



Google Cloud NetApp卷

Trident

NetApp
January 14, 2026

This PDF was generated from <https://docs.netapp.com/zh-cn/trident-2502/trident-use/gcnv.html> on January 14, 2026. Always check docs.netapp.com for the latest.

目录

Google Cloud NetApp卷	1
配置Google Cloud NetApp卷后端	1
Google Cloud NetApp卷驱动程序详细信息	1
适用于GKE的云身份	1
准备配置Google Cloud NetApp卷后端	3
NFS卷的前提条件	3
Google Cloud NetApp Volumes后端配置选项和示例	3
后端配置选项	3
卷配置选项	4
示例配置	4
下一步是什么?	12
存储类定义	13

Google Cloud NetApp卷

配置Google Cloud NetApp卷后端

现在、您可以将Google Cloud NetApp卷配置为Trident的后端。您可以使用Google Cloud NetApp卷后端连接NFS和SMB卷。

Google Cloud NetApp卷驱动程序详细信息

Trident提供了`google-cloud-netapp-volumes`用于与集群通信的驱动程序。支持的访问模式包括：
： *ReadWriteOnce*(RWO)、 *ReadOnlyMany*(ROX)、 *ReadWriteMany*(rwx)、 *ReadWriteOncePod*(RWOP)。

驱动程序	协议	卷模式	支持的访问模式	支持的文件系统
google-cloud-netapp-volumes	NFS SMB	文件系统	Rwo、ROX、rwx、RWOP	nfs、 smb

适用于GKE的云身份

通过云身份、Kubnetes Pod可以通过作为工作负载身份进行身份验证来访问Google Cloud资源、而不是提供明确的Google Cloud凭据。

要在Google Cloud中利用云身份、您必须：

- 使用GKE部署的Kubbernetes集群。
- 在GKE集群上配置的工作负载标识以及在节点池上配置的GKE元数据服务器。
- 具有Google Cloud NetApp卷管理员(角色/GCP .admin)角色或自定义角色的NetApp服务帐户。
- 已安装Trident、其中包括用于指定"gcp"的云提供程序和用于指定新GCP服务帐户的云标识。下面给出了一个示例。

Trident 运算符

要使用Trident运算符安装Trident，请编辑 `tridentorchestrator_cr.yaml` 以将设置为，并将 `cloudIdentity` 设置 `cloudProvider` 为 `"GCP" iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com`。

例如：

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "GCP"
  cloudIdentity: 'iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com'
```

掌舵

使用以下环境变量设置*云提供程序(CP)*和*云身份(CI)*标志的值：

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

以下示例将安装Trident并使用环境变量将设置 `cloudProvider` 为 `GCP` `CP`，并使用环境变量 `ANNOTATION` 设置 `cloudIdentity`：

```
helm install trident trident-operator-100.2502.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$ANNOTATION"
```

<code></code>

使用以下环境变量设置*云提供程序*和*云身份*标志的值：

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

以下示例将安装Trident并将标志设置 `cloud-provider` 为 `CP`、和 `cloud-identity` `ANNOTATION`：

