**Kollective - Brightcove Integration - Deployment Guide**

**Table of contents**

[1. Introduction](#_Toc43212154)

[1.1 Purpose](#_Toc43212155)

[1.2 Definitions, Acronyms, and Abbreviations](#_Toc43212156)

[2. Backend Application Deployment](#_Toc43212157)

[2.1. Server requirements](#_Toc43212158)

[2.2. Configurations](#_Toc43212159)

[2.2.1 Production Mode](#_Toc43212160)

[2.2.2 Application Port](#_Toc43212161)

[2.2.3 Database](#_Toc43212162)

[2.2.4 Email Service](#_Toc43212163)

[2.2.5 Kollective Publish API Base URL](#_Toc43212164)

[2.2.6 Brightcove API Base URLs And Custom Field Names](#_Toc43212165)

[2.3. How to deploy](#_Toc43212166)

[3. Kollective Plugin Deployment](#_Toc43212167)

[3.1. Server requirements](#_Toc43212168)

[3.2. Plugin Configurations](#_Toc43212169)

[4. Additional Notes](#_Toc43212170)

# 

# 

# 1. Introduction

Brightcove seeks to integrate with Kollective to deliver VOD and Live to Brightcove players using the Kollective eCDN where available.

## 1.1 Purpose

This document provides the guide to deploy and set up both the backend application and the Kollective plugin.

# 2. Backend Application Deployment

## 2.1. Server requirements

The following are the requirements for the hosting server to host the backend application:

* NodeJS version 13 or later.
* Database: MongoDB

The hosting server can be a Linux or Windows server. This backend application can also be deployed to Heroku.

The MongoDB can be on the same server with the hosting server above or on a different server or using a [MongoDB Atlas cloud database](https://www.mongodb.com/cloud).

## 2.2. Configurations

This section will guide you to make the configuration changes for the backend application.

All the configurations are in the “config.env” file in the root directory for the backend application deployment package.

### 2.2.1 Production Mode

Set NODE\_ENV=production to run application in production mode, or NODE\_ENV=development while development:

NODE\_ENV=production

### 2.2.2 Application Port

Set the port for the application to run on:

PORT=3000

### 2.2.3 Database

The backend application requires a MongoDB database in order to function. You can specify the database connection to use in the following format:

DATABASE=mongodb://{host}:{port}/{databaseName}

For example:

DATABASE=mongodb://localhost:27017/Brightcove

If you want to use a MongoDB Atlas cloud database, you can set it as below:

DATABASE=mongodb+srv://{user}:{password}@{cluster}/{databaseName}?retryWrites=true&w=majority

For example:

DATABASE=mongodb+srv://myuser:mypassword@cluster0-xafsz.mongodb.net/MyDatabase?retryWrites=true&w=majority

### 2.2.4 Email Service

The backend application requires an email service setting to send notification emails.

* To send emails using SendGrid email service, you need to set EMAIL\_SERVICE=SendGrid and set SendGrid username and password as follow:

# Send email using Sendgrid email service.

EMAIL\_SERVICE=SendGrid

# Send email using Sendgrid email service.

SENDGRID\_USERNAME={apikey}

SENDGRID\_PASSWORD={password}

* To send emails using other email services, you need to set EMAIL\_SERVICE=Other and set all the email server host, port and username, password as follow:

# Set EMAIL\_SERVICE=Other to send email using normal email service using host and port

EMAIL\_SERVICE=Other

# Send email using host and port

EMAIL\_HOST=smtp.mailtrap.io

EMAIL\_PORT=25

EMAIL\_USERNAME=255ef9aaa9268e

EMAIL\_PASSWORD=1476fcea94aba0

EMAIL\_FROM=hiep.khuc@evizi.com

### 2.2.5 Kollective Publish API Base URL

The backend application calls the Kollective publish API to publish a video from Brightcove with video sources to Kollective and get back a content token. You need to update the Kollective publish API base url using the correct url that Kollective provided:

# Kollective publish API base url

KOLLECTIVE\_PUBLISH\_API\_BASE\_URL=https://content.stage.kollective.app/api

### 2.2.6 Brightcove API Base URLs And Custom Field Names

The backend application needs the following Brightcove API base URLs:

# Brightcove API base URLs

BRIGHTCOVE\_ACCESS\_TOKEN\_BASE\_URL=https://oauth.brightcove.com/v4/access\_token

BRIGHTCOVE\_CMS\_API\_BASE\_URL=https://cms.api.brightcove.com/v1

BRIGHTCOVE\_LIVE\_API\_BASE\_URL=https://api.bcovlive.io/v1

The following custom fields name settings cannot be changed:

# Required custom field names for each Brightcove Account ID

CUSTOM\_FIELD\_KOLLECTIVE\_CONTENT\_TOKEN=kollective\_content\_token

CUSTOM\_FIELD\_KOLLECTIVE\_IS\_SYNCHRONIZED=kollective\_is\_synchronized

CUSTOM\_FIELD\_KOLLECTIVE\_SYNCHRONIZED\_TIME=kollective\_synchronized\_time

CUSTOM\_FIELD\_KOLLECTIVE\_LAST\_SYNC\_STATUS=kollective\_last\_sync\_status

Each Brightcove Account ID to be used for a syndicator account needs to have the following custom fields name in order for the synchronization to work:

|  |  |  |
| --- | --- | --- |
| **#** | **Custom Field Name** | **Type** |
| 1 | kollective\_content\_token | text |
| 2 | kollective\_is\_synchronized | text |
| 3 | kollective\_synchronized\_time | text |
| 4 | kollective\_last\_sync\_status | text |

## 2.3. How to deploy

After making all the required configuration changes, to deploy the backend application to a new server, you need to do the following:

1. Copy the backend application package to the targeted deployment folder on the hosting server.
2. Open a console application and go to the root folder of the application package in step 1, and execute the following command to install all the required Node packages:

# npm install --production

1. Execute the command below to import default super-admin user:

# npm run import-users

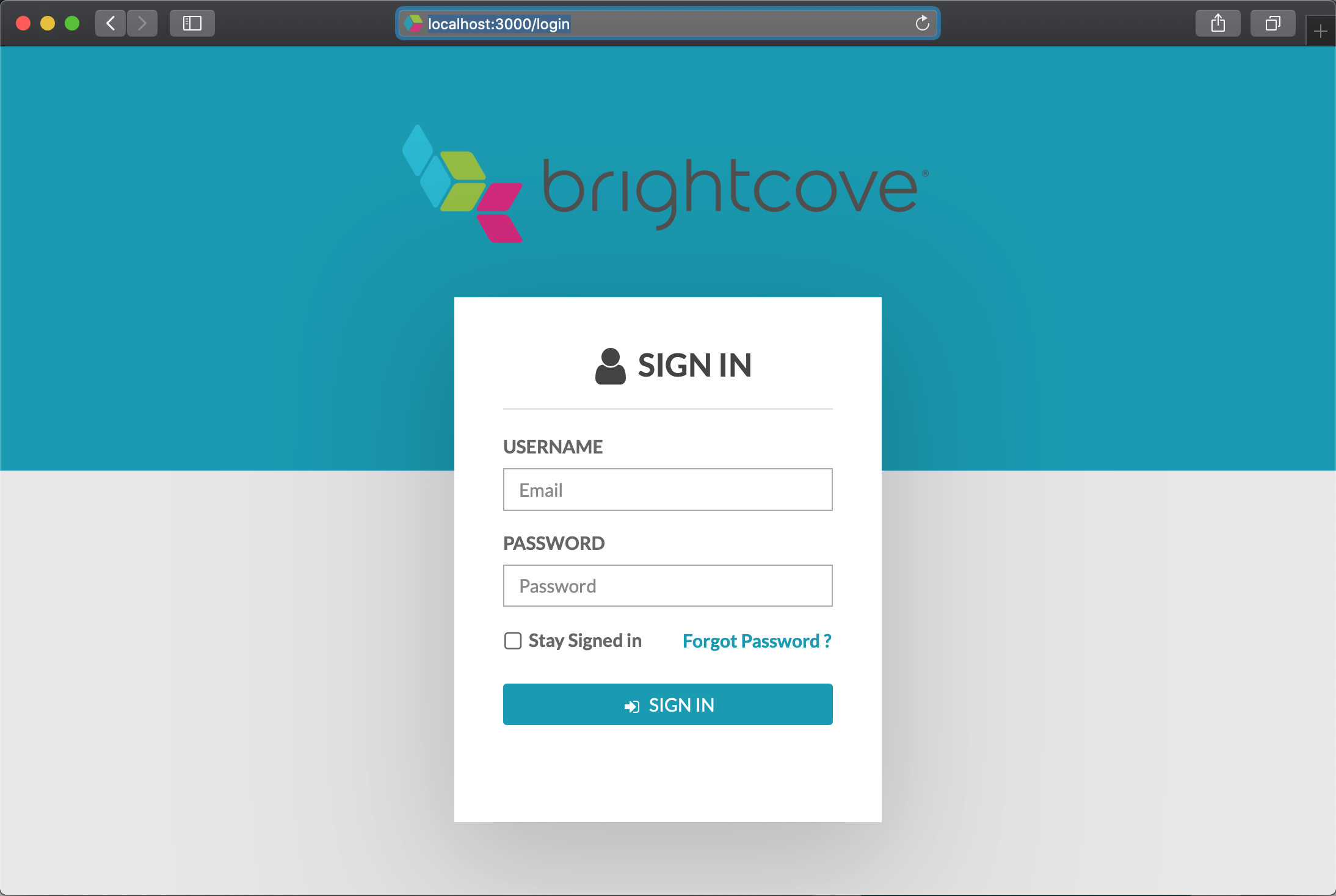
1. To start the backend application, on the same console application as in step 2, execute the following command:

# npm run start

If everything is setup correctly, then you can open the application on a browser using the configured address, for example: <http://localhost:3000/login>

You can log into the application with the default super-admin user imported above:

* Username: [super-admin@brightcove.com](mailto:super-admin@brightcove.com)
* Password: Br1ghtC0ve



# 3. Kollective Plugin Deployment

## 3.1. Server requirements

The Kollective plugin is a javascript file, so you can deploy it on any server that allows access to the file from outside using a public address.

You can find the Kollective plugin in the “kollective-player-plugin” folder in the Kollective Integration github repository.

[backend-application-root-folder]/kollective-player-plugin/k-plugin.js

For example, you deploy the Kollective plugin JavaScript file to the following address:

https://my-server/k-plugin-1.0.6.js

## 3.2. Plugin Configurations

To add the Kollective plugin to a player in the Brightcove account, please follow the steps below:

* In the player setting page in the Brightcove studio, go to the Plugins menu item and add a new Plugin:
  + **Plugin name (case sensitive, must be all lowercase “kollective”):** kollective
  + JavaScript URL: <https://my-server/k-plugin-1.0.6.js> (the address where you deployed the plugin file)
  + Options(JSON):

{

"ksdk\_url": "<https://cdn.kollective.app/ksdk/ksdk-2.1.9.min.js>",

"policy\_key": “value\_of\_the\_player\_policy\_key”

