

Telemetry doesn't have to be scary

Developer focused metrics that make sense.



Table of Contents

[Why we need metrics](#)

[Impact Analysis](#)

[Privacy. It's a thing.](#)

[Telemetry Data Model](#)

[Metric Plugin Design](#)

[Using the Data](#)

[Get Involved!](#)

New talk, who dis?



- Ben Ford
- Developer Advocate @ Puppet
- @binford2k: [Twitter](#) [GitHub](#) <#>

Why we need metrics



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

I PDK



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



Photo by <https://unsplash.com/@leliejens>

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 Puppet
modules.

Breaking changes to this file WILL impact these modules:
  * nightfly-ssh_keys (https://github.com/nightfly19/puppet-ssh_keys)
  * viirya-mit_krb5 (git://github.com/viirya/puppet-mit_krb5.git)
  * rjpearce-opendkim (https://github.com/rjpearce/puppet-opendkim)
  * 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!

Fix UTF-8 to_json_pretty.rb entry #1113

Edit Open with ▾

1 Merged daianamez... merged 1 commit into `puppetlabs:master` from `Rocco83:master` 6 days ago

Conversation 3 Commits 1 Checks 0 Files changed 1 +1 -1 ━━━━

Rocco83 commented 6 days ago Contributor ⓘ ...
Ruby's is written with UTF-8 characters, while all the other code is plain ASCII.

Fix UTF-8 to_json_pretty.rb entry 942f15a

Rocco83 requested a review from `puppetlabs/modules` as a code owner 6 days ago

puppet-community-rangefinder (bot) commented 6 days ago ⓘ ...
to_json_pretty is a *function*

Breaking changes to this file MAY impact these 9 modules (near match):

- `baurmatt-codimd`
- `liger1978-netdata`
- `whanwells-apics`
- `glorpen-g_docker`
- `jsok-vault`
- `puppetlabs-device_manager`
- `call-workstation`
- `exaldraen-choria_aaasvc`
- `treydock-xmod`

Reviewers modules

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Linked issues Successfully merging this pull request may close these issues.

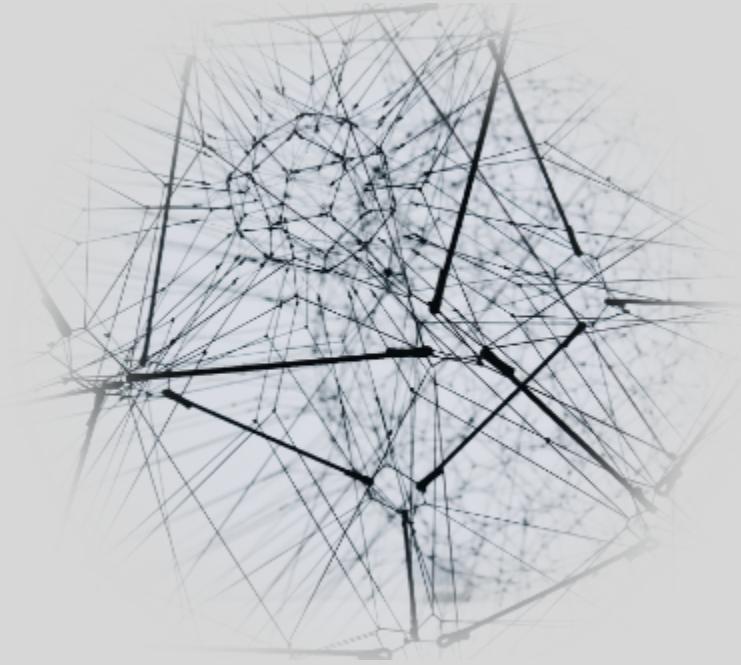
None yet



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>



Photo by <https://unsplash.com/@alinnaaaa>

Dependency data is public

The screenshot shows the Google Cloud Platform BigQuery interface. At the top, there's a navigation bar with icons for back, forward, refresh, and search, followed by the URL 'console.cloud.google.com'. Below that is the 'Google Cloud Platform' logo and a dropdown for 'Community API Project'. On the right side of the header are icons for notifications and user profile.

