



Deploying a real service to Dokku and monitoring it with Prometheus/Grafana

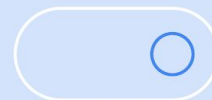
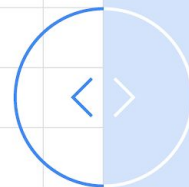


Lucy Linder
GDG Fribourg
lucy@gdgfribourg.ch

20th Fribourg Linux Seminar, March 4, 2021

Google Developers

<http://bit.ly/fribourg-linux-seminar-dokku-plus-monitoring>

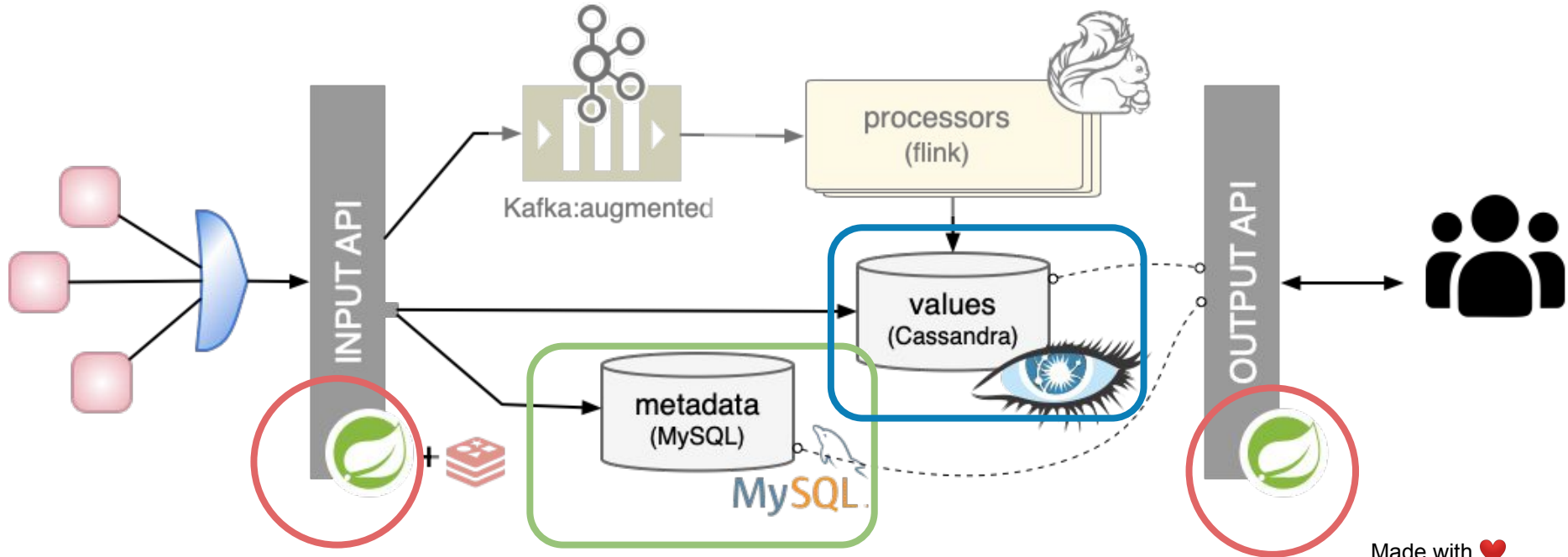


1

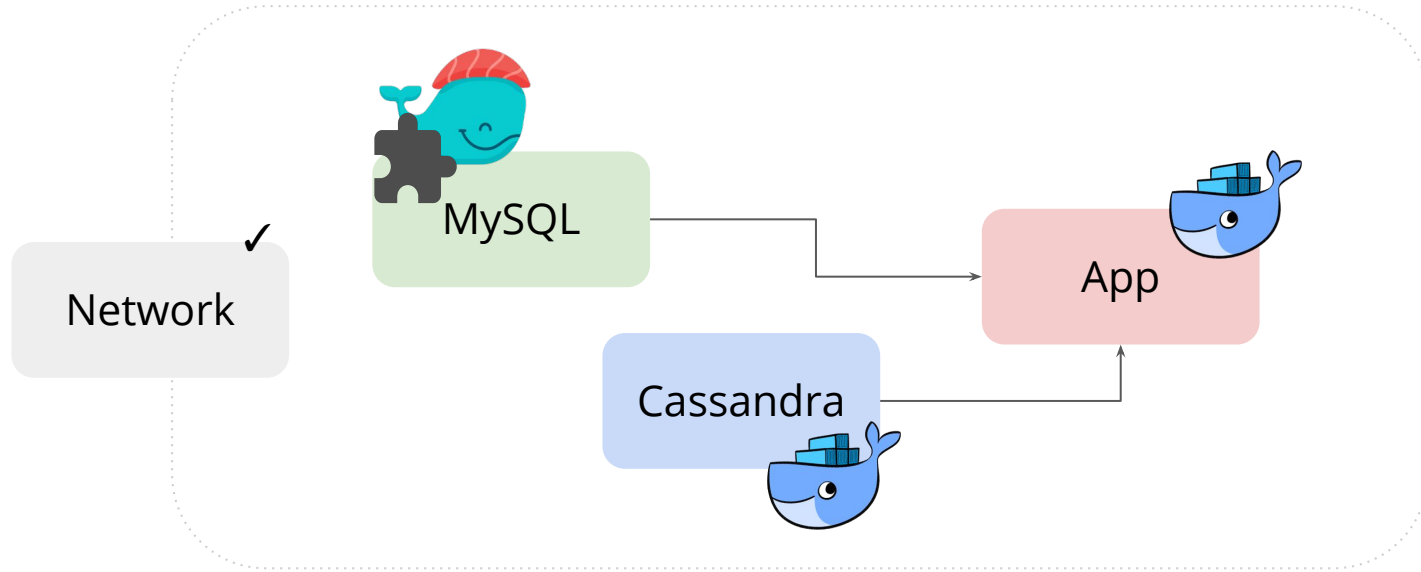


Deploying a real application
to Dokku

BBData API



Dokku install plan



MySQL ↪ Dokku Plugin

`dokku:mysql` supports upgrade, backup (AWS, etc.), and much more

```
# Install plugin
sudo dokku plugin:install https://github.com/dokku/dokku-mysql.git mysql

# Create some db
dokku mysql:create mydb --image-version 8.0.14 \
    --password "strong" --root-password "even-stronger"

