



§ OPEN SOURCE SUMMIT

China 2019



AI Ops: Anomaly detection with Prometheus

Marcel Hild, Red Hat

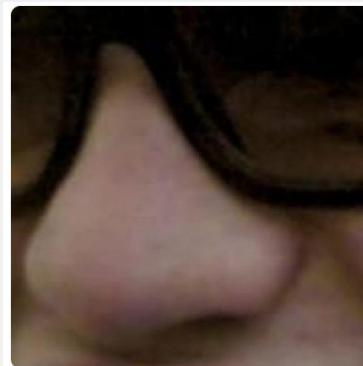


AIOps: Anomaly detection with Prometheus

Spice up your Monitoring with AI

Marcel Hild

Principal Software Engineer @ Red Hat AI CoE / Office of the CTO



Marcel Hild

durandom

old school opensource hacker and
daemon zombie slayer at **@b4mad**
and Red Hat's **@AICoE** CTO Office

 Red Hat

 Kiel, Germany

 hild@b4mad.net

 <http://durandom.de>

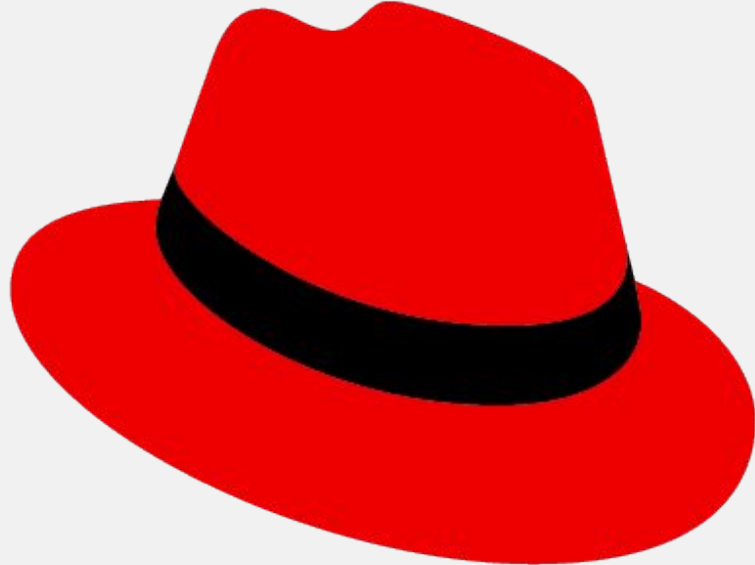
Organizations





Kiel





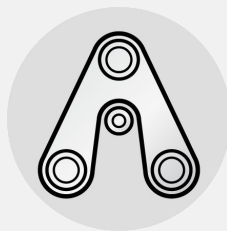
Red Hat

Office of the CTO

HOW RED HAT SEES AI



Represents a workload requirement for our **platforms** across the hybrid cloud.



Applicable to Red Hat's existing core business in order to increase **Open Source** development and production **efficiency**.



Valuable to our customers as specific services and product capabilities, providing an **Intelligent Platform** experience.



Enable customers to build **Intelligent Apps** using Red Hat products as well as our broader partner ecosystem.

010110
101010

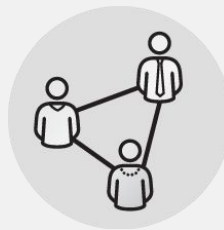
Data as the Foundation

HOW RED HAT SEES AI

Project Thoth and Bots
<http://bit.ly/2zYfb6h>

Represents a workload requirement for our **platforms** across the hybrid cloud.

Applicable to Red Hat's existing core business in order to increase **Open Source** development and production **efficiency**.



Valuable to our customers as specific services and product capabilities, providing an **Intelligent Platform** experience.



Enable customers to build **Intelligent Apps** using Red Hat products as well as our broader partner ecosystem.

010110
101010

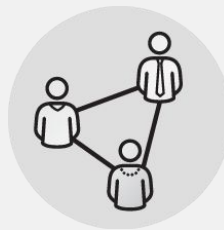
Data as the Foundation

HOW RED HAT SEES AI

Project Thoth and Bots
<http://bit.ly/2zYfb6h>

Represents a workload requirement for our **platforms** across the hybrid cloud.

Applicable to Red Hat's existing core business in order to increase **Open Source** development and production **efficiency**.



Valuable to our customers as specific services and product capabilities, providing an **Intelligent Platform** experience.



Enable customers to build **Intelligent Apps** using Red Hat products as well as our broader partner ecosystem.

OpenDataHub
<http://bit.ly/2y6Nh6m>

0110
0010

Data as the Foundation

HOW RED HAT SEES AI

Project Thoth and Bots
<http://bit.ly/2zYfb6h>

Represents a workload requirement for our **platforms** across the hybrid cloud.

Applicable to Red Hat's existing core business in order to increase **Open Source** development and production **efficiency**.

OpenDataHub
<http://bit.ly/2y6Nh6m>

Data as the Foundation

This Talk

Valuable to our customers as specific services and product capabilities, providing an **Intelligent Platform** experience.

Enable customers to build **Intelligent Apps** using Red Hat products as well as our broader partner ecosystem.

Agenda

Prometheus

Long term storage

Anatomy of an Anomaly

Integration into monitoring setup

What's **not** in this talk



shiny product and the holy
grail of monitoring

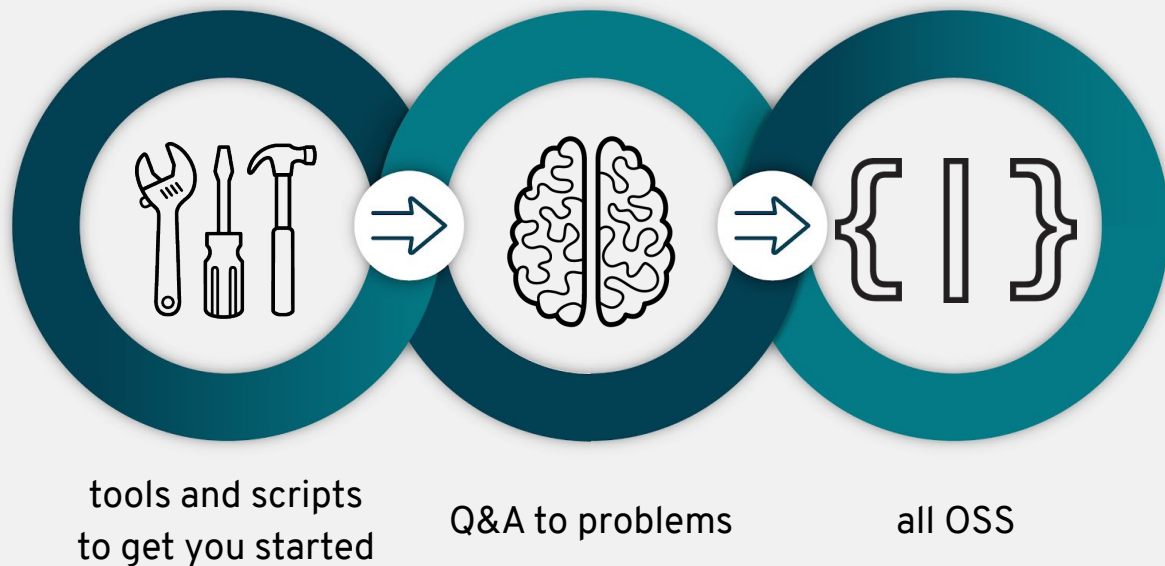


ready solution to turn your
monitoring setup into
spider demon



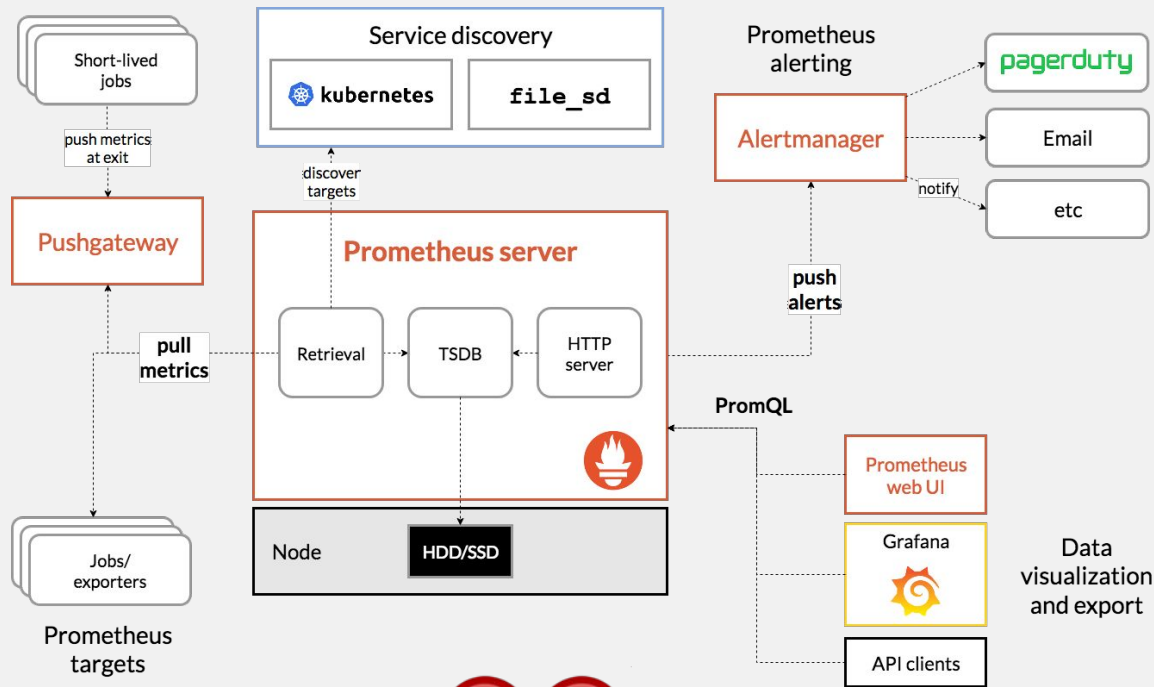
success story how we
turned our messy
monitoring into an advance
ai monitoring

What **is** in this talk



What is prometheus?

Prometheus architecture



Everybody



architecture slides

Prometheus architecture



Simplistic world view

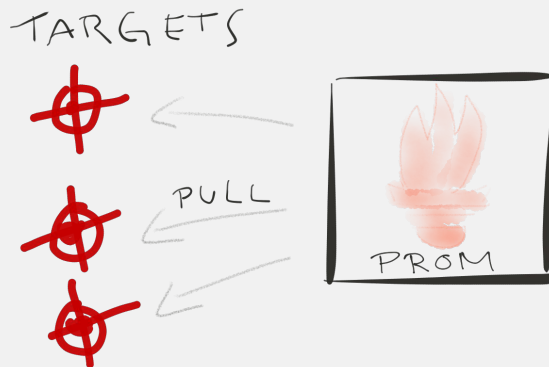
Prometheus architecture

TARGETS



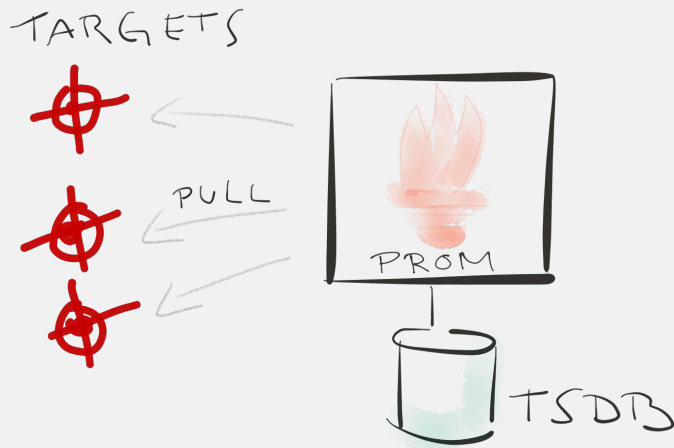
Simplistic world view

Prometheus architecture



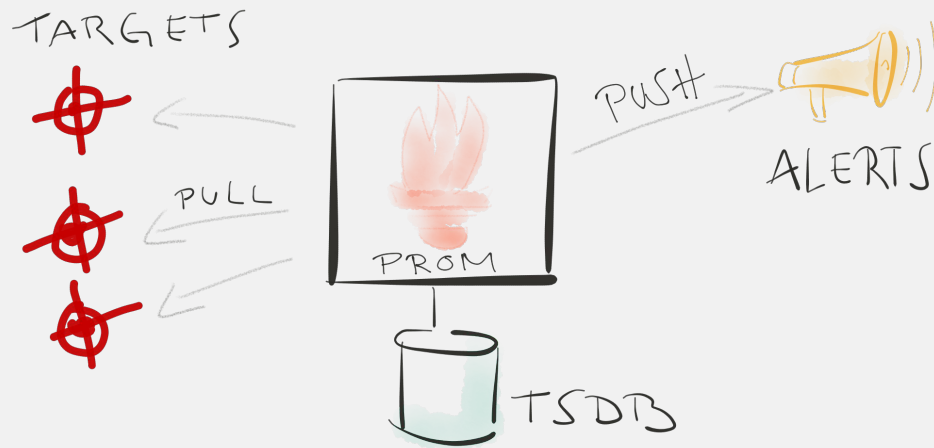
Simplistic world view

Prometheus architecture



Simplistic world view

Prometheus architecture



Simplistic world view

Prometheus is made for

MONITORING

ALERTING

SHORT TERM TIME SERIES DB

What do we need for machine learning?

