Amazon FSx for NetApp ONTAP

Trident

NetApp
January 15, 2026

目录

Amazon FSx for NetApp ONTAP

将Trident与Amazon FSx for NetApp ONTAP

"Amazon FSx for NetApp ONTAP"是一项完全托管的 AWS 服务，使客户能够启动和运行由NetApp ONTAP存储操作系统支持的文件系统。 FSx for ONTAP使您能够利用您熟悉的NetApp功能、性能和管理能力，同时享受在 AWS 上存储数据的简单性、敏捷性、安全性和可扩展性。 FSx for ONTAP支持ONTAP文件系统功能和管理 API。

您可以将Amazon FSx for NetApp ONTAP文件系统与Trident集成，以确保在 Amazon Elastic Kubernetes Service (EKS) 中运行的 Kubernetes 集群可以配置由ONTAP支持的块和文件持久卷。

文件系统是Amazon FSx中的主要资源，类似于本地的ONTAP集群。在每个 SVM 中，您可以创建一个或多个卷，这些卷是用于存储文件系统中的文件和文件夹的数据容器。 Amazon FSx for NetApp ONTAP将作为云端托管文件系统提供。新的文件系统类型称为 * NetApp ONTAP*。

通过将Trident与Amazon FSx for NetApp ONTAP结合使用，您可以确保在 Amazon Elastic Kubernetes Service (EKS) 中运行的 Kubernetes 集群可以配置由ONTAP支持的块和文件持久卷。

要求

此外"Trident的要求"要将 FSx for ONTAP与Trident集成，您需要：

- 现有的 Amazon EKS 集群或自管理 Kubernetes 集群 `kubectl`已安装。
- 集群工作节点可访问的现有Amazon FSx for NetApp ONTAP文件系统和存储虚拟机 (SVM)。
- 已准备好的工作节点"NFS 或 iSCSI"。

> 请务必按照 Amazon Linux 和 Ubuntu 所需的节点准备步骤进行操作。 "亚马逊机器图像"（AMI）取决于您的 EKS AMI 类型。

注意事项

- SMB 卷：
  - 使用以下方式支持 SMB 卷 `ontap-nas`仅限司机。
  - Trident EKS 插件不支持 SMB 卷。
  - Trident仅支持挂载到运行在 Windows 节点上的 pod 的 SMB 卷。参考 "准备配置SMB卷" 了解详情。
- 在Trident 24.02 之前，在Amazon FSx文件系统上创建的、启用了自动备份的卷无法被Trident删除。为防止在Trident 24.02 或更高版本中出现此问题，请指定 `fsxFilesystemID`，AWS `apiRegion` ，AWS `apikey`以及 AWS `secretKey`在 AWS FSx for ONTAP的后端配置文件中。

> 如果您要为Trident指定 IAM 角色，则可以省略指定以下内容： `apiRegion` ， `apiKey` ， 和 `secretKey`明确地将字段传递给Trident 。更多信息，请参阅"FSx for ONTAP配置选项和示例"。

同时使用**Trident SAN/iSCSI** 和 **EBS-CSI** 驱动程序

如果您计划将 ontap-san 驱动程序（例如 iSCSI）与 AWS（EKS、ROSA、EC2 或任何其他实例）一起使用，则节点上所需的多路径配置可能会与 Amazon Elastic Block Store (EBS) CSI 驱动程序冲突。为了确保多路径功能不会干扰同一节点上的 EBS 磁盘，您需要在多路径设置中排除 EBS。这个例子展示了 `multipath.conf`包含所需Trident设置但排除 EBS 磁盘进行多路径配置的文件：

```
defaults {
    find_multipaths no
}
blacklist {
    device {
        vendor "NVME"
        product "Amazon Elastic Block Store"
    }
}
```

身份验证

Trident提供两种身份验证方式。

- 基于凭证（推荐）：将凭证安全地存储在 AWS Secrets Manager 中。您可以使用 `fsxadmin`文件系统的用户或 `vsadmin`用户已配置到您的 SVM。

> Trident预计将以……的形式运营 `vsadmin`SVM 用户，或者使用具有相同角色但名称不同的用户。 Amazon FSx for NetApp ONTAP具有 `fsxadmin`该用户是ONTAP的有限替代品 `admin`集群用户。我们强烈建议使用 `vsadmin`用Trident。

- 基于证书： Trident将使用安装在 SVM 上的证书与 FSx 文件系统上的 SVM 进行通信。

有关启用身份验证的详细信息，请参阅您的驱动程序类型的身份验证说明：

- "ONTAP NAS 认证"
- "ONTAP SAN 身份验证"

已测试的亚马逊机器映像 (AMI)

EKS 集群支持各种操作系统，但 AWS 针对容器和 EKS 优化了某些 Amazon Machine Image (AMI)。以下 AMI 已使用NetApp Trident 25.02 进行测试。

急性心肌梗死	NAS	NAS经济	iSCSI	iSCSI经济型
AL2023_x86_64_STANDARD	是	是	是	是
AL2_x86_64	是	是	是的*	是的*
BOTTLEROCKET_x86_64	是的**	是	不适用	不适用

AL2023_ARM_64_STANDARD	是	是	是	是
AL2_ARM_64	是	是	是的*	是的*
BOTTLEROCKET_ARM_64	是的**	是	不适用	不适用

- * 如果不重启节点，则无法删除 PV
- ** 不适用于Trident版本 25.02 的 NFSv3。

如果您所需的 AMI 未在此处列出，并不意味着它不受支持；这仅仅意味着它尚未经过测试。此列表可作为 AMI 已知可用系统的指南。

使用以下工具进行测试：

- EKS版本：1.32
- 安装方法：Helm 25.06 和 AWS 附加组件 25.06
- 对于 NAS，NFSv3 和 NFSv4.1 都进行了测试。
- SAN 仅测试了 iSCSI，未测试 NVMe-oF。

已执行测试：

- 创建：存储类、PVC、Pod
- 删除：pod、pvc（常规、qtree/lun – 经济型、带 AWS 备份的 NAS）

查找更多信息

- "Amazon FSx for NetApp ONTAP文档"
- "关于Amazon FSx for NetApp ONTAP的博客文章"

创建 IAM 角色和 AWS Secret

您可以配置 Kubernetes pod 以通过 AWS IAM 角色进行身份验证来访问 AWS 资源，而不是提供显式的 AWS 凭证。

要使用 AWS IAM 角色进行身份验证，您必须拥有一个使用 EKS 部署的 Kubernetes 集群。

创建 AWS Secrets Manager 密钥

由于Trident将向 FSx 虚拟服务器发出 API 来为您管理存储，因此它需要凭据才能执行此操作。传递这些凭证的安全方法是通过 AWS Secrets Manager 密钥。因此，如果您还没有 AWS Secrets Manager 密钥，则需要创建一个包含 vsadmin 帐户凭证的密钥。

此示例创建一个 AWS Secrets Manager 密钥来存储Trident CSI 凭证：

```
aws secretsmanager create-secret --name trident-secret --description
"Trident CSI credentials"\
  --secret-string
"{\"username\":\"vsadmin\",\"password\":\"<svmpassword>\"}"
```

