

Azure Blob Storage for Salesforce: Public Sector and Support Teams

Case folders, signed documents and portal uploads in your own Azure Blob containers, on every Salesforce record

What problem does this solve for public sector and support teams?

CloudFiles is Document Automation for Salesforce. This guide describes two patterns we see with Azure Blob Storage and Salesforce.

The first is a public sector program, such as a housing agency, that tracks applications and tenancies in Salesforce. As an application moves through its statuses, it gathers forms, evidence and signed agreements. The agency runs on Microsoft Azure and wants those documents in its own Azure Blob containers, not in Salesforce storage.

The second is a support team that works Cases in Salesforce. Customers send log files and diagnostic bundles, often through a portal, and engineers need them on the Case without chasing email threads.

The shared problems:

- Salesforce storage fills with files that belong in Azure.
- Staff copy files between Salesforce and Azure by hand.
- Customers and applicants need to upload without an Azure account.
- Files produced outside Salesforce never reach the record.

CloudFiles Document Management is the Azure Blob Storage Salesforce integration for both. It lets you store Salesforce files and attachments in Azure Blob, keep Case folders in Azure Blob, and move signed documents to Azure automatically. The same app connects Salesforce to SharePoint, AWS S3, Google Drive and Azure Blob, so a team that also uses SharePoint or Google Drive works from one widget.

Which Salesforce objects are involved?

- **Housing or public sector:** Account and Contact for applicants and households, a custom **Application** or **Tenancy** object with a Status field.
- **Support:** **Case**, Account and Contact, and the Experience Cloud site customers use.
- **EmailMessage**, for attachments that arrive by email.

CloudFiles works on any standard or custom object, and each record can have its own folder in the container. Folder links survive record renames.

Where are the files stored, and how do we connect Azure Blob?

In the customer's own Azure Blob containers. To connect:

1. Add the CloudFiles Entra gallery app to the tenant.
2. Assign Storage Blob roles to the CloudFiles service principal on the storage account.
3. Restrict network access with an IP allowlist if the storage account requires it.

For event-driven sync, turned on per connection, the customer creates an Event Grid subscription on the storage account, filtered by subject to the containers CloudFiles uses. A file that arrives in the container by any route is indexed and appears on the record. A folder created by CloudFiles appears in the container once a file is uploaded into it, because Azure Blob folders exist through the files inside them.

How do Case folders in Azure Blob work for support teams?

1. **Create the Case folder.** A record-triggered Flow on Case create runs **Create Folder**, named by Case number.
2. **Share it with the customer.** **Share with Community User** shows that Case folder in the portal widget to the customer's community user.
3. **The customer uploads logs from the portal.** The portal widget uploads into Azure Blob. The customer never needs an Azure account.
4. **Engineers work from the Case.** The CloudFiles widget on the Case shows the folder in a tree view with search across the folder tree and previews for PDF, Office files, images, video and audio.
5. **Close out.** On Case close, **Remove from Community** takes the folder out of the portal view.

Files that internal tools write into the Case folder also appear on the Case, through event-driven sync.

How do we move signed documents to Azure on status change?

For the housing pattern:

1. **Folders on status change.** A record-triggered Flow on the Application's Status changing to Under Review runs **Create Folder** for that application.
2. **Signed documents to Azure.** When the Status changes to Signed, the Flow runs **Copy Salesforce Files** to copy the signed agreement from Salesforce Files into the application's Azure Blob folder and leave a link on the record.
3. **Remove the Salesforce copy on completion.** Copying is asynchronous. A platform-event Flow handles the CloudFiles completion event, and only that handler removes the Salesforce file.

4. **Blob metadata. Set Resource Property** writes Application Number, Program and Status onto the blob as metadata. **Search by Resource Metadata** finds files by those values later.
5. **Migrate existing files.** Legacy files and attachments can be migrated in bulk.

Trigger	Flow action	Result
Case created	Create Folder, Share with Community User	Case folder in Azure Blob, visible in the portal
Status set to Signed	Copy Salesforce Files	Signed document in Azure, link on the record
Copy completion event	Delete Records (standard Flow element)	Salesforce storage freed
File stored	Set Resource Property	Blob metadata for search

Who can see the documents?

Inside Salesforce, the widget shows a record's Azure Blob folder to users who can see the record, so Salesforce record access decides who sees which application or Case. Flow automation runs as the CloudFiles service principal; the widget and Content Library run as each user through their own storage connection. On the Azure side, CloudFiles has only the Storage Blob roles you assign and reaches the account only from allowlisted IP addresses. Portal users see only folders a Flow has shared with them.

For outside parties, **Create Link** issues a share link with an email wall, password, email verification code and consent step. Links can expire and carry a view limit, and every view and download is tracked.

What are the security and compliance facts?

- ISO/IEC 27001:2022 certified, with ISO/IEC 27017:2015 and ISO/IEC 27018:2019.
- Independently audited for SOC 2 Type II.
- Compliant with HIPAA (BAA available). Compliant with GDPR.
- Hosted on AWS in four regions selected per customer account: United States, EU, United Kingdom and Australia.
- TLS 1.2 minimum in transit; AES-256 at rest on production data stores.
- Your files stay in your own storage. CloudFiles transfers them over encrypted connections and does not re-host them.

Security documents (ISO certificates, SOC 2 Type II report, penetration test summary) are available under NDA from the CloudFiles Trust Center.

Get started

Install CloudFiles Document Management from AgentExchange and start a free 14-day trial, or book a demo at cloudfiles.io.

CloudFiles, Document Automation for Salesforce · cloudfiles.io · Install CloudFiles Document Management from AgentExchange