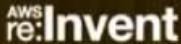


November 28, 2017

AWS re:INVENT

Serverless Authentication and Authorization

Justin Pirtle and Vladimir Budilov, Senior Solutions Architects



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Many serverless applications need a way to manage end user identities and support sign-ups and sign-ins. Join this session to learn real-world design patterns for implementing authentication and authorization for your serverless application—such as how to integrate with social identity providers (such as Google and Facebook) and existing corporate directories. We cover how to use Amazon Cognito identity pools and user pools with API Gateway, Lambda, and IAM.

What to expect from the session

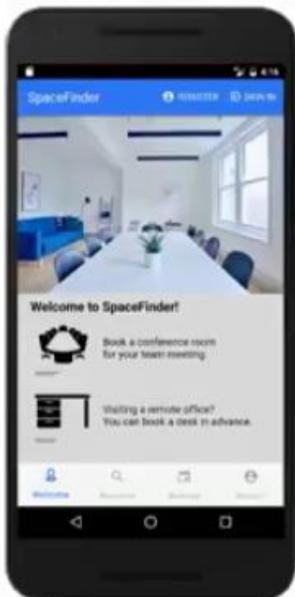
- Assumes high-level familiarity with Serverless API architectures (API Gateway, Lambda)
- Learn how to implement **identity management** for your **serverless apps**, using
 - Amazon Cognito User Pools
 - Amazon Cognito Federated Identities
 - Amazon API Gateway
 - AWS Lambda
 - AWS Identity and Access Management (IAM)



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



SpaceFinder



AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Hybrid mobile app

- Runs in web browser, Android, Apple iOS devices
- Built using Ionic 3 Framework
- Angular 4 / TypeScript
- AWS SDKs for JavaScript

Do try this at home

- Mobile app + API are open-sourced (Apache 2.0 license)
- <https://github.com/awslabs/aws-serverless-auth-reference-app>

This is a conference room finder application

GitHub, Inc. [US] | https://github.com/awslabs/aws-serverless-auth-reference-app

This repository Search Pull requests Issues Marketplace Explore

awslabs / aws-serverless-auth-reference-app

Watch 82 Star 371 Fork 88

Code Issues 10 Pull requests 0 Projects 0 Insights

Serverless reference app and backend API, showcasing authentication and authorization patterns using Amazon Cognito, Amazon API Gateway, AWS Lambda, and AWS IAM.

amazon-cognito aws-cognito serverless iam serverless-architectures aws-lambda cognito cognito-quickstart authentication authorization auth amazon-api-gateway aws aws-apigateway

143 commits 1 branch 0 releases 8 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

justonian Merge branch 'master' of github.com:awslabs/aws-serverless-auth-refer... Latest commit 88c74a2 on Dec 5, 2017

api Removed external IdP token attributes
app Removed external IdP token attributes
.editorconfig Adding git ignore and updated README
.gitignore fixing comments from pull request
DevGuide.md Upgraded Android Cordova, added platform check, and hosted
Dockerfile Updated Dockerfile for shallow git clone
LICENSE Adding git ignore and updated README
Quickstart.md Updated Quickstart guide to reference latest Docker image and
README.md Updated Readme with re:Invent 2017 presentation video refer...

```
MINGW64:/c/Users/Elite8300/Documents/DevBranch
$ cd 'C:\Users\Elite8300\Documents\DevBranch'
$ git clone https://github.com/awslabs/aws-serverless-auth-reference-app.git
Cloning into 'aws-serverless-auth-reference-app'...
remote: Counting objects: 1218, done.
remote: Total 1218 (delta 0), reused 0 (delta 0), pack-reused 1218
Receiving objects: 100% (1218/1218), 3.20 MiB | 2.64 MiB/s, done.
Resolving deltas: 100% (662/662), done.
$
```

Managing Identities



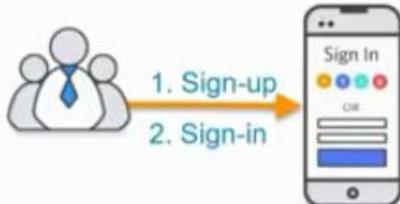
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Imagine you have to create an app that allows users to sign up, sign in, upload their profile image, and search and book conference rooms. This requires managing users' identities.

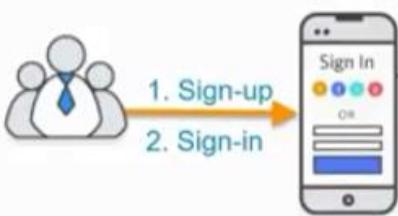
Sign-up and Sign-in



Username	Email	Password
beverly123	beverly123@example.com	Password\$123
pilotjane	pilotjane@example.com	a##eroplans3
sudhir1977	sudhir197@example.com	mmd414997a

You might create the above

Sign-up and Sign-in



Username	Email	Password
beverly123	beverly123@example.com	P@ssw0rd
pilotjane	pilotjane@example.com	a##eroplan
sudhir1977	sudhir197@example.com	P@ssw0rd



- Never store passwords in plaintext!
 - Vulnerable to rogue employees
 - A hacked DB results in all passwords being compromised

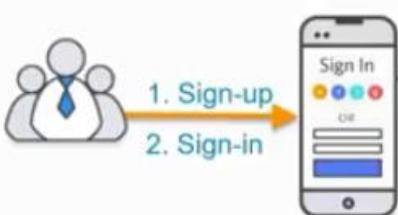
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



This is an antipattern

Sign-up and Sign-in



Username	Email	Hashed Password
beverly123	beverly123@example.com	2fa...4715efc...11
pilotjane	pilotjane@example.com	fea74fde862...90ae883
sudhir1977	sudhir197@example.com	c775cc9b3dbe4c...



- MD5/SHA1 collisions
 - Rainbow Tables
 - Dictionary attacks, brute-force (GPUs can compute billions of hashes/sec)

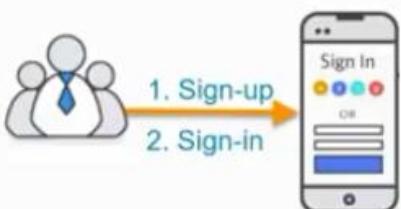
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



This is also a bad approach

Sign-up and Sign-in



Username	Email	Salted Hash
beverly123	beverly123@example.com	1e66f9358530620b2bcae79dada717c...
pilotjane	pilotjane@example.com	88fccd9cf82377d11d2fede177457d47...
sudhir1977	sudhir197@example.com	08a5981de4fecf04b1359a179962a48...



- Incorporate app-specific salt + random user-specific salt
- Use algorithm with configurable # of iterations (e.g. bcrypt, PBKDF2), to slow down brute force attacks

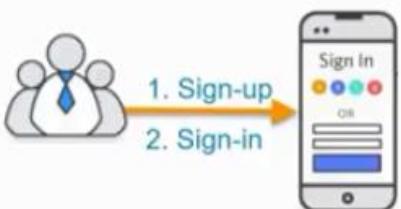
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



A good approach is to salt hash the passwords but you might also not want to send the salted password over the wire for authentication

Sign-up and Sign-in



Username	Email	SRP Verifier function
beverly123	beverly123@example.com	<password-specific verifier>
pilotjane	pilotjane@example.com	<password-specific verifier>
sudhir1977	sudhir197@example.com	<password-specific verifier>



- Secure Remote Password (SRP) Protocol
- Verifier-based protocol
- Passwords never travel over the wire
- Resistant to several attack vectors
- Perfect Forward Secrecy

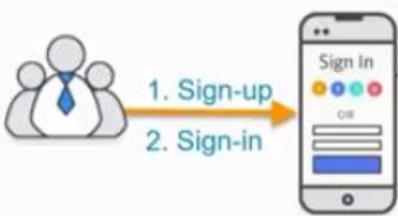
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



SRP allows you to transfer a verifier code created from the password and sent that over the wire instead.

Sign-up and Sign-in



Username	Email	SRP Verifier function
beverly123	beverly123@example.com	<password-specific verifier>
pilotjane	pilotjane@example.com	<password-specific verifier>
sudhir1977	sudhir197@example.com	<password-specific verifier>

User Flows

- Registration
- Verify email/phone
- Secure sign-in
- Forgot password
- Change password
- Sign-out

Security Requirements

- Secure password handling
- Multi-Factor Authentication
- Enforce password policies
- Encrypt all data server-side
- Support custom authentication flows
- Scalable to 100s of millions of users

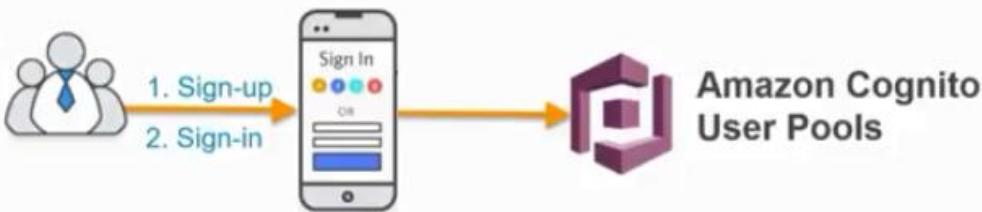
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



What about the other factors listed above

Sign-up and Sign-in



User Flows

- Registration
- Verify email/phone
- Secure sign-in
- Forgot password
- Change password
- Sign-out

Security Requirements

- Secure password handling
- Multi-Factor Authentication
- Enforce password policies
- Encrypt all data server-side
- Support custom authentication flows
- Scalable to 100s of millions of users

AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Sign-up and Sign-in



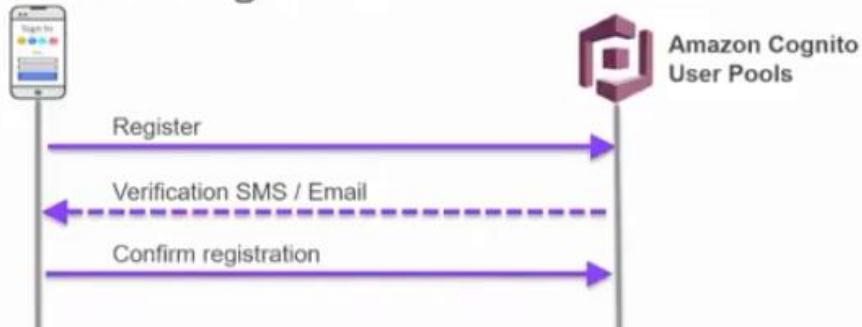
To use Cognito User Pools using our SDK, you send the users details for registration

Sign-up and Sign-in



You can also optionally set up the requirements to validate either the customers email or their SMS built into the service.

Sign-up and Sign-in



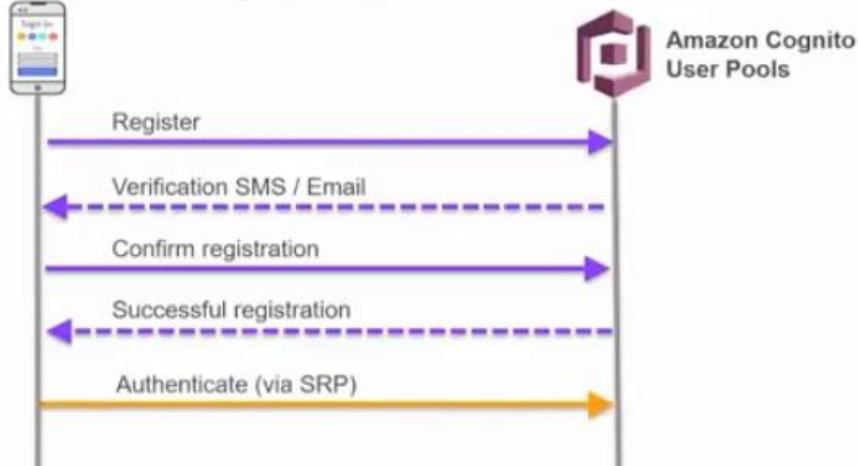
The user then has to use the code they got via email or SMS to confirm that they indeed got it

Sign-up and Sign-in



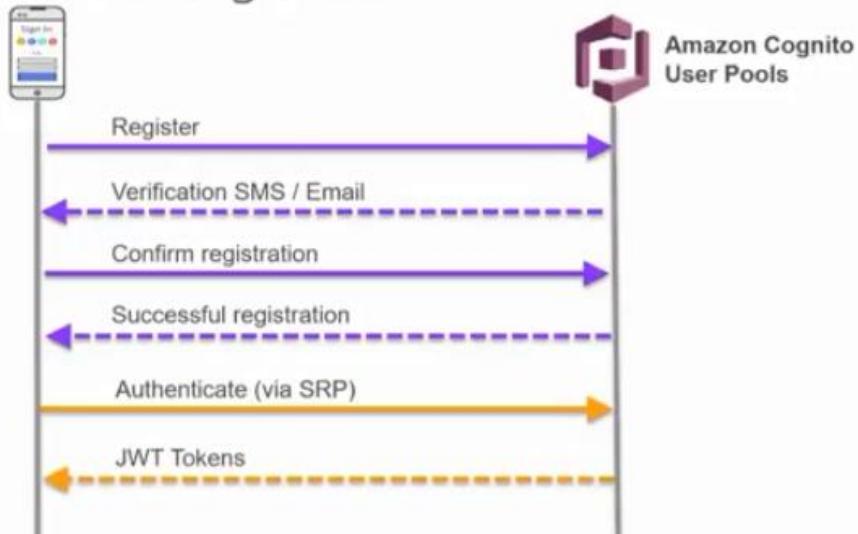
Then you have a successfully registered user that can actually authenticate with your service.

Sign-up and Sign-in



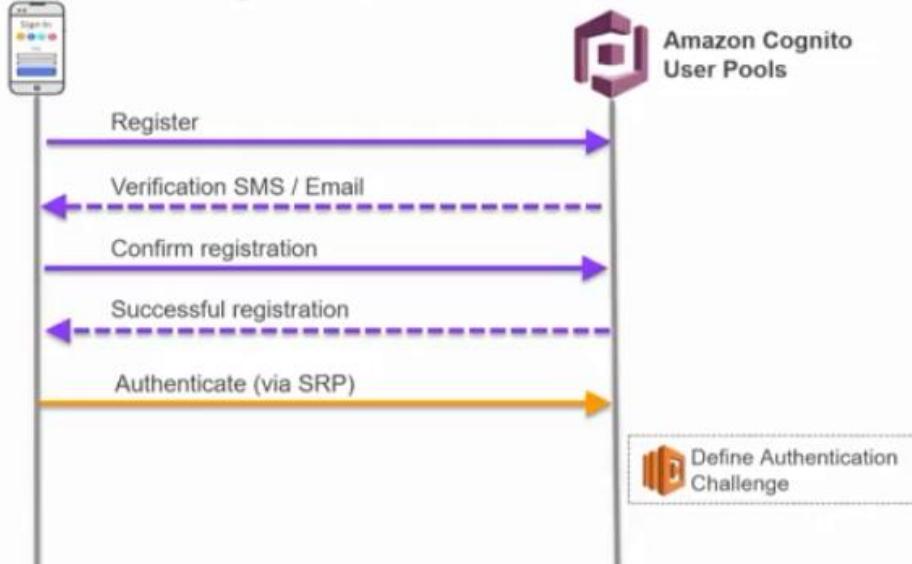
Once the user goes to authenticate during login, this is done using SRP

Sign-up and Sign-in



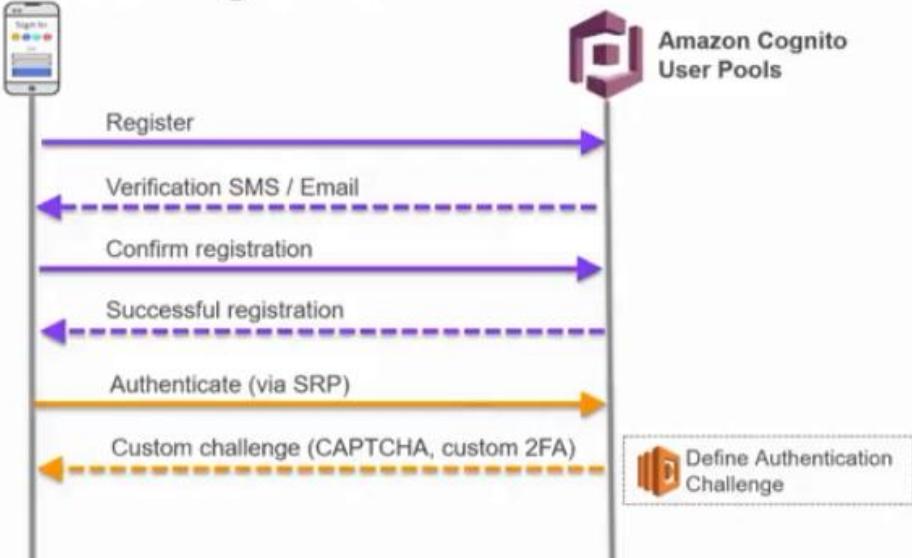
After a successful authentication, the cognito user pools service returns to you a set of JWT tokens.

Sign-up and Sign-in



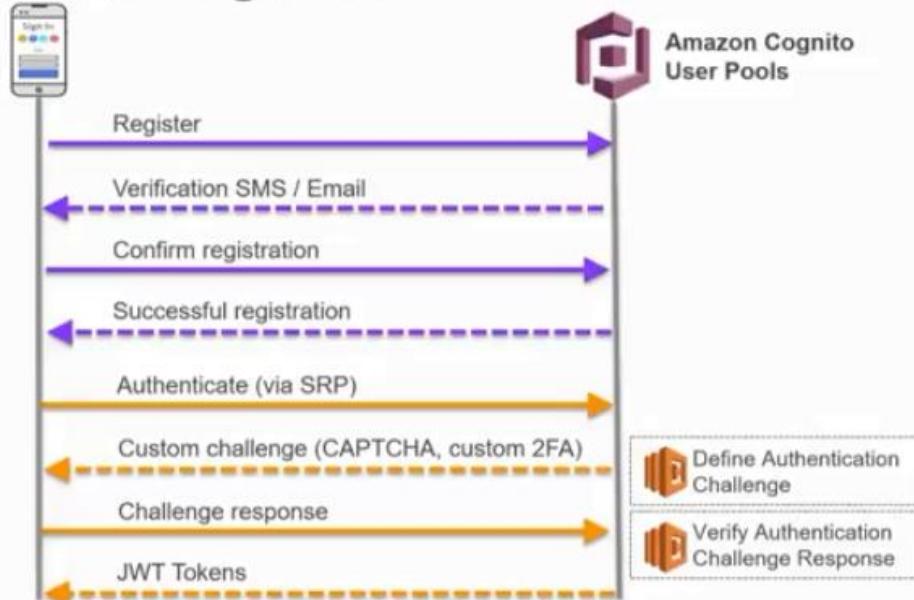
But you can also hook in and customize this authentication flow to suit your need like needing MFA option or using a CAPTHA.

Sign-up and Sign-in



You can define a custom challenge or trigger that you can hook into the workflow to generate that challenge for your users and send that challenge result back with a lambda function used to verify the challenge response.

Sign-up and Sign-in



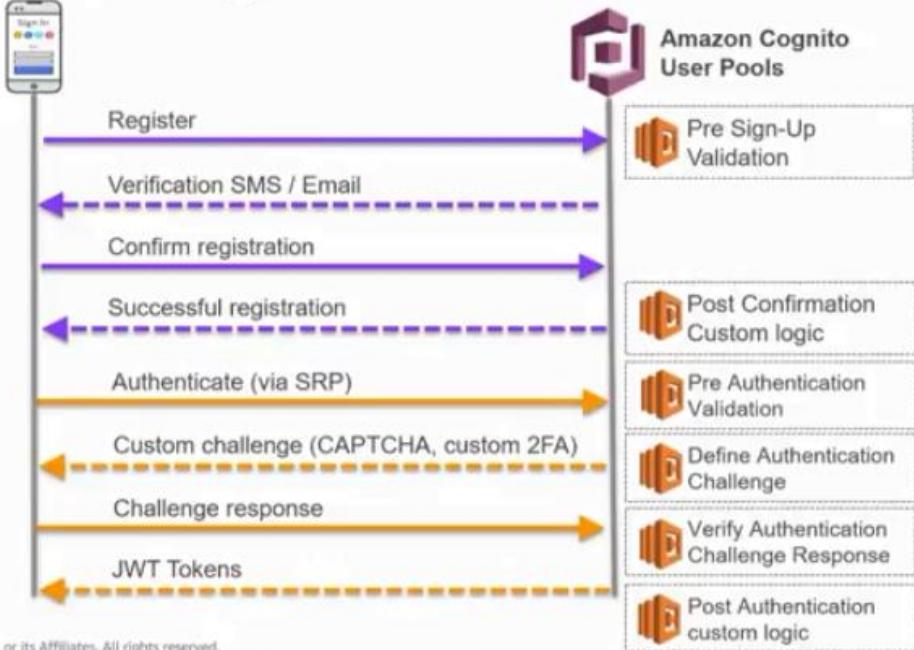
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Then you get the JWT back just as before

Sign-up and Sign-in



AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



These are some of the available lambdas that you can hook in to customize the authentication flow.

Sign-up and Sign-in



JWT token

```
eyJraWQiOiI5ZXJydERlbHRxOF13YUp5MkdadE9ieWtSREVB  
OVNCNGlEVdz2V2IUzVFFPSIsImFe2yI61lJTMjU2In0.eyJz  
dWIiOii2ZjU1NzM2OC1hODg0LTQ4NGUtYjY2M105Zm2oWYz  
YzM4MDIiLCJhdWQiOiI2bGtmczcwcwcm92a3ViaxJoMXF0bnR2  
ajAxMiIsImVtYWlsX3ZlcmimaWVkJp0cnVlJCJ0b2t1b191  
c2UiOiJpZCIsImF1dGhfdGltZSI6MTQ3ODQ0OTAzMCwiaXNz  
IjoiaHR0cHM6XC9cL2NvZ25pdG8taWRwLnVzLWVhc3QtMS5h  
bWF6b25hd3MuY29tXC91cy1lYXN0LTFFWE1sVVc5c1V5Iiwi  
Y29nbml0bzplc2VybmtzZSI6InRlc3QxMjMiLCJleHAIoJE0  
Nzg0NTI2NjAsImdpdmVuX25hbWUiOiJUZXN0IiwiawF0Ijox  
NDc4NDQ5MDYwLCJmYW1pbHfbmFtZSI6IIRlc3QiLCJ1bWFp  
bC16InRyYW5qaW1AYW1hem9uLmNvbSJ9.atQ00Sjg9V97d6t  
YonHNx0q7Zuof8-d-q0u69zNnuSJtmzGvOA97tP2e3GydY9  
K8q_2kG2izkpEMUEdaeWjz2qG5dS328Scm6pRDpC5pOkU8y  
mjH7DBPFVXhtgS3iOhyleFhtmaTaYb_1YLpaaV10m8sVFQMH  
t_jdfrAm26Fq7zyjWYTsfzhqud29Ti4zn9PhcE7aL3s7BB8CJ  
18_yFXSo5CYCpLszvHaxxlcbmPoXfr1F1PvZ07Oy8EbOaGs  
4CuKmoYiV-5RnZsA9JXj405Kp50k-v8HCl6ZACDw3OYMV87P  
e6PuEqbzQLlc8BufKThm0xBiO6NJtvI7ic2sEIQ
```

JWT token

```
eyJraWQiOii5ZXJydERLbHRxOF13YUp5MkdadE9ieWtSREVB  
OVNCNG1EVDBZ2V21UZVFFPS1sImFsZyI6IlJTMjU2In0...  
dW1iOii1Z2zJ0In2A0cC1n0bDg1Lc4nG0Cjz2H1032mz0WYz  
YzM4MDi1LCJhdWQiOiI2bGtmczcwm92a3ViaXJoMXF0bnR2  
ajAxMiIsImVtYWIaX31lcm1maWVkijp0cnVlLCJ0b2t1b191  
c2UiOiJpzc1sImF1dGhfdGltZSI6MTQ3ODQ0OTA2MCwiaXNz  
IjoiaHR0cHM6XC9cL2NuZ25pdG8taWRwLnVzLWVhc3QtMS5h  
bWF6b25hd3MuY29tXC91cy1lYXN0LTFFWE1sVVC5c1V5Iiwi  
Y29nbm10bzplc2VybmtFZSI6InRlc3QxMjILCJ1eHAiOjE0  
Nzg0NTI2NjAsImdpdmVuX25hbWUoiJUZXN0IiwiawF0Ijox  
NDc4NDQ5MDYwLCJmYWIpbHfbmFtZSI6IlRlc3QiLCJlbWFp  
bC161nRyYWSquW1AYW1hem9uLmNvbSJ9..atQ008Jq9V97d6t  
YonHNx0q7ZuoI8-d-q0u69zNnu5JtmzGv0AN97tP2e3Gydy9  
Khq_2kG21zkpEMUEdaewjzzgG0d33283cm6pR0DpC5pOKUBy  
mjH7DBPFVXhtqS3iOhyleFhtmaTaYb_1YLpaav10m8sVFOMB  
t_jdfAm26Fq7zyjWYT5fzhqud29T14zn9PhcE7al3s7BB0CJ  
18_yFXBoG5CYCpLazvhax1lcbmPoXFr1FIPvZ070y8Eb0aGh  
4CuKmoY1V-5RnZaA9JXj405Kp50k-v8HCL6ZACDw3OYMV87P  
e6FuEqb2Q11c0BuFkThm0xB106Njtvi7ic2aETQ
```

Header

```
{  
  "kid": "9errtDKltq8YwaJy2GZtObYKRDEA9SB4iDT6vNmTeQE=",  
  "alg": "RS256"  
}
```

JWT token

```
eyJraWQiOii5ZXJydERLbHRxOF13YUp5MkdadE9ieWtSREVB  
OVNCNG1EVDBZ2V21UZVFFPS1sImFsZyI6IlJTMjU2In0...  
dW1iOii1Z2zJ0In2A0cC1n0bDg1Lc4nG0Cjz2H1032mz0WYz  
YzM4MDi1LCJhdWQiOiI2bGtmczcwm92a3ViaXJoMXF0bnR2  
ajAxMiIsImVtYWIaX31lcm1maWVkijp0cnVlLCJ0b2t1b191  
c2UiOiJpzc1sImF1dGhfdGltZSI6MTQ3ODQ0OTA2MCwiaXNz  
IjoiaHR0cHM6XC9cL2NuZ25pdG8taWRwLnVzLWVhc3QtMS5h  
bWF6b25hd3MuY29tXC91cy1lYXN0LTFFWE1sVVC5c1V5Iiwi  
Y29nbm10bzplc2VybmtFZSI6InRlc3QxMjILCJ1eHAiOjE0  
Nzg0NTI2NjAsImdpdmVuX25hbWUoiJUZXN0IiwiawF0Ijox  
NDc4NDQ5MDYwLCJmYWIpbHfbmFtZSI6IlRlc3QiLCJlbWFp  
bC161nRyYWSquW1AYW1hem9uLmNvbSJ9..atQ008Jq9V97d6t  
YonHNx0q7ZuoI8-d-q0u69zNnu5JtmzGv0AN97tP2e3Gydy9  
Khq_2kG21zkpEMUEdaewjzzgG0d33283cm6pR0DpC5pOKUBy  
mjH7DBPFVXhtqS3iOhyleFhtmaTaYb_1YLpaav10m8sVFOMB  
t_jdfAm26Fq7zyjWYT5fzhqud29T14zn9PhcE7al3s7BB0CJ  
18_yFXBoG5CYCpLazvhax1lcbmPoXFr1FIPvZ070y8Eb0aGh  
4CuKmoY1V-5RnZaA9JXj405Kp50k-v8HCL6ZACDw3OYMV87P  
e6FuEqb2Q11c0BuFkThm0xB106Njtvi7ic2aETQ
```

Payload

```
{  
  "sub": "6f557368-a884-484e-b662-9fc69f3c3802",  
  "aud": "61kfs70rovkubirh1qtntvj012",  
  "email_verified": true,  
  "token_use": "id",  
  "auth_time": 1478449060,  
  "iss": "https://cognito-idp.us-east-1.amazonaws.com  
    \us-east-1_XMlUW9sUy",  
  "cognito:username": "test123",  
  "exp": 1478452660,  
  "given_name": "Test",  
  "iat": 1478449060,  
  "family_name": "Test",  
  "email": "test@example.com"  
}
```

JWT token

```
eyJraWQ1oII5ZXJydERLbHRxOFI3YUp5MkdadE9ieWtSREVB  
OVNCNG1EV0Z2V21UzVFFPSIsImFsZyI6I1JTMjU2In0.eyJz  
dWIoII2zJU1NzM2OC1hODg0LTQ4NGUyYjY2M105ZmM20WYz  
YzM4MDI1LCJhdWQiOiI2bGtmczcwcwm92a3ViAxJcMXF0bnR2  
ajAxMiIisImVtYwlsX3Z1cmimaWVkJp0cnVlC70b2t1b191  
c2U1OjIpZCisImF1dGhfdGltZSI6MTQ3ODQ00TA2MCwiAxH2  
IjoiaHR0cHM6XC9cL2NvZ25pdGtaWRwLnVzLWVhc3QtMS5h  
bWF6b25hd3MuY29tXC91cy1lYXN0LTFFWElsVVc5c1V5Iiwi  
Y29nbml0bzp1c2VybmdFtZSI6InRlc3QxMjM1LCJleHAIoE0  
Nzg0NTI2NjAsImddpmVuX25hbWUoIjUZXN0IiwiWF0Ijox  
NDc4NDQ5MDYwLCJmYW1pbH1fbmtZSI6I1Rlc3Q1LCJlbWFp  
bC16InRyYW5qaW1AYW1hem9uLmNvbSJ9.eyJ0eXA4Nj97d6t  
YonHNx0q7Zuof8-d-q0u69zNnuSjTmzGvOA97tP2e3GydY9  
K8q_2kG2IzkpEMUEdaeWjz2qG5dS328Scm6pRDPPc5pOkU8y  
mjH7DBPfVXhtgS3iOhyleFhtmaTaYb_1YLpaaV10m8sVFOMH  
tjdfAm26Fq7zyjWYTSfhqud29Ti4zn9PhcE7aL3s7BB8CJ  
18_yFXSoG5CYCplszvHaxz1cbmPoXFrlF1PvZ070y8Eb0aGs  
4CukmoYiV-5RnZsA9JXj405Kp50k-v8HCL6ZACDw3OYMV87P  
e6PuEqbzQLlc8BufKThm0xBi06Njtvi7ic2sEIQ
```



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Signature

```
HMACSHA256(base64UrlEncode(header) + "." +  
base64UrlEncode(payload), {secret});
```

JWT token

```
eyJraWQ1oII5ZXJydERLbHRxOFI3YUp5MkdadE9ieWtSREVB  
OVNCNG1EV0Z2V21UzVFFPSIsImFsZyI6I1JTMjU2In0.eyJz  
dWIoII2zJU1NzM2OC1hODg0LTQ4NGUyYjY2M105ZmM20WYz  
YzM4MDI1LCJhdWQiOiI2bGtmczcwcwm92a3ViAxJcMXF0bnR2  
ajAxMiIisImVtYwlsX3Z1cmimaWVkJp0cnVlC70b2t1b191  
c2U1OjIpZCisImF1dGhfdGltZSI6MTQ3ODQ00TA2MCwiAxH2  
IjoiaHR0cHM6XC9cL2NvZ25pdGtaWRwLnVzLWVhc3QtMS5h  
bWF6b25hd3MuY29tXC91cy1lYXN0LTFFWElsVVc5c1V5Iiwi  
Y29nbml0bzp1c2VybmdFtZSI6InRlc3QxMjM1LCJleHAIoE0  
Nzg0NTI2NjAsImddpmVuX25hbWUoIjUZXN0IiwiWF0Ijox  
NDc4NDQ5MDYwLCJmYW1pbH1fbmtZSI6I1Rlc3Q1LCJlbWFp  
bC16InRyYW5qaW1AYW1hem9uLmNvbSJ9.eyJ0eXA4Nj97d6t  
YonHNx0q7Zuof8-d-q0u69zNnuSjTmzGvOA97tP2e3GydY9  
K8q_2kG2IzkpEMUEdaeWjz2qG5dS328Scm6pRDPPc5pOkU8y  
mjH7DBPfVXhtgS3iOhyleFhtmaTaYb_1YLpaaV10m8sVFOMH  
tjdfAm26Fq7zyjWYTSfhqud29Ti4zn9PhcE7aL3s7BB8CJ  
18_yFXSoG5CYCplszvHaxz1cbmPoXFrlF1PvZ070y8Eb0aGs  
4CukmoYiV-5RnZsA9JXj405Kp50k-v8HCL6ZACDw3OYMV87P  
e6PuEqbzQLlc8BufKThm0xBi06Njtvi7ic2sEIQ
```



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Header

```
{  
  "kid": "9errtDKltq8YwaJy2GZtObykRDEA9SB4iDT6vNmTeQE=",  
  "alg": "RS256"  
}
```

Payload

```
{  
  "sub": "6f557368-a884-484e-b662-9fc69f3c3802",  
  "aud": "61kfa70rovkubirhlqtntvj012",  
  "email_verified": true,  
  "token_use": "id",  
  "auth_time": 1478449060,  
  "iss": "https://cognito-idp.us-east-1.amazonaws.com  
  \us-east-1_XM1UW9sUy",  
  "cognito:username": "test123",  
  "exp": 1478452660,  
  "given_name": "Test",  
  "iat": 1478449060,  
  "family_name": "Test",  
  "email": "test@example.com"  
}
```

Signature

```
HMACSHA256(base64UrlEncode(header) + "." +  
base64UrlEncode(payload), {secret});
```

A JWT is a form of base64 encoded blob of text as an identity token with all the user details. There are 3 types of tokens you can get from the Cognito User Pools service, an Access token, Identity token, and a Refresh token. The access tokens are used to grant programmatic or API access to the user, the identity token shown above can be used for downstream or sudo-authentication for reading the claims of the user for dynamically changing the user's experience in your application using the token. The access and identity tokens are only good for 1 hour from the time they are issued, the refresh tokens can be configured to be valid for as long as 30 days. You can use a refresh token to get a new access and identity token.

Application so far...



Now we have a way for our app where a user can sign in and sign out. What if they need to access AWS resources like S3 or DynamoDB.

Application so far...



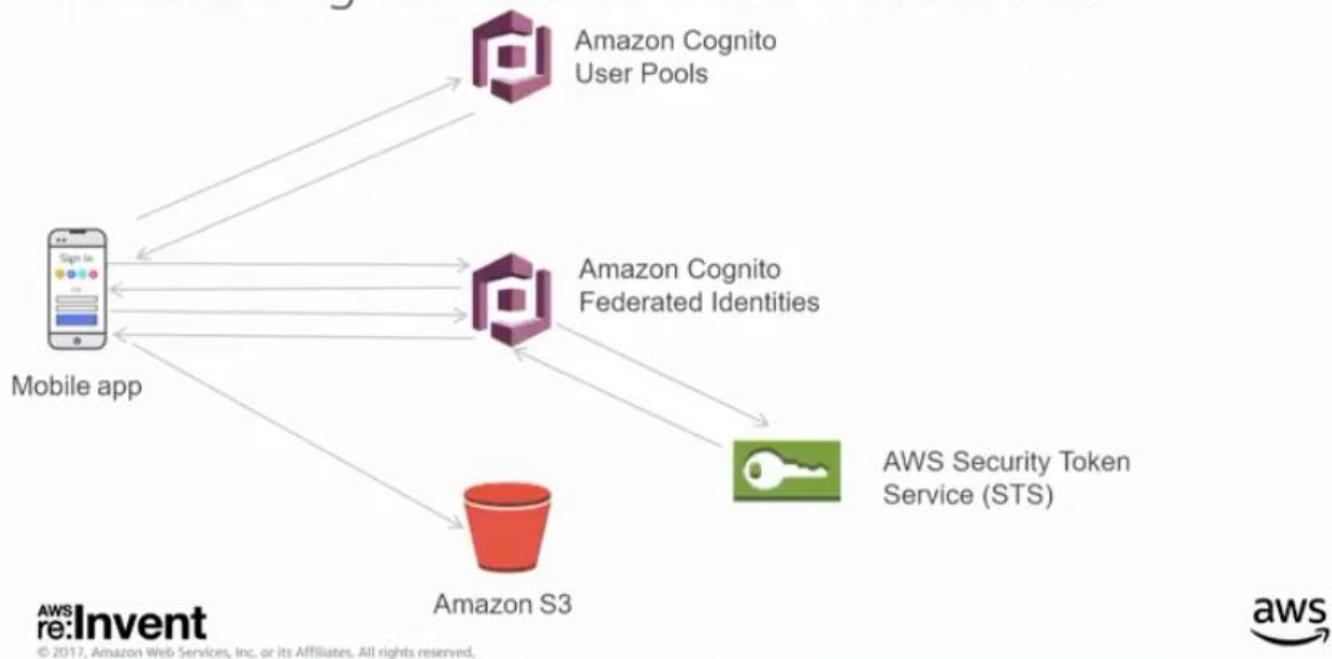
To give the user a secure, fine grain access to S3 we need something else.

