



## **GCP workflows**

### **Cloud Manager Automation**

NetApp

March 05, 2022

This PDF was generated from [https://docs.netapp.com/us-en/cloud-manager-automation/cm/wf\\_gcp\\_before.html](https://docs.netapp.com/us-en/cloud-manager-automation/cm/wf_gcp_before.html) on March 05, 2022. Always check docs.netapp.com for the latest.

# Table of Contents

- GCP workflows . . . . . 1
  - Before you begin . . . . . 1
  - Working environments . . . . . 1
- Aggregates . . . . . 17
- Volumes . . . . . 22
- Metadata . . . . . 37

# GCP workflows

## Before you begin

There are several workflows that you can use to deploy and manage Cloud Volumes ONTAP in Google Cloud.



Review the [Get started](#) section before using any of the Cloud Manager REST API workflows.

## Workflow categories

The GCP workflows are organized into the following categories:

- Working environments
- Aggregates
- Volumes
- Metadata

See [Understanding the workflow processes](#) for more information on these categories.



Cloud Volumes ONTAP requires a Google Cloud service account to run several background automation tasks such as data tiering and backup service. [Learn more about GCP service accounts](#).

## Connector setup

You must have a **Connector** for the cloud environment before creating a working environment and performing other activities using the workflows. You can create a Connector using the Cloud Manager web UI. When you create a Connector, Cloud Manager adds the GCP cloud provider account that you deployed the Connector in to your list of available accounts. Your GCP account needs to have the right permissions in order to create a Connector.

Review [Learn about GCP Connectors](#) to know how to create and deploy a GCP Connector.

## Working environments

### Create a working environment with PAYGO

You can use this workflow to create a new GCP Cloud Volumes ONTAP working environment using pay-as-you-go (PAYGO) subscription.

**Note the following when using PAYGO:**

- A marketplace subscription is required.
- A NetApp Support Site (NSS) key is recommended to register the system for support, but it's not required.
- You can add more volumes after creating the working environment. You can choose to create a volume using either [NFS](#), [CIFS](#), or [iSCSI](#) protocol.

**1. Select the region**

Perform the workflow [Get regions](#) and do the following:

- Choose the `name` value of the required region for the `region` parameter in step 8.
- Choose one of the VPCs. Choose `name` for `vpcId` parameter and `subnets: path` for `subnetId` and `subnetPath` parameters in step 8.

**2. Select the workspace**

Perform the workflow [Get tenants](#) and choose the `workspacePublicId` value for the `tenantId` parameter in step 8.

**3. Select the projects**

Perform the workflow [Get projects](#) and choose the `projectId` value of the required project for `project` parameter in step 8.

**4. Select the permutations**

Perform the workflow [Get permutations](#) and choose the `ontapVersion`, `license: type`, and `instanceType` values of the required `vsaMetadata` parameter in step 8.

**5. Select the packages configuration**

Perform the workflow [Get packages](#) and search the `licenseType` from permutations:

- Choose the `name` for `packageName` parameter.
- Choose the `diskSize` for `gcpVolumeSize`.
- Choose the `diskType` and `writingSpeedState` for the `gcpVolumeType` and `writingSpeedState` parameters.

**6. Select the service account**

Perform the workflow [Get service accounts](#) workflow and choose the `email` value of the required service accounts for the `gcpServiceAccount` parameter in step 8.

**7. (Optional) Obtain an NSS key**

An NSS key is **optional** when using PAYGO subscription. If needed, you can create a key or select an existing key, and include the NSS key in the `nssAccount` parameter in step 8.

- To create a new NSS key using the Cloud Manager web user interface, perform the task [Generate NSS user ID](#) and choose the `id`.
- To select an existing NSS key, perform the [Get NSS keys](#) workflow and choose the `id` value of the required NSS user.

## 8. Create the working environment

HTTP method	Path
POST	/occm/api/gcp/vsa/working-environments

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/working-
environments' --header 'x-agent-id: <AGENT_ID>' //<1> --header
'Authorization: Bearer <ACCESS_TOKEN>' //<2> ---header 'Content-Type:
application/json' --d @JSONinput
```

- (1) Replace `<AGENT_ID>` with your agent ID.
- (2) Replace `<ACCESS_TOKEN>` with your obtained bearer access token.

### Input

The JSON input example includes the minimum list of parameters.



This request uses PAYGO as indicated in the `licenseType` parameter.

### JSON input example

```

{
  "name": "zivgcp01we03",
  "svmPassword": "password",
  "vpcId": "default",
  "region": "us-west1-b",
  "tenantId": "tenantID",
  "subnetPath": "projects/occm-dev/regions/us-west1/subnetworks/default",
  "subnetId": "projects/occm-dev/regions/us-west1/subnetworks/default",
  "dataEncryptionType": "GCP",
  "vsaMetadata": {
    "ontapVersion": "ONTAP-9.9.0X4.T1.gcp",
    "licenseType": "gcp-cot-explore-paygo",
    "instanceType": "custom-4-16384"
  },
  "gcpVolumeSize": {
    "size": 100,
    "unit": "GB"
  },
  "gcpVolumeType": "pd-ssd",
  "gcpLabels": [],
  "writingSpeedState": "NORMAL",
  "packageName": "gcp_poc",
  "gcpServiceAccount": "xxxxxx@occm-dev.iam.xxx.com",
  "project": "occm-dev",
  "backupVolumesToCbs": false
}

```

## JSON output example

```

{
  "publicId": "vsaworkingenvironment-2qkd75xv",
  "name": "zivgcp01we03",
  "tenantId": "tenantIDshownhere",
  "svmName": "svm_zivgcp01we03",
  "creatorUserEmail": "user_email",
  "status": null,
  "providerProperties": null,
  "reservedSize": null,
  "clusterProperties": null,
  "ontapClusterProperties": null,
  "cloudProviderName": "GCP",
  "snapshotPolicies": null,
  "actionsRequired": null,
  "activeActions": null,
  "replicationProperties": null,
  "schedules": null,
  "svms": null,
  "workingEnvironmentType": "VSA",
  "supportRegistrationProperties": null,
  "supportRegistrationInformation": null,
  "capacityFeatures": null,
  "encryptionProperties": null,
  "supportedFeatures": null,
  "isHA": false,
  "haProperties": null,
  "k8sProperties": null,
  "fpolicyProperties": null,
  "saasProperties": null,
  "cbsProperties": null,
  "complianceProperties": null,
  "monitoringProperties": null
}

```

## Create a working environment with BYOL

You can use this workflow to create a new Cloud Volumes ONTAP working environment using bring your own license (BYOL) licensing.

**Note the following when using BYOL licensing:**

- A marketplace subscription is not required.
- A NetApp Support Site (NSS) key is required to register the system for support.
- You can add more volumes after creating the working environment. You can choose to create a volume using either [NFS](#), [CIFS](#), or [iSCSI](#) protocol.

**1. Select the region**

Perform the workflow [Get regions](#) and do the following:

- Choose the `name` value of the required region for the `region` parameter in step 8.
- Choose one of the VPCs. Choose `name` for `vpcId` parameter and `subnets: path` for `subnetId` and `subnetPath` parameters in step 8.

**2. Select the workspace**

Perform the workflow [Get tenants](#) and choose the `workspacePublicId` value for the `tenantId` parameter in step 8.

**3. Select the project**

Perform the workflow [Get projects](#) and choose the `projectId` value of the required project for `project` parameter in step 8.

**4. Select the permutations**

Perform the workflow [Get permutations](#) and choose the `ontapVersion`, `license: type`, and `instanceType` values of the required `vsaMetadata` parameter in step 8.

**5. Select the packages configuration**

Perform the workflow [Get packages](#) and search the `licenseType` from permutations:

- Choose the `name` for `packageName` parameter.
- Choose the `diskSize` for `gcpVolumeSize`.
- Choose the `diskType` and `writingSpeedState` for the `gcpVolumeType` and `writingSpeedState` parameters.

**6. Select the service account**

Perform the workflow [Get service accounts](#) workflow and choose the `email` value of the required service accounts for the `gcpServiceAccount` parameter in step 8.

**7. Obtain an NSS key**

An NSS key is **required** when using BYOL subscription. If needed, you can create a key or select an existing key, and include the NSS key in the `nssAccount` parameter in step 8.



- To create a new NSS key using the Cloud Manager web user interface, perform the task [Generate NSS user ID](#) and choose the `id`.
- To select an existing NSS key, perform the [Get NSS keys](#) workflow and choose the `id` value of the required NSS user.

## 8. Create the working environment

HTTP method	Path
POST	/occm/api/gcp/vsa/working-environments

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/working-
environments' --header 'x-agent-id: <AGENT_ID>' //<1> --header
'Authorization: Bearer <ACCESS_TOKEN>' //<2> ---header 'Content-Type:
application/json' --d @JSONinput
```

