



Connector Hub Implementation Handbook

Version 4.5

Table of Contents

[Pre-Setup: Connector Hub Project Plan](#)

[Uplevel Architecture:](#)

[Connector Hub Setup Requirements](#)

[Data Source Connection Requirements](#)

[Other Requirements:](#)

[Org Chart:](#)

[Setting Up the VM & Connector Hub](#)

[Prerequisites](#)

[Shell Script: Install Connector Hub](#)

[Logging in to the Connector Hub](#)

[Using Uplevel Okta](#)

[Using OKTA as OIDC Identity Provider](#)

[Using Azure AD as OIDC Identity Provider](#)

[Connecting Individual Data Sources](#)

[Messaging](#)

[Setting Up the Slack Standard/Business+ Connector](#)

[Setting Up Slack Enterprise Grid Connector](#)

[Setting Up Microsoft Teams Connector](#)

[Calendar](#)

[Setting Up Google Calendar Connector](#)

[Setting up Google Calendar Connector - OAuth](#)

[Setting Up O365 Connector](#)

[Work Management Tools](#)

[Setting Up Jira Cloud Connector](#)

[Note: you will need an understanding of the Jira projects you'd like to include in the Uplevel analysis.](#)

[Setting Up Jira OnPrem Connector](#)

[Source Code](#)

[Setting Up Github Cloud/OnPrem Connector using PAT](#)

[Setting Up Github Cloud/OnPrem Connector using Github Apps](#)

[Setting Up Gitlab Cloud/OnPrem Connector](#)

[Setting Up Gerrit Connector](#)

[Setting Up Bitbucket Cloud Connector](#)

[Setting Up Bitbucket OnPrem Connector](#)

[Final Step: Push Data to Uplevel](#)

[Appendix: Credential Encryption](#)

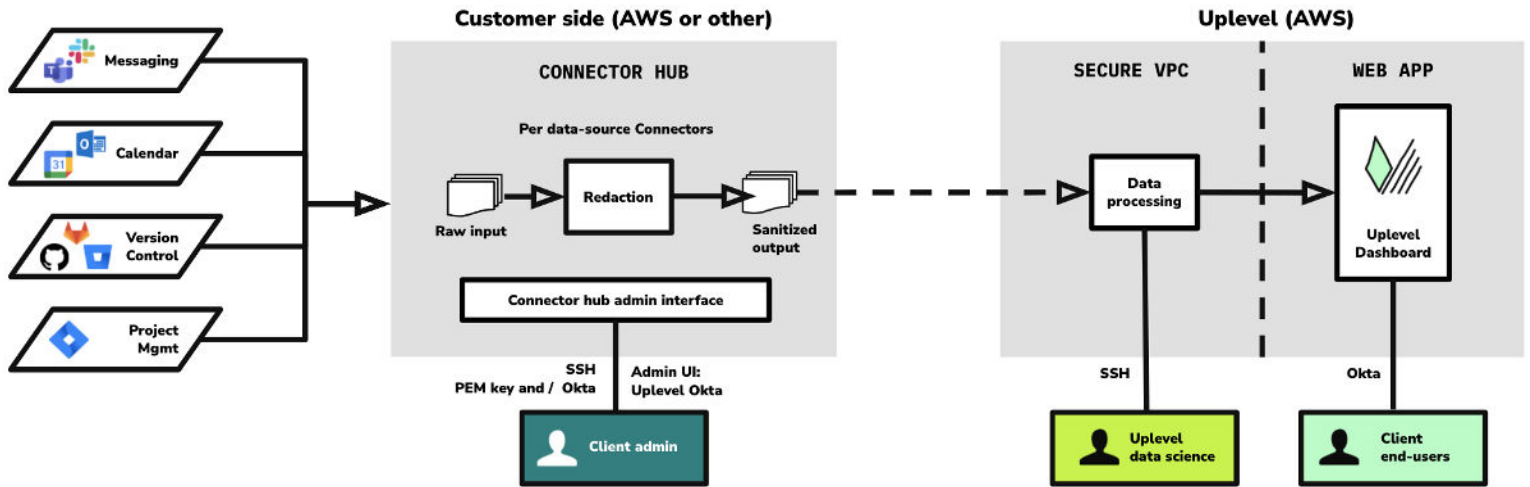
Pre-Setup: Connector Hub Project Plan

Below is an overview of the steps needed to complete the data connection process with Uplevel. We've outlined approximate timing for each step and who needs to be involved. Below the timeline, you'll find all the requirements and instructions, including the credentials and user privileges, needed for each data source.

Task	Owner	Time
Kick off meeting with IT	IT team + Uplevel	60 min meeting
Set up VM	IT team	Approx 1 hour of work
Install Uplevel Connector Hub	IT team + Uplevel	60 min
Connect data sources	Data Source Admins*	1 hour per data source
Upload org chart	IT team	15 mins
Push data to Uplevel	IT team	2 mins

* credential requirements differ per data source - please see below for the full list of instructions

Uplevel Architecture:



Connector Hub Setup Requirements

Either an AWS Linux t2.xlarge instance or VM with the following specifications:

- Recommended server hardware spec and network access:
 - OS: Linux
 - CPU: 4 cores
 - RAM: 16GB
 - DISK: 100GB
 - Ingress (inbound traffic): Only TCP Port 22 for SSH
 - Egress (outbound traffic): Any 0.0.0.0/0 IP
- [Ensure Docker engine and compose are installed.](#)

Data Source Connection Requirements

Messaging:

- Slack Standard and Business+: Workspace Owner
- Slack Enterprise Grid: Workspace Owner
- Microsoft Teams: Administrator

Calendar:

- Gmail: Administrator
- O365: Administrator
- Microsoft Exchange: Administrator
- .csv file with email addresses for org

Project Management:

- Jira Cloud: Administrator
- Jira OnPrem: Administrator
- Understanding of which Jira projects to track

Git:

- Github Cloud: Administrator
- Github OnPrem: Administrator
- GitLab Cloud: Administrator
- GitLab OnPrem: Administrator
- BitBucket Cloud: Administrator
- BitBucket OnPrem: Administrator
- Gerrit: Administrator
- Understanding of which repos to track

Other Requirements:

Org Chart:

- **Format:** .csv file in the following format:
email
first_name
last_name
job_title
department
manager_email
office_location - optional
reporting_group - optional

Setting Up the VM & Connector Hub

As an IT administrator, you can install the Connector Hub manually via shell script on a dedicated VM.

Prerequisites

Your Uplevel contact will provide you with the following. You'll need these during setup.

- 1.) Client ID (typically this is your company domain name)
- 2.) Vault Passcode (for encrypting credentials locally typically sent via pwpush.com)
- 3.) License file (a *.json file)

Shell Script: Install Connector Hub

1. **Spin up a dedicated linux VM with SSH access.** See recommended spec below. (Please follow your organization's standard operating procedure.)

- **Recommended server hardware spec and network access:**
 - OS: Linux/Unix, CentOS 7+
 - CPU: 4 cores
 - RAM: 16GB
 - DISK: 100GB
 - Ingress (inbound traffic): Only TCP Port 22 for SSH
 - Egress (outbound traffic): Any 0.0.0.0/0 IP

(The above instance type is similar to an Amazon Linux t2.xlarge.)

SSH Session Security: For SSH to utilize a stronger encryption for server/client communication, please consider updating the SSH server configuration in your VM, replacing Cipher Block Chaining, CBC mode support with a more secure alternative (e.g. CTR or GCM). [Here is some reference documentation.](#)

2. **SSH with local port forwarding into the VM.** (Note: If using Windows as your local machine, please use [PuTTY](#) to port forward with SSH)

For Mac/Linux, find an example SSH command with port forwarding. To use the below example, you will need to replace the `ec2-user`, `internal-ip` and `external-ip` as well as the location to `keypair.pem` pem/rsa file with your own):

```
# SSH command with localhost port forwarding
$ ssh -i keypair.pem -L 127.0.0.1:8080:internal-ip:8080 ec2-user@external-ip
```

3. Install Docker. The Connector Hub relies on docker engine/compose running in the VM. [Please make sure a docker engine and compose are installed.](#)

```
# Here is a helper script for a Centos 7 VM - copy and run both rows
$ sudo curl -fSO https://intelli-learn-uplevel.s3-us-west-2.amazonaws.com/connector/centos-docker-compose-setup.sh

sudo sh centos-docker-compose-setup.sh
$
```

4. Install the Uplevel Connector Hub. (Note: You will be prompted to enter your ClientID and Vault Password during this step)

```
# Test docker engine/compose installation - should print version number
$ sudo docker --version
$ sudo docker-compose --version
$ sudo docker info

# Download the latest Connector Hub script - copy and run both rows
$ sudo curl -fSO https://intelli-learn-uplevel.s3-us-west-2.amazonaws.com/connector/start-uplevel-connector-hub.sh

# Run the startup script as sudo
sudo sh start-uplevel-connector-hub.sh
$
```

5. After installation is completed, navigate to <http://localhost:8080> from your local machine while keeping the SSH session alive and upload the provided license file.

Logging in to the Connector Hub

Once the Connector Hub is set up, there are multiple options to access once you navigate to <http://localhost:8080>. By default, you will be met with Uplevel's Okta login screen, but you have the option to use your own Okta instance or Azure AD. The steps for each option are outlined below.

Using Uplevel Okta

Once the Connector Hub is set up, contact your Uplevel Representative to provide you login credentials via Okta. You will be sent an activation email to set up your credentials. Once they

are set up, you can use those credentials to log into the Connector Hub after navigating to <http://localhost:8080> .