# Initialise db with scripts
cat db-structure.sql | dokku mysql:import mydb
cat db-data.sql | dokku mysql:import mydb

# Link database to an app
dokku mysql:link mydb myapp -a "MY_DATABASE"
```

In Container app
MY_DATABASE_URL="mysql://[...]"

Cassandra / BBData ↦ Dockerfile

dokku

```
# Create app and attach to network
dokku apps:create $APP
dokku network:set $APP attach-post-create $NETWORK

# Mount volumes
dokku storage:mount $APP \
  /var/lib/dokku/data/storage/$APP:/container/path

# Set config
dokku config:set $APP ENV_X=value ENV_Y='${value}'
```

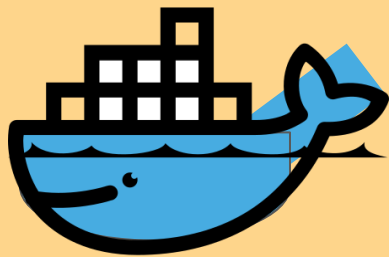
local

```
# In a repository with a Dockerfile
git remote add dokku "dokku@$DOKKU_HOST:$APP"
git push dokku master:master
```

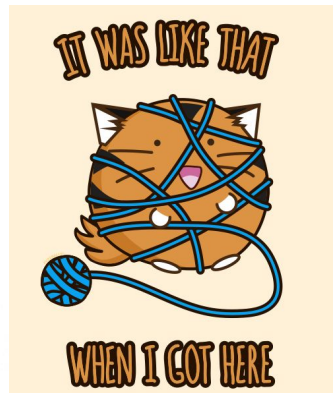


IS IT UP ?

