Django + prometheus

Amit Saha

About me

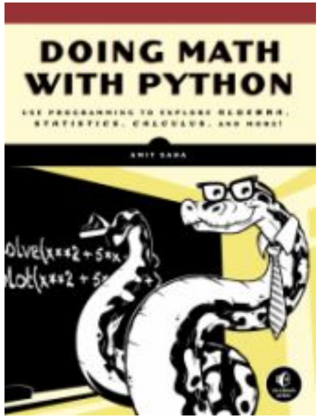
DevOps at Cover Genius

COVER GENIUS

SMARTER INSURANCE

Occasional Speaker

Author of 2 books, various articles



Agenda

Relevant to you - if your organization is using prometheus for monitoring

Django + prometheus issues (with source walkthrough)

Django + statsd + prometheus (with source walkthrough)

Monitoring 101

Why should I monitor?

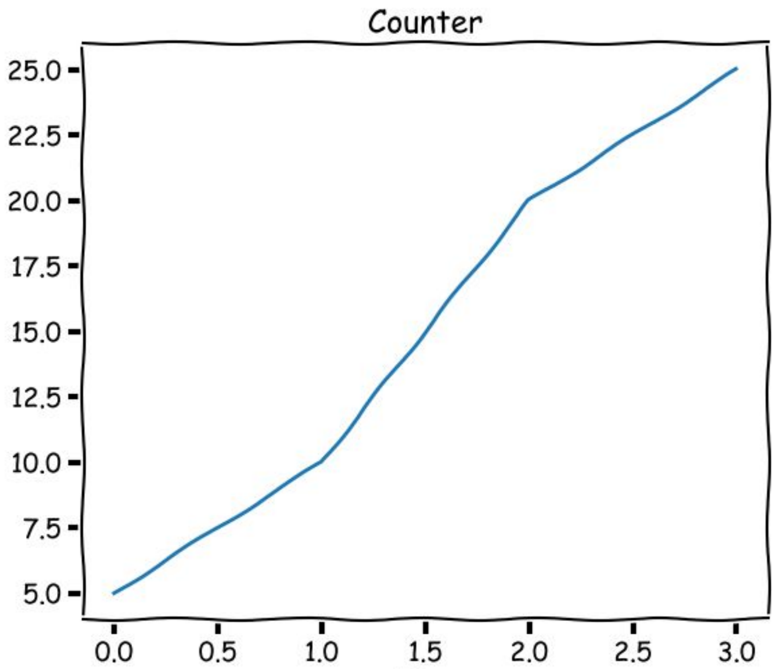
Capacity planning, autoscaling, hardware configuration, performance troubleshooting

