FileCloud Server
Version 23.263
Third Party Integrations Settings

13 October, 2026

Table of Contents

Third Party Integrations 1
Enable Antivirus Scanning 2
Which solution do you want to use? 3
Use ClamAV Antivirus Scanning 3
Use ICAP Antivirus Scanning 11
Integrating FileCloud with Salesforce 16
Adding FileCloud to Salesforce 17
Configuring FileCloud with Salesforce 17
Restricting Permissions on Salesforce Team Folders 21
SIEM Integration 23
Open the SIEM settings page 23
Set up SIEM 24
Syslog Integration 26
Managing SIEM Mappings 32
SIEM Integration with Splunk Enterprise 49
reCaptcha Settings 57
To configure reCaptcha: 57
SSO API: Configure Import of SSO Groups and Users 59
To allow duplicate usernames for different domains 74
Okta: Set Up FileCloud Integration for SSO Group/User Import 75
Azure: Set up FileCloud Integration for SSO Group/User Import 85
Google: Set up FileCloud Integration for SSO Group/User Import 93
CASB integration 113
McAfee CASB integration 115
ICAP DLP 117
What is ICAP? 117
Integrating ICAP DLP with FileCloud 117
Microsoft Teams 120
For MS Teams Admins: Configuring FileCloud in Teams 120

For FileCloud Admins: Enabling Integration with MS Teams 136
Setting Up AutoCAD File Preview with Autodesk Viewer 140
Setting up integration of FileCloud and Autodesk Viewer 140
AI Integration 147
CDR Integration 152
Integrating Forcepoint CDR with FileCloud 152
Deleting file versions in quarantine 156
Removing all files from quarantine 158
eSignature Integration 161
How the eSignature process works 161
Integrating with Signority 165
Integrating with Signit 170

Third Party Integrations

Third Party Integrations settings enable you to integrate external tools such as ClamAV, ICAP and reCaptcha with FileCloud. If you are using the Advanced edition, you can set up access to FileCloud through Salesforce or include security information, CASB, and event management (SIEM) software features in FileCloud.

Enable Antivirus Scanning

- The antivirus security feature works on both Linux and Windows.
- The antivirus product may or may not be deployed on the same server as the one running the FileCloud instance.
- Antivirus scanning applies when files are uploaded to FileCloud.
- Virus scanning of a file is scheduled as soon as file upload is complete.
- Virus scanning is managed by FileCloud.

You must address virus scanning as it is a critical security feature, especially when file storage is involved.

- FileCloud allows users to upload files with arbitrary content.
- It is of utmost importance to make sure that the uploaded files are checked for malicious content in the form of viruses, trojans, malware, etc.
- FileCloud readily integrates with a variety non-commercial and commercially licensed antivirus solutions available in the market.

You can configure FileCloud to scan uploaded files in the following ways:

- Use ClamAV, an open source antivirus software that is included with FileCloud.
- Use ICAP to integrate your own choice of antivirus scanning software with FileCloud.

What is ICAP?

Internet Content Adaptation Protocol (ICAP) is a generic protocol that allows web servers to offload specialized tasks. This delegation is helpful when the tasks require custom-built servers.

Examples of such specialized tasks include:

- DLP (data loss prevention) based content scanning
- URL filtering
- antivirus scanning

Which do I use, ClamAV or ICAP?

If you have already purchased your own antivirus solution and want to use it, then choose ICAP.

If you do not want to use ClamAV for various reasons, then choose ICAP.

If you want to use antivirus scanning included with FileCloud, then choose ClamAV.

Which solution do you want to use?

> ⚠ Neither of these options provides protection for the server on which FileCloud is deployed. The antivirus solution configured here applies only for the uploaded files.

- ClamAV
- ICAP

Use ClamAV Antivirus Scanning

> ℹ FileCloud does not provide support for ClamAV or its virus signature databases, which are third-party software applications. If you need assistance with your ClamAV configuration or setup please check the ClamAV Troubleshooting FAQ.

> ⚠ ClamAV integration with Azure/S3 external networks is not supported.

You can configure FileCloud to scan uploaded files using ClamAV, an open source antivirus software.

ClamAV is available for:

- Windows
- Linux

When a virus is detected in an uploaded file, the following actions occur:

1. The incoming file is deleted.
2. An alert is displayed in the admin portal.
3. A toast is displayed in the user portal.
4. An entry is added in the audit log about virus detection in the file and subsequent deletion of the file.

To Use ClamAV

Install ClamAV in Ubuntu

These instructions are for Ubuntu Linux, but they can be used for other Linux systems using equivalent commands.

To install ClamAV in Ubuntu:

1. Install the ClamAV package

```
sudo apt-get install clamav-daemon
```

2. You might need to run 'freshclam' to update the antivirus database files

```
sudo freshclam
```

3. Update the ClamAV-Daemon mode to use TCP, by running the sudo dpkg-reconfigure clamav-base

```
sudo dpkg-reconfigure clamav-daemon
```

4. In the reconfigure wizard, choose Socket Type TCP and Interface as localhost to listen to.

5. After reconfigure finishes, verify the clamd.conf file is setup correctly (/etc/clamav/clamd.conf)

   NOTE: TCPAddr localhost may not work. You can enter the filecloud URL in place of TCPAddr to make it work

```
TCPSocket 3310
TCPAddr localhost
StreamMaxLength 100M
```

6. Additional commands for Ubuntu 16

```
#The Socket Configuration changes are also required as below:
```

```
#Edit the file /etc/systemd/system/clamav-daemon.service.d/extend.conf

[Socket]
SocketUser=clamav
ListenStream=/var/run/clamav/clamd.ctl
SocketGroup=clamav
SocketMode=666
ListenStream=xx.xx.xx.xx:3310


# Note that xx.xx.xx.xx = IP address of server or 127.0.0.1


#After that run:


systemctl --system daemon-reload
systemctl restart clamav-daemon.service
```

7. Start ClamAV-Daemon

```
sudo /etc/init.d/clamav-daemon start
```

Install ClamAV on Windows

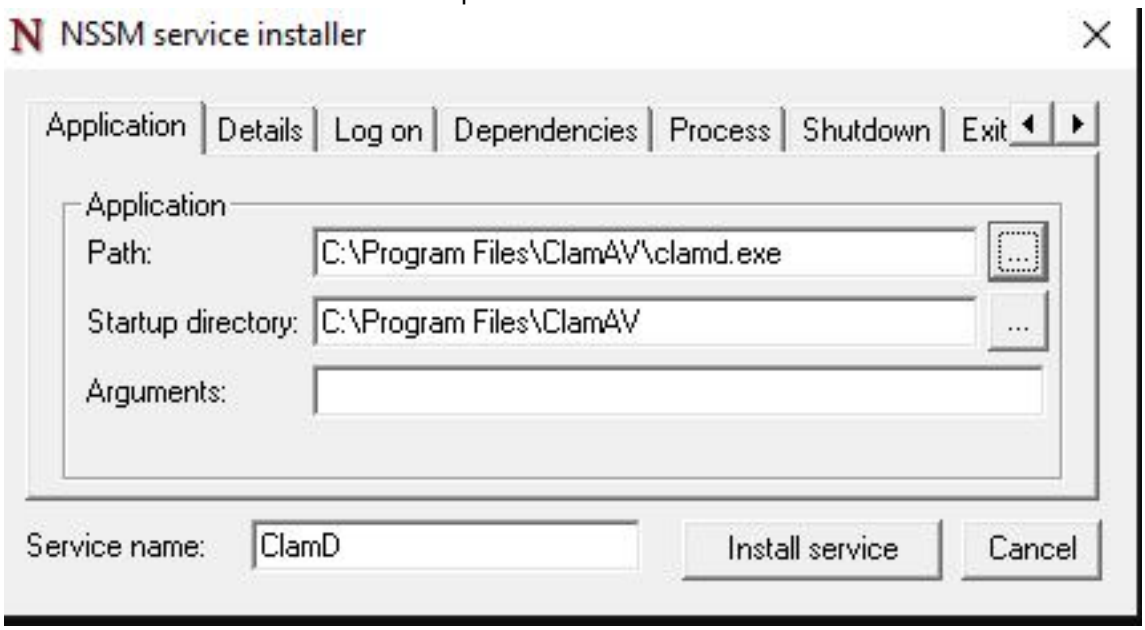
- The native ClamAV version does not have a GUI.
- The virus database definition can be updated using freshclam using a Windows task scheduler.

To install ClamAV on Windows:

1. Download the latest version of the ClamAV installer from: http://www.clamav.net/downloads
2. Install ClamAV by running the latest msi file downloaded.
3. Download the nssm Service Manager from: https://patch.codelathe.com/tonidocloud/live/3rdparty/nssm/nssm.zip
4. Unzip the nssm folder and move the nssm folder to the C:\ drive.
5. Navigate to the nssm folder in the command line and run the following command:

```
C:\nssm>nssm install ClamD
```

The nssm service install tool window opens:



6. To install the service, select the clamd.exe file path in **Application Path** and click **Install Service**.
7. Copy **clamd.conf.sample** and **freshclam.conf.sample** from **C:\Program Files\ClamAV\conf_examples** to **C:\Program Files\ClamAV**, and rename them **clamd.conf** and **freshclam.conf**
8. In **clamd.conf** and **freshclam.conf**, comment out the line beginning with *Example*.
9. If ClamAV is installed on a server other than the FileCloud server:

   Bind the IP address of the server in **clamd.conf** by changing the IP address for **TCPAddr**.
10. To update the ClamD database, enter:

```
cd C:\Program Files\ClamAV
freshclam.exe
```

The console response should appear similar to:

```
C:\Program Files\ClamAV>freshclam.exe
ClamAV update process started at Thu Mar 25 07:42:28 2021
daily database available for download (remote version: 26120)
Time:    1.1s, ETA:    0.0s [========================>]  100.57MiB/100.57MiB
Testing database: 'C:\Program Files\ClamAV\database\tmp.499b3311a3\clamav-e97910f98bef89730f7030c4c8d55340.tmp-daily.cvd' ...
Database test passed.
daily.cvd updated (version: 26119, sigs: 3965409, f-level: 63, builder: raynman)
main database available for download (remote version: 59)
Time:    1.3s, ETA:    0.0s [========================>]  112.40MiB/112.40MiB
Testing database: 'C:\Program Files\ClamAV\database\tmp.499b3311a3\clamav-6a0c33de13396c36c4b039985d2b5d2f.tmp-main.cvd' ...
Database test passed.
main.cvd updated (version: 59, sigs: 4564902, f-level: 60, builder: sigmgr)
bytecode database available for download (remote version: 333)
Time:    0.1s, ETA:    0.0s [========================>]  286.79KiB/286.79KiB
Testing database: 'C:\Program Files\ClamAV\database\tmp.499b3311a3\clamav-704d345ffd1cf864b6a0781e984101dc.tmp-bytecode.cvd' ...
Database test passed.
bytecode.cvd updated (version: 333, sigs: 92, f-level: 63, builder: awillia2)
```

11. Start the service **ClamD** from Windows Services.

12. Verify the service is running and bind it to the localhost IP address or the IP address of the ClamAV server by running the following command:

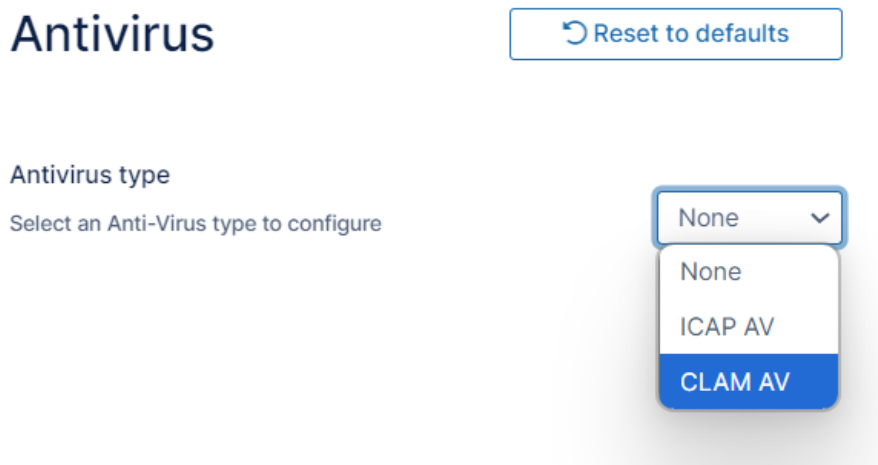
```
netstat -ano |findstr 3310
```

Integrate ClamAV with FileCloud

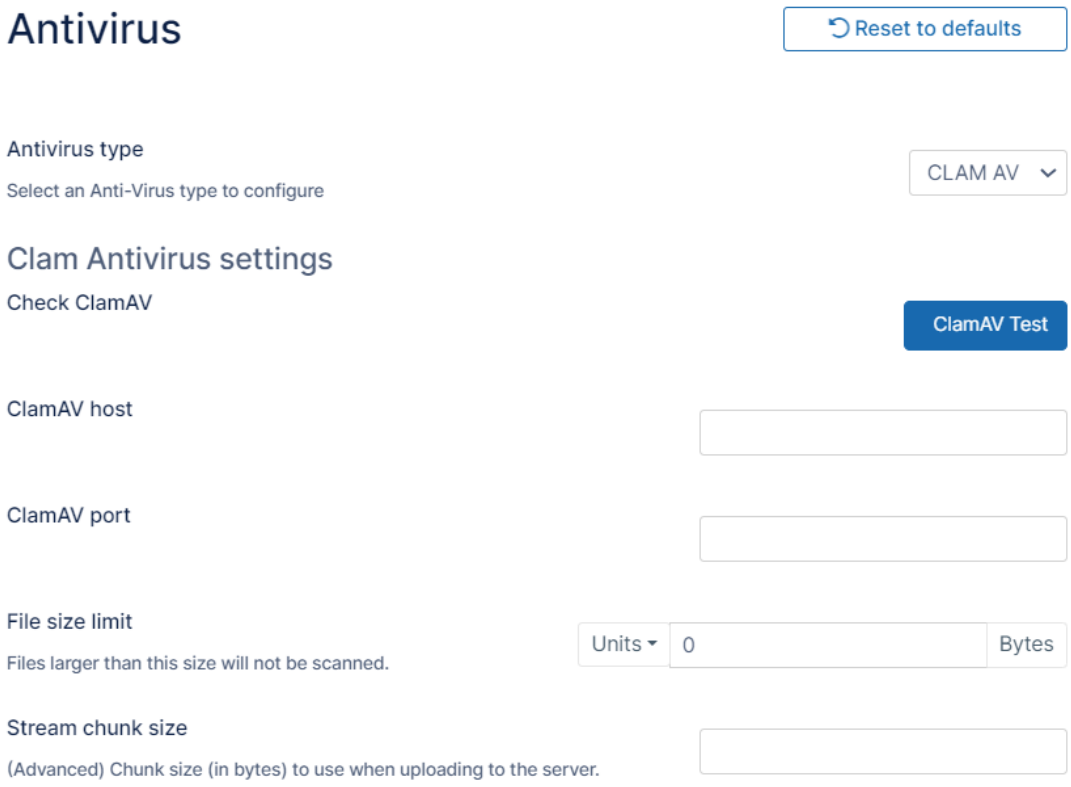
Once ClamAV is set up and started, add details of the ClamAV service to FileCloud.

To integrate ClamAV with FileCloud:

1. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings**. Then, on the **Settings** navigation page, click **Third Party Integrations** .
   The **Antivirus** page opens by default.
2. In **Anti-Virus Type**, choose **Clam AV**.



Clam Antivirus settings appear.



3. Enter the following information:

Setting	Description
ClamAV Host	Enter the URL or IP of the system where Clam AV is running. This can be local or remote system.
ClamAV Port	The port used by ClamAV (This is set when ClamAV is installed in the previous section)
Skip scanning for files greater than	This is the file limit in bytes that will be scanned. For example, very large files can be excluded from scanning. Default value is 25MB
Stream Chunk Size	This is a advanced setting used to stream the file content to ClamAV for scanning. Default is 8KB.

4. Click **Save**.
5. To verify connectivity, click the **ClamAV Test** button.

Once the ClamAV configuration is set up, every file uploaded to FileCloud will be scanned before being added to FileCloud storage.

- If a file fails AV check (i.e. a virus detected) then the file will be deleted and an entry will be added to the Audit log with the details of the file.

If scanning fails

If scanning fails because the ClamAV server is down, a message appears on your screen, and your Manage Alerts page displays the warning:
Unable to communicate with ClamAV Server. Check immediately.

By default, if ClamAV fails to scan a file because the ClamAV server is down, the file is not deleted.

To automatically delete files if ClamAV scan fails because the ClamAV server is unavailable:

1. Open the configuration file:
   Windows: XAMPP DIRECTORY/htdocs/config/cloudconfig.php
   Linux: /var/www/config/cloudconfig.php
2. Add the line:

   ```
   define("TONIDOCLOUD_CLAMAV_DELETE_ON_SCAN_FAIL", "1");
   ```

   Now, when scan fails, the file is deleted, and the audit log displays the message: **ClamAV removed [FILE_PATH] due to scan fail**.

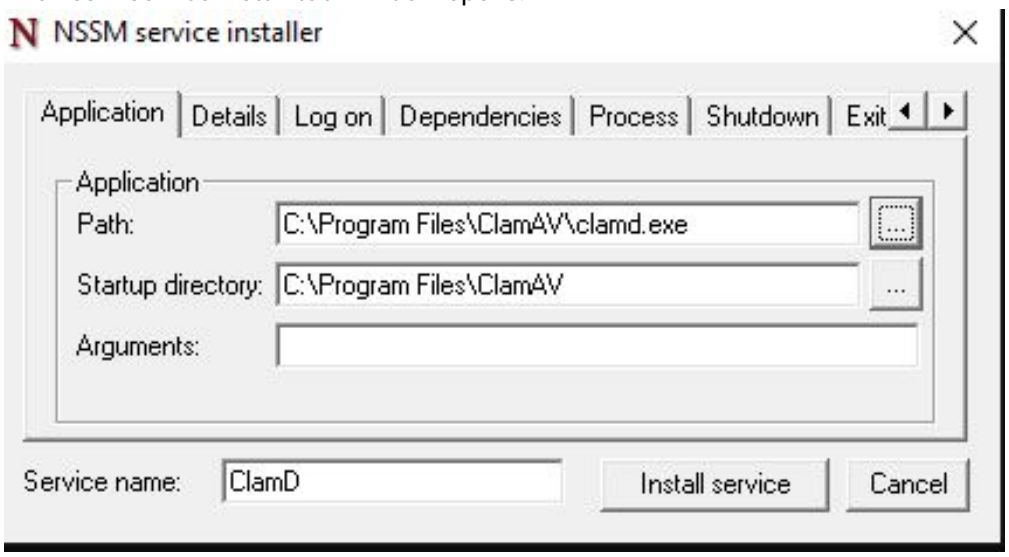
   If TONIDOCLOUD_CLAMAV_DELETE_ON_SCAN_FAIL is enabled and the CLAMAV server is not available, FileCloud does not allow files to be uploaded.

Clam backup

1. Download the latest version of ClamAV installer from:
   http://www.clamav.net/downloads/production/ClamAV-0.103.1.exe
2. Install ClamAV by running the ClamAV-0.103.1.exe installer file.
3. Download the nssm Service Manager from:
   https://patch.codelathe.com/tonidocloud/live/3rdparty/nssm/nssm.zip .
4. Uzip the nssm folder and move the nssm folder to the C:\ drive or, if you are installing ClamAV in the FileCloud Server, to the C:\xampp folder.
5. Navigate to the nssm folder in command line and run the below command

```
C:\nssm>nssm install ClamD
```

The nssm service install tool window opens:



6. To install the service, select the clamd.exe file path in **Application Path** and click **Install Service**.
7. Copy **clamd.conf.sample** and **freshclam.conf.sample** from **C:\Program Files\ClamAV\conf_examples** to **C:\Program Files\ClamAV**, and rename them **clamd.conf** and **freshclam.conf**
8. In **clamd.conf** and **freshclam.conf**, comment out the line beginning with *Example*.
9. If ClamAV is installed on a server other than FileCloud server, bind the IP address of the server in **clamd.conf** by changing the IP address for **TCPAddr**.
10. To update the ClamD database, enter:

```
cd C:\Program Files\ClamAV
freshclam.exe
```

The console response should appear similar to:

```
C:\Program Files\ClamAV>freshclam.exe
ClamAV update process started at Thu Mar 25 07:42:28 2021
daily database available for download (remote version: 26120)
Time:    1.1s, ETA:    0.0s [=========================>]  100.57MiB/100.57MiB
Testing database: 'C:\Program Files\ClamAV\database\tmp.499b3311a3\clamav-e97910f98bef89730f7030c4c8d55340.tmp-daily.cvd' ...
Database test passed.
daily.cvd updated (version: 26119, sigs: 3965409, f-level: 63, builder: raynman)
main database available for download (remote version: 59)
Time:    1.3s, ETA:    0.0s [=========================>]  112.40MiB/112.40MiB
Testing database: 'C:\Program Files\ClamAV\database\tmp.499b3311a3\clamav-6a0c33de13396c36c4b039985d2b5d2f.tmp-main.cvd' ...
Database test passed.
main.cvd updated (version: 59, sigs: 4564902, f-level: 60, builder: sigmgr)
bytecode database available for download (remote version: 333)
Time:    0.1s, ETA:    0.0s [=========================>]  286.79KiB/286.79KiB
Testing database: 'C:\Program Files\ClamAV\database\tmp.499b3311a3\clamav-704d345ffd1cf864b6a0781e984101dc.tmp-bytecode.cvd' ...
Database test passed.
bytecode.cvd updated (version: 333, sigs: 92, f-level: 63, builder: awillia2)
```

11. Start the service **ClamD** from Windows Services.
12. Verify the service is running and bind it to the localhost IP address or the IP address of the ClamAV server by running the following command:

```
netstat -ano |findstr 3310
```

Use ICAP Antivirus Scanning

FileCloud uses Internet Content Adaption Protocol (ICAP) to integrate with any antivirus product currently supporting ICAP.

What is ICAP?

ICAP is a generic protocol that allows web servers to offload specialized tasks to custom-built servers. Examples of such specialized tasks include DLP (data loss prevention) based content scanning, URL filtering and antivirus scanning.

FileCloud's ICAP integration feature:

- Works on both Linux and Windows servers
- Is part of FileCloud server itself
- Provides flexibility and scalability - the ICAP antivirus server does not have to be deployed on the same server as the one running the FileCloud server instance.
- Triggers virus scanning only when files are uploaded to FileCloud.
- Scanning is scheduled "inline" as soon as the file upload is completed

> If you have already purchased your own antivirus solution and want to use it, or if you do not want to use ClamAV for various reasons, we highly recommended using this feature.

We also recommend that the ICAP Antivirus server administrator consult the antivirus product documentation to understand the operational and configuration parameters, capabilities and limitations. As virus scanning is a critical feature for maintaining water-tight security and smooth functioning of any workplace, consulting the documentation is important before configuring FileCloud's ICAP integration settings, it would also help in troubleshooting and maintenance.

How ICAP detects a virus

After a file is scanned, FileCloud checks for the following response headers on the file scanning result:

- X-Infection-Found
- X-Violations-Found

- X-Virus-ID

If any of these headers are found, FileCloud performs the actions listed below, under **When ICAP detects a virus**.

When ICAP detects a virus

Similar to the case of ClamAV, if FileCloud's ICAP Client has been configured correctly with a properly deployed ICAP AV server, when a virus is detected in an uploaded file, the following actions occur:

1. The incoming file is deleted.
2. An alert is displayed in the Admin Portal.
3. A toast is displayed in the User Portal.
4. An entry is added in the audit log about virus detection in the file and subsequent deletion of the file.

Integrating ICAP with FileCloud

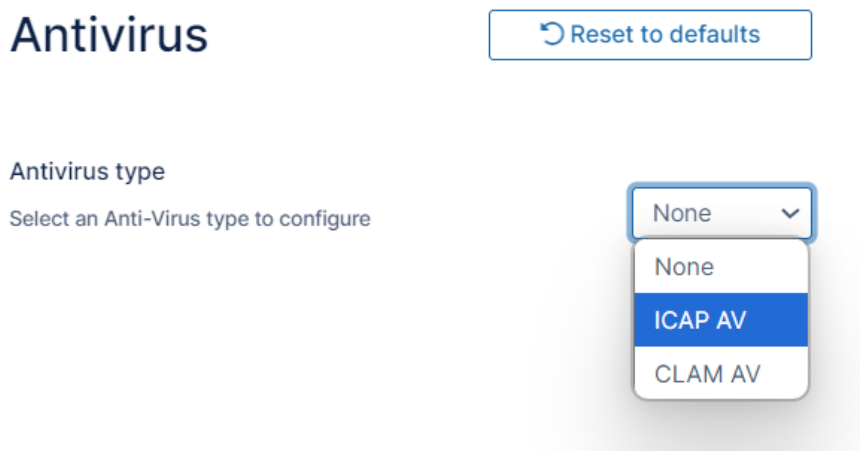
Using ICAP to integrate Antivirus capabilities into FileCloud requires customers to:

1. Set up an ICAP antivirus server.
2. Configure FileCloud's inbuilt ICAP client to access your antivirus server.
   FileCloud has made it easy for administrators to connect FileCloud to your antivirus server by including an inbuilt ICAP Client.

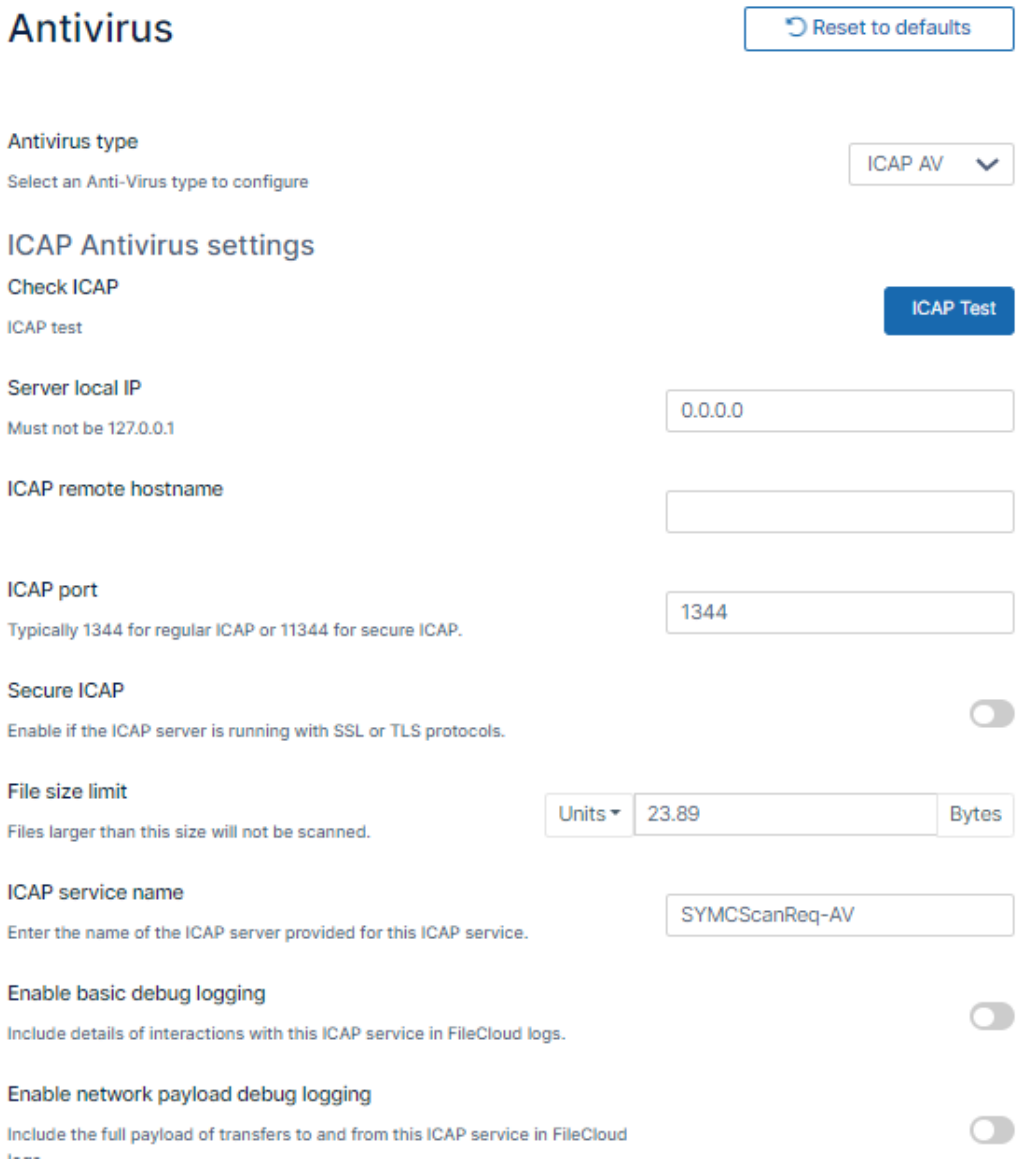
   The configuration steps apply to both Windows and Linux servers.

To configure FileCloud to use your antivirus server:

1. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings**. Then, on the **Settings** navigation page, click **Third Party Integrations** .
   The **Antivirus** page opens by default.
2. In **Anti-Virus Type**, choose **ICAP AV**.



ICAP Antivirus settings appear.



3. Configure the various parameters for the ICAP Client as described below.

Setting	Description
Server local IP	In most cases, leave the default value of 0.0.0.0. If you are using a separate FileCloud policy with ICAP, enter the Private (LAN) IP of the FileCloud server.
ICAP remote hostname	Enter the hostname or IP of the system where the ICAP AV is deployed.

Setting	Description
ICAP port	Leave the default value of 1344 as it is. In rare cases, this might need to be changed to whatever port the ICAP AV server is listening on.
Secure ICAP	Enable if the ICAP server is running with SSL or TLS protocols.
File size limit	This is the file limit in bytes that will be scanned. For example, very large files can be excluded from scanning. Default value is 25MB
ICAP service name	Consult the ICAP AV server product documentation to know this value. It must be set correctly otherwise integration won't work.
Enable basic debug logging	Check this to enable logging of detailed operational debug messages in the (error) logs.
Enable network payload debug logging	Check this to enable logging of detailed network communication related debug messages in the (error) logs.

4. To save your changes, click **Save**.
5. To confirm that the configuration has been done correctly, click **ICAP Test**.

User details sent with scan requests

To help the ICAP server determine if a scan is required, the following headers are sent with every scan request:

Header X-FILECLOUD-USER-NAME - name of user performing the upload.
Header X-FILECLOUD-USER-EMAIL - email of user performing the upload.
Header X-FILECLOUD-USER-TYPE - type of user performing the upload. Possible values are "full", "guest", and "external".
Header X-FILECLOUD-GROUP-NAMES - comma-separated list of group names that user performing the upload is a member of.

To disable sending of these headers:

1. Open the configuration file:
   Windows: XAMPP DIRECTORY/htdocs/config/cloudconfig.php
   Linux: /var/www/config/cloudconfig.php
2. Add the line:

```
define("TONIDOCLOUD_ICAPAV_DISABLE_ADDITIONALHEADERS", "1");
```

If scanning fails

If scanning fails because the ICAP server is down, a message appears on your screen, and your Manage Alerts page displays the message:

☐	2020-Jul-29 07:06 PM	Warning	Unable to communicate with ICAP/AV Server. Check immediately.

By default, if ICAP fails to scan a file because the ICAP server is down, the file is not deleted.

To automatically delete files if ICAP scan fails because the ICAP server is unavailable:

1. Open the configuration file:
   Windows: XAMPP DIRECTORY/htdocs/config/cloudconfig.php
   Linux: /var/www/config/cloudconfig.php
2. Add the line:

   ```
   define("TONIDOCLOUD_ICAP_DELETE_ON_SCAN_FAIL",1);
   ```

   Now, when scan fails, the file is deleted, and the audit log displays the message: *ICAP removed [FILE_PATH] due to scan fail.*

   > If TONIDOCLOUD_ICAP_DELETE_ON_SCAN_FAIL is enabled and the ICAP server is not available, FileCloud does not allow files to be uploaded.

To extend the allowed response time before an ICAP scan request fails:

1. Open the configuration file:
   Windows: XAMPP DIRECTORY/htdocs/config/cloudconfig.php
   Linux: /var/www/config/cloudconfig.php
2. Add the line:

   ```
   define("TONIDOCLOUD_ICAP_AV_TIMEOUT",80);
   ```

   Where the value entered specifies that amount of response time allowed in seconds. As the default allowed response time is 60 seconds, the value should be higher than 60 to increase the allowed response time.

Integrating FileCloud with Salesforce

The Salesforce Plugin reached End-of-Life (EoL) on September 1, 2025, and support is no longer be generally available for the Salesforce Plugin.
No replacement product is planned.

Salesforce Integration
FileCloud makes files stored in any on-premises, public or hybrid cloud available within Salesforce. To configure this function, integrate FileCloud with Salesforce.
Key benefits:

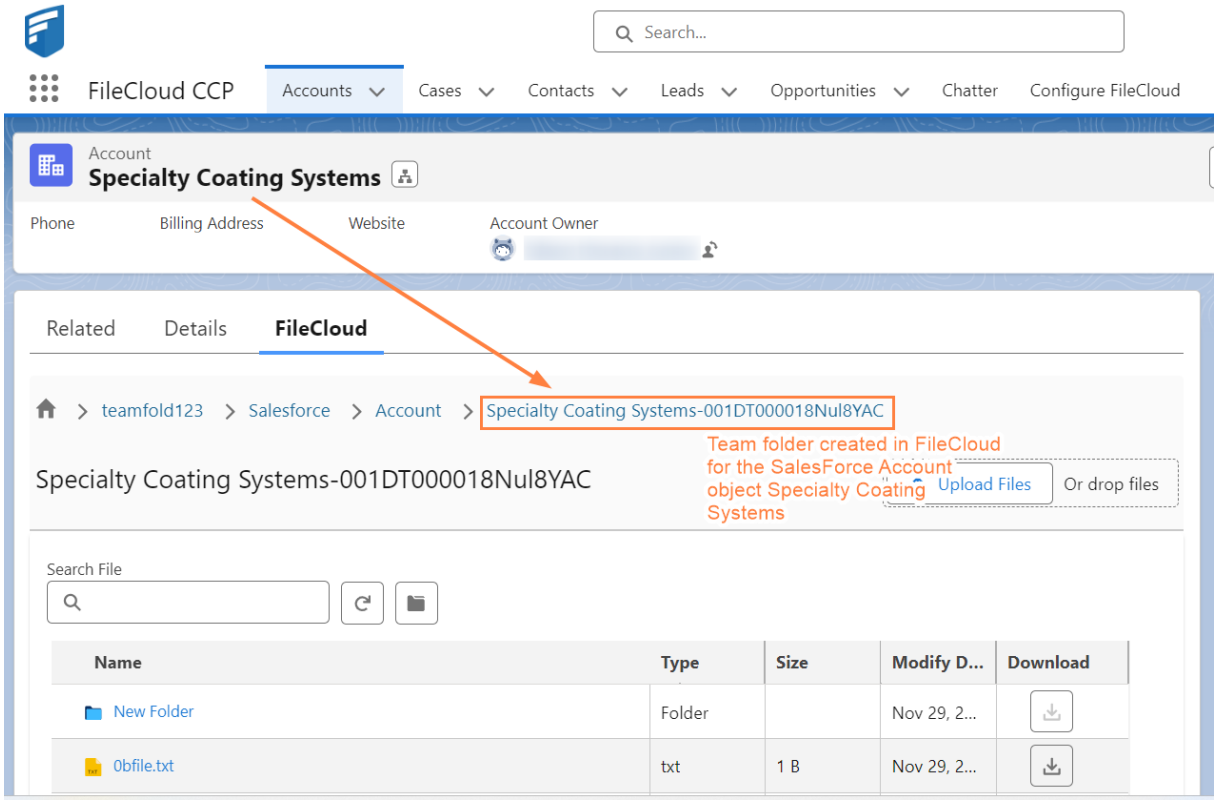
- Upload, download, access and share remote files from within Salesforce.
- Store files on-premises or in the public cloud (Amazon AWS, Microsoft Azure). Access files securely inside Salesforce from anywhere.
- Share files and collaborate with team members, even if they are not Salesforce users.
- Integrate Salesforce with existing file servers and file permissions.
- Get advanced file analytics about who has shared and downloaded files.
- Link FileCloud content to specific Salesforce records.

Limitations

- To be able to integrate FileCloud with Salesforce, you must have the Salesforce component in your license.
- You cannot give External users access to FileCloud's integration with Salesforce.
- Only one Salesforce account and one FileCloud account can be mapped together. Mapping occurs the first time the user logs in to FileCloud through Salesforce. If a user tries to map a second FileCloud account to a Salesforce account, or a second Salesforce account to a FileCloud account, an error message is returned.

To integrate FileCloud with Salesforce, create a Salesforce Team Folder in FileCloud. When you create Salesforce objects (Accounts, Cases, Contacts, etc.), sub-folders are created in the Salesforce Team Folder in FileCloud for each object.

You can access FileCloud in the Salesforce interface to access the an object's Team Folders to perform FileCloud operations on them.



You can configure the Salesforce Team Folders so that only the owner (creator) of the object and users you have designated as managers have access to each object's Team Folder. If you do not add this configuration, anyone with access to the parent Salesforce Team Folder has access to all objects' Team Folders.

Adding FileCloud to Salesforce

Configuring FileCloud with Salesforce

After you install/integrate FileCloud with Salesforce, complete the following:

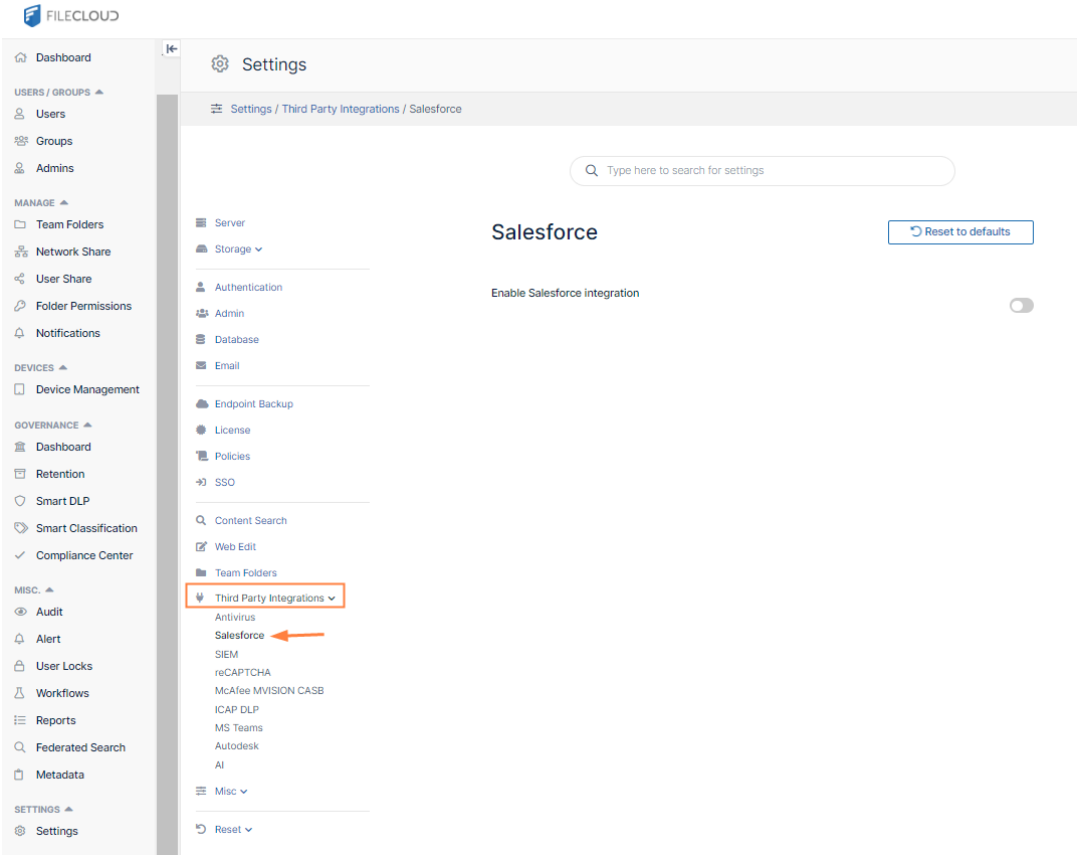
1. Edit the **.htaccess** file.
    a. Windows: go to **C:\xampp\htdoc**
    Linux: go to: **/var/www/html/config**
    b. Open the file **.htaccess**
    c. Locate **Header set Content-Security-Policy** and in the list following **frame-ancestors**, append *****.http://visualforce.com ***.lightning.force.com** ***.my.salesforce.com,**

***.vf.force.com;**

The edit is shown in the highlighted portion below:

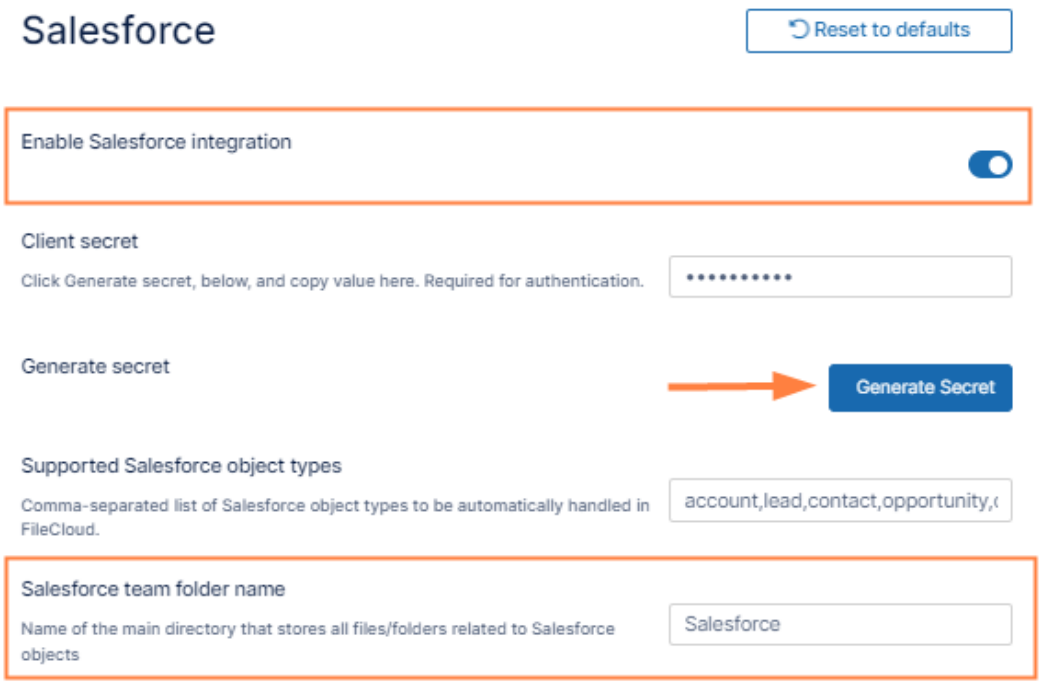
```
<IfModule mod_headers.c>
Header set X-Frame-Options "SAMEORIGIN"
Header set X-Content-Type-Options "nosniff"
Header set Strict-Transport-Security "max-age=31536000; includeSubDomains"
Header set X-XSS-Protection "1; mode=block"
Header set Content-Security-Policy: "default-src 'self' blob: *.live.com *.amazonaws.com *.core.windows.net www.google.com http://127.0.0.1:34320/v1/fileassociations *.autodesk.com; \
    style-src 'unsafe-inline' 'self' *.autodesk.com; \
    script-src 'unsafe-inline' 'unsafe-eval' 'self' www.google.com www.gstatic.com  teams.microsoft.com *.teams.microsoft.com *.skype.com *.autodesk.com; \
    frame-src 'self' www.google.com *.live.com docs.google.com accounts.google.com; \
    font-src 'self' data: *.autodesk.com; \
    img-src www.gstatic.com 'self' data: blob: *.duosecurity.com *.live.com *.amazonaws.com *.core.windows.net *.office.net *.autodesk.com; \
    frame-ancestors 'self' teams.microsoft.com *.teams.microsoft.com *.skype.com *.my.salesforce.com *.visualforce.com *.lightning.force.com *.my.salesforce.com *.vf.force.com; \
    worker-src 'self' blob: *.autodesk.com"
Header set Cache-Control no-cache="Set-Cookie"
</IfModule>
```

2. Configure Salesforce in FileCloud.
   a. In the FileCloud admin portal, open the **Salesforce** settings page.
      In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the **Settings** navigation page, click **Third Party Integrations** .
   b. In the inner navigation bar on the left of the **Third Party Integrations** page, expand the **Third Party Integrations** menu, and click **Salesforce**, as shown below.

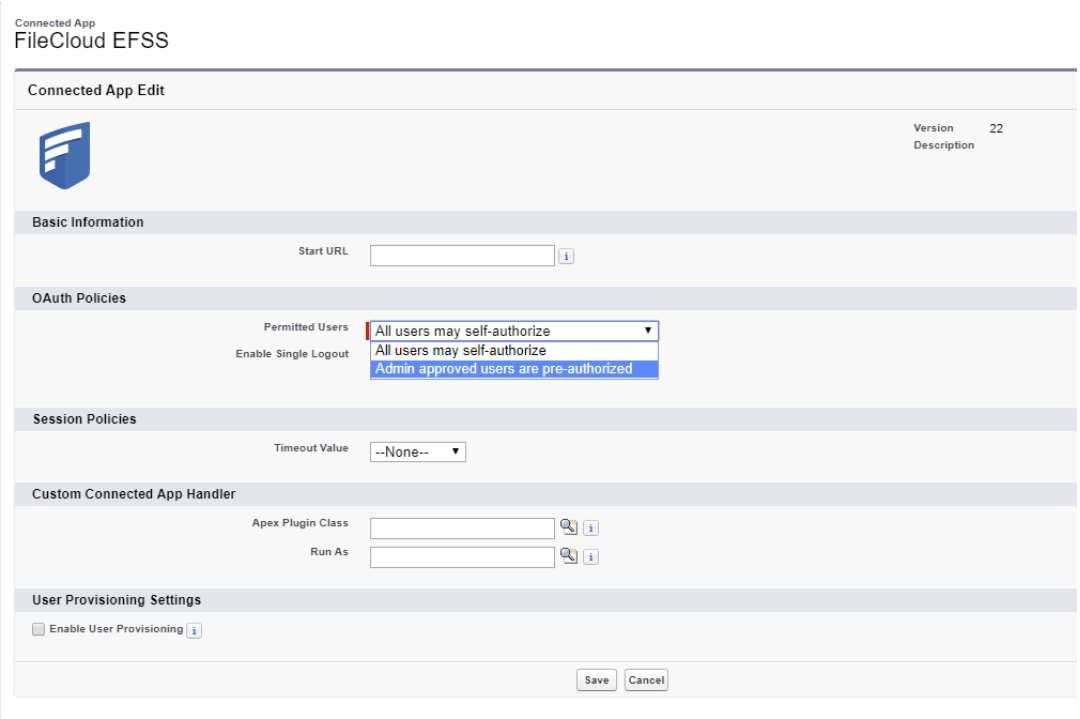


      The **Salesforce** settings page opens.
   c. Enable the **Enable Salesforce integration** setting.
   d. Click **Generate Secret**, then copy the key and click **Save**.
   e. In FileCloud Team Folders, create a Team Folder named **Salesforce**. Sub-folders for your Salesforce objects will automatically be created in this Team Folder. (If you have given the folder another name, but make sure you change the folder name entered in **Salesforce**

team folder name to match it.)

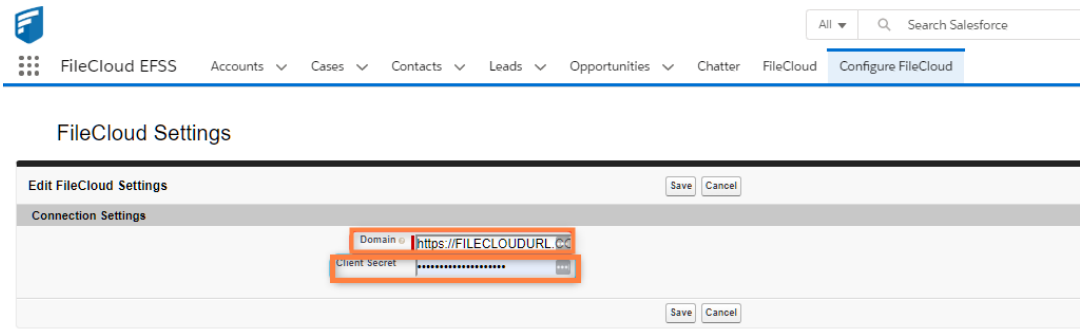


3. Configure which users have access to FileCloud's integration with Salesforce.
   a. In the Salesforce **App Manager,** click the drop-down list across from **FileCloud EFSS**, and click **Manage**.
   b. Click **Edit Policies**.
   c. Under **OAuth policies**, in the **Permitted Users** drop-down list choose **Admin approved users are pre-authorized**.

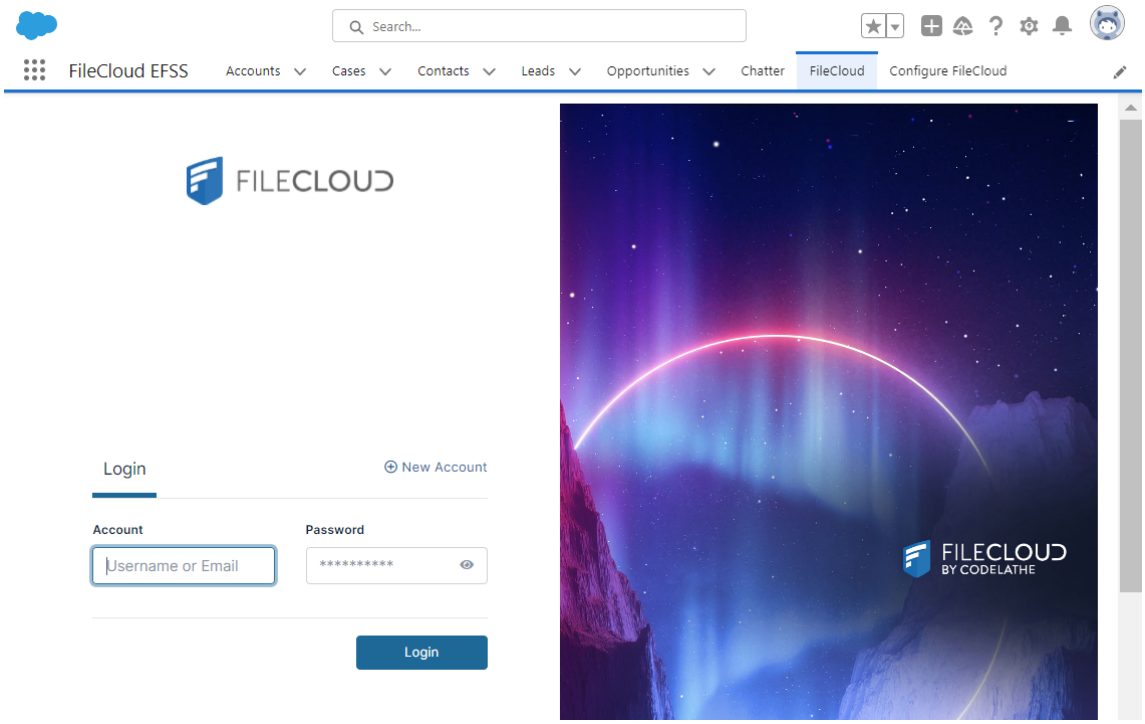


d. Click **Save**.

4. Proceed with the configuration of FileCloud within Salesforce.
   a. Access Salesforce and click on the **Configure FileCloud** tab.
   b. On the **Configure FileCloud** tab click edit.
   c. Add your FileCloud URL under **Domain** and paste the Secret Key generated in Step 2 into **Client Secret.**
   d. Click **Save**.

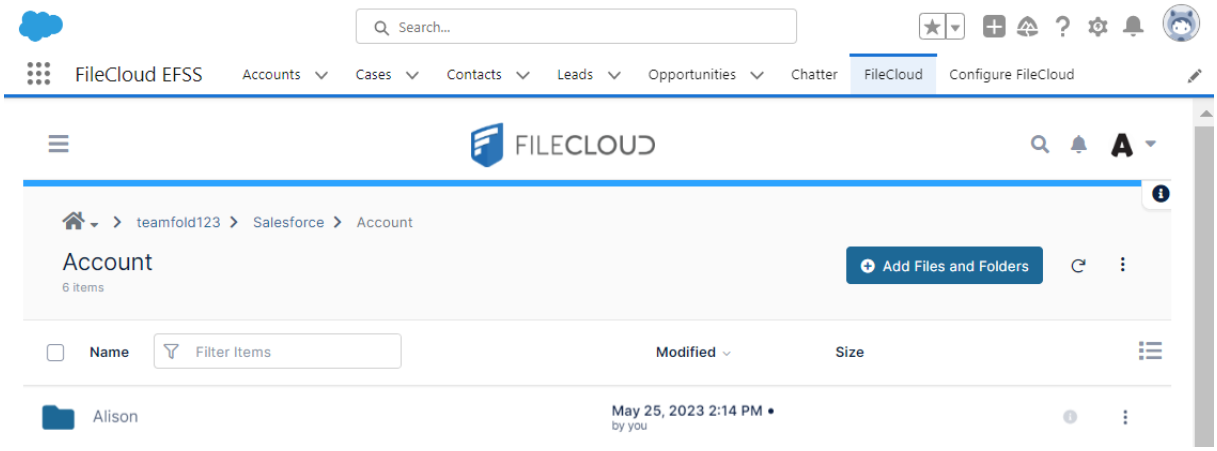


5. **Click the FileCloud tab (to the left of Configure FileCloud tab).**
   FileCloud should load and allow you to log in.



Restricting Permissions on Salesforce Team Folders

Now that you have integrated FileCloud and Salesforce, when you create an object in Salesforce, a sub-folder in the Salesforce Team Folder in FileCloud is created for the object.



Since you may want more restrictive permissions on each object's folder when it is created, you can configure FileCloud to only enable the owner (creator) of the object and a group of users that you designate as managers to have access to the object folder.

To configure more restrictive default permissions on Team Folders for Salesforce objects:

1. If you have not already shared the Salesforce Team Folder with all FileCloud users or groups who may want to access an object sub-folder, give them access to the Salesforce Team Folder in FileCloud now.
2. Open the configuration file:
   Windows: XAMPP DIRECTORY/htdocs/config/cloudconfig.php
   Linux: /var/www/config/cloudconfig.php
3. Add the following lines, listing the emails of users who you want to be able to access all Salesforce object folders in the second setting:

```
define('TONIDOCLOUD_SALESFORCE_RESTRICT_ACCESS_ENABLED', '1');
define('TONIDOCLOUD_SALESFORCE_MANAGER_USERS_EMAILS', ['email1@filecloud.com',
'email2@filecloud.com']);
```

4. Save your changes.
   Note: To turn off these restrictions, set TONIDOCLOUD_SALESFORCE_RESTRICT_ACCESS_ENABLED to 0.

SIEM Integration

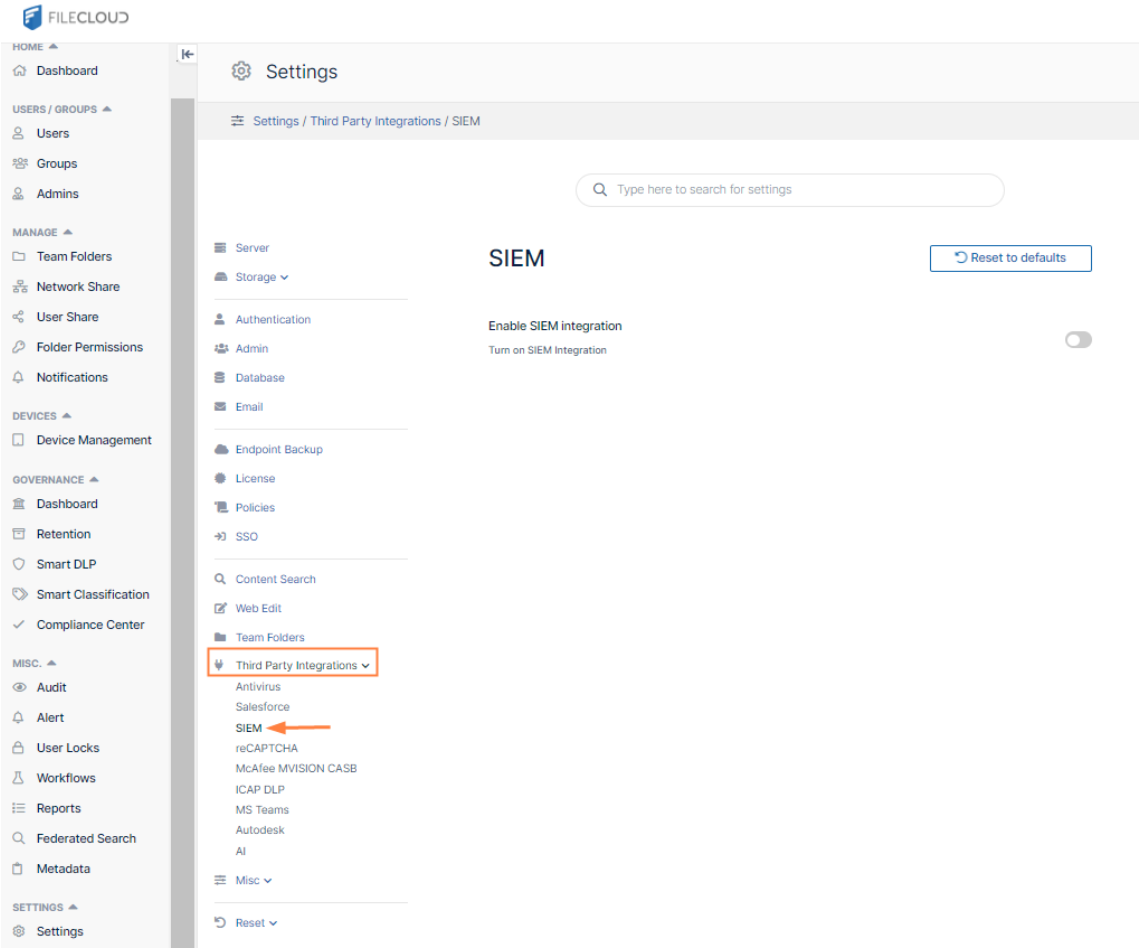
Security information and event management (SIEM) products and services provide analysis of security alerts generated by applications and network hardware.

FileCloud can integrate its system alerts and auditing with external SIEM systems, enabling you to monitor all alerts and potential security issues in one place.

Open the SIEM settings page

To go to the SIEM settings page

1. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the **Settings** navigation page, click **Third Party Integrations** .
2. In the inner navigation bar on the left of the **Third Party Integrations** page, expand the **Third Party Integrations** menu, and click **SIEM**, as shown below.

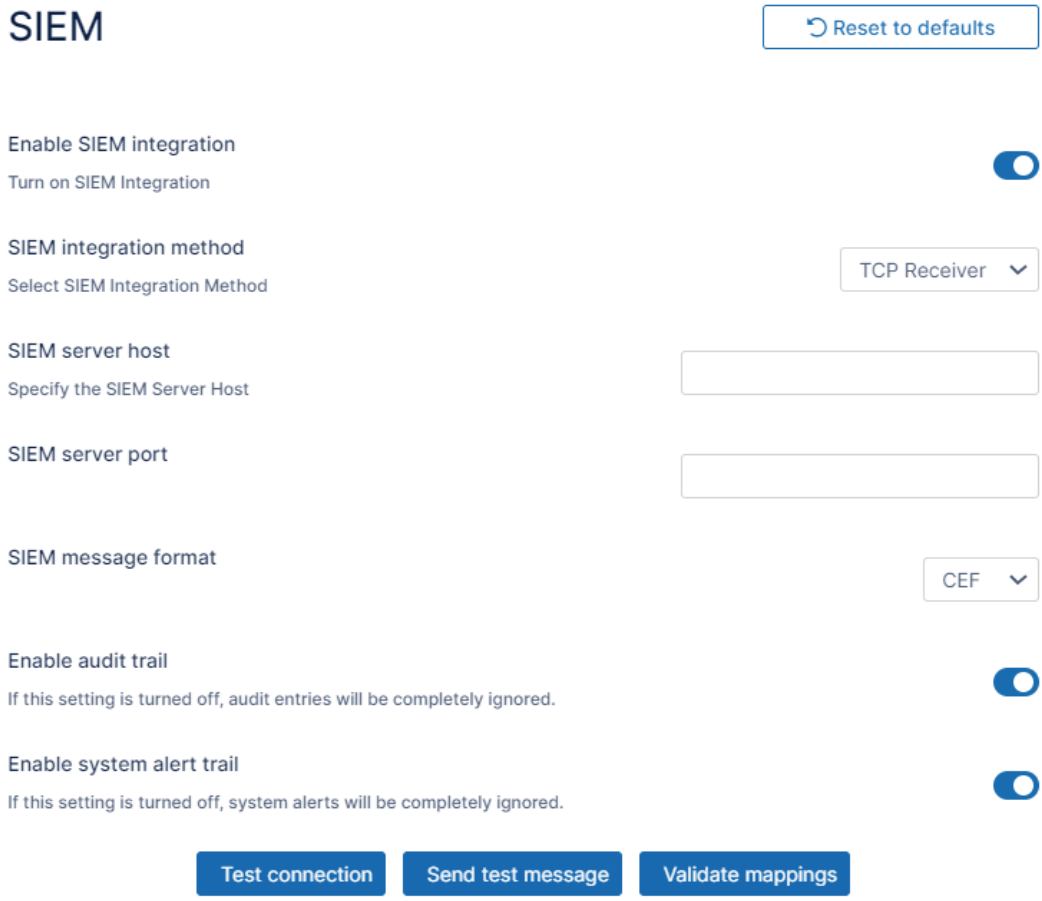


The **SIEM** settings page opens.

