适用于 NetApp ONTAP 的 Amazon FSX

Trident

NetApp
February 02, 2026

目录

适用于 NetApp ONTAP 的 Amazon FSX

将Trident与Amazon FSx for NetApp ONTAP结合使用

"适用于 NetApp ONTAP 的 Amazon FSX" 是一种完全托管的AWS服务、可使客户启动和运行由NetApp ONTAP 存储操作系统提供支持的文件系统。借助适用于ONTAP 的FSx、您可以利用您熟悉的NetApp功能、性能和管理功能、同时利用在AWS上存储数据的简便性、灵活性、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进行集成、以确保在Amazon Elic Kubelnetes Service (EKS)中运行的Kubelnetes集群可以配置ONTAP支持的块和文件永久性卷。

文件系统是 Amazon FSX 中的主要资源，类似于内部部署的 ONTAP 集群。在每个 SVM 中，您可以创建一个或多个卷，这些卷是将文件和文件夹存储在文件系统中的数据容器。借助Amazon FSx for NetApp ONTAP、将在云中作为托管文件系统提供。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 Elic Kubelnetes Service (EKS)中运行的Kubelnetes集群可以配置ONTAP支持的块和文件永久性卷。

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 集群或自管理 Kubernetes 集群。
- 可从集群的工作节点访问的现有Amazon FSx for NetApp ONTAP文件系统和Storage Virtual Machine (SVM)。
- 为准备工作的工作节点 "NFS或iSCSI"。

> 确保按照Amazon Linux和Ubuntu所需的节点准备步骤进行操作 "Amazon Machine 映像" （ AMIS ），具体取决于您的 EKS AMI 类型。

注意事项

- SMB卷：
  - SMB卷支持使用 `ontap-nas` 仅限驱动程序。
  - Trident EKS加载项不支持SMB卷。
  - Trident仅支持挂载到Windows节点上运行的Pod的SMB卷。有关详细信息、请参见 "准备配置SMB卷" 。
- 在Trident 24.02之前的版本中、Trident无法删除在已启用自动备份的Amazon FSx文件系统上创建的卷。要在Trident 24.02或更高版本中防止此问题，请在AWS FSx for ONTAP的后端配置文件中指定 `fsxFilesystemID`、AWS、 `apikey`AWS `apiRegion`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 "适用于ONTAP 的FSX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设置的文件，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提供两种身份验证模式。

- 基于凭据(建议)：将凭据安全地存储在AWS机密管理器中。您可以使用文件系统的用户、也可以使用 `fsxadmin` `vsadmin` 为SVM配置的用户。

> Trident应以SVM用户身份运行、或者以具有相同角色的其他名称的用户身份运行 `vsadmin` 。Amazon FSx for NetApp ONTAP的某个 `fsxadmin`用户只能有限地替代`ONTAP `admin`` 集群用户。强烈建议将与`Trident`结合使用 `vsadmin`。

- 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、请参阅适用于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"

测试过的**Amazon**计算机映像**(AMI)**

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计算机映像(AMI)。以下 AMI 已通过 NetApp Trident 25.02 测试。

AMI	NAS	NAS经济型	iSCSI	iSCSI经济型
AL2023_x86_64_STANDARD	是的。	是的。	是的。	是的。
AL2_x86_64	是的。	是的。	是 *	是 *
BOTTLEROCKET_x86_64	是**	是的。	不适用	不适用

AL2023_ARM_64_STANDARD	是的。	是的。	是的。	是的。
AL2_ARM_64	是的。	是的。	是 *	是 *
BOTTLEROCKET_ARM_64	是**	是的。	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。

如果此处未列出您所需的AMI、并不表示它不受支持、而只是表示它尚未经过测试。此列表可作为已知可运行的 AMI 的指南。

使用以下项执行的测试：

- EKS版本：1.32
- 安装方法：Helm 25.06 和 AWS 附加组件 25.06
- 对于NAS、我们同时测试了NFSv3和NFSv4.1。
- 对于SAN、测试的是仅iSCSI、而不是NVMe-oF。

执行的测试：

- 创建：存储类、PVC、POD
- 删除：POD、PVC (常规、qtree/LUN—经济型、NAS与AWS备份)

了解更多信息

- "Amazon FSX for NetApp ONTAP 文档"
- "有关适用于 NetApp ONTAP 的 Amazon FSX 的博客文章"

