Telemetry doesn't have to be scary

Developer focused metrics that make sense.

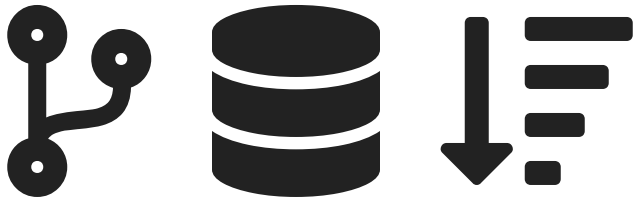


Table of Contents

New talk, who dis?

- Ben Ford
- Developer Advocate @ Puppet
- @binford2k:

Why we need metrics

Photo by https://unsplash.com/@benknight17

It's not a huge task to design and build a simple, single-purpose, bespoke Puppet module in a short period of time.

Maintenance is a different story

- Hey, can we add ArchLinux support?
- What about Windows 2012?
- Here's a PR for a defined type to manage the database too... (MariaDB only)

Development resources

? Every single platform or feature you support takes time to maintain it!

Data driven development decisions

? Two roads diverged in a wood, and I—
I took the one less traveled by,
And that has made all the difference.
-- Robert Frost

Impact Analysis

Photo by https://unsplash.com/@the_roaming_platypus

Rangefinder

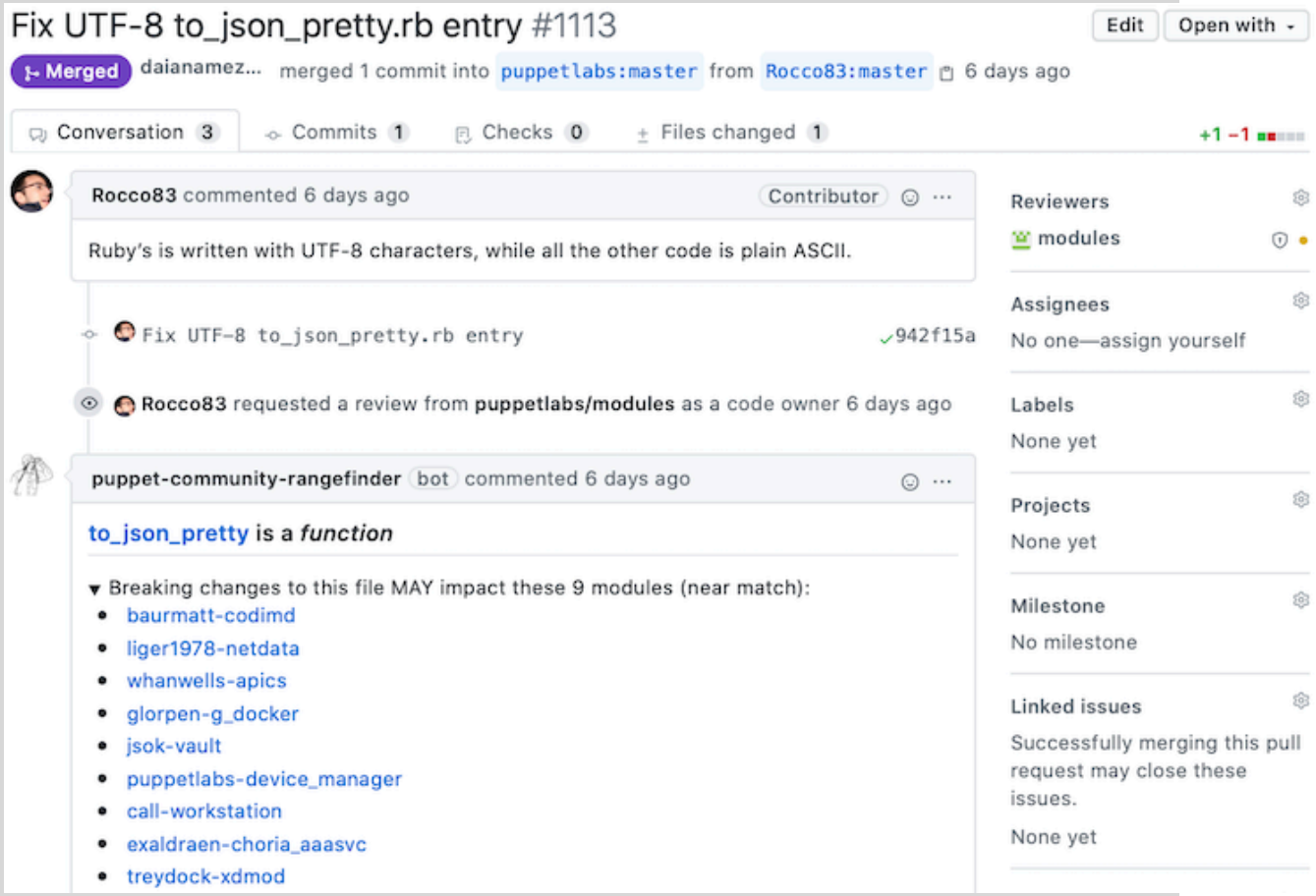
```
[~/Projects/puppetlabs-concat]$ rangefinder manifests/fragment.pp
[concat::fragment] is a _type_
=================================
The enclosing module is declared in 173 of 575 indexed public Pup

Breaking changes to this file WILL impact these modules:
  * nightfly-ssh_keys (https://github.com/nightfly19/puppet-ssh_k
  * viirya-mit_krb5 (git://github.com/viirya/puppet-mit_krb5.git)
  * rjpearce-opendkim (https://github.com/rjpearce/puppet-opendki
  * shadow-tor (git://github.com/LeShadow/puppet-tor.git)
[...]

Breaking changes to this file MAY impact these modules:
  * empi89-quagga (UNKNOWN)
  * unyonsys-keepalived (UNKNOWN)
  * Flameeyes-udevnet (UNKNOWN)
  * ricbra-ratbox (git://github.com/ricbra/puppet-ratbox.git)
[...]
```

