Object Storage Service(OBS)

Image Processing Feature Guide

Issue 01

Date 2024-10-24

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud 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.

Contents

1 Introduction

1.1 What Is Image Processing?

Introduction

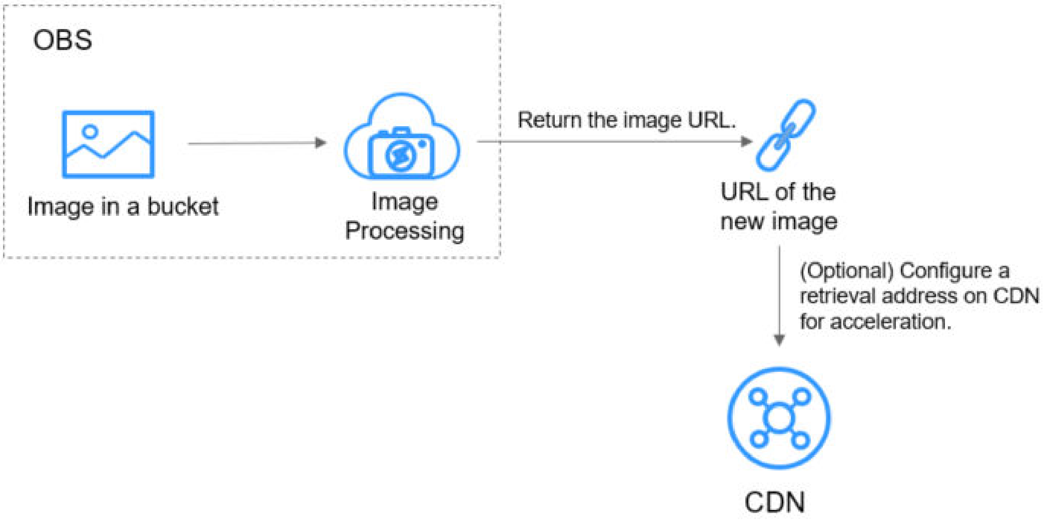
Image processing is a feature integrated in Object Storage Service (OBS). It provides stable, secure, efficient, and inexpensive image processing services. By using this feature, you can slim, crop, resize, and watermark images, as well as convert the formats of images.

You can access this feature via OBS Console and REST APIs, to process images stored in OBS anytime and anywhere and obtain the processed images right away.

Architecture

You can upload your images to OBS using OBS Console, OBS Browser+, REST APIs, or third-party clients. Before downloading and using an image, you can create an image style or pass image processing parameters to process it, such as cropping and compressing the image. After processing, you can obtain the URL of the new image. Furthermore, you can deliver the new image to Content Delivery Network (CDN) for acceleration and directly download it from CDN when needed. **Figure 1-1** shows the flowchart of image processing.

Figure 1-1 Image processing flowchart



NOTE

After processing, new images will be directly displayed in a browser and are not stored in OBS, so they do not incur any storage costs.

Billing

Image processing is currently free of charge.

1.2 Application Scenarios

Image processing enables you to resize, crop, or compress images on cloud. You do not have to download space-consuming software to your local computers.

For example, you can add effects to or resize images in your cloud album anytime and anywhere, and quickly share the images with your friends online.

Shopping or other likewise websites where images are frequently accessed can use image processing to simply process the images and delivers them to Content Delivery Network (CDN) for acceleration. The processed images can then be faster downloaded from CDN.

1.3 Constraints

Operations

- All image processing operations will not change original images.
- Archive storage does not support image processing.
- OBS buckets that use SSE-KMS do not support image processing.
- Only version 3.0 buckets support image processing. To check a bucket's version, go to the **Basic Information** area on the bucket's **Overview** page on OBS Console.

- Images must be accessible to anonymous users, so that they can be accessed in a web browser by image processing requests that have no signature included. A request URL example is given as follows:

  https://bucketname.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=style/stylename

  For details about how to make images accessible to anonymous users, see **Granting Public Read Permissions on a Bucket for Anonymous Users**.
- For security and compliance purposes, using a default bucket domain name (**bucket domain name or static website domain name**) to preview objects in the bucket will be prohibited by OBS. When you use such domain name to access objects (such as videos, images, and web pages) through a browser, no content will be displayed. Instead, the content is downloaded as an attachment.

  For details, see **How Do I Preview OBS Objects in a Browser Online?**

Images

- Supported original image formats include JPG, JPEG, PNG, BMP, WebP, GIF, and TIFF.
- Supported formats after image processing include JPG, PNG, BMP, and WebP.
- An original image allowed for processing cannot exceed 25 MB in size, and its width and height after processing cannot exceed 4,096 pixels and 5,000 pixels, respectively.
- An animated image (like a GIF or WebP image) will be returned without processing if it is greater than 2 MB in size or has over 50 frames.
- Currently, processing images in CMYK may change their color.
- An input image can contain 250 megapixels at most.

Commands

A maximum of 20 commands are allowed, with a total length of no more than 512 characters.

Regions

Image processing is currently only available in some regions. For details about the supported regions, see image processing on the **Function Overview** page.

1.4 Basic Concepts

Styles

A style is an aggregation of parameters or image processing operations. When performing the same operations on multiple images, you can create an image style as a template to avoid repetitive operations. A bucket can have a maximum of 100 styles.

EXIF Data

Exchangeable Image File (EXIF) data is contained in images taken by some cameras or smartphones. EXIF data is a group of parameters inserted into JPEG or

TIFF images. It encompasses the shooting information, such as the camera model, date and time when an image was taken, shooting mode, thumbnails, and other characteristics.

1.5 Accessing Image Processing

You can access image processing in either of the following ways:

- Using OBS Console

  On the image processing page, you can create a processing style for an image by specifying parameters on the GUI or by editing code. You can then view the new image effect on the **Preview Image** tab page on the image's details page. You can also click **Copy Link** to get the link for accessing the new image.

  To get started with image processing on OBS Console, see **Process**.

- Using the REST API

  You can call OBS Representational State Transfer (REST) APIs to access image processing. In the REST architectural style, resources on a network are identified by Uniform Resource Identifiers (URIs) and applications on clients locate resources using Uniform Resource Locators (URLs). The URL is in the **https://***Endpoint***/***uri* format. By putting a URL that complies with the command rules of image processing in a browser, you can get the new image after processing. For more information, see the **Object Storage Service API Reference**.

  To get started with image processing with API, see **Process**.

1.6 Image Processing Functions

Table 1-1 describes what you can do on images stored in OBS by using image processing.

Table 1-1 Image processing functions

Function	Description	How to Use
Getting Information About an Image	Returns an image's basic information, including the format, size, and average color value.	Making an API call
Setting Image Effects	Adjusts an image's effects, including the brightness, contrast, sharpness, and blur.	Using GUI on OBS Console Editing code on OBS Console Making an API call

Function	Description	How to Use
Resizing Images	Resizes images based on a specified width, height, or percentage.	Editing code on OBS Console Making an API call
Watermarking Images	Adds a text watermark, an image watermark, or both of them to a specific position of an image. Text watermarks can have a color, font, and size configured, and image watermarks can be resized, rotated, or cropped before being added to images.	Using GUI on OBS Console Editing code on OBS Console Making an API call
Converting Formats	Converts images to various formats, and supports interlaced rendered images after conversion.	Using GUI on OBS Console Editing code on OBS Console Making an API call
Rotation Settings	Rotates images clockwise, and automatically rotates images based on the recorded rotation information of cameras or smartphones.	Editing code on OBS Console Making an API call
Common Cropping	Crops images based on the specified width, height, radius of the inscribed circle, index, and radius of rounded corners.	Editing code on OBS Console Making an API call
Changing Image Quality	Compresses JPG images based on the relative quality and absolute quality.	Editing code on OBS Console Making an API call
Slimming Images	Reduces the image size without compromising its quality.	Editing code on OBS Console Making an API call
Image Processing Persistency	Asynchronously stores images after processing in a specified OBS bucket for you to directly access the processed images.	Editing code on OBS Console Making an API call

Function	Description	How to Use
Access with Commands	Orchestrates multiple process commands in sequence. With this function, you can add multiple process commands to the URL of the image that you want to process, and separate each command using the designated delimiter. Then the commands are executed one by one from left to right.	Editing code on OBS Console Making an API call
Creating an Image Style	Customizes image styles for you to batch process images that require the same processing operations.	Using GUI on OBS Console Editing code on OBS Console

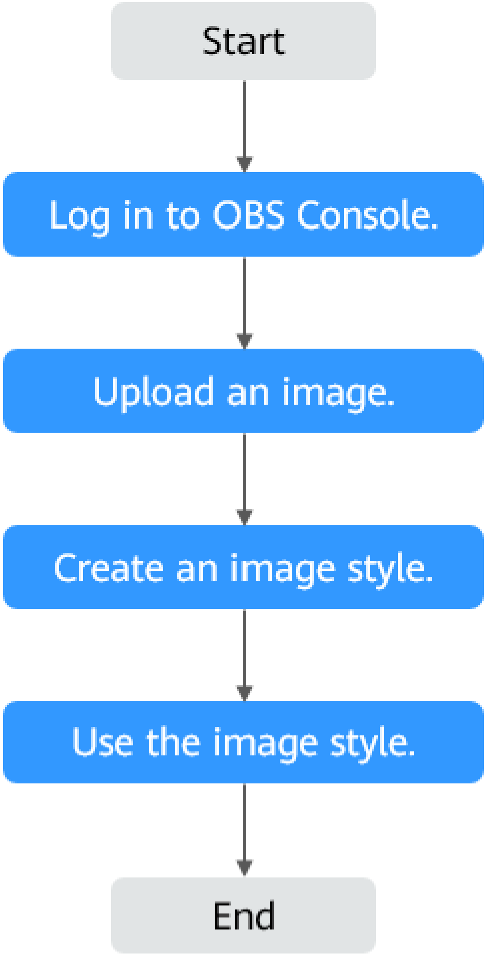
2 Using Image Processing on OBS Console

2.1 Process

On the image processing page of OBS Console, you can create a processing style for an image by specifying parameters on the GUI or by editing code. You can then view the new image effect on the **Preview Image** tab page on the image's details page. You can also click **Copy Link** to get the link for accessing the new image.

Figure 2-1 shows the flowchart of using image processing on OBS Console.

Figure 2-1 Flowchart of using image processing on OBS Console



2.2 Logging In to OBS Console

Before using image processing, you must have a cloud service account and enable OBS. Once OBS is enabled, image processing is available for use. To use OBS properly, make sure that you have sufficient account balance or you have a valid payment method. If you have already enabled OBS and you can use it properly, skip this section.

Procedure

Step 1 Visit the **Huawei Cloud official website**.

Step 2 Create a HUAWEI ID.

If you already have one, start from **Step 3**.

1. On the right of the top navigation bar, click **Sign Up**.
2. Complete the account creation as instructed.

   After the creation is complete, you will be navigated to your information page.

Step 3 On the right of the top navigation bar, click **Sign In**, and enter the username and password.

Step 4 On the right of the top navigation menu, click **Console** to go to the management console.

Step 5 Click **Service List** and choose **Storage** > **Object Storage Service**.

Step 6 (Recommended) Make sure that you have sufficient account balance or you have a valid payment method, so that you can properly use OBS.

----End

2.3 Uploading an Image

You can upload images using OBS Console, OBS Browser+, or REST APIs.

This section describes how to upload images on OBS Console. Skip this section if the image to be processed already exists in your bucket.

NOTE

For the constraints on the format and size of images you can upload, see **Constraints**.

Prerequisites

At least one bucket has been created.

Procedure

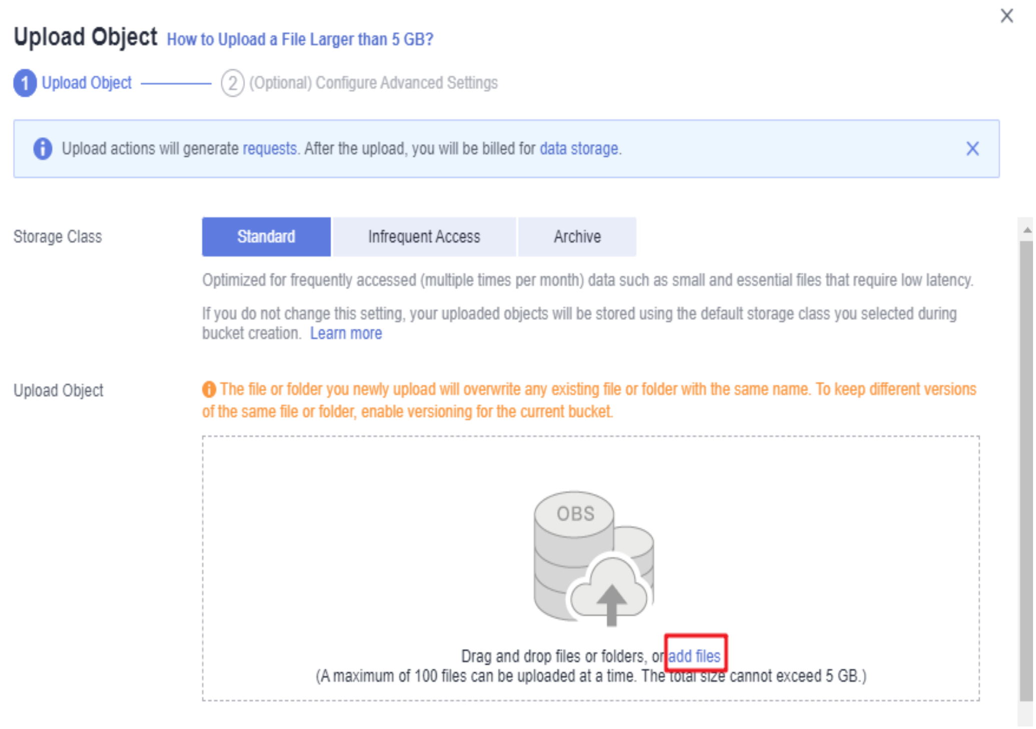
Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 In the bucket list, click the bucket you want to go to the **Objects** page.

Step 3 Click **Upload Object** to open the **Upload Object** dialog box.

Step 4 Click **add files** marked by red box in **Figure 2-2** to open the local file browser.

Figure 2-2 Uploading an image



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object — 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object: The file or folder you newly upload will overwrite any existing file or folder with the same name. To keep different versions of the same file or folder, enable versioning for the current bucket.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Step 5 Select the image that you want to upload and click **Open**.

Step 6 Click **Upload**.

----End

2.4 Creating an Image Style

Introduction

Each image style specifies a set of processing operations, including cropping, compression, and adding watermarks. For multiple images that require the same processing operations in a bucket, you can create an image style to apply it to the images.

When creating a style, you can preview the style effect of the sample image on the right in real time.

When using REST APIs to access image processing, you can call the style name in the URL to avoid entering complex commands. For details about the domain name rules for API access, see **Access with a Style Name**.

You can create up to 100 styles for a bucket.

You can also replicate existing image processing styles from another bucket.

Procedure for Creating an Image Style

Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 Click a desired bucket name. In the navigation pane, choose **Data Processing** > **Image Processing**.