Using OKTA as OIDC Identity Provider

To integrate with your Okta instance, follow the instructions below:

1. Configure an App in OKTA

- Login to OKTA Admin console
- Select Applications and click on Add Application.
- Select **Web** as the application platform type and click Next
- Name the application: (Uplevel Connector Hub)
- Base URL (optional): IP/DNS address of your onprem Uplevel Connector Hub
- Enter Login Redirect URL: <https://uplevelteam.okta.com/oauth2/v1/authorize/callback>
- Assign a Group.
 - Be sure to verify the users that need access are assigned to the selected group.
- Click “Done”

2. **Share Client Credentials information with Uplevel** (we will configure them into our identity provider service - OKTA):

- OIDC Client ID
- OIDC Client Secret
- Your OKTA org domain or Discovery URL.
 - Example: [https://\[theOktaIdPOrg\]/well-known/openid-configuration](https://[theOktaIdPOrg]/well-known/openid-configuration)

Once your Uplevel Representative has confirmed we have completed the configuration on our side, using the information you provided from step 2, you can continue with the next step.

3. Create an ‘**okta.env**’ text file (please note the ‘**.env**’ extension).

- Copy/paste the below template into the okta.env file and fill in the **<variables>** with items from your OKTA application
- Upload the completed file into the user root directory of the VM hosting the Connector Hub.

Copy/paste template:

```
OKTA_ORG_URL=<Okta domain>
OKTA_CLIENT_ID=<Client ID>
OKTA_CLIENT_SECRET=<Client secret>
OKTA_CALLBACK_URI=<Sign-in redirect URIs>
```


Completed example:

```
OKTA_ORG_URL=https://dev-984437.oktapreview.com
OKTA_CLIENT_ID=0oan42hm*****
OKTA_CLIENT_SECRET=B3CK*****
OKTA_CALLBACK_URI=http://localhost:8080/users/callback
```

4. Update the Connector Hub. Updating will pick up and use the uploaded okta.env file.

- SSH with Port forwarding into the VM hosting the connecting hub as usual.
- Run the below shell command (Note: You will be prompted to enter your ClientID and Vault Encryption Passcode)

```
$ sudo sh upgrade.sh
```

5. Open browser and access Connector Hub. It should redirect you to your OKTA login page if not already signed in via SSO (please try using an incognito window if you are experiencing any redirecting and/or login issues).

- If you are accessing via `http://localhost:8080`, you will still need to SSH with port forwarding as usual.
- If you are using an internally accessible URL, you do not need to SSH with port forwarding (provided you've allowed HTTP traffic to the VM).

Using Azure AD as OIDC Identity Provider

You can use Azure AD to manage authentication and authorization for the Uplevel Connector Hub by registering an application in your Azure portal.

1. Register an enterprise app in azure portal

- Go to: <https://portal.azure.com> then click App Registrations
 - Enter Name: **Uplevel Connector Hub**
 - Enter Redirect URI (Web):
 - **`https://uplevelteam.okta.com/oauth2/v1/authorize/callback`**
 - Check (x) Single-tenant as the supported account type
 - Click [Register] button

Register an application

* Name

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

Uplevel Connector Hub 

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (UpLevel, Inc only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[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://uplevelteam.okta.com/oauth2/v1/authorize/callback> 

By proceeding, you agree to the [Microsoft Platform Policies](#) 

[Register](#)

2. In the Overview menu:

- Copy Application Client ID and send it to your Uplevel Representative
- Click [Endpoints] button
- Copy the OpenID Connect metadata document URL and provide to your Uplevel Representative

Search (Cmd+*f*) <<

 Delete  Endpoints

- Overview
- Quickstart
- Integration assistant (preview)
- Manage**
- Branding
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- Owners
- Roles and administrators (Preview)
- Manifest
- Support + Troubleshooting**
- Troubleshooting
- New support request

Display name
Uplevel Connector

Supported account types
Multiple organizations

Application (client) ID
89064610-4e18-40b6-b000-000000000000



Redirect URIs
1 web, 0 spa, 0 public client

Directory (tenant) ID
97a8075c-3f55-4c5e-8000-000000000000

Application ID URI
Add an Application ID URI

Object ID
553fb4a8-5016-421f-b40f-b8adad8fcba6

Managed application in local directory
Uplevel Connector

 Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#) 

Call APIs



Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

[View API permissions](#)

Documentation

- [Microsoft identity platform](#)
- [Authentication scenarios](#)
- [Authentication libraries](#)
- [Code samples](#)
- [Microsoft Graph](#)
- [Glossary](#)
- [Help and Support](#)

3. In the certificates & secrets menu:

- Click [New client secret] button
- Enter Description: Uplevel
- Check (x) Never
- Click on [Add] button
- Copy Client Secret value and provide to your Uplevel Representative

Search (Cmd+/)

- Overview
- Quickstart
- Integration assistant (preview)
- Manage**
- Branding
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- Owners
- Roles and administrators (Preview)
- Manifest
- Support + Troubleshooting**
- Troubleshooting
- New support request

Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this blade.

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

Thumbprint	Start date	Expires
No certificates have been added for this application.		

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

New client secret

Description	Expires	Value
Password uploaded on Fri Jul 31 2020	12/31/2299	w80Z5Rbgrm_H-jAQ: [REDACTED]

4. Assign Group/Users to the registered application

- Be sure to verify the users that need access to the Uplevel Connector Hub are assigned to the required group

5. Share Client Credentials information with Uplevel (We will configure them into our identity provider service - OKTA):

- Application Client ID
- Application Client Secret
- Application OpenID Connect metadata document URL

Reference Links:

- [Okta/Azure Federation](#)
- [Microsoft Quickstart Guide](#)

Connecting Individual Data Sources

Once you've logged into the Connector Hub, the next step is to connect the individual data sources that your company uses. As a best practice, Uplevel recommends that service accounts are set up for each integration.

Messaging

We recommend setting up a service account to connect your Messaging tool. The data points that will be passed to Uplevel for processing are:

- Channel & DMs (meta only): Time, sender, length, emojis, wordcount, @mentions

Setting Up the Slack Standard/Business+ Connector

0. All Slack Business+ plans have a "Standard Export" ability by default. To export private channel and DM data, fill out the Slack application. For more details on Slack exports, take a look at [Slack's help documentation](#).

Export Data

Import Export

You may be eligible to export additional data.
To request the ability to export all channels and conversations, submit an application.

Messages and files can be exported from your workspace with a Standard Export.

Here's what's included:

- Messages and file links sent in public channels

What's not included:

- Messages and file links sent in private channels
- Messages and file links sent in direct messages
- Editing and deletion logs

For information on the format of channel messages, please [consult our API docs](#).

Export date range

Choose one... Start Export

Schedule Exports

You can schedule exports for your workspace's history on a daily, weekly or monthly basis. You will receive an email with a link to your export file when it is ready.

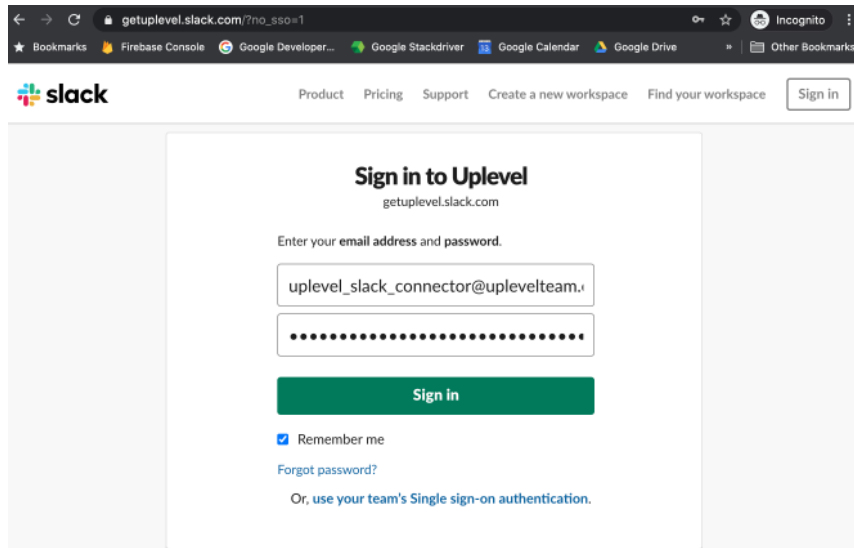
Frequency

Never Save

Daily exports will be run every night, weekly every Monday, and monthly on the 1st of each month.

1. Create an “Uplevel” Slack Workspace Owner account. This account can be named anything, but we recommend something close to “Uplevel Service Account”. Make sure 2FA and SSO are both disabled.

- A Workspace Owner account can get around an SSO requirement by navigating to this link: https://<workspace>.slack.com/?no_sso=1 and selecting **Forgot Password**.
- Follow the link in the email to establish a password and use that password to connect.



- Confirm the service account authentication type by going to your slack admin page: <https://<workspace>.slack.com/admin> and make sure it is a Workspace Owner account with SSO and 2FA turned off.

The screenshot shows the Slack Admin console interface. At the top, there's a navigation bar with 'Menu' and 'Uplevel' logos. Below that, the 'Manage members' section is visible, featuring buttons for 'Export Member List' and 'Invite People'. A search bar for 'Search current members' and a 'Filters (1)' dropdown are also present. The main content is a table of members with columns for Name, Account type, Billing status, and Authentication. Two members are listed: 'stewart' (Workspace Owner, Active, SSO) and 'Uplevel Slack Connector (you)' (Workspace Owner, Inactive, Default). The 'Uplevel Slack Connector' row is highlighted with a red border, and the 'Default' authentication method is circled in red.

Name ↑	Account type	Billing status	Authentication
stewart stewart@uplevelteam.com	Workspace Owner	Active	SSO
Uplevel Slack Connector (you) Uplevel Slack Connector • uplevel_slack_con	Workspace Owner	Inactive	Default

2. Sign in to Slack with the Uplevel service account and go to your export page (<https://<workspace>.slack.com/services/export>). You will perform 2 exports here.

- Set up a **one year export**. Under **Export date range**, select the **Specific date range** option (see image below). Select one year's worth of data and click **Start Export**.

