

Single Sign-On and Provisioning

Let your users sign in to Resourcecloud with your identity provider: through OpenID Connect (Authorization Code Flow with PKCE) for Microsoft Entra ID, Keycloak, Okta, Google Workspace and others — or through SAML 2.0 for ADFS, Keycloak in SAML mode and older corporate IdPs. And have the identity provider create, change and deactivate the accounts — and place people in teams — automatically, through SCIM.

Overview

Single sign-on is configured by an **RC Admin** in *RC Settings*. Once it is switched on, the login page shows a sign-on button below the password form. Password login stays available next to it, so a wrong setting can never lock you out.

	OpenID Connect	SAML 2.0
Use it when your IdP is	Entra ID, Keycloak, Okta, Google Workspace, Authentik, Zitadel ...	ADFS , Shibboleth, Keycloak, any corporate IdP that only speaks SAML
What the IdP issues	a signed ID token	a signed assertion
Set up with	issuer, client ID, client secret	IdP metadata (URL or XML) — and our metadata for the IdP
Chapters	Quick start, Entra ID	How it works, Quick start, Keycloak, ADFS

Separately from signing in, **automatic provisioning (SCIM)** lets the identity provider manage the accounts themselves: create, change, *deactivate* (never delete), and assign to teams through groups.

Both can be configured and switched on side by side; the login page then shows one button each. Which account a person is, and signing it in, works the same way for both — see How accounts are matched.

What happens when a user clicks the button of an **OpenID Connect** provider (the SAML flow is described further down):

- 1 **Resourcecloud sends the browser to your provider.** The request carries a one-time `state`, a `nonce` and a PKCE challenge (S256) that only this browser session can answer.
- 2 **The user signs in at the provider** — with whatever it requires: password, MFA, conditional access.
- 3 **The provider sends the browser back** to `/oidc_callback.php` with a short-lived authorization code.
- 4 **Resourcecloud exchanges the code** for an ID token, directly with the provider (server to server, with the PKCE verifier and the client secret), and checks the token's signature and claims.
- 5 **Resourcecloud finds the user's account** — or, if you allowed it, creates one — and signs the user in. If the account uses a passkey as second factor, the passkey step still follows.

Everything is configured **per instance**. On a multi-tenant (SaaS) installation each tenant has its own database and therefore its own provider, client or certificates, and claim or attribute mapping.

Before you start

You need	Notes
Database version 151 (OpenID Connect), 153 (SAML) or 154 (SCIM) or later	Migration 151 adds the OpenID Connect settings and the account link, 153 the SAML settings, 154 the provisioning tables. They run by themselves on the first request after an upgrade.
The RC Admin role	Only RC Admins can see and change the settings.

Permission to register an application at your provider	For Entra ID: at least the <i>Application Developer</i> role, or an administrator who does it for you. For ADFS: you add a <i>relying party trust</i> — or an ADFS administrator does.
An https address for Resourcecloud	Entra ID, Okta and Google only accept https redirect URIs (Entra and Google also allow <code>http://localhost</code> for testing). The address is the <code>baseUrl</code> of the installation.
Outbound https from the server to the provider	The server — not the browser — fetches the provider's configuration, its signing keys and the token. If your server reaches the internet through a proxy, set it under <i>RC Settings</i> → <i>Configuration</i> → <i>Proxy Configuration</i> ; it is used for these calls too. Certificates are always verified.
For SAML: an IdP that can sign with SHA-256	Answers signed with SHA-1 are refused. Every ADFS since 3.0 and every current Keycloak signs with SHA-256 by default.
At least one RC Admin who can sign in with a password	Your way back in if the single sign-on setup turns out to be wrong. See Operating it.

OpenID Connect · Quick start

- 1 Open *RC Settings* → *External Services* and find the panel **Single Sign-On (OpenID Connect)**. Copy the **redirect URI** it shows, for example `https://rc.example.com/oidc_callback.php`.
- 2 At your provider, register Resourcecloud as an application (a *web / confidential* client) with exactly that redirect URI. Note the **client ID**, create a **client secret**, and note the provider's **issuer URL**. Entra ID is walked through below.
- 3 Enter issuer, client ID and client secret in the panel. Check the claim mapping and choose how existing accounts are found.
- 4 Switch **Enable Single Sign-On** on and press **Submit**. Resourcecloud asks the provider for its configuration first and refuses to save if the issuer does not answer — the message names the problem.

- 5 Open the login page in a private window. The sign-on button is there. Sign in with a test user.
- 6 Check the result: you should land on your usual start page. The first sign-on of an account links it to the identity at the provider (see How accounts are matched).

Test with a second browser first. Stay signed in as RC Admin in your normal window while you try the button in a private one. Then a surprise costs nothing.

OpenID Connect · Microsoft Entra ID

These steps use the Microsoft Entra admin center (entra.microsoft.com). Menu names can shift slightly between releases.

1 · Register the application

- 1 Go to *Entra ID* → *App registrations* → *New registration*.
- 2 **Name:** for example `Resourcecloud`.
Supported account types: *Accounts in this organizational directory only (single tenant)*.
- 3 **Redirect URI:** platform **Web**, and the URI from the Resourcecloud panel. Press *Register*.
- 4 On the *Overview* page, copy the **Application (client) ID** and the **Directory (tenant) ID**.
- 5 Open *Certificates & secrets* → *Client secrets* → *New client secret*. Choose an expiry and press *Add*. Copy the **Value** right away — not the *Secret ID*. Entra shows the value only once.

Client secrets expire — after 24 months at the most. When a secret expires, the sign-on button keeps showing but every attempt fails with “*The identity provider could not confirm your sign-on*”. Put the expiry date in your calendar. To rotate: create a second secret in Entra, paste it into the panel, then delete the old one.

The issuer must belong to your tenant. Use the tenant ID, not common , organizations or consumers : Resourcecloud refuses those, because with them anybody with a Microsoft account of any tenant could reach the login.

2 · Claims in the ID token

An Entra ID token contains `sub` , `name` and `preferred_username` (the user principal name, such as `anna.schmidt@example.com`) by default. First and last name and the e-mail address are *optional claims*.

- 1 Open *Token configuration* → *Add optional claim*, choose token type **ID**.
- 2 Tick `email` , `given_name` and `family_name` and press *Add*. When Entra offers to add the Microsoft Graph *email* permission, accept.

Resourcecloud also asks the provider's userinfo endpoint for a claim that is missing from the token, so the match usually works even without this step — but putting the claims into the token is the dependable way.