Figure 2-3 Image processing

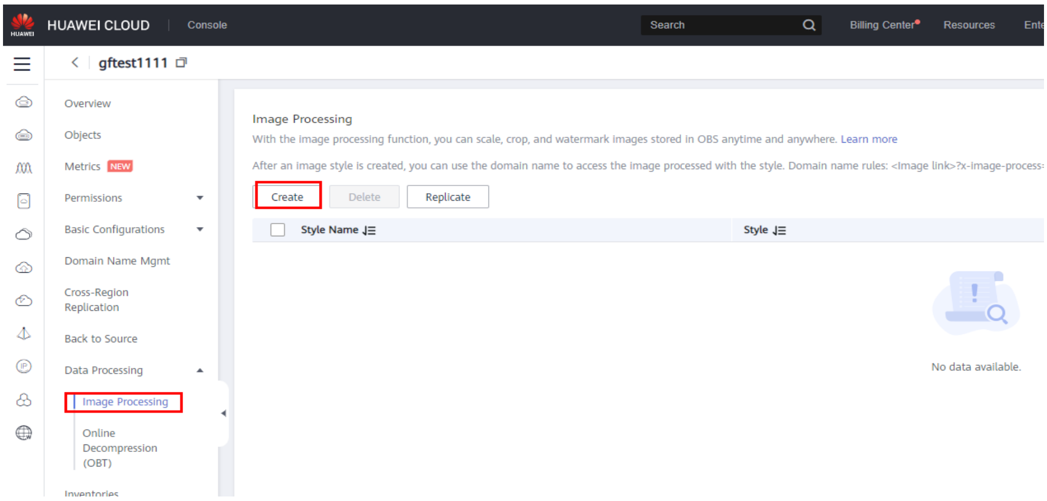


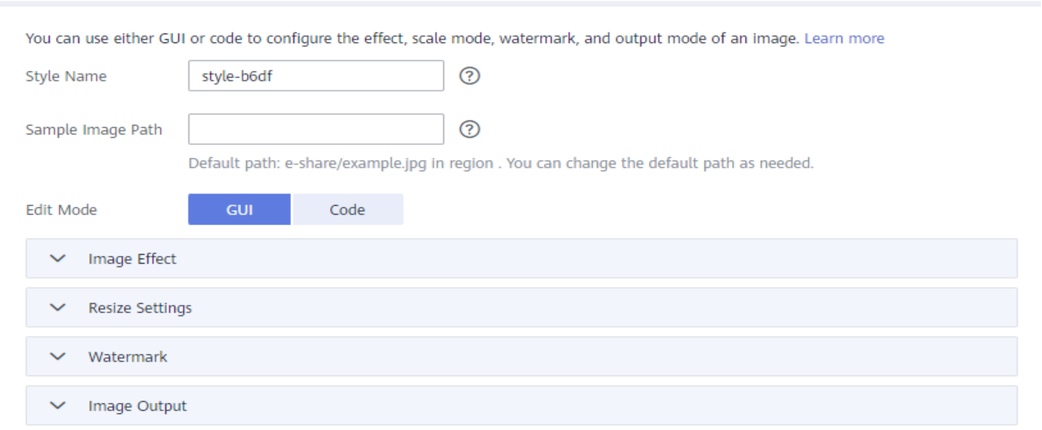
Image Processing

With the image processing function, you can scale, crop, and watermark images stored in OBS anytime and anywhere. Learn more

After an image style is created, you can use the domain name to access the image processed with the style. Domain name rules: <Image link>?x-image-process:

Step 3 Click **Create** to go to the style editing page shown in **Figure 2-4**.

Figure 2-4 Creating a style



You can use either GUI or code to configure the effect, scale mode, watermark, and output mode of an image. Learn more

Style Name: style-b6df

Sample Image Path:

Default path: e-share/example.jpg in region . You can change the default path as needed.

Edit Mode: GUI

Step 4 On the editing page, specify a style name and configure properties:

- **Style Name**

  Enter an easy-to-remember style name. Only letters (uppercase or lowercase), digits, periods (.), underscores (_), and hyphens (-) are allowed. The style name contains 1 to 256 characters, for example, **rotate_0001**.

- **Edit Mode**

  Choose either **GUI** or **Code** to create an image style by configuring parameters or by editing code.

  An example code is as follows:

  ```
  image/sharpen,100/blur,r_1,s_1/resize,m_lfit,h_400,w_400,limit_1
  ```

- Other settings

  Configure the image effect, resizing, watermark, and output.

Step 5 Click **OK** to save the style. The new style will be displayed in the style list.

----End

Procedure for Replicating Image Styles

Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 Click a desired bucket name. In the navigation pane, choose **Data Processing** > **Image Processing**.

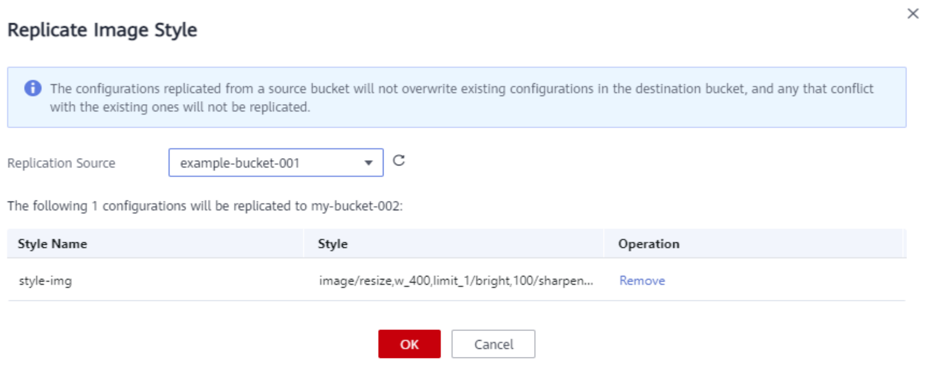
Step 3 Click **Replicate**.

Step 4 Select a replication source, which is the source bucket where the required image styles were configured.

NOTE

- The image styles replicated from a source bucket will not overwrite existing styles in the current bucket, and any that conflict with the existing ones will not be replicated.
- The version of both the source and destination buckets must be 3.0.
- You can remove the styles that you do not want to replicate.

Figure 2-5 Replicating image styles



Replicate Image Style

The configurations replicated from a source bucket will not overwrite existing configurations in the destination bucket, and any that conflict with the existing ones will not be replicated.

Replication Source: example-bucket-001

The following 1 configurations will be replicated to my-bucket-002:

Style Name	Style	Operation
style-img	image/resize,w_400,limit_1/bright,100/sharpen...	Remove

Step 5 Click **OK** to replicate the image styles to the current bucket.

----End

2.5 Using the Image Style

You can use existing image styles in a bucket in either of the following ways:

- **Copying a link** (see **Copying a Link**): On an image's details page of OBS Console, you can copy a link for accessing the image with a style applied on the **Preview Image** tab page. By pasting the copied link in a browser, you can get the new image.
- **Editing a domain name**: You can edit a domain name by referring to the format given below and open it in a browser to have the image with the style applied.

```
<Image link>?x-image-process=style/<Style name>
```

  Image link can be obtained from the image details page. For details, see **Editing a Domain Name**. *Style name* is the one defined when the style was created. A style name can contain only uppercase or lowercase letters, digits, periods (.), underscores (_), and hyphens (-) and is 1 to 256 characters long, for example, **rotate_0001**.

Copying a Link

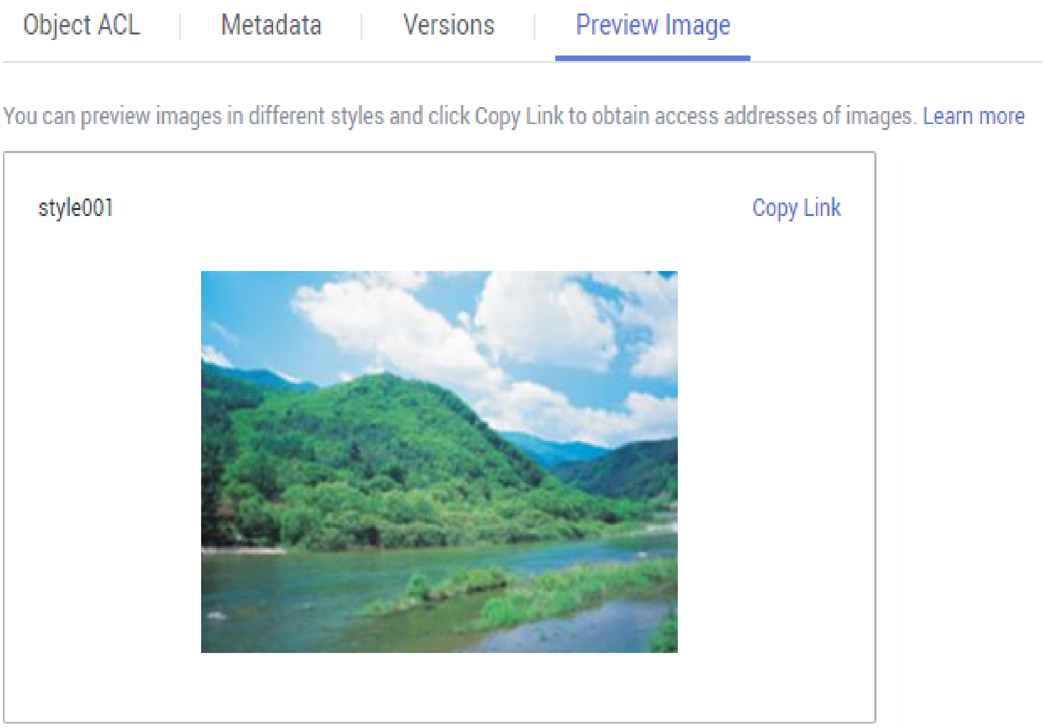
Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 Click the name of the bucket that houses desired styles to go to the **Objects** page.

Step 3 Click the name of an existing image or of a newly uploaded image to go to its details page.

Step 4 Click the **Preview Image** tab to preview the image effect in one or more styles, as shown in **Figure 2-6**.

Figure 2-6 Preview Image



You can preview images in different styles and click Copy Link to obtain access addresses of images. Learn more

Step 5 Click **Copy Link** in the style card you want. If **Copied successfully.** is prompted, you have got the link for accessing the image with the style applied.

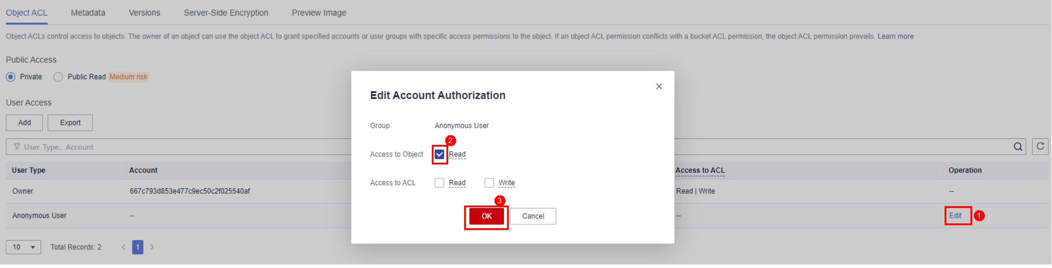
----End

Editing a Domain Name

Step 1 On the OBS object list page, click an image to be processed to go to its details page.

Step 2 Choose **Object ACL** > **Public AccessUser Access** > **Anonymous User** and click **Edit**. In the displayed dialog box, grant the object read permission to anonymous users and click **SaveOK**. For details, see **Figure 2-7**.

Figure 2-7 Granting the object read permission to anonymous users



Object ACLs control access to objects. The owner of an object can use the object ACL to grant specified accounts or user groups with specific access permissions to the object. If an object ACL permission conflicts with a bucket ACL permission, the object ACL permission prevails. Learn more

Public Access

● Private ○ Public Read Medium risk

User Access

User Type	Account	Access to ACL	Operation
Owner	667c793d853e477c9ec50c2f025540af	Read | Write	--
Anonymous User	--	--	Edit

Edit Account Authorization

Group Anonymous User

Access to Object ☑ Read

Access to ACL □ Read □ Write

Step 3 Click the icon ⧉ next to **Link** to copy the image link.

Step 4 Add **?x-image-process=style/<*Style name*>** to the end of the copied link and open the new link in a browser. Then, you can access the image with the style applied.

Link example:

https://bucketname.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=style/stylename

NOTE

The links and style names above are for reference only. Change them accordingly in practice.

----End

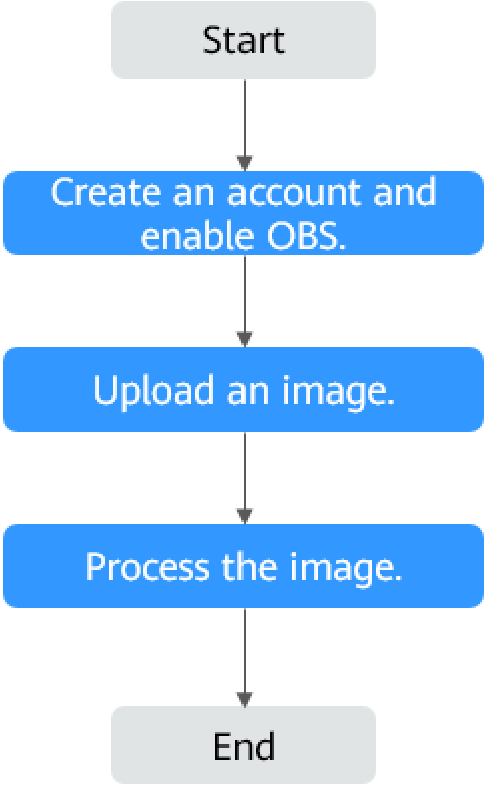
3 Using Image Processing with APIs

3.1 Process

You can call OBS REST APIs to access image processing. In the REST architectural style, resources on a network are identified by Uniform Resource Identifiers (URIs) and applications on clients locate resources using Uniform Resource Locators (URLs). The URL is in the **https://***Endpoint***/***uri* format. By putting a URL that complies with the command rules of image processing in a browser, you can get the new image after processing.

Figure 3-1 shows the flowchart of using image processing with REST APIs.

Figure 3-1 Flowchart of using image processing with REST APIs



3.2 Creating an Account and Enabling OBS

Before using image processing, you must have a cloud service account and enable OBS. Once OBS is enabled, image processing is available for use. To use OBS properly, make sure that you have sufficient account balance or you have a valid payment method. If you have already enabled OBS and you can use it properly, skip this section.

Procedure

Step 1 Visit the **Huawei Cloud official website**.

Step 2 Click **Sign Up** in the upper right corner.

Step 3 Complete the account creation as instructed.

After the creation is complete, you will be navigated to your information page.

Step 4 Create an OBS bucket, so that you can use image processing properly.

----End

3.3 Uploading an Image

You can upload images using OBS Console, OBS Browser+, or REST APIs.

For details, see **Uploading an Object**.

Skip this section if the image to be processed already exists in your bucket.

NOTE

If you want to use a user-defined domain name for image processing, you need to grant anonymous users the read permission for the image. For more information, see **Granting All Accounts the Read Permission for Certain Objects**.

3.4 Processing the Image

This section describes the URL constitution to process images using REST APIs. Once you have enabled OBS successfully, you can call REST APIs to process images simply by putting a URL that complies with the command rules of image processing in a browser.

URL Composition

A URL consists of the OBS domain name, bucket name, the original image name, and processing command or style name.

Access with Commands

URL format: **https://***bucketName.endpoint***/***objectName***?x-image-process=image/***commands*

- *endpoint* represents the endpoint address of the region where the bucket is located. You can obtain the endpoint from the bucket's basic information. For more information, see **Regions and Endpoints**.
- *bucketName* represents the name of the bucket that accommodates the image to be processed on OBS.
- *objectName* represents the name of the original image stored in the *bucketName* bucket. The name suffix must be supported by image processing.
- *commands* represents the processing commands. Three types of **delimiters** can be used between commands or command parameters. If no commands are specified, the original image will be returned.

Example URL: **https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/crop,x_100,y_50**

- **Delimiters**

Delimiters are used to separate one field from another in the commands in a URL. For details, see **Table 3-1**.

Table 3-1 Delimiters

Name	Character	Sequence	Description
Parameter delimiter	_	Fixed	Delimiter between a parameter and its value

Name	Character	Sequence	Description
Command delimiter	,	Irrelevant	Delimiter between command parameters
Pipe delimiter	/	Relevant	Delimiter between two processing commands. See **Pipes**.

- **Pipes**

If an image needs to be processed by multiple operations (for example, it needs to be cropped first and then resized), the operation commands must be separated with pipe delimiters (/). The operations (also pipes) will be executed sequentially from left to right.

For example, **https://image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100,h_100/quality,q_80** has two pipes. The pipes will be executed from left to right in sequence and the command output of the previous pipe will be used as the input of the next pipe.

Access with a Style Name

URL format: **https://***bucketName.endpoint***/***objectName***?x-image-process=style/***stylename*

- *endpoint* represents the endpoint address of the region where the bucket is located. You can obtain the endpoint from the bucket's basic information. For more information, see **Regions and Endpoints**.
- *bucketName* is the bucket name on OBS.
- *objectName* represents the name of the original image stored in the *bucketName* bucket. The name suffix must be supported by image processing.
- *stylename* is the name of an existing style in the *bucketName* bucket on OBS Console. At present, REST APIs cannot be used to perform operations related to styles, such as creating, changing, or deleting styles.

Example URL: **https://image-demo.obs.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=style/stylename**

4 Typical Cases

4.1 Processing an Image with GUI

This case describes how to add a FZShuSong text watermark to the top left of an image in the GUI mode on OBS Console.