Identify the public Forge modules using parts of your code.

---

https://github.com/puppetlabs/puppet-community-rangefinder

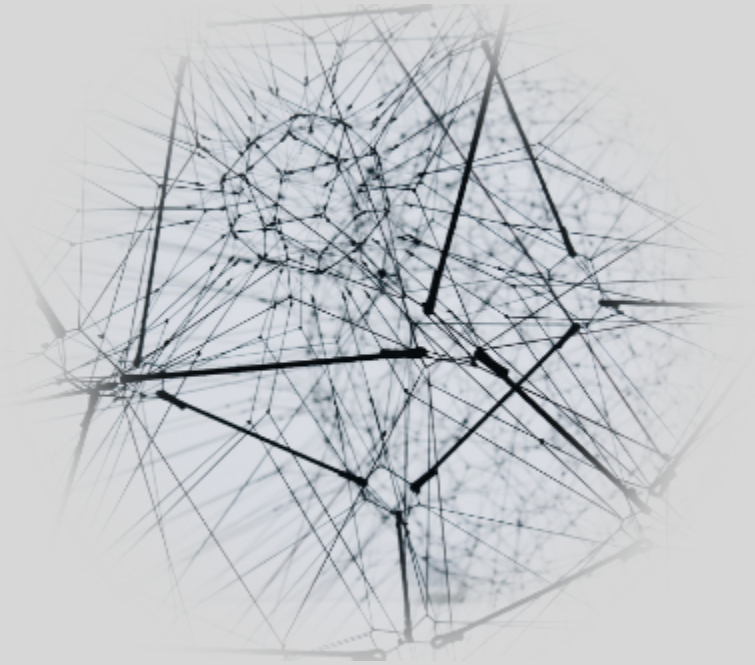
Run it on a pull request!



Rangefinder also comes in a GitHub App flavor.