- (1) Replace `<AGENT_ID>` with your agent ID.  
(2) Replace `<ACCESS_TOKEN>` with your obtained bearer access token.

### Input

The JSON input example includes the minimum list of parameters. This request uses BYOL licensing as indicated in the `licenseType` parameter. The `serialNumber` is required.

### JSON input example

```

{
  name: "gcpwe123"
  backupVolumesToCbs: true
  capacityTier: "cloudStorage"
  dataEncryptionType: "GCP"
  enableCompliance: true
  gcpLabels: []
  gcpServiceAccount: "fabric-pool@occm-dev.iam.gserviceaccount.com"
  gcpVolumeSize: {size: 500, unit: "GB", _identifier: "500 GB"}
  gcpVolumeType: "pd-ssd"
  name: "gcpwe123"
  nssAccount: "0xxx-000-4c70-9cee-304f36b74db6"
  packageName: "gcp_poc"
  project: "occm-dev"
  region: "europe-west3-c"
  serialNumber: "00000108000000000000"
  subnetId: "projects/occm-dev/regions/europe-west3/subnetworks/vpc4qa-2-europe-west3"
  subnetPath: "projects/occm-dev/regions/europe-west3/subnetworks/vpc4qa-2-europe-west3"
  svmPassword: "Netappl23"
  tenantId: "workspaceNqaJyVMz"
  tierLevel: "standard"
  volume: {
    exportPolicyInfo: {
      policyType: "custom",
      ips: ["172.22.13.0/24"],
      nfsVersion: ["nfs3", "nfs4"]
    }
  }
}
vpcId: "vpc4qa-2"
vsaMetadata: {
  ontapVersion: "ONTAP-9.10.1RC1.T1.gcp",
  licenseType: "gcp-cot-premium-byol"
}
instanceType: "n2-standard-4"
licenseType: "gcp-cot-premium-byol"
ontapVersion: "ONTAP-9.10.1RC1.T1.gcp"
writingSpeedState: "NORMAL"
}

```

## Output

The JSON output example includes an example of the `VsaWorkingEnvironmentResponse` response.

JSON output example

```

{
  "publicId": "vsaworkingenvironment-9nhkrtu0",
  "name": "yuvalbyol3101",
  "tenantId": "tenantIDshownhere",
  "svmName": "svm_yuvalbyol3101",
  "creatorUserEmail": "user_email",
  "status": null,
  "providerProperties": null,
  "reservedSize": null,
  "clusterProperties": null,
  "ontapClusterProperties": null,
  "cloudProviderName": "GCP",
  "snapshotPolicies": null,
  "actionsRequired": null,
  "activeActions": null,
  "replicationProperties": null,
  "schedules": null,
  "svms": null,
  "workingEnvironmentType": "VSA",
  "supportRegistrationProperties": null, "supportRegistrationInformation":
null,
  "capacityFeatures": null,
  "encryptionProperties": null,
  "supportedFeatures": null,
  "isHA": false,
  "haProperties": null,
  "fpolicyProperties": null,
  "saasProperties": null,
  "cbsProperties": null,
  "complianceProperties": null,
  "monitoringProperties": null,
  "licensesInformation": null
}

```

## Get working environment

You can retrieve the public identifier, working environment ID, the storage virtual machine name for Cloud Volumes ONTAP working environments and other Cloud Volumes ONTAP related details (visible to currently logged in user) which would be used in other workflows.

### 1. Get the working environments

HTTP method	Path
GET	/occm/api/gcp/vsa/working-environments/{workingEnvironmentId}

## curl

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/working-
environments' --header 'Content-Type: application/json' --header 'x-agent-
id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

## Input

(Optional) Query parameters:

- fields string
- tenantId string

## Output

The JSON output example includes details of a single node Google Cloud working environment.

## JSON output example

```
[
  {
    "publicId": "vsaworkingenvironment-sfrf3wvj",
    "name": "zivgcp01we02",
    "tenantId": "tenantIDshownhere",
    "svmName": "svm_zivgcp01we02",
    "creatorUserEmail": "user_email",
    "status": null,
    "providerProperties": null,
    "reservedSize": null,
    "clusterProperties": null,
    "ontapClusterProperties": null,
    "cloudProviderName": "GCP",
    "snapshotPolicies": null,
    "actionsRequired": null,
    "activeActions": null,
    "replicationProperties": null,
    "schedules": null,
    "svms": null,
    "workingEnvironmentType": "VSA",
    "supportRegistrationProperties": null,
    "supportRegistrationInformation": [],
    "capacityFeatures": null,
    "encryptionProperties": null,
    "supportedFeatures": null,
  }
]
```

```

    "isHA": false,
    "haProperties": null,
    "k8sProperties": null,
    "fpolicyProperties": null,
    "saasProperties": null,
    "cbsProperties": null,
    "complianceProperties": null,
    "monitoringProperties": null
  },
  {
    "publicId": "vsaworkingenvironment-2qkd75xv",
    "name": "zivgcp01we03",
    "tenantId": "tenantIdshownhere",
    "svmName": "svm_zivgcp01we03",
    "creatorUserEmail": "user_email",
    "status": null,
    "providerProperties": null,
    "reservedSize": null,
    "clusterProperties": null,
    "ontapClusterProperties": null,
    "cloudProviderName": "GCP",
    "snapshotPolicies": null,
    "actionsRequired": null,
    "activeActions": null,
    "replicationProperties": null,
    "schedules": null,
    "svms": null,
    "workingEnvironmentType": "VSA",
    "supportRegistrationProperties": null,
    "supportRegistrationInformation": [],
    "capacityFeatures": null,
    "encryptionProperties": null,
    "supportedFeatures": null,
    "isHA": false,
    "haProperties": null,
    "k8sProperties": null,
    "fpolicyProperties": null,
    "saasProperties": null,
    "cbsProperties": null,
    "complianceProperties": null,
    "monitoringProperties": null
  }
]

```

## Delete a working environment

You can delete an existing GCP Cloud Volumes ONTAP working environment.

### 1. Select the working environment

Perform the workflow [Get working environment](#) and choose the `publicId` value of the working environment for the `workingEnvironmentId` path parameter.

### 2. Delete the working environment

HTTP method	Path
DELETE	/occm/api/gcp/vsa/working-environments/{workingEnvironmentId}

### Curl example

```
curl --location --request DELETE
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/working-
environments/<WORKING_ENV_ID>' --header 'Content-Type: application/json'
--header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer
<ACCESS_TOKEN>'
```

### Input

Path parameter `<WORKING_ENV_ID>` (`workingEnvironmentId`) string

(Optional) Query parameters:

- `localDelete` boolean

If `true` the Cloud Volumes ONTAP instance in the cloud is not terminated, but Cloud Manager no longer manages it (default is `false`). If `false` the Cloud Volumes ONTAP instance is deleted including all the cloud resources created for this working environment.

