LESSON 2

In this lesson, we'll create a web site for our video hosting platform. We'll integrate the web site with auth0, as the means of authenticating users.

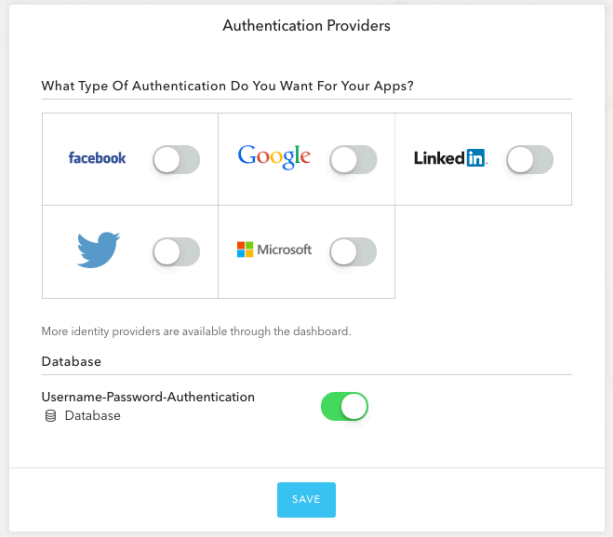
In true serverless style, this will be a static web site, meaning that it can be hosted on S3, or any CDN. It is comprised entirely of static HTML, JS and CSS and does not need to be served by a traditional web server.

NOTE: PLEASE CREATE ALL YOUR RESOURCES IN THE N. VIRGINIA REGION (US-EAST-1)

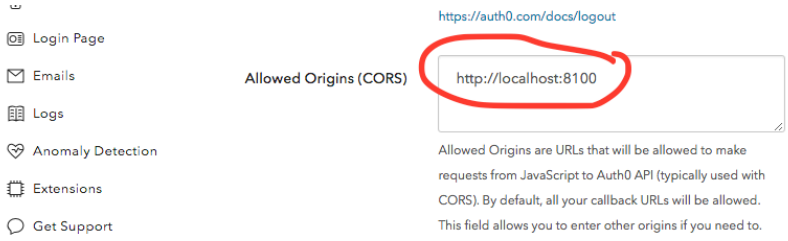
1. CREATE AUTH0 ACCOUNT

You'll need to create a free auth0 account. Visit http://auth0.com and follow the sign up steps.

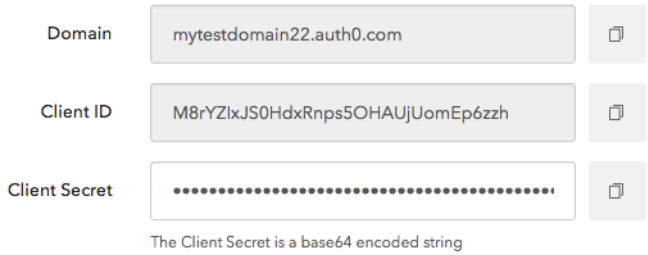
- You'll be asked to enter an **account name**. Enter a name that is unique to you:
  eg. janesmith-24hrvideo.auth0.com
- Enter **"US West"** as your **region**
- Create the account
- Next you'll be asked to select authentication providers. Turn them all off, and just make sure that you have **Username-Password-Authentication authentication enabled**. Press save.



- Auth0 will create a default application for you, and navigate you to the application.
- Choose the "Settings" tab.
- Scroll down until you find the textbox called **Allowed Origins (CORS)**.
  Enter the following value: **http://localhost:8100**



- Scroll down and hit the **Save Changes** button
- We now need to retrieve some values from Auth0 that will be needed throughout this workshop. Scroll up on the same Settings page, and find the **Domain**, **Client ID** and **Client Secret**. Copy these into your favourite text editor so you have them at the ready.



2. SETUP WEBSITE LOCALLY

This web site would normally be deployed via a CDN, but for the purposes of this workshop we're going to host it locally on your computer.

- Edit the following file in your favourite text editor:

  lab-2/website/js/config.js

  Enter your auth0 domain and client ID (you made a note of these earlier) & save the file.

- Open a terminal / command-prompt and navigate to the following folder:

  lab-2/website

- Run the following command, to bring down dependencies from npm:

  npm install

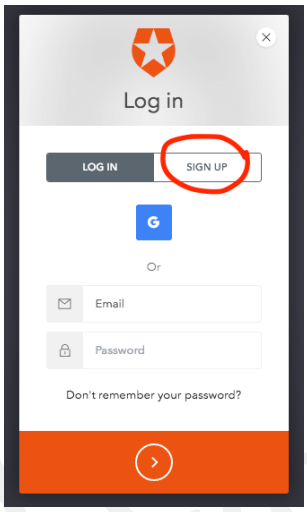
- Run the following command, to start a local web server at port 8100:

  npm start

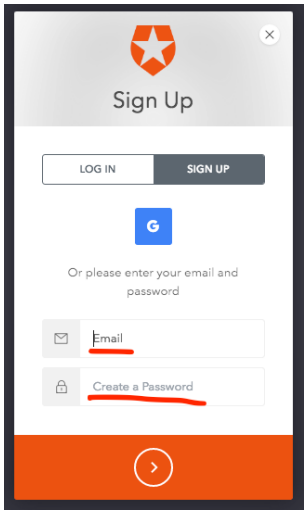
3. GIVE IT A SPIN!

- Open a web browser and navigate to:
  http://localhost:8100

  You should see the 24 hour video web site. There's not much here yet… that's OK! We're going to iteratively build on the site during the workshop.
- Notice the **Sign In** button in the top-right? Click on it to launch the authentication popup:
  This popup is rendered entirely by auth0 – a huge timesaver if you need authentication in your platform!
- You'll need to create an account, so click on the **Sign Up** tab:



- Enter an email address and password for your new account. This will be saved to your custom auth0 database of users. **Make it something memorable** because you'll need to sign in/out multiple times throughout this workshop



- Press the big orange button at the bottom to create your account. A popup will launch to complete the sign up.
  Did you receive an error? Make sure you Always Allow Popups for this site.

- You'll be automatically signed in after your account is created. Look in the top right-hand corner of the web-site. You should see your name & a profile image / avatar (auth0 will use gravatar.com to find an image for your email address), plus a sign out button.

Congratulations – you now have a serverless web site with full user-sign up and authentication capabilities!

Get Your Hands Dirty

- Now use Auth0 to hookup a 3rd party social provider, such as Facebook or Twitter. Note: You will need to create an app with each provider that you hookup. Auth0's web site contains instructions.
- Auth0 supports running node.js rules on each user login. Add a rule to:
    - Force email verification (there is a pre-built Auth0 rule for this)
    - Only allow users from a specific white-list
- Auth0 supports the creation of delegation tokens to grant user's direct access inside your AWS account via IAM. It's worth understanding that this is possible and how it works. Read through the Auth0 documentation on this approach, and if you are game setup your web site to get an AWS delegation token.
  https://auth0.com/docs/integrations/aws