创建IAM角色和AWS机密

您可以通过作为AWS IAM角色进行身份验证(而不是提供显式AWS凭据)来配置Kubbernetes Pod以访问AWS资源。

要使用AWS IAM角色进行身份验证、您必须使用EKS部署Kubenetes集群。

创建AWS机密管理器密钥

由于Trident将对FSx Sx Sv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。传递这些凭据的安全方法是通过AWS机密管理器密钥。因此、如果您还没有、则需要创建一个AWS机密管理器密钥、其中包含vsadmin帐户的凭据。

以下示例将创建一个AWS机密管理器密钥来存储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来为Trident提供所需的权限。

以下示例将使用AWS命令行界面创建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

您可以使用 EKS Pod Identity 配置 Kubernetes 服务账户，使其代入 AWS Identity and Access Management (IAM) 角色，或使用 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 Identity 关联提供了管理应用程序凭证的能力，类似于 Amazon EC2 实例配置文件向 Amazon EC2 实例提供凭证的方式。

在您的 **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 身份：

```
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 Identity Agent" 。

创建 **trust-relationship.json**：

创建 trust-relationship.json 文件，使 EKS 服务主体能够承担 Pod Identity 的此角色。然后创建一个具有以下信任策略的角色：

```
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
```

信任关系**.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"
        }
      }
    }
  ]
}
```

更新文件中的以下值 `trust-relationship.json`：

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

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

将**IAM**角色附加到**IAM**策略：

创建角色后、使用以下命令将在上述步骤中创建的策略附加到此角色：

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

验证**OCD**提供程序是否关联：

验证OIDC提供程序是否已与集群关联。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与集群关联：

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

如果您使用 **eksctl**，请使用以下示例为 EKS 中的服务账户创建 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Kubelnetes中适用于NetApp ONTAP的Amazon FSx存储管理、使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：

- 掌舵
- 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" 。

通过舵安装**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. 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、应用程序版本和修订版号。

```
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
```

服务帐户协会 **(IRSA)**

1. 添加Trident Helm存储库：

```
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`命令查看安装详细信息、例如名称、命名空间、图表、状态、应用程序版本和修订版号。

```
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，请确保在客户端计算机上启用了 iSCSI。如果您使用的是 AL2023 工作节点操作系统，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**插件安装**Trident**

Trident EKS加载项包括最新的安全修补程序和错误修复、并已通过AWS验证、可与Amazon EKS配合使用。通过EKS加载项、您可以始终确保Amazon EKS集群安全稳定、并减少安装、配置和更新加载项所需的工作量。

前提条件

在配置适用于AWS EKS的Trident加载项之前、请确保满足以下条件：

- 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

启用适用于**AWS**的**Trident**加载项

管理控制台

1. 打开Amazon EKS控制台，网址为 https://console.aws.amazon.com/eks/home#/clusters。
2. 在左侧导航窗格中，选择*群集*。
3. 选择要为其配置NetApp Trident CSI加载项的集群的名称。
4. 选择*Add-ones*，然后选择*Get more add-ones*。
5. 请按照以下步骤选择附加组件：
   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 NetApp 的 Trident 框右上角的复选框。
   c. 选择 * 下一步 * 。
6. 在“*配置选定的附加项*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. 选择要使用的*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可用的配置值：
      - 选择要使用的*版本*。
      - 按照 附加组件配置模式，将 配置值 部分中的 **configurationValues** 参数设置为您在上一步中创建的角色 ARN（值应采用以下格式）：

```
{

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

}
```

+
如果您为冲突解决方法选择覆盖、则可以使用Amazon EKS附加设置覆盖现有附加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解决冲突。在选择此选项之前、请确保Amazon EKS附加组件未管理您需要自行管理的设置。

7. 选择“下一步”。
8. 在*Review and add*页上，选择*Cree*。

   加载项安装完成后、您将看到已安装的加载项。

AWS命令行界面

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"
  }
}
```

替换 `<role ARN>`为上一步中创建的角色的ARN。

2.安装 **Trident EKS** 插件。

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

eksc

以下示例命令将安装Trident EKS加载项：

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

更新**Trident EKS**加载项

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*群集*。
3. 选择要更新NetApp Trident CSI加载项的集群的名称。
4. 选择*Add-ones*选项卡。
5. 选择* Trident按NetApp显示*，然后选择*编辑*。
6. 在“*按NetApp配置Trident”页上，执行以下操作：
   a. 选择要使用的*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命令行界面

以下示例将更新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>'\"}"
```

eksc

- 检查FSxN Trident CSI加载项的当前版本。请替换 `my-cluster`为您的集群名称。

  ```
  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'"}
```

- 将此加载项更新到上一步输出中的update下返回的版本。

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

如果您删除了该 `--force` 选项、并且任何Amazon EKS附加设置与您的现有设置冲突、则更新Amazon EKS附加设置将失败；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附加组件不会管理您需要管理的设置、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 Kubenetes字段管理的详细信息，请参阅 "Kubbernetes现场管理"。

