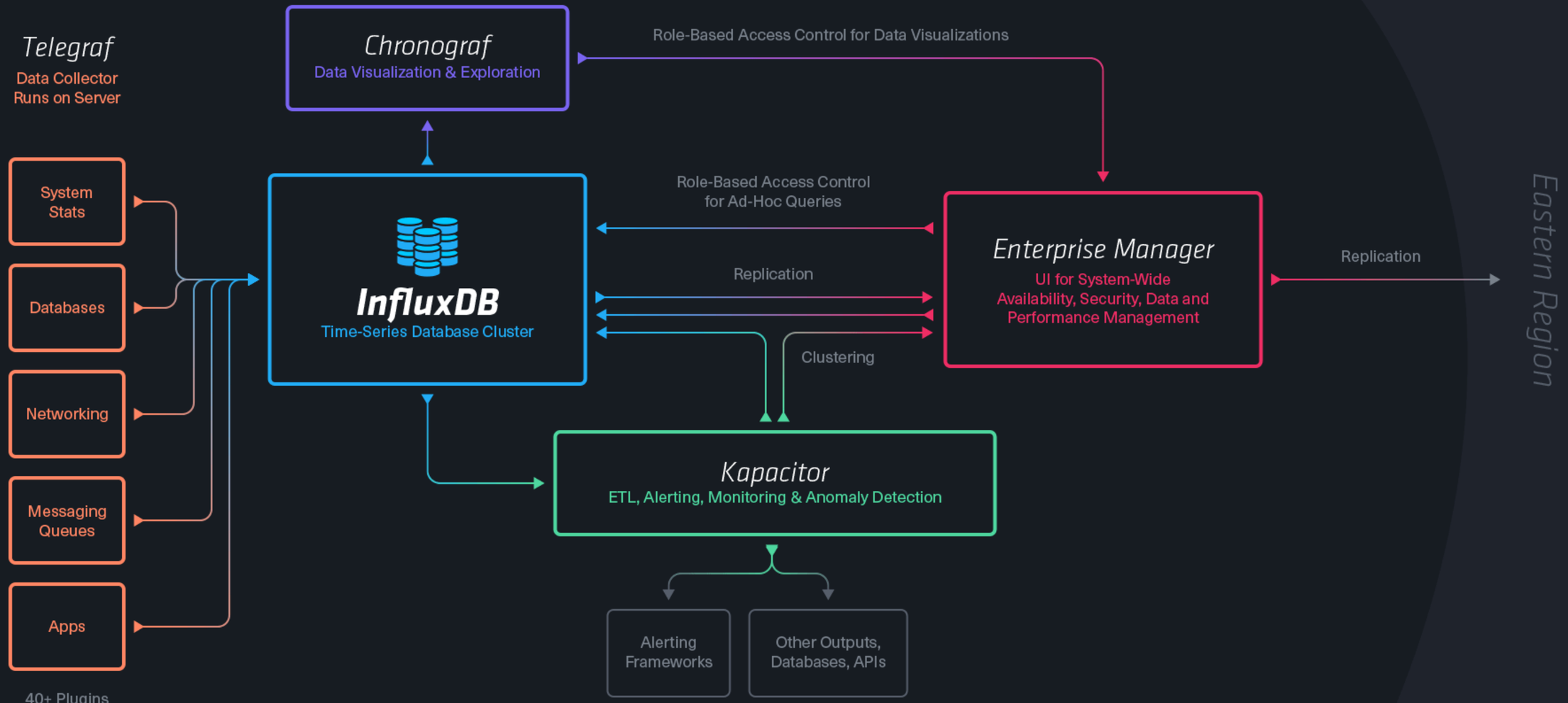
INFLUXDB

TIME SERIES DATABASE

Western Region



USE CASES

DATA MODEL

- String
- Integer
- Floats
- Boolean

```
influxd -config /usr/local/etc/influxdb.conf
```

Write Data Documentation Database: mydb

Query: select mean(value) from cpu where time >= '2015-12-29 10:04:40' AND time < '2015-12-29 10:04:59.0' group by time(10s)

