AWS Cognito

Your people sign in to Gebo.ai with the accounts of an Amazon Cognito user pool. Add Gebo.ai as an app client of the pool, then enter it in Setup.

TIME	15 minutes.
YOU NEED	An AWS account that can manage the user pool, a Cognito domain of the pool, and Gebo.ai open at the address your users use.

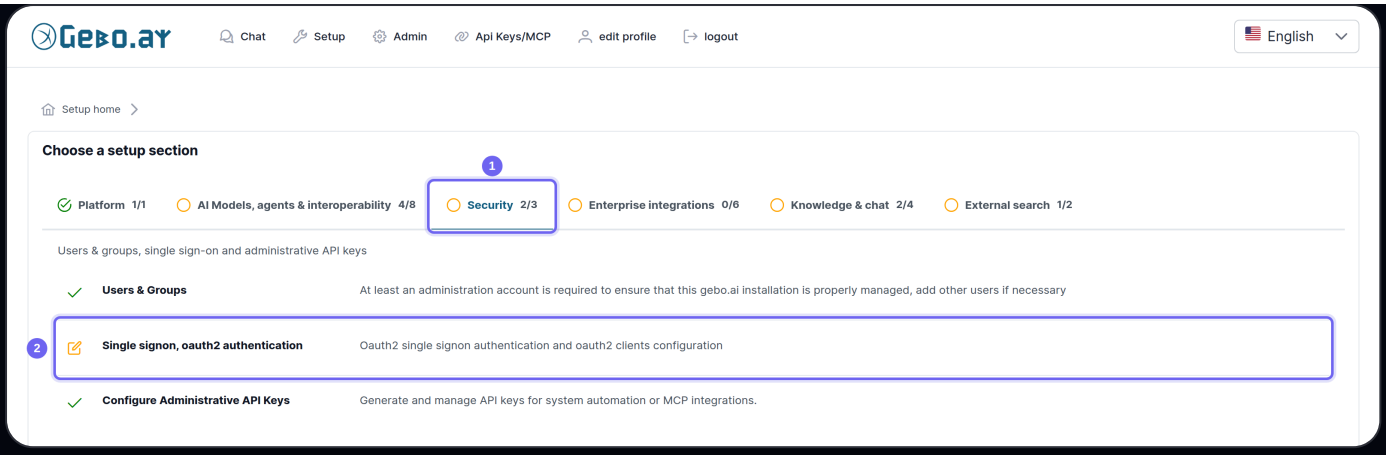
1 Add Gebo.ai to the user pool

1. In the AWS console open **Amazon Cognito** → **User pools** and the pool. Copy its **User pool ID**, for example `eu-west-1_AbCdEf123`: the part before the underscore is the region.
2. In **App clients** click **Create app client**, choose **Traditional web application**, name it *Gebo.ai* and create it. Copy the **Client ID** and the **Client secret**.
3. In **Domain** check the pool's Cognito domain, `https://<prefix>.auth.<region>.amazoncognito.com`: Gebo.ai needs only the prefix.

> **The Cognito domain**
> Gebo.ai builds the sign-in addresses from the prefix of the pool's Cognito domain and from the region: enter that prefix, also when the pool has a custom domain.

