

DocHub Customer Onboarding Runbook

Version: 1.0 **Audience:** Salesforce administrator + Microsoft 365 administrator (work together) **Time required:** 60–90 minutes from start to first successful document edit **Last updated:** 2026-05-21

Table of Contents

- [0. Before you begin](#)
 - [1. Prerequisites](#)
 - [2. Microsoft Entra app registration](#)
 - [3. SharePoint staging site setup](#)
 - [4. Salesforce package installation](#)
 - [5. Configure DocHub Custom Metadata](#)
 - [6. Salesforce Connected App for save-back](#)
 - [7. Network configuration](#)
 - [8. AWS-side provisioning by DocHub](#)
 - [9. End-to-end verification](#)
 - [10. Troubleshooting](#)
 - [11. Operations and maintenance](#)
 - [12. Appendix](#)
-

0. Before you begin

What this runbook gets you

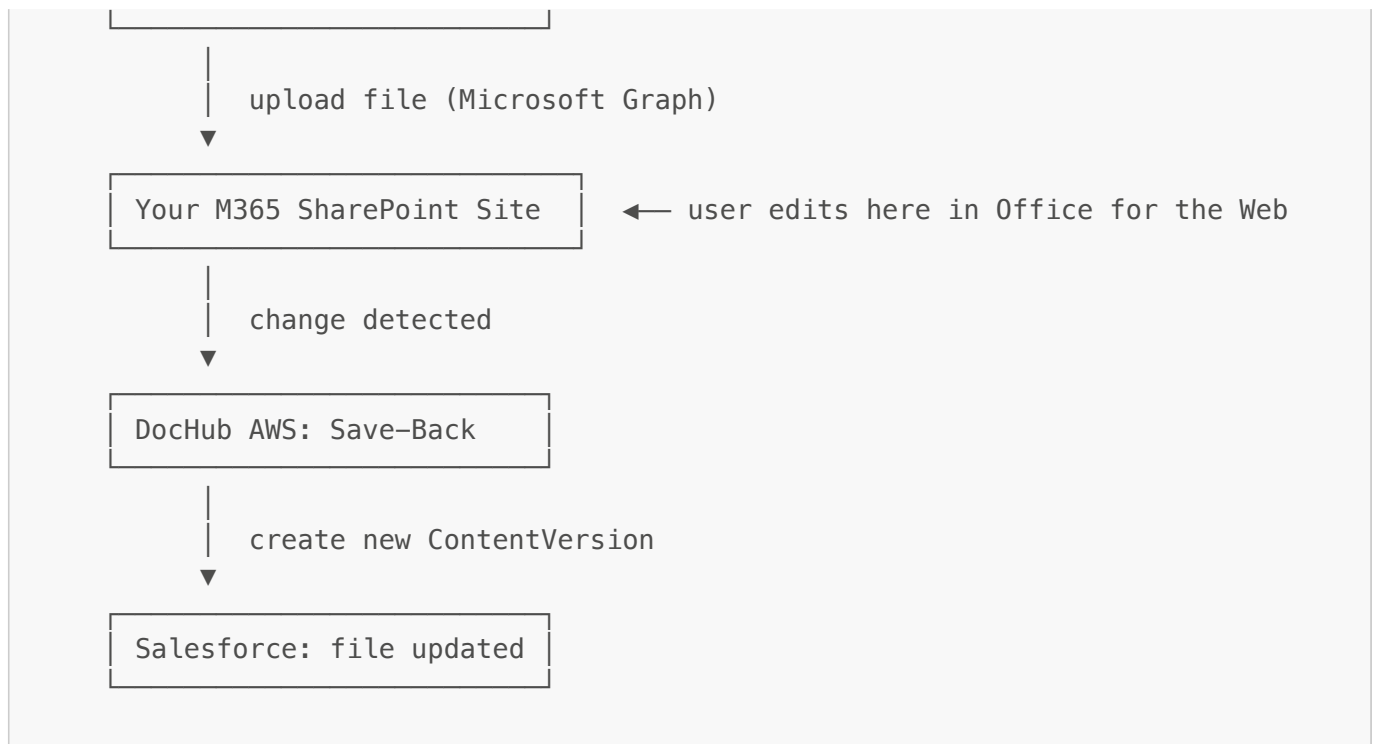
By the end of this document, your users will be able to:

- Click "Edit" on a **.docx** file attached to any Salesforce record
- Edit the file in Microsoft Word for the Web (in their browser)
- Save and close — the updated version appears back in Salesforce within 60–90 seconds

The complete edit history is preserved as Salesforce ContentVersions. Salesforce remains the system of record for your files; Microsoft 365 is used as an editing workspace only.

Architecture in 60 seconds





Files transit through your own Microsoft 365 tenant. DocHub-managed AWS infrastructure handles orchestration, but files are stored in **your** SharePoint, not ours.

What this runbook does NOT cover

- **Government cloud variants** (GCC High, DoD, US Government) — DocHub v1.0 supports Microsoft 365 Commercial and GCC only
- **Per-user OAuth identity mode** — DocHub v1.0 uses a single service-account identity per customer
- **File types beyond Office formats** — **.docx**, **.xlsx**, **.pptx** only in v1.0
- **AppExchange installation procedure** — covered in your AppExchange listing
- **Multi-region SharePoint** — single staging site per customer

Who you'll need

Two people, working together for ~90 minutes:

Role	Who	What they'll do
Salesforce Administrator	Your internal SF admin	Install package, configure Custom Metadata, assign permissions
Microsoft 365 Administrator	Your internal M365 admin	Register Entra app, create SharePoint site, grant permissions

If these are the same person, allocate the full 90 minutes.

What DocHub will provision for you

After you complete sections 1–7, you'll send a provisioning request to DocHub Support. DocHub will provision two things on the AWS backend specific to your customer org and reply within 1–2 business days with:

1. An AWS API endpoint URL specific to your org

2. A pair of AWS access credentials (access key + secret) for your Salesforce org to authenticate with the API

You'll plug these into Salesforce in section 8.

Conventions in this document

- **Setup → Custom Metadata Types** means: in Salesforce, navigate to Setup, then to Custom Metadata Types
- **[BRACKETED_VALUES]** are placeholders you replace with your own values
-  checkmarks indicate verification steps — do not skip them
-  warnings indicate steps where mistakes are difficult to undo
-  notes provide context or reasoning for the step

1. Prerequisites

Before starting, verify you have all of the following. Skipping a prerequisite here causes failures later that are difficult to diagnose.

Microsoft 365 requirements

Tenant type — one of:

- Microsoft 365 Commercial (most common)
- Microsoft 365 GCC

Other variants (GCC High, DoD, US Government) are not supported in v1.0.

Licenses required:

For	Minimum license
Service account (1 user)	Microsoft 365 Business Standard or higher, with Office for the Web enabled
Each end user editing files	Same as above — Office for the Web access required

Confirm Office for the Web is enabled for your tenant:

1. Go to Microsoft 365 admin center → Settings → Org settings
2. Find "Microsoft 365 on the web"
3. Confirm "Let users open files in Microsoft 365 on the web" is enabled

SharePoint Online requirements:

- SharePoint Online must be active on the tenant (default in any M365 business plan)
- You'll create a dedicated staging site in section 3

Admin permissions in M365:

You need one M365 administrator with **one** of the following role combinations:

- Global Administrator, OR
- Cloud Application Administrator + SharePoint Administrator combined

The administrator must be able to consent to API permissions on behalf of the tenant.

✅ **Verify M365 admin access:** log in to <https://admin.microsoft.com> and confirm "Microsoft Entra ID" appears in the left navigation. If absent, you don't have sufficient permissions.