- https://github.com/apps/puppet-community-rangefinder

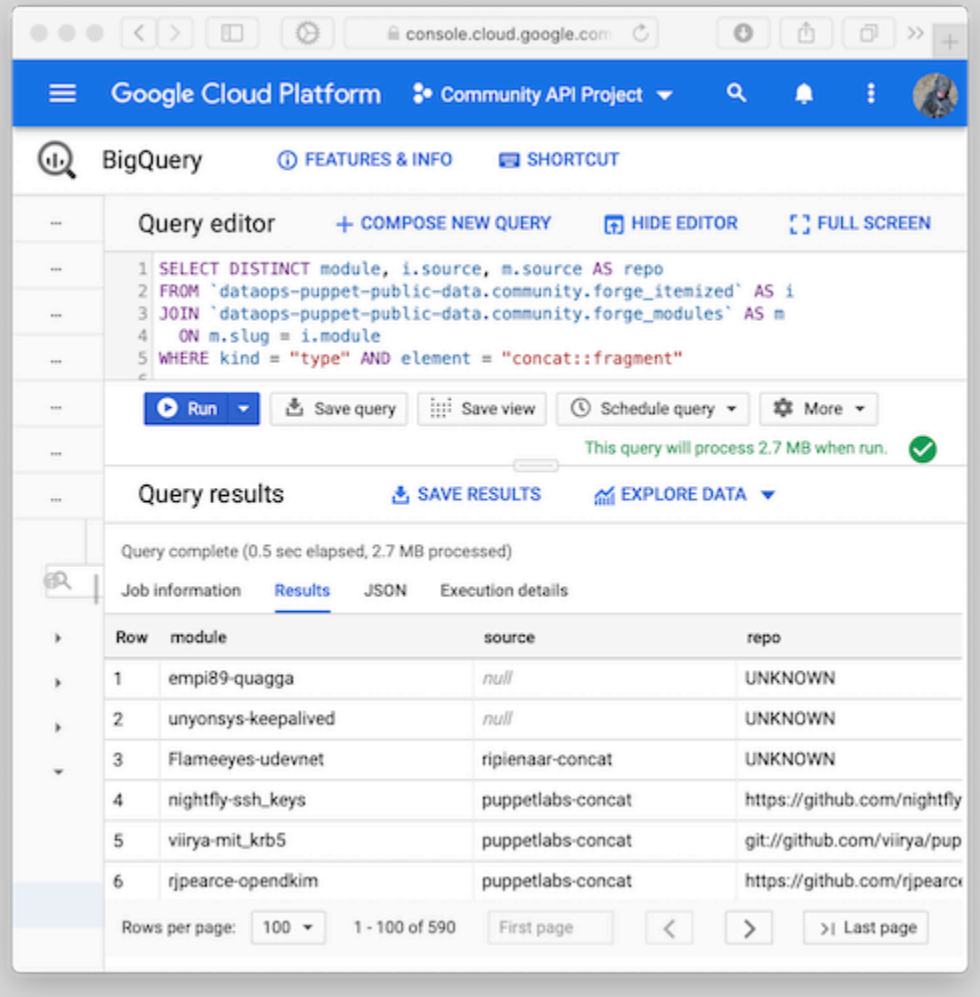
Static analysis of dependencies



- Works by dissecting latest releases of all modules on the Forge
- Static code analysis identifies what every module uses
    - resource types
    - functions
    - classes
- Runs weekly to generate dependency network

- This uses my `itemize` library: https://github.com/binford2k/binford2k-itemize

Dependency data is public



The data Rangefinder uses is stored in a public BigQuery database.

Find more information about how it works and other cool queries you can run on the dataset p
it at my blog: https://binford2k.com/2020/04/06/rangefinder/

Major limitation

- Rangefinder only works on public Forge modules (currently).
- Cannot identify how code is used by profiles or private modules.

Privacy. It's a thing.

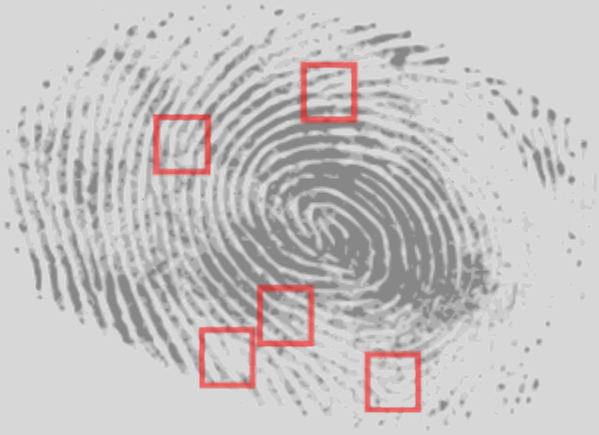
Photo by https://unsplash.com/@pawel_czerwinski

Business Identifiable Information

? Sharing too much implementation information about how an infrastructure works is ju
asking for security compromises.

- Most of you know BII's big cousin, PII.
- But many of the same considerations apply to businesses.
- Many companies don't allow phone-home metrics at all.

