

Lab10

Authentication with AWS

(Amplify + Cognito)

WebApp

DataLab, CS, NTHU

2019 Spring

AWS Amplify

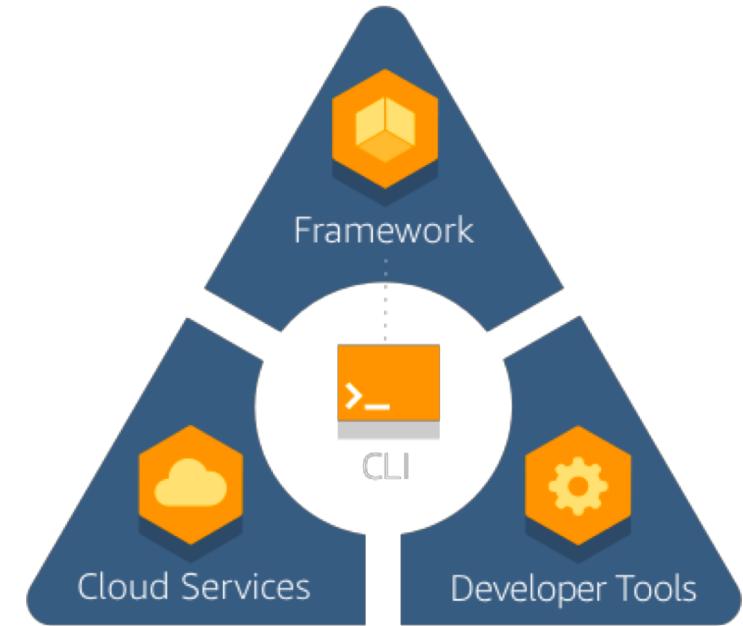


- AWS Amplify is a deployment and hosting service for modern web applications
- Provisions and manages backend services for your mobile applications
- Provides a simple framework to easily integrate your backend with your iOS, Android, Web, and React Native frontends

AWS Amplify



- Amplify is associated with services including:
 - Hosting
 - Auth
 - Analytics
 - Storage
 - Api
 - Function
- We will be using the auth service, AWS cognito