Salesforce requirements

Edition — one of:

- Enterprise Edition
- Unlimited Edition
- Performance Edition
- Developer Edition (for evaluation)

Professional Edition is not supported (no managed package install access).

Salesforce API version: Your org must be on API version 60.0 or higher. All orgs created after Spring '24 meet this automatically; older orgs were auto-upgraded.

Admin permissions in Salesforce:

- System Administrator profile, OR
- A custom profile with "Install Packages", "Customize Application", and "Modify All Data" enabled

Salesforce Files:

- Salesforce Files must be enabled (default — only disabled by explicit admin action)
- Your users must have access to ContentDocument and ContentVersion objects via their profiles or permission sets

✅ **Verify Salesforce admin access:** log in to your Salesforce org and confirm you can navigate to Setup → Apps → Packaging → Installed Packages without an access error.

Network and firewall

If your organization's network restricts outbound traffic from end-user browsers, ensure the following domains are allowlisted:

Domain	Purpose
*.execute-api.us-west-2.amazonaws.com	DocHub AWS API endpoint
*.officeapps.live.com	Office for the Web editor UI
*.sharepoint.com	SharePoint file delivery
login.microsoftonline.com	Microsoft authentication

Add these to your network allowlist **before starting**. End users behind restrictive proxies will see "connection failed" errors otherwise.

Account preparation

You'll need to create or identify two M365 accounts before starting section 2:

Account A — Service account:

- A real M365 user (not a shared mailbox), with a regular license
- Dedicated to DocHub — should not be a person's account
- Suggested naming: `dochub-svc@[YOUR_DOMAIN]` or similar
- Will own the SharePoint staging site
- Will perform file uploads on behalf of all DocHub users

⚠ Make a note of the service account's UPN (user principal name) — you'll need it in sections 3 and 8.

Account B — Test user:

- A real M365-licensed user (could be yourself for testing)
- Will be used to verify the end-to-end flow in section 9

Information to gather

Before starting, have these values written down. You'll be asked for them in later sections:

Value	Where to get it	Format example
Salesforce org ID (18-character form)	Salesforce Setup → Company Information, OR run <code>System.debug(UserInfo.getOrganizationId());</code> in Developer Console → Execute Anonymous	<code>00Dxx00000000000XYZ</code> (18 chars, alphanumeric)
Microsoft Entra tenant ID	Microsoft Entra admin center → Overview	GUID format: <code>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</code>
Your SharePoint domain	Your tenant's SharePoint root URL	<code>[YOUR_DOMAIN].sharepoint.com</code>

💡 The 15-character form of the Salesforce org ID is also visible in Setup, but DocHub requires the 18-character form. Always use Execute Anonymous to retrieve the 18-character form to be safe.

✅ **Before proceeding to section 2**, confirm you have:

- ☐ Both Salesforce and M365 admin access verified
- ☐ Service account UPN noted
- ☐ All three values above written down
- ☐ ~90 minutes of focused time blocked off
- ☐ Your DocHub support contact email
- ☐ Network domains allowlisted (if applicable to your org)

2. Microsoft Entra app registration

Performed by: Microsoft 365 administrator **Estimated time:** 15 minutes

You'll create a new application registration in Microsoft Entra that DocHub uses to authenticate with Microsoft Graph and access your SharePoint site. The app is granted access to **one specific SharePoint site only** — not your entire tenant.

Why this matters for security

DocHub uses Microsoft Graph's **Sites.Selected** permission scope. This is a deliberately narrow permission: even though your tenant may have many SharePoint sites, this app can only access the single site you explicitly grant in section 3. If DocHub credentials were ever compromised, the blast radius is limited to one site.

💡 This is more restrictive than the typical **Files.ReadWrite.All** permission many integrations request. DocHub chose this trade-off deliberately for enterprise security posture.

2.1 Create the app registration

1. Navigate to **Microsoft Entra admin center**: <https://entra.microsoft.com>
2. In the left navigation: **Applications → App registrations**
3. Click **+ New registration**

Fill in the form:

Field	Value
Name	DocHub Integration (or any name you'll recognize later)
Supported account types	Accounts in this organizational directory only (Single tenant)
Redirect URI	Leave empty — do not configure a redirect URI

Click **Register**.

✅ You should land on the app's "Overview" page showing the new app.

2.2 Record your credentials

On the Overview page, find and copy these two values:

Value	Format
Application (client) ID	GUID format
Directory (tenant) ID	GUID format

Save both to your secure notes. You'll need them in section 5 (Salesforce configuration).

2.3 Generate a client secret

1. In the left navigation of the app: **Manage → Certificates & secrets**
2. Under the "Client secrets" tab, click **+ New client secret**
3. Fill in:

Field	Value
Description	DocHub production secret
Expires	24 months (max). Set a calendar reminder to rotate before this date.

4. Click **Add**

⚠ Important: The secret VALUE is shown ONLY ONCE on this screen. Copy it immediately to your secure password manager.

Value	Format
Client secret value	40–50 character alphanumeric string with special characters

You'll need this in section 8 (sent to DocHub Support). If you lose it before saving, you must generate a new one and start this subsection over.

2.4 Add Microsoft Graph permissions

1. In the left navigation of the app: **Manage → API permissions**
2. Click **+ Add a permission**
3. Select **Microsoft Graph**
4. Select **Application permissions** — **⚠** NOT delegated permissions; this distinction matters
5. In the search box, type: **Sites.Selected**
6. Check the box next to **Sites.Selected**
7. Click **Add permissions** at the bottom

You should now see **Sites.Selected** listed under "Configured permissions" with status "Not granted for [Your Tenant]".

2.5 Grant admin consent

The permission was added but is not yet active. An administrator must consent on behalf of the tenant.

1. On the "API permissions" page, click **Grant admin consent for [Your Tenant Name]**
2. Confirm the prompt
3. Wait for the page to refresh

✓ The status next to **Sites.Selected** should change to **Granted for [Your Tenant Name]** with a green checkmark.

If you see "Not granted" or a yellow warning, you don't have sufficient permissions to grant tenant-wide consent. You'll need a Global Administrator to perform this step.

2.6 Verify the app is registered correctly

Before moving on, sanity-check the configuration:

Check	Expected value
App appears in App Registrations list	Yes
Application (client) ID recorded	Yes, in your secure notes
Directory (tenant) ID recorded	Yes, in your secure notes
Client secret recorded	Yes, in your secure password manager
Permission Sites.Selected	Configured
Permission type	Application (not Delegated)

Check	Expected value
Permission status	"Granted for [Tenant]" with green check

✅ All seven checks pass before proceeding to section 3.

Troubleshooting

"Grant admin consent" button is greyed out. You don't have permissions to grant tenant-wide consent. Required: Global Administrator, Privileged Role Administrator, or Cloud Application Administrator role. Ask your tenant admin to perform step 2.5 only — they don't need to redo the registration.

Permission status says "Granted" but you see a warning icon. Check that you selected **Application permissions**, not **Delegated permissions**. DocHub uses app-only authentication. If you accidentally added the delegated version of **Sites.Selected**, remove it and re-add as Application permission.

Multiple Sites.Selected results appear in search. Verify you're under Microsoft Graph (not SharePoint or another API). The correct one is under Microsoft Graph → Application permissions.

3. SharePoint staging site setup

Performed by: Microsoft 365 administrator (with SharePoint administrator if those are different people)

Estimated time: 20 minutes

You'll create a dedicated SharePoint site that DocHub uses as temporary storage when users edit files. Files are uploaded here when an edit session starts, edited in place by users, then deleted after the new version is saved back to Salesforce. End users never see this site directly.