Application so far...

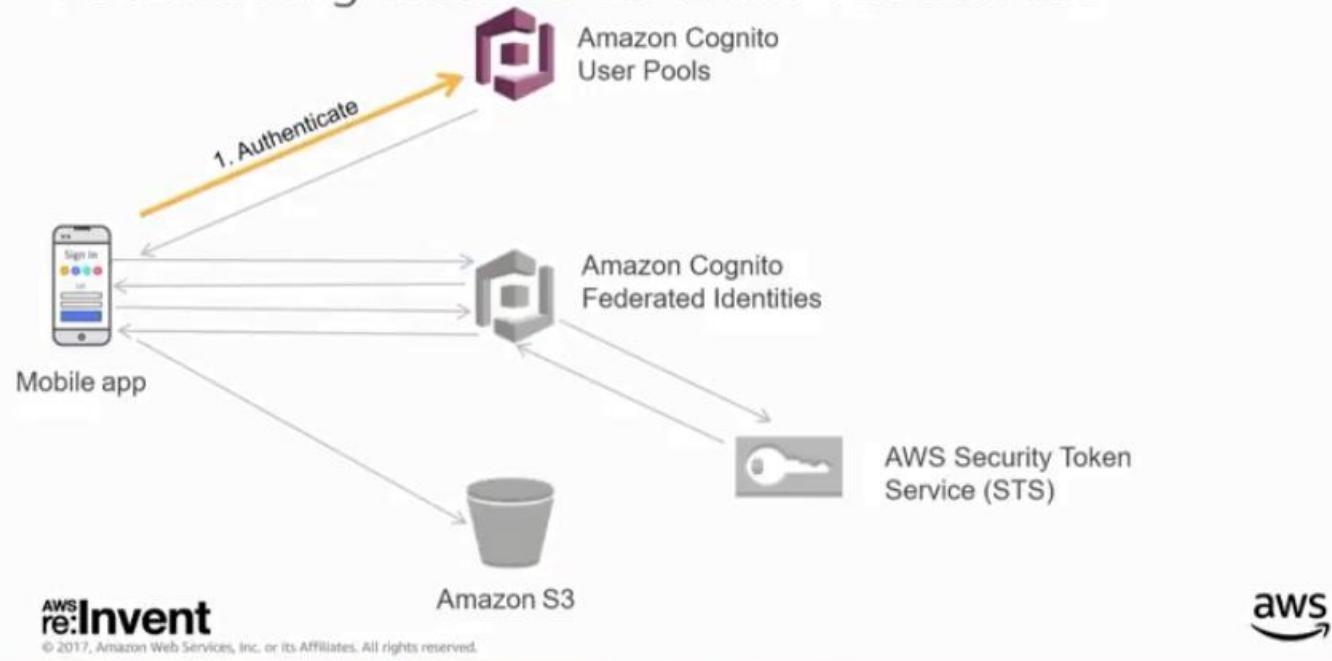


The Amazon Cognito Federated Identities (CFI) allows you to exchange tokens from user pools or other bearer token based providers for AWS native credentials in the form of IAM Role.

Federating access to AWS resources

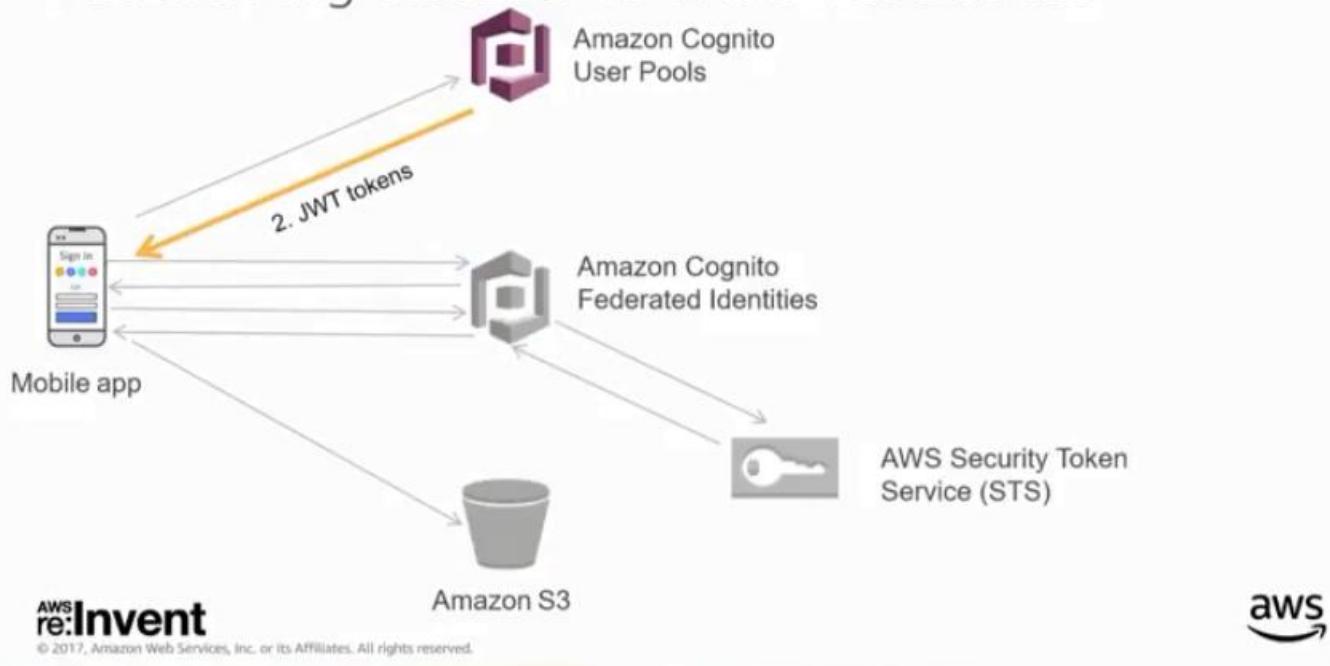


Federating access to AWS resources



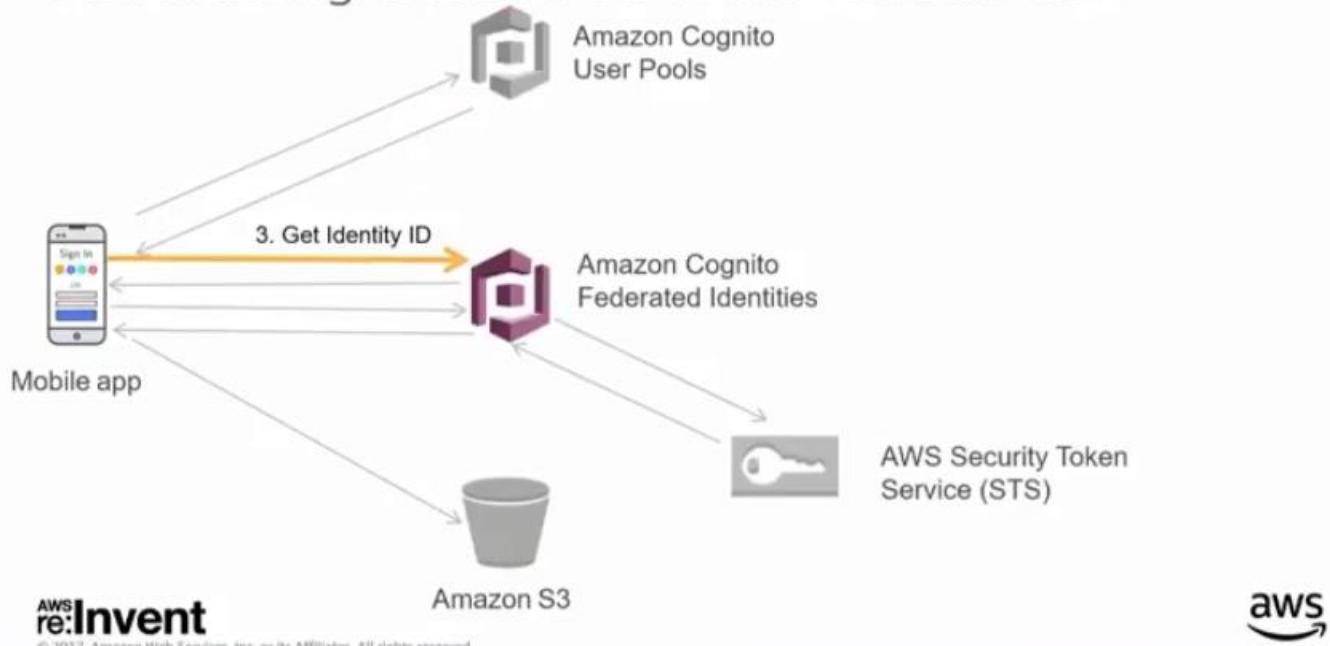
First you authenticate with Cognito User Pools

Federating access to AWS resources



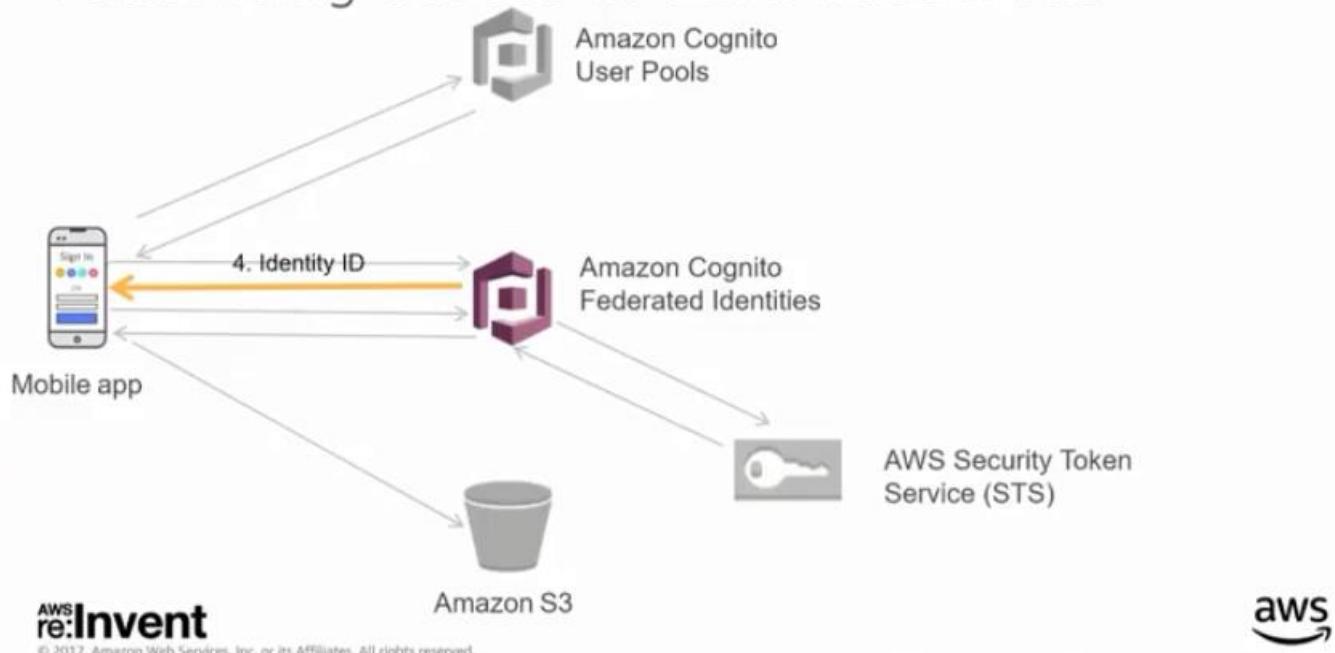
You get back a set of JWT tokens

Federating access to AWS resources



You then make a subsequent call to go ahead and pass the identity token to CFI,

Federating access to AWS resources



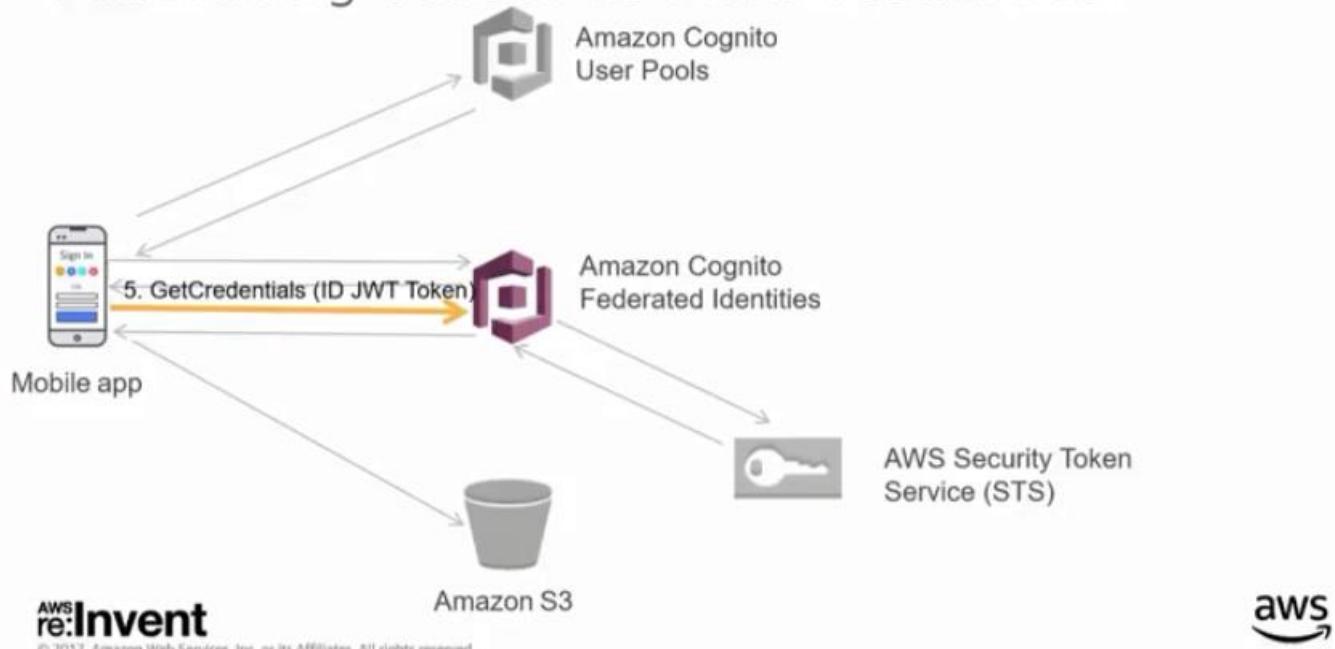
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



CFI will then return an Identity ID back to you.

Federating access to AWS resources



AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



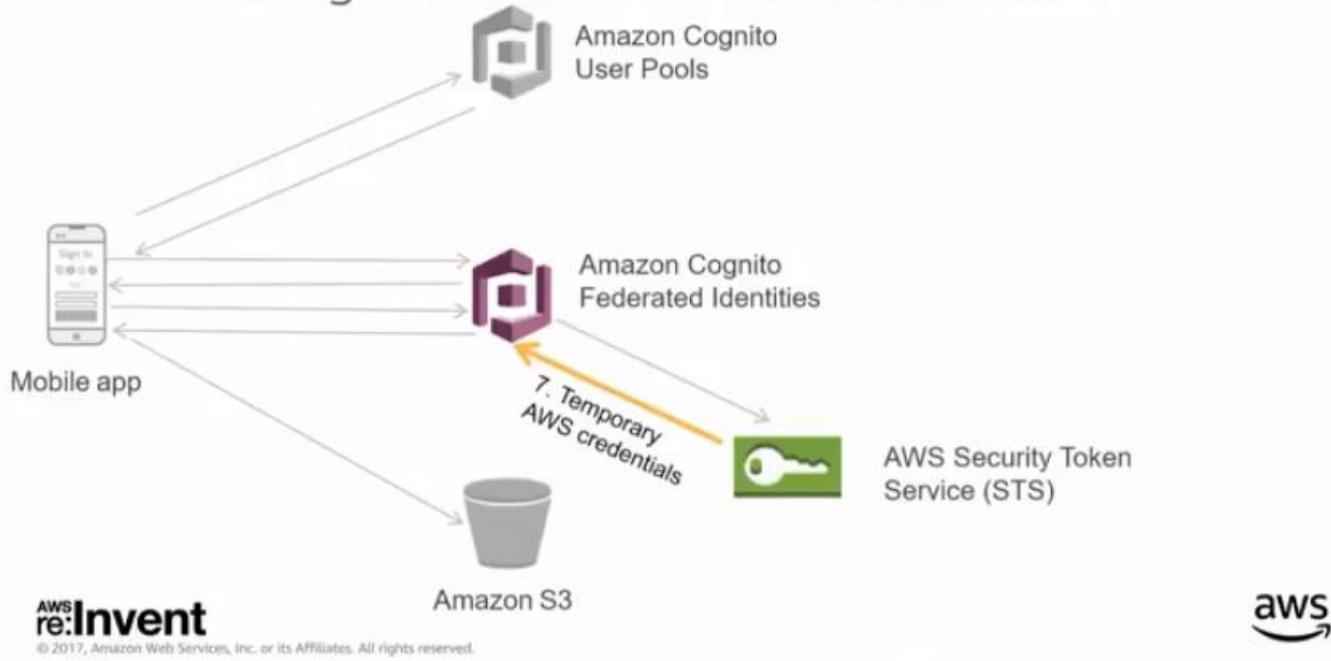
With the Identity ID and the JWT Identity Token, you can then call to the GetCredentials request as above

Federating access to AWS resources

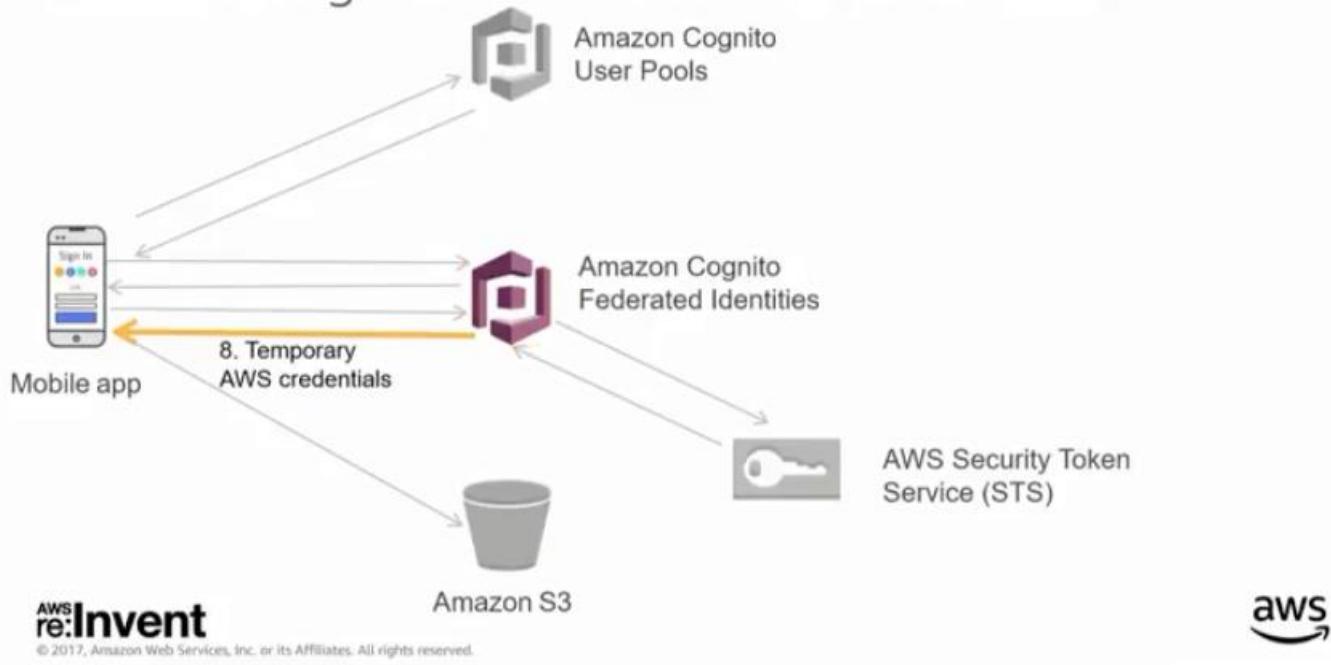


Cognito then goes to the AWS Security Token Service (STS) that will generate the particular IAM credentials for that role

Federating access to AWS resources

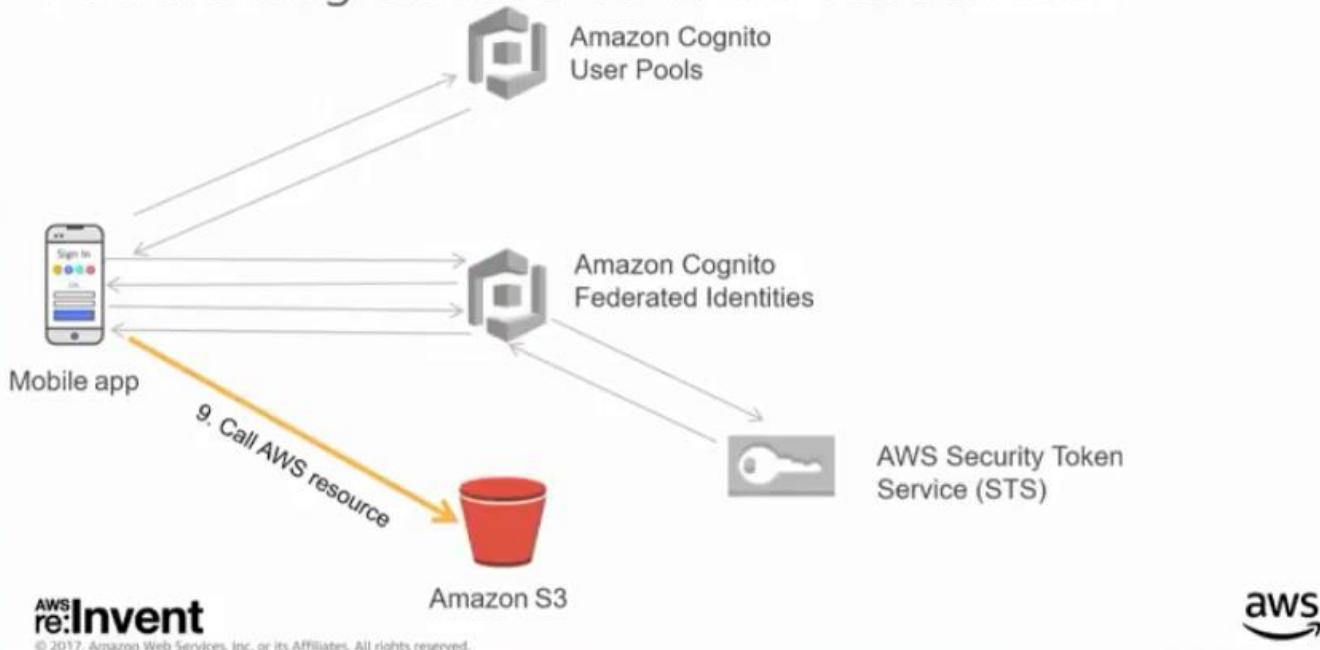


Federating access to AWS resources



Cognito then returns the IAM credentials back to the user in the form of an **IAM Role containing an access key, secret access key, and a session token**.

Federating access to AWS resources



Once you have the IAM Role credentials, you can then go ahead and use any AWS SDK from any language to interact with those AWS Resources like S3, DynamoDB, API Gateway, etc.

"What AWS permissions will those users have?"

"How do I give different users different AWS permissions?"



Fine-grained Role-Based Access Control

Unauthenticated users:

- Default role

Authenticated users

- Default role
- Choose role from rule
- Choose role from token

Fine-grained RBAC (role from rule)

Readable Attributes

Scopes Address Email Phone Number Profile

Attributes

<input type="checkbox"/> address	<input type="checkbox"/> nickname
<input type="checkbox"/> birthdate	<input type="checkbox"/> phone number
<input checked="" type="checkbox"/> email	<input type="checkbox"/> phone number verified
<input checked="" type="checkbox"/> email verified	<input type="checkbox"/> picture
<input checked="" type="checkbox"/> family name	<input checked="" type="checkbox"/> preferred username
<input type="checkbox"/> gender	<input type="checkbox"/> profile
<input checked="" type="checkbox"/> given name	<input type="checkbox"/> zoneinfo
<input type="checkbox"/> locale	<input type="checkbox"/> updated at
<input type="checkbox"/> middle name	<input type="checkbox"/> website
<input checked="" type="checkbox"/> name	<input checked="" type="checkbox"/> custom:department

Writable Attributes

Scopes Address Profile

Attributes

<input type="checkbox"/> address	<input type="checkbox"/> nickname
<input type="checkbox"/> birthdate	<input type="checkbox"/> phone number
<input checked="" type="checkbox"/> email*	<input type="checkbox"/> picture
<input checked="" type="checkbox"/> family name*	<input checked="" type="checkbox"/> preferred username
<input type="checkbox"/> gender	<input type="checkbox"/> profile
<input checked="" type="checkbox"/> given name*	<input type="checkbox"/> zoneinfo
<input type="checkbox"/> locale	<input type="checkbox"/> updated at
<input type="checkbox"/> middle name	<input type="checkbox"/> website
<input checked="" type="checkbox"/> name	<input type="checkbox"/> custom:department

*Required attributes are always writable



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



In your user pool, you can define different attributes like a department attribute above. You can then make a set of rules based on the attributes as below

Fine-grained RBAC (role from rule)

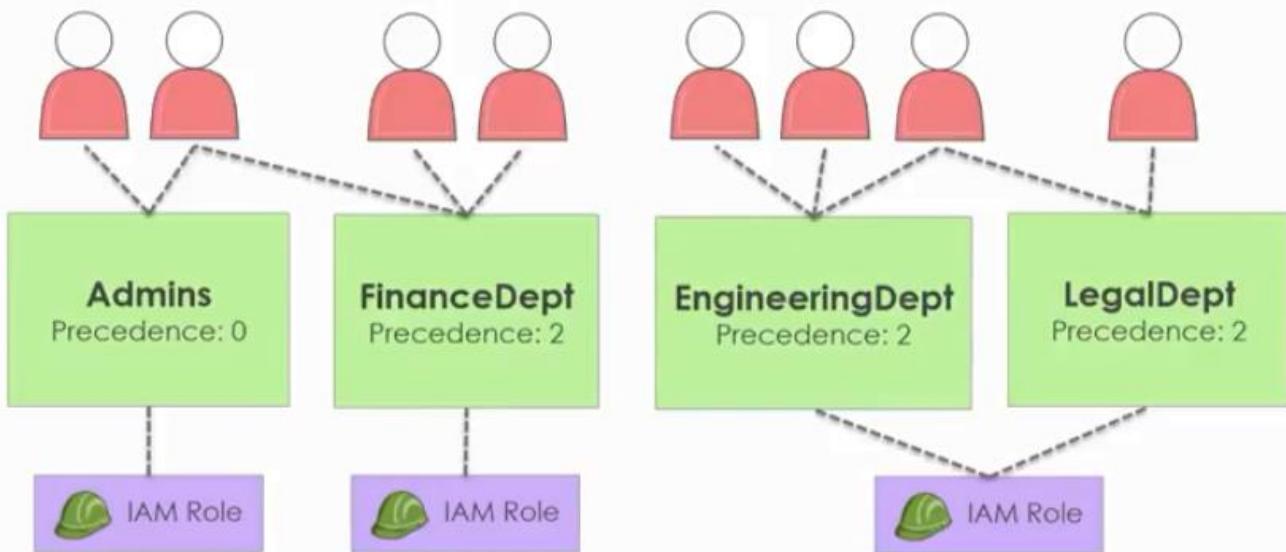
Claim	Match Type	Value	Role
<input type="text" value="custom:department"/>	<input type="button" value="Equals ▾"/>	<input type="text" value="Engineering"/>	<input type="button" value="EngineersRole ▾"/>
Add another rule			

If no rules match, the role resolution will be invoked. By default, it will fall back to the default role specified for this Identity Pool. You can also choose to DENY the request.

Role resolution

In this case we are parsing the department and if someone is a member of the engineering department, we give them the engineers role. You can define 1 or more rules that will be processed in a sequential order and the first matching role will be the effective role the user gets. **If you have a user that does not match any of the rules, you can decide to deny access to them or give them the default unauthenticated role.**

Fine-grained RBAC (role from token)



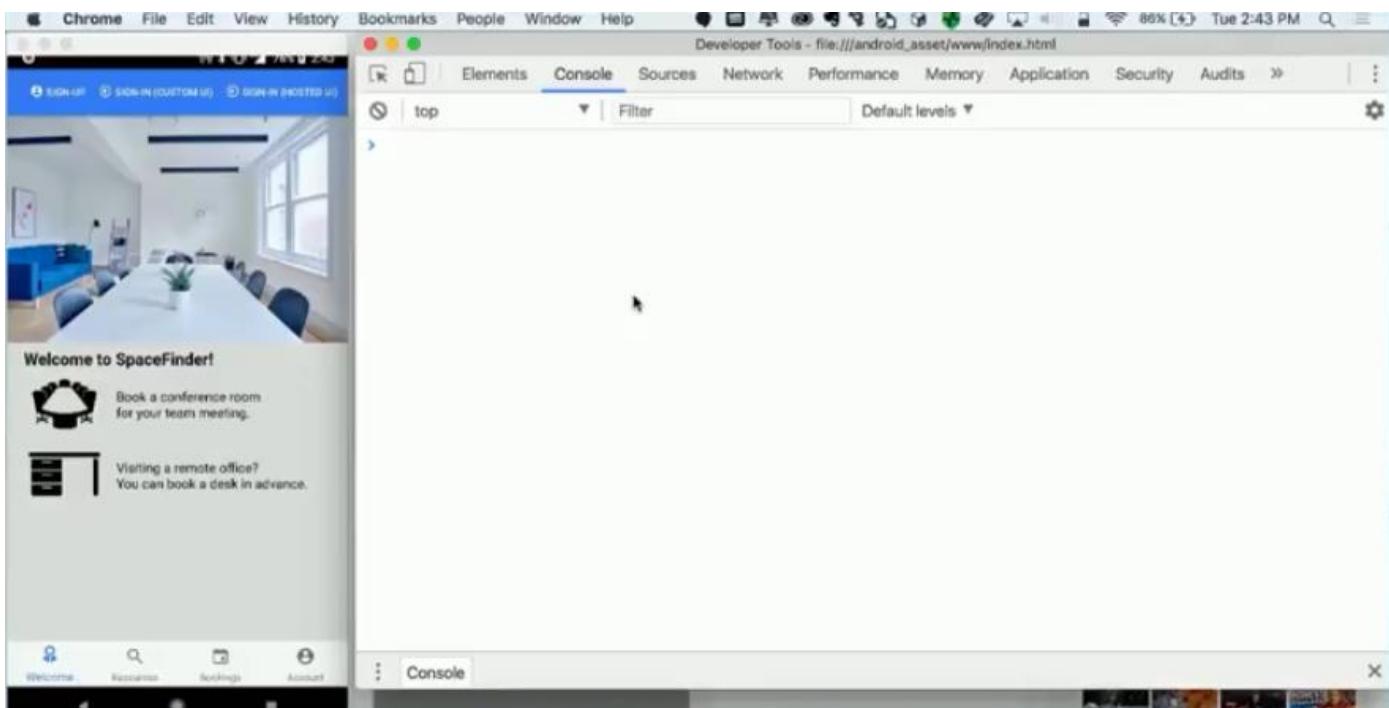
AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

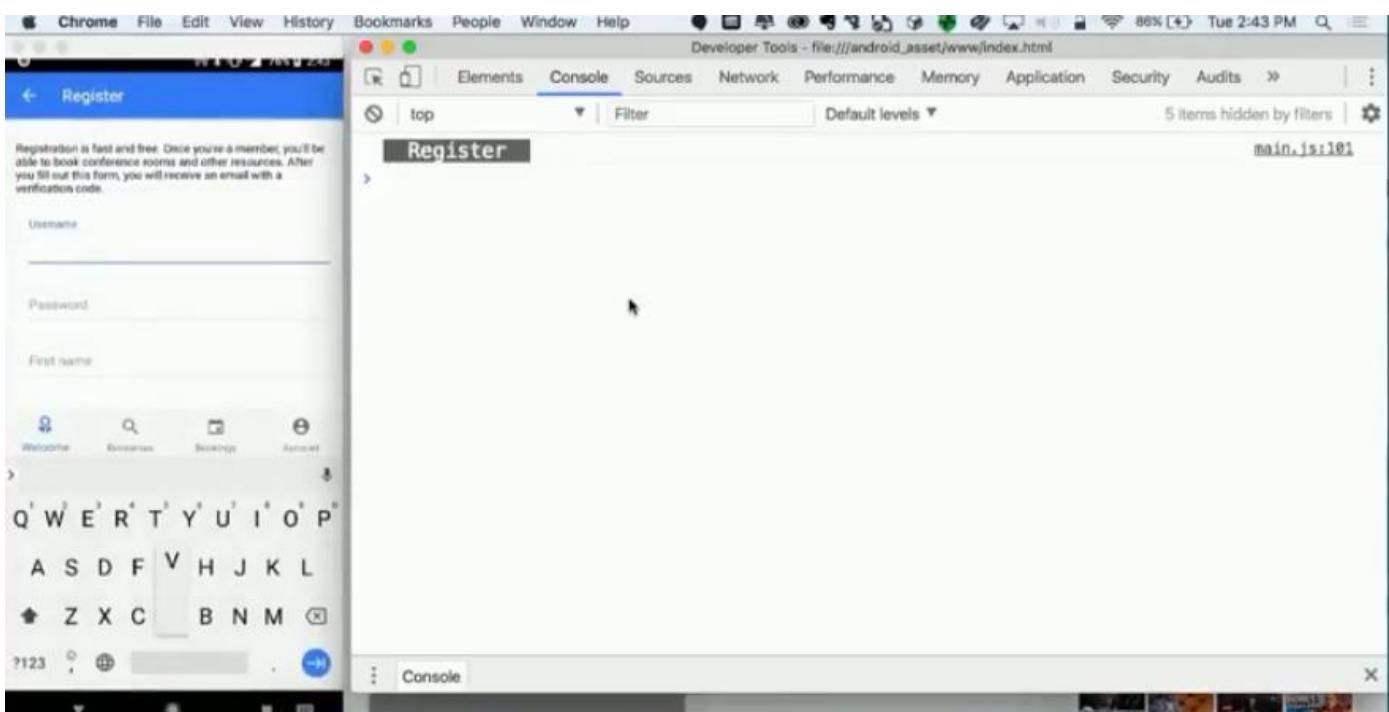


Using Cognito groups functionality, you are able to define different roles for different groups of users. A user can be in more than 1 group. With each group, you can specify an IAM Role for each group and a precedence tag. The user gets the group with the highest precedence tag if they belong to multiple groups since a user can only have 1 IAM role at any time. Cognito will then issue the appropriate token for a group dynamically.