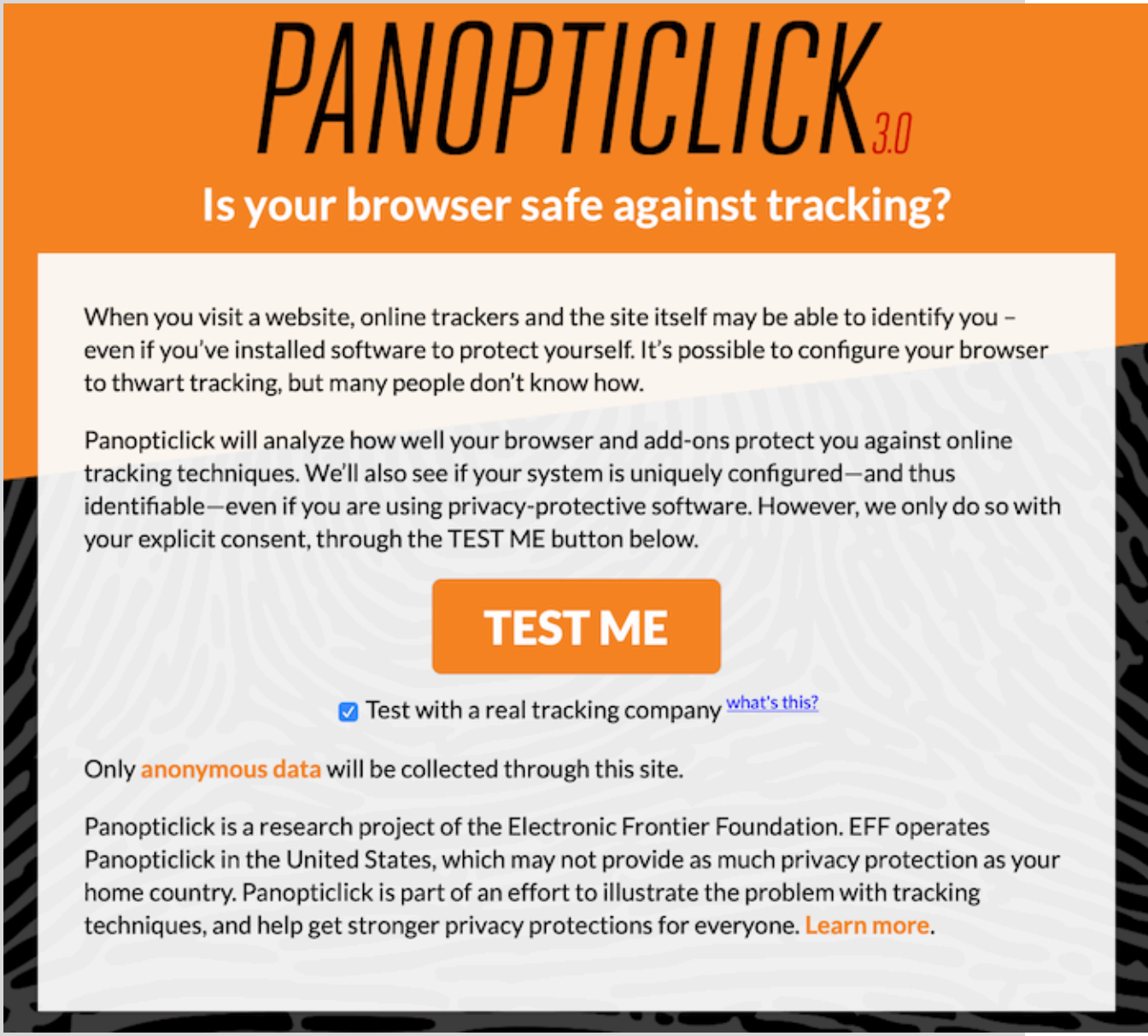
Fingerprinting



- A more insidious danger is fingerprinting.
- Identifying individuals by properties or habits.
- Also applies to infrastructures.

? If you saw that a site was in the `CEST` time zone, had the locale set to `fr-ch`, and thousands of nodes classified with various HPC modules, then you might guess that were looking at CERN.

Live example of fingerprinting



http://panopticlick.eff.org/

Hard requirements

1. Provide real value to the end user. (that's you)
2. Transparent about what data is collected, why, and what it's used for.
3. Give you the ability to opt in or out without gating other benefits.
4. The data collected can be aggregated to share publicly.