3.1 Create the staging site

1. Navigate to **SharePoint admin center**: [https://\[YOUR_DOMAIN\]-admin.sharepoint.com](https://[YOUR_DOMAIN]-admin.sharepoint.com)
2. In the left navigation: **Sites → Active sites**
3. Click **+ Create**
4. Choose **Team site** (not Communication site)
5. Fill in:

Field	Value
Site name	DocHub Staging
Group email address	Auto-generated from name
Site address	DocHubStaging — becomes https://[YOUR_DOMAIN].sharepoint.com/sites/DocHubStaging
Privacy settings	Private — only members can access
Time zone	Your business time zone
Site owner	The service account from section 1 (Account A)

6. Click **Next**

When prompted for "Add additional owners or members":

- **Owners:** just the service account (no other users)
- **Members:** leave empty

⚠ Do NOT add end users here — they don't need direct site access.

7. Click **Finish**

Wait 30–60 seconds for SharePoint to provision the site.

✅ The site appears in your "Active sites" list with status "Active".

3.2 Retrieve the SharePoint Site ID

You'll need the SharePoint Site ID (not the friendly URL) for section 5.

1. Open a new browser tab and go to **Microsoft Graph Explorer**:
<https://developer.microsoft.com/graph/graph-explorer>
2. Sign in with your M365 administrator account
3. Click **Modify permissions** (if shown) and consent to **Sites.Read.All**, **Sites.ReadWrite.All**, and **Sites.FullControl.All** for your Graph Explorer session
4. In the query bar at the top:
 - HTTP method: **GET**
 - URL:
[https://graph.microsoft.com/v1.0/sites/\[YOUR_DOMAIN\].sharepoint.com:/sites/DocHubStaging](https://graph.microsoft.com/v1.0/sites/[YOUR_DOMAIN].sharepoint.com:/sites/DocHubStaging)
 - Replace **[YOUR_DOMAIN]** with your tenant's SharePoint domain
5. Click **Run query**

You should get a 200 response with JSON like:

```
{
  "id": "[YOUR_DOMAIN].sharepoint.com,[SITE_COLLECTION_GUID],[WEB_GUID]",
  "displayName": "DocHub Staging",
  ...
}
```

⚠ **Copy the entire `id` value** including the commas. It's three parts joined with commas (hostname, site collection GUID, web GUID) and DocHub needs all of it.

Value	Format
SharePoint Site ID	<code>[hostname],[guid],[guid]</code> — comma-separated, 3 parts

Save this to your notes.

3.3 Retrieve the Drive ID

Each SharePoint site has a default document library (called a "drive" in Graph terminology). DocHub stages files in this default drive.

1. Still in Graph Explorer, change the URL to:

```
https://graph.microsoft.com/v1.0/sites/[YOUR_SITE_ID]/drive
```

Replace `[YOUR_SITE_ID]` with the value you saved in 3.2 (the full comma-separated value).

2. Click **Run query**

You should get a response like:

```
{
  "id": "b!...",
  "name": "Documents",
  "driveType": "documentLibrary",
  ...
}
```

Value	Format
SharePoint Drive ID	b! followed by a long base64-encoded string

Save this to your notes.

3.4 Grant the Entra app access to this site

This is the critical security step. Your Entra app has the `Sites.Selected` permission, but is not yet granted access to any specific site. We grant it access to ONLY the DocHub staging site.

You'll do this via Microsoft Graph because there's no UI for `Sites.Selected` grants.

1. In Graph Explorer, change the URL to:

```
https://graph.microsoft.com/v1.0/sites/[YOUR_SITE_ID]/permissions
```

2. Change HTTP method to **POST**
3. In the request body tab, paste:

```
{
  "roles": ["write"],
  "grantedToIdentities": [
    {
      "application": {
        "id": "[YOUR_CLIENT_ID_FROM_SECTION_2.2]",
        "displayName": "DocHub Integration"
      }
    }
  ]
}
```

```
]
}
```

4. Replace `[YOUR_CLIENT_ID_FROM_SECTION_2.2]` with your application (client) ID
5. Click **Run query**

You should get a 201 response with a permission object showing the grant.

3.5 Verify the grant

1. Change method back to **GET** and re-run the query (without the body)
2. Confirm you see at least two entries in the response:
 - The site's owner group (always present)
 - Your DocHub Integration app with `roles: ["write"]`

If both are present, the grant is working.

✅ **Before moving on**, you have:

- ☐ SharePoint staging site created and active
- ☐ Service account is the site owner
- ☐ Site ID recorded (full comma-separated form)
- ☐ Drive ID recorded
- ☐ Entra app granted write access to this site
- ☐ Permission grant verified via Graph

Troubleshooting

Graph Explorer returns "Insufficient privileges to complete the operation". You haven't consented to the required Graph permissions in Graph Explorer itself. Click "Modify permissions" and consent to `Sites.Read.All`, `Sites.ReadWrite.All`, and `Sites.FullControl.All`. These are YOUR user permissions in Graph Explorer, separate from the app permissions you configured in section 2.

Graph returns 404 when you GET the site by URL. Your SharePoint domain or site name doesn't match. Check the URL format: `[YOUR_DOMAIN].sharepoint.com:/sites/DocHubStaging`. The colon and `/sites/` matter. Common error: forgetting the colon.

The permissions POST returns 403. Either you don't have `Sites.FullControl.All` consented in Graph Explorer, OR you're trying to grant a permission on a site you don't own. Verify both.

You see the site in SharePoint admin but it doesn't appear in Graph. SharePoint sites take 1–5 minutes to fully provision and index in Graph. Wait 2 minutes and retry.

Drive ID looks different from the example. Drive IDs start with `b!` followed by a long base64-encoded string. The exact length varies. As long as it starts with `b!`, it's correct.

4. Salesforce package installation

Performed by: Salesforce administrator **Estimated time:** 10 minutes

4.1 Install the package

You should have received a package installation URL from DocHub. It looks like:

```
https://login.salesforce.com/packaging/installPackage.apexp?p0=[PACKAGE_VERSION_ID]
```

The `04t...` identifier in the URL is the specific package version. Newer DocHub versions will have different IDs.

1. While logged in to your target Salesforce org, paste the install URL into the browser
2. You'll be prompted with: "Install DocHub for [Your Org]"
3. Choose **Install for All Users**
4. Click **Install**

⚠ Do NOT choose "Install for Admins Only" — end users need access to the LWC components. Permission control is handled by the package's permission sets in section 4.3, not by who has the package metadata.

Salesforce will show "This app is taking a long time to install" if it takes more than ~30 seconds. This is normal. The package will continue installing in the background, and you'll receive an email when complete (usually within 5 minutes).

✅ Confirm installation completed: Setup → Apps → Packaging → **Installed Packages**. You should see:

Package	Namespace	Version
DocHub	<code>dochub</code>	(the version you installed)

4.2 Verify installed components

Setup → Apps → Packaging → Installed Packages → click **DocHub** → click **View Components**.

You should see these component types (counts may vary slightly by version):

Component Type	Approximate count
Apex Class	~10–15
Custom Label	7
Custom Metadata Type	2
External Credential	1
Lightning Web Component	1 (<code>filesPanel</code>)
Named Credential	1
Permission Set	2
Platform Event	1

✅ All listed component types present. If any are missing, the install was incomplete — contact DocHub Support before proceeding.