The screenshot shows the Slack 'Export Data' page. The 'Export' tab is active. A blue information box at the top states: 'Exports of all channels and conversations are now available. Workspace Owners can now perform exports of all channels and conversations. [Learn more](#)'. Below this, text explains that as a Workspace Owner, you can export all messages and files from your workspace. Two sections follow: 'Here's what's included:' (Messages sent in public and private channels, Messages sent in direct messages, File links shared in public and private channels, File links shared in direct messages) and 'What's not included:' (Editing and deletion logs). A link to 'consult our API docs' is provided for more information. At the bottom, the 'Export date range' section shows a dropdown menu with 'Specific date range...' selected. Below the dropdown, a date range is specified: 'Starting April 4th, 2021' and 'Ending April 4th, 2022', both highlighted with red boxes. A green 'Start Export' button is located to the right of the date range.

(Image shows Slack Business+. Please note that Slack standard will look slightly different)

- Schedule a **daily recurring export** by setting the frequency to “Daily” then clicking “Save” [Continued on next page]

As a Workspace Owner, you can export all messages and files from your workspace – including private channels and direct messages. Only Workspace Owners may export this data.

Here's what's included:

- Messages sent in public and private channels
- Messages sent in direct messages
- File links shared in public and private channels
- File links shared in direct messages

What's not included:

- Editing and deletion logs

For information on the format of channel messages, please [consult our API docs](#).

Export date range

Choose one...

Schedule Exports

You can schedule exports for your workspace's history on a daily, weekly or monthly basis. You will receive an email with a link to your export file when it is ready.

Frequency

Daily

Daily exports will be run every night, weekly every Monday, and monthly on the 1st of each month.

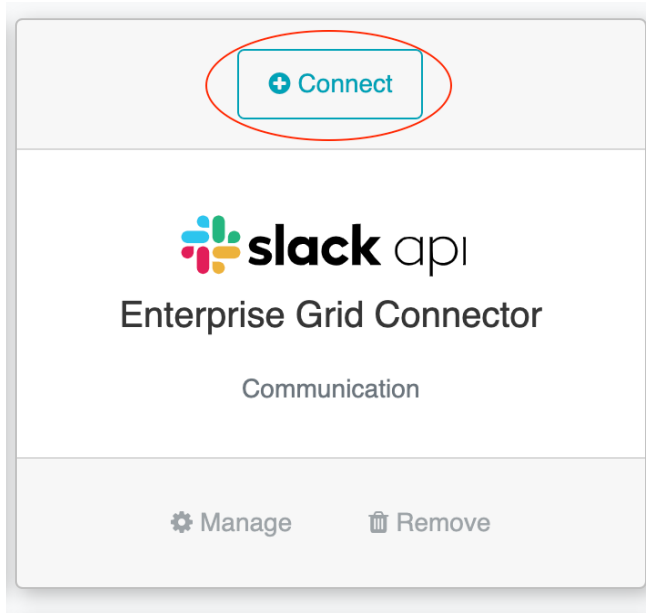
(Image shows Slack Business+. Please note that Slack standard will look slightly different)

3. Navigate back to the connector hub and enter the credentials for the Workspace Owner service account to connect.

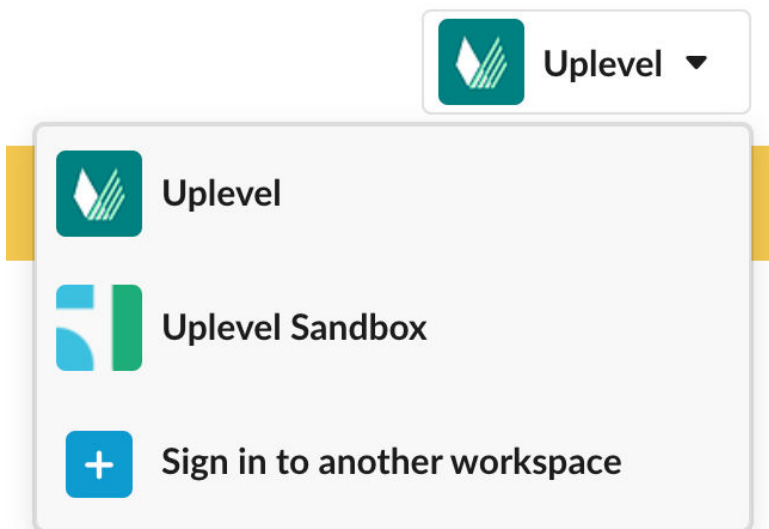
Setting Up Slack Enterprise Grid Connector

0. Please make sure you have [Discovery APIs](#) enabled for your Slack Enterprise Grid Plan. To enable, submit a request through your Slack Account Representative.

1. Click on the **Connect** button for the Slack Enterprise Grid Connector. This will kick off the Oauth flow, requesting the `discovery.read` scope permission.



2. On the upper right corner, select your Enterprise Grid workspace (**Important:** the workspace must be reselected regardless if it is already selected) Sign in if prompted with your Slack Workspace Owner credentials.





3. Click on the **Allow** button to grant your permission. This will redirect you back to the Uplevel Connector Hub.

Uplevel Connector is requesting permission to access the Uplevel Sandbox Slack organization



What will Uplevel Connector be able to view?

 Content and info about you 

What will Uplevel Connector be able to do?

 Administer Slack for your organization 

4. Click on the **Continue**.

- Note, an `xoxp-*` token is automatically populated/redacted in the password field. Please do not share this token. This token will be encrypted and stored locally.

5. Upload a csv file containing the list of email addresses of participants. This list should be formatted as one email address per line, without any quotes.

- Below is an example content of a valid emails.csv file:

```
dave@uplevelteam.com
ejiro@uplevelteam.com
stef@uplevelteam.com
ravs@uplevelteam.com
```

6. Click on Continue, then click on the Create An Archive button

7. Close browser, and exit SSH shell

Setting Up Microsoft Teams Connector

To connect the Microsoft Teams Connector, a Admin in Microsoft Azure will need to:

- Register an Enterprise application.
- Create a service account with a user role.
- Request access to protected API by [completing this form](#). (Please refer to the **Request Access to Protected API** Section below)

As an Admin, please follow the below steps to register an azure app for the Microsoft Teams Connector. This will ensure that a service account with a user role can be used to authorize read-only access to participants' calendar events

1. Create a text file named **azure.env**, copy the below variables and paste into the file. This file will be uploaded into the Office 365 Connector.

```
MSAL_CLIENT_ID=  
MSAL_TENANT_ID=  
MSAL_CLIENT_SECRET=
```

2. Navigate to: <https://portal.azure.com> and select Azure AD > App Registrations

- Enter Name: Uplevel Teams Connector
- Enter Redirect URI (Web):
`http://localhost:8080/service/microsoft/teams/connector/callback`
- Check (x) Multi-tenant as a supported account type
- Click [Register] button

Register an application



* Name

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

Uplevel Teams Connector

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (UpLevel, Inc only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[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

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

3. In the overview menu:

- Copy Application (client) ID and paste into azure.env file as **MSAL_CLIENT_ID=value**
- Copy Directory (tenant) ID and paste into azure.env file as **MSAL_TENANT_ID=value**

Uplevel Connector



Search (Cmd+/)

Delete Endpoints

- Overview
- Quickstart
- Integration assistant (preview)
- Manage
 - Branding
 - Authentication
 - Certificates & secrets
 - Token configuration
 - API permissions
 - Expose an API
 - Owners
 - Roles and administrators (Preview)
 - Manifest
- Support + Troubleshooting
 - Troubleshooting
 - New support request

Display name Uplevel Connector	Supported account types Multiple organizations
Application (client) ID 89064610-4e18-40b6-b88c-97a8075c3f55	Redirect URIs 1 web, 0 spa, 0 public client
Directory (tenant) ID 97a8075c-3f55-4c5e-8000-000000000000	Application ID URI Add an Application ID URI
Object ID 553fb4a8-5016-421f-b40f-b8adad8fca6	Managed application in local directory Uplevel Connector

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

Call APIs



Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

[View API permissions](#)

Documentation

- Microsoft identity platform
- Authentication scenarios
- Authentication libraries
- Code samples
- Microsoft Graph
- Glossary
- Help and Support

4. Authentication menu:

- Check [x] Access tokens
- Confirm that Multitenant is checked
- Click [Save] button

Home > UpLevel, Inc > Uplevel Teams Connector

Uplevel Teams Connector | Authentication

Search (Cmd+/) << Save Discard Got feedback?

- Overview
- Quickstart
- Integration assistant

Manage

- Branding
- Authentication**
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators | Preview
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Platform configurations

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

+ Add a platform

Web

Quickstart Docs Discard

Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating or signing out users. Also referred to as reply URLs. [Learn more about Redirect URIs and their restrictions](#)

http://localhost:8080/service/microsoft/teams/connector/callback

Add URI

Front-channel logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

e.g. https://example.com/logout

Implicit grant and hybrid flows

Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. [Learn more about tokens.](#)

Select the tokens you would like to be issued by the authorization endpoint:

- Access tokens (used for implicit flows)
- ID tokens (used for implicit and hybrid flows)

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (UpLevel, Inc only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)

[Help me decide...](#)

5. Certificates & secrets menu:

- Click [New client secret] button

- Enter Description: Uplevel
- Check (x) Never
- Click on [Add] button
- Copy Client secret value and paste into azure.env file as:
MSAL_CLIENT_SECRET=value

Home > UpLevel, Inc | App registrations >

Uplevel Connector | Certificates & secrets ✕

Search (Cmd+/)

- Overview
- Quickstart
- Integration assistant (preview)

Manage

- Branding
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- Owners
- Roles and administrators (Preview)
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Add a client secret

Description

Expires

In 1 year
 In 2 years
 Never

Add Cancel

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value
-------------	---------	-------

No client secrets have been created for this application.

Home > UpLevel, Inc | App registrations >

Uplevel Connector | Certificates & secrets ✕

Search (Cmd+/)

- Overview
- Quickstart
- Integration assistant (preview)

Manage

- Branding
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- Owners
- Roles and administrators (Preview)
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

i Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this blade.