Procedure

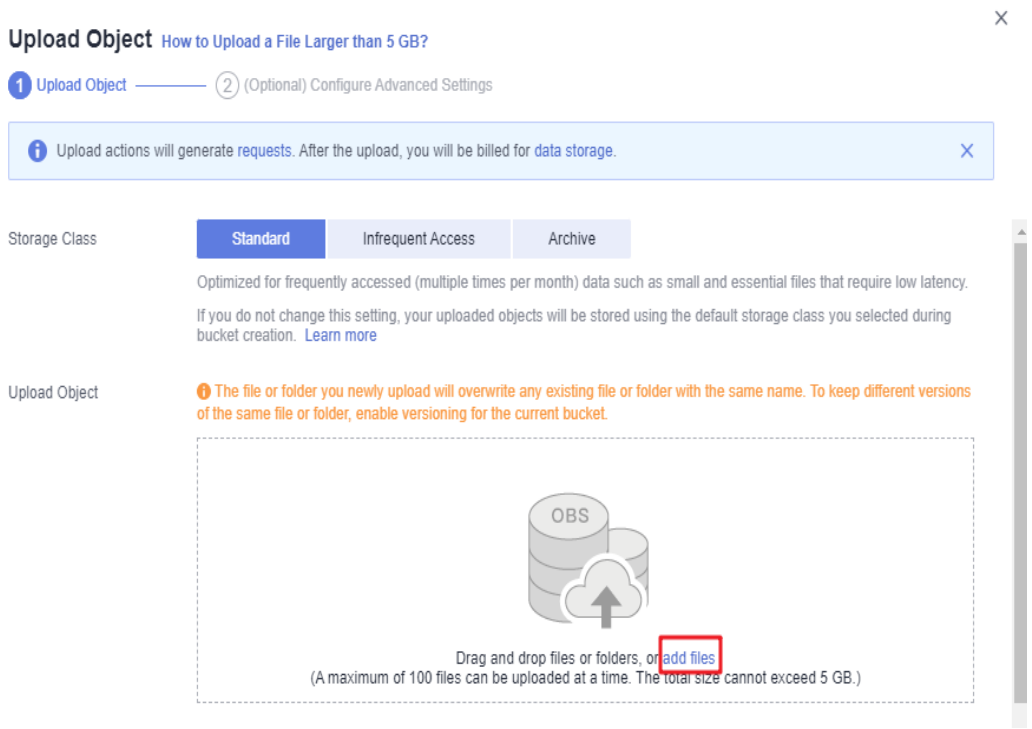
Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 In the bucket list, click the bucket you want to go to the **Objects** page.

Step 3 Click **Upload Object** to open the **Upload Object** dialog box.

Step 4 Click **add files** marked by red box in **Figure 4-1** to open the local file browser.

Figure 4-1 Uploading an image



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object —— 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object: The file or folder you newly upload will overwrite any existing file or folder with the same name. To keep different versions of the same file or folder, enable versioning for the current bucket.

Drag and drop files or folders, or add files

(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

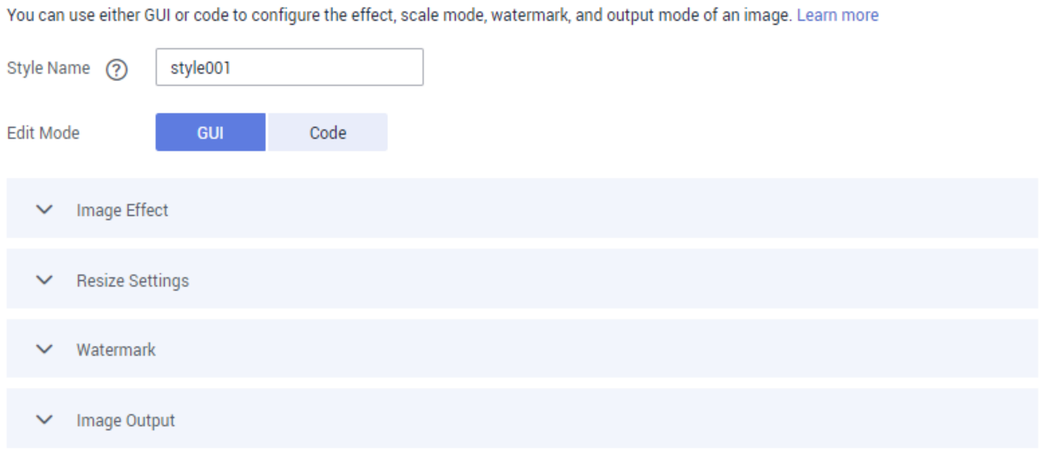
Step 5 Select the image that you want to upload and click **Open**.

Step 6 Click **Upload** to upload the image. The uploaded image is displayed in the object list.

Step 7 In the navigation pane, choose **Data Processing** > **Image Processing**.

Step 8 Click **Create** to go to the style editing page shown in **Figure 4-2**.

Figure 4-2 Creating a style



You can use either GUI or code to configure the effect, scale mode, watermark, and output mode of an image. Learn more

Style Name: style001

Edit Mode: GUI | Code

Step 9 Set **Style Name** to **obs-111** and choose **GUI** for **Edit Mode**.

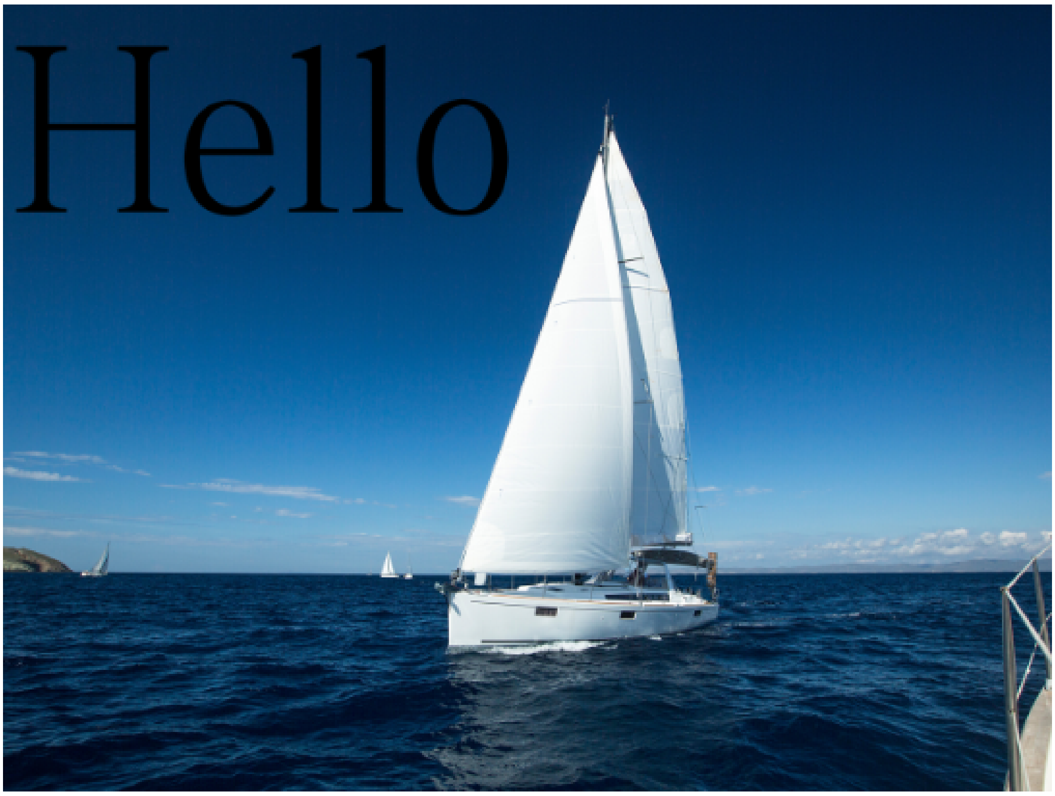
Step 10 Expand **Watermark**.

1. Choose **Text watermark** for **Watermark Type**.

2. Specify **Content** to **Hello**.
3. Choose **FZShuSong** for **Text Font**.
4. Set **Text Size** to **600**.
5. Keep the default value **100** for **Watermark Transparency**.
6. Select the top left sign (↖ ↑ ↗ ← - → ↙ ↓ ↘) for **Watermark Position**.
7. Keep **Text Shadow** disabled (○).
8. Keep the default value **10** for both **Vertical Margin** and **Horizontal Margin**.

The style effect will be displayed on the right in real time. **Figure 4-3** shows the final style effect.

Figure 4-3 Effect of style **obs-111**

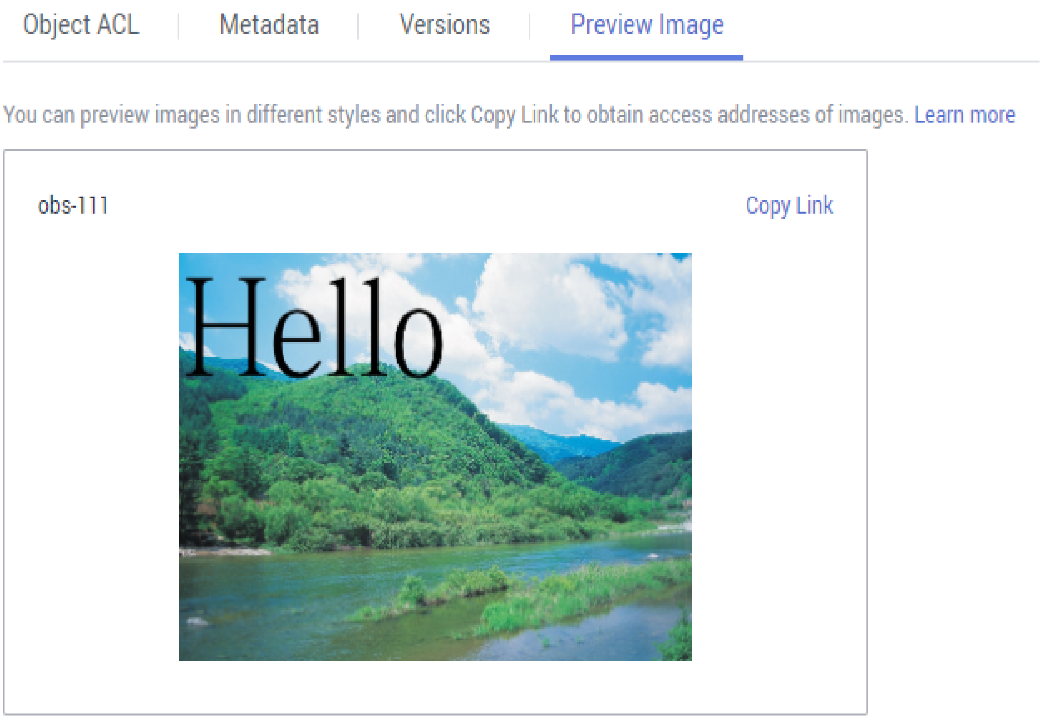


Step 11 Click **OK** to save the style. The new style **obs-111** will be displayed in the style list.

Step 12 In the navigation pane, choose **Objects**. Click **mountain.jpg** in the object list to go to the file details page.

Step 13 Click the **Preview Image** tab to preview the effect of the image with style **obs-111** applied. For details, see **Figure 4-4**.

Figure 4-4 Preview Image



You can preview images in different styles and click Copy Link to obtain access addresses of images. Learn more

Step 14 In the image style card, click **Copy Link**. When **Copied successfully.** is prompted, you will have the address for accessing the image with the style applied.

----End

4.2 Processing an Image by Editing Code

This case describes how to resize an image by editing code on OBS Console.

Procedure

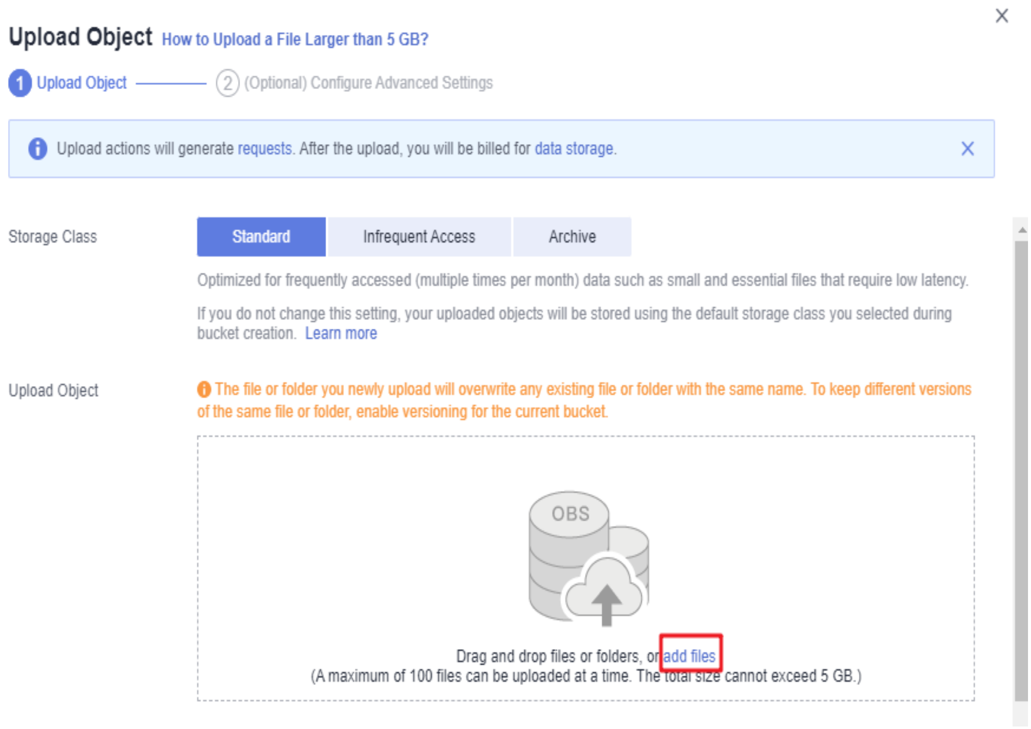
Step 1 On the console homepage, click **Service List** in the upper left corner and choose **Storage** > **Object Storage Service**.

Step 2 In the bucket list, click the bucket you want to go to the **Objects** page.

Step 3 Click **Upload Object** to open the **Upload Object** dialog box.

Step 4 Click **add files** marked by red box in **Figure 4-5** to open the local file browser.

Figure 4-5 Uploading an image



Upload Object How to Upload a File Larger than 5 GB?

1 Upload Object —— 2 (Optional) Configure Advanced Settings

Upload actions will generate requests. After the upload, you will be billed for data storage.

Storage Class: Standard | Infrequent Access | Archive

Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.

If you do not change this setting, your uploaded objects will be stored using the default storage class you selected during bucket creation. Learn more

Upload Object: The file or folder you newly upload will overwrite any existing file or folder with the same name. To keep different versions of the same file or folder, enable versioning for the current bucket.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Step 5 Select the image that you want to upload and click **Open**.

Step 6 Click **Upload** to upload the image. The uploaded image is displayed in the object list.

Step 7 In the navigation pane, choose **Data Processing** > **Image Processing**.

Step 8 Click **Create** to go to the style editing page

Step 9 Set **Style Name** to **style002** and choose **Code** for **Edit Mode**.

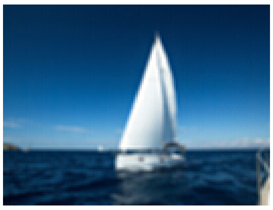
Step 10 In the code input box, type the following command for resizing.

Specify a rectangle whose width (indicated by **w**) and height (indicated by **h**) are both 100. Lock the aspect ratio, and obtain the smallest image in the extended area of the 100 x 100 rectangle.

```
image/resize,m_mfit,h_100,w_100
```

The style effect will be displayed on the right in real time. **Figure 4-6** shows the final style effect.