4.3 Assign permission sets

DocHub ships two permission sets:

Permission Set	Who gets it
DocHubAdmin	You (the Salesforce admin), and anyone managing DocHub configuration
DocHubUser	All end users who will edit files via DocHub

💡 The **DocHubAdmin** permission set includes everything in **DocHubUser** plus configuration access. You don't need to assign both to the same user.

Assign **DocHubAdmin** to yourself:

1. Setup → Users → Permission Sets
2. Click **DocHubAdmin**
3. Click **Manage Assignments** → **Add Assignments**
4. Select your user → **Assign**

Assign **DocHubUser** to end users:

For a small number of users, repeat the above with **DocHubUser**. For many users, use the permission set group assignment workflow your org already uses for other packages.

✅ Confirm at least your own admin user has **DocHubAdmin** assigned. You'll need it for the next section.

5. Configure DocHub Custom Metadata

Performed by: Salesforce administrator **Estimated time:** 10 minutes

This is where the values you collected from sections 2 and 3 get plugged into Salesforce.

5.1 Locate the Config Custom Metadata Type

1. Setup → Custom Code → Custom Metadata Types
2. You should see **Config** (managed by DocHub) — click **Manage Records** next to it
3. You should see one record: **Default** — click **Edit**

💡 DocHub uses a single Config record (named **Default**) per org. Multi-tenancy on the AWS side keys off your Salesforce Org ID, not multiple records here.

5.2 Fill in the Config fields

You'll set 9 fields. Here's what each requires, in the order they appear:

Field API name	What it is	Value to enter
EntraTenantId__c	Your Microsoft Entra tenant ID	From section 2.2 (GUID format)
EntraClientId__c	Your DocHub Integration app's Application (client) ID	From section 2.2 (GUID format)

Field API name	What it is	Value to enter
<code>SharePointSiteId__c</code>	Your DocHub Staging site's Site ID	From section 3.2 (comma-separated form)
<code>SharePointDriveId__c</code>	The site's default drive ID	From section 3.3 (starts with b!)
<code>M365Endpoint__c</code>	Microsoft 365 cloud type	Commercial for standard M365, GCC for Government Community Cloud
<code>ServiceAccountUserPrincipalName__c</code>	The service account UPN from section 1	e.g. dochub-svc@[YOUR_DOMAIN]
<code>LockTtlMinutes__c</code>	How long a document edit lock holds before auto-expiring	30 (recommended default)
<code>SessionTtlHours__c</code>	How long a stale edit session persists before sweeper cleanup	24 (recommended default)
<code>IdentityMode__c</code>	Authentication mode	ServiceAccount — only supported value in v1.0

⚠ The Entra Client Secret (from section 2.3) is NOT entered here. That value goes to DocHub Support via the secure channel in section 8 — it never enters Salesforce.

⚠ Do NOT enter your Salesforce Connected App's consumer key/secret here either. Those are configured in section 6 and also sent to DocHub Support securely.

5.3 Save and verify

1. Click **Save** at the bottom of the form
2. Salesforce returns to the Config records list — your **Default** record should show all fields populated
3. Click **Default** to view the saved record
4. Confirm every value displays as expected

✅ Field-by-field verification:

- ☐ `EntraTenantId__c` matches the value from section 2.2
- ☐ `EntraClientId__c` matches the value from section 2.2
- ☐ `SharePointSiteId__c` is the full comma-separated form from section 3.2
- ☐ `SharePointDriveId__c` starts with **b!**
- ☐ `M365Endpoint__c` is **Commercial** or **GCC**
- ☐ `ServiceAccountUserPrincipalName__c` is the service account UPN
- ☐ `LockTtlMinutes__c` is **30**
- ☐ `SessionTtlHours__c` is **24**
- ☐ `IdentityMode__c` is **ServiceAccount**

Troubleshooting

"You can't deploy this Custom Metadata record" error. You don't have the **DocHubAdmin** permission set. Assign it (section 4.3) and try again.

M365Endpoint__c picklist shows only Commercial. GCC support requires DocHub-side provisioning in a different AWS region. If you need GCC support, mention this in your provisioning request to DocHub Support in section 8.

SharePointSiteId__c field rejects your input. The field has a minimum length. Ensure you copied the entire comma-separated Site ID from Graph Explorer, including all three parts.

6. Salesforce Connected App for save-back

Performed by: Salesforce administrator **Estimated time:** 15 minutes

The DocHub AWS service needs to call back into your Salesforce org to create new ContentVersions after edits. This requires a Connected App configured with the **client_credentials** OAuth flow.

💡 In DocHub v1.0, the Connected App is created manually via the Salesforce UI rather than packaged with the DocHub managed package. This is a Salesforce platform limitation: Connected Apps that ship with packages are constrained in ways that prevent automated provisioning.

6.1 Create the Connected App via External Client App Manager

Salesforce offers two ways to create an OAuth app: the legacy Connected App Manager and the newer External Client App Manager. Use **External Client App Manager** — it has fewer restrictions in Partner DE and customer orgs.

1. Setup → Apps → **External Client App Manager**
2. Click **New External Client App**
3. Fill in:

Field	Value
External Client App Name	DocHub Save-Back
API Name	DocHub_Save_Back (auto-filled)
Distribution State	Local (single-org use)
Contact Email	Your admin email
Description	OAuth client for DocHub AWS to call Salesforce REST API for save-back

4. Click **Create**

6.2 Enable OAuth settings

After creating the app, you'll be returned to the app's overview page.

1. On the app's overview, find the **OAuth Settings** section
2. Click **Edit** next to OAuth Settings
3. Configure:

Field	Value
Enable OAuth	✓ Checked
Callback URL	https://login.salesforce.com/services/oauth2/success (placeholder — not used by client_credentials flow, but required)
Selected OAuth Scopes	Add: Manage user data via APIs (api) , Perform requests at any time (refresh_token, offline_access)
Enable Client Credentials Flow	✓ Checked
Run As	The integration user that DocHub save-back will impersonate (typically a dedicated dochub-integration@[YOUR_DOMAIN] user — see 6.3)

4. Click **Save**

6.3 Create or select the "Run As" integration user

The client_credentials flow runs as a specific Salesforce user. DocHub's save-back operations (creating new ContentVersions) execute as this user.

Recommended approach: create a dedicated integration user.

1. Setup → Users → Users → **New User**
2. Configure:

Field	Value
First Name	DocHub
Last Name	Integration
Email	dochub-integration@[YOUR_DOMAIN]
Username	dochub-integration@[YOUR_DOMAIN]
User License	Salesforce Integration (recommended) OR Salesforce
Profile	Standard User (or a custom profile with ContentVersion CRUD)
Active	✓ Checked
Generate new password	✓ Checked (a temporary password is sent; you don't need it for client_credentials)

3. Click **Save**

4. Assign [DocHubUser](#) permission set to this integration user (Setup → Users → click the user → Permission Set Assignments → Edit Assignments → add [DocHubUser](#))

5. Return to your External Client App: Setup → External Client App Manager → DocHub Save-Back → OAuth Settings → **Edit**

6. In **Run As**, search for and select [dochub-integration@\[YOUR_DOMAIN\]](#)

7. Click **Save**

💡 The Salesforce Integration license is a separate, lower-cost license type designed exactly for service-to-service flows like this. If your org doesn't have one available, a regular Salesforce license works too.

6.4 Capture the consumer key and secret

1. Setup → External Client App Manager → click **DocHub Save-Back**
2. In the app overview, find the **OAuth Settings** section
3. Click **Manage Consumer Details** (you may be asked to verify your identity)
4. Copy these two values:

Value	Format
Consumer Key	Long base64-like string, 70–100 characters
Consumer Secret	Long base64-like string, 70–100 characters

Save both to your secure password manager. You'll send these to DocHub Support in section 8.