The main area is titled 'BigQuery' with tabs for 'FEATURES & INFO' and 'SHORTCUT'. Below the tabs are buttons for 'Query editor' (selected), '+ COMPOSE NEW QUERY', 'HIDE EDITOR', and 'FULL SCREEN'.

The 'Query editor' section contains a code editor with the following SQL query:

```
1 SELECT DISTINCT module, i.source, m.source AS repo
2 FROM `dataops-puppet-public-data.community.forge_itemized` AS i
3 JOIN `dataops-puppet-public-data.community.forge_modules` AS m
4   ON m.slug = i.module
5 WHERE kind = "type" AND element = "concat::fragment"
```

Below the code editor are buttons for 'Run', 'Save query', 'Save view', 'Schedule query', and 'More'. A note says 'This query will process 2.7 MB when run.' with a checkmark icon.

The 'Query results' section shows a table with the following data:

Row	module	source	repo
1	empi89-quagga	null	UNKNOWN
2	unyonsys-keepalived	null	UNKNOWN
3	Flameeyes-udevnet	riplenaar-concat	UNKNOWN
4	nightfly-ssh_keys	puppetlabs-concat	https://github.com/nightfly
5	viirya-mit_krb5	puppetlabs-concat	git://github.com/viirya/pup
6	rjpearce-opendkim	puppetlabs-concat	https://github.com/rjpearce

At the bottom of the results table are buttons for 'Rows per page' (set to 100), 'First page', 'Last page', and navigation arrows.



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.



Photo by <https://unsplash.com/@sguestsmith>

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 just 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 you were looking at CERN.

Live example of fingerprinting

The screenshot shows the Panopticlick 3.0 test page. The header features the title "PANOPTICCLICK 3.0" in large, bold, black letters, with "3.0" in red. Below it is the question "Is your browser safe against tracking?" in white. A central text block explains that websites can identify users even with privacy software. It then describes the test's purpose: analyzing browser protection against tracking and unique configuration. A large orange button labeled "TEST ME" is prominent. Below it is a checked checkbox for testing with a real tracking company, with a "what's this?" link. A note states that only anonymous data is collected. At the bottom, it mentions the project is a research effort by the Electronic Frontier Foundation.

When you visit a website, online trackers and the site itself may be able to identify you – even if you've installed software to protect yourself. It's possible to configure your browser to thwart tracking, but many people don't know how.

Panopticclick will analyze how well your browser and add-ons protect you against online tracking techniques. We'll also see if your system is uniquely configured—and thus identifiable—even if you are using privacy-protective software. However, we only do so with your explicit consent, through the TEST ME button below.

TEST ME

Test with a real tracking company [what's this?](#)

Only **anonymous data** will be collected through this site.

Panopticclick is a research project of the Electronic Frontier Foundation. EFF operates Panopticclick in the United States, which may not provide as much privacy protection as your home country. Panopticclick is part of an effort to illustrate the problem with tracking techniques, and help get stronger privacy protections for everyone. [Learn more.](#)

