

Dedicated Distributed Storage Service

API Reference

Issue	01
Date	2026-07-21



Copyright © Huawei Technologies Co., Ltd. 2026. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

- 1 Before You Start..... 1**
- 2 API Overview..... 4**
- 3 Calling APIs..... 5**
 - 3.1 Making an API Request..... 5
 - 3.2 Authentication..... 9
 - 3.3 Response..... 10
- 4 APIs..... 13**
 - 4.1 DSS Management..... 13
 - 4.1.1 Obtaining Details of a DSS Storage Pool..... 13
 - 4.1.2 Obtaining Details of DSS Storage Pools..... 18
 - 4.1.3 Updating the Name of a DSS Storage Pool..... 24
 - 4.1.4 Obtaining Details of Versions..... 26
 - 4.1.5 Obtaining Details of a Version..... 31
- 5 Permissions and Supported Actions..... 36**
 - 5.1 Introduction..... 36
 - 5.2 Actions Supported by Identity Policy-based Authorization..... 37
 - 5.3 Policy-based Authorization..... 40
- A Appendixes..... 42**
 - A.1 DSS Storage Pool Status..... 42
 - A.2 Error Codes..... 42
 - A.3 Status Codes..... 44
 - A.4 Obtaining a Project ID..... 45

1 Before You Start

Overview

Welcome to *Dedicated Distributed Storage Service API Reference*. Dedicated Distributed Storage Service (DSS) provides you with dedicated storage pools which are physically isolated from other pools to ensure high security. With data redundancy and cache acceleration techniques, DSS provides highly reliable, durable, stable low-latency storage resources. By flexibly interconnecting with various compute services, such as Dedicated Computing Cluster (DCC), Elastic Cloud Server (ECS), and Bare Metal Server (BMS), DSS is perfect for high-performance computing (HPC), online analytical processing (OLAP), and mixed loads.

This document describes how to use application programming interfaces (APIs) to perform operations on DSS resources, such as creating, querying, deleting, and updating DSS resources. For details about all supported operations, see [API Overview](#).

If you plan to access DSS through an API, ensure that you are familiar with DSS concepts. For details, see [Service Overview](#).

After the storage pool is deployed and becomes available, you need to create disks in the storage pool. For details about disk APIs, see [Elastic Volume Service API Reference](#).

API Calling

DSS supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see [Calling APIs](#).

Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoint of the DSS service, see [Regions and Endpoints](#).

Concepts

- Account

An account is created upon successful signing up. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity, which should not be used directly to perform routine management. To ensure account security, create Identity and Access Management (IAM) users and grant them permissions for routine management.

- User

An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).

API authentication requires information such as the account name, username, and password.

- Region

Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

For details, see [Region and AZ](#).

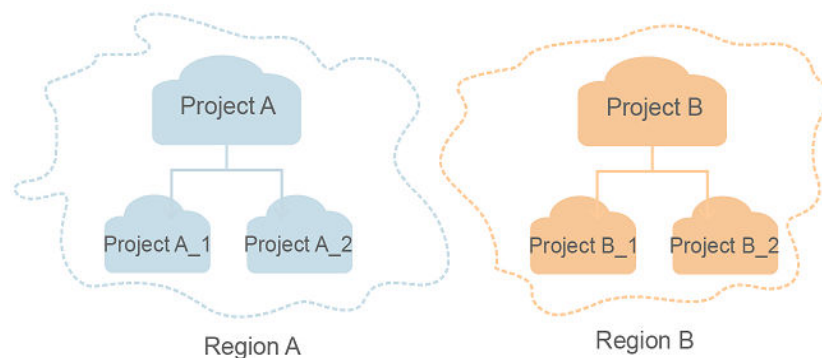
- AZ

An AZ contains one or more physical data centers with independent power, cooling, fire suppression, and water systems. Within an AZ, compute, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

- Project

A project corresponds to a region. Default projects are defined to group and physically isolate resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources under their accounts in the region associated with the project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



- Enterprise project

Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.

For details about enterprise projects and about how to obtain enterprise project IDs, see [*Enterprise Management User Guide*](#).

2 API Overview

[Table 2-1](#) provides an overview of the DSS APIs.

Table 2-1 API overview

API	Description
Obtaining Details of a DSS Storage Pool	Obtain the details of a specified DSS storage pool, including the pool name, ID, capacity, type, and creation time.
Obtaining Details of DSS Storage Pools	Obtain the DSS storage pools requested by a tenant, including the pool names, IDs, capacities, types, and creation time. Filter query and pagination query are supported.

For details about disk APIs, see [Elastic Volume Service API Reference](#).

