



## **AWS workflows**

### Cloud Manager Automation

NetApp

October 07, 2021

# Table of Contents

- Workflows for Amazon Web Services ..... 1
  - Working environments ..... 1
  - Aggregates ..... 16
  - Volumes ..... 20
  - Metadata ..... 35
  - Miscellaneous ..... 43

# Workflows for Amazon Web Services

There are several workflows you can use with the Amazon Web Services public cloud.



Before using any of the Cloud Manager REST API workflows, make sure to first review [Getting started](#).

## Working environments

### Create a working environment with PAYGO licensing

You can use this workflow to create a new Cloud Volumes ONTAP working environment using pay as you go (PAYGO) licensing. A new volume is also created with the working environment.

#### Before you begin

You must have a **Connector** for the cloud environment before beginning the workflow. You can create a Connector using the Cloud Manager web user interface. When you create a Connector, Cloud Manager adds the cloud provider account that you deployed the Connector in to your list of available accounts. See the [Cloud Manager documentation](#) for more information.

Also note the following when using PAYGO licensing:

- 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.

#### 1. Select the region

Perform the workflow [Get regions](#) and choose the `code` value for the `region` parameter 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 permutations configuration

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

#### 4. Select the VPC

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

- Choose the `vpcId` value for the `vpcId` parameter in step 8.
- Choose the `cidrBlock` values for the `ips` value of the `volume` parameter in step 8.

## 5. Select the EBS volume configuration

Perform the workflow [Get EBS volume types](#) and choose the `size` and `supportedVolumeTypes` values for the `ebsVolumeSize` parameter in step 8.

## 6. Attach a marketplace subscription

Perform the workflow [Attach SaaS subscription](#).

## 7. Optionally obtain an NSS key

An NSS key is optional when using PAYGO licensing. 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 [Add NSS credentials key](#) and choose the `id`.
- To select an existing NSS key, perform the workflow [Get NSS keys](#) and choose the `id` of the required NSS user.

## 8. Create the working environment

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

### curl example

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

### Input

The JSON input example includes the minimum list of parameters.



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

### JSON input example

```

{
  "name": "ziv01we02",
  "svmPassword": "user_password",
  "vpcId": "vpc-b16c90d4",
  "region": "us-east-1",
  "tenantId": "tenantIDgoeshere",
  "subnetId": "subnet-f4da95ac",
  "dataEncryptionType": "AWS",
  "vsaMetadata": {
    "ontapVersion": "ONTAP-9.9.0.T1",
    "licenseType": "cot-explore-paygo",
    "instanceType": "m5.xlarge"
  },
  "ebsVolumeSize": {
    "size": 100,
    "unit": "GB"
  },
  "ebsVolumeType": "gp2",
  "volume": {
    "name": "ziv02vol01",
    "size": {
      "size": 200,
      "unit": "GB"
    }
  },
  "exportPolicyInfo": {
    "policyType": "custom",
    "ips": [
      "172.31.0.0/16"
    ],
    "nfsVersion": [
      "nfs3",
      "nfs4"
    ]
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true
},
"writingSpeedState": "NORMAL"
}

```

## Output

The JSON output example includes an example of the VsaWorkingEnvironmentResponse.

## JSON output example

```
{
  "publicId": "VsaWorkingEnvironment-0NWSblaX",
  "name": "ziv01we02",
  "tenantId": "tenantIDgoeshere",
  "svmName": "svm_ziv01we02",
  "creatorUserEmail": "user_email",
  "status": null,
  "awsProperties": null,
  "reservedSize": null,
  "encryptionProperties": null,
  "clusterProperties": null,
  "ontapClusterProperties": null,
  "actionsRequired": null,
  "interClusterLifs": null,
  "cronJobSchedules": null,
  "snapshotPolicies": null,
  "svms": null,
  "activeActions": null,
  "replicationProperties": null,
  "schedules": null,
  "cloudProviderName": "Amazon",
  "isHA": false,
  "workingEnvironmentType": "VSA",
  "supportRegistrationProperties": null,
  "supportRegistrationInformation": null,
  "haProperties": null,
  "capacityFeatures": null,
  "cloudSyncProperties": null,
  "supportedFeatures": null,
  "k8sProperties": null,
  "fpolicyProperties": null,
  "saasProperties": null,
  "cbsProperties": null,
  "complianceProperties": null,
  "monitoringProperties": null
}
```

## Create a working environment with BYOL licensing

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

## Before you begin

You must have a **Connector** for the cloud environment before beginning the workflow. You can create a Connector using the Cloud Manager web user interface. When you create a Connector, Cloud Manager adds the cloud provider account that you deployed the Connector in to your list of available accounts. See the [Cloud Manager documentation](#) for more information.

