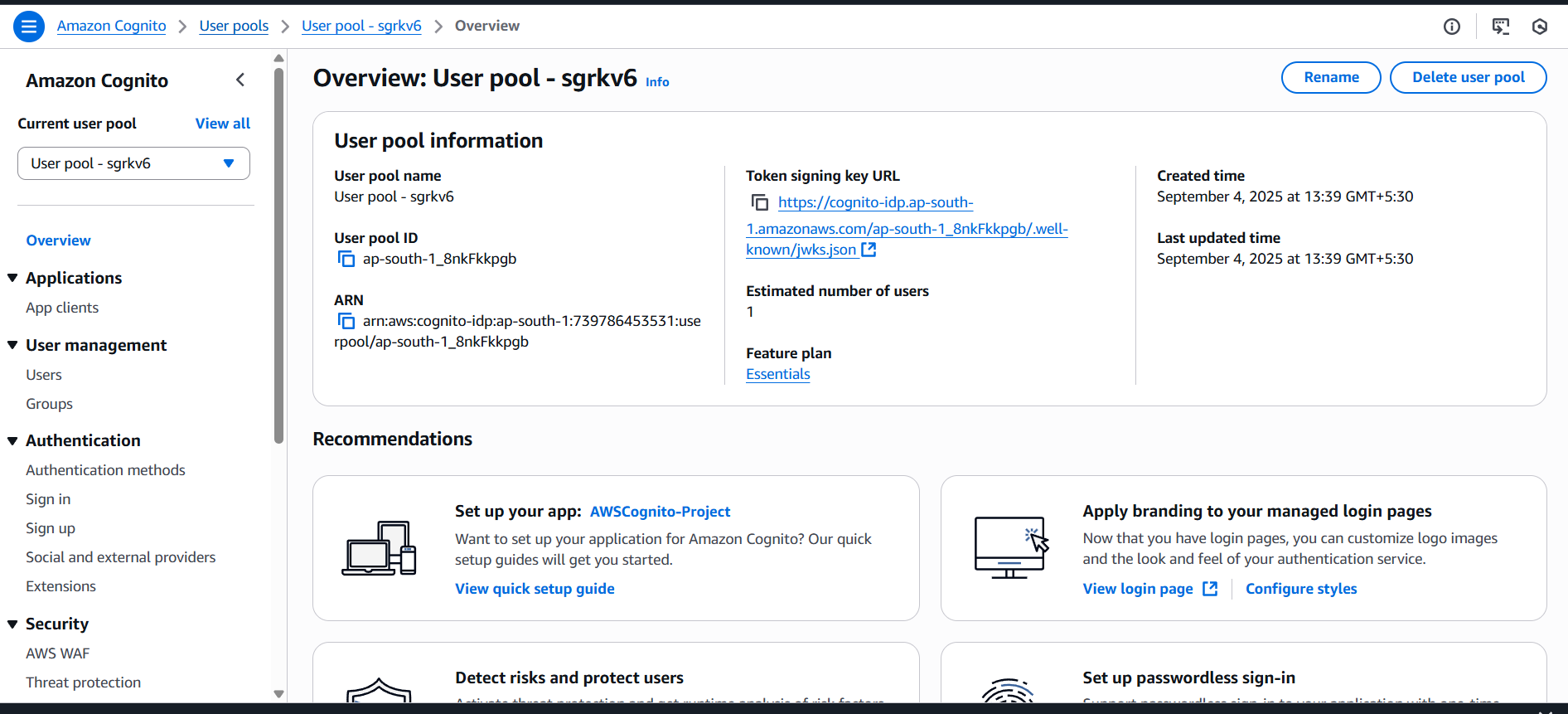
**1. Create a Cognito User Pool**

1. Go to **AWS Management Console → Cognito → User Pools**.
2. Click **Create User Pool**.
3. Configure:
   * Pool name: UserPool-sgrkv6
   * Sign-in options → Choose **Email** (or Username, or both).
   * Enable **Self sign-up** if you want users to register.
   * Configure password policy (defaults are fine for now).
4. Create a **User Pool Client** (App client):
   * Example: AWSCognito-Project.
   * **Do not enable client secret** if this will be called from browsers/mobile apps.
5. Save:
   * **User Pool ID** (e.g., ap-south-1\_XXXXXX)
   * **App Client ID** (e.g., 2jhq4d1fabc123example)



**2. Create a Lambda Function**

1. Go to **AWS Management Console → Lambda → Create function**.
2. Select **Author from scratch**.
   * Function name: Lambdaproject
   * Runtime: **Python 3.12** (or Node.js if you prefer).
3. Example Python code:

import json

def lambda\_handler(event, context):

return {

"statusCode": 200,

"body": json.dumps({

"message": "Hello from Lambda!",

"input": event

})

}

1. Click **Deploy**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

**3. Create an API Gateway**

1. Go to **Amazon API Gateway → Create API → REST API**.  
   (You can also use **HTTP API** for a simpler, faster option).
2. Create a **resource**: /hello.
3. Create a **method**: GET.
   * Integration type: **Lambda Function**.
   * Choose your Lambda (Lambdaproject).
4. Deploy the API:
   * Create a **Stage** (e.g., dev).
5. Copy the **Invoke URL**:
6. https://<api-id>.execute-api.ap-south-1.amazonaws.com/dev/hello

A screenshot of a computer

AI-generated content may be incorrect.

**4. Add Cognito Authorizer to API Gateway**

1. In API Gateway, go to **Authorizers → Create new authorizer**.
2. Configure:
   * Type: **Cognito**
   * Name: ClientA
   * Choose your **Cognito User Pool**
   * Region: same region as pool (ap-south-1)
3. Save it.
4. Go to your API method (/hello → GET) →
   * Click **Method Request**
   * Under **Authorization**, select ClientA (your Cognito authorizer).
5. **Deploy API again** to apply changes.

A screenshot of a computer

AI-generated content may be incorrect.

**5. Test the Setup**

**A) Get a JWT Token**

**Cognito Hosted UI**

* URL format:
* https://<your-domain>.auth.<region>.amazoncognito.com/login?
* client\_id=<APP\_CLIENT\_ID>&
* response\_type=token&
* scope=openid&
* redirect\_uri=https://oauth.pstmn.io/v1/callback
* Log in → You’ll get redirected with #id\_token=...&access\_token=....

A screenshot of a computer

AI-generated content may be incorrect.

**B) Call API with Token**

Use **AccessToken** in the Authorization header:

curl -X GET "https://<api-id>.execute-api.ap-south-1.amazonaws.com/dev/hello" \

-H "Authorization: Bearer <JWT\_ACCESS\_TOKEN>"

✅ If token is valid → You’ll get Lambda’s response.  
❌ If token is invalid/expired → API Gateway returns **401 Unauthorized**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**6. Final Workflow**

1. User signs up/signs in with **Cognito** → receives JWT token.
2. User sends request to **API Gateway** with Authorization: Bearer <token>.
3. **API Gateway Authorizer** validates token with Cognito.
4. If valid → forwards request to **Lambda**.
5. Lambda runs code → returns response.