设置 Astra 控制中心

Astra Control Center

NetApp
November 21, 2023

目录

设置 Astra 控制中心

Astra 控制中心支持并监控 ONTAP 和 Astra 数据存储作为存储后端。安装 Astra 控制中心，登录到 UI 并更改密码后，您将需要设置许可证，添加集群，管理存储以及添加存储分段。

任务

- 添加 Astra 控制中心的许可证
- [添加集群]
- [添加存储后端]
- [添加存储分段]

添加 Astra 控制中心的许可证

您可以使用 UI 或添加新许可证 "API" 获得完整的 Astra 控制中心功能。如果没有许可证，则只能使用 Astra 控制中心来管理用户和添加新集群。

有关如何计算许可证的详细信息，请参见 "许可"。

> 要更新现有评估版或完整许可证，请参见 "更新现有许可证"。

Astra 控制中心许可证使用 Kubernetes CPU 单元测量 CPU 资源。此许可证需要考虑分配给所有受管 Kubernetes 集群的工作节点的 CPU 资源。在添加许可证之前，您需要从获取许可证文件（NLF） "NetApp 支持站点"。

您还可以使用评估版许可证试用 Astra 控制中心，这样，您可以在自下载此许可证之日起的 90 天内使用 Astra 控制中心。您可以通过注册注册注册免费试用版 "此处"。

> 如果您的安装增长到超过许可的 CPU 单元数，则 Astra 控制中心将阻止您管理新应用程序。超过容量时，将显示警报。

您需要的内容

从下载 Astra 控制中心时 "NetApp 支持站点"，您还下载了 NetApp 许可证文件（NLF）。确保您有权访问此许可证文件。

步骤

1. 登录到 Astra 控制中心 UI 。
2. 选择 * 帐户 * > * 许可证 * 。
3. 选择 * 添加许可证 * 。
4. 浏览到您下载的许可证文件（NLF）。
5. 选择 * 添加许可证 * 。
    - 帐户 * > * 许可证 * 页面显示许可证信息，到期日期，许可证序列号，帐户 ID 和使用的 CPU 单元。

> 如果您拥有评估许可证，请务必存储帐户 ID ，以避免在未发送 ASUP 的情况下 Astra 控制中心出现故障时丢失数据。

添加集群

要开始管理应用程序，请添加 Kubernetes 集群并将其作为计算资源进行管理。您必须为 Astra 控制中心添加一个集群，才能发现您的 Kubernetes 应用程序。对于 Astra 数据存储，您希望添加 Kubernetes 应用程序集群，其中包含使用由 Astra 数据存储配置的卷的应用程序。

> 我们建议，在将其他集群添加到 Astra 控制中心进行管理之前，先由 Astra 控制中心管理其部署所在的集群。要发送 Kubemmetrics 数据和集群关联数据以获取指标和故障排除信息，必须对初始集群进行管理。您可以使用 * 添加集群 * 功能通过 Astra 控制中心管理集群。

> 当Astra Control管理集群时、它会跟踪集群的默认存储类。如果您使用更改存储类 `kubectl` 命令、Astra Control将还原更改。要更改由Astra Control管理的集群中的默认存储类、请使用以下方法之一：
>
> - 使用 Astra Control API `PUT /managedClusters` 端点、并为分配一个不同的默认存储类 `DefaultStorageClass` 参数。
> - 使用Astra Control Web UI分配其他默认存储类。请参见 [更改默认存储类]。