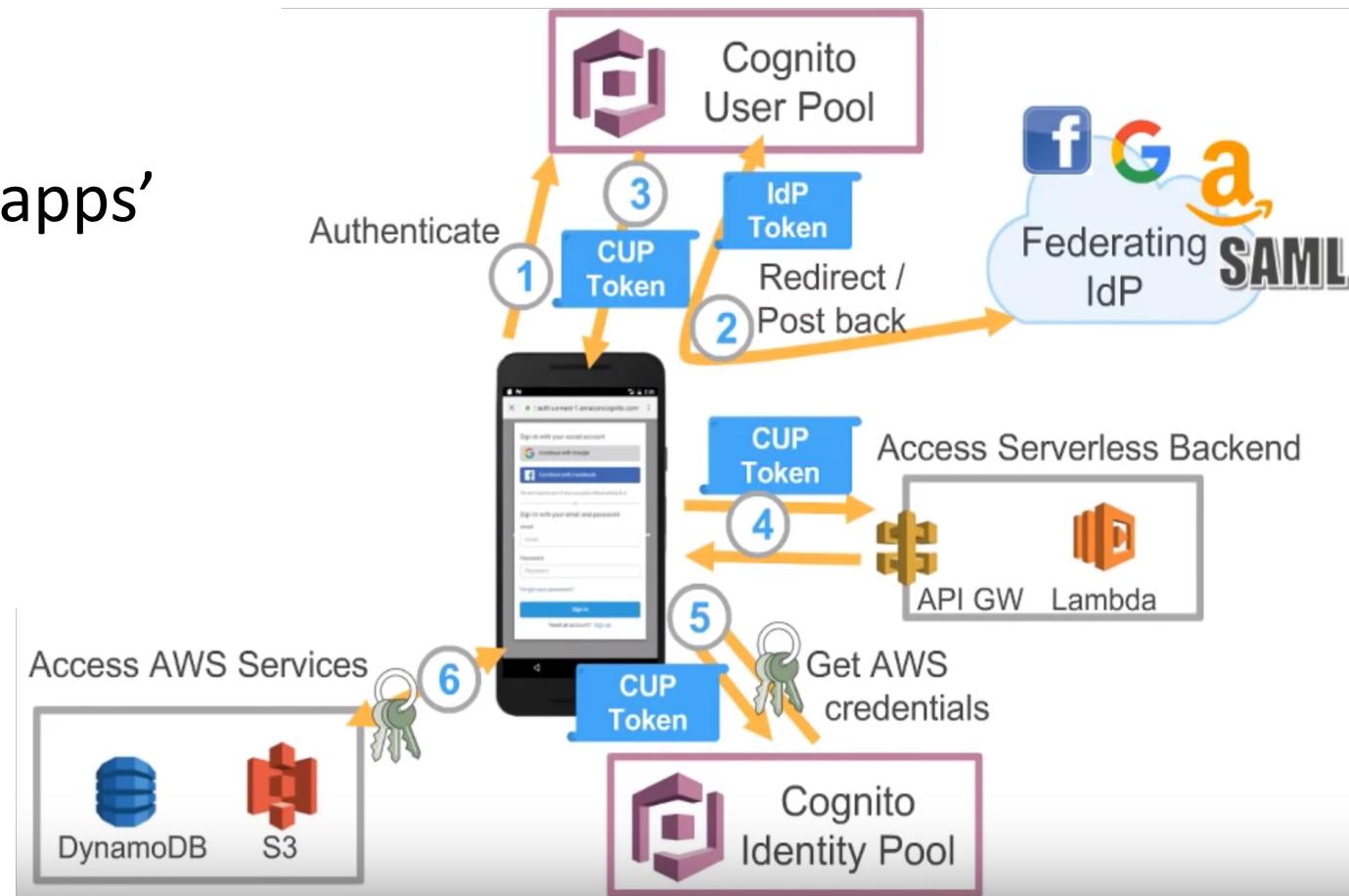
AWS Cognito



- Cognito provides sign-up and authentication functions to your mobile and web apps
- Also enables authentication through external identity providers, and provides temporary credentials to access your app's backend resources in AWS

AWS Cognito

- Use user pool to stores your users' profile attributes
- Use identity pool to keep your apps' federated identities organized



Lab-weathermood-auth

- In today's lab, we will add basic authentication to our websites
 - clone the backbone project or use your own
 - <https://shwu10.cs.nthu.edu.tw/courses/web-app/2019-spring/lab-weathermood-auth.git>



Install Amplify

- npm install -g @aws-amplify/cli
- amplify configure
 - Create an IAM user for amplify
- amplify init

```
luke@MacBook-Pro-2 ~ /Documents/WebAPP/lab-weathermood-auth/lab-weathermood-auth amplify init
Note: It is recommended to run this command from the root of your app directory
? Enter a name for the project lab-weathermood-auth
? Enter a name for the environment env
? Choose your default editor: Visual Studio Code
? Choose the type of app that you're building javascript
Please tell us about your project
? What javascript framework are you using react
? Source Directory Path: src
? Distribution Directory Path: dist
? Build Command: npm run-script build
? Start Command: npm run-script start

? Do you want to use an AWS profile? Yes
? Please choose the profile you want to use amplify-90gyy
```

Add hosting service

- amplify add hosting

```
luke@MacBook-Pro-2 ~Documents/WebAPP/lab-weathermood-auth/lab-weathermood-auth ➔ amplify add hosting  
? Select the environment setup: DEV (S3 only with HTTP)  
? hosting bucket name lab-weathermood-auth-20190518012705-hostingbucket  
? index doc for the website index.html  
? error doc for the website index.html
```

You can now publish your app using the following command:

Command: amplify publish

- amplify push

- Push local changes to the cloud
- This will generate a export file, which will be used later

Add authentication service

- amplify add auth

```
luke@MacBook-Pro-2 ~ ~/Documents/WebAPP/lab-weathermood-auth master amplify add auth
Using service: Cognito, provided by: awscloudformation
```

The current configured provider is Amazon Cognito.