Show me your DATA

Long term storage of Prometheus data

Too good to be true...



- Prometheus at scale
- Global query view
- Reliable historical data storage
- Unlimited retention
- Downsampling

thanos is in the making,
but until then?

Works great, but...



gh/AIcOE/p-influx
<http://bit.ly/2y6CvwX>

- easily hooked into prometheus with write and read endpoint
- Reliable historical data storage
- Great for data science
 - Pandas integration

Eats RAM for breakfast

Let's just store it...



prometheus scraper

- container can be configured to scrape any prometheus server
- can scrape all or a subset of the metrics
- stores data in ceph or S3 compliant storage
- can be queried with spark sql
- Future Proof: path to Thanos

gh/AIcOE/p-lts
<http://bit.ly/2Qw9pho>



Harness the power of spark to

- Query stored JSON files
- Distribute the workload
- Use spark library

notebook
<http://bit.ly/2PIZZVG>

```
def get_stats(df):  
    # calculate mean  
    mean = df.agg(F.avg(F.col("values"))).head()[0]  
  
    # calculate variance  
    var = df.agg(F.variance(F.col("values"))).head()[0]  
  
    # calculate standard deviation  
    stddev = df.agg(F.stddev(F.col("values"))).head()[0]  
  
    # calculate median  
    median = float(df.approxQuantile("values", [0.5], 0.25)[0])  
  
    return mean, var, stddev, median  
  
mean, var, stddev, median = get_stats(data)  
  
print("\tMean(values): ", mean)  
print("\tVariance(values): ", var)  
print("\tStddev(values): ", stddev)  
print("\tMedian(values): ", median)
```

```
Mean(values): 67087.9063346175  
Variance(values): 56691431555.4375  
Stddev(values): 238099.62527361838  
Median(values): 628.0
```

Things changed



- Prometheus at scale
- Global query view
- Reliable historical data storage
- Unlimited retention
- Downsampling

Success on OpenShift

What do we REALLY need
for machine learning?