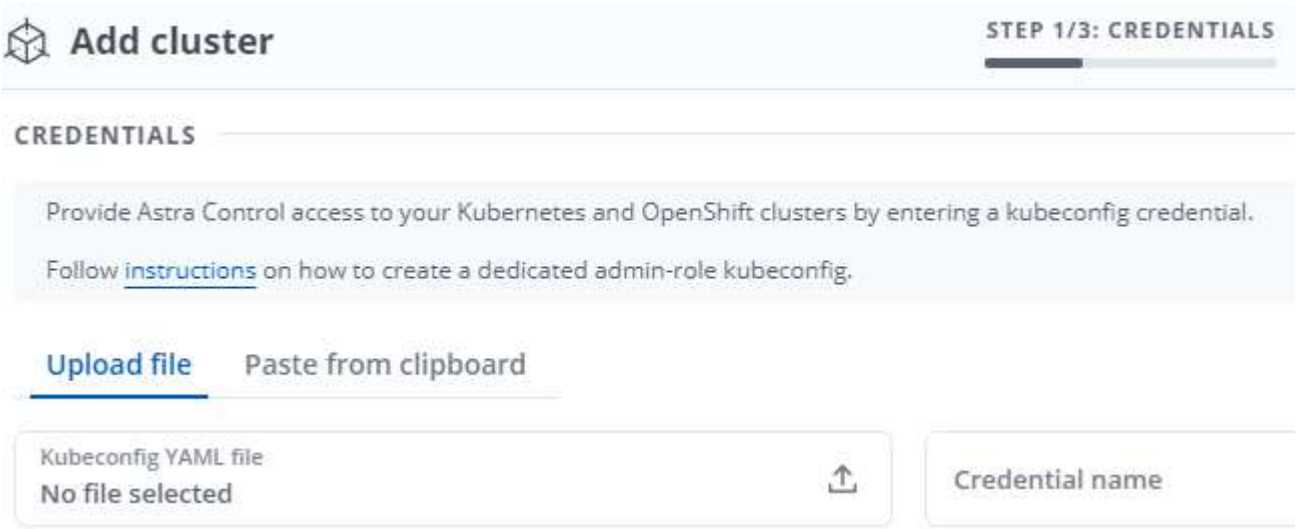
您需要的内容

- 在添加集群之前，请查看并执行必要的操作 "前提条件任务"。

步骤

1. 从 Astra 控制中心用户界面的 * 信息板 * 中，选择集群部分中的 * 添加 * 。
2. 在打开的*添加集群*窗口中、上传 `kubeconfig.yaml` 归档或粘贴的内容 `kubeconfig.yaml` 文件

   > 。 `kubeconfig.yaml` 文件应仅包含一个集群的集群凭据*。

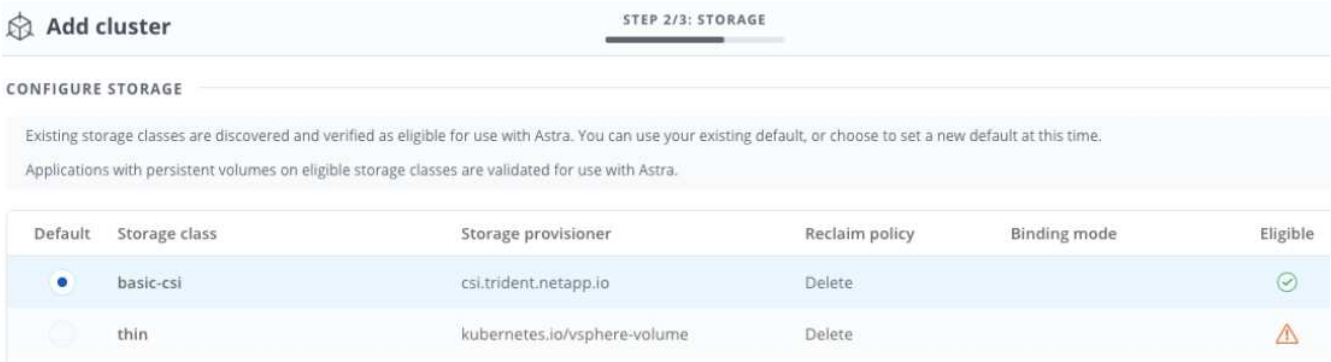


   > 创建自己的 `kubeconfig` file中、您只能定义*一*上下文元素。请参见 "Kubernetes 文档" 有关创建的信息 `kubeconfig` 文件。