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

↑ Upload certificate

Thumbprint	Start date	Expires
------------	------------	---------

No certificates have been added for this application.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value
Password uploaded on Fri Jul 31 2020	12/31/2299	w80Z5Rbgm_H-jAQ[REDACTED]

6a. In the API permissions menu:

- Click on **[Add a permission]** and then select **Microsoft Graph**
- Select the **Application permissions** and search for **Calendars**
- Search for and select all these **8 read-only permissions**:
 - **User.Read.All** #Allows connecting and reading other users profile
 - **GroupMember.Read.All** #Allows the app to list groups, read basic group properties and read membership of all your groups.
 - **Team.ReadBasic.All** #Read the names and descriptions of teams, on your behalf.
 - **Channel.ReadBasic.All** #Read channel names and channel descriptions, on your behalf.
 - **TeamMember.Read.All** #Read the members of teams, on your behalf.
 - **ChannelMember.Read.All** #Read the members of channels, on your behalf.
 - **ChannelMessage.Read.All** #allows reading channel messages.
 - **Chat.Read.All** #Allows an app to read your 1 on 1 or group chat messages in Microsoft Teams, on your behalf.
- Click on **[Add permissions]** button

Home > UpLevel, Inc > Uple

Request API permissions

Uplevel Team

Search (Cmd+/)

- Overview
- Quickstart
- Integration assistant

Manage

- Branding
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators | F
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Microsoft Graph
https://graph.microsoft.com/ Docs

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select permissions **#1) search here for all 8 read-only permissions** expand all

Permission	Admin consent required
> IdentityRiskyUser	
∨ User (1)	
<input checked="" type="checkbox"/> User.Read.All ⓘ Read all users' full profiles	Yes

#2) select

6b. In the API Permissions menu:

- Click on **[Grant admin consent for {Company Name}]** button
 - Please ensure that all **8 read-only permissions** have been Granted by the Admin for your organization (they should all have **green** check marks).

- Search (Cmd+/)
- Refresh | Got feedback?
- Overview
- Quickstart
- Integration assistant
- Manage
 - Branding
 - Authentication
 - Certificates & secrets
 - Token configuration
 - API permissions**
 - Expose an API
 - App roles
 - Owners
 - Roles and administrators | Preview
 - Manifest
- Support + Troubleshooting
 - Troubleshooting
 - New support request

Successfully granted admin consent for the requested permissions.

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 UpLevel, Inc

API / Permissions name	Type	Description	Admin consent req...	Status
▼ Microsoft Graph (9)				
Channel.ReadBasic.All	Application	Read the names and descriptions of all channels	Yes	Granted for UpL...
ChannelMember.Read.All	Application	Read the members of all channels	Yes	Granted for UpLev
ChannelMessage.Read.All	Application	Read all channel messages	Yes	Granted for UpLev
Chat.Read.All	Application	Read all chat messages	Yes	Granted for UpLev
GroupMember.Read.All	Application	Read all group memberships	Yes	Granted for UpLev
Team.ReadBasic.All	Application	Get a list of all teams	Yes	Granted for UpL...
TeamMember.Read.All	Application	Read the members of all teams	Yes	Granted for UpL...
User.Read	Delegated	Sign in and read user profile	No	Granted for UpL...
User.Read.All	Application	Read all users' full profiles	Yes	Granted for UpL...

Optional

7. In the Branding menu:

- Download this icon and Upload as new logo



8. Navigate to: <https://admin.microsoft.com/> Users -> Active users

- Click on Add a user
- Setup basic user information with **User (no admin center access)** role

Add a user ×

Basics

Product licenses

Optional settings

Finish

Optional settings

You can choose what role you'd like to assign for this user, and fill in additional profile information.

Roles (User: no administration access) ^

Admin roles give users permission to view data and complete tasks in admin centers. Give users only the access they need by assigning the least-permissive role.

[Learn more about admin roles](#)

User (no admin center access)

Admin center access

Global readers have read-only access to admin centers, while Global admins have unlimited access to edit all settings. Users assigned other roles are more limited in what they can see and do.

- Exchange admin ⓘ
- Global admin ⓘ
- Global reader ⓘ
- Helpdesk admin ⓘ
- Service support admin ⓘ
- SharePoint admin ⓘ
- Teams service admin ⓘ
- User admin ⓘ

Show all by category v

Back Next

9. Microsoft requires that additional validation is needed to access some protected Teams API. To request access to these API, this [form must be completed](#):

Below are details to assist in completing the form:

- #1) Enter an Admin email eg: Ejiro@uplevelteam.onmicrosoft.com
- #2) Yes
- #3) Publish name: **Uplevel Inc**

- #4) App name: **Uplevel Teams Connector**
- #5) App id(s): Provide **Client ID** from above (see Application (client) ID from azure)
- #6) Enter the following: The Uplevel App is the engineering effectiveness platform that leverages machine learning & organizational science to champion behavior change. It empowers engineers to do their best work.
- #7) Enter the following: The Uplevel App needs read access to all messages because it analyzes the sentiment of the message, the length of the messages, the time the messages were sent, and the participant that sent the messages.
- #8) Data retention: Select [*] It is obvious to any admin installing this app that it will make a cop of Microsoft Teams messages
- #9) Tenant ID: Provide the **Tenant ID** from above (see Directory (tenant) ID from azure)
- #10) Does your organization own all those tenants? Select [*] Yes
- #11) Homepage URL: <https://uplevelteam.com>
- #12) Terms of service URL: <https://uplevelteam.com/privacy-policy/>
- #13) Privacy statement URL: <https://uplevelteam.com/privacy-policy/>

10. Navigate back to the connector hub and enter the credentials for the service account to connect.

- Upload a csv file containing the list of email addresses of participants. This list should be formatted as one email address per line, without any quotes.
- Below is an example content of a valid emails.csv file:

```
dave@uplevelteam.com
ejjro@uplevelteam.com
stef@uplevelteam.com
ravs@uplevelteam.com
```

11. Click on Continue, then click on the Create An Archive button.

12. Close browser, and exit SSH shell.

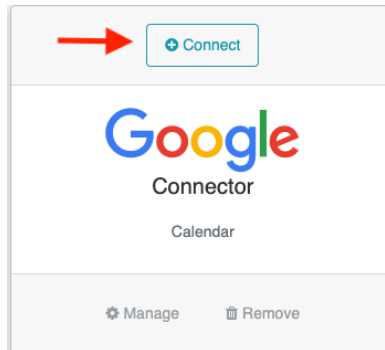
Calendar

To set up the calendar connector, we recommend setting up a service account. The calendar data points that will be passed to Uplevel for processing are:

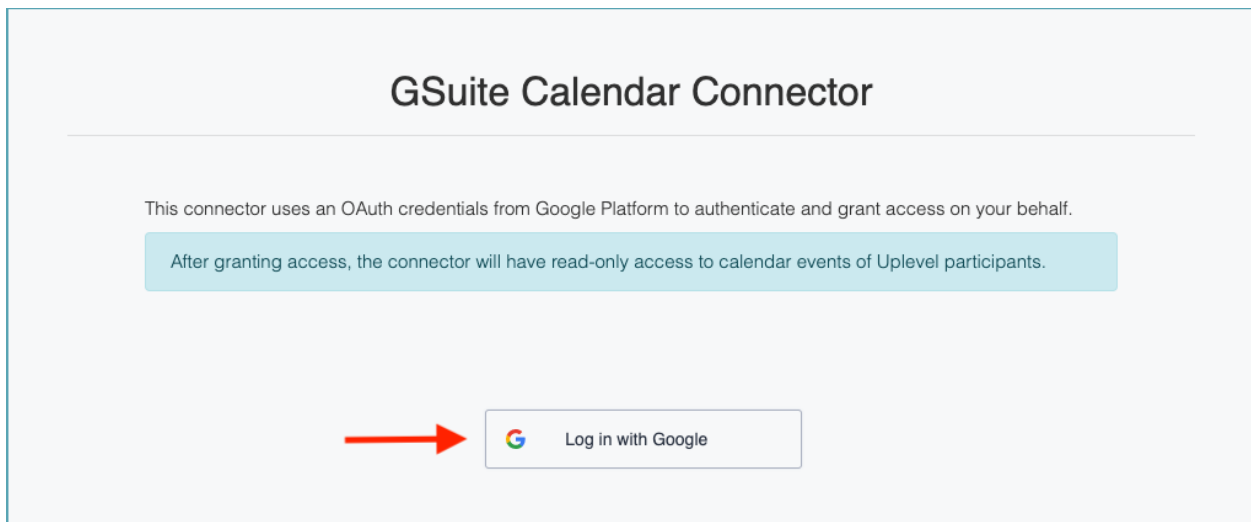
- Only calendars from selected users
- Meeting time, acceptance, invitees, organizational level, recurrence, titles (optional)
- **We always exclude:** Attachments, body, private events

Setting Up Google Calendar Connector

1. Set up a calendar admin service account for Uplevel.
2. Navigate to the Google Calendar connector tile and click “Connect”.



3. Click “Login with Google” and enter the service account credentials.



4. Allow calendar access and you'll be directed back to the Connector Hub. [Continued on next page]

This will allow **UpLevel Connector** to:

31 View your calendars



Make sure you trust UpLevel Connector

You may be sharing sensitive info with this site or app. Learn about how UpLevel Connector will handle your data by reviewing its [privacy policies](#). You can always see or remove access in your [Google Account](#).

[Learn about the risks](#)

Cancel

Allow

4. Upload a csv file containing the list of email addresses of participants. This list should be formatted as one email address per line, without any quotes.

- Below is an example content of a valid emails.csv file:

```
dave@uplevelteam.com
ejiro@uplevelteam.com
stef@uplevelteam.com
ravs@uplevelteam.com
```

5. Click on Continue, then click on the Create An Archive button.

6. Close browser, and exit SSH shell. [Continued on next page]

CREATE NEW ARCHIVE.

Please specify preferences that are required.
To continue, a file containing the email addresses of participants is required.

Get Email Addresses

1. Create an **emails.csv** file containing email addresses of participants. *(list should be formatted as one email address per line, without any quotes).*

```
dave@uplevelteam.com
ejiro@uplevelteam.com
stef@uplevelteam.com
```

2. Download [sample-emails.csv](#) to see format.
3. Upload completed **email.csv** below.

Upload Email CSV File

No file chosen

Use previously uploaded email list?

