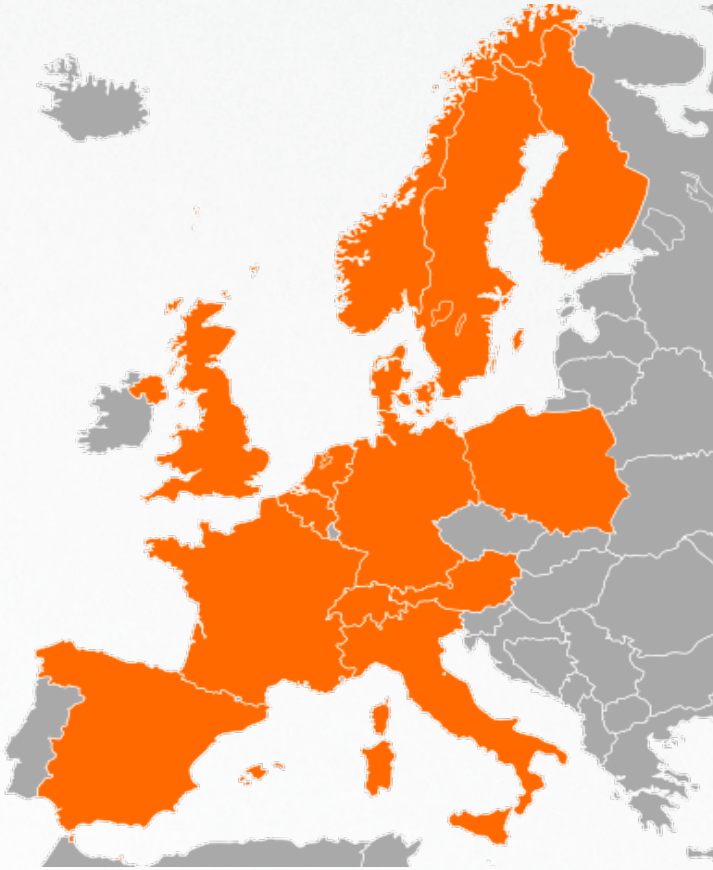
zalando

ZMON - Monitoring Our Platform

DevOps Meetup Dublin | September 3, 2015 | jan.mussler@zalando.de | @JanMussler

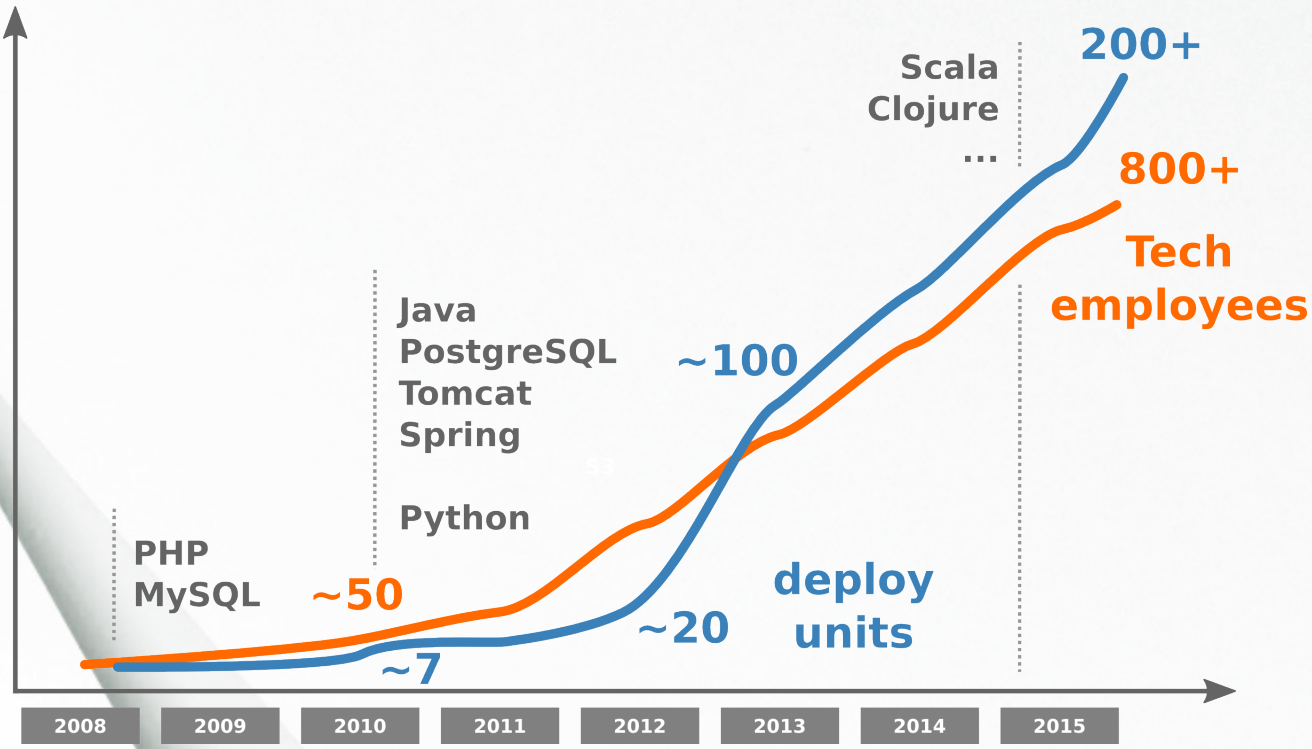
ONE of EUROPE'S LARGEST ONLINE FASHION RETAILERS

15 countries
3 fulfillment centers
16+ million active customers
2.2+ billion € revenue 2014
130+ million visits per month
8.000+ employees

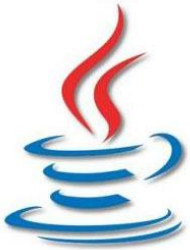


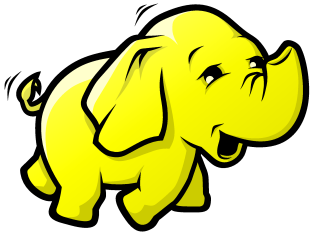
Visit us: tech.zalando.com

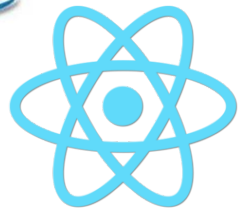
Zalando's Technology History



(Some!) Technologies We Use







Monitoring Situation Until Late 2013

ICINGA plus custom frontend (ZMON 1)

Did not scale with growth:

- Our UI became too slow
- Number of systems to check too many
- Number of teams that wanted checks grew
- Every request had to go through single team

Goals of new ZMON development

Improve performance and throughput

Autonomy for individual teams

Flexibility and extendability

Integration into tooling (CMDB, DeployCtl …)

The basic terminology ...

