Data Processing with Python

A containerized, scheduled, and monitored pipeline on the Google Cloud Platform

Meetup aggregate(PyLadies, PyAmsterdam, ETL).byebye(2020)
December 22, 2020
Nancy Irisarri Mendez

The use case

Business

A data processor, who manages the life cycle of sensitive (patient) data, and an organization of medical professionals, who have knowledge of the domain.

Context

Created a dashboard with functionality for filtering and drilling down at patient level, as well as benchmarking on a national level.

- Codman Explorative Dashboard
  https://dica.nl/dica/codmandashboard

Problem statement

How to take full advantage of the domain knowledge while also making sure that the dashboard data is constantly updated?

Challenges deep-dive

Challenge 1

Domain knowledge

Those with domain knowledge in the organization of medical professionals are R programmers.

Challenge 2

Sensitive patient data

The data processor and its employees are responsible and have permissions for storing and processing sensitive data.

Challenge 3

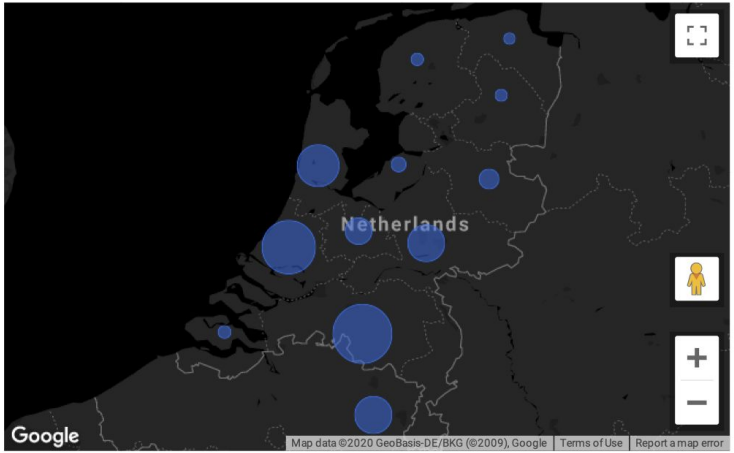
Weekly data updates

New patient data is added weekly and indicator calculations can also change weekly.

Solution

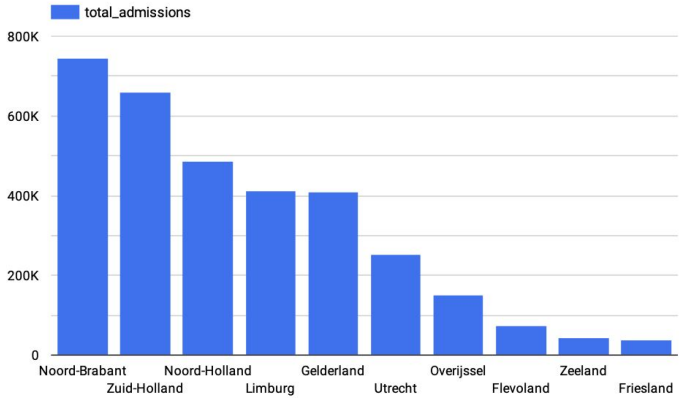
Upload Ship

A containerized, scheduled, and monitored pipeline on the Google Cloud Platform.



total_admissions 23,275 744,640

	province	total_admissions
1.	Noord-Brabant	744,640
2.	Zuid-Holland	658,978
3.	Noord-Holland	485,948
4.	Limburg	412,827
5.	Gelderland	408,422
6.	Utrecht	253,929
7.	Overijssel	150,657
8.	Flevoland	75,390
9.	Zeeland	43,074
10.	Friesland	37,697
11.	Drenthe	34,641
12.	Groningen	23,275



What can you build?