Also 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 a volume after creating the working environment.

To optionally create a new volume with the working environment, you must modify the JSON input provided on the REST API call. See [Create a working environment with PAYGO licensing](#) for an example.

### 1. Select the region

Perform the workflow [Get regions](#) and choose the `code` value for the `region` parameter in step 7.

### 2. Select the workspace

Perform the workflow [Get tenants](#) and choose the `workspacePublicId` value for the `tenantId` parameter in step 7.

### 3. Select the permutations configuration

Perform the workflow [Get permutations](#) and choose the `ontapVersion` and `license: type` and `instanceType` values for the `vsaMetadata` parameter in step 7.

You will also need to include the `platformSerialNumber` value in `vsaMetadata` parameter in the REST API call.

### 4. Select the VPC

Perform the workflow [Get VPCs](#) and choose the `vpcId` value for the `vpcId` parameter in step 7.

### 5. Select the EBS volume configuration

Perform the workflow [Get EBS volume types](#) and choose the `size` and `supportedVolumeTypes` values for the `ebsVolumeSize` parameter in step 7.

### 6. Obtain the required NSS key

An NSS key is required when using BYOL licensing. You can create a key or select an existing key, and include the NSS key in the `nssAccount` parameter in step 7.

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

## 7. Create the working environment

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

### curl example

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

### Input

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

### JSON input example

```
{
  "name": "username",
  "svmPassword": "password",
  "vpcId": "vpc-b16c90d4",
  "region": "us-east-1",
  "tenantId": "tenantIDgoeshere",
  "subnetId": "subnet-f4da95ac",
  "dataEncryptionType": "AWS",
  "nssAccount": "ab332ce3-aa49-4995-8e09-752a0601c682",
  "vsaMetadata": {
    "ontapVersion": "ONTAP-9.9.0.T1",
    "licenseType": "cot-premium-byol",
    "instanceType": "m5.xlarge",
    "platformSerialNumber": "90120130000000000026"
  },
  "ebsVolumeSize": {
    "size": 100,
    "unit": "GB"
  },
  "ebsVolumeType": "gp2"
}
```

### Output

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

### JSON output example



```

{
  "publicId": "VsaWorkingEnvironment-wL2MaBJs",
  "name": "username",
  "tenantId": "tenantIDshownhere",
  "svmName": "svm_ziv02we03",
  "creatorUserEmail": "user_email",
  "status": null,
  "awsProperties": null,
  "reservedSize": null,
  "encryptionProperties": null,
  "clusterProperties": null,
  "ontapClusterProperties": null,
  "actionsRequired": null,
  "interClusterLifs": null,
  "cronJobSchedules": null,
  "snapshotPolicies": null,
  "svms": null,
  "activeActions": null,
  "replicationProperties": null,
  "schedules": null,
  "cloudProviderName": "Amazon",
  "isHA": false,
  "workingEnvironmentType": "VSA",
  "supportRegistrationProperties": null,
  "supportRegistrationInformation": null,
  "haProperties": null,
  "capacityFeatures": null,
  "cloudSyncProperties": null,
  "supportedFeatures": null,
  "k8sProperties": null,
  "fpolicyProperties": null,
  "saasProperties": null,
  "cbsProperties": null,
  "complianceProperties": null,
  "monitoringProperties": null
}

```

## Get working environments

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

## 1. Get the list of working environments

HTTP method	Path
GET	occm/api/vsa/working-environments

### curl

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

### Input

Optional query parameters:

- fields
- tenantId

### Output

The JSON output example includes an array of VSA working environments.

