


Brightcove M365 Connector for SharePoint — Installation Guide

Brightcove Global Services | Last updated: March 2026 | Version 1.0.0.0

 This document is intended for Brightcove customers and authorized partners.

1. About this guide

This guide walks administrators through installing and configuring **Brightcove M365 Connector for SharePoint** in a Microsoft 365 tenant.

The connector consists of two deployable components:

- **SharePoint Framework (SPFx) app** — A tenant-scoped solution that provides web parts for embedding Brightcove videos, playlists, and experiences on modern SharePoint pages, plus provisioned site pages for connector configuration and content management.
- **Proxy API (Azure Function App)** — A secure server-side API layer that stores Brightcove API credentials in Azure Key Vault and brokers authenticated requests between SharePoint site instances and Brightcove. This eliminates the need to expose Brightcove API credentials within the SharePoint site.

Who should use this guide

- **SharePoint Admin** — Tenant App Catalog access; can upload SPFx packages and approve API permissions.
- **Azure Admin** — Can create resource groups, Function Apps, Key Vaults, and App Registrations in Microsoft Entra ID.
- **Brightcove Admin** — Can create API authentication credentials in Video Cloud Studio.

What's included in your package

File	Description
<code>brightcove-video-connector.sppkg</code>	Compiled SPFx package to upload to the SharePoint Tenant App Catalog.
<code>functionapp.zip</code>	Compiled Azure Function (Proxy API) to deploy to your Function App.
This guide	Installation and configuration instructions.
User Guide	Separate document covering how to use the connector's web parts.

What you'll do at a high level

1. Create Brightcove API credentials in Video Cloud Studio.
2. Provision Azure resources (Resource Group, Key Vault, Function App) and create an App Registration in Microsoft Entra ID.
3. Deploy the compiled Proxy API (`functionapp.zip`) to your Function App.
4. Upload the SPFx package (`brightcove-video-connector.sppkg`) to the SharePoint Tenant App Catalog, enable it, and approve its API permission.
5. Add the app to a SharePoint site.
6. Configure the Brightcove Connector Settings page (proxy connection + Brightcove account credentials).
7. Verify the installation by accessing the content management page and separately adding a Brightcove web part to a page.

What this guide does not cover

- Creating Brightcove players or experiences in Video Cloud Studio.
- Customizing the web parts beyond the settings exposed in the UI.
- Configuring Azure resource monitoring, logging, access policies or cost controls

2. Prerequisites

2.1 Accounts and roles

- **Brightcove Video Cloud account** with Admin permissions to create API authentication credentials.
- **Microsoft 365 tenant** with SharePoint Online.
- **Azure subscription** within the same Microsoft Entra ID directory (or a different directory if using multi-tenant configuration — see Section 4.5).
- **Microsoft Entra ID permissions** to create and manage App Registrations and grant admin consent. Recommended role: *Application Administrator* or *Cloud Application Administrator*.

2.2 Who does what

Role	Responsibilities
Brightcove Admin	Creates API credentials in Video Cloud Studio.
Azure Admin	Creates resource group, Function App, Key Vault, App Registration; configures authentication and CORS; deploys Proxy API.
SharePoint Admin	Manages App Catalog; uploads and deploys the SPFx package; approves API permissions.
Site Owner / Editor	Adds the app to a site; configures connector settings; adds Brightcove web parts to pages.

2.3 Environment readiness

- A **SharePoint Tenant App Catalog** exists (or you have permission to create one).
- A **modern SharePoint site** where the connector will be used.
- An **Azure region** selected for hosting the Function App (choose a region close to your users).

2.4 Brightcove content readiness

- At least one published video, playlist, or In-Page Experience (IPX) in your Brightcove Video Cloud account.
- API credentials created for each Brightcove account you plan to expose in SharePoint (see Section 3). Each API credential should be associated with a

single Brightcove account.

2.5 Decision points (capture before you start)

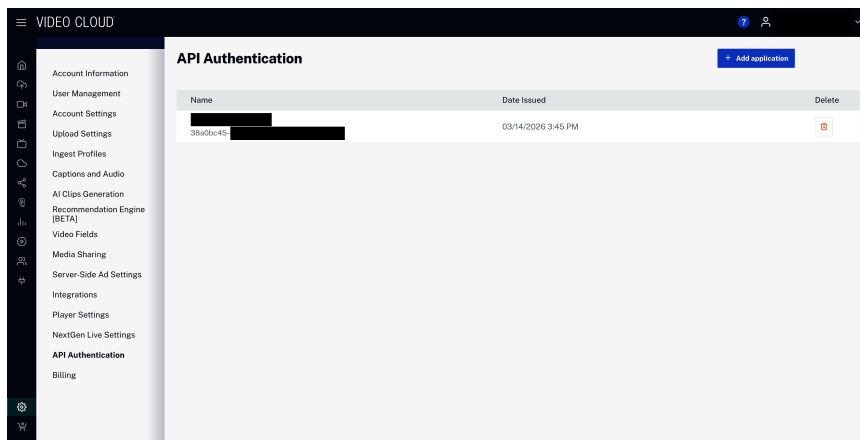
- Azure resource group name and region.
- Function App name (must be globally unique — becomes part of the URL).
- Key Vault name (must be globally unique).
- **Tenant topology:** Will the Azure subscription and SharePoint Online be in the **same Entra ID tenant** or **different tenants**? This affects how you configure the App Registration (see Section 4.5).
- Which Brightcove accounts (and corresponding API credentials) will be available to authors in SharePoint.

3. Create Brightcove API credentials

Before setting up Azure or SharePoint, create the API credentials that the connector will use to communicate with Brightcove Video Cloud.

3.1 Navigate to API Authentication

In Video Cloud Studio, go to **Admin** → **API Authentication**.



3.2 Register a new application

Click **Register New Application** and configure as follows:

- **Name:** A descriptive name (e.g., `sp-connector-production`).






- **Select Accounts for Authorization:** Select the **single** Brightcove account this credential will be used for.

⚠ Important: Each API credential should be associated with only one Brightcove account. If you need to connect multiple accounts to SharePoint, create separate credentials for each one. The connector UI displays a single account ID per credential entry. Credentials linked to multiple accounts will work, but only the account ID entered in the connector settings will be used for API calls.

3.3 Required API scopes

Select the following permissions:



CMS

- Notifications 
- Playlist Read 
- Playlist Read/Write 
- Video Read 
- Video Read/Write 

Dynamic Ingest

- Create 
- Push Files 

Gallery Experiences

- Read 
- Read/Write 

Ingestion Profiles

- Configuration Read 
- Read 

Players

- Read 

- Read/Write 

Exposed Brightcove APIs

Ad Monetization <input type="checkbox"/> Policy Read <input type="checkbox"/> Policy Read/Write	Analytics <input type="checkbox"/> Read	Audience <input type="checkbox"/> Read <input type="checkbox"/> Read/Write
CMS <input checked="" type="checkbox"/> Notifications <input checked="" type="checkbox"/> Playlist Read <input checked="" type="checkbox"/> Playlist Read/Write <input type="checkbox"/> Sharing Relationships Read <input type="checkbox"/> Sharing Relationships Read/Write <input checked="" type="checkbox"/> Video Read <input checked="" type="checkbox"/> Video Read/Write	Cloud Payout <input type="checkbox"/> Read <input type="checkbox"/> Read/Write	Dynamic Ingest <input checked="" type="checkbox"/> Create <input checked="" type="checkbox"/> Push Files
Gallery Experiences <input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Read/Write	Image Token Management <input type="checkbox"/> Image Token Management Read <input type="checkbox"/> Image Token Management Write	Ingest Integrations <input type="checkbox"/> Integrations create <input type="checkbox"/> Integrations delete <input type="checkbox"/> Integrations read <input type="checkbox"/> Integrations update
Ingestion Profiles <input checked="" type="checkbox"/> Configuration Read <input type="checkbox"/> Configuration Read/Write <input checked="" type="checkbox"/> Read <input type="checkbox"/> Read/Write	Interactivity API <input type="checkbox"/> Read <input type="checkbox"/> Read/Write	Livev2 Jobs <input type="checkbox"/> All <input type="checkbox"/> Create <input type="checkbox"/> Delete <input type="checkbox"/> Read <input type="checkbox"/> Update
Livev2 SSAI <input type="checkbox"/> All <input type="checkbox"/> Create <input type="checkbox"/> Delete <input type="checkbox"/> Read <input type="checkbox"/> Update	Livev2 Settings <input type="checkbox"/> Read	Playback Auth Account Config <input type="checkbox"/> Account Config Read
Playback Auth Audit <input type="checkbox"/> Audit	Playback Auth Blacklist <input type="checkbox"/> Blacklist	Playback Auth Devices <input type="checkbox"/> Devices Read <input type="checkbox"/> Devices Write
Playback Auth Key <input type="checkbox"/> Key Read <input type="checkbox"/> Key Write	Playback Auth Rights <input type="checkbox"/> Playback Rights Read <input type="checkbox"/> Playback Rights Write	Players <input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Read/Write
		SSAI <input type="checkbox"/> Read <input type="checkbox"/> Read/Write

3.4 Save and record credentials

Click **Save**. Record the following values — you will need them in Section 6:

- **Account ID** (numeric)
- **Client ID**
- **Client Secret**



The Client Secret is shown only once. Copy and store it securely.