Do you want to use the default authentication and security configuration? Default configuration

Warning: you will not be able to edit these selections.

How do you want users to be able to sign in when using your Cognito User Pool? Username

Warning: you will not be able to edit these selections.

What attributes are required for signing up? (Press <space> to select, <a> to toggle all, <i> to invert selection) Email

Successfully added resource cognito541d1555 locally

Add packages to your project

- Install package for development
 - npm install -g yarn
 - yarn add aws-amplify aws-amplify-react
- Import packages

```
import Amplify from 'aws-amplify';
import awsmobile from '../aws-exports';
import { withAuthenticator } from 'aws-amplify-react';

Amplify.configure(awsmobile);
```

Add authentication in your project

- Wrap your main component with HOC function

```
export default withAuthenticator(connect(state => ({  
  ...state.main,  
  searchText: state.searchText,  
}))(Main));
```

- amplify push
- npm run start
- You should see a login screen in your website

Sign in to your account

用户名 *

Enter your username

密码 *

Enter your password

[Forget your password?](#) [Reset password](#)

No account? [Create account](#)

登录

Add signin with google

- Create google client ID for google login later
- Go to google developer console
 - <https://console.developers.google.com>
- Create a project
- Credentials page -> Create OAuth ID
 - Configure consent screen
 - Configure your Url
 - Copy ID and secret after finish



Add signin with google

- Create google client ID for google login later
- Go to google developer console
 - <https://console.developers.google.com>
- Create a project
- Credentials page -> Create OAuth ID
 - Configure consent screen
 - Configure your Url
 - Copy ID and secret after finish

← 建立 OAuth 用戶端 ID

應用程式類型
 網路應用程式
 Android [瞭解詳情](#)
 Chrome 應用程式 [瞭解詳情](#)
 iOS [瞭解詳情](#)
 其他

名稱 [?](#)
網路用戶端 1

限制
請輸入 JavaScript 來源及/或重新導向 URI。[瞭解詳情](#)
必須先將來源和重新導向網域加到 [OAuth 同意設定](#)的授權網域清單中。

已授權的 JavaScript 來源
這是用戶端應用程式的來源 URI，可用於瀏覽器發出的要求。其中不得包含萬用字元 (`https://*.example.com`) 或是路徑 (`https://example.com/subdir`)。如果您使用的是非標準通訊埠，就必須把這個通訊埠包含在來源 URI 中。

http://localhost:8080 [刪除](#)

https://www.example.com [刪除](#)
請輸入網域，或按 Enter 鍵新增網域

已授權的重新導向 URI
重新導向 URI 用於網路伺服器發出的要求。使用者透過 Google 進行驗證後，系統就會將他們重新導向至應用程式中的這個路徑。此路徑會附帶存取的授權碼。路徑中必須含有通訊協定，不得含有網址片段或相對路徑，而且不能是公開的 IP 位址。

http://localhost:8080 [刪除](#)

https://www.example.com [刪除](#)
請輸入網域，或按 Enter 鍵新增網域

Add signin with google

- Create google client ID for google login later
- Go to google developer console
 - <https://console.developers.google.com>
- Create a project
- Credentials page -> Create OAuth ID
 - Configure consent screen
 - Configure your Url
 - Copy ID and secret after finish



Add signin with google

```
luke@MacBook-Pro-2 ➔ ~/Documents/WebAPP/lab-weathermood-auth ➔ ✚ master • ➔ amplify update auth
Please note that certain attributes may not be overwritten if you choose to use defaults settings.
Using service: Cognito, provided by: awscloudformation
What do you want to do? Walkthrough all the auth configurations
Select the authentication/authorization services that you want to use: User Sign-Up, Sign-In, connected with AWS IAM controls (Enables per-user Storage features for images or other content, Analytics, and more)
Allow unauthenticated logins? (Provides scoped down permissions that you can control via AWS IAM) No
Do you want to enable 3rd party authentication providers in your identity pool? Yes
Select the third party identity providers you want to configure for your identity pool: Google
```

