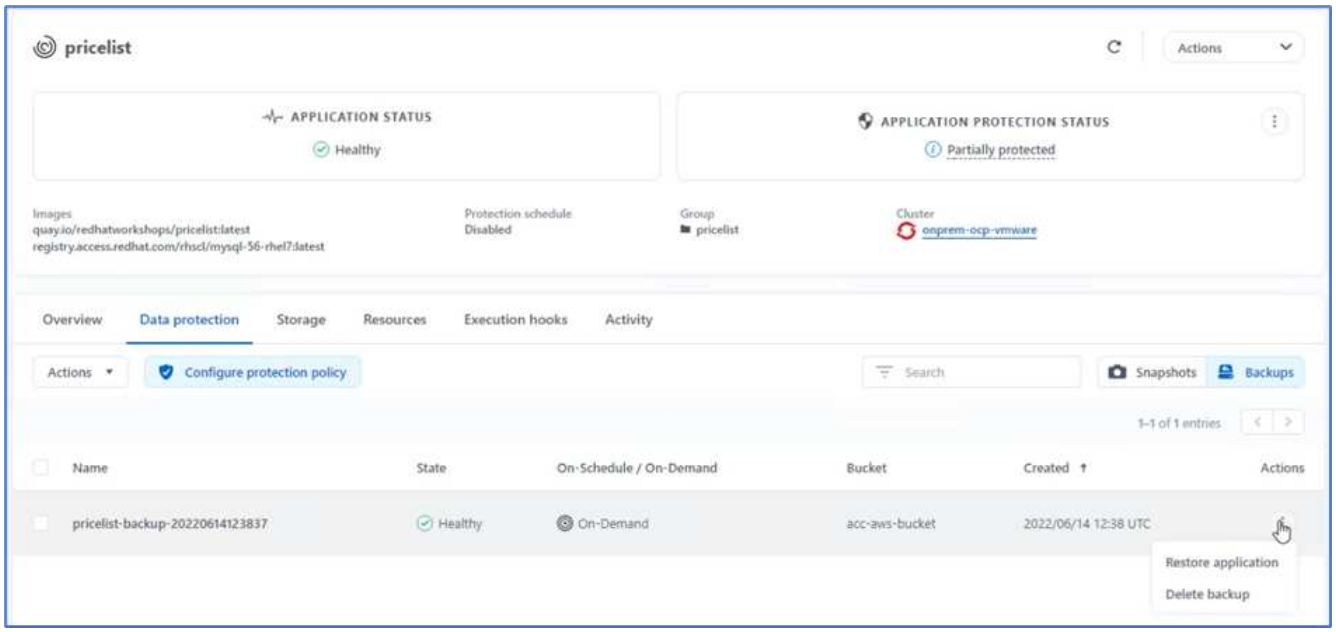
22. 远程备份完成后、通过停止托管PV后备卷的Storage Virtual Machine (SVM)来模拟内部灾难。



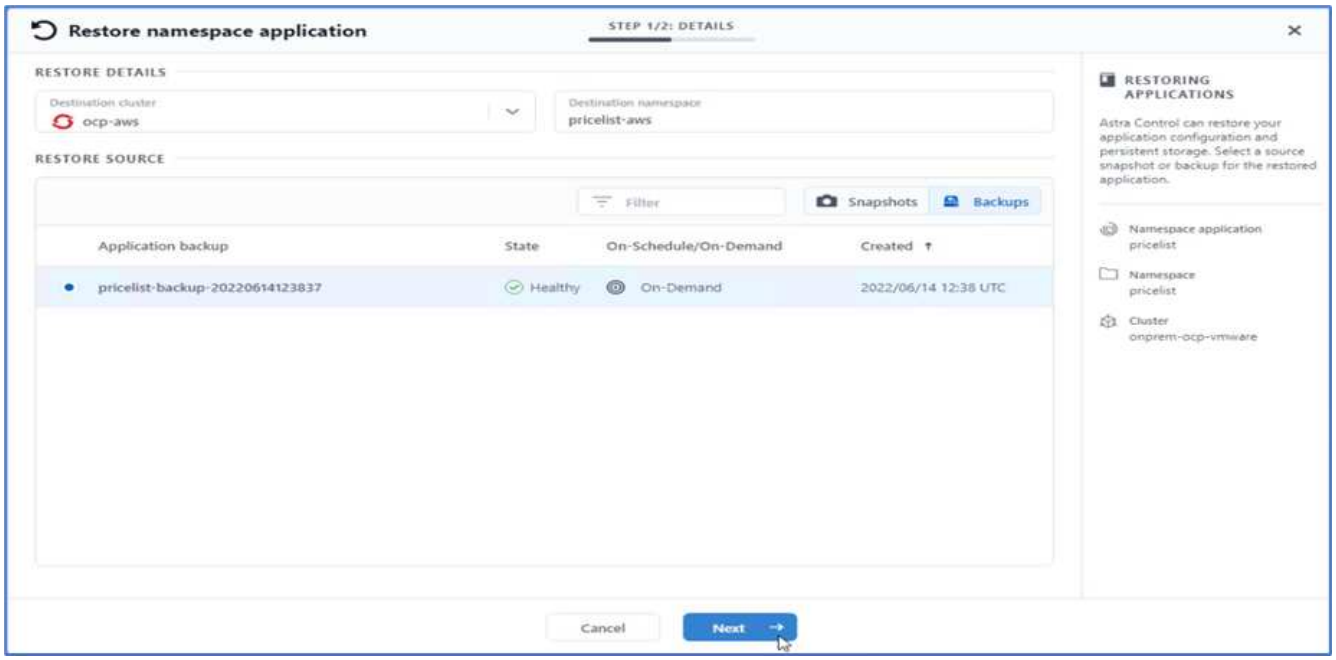
23. 刷新网页以确认中断。此网页不可用。

正如预期、网站已关闭、因此、让我们使用Astra快速从远程备份恢复应用程序、并将其恢复到AWS中运行的OpenShift集群。

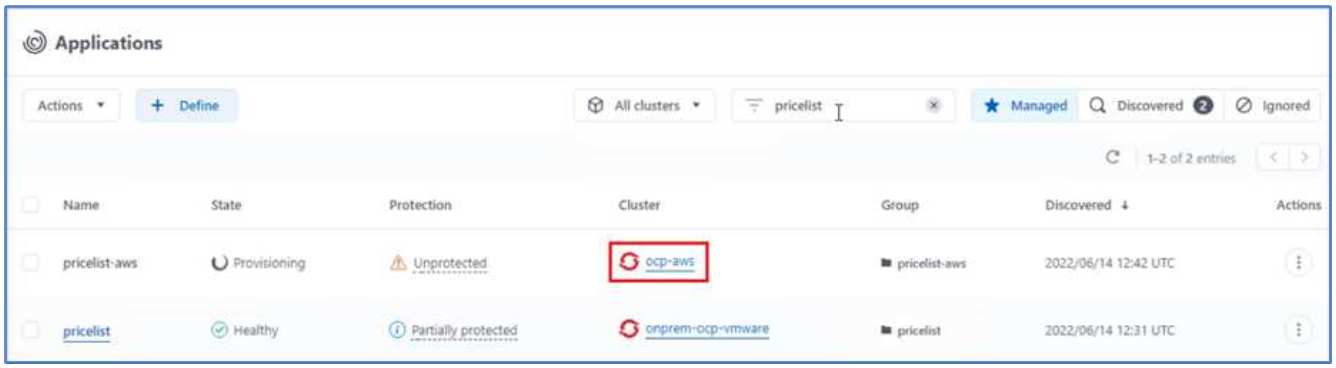
24. 在Astra Control Center中、单击Pricelist应用程序、然后选择Data Protection > Backups。选择备份、然后单击操作下的还原应用程序。