- `forceDelete` boolean

If `true` the working environment is deleted even if it is part of one or more SnapMirror relationships (default is `false`).

### Output

None

## Create CIFS server configuration

If you want to create CIFS volumes on your Cloud Volumes ONTAP system, you first need to configure the CIFS server. You can choose to set up the CIFS server in a workgroup or in an Active Directory domain. Review the [ONTAP documentation](#) for more information.

Choose the workflow that is specific to your goal:

- [Set up a CIFS server in a workgroup](#)
- [Set up a CIFS server in an Active Directory domain](#)

## Set up a CIFS server in a workgroup

You can configure a CIFS server in a workgroup when the Microsoft Active Directory domain infrastructure is not available.

### 1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the working environment used in the `workingEnvironmentId` path parameter.

### 2. Create the CIFS configuration

Create the CIFS server configuration.

HTTP method	Path
POST	/occm/api/gcp/vsa/working-environments/{workingEnvironmentId}/cifs-workgroup

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/working-
environments/<WORKING_ENV_ID>/cifs-workgroup' --header 'Content-Type:
application/json' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

### Input

- Path parameter `<WORKING_ENV_ID>` `workingEnvironmentId` string

### JSON input example

```
{
  "serverName": "SMB_SERVER02",
  "workgroupName": "workgroup02",
  "svmName": "svm_ziv01we01"
}
```

### Output

None.

## Set up a CIFS server in an Active Directory domain

You can create a CIFS server on the SVM and specify the Active Directory (AD) domain to which it belongs.

### 1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the working environment used in the `workingEnvironmentId` path parameter.

### 2. Determine the Active Directory configuration

You need the following configuration parameters for an Active Directory server.

Input parameter	Description
<code>dnsDomain</code>	Use the Active Directory domain as the DNS name.
<code>ipAddresses</code>	Define the primary DNS IP address and optionally add a secondary IP address.
<code>netBIOS</code>	Use the CIFS server NetBIOS name.
<code>organizationalUnit</code>	Include the organizational unit as appropriate.
<code>activeDirectoryDomain</code>	Set the Active Directory domain to join.
<code>activeDirectoryUsername</code>	A username with authorization to join the domain.
<code>activeDirectoryPassword</code>	The password for the authorized username.

### 3. Create the CIFS configuration

Create the CIFS server configuration.

HTTP method	Path
POST	<code>/occm/api/gcp/vsa/working-environments/{workingEnvironmentId}/cifs</code>

#### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/working-
environments/<WORKING_ENV_ID>/cifs' --header 'Content-Type:
application/json' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

#### Input

- Path parameter `<WORKING_ENV_ID>` `workingEnvironmentId` string

#### JSON input example



```
{
  "dnsDomain": "zivh.netapp.com",
  "ipAddresses": [
    "172.31.5.241"
  ],
  "netBIOS": "zivaws02we03",
  "organizationalUnit": "CN=Computers",
  "activeDirectoryDomain": "zivh.netapp.com",
  "activeDirectoryUsername": "administrator",
  "activeDirectoryPassword": "password"
}
```

## Output

None.

## Get CIFS server configurations

You can use this workflow to retrieve the CIFS server configurations for an existing Cloud Volumes ONTAP working environment.

### 1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the working environment used in the `workingEnvironmentId` path parameter.

### 2. Get the CIFS configurations

HTTP method	Path
GET	/occm/api/gcp/vsa/working-environments/{workingEnvironmentId}/cifs

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/working-
environments/<WORKING_ENV_ID>/cifs' --header 'Content-Type:
application/json' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>'
```

## Input

- Path parameter `<WORKING_ENV_ID>` `workingEnvironmentId` string
- (Optional) Query parameter `svm` string

## Output

The JSON output example includes the CIFS configurations for an existing Cloud Volumes ONTAP ONTAP working environment.

## JSON output example

```
[
  {
    "dnsDomain": "zivh.netapp.com",
    "activeDirectoryDomain": "zivh.netapp.com",
    "ipAddresses": [
      "172.31.5.241"
    ],
    "netBIOS": "zivaws02we01",
    "organizationalUnit": "CN=Computers",
    "authenticationType": "domain"
  }
]
```

## Delete CIFS server configuration

You can use this workflow to delete a CIFS server configuration for an existing Cloud Volumes ONTAP working environment.

### 1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the working environment used in the `workingEnvironmentId` path parameter.

### 2. Delete the CIFS configurations

HTTP method	Path
POST	/occm/api/gcp/vsa/working-environments/{workingEnvironmentId}/delete-cifs

## curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/working-
environments/<WORKING_ENV_ID>/delete-cifs' --header 'Content-Type:
application/json' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>'
```

## Input

- Path parameter `<WORKING_ENV_ID>` `workingEnvironmentId` string
- Optional JSON body

```
{
  "activeDirectoryUsername": "string",
  "activeDirectoryPassword": "string",
  "svmName": "string"
}
```

## Output

None.

# Aggregates

## Get aggregates

You can retrieve a list of available disk aggregates for Cloud Volumes ONTAP in Google Cloud.

### 1. Select the working environment

Perform the workflow [Get working environment](#) and choose the `publicId` value of the working environment for the `workingEnvironmentId` path parameter.

### 2. Get the list of aggregates

HTTP method	Path
GET	/occm/api/gcp/vsa/aggregates/{workingEnvironmentId}

## curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/aggregates/<WORKING_ENV_ID>' --header 'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

## Input

Path parameter:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`) string

## Output

An array of aggregates for the indicated working environment is returned as shown in the JSON output example.



The capacity (sizes) in the output are in MB/GB/TB (1000th order) because these are ONTAP aggregates, whereas in Cloud Manager the capacity is specified as MiB, GiB (1024 order).

## JSON output example

```
[
  {
    "name": "aggr1",
    "availableCapacity": {
      "size": 87.55,
      "unit": "GB"
    },
    "totalCapacity": {
      "size": 88.57,
      "unit": "GB"
    },
    "usedCapacity": {
      "size": 1.02,
      "unit": "GB"
    },
    "volumes": [
      {
        "name": "svm_zivgcp01we02_root",
        "totalSize": {
          "size": 1.0,
          "unit": "GB"
        },
        "usedSize": {
          "size": 7.59124755859375E-4,
          "unit": "GB"
        },
        "thinProvisioned": false,
        "isClone": false,
        "rootVolume": true
      }
    ],
    "providerVolumes": [
      {
        "id": "000000000000000000",
        "name": "zivgcp01we02datadisk1",
        "size": {
          "size": 100.0,
          "unit": "GB"
        },
        "state": "READY",
        "device": "zivgcp01we02datadisk1",
        "instanceId": "zivgcp01we02",
        "diskType": "pd-ssd",
        "encrypted": true,
      }
    ]
  }
]
```

```

        "iops": null
    },
    ],
    "disks": [
        {
            "name": "NET-1.2",
            "position": "data",
            "ownerNode": "zivgcp01we02-01",
            "device": "zivgcp01we02datadisk1",
            "vmDiskProperties": null
        }
    ],
    "state": "online",
    "encryptionType": "cloudEncrypted",
    "encryptionKeyId": null,
    "isRoot": false,
    "homeNode": "zivgcp01we02-01",
    "ownerNode": "zivgcp01we02-01",
    "capacityTier": null,
    "capacityTierUsed": null,
    "sidlEnabled": true,
    "snaplockType": "non_snaplock"
}
]

```

## Create aggregate

You can create new aggregate within a Cloud Volumes ONTAP working environment using this workflow.

### 1. Select the working environment

Perform the workflow [Get working environment](#) and choose the `publicId` value for the `workingEnvironmentId` parameter in the JSON input.