Entity:

Anything you may want to monitor

Can be used as a "dimension"

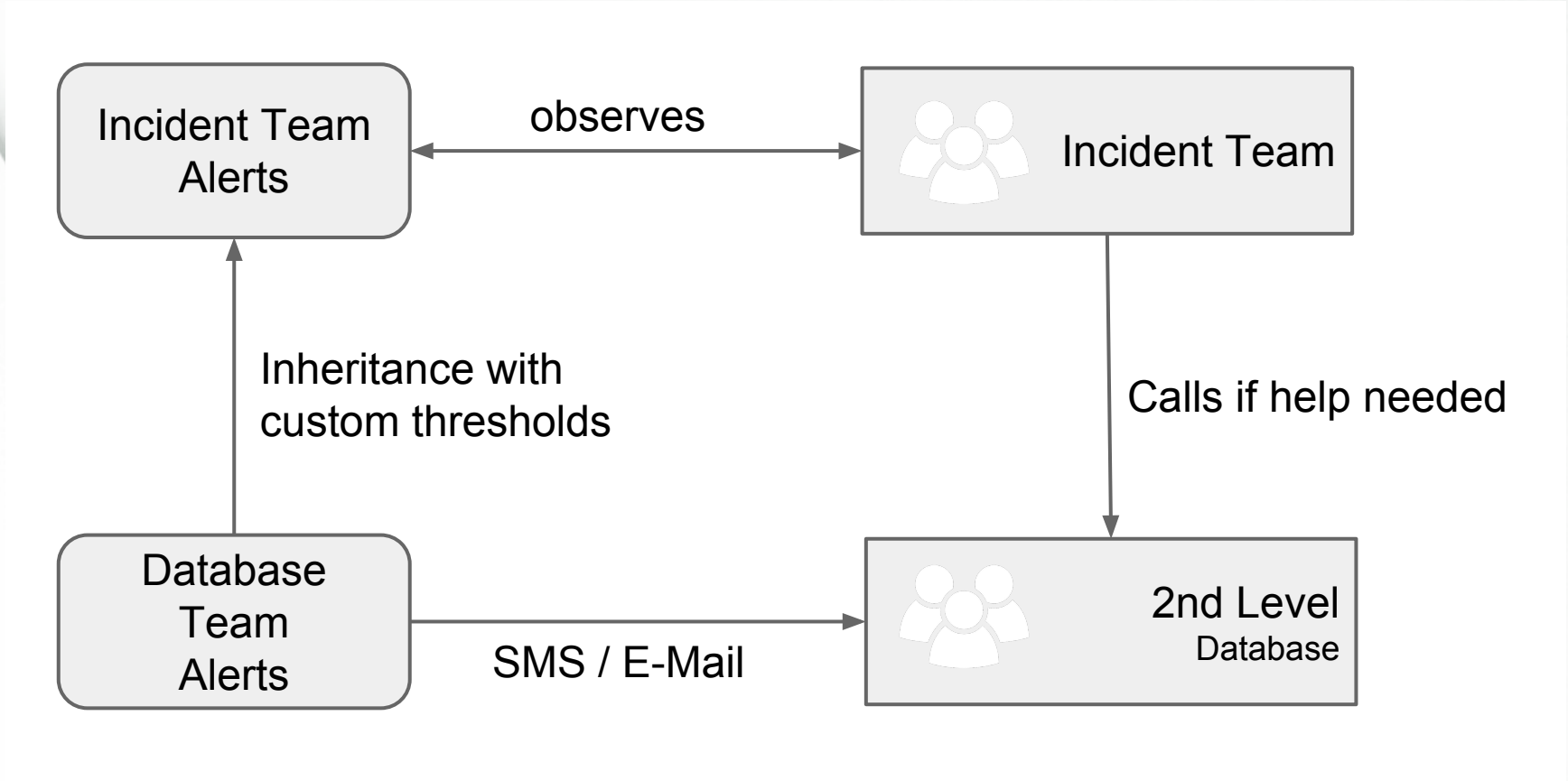
Checks:

Runnable Python snippet fetching data

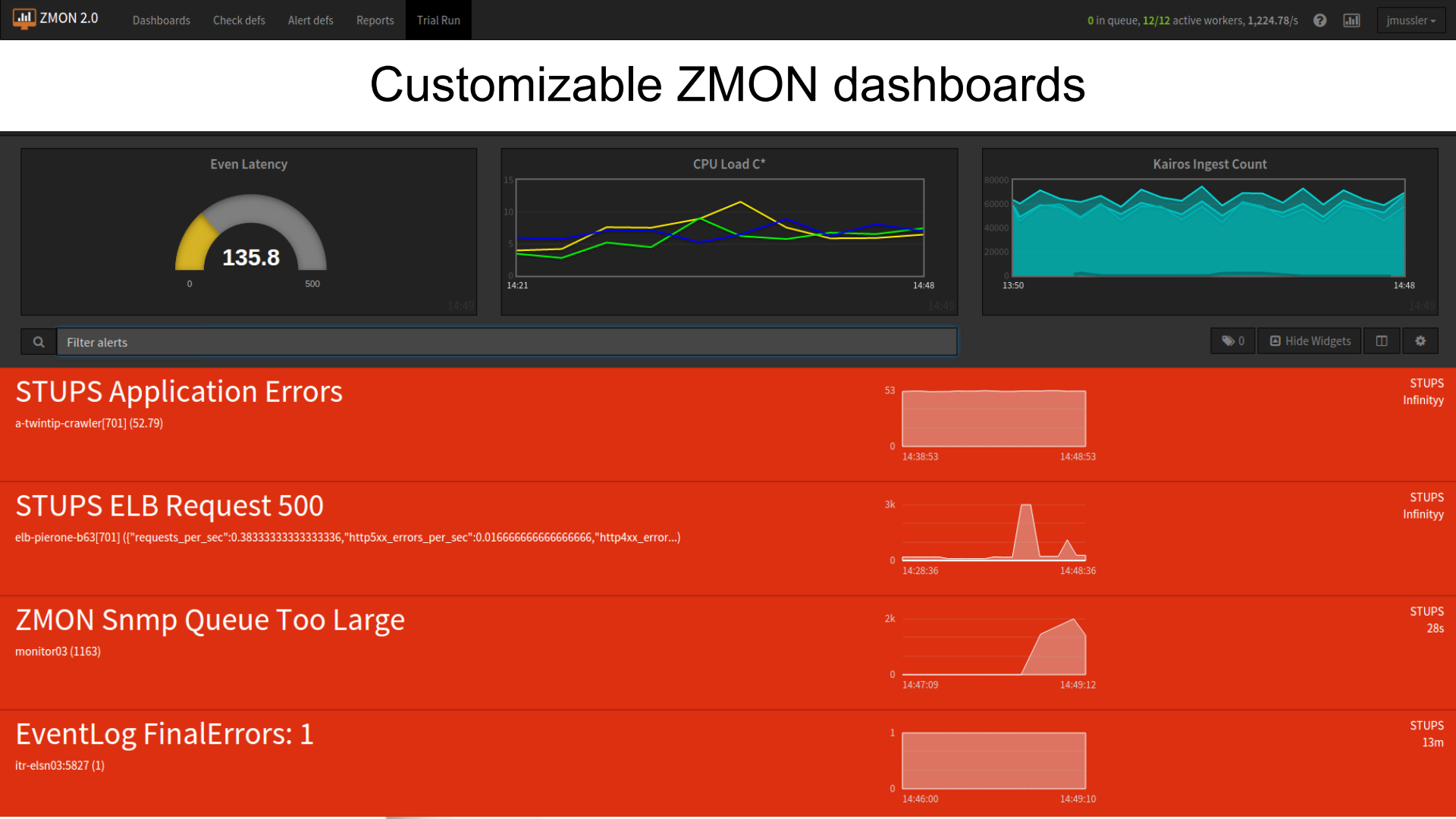
Alert on Check:

Python expression yielding true or false

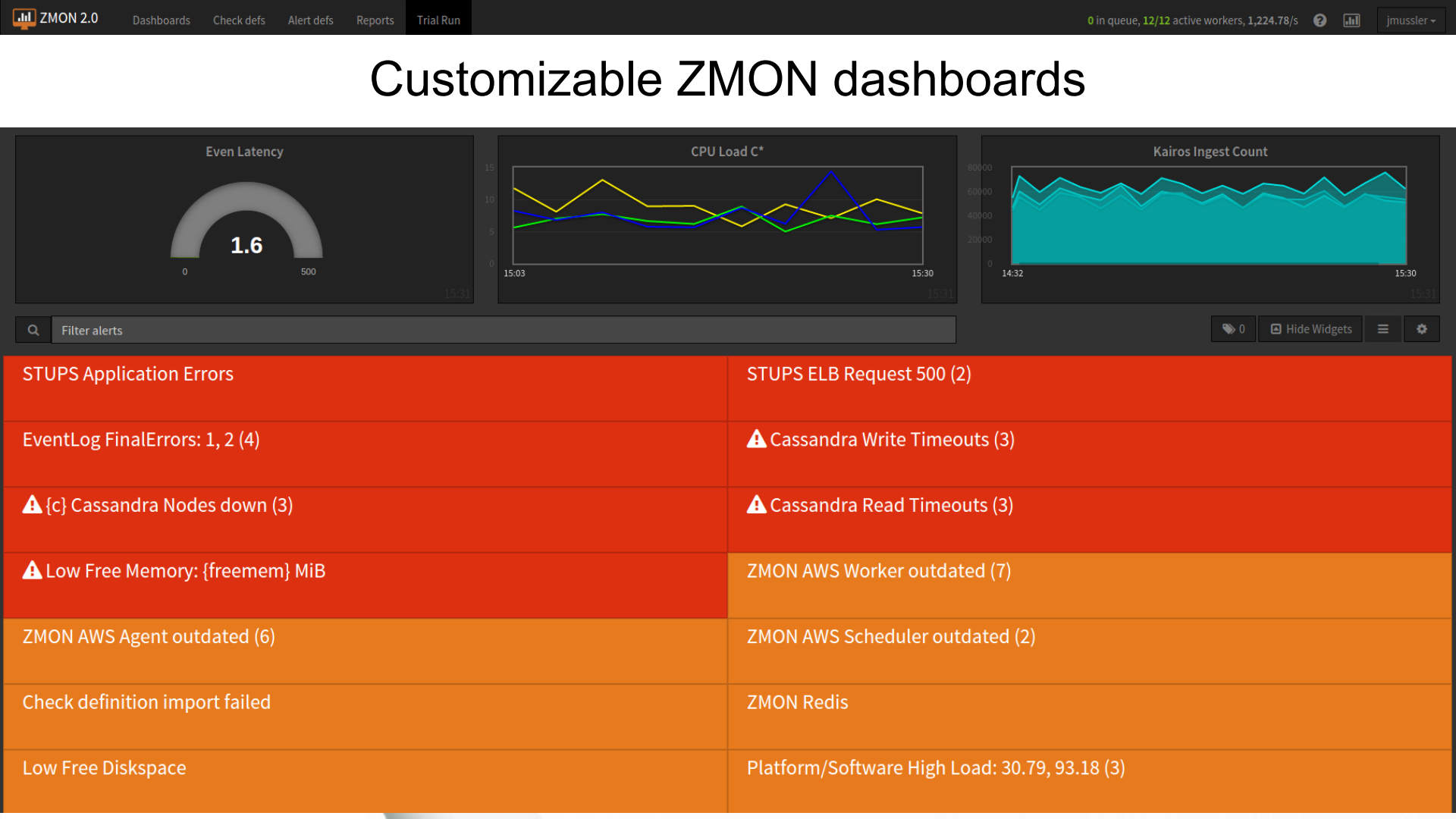
Zalando Tech - 24x7 team setup



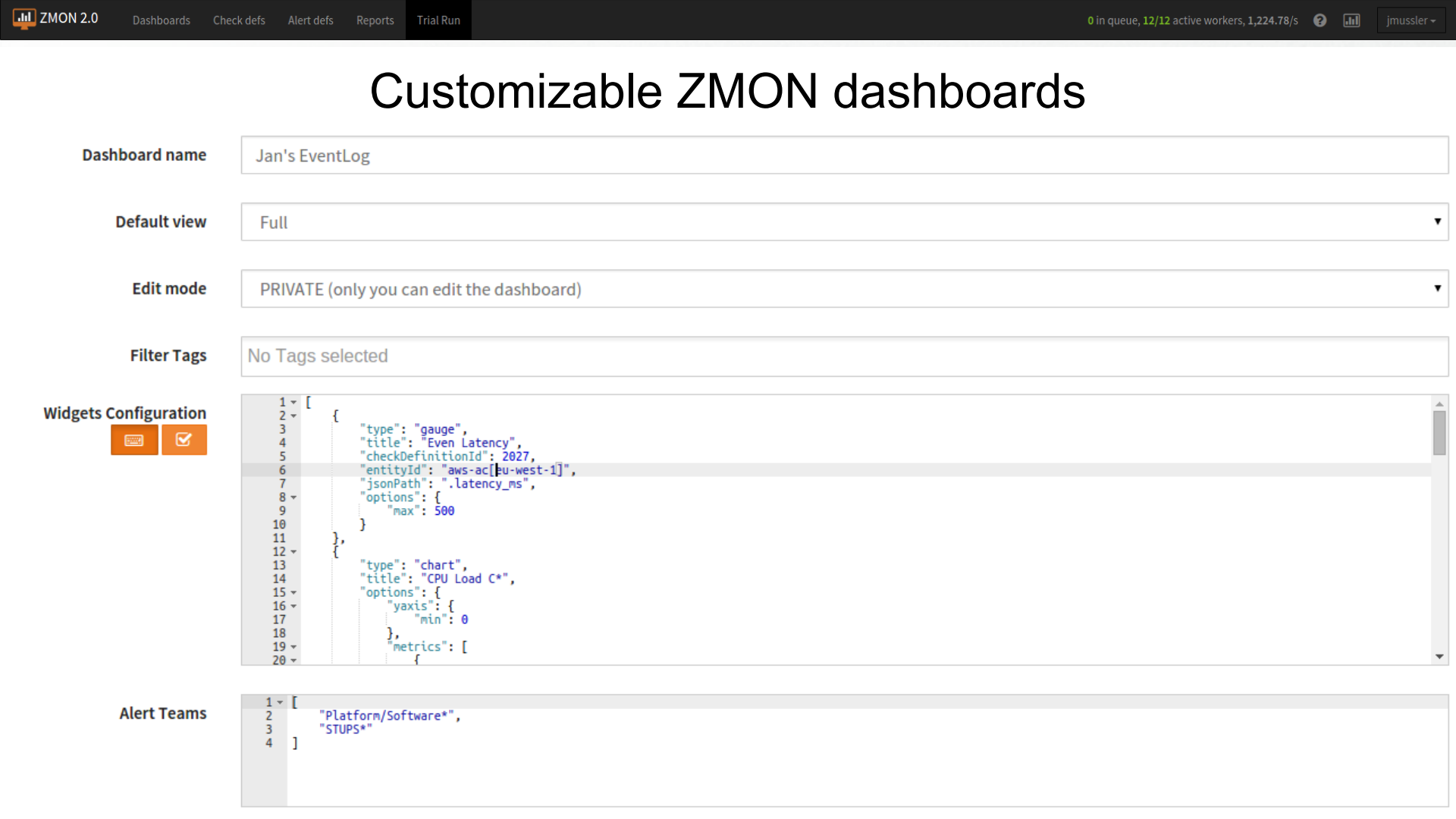
Customizable ZMON dashboards



Customizable ZMON dashboards

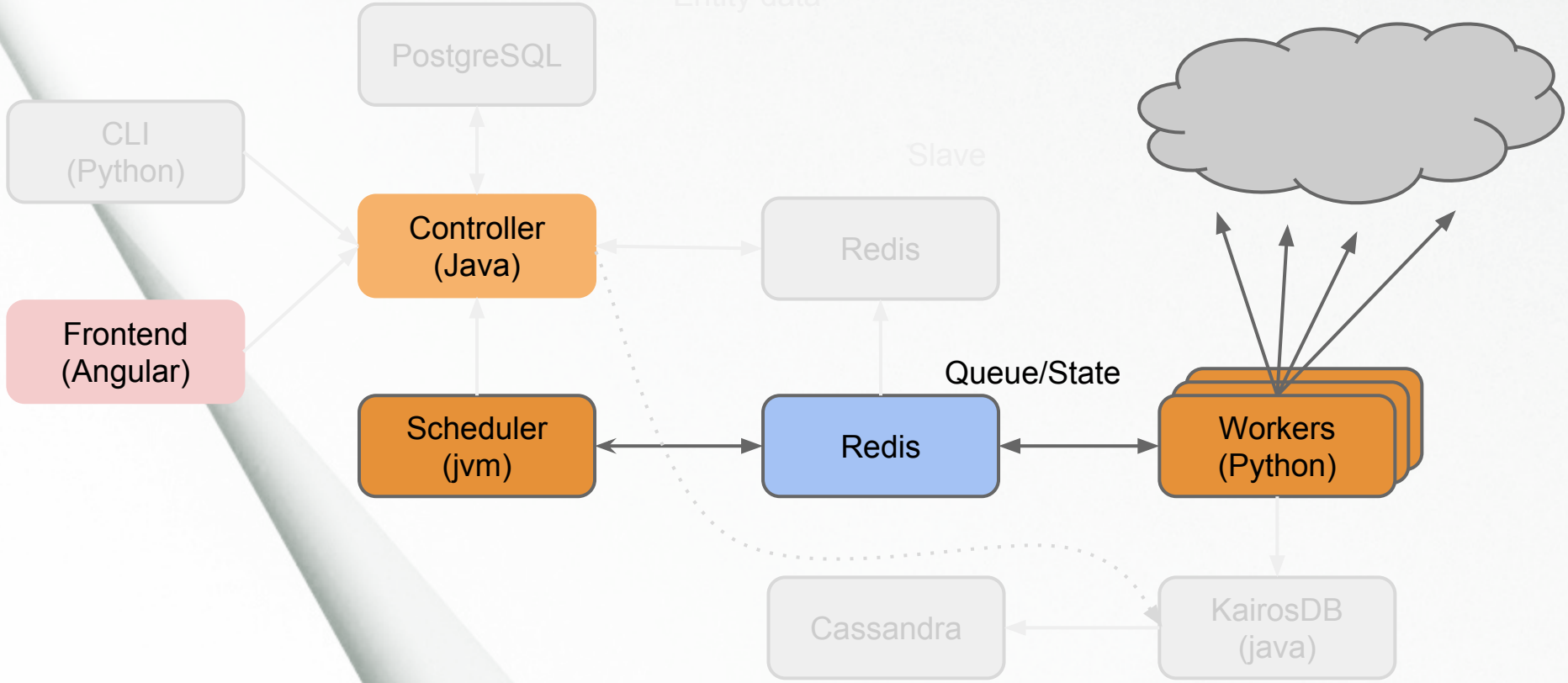


Customizable ZMON dashboards



Display historic data using Grafana

ZMON's core components



Entities

- hosts, databases, applications, instances ...
- generic key value object
- 4000+ entities in our deployment

Entity "node01:8080"

```
{
 "id": "node01:8080",
 "type": "instance",
 "host": "node01",
 "ports": {"8080":8080,"8181":8181},
 "application_id": "zmon",
 "application_version": "0.1.0",
 "dc":"dc1"
}
```

Database Entity

Entity: customer-live-slave

```
{
 "id": "customer-live-slave",
 "type": "database",
 "role": "slave",
 "environment": "live",
 "shards": {
   "customer1": "customer1.db:5432/customer1"
   "customer2": "customer2.db:5432/customer2"
   "customer3": "customer3.db:5432/customer3"
   "customer4": "customer4.db:5432/customer4"
 }
}
```

Entity Service

Integrated easy-to-use entity store with REST API

```
>zmon entities push local-postgres.yaml
```

local-postgres.yaml

```
id: localhost:5432
type: postgres
host: localhost
port: 5432
shards:
  local_zmon_db: "localhost:5432/local_zmon_db"
```

Checks

- select subset of entities
- executes Python expression
  - powerful using **eval** with custom context
  - Builtins: HTTP, PostgreSQL, MySQL, Cloudwatch, Redis, SNMP, tcp, SOAP, Scalyr...
- returns "value" object
  - Quickly, every check returned "dicts"

Managing checks

REST API to update / auto-import from SCM

```
zmon check-definitions update select-1-check.yaml
```

select-1-check.yaml

```
name: "Select 1"
owning_team: "Team 1"
command: |
  sql().execute("select 1 as a").results()
entities:
 -  type: postgres
interval: 15
description: "test connection"
```

ID: 2038, Status: , Team: Platform/Software

Check: Number of events written by queue

Trial Run History

Description Number of events written by queue

Command

```
def check():
    items = http(url='/ws/dispatcher/queuestatus').json().iteritems()
    r = {}
    for (tk,tv) in items:
      for (k,v) in tv.iteritems():
        r[k]=v["written"]

    r["_use_scheduled_time"] = True
    return r
```

Interval 1m

Entities environment = live AND project = eventlog-dispatcher AND type = zomcat

Details

+ Add New Alert Definition

Filter alert definitions

Actions	Name	Condition	Entities	Team	Responsible Team	Priority	Status
	Number of Events written	False	(no entity filter)	Platform/Software	Platform/Software		

Alerts

- Executes using a check’s value, bound to single check