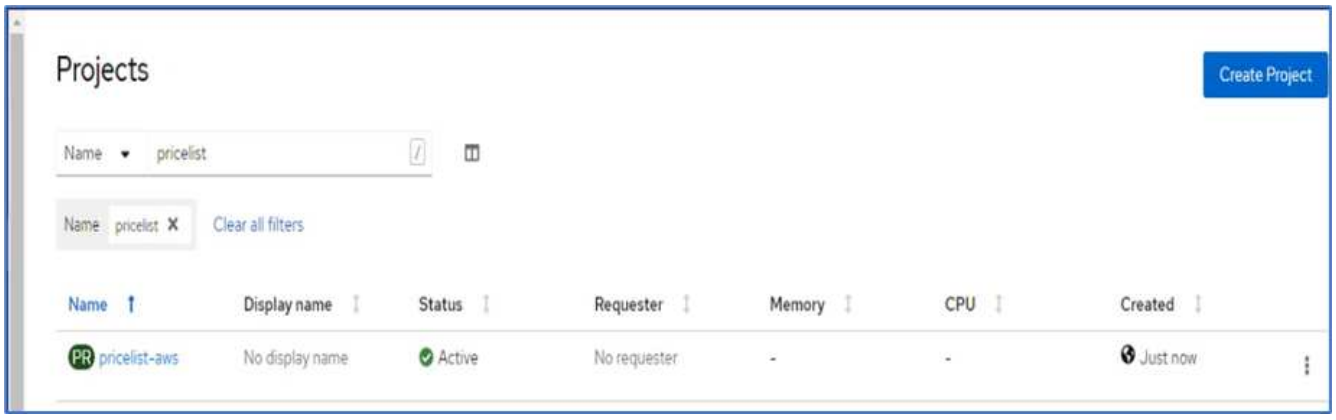
25. 选择`OCP-AWS`作为目标集群、并为命名空间提供一个名称。单击按需备份、下一步、然后单击还原。



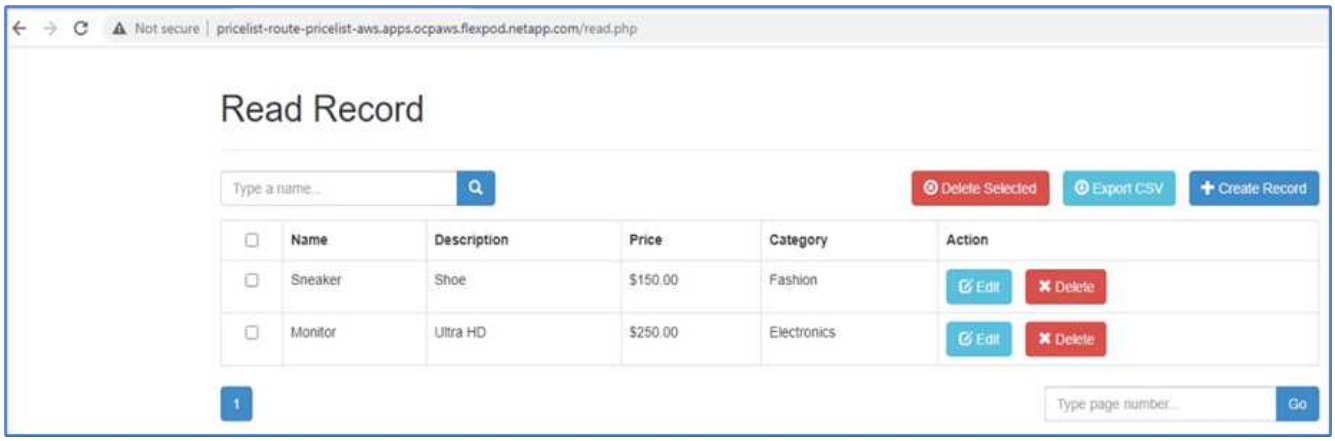
26. 在AWS中运行的OpenShift集群上会显示一个名为`pricelist-app`的新应用程序。



