Cognito?

What is it, and why do I care?

Disclaimers

I do not work for Amazon

My use of Cognito is at the “evaluation” level, not production

All trademarks belong to their owners, whether or not identified

Cognito is a collection of services

Identity Pools (IDP)

- Database of users
- Provides authentication, token-based authorization, 2FA
- Handles communications for password resets, lost passwords, …

Federated Identities

- Allow users to login with external identity providers
- Associate IAM roles with federated identity tokens

Sync

- Store user-relevant data and share across devices

Classified as a “Mobile” service

Provides several services useful in possibly-untrusted clients/networks

- Time-limited AWS keys
- Synchronization of user data across devices
- Login using Secure Remote Password (SRP) protocol

But also available from server-side SDKs (eg: Java, Python)

Part 1: Identity Pools

Features

Maintains a store of users with passwords and profile data

Signup may be user-initiated or system-initiated

Sends messages to users for email/SMS verification, password reset

Two-factor authentication via SMS

Clients can correct directly via AWS-supplied SDK, or mediated by app-server

Creating the Identity Pool

This isn’t supported by CloudFormation (yet)

Can add “profile” attributes at time of pool creation, not afterward

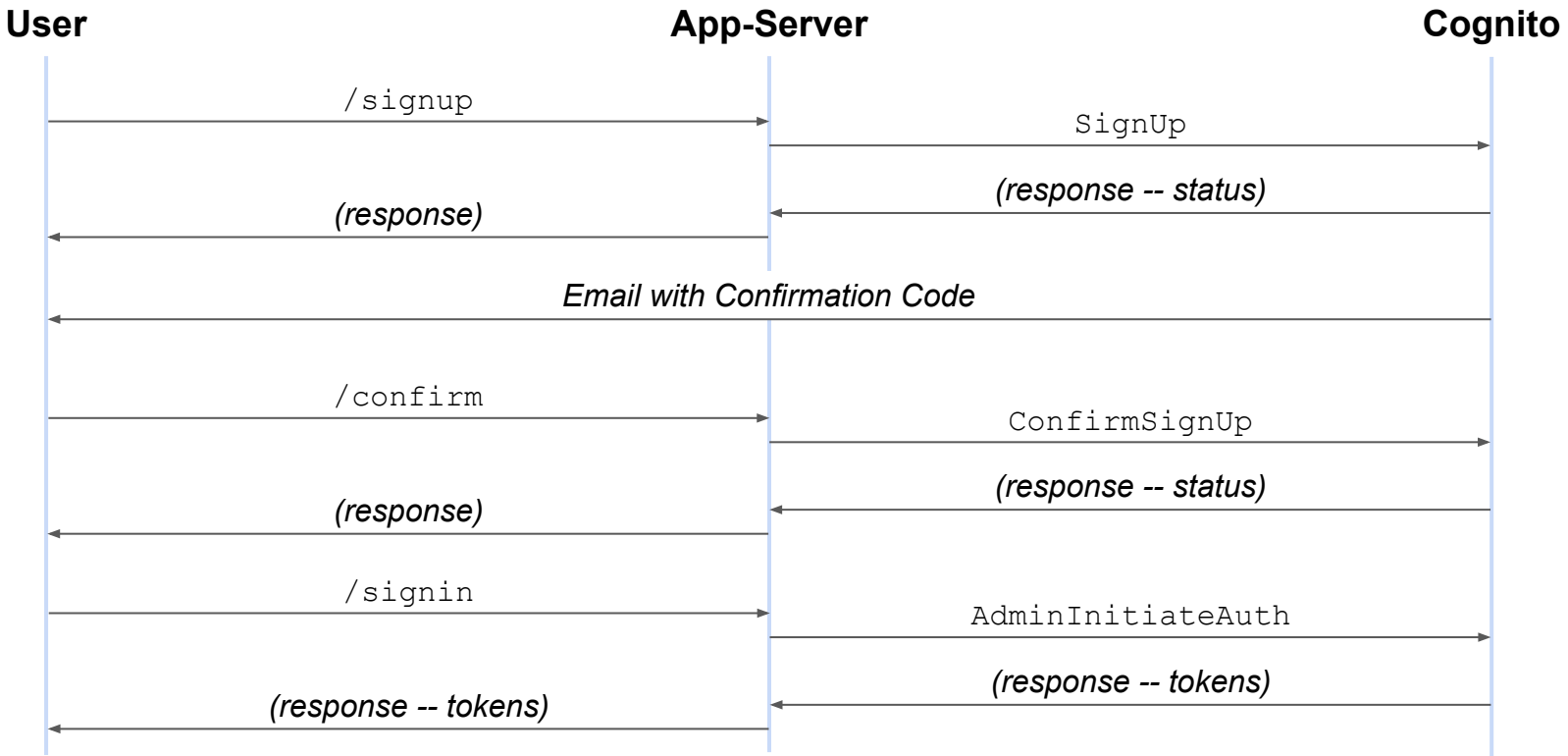
- Need to include email/SMS if you want verification
- Probably not worthwhile to mark these as aliases

Do you want to allow self-signup or force admin creation?