Figure 4-6 Effect of style **style002**

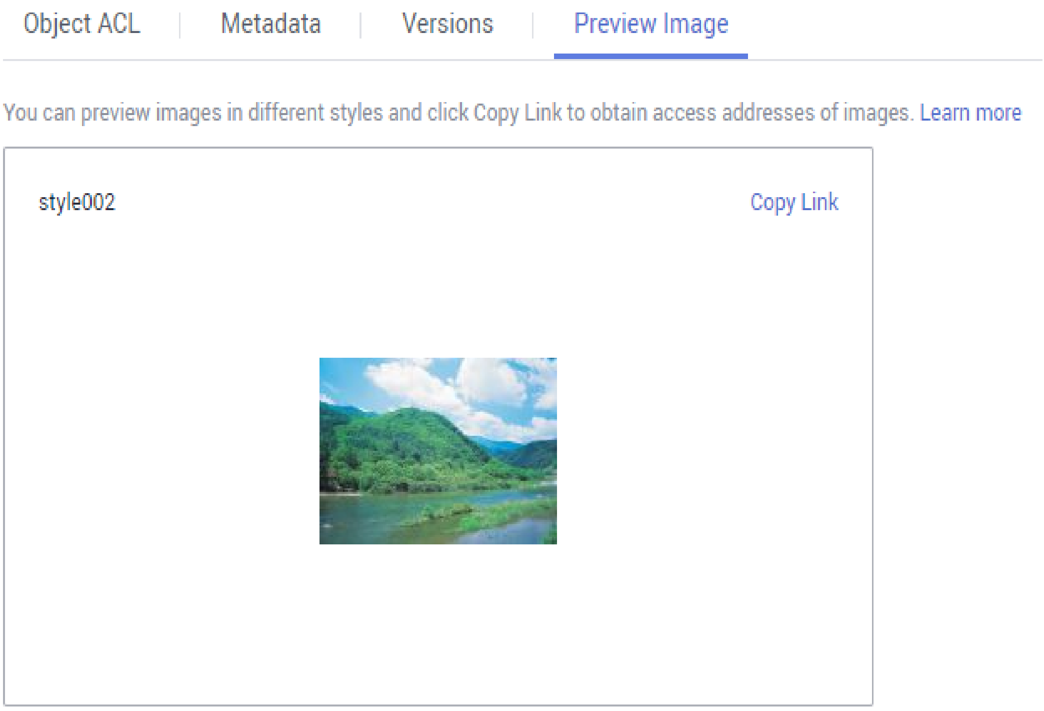


Step 11 Click **OK** to save the style. The new style **style002** will be displayed in the style list.

Step 12 In the navigation pane, choose **Objects**. Click **mountain.jpg** in the object list to go to the file details page.

Step 13 Click the **Preview Image** tab to preview the effect of the image with style **style002** applied. For details, see **Figure 4-7**.

Figure 4-7 Preview Image



You can preview images in different styles and click Copy Link to obtain access addresses of images. Learn more

Step 14 In the image style card, click **Copy Link**. When **Copied successfully.** is prompted, you will have the address for accessing the image with the style applied.

----End

5 Getting Information About an Image

You can get the information about an image only by making an API call.

The image information you can get includes the width, height, size, and format of an image. If the image contains **EXIF Data**, the EXIF data will be returned in JSON.

This operation is represented by **info**.

Example

Below gives an example of querying information about the **example.jpg** image:

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/info

```
{
    "format":"JPEG",
    "height":2000,
    "size":1046583,
    "width":2668
}
```

6 Getting the Average RGB Value of an Image

You can get the average RGB value of an image only by making an API call.

The returned average RGB value is a hexadecimal string in JSON.

This operation is represented by **average-hue**.

Example

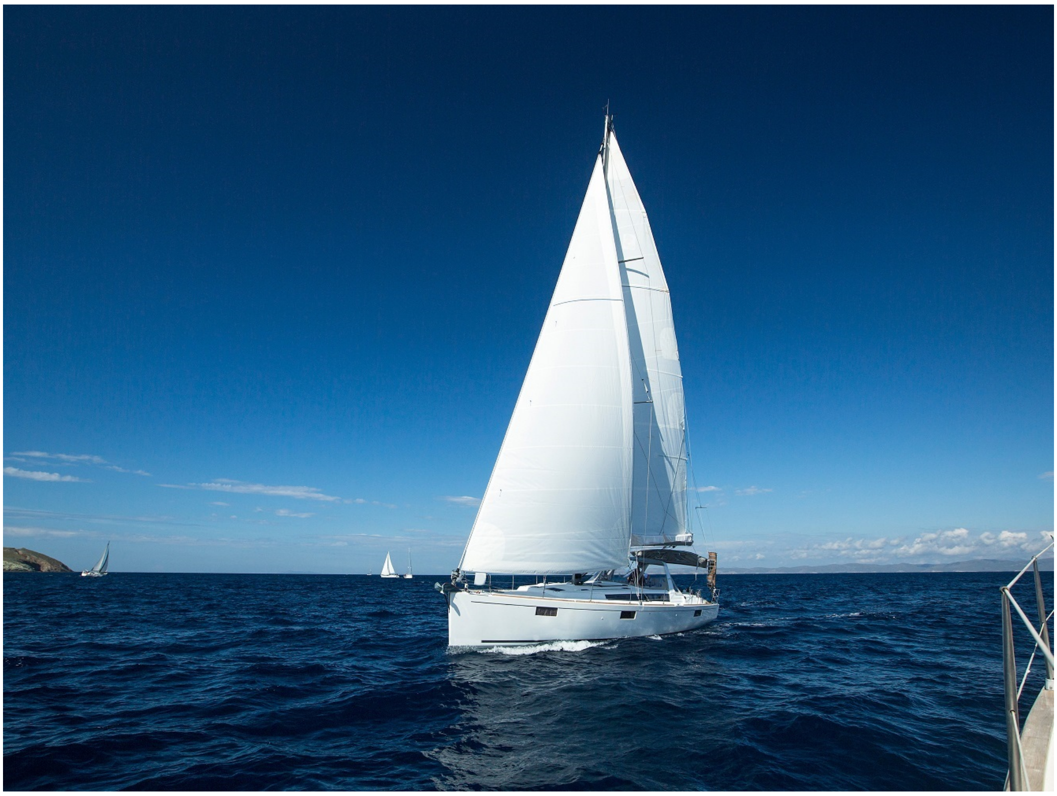
Below gives an example of querying the average RGB value of the **example.jpg** image:

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/average-hue

If information similar to the following is displayed, the average RGB value of the image is obtained:

```
{"RGB":"#28577D"}
```

Original **example.jpg** image:



Color corresponding to the average RGB value (#28577D):

7 Setting Image Effects

7.1 Brightness

You can use the GUI or code mode on OBS Console or make an API call to adjust the brightness of an image.

Table 7-1 describes the parameters.

This operation is represented by **bright**.

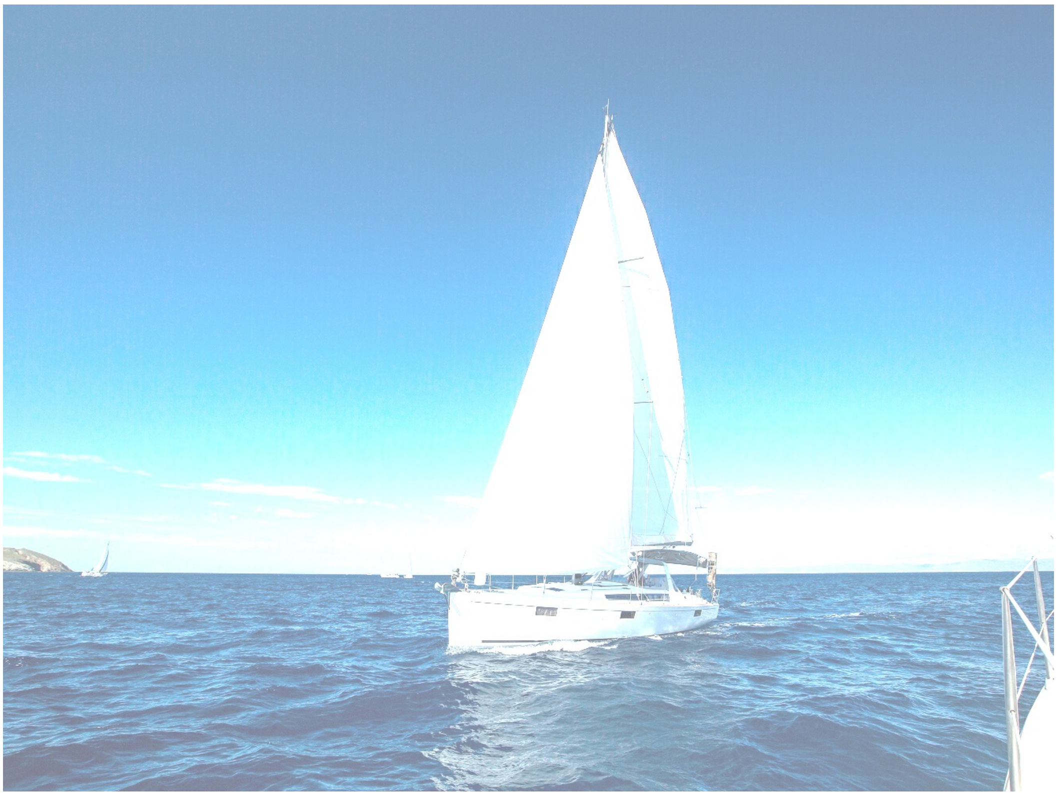
Table 7-1 Brightness description

Parameter	Value Description	Code Example
value	It adjusts the brightness of an image and ranges from **-100** to **100**. The original brightness value is **0**. A larger value means a higher level of brightness.	image/bright,50

Examples

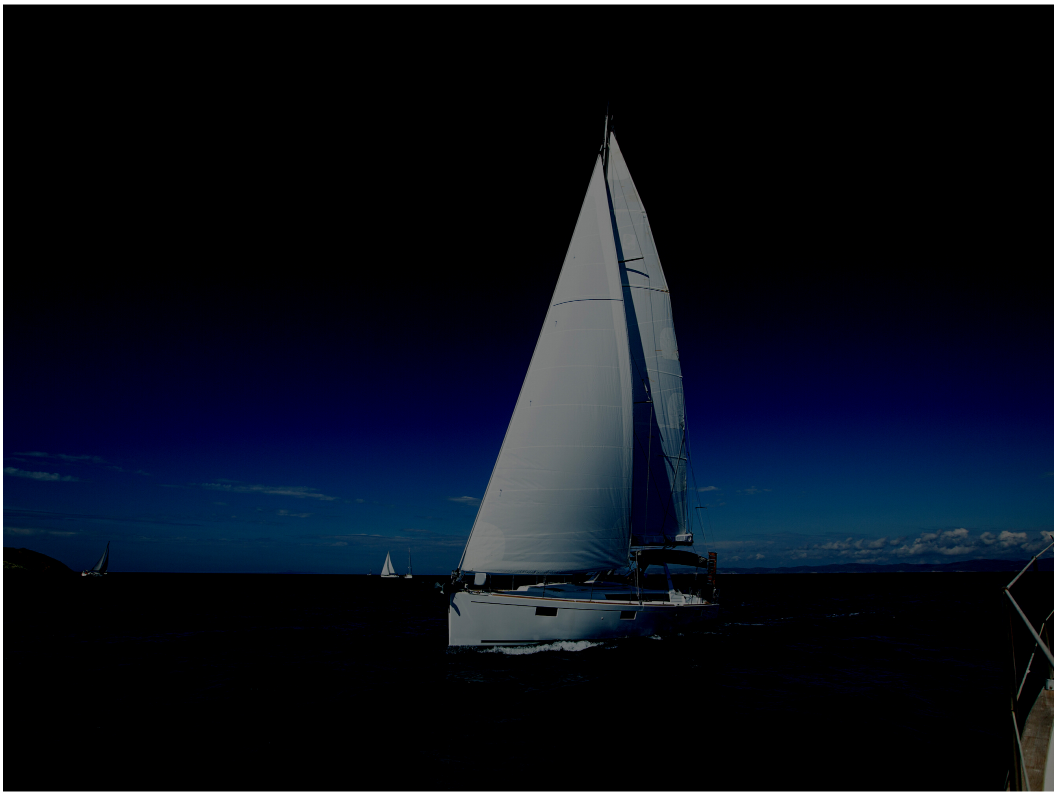
- Set the brightness to **50**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/bright,50



- Set the brightness to **-50**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/bright,-50



7.2 Contrast

You can use the GUI or code mode on OBS Console or make an API call to adjust the contrast of an image.

Table 7-2 describes the parameters.

This operation is represented by **contrast**.

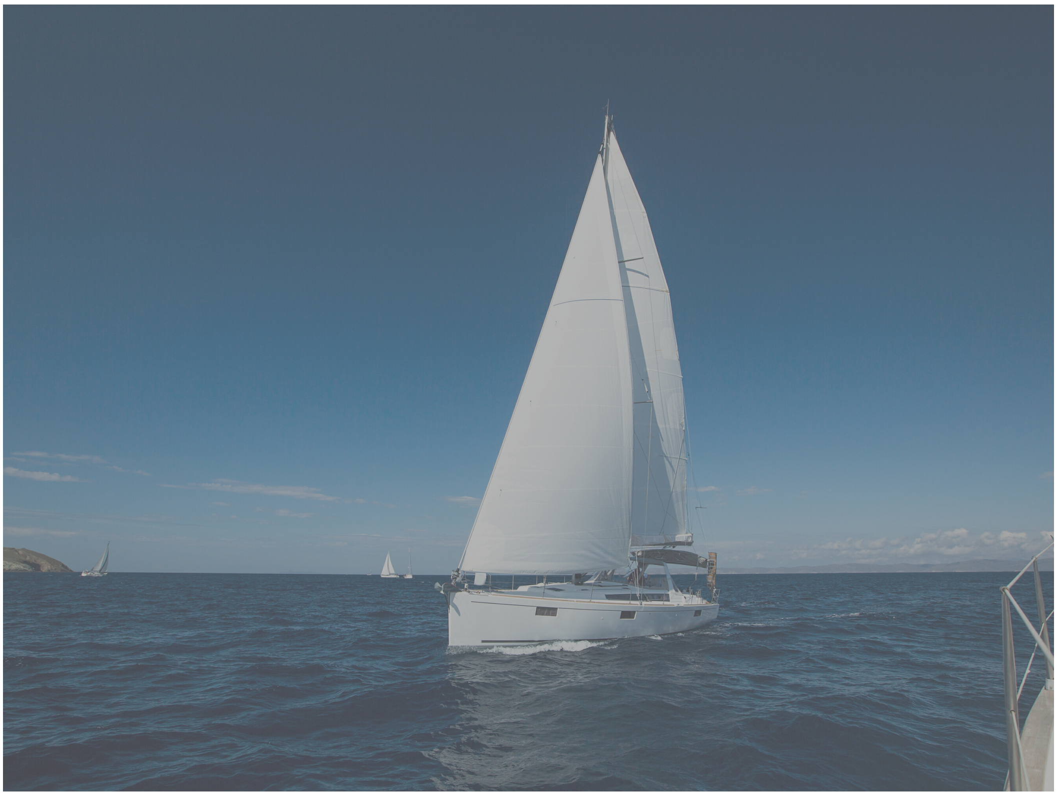
Table 7-2 Contrast description

Parameter	Value Description	Code Example
value	It adjusts the contrast of an image and ranges from **-100** to **100**. The original contrast value is **0**. A larger value means a higher level of contrast.	image/contrast,-50

Example

- Set the contrast to **-50**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/contrast,-50



7.3 Sharpness

You can use the GUI or code mode on OBS Console or make an API call to adjust the sharpness of an image.

Table 7-3 describes the parameters.

This operation is represented by **sharpen**.

Table 7-3 Sharpness description

Parameter	Value Description	Code Example
value	It adjusts the sharpness of an image and ranges from **50** to **399**. The recommended value is **100**. Technically, a larger value means a clearer image, but a too large value will make the image grainy.	image/sharpen,100

Example

- Set the sharpness to **100**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/sharpen,100

7.4 Blur

You can use the GUI or code mode on OBS Console or make an API call to adjust the blur of an image.

Table 7-4 describes the parameters.

This operation is represented by **blur**.

Table 7-4 Blur parameters

Parameter	Value Description	Code Example
r	It represents the blur radius. Its value ranges from **1** to **50**. A larger radius means a larger blurred area.	image/blur,r_3,s_2
s	It represents the standard deviation of normal distribution. Its value ranges from **1** to **50**. A large standard deviation significantly blurs, while a small standard deviation blurs less.	

NOTE

In GUI mode, the parameter **r** and **s** increase or decrease simultaneously.

Example

- Set **r** to **3** and **s** to **2**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/blur,r_3,s_2

8 Resizing Images

You can use the GUI or code mode on OBS Console or make an API call to resize images. Images can be resized based on a specific rule or a fixed width, height, or percentage.