You've opted to allow users to authenticate via Google. If you haven't already, you'll need to go to <https://developers.google.com/identity> and create an App ID.

```
Enter your Google Web Client ID for your identity pool: 1060937965775-8f576jlvsko0trg58g1s9t73qhbqismc.apps.googleusercontent.com
Multifactor authentication (MFA) user login options: OFF
Email based user registration/forgot password: Enabled (Requires per-user email entry at registration)
Please specify an email verification subject: Your verification code
Please specify an email verification message: Your verification code is {####}
Do you want to override the default password policy for this User Pool? Yes
Enter the minimum password length for this User Pool: 8
Select the password character requirements for your userpool: Requires Lowercase, Requires Uppercase, Requires Numbers
Specify the app's refresh token expiration period (in days): 30
Do you want to specify the user attributes this app can read and write? No
Do you want to use an OAuth flow? No
Successfully updated resource cognito0e75f339 locally
```

Add signin with google

- Pass your google client ID to withAuthenticator function

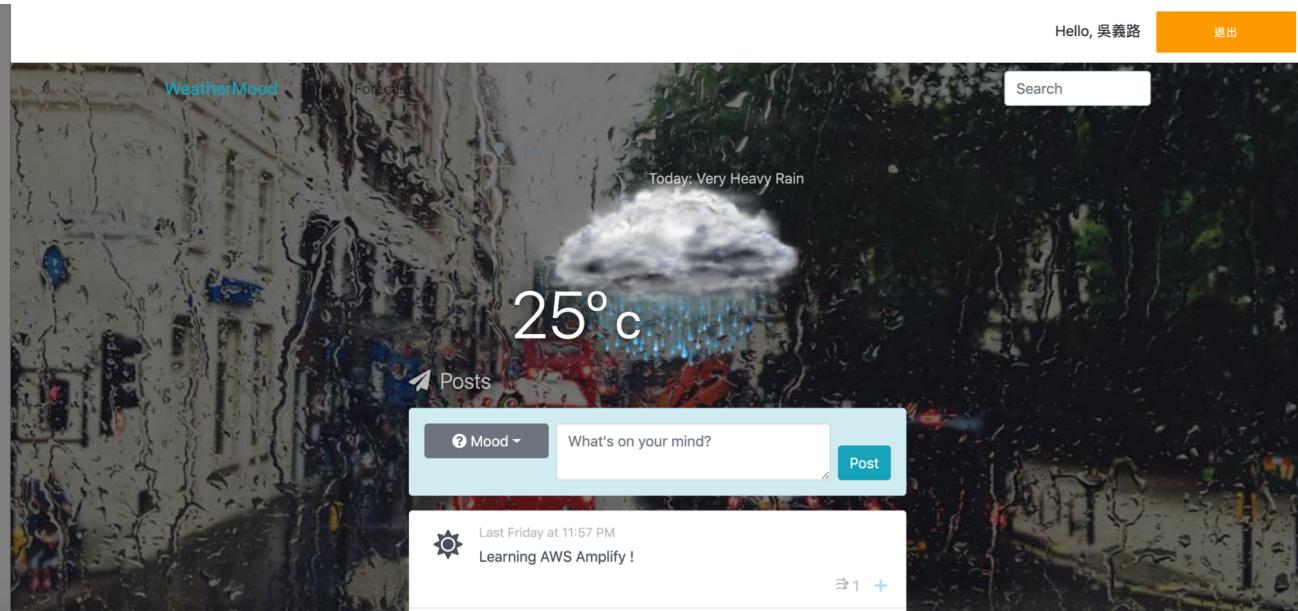
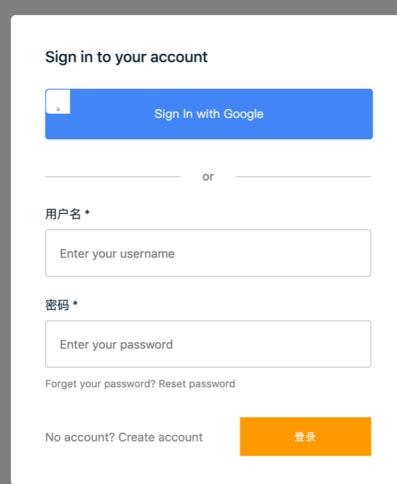
```
const federated = {
  google_client_id: "1060937965775-8f576jlvsk0trg58g1s9t73qhbqismc.apps.googleusercontent.com"
}

export default withAuthenticator(connect(state => ({
  ...state.main,
  searchText: state.searchText,
}))(Main), true, [], federated);
```