⚠️ Treat both values as secrets. Anyone with these credentials can call your Salesforce org as the integration user.

6.5 Verify the OAuth flow works

You can test the client_credentials flow with **curl** before sending credentials to DocHub Support. This catches misconfiguration before it costs you a support round-trip.

```
curl -X POST https://[YOUR_DOMAIN].my.salesforce.com/services/oauth2/token \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "grant_type=client_credentials" \
-d "client_id=[YOUR_CONSUMER_KEY]" \
-d "client_secret=[YOUR_CONSUMER_SECRET]"
```

Expected response (HTTP 200):

```
{
  "access_token": "00D...!AQEAQ...",
  "signature": "...",
  "scope": "api",
  "instance_url": "https://[YOUR_DOMAIN].my.salesforce.com",
  "token_type": "Bearer",
  "issued_at": "..."
}
```

If you get an error response, see Troubleshooting below.

✅ **Before moving on**, you have:

- ☐ External Client App created
- ☐ Client Credentials Flow enabled

- ☐ Integration user created and **DocHubUser** permission set assigned
- ☐ Integration user selected as "Run As" user
- ☐ Consumer Key recorded
- ☐ Consumer Secret recorded
- ☐ **curl** test returned an **access_token**

Troubleshooting

curl returns **"invalid_grant: this client requires secured authorization"**. The "Enable Client Credentials Flow" checkbox isn't ticked, OR the "Run As" user isn't set. Re-check section 6.2.

curl returns **"invalid_client_id"**. The Consumer Key has whitespace or was truncated. Re-copy from "Manage Consumer Details".

curl returns **"user is inactive"**. The integration user is inactive. Setup → Users → search for the integration user → confirm "Active" is checked.

Connected App creation fails with "You can't create a connected app". Some Partner DE orgs have legacy Connected App creation gated. The External Client App Manager (used in section 6.1) is NOT subject to this restriction. If you're trying the legacy Connected App Manager, switch to External Client App Manager.

"Manage Consumer Details" prompts for verification but rejects all codes. Your admin user needs MFA configured. Setup → Users → your user → Single Sign-On Information → confirm an authenticator is registered.

7. Network configuration

Performed by: Salesforce administrator **Estimated time:** 5 minutes

Salesforce needs explicit allowlisting to make outbound HTTP callouts to the DocHub AWS endpoint. The DocHub managed package ships with a Named Credential that handles this once you configure the endpoint.

7.1 Locate the DocHub Named Credential

1. Setup → Security → Named Credentials
2. You should see **DocHub AWS** (managed by DocHub) — click it
3. You should land on the Named Credential detail page

You'll see two related records:

- **DocHub AWS** (Named Credential)
- **DocHub AWS Credentials** (External Credential)

7.2 Configure the Named Credential URL

The Named Credential ships with a placeholder URL. You'll replace it with your AWS endpoint after DocHub Support provides it in section 8. For now, verify the structure is correct:

1. Click **Edit** on the DocHub AWS Named Credential
2. Confirm these settings:

Field	Expected value
Label	DocHub AWS
Name	DocHub_AWS (do not change)
URL	(placeholder for now — set in section 8.4)
Enabled For Callouts	✓ Checked
External Credential	DocHub AWS Credentials
Generate Authorization Header	✓ Checked
Allow Formulas in HTTP Header	unchecked
Allow Formulas in HTTP Body	unchecked
Client Certificate	(none)

3. Click **Save**

7.3 Verify the External Credential

1. On the same page, click **DocHub AWS Credentials** (the External Credential)
2. Confirm:

Field	Expected value
Authentication Protocol	AWS Signature Version 4
Label	DocHub AWS Credentials
Region	us-west-2
Service	execute-api

The Permission Set Mappings and Principals tabs will be configured in section 8.5 after you receive credentials from DocHub Support.

✓ **Before moving on**, you have:

- ☐ DocHub AWS Named Credential exists and is enabled for callouts
- ☐ External Credential exists with AWS Signature Version 4 protocol
- ☐ Region is **us-west-2** and Service is **execute-api**

You'll complete the URL and credential configuration after section 8.

8. AWS-side provisioning by DocHub

Performed by: Salesforce administrator (with DocHub Support) **Estimated time:** 1–2 business days waiting + 15 minutes configuration

DocHub provisions per-customer AWS infrastructure (API endpoint, IAM credentials, Secrets Manager entries containing your Entra and Salesforce credentials) on your behalf. This is a managed-service step — you don't get direct AWS access.

8.1 Prepare your provisioning request

Gather the following values from previous sections:

Source	Value
Section 1 (prerequisites)	Salesforce Org ID (18-character form)
Section 1 (prerequisites)	Salesforce org type (Production / Sandbox / Developer)
Section 1 (prerequisites)	Salesforce instance URL (e.g. https://[YOUR_DOMAIN].my.salesforce.com)
Section 2.2	Microsoft Entra tenant ID
Section 2.2	Microsoft Entra application (client) ID
Section 2.3	Microsoft Entra client secret
Section 3.2	SharePoint Site ID
Section 3.3	SharePoint Drive ID
Section 5.2	M365Endpoint__c value (Commercial or GCC)
Section 6.4	Salesforce Connected App Consumer Key
Section 6.4	Salesforce Connected App Consumer Secret

8.2 Send the request to DocHub Support

Email your DocHub Support contact using the secure channel they provide (encrypted email, password-protected file share, or secure portal).

Subject: [DocHub AWS Provisioning Request – \[Your Company Name\]](#)

Body template:

Please provision DocHub AWS infrastructure for our customer org.

== Salesforce ==

Company name: [YOUR COMPANY NAME]
Salesforce Org ID: [18-CHAR FORM]
Org type: [Production / Sandbox / Developer]
Instance URL: [https://\[YOUR_DOMAIN\].my.salesforce.com](https://[YOUR_DOMAIN].my.salesforce.com)
Connected App
Consumer Key: [FROM SECTION 6.4]
Consumer Secret: [FROM SECTION 6.4]

== Microsoft 365 ==

M365 endpoint: [Commercial / GCC]
Entra tenant ID: [FROM SECTION 2.2]
Entra app (client) ID: [FROM SECTION 2.2]
Entra client secret: [FROM SECTION 2.3]
SharePoint Site ID: [FROM SECTION 3.2]

```

SharePoint Drive ID:      [FROM SECTION 3.3]

== Contact ==
Primary admin name:       [YOUR NAME]
Primary admin email:      [YOUR EMAIL]
Backup contact:           [BACKUP NAME + EMAIL]

```

⚠ Never send credentials over unencrypted email. Use your DocHub Support secure channel.

8.3 What DocHub Support does

Within 1–2 business days, DocHub Support will:

1. Create your AWS API Gateway endpoint and per-org route
2. Create a per-customer IAM user with narrow `execute-api:Invoke` permission
3. Store your Entra and Salesforce credentials in AWS Secrets Manager (encrypted with a customer-managed KMS key)
4. Run an internal smoke test to verify Entra and Salesforce credentials are valid

DocHub Support will reply with:

Value	Format	Where you'll use it
AWS API endpoint URL	<code>https://[ID].execute-api.us-west-2.amazonaws.com</code>	Section 8.4
AWS Access Key ID	<code>AKIA...</code> (20 characters)	Section 8.5
AWS Secret Access Key	40 characters	Section 8.5

8.4 Set the AWS endpoint URL in Salesforce

When you receive the endpoint URL from DocHub:

1. Setup → Security → Named Credentials → **DocHub AWS** → **Edit**
2. In the **URL** field, paste the endpoint URL from DocHub Support (exactly as provided, no trailing slash)
3. Click **Save**

✅ The Named Credential's URL should now be your provisioned endpoint.