Not every Entra user has an `email` claim (it comes from the mailbox attribute), and the user principal name is not always the mail address. If your Resourcecloud accounts carry the sign-in name as e-mail, set the **E-Mail claim** in the Resourcecloud panel to `preferred_username` . The mapping accepts any claim name.

3 · Decide who may sign in

By default every user of your tenant can sign in to a new application. To limit it, open *Entra ID* → *Enterprise applications* → *Resourcecloud* → *Properties*, set **Assignment required?** to Yes, and assign the users or groups under *Users and groups*. Everybody else is stopped by Entra before Resourcecloud sees them.

If your tenant does not allow users to consent to applications, an administrator grants consent once under *API permissions* → *Grant admin consent*. The permissions needed are the delegated Microsoft Graph permissions `openid` , `profile` and `email` .

4 · Settings in Resourcecloud

Field	Value for Entra ID
Issuer	<code>https://login.microsoftonline.com/<tenant-id>/v2.0</code> — with the Directory (tenant) ID from step 1
Client ID	the Application (client) ID
Client Secret	the secret <i>Value</i>
Scopes	<code>openid profile email</code> (the default)
Claim mapping	Login name: <code>preferred_username</code> · E-mail: <code>email</code> · First name: <code>given_name</code> · Last name: <code>family_name</code> (all defaults)
Find the existing account by	<i>E-Mail</i> if your accounts carry the mail address; <i>Login Name</i> if <code>ad_name</code> holds the sign-in name; <i>Login Name</i> , then <i>E-Mail</i> if you are unsure

Switch the feature on, press **Submit**, and continue with step 5 of the Quick start.

OpenID Connect · Other providers

Any provider that publishes an OpenID Connect discovery document at `<issuer>/.well-known/openid-configuration` and signs ID tokens with RS256/384/512, PS256/384/512 or ES256/384/512 can be used. Register a *confidential web client* with the redirect URI from the panel and the *authorization code* flow. If the provider lets you choose, require PKCE with S256.

Provider	Issuer	Notes
Keycloak	<code>https://<host>/realms/<realm></code>	Client authentication <i>on</i> (confidential), standard flow <i>on</i> . Older Keycloak versions use <code>/auth/realms/<realm></code> .

Okta	<code>https://<org>.okta.com</code> or <code>https://<org>.okta.com/oauth2/default</code>	Application type <i>Web</i> . Use the issuer of the authorization server your app belongs to.
Google Workspace	<code>https://accounts.google.com</code>	Every Google account can reach the sign-in — restrict it to your domain in the OAuth consent screen (<i>Internal</i>).
Authentik, Zitadel, and others	shown on the provider's application page	The issuer must match the one in the discovery document exactly — see Troubleshooting.

The flow has been exercised end to end against a test provider, and Entra ID's published configuration and signing keys have been checked. The other providers follow the same standard but have not been tried one by one.

OpenID Connect · Settings reference

All fields are in *RC Settings* → *External Services* → *Single Sign-On (OpenID Connect)*.

Field	Meaning
Enable Single Sign-On	Shows the button on the login page. Off by default. Cannot be switched on without issuer and client ID, and only when the provider answers.
Issuer	The provider's issuer URL, https only. A trailing slash is ignored. Must equal the <code>issuer</code> in the provider's discovery document.
Client ID	As registered at the provider.
Client Secret	Write-only: the field is never filled in again. A note shows whether one is stored. Leave the field empty to keep it; tick <i>Remove the stored secret</i> to delete it. Without a secret Resourcecloud acts as a <i>public client</i> and relies on PKCE alone — only do that if your provider is set up that way.
Scopes	Space-separated. <code>openid</code> is always requested. Default <code>openid profile email</code> .

Button Text on the Login Page	For example <i>Sign in with Microsoft</i> . Empty means <i>Single Sign-On</i> .
Claim mapping — Login Name	Claim compared with the account's <code>ad_name</code> . Default <code>preferred_username</code> .
Claim mapping — E-Mail	Claim compared with the account's mail address. Default <code>email</code> .
Claim mapping — First / Last Name	Used only when an account is <i>created</i> . Defaults <code>given_name</code> and <code>family_name</code> ; if both are missing, the provider's <code>name</code> claim is split at its last blank.
Find the existing account by	<i>E-Mail, Login Name (ad_name), or Login Name, then E-Mail</i> . Applies to the first sign-on of an account only.
Create the account at the first sign-on	Off: people without an account are turned away. On: an account is created — see below.

OpenID Connect · What is verified

The sign-on only works if all of these hold; otherwise the user is returned to the login page with a short reason.

- **The answer belongs to this login.** The callback must present the `state` this very browser session was given, within 10 minutes, and only once — a replayed or foreign callback is refused.
- **PKCE (S256)** binds the code to the browser session that started the login; a stolen code is useless without the verifier.
- **The ID token is signed by the provider.** The signature is checked against the keys published at the provider's `jwks_uri` . Only RS*, PS* and ES* algorithms are accepted; `none` and the HS* algorithms are rejected. A rotated signing key is picked up automatically.
- **The claims fit.** Issuer equals the configured issuer, audience contains this client ID, `azp` (when present) is this client, the token has not expired, and its `nonce` is the one of this login.

- **The discovery document is about the issuer asked for.** All provider URLs must be https. (For a test provider on plain http, the server environment variable `rc_sso_allow_http=1` (the older name `rc_oidc_allow_http=1` still works) permits it; never set it in production.)
- **Userinfo is only trusted for the same person** — its `sub` must equal the token's.

Entra ID does not send `email_verified`, and Resourcecloud treats an absent value as verified. Because the issuer is bound to *your* tenant, only your own directory users can get as far as matching by e-mail; if users in your directory can edit their own mail attribute, use *Assignment required* and keep that attribute under administrative control — or match by login name.

SAML 2.0 · How it works

For identity providers that do not speak OpenID Connect — **ADFS**, Keycloak in SAML mode, Shibboleth and older corporate IdPs. Resourcecloud acts as the *service provider* (SP) and takes the *signed assertion* your identity provider (IdP) issues.

- 1 **Resourcecloud sends the browser to the IdP** with an *AuthnRequest* (HTTP-Redirect binding). It remembers that request, so only an answer to it will be accepted.
- 2 **The user signs in at the IdP** — password, MFA, whatever the IdP requires.
- 3 **The IdP posts its signed answer** to the assertion consumer service, `/saml_acs.php` (HTTP-POST binding).
- 4 **Resourcecloud checks the answer**: signature, issuer, audience, destination, recipient, the request it answers, and its validity window — see What is verified.
- 5 **The browser is sent on to** `/saml_complete`, which hands the result only to the browser that started the login, finds or creates the account and signs it in. A passkey as second factor is still asked for.

