Storing & Analyzing the Internet

Orchestructure, July 25, 2018

bit.ly/orchestructure-censys-2018-07-25

Andrew Sardone
andrew@censys.io

David Adrian
dadrian@censys.io

About Censys

- Discover the devices, networks, and infrastructure on the Internet and monitor how they change over time
- Created by the University of Michigan research team behind ZMap
  - Open source Internet-wide scanning and measurement tools

A Search Engine Backed by Internet-Wide Scanning

Zakir Durumeric[†] David Adrian[†] Ariana Mirian[†] Michael Bailey[‡] J. Alex Halderman[†]

[†] University of Michigan [‡] University of Illinois, Urbana Champaign

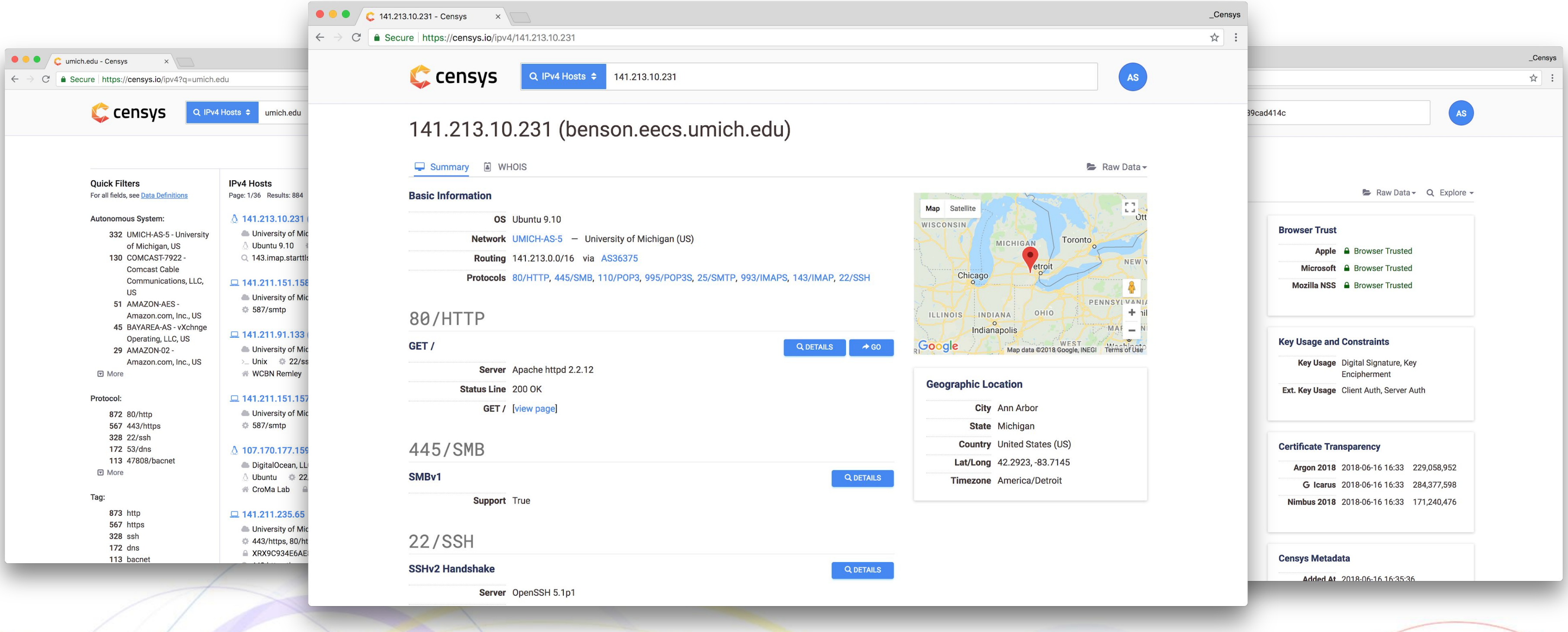
{zakir, davadria, amirian, jhalderm}@umich.edu mdbailey@illinois.edu

ABSTRACT

Fast Internet-wide scanning has opened new avenues for

would need to develop a high-performance application scanner to make HTTPS connections to hosts listening on port