Consistent DATA

Prometheus Metric Types



Gauge

A Time Series



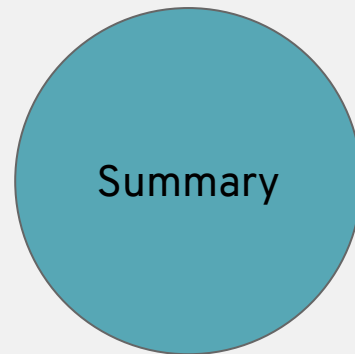
Counter

Monotonically
Increasing



Histogram

Cumulative
Histogram of
Values



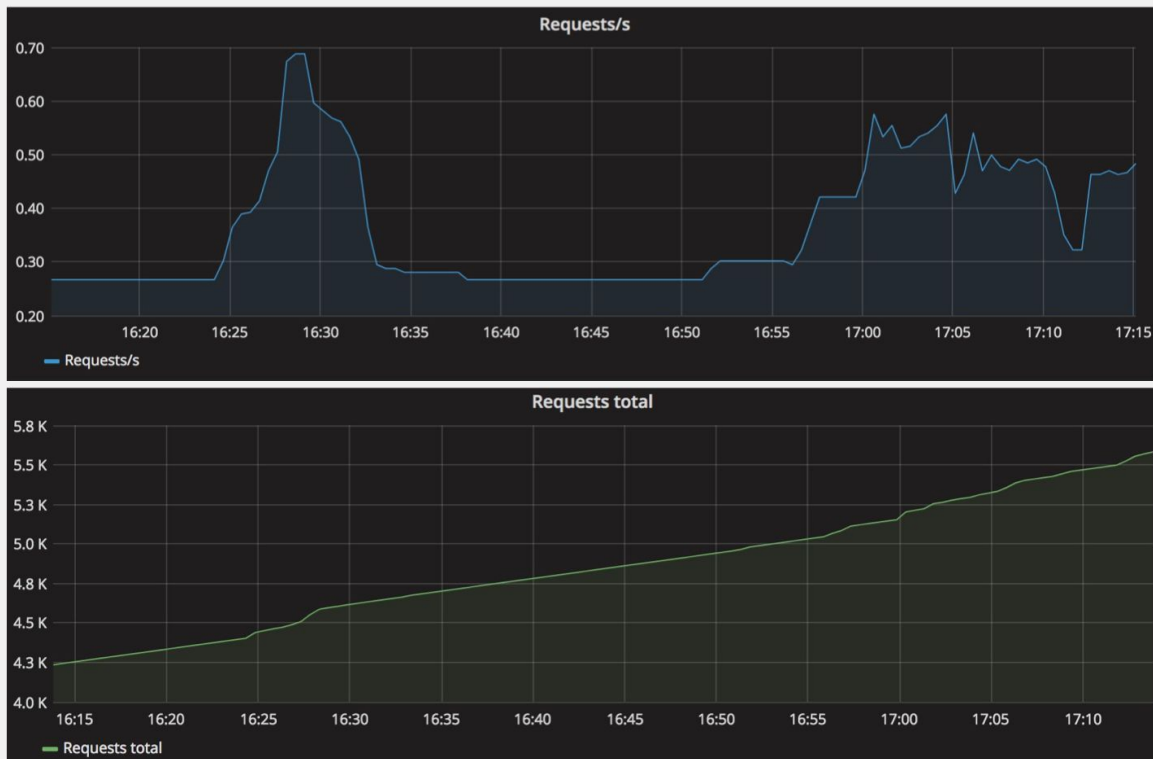
Summary

Snapshot of
Values in a
Time Window

Prometheus Metric Types

Gauge

Counter



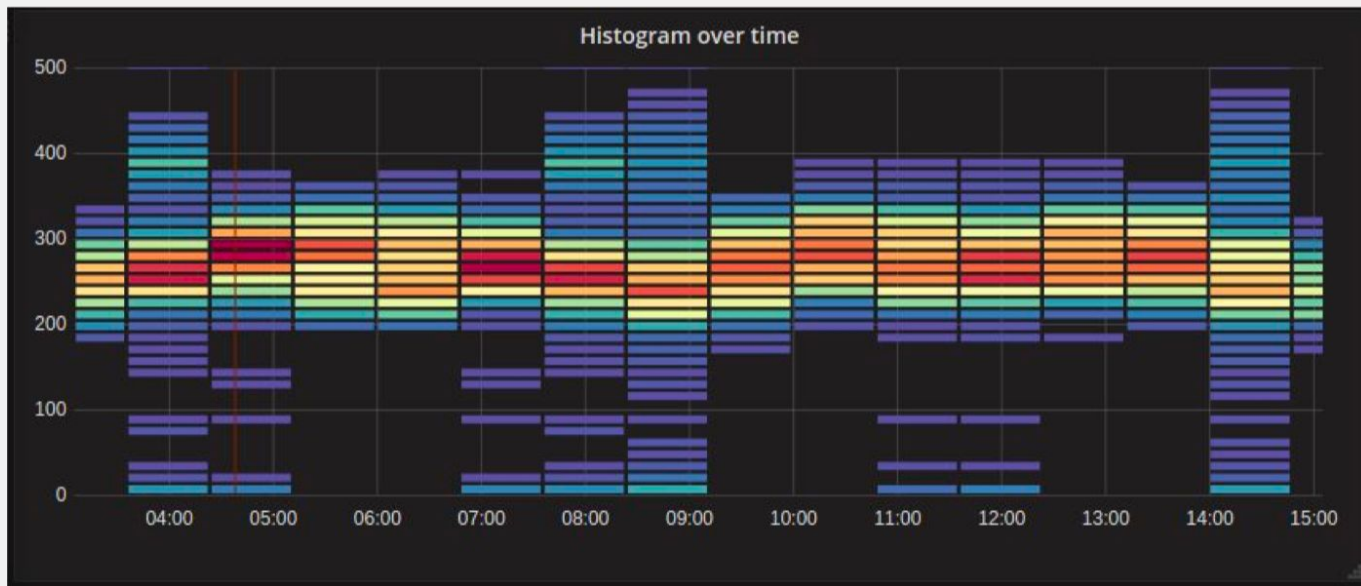
Prometheus Metric Types

Histogram

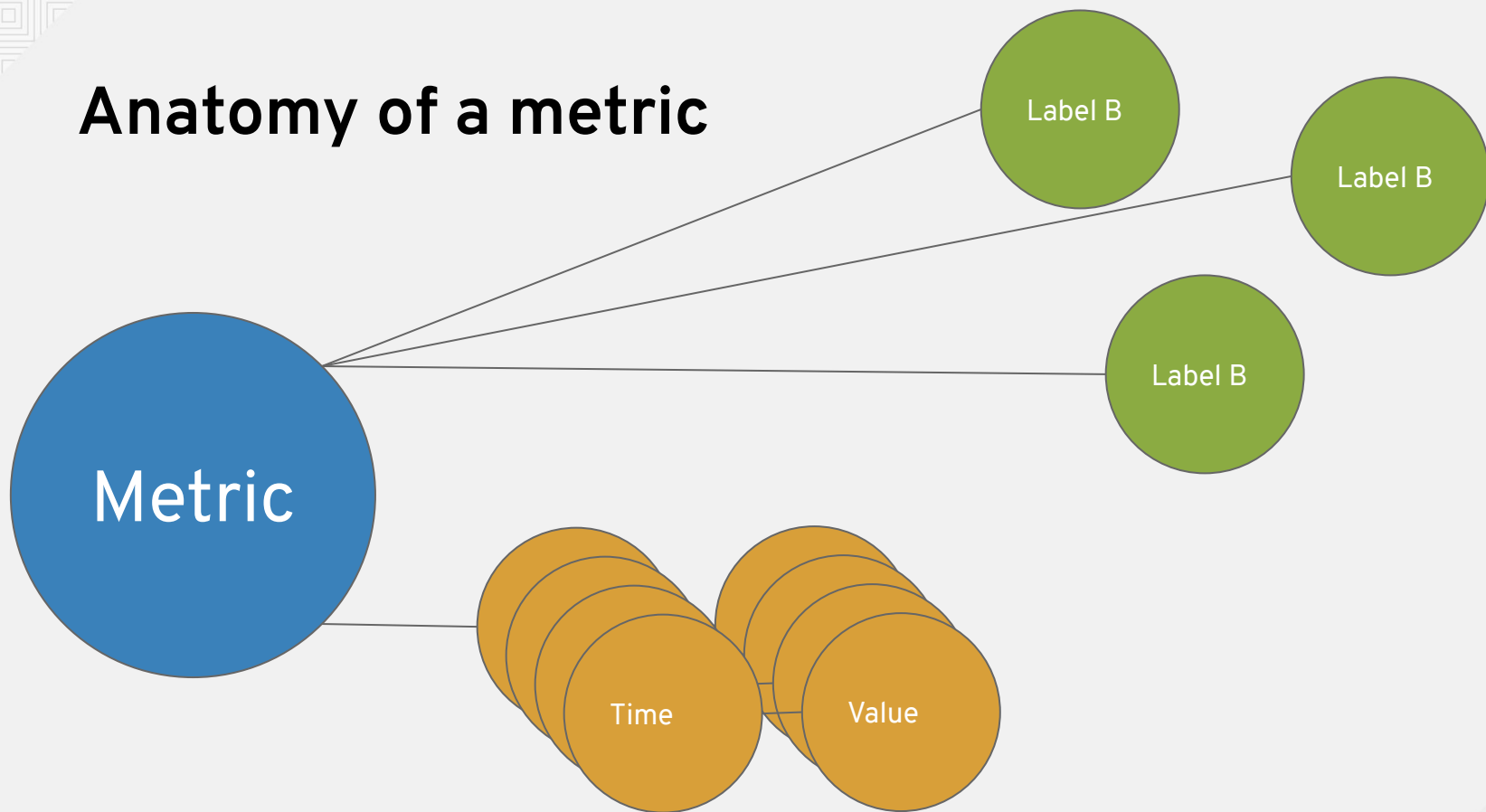
Cumulative

Summary

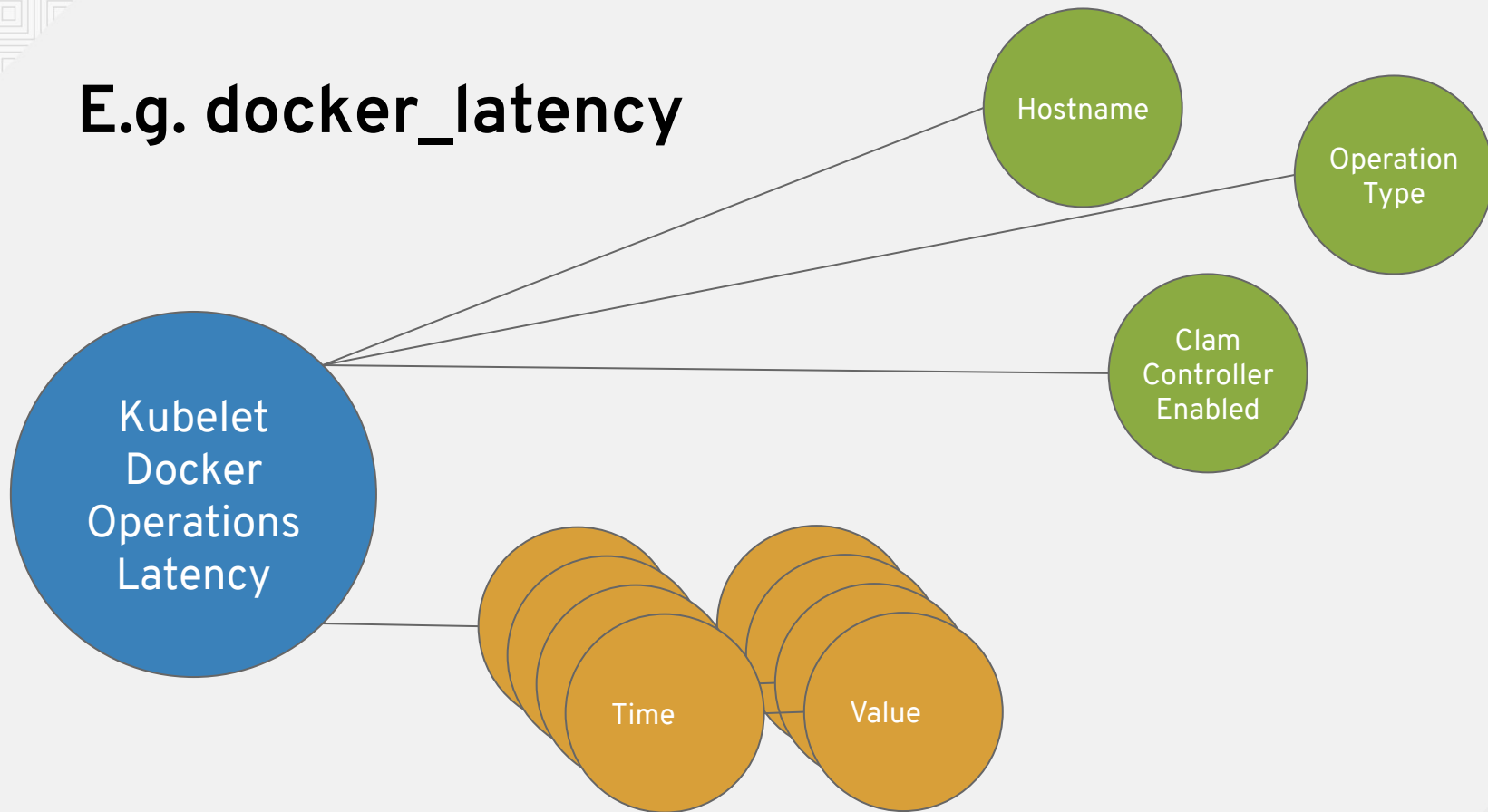
Time Window



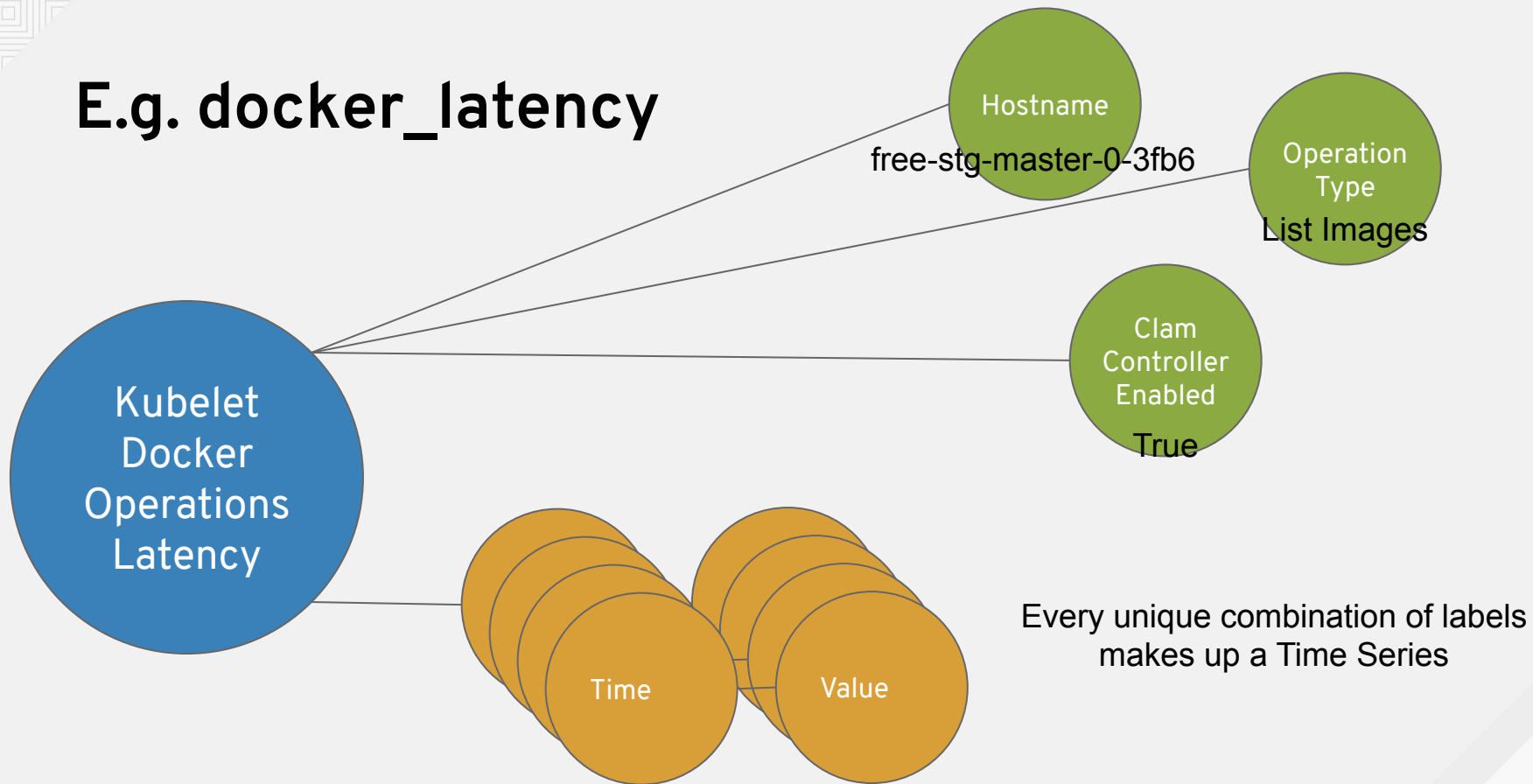
Anatomy of a metric



E.g. docker_latency



E.g. docker_latency

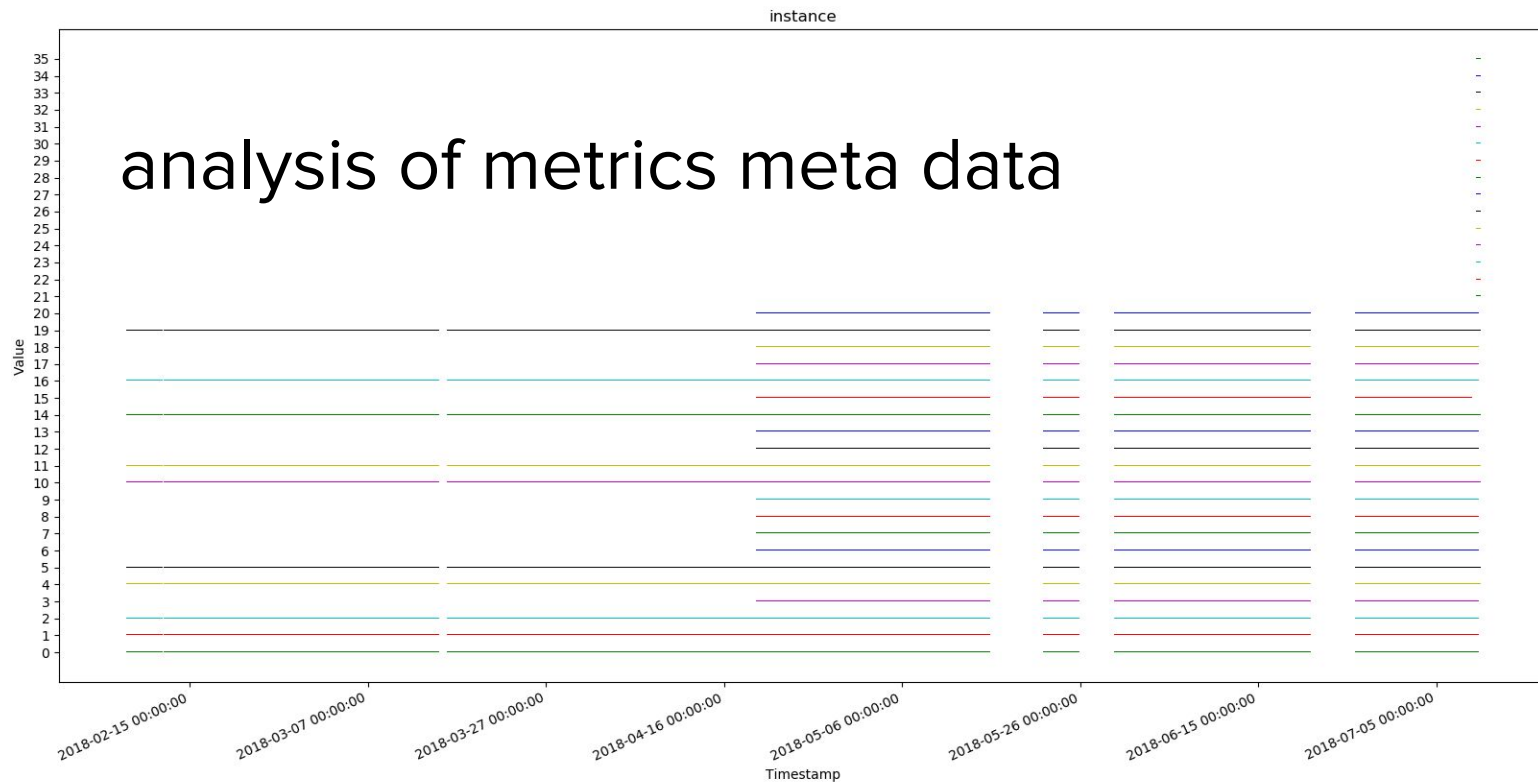


Monitoring is hard

GET /metrics

```
# HELP go_gc_duration_seconds A summary of t
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 9.7014e
go_gc_duration_seconds{quantile="0.25"} 0.00
go_gc_duration_seconds{quantile="0.5"} 0.000
go_gc_duration_seconds{quantile="0.75"} 0.00
go_gc_duration_seconds{quantile="1"} 0.10290
go_gc_duration_seconds_sum 0.239829369
go_gc_duration_seconds_count 196