27. 在OpenShift Web控制台中验证相同的。



28. 运行`pricelist-AWS`项目下的所有Pod后、转到routes并单击URL以启动网页。



此过程将验证价格表应用程序是否已成功还原、以及在Astra控制中心的帮助下、在AWS上无缝运行的OpenShift集群上是否保持了数据完整性。

利用**Snapshot**副本和**DevTest**应用程序移动性保护数据

此用例由两部分组成、如下各节所述。

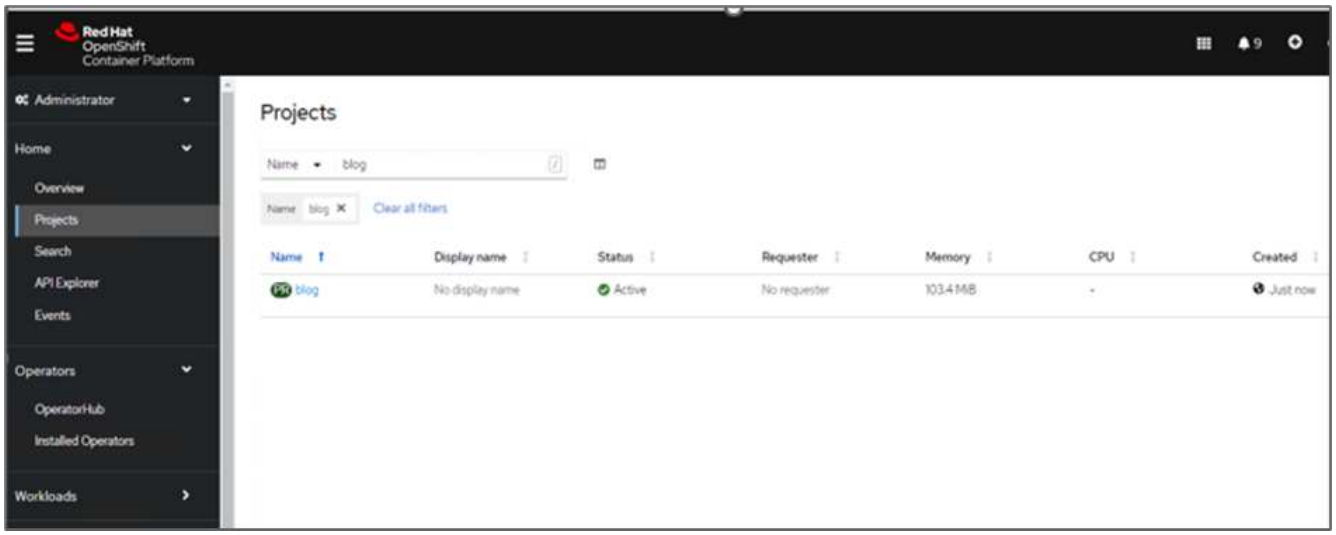
第**1**部分

借助Astra控制中心、您可以创建应用程序感知型快照以实现本地数据保护。如果意外删除或损坏数据、您可以使用先前记录的快照将应用程序和关联数据还原到已知正常状态。

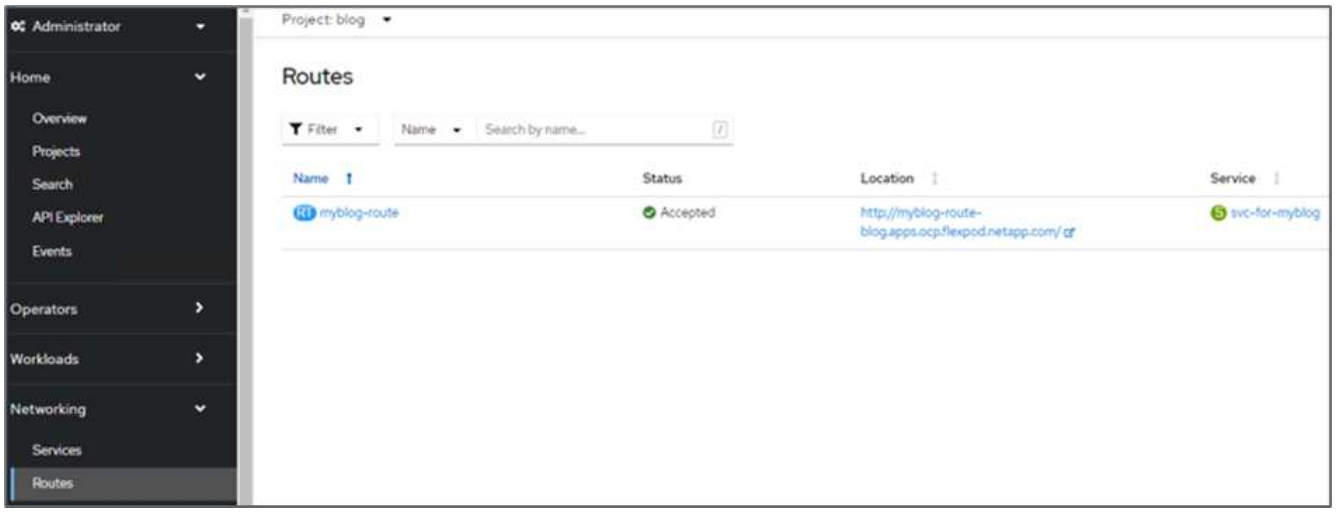
在这种情况下、开发和测试(DevTest)团队部署了一个示例有状态应用程序(博客站点)、该应用程序是一个Ghost博客应用程序、并添加了一些内容、然后将该应用程序升级到最新版本。Ghost应用程序对数据库使用SQLite。在升级应用程序之前、可以使用Astra控制中心创建一个快照(按需)来进行数据保护。详细步骤如下：

1. 部署示例博客应用程序并从ArgoCD进行同步。

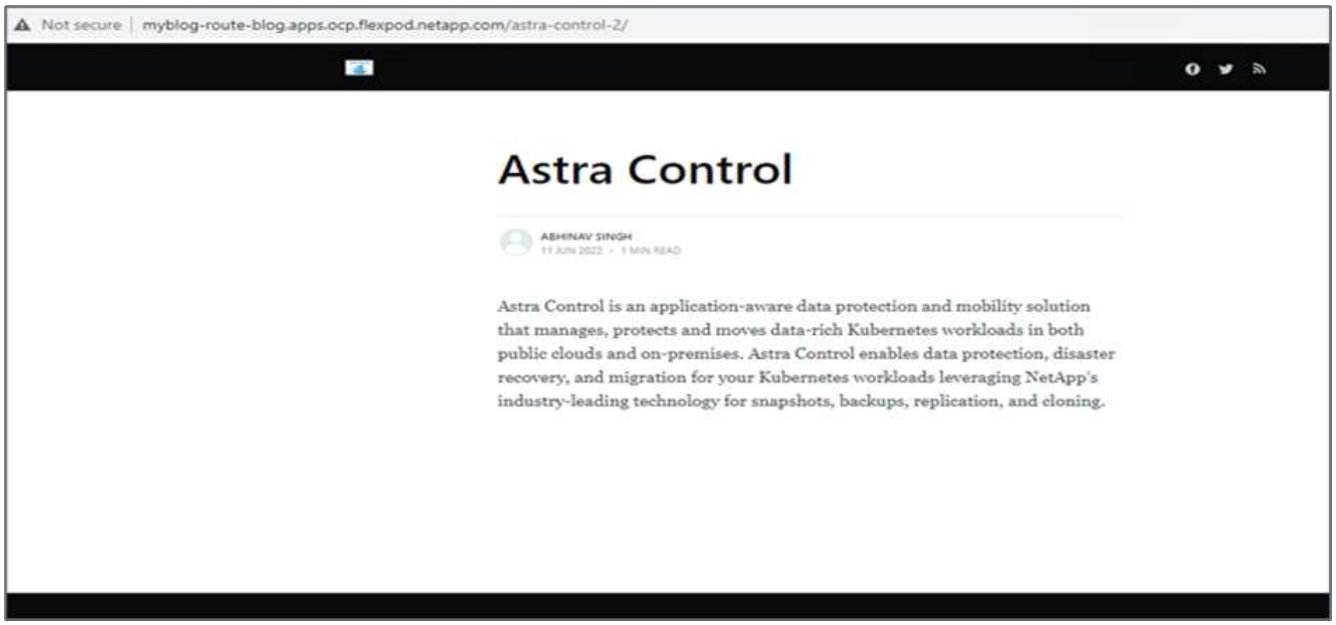
2. 登录到第一个OpenShift集群、转到Project、然后在搜索栏中输入Bllog。