About Censys



umich.edu - Censys

Secure https://censys.io/ipv4?q=umich.edu

censys IPv4 Hosts umich.edu

Quick Filters
For all fields, see Data Definitions

Autonomous System:

332	UMICH-AS-5 - University of Michigan, US
130	COMCAST-7922 - Comcast Cable Communications, LLC, US
51	AMAZON-AES - Amazon.com, Inc., US
45	BAYAREA-AS - vXchnge Operating, LLC, US
29	AMAZON-02 - Amazon.com, Inc., US

More

Protocol:

872	80/http
567	443/https
328	22/ssh
172	53/dns
113	47808/bacnet

More

Tag:

873	http
567	https
328	ssh
172	dns
113	bacnet

IPv4 Hosts
Page: 1/36 Results: 884

141.213.10.231
University of Mic
Ubuntu 9.10
143.imap.starttls

141.211.151.158
University of Mic
587/smtp

141.211.91.133
University of Mic
Unix 22/ss
WCBN Remley

141.211.151.157
University of Mic
587/smtp

107.170.177.159
DigitalOcean, LL
Ubuntu 22
CroMa Lab

141.211.235.65
University of Mic
443/https, 80/ht
XRX9C934E6AE

141.213.10.231 - Censys _Censys

Secure https://censys.io/ipv4/141.213.10.231

censys IPv4 Hosts 141.213.10.231 AS

141.213.10.231 (benson.eecs.umich.edu)

Summary WHOIS Raw Data

Basic Information

OS	Ubuntu 9.10
Network	UMICH-AS-5 — University of Michigan (US)
Routing	141.213.0.0/16 via AS36375
Protocols	80/HTTP, 445/SMB, 110/POP3, 995/POP3S, 25/SMTP, 993/IMAPS, 143/IMAP, 22/SSH

Geographic Location

City	Ann Arbor
State	Michigan
Country	United States (US)
Lat/Long	42.2923, -83.7145
Timezone	America/Detroit

80/HTTP

GET / DETAILS GO

Server	Apache httpd 2.2.12
Status Line	200 OK
GET /	[view page]

445/SMB

SMBv1 DETAILS

Support	True

22/SSH

SSHv2 Handshake DETAILS

Server	OpenSSH 5.1p1

_Censys

39cad414c AS

Raw Data Explore

Browser Trust

Apple	Browser Trusted
Microsoft	Browser Trusted
Mozilla NSS	Browser Trusted

Key Usage and Constraints

Key Usage	Digital Signature, Key Encipherment
Ext. Key Usage	Client Auth, Server Auth

Certificate Transparency

Argon 2018	2018-06-16 16:33	229,058,952
G Icarus	2018-06-16 16:33	284,377,598
Nimbus 2018	2018-06-16 16:33	171,240,476

Censys Metadata

Added At	2018-06-16 16:35:36

censys.io/about

We’re Hiring!

censys.io/careers

Censys Pipeline

Internet-wide Scanning

IPv4, popular websites, CT Logs

Emit facts about the public Internet

```
{
  "ip": "…",
  protocols: {
    "mysql": {
      "server_version": "…",
      "compatibility_flags": "…"
    },
    …
  }
}
```