Telemetry Data Model

Photo by https://unsplash.com/@crissyjarvis

Public dataset

No access to *individual records* themselves

```
-- The top ten most used classes in the ecosystem
SELECT name, count
FROM `dataops-puppet-public-data.aggregated.class_usage_count`
ORDER BY count DESC
LIMIT 10
```

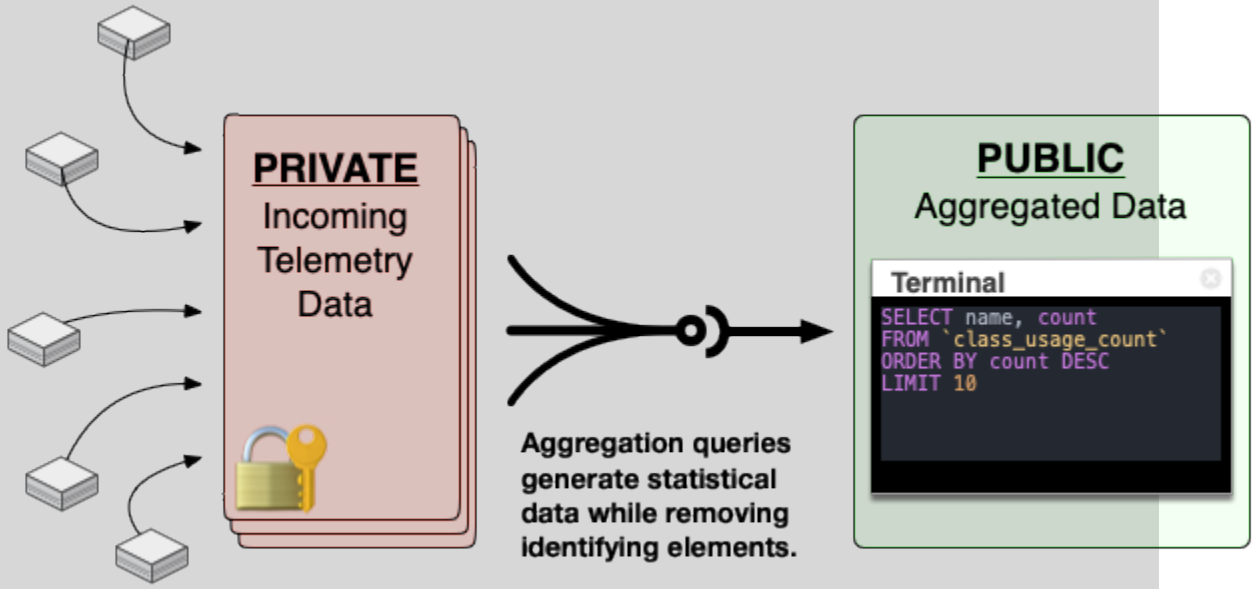

```
[
  { "name": "Resource_api::Agent", "count": "272" },
  { "name": "Account",             "count": "272" },
  { "name": "Ssl::Params",         "count": "272" },
  { "name": "Classification",      "count": "272" },
  { "name": "Os_patching",         "count": "269" },
  { "name": "Ntp",                 "count": "265" },
  { "name": "Ntp::Install",        "count": "265" },
  { "name": "Ntp::Config",         "count": "265" },
  { "name": "Ntp::Service",        "count": "265" },
  { "name": "Zsh::Params",         "count": "265" }
]
```

You'll need a Google Cloud account and then you can access the dataset with your browser via BigQuery Console. Then you can run any queries you'd like.

Aggregating data

Data is generated via SQL queries:

```
-- Generate an aggregate table of all classes used and their coun
SELECT classes.name, classes.count
FROM `bto-dataops-datalake-prod.dujour.community_metrics`,
    UNNEST(classes) as classes
GROUP BY classes.name, classes.count
```



---

https://github.com/puppetlabs/dropsonde-aggregation

Private dataset

So about that sensitive data...

- Telemetry reports are not terribly sensitive, but they are fingerprintable.
- So we store them only in a secured dataset.
- Limited access, only myself and our BigQuery admin team.

? Our own developers who want to use collected metrics data have to do it via aggregati
queries just like you would.

Metric Plugin Design

Photo by https://unsplash.com/@danielkcheung

Plugin hook design

- Plugins are implemented as hooks.
    - `description`
    - `schema`
    - `example`
    - `run`
- Some hooks are optional
    - `initialize`
    - `setup`
    - `cleanup`

Schema hook

Describes data that the plugin is allowed to return.