From Time: 365 Days Behind

To Time: 0 Days Ahead

The connector only has read-only access to calendar events of participants.

Setting up Google Calendar Connector - OAuth

To connect the Google Calendar Connector, an admin account will need to:

- Create an **OAuth2 Credentials** in Google Cloud Platform (GCP).
- Download the OAuth2 JSON credential file

Create an OAuth2 Credentials in GCP

1. Go to Google Cloud Platform Console and create a new project:
 - <https://console.cloud.google.com>
2. Click on **Enable APIs & services** from the left menu of the project
 - Click **+ Enable APIs and Services** button
 - Search for "calendar" and select the **Google Calendar API**
 - Click **Enable**



Google Calendar API

Google Enterprise API

Integrate with Google Calendar using the Calendar API.

ENABLE TRY THIS API

- 2. Register an app by clicking on the **OAuth consent screen** from the menu
 - In the User Type, select **Internal** and click **Create**

Google Cloud Platform Connector Project Search Products, resources, docs (/)

API APIs & Services	OAuth consent screen
<ul style="list-style-type: none"> Enabled APIs & services Library Credentials OAuth consent screen Domain verification Page usage agreements 	<p>Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.</p> <p>User Type</p> <p><input checked="" type="radio"/> Internal ?</p> <p>Only available to users within your organization. You will not need to submit your app for verification. Learn more about user type</p> <p><input type="radio"/> External ?</p> <p>Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. Learn more about user type</p> <p>CREATE</p>

- Enter require fields for App name, User support email and Contact email addresses
- Click **Save and Continue** button

- Click **Add or Remove Scope** button
 - Search for calendar.readonly
 - Select the shown URL

Connector Project Search Products

Edit app registration

OAuth consent screen — 2 Scopes — 3

Scopes express the permissions you request users app and allow your project to access specific types from their Google Account. [Learn more](#)

ADD OR REMOVE SCOPES

Update selected scopes

Only scopes for enabled APIs are listed below. To add the API in the [Google API Library](#) or use the Pasted Sc new APIs you enable from the Library.

Filter calendar.readonly

<input type="checkbox"/>	API Properties	
<input type="checkbox"/>	https://www.googleapis.com/auth/calendar.readonly	UR
<input type="checkbox"/>	.../auth/userinfo.profile	See your made pu

- Select the **checkbox** for Google Calendar API
- Click **Update**

Connector Project Search Products

Edit app registration

OAuth consent screen — 2 Scopes — 3

Scopes express the permissions you request users app and allow your project to access specific types from their Google Account. [Learn more](#)

ADD OR REMOVE SCOPES

Update selected scopes

Only scopes for enabled APIs are listed below. To add a missing API in the [Google API Library](#) or use the Pasted Scopes text box APIs you enable from the Library.

Filter https://www.googleapis.com/auth/calendar.readonly

<input checked="" type="checkbox"/>	API ↑	Scope	User-facing des
<input checked="" type="checkbox"/>	Google Calendar API	.../auth/calendar .readonly	See and down your Google C

- Click **Save and Continue**
- Confirm Scopes is properly configured for **calendar.readonly** the summary section

Scopes

[EDIT](#)

API ↑	Scope	User-facing description
Google Calendar	.../auth/calendar	See and download any calendar you can access using your
API	.readonly	Google Calendar

[BACK TO DASHBOARD](#)

3. Click on the **Credential** from the menu

- Click + **Create Credentials** >> **OAuth client ID**
- Select the **Web application** application type from the dropdown
- Enter name: Uplevel Connector Hub
- Enter Redirect URL: `http://localhost:8080/service/google/calendar/callback`
- Click **Save** and **Download JSON**

The screenshot shows the Google Cloud Platform interface for the 'Connector Project'. The left sidebar shows 'APIs & Services' with 'Credentials' selected. The main content area shows 'Credentials' with a '+ CREATE CREDENTIALS' button and a 'DELETE' button. Below this is a 'Create credentials to access APIs' dropdown menu. The menu options are: 'API key' (Identifies your project using a simple API key to check quota and access), 'OAuth client ID' (Requests user consent so your app can access the user's data), and 'Service account' (Enables server-to-server, app-level authentication using robot account). The 'OAuth client ID' option is highlighted with a red box.

- API APIs & Services
- Enabled APIs & services
- Library
- Credentials**
- OAuth consent screen
- Domain verification
- Page usage agreements

Create OAuth client ID

Application type *
Web application

Name *
Uplevel Connector Hub

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

Authorized JavaScript origins ?

For use with requests from a browser

+ ADD URI

Authorized redirect URIs ?

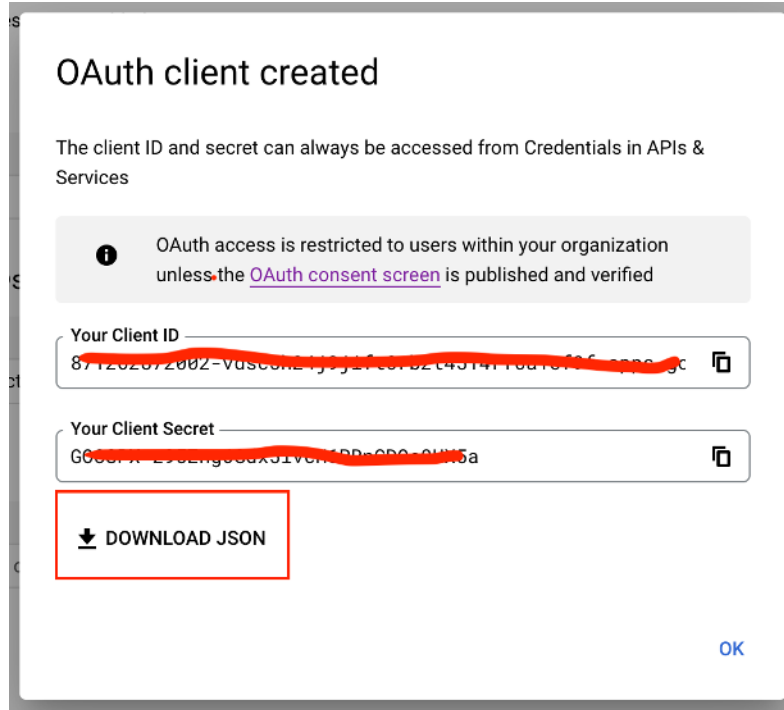
For use with requests from a web server

URIs 1 *
http://localhost:8080/service/google/calendar/callback

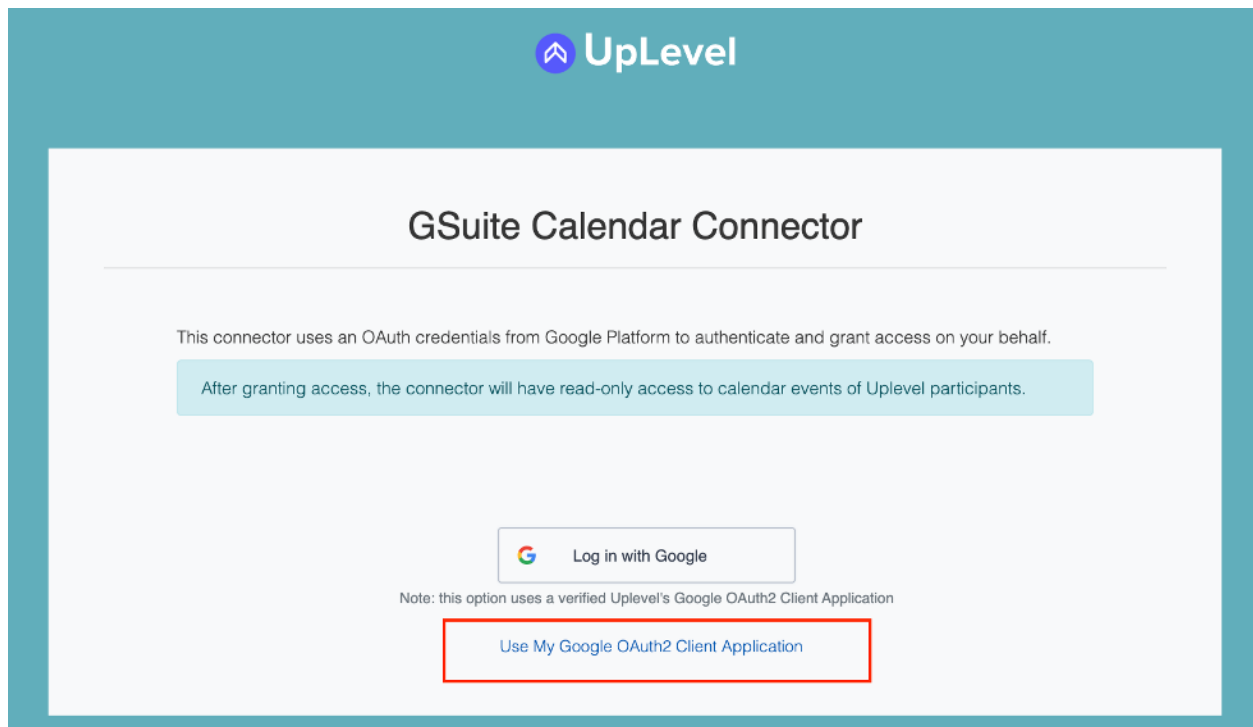
+ ADD URI

Note: It may take 5 minutes to a few hours for settings to take effect

CREATE CANCEL



4. Click "Connect" on the Google calendar connector and upload the downloaded JSON file into the Google Connector
 - Follow the OAuth authentication flow



GSuite Calendar Connector

This connector uses Google OAuth 2.0 credentials from the Google API Console to authenticate.

Steps To Connect:

- Please ask an uplevel rep for steps to create a **Google OAuth 2.0 Client App** and how to obtain a **credentials JSON** file.
- You will then need to upload the **credentials JSON** file below.