创建 IAM 策略

Trident也需要 AWS 权限才能正常运行。因此，您需要创建一个策略，赋予Trident所需的权限。

以下示例使用 AWS CLI 创建 IAM 策略：

```
aws iam create-policy --policy-name AmazonFSxNCSIDriverPolicy --policy
-document file://policy.json
  --description "This policy grants access to Trident CSI to FSxN and
Secrets manager"
```

策略 **JSON** 示例：

```
{
  "Statement": [
    {
      "Action": [
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes",
        "fsx:CreateVolume",
        "fsx:RestoreVolumeFromSnapshot",
        "fsx:DescribeStorageVirtualMachines",
        "fsx:UntagResource",
        "fsx:UpdateVolume",
        "fsx:TagResource",
        "fsx:DeleteVolume"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Effect": "Allow",
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-
id>:secret:<aws-secret-manager-name>*"
    }
  ],
  "Version": "2012-10-17"
}
```

创建 **Pod** 身份或 **IAM** 角色以关联服务帐户 **(IRSA)**

您可以配置 Kubernetes 服务账户，使其承担 AWS Identity and Access Management (IAM) 角色（使用 EKS Pod Identity 或 IAM 角色进行服务账户关联）(IRSA)。任何配置为使用该服务账户的 Pod 都可以访问该角色有权访问的任何 AWS 服务。

Pod 身份

Amazon EKS Pod 身份关联允许您管理应用程序的凭证，类似于 Amazon EC2 实例配置文件向 Amazon EC2 实例提供凭证的方式。

在 **EKS** 集群上安装 **Pod Identity**：

您可以通过 AWS 控制台创建 Pod 标识，也可以使用以下 AWS CLI 命令：

```
aws eks create-addon --cluster-name <EKS_CLUSTER_NAME> --addon-name
eks-pod-identity-agent
```

更多信息请参阅"设置 Amazon EKS Pod 身份代理"。

创建 **trust-relationship.json** 文件：

创建 trust-relationship.json 文件，使 EKS 服务主体能够承担 Pod 身份的此角色。然后使用以下信任策略创建角色：

```
aws iam create-role \
  --role-name fsxn-csi-role --assume-role-policy-document file://trust-
relationship.json \
  --description "fsxn csi pod identity role"
```

trust-relationship.json 文件：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

将角色策略附加到 **IAM** 角色：

将上一步创建的角色策略附加到已创建的 IAM 角色：

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:111122223333:policy/fsxn-csi-policy \
  --role-name fsxn-csi-role
```

创建 **Pod** 身份关联：

在 IAM 角色和Trident服务帐户 (trident-controller) 之间创建 Pod 身份关联

```
aws eks create-pod-identity-association \
  --cluster-name <EKS_CLUSTER_NAME> \
  --role-arn arn:aws:iam::111122223333:role/fsxn-csi-role \
  --namespace trident --service-account trident-controller
```

服务帐户关联 **(IRSA)** 的 **IAM** 角色

使用 **AWS CLI**：

```
aws iam create-role --role-name AmazonEKS_FSxN_CSI_DriverRole \
  --assume-role-policy-document file://trust-relationship.json
```

trust-relationship.json 文件：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::<account_id>:oidc-
provider/<oidc_provider>"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "<oidc_provider>:aud": "sts.amazonaws.com",
          "<oidc_provider>:sub":
"system:serviceaccount:trident:trident-controller"
        }
      }
    }
  ]
}
```

更新以下值 `trust-relationship.json`文件：

- **<account_id>** - 您的 AWS 账户 ID
- **<oidc_provider>** - 您的 EKS 集群的 OIDC。您可以通过运行以下命令获取oidc_provider：

```
aws eks describe-cluster --name my-cluster --query
"cluster.identity.oidc.issuer"\
  --output text | sed -e "s/^https:\/\///"
```

将 **IAM** 角色与 **IAM** 策略关联：

角色创建完成后，使用以下命令将策略（在上一步中创建的策略）附加到该角色：

```
aws iam attach-role-policy --role-name my-role --policy-arn <IAM policy
ARN>
```

请核实 **OICD** 提供商是否已关联：

请确认您的 OIDC 提供程序已与您的集群关联。您可以使用以下命令进行验证：

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

如果输出为空，请使用以下命令将 IAM OIDC 关联到您的集群：

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name
--approve
```

如果您使用的是 **eksctl**，请使用以下示例在 EKS 中为服务帐户创建 IAM 角色：

```
eksctl create iamserviceaccount --name trident-controller --namespace
trident \
  --cluster <my-cluster> --role-name AmazonEKS_FSxN_CSI_DriverRole
--role-only \
  --attach-policy-arn <IAM-Policy ARN> --approve
```

安装Trident

Trident简化了 Kubernetes 中Amazon FSx for NetApp ONTAP存储管理，使您的开发人员和管理员能够专注于应用程序部署。

您可以使用以下方法之一安装Trident：

- 舵
- EKS 附加组件

如果要使用快照功能，请安装 CSI 快照控制器插件。请参阅"为CSI卷启用快照功能"了解更多信息。

通过 **Helm** 安装**Trident**。

Pod 身份

1. 添加Trident Helm 仓库：

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-
chart
```

2. 请按照以下示例安装Trident：

```
helm install trident-operator netapp-trident/trident-operator
--version 100.2502.1 --namespace trident --create-namespace
```

您可以使用 `helm list`查看安装详细信息的命令，例如名称、命名空间、图表、状态、应用程序版本和修订号。

```
helm list -n trident
```

```
NAME                NAMESPACE   REVISION    UPDATED
STATUS       CHART                          APP VERSION

trident-operator    trident     1           2024-10-14
14:31:22.463122 +0300 IDT    deployed     trident-operator-
100.2502.0    25.02.0
```

服务账户协会（**IRSA**）

1. 添加Trident Helm 仓库：

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-
chart
```

2. 设置*云提供商*和*云身份*的值：

```
helm install trident-operator netapp-trident/trident-operator
--version 100.2502.1  \
--set cloudProvider="AWS" \
--set cloudIdentity="'eks.amazonaws.com/role-arn:
arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>'" \
--namespace trident \
--create-namespace
```

您可以使用 `helm list`查看安装详细信息的命令，例如名称、命名空间、图表、状态、应用程序版本和修订号。

```
helm list -n trident
```

```
NAME                NAMESPACE   REVISION    UPDATED
STATUS       CHART                          APP VERSION

trident-operator    trident     1           2024-10-14
14:31:22.463122 +0300 IDT    deployed     trident-operator-
100.2506.0    25.06.0
```

如果您计划使用 iSCSI，请确保您的客户端计算机上已启用 iSCSI。如果您使用的是 AL2023 工作节点操作系统，可以通过在 Helm 安装中添加节点准备参数来自动安装 iSCSI 客户端：