DEMO



Let us start by signing up while we watch the logs that are generated



Developer Tools - file:///android_asset/www/index.html

Elements Console Sources Network Performance Memory Application Security Audits

Default levels | 5 items hidden by filters main.js:101

Register

First name: Vladimir
Last name: Budlov
Email address: Vladimir@aws-demos.com

SIGN UP CANCEL

Welcome Resources Bookings Account

Vladimir@aws.com · vladimir@aws.com

q w e r t y u i o p
a s d f g h j k l
z x c v b n m

?123

CONFIRM SIGN UP

Developer Tools - file:///android_asset/www/index.html

Elements Console Sources Network Performance Memory Application Security Audits

Default levels | 6 items hidden by filters main.js:101 main.js:501 main.js:502 main.js:101

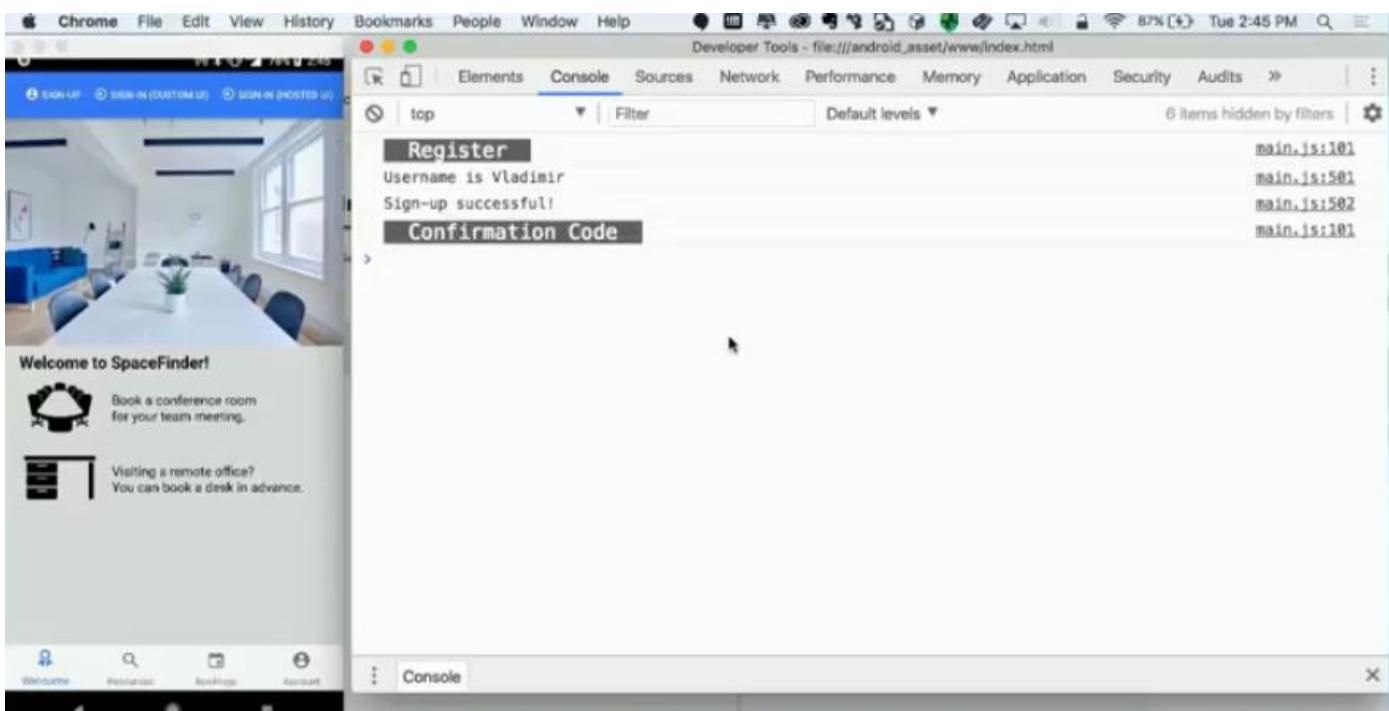
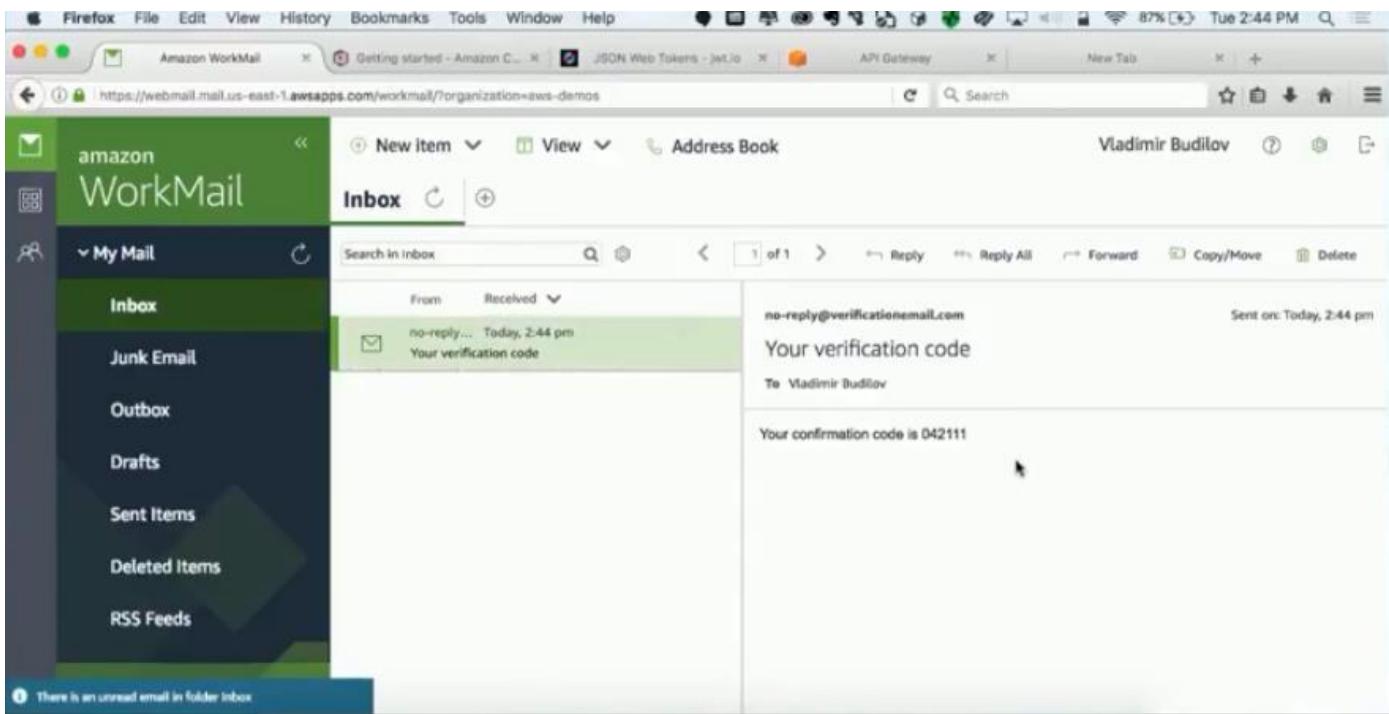
Register

Username is Vladimir
Sign-up successful!

Confirmation Code

Console

When I clicked the Register button, Cognito will sign me up but I am not confirmed yet. What is going to happen is that Cognito will send us a confirmation email to our email address with the code to confirm with.



We can now use that confirmation code to get confirmed. Then we can sign into the application with our username and password.

The screenshot shows a mobile browser window with a sign-in screen for "Cognito User Pools". The user has entered "Vladimir" for both "Username" and "Password". The developer tools console shows the following log messages:

- Register
- Username is Vladimir
- Sign-up successful!
- Confirmation Code
- Sign-In
- Authenticating user Vladimir

The log entries correspond to the main.js file at various line numbers: 101, 501, 502, 101, 101, 621.

The screenshot shows the same mobile browser window after a successful sign-in. A "Success!" dialog box is displayed, stating "You are now signed in." with details: "Username: Vladimir", "First name: Vladimir", and "Last name: Buldor". The developer tools console shows the following log messages:

- Register
- Username is Vladimir
- Sign-up successful!
- Confirmation Code
- Sign-In
- Authenticating user Vladimir
- > e {idToken: e, refreshToken: e, accessToken: e}
- Cognito User Pools Identity Token: [long token string]
- Cognito User Pools Access Token: [long token string]

The log entries correspond to the main.js file at various line numbers: 101, 501, 502, 101, 101, 621, 655, 18H16, 8201LrTic5ApJelEzb5yoVmIt1wlmphvF51rUwrfFyYlxbTBZEZyaWYSFgjcccyPXyTtWluo47u12xked10b6VwdBMNFfyZzgMR10UhVJ_zFn95WZoHfxPeuISay_nlbHrQZGg, 657.

Sign-in using Cognito User Pools

Vladimir

Password

Cognito User Pools Refresh Token:

```
eyJidHk101JKV101LCJlbmI01JBmjU2R0NNIiwIYwxnIjo1UNBLU9BRAVfIq0.KYvfgwBakJh2ldtTJAIR9UD0EbE_CzauCK110yX32WCNK
Enw5TJ50IQwdwjlqGciaeM9MF-wEReduXptAMRSzEfItBBNwk_4Q_C2XndPajAQR7_981gN52G5uJ74N4hWiyIWTO5mz0jQ-
rQgM9VGcRFiYKyn-oIw777tGA-BN5572-
_cPeYXdwtn0SH0j5zv56Y61_GbiaCozeVsMsKwdg5e438_zMeEzLlikad0vN9w9rCrRfuKf19N0mcfzL0ER6oIFidlh2FFvbvpnyZC0yYfVUZ
FwVjvFJvH0Ml4EmTcvX8LVxGxhZK9hqlIsosuByv07uf0-s_-SG-
```

Cognito User Pools User Groups: Vladimir belongs to group c

Cognito User Pools User Attributes:

```
> {sub: "6bbddc62-b2ec-4487-a5a8-3d25e1a1d10b", email_verified: true, iss: "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_Iaovdw202", cognito:username: "Vladimir", given_name: "Vladimir", -}
```

Success!

You are now signed in.

Username: Vladimir
First name: Vladimir
Last name: Butikov

Cognito User Pools Identity Token:

```
eyJraW0l01JuJuhnF42SyznI1JdutJN1ZlcyltcL1RcL21SShlnT0xldHFJvMjaNlwvVmZaa2Fvdz0iLCJhb6ci0iJSUzI1NIj9.eyJzdWlI01I
2YjhkNmM2h11M2VjlT000ctYTvh0C0zZD1ZTfhMwQxhG11CJlbwFpbF92ZxJpZmllZCI6dHJ1ZSwiaX0NiJoiaHR8cHM6XC9cL2NvZ25
pdG8taRwlwLnVwh30tM55hbwF6b25hd3mYv29tXc91cy1LYXN0LTffSFVdmR3mMyIiwiY29nbml0bzpc2VybmfTz5167l2sYwRpbwl
yIiwiZ2l2Zw5fbmftZS161ZsYwRpbwlyIiwiYXVkiJoiMzfwdnE3NG0B2G5sMz101jwNKF0b2xbpSISimV2N5wX2lKIjoiY2U0YzE4MhI
tZD04ZC@xMwU3L1LhNWEtMmWOTY20TMzZWZnIiwiid9rZW5fdXNlIjoiw01lCJhdXRoX3RpwpwU10jE1MTE5MDkxMtcsImV4cIC6MTk
xMjcxNyiwAf0IjoxNTEx0TA5MTE3LCJmYiW1pbh1fpmFtZS16IKj1ZGlsb3Y1lbwFpbC161ZsYwRpbwlyQGF3cy1kZw1vcy5jb201fQ.
qqVXYhDPZ11eePdm3AkZdcL5XKC2A0aWlpzbCHeJ-
srMsG_YdfMhRqis4gBYP6Eu2g3mXjz2p0c0xaMr2mwE1oC1wbwzurlie2ac1IN50u0ZjatdkA_DhXnUOpmpvypq_18HI6-
G0ogPwme63CvF09ab0tcsGeioFj0H66TfcrnFJSVck7w9pfDZw0DGaffmk7HnJv-
B20Tlr1c9ApjeEzb5yoVm1wLmpHvF51ru7fYlxTB8EzyaWYSPFgjcecyPxYttwluu047u12u2018ked10b6VwdBMNffyZzgMR18UhV3_zFn
95Wz0HfxPeiISav_nlbHrQZGg
```

Cognito User Pools Access Token:

```
eyJraW0l01JuJuhnF42SyznI1JdutJN1ZlcyltcL1RcL21SShlnT0xldHFJvMjaNlwvVmZaa2Fvdz0iLCJhb6ci0iJSUzI1NIj9.eyJzdWlI01I
hM2h11M2VjlT000ctYTvh0C0zZD1ZTfhMwQxhG11CJlbmVudf9pZC16In1NMGx0DF1LwQ0B0G0tMTFUny05YTvhLTJ1MDk2NjKzM2M
YSisRva2v0X3vZv16ImfYjZv2YzclyiInJnB381IjoiYXzdLmVnZ25pdG8uc2lnbmllnVzXiuYwRtaw41lCJpc3M0i1jodRhwcpc1lwv
Y29nbml0by1pZAUdXmtZWFzD0xlmFtyXpvbmF3cy5jb21c3vLwVh30tMv9JYw92ZCyzilC3leA10jE1MTE5MT13MTCsImlhcd16
MTUxMtkw0TEXhlywianRpjoiNzRjN2YyNz1tZj20C00NTU2Ltg2NWIzZG04YwVmZGZKymExIiwiY2xpZw50x21kIjoiMzfwdnE3NG0B2G5s
MwZ10TjwNKF0b2xbpSISinVzXkJuY11ijoiVmnhZG1taX1f0,AAEebEW10gG3L30w5YA-spCALNLEUOFH160o6p-
06AcgPr31jT_JA2cws5tLb0THLstMpkeQvCQwXnxggT_ym5z151CsYjPHsX9jgFh0Mkdjbeh3nwPnQIGPrMzprn-aFk7TB-
8XfmpGm2YEGahzrnwR8uN193nFCB982T42_ERx1crCRACK55F1CELBw0tUhm1S2g_459MzomEU7Kn4q0MsKN0tek6p2R3IUqPLTJ317h
isxtCx0tExY012H4nUem2x7daH01Tzh21nwaukfV0Lqy-rnw5RL15Hq25UMR1yKhwBdyGxmWlvwpDeuAlbbk4z-0r0FdcwZeg
```

Cognito User Pools Refresh Token:

```
eyJidHk101JKV101LCJlbmI01JBmjU2R0NNIiwIYwxnIjo1UNBLU9BRAVfIq0.KYvfgwBakJh2ldtTJAIR9UD0EbE_CzauCK110yX32WCNK
Enw5TJ50IQwdwjlqGciaeM9MF-wEReduXptAMRSzEfItBBNwk_4Q_C2XndPajAQR7_981gN52G5uJ74N4hWiyIWTO5mz0jQ-
rQgM9VGcRFiYKyn-oIw777tGA-BN5572-
_cPeYXdwtn0SH0j5zv56Y61_GbiaCozeVsMsKwdg5e438_zMeEzLlikad0vN9w9rCrRfuKf19N0mcfzL0ER6oIFidlh2FFvbvpnyZC0yYfVUZ
FwVjvFJvH0Ml4EmTcvX8LVxGxhZK9hqlIsosuByv07uf0-s_-SG-
```

The 3 JWT tokens we mentioned earlier are printed out above

Sign In

Sign-in using Cognito User Pools

Vladimir

Success!

You are now signed in.

Username: Vladimir
First name: Vladimir
Last name: Budičić

OK

Developer Tools - file:///android_asset/www/index.html

Console

Default levels

9 items hidden by filters

top

Cognito User Pools User Groups: Vladimir belongs to group c

Cognito User Pools User Attributes:

aws:sub: "6b0d6c62-83ec-4487-a5a8-3d25elaid10b", email_verified: true, iss: "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_1aoqvwd2o2", cognito:username: "Vladimir", given_name: "Vladimir", ...

Cognito Identity ID: us-east-1:36557168-c37c-4e4a-bb5e-de81479913af

AWS Access Key ID: ASIAJZPR1JWQIHNEQ

AWS Secret Access Key: wLuAwPtw02qAcSNLwJRzWe6tigtGTQ0u6+o1/z4f

AWS Session Token:

AgpGb3jPz2LuEFsaCKVzLwVhc30tMSKAQgCRdh15MsNzIxK3ejremEyko24yWL0JFPD10I5zghZUhj3uX5omGY4k/PEJEoy1mzBMy/qyazdB/8VqxCwGuesoMCD0q/ftNcmL5hyEAAx0ZFS4n2h9qEm3ZmyekmI5cN0qyK3r3cw8dJXkB5hwF2x/Vy1lBgeDeLx7Lm8NMD5bwRc479HaJ4K0nSZA14UDBbjp41kZdskaAvj+RroMo97rUYeUcJGXFnVcBoPyVTJLxuCgHy75g1SX/J5FVLUxbv85STkpJ39jXjXyKtVhWjZ9/gkxCOY4F35AmzbFc01ft1kjB3J4pvvrPpb4ttxo+0+mUV/FUYUKCggd/HhxqhgY1sP//////////ARAAGgw3ND1zNjA1MDgzb2zc1DjZPmod0HxngN6+bXira8cihhxJvJv5YrdG8c91ksbMJH49wmsIfownrAcYhb10ky0tTk3vwJ4x1uU/1NgkGrV17VfL35+ROTZFUHrsretIEFaovXCGcV1lwWq8PZpn/YwMLE775sgsu0gx1xj1dwqdg0Bt7NPyD1Z0rFnXkR2YBwKvJgXzBb/6/Nluj4ldfjH1Y1ZhzqRAHJS80uQfVpgQXc+HoWBLiy4M+w4Q3Xtvvtbp+oHb0ho9H6Nk68QPrTudxeFyZ+v+n9b18hkwmt96ygbMVEv5uswBhY1grXdxJt7B8wLwN2sA4gmMaJwNaJh1XBj8UKz2nPNB0h0ia0GDSuJ9hMCFN2DjSMly8X9jLbc5g1SbTRSBK//rvuNGVNsAgs871AbuDs3ab034g6/4X9NmxdvJsdCLs4C8j3312SrC0GjKTB210PrlucJmFEGLyrrX9bHta8cf+408bhz/J+V3UN9R1LwvqlDfJ+z+e9q2713esfqf5G3NF4zIz28N0yPwNz2sA6L6rds3JALGR0Re1Xbm-BPsIZB/1c7tuxBf8/y0VGxpxduyL6KdmWk+4mbY0mnHgBnjpXup5Crv4yHxGeboqzKwge91oMaC5IR1L1GeuzPjsnQXQg16euukbnEY38us07fx041I7B7K46x3n1zS2B7n23FAcG/yGYYAn3229j5M7YIpFld/Un05Ke4zobqAxv2nahKur/x0BPMuZp/hBupHuK/NT1v0jBPM7j2baftE0eyvM/YwCwLuk676eN9qnCs075wBg3Ny rEzZhSho]+Ls6KgJqHigdFmzDmjvuS2MAxsttYMWtsBSWehaKy+rcSUfuxunHC62ya1zf21mxLr+2lF1epHLDRX9d68fY434LLILG7K0gs56yZKULMV9kdqX339u80/rfp5JgDkAIRT/7UJalop0xySN8n@tGAGVJkGr80PfPvgg0w/B330AU=

And the IAM role access credentials that are vended by STS is shown above

The screenshot shows the jwt.io website's debugger interface. On the left, under 'Encoded', the token is displayed as a long string of characters: eyJraWQiOjJubnF4V2syN1JIdUtJN1ZcLytcL1RcL2I... The right side, under 'Decoded', shows the token structure. The 'HEADER' section contains the algorithm (RS256) and kid (nnqxWk27RHuKI7V/+T/1RHgOLetqIVbZ6/VfZkaUw=). The 'PAYLOAD' section contains the user information: sub (6b8d6c62-b3ec-4487-a5a8-3d25e1a1d10b), email_verified (true), iss (https://cognito-idp.us-east-1.amazonaws.com/us-east-1_Iaoovdw2o2), cognito:username (Vladimir), and given_name (Vladimir). The 'VERIFY SIGNATURE' button is visible at the bottom.

This screenshot shows a very large and complex payload for a JWT token. The payload is filled with numerous parameters, including sub, email_verified, iss, cognito:username, given_name, aud, event_id, token_use, auth_time, exp, iat, family_name, and email, all associated with the value 'Vladimir'. The token itself is extremely long, starting with 5pdG8taWRwLnVzLWVhc3QtMS5hbWF6b25hd3MuY29tX and ending with n_1BHT6-. The rest of the interface remains consistent with the first screenshot, showing the header and verify signature buttons.

The payload in the JWT contains all the information that the user entered, we can use these details to perform authorization within the application to grant access to certain resources or not as well as in our downstream services for further customization.

The screenshot shows a JWT token being analyzed on jwt.io. The token itself is displayed in a large text area on the left, consisting of several segments of encoded data. To the right, there are two input fields for verification: one for "VERIFY SIGNATURE" which contains a placeholder for "Public Key or Certificate" and another for "Private Key (RSA)" which contains a placeholder for "Private Key (RSA). Enter the it in plain text only if you want to generate a new token. The".

The JWT signature can also be used in our downstream services to make sure that this is a valid JWT token created by Cognito.

This screenshot displays a mobile browser interface with developer tools open. On the left, a user profile page for "My Account" is visible, showing fields for "Your profile image" (with a "SELECT IMAGE" button), "Change password" (with a "CHANGE PASSWORD" button), and "View Admin features" (disabled). On the right, the developer tools' "Console" tab is active, showing a log of AWS-related API calls and their responses. Key entries include:

- Cognito User Pools User Groups: Vladimir belongs to group c
- Cognito User Pools User Attributes: (large JSON object)
- Cognito Identity ID: us-east-1:36557168-c37c-4e4a-bb5e-de81479913af
- AWS Access Key ID: ASIAJZ7PRIJEWQIHVNED
- AWS Secret Access Key: wUwAptw02qAcSNPLwjRZew6tietgGTQu6+o1/z4f
- AWS Session Token: (large JSON object)

Let us now see an example of the user uploading an image

Chrome File Edit View History Bookmarks People Window Help

Developer Tools - file:///android_asset/www/index.html

Elements Console Sources Network Performance Memory Application Security Audits

Cognito User Pools User Groups: Vladimir belongs to group c main.js:681

Cognito User Pools User Attributes: main.js:685

> {sub: "b68d6c62-b3ec-4487-a5a8-3d25elaid10b", email_verified: true, iss: "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_Iaovdw2o2", cognito:username: "Vladimir", given_name: "Vladimir", ...}

Cognito Identity ID: us-east-1:36557168-c37c-4e4a-bb5e-de81479913af main.js:690

AWS Access Key ID: ASIAJZ7PRIJEWQIHNNEQ main.js:692

AWS Secret Access Key: wUwAPtw02qAcSNPLwjRZew6tietgGTQu6+o1/z4f main.js:694

AWS Session Token: main.js:696

AgogB3jPZ2luEfsoCXvzLvhc3QtMSKAAGCrDh15NsIzK3ejremEyko24yWLOJFPD1D0ISzghZUhj3uX5omGY4k/PEJEoyImzBy/gqazdB/8VqxCvGuesoMCDDq/ftNcm1ShyEAAMDZFS4n2h9qEm32myeekm15cNDqykKr3cw8djXKb5hwF2X/vYwlBgeDeLX7Lm@NMD5bwRc479HaJ4K0nSZA14UDBbjfp41KzdskApvJ+RroBMo97+RUyeUcJGXFnVCboFmYVtJixuQpHt5giSX/j5FVuRxRbV0SSTk3j9jXjXyKTvhjWz9/gRxC0Y4F35AmDzFc0ifTjBJ4pvzrPb4tox2o+mUV/FUYUKCggd/HxgqhgYIsP//////////ARAAGw3NDIzNjA1MDgzNzc1DjzPmod8HxmgN6+bXiraBcjhXvJvV5YrdG8c91ksbMjH49wmsIfownrAcYhbiQkyotTk3vwJ4wXlW/!NgkGrV17VF135+ROTZFURHsretIEeoAVXCgCvLwWq8PZpn/YWmlE77sgsu0gx1Xjdwqdq8t7NPy01zQrFnXUkr2YBwKvJgXzVb/6/NLuJ4ldfjHY1ZHqRAHJS0uqFvgQxc+HoW0Liy4M+w4Q3XtVvtbp+KwDeoHo9H6Nk680PvTudxeFyZ+vnb10hkwtBb6ygMBEVSnVob003K0gwvs5uvAw8hY01grXdxJb78AWlWN2sA4AgmMaWajnjhlX8j8Ukz2nPNBg5ifo08GvsUuh9MCFND2jSMylq8X9jLbc5heTRSBK/r/vuNGvNqa5871AbUscs3Ab034gB@/4X9CMmxvdJdCS14cB3J12SrC0Gjkt82105qrJmFEGLyrX9bHta0dcF+40Bbhz/jVU3VN9RL1vwqlQDT/J+ze8q2T3esfqf5G3NF4zIzc0NyePZ0zxU6L6rdsJA1GLOReiXbh-BPsIZ8/!c7tuxBzF8/ywOGXpxDuyL6KdMWKK+amb0mnhkGbnjpXup5Cr4wYn49HjxheBoqzKwge91ohCa5IR1LgeuzpSjw0Qg16uuRkbnlLEY38usT0TX041I78JK4Gx3n1zB7n2JFaCG/gYYJAn3229j5M7YIpFld/Un05Ke4zobAxv2nahKurX00FMU2p+hBupK/NTLvbjBPQM7j2bafe0eyvM/YwCwLUk676eN9qncs075w6Wg3NyrezZhShoJ+Ls6kGjqHiqdFmzDmjvuSZMAxsttYMWtusBSWehakY+rCSufuxunHC62ya1zF21WxLR+2lF1EpHLDRx9d68Fy434LL1LG7K0gs56yZxULMV9kdqX339uB0/rfp5JgdDAkRT/7Ujalop0xySNBn0tGAGVJkGr00P

Account main.js:101

Displaying ImageSelector main.js:104

Console

Chrome File Edit View History Bookmarks People Window Help

Developer Tools - file:///android_asset/www/index.html

Elements Console Sources Network Performance Memory Application Security Audits

Your profile image

SELECT IMAGE

Change password

As a general security best practice, we recommend changing your password every few months.

CHANGE PASSWORD

View Admin features (disabled)

Uploading image to Amazon S3... You can see that calls

Image selected: [file:///data/user/10/com.ionicframework.spacefindermobileapp260912/cache/tmp_IMG_20171128_0814335.jpg] main.js:2320

Converting to Base64 image main.js:2321

Converting to Blob main.js:2324

file:///data/user/10/com.ionicframework.spacefindermobileapp260912/cache/tmp_IMG_20171128_0814335.jpg main.js:2301

Uploading image to S3 main.js:104

Attempting image upload to spacefinder-development-stack-userdatabucket-1a7yus75ujeia/us-east-1:36557168-c37c-4e4a-bb5e-de81479913af/775135716-1511909204261.jpg main.js:2254

Console

The screenshot shows a mobile application interface on the left and a developer tools console on the right. The mobile app has a "My Account" header, a "Your profile image" section with a thumbnail of a group photo, a "SELECT IMAGE" button, a "Change password" section with a note about security best practices, and a "CHANGE PASSWORD" button. Below these are sections for "View Admin features?" and "View Admin features (disabled)". The developer tools console shows the following log entries:

```

PZpn/YwM1E775gsu0gx1X1jdwqdgQbt7NPyD1ZqrFnXUKr2YBwKvJgXzVb/6/Nluj4LdfjHt1ZhZqRAHJS8UqFvgQXc+HoW0l:iy4Mh4Q3Xt
vVtbp+KwDeoHoa9H6Nk680PVtJdxeFyZ+v9b10hkwtBb6ygMBEVSnVob003K0gws5uvAwBhY1grxDxtJb78AW1wN2sA4AgmMoWajnjh1
X8j8Ukz2nPNBg51fo8GdvSuUh9MCFN2jSMyLq8X9jLbcSHeTRSBK/r/vuNGvNQs871AbUsCs3Ab834gB0/4X9CMmxvdJdCS14C8JJ312
SrCOgjkT821850rlucJmFEGLyrX9bhA0dcf+40Bbhz/jVU3VN9RL1vwqlQDT/J+ze8q2T3esfaf5G3NF4zIzc0NyePZ0zxJ6L6rdsJA1GR
0ReiXbh-BPsIZ8/1c7txbZf8/yw0VGpxduyl6KdMWKK+4mbY0mnhGBnjpXup5CrV4wYn49HjhXeBoqzKwge91oMc5IR1GeuZp5jsWQxQ
g16uuRkbnlLEY38usT0fX041178JK4Gx3n1zsB7n2JFaCG/gYYJAAn32Z9j5M7Y1pFld/UnQ5Ke4zoBqAxv2nahKurX00FMU2p+hBupUK/NTl
v0jBPQ7j2bafe0eyvM/YwCwLUk676eN9qncs075w6Wg3NyrEzZhShj+Ls6KGJqH1qdFmz0mvu5ZMAxsttYNTusBSWehaKy+rCSUFuxun
HC62ya1F21WxLR=2l1EpHLDRx9d6fY434LLILG7K0gs56yZKULMV9kdqX339uB0/rfp5JgdAKRT/7UJalopOxySNBn0tGAGVJKgR00P
rPvvg00w/8330AU=

```

Account

Displaying ImageSelector
Image selected: [file:///data/user/10/com.ionicframework.spacefindermobileapp260912/cache/tmp_1 main.js:2320 MG_20171128_0814335.jpg]
Converting to Base64 image
Converting to Blob
file:///data/user/10/com.ionicframework.spacefindermobileapp260912/cache/tmp_IMG_20171128_081435.jpg
Uploading image to S3
Attempting image upload to spacefinder-development-stack-userdatabucket-1a7yus75ujeia/us-east-1:36557168-c37c-4e4a-bb5e-de81479913af/775135716-1511909204261.jpg
Successfully uploaded image to S3.
Image can be viewed at: https://s3.amazonaws.com/spacefinder-development-stack-userdatabucket-1 a7yu..us-east-1:36557168-c37c-4e4a-bb5e-de81479913af/775135716-1511909204261.jpg

main.js:101
main.js:104
main.js:2320
main.js:2321
main.js:2324
main.js:2380
main.js:104
main.js:2254
main.js:2266
main.js:2268

Console

The uploaded image went directly from the mobile device to S3, it did not go through another server. That was possible using IAM Roles, and the image is stored in an S3 bucket that only this user has access to.

The screenshot shows the Amazon Cognito console landing page. It features the Cognito logo and the heading "Amazon Cognito". Below the heading, there is a brief description of what Cognito does: "Amazon Cognito makes it easy for you to have users sign up and sign in to your apps, federate identities from social identity providers, secure access to AWS resources and synchronize data across multiple devices, platforms, and applications." At the bottom of the page are two blue buttons: "Manage your User Pools" and "Manage Federated Identities".

The screenshot shows the AWS User Pools console. At the top, there are several tabs: 'Amazon WorkMail', 'User Pools - Amazon Cogni...', 'JSON Web Tokens - jwt.io', 'API Gateway', and 'New Tab'. The URL in the address bar is https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1_laovdw2o2/details?_k=kwmmwrx9. The main header says 'aws User Pools | Federated Identities' with 'jpirtie' as the user. The region is 'US East (Virginia)' and support links are available. Below the header, the title 'Your User Pools' is displayed, followed by a large blue button labeled 'Create a user pool'. The main content area shows one user pool card:

spacefinder-development-userPool

The screenshot shows the details for the 'spacefinder-development-userPool'. The left sidebar has a 'General settings' tab selected, along with other options like 'Users and groups', 'Attributes', 'Policies', etc. The main content area displays the following information:

Pool Id	us-east-1_laovdw2o2
Pool ARN	arn:aws:cognito-idp:us-east-1:742360508337:userpool/us-east-1_laovdw2o2
Estimated number of users	5
Required attributes	given_name, family_name, email
Alias attributes	none
Username attributes	none
Custom attributes	custom:ext_idp_access_tkn, custom:ext_idp_refresh_tkn, custom:ext_idp_expires_in
Minimum password length	8
Password policy	uppercase letters, lowercase letters, special characters, numbers

The **Pool Id** is the unique ID that you will be referencing throughout your application,

General settings

Users and groups

Attributes

Policies

MFA and verifications

Advanced security beta

Message customizations

Tags

Devices

App clients

Triggers

Analytics

App integration

App client settings

Domain name

UI customization

Import users

User name

Create user

Username	Enabled	Status	Updated	Created
Google_106300170295478073663	Enabled	EXTERNAL_PROVIDER	Nov 28, 2017 5:11:01 PM	Nov 28, 2017 9:51:17 AM
Okta_justin-okta@aws-demos.com	Enabled	EXTERNAL_PROVIDER	Nov 28, 2017 5:12:42 PM	Nov 28, 2017 9:47:58 AM
Vladimir	Enabled	CONFIRMED	Nov 28, 2017 10:44:58 PM	Nov 28, 2017 10:44:23 PM
admin1	Enabled	CONFIRMED	Nov 28, 2017 9:30:26 AM	Nov 28, 2017 9:30:24 AM

These are the users that have logged in or registered with this particular Cognito User Pool,

General settings

Users and groups

Attributes

Policies

MFA and verifications

Advanced security beta

Message customizations

Tags

Devices

App clients

Triggers

Analytics

App integration

App client settings

Domain name

UI customization

Create group

Group Name	Description	Precedence	Updated	Created
adminGroup	Cognito user group for administrators	-	Nov 28, 2017 9:30:25 AM	Nov 28, 2017 9:30:25 AM
clientGroup	Cognito user group for spacefinder users	1	Nov 28, 2017 9:30:25 AM	Nov 28, 2017 9:30:25 AM
us-east-1_laovdw2o2_Google	Autogenerated group for users who sign in using Google	-	Nov 28, 2017 9:31:41 AM	Nov 28, 2017 9:31:41 AM
us-east-1_laovdw2o2_Okta	Autogenerated group for users who sign in using Okta	-	Nov 28, 2017 9:32:00 AM	Nov 28, 2017 9:32:00 AM

These are the groups created for use with this Cognito User Pool that users can belong to,

The screenshot shows the AWS User Pools console with the URL https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1/_aoovdw2o2/groups/adminGroup?_k=lg26. The page title is "Groups > adminGroup". On the left, there's a sidebar with various settings like General settings, Users and groups, Attributes, Policies, etc. The main content area shows the group details:

- Description: Cognito user group for administrators
- Role ARN: `arn:aws:iam::742360508337:role/spacefinder-development-s-CognitoIdentityPoolAuthA-J5HGS0N215N2`
- Precedence: 0
- Updated: Nov 28, 2017 9:30:25 AM
- Created: Nov 28, 2017 9:30:25 AM

You can see that this group is associated with a particular IAM Role (Role ARN), this means that any user placed into this group will have this IAM Role as part of their token when authenticated.

The screenshot shows the AWS User Pools console with the URL https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1/_aoovdw2o2/attributes?_k=zuec9. The page title is "spacefinder-development-userPool". On the left, there's a sidebar with various settings like General settings, Users and groups, **Attributes**, Policies, etc. The main content area shows the sign-in configuration:

How do you want your end users to sign in?

You can choose to have users sign in with an email address, phone number, username or preferred username plus their password. [Learn more.](#)

Username - Users can use a username and optionally multiple alternatives to sign up and sign in.

- Also allow sign in with verified email address
- Also allow sign in with verified phone number
- Also allow sign in with preferred username (a username that your users can change)

Email address or phone number - Users can use an email address or phone number as their "username" to sign up and sign in.

- Allow email addresses
- Allow phone numbers
- Allow both email addresses and phone numbers (users can choose one)

Which standard attributes are required?

These attributes were selected when the pool was created and cannot be changed.

