



Google Cloud NetApp Volumes

Trident

NetApp
October 29, 2026

This PDF was generated from <https://docs.netapp.com/zh-cn/trident-2606/trident-use/gcnv.html> on October 29, 2026. Always check docs.netapp.com for the latest.

目录

Google Cloud NetApp Volumes	1
配置 Google Cloud NetApp Volumes	1
概述	1
准备配置	1
配置 NAS 存储	2
为 SAN 工作负载配置 Google Cloud NetApp Volumes	5
概述	5
Flex Unified 存储池	5
配置 Trident SAN 后端	5
创建 StorageClass	6
配置块卷	6
块卷行为	8
访问模式	8
volumeMode 行为	8
支持的操作	9
额外的 GiB 过度配置行为	9
Pod 示例	9
连接和挂载行为	10
准备配置 Google Cloud NetApp Volumes 后端	11
NFS 或 SMB 卷的先决条件	11
Google Cloud NetApp Volumes 后端配置选项和示例	11
后端配置选项	11
卷配置选项	12
示例配置	13
下一步是什么？	18
存储类定义	19
为 Google Cloud NetApp Volumes 配置自动分层	21
概述	21
概念	22
配置模型	22
Trident 26.06 中支持的功能	22
Trident 26.06 中不支持的功能	23
后端配置参数	23
使用 PersistentVolumeClaim 注释的卷级覆盖	23
行为和限制	24
配置 Google Cloud NetApp Volumes ONTAP 模式存储池	24
工作原理	24
支持和不支持的范围	25
前提条件	25

Google Cloud NetApp Volumes

配置 Google Cloud NetApp Volumes

您可以将 Google Cloud NetApp Volumes 配置为 Trident 的后端，以便为 Kubernetes 工作负载配置存储。

概述

Trident 支持用于 NAS (NFS 和 SMB) 和块 (iSCSI) 工作负载的 Google Cloud NetApp Volumes。

- NAS 工作负载使用 `google-cloud-netapp-volumes` 后端
- 块 (iSCSI) 工作负载使用 `google-cloud-netapp-volumes-san` 后端

NAS 卷提供基于文件的存储，并使用 NFS 或 SMB 协议进行访问。这些卷支持跨多个 pod 或节点的共享访问。

块卷提供原始块存储，并作为连接到 Kubernetes 节点的 iSCSI 设备进行访问。当应用程序需要块级访问时，将使用这些卷。

这适用于以下环境：

- Trident 26.02 及更高版本
- Google Kubernetes Engine (GKE) 或 Red Hat OpenShift
- Google Cloud NetApp Volumes 存储池

要配置块 (iSCSI) 存储，请参见 ["配置块存储 \(iSCSI\)"](#)。

准备配置

云身份使 Kubernetes 工作负载能够通过作为工作负载身份进行身份验证而不是使用静态凭据来访问 Google Cloud 资源。

要在 Google Cloud NetApp Volumes 中使用云标识，必须具有：

- 使用 Google Kubernetes Engine (GKE) 部署的 Kubernetes 集群
- 已在 GKE 群集上启用工作负载标识，已在节点池上启用元数据服务器
- 具有 Google Cloud NetApp Volumes Admin 角色(`roles/netapp.admin`) 的 Google Cloud 服务帐户或同等自定义角色
- 安装 Trident 时，云提供商设置为 `GCP` 并配置了云身份注释

Trident 操作员

要使用 Trident 操作员安装 Trident，请编辑 `tridentorchestrator_cr.yaml`：

```
apiVersion: trident.netapp.io/v1
```

```
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  namespace: trident
  cloudProvider: "GCP"
  cloudIdentity: "iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com"
```

Helm

使用 Helm 安装 Trident 时设置云提供商和云标识：

```
helm install trident trident-operator-100.6.0.tgz \
  --set cloudProvider=GCP \
  --set cloudIdentity="iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com"
```

tridentctl

通过指定云提供商和云身份来安装 Trident：

```
tridentctl install \
  --cloud-provider=GCP \
  --cloud-identity="iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com" \
  -n trident
```

