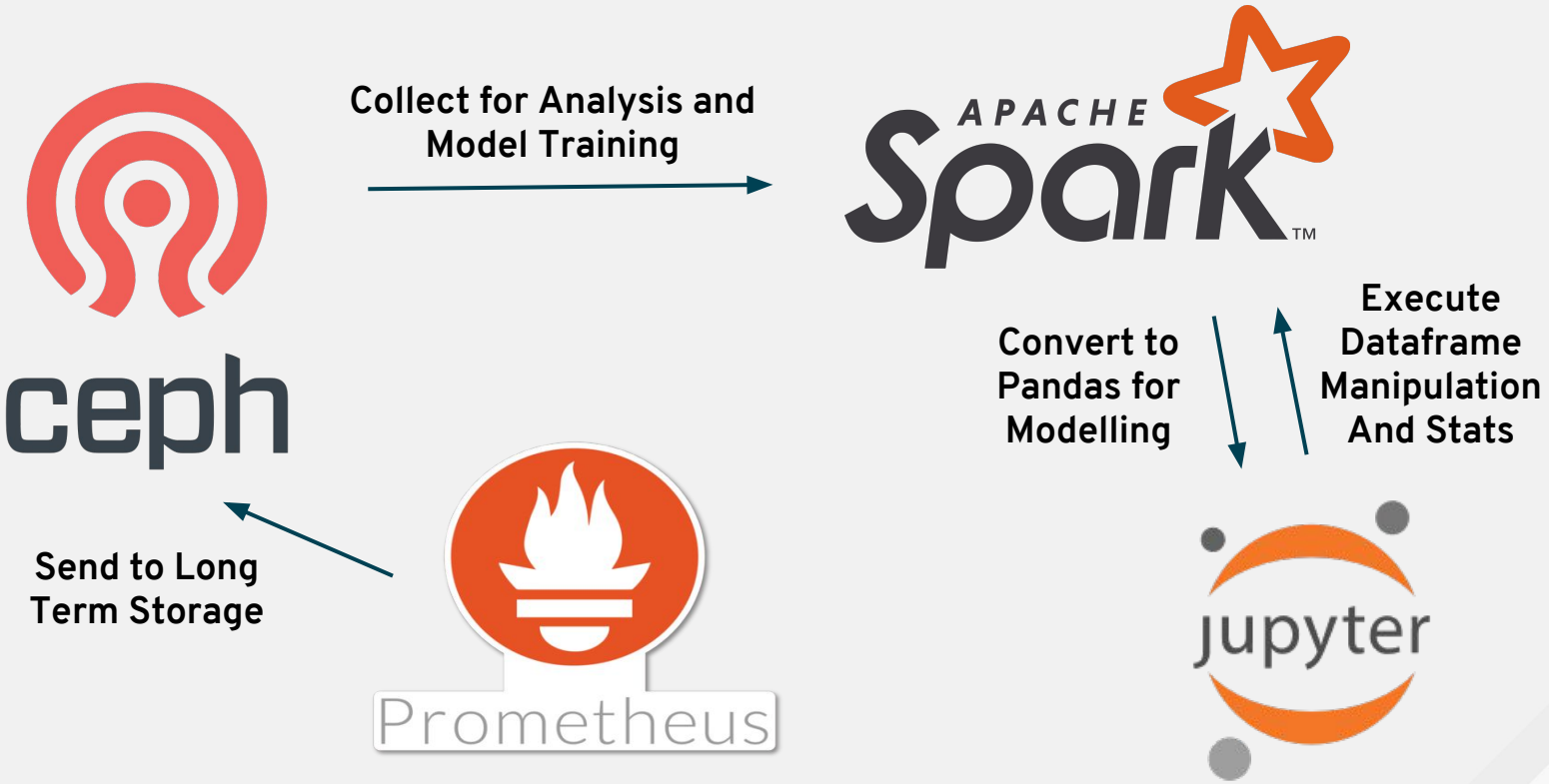
redhat.

Data Science on Prometheus Metrics

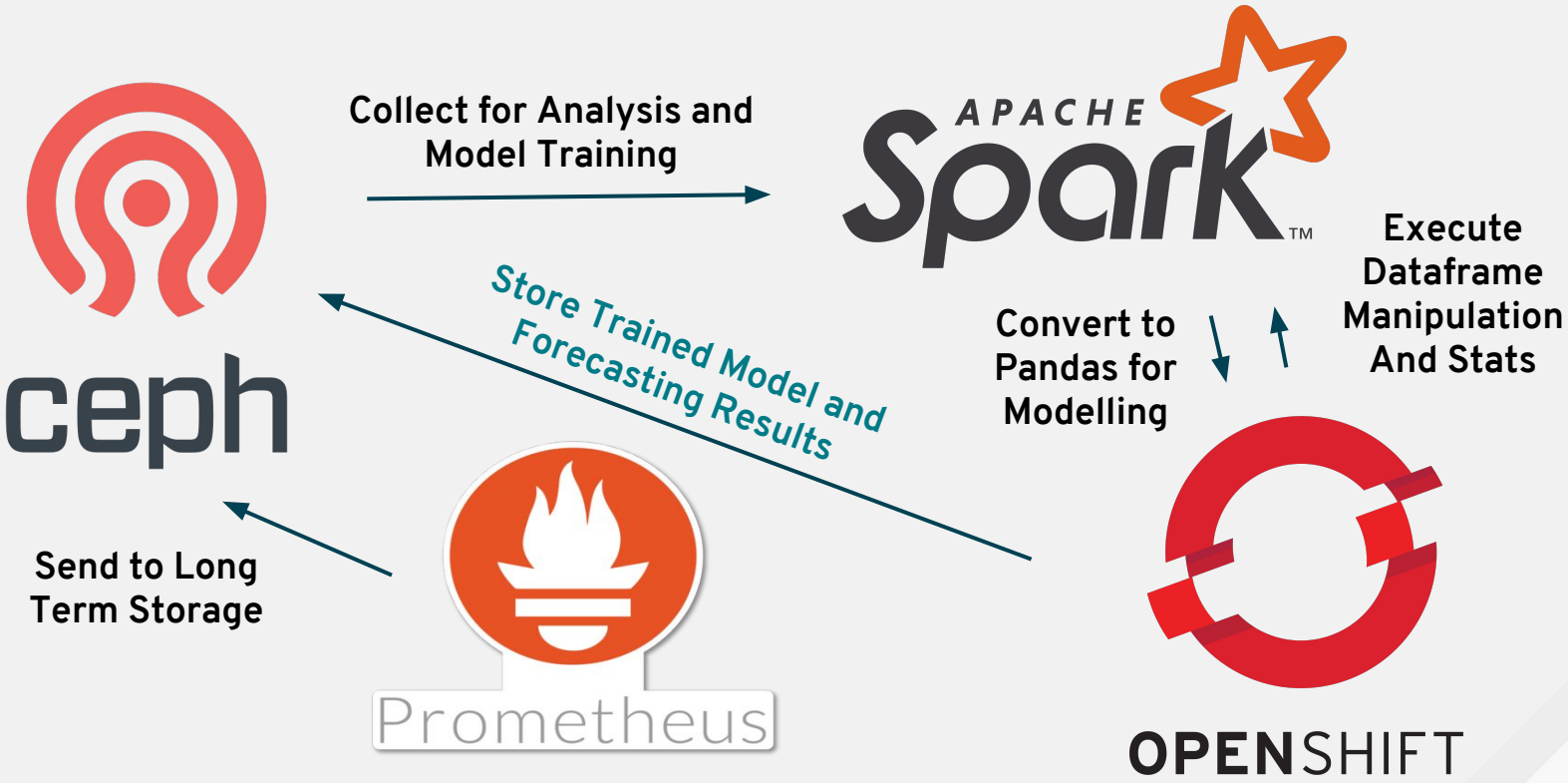
Natasha Frumkin
AI/Data Science Intern
July 12, 2018