### JSON output example

```
[
  {
    "publicId": "VsaWorkingEnvironment-79VKenHW",
    "name": "ziv01we02",
    "tenantId": "tenantIDshownhere",
    "svmName": "svm_ziv01we02",
    "creatorUserEmail": "user_email",
    "status": null,
    "awsProperties": null,
    "reservedSize": null,
    "encryptionProperties": null,
    "clusterProperties": null,
    "ontapClusterProperties": null,
    "actionsRequired": null,
    "interClusterLifs": null,
    "cronJobSchedules": null,
    "snapshotPolicies": null,
    "svms": null,
    "activeActions": null,
    "replicationProperties": null,
    "schedules": null,
    "cloudProviderName": "Amazon",
    "isHA": false,
    "workingEnvironmentType": "VSA",
```

```

    "supportRegistrationProperties": null,
    "supportRegistrationInformation": [],
    "haProperties": null,
    "capacityFeatures": null,
    "cloudSyncProperties": null,
    "supportedFeatures": null,
    "k8sProperties": null,
    "fpolicyProperties": null,
    "saasProperties": null,
    "cbsProperties": null,
    "complianceProperties": null,
    "monitoringProperties": null
  },
  {
    "publicId": "VsaWorkingEnvironment-61kN4p5P",
    "name": "ziv01we03",
    "tenantId": "tenantIDshownhere",
    "svmName": "svm_ziv01we03",
    "creatorUserEmail": "user_email",
    "status": null,
    "awsProperties": null,
    "reservedSize": null,
    "encryptionProperties": null,
    "clusterProperties": null,
    "ontapClusterProperties": null,
    "actionsRequired": null,
    "interClusterLifs": null,
    "cronJobSchedules": null,
    "snapshotPolicies": null,
    "svms": null,
    "activeActions": null,
    "replicationProperties": null,
    "schedules": null,
    "cloudProviderName": "Amazon",
    "isHA": false,
    "workingEnvironmentType": "VSA",
    "supportRegistrationProperties": null,
    "supportRegistrationInformation": [],
    "haProperties": null,
    "capacityFeatures": null,
    "cloudSyncProperties": null,
    "supportedFeatures": null,
    "k8sProperties": null,
    "fpolicyProperties": null,
    "saasProperties": null,
    "cbsProperties": null,
  }
}

```

```

    "complianceProperties": null,
    "monitoringProperties": null
  },
  {
    "publicId": "VsaWorkingEnvironment-E9WanX81",
    "name": "ziv01we04",
    "tenantId": "tenantIDshownhere",
    "svmName": "svm_ziv01we04",
    "creatorUserEmail": "user_email",
    "status": null,
    "awsProperties": null,
    "reservedSize": null,
    "encryptionProperties": null,
    "clusterProperties": null,
    "ontapClusterProperties": null,
    "actionsRequired": null,
    "interClusterLifs": null,
    "cronJobSchedules": null,
    "snapshotPolicies": null,
    "svms": null,
    "activeActions": null,
    "replicationProperties": null,
    "schedules": null,
    "cloudProviderName": "Amazon",
    "isHA": false,
    "workingEnvironmentType": "VSA",
    "supportRegistrationProperties": null,
    "supportRegistrationInformation": [],
    "haProperties": null,
    "capacityFeatures": null,
    "cloudSyncProperties": null,
    "supportedFeatures": null,
    "k8sProperties": null,
    "fpolicyProperties": null,
    "saasProperties": null,
    "cbsProperties": null,
    "complianceProperties": null,
    "monitoringProperties": null
  }
]

```

## Delete a working environment

You can delete an existing Cloud Volumes ONTAP working environment.

