



Amazon FSx for NetApp ONTAP

Trident

NetApp
October 29, 2026

目录

Amazon FSx for NetApp ONTAP	1
将 Trident 与 Amazon FSx for NetApp ONTAP 结合使用	1
要求	1
注意事项	1
身份验证	2
已测试的 Amazon Machine Images (AMIs)	2
查找更多信息	3
创建 IAM 角色和 AWS Secret	3
创建 AWS Secrets Manager 密钥	3
创建 IAM 策略	4
安装 Trident	7
通过 helm 安装 Trident	8
通过 EKS 附加组件安装 Trident	9
配置存储类	14
创建 StorageClass 对象	14
配置 SMB 卷	21
后端高级配置和示例	21
用于配置卷的后端配置选项	24
配置 PVC	25
创建 PVC	25
部署应用程序	27
部署示例应用程序	27
在 EKS 集群上配置 Trident EKS 附加组件	28
前提条件	28
步骤	29
使用 CLI 安装/卸载 Trident EKS 附加组件	31

Amazon FSx for NetApp ONTAP

将 Trident 与 Amazon FSx for NetApp ONTAP 结合使用

"Amazon FSx for NetApp ONTAP" 是一项完全托管的 AWS 服务，运行由 NetApp ONTAP 存储操作系统支持的文件系统。它提供 ONTAP 功能、性能和管理，具有 AWS 的可扩展性和操作简便性。文件系统是 Amazon FSx 中的主要资源，类似于本地 ONTAP 集群。每个文件系统包含一个或多个存储虚拟机 (SVM)，每个 SVM 包含一个或多个存储文件和目录的卷。这种集成使在 Amazon Elastic Kubernetes Service (EKS) 中运行的 Kubernetes 集群能够为块和文件工作负载配置 ONTAP 支持的持久卷。

要求

除了["Trident 要求"](#)，要将 FSx for ONTAP 与 Trident 集成，您还需要：

- 已安装 `kubectl` 的现有 Amazon EKS 集群或自我管理的 Kubernetes 集群。
- 可从群集的工作节点访问的现有 Amazon FSx for NetApp ONTAP 文件系统和 Storage Virtual Machine (SVM)。
- 已为 ["NFS 或 iSCSI"](#) 准备好的工作节点。



请确保遵循 Amazon Linux 和 Ubuntu ["Amazon Machine Images"](#) (AMI) 所需的节点准备步骤，具体取决于您的 EKS AMI 类型。

注意事项

- SMB 卷：
 - 仅使用 `ontap-nas` 驱动程序支持 SMB 卷。
 - Trident EKS 加载项不支持 SMB 卷。
 - Trident 仅支持将 SMB 卷挂载到在 Windows 节点上运行的 Pod。有关详细信息，请参阅 ["准备配置 SMB 卷"](#)。
- 在 Trident 24.02 之前，在启用了自动备份的 Amazon FSx 文件系统上创建的卷无法被 Trident 删除。要防止 Trident 24.02 或更高版本中的此问题，请在 AWS FSx for ONTAP 的后端配置文件中指定 `fsxFilesystemID`、`AWS apiRegion`、`AWS apikey` 和 `AWS secretKey`。



如果要为 Trident 指定 IAM 角色，则可以省略为 Trident 明确指定 `apiRegion`、``apiKey`` 和 ``secretKey`` 字段。有关详细信息，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。此示例显示了一个 ``multipath.conf`` 文件，其中包含所需的 Trident 设置，同时将 EBS 磁盘排除在多路径之外：

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

身份验证

Trident 提供了两种身份验证模式。

- 基于凭据（推荐）：在 AWS Secrets Manager 中安全地存储凭据。您可以使用文件系统的 `fsxadmin` 用户或为 SVM 配置的 `vsadmin` 用户。



Trident 希望以 `vsadmin` SVM 用户身份或以具有相同角色的不同名称的用户身份运行。Amazon FSx for NetApp ONTAP 有一个 `fsxadmin` 用户，它是 ONTAP admin 集群用户的有限替代品。我们强烈建议将 `vsadmin` 与 Trident 一起使用。

- 基于证书：Trident 将使用 SVM 上安装的证书与 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：

- ["ONTAP NAS 身份验证"](#)
- ["ONTAP SAN 身份验证"](#)

已测试的 Amazon Machine Images (AMIs)

EKS 集群支持各种操作系统，但 AWS 已针对容器和 EKS 优化了某些 Amazon Machine Images (AMI)。以下 AMI 已通过 NetApp Trident 25.02 测试。

AMI	NAS	NAS-经济型	iSCSI	iSCSI-经济性
AL2023_x86_64_STANDARD	是	是	是	是
AL2_x86_64	是	是	是*	是*
BOTTLEROCKET_x86_64	是**	是	不适用	不适用
AL2023_ARM_64_STANDARD	是	是	是	是
AL2_ARM_64	是	是	是*	是*
BOTTLEROCKET_ARM_64	是**	是	不适用	不适用

- * 如果不重新启动节点，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，这并不意味着它不受支持；它只是意味着它尚未经过测试。此列表可作为已知有效的 AMI 的指南。

执行测试使用：

- EKS 版本：1.32
- 安装方法：Helm 25.06 和作为 AWS 插件 25.06
- 对于 NAS，已测试 NFSv3 和 NFSv4.1。
- 对于 SAN，仅测试了 iSCSI，而未测试 NVMe-oF。