# HELP go_goroutines Number of goroutines th
# TYPE go_goroutines gauge
go_goroutines 144
# HELP go_memstats_alloc_bytes Number of byt
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 4.5694928e+07
# HELP go_memstats_alloc_bytes_total Total r
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 4.19435624e+09
```

- prometheus doesn't enforce a schema
 - /metrics can expose anything it wants
 - no control over what is being exposed by endpoints or targets
 - it can change if your endpoints change versions
- # of metrics to choose from
 - 1000+ for OpenShift
- State of the Art is Dashboards and Alerting
 - Dashboards and Alerting need domain knowledge
- No tools to explore meta-information in metrics



analysis of metrics meta data

Meta-data tooling
<http://bit.ly/2A1hXHX>

Anomaly Types

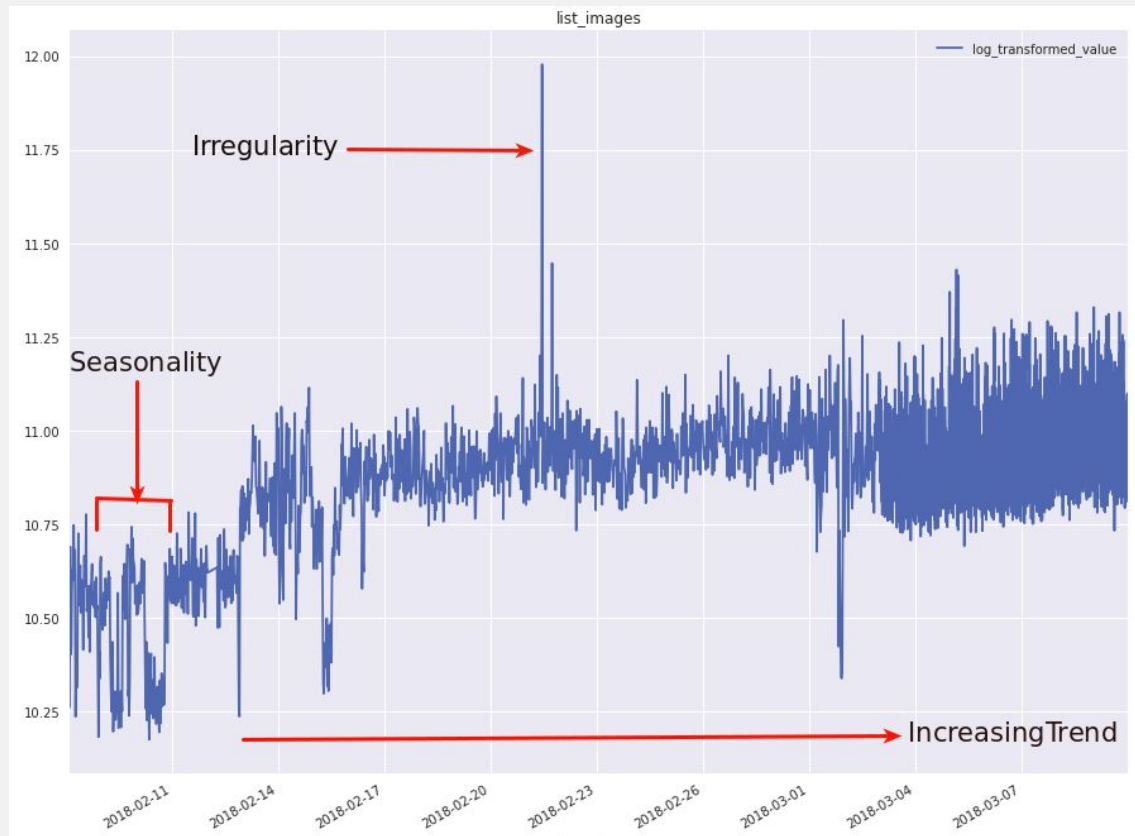
Components of Time Series

Trend

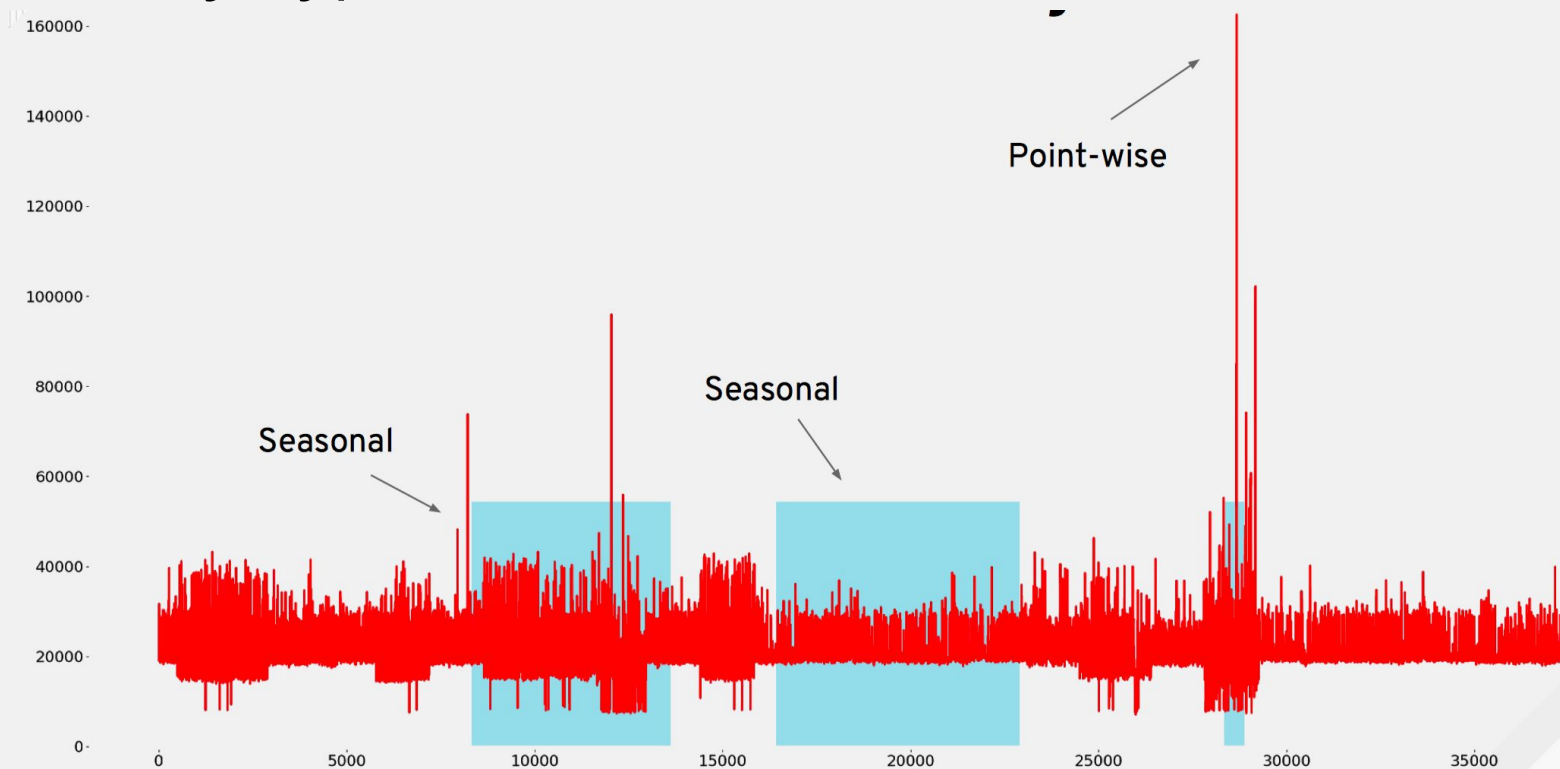
Increase or decrease in the series over a period of time.

Seasonality

Regular pattern of up and down fluctuations. It is a short-term variation occurring due to seasonal factors.

