



# THE METRICS THAT MATTER



## JON CARDONA

Services Engineer



Hollywood

✉️ [hollywood@github.com](mailto:hollywood@github.com)



[hollyw00d](https://twitter.com/hollyw00d)



## STEFAN STÖLZLE

Services Engineer



stoe

✉️ [stefan@github.com](mailto:stefan@github.com)



[me\\_stoe](https://twitter.com/me_stoe)



## LARS SCHNEIDER

Services Account Engineer



[larsxschneider](https://github.com/larsxschneider)

✉️ [lars@github.com](mailto:lars@github.com)



[kit3bus](https://twitter.com/kit3bus)

# AGENDA

- Why metrics matter
- Why context matters
- Who uses metrics?
- What data do I need?
- Dashboard examples
- How to build my dashboard



# WHY METRICS MATTER

# WHY METRICS MATTER



The ability to attach a tangible measurement to internal processes is an essential tool to help identify how a process is functioning.

The objective of these measurements, used along with deeper contextual meaning behind each value, will help provide a baseline for any potential process improvements.



# WHY CONTEXT MATTERS

# BECAUSE NUMBERS CAN BE MISLEADING



## Example

Organization A has a multitude of product lines. Within their offices hang multiple monitors displaying a dashboard full of development metrics grouped by development team. An example of some of the metrics that are being displayed are:

- Average Number of Daily Commits
- Average Age of Pull Requests
- Average Lines of Code per Developer

What are some ways that viewing these numbers at a high level could create false positives?



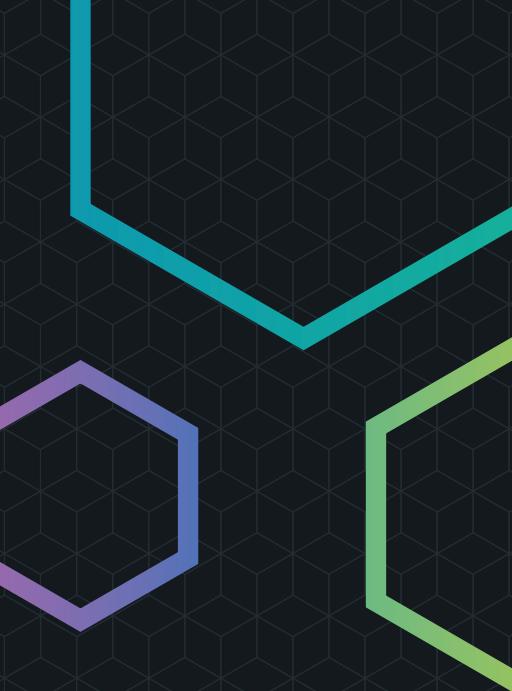
# WHO USES METRICS?

# ADMINS

As a GitHub Enterprise Admin I want to ...

- get alerts regarding the GitHub Enterprise instance performance.
- know about the GitHub Enterprise usage.
- see the GitHub Enterprise instance health.

# CXOS



As the CxO of a company using GitHub Enterprise I want to ...

- see the (total) number of users on GitHub Enterprise.
- see the (total) number of repositories on GitHub Enterprise.
- see the number of active Pull Requests over time.

# DEV MANAGER/DIRECTOR/TEAM LEAD

As the Manager of a team using GitHub Enterprise I want to ...

- see the number of active Pull Requests over time.
- see the number of currently open Pull Requests.
- see the number of commits over time.
- know how long it takes (on average) to close a Pull Request.
- know how long it takes (on average) to review a Pull Request.

# DEVELOPERS



As the Developer of a team using GitHub Enterprise I want to...

- see if I'm constantly adding code to, or rewriting code, in the same files or the same conceptual area
- see the type of files I'm committing the most (e.g. JavaScript vs C#)
- see the number of open Pull Requests that require my approval
- see the number of stale issues assigned to me or created by me



# WHAT DATA DO I NEED? AND WHERE DO I GET IT FROM?

# MONITORING GITHUB ENTERPRISE

We recommend configuring all three (3) monitoring options, as each piece helps build the whole picture of system/application health.

- SNMP
- Collectd
- Log Forwarding

# SNMP

- SNMP is able to query for different statistics on the instance and with alert thresholds trigger an alert if those are exceeded.
- Note SNMP traps are not supported in GitHub Enterprise, so this information must be queried by the remote SNMP system.