Data Pipeline

Stream processing of scan data

Raw data stored in custom database

Warehoused in Google BigQuery

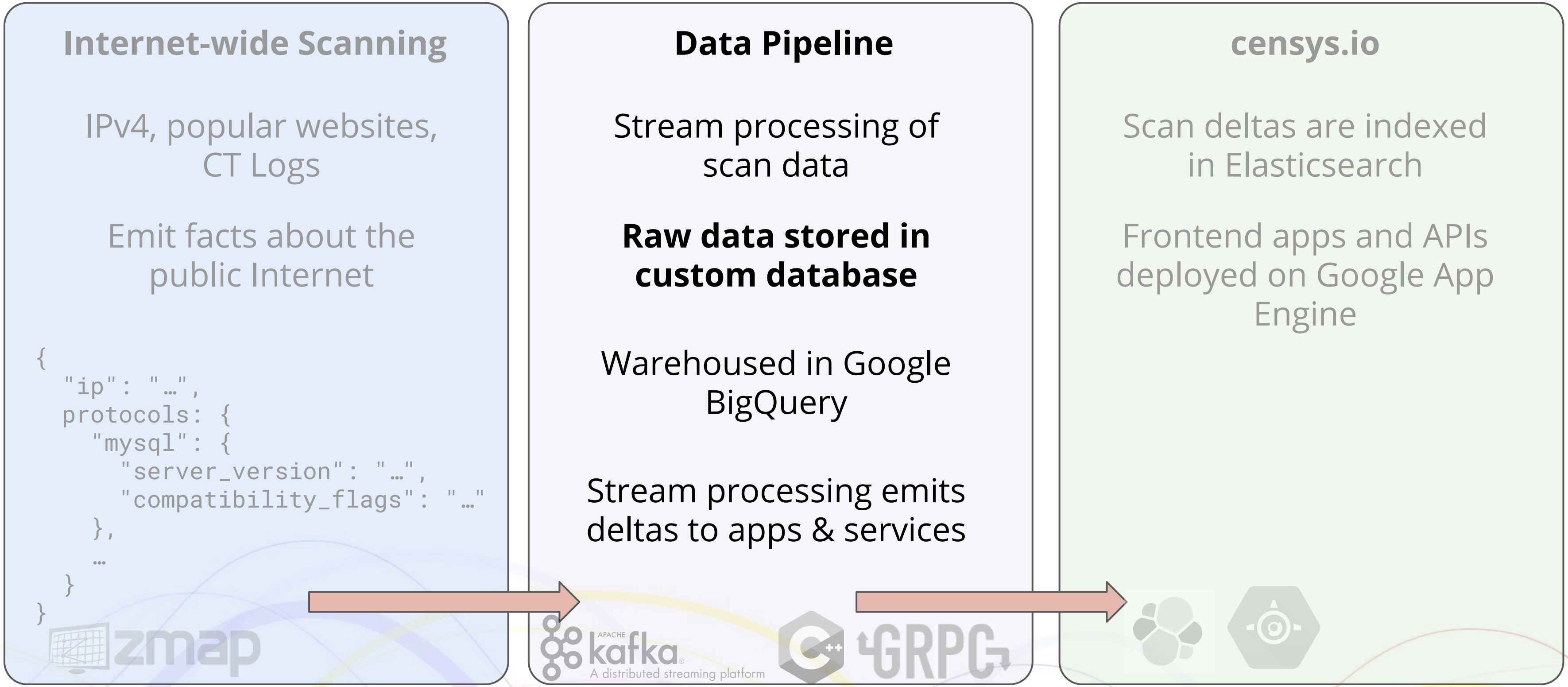
Stream processing emits deltas to apps & services

censys.io

Scan deltas are indexed in Elasticsearch

Frontend apps and APIs deployed on Google App Engine

Censys Pipeline



Internet-wide Scanning

IPv4, popular websites, CT Logs

Emit facts about the public Internet

```
{
  "ip": "…",
  protocols: {
    "mysql": {
      "server_version": "…",
      "compatibility_flags": "…"
    },
    …
  }
}
```

Data Pipeline

Stream processing of scan data

Raw data stored in custom database

Warehoused in Google BigQuery

Stream processing emits deltas to apps & services

censys.io

Scan deltas are indexed in Elasticsearch

Frontend apps and APIs deployed on Google App Engine

Old Data Pipeline Problems

- *"Home-rolled"* or *"bespoke"* is not how we want to describe our database
    a. RAID is not a backup strategy
- Code became unmanageable to change
    a. Stream & batch processing was an amalgamation of C++ & Python
- Built to run on hardware with fast persistent storage
    a. No autoscaling based on scan data backlog or new enrichment services