Screenshot of the AWS User Pools console showing standard attributes configuration.

Standard Attributes Configuration:

- MFA and verifications:**
 - Also allow sign in with verified email address
 - Also allow sign in with verified phone number
 - Also allow sign in with preferred username (a username that your users can change)
- Email address or phone number:** Users can use an email address or phone number as their "username" to sign up and sign in.
 - Allow email addresses
 - Allow phone numbers
 - Allow both email addresses and phone numbers (users can choose one)

Which standard attributes are required?

These attributes were selected when the pool was created and cannot be changed.

Required	Attribute	Required	Attribute
<input type="checkbox"/>	address	<input type="checkbox"/>	nickname
<input type="checkbox"/>	birthdate	<input type="checkbox"/>	phone number
<input checked="" type="checkbox"/>	email	<input type="checkbox"/>	picture
<input checked="" type="checkbox"/>	family name	<input type="checkbox"/>	preferred username
<input type="checkbox"/>	gender	<input type="checkbox"/>	profile
<input type="checkbox"/>	given name	<input type="checkbox"/>	zoneinfo

Cognito comes with default attributes but we can also create custom attributes like 'what is your favorite color' or 'has the user paid?'

Screenshot of the AWS User Pools console showing app client configuration.

spacefinder-development-userPool

General settings:

- Users and groups
- Attributes
- Policies
- MFA and verifications
- Advanced security beta
- Message customizations**
- Tags
- Devices
- App clients**
- Triggers
- Analytics

Which app clients will have access to this user pool?

The app clients that you add below will be given a unique ID and an optional secret key to access this user pool.

App client details:

spacefinder-development-userPool-admin-client	
App client id	3ef4ne8839uove8d8acfBass7i
Show Details	

The screenshot shows the AWS User Pools console under the 'Federated Identities' tab. On the left, a sidebar lists various options like Policies, MFA and verifications, Advanced security, Message customizations, Tags, Devices, App clients (which is selected), Triggers, Analytics, App integration, Federation, Identity providers, and Attribute mapping. The main area displays two app client entries:

- spacefinder-development-userPool-admin-client**
App client id: 3ef4ne6839uove8d8aofbass7
[Show Details](#)
- spacefinder-development-userPool-app-client**
App client id: 31pvq74d4dn1fu92p5qtolim
[Show Details](#)

In order to actually talk to Cognito, you need an **App ID**. Here we have configured 2 app IDs, an **admin-client App ID** and the **app-client App ID** that is being used by this Cognito application as a client.

The screenshot shows the AWS User Pools console under the 'Federated Identities' tab. The sidebar is identical to the previous screenshot. The main area shows the configuration for the app-client ID:

- spacefinder-development-userPool-app-client**
App client id: 31pvq74d4dn1fu92p5qtolim
[Show Details](#)

At the bottom of the page, there are two buttons: [Add another app client](#) and [Return to pool details](#).

The app-client ID is configured to use SRP as below

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cognito JSON Web Tokens - jwt.io API Gateway New Tab

https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1_laovdw2o2/clients?_k=sdurdd

Search

jpirie US East (Virginia) Support

AWS User Pools | Federated Identities

Attribute mapping

App client secret
(no secret key)

Refresh token expiration (days)
30

Enable sign-in API for server-based authentication (ADMIN_NO_SRP_AUTH) [Learn more.](#)

Only allow Custom Authentication (CUSTOM_AUTH_FLOW_ONLY) [Learn more.](#)

Set attribute read and write permissions

Save app client changes

Add another app client [Return to pool details](#)

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cognito JSON Web Tokens - jwt.io API Gateway New Tab

https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1_laovdw2o2/clients?_k=sdurdd

Search

jpirie US East (Virginia) Support

AWS User Pools | Federated Identities

Attributes

Select the user attributes this app client can read and write. You can select standard scopes that include multiple attributes and you can select a set of individual attributes.

Readable Attributes

Scopes Address Email
 Phone Number Profile

Attributes

address phone number
 birthdate phone number verified
 email picture
 email verified preferred username
 family name profile
 gender zoneinfo
 given name updated at
 locale website
 custom:ext idp access

Writable Attributes

Scopes Address Profile

Attributes

address phone number
 birthdate picture
 email* preferred username
 family name* profile
 gender zoneinfo
 given name* updated at
 locale website
 middle name website
 name custom:ext idp access
 tkn custom:ext idp refresh

Readable Attributes

Scopes: Address Email
 Phone Number Profile

Attributes

- address
- birthdate
- email
- email verified
- family name
- gender
- given name
- locale
- middle name
- name
- nickname
- phone number
- picture
- preferred username
- profile
- zoneinfo
- updated at
- website
- custom:ext idp access tkn
- custom:ext idp refresh tkn
- custom:ext idp expires in

Writable Attributes

Scopes: Address Profile

Attributes

- address
- birthdate
- email*
- family name*
- gender
- given name*
- locale
- middle name
- name
- nickname
- phone number
- picture
- preferred username
- profile
- zoneinfo
- updated at
- website
- custom:ext idp access tkn
- custom:ext idp refresh tkn
- custom:ext idp expires in

*Required attributes are always writable

We can also set what this client has access to do in regards to reading and writing. Like preventing the client from modifying the 'paid' attribute value. We will want an admin user to be able to modify the 'paid' attribute value as below

General settings

Which app clients will have access to this user pool?

The app clients that you add below will be given a unique ID and an optional secret key to access this user pool.

App clients

spacefinder-development-userPool-admin-client	
App client id	3ef4ne8839uove8d8aoBass7i

Show Details

spacefinder-development-userPool-app-client	
App client id	31pvq74d4dn11fu92p5qtolim

The screenshot shows the AWS User Pools console under the 'Federated Identities' section. On the left, a sidebar lists various options: Devices, App clients (selected), Triggers, Analytics, App integration (App client settings, Domain name, UI customization, Resource servers), and Federation (Identity providers, Attribute mapping). The main panel displays settings for an app client named 'spacefinder-development-userPool-app-client'. It includes fields for 'App client secret' (no secret key), 'Refresh token expiration (days)' (set to 1), and two checkboxes: 'Enable sign-in API for server-based authentication (ADMIN_NO_SRP_AUTH)' (checked) and 'Only allow Custom Authentication (CUSTOM_AUTH_FLOW_ONLY)' (unchecked). Below these are buttons for 'Set attribute read and write permissions' and 'Save app client changes'. A modal window at the bottom is titled 'spacefinder-development-userPool-app-client'.

We have turned off SRP here,

The screenshot shows the same AWS User Pools console interface. The 'Attributes' section is now expanded, showing tabs for 'Readable Attributes' and 'Writable Attributes'. A note below the tabs states: 'Select the user attributes this app client can read and write. You can select standard scopes that include multiple attributes and you can select a set of individual attributes.' The 'Writable Attributes' tab is currently selected.

The screenshot shows the AWS User Pools console with the URL https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1_jaovdw2o2/clients?_k=sdurdd. The page displays two sections: 'Readable Attributes' and 'Writable Attributes'. Under 'Readable Attributes', several checkboxes are checked, including 'Scopes', 'Address', 'Email', 'Phone Number', and 'Profile'. Under 'Writable Attributes', many checkboxes are checked, including 'address', 'birthdate', 'email', 'email verified', 'family name', 'gender', 'given name', 'locale', 'middle name', 'name', and various custom attributes like 'custom:ext idp access tkn' and 'custom:ext idp refresh tkn'. Other attributes like 'phone number', 'picture', 'preferred username', 'profile', 'zoneinfo', 'updated at', 'website', 'middle name', 'name', and 'nickname' have checkboxes that are not checked.

The admin app ID has access to all the available attributes.

The screenshot shows the AWS User Pools console with the URL https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1_jaovdw2o2/triggers?_k=c9nfnc. The left sidebar shows navigation options: General settings, Users and groups, Attributes, Policies, MFA and verifications, Advanced security (beta), Message customizations, Tags, Devices, App clients, Triggers (selected), Analytics, App integration, App client settings, and Device农农. The main content area is titled 'spacefinder-development-userPool' and features a heading 'Do you want to customize workflows with triggers?'. It explains that users can make advanced customizations with AWS Lambda functions for different events. Two sections are shown: 'Pre sign-up' and 'Pre authentication'. Both sections have a 'Lambda function' dropdown menu set to 'none'.

Triggers are a way for you to customize the authentication and the authorization flows,

The screenshot shows the AWS User Pools console under the Federated Identities tab. On the left sidebar, the 'Triggers' option is selected and highlighted in orange. The main content area is divided into four sections: 'Pre sign-up', 'Pre authentication', 'Custom message', and 'Post authentication'. Each section contains a brief description and a dropdown menu labeled 'Lambda function' with the value 'none'.

- Pre sign-up:** This trigger is invoked when a user submits their information to sign up, allowing you to perform custom validation to accept or deny the sign up request.
- Pre authentication:** This trigger is invoked when a user submits their information to be authenticated, allowing you to perform custom validations to accept or deny the sign in request.
- Custom message:** This trigger is invoked before a verification or MFA message is sent, allowing you to customize the message dynamically. Note that static custom messages can be edited on the Verifications panel.
- Post authentication:** This trigger is invoked after a user is authenticated, allowing you to add custom logic, for example for analytics.

Triggers are simple lambdas that we can use

The screenshot shows the AWS User Pools console under the Federated Identities tab. The 'Analytics' option is selected and highlighted in orange. A central call-to-action box asks, 'Do you want to add analytics and user campaigns to this user pool?'. It includes a note: 'You can add analytics and create campaigns to engage your users with Amazon Pinpoint. Additional charges apply.' and a 'Learn more' link. Below the box, there is a button labeled 'Add analytics and campaigns'.

- General settings**
- Users and groups
- Attributes
- Policies
- MFA and verifications
- Advanced security beta
- Message customizations
- Tags
- Devices
- App clients
- Triggers**
- Analytics**
- App integration
- App client settings
- Domain name

You can also turn on analytics

Screenshot of the AWS Cognito Federated Identities console showing the creation of a new identity pool named "spacefinder_development_identityPool". The pool currently has 5 identities.

Getting started

- Developer Guide
- Getting started with Android
- Getting started with iOS
- Identity API reference
- Sync API Reference

In-Depth Guides

- Using Cognito to Sync Data
- Using Cognito in Your Website
- Using the Cognito Credentials Provider

Community

- Cognito Developer Forum
- AWS Mobile Blog

Cognito User Pools is your User directory, it stores the users and allows for an easier way to create the sign up and sign in functionalities. **Cognito Federated Identities** is what allows those users that have logged in with Cognito User Pools to get AWS Credentials, you exchange JWT tokens for AWS IAM Role credentials so that those users can access AWS services like S3 buckets, DynamoDB and API Gateways.

Screenshot of the AWS Cognito Federated Identities console showing the dashboard for the "spacefinder_development_identityPool".

Identity pool

- Dashboard
- Sample code
- Identity browser

Identities this month

5

Total identities

5

Cognito Sync helps you sync user data across devices. Get started using the Mobile SDK: Android, iOS

Authentication methods

Method	Percentage	Count
us-east-1_laovdw2c2	120.0%	6
us-east-1_laovdw2c2	20.0%	6

Filters: Total identities ▾ Past 14 days ▾

https://console.aws.amazon.com/cognito/console/identityPoolId/15788ce65-8742-4433-8746-156762702cde

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail Editing spacefinder_develo... JSON Web Tokens - jwt.io API Gateway New Tab

Services Resource Groups Cognito API Gateway jpirate @ aws-serverless-auth-r... N. Virginia Support

Federated Identities spacefinder_development_identityPool

Identity pool

Dashboard

Sample code

Identity browser

Edit identity pool

From this page you can modify the details of your identity pool. An identity pool must have a unique name and a set of authenticated and unauthenticated roles. The roles are saved with your identity pool and whenever we receive a request to authorize a user we will automatically utilize the roles you specify here. You will be required to specify the identity pool id from this page when initializing the Amazon Cognito client SDK. Learn more about using IAM roles with Amazon Cognito.

Identity pool name* spacefinder_development

Identity pool ID us-east-1:5788ce65-8742-4433-8746-156762702bdd (Show ARN)

Unauthenticated role spacefinder-development-s-CognitoIdentityPoolUnAuth-1100XTUZO8KIO Create new role

Authenticated role spacefinder-development-s-CognitoIdentityPoolAuthS-1RL6MJA6B1M1L Create new role

Unauthenticated identities

Waiting for aws.amazon.com...

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail Editing spacefinder_develo... JSON Web Tokens - jwt.io API Gateway New Tab

Unauthenticated identities

Authenticated role spacefinder-development-s-CognitoIdentityPoolAuthS-1RL6MJA6B1M1L Create new role

Unauthenticated identities

Authentication providers

Push synchronization

Cognito Streams

Cognito Events

Delete identity pool

The screenshot shows the AWS Cognito Identity Pool configuration interface. At the top, there's a navigation bar with tabs like 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', 'Window', 'Help'. Below the navigation bar, the URL is https://console.aws.amazon.com/cognito/pool/edit/?region=us-east-1&id=us-east-1:5788ce65-8742-4433-8746-15876. The main content area has a heading 'Authentication providers' with a minus sign icon. A note below it says: 'Amazon Cognito supports the following authentication methods with Amazon Cognito Sign-In or any public provider. If you allow your users to authenticate using any of these public providers, you can specify your application identifiers here. Warning: Changing the application ID that your identity pool is linked to will prevent existing users from authenticating using Amazon Cognito. Learn more about public identity providers.' Below this, there are tabs for 'Cognito', 'Amazon', 'Facebook', 'Google+', 'Twitter / Digits', 'OpenID', 'SAML', and 'Custom'. A note below the tabs says: 'Configure your Cognito Identity Pool to accept users federated with your Cognito User Pool by supplying the User Pool ID and the App Client ID.' There are two input fields: 'User Pool ID' with value 'us-east-1_laovdw2o2' and 'App client id' with value '3ef4ne8839uove8d8acfbass7'. Below these fields is a section titled 'Authenticated role selection' with a note: 'By default the authenticated role defined above will be applied to authenticated users, or you can select a role through rules or for this authentication provider. The rules are applied in order they are saved. They can be reordered by dragging and rearranging the rule order. If multiple roles are available for a user, your app can specify the role with the CustomRoleARN parameter. Learn more.' A button labeled 'Choose role from token' is present. The entire interface is framed by a light gray border.

What you see here is a list of available providers that you can integrate within Federated Identities. This means that you can use Cognito to allow your users to sign up and sign in, but you can also use Facebook, Google, OpenID providers too. You can use a Custom login too.

This screenshot shows the same AWS Cognito Identity Pool configuration page as the previous one, but with a different focus on 'Role resolution'. The 'Authenticated role selection' section is still visible at the top. Below it, a note says: 'If no roles are specified in the token, the role resolution will be invoked. By default, it will fall back to the default role specified for this Identity Pool. You can also choose to DENY the request.' Under 'Role resolution', there are two options: 'Use default Authenticated role' (which is selected) and 'DENY'. The 'DENY' option is highlighted with a dark gray background. At the bottom of the page, there's another 'User Pool ID' field with value 'us-east-1_laovdw2o2' and an 'Unlock' button. The entire interface is framed by a light gray border.

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail Editing spacefinder_devel... JSON Web Tokens - jwt.io API Gateway New Tab

https://console.aws.amazon.com/cognito/pool/edit/?region=us-east-1&id=us-east-1:5788ce65-8742-4433-8746-15676

Search

Authentication providers

Amazon Cognito supports the following authentication methods with Amazon Cognito Sign-In or any public provider. If you allow your users to authenticate using any of these public providers, you can specify your application identifiers here. Warning: Changing the application ID that your identity pool is linked to will prevent existing users from authenticating using Amazon Cognito. Learn more about public identity providers.

Cognito Amazon Facebook Google+ Twitter / Digits OpenID SAML Custom

Facebook App ID Optional Example: 7346241598935555

Authenticated role selection

By default the authenticated role defined above will be applied to authenticated users, or you can select a role through rules or for this authentication provider. The rules are applied in order they are saved. They can be reordered by dragging and rearranging the rule order. If multiple roles are available for a user, your app can specify the role with the CustomRoleARN parameter. Learn more..

Use default role

Push synchronization

Cognito Streams

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail Editing spacefinder_devel... JSON Web Tokens - jwt.io API Gateway New Tab

https://console.aws.amazon.com/cognito/pool/edit/?region=us-east-1&id=us-east-1:5788ce65-8742-4433-8746-15676

Search

Authentication providers

Amazon Cognito supports the following authentication methods with Amazon Cognito Sign-In or any public provider. If you allow your users to authenticate using any of these public providers, you can specify your application identifiers here. Warning: Changing the application ID that your identity pool is linked to will prevent existing users from authenticating using Amazon Cognito. Learn more about public identity providers.

Cognito Amazon Facebook Google+ Twitter / Digits OpenID SAML Custom

Twitter Consumer Key Optional Example: xvz1evFS4wEEPTGEFPHBog

Twitter Consumer Secret Optional Example: kAcSOqF21Fu85e7zjz7ZN2U4ZRhfV3WpwPAoE3Z7kBw

Authenticated role selection

By default the authenticated role defined above will be applied to authenticated users, or you can select a role through rules or for this authentication provider. The rules are applied in order they are saved. They can be reordered by dragging and rearranging the rule order. If multiple roles are available for a user, your app can specify the role with the CustomRoleARN parameter. Learn more..

Use default role

Push synchronization

Application so far...



Application so far...



Now we are ready for the business logic. We can use API Gateway, Lambda, and DynamoDB as below. You are able to offload all the authentication and authorization decisions of your business logic to the API Gateway



SpaceFinder API



Admin only	POST	/locations
	GET	/locations
	GET	/locations/{locationId}
Admin only	DELETE	/locations/{locationId}
	GET	/locations/{locationId}/resources
Admin only	POST	/locations/{locationId}/resources
Admin only	DELETE	/locations/{locationId}/resources/{resourceId}
	GET	/locations/{locationId}/resources/{resourceId}/bookings
	GET	/users/{userId}/bookings
	POST	/users/{userId}/bookings
	DELETE	/users/{userId}/bookings/{bookingId}



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Within the API, we have admin only operations as above. We want only admin users of our User pool to be able to call and get responses from those admin-only APIs.

API Gateway: three types of authorization

Amazon Cognito User Pools

User Pools Authorizers

Amazon Cognito Federated Identities

AWS IAM authorization

Custom Identity Providers

Custom Authorizers



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



There are 3 different types of authorization options available.

API Gateway: three types of authorization

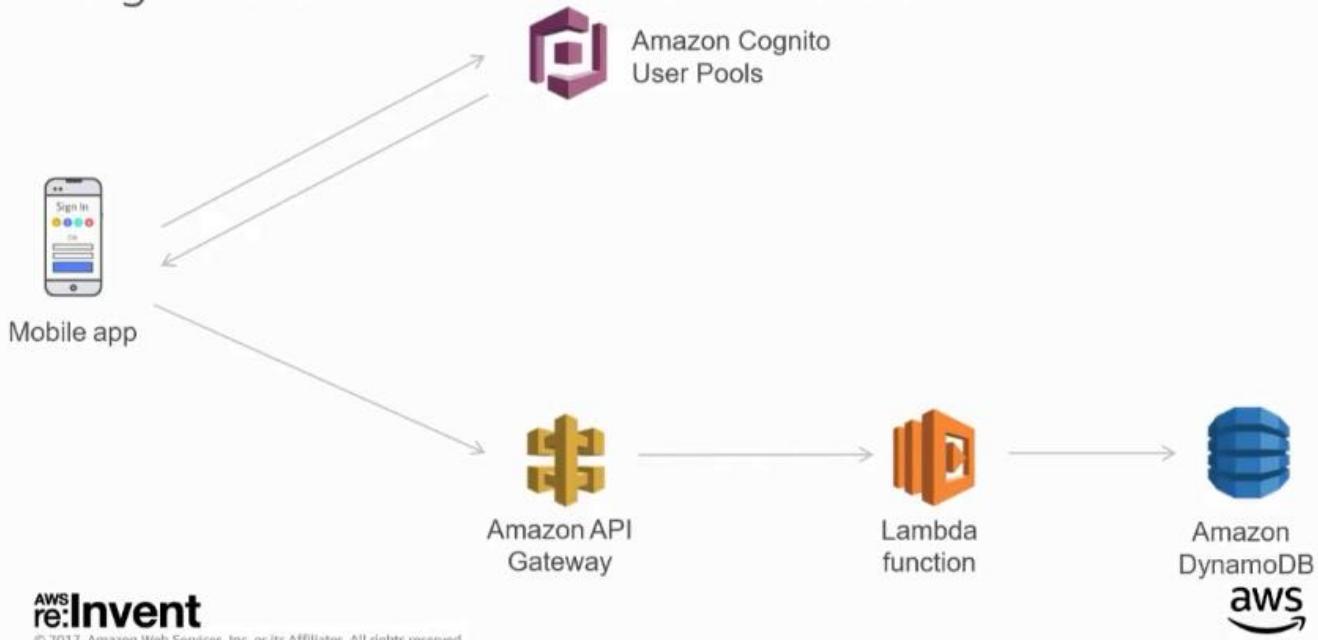


aws re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



This type can only check if a user belongs to our User Pool, it is not going to be able to differentiate our admin users from ordinary users.

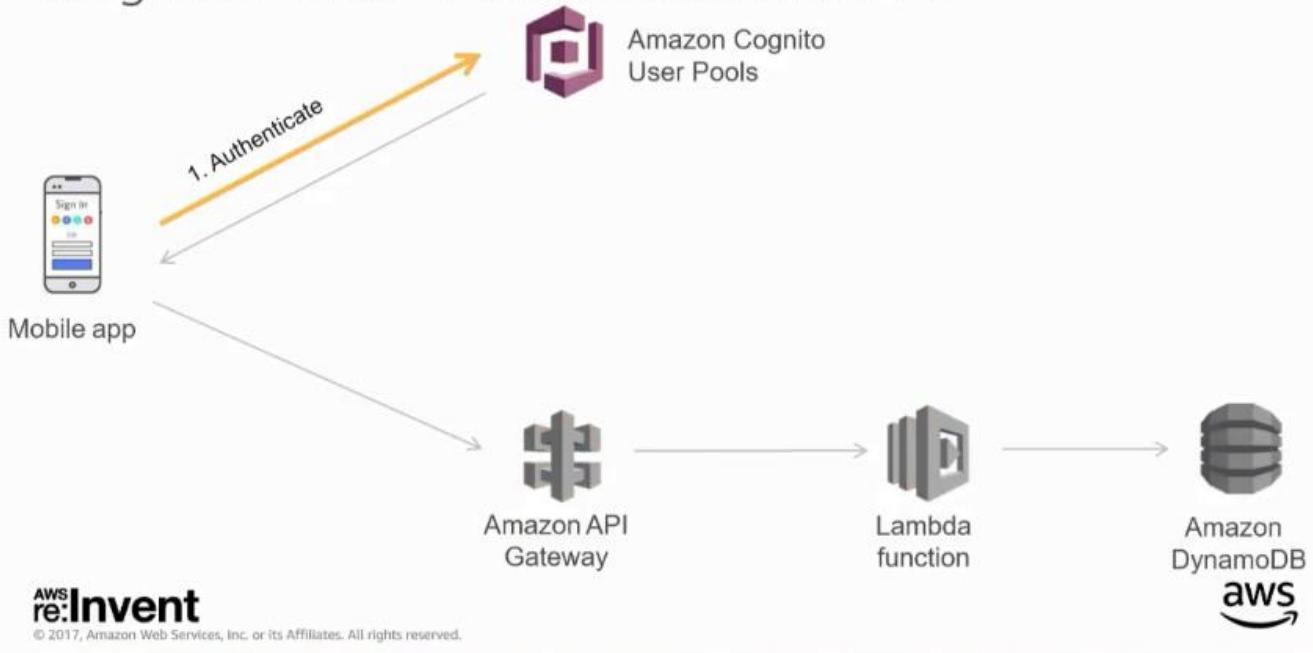
Cognito User Pools Authorizers



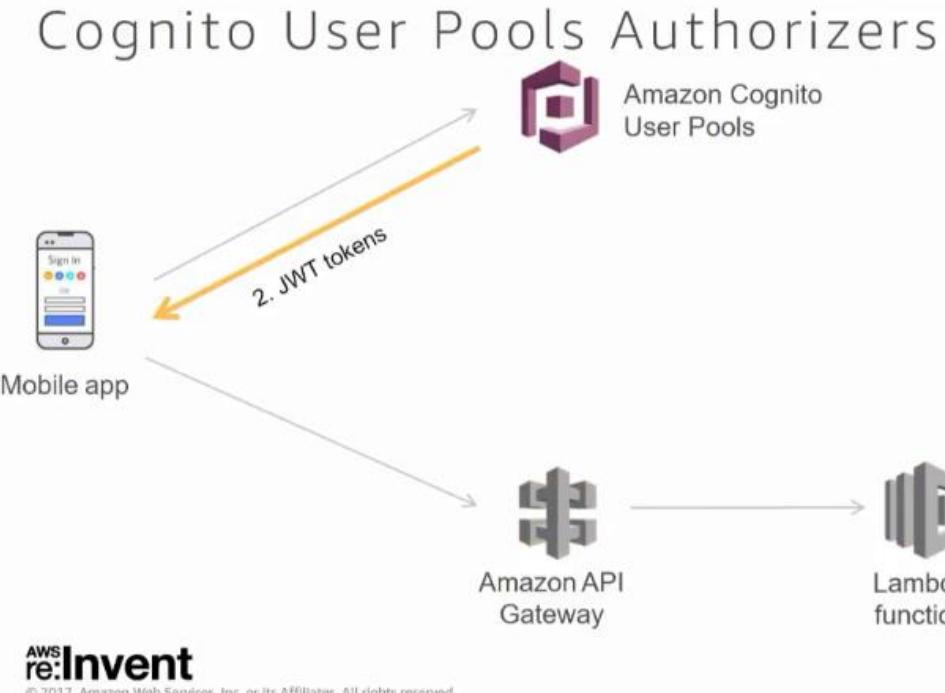
aws re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

If you have users that are using Cognito User Pools, this can work as described below

Cognito User Pools Authorizers

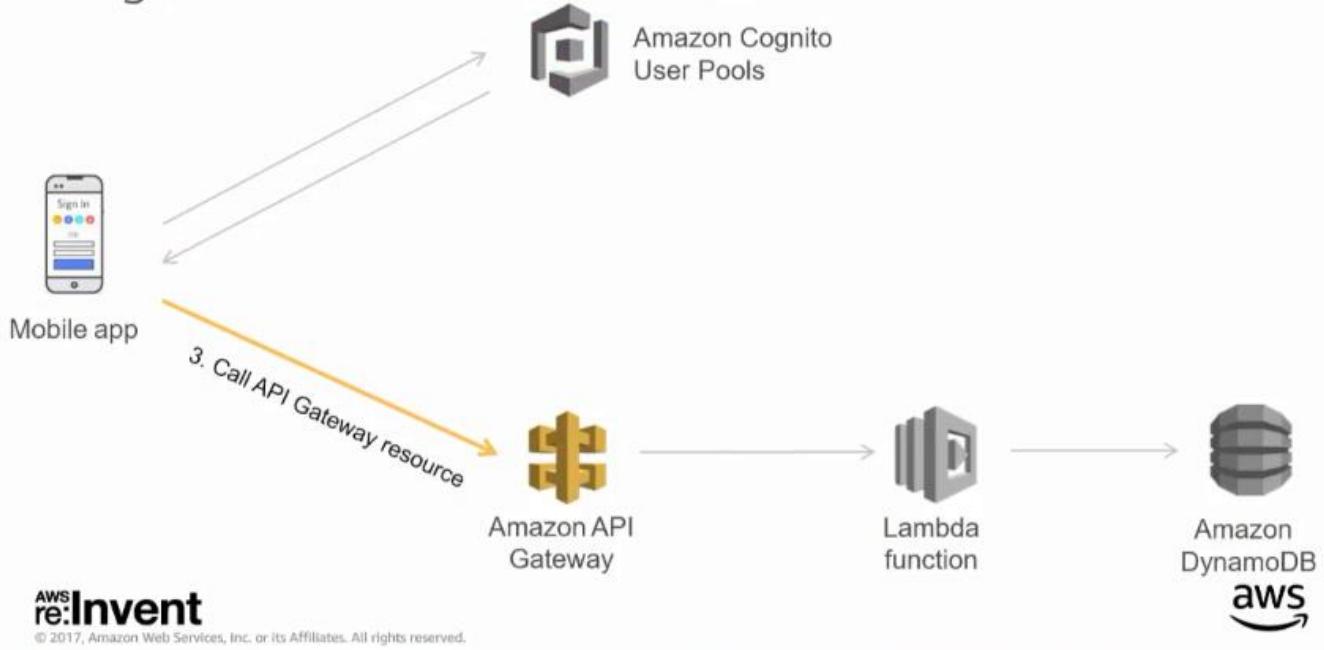


You first have those users authenticate with Cognito User Pools,



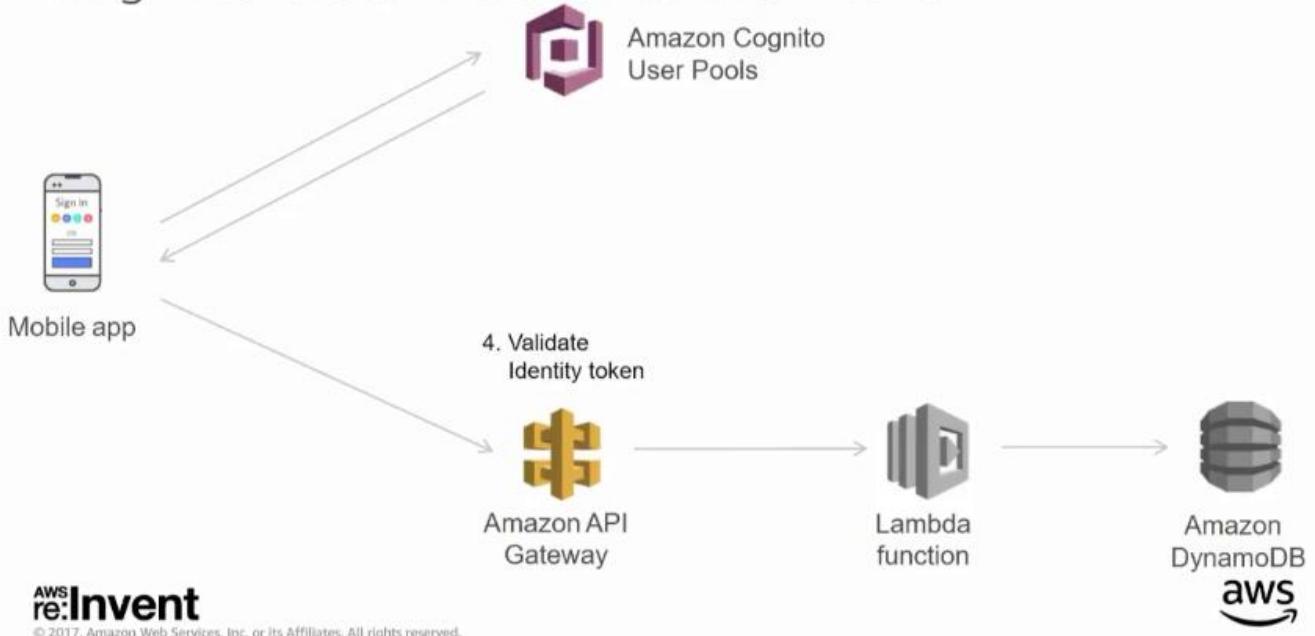
They then get back a set of JWT tokens

Cognito User Pools Authorizers



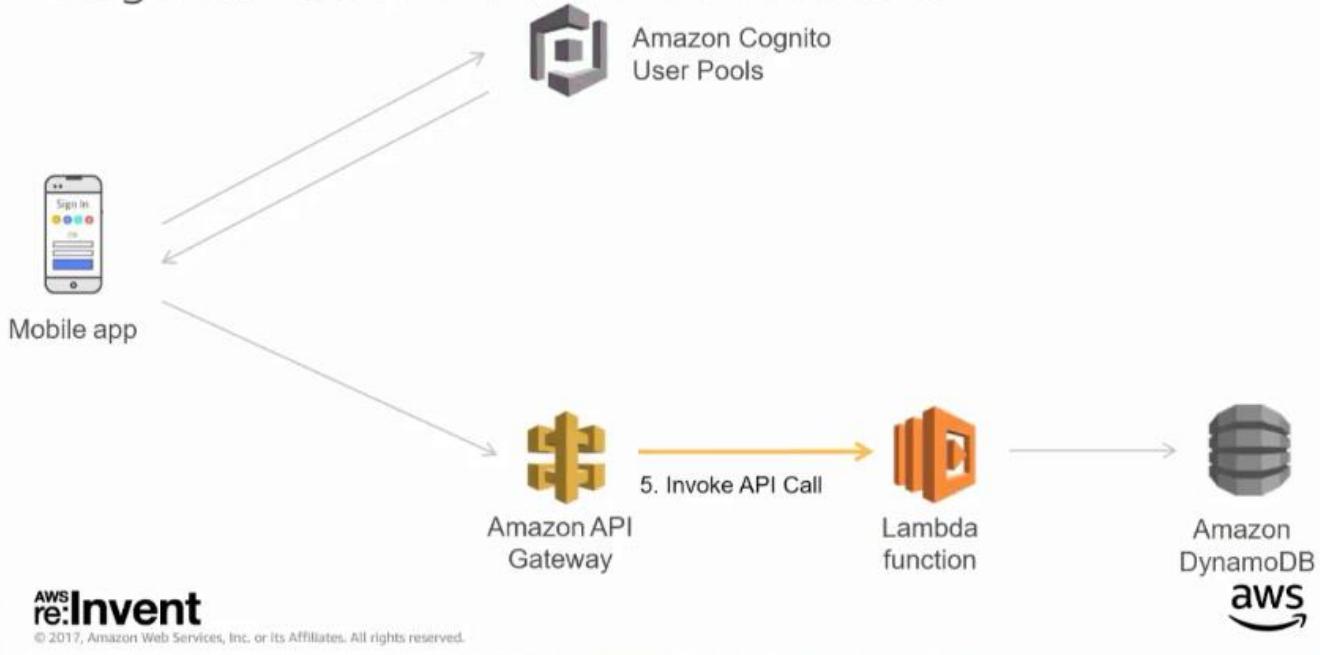
You then take the Identity token and sent it to API Gateway with the header of your choice in your request, most likely in the authorization header.