Start from Resourcecloud. Only *service-provider-initiated* sign-on is supported: an answer nobody asked for — the IdP's own "IdP-initiated" portal button — is refused on purpose, because that is the way answers are replayed or planted. To put a Resourcecloud tile into an IdP portal, link it to `<baseUrl>/saml_login` : that starts the sign-on from our side.

SAML 2.0 · Quick start

- 1 Open *RC Settings* → *External Services* and find the panel **Single Sign-On (SAML)**. It shows three values for the IdP: the **Entity ID**, the **Assertion Consumer Service URL** and the **Metadata URL** (`https://rc.example.com/saml_metadata`).
- 2 At the IdP, register Resourcecloud as a relying party / client. The easy way is to let the IdP **import the metadata URL**; otherwise enter Entity ID and ACS URL by hand. Make sure the IdP
 - signs the **assertion or the response** (either is fine; the assertion is the stronger choice) with **SHA-256 or better** — SHA-1 is refused;
 - sends a **persistent** NameID, and the attributes you want to map (e-mail, names);
 - does *not* require signed requests or encrypt the assertion — Resourcecloud does neither (see Limitations).
- 3 Back in the panel, paste the **IdP's metadata URL** — or its metadata XML — into *Import IdP Metadata* and press **Submit**. Entity ID, single sign-on URL and signing certificate(s) are filled in from it. (You can also type them in.)
- 4 Check the attribute mapping: the names must equal what the IdP sends. Typical names are listed below.
- 5 Choose how existing accounts are found and whether accounts are created, switch **Enable SAML Single Sign-On** on and press **Submit**.
- 6 Open the login page in a private window and sign in with a test user — see the warning in the OpenID Connect quick start: keep your RC Admin session open in the other window.

The IdP has to reach the metadata URL if it imports it (an ADFS server needs outbound access to your Resourcecloud address). If it cannot, open the metadata URL in your browser, save the XML and import that file at the IdP.

SAML 2.0 · Keycloak

This is the setup the automated interoperability test builds against a real Keycloak (26.x) — it is the tested path. Menu names are those of the Keycloak admin console.

- 1 Select your realm, then *Clients* → *Import client*. Upload the Resourcecloud **SP metadata** (save `https://rc.example.com/saml_metadata` as a file). Keycloak fills in Client ID (= the Entity ID), the ACS URL and the redirect URI. Press *Save*.
- 2 *Settings* → *Signature and Encryption*: **Sign assertions** on (recommended) — **Sign documents** may stay on as well. Signature algorithm `RSA_SHA256`.
- 3 *Keys*: set **Client signature required** to **Off**. **This one matters**: Keycloak's default expects signed requests, Resourcecloud sends unsigned ones, and Keycloak then answers with an error page instead of a login form.
- 4 *Settings* → *Name ID format*: `username` or `persistent`. Leave *Force name ID format* off.
- 5 *Client scopes* → *<client>-dedicated* → *Add mapper* → *By configuration* → *User Property*, once for each of:

Name	User Property	SAML Attribute Name	Name Format
<code>email</code>	<code>email</code>	<code>email</code>	Basic
<code>firstName</code>	<code>firstName</code>	<code>firstName</code>	Basic
<code>lastName</code>	<code>lastName</code>	<code>lastName</code>	Basic
<code>username</code>	<code>username</code>	<code>username</code>	Basic

- 6 In Resourcecloud, import the IdP metadata from `https://<keycloak>/realms/<realm>/protocol/saml/descriptor`. The attribute names in the panel (`username` , `email` , `firstName` , `lastName`) already match the mappers above.

Keycloak users need e-mail, first and last name filled in, or Keycloak interrupts the sign-in with an *Update account information* step.

SAML 2.0 · ADFS

Not yet verified against an ADFS server. The integration is tested against Keycloak (see above) and against hand-made signed answers; this chapter follows ADFS's standard configuration. If your ADFS version differs somewhere, the log says exactly what was rejected — see Troubleshooting.

- 1 *AD FS Management* → *Relying Party Trusts* → *Add Relying Party Trust* → *Claims aware* → *Start*. Choose *Import data about the relying party published online or on a local network* and enter the Resourcecloud metadata URL — or *Import from a file* with the saved XML. (Entering the data by hand works too: enable *SAML 2.0 WebSSO*, ACS URL as the URL, the Entity ID as the identifier, and skip the certificate.)
- 2 Pick an access control policy that fits (for example *Permit everyone*) and finish. Leave *Configure claims issuance policy* ticked.
- 3 *Edit Claim Issuance Policy* → *Add Rule* → *Send LDAP Attributes as Claims*. Attribute store *Active Directory*, mapping:

LDAP attribute	Outgoing claim type
E-Mail-Addresses	E-Mail Address
Given-Name	Given Name
Surname	Surname
User-Principal-Name	UPN

- 4 *Add Rule → Transform an Incoming Claim*: incoming claim type **UPN** (or E-Mail Address), outgoing claim type **Name ID**, outgoing name ID format **Persistent Identifier**, pass through all claim values. *Not Transient* — a transient NameID is different at every sign-on and is refused.
- 5 Open the trust's *Properties. Advanced*: secure hash algorithm **SHA-256**. *Encryption*: no certificate (Resourcecloud cannot decrypt). *Signature*: nothing needed. ADFS signs the assertion by default.
- 6 In Resourcecloud, import `https://<adfs>/FederationMetadata/2007-06/FederationMetadata.xml` into *Import IdP Metadata*. This fills in the entity ID (`http://<adfs>/adfs/services/trust` — note: `http`), the single sign-on URL (`https://<adfs>/adfs/ls/`) and the token-signing certificate.
- 7 Set the attribute names to ADFS's claim types (see the table below).

ADFS rolls its token-signing certificate by itself, normally once a year. From that day the certificate stored in Resourcecloud no longer fits and every sign-on fails with “*Signature validation failed*”. While the old and the new certificate overlap, the federation metadata lists both: re-import it before the old one expires. The settings panel shows each certificate's expiry date, in red once it has passed — see Certificates and rollover.

SAML 2.0 · Settings reference

All fields are in *RC Settings → External Services → Single Sign-On (SAML)*. The panel's own lines, **Entity ID**, **ACS URL** and **Metadata URL**, are what you give the IdP.