3. 请提供凭据名称。默认情况下，凭据名称会自动填充为集群的名称。

4. 选择 * 配置存储 * 。
5. 选择要用于此 Kubernetes 集群的存储类，然后选择 * 审核 * 。

> 您应选择一个由 ONTAP 存储或 Astra 数据存储提供支持的 Trident 存储类。



6. 查看相关信息，如果一切正常，请选择 * 添加集群 * 。

结果

集群将进入 * 正在发现 * 状态，然后更改为 * 正在运行 * 。您已成功添加 Kubernetes 集群，现在正在 Astra 控制中心中对其进行管理。

> 添加要在 Astra 控制中心中管理的集群后，部署监控操作员可能需要几分钟的时间。在此之前，通知图标将变为红色并记录一个 * 监控代理状态检查失败 * 事件。您可以忽略此问题，因为当 Astra 控制中心获得正确状态时，问题描述将解析。如果问题描述 在几分钟内未解析、请转至集群并运行 `oc get pods -n netapp-monitoring` 作为起点。您需要查看监控操作员日志以调试此问题。

添加存储后端

您可以添加存储后端，以使 Astra Control 能够管理其资源。您可以在受管集群上部署存储后端、也可以使用现有存储后端。

通过将 Astra Control 中的存储集群作为存储后端进行管理，您可以在永久性卷（PV）和存储后端之间建立链接，并获得其他存储指标。

现有**Astra Data Store**部署所需的资源

- 您已添加Kubernetes应用程序集群和底层计算集群。

> 为Astra Data Store添加Kubernetes应用程序集群并由Astra Control管理后、此集群将显示为 `unmanaged` 发现的后端列表中。接下来，您必须添加包含 Astra 数据存储的计算集群并将 Kubernetes 应用程序集群置于底层。您可以从用户界面中的 * 后端 * 执行此操作。选择集群的"Actions"菜单、然后选择 `Manage`，和 "添加集群"。在的集群状态之后 `unmanaged` 更改Kubernetes集群的名称后、您可以继续添加后端。

新的**Astra Data Store**部署所需的资源

- 您已拥有 "已上传要部署的安装包版本" 到Astra Control可访问的位置。

- 您已添加要用于部署的Kubernetes集群。
- 您已上传 Astra Data Store许可证 部署到可供Astra Control访问的位置。

选项

- [部署存储资源]
- [使用现有存储后端]

部署存储资源

您可以部署新的Astra数据存储并管理关联的存储后端。

步骤

1. 从信息板或后端菜单导航：
   - 从*信息板*：从资源摘要中、从存储后端窗格中选择一个链接、然后从后端部分中选择*添加*。
   - 从 * 后端 * ：
     i. 在左侧导航区域中，选择 * 后端 * 。
     ii. 选择 * 添加 * 。
2. 在*部署*选项卡中选择* Astra Data Store*部署选项。
3. 选择要部署的Astra Data Store软件包：
   a. 输入Astra Data Store应用程序的名称。
   b. 选择要部署的Astra数据存储的版本。

   > 如果您尚未上传要部署的版本、可以使用*添加软件包*选项或退出向导并使用 "软件包管理" 上传安装包。

4. 选择先前上传的Astra Data Store许可证、或者使用*添加许可证*选项上传要用于应用程序的许可证。

   > 具有完全权限的Astra Data Store许可证将与您的Kubernetes集群关联、并且这些关联的集群应自动显示。如果没有受管集群、您可以选择*添加集群*选项将其添加到Astra Control管理中。对于Astra Data Store许可证、如果许可证和集群之间未建立关联、您可以在向导的下一页定义此关联。

5. 如果尚未将Kubernetes集群添加到Astra Control管理中、则需要从* Kubernetes cluster*页面中执行此操作。从列表中选择一个现有集群或选择*添加底层集群*将集群添加到Astra Control管理中。
6. 为要为Astra数据存储提供资源的Kubernetes集群选择一个模板大小。您可以选择以下选项之一：
   - 如果您选择 `Recommended Kubernetes worker node requirements`下、根据您的许可证允许的内容选择一个从大到小的模板。
   - 如果您选择 `Custom Kubernetes worker node requirements`下、选择每个集群节点所需的核心数和总内存。您还可以显示集群中符合核心和内存选择标准的节点数。

   > 选择模板时、请为大型工作负载选择具有更多内存和核心的大型节点、为小型工作负载选择更多节点。您应根据许可证允许的内容选择模板。每个建议的模板选项都会建议符合条件的节点数、这些节点满足每个节点的内存、核心和容量模板模式。