```
helm install trident-operator netapp-trident/trident-operator
--version 100.2502.1 --namespace trident --create-namespace --
set nodePrep={iscsi}
```

通过 **EKS** 插件安装**Trident**

Trident EKS 插件包含最新的安全补丁和错误修复，并经过 AWS 验证，可与 Amazon EKS 配合使用。 EKS 插件使您能够持续确保您的 Amazon EKS 集群安全稳定，并减少安装、配置和更新插件所需的工作量。

前提条件

在为 AWS EKS 配置Trident插件之前，请确保您已具备以下条件：

- 带有附加订阅的 Amazon EKS 集群帐户
- AWS 对 AWS Marketplace 的权限：
  `"aws-marketplace:ViewSubscriptions",`
  `"aws-marketplace:Subscribe",`
  `"aws-marketplace:Unsubscribe`
- AMI 类型：Amazon Linux 2 (AL2_x86_64) 或 Amazon Linux 2 Arm (AL2_ARM_64)
- 节点类型：AMD 或ARM
- 现有的Amazon FSx for NetApp ONTAP文件系统

启用适用于 **AWS** 的**Trident**插件

管理控制台

1. 打开 Amazon EKS 控制台 https://console.aws.amazon.com/eks/home#/clusters。
2. 在左侧导航窗格中，选择“集群”。
3. 选择要为其配置NetApp Trident CSI 插件的集群名称。
4. 选择“附加组件”，然后选择“获取更多附加组件”。
5. 请按照以下步骤选择插件：
   a. 向下滚动到“AWS Marketplace 附加组件”部分，然后在搜索框中输入“Trident”。
   b. 选中“Trident by NetApp”框右上角的复选框。
   c. 选择“下一步”。
6. 在“配置所选插件”设置页面上，执行以下操作：

   如果您使用 **Pod Identity** 关联，请跳过这些步骤。

   a. 请选择您要使用的*版本*。
   b. 如果您使用IRSA身份验证，请确保设置可选配置设置中提供的配置值：
      - 请选择您要使用的*版本*。
      - 按照*附加组件配置方案*进行操作，并将*配置值*部分中的*configurationValues*参数设置为您在上一步创建的角色 ARN（值应采用以下格式）：

```
{

  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
  "cloudProvider": "AWS"

}
```

+
如果选择“覆盖”作为冲突解决方法，则现有插件的一个或多个设置可能会被 Amazon EKS 插件设置覆盖。如果您不启用此选项，并且与您现有的设置存在冲突，则操作将失败。您可以使用生成的错误信息来排查冲突问题。在选择此选项之前，请确保 Amazon EKS 插件不会管理您需要自行管理的设置。

7. 选择“下一步”。
8. 在“审核和添加”页面上，选择“创建”。

   插件安装完成后，您将看到已安装的插件。

AWS CLI

1.创建 **`add-on.json`**文件：

对于 **Pod** 标识，请使用以下格式：

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
}
```

对于**IRSA**认证，请使用以下格式：

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
  "serviceAccountRoleArn": "<role ARN>",
  "configurationValues": {
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
    "cloudProvider": "AWS"
  }
}
```

代替 `<role ARN>`使用上一步创建的角色的 ARN。

2.安装**Trident EKS** 插件。

```
aws eks create-addon --cli-input-json file://add-on.json
```

eksctl

以下示例命令安装Trident EKS 插件：

```
eksctl create addon --name netapp_trident-operator --cluster
<cluster_name> --force
```

更新**Trident EKS** 插件

管理控制台

1. 打开 Amazon EKS 控制台 https://console.aws.amazon.com/eks/home#/clusters。
2. 在左侧导航窗格中，选择“集群”。
3. 选择要为其更新NetApp Trident CSI 插件的集群名称。
4. 选择“附加组件”选项卡。
5. 选择“ NetApp Trident ”，然后选择“编辑”。
6. 在“配置NetApp Trident ”页面上，执行以下操作：
   a. 请选择您要使用的*版本*。
   b. 展开“可选配置设置”，并根据需要进行修改。
   c. 选择“保存更改”。

AWS CLI

以下示例更新 EKS 插件：

```
aws eks update-addon --cluster-name <eks_cluster_name> --addon-name
netapp_trident-operator --addon-version v25.6.0-eksbuild.1 \
  --service-account-role-arn <role-ARN> --resolve-conflict preserve \
  --configuration-values "{\"cloudIdentity\":
\"'eks.amazonaws.com/role-arn: <role ARN>'\"}"
```

eksctl

- 请检查您的 FSxN Trident CSI 插件的当前版本。代替 `my-cluster`使用您的集群名称。

  ```
  eksctl get addon --name netapp_trident-operator --cluster my-cluster
  ```

  示例输出：

```
NAME                          VERSION              STATUS    ISSUES
IAMROLE    UPDATE AVAILABLE    CONFIGURATION VALUES
netapp_trident-operator    v25.6.0-eksbuild.1    ACTIVE    0
{"cloudIdentity":"'eks.amazonaws.com/role-arn:
arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'"}
```

- 将插件更新到上一步输出中“UPDATE AVAILABLE”下返回的版本。

  ```
  eksctl update addon --name netapp_trident-operator --version
  v25.6.0-eksbuild.1 --cluster my-cluster --force
  ```

如果你移除 `--force`如果选项和任何 Amazon EKS 插件设置与您现有的设置冲突，则更新 Amazon EKS 插件将失败；您将收到一条错误消息，以帮助您解决冲突。在指定此选项之前，请确保 Amazon EKS 插件不会管理您需要管理的设置，因为此选项会覆盖这些设置。有关此设置的其他选项的更多信息，请参阅"插件"。有关 Amazon EKS Kubernetes 字段管理的更多信息，请参阅"Kubernetes 字段管理"。

卸载/移除**Trident EKS** 插件

您可以通过两种方式移除 Amazon EKS 插件：

- 保留集群上的附加软件 – 此选项移除 Amazon EKS 对所有设置的管理。它还取消了 Amazon EKS 通知您更新以及在您启动更新后自动更新 Amazon EKS 插件的功能。但是，它可以保留集群上的附加软件。此选项使插件成为自管理安装，而不是 Amazon EKS 插件。选择此选项，插件不会出现停机时间。保留 `--preserve`命令中的选项用于保留插件。
- 从集群中完全移除附加软件 – NetApp建议，仅当集群中没有任何资源依赖于 Amazon EKS 附加组件时，才从集群中移除该附加组件。移除 `--preserve`选项 `delete`移除插件的命令。

如果插件关联了 IAM 帐户，则不会删除该 IAM 帐户。

管理控制台

1. 打开 Amazon EKS 控制台 https://console.aws.amazon.com/eks/home#/clusters。
2. 在左侧导航窗格中，选择“集群”。
3. 选择要从中移除NetApp Trident CSI 插件的集群名称。
4. 选择“附加组件”选项卡，然后选择“ NetApp Trident ”。
5. 选择*删除*。
6. 在“移除 netapp_trident-operator 确认”对话框中，执行以下操作：
   a. 如果您希望 Amazon EKS 停止管理插件的设置，请选择“在集群上保留”。如果您希望在集群上保留附加软件，以便您可以自行管理附加软件的所有设置，请执行此操作。
   b. 输入 **netapp_trident-operator**。
   c. 选择*删除*。

AWS CLI

代替 `my-cluster`输入集群名称，然后运行以下命令。