# SNMP CPU

## Example

```
snmpget -v3 -u "public" github.example.com UCD-SNMP-MIB::laLoadFloat.3
```

Severity	Threshold
Warning	Fifteen minute load average exceeds 1x CPU cores
Critical	Fifteen minute load average exceeds 2x CPU cores

# SNMP MEMORY



## Example

```
snmpget -v3 -u "public" github.example.com UCD-SNMP-MIB::memAvailReal.0
```

Severity	Threshold
Warning	Sustained RSS usage exceeds 50% of total available memory
Critical	Sustained RSS usage exceeds 70% of total available memory

# SNMP STORAGE



## Example

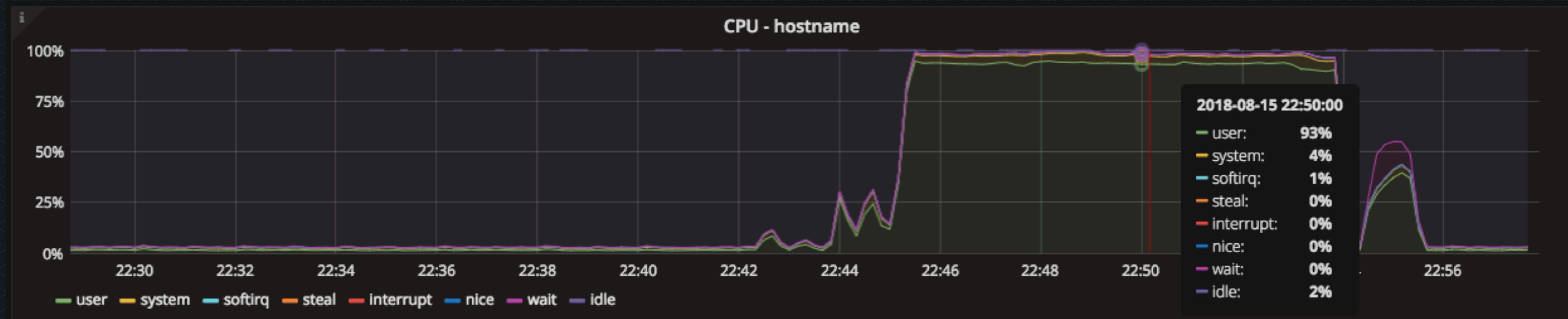
```
snmpget -v3 -u "public" github.example.com UCD-SNMP-MIB::dskPercent.2
```

Severity	Threshold
Warning	Disk use exceeds 70% of total available
Critical	Disk use exceeds 85% of total available

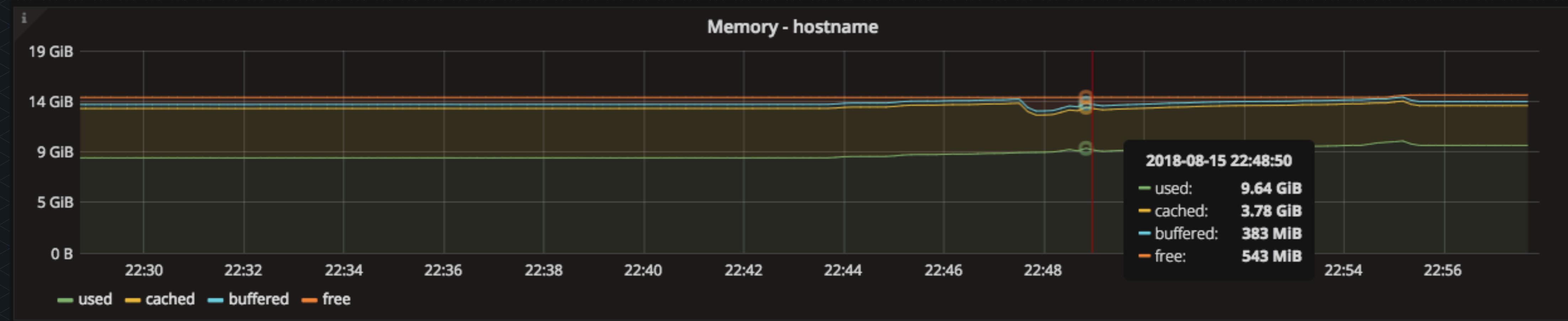
# COLLECTD

- Forwarding Collectd metrics allows you to see/parse the same metrics that are available on the /setup/monitor page of the Management Console.