卸载/删除**Trident EKS**加载项

您可以通过两种方式删除Amazon EKS附加项：

- 保留集群上的附加软件–此选项将删除Amazon EKS对任何设置的管理。此外、它还会使Amazon EKS无法通知您更新、并在您启动更新后自动更新Amazon EKS附加项。但是、它会保留集群上的附加软件。此选项可使附加组件成为自管理安装、而不是Amazon EKS附加组件。通过此选项、此附加组件不会出现停机。保留命令中的 `--preserve` 选项以保留此附加项。
- 从集群中完全删除附加软件–NetApp建议您仅在集群中没有依赖于此附加软件的资源时、才从集群中删除此附加软件。从命令中删除 `--preserve` 此选项 `delete` 以删除此加载项。

如果此附加项具有关联的IAM帐户、则不会删除此IAM帐户。

管理控制台

1. 打开Amazon EKS控制台，网址为 https://console.aws.amazon.com/eks/home#/clusters。
2. 在左侧导航窗格中，选择*群集*。
3. 选择要删除的NetApp Trident CSI加载项的集群名称。
4. 选择*Add-ons*选项卡，然后选择Trident by NetApp。*
5. 选择 * 删除 * 。
6. 在*Remove NetApp_trident-operator con確 認*对话框中，执行以下操作：
   a. 如果您希望Amazon EKS停止管理此附加组件的设置、请选择*保留集群*。如果要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命令行界面

请使用集群的名称进行替换 `my-cluster` 、然后运行以下命令。

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

eksc

以下命令将卸载Trident EKS加载项：

```
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。该文件需要指定所需的存储类型(NAS或SAN)、文件系统和用于获取该文件的SVM以及如何向其进行身份验证。以下示例显示了如何定义基于NAS的存储以及如何使用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: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
```

- 验证是否已成功创建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
```

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：

- `ontap-san`：配置的每个PV都是其自身Amazon FSx for NetApp ONTAP卷中的一个LUN。建议用于块存储。
- `ontap-nas`：配置的每个PV都是一个完整的Amazon FSx for NetApp ONTAP卷。建议用于NFS和SMB。
- `ontap-san-economi`： 为 NetApp ONTAP 卷配置的每个 PV 都是一个 LUN ，每个 Amazon FSX 具有可配置的 LUN 数量。
- `ontap-nas-economy.`：配置的每个 PV 都是一个 qtree ，对于 NetApp ONTAP 卷，每个 Amazon FSx 的 qtree 数量是可配置的。
- `ontap-nas-flexgroup`： 配置的每个 PV 都是适用于 NetApp ONTAP FlexGroup 卷的完整 Amazon FSX 。

有关驱动程序详细信息、请参见 "NAS驱动程序" 和 "SAN驱动程序"。

创建配置文件后、运行此命令在EKS中创建该文件：

```
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

有关后端配置选项，请参见下表：

参数	**Description**	示例
ve版本		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
m年 月 日	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是使用IPv6标志安装的、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]。如果在字段下 aws`提供 `fsxFilesystemID、则无需提供、 managementLIF`因为Trident会从AWS检索SVM `managementLIF`信息。因此、您必须提供SVM下某个用户的凭据(例如vsadmin)、并且该用户必须具有此 `vsadmin 角色。	"10.0.0.1"、"[2001：1234：abc：：：fefe]"