Your business needs to stay running

Understand system/application behavior

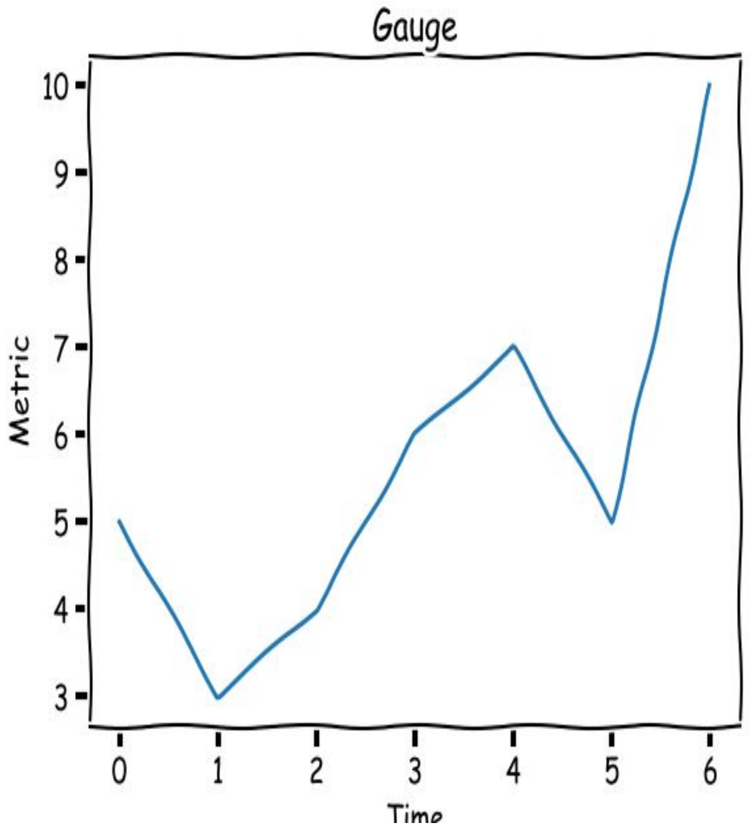
Counter

A metric whose value increases during the lifetime of a process/system



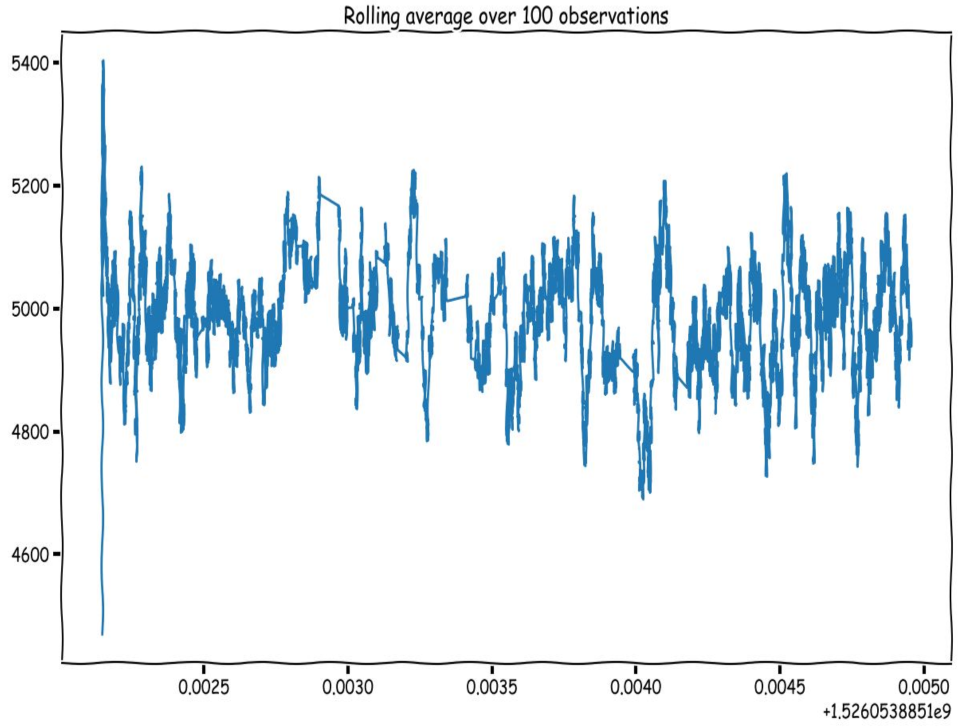
Gauge

A metric whose value can go up or down arbitrarily - usually with a floor and ceiling