```
aws eks delete-addon --cluster-name my-cluster --addon-name
netapp_trident-operator --preserve
```

eksctl

以下命令卸载Trident EKS 插件：

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

配置存储后端

ONTAP SAN 和 NAS 驱动程序集成

要创建存储后端，您需要创建 JSON 或 YAML 格式的配置文件。该文件需要指定您想要的存储类型（NAS 或 SAN）、文件系统、要从中获取存储的 SVM 以及如何对其进行身份验证。以下示例展示了如何定义基于 NAS 的存储，以及如何使用 AWS Secret 来存储要使用的 SVM 的凭证：

YAML

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFilesystemID: fs-xxxxxxxxxx
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxxx:secret:secret-
name"
    type: awsarn
```

JSON

```
{
  "apiVersion": "trident.netapp.io/v1",
  "kind": "TridentBackendConfig",
  "metadata": {
    "name": "backend-tbc-ontap-nas"
    "namespace": "trident"
  },
  "spec": {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "tbc-ontap-nas",
    "svm": "svm-name",
    "aws": {
      "fsxFilesystemID": "fs-xxxxxxxxxx"
    },
    "managementLIF": null,
    "credentials": {
      "name": "arn:aws:secretsmanager:us-west-2:xxxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

运行以下命令以创建和验证Trident后端配置 (TBC)：

- 从 yaml 文件创建 Trident 后端配置 (TBC)，并运行以下命令：

```
kubectl create -f backendconfig.yaml -n trident
```

```
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created
```

- 验证 Trident 后端配置 (TBC) 是否已成功创建：

```
Kubectl get tbc -n trident
```

```
NAME                           BACKEND NAME          BACKEND UUID
PHASE    STATUS

backend-tbc-ontap-nas          tbc-ontap-nas         933e0071-66ce-4324-
b9ff-f96d916ac5e9    Bound    Success
```

FSx 适用于ONTAP驱动程序的详细信息

您可以使用以下驱动程序将Trident与Amazon FSx for NetApp ONTAP集成：

- `ontap-san`每个已配置的 PV 都是其自身Amazon FSx for NetApp ONTAP卷中的一个 LUN。推荐用于块存储。
- `ontap-nas`每个已配置的 PV 都是一个完整的Amazon FSx for NetApp ONTAP卷。推荐用于NFS和SMB。
- `ontap-san-economy`每个已配置的 PV 都是一个 LUN，每个Amazon FSx for NetApp ONTAP卷可配置 LUN 的数量。
- `ontap-nas-economy`每个已配置的 PV 都是一个 qtree，每个Amazon FSx for NetApp ONTAP卷可以配置 qtree 的数量。
- `ontap-nas-flexgroup`每个已配置的 PV 都是一个完整的Amazon FSx for NetApp ONTAP FlexGroup卷。

有关司机详情，请参阅"NAS驱动程序"和"SAN驱动程序"。

配置文件创建完成后，运行以下命令将其创建在 EKS 中：