## 1. Select the working environment to use

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

## 2. Delete the working environment

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

### Curl example

```
curl --location --request DELETE
'https://cloudmanager.cloud.netapp.com/occm/api/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`)

Optional query parameters:

- `localDelete`

If `true` the Cloud Volumes ONTAP instance in the cloud is not terminated, but Cloud Manager no longer manages it (default is `false`).

- `forceDelete`

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 [link](#) 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/vsa/working-environments/{workingEnvironmentId}/cifs-workgroup

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/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 `workingEnvironmentId`

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

## 3. Create the CIFS configuration

Create the CIFS server configuration.

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

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/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 `workingEnvironmentId`

### 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/vsa/working-environments/{workingEnvironmentId}/cifs

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/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 `workingEnvironmentId`
- Optional query parameter `svm`

## 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/vsa/working-environments/{workingEnvironmentId}/delete-cifs

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/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 `workingEnvironmentId`
- Optional JSON body

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

## Output

None.

# Aggregates

## Get aggregates

You can retrieve a list of available disk aggregates.

### 1. Select the working environment to use

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

### 2. Get the list of aggregates

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

### curl example

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

## Input

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

## Output

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

### 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_ziv01we01_root",
      "totalSize": {
        "size": 1.0,
        "unit": "GB"
      },
      "usedSize": {
        "size": 0.00115203857421875,
        "unit": "GB"
      },
      "thinProvisioned": false,
      "isClone": false,
      "rootVolume": true
    }
  ],
  "providerVolumes": [
    {
      "id": "vol-066fea889cbc6a65c",
      "name": "vol-066fea889cbc6a65c",
      "size": {
        "size": 100.0,
        "unit": "GB"
      },
      "state": "in-use",
      "device": "/dev/xvdg",
      "instanceId": "i-0fa9a2879e67a8829",
      "diskType": "gp2",
      "encrypted": true,
      "iops": null
    }
  ],
  "disks": [
    {
      "name": "NET-1.3",
      "position": "data",

```

```

        "ownerNode": "ziv01we01-01",
        "device": "xvdg vol066fea889cbc6a65c",
        "vmDiskProperties": null
    }
],
"state": "online",
"encryptionType": "cloudEncrypted",
"encryptionKeyId": null,
"isRoot": false,
"homeNode": "ziv01we01-01",
"ownerNode": "ziv01we01-01",
"capacityTier": null,
"capacityTierUsed": null,
"sidlEnabled": true,
"snaplockType": "non_snaplock"
}
]

```

## Create aggregate

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

### 1. Select the working environment to use

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

### 2. Create the aggregate

HTTP method	Path
POST	occm/api/vsa/aggregates

### curl example

```

curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/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": "ziv01agg04",
  "workingEnvironmentId": "VsaWorkingEnvironment-9e6p8LuF",
  "numberOfDisks": 1,
  "diskSize": {
    "size": 100,
    "unit": "GB"
  },
  "providerVolumeType": "gp2"
}
```

## Output

None

## Add disks to aggregate

You can add disks to an existing aggregate.

### 1. Select the working environment to use

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

### 2. Select the aggregate to delete

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

### 3. Add the disks

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

## curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/aggregates/<WORKING_EN
V_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`)
- `<AGGR_NAME>` (`aggregateName`)

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.

### 1. Select the working environment to use

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

### 2. Select the aggregate to delete

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

### 3. Delete the aggregate

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

### curl example

```
curl --location --request DELETE
'https://cloudmanager.cloud.netapp.com/occm/api/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 parameter `<WORKING_ENV_ID>` (`workingEnvironmentId`)

Path parameter `<AGGR_NAME>` (`aggregateName`)

### Output

None

## Volumes

### Create volume using NFS

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



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` (working environment) and the `svmName` (SVM name).

### 2. Select the aggregate

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



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. Select the virtual private cloud

Perform the workflow [Get virtual private clouds](#) and choose the `cidrBlock` value of the required VPC for the `ips` parameter or fill in the desired `exportPolicyInfo` value manually.

### 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/vsa/volumes

### curl example

```
curl --location --request POST
'https://staging.cloudmanager.cloud.netapp.com/occm/api/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-5dZfyKS5",
  "svmName": "svm_ziv01we01",
  "aggregateName": "ziv01agg01",
  "name": "ziv01vol04",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "exportPolicyInfo": {
    "policyType": "custom",
    "ips": [
      "172.31.0.0/16"
    ],
    "nfsVersion": [
      "nfs3",
      "nfs4"
    ]
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0
}
```

### Output

None

## Create 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/vsa/volumes

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/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 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 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 `iqn`'s 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/vsa/volumes

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/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-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 `iscasiInfo → igroups` value. Also select the `osType` value for the `iscasiInfo → 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/vsa/volumes

### curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/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 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.

### 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 volumes

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

### curl example