- Defines team and responsible team
- Allows inheritance from other alert
- Evaluates Python expression yielding True/False
- No "WARNING" state, no "UNKNOWN" state
- Priorities and tags

ID: 123, Status: , Team: Platform/Software

Alert: Cassandra Node Load

Edit Clone Inherit Delete History ▶ Trial Run ▶ Evaluate Report an Incident
Comments (0)

Description	Load high on Cassandra Nodes
Condition	`value["load5"]>13`
Responsible Team	Platform/Software

Details

Alerts/Checks | Downtimes (0) | History | Children (0)

Alerts (0) | In downtime (0) | OK (13)

Filter entities

Actions		Name	Timestamp	Last run	Value	Captures
		cassandra01	2015-06-12 18:02 (1m ago)	66ms	{"load1":2.43,"load15":3.46,"load5":3.29}	{}
		cassandra02	2015-06-12 18:02 (1m ago)	61ms	{"load1":3.17,"load15":3.01,"load5":3.35}	{}
		cassandra03	2015-06-12 18:02 (1m ago)	34ms	{"load1":0.91,"load15":1.85,"load5":1.47}	{}

ID: 1129, Status: , Team: Incident

Alert: {c} Cassandra Node Down

Edit Clone Inherit Delete History ▶ Trial Run ▶ Evaluate Report an Incident
Comments (0)

Description

At least one Cassandra Node is down in Live Cluster

Analysis

You may use nodetool on multiple hosts in the cluster to verify that nodes are down(D). however, node may appear up(U) when running nodetool on the same machine, but node is still unavailable to cluster.

```
nodetool status

Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address        Load       Tokens  Owns (effective)  Host ID                               Rack
```

Condition `capture(c=value["DownEndpointCount"]) > 0`

Responsible Team Platform/Software

Details

Check Name, ID, History Cassandra Cluster Status, ID: 501, History
Check Team Platform/Software
Check Command `jmx(port=7199).query('org.apache.cassandra.net:type=FailureDetector','DownEndpointCount','UpEndpointCount').results()`

Check Interval 1m
Check Entities host_role_id = 104 AND type = host

Trial Run - Quick feedback and download YAML

Details

Run

Name

Unauthorized Access to Event Data Service {rate}

Check Command

```
            if ks[-2]=='snapshot':
                ep = '.'.join(ks[4:-2])
            else:
              if isinstance(v, dict):
                ep = '.'.join(ks[4:])
              else:
                ep = '.'.join(ks[4:-1])

            if not ep in r:
                r[ep] = {}

            # zmon.response. 200 . GET . EP .

            if ks[3] not in r[ep]:
                r[ep][ks[3]] = {}

            if ks[2] not in r[ep][ks[3]]:
                r[ep][ks[3]][ks[2]] = {}
```

Valid Python expression used to query the instances specified in entities filter.