参数	Description	示例
dataLIF	协议 LIF 的 IP 地址。* ONTAP NAS驱动程序*：NetApp建议指定dataLIF。如果未提供此参数、则Trident将从SVM提取数据LIF。您可以指定要用于NFS挂载操作的完全限定域名(FQDN)、以便创建轮叫DNS来在多个dataLIF之间进行负载平衡。可以在初始设置后更改。请参阅 。* ONTAP SAN驱动程序*：不为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 LI。如果明确定义了dataLIF、则会生成警告。如果Trident是使用IPv6标志安装的、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]。	
autosExportPolicy	启用自动创建和更新导出策略[布尔值]。使用 `autoExportPolicy`和 `autoExportCIDRs`选项、Trident可以自动管理导出策略。	false
autosExportCIDR	用于筛选KubeNet节点IP的CIDR列表(启用时)。 `autoExportPolicy`使用 `autoExportPolicy`和 `autoExportCIDRs`选项、Trident可以自动管理导出策略。	"["0.0.0.0/0、"：：：/0"]"
标签	要应用于卷的一组任意 JSON 格式的标签	""
客户端证书	客户端证书的 Base64 编码值。用于基于证书的身份验证	""
clientPrivateKey	客户端专用密钥的 Base64 编码值。用于基于证书的身份验证	""
trustedCACertifate	受信任 CA 证书的 Base64 编码值。可选。用于基于证书的身份验证。	""
用户名	用于连接到集群或SVM的用户名。用于基于凭据的身份验证。例如、vsadmin。	
密码	用于连接到集群或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SVM管理LIF则派生。
s存储前缀	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	Description	示例
`limitAggregateUsage`	*请勿指定Amazon FSx for NetApp ONTAP。*提供的和 `vsadmin`不包含使用Trident检索聚合使用情况并对其进行限制所需的 `fsxadmin`权限。	请勿使用。
`limitVolumeSize`	如果请求的卷大小超过此值、则配置失败。此外、还会限制它为qtrees和FlexVol volume管理的卷的大小上限、并且此选项允许自定义每个LUN `qtreesPerFlexvol`的qtrees的最大数量	"" （默认情况下不强制实施）
`lunsPerFlexvol`	每个FlexVol volume的最大LUN数必须在[50、200]范围内。仅SAN。	“100”
`debugTraceFlags`	故障排除时要使用的调试标志。例如、 {"api"：false、"METHO"：true} 请勿使用 `debugTraceFlags` 除非您正在进行故障排除并需要详细的日志转储。	空
`nfsMountOptions`	NFS挂载选项的逗号分隔列表。通常会在存储类中为Kubnetes-永久性卷指定挂载选项、但如果在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。如果在存储类或配置文件中未指定挂载选项、则Trident不会在关联的永久性卷上设置任何挂载选项。	""
`nasType`	配置NFS或SMB卷创建。选项包括 `nfs， smb`或为空。*必须设置为 `smb` 对于SMB卷。*如果设置为空、则默认为NFS卷。	`nfs`
`qtreesPerFlexvol`	每个FlexVol volume的最大qtrees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管理控制台或ONTAP命令行界面创建的SMB共享的名称、或者允许Trident创建SMB共享的名称。对于Amazon FSx for ONTAP后端、此参数是必需的。	`smb-share`

参数	Description	示例
useREST	用于使用 ONTAP REST API 的布尔参数。如果设置为 true，则Trident将使用ONTAP REST API与后端进行通信。此功能需要使用ONTAP 9.11.1及更高版本。此外、使用的ONTAP登录角色必须有权访问 ontap 应用程序。预定义的和角色可以满足这一 vsadmin 要求 cluster-admin 。	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机密密钥。	"" "" ""
credentials	指定要存储在AWS密码管理器中的FSx SVM凭据。- name：密钥的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机密管理器密钥" 。	

用于配置卷的后端配置选项

您可以在中使用这些选项控制默认配置 defaults 配置部分。有关示例，请参见以下配置示例。

参数	Description	Default
spaceAllocation	LUN 的空间分配	true
s页面预留	空间预留模式；"无"(精简)或"卷"(厚)	无
sSnapshot 策略	要使用的 Snapshot 策略	无
qosPolicy	要为创建的卷分配的 QoS 策略组。选择每个存储池或后端的qosPolicy或adaptiveQosPolicy之一。将QoS策略组与Trident结合使用需要使用ONTAP 9™8或更高版本。您应使用非共享QoS策略组、并确保此策略组分别应用于每个成分卷。共享QoS策略组会对所有工作负载的总吞吐量实施上限。	""
adaptiveQosPolicy	要为创建的卷分配的自适应 QoS 策略组。选择每个存储池或后端的qosPolicy或adaptiveQosPolicy之一。不受 ontap-nas-economy.	""

参数	Description	Default
sSnapshot 预留	为快照"0"预留的卷百分比	如果 snapshotPolicy`为 `none，else“”
splitOnClone	创建克隆时，从其父级拆分该克隆	false
加密	在新卷上启用NetApp卷加密(NVE)；默认为 false。要使用此选项，必须在集群上获得 NVE 的许可并启用 NVE 。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。请参见 “使用Linux统一密钥设置(LUKS)”。仅限SAN。	""
分层策略	要使用的层策略 none	
unixPermissions	新卷的模式。对于**SMB**卷保留为空。	""
securityStyle	新卷的安全模式。NFS支持 mixed 和 unix 安全模式。SMB支持 mixed 和 ntfs 安全模式。	NFS默认值为 unix。SMB默认值为 ntfs。

提供 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： “准备配置SMB卷”。

配置存储类和PVC

配置Kubnetes 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来请求PV访问权限的永久性卷请求(PVC)。然后、您可以将PV挂载到POD。