```

curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/volumes?workingEnvironmentId=<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)

## Output

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

## JSON output example

```
[
  {
    "name": "ziv02vol01",
    "uuid": "cb488216-5bd1-11eb-8a9b-615eb82c79d8",
    "svmName": "svm_ziv01we02",
    "size": {
      "size": 100.0,
      "unit": "GB"
    },
    "usedSize": {
      "size": 5.18798828125E-4,
      "unit": "GB"
    },
    "junctionPath": "/ziv02vol01",
    "volumeTotalInodes": 3112959,
    "volumeUsedInodes": 96,
    "mountPoint": "172.31.1.199:/ziv02vol01",
    "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_ziv01we02-ziv02vol01",
      "policyType": "custom",
      "ips": [
        "172.31.0.0/16"
      ],
      "nfsVersion": [
        "nfs3"
      ]
    }
  }
]
```

```

    },
    "shareNames": [],
    "shareInfo": [],
    "parentVolumeName": "",
    "rootVolume": false,
    "state": "online",
    "volumeType": "rw",
    "aggregateName": "aggr1",
    "parentSnapshot": null,
    "autoSizeMode": "grow",
    "maxGrowSize": {
        "size": 1100.0,
        "unit": "GB"
    },
    "providerVolumeType": "gp2",
    "cloneNames": [],
    "moving": false,
    "primaryNoFailoverMountPoint": null,
    "secondaryNoFailoverMountPoint": null,
    "capacityTier": null,
    "capacityTierUsedSize": null,
    "cifsShareAccessPoint": null,
    "primaryCifsShareAccessPoint": null,
    "secondaryCifsShareAccessPoint": null,
    "tieringPolicy": "none",
    "tierInactiveUserData": {
        "size": 0.0,
        "unit": "GB"
    },
    "tierInactiveUserDataPercent": 0,
    "comment": null,
    "qosPolicyGroupName": null,
    "snaplockType": "non_snaplock",
    "constituentsAggregates": [],
    "snapshotsUsedSize": {
        "size": 1597440.0,
        "unit": "Byte"
    },
    "cbsBackupsInfo": null,
    "minimumCoolingDays": null,
    "targetName": "iqn.1992-
08.com.netapp:sn.7d147b755bd011ebb076ef46475a0933: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 environments](#) and choose the `publicId` value for the working environment and `svmName`.

### 2. Select the volume

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

### 3. Modify the volume

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

### curl example

```
curl --location --request PUT
'https://cloudmanager.cloud.netapp.com/occm/api/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`)
- <SVM\_NAME> (`svmName`)
- <VOLUME\_NAME> (`volumeName`)

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

### JSON input example

```

{
  "exportPolicyInfo": {
    "policyType": "custom",
    "ips": [
      "172.31.0.0/16"
    ],
    "nfsVersion": [
      "nfs3"
    ]
  }
}

```

## Output

None

## Delete volume

You can delete an existing volume.

### 1. Select the working environment

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

### 2. Select the volume

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

### 3. Delete the volume

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

## curl example

```

curl --location --request DELETE
'https://cloudmanager.cloud.netapp.com/occm/api/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`)
- `<SVM_NAME>` (`svmName`)

- <VOLUME\_NAME> (volumeName)

## 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 environments](#) and choose the `publicId` value for the working environment and `svmName`.

### 2. Select the aggregate

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

### 3. Select the permutations configuration

Perform the workflow [Get permutations](#) and choose the `size` and `diskType` values for step 4.

### 4. Generate the volume quote

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

## curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/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-5dZfyKS5",
  "svmName": "svm_ziv01we01",
  "aggregateName": "ziv01agg01",
  "name": "ziv1vol02",
  "size": {
    "size": "100",
    "unit": "GB"
  },
  "enableThinProvisioning": "true",
  "providerVolumeType": "gp2",
  "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": "ziv01agg01",
  "newAggregate": false,
  "autoVsaCapacityManagement": true
}

```

## Get iGroups

You can use this workflow to retrieve all the iGroups.