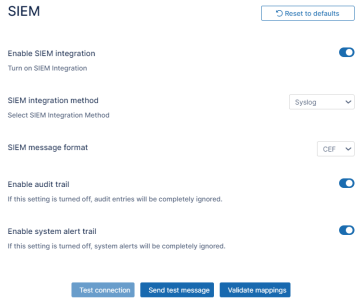
Set up SIEM

1. To activate SIEM integration, click the grayed out **Enable SIEM integration** button.

SIEM integration fields appear.



2. Modify the settings using the information in the following table.

Option	Description
SIEM Integration method	Specifies the SIEM Integration method. Following options are available: • **TCP Receiver** - messages are sent to the specified SIEM server endpoint (host and port) via TCP socket connection. • **UDP Receiver** - messages are sent to the specified SIEM server endpoint (host and port) via UDP socket connection. • **Syslog** - messages are written directly to the Syslog, which can be imported by the SIEM server. If you choose **Syslog**, the **SIEM server host** and **SIEM server port** fields are not shown, and the **Test connection** button is disabled.  **Note:** SIEM software providers should specify supported integration methods in the SIEM documentation.
SIEM server host (TCP and UDP integration only)	URL or IP Address of the SIEM server.
SIEM server port (TCP and UDP integration only)	Port exposed by the SIEM Server for the given socket connection.
SIEM message format	Specifies the SIEM Message format. The following formats are available: • **CEF** - Common Event Format • **LEEF** - Log Event Extended Format. If you select **LEEF** the fields **LEEF Version** and **LEEF Message Delimiter** also appear: NOTE: SIEM software provider should specify supported formats in the SIEM documentation.

Option	Description
LEEF version (LEEF Format only)	Specifies the version of the LEEF format message. Available versions: • 1.0 • 2.0
LEEF message delimiter (LEEF Format only)	The delimiter to be used for LEEF messages. The options are **whitespace** and **tab**. Choose the option that is compatible with the SIEM tool you are using.
Enable audit trail	Specifies whether Audit records should be processed and send to the SIEM Server. Please check the Managing SIEM mappings section for more details.
Enable system alert trail	Specifies whether System Alerts generated within FileCloud should be processed and send to the SIEM Server. Please check the Managing SIEM mappings section for more details.
Test connection (TCP and UDP integration only)	Tests connection to the server specified by the Host and Port. **NOTE: All settings have to be saved first. Connection tests are based on the *currently* saved settings.**
Send test message	Sends a test message in the given format (CEF/LEEF) to the SIEM server specified by the Host and Port or saves a test message to the Syslog. **NOTE: All settings have to be saved first. Connection tests are based on the *currently* saved settings.**
Validate mappings	Validates all defined mappings. Please check the Managing SIEM mappings section for more details.

Syslog Integration

In order to provide more flexibility, FileCloud allows admins to specify two important Syslog parameters - ident and facility. **Ident** specifies the name of the application logged in Syslog. **Facility** specifies where all FileCloud messages are sent and can be utilized by the system level Syslog configuration (e.g. in "rsyslog"). Both settings can be overridden in the *cloudconfig.php* configuration file by inputting the following settings:

- Ident - to specify ident value, add the following setting to *cloudconfig.php*

  ```
  define('TONIDOCLOUD_SIEM_SYSLOG_IDENT', 'IDENT_VALUE');
  ```

  If no value is provided, by default it will be set to 'SIEM'.

- Facility -to specify ident value please add the following setting: to the *cloudconfig.php*

```
define('TONIDOCLOUD_SIEM_SYSLOG_FACILITY', LOG_LOCAL2);
```

If no value is provided, by default it will be set to LOG_LOCAL5. Below is a full list of supported values.

`LOG_AUTH`	Security/authorization messages (use `LOG_AUTHPRIV` instead in systems where that constant is defined)
`LOG_AUTHPRIV`	Security/authorization messages (private)
`LOG_CRON`	Clock daemon (cron and at)
`LOG_DAEMON`	Other system daemons
`LOG_KERN`	Kernel messages
`LOG_LOCAL0` ... `LOG_LOCAL7`	Reserved for local use. These are not available in Windows
`LOG_LPR`	Line printer subsystem
`LOG_MAIL`	Mail subsystem
`LOG_NEWS`	USENET news subsystem
`LOG_SYSLOG`	Messages generated internally by syslogd
`LOG_USER`	Generic user-level messages
`LOG_UUCP`	UUCP subsystem

LOG Values can also be seen in the official PHP documentation.

> ⚠ Please note that there are no quotation marks used for LOG values, as these have to be set to one of the PHP constants.

Managing SIEM Mappings

The biggest challenge when working with the external SIEM servers is to map messages existing in the system to the correct CEF/LEEF format. In order to allow administrators to have full control of how to represent FileCloud's system alerts and audit records in the external SIEM system a special, flexible mapping syntax is supported.

Accessing SIEM mappings files

NOTE:
For this step you will need to access **WWWROOT.** It is typically located at:

Windows	Linux
c:\xampp\htdocs	**/var/www/html**

Create and access SIEM mappings files:

Navigate to the following directory:

```
WWWROOT/app/siem/maps
```

It contains the following files:

```
auditmap-sample.php
systemalertsmap-sample.php
```

which store mapping samples for audit and system alerts respectively.

Modify the mappings to correspond to your system, and save them as **auditmap.php** and **systemalertsmap.php**.

- **auditmap.php** enables FileCloud to convert audit entries to the valid SIEM messages.
- **systemalertsmap.php** enables FileCloud to convert FileCloud's system alerts to the valid SIEM messages.

NOTE: Mappings are stored in the .php file, so they have to follow all PHP syntax rules as well as the internal mappings rules and syntax. To validate all mappings please navigate to **Settings → Third Party Integrations → SIEM** and click the **Validate mappings** button.

When you upgrade FileCloud, if you previously integrated with SIEM and already have auditmap.php and systemalertsmap.php files, you do not have to recreate or edit them unless you want to change existing mappings.

SIEM mapping format

A sample SIEM mapping is a PHP array entry, which itself is an array. It contains following fields:

id (Required) - identifies the SystemAlert/Audit entry this map refers to. **NOTE: It can be a string literal which matches the audit operation name or one of the SiemArea values available in FileCloud, an array of values or a wildcard '*' that specifies that the mapping is applied to ALL audit entries/system alerts.**

prefilter (Optional) - A collection of preconditions that event has to meet in order to be processed and sent to the SIEM system. It is an array of filters, where each filter has the following format: property => value, where:

- property is a valid property available for the Audit / System Alert record (TBD - add lists of properties)
- value is a value that has to be matched in order to process the Audit / System Alert record, i.e.

Sample System Alert Mappings

```
    'prefilter' => [
        'level' => SysAlert::SYSALERT_LEVEL_MELTDOWN
    ],
```

specifies that only System Alerts with the Meltdown criticality level would be sent to the SIEM server.

map (Required) - specifies the actual mapping between the FileCloud object being processed and the SIEM-formatted message that will be sent to the SIEM server. SIEM object as to contain the following four fields:

- eventClass - class of the event in the SIEM system.
- eventName - name of the event.
- severity - this is a SIEM side severity, which is a number from the 1-10 range.
- extension - a collection (array) of additional key value pairs that will be stored in the SIEM system (i.e. user that performed the action, ip address of the request, etc.). The key can be any arbitrary string.

To allow a very flexible way to resolve those mappings value a special 'language' was created. Values can be provided in any of the following ways:

- As a literal value (i.e. string or number), i.e.

  Sample System Alert Mappings

```
'eventClass' => 'authentication',
'eventName' => 'invalid login',
'severity' => 3
```

- As a property biding that will resolve the value, based on the actual value provided by the FileCloud audit, system alert being processed:
  Sample System Alert Mappings

  ```
  'eventClass' => '$siemArea',
  'eventName' => '$description',
  'user' => '$username',
  'ip' => '$ip'
  ```

  Please check a full list of supported properties for more details. (TBD)
- As a method call:

  Sample System Alert Mappings

  ```
  'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'], ['$level']],
  ```

  NOTE: Users can create their own methods that can be utilized here. The first parameter is the PHP callback (class, method name) and the second parameter is the array of values (Optional) that will be processed by that callback. Parameters can be set to literal values or runtime-resolvable properties as described earlier. In FileCloud 19.2 *getSysAlertSeverity* is the only method available out of the box. It converts internal System Alerts severity into the 1-10 range required by SIEM integration in the following way:
  - Meltdown: 10
  - Critical: 7
  - Warning: 4
  - Information: 1

Sample mappings
System Alerts:
Sample System Alert Mappings

```
//Report all meltdowns
$mappings[] = [
    'id' => '*', //Wildcard denotes all Alerts
    'prefilter' => [
        'level' => SysAlert::SYSALERT_LEVEL_MELTDOWN
    ],
    'map' => [
```

```
        'eventClass' => '$siemArea',
        'eventName' => '$description',
        'severity' => 10,
        'extension' => [
            'user' => '$username',
            'ip' => '$ip'
        ]
    ]
];

//AV system alert - infected file found
$mappings[] = [
    'id' => SiemArea::INFECTED_FILE,
    'map' => [
        'eventClass' => 'System Error',
        'eventName' => '$description',
        'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'],
['$level']],
        'extension' => [
            'user' => '$username',
            'ip' => '$ip',
            'path' => '$alertContext.filePath',
            'file' => '$alertContext.fileName'
        ]
    ]
];

//Type mismatch report
$mappings[] = [
    'id' => SiemArea::INVALID_FILE_TYPE,
    'map' => [
        'eventClass' => 'System Error',
        'eventName' => '$description',
        'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'],
['$level']],
        'extension' => [
            'user' => '$username',
            'ip' => '$ip',
            'path' => '$alertContext.file'
        ]
    ]
];
```

Audit:

```
//Report all audit events
$mappings[] = [
    'id' => '*',
    'prefilter' => [],
    'map' => [
        'eventClass' => '$operation',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'user' => '$userName',
```

```
            'userAgent' => '$userAgent',
            'ip' => '$ip',
            'notes' => '$notes'
        ]
    ]
];


//Failed login attempt
$mappings[] = [
    'id' => 'loginguest',
    'prefilter' => [
        //List of conditions that audit entry has to met in order to be processed (or
filtered out if excluded option is there)
        'resultCode' => '0', //incidents only
        'exclude' => false// - optional 'include' is used by default
    ],
    'map' => [
        'eventClass' => 'login',
        'eventName' => 'Invalid login attempt',
        'severity' => 2,
        'extension' => [
            'user' => '$userName',
            'ip' => '$ip'
        ]
    ]
];
```

Managing SIEM Mappings

The biggest challenge when working with the external SIEM servers is to map messages existing in the system in the correct CEF/LEEF format. In order to allow administrators to have full control over how to represent FileCloud's System Alerts and Audit records in the external SIEM system a flexible mapping syntax is supported.

SIEM Mappings - general rules

Create and access SIEM mappings files

Access **WWWROOT.** It is typically located at:

Windows	Linux
c:\xampp\htdocs	**/var/www/html**

Navigate to the following directory:

```
WWWROOT/app/siem/maps
```

It contains the following files:

```
auditmap-sample.php
systemalertsmap-sample.php
```

These files store mappings for audit and system alerts.

Modify the mappings to correspond to your system, and save them as **auditmap.php** and **systemalertsmap.php**.

- **auditmap.php** enables FileCloud to convert audit entries to valid SIEM messages.
- **systemalertsmap.php** enables FileCloud to convert FileCloud's system alerts to valid SIEM messages.

> Mappings are stored in the .php file, so they have to follow all PHP syntax rules as well as internal mappings rules and syntax. To validate all mappings, navigate to **Settings > Third Party Integrations > SIEM** and click on **Validate mappings**.

SIEM mapping format

A sample SIEM mapping is a PHP array entry, which itself is an array. It contains the following fields:

id (required) - identifies the SystemAlert / Audit entry this map refers to.
Note that it can be a string literal that matches the audit operation name or one of the SiemArea values available in FileCloud, an array of values, or a wildcard '' that specifies that the mapping is applied to all audit entries/system alerts.*

prefilter (optional) - A collection of preconditions that an event has to meet in order to be processed and sent to the SIEM system. It is an array of filters, where each filter has the following format: property => value

where:

- property is a valid property available for the Audit/System Alert record
- value is a value that has to be matched in order to process the Audit / System Alert record, i.e.

Sample System Alert Mappings

```
    'prefilter' => [
        'level' => SysAlert::SYSALERT_LEVEL_MELTDOWN
    ],
```

specifies that only System Alerts with the Meltdown criticality level would be sent to the SIEM server.

map (Required) - specifies the actual mapping between the FileCloud object being processed and the SIEM-formatted message that will be sent to the SIEM server. SIEM object to contain the following four fields:

- eventClass - class of the event in the SIEM system.
- eventName - The name of the event.
- severity - this is a SIEM side severity, which is a number from the 1-10 range.
- extension - a collection (array) of additional key-value pairs that will be stored in the SIEM system (i.e. the user that performed the action, IP address of the request, etc.). The key can be any arbitrary string.

To resolve mappings, provide values in any of the following ways:

- As a literal value (string or number)

  Sample System Alert Mappings

  ```
  'eventClass' => 'authentication',
  'eventName' => 'invalid login',
  'severity' => 3
  ```

- As a property binding that resolves the value with the actual value provided by the FileCloud audit system alert being processed:

  Sample System Alert Mappings

  ```
  'eventClass' => '$siemArea',
  'eventName' => '$description',
  'user' => '$username',
  'filename' => '$request.filename', //Access a field in the request object/array
  'filePath' => '$realpath > $request.path > $notes' //The filePath will be resolved
  to the first non-empty value
  'ip' => '$ip'
  ```

  Properties should appear on the right-hand side of the arrow operator (=>). The property name must be prefixed with a dollar sign ($). Properties can take one of the following values:

  - A standalone value - '$property'
  - An array of values of an object with properties. The following syntax can be used to access any of the values: '$array.field' or '$object.field', for example, '$request.filename'. This can be applied recursively if the internal field is also an array or object, for example, '$response.meta.type'.
  - As a chain of fallback properties ('$property1 > $property2.field > $property3') - the value is resolved to the first non-empty property value. For example, the following syntax is resolved to filename if present or to the $request.fname otherwise: 'fname' => '$filename > $request.fname'. This allows the admin to provide more generic rules.

- As a method call:

  Sample System Alert Mappings

  ```
  'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'], ['$level']],
  ```

  NOTE: Users can create and use their own methods here. The first parameter is the PHP callback (class, method name) and the second parameter is the array of values (optional) that is processed by that callback. Parameters can be set to literal values or runtime-resolvable properties as described earlier. In FileCloud 19.2 *getSysAlertSeverity* is the only method available out of the box. It assigns internal System Alerts a severity of 1-10 as required by SIEM integration in the following way:

  - Meltdown: 10
  - Critical: 7
  - Warning: 4
  - Information: 1

Shared properties

Properties listed below can be used in both System Alerts and Audit mappings.

Property	Description	Values
who	Author of the operation	Name of the user or process that has triggered the operation
ip	IP Address	A regular IPv4 address
ts	Operation timestamp	Timestamp

Audit mappings

Audit stores information about actions being performed within the system. Currently, audit stores information about 200+ unique operations being performed within FileCloud. Each Audit record contains some generic information, shared with the System Alerts properties (see Shared Properties, above), common for each audit entry, and some unique properties, stored only for a group of actions.

Shared Audit Properties

Property	Description	Values
request	Request payload	The full request payload provided as a collection of key-value pairs that can be extracted in the mapping. Each operation carries a unique request. The request can be mapped as a full object, and its info will be sent to the SIEM server as a string. For example: `'request' => '$request'`, will be sent as `{"op":"loginguest","userid":"john.doe","password":"xxx"}` Each field can also be sent individually if provided in the mapping: `'loggedUser' => '$request.userid'`, where `userid` is one of the parameters of the request.
response	Response payload	Similar to the request, the response provides a collection of key-value pairs that can be extracted in the mapping or sent as a string. Each operation has a different response, so it is better to use this for dedicated rules. NOTE: Responses are not stored in audit by default, and they have to be enabled in **Admin > Settings > Admin (Audit Settings section) > Audit Logging Level (FULL)**, This is not recommended for production as it may affect performance and usually is not needed for auditing.
notes	Context of the operation	This field provides the most important information about each operation. The content is unique for each operation.
userAgent	The User-Agent that triggered the operation	NOTE: Web browser is used as a generic user-agent for all web browsers.
userName	Name of the user that triggered the operation	
operation	Name of the operation that was triggered	
resultCode	Result of the operation	1 - the operation was performed successfully (for example, login attempt was successful, a file was deleted) 0 - operation failed (for example, login was not possible, a file was not deleted due to invalid permissions)
recordId	A MongoDB id of the audit entry	This is a MongoDB ObjectId

Property	Description	Values
hostname	A name of the host	The name of the current host. This allows SIEM to differentiate tenants.

Operation-specific Audit Properties

Property	Description	Values	Supported operations
auditArea	Provides information about the system area of the operation	Name of the system area	Currently only supported for operations from the following groups: • workflows • retention
serviceId	Additional information about the operation target	Carries additional information about the operations such as the name of the workflow or the id of the retention policy that was updated	Available only when the auditArea field is present
bandwidth	Information about the size of the file	File size in bytes	Available for the following operations: • upload (file upload operation) • downloadfile
realpath	File or folder realpath	FileCloud's original location of the file/folder, for example. /johndoe/document/internal/doc.txt	Available only for retention-related and dlp operations

Property	Description	Values	Supported operations
metadata	A list of non-empty, custom attributes assigned to the file or folder	Any non-empty attributes assigned by the Custom metadata sets as a result of the Smart Classification rule	The following operations are supported: • downloadfilemulti - Download multiple files • downloadfile - Download single file • getaudio - Play audio file • getvideo - Play video file • getfsslideimage - View image file • docconvert - Open/view file • quickshare - Quick share • addusertoshare - Add specific users to share • addgrouptoshare - Add specific groups to share • setallowpublicaccess - Make share public (after sharing only with certain users/ groups)
deviceInfo	Name of the client application	Name of the application, i.e. FileCloud Drive	Any operation that is performed by one of the client apps: Drive or Sync

Sample mappings

The following shows sample mappings for the most common operations:

```
/***************************************** Downloads
*****************************************/
// Download file
$mappings[] = [
    'id' => 'downloadfile',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'FileOperations',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'fname' => '$request.filename > $notes', // $notes is a fallback for
downloadfilemulti operation
```

```
            'filePath' => '$realpath > $request.filePath', // realpath is used for
downloadfilemulti
            'fsize' => '$bandwidth',
            'cs1' => '$metadata',
            'cs1Label' => 'Metadata assigned to the file'
        ]
    ]
];

/****************************************** Uploads
******************************************/
// Upload
$mappings[] = [
    'id' => 'upload',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'FileOperations',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'fname' => '$request.filename', // $notes can be used as well
            'filePath' => '$request.path',
            'fsize' => '$bandwidth'
        ]
    ]
];

/****************************************** Shares
******************************************/
// addusertoshare - Adding user to the existing share
$mappings[] = [
    'id' => 'addusertoshare',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Shares',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'filePath' => '$notes',
            'duser' => '$request.userid',
            'cs1' => '$metadata',
            'cs1Label' => 'Metadata assigned to the file'
        ]
    ]
];

// updateshare - updating existing share
```

```
$mappings[] = [
    'id' => 'updateshare',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Shares',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'filePath' => '$request.sharelocation',
            'cs1' => '$metadata',
            'cs1Label' => 'Metadata assigned to the file'
        ]
    ]
];

// setuseraccessforshare - sets user permissions for share
$mappings[] = [
    'id' => 'setuseraccessforshare',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Shares',
        'eventName' => '$operation',
        'severity' => 6, // this can be a potentially risky operation since data
exposure and leakage might happen
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'filePath' => '$notes',
            'duser' => '$request.userid',
            'cs1' => '$metadata',
            'cs1Label' => 'Metadata assigned to the file',
            'cs2' => '$request.shareid',
            'cs2Label' => 'Share Identifier'
        ]
    ]
];

// setallowpublicaccess - happens when a share is mad public
$mappings[] = [
    'id' => 'setallowpublicaccess',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Shares',
        'eventName' => '$operation',
        'severity' => 6, // this can be a potentially risky operation since data
exposure and leakage might happen
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
```

```
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'filePath' => '$notes',
            'ispublic' => '$request.allowpublicaccess', // 1 - public share, 0 -
private share
            'cs1' => '$metadata',
            'cs1Label' => 'Metadata assigned to the file',
            'cs2' => '$request.shareid',
            'cs2Label' => 'Share Identifier'
        ]
    ]
];

/***************************************** Smart DLP
*****************************************/
// DLP Violation
$mappings[] = [
    'id' => 'dlp',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'DLP Violation',
        'eventName' => '$operation',
        'severity' => 6,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'filePath' => '$realpath',
            'msg' => '$notes.message',
            'shareTargetEmail' => '$notes.shareTargetEmail',
            'cs1' => '$metadata',
            'cs1Label' => 'Metadata assigned to the file',
            'cs3' => '$request.op', // operation that triggered the violation /
$notes.action can be uses as well for a less granular info: DOWNLOAD / SHARE / LOGIN
            'cs3Label' => 'DLP Violation trigger',
            // Additional information can be grabbed from the request object
            'cs4' => '$notes.violatedRule', // DLP rule that was violated
            'cs4Label' => 'DLP Violation rule'
        ]
    ]
];

/************************************** Smart Classification
**************************************/
// Smart Classification - apply match action
$mappings[] = [
    'id' => 'ccsapplymatchaction',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'CCE match',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'suser' => '$userName',
```

```
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'msg' => '$notes',
            'filePath' => '$realpath',
            'cs5' => '$svcid',
            'cs5Label' => 'Content classification rule name'
        ]
    ]
];

/****************************************** Login
*****************************************/
//Failed login attempt
$mappings[] = [
    'id' => 'loginguest',
    'prefilter' => [
        //List of conditions that audit entry has to met in order to be processed (or
filtered out if excluded option is there)
        'resultCode' => '0', //incidents only
        'exclude' => false // optional 'include' is used by default
    ],
    'map' => [
        'eventClass' => 'login',
        'eventName' => 'Invalid login attempt',
        'severity' => 2,
        'extension' => [
            'user' => '$userName',
            'ip' => '$ip',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
        ]
    ]
];


//Failed SSO login attempt
$mappings[] = [
    'id' => 'samlsso',
    'prefilter' => [
        //List of conditions that audit entry has to met in order to be processed (or
filtered out if excluded option is there)
        'resultCode' => '0', //incidents only
        'exclude' => false // optional 'include' is used by default
    ],
    'map' => [
        'eventClass' => 'login',
        'eventName' => 'Invalid SSO login attempt',
        'severity' => 2,
        'extension' => [
            'user' => '$userName',
            'ip' => '$ip',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
        ]
    ]
```

```
];

//Successful SSO login attempt
$mappings[] = [
    'id' => 'samlsso',
    'prefilter' => [
        //List of conditions that audit entry has to met in order to be processed (or
filtered out if excluded option is there)
        'resultCode' => '1',
        'exclude' => false // optional 'include' is used by default
    ],
    'map' => [
        'eventClass' => 'login',
        'eventName' => 'Successfull SSO login attempt',
        'severity' => 2,
        'extension' => [
            'user' => '$userName',
            'ip' => '$ip',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
        ]
    ]
];

 /************************************* AV - Virus removed
************************************/
// When AV finds and removes the file containing a Virus (i.e. ICAP AV)
$mappings[] = [
    'id' => 'virusremoved',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'virusremoved',
        'eventName' => 'Virus Removed',
        'severity' => 8,
        'extension' => [
            'user' => '$userName',
            'userAgent' => '$userAgent',
            'ip' => '$ip',
            'fname' => '$request.filename',
            'filePath' => '$request.path',
            'notes' => '$notes'
        ]
    ]
];

/****************************** Group management
****************************************/

// Group rename
$mappings[] = [
    'id' => 'updategroup',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Groups',
        'eventName' => '$operation',
        'severity' => 6,
        'extension' => [
```

```
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'msg' => '$notes'
        ]
    ]
];

$mappings[] = [
    'id' => 'addmembertogroup',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Groups',
        'eventName' => '$operation',
        'severity' => 5,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'duser' => '$request.userid',
            'msg' => '$notes'
        ]
    ]
];

$mappings[] = [
    'id' => 'deletememberfromgroup',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Groups',
        'eventName' => '$operation',
        'severity' => 5,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'duser' => '$request.userid',
            'msg' => '$notes'
        ]
    ]
];

/******************************** User management
*****************************************/

$mappings[] = [
    'id' => 'adduser',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Users',
        'eventName' => '$operation',
```

```
        'severity' => 5,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'duser' => '$request.username', // name of the user that has been added
            'msg' => '$notes' // More info about the user
        ]
    ]
];

// Admin status change
$mappings[] = [
    'id' => 'setadminstatus',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Users',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'duser' => '$request.profile',
            'msg' => '$request.adminstatus'
        ]
    ]
];

// User password changed by admin
$mappings[] = [
    'id' => 'setuserpassword',
    'prefilter' => [],
    'map' => [
        'eventClass' => 'Users',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'suser' => '$userName', // Admin who performed the operation
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'duser' => '$request.profile' // User whose password has been changed
        ]
    ]
];



/***************************************** Generic
****************************************/
// A generic map for all events
```

```
$mappings[] = [
    'id' => '*',
    'prefilter' => [],
    'map' => [
        'eventClass' => '$operation',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'suser' => '$userName',
            'shost' => '$hostname', // name of the host
            'recordId' => '$recordId', // Audit record id
            'requestClientApplication' => '$userAgent',
            'src' => '$ip',
            'msg' => '$notes',
            'fname' => '$request.filename',
            'filePath' => '$realpath > $request.path > $request.filepath',
            'duser' => '$request.userid'
        ]
    ]
];
```

System Alert mappings

FileCloud allows admins to create mappings for System Alerts generated by the system due to unexpected or unwanted behaviors. System Alert mappings contain properties that can be sent to the SIEM server or logged in the syslog for further processing.

Supported properties

Property	Description	Values

siemArea	System area where the alert was raised	One of the following values:
		`SiemArea::INFECTED_FILE`
		`SiemArea::INVALID_FILE_TYPE`
		`SiemArea::AV_CHECK_FAILED`
		`SiemArea::UNHANDLED_EXCEPTION`
		`SiemArea::SYSTEM_ERROR`
		`SiemArea::DISK_SPACE_EXCEEDED`
		`SiemArea::INDEX_DB_FAILURE`
		`SiemArea::RMC_INVALID_POLICY`
		`SiemArea::SEND_EMAIL_FAILED`
		`SiemArea::BACKGROUNDING_FAILED`
		`SiemArea::METADATA_HEALTH_CHECK`
		`SiemArea::WORKFLOW`
		`SiemArea::ZIP_BACKUP_FAILURE`
		`SiemArea::SIEM_SERVER_CONNECTION`
		`SiemArea::DLP_SHARE_KILL`

level	System alert critical level	One of the following values:
		`SysAlert::SYSALERT_LEVEL_MELTDOWN`
		`SysAlert::SYSALERT_LEVEL_CRITICAL`
		`SysAlert::SYSALERT_LEVEL_WARNING`
		`SysAlert::SYSALERT_LEVEL_INFORMATION`

type	Type of system alert	One of the following values:
		`SysAlert::SYSALERT_TYPE_DLP_SHARE_KILL_FAILED`
		`SysAlert::SYSALERT_TYPE_DLP_SHARE_KILLED`
		`SysAlert::SYSALERT_TYPE_CODE_CONFIGURATION_ERRROR`
		`SysAlert::SYSALERT_TYPE_CODE_AV_FAILURE`
		`SysAlert::SYSALERT_TYPE_CODE_SIGNATURE_FAILURE`
		`SysAlert::SYSALERT_TYPE_CODE_EXCEPTION`
		`SysAlert::SYSALERT_TYPE_CODE_ERROR`
		`SysAlert::SYSALERT_TYPE_QUOTA_EXCEEDED`

description	Alert description	

Property	Description	Values
notes	Alert notes	
username	The user whose actions raised the alert	
alertContext	Additional information, related to the alert	Various contexts, depending on the Alert. For example: **file** - filename for the File version deletion operation **filePath** - file location for the Infected file **fileName** - file name for the Infected file

Sample mappings

Sample System Alert Mappings

```
//Report all meltdowns
$mappings[] = [
    'id' => '*', //Wildcard denotes all Alerts
    'prefilter' => [
        'level' => SysAlert::SYSALERT_LEVEL_MELTDOWN
    ],
    'map' => [
        'eventClass' => '$siemArea',
        'eventName' => '$description',
        'severity' => 10,
        'extension' => [
            'user' => '$username',
            'ip' => '$ip'
        ]
    ]
];

//AV system alert - infected file found
$mappings[] = [
    'id' => SiemArea::INFECTED_FILE,
    'map' => [
        'eventClass' => 'System Error',
        'eventName' => '$description',
        'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'],
['$level']],
        'extension' => [
            'user' => '$username',
            'ip' => '$ip',
            'path' => '$alertContext.filePath',
```

```
            'file' => '$alertContext.fileName'
        ]
    ]
];

//Type mismatch report
$mappings[] = [
    'id' => SiemArea::INVALID_FILE_TYPE,
    'map' => [
        'eventClass' => 'System Error',
        'eventName' => '$description',
        'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'],
['$level']],
        'extension' => [
            'user' => '$username',
            'ip' => '$ip',
            'path' => '$alertContext.file'
        ]
    ]
];
```

SIEM Integration with Splunk Enterprise

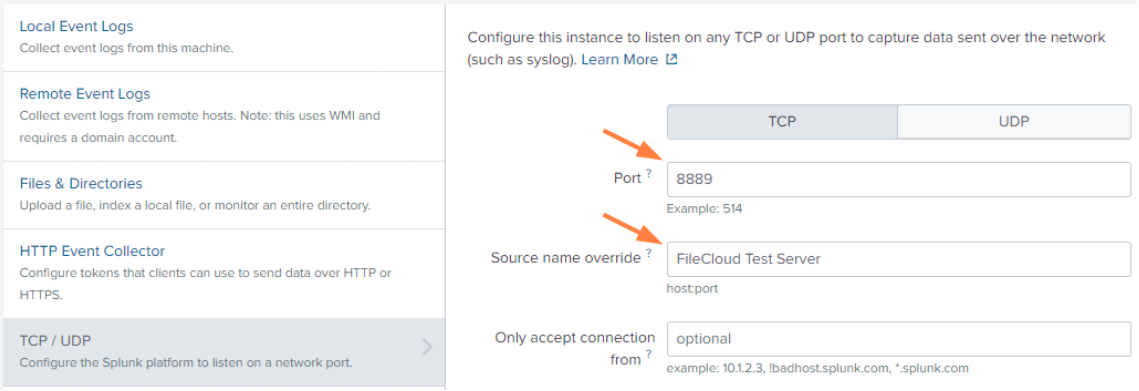
You can set up FileCloud's SIEM Integration feature with your Splunk server to receive audit logs and send event alerts to the administrator's email.

Splunk Server Configuration

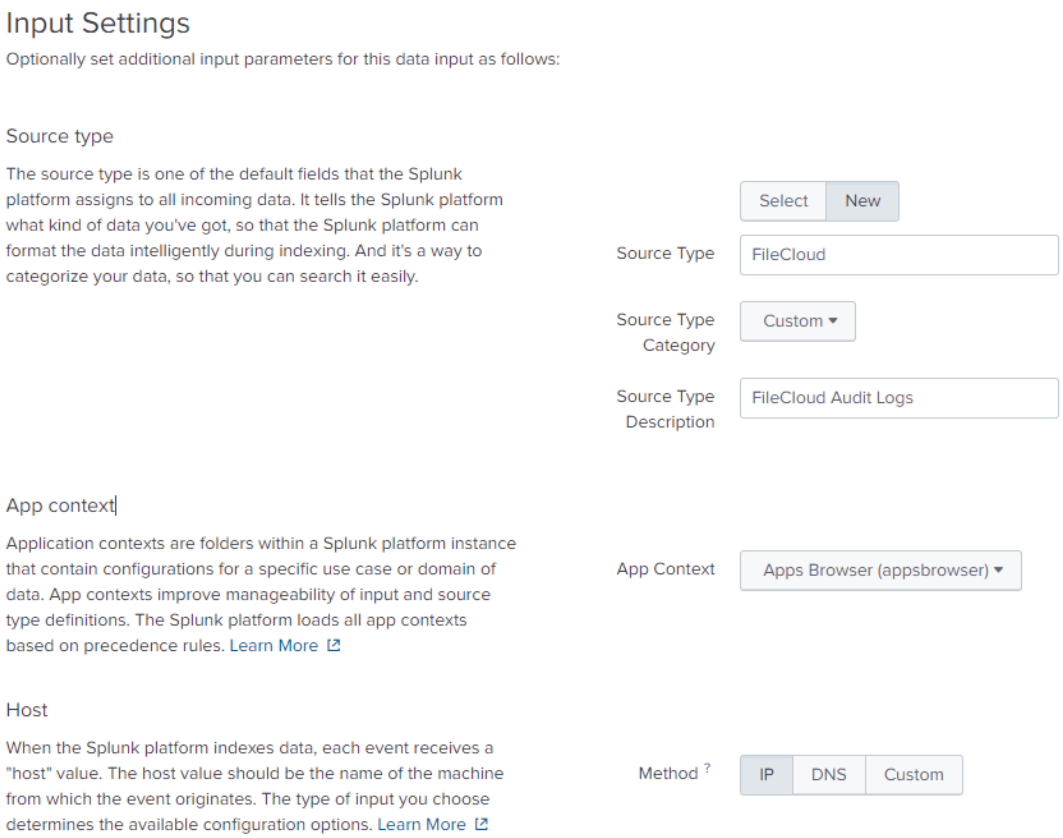
To configure Splunk server to receive data inputs from FileCloud through a designated TCP port and a specified source type, create a TCP Data Input entry that specifies the port that receives messages from the FileCloud and create a custom source type for FileCloud..

1. Log in to Splunk.
2. Click **Add Data**.
3. In the **TCP** row, click **Add new**.
   An **Add Data** wizard opens.

4. In the **Select Source** screen, in **Port**, enter the port that will receive messages from FileCloud. In **Source name override**, enter a name for the FileCloud server.



5. Go to the next screen.
6. In the **Input Settings** screen**,** enter the following settings:
   - Click **New**.
   - In **Source Type**,enter **FileCloud**.
   - In **Source Type Category**, choose **Custom**.
   - In **Source Type Description**, enter **FileCloud Audit Logs**.
   - In **App Context**, choose **Apps Browser (appsbrowser)**.
   - For **Host**, choose one of the following:
     - **IP** - Uses IP address of the host where the event originated.
     - **DNS** - Uses Doman Name Services (DNS) to convert the IP address to a host name that events are tagged with.
     - **Custom** - When you click this option, a **Host field value** field appears. This option uses the value you enter in **Host field value** to tag events.
   - Set **Index** to **Default**.



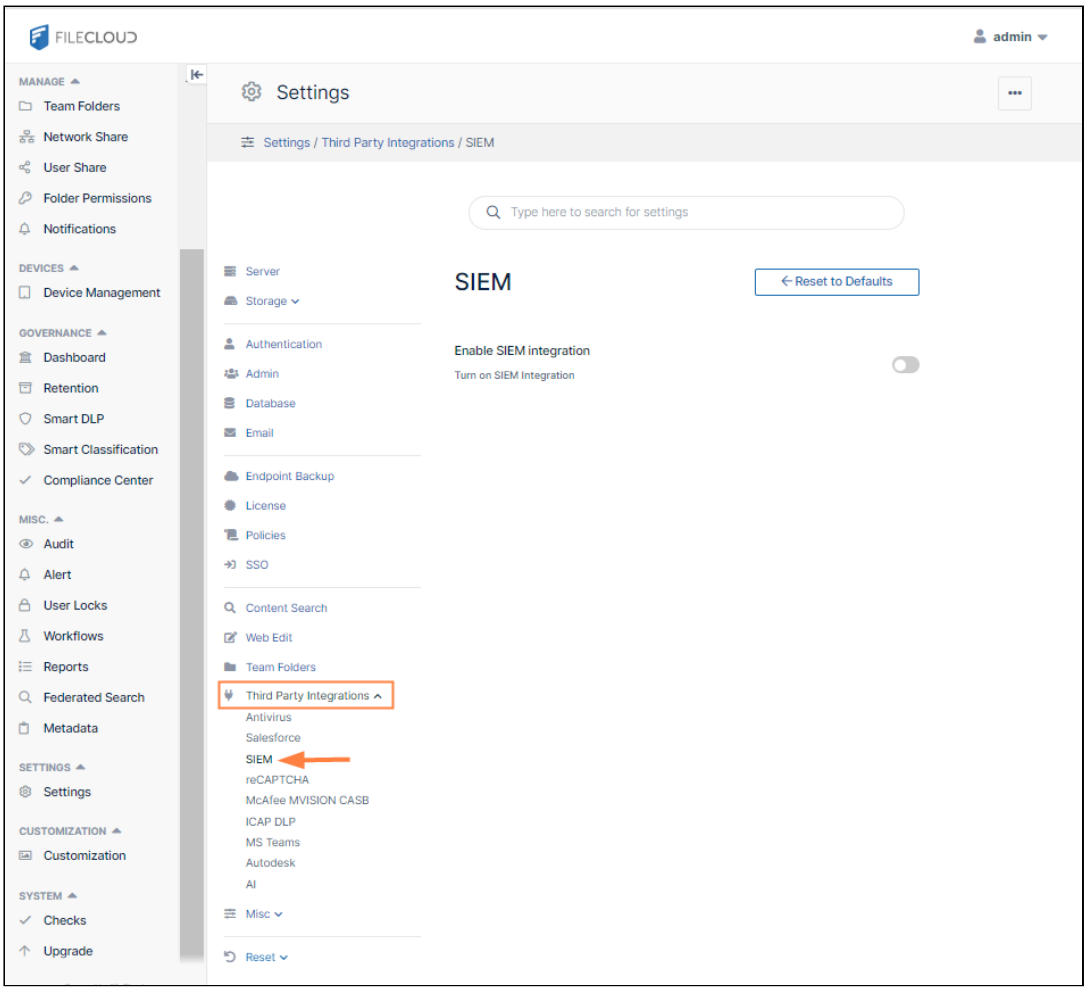
7. Go to the next screen in the wizard, **Review**, and check your settings.
8. Click next to complete your TCP Data Input entry configuration.

Setting up FileCloud to connect to the Splunk Server

Once the TCP Data Input entry is configured in Splunk, configure the SIEM Integration settings in FileCloud.

1. Go to the **SIEM** settings page.
   To go to the SIEM settings page

   a. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the **Settings** navigation page, click **Third Party Integrations** .
   b. In the inner navigation bar on the left of the **Third Party Integrations** page, expand the **Third Party Integrations** menu, and click **SIEM**, as shown below.

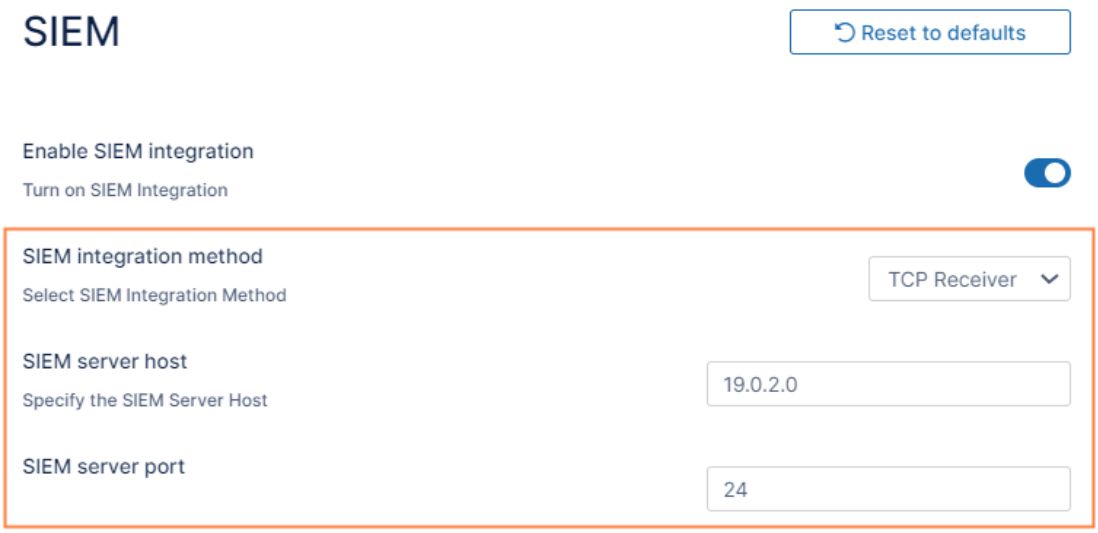


The **SIEM** settings page opens.

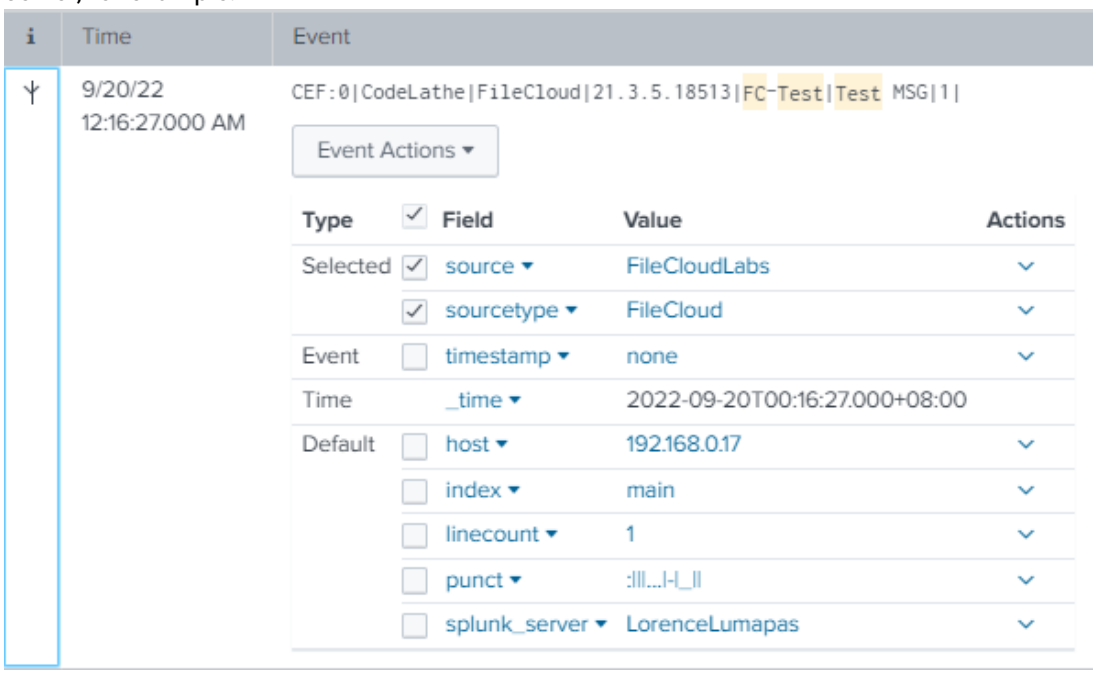
2. To activate SIEM integration, click the grayed out **Enable SIEM integration** button.

SIEM integration fields appear.

3. In **SIEM Integration Method**, choose **TCP Receiver**.
   In **SIEM Server Host**, enter the IP address or the hostname of the Splunk server.
   In **SIEM Server Port**, you may enter a unique port that is not currently used by the Splunk server for sending messages.
   For the other settings, see SIEM Integration.



4. Validate your configuration by clicking the **Test Connection**, **Send Test Message**, and **Validate Mappings** buttons. The **Send Test Message** button should send a test connection to the Splunk server, for example:

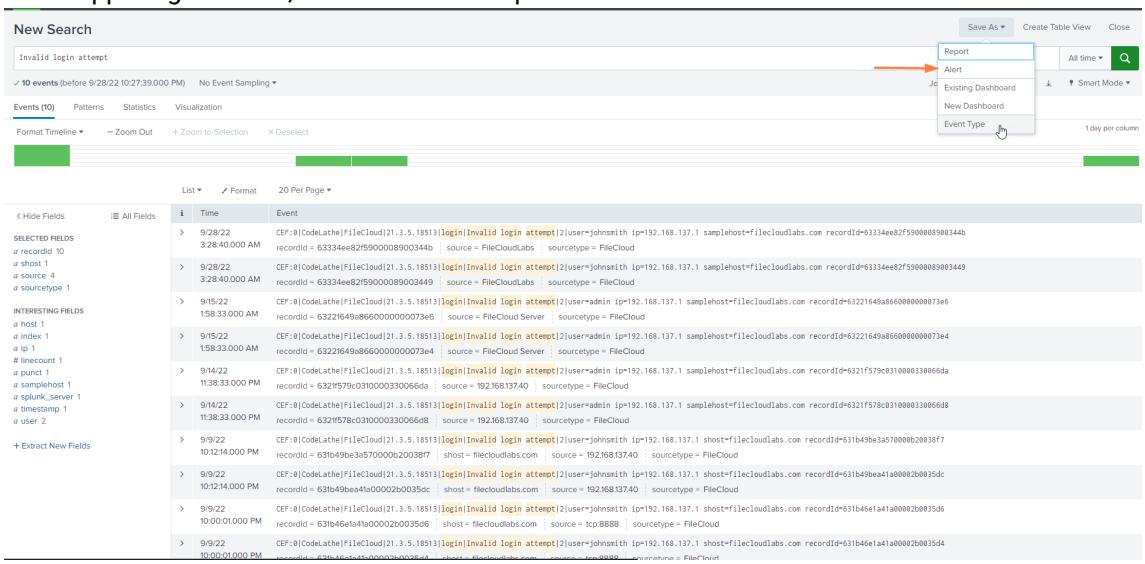


NOTE: Additional fields can be added by modifying the mappings from the **auditmap.php** and **systemalertsmap.php** files in FileCloud. See Managing SIEM Mappings for more information.

Setting up FileCloud event alerts in Splunk

1. Run a search for the event type from the Splunk Search screen and confirm that you get the expected data from the results.

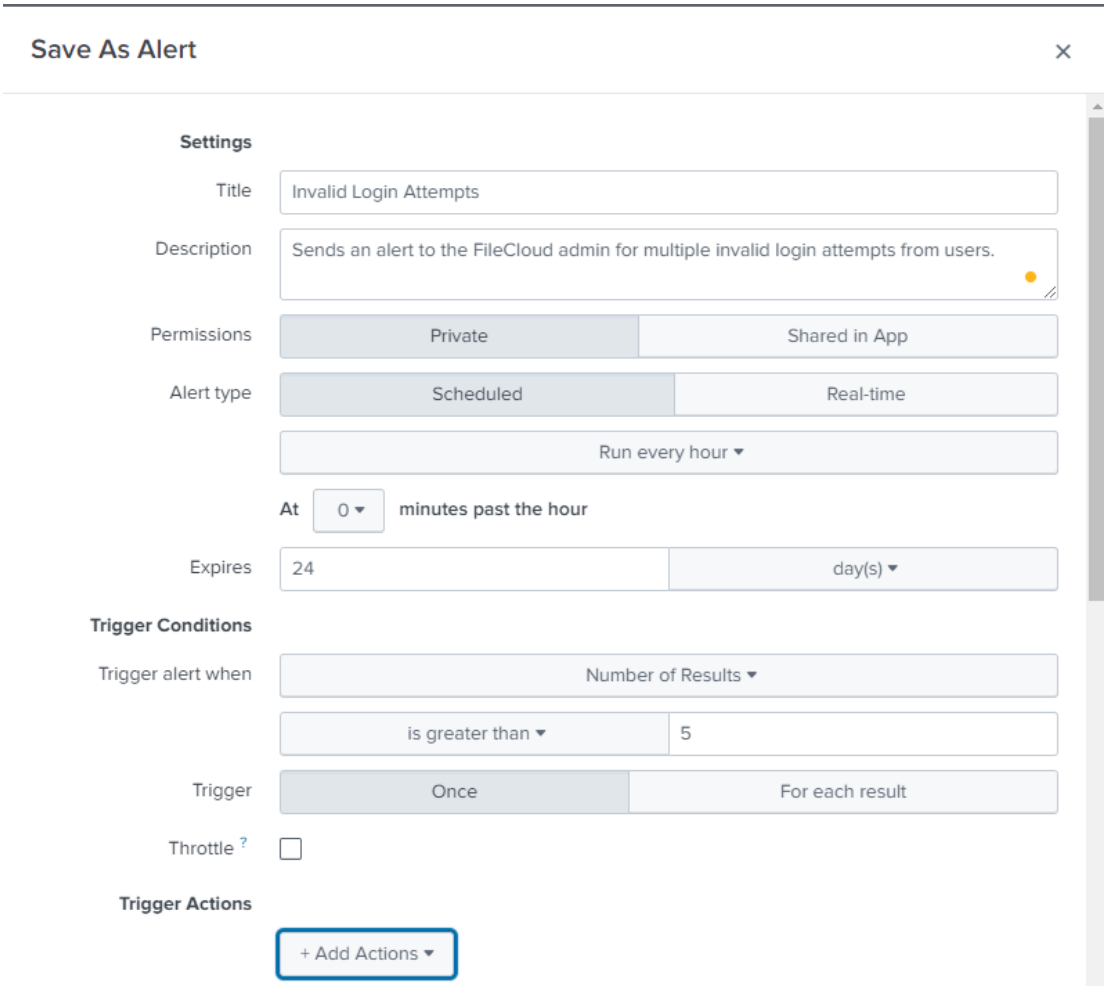
2. In the upper-right corner, in the **Save As** drop-down list choose **Alert**:

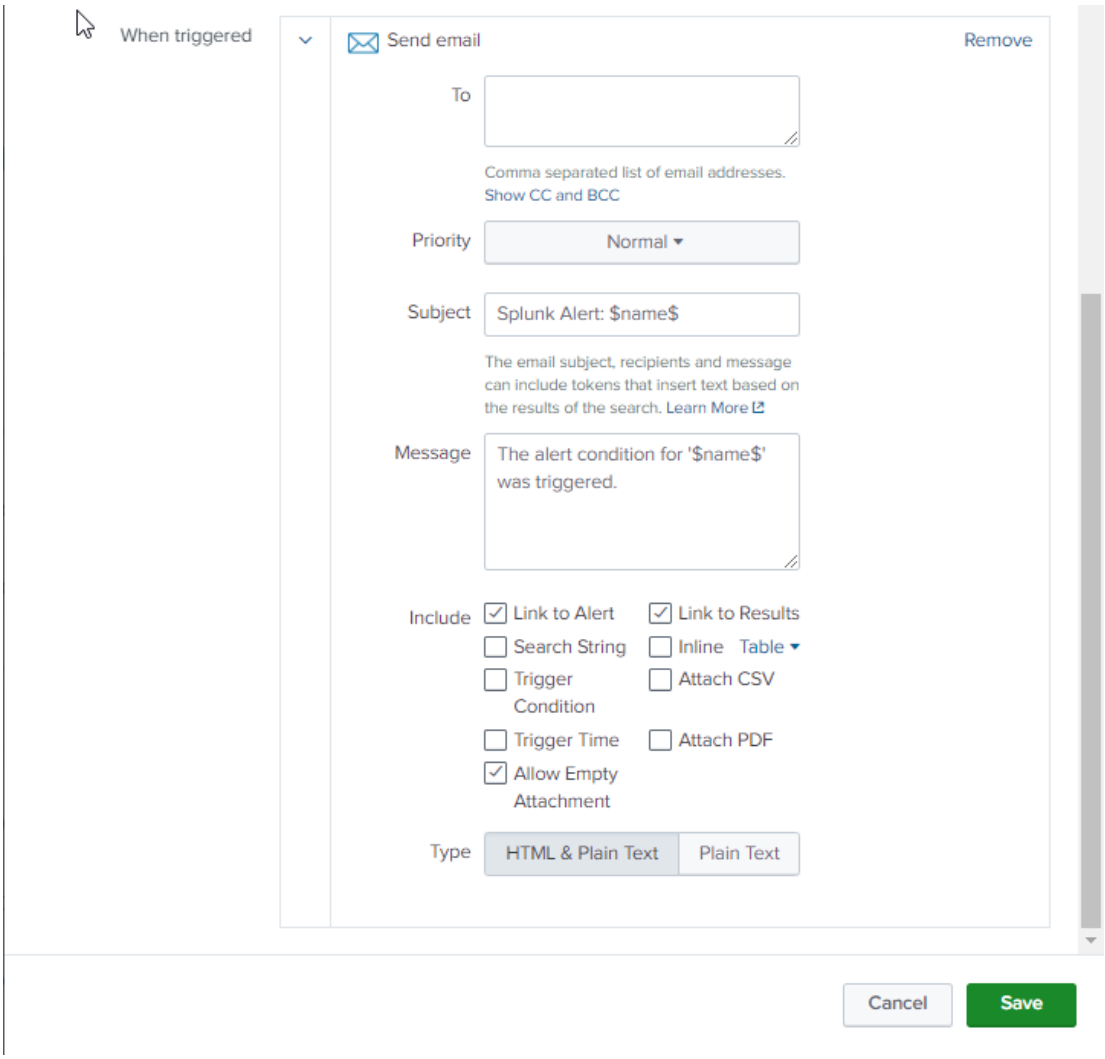


The **Save As Alert** dialog box opens.

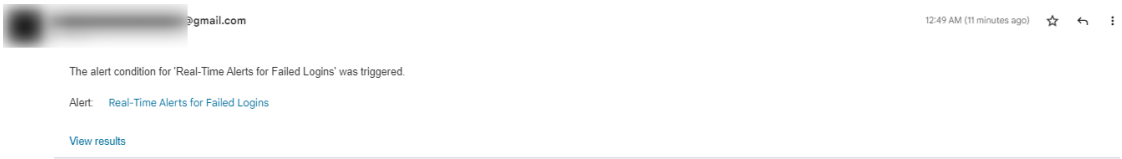
3. Fill in the fields. Enter the following fields as indicated:
   - **Alert Type -** Choose **Scheduled** to search for alert events on a schedule. Choose **Real-time** to trigger an alert when an alert event occurs.
     If you choose **Scheduled**, also choose a frequency in the drop-list below it.
   - **Trigger alert when** - Choose **Number of Results**, and enter a number.
   - In **Trigger Actions**, click **Add Actions**, and choose **Send email** as the action that is triggered by an alert.
   - In **To**, enter the recipient of the email.

4. Click **Save**.


Save As Alert
Settings
Title
Invalid Login Attempts
Description
Sends an alert to the FileCloud admin for multiple invalid login attempts from users.
Permissions
Private
Shared in App
Alert type
Scheduled
Real-time
Run every hour
At
0
minutes past the hour
Expires
24
day(s)
Trigger Conditions
Trigger alert when
Number of Results
is greater than
5
Trigger
Once
For each result
Throttle
Trigger Actions
+ Add Actions



5. Test to confirm that alerts are received by the mail in **To**, above. Below is an example of an email alert sent from Splunk.



reCaptcha Settings

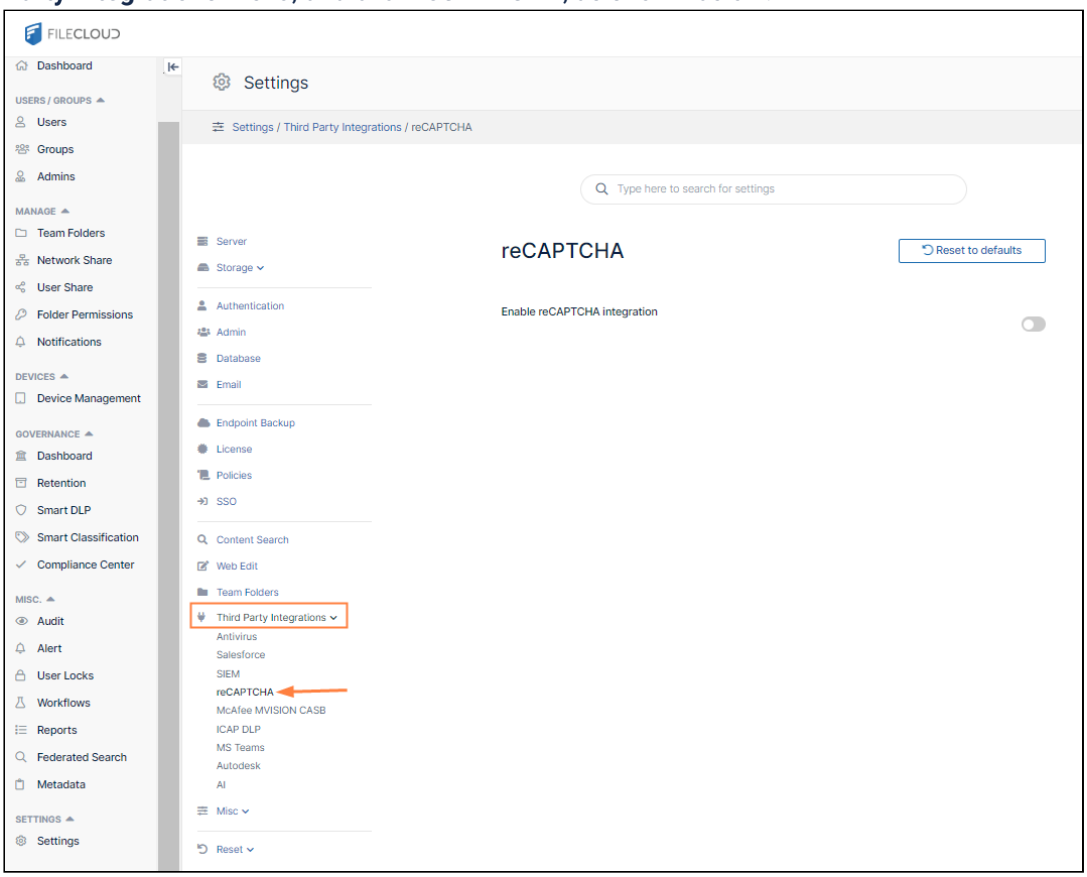
FileCloud supports reCaptcha v2. When you enable reCaptcha integration, reCaptcha is applied when users log in to FileCloud and when they access a password-protected file or folder share.

To configure reCaptcha:

1. Register your site at https://developers.google.com/recaptcha and get a key pair.
2. Open the ReCAPTCHA settings page.
   To go to the reCAPTCHA settings page

   a. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the **Settings** navigation page, click **Third Party Integrations** .

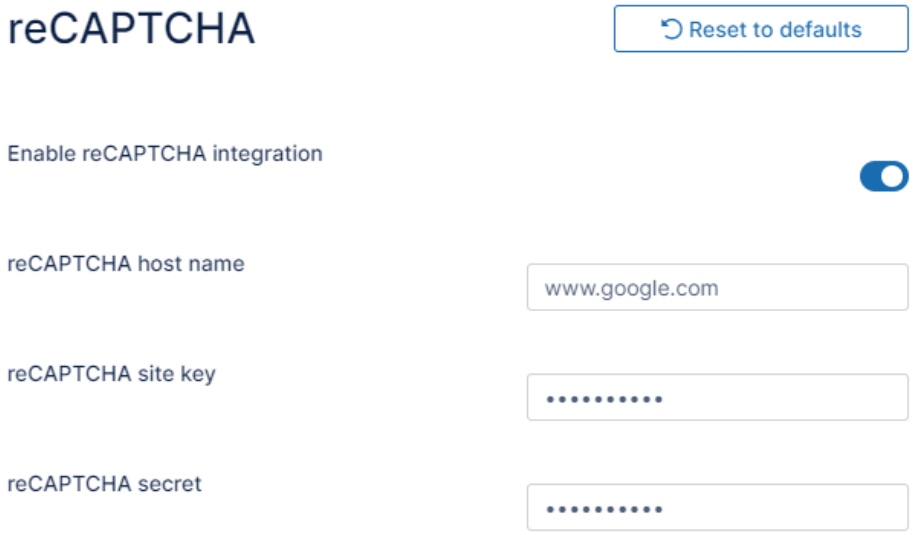
   b. In the inner navigation bar on the left of the **Third Party Integrations** page, expand the **Third Party Integrations** menu, and click **reCAPTCHA**, as shown below.



   The **reCAPTCHA** settings page opens.
   Open the **reCAPTCHA** settings page.

3. Enable the setting **Enable reCAPTCHA integration**.
   Additional reCAPTCHA settings appear.

4. If you plan to use a non-default reCAPTCHA site, enter the site hostname into **reCAPTCHA Host Name** in the format www.hostname.com.
   Note: If you are in a location that cannot access **www.google.com**, enter **www.recaptcha.net** (https://developers.google.com/recaptcha/docs/faq#can-i-use-recaptcha-globally)
5. Enter your key pair into **reCAPTCHA Site Key** and **reCAPTCHA Secret**.



6. Click **Save**.

SSO API: Configure Import of SSO Groups and Users

> Beginning in FileCloud 23.251, admins can import FileCloud groups and users from Okta, Google, and Azure SSO providers. In the future, importing groups and users from additional providers may be available.