Field	Meaning
Enable SAML Single Sign-On	Shows the button on the login page. Off by default. Cannot be switched on without IdP Entity ID, single sign-on URL and a certificate; saving builds the settings exactly as a login would and refuses anything that login would refuse.
Import IdP Metadata	The IdP's metadata URL (https; fetched by the server, through the global proxy if one is set) or its metadata XML pasted in. On Submit it fills in the three IdP fields below; empty leaves them as they are. The URL it came from is shown underneath. The IdP must offer its single sign-on with the HTTP-Redirect binding.

IdP Entity ID	Must equal the <code>Issuer</code> in the IdP's answers.
IdP Single Sign-On URL	https. Where the browser is sent with the AuthnRequest.
IdP Signing Certificate(s)	PEM. Several may be listed — that is how a certificate rollover is bridged. Below the box: subject, expiry date and SHA-256 fingerprint of each; compare the fingerprint with the IdP.
Button Text on the Login Page	Empty means <i>Single Sign-On (SAML)</i> .
Own Entity ID (optional)	Empty: <code><baseUrl>/saml_metadata.php</code> . Only set it if the IdP insists on another identifier — and register it there again.
Attribute mapping — Login Name, E-Mail	Name of the SAML attribute compared with the account's <code>ad_name</code> / mail address. Defaults <code>username</code> , <code>email</code> . The value <code>NameID</code> means the assertion's NameID.
Attribute mapping — First / Last Name	Used only when an account is <i>created</i> . Defaults <code>firstName</code> , <code>lastName</code> .
Stable Person ID (optional)	An attribute that never changes for a person — an immutable directory id. Empty uses the NameID, which is right for a persistent NameID. Name an attribute here if the NameID is an e-mail address or a UPN that may be renamed, or if the IdP can only send transient NameIDs.
Find the existing account by	E-Mail, Login Name, or Login Name then E-Mail — the same rules as for OpenID Connect, see below. Applies to the first sign-on of an account only.
Create the account at the first sign-on	Off: people without an account are turned away. On: an account is created, as described below.

SAML 2.0 • Attribute names

The names must match what the IdP puts into the assertion *exactly* (upper and lower case count). With ADFS and Entra ID these are long claim-type URIs.

IdP	E-Mail	First name	Last name	Login name
Keycloak (mappers as above)	email	firstName	lastName	username
ADFS	http://schemas.xmls oap.org/ws/2005/05/ identity/claims/ema iladdress	http://schemas.xmls oap.org/ws/2005/05/ identity/claims/giv enname	http://schemas.xmls oap.org/ws/2005/05/ identity/claims/sur name	http://schemas.xmls oap.org/ws/2005/05/ identity/claims/upn
Entra ID (SAML app, default claims)	as ADFS	as ADFS	as ADFS	http://schemas.xmls oap.org/ws/2005/05/ identity/claims/nam e (the UPN)
Shibboleth / eduPerson	urn:oid:0.9.2342.19 200300.100.1.3	urn:oid:2.5.4.42	urn:oid:2.5.4.4	urn:oid:0.9.2342.19 200300.100.1.1

Finding out what an IdP really sends. Use a SAML-tracer browser extension, or decode the SAMLResponse of one sign-on (base64) and read the `<saml:Attribute Name="...">` lines. Whatever name is there goes into the panel, character for character. An attribute the IdP does not send is not an error — it is simply empty (and cannot be used to match an account).

SAML 2.0 · What is verified

The sign-on only works if all of these hold; otherwise the user is returned to the login page with a short reason.

- **The answer is signed by the IdP.** The XML signature over the Response or the Assertion must verify against one of the configured certificates. Unsigned answers, answers signed with another key and answers altered after signing are refused. Exactly one assertion may be present, which defeats signature-wrapping attacks.
- **No SHA-1.** A signature or digest using SHA-1 anywhere in the answer is refused.
- **The content fits.** Issuer equals the IdP Entity ID, Audience our Entity ID, Destination and Recipient our ACS URL exactly, the status is Success, and the answer is within its NotBefore / NotOnOrAfter window (clocks may differ by up to 3 minutes).

- **It answers our request.** `InResponseTo` must be the AuthnRequest Resourcecloud sent. Unsolicited (IdP-initiated) answers are refused. A login can be answered once, within 10 minutes; a repeated answer is refused.
 - **It is for this browser.** The IdP's answer is a cross-site POST, which the browser sends *without* the session cookie. Resourcecloud therefore keeps the pending login on the server, and the redirect that follows (which does carry the cookie) hands the result only to the session that started the login. Without that, anybody could post *their own* valid answer from a victim's browser and log the victim in as the attacker.
 - **The person has a stable id.** A transient NameID is refused unless an attribute is named as Stable Person ID.
 - **The document is well-formed SAML** (validated against the SAML schema; external entities are not loaded). All IdP URLs must be https — for a test IdP on plain http the server environment variable `rc_sso_allow_http=1` permits it; never set it in production.
-

SAML 2.0 · Certificates and rollover

Everything the IdP signs is checked against the certificate(s) stored in the panel, so an IdP certificate that is replaced, or expires, breaks the sign-on until Resourcecloud knows the new one.

- 1 **Before the cutover**, get the new certificate: re-import the IdP's metadata (it lists both while they overlap), or paste the new PEM *below* the old one in *IdP Signing Certificate(s)*. Submit.
- 2 **After the cutover**, once the IdP signs with the new certificate, remove the old one from the box.

The panel prints each certificate's expiry date; an expired one is shown in red. Note the dates of your IdP's certificates in a calendar — nothing in Resourcecloud can warn you earlier than you look at the panel.

How accounts are matched

The rules are the same for OpenID Connect and SAML — one implementation serves both; each protocol has its own matching and account-creation settings. At every sign-on Resourcecloud works through these steps and uses the first that applies:

- 1 **The identity is already linked to an account.** That account signs in. Its e-mail address, name or login name may have changed at the provider or in Resourcecloud since — the link does not care. A deactivated account is refused.
- 2 **No link yet — look for an account** by the chosen method: login name against `ad_name` , and/or e-mail against the account's mail address. Exactly one active account must match. It is then **linked** to the identity, and signs in.
- 3 **No account found.** If *Create the account* is on, one is created. Otherwise the user sees “*there is no ResourceCloud account for you*”.

Why the link matters. The link is made with the provider's stable identifier of the person — OpenID Connect: the `sub` claim; SAML: the `NameID`, or the attribute you named as Stable Person ID — stored as a hash together with the issuer (or the IdP's entity ID). After the first sign-on, an e-mail address that is renamed or reassigned at the provider cannot lead somebody else into the account. An account is linked to *one* identity: it refuses a different one — also one that arrives through the other protocol — — an RC Admin has to release the link first.