配置 NAS 存储



对于 Google Cloud NetApp Volumes UNIFIED 存储池，Trident 在卷操作期间应用 UNIFIED 特定的命名和验证规则。

定位卷时，Trident 可以评估多个兼容的卷名变体（例如，连字符和下划线格式），以提高导入和发现的可靠性。

驱动程序详细信息

Trident 提供 google-cloud-netapp-volumes 驱动程序，用于从 Google Cloud NetApp Volumes 配置 NAS 存储。

驱动程序支持以下访问模式：

- ReadWriteOnce (RWO)
- ReadOnlyMany (ROX)

- ReadWriteMany (RWX)
- ReadWriteOncePod (RWOP)

驱动程序	协议	volumeMode	支持的访问模式	支持的文件系统
google-cloud-netapp-volumes	NFS SMB	Filesystem	RWO、ROX、RWX、RWOP	nfs, smb

配置 Trident NAS 后端

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
    - labels:
        cloud: gcp
        network: "<vpc-network>"

```

配置 NAS 卷

NAS 卷使用 `google-cloud-netapp-volumes` 后端进行调配，并支持 NFS 和 SMB 协议。

StorageClass 适用于 NFS 卷

要配置 NFS 卷，请将 `nasType` 设置为 `nfs`。

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "nfs"
allowVolumeExpansion: true

```

StorageClass 适用于 SMB 卷

要配置 SMB 卷，请将 `nasType` 设置为 `smb` 并提供凭据。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
allowVolumeExpansion: true
```

PersistentVolumeClaim 示例 (RWX)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwx
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs
```

PersistentVolumeClaim 示例 (RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs
```



NAS 卷使用 `volumeMode: Filesystem`。

为 SAN 工作负载配置 Google Cloud NetApp Volumes

您可以将 Trident 配置为使用 iSCSI 协议从 Google Cloud NetApp Volumes 配置块存储卷。SAN 卷使用 ``google-cloud-netapp-volumes-san`` 存储驱动程序从 Flex Unified 存储池进行配置。



此驱动程序专用于块工作负载，不支持 NAS 协议。



`google-cloud-netapp-volumes-san` 后端需要配置 iSCSI 块卷。`google-cloud-netapp-volumes` 后端仅支持 NAS 协议，不能用于 SAN 工作负载。

概述

Trident 支持 Google Cloud NetApp Volumes SAN (iSCSI) 工作负载，使用 ``google-cloud-netapp-volumes-san`` 驱动程序。

SAN 卷从 Flex Unified 存储池进行调配，并作为 iSCSI 块设备呈现给 Kubernetes 节点。

这适用于以下环境：

- Trident 26.02 及更高版本
- Google Kubernetes Engine (GKE) 或 Red Hat OpenShift
- Google Cloud NetApp Volumes Flex 统一存储池
- 基于 iSCSI 的工作负载

Flex Unified 存储池

Flex Unified 存储池使用 iSCSI 协议提供块存储，是 SAN 配置所必需的：

- 支持 Flex Unified REGIONAL 池。
- 从 Trident 26.02.1 开始支持 Flex Unified ZONAL 池。
- SAN 工作负载仅支持 **Flex** 服务级别。

配置 Trident SAN 后端

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-san
  namespace: trident
spec:
  version: 1
```

```
storageDriverName: google-cloud-netapp-volumes-san
projectNumber: "<project-number>"
location: "<region>"
sdkTimeout: "600"
storage:
- labels:
    cloud: gcp
    performance: flex
    network: "<vpc-network>"
    serviceLevel: Flex
```

创建 StorageClass

配置 SAN 后端后，创建一个引用 `google-cloud-netapp-volumes-san` 驱动程序的 StorageClass。

文件系统类型在 StorageClass 中定义，而不是在后端。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

支持的文件系统类型：

- ext4 (默认)
- ext3
- xfs



SAN 驱动程序仅支持 Flex 服务级别，不使用特定于 NAS 的后端参数，例如 `exportRule`、`unixPermissions`、`nasType`、`snapshotDir`、``nfsMountOptions`` 或与分层相关的设置。