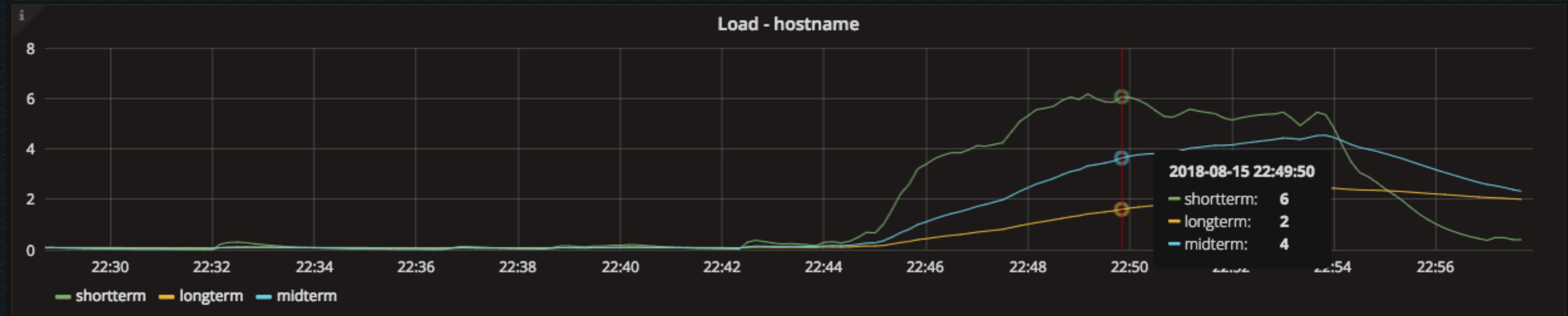
# COLLECTD CPU



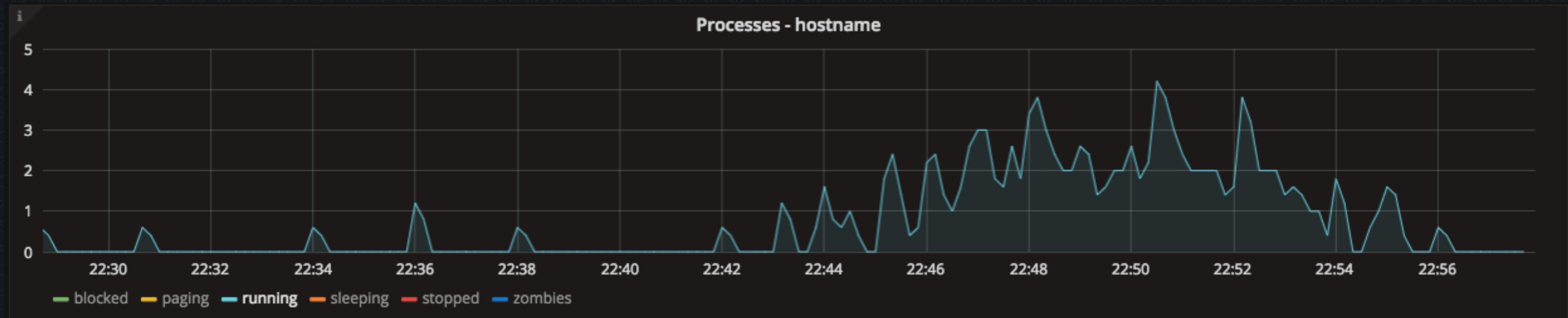
# COLLECTD MEMORY



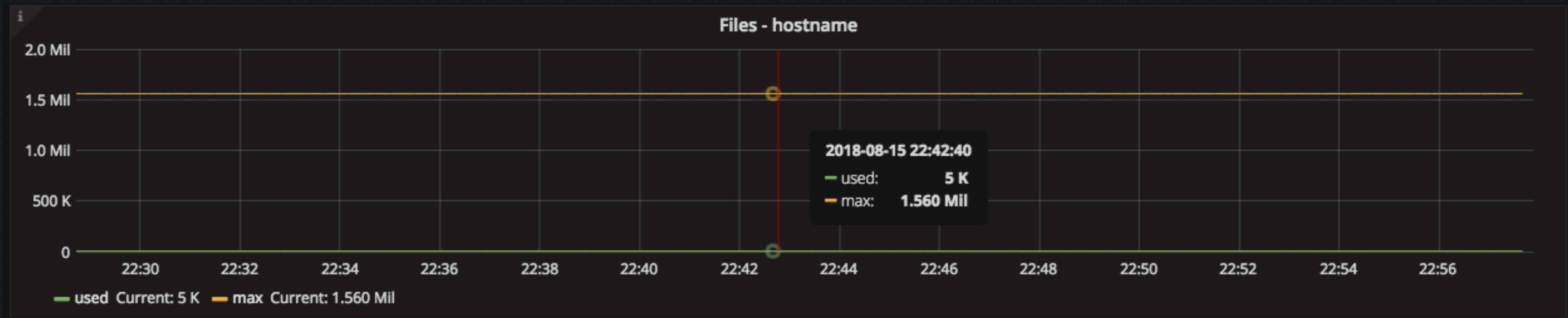
# COLLECTD LOAD



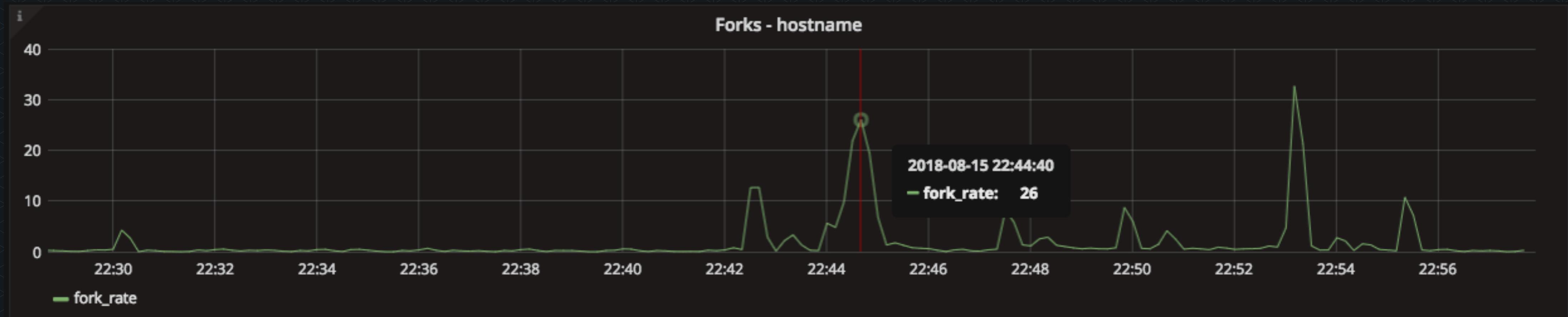
# COLLECTD PROCESSES



# COLLECTD FILES



# COLLECTD FORKS



# LOG FORWARDING



- Log forwarding is recommended as the instance only maintains ~7 days of logs on disk and once these have been rotated away they are "lost".
- Having a longer history available can help solve some issues that seemingly resolve themselves but reoccur later.
- Are those metrics for Enterprise admins only!?

# WHERE ARE THE LOG FILES?

- ssh -p 122 admin@github/example.com
- /var/log/\*
- /var/log/github/\*
- log forwarding --> Splunk, Logstash, Grafana, etc



# HOW MUCH DATA IS IN THE LOG FILES?

- 7 day data retention in syslog by default
- 4 GiB rolling buffer data retention in journald by default
- log forwarding depends on your own configuration

# /var/log/babeld/babeld.log

## Examples

- Top IP addresses

```