Query Templates

cpu

time	mean
2015-12-29T10:04:40Z	0.5466666666666666
2015-12-29T10:04:50Z	1

QUERY LANGUAGE

```
SELECT
  COUNT(duration) as count_duration,
  MIN(duration) as min,
  MAX(duration) as max,
  MEAN(duration) as MEAN
FROM events
WHERE time > now() - 1h
GROUP BY time(30s)
```

```
INSERT cpu_load,server_name=gilbert value=2
```

```
curl -i -XPOST 'http://localhost:8086/write?db=mydb'
  --data-binary 'cpu_load,server_name=gilbert value=2'
```

RETENTION

```
CREATE RETENTION POLICY two_hours
  ON food_data DURATION 2h REPLICATION 1 DEFAULT
```

DOWNSAMPLING

```
CREATE CONTINUOUS QUERY cq_30m ON food_data
BEGIN
  SELECT mean(temperature) AS mean_temperature,
         mean(delivery_time) AS mean_delivery_time
  INTO food_data."default".downsampled_orders
  FROM orders
  GROUP BY time(30m)
END
```

TELEGRAF

INPUT PLUGINS

aerospike, apache, bcache, couchdb, disque, dns query time, docker, dovecot, elasticsearch, exec, haproxy, httpjson, influxdb, jolokia, leofs, lustre2, mailchimp, memcached, mesos, mongodb, mysql, net_response, nginx, nsq, phpfpm, phusion passenger, ping, postgresql, powerdns, procstat, prometheus, puppetagent, rabbitmq, raindrops, redis, rethinkdb, riak, sensors, snmp, sql server, twemproxy, zfs, zookeeper, win_perf_counters, system, cpu, mem, net, netstat, disk, diskio, swap, statsd, mqtt_consumer, kafka_consumer, nats_consumer, github_webhooks

OUTPUT PLUGINS

influxdb, amon, amqp, aws kinesis,
aws cloudwatch, datadog, graphite,
kafka, librato, mqtt, nsq, opentsdb,
prometheus, riemann

```
telegraf -sample-config
  -input-filter cpu
  -output-filter influxdb
  > telegraf.conf
```