<http://panopticclick.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
```

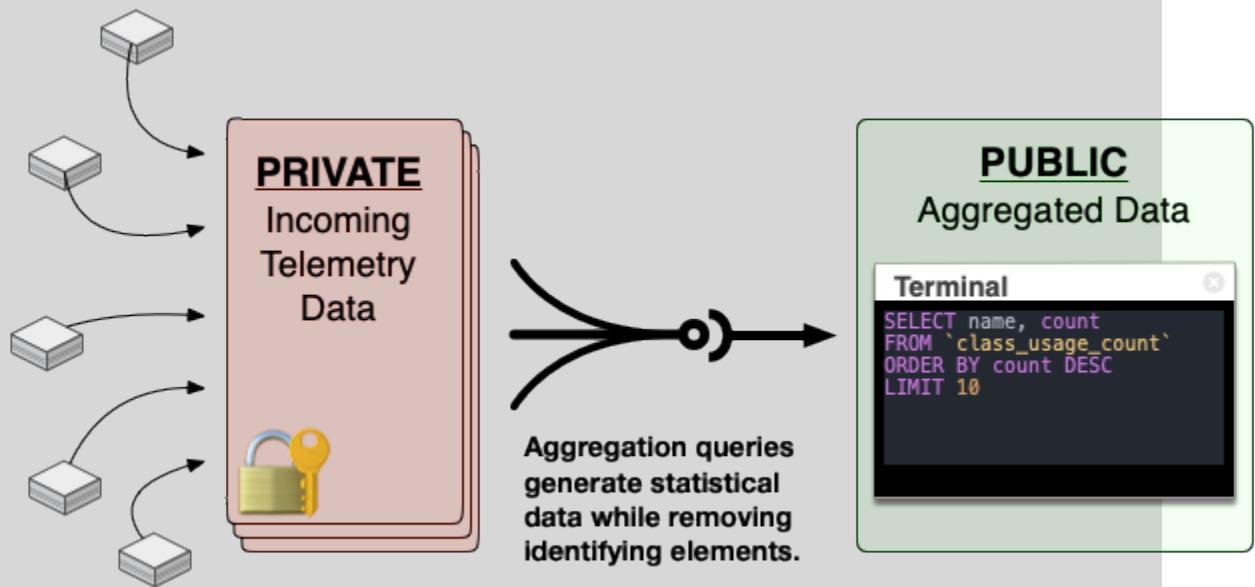
```
[{"name": "Resource_api::Agent", "count": "272"}, {"name": "Account", "count": "272"}, {"name": "Ssl::Params", "count": "272"}, {"name": "Classification", "count": "272"}, {"name": "Os_patching", "count": "269"}, {"name": "Ntp", "count": "265"}, {"name": "Ntp::Install", "count": "265"}, {"name":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

The screenshot shows the Google Cloud Platform BigQuery interface. The left sidebar lists various sections: Query history, Saved queries, Job history, Transfers, Scheduled queries, Reservations, and BI Engine. Under Resources, there is a search bar and a tree view of datasets. The 'dataops-puppet-public-data' dataset is expanded, showing its sub-tables: aggregated, class_usage_count, count, module_usage_count, and modules_per_environment. The 'modules_per_environment' table is selected and previewed in the main area. The preview shows the following data:

Row	count	occurrences
1	19	1
2	17	1
3	236	1
4	0	1
5	16	2

- 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` project.

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.

The screenshot shows a web browser window for the Vox Pupuli Tasks application at voxpupu.li. The interface includes a navigation bar with links for 'Vox Pupuli Tasks', 'Repositories', 'About', and 'Login'. Below the navigation is a greeting message 'Hej! Nice to meet you :)'. A note indicates 'Last sync was 2020-05-10 11:57:37 UTC'. The main content area is titled 'Operating System Support' and displays five cards for different distributions:

Operating System	Status	Count / Total
Ubuntu	Does not support EOL ⓘ	4 / 143
Ubuntu	Supports latest ⓘ	2 / 143
Debian	Does not support EOL ⓘ	2 / 143
Debian	Supports latest ⓘ	32 / 143
CentOS	Does not support EOL ⓘ	18 / 143
CentOS	Supports latest ⓘ	28 / 143
FreeBSD	Does not support EOL ⓘ	1 / 143
FreeBSD	Supports latest ⓘ	9 / 143
Fedor	Does not support EOL ⓘ	0 / 143
Fedor	Supports latest ⓘ	0 / 143

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 have multiple compile nodes, just classify the primary Puppet server. This will install the tool 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