配置块卷

ReadWriteOnce (RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwo
```

```
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadWriteOncePod (RWOP)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwop
spec:
  accessModes:
    - ReadWriteOncePod
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadOnlyMany (ROX)

ROX 的常见模式是克隆现有 ReadWriteOnce 卷并将克隆挂载为只读。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rox
spec:
  accessModes:
    - ReadOnlyMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
  dataSource:
    kind: PersistentVolumeClaim
    name: gcnv-san-rwo
```

ReadWriteMany (RWX) — 仅原始块

只有当 `volumeMode: Block` 时才支持 ReadWriteMany。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-raw-rwx
spec:
  accessModes:
    - ReadWriteMany
  volumeMode: Block
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

块卷行为

块卷被调配为 iSCSI LUN，并作为块设备呈现给 Kubernetes 节点。

块卷：

- 使用 iSCSI 协议
- 支持文件系统和原始块呈现
- 由 Trident 连接和管理
- 支持多种 Kubernetes 访问模式

访问模式

Trident 配置的块卷支持以下访问模式：

- ReadWriteOnce (RWO)
- ReadOnlyMany (ROX)
- ReadWriteOncePod (RWOP)
- ReadWriteMany (RWX)，仅在 `volumeMode: Block` 时受支持

volumeMode 行为

`volumeMode` 字段控制块卷的暴露方式：

- Filesystem Trident 格式化和装载卷。

- Block Trident 附加设备并将其公开为原始块设备。

支持的操作

使用 google-cloud-netapp-volumes-san 驱动程序配置的块卷支持：

- 创建
- 删除
- 克隆
- Snapshot
- 调整大小
- 导入

额外的 GiB 过度配置行为

Google Cloud NetApp Volumes 块卷包括内部元数据开销。与调配的容量相比，这种开销减少了内核可见的设备大小。

测试显示：

- 初始创建时约 300 KiB 的开销
- 调整大小后开销高达约 107 MiB

由于 Google Cloud NetApp Volumes 仅接受全 GiB 分配，Trident 确保可用设备大小始终满足或超过 PVC 请求：

- 将请求的大小舍入至下一个整 GiB
- 添加额外的 1 GiB 缓冲区

示例：

- PVC 请求：100 GiB
- Google Cloud NetApp Volumes 中的预配大小：101 GiB
- 应用程序可见的可用空间：至少 100 GiB

Pod 示例

文件系统挂载的块卷 (RWO)

```
apiVersion: v1
kind: Pod
metadata:
  name: app-rwo
spec:
  containers:
```

```

- name: app
  image: ubuntu:22.04
  command: ["sleep", "infinity"]
  volumeMounts:
  - name: data
    mountPath: /mnt/data
volumes:
- name: data
  persistentVolumeClaim:
    claimName: gcnv-san-rwo

```

原始块设备 (RWX)