Refused matches

- An e-mail address the provider marks as *not verified* (`email_verified: false`) is never used to find an account.
- If two accounts share the matching e-mail address or login name, the sign-on is refused as ambiguous — fix the duplicate in Resourcecloud.
- An account that is deactivated stays deactivated; single sign-on does not reactivate it.

Accounts created at the first sign-on

Property	Value
Role	Worker
Team	the Default Support Team (<i>RC Settings</i> → <i>Configuration</i>); company, department and area follow from the team

Login name (<code>ad_name</code>)	the login-name claim or attribute, or the e-mail address if that is missing
User name	derived from it (the part before an <code>@</code>), with a number appended if it is taken
Password	random and never shown — the user signs in through the provider (an admin can set one, or the user can use <i>Forgot password</i>)
Licensing	counts as a licensed user from now on
Also created	team membership, the primary role, and the personal welcome task

An account is *not* created when the login name or e-mail address is missing or longer than 45 characters, when the default team or the Worker role does not exist, or when the new account would collide with an existing one. Anything beyond Worker in the default team — a different team, extra roles — is assigned by an RC Admin afterwards, as for any other user.

Think before switching on account creation. Then everybody the provider lets in becomes a licensed Resourcecloud user. With Entra ID, set *Assignment required* so that only the intended users or groups can reach the application at all (see Who may sign in).

Automatic provisioning (SCIM)

With single sign-on, people can *sign in* through your identity provider. With **SCIM 2.0** the identity provider also *creates, changes and deactivates their accounts* here and puts them in the right teams — so a new colleague in your directory has an account on day one, and a leaver loses access without anybody remembering to switch it off.

The IdP ...	... and Resourcecloud does
creates a user	creates the account: a Worker in the <i>starting team</i> , with name, e-mail address and the IdP's id of the person
changes a user (name, e-mail, title, phone)	changes the account

disables or unassigns a user (<code>active = false</code>)	deactivates the account — it is not deleted. A person who is signed in is signed out with their next click, and cannot sign in again
deletes a user	does the same: deactivates . Nothing is ever deleted through provisioning
re-enables a user	reactivates the same account — tasks, time and history are all still there
sends a group and its members	places the members in the team and gives them the secondary role that an RC Admin mapped to the group

Provisioning and sign-in are separate. SCIM only manages accounts; people still sign in through OpenID Connect or SAML. The two meet at the *userName*: it becomes the account's login name (`ad_name`), and the e-mail address its mail address — exactly what the sign-on matches an account on, so a provisioned person finds their account at their first sign-in. Entra ID sends the user principal name as *userName*; set *Find the existing account by* to *E-Mail* or *Login Name* accordingly.

Quick start

- 1 Open *RC Settings* → *External Services* → *Automatic Provisioning (SCIM)*. It shows the **Tenant URL**, `https://rc.example.com/scim/v2`.
- 2 Press **Generate new token**. The token is shown **once**, in a window — copy it before you close it. Resourcecloud keeps only a fingerprint of it and cannot show it again.
- 3 Choose the **Team for new users** (empty: the Default Support Team), switch **Accept provisioning requests** on and press **Save**. SCIM cannot be switched on before a token exists.
- 4 At the identity provider, enter the Tenant URL and the token — for Entra ID see below — and start provisioning.
- 5 When the IdP has sent its groups, they appear in the table *Groups* in the same panel. Choose for each the **team** its members belong in and/or a **secondary role** they get, and press **Save**. The members are placed at once.

The address must be reachable from the identity provider. Microsoft's provisioning service calls your Tenant URL from the internet, so Resourcecloud must be served over **https with a certificate Microsoft trusts**. An instance that is only reachable inside your network cannot be provisioned by Entra ID's cloud service (it needs the on-premises provisioning agent, which this guide does not cover).

Microsoft Entra ID

- 1 In the Entra admin center: *Enterprise applications* → *New application* → *Create your own application*, name it (for example `Resourcecloud provisioning`), choose *Integrate any other application you don't find in the gallery (Non-gallery)*. (You can also use the application you already created for single sign-on.)
- 2 *Provisioning* → *Get started*. **Provisioning Mode: Automatic**. Under *Admin Credentials* enter **Tenant URL** = the URL from Resourcecloud and **Secret Token** = the token. Press *Test Connection* — it must succeed — and *Save*.

- 3 *Mappings* → *Provision Microsoft Entra ID Users*. These are the attributes Resourcecloud understands; the rest can be removed:

Entra ID attribute	SCIM attribute	Becomes
<code>userPrincipalName</code>	<code>userName</code>	login name (<code>ad_name</code>) — required
<code>Switch([IsSoftDeleted], , "False", "True", "True", "False")</code>	<code>active</code>	active / deactivated
<code>mail</code>	<code>emails[type eq "work"].value</code>	e-mail address (if empty and the <code>userName</code> looks like one: the <code>userName</code>)
<code>givenName</code>	<code>name.givenName</code>	first name
<code>surname</code>	<code>name.familyName</code>	last name
<code>jobTitle</code>	<code>title</code>	job title
<code>mobile</code>	<code>phoneNumbers[type eq "mobile"].value</code>	mobile phone
<code>objectId</code>	<code>externalId</code>	the IdP's own id of the person — change the default <code>mailNickname</code> to <code>objectId</code> , it never changes

- 4 *Mappings* → *Provision Microsoft Entra ID Groups*: `displayName` → `displayName`, `members` → `members`, `objectId` → `externalId`.

- 5 *Settings* → *Scope*: *Sync only assigned users and groups*. Under *Users and groups* assign the people **and the groups you want pushed**. (Entra does not follow nested groups.)

- 6 Set *Provisioning Status* to **On** and save. To try it on one person first, use *Provision on demand*: pick a user, run it, and look at the result in Resourcecloud. Entra's first cycle can take a while; later ones run roughly every 40 minutes.

What Entra does when somebody **leaves**: removing the assignment (or disabling the account) makes it send `active = False` — Resourcecloud deactivates the account. When the user is finally deleted in Entra, after its retention period, Entra sends a delete — Resourcecloud deactivates again; the account stays, inactive. If the same person ever returns, Entra finds the account by its `userName` and re-enables it instead of creating a second one.

Groups: what they do

A group is only a name the IdP sends. **What membership means is your decision**, made in the table *Groups*:

Setting	Effect for the members
Team	The person is placed in that team. While their <i>home team</i> is still the starting team, the group's team becomes their home team (this is the "assign to a team" step). Otherwise they become an additional member of it, with a nominal 0 % share — adjust the share in the team settings if they really work there.
Secondary role	The person gets that role in addition to their primary one.
no team, no role	The group is known and its members are recorded, but nothing happens. Every group the IdP sends starts like this.