Scrapes metrics from targets and stores in a time series

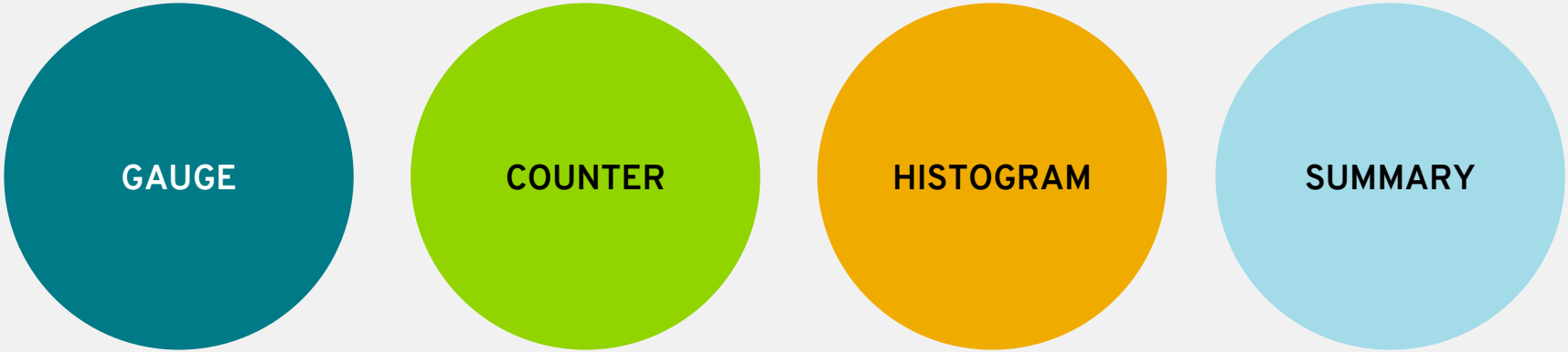
Development Data Flow



Production Data Flow



Prometheus Metric Types



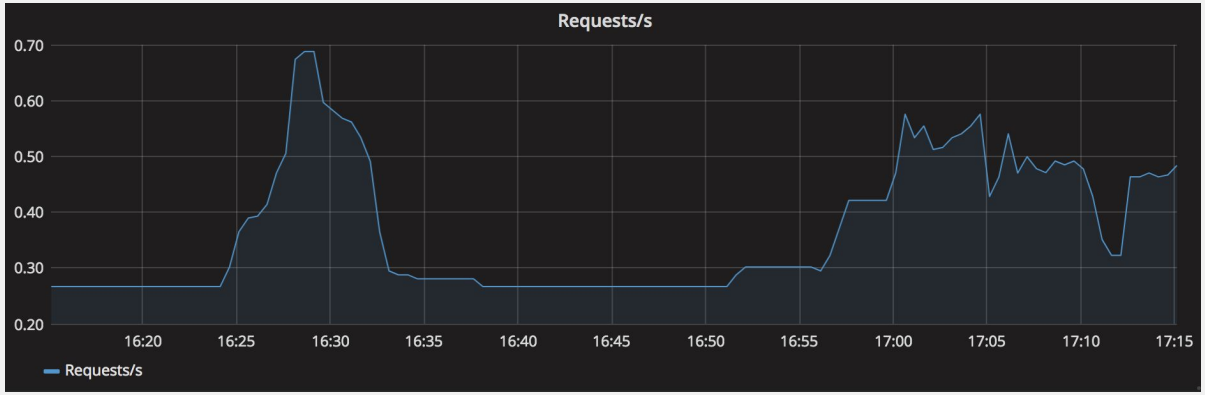
A Time Series

Monotonically Increasing

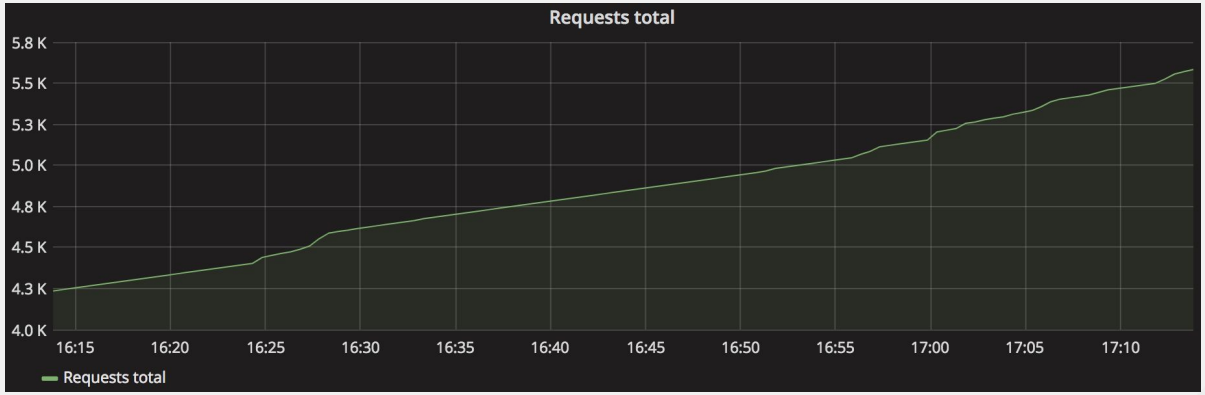
Cumulative Histogram of Values

Snapshot of Values in a Time Window

GAUGE



COUNTER

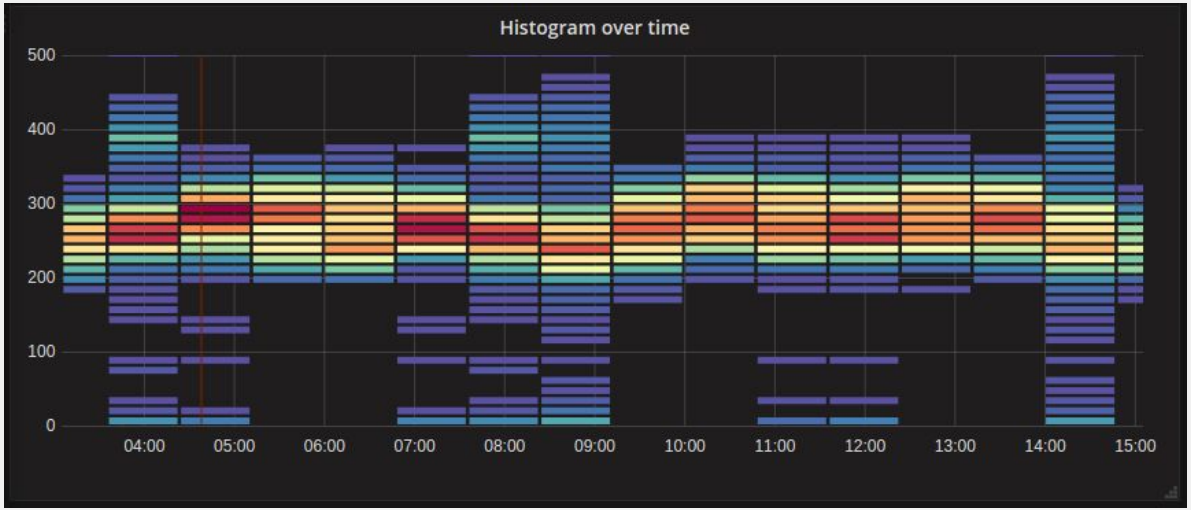


HISTOGRAM

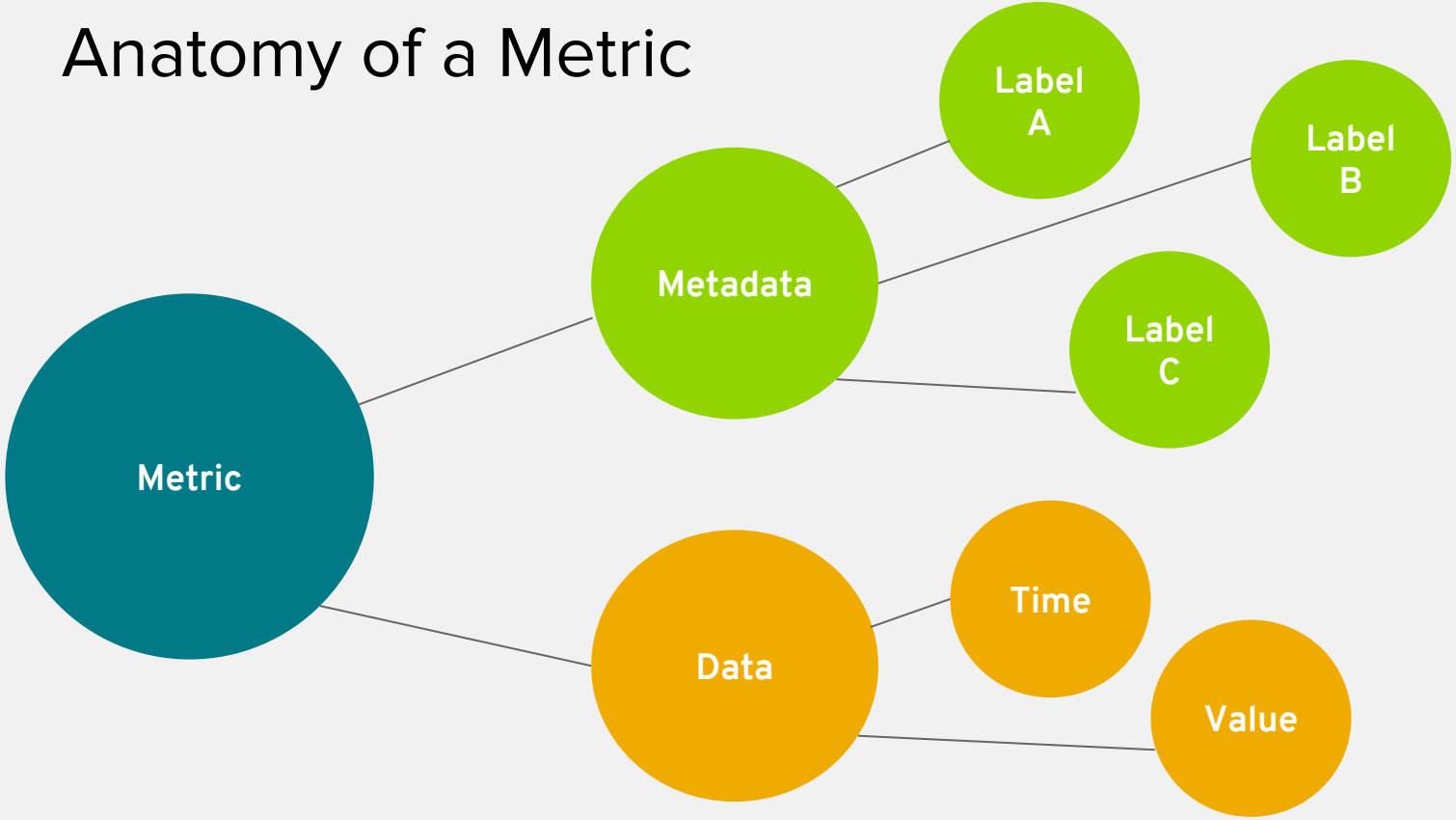
Cumulative

SUMMARY

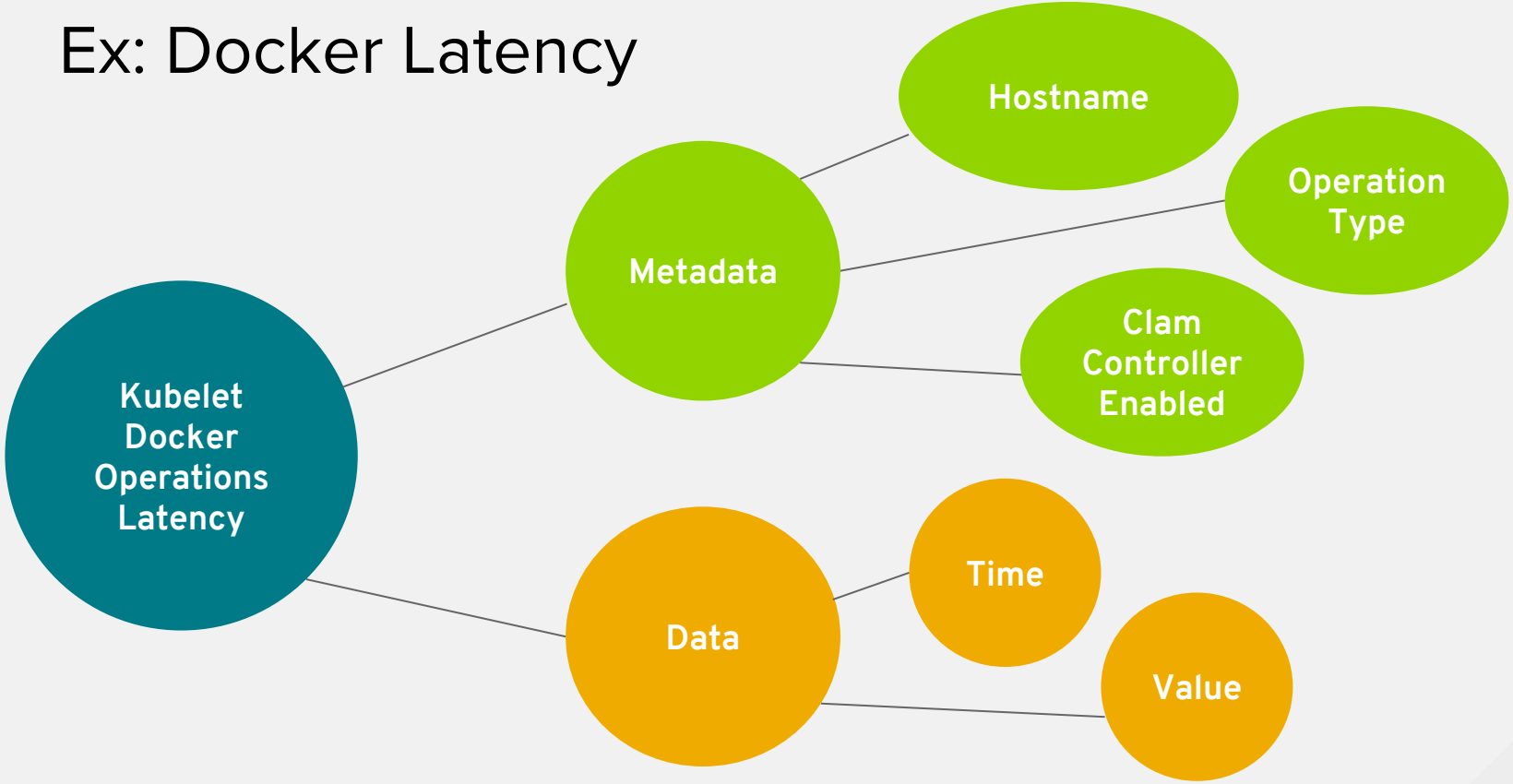
Time Window



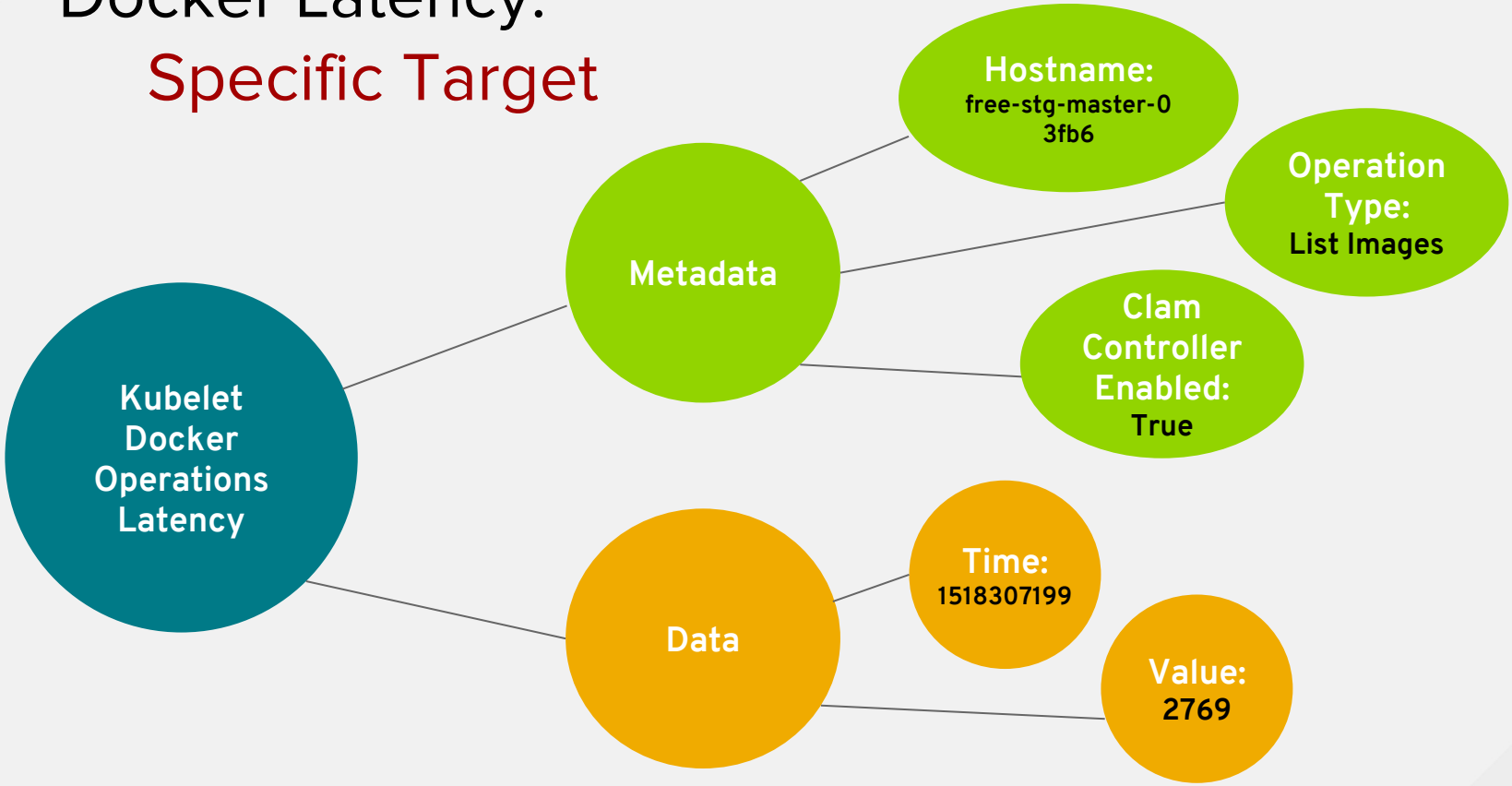
Anatomy of a Metric



Ex: Docker Latency

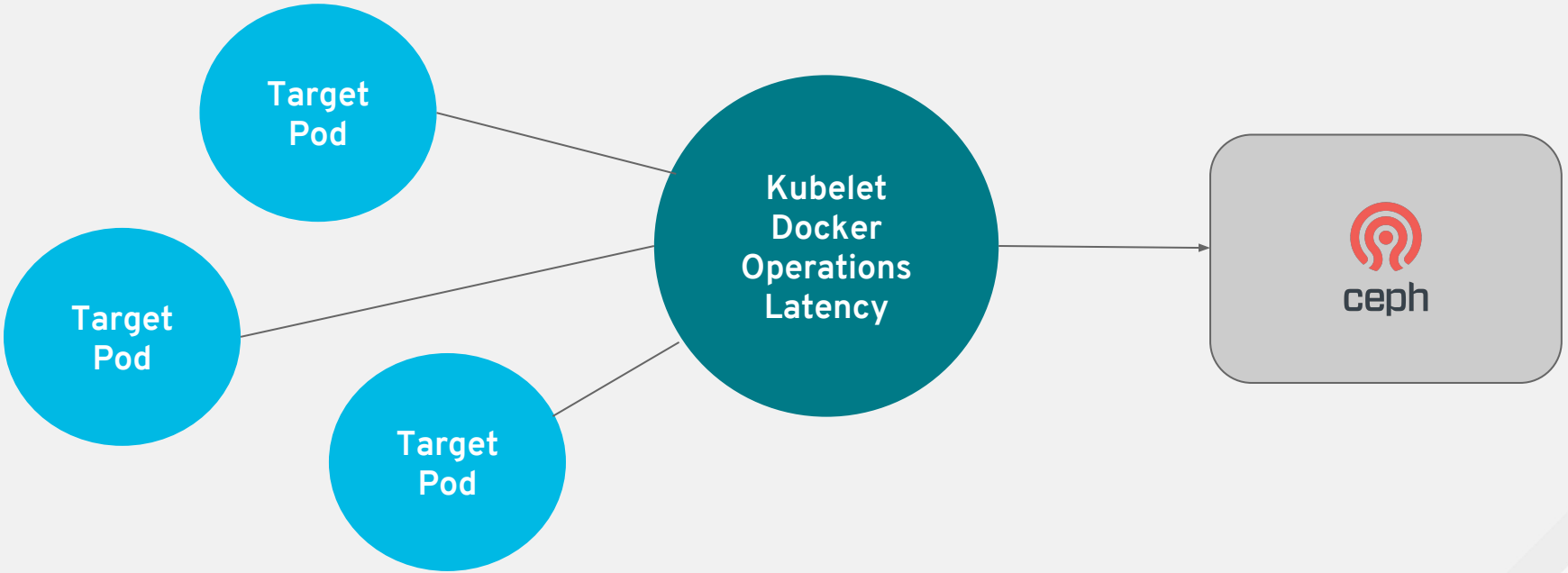


Docker Latency:
Specific Target



Metric Targets

Each Target Corresponds to a Time Series



Data Preprocessing

Data Preprocessing

Raw JSON Format

{"metric": {"__name__": "kubelet_docker_operations_latency_microseconds", "beta_kubernetes_io_arch": "amd64", "beta_kubernetes_io_fluentd_ds_ready": "true", "beta_kubernetes_io_instance_type": "m4.xlarge", "beta_kubernetes_io_os": "linux", "clam_controller_enabled": "True", "failure_domain_beta_kubernetes_io_region": "us-east