Google Cloud administration

Cloud Volumes ONTAP

NetApp
March 10, 2026

Table of Contents

Google Cloud administration

Change the Google Cloud machine type for Cloud Volumes ONTAP

You can choose from several machine types when you launch Cloud Volumes ONTAP in Google Cloud. You can change the instance or machine type at any time if you determine that it is undersized or oversized for your needs.

About this task

- Automatic giveback must be enabled on a Cloud Volumes ONTAP HA pair (this is the default setting). If it isn't, then the operation will fail.

  ONTAP 9 Documentation: Commands for configuring automatic giveback

- Changing the machine type can affect Google Cloud service charges.
- The operation restarts Cloud Volumes ONTAP.

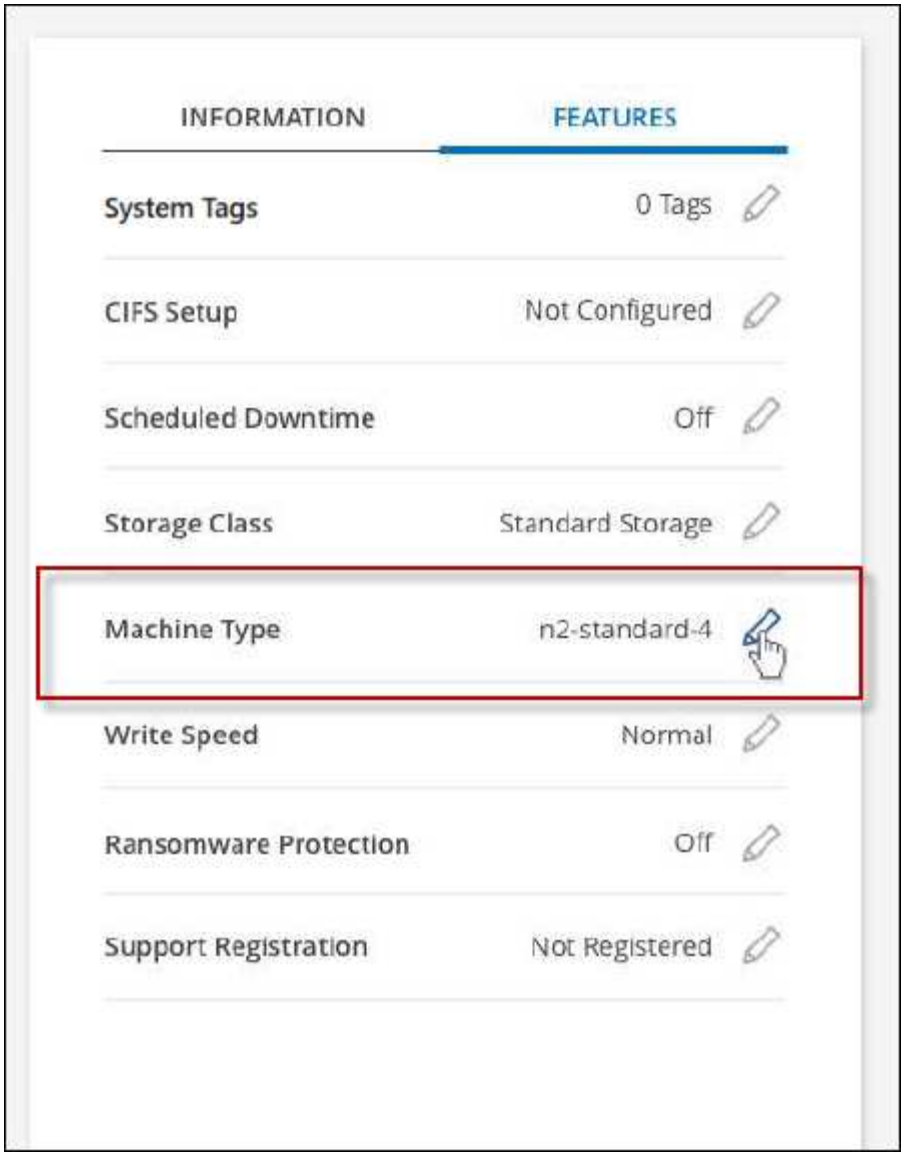
  For single-node systems, I/O is interrupted.

