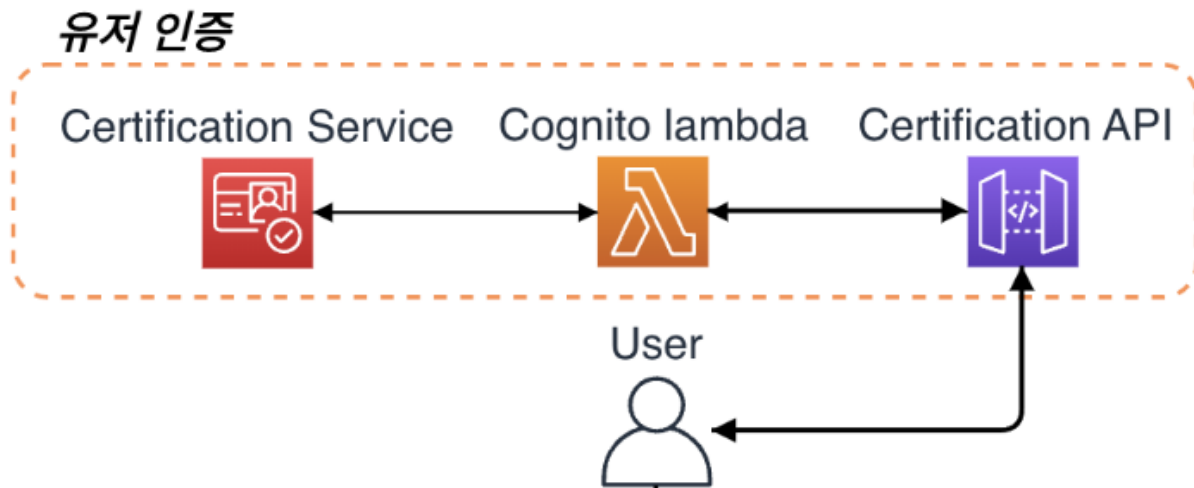


# aws Cognito 발표 자료

👤 생성자	태환 김
🕒 생성 일시	@March 23, 2023 9:16 AM
🏷️ 태그	
👤 최종 편집자	태환 김
🕒 최종 편집 일시	@April 23, 2023 8:11 PM



## AWS Cognito 개요

AWS Cognito는 웹 및 모바일 앱에 대한 사용자 가입, 로그인 및 액세스 제어를 쉽고 빠르게 추가할 수 있는 완전관리형 자격 증명 공급자 서비스입니다.

- 사용자 풀: 사용자 프로필, 인증 설정, 접근 제어 정책을 저장하는 사용자 디렉토리

**도메인 정보**

호스팅 UI 및 OAuth 2.0 엔드포인트에 대한 도메인을 구성합니다. 호스팅 UI 인증 엔드포인트를 생성하기 위해 Cognito가 필요한 경우 도메인을 선택해야 합니다. 기존 도메인이 있는 경우 새 도메인을 할당하기 전에 도메인을 삭제해야 합니다.

작업 ▼

<b>Cognito 도메인</b> 도메인 https://your-domain-prefix.auth.ap-northeast-2.amazoncognito.com	<b>사용자 지정 도메인</b> 도메인 -
---	-------------------------------

Sign in with your username and password

Username

Username

Password

Password

[Forgot your password?](#)

**Sign in**

Need an account? [Sign up](#)

#### 자격 증명 공급자 | 정보

이 앱 클라이언트에 사용할 수 있는 자격 증명 공급자를 선택합니다.

자격 증명 공급자 선택 ▼

**Cognito 사용자 풀** ✕

사용자는 이메일, 전화번호 또는 사용자 이름을 사용하여 Cognito에 로그인할 수 있습니다.

#### OAuth 2.0 권한 부여 유형 | 정보

OAuth 권한 부여 유형을 하나 이상 선택하여 Cognito가 이 앱에 토큰을 전달하는 방법을 구성합니다. 선택한 앱 유형에 따라 제안 옵션을 채웁니다.

OAuth 2.0 권한 부여 유형 선택 ▼

**암시적 권한 부여** ✕

클라이언트가 액세스 토큰(그리고 범위에 따라 선택적으로 ID 토큰)을 직접 가져와야 함을 지정합니다.