Mapping, changing the mapping and clearing it all take effect immediately for the members already in the group. *Prepare a group by name* lets you create the mapping before the IdP has sent the group.

What provisioning never touches

- **What an administrator did by hand.** Provisioning remembers exactly what it handed out. A team membership, a home team or a role that was already there is not recorded as its own and is never removed when the person leaves the group; a home team an admin chose is never moved.
- **When a person leaves a group**, what the group gave is taken back — and if the group's team was their home team, they are handed to their next group's team, or back to the starting team.
- **RC Admin** cannot be handed out through a group: the panel does not offer it and the server ignores it even if a mapping says so — otherwise whoever controls a group at the IdP, or holds the token, could make themselves an administrator. Primary roles (Worker, Teamlead ...) are

not offered either.

- **Accounts that hold the RC Admin role are read-only to provisioning:** the IdP can read them, but not change, deactivate or delete them, and a group does not place them. A leaked token therefore cannot lock out or rewrite an administrator.
- **Customer logins** are not part of the directory and are invisible to provisioning.

Operating it

- **Seats.** A licensed seat is an active, non-customer login. Creating, deactivating and re-enabling accounts through SCIM updates the seat count the same way the user administration does.
- **Rotating the token.** *Generate new token* makes the old one stop working at once. Enter the new one at the IdP right away; until you do, provisioning fails with "authentication failed" (the IdP retries). *Revoke token* switches provisioning off.
- **See that it works.** The panel shows when the token was made and when it was last used. Each request leaves a line in the application log, `SCIM POST /scim/v2/Users -> 201` (never the token or the person's data).
- **New accounts have no password anybody knows.** They sign in through your identity provider (an admin can still set a password in the user administration, or the person uses *Forgot password*). Each new account also gets the usual welcome task.
- **Switching it off.** Untick *Accept provisioning requests* (or revoke the token); accounts that were created stay as they are. In an emergency: `UPDATE global_settings SET scim_enabled = 0 WHERE id = 1;`

Troubleshooting provisioning

Entra shows failed operations under *Provisioning* → *Provisioning logs*. The status and message there, and the line in Resourcecloud's application log, tell the story:

Response	Meaning and fix
401 "Authentication failed"	Wrong or old token, or provisioning is switched off. Generate a token, enter it at the IdP, check <i>Accept provisioning requests</i> . After many failures from one address the server answers 429 for a while.
409 uniqueness	Another account already has that <i>userName</i> , e-mail address or <i>externalId</i> (case does not matter). Entra looks a user up before creating it, so this usually means two directory people share an address, or an account created by hand has the same <i>externalId</i> . Fix the duplicate at either end.

400 invalidValue — “longer than 45 characters”	An account's login name and e-mail address hold at most 45 characters. A user principal name longer than that cannot be provisioned; the account is not created. (Names and titles longer than their fields are shortened, not refused.)
400 invalidFilter	The IdP asked for something other than <code>eq</code> , <code>co</code> or <code>sw</code> on <code>userName</code> , <code>externalId</code> , <code>displayName</code> , <code>id</code> or <code>e-mail</code> . Not supported.
400 unknown group member	A group lists a person that does not exist here (yet) — or is a customer login. The whole change is refused so that nothing is half-applied; Entra retries it after the user has been created.
403 “holds the RC Admin role”	The IdP tried to change, deactivate or delete an administrator account. By design; change administrators in the user administration.
500 “No default team is set up”	There is no valid team for new users: choose one under <i>Team for new users</i> — it must be a Team, not an area or a department.
The person was created but is in the wrong team	Their group is not mapped (the table shows it with <i>(no team)</i>), or an administrator chose their home team by hand — then the group's team is an extra membership. Look at the groups table.
The person cannot sign in after being created	Provisioning only creates the account. Check that single sign-on is set up and matches by the right field (How accounts are matched).

Limits of provisioning

- One identity provider per instance. Users and groups only: no bulk operations, sorting or ETags; the list endpoints return at most 200 per page.
- A group maps to **one team and one secondary role**. Roles with wider reach (RC Admin) and primary roles are assigned in the user administration.
- Attributes Resourcecloud has no field for — manager, department, address, the enterprise extension — are accepted and ignored.
- No password is synchronised, and there is no “change password” through SCIM.
- Deleting never deletes: a person who left stays as an inactive account (so their history stays), and so does a person whose account you want gone — remove those in the user administration.
- Entra ID's provisioning service has to reach the Tenant URL over the internet (see above).

- Not verified against a live Entra ID tenant: the automated tests drive the interface exactly as Entra documents its requests (look-up before create, capitalised `Replace`, `active` as the text `"False"`, filtered paths, enterprise-extension attributes). Use *Provision on demand* for a first real run.

Operating it

Password login stays — and is your fallback

Single sign-on adds a way in; it never removes one. Users with a password can still use the form, and so can you. Keep **at least one RC Admin account with a password you know**, and do not rely on an account that exists only through the provider.

If single sign-on is misconfigured and you cannot even reach the settings page, sign in with the password of an RC Admin. If that is not possible either, switch the feature off in the database:

```
UPDATE global_settings SET oidc_enabled = 0, saml_enabled = 0 WHERE id = 1;
```

Releasing the link of an account

When a person gets a new identity at the provider (a new directory, a rebuilt user), the account refuses it with *"linked to a different identity"*. There is no screen for this yet; an administrator with database access releases the link, and the next sign-on links the account anew:

```
UPDATE login SET oidc_subject = NULL WHERE ad_name = '<login name>';
```

Rotating the client secret (OpenID Connect)

Create the new secret at the provider, paste it into the panel and press Submit, then delete the old secret at the provider. The sign-on keeps working throughout.

Changing the issuer or IdP, or moving between protocols

A different issuer — or a different SAML IdP, or moving from OpenID Connect to SAML — means different identities: accounts linked to the old one will not be recognised and fall back to matching by e-mail or login name — and are then refused, because they are still linked to the old identity. Release the links first (`UPDATE login SET oidc_subject = NULL`) if you move to a new provider.

Caching (OpenID Connect)

The provider's configuration and signing keys are kept for one hour in the server's temporary directory, per issuer. Saving the settings with single sign-on enabled refreshes them at once.

Where to see what went wrong

The login page shows only a short, fixed sentence. The detail goes to the application log, as a line starting with `OIDC login failed (<code>): ...` or `SAML login failed (<code>): ...`. In a Docker installation that is `docker logs resccloud-web`.