Alert Condition

```
def alert():
    rate = 0
    for k,v in value.iteritems():
        for hm, vs in v.iteritems():
            ks = vs.keys()
            if '401' in ks:
                rate += vs['401']['mRate']
    return capture(rate=rate) > 0.1
```

Valid Python expression to return true when alert should be "on". Use `value` to access the check result value.

Parameters

Add Parameter

Included Entities

```
[
    {
        "project": "eventlog-data-service-zompy",
        "environment": "live",
        "type": "zompy"
    },
    {
        "project": "eventlog-data-service-zompy",
        "type": "zompy"
    }
]
```

Sharing and reuse of alerts and checks

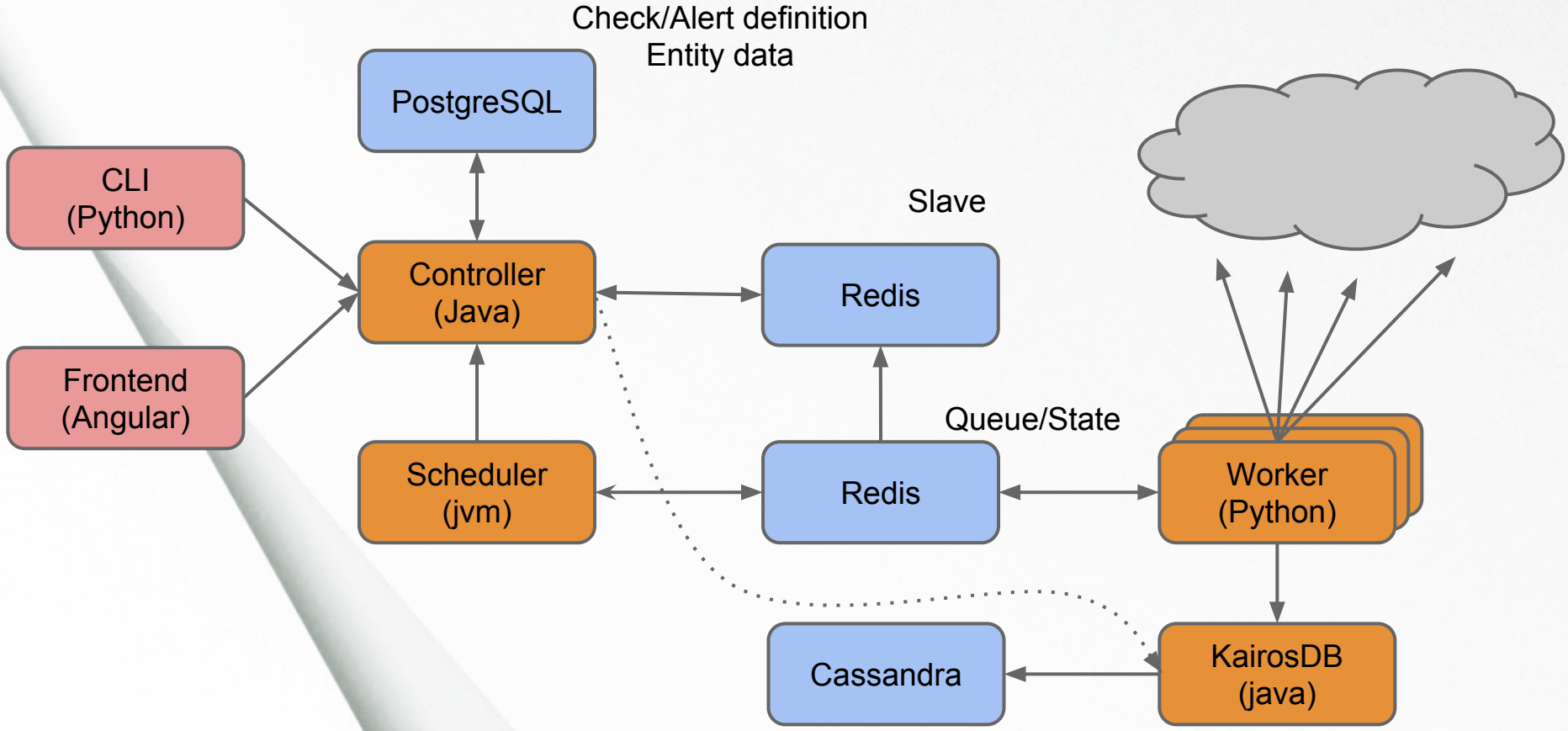
Anyone can add alerts to checks

Alerts are owned by team

Monitor application boundaries/dependencies

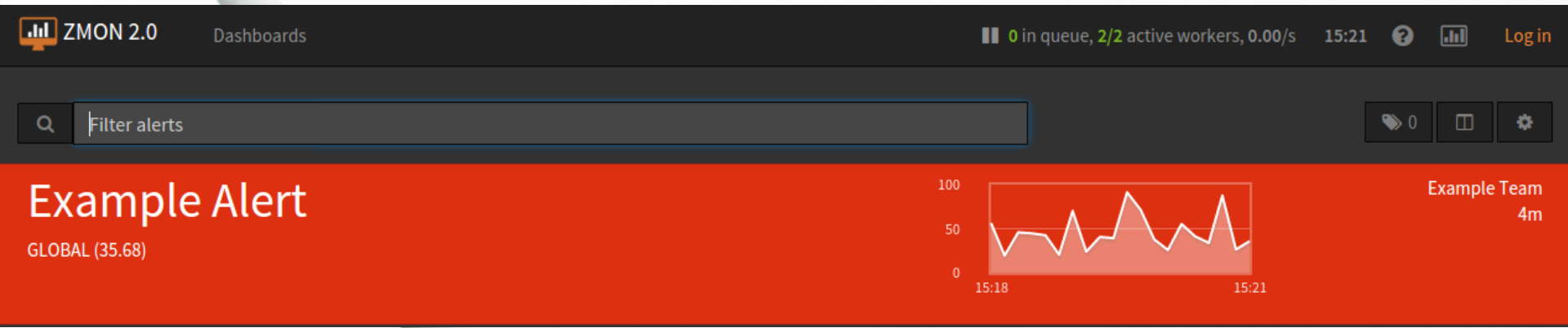
Make use of inheritance to customize

ZMON Core + UI + KairosDB



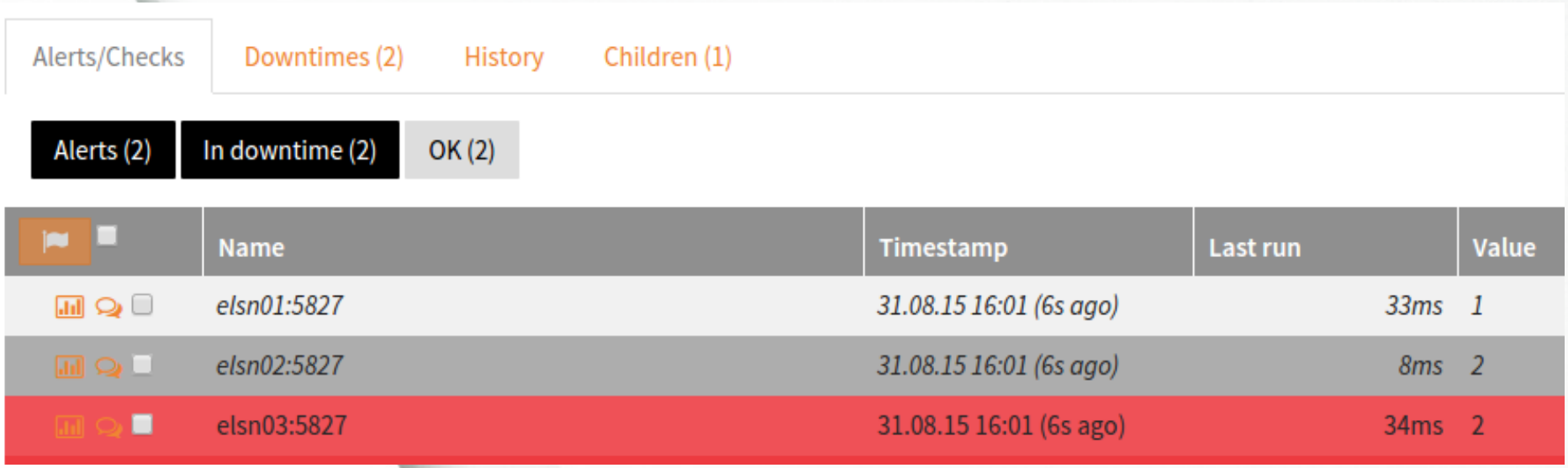
Vagrant Box deploys Docker images

```
→  github-zmon git:(master) vagrant ssh
vagrant@zmon:~$ docker ps
CONTAINER ID        IMAGE                                                  COMMAND                  CREATED             STATUS
fa7a9cf51be9        os-registry.stups.zalan.do/stups/zmon-scheduler-ng:0.1.10     "java -jar zmon-sche   18 minutes ago      Up 18 minutes
bcae089d51bc        os-registry.stups.zalan.do/stups/zmon-worker:0.1.10           "python app.py"        19 minutes ago      Up 19 minutes
cebb102bfdfa        os-registry.stups.zalan.do/stups/zmon-controller:0.1.9        "catalina.sh run"      19 minutes ago      Up 19 minutes
90053f1ccae7        os-registry.stups.zalan.do/stups/zmon-eventlog-service:0.1.9  "java -jar zmon-even   21 minutes ago      Up 21 minutes