4. Azure setup (one-time)

This section walks your Azure admin through provisioning and configuring the Azure resources required by the Proxy API. You will need administrative privileges in Azure to complete these steps.

Recommended naming convention: `${resourceType}-bcvc-${companyName}-${environment}-${region}`

For example: `rg-bcvc-acme-prd-eus` (resource group), `kv-bcvc-acme-prd-eus` (Key Vault), `fa-bcvc-acme-prd-eus` (Function App).

4.1 Create a resource group

Where: Azure Portal → Resource Groups → Create

- **Name:** Use your naming convention (e.g., `rg-bcvc-acme-prd-eus`).
- **Region:** Select the region closest to your users.
- Click Review + create → Create

Microsoft Azure

Home > Resource Manager | Resource groups

Create a resource group

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

Subscription * ⓘ

Resource group name * ⓘ

Region * ⓘ

Previous Next Review + create

4.2 Provision a Key Vault

Where: Azure Portal → Key Vaults → Create

Configuration:

- **Resource group:** Select the resource group created in 4.1.
- **Key vault name:** Use your naming convention (e.g., `kv-bcvc-acme-prd-eus`).
- **Region:** Select the region closest to your users.
- **Pricing tier:** Standard.
- **Days to retain deleted vaults:** 90
- **Purge protection:** Disable
- **Access configuration** → **Permission model:** Azure role-based access control (RBAC).

- **Networking** → **Enable public access**: Checked.
- **Networking** → **Allow public access from**: All networks.
- Click Review + create → Create

⚠ The vault URI is shown on the overview blade. Copy it, you will need it in section 4.9

The screenshot shows the 'Create a key vault' wizard in the Microsoft Azure portal, specifically the 'Basics' tab. The interface includes a navigation bar with 'Home' and 'Create a key vault' options. Below the navigation bar, there are tabs for 'Basics', 'Access configuration', 'Networking', 'Tags', and 'Review + create'. The 'Basics' tab is active, displaying a description of Azure Key Vault and a 'Project details' section. The 'Project details' section contains two dropdown menus: 'Subscription' (with a red asterisk) and 'Resource group' (with a red asterisk). The 'Subscription' dropdown is set to a value starting with 'rg-bcvc-acme-prd-eus', and the 'Resource group' dropdown is set to 'rg-bcvc-acme-prd-eus'. Below these, the 'Instance details' section contains three dropdown menus: 'Key vault name' (with a red asterisk and a help icon), 'Region' (with a red asterisk), and 'Pricing tier' (with a red asterisk and a help icon). The 'Key vault name' dropdown is set to 'kv-bcvc-acme-prd-eus' and has a green checkmark. The 'Region' dropdown is set to 'East US'. The 'Pricing tier' dropdown is set to 'Standard'. Below the 'Instance details' section, the 'Recovery options' section contains a paragraph of text and two radio button options: 'Soft-delete' (with a help icon) and 'Days to retain deleted vaults' (with a red asterisk and a help icon). The 'Soft-delete' option is selected and set to 'Enabled'. The 'Days to retain deleted vaults' option is set to '90'. Below the 'Recovery options' section, there are three buttons: 'Previous', 'Next', and 'Review + create'.

Microsoft Azure

Home

Create a key vault ...

Basics Access configuration Networking Tags Review + create

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules or HSMs. The HSMs used are Federal Information Processing Standards (FIPS) 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Instance details

Key vault name * ✓

Region *

Pricing tier *

Recovery options

Soft delete protection will automatically be enabled on this key vault. This feature allows you to recover or permanently delete a key vault and secrets for the duration of the retention period. This protection applies to the key vault and the secrets stored within the key vault.

To enforce a mandatory retention period and prevent the permanent deletion of key vaults or secrets prior to the retention period elapsing, you can turn on purge protection. When purge protection is enabled, secrets cannot be purged by users or by Microsoft.

Soft-delete ☐ Enabled

Days to retain deleted vaults *

Purge protection ☐ ☒ Disable purge protection (allow key vault and objects to be purged during retention period)

☐ Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)

Previous Next Review + create

Home

Create a key vault

Basics Access configuration Networking Tags Review + create

Configure data plane access for this key vault

To access a key vault in data plane, all callers (users or applications) must have proper authentication.

Permission model

Grant data plane access by using a [Azure RBAC](#) or [Key Vault access policy](#)

- ☒ Azure role-based access control (recommended) ⓘ
- ☐ Vault access policy ⓘ

Resource access

- ☐ Azure Virtual Machines for deployment ⓘ
- ☐ Azure Resource Manager for template deployment ⓘ
- ☐ Azure Disk Encryption for volume encryption ⓘ

Home

Create a key vault

Basics Access configuration Networking Tags Review + create

You can connect to this key vault either publicly, via public IP addresses or service endpoints, or privately, using a private endpoint.

Enable public access ☒

Public Access

Allow access from:

- ☒ All networks
- ☐ Selected networks

i Traffic from all public networks can access this resource. This is not recommended for private applications or environments. [Learn more about key vault network security configuration](#)

Private endpoint

Create a private endpoint to allow a private connection to this resource. Additional private endpoint connections can be created with this resource.

+ Create a private endpoint

Name	Subscription	Resource group
Click on add button to add private endpoint		

Vault URI : <https://kv-bcvc-acme-prd-eus.vault.azure.net/>
Sku (Pricing tier) : Standard
Directory ID : c84b4cc9-4cfb-4c8e-a320-e6901d163189
Directory Name : Default Directory
Soft-delete : [Enabled](#)
Purge protection : [Disabled](#)

4.3 Provision a Function App

Where: Azure Portal → Function App → Create

Configuration:

- **Hosting plan:** Flex Consumption
- **Resource group:** Select the resource group created in 4.1.
- **Function app name:** Use your naming convention (e.g., `fa-bcvc-acme-prd-eus`).
- **Secure unique default hostname:** On
- **Region:** Select the region closest to your users.
- **Runtime stack:** Node.js
- **Version:** 22 LTS
- **Instance size:** 2048 MB (recommended for production).
- **Zone redundancy:** Disabled
- **Blob service diagnostic settings:** Configure later
- **Enable public access:** On
- **Enable virtual network integration:** Off
- **Backend providers:** Bring your own: Azure Storage
- **Continuous deployment:** Disable
- **Basic authentication:** Disable



Azure will automatically provision a **Storage Account** for the Flex Consumption Function App to store deployment packages. You do not need to create one separately.



Azure requires a connection to Application Insights for the Flex Consumption plan to enable the Log Stream; without it, real-time console log output will not be available in the portal.

Microsoft Azure

Home > Function App > Create Function App

Create Function App (Flex Consumption) ...

BasicsStorageAzure OpenAINetworkingMonitoringDurable FunctionsDeploymentAuthenticationTagsReview + create

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Subscription

Resource Group *

rg-bcvc-acme-prd-eus

Create new

Instance Details

Function App name *

fa-bcvc-acme-prd-eus

-3ckf2fvds4cyeg.canadacentral-01.azurewebsites.net

Secure unique default hostname on.

[More about this update](#)

Region *

Canada Central

Runtime stack *

Node.js

Version *

22 LTS

Instance size *

2048 MB

Zone redundancy

Instances of your app are distributed across availability zones for increased reliability. [More about zone redundancy](#)

Zone redundancy

☐ Enabled: Your Flex Consumption app will be zone redundant. This changes your app's required instance count per function or function group.

☒ Disabled: Your Flex Consumption app will not be zone redundant.

Review + create

< Previous

Next: Storage >

Microsoft Azure

Home

>

Function App

>

Create Function App

Create Function App (Flex Consumption) ...