已执行的测试：

- 创建：Storage Class、pvc、pod
- 删除：pod、pvc（常规、qtree/lun – economy、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

您可以通过作为 AWS IAM 角色进行身份验证，而不是提供明确的 AWS 凭据，来配置 Kubernetes Pod 以访问 AWS 资源。



要使用 AWS IAM 角色进行身份验证，必须使用 EKS 部署 Kubernetes 集群。

创建 AWS Secrets Manager 密钥

由于 Trident 将针对 FSx vserver 发布 API 来为您管理存储，因此需要凭据才能执行此操作。传递这些凭据的安全方法是通过 AWS Secrets Manager 密钥。因此，如果您还没有 AWS Secrets Manager 密钥，则需要创建包含 vsadmin 帐户凭据的 AWS Secrets Manager 密钥。

此示例创建 AWS Secrets Manager 密码以存储 Trident CSI 凭据：

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

创建 IAM 策略

Trident 还需要 AWS 权限才能正常运行。因此，您需要创建一个策略，为 Trident 提供所需的权限。

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"  
}
```

为服务帐户关联 (IRSA) 创建 Pod Identity 或 IAM 角色

您可以将 Kubernetes 服务帐户配置为具有 EKS Pod Identity 或 IAM role for Service account association (IRSA) 的 AWS Identity and Access Management (IAM) 角色。然后，配置为使用服务帐户的任何 Pod 都可以访问角色有权访问的任何 AWS 服务。

Pod 身份

Amazon EKS Pod Identity 关联提供了管理应用程序凭据的功能，类似于 Amazon EC2 实例配置文件向 Amazon EC2 实例提供凭据的方式。

在 EKS 集群上安装 Pod Identity:

您可以通过 AWS 控制台或使用以下 AWS CLI 命令创建 Pod 标识:

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

有关详细信息, 请参阅 ["设置 Amazon EKS Pod Identity Agent"](#)。

创建 trust-relationship.json:

创建 trust-relationship.json 以启用 EKS Service Principal 承担 Pod Identity 的此角色。然后使用此信任策略创建角色:

```
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:

将上一步中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** 标识关联:

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 `trust-relationship.json` 文件中的以下值：

- **<account_id>** - 您的 AWS 账户 ID
- **<oidc_provider>** - 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** 策略绑定：

创建角色后，使用此命令将策略（在上述步骤中创建的）附加到角色：

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

验证 **OIDC** 提供者是否关联：

验证您的 OIDC 提供程序是否与您的集群关联。您可以使用以下命令验证它：

```
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 关联到您的群集：

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

如果使用 **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：

- Helm
- EKS 附加项

如果要使用快照功能，请安装 CSI 快照控制器加载项。有关详细信息，请参见["为 CSI 卷启用快照功能"](#)。

通过 helm 安装 Trident

Pod 身份

1. 添加 Trident Helm 存储库：

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

2. 请使用以下示例安装 Trident：

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

您可以使用 `helm list` 命令查看安装详细信息，例如名称、命名空间、图表、状态、应用版本和修订版号。

```
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		

服务账户关联 (IRSA)

1. 添加 Trident Helm 存储库：

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

2. 设置 **cloud provider** 和 **cloud identity** 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` 命令查看安装详细信息，例如名称、命名空间、图表、状态、应用版本和修订版号。

```
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-
100.2510.0	25.10.0		

如果您计划使用 iSCSI，请确保在您的客户端计算机上启用了 iSCSI。如果您使用的是 AL2023 Worker 节点操作系统，则可以通过在 `helm` 安装中添加 `node prep` 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 附加组件安装 Trident

Trident EKS 加载项包含最新的安全补丁、错误修复，并经过 AWS 验证可与 Amazon EKS 配合使用。EKS 加载项使您能够始终如一地确保 Amazon EKS 集群的安全和稳定，并减少安装、配置和更新加载项所需的工作量。

前提条件

在为 AWS EKS 配置 Trident 加载项之前，请确保具备以下条件：

- 具有附加订阅的 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 Amazon FSx for NetApp ONTAP 文件系统

为 AWS 启用 Trident 加载项

管理控制台

1. 在 <https://console.aws.amazon.com/eks/home#/clusters> 打开 Amazon EKS 控制台。
2. 在左侧导航窗格中，选择 **Clusters**。
3. 选择要为其配置 NetApp Trident CSI 加载项的群集的名称。
4. 选择 **Add-ons**，然后选择 **Get more add-ons**。
5. 按照以下步骤选择附加组件：
 - a. 向下滚动到 **AWS Marketplace add-ons** 部分，然后在搜索框中键入 **"Trident"**。
 - b. 选中 Trident by NetApp 框右上角的复选框。
 - c. 选择 **Next**。

6. 在 **Configure selected add-ons** 设置页面上，执行以下操作：



如果您使用的是 **Pod Identity** 关联，请跳过这些步骤。

- a. 选择您想要使用的 **Version**。
- b. 如果使用 IRSA 身份验证，请确保在可选配置设置中设置可用的配置值：
 - 选择您想要使用的 **Version**。
 - 按照*附加组件配置模式*，将*配置值*部分的*configurationValues*参数设置为您在上一步骤中创建的角色 arn（值应采用以下格式）：

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

+

如果为冲突解决方法选择覆盖，则可以使用 Amazon EKS 加载项设置覆盖现有加载项的一个或多个设置。如果未启用此选项，并且与现有设置存在冲突，则操作将失败。您可以使用生成的错误消息来排除冲突。在选择此选项之前，请确保 Amazon EKS 加载项不会管理您需要自我管理的设置。

7. 选择 **Next**。
8. 在 **Review and add** 页面上，选择 **Create**。

加载项安装完成后，您将看到已安装的加载项。

AWS CLI

1. 创建 `add-on.json` 文件：

对于 **Pod Identity**，请使用以下格式：



使用

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

对于 **IRSA** 身份验证，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"
  }
}
```