```

apiVersion: v1
kind: Pod
metadata:
  name: app-raw-rwx
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeDevices:
    - name: data
      devicePath: /dev/xda
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-raw-rwx

```

连接和挂载行为

对于从 Google Cloud NetApp Volumes 配置的 SAN 卷：

- Trident 在 Flex Unified 存储池中创建逻辑单元号 (LUN)。
- 在发布期间，Trident 将 LUN 映射到每个节点的主机组。
- 在节点暂存期间，Trident：
 - 登录到 iSCSI 目标
 - 发现 LUN
 - 配置多路径
- 如果 volumeMode: Filesystem，Trident 会根据需要格式化设备并挂载它。
- 如果 volumeMode: Block，Trident 会连接设备并将其直接暴露给 pod，无需格式化或挂载。



SAN 块卷不提供分布式锁定或写入协调。当一个块卷被多个节点访问时（ReadWriteMany 与 volumeMode: Block），应用程序或文件系统必须管理并发。

准备配置 Google Cloud NetApp Volumes 后端

在配置 Google Cloud NetApp Volumes 后端之前，需要确保满足以下要求。

NFS 或 SMB 卷的先决条件

如果是首次使用 Google Cloud NetApp Volumes 或在新位置使用，则需要一些初始配置才能设置 Google Cloud NetApp Volumes 并创建 NFS 或 SMB 卷。请参阅 ["开始之前"](#)。

在配置 Google Cloud NetApp Volumes 后端之前，请确保您具有以下内容：

- 使用 Google Cloud NetApp Volumes 服务配置的 Google Cloud 帐户。请参阅 ["Google Cloud NetApp Volumes"](#)。
- 您的 Google Cloud 帐户的项目编号。请参阅 ["识别项目"](#)。
- 具有 NetApp Volumes Admin (roles/netapp.admin) 角色的 Google Cloud 服务帐户。请参阅 ["身份和访问管理角色和权限"](#)。
- 您的 GCNV 帐户的 API 密钥文件。请参阅 ["创建服务帐户密钥"](#)
- 存储池。请参阅 ["存储池概述"](#)。

有关如何设置 Google Cloud NetApp Volumes 访问权限的详细信息，请参阅 ["设置对 Google Cloud NetApp Volumes 的访问权限"](#)。

Google Cloud NetApp Volumes 后端配置选项和示例

了解 Google Cloud NetApp Volumes 的后端配置选项，并查看配置示例。

后端配置选项

每个后端在一个 Google Cloud 区域中配置卷。要在其他区域中创建卷，您可以定义其他后端。

参数	说明	默认
version		始终为 1
storageDriverName	存储驱动程序的名称	storageDriverName 的值必须指定为"google-cloud-netapp-volumes"。
backendName	(可选) 存储后端的自定义名称	驱动程序名称 + "_" + API 密钥的一部分
storagePools	用于指定卷创建的存储池的可选参数。	
projectNumber	Google Cloud 帐户项目编号。该值位于 Google Cloud 门户主页上。	

参数	说明	默认
location	Trident 创建 GCNV 卷的 Google Cloud 位置。在创建跨区域 Kubernetes 集群时，在 `location` 中创建的卷可用于在多个 Google Cloud 区域的节点上计划的工作负载。跨区域流量会产生额外费用。	
apiKey	具有 netapp.admin`角色的 Google Cloud 服务帐户的 API 密钥。它包括 Google Cloud 服务帐户私钥文件的 JSON 格式内容（逐字复制到后端配置文件中）。`apiKey` 必须包括以下键的键值对： `type`、`project_id`、`client_email`、 `client_id`、`auth_uri`、`token_uri`、 `auth_provider_x509_cert_url` 和 `client_x509_cert_url`。	
nfsMountOptions	NFS 挂载选项的精细控制。	"nfsvers=3"
limitVolumeSize	如果请求的卷大小高于此值，则设置失败。	"（默认情况下不强制执行）"
serviceLevel	存储池及其卷的服务级别。值为 flex、standard、premium`或`extreme。	
labels	要应用于卷的任意 JSON 格式标签集	""
network	用于 GCNV 卷的 Google Cloud 网络。	
debugTraceFlags	故障排除时使用的调试标志。示例， { "api": false, "method": true }。除非正在进行故障排除并需要详细的日志转储，否则不要使用此选项。	空
nasType	配置 NFS 或 SMB 卷创建。选项为 nfs、`smb` 或 null。设置为 null 默认为 NFS 卷。	nfs
supportedTopologies	表示此后端支持的区域和可用区列表。有关详细信息，请参阅 "使用 CSI 拓扑" 。例如： supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a	

卷配置选项

您可以在配置文件的 defaults 部分中控制默认卷配置。

参数	说明	默认
exportRule	新卷的导出规则。必须是任何 IPv4 地址组合的逗号分隔列表。	"0.0.0.0/0"
snapshotDir	访问 .snapshot 目录	true, false (默认行为可能会有所不同。显式设置) NFSv3 的 "false"
snapshotReserve	为快照预留的卷百分比	" (接受默认值 0)

参数	说明	默认
unixPermissions	新卷的 unix 权限（4 位八进制数字）。	""

示例配置

以下示例显示了将大多数参数保留为默认值的基本配置。这是定义后端的最简单方法。

▼ 最小配置

这是绝对最小后端配置。通过此配置，Trident 发现配置位置中委托给 Google Cloud NetApp Volumes 的所有存储池，并将新卷随机放置在其中一个池中。由于 `nasType` 被省略，`nfs` 默认值适用，后端将为 NFS 卷进行配置。

当您刚刚开始使用 Google Cloud NetApp Volumes 并尝试使用时，此配置非常理想，但在实践中，您可能需要为配置的卷提供额外的范围。



请将 <id_value> 和 <key_value> 替换为您的服务帐户凭据。

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"