Data Studio


Implementation

Takeaways

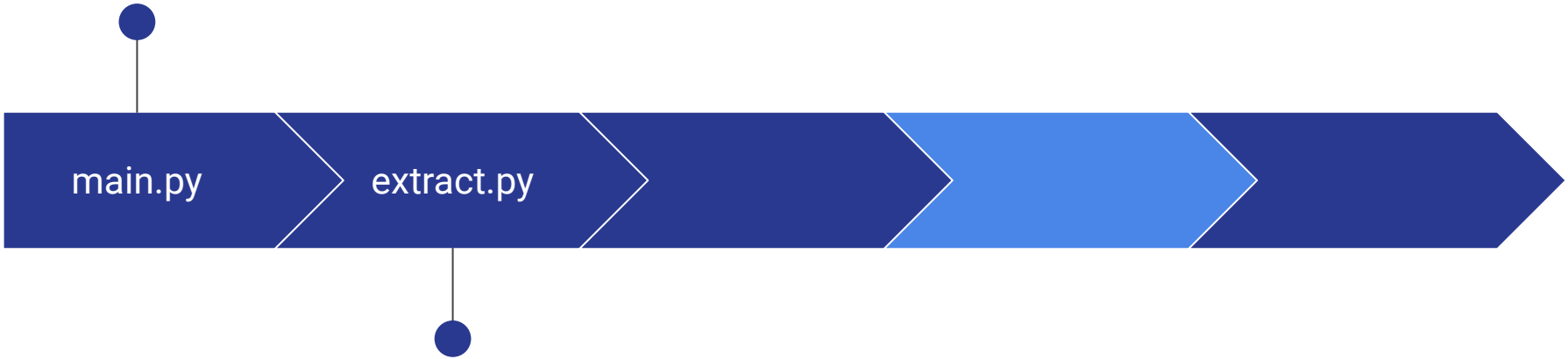
What I would
most like to share

- Code structure
- Exception handling
- Dockerfile
- Google Cloud Platform services