Histogram/Timer

A metric to track *observations*



Monitoring systems

Collection, aggregation, storage and querying of metrics

One software or a set of software makes up a monitoring system

statsd, Prometheus, various hosted products

Monitoring a Django Application

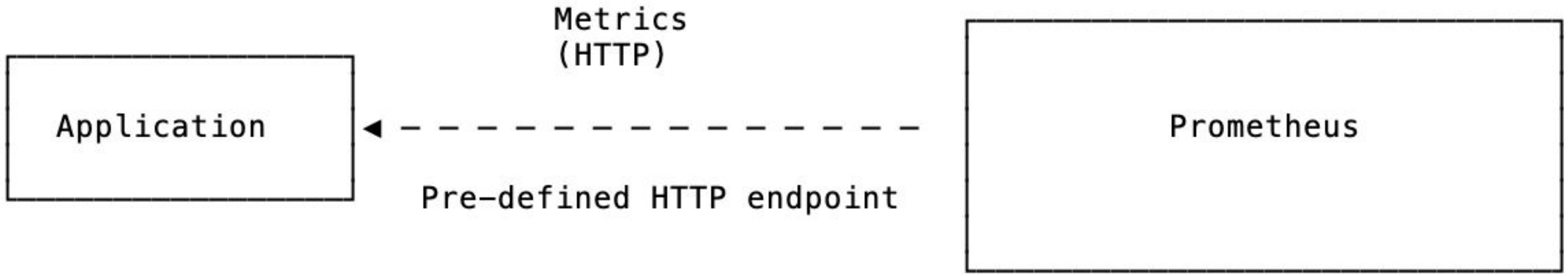
Application Assumptions

WSGI, deployed via uwsgi/gunicorn

Worker process model

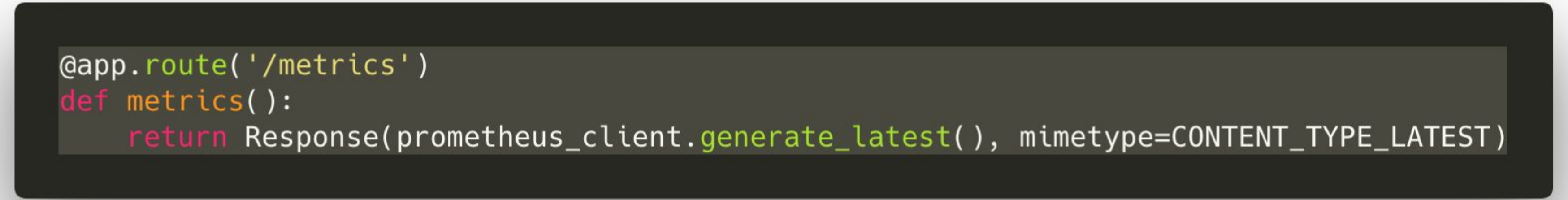
Monitoring with prometheus client
(Source reference: Demo 1)

django application <- Prometheus



django application <- Prometheus

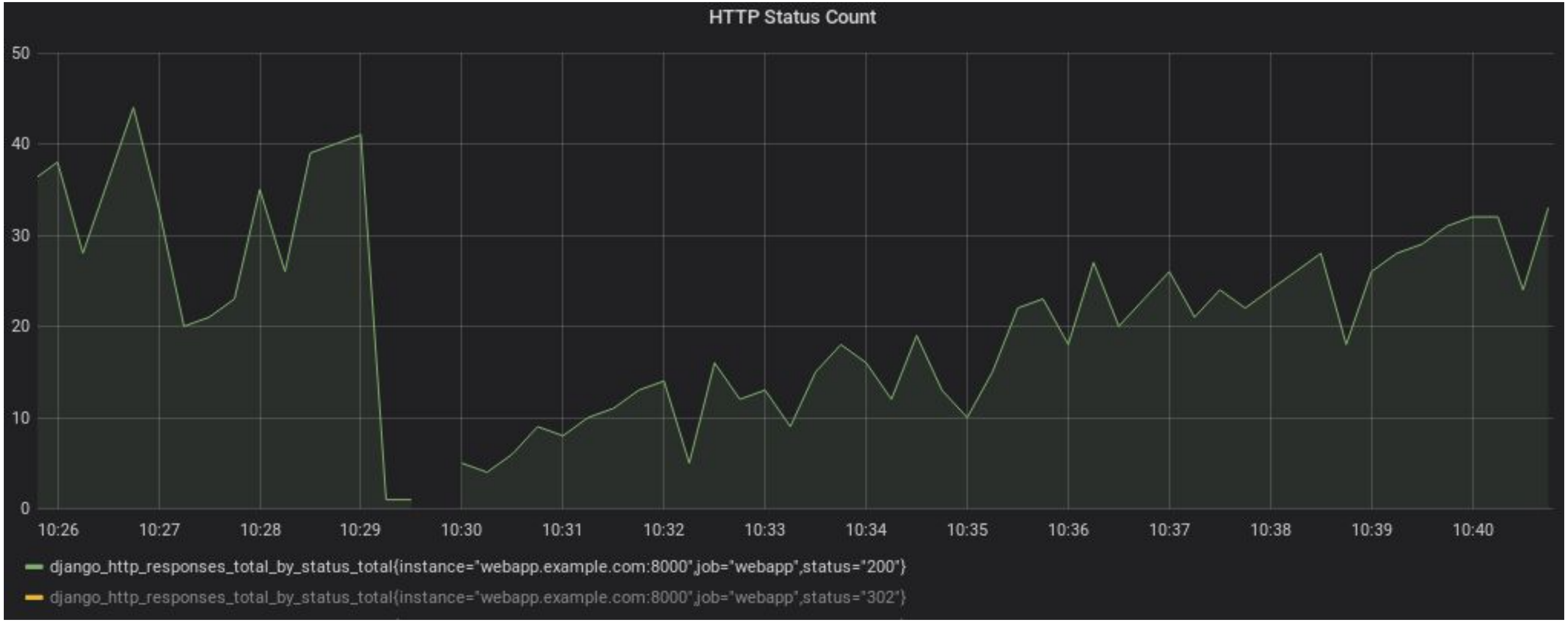
Application exposes a HTTP endpoint for prometheus to scrape - usually, /metrics



```
@app.route('/metrics')
def metrics():
    return Response(prometheus_client.generate_latest(), mimetype=CONTENT_TYPE_LATEST)
```