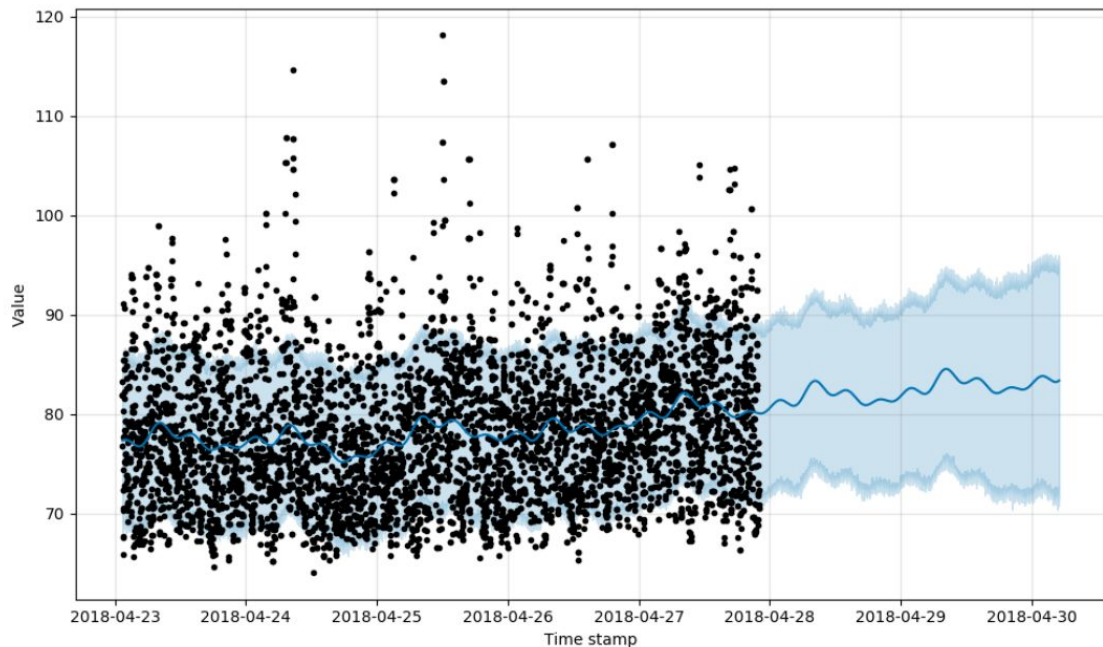


Anomaly Types



Anomaly Detection with Prophet

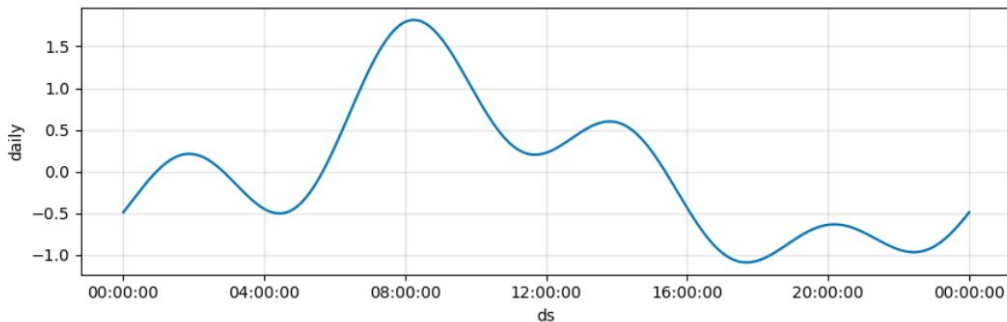
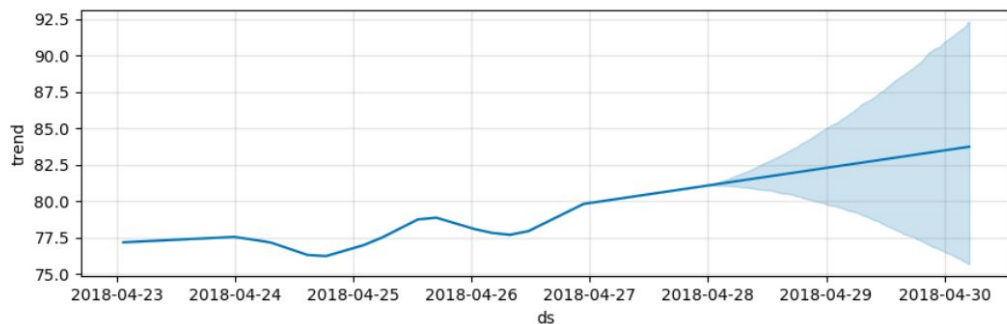
Predicting future data and dynamic thresholds



- `list_images` operation
- on OpenShift
- monitored by prometheus
- detecting outliers
- upper and lower bands

Anomaly Detection with Prophet

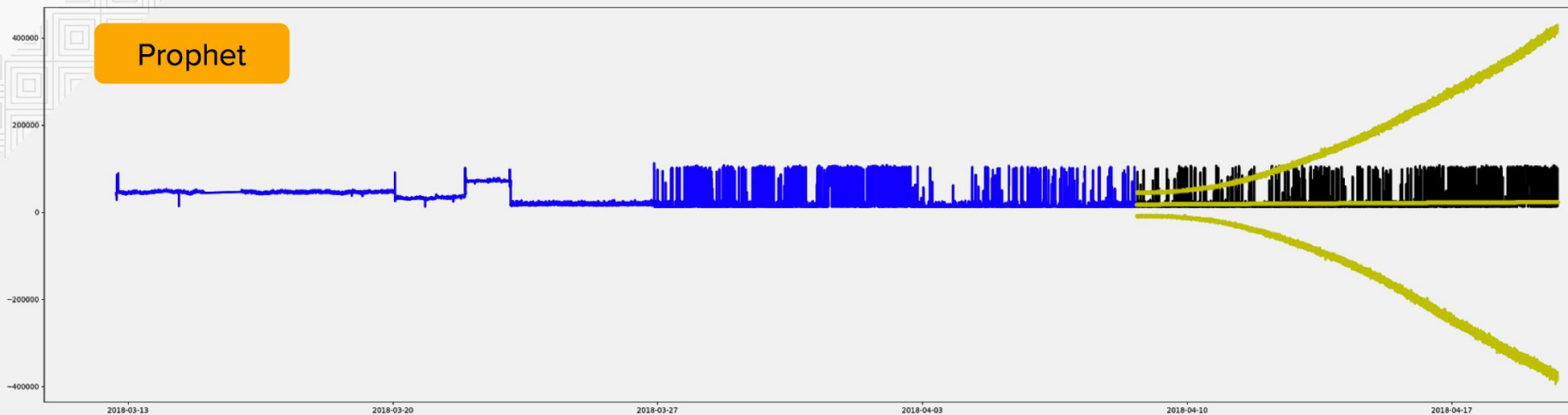
Extracting trends and seasonality



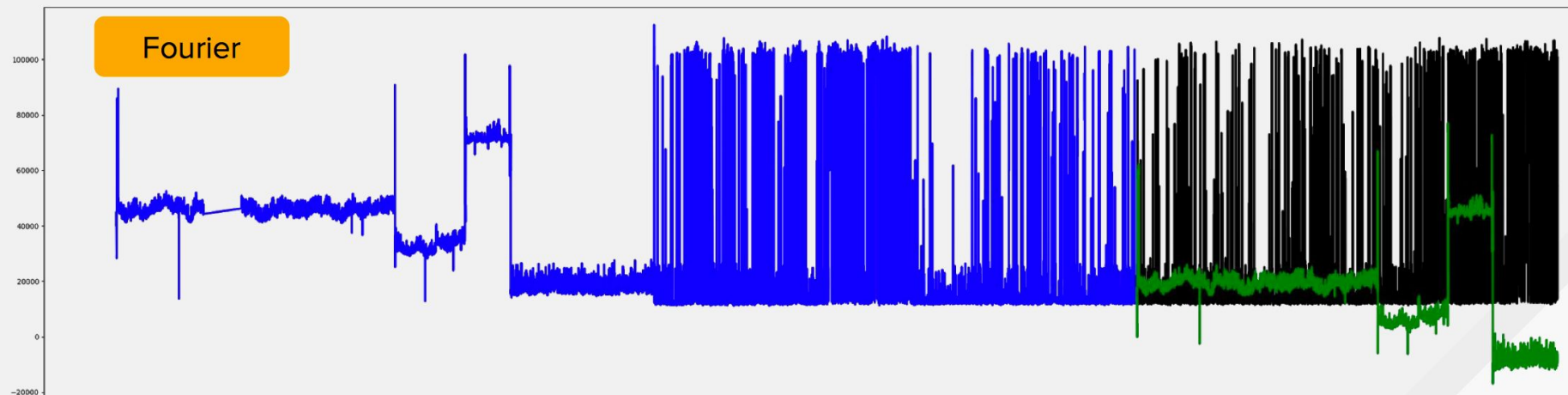
- `list_images` operation
- on OpenShift
- monitored by prometheus
- upward trends
- intraday seasonality