Systems that authenticate users with Okta, Google, or Azure SSO can also import the users and their groups from the SSO provider. This requires integration of FileCloud and the SSO provider, separate from the configuration of the SSO provider(s) on the SSO settings pageor through the idpconfig file.

To set up the integration of the SSO provider and FileCloud for group and user import:

- Step 1: Set up FileCloud to integrate with the SSO provider for group/user import in the SSO provider's application.
- Step 2: Set up the SSO provider to integrate with FileCloud for group/user import in the FileCloud admin portal.
- Optional: If you want to allow users with identical email usernames but different email domains to be created as separate users, see To allow duplicate usernames for different domains, below.

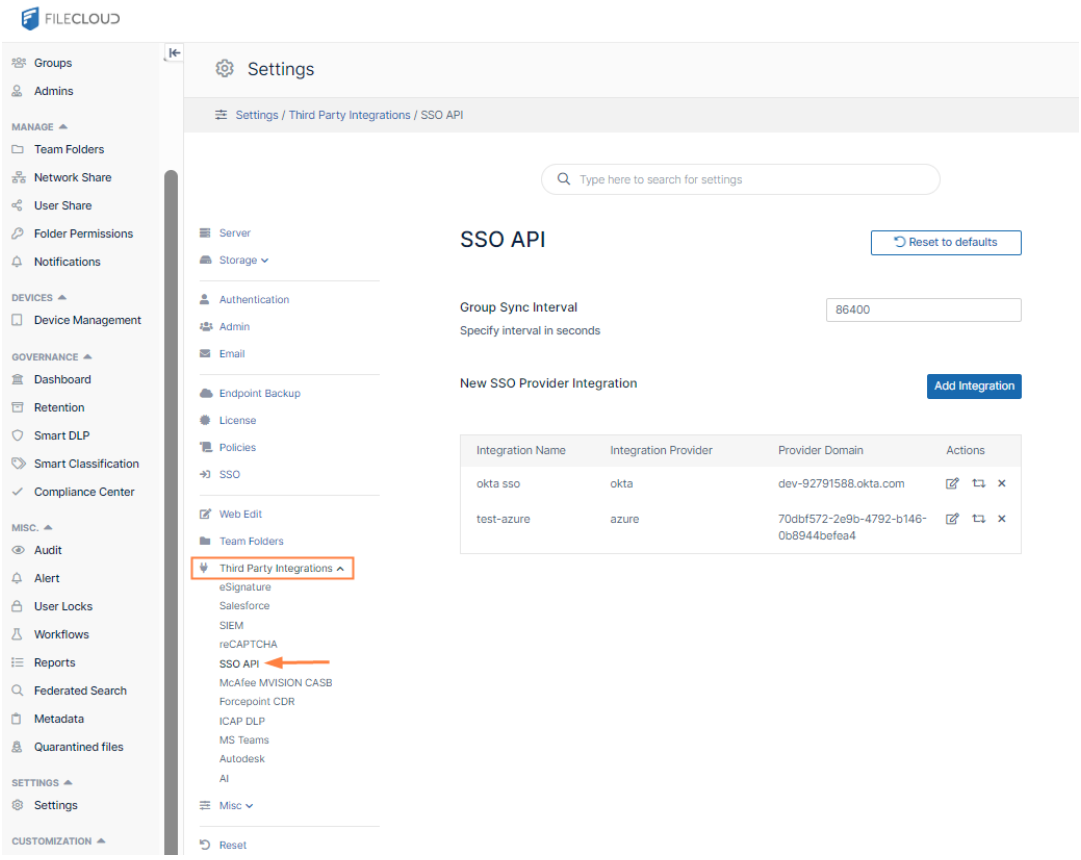
Step 1: Set up FileCloud to integrate with the SSO provider in the SSO provider's application:

Currently, the SSO providers available for integration with FileCloud for group/user import are Okta, Google, and Azure.

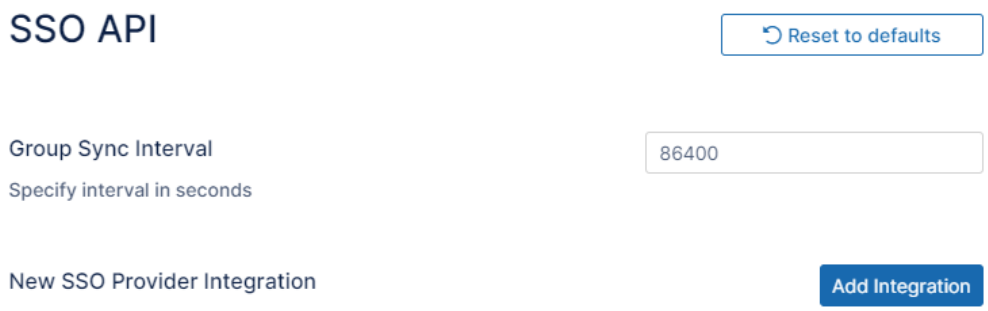
- Okta: Set Up FileCloud Integration for SSO Group/User Import
- Azure: Set up FileCloud Integration for SSO Group/User Import
- Google: Set up FileCloud Integration for SSO Group/User Import.

Step 2: Set up the SSO provider for importing groups and users into FileCloud:

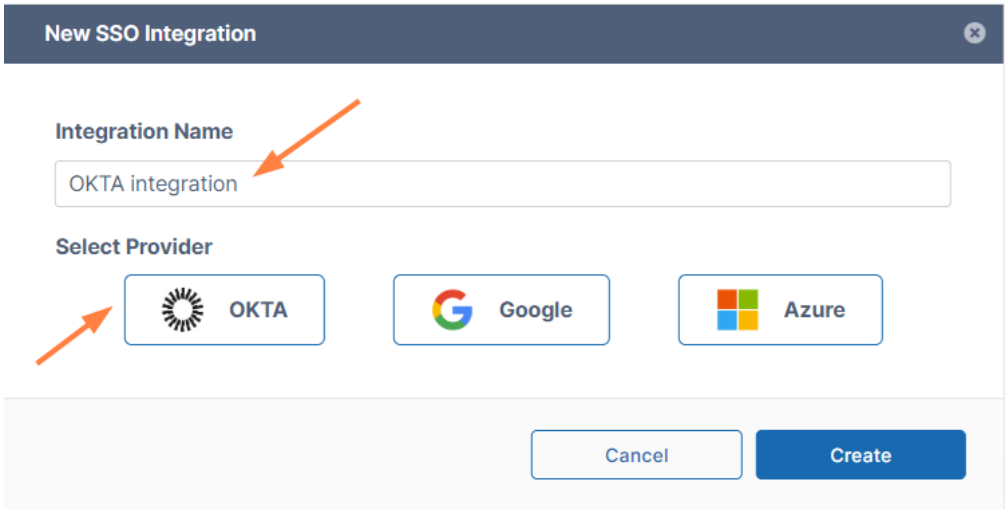
1. Open the **SSO API** page.
   a. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the **Settings** navigation page, click **Third Party Integrations** .
   b. In the inner navigation bar on the left of the **Third Party Integrations** page, expand the **Third Party Integrations** menu, and click **SSO API**, as shown below.



The SSO API settings page opens.



2. By default, the group sync is set to occur every 86400 seconds (once a day).
   To change how often group sync occurs, modify the value of **Group Sync Interval**. Specify the value in seconds.
3. Click **Add Integration**.
   The **New SSO Integration** dialog box opens.
4. Enter a name for the integration and click the button for the corresponding SSO provider:



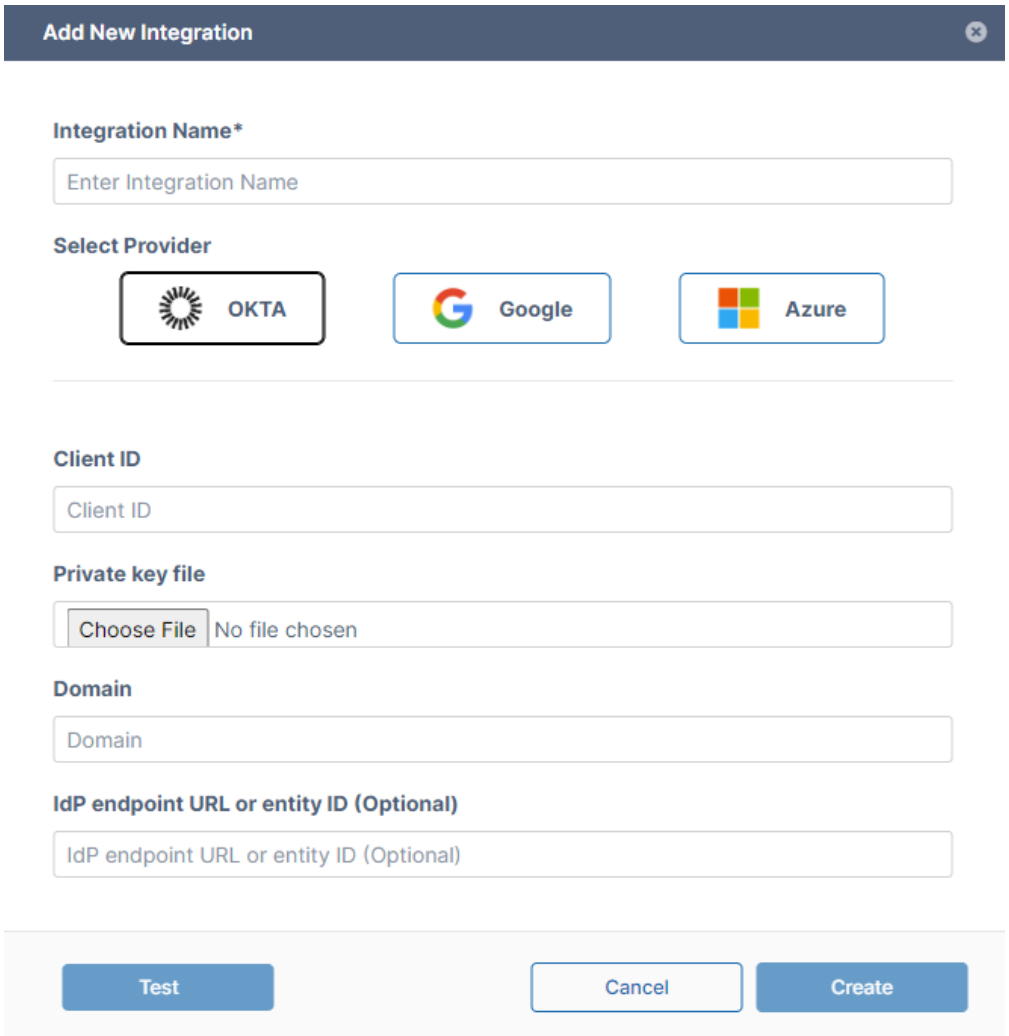
The dialog box expands.

Enter the integration values for the specific SSO provider:

OKTA

Enter integration values for Okta

1. When you click the **OKTA** button under **Select Provider**, the following settings appear. Enter the value for each as indicated in the table below.



Integration Name

You may enter any name.

Client ID

Enter the Client ID created for you when you set up the integration with FileCloud in Okta. You may copy it from the Okta Admin Console's listing for the application and paste it into the field. The following image shows where it appears in the Okta Admin Console.

Private key file

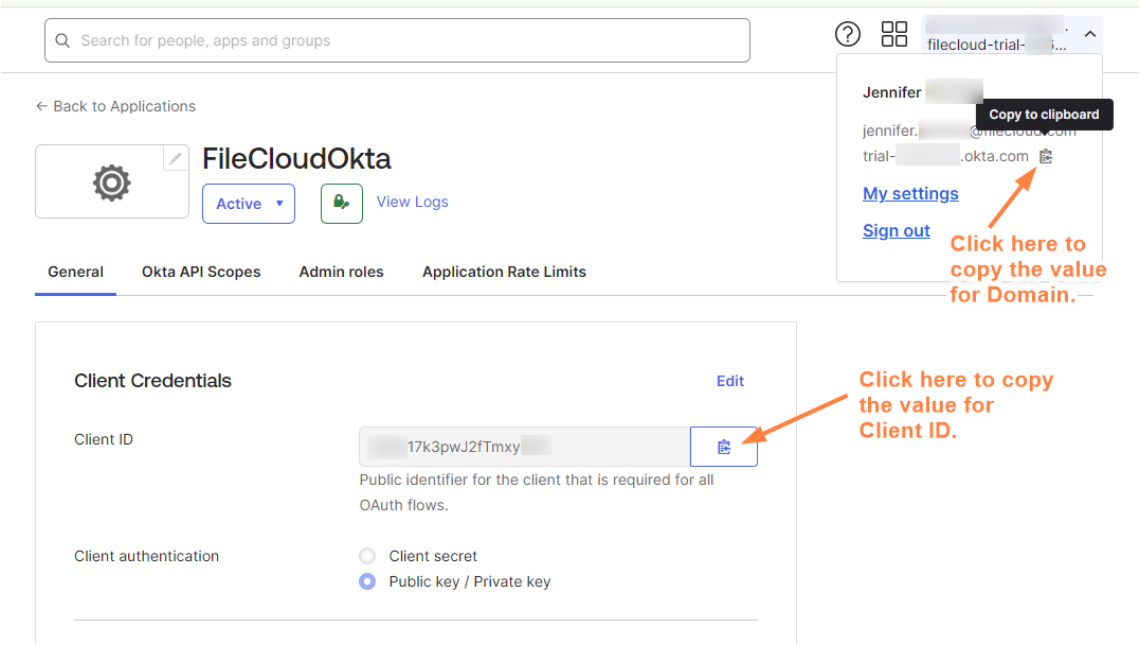
Choose the .pem file that you saved your private key in. You may have created the file and saved it when you were setting up the integration with FileCloud in Okta.

Domain

Enter the domain that Okta created for your user in Okta when you set up the integration with FileCloud in Okta. You may copy it from the Okta Admin Console's User drop-down box and paste it into the field. The following image shows where it appears in the Okta Admin Console.

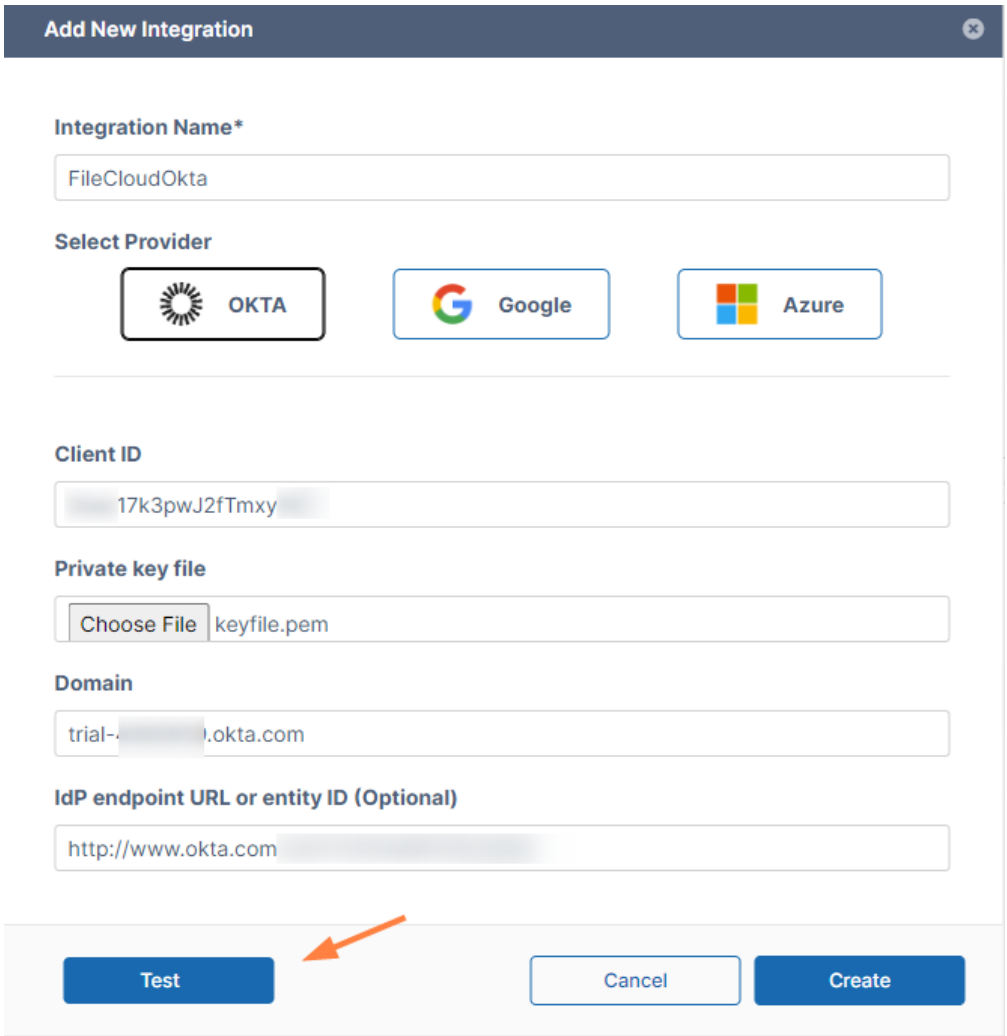
IdP endpoint URL or entity ID (Optional)

Enter if you are using multiple IdP's. Enter the value in the field **IdP endpoint URL or entity ID** from the FileCloud SSO settings.

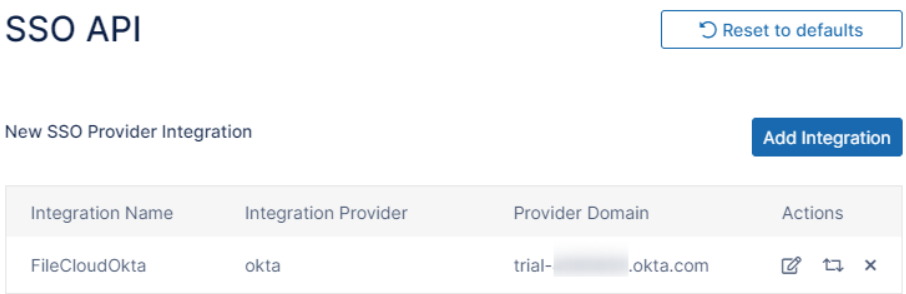


Location of values for FileCloud fields in Okta Admin Console

2. Once you have filled in the fields, click **Test** to make sure your integration works.



3. If the test is successful, click **Create**.
   The integration is added to the list of SSO integrations:



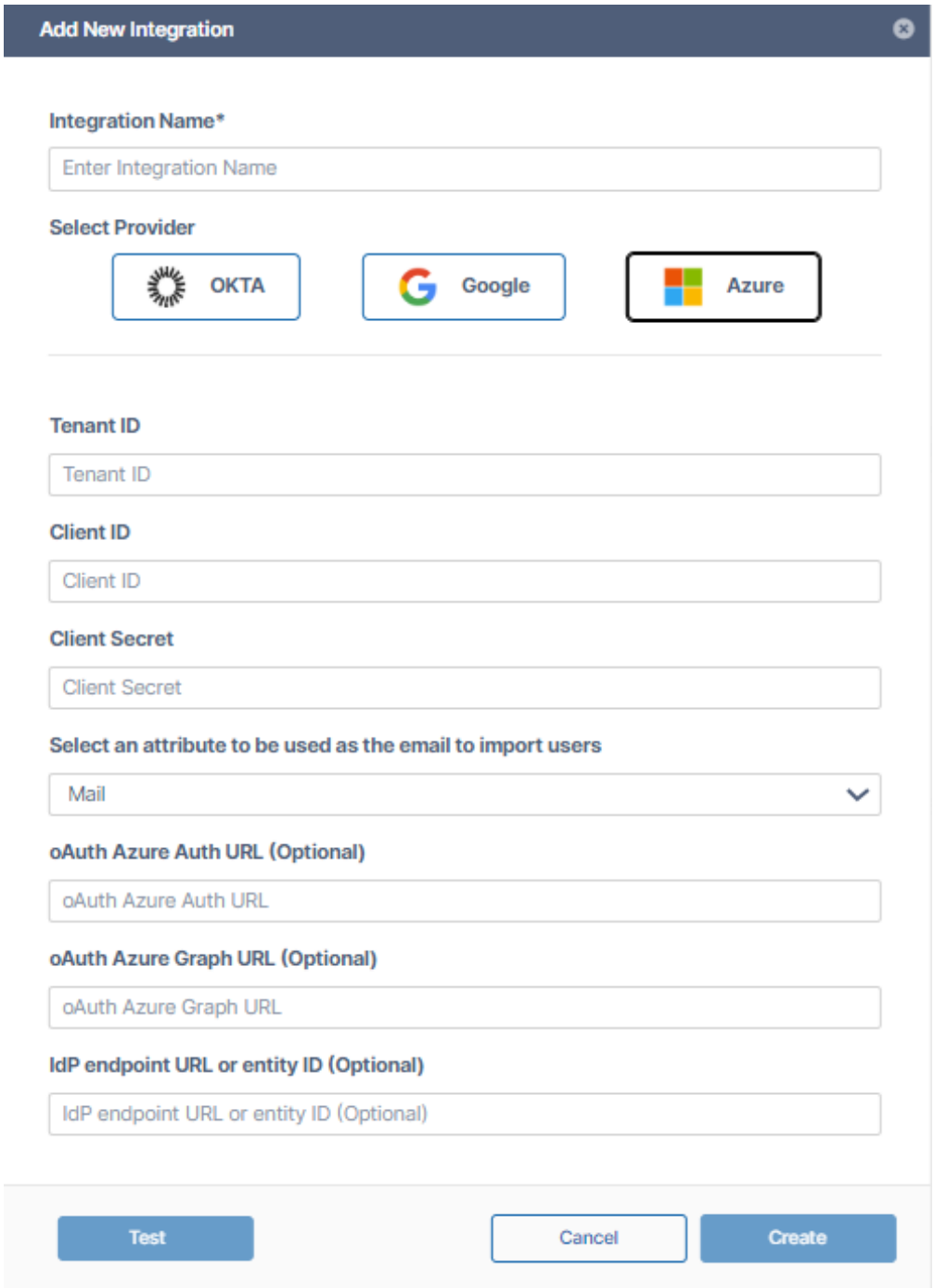
4. By default built-in OKTA groups are not listed when you import groups from OKTA.

   To list built-in OKTA groups, in cloudconfig.php, add the setting define('TONIDOCLOUD*ADMIN*SSO*API*LIST*ALL*GROUPS',1); OKTA groups:

Azure

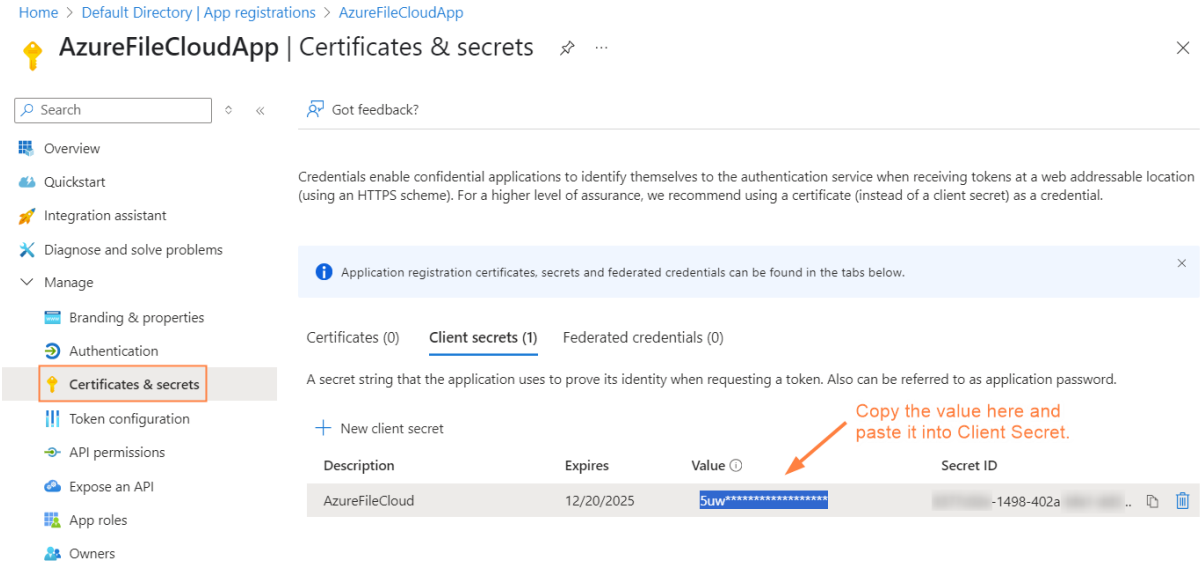
Enter integration values for Azure

1. When you click the **Azure** button under **Select Provider**, the following settings appear. Enter the value for each as indicated in the table below.

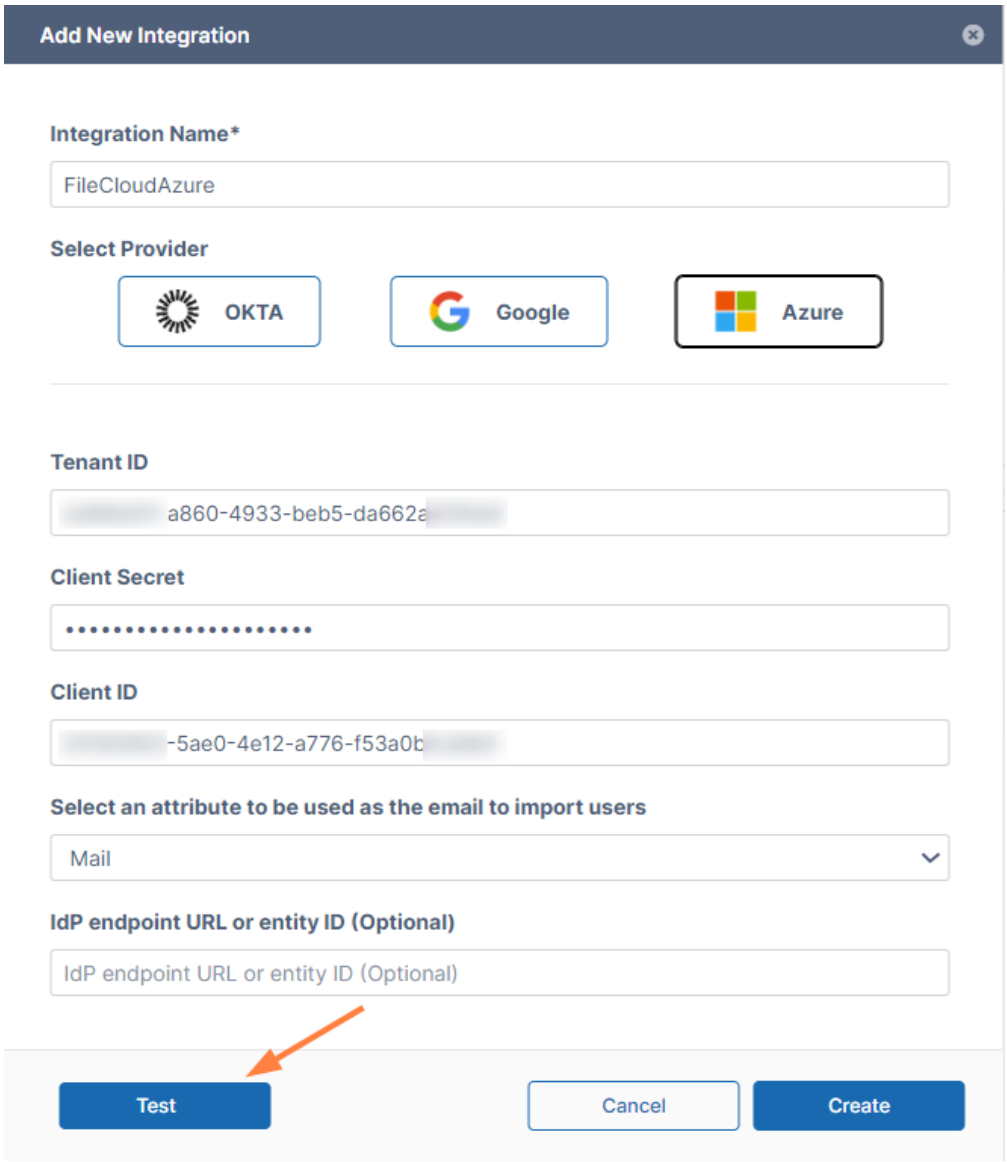


Integration Name	You may enter any name.

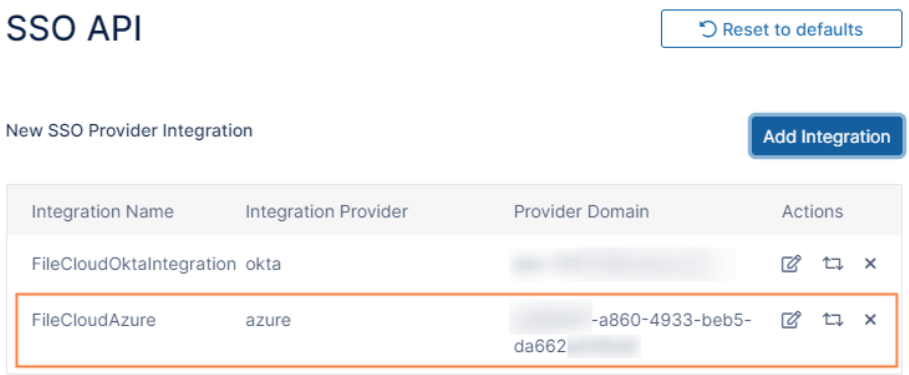
Tenant ID	Enter the **Directory (tenant) ID** that you saved from the Overview page when you set up your integration with FileCloud in Azure, or copy it directly from that page in the Azure portal and paste it into the **Tenant ID** field. The first of the images below shows where it appears in the Azure portal.
Client Secret	Enter the **Value** that you saved from the Certificates & secrets page when you set up your integration with FileCloud in Azure, or copy it directly from that page in the Azure portal and paste it into the **Client Secret** field. The second of the images below shows where it appears in the Azure portal.
Client ID	Enter the **Application (client) ID** that you saved from the Overview page when you set up your integration with FileCloud in Azure, or copy it directly from that page in the Azure portal and paste it into the **Client ID** field. The first of the images below shows where it appears in the Azure portal.
Select an attribute to be used as the email to import users	Select the attribute that is used to authenticate the user in SSO. Options are **Mail** or **userPrincipalName**.
oAuth Azure Auth URL (Optional)	In general, this is for use by Azure GovCloud users. Enter the URL of your Azure authorization domain.
oAuth Azure Graph URL (Optional)	In general, this is for use by Azure GovCloud users. Enter the URL of your Azure graph domain.
IdP endpoint URL or entity ID (Optional)	Enter if you are using multiple IdP's. Enter the value in the field **IdP endpoint URL or entity ID** from the FileCloud SSO settings.



2. Once you have filled in the fields, click **Test** to make sure your integration works.



3. If the test is successful, click **Create**.
   The integration is added to the list of SSO integrations:



4. By default built-in Azure groups are not listed when you import groups from Azure SSO.

For help listing built-in Azure groups, please Contact FileCloud Support.

To list built-in Azure groups:

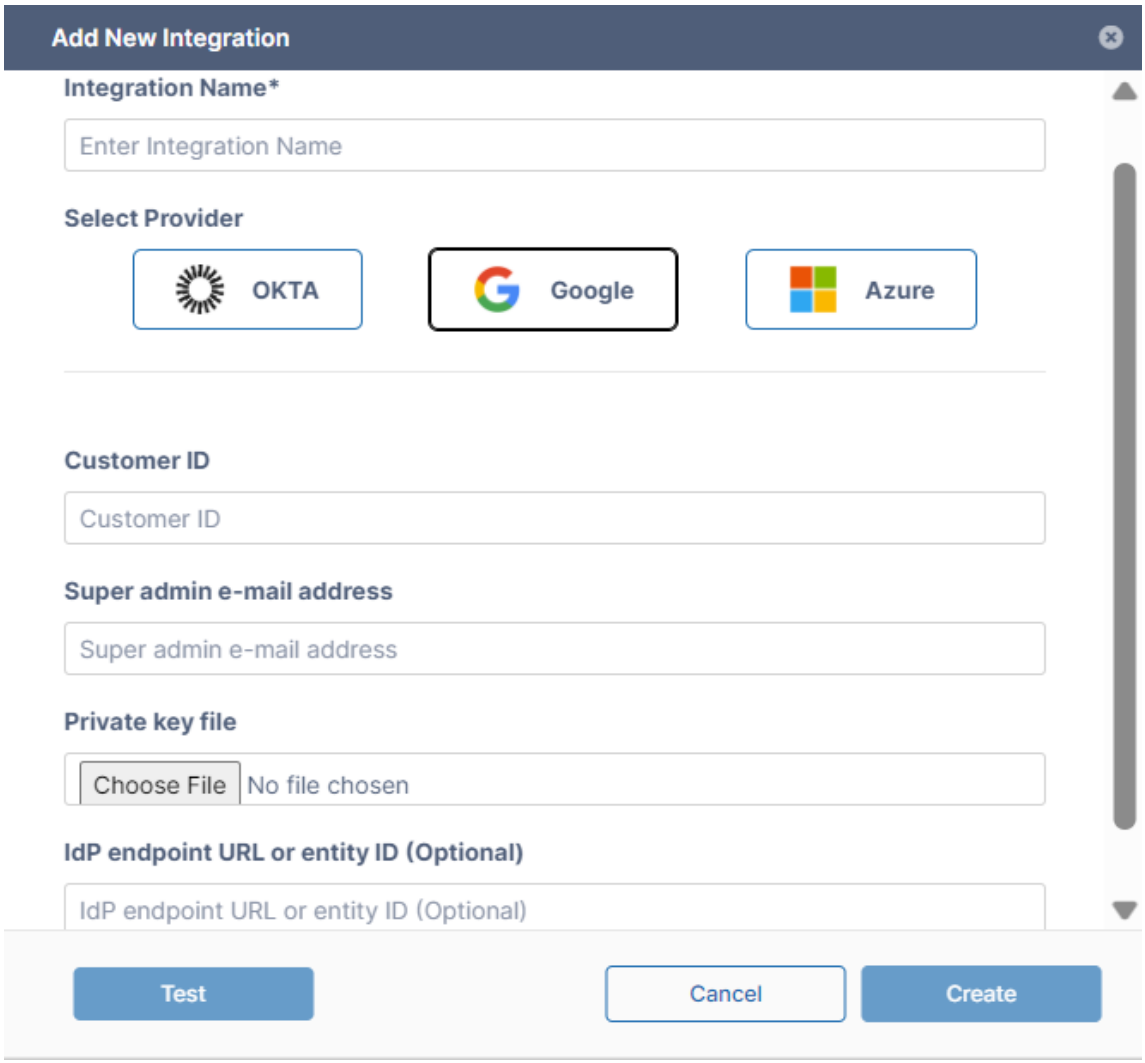
1. Open the configuration file at:
   Windows: **xampp/htdocs/config/cloudconfig.php**
   Linux: **/var/www/html/config/cloudconfig.php**
2. Add the setting:

```
define('TONIDOCLOUD_ADMIN_SSO_API_LIST_ALL_GROUPS',1);
```

Google

Enter integration values for Google

1. When you click the **Google** button under **Select Provider**, the following settings appear. Enter the value for each as indicated in the table below.



Integration Name

You may enter any name.

Customer ID

Find the value that you saved for EntityID in the Google admin portal and copy the value after **idpid=** at the end into **Customer ID**. For example, if the value you saved was: https://accounts.google.com/o/saml2?idpid=ABC123DEF, enter **ABC123DEF** into **Customer ID**. The image below shows where it appears in the Google admin portal.

Super admin e-mail address

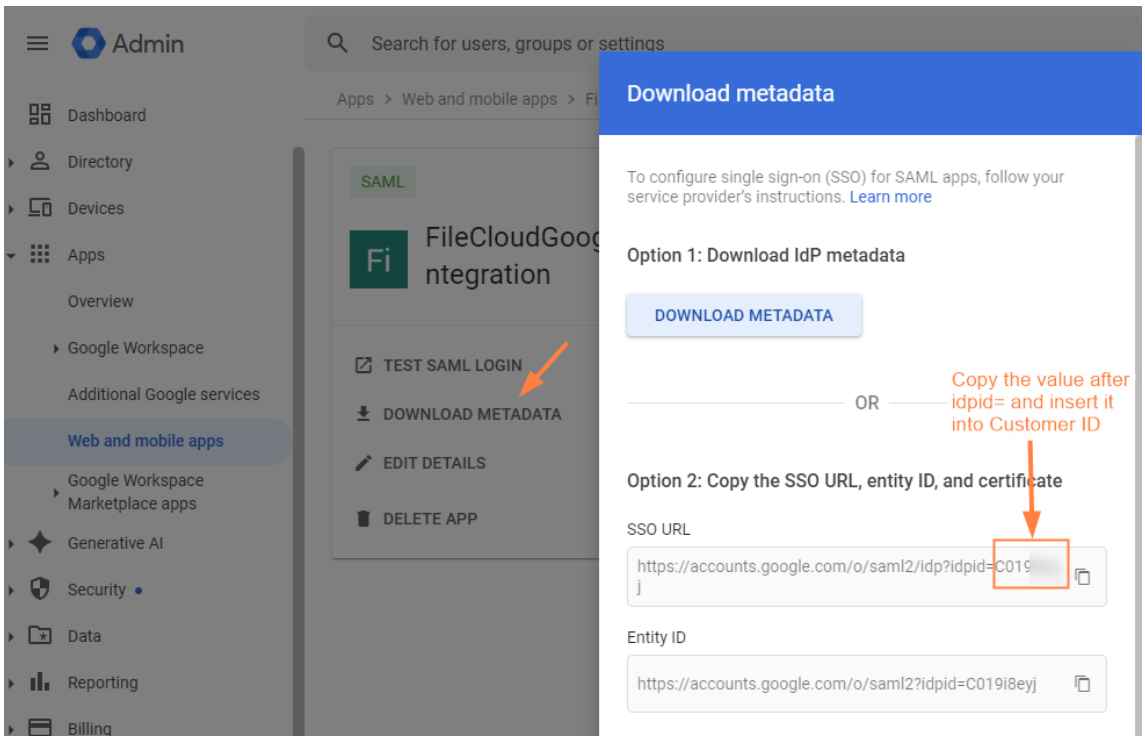
The e-mail address of the superadmin who added the integration of FileCloud and Google SSO in the Google admin portal and the Google Cloud Console.

Private key file

The json file that was created in the Google Cloud Console.

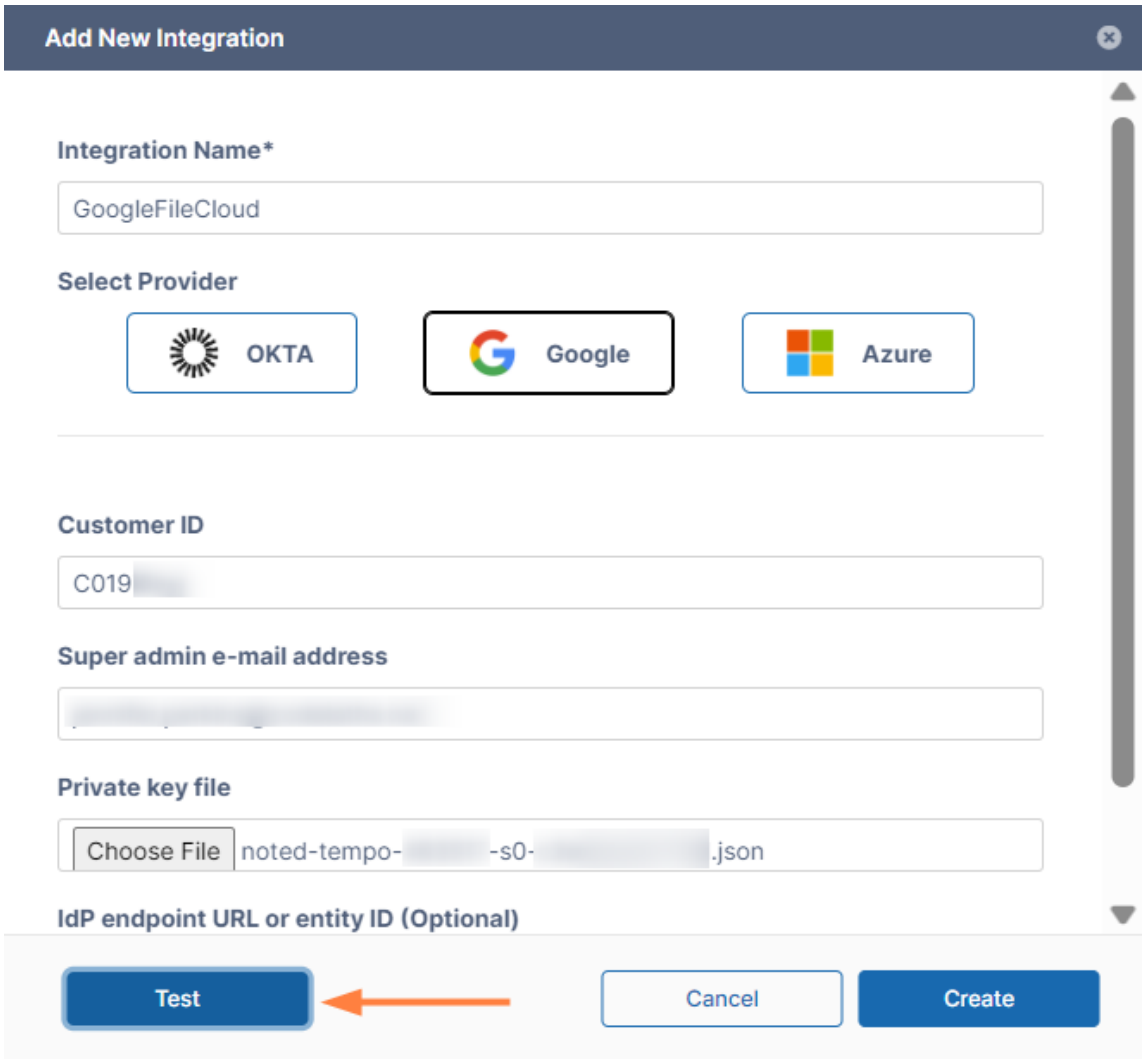
IdP endpoint URL or entity ID

If you are using multiple IdP's, enter the IdP endpoint URL or entity ID from the FileCloud SSO settings.

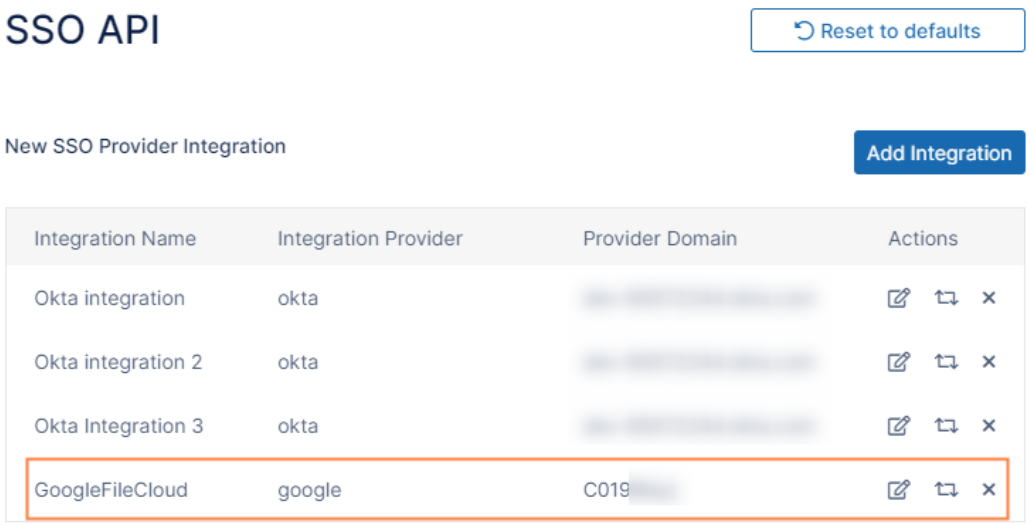


Location of the **Customer ID** value in the Google Admin Portal.

2. Once you have filled in the fields, click **Test** to make sure your integration works.



3. If the test is successful, click **Create**.
   The integration is added to the list of SSO integrations:



To allow duplicate usernames for different domains

If you are using SSO, SSO API, or Active Directory (AD) to import users, FileCloud uses the imported user's email username (the portion before @) as the FileCloud username. However, by default, FileCloud does not allow creation of users with the same usernames.

If any of your users have multiple email addresses, there may be a situation where the email username (the portion before @) is the same but the domains are different in their multiple emails. In this case, FileCloud will attempt to convert the second email username (and any further emails with the same username) to the FileCloud username, which will fail because the username already exists.

However, if the multiple email addresses actually represent different users, you may use the following setting to enable creation of multiple users:

1. Open cloudconfig.php at
   - Windows: xampp/htdocs/config/cloudconfig.php
   - Linux: /var/www/config/cloudconfig.php
2. Add the following and save:

   ```
   define("TONIDOCLOUD_ALLOW_DUPUSERNAMES",1);
   ```

   Now:
   If you create a a user with the username **jocelynp** and email **jocelynp@gmail.com**, FileCloud creates a user with the username **jocelynp**.

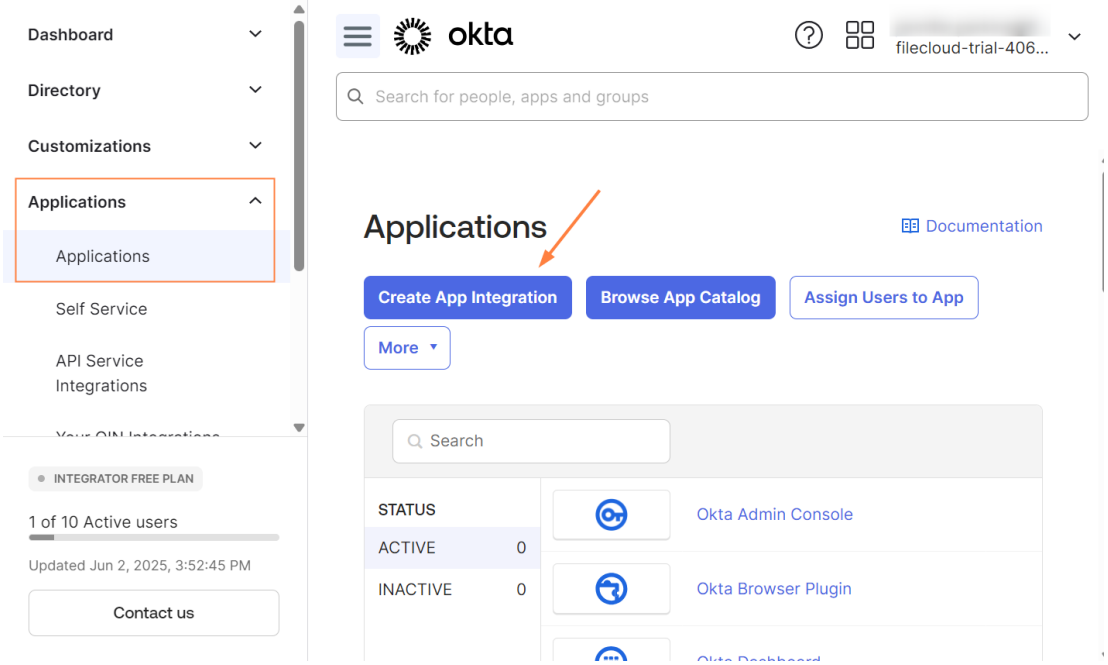
Next, if you attempt to create a user with the username **jocelynp** and email **jocelynp@optimum.net**, FileCloud creates a user with the username **jocelynp_optimum**. This follows for any subsequent emails with the same username and a unique domain.

Now import groups and users through your SSO integration on the Managed Groups page.

Okta: Set Up FileCloud Integration for SSO Group/User Import

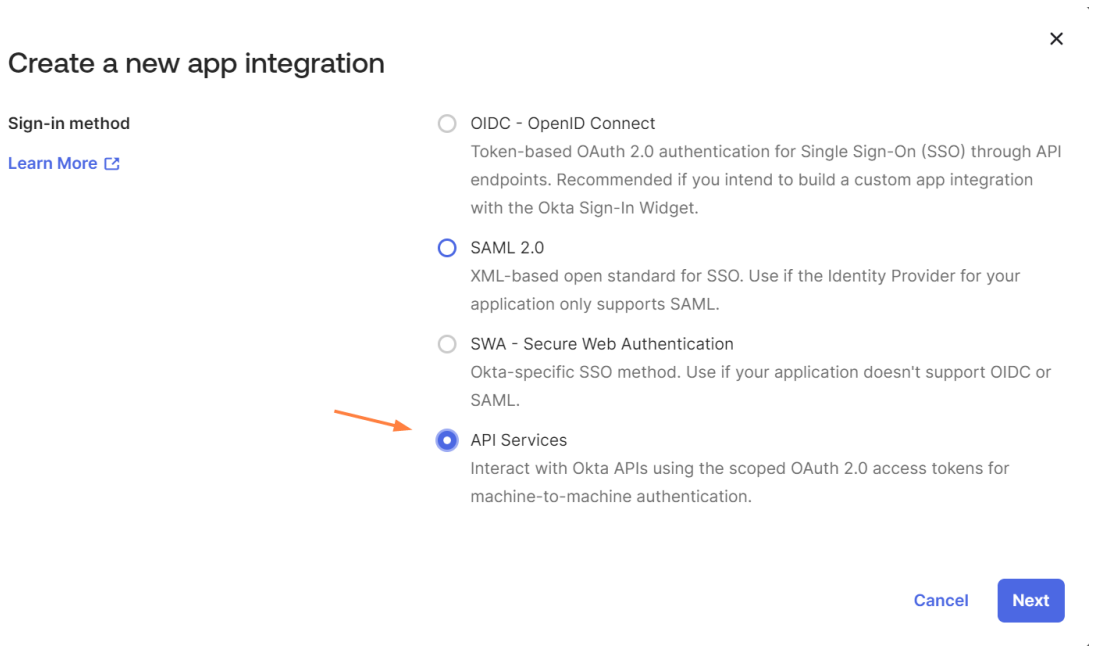
To configure FileCloud/Okta integration in Okta for SSO group/user import:

1. Log in to the Okta admin portal, and navigate to **Applications > Applications.**
2. Click **Create App Integration**.

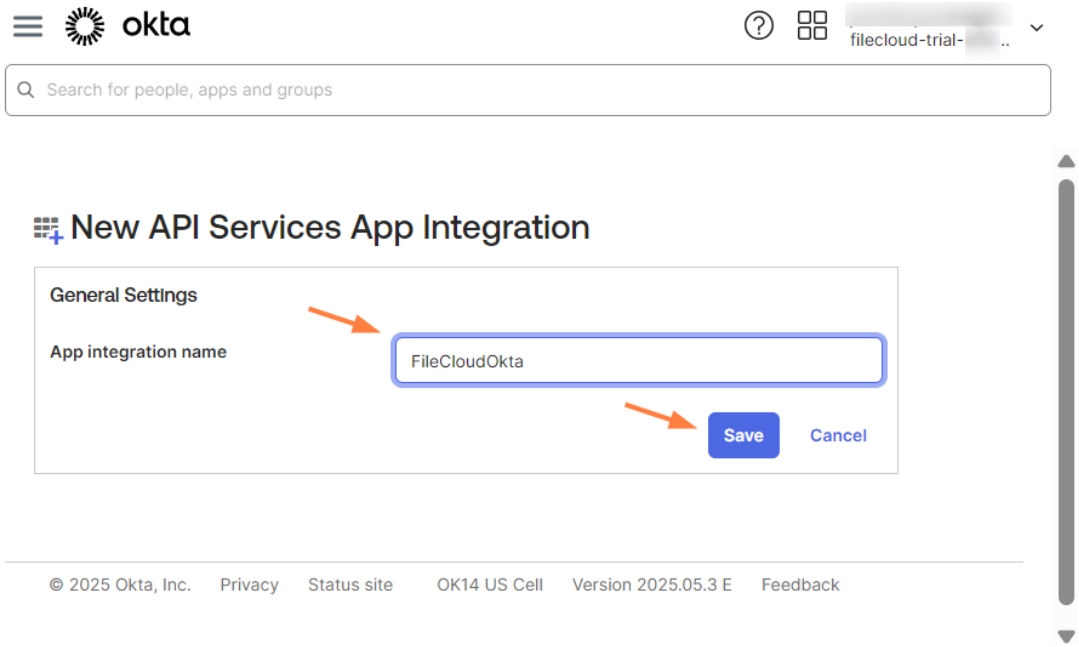


A list of sign-in methods opens.

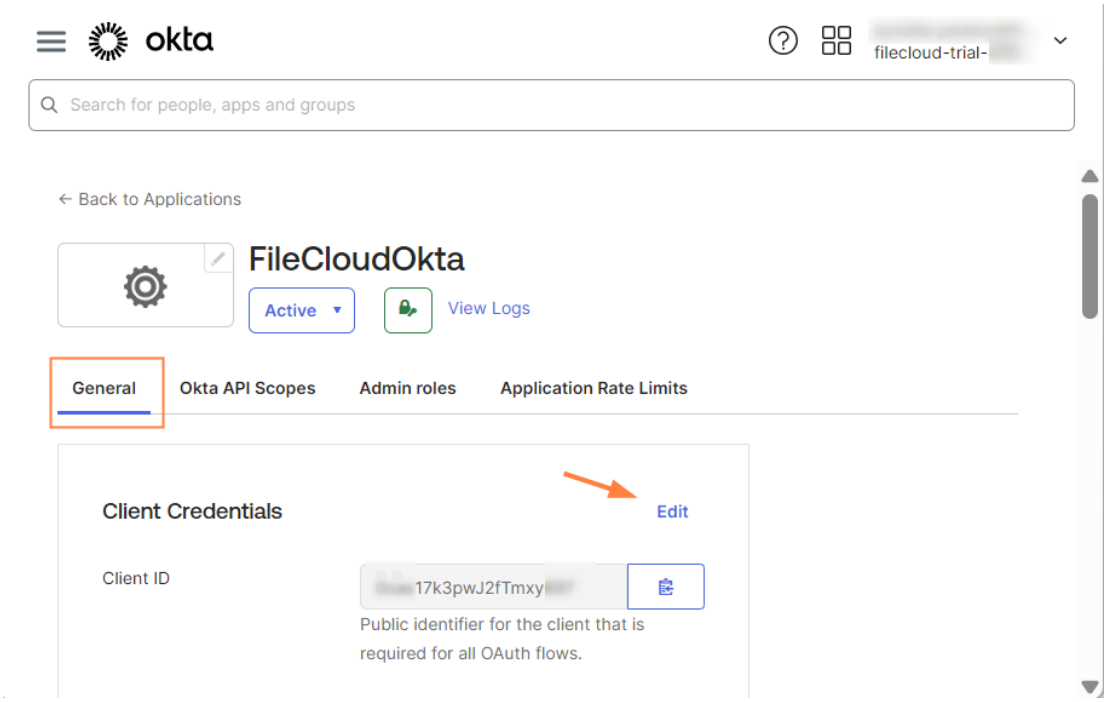
3. Choose **API Services**, and click **Next**.



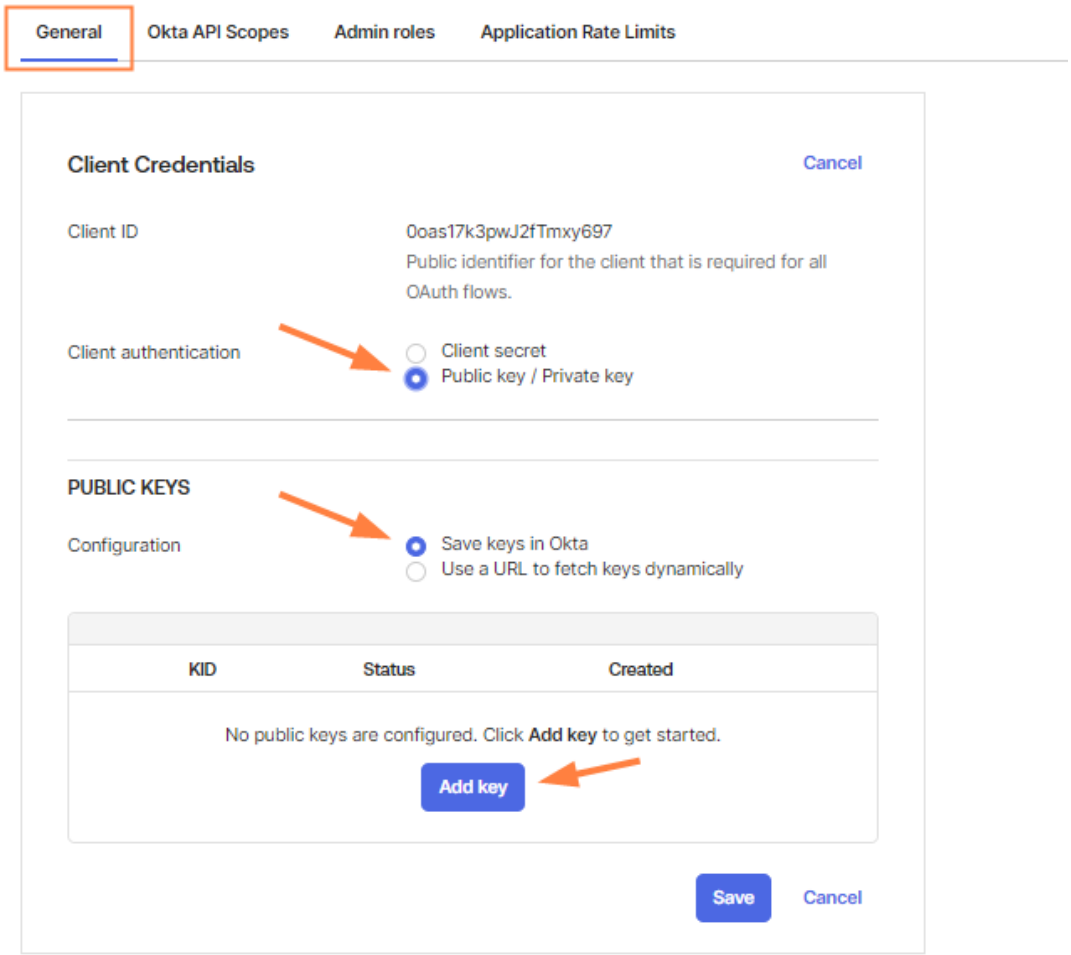
4. Enter a name for the app integration and click **Save**.



5. Your new app opens to the General tab. Click **Edit**.

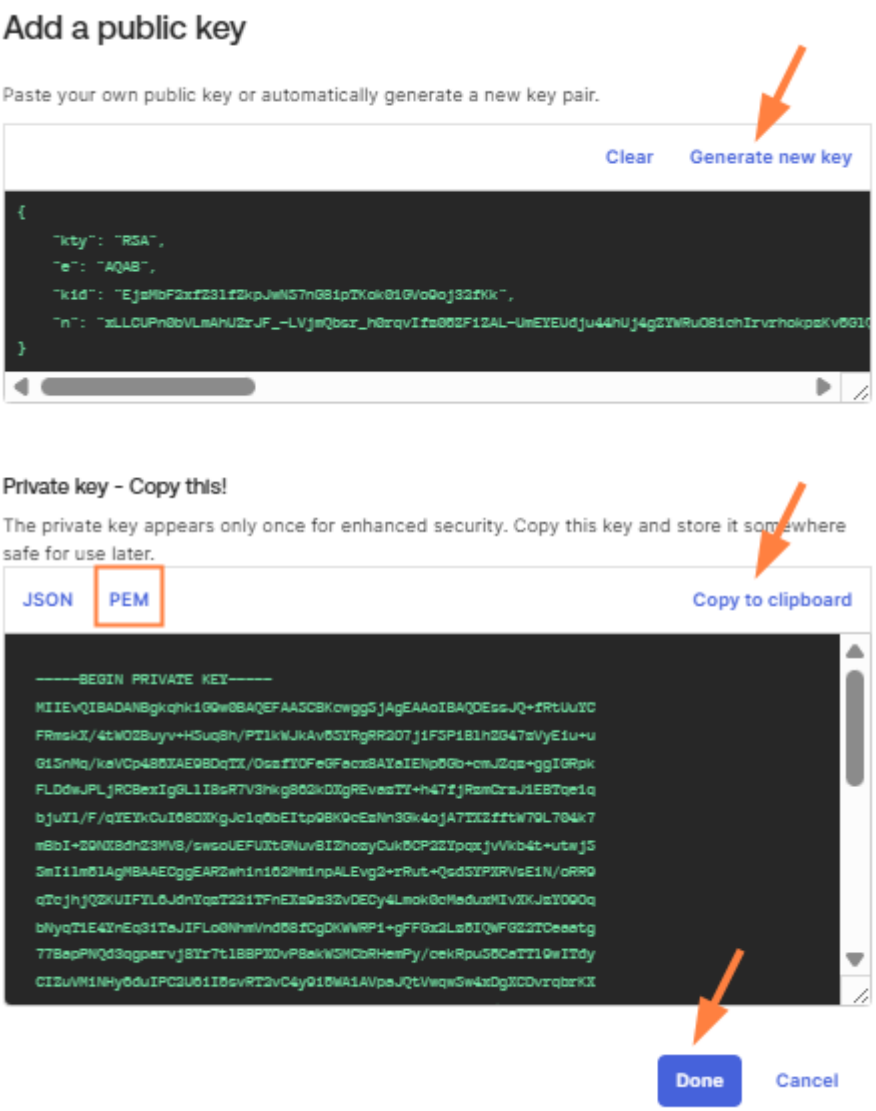


6. For **Client authentication**, select **Public key / Private key**.
7. For **Configuration**, choose **Save keys in Okta**.
8. Click **Add key**.

