**How does AWS IAM role, STS and Identity Pool work with each other.**

JUN 23, 2021 BY SHUO YANG

*This is a guest blog post from Shuo Yang in his blog series*[*“Transitioning to Programming the Cloud”*](https://syang.substack.com/)*, as a part of our blog posts focusing on Identity, Security and Access.*

We talked about IAM in the past 3 posts, [identities in IAM](https://syang.substack.com/p/aws-iam-in-a-laymans-words-1-identities), [manage users privilege as an IT person](https://syang.substack.com/p/iam-in-laymans-terms-2-managing-internal) and [control privilege boundaries](https://syang.substack.com/p/aws-iam-in-a-laymans-terms-3-iam):

* These are IAM topics that are probably trivial to security professionals, but we hope we gave enough ideas to a small startup which does not have full time security professionals yet.

We also talked about how applications use AWS Cognito Identity Pool to get AWS temporary credentials to access AWS resources in early posts of “[What I wish I could have learned before starting using AWS Cognito](https://syang.substack.com/p/what-i-wish-i-could-have-learned)” and “[Authentication and authorization with AWS Amplify under the hood](https://syang.substack.com/p/authentication-and-authorization)”:

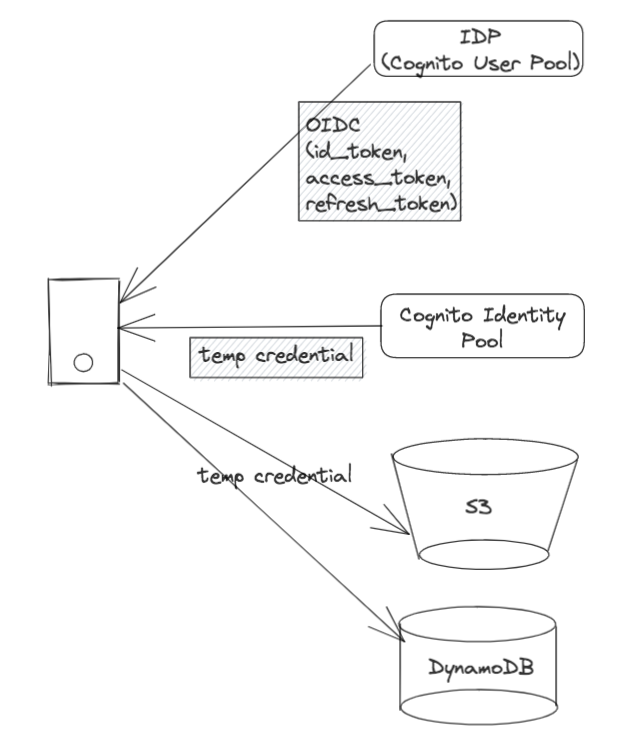
* Application engineers likely have already used the capability provided by Cognito Identity Pool (CIP), but we did not touch deeply enough on the relationship between CIP, STS and IAM roles.

In this post, we want to give a holistic view across these three AWS building blocks: CIP, STS and IAM roles.

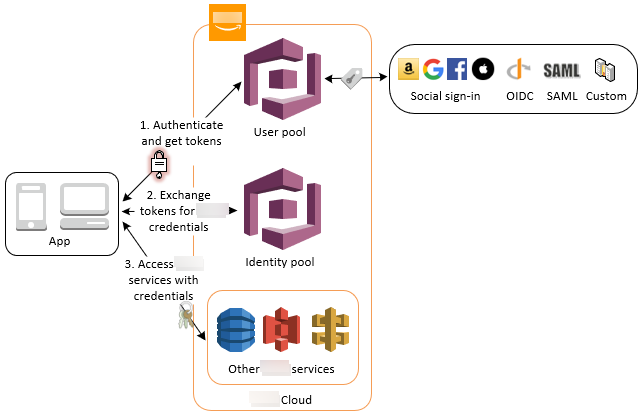
**How does an app access AWS resources by getting credentials from Cognito Identity Pool?**

When an application uses Cognito as its AuthN and AuthZ infrastructure, see the post of “[What I wish I could have learned before starting using AWS Cognito](https://syang.substack.com/p/what-i-wish-i-could-have-learned)”, from 10,000 feet high,

* The application gets an OIDC token from a Cognito User Pool (other IDP authentication mechanisms can also work at this step)
* Then applications get AWS temporary credentials from Cognito Identity Pool by providing the IDP-issued tokens (retrieved from the previous step);
* Then the application can access the AWS resources with the AWS temporary credentials.