将 <role ARN> 替换为上一步中创建的角色 ARN。

2. 安装 Trident EKS 附加组件。

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

eksctl

以下示例命令安装 Trident EKS 加载项：

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

更新 Trident EKS 附加组件

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 **Clusters**。
3. 选择要为其更新 NetApp Trident CSI 加载项的群集的名称。
4. 选择 **Add-ons** 选项卡。
5. 选择 **Trident by NetApp**，然后选择 **Edit**。
6. 在 **Configure Trident by NetApp** 页面上，执行以下操作：
 - a. 选择您想要使用的 **Version**。
 - b. 展开 **Optional configuration settings** 并根据需要进行修改。
 - c. 选择 ***Save changes***。

AWS CLI

以下示例更新 EKS 加载项：

```
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

- 检查 FSxN Trident CSI 附加组件的当前版本。将 `my-cluster` 替换为您的集群名称。

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

输出示例：

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'\"}			

- 将加载项更新到上一步输出中的 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
```

如果您删除该 `--force` 选项，并且任何 Amazon EKS 加载项设置与现有设置冲突，则更新 Amazon EKS 加载项失败；您会收到一条错误消息，以帮助您解决冲突。在指定此选项之前，请确保 Amazon EKS 加载项不管您需要管理的设置，因为这些设置会被此选项覆盖。有关此设置的其他选项的详细信息，请参见 "[插件](#)"。有关 Amazon EKS Kubernetes 字段管理的详细信息，请参见 "[Kubernetes 字段管理](#)"。

卸载/删除 Trident EKS 插件

您有两种删除 Amazon EKS 加载项的选项：

- 保留集群上的附加软件 – 此选项可删除任何设置的 Amazon EKS 管理。它还删除了 Amazon EKS 通知您更新并在您启动更新后自动更新 Amazon EKS 加载项的功能。但是，它会保留集群上的加载项软件。此选项使加载项成为自我管理的安装，而不是 Amazon EKS 加载项。使用此选项，附加组件不会停机。保留命令中的 `--preserve` 选项以保留加载项。
- 从集群中完全移除附加软件 – NetApp 建议仅当您的集群上没有依赖该附加组件的资源时，才从集群中移除 Amazon EKS 附加组件。从 `--preserve` 命令中移除 `delete` 选项以移除该附加组件。



如果附加组件具有关联的 IAM 帐户，则不会删除该 IAM 帐户。

管理控制台

1. 在 <https://console.aws.amazon.com/eks/home#/clusters> 打开 Amazon EKS 控制台。
2. 在左侧导航窗格中，选择 **Clusters**。
3. 选择要删除其 NetApp Trident CSI 加载项的群集的名称。
4. 选择 **Add-ons** 选项卡，然后选择 **Trident by NetApp**。
5. 选择 **Remove**。
6. 在 删除 `netapp_trident-operator` 确认 对话框中，执行以下操作：
 - a. 如果希望 Amazon EKS 停止管理加载项的设置，请选择 **Preserve on cluster**。如果要将加载项软件保留在集群上，以便可以自行管理加载项的所有设置，请执行此操作。
 - b. 输入 `netapp_trident-operator`。
 - c. 选择 **Remove**。

AWS CLI

将 `my-cluster` 替换为群集的名称，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 加载项：

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

配置存储类

`https://kubernetes.io/docs/concepts/storage/storage-classes/` ["Kubernetes StorageClass 对象"^] 标识置备程序并指示置备程序如何置备卷。本节将向您展示如何配置将 Trident 指定为置备程序的 Kubernetes StorageClass 对象。

创建 StorageClass 对象

当您为 FSx for ONTAP 创建 StorageClass 时，Trident 将自动创建后端配置。



如果要手动配置存储后端，请参阅 [创建没有自动后端配置的 Kubernetes StorageClass](#) 部分了解如何分别创建 Trident 后端和存储类。

指定所需 StorageClass 参数

创建 StorageClass 时需要定义以下三个参数：

参数	必填项	类型	说明
<code>fsxFilesystemID</code>	是	string	FSx for NetApp ONTAP 文件系统 ID
<code>storageDriverName</code>	是	string	Trident 存储驱动程序（例如， <code>ontap-nas</code> 或 <code>ontap-san</code> ）
<code>credentialsName</code>	是	string	包含 FSx for ONTAP 凭据的 Kubernetes Secret 的名称

指定可选参数

您可以通过 StorageClass 传递可选的后端参数。在 StorageClass parameters 部分中将所有可选值定义为字符串。有关后端参数的完整列表，请参阅：["FSx for NetApp ONTAP 后端配置"](#)。