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for [creating an IAM user](#) as an example to demonstrate how to call an API.

Request URI

A request URI is in the following format:

{URI-scheme}://{Endpoint}/{resource-path}?{query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

Table 3-1 URI parameter description

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints . For example, the endpoint of IAM in region CN-Hong Kong is iam.ap-southeast-1.myhuaweicloud.com .
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the resource-path of the API used to obtain a user token is /v3/auth/tokens .
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of <i>Parameter name=Parameter value</i> . For example, ?limit=10 indicates that a maximum of 10 data records will be displayed.

IAM is a global service. You can create an IAM user using the endpoint of IAM in any region. For example, to create an IAM user in the **CN-Hong Kong** region, obtain the endpoint of IAM (**iam.ap-southeast-1.myhuaweicloud.com**) for this region and the **resource-path** (**/v3.0/OS-USER/users**) in the URI of the API for **creating an IAM user**. Then construct the URI as follows:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

Figure 3-1 Example URI



NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server.

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API for **creating an IAM user**, the request method is **POST**. An example request is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows.

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of <i>Hostname:Port number</i> . If the port number is not specified, the default port is used. The default port number for https is 443 .	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value application/json is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in Obtaining a Project ID .	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4

Parameter	Description	Mandatory	Example Value
X-Auth-Token	Specifies the user token. It is a response to the API for obtaining a user token (This is the only API that does not require authentication). After the request is processed, the value of X-Subject-Token in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZlhvc NAQcCo...ggg1B BIINPXsidG9rZ

 NOTE

In addition to supporting authentication using tokens, APIs support authentication using AK/SK, which uses SDKs to sign a request. During the signature, the **Authorization** (signature authentication) and **X-Sdk-Date** (time when a request is sent) headers are automatically added in the request.

For more details, see "Authentication Using AK/SK" in [Authentication](#).

The following shows an example request of the API for [creating an IAM user](#) when AK/SK authentication is used:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

(Optional) Request Body

This part is optional. A request body is generally sent in a structured format (for example, JSON or XML), which is specified by **Content-Type** in the request header. It is used to transfer content other than the request header. If the request body contains full-width characters, these characters must be coded in UTF-8.

The request body varies depending on APIs. Certain APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

The following shows an example request (a request body included) of the API for [creating an IAM user](#). You can learn about request parameters and related description from this example. The bold parameters need to be replaced for a real request.

- **accountid**: account ID of an IAM user
- **username**: name of an IAM user
- **email**: email of an IAM user
- *********: login password of an IAM user

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
```

```
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****

{
  "user": {
    "domain_id": "accountid",
    "name": "username",
    "password": "*****",
    "email": "email",
    "description": "IAM User Description"
  }
}
```

If all data required for the API request is available, you can send the request to call the API through [curl](#), [Postman](#), or coding.

3.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token authentication: Requests are authenticated using tokens.
- AK/SK authentication: Requests are encrypted using AK/SK pairs. AK/SK authentication is recommended because it is more secure than token authentication.

Token Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to the request header to get permissions for calling the API. You can obtain a token by calling the [Obtaining a User Token](#) API.

DSS is a project-level service. When you call the API, set **auth.scope** in the request body to **project**.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username", // IAM user name
          "password": "*****", // IAM user password
          "domain": {
            "name": "domainname" // Name of the account to which the IAM user belongs
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx" // Project Name
      }
    }
  }
}
```

```
}  
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects  
Content-Type: application/json  
X-Auth-Token: ABCDEFJ....
```

AK/SK Authentication

NOTE

AK/SK authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token authentication is recommended.

In AK/SK authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key, which is used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK authentication, you can use an AK/SK to sign requests based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see [API Request Signing Guide](#).

NOTE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

3.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see [Error Codes](#).

For example, if status code **201** is returned for calling the API used to [create an IAM user](#), the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

[Figure 3-2](#) shows the response header for the API used to [create an IAM user](#).

 NOTE

For security purposes, you are advised to set the token in ciphertext in configuration files or environment variables and decrypt it when using it.