But it was fast…

Moving our data pipeline to the cloud

Data Pipeline

Streaming processing of scan data

Raw data stored in Google Bigtable

Warehoused in Google BigQuery

Stream processing emits deltas to apps & services

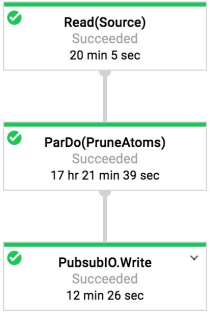
Bigtable + Dataflow + Kubernetes + gRPC + Airflow
=
❤-ish

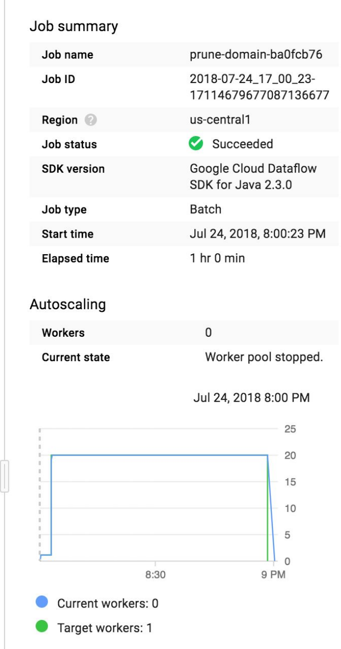
Why Bigtable?

- Distributed hash-table
  - Row-level atomicity
  - Read-modify-write
- Fast, scales linearly
  - Fast lookup and scan
  - ~10K QPS / node (advertised)
- Column version history
  - Consistent snapshots
- Authoritative source-of-truth
  - Use to populate downstream services (e.g. BigQuery, Elasticsearch)

Why Google Dataflow?

- Google Cloud’s stream & batch data processing
- Built with the Apache Beam SDK
  - Single abstraction for stream and batch processing
  - Out-of-the-box integration with our target data stores
  - Throughput-based autoscaling for streaming jobs





