This page shows examples and tips for using buckets to host a static website (/storage/docs/hosting-static-website).

The following scenarios demonstrate the objects that are served to you when you attempt to access them with various URLs.

Suppose a bucket named `www.example.com` has been configured as a website with the following settings and files:

- `MainPageSuffix` = "index.html"
- `NotFoundPage` = "404.html"
- The bucket contains three publicly shared objects: "index.html", "404.html", and "dir/index.html".

An empty object might exist at **`http://www.example.com/dir/`**. If an empty object exists, it is returned instead of
ory's **`index.html`** file. See the troubleshooting topic (/storage/docs/troubleshooting#empty-obj) if you get a 301 HT
nse when attempting to access **`http://www.example.com/dir/`**.

The following table shows the content served for selected URLs:

URL Requested	Content Served	HTTP response code
http://www.example.com http://www.example.com/ http://www.example.com/index.html	The object "index.html".	200
http://www.example.com/hello	The object "404.html".	404
http://www.example.com/dir/index.html	The object "dir/index.html".	200
http://www.example.com/dir	The object "dir/index.html".	301

http://www.example.com/dir/	The object "dir/index.html", assuming no zero-byte object exists for /dir/	200
	A zero byte empty object, if it exists for /dir/. See the Troubleshooting topic (/storage/docs/troubleshooting#empty-obj) for removing this zero byte object.	301

Suppose a bucket named `www.example.com` has been configured as a website with the following settings and files:

- `MainPageSuffix` = "main.html"
- `NotFoundPage` = "404.html"
- The bucket contains two publicly shared objects: "main.html" and "404.html".

The following table shows the content served for selected URLs:

URL Requested	Content Served	HTTP response code
http://www.example.com http://www.example.com/	The object "main.html".	200
http://www.example.com/index.html	The object "404.html".	404

If an object is shared publicly, you can also view that object with the URL:

For example, the URL for an `index.html` object would be:

For more information about working with publicly accessible data, see Accessing Public Data (/storage/docs/access-public-data).

The following are some tips to keep in mind when using a bucket to host a static website.

Suppose you also want to serve content at `test.example.com`, from a different bucket than the one that serves content at `www.example.com`. To do so:

1. Create a new bucket `test.example.com`. Because you already verified the domain `example.com`, you can create buckets to support subdomains without additional verification.
2. Add a new `CNAME` record (/storage/docs/hosting-static-website#cname) for the subdomain.

You can control how or if your website assets are cached by configuring the `Cache-Control` metadata (/storage/docs/metadata#cache-control). Generally, only set cache control metadata for objects that are accessible to all anonymous users, which is a requirement for any object served from a Cloud Storage bucket as part of a static website.

Cloud Storage applies a cache control setting of 3600 seconds to objects that are accessible to all anonymous users, unless you specify explicit cache control settings. See Viewing and Editing Metadata (/storage/docs/viewing-editing-metadata) for instructions for setting object metadata, such as `Cache-Control`.

The `MainPageSuffix` and `NotFoundPage` website configurations are only used for requests that come to Cloud Storage through the `CNAME` endpoint (/storage/docs/request-endpoints#cname) or through Cloud

Load Balancing (/load-balancing/docs/https/adding-a-backend-bucket-to-content-based-load-balancing). For example, a request to `www.example.com` shows the index page, but an equivalent request to `storage.googleapis.com/www.example.com` does not.

Thus, API behavior for requests to Cloud Storage domains, such as `storage.googleapis.com/www.example.com`, is preserved. For example, you can continue to list objects in the `www.example.com` bucket as you would for any other bucket. In the case of the `www.example.com` bucket, the object listing you receive includes `404.html` and `index.html`.

You can use Cloud Storage to host static assets for a dynamic website that is hosted, for example, in Google App Engine (/appengine/) or in Google Compute Engine (/compute/). Some benefits of hosting your static assets, like images or JavaScript files, in a bucket include:

- Cloud Storage behaves essentially like a Content Delivery Network (https://wikipedia.org/wiki/Content_delivery_network) (CDN) with no work on your part because publicly readable objects are, by default, cached in the Cloud Storage network.
- Bandwidth charges for accessing content typically cost less with Cloud Storage.
- The load on your web servers is lessened when serving the static content from Cloud Storage.

You can also control caching, such as disabling caching or setting cache lifetime, for your publicly readable static assets by using cache control request headers. For more information, see Setting cache parameters (#tip-cache).

When hosting static assets for a dynamic website, you do not need to create a `CNAME` record and point to a bucket with the same name as you do for a static website. For example, you could have a bucket named `www_example_com_assets` with appropriate assets configured as shared publicly and then access those assets using the Cloud Storage domain. For example, suppose you have the JavaScript file `library.js` in the bucket `www_example_com_assets` that is shared publicly, then you can access it as `http://storage.googleapis.com/www_example_com_assets/library.js`.

If you are serving assets from a bucket configured as a static website or serving static assets from a bucket for a dynamic website hosted outside of Cloud Storage, you should monitor the charges to your project containing the bucket. Serving content incurs Cloud Storage costs for storing the content, using the network, and performing retrieval operations. For details, see the Cloud Storage Pricing page (/storage/pricing).

The simple pricing example (/storage/pricing#pricing-example-simple) on the pricing page can be used as an approximation for the use case of a low-traffic, static website. You can use the Pricing Calculator (/products/calculator/) to generate a cost estimate based on your projected usage.

New Google Cloud users might be eligible for a free trial (/free-trial).

If you are a current Google Cloud user, you can get a detailed breakdown of your project costs on the billing page (https://console.cloud.google.com/billing).

See Troubleshooting (/storage/docs/troubleshooting#static-website) for common issues associated with using a bucket configured to serve static website content.

- Learn about other web serving options (/solutions/web-serving-overview) on Google Cloud.
- Try other Google Cloud tutorials that use Cloud Storage (/docs/tutorials#%22cloud%20storage%22).