2b9b36e79cc5        os-registry.stups.zalan.do/stups/zmon-kairosdb:0.1.6          "/usr/bin/config-kai   22 minutes ago      Up 22 minutes
758ea982e432        abh1nav/cassandra:latest                               "/sbin/my_init"          27 minutes ago      Up 27 minutes
58097a3e0869        os-registry.stups.zalan.do/stups/zmon-redis:0.1.4             "/entrypoint.sh redi   34 minutes ago      Up 34 minutes
bef7fd8cfe48        os-registry.stups.zalan.do/stups/zmon-ldap:0.1.4              "/slapd.sh"            36 minutes ago      Up 36 minutes
6cd8bc4ae386        os-registry.stups.zalan.do/stups/zmon-postgres:0.1.4          "/docker-entrypoint.   36 minutes ago      Up 36 minutes
```



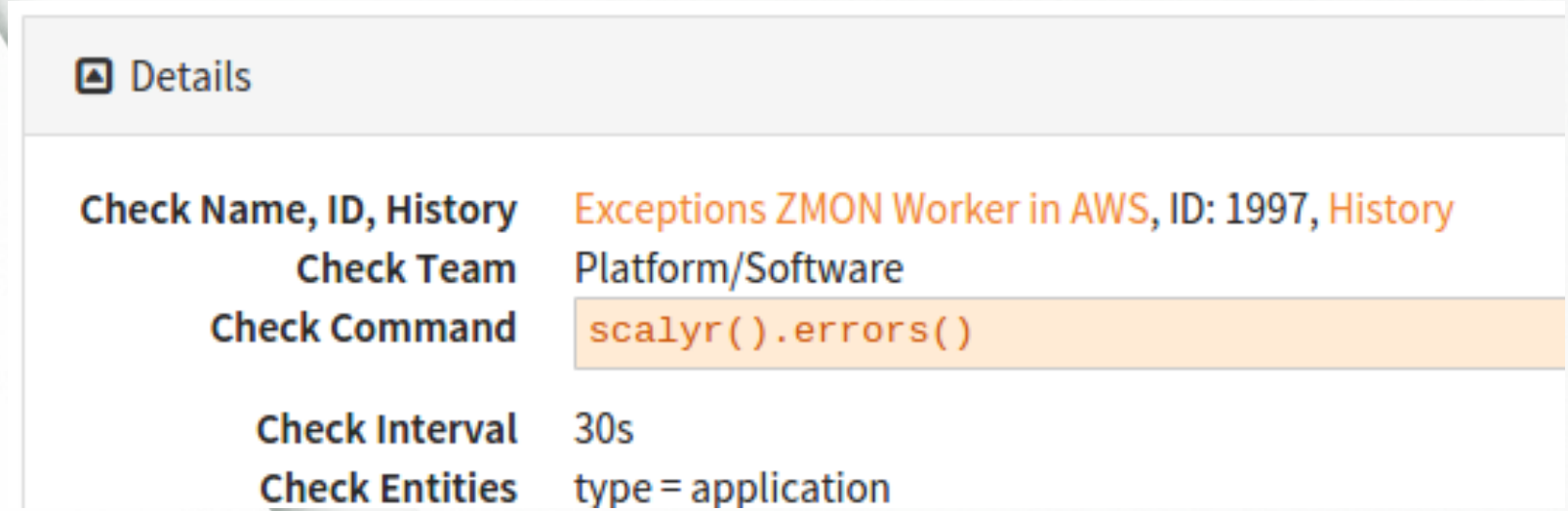
Downtimes

- Set or schedule downtimes using the UI
- Use API to automate downtimes, e.g. in deployment tool



Extendability - Check and Alert functions

- Improve user experience through provided functions



Extendability - Check and Alert functions

```
def errors(self, minutes=30):
    return self.timeseries(self.default_ts_id, minutes)

def timeseries(self, ts_id, minutes=30, buckets=1):

    val = {
        'token': self.read_key,
        'queries': [
            {'timeseriesId':ts_id,
             'startTime': str(minutes)+'m',
             'buckets': buckets}
        ]
    }

    r = requests.post(self.timeseries_url, data=json.dumps(val), headers={"Content-Type": "application/json"})
    j = r.json()
    if j['status'] == 'success':
        if len(j['results'][0]['values'])==1:
            return j['results'][0]['values'][0] * minutes
        return map(lambda x: x * minutes / buckets, j['results'][0]['values'])
    return j
```

The Microservices World

Key Metrics for your service?