grep -o 'ip=[^ ]*' babeld.log | sort | uniq -c | sort -nr | head
```

- Top requested repositories

```
grep -o 'repo=[^ ]*' babeld.log | sort | uniq -c | sort -nr | head
```

- Top requested organizations or users

```
grep -o 'repo=[^/]*' babeld.log | sort | uniq -c | sort -nr | head
```

- Number of pushes in 10 minute intervals

```
grep 'cmd=git-receive-pack.*op done' babeld.log | cut -c 1-15 | uniq -c
```

- Number of fetches and clones in 10 minute intervals

```
grep 'cmd=git-upload-pack.*op done' babeld.log | cut -c 1-15 | uniq -c
```

# /var/log/auth.log

## Examples

- Users failing authentication

```
grep -F 'at=failure' auth.log | grep -o ' login=[^ ]*' | sort | uniq -c | sort -nr | head
```

- Authentications failing because of invalid LDAP usernames

```
grep 'Invalid ldap username' auth.log | grep -o ' login=[^ ]*' | sort | uniq -c | sort -nr | head
```

- /var/log/github/unicorn.log

```
grep 'path_info="/api/v3' unicorn.log | grep -v 'api/v3/internal' | grep -v 'auth=oauth' | grep -o 'auth_fingerprint=[^ ]*' | sort | uniq -c | sort -rn
```