**⚠️ 암시적 권한 부여 흐름은 URL에 OAuth 토큰을 노출합니다. 퍼블릭 클라이언트의 경우 PKCE와 함께 권한 부여 코드 흐름만 사용하는 것이 좋습니다.**

사용자 풀을 기본값으로 생성한 뒤, 도메인을 추가.

앱 클라이언트 호스팅 UI 편집에서 권한 부여를 설정.

그러면 로그인화면을 따로 만들지않아도 cognito 사용자 풀 안에 로그인 화면을 포함한 기능이 포함되어있어서 편리합니다. (물론 본인이 만든 프론트엔드를 쓰셔도 무방합니다)

권한 부여 유형으로 보안을 철저하게해서 토큰을 받는 방식도 있지만,

저희는 테스트 겸 암시적 권한 부여로 쉽게 url로 토큰을 발급받는 형식으로 했습니다.

← → ↺ 🔒 t4riua3i23.execute-api.ap-northeast-2.amazonaws.com/cognito/token#id\_token=eyJraWQiOiJrZmxHQzF

```
{ "message" : "Unauthorized" }
```

로그인 화면에서 사용자 로그인을 하면 URL에서 토큰을 발급받습니다.

## 통합

통합을 경로에 연결 | 통합 관리

### cog-api에 대한 경로

▼ /cognito

▼ /token

GET AWS Lambda

### 경로에 대한 통합 세부 정보

GET /cognito/token (cozzt78)

Lambda 함수

cognito\_token (ap-northeast-2) [🔗](#)

설명

-

HTTP API를 기본값으로 생성 후, GET 경로를 지정하고(예:"GET /cognito/token") cognito관련 Lambda와 통합 연결을 합니다  
(원래는 발급받은 토큰을 사용자가 애플리케이션을 통해 토큰을 넣고 인증하는 방식이지만, 저희가 그것까진 구현을 하는 역할이 아니기때문에 엔드포인트에 토큰을 넣었을 때 응답을 받으면 되는 방식으로 구현을 하였습니다)

## 권한 부여

경로에 권한 부여자 연결 | 권한 부여자 관리

### cog-api에 대한 경로

▼ /cognito

▼ /token

GET JWT 인증

### GET /cognito/token 경로의 권한 부여자

권한 부여자 이름

권한 부여자 유형

권한 부여자 ID

cog-jwt	JWT	4r7umx
---------	-----	--------

자격 증명 소스

이 권한 부여자가 호출되면 API Gateway는 이 선택 표현식을 사용하여 토큰의 소스를 결정합니다.

**`$request.querystring.access_token`**

발급자

자격 증명 공급자와 발행자 URL

`https://cognito-idp.ap-northeast-2.amazonaws.com/ap-northeast-2_XXXXXX`

대상

이 권한 부여자와 연결된 대상

- Amazon Cognito Identity Provider

권한 부여를 JWT로 생성하고 GET 경로에 연결합니다.