Code Overview

Takes care of
executing the pipeline

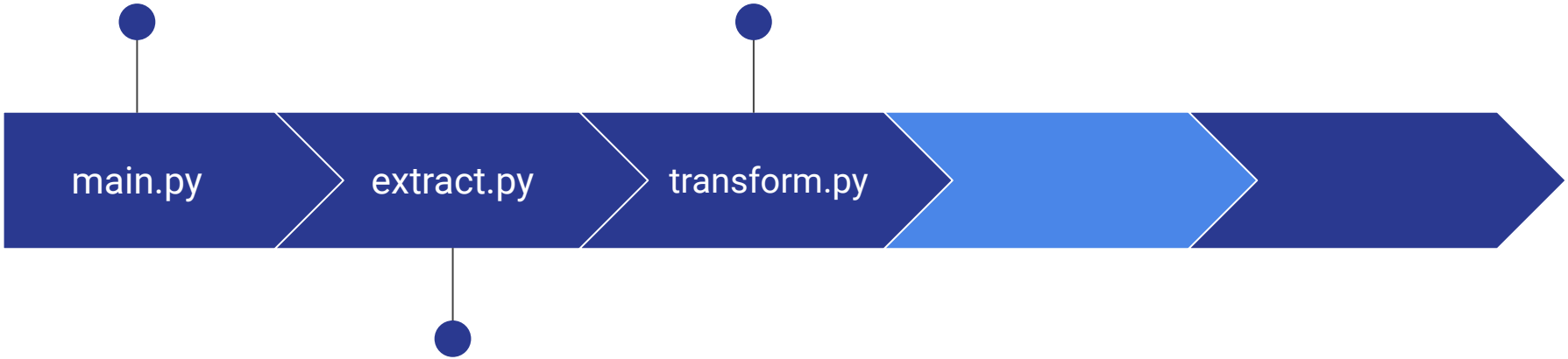
Takes care of
executing the pipeline



Downloads external
sources; connects to
databases

Takes care of executing the pipeline

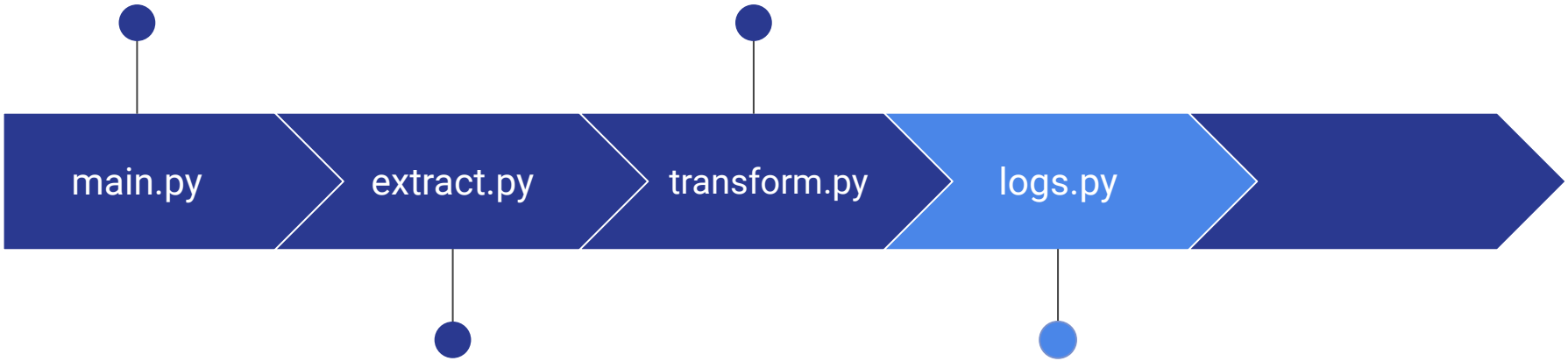
Produces output by running R scripts on the data



Downloads external sources; connects to databases

Takes care of executing the pipeline

Produces output by running R scripts on the data



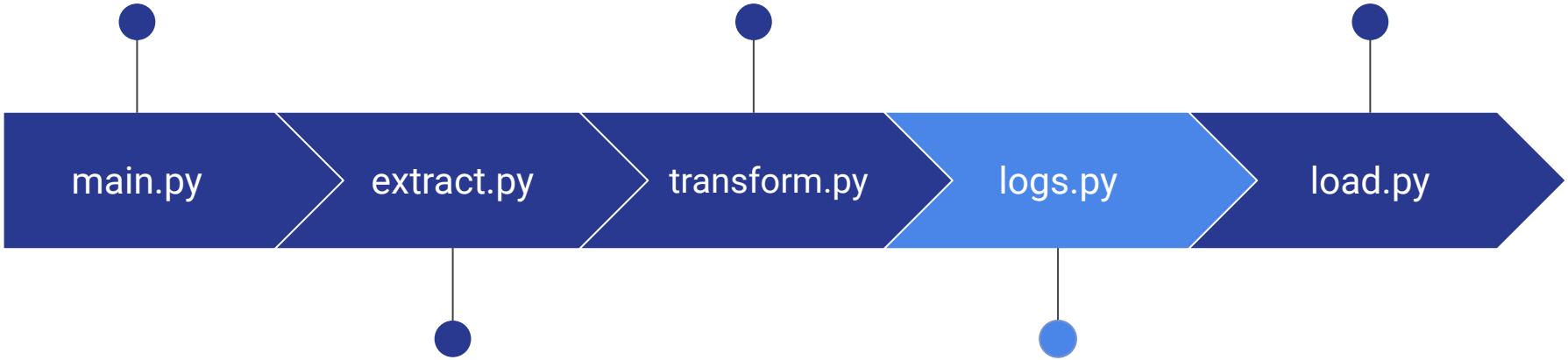
Downloads external sources; connects to databases

