

# Dealing with the Pesky Path Parameter Problem



# Alex Leong

Software Engineer @ Buoyant

 @adlleong

 @adleong

 @alex







# Prometheus Overview



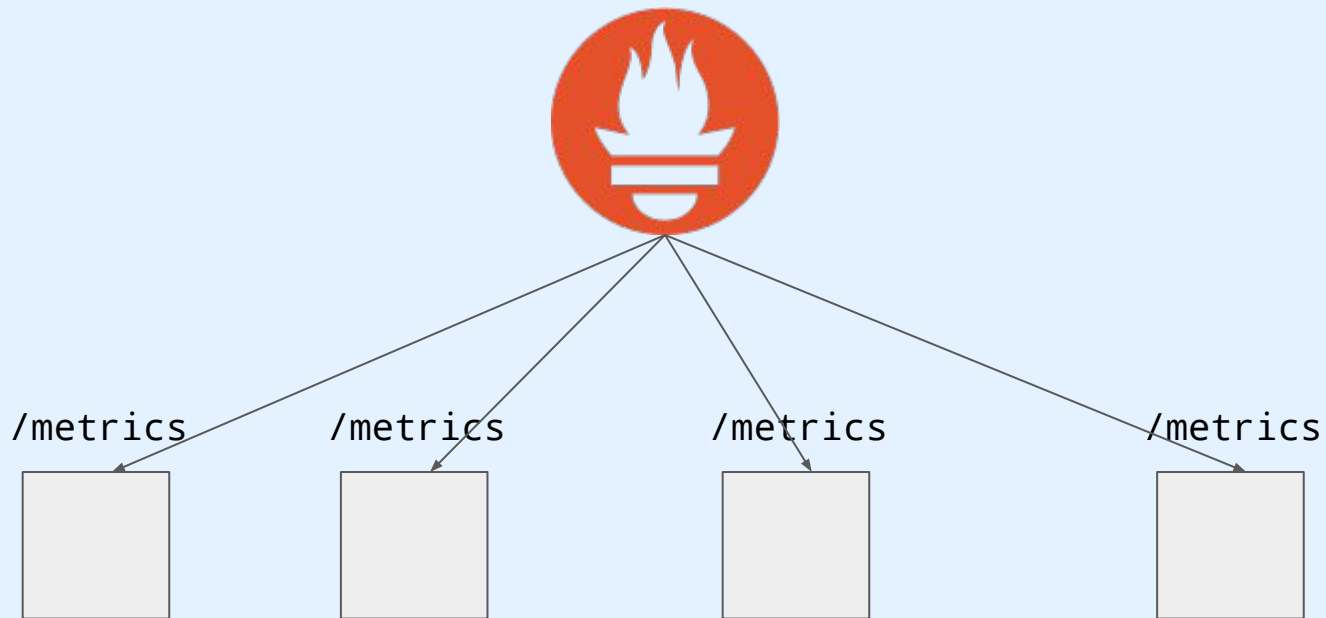
# Prometheus

- Open-Source
- CNCF Graduated Project
- Time-Series Database
- Useful for monitoring service metrics

# /metrics

request\_total 17

# Prometheus



# /metrics

request\_total{method=GET} 10

request\_total{method=POST} 7



# /metrics

request\_total{method=GET, status=200} 5

request\_total{method=POST, status=200} 3

request\_total{method=GET, status=404} 5

request\_total{method=POST, status=404} 4

# /metrics

$$|T| = \prod_{L_i} |L_i|$$

# of timeseries = multiply the # of values  
for each label

# Prometheus

- Can handle “millions” of time series
- e.g 6 labels with 10 values each
- Avoid adding too many labels
- Avoid “high cardinality” labels



UNBOUNDED  
CARDINALITY

# Unbounded Cardinality

- What happens when we add a “path” label?

```
request_total{path=/foo} 5231
```

```
request_total{path=/bar} 197
```

```
request_total{path=/bas} 2682
```

```
request_total{path=/butts} 1
```

# Unbounded Cardinality

- Unsanitized user input
- Never allow user input in labels
- Use whitelist

# /metrics

request\_total{path=/books/51.json} 5

request\_total{path=/books/52.json} 1

request\_total{path=/books/53.json} 1

request\_total{path=/books/54.json} 2

request\_total{path=/books/55.json} 1

request\_total{path=/books/56.json} 1

request\_total{path=/books/57.json} 3







# Path Metrics in Linkerd





An open source *service mesh* and CNCF member project.