```
tridentctl install --cloud-provider=$CP --cloud
-identity="$ANNOTATION" -n trident
```

准备配置Google Cloud NetApp卷后端

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

NFS卷的前提条件

如果您是首次使用Google Cloud NetApp卷或在新位置使用、则需要进行一些初始配置才能设置Google Cloud NetApp卷和创建NFS卷。请参阅 ["开始之前"](#)。

在配置Google Cloud NetApp卷后端之前、请确保您满足以下条件：

- 配置有Google Cloud NetApp卷服务的Google Cloud帐户。请参阅 ["Google Cloud NetApp卷"](#)。
- 您的Google Cloud帐户的项目编号。请参阅 ["确定项目"](#)。
- 具有NetApp卷管理员角色的Google Cloud服务帐户 (`roles/netapp.admin`)。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。请参见 ["创建服务帐户密钥"](#)
- 存储池。请参阅 ["存储池概述"](#)。

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

Google Cloud NetApp Volumes后端配置选项和示例

了解Google Cloud NetApp卷的后端配置选项并查看配置示例。

后端配置选项

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

参数	说明	默认
<code>version</code>		始终为 1
<code>storageDriverName</code>	存储驱动程序的名称	的值 <code>storageDriverName</code> 必须指定为"gosle-Cloud NetApp-volumes"。
<code>backendName</code>	(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
<code>storagePools</code>	用于指定用于创建卷的存储池的可选参数。	
<code>projectNumber</code>	Google Cloud 帐户项目编号。此值可在Google Cloud 门户主页上找到。	
<code>location</code>	Trident创建GCNV卷的Google Cloud位置。创建跨区域Kubernetes集群时、在中创建的卷 <code>location</code> 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成本。	

参数	说明	默认
apiKey	具有角色的Google Cloud服务帐户的API密钥 netapp.admin。它包括 Google Cloud 服务帐户专用密钥文件的 JSON 格式的内容（逐字复制到后端配置文件）。 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	精细控制 NFS 挂载选项。	"nfsvers=3"
limitVolumeSize	如果请求的卷大小超过此值、则配置失败。	""（默认情况下不强制实施）
serviceLevel	存储池及其卷的服务级别。值为 flex、 、 standard premium`或 `extreme。	
network	用于GCNV卷的Google Cloud网络。	
debugTraceFlags	故障排除时要使用的调试标志。例如， { "api":false, "method":true}。除非您正在进行故障排除并需要详细的日志转储，否则请勿使用此功能。	空
nasType	配置NFS或SMB卷创建。选项为 nfs、 `smb`或null。默认情况下、将设置为空会将NFS卷设置为空。	nfs
supportedTopologies	表示此后端支持的区域和区域的列表。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部分中控制默认卷配置 defaults。

参数	说明	默认
exportRule	新卷的导出规则。必须是IPv4地址任意组合的逗号分隔列表。	"0.0.0.0/0"
snapshotDir	对目录的访问权限 .snapshot	对于NFSv4、为"TRUE"；对于NFSv3、为"false"
snapshotReserve	为快照预留的卷百分比	""(接受默认值0)
unixPermissions	新卷的UNIX权限(4个八进制数字)。	""

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。

最低配置

这是绝对的最低后端配置。使用此配置时、Trident会发现在配置位置委派给Google Cloud NetApp卷的所有存储池、并随机将新卷放置在其中一个池上。由于 `nasType` 省略了、因此会 `nfs` 应用默认设置、后端将为NFS卷配置。

当您刚刚开始使用Google Cloud NetApp卷并尝试操作时、此配置是理想的选择、但实际上、您很可能需要为您配置的卷提供额外的范围界定。

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq7OlwWgLwGa==\n
    -----END PRIVATE KEY-----\n

---
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
```



```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----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

```

此后端配置可在一个文件中定义多个虚拟池。虚拟池在一节中进行了定义 `storage`。如果您有多个存储池支持不同的服务级别、而您希望在Kubbernetes中创建表示这些服务级别的存储类、则这些选项板非常有用。虚拟池标签用于区分池。例如、在以下示例中 `performance`、标签和 `serviceLevel` 类型用于区分虚拟池。

此外、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: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----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的云身份

```

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

```

支持的拓扑配置

Trident可以根据区域和可用性区域为工作负载配置卷。`supportedTopologies` 此后端配置中的块用于提供每个后端的区域和分区列表。此处指定的区域和分区值必须与每个Kubernetes集群节点上标签中的区域和分区值匹配。这些区域和分区表示可在存储类中提供的允许值列表。对于包含后端提供的部分区域和区域的存储类、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，则后端配置出现问题。您可以使用命令说明后端 `kubectl get tridentbackendconfig <backend-name>`、或者运行以下命令查看日志以确定原因：

```
tridentctl logs
```

确定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

下面是有关上述后端的基本 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"
```

使用字段的示例定义 **parameter.selector** :

使用、parameter.selector 您可以为每个指定 StorageClass ["虚拟池"](#) 用于托管卷的。卷将在选定池中定义各个方面。

```

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

```

有关存储类的详细信息，请参见 ["创建存储类。"](#)。

SMB卷的示例定义

使用 `nasType`、`node-stage-secret-name``和 ``node-stage-secret-namespace`，您可以指定SMB卷并提供所需的Active Directory凭据。任何具有任何/无权限的Active Directory用户/密码均可用作节点阶段密钥。

默认命名空间上的基本配置

```
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: "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`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
ACCESS MODES	STORAGECLASS	AGE	
RWX	gcnv-nfs-sc	1m	

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。