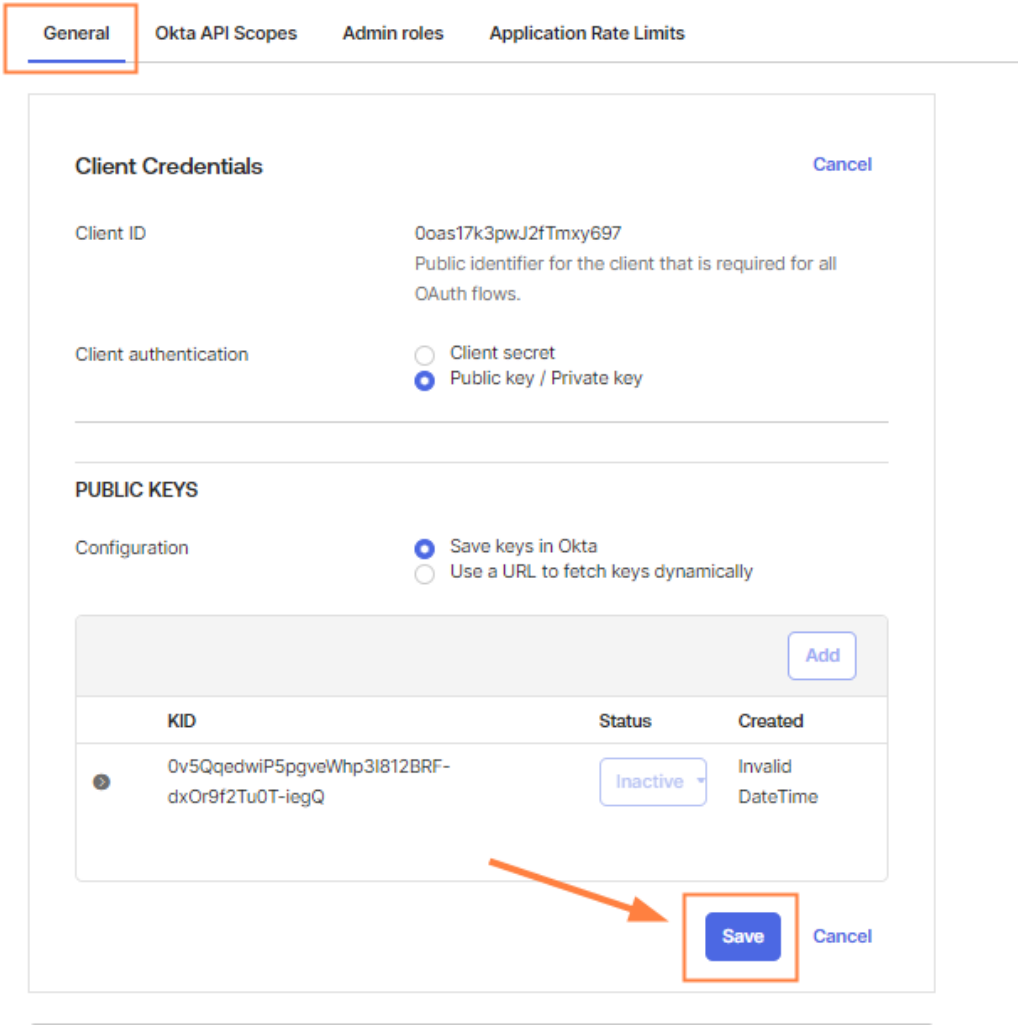


The **Add a public key** window opens.

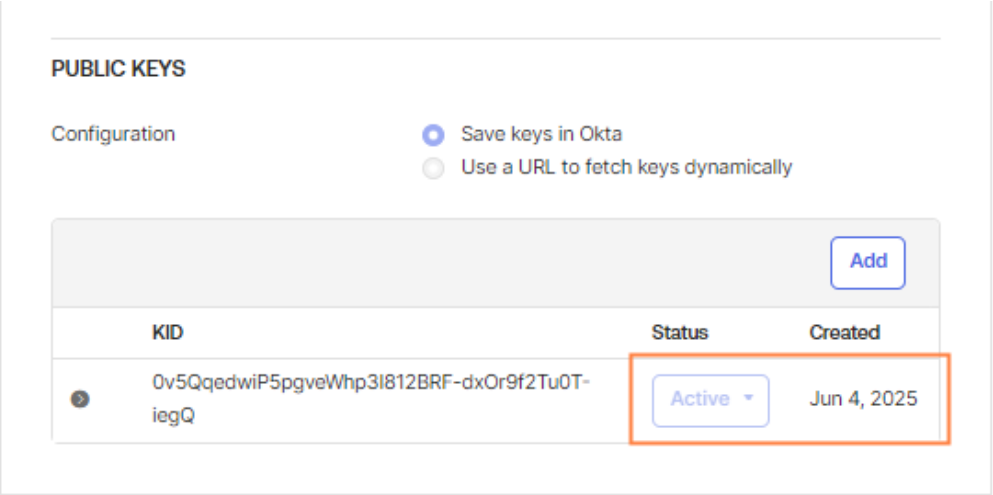
9. Paste in your own key or click **Generate new key**.
10. If you click **Generate new key**, under **Private key - Copy this!** click **PEM**, and then click **Copy to clipboard**, and save the copied key to a text file with a **.pem** extension so you can upload it to FileCloud.
    If you do not save as a **.pem** file, you will not be able to upload the private key to FileCloud.



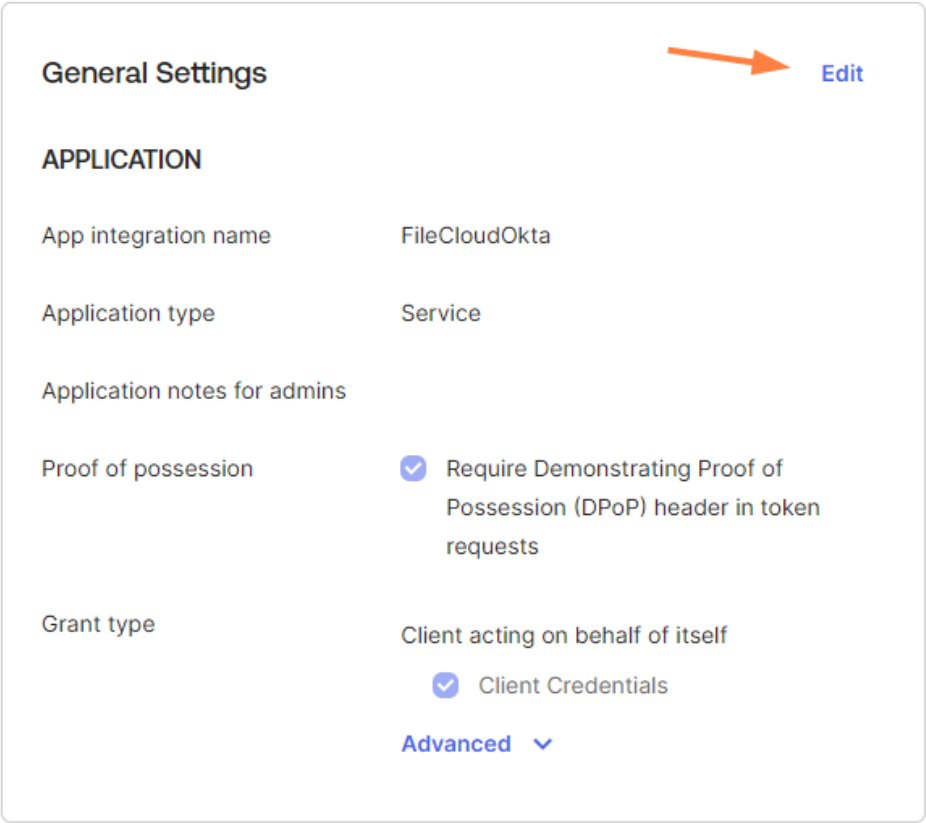
11. Click **Done**.
12. Click **Save**, or your public key will not be saved.



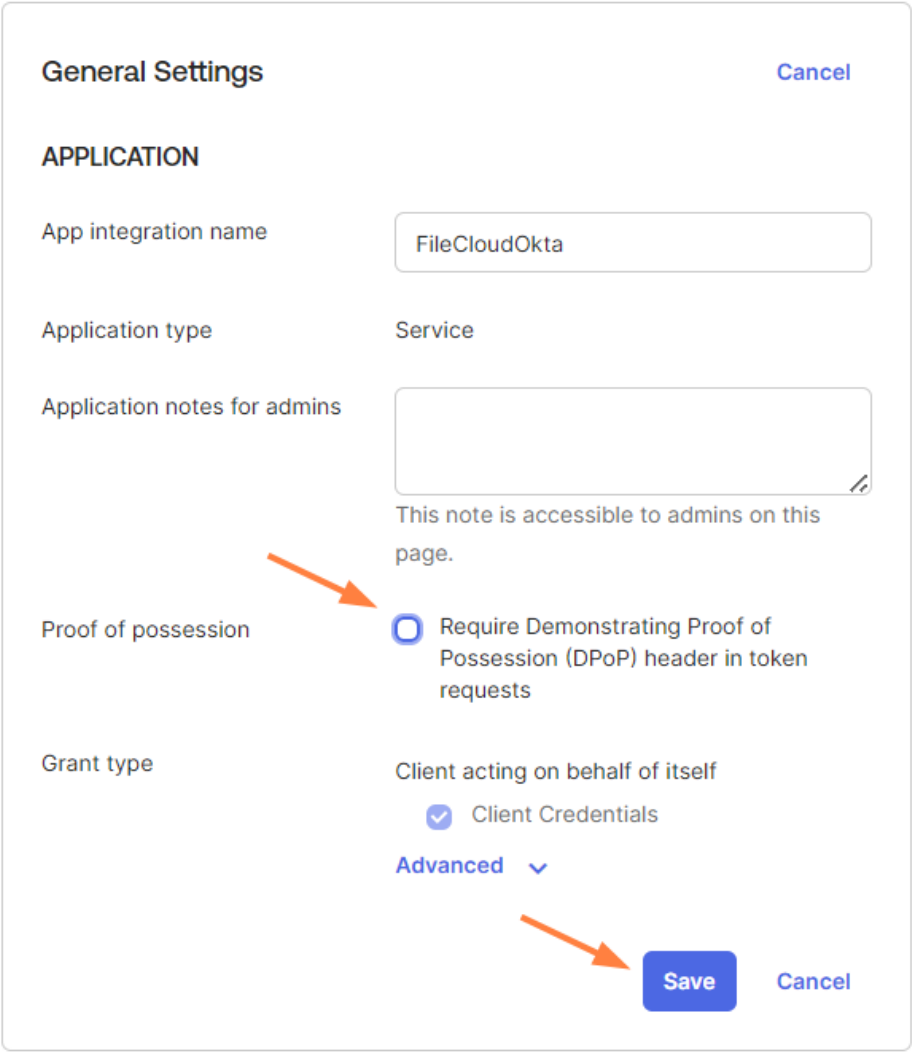
Once you click **Save**, your key should show a **Status** of **Active** and a **Created** date.



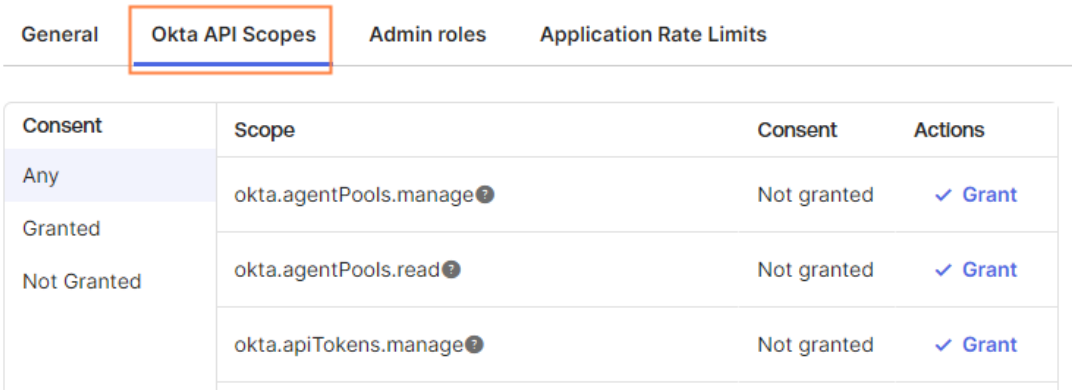
13. Remain on the General tab. Scroll down to **General Settings**, and click **Edit**



14. Uncheck **Proof of possession**, and click **Save**.



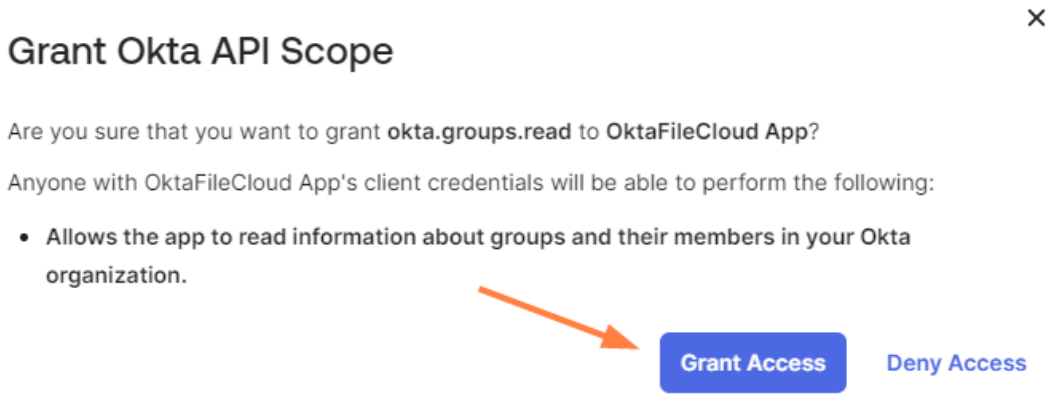
Click the **Okta API Scopes** tab.



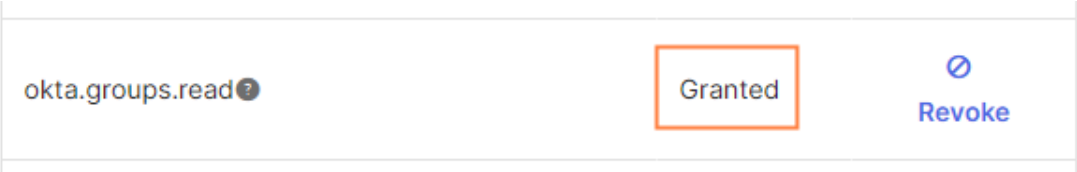
15. Scroll down to **okta.groups.read** and click **Grant** to enable FileCloud to read Okta groups.

You are prompted to grant **okta.groups.read** scope to the app.

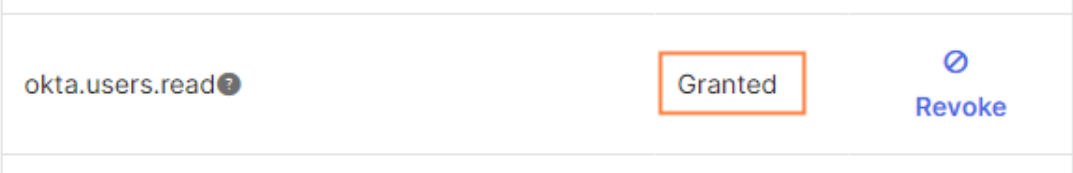
16. Click **Grant Access**.



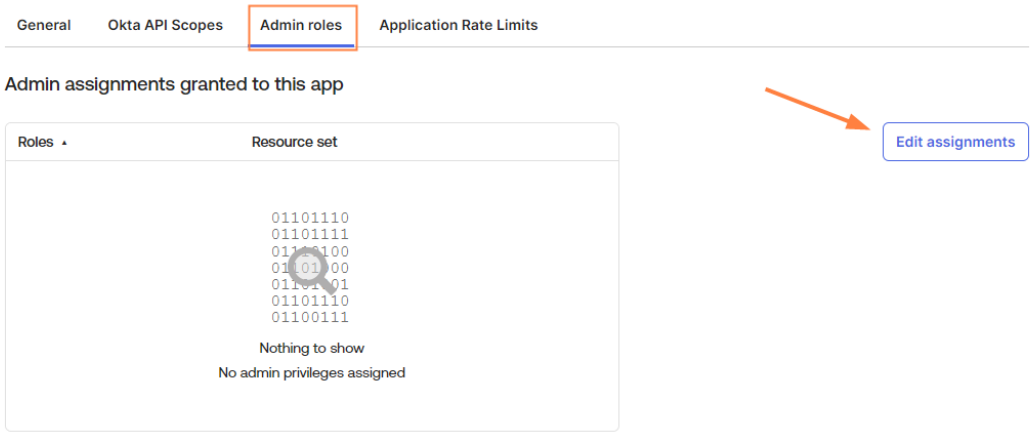
Now the row for **okta.groups.read** should appear as:



17. Scroll down to **okta.users.read** and click **Grant Access** to enable FileCloud to read Okta users. The **Grant Okta API Scope** notification does not appear again.

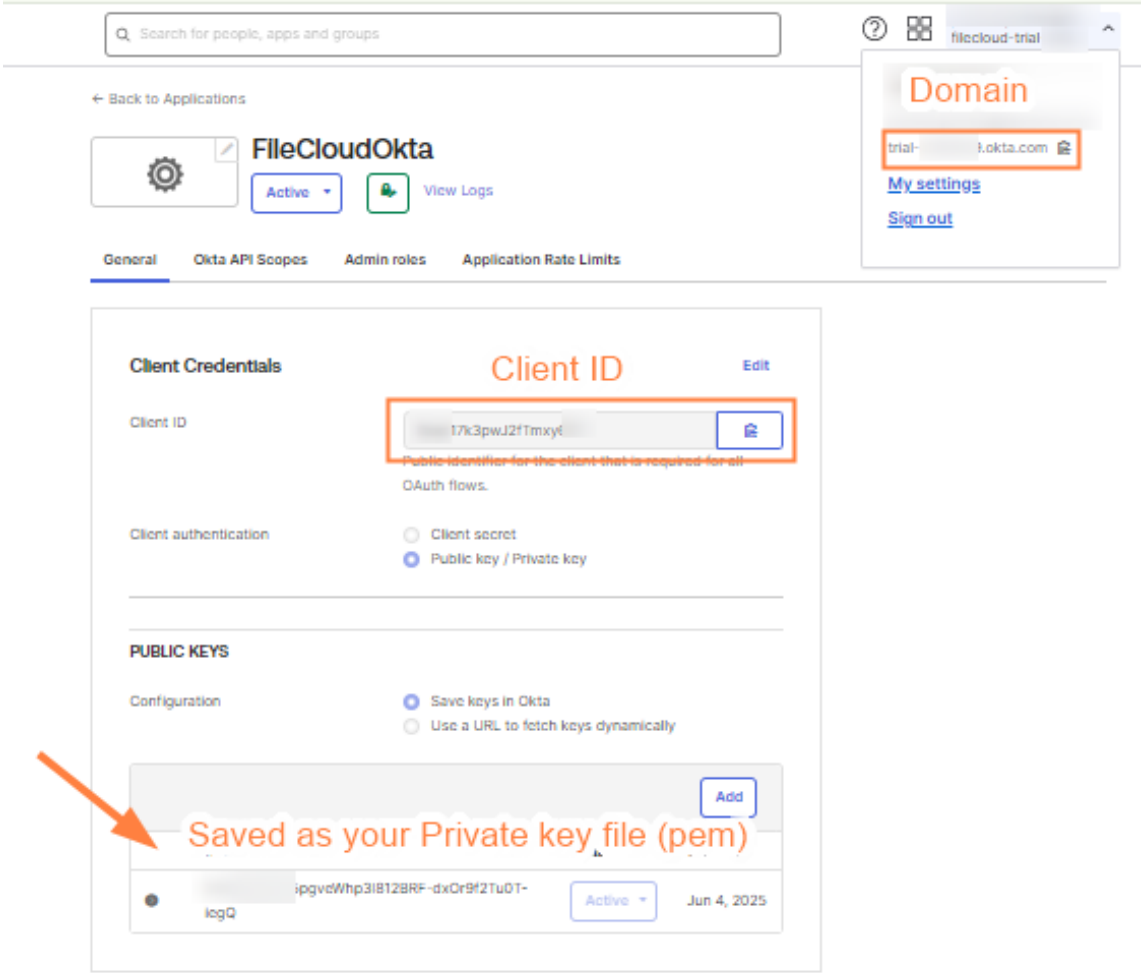


18. Click the **Admin roles** tab.
19. Click **Edit assignments**.



20. In **Role**, choose a role that should have access to Okta groups and users, or choose **Read-only Administrator**.
21. Click **Save Changes**.

    You have finished setting up integration on the Okta side.
    Now you have the values you need to set up integration on the FileCloud side: the domain in the user drop-down box, the **Client ID** on the General tab, and the .pem keyfile that you saved.

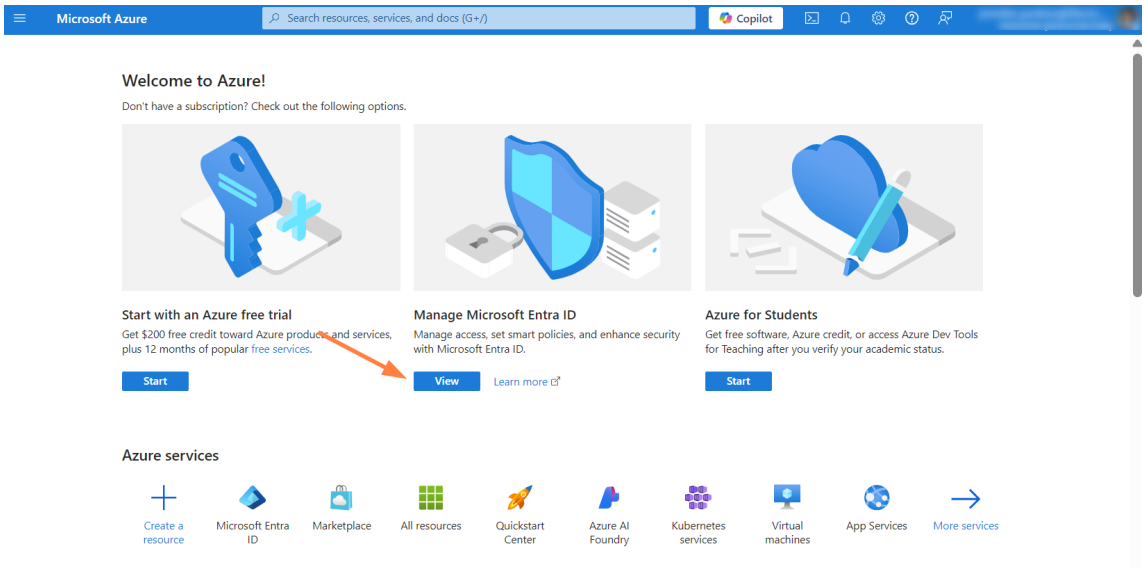


To enter the values into the FileCloud side, see SSO API: Configure Import of SSO Groups and Users.

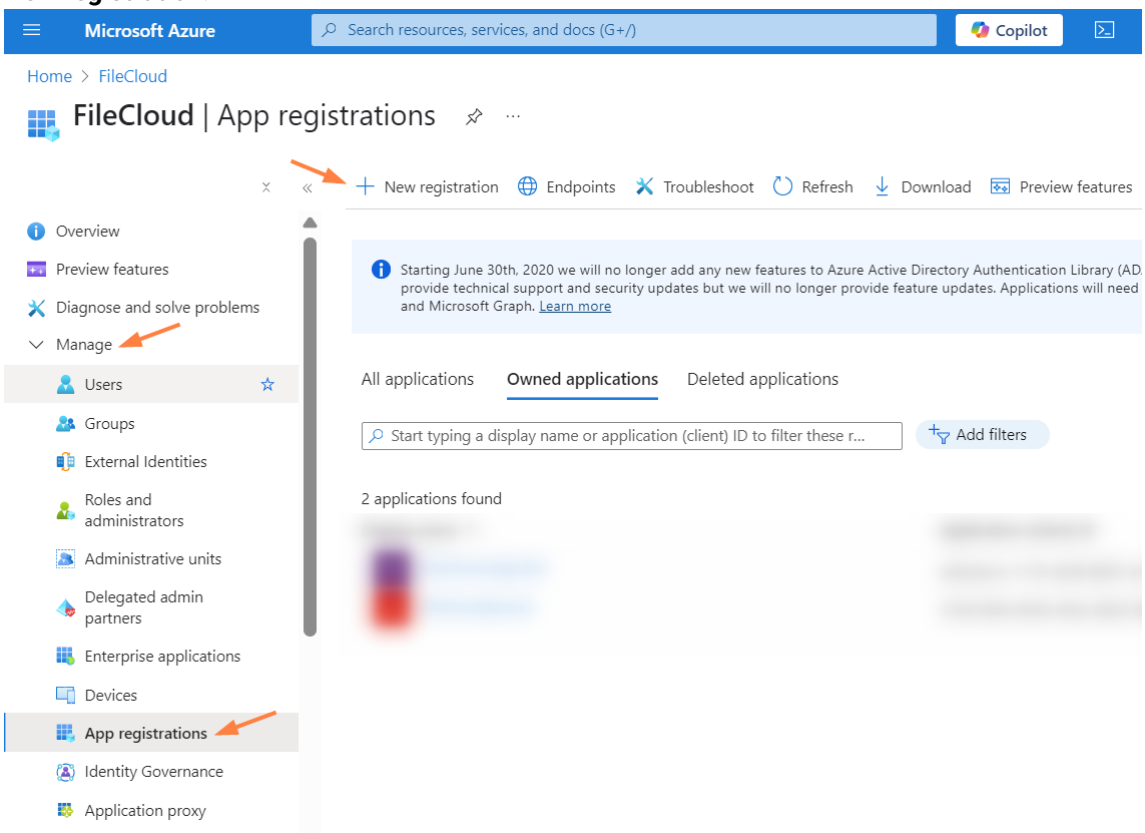
Azure: Set up FileCloud Integration for SSO Group/User Import

To configure FileCloud/Azure integration in Azure for SSO group/user import:

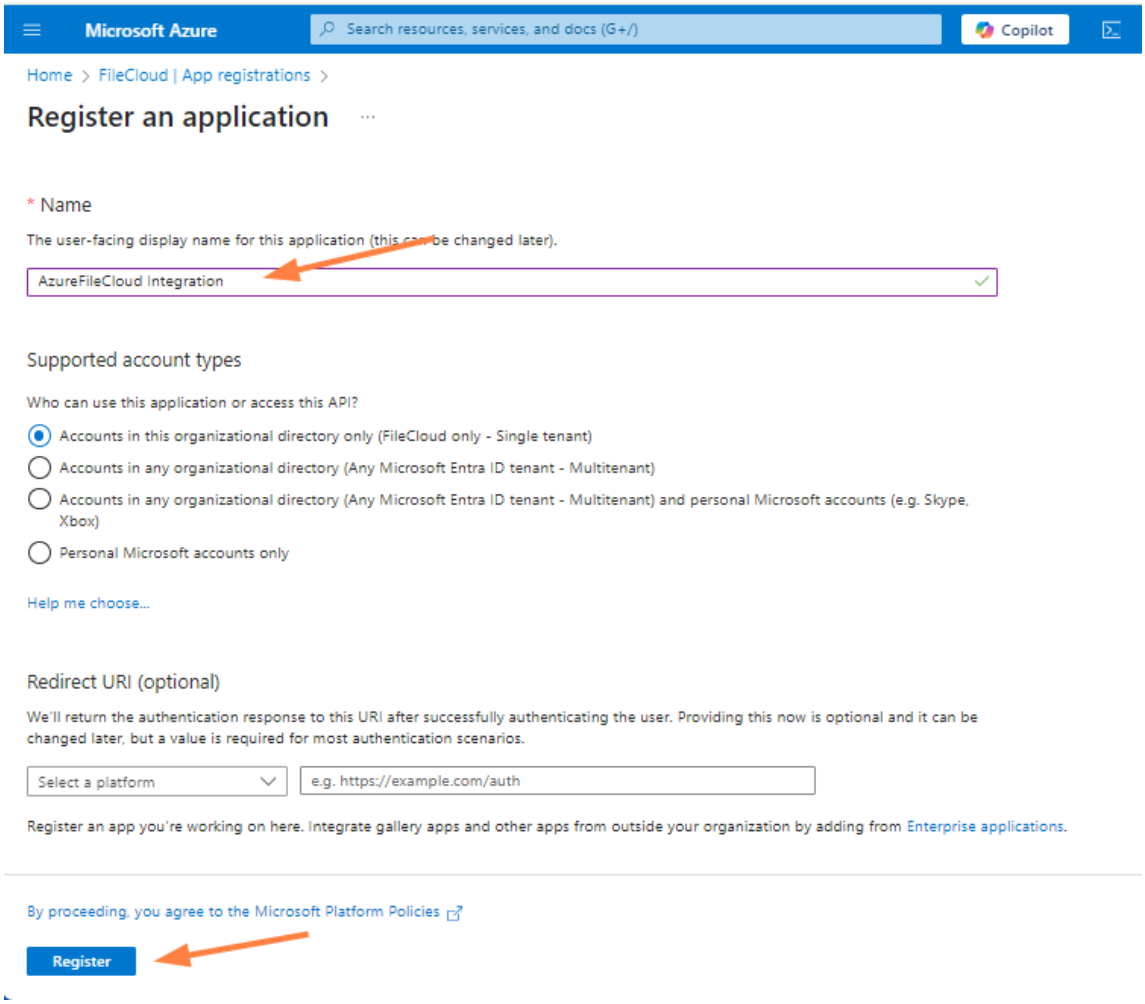
1. Log into portal.azure.com, and go to Microsoft Entra ID (https://entra.microsoft.com/) .



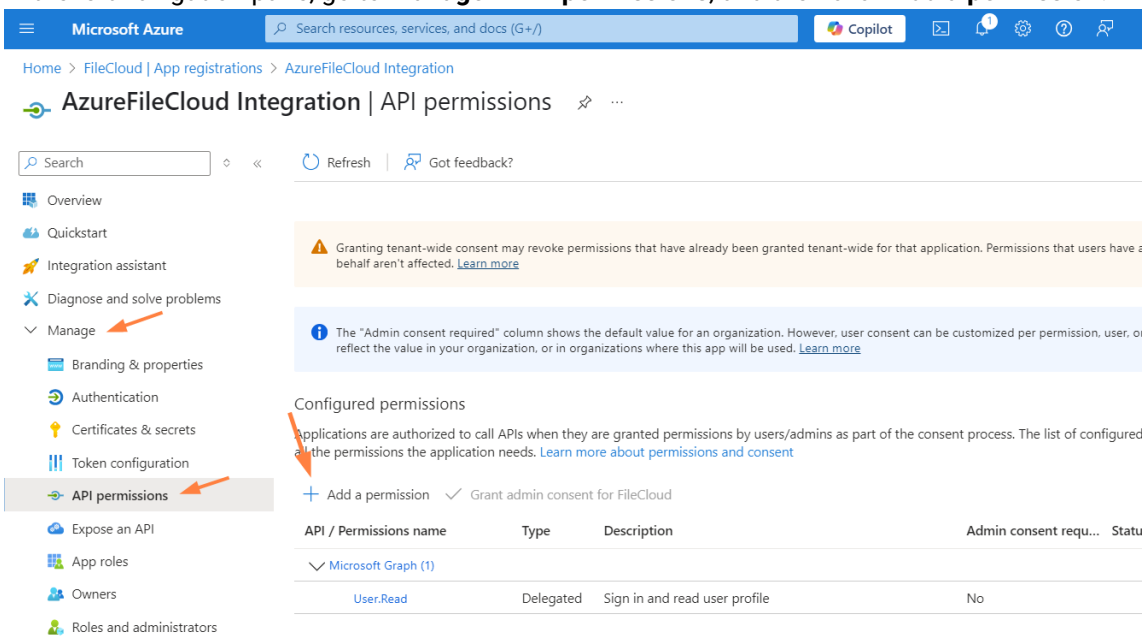
2. In the left navigation pane, go to Manage > App registrations, and at the top of the page, click **New registration**.



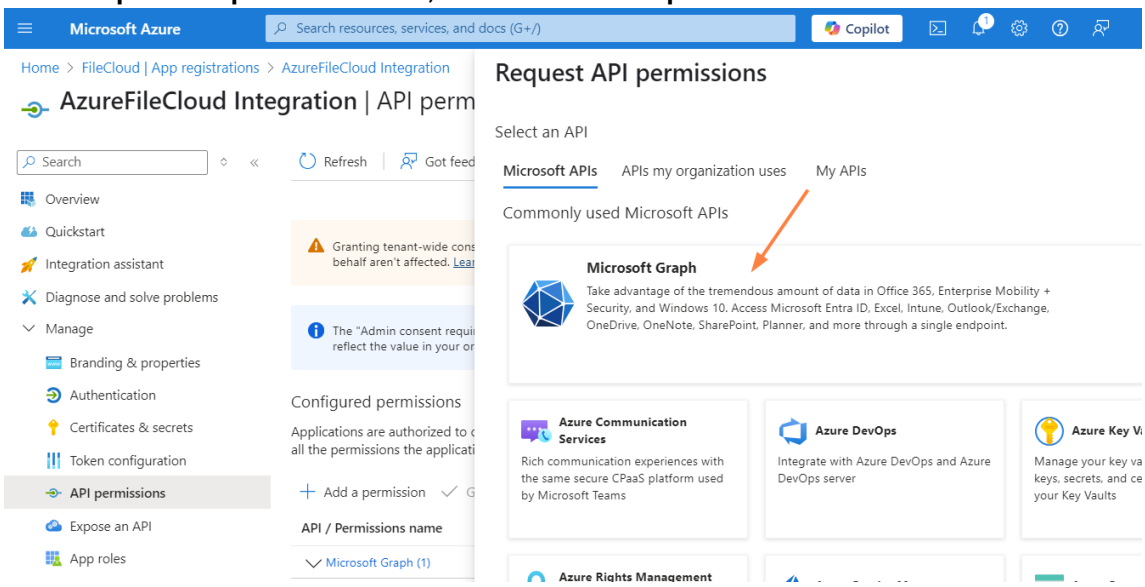
3. Enter a name for the application, and then click **Register**.



4. In the left navigation pane, go to **Manage > API permissions**, and then click **Add a permission**.

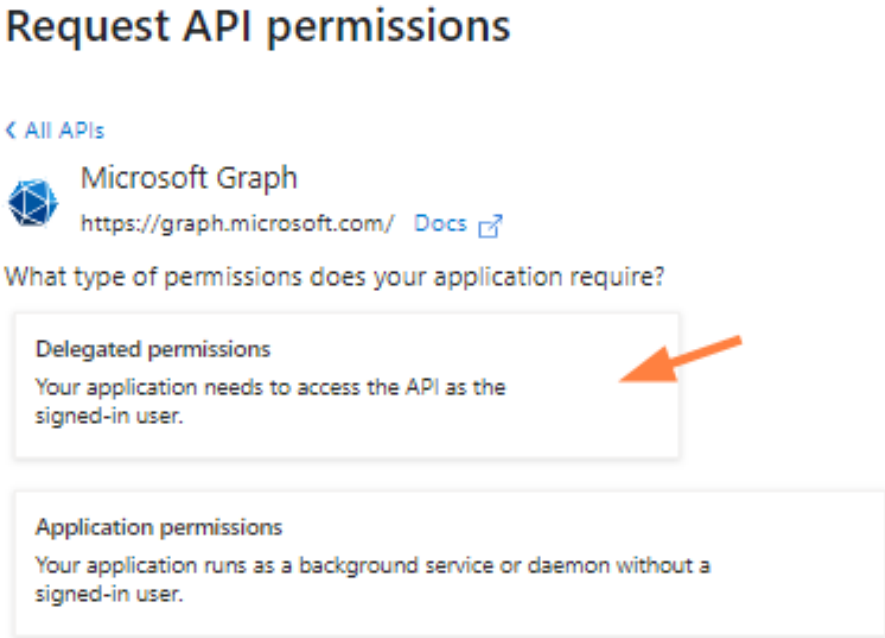


5. In the **Request API permissions** box, click **Microsoft Graph**.



In the **Request API Permissions** box, you are prompted to choose a type of permission.

6. First, select **Delegated permissions**, and request the permissions specified below.

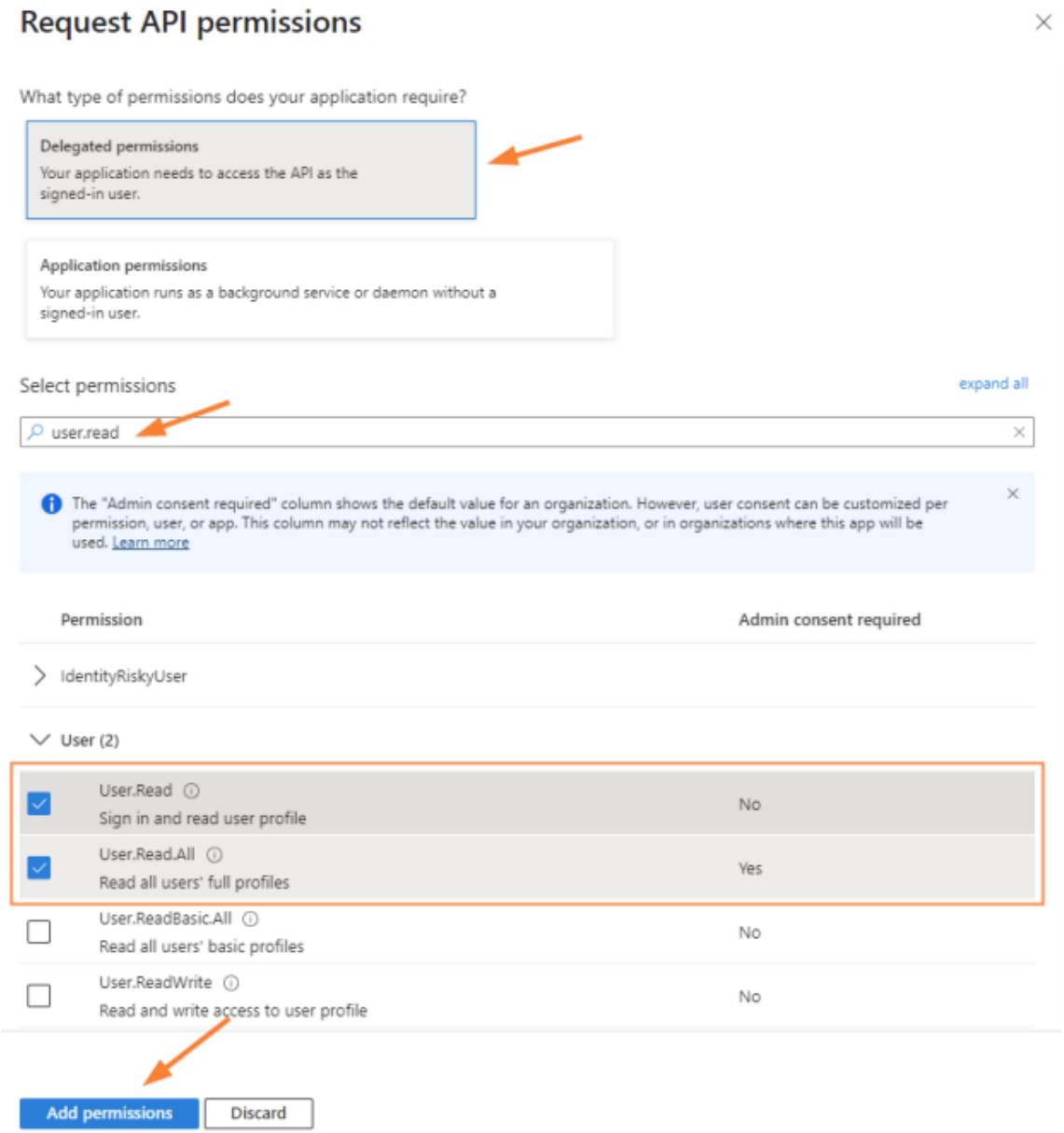


Delegated permissions to request:

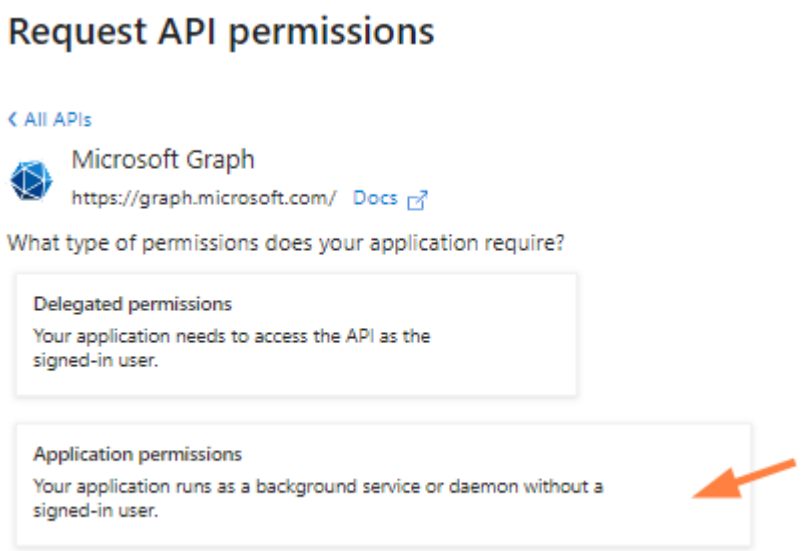
- Directory.Read.All
- Group.Read.All
- GroupMember.Read.All

- User.Read
- User.Read.All

7. Search for the permission type in the Select permissions search bar to find it more quickly, and then check the permissions.
8. When you are done checking all of the above permissions, click **Add permissions**.



Now, in the Request API permissions box, choose Application permissions, and request the permissions specified below:



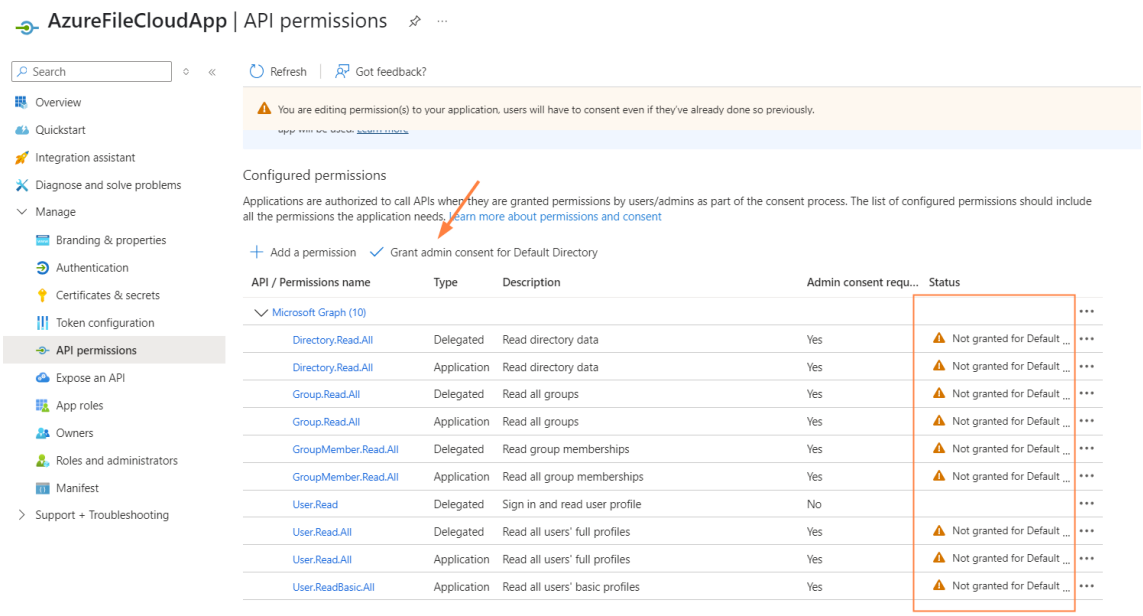
Application permissions to request:

- Directory.Read.All
- Group.Read.All
- GroupMember.Read.All
- User.Read.All
- User.ReadBasic.All

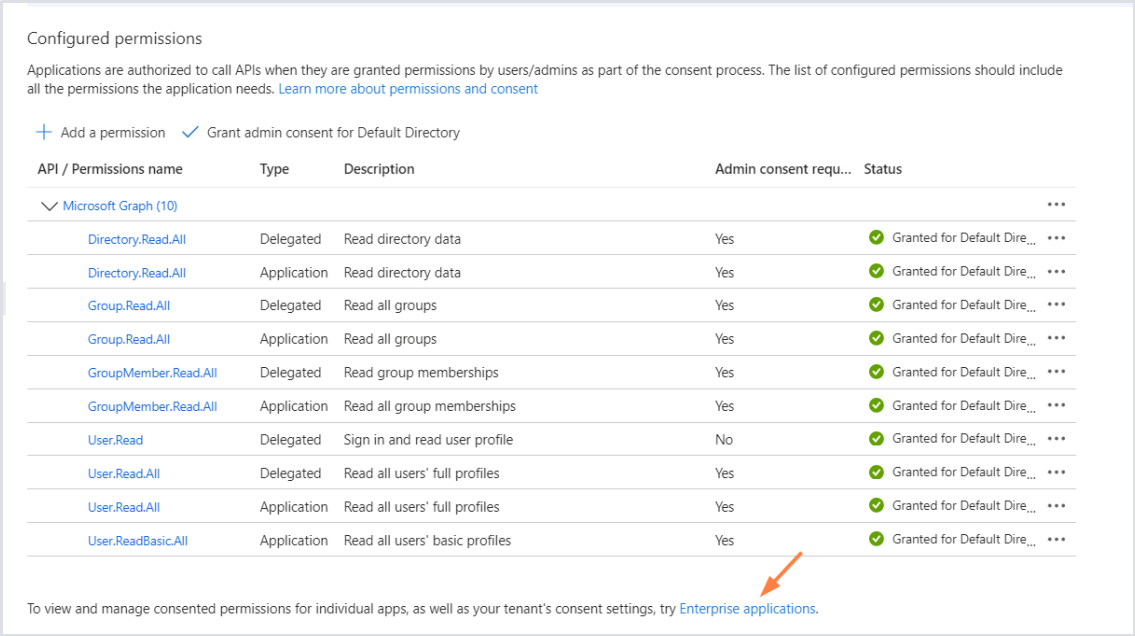
9. Repeat steps 7 and 8 to request the Application permissions.

   Initially, most of the permissions show that permission has not been granted.

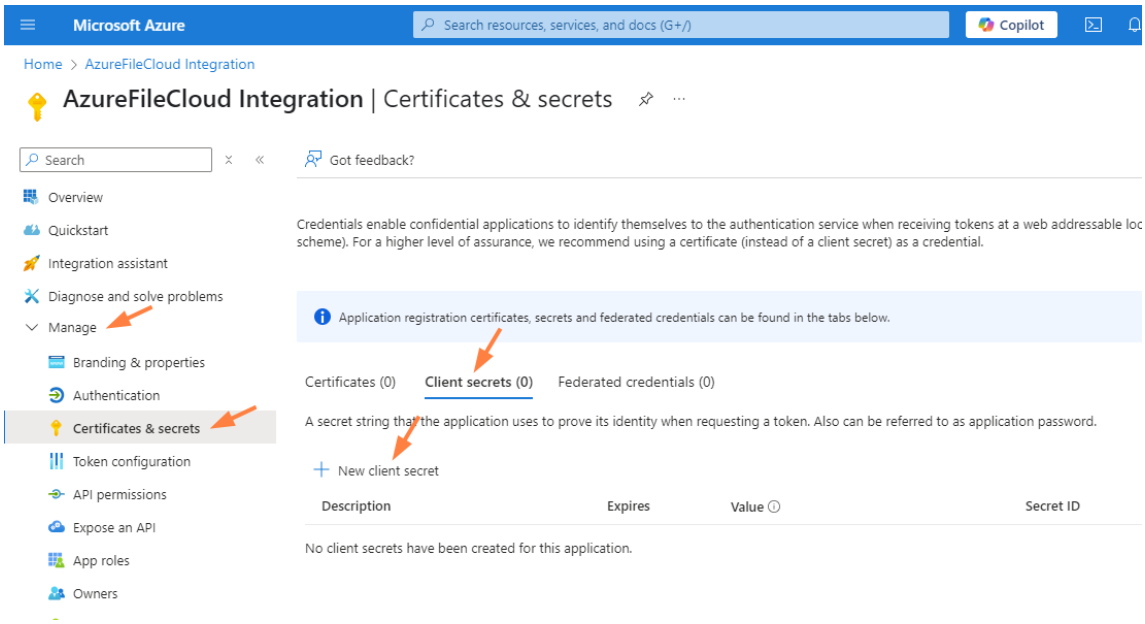
10. Above the list, click **Grant admin consent for [Tenant name]**, and when prompted, click **Yes.**
    Note: If you are not a global admin, you must ask your global admin to grant the API permissions for you.



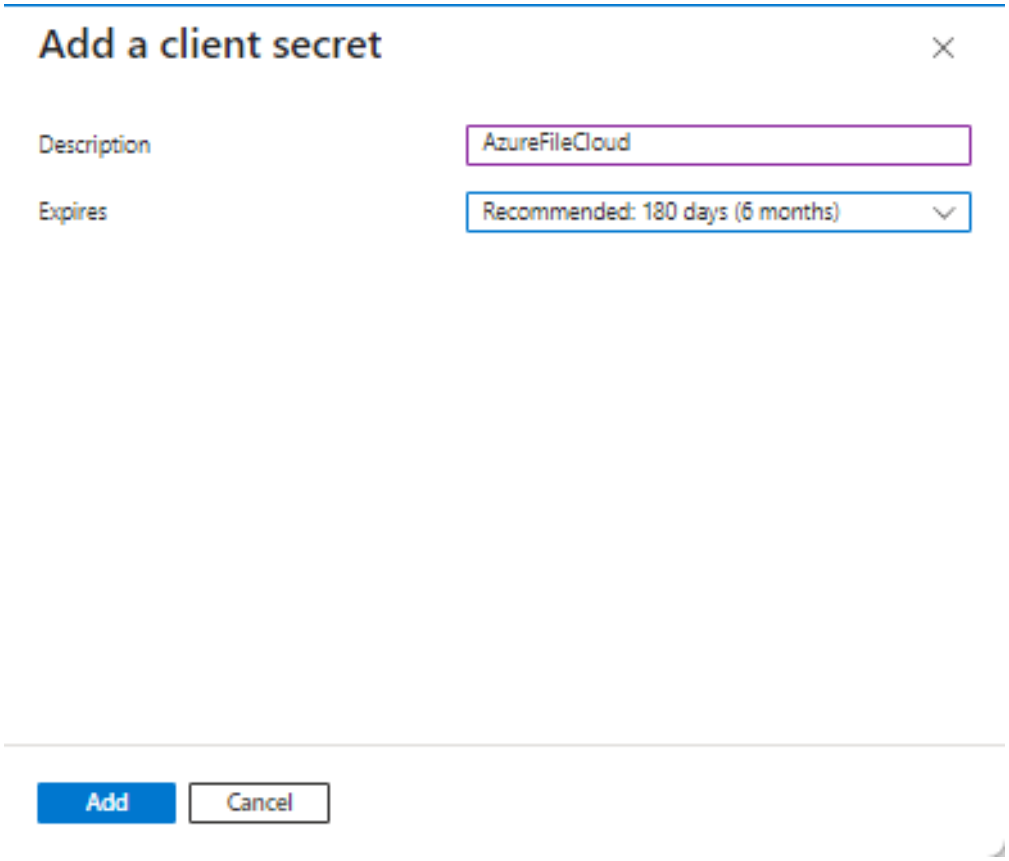
When all of your permissions have been granted, your list of permissions should appear similar to:



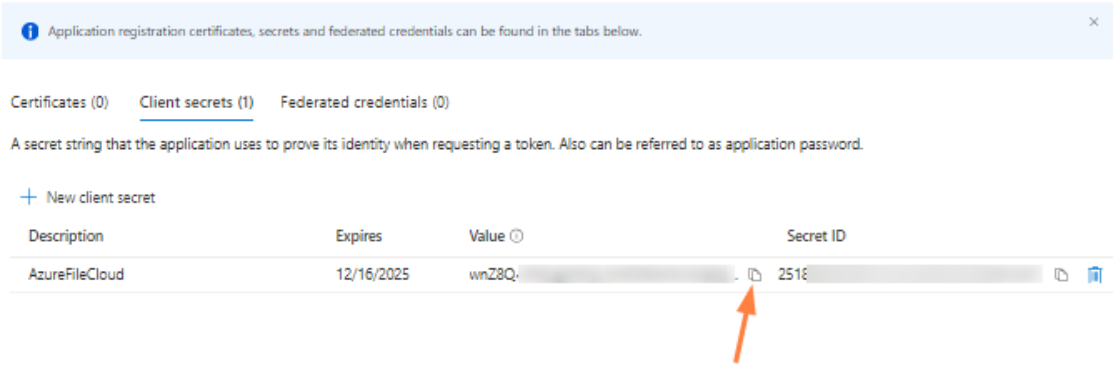
11. Now, in the left navigation pane, go to **Manage > Certificates & secrets**, and click the **Client secrets** tab.
12. Click **New client secret**.



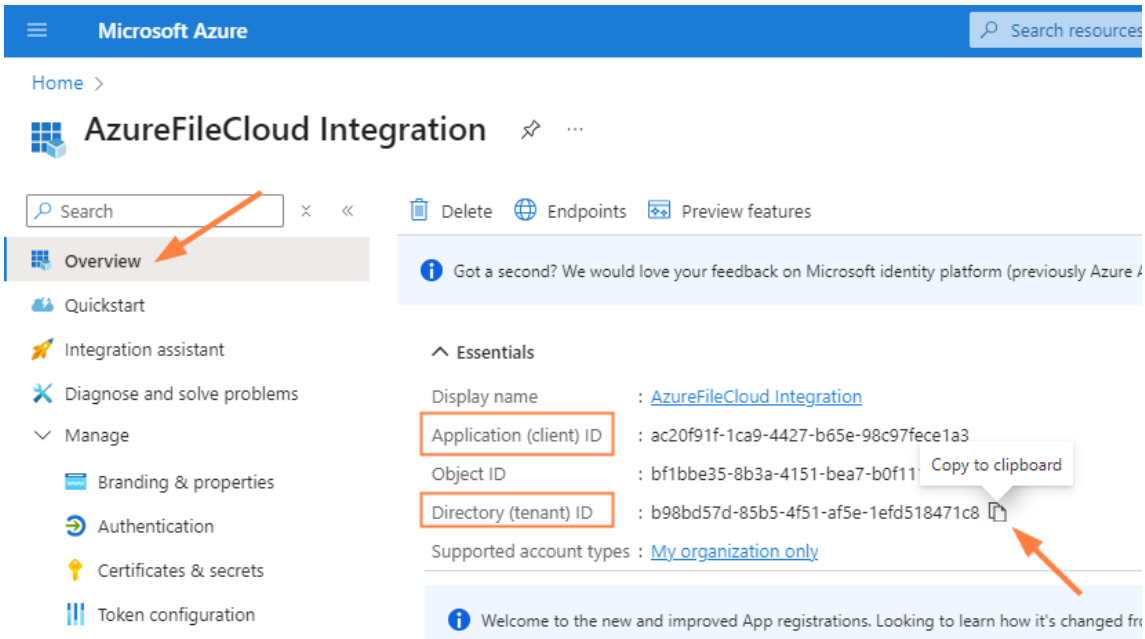
13. In the **Add a client secret** box, enter a description for the client secret, and choose an expiration date, then click **Add**.



14. Click the copy icon next to **Value** and save it. You will use it to fill in the **Client Secret** field in FileCloud.



15. In the left navigation pane, click **Overview**.
16. Hover over **Directory (tenant) ID** and click the copy icon. Save the **Directory (tenant) ID**. You will use it to fill in the **Tenant ID** field in FileCloud.
17. Hover over **Application (client) ID** and click the copy icon. Save the **Application (client) ID**. You will use it to fill in the **Client ID** field in FileCloud

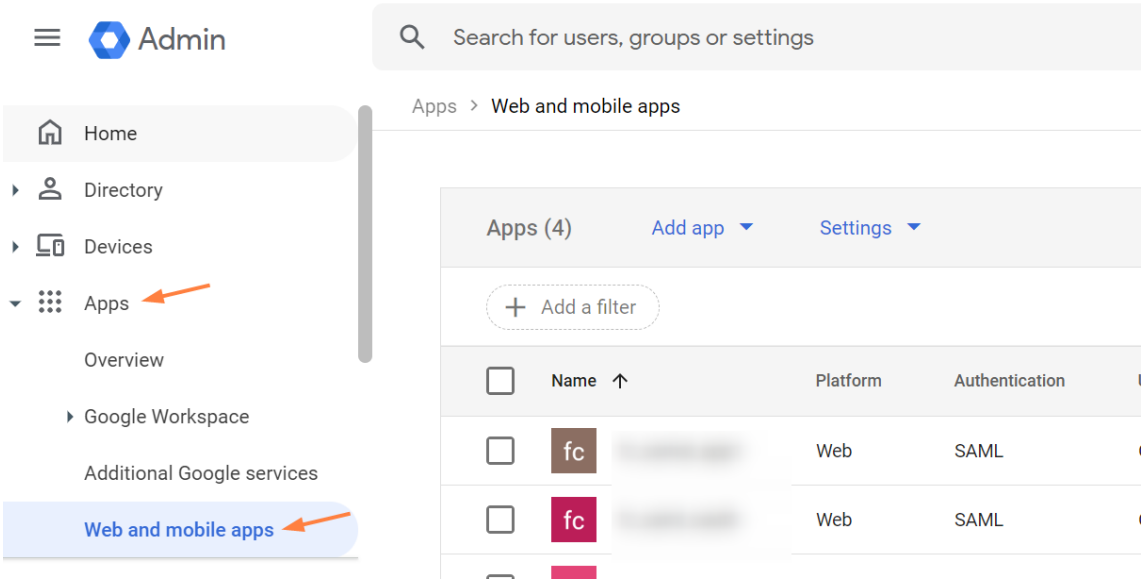


To enter the integration values into the FileCloud side, see SSO API: Configure Import of SSO Groups and Users.

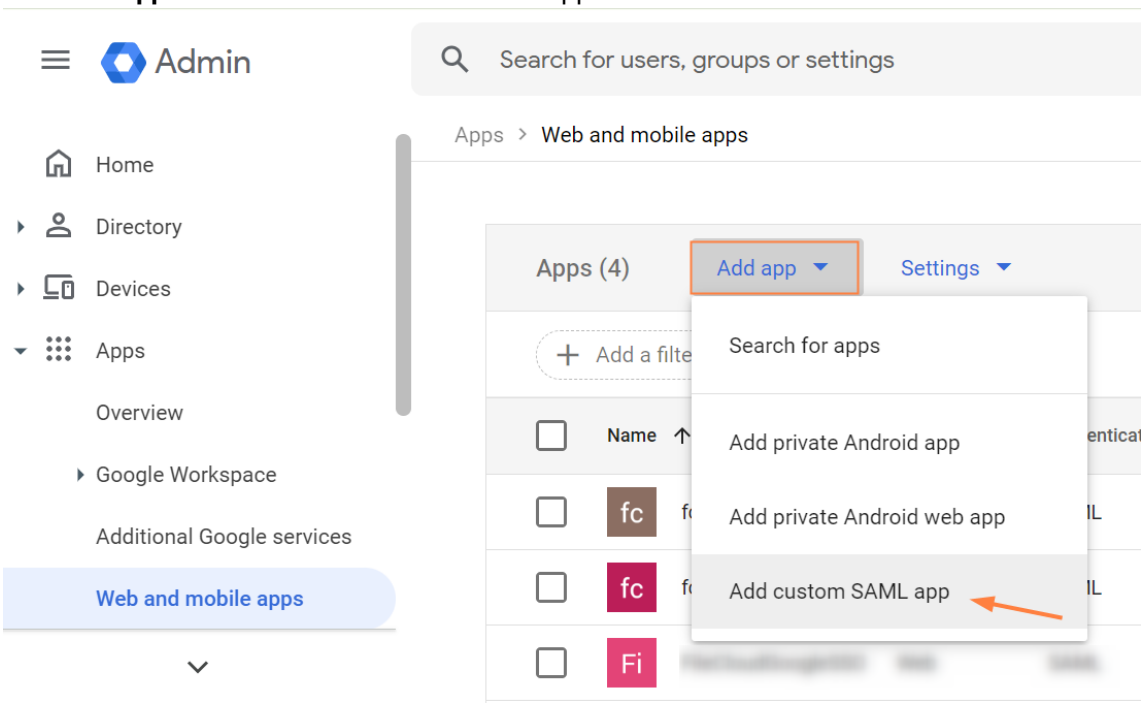
Google: Set up FileCloud Integration for SSO Group/User Import

To configure FileCloud/Google integration in Google for SSO group/user import:

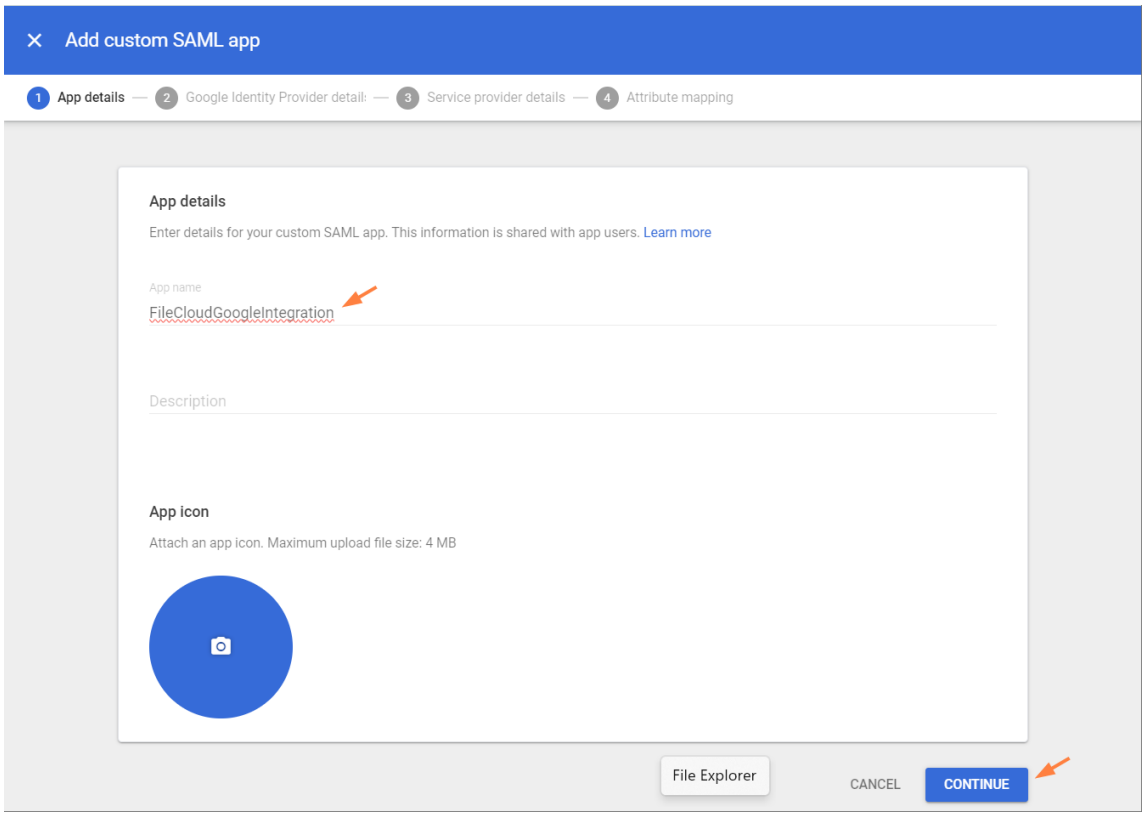
1. Log in to the Google Workspace Admin Center at admin.google.com.
2. In the left navigation pane, go to **Apps > Web and mobile apps**.