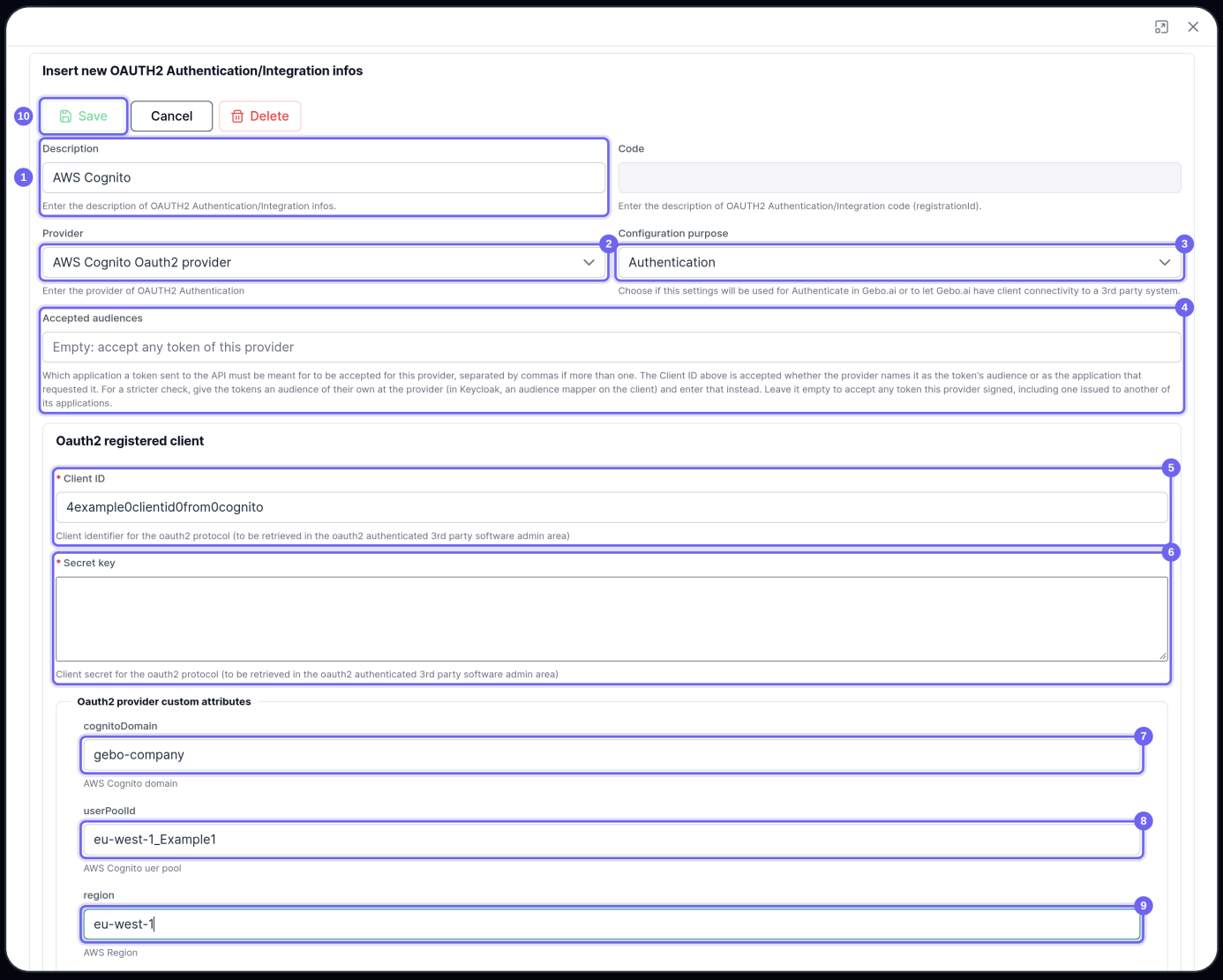
2 Open the single sign-on settings

In **Setup** open **Security** (1) and click **Single signon, oauth2 authentication** (2), then **+ Oauth2 config**.



The Security tab of Setup.

3 Enter the app client



A new configuration for AWS Cognito. The secret is still to be entered.

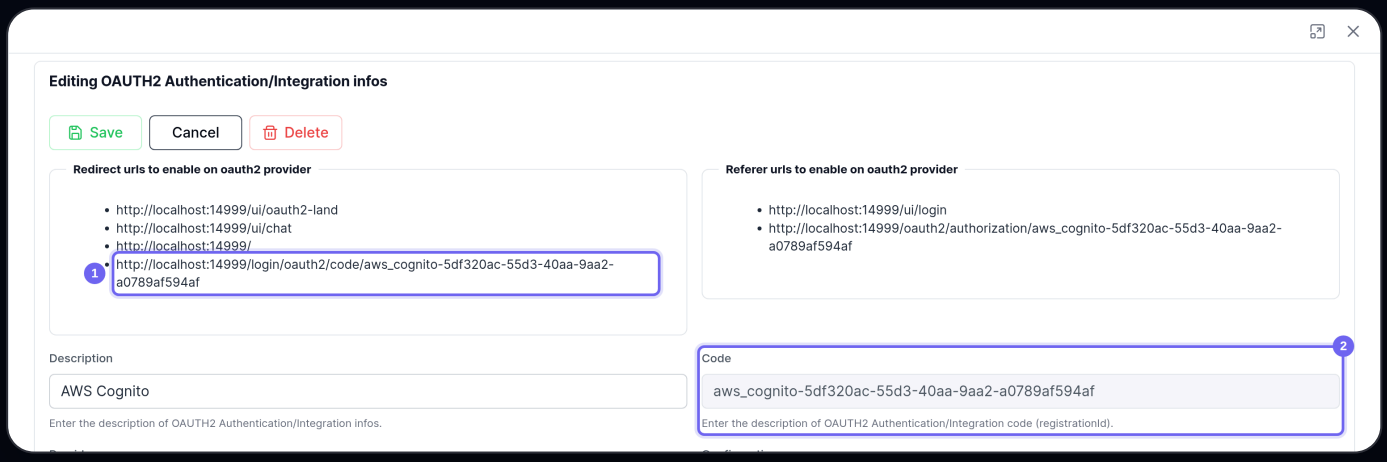
1. **Description:** the name on the login button, for example *AWS Cognito*.
2. **Provider:** *AWS Cognito Oauth2 provider*.