-2", "failure_domain_beta_kubernetes_io_zone": "us-east-2a", "fluentd_test": "true", "hostname": "free-stg-master-03fb6", "instance": "ip-172-31-78-254.us-east-2.compute.internal", "job": "kubernetes-nodes", "kubernetes_io_hostname": "ip-172-31-78-254.us-east-2.compute.internal", "node_role_kubernetes_io_master": "true", "operation_type": "list_containers", "quantile": "0.99", "region": "us-east-2", "type": "master"},
"values": [[1518307199, "12844"], [1518308638, "13212"], [1518310077, "13830"], [1518311516, "13395"], [1518312955, "16546"], [1518314394, "15174"], [1518315833, "14455"], [1518317272, "12949"], [1518318711, "13439"], [1518320150, "14386"], [1518321589, "12447"], [1518323028, "15947"], [1518324467, "14893"], [1518325906, "14096"], [1518327345, "14735"], [1518328784, "12969"], [1518330223, "14067"], [1518331662, "16286"], [1518333101, "14008"], [1518334540, "12923"], [1518335979, "11888"], [1518337418, "12263"], [1518338857, "11751"], [1518340296, "13534"], [1518341735, "15522"], [1518343174, "14912"], [1518344613, "13461"], [1518346052, "12800"], [1518347491, "15954"], [1518348930, "14826"], [1518350369, "14172"], [1518351808, "13