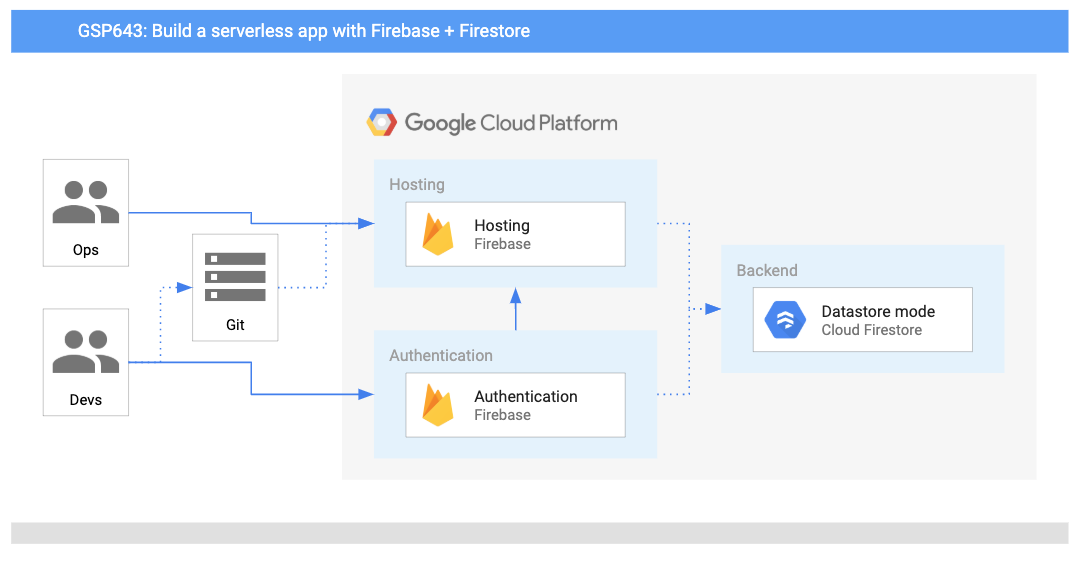
**Build a Serverless Web App with Firebase**

Overview

Twelve years ago, Lily started the Pet Theory chain of veterinary clinics. The Pet Theory chain has expanded rapidly over the last few years. Their old appointment system is not able to handle the increased load or allow clients to schedule their own appointments, so Lily asked Patrick, in IT, and Ruby, a consultant, to build a cloud-based system that easily scale. In this lab you will build a fully fledged Firebase web app that allows users to log information and schedule appointments in real time.

Architecture

This diagram gives you an overview of the services you will be using and how they connect to one another:



**Objectives**

In this lab, you will learn how to:

* Configure Firestore Security to automate server-side authentication and authorization.
* Add Google sign-in to your web app.
* Configure your database so users can add their contact information.
* Explore and deploy code that allows users to schedule appointments.
* Explore Firebase's real time updates in your web app.

**Firebase Backend**

Ruby sends Patrick an email:

|  |  |
| --- | --- |
| Ruby  *Ruby, Software Consultant* | Hi Patrick,  Awesome work last week. Great to see that the clinic's data has been migrated to Firestore!  It looks like the next task is to use Firebase to host the Pet Theory website.  Ruby |
| Patrick  *Patrick, IT Administrator* | Hi Ruby,  I haven't heard of Firebase hosting before, what are the benefits? Where would I get started?  Patrick |
| Ruby  *Ruby, Software Consultant* | Hi Patrick  The primary benefit of Firebase hosting is that it is serverless, so there is no infrastructure to manage. Security rules are also embedded within the application, so permissions can be restricted to minimize issues when handling customer data.  It also has a "pay as you use" model, which means Firebase is a comprehensive mobile development platform for our use case.  Ruby |
| Patrick  *Patrick, IT Administrator* | Hi Ruby  Sounds like Firebase will make security and infrastructure management (a big part of my job) a whole lot easier. I'm excited to not be billed for idle servers either!  Patrick |

Ruby sends Patrick some background information in an email, and they hold a meeting to work out the key activities. From this meeting they determine that he needs to:

* Configure a Firebase project.
* Establish security policies.
* Add the Firestore CLI to the Google Cloud project.

Next, help Patrick accomplish these tasks.

**Firebase Localhost**

You have assisted Patrick in setting up a working Firebase hosting environment where a web developer can deploy their code.

However, Patrick has never enabled Firebase authentication nor has he deployed code to Firebase, so he emails Ruby for some help...

|  |
| --- |
| Hey Ruby,  Thanks for all the tips! The Firebase environment looks like it is all set up. My next task is to deploy the website developers' code.  Can you help me understand what this entails and what I need to do next?  Patrick |
| Hi Patrick,  That's great to hear! I'll send you the instructions on how to run the application and add the following features:   * Set up web authentication for logging in. * Enable customer details to be logged on the Profile page. * Create a self service portal for appointments.   Ruby |
| Hi Ruby  That sounds like it will be quite a bit of work.  Does that mean I'll have to make structural changes every time I want to add something new? Not to mention the time it will take to see those updates...  Patrick |
| Hey Patrick  You can do most of the heavy lifting with Firebase libraries.  Firebase has some great command line tooling to help you connect your localhost to the backend Firebase project.  Use firebase init to tell Firebase which products you would like to use.  Once your project is set up you can simply call firebase deploy from the command line.  Ruby |
| Hi Ruby  Wow, that's very comforting! Firebase hosting just gets better and better :-)  Patrick |