8.5 Configure the External Credential principal

This is where the AWS credentials live in Salesforce. They never appear in any source code, customer metadata, or DocHub package — they're stored only in the External Credential's principal.

1. Setup → Security → Named Credentials → **External Credentials** tab → **DocHub AWS Credentials**
2. Scroll to **Principals** → click **New**
3. Fill in:

Field	Value
Parameter Name	DocHubAwsPrincipal
Identity Type	Named Principal
Sequence Number	1

4. Click **Save** — you're returned to the principal list
5. Click your new principal (DocHubAwsPrincipal) — click **Edit Authentication Parameters**
6. Add two parameters:

Parameter Name	Value
AwsAccessKeyId	The Access Key ID from DocHub Support
AwsSecretAccessKey	The Secret Access Key from DocHub Support

7. Click **Save** for each parameter

8.6 Grant the principal to your permission sets

The principal exists but isn't yet usable by Apex code. You must grant it to the permission sets that need to make the callout.

1. Setup → Users → Permission Sets → **DocHubUser** → **External Credential Principal Access** → **Edit**
2. Add the **DocHub AWS Credentials — DocHubAwsPrincipal** entry → **Save**
3. Repeat for the **DocHubAdmin** permission set

✓ **Before moving on**, you have:

- ☐ Named Credential URL set to your provisioned endpoint
- ☐ External Credential principal created with both AWS credentials
- ☐ Principal granted to both DocHubUser and DocHubAdmin permission sets

Troubleshooting

DocHub Support email gets bounced. Confirm the email address through your DocHub sales contact. Don't use a public-facing support address for credential exchange.

Internal smoke test fails — DocHub reports your Entra credentials don't work. The most common cause: the Entra app's **Sites.Selected** grant on the specific site (section 3.4) wasn't completed. Re-run section 3.4 and confirm the GET returns your DocHub app in the permissions list.

Internal smoke test fails — DocHub reports your Salesforce credentials don't work. The most common cause: the integration user is inactive, or the Connected App's "Enable Client Credentials Flow" toggle is off. Re-run section 6.5 **curl** test to verify the flow works from your side.

External Credential principal "Save" button is greyed out. You don't have permission to manage External Credentials. Required: System Administrator profile OR custom profile with "Customize Application" enabled.

9. End-to-end verification

Performed by: Salesforce administrator **Estimated time:** 15 minutes

This is the moment of truth — installing the DocHub LWC on a Lightning page and editing a real file end-to-end.

9.1 Add the Files Panel to a Lightning record page

Pick a standard object you have records for (Account is a good choice for testing).

1. Open any Account record in your org
2. Click the gear icon → **Edit Page**
3. In the Lightning App Builder, find the **DocHub Files Panel** component in the left sidebar (under "Custom" or "Custom — Managed")
4. Drag it onto the page in your preferred location (the right sidebar usually works well)
5. Click **Save** → confirm activation choice (e.g. "Activate for this object → Org Default")

✅ The page reloads with DocHub Files Panel visible on Accounts.

9.2 Upload a test docx

1. Return to the Account record
2. In the DocHub Files Panel, click **Upload File** (or drag a **.docx** from your desktop)
3. Confirm the file appears in the panel with a docx icon and "Edit" action available

💡 If the panel shows "No files yet" but you've uploaded, refresh the page.

9.3 Trigger an edit

1. Click **Edit** on the test file
2. ✅ Within a few seconds, a new browser tab opens to Microsoft Word for the Web showing your file

If the tab opens to a Microsoft sign-in page instead, your test user (Account B from section 1) needs to sign in to M365 first. Complete sign-in.

9.4 Edit, save, close

1. In the Word for the Web tab, make a small change (type a few words)
2. Press **Ctrl+S** (or Cmd+S on Mac) to force a save — don't rely on autosave
3. Close the Word for the Web tab

9.5 Verify the file refreshes in Salesforce

Return to the Salesforce tab.

1. ✅ Within ~1 second, you should see a spinner icon and "Saving your changes..." text next to the file you just edited
2. ✅ The file's action dropdown menu (Edit, Download, etc.) is disabled while the spinner is showing
3. ✅ Within 60–90 seconds, the spinner clears and the file row returns to normal
4. ✅ A green "File updated" toast appears at the top of the screen
5. Click **Download** on the file — verify your edit is present in the downloaded copy

💡 The 60–90 second wait is normal. It's primarily Microsoft's save-propagation delay from Word for the Web to SharePoint (typically 30–60 seconds), plus DocHub's polling cycle (up to 60 seconds), plus the save-back operation (~5 seconds). The spinner UX makes the wait pleasant; the actual processing is bounded.

9.6 Verify ContentVersion history

1. Click the file name in the Files Panel → **View Details**
2.  You should see at least two ContentVersion entries: the original (V1) and your edited version (V2)
3.  The latest version shows your edit

You can repeat this test as many times as you want — each save creates a new ContentVersion.

Troubleshooting

"Edit" action doesn't appear on the file. Edit is only available for **.docx**, **.xlsx**, **.pptx** files. Confirm the file extension. Image files, PDFs, and other formats show only the standard actions (View Details, Download, Delete).

Edit click does nothing. Open browser dev tools → Console. Look for errors. The most common cause: AWS endpoint URL not set in Named Credential (section 8.4). Re-check the URL.

"You don't have permission to perform this action" toast. The current user lacks the **DocHubUser** permission set. Assign it (section 4.3).

Word for the Web tab opens to a generic Microsoft error page. Check the URL in the new tab. If it contains **aka.ms/officehome** or similar, M365 doesn't recognize the user. The user needs an Office for the Web license. Verify in M365 admin center.

Word for the Web opens but says "We're sorry, but this document cannot be opened in your web browser". This usually means SharePoint can't find the file. Confirm with DocHub Support that the Entra app's Sites.Selected grant succeeded on your specific site.

Edit works but file never refreshes after closing the tab.

- Wait the full 90 seconds before assuming failure (Microsoft's save delay is the largest variable)
- If still no refresh, check Salesforce debug logs for **DocHub:** entries; copy any error and contact DocHub Support
- The browser tab should show a small "saving" indicator before you close — if it doesn't, your edit may not have been committed by Office

Spinner appears but never clears (stays > 5 minutes). The spinner times out automatically after 30 minutes. If you see it stuck past 5 minutes consistently, contact DocHub Support — there's likely a save-back failure on the AWS side that needs investigation.

10. Troubleshooting

Common issues organized by symptom. If your issue isn't here, see section 11.4 for support contact.