073"], [1518353247, "13810"], [1518354686, "11952"], [1518356125, "15211"], [1518357564, "13696"], [1518359003, "12855"], [1518360442, "13103"], [1518361881, "13125"], [1518363320, "14264"], [1518364759, "12228"], [1518366198, "13045"], [1518367637, "13756"], [1518369076, "14004"], [1518370515, "14946"], [1518371954, "12428"], [1518373393, "12159"], [1518374832, "13614"], [1518376271, "15157"], [1518377710, "14483"], [1518379149, "11738"], [1518380588, "13395"], [1518382027, "14940"], [1518383466, "14253"], [1518384905, "11972"], [1518386344, "13731"], [1518387783, "13236"], [1518389222, "14539"], [1518390661, "12235"], [1518392100, "14209"], [1518393539, "15757"]]}

Data Preprocessing: Spark Dataframe

```
+------+----------------+------------------+------------------+
|values|   operation_type|         timestamp|        log_values|
+------+----------------+------------------+------------------+
| 28654|      list_images|2018-02-08 02:23:53|10.263048328453317|
|  1391|          version|2018-02-08 02:23:53| 7.237778191923443|
|     0|   stop_container|2018-02-08 02:23:53|              null|
|     0|             info|2018-02-08 02:23:53|              null|
|     0| create_container|2018-02-08 02:23:53|              null|
|     0|inspect_container|2018-02-08 02:23:53|              null|
|     0|       pull_image|2018-02-08 02:23:53|              null|
|  1784|    inspect_image|2018-02-08 02:23:53| 7.486613313139955|
|     0| remove_container|2018-02-08 02:23:53|              null|
|     0|  start_container|2018-02-08 02:23:53|              null|
|  5780|  list_containers|2018-02-08 02:23:53| 8.662158961666423|
|     0|       pull_image|2018-02-08 02:47:52|              null|
| 43937|      list_images|2018-02-08 02:47:52|10.690512068687417|
|     0|             info|2018-02-08 02:47:52|              null|
|     0| remove_container|2018-02-08 02:47:52|              null|
|  3061|    inspect_image|2018-02-08 02:47:52| 8.026496938945412|
|  8838|  list_containers|2018-02-08 02:47:52| 9.086815885690685|
|     0| create_container|2018-02-08 02:47:52|              null|
|     0|   stop_container|2018-02-08 02:47:52|              null|
|     0|inspect_container|2018-02-08 02:47:52|              null|
+------+----------------+------------------+------------------+
only showing top 20 rows
```