# GITHUB API

- Metrics for everyone



# COMMIT ACTIVITY BY TEAM & BRANCH

```
query getCommitActivityByTeam($org: String!, $ref: String!) {  
  organization(login:$org){  
    teams(first: 10){  
      nodes {  
        name  
        repositories(first:10){  
          nodes{  
            name  
            # To get a list of valid qualifiedNames, run branches-by-repository query  
            ref(qualifiedName:$ref){  
              target{  
                ... on Commit{  
                  history(first:100){  
                    nodes{  
                      messageHeadline  
                      committedDate  
                      author{  
                        name  
                        email  
                      }  
                    }  
                    pageInfo { # For pagination through history  
                      hasNextPage  
                      endCursor  
                    }  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}  
  
variables {  
  "org": "orgName"  
  "ref": "not-master"  
}
```

# OPEN PULL REQUESTS

```
query getOpenPullRequests($org: String!) {  
  search(first: 25, type: ISSUE, query: "org:$org type:pr state:open sort:created-asc") {  
    nodes {  
      ... on PullRequest {  
        id  
        number  
        title  
        state  
        createdAt  
        commits(last: 1) {  
          nodes {  
            commit {  
              lastCommittedDate: committedDate  
            }  
          }  
        }  
        comments(last: 1) {  
          nodes {  
            lastCommentDate: createdAt  
          }  
        }  
      }  
    pageInfo {  
      hasNextPage  
      endCursor  
    }  
  }  
}
```

# PULL REQUEST REVIEWS

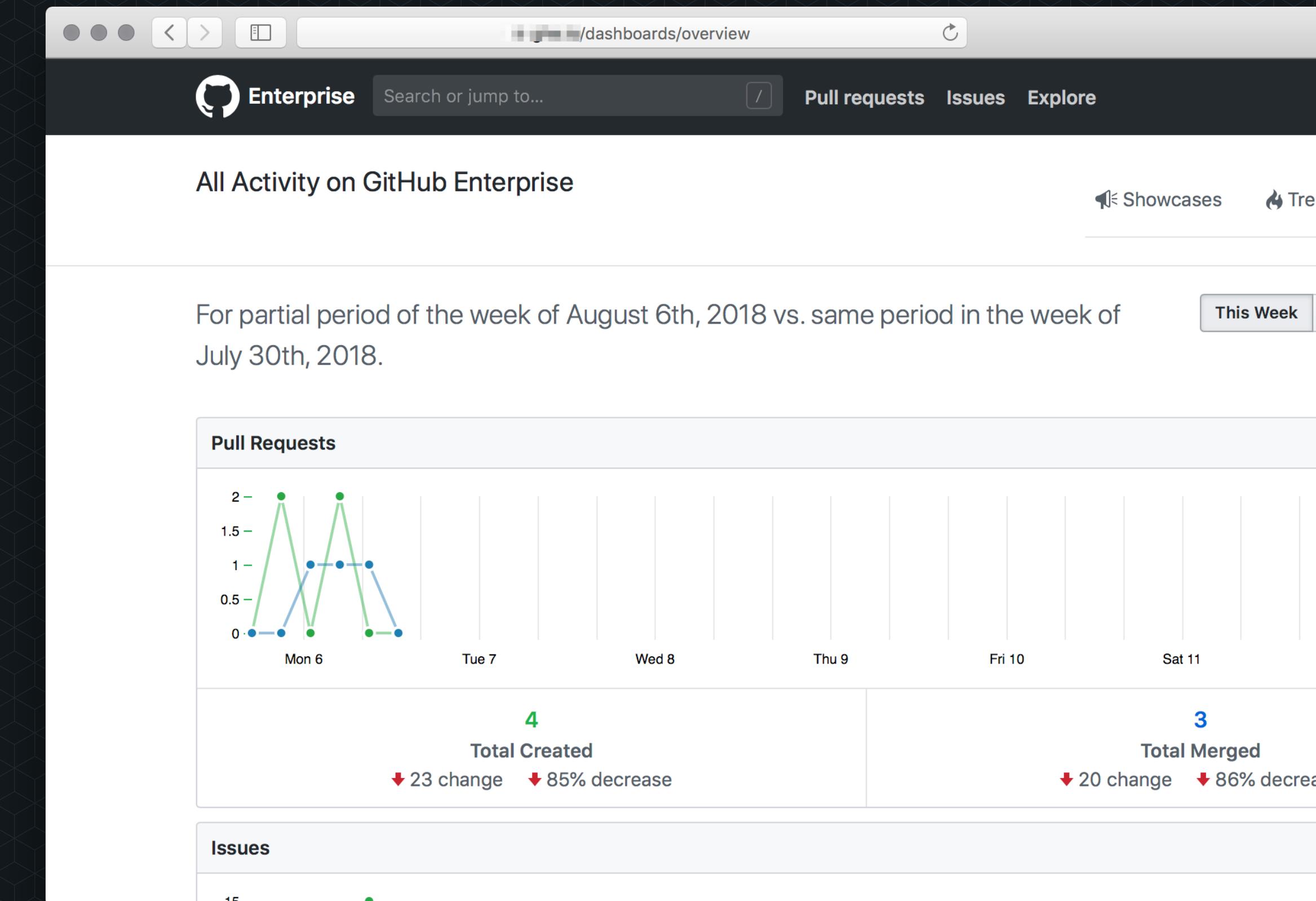
```
{  
  search(first: 25, type: ISSUE, query: "org:org-name type:pr state:open sort:created-asc") {  
    nodes {  
      ... on PullRequest {  
        id  
        number  
        title  
        state  
        createdAt  
        reviews(first:100) {  
          totalCount  
          nodes {  
            author {  
              login  
            }  
            body  
            createdAt  
            state  
          }  
        }  
      }  
    }  
    pageInfo {  
      hasNextPage  
      endCursor  
    }  
  }  
}
```