Figure 3-2 Header fields of the response to the request for creating an IAM user

```
"X-Frame-Options": "SAMEORIGIN",  
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fceac8a",  
"Transfer-Encoding": "chunked",  
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",  
"Server": "api-gateway",  
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",  
"X-Content-Type-Options": "nosniff",  
"Connection": "keep-alive",  
"X-Download-Options": "noopen",  
"X-XSS-Protection": "1; mode=block;",  
"X-IAM-Trace-Id": "token_ null_af2953f2bcc67a42325a69a19e6c32a2",  
"Date": "Tue, 21 May 2024 09:03:40 GMT",  
"Content-Type": "application/json; charset=utf8"
```

(Optional) Response Body

The body of a response is often returned in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following shows part of the response body for the API used to [create an IAM user](#).

```
{  
  "user": {  
    "id": "c131886aec...",  
    "name": "IAMUser",  
    "description": "IAM User Description",  
    "areacode": "",  
    "phone": "",  
    "email": "****@***.com",  
    "status": null,  
    "enabled": true,  
    "pwd_status": false,  
    "access_mode": "default",  
    "is_domain_owner": false,  
    "xuser_id": "",  
    "xuser_type": "",  
    "password_expires_at": null,  
    "create_time": "2024-05-21T09:03:41.000000",  
    "domain_id": "d78cbac1.....",  
    "xdomain_id": "30086000.....",  
    "xdomain_type": "",  
    "default_project_id": null  
  }  
}
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{  
  "error_msg": "The request message format is invalid.",  
  "error_code": "IMG.0001"  
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 APIs

4.1 DSS Management

4.1.1 Obtaining Details of a DSS Storage Pool

Function

Obtaining Details of a DSS Storage Pool

Calling Method

For details, see [Calling APIs](#).

Authorization Information

Each account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions. For details about the required permissions, see [Permissions Policies and Supported Actions](#).

URI

GET /v1/{project_id}/pools/{dss_id}

Table 4-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID. For details, see Obtaining a Project ID .
dss_id	Yes	String	The storage pool ID.

Table 4-2 Query Parameters

Parameter	Mandatory	Type	Description
usage	No	Boolean	Whether capacity information of the storage pool is returned. The value can be true (returns capacity information) or false (does not return capacity information). The default value is false .

Request Parameters

Table 4-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The user token. It can be obtained by calling the IAM API used to obtain a user token. The value of X-Subject-Token in the response header is the user token.

Response Parameters

Status code: 200

Table 4-4 Response body parameters

Parameter	Type	Description
name	String	The storage pool name.
id	String	The storage pool ID.
project_id	String	The ID of the belonging project.
capacity	Integer	The requested storage pool capacity, in GB.
type	String	The storage pool type. SAS : high I/O storage pool SSD : ultra-high I/O storage pool
status	String	The storage pool status. For details, see DSS Storage Pool Status .

Parameter	Type	Description
availability_zone	String	The AZ where the storage pool resides.
created_at	String	The time when the storage pool was created.
total_capacity_gb	Integer	The total capacity.
used_capacity_gb	Integer	The used capacity.
provisioned_capacity_gb	Integer	The allocated capacity.
max_over_subscription_ratio	Float	The overcommitment ratio.

Example Requests

```
GET https://{endpoint}/v1/{project_id}/pools/{dss_id}?usage=tur
```

Example Responses

Status code: 200

Details of storage pools

```
{
  "name" : "dedicatedStorage01",
  "id" : "c950ee97-587c-4f24-8a74-3367e3da570f",
  "project_id" : "63d910f2705a487ebe4e1c274748d9e1",
  "capacity" : 100,
  "type" : "SSD",
  "availability_zone" : "AZ1",
  "status" : "available",
  "created_at" : "2022-12-18T15:57:56.299000",
  "total_capacity_gb" : 1000,
  "used_capacity_gb" : 300,
  "provisioned_capacity_gb" : 700,
  "max_over_subscription_ratio" : "1.0"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dss.v1.region.DssRegion;
import com.huaweicloud.sdk.dss.v1.*;
import com.huaweicloud.sdk.dss.v1.model.*;

public class ShowPoolSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DssClient client = DssClient.newBuilder()
        .withCredential(auth)
        .withRegion(DssRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowPoolRequest request = new ShowPoolRequest();
    request.withDssId("{dss_id}");
    try {
        ShowPoolResponse response = client.showPool(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdss.v1.region.dss_region import DssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowPoolRequest()
        request.dss_id = "{dss_id}"
```

```
response = client.show_pool(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth, err := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        SafeBuild()

    if err != nil {
        fmt.Println(err)
        return
    }

    hcClient, err := dss.DssClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        SafeBuild()

    if err != nil {
        fmt.Println(err)
        return
    }

    client := dss.NewDssClient(hcClient)

    request := &model.ShowPoolRequest{}
    request.DssId = "{dss_id}"
    response, err := client.ShowPool(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Details of storage pools

Error Codes

See [Error Codes](#).

4.1.2 Obtaining Details of DSS Storage Pools

Function

This API is used to obtain the DSS storage pools requested by a tenant. Filter query and pagination query are supported.

Calling Method

For details, see [Calling APIs](#).