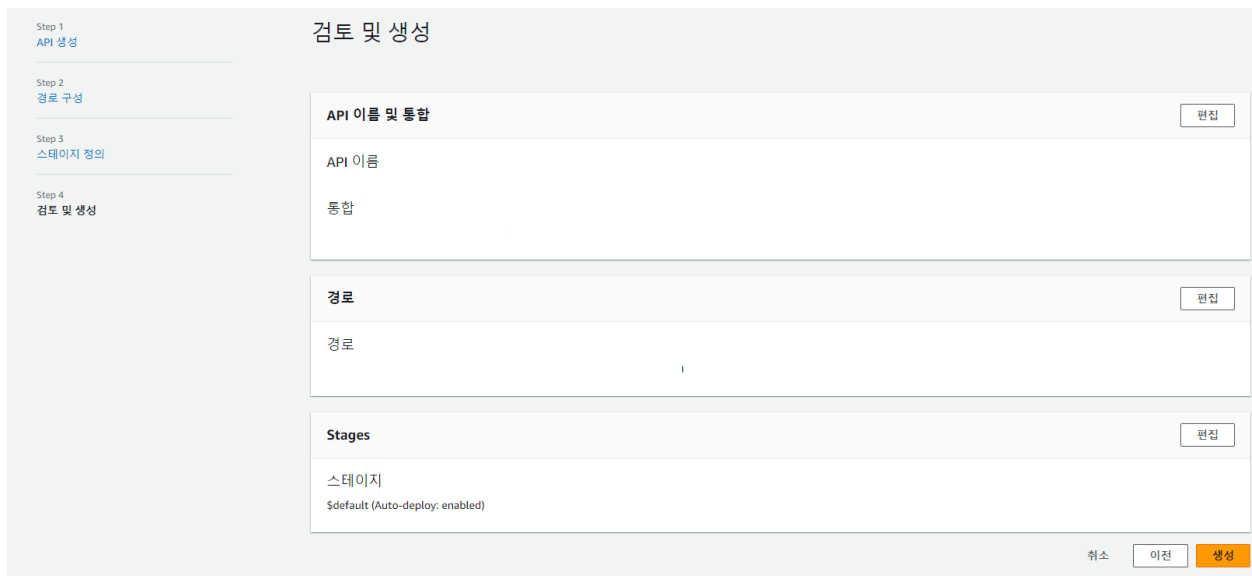
### 자격 증명 소스

토큰의 소스를 정의하는 선택 표현식을 입력하십시오.

```
$request.querystring.access_token
```

자격 증명 소스에서 "\$request.querystring.access\_token" 라는 부분은 따로 매개변수 매핑을 생성하지 않아도 되고 URL에서 "access\_token" 쿼리 매개변수 값을 추출하는 데 사용되는 코드의 변수 또는 매개변수를 나타냅니다.

(<https://www.youtube.com/watch?v=uSwXzdGOubg&t=171s>)



Step 1  
API 생성

Step 2  
경로 구성

Step 3  
스테이지 정의

Step 4  
검토 및 생성

### 검토 및 생성

**API 이름 및 통합** 편집

API 이름  
통합

**경로** 편집

경로  
경로

**Stages** 편집

스테이지  
\$default (Auto-deploy: enabled)

취소 이전 생성

HTTP API로 생성한 이유는 REST API에 비해 통합, 경로 지정, 자동배포가 쉽고 편리하게 간소화되었기때문입니다.

통합 연결과 권한 부여 연결이 끝났으면

## 경로

Lambda

>

함수

>

cognito\_token

cognito\_token

함수 개요 정보

코드

테스트

모니터링

구성

별칭

버전

일반 구성

트리거

권한

대상

함수 URL

환경 변수

태그

트리거 (1) 정보

Q

트리거 찾기

트리거

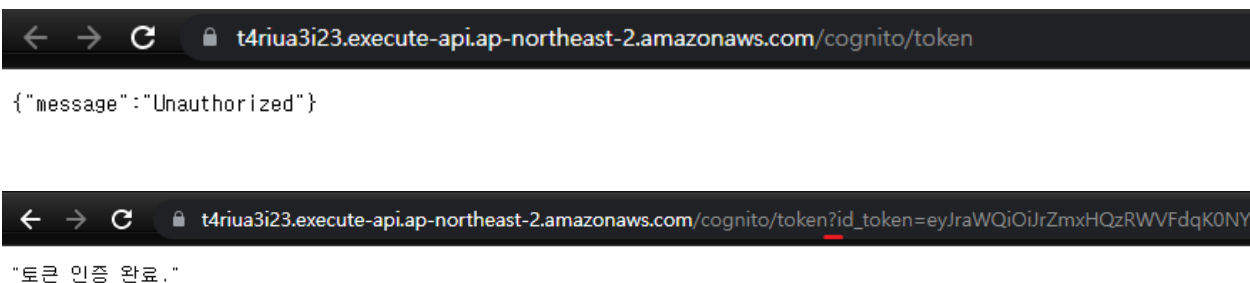
API 게이트웨이: cog-api

arn:aws:execute-api:ap-northeast-2:180693256225:t4riua3i23/\*/\*/cognito/token

API 엔드포인트: <https://t4riua3i23.execute-api.ap-northeast-2.amazonaws.com/cognito/token>

세부 정보

생성한 기본 람다 함수가 api 게이트웨이에 트리거가 추가된 것을 확인할 수 있습니다.



엔드포인트에 접속 후 발급받은 토큰을 GET /cognito/token?access\_token~  
넣으면 URL에 access 토큰을 쿼리 매개변수로 받아서 200 ok로 응답 받을 수 있습니다.  
이제 발급받은 토큰이 사실임을 확인했으니, access 토큰을 가지고 출석 요청 및 구매 조회를  
하면 됩니다.