```
kubectl create -f configuration_file
```

要验证状态，请运行以下命令：

```
kubectl get tbc -n trident
```

```
NAME                      BACKEND NAME            BACKEND UUID
PHASE   STATUS
backend-fsx-ontap-nas   backend-fsx-ontap-nas   7a551921-997c-4c37-a1d1-
f2f4c87fa629   Bound   Success
```

后端高级配置和示例

请参阅下表了解后端配置选项：

参数	描述	示例
`version`		始终为 1
`storageDriverName`	存储驱动程序的名称	`ontap-nas`， `ontap-nas-economy`， `ontap-nas-flexgroup`， `ontap-san`， `ontap-san-economy`
`backendName`	自定义名称或存储后端	驱动程序名称 + "_" + dataLIF
`managementLIF`	集群或 SVM 管理 LIF 的 IP 地址可以指定完全限定域名 (FQDN)。如果Trident安装时使用了 IPv6 标志，则可以设置为使用 IPv6 地址。 IPv6 地址必须用方括号定义，例如 [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。如果您提供 `fsxFilesystemID`在 `aws`字段中，您无需提供 `managementLIF`因为Trident可以检索 SVM `managementLIF`来自 AWS 的信息。因此，您必须提供 SVM 下用户的凭据（例如：vsadmin），并且该用户必须拥有以下权限： `vsadmin`角色。	"10.0.0.1", "[2001:1234:abcd::fefe]"

参数	描述	示例
dataLIF	LIF协议的IP地址。 * ONTAP NAS驱动程序*： NetApp建议指定dataLIF。如果未提供， Trident将从SVM 获取 dataLIF。您可以指定一个完全限定域名 (FQDN) 用于 NFS 挂载操作，从而创建轮询 DNS 以在多个 dataLIF 之间进行负载均衡。初始设置后可以更改。参考。 * ONTAP SAN 驱动程序*：不要为 iSCSI 指定。 Trident使用ONTAP选择性 LUN 映射来发现建立多路径会话所需的 iSCI LIF。如果显式定义了dataLIF，则会生成警告。如果Trident安装时使用了 IPv6 标志，则可以设置为使用 IPv6 地址。 IPv6 地址必须用方括号定义，例如 [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。	
autoExportPolicy	启用自动导出策略创建和更新 [布尔值]。使用 `autoExportPolicy`和 `autoExportCIDRs`可选功能包括：Trident可以自动管理出口策略。	false
autoExportCIDRs	用于过滤 Kubernetes 节点 IP 的 CIDR 列表 `autoExportPolicy`已启用。使用 `autoExportPolicy`和 `autoExportCIDRs`可选功能包括：Trident可以自动管理出口策略。	"["0.0.0.0/0", "::/0"]"
labels	要应用于卷的任意 JSON 格式标签集	""
clientCertificate	客户端证书的 Base64 编码值。用于基于证书的身份验证	""
clientPrivateKey	客户端私钥的 Base64 编码值。用于基于证书的身份验证	""
trustedCACertificate	受信任 CA 证书的 Base64 编码值。可选。用于基于证书的身份验证。	""
username	连接到集群或 SVM 的用户名。用于基于凭证的身份验证。例如，vsadmin。	
password	连接集群或SVM的密码。用于基于凭证的身份验证。	
svm	使用的存储虚拟机	如果指定了 SVM 管理 LIF，则派生出该 LIF。
storagePrefix	在 SVM 中配置新卷时使用的前缀。创建后无法修改。要更新此参数，您需要创建一个新的后端。	trident

参数	描述	示例
`limitAggregateUsage`	*请勿指定用于Amazon FSx for NetApp ONTAP。*提供的 `fsxadmin`和 `vsadmin`不包含检索汇总使用情况和使用Trident限制它所需的权限。	请勿使用。
`limitVolumeSize`	如果请求的卷大小大于此值，则配置失败。此外，它还限制了其管理的 qtree 和 LUN 卷的最大大小，以及 `qtreesPerFlexvol`此选项允许自定义每个FlexVol volume的最大 qtree 数量。	（默认情况下不强制执行）
`lunsPerFlexvol`	每个 Flexvol 卷的最大 LUN 数必须在 [50, 200] 范围内。仅限SAN。	“100”
`debugTraceFlags`	故障排除时要使用的调试标志。例如，{"api":false, "method":true} 请勿使用 `debugTraceFlags`除非您正在进行故障排除并且需要详细的日志转储。	无效的
`nfsMountOptions`	以逗号分隔的 NFS 挂载选项列表。Kubernetes 持久卷的挂载选项通常在存储类中指定，但如果存储类中没有指定挂载选项， Trident将回退到使用存储后端配置文件中指定的挂载选项。如果在存储类或配置文件中未指定任何挂载选项， Trident将不会在关联的持久卷上设置任何挂载选项。	""
`nasType`	配置 NFS 或 SMB 卷的创建。选项有 `nfs`， `smb`或空值。 *必须设置为 `smb`适用于 SMB 卷。*设置为 null 则默认使用 NFS 卷。	`nfs`
`qtreesPerFlexvol`	每个FlexVol volume的最大 Qtree 数量必须在 [50, 300] 范围内	`"200"`
`smbShare`	您可以指定以下名称之一：使用 Microsoft 管理控制台或ONTAP CLI 创建的 SMB 共享的名称，或者允许Trident创建 SMB 共享的名称。此参数是Amazon FSx for ONTAP后端所必需的。	`smb-share`
`useREST`	使用ONTAP REST API 的布尔参数。设置为 `true`Trident将使用ONTAP REST API 与后端进行通信。此功能需要ONTAP 9.11.1 及更高版本。此外，所使用的ONTAP登录角色必须具有访问权限。 `ontap`应用。预定义项满足了这一点。 `vsadmin`和 `cluster-admin`角色。	`false`

参数	描述	示例
aws	您可以在 AWS FSx for ONTAP的配置文件中指定以下内容： - fsxFilesystemID：指定 AWS FSx 文件系统的 ID。 - apiRegion AWS API 区域名称。 - apikey AWS API 密钥。 - secretKey AWS 密钥。	"" "" ""
credentials	指定要存储在 AWS Secrets Manager 中的 FSx SVM 凭证。 - name：包含 SVM 凭证的密钥的 Amazon 资源名称 (ARN)。 - type`设置为 `awsarn。请参阅"创建 AWS Secrets Manager 密钥"了解更多信息。	

卷配置的后端配置选项

您可以使用以下选项控制默认配置。 `defaults`配置部分。例如，请参见下面的配置示例。

参数	描述	默认
spaceAllocation	LUN 的空间分配	true
spaceReserve	空间预留模式；“无”（细）或“大量”（粗）	none
snapshotPolicy	要使用的快照策略	none
qosPolicy	要为创建的卷分配的 QoS 策略组。每个存储池或后端选择 qosPolicy 或 adaptiveQosPolicy 之一。将 QoS 策略组与Trident结合使用需要ONTAP 9.8 或更高版本。您应该使用非共享的 QoS 策略组，并确保该策略组单独应用于每个成员。共享的 QoS 策略组强制规定所有工作负载的总吞吐量上限。	""
adaptiveQosPolicy	要为创建的卷分配的自适应 QoS 策略组。每个存储池或后端选择 qosPolicy 或 adaptiveQosPolicy 之一。 ontap-nas-economy 不支持此功能。	""
snapshotReserve	为快照“0”预留的卷百分比	如果 snapshotPolicy`是 `none，else “”
splitOnClone	创建时将克隆体从其母体中分离出来	false

参数	描述	默认
`encryption`	在新卷上启用NetApp卷加密 (NVE)；默认设置为 `false`。要使用此选项，必须在集群上获得 NVE 许可并启用 NVE。如果后端启用了 NAE，则在Trident中配置的任何卷都将启用 NAE。更多信息，请参阅："Trident如何与 NVE 和 NAE 协同工作"。	`false`
`luksEncryption`	启用LUKS加密。参考"使用 Linux 统一密钥设置 (LUKS)"。仅限 SAN。	""
`tieringPolicy`	分层策略的使用 `none`	
`unixPermissions`	新卷模式。 **SMB**卷请留空。	""
`securityStyle`	新卷的安全样式。 NFS 支持 `mixed`和 `unix`安全措施。中小企业支持 `mixed`和 `ntfs`安全措施。	NFS 默认值为 `unix`。 SMB 默认值为 `ntfs`。

准备配置SMB卷

您可以使用以下方式配置 SMB 卷： `ontap-nas`司机。在你完成之前ONTAP SAN 和 NAS 驱动程序集成请完成以下步骤。

开始之前

在使用以下方式配置 SMB 卷之前： `ontap-nas`驾驶员，您必须具备以下条件。

- 一个 Kubernetes 集群，包含一个 Linux 控制器节点和至少一个运行 Windows Server 2019 的 Windows 工作节点。 Trident仅支持挂载到运行在 Windows 节点上的 pod 的 SMB 卷。
- 至少有一个包含您的 Active Directory 凭据的Trident密钥。生成秘密 `smbcreds`：

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- 配置为 Windows 服务的 CSI 代理。要配置 `csi-proxy`，请参阅"GitHub：CSI代理"或者"GitHub：适用于 Windows 的 CSI 代理"适用于在 Windows 上运行的 Kubernetes 节点。

步骤

1. 创建SMB共享。您可以通过以下两种方式之一创建 SMB 管理共享："Microsoft 管理控制台"共享文件夹管理单元或使用ONTAP CLI。使用ONTAP CLI 创建 SMB 共享：
   a. 如有必要，请创建共享的目录路径结构。

      这 `vserver cifs share create`该命令检查在创建共享时 -path 选项中指定的路径。如果指定的路径不存在，则命令执行失败。

   b. 创建与指定 SVM 关联的 SMB 共享：

```
vserver cifs share create -vserver vserver_name -share-name
share_name -path path [-share-properties share_properties,...]
[other_attributes] [-comment text]
```

c. 确认共享已创建：

```
vserver cifs share show -share-name share_name
```

请参阅"创建 SMB 共享"了解详细信息。

2. 创建后端时，必须配置以下内容以指定 SMB 卷。有关所有 FSx for ONTAP后端配置选项，请参阅"FSx for ONTAP配置选项和示例"。

参数	描述	示例
`smbShare`	您可以指定以下名称之一：使用 Microsoft 管理控制台或ONTAP CLI 创建的 SMB 共享的名称，或者允许Trident创建 SMB 共享的名称。此参数是Amazon FSx for ONTAP后端所必需的。	`smb-share`
`nasType`	*必须设置为 `smb`.*如果为空，则默认为空。 `nfs` 。	`smb`
`securityStyle`	新卷的安全样式。 必须设置为 **`ntfs`**或者 **`mixed`**适用于 **SMB** 卷。	`ntfs`或者 `mixed`适用于 SMB 卷
`unixPermissions`	新卷模式。 对于 **SMB** 卷，此项必须留空。	""

配置存储等级和PVC

配置 Kubernetes StorageClass 对象并创建存储类，以指示Trident如何配置卷。创建一个使用已配置的 Kubernetes StorageClass 的 PersistentVolumeClaim (PVC) 来请求访问 PV。然后您可以将光伏组件安装到支架上。

创建存储类

配置 **Kubernetes StorageClass** 对象

这 "Kubernetes StorageClass 对象"该对象将Trident标识为该类使用的配置器，并指示Trident如何配置卷。使用此示例为使用 NFS 的卷设置 Storageclass（有关属性的完整列表，请参阅下面的Trident属性部分）：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  provisioningType: "thin"
  snapshots: "true"
```