```

```

    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
    credentials:
      name: backend-tbc-gcnv-secret

```

▼ SMB 卷的配置

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv1
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123456789"
  location: asia-east1
  serviceLevel: flex
  nasType: smb
  apiKey:
    type: service_account
    project_id: cloud-native-data
    client_email: trident-sample@cloud-native-
data.iam.gserviceaccount.com
    client_id: "123456789737813416734"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/trident-
sample%40cloud-native-data.iam.gserviceaccount.com
    credentials:
      name: backend-tbc-gcnv-secret

```

▼ 使用 StoragePools 过滤器的配置

```

---
apiVersion: v1
kind: Secret

```

```

metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
---

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  storagePools:
    - premium-pool1-europe-west6
    - premium-pool2-europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret

```

▼ 虚拟池配置

此后端配置在单个文件中定义多个虚拟池。虚拟池在 `storage` 部分中定义。当您有多个支持不同服务级别的存储池，并且希望在 Kubernetes 中创建表示这些级别的存储类时，它们非常有用。虚拟池标签用于区分池。例如，在下面的示例中，`performance` 标签和 `serviceLevel` 类型用于区分虚拟池。

您还可以设置一些适用于所有虚拟池的默认值，并覆盖各个虚拟池的默认值。在以下示例中，

snapshotReserve 和 exportRule 用作所有虚拟池的默认值。

有关详细信息，请参阅 ["虚拟池"](#)。

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
  defaults:
    snapshotReserve: "10"
    exportRule: 10.0.0.0/24
  storage:
    - labels:
```

```

    performance: extreme
    serviceLevel: extreme
  defaults:
    snapshotReserve: "5"
    exportRule: 0.0.0.0/0
- labels:
    performance: premium
    serviceLevel: premium
- labels:
    performance: standard
    serviceLevel: standard

```

▼ 适用于 GKE 的云标识

当您使用 GKE 云身份安装 Trident 时，后端将作为工作负载身份进行身份验证，并省略 `apiKey` 块。云身份的配置方式对于两种 Google Cloud NetApp Volumes 驱动程序相同。唯一的区别是 `storageDriverName` 值：对于 NAS（NFS 或 SMB）卷使用 `google-cloud-netapp-volumes`，对于 SAN（iSCSI 块）卷使用 `google-cloud-netapp-volumes-san`。



对 `google-cloud-netapp-volumes-san` 驱动程序的云身份支持需要 Trident 26.06 或更高版本。SAN 卷是从 Flex Unified 存储池调配的，因此请将 `serviceLevel` 设置为 `flex`。

NAS 驱动程序 (<code>google-cloud-netapp-volumes</code>)

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: Premium
  storagePool: pool-premium1

```

SAN 驱动程序 (<code>google-cloud-netapp-volumes-san</code>)

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv-san
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san

```

```
projectNumber: '012345678901'
network: gcnv-network
location: us-west2
serviceLevel: flex
storagePool: pool-flex1
```

▼ 支持的拓扑配置

Trident 便于根据区域和可用区为工作负载调配卷。此后端配置中的 `supportedTopologies` 块用于为每个后端提供区域和区域的列表。此处指定的区域和区域值必须与每个 Kubernetes 集群节点上标签的区域和区域值匹配。这些区域和区域表示可以在存储类中提供的允许值列表。对于包含后端提供的区域和区域子集的存储类，Trident 会在上述区域和区域中创建卷。有关详细信息，请参阅 ["使用 CSI 拓扑"](#)。

```
---
version: 1
storageDriverName: google-cloud-netapp-volumes
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: asia-east1
serviceLevel: flex
supportedTopologies:
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-a
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-b
```