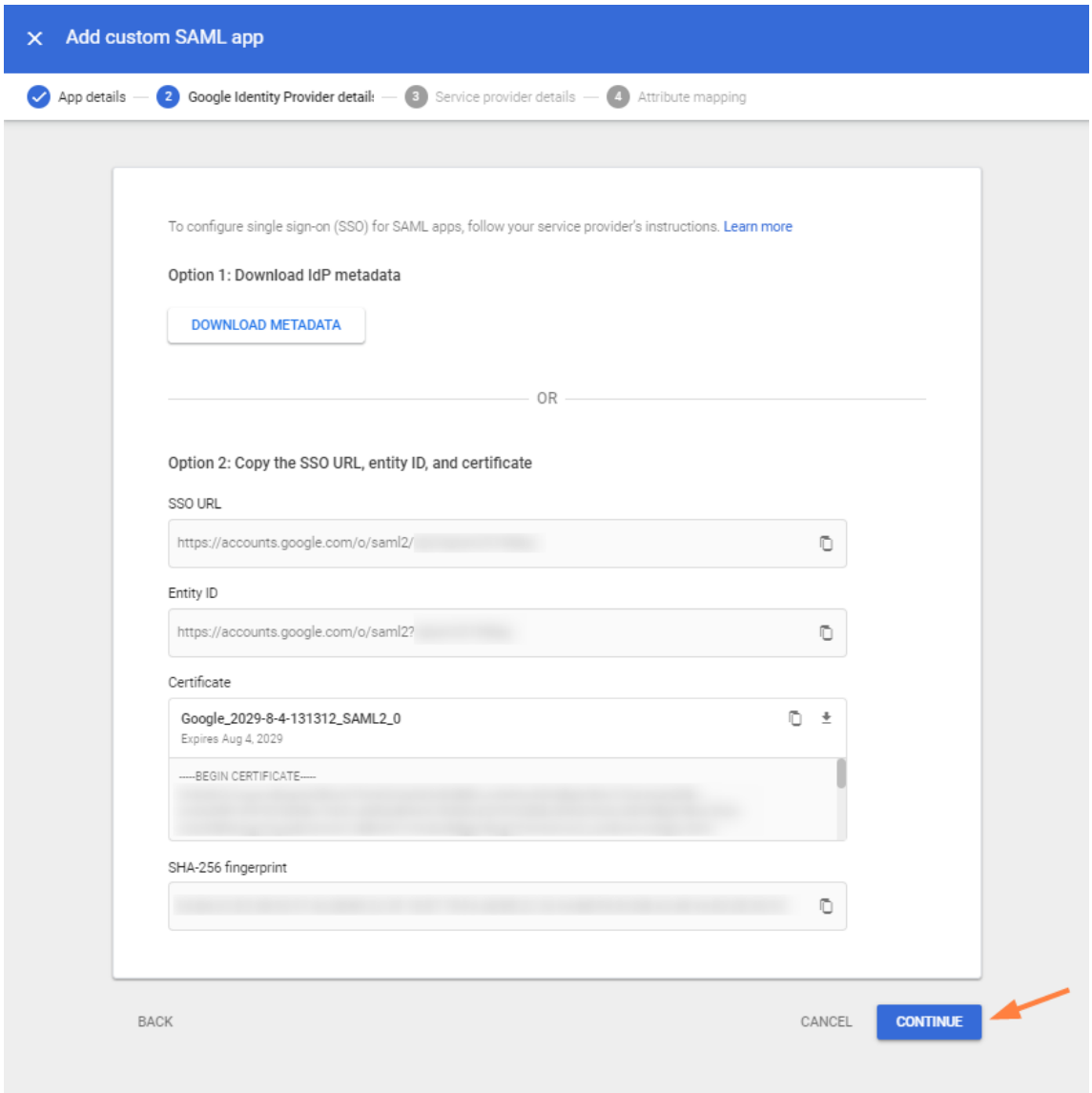
3. Click **Add app** and choose **Add custom SAML** app.



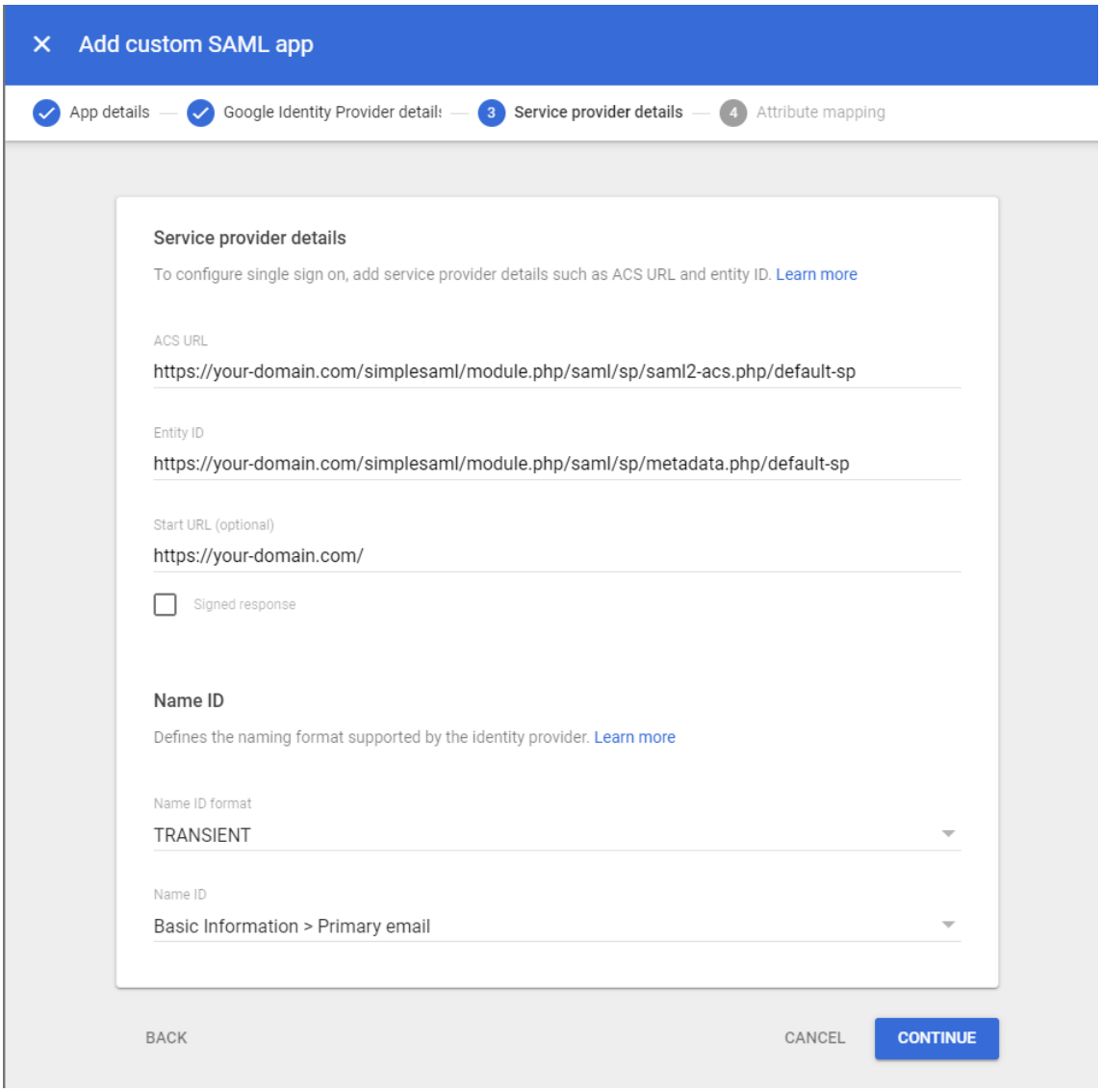
4. Enter an **App name**, and click **CONTINUE**.



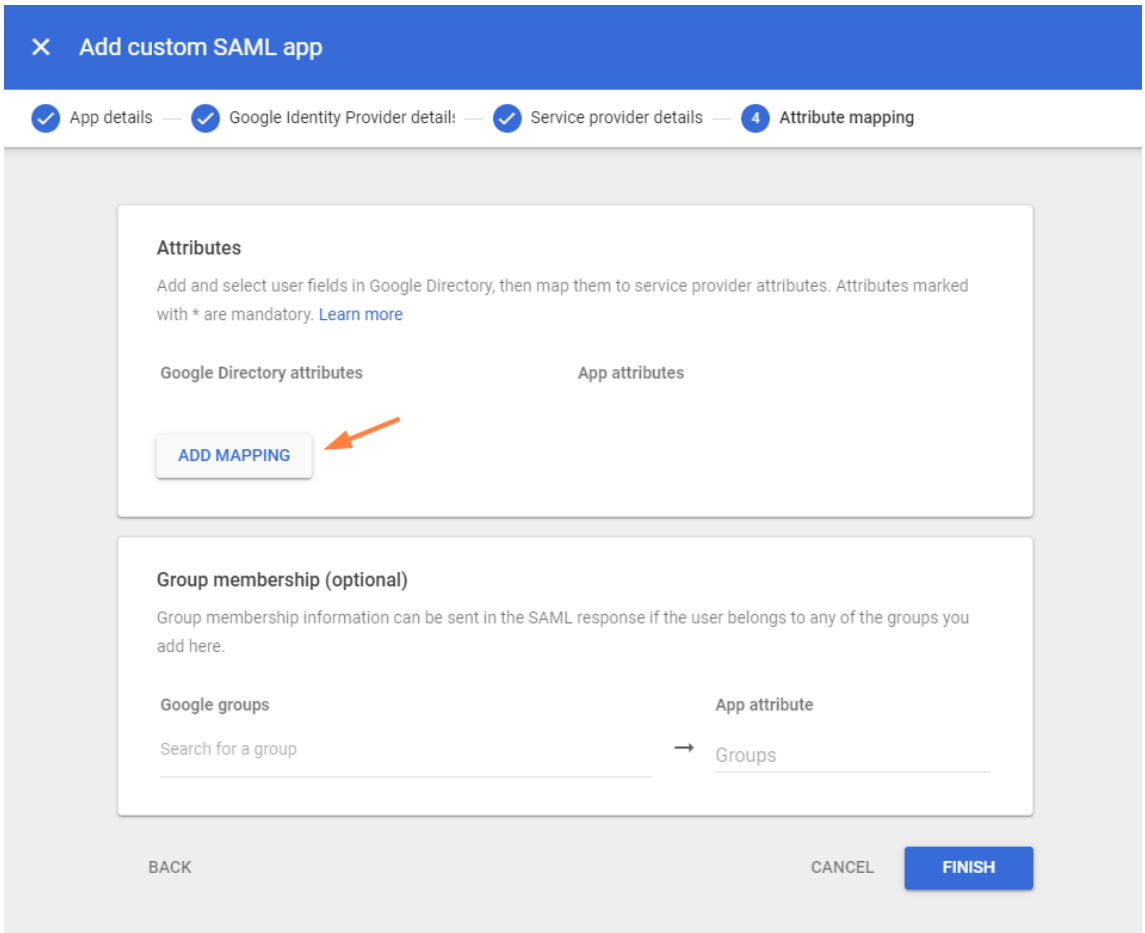
5. Click **CONTINUE**.



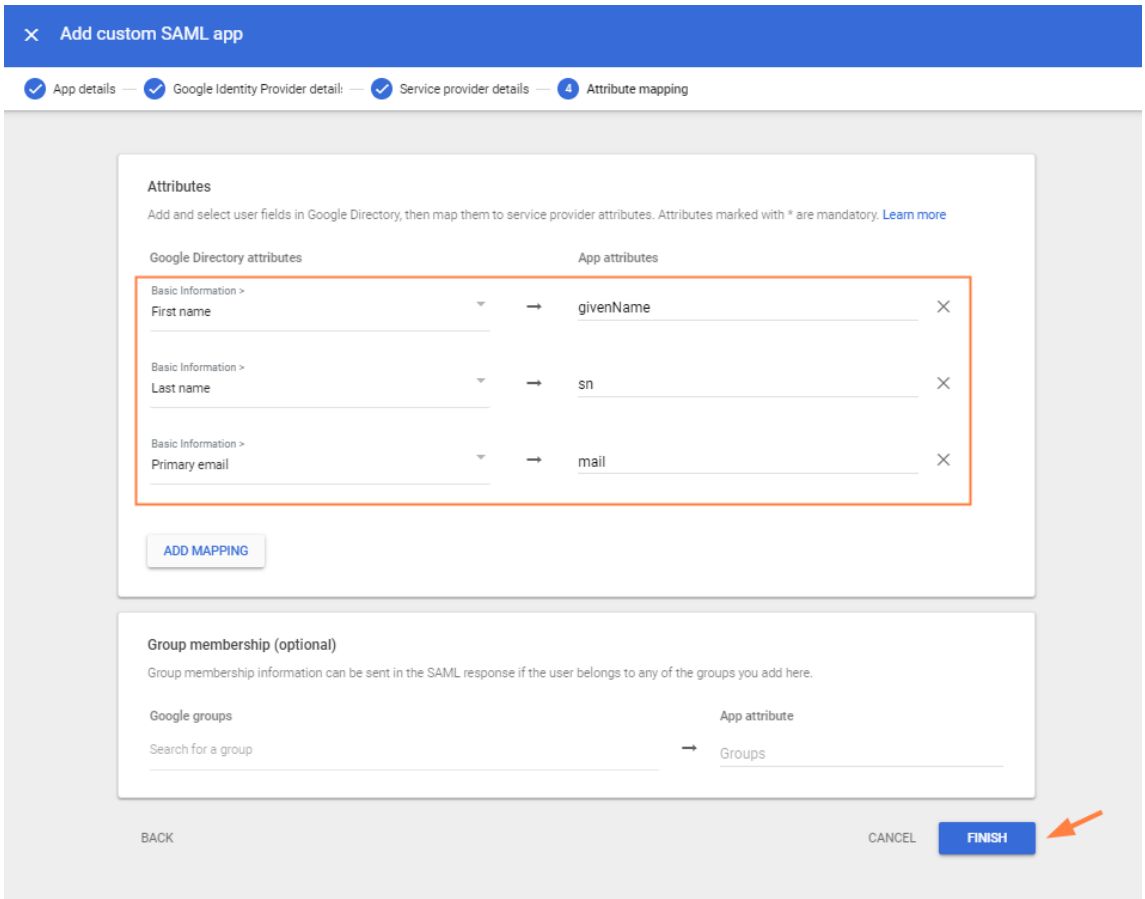
6. Fill in the fields as follows, replacing **your-domain.com** with your FileCloud domain. Click **CONTINUE**.
   ACS URL: https://your-domain/simplesaml/module.php/saml/sp/saml2-acs.php/default-sp
   Entity ID: https://your-domain/simplesaml/module.php/saml/sp/metadata.php/default-sp
   Start URL: https://your-domain/
   Name ID Format: **TRANSIENT**
   NameID: **Basic Information > Primary Email**



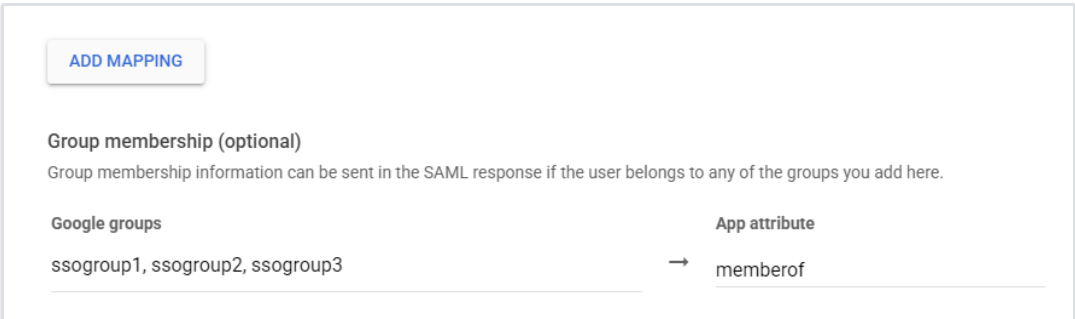
7. Click **ADD MAPPING**.



8. Choose the **Google Directory attributes** below, and add the specific values shown to **App attributes**.



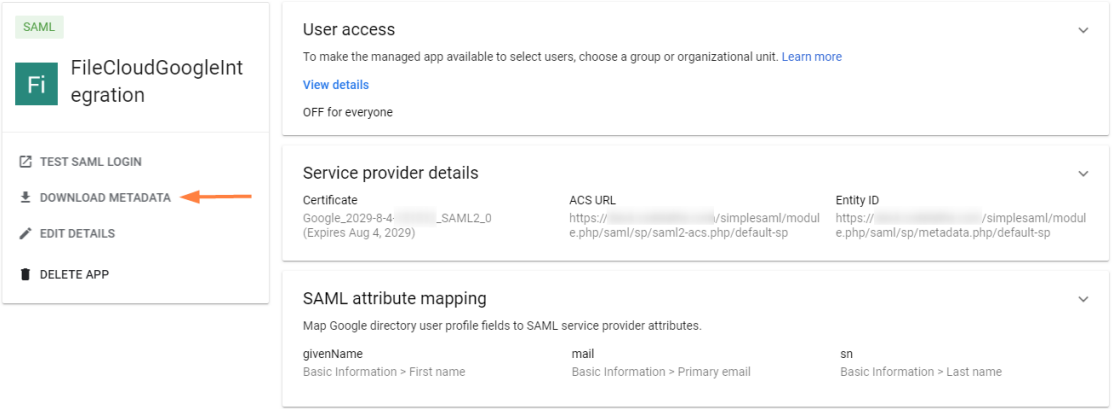
9. **Optional**: If you have chosen to sync groups each time a user logs in to FileCloud, you must also add the groups that the user is a member of under **Group membership (optional)**.
   a. In **Google groups**, on one line enter all of the groups in Google that users imported to FileCloud may belong to.
   b. In **App attribute**, enter **memberof**.
   c. Click **Save**.



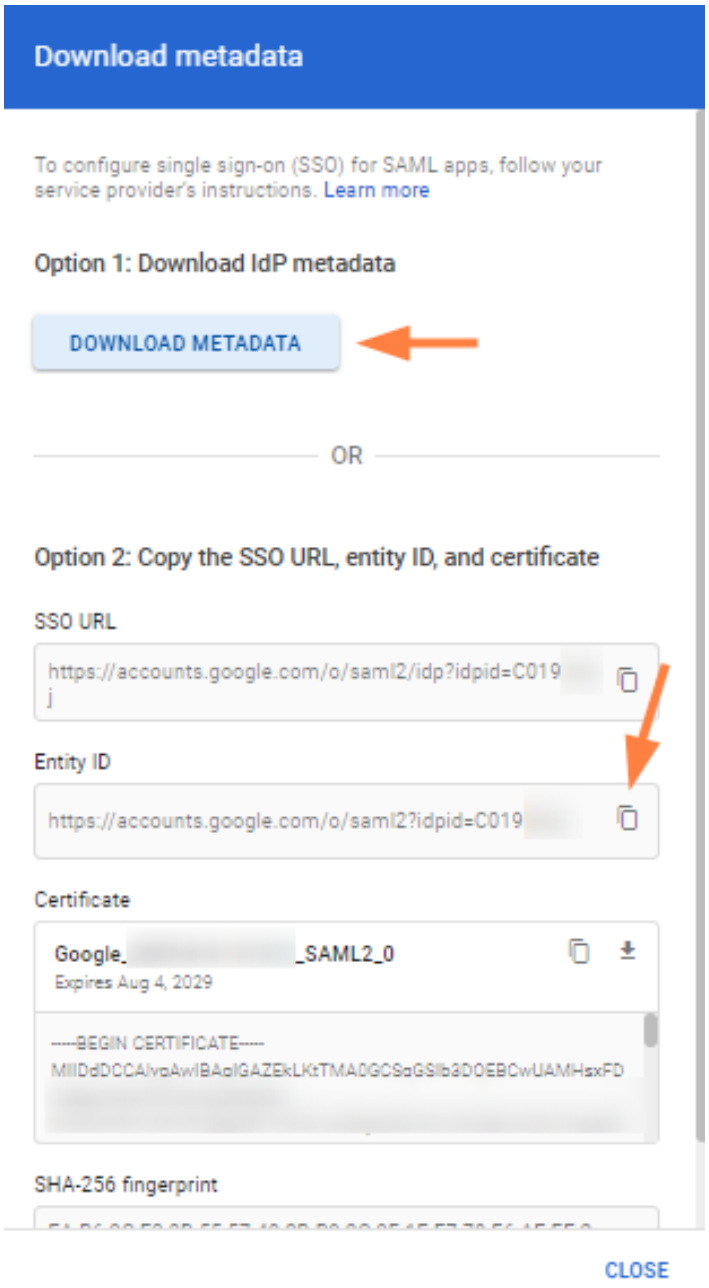
   This indicates to FileCloud which Google groups to sync each time a user imported from Google logs in. However, you must also enable **Sync groups on every login** in FileCloud to activate group syncing on login.

10. Click **FINISH**.

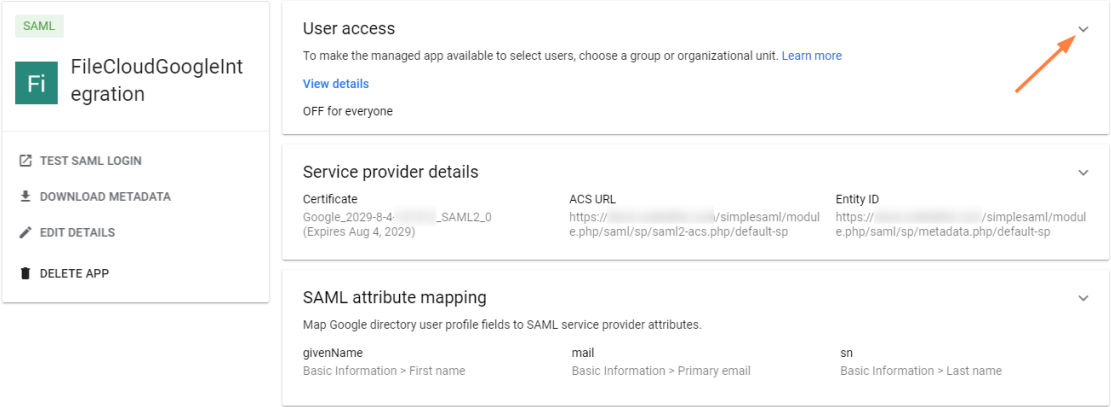
11. Click **DOWNLOAD METADATA**.



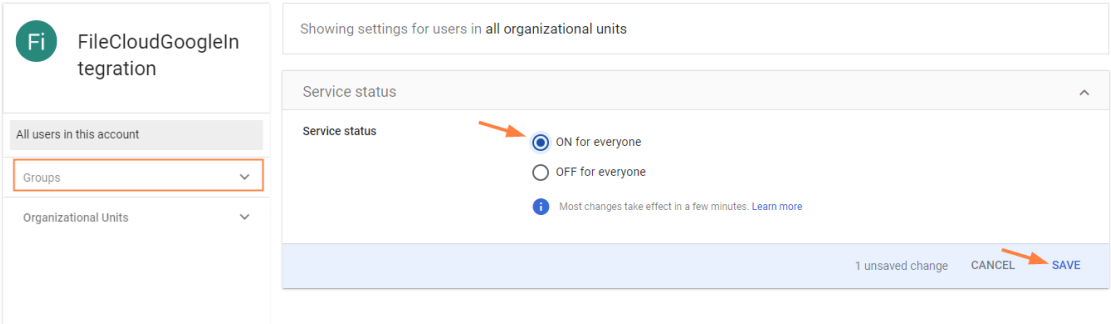
12. In the **Download metadata** popup, click **DOWNLOAD METADATA**.
    The file **GoogleIDPMetadata.xml** is automatically downloaded.
13. Click the copy icon next to **Entity ID**, and save it. You will need it to complete your configuration in FileCloud.



14. Click **CLOSE**.
15. Click the down arrow in the User access box.



16. Select **ON for everyone**.
    If you want to only enable this for certain groups, click the **Groups** down arrow and add the groups.



17. Click **SAVE.**

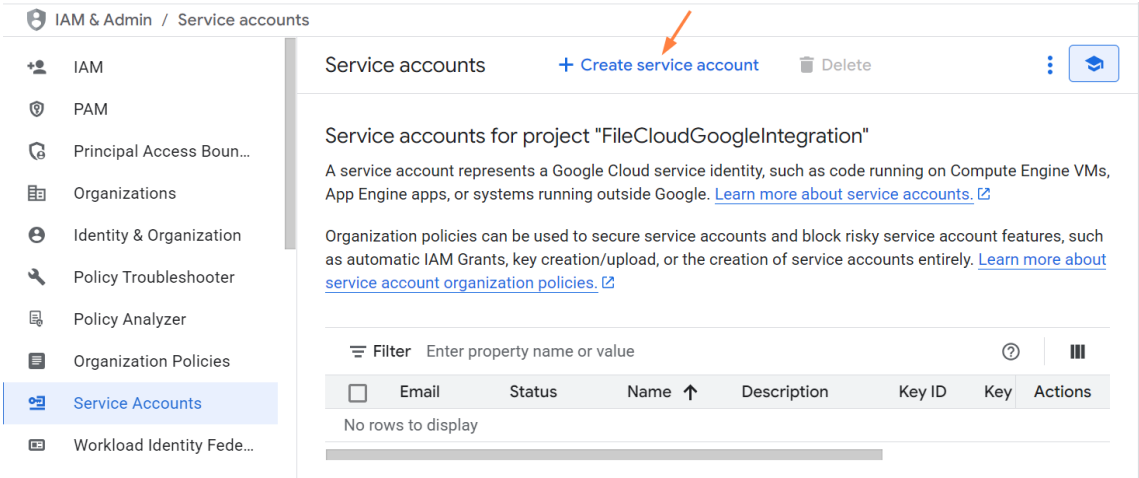
Now, create a key file and grant OAuth scopes.

Create a key file and grant FileCloud access to Google

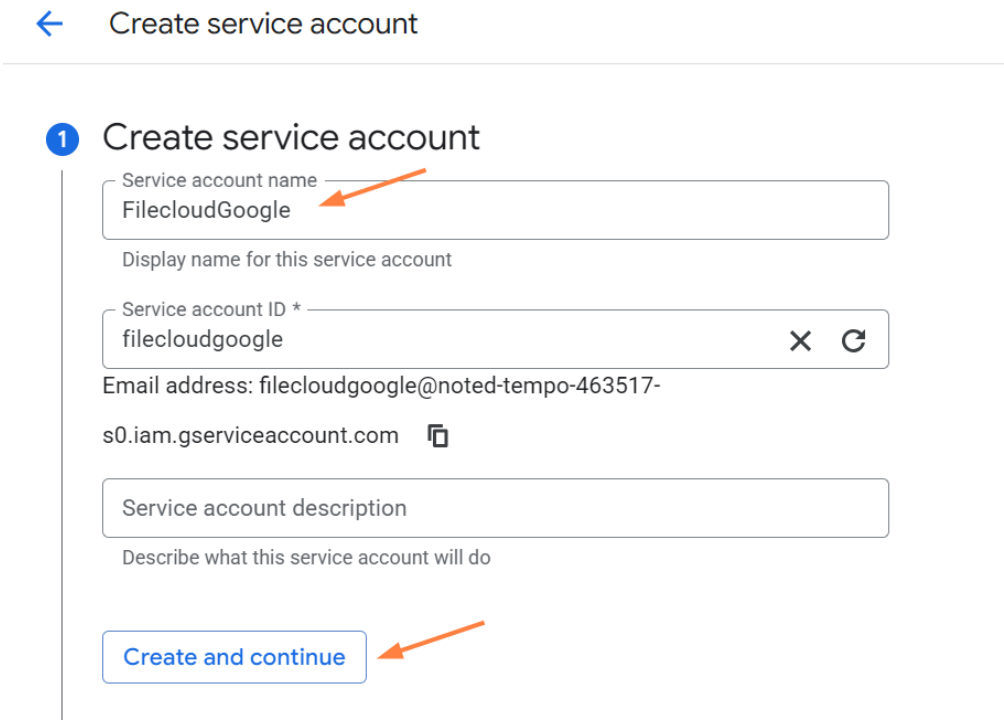
Using your superadmin account, create a service account that grants FileCloud the necessary access to the Google api for SSO authentication. If you do not have a superadmin account, have a superadmin perform the following steps for you.

To create the service account:

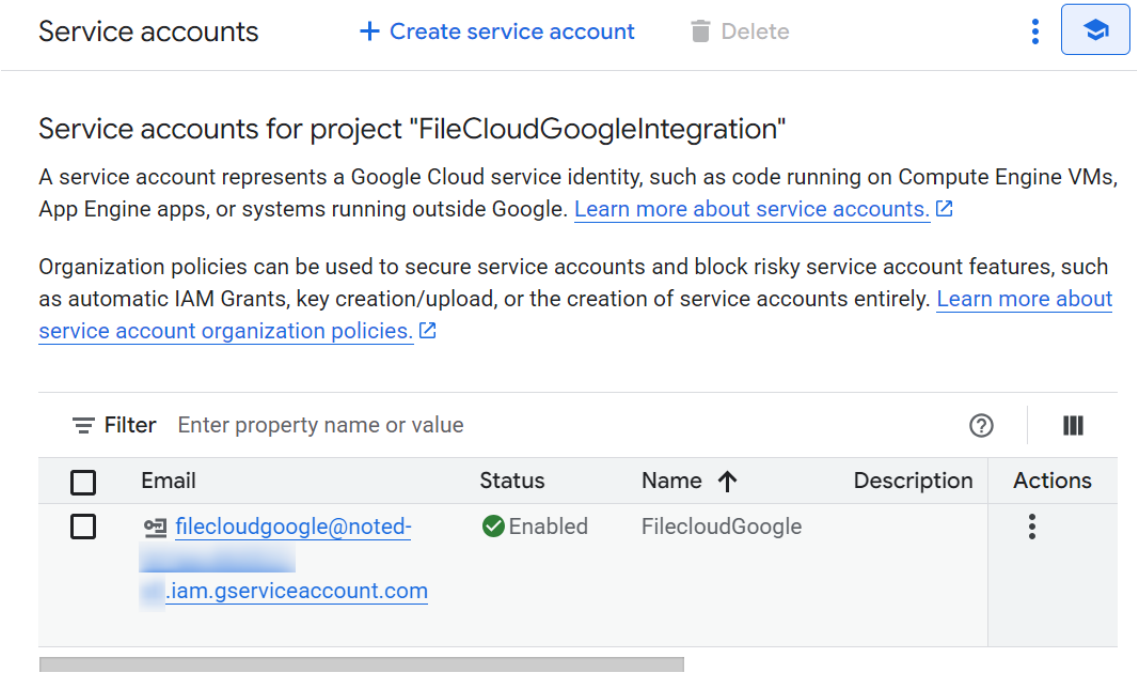
1. Log in to https://console.cloud.google.com/iam-admin/serviceaccounts.
2. Select your project, or create a new one.
3. Click **Create service account**.



4. Enter a **Service account name** and click **Create and continue**.

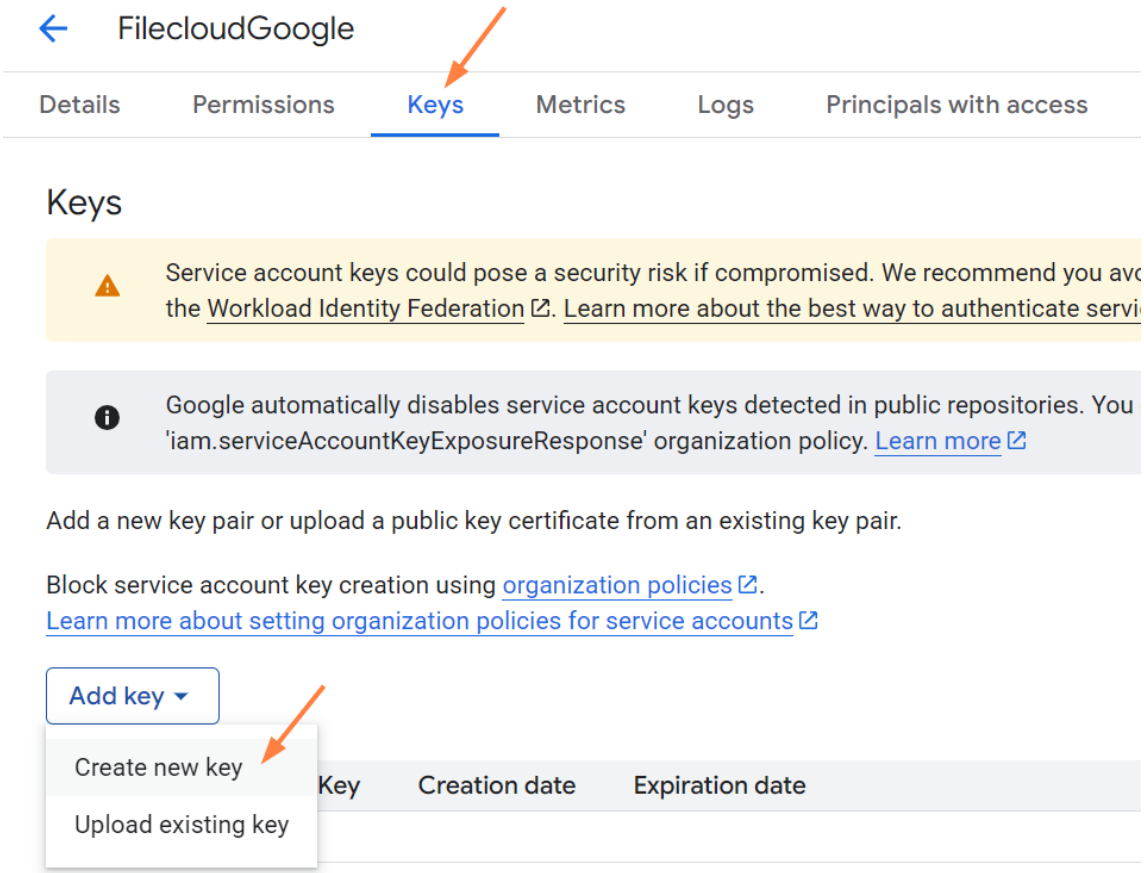


5. Continue through the **Permissions** and the **Principals with access** sections without entering any values, and click **Done**.
   The service account is saved.

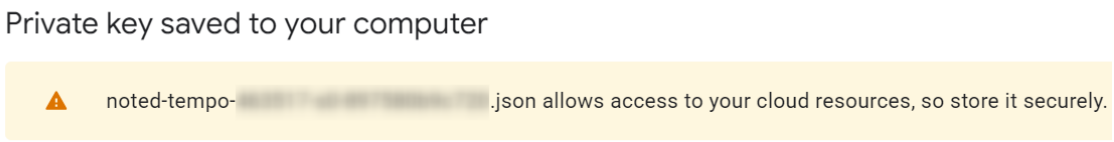


Create and download the private key file

1. On the Service accounts page, click the service account you created.
2. Click the **Keys** tab, and then click **Add key** and choose **Create new key**.



3. Select **JSON**, and click **Create**.
   The private key file is saved in a json file. Note its name so you are able to upload it later when you set up Google/FileCloud SSO integration in FileCloud.

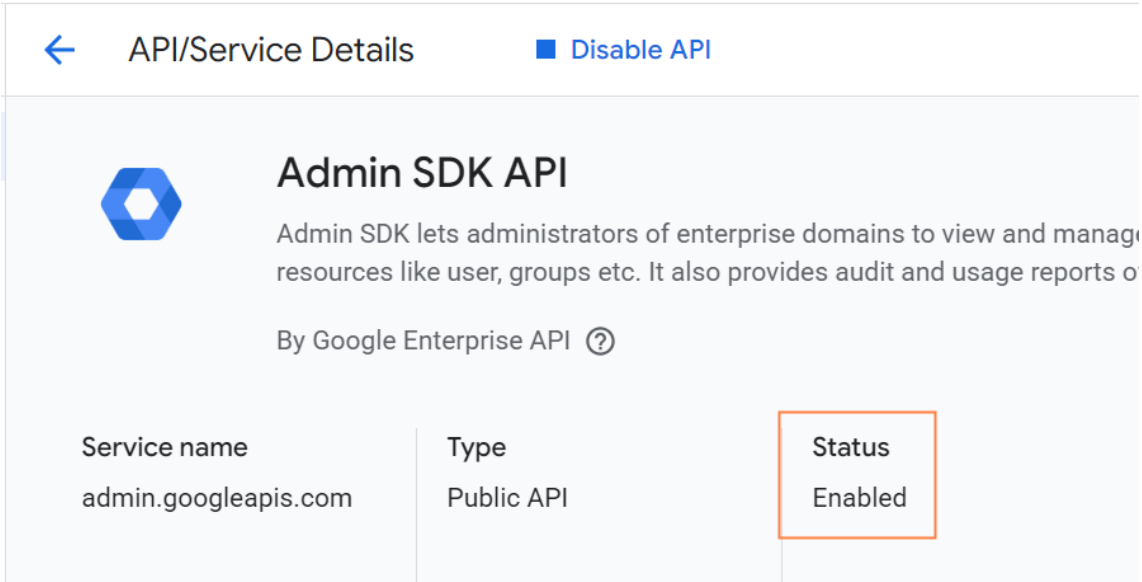


To enter the values into the FileCloud side, see SSO API: Configure Import of SSO Groups and Users.

Enable the Admin SDK API library

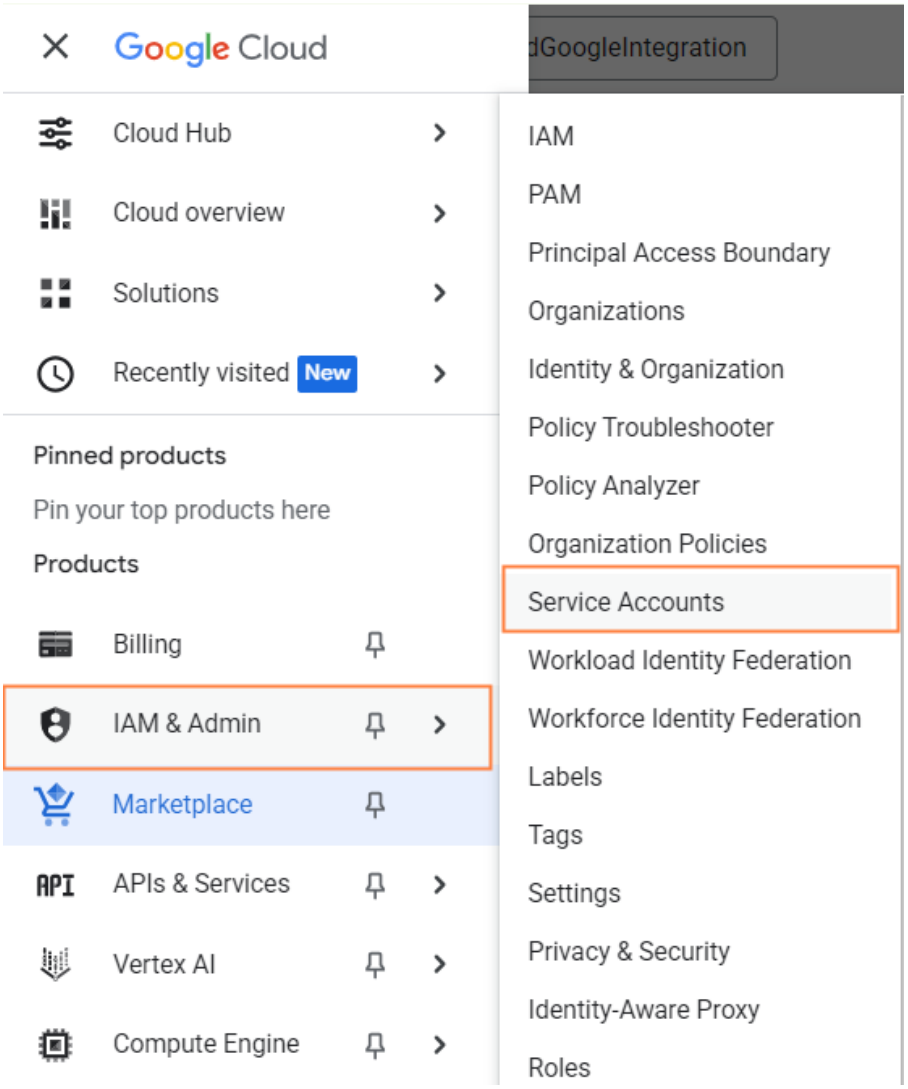
1. Go to https://console.cloud.google.com/apis/library
2. Search for **Admin SDK**.

3. Click it, and then click **Enable**.
   Status should appear as **Enabled**.

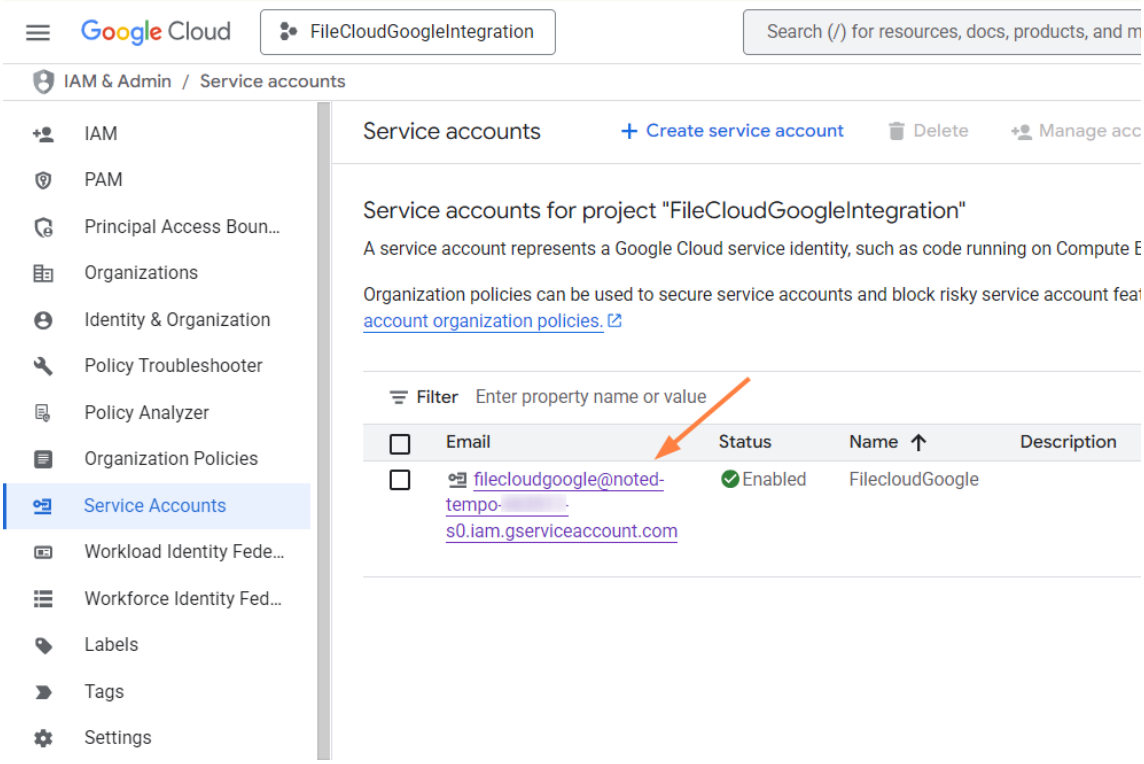


Get the service account Client ID:

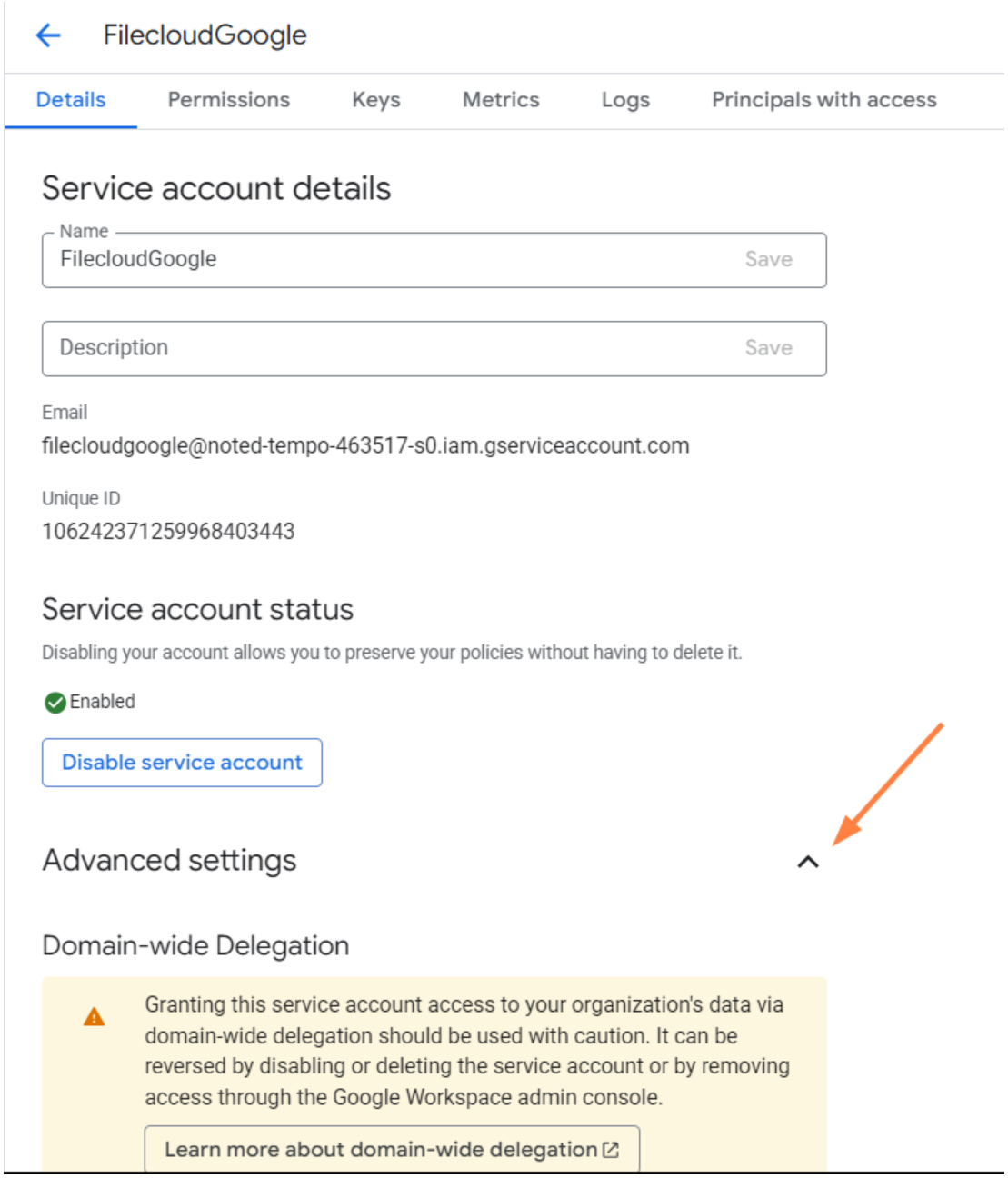
1. In the left navigation pane, go to **IAM & Admin > Service Accounts**.



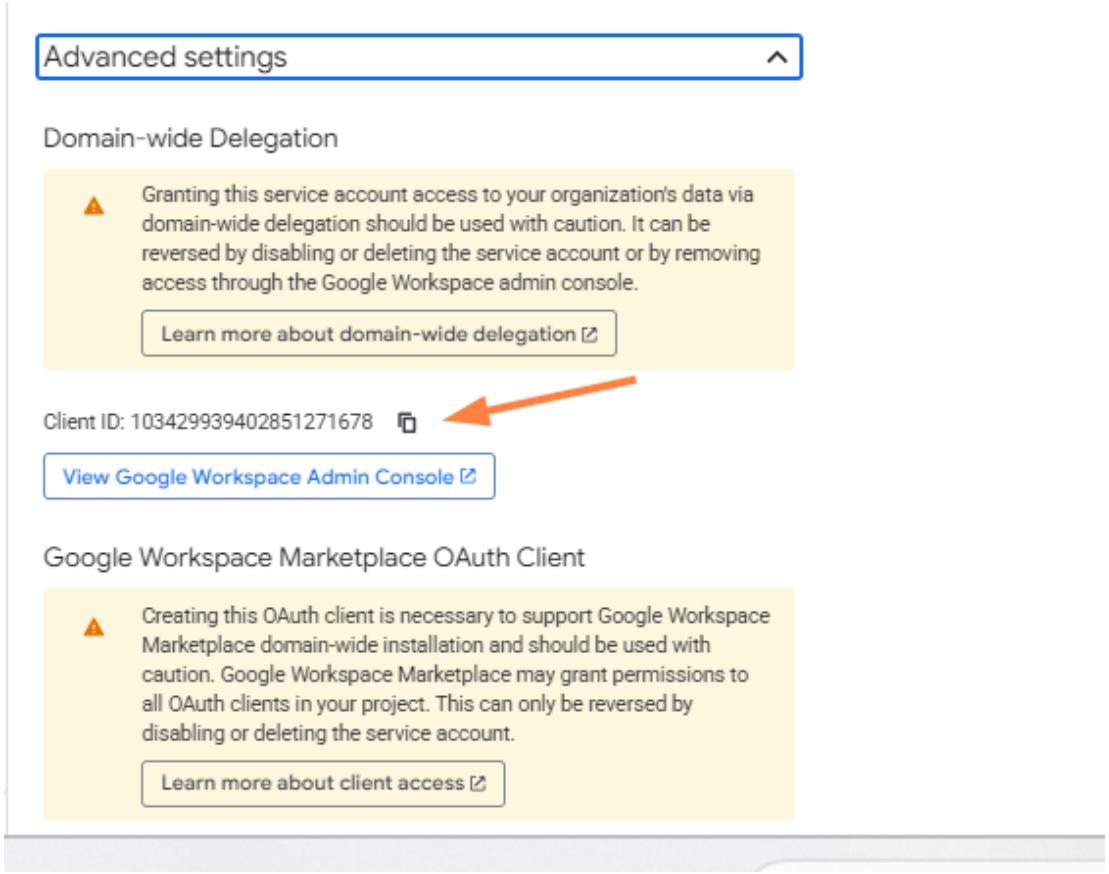
2. Click your service account to open it.



3. At the bottom of the page, click **Advanced settings.**



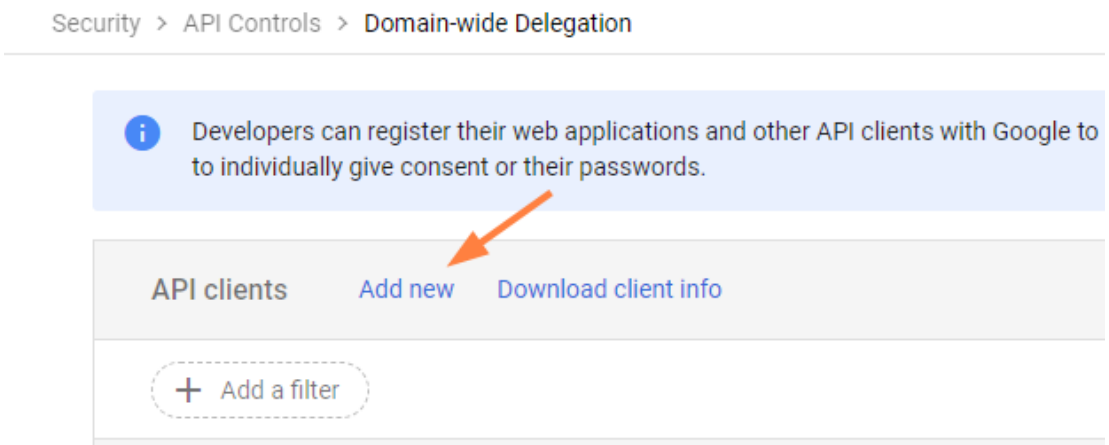
4. Click the copy icon next to **Client ID**, and save it.



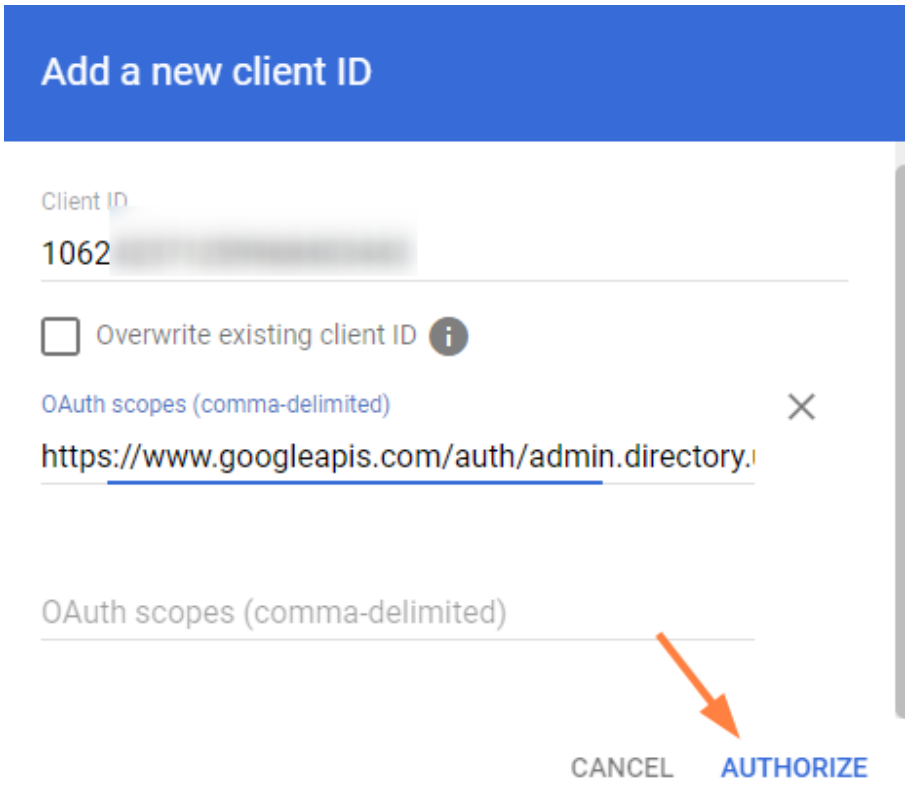
You will paste it into the **Client ID** field in the next section.

Grant OAuth Scopes via the Admin Console

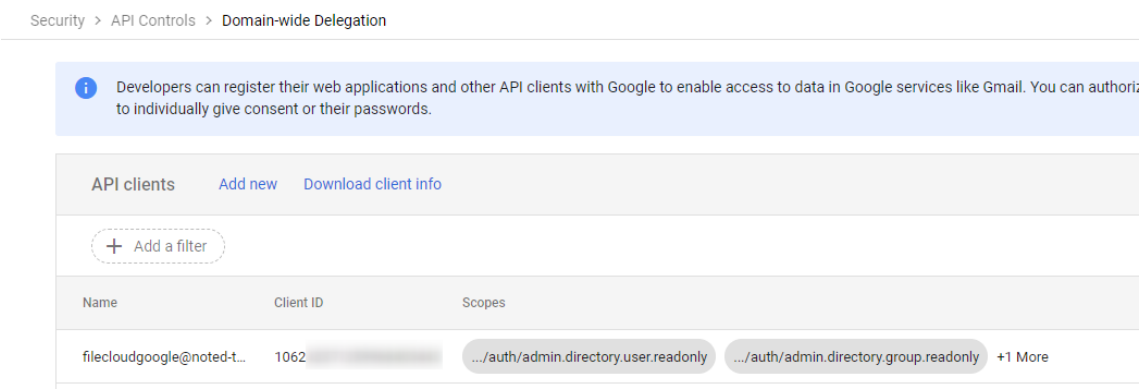
1. Log back into Google Workspace Admin Center and go to https://admin.google.com/ac/owl/domainwidedelegation.
2. Click **Add new**.



3. In the **Add a new client ID** dialog box, and enter the following values:
   Client ID - Enter the Client ID you saved in the previous section from https://console.cloud.google.com.
   OAuth scopes - Enter the following as a string with the commas included:
   https://www.googleapis.com/auth/admin.directory.user.readonly,
   https://www.googleapis.com/auth/admin.directory.group.readonly,
   https://www.googleapis.com/auth/admin.directory.group.member.readonly
4. Click **AUTHORIZE**.



5. The OAuth scopes are now added to the Client ID.



To enter the integration values into the FileCloud side, see SSO API: Configure Import of SSO Groups and Users.

CASB integration

> For security purposes, to initially access the API, you must now change the default API key. If you do not change it, when you enter a command to call the API, an error is returned.
> **Note**: You are only required to change the default API key initially; after that, you can continue to use the new key you entered.

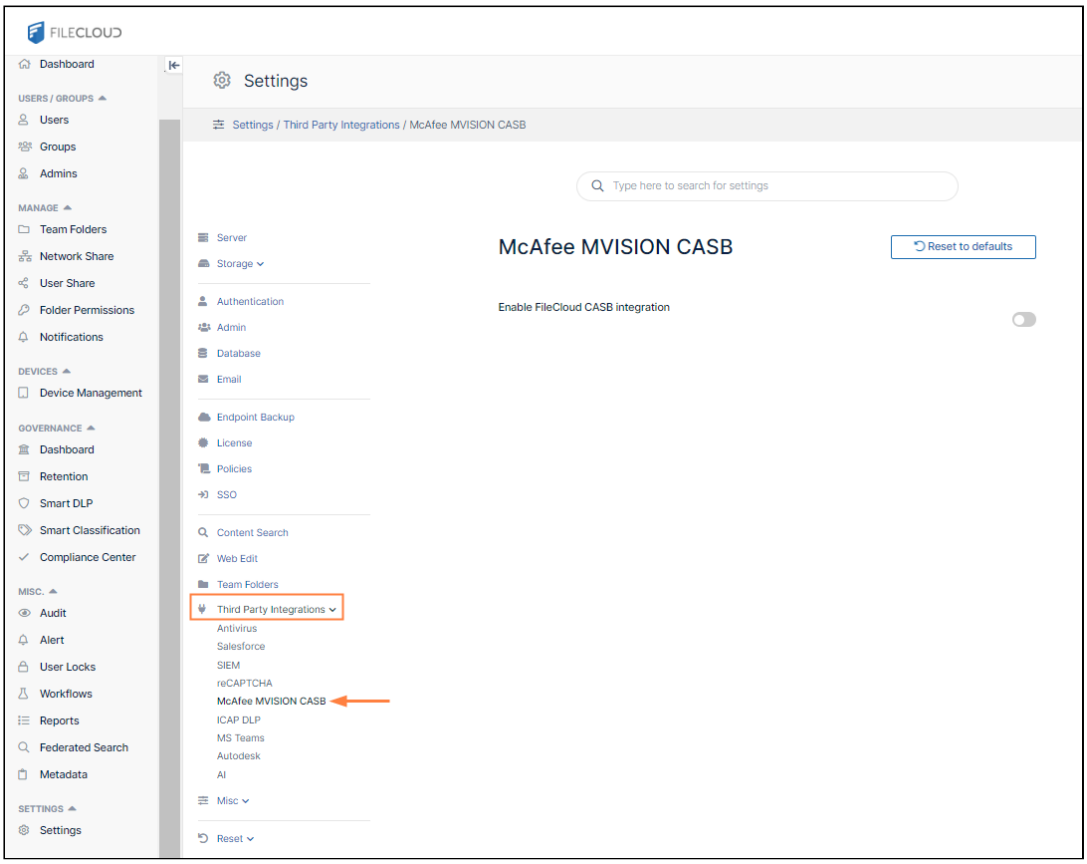
FileCloud includes a smart data leak prevention (DLP) functionality that monitors user actions and and prevents them if they pose a security risk.