使用此示例为使用 iSCSI 的卷设置 Storageclass：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  provisioningType: "thin"
  snapshots: "true"
```

要在 AWS Bottlerocket 上配置 NFSv3 卷，请添加所需的 `mountOptions`到存储类：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
mountOptions:
  - nfsvers=3
  - nolock
```

请参阅"Kubernetes 和Trident对象"有关存储类如何与……交互的详细信息 `PersistentVolumeClaim`以及控制Trident如何分配容量的参数。

创建存储类

步骤

1. 这是一个 Kubernetes 对象，所以请使用 `kubectl`在 Kubernetes 中创建它。

```
kubectl create -f storage-class-ontapnas.yaml
```

2. 现在您应该在 Kubernetes 和Trident中都看到 **basic-csi** 存储类，并且Trident应该已经发现了后端上的存储池。

```
kubectl get sc basic-csi
```

```
NAME          PROVISIONER             AGE
basic-csi     csi.trident.netapp.io   15h
```

制作PVC管

一个 "*PersistentVolumeClaim*"（PVC）是对集群上持久卷的访问请求。

PVC 可以配置为请求存储特定尺寸或访问模式。使用关联的 StorageClass，集群管理员可以控制的不仅仅是 PersistentVolume 的大小和访问模式，还可以控制性能或服务级别。

制作好 PVC 后，就可以将容积安装到舱体中。

样品清单

PersistentVolumeClaim 样本清单

这些示例展示了PVC的基本配置选项。

带**RWX**接口的**PVC**

此示例展示了一个具有 RWX 访问权限的基本 PVC，它与一个名为 StorageClass 的 StorageClass 相关联。 `basic-csi` 。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-gold
```

使用 **iSCSI** 示例的 **PVC**

此示例展示了一个与名为 StorageClass 的存储类关联的、具有 RWO 访问权限的 iSCSI 基本 PVC。 `protection-gold` 。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: pvc-san
spec:
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 1Gi
storageClassName: protection-gold
```

创建 **PVC**

步骤

1. 创建 PVC。

```
kubectl create -f pvc.yaml
```

2. 核实PVC状态。

```
kubectl get pvc
```

```
NAME          STATUS VOLUME     CAPACITY ACCESS MODES STORAGECLASS AGE
pvc-storage Bound  pv-name 2Gi       RWO                          5m
```

请参阅"Kubernetes 和Trident对象"有关存储类如何与……交互的详细信息 `PersistentVolumeClaim`以及控制Trident如何分配容量的参数。

Trident属性

这些参数决定了应使用哪些 Trident 管理的存储池来配置给定类型的卷。

属性	类型	价值观	提供	要求	由……支持
媒体[1]	string	机械硬盘、混合硬盘、固态硬盘	Pool 包含此类媒体；混合型媒体是指两者兼具。	指定的媒体类型	ontap-nas 、ontap-nas-economy、ontap-nas-flexgroup 、ontap-san 、solidfire-san
供应类型	string	薄的，厚的	池支持这种配置方法	指定的配置方法	厚：全部 ontap ；薄：全部 ontap 和 solidfire-san
后端类型	string	ontap-nas 、ontap-nas-economy、ontap-nas-flexgroup 、ontap-san 、solidfire-san 、gcp-cvs 、azure-netapp-files、ontap-san-economy	池属于这种类型的后端	指定的后端	所有司机
snapshots	布尔值	真，假	存储池支持带快照的卷	已启用快照的卷	ontap-nas 、ontap-san 、solidfire-san 、gcp-cvs
个克隆	布尔值	真，假	存储池支持卷克隆	已启用克隆的卷	ontap-nas 、ontap-san 、solidfire-san 、gcp-cvs

属性	类型	价值观	提供	要求	由.....支持
加密	布尔值	真，假	存储池支持加密卷	已启用加密的卷	ontap-nas、ontap-nas-economy、ontap-nas-flexgroups、ontap-san
IOPS	整数	正整数	Pool 能够保证在此范围内的 IOPS	容量保证了这些IOPS	solidfire-san

[1]： ONTAP Select系统不支持此系统

部署示例应用程序

创建存储等级和 PVC 后，即可将光伏组件安装到舱体上。本节列出了将 PV 连接到 pod 的示例命令和配置。

步骤

1. 将音量旋钮安装在一个音控器中。

```
kubectl create -f pv-pod.yaml
```

以下示例展示了将PVC连接到舱体的基本配置：基本配置：

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
       claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage
```

您可以使用以下方式监控进度 `kubectl get pod --watch`。

2. 确认卷已挂载到 `/my/mount/path`。

```
kubectl exec -it pv-pod -- df -h /my/mount/path
```

```
Filesystem                                                      Size
Used  Avail Use%  Mounted on
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06  1.1G
320K  1.0G   1%   /my/mount/path
```

现在您可以删除该 Pod 了。 Pod 应用将不复存在，但卷将保留。