*High level procedure on how an app accesses AWS resources using credentials issued from Cognito Identity Pool*

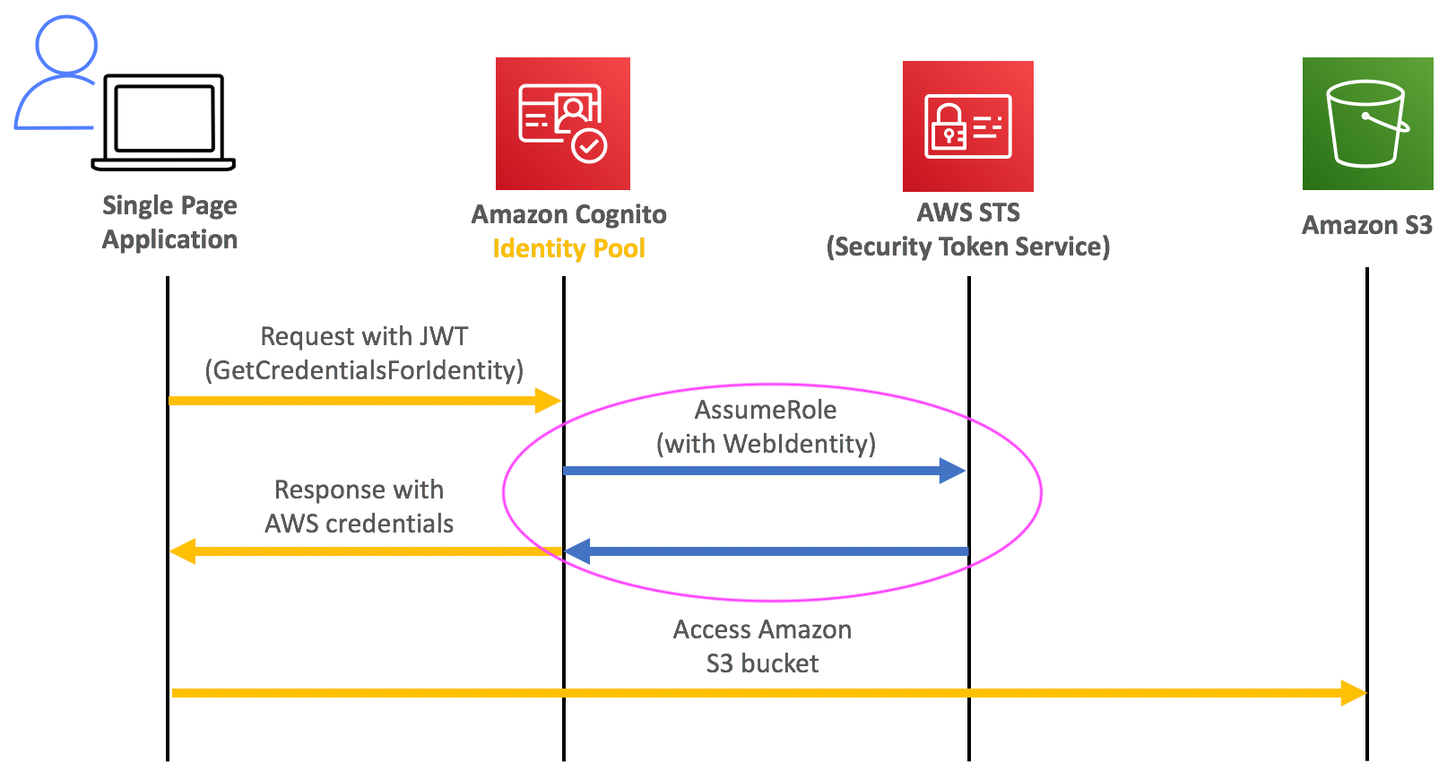
This three-step procedure diagram is what we typically see on the Internet when people talk about application development using Cognito as an auth mechanism (see the next diagram)