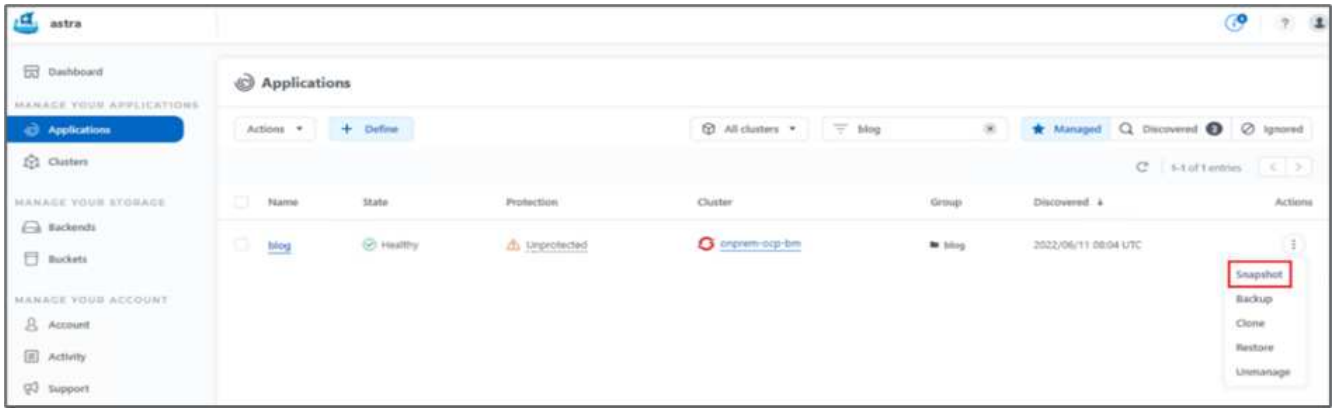
3. 从侧面菜单中、选择网络>路由、然后单击URL。



4. 此时将显示博客主页。向博客站点添加一些内容并发布。



5. 转到Astra控制中心。首先从"已发现"选项卡管理应用程序、然后创建Snapshot副本。

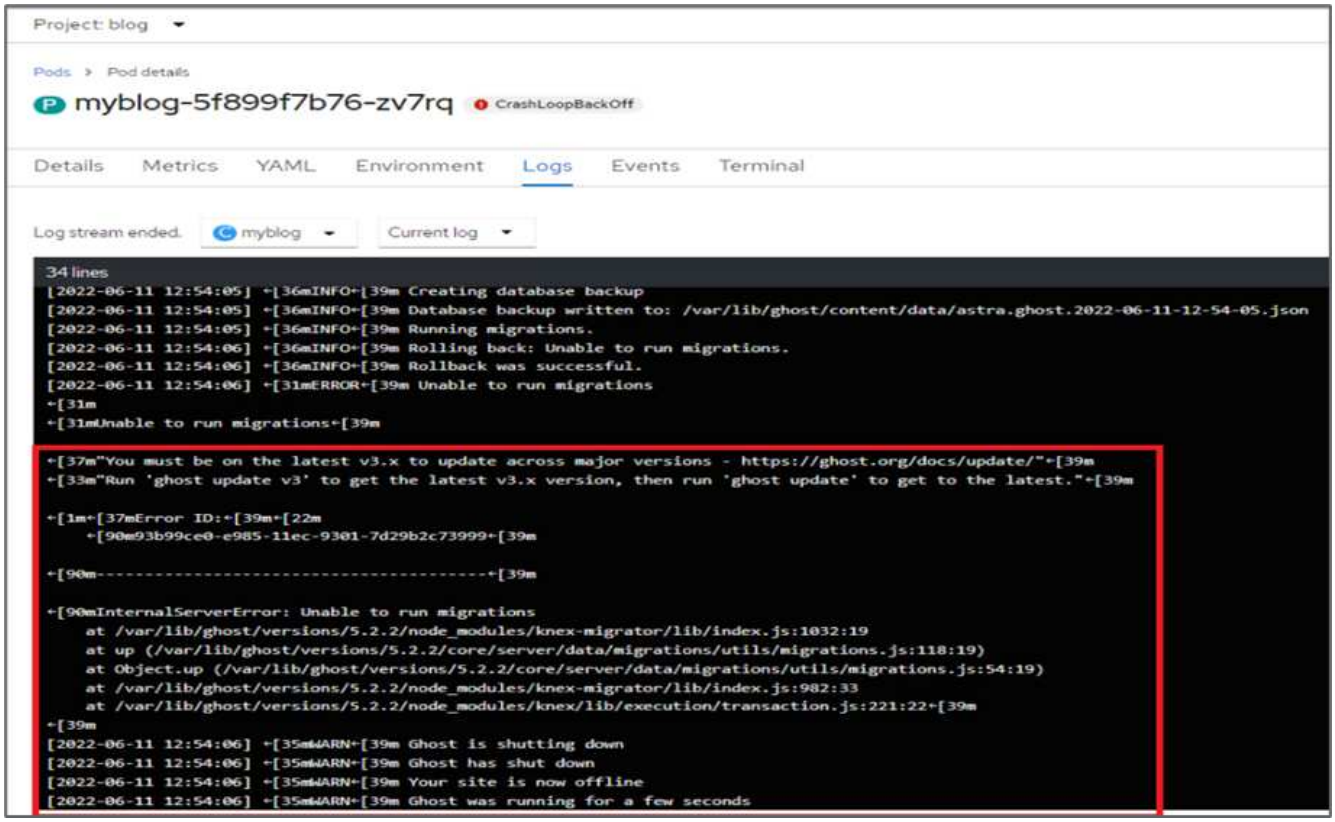


您还可以通过按定义的计划创建快照、备份或同时创建这两者来保护应用程序。有关详细信息，请参见 "通过快照和备份保护应用程序"。

6. 成功创建按需快照后、将应用程序升级到最新版本。当前映像版本为`Ghost：3.6-alpine`、目标版本为`Ghost：Latest`。要升级此应用程序、请直接更改Git存储库并将其同步到Argo CD。