Cognito User Pools Authorizers



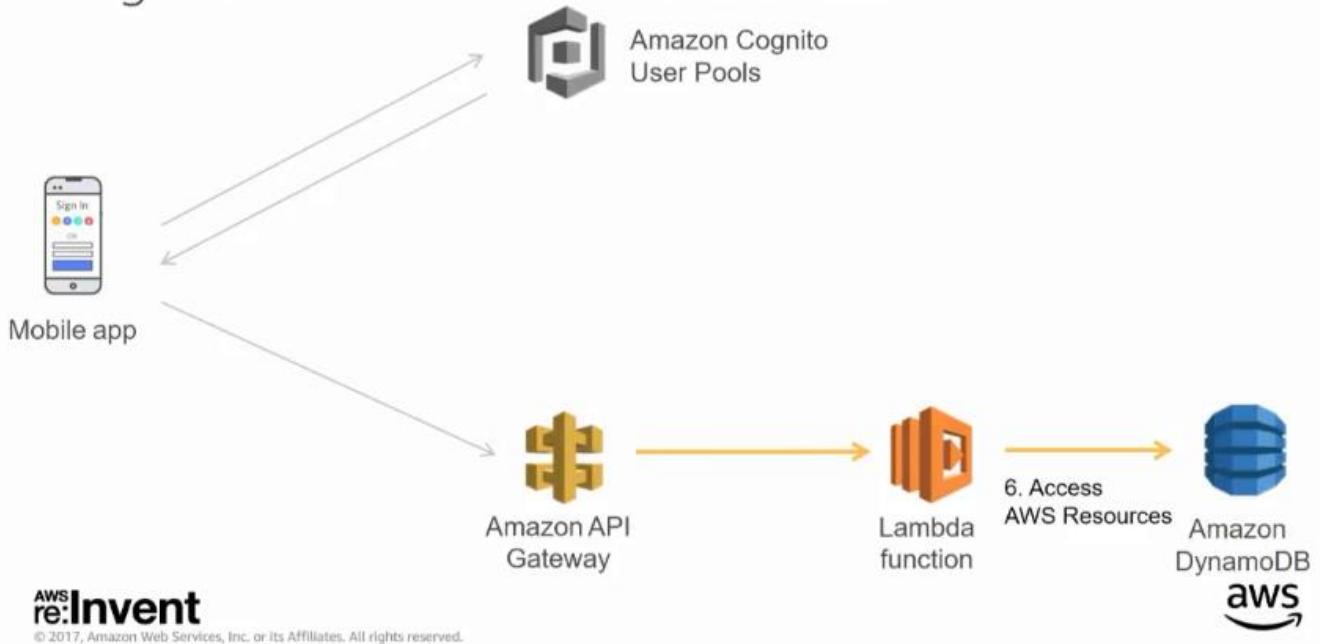
API Gateway can then read the token, check its signature, check if the token is still valid since an Identity token is only good for 1 hour,

Cognito User Pools Authorizers



Only if the Identity token is valid will the API Gateway go ahead and invoke the lambda function of interest.

Cognito User Pools Authorizers



The lambda can then use the IAM execution role (specific to each lambda function) that allows it to talk to other AWS resources like DynamoDB.

API Gateway: three types of authorization

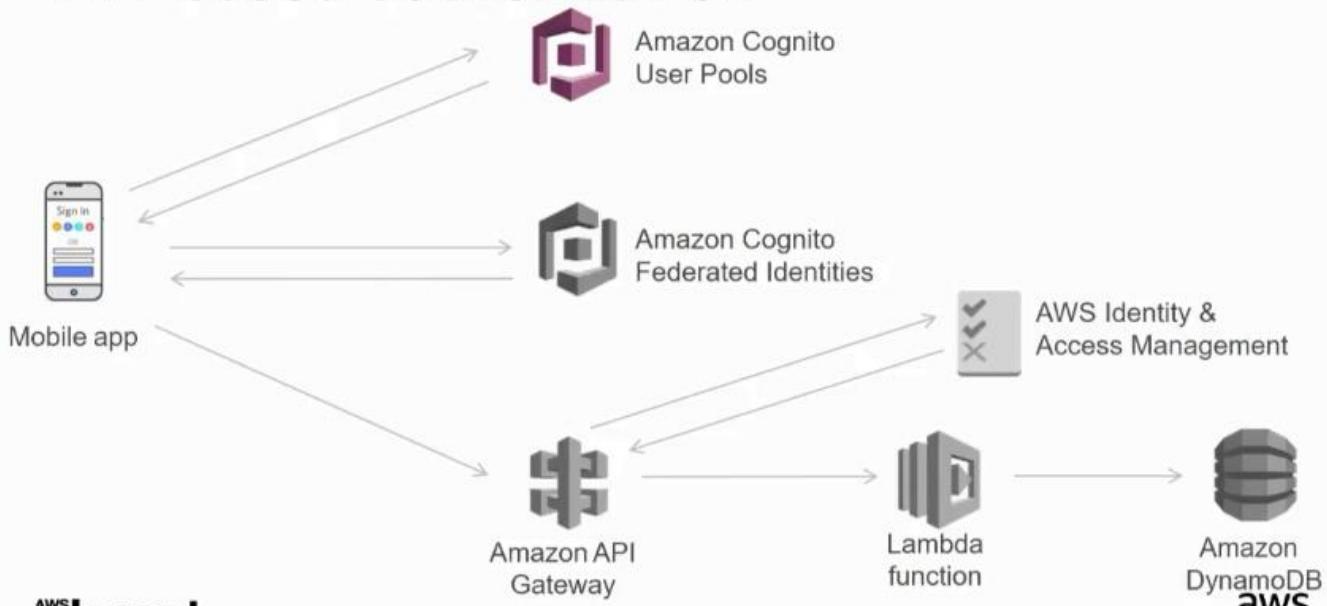


aws re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



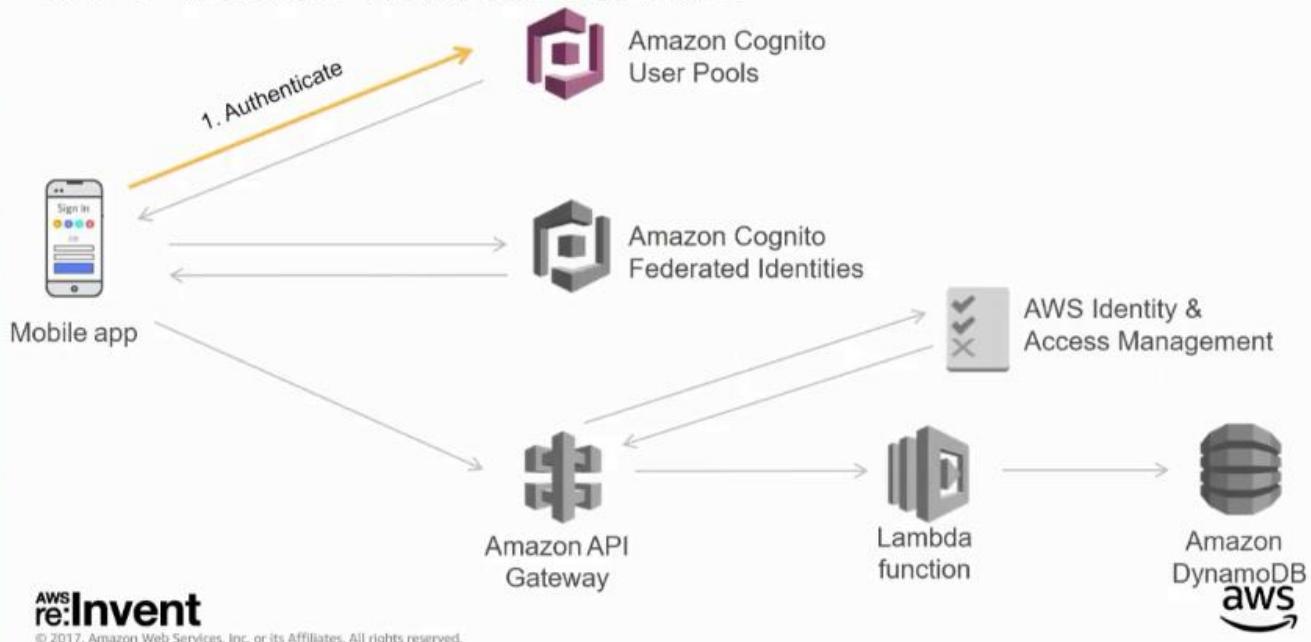
AWS IAM based authorization option

IAM-based authorization

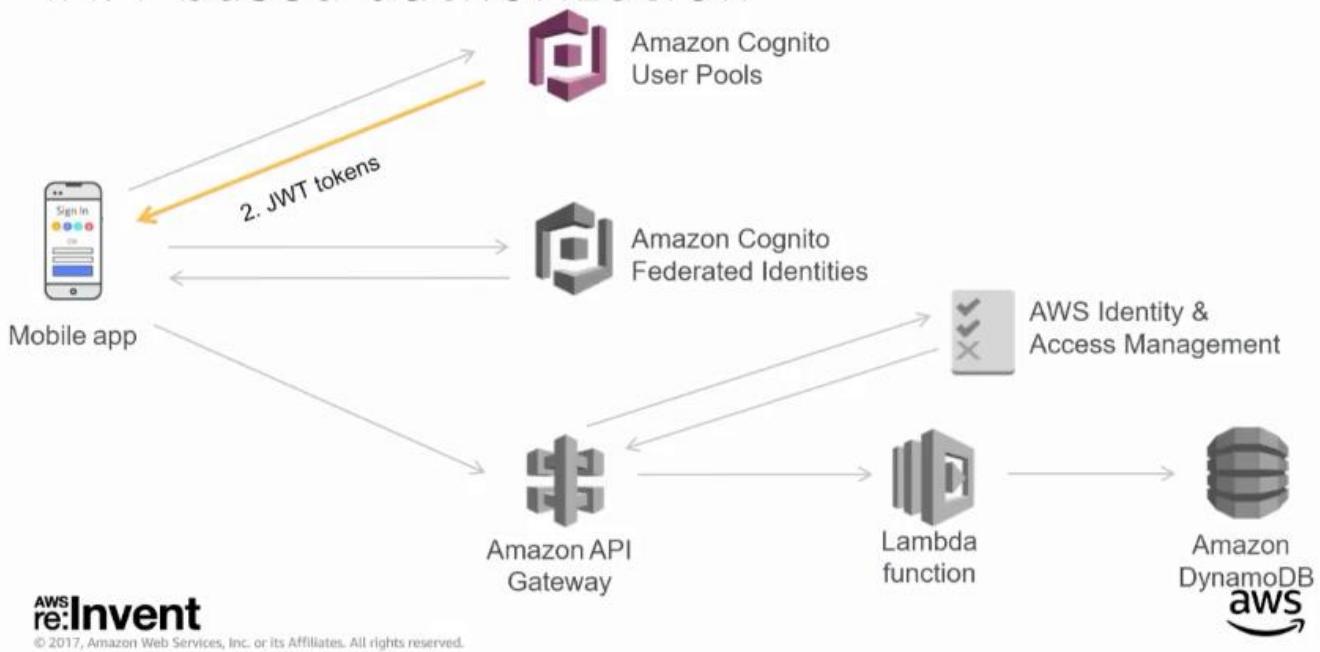


aws re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

IAM-based authorization



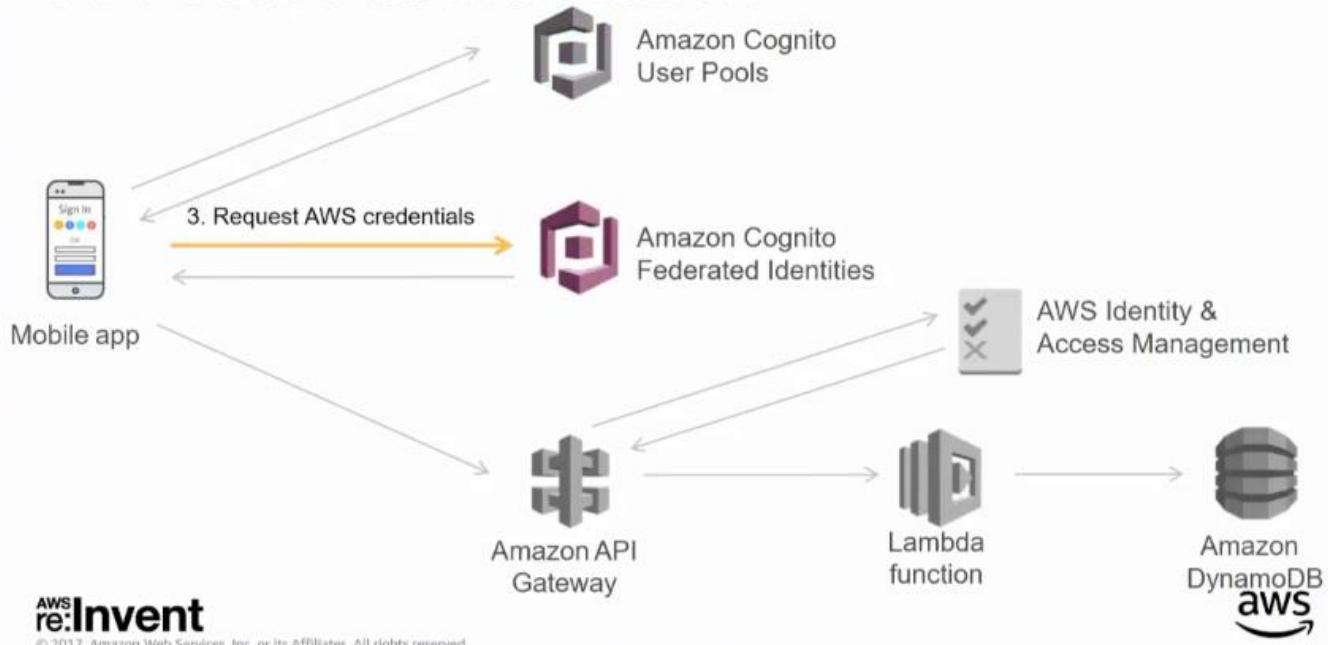
IAM-based authorization



AWS re:Invent

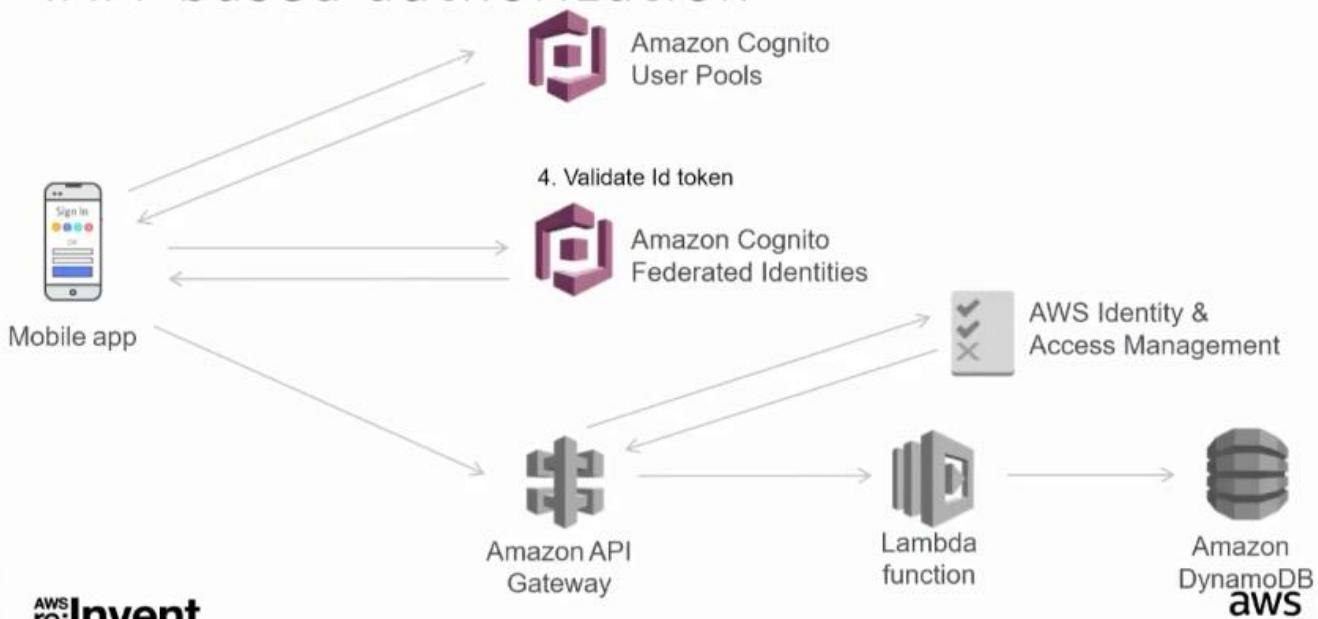
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

IAM-based authorization



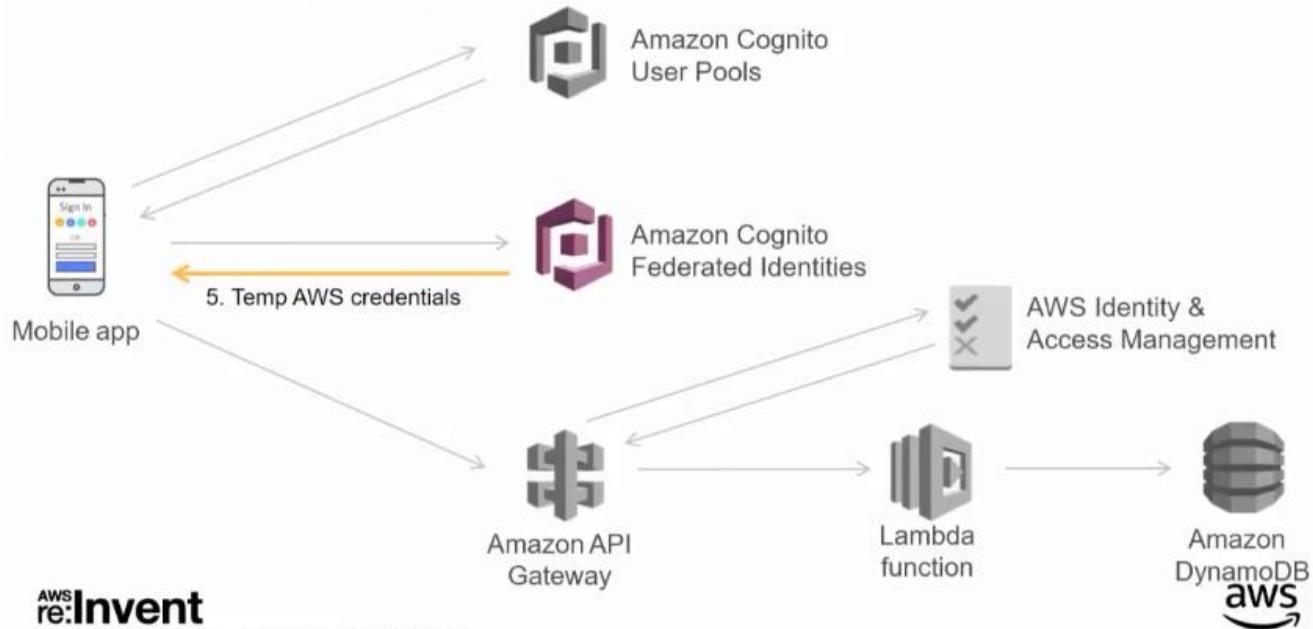
You then get your IAM Role by requesting your credentials,

IAM-based authorization

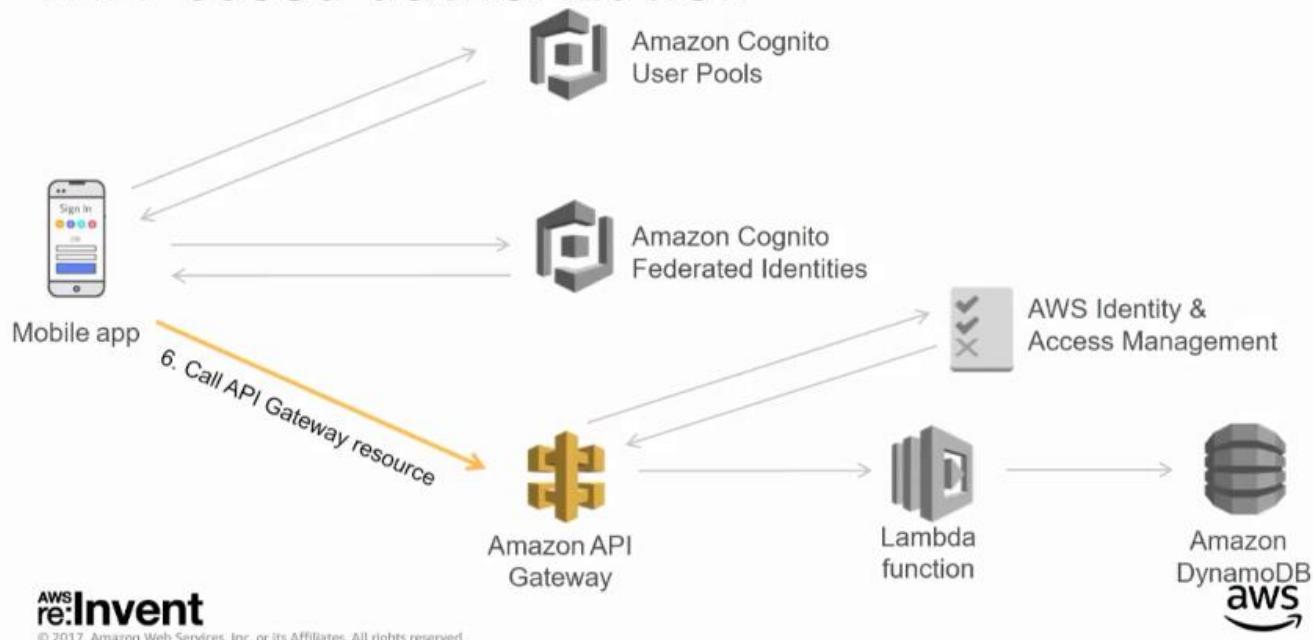


Cognito takes care of the token validation behind the scenes

IAM-based authorization

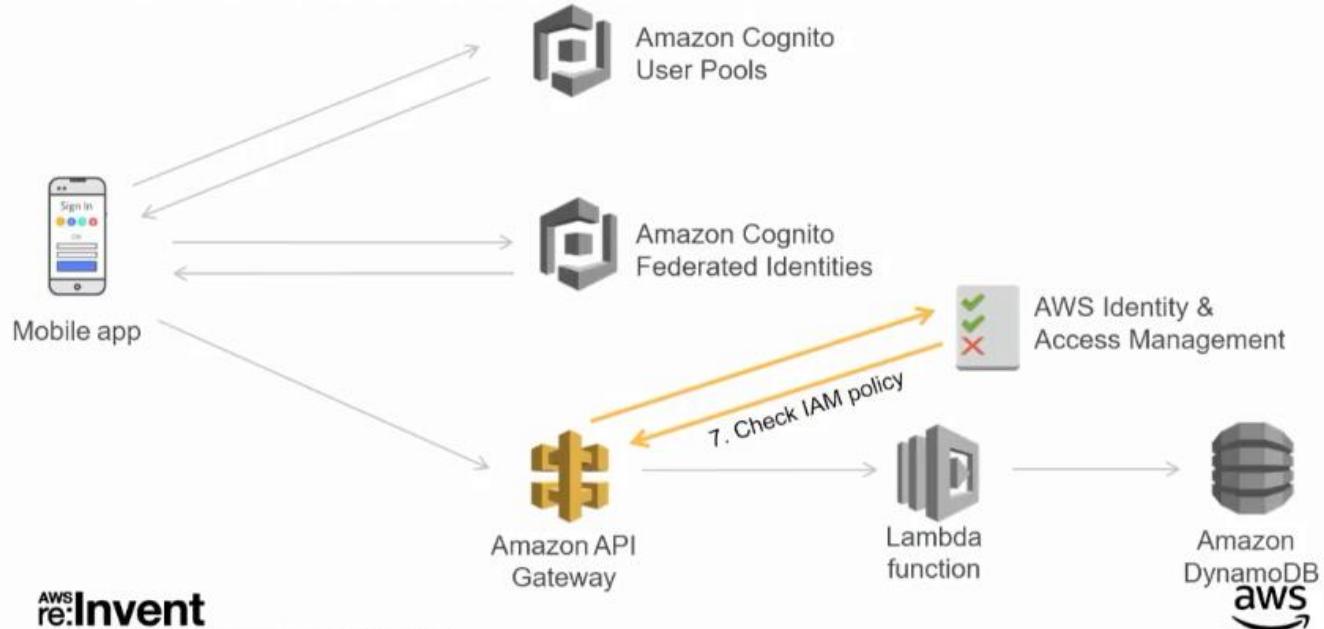


IAM-based authorization



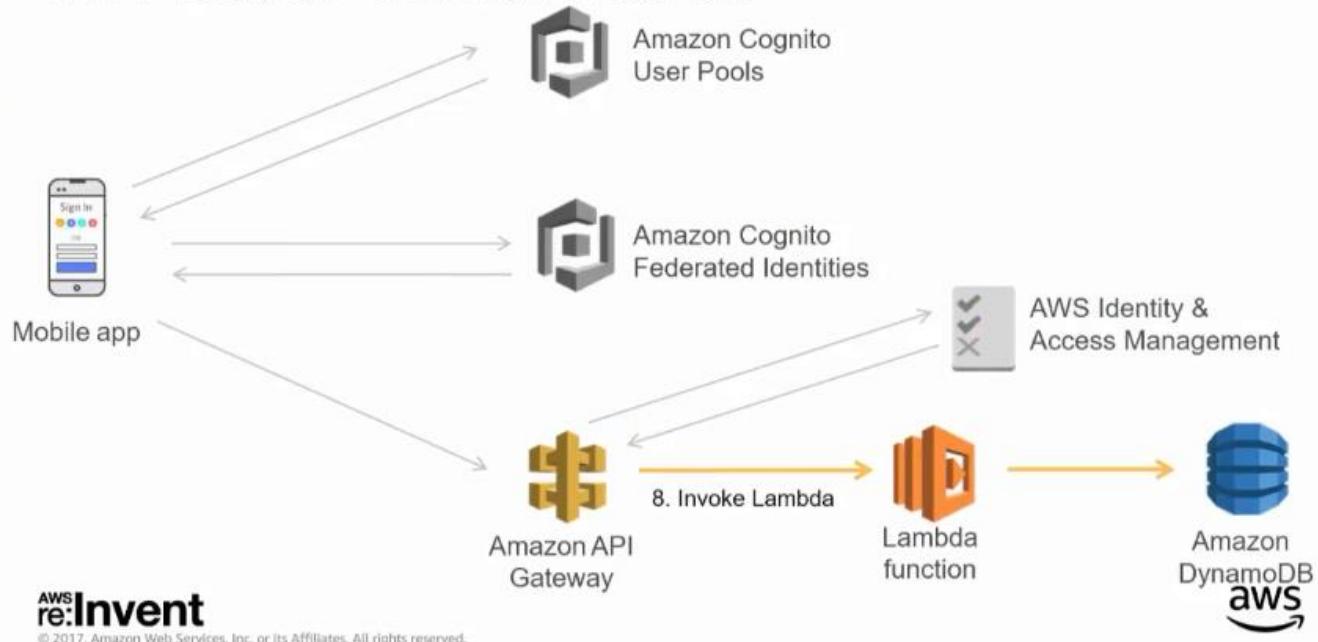
You can then send your requests to PAI Gateway, signing it with a process called 'signature-before'.

IAM-based authorization



API Gateway is able to extract from the signature which role is used for signing the request, and it will check what policies that role is associated with.

IAM-based authorization



Based on the policy, API Gateway is able to determine if the user can perform the requested operation.