### 2. Select the GCP disk types

Perform the [Get GCP disk types](#) workflow and choose the `size` and `supportedDiskType` values of the required `diskSize` and `providerVolumeType` parameters in the JSON input.

### 3. Create the aggregate

HTTP method	Path
POST	<code>occm/api/gcp/vsa/aggregates</code>

## curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/aggregates'
--header 'Content-Type: application/json' --header 'x-agent-id:
<AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

## Input

The JSON input example includes the minimum list of input parameters.

## JSON input example

```
{
  "name": "ziv01agg01",
  "workingEnvironmentId": "vsaworkingenvironment-sfrf3wvj",
  "numberOfDisks": 1,
  "diskSize": {
    "size": 100,
    "unit": "GB"
  },
  "providerVolumeType": "pd-ssd"
}
```

## Output

None

## Add disks to aggregate

You can add disks to an existing aggregate.

### 1. Select the working environment

Perform the workflow [Get GCP single node working environment](#) and choose the `publicId` value for the `workingEnvironmentId` path parameter.

### 2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose name of the required aggregate for the `aggregateName` path parameter.

### 3. Add the disks to the aggregate

HTTP method	Path
POST	/occm/api/gcp/vsa/aggregates/{workingEnvironmentId}/{aggregateName}/disks

## curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/aggregates/<WORKING_ENV_ID>/<AGGR_NAME>/disks' --header 'Content-Type: application/json'
--header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

## Input

You must include the following path parameters:

- <WORKING\_ENV\_ID> (workingEnvironmentId) string
- <AGGR\_NAME> (aggregateName) string

Also, the JSON input example includes an input parameter as shown.

## JSON input example

```
{
  "numberOfDisks": "1"
}
```

## Output

None

## Delete aggregate

You can delete an existing disk aggregate in a Cloud Volumes ONTAP working environment.

### 1. Select the working environment

Perform the workflow [Get working environment](#) and choose the `publicId` value of the working environment for the `workingEnvironmentId` path parameter.

### 2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the `name` value of the required aggregate for the `aggregateName` path parameter.

### 3. Delete the aggregate

HTTP method	Path
DELETE	/occm/api/gcp/vsa/aggregates/{workingEnvironmentId}/{aggregateName}

## curl example

```
curl --location --request DELETE
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/aggregates/<WORKING_ENV_ID>/<AGGR_NAME>' --header 'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

## Input

Path parameters:

- <WORKING\_ENV\_ID> (workingEnvironmentId) string
- <AGGR\_NAME> (aggregateName) string

## Output

None

# Volumes

## Create a volume using NFS

You can use this workflow to create a volume accessed through NFS protocol.



If the properties `aggregateName` and `maxNumOfDisksApprovedToAdd` are not provided on the REST API call, the response will fail with a suggested name for the aggregate and the number of disks needed to fulfill the request.

### 1. Select the working environment

Perform the workflow [Get working environment](#) and choose the `publicId` and the `svmName` values for the `workingEnvironmentId` and the `svmName` parameters.

### 2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the `name` value of the aggregate for the `name` parameter.



If aggregate name does not exist and the `createAggregateIfNotFound` query parameter is set `true`, the create volume request is allowed if the named aggregate is not found.

### 3. Choose the size for the disk

Choose the size value for the `size:size` parameter. The `size:unit` must be one of the following: TB, GB, MB, KB, or Byte.

### 4. Select the region

Perform the workflow [Get regions](#) workflow and pick `ipCidrRange` value of the required region→ `subnets` for `exportPolicy`→`ips` value.



## 5. Create the quote

Perform the workflow [Create quote](#). This is a recommended step but is not mandatory.

## 6. Create the volume

HTTP method	Path
POST	/occm/api/gcp/vsa/volumes

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes' --header
'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>'
--header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

### Input

The JSON input example includes the minimum list of input parameters, including:

- <WORKING\_ENV\_ID> (workingEnvironmentId)
- <SVM\_NAME> (svmName)
- <AGGR\_NAME> (aggregateName)

If aggregate name does not exist, you can set the `createAggregateIfNotFound` query parameter to `true` which allows the aggregate not-found condition.

### JSON input example

```
{
  "workingEnvironmentId": "vsaworkingenvironment-sfrf3wvj",
  "svmName": "svm_zivgcp0lwe02",
  "aggregateName": "ziv0lagg01",
  "name": "zivagg01vol01",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0,
  "exportPolicyInfo": {
    "ips": [
      "10.138.0.0/20"
    ],
    "nfsVersion": [
      "nfs3", "nfs4"
    ],
    "policyType": "custom"
  }
}
```

## Output

None

## Create a volume using CIFS

You can use this workflow to create a volume accessed through CIFS.



If the properties `aggregateName` and `maxNumOfDisksApprovedToAdd` are not provided on the REST API call, the response will fail with a suggested name for the aggregate and the number of disks needed to fulfill the request.

### 1. Choose the CIFS configuration

A CIFS server configuration must be defined for your working environment. You can do one of the following:

- If a CIFS configuration already exists, perform the workflow [Get CIFS server configurations](#) to access the configuration parameters.
- If a CIFS configuration does not exist, perform the workflow [Create CIFS server configuration](#) to create one.

## 2. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the `workingEnvironmentId` (working environment) and the `svmName` (SVM name).

## 3. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the name for the `aggregateName` value.



If aggregate name does not exist and the `createAggregateIfNotFound` query parameter is set `true`, the create volume request is allowed if the named aggregate is not found.

## 4. Choose the size for the disk

Choose the size value for the `size:size` parameter. The `size:unit` must be one of the following: TB, GB, MB, KB, or Byte.

## 5. Create the quote

Perform the workflow [Create quote](#). This is a recommended step but is not mandatory.

## 6. Create the volume

HTTP method	Path
POST	/occm/api/gcp/vsa/volumes

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes' --header
'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>'
--header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

### Input

The JSON input example includes the minimum list of input parameters, including:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`) string
- `<SVM_NAME>` (`svmName`) string
- `<AGGR_NAME>` (`aggregateName`) string

If an aggregate name does not exist, you can set the `createAggregateIfNotFound` query parameter to `true` which allows the aggregate not-found condition.

### JSON input example

```
{
  "workingEnvironmentId": "VsaWorkingEnvironment-SfpVUZSc",
  "svmName": "svm_zivaws02we01",
  "aggregateName": "aggr1",
  "name": "zivaws02we02vol02Cifs",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "shareInfo": {
    "accessControl": {
      "permission": "full_control",
      "users": [
        "Everyone"
      ],
      "users": "Everyone;"
    },
    "shareName": "zivaws02we01vol02Cifs_share"
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0
}
```

## Output

None

## Create a volume using iSCSI

You can use this workflow to create a volume accessed through iSCSI. There are two workflows available depending on whether a new or existing iGroup is used. You need to select the correct workflow:

- [Create volume using iSCSI with a new iGroup](#)
- [Create volume using iSCSI with an existing iGroup](#)

### Create volume using iSCSI with a new iGroup



If the properties `aggregateName` and `maxNumOfDisksApprovedToAdd` are not provided on the REST API call, the response will fail with a suggested name for the aggregate and the number of disks needed to fulfill the request.

### 1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the `workingEnvironmentId` parameter and the `svmName` value for the `svmName` parameter.

### 2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the name for the `aggregateName` value.

### 3. Choose the size for the disk

Choose the size value for the `size:size` parameter. The `size:unit` must be one of the following: TB, GB, MB, KB, or Byte.

### 4. Choose the iscsiInfo parameters

You must choose the following values for the REST API call:

- A unique igroup name for `igroupCreationRequest` → `igroupName` parameter
- The required iqns to `igroupCreationRequest` → `initiators` parameter.
- The required operating system for the `osName` parameter from one of the following:
  - windows
  - linux
  - vmware
  - windows\_2008
  - windows\_gpt

### 5. Create the quote

Perform the workflow [Create quote](#). This is a recommended step but is not mandatory.

### 6. Create the volume

HTTP method	Path
POST	/occm/api/gcp/vsa/volumes

#### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes' --header
'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>'
--header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

#### Input

The JSON input example includes the minimum list of input parameters, including:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`) string

- <SVM\_NAME> (svmName) string
- <AGGR\_NAME> (aggregateName) string

If aggregate name does not exist, you can set the `createAggregateIfNotFound` query parameter to `true` which allows the aggregate not-found condition.

### JSON input example