*This is what you typically see on the Internet about how Identity Pool issues AWS temporary credentials to the applications*

Accessing Amazon Services Using an Identity Pool After Sign-in - Amazon Cognito This is what you typically see on the Internet about how Identity Pool issues AWS temporary credentials to the applications

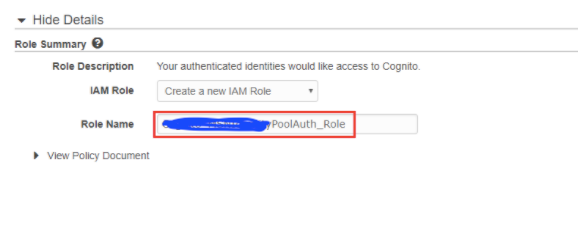
**How does Identity Pool work with AWS STS?**

Let’s zoom in the 2nd step in the previous procedure diagram a bit more. How does Identity Pool issue the AWS temporary credentials? Under the hood, it calls AssumeRole with the web identity (previously retrieved tokens from IdP) to AWS STS to get the AWS temporary credentials.

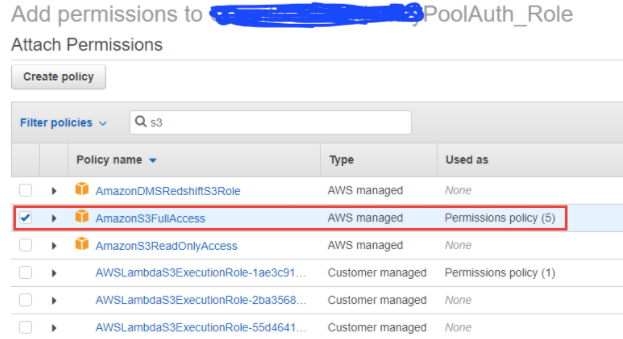
*Under the hood, Identity Pool get credentials from STS*

**Under the hood, Identity Pool get credentials from STS**

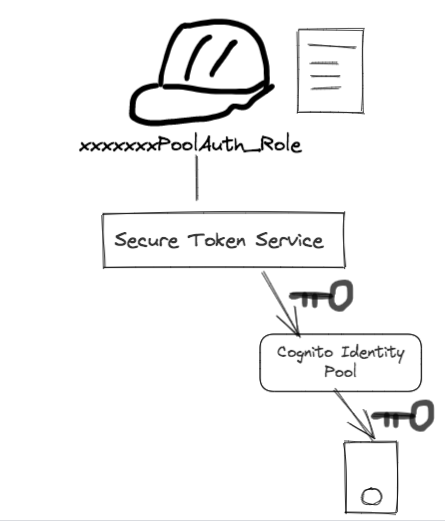
How does AWS STS issue temporary credentials with the right IAM role? During our setup of Identity Pool, we have set up the IAM roles for the authenticated users when we configure the Cognito Identity Pool:



And then we set up the right IAM policy for that role:



The above step exactly sets up the IAM roles, which we talked about in [our previous post](https://syang.substack.com/p/aws-iam-in-a-laymans-words-1-identities) about IAM identities: STS issues the temporary AWS credentials representing the role we defined in the IAM service.



**Teleport cybersecurity blog posts and tech news**

Every other week we'll send a newsletter with the latest cybersecurity news and Teleport updates.

Top of Form

SUBSCRIBE

Bottom of Form

**Summary:**

This post gives a holistic view of the issuing chain of an application’s temporary credentials:

* When an application needs to get AWS temporary credentials from Cognito Identity Pool
* Cognito Identity Pool communicates with AWS STS
* STS issues the temporary credentials representing the right role we set up in AWS Identity Role

I feel this post may be the first article talking all these three together, and we hope we give a simple yet clear description of the under-the-hood mechanism. Please share this if you think it helps people who do full stack development on AWS.