  For HA pairs, the change is nondisruptive. HA pairs continue to serve data.

The NetApp Console changes one node at a time by initiating takeover and waiting for give back. NetApp's Quality Assurance team tested both writing and reading files during this process and didn't see any issues on the client side. As connections changed, some retries were observed on the I/O level, but the application layer overcame the rewiring of NFS/CIFS connections.

Steps

1. On the **Systems** page, select the system.
2. On the Overview tab, click the Features panel and then click the pencil icon next to **Machine type**.



If you are using a node-based pay-as-you-go (PAYGO) license, you can optionally choose a different license and machine type by clicking the pencil icon next to **License type**.

1. Choose an machine type, select the check box to confirm that you understand the implications of the change, and then click **Change**.

Result

Cloud Volumes ONTAP reboots with the new configuration.

Convert existing Cloud Volumes ONTAP deployments to Infrastructure Manager

Beginning on February 09, 2026, new Cloud Volumes ONTAP deployments in Google Cloud can use Google Cloud Infrastructure Manager. Google is about to deprecate Google Cloud Deployment Manager in favor of Infrastructure Manager. Therefore, you need to manually run a transition tool to convert your existing Cloud Volumes ONTAP deployments from Deployment Manager to Infrastructure Manager. This is a one-time

process, after which your systems will automatically start using Infrastructure Manager.

About this task

The transition tool is available in the NetApp Support site, and creates the following artifacts:

- Terraform artifacts, saved in `conversion_output/deployment_name`.
- Summary of the conversion, saved in `conversion_output/batch_summary_<deployment_name>_<timestamp>.json`.
- Debug logs, saved in the `<gcp project number>-<region>-blueprint-config/<cvo name>` directory. You need these logs for troubleshooting. The `<gcp project number>-<region>-blueprint-config` bucket stores the Terraform logs.

> Cloud Volumes ONTAP systems using Infrastructure Manager store data and records in Google Cloud Storage buckets. You might incur extra costs for these buckets, but do not edit or delete the buckets or their content:
>
> - `gs://netapp-cvo-infrastructure-manager-<project id>`: for ONTAP versions and SVM Terraform Templates used for new Cloud Volumes ONTAP deployments. Inside this, the `dm-to-im-convert` bucket contains the Cloud Volumes ONTAP Terraform files.
> - `<gcp project number>-<region>-blueprint-config`: for storing Google Cloud Terraform artifacts.

Before you begin

- Ensure that your Cloud Volumes ONTAP system is 9.16.1 or later.
- Ensure that none of the Cloud Volumes ONTAP resources or their properties have been manually edited from the Google Cloud Console.
- Ensure that the Google Cloud APIs are enabled. Refer to Enable Google Cloud APIs. Ensure that along with the other APIs, you enable the Google Cloud Quotas API.
- Verify that the NetApp Console agent's service account has all required permissions. Refer to Google Cloud permissions for the Console agent.

  For private mode deployments, ensure these additional prerequisites:

  - Ensure that you have the latest Console agent version. Download the product installer from the NetApp Support Site and then manually install the agent on your host so that the agent can use Infrastructure Manager APIs.
  - If you are running the tool in a private mode, ensure that along with the other APIs, you have enabled the Cloud Build API. Enable Google Cloud APIs.
  - Ensure that you have completed the network configurations and created the worker pool for private mode deployments. Refer to Infrastructure Manager configuration for private mode deployments.