```
{
  "workingEnvironmentId": "VsaWorkingEnvironment-SfpVUZSc",
  "svmName": "svm_zivaws02we01",
  "aggregateName": "aggr1",
  "name": "zivaws02we01vol01Iscsi",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "iscsiInfo": {
    "igroupCreationRequest": {
      "igroupName": "zivIgroup",
      "initiators": [
        "iqn.1994-05.com.redhat:96de86825216",
        "iqn.1994-05.com.redhat:96de86823426"
      ]
    },
    "osName": "linux"
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0
}
```

### Output

None

### Create volume using iSCSI with an existing iGroup



If the properties `aggregateName` and `maxNumOfDisksApprovedToAdd` are not provided on the REST API call, the response will fail with a suggested name for the aggregate and the number of disks needed to fulfill the request.

#### 1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the `workingEnvironmentId` parameter and the `svmName` value for the `svmName` parameter.

## 2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the name for the `aggregateName` value.

## 3. Choose the size for the disk

Choose the size value for the `size:size` parameter. The `size:unit` must be one of the following: TB, GB, MB, KB, or Byte.

## 4. Choose the iGroup

Perform the workflow [Get iGroups](#) and choose the `igroups` for the `iscsiInfo → igroups` value. Also select the `osType` value for the `iscsiInfo → osName`.

## 5. Create the quote

Perform the workflow [Create quote](#). This is a recommended step but is not mandatory.

## 6. Create the volume

HTTP method	Path
POST	/occm/api/gcp/vsa/volumes

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes' --header
'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>'
--header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

### Input

The JSON input example includes the minimum list of input parameters, including:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`) string
- `<SVM_NAME>` (`svmName`) string
- `<AGGR_NAME>` (`aggregateName`) string

If an aggregate name does not exist, you can set the `createAggregateIfNotFound` query parameter to `true` which allows the aggregate not-found condition.

### JSON input example

```
{
  "workingEnvironmentId": "VsaWorkingEnvironment-UvFmWXoD",
  "svmName": "svm_zivaws01we01",
  "aggregateName": "aggr1",
  "name": "zivaws01we01vol05Iscsi",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "iscsiInfo": {
    "igroups": ["zivIgroup1"],
    "osName": "linux"
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0
}
```

## Output

None

## Get volumes

You can retrieve the list of volumes of a single node GCP working environment.

### 1. Select the working environment

Perform the workflow [Get working environment](#) and choose the `publicId` value of the working environment for the `workingEnvironmentId` query parameter.

### 2. Get the volumes

HTTP method	Path
GET	/occm/api/gcp/vsa/volumes

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes?workingEnv
ironmentId=<WORKING_ENV_ID>' --header 'Content-Type: application/json'
--header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer
<ACCESS_TOKEN>'
```



## Input

Query parameter <WORKING\_ENV\_ID> (workingEnvironmentId) string

## Output

The JSON output example includes the list of volumes for the working environment.

## JSON output example

```
[
  {
    "name": "zivagg01vol01",
    "uuid": "0x00000-0xx0-00xx-xx00-00xxxx000",
    "svmName": "svm_zivgcp01we02",
    "size": {
      "size": 100.0,
      "unit": "GB"
    },
    "usedSize": {
      "size": 2.93731689453125E-4,
      "unit": "GB"
    },
    "junctionPath": "/zivagg01vol01",
    "volumeTotalInodes": 3112959,
    "volumeUsedInodes": 96,
    "mountPoint": "10.138.0.150:/zivagg01vol01",
    "compressionSpaceSaved": {
      "size": 0.0,
      "unit": "GB"
    },
    "deduplicationSpaceSaved": {
      "size": 0.0,
      "unit": "GB"
    },
    "thinProvisioning": true,
    "compression": true,
    "deduplication": true,
    "snapshotPolicy": "default",
    "securityStyle": "unix",
    "exportPolicyInfo": {
      "name": "export-svm_zivgcp01we02-zivagg01vol01",
      "policyType": "custom",
      "ips": [
        "10.138.0.0/20"
      ],
      "nfsVersion": [
        "nfs3",
        "nfs4"
      ]
    }
  }
]
```

```

    ],
    "shareNames": [],
    "shareInfo": [],
    "parentVolumeName": "",
    "rootVolume": false,
    "state": "online",
    "volumeType": "rw",
    "aggregateName": "ziv01agg01",
    "parentSnapshot": null,
    "autoSizeMode": "grow",
    "maxGrowSize": {
        "size": 1100.0,
        "unit": "GB"
    },
    "providerVolumeType": "pd-ssd",
    "cloneNames": [],
    "moving": false,
    "primaryNoFailoverMountPoint": null,
    "secondaryNoFailoverMountPoint": null,
    "capacityTier": null,
    "capacityTierUsedSize": null,
    "cifsShareAccessPoint": null,
    "primaryCifsShareAccessPoint": null,
    "secondaryCifsShareAccessPoint": null,
    "tieringPolicy": "none",
    "tierInactiveUserData": null,
    "tierInactiveUserDataPercent": null,
    "comment": null,
    "qosPolicyGroupName": null,
    "snaplockType": "non_snaplock",
    "constituentsAggregates": [],
    "snapshotsUsedSize": {
        "size": 0.0,
        "unit": "Byte"
    },
    "cbsBackupsInfo": null,
    "minimumCoolingDays": null,
    "targetName": "iqn.1992-
08.com.netapp:sn.986656ab5e3e11eb9cb735a0758d479a:vs.2",
    "iscsiEnabled": false,
    "isFlexGroupVolume": false
}
]

```

## Modify volume

You can modify the configuration of an existing volume.

### 1. Select the working environment

Perform the workflow [Get working environment](#) and choose the `publicId` and `svmName` values of the working environment for used as the `workingEnvironmentId` and `svmName` path parameters.

### 2. Select the volume

Perform the workflow [Get volumes](#) and choose the `name` for the `volumeName` path parameter.

### 3. Modify the volume

HTTP method	Path
PUT	/occm/api/gcp/vsa/volumes/{workingEnvironmentId}/{svmName}/{volumeName}

### curl example

```
curl --location --request PUT
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes/<WORKING_ENV_ID>/<SVM_NAME>/<VOLUME_NAME>' --header 'Content-Type: application/json'
--header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

### Input

Path parameters:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`) string
- `<SVM_NAME>` (`svmName`) string
- `<VOLUME_NAME>` (`volumeName`) string

The JSON input example includes the minimum list of input parameters.

### JSON input example

```
{
  "exportPolicyInfo": {
    "policyType": "custom",
    "ips": [
      "10.000.0.0/20"
    ],
    "nfsVersion": [
      "nfs3"
    ]
  }
}
```

## Output

None

## Delete volume

You can delete an existing volume.

### 1. Select the working environment

Perform the [Get working environment](#) workflow and choose the `publicId` and `svmName` values of the required working environment for `workingEnvironmentId` and `svmName` path parameters.

### 2. Select the volume

Perform the workflow [Get volumes](#) and choose the name for the `volumeName` path parameter.

### 3. Delete the volume

HTTP method	Path
DELETE	/occm/api/gcp/vsa/volumes/{workingEnvironmentId}/{svmName}/{volumeName}

## curl example

```
curl --location --request DELETE
'https://cloudmanager.cloud.netapp.com/occm/api/azure/vsa/volumes/<WORKING_ENV_ID>/<SVM_NAME>/<VOLUME_NAME>' --header 'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

## Input

Path parameters:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`) string
- `<SVM_NAME>` (`svmName`) string

- <VOLUME\_NAME> (volumeName) string

## Output

None

## Create quote

You can create a quote for a new volume which returns a resource quote needed to satisfy the request. The resource quote contains aggregate information where the volume will be created and confirms if the space is available. This is a recommended step but is not mandatory.