In case there is no output, the standard error messages are used to debug

Takes care of executing the pipeline

Produces output by running R scripts on the data

Uploads output as a table



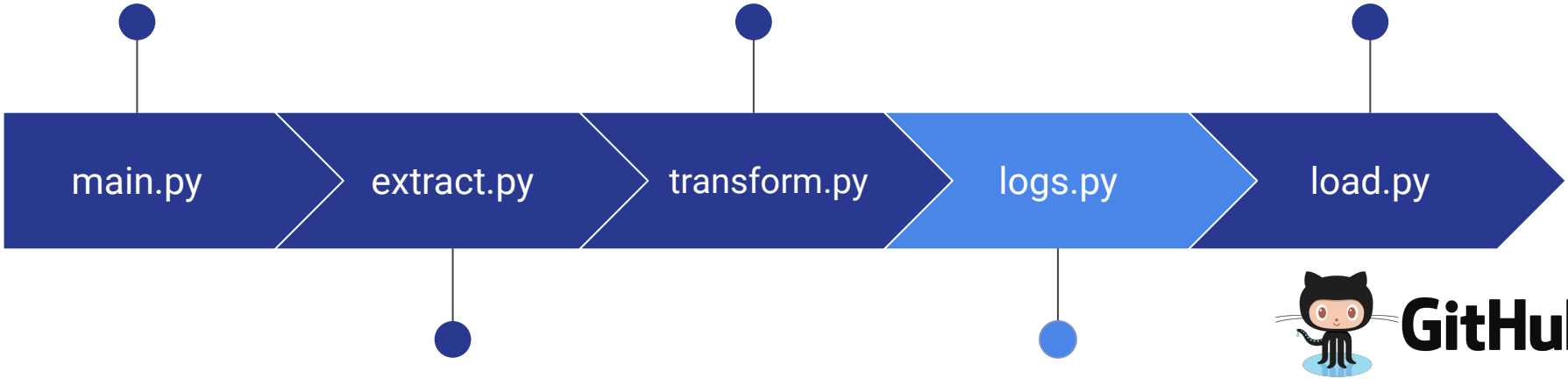
Downloads external sources; connects to databases

