**Build a Resilient, Asynchronous System with Cloud Run and Pub/Sub**

**Overview**

For the labs in the [Serverless Cloud Run Development](https://www.cloudskillsboost.google/course_templates/741) course, you will read through a fictitious business scenario and assist the characters with their serverless migration plan.

Twelve years ago, Lily started the Pet Theory chain of veterinary clinics. Over the years, the number of clinics has grown, and so has the need for automation. The way Pet Theory handles the results of medical tests when they come back from the lab is too slow and error-prone, and Lily wants to improve this.

Currently, Patrick, Pet Theory's IT administrator, handles test results manually. Whenever a test result comes back, he composes and sends an email to the client whose pet was tested, then he taps out a text message on his phone and sends the results as a text to the client.

Patrick is working with Ruby, a software consultant, to design a more scalable system. They want to build a solution that doesn't require a lot of ongoing maintenance. Patrick and Ruby have decided to go with serverless technology.

**Objectives**

In this lab, you will learn how to:

* Create a Pub/Sub topic and subscription
* Create a Cloud Run service that receives HTTP requests and publishes messages to Cloud Pub/Sub
* Create a Cloud Run service that receives messages from Cloud Pub/Sub
* Create a Pub/Sub subscription that triggers a Cloud Run service
* Test the resiliency of a system

Prerequisites

This lab assumes familiarity with the Cloud Console and shell environments. This lab is part of a series. Taking the previous labs could be helpful, but is not necessary:

* [Importing Data to a Firestore Database](https://google.qwiklabs.com/catalog_lab/2163)
* [Build a Serverless Web App with Firebase](https://google.qwiklabs.com/catalog_lab/2166)
* [Build a Serverless App with Cloud Run that Creates PDF Files](https://google.qwiklabs.com/catalog_lab/2161)

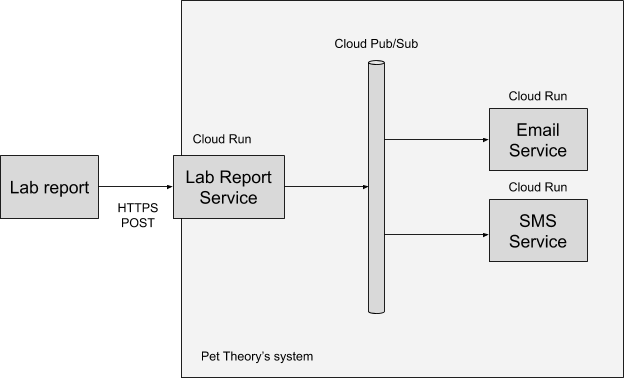
Scenario

Pet Theory would like to automate the process of sharing client test results. They have experienced a tough time keeping up with an increased volume of appointments, so Lily decides to ask Ruby for some assistance...

|  |
| --- |
| Hi Ruby,  Thanks for sorting out the insurance portal.  I was wondering if something could be done about the medical test results? We need a more efficient way of sending results to our clients.  Lily |
| Hey Lily,  Sure - let me see what I can do. I have a few ideas that may improve the situation.  Ruby |

Pet Theory uses an external company for medical tests. Once the lab company completes a medical test, they send the results back to Pet Theory.

The lab company uses a HTTP(s) POST to Pet Theory's web endpoint for medical lab results. The illustration below outlines the general architecture.



After looking at the general process followed, Ruby believes that a system can be designed in which Pet Theory is able to:

1. Receive the HTTP POST request and confirm receipt to the medical lab.
2. Email the test result to the client.
3. Send a text message (SMS) and an email to the client with the test result.

Ruby's design isolates each of the above activities and requires:

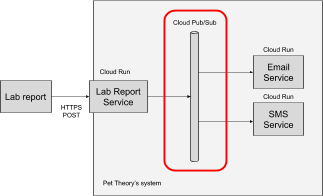
* A service to perform the request and response for the medical result(s)
* A service to email test results to the client
* A service to send a text message (SMS) to the client
* Pub/Sub to be used for inter-service communication
* Serverless infrastructure to be used for the application architecture

Through the use of single use functions, Ruby is looking to develop code that is easier to write and contains fewer bugs.

|  |
| --- |
| Hi Patrick,  Lily would like me to build a prototype to help with the processing of medical records.  To get started, could you set up a Pub/Sub Topic called new-lab-report.  Ruby |
| Hey Ruby,  That sounds like a cool project. I can get that finished for you this morning, both activities are really quick to set up on Google Cloud.  Patrick |

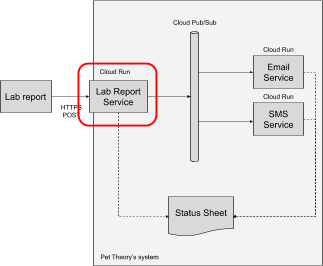
**Create a Pub/Sub topic**

Help Patrick to create a Pub/Sub topic called new-lab-report.



When a service publishes a Pub/Sub message, that message must be tagged with a topic. The Lab Report is consumed via the service to be created and publish a message for each report found.

Help Ruby to set up the new Lab Report Service.



This service will serve the purpose of prototyping, so it will only do two things:

1. Receive the lab report HTTPS POST containing the report data.
2. Publish a message on Pub/Sub.