### 1. Select the working environment

Perform the workflow [Get working environment](#) and choose the `publicId` and `'svmName'` values of the required working environment for `workingEnvironmentId` and `svmName` parameters in the JSON input.

### 2. Select the aggregate

Perform the [Get aggregates](#) workflow and choose the `name` value of the required aggregate for the `aggregateName` parameter in the JSON input.

### 3. Select the gcp disk type

Perform the [Get gcp disk types](#) workflow and choose the `size` and `supportedDiskType` values of the required disk type for the `diskSize` and `providerVolumeType` parameters in the JSON input.

### 4. Generate the volume quote

HTTP method	Path
POST	/occm/api/gcp/vsa/quote

## curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes/quote'
--header 'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>'
--header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

## Input

The JSON input example includes the list of input parameters.

## JSON input example

```
{
  "workingEnvironmentId": "vsaworkingenvironment-sfrf3wvj",
  "svmName": "svm_zivgcp0lwe02",
  "aggregateName": "ziv0lagg01",
  "name": "zivagg01vol01",
  "size": {
    "size": "100",
    "unit": "GB"
  },
  "enableThinProvisioning": "true",
  "providerVolumeType": "pd-ssd",
  "verifyNameUniqueness": "true"
}
```

## Output

The JSON output example includes an example of the quote details.

## JSON output example

```
{
  "numOfDisks": 0,
  "diskSize": {
    "size": 100.0,
    "unit": "GB"
  },
  "aggregateName": "ziv0lagg01",
  "newAggregate": false,
  "autoVsaCapacityManagement": true
}
```

## Get iGroups

You can use this workflow to retrieve all the initiator groups (iGroups).

### 1. Create the working environment

Perform the workflow [Create GCP single node working environment](#) and choose the `publicId` and `svmName` values for the working environment `workingEnvironmentId` and `svmName` path parameters.

### 2. Get the CIFS configurations

HTTP method	Path
GET	/occm/api/gcp/vsa/volumes/igroups/{workingEnvironmentId}/{svmName}

## curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes/igroups/<WORKING_ENV_ID>/<SVM_NAME>' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>' --header 'Content-Type:
application/json'
```

## Input

- Path parameter <WORKING\_ENV\_ID> workingEnvironmentId string
- Path parameter <SVM\_NAME> svmName string

## Output

The JSON output example includes a list of iGroups.

## JSON output example

```
[
  {
    "igroupName": "zivIgroup1",
    "osType": "linux",
    "portsetName": "",
    "igroupType": "iscsi",
    "initiators": [
      "iqn.1994-05.com.redhat:1d9ac633937c"
    ]
  },
  {
    "igroupName": "zivIgroup2",
    "osType": "linux",
    "portsetName": "",
    "igroupType": "iscsi",
    "initiators": [
      "iqn.1994-05.com.redhat:96de86825216"
    ]
  }
]
```

# Metadata

## Get GCP regions

This workflow retrieves the GCP regions in which a Cloud Volumes ONTAP working environment might be created.

## 1. Get the list of regions

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/regions

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/regions'
--header 'Content-Type: application/json' --header 'x-agent-id:
<AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

### Input

None

### Output

The JSON output provides an example of a list of GCP regions.

### JSON output example

```
[
  {
    "displayName": "asia-east1",
    "name": "asia-east1",
    "zones": [
      {
        "name": "asia-east1-a"
      },
      {
        "name": "asia-east1-b"
      },
      {
        "name": "asia-east1-c"
      }
    ],
    "vpcs": [
      {
        "name": "default",
        "subnets": [
          {
            "ipCidrRange": "10.140.0.0/20",
            "name": "default",
            "path": "projects/occm-dev/regions/asia-
east1/subnetworks/default",
            "availableIps": 4090,
            "minimumRequiredIps": 6
          }
        ]
      }
    ]
  }
]
```



```

    ],
    "firewalls": [
      {
        "name": "allow-all",
        "vpc": "default"
      },
      {
        "name": "allow-ssh-netapp",
        "vpc": "default"
      }
    ]
  },
  {
    "name": "eli-vpc",
    "subnets": [
      {
        "ipCidrRange": "10.0.0.0/00",
        "name": "eli-subnet",
        "path": "projects/occm-dev/regions/asia-
east1/subnetworks/eli-subnet",
        "availableIps": 250,
        "minimumRequiredIps": 6
      }
    ],
    "firewalls": [
      {
        "name": "eli-vpc-allow-http",
        "vpc": "eli-vpc"
      },
      {
        "name": "eli-vpc-allow-https",
        "vpc": "eli-vpc"
      }
    ]
  },
],
},
{
  "displayName": "asia-northeast1",
  "name": "asia-northeast1",
  "zones": [
    {
      "name": "asia-northeast1-a"
    },
    {
      "name": "asia-northeast1-b"
    }
  ]
}

```

```

    },
    {
      "name": "asia-northeast1-c"
    }
  ],
  "vpcs": [
    {
      "name": "default",
      "subnets": [
        {
          "ipCidrRange": "00.000.0.0/20",
          "name": "default",
          "path": "projects/occm-dev/regions/asia-northeast1/subnetworks/default",
          "availableIps": 4090,
          "minimumRequiredIps": 6
        }
      ],
      "firewalls": [
        {
          "name": "allow-all",
          "vpc": "default"
        },
        {
          "name": "allow-ssh-netapp",
          "vpc": "default"
        }
      ]
    }
  ]
}
}

```

## Get GCP permutations

You can use the permutations endpoint to retrieve the Cloud Volumes ONTAP configuration information such as `ontapVersion`, `license`, `instanceType`, `region` and more. You can check the possible permutations that can potentially be provided for the GCP workflows while provisioning a Cloud Volumes ONTAP instance.

### 1. Get the permutations

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/permutations

## curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/permutati
ons?latest_only=true' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>' --header 'Content-Type:
application/json'
```

## Input

There are several **optional** query parameters you can use:

- region string
- version string
- license string
- machine\_type string
- latest\_only string

## Output

The JSON output example includes the list of Cloud Volumes ONTAP configurations.

## JSON output example

```
[
  {
    "ontapVersion": "ONTAP-9.9.0X4.T1.gcp",
    "license": {
      "type": "gcp-cot-explore-paygo",
      "name": "Cloud Volumes ONTAP Explore",
      "description": "Suitable for smaller capacity applications.
Supports up to 2 TB of underlying GCP storage.",
      "subName": "",
      "subDescription": "Support of tiering to object storage is not
included.",
      "capacity_limit": "2TB",
      "platformLicenseRequired": false,
      "default": false,
      "capacityLimit": {
        "size": 2.0,
        "unit": "TB"
      }
    },
    "instanceType": "custom-4-16384",
    "region": {
      "name": "asia east 1",
      "code": "asia-east1",
```

```

        "location": "Changhua County, Taiwan",
        "s3Region": null
    },
    "defaultInstance": false,
    "features": [
        "cpu: Intel Skylake"
    ],
    "upgradeableFrom": [
        "9.8",
        "9.9.0"
    ]
},
{
    "ontapVersion": "ONTAP-9.9.0X4.T1.gcp",
    "license": {
        "type": "gcp-cot-explore-paygo",
        "name": "Cloud Volumes ONTAP Explore",
        "description": "Suitable for smaller capacity applications.
Supports up to 2 TB of underlying GCP storage.",
        "subName": "",
        "subDescription": "Support of tiering to object storage is not
included.",
        "capacity_limit": "2TB",
        "platformLicenseRequired": false,
        "default": false,
        "capacityLimit": {
            "size": 2.0,
            "unit": "TB"
        }
    },
    "instanceType": "custom-4-16384",
    "region": {
        "name": "asia east 2",
        "code": "asia-east2",
        "location": "Hong Kong",
        "s3Region": null
    },
    "defaultInstance": false,
    "features": [
        "cpu: Intel Skylake"
    ],
    "upgradeableFrom": [
        "9.8",
        "9.9.0"
    ]
}

```

]

## Get tag keys

You can perform this workflow to retrieve all the labels in the specified project and zone.