Authorization Information

Each account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions. For details about the required permissions, see [Permissions Policies and Supported Actions](#).

URI

GET /v1/{project_id}/pools/detail

Table 4-5 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID. For details, see Obtaining a Project ID .

Table 4-6 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	The maximum number of query results that can be returned. The value ranges from 1 to 1000 , and the default value is 1000 . The returned value cannot exceed this limit.
offset	No	Integer	The start address of the pagination query (query offset). All storage pools after this offset are queried. The value must be a positive integer or 0 .
status	No	String	The storage pool status, which can be available , deploying , or extending .
name	No	String	The storage pool name.
usage	No	Boolean	Whether capacity information of the storage pool is returned. The value can be true (returns capacity information) or false (does not return capacity information). The default value is false .

Request Parameters

Table 4-7 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The user token. It can be obtained by calling the IAM API used to obtain a user token. The value of X-Subject-Token in the response header is the user token.

Response Parameters

Status code: 200

Table 4-8 Response body parameters

Parameter	Type	Description
count	Integer	The number of storage pools.
pools	Array of PoolDetails objects	The details of storage pools.

Table 4-9 PoolDetails

Parameter	Type	Description
name	String	The storage pool name.
id	String	The storage pool ID.
project_id	String	The ID of the belonging project.
capacity	Integer	The requested storage pool capacity, in GB.
type	String	The storage pool type. SAS : high I/O storage pool SSD : ultra-high I/O storage pool
status	String	The storage pool status. For details, see DSS Storage Pool Status .
availability_zone	String	The AZ where the storage pool resides.
created_at	String	The time when the storage pool was created.
total_capacity_gb	Integer	The total capacity.
used_capacity_gb	Integer	The used capacity.
provisioned_capacity_gb	Integer	The allocated capacity.
max_over_subscription_ratio	Float	The overcommitment ratio.

Example Requests

```
GET https://{endpoint}/v1/{project_id}/pools/detail?status=available&usage=true
```

Example Responses

Status code: 200

Details of storage pools

```
{
  "pools" : [ {
    "name" : "dedicatedStorage01",
    "id" : "c950ee97-587c-4f24-8a74-3367e3da570f",
    "project_id" : "63d910f2705a487ebe4e1c274748d9e1",
    "capacity" : 1000,
    "type" : "SSD",
    "availability_zone" : "AZ1",
    "status" : "available",
    "created_at" : "2014-12-18T15:57:56.299000",
    "total_capacity_gb" : 850,
    "used_capacity_gb" : 300,
    "provisioned_capacity_gb" : 700,
    "max_over_subscription_ratio" : 1.0
  }, {
    "name" : "dedicatedStorage02",
    "id" : "6edbc2f4-1507-44f8-ac0d-eed1d2608d38",
    "project_id" : "63d910f2705a487ebe4e1c274748d9e1",
    "capacity" : 1000,
    "type" : "SSD",
    "availability_zone" : "AZ1",
    "status" : "available",
    "created_at" : "2014-12-18T15:57:56.299000",
    "total_capacity_gb" : 850,
    "used_capacity_gb" : 300,
    "provisioned_capacity_gb" : 700,
    "max_over_subscription_ratio" : 1.0
  } ],
  "count" : 2
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dss.v1.region.DssRegion;
import com.huaweicloud.sdk.dss.v1.*;
import com.huaweicloud.sdk.dss.v1.model.*;

public class ListPoolsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DssClient client = DssClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(DssRegion.valueOf("<YOUR REGION>"))
        .build();
ListPoolsRequest request = new ListPoolsRequest();
try {
    ListPoolsResponse response = client.listPools(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdss.v1.region.dss_region import DssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DssRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = ListPoolsRequest()
        response = client.list_pools(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth, err := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        SafeBuild()

    if err != nil {
        fmt.Println(err)
        return
    }

    hcClient, err := dss.DssClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        SafeBuild()

    if err != nil {
        fmt.Println(err)
        return
    }

    client := dss.NewDssClient(hcClient)

    request := &model.ListPoolsRequest{}
    response, err := client.ListPools(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Details of storage pools

Error Codes

See [Error Codes](#).

4.1.3 Updating the Name of a DSS Storage Pool

Function

This API is used to update the name of a DSS storage pool.

Calling Method

For details, see [Calling APIs](#).

Authorization Information

Each account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions. For details about the required permissions, see [Permissions Policies and Supported Actions](#).

URI

PUT /v1/{project_id}/pools/{dss_id}

Table 4-10 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID. For details, see Obtaining a Project ID .
dss_id	Yes	String	The DSS storage pool ID.

Request Parameters

Table 4-11 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The user token. You can obtain it by calling the IAM API for obtaining a user token. The token is the value of X-Subject-Token in the response header.

Table 4-12 Request body parameters

Parameter	Mandatory	Type	Description
pool	No	DssUpdatePoolDetail object	

Table 4-13 DssUpdatePoolDetail

Parameter	Mandatory	Type	Description
name	No	String	The DSS storage name.

Response Parameters

Status code: 200

Table 4-14 Response body parameters

Parameter	Type	Description
name	String	The storage pool name.
id	String	The storage pool ID.
project_id	String	The ID of the belonging project.
capacity	Integer	The requested storage pool capacity, in GB.
type	String	The storage pool type. SAS : high I/O storage pool SSD : ultra-high I/O storage pool
status	String	The storage pool status. For details, see DSS Storage Pool Status .
availability_zone	String	The AZ where the storage pool resides.
created_at	String	The time when the storage pool was created.
total_capacity_gb	Integer	The total capacity.
used_capacity_gb	Integer	The used capacity.
provisioned_capacity_gb	Integer	The allocated capacity.
max_over_subscription_ratio	Float	The overcommitment ratio.

Example Requests