*Alternatively: For a more simplified approach to connect, please click on the **Uplevel's Google OAuth2 Client App** link below. However, it does require a **service account** with admin permission.*

Connect Google Calendar

Upload Credentials JSON File

No file chosen

Continue

Uplevel's Google OAuth2 Client App

Setting Up O365 Connector

To connect the Office 365 Calendar Connector, it requires a Microsoft Azure Admin to register an Enterprise application and create a service account with a “User” role. As an Admin, please follow the below steps to register an azure app for the O365 Connector, this would ensure that a service account with a user role can be used to authorize access to participants calendar events. Note: for information on limiting the scope of the application, see the [documentation here](#).

1. Create a text file named **azure.env**, copy the below variables and paste into the file. This file will be uploaded into the Office 365 Connector.

```
MSAL_CLIENT_ID=  
MSAL_TENANT_ID=
```

MSAL_CLIENT_SECRET=

2. Navigate to: <https://portal.azure.com> Azure AD -> App Registrations

- Enter Name: Uplevel Connector
- Enter Redirect URI (Web):
 - <http://localhost:8080/service/microsoft/calendar/callback>
- Check (x) Multi-tenant as a supported account type
- Click [Register] button

[Home](#) > [UpLevel, Inc | App registrations](#) >

Register an application ×

* Name

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

Uplevel Connector ✓

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (UpLevel, Inc only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

[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 ✓
<http://localhost:8080/service/microsoft/calendar/callback> ✓

By proceeding, you agree to the [Microsoft Platform Policies](#) 🔗

Register

3. In the Overview menu:

- Copy Application (client) ID and paste into azure.env file as **MSAL_CLIENT_ID=value**
- Copy Directory (tenant) ID and paste into azure.env file as **MSAL_TENANT_ID=value**

Uplevel Connector

Search (Cmd+)

Delete Endpoints

- Overview
- Quickstart
- Integration assistant (preview)
- Manage
 - Branding
 - Authentication
 - Certificates & secrets
 - Token configuration
 - API permissions
 - Expose an API
 - Owners
 - Roles and administrators (Preview)
 - Manifest
- Support + Troubleshooting
 - Troubleshooting
 - New support request

Display name	Uplevel Connector	Supported account types	Multiple organizations
Application (client) ID	89064610-4e18-40b6-8000-000000000000	Redirect URIs	1 web, 0 spa, 0 public client
Directory (tenant) ID	97a8075c-3f55-4c5e-8000-000000000000	Application ID URI	Add an Application ID URI
Object ID	553fb4a8-5016-421f-b40f-b8adad8fcba6	Managed application in local directory	Uplevel Connector

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

Call APIs



Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

[View API permissions](#)

Documentation

- Microsoft identity platform
- Authentication scenarios
- Authentication libraries
- Code samples
- Microsoft Graph
- Glossary
- Help and Support

4. In the Authentication menu:

- Check [x] Access tokens
- Confirm that Multitenant is checked
- Click [Save] button

[Continued on next page]

Home > UpLevel, Inc | App registrations >

Uplevel Connector | Authentication

Search (Cmd+/) Save Discard Got feedback?

Overview
Quickstart
Integration assistant (preview)

Manage

- Branding
- Authentication**
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- Owners
- Roles and administrators (Preview)
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

+ Add a platform

Web

Quickstart Docs

Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating users. Also referred to as reply URLs. [Learn more about Redirect URIs and their restrictions](#)

http://localhost:8080/service/microsoft/calendar/callback

Add URI

Logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

e.g. https://myapp.com/logout

Implicit grant

Allows an application to request a token directly from the authorization endpoint. Checking Access tokens and ID tokens is recommended only if the application has a single-page architecture (SPA), has no back-end components, does not use the latest version of MSAL.js with auth code flow, or it invokes a web API via JavaScript. ID Token is needed for ASP.NET Core Web Apps. [Learn more about the implicit grant flow](#)

To enable the implicit grant flow, select the tokens you would like to be issued by the authorization endpoint:

Access tokens
 ID tokens

Supported account types

Who can use this application or access this API?

Accounts in this organizational directory only (UpLevel, Inc only - Single tenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)

[Help me decide...](#)

5. In the Certificates & secrets menu:

- Click [New client secret] button
- Enter Description: Uplevel
- Check (x) Never
- Click on [Add] button
- Copy Client secret value and paste into azure.env file as: **MSAL_CLIENT_SECRET=value**

Search (Cmd+/)

- Overview
- Quickstart
- Integration assistant (preview)

Manage

- Branding
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- Owners
- Roles and administrators (Preview)
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Add a client secret

Description

Expires

In 1 year

In 2 years

Never

Add Cancel

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[+ New client secret](#)

Description	Expires	Value
No client secrets have been created for this application.		

Search (Cmd+/)

- Overview
- Quickstart
- Integration assistant (preview)

Manage

- Branding
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- Owners
- Roles and administrators (Preview)
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this blade.

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

[↑ Upload certificate](#)

Thumbprint	Start date	Expires
No certificates have been added for this application.		

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[+ New client secret](#)

Description	Expires	Value
Password uploaded on Fri Jul 31 2020	12/31/2299	w80Z5Rbgm_H-jAQ: [REDACTED]

5a. In the API permissions menu:

- Click on [Add a permission] and then select Microsoft Graph
- Select the **Application permissions** and search for Calendars
- Check [x] Calendars.Read
- Click on [Add permissions] button

Home > UpLevel, Inc | App registrations >

Uplevel Connector | API

Search (Cmd+/) << < Refresh >

- Overview
- Quickstart
- Integration assistant (preview)

Manage

- Branding
- Authentication
- Certificates & secrets
- Token configuration
- API permissions**
- Expose an API
- Owners
- Roles and administrators (Preview)
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Request API permissions

< All APIs

Microsoft Graph
https://graph.microsoft.com/ Docs

What type of permissions does your application require?

- Delegated permissions
Your application needs to access the API as the signed-in user.
- Application permissions**
Your application runs as a background service or daemon without a signed-in user.

Select permissions

Type to search

Permission	Admin consent required
> AccessReview	
> AdministrativeUnit	
> Application	
> AppRoleAssignment	
> ApprovalRequest	
> AuditLog	
> BitlockerKey	
> Calendars (1)	
<input checked="" type="checkbox"/> Calendars.Read Read calendars in all mailboxes	Yes
<input type="checkbox"/> Calendars.ReadWrite Read and write calendars in all mailboxes	Yes
> CallRecords	
> Calls	
> Channel	
> ChannelMember	
> ChannelMessage	
> ChannelSettings	
> Chat	
> Contacts	

Add permissions Discard

5b. In the API Permissions menu:

- Click on [Grant admin consent for {Company Name}] button

UpLevel Connector | API permissions ✕

Search (Cmd+/) <<

Refresh

Successfully granted admin consent for the requested permissions.

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 UpLevel, Inc

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (2)				
Calendars.Read	Application	Read calendars in all mailboxes	Yes	Granted
User.Read	Delegated	Sign in and read user profile	-	Granted

Manage

- Branding
 - Authentication
 - Certificates & secrets
 - Token configuration
 - API permissions**
 - Expose an API
 - Owners
 - Roles and administrators (Preview)
 - Manifest
- ### Support + Troubleshooting
- Troubleshooting
 - New support request

5c. In the Delegated Permissions menu, select `offline access`

Request API permissions



[← All APIs](#)



Microsoft Graph

<https://graph.microsoft.com/> [Docs](#) [↗](#)

What type of permissions does your application require?

Delegated permissions

Your application needs to access the API as the signed-in user.

Application permissions

Your application runs as a background service or daemon without a signed-in user.

Select permissions

[expand all](#)

Permission	Admin consent required
▼ OpenId permissions (1)	
<input checked="" type="checkbox"/> offline_access ⓘ Maintain access to data you have given it access to	No

Add permissions

Discard

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 Uplevel

API / Permissions name	Type	Description	Admin consent requ...	Status
▼ Microsoft Graph (2)				
Calendars.Read	Application	Read calendars in all mailboxes	Yes	✓ Granted for Uplevel
offline_access	Delegated	Maintain access to data you have given it access to	No	

Optional

6. In the Branding menu:

- Download this icon and Upload as new logo



7. Go to: <https://admin.microsoft.com/> and click Users -> Active users

- Click on Add a user
- Setup basic user information with user (no admin center access) role

Add a user ×

- ✖ Basics
- ✖ Product licenses
- **Optional settings**
- Finish

Optional settings

You can choose what role you'd like to assign for this user, and fill in additional profile information.

Roles (User: no administration access) ^

Admin roles give users permission to view data and complete tasks in admin centers. Give users only the access they need by assigning the least-permissive role.

[Learn more about admin roles](#)

User (no admin center access)

Admin center access

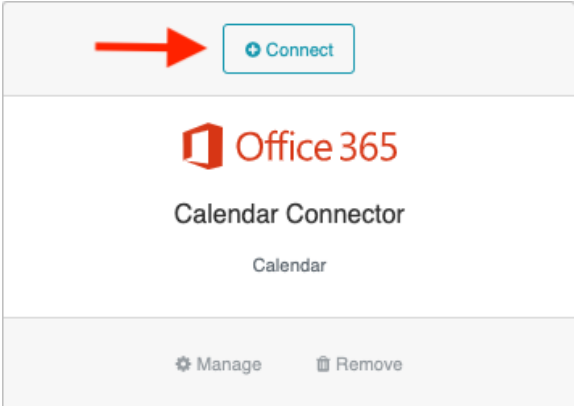
Global readers have read-only access to admin centers, while Global admins have unlimited access to edit all settings. Users assigned other roles are more limited in what they can see and do.

- Exchange admin ⓘ
- Global admin ⓘ
- Global reader ⓘ
- Helpdesk admin ⓘ
- Service support admin ⓘ
- SharePoint admin ⓘ
- Teams service admin ⓘ
- User admin ⓘ

Show all by category v

Back Next

8. Once the service account has been set up, navigate to the Connector Hub and click “Connect” on the Office 365 Connector.



- Click “Login with Microsoft Account” and upload the azure.env file.