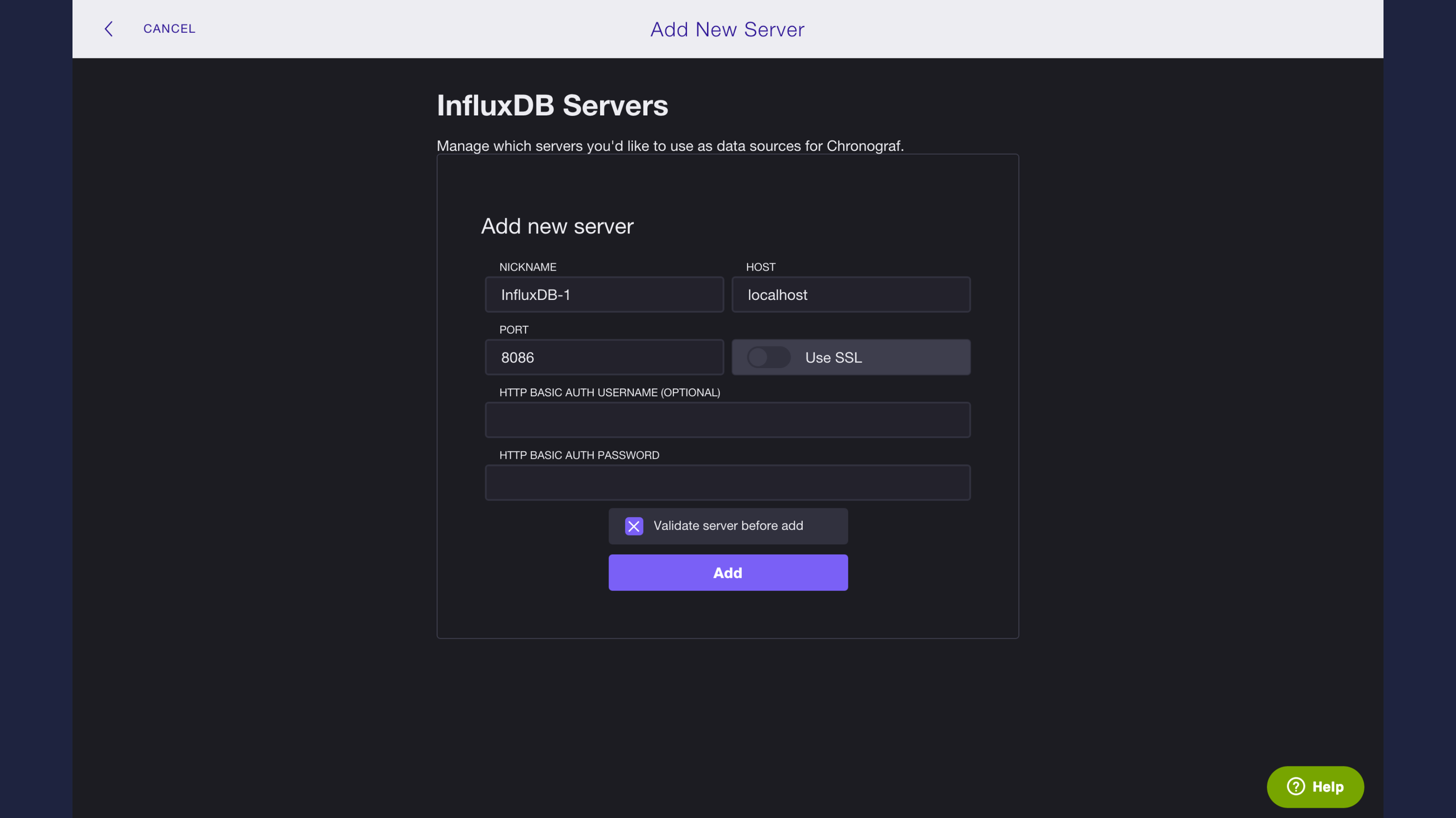
```
[[outputs.influxdb]]
  urls = ["http://localhost:8086"]
  database = "telegraf"
  precision = "s"
```

```
[[inputs.cpu]]
  percpu = true
  totalcpu = true
  drop = ["time_*"]
```

```
telegraf -config telegraf.conf
```

CHRONOGRAF

chronograf


CANCEL
Add New Server
InfluxDB Servers
Manage which servers you'd like to use as data sources for Chronograf.
Add new server
NICKNAME
InfluxDB-1
HOST
localhost
PORT
8086
Use SSL
HTTP BASIC AUTH USERNAME (OPTIONAL)
HTTP BASIC AUTH PASSWORD
Validate server before add
Add
Help

CANCEL

Average idle CPU usage

DONE

Auto Refresh: None

Past 15 minutes

0

BUILDER

SELECT value FROM "measurement" WHERE tmpltime()