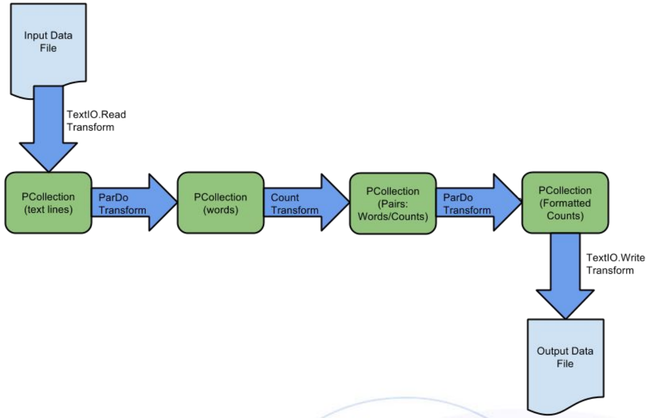
Job summary

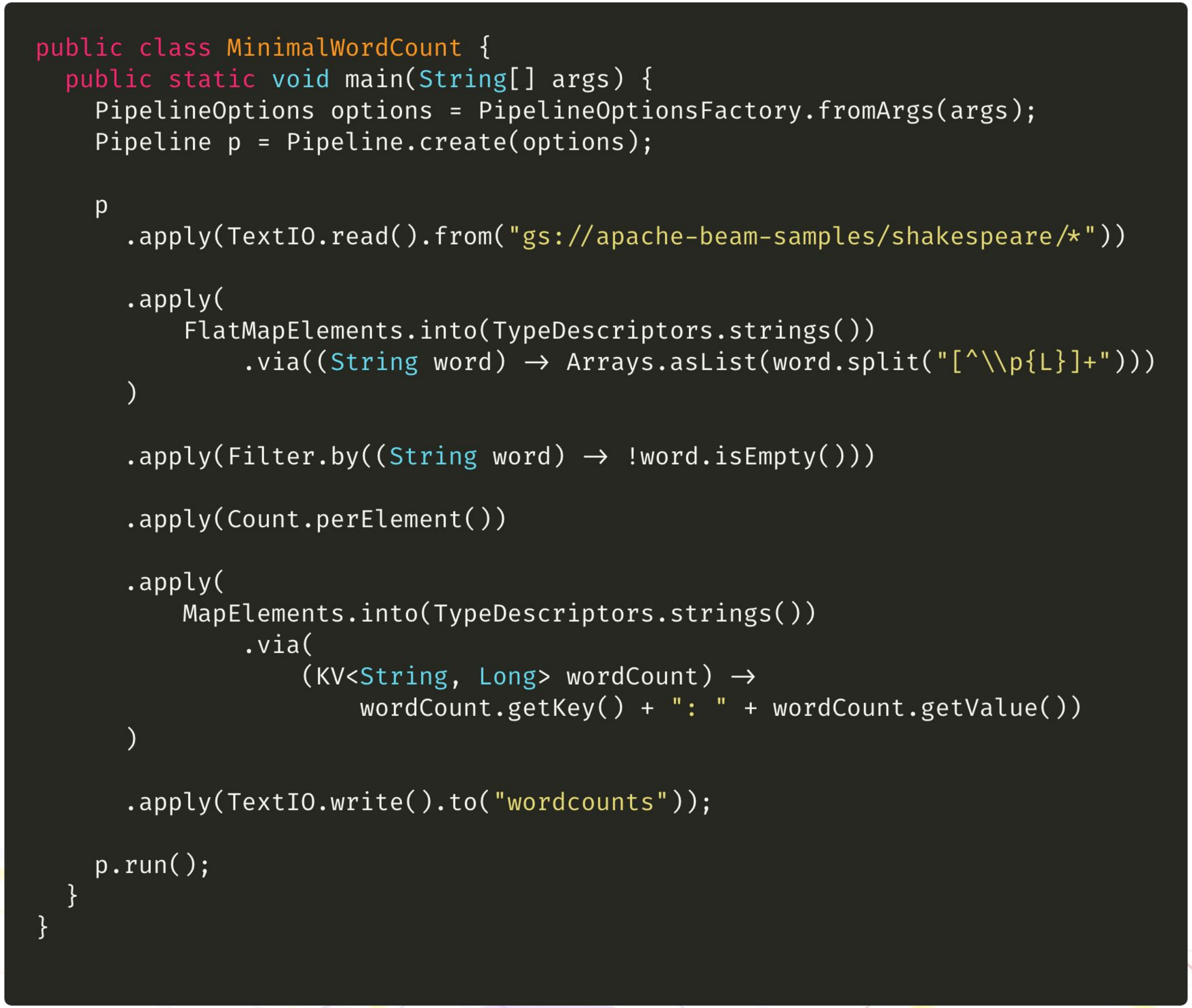
Job name	prune-domain-ba0fcb76
Job ID	2018-07-24_17_00_23-17114679677087136677
Region	us-central1
Job status	✓ Succeeded
SDK version	Google Cloud Dataflow SDK for Java 2.3.0
Job type	Batch
Start time	Jul 24, 2018, 8:00:23 PM
Elapsed time	1 hr 0 min

Autoscaling

Workers	0
Current state	Worker pool stopped.

Example Apache Beam Word Count





```java
public class MinimalWordCount {
  public static void main(String[] args) {
    PipelineOptions options = PipelineOptionsFactory.fromArgs(args);
    Pipeline p = Pipeline.create(options);

    p
      .apply(TextIO.read().from("gs://apache-beam-samples/shakespeare/*"))

      .apply(
          FlatMapElements.into(TypeDescriptors.strings())
              .via((String word) -> Arrays.asList(word.split("[^\\p{L}]+")))
      )

      .apply(Filter.by((String word) -> !word.isEmpty()))

      .apply(Count.perElement())

      .apply(
          MapElements.into(TypeDescriptors.strings())
              .via(
                  (KV<String, Long> wordCount) ->
                      wordCount.getKey() + ": " + wordCount.getValue())
      )

      .apply(TextIO.write().to("wordcounts"));

    p.run();
  }
}
```

via https://beam.apache.org/get-started/wordcount-example/

Why Google Kubernetes Engine?