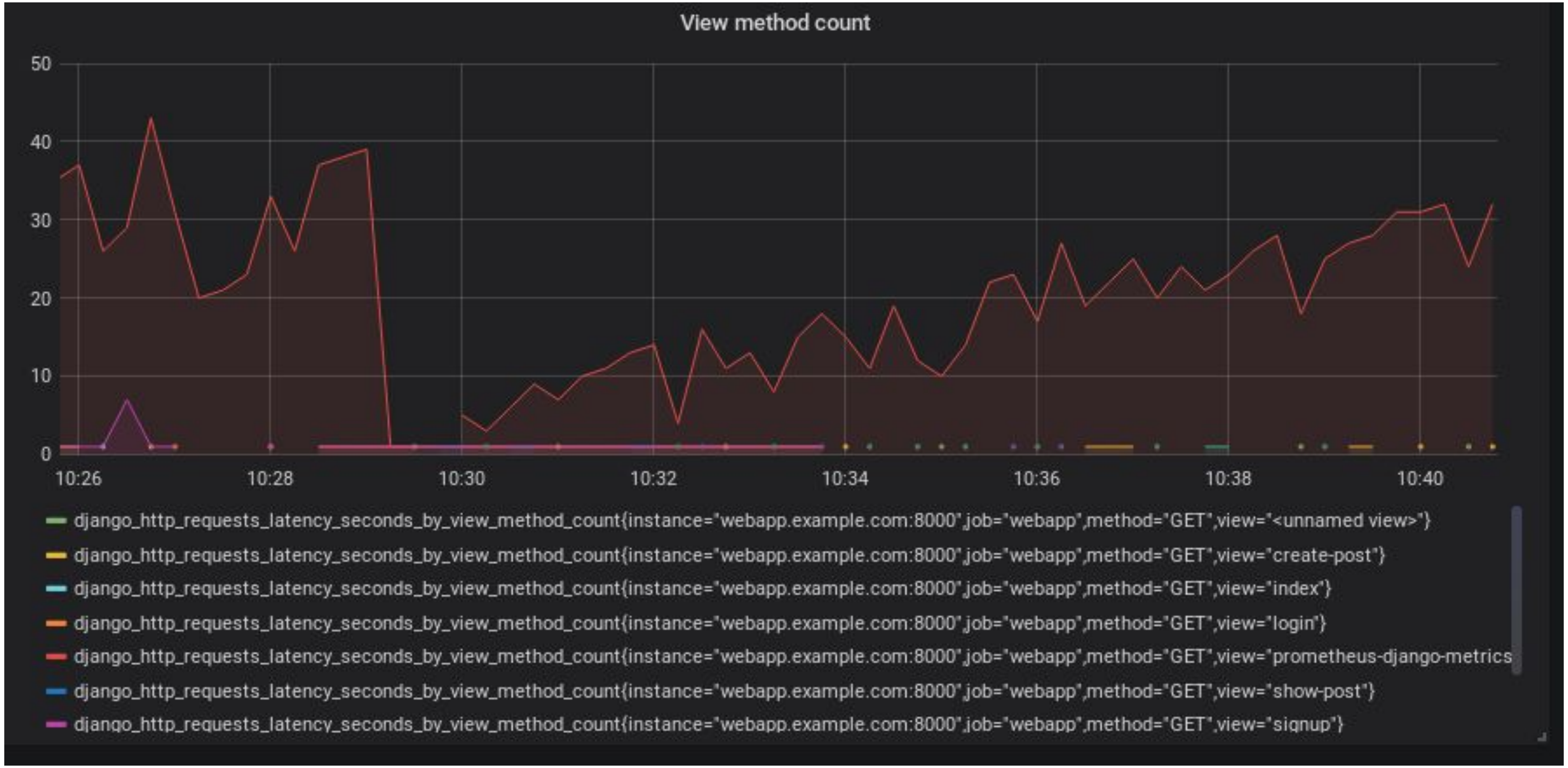
Time passes….