9. Once the above steps are done, the final step is to upload a CSV file with all email addresses in your engineering org. This tells the connector which calendars to pull data from. Once uploaded, click “continue” then “create an archive” and the archive process will begin.

- Below is an example content of a valid emails.csv file:

```
dave@uplevelteam.com
ejiro@uplevelteam.com
stef@uplevelteam.com
ravs@uplevelteam.com
```

10. Click on Continue, then click on the Create An Archive button.

11. Close browser, and exit SSH shell.

Work Management Tools

This section outlines how to connect your Work Management tool to Uplevel. We recommend setting up a service account. The Work Management data points that will be passed to Uplevel for processing are:

- Issues & changelog within all specified Projects including all users contributing to those projects, with the option to redact the description.
 - We can include / exclude by project as needed.

Setting Up Jira Cloud Connector

Click “Connect” on the Jira Cloud Connector. Follow the instructions listed in the Connector Hub.

Note: you will need an understanding of the Jira projects you’d like to include in the Uplevel analysis.

1. Create a service account.

- Login to Jira Admin portal <https://admin.atlassian.com>.
- Click on Manage users and create a Service Account.
- Role should be a Basic user role.
- Access required: “Has access on site and Jira Software” should be checked.

2. Create an token

- Login using Service Account credentials to <https://id.atlassian.com/manage/api-tokens>
- Click Create API token.
- From the dialog that appears, enter “Uplevel Connector: as a Label and click Create.
- Click Copy to clipboard.
 - Note, you will not be able to view this token again.

3. Connect

- Paste the copied Token into the API Token field in the connector hub.
- Paste your Jira Base URL below the API token.

4. Select the Jira Projects that should be included in the analysis.

5. Click Create an archive.

Setting Up Jira OnPrem Connector

Click “Connect” on the Jira Onprem Connector. Follow the instructions listed in the Connector Hub.

Note: you will need an understanding of the Jira projects you’d like to include in the Uplevel analysis.

1. Create a service account.

- Login to Jira Admin portal <https://admin.atlassian.com>.
- Click on Manage users and create a Service Account.
- Role should be a Basic user role.
- Access required: “Has access on site and Jira Software” should be checked.

2. Enter the credentials for the service account within the connector hub to connect.

3. Select the Jira Projects that should be included in the analysis.

4. Create an archive.

Source Code

This section outlines how to connect your Source Code tool to Uplevel. We recommend setting up a service account. The Source Code data points that will be passed to Uplevel for processing are:

- Pull requests, commits, pull request comments related to default branches from specified organizations by anyone who participated in them
- We always exclude: Full source code, data related to private branches

Setting Up Github Cloud/OnPrem Connector using PAT

1. Create a PAT.

- Navigate to github.com or your hosted/onprem Github Enterprise URL and login with a service account that has “member” permissions with read only access.
 - Make sure this account has access to all repos that will be included in the analysis.
- From your avatar in the top right, click on Settings.
- Click on Developer settings from the left navigation.
- Click on Personal access tokens from the left navigation.
- Click the Generate new token button.
- Enter Token description: Uplevel Connector
- Select [x]repo scopes.
 - Note, Uplevel will only make read-only request for repo metadata
- Click the Generate token button.
- Copy the revealed Token.
 - Note, you will not be able to view this token again

2. Paste the copied Token into the connector hub to connect

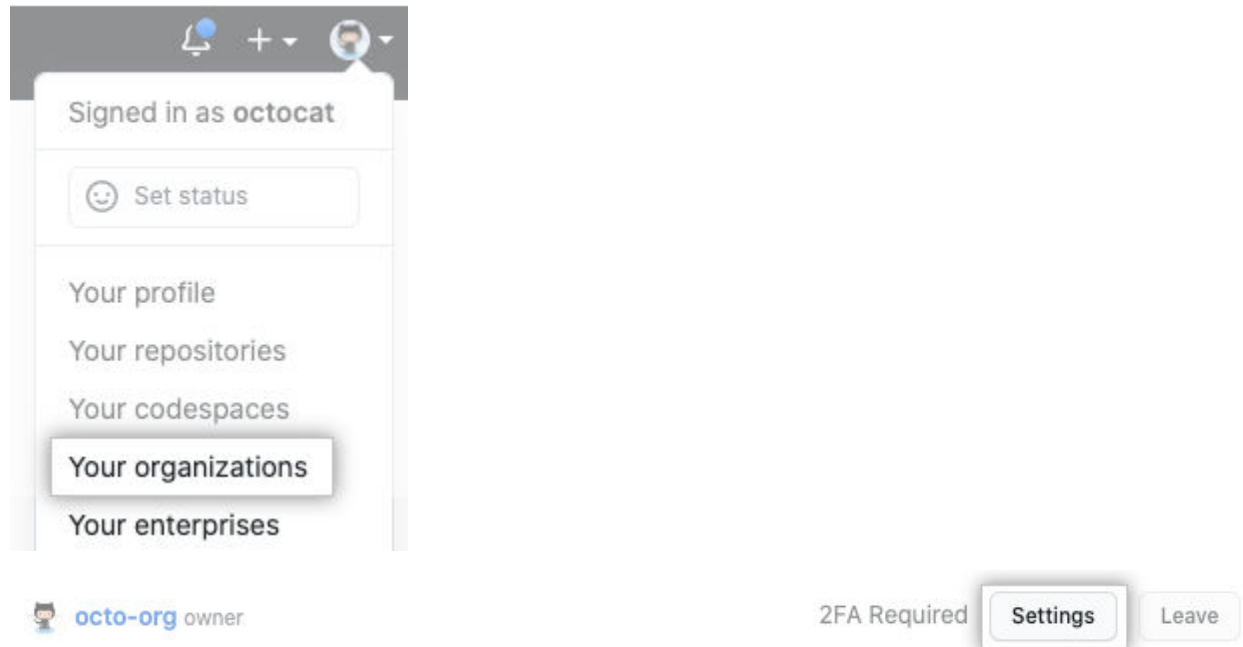
Setting Up Github Cloud/OnPrem Connector using Github Apps

Create a Github App owned by an organization

Reference link:

<https://docs.github.com/en/developers/apps/building-github-apps/creating-a-github-app>

1. Navigate to your account settings > Your organization. Then, click on the Settings button to the right of the organization.



2. Navigate to: Developer settings > Github Apps > [New Github App] button

3. Enter the following on the web form:

- **Github App name:** YourCompanyName+Uplevel Connector (eg **Octo+Uplevel Connector**)
- **Description:** Uplevel is the engineering effectiveness platform that leverages machine learning & organizational science to champion behavior change.
- **Homepage URL:** <https://uplevelteam.com>
- Deselect the [] **Active** checkbox for Webhook

Webhook

Active

We will deliver event details when this hook is triggered.

- Choose these Repository and Organization Read-only permissions (6 total)
[Continued on next page]

- **Permissions**

- + Contents [Access: Read-only]
- + Issues [Access: Read-only]
- + Metadata [Access: Read-only]
- + Pull requests [Access: Read-only]
- + Projects [Access: Read-only]
- + Members [Access: Read-only]

4. Select where the app can be installed

Where can this integration be installed?


Only on this account
Only allow this integration to be installed on the octocat account.

Any account
Allow this integration to be installed by any user or organization.

Create GitHub App

5. Click the [Create GitHub App] button. This will take you to the app settings page as shown below.

General	<h3>About</h3> <p>Owned by: @UpLevelTeam</p> <p>App ID: 20100</p> <p>Client ID:</p> <p>Revoke all user tokens</p> <p>GitHub Apps can use OAuth credentials to identify users. Learn more about identifying users by reading our integration developer documentation.</p> <p>Public link</p> <p>https://github.com/apps/uplevel-connector</p>
Permissions & events	
Install App	
App managers	
Advanced	
Optional features	

Public page 

6. Copy and take note of the App ID value - this will be entered into the connector hub.
7. Scroll down and click on the [Generate a private key] button to download a PEM file (please take note, this will be uploaded into the connector hub).

Private keys

Generate a private key

You need a private key to sign access token requests.

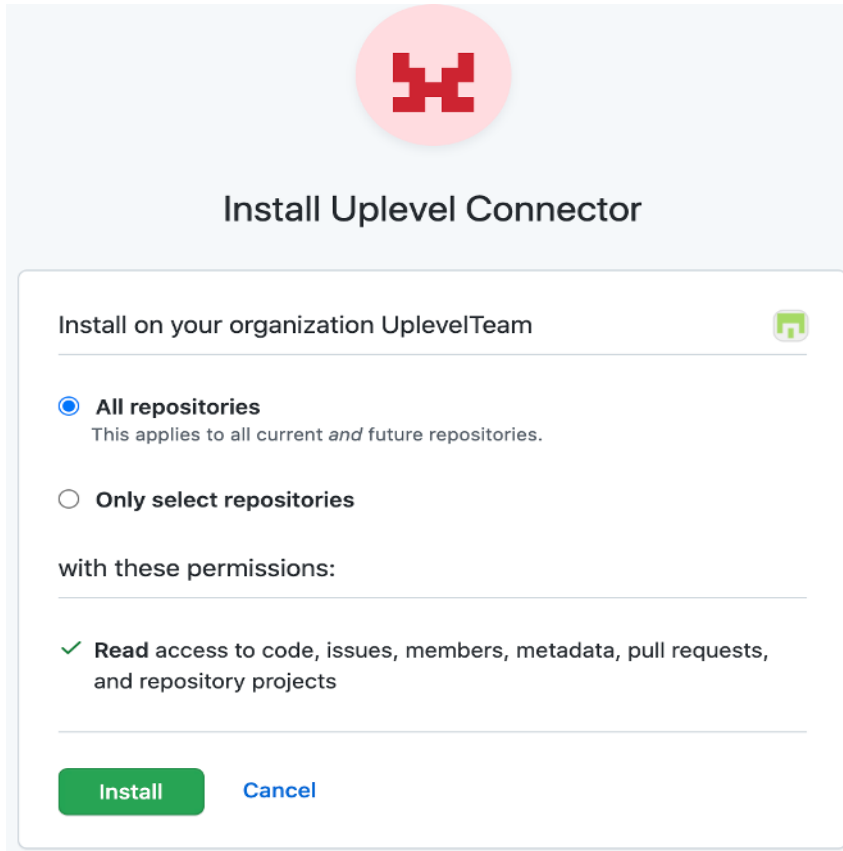
[Generate a private key](#)

8. On the same page In the left nav, click on the [Install App] menu >> Install

Reference link:

<https://docs.github.com/en/developers/apps/managing-github-apps/installing-github-apps>

- Install on All or Only selected repositories. (Repeat installation on other organizations if necessary, you just need to allow the app to be installed on any organization).