FILTER BY

measuren

8

Select a ta

Servers

InfluxDB-1

Databases

telegraf

Retention Policies

default

Make Default

Apply

EXTRACT BY

value

BY

Select a function

Select a field

BY

GROUP BY

+ ADD QUER

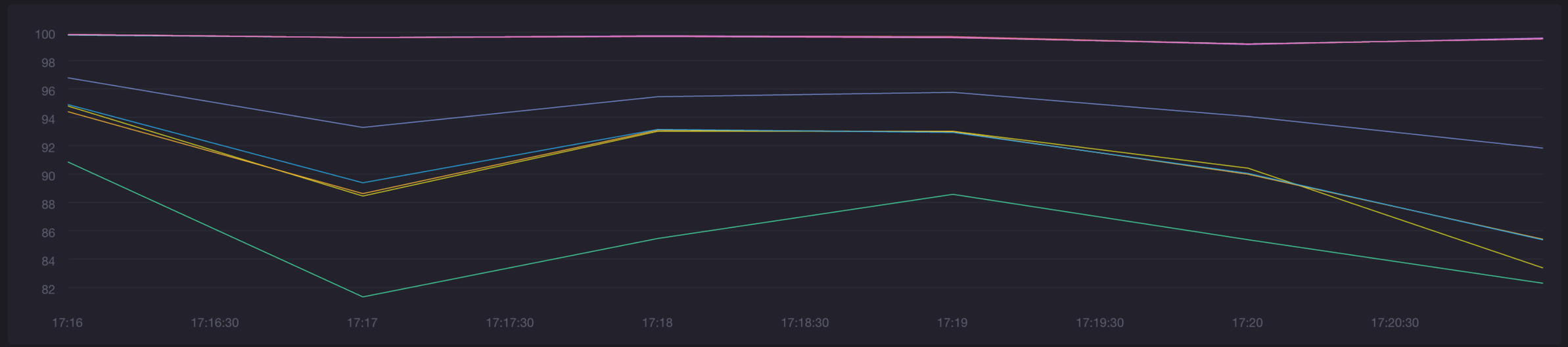
CANCEL

Average idle CPU usage

DONE

Auto Refresh: None

Use `WHERE tmpltime()` in a query to enable time range controls.



BUILDER

```
SELECT mean("usage_idle") FROM "telegraf"."default"."cpu" WHERE time > now() - 5m GROUP BY time(1m), "cpu"
```