Basics

Storage

Azure OpenAI

Networking

Monitoring

Durable Functions

Deployment

Authentication

Storage

Select a storage account or create a new one. Accounts must support blobs, queue, and Table storage. [Learn more](#)

Storage account * (New) rgbcvcacmepreusa30d

[Create new](#)

You can configure networking on new storage accounts when you enable virtual network integration and configure outbound access networking on your function app.

Diagnostic Settings

Configure diagnostic settings to enable data monitoring for your storage account. [Learn more](#)

Blob service diagnostic settings

☒

Configure later (Recommended for custom controls)

Select this option for control over log destinations, retention policies, and specific logs and metrics.

☐

Configure now (Recommended for basic controls)

Select to configure Azure Log Analytics with storage, write logs, and transaction metrics for the blob service. You can still modify these settings later.

Microsoft Azure

Home

>

Function App

>

Create Function App

Create Function App (Flex Consumption) ...

Basics

Storage

Azure OpenAI

Networking

Monitoring

Durable Functions

Deployment

Authentication

Function Apps can be provisioned with the inbound address being public to the internet or isolated to an Azure virtual network. Function Apps can also be provisioned with outbound traffic able to reach endpoints in a virtual network; be governed by network security groups or affected by virtual network routes. By default, your app is open to the internet and cannot reach into a virtual network. These aspects can also be changed after the app is provisioned. [Learn more](#)

You can configure networking on new services when you enable virtual network integration and configure outbound access networking on your function app.

Enable public access *

☒ On
☐ Off

Enable virtual network integration *

☐ On
☒ Off

Microsoft Azure

Home

>

Function App

>

Create Function App

Create Function App (Flex Consumption) ...

Basics

Storage

Azure OpenAI

Networking

Monitoring

Durable Functions

Deployment

Authentication

Durable Functions allows you to write stateful functions (e.g. orchestrations and entities) that are long-running and execute reliably, with automatic recovery from infrastructure failures. [Learn More](#)

Backend providers *

☒

Bring your own: Azure Storage

For scenarios with small to moderate throughput and low performance requirements. Durable Functions will share the storage account of the function app, using the host storage (AzureWebJobsStorage) connection. Storage account name: rgbcvcacmepreusa30d

☐

Azure managed: Durable Task Scheduler

For best performance and user experience. Fully managed by Azure with convenient features such as built-in monitoring dashboard, automatic disaster recovery, data encryption, and other enterprise-friendly features.

Microsoft Azure

Home > Function App > Create Function App

Create Function App (Flex Consumption)

Basics Storage Azure OpenAI Networking Monitoring Durable Functions **Deployment** Authentication

Continuous deployment settings

Set up continuous deployment to easily deploy code from your GitHub repository via GitHub Actions. [Learn more](#)

Continuous deployment ☒ Disable ☐ Enable

GitHub settings

Set up GitHub Actions to push content to your app whenever there are code changes made to your repository. Note: Your GitHub account must have write access to the selected repository in order to add a workflow file which manages deployments to your app.

GitHub account [Authorize](#)

Organization [Select organization](#)

Repository [Select repository](#)

Branch [Select branch](#)

Workflow configuration

Click the button below to preview what the GitHub Actions workflow file will look like before setting up continuous deployment.

[Complete the Basics tab and the form above to preview the GitHub Actions workflow file.](#)

[Preview file](#)

Authentication settings

Choose if you would like to allow basic authentication to deploy code to your app. [Learn more](#)

Basic authentication ☒ Disable ☐ Enable

Application Insights (optional)

Application Insights provides detailed logging, request tracing, and performance monitoring. However, it adds cost and complexity.

If you choose to enable Application Insights:

- You can enable it during Function App creation or add it later from **Settings** → **Application Insights** → **Turn on Application Insights**.
- A Log Analytics Workspace will be created automatically.
- Be aware this can generate ongoing costs depending on log volume.

If you do **not** enable Application Insights, you can still view real-time logs via:

- **Function App** → **Monitoring** → **Log stream** (live tail of console output)
- **Function App** → **Functions** → **[function name]** → **Monitor** (invocation history, requires Application Insights)

For production environments where you need detailed request-level tracing (e.g., diagnosing intermittent auth failures), Application Insights is recommended.

Diagnostic Settings (Alternative to Application Insights)

If you want to persist logs for long-term auditing or querying without using the full Application Insights suite, you can use **Diagnostic Settings** to send logs to a **Log Analytics Workspace**.

Why use Diagnostic Settings?

- **Centralized Logging:** Aggregate logs from multiple Function Apps into one workspace.
- **KQL Querying:** Use the Kusto Query Language (KQL) to filter, analyze, and create dashboards from your logs.
- **Lower Overhead:** Captures standard platform and application logs without the advanced instrumentation overhead of Application Insights.

To enable Diagnostic Settings:

1. Navigate to your **Function App** → **Monitoring** → **Diagnostic settings**.
2. Click **+ Add diagnostic setting**.
3. **Logs:** Check the box for **Function Application Logs** (this captures `console.log` and `context.log` output).
4. **Destination details:** Check **Send to Log Analytics workspace**.
5. Select your **Subscription** and an existing **Log Analytics Workspace**.
6. Click **Save**.

Viewing the Logs:

- Once enabled, go to **Function App** → **Monitoring** → **Logs**.
- It may take **5–15 minutes** for the first logs to appear after the setting is saved.
- Run a simple query like `FunctionAppLogs | order by TimeGenerated desc` to see your latest Node.js output.

4.4 Create Function App managed identity and assign Key Vault role

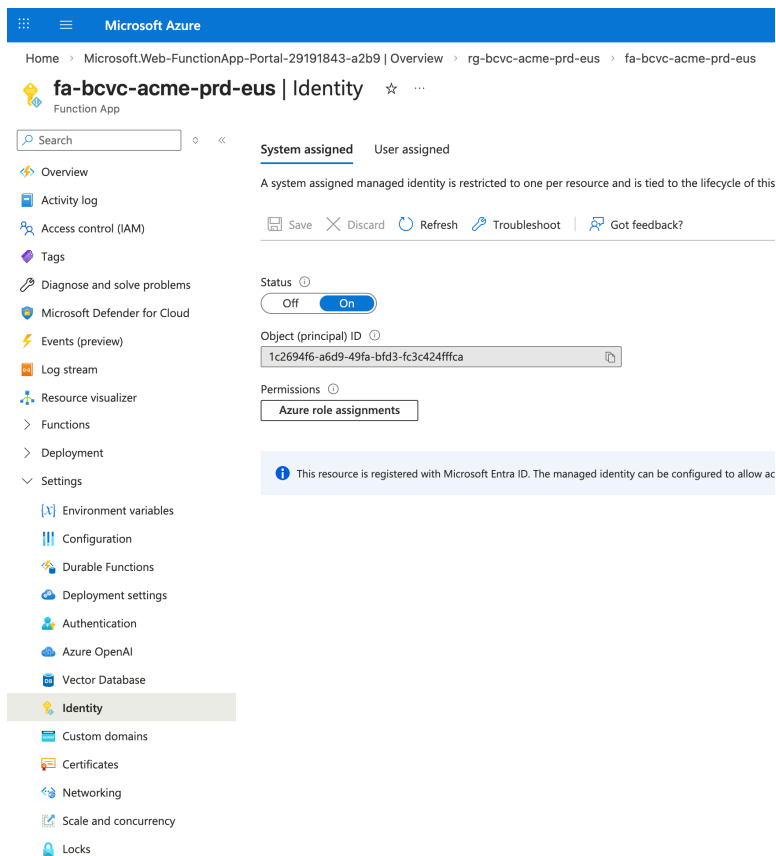
The Function App needs a managed identity with permission to read and write secrets in the Key Vault.

Enable managed identity:

1. Navigate to your Function App → **Settings** → **Identity**.
2. On the **System assigned** tab, toggle **Status** to **On**.
3. Click **Save** and then **Yes** to enable the system managed identity.

Assign Key Vault role:

1. On the same Identity page, click **Azure role assignments**.
2. Click **Azure role assignments**.
3. Click **Add role assignment (Preview)**
4. Configure:
 - **Scope:** Key Vault
 - **Subscription:** Your subscription
 - **Resource:** The Key Vault created in 4.2
 - **Role:** Key Vault Secrets Officer
5. Click **Save**.



Add role assignment (Preview)

Scope ⓘ
Key Vault

Subscription
Visual Studio Enterprise Subscription

Resource ⓘ
kv-bcvc-acme-prd-eus ⓘ

Role ⓘ
Key Vault Secrets Officer ⓘ

[Learn more about RBAC](#)


4.5 Create an App Registration

The Proxy API uses Microsoft Entra ID to authenticate requests from SharePoint. You need to create an App Registration that the SPFx app will use to acquire access tokens.