下一步是什么？

创建后端配置文件后，运行以下命令：

```
kubectl create -f <backend-file>
```

要验证是否已成功创建后端，请运行以下命令：

```
kubectl get tridentbackendconfig
```

NAME	BACKEND NAME	BACKEND UUID
PHASE STATUS		
backend-tbc-gcnv	backend-tbc-gcnv	b2fd1ff9-b234-477e-88fd-713913294f65
Bound Success		

如果后端创建失败，则后端配置有问题。您可以使用 `kubectl get tridentbackendconfig <backend-`

name> 命令描述后端或通过运行以下命令查看日志以确定原因：

```
tridentctl logs
```

在识别并更正配置文件的问题后，您可以删除后端并再次运行 create 命令。

存储类定义

以下是参考上述后端的基本 StorageClass 定义。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

使用 `parameter.selector` 字段的定义示例：

使用 `parameter.selector`，您可以为每个 `StorageClass` 指定用于托管卷的“[虚拟池](#)”。卷将具有所选池中定义的方面。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
```

```
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
  backendType: google-cloud-netapp-volumes
```

有关存储类的更多详细信息，请参阅 ["创建存储类"](#)。

SMB 卷的定义示例

使用 `nasType`、`node-stage-secret-name` 和 `node-stage-secret-namespace`，可以指定 SMB 卷并提供所需的 Active Directory 凭据。任何具有任何权限/无权限的 Active Directory 用户/密码都可以用于节点阶段密码。

▼ 默认命名空间的基本配置

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

▼ 每个命名空间使用不同的机密

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "${pvc.namespace}"
```

▼ 每个卷使用不同的密钥

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
```

```

metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}

```



nasType: smb 支持 SMB 卷的池的筛选器。nasType: nfs 或 nasType: null NFS 池的筛选器。

PVC 定义示例

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc

```

要验证 PVC 是否已绑定，请运行以下命令：

```
kubectl get pvc gcnv-nfs-pvc
```

NAME	STATUS	VOLUME	CAPACITY
gcnv-nfs-pvc	Bound	pvc-b00f2414-e229-40e6-9b16-ee03eb79a213	100Gi
		gcnv-nfs-sc 1m	

为 Google Cloud NetApp Volumes 配置自动分层

自动分层通过 Trident 后端参数和 PersistentVolumeClaim 注释在卷配置期间进行配置。您可以使用 Trident 为 Google Cloud NetApp Volumes 配置自动分层。

概述

自动分层允许 Trident 配置卷，自动将非活动数据从性能层移动到容量层。这降低了存储成本，同时保留了频繁

访问数据的性能。

Trident 仅在卷创建时应用自动分层设置。Trident 26.06 不支持预配后更改。

概念

自动分层

自动分层根据访问模式将不经常访问的数据从性能层移动到容量层。数据移动是异步发生的，并不是立即发生的。

分层策略

分层策略确定是否为卷启用自动分层。

支持以下策略：* `auto`：根据访问模式启用自动分层 * `none`：禁用自动分层

冷却天数

冷却天数指定数据块在符合分层条件之前必须保持非活动状态的最短天数。仅当分层策略设置为 `'auto'` 时，才适用冷却天数。

配置模型

配置范围

可以在多个范围内配置自动分层：

- 存储池范围 适用于从池中配置的所有卷。
- 卷范围 通过 `PersistentVolumeClaim` 注释应用于单个卷。

Trident 根据每个设置的定义位置确定有效配置。

配置优先级

当在多个作用域中定义了相同的设置时，Trident 应用以下优先级顺序：

1. `PersistentVolumeClaim` 注释
2. Trident 后端配置
3. 存储池默认值

在较高优先级定义的设置将覆盖较低级别的值。

Trident 26.06 中支持的功能

Trident 26.06 支持 Google Cloud NetApp Volumes 的以下自动分层功能：

- 在卷配置期间启用或禁用自动分层
- 在 Trident 后端配置中定义分层策略

- 使用 PVC 注释覆盖分层策略和每个卷的冷却天数
- 为启用自动分层的卷配置冷却天数

Trident 26.06 中不支持的功能

不支持以下操作：

- 创建卷后修改自动分层设置
- 使用 Kubernetes 更新更改现有卷的分层策略
- 在 Trident 管理的配置工作流之外应用自动分层设置

后端配置参数

以下参数在 Trident 后端配置中定义时控制自动分层行为：

参数	必填项	说明
tieringPolicy	否	卷的分层策略 ((auto` 或 `none)
tieringMinimumCoolingDays	否	数据分层前的非活动天数（范围：2–183，默认值：31）