```
kubectl delete pod pv-pod
```

在 EKS 集群上配置Trident EKS 插件

NetApp Trident简化了 Kubernetes 中Amazon FSx for NetApp ONTAP存储管理，使您的开发人员和管理员能够专注于应用程序部署。 NetApp Trident EKS 插件包含最新的安全补丁和错误修复，并经过 AWS 验证，可与 Amazon EKS 配合使用。 EKS 插件使您能够持续确保您的 Amazon EKS 集群安全稳定，并减少安装、配置和更新插件所需的工作量。

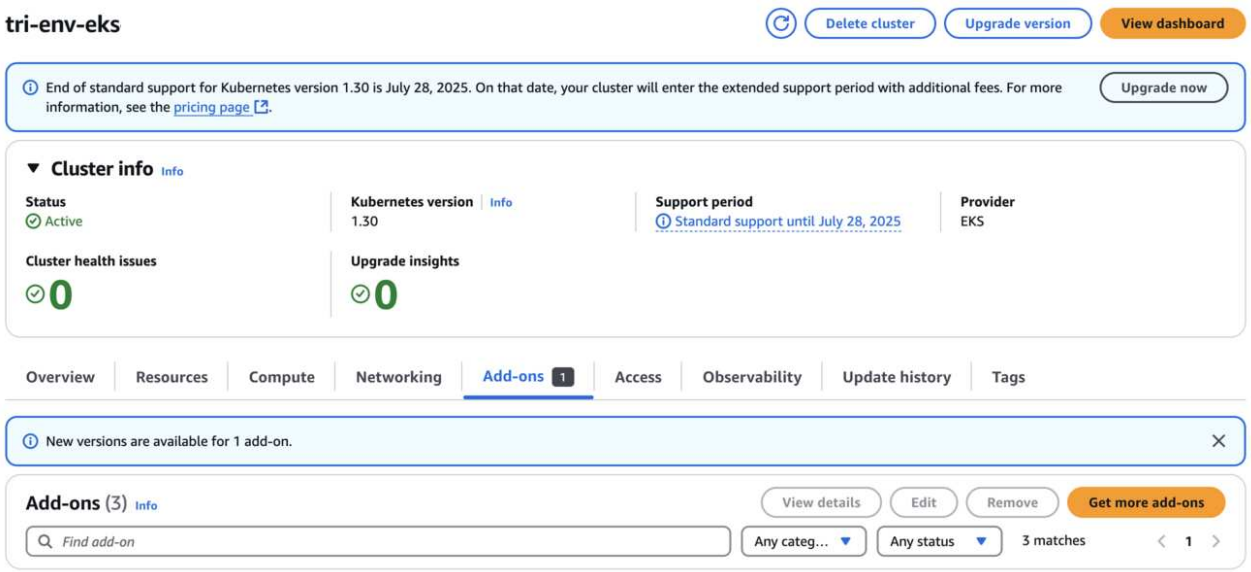
前提条件

在为 AWS EKS 配置Trident插件之前，请确保您已具备以下条件：

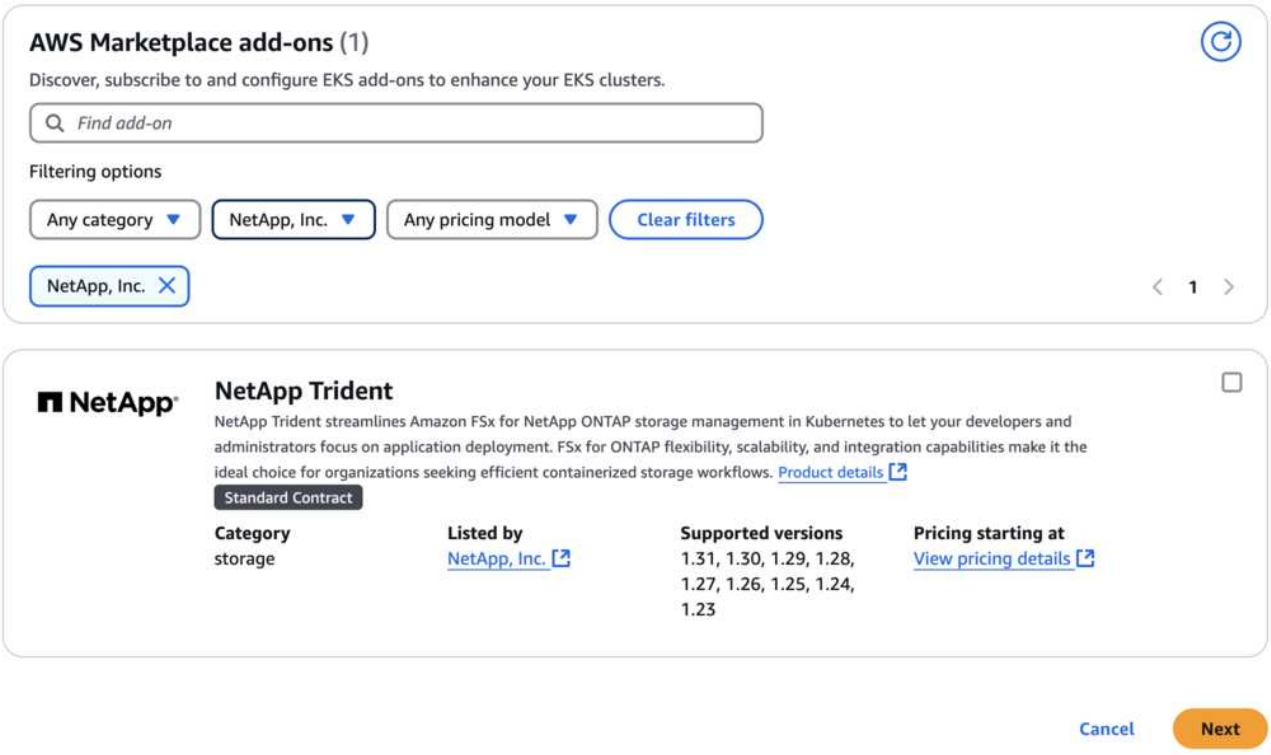
- 具有使用插件权限的 Amazon EKS 集群账户。参考"Amazon EKS 附加组件"。
- AWS 对 AWS Marketplace 的权限：
  `"aws-marketplace:ViewSubscriptions",`
  `"aws-marketplace:Subscribe",`
  `"aws-marketplace:Unsubscribe`
- AMI 类型：Amazon Linux 2 (AL2_x86_64) 或 Amazon Linux 2 Arm (AL2_ARM_64)
- 节点类型：AMD 或ARM
- 现有的Amazon FSx for NetApp ONTAP文件系统

步骤

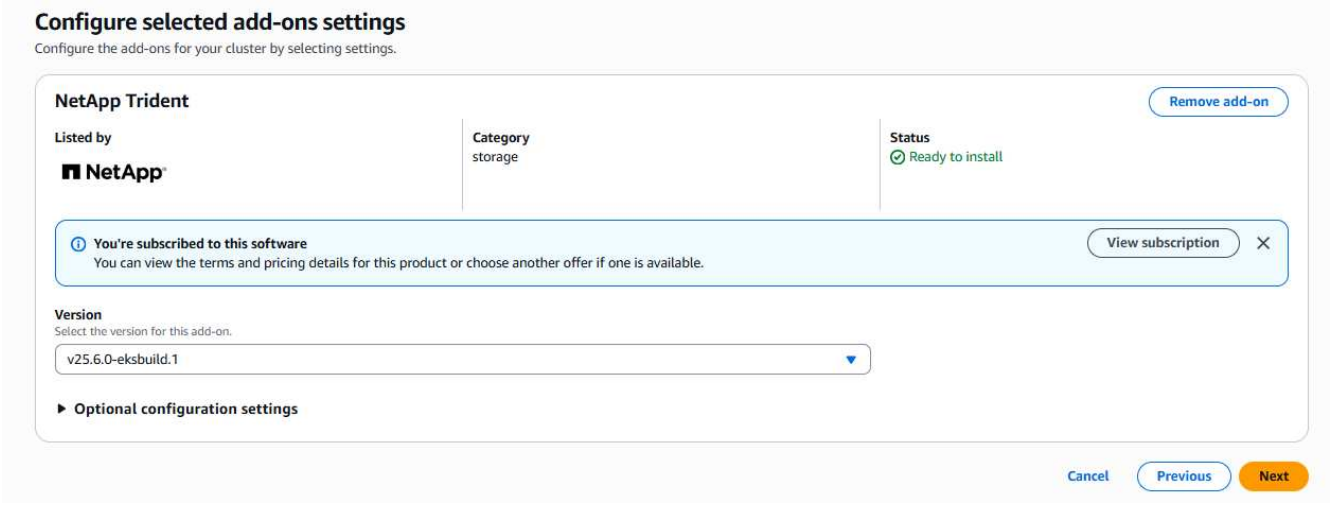
1. 请务必创建 IAM 角色和 AWS 密钥，以使 EKS pod 能够访问 AWS 资源。有关说明，请参阅"创建 IAM 角色和 AWS Secret"。