10.1 "Edit" button doesn't appear

Likely cause	Resolution
User doesn't have DocHubUser permission set	Assign in Setup → Users → Permission Sets
File is not a supported format (.docx/.xlsx/.pptx)	Only Office formats support online editing
DocHub Files Panel LWC isn't on the page	Re-do section 9.1 to add it

Likely cause	Resolution
The component is on the page but only as a placeholder (no record context)	Confirm the page has a record context — the panel is record-specific

10.2 Edit click produces "Unable to start edit session" toast

Likely cause	Resolution
AWS endpoint URL missing or wrong in Named Credential	Set in Setup → Named Credentials → DocHub AWS → URL (section 8.4)
AWS credentials missing from External Credential principal	Re-do section 8.5
External Credential principal not granted to the user's permission set	Re-do section 8.6
Custom Metadata Config record incomplete	Verify all 9 fields in section 5.2
DocHub Support hasn't completed AWS provisioning	Check email for confirmation; contact support

10.3 Edit opens Word for the Web with "401 Unauthorized"

Likely cause	Resolution
Entra app's Sites.Selected grant on your site failed	Re-do section 3.4 and verify with 3.5
Entra client secret expired	Generate a new secret (section 2.3) and send to DocHub Support
Service account M365 license inactive	Verify in M365 admin center → Users

10.4 File doesn't refresh after edit (waited > 90 seconds)

Likely cause	Resolution
User closed Word tab before Office committed the save	Re-edit, save with Ctrl+S, wait 5+ seconds, then close
DocHub Salesforce Connected App credentials wrong on AWS side	Test with curl (section 6.5); if test passes, contact DocHub Support
Salesforce integration user (Run As) is inactive	Reactivate in Setup → Users
File deleted between Edit and save	Check ContentDocument exists in Salesforce

10.5 "Document is currently being edited in another session" toast

This usually means a previous edit attempt failed mid-flow and left a lock that hasn't expired yet.

Resolution

Wait 2 minutes and retry — pending locks auto-expire in 120 seconds

If problem persists, contact DocHub Support with the document ID

10.6 Files Panel doesn't show files that exist on the record

Likely cause	Resolution
User can't see ContentDocuments on this record	Check the record's Sharing → Files settings
Files exist but weren't shared via the record	Files attached directly (vs via Files related list) sometimes don't appear; reattach
User profile lacks ContentVersion read access	Setup → Users → Profile → check Standard Object Permissions

10.7 Enabling debug logs for diagnosis

If you need to capture detailed logs for DocHub Support:

1. Setup → Environments → Logs → **Debug Logs**
2. Click **New** under "User Trace Flags"
3. Select your user (or the affected user)
4. Debug Level: create a new one with these levels:

Category	Level
Apex Code	FINEST
Apex Profiling	INFO
Callout	FINEST
Database	INFO
System	INFO
Validation	INFO
Visualforce	INFO
Workflow	INFO

5. Expiration Date: ~1 hour from now
6. Click **Save**
7. Reproduce the issue
8. Setup → Debug Logs → click the most recent log entry → Download

⚠ Debug logs may contain sensitive data including user IDs and document IDs. Share only with DocHub Support via your secure channel.

11. Operations and maintenance

11.1 Credential rotation schedule

Credential	Rotation cadence	How
Entra client secret	Every 24 months (set as expiry)	Section 2.3 — generate new secret, send to DocHub Support, old secret can be deleted after new is confirmed live
Salesforce Connected App consumer secret	Annually (recommended)	Setup → External Client App Manager → DocHub Save-Back → Manage Consumer Details → Reset; send new secret to DocHub Support
AWS credentials (provided by DocHub)	DocHub rotates server-side	No action required from you

Set calendar reminders for both expiry dates.

11.2 Monitoring file edit health

DocHub doesn't currently push proactive alerts to customers. Monitor these signals from the Salesforce side:

Signal	Where	What it tells you
Failed Edit attempts in user feedback	User support tickets	Authentication or config issue
Apex callout failures	Setup → Apex Jobs → check for failed <code>DocumentUpdated__e</code> events	Save-back path broken
Old ContentVersions without subsequent edits	Reports on ContentVersion sorted by LastModifiedDate	Service may be degraded — confirm with a manual test

If you suspect widespread failure, run the section 9 end-to-end test as a smoke check.

11.3 Upgrading the DocHub package

When DocHub releases a new package version:

1. DocHub will email you the new install URL
2. Install in a sandbox first; verify the section 9 end-to-end test still passes
3. Schedule production upgrade during low-use hours
4. Install in production — Custom Metadata, Connected App, and External Credential principals are preserved across upgrades
5. Re-run section 9 in production to confirm

⚠ Don't install a new version while users are mid-edit. Document locks may briefly be lost. Notify users in advance and confirm Files Panel shows no active edit sessions before upgrading.

11.4 Contacting DocHub Support

For DocHub-specific issues (not Salesforce or M365 issues), contact DocHub Support via the channel provided during onboarding.

Include in every support request:

- Your Salesforce Org ID (18-character form)
- Approximate time the issue occurred (with time zone)
- ContentDocument ID of the affected file (if applicable)
- Steps to reproduce
- Browser/OS combination of the affected user
- Salesforce debug log (section 10.7) for the failed action

DocHub Support response targets:

Severity	First response
P1 — production down for all users	4 business hours
P2 — production degraded or affecting subset of users	1 business day
P3 — non-production or single-user issue	2 business days
P4 — question or feature request	5 business days