# DASHBOARD EXAMPLES

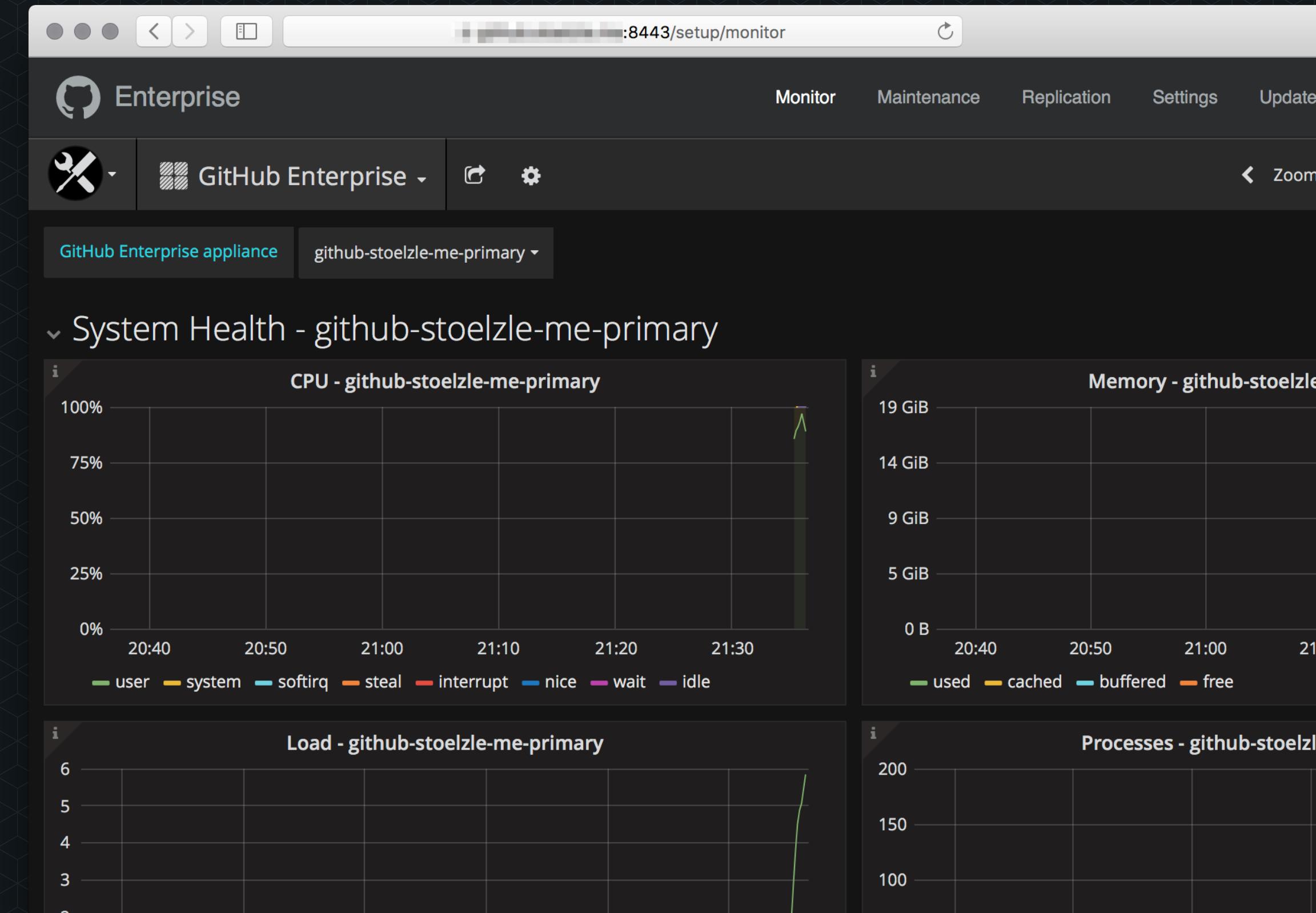
# GITHUB ENTERPRISE ACTIVITY DASHBOARD

github.example.com/dashboards/overview



# GITHUB ENTERPRISE ADMIN MONITORING

github.example.com:8443/setup/monitor



# HYGIEIA

[github.com/capitalone/Hygieia](https://github.com/capitalone/Hygieia)

capitalone / Hygieia

Code Issues Pull requests Projects Wiki Insights

CapitalOne DevOps Dashboard <http://capitalone.github.io/Hygieia>

devops dashboard hygieia delivery-pipeline visualization continuous-delivery continuous-deployment continuous-integration

2,179 commits 8 branches 9 releases 100 contributors Apache-2.0

Branch: master New pull request Find file Clone or download

nireeshT and matthewbeasley Sonar collector - null check for project data (#2436) Latest commit 31e3df4 2 days ago

.mvn/wrapper added maven wrapper 3 years ago

Ui-protractor-tests Update README (#2095) 12 days ago

Ui-tests Circumvent docker-maven-plugin (#2039) 6 months ago

UI Team widget (#2398) a month ago

api-audit PT audit changes - Generic collector for Performance tests 3 days ago

api PT audit changes - Generic collector for Performance tests 3 days ago

certs Centralized enterprise certificate management + addition in Bitbucket... 3 months ago

collectors Sonar collector - null check for project data (#2436) 2 days ago

core PT audit changes - Generic collector for Performance tests 3 days ago

db Docker compose refactor (#1972) 8 months ago

hygieia-jenkins-plugin Changed the datatype of 'CodeQualityMetric' 'value' variable to String... 2 months ago