In Version 20.2, FileCloud has added integration with external cloud access security broker (CASB) software to enable you to expand your DLP monitoring and risk prevention. This enables you to expand activity monitoring and measures taken when there is a possible security breach.

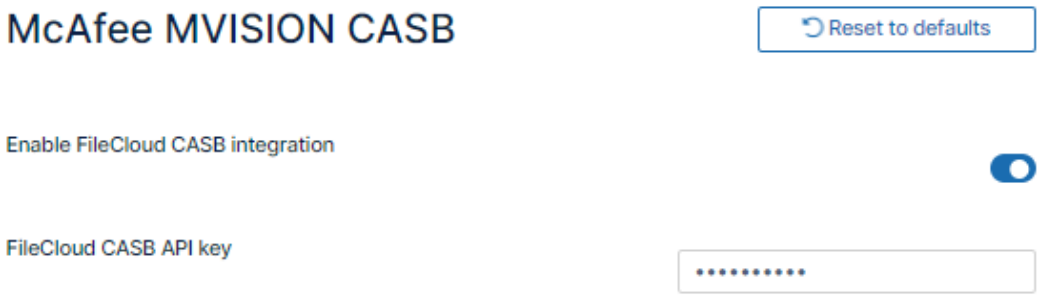
Currently, FileCloud supports integration with McAfee CASB software.

To enable CASB integration with FileCloud:

1. Open the McAfee MVISION CASB settings page.
   To go to the McAfee MVISION CASB settings page
   a. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the Settings navigation page, click Third Party Integrations .
   b. In the inner navigation bar on the left of the Third Party Integrations page, expand the **Third Party Integrations** menu, and click **McAfee MVISION CASB**, as shown below.



2. Check **Enable FileCloud CASB** Integration.
   The field **FileCloud CASB API Key** appears.



3. Change the value of **FileCloud CASB API key** to any alphanumeric string.
4. Click **Save**.
5. Add the value of the **FileCloud CASB API key** to McAfee MVISION CASB. See McAfee's product documentation for instructions.

McAfee CASB integration

McAfee CASB integration

Beginning with version 20.2, FileCloud supports integration with McAfee CASB.

This enables you to use McAfee CASB to apply extensive DLP rules when monitoring user events such as actions on files and folders and logins to the system. If a CASB DLP rule is violated, McAfee takes actions such as notifying a user, deleting a file, or removing a share.

For example, you could set up McAfee CASB to monitor the content of files when they are shared in a public FileCloud folder.

McAfee CASB supported features

User Activity	File Upload, File Update, File Download, File has been Shared publicly, Folder has been shared publicly
	User logged in
DLP Features	Content- aware Public Shared Link, or Pure Public Shared link Policy evaluation for Item Shared event
	Content-ware Policy evaluation for File Upload/Update event
	Response Actions: Incident Remove Shared link Email notification Send user notification Delete

FileCloud events and McAfee responses

To receive information about events, McAfee registers a webhook with FileCloud, which enables FileCloud to push information about events as they occur to McAfee CASB.

FileCloud pushes information to McAfee when a user performs one of the following actions:

- adds a file
- updates a file
- adds an external file
- downloads a file
- logs in successfully
- creates a share
- creates an account
- deletes an account

McAfee responds to events that may compromise security using FileCloud's API. FileCloud's API includes the following endpoints:

- register
- deregister
- getwebhook
- downloadfile
- upload
- deletefile
- getshareinformation
- removeuserfromshare
- removegroupfromshare
- deleteshare
- getuserinformation

For more information about using these APIs, see the API documentation at https://fcapi.getfilecloud.com/

ICAP DLP

> The ability to configure ICAP DLP as a provider for FileCloud's CCE is available in Version 20.3 and higher.

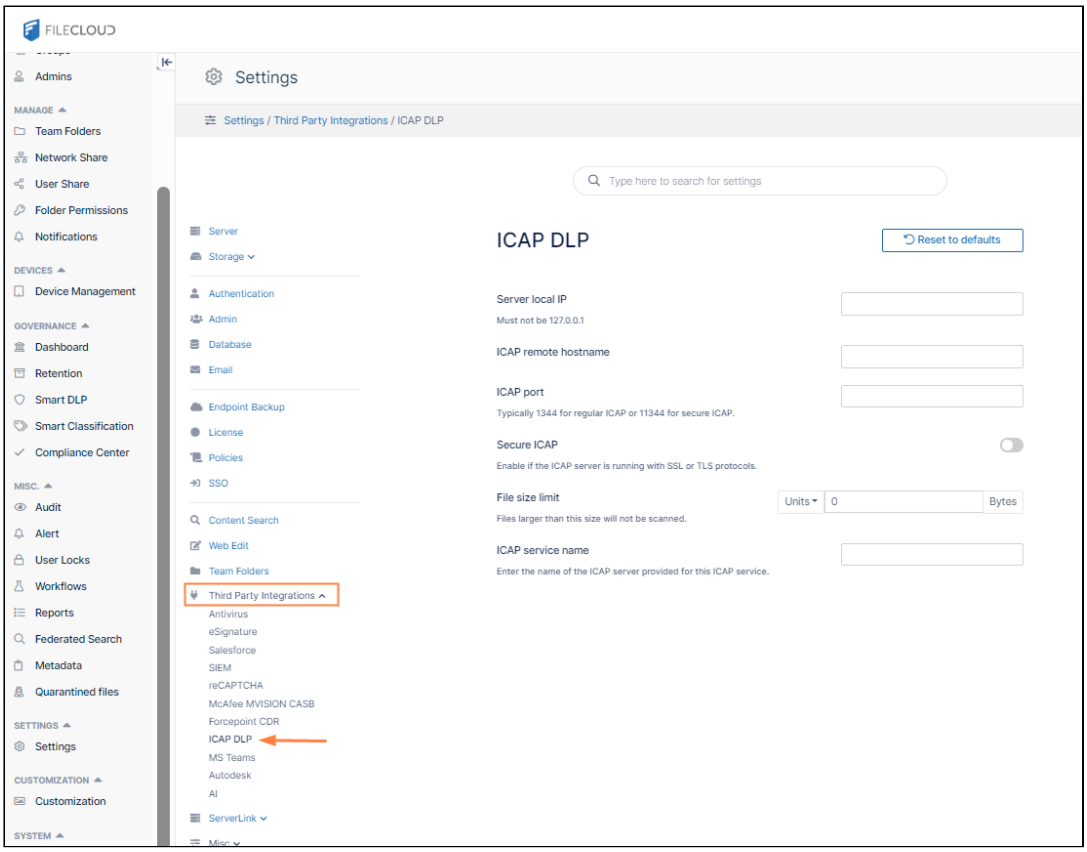
ICAP DLP has been added as a provider for FileCloud's Smart Classification (CCE), enabling you set up a content classification rule that flags files for blocking or deletion by DLP rules. You must configure it as a third-party provider in FileCloud to use it with the CCE.

What is ICAP?

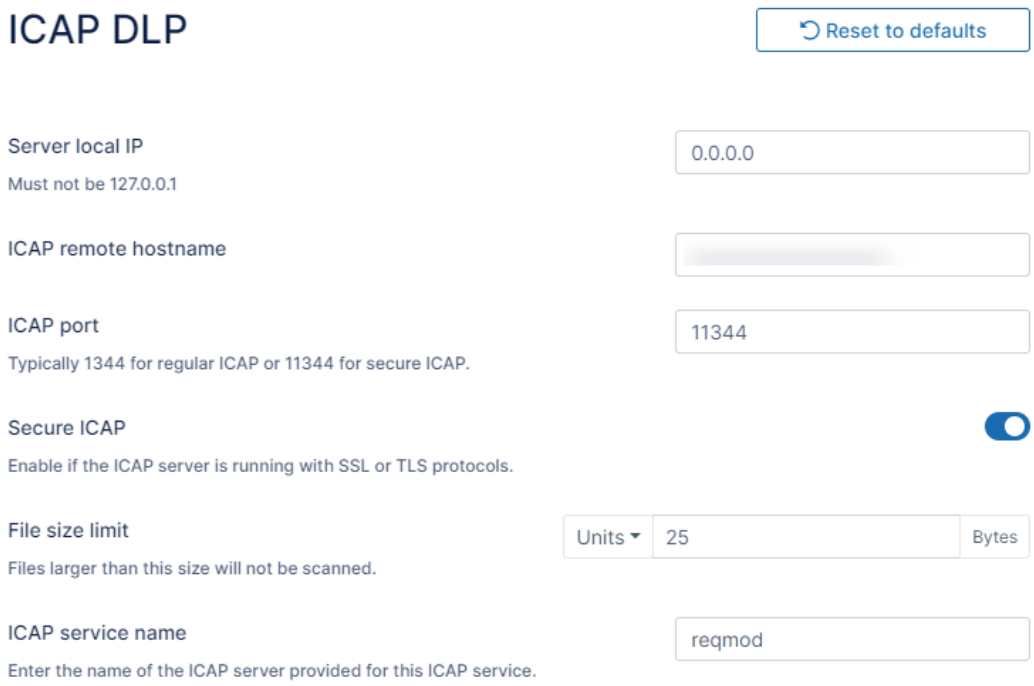
ICAP is a generic protocol that allows web servers to offload specialized tasks to custom-built servers. Examples of such specialized tasks include DLP (data loss prevention) based content scanning, URL filtering and antivirus scanning.

Integrating ICAP DLP with FileCloud

1. Open the ICAP DLP settings page.
   To go to the ICAP DLP settings page
   a. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the Settings navigation page, click Third Party Integrations .
   b. In the inner navigation bar on the left of the Third Party Integrations page, expand the **Third Party Integrations** menu, and click **ICAP DLP** as shown below.



2. Fill in the fields. See the table below for information.



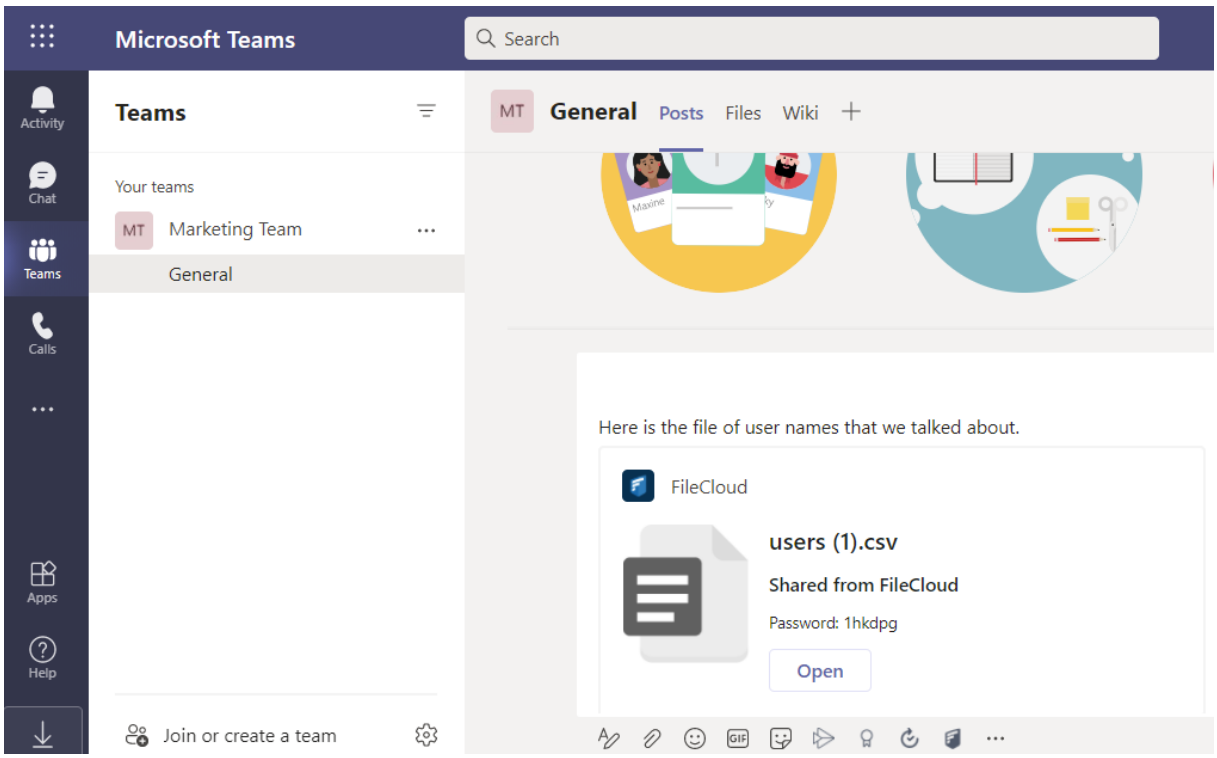
Setting	Description
Server Local IP	In most cases, enter the value **0.0.0.0**. If you are using a separate FileCloud policy with ICAP, enter the Private (LAN) IP of the FileCloud server.
ICAP Remote Hostname	Enter the hostname or IP of the system where the ICAP DLP is deployed.
ICAP Port	Leave the default value of 1344 or use 11344 for secure ICAP. In rare cases, this might need to be changed to whatever port the ICAP DLP server is listening on.
Secure ICAP	Enable if the ICAP server is running with SSL or TLS protocols.
File Size Limit	To exclude very large files from scanning, specify the file size limit in bytes. Default value is 25MB.
ICAP Service Name	Consult the ICAP DLP server product documentation for this value. It must be set correctly; otherwise, integration won't work.

3. Click **Save**.

   After you have configured its settings in FileCloud, you can use ICAP DLP with FileCloud Smart Classification to set metadata values.

Microsoft Teams

FileCloud can be configured to function within MS Teams so users can share content in Team's chats and channels.



To set up integration:

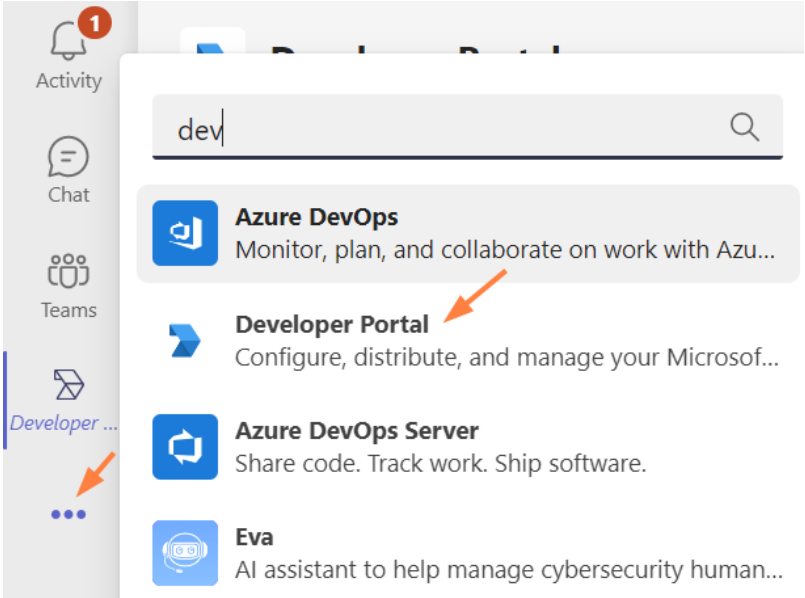
1. The Teams administrator must create a FileCloud app.
2. The FileCloud administrator must enable Teams integration in FileCloud.
3. Then, FileCloud users can add the FileCloud app to their Teams installations in order to share FileCloud content in messages and view the FileCloud browser while working in Teams.

For MS Teams Admins: Configuring FileCloud in Teams

Before users can access FileCloud through MS Teams, the Teams administrator must perform the following configuration in Teams. After that, the FileCloud Admin must Enable FileCloud/Teams integration in the FileCloud Admin portal.

1. Confirm that you have FileCloud Version 21.2 or higher installed.
2. Create an MS Teams bot in the Teams' **Developer Portal**:
    a. Open **MS Teams**.

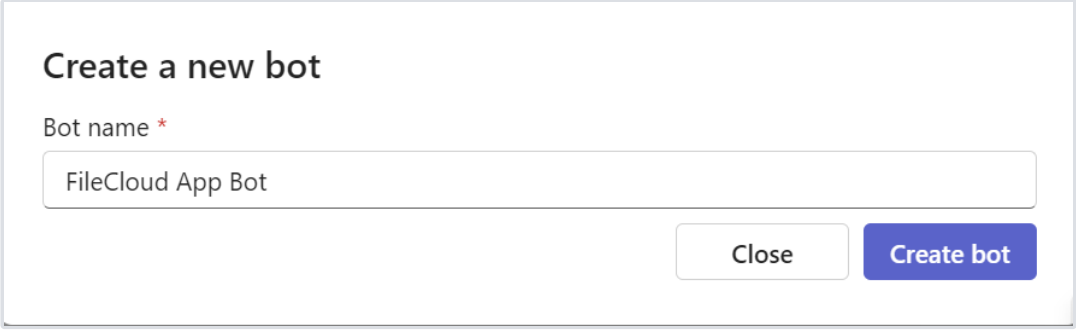
b. If you do not have the **Developer Portal** app installed already, click the **More** icon in the navigation pane, search for **Developer Portal**, and add it.



c. Click the **Developer Portal** icon in the navigation pane, and go to **Tools > Bot Management**.

d. Click **New Bot**.

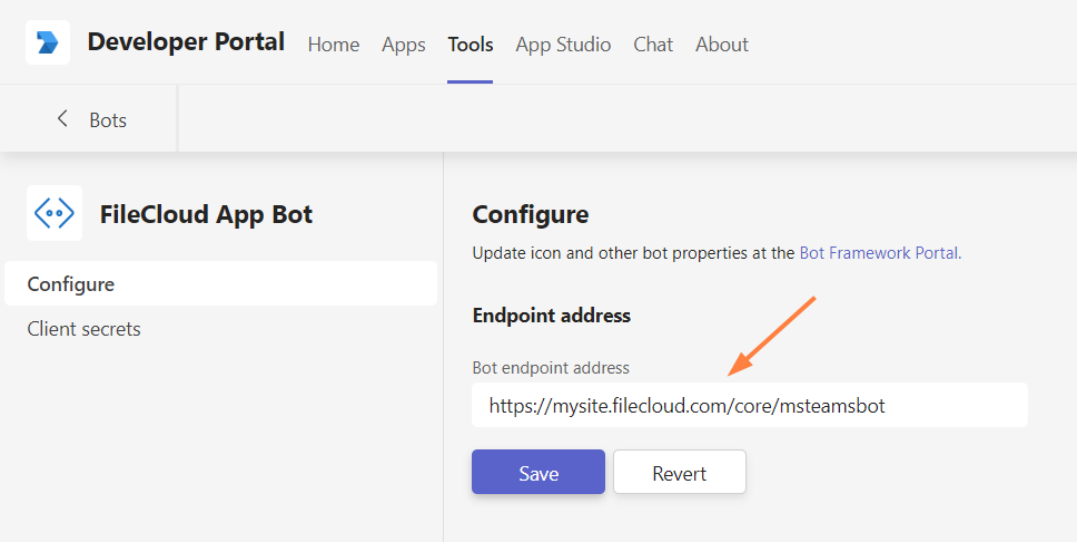
e. Name the bot, and click **Create Bot**.



The bot appears opened on the **Bot management** screen.

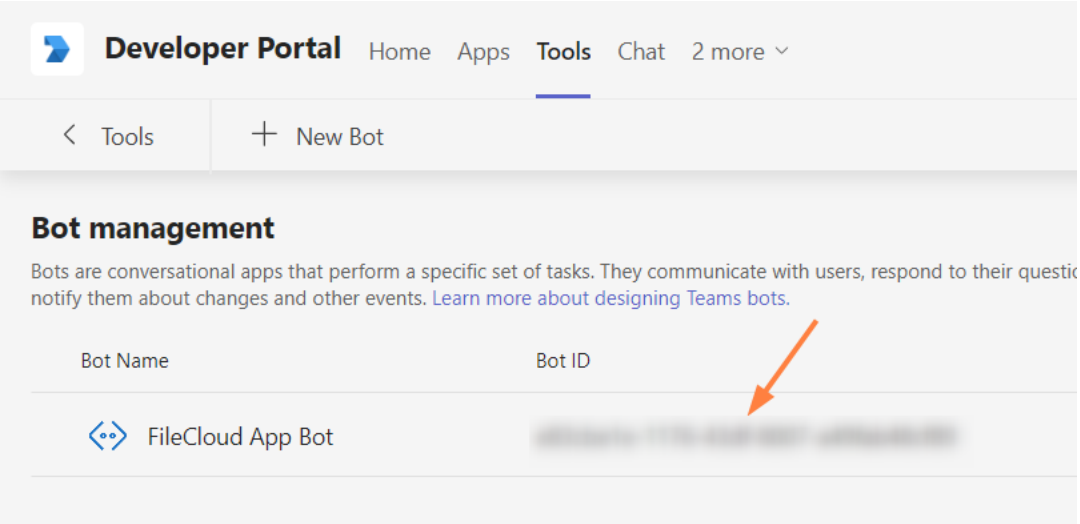
f. Click the bot to open it.

g. Change the **Endpoint address** to point to the bot in your FileCloud server, and click **Save**. Use **https://[your FileCloud server]/core/msteamsbot**



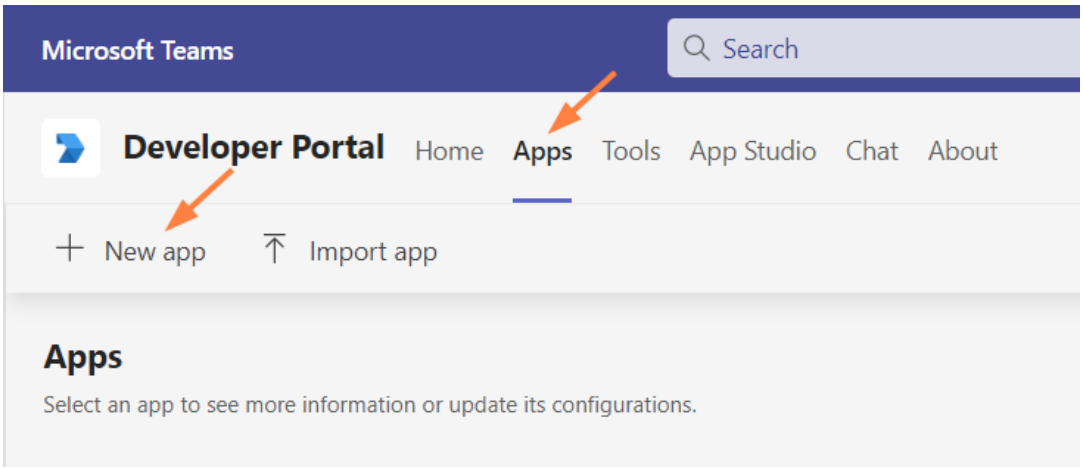
h. Return to the **Bot Management** screen.

i. Copy the **Bot ID**. You will need it to set up MS Teams integration in the FileCloud admin portal.



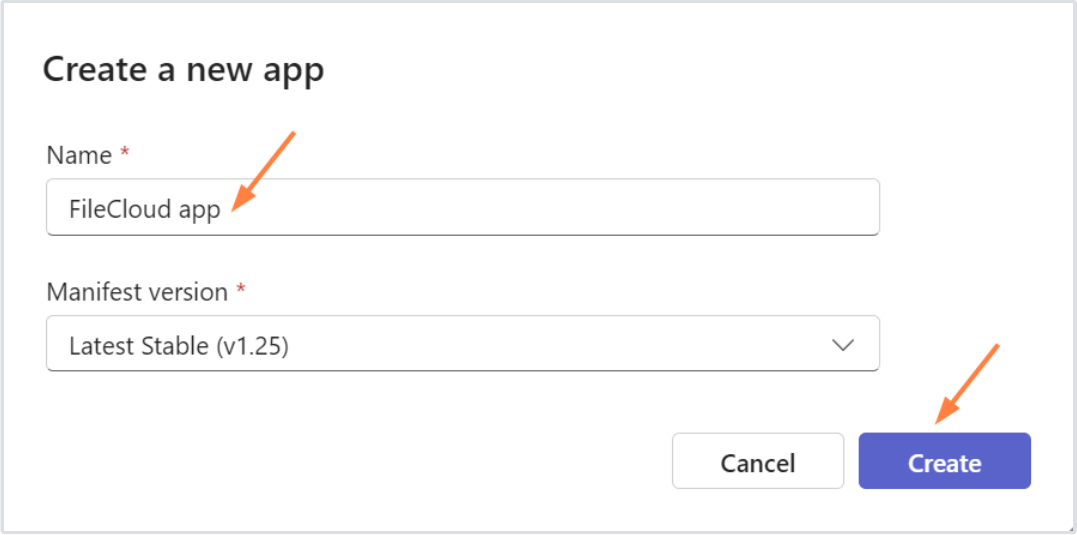
3. Create the MS Teams application in Teams' **Developer Portal**.

   a. In the **Developer Portal**, click the **Apps** tab, and then click **New App**.



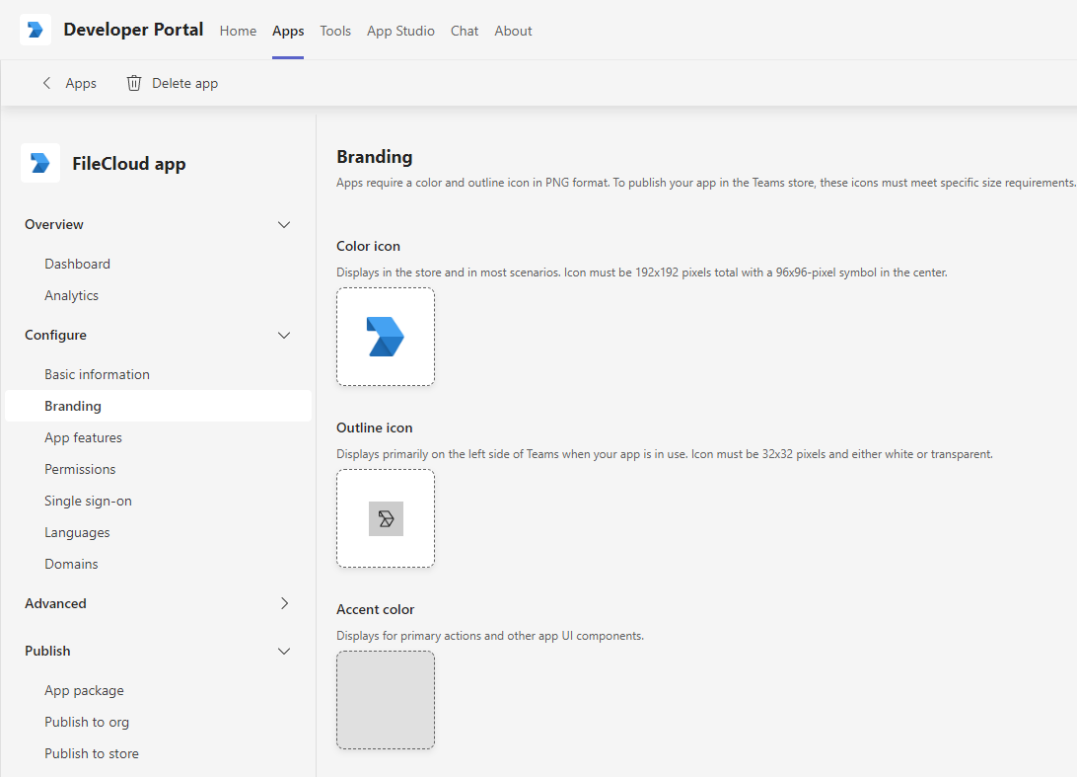
An **Add App** window opens.

b. Enter a name for your FileCloud app and click **Create**.



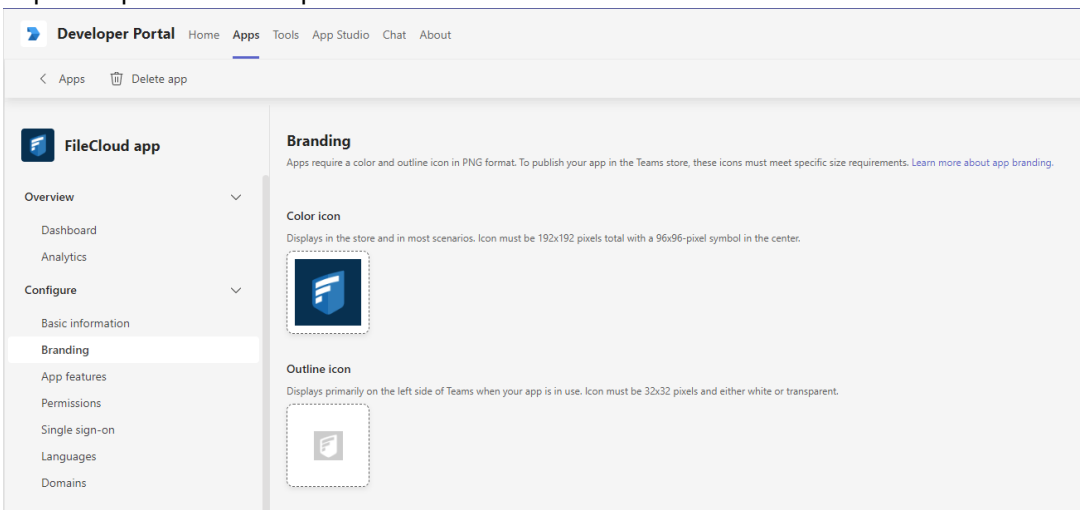
The **Dashboard** screen for the app opens.

c. In the navigation pane, click **Branding**.
The **Branding** screen opens.

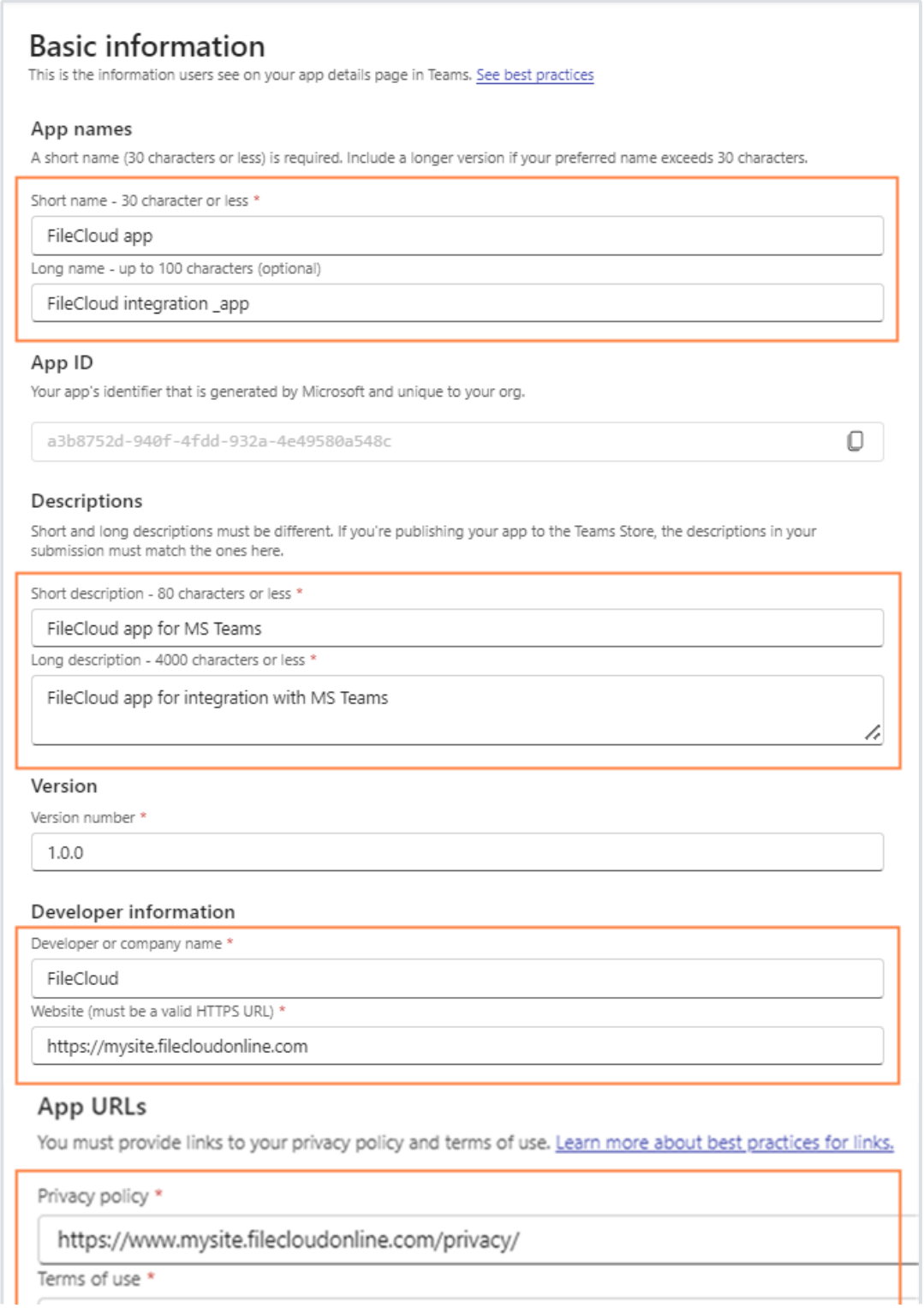


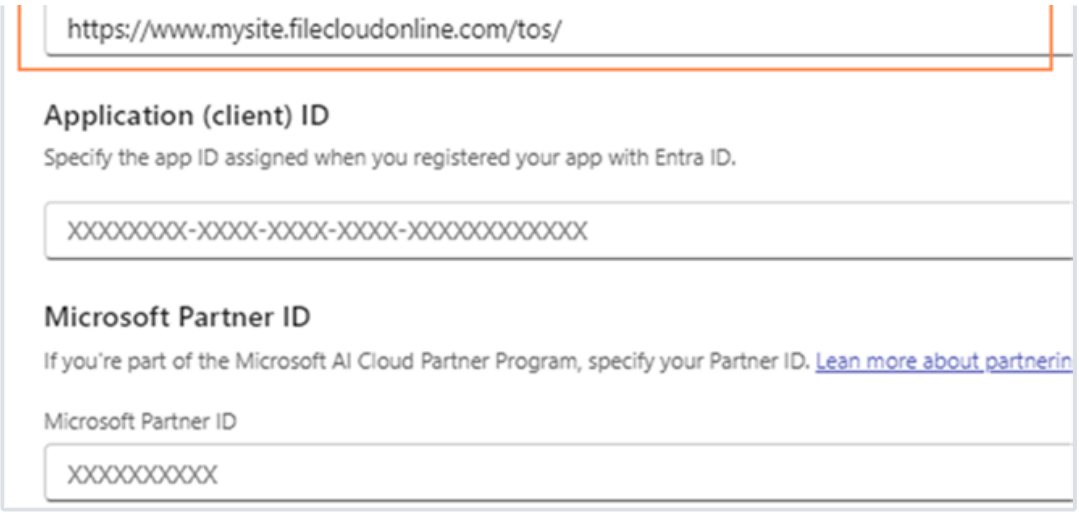
d. Download the following two images (right-click and choose **Save image as**).

e. Upload the first image for **Color icon,** and the second image for **Outline icon**.
You may use custom images, but they must be 192px X 192px for the color image and 32px X 32px for the transparent outline.



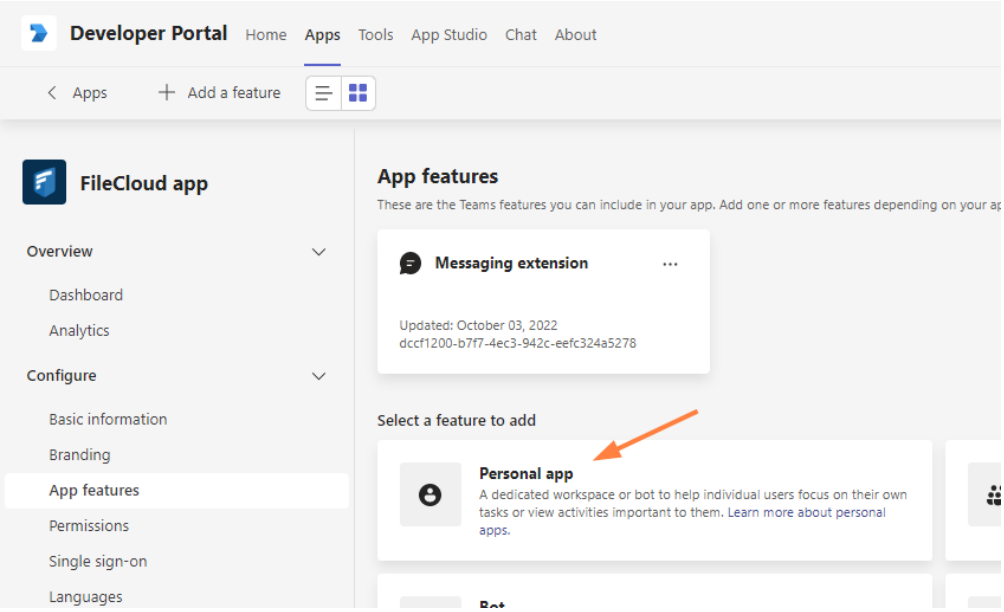
4. Make sure the required fields in the **Basic information** page are entered, and add a tab to your personal app or you may run into errors when you edit the manifest.json file in the following steps.
   a. In the navigation pane, click **Basic information.**
   b. Enter any of the circled fields that are not filled in.



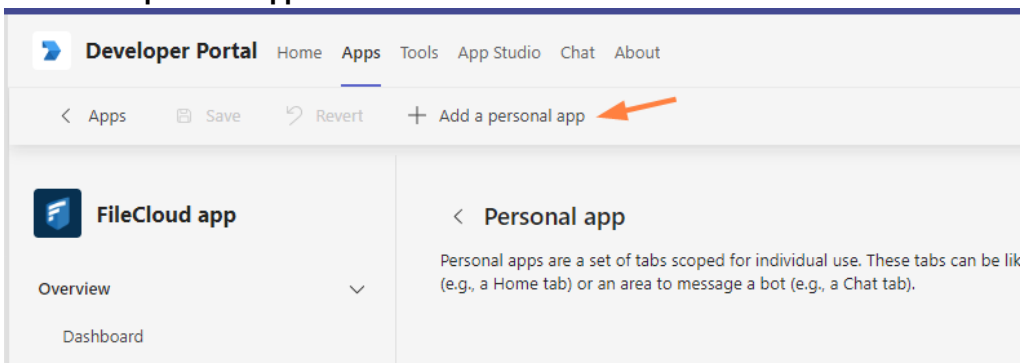


c. Click **Save**.

d. In the navigation pane, click **App Features**, and click **Personal app**.

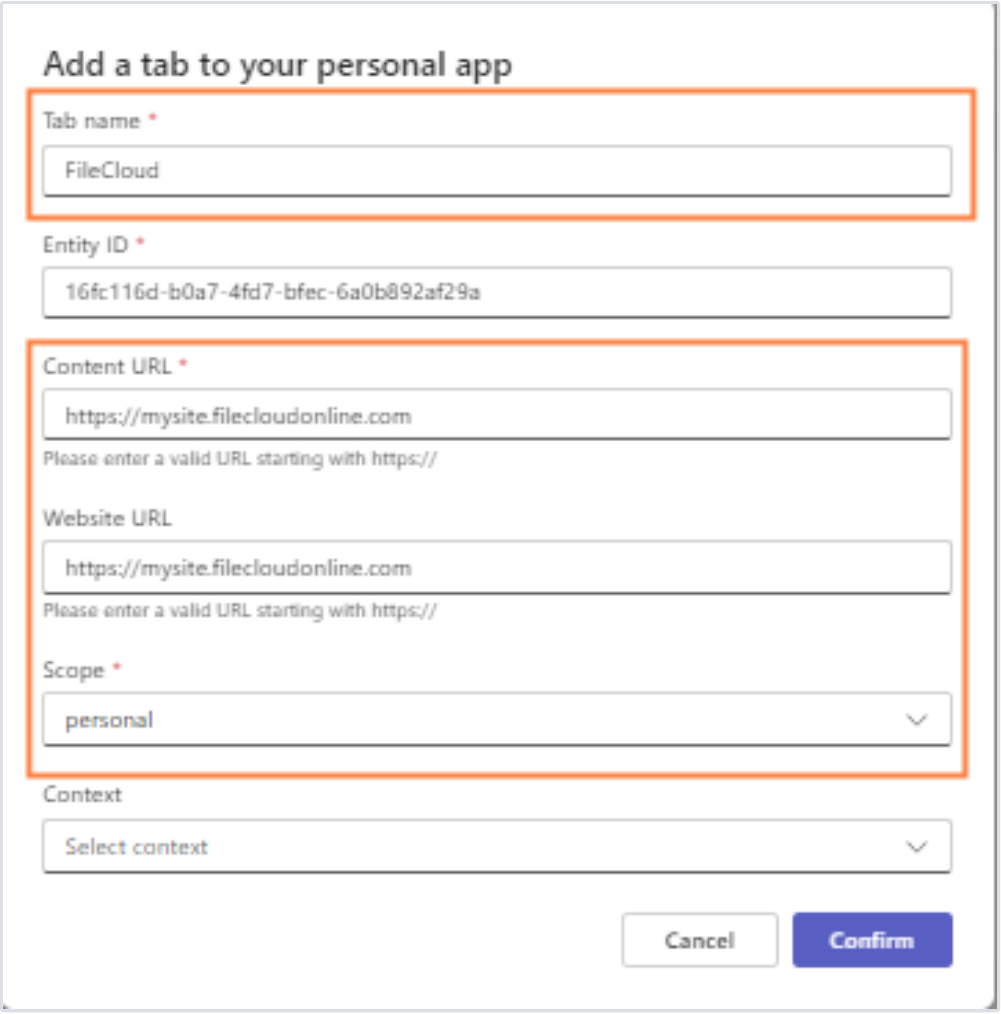


i. Click **Add a personal app**.

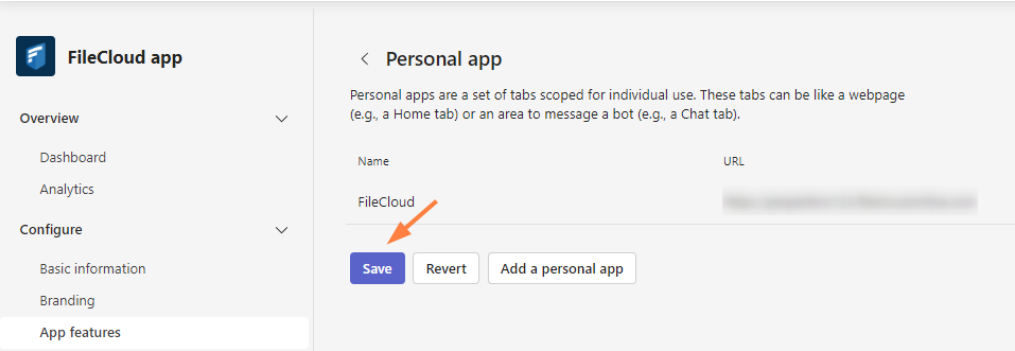


The **Add a tab to your personal app** dialog box opens.

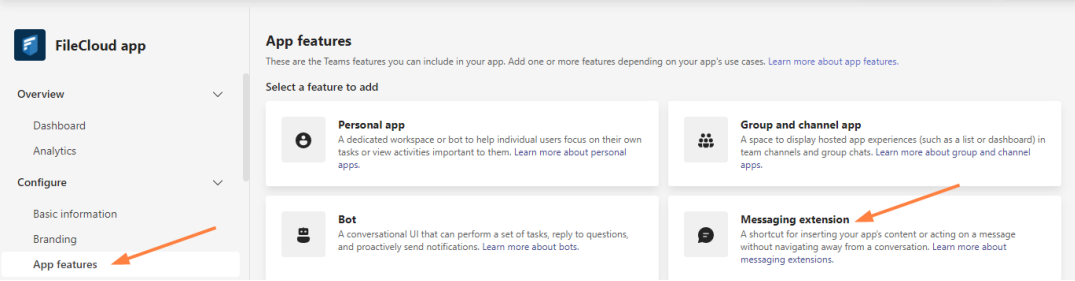
ii. Fill in the required fields as follows. Your **Entity ID** will be entered for you.



iii. Click **Confirm**.
In the **Personal app** screen, click **Save**.



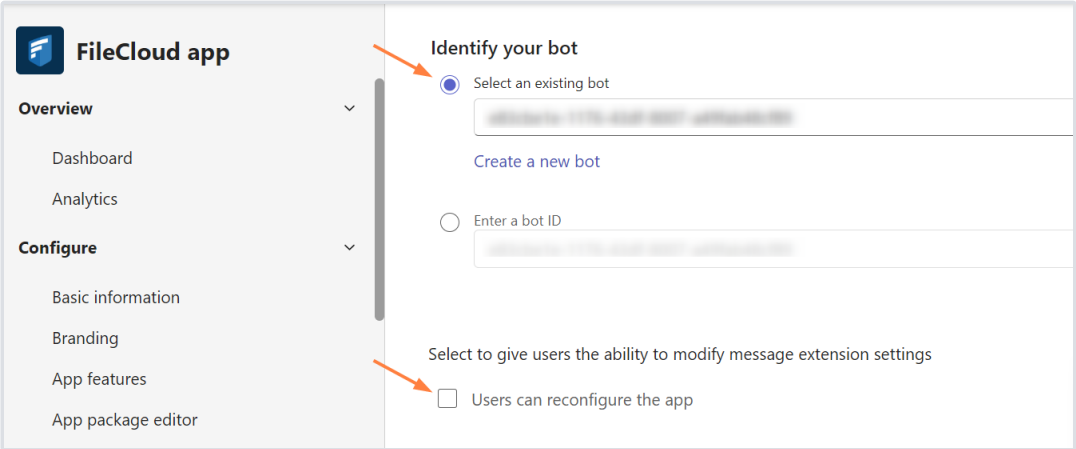
5. Set up your MS Teams bot.
   a. In the navigation pane, click **App Features**, and then click **Messaging Extension**.



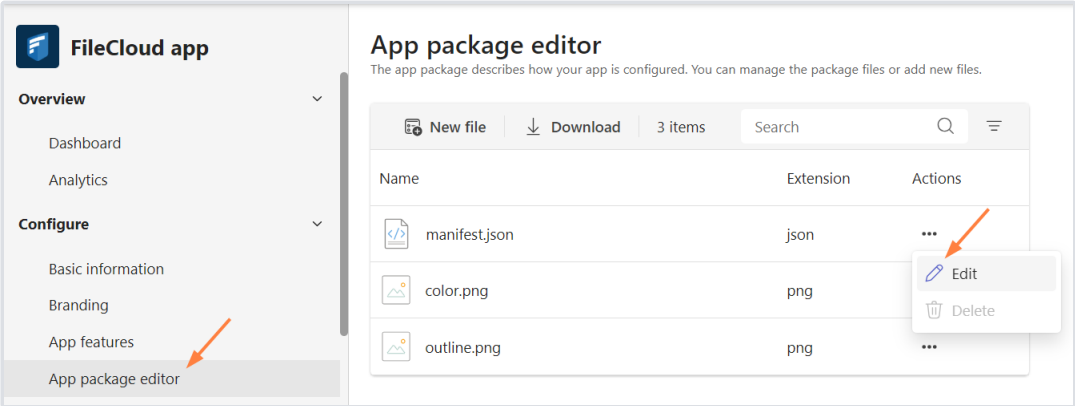
The **Messaging Extension** screen opens.

b. Choose **Select an existing bot**, and select the FileCloud bot that you just created.

c. Uncheck **Users can reconfigure app**.



d. Click **App package editor** in the left panel, and beside the **manifest.json** file, click the more icon and choose **Edit**.

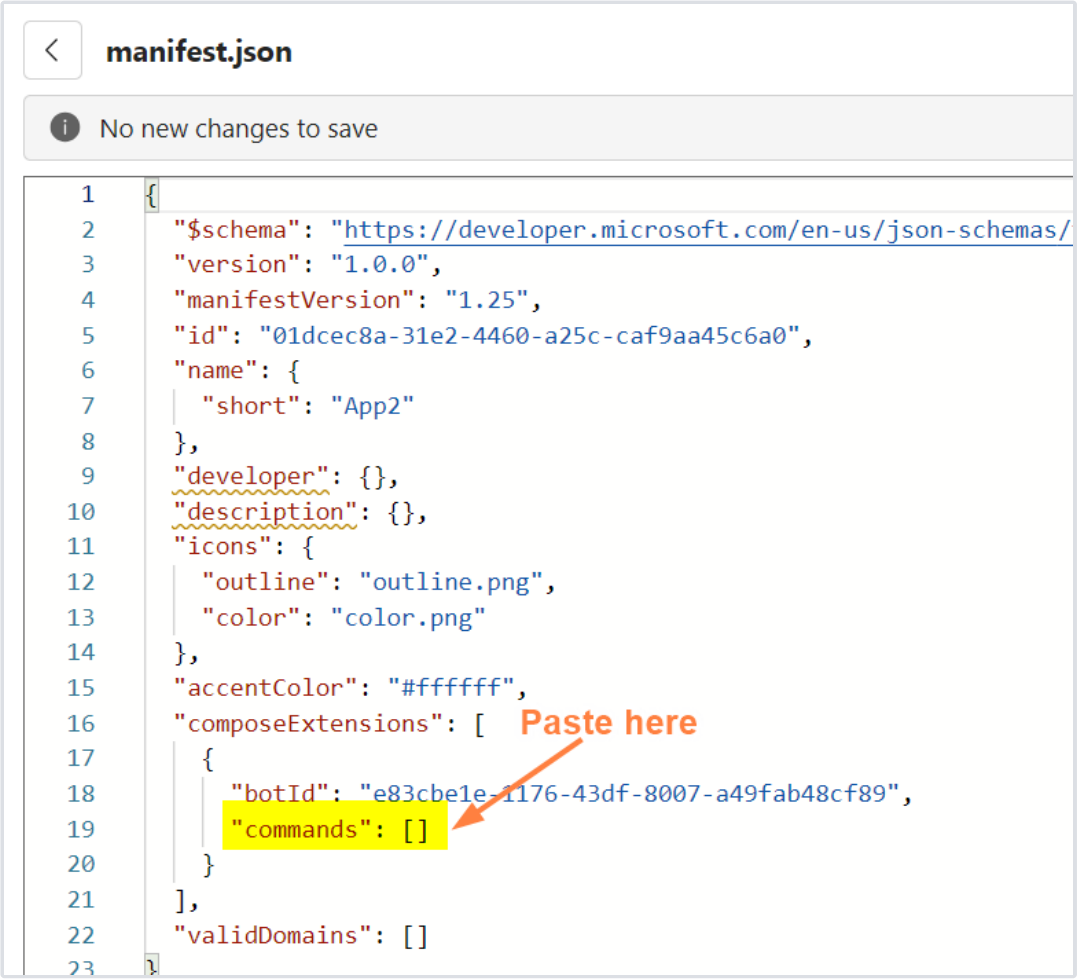


e. Copy the following code; in “url” change [your FileCloud server] to your FileCloud server’s url.

```
"commands": [
        {
          "id": "FileCloud",
          "type": "action",
          "title": "FileCloud",
          "description": "Share from FileCloud",
```

```
            "initialRun": false,
            "fetchTask": true,
            "context": [
              "commandBox",
              "compose",
              "message"
            ],
            "taskInfo": {
              "title": "Share from FileCloud",
              "width": "medium",
              "height": "medium",
              "url": "https://[your FileCloud server].com/core/msteamsbot"
            }
          }
        ]
        "messageHandlers": [
          {
            "type": "link",
            "value": {
              "domains": [
                "[your filecloud domain].com"
              ],
              "supportsAnonymizedPayloads": false
            }
          }
        ]
      ]
```

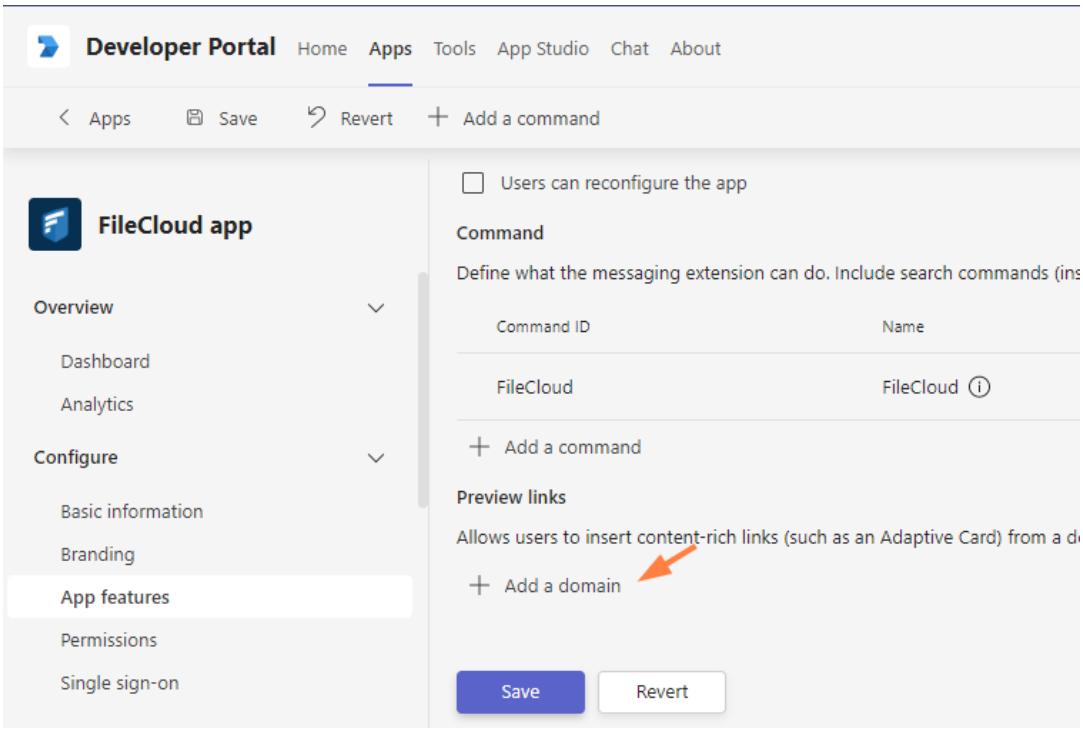
f. Replace the “commands” key and its empty value with the information above that you copied.



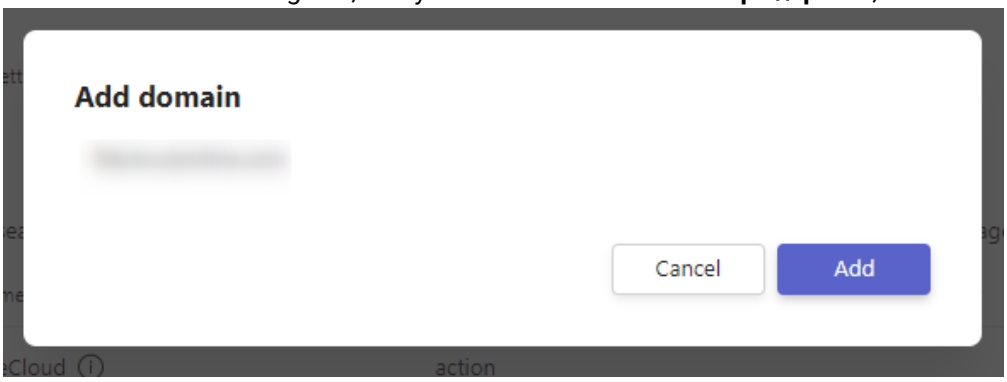
g. Return to the **App package editor**, and click **Save** and then **Update**.
The **App package editor** validates your changes and displays any errors that you should correct.

h. In the navigation pane, click **App Features**, and click **Messaging Extension**.

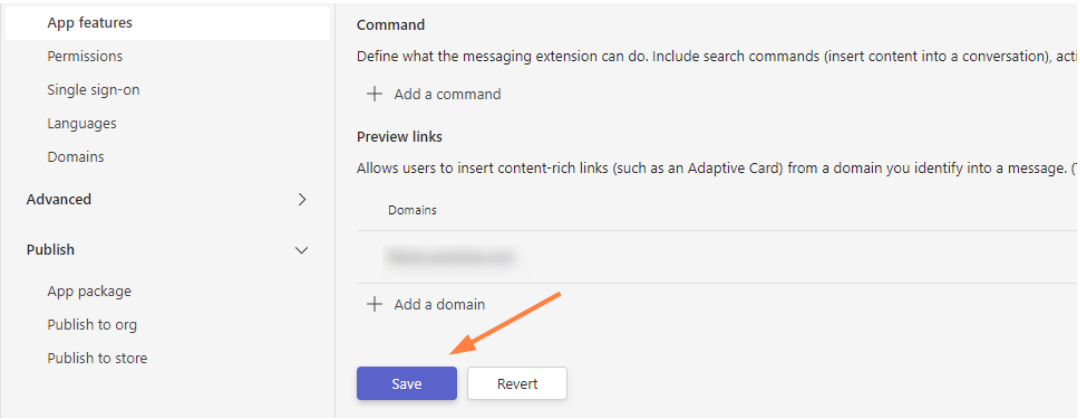
Now, in the **Messaging Extension** screen, click **Add a domain**.



i. In the **Add Domain** dialog box, add your domain without the **https:// prefix**, and click **Add**.

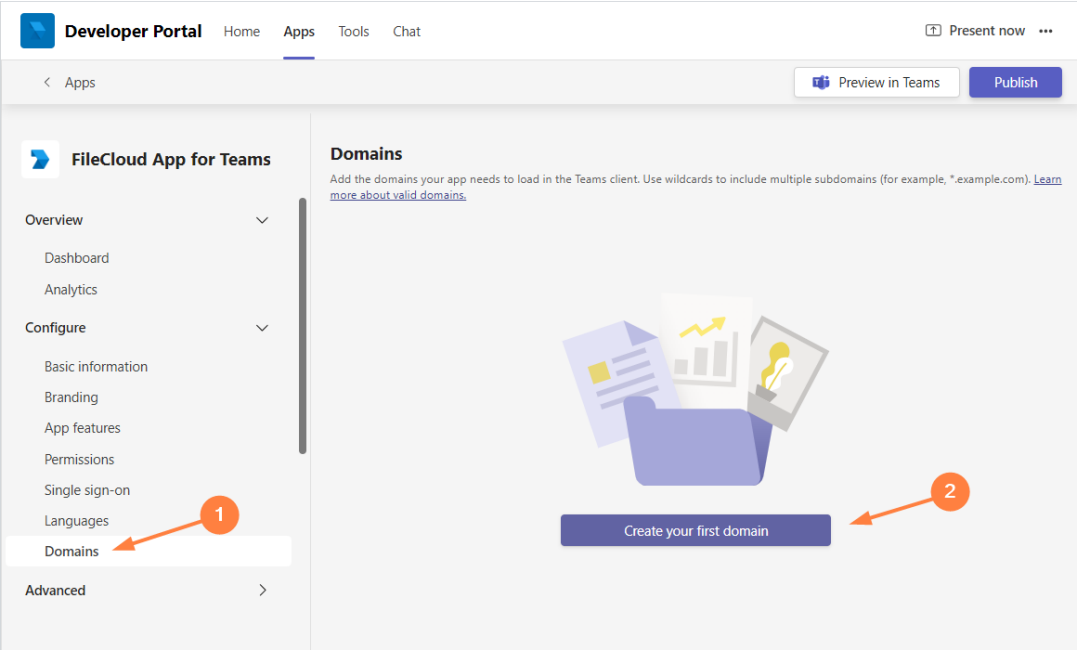


j. In the **Messaging Extension** screen, click **Save**.