使用 PersistentVolumeClaim 注释的卷级覆盖

支持的注释

PersistentVolumeClaim 批注允许按卷覆盖自动分层设置。

标注	说明
trident.netapp.io/tieringPolicy	覆盖卷的分层策略
trident.netapp.io/tieringMinimumCoolingDays	覆盖该卷的冷却天数

示例：PersistentVolumeClaim 使用自动分层覆盖

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: auto-tiering-pvc
  annotations:
    trident.netapp.io/tieringPolicy: auto
    trident.netapp.io/tieringMinimumCoolingDays: "45"
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: google-cloud-netapp-volumes-auto-tiering
```

```
resources:
  requests:
    storage: 500Gi
```

行为和限制

配置行为

- 自动分层设置仅在创建卷时进行评估和应用。
- Trident 在配置后不会协调分层配置。
- 当分层策略设置为 `none` 时，冷却天数将被忽略。

平台限制

- 只有 NAS 卷（NFS 和 SMB）才支持自动分层。
- 块卷 (iSCSI) 不支持自动分层。
- Google Cloud NetApp Volumes 存储池必须在 Google Cloud 中启用自动分层。

支持的值

- `tieringMinimumCoolingDays` 的有效范围：2 到 183
- 默认值：31

配置 Google Cloud NetApp Volumes ONTAP 模式存储池

从 Trident 26.06 开始，您可以配置现有 `ontap-san` 和 `ontap-nas` 后端以在 Google Cloud NetApp Volumes (GCNV) ONTAP 模式存储池上预置存储。此功能是技术预览版。

备注	此功能是 Trident 26.06 中的技术预览功能。技术预览功能不支持用于生产环境。在正式发布 (GA) 之前，相关功能、配置字段和支持矩阵可能会发生变化。
----	--

工作原理

Trident 重用现有的 `ontap-san` 和 `ontap-nas` 驱动程序逻辑，并通过 GCNV 代理端点路由 ONTAP REST 调用。这种方法使您能够在 GCNV 托管的 ONTAP 集群上置备和管理存储，同时保持与直接 ONTAP 后端相同的操作 workflow。ONTAP 模式不会引入新的存储驱动程序。您可以通过向 `ontap-san` 或 `ontap-nas` 后端添加 `gcnv` 配置块来针对每个后端启用它。

配置路径如下所示：

PersistentVolumeClaim → Trident (`ontap-san` 或 `ontap-nas`) → ONTAP REST 客户端 → GCNV 代理 → GCNV 池中的 ONTAP 集群

支持和不支持的范围

此技术预览版支持以下功能：

- 驱动程序： `ontap-san` (iSCSI) 和 `ontap-nas` (NFS 或 SMB)。
- 通过代理的生命周期操作：后端初始化、卷创建和删除以及相关的 ONTAP REST 流。
- 身份验证模型：Workload Identity Pool (WIP)、服务帐户密钥和 Application Default Credentials (ADC) 回退。

以下内容超出了此技术预览的范围：

- `ontap-san-economy`、`ontap-nas-economy` 和 `ontap-nas-flexgroup` 驱动程序。
- ASA r2 personality 路径。
- ZAPI 回退。ONTAP 模式仅使用 ONTAP REST。

前提条件

在配置 ONTAP 模式后端之前，请确保您已具备以下条件：

- Trident 26.06 或更高版本。
- 目标项目和位置中的 GCNV ONTAP 模式存储池。
- 用于代理操作的 GCP IAM 权限，适用于您的环境。
- 包含 `proxyURL`、`projectNumber`、`location` 和 `poolID` 的后端配置。
- 对于 iSCSI SAN 工作流，您的平台的节点端 iSCSI 和多路径先决条件。请参阅 ["准备工作节点"](#)。

配置后端

A `TridentBackendConfig` 需要 `credentials`。对于 ONTAP 模式，该密钥持有 GCP 代理凭据，而非 ONTAP `managementLIF/username/password`。 `svm` 是可选的：每个 GCNV ONTAP 模式池当前有一个 SVM，当省略时，Trident 会从池中派生它。

使用与本机 GCNV 相同的服务帐户模式：非敏感字段位于 `gcnv.apiKey`、`private_key` 中，敏感字段位于 `private_key_id` 中。请参阅 ["Google Cloud NetApp Volumes 配置示例"](#)。

ontap-san 后端

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-ontap-san
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-san
  backendName: gcnv-ontap-san
  credentials:
```

```

    name: gcnv-sa-secret
    type: secret
  gcnv:
    proxyURL: "https://netapp.googleapis.com"
    projectNumber: "<project-number>"
    location: "<region-or-zone>"
    poolID: "<pool-id>"
    svm: "<svm-name>"