CoE/prophet
<http://bit.ly/2pLzGNj>

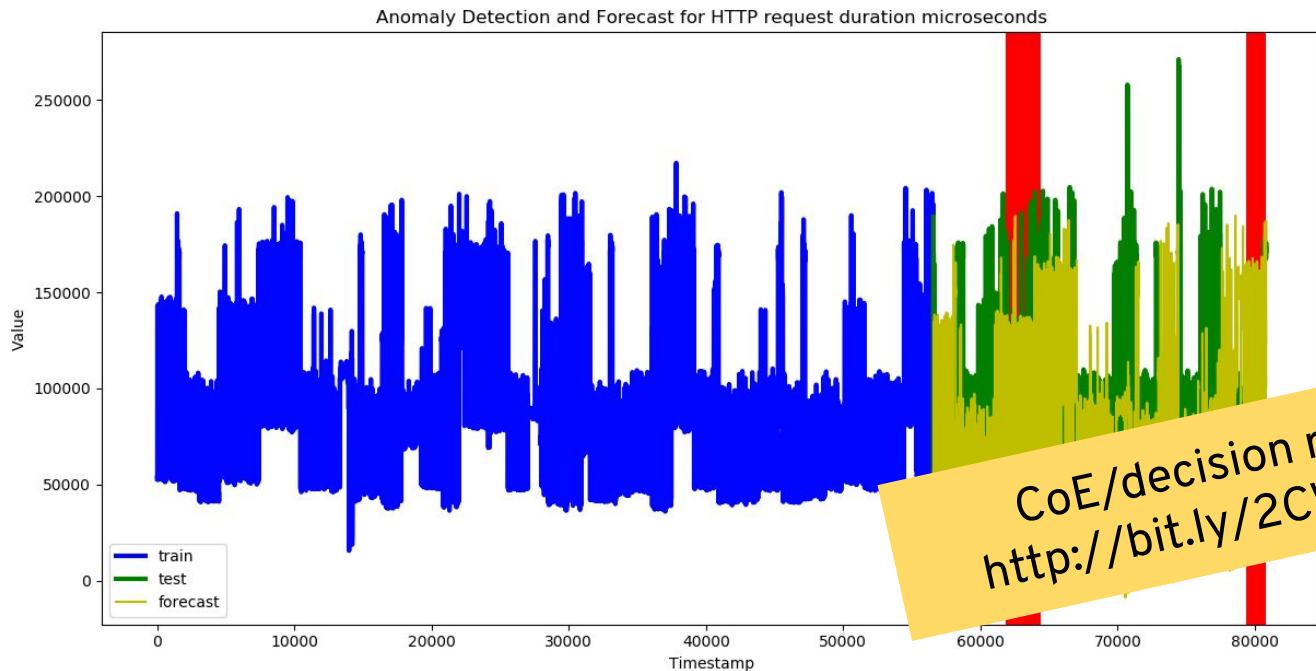
Prophet



Fourier

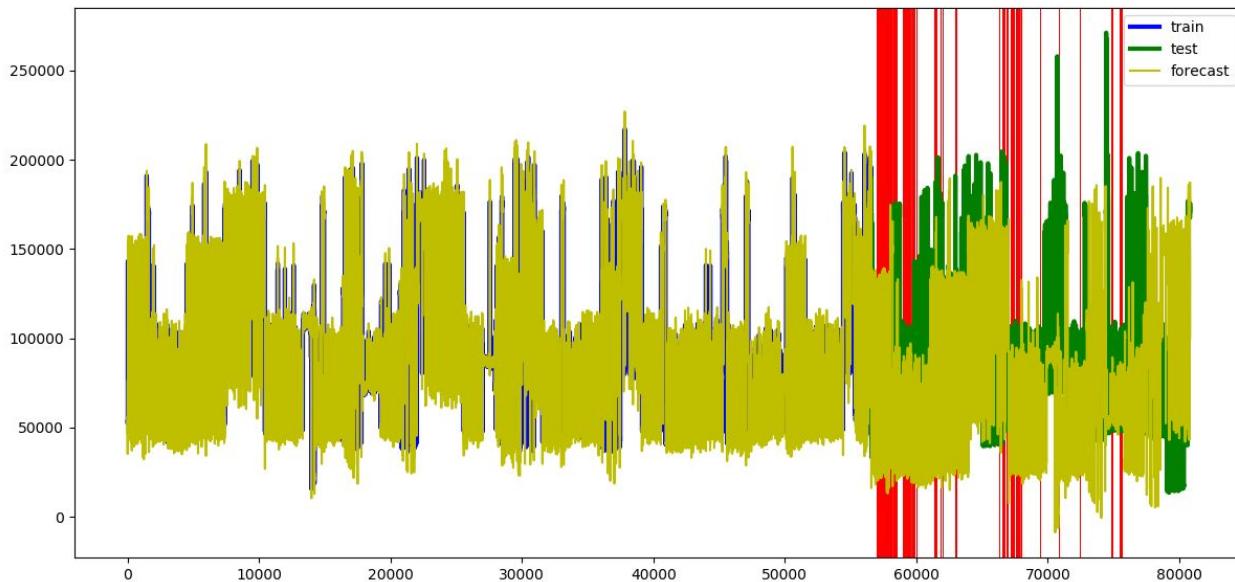


The Accumulator

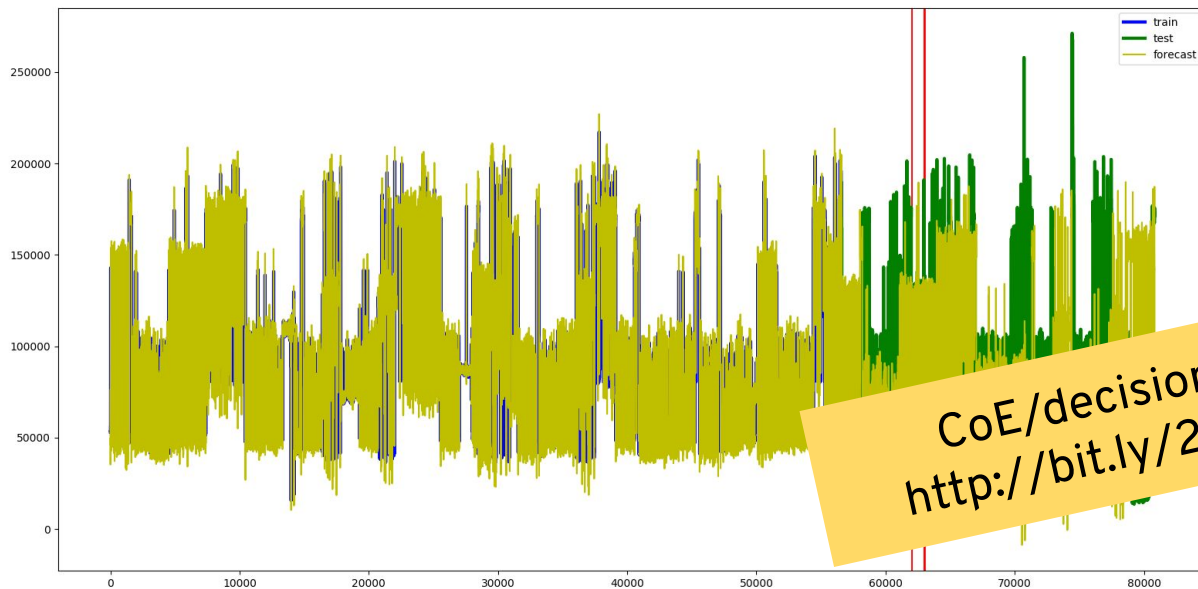


CoE/decision rules
<http://bit.ly/2CWY9YB>

The Tail Probability



Combined

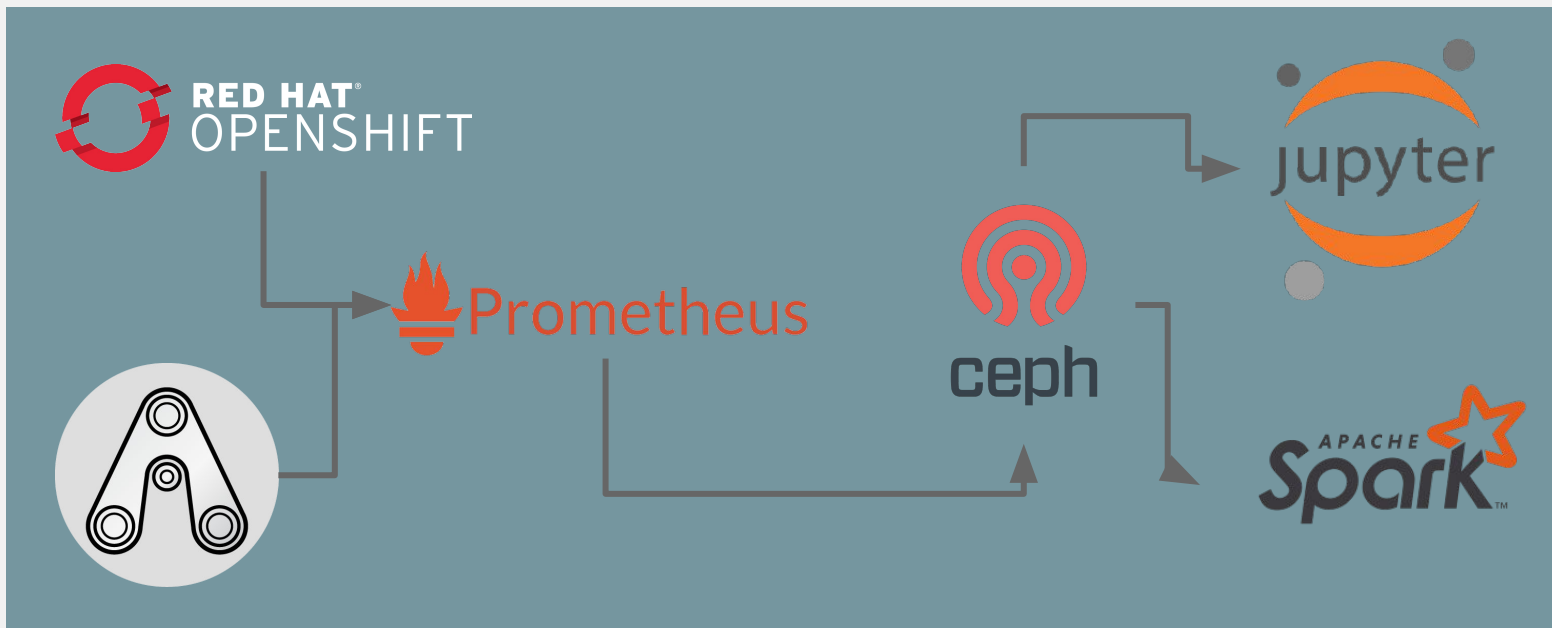


CoE/decision rules
<http://bit.ly/2CWY9YB>

architecture setup so far

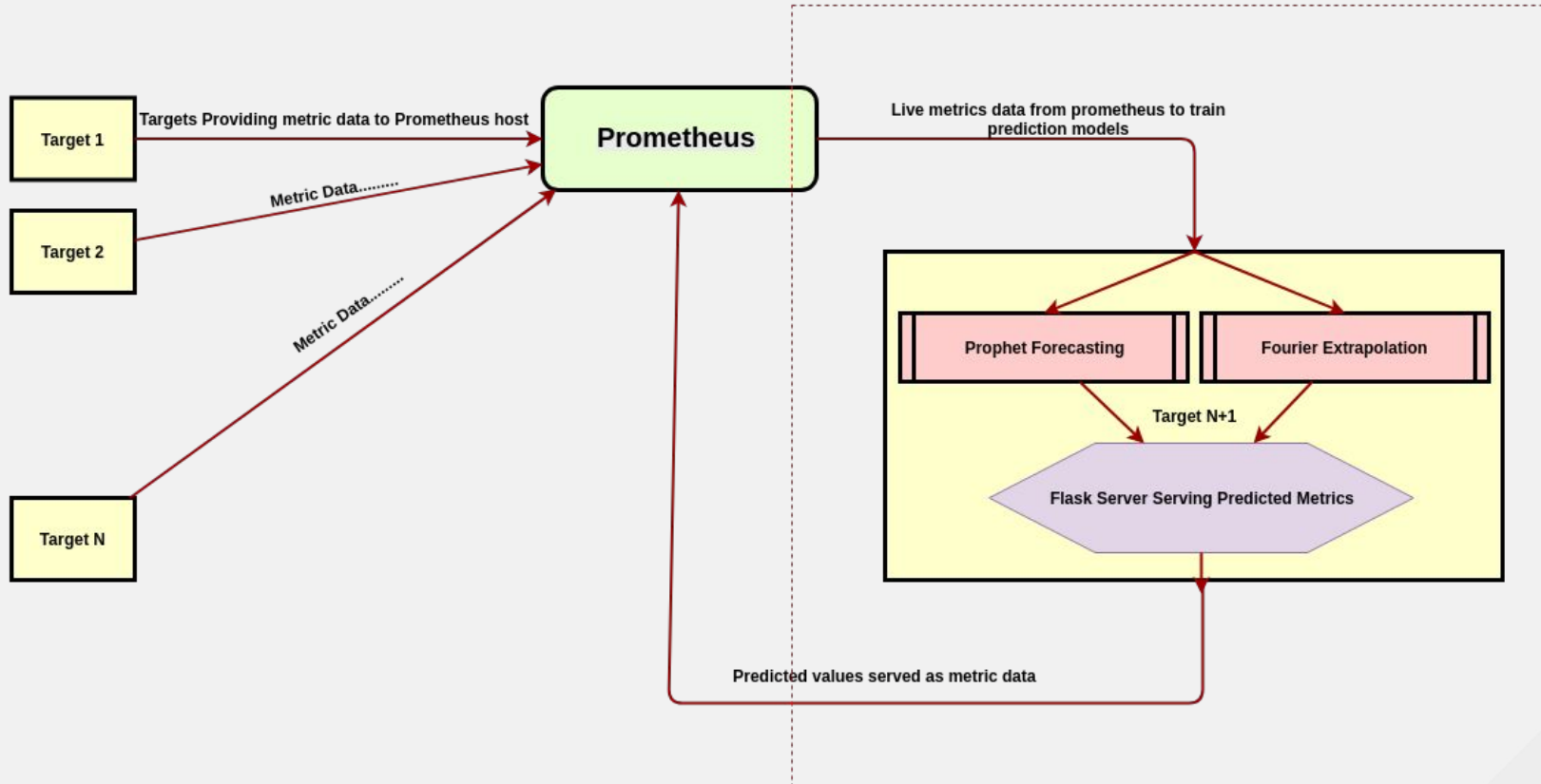
Research Setup

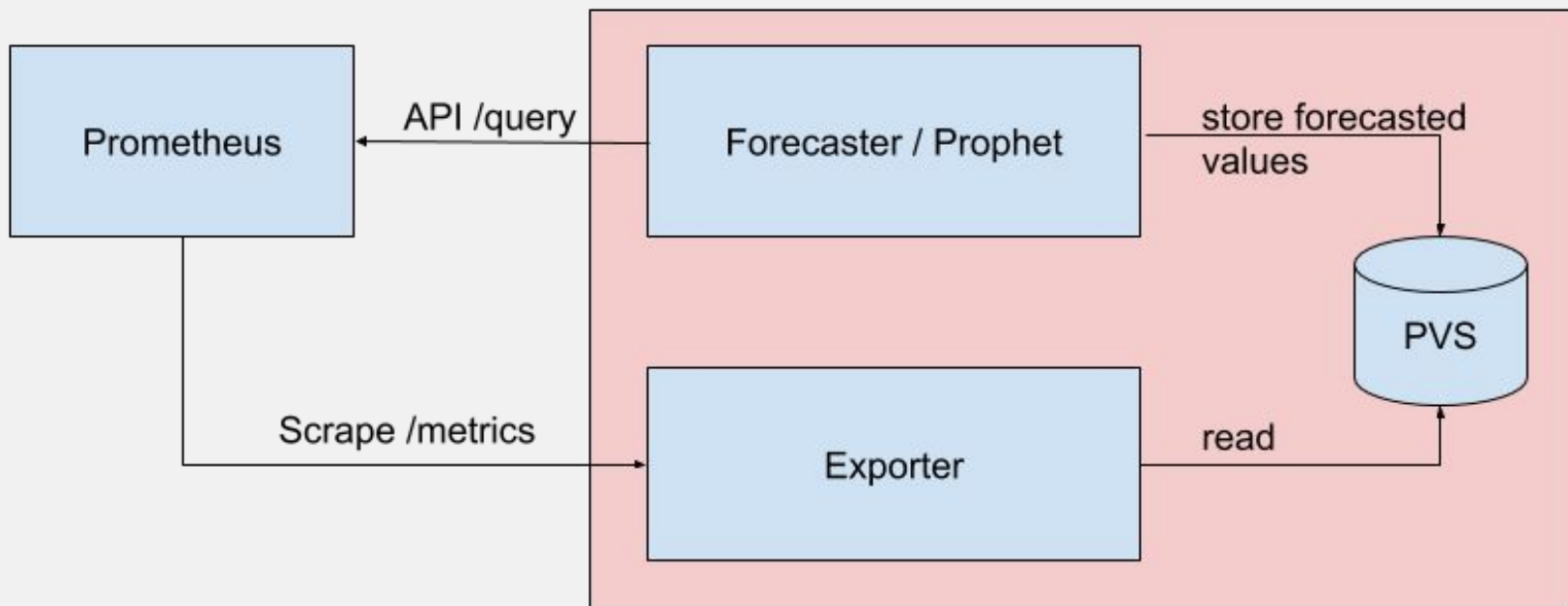
100% OpenSource Tooling



Now what? I want to

Prometheus Training Pipeline
















GitHub, Inc. [US]