We have a problem, however



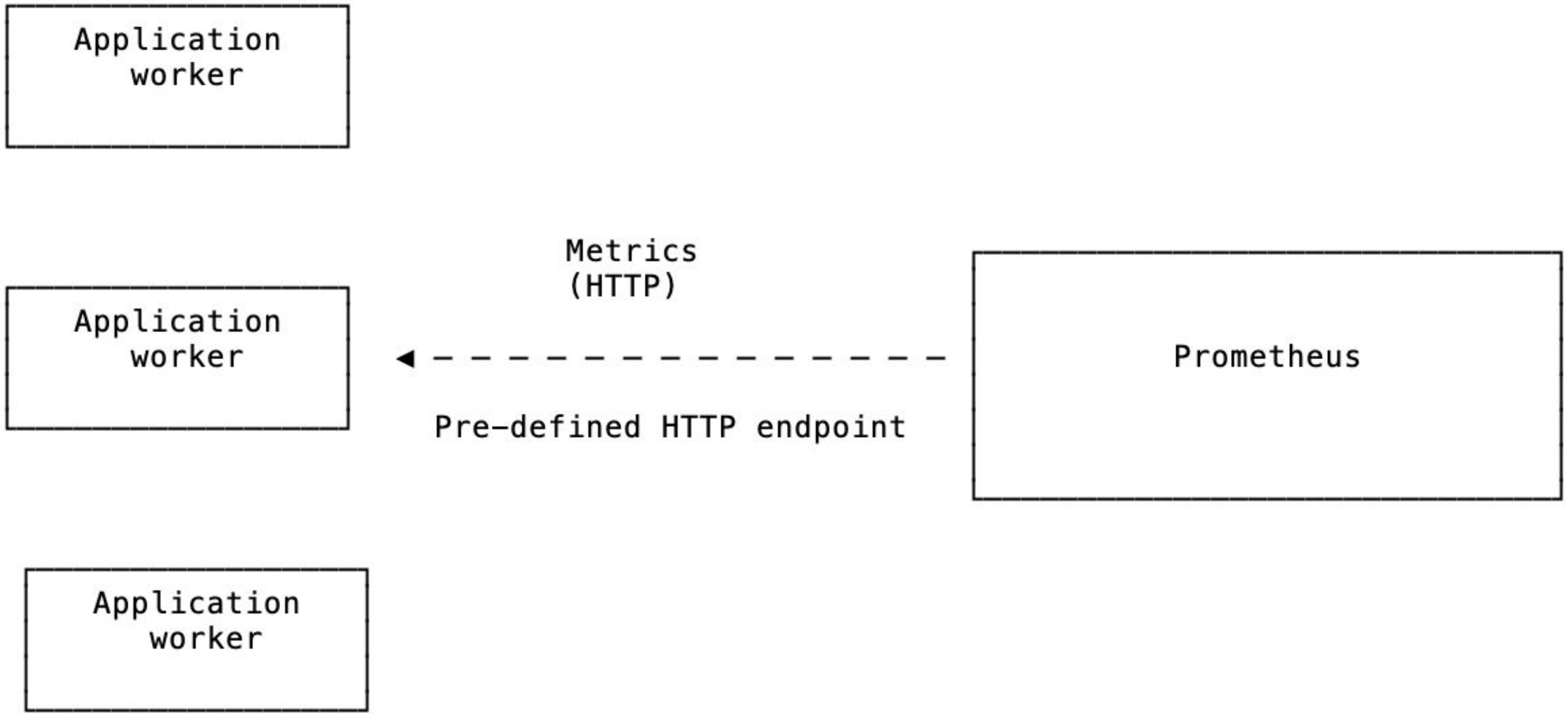
django_http_responses_total_by_status_total{instance="webapp.example.com:8000",job="webapp",status="200"}
django_http_responses_total_by_status_total{instance="webapp.example.com:8000",job="webapp",status="302"}

We have a problem, however



🟩 django_http_requests_latency_seconds_by_view_method_count{instance="webapp.example.com:8000",job="webapp",method="GET",view="<unnamed view>"}
🟨 django_http_requests_latency_seconds_by_view_method_count{instance="webapp.example.com:8000",job="webapp",method="GET",view="create-post"}
🟦 django_http_requests_latency_seconds_by_view_method_count{instance="webapp.example.com:8000",job="webapp",method="GET",view="index"}
🟧 django_http_requests_latency_seconds_by_view_method_count{instance="webapp.example.com:8000",job="webapp",method="GET",view="login"}
🟥 django_http_requests_latency_seconds_by_view_method_count{instance="webapp.example.com:8000",job="webapp",method="GET",view="prometheus-django-metrics
🟦 django_http_requests_latency_seconds_by_view_method_count{instance="webapp.example.com:8000",job="webapp",method="GET",view="show-post"}
🟪 django_http_requests_latency_seconds_by_view_method_count{instance="webapp.example.com:8000",job="webapp",method="GET",view="signup"}