- A button with “Sign in with google” should appear

Add signin with google

- A greeting bar appears after signed in



Parameters of the withAuthenticator function

- There are several parameters you can customize in the function withAuthenticator
 - includeGreetings
 - authenticatorComponents
 - federated
 - theme
 - signUpConfig

(Optional)

- The withAuthenticator function is basically a wrapper of the Authenticator component
- Use the Authenticator directly can give you more freedom

```
<Authenticator  
  authState="signIn"  
  authData={CognitoUser | 'username'}  
  onStateChange={(authState) => console.log(authState)}  
  federated={myFederatedConfig}  
  theme={myCustomTheme}  
  hide={[  
    Greetings,  
    SignIn,  
    ConfirmSignIn,  
    RequireNewPassword,  
    SignUp,  
    ConfirmSignUp,  
    VerifyContact,  
    ForgotPassword,  
    TOTPSetup,  
    Loading  
  ]}  
  hideDefault={true}  
  amplifyConfig={myAWSExports}  
  errorMessage={myMessageMap}>  
<Greetings/>  
<SignIn federated={myFederatedConfig}/>  
<ConfirmSignIn/>  
<RequireNewPassword/>  
<SignUp/>  
<ConfirmSignUp/>  
<VerifyContact/>  
<ForgotPassword/>  
<TOTPSetup/>  
<Loading/>  
</Authenticator>
```

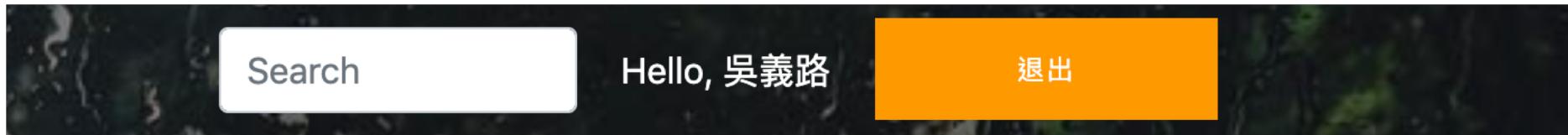
Get authentication data in your component

- `this.props.authState`
 - The current authentication state (e.g. “SignIn”, “SignUp”, “SignedIn”...)
- `this.props.authData`
 - User data

```
▼ CognitoUser {username: "jackraken", pool: CognitoUserPool, Session: null, client: Client, signInUserSession: CognitoUserSession, ...} ⓘ
  Session: null
  ▶ attributes: {sub: "7c5de6e6-a835-415e-b302-c4b20d68c5a0", email_verified: true, phone_number_verified: false, phone_number: "+8860988516137", email: "apple894894@gmail.com"}
  authenticationFlowType: "USER_SRP_AUTH"
  ▶ client: Client {endpoint: "https://cognito-idp.us-east-1.amazonaws.com/", userAgent: "aws-amplify/0.1.x js"}
  keyPrefix: "CognitoIdentityServiceProvider.6452ov8l3tf50b6eh3btd58evv"
  ▶ pool: CognitoUserPool {userPoolId: "us-east-1_yBE03PJeI", clientId: "6452ov8l3tf50b6eh3btd58evv", client: Client, advancedSecurityDataCollectionFlag: true, storage: Storage}
  preferredMFA: "NOMFA"
  ▶ signInUserSession: CognitoUserSession {idToken: CognitoIdToken, refreshToken: CognitoRefreshToken, accessToken: CognitoAccessToken, clockDrift: 0}
  ▶ storage: Storage {todos: "[{"id": "92eb58f4-4e1d-4864-9d6b-6b36aae52c27", "moo...text": "123", "ts": 1557673243, "doneTs": 1557673249}]", CognitoIdentityServiceProvider.6452ov8l3tf50b6eh3btd58evv...}
  userDataKey: "CognitoIdentityServiceProvider.6452ov8l3tf50b6eh3btd58evv.jackraken.userData"
  username: "jackraken"
  ▶ __proto__: Object
```