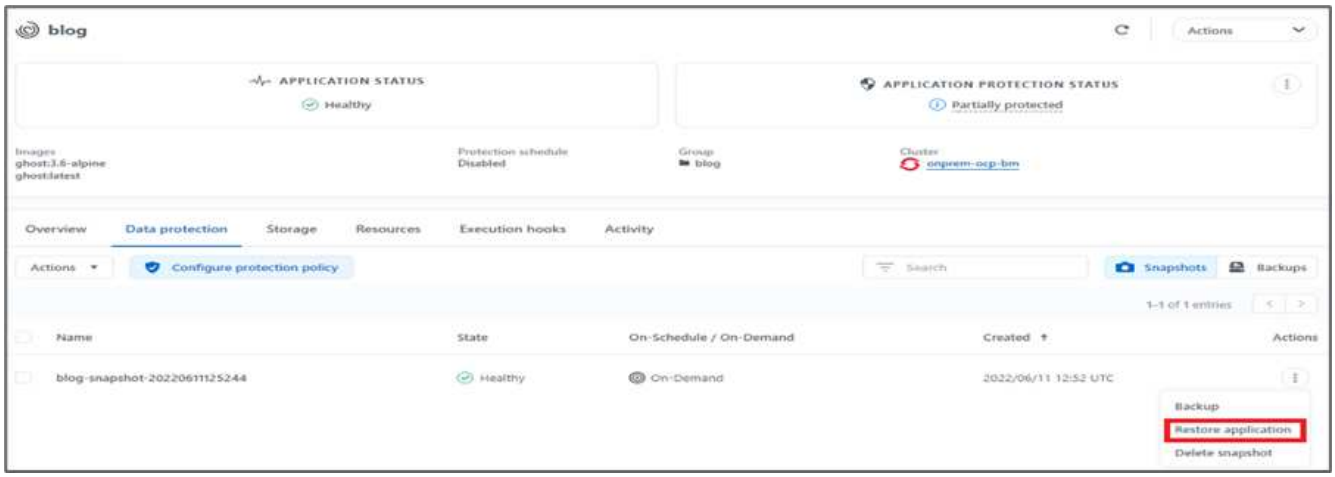
```
spec:
  containers:
  - name: myblog
    image: ghost:latest
    imagePullPolicy: Always
    ports:
    - containerPort: 2368
```