3. **Configuration purpose:** *Authentication*.
4. **Accepted audiences:** only for tools that call the Gebo.ai API with tokens of this provider (page 28b). Enter the **Client ID** of item 5 to accept only tokens meant for Gebo.ai; empty, any token this provider signed is accepted. Signing in with the button does not depend on it.
5. **Client ID:** the identifier the provider gave Gebo.ai.
6. **Secret key:** the client secret the provider gave Gebo.ai.
7. **cognitoDomain:** the prefix of the Cognito domain only, for example `gebo-company`.
8. **userPoolId:** the **User pool ID**.
9. **region:** the region of the pool, for example `eu-west-1`.
10. Click **Save**.

---

4 Give the provider the redirect address

Save closes the dialog. Click **Modify** next to the new configuration: Gebo.ai has given it a **Code** (2) and lists the addresses to enable at the provider. The redirect address to register is the one ending in `/login/oauth2/code/` followed by the code (1).



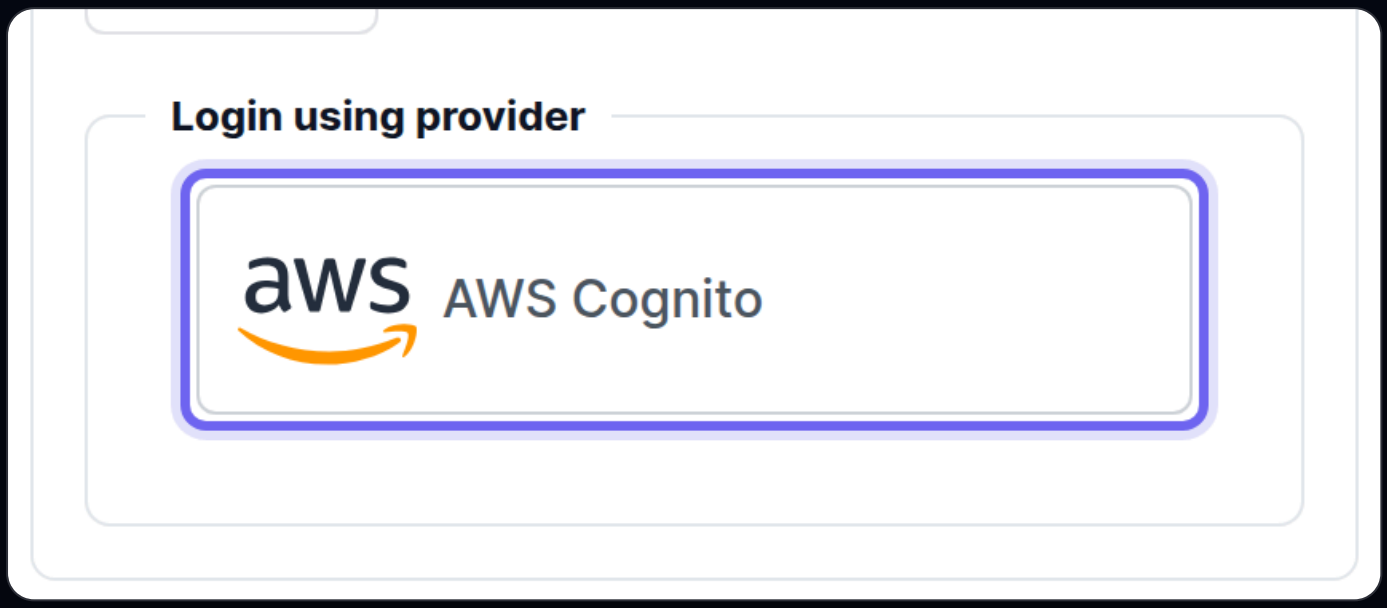
The saved configuration: its redirect address (1) and its code (2).

In Cognito open the app client, edit its **Login pages**: paste the address among the **Allowed callback URLs**, keep the **Authorization code grant** and the **OpenID**, **Email** and **Profile** scopes, and save.

> **Open Gebo.ai at its real address first**
>
> The addresses are built from the address your browser has Gebo.ai open at. Open Gebo.ai at the address your users use, for example `https://gebo.company.com`, before copying it: an address with `localhost` works only on the server itself.

5 Let people in

The login page now shows a button with the **Description** of the configuration.



The AWS Cognito button on the login page.

By default Gebo.ai lets in the people an administrator has created: for each of them create a user whose **Username** is their e-mail address in the user pool and whose **Authentication provider** is this provider, as described in page 29. To make every person the provider recognizes a user on their own, see page 28b.

← Google

Next: Keycloak →