NOTE

- A long side refers to the side with a larger ratio of its original size to its target size, and a short side refers to the side with a smaller ratio. Assume that the original size of an image is 400 × 200 pixels and it is resized to 100 × 100 pixels. The ratio of 400 pixels to 100 pixels is 4 and that of 200 pixels to 100 pixels is 2, so the long side is 400 pixels and the short side is 200 pixels.
- For a target image after resizing, its long side cannot exceed 9,999 pixels, and the product of its width and height cannot exceed 24,999,999 pixels.
- If you only specify the height or width for resizing, the target image keeps the same aspect ratio and format as the original image.
- By default, resizing is not allowed to scale up an image. If you want an image to become larger after resizing, you need to set **limit** to **0** to obtain the enlarged image, or the original image will be returned. To do this, use the following format:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_500,limit_0

Table 8-1 describes the parameters.

This operation is represented by **resize**.

Table 8-1 Resizing parameters

Parameter	Value Description	Code Example
m	Type of resizing. The value can be **lfit** (the default value), **mfit**, **fill**, **pad**, or **fixed**. ● **lfit**: Specify a rectangle with a given width (indicated by **w**) and height (indicated by **h**), lock the aspect ratio, and obtain the largest image in the rectangle. ● **mfit**: Specify a rectangle with a given width (indicated by **w**) and height (indicated by **h**), lock the aspect ratio, and obtain the smallest image in the rectangle's extended area. ● **fill**: Specify a rectangle with a given width (indicated by **w**) and height (indicated by **h**) and lock the aspect ratio. Obtain the smallest image in the rectangle's extended area, and center and crop the image. **fill**-based resizing actually centers and crops a target image resized with **mfit**. ● **pad**: Specify a rectangle with a given width (indicated by **w**) and height (indicated by **h**) and lock the aspect ratio. Obtain the largest image in the rectangle and fill the blank area with color. **pad**-based resizing actually fills the blank area of a target image resized with **lfit**. ● **fixed**: Resize an image based on a fixed width and height. ● **ratio**: Specify an aspect ratio (a ratio of **w** to **h**), in the range of **1** to **1000**, and	image/resize,m_lfit,h_100,w_100

Parameter	Value Description	Code Example
	obtain the largest image that meets the specified ratio.	
p	Percentage of the aspect ratio, in the range of **1** to **1000**. If the value is: ● < 100: The image is scaled down. ● = 100: The image is kept unchanged in size. ● > 100: The image is scaled up.	image/resize,p_50
h	Height of the target image, in the range of **1** to **9999**. The product of the target image's width and height cannot exceed 24,999,999.	image/ resize,m_lfit,h_100
w	Width of the target image, in the range of **1** to **9999**. The product of the target image's width and height cannot exceed 24,999,999.	image/ resize,m_fixed,h_100,w _100
l	The long side of the target image, in the range of **1** to **4096**. The product of the target image's width and height cannot exceed 24,999,999. The long side has a specified value, and the short side is scaled based on the ratio.	image/resize,l_100
s	The short side of the target image, in the range of **1** to **4096**. The product of the target image's width and height cannot exceed 24,999,999. The short side has a specified value, and the long side is scaled based on the ratio.	image/resize,s_100

Parameter	Value Description	Code Example
color	Color for filling the blank area after resizing. This parameter can be used when you set **m** to **pad**. The value is a hexadecimal code, from **000000** to **FFFFFF** (default value, representing white).	image/resize,m_pad,h_100,w_100,color_FF0000
limit	Whether to limit the size of the target image when the target image is larger than the original one. The value can be **0** or **1** (default value). ● **0**: The size is not limited. ● **1**: The size is limited.	image/resize,p_150,limit_0

⚠ CAUTION

If a resized image is aliased, you can add **/marker,u_plus** to the end of the image processing URL for optimization.

For example, by adding **/marker,u_plus**, the processing URL **https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_fixed,w_2668,h_1999,limit_0** becomes **https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_fixed,w_2668,h_1999,limit_0/marker,u_plus**. The latter displays an image with better quality.

Examples

- Set **h** to **100** and **m** to **lfit** (the default value) to process the width proportionally.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_lfit,h_100

- Lock the aspect ratio and specify the short side to resize the image into 100 x 100 pixels.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_lfit,h_100,w_100

- Set the long side size to **100** and scale the short side based on the ratio.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,l_100

- Fix the width and height, center and crop the image, resize the image into 100 x 100 pixels.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_fill,h_100,w_100

- Fix both the width and height to **100**.

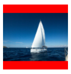
  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_fixed,h_100,w_100

- Fix the width and height. Resize the image into 100 x 100 pixels by specifying the short side and fill the blank area with white.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_pad,h_100,w_100

- Fix the width and height. Resize the image into 100 x 100 pixels by specifying the short side and fill the blank area with red.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_pad,h_100,w_100,color_FF0000

- Scale up the image to 150% of its original size and set **limit** to **0** to obtain the enlarged image.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,p_150,limit_0

- Set **p** to **30** to scale down the image to 30% of its original size.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,p_30

- Set an image's aspect ratio to 3:2.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,m_ratio,w_3,h_2

9 Rotating Images

9.1 Rotation Settings

You can edit code on OBS Console or make an API call to rotate images clockwise.

Table 9-1 describes the parameters.

This operation is represented by **rotate**.

Table 9-1 Rotation description

Parameter	Value Description	Code Example
value	It defines the angle of a clockwise rotation and ranges from **0** to **360**. The default value is **0**, which means the image is not rotated. A larger value means the image is clockwise rotated by a larger angle.	image/rotate,90

NOTE

- After rotation, dimensions of an image may increase.

Examples

Set the width to **100** and the rotation angle to **90**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100/rotate,90

Set the width to **100** and the rotation angle to **220**.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100/rotate,220

9.2 Adaptive Orientation

You can use the GUI or code mode on OBS Console or make an API call to use adaptive orientation.

Images shot by cameras or smartphones may contain **EXIF Data**, like an orientation parameter. Since an image's rotation information is recorded in the orientation parameter, a browser can automatically rotate the image to the right position based on the orientation parameter.

With adaptive orientation configured, images that contain the orientation parameter will be rotated automatically. For details, see **Table 9-2**.

This operation is represented by **auto-orient**.

Table 9-2 Adaptive orientation description

Parameter	Value Description	Code Example
value	The value can be **0** or **1** (default value). **0**: Adaptive orientation is not configured. The image will not rotate automatically and will keep the default orientation. **1**: Adaptive orientation is configured. The image will rotate automatically before being resized.	image/resize,w_100/auto-orient,0

NOTE

- The adaptive orientation can be configured only when the height and width of an image are both smaller than 4,096.
- If an image's EXIF data does not contain any rotation parameters or the image has no EXIF data at all, the **auto-orient** parameter does not work for the image.

Example

- Set the width of an image to **100**, and do not configure the adaptive orientation.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100/auto-orient,0

10 Cropping Images

10.1 Common Cropping

You can edit code on OBS Console or make an API call to crop images.

You can start at any point on an image and crop the image into a rectangle with specified width and height. Table 10-1 describes the parameters.

This operation is represented by **crop**.

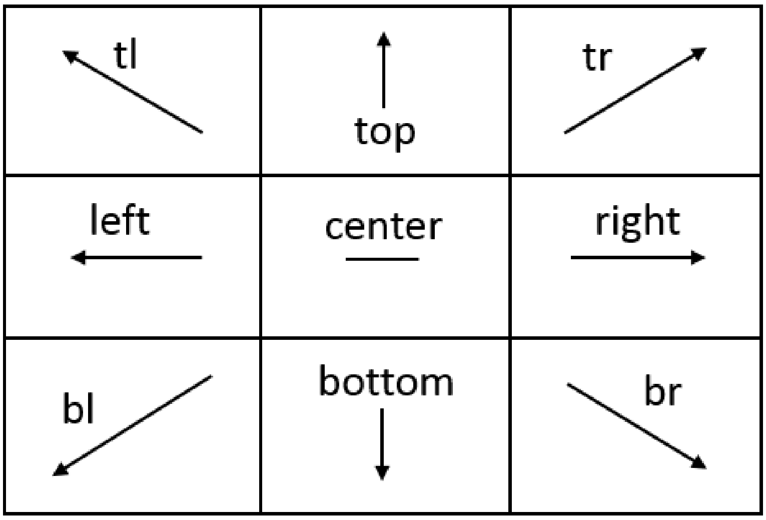
Table 10-1 Common cropping parameters

Parameter	Value Description	Code Example
g	The location where cropping starts. **g** can be set to **tl**, **top**, **tr**, **left**, **center**, **right**, **bl**, **bottom**, or **br**. Figure 10-1 shows a 3x3 grid illustrating these values. Each value locates at the top left corner of the cell.	image/crop,x_10,y_10,w_200,h_200,g_br
h	Cropping height. Its value ranges from **0** to *Original height*.	
w	Cropping width. Its value ranges from **0** to *Original width*.	
x	x-coordinate of the start point. The top left corner is the default origin. The parameter value ranges from **0** to *Original image width*.	

Parameter	Value Description	Code Example
y	y-coordinate of the start point. The top left corner is the default origin. The parameter value ranges from **0** to *Original image height*.	

Figure 10-1 shows the cropping origins.

Figure 10-1 3x3 grid of cropping origins



NOTE

- If x is larger than the origin width, or y is larger than the origin height, the cropping cannot be executed and a fault will be returned.
- If h is larger than the origin height and w is larger than the original width, the image will be cropped to the boundaries.

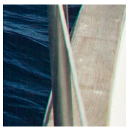
Examples

- Set the start point of cropping to **(1000, 500)**, and set both the width and height to **1000**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/crop,x_1000,y_500,w_1000,h_1000

- Set the start point of cropping to **(10, 10)** in the bottom right (**br**) cell, and set both the width and height to **200**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/crop,x_10,y_10,w_200,h_200,g_br



10.2 Inscribed Circles

You can edit code on OBS Console or make an API call to get an inscribed circle of an image.

Choose the image center as the center of a circle, and crop the image according to the specified radius. The image then is cropped into a circle. **Table 10-2** describes the parameters.

This operation is represented by **circle**.

Table 10-2 Inscribed circle parameters

Parameter	Value Description	Code Example
r	Radius specified for the inscribed circle. Its value ranges from **0** to *Half of the image's short side*.	image/circle,r_100

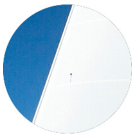
NOTE

- If the output format is set to JPG, the area outside the inscribed circle is white. If the output format is set to a vector format such as PNG, WebP, or BMP, the area outside the inscribed circle is transparent. PNG is the recommended output format.
- Even if **r** is larger than half of the short side, the generated inscribed circle is still the image's largest inscribed circle (with a radius equal to half of the short side).

Example

Set the radius to **100** and output format to **jpg** (with the area outside the inscribed circle being white).

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/circle,r_100



10.3 Indexcropping

You can edit code on OBS Console or make an API call to indexcrop images.

Set the top left corner of the image as the starting point. Set x-axis overlapping with the width, and y-axis overlapping with the height. Crop the image into several consecutive partitions horizontally or vertically, each having equal width or height. Get the partition you want according to the index. **Table 10-3** describes the parameters.

This operation is represented by **indexcrop**.

Table 10-3 Indexcropping parameters

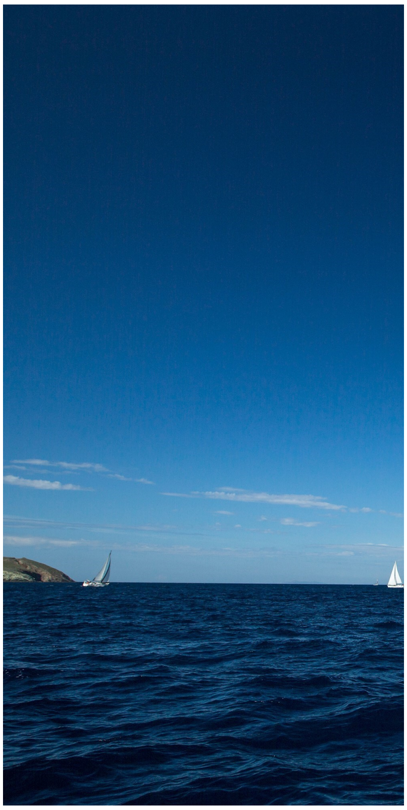
Parameter	Value Description	Code Example
x	Width of each partition in horizontal cropping, ranging from **1** to *Original image width*. **x** and **y** cannot be chosen at the same time.	image/ indexcrop,x_1000,i_0

Parameter	Value Description	Code Example
y	Height of each partition in vertical cropping, ranging from **1** to *Original image height*. **x** and **y** cannot be chosen at the same time.	
i	If there are *n* partitions in total, the value of **i** ranges from **0** to *n*–*1*. When **i** is set to **0**, you obtain the first partition. If you set **i** to a value larger than *n*–*1*, the original image will be returned.	

Examples

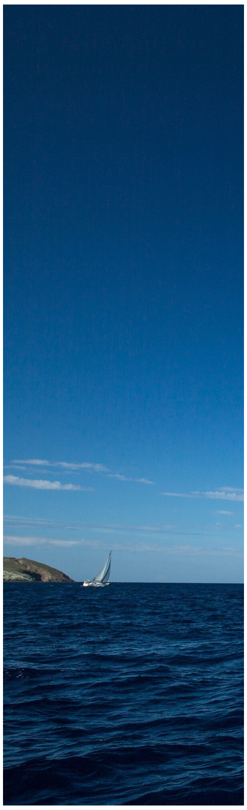
- Indexcrop an image into partitions with a width of **1000** horizontally and choose the first partition.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/indexcrop,x_1000,i_0



- Indexcrop an image into partitions with a width of **600** horizontally and choose the first partition.

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/indexcrop,x_600,i_0



10.4 Rounded Corner Cropping

You can edit code on OBS Console or make an API call to round corners of an image.

This operation allows you to cut off the corners of an image based on the specified radiuses. You can configure the radius of image corners to round in two ways. For details, see **Table 10-4**.

This operation is represented by **rounded-corners**.

Table 10-4 Rounded corner parameters

Parameter	Value Description	Code Example
r	It indicates both horizontal and vertical radiuses of image corners you will round. You can configure it by using the number of pixels (for example, **200**) or the percentage (for example, **25p**). The number of pixels ranges from **1** to **4096**. If this number is greater than half of the original image's short side, the half of the short side is applied. The percentage ranges from **1p** to **50p**. **r** cannot be used together with **rx** or **ry**.	image/rounded-corners,r_100
rx	It indicates the horizontal radius of image corners you will round. You can configure it by using the number of pixels (for example, **200**) or the percentage (for example, **25p**). The number of pixels ranges from **1** to **4096**. If this number is greater than half of the original image's short side, the half of the short side is applied. The percentage ranges from **1p** to **50p**. **rx** must be used together with **ry**.	image/rounded-corners,rx_100,ry_200
ry	It indicates the vertical radius of image corners you will round. You can configure it by using the number of pixels (for example, **200**) or the percentage (for example, **25p**). The number of pixels ranges from **1** to **4096**. If this number is greater than half of the original image's short side, the half of the short side is applied. The percentage ranges from **1p** to **50p**. **ry** must be used together with **rx**.	

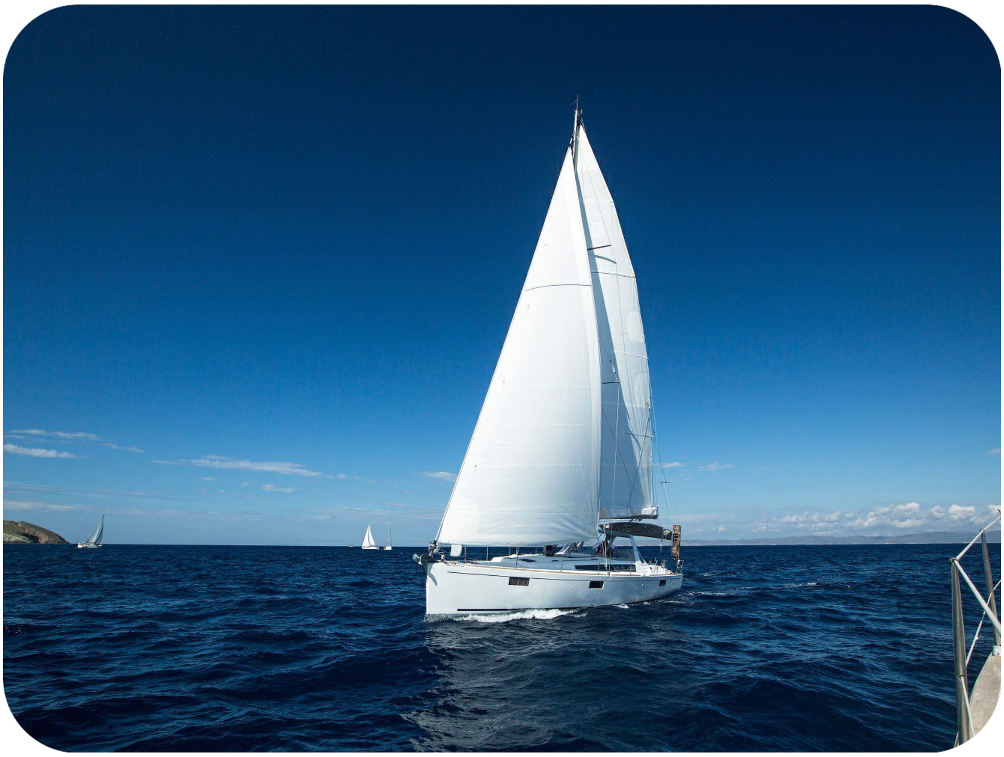
NOTE

If the output image format is JPG, the cut-out corner area is white. If the output image format is PNG, WebP, or BMP, the cut-out corner area is transparent. You are advised to save rounded corner images in PNG.