### 1. Select the project

Perform the workflow [Get projects](#) and choose `projectId` value of the required project for `projectId` query parameter.

### 2. Select the region

Perform the [Get regions](#) workflow and choose the `zone: name` value of the required region for the `zone` query parameter.

### 3. Get the tag keys

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/tag-keys

### curl example

```
curl --location --request GET 'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/tag-keys?projectId=<PROJECT_ID>&zone=<ZONE>' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>' --header 'Content-Type: application/json'
```

### Input

Query parameters:

- `<PROJECT_ID>` `projectId` string
- `<ZONE>` `zone` string

### Output

The JSON output example includes the list of Azure storage account types.

### JSON output example

```
[
  {
    "key": "working-environment-id",
    "values": [
      "vsaworkingenvironment-sfrf3wvj",
      "vsaworkingenvironment-2qkd75xv"
    ]
  },
]
```

```

{
  "key": "count-down",
  "values": [
    "3",
    "0",
    "2"
  ]
},
{
  "key": "username",
  "values": [
    "administrator"
  ]
},
{
  "key": "keepme",
  "values": [
    "10"
  ]
},
{
  "key": "cloud-ontap-version",
  "values": [
    "9_9_0x4"
  ]
},
{
  "key": "cloud-ontap-dm",
  "values": [
    "zivgcp01we02-deployment",
    "zivgcp01we03-deployment"
  ]
},
{
  "key": "platform-serial-number",
  "values": [
    "000000300000000000009",
    "0000000000000000096011"
  ]
}
]
},
{
  "key": "netapp:cloud-compliance:cloudManager:ClientId",
  "values": [
    "sNwn2FzHxFrucwz8j1huxNIYI7aRNqTC"
  ]
}

```

```
]
    }
]
```

## Create Buckets

You can perform this workflow to create a new bucket. This is a GCP storage bucket (data container in GCP) required for data tiering and backup.

### 1. Create a bucket

HTTP method	Path
POST	/occm/api/gcp/vsa/metadata/create-bucket

#### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/create-
bucket' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer
<ACCESS_TOKEN>' --header 'Content-Type: application/json' --d JSONinput
```

#### JSON input example

```
{
  "projectId": "occm-dev",
  "bucketName": "zivgcpbucket02",
  "location": "us-west1",
  "storageClass": "standard"
}
```

#### Output

None

## Get buckets

You can perform this workflow to retrieve the S3 buckets. The S3 buckets are the GCP storage buckets (data containers in GCP) required for data tiering and backups.

### 1. Get the buckets

HTTP method	Path
GET	/occm/api/vsa/metadata/buckets

#### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/metadata/buckets'
--header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer
<ACCESS_TOKEN>' --header 'Content-Type: application/json'
```

### Input

None

### Output

The JSON output example includes the list of S3 buckets.

### JSON output example

```
[
  {
    "bucketName": "3.9.0.bins.08112020",
    "region": "us-east-1",
    "tags": {}
  },
  {
    "bucketName": "0000000000-awsmacietrail-dataevent",
    "region": "us-east-1",
    "tags": {}
  }
]
```

## Get GCP packages

You can perform this workflow to retrieve the pre-defined packages configuration.

### 1. Get the packages

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/packages

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/packages'
--header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer
<ACCESS_TOKEN>' --header 'Content-Type: application/json'
```

### Input

None



## Output

The JSON output example includes the list of GCP packages.

### JSON output example

```
[
  {
    "name": "gcp_poc",
    "displayName": "POC and small workloads",
    "description": "No description yet",
    "licenseType": "gcp-cot-explore-paygo",
    "instanceTypeMapping": [
      {
        "region": "default",
        "instanceType": "custom-4-16384"
      }
    ],
    "diskType": "pd-ssd",
    "diskSize": {
      "size": 100.0,
      "unit": "GB"
    },
    "capacityTier": null,
    "instanceTenancy": null,
    "writingSpeedState": "NORMAL"
  },
  {
    "name": "gcp_standard",
    "displayName": "Database and application data production
workloads",
    "description": "No description yet",
    "licenseType": "gcp-cot-standard-paygo",
    "instanceTypeMapping": [
      {
        "region": "default",
        "instanceType": "n1-standard-8"
      }
    ],
    "diskType": "pd-ssd",
    "diskSize": {
      "size": 100.0,
      "unit": "GB"
    },
    "capacityTier": "GCP",
    "instanceTenancy": null,
    "writingSpeedState": "NORMAL"
  },
]
```

```

{
  "name": "gcp_dr",
  "displayName": "Cost effective DR",
  "description": "No description yet",
  "licenseType": "gcp-cot-standard-paygo",
  "instanceTypeMapping": [
    {
      "region": "default",
      "instanceType": "n1-standard-8"
    }
  ],
  "diskType": "pd-standard",
  "diskSize": {
    "size": 100.0,
    "unit": "GB"
  },
  "capacityTier": "GCP",
  "instanceTenancy": null,
  "writingSpeedState": "NORMAL"
},
{
  "name": "gcp_fastest",
  "displayName": "Highest performance production workloads",
  "description": "No description yet",
  "licenseType": "gcp-cot-premium-paygo",
  "instanceTypeMapping": [
    {
      "region": "default",
      "instanceType": "n1-standard-32"
    }
  ],
  "diskType": "pd-ssd",
  "diskSize": {
    "size": 100.0,
    "unit": "GB"
  },
  "capacityTier": "GCP",
  "instanceTenancy": null,
  "writingSpeedState": "NORMAL"
}
]

```

## Get Snapshot policies

You can perform this workflow to retrieve the default snapshot policies available on the cluster.

## 1. Get the snapshot policies

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/default-snapshot-policies

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/default-
snapshot-policies' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>' --header 'Content-Type:
application/json'
```

### Input

None

### Output

The JSON output example includes the list of default snapshot policies available on the cluster.

### JSON output example

```
[
  {
    "name": "default",
    "schedules": [
      {
        "frequency": "hourly",
        "retention": 6
      },
      {
        "frequency": "daily",
        "retention": 2
      },
      {
        "frequency": "weekly",
        "retention": 2
      }
    ],
    "description": "Default policy with hourly, daily & weekly
schedules."
  },
  {
    "name": "none",
    "schedules": [],
    "description": "Policy for no automatic snapshots."
  }
]
```

## Get supported features

You can perform this workflow to retrieve and check the supported features while provisioning a Cloud Volumes ONTAP instance.

### 1. Select the permutations

Perform the workflow [Get permutations](#) and choose the `ontapVersion`, `license: type`, `instanceType`, and `region: code` values of the required permutations for the query parameters of supported features request.