Basic Statistics and Transformations

Mean

Variance

Standard Deviation

Median

Minimum Value

Maximum Value

rhmax

rhmax(data point) = data point/max(all data)

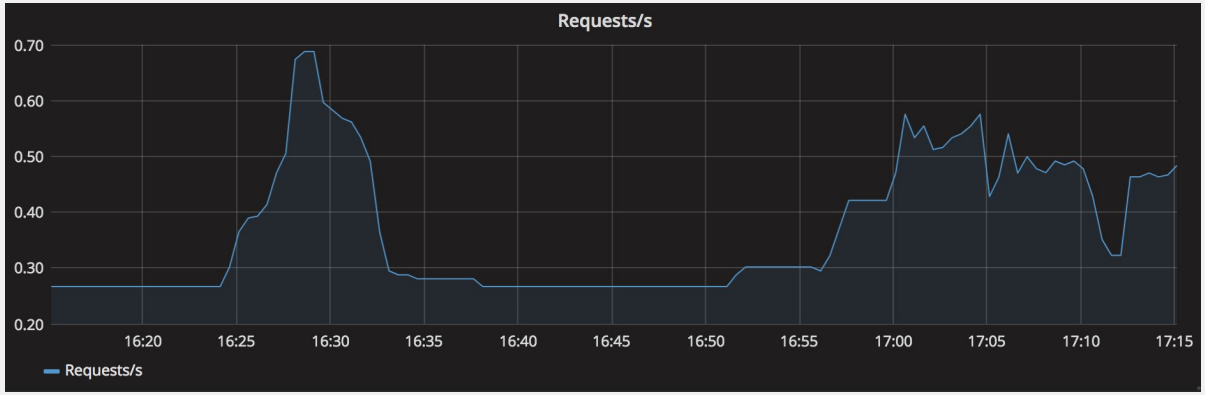
delta

delta(data point) = data point - previous data point

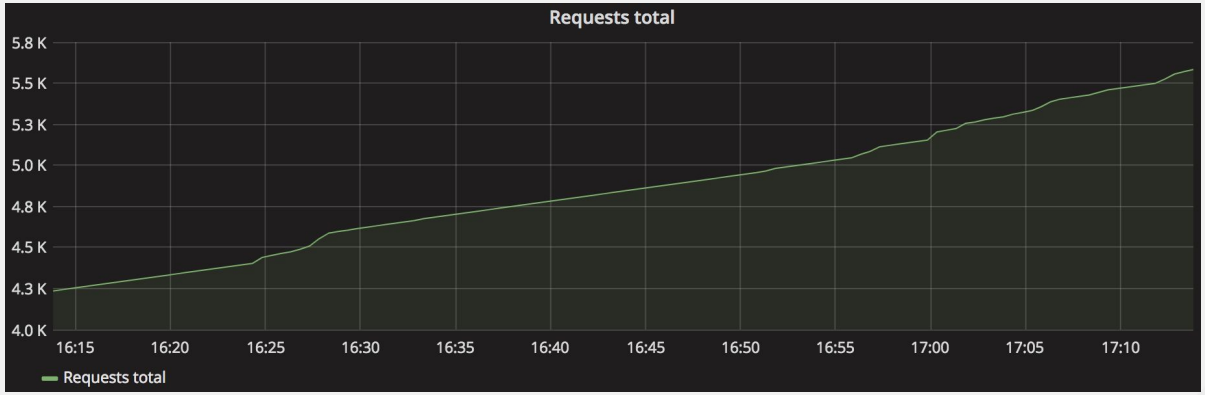
Anomaly Detection and Predictions

Single Value Metric Anomaly Detection

GAUGE



COUNTER



Single Value Metric Anomaly Detection

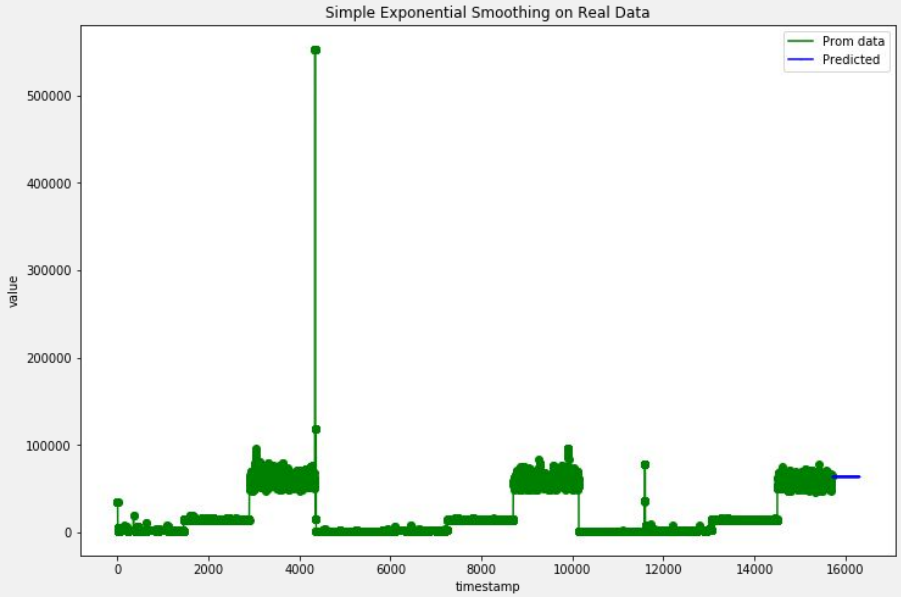
- Monotonically Increasing?
- Boolean Valued?
- Volatile?
- Does the data have inherent bounds?
- Does the moving average change over time?
- How often does a counter typically reset?

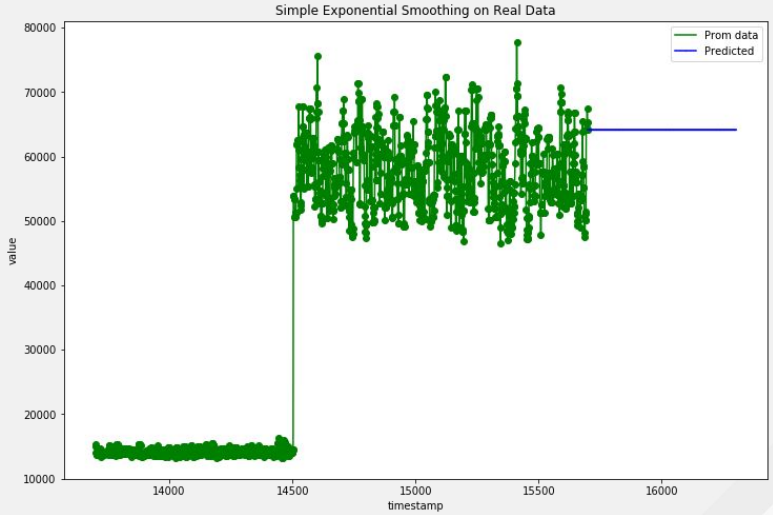
Data Science for Anomaly Detection

- Exponential Smoothing
  - Simple
  - Holt Winters (Triple)
- ARIMA (Autoregressive Integrated Moving Average)
- Prophet Modelling
- RNNs (Recursive Neural Networks) → Subhojit's work

