ContainerSolutions

Monitor My Socks

Using Prometheus in a polyglot open source microservices reference architecture

Phil Winder

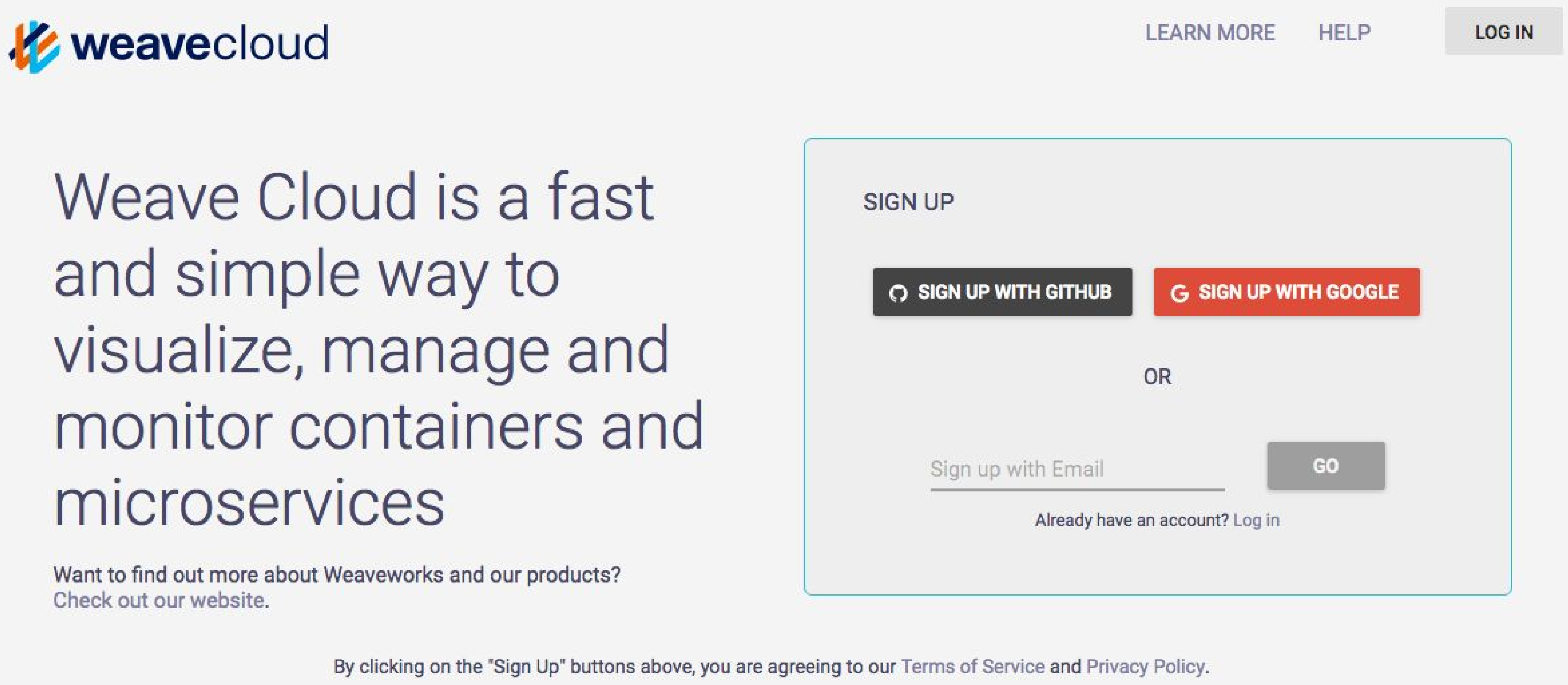
Freelance Engineer
@DrPhilWinder
phil@WinderResearch.com

Winder Research

ContainerSolutions

weave~~works~~ socks

How did it come to this?