创建存储类。

配置Kubnetes StorageClass对象

这 “Kubnetes StorageClass对象”对象将Trident标识为用于该类的供应器，并指示Trident如何供应卷。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 Bottler套 件上配置NFS3卷、请将所需添加 `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
```

有关存储类如何与和参数交互以控制Trident如何配置卷的详细信息 `PersistentVolumeClaim`、请参见"Kubernetes 和 Trident 对象"。

创建存储类。

步骤

1. 这是一个Kubbernetes对象、因此请使用 `kubectl` 以在Kubernetes中创建。

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

2. 现在，Kubernetes和Trident中都应显示一个*BASIC－Csi*存储类，并且Trident应已发现后端的池。

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

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

创建PVC

A "*PersigentVolumeClaim*" (PVC)是指请求访问集群上的永久卷。

可以将PVC配置为请求特定大小的存储或访问模式。通过使用关联的StorageClass，集群管理员可以控制不限于持续卷大小和访问模式(例如性能或服务级别)。

创建PVC后、您可以将卷挂载到Pod中。

示例清单

PersentVolumeClaim示例清单

这些示例显示了基本的PVC配置选项。

PVC、可接入**rwx**

此示例显示了一个具有rwx访问权限的基本PVC，该PVC与名为的StorageClass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**

此示例展示了具有 RWO 访问权限的 iSCSI 基本 PVC，它与名为 `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. 验证PVC状态。

```
kubectl get pvc
```

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

有关存储类如何与和参数交互以控制Trident如何配置卷的详细信息 `PersistentVolumeClaim`、请参见"Kubernetes 和 Trident 对象"。

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	Type	值	优惠	请求	支持
介质[1]	string	HDD ，混合，SSD	Pool 包含此类型的介质；混合表示两者	指定的介质类型	ontap-nas ，ontap-nas-economy. ontap-nas-flexgroup ，ontap-san ，solidfire-san
配置类型	string	精简，厚	Pool 支持此配置方法	指定的配置方法	Thick：All ONTAP ；Thin ：All ONTAP & solidfire-san
后端类型	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	池	true false	Pool 支持具有快照的卷	启用了快照的卷	ontap-nas 、ontap-san 、solidfire-san
克隆	池	true false	Pool 支持克隆卷	启用了克隆的卷	ontap-nas 、ontap-san 、solidfire-san
加密	池	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

属性	Type	值	优惠	请求	支持
IOPS	内部	正整数	Pool 能够保证此范围内的 IOPS	卷保证这些 IOPS	solidfire-san

[1]： ONTAP Select 系统不支持

部署示例应用程序

创建存储类和PVC后、您可以将PV挂载到Pod。本节列出了将PV连接到POD的示例命令和配置。

步骤

1. 将卷挂载到Pod中。

