



使用 **tridentctl** 进行部署

Astra Trident

NetApp
September 04, 2024

目录

- 使用 tridentctl 进行部署 1
 - 第 1 步：确定 Kubernetes 集群的资格 1
 - 第 2 步：下载并提取安装程序 2
 - 第 3 步：安装 Astra Trident 2
- 自定义 tridentctl 部署 3

使用 tridentctl 进行部署

您可以使用部署Astra Trident tridentctl。



如果您尚未熟悉 ["基本概念"](#)，现在是一个实现这一目标的好时机。



要自定义部署，请参见 ["此处"](#)。

您需要的内容

要部署 Astra Trident，应满足以下前提条件：

- 您对受支持的 Kubernetes 集群具有完全权限。
- 您可以访问受支持的 NetApp 存储系统。
- 您可以从所有 Kubernetes 工作节点挂载卷。
- 您有一个Linux主机 kubectl (或 oc(如果您使用的是OpenShift)已安装并配置为管理要使用的Kubernetes集群。
- 您已设置 KUBECONFIG 环境变量以指向您的Kubernetes集群配置。
- 您已启用 ["Astra Trident 所需的功能门"](#)。
- 如果您将 Kubernetes 与 Docker Enterprise 结合使用，["按照其步骤启用 CLI 访问"](#)。

明白了吗？太棒了！让我们开始吧。



有关自定义部署的信息，请参见 ["此处"](#)。

第 1 步：确定 Kubernetes 集群的资格

首先，您需要登录到 Linux 主机并验证它是否正在管理 `_b工作_`，["支持的 Kubernetes 集群"](#) 您具有所需权限。



您可以使用OpenShift oc 而不是 kubectl 在下面的所有示例中、您应先以*系统：admin*身份运行登录 oc login -u system:admin 或 oc login -u kube-admin。

要检查 Kubernetes 版本，请运行以下命令：

```
kubectl version
```

要查看您是否具有 Kubernetes 集群管理员权限，请运行以下命令：

```
kubectl auth can-i '*' '*' --all-namespaces
```

要验证是否可以从 Docker Hub 启动使用映像的 POD 并通过 Pod 网络访问存储系统，请运行以下命令：

```
kubectl run -i --tty ping --image=busybox --restart=Never --rm -- \
ping <management IP>
```

确定 Kubernetes 服务器版本。您将在安装 Astra Trident 时使用它。

第 2 步：下载并提取安装程序



Trident 安装程序将创建 Trident Pod，配置用于保持其状态的 CRD 对象，并初始化执行配置卷和将卷附加到集群主机等操作的 CSI sidecars。

您可以从下载并提取最新版本的Trident安装程序包 "[GitHub上的_assets_部分](#)"。

例如，如果最新版本为 21.07.1：

```
wget https://github.com/NetApp/trident/releases/download/v21.07.1/trident-
installer-21.07.1.tar.gz
tar -xf trident-installer-21.07.1.tar.gz
cd trident-installer
```

第 3 步：安装 Astra Trident

通过执行在所需命名空间中安装Astra Trident `tridentctl install` 命令：

```
./tridentctl install -n trident
....
INFO Starting Trident installation.                namespace=trident
INFO Created service account.
INFO Created cluster role.
INFO Created cluster role binding.
INFO Added finalizers to custom resource definitions.
INFO Created Trident service.
INFO Created Trident secret.
INFO Created Trident deployment.
INFO Created Trident daemonset.
INFO Waiting for Trident pod to start.
INFO Trident pod started.                          namespace=trident
pod=trident-csi-679648bd45-cv2mx
INFO Waiting for Trident REST interface.
INFO Trident REST interface is up.                 version=21.07.1
INFO Trident installation succeeded.
....
```

安装程序完成后，其外观将会如此。根据 Kubernetes 集群中的节点数，您可能会发现更多 Pod：

```
kubectl get pod -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-679648bd45-cv2mx      4/4     Running   0           5m29s
trident-csi-vgc8n                  2/2     Running   0           5m29s

./tridentctl -n trident version
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 21.07.1       | 21.07.1       |
+-----+-----+
```

如果您看到与上述示例类似的输出，则表示您已完成此步骤，但尚未完全配置 Astra Trident。请继续执行下一步。请参见 ["部署后任务"](#)。

但是、如果安装程序未成功完成或您未看到*正在运行* trident-csi-<generated id>、未安装此平台。



有关在部署期间排除问题的信息，请参见 ["故障排除"](#) 部分。

自定义 tridentctl 部署

使用 Trident 安装程序可以自定义属性。例如、如果已将Trident映像复制到专用存储库、则可以使用指定映像名称 `--trident-image`。如果已将Trident映像以及所需的CSI sidecar映像复制到专用存储库、则最好使用指定该存储库的位置 `--image-registry` 交换机、其形式为 `<registry FQDN>[:port]`。

如果您使用的是Kubernetes的分发版、其中 kubelet 将其数据保留在非正常路径上 `/var/lib/kubelet`、您可以使用指定备用路径 `--kubelet-dir`。

如果您需要自定义安装，使其超出安装程序参数的允许范围，则还可以自定义部署文件。使用 `--generate-custom-yaml` 参数将在安装程序中创建以下YAML文件 `setup` 目录：

- trident-clusterrolebinding.yaml
- trident-deployment.yaml
- trident-crds.yaml
- trident-clusterrole.yaml
- trident-daemonset.yaml
- trident-service.yaml
- trident-namespace.yaml
- trident-serviceaccount.yaml
- trident-resourcequota.yaml

生成这些文件后、您可以根据需要进行修改、然后使用 `--use-custom-yaml` 安装自定义部署。

```
./tridentctl install -n trident --use-custom-yaml
```

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。