Examples

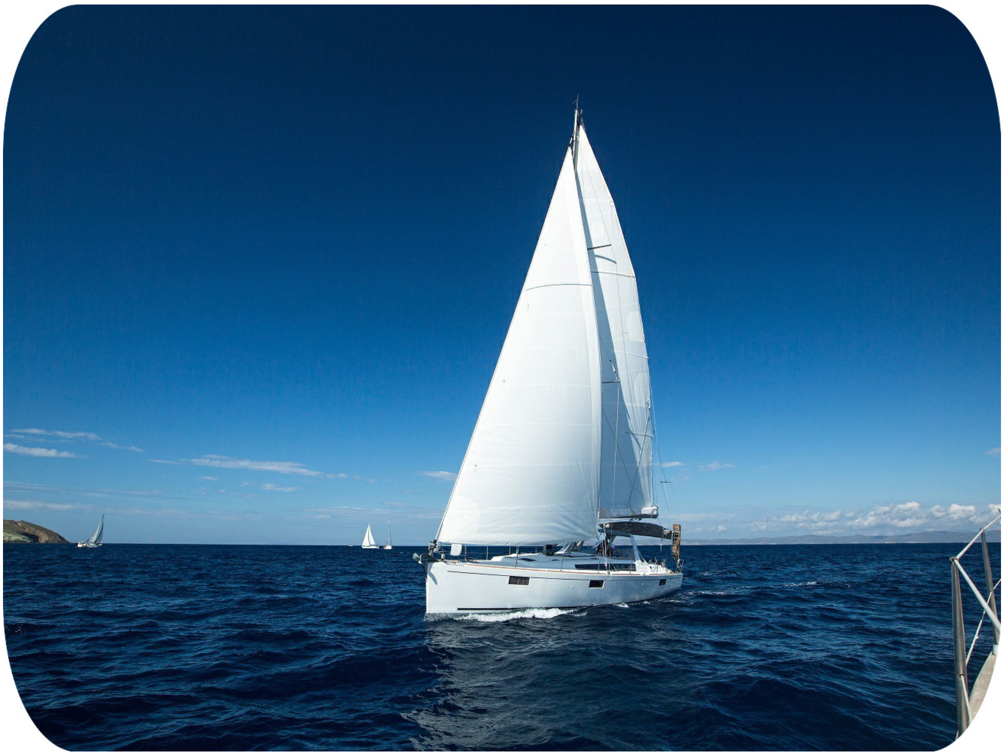
- Round the corners of image **example.jpg** based on the horizontal and vertical radiuses of **100** and save the new image as PNG.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/rounded-corners,r_100/format,png

- Round the corners of image **example.jpg** based a horizontal radius of **100** and a vertical radius of **200**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/rounded-corners,rx_100,ry_200



11 Watermarking Images

11.1 Public Parameters

You can use the GUI or code mode on OBS Console or make an API call to configure public parameters for watermarking.

You can add a text or image watermark to the original image.

The Base64 code for URL transmission applies to paths of content and fonts of text watermarks, or paths of image watermarks. It is not advisable to put standard Base64 code directly into the URL for transmission. In Base64 encoding for URL transmission, contents are coded into character strings by standard Base64 code. After verifying these strings, replace the plus sign (+) with hyphen (-), and slash (/) with underline (_). For details about encoding, see those specified in RFC4648.

This operation is represented by **watermark**.

Public parameters are those applicable to both image watermarks and text watermarks. You can add these two types of watermarks to the same image. **Table 11-1** describes the parameters.

Table 11-1 Public parameters

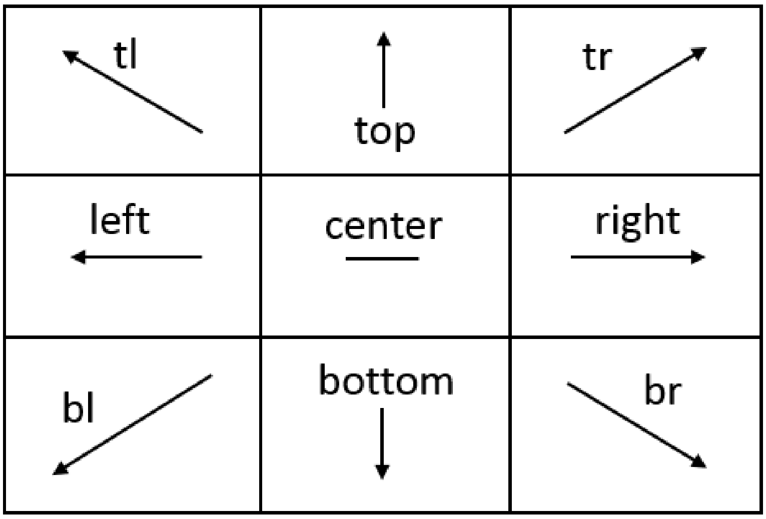
Parameter	Value Description	Code Example
g	Optional. **g** represents the watermark location. It can be set to **tl**, **top**, **tr**, **left**, **center**, **right**, **bl**, **bottom**, or **br**. The default value is **tl**. **Figure 11-1** shows a 3x3 grid illustrating these values. Each value locates at the top left corner of the cell.	image/watermark,image_aW1hZ2UtZGVtby 9sb2dvLnBuZw==,g_br,t_90,x_10,y_10

Parameter	Value Description	Code Example
x	Optional. **x** represents the horizontal distance from the image edge. By default, the origin is at the top left corner. The value ranges from **0** to **4096**, in pixels. The default value is **10**.	
y	Optional. **y** represents the vertical distance from the image edge. By default, the origin is at the top left corner. The value ranges from **0** to **4096**, in pixels. The default value is **10**.	
voffset	Optional. **voffset** represents the vertical offset from the horizontal centerline of the image. It moves a watermark upward or downward from the horizontal centerline of the image. The value ranges from **-1000** to **1000**, in pixels. The default value is **0**. **voffset** is meaningful only when **g** is set to **left**, **center**, or **right**.	
align	Optional. It represents the alignment of a watermark. The value can be **0** (default value), **1**, or **2**. ● **0**: top alignment ● **1**: center alignment ● **2**: bottom alignment	
order	Optional. It represents the placement order of the text and image watermarks. The value can be **0** (default value) or **1**. ● **0**: image watermark in front of the text watermark ● **1**: text watermark in front of the image watermark	
t	Optional. It represents the transparency of a watermark. Its value ranges from **0** to **100**. The default value is **100**, indicating the watermark is not transparent at all.	

Parameter	Value Description	Code Example
interval	Optional. It represents the distance between a text watermark and an image watermark. Its value ranges from **0** to **1000**.	

Figure 11-1 shows a 3x3 grid illustrating the watermark locations.

Figure 11-1 3x3 grid of watermark locations



NOTE

If both a text watermark and an image watermark are added to the same image, you can configure **x**, **y**, and **voffset** to adjust the watermark locations and their layout.

11.2 Image Watermarks

You can pre-process an image watermark before adding it to an image.

The pre-processing operations include those described in **Resizing Images**, **Rotating Images**, and **Cropping Images**, but cropping an image watermark into an inscribed circle is not allowed. When resizing an image watermark, you can scale it to a specific percentage of the original image size.

Table 11-2 describes the image watermark parameters.

Table 11-2 Image watermark parameters

Parameter	Value Description	Code Example
image	Mandatory. It indicates the watermark image path. The watermark image path is *bucketName***/***objectName* (encoding required) or *bucketName***/***objectName***?x-image-process=***image***/***command* (encoding required). **NOTICE** The content must be URL-safe Base64 encoded in the **encodedObject = url_safe_base64_encode(object)** format. For example, object **panda.png** will be encoded as **cGFuZGEucG5n**.	image/resize,w_400/watermark,image_aW1h Z2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb 2Nlc3M9aW1hZ2UvcmVzaXplLFBfMzA=,t_90, g_br,x_10,y_10
P	It indicates the percentage of the original image the image watermark will be scaled to. The value ranges from **1** to **100**. **NOTICE** **P** used here for resizing is case-sensitive. To adjust the size of an image watermark, see **Resizing Images** (except the **p** parameter).	image-demo/logo.png?x-image-process=image/resize,P_50

Examples of Making API Calls

- Use the watermark image **logo.png** (path: **hw-image-demo/logo.png**) and make the image watermark located at the bottom right, with a transparency of **90** and default horizontal and vertical margins of **10**.

  Relevant parameters:

  Watermark image: **hw-image-demo/logo.png**

  Safe Base64 code: **aHctaW1hZ2UtZGVtby9sb2dvLnBuZw==**

  Watermark position (bottom right): **g_br**

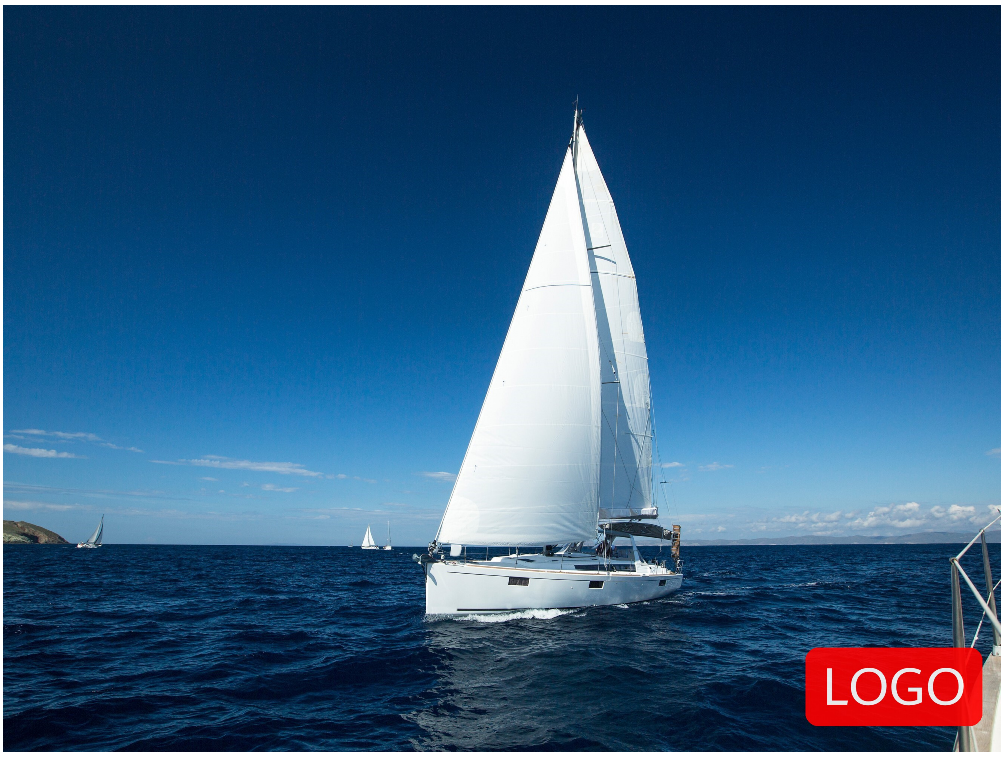
  Transparency: **t_90**

  Horizontal and vertical margins: **x_10,y_10**

  Request URL:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZw==,g_br,t_90,x_10,y_10

Figure 11-2 Example 1



- Use the watermark image **logo.png** (path: **hw-image-demo/logo.png**) and resize the image watermark by setting its width to **50**. Keep other parameters the same as those in the previous example.

  Relevant parameters:

  Watermark image: **hw-image-demo/logo.png?x-image-process=image/resize,w_50**

  Safe Base64 code: **aHctaW1hZ2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb2Nlc3M9aW1hZ2UvcmVzaXplLHdfNTA=**

  Watermark position (bottom right): **g_br**

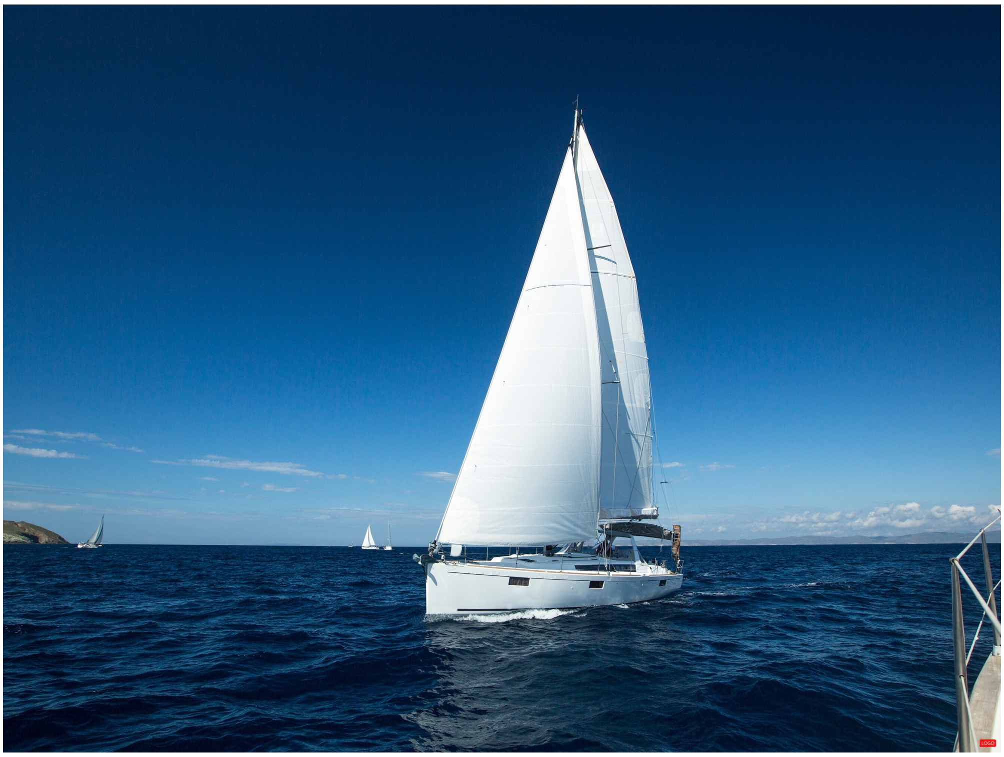
  Transparency: **t_90**

  Horizontal and vertical margins: **x_10,y_10**

  Request URL:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb2Nlc3M9aW1hZ2UvcmVzaXplLHdfNTA=,g_br,t_90,x_10,y_10

Figure 11-3 Example 2



- Use the watermark image **logo.png** (path: **hw-image-demo/logo.png**) and resize the image watermark to 50 percent of the original image's size. Keep other parameters the same as those in the previous example.