```
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
```

您可以使用监控进度 `kubectl get pod --watch`。

2. 验证卷是否已挂载到上 `/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。Pod应用程序将不再存在、但卷将保留。

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

在EKS集群上配置Trident EKS加载项

NetApp Trident简化了Kubelnetes中适用于NetApp ONTAP的Amazon FSx存储管理、使开发人员和管理员能够专注于应用程序部署。NetApp Trident EKS加载项包括最新的安全修补程序和错误修复、并已通过AWS验证、可与Amazon EKS配合使用。通过EKS加载项、您可以始终确保Amazon EKS集群安全稳定、并减少安装、配置和更新加载项所需的工作量。

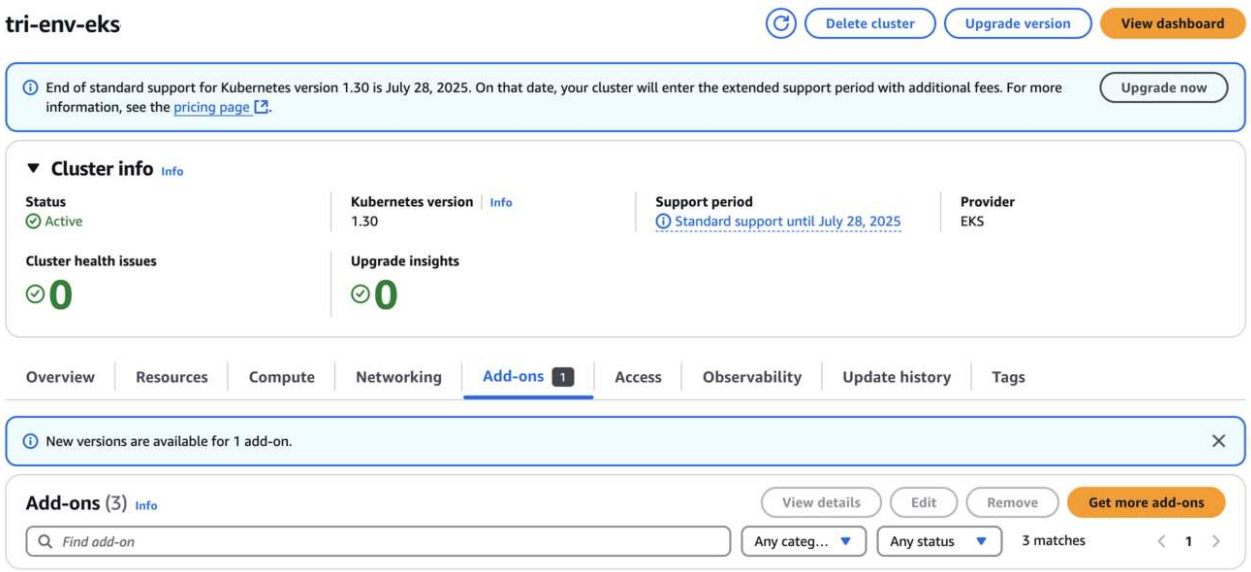
前提条件

在配置适用于AWS EKS的Trident加载项之前、请确保满足以下条件：

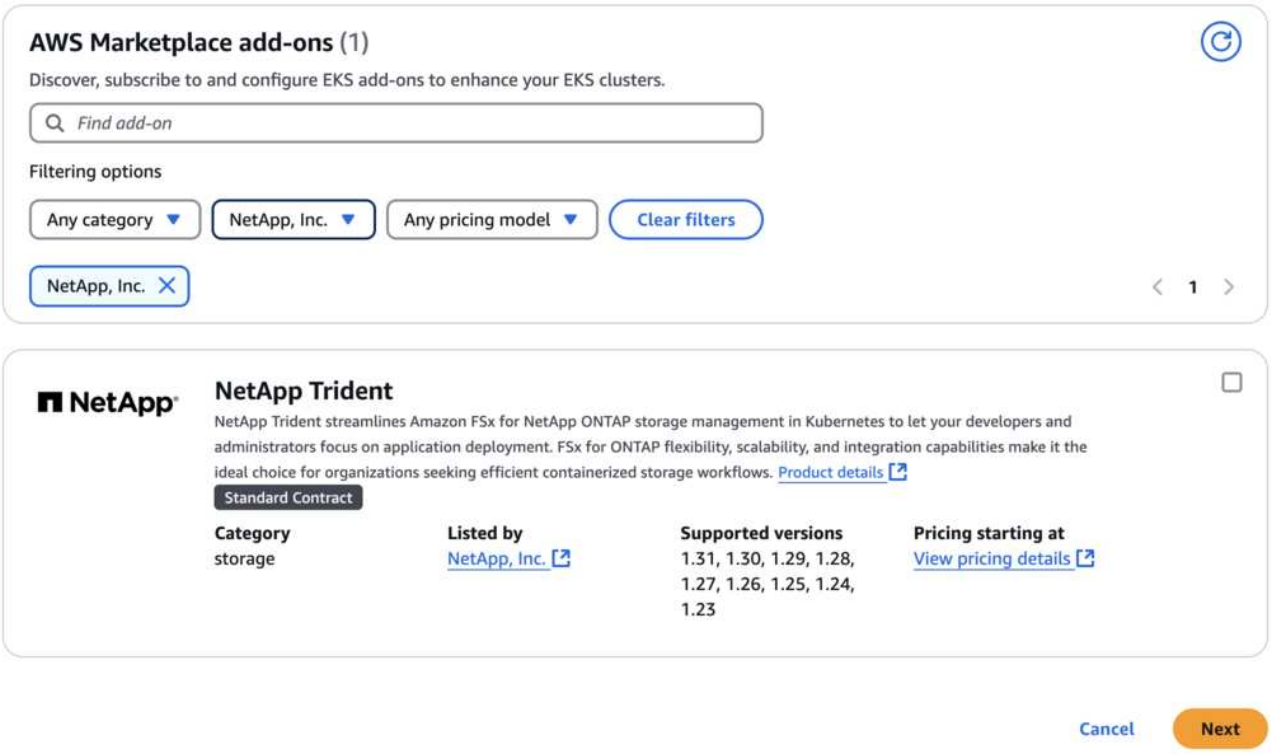