```
PUT https://{endpoint}/v1/{project_id}/pools/{dss_id}

{
  "pool" : {
    "name" : "dss-test"
  }
}
```

Example Responses

Status code: 200

The DSS details.

```
{
  "name" : "dedicatedStorage01",
  "id" : "c950ee97-587c-4f24-8a74-3367e3da570f",
  "project_id" : "63d910f2705a487ebe4e1c274748d9e1",
  "capacity" : 100,
  "type" : "SSD",
  "availability_zone" : "AZ1",
  "status" : "available",
  "created_at" : "2022-12-18T15:57:56.299000",
  "total_capacity_gb" : 1000,
  "used_capacity_gb" : 300,
  "provisioned_capacity_gb" : 700,
  "max_over_subscription_ratio" : "1.0"
}
```

Status Codes

Status Code	Description
200	The DSS details.

Error Codes

See [Error Codes](#).

4.1.4 Obtaining Details of Versions

Function

This API is used to obtain the details of the versions.

Calling Method

For details, see [Calling APIs](#).

Authorization Information

Each account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions. For details about the required permissions, see [Permissions Policies and Supported Actions](#).

URI

GET /

Request Parameters

Table 4-15 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The user token. It can be obtained by calling the IAM API used to obtain a user token. The value of X-Subject-Token in the response header is the user token.

Response Parameters

Status code: 200

Table 4-16 Response body parameters

Parameter	Type	Description
versions	Array of Version objects	The list of the version objects.

Table 4-17 Version

Parameter	Type	Description
id	String	The version ID.
links	VersionLinks object	The version links.
status	String	The version status.
updated	String	The version release time.
version	String	The maximum microversion. If there is no microversion, this parameter is left blank.
min_version	String	The minimum microversion. If there is no microversion, this parameter is left blank.

Table 4-18 VersionLinks

Parameter	Type	Description
rel	String	The link description.
href	String	The version query link.

Example Requests

```
https://{endpoint}/
```

Example Responses

Status code: 200

Details of the versions

```
{
  "versions" : [ {
    "id" : "v1",
    "links" : {
      "rel" : "self",
      "href" : "https://dss.br-iaas-odin1.huaweicloud.com/v1"
    },
    "status" : "CURRENT",
    "updated" : "2017-11-30T00:00:00Z",
    "version" : "",
    "min_version" : ""
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dss.v1.region.DssRegion;
import com.huaweicloud.sdk.dss.v1.*;
import com.huaweicloud.sdk.dss.v1.model.*;

public class ListVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
```

```
.withSk(sk);