### 1. Select the working environment

Perform the workflow [Get working environments](#) 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/vsa/volumes/igroups/{workingEnvironmentId}/{svmName}

## curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/volumes/igroups/<WORKI
NG_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>
- Path parameter <SVM\_NAME>

## 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 regions

This workflow retrieves the AWS regions in which an Cloud Volumes ONTAP working environment may be created.

## 1. Get the list of regions

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

### Curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/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 AWS regions.

### JSON output example

```
[
  {
    "name": "US East",
    "code": "us-east-1",
    "location": "N. Virginia",
    "s3Region": "US"
  },
  {
    "name": "US West",
    "code": "us-west-1",
    "location": "N. California",
    "s3Region": "us-west-1"
  },
  {
    "name": "EU",
    "code": "eu-central-1",
    "location": "Frankfurt",
    "s3Region": "eu-central-1"
  },
  {
    "name": "Asia Pacific",
    "code": "ap-south-1",
    "location": "Mumbai",
    "s3Region": "ap-south-1"
  },
]
```

## Get permutations

You can use the permutations endpoint to retrieve the Cloud Volumes ONTAP configuration information.

### 1. Get the permutations

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

### curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/metadata/permutations'
--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
- version
- license
- instance\_type
- default\_instance\_type
- feature
- latest\_only
- ami

### Output

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

### JSON output example

```
[
  {
    "ontapVersion": "ONTAP-9.9.0X4.T1",
    "license": {
      "type": "cot-explore-paygo",
      "name": "Cloud Volumes ONTAP Explore",
      "description": "Suitable for smaller capacity applications. Supports up to 2 TB of underlying AWS 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": "m5.xlarge",
    "region": {
      "name": "EU",
      "code": "eu-central-1",
      "location": "Frankfurt",
      "s3Region": "eu-central-1"
    },
    "defaultInstance": true,
    "features": ["ena", "kvm", "network-utilization"],
    "upgradeableFrom": ["9.8", "9.9.0"]
  }
]
```

## Get list of virtual private clouds

You can use this workflow to retrieve a list of the available virtual private clouds.

### 1. Select the region to use

Perform the workflow [Get regions](#) and choose the `code` value for the region.

### 2. Get the VPCs

HTTP method	Path
GET	/occm/api/vsa/metadata/vpcs?region=us-east-1

### Curl example



```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/metadata/vpcs?region=u
s-east-1' --header 'Content-Type: application/json' --header 'x-agent-id:
<AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

## Input

Query parameters:

- region

## Output

The JSON output example includes the list of virtual private clouds.

### JSON output example

```
[
  {
    "vpcId": "vpc-b16c90d4",
    "state": "available",
    "cidrBlock": "172.31.0.0/16",
    "tags": [
      {
        "key": "Name",
        "value": "VPC for VSA"
      },
      {
        "key": "last",
        "value": "ioio"
      }
    ],
    "default": true,
    "subnets": [
      {
        "subnetId": "subnet-cld99699",
        "cidr": "172.31.5.0/24",
        "subnetName": "subnet5",
        "availabilityZone": "us-east-1a",
        "availableIps": 247,
        "minimumRequiredIps": 8,
        "outpostArn": null
      },
      {
        "subnetId": "subnet-deebdbe3",
        "cidr": "172.31.6.0/24",
        "subnetName": "Proxy Subnet",
        "availabilityZone": "us-east-1e",

```

```

        "availableIps": 248,
        "minimumRequiredIps": 8,
        "outpostArn": null
    }
],
"securityGroups": [
    {
        "securityGroupId": "sg-000a2be93c8668b1b",
        "description": "NetApp OCCM Instance External Security
Group",
        "name": "lilush2000OCCM1590415972561-OCCMSecurityGroup-
JDB72N6W90UG"
    },
    {
        "securityGroupId": "sg-0017cf49544fbb31e",
        "description": "Enable HTTP and NFS for Cloud Restore
Instance",
        "name": "Cloud-Restore-Instance-account-TgPvchSk-
CloudRestoreInstanceSecurityGroup-ZH4GN8E8AU1K"
    }
],
"tenancy": "default"
},
{
    "vpcId": "vpc-fe5c1f98",
    "state": "available",
    "cidrBlock": "140.30.0.0/16",
    "tags": [
        {
            "key": "Name",
            "value": "VPCWithNOS3"
        }
    ],
    "default": false,
    "subnets": [
        {
            "subnetId": "subnet-e84722d4",
            "cidr": "140.30.5.0/24",
            "subnetName": "subnet1",
            "availabilityZone": "us-east-1e",
            "availableIps": 250,
            "minimumRequiredIps": 8,
            "outpostArn": null
        }
    ],
    "securityGroups": [

```