In case there is no output, the standard error messages are used to debug

Takes care of executing the pipeline

Produces output by running R scripts on the data

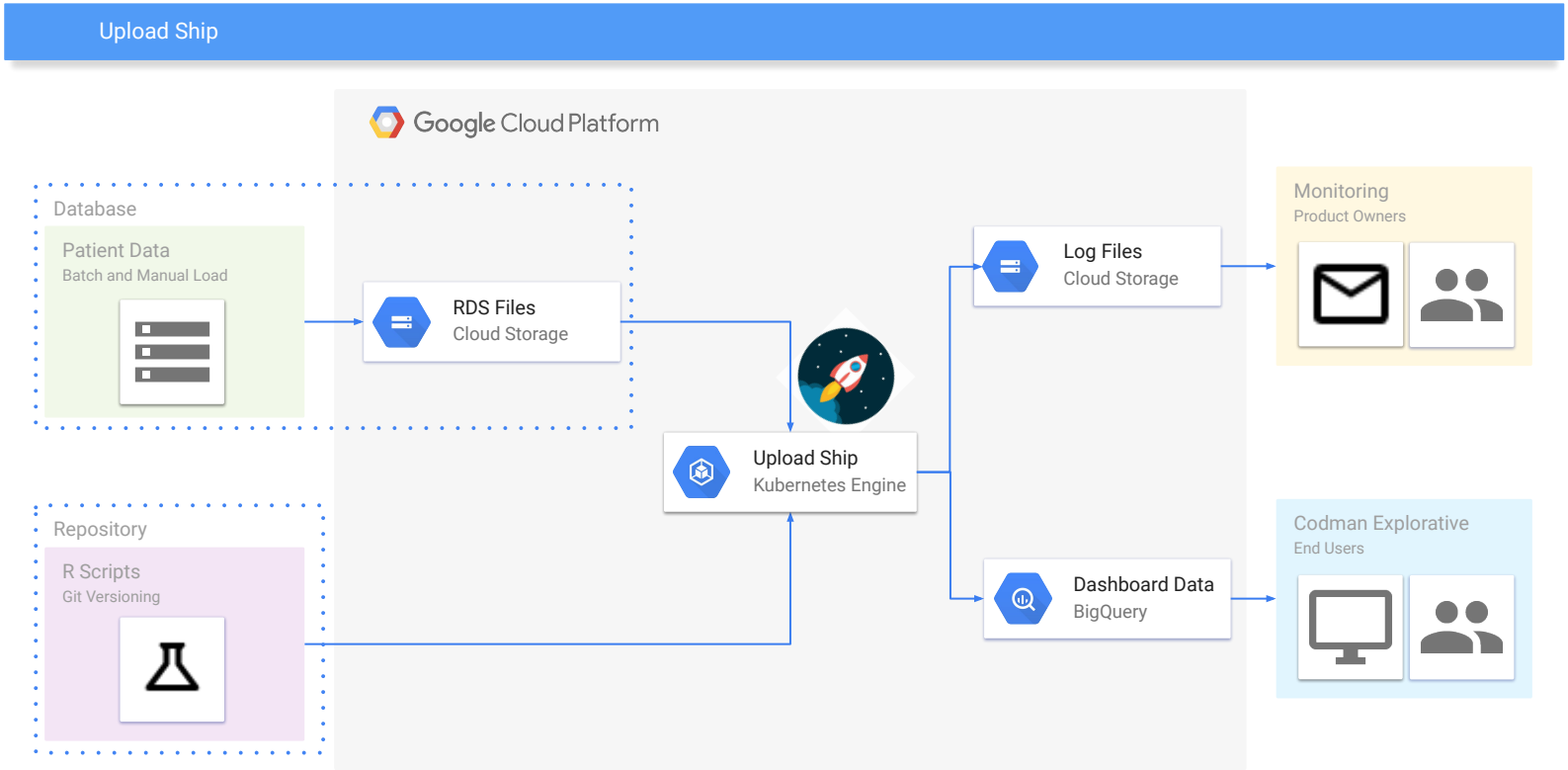
Uploads output as a table



Downloads external sources; connects to databases

In case there is no output, the standard error messages are used to debug