### 2. Get the supported features

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/supported-features

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/supported-features?region=<REGION>&ontapVersion=<ONTAP_VERSION>&dataEncryptionType=<ENCRP_TYPE>&licenseType=<LICENSE_TYPE>&instanceType=<INST_TYPE>' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>' --header 'Content-Type: application/json'
```

## Input

Query parameters:

- <REGION> region string
- <ONTAP\_VERSION> ontapVersion string
- <LICENSE\_TYPE> licenseType string
- <INST\_TYPE> instanceType string
- <ENCRP\_TYPE> dataEncryptionType string

## JSON output example

```
{
  "wormSupportedVersion": true,
  "cbsSupportedVersion": true,
  "httpsStorageAccountSupportedVersion": false,
  "tieringWithServiceAccount": true
}
```

## Get supported capacity tiers

You can retrieve the supported capacity tiers for Google Cloud disk types.

### 1. Select the permutations

Perform the workflow [Get permutations](#) and choose the ontapVersion, license: type, instanceType, and region: code values of the required permutations for the query parameters of supported capacity tiers request.

### 2. Get the supported features

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/supported-capacity-tiers

## curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/supported
-capacity
-tiers?region=<REGION>&ontapVersion=<ONTAP_VERSION>&dataEncryptionType=<EN
CRP_TYPE>&licenseType=<LICENSE_TYPE>&instanceType=<INST_TYPE>' --header
'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
--header 'Content-Type: application/json'
```

## Input

Query parameters:

- <REGION> region string
- <ONTAP\_VERSION> ontapVersion string
- <LICENSE\_TYPE> licenseType string
- <INST\_TYPE> instanceType string
- <ENCRP\_TYPE> dataEncryptionType string

## Output

The output shows a list of supported capacity tiers for the GCP disk types.

## JSON output example

```

{
  "supportedCapacityTiersPerVolumeType": [
    {
      "volumeType": "pd-standard",
      "supportedCapacityTiers": [
        "cloudStorage"
      ],
      "availableTieringPolicies": [
        "none",
        "snapshot_only"
      ]
    },
    {
      "volumeType": "pd-ssd",
      "supportedCapacityTiers": [
        "cloudStorage"
      ],
      "availableTieringPolicies": [
        "none",
        "snapshot_only"
      ]
    }
  ],
  "capacityTiersDisableReasons": [
    "Cannot create capacity tiered volume on Cloud Volumes ONTAP  
Explore license"
  ],
  "compositeSupported": true,
  "forceCompositeVersion": false
}

```

## Get service accounts

This workflow retrieves a list of service accounts from the specified project.

### 1. Get the list of service accounts

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/service-accounts

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/service-accounts' --header 'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

## Input

None

## Output

The JSON output retrieves a list of service accounts from the project.

## JSON output example

```
{
  "accounts": [
    {
      "name": "projects/occm-dev/serviceAccounts/00000-compute@serviceaccount.com",
      "projectId": "occm-dev",
      "email": "00000-compute@serviceaccount.com",
      "displayName": "Compute Engine default service account",
      "isEnabled": false
    },
    {
      "name": "projects/occm-dev/serviceAccounts/xxx-000@occmaccount.com",
      "projectId": "occm-dev",
      "email": "xxxx-000@occmaccount.com",
      "displayName": "admin",
      "isEnabled": false
    }
  ]
}
```

## Get projects

This workflow retrieves the list of projects that the caller has permission on.

### 1. Get the list of regions

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/projects

## curl example



```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/projects'
--header 'Content-Type: application/json' --header 'x-agent-id:
<AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

**Input**

None

**Output**

The JSON output provides an example of a list of authorized projects.

**JSON output example**

```

{
  "projects": [
    {
      "isDefault": false,
      "projectNumber": "958377592668",
      "projectId": "occm-host",
      "lifecycleState": "ACTIVE",
      "name": "OCCM-host",
      "createTime": "2019-07-24T14:36:32.472Z",
      "parent": {
        "type": "folder",
        "id": "339830134733"
      },
      "subscriptionId": null
    },
    {
      "isDefault": false,
      "projectNumber": "844924364732",
      "projectId": "occm-slave",
      "lifecycleState": "ACTIVE",
      "name": "OCCM-slave",
      "createTime": "2019-07-24T14:36:32.405Z",
      "parent": {
        "type": "folder",
        "id": "339830134733"
      },
      "subscriptionId": null
    },
    {
      "isDefault": true,
      "projectNumber": "92083494653",
      "projectId": "occm-dev",
      "lifecycleState": "ACTIVE",
      "name": "OCCM-Dev",
      "createTime": "2018-05-24T17:23:50.505Z",
      "parent": {
        "type": "folder",
        "id": "339830134733"
      },
      "subscriptionId": "gcp-saasMpIntegrationProductId-
saasMpCustomerIdentifier6"
    }
  ]
}

```

## Get GCP encryption keys

This workflow retrieves the GCP encryption keys for a specific region.

### 1. Select the region

Perform the workflow [Get regions](#) and choose the `name` value of the required region for the `region` path parameter.

### 2. Select the project

Perform the [get projects](#) workflow and choose the `projectId` value of the required project for the `project` path parameter.

### 3. Get the encryption keys

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/gcp-encryption-keys

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/gcp-
encryption-keys?region=<REGION>&project=<PROJECT>' --header 'Content-Type:
application/json' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>'
```

### Input

Required query parameters:

- `<REGION>` `region`: string
- `<PROJECT>` `project`: string

### Output

The JSON output provides an example of a list of GCP encryption keys for a specific region.

### JSON output example

```
[
  {
    "name": "key1",
    "id": "projects/occm-dev/locations/global/keyRings/test/cryptoKeys/key1",
    "keyRing": "test",
    "location": "global"
  },
  {
    "name": "key2",
    "id": "projects/occm-dev/locations/global/keyRings/test/cryptoKeys/key2",
    "keyRing": "test",
    "location": "global"
  }
]
```

## Get GCP disk types

This workflow retrieves the GCP supported disk types.

### 1. Get the disk types

HTTP method	Path
GET	/occm/api/gcp/vsa/metadata/gcp-disk-types

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/gcp-disk-
types' --header 'Content-Type: application/json' --header 'x-agent-id:
<AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

### Input

None

### Output

The JSON output provides an example of a list of GCP supported disk types.

### JSON output example

```
[
  {
    "size": {
      "size": 100.0,
      "unit": "GB"
    },
    "supportedDiskTypes": [
      "pd-standard",
      "pd-ssd"
    ],
    "supportedOccmLicenses": [
      "Standard (BYOL)",
      "Cloud Volumes ONTAP Standard",
      "Cloud Volumes ONTAP Explore",
      "Standard (hourly)",
      "Cloud Volumes ONTAP BYOL",
      "Explore (hourly)",
      "Cloud Volumes ONTAP Premium"
    ]
  },
  {
    "size": {
      "size": 500.0,
      "unit": "GB"
    },
    "supportedDiskTypes": [
      "pd-standard",
      "pd-ssd"
    ],
    "supportedOccmLicenses": [
      "Standard (BYOL)",
      "Cloud Volumes ONTAP Standard",
      "Cloud Volumes ONTAP Explore",
      "Standard (hourly)",
      "Cloud Volumes ONTAP BYOL",
      "Explore (hourly)",
      "Cloud Volumes ONTAP Premium"
    ]
  }
]
```

## Get instance types not supporting acceleration and capacity tiering

This workflow retrieves the GCP instance types which do not support the acceleration and capacity tiering. This means that these kind instance types do not use any sort of

hardware accelerator. You can check the instance types while provisioning the Cloud Volumes ONTAP.

#### 1. Get the instance types

HTTP method	Path
GET	occm/api/gcp/vsa/metadata/instance-types-not-supporting-acceleration-and-capacity-tiering

#### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/metadata/instance-
types-not-supporting-acceleration-and-capacity-tiering' --header 'Content-
Type: application/json' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>'
```

#### Input

None

#### Output

The JSON output provides an example of a list of instance types.

#### JSON output example

```
{
  "instanceTypes": []
}
```

## Copyright Information

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.