```

    {
      "securityGroupId": "sg-0ccb5855f56acd419",
      "description": "ONTAP Cloud firewall rules for management
and data interface",
      "name": "SG-Version10-duahpJbS-
NetAppExternalSecurityGroup-EB0BZTI536LE"
    },
    {
      "securityGroupId": "sg-1911fa66",
      "description": "default VPC security group",
      "name": "default"
    }
  ],
  "tenancy": "default"
}
]

```

## Get EBS volume types

You can use this workflow to retrieve the supported EBS volume types.

### 1. Get the EBS volume types

HTTP method	Path
GET	/occm/api/vsa/metadata/ebs-volume-types

### curl example

```

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

```

### Input

None

### Output

The JSON output example includes the list of EBS volume types.

### JSON output example

```

[
  {
    "description": "",
    "size": {
      "size": 100.0,

```

```

        "unit": "GB"
    },
    "supportedVolumeTypes": [
        "standard",
        "io1",
        "io2",
        "gp2"
    ],
    "supportedOccmLicenses": [
        "Explore (hourly)",
        "Standard (hourly)",
        "Standard (BYOL)",
        "Cloud Volumes ONTAP Explore",
        "Cloud Volumes ONTAP Standard",
        "Cloud Volumes ONTAP Premium",
        "Cloud Volumes ONTAP BYOL",
        "Cloud Volumes ONTAP Explore",
        "Cloud Volumes ONTAP Standard",
        "Cloud Volumes ONTAP Premium",
        "Cloud Volumes ONTAP BYOL"
    ],
    "isDefault": false
},
{
    "description": "",
    "size": {
        "size": 500.0,
        "unit": "GB"
    },
    "supportedVolumeTypes": [
        "standard",
        "io1",
        "io2",
        "gp2",
        "st1"
    ],
    "supportedOccmLicenses": [
        "Explore (hourly)",
        "Standard (hourly)",
        "Standard (BYOL)",
        "Cloud Volumes ONTAP Explore",
        "Cloud Volumes ONTAP Standard",
        "Cloud Volumes ONTAP Premium",
        "Cloud Volumes ONTAP BYOL",
        "Cloud Volumes ONTAP Explore",
        "Cloud Volumes ONTAP Standard",

```

```
        "Cloud Volumes ONTAP Premium",
        "Cloud Volumes ONTAP BYOL"
    ],
    "isDefault": false
}
]
```

## Miscellaneous

### Create AWS cloud provider account

You can use this workflow to create an AWS cloud provider account.

#### Before you begin

You must have AWS credentials.

#### 1. Get the SaaS marketplace account

Perform the workflow [Get SaaS marketplace account](#) and choose the `id` value for the required subscription for `subscriptionId` parameter.

#### 2. Create the account

HTTP method	Path
POST	/occm/api/accounts/aws

#### curl example

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

#### Input

The JSON input example includes the list of parameters.

#### JSON input example

```
{
  "accountName": "zivAccountTest2",
  "providerKeys": {
    "awsAccessKeys": {
      "accessKey": "accesskeystring",
      "secretKey": "secretkeystring"
    }
  },
  "subscriptionId": "subscriptionIDgoeshere"
}
```

## Output

The JSON output provides an example of the cloud provider details.

## JSON output example

```
{
  "publicId": "CloudProviderAccount-LCwgVOy7",
  "accountName": "zivAccountTest2",
  "accountType": "AWS_KEYS",
  "accountId": "accountIDshownhere",
  "accessKey": " accesskeyshownhere",
  "assumeRole": null,
  "occmRole": null,
  "vsaList": [],
  "subscriptionId": "subscriptionIDshownhere"
}
```

## Copyright Information

Copyright © 2021 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.