示例 StorageClass 配置文件。

以下示例显示了触发自动后端配置的 StorageClass。

YAML

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-fsx-demo
  annotations:
    description: "Demo StorageClass for FSx for NetApp ONTAP"
provisioner: csi.trident.netapp.io
parameters:
  fsxFilesystemID: "fs-0abc123"
  storageDriverName: "ontap-nas"
```

```
credentialsName: trident-fsx-credentials
allowVolumeExpansion: true
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

JSON

```
{
  "apiVersion": "storage.k8s.io/v1",
  "kind": "StorageClass",
  "metadata": {
    "name": "ontap-fsx-demo",
    "annotations": {
      "description": "Demo StorageClass for FSx for NetApp ONTAP"
    }
  },
  "provisioner": "csi.trident.netapp.io",
  "parameters": {
    "fsxFilesystemID": "fs-0abc123",
    "storageDriverName": "ontap-nas",
    "credentialsName": "trident-fsx-credentials"
  },
  "allowVolumeExpansion": true,
  "reclaimPolicy": "Delete",
  "volumeBindingMode": "Immediate"
}
```

创建 StorageClass

创建配置文件后，请运行以下命令来创建存储类。

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

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的池。

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

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

应用 StorageClass 后，Trident 会自动创建后端。然后，您可以创建引用此 StorageClass 的 PersistentVolumeClaims。

Trident 在 StorageClass 注释中记录后端创建的结果。

标注	说明
trident.netapp.io/configuratorStatus	配置结果 (Success 或 Failure)
trident.netapp.io/configuratorMessage	详细状态或错误消息
trident.netapp.io/configuratorName	内部配置器资源的名称
trident.netapp.io/managed	表示 StorageClass 由 Trident 管理
trident.netapp.io/additionalStoragePools	为此后端创建的存储池

要验证状态，请运行：

```
kubectl get storageclass ontap-fsx-demo -o yaml
```

确认 trident.netapp.io/configuratorStatus`已设置为 `Success。如果值为 Failure，请检查`trident.netapp.io/configuratorMessage`以了解错误。

添加其他 FSxN 文件系统

如果在继续使用相同的 StorageClass 的同时需要额外的存储容量，请添加其他 FSxN 文件系统 ID。

编辑 StorageClass 并添加以下注释：

```
metadata:
  annotations:
    trident.netapp.io/additionalFsxnFileSystemID: '["fs-
xxxxxxxxxxxxxxxxxxxxx"]'
```

应用更改后，Trident 更新后端配置并更新 StorageClass 注释。

操作注意事项和限制

- 删除具有自动后端配置的 StorageClass 通常会删除关联的 Trident 后端。这可能会中断存储连接并中断正在运行的工作负载。在删除托管的 StorageClass 之前，请验证影响。
- 只有适用于 NetApp ONTAP 的 AWS FSx 才支持自动后端配置。

创建没有自动后端配置的 Kubernetes StorageClass

如果要单独创建 Trident 后端和 StorageClass，请按照以下步骤操作。

了解自动后端配置的工作原理

Trident 从 StorageClass 定义中导出后端配置。当您应用 StorageClass 时，Trident 验证所需的参数，创建后端，并使用状态注释 StorageClass。

Trident 只创建一次 VolumeSnapshotClass。Trident 会为后续的 StorageClasses 重复使用同一个 VolumeSnapshotClass。

创建 Trident 后端

要创建 Trident 后端，需要创建 JSON 或 YAML 格式的配置文件。文件需要指定所需的存储类型（NAS 或 SAN）、文件系统和获取该文件的 SVM 以及使用该文件进行身份验证的方式。以下示例演示如何定义基于 NAS 的存储并使用 AWS 密钥将凭据存储到要使用的 SVM：

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:xxxxxxx: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:xxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}

```

FSx for ONTAP 驱动程序详细信息

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

驱动程序名称	说明
ontap-san	每个配置的 PV 都是其自己的 Amazon FSx for NetApp ONTAP 卷中的一个 LUN。推荐用于块存储。
ontap-nas	配置的每个 PV 都是完整的 Amazon FSx for NetApp ONTAP 卷。推荐用于 NFS 和 SMB。
ontap-san-economy	每个配置的 PV 都是一个 LUN，每个 Amazon FSx for NetApp ONTAP 卷具有可配置数量的 LUN。
ontap-nas-economy	配置的每个 PV 都是一个 qtree，每个 Amazon FSx for NetApp ONTAP 卷具有可配置数量的 qtree。
ontap-nas-flexgroup	每个已配置的 PV 都是一个完整的 Amazon FSx for NetApp ONTAP FlexGroup 卷。

有关驱动程序的详细信息，请参阅 ["NAS 驱动程序"](#) 和 ["SAN 驱动程序"](#)。

创建后端

创建配置文件后，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
```

- 验证已成功创建 Trident 后端配置 (TBC)：

```
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	

有关其他配置选项的详细信息，请参见以下 [后端高级配置和示例](#) 部分。

配置存储类*无*自动后端配置

以下是与 Trident 和 FSx for ONTAP 配合使用的存储类配置示例。

NFS 存储类

您可以使用此示例为使用 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"
```

iSCSI 的存储类

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"
```

使用 NFSv3 和 AWS Bottlerocket 的存储类

要在 AWS Bottlerocket 上配置 NFSv3 卷，请将所需的 `mountOptions` 添加到存储类：