media Jenkins 2.0 support in Hygieia Jenkins plugin (#1055) 2 years ago

test-servers Redirecting README files to gh-pages branch (#1624) a year ago

.gitignore Centralized enterprise certificate management + addition in Bitbucket... 3 months ago

.travis.yml Travis build notifications modified (#2256) 4 months ago

CHANGELOG.md Introduce CHANGELOG to keep track of changes (#1807) 10 months ago

Dockerfile Circumvent docker-maven-plugin (#2039) 6 months ago

Hygieia2.md Redirecting README files to gh-pages branch (#1624) a year ago

# HYGIEIA

[capitalone.github.io/Hygieia/screenshots.html](https://capitalone.github.io/Hygieia/screenshots.html)



# HUBBLE

[github.com/autodesk/hubble](https://github.com/autodesk/hubble)

Autodesk / hubble

Code Issues Pull requests Projects Insights

Collaboration, usage, and health data visualization for GitHub Enterprise <https://autodesk.github.io/hubble>

hubble-enterprise github-enterprise analytics github git

251 commits 7 branches 4 releases 10 contributors MIT

Branch: master New pull request Find file Clone or download

pluehne Generate more demo data for Git LFS traffic chart Latest commit f542611 on May 28

docs Generate more demo data for Git LFS traffic chart 4 months ago

updater Track Git LFS traffic 4 months ago

.codecov.yml report JavaScript code coverage in PRs (#84) 8 months ago

.gitignore Made .gitignores more consistent. 11 months ago

.travis.yml report JavaScript code coverage in PRs (#84) 8 months ago

CHANGELOG.md Version bump for release 0.3.0 5 months ago

CONTRIBUTING.md add JS testing framework and "vendor" libraries (#61) 9 months ago