- Returns a BigQuery schema.
- JSON format with an array of row definitions

```
def self.schema
  [
    {
      "description": "The number of environments",
      "mode": "NULLABLE",
      "name": "environment_count",
      "type": "INTEGER"
    }
  ]
end
```

? See full schema of all plugins with `dropsonde dev schema`

Run hook

Generates and returns data for the metric

- Returns an array of rows of data.
- Can return any data that the Puppet Server node can generate.
- Use PuppetDB, codebase, etc.
- Methods provided to filter out private modules.
- Must match schema or plugin fails.

```
def self.run
  [
    :environment_count => Puppet.lookup(:environments).list.count
  ]
end
```

Example hook

Representative example of sample data

- Provide one element of randomized representative data.
- Used to generate a dataset that's shaped like the real dataset.
- Must match the `schema` or plugin fails.
- Used for developers to generate aggregation queries.

```
def self.example
  [
    :environment_count => rand(1..100),
  ]
end
```

- Available as `dataops-puppet-public-data:community.community_metrics`

Previewing a telemetry report

```
$ dropsonde --enable environments preview

                      Puppet Telemetry Report Preview
                      ===============================

Dropsonde::Metrics::Environments
--------------------------------
This group of metrics gathers information about environments.
- environment_count: The number of environments
    4


Site ID:
bab5a61cb7af7d37fa65cb1ad97b2495b4bdbc85bac7b4f9ca6932c8cd9038dd
```

- Before you submit data, you can see what's generated
- This example selects only a single metric
- Site ID is a cryptographic hash that just correlates reports
    - Can be regenerated by providing a `seed` value

Data validation

- The plugin cannot return data that it does not declare.
- The database schema is also defined from the plugin schemas.