So... easy ?



2



Monitoring with

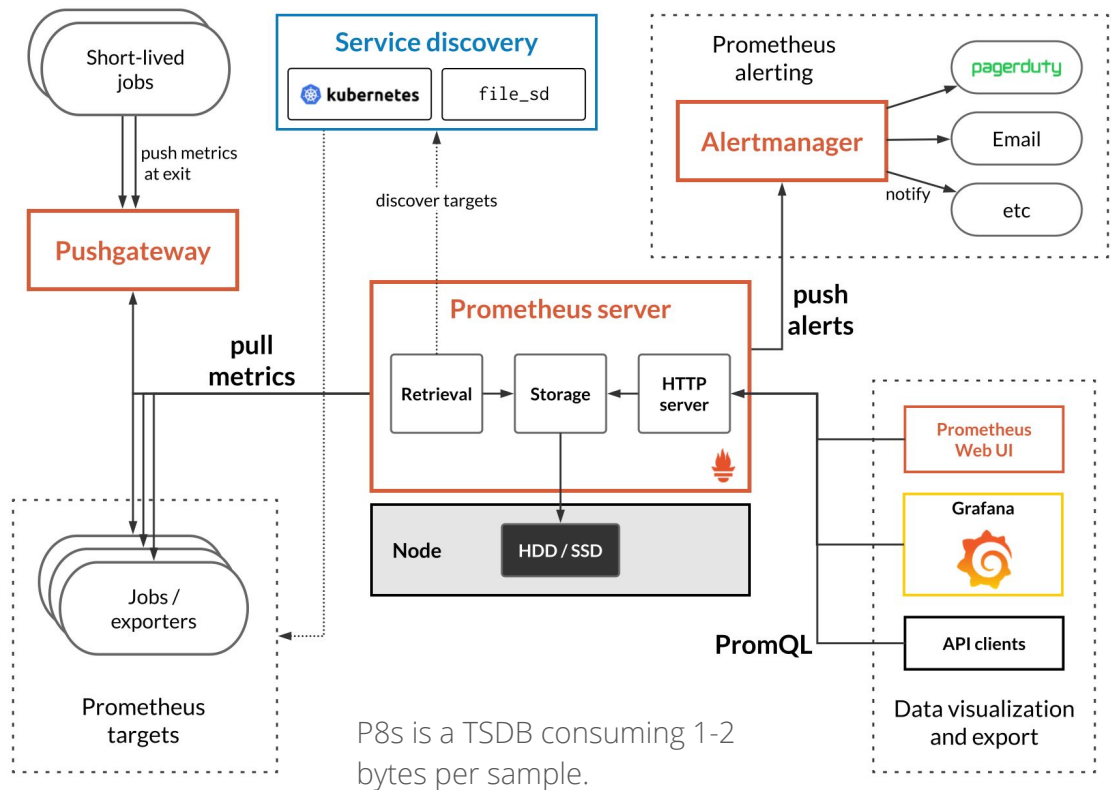


Prometheus +

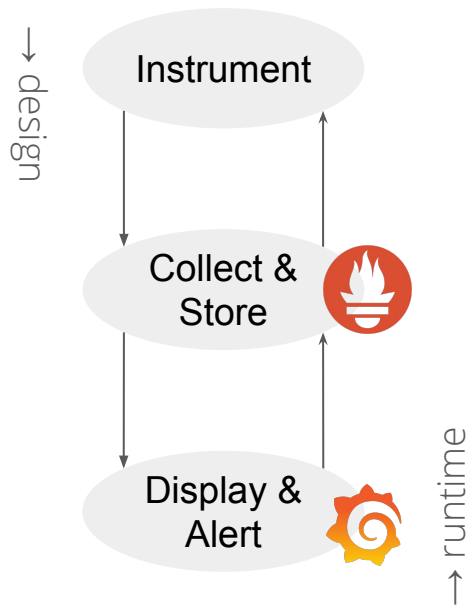


Grafana

Prometheus ecosystem



Simpler than it looks



Setup BBData Monitoring

Targets

- ✓ BBData: SpringBoot
- ✓ Cassandra: JMX Exporter

New images

- prom/prometheus:v2.25.0
- grafana/grafana:7.4.2

```
# config mounted in /etc/prometheus/prometheus.yml

global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['127.0.0.1:9090']

  - job_name: 'bbdata-api'
    metrics_path: '/prometheus'
    static_configs:
      - targets: ['bbdata:8111']

  - job_name: 'cassandra'
    static_configs:
      - targets: ['cassandra:7070']
```