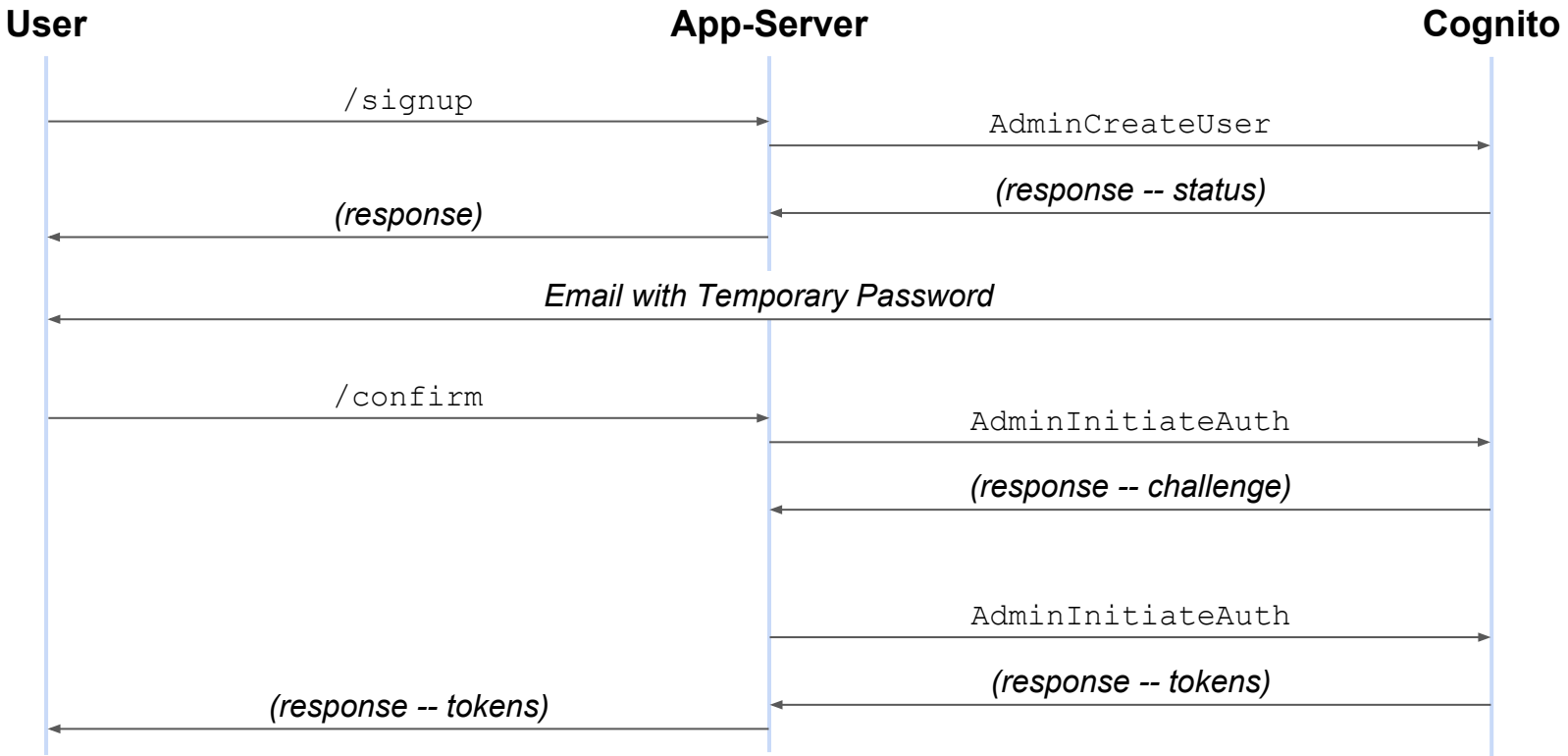
In addition to pool, you must create an “app” (aka client ID)

- *Do not create/require a client secret!*

User Flow: Self-signup



User Flow: Admin-created



User Tokens

Access Token

- JWT token using RS256 signature
- Expires after 1 hour (no ability to increase/decrease)
- Can be used to retrieve user profile data

ID Token

- Similar to Access token but includes user profile data

Refresh Token

- Opaque string used to retrieve new Access tokens
- Default expiration is 30 days, can be configured when creating pool

Sample Access Token


```
{
    "kid":"1D18kVZ5Z\/C1NjxSzwLmfDaO5BSru\/fsxvpCBvz1hnw=",
    "alg":"RS256"
}

{
    "sub":"fb3169fb-45d0-459b-9255-ca3d86f74f00",
    "token_use":"access",
    "scope":"aws.cognito.signin.user.admin",
    "iss":"https:\/\/cognito-idp.us-east-1.amazonaws.com\/us-east-1_myouCfMMh",
    "exp":1492382746,
    "iat":1492379146,
    "jti":"f89c3999-8b33-4904-8e36-2fb9f250db70",
    "client_id":"63bqoilc7topfsqblqgfcgse4o",
    "username":"test1354@mailinator.com"
}
```