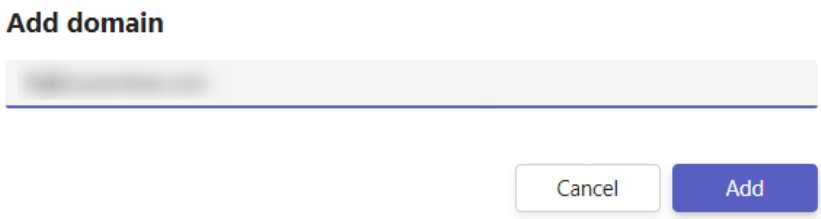


6. Add your domain to a global domains list.

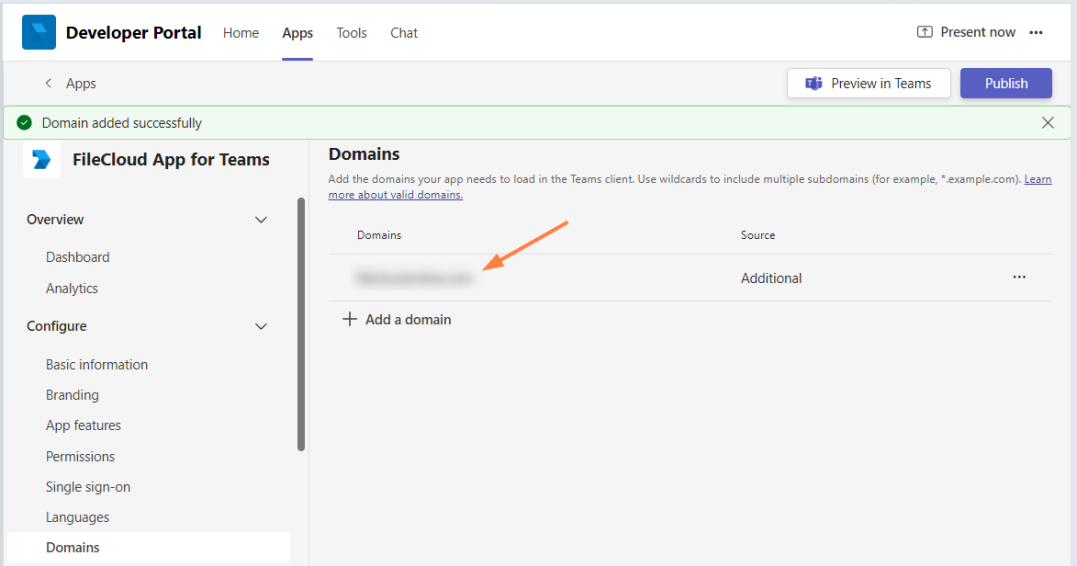
a. In the navigation pane, click **Domains**, and then click **Create your first domain**.
(If you already have domains listed, click **Add a domain**.)



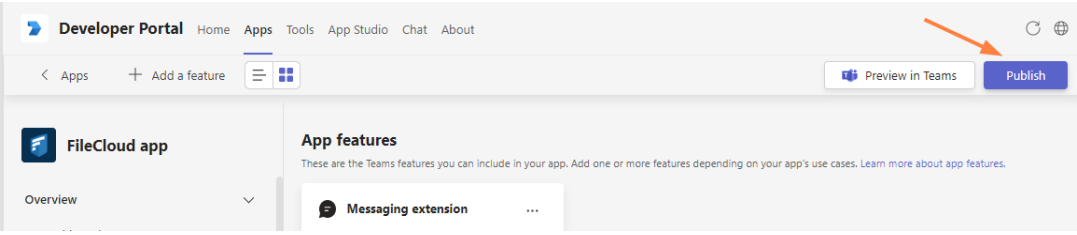
b. In the **Add domain** dialog box, enter your domain (the same one you entered above, in step 4h).



c. Click **Add**.
The domain is added to the list:

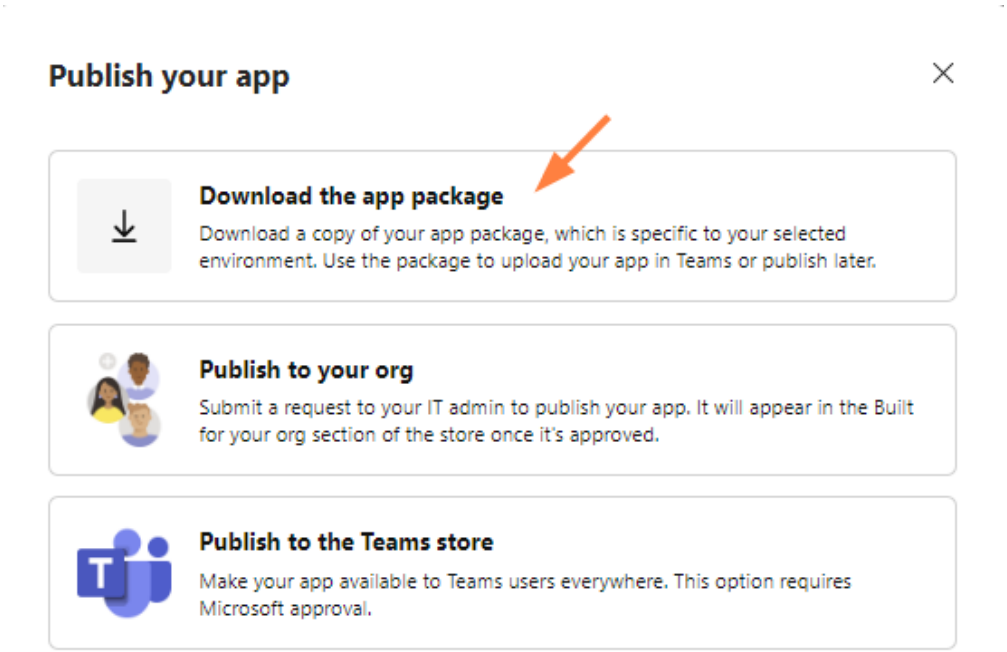


7. Export the application manifest zip file from Teams' **Developer Portal**.
   a. Click **Publish**.



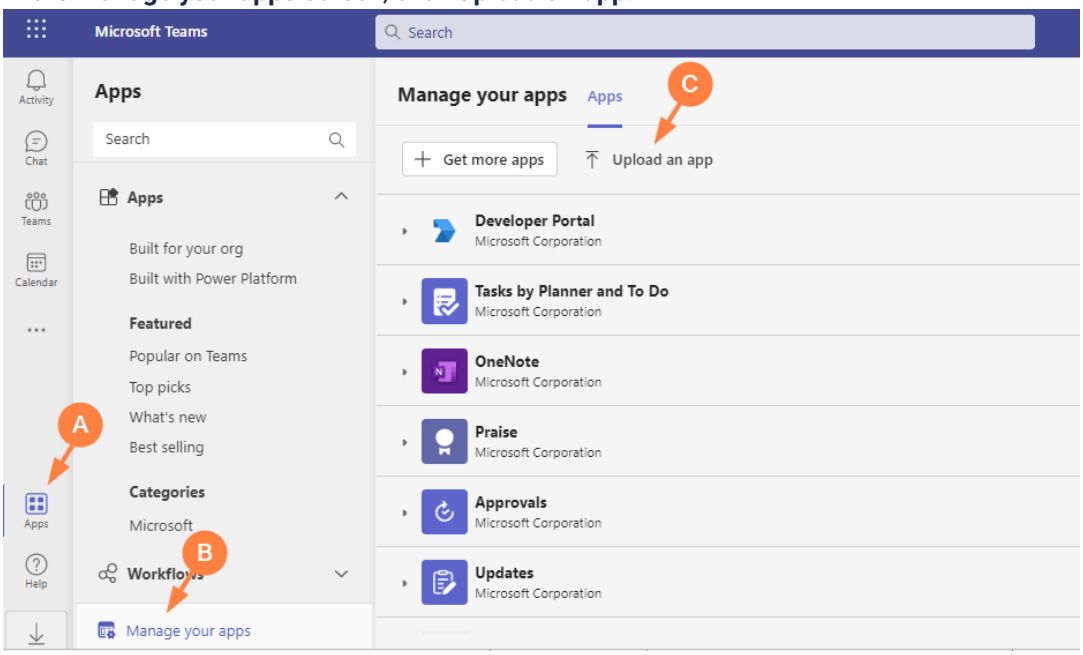
   The **Publish your app** dialog box opens.

   b. Click **Download the app package**.



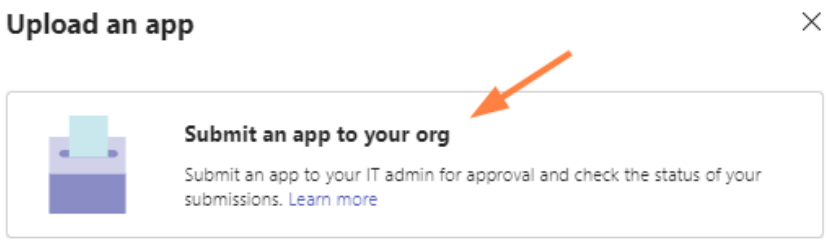
c. Save the downloaded app package zip file.

8. Upload the application and submit it for approval in MS Teams.
   a. In the MS Teams navigation pane, click **Apps.**
   b. In the left panel click **Manage your apps**.
   c. In the **Manage your apps** screen, click **Upload an app**.



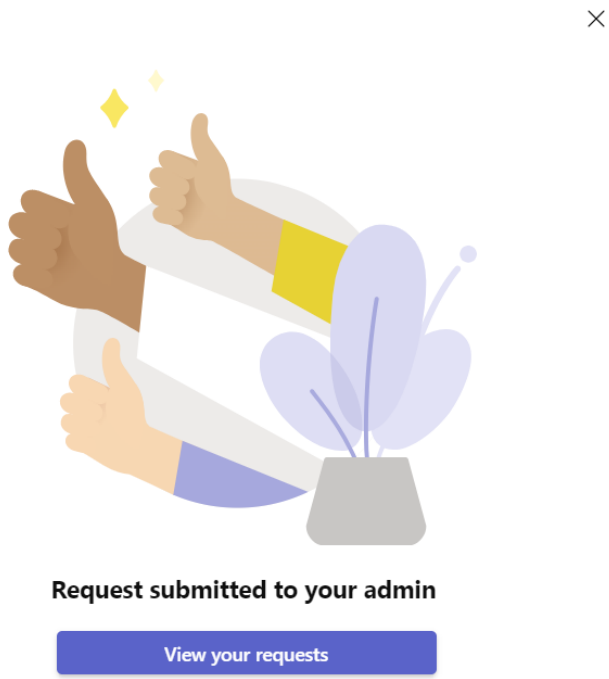
The **Upload an app** dialog box opens.

d. Click **Submit an app to your org.**



Your file explorer opens.

e. Select your app package zip file.
You should now see:



f. As the Teams administrator, approve and publish the app.
For more information, see https://docs.microsoft.com/en-us/MicrosoftTeams/manage-apps#approve-a-custom-app .
The app's **Status** changes to **Approved**, and the app becomes available in your company's app store.

9. Next enable MS Teams integration in FileCloud

For FileCloud Admins: Enabling Integration with MS Teams

After FileCloud configuration in MS Teams has been completed by a Teams administrator, a FileCloud administrator must enable FileCloud/MS Teams integration in the FileCloud Admin portal.

> If you update FileCloud from a version prior to 21.2, you must manually add some configurations to the .htaccess file so that login to FileCloud from MS Teams works correctly. See **Configuration after FileCloud upgrade**, below.

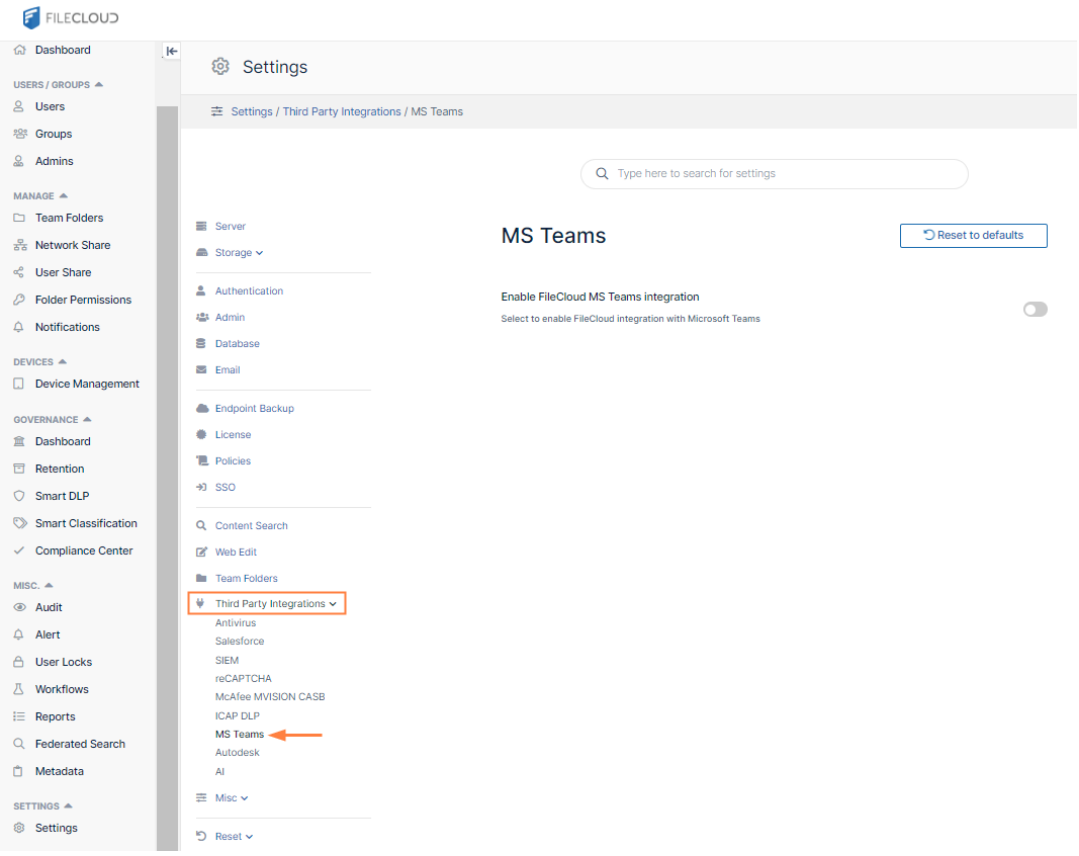
> FileCloud Server must be able to communicate with Microsoft Servers in order for this integration to work. Internet connectivity, or access to the URL https://login.botframework.com/v1/.well-known/keys is required, as well as 2-way communication with the domains teams.microsoft.com, *.teams.microsoft.com, and *.skype.com.

⚠️ **Note regarding Chrome and Edge users**
Users who access MS Teams through Chrome or MS Edge will not be able to log in to FileCloud from MS Teams' FileCoud tab unless the cookie **SameSite** value is set to **None**.

For instructions on setting the **SameSite** value, see Improving Cookie Security.

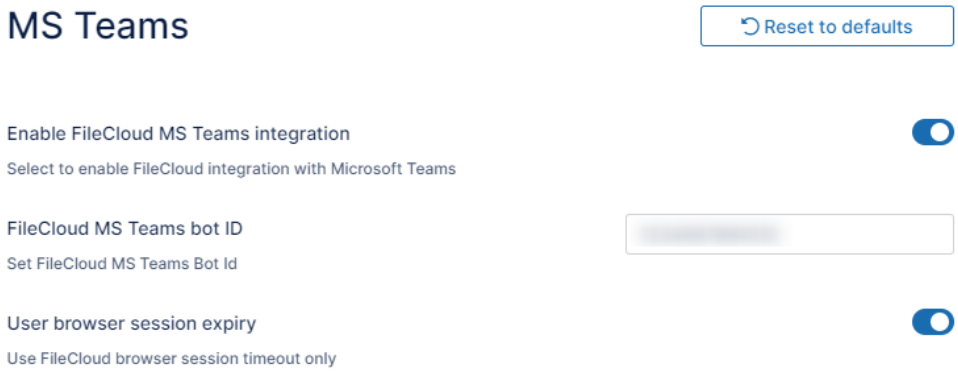
To enable FileCloud integration with MS Teams:

1. Open the MS Teams settings page.
   To go to the MS Teams settings page
   a. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the Settings navigation page, click **Third Party Integrations**



.

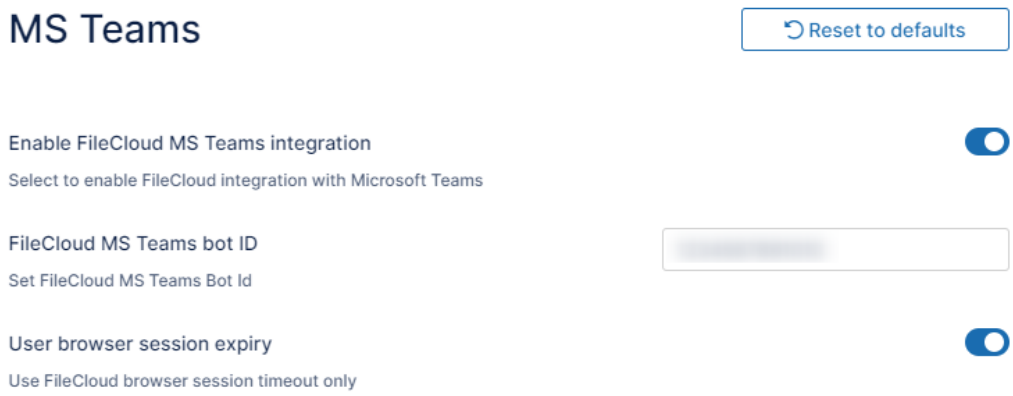
b. In the inner navigation bar on the left of the Third Party Integrations page, expand the **Third Party Integrations** menu, and click **MS Teams**, as shown below.



The **reCAPTCHA** settings page opens.
Open the **reCAPTCHA** settings page.

2. Enable the field **Enable FileCloud MS Teams integration**.

3. Enter the MS Teams Bot Id into **FileCloud MS Teams bot ID**.
   Get the MS Teams Bot Id from the Teams administrator or from Bot Management in MS Teams' App Studio app (see For MS Teams Admins: Configuring FileCloud in Teams).
4. Check **Use browser session expiry** to use the FileCloud session timeout setting (located in **Settings** on the **Server** tab).



5. Click **Save**.

Configuration after FileCloud upgrade

If you upgrade FileCloud from a version prior to 21.2, edit your .htaccess file so that login to FileCloud from MS Teams works correctly:

1. Open the **.htaccess** file:
   - in Windows, C:\xampp\htdocs\.htaccess
   - in Linux, /var/www/.htaccess
2. Find the **Content-Security-Policy** header.
3. Add:
   teams.microsoft.com *.teams.microsoft.com *.skype.com
   to each of the following three directives in the Content-Security-Policy:
   - **script-src**
   - **frame-src**
   - **frame-ancestors**
4. Make sure that each directive is followed by 'self' and ends with a semicolon.
   Example configuration:

```
<IfModule mod_headers.c>
Header set X-Frame-Options "SAMEORIGIN"
Header set X-Content-Type-Options "nosniff"
Header set Strict-Transport-Security "max-age=31536000; includeSubDomains"
Header set X-XSS-Protection "1; mode=block"
Header set Content-Security-Policy: "default-src 'self' *.live.com *.amazonaws.com *.core.windows.net www.google.com
http://127.0.0.1:34320/v1/fileassociations; style-src 'unsafe-inline' 'self';script-src 'unsafe-inline' 'unsafe-eval'
'self' www.google.com  www.gstatic.com docs.google.com teams.microsoft.com *.teams.microsoft.com *.skype.com;frame-src
'self' www.google.com *.live.com docs.google.com teams.microsoft.com *.teams.microsoft.com *.skype.com; font-src 'self'
data:;img-src www.gstatic.com 'self' data: *.duosecurity.com *.live.com *.amazonaws.com *.core.windows.net *.office.net;
frame-ancestors 'self' teams.microsoft.com *.teams.microsoft.com *.skype.com;"
</IfModule>
```

Redirection to Login Screen

If you have integrated your system with MS Teams, and login frequently redirects users back to the login page:

1. Open cloudconfig.php:
   Windows Location: XAMPP DIRECTORY/htdocs/config/cloudconfig.php
   Linux Location: /var/www/config/cloudconfig.php
2. Add the following settings:

```
define("TONIDOCLOUD_COOKIE_SAME_SITE_TYPE", "None");
define("TONIDOCLOUD_SECURE_COOKIE", 1);
define("TONIDOCLOUD_HTTPONLY_COOKIE", 1);
```

Setting Up AutoCAD File Preview with Autodesk Viewer

Beginning with FileCloud 23.1, if a file has multiple 2D and 3D viewing options, the Autodesk viewer in FileCloud lets users display the different views.

Integration with Autodesk Viewer is available in FileCloud Version 22.1 and higher.
Each time an AutoCAD file is previewed, it is stored outside FileCloud on Autodesk's servers for 30 days.
The first time an AutoCAD file is previewed from your site, Autodesk charges you in flex tokens (cloud credits). Subsequent times the (unmodified) file is previewed, by any user on the site, you are not charged. You are charged again the initial time a file is previewed after being modified.
For information about purchasing flex tokens, see https://forge.autodesk.com/pricing

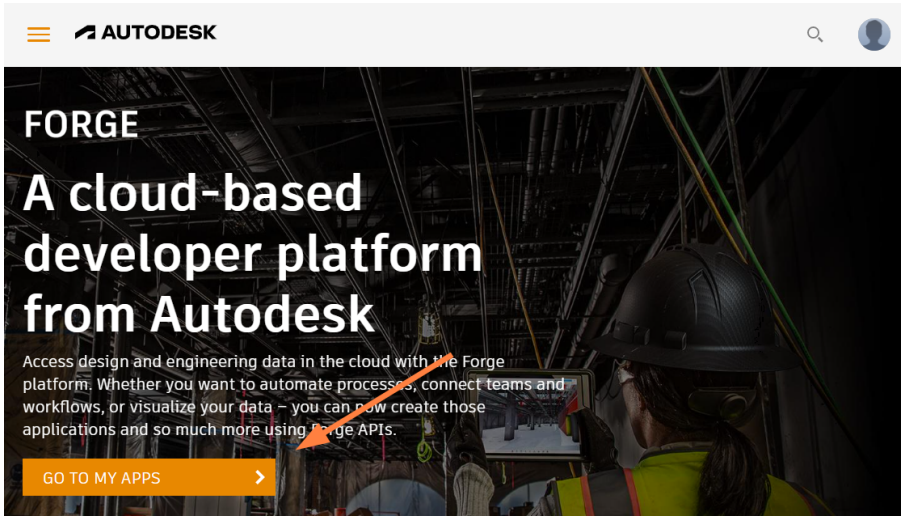
After you configure FileCloud integration with Autodesk Viewer, when users preview 3D and 2D model data file types, they are shown in Autodesk Viewer.

Setting up integration of FileCloud and Autodesk Viewer

Note: If your firewall blocks URLs that do not appear in an allowed list, make sure you add the Autodesk URL to the allowed list.

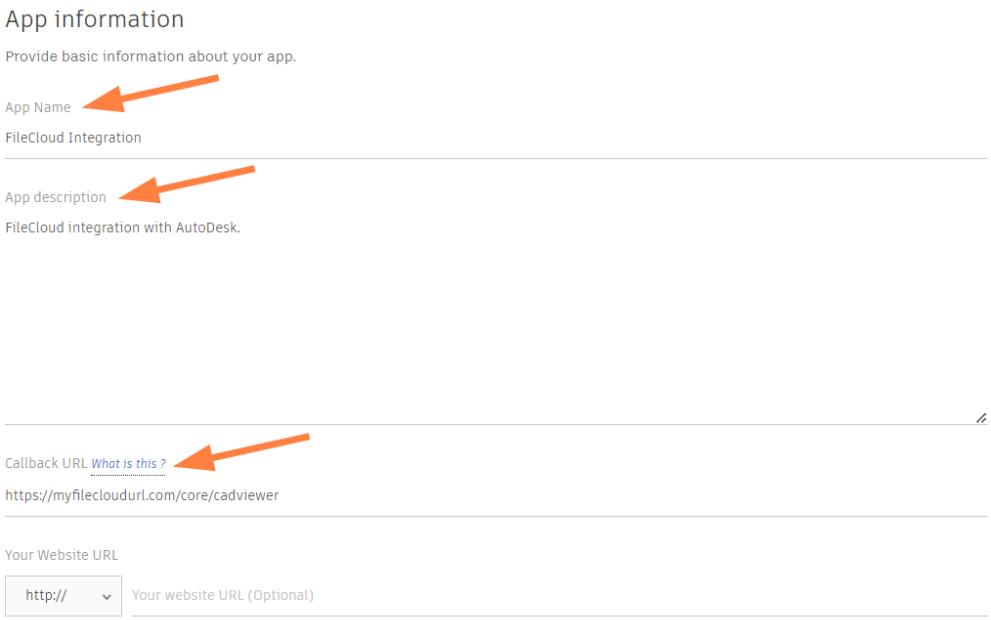
To integrate FileCloud with Autodesk Viewer:

1. Go to https://forge.autodesk.com/.
2. Sign in to your Autodesk account, or create a new one.
3. Click **GO TO MY APPS**.

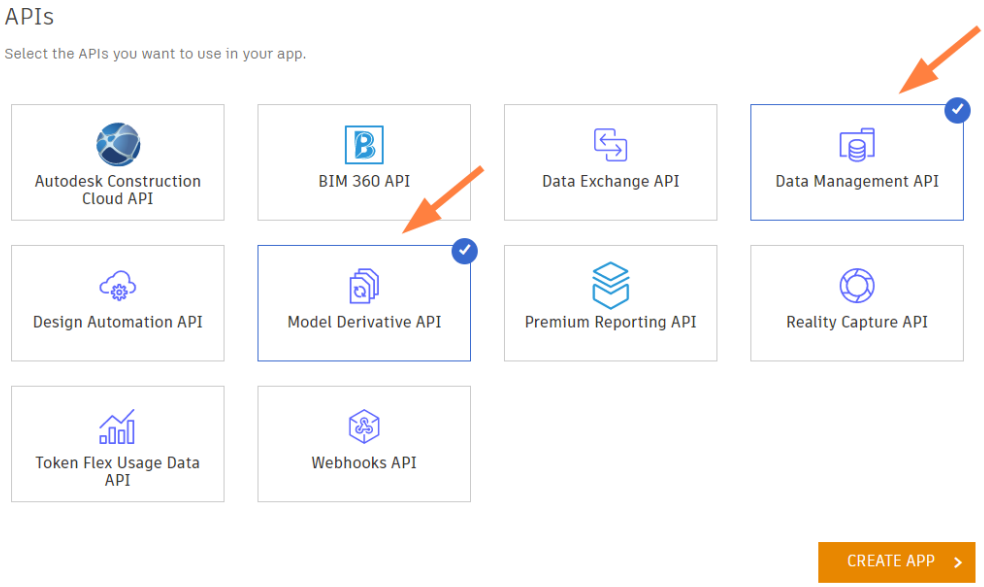


4. Click **CREATE APP**.

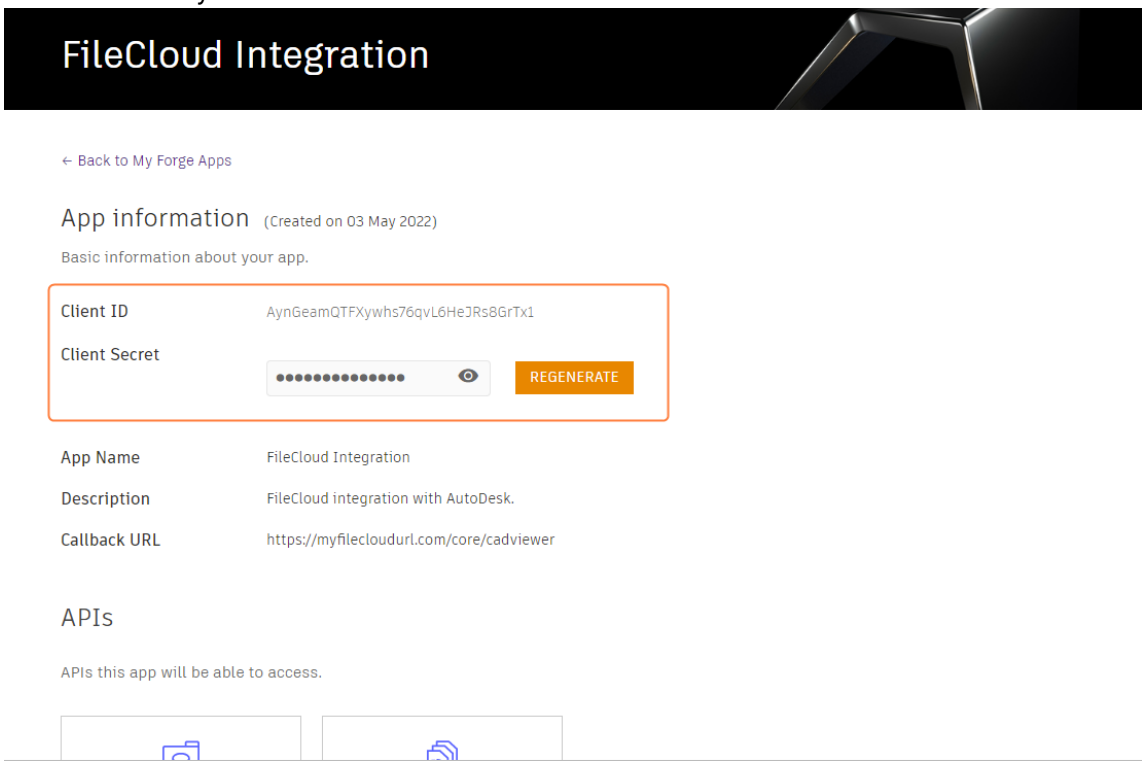
5. Fill in the fields.
   - For Callback URL, enter your FileCloud url + **/core/cadviewer**, for example, https://myfilecloudurl.com/core/cadviewer.
   - You may leave **Site URL** blank, but must fill all other fields.



   - In the APIs section, select only **Data Management API** and **Model Derivative API**.



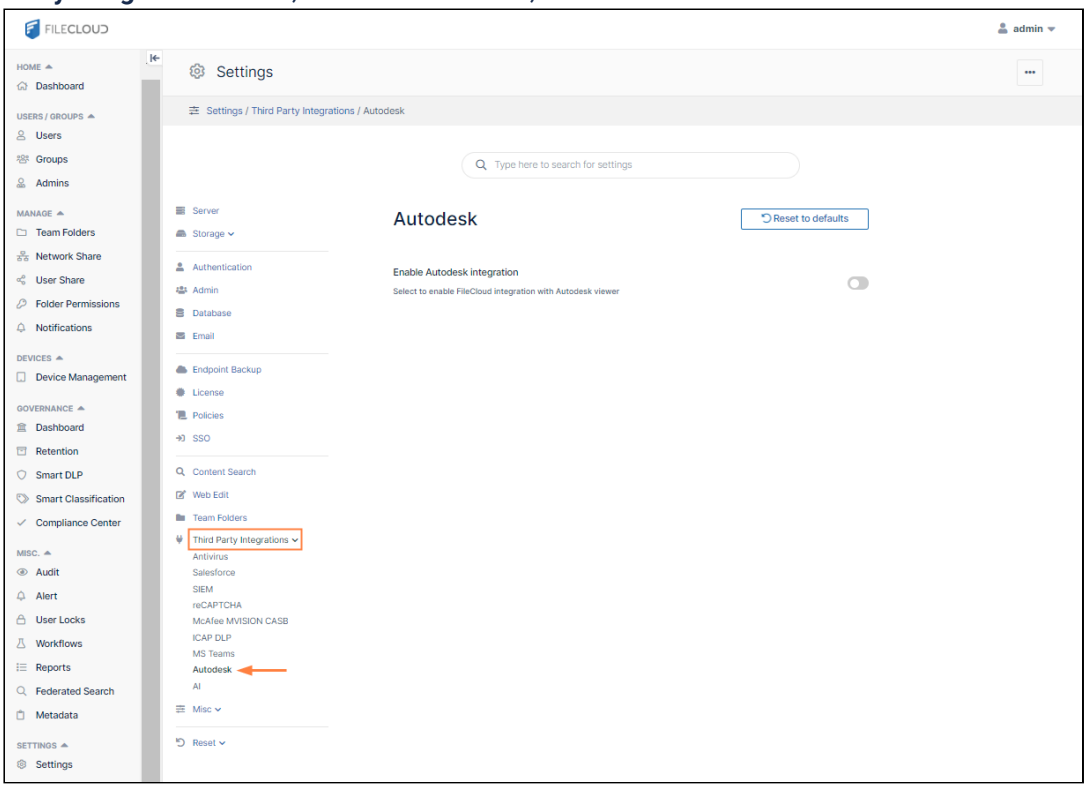
6. Click **CREATE APP**.
   The screen lists your **Client ID** and **Client Secret**.



7. In the FileCloud admin portal, open the Autodesk settings page.
   To go to the Autodesk settings page

a. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the **Settings** navigation page, click **Third Party Integrations** .

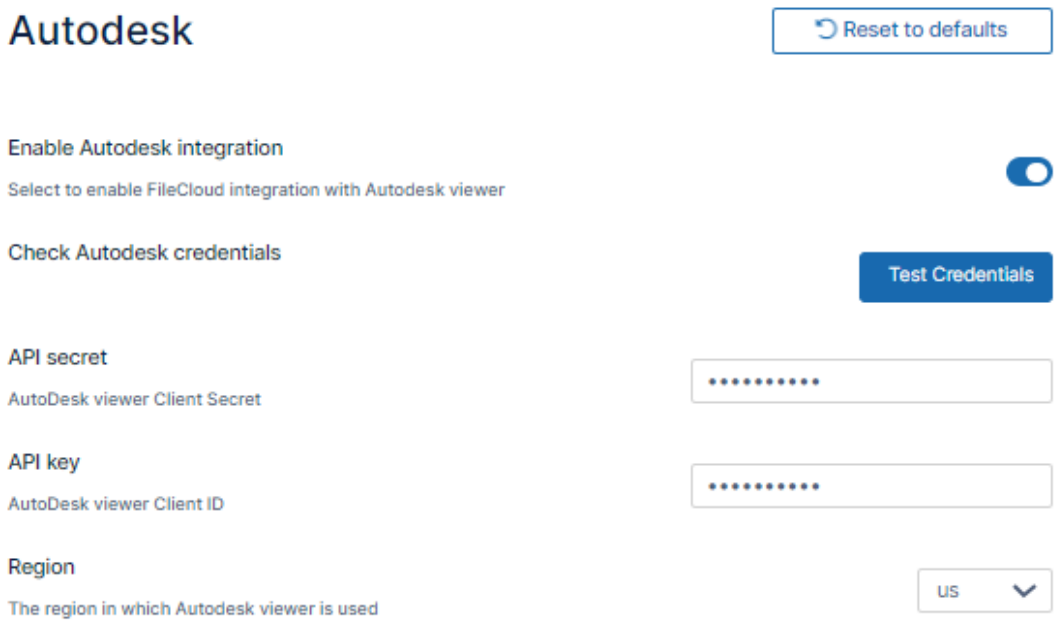
b. In the inner navigation bar on the left of the **Third Party Integrations** page, expand the **Third Party Integrations** menu, and click **Autodesk**, as shown below.



The **Autodesk** settings page opens.

8. Enable the field **Enable Autodesk integration**.
   Additional fields appear.
9. In **API Secret,** enter your Autodesk Viewer **Client Secret**.
10. In **API key**, enter your Autodesk Viewer **Client ID**.

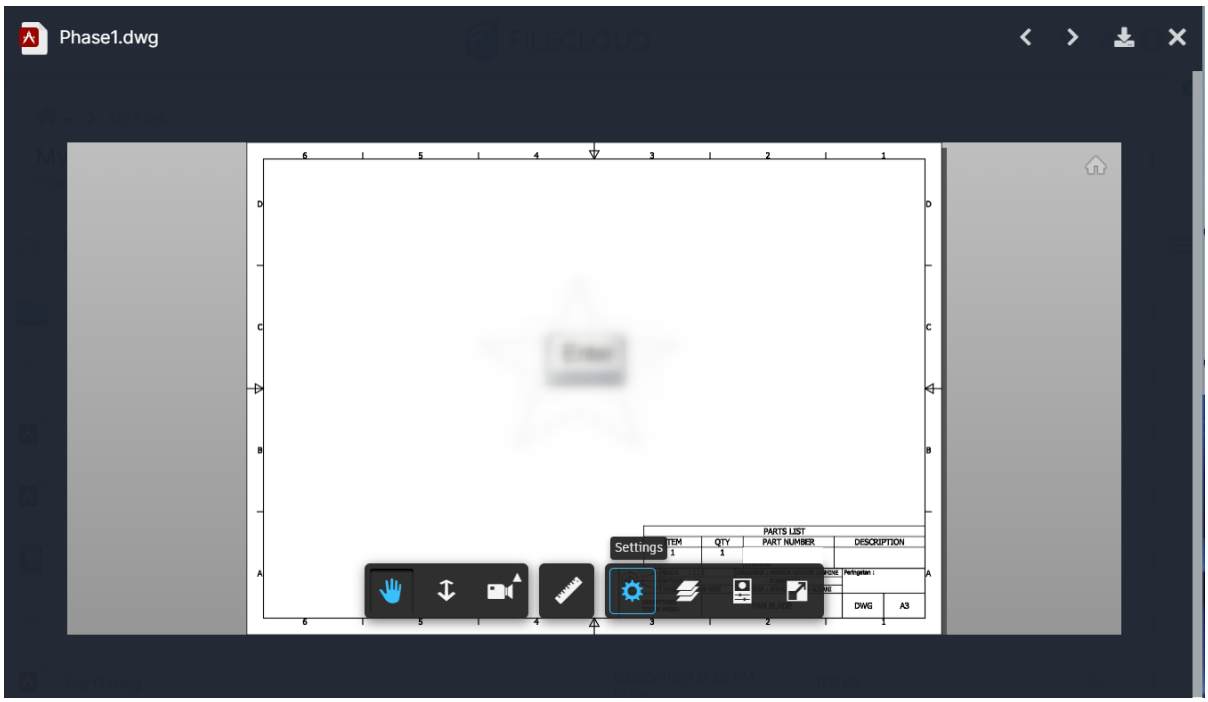
11. Change the **Region** if it is not accurate.



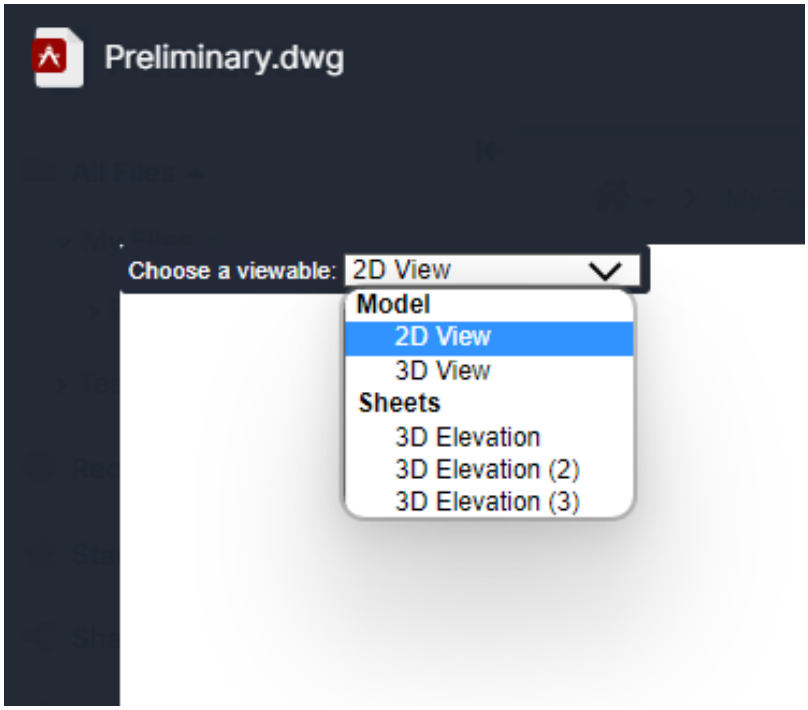
12. Click **Save**.
13. Make the following change to the Apache SSL config file in the **<VirtualHost>** definition:
    a. Open httpd-ssl.conf:
    Windows Location: **XAMPP DIRECTORY\apache\conf\extra\httpd-ssl.conf**
    Linux Location: **/etc/apache2/sites-enabled/000-default.conf**
    b. Near the end of the file, but before **</VirtualHost>** , add the following:

```
AllowEncodedSlashes NoDecode
```

Your integration of Autodesk Viewer and FileCloud is now complete.
When users preview a model data file in FileCloud, they see the image in a screen similar to:



For files that have multiple views, the following drop-down list appears in the upper-left corner:



Note: The drop-down list with multiple options for viewing only appears for files that have multiple views available.

AI Integration

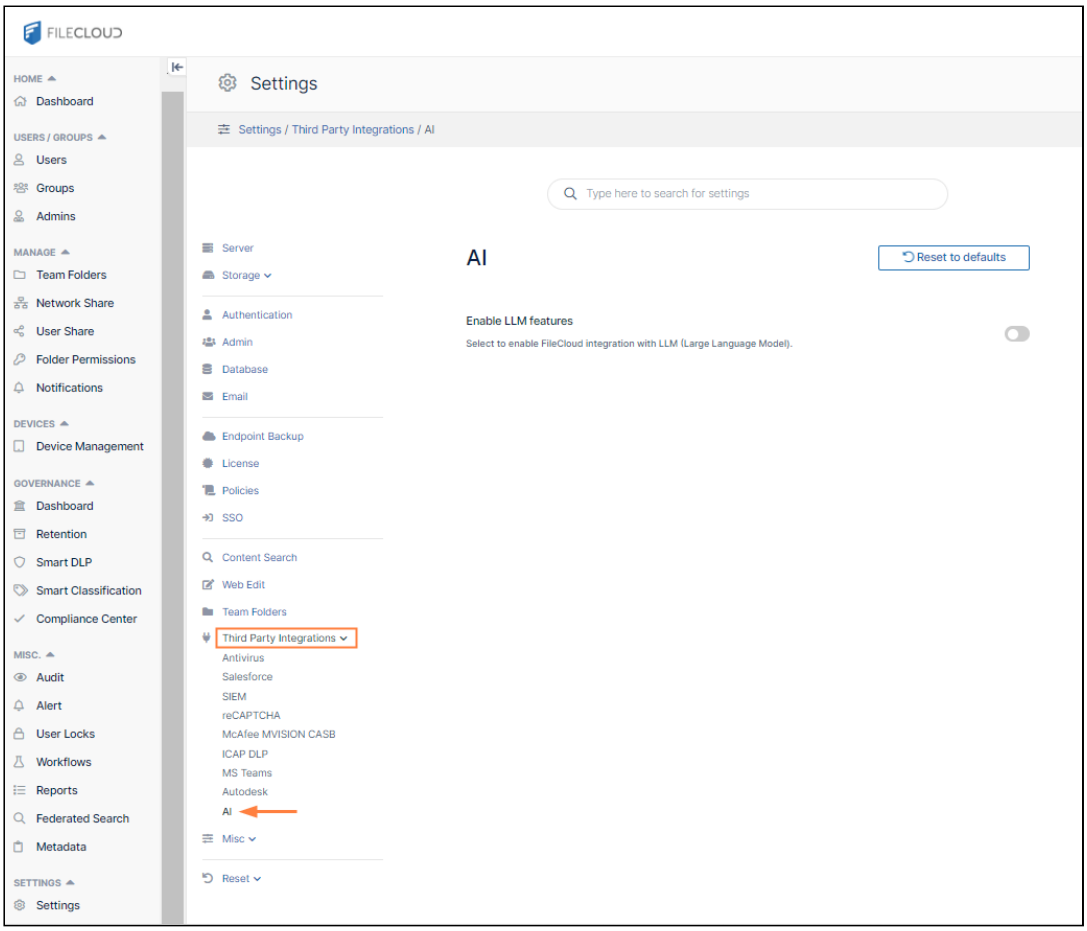
> The ability to configure a Large Language Model for FileCloud Smart Classification is available in versions 23.232 and higher.

FileCloud's Smart Classification includes an AI classifier which requires integration with a Large Language Model (LLM) to function. A Large Language Model, which is trained on very large amounts of data, is a type of algorithm used in AI.

Currently, OpenAI is the only provider available for integrating FileCloud with a LLM.

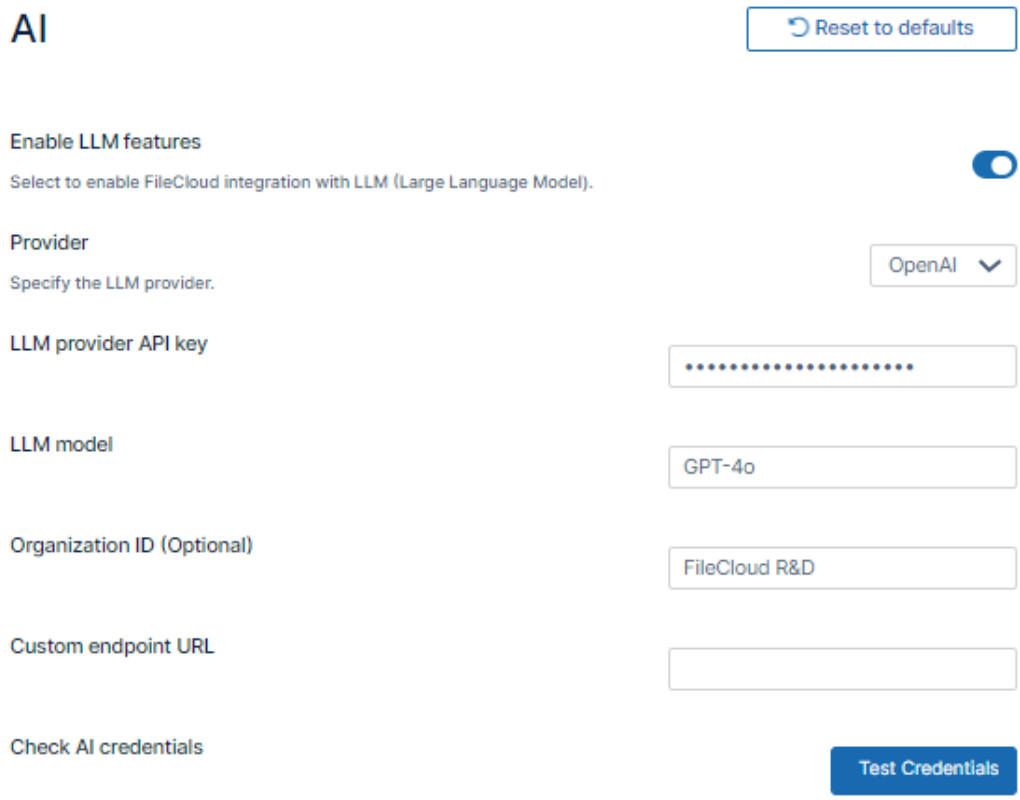
To integrate FileCloud with OpenAI:

1. Open the AI settings page.
   To go to the AI settings page
   a. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the **Settings** navigation page, click **Third Party Integrations** .
   b. In the inner navigation bar on the left of the **Third Party Integrations** page, expand the **Third Party Integrations** menu, and click **AI**, as shown below.

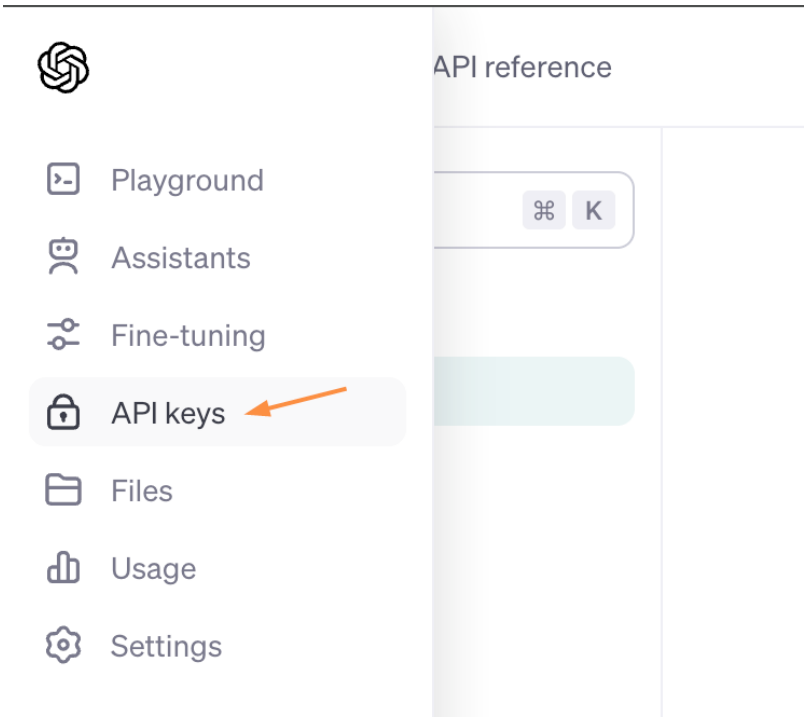


The **AI** settings page opens.

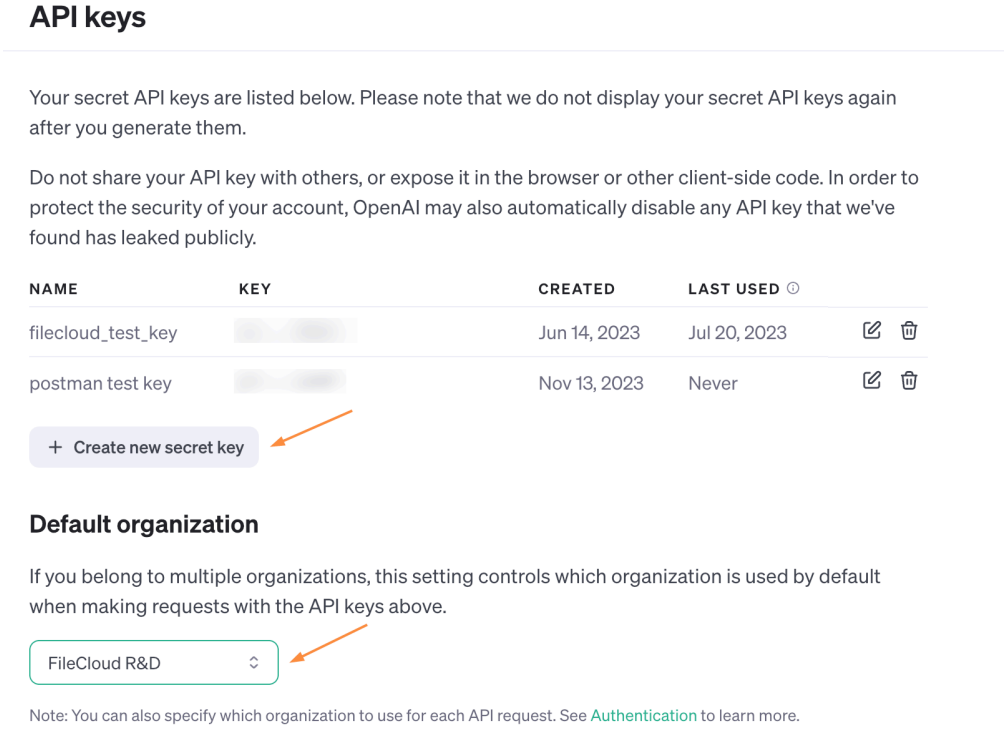
2. Enable the setting **Enable LLM features**.
   The AI settings appear.



3. Check **Enable LLM Features**.
4. In **Provider**, choose **OpenAI**.
5. Enter the values for **LLM provider API Key** and **Organization ID**.
   To get these values, log in to the OpenAI platform at https://platform.openai.com/login (you must have a valid OpenAI subscription) and click **API keys** in the left navigation panel.

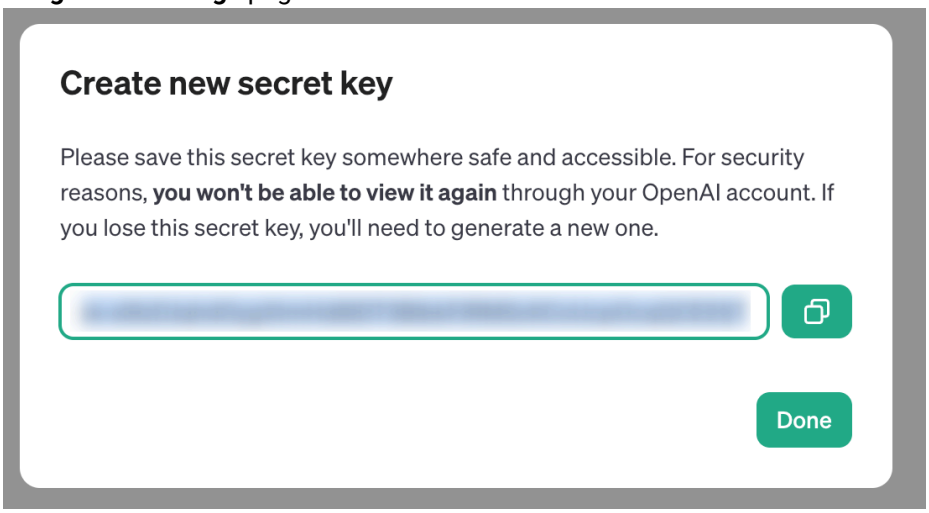


The **API keys** page opens:



On the **API keys** page:

- Click **Create new secret key** and create a new key. Copy and save it (you cannot access it again through your AI account), and then enter it into **API key** on the FileCloud **AI Integration Settings** page.



- Under **Default organization**, view your organizations, and optionally, enter one into **Organization** on the FileCloud **AI Integration Settings** page to have it used with each API request.

6. In **Model**, enter the value for your model. For help determining your model, see https://platform.openai.com/docs/models.
7. In most cases you are not required to enter a **Custom URL**. It is only necessary if you use a custom OpenAI instance.
8. Click **Test Credentials** to confirm that **FileCloud** and **AI** are properly integrated.

To ensure optimal functionality and avoid disruptions, confirm that the LLM you select is currently supported by OpenAI. View the list of supported models and their deprecation timelines on OpenAI's model deprecation page.

CDR Integration

> FileCloud integration with Forcepoint CDR is available in version 23.241.4 and higher.
> Forcepoint CDR is only available for customers with Advanced licenses; if you are upgrading FileCloud and intend to use Forcepoint CDR, please also upgrade your license.

When Forcepoint CDR (Content Disarm and Reconstruction) is integrated with FileCloud, each file (of a supported type) uploaded into FileCloud is put into a non-editable quarantine state and sent to Forcepoint CDR. Forcepoint CDR rebuilds the file, omitting any potentially malicious code, and returns the sanitized file to FileCloud.

Limitations:

- Forcepoint CDR does not send a notification to the user's FileCloud account if a threat is found; it simply returns the file to FileCloud with the threat removed.
- Only files in Network Folders that are changed within FileCloud are scanned and sanitized; files in Network Folders that are changed outside FileCloud are not.
- File changes made to a file in a WOPI Web edit co-editing session are not sent to Forcepoint for sanitization until all users in the session have closed the file for edit.
- While a file is in quarantine, FileCloud rejects new uploads of the file.

Integrating Forcepoint CDR with FileCloud

Required settings:

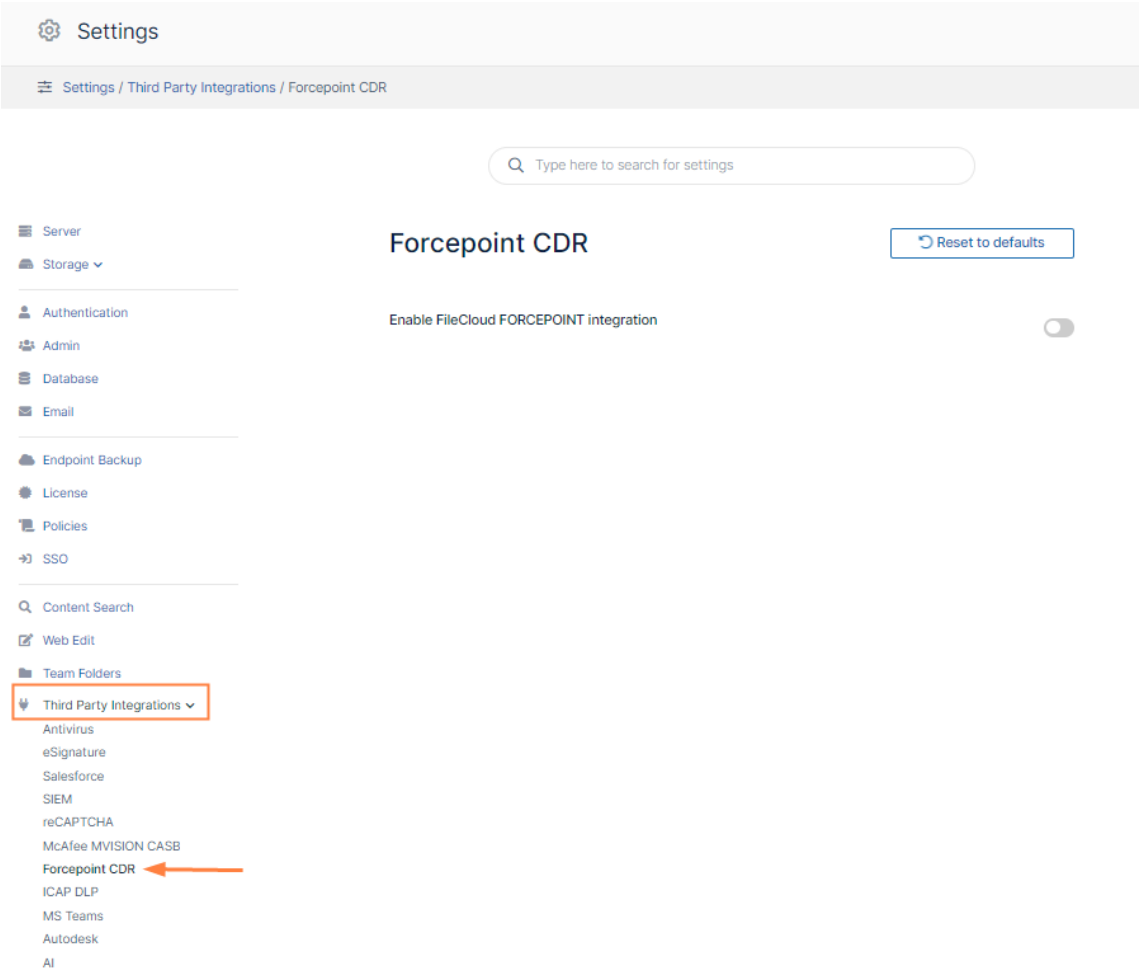
- Forcepoint CDR integration works only if locking is enabled (the default setting). For help enabling locking, see Misc Settings.
- We recommend setting **Number of old versions to keep for each file** to **1** or higher (default is **3**) before using Forcepoint CDR integration to avoid losing data. Without this setting, loss of original versions of files will occur if Forcepoint CDR returns an unsupported file and **Delete extensions that Forcepoint CDR does not support** is enabled. For help setting this value, see Setting up Managed Storage.

To set up integration Forcepoint CDR with FileCloud:

1. Open the **Forcepoint CDR** page.

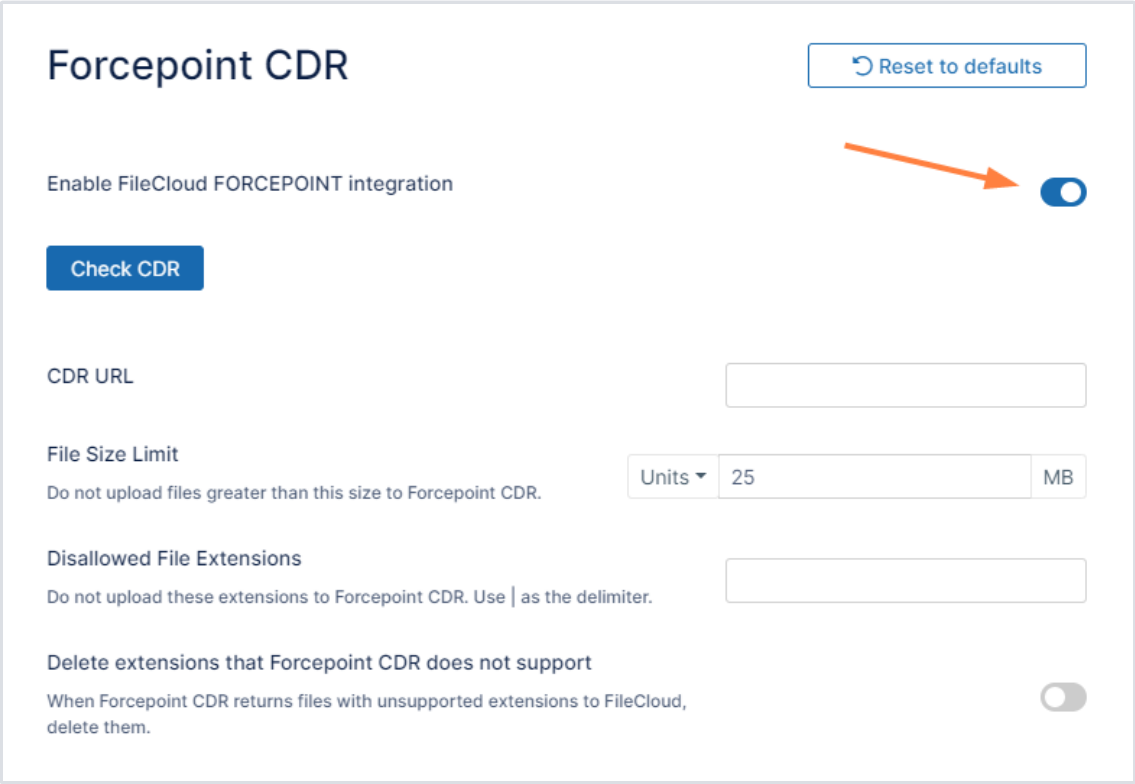
To go to the Forcepoint CDR page

1. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings.** Then, on the **Settings** navigation page, click **Third Party Integrations** .
2. In the inner navigation bar on the left of the **Third Party Integrations** page, expand the **Third Party Integrations** menu, and click **Forcepoint CDR**, as shown below.



The Forcepoint CDR settings page opens.