Single Exponential Smoothing

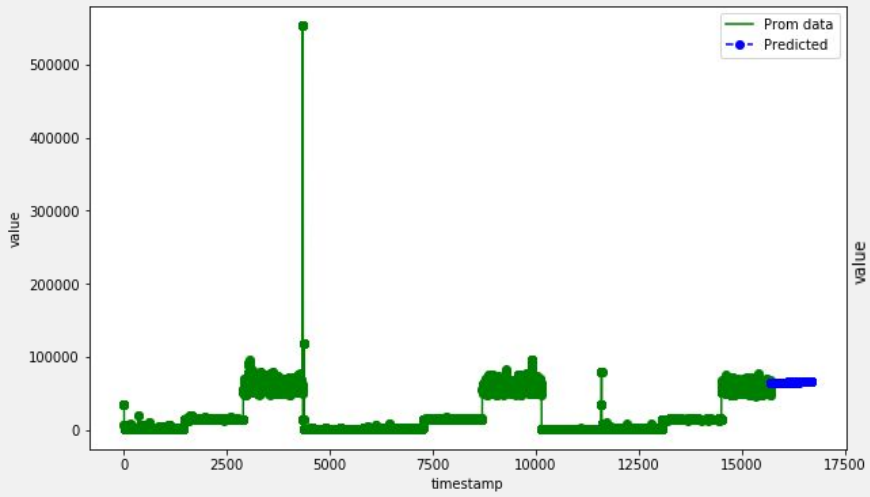
kubelet docker operations latency microseconds

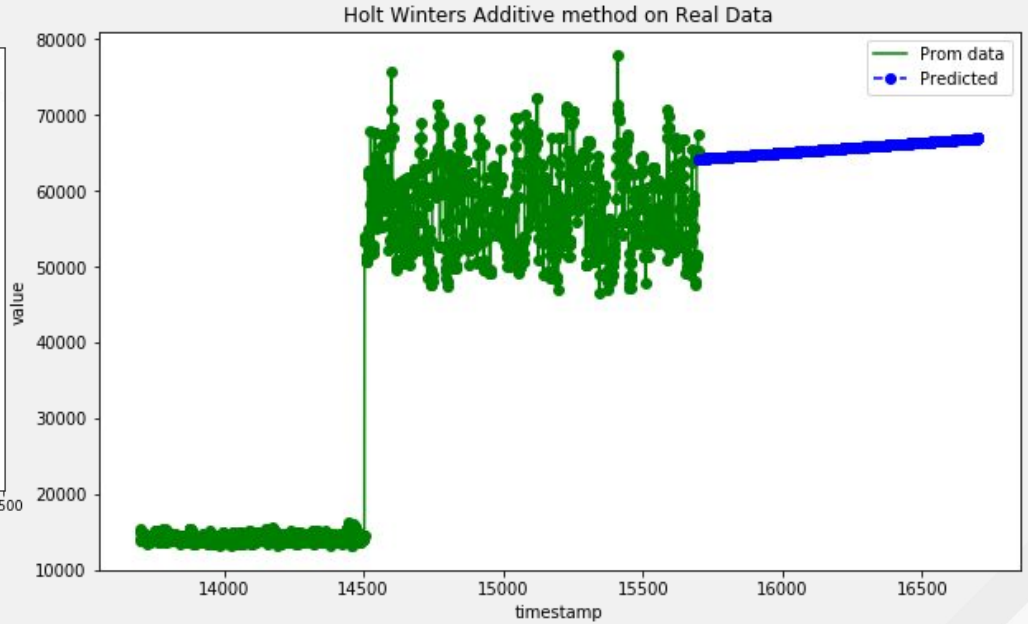




Triple Exponential Smoothing (Holt Winters)

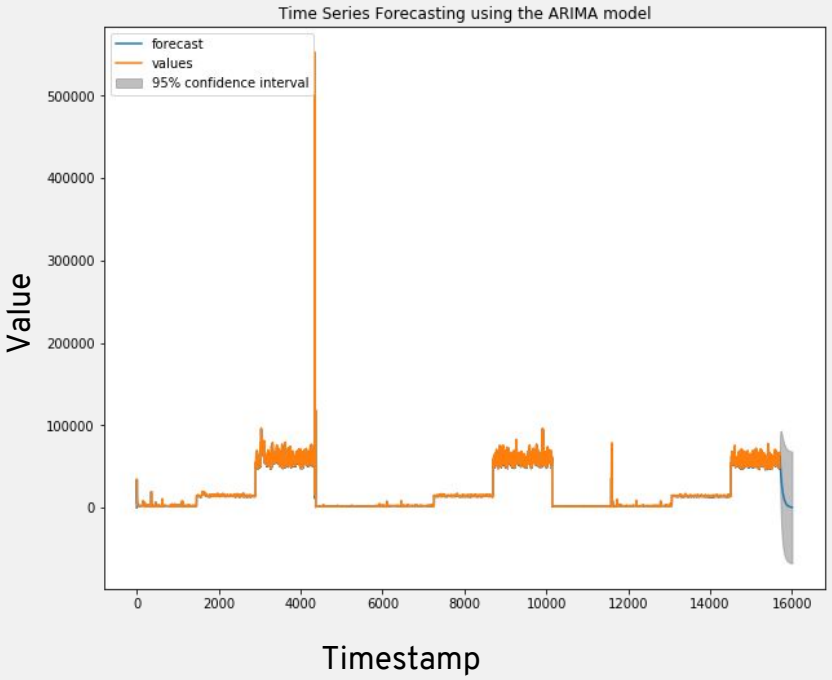
kubelet docker operations latency microseconds