7. 您可以看到、由于博客站点关闭以及整个应用程序损坏、不支持直接升级到最新版本。

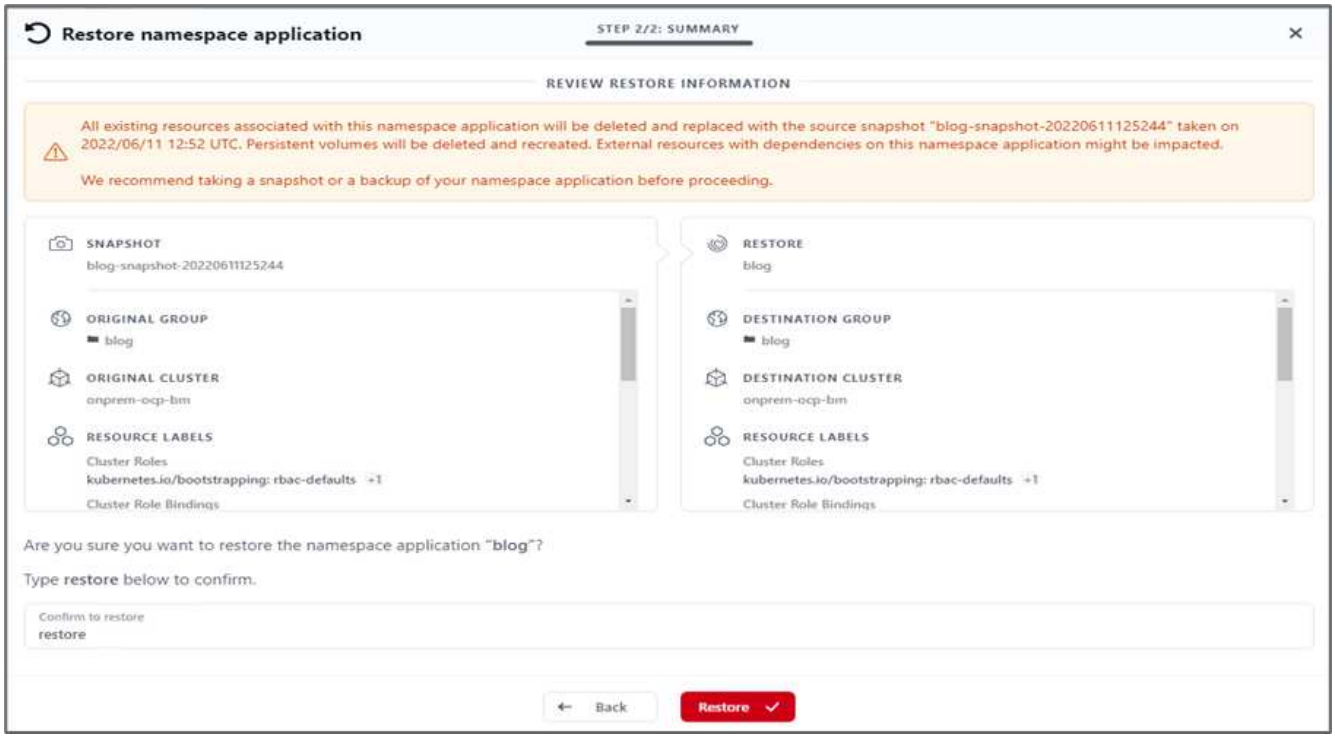


8. 要确认博客站点不可用、请刷新URL。

9. 从快照还原应用程序。



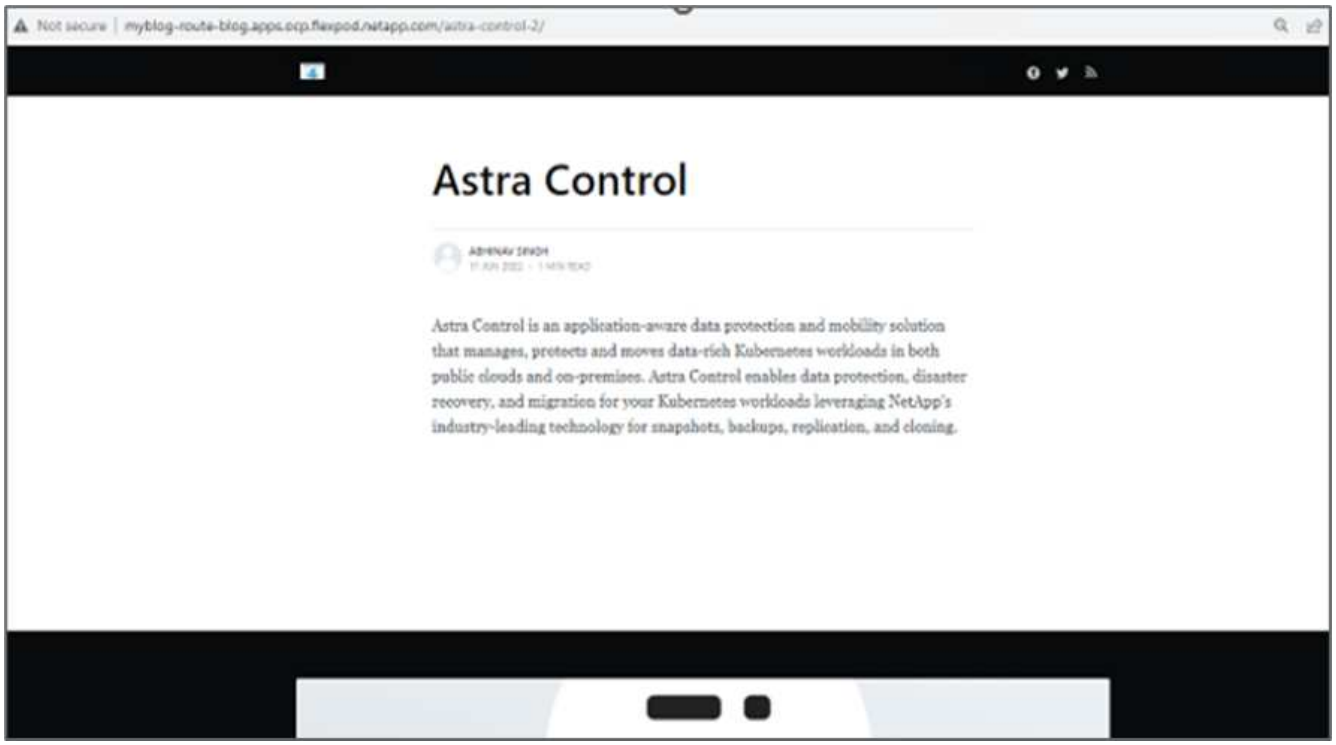
10. 此应用程序将在同一个OpenShift集群上还原。



11. 应用程序还原过程将立即启动。

12. 只需几分钟、即可从可用快照成功还原应用程序。

13. 要查看此网页是否可用、请刷新此URL。

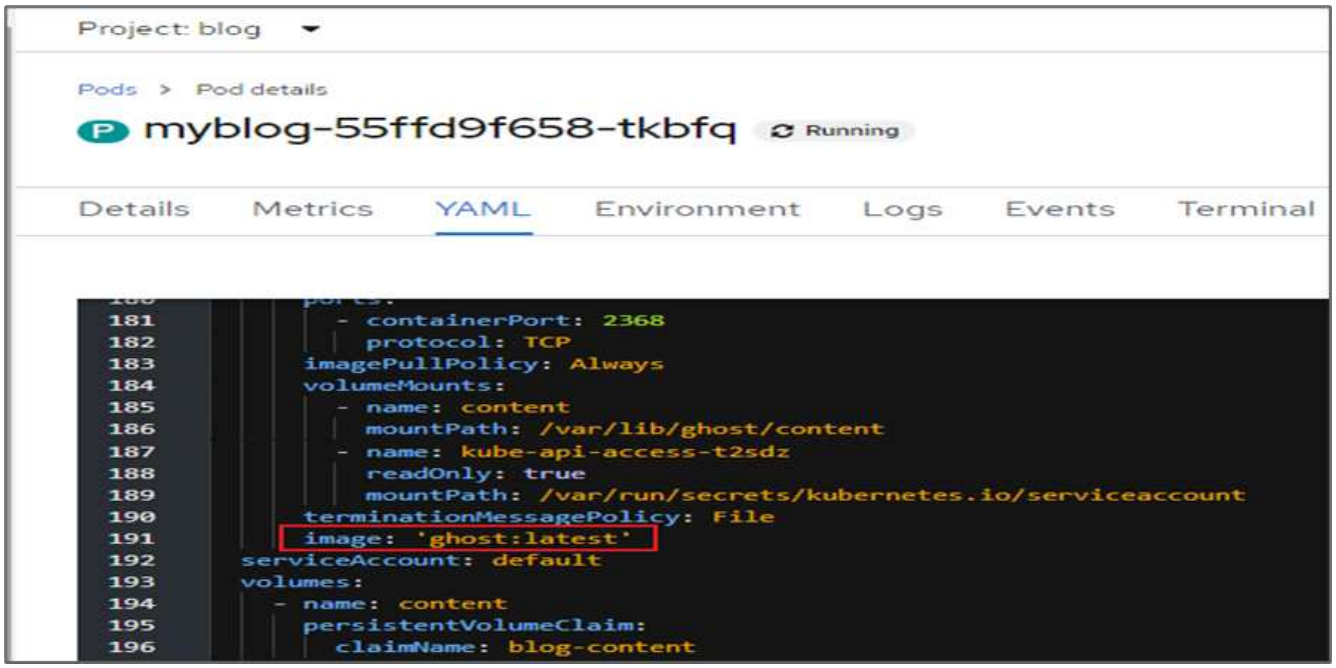


在Astra控制中心的帮助下、DevTest团队可以使用快照成功恢复博客站点应用程序及其关联数据。

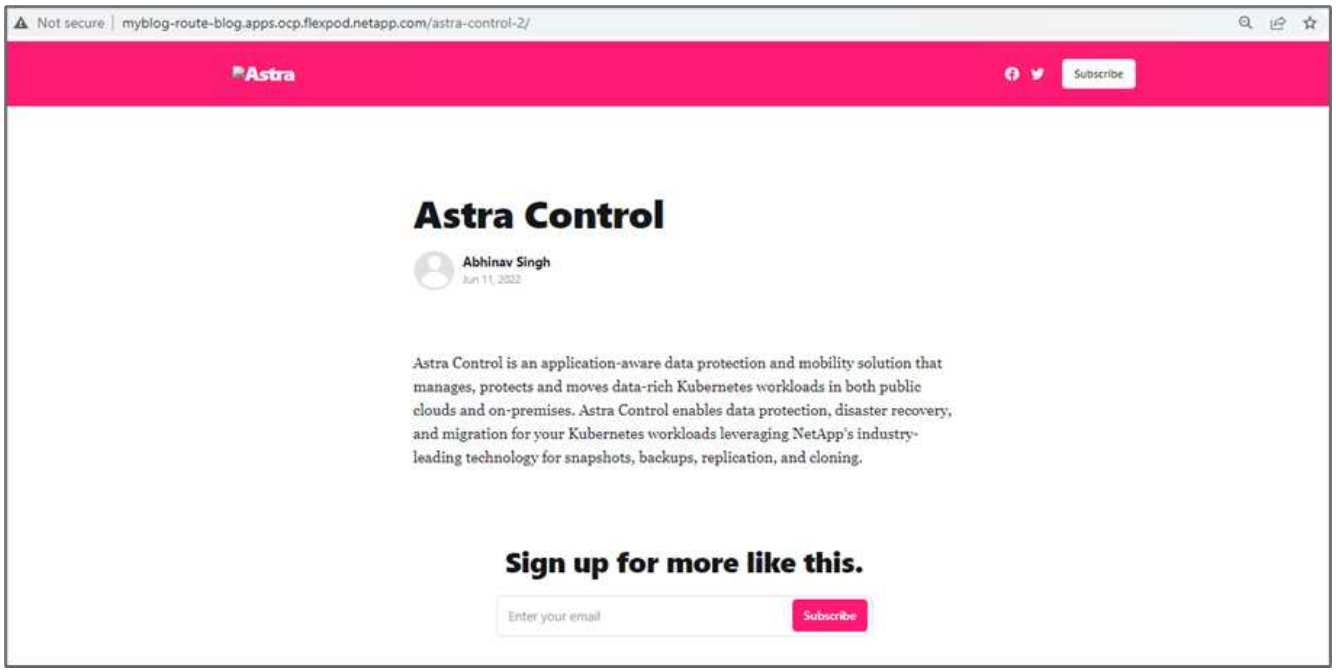
第2部分