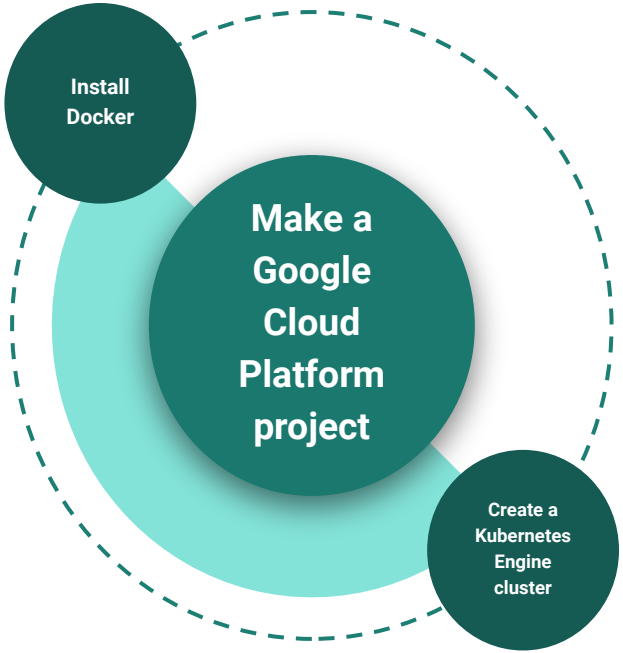
Architecture



README

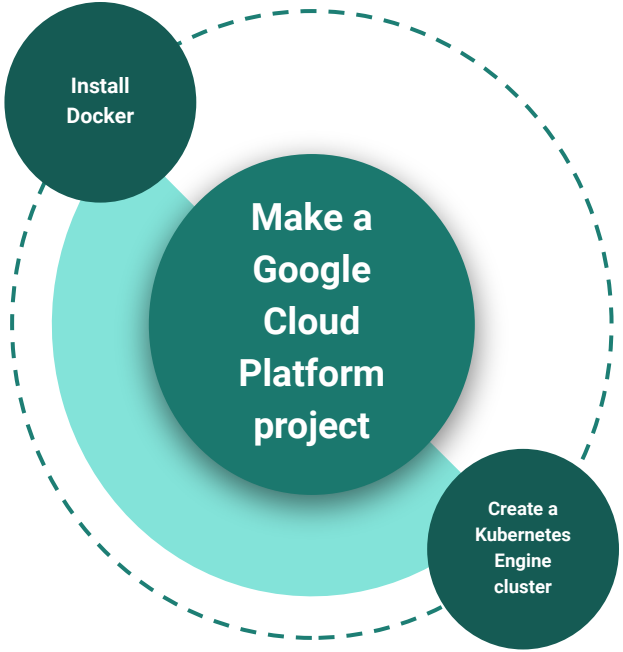
README

Setup

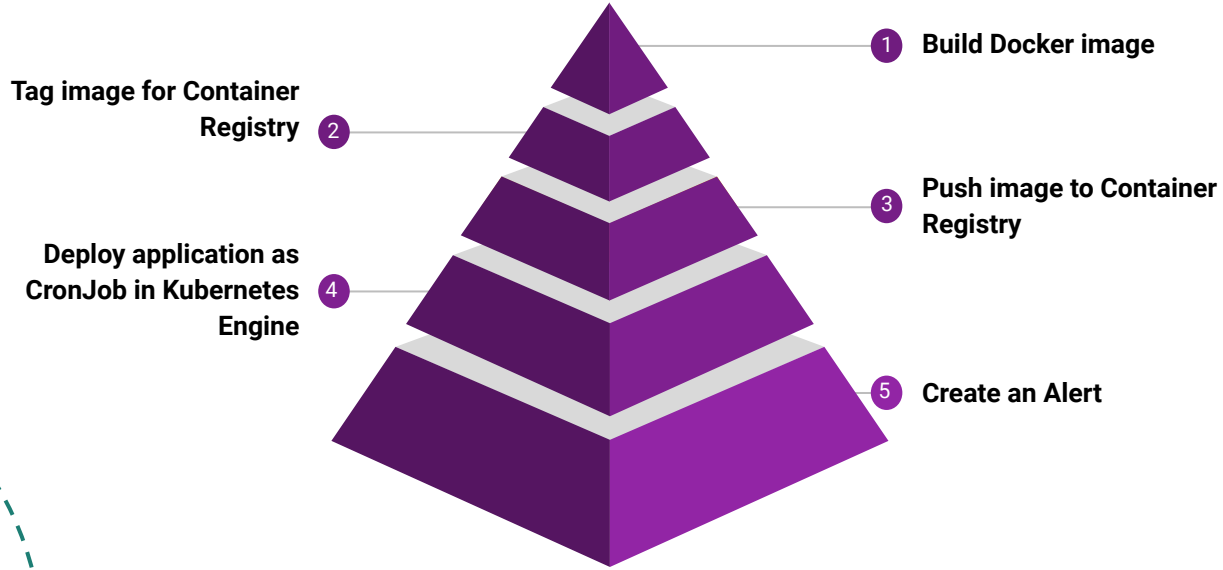


README

Setup



Running in Kubernetes Engine



Let’s run it!

Google Cloud Platform

Meetup aggregate(PyLadies, PyAmsterdam, ETL).byebye(2020)
December 22, 2020
Nancy Irisarri Mendez

Thank you for your attention!