```
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

```

Trident StorageClass 属性

这些参数确定应使用哪些 Trident 管理的存储池来配置给定类型的卷。

属性	类型	值	提供	请求	支持方
媒体 ¹	string	hdd、hybrid、ssd	池包含此类型的介质；混合意味着两者兼有	指定媒体类型	ontap-nas ， ontap-nas-economy ， ontap-nas-flexgroup ， ontap-san ， solidfire-san
provisioningType	string	薄、厚	池支持此配置方法	已指定配置方法	thick：所有 ONTAP；thin：所有 ONTAP 和 solidfire-san
backendType	string	ontap-nas ， ontap-nas-economy ， ontap-nas-flexgroup ， ontap-san ， solidfire-san ， azure-netapp-files, ontap-san-economy	池属于此类型的后端	指定后端	所有驱动程序
snapshots	布尔值	true, false	池支持具有快照的卷	已启用快照的卷	ontap-nas 、 ontap-san 、 solidfire-san
个克隆	布尔值	true, false	池支持克隆卷	已启用克隆的卷	ontap-nas 、 ontap-san 、 solidfire-san

属性	类型	值	提供	请求	支持方
加密	布尔值	true, false	池支持加密卷	已启用加密的卷	ontap-nas , ontap-nas-economy , ontap-nas-flexgroups , ontap-san
IOPS	int	正整数	池能够保证此范围内的 IOPS	卷保证这些 IOPS	solidfire-san

1: ONTAP Select 或 FSx for ONTAP 系统不支持

请参阅["Kubernetes 和 Trident 对象"](#)，了解存储类如何与 `PersistentVolumeClaim` 交互，以及用于控制 Trident 配置卷的参数详情。

创建存储类

配置 StorageClass 后，您可以在 Kubernetes 中创建它。

步骤

1. 这是一个 Kubernetes 对象，因此请使用 `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的池。

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

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

配置 SMB 卷

您可以使用 `ontap-nas` 驱动程序配置 SMB 卷。但是，要执行此操作，您必须完成以下步骤：["准备配置 SMB 卷"](#)。

后端高级配置和示例

有关后端配置选项，请参见下表：

参数	说明	示例
version		始终为 1

参数	说明	示例
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，也可以指定完全限定域名（FQDN）。如果使用 IPv6 标志安装了 Trident，则可以设置为使用 IPv6 地址。IPv6 地址必须以方括号定义，例如 [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。如果你在 fsxFilesystemID`下的 `aws` 字段中提供了 `managementLIF`，则无需再提供 managementLIF，因为 Trident 会从 AWS 检索 SVM 信息。因此，你必须为 SVM 下的用户（例如：vsadmin）提供凭据，并且该用户必须具有 `vsadmin` 角色。	"10.0.0.1", "[2001:1234:abcd::fefe]"
dataLIF	协议 LIF 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。可以指定要用于 NFS 挂载操作的完全限定域名 (FQDN)，允许您创建轮询 DNS 以跨多个 dataLIF 进行负载平衡。可以在初始设置后更改。 ONTAP SAN 驱动程序：不要为 iSCSI 指定。Trident 使用 ONTAP Selective LUN Map 来发现建立多路径会话所需的 iSCSI LIF。如果明确定义了 dataLIF，则会生成警告。如果使用 IPv6 标志安装了 Trident，则可以设置为使用 IPv6 地址。IPv6 地址必须以方括号定义，例如 [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。	
autoExportPolicy	启用自动导出策略创建和更新 [Boolean]。使用 `autoExportPolicy` 和 `autoExportCIDRs` 选项，Trident 可以自动管理导出策略。	false
autoExportCIDRs	启用 `autoExportPolicy` 时用于过滤 Kubernetes 节点 IP 的 CIDR 列表。使用 `autoExportPolicy` 和 `autoExportCIDRs` 选项，Trident 可以自动管理导出策略。	"["0.0.0.0/0", "::/0"]"
labels	要应用于卷的任意 JSON 格式标签集	""

参数	说明	示例
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。用于基于凭据的身份验证。例如，vsadmin。	
password	连接到集群或 SVM 的密码。用于基于凭据的身份验证。	
svm	要使用的 Storage Virtual Machine	如果指定了 SVM managementLIF，则派生。
storagePrefix	在 SVM 中预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
limitAggregateUsage	*请勿为 Amazon FSx for NetApp ONTAP 指定。*提供的 `fsxadmin` 和 `vsadmin` 不包含检索聚合使用情况并使用 Trident 限制它所需的权限。	请勿使用。
limitVolumeSize	如果请求的卷大小高于此值，则设置失败。还限制其为 qtree 和 LUN 管理的卷的最大大小，并且该 `qtreesPerFlexvol` 选项允许自定义每个 FlexVol 卷的最大 qtree 数	"（默认情况下不强制执行）
lunsPerFlexvol	每个 FlexVol volume 的最大 LUN 数必须在 [50, 200] 范围内。仅限 SAN。	"100"
debugTraceFlags	故障排除时使用的调试标志。例如，{"api":false, "method":true} 除非正在进行故障排除并需要详细的日志转储，否则不要使用 debugTraceFlags。	空
nfsMountOptions	NFS 挂载选项的逗号分隔列表。Kubernetes 持久卷的挂载选项通常在存储类中指定，但如果存储类中未指定挂载选项，则 Trident 将回退到使用存储后端配置文件中指定的挂载选项。如果存储类或配置文件中未指定挂载选项，Trident 将不会在关联的持久卷上设置任何挂载选项。	""

参数	说明	示例
nasType	配置 NFS 或 SMB 卷创建。选项为 nfs、smb 或 null。*对于 SMB 卷，必须设置为 `smb`。*设置为 null 默认为 NFS 卷。	nfs
qtreesPerFlexvol	每个 FlexVol 卷的最大 Qtree 数，必须在 [50, 300] 范围内	"200"
smbShare	您可以指定以下选项之一：使用 Microsoft Management Console 或 ONTAP CLI 创建的 SMB 共享的名称，或允许 Trident 创建 SMB 共享的名称。Amazon FSx for ONTAP 后端需要此参数。	smb-share
useREST	使用 ONTAP REST API 的布尔参数。当设置为 `true` 时，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 登录角色必须能够访问 `ontap` 应用程序。这通过预定义的 `vsadmin` 和 `cluster-admin` 角色来满足。	false
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 凭据。 - name：机密的 Amazon 资源名称 (ARN)，其中包含 SVM 的凭据。 - type：设置为 awsarn。有关详细信息，请参见 "创建 AWS Secrets Manager 密钥" 。	

用于配置卷的后端配置选项

您可以使用配置的 defaults 部分中的这些选项来控制默认配置。有关示例，请参阅下面的配置示例。

参数	说明	默认
spaceAllocation	LUN 的空间分配	true
spaceReserve	空间预留模式；"none"（精简）或 "volume"（厚）	none
snapshotPolicy	要使用的 Snapshot 策略	none

参数	说明	默认
qosPolicy	要为创建的卷分配的 QoS 策略组。为每个存储池或后端选择 qosPolicy 或 adaptiveQosPolicy 中的一个。在 Trident 中使用 QoS 策略组需要 ONTAP 9.8 或更高版本。您应该使用非共享 QoS 策略组，并确保该策略组单独应用于每个组成部分。共享 QoS 策略组强制执行所有工作负载总吞吐量的上限。	""
adaptiveQosPolicy	要为创建的卷分配的自适应 QoS 策略组。为每个存储池或后端选择 qosPolicy 或 adaptiveQosPolicy 中的一个。ontap-nas-economy 不支持此功能。	""
snapshotReserve	为快照"0"保留的卷的百分比	如果 snapshotPolicy 是 none， else ""
splitOnClone	创建时从其父级拆分克隆	false
encryption	在新卷上启用 NetApp 卷加密 (NVE)；默认为 false。必须在集群上获得许可并启用 NVE 才能使用此选项。如果在后端启用了 NAE，则在 Trident 中配置的任何卷都将启用 NAE。有关详细信息，请参阅： "Trident 如何与 NVE 和 NAE 配合使用" 。	false
luksEncryption	启用 LUKS 加密。请参阅 "使用 Linux Unified Key Setup (LUKS)" 。仅限 SAN。	""
tieringPolicy	要使用的分层策略 none	
unixPermissions	新卷的模式。对于 SMB 卷，请留空。	""
securityStyle	新卷的安全样式。NFS 支持 `mixed` 和 `unix` 安全样式。SMB 支持 `mixed` 和 `ntfs` 安全样式。	NFS 默认值为 unix。SMB 默认值为 ntfs。

配置 PVC

本节包含有关如何创建使用已配置 Kubernetes StorageClass 来请求 PV 的 PersistentVolumeClaim (PVC) 的说明。如果成功，您随后可以将该 PV 挂载到 pod。

创建 PVC