7. 配置节点：
   a. 添加节点标签以标识支持此Astra数据存储集群的工作节点池。

      在开始部署或部署失败之前、必须将此标签添加到集群中要用于部署Astra Data Store的每个节点上。

   b. 手动配置每个节点的容量(GiB)或选择允许的最大节点容量。
   c. 配置集群中允许的最大节点数或允许集群中的最大节点数。
8. (仅限Astra Data Store完整许可证)输入要用于保护域的标签的密钥。

   为每个节点的密钥至少创建三个唯一标签。例如、如果您的密钥为 `astra.datastore.protection.domain`、您可以创建以下标签：`astra.datastore.protection.domain=domain1`,`astra.datastore.protection.domain=domain2`，和 `astra.datastore.protection.domain=domain3`。

9. 配置管理网络：
   a. 输入Astra Data Store内部管理的管理IP地址、该地址与工作节点IP地址位于同一子网上。
   b. 选择对管理网络和数据网络使用相同的NIC、或者单独进行配置。
   c. 输入用于存储访问的数据网络IP地址池、子网掩码和网关。
10. 查看配置并选择*部署*以开始安装。

结果

成功安装后、后端将显示在中 `available` 后端列表中的状态以及活动性能信息。

您可能需要刷新页面才能显示后端。

使用现有存储后端

您可以将已发现的ONTAP 或Astra数据存储存储后端引入Astra控制中心管理。

步骤

1. 从信息板或后端菜单导航：
   - 从*信息板*：从资源摘要中、从存储后端窗格中选择一个链接、然后从后端部分中选择*添加*。
   - 从 * 后端 * ：
     i. 在左侧导航区域中，选择 * 后端 * 。
     ii. 在受管集群中发现的后端上选择*管理*、或者选择*添加*来管理其他现有后端。
2. 选择 * 使用现有 * 选项卡。
3. 根据后端类型执行以下操作之一：
   - * Astra 数据存储库 * ：
     i. 选择* Astra Data Store*。
     ii. 选择受管计算集群并选择 * 下一步 * 。

iii. 确认后端详细信息并选择*添加存储后端*。

- * ONTAP * ：
  i. 选择* ONTAP 并选择*下一步。
  ii. 输入ONTAP 集群管理IP地址和管理员凭据。

> 您在此处输入凭据的用户必须具有 `ontapi` 在ONTAP 集群上的ONTAP 系统管理器中启用用户登录访问方法。如果您计划使用SnapMirror复制、请启用访问方法 `ontapi` 和 `http` 适用于两个ONTAP 集群上的用户。请参见 "管理用户帐户" 有关详细信息 ...

  iii. 选择 * 审阅 * 。
  iv. 确认后端详细信息并选择*添加存储后端*。

结果

后端显示在中 `available` 包含摘要信息的列表中的状态。

> 您可能需要刷新页面才能显示后端。

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。

添加存储分段时， Astra Control 会将一个存储分段标记为默认存储分段指示符。您创建的第一个存储分段将成为默认存储分段。

如果要将应用程序配置和永久性存储克隆到同一集群，则不需要存储分段。

使用以下任一存储分段类型：

- NetApp ONTAP S3
- NetApp StorageGRID S3
- 通用 S3

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

- Microsoft Azure

> 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。

- Microsoft Azure

有关如何使用 Astra Control API 添加存储分段的说明，请参见 "Astra Automation 和 API 信息"。

步骤

1. 在左侧导航区域中，选择 * 桶 * 。

a. 选择 * 添加 * 。

b. 选择存储分段类型。

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

c. 创建新的存储分段名称或输入现有存储分段名称和可选的问题描述。

> 存储分段名称和问题描述显示为备份位置，您可以稍后在创建备份时选择该位置。此名称也会在配置保护策略期间显示。

d. 输入 S3 端点的名称或 IP 地址。

e. 如果希望此存储分段成为所有备份的默认存储分段、请选中 `Make this bucket the default bucket for this private cloud` 选项

> 创建的第一个存储分段不会显示此选项。

f. 通过添加继续 凭据信息。

添加 S3 访问凭据

随时添加 S3 访问凭据。

步骤

1. 从 " 分段 " 对话框中，选择 * 添加 * 或 * 使用现有 * 选项卡。
   a. 在 Astra Control 中输入凭据名称，以便与其他凭据区分开。
   b. 通过粘贴剪贴板中的内容来输入访问 ID 和机密密钥。

更改默认存储类

您可以更改集群的默认存储类。

步骤

1. 在Astra控制中心Web 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*。

下一步是什么?