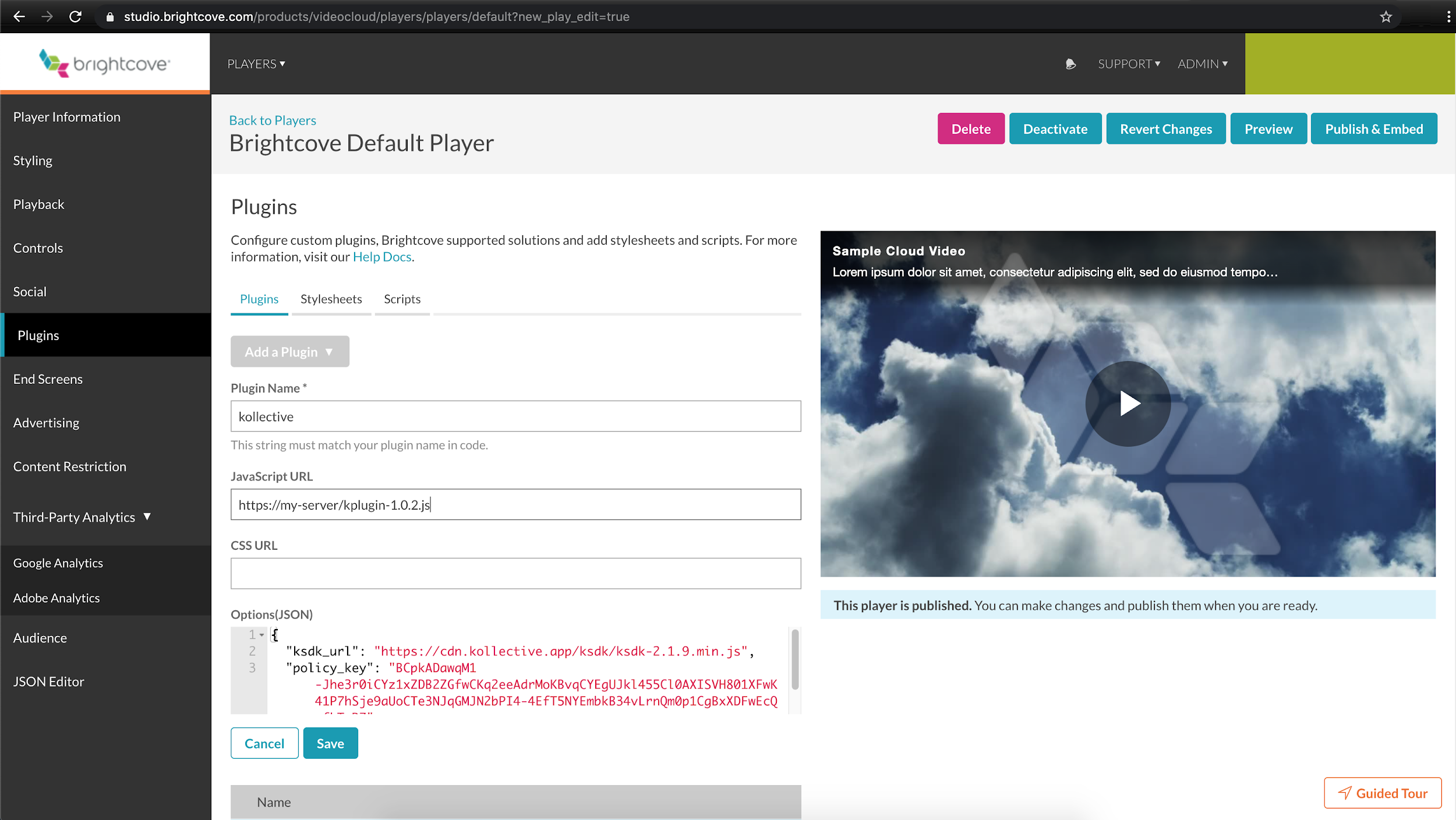
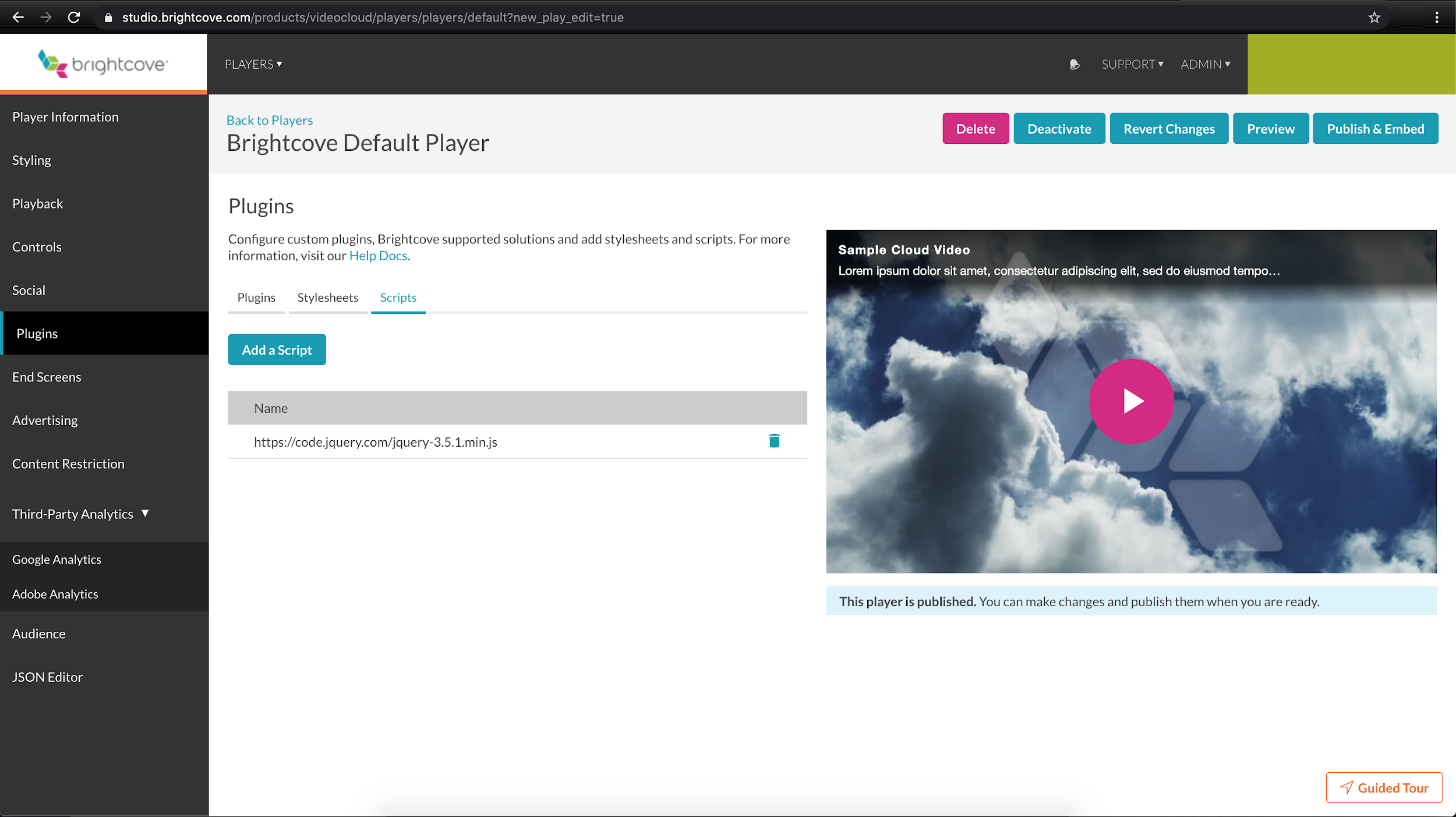
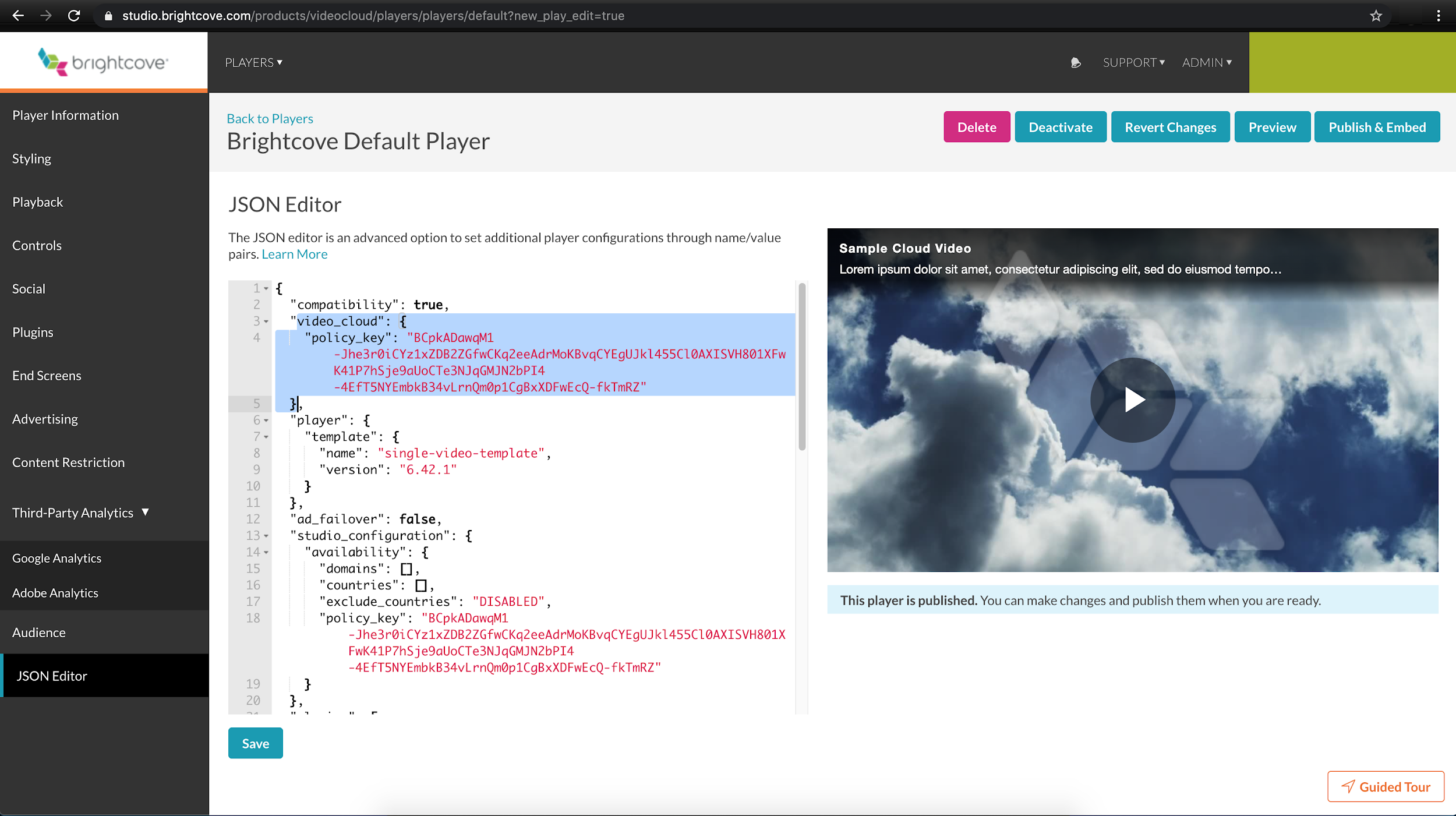
}

You can acquire the policy key by looking at the JSON editor for the player you intend to use.

* + In the Script tab, add a script and add the following file:

<https://code.jquery.com/jquery-3.5.1.min.js>

* Please note, the “ksdk\_url” points to the ksdk library from Kollective, and you can get the player “policy\_key” in the “JSON Editor” menu item at the bottom of the left menu bar as the screenshots below.
* Remember to Publish the changes you made to the player after Save.



# 

# 4. Additional Notes

Kollective needs to **add the domain name where you deployed your backend application to their whitelist** so that playback on the video detail page or live event detail page will work, otherwise you will always get the CORS error.

You will also need to request Kollective to **add the domain where you deployed your CMS pages into their whitelist** in order for the playback to work on those pages.

For example, you have a page at <https://my-server.com/my-video-page>that uses the Brightcove player with the Kollective plugin above, then the “my-server.com” will need to be added to the Kollective whitelist so that playback on the page will work, otherwise you will get the CORS error.