2. Toggle on **Enable FileCloud FORCEPOINT integration** to view the Forcepoint fields.



3. Fill in the fields as indicated in the following table.

Field	Description	Default value	Notes
Enable FileCloud FORCEPOINT Integration	Turn integration with Forcepoint CDR on and off.	disabled	
Check CDR	Click to confirm that your **CDR URL** is valid.	N/A	
CDR URL	The URL of your company's Forcepoint CDR server	blank	

Field	Description	Default value	Notes
File Size Limit	The largest size of a file that FileCloud can send to Forcepoint CDR.	25	The maximum size we have tested that was processed successfully in Forcepoint was 100 MB. However, the maximum size any Forcepoint server can process depends on the hardware configuration of the Forcepoint server.
Disallowed File Extensions	File extensions that you want to prevent from being uploaded to Forcepoint CDR. These files remain in FileCloud but are not sanitized. (There are also file types that Forcepoint CDR cannot process. These files are treated differently; they are uploaded to Forcepoint and returned as unsupported).	blank	
Delete extensions that Forcepoint CDR does not support	Deletes files that are returned because they have extensions that Forcepoint CDR does not support. File types that are not supported for sanitization include file types that Forcepoint does not support in general, such as PSD and MP4, and file types blocked by your Forcepoint CDR configuration. For more information see Forcepoint's online CDR help.	disabled	

4. To ensure that integration with Forcepoint CDR runs efficiently, add the following configuration in the message queue config file:
   a. Open the message queue config file:
      Windows location: **C:/xampp/htdocs/src/Scripts/config/default.json**
      Linux location: **/var/www/html/src/Scripts/config/default.json**

b. Set the field **parallel_high_priority_workers_count** to a value of **1** or higher.
We recommend initially setting the value to around 20% of the value in **parallel_workers_count**, and modifying it as necessary for your environment.

If Forcepoint CDR cannot sanitize a file due to an error a notification is sent to the user and both a notification and an email are sent to the admin.

- Files that cannot be sanitized due to an error are repeatedly resent for sanitization until it is successful or the admin goes to the **Quarantined Files** page and either deletes the non-sanitized file version or removes it from quarantine.
- If **Delete extensions that Forcepoint CDR does not support** is enabled, files with extensions that are not supported by Forcepoint CDR are deleted from FileCloud. If there is a prior version of the file in FileCloud (if it was an update to a file) the original version is not deleted.
- To customize the email sent to the user, go to Customization > Email Templates and edit the template **Errors During Sanitization On Forcepoint CDR Email Template**.

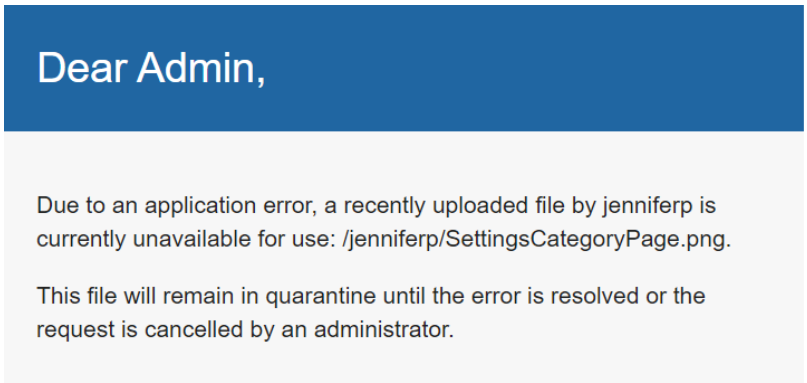
While a file is being sanitized, the file and its parent folders are locked for editing and other changes. The screen does not reflect that the file has been returned from Forcepoint CDR and is now unlocked until the user refreshes the screen, as in the following video.

Notice that the size of the file in the video is reduced after processing. This may happen when the file is recreated in Forcepoint CDR, making it slightly smaller or larger. If it takes longer than a minute to process the uploaded/modified file in Forcepoint CDR, the file's modified date will reflect the time change.

Deleting file versions in quarantine

If a file cannot be sanitized due to an error, it is repeatedly resent for sanitization until it is successful or an admin either deletes the non-sanitized file version or removes it from quarantine. Each time it is

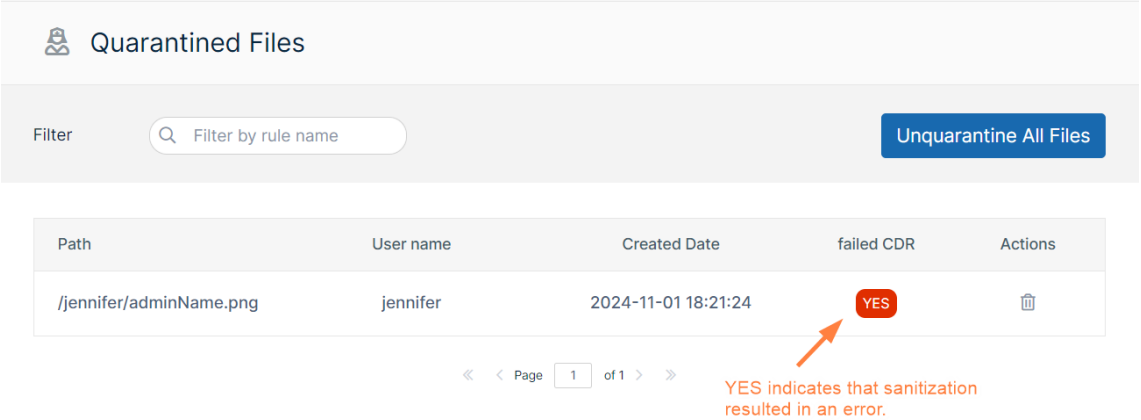
sent for sanitization and fails FileCloud sends you a notification:



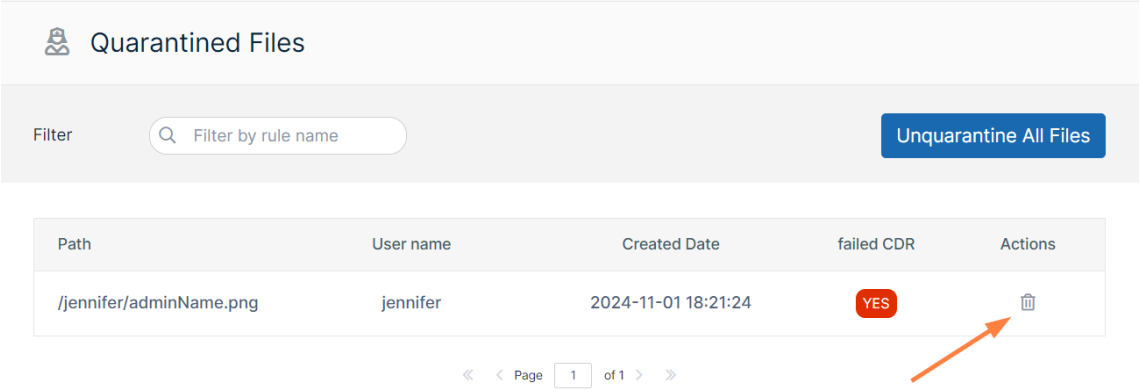
The file listed in the **Quarantined Files** screen is the version of the file that has been quarantined and sent for sanitization, which is the latest version of the file. Earlier versions of the file may exist in FileCloud and will remain in FileCloud even if you delete the versions in quarantine.

To delete a file that is repeatedly being sent for sanitization:

1. In the admin portal navigation panel, click **Quarantined Files**.
   All quarantined files are listed, including those that haven't finished the initial sanitization process as well as those that are being repeatedly sent for sanitization due to an error.
   Files listed because they have not finished the initial sanitization process do not have the **Delete** option.
   If files are listed because of an error, the failed CDR column displays **YES**.



2. To delete a file version (a sanitization request) stuck in quarantine, in the **Actions** column, click the Delete icon.



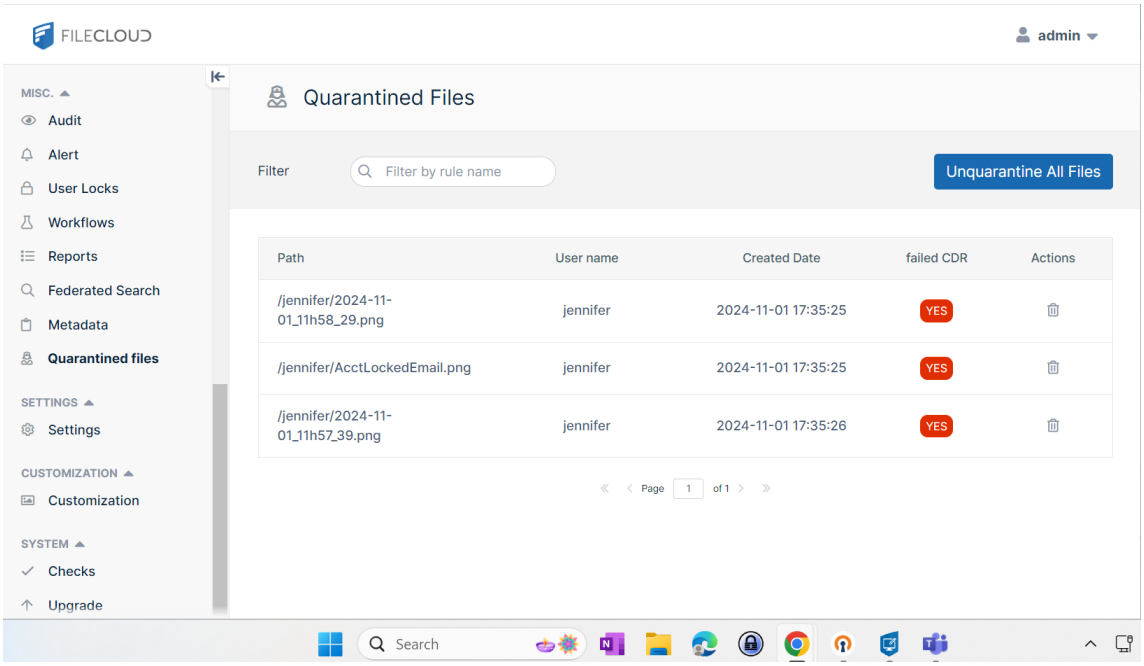
The file no longer appears in the **Quarantined Files** page.
The deleted version of the file is removed from FileCloud, and is no longer sent for sanitization.
Note: If this is the only version of the file in FileCloud, the file is deleted permanently from FileCloud and is not sent to the recycle bin.

Removing all files from quarantine

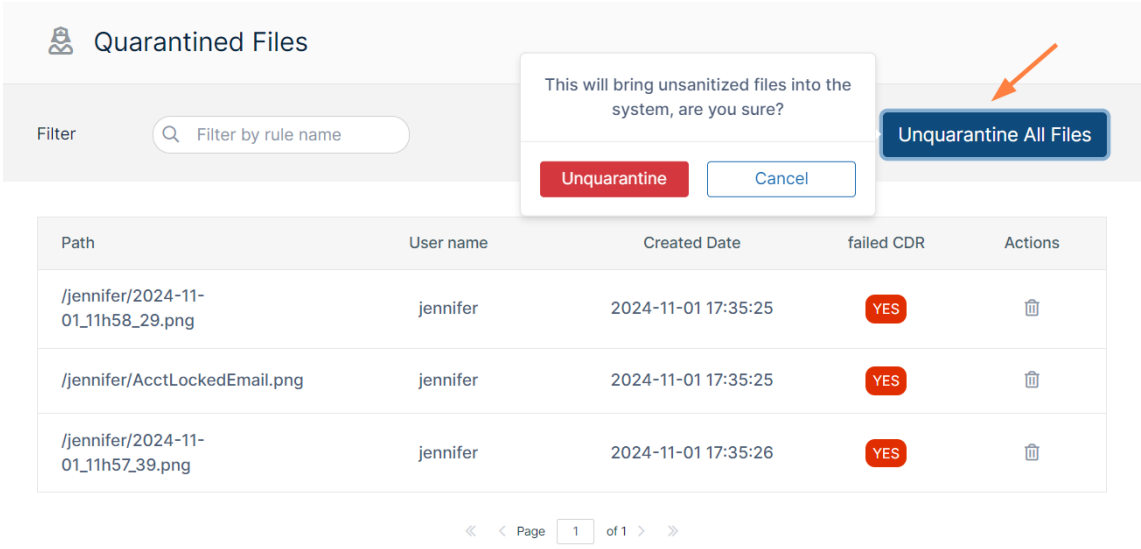
The **Quarantined Files** page includes a button for removing all files from quarantine. If you click this button, all file versions in quarantine, including those that have not yet completed the sanitization process and those that are stuck in the sanitization process due to an error, are removed from quarantine but not deleted from FileCloud. All of these versions of files become available for use in FileCloud in their non-sanitized state.

To remove all files from quarantine:

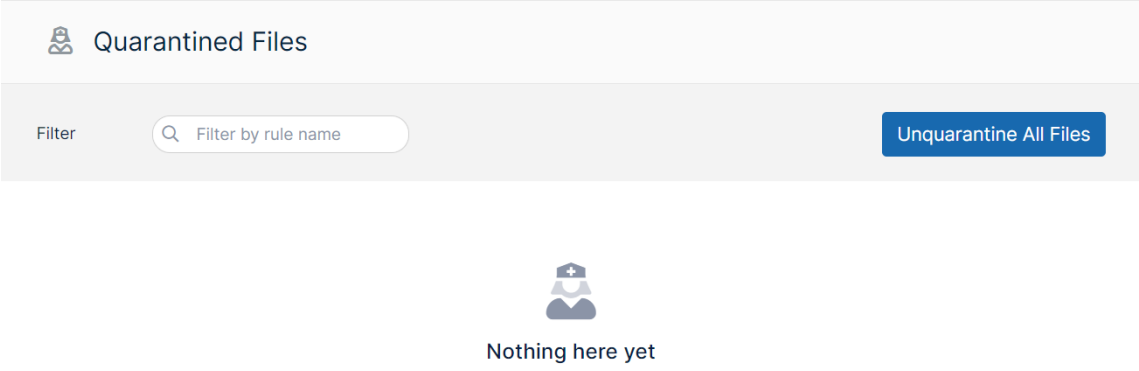
1. In the admin portal navigation panel, click **Quarantined files**.
   All quarantined files are listed.



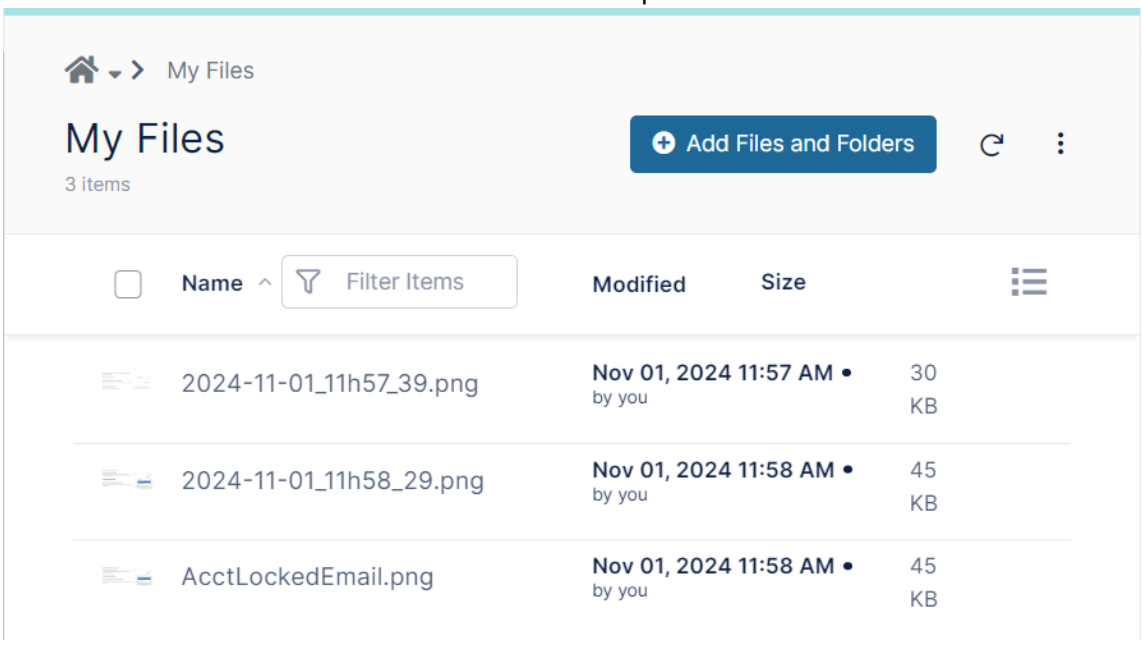
2. To remove all files from quarantine, click the **Unquarantine All Files** button.
   A confirmation box that warns you that unsanitized files will be brought into the system pops up.
3. If it is okay for the files to remain unsanitized, click **Unquarantine** in the confirmation box.



All of the files are removed from the **Quarantined files** screen:



The files remain available to the user who created or uploaded them:



eSignature Integration

> Integration of FileCloud with Signit's eSignature solution is available in FileCloud versions 23.262 and higher.
> Integration of FileCloud with Signority's eSignature platform is available in FileCloud versions 23.241.4 and higher.
> Some users' plans may not include the digital signature option discussed here.

FileCloud can be configured to integrate with Signority or Signit to enable your users to submit files for eSignatures:

How the eSignature process works

In the admin portal, you configure Signority or Signit as an eSignature platform.

In FileCloud, the user selects one or more files to be signed. FileCloud creates a new PDF containing the file or files and prompts the user for a PDF name, a signed document location in FileCloud, and document signers. (The original files remain as they are, and can be used to create other signature documents.)

After recipients are added:

- In Signority, the user enters the Signority site and is prompted to add signature fields to the document.
  The user is then given the option of sending a document for eSignature or digital signature.
  Once a send option is chosen, Signority sends the signing requests to each recipient.
- In Signit, a Signit screen for adding signature fields to the document opens in FileCloud. After adding signature fields, the user directs Signit to send the signing requests to each recipient.

The user can now follow the signature document's status in FileCloud as the recipients sign it, and it moves from **In Progress** to **Completed**. The user receives emails when each recipient finishes signing the document and when all recipients have finished signing.

See (23.263a) eSignature Requests for a step by step description of the above process.

Once the document is signed, the user can access the signed PDF in the FileCloud location specified for storing the signed document.

> Incomplete eSignature documents are stored in the eSignature application (Signority or Signit) as well as FileCloud. In Signority, when a signing request is completed, deleted, or expired, the eSignature document is removed from the eSignature application.

The signer's experience

Reviewing and signing the document

The signer receives an email alerting them that a document is waiting for them to sign it. The signer clicks a link in the email to access the document. Once the signer has reviewed and signed the document, their task is complete. The eSignature application automatically sends the signed document back to FileCloud.

Prerequisites for FileCloud eSignature integration

- You must have either:
  - a trial or paid Signority account, obtained either prior to setting up integration or obtained through FileCloud's admin portal **eSignature** screen.
  - a paid Signit account, obtained prior to setting up integrations.
- eSignature must be enabled in a user's policy for the user to be able to use the eSignature feature. By default, the feature is enabled in all policies.
- Document Converter must be installed and running in FileCloud.

File types supported for eSignature

The file types currently supported in FileCloud for eSignature are:

- PDF
- DOCX
- PPTX
- PNG
- JPG/JPEG

File types that are not supported for eSignature do not show the **eSignature** option in the More [...] drop-down list.

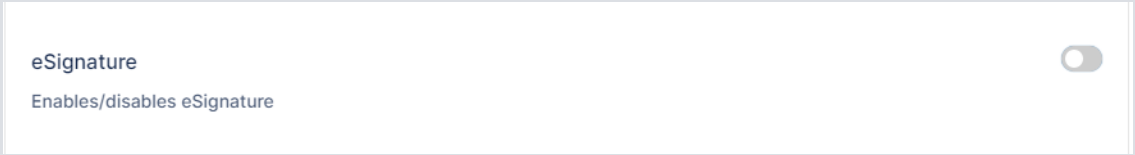
Limitations

- The combined size of all files in a single signing request may not exceed 50 MB.
- Translations of the user interface are available in Arabic, French, Portuguese, and Spanish. However:
  - Messages from the server, which appear when users view a request in the eSignature screen, are not translated.
  - The admin portal does not support these translations.
- Audit details are omitted by default, but can be returned combined with an eSignature document or as a separate file. See To return audit details, below, for information on returning audit details.
- DLP rules that block files from being shared do not block them from being sent for eSignatures.

To enable/disable eSignatures for certain users:

By default, the eSignature feature is enabled in users' policies.

1. In the FileCloud admin portal's left navigation bar, scroll down and click **Settings**. Then, on the **Settings** navigation page, click **Policies** .
   The **Policies** page opens.
2. Edit the users' policy.
3. Click the **User Policy** tab.
   Scroll down until you see the **eSignature** field.
4. Enable or disable **eSignature**.



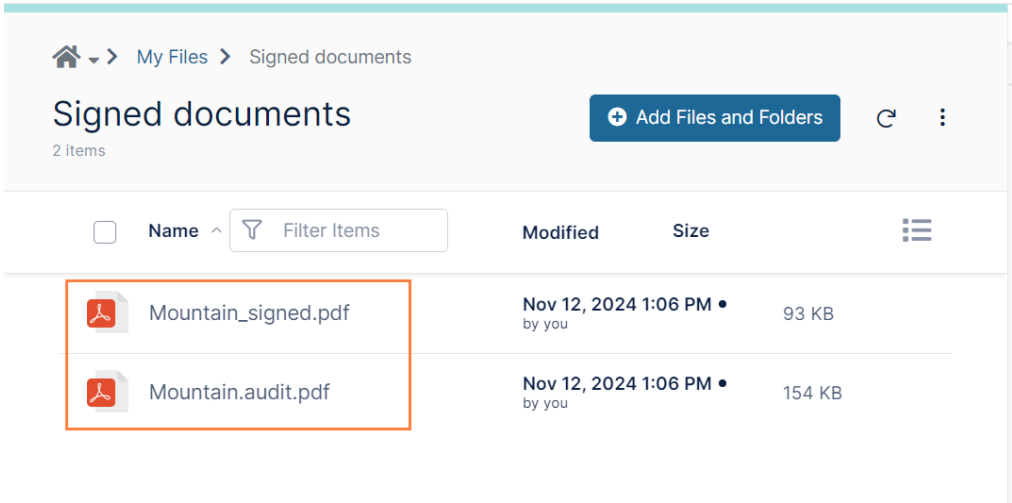
For information about the eSignature process for users, see eSignature Requests.

To return audit details

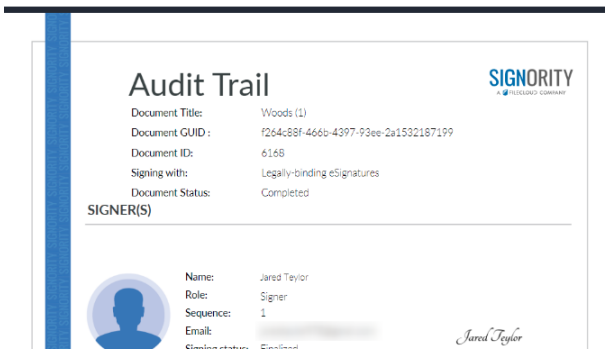
When the files are accessed in FileCloud:

By default, when the eSignature document is returned to the destination path specified by the user or the Completed panel in the eSignature page in FileCloud, the document is returned without audit details. However, you may configure the process to either return a separate audit file or audit details attached to the eSignature document in the destination folder,

When separate signed and audit files are configured, they appear in the destination folder as:



When audit details are attached to the eSignature document, the audit information follows the signature file in the eSignature pdf:



The options for receiving audit details are the following:

Setting code	Description
100	No audit details (default)
101	A separate audit file is sent to the destination with the eSignature document.
102	eSignature: Audit details are attached to the eSignature document. Digital signature: (Applies to Signority only) A separate audit file is sent to the destination with the eSignature document.

To change the audit details setting:

1. Open the cloudconfig.php file
   Windows Location: XAMPP DIRECTORY/htdocs/config/cloudconfig.php
   Linux Location: /var/www/html/config/cloudconfig.php

2. Add the following line, setting the value according to the codes specified above.

```
define('TONIDOCLOUD_ESIGNATURE_DEFAULT_AUDITTRAIL_SETTING', '101');
```

When the files are accessed from the Email notifying the requestor that the file has been signed:

By default, in the email notifying a signature requestor that an eSignature document has been signed, a link to the signed document is included.
For Signority users, if there is an audit details file, by default the email includes separate links to the signed file and the audit details file.

Signority users may sign in to Signority and configure whether or not the audit trail is sent as well as whether multiple signed documents are combined into a single PDF.

To specify in Signority what is included in the email notifying the requestor that the document has been signed:

1. Log in to Signority.
2. In the Signority navigation panel, click **Admin**, and then click **Settings > Global Settings**.
3. In **Global Settings**, locate **Notifications** settings and configure options for the requestor notification email.

Integrating with Signority

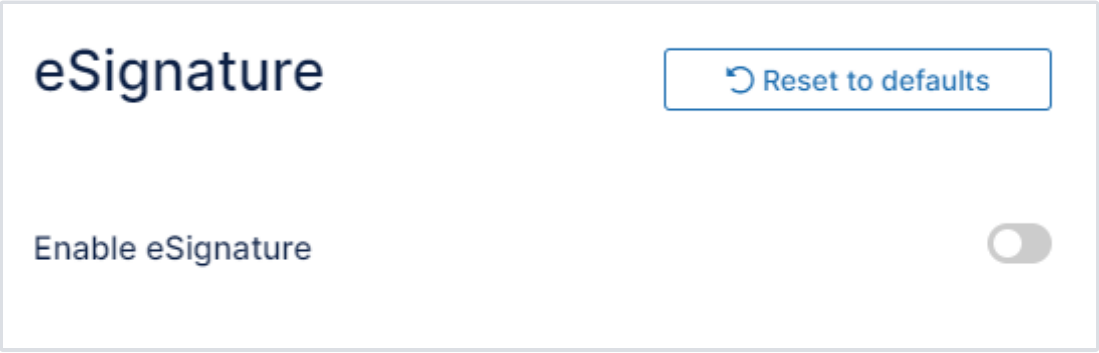
To ensure that your connection to Signority works properly, if you are using a firewall or blocking public connections, please allow requests from 52.60.130.76, Signority's IP address.

Integrate FileCloud with Signority

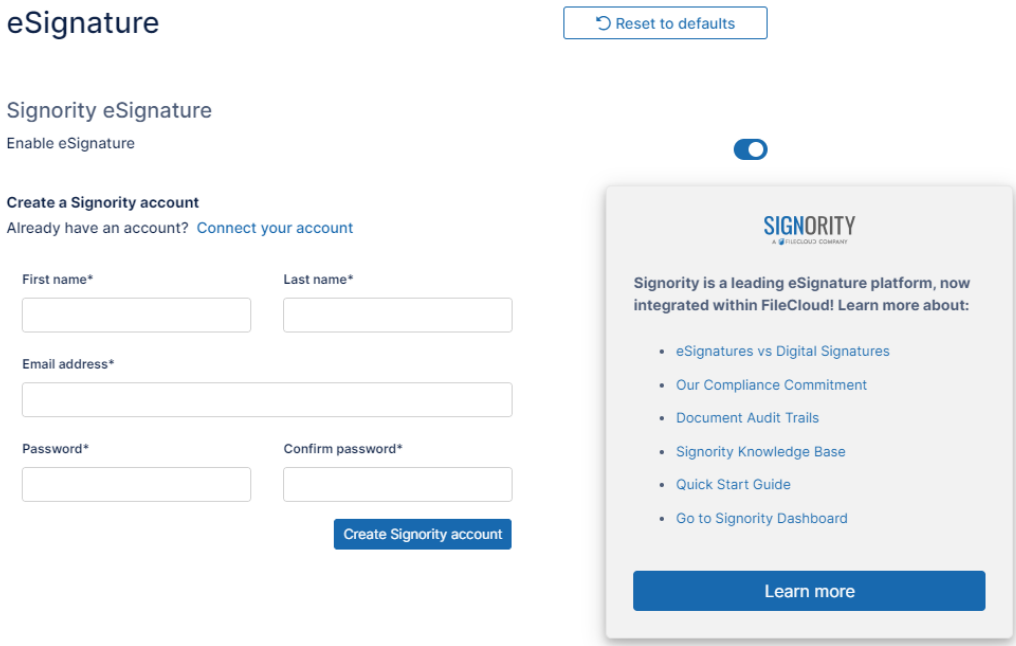
The first time you access the eSignature screen you are required to either obtain a trial Signority account through FileCloud or enter the credentials for your existing Signority account

To enable eSignature:

1. Install and run FileCloud Document Converter if it is not running already. For help, see FileCloud Document Converter.
2. Go to the **eSignature** page.
   The first time you access this page, it appears as:



a. Toggle on the **Enable eSignature** switch.
   By default eSignature is set up to use Signority.
   Fields for creating a trial Signority Account appear along with the link **Connect your account** for users who already have accounts.

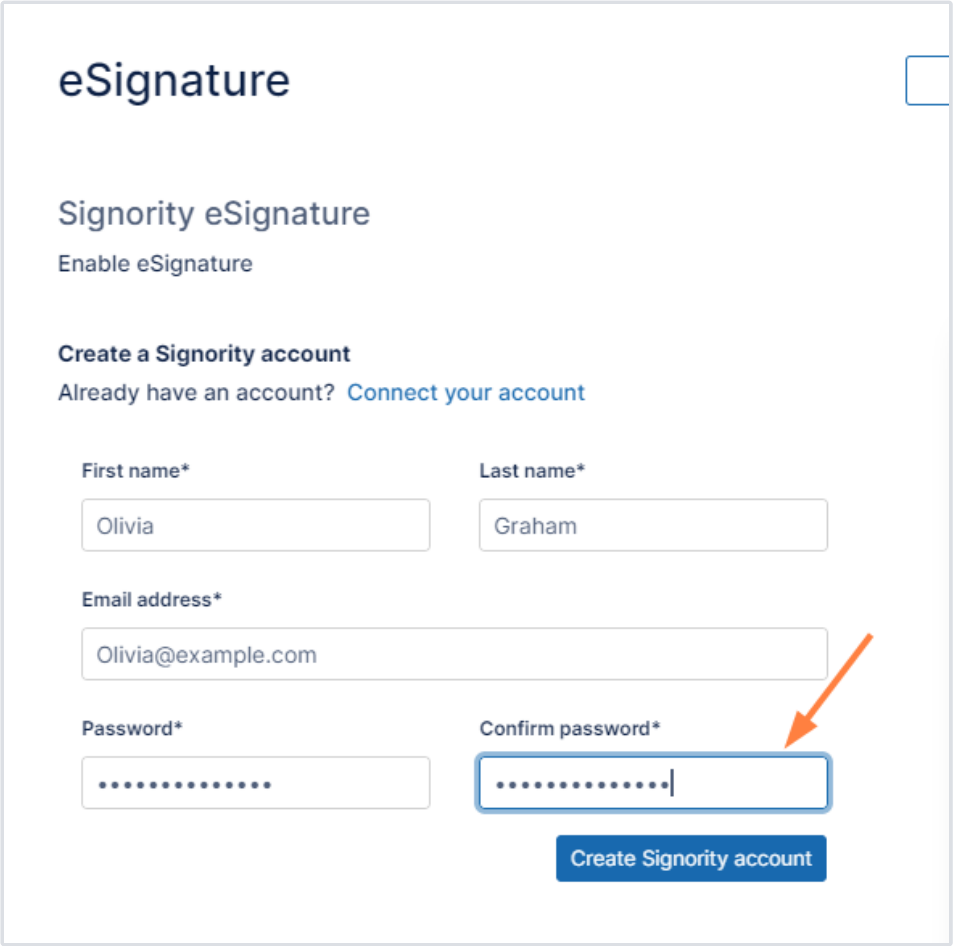


To create a trial Signority account:

1. Enter the information requested, and click **Create Signority Account**.
   Password must be longer than 6 characters and cannot use common words or character strings stored in a predefined dictionary.

   Note: It is recommended that passwords:

   - contain one uppercase letter, one lowercase letter, one number, one special character
   - have a length of 8 - 20 characters



Your Signority account is created. The screen now appears as follows, with **Enable eSignature** checked and your **API Key** automatically filled in.

2. Click **Test** to make sure integration with Signority works correctly.

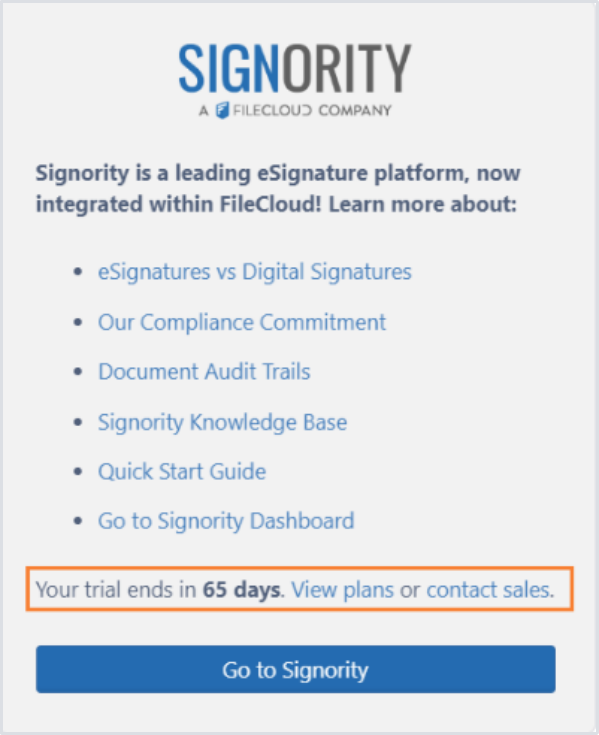
If the test returns a success message, FileCloud integration with Signority is complete, and your users are now able to obtain eSignatures on files.

The trial account lasts for 90 days. You are sent notifications prior to expiration reminding you that before the trial expires, you must purchase a paid account to maintain eSignature capability. If you haven't purchased a paid account by the time your trial account expires, any documents

already sent for eSignature can still be signed and processed; however, no new documents can be sent for eSignatures.

To convert a Signority account from trial to paid

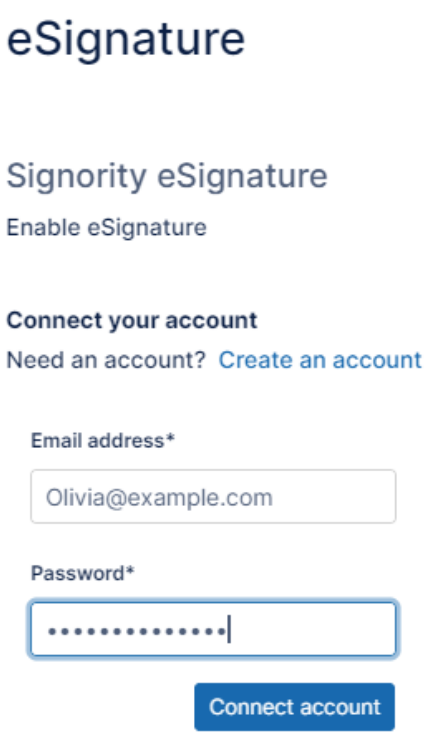
The eSignature screen includes a Signority box that keeps count of the days remaining in your trial account and gives you links for viewing Signority account plans or contacting Signority sales:



Click **contact sales** to proceed with your Signority license purchase.

If you already have a Signority account:

1. Click **Connect your account**.
2. Enter your email address and password into the corresponding fields, and click **Connect Account**.



The screen now appears as follows, with **Enable eSignature** checked and your **API Key** automatically filled in.

3. Click **Test** to make sure integration with Signority works correctly.

If the test returns a success message, FileCloud integration with Signority is complete, and your users are now able to obtain eSignatures on files.

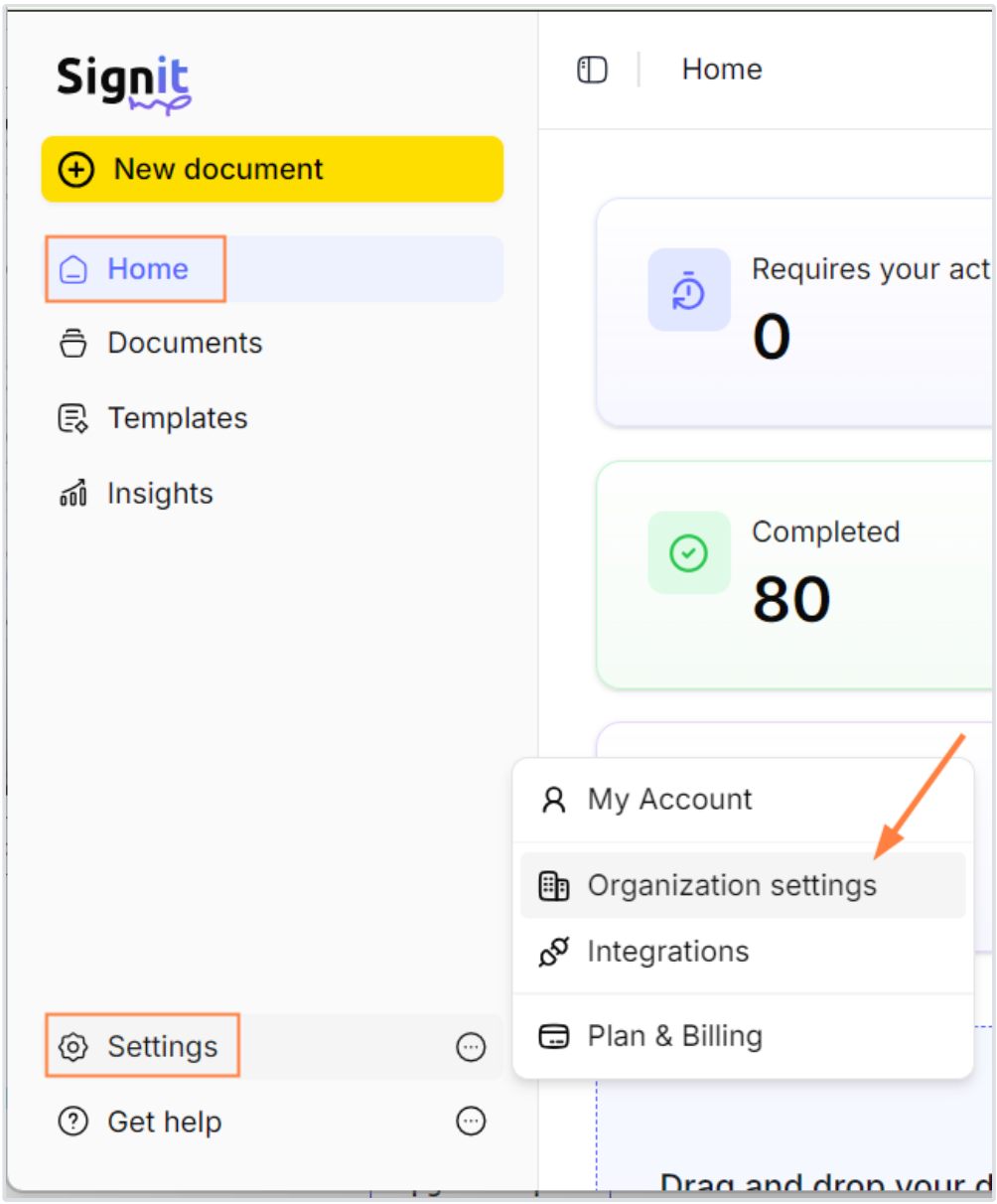
For information about the eSignature process for users, see eSignature Requests.

Integrating with Signit

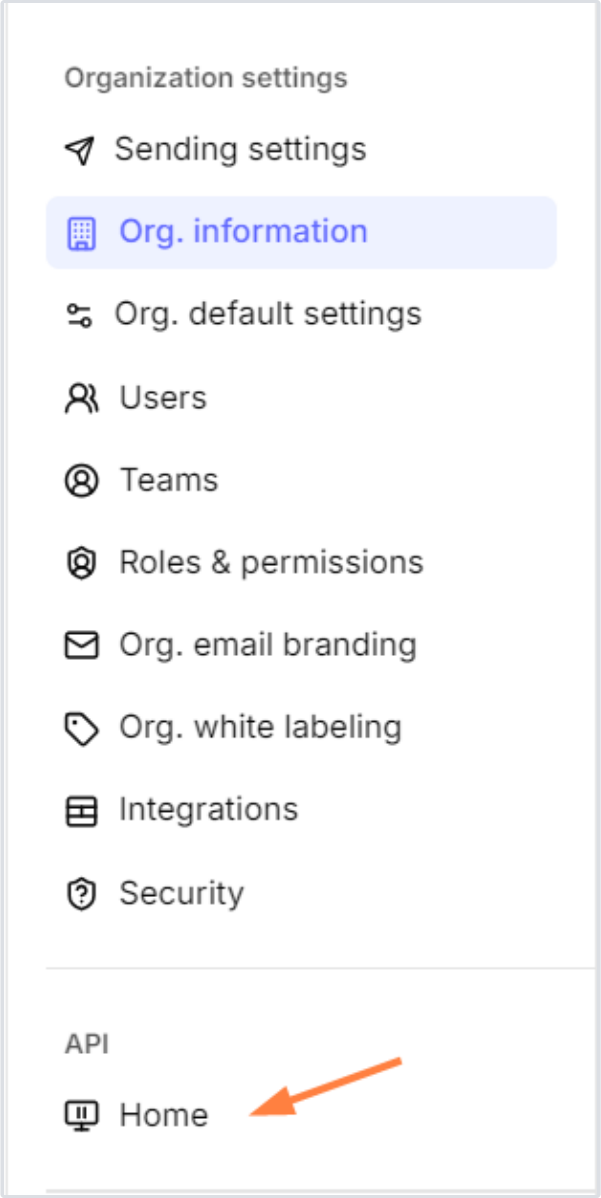
> ⚠ To ensure that your connection to Signit works properly, if you are using a firewall or blocking public connections, please allow requests from 141.147.129.79, Signit's IP address.

To integrate FileCloud with Signit:

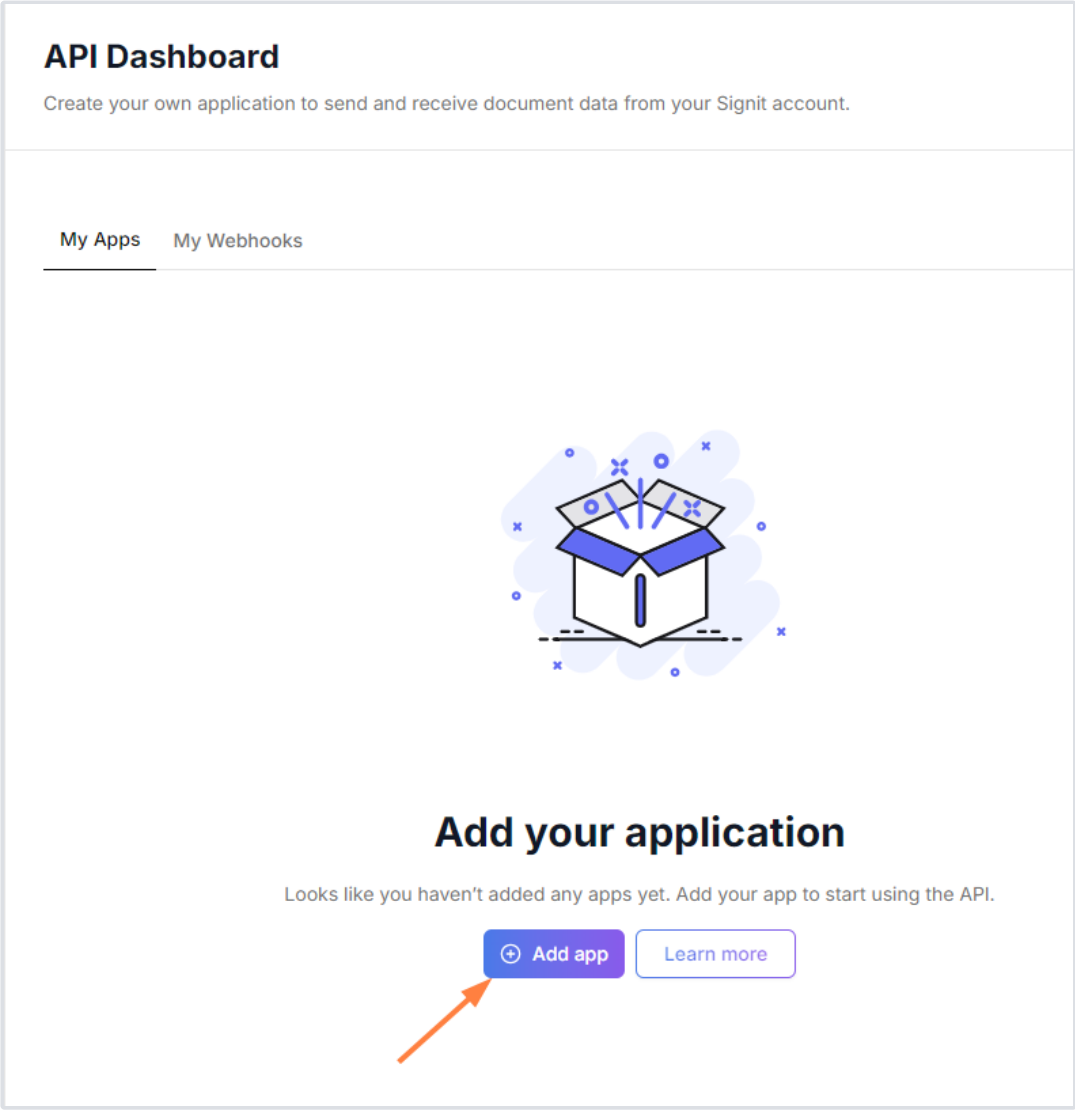
1. If you do not already have a Signit account, create one at https://signit.sa/en.
2. Log in to your Signit account, and create an application for your integration with FileCloud in your Signit API dashboard.
   a. In your Signit dashboard, go to **Home > Settings > Organization settings**.



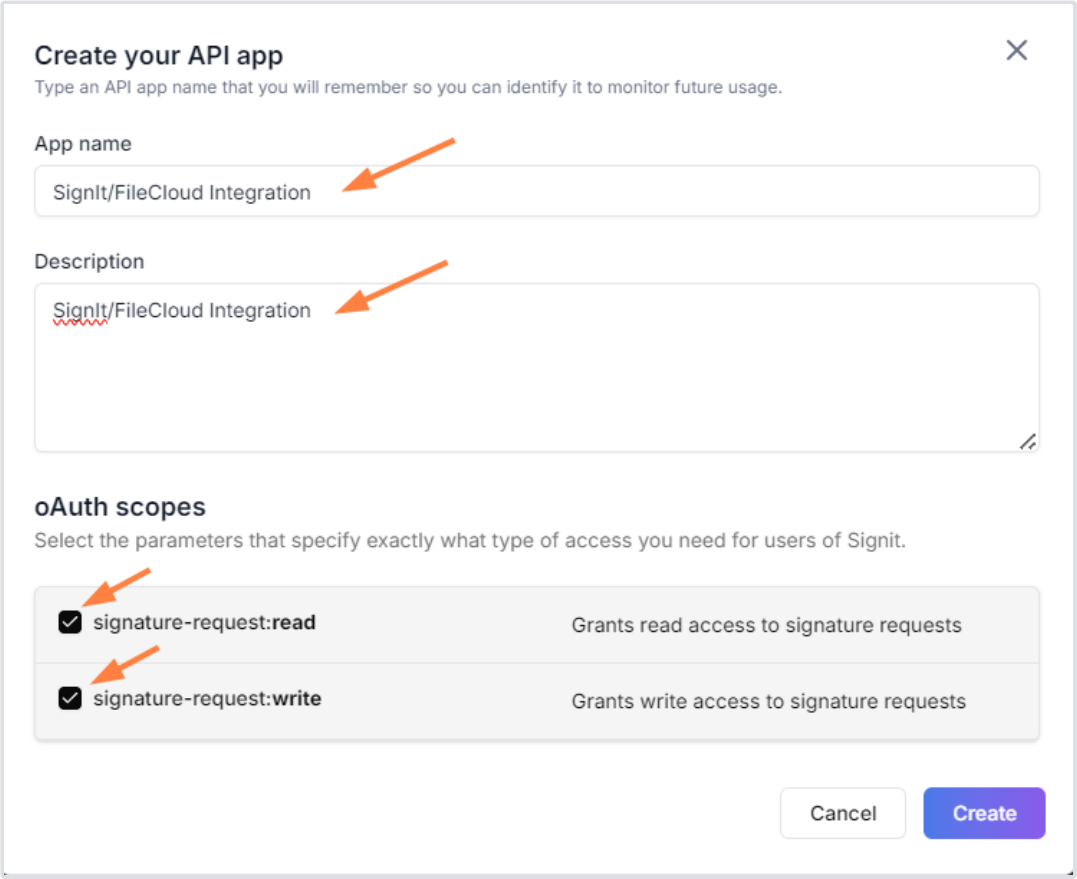
b. In the Organization Settings navigation bar, under **API**, click **Home**.



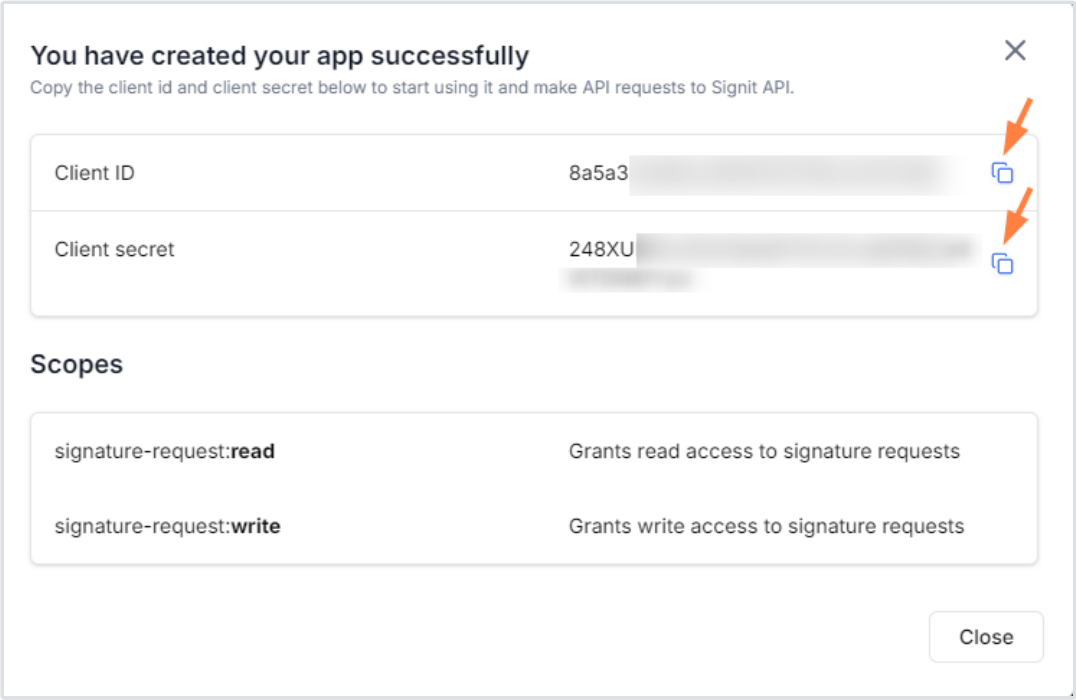
c. In the **My Apps** tab, click **Add app**.



d. In the **Create your API app** dialog box, enter values for **App name** and **Description**, and check **signature-request:read** and **signature-request write**.

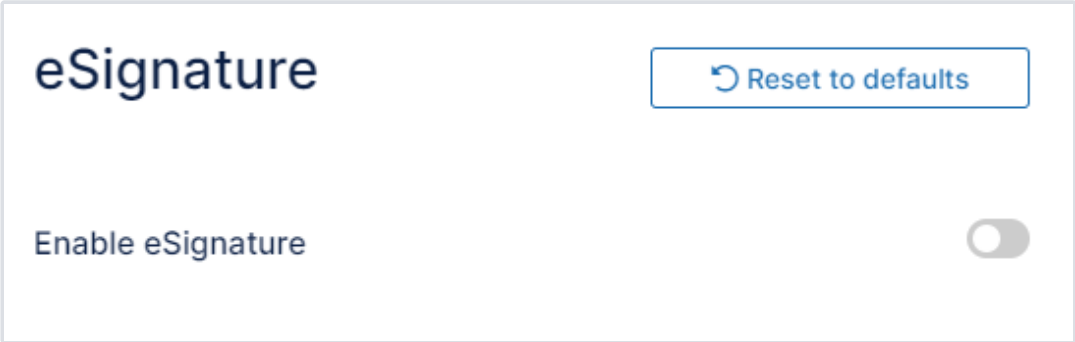


e. Click **Create**.
The dialog box displays your **Client ID** and **Client secret**. It is important that you save these because you will need them to configure the settings for integration in FileCloud.
Note: You will not be able to retrieve the **Client secret** if you do not save it now.

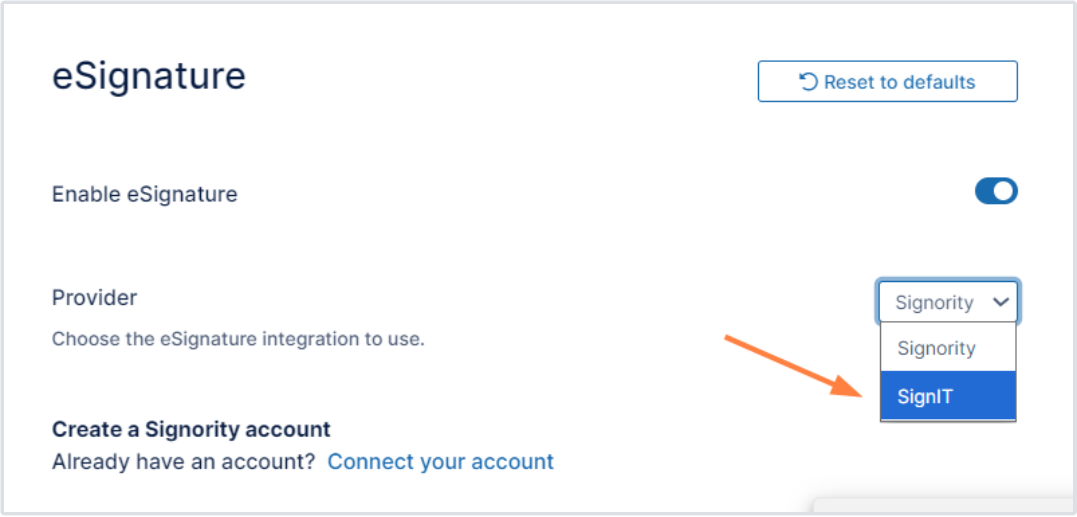


f. Click **Close**. The app is now listed under **My Apps**

3. Log in to the FileCloud admin portal and set up integration with Signit.
   a. In FileCloud, go to **Settings > Third Party Integrations > eSignature**.
   The first time you access this page, it appears as:



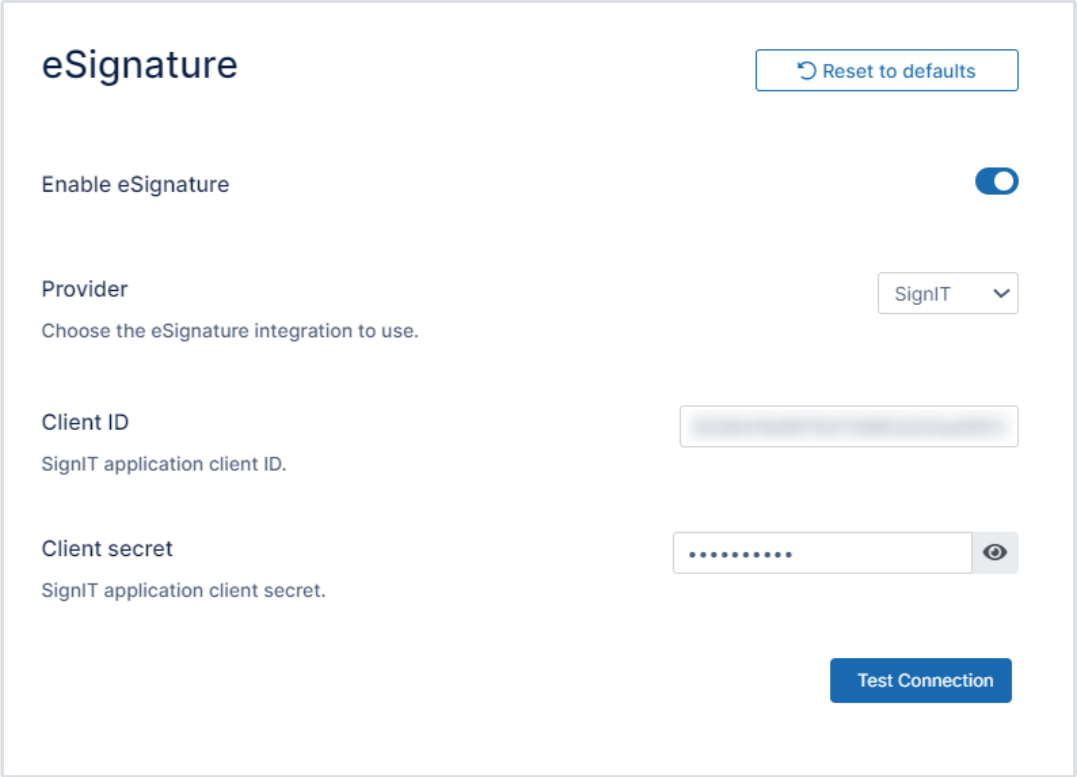
   b. Toggle on the **Enable eSignature** switch.
   By default eSignature is set up to use Signority.
   c. In the **Provider** drop-down list, choose **Signit**.



Now the fields for Signit integration are shown.

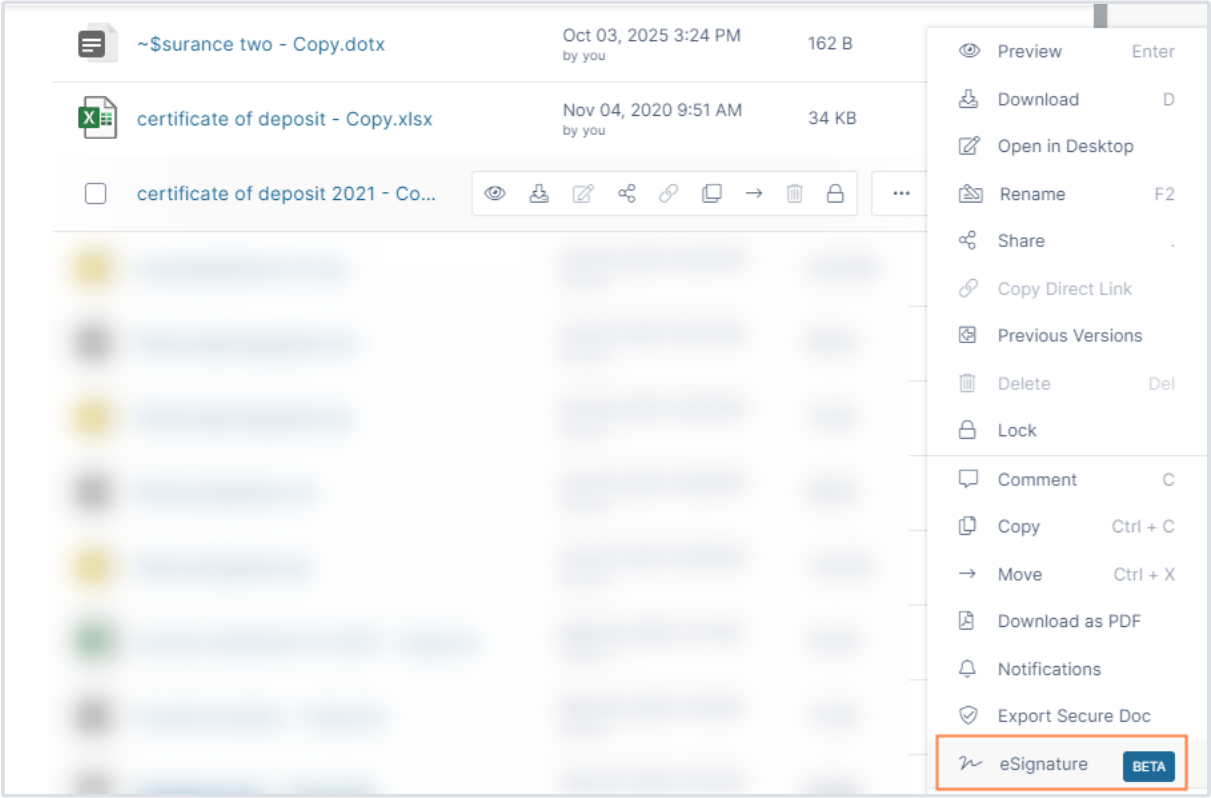
d. In **Client ID**, enter the **Client ID** from your Signit application.

e. In **Client Secret**, enter the **Client Secret** from your Signit application.



f. Click **Test Connection** to confirm that the integration is successful.

Integration of FileCloud and Signit is now complete. In FileCloud, when users click the More icon in the row for a file, the eSignature option appears:


~$urance two - Copy.dotx
Oct 03, 2025 3:24 PM
by you
162 B
certificate of deposit - Copy.xlsx
Nov 04, 2020 9:51 AM
by you
34 KB
certificate of deposit 2021 - Co...
Preview Enter
Download D
Open in Desktop
Rename F2
Share
Copy Direct Link
Previous Versions
Delete Del
Lock
Comment C
Copy Ctrl + C
Move Ctrl + X
Download as PDF
Notifications
Export Secure Doc
eSignature BETA