IAM Policy Detail

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": "execute-api:Invoke",  
            "Effect": "Allow",  
            "Resource": "arn:aws:execute-api:*:*:ff5h9tpwf/*"  
        },  
        {  
            "Action": "execute-api:Invoke",  
            "Effect": "Deny",  
            "Resource": "arn:aws:execute-api:*:*:ff5h9tpwf/*/POST/locations/*"  
        }  
    ]  
}
```



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



This is a simple IAM Policy, this policy allows the user to access all operations on the API **except** the specified POST on the specified path.

API Gateway: three types of authorization

Amazon Cognito
User Pools

User Pools Authorizers

Amazon Cognito
Federated Identities

AWS IAM authorization

Custom Identity Providers

Custom Authorizers



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

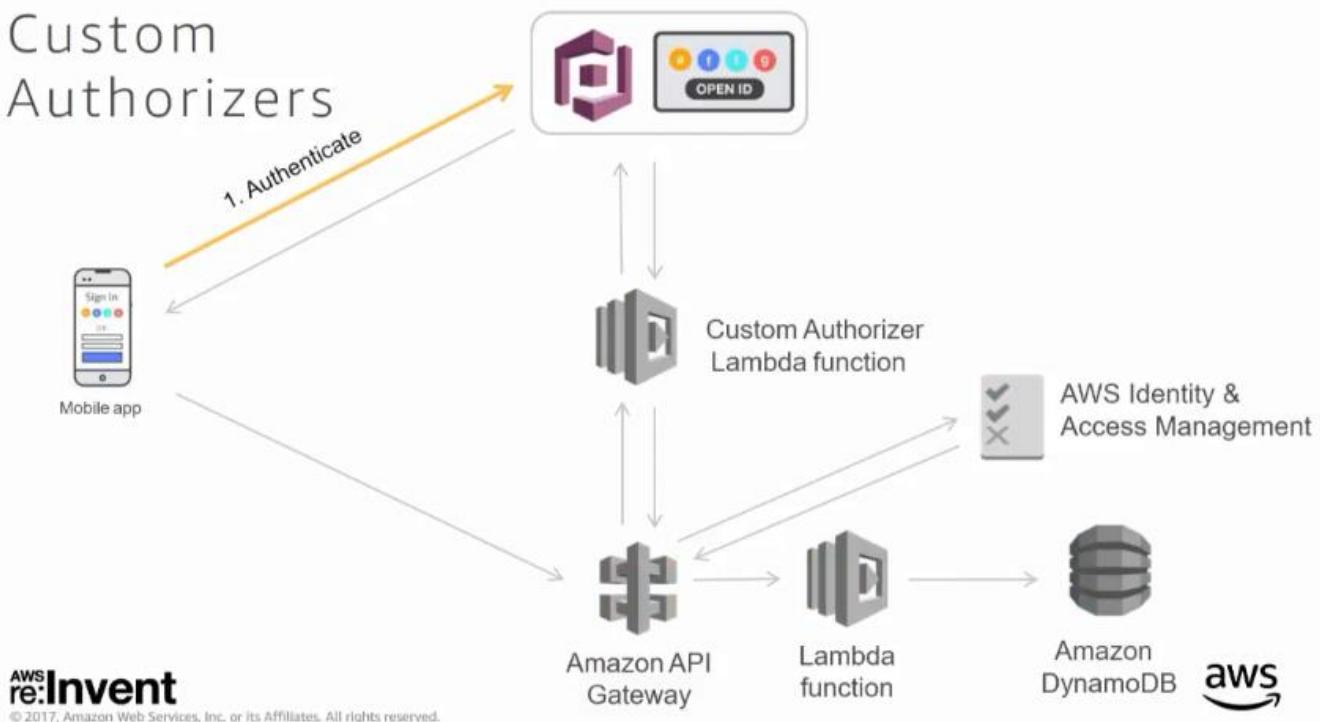


The Custom Authorizer option uses lambda functions

Custom Authorizers

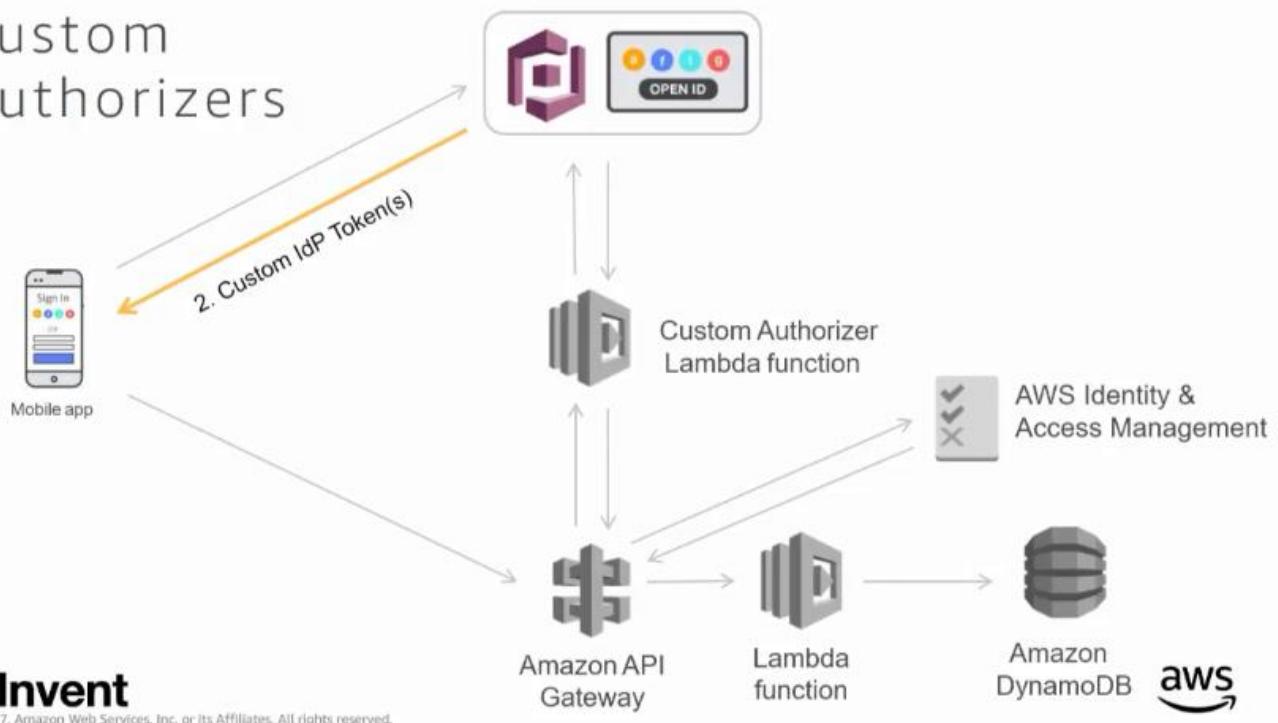


Custom Authorizers

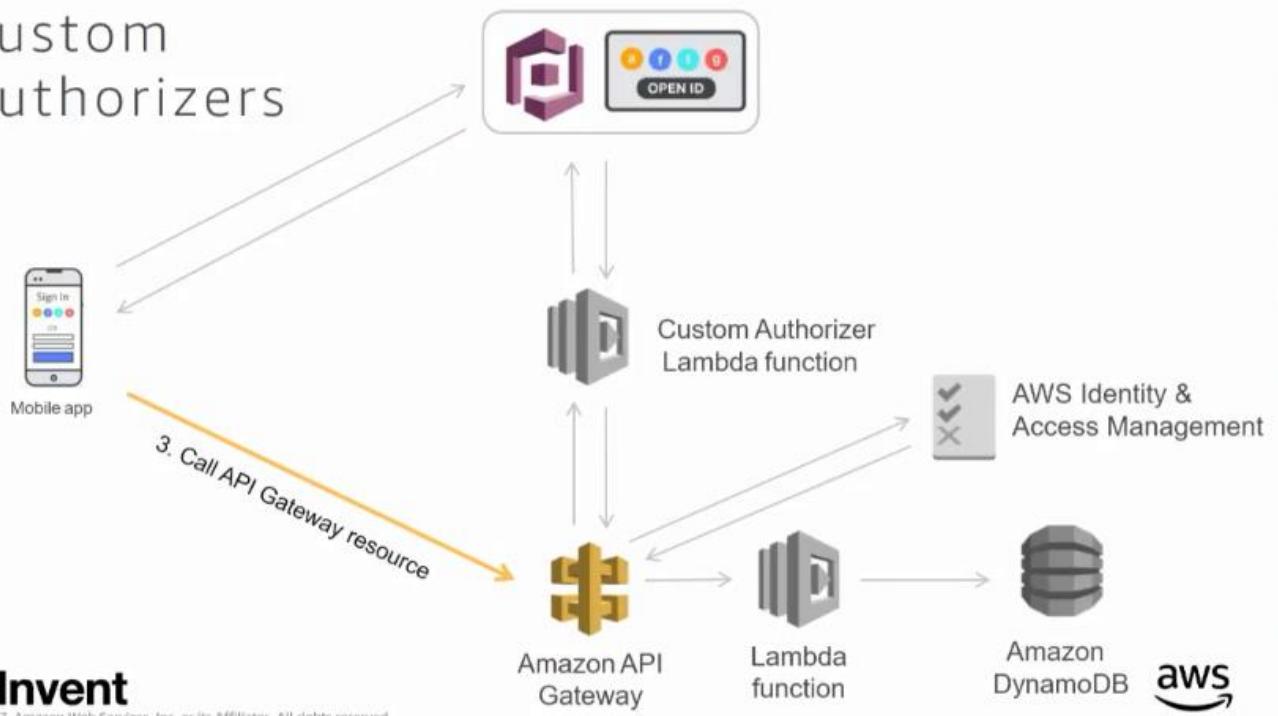


First you authenticate with your identity provider of choice, you don't have to use Cognito at all, as long as you have an identity provider that can provide a bearer token for you.

Custom Authorizers



Custom Authorizers



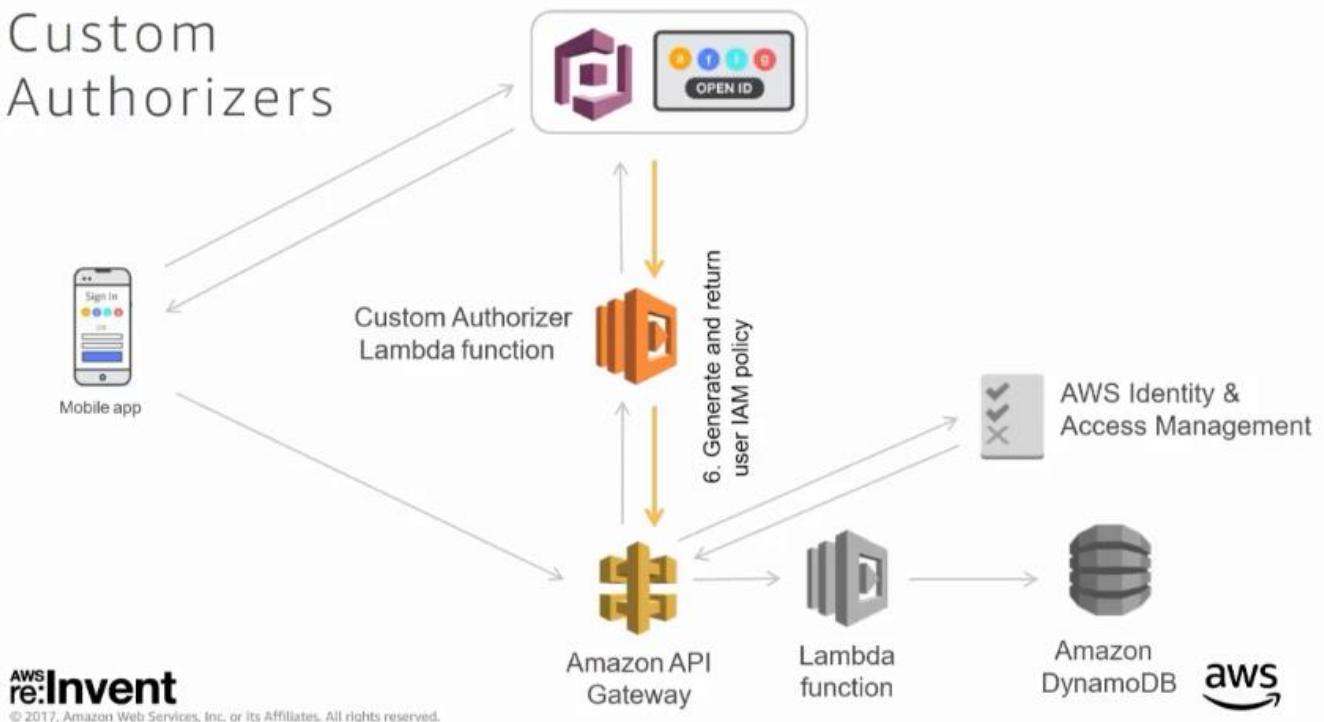
You can then send the token along to API Gateway in the header of your choice. API Gateway checks its local cache to see for that token if we have a known valid policy defined for the user via IAM credentials.

Custom Authorizers



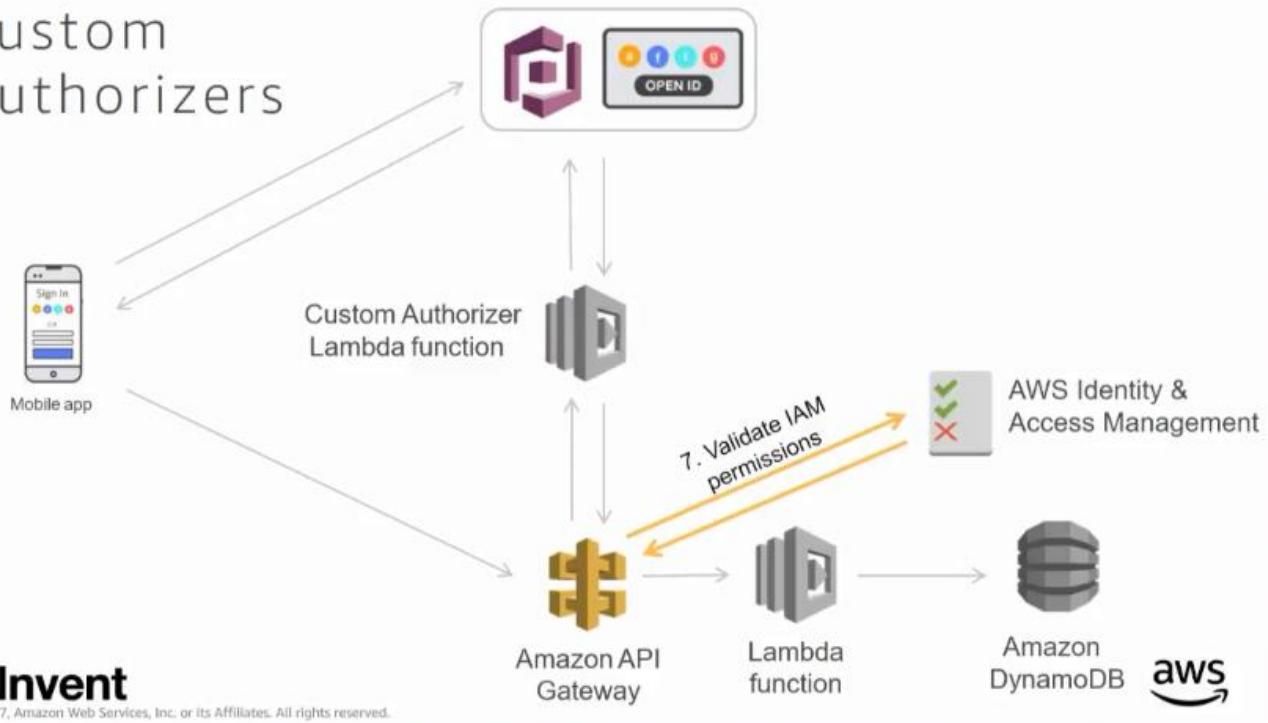
If the API Gateway does not know who the user is, it then invokes a lambda function (your custom authorizer lambda function) to run any business logic of your choice to talk to the 3rd party IDP to validate the user and determine what access they should have.

Custom Authorizers



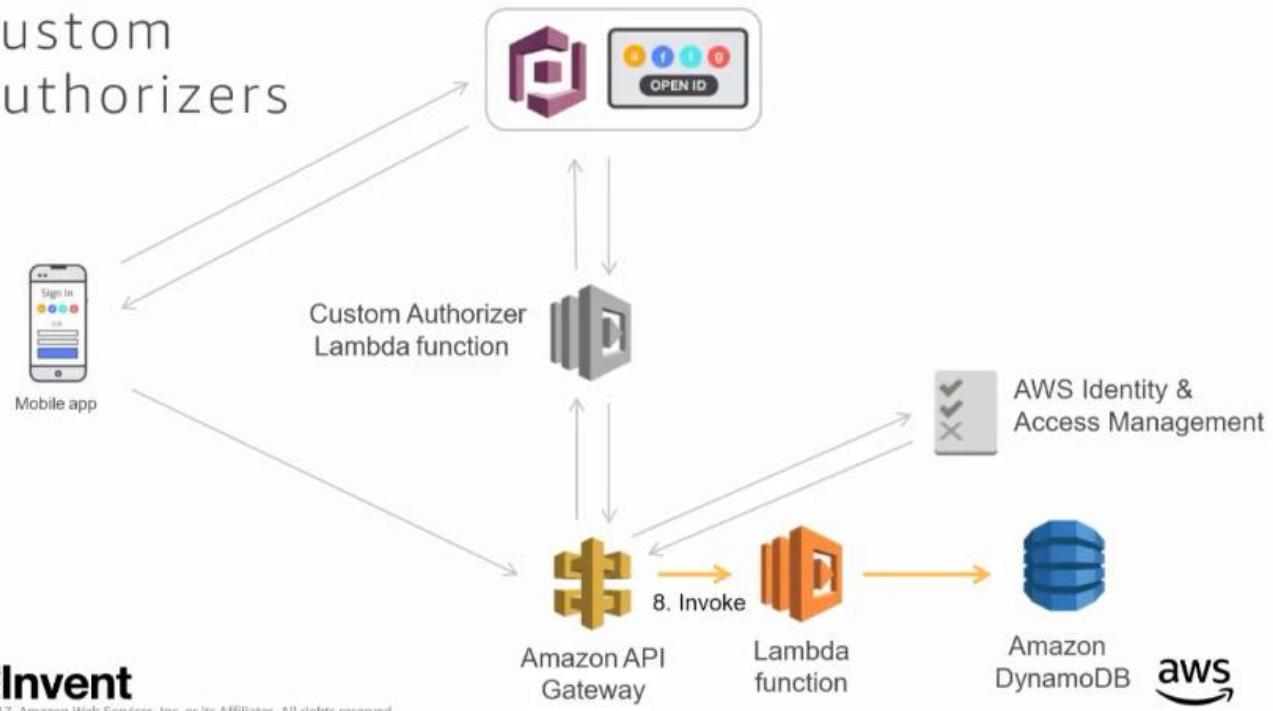
After authorization, the custom authorizer then sends back a well-formed IAM Policy back to the user. The policy is then cached locally for that user for subsequent calls, you can configure the cache duration and validity for the cached policy (the default cache duration is 5 minutes)

Custom Authorizers



API Gateway then checks what the policy is allowed to do and compares with the intended request

Custom Authorizers



Custom Authorizer Lambda function

Sample Code

```
var testPolicy = new AuthPolicy("userIdentifier", "XXXXXXXXXXXXXX", apiOptions);  
  
testPolicy.allowMethod(AuthPolicy.HttpVerb.POST, "/locations/*");  
testPolicy.allowMethod(AuthPolicy.HttpVerb.DELETE, "/locations/*");  
  
callback(null, testPolicy.getPolicy());
```



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



We have a lambda function blueprint in python and JavaScript that can generate a well-formed policy.

API Gateway: three types of authorization

Amazon Cognito
User Pools



User Pools Authorizers

Amazon Cognito
Federated Identities



AWS IAM authorization

Custom Identity Providers



Custom Authorizers



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



IAM based authorization is the most secure form of authorization since it uses request signing that makes the requests unique with every call.

DEMO

The screenshot shows the AWS API Gateway console in a Firefox browser. The URL is <https://console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/ye07xoo3oc/resources/md8nfinwi7>. The left sidebar shows the navigation menu with 'APIs' selected. The main area displays a tree view of a resource structure under the path `/`:

- `/locations`: GET, OPTIONS, POST
- `/[locationId]`: DELETE, GET, OPTIONS
- `/resources`: GET, OPTIONS, POST
- `/[resourceId]`: DELETE, GET, OPTIONS

A message at the bottom right says "No methods defined for the resource."

This screenshot is similar to the one above, but it highlights the `DELETE` method for the `/[resourceId]` endpoint. The URL is the same: <https://console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/ye07xoo3oc/resources/md8nfinwi7>. The highlighted `DELETE` method is part of the `/[resourceId]` node.

The screenshot shows the AWS API Gateway console. The left sidebar lists the Spacefinder-API resources, including the /locations resource with its GET method selected. The main area displays the Method Request, Integration Request, Method Response, and Integration Response stages. The Method Request stage shows the authorization details: Auth: spacefinder-userPool-authorizer and ARN: arn:aws:execute-api:us-east-1:742360508337:ye07xoo3o/GET/locations. The Integration Request stage indicates the Type: LAMBDA_PROXY. The Method Response stage is set to "Select an integration response." The Integration Response stage notes that "Proxy integrations cannot be configured to transform responses." A vertical sidebar on the right shows the Lambda function name: Lambda spacefinder-development-locations.

With each API endpoint, we are able to define particular authorization details needed for the request.

The screenshot shows the AWS API Gateway console with the Method Request configuration for the /locations - GET endpoint. The Settings section includes Authorization: spacefinder-userPool-authorizer, Request Validator: NONE, and API Key Required: false. There are sections for URL Query String Parameters, HTTP Request Headers, and Request Body.

We are requiring authorization for this particular request

The screenshot shows the AWS API Gateway console. The left sidebar lists the API, Resources, Stages, Authorizers, and other API-related settings. The main area shows the method execution settings for the GET request on the /locations endpoint. The 'Authorization' dropdown menu is open, showing options like NONE, AWS_IAM, Token authorizer, API Key Required, spacefinder-custom-authorizer, Cognito user pool authorizers, and spacefinder-userPool-authorizer. The 'spacefinder-userPool-authorizer' option is highlighted with a blue box.

We can pick any type of authorization that we want that API request.

The screenshot shows the AWS API Gateway console with the 'Authorizers' section selected in the sidebar. It displays two existing authorizers: 'spacefinder-userPool-authorizer' and 'spacefinder-custom-authorizer'. Each authorizer has its configuration details listed, such as the type (Cognito User Pool or Lambda Function), token source, and validation settings. Buttons for 'Edit' and 'Test' are visible at the bottom of each authorizer's card.

We can also define our authorizers as above.

The screenshot shows a web browser window with the following details:

- Top Bar:** Chrome, File, Edit, View, History, Bookmarks, People, Window, Help.
- Developer Tools:** Opened under the "Tools" menu, showing the "Console" tab. The console output shows several log entries related to Cognito User Pools Identity Token and Access Token.
- User Profile Page:**
 - My Account:** Shows a profile picture of a group of people.
 - Your profile image:** A placeholder for a profile picture.
 - Change password:** A note about security best practices and a "CHANGE PASSWORD" button.
 - View Admin features?** A note about being logged in without Admin privileges and a "View Admin features (disabled)" link.
- Bottom Bar:** Welcome, Resources, Booking, Account.

We can also take a token as above and pass it to the custom authorizer as below

The screenshot shows the AWS API Gateway interface with the following details:

- Left Sidebar:** APIs, Spacefinder-API, Resources, Stages, Authorizers (selected), Gateway Responses, Models, Documentation, Binary Support, Dashboard, Usage Plans, API Keys, Custom Domain Names, Client Certificates.
- Authorizers Tab:** Shows "spacefinder-custom-authorizer - Test Authorizer".
- Test Token Form:**
 - Authorization (header): `i2xked10b6VwdBMNFlyZzgMRI0UhVJ_zFn95WZcHfxPeUlsav_nlbHrQZGg`
 - Test button
- Right Panel:**
 - Custom (us-east-1)
 - Token Validation: none
 - Authorization Caching: Authorization cached for 5 minutes
- Bottom:** Edit, Test buttons for the authorizer.

The screenshot shows the AWS API Gateway console. On the left, a sidebar lists various services: APIs, Spacefinder-API, Resources, Stages, Authorizers (selected), Gateway Responses, Models, Documentation, Binary Support, Dashboard, Usage Plans, API Keys, Custom Domain Names, and Client Certificates. The main content area is titled "Authorize" and shows a modal dialog for a "spacefinder-custom-authorizer - Test Authorizer". The modal displays the following JSON policy:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": "execute-api:Invoke",  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:execute-api:us-east-1:742360508337:ye07xoo3oc/n  
            ]  
        },  
        {  
            "Action": "execute-api:Invoke",  
            "Effect": "Deny",  
            "Resource": [  
                "arn:aws:execute-api:us-east-1:742360508337:ye07xoo3oc/n  
            ]  
        }  
    ]  
}
```

The modal also shows "Response Code: 200", "Latency 49", and "Policy" sections. On the right, there's a preview pane showing "Test" results. At the bottom of the modal is a "Close" button.

By running the custom authorizer, we can then see what the token is authorized to do

The screenshot shows a mobile browser developer tools console. The left panel displays a user profile screen with a "My Account" section, a "Your profile image" placeholder, a "SELECT IMAGE" button, a "Change password" section, and a "View Admin features?" section. The right panel shows the developer tools interface with tabs for Elements, Console, Sources, Network, Performance, Memory, Application, Security, and Audits. The "Console" tab is selected, showing the output "Default levels". Below the tabs, there are buttons for "top", "Filter", and "Default levels". The bottom of the screen shows a navigation bar with icons for Home, Search, Bookmarks, and Account, and a "Console" tab.

The screenshot shows a mobile browser interface. On the left, a modal dialog titled "Error encountered" displays the message: "An error occurred when trying to load the locations. Please check the console logs for more information." On the right, the developer tools' Network tab is open, showing a request to "https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations". The status is 401, and the response body is empty. The developer tools' Console tab at the bottom shows the following log entries:

```
Request without authorization:  
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations  
Headers: > {}  
Body: null  
✖ GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations 401 () polyfills.js:3  
✖ Response {_body: {}, status: 401, ok: false, statusText: "OK", headers: Headers, ...} main.js:1619
```

We will get a 401 error response back if there is no required header passed with the request as above.

The screenshot shows a mobile browser interface. On the left, the "Locations" screen lists several hotel entries: Aria (Las Vegas), Encore (Las Vegas), Test Building (Test), The Mirage (Las Vegas), and The Venetian (Las Vegas). On the right, the developer tools' Network tab is open, showing a request to "https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations". The status is 200, and the response body contains JSON data. The developer tools' Console tab at the bottom shows the following log entries:

```
Request without authorization:  
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations  
Headers: > {}  
Body: null  
✖ GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations 401 () polyfills.js:3  
✖ Response {_body: {}, status: 401, ok: false, statusText: "OK", headers: Headers, ...} main.js:1619  
User Pools Authorizer Request:  
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations  
Headers:  
Authorization: Array(1)  
Authorization: ["eyJraWQ10jJubnF4V2syN1JIdUtJN1ZcLytcL1Rcl2lSSHlnT0...VwdBMNFfyZzgMRI0UhVJ_zFn95WZoHfxPeUISav_nll"]  
__proto__: Object  
Body: null
```

If we send a valid authorization header which is our identity token along in the request header, we get a valid response

The screenshot shows a mobile browser interface with developer tools open. The top navigation bar includes 'Chrome', 'File', 'Edit', 'View', 'History', 'Bookmarks', 'People', 'Window', 'Help'. The developer tools tab bar has 'Elements', 'Console', 'Sources', 'Network', 'Performance', 'Memory', 'Application', 'Security', 'Audits'. The 'Console' tab is selected, showing the following log output:

```
Response {_body: {}, status: 401, ok: false, statusText: "OK", headers: Headers, ...} main.js:1619
User Pools Authorizer Request:
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations
Headers:
  Authorization: Array(1)
    Authorization: ["eyJraWQiOjJubnF4V2syN1JIdUtJN1ZcLytcL1RcL2l5SHlnT0..VwdBMNFfyZzgMRI0UhVJ_zFn95WZoHfxPeUISav..."]
    __proto__: Object
Body: null
Resources main.js:101
IAM Authorization Request:
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations/ca687440-d41e-11e7-b824-ad
e78d2bcd84/resources
Headers:
  Authorization: Array(3), Accept: Array(1), Content-Type: Array(1), x-amz-date: Array(1), x-amz-security-token
    Accept: ["application/json"]
    Authorization: ["AWS4-HMAC-SHA256 Credential=ASIAIM264AREXW4ULATA/20171128/us-east-1/execute-api/aws4_re
    Content-Type: ["application/json"]
    x-amz-date: ["20171128T230519Z"]
    x-amz-security-token: ["AgoGb3JpZ2luEFsaCXvzLWVhc30tMSKAAlKaKy67FbDbRDRC/7...l6Ub5FPvTMfKsFB091+ZH3i10HyzPgr8P
    __proto__: Object
Body: null
```

The bottom of the screen shows a footer with 'Welcome', 'Resources', 'Bookings', and 'Account' buttons.

In this case we are using an IAM based authorization request where we are sending several headers along with the request

The screenshot shows a mobile browser interface with developer tools open. The top navigation bar includes 'Chrome', 'File', 'Edit', 'View', 'History', 'Bookmarks', 'People', 'Window', 'Help'. The developer tools tab bar has 'Elements', 'Console', 'Sources', 'Network', 'Performance', 'Memory', 'Application', 'Security', 'Audits'. The 'Console' tab is selected, showing the following log output:

```
Response {_body: {}, status: 401, ok: false, statusText: "OK", headers: Headers, ...} main.js:1619
User Pools Authorizer Request:
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations
Headers:
  Authorization: Array(1)
    Authorization: ["eyJraWQiOjJubnF4V2syN1JIdUtJN1ZcLytcL1RcL2l5SHlnT0..VwdBMNFfyZzgMRI0UhVJ_zFn95WZoHfxPeUISav..."]
    __proto__: Object
Body: null
Resources main.js:101
IAM Authorization Request:
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations/ca687440-d41e-11e7-b824-ad
e78d2bcd84/resources
Headers:
  Authorization: Array(3), Accept: Array(1), Content-Type: Array(1), x-amz-date: Array(1), x-amz-security-token
    Accept: ["application/json"]
    Authorization: ["AWS4-HMAC-SHA256 Credential=ASIAIM2G4AREXW4ULATA/20171128/us-east-1/execute-api/aws4_request"
    Content-Type: ["application/json"]
    x-amz-date: ["20171128T230519Z"]
    __proto__: Array(0)
  __proto__: Object
Body: null
```

The bottom of the screen shows a footer with 'Welcome', 'Resources', 'Bookings', and 'Account' buttons.

All the different headers make up the signature signing

Resource Availability

Showing availability for Nov 28, 2017

Time	Status	Action
6am - 7am	AVAILABLE	BOOK
7am - 8am	AVAILABLE	BOOK
8am - 9am	AVAILABLE	BOOK
9am - 10am	AVAILABLE	BOOK
10am - 11am	Booking...	BOOK
11am - noon	AVAILABLE	BOOK
noon - 1pm	AVAILABLE	BOOK
1pm - 2pm	AVAILABLE	BOOK
2pm - 3pm	AVAILABLE	BOOK
3pm - 4pm	AVAILABLE	BOOK
4pm - 5pm	AVAILABLE	BOOK

Body: null

Resource Availability

IAM Authorization Request:

```
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/location--b824-ade78d2bcd84/resource/cab3ae10-d41e-11e7-b824-ade78d2bcd84/bookings
```

Headers:

```
> {Authorization: Array(3), Accept: Array(1), Content-Type: Array(1), x-amz-date: Array(1), x-amz-security-token}
```

Body: null

IAM Authorization Request:

```
POST https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/users/us-east-1:36557168-c37c-4e4a-bb5e-de81479913af/bookings
```

Headers:

```
> {Content-Type: Array(1), Authorization: Array(3), Accept: Array(1), x-amz-date: Array(1), x-amz-security-token}
```

Body: {"locationId": "ca687440-d41e-11e7-b824-ade78d2bcd84", "locationName": "Aria", "resourceId": "cab3ae10-d41e-11e7-b824-ade78d2bcd84", "resourceName": "Ironwood 8", "resourceDescription": "Aria Level 3", "startTimeIsoTimestamp": "2017-11-28T08:00:00Z", "endTimeIsoTimestamp": "1511856000000", "endTimeEpochTime": 1511856000000, "timeslot": "8am - 9am", "userId": "us-east-1:36557168-c37c-4e4a-bb5e-de81479913af", "userFirstName": "Vladimir", "userLastName": "Budilov"}

Console

Resource Availability

Showing availability for Nov 28, 2017

Time	Status	Action
6am - 7am	AVAILABLE	BOOK
7am - 8am	AVAILABLE	BOOK
8am - 9am	Vladimir Budilov	CANCEL
9am - 10am	AVAILABLE	BOOK
10am - 11am	AVAILABLE	BOOK
11am - noon	AVAILABLE	BOOK
noon - 1pm	AVAILABLE	BOOK
1pm - 2pm	AVAILABLE	BOOK
2pm - 3pm	AVAILABLE	BOOK
3pm - 4pm	AVAILABLE	BOOK
4pm - 5pm	AVAILABLE	BOOK

Body: null

Resource Availability

IAM Authorization Request:

```
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/location--b824-ade78d2bcd84/resource/cab3ae10-d41e-11e7-b824-ade78d2bcd84/bookings
```

Headers:

```
> {Authorization: Array(3), Accept: Array(1), Content-Type: Array(1), x-amz-date: Array(1), x-amz-security-token}
```

Body: null

IAM Authorization Request:

```
POST https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/users/us-east-1:36557168-c37c-4e4a-bb5e-de81479913af/bookings
```

Headers:

```
> {Content-Type: Array(1), Authorization: Array(3), Accept: Array(1), x-amz-date: Array(1), x-amz-security-token}
```

Body: {"locationId": "ca687440-d41e-11e7-b824-ade78d2bcd84", "locationName": "Aria", "resourceId": "cab3ae10-d41e-11e7-b824-ade78d2bcd84", "resourceName": "Ironwood 8", "resourceDescription": "Aria Level 3", "startTimeIsoTimestamp": "2017-11-28T08:00:00Z", "endTimeIsoTimestamp": "1511856000000", "endTimeEpochTime": 1511856000000, "timeslot": "8am - 9am", "userId": "us-east-1:36557168-c37c-4e4a-bb5e-de81479913af", "userFirstName": "Vladimir", "userLastName": "Budilov"}

Console

The screenshot shows a browser window with developer tools open, specifically the 'Console' tab. The URL in the address bar is 'Developer Tools - file:///android_asset/www/index.html'. The console output displays an 'IAM Authorization Request' for an AWS Lambda function. The request includes headers for Authorization, Accept, Content-Type, x-amz-date, and x-amz-security-token. The body of the request contains a JSON object with location details, a timeslot, and user information (FirstName: Vladimir, LastName: Budilov). The AWS Lambda function name is 'ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/location--b824-ade78d2bcd84/bookings'. The code snippet is annotated with main.js:3323 and main.js:101.

```
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/location--b824-ade78d2bcd84/resources/cab3ae10-d41e-11e7-b824-ade78d2bcd84/bookings
Headers:
> {Authorization: Array(3), Accept: Array(1), Content-Type: Array(1), x-amz-date: Array(1), x-amz-security-token: Array(1)}
Body: null
IAM Authorization Request:
POST https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/users/us-east-1:36557168-c37c-4e4a-bb5e-de81479913af/bookings
Headers:
> {Content-Type: Array(1), Authorization: Array(3), Accept: Array(1), x-amz-date: Array(1), x-amz-security-token: Array(1)}
Body: {"locationId": "ca687440-d41e-11e7-b824-ade78d2bcd84", "locationName": "Aria", "resourceId": "cab3ae10-d41e-11e7-b824-ade78d2bcd84", "resourceName": "Ironwood 8", "resourceDescription": "Aria Level 3", "startTimeIsoTimestamp": "2017-11-28T08:00:00.000Z", "startTimeEpochTime": 1511856000000, "endTimeIsoTimestamp": "2017-11-28T09:00:00.000Z", "endTimeEpochTime": 1511856900000, "timeslot": "8am - 9am", "userId": "us-east-1:36557168-c37c-4e4a-bb5e-de81479913af", "userFirstName": "Vladimir", "userLastName": "Budilov"}
My Bookings
IAM Authorization Request:
GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/users/us-east-1:36557168-c37c-4e4a-bb5e-de81479913af/bookings
Headers:
> {Authorization: Array(3), Accept: Array(1), Content-Type: Array(1), x-amz-date: Array(1), x-amz-security-token: Array(1)}
Body: null
```

The screenshot shows a browser window with the following details:

- Address Bar:** https://s3.amazonaws.com/spacefinder-public-image-repository/building.png
- Error Dialog:** An alert box titled "Error encountered" states: "An error occurred when trying to add the location. Please check the console logs for more information." It has an "OK" button.
- Developer Tools Console:** The "Console" tab is selected. The log output is as follows:
 - Body: null
 - Account** [main.js:101]
 - Resource Availability** [main.js:101]
 - Locations** [main.js:101]
 - IAM Authorization Request:** GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/location...b824-ade78d2bcd84/resource s/cab3ae10-d41e-11e7-b824-ade78d2bcd84/bookings
 - Headers:
 - Body: null
 - Add a Location** [main.js:101]
 - IAM Authorization Request:** POST https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations
 - Headers:
 - Body: {"name": "Test Location", "description": "Test", "imageUrl": "https://s3.amazonaws.com/spacefinder-public-image-repository/building.png"}
 - ▶▶ POST https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations 403 () polyfills.js:3
 - ▶▶ Response {_body: {}, status: 403, ok: false, statusText: "OK", headers: Headers, ...} main.js:65

We are not an admin, so our IAM policy did not allow us to do that request

The screenshot shows a mobile application interface with a navigation bar at the top. Below the bar, there's a form with fields for 'Location Name' and 'Description'. A modal dialog box is displayed in the center, stating 'Signed out' and 'You have signed out of the system.' with an 'OK' button. The background shows a blurred view of the application's main screen.

Developer Tools - file:///android_asset/www/index.html

Elements Console Sources Network Performance Memory Application Security Audits 4 items hidden by filters

Body: null

Account main.js:101

Resource Availability main.js:101

Locations main.js:101

IAM Authorization Request: main.js:3323

GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/location--b824-ade78d2bcd84/resources/cab3ae10-d41e-11e7-b824-ade78d2bcd84/bookings

Headers:

► (Authorization: Array(3), Accept: Array(1), Content-Type: Array(1), x-amz-date: Array(1), x-amz-security-token: Array(1))

Body: null

Add a Location main.js:101

IAM Authorization Request: main.js:3323

POST https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations

Headers:

► {Content-Type: Array(1), Authorization: Array(3), Accept: Array(1), x-amz-date: Array(1), x-amz-security-token: Array(1)}

Body: {"name": "Test Location", "description": "Test", "imageUrl": "https://s3.amazonaws.com/spacefinder-public-image-repository/building.png"}

► POST https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations 403 () polyfills.js:3

► Response {_body: {}, status: 403, ok: false,.statusText: "OK", headers: Headers, ...} main.js:65

Sign Out main.js:101

Console

The screenshot shows a mobile application interface with a navigation bar at the top. Below the bar, there's a form for 'Sign-in using Cognito User Pools'. It has fields for 'Username' (containing 'admin1') and 'Password' (containing '*****'). Below the form are 'SIGN IN' and 'FORGOT PASSWORD' buttons. A progress indicator 'Signing in...' is visible. The background shows a blurred view of the application's main screen.

Developer Tools - file:///android_asset/www/index.html

Elements Console Sources Network Performance Memory Application Security Audits

Authenticating user admin1 main.js:621

SIGN IN FORGOT PASSWORD

Signing in...

Console

Sign-in using Cognito User Pools

Success!

You are now signed in as an Administrator.

Username: admin1
First name: Admin
Last name: User

```

KE2NRq57p7df20gt9rlGwAp_kCCBEshySWLWnB9gNPb0nn4bNzsux_vVWkafxuorJLry4CSEnvBBxSnQvCpi40EdBSlyWr9lBuB9tWKEe3w
StHXIJb76X0DWEU0AaKqWA.EwYjHp8LB3ho16JP1E7zDQ
Cognito User Pools User Groups: admin1 belongs to group adminGroup
main.js:681
Cognito User Pools User Attributes:
main.js:685
  {sub: "8156a496-12cf-4700-a59c-2f97b0896858", cognito:groups: Array(1), cognito:preferred_role:
  ▶ "arn:aws:iam::742360508337:role/spacefinder-development-s-CognitoIdentityPoolAuthA-J5HGS0N215N2", iss:
  "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_Iaovdw2o2", cognito:username: "admin1", ...}
  Cognito Identity ID: us-east-1:efe4906f-f410-48ca-9423-e439a0f57afc
  main.js:690
  AWS Access Key ID: ASIAINHRBCCGOJJZMBXQ
  main.js:692
  AWS Secret Access Key: tAOAArAlhaACNx8glgnDAEi0jW+ZFmpmUgzateG3l
  main.js:694
  AWS Session Token:
  main.js:696
  AgoGb3JpZ2luEfsaCXVzLWVhc30tMSKAAn5y9yQ0Uq1GkzWvEDjnxvETRJaBcukPbjm801q3elQ20BxcDo|paGH1mThaHhkfNgUSP2L4zv1b
  S3HbwJUvEcT88TV/5/X35uuv70x0D1CVoW6QAU5Woo8kqza2lRTVQn/CxLxGRy0ah74L1Dbd4+rg05jRm10KRnu22c40LAewfmon7YCbnTnymAGM
  SN180UkmjaGJAwPD1x7Mh7WjR00Vbd1y17X1ZmqZaebhiIuysF1L0qdGq9xP95C9JptsfCMfbpincUJXqP2C3ZdI7LM1MtA2N6Tuejb
  5jkjJ0dg9CpfTGKRGSPWmzLMyePupBcahyLcbnRyIx0F5VhB8BqhgyIsP//////////ARAAGgv3NDiZnja1MDgZnzcIDP02DfQKsaAVIFWt
  HyraBvoz5bpPAf4u9HNyfGG6Qk5kjJH1TvhBAnjweS32PmVXGL5YdZt/yuRg140k+6YL02tf6KAqodyaANfdYOHuNuqv1RWgjsVgdQbAk
  0+7rps2CDH10RyTkgtu53EG1lgsZQGf40ZM++WMtMvVHGcx86XN71lcKU25HUY95Ae/FW7hjFZK1CrX/6vNSPdkEbl8+QmniVdBhFQbGtpG
  HMsasuz1F71mkD424wZl+kH230dqbtovsnD/v0FjJ0PvXXt0a7Hu47chbRCK7uGp9Bsn/kujeerZu+Zjde9ehEl2zw/Pyskcx4TiW4zu1Yp
  zUoP6JMDws7GwoQZtq/mDccoaG8wdG88hgchlk0l+nAWAR9H2J7Lfeb8jRrC0/e+aEn3PDRsWsKEIAVad1nsuqAEADvaf86f3j85wCT0
  /UX7qIXQ4Djc7UYMonimfdVRHHEmsG4L1w/vhugZu5GE3L0lBhsDITf4m/5RITHfp3tPzeJ2W7NwDE5zjybfbZbgRvY29PZf1y056IyeSGpzm
  YgHo$2TQNsMsGD4goKjy3e4d0zPE20klw3a2BxxT Ted8a575c5CIHP03RZScnqR9xF0L2v7D1zo1CSjbp/tDXzmlL2f5oDn1jotM+n
  YPgruJEO4XPkJ0kvej09C8pxXh+Wn0DP07zsqlf/rsxtPwQaDNBuenfqan02jdBWJowseJV6BT7Q0DwPilJZwD1Gsvx840V/H9ZT0o7xB
  03n+oTwfeE59BFakZes5v9H8KTERieb2apZpAiCBVCrSIV10Yb9jdhludwSmN046W5JBzop7h0XxJNbypjKj05egjzuEsC2G1fGTGrnsI6y
  to3LVALE8D0uboMytWNeKFFgtKQP7h5up1n5o7ucJycpc6R890HF+C2V+4xMnKHCC3gF6j1zemGuv10/2K6KqcFEhH/wAG0W7r0qia16
  knPxoYoWrtj30AU=
  
```

We are now an admin

Locations

Load Locations without Auth

Load Locations with Auth

ADD A LOCATION

```

Cognito User Pools User Groups: admin1 belongs to group adminGroup
main.js:681
Cognito User Pools User Attributes:
main.js:685
  {sub: "8156a496-12cf-4700-a59c-2f97b0896858", cognito:groups: Array(1), cognito:preferred_role:
  ▶ "arn:aws:iam::742360508337:role/spacefinder-development-s-CognitoIdentityPoolAuthA-J5HGS0N215N2", iss:
  "https://idp.us-east-1.amazonaws.com/us-east-1_Iaovdw2o2", cognito:username: "admin1", ...}
  aud: "3ipvq74d4dn1lf92p5qtolm"
  auth_time: 1511910444
  cognito:groups: ["adminGroup"]
  cognito:preferred_role: "arn:aws:iam::742360508337:role/spacefinder-development-s-CognitoIdentityPoolAuthA-J5HGS0N215N2"
  cognito:roles: ["arn:aws:iam::742360508337:role/spacefinder-development-s-CognitoIdentityPoolAuthA-J5HGS0N215N2"]
  cognito:username: "admin1"
  email: "admin@example.com"
  event_id: "e5672214-d490-11e7-9a5a-2b0966933efa"
  exp: 1511914044
  family_name: "User"
  given_name: "Admin"
  iat: 1511910444
  iss: "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_Iaovdw2o2"
  sub: "8156a496-12cf-4700-a59c-2f97b0896858"
  token_use: "id"
  __proto__: Object
  Cognito Identity ID: us-east-1:efe4906f-f410-48ca-9423-e439a0f57afc
  main.js:690
  AWS Access Key ID: ASIAINHRBCCGOJJZMBXQ
  main.js:692
  
```

The screenshot shows a mobile application interface for managing locations. On the left, there's a sidebar with 'Locations' at the top, followed by 'Load Locations without Auth' and 'Load Locations with Auth'. Below this are five location entries: 'Aria Las Vegas', 'Encore Las Vegas', 'Test Building Test', 'Test Location Test', and 'The Mirage Las Vegas'. Each entry has a small thumbnail image, a name, a location, and a red 'DELETE' button. At the bottom of the sidebar are 'Welcome', 'Resources', 'Bookings', and 'Account' buttons.