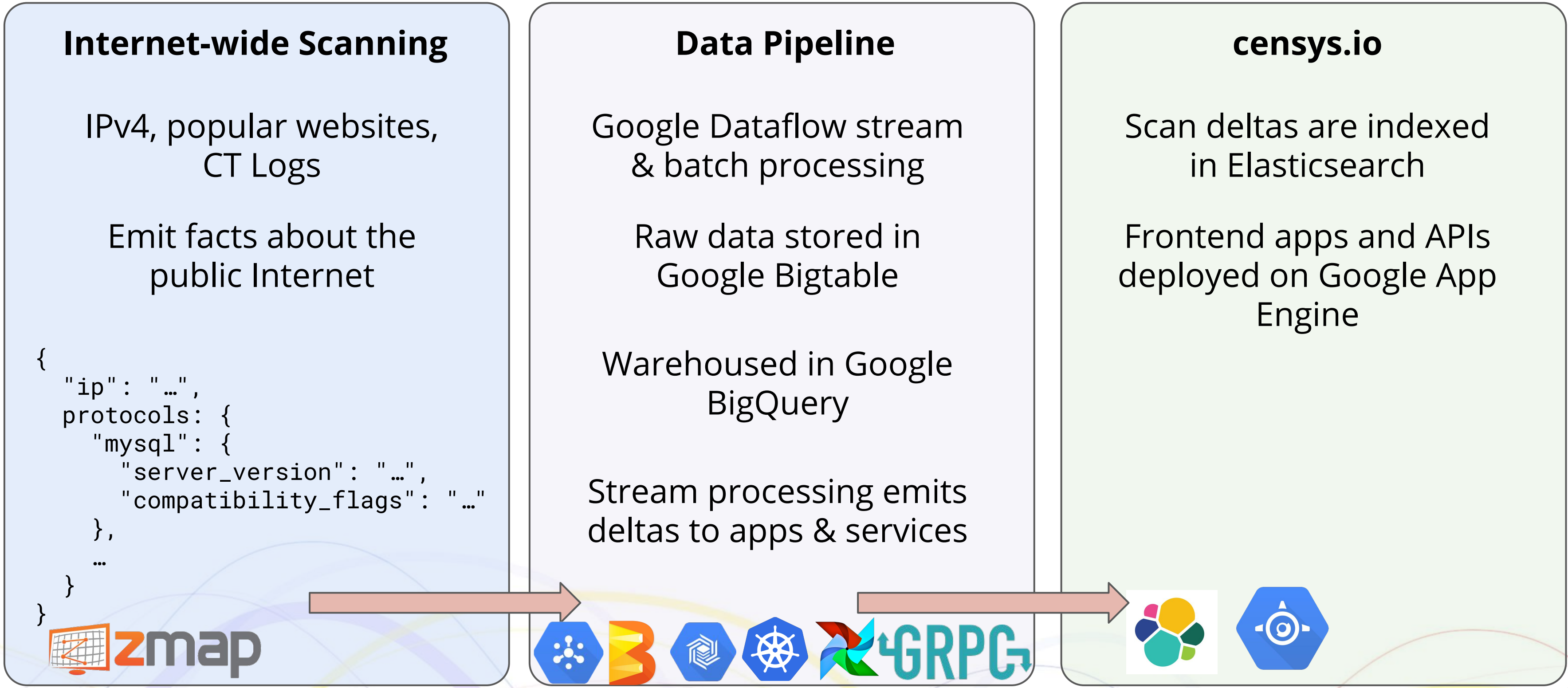
- Pipeline includes various enrichment services
  - Routing info and geolocation lookups of hosts
  - Schema validation and Elasticsearch bulk indexing
  - Fancy attribution analysis
- Services don’t necessarily live in Java Dataflow land
- Unify services’ communication via gRPC APIs
- GKE makes it almost too easy to run a cluster of our services :-)

Why Apache Airflow?