2. 在 EKS Kubernetes 集群上，导航到“附加组件”选项卡。



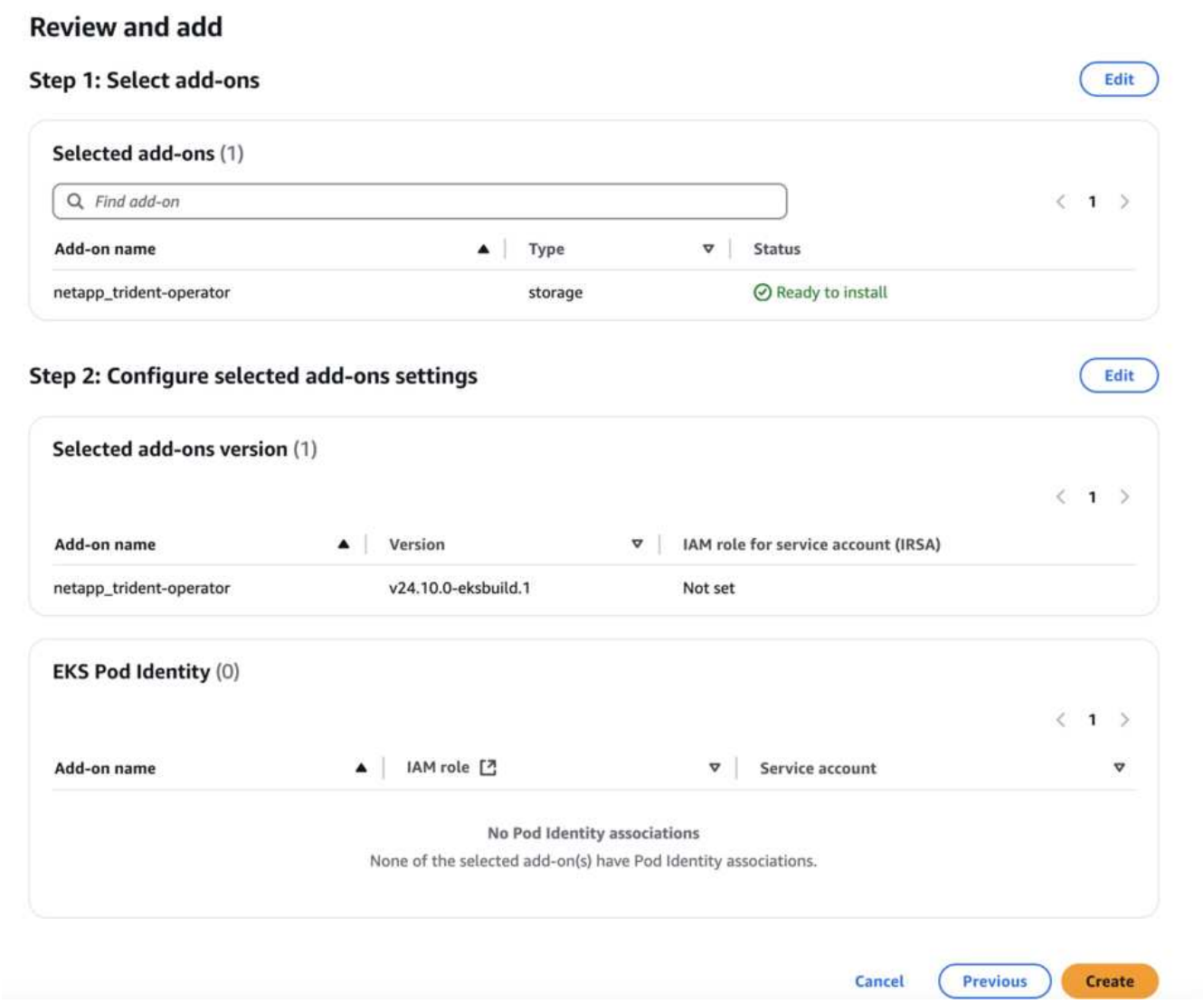
3. 前往 **AWS Marketplace** 插件，然后选择 *storage* 类别。



4. 找到 * NetApp Trident*，选中Trident插件的复选框，然后单击 下一步。
5. 选择所需的插件版本。

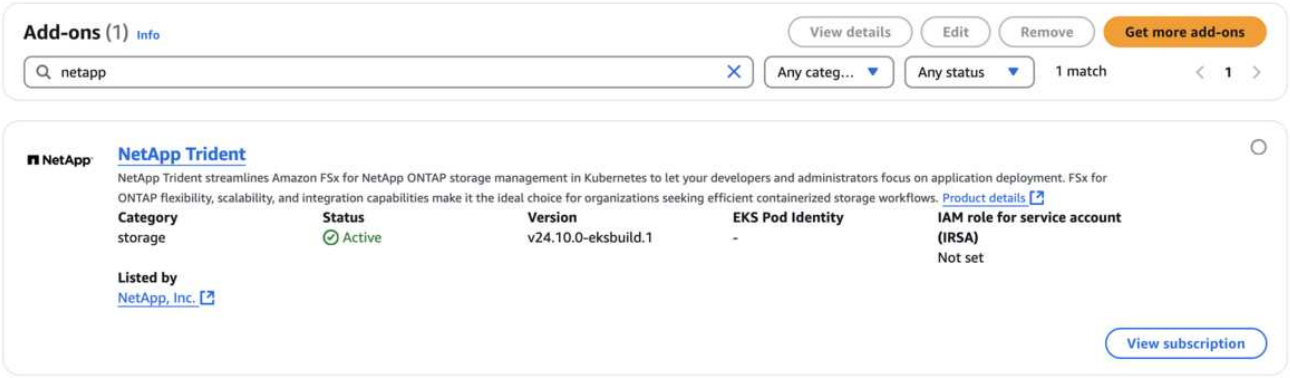


6. 配置所需的附加组件设置。



7. 如果您使用的是 IRSA（服务帐户的 IAM 角色），请参阅其他配置步骤。"此处" 。
8. 选择“创建”。

9. 确认插件状态为_Active_。



10. 运行以下命令以验证Trident是否已正确安装在集群上：

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

11. 继续进行设置并配置存储后端。有关信息，请参阅"配置存储后端"。

使用 CLI 安装/卸载Trident EKS 插件

使用 **CLI** 安装**NetApp Trident EKS** 插件：

以下示例命令安装Trident EKS 插件：

```
eksctl create addon --cluster clusterName --name netapp_trident-operator
--version v25.6.0-eksbuild.1 （另有专用版本）
```

使用 **CLI** 卸载**NetApp Trident EKS** 插件：

以下命令卸载Trident EKS 插件：

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```