- The conversion tool uses the following domains. Enable them on port 443 in your network:

Domain	Port	Protocol	Direction	Purpose
cloudresourcemanager.googleapis.com	443	TCP	EGRESS	Project validation
deploymentmanager.googleapis.com	443	TCP	EGRESS	Deployment discovery
config.googleapis.com	443	TCP	EGRESS	Infrastructure Manager API
storage.googleapis.com	443	TCP	EGRESS	GCS bucket operations
iam.googleapis.com	443	TCP	EGRESS	Service account validation
compute.googleapis.com	443	TCP	EGRESS	Compute API calls used by Google Cloud and Terraform Import and Plan
cloudbuild.googleapis.com	443	TCP	EGRESS	Build operations only required for private mode
openidconnect.googleapis.com	443	TCP	EGRESS	Authentication
oauth2.googleapis.com	443	TCP	EGRESS	OAuth2 token exchange
registry.terraform.io	443	TCP	EGRESS	Terraform provider registry
releases.hashicorp.com	443	TCP	EGRESS	Terraform binary downloads
apt.releases.hashicorp.com	443	TCP	EGRESS	HashiCorp APT repository
us-central1-docker.pkg.dev	443	TCP	EGRESS	GCP Artifact Registry
metadata.google.internal	80	HTTP	Internal	VM metadata & auth tokens
pypi.org	443	TCP	EGRESS	Python package index
files.pythonhosted.org	443	TCP	EGRESS	Python package downloads
checkpoint-api.hashicorp.com	443	TCP	EGRESS	Terraform version check
download.docker.com	443	TCP	EGRESS	Docker APT repository
security.ubuntu.com	80/443	TCP	EGRESS	Ubuntu security updates

Domain	Port	Protocol	Direction	Purpose
*.gce.archive.ubuntu.com	80	TCP	EGRESS	Ubuntu package mirror

Prepare the environment for running the tool

Run these steps before running the tool.

Steps

1. Create a role and attach it to a service account:
   a. Create a YAML file with the following permissions:

```
title: NetApp Dm TO IM Convert Solution
description: Permissions for the service account associated with the
VM where the tool will run.
stage: GA
includedPermissions:
- compute.addresses.get
- compute.disks.get
- compute.forwardingRules.get
- compute.healthChecks.get
- compute.instanceGroups.get
- compute.instances.get
- compute.regionBackendServices.get
- config.deployments.create
- config.deployments.get
- config.deployments.getLock
- config.deployments.lock
- config.deployments.unlock
- config.deployments.update
- config.deployments.delete
- config.deployments.updateState
- config.operations.get
- deploymentmanager.deployments.get
- deploymentmanager.deployments.list
- deploymentmanager.manifests.get
- iam.serviceAccounts.get
- storage.buckets.create
- storage.objects.create
- storage.objects.delete
- storage.objects.get
- storage.objects.list
```

Include additional permission for private mode deployments

If you are running the tool in a private mode, add the `cloudbuild.workerpools.get` permission also to the YAML file.

b. Create a custom role in Google Cloud with the permissions defined in the YAML file.
`gcloud iam roles create dmtoim_convert_tool_role --project=PROJECT_ID \ --file=YAML_FILE_PATH`
For more information, refer to Creating and managing custom roles.

c. Attach the custom role to the service account that you'll use to create the VM.

d. Add the `roles/iam.serviceAccountUser` role to this service account. Refer to Service accounts overview.

2. Create a VM with the following configurations. You run the tool on this VM.
   - Machine Type: Google Compute Engine machine type e2-medium
   - OS: Based on your requirement, select either of these images:
     - Ubuntu 25.10 AMD64 Minimal (image: ubuntu-minimal-2510-amd64)
     - SUSE Linux Enterprise Server 15 SP7 x86_64
   - Networking: Firewall allowing HTTP and HTTPs
   - Disk Size: 20GB
   - Security: Service accounts: the service account you created
   - Security: Access Scope - access set for each API:
     - Cloud Platform: Enabled
     - Compute Engine: Read only
     - Storage: Read only (default)
     - Google Cloud Logging (previously Stackdriver Logging) API: Write only (default)
     - Stackdriver Monitoring (now part of Google Cloud Operations) API: Write only (default)
     - Service Management: Read only (default)
     - Service Control: Enabled (default)
     - Google Cloud Trace (previously Stackdriver Trace): Write only (default)
3. Connect to the newly created VM using SSH: `gcloud compute ssh dmtoim-convert-executor-vm --zone <region where VM is deployed>`
4. Download the conversion tool from the NetApp Support site by using your NSS credentials: `wget <download link from NetApp Support site>`
5. Extract the downloaded TAR file: `unzip <downloaded file name>`