The main content area displays a developer tools console. The 'Console' tab is selected. The output shows:

```

Developer Tools - file:///android_asset/www/index.html
Elements Console Sources Network Performance Memory Application Security Audits ...
top Filter Default levels 4 items hidden by filters ...
Cognito User Pools User Groups: admin1 belongs to group adminGroup
main.js:681
Cognito User Pools User Attributes:
main.js:685
  {
    "sub": "arn:aws:iam::742360508337:role/spacefinder-development-s-CognitoIdentityPoolAuthA-J5HGS0N21SN2",
    "iss": "https://idp.us-east-1.amazonaws.com/us-east-1_Iaovdw2o2",
    "cognito:username": "admin1",
    "aud": "3lpvq74d4dlif92p5qtolim",
    "auth_time": 1511910444
    ...
  }
  > cognito:groups: ["adminGroup"]
  > cognito:preferred_role: "arn:aws:iam::742360508337:role/spacefinder-development-s-CognitoIdentityPoolAuthA-J5HGS0N21SN2"
  > cognito:roles: ["arn:aws:iam::742360508337:role/spacefinder-development-s-CognitoIdentityPoolAuthA-J5HGS0N21SN2"]
  > cognito:username: "admin1"
  > email: "admin@example.com"
  > event_id: "e5672214-d490-11e7-9a5a-2b0966933efa"
  > exp: 1511914844
  > family_name: "User"
  > given_name: "Admin"
  > iat: 1511910444
  > iss: "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_Iaovdw2o2"
  > sub: "B156a496-12cf-4700-a59c-2f97b0896858"
  > token_use: "id"
  > __proto__: Object
Cognito Identity ID: us-east-1:efe4906f-f410-48ca-9423-e439a0f57afc
main.js:690
AWS Access Key ID: ASIAINHRBCCG0JJZMBXQ
main.js:692

```

At the bottom of the developer tools window, it says '© 2008-2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.

We are now able to create a resource since we are now an admin

The screenshot shows a mobile application interface for managing resources. On the left, there's a sidebar with 'Resources' at the top, followed by 'Test Location' and 'Test Room Test'. Below this is a blue button labeled 'ADD A RESOURCE'. At the bottom of the sidebar are 'Welcome', 'Resources', 'Bookings', and 'Account' buttons.

The main content area displays a developer tools console. The 'Console' tab is selected. The output shows:

```

Developer Tools - file:///android_asset/www/index.html
Elements Console Sources Network Performance Memory Application Security Audits ...
top Filter Default levels 6 items hidden by filters ...
Authenticating user admin1
main.js:621
  > e {idToken: e, refreshToken: e, accessToken: e}
  > e {idToken: e, refreshToken: e, accessToken: e}
Cognito User Pools Identity Token:
main.js:626
  keyJraWQ101jubnF4V2syN1jd0tJN12LytcL1RcL2lSShlnT0xldHFJVmJaNlwVmZaa2FVdz01LCJhbGcI01JSUzI1NIJ9.eyJzdWI0iI
  4MTU2YTQ5N10xMnMnLTQ3MDATYTUSYy0yzJk3YJA40TY4NTg1LCjb2duaXrv0mdyB3WcyI6WyJhZG1pbkdyb3VwIl0sImNvZ25pdG86CHJ
  1ZmVycmVXk3JvbGU101jhcm46YXzd0mhbTo6Nz0yMzYNTA4MzM30nJvbGvC13NwYwNlZmluZGVyLWR1dmVsB3tZW50LXmTQ29nb1b01
  kZWS0aXR5UG9vbaEF1dGh8Lu15EdTME4yMTVOMiisImLzcyI61mh0dBz01lwvXc9jb2duaXrvLwLkcC51cy1lyXN0LTEuYw1hem9uYXzdLnN
  vbVwvdXmtZWFzdc0x0khb32kdzJvM1isImNvZ25pdG86DXNlm5hWu101jhZG1pbjE1LCJnaZ1bl9uYw1lji0QRwta41LcJzb2duaXR
  v0nJvbGvz1jbpbImFybphd3M6aw0t0j03NDIzNjA1MDgzm2c6cm9sZVwv3Bhy2Vmav5kZXitZGV22wvxcG1lbnQtcyIDb2duaXrvSwR1bnr
  pdHlqb29sOXvBaEtsJVR1mWtJixNu4yI10sImf1ZC161jMxchXzNzRkNGruBDFm1TkycDvxdG9saw0lJCJldmVudF9pZC161mU1NjMJE
  0LW00T0tMTFLNv05YTvhLTj1MDk2Njk2H2Vm5isInRva2Vu3VzZS16Im1kiwiYXv0aF98aW1lji0xNTEx0TeWND00Lc1jeHai0jE1MTE
  5MTQ0NDQs1lhdCI6MTuXMTkxDQ0NCw1zFtaWx5X25hbWU101jVc2VyI1iwiZw1haWw101jhZG1pbkBlEGfcGx1Ln0vbSJ9.pYE0kwvgv0
  V13_F-CG_P2BZs3bcvJXNMN3GDUxvngh4DKpWpxj1Rnkmm_BElguSwv10uQSK04tanW-uyntmxvN8JMHg-b1221fkcod2MRcnri-
  S1KRPjitswNmpavJsdunky0az3L_42diq75Vu05sKgyBmtkbD_yRxssllpKjHYS7tcek1i-gLNMscl0XGJ7k24wKnWzvVcu17W49401-
  KfR7CYdudbdIdvajtg1lb10nxw1n1UBr3_d_5KSzu5Ge1vlQyq15mjD0uN4HA5og62Kw7_6EB1vd8GZlZmyBZ7q-
  MSLAm0xyBx1jQKXy8Ecc18UyKEW9V1Cw
Cognito User Pools Access Token:
main.js:657
  eyJraWQ101jubnF4V2syN1jd0tJN12LytcL1RcL2lSShlnT0xldHFJVmJaNlwVmZaa2FVdz01LCJhbGcI01JSUzI1NIJ9.eyJzdWI0iI4MTU2
  YTQ5N10xMnMnLTQ3MDATYTUSYy0yzJk3YJA40TY4NTg1LCjb2duaXrv0mdyB3WcyI6WyJhZG1pbkdyb3VwIl0sImNvZ25pdG86CHJ
  NzIyMT0tZDQ5MC0xMnMj3LTLhlnWetMni0TY20TMzWZh1iwd9rZw5fdXnl1oi1WnJZKNI1iwiwic2NvcGU101jh3MuY29nb1b01y5zaWdu
  aWudXNl1c15hZG1pbjls1mlzcyI61mh0dBz01lwvXc9jb2duaXrvLwLkcC51cy1lyXN0LTEuYw1hem9uYxdlrnvbwvdXmZWFzdc0x0lh
  b32kdzjvM1isImV4acC16MTkxDQ0NCw1awF01joxNTEx0TEwND01LCJqdGk10i140WI12W1Y102YjAwL7Q4Yw01Yj1c3MS1hZTQ4M2Ex
  NWJLzmEiLCjbGllbnRfaWQ10i1zMX82cTc0ZDRkbmvxZnUSMnA1cXrvbGlt1iwiwic2NvcGU101jhZG1pbjEifQ.gV0_o1vnwo67Ynh1
  r5WZGfwdfbfoCF3bnU0IpCrer0g7hs9lgeseAsQ6Rw7fSwXqGyIMbjD4EM0ExTpj7rCX-
  KvrHPvVLMAfmk1ZF0Y6CtqKbtBwHkTRJjwv91r0RCvdkK2XyQ0lgiu8jabd84RzY_rV1xpDeBNL02_Cj1QJRuNRzotePejABkgTgm-
  nry0kPASeUw-tBaMU-
  mPcCSAzsUsng4NjhNrhrh64aP1D3IpBpcmEgwWhnjW5f0s6kox2Cfa3bDyYw1hQvtgEA_KIrwKGd1_COM_lThuA9_uCJqG5CpRa_uMoIXSyA
  4670QBudKd3ig1c43LEdBaBg

```

At the bottom of the developer tools window, it says '© 2008-2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.

We can now copy this admin identity token and see what policy it is using as below

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cognito JSON Web Tokens - jwt.io API Gateway New Tab

https://console.aws.amazon.com/apigateway/home?region=us-east-1#apis/ye07xoo3oc/authorizers

Spacefinder-API

Authorizers enable you to control access to your APIs using Amazon Cognito User Pools or a Lambda function.

+ Create New Authorizer

spacefinder-userPool-authorizer

Cognito User Pool
spacefinder-development-userPool - lsaovdw2o2 (us-east-1)

Token Source Authorization Token Validation none

spacefinder-custom-authorizer

Lambda Function
spacefinder-development-authorizer-Custom (us-east-1)

Lambda Event Payload Token

Token Source Authorization Token Validation none

Authorization Caching Authorization cached for 5 minutes

Edit Test

Feedback English (US) © 2006 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cognito JSON Web Tokens - jwt.io API Gateway New Tab

https://console.aws.amazon.com/apigateway/home?region=us-east-1#apis/ye07xoo3oc/authorizers

Spacefinder-API

Authorizers enable you to control access to your APIs using Amazon Cognito User Pools or a Lambda function.

+ Create New Authorizer

spacefinder-custom-authorizer - Test Authorizer

You can test your authorizer by providing values that will be used to invoke your Lambda function or make a call to your Cognito User Pool.

Authorization Token

Test

Response

Response Code: 200

Latency 49

Policy

Close

spacefinder-custom-authorizer

Lambda Function
spacefinder-development-authorizer-Custom (us-east-1)

Token Validation none

Test

Feedback English (US) © 2006 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

We can check this in our custom authorizer

The screenshot shows the AWS API Gateway console. On the left, there's a sidebar for the 'Spacefinder-API' stage. Under 'Authorizers', a 'Create New' button is visible. A modal window titled 'spacefinder-custom-authorizer - Test Authorizer' is open, showing a JSON policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "execute-api:Invoke",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:execute-api:us-east-1:742360508337:ye07xoo3oc/n"
      ]
    }
  ]
}
```

Below the policy, there's a 'Test' button. At the bottom of the modal is a 'Close' button.

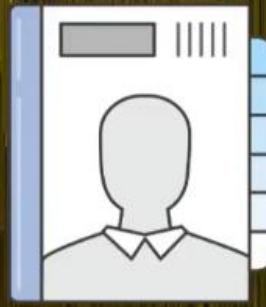
This policy allows everything, there is no deny statement for this admin role

Architecture so far...



Now the microservice is complete. What is we want to allow users to federate their existing identity provider?

3rd Party Federation



App Integration and Federation

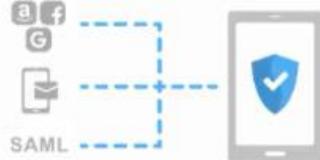
1

Built-in, Customizable
User Interface for Sign up
/ Sign in



2

Federation with Facebook,
Login with Amazon,
Google, and SAML2
providers



3

OAuth 2.0 Support



AWS re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



We have released a feature called Hosted UI that allows you to create a custom UI that federates with Facebook, Google, SAML2 providers, etc.

Integrating with Social IdPs



You simply have the Hosted UI presented to the user,

Integrating with Social IdPs



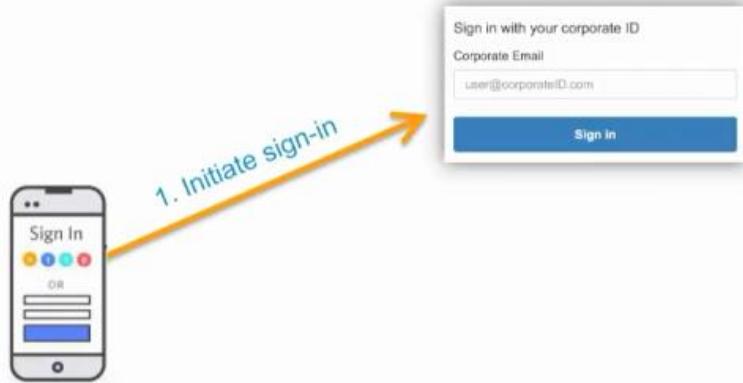
The user is then redirected to the 3rd party sign in page as above,

Integrating with Social IdPs



The user is then redirected to Cognito User Pools with all the credentials and additional credentials from the 3rd party providers too.

Integrating with Enterprise IdPs

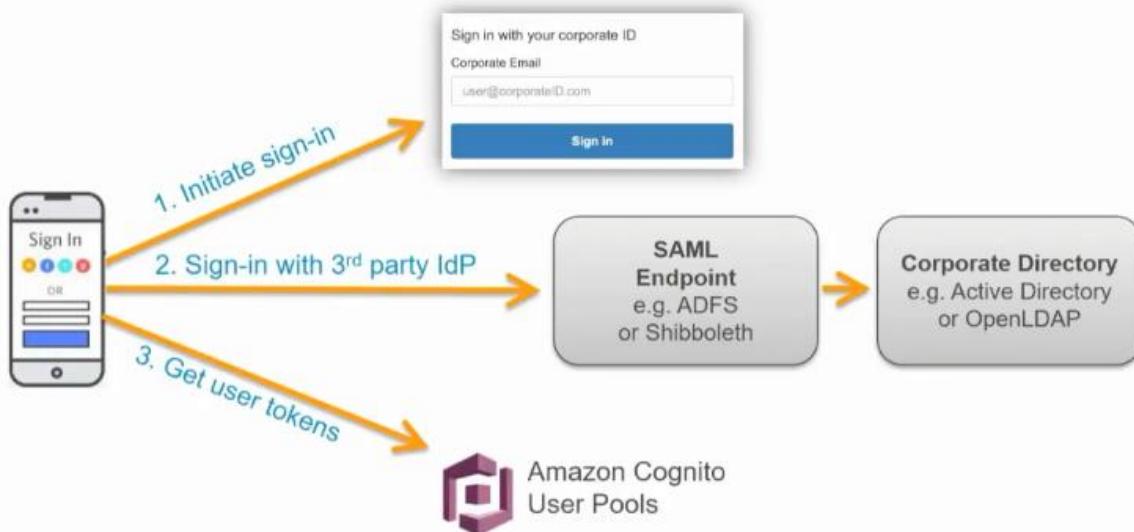


With SAML for your enterprise IdPs, the process is similar

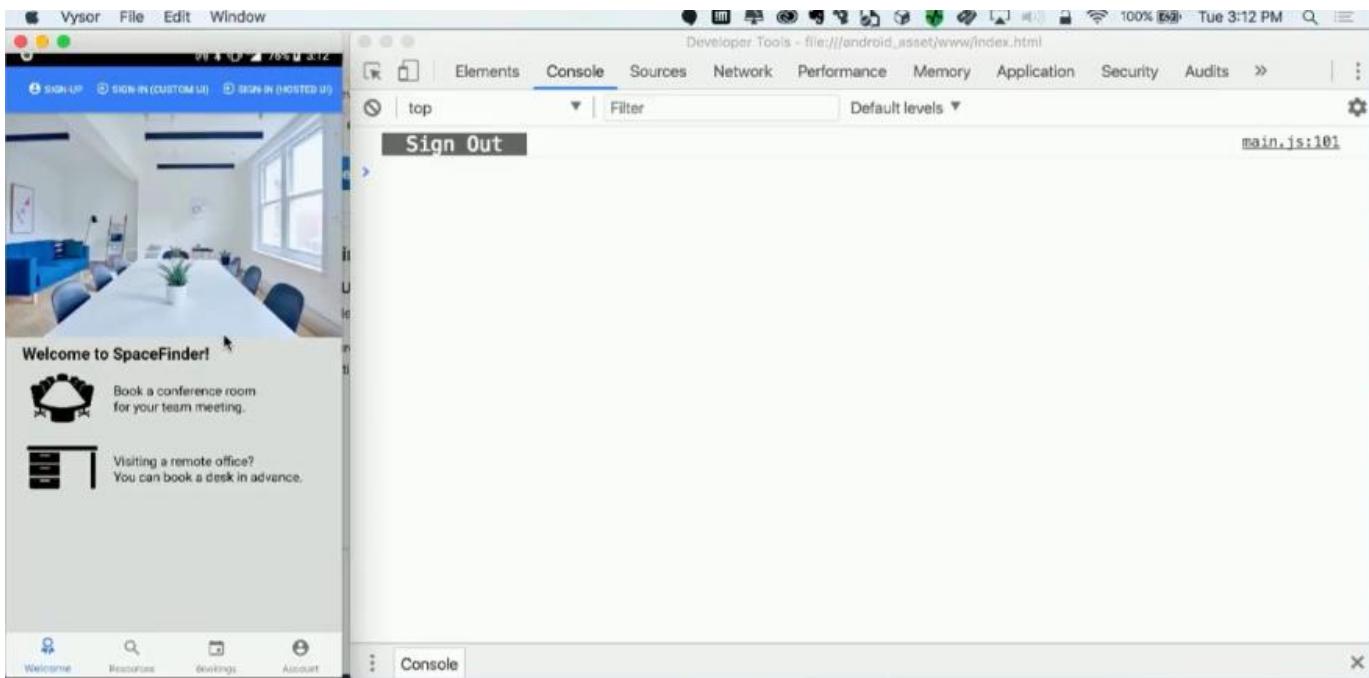
Integrating with Enterprise IdPs



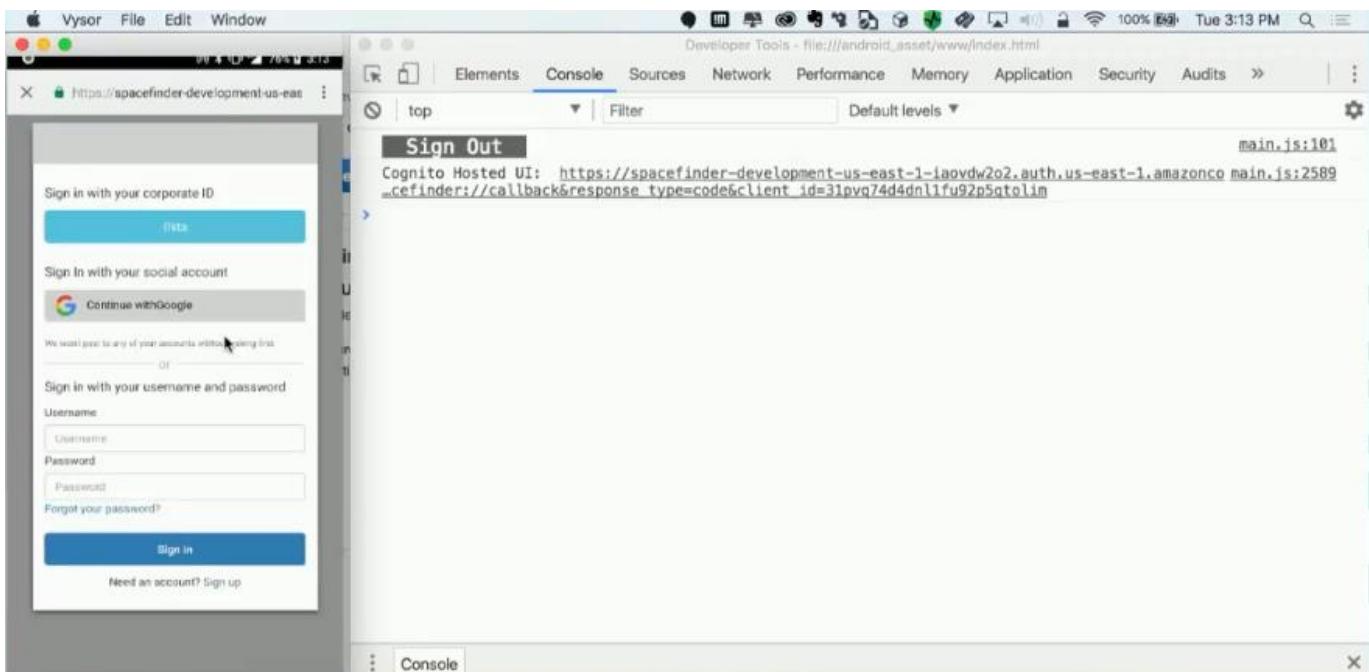
Integrating with Enterprise IdPs



DEMO



We now have a Hosted UI option that we can click on



We get the above UI that is customizable. We can sign in with our username and password, or federate with any of the 3rd party providers and Cognito will take care of the coordination like mapping attributes or having an endpoint that you can receive the POST from in SAML response.

Vysor File Edit Window

https://accounts.google.com

Google

Choose an account
to continue to [amazoncognito.com](#)

Justin Pirtle
justin-google@aws-demos.com

Use another account

Sign Out

Cognito Hosted UI: <https://spacefinder-development-us-east-1-iaovdw2o2.auth.us-east-1.amazoncognito.com/main.js:2589>

cefinder://callback&response_type=code&client_id=31pvq74d4dn1l1fu92p5gtolim

main.js:101

Developer Tools - file:///android_asset/www/index.html

Elements Console Sources Network Performance Memory Application Security Audits

top Filter Default levels

Console

Vysor File Edit Window

https://accounts.google.com

Google

Sign in
to continue to [amazoncognito.com](#)

Email or phone

Forgot email?

More options

NEXT

Sign Out

Cognito Hosted UI: <https://spacefinder-development-us-east-1-iaovdw2o2.auth.us-east-1.amazoncognito.com/main.js:2589>

cefinder://callback&response_type=code&client_id=31pvq74d4dn1l1fu92p5gtolim

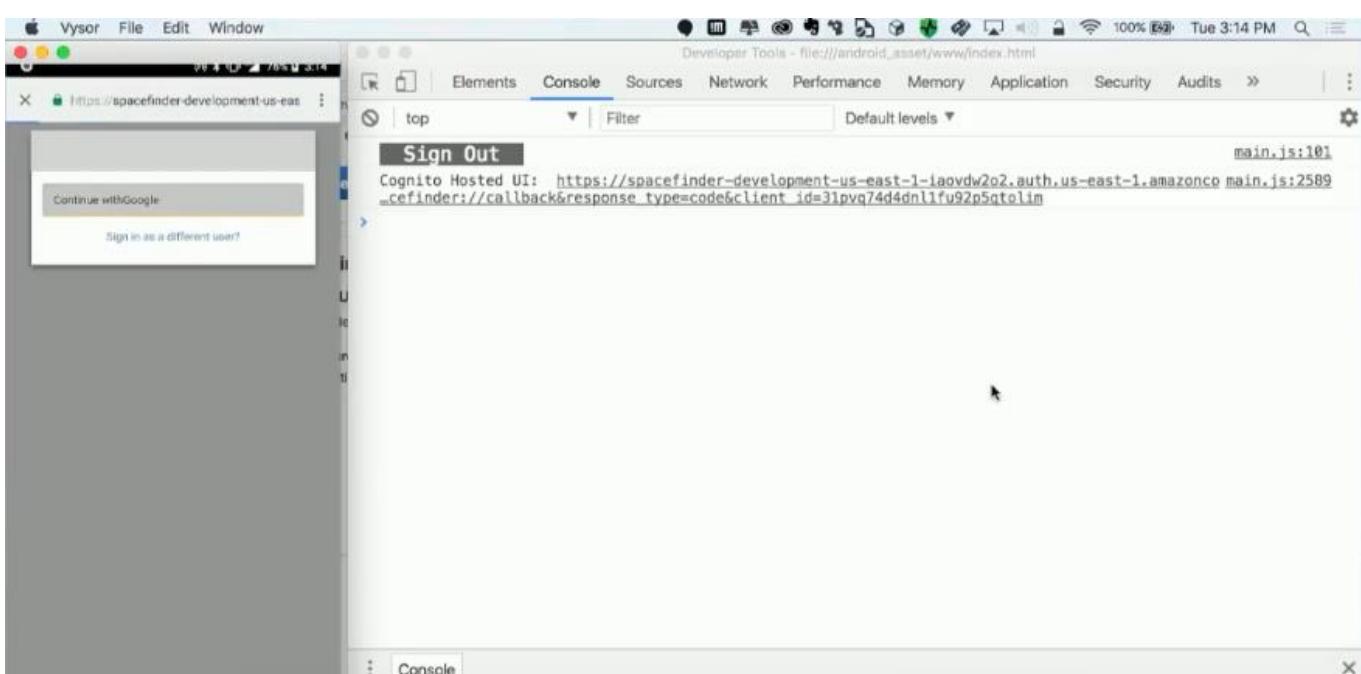
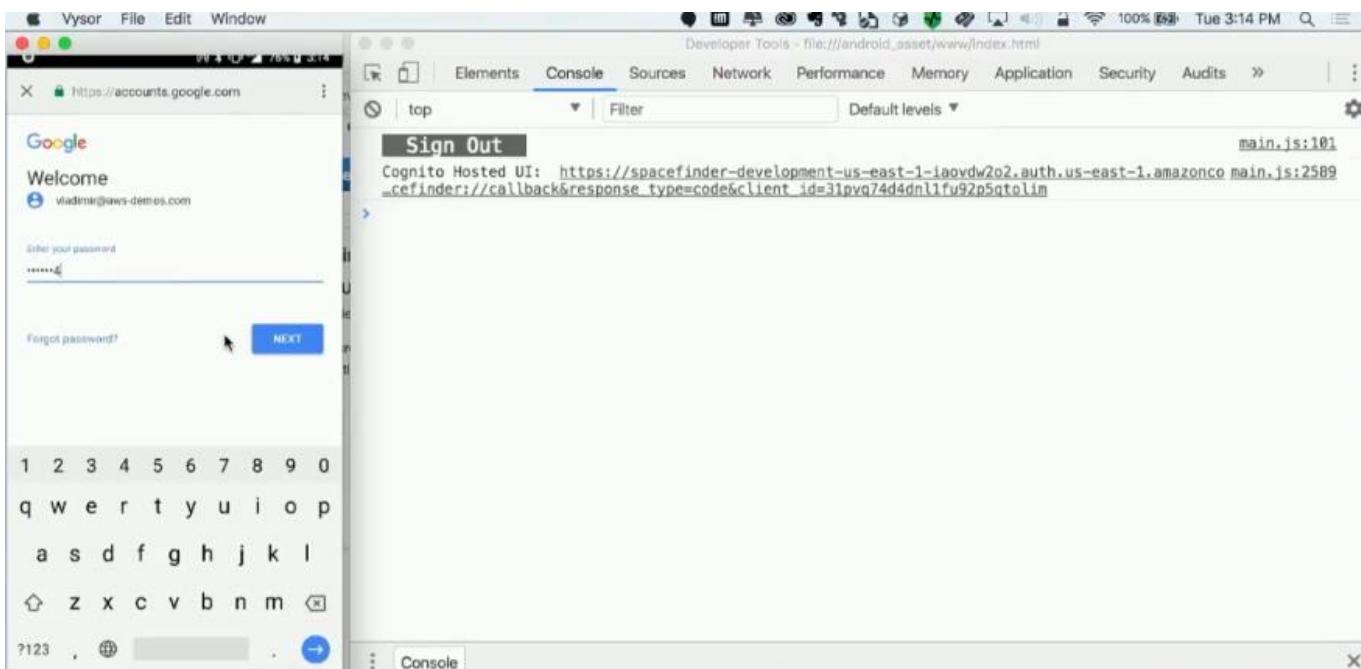
main.js:101

Developer Tools - file:///android_asset/www/index.html

Elements Console Sources Network Performance Memory Application Security Audits

top Filter Default levels

Console



The screenshot shows a mobile application interface on the left and a browser-based developer tools interface on the right. The mobile app displays a 'Success!' message: 'You are now signed in.' It shows the user's details: Username: Google_10154877591919406750, First name: Vladimir, Last name: Budilov. Below this is a 'OK' button. The developer tools' console tab shows a 'Sign Out' request with the URL: <https://spacefinder-development-us-east-1-iaovdw2o2.auth.us-east-1.amazonaws.com/main.js:2589>. The status bar at the top indicates 100% battery and the time is Tue 3:14 PM.

Because this is a mobile app, especially if you were doing a SAML response directly. SAML only supports a redirect response for a POST back to your client. But on a mobile device this is not an easy option. Cognito helps by intercepting that POST response from the SAML provider and provides us a standardized set of these JWT tokens that are always the same regardless of where you came from. We then get the callback from Cognito itself to actually do the sign in.

The screenshot shows a mobile application 'Locations' screen on the left and developer tools on the right. The mobile app lists several locations: Aria (Las Vegas), Encore (Las Vegas), Test Building (Test), Test Location (Test), The Mirage (Las Vegas), and The Venetian (Las Vegas). The developer tools' console tab shows the following data:

```

Cognito User Pools User Attributes: > Object
Cognito Identity ID: us-east-1:93993e81-e8e2-48a4-8cbd-c008daf2ab14
AWS Access Key ID: ASIAIP7W003QVIA2LA
AWS Secret Access Key: 7wTy3FeJsgalAIRqOsPxASUjWlwiJnZFgj2qUzb+
AWS Session Token:
Ag0Gb3JpZ2luEFsaCVxLwVhC30tMSKAAhNtkKnTbsUh687wugfuG9nxyXP/xxdVCn3pfaj7WyQX438ezQrByvU0FLDOL8m1NDGtpWcqLu3x
UvWldzumHsBh1PgCfPh2udLnjsnTLnq0k5RLPnZixA147usJNqaOR2YdX9Fr9Gy09AIssXXsepVbymFaTLF5nJgbzqYFT0MyvoOo
TlqbIbpyhkV45xHaeIHivGY4RKzIwNU6Uz6vrddbgxCzHsxflibySf105ickyR60lxIcPl3fdwF1hbedjpj2vvgn67Yht31VbPuc8J/LRae
dFPjHe1/2jaoCD2TcnZhrc5C3jxyiUmnTs09YY4RHMmf1lbEhqnyIsP//////////ARAAQgw3NDiZnJiA1HDg2MzciDPB8+gAx19oTfkX
Byra8fPQngPu6anwrBZikIL7EP+0BRjRgyvuS9QG7oDOr/eCeEKTptkrM2cYMShz02X3auPB8NAQL50gRwykaK4+mhe6V5RFk0fVzKjOb5
sxgLB8dxZ134sd7M/KQwH9ATr9qPVPR+jtLUAl15ale6fzJ4ze7SE3U08ZTV3+00+dgPLbrMnR15Ad3gpzuV6nfZQMdnxIHC4BecltPuy
euRKzhzEUZlwr0fFGUPn0C03Y5nX6jfqHmj5sA5uxFAQ3clGC4qY2iEyGGKzcIbaHbUvNyV1cknFPDBqwmLIX9pvWE5JcyuvvHi9450GN
EL27HzdJ1815hydNxMq027mYCjwcvYXZKfyZTSc+y1rdR3xZhlpgYCT95rX2hZ04oSvCqdDccftizngBFIBW13ie3lN00mtqhipRiq9du
6bHsa2c11EKfSDllg39VaD10ukujH/rfBcozvF7w6P/FcUpW5t89mlzqowutozMoY71h7m2+Yhcb7281PvytuTSdqZY1sMmGaPjubNx0uj/
8947koq7Ga7foraV1H2H2meLPG2h929yZw1K+R+Dzdd9sx+AfjZ7EelioniaTur2k9phLPOH6a6VbC4ysRaDGJEqwoC3VxlaqMABgwcf
elRw8uSOGDj77kuoqKSzxBVdtpeloyhMNW7ketZjkBv1vtKSM5yrlFgydWALWNB2ynohF41E9MpJHfiwUS/1157Kqk/MqheTaNky0Q1T552
zFnab1WR/vRGBFWijrIISp/6UjgdR1A/3Sz+hqet/MKZupZ04tx+ksepeAVZnxup0crC9vsAjTHA0cj6pa7vhLBZhtT0eBbb1jrgc24Pob
jFnm06w3Hayj775JBoRe+1g/hFJ0ttqjeDWIF6c/AayFjSTLWh7F04GRwxLFEB9A0DfbKRueYff08A2w9Vd5jbHEUhcX4c7 wahvHtMhugA
lFPtud0W1tv30AU=

```

Locations

User Pools Authorizer Request:
GET <https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations>
Headers: > {Authorization: Array(1)}
Body: null

We get the same set of tokens as before.

Vysor File Edit Window

Developer Tools - file:///android_asset/www/index.html

top | Filter Default levels

AWS ACCESS KEY ID: ASIAJLP/WUUSQAVAZLAZRA main.js:15:092
AWS Secret Access Key: 7wTy3FeJsaglAIrqOsPxASUjWlw1JnZFgj2qUzb+ main.js:694
AWS Session Token: AgoB3JpZ2luEFsaCXVzLwVh3QtMSKAAhNtkKnTbsUh687wugfG9nxYP/xxdVcN3pfaJ7WYQ3X438eZqByvUQFLDOL8m:NDGtpWcqLu3xUvWld2umTHsbhIPgCfHp2udLnJnsNDTLnXq0k5RLPnZixA1a7fusJNqa0R2YDx9F9rGyj9aIsuXXsePvbYm6FaIlf5njgbzqYFTbDMyyo0oTlQbIBpyhkV45xhaeIHivGY4RKz1wU6U26vrdbbgCx2HsxfllibyS105ickyR6DlxIcPl3fdwFlhbedpjz2vvn6tYht31VbPuc8JLRAe dFPJHe1/2jaocD2TcnZrhC5c3Jxy41uumTs@9Y4rRHMm1flbEhqhgYsP//////////ARAAggw3NDiZnJA1MDgzmzc1DPBg+Ax19otfekX8yraBfP0ngFPu6anwrBZ1kL7EP+0BRjRgyvS90G70DqR/eCeEKTptkrM2cYMHsh02X3aUPB8NA0L500RwykaK4+mhe6V5RFk0fVzKJb05sxcgL88dxZ134sd7M/YCgjwcyXZKfy5tCzHlgpYct9srx2hZ04oSLvCqdDccftizngBFIB13ie31N00mtqipR1q9dueuRKhzhEuZUwrfJFGuPn@0C3Y5nx6jfqhWj15s5uxFAQ3cLGc4qY21EyGKzcIbaHbuVny1CknFPD8gwmlIX9pvWE5Jcyuvvh1i94560NEL27HdJ1815hydNKnq7etZjK0vivtKSMl5yrLfgydWALMNBynohF41E9MpJHfiwUs/1157Kqk/MqneTaKy0Q1T5526bhSa2c11EkFSdlg39vab10kujgh/rFBcozvF7w6P/FcUpw58mlzqowutozMoY71h7m2+Yhcb7281PvtytT5DqZY1sMmGpJubNx0uj/894TkoqTGA7eoraViH2h2meLPG2h929yZw1Krh+Dzd9sx+AnFjZ7EelnhonIaTur2k9phLPP0H6a6vbC4YsRa0GJEowwC3VxlaqABgWwfelRwBuS0G0J77kuogK5zx8vd1peloymMNw7etZjK0vivtKSMl5yrLfgydWALMNBynohF41E9MpJHfiwUs/1157Kqk/MqneTaKy0Q1T552zFnaiW/R/vRGBFWIriISP/6UjgdR1A/35z+hqet/MKzuPzD4tx+kspe0AVznvup0crwC9vsAjTHA0cj6Pa7vh18FZh70e8bbijrge24PobjNm06w3Hayj775jBboRe+1g/hFJ0ttqjeDWIF6c/AayFjSTLWh7FQ4GRwxLFEB9A0fbKRueYff08Azw9Vd5jbHEUhcX4c7wahwHTmHugAlvPTud0W1tv30AU=

Locations main.js:101

User Pools Authorizer Request: GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations main.js:3342
 Headers: ▶ {Authorization: Array(1)}
 Body: null

Sign Out main.js:101

Cognito Hosted UI: <https://spacefinder-development-us-east-1-iaovdy202.auth.us-east-1.amazonc> main.js:2589 _cefinder://callback&response_type=code&client_id=3lpyv74d4dn1fu92p5ntolim

Console

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Next, let us sign out and try a SAML2.0 compliant provider like Okta below

Vysor File Edit Window

Developer Tools - file:///android_asset/www/index.html

top | Filter Default levels

AWS ACCESS KEY ID: ASIAJLP/WUUSQAVAZLAZRA main.js:15:092
AWS Secret Access Key: 7wTy3FeJsaglAIrqOsPxASUjWlw1JnZFgj2qUzb+ main.js:694
AWS Session Token: AgoB3JpZ2luEFsaCXVzLwVh3QtMSKAAhNtkKnTbsUh687wugfG9nxYP/xxdVcN3pfaJ7WYQ3X438eZqByvUQFLDOL8m:NDGtpWcqLu3xUvWld2umTHsbhIPgCfHp2udLnJnsNDTLnXq0k5RLPnZixA1a7fusJNqa0R2YDx9F9rGyj9aIsuXXsePvbYm6FaIlf5njgbzqYFTbDMyyo0oTlQbIBpyhkV45xhaeIHivGY4RKz1wU6U26vrdbbgCx2HsxfllibyS105ickyR6DlxIcPl3fdwFlhbedpjz2vvn6tYht31VbPuc8JLRAe dFPJHe1/2jaocD2TcnZrhC5c3Jxy41uumTs@9Y4rRHMm1flbEhqhgYsP//////////ARAAggw3NDiZnJA1MDgzmzc1DPBg+Ax19otfekX8yraBfP0ngFPu6anwrBZ1kL7EP+0BRjRgyvS90G70DqR/eCeEKTptkrM2cYMHsh02X3aUPB8NA0L500RwykaK4+mhe6V5RFk0fVzKJb05sxcgL88dxZ134sd7M/YCgjwcyXZKfy5tCzHlgpYct9srx2hZ04oSLvCqdDccftizngBFIB13ie31N00mtqipR1q9dueuRKhzhEuZUwrfJFGuPn@0C3Y5nx6jfqhWj15s5uxFAQ3cLGc4qY21EyGKzcIbaHbuVny1CknFPD8gwmlIX9pvWE5Jcyuvvh1i94560NEL27HdJ1815hydNKnq7etZjK0vivtKSMl5yrLfgydWALMNBynohF41E9MpJHfiwUs/1157Kqk/MqneTaKy0Q1T5526bhSa2c11EkFSdlg39vab10kujgh/rFBcozvF7w6P/FcUpw58mlzqowutozMoY71h7m2+Yhcb7281PvtytT5DqZY1sMmGpJubNx0uj/894TkoqTGA7eoraViH2h2meLPG2h929yZw1Krh+Dzd9sx+AnFjZ7EelnhonIaTur2k9phLPP0H6a6vbC4YsRa0GJEowwC3VxlaqABgWwfelRwBuS0G0J77kuogK5zx8vd1peloymMNw7etZjK0vivtKSMl5yrLfgydWALMNBynohF41E9MpJHfiwUs/1157Kqk/MqneTaKy0Q1T552zFnaiW/R/vRGBFWIriISP/6UjgdR1A/35z+hqet/MKzuPzD4tx+kspe0AVznvup0crwC9vsAjTHA0cj6Pa7vh18FZh70e8bbijrge24PobjNm06w3Hayj775jBboRe+1g/hFJ0ttqjeDWIF6c/AayFjSTLWh7FQ4GRwxLFEB9A0fbKRueYff08Azw9Vd5jbHEUhcX4c7wahwHTmHugAlvPTud0W1tv30AU=

Locations main.js:101

User Pools Authorizer Request: GET https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations main.js:3342
 Headers: ▶ {Authorization: Array(1)}
 Body: null

Sign Out main.js:101

Cognito Hosted UI: <https://spacefinder-development-us-east-1-iaovdy202.auth.us-east-1.amazonc> main.js:2589 _cefinder://callback&response_type=code&client_id=3lpyv74d4dn1fu92p5ntolim

Console

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Vysor File Edit Window

Developer Tools - file:///android_asset/www/index.html

top Filter Default levels

AWS ACCESS KEY ID: ASIAJLPWU00USQAVIAZLA **main.js:151094**

AWS Secret Access Key: 7wTy3FeJsaglAIrq0sPxASUjWLwiJnZFgj2qUzb+ **main.js:694**

AWS Session Token: AgoG3JpZ2luEfsaCXvzLWvhc30tMSKAAhNtkKnTbsUh6B7wugfuG9nxYP/xxdvcn3pfaj7WYQ3X438ezQmByvUQFLD0L8m1NDGtpWcqLu3xUvWldzumTHsbhIPgCfHp2udLnjsN0TLnxQk05R1PnZixA14a7fusJNqa0R2YDx9F9GYj09AIsuXXsePvbYm6FaTLFSnJgbzqYFtDMv0o0TlQbIbphykV45xhaeIHivGY4RKz1wU6UZ6vrdbbgXc2Hsxflliby5f105ickyR60lxIcPl3fdwIhbedjpj2vvgvnTYht31VbPuc8j/LRaeDPJHe1/2ja0CD2TcnZrhHC5c3Jxy4iUmnTs09YrY4RHmnlfbEghgYIsP//////////ARAAAGw3ND1zNjA1M0gzmzc1DPB8g+gAx190TfekXByraBFpQngFPu6anwrBZ1k1L7EP+0BRjRgyvuS9QG70Dqr/eCeEKTptkrM2cYMHshz02X3aUPB8NAQl50oRwykaK4+mhe6V5RFkqFvZKJob5sxclL88dxZ134sd7wM/KQwH9ATr9qPVPr+jtLUAl5ale6Fz34Ze7SE3U082tV3+0+dgPLBrMnR15Ad3gPzuVGNfZQMbNxIHC48ec1tCPueuRKzhzEUZUwr0jFGuPn0C03Y5nX6f/qhWjJ5sA5uxFAQ3cLGc4qY2YiEygGkzcIbaHbUvny1CknFPD8gwmlLIX9pvWE5Jcyuvvh19458GNEL2THzdJ1815hydNdkn027mYCgjwcfY7sc+y1drZLhlgYct5rXh2zQ04oSLvCqdccFTizngBF1WB13ie3LN00mthqipRiq9du6bhSa2c11EkfSD0llg39va0i0kujgh/rFBcozvF7w6P/FcuPw5t89mlzqowutoMoY71h7m2+Yhcb7281pvytuTS0oZy1sMmgapubNx0uj/894TkoqTGA7EoraV1H2H2meLPG2H929yZw1KRp+Hdzzd9sx+AnFj27EelnionIaTur2kPhLPP0H6a6VbC4YsRaDGjeowC3VxlaqMABgWwcfelRwBuS0Gdj77kuoqKszxBvdpeLoyhMNW7ketZjK0vivtKSM5yrlFgydwALNB2ynohF41E9MpJHfiw0s/1157Kqk/MqheTaNky001T552zFnab2IWR/vRBPFWjriISP/6UjgdR1A/3Sz+hqet/MKZuPzD4tx+ksp0AVzvxup0crwC9vsAjTHAcj6Pa7vh1BFZhT0e8bbijrgc24PobjFnm06w3Hayj775jbboRe+1g/hFJ0ttajeDWIF6c/AayFjSTLWh7FQ4GRwxLFEB9A0fbKRueYff08Azw9Vd5jbHEUhcX4c7wahvHtMhugAlvPTud0w1tv30AU=

Locations **main.js:101**

User Pools Authorizer Request: **main.js:3342**

GET <https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations>

Headers: ▶ {Authorization: Array(1)}

Body: null

Sign Out **main.js:101**

Cognito Hosted UI: <https://spacefinder-development-us-east-1-iaovdw2o2.auth.us-east-1.amazonc0 main.js:2589>

[_efinder://callback&response_type=code&client_id=31pyq74d4dn1lfu92p5qtolim](#)

Console

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Vysor File Edit Window

Developer Tools - file:///android_asset/www/index.html

top Filter Default levels

AWS ACCESS KEY ID: ASIAJLPWU00USQAVIAZLA **main.js:151094**

AWS Secret Access Key: 7wTy3FeJsaglAIrq0sPxASUjWLwiJnZFgj2qUzb+ **main.js:694**

AWS Session Token: AgoG3JpZ2luEfsaCXvzLWvhc30tMSKAAhNtkKnTbsUh6B7wugfuG9nxYP/xxdvcn3pfaj7WYQ3X438ezQmByvUQFLD0L8m1NDGtpWcqLu3xUvWldzumTHsbhIPgCfHp2udLnjsN0TLnxQk05R1PnZixA14a7fusJNqa0R2YDx9F9GYj09AIsuXXsePvbYm6FaTLFSnJgbzqYFtDMv0o0TlQbIbphykV45xhaeIHivGY4RKz1wU6UZ6vrdbbgXc2Hsxflliby5f105ickyR60lxIcPl3fdwIhbedjpj2vvgvnTYht31VbPuc8j/LRaeDPJHe1/2ja0CD2TcnZrhHC5c3Jxy4iUmnTs09YrY4RHmnlfbEghgYIsP//////////ARAAAGw3ND1zNjA1M0gzmzc1DPB8g+gAx190TfekXByraBFpQngFPu6anwrBZ1k1L7EP+0BRjRgyvuS9QG70Dqr/eCeEKTptkrM2cYMHshz02X3aUPB8NAQl50oRwykaK4+mhe6V5RFkqFvZKJob5sxclL88dxZ134sd7wM/KQwH9ATr9qPVPr+jtLUAl5ale6Fz34Ze7SE3U082tV3+0+dgPLBrMnR15Ad3gPzuVGNfZQMbNxIHC48ec1tCPueuRKzhzEUZUwr0jFGuPn0C03Y5nX6f/qhWjJ5sA5uxFAQ3cLGc4qY2YiEygGkzcIbaHbUvny1CknFPD8gwmlLIX9pvWE5Jcyuvvh19458GNEL2THzdJ1815hydNdkn027mYCgjwcfY7sc+y1drZLhlgYct5rXh2zQ04oSLvCqdccFTizngBF1WB13ie3LN00mthqipRiq9du6bhSa2c11EkfSD0llg39va0i0kujgh/rFBcozvF7w6P/FcuPw5t89mlzqowutoMoY71h7m2+Yhcb7281pvytuTS0oZy1sMmgapubNx0uj/894TkoqTGA7EoraV1H2H2meLPG2H929yZw1KRp+Hdzzd9sx+AnFj27EelnionIaTur2kPhLPP0H6a6VbC4YsRaDGjeowC3VxlaqMABgWwcfelRwBuS0Gdj77kuoqKszxBvdpeLoyhMNW7ketZjK0vivtKSM5yrlFgydwALNB2ynohF41E9MpJHfiw0s/1157Kqk/MqheTaNky001T552zFnab2IWR/vRBPFWjriISP/6UjgdR1A/3Sz+hqet/MKZuPzD4tx+ksp0AVzvxup0crwC9vsAjTHAcj6Pa7vh1BFZhT0e8bbijrgc24PobjFnm06w3Hayj775jbboRe+1g/hFJ0ttajeDWIF6c/AayFjSTLWh7FQ4GRwxLFEB9A0fbKRueYff08Azw9Vd5jbHEUhcX4c7wahvHtMhugAlvPTud0w1tv30AU=

Locations **main.js:101**

User Pools Authorizer Request: **main.js:3342**

GET <https://ye07xoo3oc.execute-api.us-east-1.amazonaws.com/development/locations>

Headers: ▶ {Authorization: Array(1)}

Body: null

Sign Out **main.js:101**

Cognito Hosted UI: <https://spacefinder-development-us-east-1-iaovdw2o2.auth.us-east-1.amazonc0 main.js:2589>

[_efinder://callback&response_type=code&client_id=31pyq74d4dn1lfu92p5qtolim](#)

Console

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Vysor File Edit Window

Developer Tools - file:///android_asset/www/index.html

SIGN-UP SIGN-IN (CUSTOM UI) SIGN IN (HOSTED UI)

Welcome to SpaceFinder!

Signing in...

Waiting a remote office? You can book a desk in advance.

AWS ACCESS KEY ID: 1A14P1WU00UQAVIAZLA

AWS Secret Access Key: 7wTy3FeJsglAIrqOsPxASUjWLwiJnZFgj2qUzb+ main.js:694

AWS Session Token:

AgogB3JpZ2LuEfsaCXVzLWvhc30tMSKAhhNtkKnTbsUh6B7wugfuG9nxYP/xxdVCn3pfaj7WY0Q3X43ezQmByvUQFLD0L8m1NDGtpWcqLu3xUVldzumTHsBH1PgCfhp2udLnJsnDNLnxq0k5RLnPz1xa14a7fusJNqa0R2Y0x9Fr9Gyj09AisuxXsePvbYm6FaTLF5nJgbzqYFTb0Mvyo0oTlqbIbpyhkV45xHaeIHivGY4RkZ1wU6UZ6vrdedbXcZhsxflLlibyf185ickyR60lxclp13dufIhbedjpj2vvn6TYht31vbPucBJ/LRAedFPJHei/2jaaoCD2TcnZrhC5c3Jxy41uumts09Yy4RHMmlfbEghgYIsP//////////ARAAGw3NDIzNjA1MDgZmc1DPBg+gAx190TfkExByraBTPQngFPu6anwrBZ1kL7EP+0BRjRgyvu590G70d0r/eCeEKTptkrM2cYMHshz02X3aUPBNAQl500rwkaK4+mhe6V5RFk0fVzKJob5sxclg88dXZ134sd7w/MK0wH9ATr9qPVPr+jtluAlI5ale6Fz7SE3U08ZtV3+00+dgPl8rMnR15Ad3gPzuV6NFZQMbNxIHC4BecltCPuyeuRKzhzEUlwrojFGuPn0CQ3Y5nX6jfqHwJ5sA5uxFAQ3cLGc4qY2Y1EyG6kzcIbahbUvNy1CknFPDBqwwmlX9pvwESJcyuvvH19450GNEL27H2d1815hydNxmq027mYcjwgcYXZKfTzSc+y1rd3xZLhp2z040SLvCqdDccfTizingBFITWB13ie3LN00thq1qpRiq9du6bhsa2c1iEkfSDllg39Va0iIOkughj/rFBcozvF7w6P/FcuUp5189emlzbqowutozMoY71h7m2+Yhcb7281PyvtuTSQdzY1sMmGApjubNxu0j894TkoqTga7EoraV1H2H2meLPG2H929yZw1kR+HDzdd9sx+AnF1Z7EelnlionIaTur2kPhLPP0H6a6vbC4YsRaDGJEowwC3VxlaMABgwcfelRwBuSOG0j77kuoqKSzxVdTpeloyhMNW7ketZjK0vivtKSM5lyrFgydWALWNB2ynohF4IE9MpJHfiwUS/i157Kqk/MqheTaNky0Q1T552zFnab1wR/vRGBFWjriiSP/6UjgdR1A/3Sz+hget/MKZupz04tx+kspc0AVzvnxup0crwC9vsAJTHAQCj6Pa7vh18FZht0e8bbijrge24PobjNm06w3Hayj775jBboRe+1g/hFJ0ttqjeDWIF6c/AayFjSTLWh7FQ4GRwxLFEB9A0DfbKRueYff08Azw9Vd5jbHEUhcX4c7wahvHtMugAlvPTud0w1tv30AU=

Locations

User Pools Authorizer Request:

GET https://ye0x0o3c.execute-api.us-east-1.amazonaws.com/development/locations

Headers: ► {Authorization: Array(1)}

Body: null

Sign Out

Cognito Hosted UI: https://spacefinder-development-us-east-1-iaovdw2o2.auth.us-east-1.amazoncognito/main.js:2589

cefinder://callback&response_type=code&client_id=31pvq74d4dn1fu92p5tolim

Console

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Vysor File Edit Window

Developer Tools - file:///android_asset/www/index.html

SIGN-UP SIGN-IN (CUSTOM UI) SIGN IN (HOSTED UI)

Welcome to SpaceFinder!

Success!

You are now signed in.

Username: Okta_justin-okta@aws-demos.com

First name: Justin

Last name: Pirtle

OK

Forgot password? Book a desk in advance.

AWS ACCESS KEY ID: 1A14P1WU00UQAVIAZLA

AWS Secret Access Key: 7wTy3FeJsglAIrqOsPxASUjWLwiJnZFgj2qUzb+ main.js:694

AWS Session Token:

AgogB3JpZ2LuEfsaCXVzLWvhc30tMSKAhhNtkKnTbsUh6B7wugfuG9nxYP/xxdVCn3pfaj7WY0Q3X43ezQmByvUQFLD0L8m1NDGtpWcqLu3xUVldzumTHsBH1PgCfhp2udLnJsnDNLnxq0k5RLnPz1xa14a7fusJNqa0R2Y0x9Fr9Gyj09AisuxXsePvbYm6FaTLF5nJgbzqYFTb0Mvyo0oTlqbIbpyhkV45xHaeIHivGY4RkZ1wU6UZ6vrdedbXcZhsxflLlibyf185ickyR60lxclp13dufIhbedjpj2vvn6TYht31vbPucBJ/LRAedFPJHei/2jaaoCD2TcnZrhC5c3Jxy41uumts09Yy4RHMmlfbEghgYIsP//////////ARAAGw3NDIzNjA1MDgZmc1DPBg+gAx190TfkExByraBTPQngFPu6anwrBZ1kL7EP+0BRjRgyvu590G70d0r/eCeEKTptkrM2cYMHshz02X3aUPBNAQl500rwkaK4+mhe6V5RFk0fVzKJob5sxclg88dXZ134sd7w/MK0wH9ATr9qPVPr+jtluAlI5ale6Fz7SE3U08ZtV3+00+dgPl8rMnR15Ad3gPzuV6NFZQMbNxIHC4BecltCPuyeuRKzhzEUlwrojFGuPn0CQ3Y5nX6jfqHwJ5sA5uxFAQ3cLGc4qY2Y1EyG6kzcIbahbUvNy1CknFPDBqwwmlX9pvwESJcyuvvH19450GNEL27H2d1815hydNxmq027mYcjwgcYXZKfTzSc+y1rd3xZLhp2z040SLvCqdDccfTizingBFITWB13ie3LN00thq1qpRiq9du6bhsa2c1iEkfSDllg39Va0iIOkughj/rFBcozvF7w6P/FcuUp5189emlzbqowutozMoY71h7m2+Yhcb7281PyvtuTSQdzY1sMmGApjubNxu0j894TkoqTga7EoraV1H2H2meLPG2H929yZw1kR+HDzdd9sx+AnF1Z7EelnlionIaTur2kPhLPP0H6a6vbC4YsRaDGJEowwC3VxlaMABgwcfelRwBuSOG0j77kuoqKSzxVdTpeloyhMNW7ketZjK0vivtKSM5lyrFgydWALWNB2ynohF4IE9MpJHfiwUS/i157Kqk/MqheTaNky0Q1T552zFnab1wR/vRGBFWjriiSP/6UjgdR1A/3Sz+hget/MKZupz04tx+kspc0AVzvnxup0crwC9vsAJTHAQCj6Pa7vh18FZht0e8bbijrge24PobjNm06w3Hayj775jBboRe+1g/hFJ0ttqjeDWIF6c/AayFjSTLWh7FQ4GRwxLFEB9A0DfbKRueYff08Azw9Vd5jbHEUhcX4c7wahvHtMugAlvPTud0w1tv30AU=

Locations

User Pools Authorizer Request:

GET https://ye0x0o3c.execute-api.us-east-1.amazonaws.com/development/locations

Headers: ► {Authorization: Array(1)}

Body: null

Sign Out

Cognito Hosted UI: https://spacefinder-development-us-east-1-iaovdw2o2.auth.us-east-1.amazoncognito/main.js:2589

cefinder://callback&response_type=code&client_id=31pvq74d4dn1fu92p5tolim

Console

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cogni... JSON Web Tokens - jwt.io API Gateway New Tab Search

User Pools | Federated Identities

Attributes: Domain names can only contain lower-case letters, numbers, and hyphens. Learn more about domain prefixes.

Policies

MFA and verifications

Advanced security beta

Message customizations

Tags

Devices

App clients

Triggers

Analytics

Go to summary

Customize UI

Domain prefix: https://spacefinder-development-us-east-1-jaovdw2o2.auth.us-east-1.amazoncognito.com

Delete domain

App integration

App client settings

Domain name

UI customization

Resource servers

Federation

Identity providers

Attribute mapping

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cogni... JSON Web Tokens - jwt.io API Gateway New Tab Search

User Pools | Federated Identities

App client spacefinder-development-userPool-app-client

ID 31pvq74d4dn1fu92p5qt0im

Enabled Identity Providers Select all

Google Okta Cognito User Pool

Sign in and sign out URLs

Enter your callback URLs below that you will include in your sign in and sign out requests. Each field can contain multiple URLs by entering a comma after each URL.

Callback URL(s)

http://localhost:8100, spacefinder://callback

Sign out URL(s)

OAuth 2.0

Select the OAuth flows and scopes enabled for this app. Learn more about flows and scopes.

Allowed OAuth Flows

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cogni... JSON Web Tokens - jwt.io API Gateway New Tab Search

https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1_laovdw2o2/app-integration-app-settings/jpirle US East (Virginia) Support

Callback URL(s)
http://localhost:8100, spacefinder://callback

Sign out URL(s)

OAuth 2.0

Select the OAuth flows and scopes enabled for this app. [Learn more about flows and scopes.](#)

Allowed OAuth Flows

Authorization code grant Implicit grant Client credentials

Allowed OAuth Scopes

phone email openid aws.cognito.signin.user.admin profile

[Go to summary](#) Choose domain name

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cogni... JSON Web Tokens - jwt.io API Gateway New Tab Search

https://console.aws.amazon.com/cognito/users/?region=us-east-1#/pool/us-east-1_laovdw2o2/federation-identity-providers/jpirle US East (Virginia) Support

Identity providers

Policies MFA and verifications Advanced security ^{beta} Message customizations Tags Devices App clients Triggers Analytics

App integration

App client settings Domain name UI customization Resource servers

Federation

Identity providers Attribute mapping

Select and configure the external identity providers you want to enable. You will also need to choose which identity providers to enable for each app on the Apps settings tab under App integration. [Learn more about identity federation with Cognito User Pools.](#)

Facebook

Google

Login with Amazon

SAML

[Go to summary](#) Configure attribute mapping

The screenshot shows the AWS User Pools console under the 'Federated Identities' section. On the left sidebar, 'Identity providers' is selected. In the main area, there's a 'Facebook' card with a blue 'f' icon. To its right, there are input fields for 'Facebook app ID' (containing '1234567890123456'), 'App secret' (containing '123456789b00def123456a12345678d1'), and 'Authorize scope' (containing 'public_profile,email'). A blue 'Enable Facebook' button is at the bottom. Other identity provider cards like Google and Login with Amazon are visible above the SAML card.

Register an application on the Facebook developer portal, then come and paste in the app ID and secret in the form above, and the scopes you want to allow.

The screenshot shows the AWS User Pools console under the 'Federated Identities' section. On the left sidebar, 'Identity providers' is selected. In the main area, there's a 'SAML' card with a blue user icon and the word 'SAML'. Below it, a box contains the text: 'You can use a corporate identity provider to sign in users through SAML federation.' There are two options for providing metadata: 'Select file' (with a blue file icon) or 'Provide metadata document endpoint URL'. A 'Provider name:' field is also present. Other identity provider cards like Facebook, Google, and Login with Amazon are visible above the SAML card.

The screenshot shows the AWS User Pools console under the Federated Identities section. On the left, there are tabs for 'Identity providers' (which is selected) and 'Attribute mapping'. In the center, there's a large blue circular icon with a person icon labeled 'SAML'. Below it, a text box says: 'You can use a corporate identity provider to sign in users through SAML federation.' followed by a link 'Learn more about SAML.'. To the right, there are two input fields: one for 'Select file' and another for 'Provide metadata document endpoint URL'. Below these is a 'Provider name:' field containing 'jpirtle'. Underneath is a 'Identifiers (optional)' field with an empty input box. At the bottom right is a 'Create provider' button.

This screenshot shows the same AWS User Pools interface, but the 'Identity providers' tab is now active. It displays a configuration form for a provider named 'Okta'. The 'Identifiers (optional)' field is empty. Below it is a 'Metadata document' field containing the URL 'https://dev-157075.oktapreview.com/app/exkcro25/dXwXCC'. The 'Provider name:' field contains 'Okta'. The 'Identifiers (optional)' field has the letter 'I' typed into it. At the bottom right are 'Cancel editing' and 'Update provider' buttons.

You need to provide the metadata document from your SAML2.0 provider, along with the Identifiers details that is a particular string.

Screenshot of the AWS User Pools console showing attribute mapping for Google. The left sidebar lists various settings like General settings, Users and groups, Attributes, Policies, etc. The main area shows how to map identity provider attributes to user pool attributes for Google.

How do you want to map identity provider attributes to user pool attributes?

You can map user attributes from external identity providers to populate user attributes in this user pool. [Learn more about attribute mapping.](#)

Capture	Google attribute	User pool attribute
<input type="checkbox"/>	names	
<input type="checkbox"/>	genders	
<input type="checkbox"/>	birthdays	
<input type="checkbox"/>	phoneNumbers	
<input checked="" type="checkbox"/>	access_token	custom:ext_idp_access_tkn
<input type="checkbox"/>	token_type	

Screenshot of the AWS User Pools console showing attribute mapping for Google. The left sidebar lists various settings like App clients, Triggers, Analytics, App integration, Federation, Identity providers, and Attribute mapping. The Attribute mapping section is highlighted.

<input type="checkbox"/>	genders	
<input type="checkbox"/>	birthdays	
<input type="checkbox"/>	phoneNumbers	
<input checked="" type="checkbox"/>	access_token	custom:ext_idp_access_tkn
<input type="checkbox"/>	token_type	
<input checked="" type="checkbox"/>	expires_in	custom:ext_idp_expires_in
<input checked="" type="checkbox"/>	refresh_token	custom:ext_idp_refresh_tkn
<input checked="" type="checkbox"/>	email	Email
<input checked="" type="checkbox"/>	email_verified	Email Verified
<input checked="" type="checkbox"/>	name	Name

We can map Google's provided attributes to user pool attributes as above

The screenshot shows the AWS User Pools console under the 'Federated Identities' section. On the left, a sidebar lists various options like Policies, MFA and verifications, Advanced security, Message customizations, Tags, Devices, App clients, Triggers, Analytics, App integration, Federation, Identity providers, and Attribute mapping. The 'Attribute mapping' option is highlighted. The main area shows a configuration for mapping SAML attributes from Okta to a user pool. It includes tabs for Facebook, Google, Amazon, and SAML, with SAML selected. A dropdown menu for 'Okta' is open. Below it, there are three rows of mappings:

Capture	SAML attribute	User pool attribute
<input checked="" type="checkbox"/>	firstName	Given Name
<input checked="" type="checkbox"/>	lastName	Family Name
<input checked="" type="checkbox"/>	email	Email

Below these rows is a button labeled 'Add SAML attribute'. At the bottom right of the main area is a 'Go to summary' link.

SAML attributes can also be mapped as above

The screenshot shows the AWS User Pools console under the 'Users and groups' section. On the left, a sidebar lists various options like Attributes, Policies, MFA and verifications, Advanced security, Message customizations, Tags, Devices, App clients, Triggers, Analytics, App integration, Federation, and Identity providers. The 'User Pools' option is highlighted. The main area displays a list of users with columns for Username, Enabled, Status, Updated, and Created. There are buttons for 'Import users' and 'Create user' at the top of the list table.

Username	Enabled	Status	Updated	Created
Google_106300170295478073663	Enabled	EXTERNAL_PROVIDER	Nov 28, 2017 5:11:01 PM	Nov 28, 2017 9:51:17 AM
Okta_justin-okta@aws-demos.com	Enabled	EXTERNAL_PROVIDER	Nov 28, 2017 5:12:42 PM	Nov 28, 2017 9:47:58 AM
Vladimir	Enabled	CONFIRMED	Nov 28, 2017 10:44:58 PM	Nov 28, 2017 10:44:23 PM
admin1	Enabled	CONFIRMED	Nov 28, 2017 9:30:26 AM	Nov 28, 2017 9:30:24 AM
user1	Enabled	CONFIRMED	Nov 28, 2017 9:30:26 AM	Nov 28, 2017 9:30:24 AM

So, when we look at our users in the User Pool as above, we can see both custom attributes as well as mapped 3rd party attributes.

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cogni... JSON Web Tokens - jwt.io API Gateway New Tab Search

User Pools | Federated Identities Jpirtle US East (Virginia) Support

spacefinder-development-userPool

General settings

Users and groups

Attributes

Policies

MFA and verifications

Advanced security beta

Message customizations

Tags

Devices

App clients

Triggers

Analytics

App integration

App client settings

Domain name

Users > Google_106300170295478073663

Add to group Reset password Enable MFA Disable user

Groups: us-east-1_laovdw2o2_Google

User Status: Enabled / EXTERNAL_PROVIDER

SMS Status: Disabled

Last Modified: Nov 28, 2017 5:11:01 PM

Created: Nov 28, 2017 9:51:17 AM

custom:ext_idp_access_tkn:ya29.GIwSBXG-PWdRhkUp1a6fDSILII1q66Gtj2zrnq8Yqc9wT7FfMGLngia42OYSALozApCflTAe-
DzNkx3cjRVQ4OEfpFR1EZdbK3ew_M_N7-580oigjzGvouhJkhJ2w

Firefox File Edit View History Bookmarks Tools Window Help

Amazon WorkMail User Pools - Amazon Cogni... JSON Web Tokens - jwt.io API Gateway New Tab Search

User Pools | Federated Identities Jpirtle US East (Virginia) Support

Domain name

UI customization

Resource servers

Federation

Identity providers

Attribute mapping

custom:ext_idp_access_tkn:ya29.GIwSBXG-PWdRhkUp1a6fDSILII1q66Gtj2zrnq8Yqc9wT7FfMGLngia42OYSALozApCflTAe-
DzNkx3cjRVQ4OEfpFR1EZdbK3ew_M_N7-580oigjzGvouhJkhJ2w

sub: 760c87b5-e9ce-4443-97bc-02cb5b992008

identities: [{"userId": "106300170295478073663", "providerName": "Google", "providerType": "Google", "issuer": null, "primary": true, "dateCreated": "1511862677390"}]

email_verified: true

custom:ext_idp_expires_in: 3600

name: Justin Pirtle

given_name: Justin

family_name: Pirtle

email: justin-google@aws-demos.com

Screenshot of the AWS User Pools console showing the configuration for the "spacefinder-development-userPool". The left sidebar lists various settings like General settings, Users and groups, Attributes, Policies, MFA and verifications, Advanced security (beta), Message customizations, Tags, Devices, App clients, Triggers, Analytics, and App integration. The main content area is titled "What customizations do you want to make to the end-user experience?" and includes sections for App client to customize (set to "Defaults for all clients without individual settings") and Logo (optional) with a file upload field.

Screenshot of the AWS User Pools console showing the UI customization section. The "UI customization" tab is selected. It displays a grid of customization options: Labels, Error messages, Input fields, Identity provider buttons, Text descriptions, Identity provider descriptions, Submit button, Legal text, Banner, and Logo. Each option has a "Customize..." link next to it.

You can customize the Hosted UI.



Migration approach #1: Bulk import

(1) Create CSV

- Doesn't contain passwords
- Max 100,000 users at a time

(2) Run the import job

```
$ aws cognito-idp create-user-import-job  
  
$ curl -v -T "path/to/csvfile" -H "x-amz-server-side-encryption:aws:kms"  
"PRE_SIGNED_URL"  
  
$ aws cognito-idp start-user-import-job
```

```
cognito:mfa_enabled  
cognito:username  
phone_number  
phone_number_verified  
email  
email_verified  
name  
given_name  
family_name  
middle_name  
nickname  
preferred_username  
profile  
picture  
website  
gender  
birthdate  
zoneinfo  
locale  
address  
updated_at
```



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Migration approach #1: Bulk import

(1) Create CSV

- Doesn't contain passwords
- Max 100,000 users at a time

(2) Run the import job

(3) Users change passwords on initial login

```
cognito:mfa_enabled  
cognito:username  
phone_number  
phone_number_verified  
email  
email_verified  
name  
given_name  
family_name  
middle_name  
nickname  
preferred_username  
profile  
picture  
website  
gender  
birthdate  
zoneinfo  
locale  
address  
updated_at
```



© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Migration approach #2: One-at-a-time

This approach migrates users one at a time as they sign-in to your app:

- (1) First, try authenticating against Cognito User Pools
- (2) If that fails because of “User Not Found”, authenticate against the former IdP
- (3) If authentication with former IdP is successful, then create user in the Cognito User Pool with the same username/password

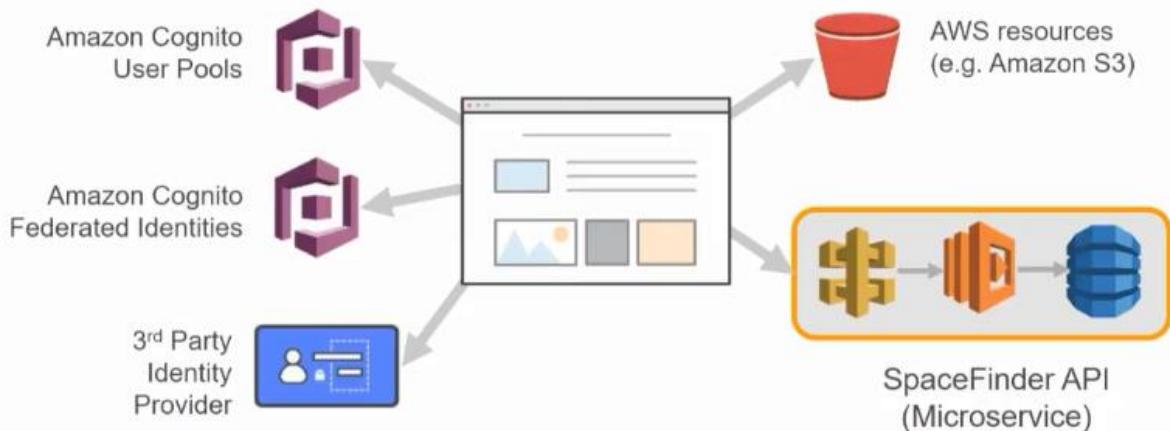
Wrap up



SpaceFinder mobile app



SpaceFinder web app



AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



SpaceFinder



Do try this at home

- Mobile app + API are open-sourced (Apache 2.0 license)

[https://github.com/awslabs/
aws-serverless-auth-reference-app](https://github.com/awslabs/aws-serverless-auth-reference-app)

AWS re:Invent

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Related Sessions

- **MBL305** – Implement User Onboarding, Sign-Up, and Sign-In for Mobile and Web Applications with Amazon Cognito
- **SID332** – Identity Management for Your Users and Apps: A Deep Dive on Amazon Cognito
- **SID343** – User Management and App Authentication with Amazon Cognito
- **SRV425** – Serverless OAuth: Authorizing 3rd-party Applications to your Serverless API

Remember to
complete your
evaluations

