SugarBox App Partner CDN Console

SugarBox's worldwide network of edge servers caches and processes your content at strategic locations around the globe, reducing the distance data must travel to reach your users.

When users access your application, the edge server nearest to them delivers the content instead of your origin server. This reduces network hops and latency, resulting in faster load times and a smoother user experience. Users perceive significantly improved performance because your content is delivered from a nearby location rather than a distant central server.

SugarBox app partner CDN console – overview

The SugarBox App Partner CDN Console provides granular control over cache duration and policies for your content.

To get started, create a content distribution site by specifying your origin server and defining how SugarBox ingests new content. Your origin server stores the original versions of your files (also called objects). The origin server can be an HTTP server, a MediaPackage channel, or cloud object storage.

SugarBox CDN workflow

The following workflow describes how SugarBox delivers content when users access your app or website.

1. A user browses your application and requests content.
2. DNS redirects the request to the nearest SugarBox edge server.
3. The edge server checks its cache for the requested files.
   - If cached: SugarBox delivers the files immediately to the user.
   - If not cached: SugarBox performs the following:
     a) Validates the request against your content distribution site configuration and forwards it to your origin server.
     b) The origin server returns the requested files.
     c) SugarBox begins streaming the files to the user as soon as the first byte arrives and simultaneously caches them at the edge location for a specified duration for subsequent requests.

Getting started with using SugarBox CDN

Sign a contract with SugarBox to begin using SugarBox CDN. After onboarding, you will receive login credentials for the SugarBox App Partner CDN Console. Use the console to configure your content distribution settings, then reference your content using the SugarBox domain name in your webpage or application URLs.

To access the SugarBox App Partner CDN console

Note: We recommend changing your username and password after your first sign-in.

1. Open the SugarBox CDN Console at https://cdn-console.stg.sboxdc.com/login.
2. Enter the username and password you received during onboarding.

Console capabilities

You can perform the following actions in the console:

- Create a new content distribution site
- Manage existing content distribution sites:
    - Edit
    - Enable
    - Disable
    - View
    - Delete

For detailed instructions on managing existing sites, see *Managing Content Distribution Sites*.

Creating a content distribution site

Create a content distribution site to configure the following settings:

- Origin server URL - The location where SugarBox fetches your files (flies are also called objects)
- Security protocol - Whether to require HTTPS for user access to your content
- Cache duration - How long (in seconds) SugarBox caches your objects at edge locations
- Cache key - Values to include in the cache key, which uniquely identifies each cached file for the distribution
- Request settings - Which HTTP headers, cookies, or query strings SugarBox should forward to your origin server
- CNAME record (optional) - An alternate domain name for your content

To create a content distribution site

1. On the Home page, click the **CREATE SITE** tab.
2. On the **CREATE SITE** page, enter your origin server details, so SugarBox CDN can forward incoming user requests to it.
3. Configure your origin server connection settings:
    - Select whether SugarBox should use HTTP or HTTPS to access your origin server
    - Specify connection parameters such as maximum connection attempts and response timeout
    - (Optional) Specify an origin path to direct all requests to a specific directory on your origin server
    - (Optional) Configure custom headers to capture information about each user request

    For details, see *Origin Server* and *Connection Policies and Custom Headers*.
4. In the **Default Segment Configuration** section, configure cache policies and related behaviors that apply to all objects.
    Note: By default, SugarBox applies these policies to all objects regardless of path

pattern.

About segments: You can apply different cache policies to different content segments. For example, in an OTT app, you might cache recently released movies differently than older content. To do this, first create the content distribution site, then create segments and configure their individual policies. For details, see *Setting Up Segment Configuration*.

Configure the following settings:

- **Path Pattern** - Not editable when creating a new site. By default, all objects use the same cache behavior. To create segment-specific policies, see Setting Up Segment Configuration.
- **Compress Objects Automatically** - Select a compression type for delivering cached files. Compression reduces file size, improving download speeds and page load times. For details, see Setting Up Segment Configuration.
- **Viewer Protocol Policy** - Select the protocol (HTTP/HTTPS) that users must use to access your content. For details, see Setting Up Segment Configuration.
- **Allowed HTTP Methods** - Specify which HTTP methods SugarBox should process and forward to your origin server. For details, see Setting Up Segment Configuration.
- **Cache Configurations** - Create or select cache policies to control caching behavior for all objects. To customize caching for specific content segments, configure additional policies after creating the site.

  Configure the following:
  - **Cache Policy** - Specify cache duration and cache key settings. For details, see Cache Key Settings and Creating a Cache Policy.
  - **Origin Request Policy** - Specify which HTTP headers, query parameters, or cookies SugarBox should forward to your origin server. For details, see Setting Up Origin Request Policy.

5. In the **Settings** section, configure site-wide settings such as CNAME records and HTTP version support. For details, see *Settings*.
6. Click **CREATE SITE** to save your content distribution site.
7. Enable the site to begin processing live requests.
   For details, see *Enabling a Content Distribution Site*.