Multi-tenant (SaaS) installations

Each tenant configures its own provider. The redirect URI (OpenID Connect) and the Entity ID, ACS URL and metadata URL (SAML) of a tenant are built from *its* address, and its client secret or IdP certificates are stored in the tenant's own database. A customer registers their own application at their own provider.

Troubleshooting

Messages on the login page (both protocols)

Message	Code	Usual cause and fix
Single sign-on is not set up for this installation.	<code>disabled</code>	The feature was switched off while the user was signing in, or the button was opened from a stale page.

The sign-on took too long or was started in another window.	state	More than 10 minutes between button and return, the login was started in another browser or tab, the back button was used, or cookies are blocked. Try again in one window.
The identity provider could not confirm your sign-on.	provider	Anything between Resourcecloud and the provider — read the log line, see the next table.
There is no ResourceCloud account for you.	noaccount	No account matches, and account creation is off (or not possible). Create the account, adjust the matching method or the claim mapping, or switch on account creation.
Your ResourceCloud account is deactivated.	inactive	Reactivate the account, or leave it as it is.
More than one account matches your identity.	ambiguous	Two accounts share an e-mail address or login name. Fix the duplicate.
Linked to a different identity at the provider.	linked	See Releasing the link.
Your account has no valid role.	role	The account's role is not one Resourcecloud knows. Fix it in the user administration.

OpenID Connect: details in the log (code provider)

Log says	Cause and fix
discovery names issuer "...", configured is "..."	The issuer in the panel is not exactly the one the provider reports. Copy it from the discovery document. For Entra ID: tenant ID, ending in /v2.0 ; common does not work.
... is not an acceptable https URL	The issuer or an endpoint is plain http.
... answered HTTP 0 ... Could not resolve host / timed out	The server cannot reach the provider: DNS, firewall, or the proxy under <i>Configuration</i> → <i>Proxy Configuration</i> .
token endpoint answered HTTP 401: invalid_client	Wrong client secret, or it has expired. Entra ID: AADSTS7000215 (wrong value — you may have copied the <i>Secret ID</i>) or AADSTS7000222 (expired).

token endpoint answered HTTP 400: invalid_grant	The redirect URI differs from the registered one, or the code was used twice. Compare the URI character by character, including https and the port.
ID token was not issued for this client (aud)	Wrong client ID in the panel.
ID token iss "..." is not the configured issuer	The provider signs with a different issuer than it advertises — typically a v1/v2 mix-up in Entra ID. Use the /v2.0 issuer and an app registration that issues v2 tokens.
ID token has expired / is dated in the future	The server clock is off by more than a minute. Fix time synchronisation (NTP).
no published key verifies the ID token	The token is signed with a key the provider does not publish — or the key set could not be fetched. Check proxy and reachability.
provider returned access_denied	The user declined, or the provider does not let this user in (Entra ID: <i>Assignment required</i> without an assignment, a conditional-access block).
cannot create an account: username/e-mail claim missing or longer than 45 characters	The provider sends no login-name/e-mail claim under the mapped name, or the value is too long. Fix the claim mapping.
cannot create an account: role "Worker" or the default team does not exist	Set a valid Default Support Team in <i>RC Settings</i> → <i>Configuration</i> .

OpenID Connect: messages at Entra ID

Entra ID message	Cause and fix
AADSTS50011 — redirect URI does not match	The URI in the app registration must equal the one in the Resourcecloud panel exactly, and be of platform <i>Web</i> .
AADSTS700016 — application not found in the directory	Wrong client ID, or the app is registered in a different tenant than the issuer's.
AADSTS65001 — consent required	An administrator grants consent under <i>API permissions</i> .

AADSTS50105 — user not assigned to the application	<i>Assignment required</i> is on and this user has no assignment.
--	---

SAML: details in the log

The log line is `SAML login failed (<code>): <detail>`. The detail of a rejected answer starts with `answer rejected:` and comes from the XML-signature and SAML checks.

Log says	Cause and fix
No Signature found. SAML Response rejected	The IdP signs neither the response nor the assertion. Switch signing on (ADFS signs the assertion by default; Keycloak: <i>Sign assertions</i> or <i>Sign documents</i>).
Signature validation failed	The certificate in the panel is not the one the IdP signs with — wrong certificate, or the IdP rolled it. Compare the fingerprint, re-import the metadata. See Certificates and rollover.
Reference validation failed	The signed content was changed after signing — by the IdP's own post-processing, or on the way. Keep a proxy from rewriting the posted form.
the answer is signed with SHA-1	Set the IdP to SHA-256 (ADFS: relying party trust → <i>Advanced</i> → secure hash algorithm).
Could not validate timestamp: expired / not yet valid. Check system clock.	The clocks of the IdP and of Resourcecloud differ by more than 3 minutes (fix time synchronisation), or the user took more than the assertion's lifetime (usually 5 minutes) between sign-in and the answer.
Invalid audience for this Response (expected 'X', got 'Y')	The identifier of the relying party at the IdP is not our Entity ID. Make the two equal — the IdP's value is the one to change.
Invalid issuer in the Assertion/Response	The <i>IdP Entity ID</i> in the panel is not what the IdP puts into its answers. For ADFS it is <code>http://<adfs>/adfs/services/trust</code> — <code>http</code> , not <code>https</code> . Copy it from the metadata.

The response was received at X instead of Y	The ACS URL registered at the IdP is not the one Resourcecloud answers at — or the <code>baseURL</code> of the installation is not the public address (behind a proxy).
A valid SubjectConfirmation was not found on this Response	Recipient or validity of the bearer confirmation does not fit; in practice an ACS URL mismatch.
The InResponseTo of the Response ... does not match	The sign-on was started a second time (another tab) and this answer belongs to the earlier request; or a stale answer was replayed. Start again from the login page.
SAML Response must contain 1 assertion	The IdP sends several assertions in one answer. Not supported.
The status code of the Response was not Success	The IdP itself refused the sign-on — the user may not use this application, or a requested setting cannot be met. The IdP's own log says why (ADFS: <i>Event Viewer</i> → <i>Applications and Services Logs</i> → <i>AD FS</i> → <i>Admin</i>).
NameID not found in the assertion	The IdP sent no NameID. ADFS needs the <i>Name ID</i> claim rule; Keycloak a name ID format.
the NameID is transient	The IdP sends a different NameID at each sign-on. Configure a persistent one, or name an attribute under <i>Stable Person ID</i> .
the answer carries no stable identifier of the person	The attribute named as Stable Person ID is not in the assertion (names are case-sensitive).
Invalid SAML Response. Not match the saml-schema-protocol-2.0.xsd	The IdP's answer is not valid SAML 2.0.
answer does not belong to a pending login (code state)	An IdP-initiated answer, an answer posted twice, or more than 10 minutes between the click and the answer. Start from the login page.
the result was not asked for by this browser session (code state)	Cookies are blocked, or the sign-on continued in a different browser or profile than it started in.