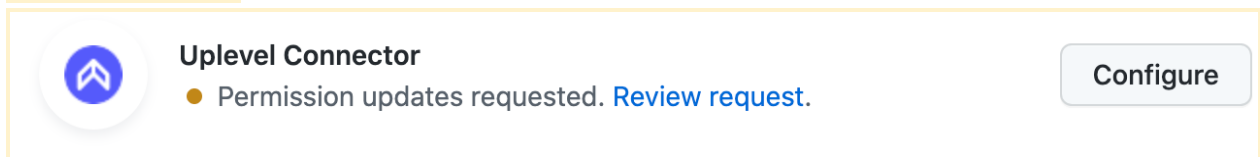
9. Once installation is complete, the page will redirect to a URL with an **Installation ID** at the end. Please copy and take note of it, it will be entered into the connector hub.

- The URL should look similar to this:

`//github.com/organizations/UpLevelTeam/settings/installations/22929903`

Note:

If you made any scope permission changes after the app has been installed, you will need to review and approve the pending request from within the list of GitHub Apps integrated into your organization.



10. You should now have all 3 items that is required to connect with the Uplevel Github connector.

- App ID
- Installation ID

- Private key PEM file

(Click the link [Use Github App to authenticate](#) at the bottom of the main Github cloud connector to start):

Github Cloud Solution - Github.com

Your Personal Access Token

Github.Com Base API

Continue

[Use Github App to authenticate](#)

Then enter the noted App ID, Installation ID, and upload the PEM file to continue

[Continued on next page]

Github Cloud Solution - Github.com

uplevel-connector.2022-01-29.private-key.pem

Use previously uploaded pem file?

Your AppId Id

Your Installation Id

Github.Com Base API

Setting Up Gitlab Cloud/OnPrem Connector

1. Create a Access Token

- Navigate to gitlab.com or your hosted/onprem Gitlab Enterprise URL and login with a service account that has “member” permissions with read only access
 - Make sure this account has access to all repos that will be included in the analysis
- From your avatar in the top right, click on Settings
- On the User Settings menu, select Access Tokens
- Enter token Name: Uplevel Connector.
- Choose or enter an Expiry date (eg 2023/01/01)
- It's best to use a few years ahead.

- Select [x]read_api and [x]read_repository scopes
 - Note, Uplevel will only make read-only request for repo metadata
- Click the Create personal access token button
- Copy and save the revealed Token
 - Note, you will not be able to view this token again

2. Paste the copied Token into the connector hub to connect.

Note: you will need an understanding of the repositories you'd like to include in the Uplevel analysis.

Setting Up Gerrit Connector

- Navigate to your hosted/onprem Gerrit Enterprise URL eg <http://localhost:7990/> and login with a service account that has “member” permissions with read only access
 - Make sure this account has access to all repos that will be included in the analysis
- From your avatar/username in the top right, click on Settings.
- On the Settings Page, select HTTP Credentials/Password.
- Take note of the Username
- Click the Generate Password button
- Copy and save the revealed password.
 - Note, you may not be able to view this password again
- Paste the noted Username into the form field below.
- Paste the copied Password and your hosted/onprem Gerrit Enterprise URL into the fields in the connector hub to connect

Setting Up Bitbucket Cloud Connector

1. Create a Key

- Navigate to <https://bitbucket.org> and login with a service account that has “member” permissions with read only access
 - Make sure this account has access to all repos that will be included in the analysis
- From your avatar in the bottom left, click on your name from the workspace menu
- Click on Settings on your personal settings page
- Click OAuth consumers from the left navigation

- Click the Add consumer button.
 - Enter *Name: Uplevel Connector
 - Enter http://localhost:8080/service/bitbucket/cloud/connector/permission/ as the Callback URL
- For Grant-Types and Permissions, select the following:
 - This is a private consumer
 - Account:Email and Account:Read
 - Workspace Membership:Read
 - Projects:Read
 - Repositories:Read
 - Pull request:Read
 - Issues:Read
 - Snippets:Read
- Click the Save button.
- Click the name of the saved OAuth consumer (eg Uplevel Connector) to reveal the Key/Secret

2. Connect

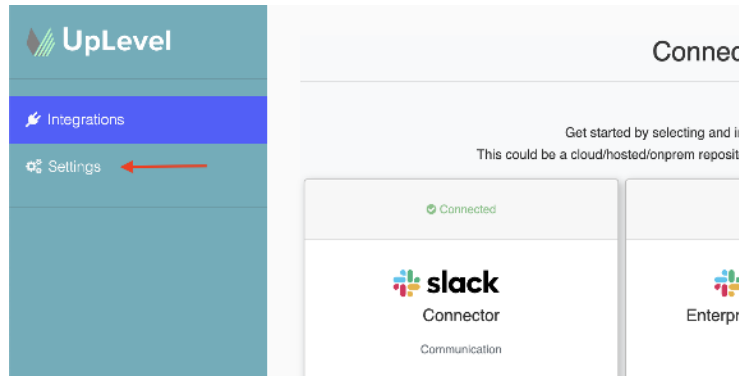
- Copy and Paste the Key/Secret into the form in the connector hub

Setting Up Bitbucket OnPrem Connector

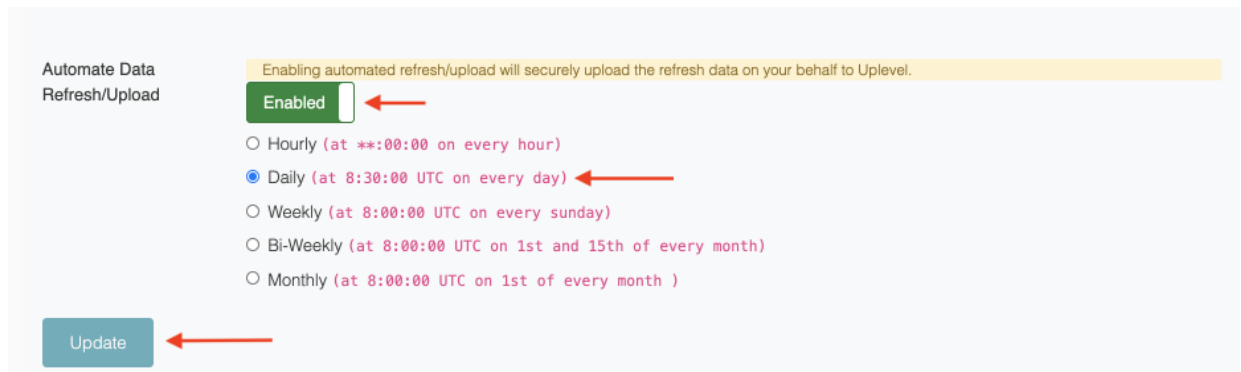
- Navigate to you self-hosting bitbucket URL eg http://localhost:7990/ and login with a service account that has “member” permissions with read only access
 - Make sure this account has access to all repos that will be included in the analysis
- From your avatar in the top right (may depend on your version), click on Manage Account menu.
- Click on HTTP/Personal Access Token from the left navigation. Click the Create a token button.
 - Enter *Name: Uplevel Connector
- Select from the dropdown options Read permissions for both Projects and Repositories.
- Click the Create button.
- Copy the revealed Token. Note, you will not be able to view this token again
- Click on the Continue button.
- Paste the copied Token into the form field in the connector hub

Final Step: Push Data to Uplevel

Once all of the data sources are connected, the final step is to push the data to Uplevel for analysis. To do this, Click “Settings” on the left navigation bar in the Connector Hub.



From there, toggle the Automate Data Refresh button to “Enabled”, keep the frequency at “Daily”, and click “Update”.



Once this is done, please contact your Uplevel Representative The Uplevel team will check the data to make sure there aren't any issues and will then begin processing. This takes about 1-2 weeks. Once that is done, we'll reach out and kick off the user launch process.

Appendix: Credential Encryption

The purpose of this section is to provide details into how credentials are stored and encrypted within the ConnectorHub architecture.

Uplevel uses an AWS-deployed [Hashicorp Vault](#) server to store a key for each of our clients. Please note that we **do not** store the actual credentials of your third-party services; we only store a key that allows the ConnectorHub to encrypt and decrypt those credentials within the client's side.

As an analogy, we keep the combination to the safe, but the actual safe (and the valuables inside it) are in your house.

We do this for two reasons:

1. By keeping an encryption key on our end, we ensure that a malicious actor who has gained access to the ConnectorHub on your end is unable to decrypt and access the credentials. In the event that such a breach occurs, it will be trivial for us to revoke the encryption key and remove the ability to decrypt the credentials.
2. By keeping the encrypted data on your end, we ensure that a malicious actor who has gained access to our Vault deployment cannot gain access to the credentials.

Essentially, this model means that a malicious actor would have to compromise two completely different systems in order to gain access to the credentials.

When the ConnectorHub gets a credential for a third-party service, it sends a request for the encryption key to our Vault server. When the key is retrieved, the ConnectorHub immediately encrypts the credential with it before writing it to disk. This ensures that every credential is encrypted at rest. Similarly, when it needs to login to begin to access the API's, the ConnectorHub sends a request for the key and decrypts the key in order to use it.

In order to get set up with our system, we need to add you to register you and add you to our Vault. Once we have done so, we will provide a username and password to allow you to access

it. Every time the ConnectorHub is started up, it will require this username and password in order to successfully encrypt and decrypt credentials.