DssClient client = DssClient.newBuilder()
    .withCredential(auth)
    .withRegion(DssRegion.valueOf("<YOUR REGION>"))
    .build();
ListVersionRequest request = new ListVersionRequest();
try {
    ListVersionResponse response = client.listVersion(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdss.v1.region.dss_region import DssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = DssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListVersionRequest()
        response = client.list_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1/model"
```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth, err := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        SafeBuild()

    if err != nil {
        fmt.Println(err)
        return
    }

    hcClient, err := dss.DssClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        SafeBuild()

    if err != nil {
        fmt.Println(err)
        return
    }

    client := dss.NewDssClient(hcClient)

    request := &model.ListVersionRequest{}
    response, err := client.ListVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Details of the versions

Error Codes

See [Error Codes](#).

4.1.5 Obtaining Details of a Version

Function

This API is used to obtain the details of a version.

Calling Method

For details, see [Calling APIs](#).

Authorization Information

Each account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions. For details about the required permissions, see [Permissions Policies and Supported Actions](#).

URI

GET /{api_version}

Table 4-19 Path Parameters

Parameter	Mandatory	Type	Description
api_version	Yes	String	The target API version. The value is v1 .

Request Parameters

Table 4-20 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The user token. It can be obtained by calling the IAM API used to obtain a user token. The value of X-Subject-Token in the response header is the user token.

Response Parameters

Status code: 200

Table 4-21 Response body parameters

Parameter	Type	Description
version	Version object	

Table 4-22 Version

Parameter	Type	Description
id	String	The version ID.
links	VersionLinks object	The version links.
status	String	The version status.
updated	String	The version release time.
version	String	The maximum microversion. If there is no microversion, this parameter is left blank.
min_version	String	The minimum microversion. If there is no microversion, this parameter is left blank.

Table 4-23 VersionLinks

Parameter	Type	Description
rel	String	The link description.
href	String	The version query link.

Example Requests

```
https://{endpoint}/{api_version}
```

Example Responses

Status code: 200

Details of a version

```
{
  "version" : {
    "id" : "v1",
    "links" : {
      "rel" : "self",
      "href" : "https://dss.br-iaas-odin1.huaweicloud.com/v1"
    },
    "status" : "CURRENT",
    "updated" : "2017-11-30T00:00:00Z",
```

```
"version" : "",  
"min_version" : ""  
}  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dss.v1.region.DssRegion;  
import com.huaweicloud.sdk.dss.v1.*;  
import com.huaweicloud.sdk.dss.v1.model.*;  
  
public class ShowVersionsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        DssClient client = DssClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DssRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowVersionsRequest request = new ShowVersionsRequest();  
        request.withApiVersion("{api_version}");  
        try {  
            ShowVersionsResponse response = client.showVersions(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdss.v1.region.dss_region import DssRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdss.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = DssClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DssRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowVersionsRequest()
        request.api_version = "{api_version}"
        response = client.show_versions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dss "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dss/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth, err := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        SafeBuild()

    if err != nil {
        fmt.Println(err)
        return
    }

    hcClient, err := dss.DssClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        SafeBuild()

    if err != nil {
```

```
        fmt.Println(err)
        return
    }

    client := dss.NewDssClient(hcClient)

    request := &model.ShowVersionsRequest{}
    request.ApiVersion = "{api_version}"
    response, err := client.ShowVersions(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Details of a version

Error Codes

See [Error Codes](#).

5 Permissions and Supported Actions

5.1 Introduction

You can use Identity and Access Management (IAM) for fine-grained permissions management of your DSS resources. If your HUAWEI IDaccount does not need individual IAM users, you can skip this section.

With IAM, you can control access to specific Huawei Cloudcloud resources by granting permissions to principals (IAM users, user groups, and agencies). IAM supports role/policy-based authorization and identity policy-based authorization.

The following table describes the differences between the two authorization models.

Table 5-1 Differences between role/policy-based authorization and identity policy-based authorization

Autho rization Model	Authoriz ation Using	Permissio ns	Authorization Method	Scenario
Role/ Policy	User-permissi on- authoriz ation scope	• Syste m- define d roles • Syste m- define d policie s • Custom policie s	Assigning roles or policies to principals	To grant permissions to a user, you must first add the user to a user group and then define the scope of authorization. It is hard to provide fine-grained permissions control using role/policy-based authorization and a limited number of condition keys. This method is suitable for small- and medium-sized enterprises.

Authorization Model	Authorization Using	Permissions	Authorization Method	Scenario
Identity policy	User-policy	- System-defined identity policies - Custom identity policies	- Assigning identity policies to principals - Attaching identity policies to principals	You can grant permissions to a user by attaching an identity policy to it. User-specific authorization and a variety of key conditions allow for more fine-grained permissions control. However, this model can be hard to set up. It requires a certain amount of expertise and is suitable for medium- and large-sized enterprises.

Assume that you want to grant IAM users permission to create ECSs in CN North-Beijing4 and OBS buckets in CN South-Guangzhou. With role/policy-based authorization, the administrator needs to create two custom policies and assign both to the IAM users. With identity policy-based authorization, the administrator only needs to create one custom identity policy and configure the condition key **g:RequestedRegion** for the policy, and then attach the policy to the users or grant the users the access permissions to the specified regions. Identity policy-based authorization is more flexible than role/policy-based authorization.

Policies/Identity policies and actions in the two authorization models are not interoperable. You are advised to use the identity policy-based authorization model.