- Scheduler of batch processing
- Workloads expressed via Python DSL
- Good toolchain to build “deterministic” scheduled tasks
- Easily deployable via Google Cloud Composer

Putting it all together

Censys Cloud Pipeline



Internet-wide Scanning

IPv4, popular websites, CT Logs

Emit facts about the public Internet

```
{
  "ip": "…",
  protocols: {
    "mysql": {
      "server_version": "…",
      "compatibility_flags": "…"
    },
    …
  }
}
```

Data Pipeline

Google Dataflow stream & batch processing

Raw data stored in Google Bigtable

Warehoused in Google BigQuery

Stream processing emits deltas to apps & services

censys.io

Scan deltas are indexed in Elasticsearch

Frontend apps and APIs deployed on Google App Engine

What We’ve Learned

- Hard to autoscale when talking to network services
  - CPU usage is likely low
  - Dataflow may fuse steps, need to inject max parallelism manually with `GroupBy`
- The state of Kubernetes remains complicated
  - Resource requests/limits are not intuitive
  - Helm is a mess, but so is DIY
  - Buzzwords make learning difficult
- Load balancing correctly takes time
  - Don’t build push services when you need pull services
  - Sometimes you need an L7 load balancer (k8s Ingress)
  - Cloud Egress bandwidth is expensive
- A “cloud migration” often means rewrite, and takes a whole team
  - Andrew knows all Simpsons references
  - Don’t start a startup while a PhD student
  - isthisadag.tumblr.com

Stats

- 714MM certificates
  - *Unbounded, infinitely growing dataset*
  - *On track for 1B by the end of the year*

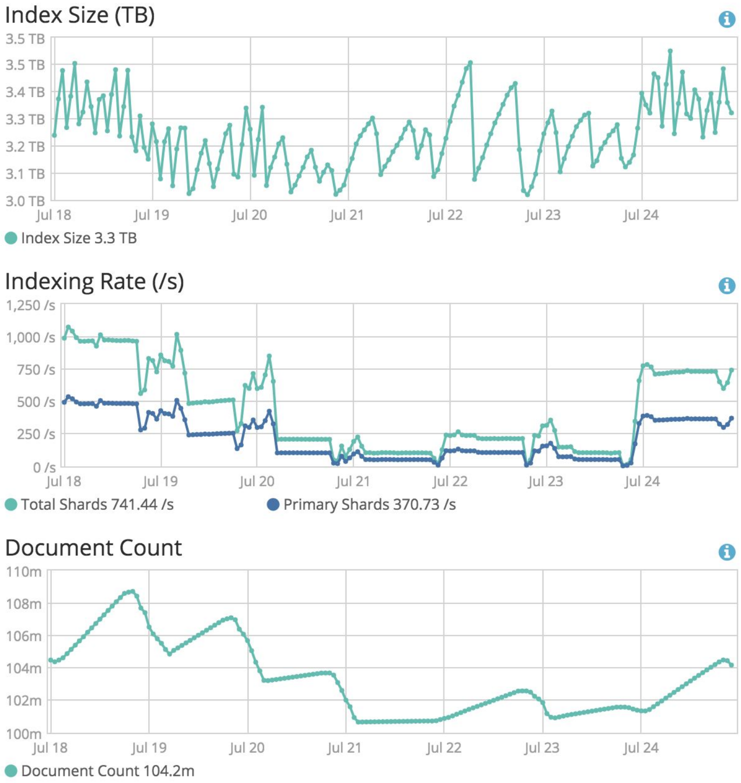
- 828,025,790 documents in production ES cluster
  - *Only going up as we add more protocols*

- ~16TB scan traffic per day
  - *Only going up as we add more protocols*

- Sustain >2K scan results per second, with bursts up to 25K
  - *Only going up as we add more protocols*

Elasticsearch

- 35-node cluster
  - 2 frontend-only
  - 33 data nodes with 1 TB storage
- ~24 TB across all indexes
- 3 indexes
  - **IPv4 (3.5 TB) [see right]**
  - Domain (0.5 TB)
  - Certificates (20 TB)



censys

Storing & Analyzing the Internet

Orchestructure, July 25, 2018

bit.ly/orchestructure-censys-2018-07-25

Andrew Sardone
andrew@censys.io

David Adrian
dadrian@censys.io

Questions?

Further Reading

- ZMap
  - Open-source tools are on Github
- The original Censys research paper
  - Other research papers co-authored by Zakir, David, or Alex
- Google Dataflow
  - Getting to Know Cloud Dataflow
- Google Bigtable
- Censys is hiring!