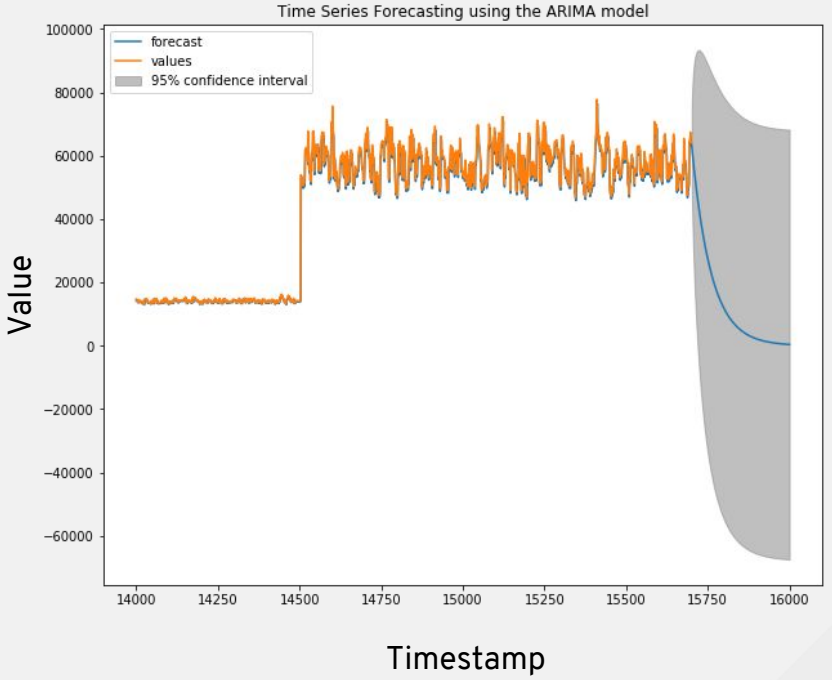




The ARIMA Model

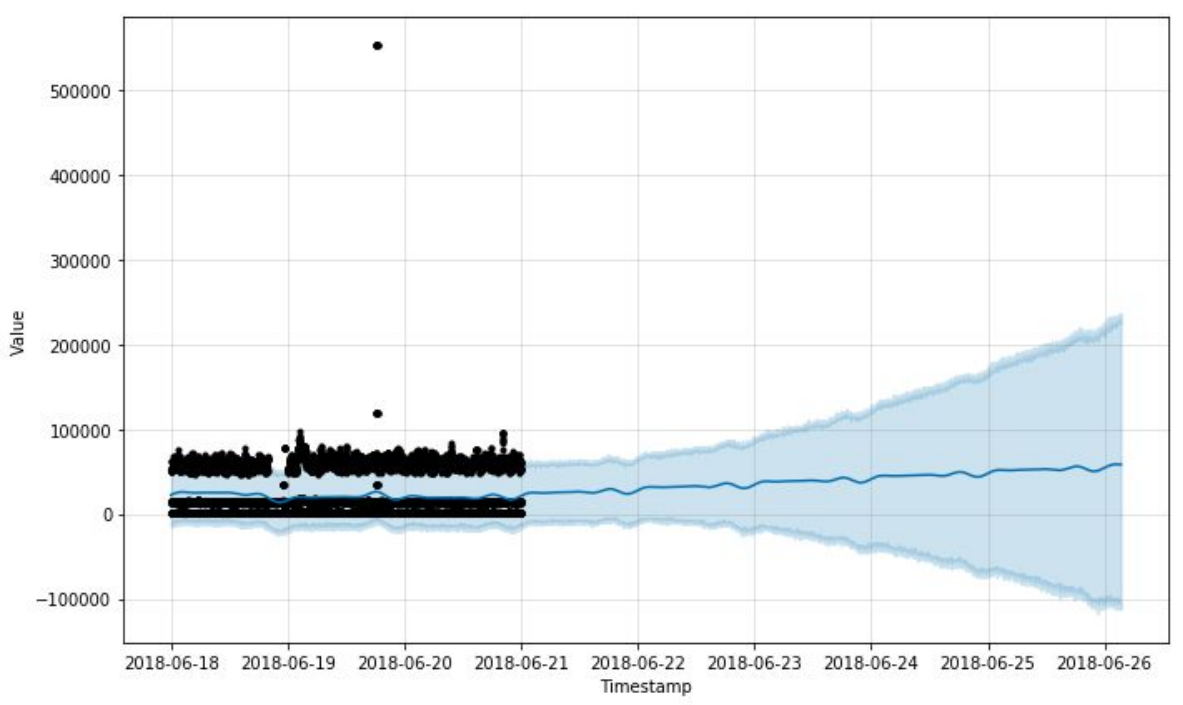
kubelet docker operations latency microseconds





Prophet Modelling

kubelet docker operations latency microseconds



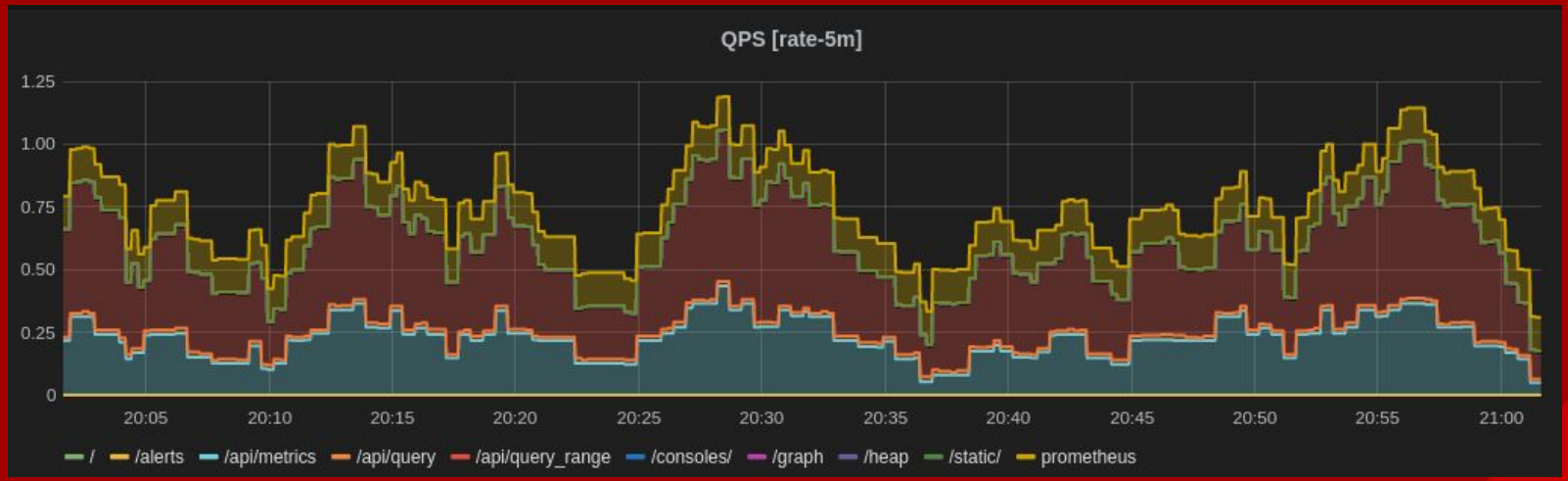
Challenges with Prometheus Dataset

- **Data comes from multiple sources**
  - Need to explore correct time series filtering
- **Data has trend and season**
  - Leverage known smoothing and decomposition techniques
- **Wide range of metric types and behavior**
  - Possibly apply different AD techniques for different series
- **Training Data has hidden anomalies and dropouts**
  - Find a way to accurately prepare historical data for training

Future Work

- Explore Histogram and Summary Metrics
- Analyze Metric Metadata
- Design Anomaly Detection for these additional metric components

THANK YOU



redhat.