Where: Azure Portal → App registrations → New registration


Configuration:

- **Name:** `Brightcove M365 Connector`

 **The App Registration name matters.** When the SharePoint admin approves the API permission (Section 5.3), the pending request will display this name. The request **will fail** unless what you put in for the App Registration's display name does not exactly match what is above.

- **Supported account types:** Choose based on your tenant topology:

Scenario	Setting
SharePoint and Azure are in the same Entra ID tenant	Single tenant only — Default Directory
SharePoint and Azure are in different Entra ID tenants	Multiple Entra ID tenants → then choose Allow all tenants or Allow only certain tenants

 **Which should I choose?** If your organization manages both Azure and SharePoint in a single Microsoft 365 tenant (the most common case), use **Single tenant only**. If the Azure subscription hosting the Function App belongs to a different Entra directory than your SharePoint tenant, you must use **Multiple Entra ID tenants** so that SharePoint users from the other tenant can obtain valid tokens.

- **Redirect URI:** Select **Web** as the platform and enter: `https://<your-tenant>.sharepoint.com/` (include the trailing slash). This redirect URI is required for the SPFx delegated auth flow.

Click **Register**.

Record the Application (client) ID — you will need this in Sections 4.6, 4.8, 5, and 6.

Microsoft Azure

Home > App registrations

Register an application

Name
The user-facing display name for this application (this can be changed later).

Brightcove M365 Connector ✓

Supported account types
Choose the account types that can use this application or access this API

Single tenant only - Default Directory Help me choose

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web https://your-tenant.sharepoint.com ✓

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

Essentials		Client credentials	
Display name	Brightcove M365 Connector	Add a certificate or secret	
Application (client) ID	1f960c8b-9b48-4738-a79c-3d800333511	Redirect URIs	1.web.0.app.0.public.client
Object ID	523a7f93-5814-486a-8006-08d114d016c	Application ID URI	api://1f960c8b-9b48-4738-a79c-3d800333511
Directory (tenant) ID	e48464c3-4b3e-4b3e-a320-e0901d161189	Managed application in L	Brightcove M365 Connector
Supported account types	Multiple organizations	State	Activated


4.6 Expose an API scope

The SPFx app needs permission to call the Proxy API on behalf of the signed-in user. You must expose an API scope on the App Registration.

Where: App Registration → **Expose an API**

- Set the Application ID URI:** Click **Set** next to "Application ID URI". Accept the default (`api://<application-client-id>`) or set a custom URI. Click **Save**.
- Add a scope:**
 - Click **Add a scope**.
 - Scope name:** `access_as_user`
 - Who can consent:** Admins only
 - Admin consent display name:** Access Brightcove M365 Connector Proxy API
 - Admin consent description:** Allow the Brightcove M365 Connector (SharePoint/Teams) to call the Brightcove Proxy API on behalf of signed-in users.
 - User consent display name:** Access Brightcove M365 Connector Proxy API

- **User consent description:** Allow the Brightcove M365 Connector to call the Brightcove Proxy API on your behalf.
- **State:** Enabled
- Click **Add scope**.

 The scope name **must** be `access_as_user`. This is the scope referenced in the SPFx package's API permission request. Using a different name will cause the permission approval to fail.

3. **Pre-authorize SharePoint Online:** This step allows SharePoint to silently acquire tokens for the Proxy API without requiring per-user consent popups.
 - Under **Authorized client applications**, click **Add a client application**.
 - **Client ID:** `00000003-0000-0ff1-ce00-000000000000` (this is the well-known application ID for SharePoint Online / Office 365).
 - Check the `access_as_user` scope.
 - Click **Add application**.

Add a scope



Scope name * ⓘ

access_as_user



api://17490c08-9bd8-4138-a78e-2d8603537a11/access_as_user

Who can consent? ⓘ

Admins and users

Admins only

Admin consent display name * ⓘ

Access Brightcove M365 Connector Proxy API



Admin consent description * ⓘ

Allow the Brightcove M365 Connector (SharePoint/Teams) to call the Brightcove Proxy API on behalf of signed-in users.

User consent display name ⓘ

Access Brightcove M365 Connector Proxy API



User consent description ⓘ

Allow the Brightcove M365 Connector to call the Brightcove Proxy API on your behalf.

State ⓘ

Enabled

Disabled

Add a client application



Client ID ⓘ

00000003-0000-0ff1-ce00-000000000000



Authorized scopes ⓘ



api://17490c08-9bd8-4138-a78e-2d8603537a11/access_as_user

Scopes defined by this API

Define custom scopes to restrict access to data and functionality protected by the API. An application that requires access to parts of this API can request that a user or admin consent to one or more of these.

Adding a scope here creates only delegated permissions. If you are looking to create application-only scopes, use 'App roles' and define app roles assignable to application type. [Go to App roles](#).

+ Add a scope

Scopes	Who can consent	Admin consent display ...	User consent display na...	State
api://4f382046-2164-45ef-b217-7ed4a8b261dd/acces...	 Admins only	Access Brightcove M365 ...	Access Brightcove M365 ...	Enabled

Authorized client applications

Authorizing a client application indicates that this API trusts the application and users should not be asked to consent when the client calls this API.

+ Add a client application

Client Id	Scopes
00000003-0000-0ff1-ce00-000000000000	1

4.7 Configure API permissions

Where: App Registration → **API permissions**

The default registration includes **Microsoft Graph** → **User.Read** (Delegated). This is sufficient — no additional permissions are required.

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for Default Directory

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				...
User.Read	Delegated	Sign in and read user profile	No	...


4.8 Configure CORS for the Function App

Where: Function App → **API** → **CORS**

Add your SharePoint Online origin(s) as allowed origins and click save:

- `https://<your-tenant>.sharepoint.com`



Do not use a wildcard () in production. Only add the specific SharePoint tenant URLs that need access.



Cross-Origin Resource Sharing (CORS) allows JavaScript code run that should be allowed to make cross-origin calls (for example: http Slashes are not allowed as part of domain or after TLD. [Learn more](#)

Request Credentials

☐ Enable Access-Control-Allow-Credentials ⓘ

Allowed Origins

https://portal.azure.com

https://yourtenant.sharepoint.com

4.9 Configure Function App environment variables

Where: Function App → **Settings** → **Environment Variables** → **App settings**

Add the following environment variables:

Name	Value	Description
KEY_VAULT_URL	https://<your-vault-name>.vault.azure.net/	Required. Full URL of the Key Vault created in 4.2.

Click **Apply and confirm** to save.

ird-eus | Environment variables ☆ ...

« App settings Connection strings

Search + Add Refresh Show values Advanced edit Pull reference values

Name	Value	Deployment slot setting	Source	Delete
APPLICATIONINSIGHTS_CONNECTIO...	Show value		App Service	
AzureWebJobsStorage	Show value		App Service	
DEPLOYMENT_STORAGE_CONNECTI...	Show value		App Service	
KEY_VAULT_URL	https://kv-bcvc-acme-prd-eus.vault.		App Service	

4.10 Deploy the Proxy API to the Function App

Deploy the `functionapp.zip` file to your Function App using one of these methods:

Option A — Azure CLI:

```
az functionapp deployment source config-zip \  
-g <resource-group-name> \  
-n <function-app-name> \  
--src functionapp.zip
```

Option B — Azure Portal:

Navigate to Function App → **Deployment** → **Deployment Center** and use Source: Publish files (new) to upload `functionapp.zip` and click save.

You can monitor the deployment and its status from the Logs tab.

Verify the deployment:


On the Overview blade verify you see **bcProxy** and **health** listed under the Functions tab and the status for each is enabled.

Then open a browser and navigate to:

```
https://<your-function-app-name>.azurewebsites.net/api/health
```

You should see a JSON response:

```
{  
  "ok": true,  
  "name": "brightcove-proxy-api",  
  "version": "1.0.0",  
  "time": "2026-03-24T..."  
}
```

 The `/api/health` endpoint is intentionally public and does not require authentication. All routes under `/api/proxy/*` are protected and require a valid bearer token.

⚠ You can obtain `<your-function-app-name>` from the overview blade and Default domain

eus | Deployment Center ☆ ...