借助Astra控制中心、您可以将整个应用程序及其数据从一个Kubernetes集群移动到另一个集群、无论这些集群位于何处(内部或云中)。

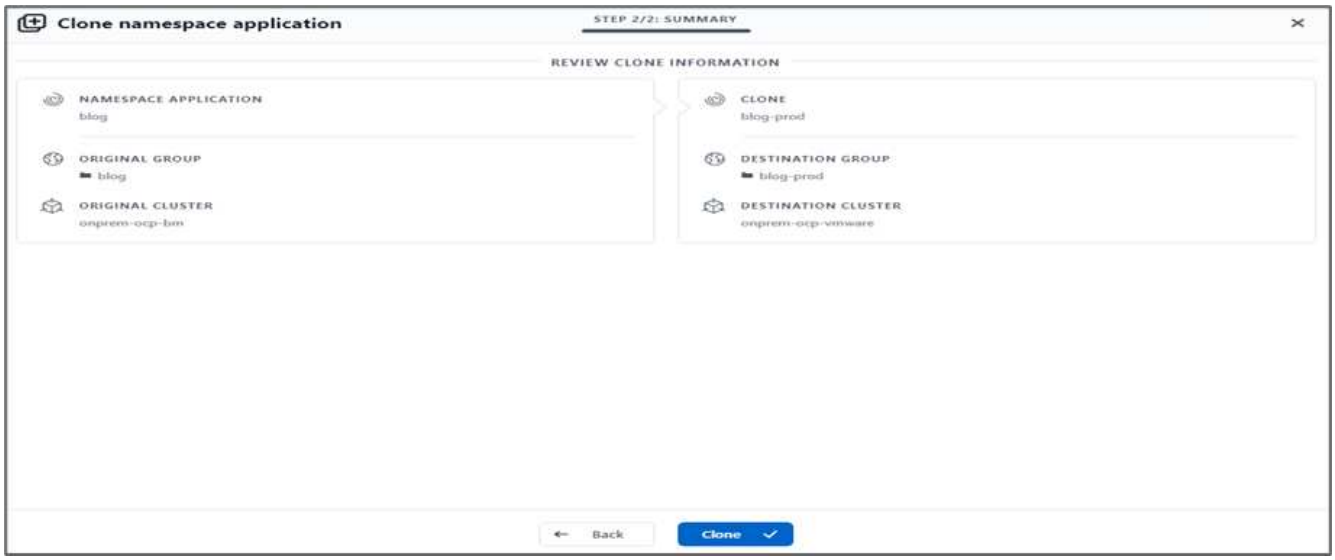
1. DevTest团队最初会先将应用程序升级到受支持的版本(`ghost-4.6. alpine`)、然后再升级到最终版本(`ghost-latest`)以使其可投入生产。然后、他们会将克隆的应用程序升级到在其他FlexPod 系统上运行的生产OpenShift集群。
2. 此时、该应用程序将升级到最新版本、并可克隆到生产集群。



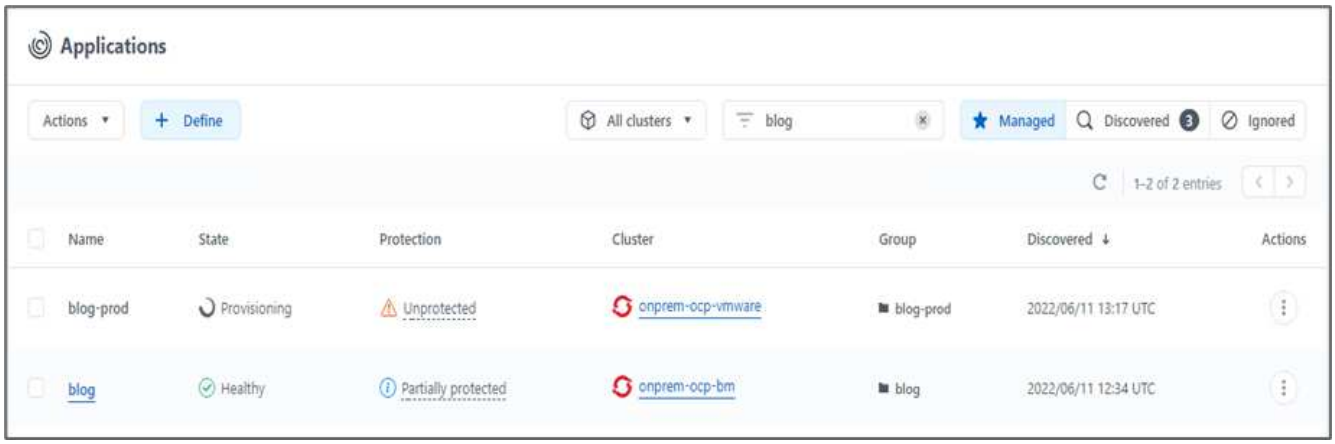
3. 要验证新主题、请刷新博客站点。



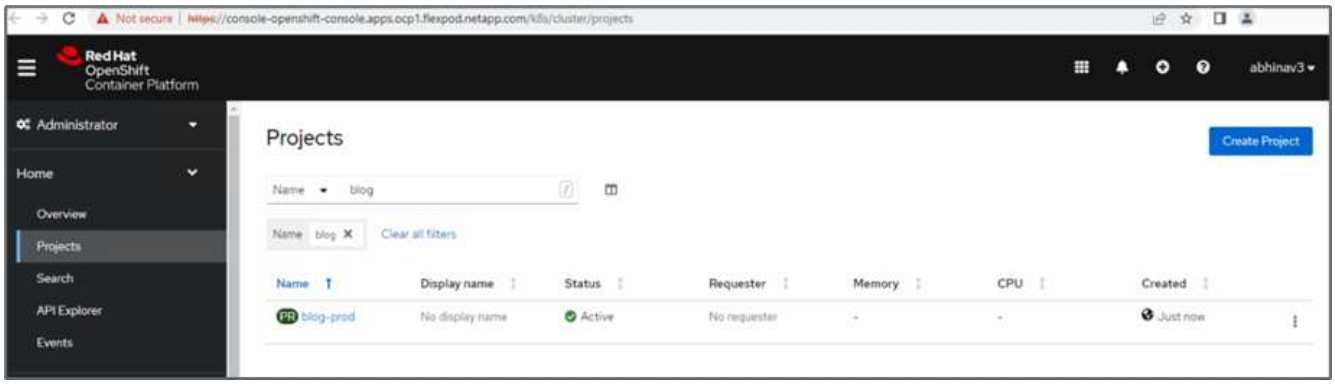
4. 从Astra控制中心、将应用程序克隆到在VMware vSphere上运行的另一个生产OpenShift集群。