Create your own greeting component

- Set includeGreetings to false
- Practice: combine authData and SignOut component to create your own greeting component
 - Import { SignOut} from 'aws-amplify-react';



Customize logout button

- Create your own logout button instead of the default one



```
import React from "react";
import { Auth } from 'aws-amplify';
import {Button} from 'reactstrap';

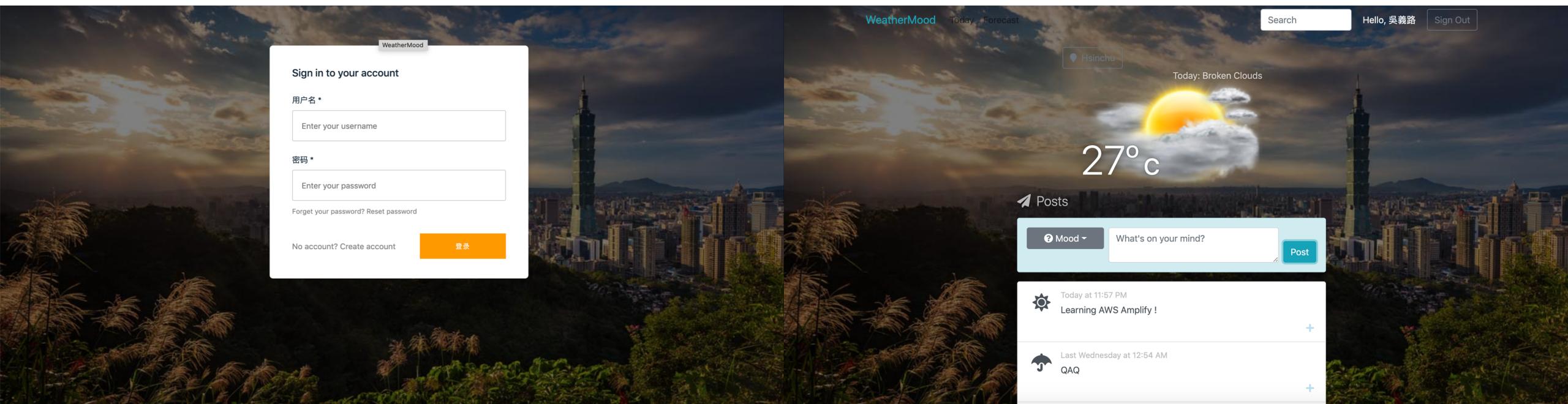
export default class CustomSignOut extends React.Component {
  constructor(props) {
    super(props);
    this.signOut = this.signOut.bind(this);
  }

  signOut() {
    Auth.signOut();
  }
}
```

Congrats!

- Now you have a basic authentication flow in your website
- Make TAs to check your website after completion
- For further study, checkout:

<https://aws-amplify.github.io/docs/js/authentication>



Grading

Lab - Weathermood Auth

In this assignment, you are asked to add authentication to your website.

Requirement

1. (60%) Basic login,logout and signup function.
2. (20%) Signin with google account.
3. (20%) Custom greeting component.

Submit

1. One submission per team.