现在，您已登录并将集群添加到 Astra 控制中心，即可开始使用 Astra 控制中心的应用程序数据管理功能。

- "管理用户"
- "开始管理应用程序"
- "保护应用程序"
- "克隆应用程序"
- "管理通知"
- "连接到 Cloud Insights"
- "添加自定义 TLS 证书"

了解更多信息

- "使用 Astra Control API"
- "已知问题"

添加集群的前提条件

在添加集群之前，应确保满足前提条件。您还应运行资格检查，以确保集群已准备好添加到 Astra 控制中心。

添加集群之前需要满足的要求

确保集群满足中所述的要求 "应用程序集群要求"。

> 如果您计划将第二个 OpenShift 4.6 ， 4.7 或 4.8 集群添加为托管计算资源，则应确保已启用 Astra Trident 卷快照功能。请参见官方的 Astra Trident "说明" 使用 Astra Trident 启用和测试卷快照。

- 使用配置了的Astra Trident StorageClasses "支持的存储后端" (对于任何类型的集群都是必需的)
- 在备份 ONTAP 系统上设置的超级用户和用户 ID ，用于使用 Astra 控制中心备份和还原应用程序。在ONTAP 命令行中运行以下命令：
  ```
  export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -superuser sysm --anon 65534
  ```
- Astra Trident `volumesnapshotclass` 由管理员定义的对象。请参见 Astra Trident "说明" 使用 Astra Trident 启用和测试卷快照。
- 确保您仅为 Kubernetes 集群定义了一个默认存储类。

运行资格检查

运行以下资格检查，以确保您的集群已准备好添加到 Astra 控制中心。

步骤

1. 检查 Trident 版本。

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

如果存在 Trident ，您将看到类似于以下内容的输出：

```
NAME      VERSION
trident   21.04.0
```

如果 Trident 不存在，您将看到类似于以下内容的输出：

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

> 如果未安装 Trident 或安装的版本不是最新的，则需要先安装最新版本的 Trident ，然后再继续操作。请参见 "Trident 文档" 有关说明，请参见。

2. 检查存储类是否正在使用受支持的 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
NAME                    PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
ontap-gold (default)    csi.trident.netapp.io          Delete
Immediate           true                   5d23h
thin                    kubernetes.io/vsphere-volume   Delete
Immediate           false                  6d
```

创建管理员角色 kubeconfig

执行这些步骤之前，请确保您的计算机上具有以下内容：

- `kubectl` 已安装v1.19或更高版本
- 具有活动上下文集群管理员权限的活动 kubeconfig

步骤

1. 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`。

      根据需要调整名称和命名空间。如果在此处进行了更改，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: astracontrol-service-account
  namespace: default
```

a. 应用服务帐户：

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

2. （可选）如果集群使用限制性的 POD 安全策略，该策略不允许创建特权 POD 或允许 Pod 容器中的进程以 root 用户身份运行，请为集群创建一个自定义 POD 安全策略，以使 Astra Control 能够创建和管理 Pod 。有关说明，请参见 "创建自定义 POD 安全策略"。
3. 按如下所示授予集群管理员权限：

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

根据需要调整创建服务帐户时修改的任何名称和命名空间。

```
<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: cluster-admin
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

a. 应用集群角色绑定：

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

4. 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-vhz87"},
{ "name": "astracontrol-service-account-token-r59kr"}
]
```

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-vhz87` 将为0、并为创建索引 `astracontrol-service-account-token-r59kr` 将为1。在输出中，记下包含 "token" 一词的服务帐户名称的索引。

5. 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
```

6. （* 可选 *）将 kubeconfig 重命名为集群的有意义名称。保护集群凭据。