Settings Logs

Save Discard Refresh Browse Sync Send us your feedback Troubleshoot

ⓘ You are now in the production slot, which is not recommended for setting up CI/CD. [Learn more](#)

Deploy and build code from your preferred source and build provider. [Learn more](#)

Source * Publish files (new) ▾

Publish files

Choose your ZIP package (maximum file size 2 gigabytes) using the form below. Then, click "Save" to upload. The site will automatically restart once the upload is complete. You can monitor progress in the "Logs" section. [Learn more](#)

functionapp.zip Browse

Functions Properties Notifications (0)

{ } Set up local environment Refresh

Filter by name...

Name	Trigger	Status	Monitor
bcProxy	HTTP	✓ Enabled	Invocations and more ...
health	HTTP	✓ Enabled	Invocations and more ...

5. SharePoint deployment

This section installs the SPFx package in your tenant and approves the API permission the package requests.

⚠ For cross-tenant setups (Azure app registration in a different tenant than SharePoint), you must modify the contents of the `brightcove-video-connector.sppkg` file prior to uploading to the Tenant App Catalog. Note this is a one time activity during the first installation. Future installations of new versions do not require you to modify the contents of the sppkg file.

1. Rename the file extension: `brightcove-video-connector.sppkg` → `brightcove-video-connector.zip`

2. Extract the zip and open `AppManifest.xml` .

3. Locate the `<WebApiPermissionRequests>` element and add your `AppId` and `ReplyUrl` :

```
<WebApiPermissionRequests>
  <WebApiPermissionRequest
    Resource="Brightcove M365 Connector"
    Scope="access_as_user"
    AppId="{your-azure-app-registration-client-id}"
    ReplyUrl="https://{your-tenant}.sharepoint.com" />
</WebApiPermissionRequests>
```

Replace:

- `{your-azure-app-registration-client-id}` — the Client ID from your Azure app registration's Overview page (created in section 4.5)
- `{your-tenant}.sharepoint.com` — your SharePoint root site URL

4. Save `AppManifest.xml` , recompress all extracted contents back into a zip (ensure `AppManifest.xml` is at the root of the zip, not inside a subfolder).

5. Rename back: `brightcove-video-connector.zip` → `brightcove-video-connector.sppkg`

6. Upload to the Tenant App Catalog and proceed with installation.

5.1 Ensure you have a Tenant App Catalog and add trusted script source

Where: SharePoint Admin Center → **More features** → **Apps** → **Open** → **App Catalog**

If your tenant does not have an App Catalog, create one (this is a one-time setup). See Microsoft's documentation on [managing the App Catalog](#).

Where: SharePoint Admin Center → **Advanced** → **Script Sources**

Click Add source and enter `https://players.brightcove.net/` for the Source expression.

Trusted script sources		
Content security policy helps minimize the risk of cross-site scripting attacks. SharePoint instructs the browser to only allow scripts from trusted sources.		
<div> <i>ⓘ</i> Violations to content security policy are logged in Purview. Starting later this year, content security policy for script sources will also be enforced. </div>		
<div> <div>+</div> Add source </div>		
Source expression	Status	Modified
<div> <input type="radio"/> https://players.brightcove.net/ </div>	Added from script sources	3/24/26, 5:33 PM


5.2 Upload and deploy the SPFx package

Where: App Catalog → **Apps for SharePoint** → **Upload**

1. Upload the file `brightcove-video-connector.sppkg` .
2. In the deploy dialog:
 - Click **Enable this app**.
 - Click **Go to API access page** to approve the pending request.

×

Enable app

 **Brightcove M365 Connector**

The app package has finished uploading. Would you like to enable the app now?

The app you're about to enable will have access to data by using the identity of the person using it. Enable this app only if you trust the developer or publisher.

This app gets data from:

- SharePoint

API access that must be approved after you enable this app

- Brightcove M365 Connector, access_as_user

Enabling this app makes the app available for site owners to add from the My apps page. The option to add this app automatically to all sites isn't available for this app.

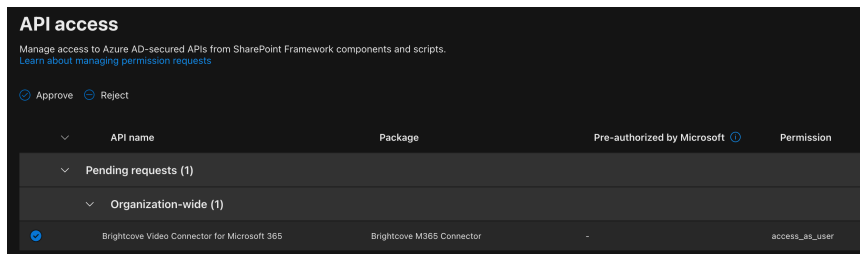
[Learn how to add an app to a site](#)

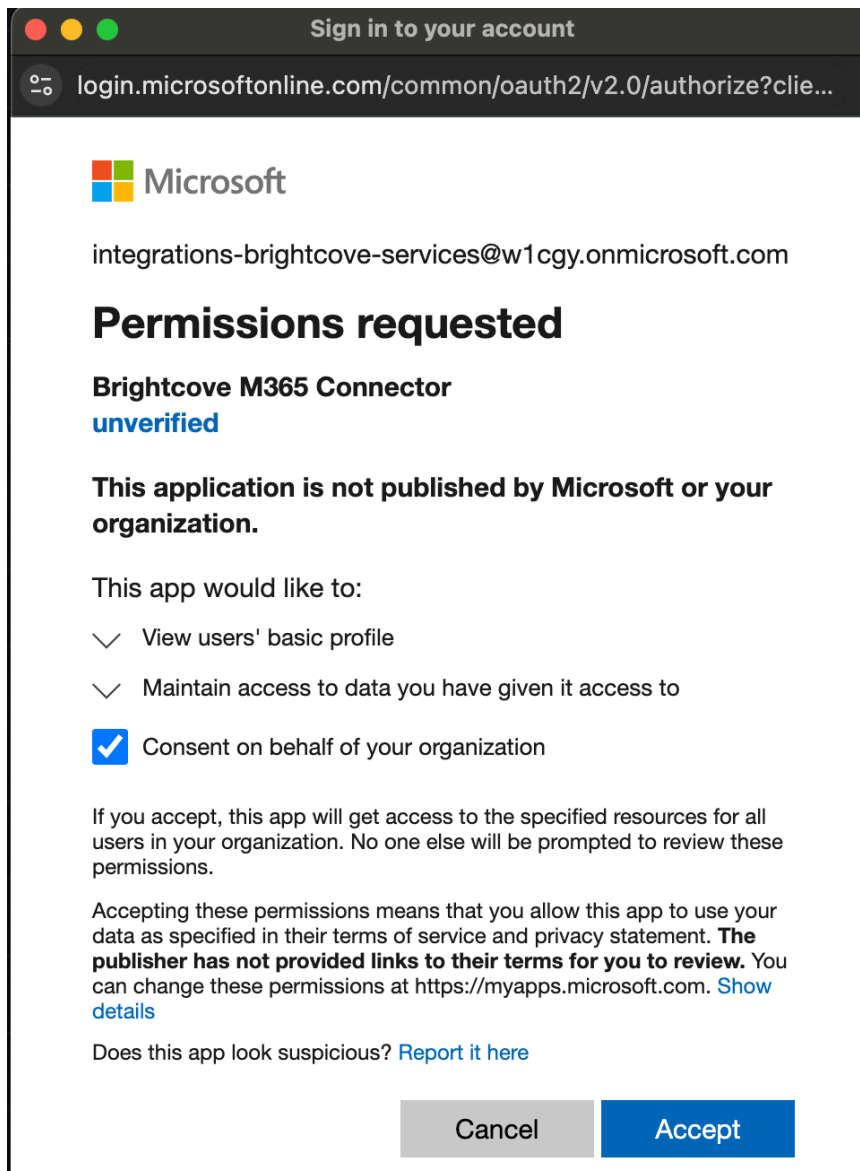
5.3 Approve the API permission (one-time)

The SPFx package requests delegated permission to call your Proxy API using the `access_as_user` scope.

Where: SharePoint Admin Center → **Advanced** → **API access**

1. You'll see a pending request for the Brightcove Proxy API (the name will match your App Registration name from Section 4.5).
2. Select the pending request.
3. Click **Approve**.
4. An admin consent popup will appear — complete the consent flow.