If you use IAM users in your account to call an API, the IAM users must be granted the required permissions. The required permissions are determined by the actions supported by the API. Only users with the policies allowing for those actions can call the API successfully.

5.2 Actions Supported by Identity Policy-based Authorization

IAM provides system-defined identity policies to define typical cloud service permissions. You can also create custom identity policies using the actions supported by cloud services for more refined access control.

In addition to IAM, the [Organizations](#) service also provides [Service Control Policies \(SCPs\)](#) to set access control policies.

SCPs do not actually grant any permissions to an entity. They only set the permissions boundary for the entity. When SCPs are attached to an organizational

unit (OU) or a member account, the SCPs do not directly grant permissions to that OU or member account. Instead, the SCPs only determine what permissions are available for that member account or those member accounts under that OU. The granted permissions can be applied only if they are allowed by the SCPs.

To learn more about how IAM is different from Organizations for access control, see [How IAM Is Different from Organizations for Access Control?](#).

This section describes the elements used by IAM custom identity policies and Organizations SCPs. The elements include actions, resources, and conditions.

- For details about how to use these elements to edit an IAM custom identity policy, see [Creating a Custom Identity Policy](#).
- For details about how to use these elements to edit a custom SCP, see [Creating an SCP](#).

Actions

Actions are specific operations that are allowed or denied in an identity policy.

- The Access Level column describes how the action is classified (List, Read, or Write). This classification helps you understand the level of access that an action grants when you use it in an identity policy.
- The Resource Type column indicates whether the action supports resource-level permissions.
 - You can use a wildcard (*) to indicate all resource types. If this column is empty (-), the action does not support resource-level permissions and you must specify all resources ("*") in your identity policy statements.
 - If this column includes a resource type, you must specify the URN in the Resource element of your identity policy statements.
 - Required resources are marked with asterisks (*) in the table. If you specify a resource in a statement using this action, then it must be of this type.

For details about the resource types defined by DSS, see [Resources](#).

- The Condition Key column contains keys that you can specify in the Condition element of an identity policy statement.
 - If the Resource Type column has values for an action, the condition key takes effect only for the listed resource types.
 - If the Resource Type column is empty (-) for an action, the condition key takes effect for all resources that action supports.
 - If the Condition Key column is empty (-) for an action, the action does not support any condition keys.

For details about the condition keys defined by DSS, see [Conditions](#).

- The Alias column lists the policy actions that are configured in identity policies. With these actions, you can use APIs for policy-based authorization. For details, see [Policies and Identity Policies](#).

The following table lists the actions that you can define in identity policy statements for DSS.

Table 5-2 Actions supported by DSS

Action	Description	Access Level	Resource Type (*: required)	Condition Key	Alias
dss:pools:list	Grants permission to list DSS pools.	List	-	g:EnterpriseProjectId	-
dss:pools:get	Grants permission to show a DSS pool.	Read	pool *	g:EnterpriseProjectId	-
dss:pools:update	Grants permission to update a DSS pool.	Write	pool *	g:EnterpriseProjectId	-

Each API of DSS usually supports one or more actions. [Table 5-3](#) lists the supported actions and dependencies.

Table 5-3 Actions and dependencies supported by DSS APIs

API	Action	Dependencies
GET /v1/{project_id}/pools/detail	dss:pools:list	-
GET /v1/{project_id}/pools/{dss_id}	dss:pools:get	-
PUT /v1/{project_id}/pools/{dss_id}	dss:pools:update	-

Resources

A resource type (Resource) defines the resources on which an identity policy takes effect. In [Table 5-4](#), if a specific operation has an associated resource type, then in the identity policy statement containing that operation, the policy will only take effect on the specified resource when the corresponding resource URN is provided. If no resource URN is specified, the policy will apply to all resources under the designated resource type. If no resource type is defined, the default value of Resource is *, meaning the policy applies to all resources. In addition, you can set conditions in the identity policy to further refine and restrict the resource type.

The following table lists the resource types that you can define in identity policy statements for DSS.

Table 5-4 Resource types supported by DSS

Resource Type	URN
pool	dss:<region>:<account-id>:pool:<pool-id>

Conditions

DSS does not support service-specific condition keys in identity policies. It can only use global condition keys applicable to all services. For details, see [Global Condition Keys](#).