Using the Data

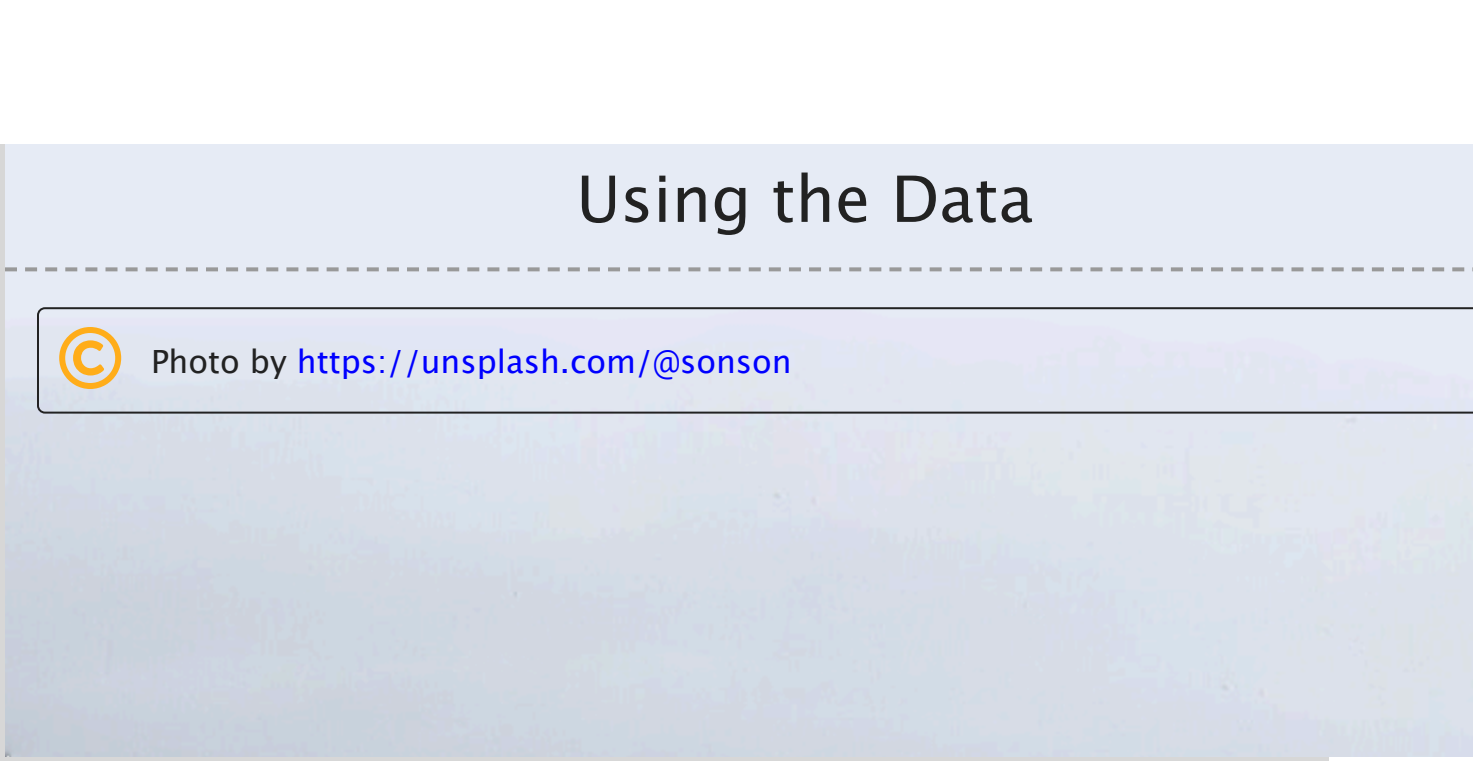
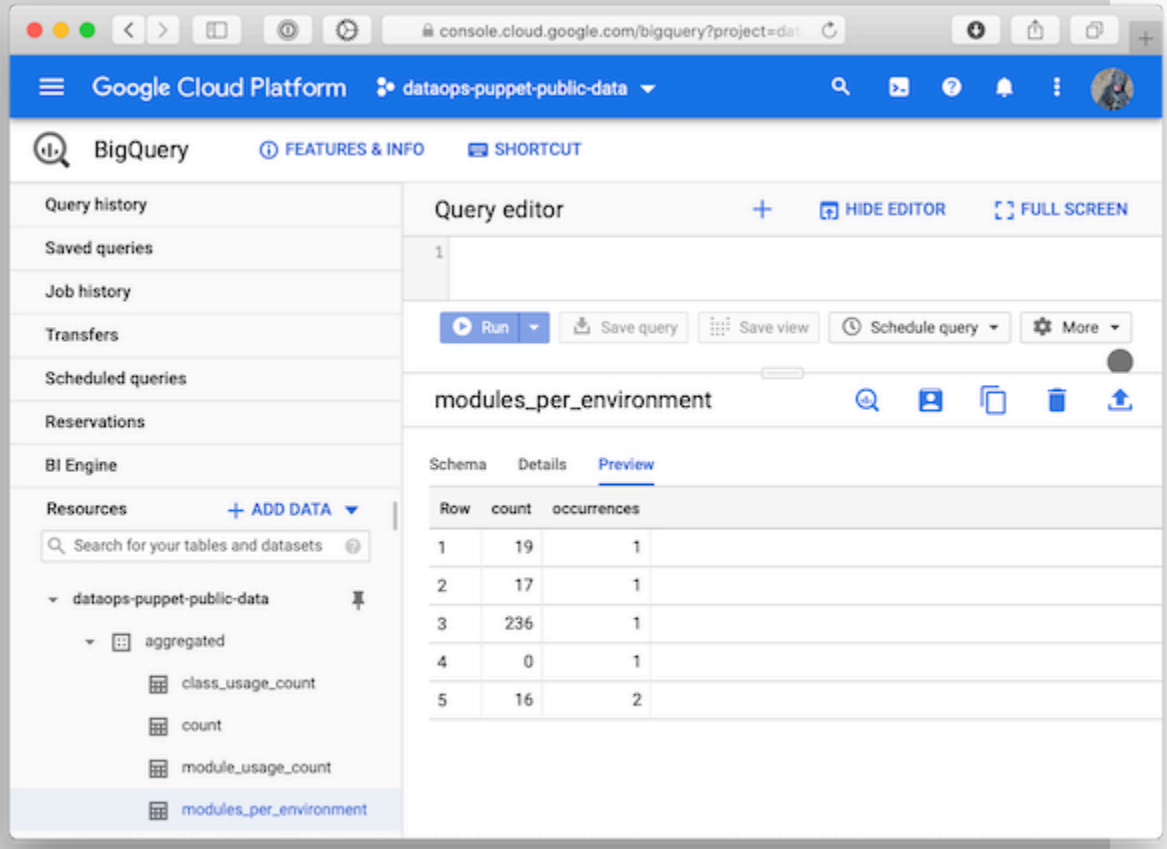


Photo by https://unsplash.com/@sonson

Public data



- All you need is a Google Cloud account.
- Then you can explore the datasets.

Once you have an account, browse to https://console.cloud.google.com/bigquery?p=dataops-puppet-public-data&d=aggregated to check out the `dataops-puppet-public-data` proj

Interesting queries

- Dataset includes:
    - Forge module metadata
    - Static analysis of Forge modules
    - GitHub repositories that look like Puppet modules
    - Aggregated telemetry data
- Combine them as you will.