django application <- Prometheus



django application <- Prometheus

Any of the workers can respond to the “scrape” request

Inconsistent metric values

Your metrics are not correct any more

Alternatives

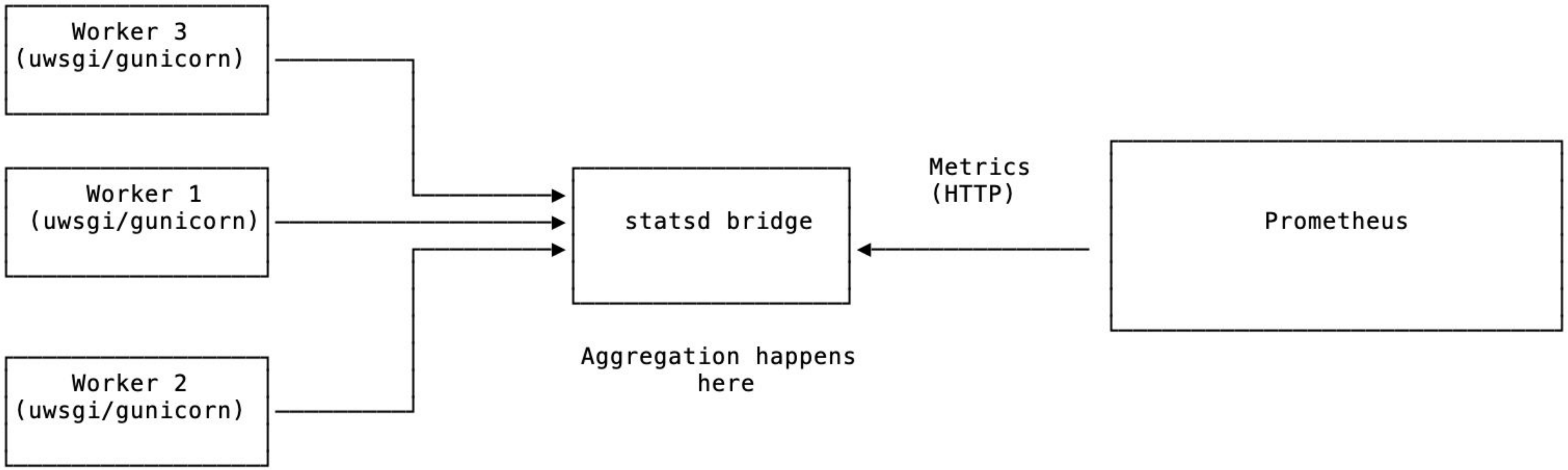
- https://github.com/korfuri/django-prometheus/blob/master/documentation/exports.md

  Expose each worker process on a different port

  Use a “shared” mode where files are used as a way to aggregate metrics

  I personally think both of them are sub-optimal

Alternative - statsd bridge/exporter



Monitoring with statsd bridge (Source reference: Demo 2)

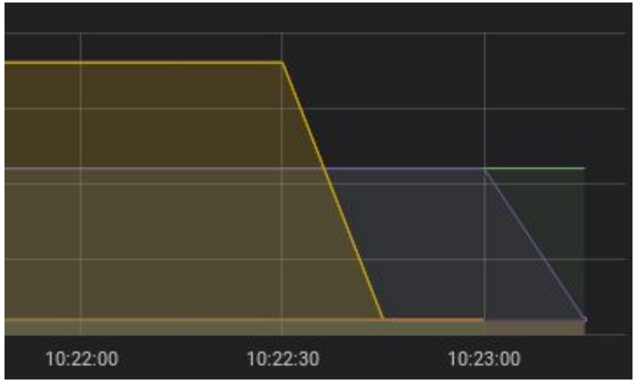
Deployment architecture

As a “side-car” along with your application

VM based deployment: (docker) container/systemd service

Kubernetes: Daemonset (one per node where your application runs)

Things to keep in mind



No persistent storage in statsd bridge

Statsd bridge dies, you don’t have metrics

One more component to run - however, it’s a single binary

Resources

Slides: https://bit.ly/31Jgr7Z

https://github.com/amitsaha/python-monitoring-talk

https://github.com/korfuri/django-prometheus

https://github.com/prometheus/statsd_exporter

https://datadogpy.readthedocs.io/en/latest/

Thank you

Web: https://echorand.me

GitHub: https://github.com/amitsaha/

Email: amit.saha@protonmail.com