  Relevant parameters:

  Watermark image: **hw-image-demo/logo.png?x-image-process=image/resize,P_50**

  Safe Base64 code: **aHctaW1hZ2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfNTA=**

  Watermark position (bottom right): **g_br**

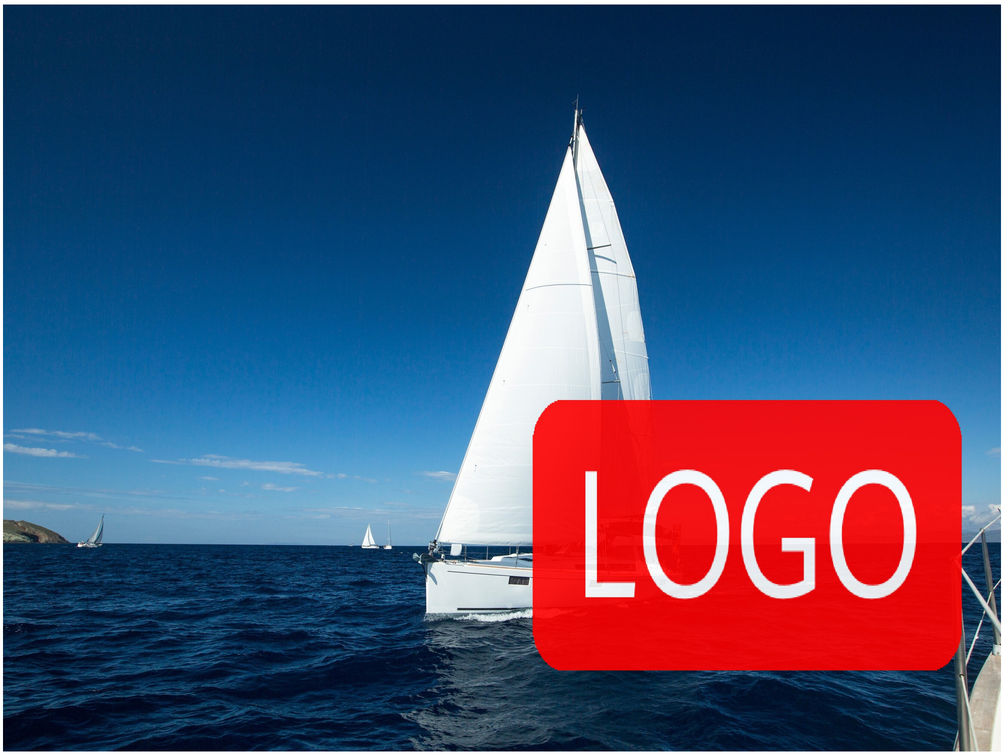
  Transparency: **t_90**

  Horizontal and vertical margins: **x_10,y_10**

  Request URL:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZz94LWltYWdlLXByb2Nlc3M9aW1hZ2UvcmVzaXplLFBfNTA=,g_br,t_90,x_10,y_10

Figure 11-4 Example 3



Example of Using GUI

To add an image watermark to an image, you can create a watermark style on OBS Console and then apply it to the image. The following gives an example of creating a watermark style on OBS Console:

Step 1 Log in to OBS Console and click the required bucket name. In the navigation pane, choose **Data Processing** > **Image Processing**.

Step 2 Click **Create** to go to the style editing page

Step 3 On the editing page, specify a style name, choose **GUI** for **Edit Mode**, and click **Watermark**.

NOTE

A style name consists of letters (uppercase or lowercase), digits, periods (.), underscores (_), and hyphens (-) and contains 1 to 256 characters, for example, **rotate_0001**.

Step 4 Configure the following watermark parameters. For details, see **Figure 11-5**.

- **Watermark Type**: Choose **Image watermark**.
- **Watermark Image Path**: Enter **e-share/image-demo/logo.png**.
- **Image Size (%)**: Set this parameter based on whether the watermark image is zoomed out.
- **Brightness**: Retain the default value **0**.
- **Contrast**: Retain the default value **0**.
- **Transparency**: Set this parameter to **90**.

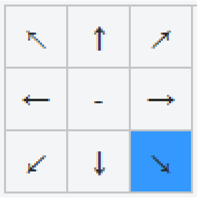
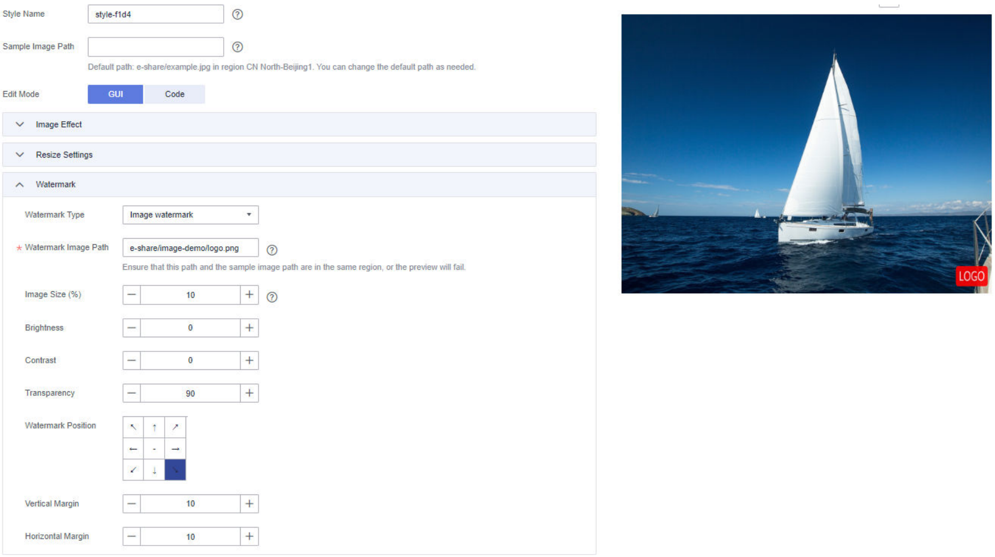
- **Watermark Position**: Select the arrow at the bottom right, as shown in .
- **Vertical Margin**: Retain the default value **10**.
- **Horizontal Margin**: Retain the default value **10**.

Figure 11-5 Configuring an image watermark



Style Name: style-f1d4

Sample Image Path:

Default path: e-share/example.jpg in region CN North-Beijing1. You can change the default path as needed.

Edit Mode: GUI | Code

Image Effect

Resize Settings

Watermark

Watermark Type: Image watermark

* Watermark Image Path: e-share/image-demo/logo.png

Ensure that this path and the sample image path are in the same region, or the preview will fail.

Image Size (%): 10

Brightness: 0

Contrast: 0

Transparency: 90

Watermark Position

Vertical Margin: 10

Horizontal Margin: 10

Step 5 Click **OK** to save the style. The new style will be displayed in the style list.

You can use the new watermark style to process images by referring to **Using the Image Style**.

----End

11.3 Text Watermarks

You can configure the size, font, and color for text images. **Table 11-3** describes the parameters.

Table 11-3 Text watermark parameters

Parameter	Value Description	Code Example
text	Mandatory for adding text watermarks **NOTICE** The content must be URL-safe Base64 encoded in the **encodeText = url_safe_base64_encode(fontText)** format, with a maximum of 64 characters (16 full-width characters at most).	image/watermark,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h1c29uZw==,g_center,rotate_30
size	Optional. It represents the size of a text watermark, ranging from **0** (exclusive) to **1000** (inclusive). The default value is **40**.	
type	Optional. It represents the font of a text watermark. Available values are those shown in **Table 11-4**. The default value is **wqy-zenhei** (**d3F5LXplbmhlaQ** after encoding). **NOTICE** - The content must be URL-safe Base64 encoded in the **encodeText = url_safe_base64_encode(fontType)** format. - Line breaks are currently not supported.	
color	Optional. It represents the color of a text watermark. The value is a six-digit hexadecimal code, from **000000** (default value, representing black) to **FFFFFF**.	
shadow	Optional. It represents the transparency of a text watermark. The value ranges from **0** (exclusive) to **100** (inclusive).	
fill	Optional. It represents the filling effect of a text watermark. The value can be **0** or **1**. ● **0**: filling not applied ● **1**: filling applied	
rotate	Optional. It represents the clockwise rotation angle of a text watermark. The value ranges from **0** (exclusive) to **360** (exclusive).	

Table 11-4 Text font encoding

Parameter	Value After URL-Safe Base64 Encoding	Value Description	Remarks
droidsansfallback	ZHJvaWRzYW5zZmFsbGJhY2s=	DroidSansFallback	According to Request for Comments (RFC), the fuller "=" can be omitted, and the value becomes **ZHJvaWRzYW5zZmFsbGJhY2s**.
fangzhengfangsong	ZmFuZ3poZW5nZmFuZ3Nvbmc=	FZFongSong	According to RFC, the fuller "=" can be omitted, and the value becomes **ZmFuZ3poZW5nZmFuZ3Nvbmc**.
fangzhengheiti	ZmFuZ3poZW5naGVpdGk=	FZSimHei	According to RFC, the fuller "=" can be omitted, and the value becomes **ZmFuZ3poZW5naGVpdGk**.
fangzhengkaiti	ZmFuZ3poZW5na2FpdGk=	FZKaiTi	According to RFC, the fuller "=" can be omitted, and the value becomes **ZmFuZ3poZW5na2FpdGk**.
fangzhengshusong	ZmFuZ3poZW5nc2h1c29uZw==	FZShuSong	According to RFC, the fuller "=" can be omitted, and the value becomes **ZmFuZ3poZW5nc2h1c29uZw**.
wqy-microhei	d3F5LW1pY3JvaGVp	WenQuanYi Micro Hei	-
wqy-zenhei	d3F5LXplbmhlaQ==	WenQuanYi Zen Hei	According to RFC, the fuller "=" can be omitted, and the value becomes **d3F5LXplbmhlaQ**.

Examples of Making API Calls

- Add the text watermark **Hello** to an image, with the text size set to **60**, color to red, and font to FZShuSong.

  Relevant parameters:

  Safe Base64 code: **SGVsbG8g5Zu**, namely **text_SGVsbG8g5Zu**

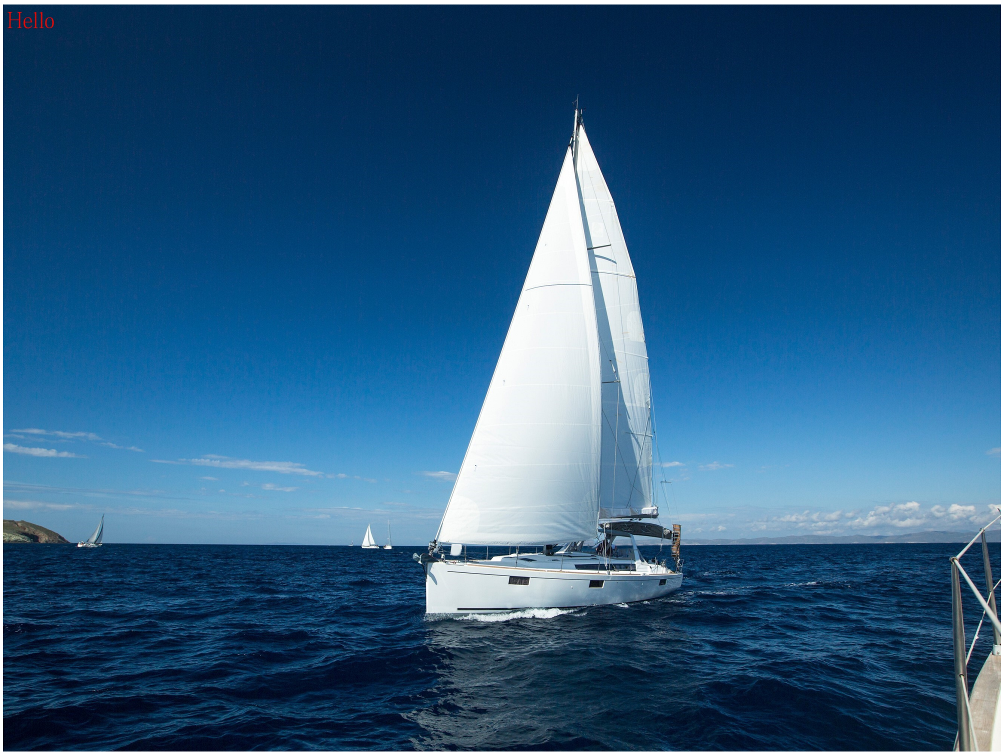
  Text size: **size_60**

  Text color: **color_FF0000**

  Text font: **type_ZmFuZ3poZW5nc2h1c29uZw==**

  Request URL:

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h1c29uZw==



- Clockwise rotate a text watermark through an angle of 30 degrees and add it to the center of an image. Keep other parameters the same as those in the previous example.

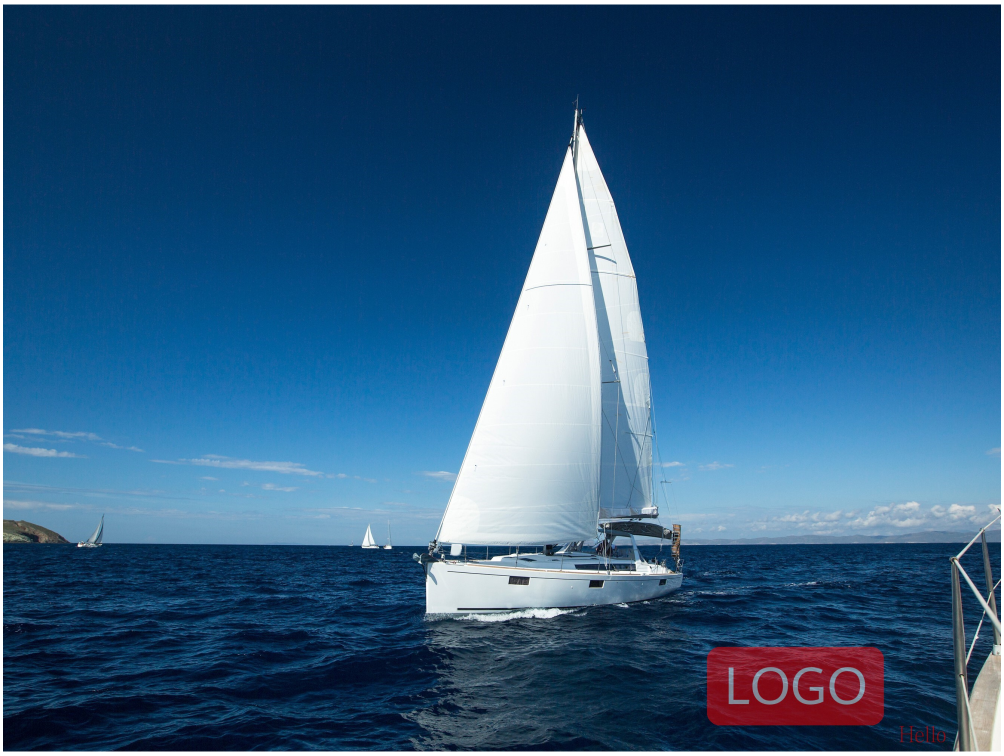
  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h1c29uZw==,g_center,rotate_30

- Add image and text watermarks at the same time. Add text watermark **Hello** at the bottom right, with the size set to **60**, shadow to **50**, and color to red.

  Add image watermark **logo.png**, with both horizontal and vertical margins set to **10**.

  Set the transparency of both text and image watermarks to **50** and put the image watermark in front and the text watermark behind, in the bottom alignment.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZw==,text_SGVsbG8g5Zu,size_60,color_FF0000,shadow_50,type_ZmFuZ3poZW5nc2h1c29uZw==,g_br,x_10,y_10,align_2,order_0

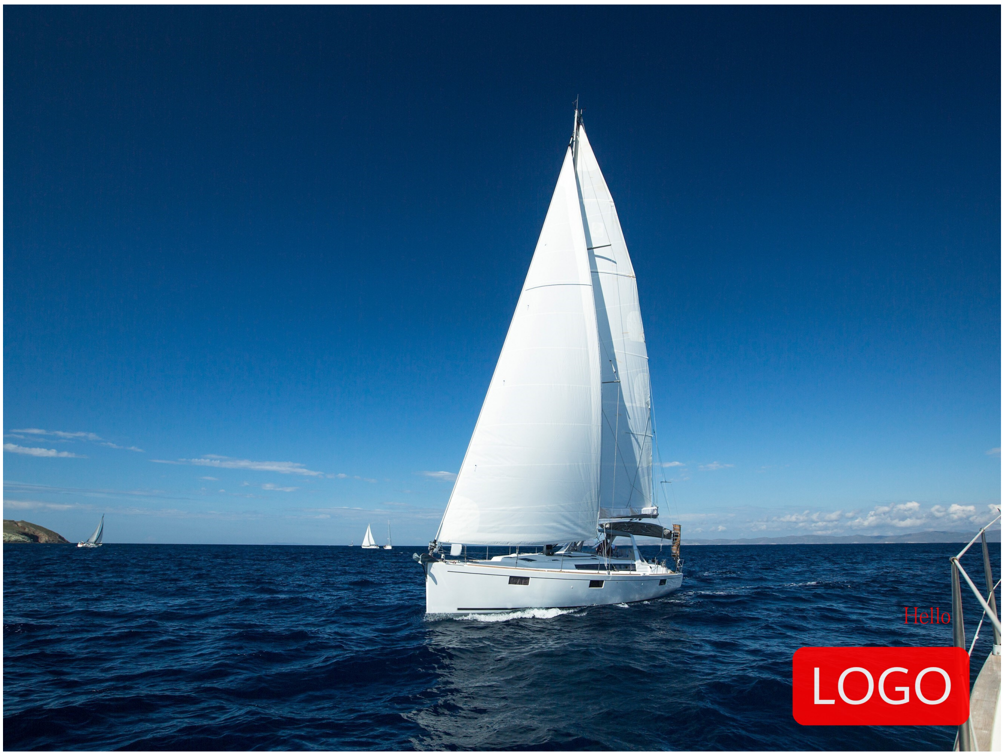


- Add an image watermark first and then a text watermark. Add text watermark **Hello** at the bottom right, with the size set to **60** and color to red.