FILTER BY

cpu

6

Select a tag key

EXTRACT BY

usage_idle BY mean

Select a field BY Select a function

GROUP BY

cpu 1m

Select a Tag

+ ADD QUERY

Idle CPU usage

Add Visualization

Add From Existing Visualizations

Name this Visualization

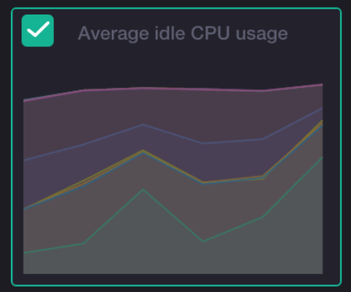
Cancel

Save

CANCEL

Select Visualizations

Add Visualizations to Dash



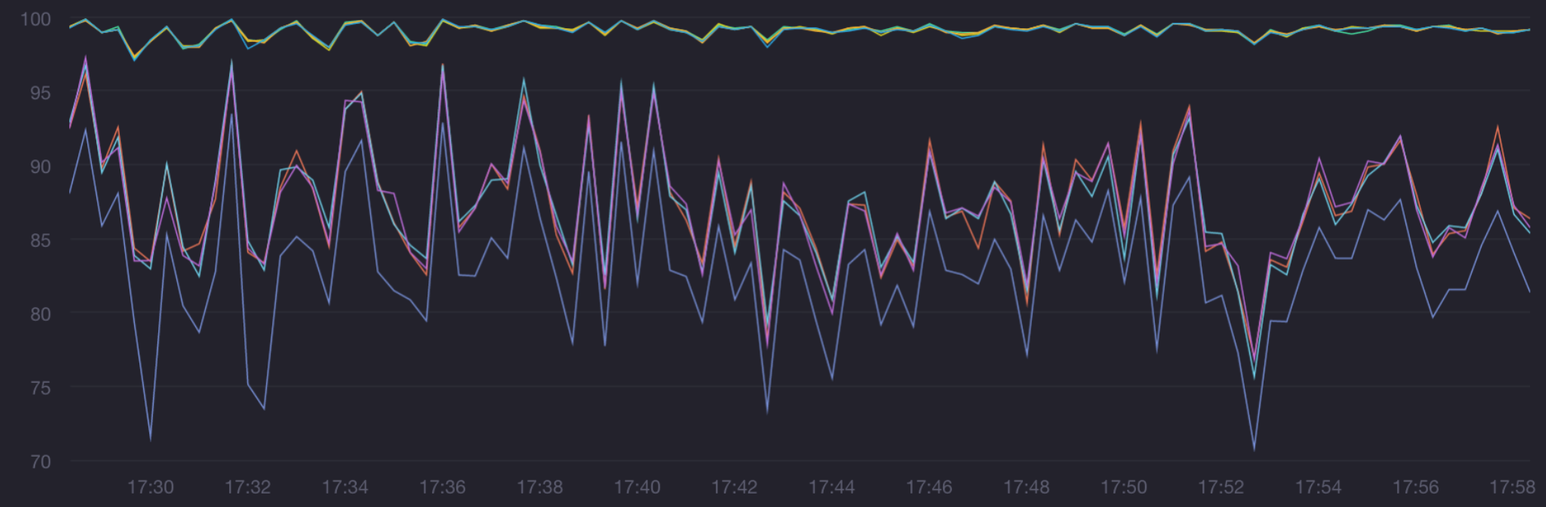
Idle CPU usage

Add Visualization

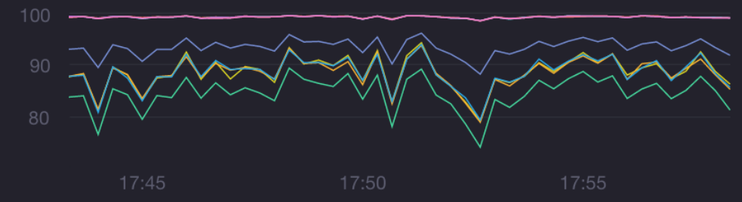
Auto Refresh: None

Past 5 minutes

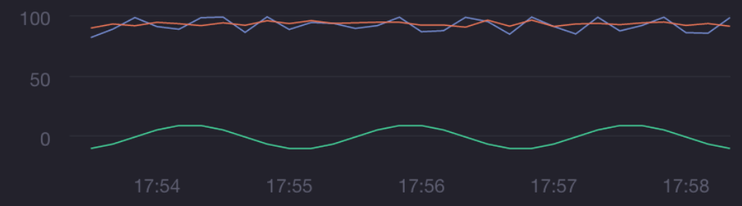
CPU system



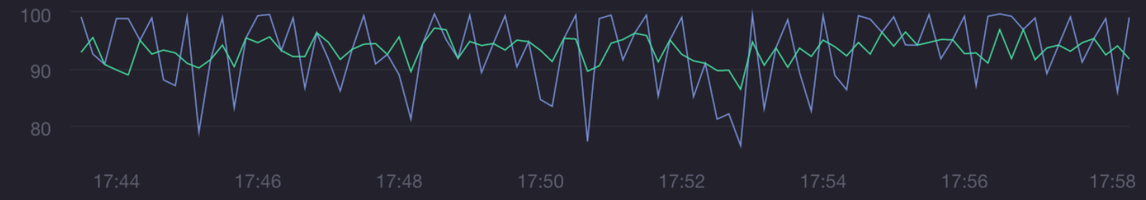
Average idle CPU usage



Random



CPU-0



CPU user

GitLab Performance Monitoring: a monitoring system using InfluxDB [...] then visualized using Grafana

– Gitlab Performance Blogpost

KAPACITOR

```
kapacitord config > kapacitor.conf
```

```
kapacitord -config kapacitor.conf
```

CPU_ALERT.TICK

```
stream
    // Select just the cpu measurement
    .from().measurement('cpu')
    .alert()
        .crit(lambda: "usage_idle" <  70)
        // Whenever we get an alert write it to a file.
        .log('/tmp/alerts.log')
```