Prometheus metrics

metric = timestamp + name + value + dimension(s)

```
2020-04-03T20:00:00Z
```

```
http_requests_total{code=200, instance=api-1,path=/foo} 1001
```

```
http_requests_total{code=404, instance=api-1,path=/bar} 1
```

```
http_requests_total{code=200, instance=api-2,path=/foo} 983
```

```
http_requests_total{code=404, instance=api-2,path=/bar} 10
```

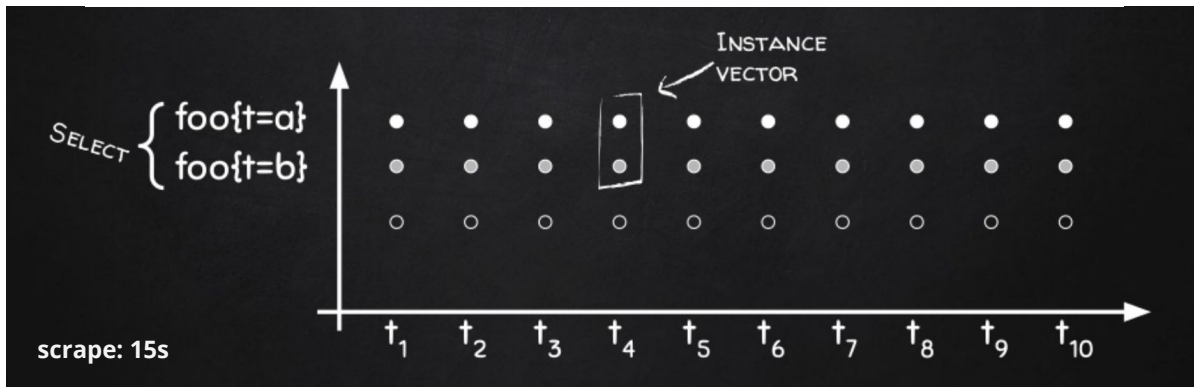
a gauge goes up and down

a counter goes up and only care about *deltas*

PromQL: instant vector & instant query

Instant vectors = data @ time, queries with optional aggregation over labels

GET /api/v1/query?time= t_4 &query=foo



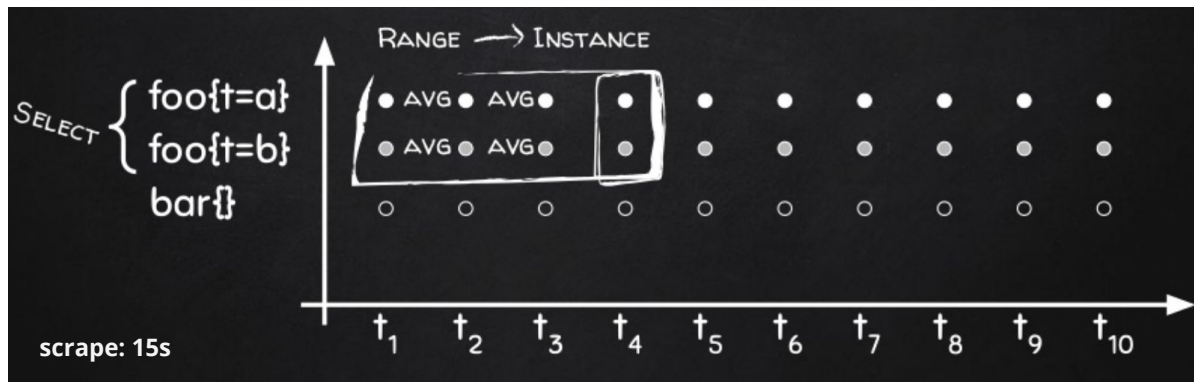
aggregate

GET /api/v1/query?time= t_4 &query=sum(foo)