```
chmod 700 create-kubeconfig.sh
mv kubeconfig-sa.txt YOUR_CLUSTER_NAME_kubeconfig
```

下一步是什么？

确认满足了这些前提条件后，您便已准备就绪 "添加集群"。

了解更多信息

- "Trident 文档"
- "使用 Astra Control API"

添加自定义 TLS 证书

您可以删除现有的自签名 TLS 证书，并将其替换为由证书颁发机构（CA）签名的 TLS 证书。

您需要的内容

- 安装了 Astra 控制中心的 Kubernetes 集群
- 对集群上要运行的命令Shell的管理访问 `kubectl` 命令
- CA 中的专用密钥和证书文件

删除自签名证书

删除现有的自签名 TLS 证书。

1. 使用 SSH ，以管理用户身份登录到托管 Astra 控制中心的 Kubernetes 集群。
2. 使用以下命令替换、查找与当前证书关联的TLS密钥 `<ACC-deployment-namespace>` 使用Astra Control Center部署命名空间：

```
kubectl get certificate -n <ACC-deployment-namespace>
```

3. 使用以下命令删除当前安装的密钥和证书：

```
kubectl delete cert cert-manager-certificates -n <ACC-deployment-
namespace>
kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
```

添加新证书

添加一个由 CA 签名的新 TLS 证书。

1. 使用以下命令使用 CA 中的专用密钥和证书文件创建新的 TLS 密钥，并将括号 <> 中的参数替换为相应的信息：

```
kubectl create secret tls <secret-name> --key <private-key-filename>
--cert <certificate-filename> -n <ACC-deployment-namespace>
```

2. 使用以下命令和示例编辑集群自定义资源定义(CRD)文件并更改 `spec.selfSigned` 值为 `spec.ca.secretName` 要引用先前创建的TLS密钥、请执行以下操作：

```
kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
<ACC-deployment-namespace>
....

#spec:
#  selfSigned: {}

spec:
  ca:
    secretName: <secret-name>
```

3. 使用以下命令和示例输出验证所做的更改是否正确以及集群是否已准备好验证证书、然后进行替换 `<ACC-deployment-namespace>` 使用Astra Control Center部署命名空间：

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
....

Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. 创建 `certificate.yaml` file使用以下示例将括号<>中的占位符值替换为相应的信息：

```yaml
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: <certificate-name>
  namespace: <ACC-deployment-namespace>
spec:
  secretName: <certificate-secret-name>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  - <astra.dnsname.example.com> #Replace with the correct Astra Control
Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. 使用以下命令创建证书：

```
kubectl apply -f certificate.yaml
```

6. 使用以下命令和示例输出，验证是否已正确创建证书以及是否使用您在创建期间指定的参数（例如名称，持续时间，续订截止日期和 DNS 名称）。

```
kubectl describe certificate -n <ACC-deployment-namespace>
....

Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. 使用以下命令和示例编辑传入 CRD TLS 选项以指向新的证书密钥，并将括号 <> 中的占位符值替换为相应的信息：

```
kubectl edit ingressroutes.traefik.containo.us -n <ACC-deployment-
namespace>
....

# tls:
#    options:
#      name: default
#    secretName: secure-testing-cert
#    store:
#      name: default

 tls:
    options:
      name: default
    secretName: <certificate-secret-name>
    store:
      name: default
```

8. 使用 Web 浏览器浏览到 Astra 控制中心的部署 IP 地址。
9. 验证证书详细信息是否与您安装的证书的详细信息匹配。
10. 导出证书并将结果导入到 Web 浏览器中的证书管理器中。

创建自定义 POD 安全策略

Astra Control 需要在其管理的集群上创建和管理 Kubernetes Pod 。如果集群使用的限制性 POD 安全策略不允许创建特权 POD 或允许 Pod 容器中的进程以 root 用户身份运行，则需要创建限制性较低的 POD 安全策略，以使 Astra Control 能够创建和管理这些 Pod 。

步骤

1. 为集群创建一个限制性低于默认值的 POD 安全策略，并将其保存在文件中。例如：

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: astracontrol
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
```

2. 为 POD 安全策略创建新角色。

```
kubectl-admin create role psp:astracontrol \
    --verb=use \
    --resource=podsecuritypolicy \
    --resource-name=astracontrol
```

3. 将新角色绑定到服务帐户。

```
kubectl-admin create rolebinding default:psp:astracontrol \
    --role=psp:astracontrol \
    --serviceaccount=astracontrol-service-account:default
```