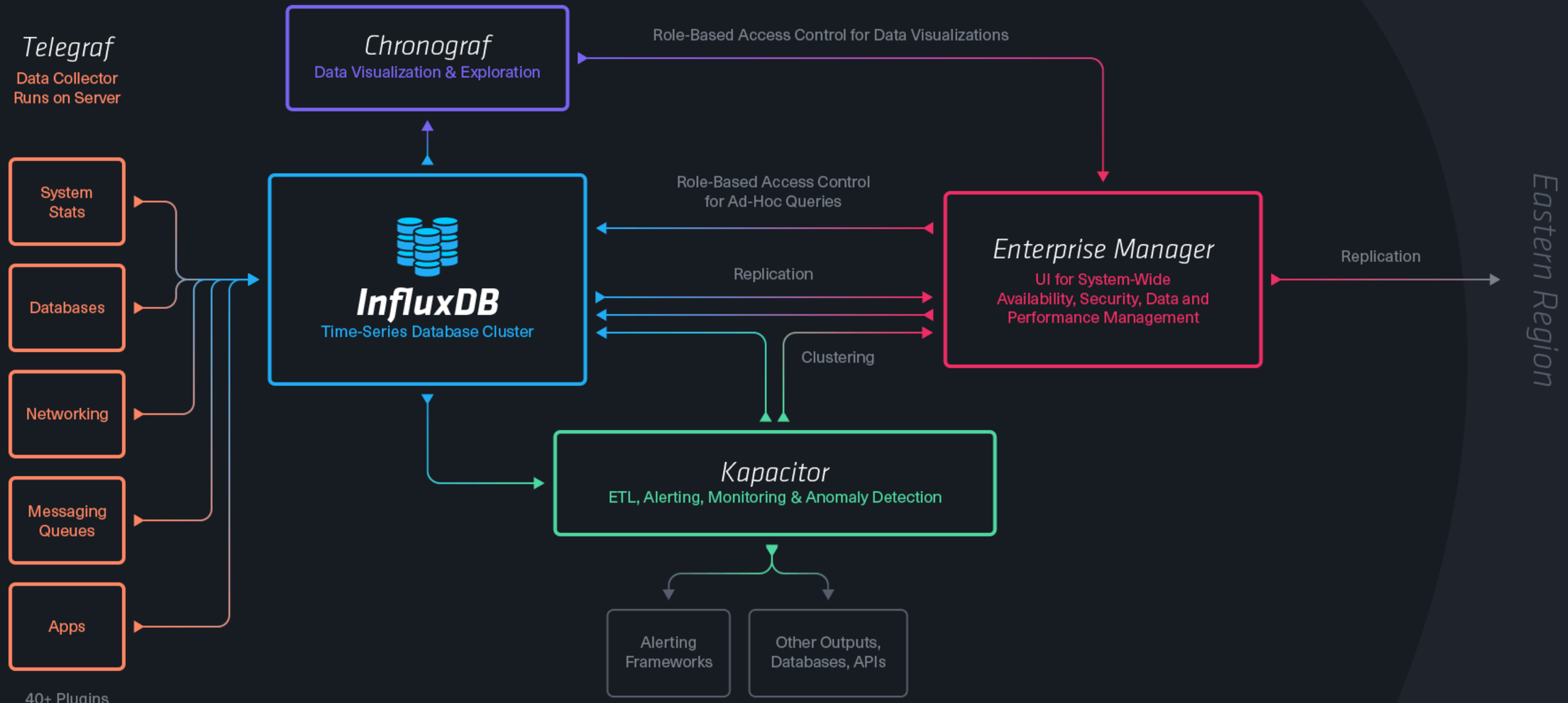
```
kapacitor define \
    -name cpu_alert \
    -type stream \
    -tick cpu_alert.tick \
    -dbrp kapacitor_example.default
```

```
kapacitor enable cpu_alert
```

```
stream
    .from().measurement('cpu')
    .alert()
        // Compare values to running mean and standard deviation
        .crit(lambda: sigma("usage_idle") > 3)
        .log('/tmp/alerts.log')
```

```
stream
    .from().measurement('cpu_usage_idle')
    .groupBy('host')
    .window().period(1m).every(1m)
    .mapReduce(influxql.mean('value'))
    .eval(lambda: 100.0 - "mean").as('used')
    .alert()
        .message('{{ .Level}}: {{ .Name }}/{{ index .Tags "host" }} has ' +
                 'high cpu usage: {{ index .Fields "used" }}')
        .warn(lambda: "used" > 70.0)
        .crit(lambda: "used" > 85.0)
        .slack().channel('#alerts')
        .pagerDuty()
```

Western Region



Bodo Tasche
@bitboxer

– CTO bitcrowd

CREDITS

* Time Series Image CC-BY-2.0 Ian Sane https://www.flickr.com/photos/31246066@N04/5261957053