<https://github.com/AICoE/prometheus-anomaly-detector>

 Dockerfile	Update Dockerfil
 Makefile	Add Makefile for ease of
 README.md	Update README.md
 app.py	Add more comments for
 ceph.py	Add functionality to retai
 model.py	Make the live data query
 prometheus.py	Make the live data query
 requirements.txt	Update requirements.txt
 train-prophet-deployment-templa...	Add deployment templat

- Ready to use container
 - Local deployment
 - Kubernetes
 - OpenShift build config

CoE/prom-ad
<http://bit.ly/2yulCfh>

Runtime configuration

```
29 # Specific metric to run the model on
30 metric_name = os.getenv('METRIC_NAME', 'kubelet_docker_operations_latency_microseconds')
```

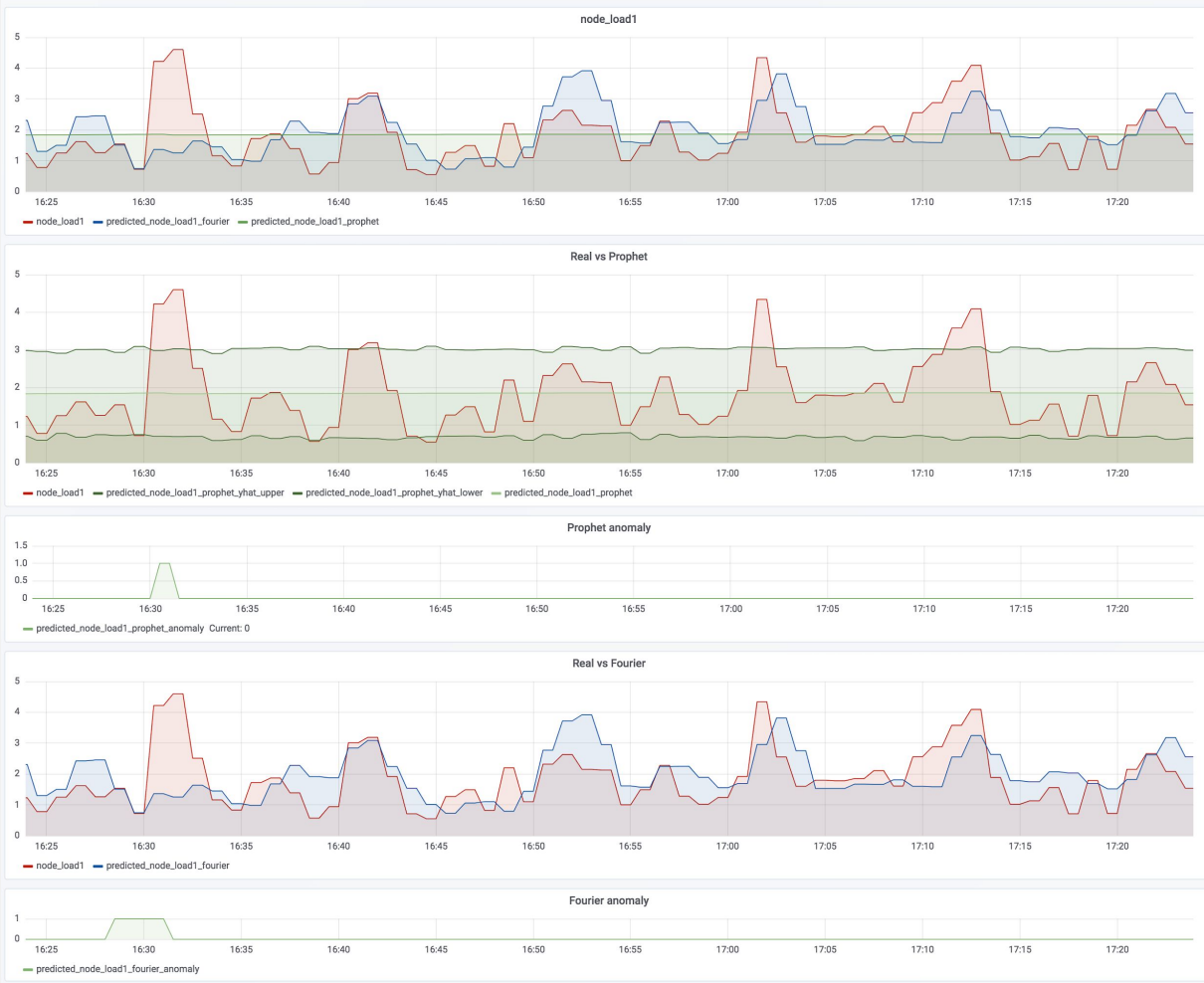
Expose predictions via **/metrics** endpoint

```
# HELP predicted_kubelet_docker_operations_latency_microseconds_prophet_anomaly Detected Anomaly using the Prophet model
# TYPE predicted_kubelet_docker_operations_latency_microseconds_prophet_anomaly gauge
predicted_kubelet_docker_operations_latency_microseconds_prophet_anomaly{beta_kubernetes_io_arch="amd64",beta_kubernetes_io_os="linux'
0001.ocp.prod.upshift.eng.rdu2.redhat.com",operation_type="version",provider="rhos",quantile="0.5",region="compute",size="small"} 0.0
# HELP predicted_kubelet_docker_operations_latency_microseconds_fourier_anomaly Detected Anomaly using the Fourier model
# TYPE predicted_kubelet_docker_operations_latency_microseconds_fourier_anomaly gauge
predicted_kubelet_docker_operations_latency_microseconds_fourier_anomaly{beta_kubernetes_io_arch="amd64",beta_kubernetes_io_os="linux'
0001.ocp.prod.upshift.eng.rdu2.redhat.com",operation_type="version",provider="rhos",quantile="0.5",region="compute",size="small"} 0.0
```

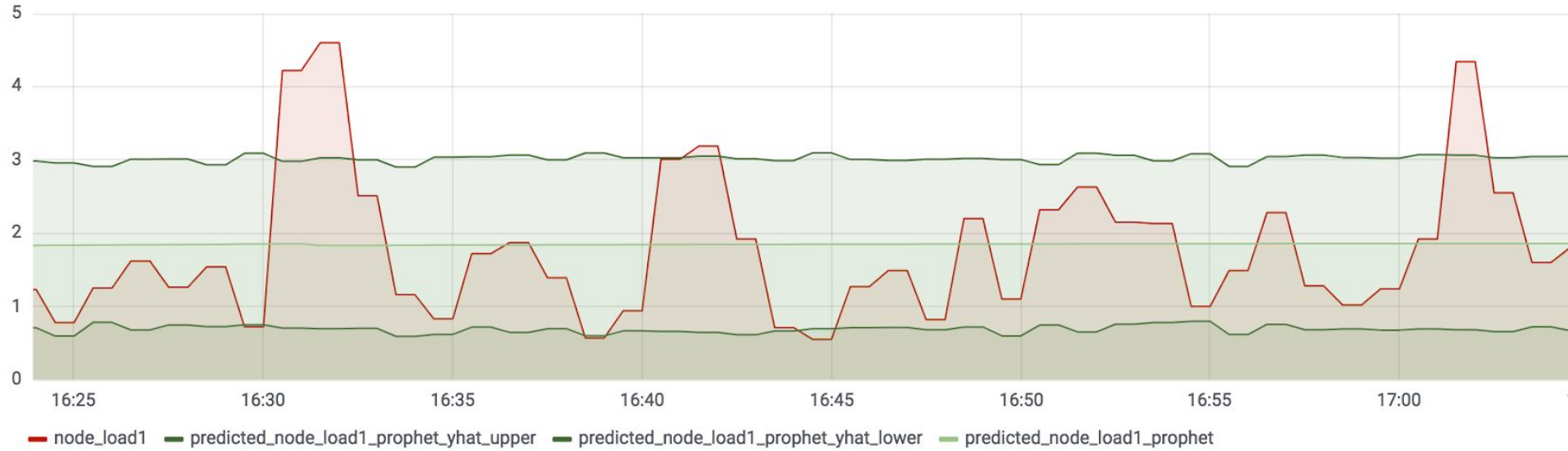
Alerting Rules

```
groups:  
- name: Testing alert  
  rules:  
  
  - alert: MetricOutOfProphetBounds  
    expr: kubelet_docker < ignoring(job, instance) predicted_values_prophet_yhat_lower or kubelet_docker > ignoring(job, instance) predicted_values_prophet_yhat_upper  
    #for: 5m  
    annotations:  
      summary: "Metric out of bounds"  
      description: "Metric is out of range of the predicted Prophet values"  
  
  - alert: MetricOutOfFourierBounds  
    expr: kubelet_docker < ignoring(job, instance) predicted_values_fourier_yhat_lower or kubelet_docker > ignoring(job, instance) predicted_values_fourier_yhat_upper  
    annotations:  
      summary: "Metric out of bounds"  
      description: "Metric is out of range of the predicted Fourier values"
```