- Request rates
- Response rates by HTTP status code
- Latency

Expose your data


```
{
  "zmon.response.200.GET.checks.all-active-check-definitions.count": 10,
  "zmon.response.200.GET.checks.all-active-check-definitions.fifteenMinuteRate": 0.18076110580284566,
  "zmon.response.200.GET.checks.all-active-check-definitions.fiveMinuteRate": 0.1518180485219247,
  "zmon.response.200.GET.checks.all-active-check-definitions.meanRate": 0.06792011610723951,
  "zmon.response.200.GET.checks.all-active-check-definitions.oneMinuteRate": 0.10512398137982051,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.75thPercentile": 1173,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.95thPercentile": 1233,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.98thPercentile": 1282,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.999thPercentile": 1282,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.99thPercentile": 1282,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.max": 1282,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.mean": 1170,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.median": 1161,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.min": 1114,
  "zmon.response.200.GET.checks.all-active-check-definitions.snapshot.stdDev": 42,
}
```

Start tracking your metrics

ID: 4953, Status: , Team: Platform/Software

Alert: Actuater AWS Metrics

Edit Clone Inherit Delete History ▶ Trial Run ▶ Evaluate Report an Incident
Comments (0)

Description	.
Condition	`False`
Responsible Team	Platform/Software

Details

Check Name, ID, History	REST Metrics (AWS), ID: 2132, History
Check Team	Platform/Software
Check Command	

```
def check():
    j = http(url=':{}/metrics'.format(entity['ports'].values()[-1:][0]) if 'ports' in entity else '/metrics',
             oauth2=True).actuator_metrics()

    # workaround for charting if metrics endpoint not covered
    j["metrics"]={"GET":{"200":{"mRate":0,"count":0,"75th":0,"99th":0,"median":0,"min":0,"max":0}}}

    return j
```

Check Interval	1m
Check Entities	type = instance

Display application statistics

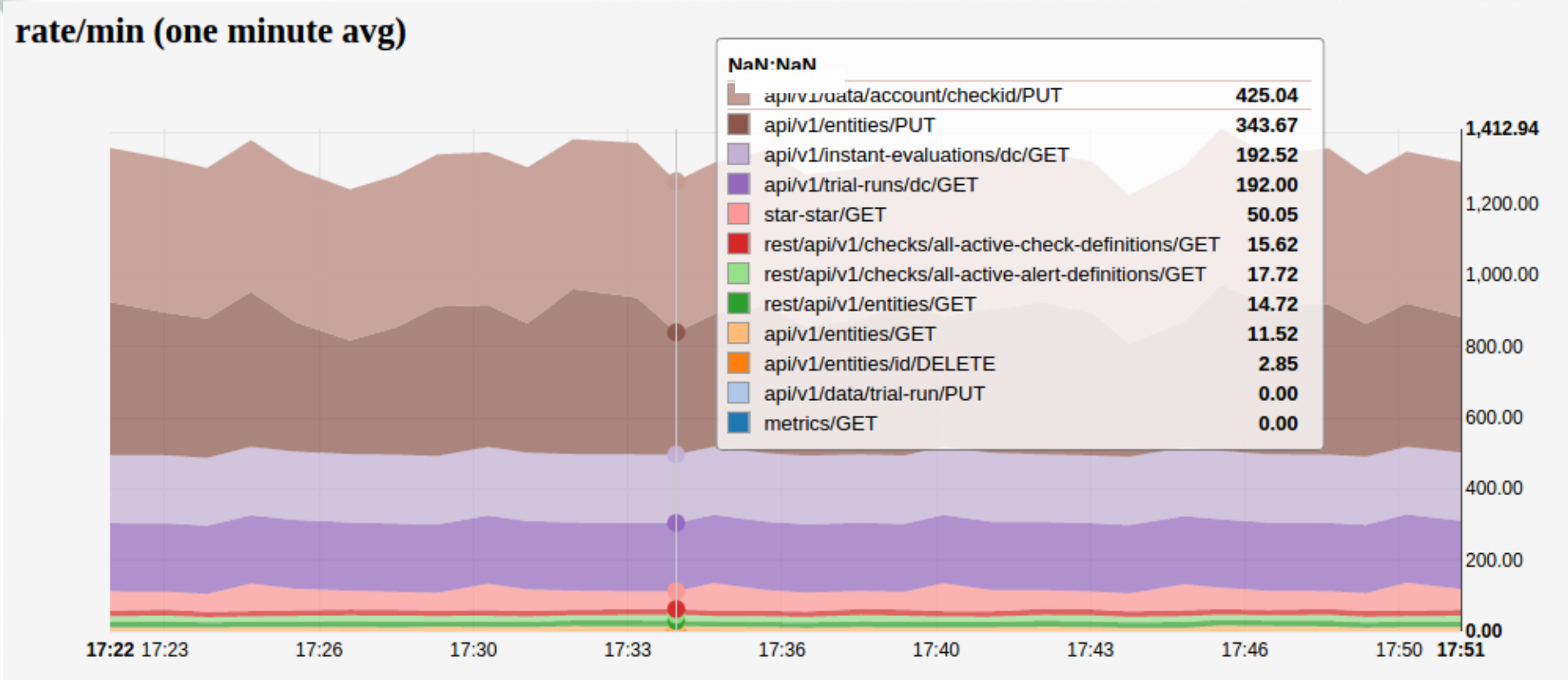
kio

Load		Response time	Errors				
98.11 req/min	2 instances	2202 ms	2.7%	1.52 4XX/min	1.17 5XX/min	1771 ms	2 instances

zmon data service

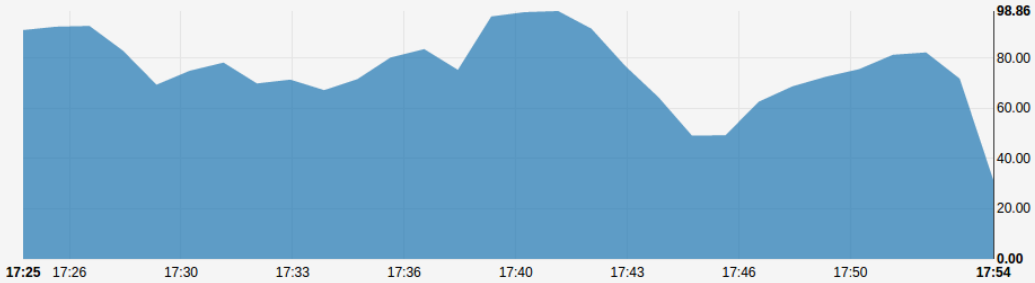
Load		Response time	Errors				
1425.34 req/min	2 instances	3508 ms	5.1%	72.98 4XX/min	0.00 5XX/min	19 ms	2 instances

Application metrics

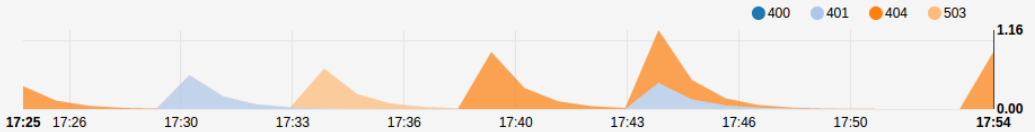


Continued ...