JWT Keys

https://cognito-idp.{region}.amazonaws.com/{userPoolId}/.well-known/jwks.json

```
"keys": [{
    "alg": "RS256",
    "e": "AQAB",
    "kid": "1D18kVZ5Z/C1NjxSzwLmfDaO5BSru/fsxvpCBvz1hnw=",
    "kty": "RSA",
    "n": "n6YKOfsqcsWF6Z…",
    "use": "sig"
}, … ]
```

Caveats and Quirks

Temporary password email does *not* confirm email address

Two-factor authentication uses SMS

(Java SDK) Sometimes response is a value, sometimes an exception

Beware rate limiting

Part 2: Federated Identities

Features

Associates external authentication provider with Amazon ID

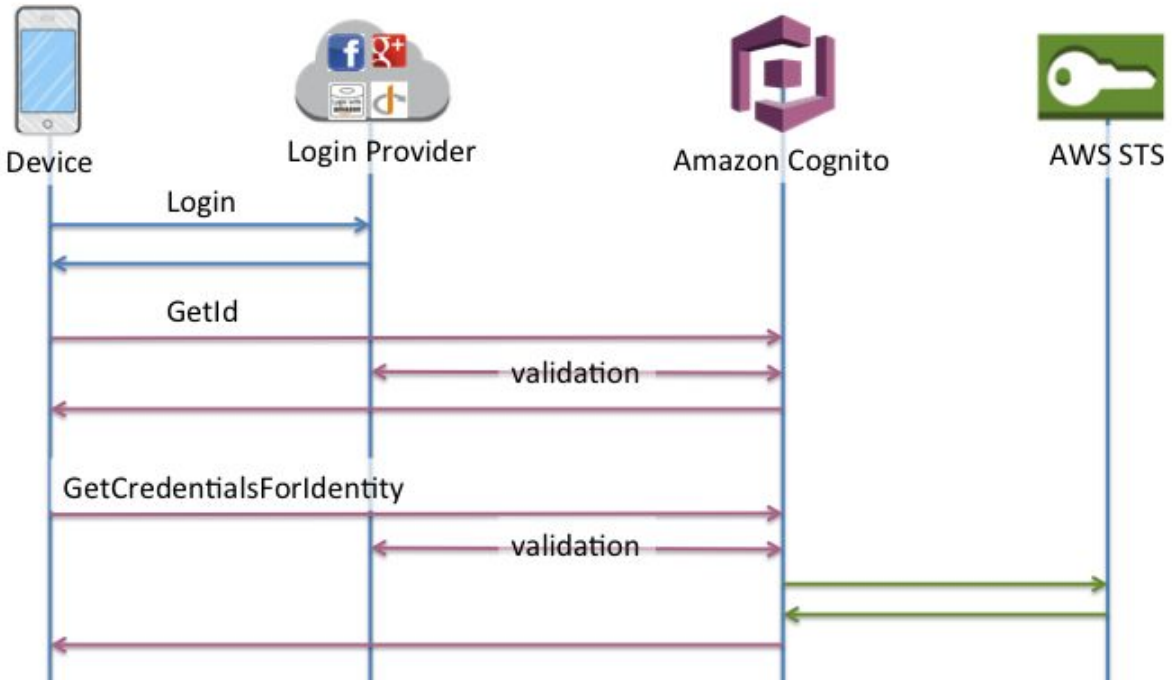
- Well-known services: Amazon, Facebook, Google, Twitter
- Cognito IDP
- Anyone supporting OpenID Connect or SAML

Identity token is used to retrieve limited-time access token from STS

- This is the basis for access to Cognito Sync, but could be used for any other AWS service

Allows unauthenticated access to a limited set of services

Authentication Flow



Source: http://docs.aws.amazon.com/cognito/latest/developerguide/authentication-flow.html

Setting up Roles for Federated Identities

Two default roles: one for authenticated users, one for unauthenticated

Must define trust relationship for `cognito-identity.amazonaws.com`

- Restrict role to specific identity pools via `cognito-identity.amazonaws.com:aud`
- Restrict role to authenticated/unauthenticated via `cognito-identity.amazonaws.com:amr`

Authenticated users may assume an explicit role (if permitted)

Caveats and Quirks

When using 3rd-party provider, you don’t control who gets authorized

- Unless you want to add a lot of fine-grained conditions to your trust relationships

With Cognito IDP identity provider must use ID token, *not* Access token

For More Information

Post describing how to use Cognito IDP in app-server

http://blog.kdgregory.com/2016/12/server-side-authentication-with-amazon.html

Serverless application using Cognito for authentication (ongoing project)

https://github.com/kdgregory/example-lambda-java

My web site

http://kdgregory.com/index.php?page=programming