```
https://kubernetes.io/docs/concepts/storage/persistent-volumes["_PersistentVolumeClaim_"] (PVC) 是访问集群上 PersistentVolume
```

的请求。PVC 可以配置为请求特定大小或访问模式的存储。使用关联的 StorageClass，集群管理员可以控制超出 PersistentVolume 大小和访问模式的更多内容——例如性能或服务级别。

创建 Trident 后端和 StorageClass 后，您可以创建 PVC。创建 PVC 后，您可以将卷挂载到 Pod 中。

示例清单

以下示例显示了基本的 PVC 配置选项。

带 RWX 访问的 PVC

此示例显示了一个与名为 `basic-csi` 的 StorageClass 相关联的具有 RWX 访问权限的基本 PVC。

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

PVC 使用 iSCSI 示例

此示例显示了一个与名为 `protection-gold` 的 StorageClass 相关联的具有 RWO 访问权限的 iSCSI 基本 PVC。

```
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. 验证 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"](#)，了解存储类如何与 `PersistentVolumeClaim` 交互，以及用于控制 Trident 配置卷的参数详情。

部署应用程序

创建存储类和 PVC 后，您可以将 PV 安装到 Pod 上。本节列出了将 PV 附加到 Pod 的示例命令和配置。

部署示例应用程序

步骤

1. 在 Pod 中挂载卷。

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

以下示例显示了将 PVC 连接到 pod 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 `kubectl get pod --watch` 监控进度。