💡 If the pending request does not appear, verify that:

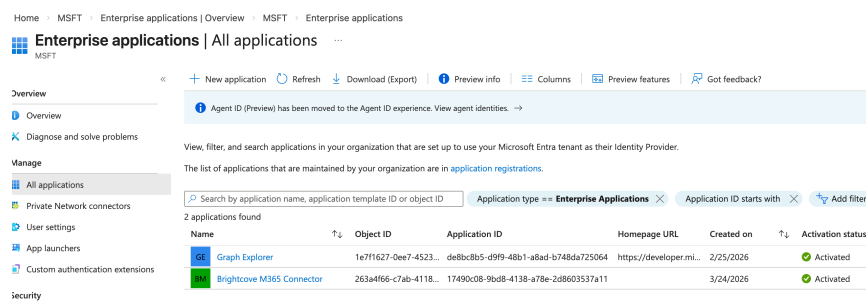
- The App Registration name in Entra matches exactly **Brightcove M365 Connector** .
- The **access_as_user** scope is properly configured on the App Registration.
- For cross-tenant setups, ensure the App Registration is set to **Multiple Entra ID tenants** and is either set to allow all tenants or allow only certain tenants with the SharePoint Entra ID tenant ID in the list of allowed tenants.

What happens behind the scenes: When you approve the API permission, SharePoint creates an **Enterprise Application** (service principal) in the SharePoint tenant's Entra ID directory. This enterprise app represents the Proxy API's App Registration in the SP tenant and is required for the delegated token flow to work.

5.3.1 Verify the Enterprise Application was created

After approval, confirm the Enterprise Application exists:

1. Go to **Microsoft Entra ID** (in the SharePoint tenant, if different from the Azure tenant) → **Enterprise applications**.
2. Search for the App Registration name (e.g., **Brightcove M365 Connector**).
3. You should see the enterprise app listed with the same Application ID as your App Registration.



⚠ Cross-tenant note: In a cross-tenant setup, this enterprise app will appear in the **SharePoint tenant's** Entra directory, not the Azure tenant where the App Registration was created. This is expected behavior — it's how Microsoft enables cross-tenant delegated access. If the enterprise app is missing, the SPFx app will fail to acquire tokens.

5.4 Add the app to a site

On the SharePoint site where you want to use the connector:

1. Go to **Settings** → **Add an app**.
2. Find and select the **Brightcove M365 Connector** app
3. Click **Add**.

When the app is added to a site, the following are automatically provisioned:

- Site pages:
 - **BrightcoveConnectorSettings.aspx** — The connector settings admin page.
 - **BrightcoveContentManagement.aspx** — The content management page for uploading and editing videos.
- Site contents:
 - **BrightcoveConfiguration list** — A hidden SharePoint list that stores proxy connection settings.

[← Back to All Company](#)

My apps

Filter

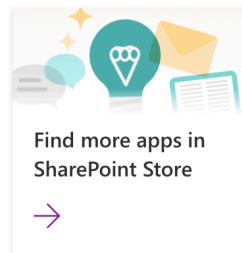
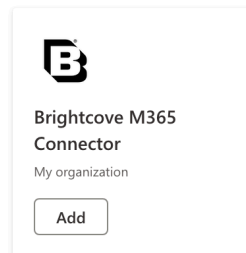
All

From my organization

From SharePoint Store

Apps you can add

These are SharePoint Store or custom apps allowed by your organization in the [classic experience](#).



AC All Company

Home
Conversations
Documents
Notebook
Pages
Site contents
Recycle bin
Edit

+ New

Contents

Subsites

	Name	Type	Items	Modified
	Documents	Document library	0	8/13/2025 10:08 PM
	Form Templates	Document library	0	9/1/2025 5:55 PM
	Site Assets	Document library	8	3/24/2026 3:53 PM
	Style Library	Document library	0	8/13/2025 10:08 PM
	BrightcoveConfiguration	List	0	3/24/2026 7:08 PM
	Brightcove M365 Connector	App		3/24/2026 7:08 PM
	Site Pages	Page library	3	3/18/2026 1:34 PM

Site Pages

All Pages

By Author

By Editor

Created By Me

»

+ Add view

▼

Name ▼

Created By ▼

▼

Created By: System Account (3)

Home.aspx

System Account

BrightcoveConnectorSettings.aspx

System Account

BrightcoveContentManagement.aspx

System Account

6. Initial configuration in SharePoint

Use the Brightcove Connector Settings page to connect SharePoint to your Proxy API and register your Brightcove account credentials.

6.1 Navigate to Connector Settings

Where: Site → **Site Pages** → **BrightcoveConnectorSettings.aspx**

You can find this page in the site's Pages library. It was automatically created when you added the app to the site in Section 5.4.

Brightcove Connector Settings

Brightcove Connector Settings

Configure proxy settings and manage Brightcove Video Cloud accounts for SharePoint.

Proxy Configuration Accounts

Azure Function Proxy API Base URL

https://fa-bcvc-acme-prd-eus-c3ckf2vd5e4cyeg.canadacentral-01.azurewebsites.net/api/proxy

Proxy API Resource (App ID URI)

api://17490c08-9bd8-4138-a78e-2d8603537a11

This should be the **base URL** of the Brightcove Proxy API and must end with /api/proxy. [How do I find this URL?](#)

Save

Test Connection

6.2 Configure proxy connection

On the **Proxy Configuration** tab, enter the following:

Field	Value
Proxy API Base URL	<code>https://<your-function-app-name>.azurewebsites.net/api/proxy</code> (the default domain from Section 4.5)

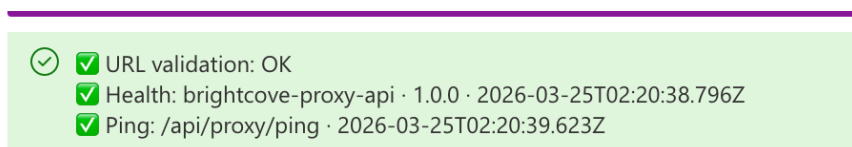
Field	Value
Proxy API Resource	<code>api://<application-client-id></code> (the Application ID URI from Section 4.6)


Click **Save**, then click **Test Connection** to verify connectivity.

The test performs two checks:

1. **Health check** — Calls `GET /api/health` (public endpoint) to verify the Function App is reachable.
2. **Authenticated ping** — Calls `GET /api/proxy/ping` (protected endpoint) to verify that token acquisition and JWT validation are working end-to-end.

If both checks pass, the proxy connection is configured correctly.



 **Troubleshooting:** If the test fails, the settings page provides a **Copy Support Bundle** button that captures diagnostic information (proxy URL, status codes, error details, browser info). Use this to share details with your Azure admin or Brightcove support.

6.3 Add Brightcove accounts

Switch to the **Accounts** tab.

1. Click **Add**.
2. Enter the credentials you recorded in Section 3:
 - **Name:** A friendly label for this account (alphanumeric and hyphens only; 4–64 characters; must start and end with a letter or number).
 - **Account ID:** Your Brightcove numeric Account ID.
 - **Client ID:** The Brightcove API Client ID.
 - **Client Secret:** The Brightcove API Client Secret.

3. Click **Test and Save**.

Repeat for each Brightcove account you want to make available on this site.

Add Account

Create a Brightcove account connection for this Microsoft 365 tenant.

Name *

Video-Cloud

Account ID *

6415866131001

Client ID *

38a0bc45-ebdd-4293-82b5-efbaee820786

Client Secret *

.....



Cancel

Test and Save

Test Account

Test results for "Video-Cloud".

Overall Status

Pass

Tested At: 3/24/2026, 7:26:24 PM

Checks

OAuth token

Pass

✓ Successfully retrieved an OAuth access token.

CMS API read probe

Pass

✓ Successfully completed a CMS API read probe.

CMS API write probe

Pass

✓ Successfully completed a CMS API write probe.

Players API read probe

Pass

✓ Successfully completed a Players API read probe.

In-Page Experiences API read probe

Pass

✓ Successfully completed an In-Page Experiences API read probe.

Dynamic Ingest API write probe

Pass

✓ Successfully completed an Dynamic Ingest API write probe.

Close

⚠ Account scoping: Accounts are scoped by the Entra tenant ID of the SharePoint user. The Proxy API extracts the `tid` claim from the user's JWT to determine which accounts are visible. This means accounts added by an admin in Tenant A are only visible to users from Tenant A.