- 具有使用加载项的权限的Amazon EKS集群帐户。请参阅 "Amazon EKS附加项"。
- AWS对AWS Marketplace的权限：
  ```
  "aws-marketplace:ViewSubscriptions",
  "aws-marketplace:Subscribe",
  "aws-marketplace:Unsubscribe
  ```
- AMI类型：Amazon Linux 2 (AL2_x86_64)或Amazon Linux 2 ARM (AL2_ARM_64)
- 节点类型：AMD或ARM
- 现有Amazon FSx for NetApp ONTAP文件系统

步骤

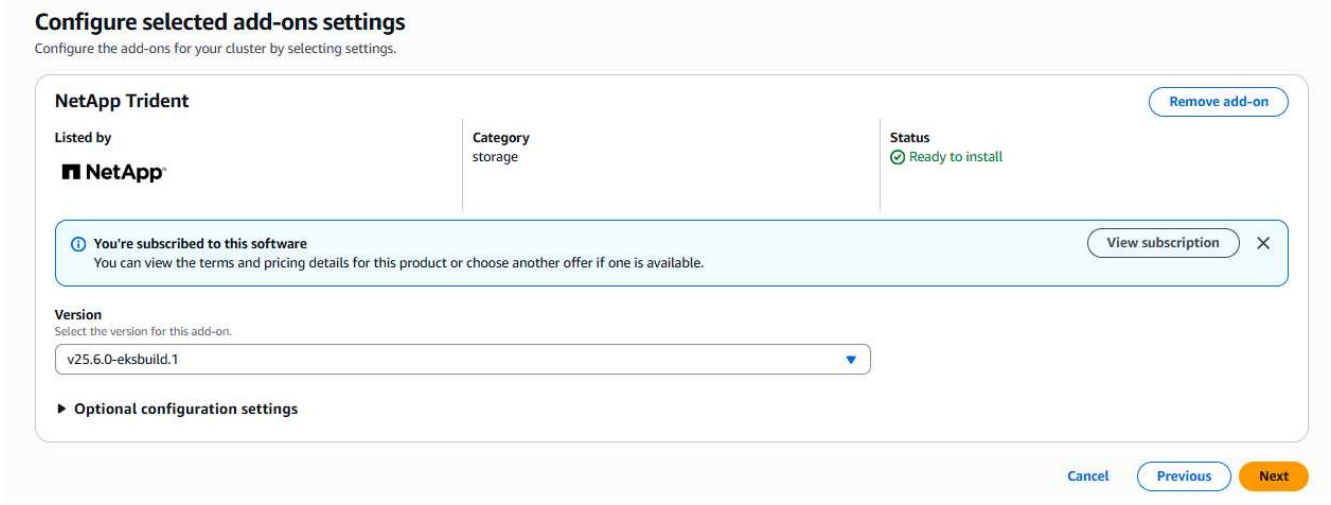
1. 请务必创建IAM角色和AWS密钥、以使EKS Pod能够访问AWS资源。有关说明，请参阅"创建IAM角色和AWS机密"。
2. 在EKS Kubernetes集群上、导航到*加载项*选项卡。



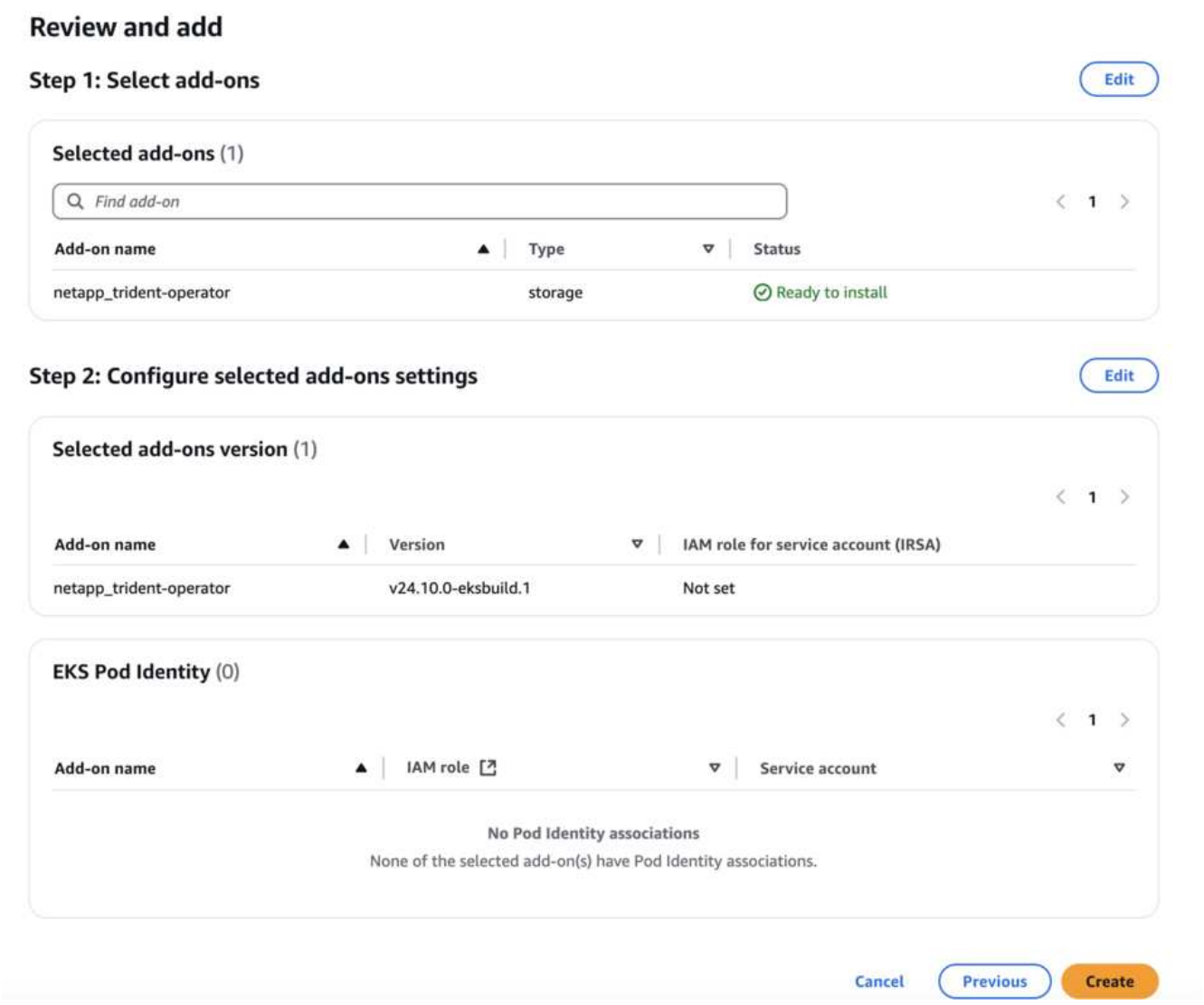
3. 转到*AWS Marketplace附加项*并选择_storage_类别。



4. 找到*Next* NetApp Trident并选中Trident插件的复选框，然后单击*Next*。