2. 验证卷是否已装入 `/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 Pod 了。Pod 应用程序将不再存在，但卷将保持不变。

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

在 EKS 集群上配置 Trident EKS 附加组件

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 加载项包含最新的安全补丁、错误修复，并经过 AWS 验证可与 Amazon EKS 配合使用。EKS 加载项使您能够始终如一地确保 Amazon EKS 集群的安全和稳定，并减少安装、配置和更新加载项所需的工作量。

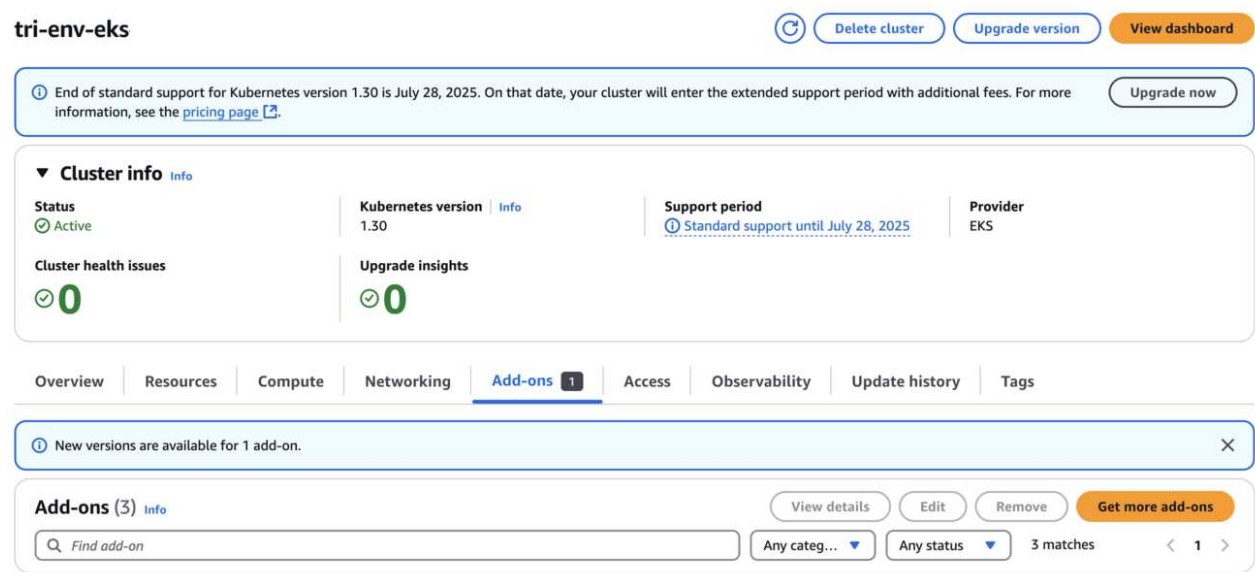
前提条件

在为 AWS EKS 配置 Trident 加载项之前，请确保具备以下条件：

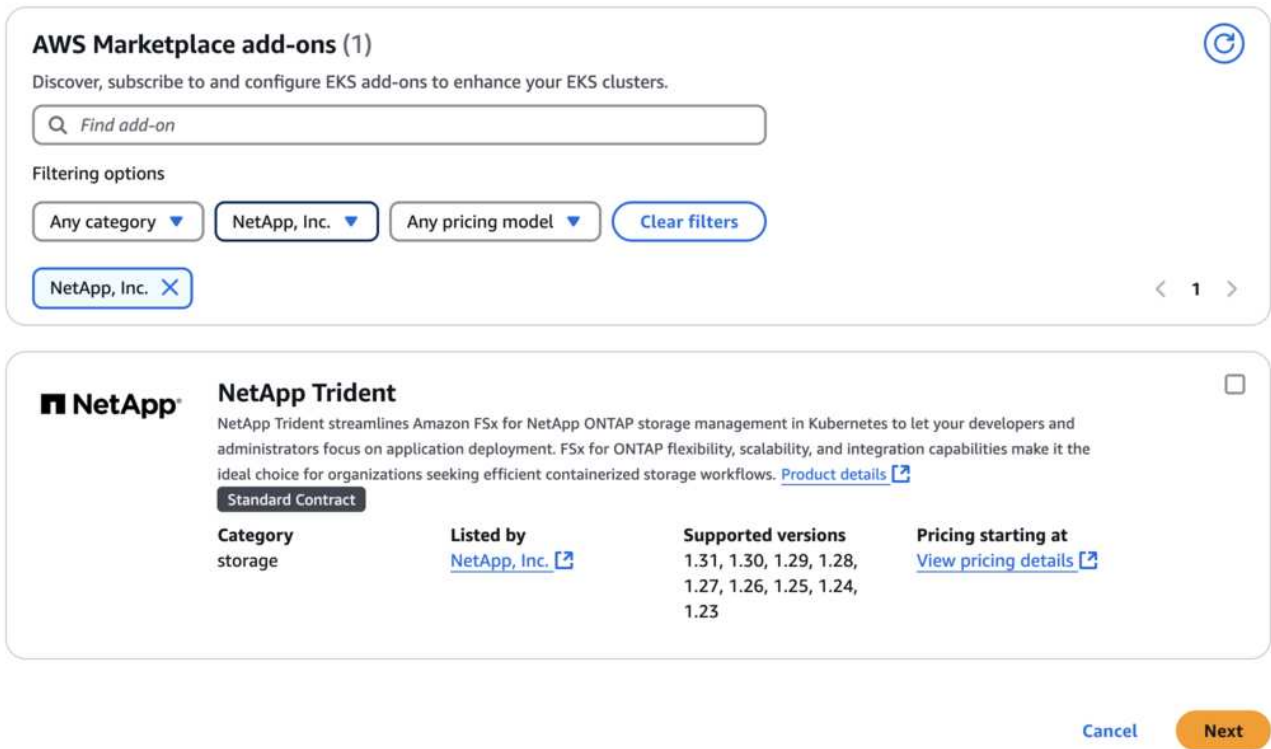
- 具有使用加载项权限的 Amazon EKS 群集帐户。请参阅 ["Amazon EKS 附加组件"](#)。
- AWS 对 AWS marketplace 的权限：
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- AMI 类型：Amazon Linux 2 (AL2_x86_64) 或 Amazon Linux 2 Arm(AL2_ARM_64)
- 节点类型：AMD 或 ARM
- 现有 Amazon FSx for NetApp ONTAP 文件系统

