

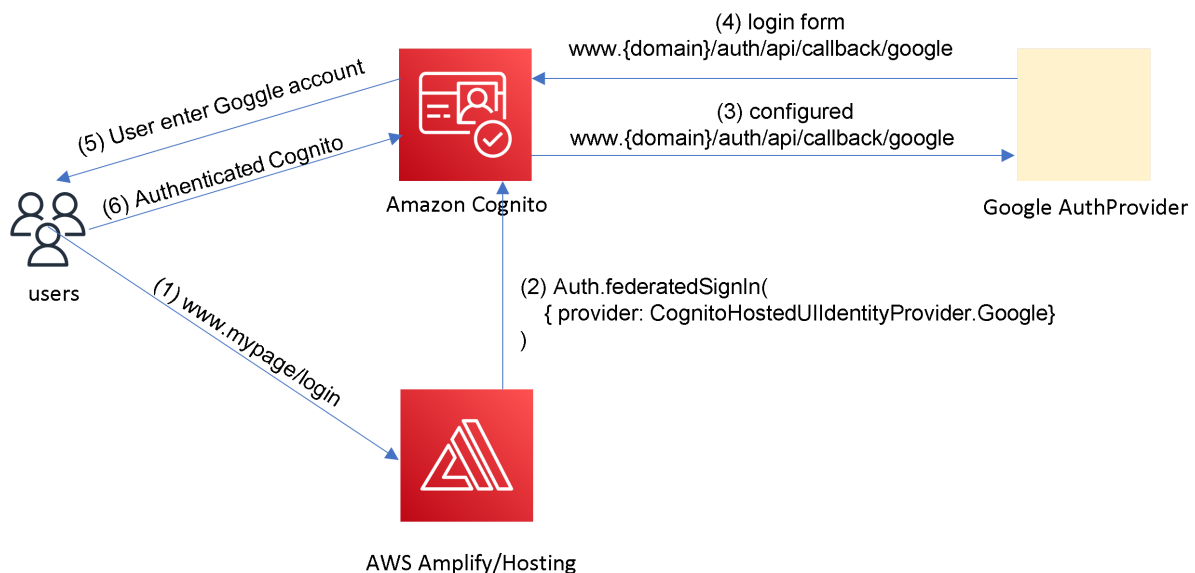
# AWS NextJS (SSR) Authentication

25 DEC 2021 Hai Tran

## Discussion

Just take note three options to do Authentication with AWS Amplify/Cognito.

1. **Amplify Auth** I first built a website using ReactJS and Amplify. Amplify provides nice Auth API, and custom UI and other Amplify UI components. Many auth providers integrated.
2. **NextAuth** I move to NextJS and **Amplify SSR hosting** hosting for quick serving static contents.
3. **Cognito Federated** have to use a bit low level setup with Cognito because I was not able to setup Amplify SSR Authentication.



## Option 1. Amplify Auth

Setup with Amplify CLI, just go through default setup and push. The many Auth features are provided via this **AuthClass**

Just copy from [here](https://docs.amplify.aws/lib/auth/getting-started/q/platform/js/#option-1-use-pre-built-ui-components) to feel how to use it.

### Basic

```
import { Amplify } from 'aws-amplify';

import { withAuthenticator } from '@aws-amplify/ui-react';
import '@aws-amplify/ui-react/styles.css';

import awsExports from './aws-exports';
```

```
Amplify.configure(awsExports);

function App({ signOut, user }) {
  return (
    <>
    <h1>Hello {user.username}</h1>
    <button onClick={signOut}>Sign out</button>
    </>
  );
}

export default withAuthenticator(App);
```

## Option 2. Next Auth

Create and setup the [...nextauth].ts file

```
/pages/api/auth/[...nextauth].ts
```

```
import NextAuth from 'next-auth';
import GoogleProvider from 'next-auth/providers/google';
import GitHubProvider from 'next-auth/providers/github';
import { NextApiRequest, NextApiResponse } from 'next';

const options = {
  providers: [
    GitHubProvider({
      clientId: process.env.GITHUB_CLIENT_ID,
      clientSecret: process.env.GITHUB_CLIENT_SECRET
    }),
    GoogleProvider({
      clientId: process.env.GOOGLE_CLIENT_ID,
      clientSecret: process.env.GOOGLE_CLIENT_SECRET
    })
  ]
}

export default async function auth(req: NextApiRequest, res:
NextApiResponse) {
  return await NextAuth(req, res, options)
}
```

Then use it via SessionProvider, here is \_\_app.js

```
import { SessionProvider } from "next-auth/react"

function Website({ Component, pageProps }) {
```

```
    return (
      <SessionProvider session={pageProps.session}>
        <Component {...pageProps}/>
      </SessionProvider>
    )
  }
}
```

And useSession in components such as index.js. When you click Sign In, you will be redirected to the provider auth flow such as Google, Github login form. After enter your accounts, a call back url will bring you back to your web.

```
import { useSession, signIn, signOut } from "next-auth/react"

export default function Component() {
  const { data: session } = useSession()
  if (session) {
    return (
      <>
        Signed in as {session.user.email} <br />
        <button onClick={() => signOut()}>Sign out</button>
      </>
    )
  }
  return (
    <>
      Not signed in <br />
      <button onClick={() => signIn()}>Sign in</button>
    </>
  )
}
```

Need to configure **.env.local**

```
COGNITO_CLIENT_ID=
COGNITO_CLIENT_SECRET=
COGNITO_ISSUER=
GOOGLE_CLIENT_ID=
GOOGLE_CLIENT_SECRET=
GITHUB_CLIENT_ID=
GITHUB_CLIENT_SECRET=
NEXTAUTH_URL=https://example.com
```

Need to configure at the auth provider side for example Google.  
Authorized JavaScript origins

```
URIs: https://example.com
```

Authorized redirect URIs for local test

```
http://localhost:3000/api/auth/callback/google
```

Authorized redirect URIs for deployment

```
https://{cognito-domain}/oauth2/idpresponse
```

## Option 3. Cognito Federated

amplify cli to configure with social auth provider

```
amplify add auth
```

Amplify CLI will ask you to enter those. AWS cognito console and your code aws-export.js should be exactly same, including the trailing slash.

```
redirectSignIn: http://localhost:3000/  
redirectSignOut: http://localhost:3000/signout/  
clientID: GOOGLE_CLIENT_ID  
secretID: GOOGLE_SECRET_ID
```

For deployment

```
redirectSignIn: http://{amplify-domain}/  
redirectSignOut: http://{amplify-domain}/signout/  
clientID: GOOGLE_CLIENT_ID  
secretID: GOOGLE_SECRET_ID
```

Need to configure call back URL at the provider side such as Google Authorized redirect URIs for local test

```
http://localhost:3000/api/auth/callback/google
```

Authorized redirect URIs for deployment

```
https://{cognito-domain}/oauth2/idpresponse
```

**Note** Option 1 and 3 works well both local and deploy. Option 2 with the web hosted by AWS Amplify SSR will trouble you. Hope AWS Amplify will support authentication for SSR soon.