6.4 Validate

After saving your proxy settings and adding at least one Brightcove account:

1. Navigate to **Site Pages** → **BrightcoveContentManagement.aspx**.
2. Verify that the **Brightcove account dropdown** displays the account(s) you just configured.
3. If the account appears and the page loads without errors, the installation is complete. 🎉👏

Brightcove Content Management

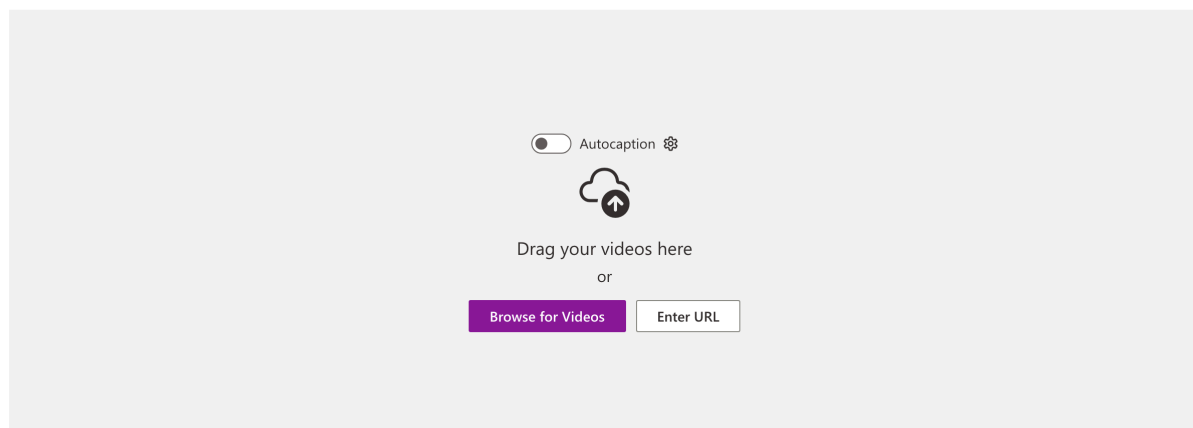
Content Management

Account

Video-Cloud (6415866131001) 

Upload

Videos are available to edit in [Brightcove Studio](#)



7. Verify web parts

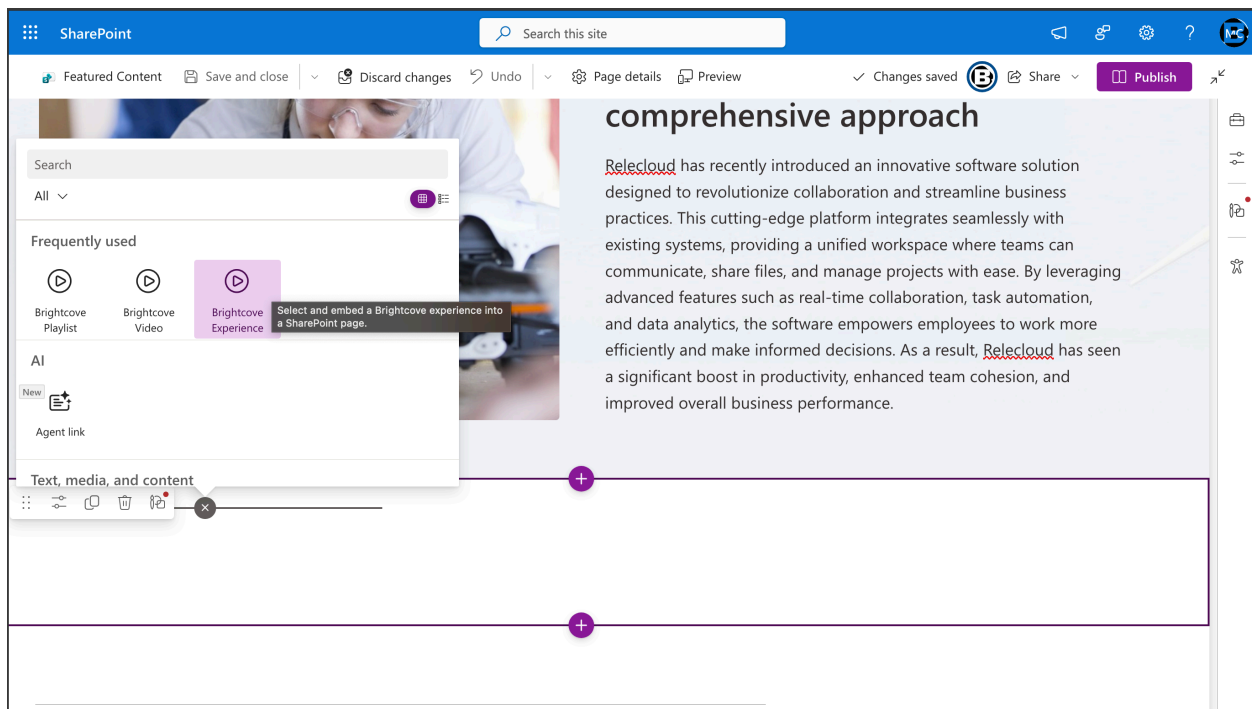
The connector includes three authoring web parts that appear in the SharePoint web part toolbox when editing a page.

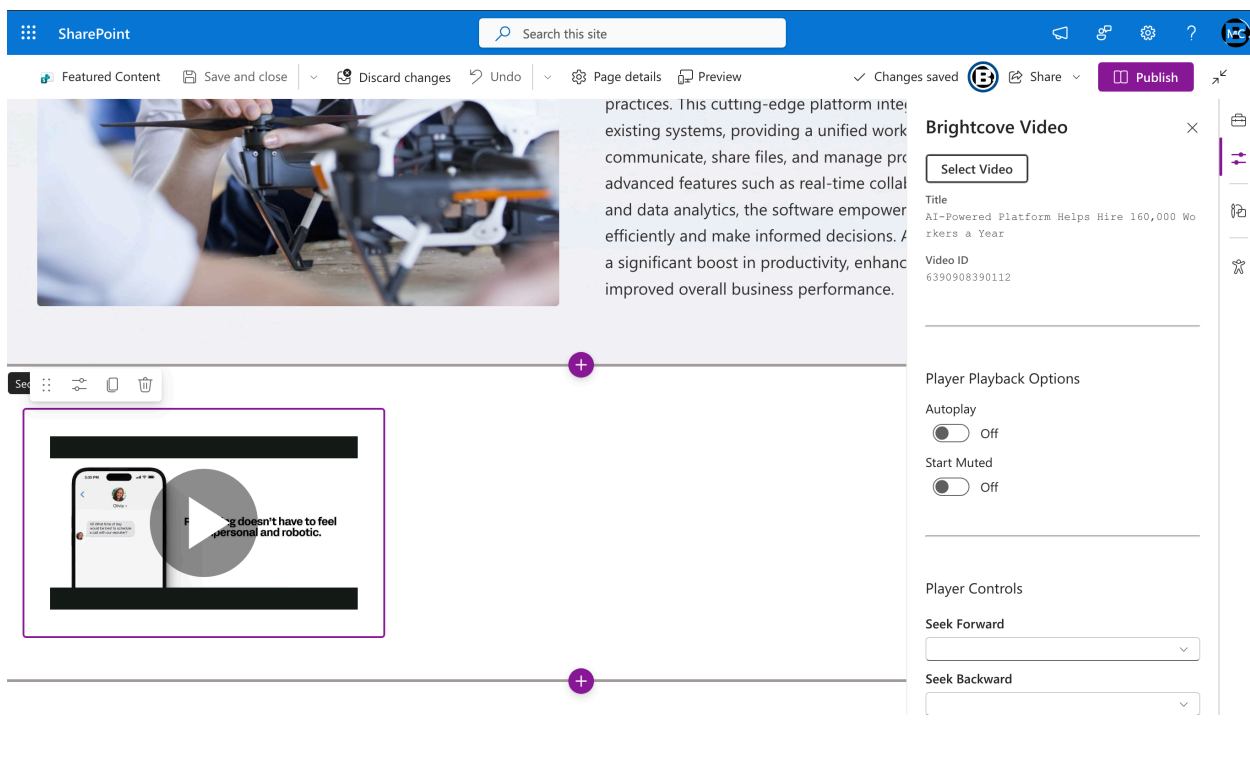
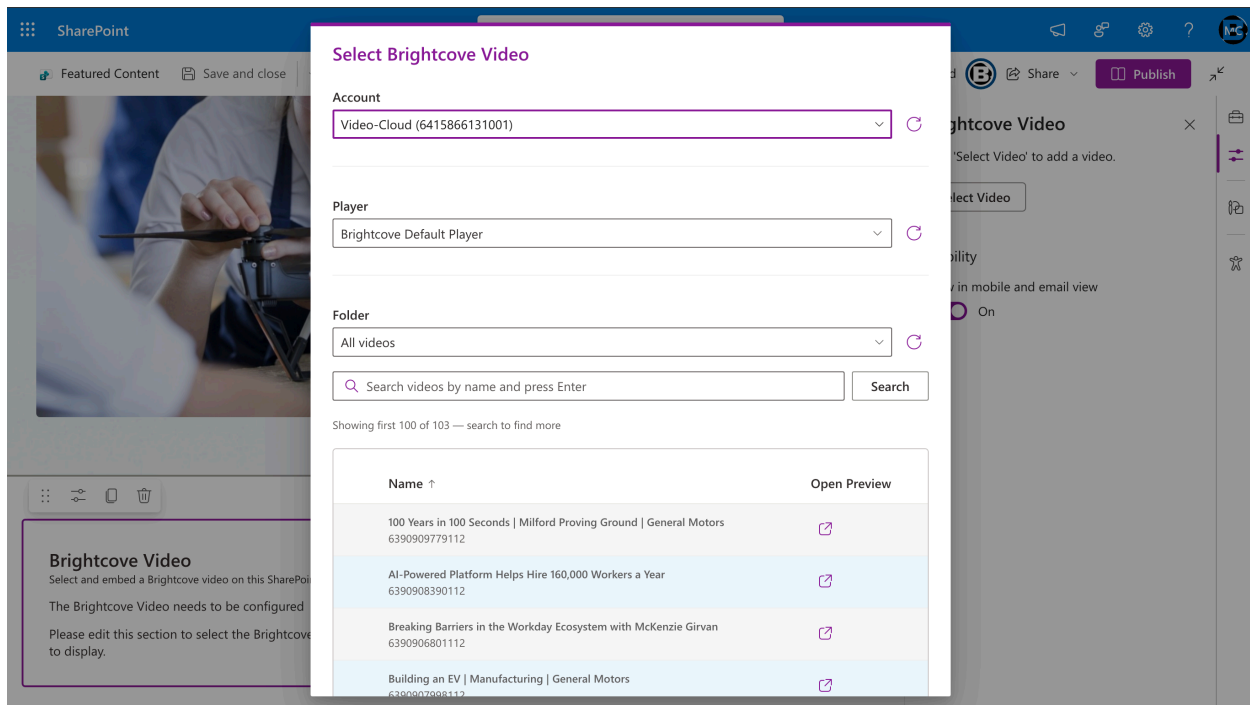
7.1 Add a Brightcove web part to a test page