步骤

- 1. 请确保创建 IAM 角色和 AWS 密钥，以使 EKS Pod 能够访问 AWS 资源。有关说明，请参见 "创建 IAM 角色和 AWS Secret"。
- 2. 在 EKS Kubernetes 集群上，导航到 **Add-ons** 选项卡。



- 3. 转到 *AWS Marketplace 加载项* 并选择 *storage* 类别。



- 4. 找到 **NetApp Trident** 并选中 Trident 附加组件的复选框，然后单击 **Next**。
- 5. 选择所需的加载项版本。

Configure selected add-ons settings

Configure the add-ons for your cluster by selecting settings.

NetApp TridentRemove add-on

Listed by

Category

Status

NetApp

storage

Ready to install

You're subscribed to this software

View subscription

You can view the terms and pricing details for this product or choose another offer if one is available.

Version

Select the version for this add-on.

v25.6.0-eksbuild.1

Optional configuration settings

Cancel

Previous

Next

6. 配置所需的附加组件设置。

Review and add

Step 1: Select add-ons

[Edit](#)

Selected add-ons (1)

Find add-on

< 1 >

Add-on name	Type	Status
netapp_trident-operator	storage	Ready to install

Step 2: Configure selected add-ons settings

[Edit](#)

Selected add-ons version (1)

< 1 >

Add-on name	Version	IAM role for service account (IRSA)
netapp_trident-operator	v24.10.0-eksbuild.1	Not set

EKS Pod Identity (0)

< 1 >

Add-on name	IAM role	Service account
No Pod Identity associations None of the selected add-on(s) have Pod Identity associations.		

Cancel

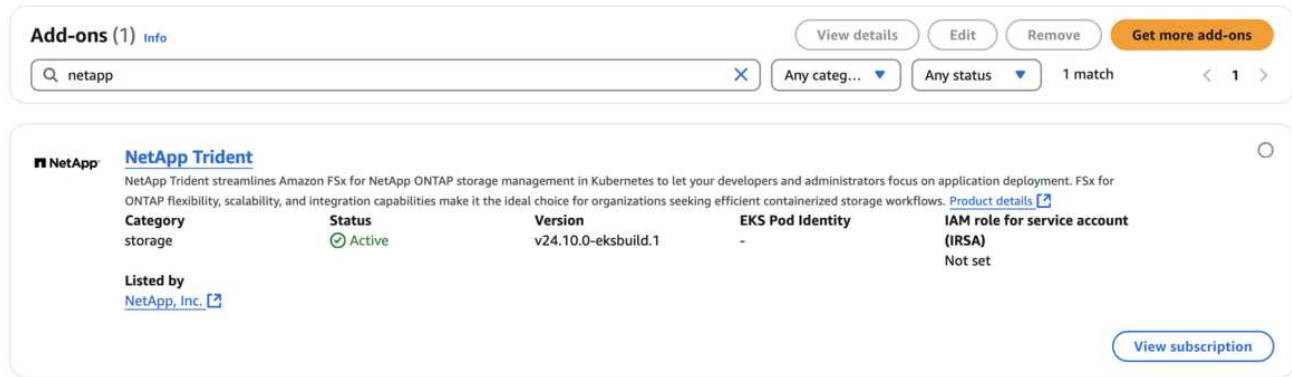
Previous

Create

7. 如果使用 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. 验证加载项的状态是否为 *Active*。



10. 运行以下命令来验证集群上是否正确安装了 Trident：

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

11. 继续设置并配置存储后端。有关信息，请参见 ["配置存储后端"](#)。

使用 CLI 安装/卸载 Trident EKS 附加组件

使用 CLI 安装 **NetApp Trident EKS** 附加组件：

以下示例命令安装 Trident EKS 加载项：

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

以下示例命令安装 Trident EKS 插件版本 25.6.1：

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.1-eksbuild.1 (带有专用版本)
```

以下示例命令安装 Trident EKS 插件版本 25.6.2：

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.2-eksbuild.1 (带有专用版本)
```

使用 CLI 卸载 **NetApp Trident EKS** 附加组件：

以下命令卸载 Trident EKS 加载项：

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

版权信息

版权所有 © 2026 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。