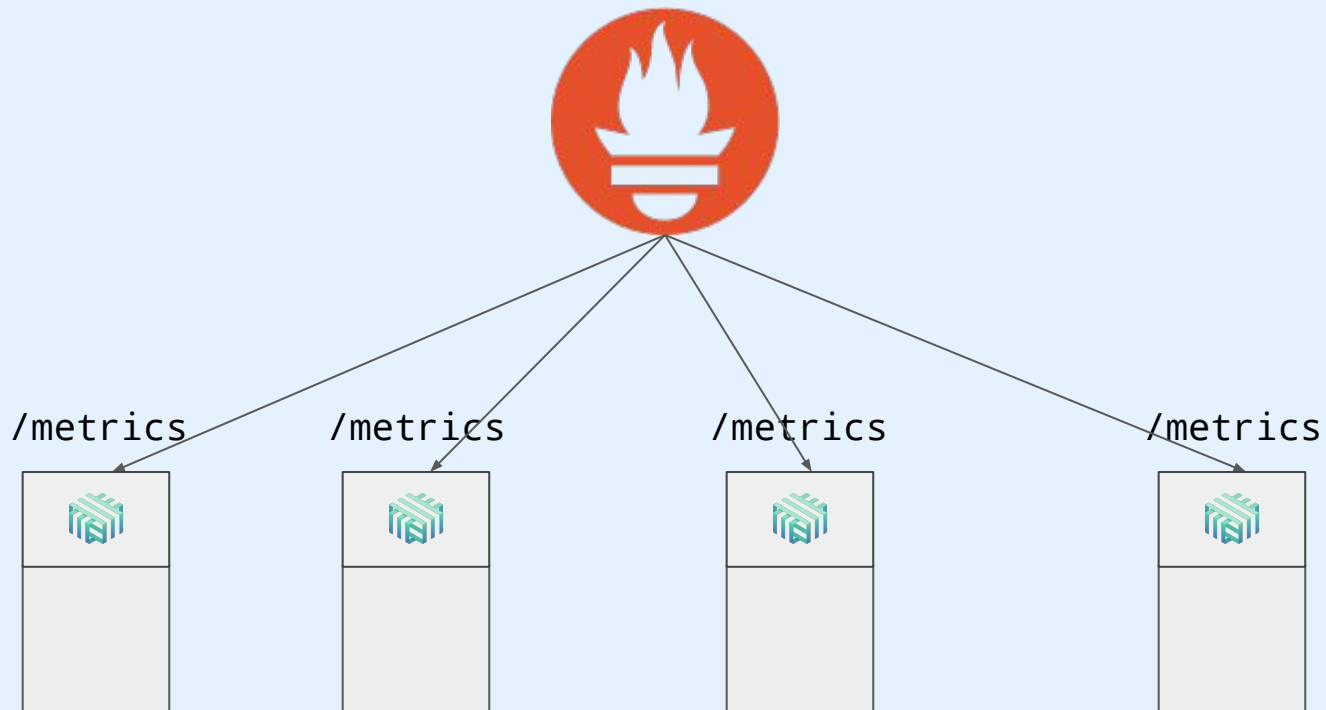
-  24+ months in production
-  3,000+ Slack channel members
-  10,000+ GitHub stars
-  100+ contributors



# Linkerd

- Sidecar proxy
- Collects rich traffic metrics
  - Request metadata, success/failure, latency
- Exposes `/metrics` endpoint

# Prometheus + Linkerd



# Prometheus + Linkerd

- The pesky path parameter problem
- Consolidate paths somehow?
- Heuristics based on path data?
- Machine learning???



# Service Profiles

# Service Profiles

- A Custom Resource Definition (CRD)
- Stores metadata about a service
- List of “routes”

# Service Profiles

- Route: a logical grouping of paths, defined by a regular expression

name: "GET /books/{id}.json"

method: GET

pathRegex: /books/\d+\.json

# Service Profiles

routes:

- name: DELETE  
/books/{id}.json  
condition:  
method: DELETE  
pathRegex:  
/books/\d+\.json
- name: GET /books.json  
condition:  
method: GET  
pathRegex: /books\.json
- name: GET /books/{id}.json  
condition:  
method: GET  
pathRegex: /books/\d+\.json
- name: POST /books.json  
condition:  
method: POST  
pathRegex: /books\.json



# Service Profiles

- Routes:
  - Are enumerable
  - User defined
  - Define collections of paths

# Service Profiles

- Service profiles also define:
  - Retryability
  - Timeouts
  - Traffic splits
  - etc.



# Conclusions

# Conclusions

- Prometheus is AWESOME!
- Be aware of label cardinality
- Don't accept unsanitized user input in labels
- Group high cardinality labels into a fixed number of buckets



Join our community!



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



[slack.linkerd.io](https://slack.linkerd.io)



[@linkerd](https://twitter.com/linkerd)

FROM YOUR FRIENDS AT



**BUOYANT**