SAML settings are not usable	The configuration is incomplete or invalid — open the panel and save it again; the message names what is wrong.
login page: <i>no ResourceCloud account for you</i> (code noaccount)	The sign-on worked, but no account matches. The attribute names in the mapping must equal the ones the IdP sends; an attribute that is not sent cannot match.

Limitations

- One OpenID Connect provider and one SAML identity provider per instance; both may be switched on side by side.
- Password login cannot be switched off; there is no “single sign-on only” mode.
- Roles and teams are not taken from provider groups or claims; they are assigned in Resourcecloud.
- Name and e-mail are read from the provider only when an account is created; later changes at the provider do not update the account.
- Signing out of Resourcecloud does not sign the user out at the provider (no single logout), so the next click on the button may sign in without a prompt.
- Releasing the link of an account is a database operation.
- For Entra ID (OpenID Connect), only tenant-specific issuers are supported.
- The client secret is stored in the database like the other credentials in the global settings (SMTP password, API keys).

SAML only

- Service-provider-initiated sign-on only; IdP-initiated answers are refused by design. The AuthnRequest goes by HTTP-Redirect, the answer comes by HTTP-POST; there is no artifact binding.
- Resourcecloud holds no key pair of its own: it does **not sign** AuthnRequests and cannot **decrypt** assertions. An IdP that requires either (Keycloak: *Client signature required*; ADFS: an encryption certificate on the trust) must be told not to.
- SHA-1 signatures are refused, and the clocks of IdP and server may differ by 3 minutes at most.
- No single logout: signing out of Resourcecloud does not end the session at the IdP.

- The IdP's signing certificate is stored when the metadata is imported; a rollover at the IdP needs a re-import (the panel shows the expiry dates).
- An account is linked to one identity: the same person cannot use OpenID Connect and SAML identities on one account without releasing the link in between.

Technical reference

Item	Where
Start of the login	<code>oidc_login.php</code> — public; only redirects to the provider. Accepts <code>?link=</code> , the page to return to, limited to this site.
Return from the provider	<code>oidc_callback.php</code> — public; acts only on a <code>state</code> of the same session.
Saving the settings	<code>set_oidc_config.func.php</code> — RC Admin only (role map and endpoint policy).
Library	<code>app_libs/oidc.func.php</code> (flow, token checks, account matching); <code>app_libs/user_create.func.php</code> (account creation, shared with the admin form).
Settings	Columns <code>oidc_*</code> of <code>global_settings</code> , row 1 — migration 151, also in <code>init_db.sql</code> .
Account link	<code>login.oidc_subject = sha256(issuer + "\n" + sub)</code> , unique.
Redirect URI	<code><baseURL>/oidc_callback.php</code> — always with <code>.php</code> , independent of the clean-URL setting.
SAML: the four endpoints	<code>saml_login.php</code> (starts; redirects to the IdP), <code>saml_acs.php</code> (the IdP's POST), <code>saml_complete.php</code> (same-site step: hands the result to the right browser, signs in), <code>saml_metadata.php</code> (SP metadata) — all public; the first three act only on a login this instance started. Saving: <code>set_saml_config.func.php</code> , RC Admin only.
SAML: library	<code>app_libs/saml.func.php</code> (flow, validation, metadata import) on <code>onelogin/php-saml 4.3</code> ; <code>app_libs/sso_account.func.php</code> (account matching, creation, sign-in — shared with OpenID Connect).

SAML: settings and state	Columns <code>saml_*</code> of <code>global_settings</code> and table <code>saml_flows</code> (logins under way, keyed by a hash of the RelayState token) — migration 153, also in <code>init_db.sql</code> . Links use <code>login.oidc_subject = sha256("saml\n" + IdP entity ID + "\n" + id)</code> .
SAML: addresses	ACS <code><baseURL>/saml_acs.php</code> — always with <code>.php</code> : a POST is not redirected to the clean URL, and the <i>Destination</i> must equal it. Entity ID <code><baseURL>/saml_metadata.php</code> ; metadata served at <code><baseURL>/saml_metadata</code> .
SAML: tests	<code>php htdocs/tests/saml_flow_test.php</code> — self-contained: hand-signed answers with one defect each (expired, wrong audience, SHA-1, smuggled assertion, ...), each asserted for its specific reason. <code>php htdocs/tests/saml_keycloak_test.php</code> — against a real Keycloak (see the file header): metadata both ways, real sign-ons with the answer posted cookie-less, every signature option, replay, IdP-initiated, login CSRF, matching, creation, passkey 2FA.
SCIM: endpoint	<code>/scim/v2/Users</code> , <code>/Groups</code> , <code>/ServiceProviderConfig</code> , <code>/ResourceTypes</code> — front controller <code>scim/index.php</code> (rewrite in <code>scim/.htaccess</code>), authenticated by the bearer token only; no session, no CSRF. Settings page savers: <code>set_scim_config.func.php</code> , <code>set_scim_group_map.func.php</code> (RC Admin only).
SCIM: library and data	<code>app_libs/scim.func.php</code> . Columns <code>scim_*</code> of <code>global_settings</code> (token kept as sha256), <code>login.scim_external_id</code> , tables <code>scim_groups</code> (the mapping), <code>scim_group_members</code> and <code>scim_grants</code> (what provisioning handed out) — migration 154, also in <code>init_db.sql</code> . Deactivation is <code>login.active = 0</code> ; <code>auth.php</code> checks it on every request.
SCIM: tests	<code>php htdocs/tests/scim_test.php</code> — an Entra-style client against <code>/scim/v2</code> (lookup-then-create, capitalised Replace, "False" as text, filtered paths, group pushes, deactivation with a live session, hand-made access left alone, admin accounts protected). <code>php htdocs/tests/scim_settings_test.php</code> — the settings page and savers as a real admin and a worker (one-time token, rotation, mapping guards).
OpenID Connect: test	<code>php htdocs/tests/oidc_flow_test.php</code> — walks real sign-ons through a mock provider (<code>htdocs/tests/oidc_mock_idp.php</code>): token faults, state replay, matching, creation, passkey 2FA.