PromQL: range vector & instant query

Range vectors = data @ [time interval]

GET /api/v1/query?time= t_4 &query=foo[1m]



to instant

GET /api/v1/query?time= t_4 &query=avg_over_time(foo[1m])

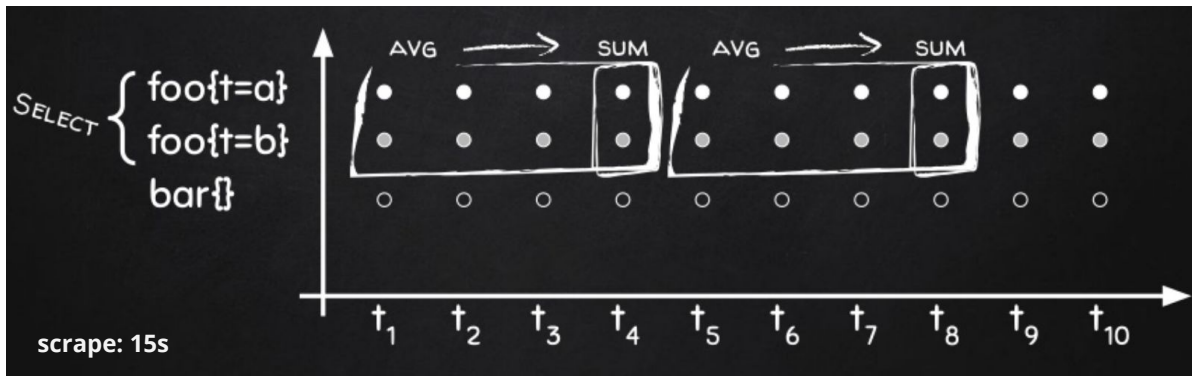
GET /api/v1/query?time= t_4 &query=sum(avg_over_time(foo[1m]))

PromQL: range query

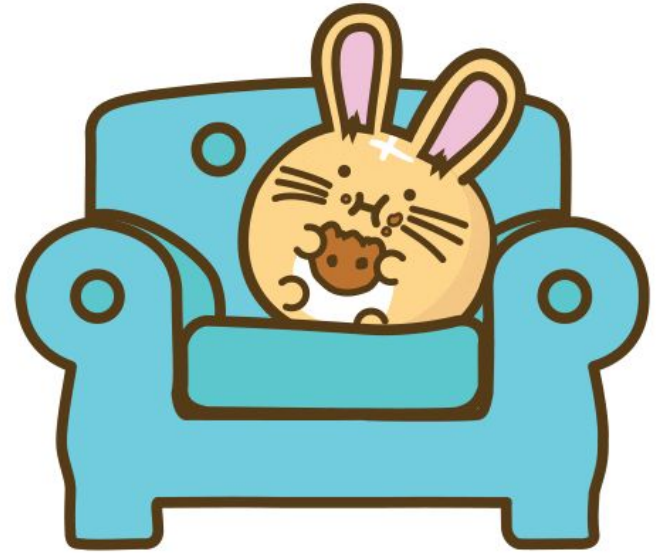
Range queries = repeat instant query at every step between [time interval]

```
GET /api/v1/query_range?start=t1&end=t8&step=60  
&query=sum(avg_over_time(foo[1m]))
```

rinse and
repeat



LIVE DEMO



(Re)sources

BBData API: <https://github.com/big-building-data/bbdata-api>

Prometheus & Monitoring series by Zaar Hai:

1. (YouTube) [Dip into Prometheus](#) [slides]
2. (YouTube) [Deep into Prometheus](#) [slides]
3. (Medium) [Making peace with Prometheus rate](#)

Fuzzballs illustrations: <https://fuzzballs.co/>



QUESTIONS ?



Slides: <http://bit.ly/fribourg-linux-seminar-dokku-plus-monitoring>

Code: <https://github.com/derlin/20th-linux-seminar-dokku-monitoring>