Ubuntu

1. Download and install these prerequisite packages:
   - Docker: 28.2.2 build 28.2.2-0ubuntu1 or later
   - Terraform: 1.14.1 or later
   - Python: 3.13.7, python3-pip, python3 venv

   ```
   sudo apt-get update
   sudo apt-get install python3-pip python3-venv -y
   wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg
   --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
   echo "deb [arch=$(dpkg --print-architecture) signed-
   by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]
   https://apt.releases.hashicorp.com noble main" | sudo tee
   /etc/apt/sources.list.d/hashicorp.list
   sudo apt update && sudo apt install terraform
   sudo apt-get install -y docker.io
   sudo systemctl start docker
   ```

   Google Cloud CLI `gcloud` is preinstalled on the VM.

SUSE Linux Enterprise Server

1. Set up Python: `sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.11 2`
2. Install pip3 for installing the package: `python3.11 -m ensurepip --upgrade`
3. Install Terraform:

   ```
   wget
   https://releases.hashicorp.com/terraform/1.7.4/terraform_1.7.4_linux
   _amd64.zip
   unzip terraform_1.7.4_linux_amd64.zip
   sudo mv terraform /usr/local/bin/
   rm terraform_1.7.4_linux_amd64.zip
   ```

4. Install Google Cloud SDK (gcloud)

   ```
   curl https://sdk.cloud.google.com | bash
   exec -l $SHELL
   ```

Run the conversion tool

These steps are for both Ubuntu and SUSE Linux Enterprise Server for running the conversion tool.

Steps

1. Add the current user to the Docker group, so that the tool can use Docker without `sudo` privileges.

```
sudo usermod -aG docker $USER
newgrp docker
```

2. Install the conversion tool:

```
cd <folder where you extracted the tool>
 ./install.sh
```

This installs the tool in an isolated environment, `dmconvert-venv`, and verifies that all required software packages are installed.

3. Enter the environment where the tool is installed: `source dmconvert-venv/bin/activate`
4. Run the conversion tool as a `non-sudo` user. Ensure that you use the same service account as the Console agent's service account, and that the service account has all the necessary permissions for Google Cloud Infrastructure Manager.

```
dmconvert \
--project-id=<the Google Cloud project ID for the Cloud Volumes ONTAP
deployment> \
--cvo-name=<Cloud Volumes ONTAP system name> \
--service-account=<the service account attached to the Console agent>
```

Run the tool in private mode deployments

Specify the `--worker-pool` parameter to run the tool in private mode deployments. For worker pool configuration, refer to Infrastructure Manager configuration for private mode deployments.

```
dmconvert \
--project-id=<the Google Cloud project ID for the Cloud Volumes
ONTAP deployment> \
--cvo-name=<Cloud Volumes ONTAP system name> \
--service-account=<the service account attached to the Console
agent> \
--worker-pool=<worker pool name>
```

After you finish

The tool displays a list of all Cloud Volumes ONTAP systems and SVM details. When it finishes running, you can see the statuses of all the converted systems. Each converted system appears in the Google Console under Infrastructure Manager in a *<system-name-imdeploy>* format, indicating that the Console now uses Infrastructure Manager APIs to manage that Cloud Volumes ONTAP system.

> Post conversion, do not delete the deployment object for Deployment Manager in the Google Cloud Console. This deployment object contains information that you might need to roll back the converted system.

If you need to roll back the conversion, you must use the same VM. If you have converted all systems and do not need to roll back to Deployment Manager, you can delete the VM.

Roll back the conversion

If you don't want to continue with the conversion, you can roll back to Deployment Manager by following these steps:

Steps

1. On the same VM that you created for running the tool, run this command:

```
dmconvert \
--project-id=<the Google Cloud project ID for the Cloud Volumes ONTAP
deployment> \
--cvo-name=<Cloud Volumes ONTAP system name> \
--service-account=<the service account attached to the Console agent> \
--rollback
```

2. Wait till the rollback is complete.

Related links

- NetApp Console Agent 4.2.0 Release Notes
- Permissions required for Google Cloud Infrastructure Manager

Copyright information

Trademark information