1. Navigate to a SharePoint page and click **Edit**.
2. Click the **+** button to add a web part.

3. Search for **Brightcove** in the toolbox.
4. You should see three web parts:
 - **Brightcove Video** — Embed a single video.
 - **Brightcove Playlist** — Embed a playlist.
 - **Brightcove Experience** — Embed an In-Page Experience (IPX).
5. Add one of the web parts to the page.
6. Open the web part's property pane and select your Brightcove account from the dropdown.
7. Browse and select content from your Brightcove library.
8. Click **Publish** (or **Republish**) to save the page.

If the web parts appear in the toolbox, content loads from your Brightcove account, and the embedded player renders on the published page, the installation is fully verified.





8. Tenant topology reference

This section provides additional detail on same-tenant versus cross-tenant configurations.

8.1 Same tenant (most common)

In this scenario, both the Azure subscription (hosting the Function App and Key Vault) and SharePoint Online belong to the **same Microsoft Entra ID directory**.

- App Registration: **Single tenant only — Default Directory**
- Token issuer: Tokens from SharePoint users will have an `iss` claim matching the same tenant ID as the App Registration.
- This is the simplest configuration and requires no additional tenant allowlisting.

8.2 Cross-tenant

In this scenario, the Azure subscription belongs to a **different Entra ID directory** than the SharePoint Online tenant. This can occur when:

- A customer's Azure infrastructure is managed by a different team or subsidiary with its own directory.
- A Brightcove partner hosts the Proxy API on behalf of multiple customers.

Configuration differences:

- App Registration: **Multiple Entra ID tenants** → **Allow all tenants** (or **Allow only certain tenants** for tighter control).
- The Proxy API validates tokens using the Microsoft common JWKS endpoint, which accepts tokens issued by any Entra tenant. Tenant scoping is enforced at the application level — the proxy extracts the `tid` claim from the JWT and uses it to scope Brightcove credential lookups in Key Vault.
- The SharePoint admin must still approve the API permission in their tenant's SharePoint Admin Center (Section 5.3). This consent flow works across tenants as long as the App Registration allows multi-tenant access.

9. Troubleshooting

Proxy health check fails

- Confirm the Function App is running in the Azure Portal (check the **Overview** page for status).

- Verify the URL is correct: `https://<function-app-name>.azurewebsites.net/api/health`.
- Open the health URL directly in a browser to check for DNS or TLS issues.

Authenticated ping fails (401 or 403)

- **401 Unauthorized:** The token was not accepted. Check:
 - `JWT_AUDIENCE` in the Function App environment variables matches the Application ID URI on the App Registration.
 - The API permission was approved in SharePoint Admin Center (Section 5.3).
 - CORS is configured with the correct SharePoint origin URL.
 - The redirect URI `https://<tenant>.sharepoint.com/` is configured on the App Registration (Section 4.5).
 - The Enterprise Application exists in the SharePoint tenant's Entra ID (Section 5.3.1).
- **403 Forbidden:** The token was valid but missing required scopes. Check:
 - The `access_as_user` scope is defined under **Expose an API** in the App Registration.
 - SharePoint Online (`00000003-0000-0ff1-ce00-000000000000`) is listed as a pre-authorized application for the `access_as_user` scope (Section 4.6).
 - The SPFx package's permission request was approved.

consent_missing or resource_mismatch errors in the browser

These errors appear in the SPFx Connector Settings "Test Connection" results when the token acquisition fails before any request reaches the Proxy API.

- **consent_missing:** The Enterprise Application was not created in the SharePoint tenant. Re-approve the API permission in SharePoint Admin Center, or have a Global Admin grant consent manually via Entra ID → Enterprise Applications.
- **resource_mismatch:** The resource identifier in the SPFx package does not match the App Registration. Ensure the App Registration name matches the

`resource` value in `webApiPermissionRequests` in the SPFx package. If using a custom-named App Registration, the SPFx package may need to be rebuilt with the matching resource name.

Viewing Function App logs without Application Insights

If Application Insights is not enabled, you can still access basic logs:

1. **Live log stream:** Function App → **Monitoring** → **Log stream**. Shows real-time console output from the running function (including `console.log`, `console.error`, and uncaught exceptions).
2. **Filesystem logs:** Enable diagnostic logs under Function App → **Monitoring** → **Diagnostic settings** and route to a Storage Account if persistent logging is needed.
3. **Kudu console:** Navigate to `https://<function-app-name>.scm.azurewebsites.net` → **Log Files** to browse filesystem logs directly.

For production environments, Application Insights is recommended as it provides structured request tracing, dependency tracking, and failure analysis that filesystem logs cannot match.

Web parts don't appear in the toolbox

- Confirm the SPFx package was deployed with **Enable this app** checked.
- Confirm the app was added to the specific site (Section 5.4).
- Try refreshing the page or clearing the browser cache.

Brightcove accounts don't appear in the dropdown

- Verify that at least one account was added and saved in Connector Settings (Section 6.3).
- Check that the proxy connection test passes.
- If using a cross-tenant setup, ensure the App Registration supports multi-tenant access and the `tid` claim in the token matches the tenant used when the accounts were added.

Key Vault access errors

- Confirm the Function App's managed identity is enabled (Section 4.4).
 - Verify the **Key Vault Secrets Officer** role is assigned to the managed identity on the Key Vault.
 - Check that `KEY_VAULT_URL` in the Function App environment variables points to the correct Key Vault.
-

10. Summary of provisioned resources

After completing this guide, you will have the following resources:

Azure

- Resource Group
- Key Vault (stores Brightcove API credentials as encrypted secrets)
- Function App (runs the Proxy API — Node.js 22, Linux, Flex Consumption plan)
- Storage Account (auto-provisioned for Function App deployment packages)
- Enterprise Application / Service Principal (created in the SharePoint tenant's Entra ID when API permission is approved)
- App Registration (enables Entra ID authentication between SharePoint and the Proxy API)

SharePoint

- SPFx package deployed to Tenant App Catalog
- API permission approved for the Proxy API
- Per-site: Brightcove Connector Settings page (`BrightcoveConnectorSettings.aspx`)
- Per-site: Brightcove Content Management page (`BrightcoveContentManagement.aspx`)
- Per-site: BrightcoveConfiguration list (stores proxy URL and API resource)
- Three authoring web parts available in the toolbox (Video, Playlist, Experience)