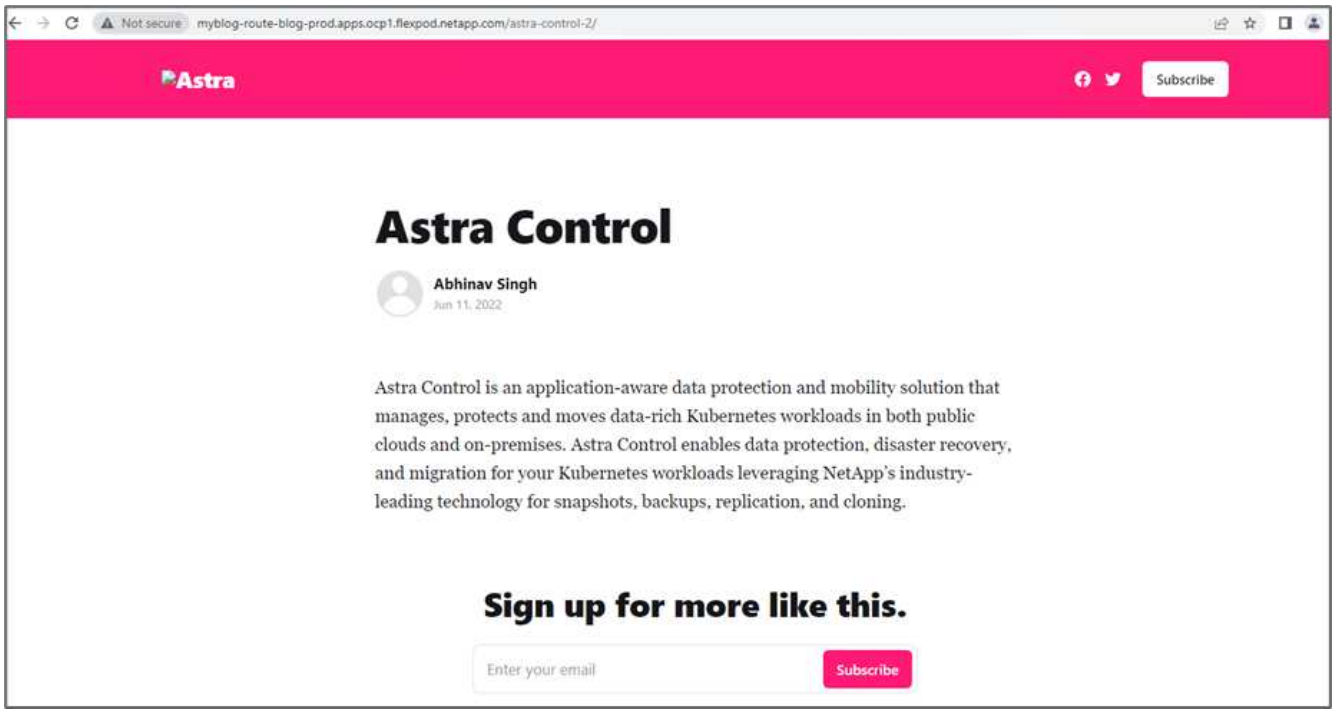
现在、生产OpenShift集群中会配置一个新的应用程序克隆。



5. 登录到生产OpenShift集群并搜索项目博客。



6. 从侧面菜单中、选择Networking > routes、然后单击Location下的URL。此时将显示与此内容相同的主页。



Astra控制中心解决方案 验证到此结束。现在、无论Kubernetes集群位于何处、您都可以将整个应用程序及其数据从一个Kubernetes集群克隆到另一个Kubernetes集群。

"接下来：总结。"

结论

"先前：使用远程备份恢复应用程序。"

在此解决方案 中、我们使用NetApp Astra产品组合为在FlexPod 和AWS上运行的容器化应用程序实施了保护计划。NetApp Astra控制中心和Astra Trident以及Cloud Volumes ONTAP 、Red Hat OpenShift和FlexPod 基础架构构成了此解决方案 的核心组件。

我们展示了通过捕获快照保护应用程序的能力、并执行完整备份来在云和内部环境中运行的不同K8s集群之间还原应用程序。

此外、我们还演示了如何跨K8s集群克隆应用程序、从而使客户能够将其应用程序迁移到所需位置的所选K8s集群。

FlexPod 不断发展、客户可以利用它来实现应用程序和业务交付流程的现代化。借助此解决方案 、FlexPod 客户可以放心地为云原生应用程序构建BCDr计划、并将公有 云作为一个位置来实施瞬时或全时灾难恢复计划、同时保持较低的解决方案 成本。

使用Astra Control、您可以将整个应用程序及其数据从一个Kubernetes集群移动到另一个集群、无论这些集群位于何处。它还可以帮助您加快云原生应用程序的部署、操作和保护速度。

故障排除

有关故障排除指导、请参见 "联机文档"。

从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请查看以下文档和 / 或网站：

- FlexPod 主页

  "https://www.flexpod.com"

- 适用于FlexPod 的Cisco验证设计和部署指南

  "https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/flexpod-design-guides.html"

- 使用Ansible为VMware部署基础架构代码FlexPod

  "https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_m6_esxi7u2.html#AnsibleAutomationWorkflowandSolutionDeployment"

- 使用Ansible将基础架构作为Red Hat OpenShift裸机的代码进行FlexPod 部署

  "https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_iac_redhat_openshift.html"

- Cisco UCS 硬件和软件互操作性工具

  "http://www.cisco.com/web/techdoc/ucs/interoperability/matrix/matrix.html"

- Cisco Intersight数据表

  "https://intersight.com/help/saas/home"

- NetApp Astra文档

  "https://docs.netapp.com/us-en/astra-control-center/index.html"

- NetApp Astra 控制中心

  "https://docs.netapp.com/us-en/astra-control-center/index.html"

- NetApp Astra Trident

  "https://docs.netapp.com/us-en/trident/index.html"

- NetApp Cloud Manager

  "https://docs.netapp.com/us-en/occm/concept_overview.html"

- NetApp Cloud Volumes ONTAP

  "https://docs.netapp.com/us-en/occm/task_getting_started_aws.html"

- Red Hat OpenShift

  "https://www.openshift.com/"

- NetApp 互操作性表工具

  "http://support.netapp.com/matrix/"

版本历史记录

version	Date	文档版本历史记录
版本 1.0	2022年7月	针对ACC22.04.0的版本。

版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

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

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

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

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

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

商标信息

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