LICENSE.md Switched to MIT license. (#4) 11 months ago

README.md Rephrase readme to make it more concise (#117) 7 months ago

logo-banner.svg Added logo banner to readme. 11 months ago

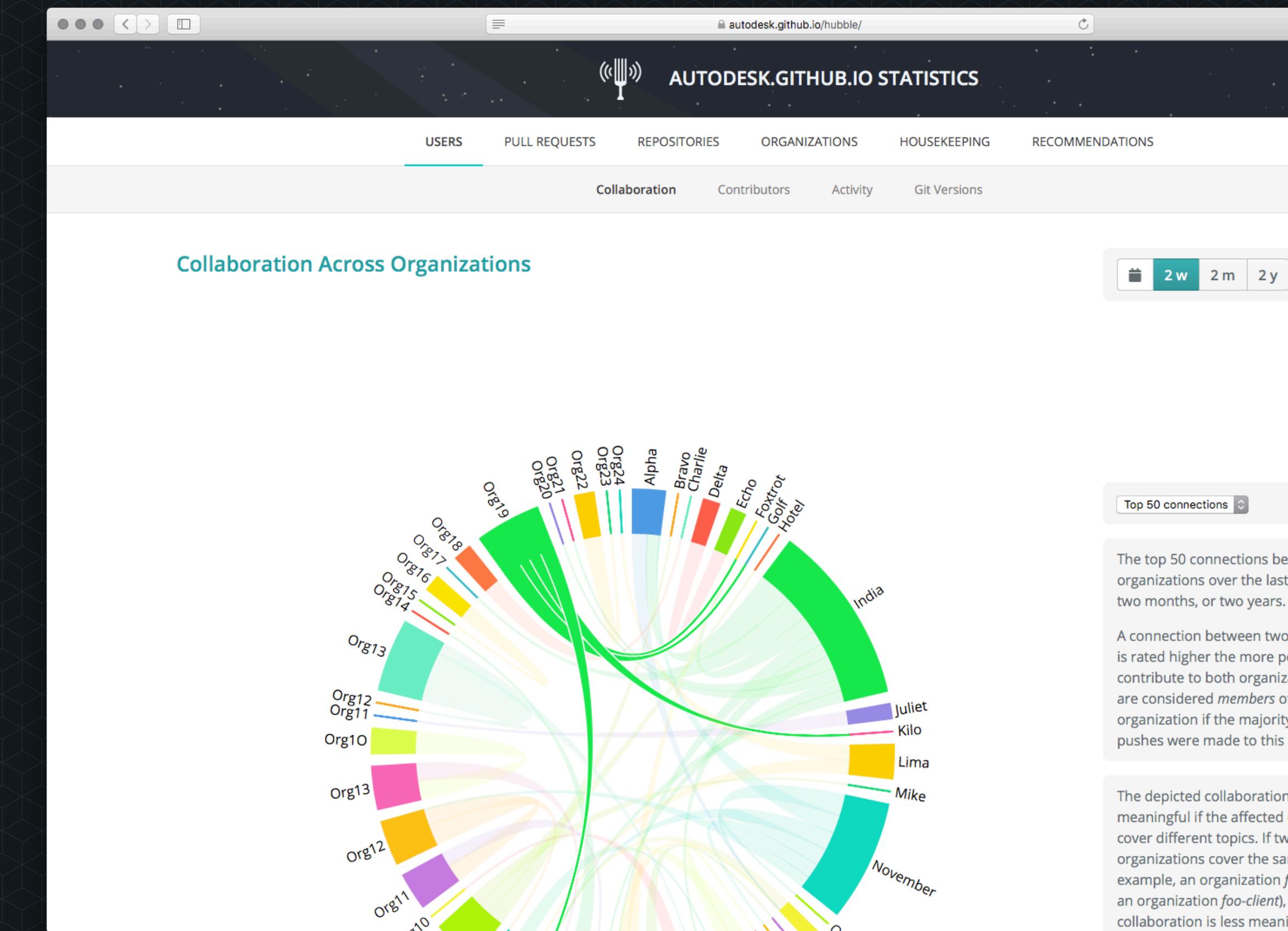
README.md

**hubble**  
--enterprise

> visualizes collaboration,  
> usage, and health data for  
> GitHub Enterprise

# HUBBLE

[autodesk.github.io/hubble](https://autodesk.github.io/hubble)





# Q&A PANEL



# THANK YOU



U N I V E R S E