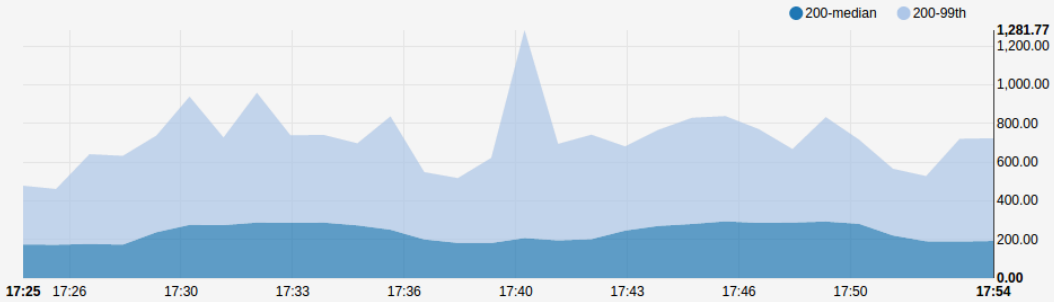
Request rates for status 200



Request rates other



Response times for status 200



Reuse of check

ID: 5255, Status: , Team: STUPS

Alert: Unauthorized Access to Event Data Service {rate}

Edit Clone Inherit Delete History ▶ Trial Run ▶ Evaluate Report an Incident
Comments (0)

Description Monitoring 401 on EventLog Data Service Endpoint

Condition

```
def alert():
    rate = 0
    for k,v in value.iteritems():
        for hm, vs in v.iteritems():
            ks = vs.keys()
            if '401' in ks:
             rate += vs['401']['mRate']
    return capture(rate=rate) > 0.1
```

Responsible Team STUPS

Details

Check Name, ID, History REST Metrics, ID: 2115, History
Check Team Platform/Software

Libraries available for

Spring boot

https://github.com/zalando/zmon-actuator

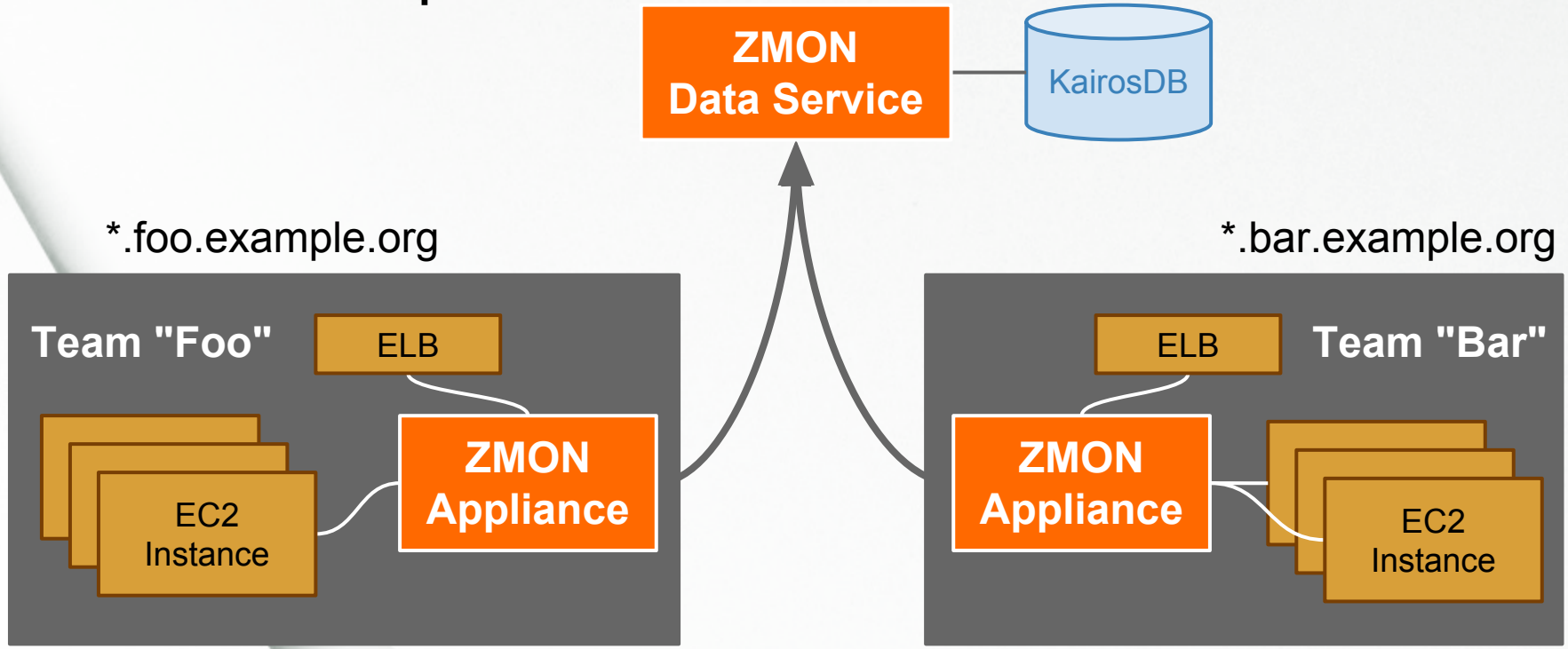
Clojure

https://github.com/zalando-stups/friboo/

Play (done, to be released)

Multi DC / AWS

ZMON in AWS Setup



Multi DC / Zone deployment possible

- Scheduler supports queue filters by entity
  - e.g. {"dc":"dc1"} vs {"dc":"dc2"} queue filters
- Scheduler can apply base filter
  - only handles entities with {"dc":"dc1"}
- Worker can report home using:
  - Redis (we use this across DCs)
  - HTTPS (AWS->DC)

ZMON AWS Agent

Uses Amazon API to fetch:

- ELBs
- EC2 instances
- RDS instances

Pushes enriched entities to entity service

Prometheus?

read "text" result

Kubernetes Example: Exports in Prometheus text format

```
kubelet_docker_operations_latency_microseconds{operation_type="inspect_container",quantile="0.9"} 9602
kubelet_docker_operations_latency_microseconds{operation_type="list_containers",quantile="0.9"} 9740
```

Check Command

```python
def check():
    d = http(':10255/metrics').text()
    r= {}
    for l in d.split("\n"):
        if l.startswith("kubelet_docker_operations_latency_microseconds{"):
            lp = l.split(" ")
            rv=' '.join(lp[1:])
            if "NaN"==rv:
                continue
            a=lp[0]
            a = a[a.find('{')+1:a.rfind('}')]
            a = [(v.split('=')[0], v.split('=')[1]) for v in a.split(",")]
            c = r
            while True:
                b=a.pop(0)[1].replace('"','')
                if len(a)==0:
                    c[b]=rv.replace('"','')
                    break
                else:
                    if b in c:
```

Valid Python expression used to query the instances specified in entities filter.

Yields a usable nested dictionary


```
{"list_images":
   {"0.9":"120252",
    "0.99":"120252",
    "0.5":"120252"},
 "version":
   {"0.9":"1281",
    "0.99":"2183",
    "0.5":"873"},
 "list_containers":
   {"0.9":"9740",
    "0.99":"23378",
    "0.5":"3717"},
  "inspect_container":
   {"0.9":"9602",
    "0.99":"18367",
    "0.5":"4419"}
}
```

Internals

ZMON's basic data flow


```
{"check":
  {"id": 1,
   "entity": {"host":"monitor01"},
   "command": "snmp().load()",
   "alerts":[
     {"id":100,
      "condition": "value['load1']>10"}
   ]
  }
}
```

ZMON's basic data flow


```
-- store check result "snmp().load()"
lpush zmon:check:1:monitor01 {"load1":5,"load5":3,"load15:2}

-- keep last 20 results (for dashboard charts)
ltrim zmon:check:1:monitor01 20

-- alert active?
sadd zmon:alert:100 monitor01

-- alert inactive?
srem zmon:alert:100 monitor01
```

ZMON Vagrant Box:

https://github.com/zalando/zmon

ZMON Homepage:

https://zalando.github.io/zmon

Zalando Tech:

https://tech.zalando.com