  Add image watermark **logo.png**, with a horizontal margin of **40** and a vertical margin of **10**.

  Set the transparency of the image watermark to **90**.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/watermark,image_aHctaW1hZ2UtZGVtby9sb2dvLnBuZw==,g_br,t_90,x_45,y_10/watermark,text_SGVsbG8g5Zu,size_60,color_FF0000,type_ZmFuZ3poZW5nc2h1c29uZw==,g_br,x_0,y_330


Hello
LOGO

12 Converting Image Formats

12.1 Converting Formats

You can use the GUI or code mode on OBS Console or make an API call to convert images to different formats:

- Supported original formats: JPG, JPEG, PNG, BMP, WebP, GIF, and TIFF
- Supported target formats: JPG, PNG, BMP, and WebP (described in **Table 12-1**)

This operation is represented by **format**.

Table 12-1 Format conversion

Parameter	Value Description	Code Example
jpg	The image is saved in JPG format. If the original image is in vector format such as WebP, BMP, or PNG, the transparent part will be padded to white.	image/format,jpg
webp	The image is saved in WebP format.	image/format,webp
bmp	The image is saved in BMP format.	image/format,bmp
png	The image is saved in PNG format.	image/format,png

Example

- Save a JPG image as a PNG image.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/format,png

12.2 Interlaced Image Loading

You can edit code on OBS Console or make an API call to load images in an interlaced way.

With format conversion, images are generated in baseline JPEG format by default. If you want to generate an image in progressive JPEG, use the **interlace** parameter. **Table 12-2** describes the parameters.

- Display of baseline JPG images: from top to bottom
- Display of progressive JPEG images: from blurry to clearer and clearer

This operation is represented by **interlace**.

Table 12-2 Interlaced loading

Parameter	Value Description	Code Example
value	The value can be **0** or **1**. **0**: The output is a JPG image that is displayed from top to bottom. **1**: The output is a JPEG image in an interlaced display.	image/format,jpg/interlace,1

Examples

- Create a JPG image in an interlaced display.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/format,jpg/interlace,1

- Create a JPG image that is displayed from top to bottom.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/format,jpg/interlace,0

13 Changing Image Quality

You can edit code on OBS Console or make an API call to change the quality of an image.

To save space, you can compress JPG images. **Table 13-1** describes the parameters.

This operation is represented by **quality**.

Table 13-1 Image compression parameters

Parameter	Value Description	Code Example
q	Relative quality of the image. The image is compressed to q% of the original. The value ranges from **1** to **100**. Formula for compression: Target quality = Original quality x q% For example, if the original image quality is 100% and the relative quality is 80%, then the target image quality is 80%. If the original quality is 80% and the relative quality is 80%, then the target quality is 64%.	image/resize,w_100,h_100/quality,q_80

Parameter	Value Description	Code Example
Q	Absolute quality of the image. The image is compressed into Q%. Q is irrelevant to and does not depend on the original image. The value ranges from **1** to **100**. Formulas for compression: ● Original quality > Q%: target quality = Q% ● Original quality = Q%: target quality = original quality = Q% ● Original quality < Q%: target quality = original quality For example, if the original image quality is 100% and the absolute quality is 80%, then the target image quality is 80%. If the original quality is 70% and the absolute quality is 80%, then the target quality is 70%.	

NOTE

- **q** is valid only for JPG images.
- If both **q** and **Q** are used, the output is based on **Q**.
- **q** and **Q** are only valid for JPG images. For images in other formats, **q** and **Q** bring no effect and cause no impact.

Examples

- Resize the image by setting the height and width both to **100** and generate a JPG image with a relative quality of 80%.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100,h_100/quality,q_80

- Resize the image by setting the height and width both to **100** and generate a JPG image with an absolute quality of 80%.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100,h_100/quality,Q_80

14 Slimming Images

You can edit code on OBS Console or make an API call to slim images.

Slimming is a simplified image compression, which generates images with an absolute quality of 75%. You do not need to configure any parameter. You can slim images just by running a command.

This operation is represented by **imageslim**.

NOTE

- Only JPG images can be slimmed.
- It is recommended that you slim images after other image processing operations.

Example

- Resize an image by setting the width and height both to **100** and then slim the resized image.

  https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/resize,w_100,h_100/imageslim

15 Image Processing Persistency

Image processing persistency allows the images after processing to be asynchronously stored in a specified OBS bucket, so that you can directly access the stored images to save time.

You can use image processing persistency only by making API calls. In an image processing request, the persistency should be configured in the *Parameter name*=*Parameter value* format. **Table 15-1** describes the parameters.

Table 15-1 Image processing persistency parameters

Parameter	Value	Description
x-image-save-object	*objectName*	Mandatory It specifies the name of the processed image that will be stored in the bucket. An object name: ● cannot contain the following special characters: \ : * ? " < > | ● must be 1 to 1023 characters long.
x-image-save-bucket	*bucketName*	Optional It specifies the bucket that will store the processed image. If this parameter is not specified, the processed image will be saved in the bucket where the original image is processed. The bucket must be an existing one in OBS and its name must be 1 to 64 characters long.

Java Sample Code

```
ObsClient obsClient = null;
String endPoint = "obs-endpoint";  // Current region
// Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store them in the configuration file or environment variables.
// In this example, the AK and SK are stored in environment variables for identity authentication. Before running this example, configure environment variables ACCESS_KEY_ID and SECRET_ACCESS_KEY.
// Obtain an AK and SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca_01_0003.html.
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY");
try {
    ObsConfiguration config = new ObsConfiguration();
    config.setEndPoint(endPoint);
    obsClient = new ObsClient(ak,sk ,config);
    TemporarySignatureRequest request = new TemporarySignatureRequest();
    request.setObjectKey("test.jpeg");  // Original object name before processing
    Map<String, Object> queryParams = new HashMap<>();
    queryParams.put("x-image-process", "image/resize,w_100");
    String objectName = "your saves objectName";  // Name of the processed object
    // Optional parameters
    String bucketName = "your saves Bucket";  // Bucket that stores the processed object
    queryParams.put("x-image-save-object", ServiceUtils.toBase64(objectName.getBytes("UTF-8")));
    queryParams.put("x-image-save-bucket", ServiceUtils.toBase64(bucketName.getBytes("UTF-8")));
    request.setQueryParams(queryParams);
    request.setBucketName("your bucket");  // Bucket that stores the original object
    TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
    //URL to access
    response.getSignedUrl();
 } catch (Exception e) {
...//Handle exceptions.
 } finally {
     if (obsClient != null) {
         obsClient.close();
      }
 }
```

Python Sample Code

```
from obs import ObsClient
import os
import traceback
import requests

# Obtain an AK and SK pair using environment variables or import the AK and SK pair in other ways. Using hard coding may result in leakage.
# Obtain an AK and SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca_01_0003.html.
ak = os.getenv("AccessKeyID")
sk = os.getenv("SecretAccessKey")
# (Optional) If you use a temporary AK and SK pair and a security token to access OBS, obtain them from environment variables.
security_token = os.getenv("SecurityToken")
# Set server to the endpoint corresponding to the bucket. CN-Hong Kong (ap-southeast-1) is used here as an example. Replace it with the one in use.
server = "https://obs.ap-southeast-1.myhuaweicloud.com"
# Create an obsClient instance.
# If you use a temporary AK and SK pair and a security token to access OBS, you must specify security_token when creating an instance.
obsClient = ObsClient(access_key_id=ak, secret_access_key=sk, server=server)
try:
    # Generate a signed URL for image persistency.
    # Name of the bucket that stores the original object
    bucketName = 'originBucketName';
    # Original object name
    objectKey = 'test.png';

    # Name of the object after processing
    targetObjectName ="save.png"
    # (Optional) Name of the bucket that stores the new object
    targetBucketName ="saveBucketName"
    queryParams={}
    queryParams["x-image-process"]="image/resize,w_100"
    queryParams["x-image-save-object"]=base64.b64encode(targetObjectName .encode("utf-8")).decode()
    # Optional parameter
    queryParams["x-image-save-bucket"]=base64.b64encode(targetBucketName .encode("utf-8")).decode()
    res = obsClient.createSignedUrl(method='GET', bucketName=bucketName, objectKey=objectKey, queryParams=queryParams, expires=3600)
    print('signedUrl:', res.signedUrl)
    print('actualSignedRequestHeaders:', res.actualSignedRequestHeaders)
    // Make a GET request for image persistency.
    r = requests.get(resp.signedUrl)
    print(r)
except:
    print(traceback.format_exc())
```

Node.js Sample Code

```
// Import the OBS library.
const ObsClient = require('esdk-obs-nodejs');
const https = require('https');
const http = require('http');
const urlLib = require('url');

// Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store them in the configuration file or environment variables.
// In this example, the AK and SK are stored in environment variables for identity authentication. Before running this example, configure environment variables ACCESS_KEY_ID and SECRET_ACCESS_KEY.
// Obtain an AK and SK pair on the management console. For details, see https://support.huaweicloud.com/intl/en-us/usermanual-ca/ca_01_0003.html.
const ak = process.env.ACCESS_KEY_ID;
const sk = process.env.SECRET_ACCESS_KEY;
const server = "obs-endpoint";  // Current region

// Create an ObsClient instance.
const obsClient = new ObsClient({
    access_key_id: ak,
    secret_access_key: sk,
    server: server
});

// Name of the bucket that stores the original object
const bucketName = 'originBucketName';
// Original object name
const objectKey = 'test.png';
const method = 'GET';

// Name of the object after processing
const targetObjectName = "save.png";
// (Optional) Name of the bucket that stores the new object
const targetBucketName = 'saveBucketName';

const queryParams = {
    "x-image-process": "image/resize,w_100",
    "x-image-save-object": Buffer.from(targetObjectName, 'utf8').toString('base64'),
    // Optional parameter
    "x-image-save-bucket": Buffer.from(targetBucketName, 'utf8').toString('base64')
}

const res = obsClient.createSignedUrlSync({
    Method: method,
    Bucket: bucketName,
    Key: objectKey,
    QueryParams: queryParams
});

// Make a GET request for image persistency.
const url = urlLib.parse(res.SignedUrl);
const request = server.startsWith('http://') ? http : https;
const req = request.request({
    method: method,
    host: url.hostname,
    port: url.port,
    path: url.path,
    rejectUnauthorized: false,
    headers: res.ActualSignedRequestHeaders || {}
});
```

NOTE

- Object and bucket names must be URL-safe Base64 encoded in the **encodedObject = url_safe_base64_encode(name)** format. For example, object **panda.png** will be encoded as **cGFuZGEucG5n**. After Base64 encoding, if the name contains plus signs (+) and slashes (/), replace them with hyphens (-) and underscores (_), respectively.
- If a signature matching error is reported, check whether the AK and SK pair is correct and whether the accessed URL is the same as the signature URL generated by the code.
- Currently, image persistency with the range header is not supported.

16 FAQ

16.1 What Is Image Processing?

Image processing is a feature integrated in Object Storage Service (OBS). It provides stable, secure, efficient, and inexpensive image processing services. By using this feature, you can slim, crop, resize, and watermark images, as well as convert the formats of images.

You can access this feature via OBS Console and REST APIs, to process images stored in OBS anytime and anywhere and obtain the processed images right away.

16.2 How Do I Access Image Processing?

- Preview image effects in different styles on OBS Console.
- Make an API call on applications.

16.3 How Is Image Processing Billed?

To learn the pricing details of image processing, see the **Product Pricing Details** tab page at **OBS Pricing Details**.

Billing Factors

- Size of the original image, in GB

  The feature will be charged at a fixed unit price based on the number of service months.

If more than one image is processed, the total size of these images will be automatically calculated. For example, if an image of 0.01 GB is cropped and compressed into 0.005 GB, the image size is still regarded as 0.01 GB by OBS.

Billing Formulas

The following formulas apply to Standard storage and Infrequent Access storage. Archive storage is not considered. For details, see **Table 16-1**.

- 0–10 TB (inclusive): free
- More than 10 TB: Cost = (Total image size – 10 TB) x Unit price (USD/GB)

Table 16-1 Billing criteria of image processing

Item	Unit Price for Standard Storage (Monthly)	Unit Price for Infrequent Access Storage (Monthly)	Unit Price for Archive Storage (Monthly)
Image processing	● 0–10 TB (inclusive): free ● More than 10 TB: See the **Product Pricing Details** tab page at **OBS Pricing Details**.	● 0–10 TB (inclusive): free ● More than 10 TB: See the **Product Pricing Details** tab page at **OBS Pricing Details**.	N/A

16.4 How Many Image Styles Can I Create in a Bucket?

A bucket can have a maximum of 100 styles.

You can create an image style in either of the following ways:

- **Using OBS Console**
- **Making an API call**

16.5 What Formats Does Image Processing Support?

Supported original formats: JPG, JPEG, PNG, BMP, WebP, GIF, and TIFF Supported target formats: JPG, PNG, BMP, and WebP

16.6 How Do I Access Image Processing with a URL?

Accessing Images Not Publicly Readable

To access images that cannot be read by the public, add image processing parameters during signature calculation to create a signed temporary URL.

A Java SDK sample code is provided as follows. For details, see **SDK Overview**.

```
// Here uses the endpoint corresponding to the CN-Hong Kong region as an example.
String endPoint = "https://obs.ap-southeast-1.myhuaweicloud.com";
// Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store
them in the configuration file or environment variables.
// In this example, the AK and SK are stored in environment variables for identity authentication. Before
running this example, configure environment variables ACCESS_KEY_ID and SECRET_ACCESS_KEY.
// Obtain an AK and SK pair on the management console. For details, see https://
support.huaweicloud.com/intl/en-us/usermanual-ca/ca_01_0003.html.
String ak = System.getenv("ACCESS_KEY_ID");
String sk = System.getenv("SECRET_ACCESS_KEY");

// Create an ObsClient instance.
ObsClient obsClient = new ObsClient(ak, sk, endPoint);

// Set the URL validity period to 3600 seconds.
long expireSeconds = 3600L;
TemporarySignatureRequest request = new TemporarySignatureRequest(HttpMethodEnum.GET,
expireSeconds); request.setBucketName("bucketname");
request.setObjectKey("objectname");

// Configure image processing parameters.
Map<String,Object> queryParams = new HashMap<String, Object>();
queryParams.put("x-image-process", "image/resize,m_fixed,w_100,h_100/rotate,100");
request.setQueryParams(queryParams);

// Obtain the URL that supports image processing.
TemporarySignatureResponse response = obsClient.createTemporarySignature(request);
System.out.println("Getting object using temporary signature url:");
System.out.println("\t" + response.getSignedUrl());
```

Accessing Images Publicly Readable

To access images that can be read by the public, add image processing parameters to the URL request.

Sample URL for obtaining the information about the **example.jpg** image:

https://hw-image-demo.obs.ap-southeast-1.myhuaweicloud.com/example.jpg?x-image-process=image/info

16.7 Where Are Processed Images Stored After CDN Acceleration Is Enabled?

If you access the image processing service through CDN, processed images will be cached by CDN. Therefore, if you perform the same processing operations on the same images again, CDN returns the processed images without accessing the image processing service, and you will not be charged for the image processing requests.

16.8 Why Was My Bucket Accessed by IP Addresses Starting with 100 After I Performed Image Processing in My Logging-enabled Bucket?

During the image processing in your logging-enabled bucket, your original image will be downloaded from your bucket over the intranet for processing. For this reason, the intranet IP address (starting with 100) will be recorded in your bucket logs, and there will be costs on download requests.