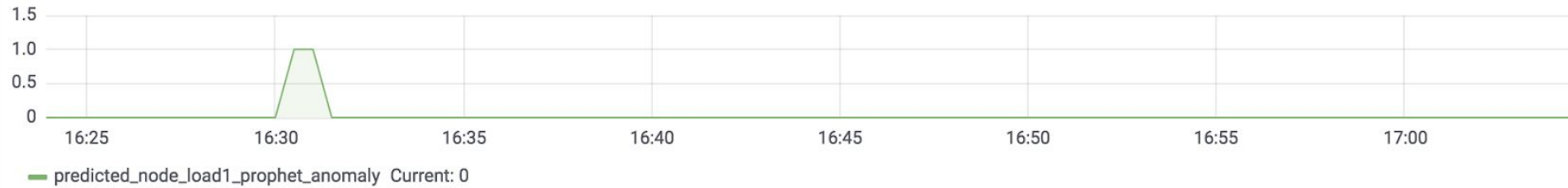
Demo Time



Real vs Prophet

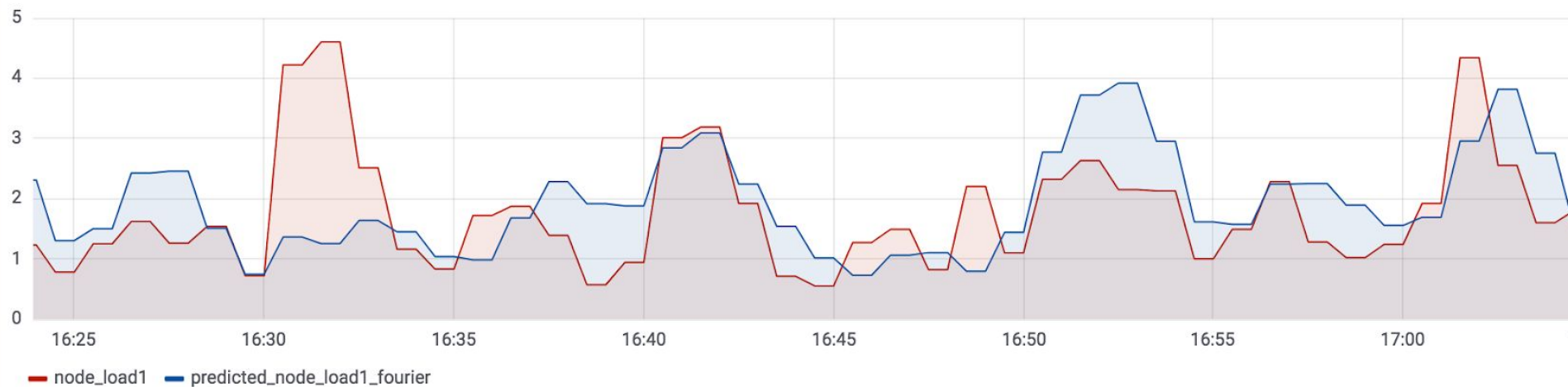


Prophet anomaly



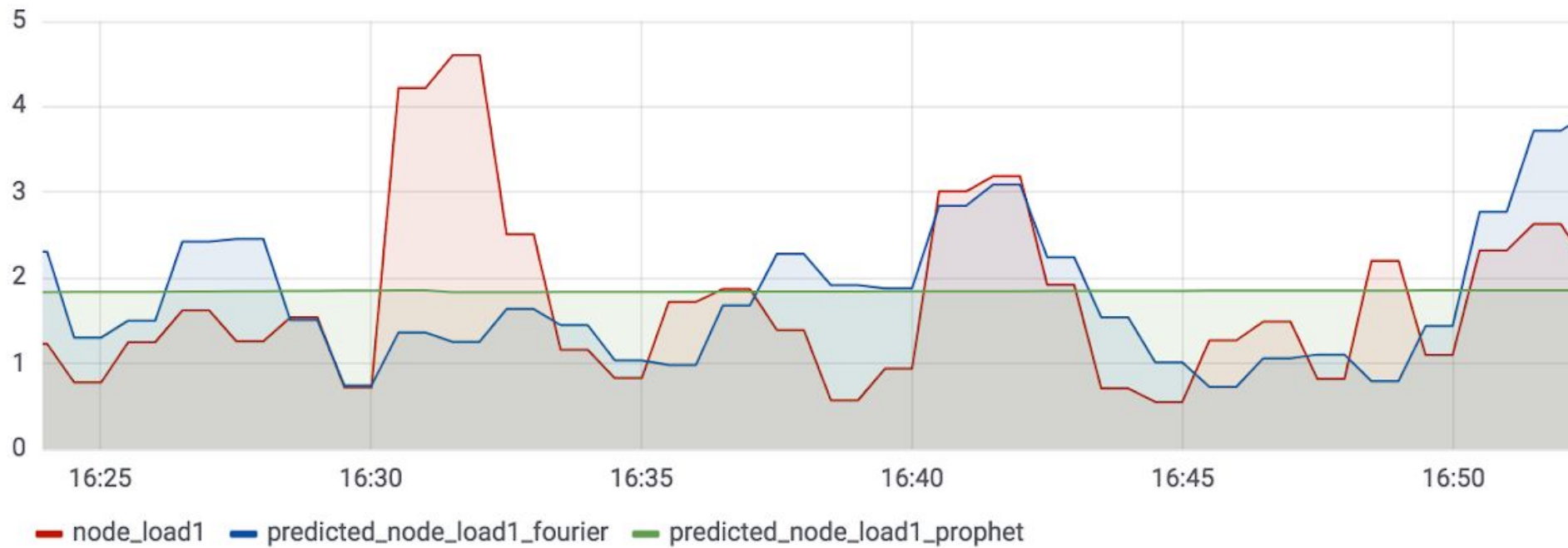


Real vs Fourier



Fourier anomaly





I like what you got

notebooks
<http://bit.ly/2PIZZVG>

gh/AICoE/p-influx
<http://bit.ly/2y6CvwX>

Project Thoth and Bots
<http://bit.ly/2zYfb6h>

CoE/prophet
<http://bit.ly/2pLzGNj>

CoE/decision rules
<http://bit.ly/2CWY9YB>

CoE/prom-ad
<http://bit.ly/2yulCfh>

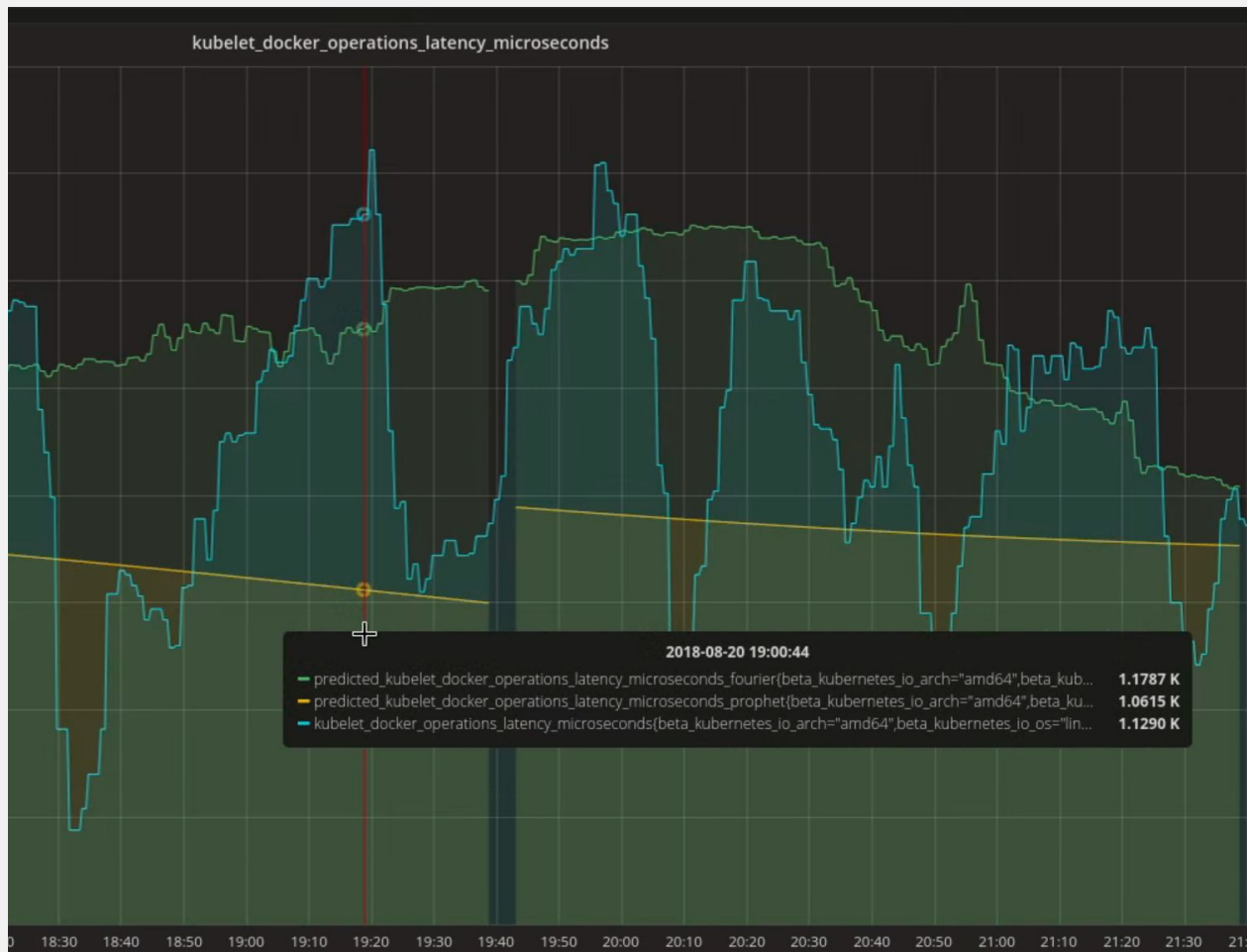
Meta-data tooling
<http://bit.ly/2A1hXHX>

OpenDataHub
<http://bit.ly/2y6Nh6m>

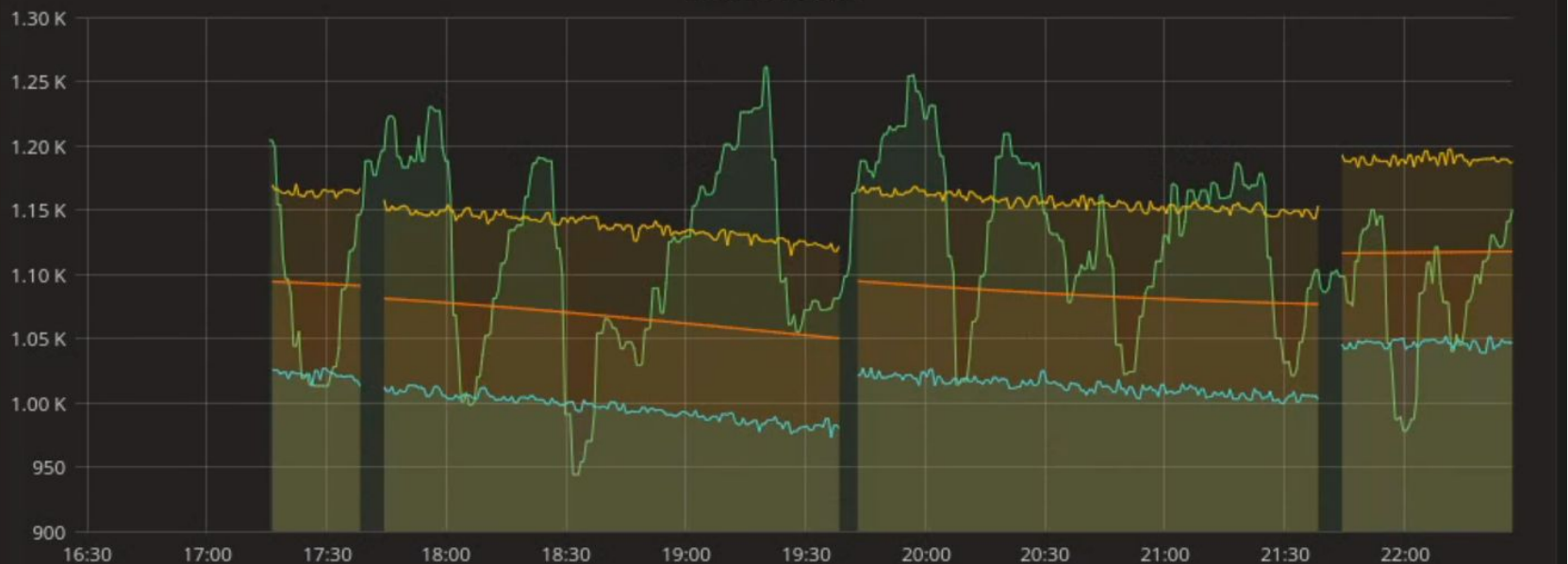
Thanos Blog Post
<https://red.ht/2HpB8Az>

gh/AICoE/p-lts
<http://bit.ly/2Qw9pho>

QUESTIONS?



Real vs Prophet



kubelet_docker_operations_latency_microseconds(beta_kubernetes_io_arch="amd64",beta_kubernetes_io_os="linux",instance="cpt-0001.ocp.prod.upshift.eng.rdu2.1

predicted_kubelet_docker_operations_latency_microseconds_prophet_yhat_upper(beta_kubernetes_io_arch="amd64",beta_kubernetes_io_os="linux",exported_insta

predicted_kubelet_docker_operations_latency_microseconds_prophet_yhat_lower(beta_kubernetes_io_arch="amd64",beta_kubernetes_io_os="linux",exported_insta

predicted_kubelet_docker_operations_latency_microseconds_prophet(beta_kubernetes_io_arch="amd64",beta_kubernetes_io_os="linux",exported_instance="cpt-0001