5. 选择所需的附加软件版本。

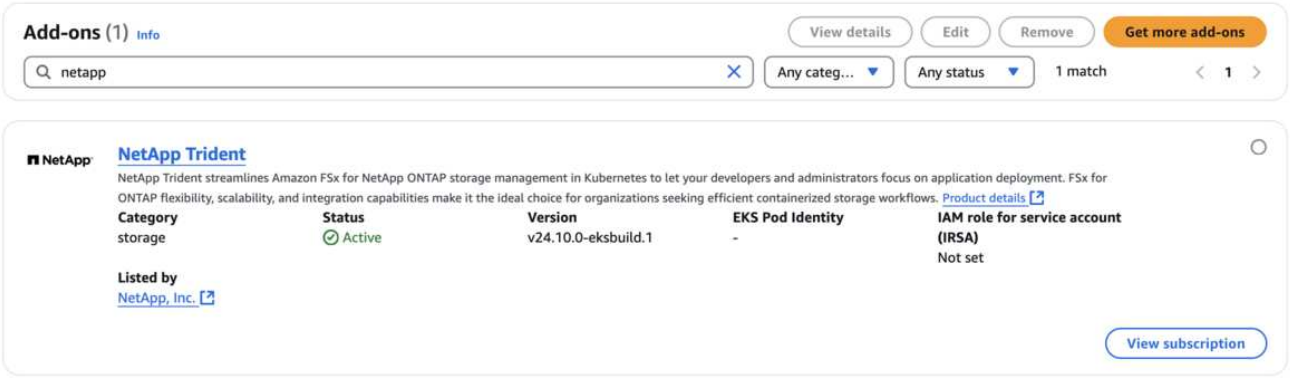


6. 配置所需的附加组件设置。



7. 如果您使用 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. 验证此加载项的状态是否为_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. 继续设置并配置存储后端。有关信息，请参见 "配置存储后端"。

使用命令行界面安装/卸载Trident EKS加载项

使用命令行界面安装**NetApp Trident EKS**加载项：

以下示例命令安装Trident EKS 附加组件：

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

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）
```

使用命令行界面卸载**NetApp Trident EKS**加载项：

以下命令将卸载Trident EKS加载项：

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

版权信息

商标信息