5.3 Policy-based Authorization

Supported Actions

DSS provides system-defined policies that can be directly used in IAM. You can also create custom policies to supplement system-defined policies for more refined access control. Actions supported by policies are specific to APIs. The following are common concepts related to policies:

- **Permissions:** statements in a policy that allow or deny certain operations
- **APIs:** REST APIs that can be called by a user who has been granted specific permissions
- **Actions:** specific operations that are allowed or denied
- **Dependencies:** actions which a specific action depends on. When allowing an action for a user, you also need to allow any existing action dependencies for that user.
- **IAM projects/Enterprise projects:** the authorization scope of a custom policy. A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions for both IAM and enterprise projects can be used and applied for both IAM and Enterprise Management. Policies that contain actions only for IAM projects can be used and applied to IAM only. For details about the differences between IAM and enterprise management, see [Differences Between IAM and Enterprise Management](#).

DSS supports the following actions that can be defined in custom policies:

- **DSS Storage Pool Management** actions, including actions supported by DSS storage pool management APIs. For example, the APIs for obtaining details of a DSS storage pool and obtaining details of DSS storage pools.
- **Disk Management** actions, including actions supported by DSS disk management APIs. For example, the API for querying details about all disks.

DSS Storage Pool Management

Permission	API	Action	IAM Project (Project)	Enterprise Project (Enterprise Project)
Obtaining details of a DSS storage pool	GET /v1/{project_id}/pools/{dss_id}	dss.action.querypool	√	√
Obtaining details of DSS storage pools	GET /v1/{project_id}/pools/detail	dss.action.listpools	√	√

Disk Management

Permission	API	Action	IAM Project (Project)	Enterprise Project (Enterprise Project)
Querying details about all disks	GET /v2/{project_id}/cloudvolumes/detail	evs:volume s:list	√	√

AAppendixes

A.1 DSS Storage Pool Status

Table A-1 Storage pool status

Status	Description
available	The storage pool is available for use.
deploying	The storage pool is being deployed and cannot be used.
extending	The storage pool capacity is being expanded. The storage pool can be used.

A.2 Error Codes

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in [API Gateway Error Codes](#).

Error Code Description

Status Code	Error Code	Error Message	Description	Handling Measure
400	DSS.001	invalid project id!	Incorrect tenant ID in the URI.	Use the correct project ID.
400	DSS.002	invalid token!	Header parameters in the HTTP request are incorrect.	Use the correct token.

Stat us Cod e	Error Code	Error Message	Description	Handling Measure
400	DSS.003	invalid token roles!	The token used is incorrect.	Check whether the token has the desired role. If not, add the role for the token.
400	DSS.1001	invalid dss id!	Invalid storage pool ID.	Modify the storage pool ID format.
400	DSS.1002	invalid dss name!	Parameter name in the URL for querying the storage pool details is incorrect.	Modify the storage pool name format.
400	DSS.1003	invalid filter limit!	Parameter limit in the URL for querying the storage pool details is incorrect.	Enter a value larger than 0 for parameter limit .
400	DSS.1004	invalid filter offset!	Parameter offset in the URL for querying the storage pool details is incorrect.	Check whether the offset parameter in the request is correct.
400	DSS.1005	invalid filter usage!	Parameter usage in the URL for querying the storage pool details is incorrect.	Check whether the usage parameter in the request is correct.
400	DSS.1006	api roles is null or empty!	User permission error.	Add the required user permission.
400	DSS.1007	User role is not allowed for this action!	You have no permission to the operation.	Add the required user permission.
400	DSS.1008	Type conversion error , parameter type is unexpected	Type conversion error. The parameter type is unexpected.	Check whether the input parameters are correct.
400	DSS.1009	url encoding failed!	Type conversion error.	Check whether the input parameters are correct.
500	DSS.1010	internal error!	The service is unavailable.	Contact technical support.

A.3 Status Codes

- Normal

Status Code	Description
200	OK
201	Created
202	Accepted
204	No Content

- Abnormal

Status Code	Description
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
413	overLimit
415	badMediaType
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

A.4 Obtaining a Project ID

Scenarios

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. Two methods are available:

- [Obtain the Project ID by Calling an API](#)
- [Obtain the Project ID from the Console](#)

Obtain the Project ID by Calling an API

You can obtain a project ID by calling the API used to [query projects based on specified criteria](#).

The API used to obtain a project ID is GET `https://{Endpoint}/v3/projects`. {Endpoint} is the IAM endpoint and can be obtained from [Regions and Endpoints](#). For details about API authentication, see [Authentication](#).

The following is an example response. The value of **id** is the project ID.

```
{
  "projects": [
    {
      "domain_id": "65ewtrgaggshhk1223245sghjlse684b",
      "is_domain": false,
      "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
      "name": "project_name",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4adasfjljaaakla12334jklga9sasfg"
      },
      "id": "a4adasfjljaaakla12334jklga9sasfg",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

Obtain a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Access [My Credentials](#).
2. On the **API Credentials** page, check the project ID in the project list.

Figure A-1 Viewing the project ID