```

ontap-nas 后端

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: gcnv-ontap-nas
  credentials:
    name: gcnv-sa-secret
    type: secret
  gcnv:
    proxyURL: "https://netapp.googleapis.com"
    projectNumber: "<project-number>"
    location: "<region-or-zone>"
    poolID: "<pool-id>"
    apiKey:
      type: service_account
      project_id: "<project-id>"
      client_email: "<service-account-email>"
      client_id: "<client-id>"
      auth_uri: "https://accounts.google.com/o/oauth2/auth"
      token_uri: "https://oauth2.googleapis.com/token"
      auth_provider_x509_cert_url:
        "https://www.googleapis.com/oauth2/v1/certs"
      client_x509_cert_url: "<client-x509-cert-url>"

```

密钥

```

apiVersion: v1
kind: Secret
metadata:
  name: gcnv-sa-secret

```

```

namespace: trident
type: Opaque
stringData:
  private_key_id: "<private-key-id>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <private-key>
    -----END PRIVATE KEY-----

```

备注 请勿将 `private_key` 或 `private_key_id` 放入 `TridentBackendConfig spec` 中——CRD 会拒绝它们。使用 `ontap-san`，并采用相同的 `gcnv` 和凭据模式进行块存储。

```
svm: "<svm-name>"
```

== Authentication

Trident resolves credentials for proxy access in the following order:

- . Workload Identity Pool (WIP)
- . Service account key
- . Application Default Credentials (ADC)

For this Tech Preview, use the following practices:

- * Prefer Workload Identity Pool where it is available.
- * Do not embed raw private keys in version-controlled backend files. Store credentials in a Kubernetes secret.
- * Apply least-privilege IAM scoping to the service account.

== Validation and initialization behavior

Trident validates an ONTAP-mode backend during initialization and fails fast rather than partially initializing.

Backend initialization fails if a required `gcnv` field is missing, the driver is unsupported for ONTAP-mode, or credential resolution fails.

== Known limitations

- * This feature is a Tech Preview. Behavior and the supported matrix can change before GA.
- * ZAPI fallback is not used in ONTAP-mode.
- * Protocol and node-readiness requirements still apply. For example, iSCSI workflows require the node-side prerequisites described in

xref:{relative_path}../trident-use/worker-node-prep.html[Prepare the worker node].

- * Existing ONTAP and GCNV environment constraints still apply.

== Upgrade and compatibility

ONTAP-mode is opt-in per backend through the `gcnv` configuration block:

- * Existing non-GCNV ONTAP backends are unaffected.

- * Mixed deployments that combine direct ONTAP backends and ONTAP-mode proxy backends are supported through backend-level configuration. Each backend is configured independently.

- * To stop using ONTAP-mode, remove or replace the affected backends. No global switch is required.

== What's next?

Apply secret before TBC; use `kubectl apply` and `-n trident` to match examples.

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。