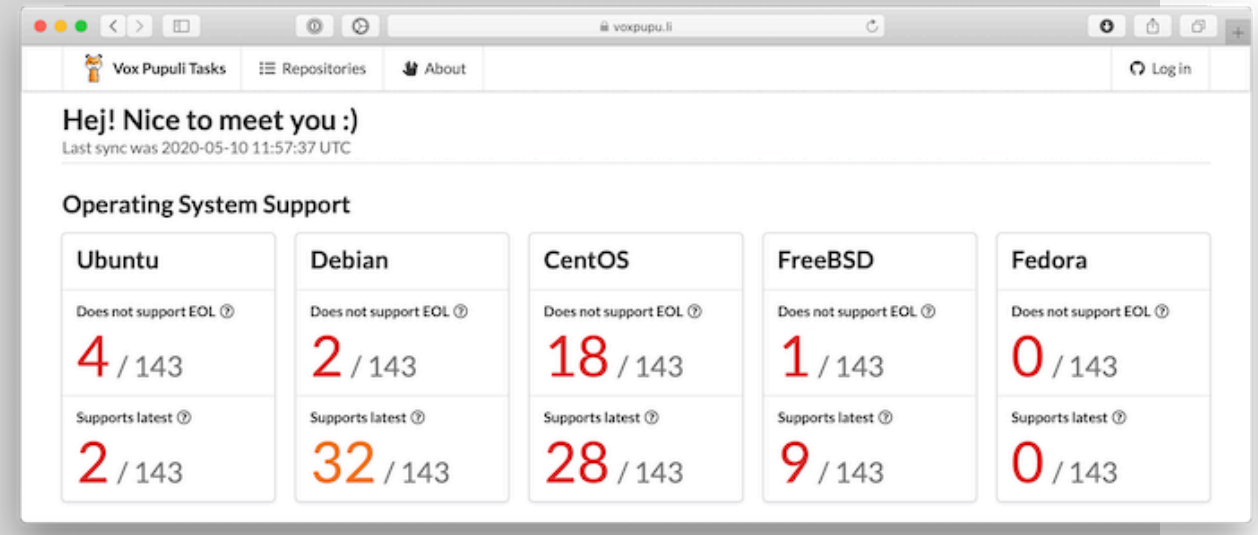
Example: a list of the modules on the Forge that define custom native types:

```
SELECT DISTINCT g.repo_name, f.slug
FROM `dataops-puppet-public-data.community.github_ruby_files` g
JOIN `dataops-puppet-public-data.community.forge_modules` f
    ON g.repo_name = REGEXP_EXTRACT(f.source, r'^(?:https?:\/\/gi
WHERE STARTS_WITH(g.path, 'lib/puppet/type')
LIMIT 1000
```

So what else is possible?

Who knows, really.

- Maybe we can incorporate Impact Analysis into PDK validations.
- Maybe the Forge will start surfacing some of these metrics. (hint: it will)
- Maybe Vox Pupuli will incorporate usage data into their Tasks management interface.
- Maybe *you* will write something I haven't even thought of.



Get Involved!

Photo by https://unsplash.com/@lunarts

What I'm asking from you

- This is totally opt-in only right now.
- `puppet module install puppetlabs/dropsonde`
- Check out the example dataset and contribute some aggregation queries.
- Develop and contribute a metric plugin.
- Doesn't have to be strictly Puppet, just ecosystem:
    - Reid already made a plugin to see the impact of building a better control repo and `Puppetfile` for better r10k deploys.
    - David could see how many people are using the Rocket.chat integration for `voxpupuli/puppet-webhook`.
- `puppet module install puppetlabs/dropsonde`
- Build some tooling that uses the public data.

? Seriously, install the module and classify your master. It would help a ton.

? `Dropsonde`: an expendable weather reconnaissance device created by the National Center for Atmospheric Research, designed to be dropped from an aircraft at altitude over water to measure storm conditions as the device falls to the surface.

? Installing Dropsonde is super straightforward. Install the `puppetlabs-dropsonde` Puppet module and classify your Puppet master with the `dropsonde` class. If you hav multiple compile nodes, just classify the primary Puppet server. This will install the too and configure a weekly cron job to submit the reports.

© Photo by https://unsplash.com/@mattbotsford

Resources and Q&A

- https://github.com/puppetlabs/dropsonde
- https://github.com/puppetlabs/puppet-community-rangefinder
- https://github.com/apps/puppet-community-rangefinder
- https://binford2k.com/tags/#telemetry-ref

- https://github.com/puppetlabs/dropsonde
- https://github.com/puppetlabs/puppetlabs-dropsonde
- https://github.com/puppetlabs/dropsonde-aggregation
- https://binford2k.com/tags/#telemetry-ref
    - Impact Analysis of Puppet Modules
    - Downstream impact of pull requests
    - Telemetry that doesn't suck
    - Gathering metrics with a new Dropsonde plugin