12. Appendix

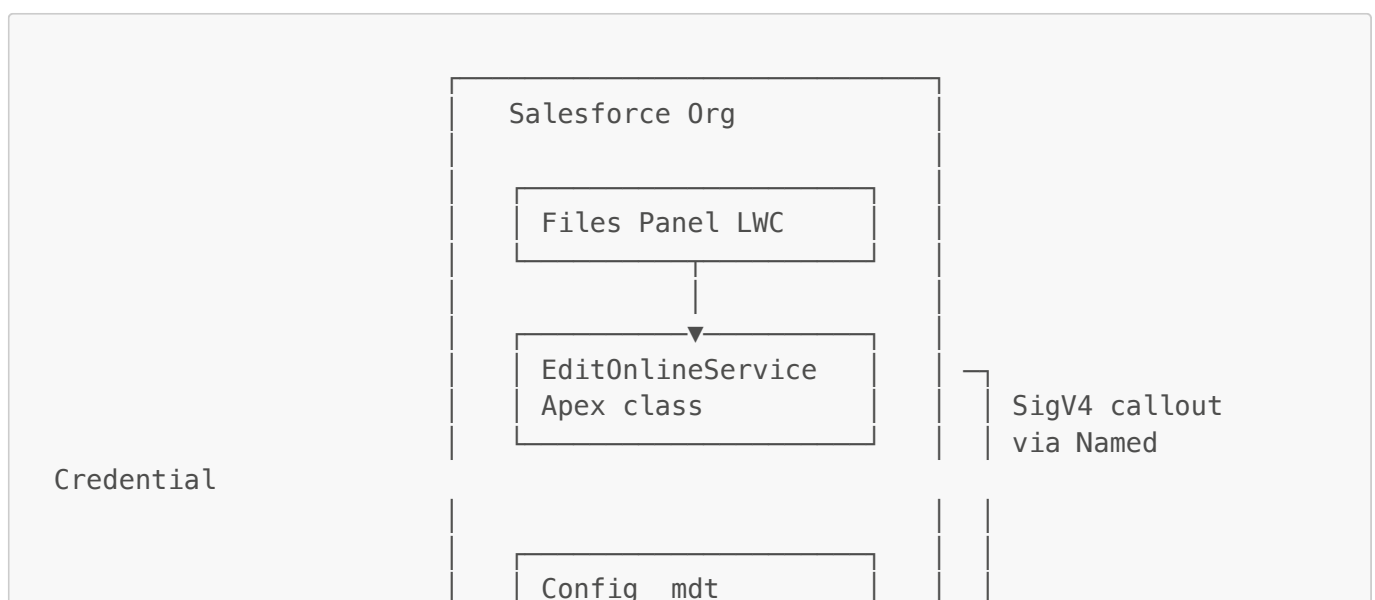
12.1 Glossary

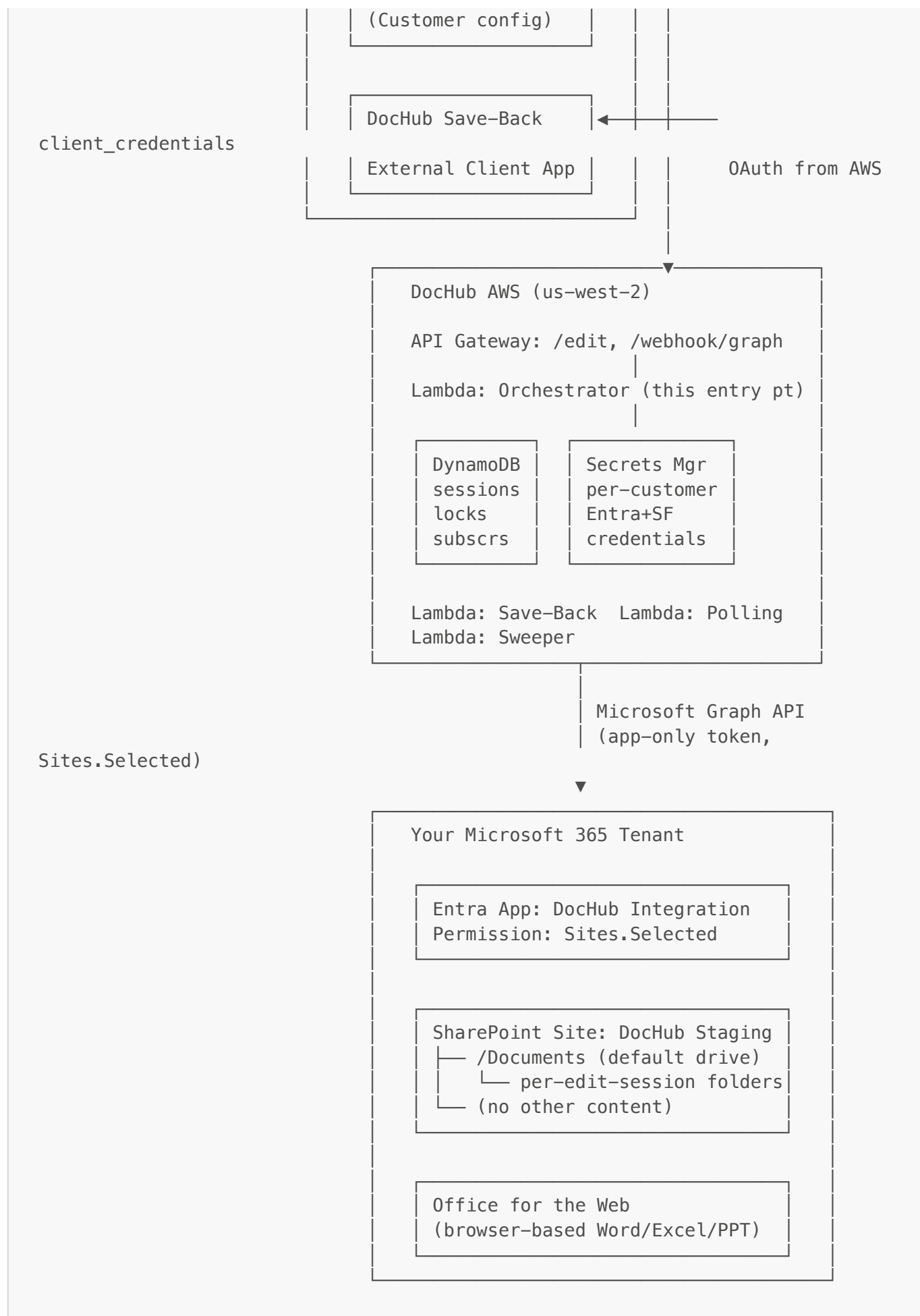
Term	Meaning
Sites.Selected	A Microsoft Graph API permission that grants an app access to specific SharePoint sites (granted per-site), not all sites in the tenant
ContentDocument	Salesforce object representing a file (parent of versions)
ContentVersion	Salesforce object representing a specific version of a file (one file = many versions over time)
External Credential	Salesforce metadata for storing authentication credentials separately from where they're used
Named Credential	Salesforce metadata for an outbound HTTP endpoint, linked to an External Credential for auth
LWC	Lightning Web Component — modern Salesforce UI component framework
Custom Metadata Type	Salesforce-managed customizable configuration record type
GCC	Government Community Cloud — Microsoft's FedRAMP Moderate cloud for US government customers
SigV4	AWS Signature Version 4 — the authentication scheme AWS API Gateway uses
Service Account	A dedicated user account used by automated systems rather than people
UPN	User Principal Name — the unique username for an M365 user (typically name@domain.com)

12.2 Config Custom Metadata Type — full field reference

Field API name	Type	Required	Description	Valid values
EntraTenantId__c	Text(50)	Yes	Microsoft Entra tenant ID	GUID format
EntraClientId__c	Text(50)	Yes	DocHub Integration app's Application (client) ID	GUID format
SharePointSiteId__c	Text(255)	Yes	SharePoint Site ID for staging	[hostname], [guid], [guid]
SharePointDriveId__c	Text(255)	Yes	SharePoint Drive ID (default doc library)	Starts with b!
M365Endpoint__c	Picklist	Yes	Microsoft cloud variant	Commercial, GCC
ServiceAccountUserPrincipalName__c	Email	Yes	Service account UPN	Email format
LockTtlMinutes__c	Number	Yes	Document edit lock TTL	5–120; default 30
SessionTtlHours__c	Number	Yes	Stale session cleanup TTL	1–48; default 24
IdentityMode__c	Picklist	Yes	Authentication mode	ServiceAccount (only value in v1.0)

12.3 Architecture diagram (detailed)





12.4 Data flow — what crosses each boundary

From	To	Data	Auth
Salesforce LWC	Salesforce Apex	<code>contentDocumentId</code> , <code>contentVersionId</code>	Standard SF session
Salesforce Apex	DocHub AWS API Gateway	Edit request payload (org ID, content version ID, user ID, config snapshot)	AWS SigV4
DocHub AWS	Salesforce REST API	OAuth token request (no payload), then file download/upload	Salesforce client_credentials OAuth
DocHub AWS	Microsoft Graph	File upload, file download, subscription create	Entra app-only OAuth (Sites.Selected)
Microsoft Graph	DocHub AWS	Change notifications (when supported)	HMAC validated on AWS side
Office for the Web	SharePoint	User's edits	M365 user session (separate from DocHub auth)

12.5 What data DocHub stores

Location	Data	Retention
AWS DynamoDB (sessions table)	Session metadata: session ID, customer ID, user ID, doc ID, drive item ID, timestamps	Auto-deleted via TTL 24 hours after creation
AWS DynamoDB (locks table)	Document-level locks: document ID, holder session ID, expiration	Auto-deleted via TTL (typically 30 min)
AWS DynamoDB (drive subscriptions)	Microsoft Graph webhook subscription details	Auto-deleted when no active sessions reference the drive
AWS Secrets Manager	Per-customer Entra credentials, Salesforce credentials, cached access tokens	Persistent — rotated per section 11.1 schedule
AWS S3 (audit log)	Per-session audit events: edit started/completed/failed	1 year retention; encrypted at rest
Your SharePoint site	The file content being edited	Deleted by DocHub immediately after save-back completes
Your Salesforce org	All ContentVersions including DocHub-created versions	Per your Salesforce data retention policy

DocHub does NOT store:

- File content (it transits through AWS Lambda memory only, never persisted)
- User passwords or personally identifying information beyond the user ID
- Search indexes of your content
- Salesforce record data beyond the file being edited

End of runbook.

For questions, contact DocHub Support via the secure channel provided during onboarding.

For feedback or runbook improvement suggestions, email your DocHub account manager.