Weave Cloud is a fast and simple way to visualize, manage and monitor containers and microservices

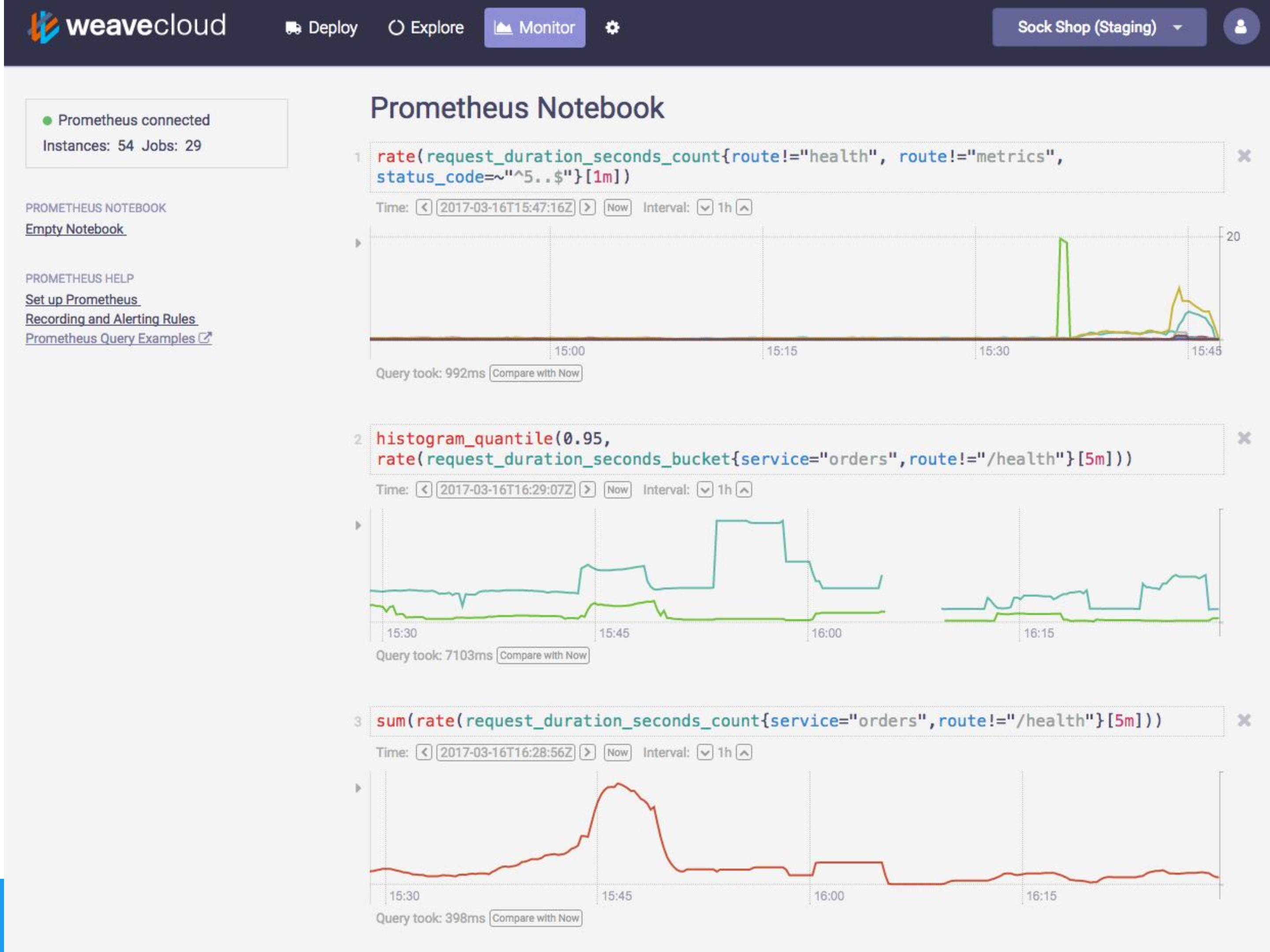
Want to find out more about Weaveworks and our products? Check out our website.

Already have an account? Log in

By clicking on the "Sign Up" buttons above, you are agreeing to our Terms of Service and Privacy Policy.

Cortex

- Managed
- Multi-tenant
- Unlimited data retention
- WIP



● Prometheus connected

Instances: 54 Jobs: 29

PROMETHEUS NOTEBOOK

Empty Notebook

PROMETHEUS HELP

Set up Prometheus

Recording and Alerting Rules

Prometheus Query Examples

Prometheus Notebook

```
rate(request_duration_seconds_count{route!="health", route!="metrics",
status_code=~"^5..$"}[1m])
```

Time: 2017-03-16T15:47:16Z Now Interval: 1h

Query took: 992ms Compare with Now

```
histogram_quantile(0.95,
rate(request_duration_seconds_bucket{service="orders",route!="/health"}[5m]))
```

Time: 2017-03-16T16:29:07Z Now Interval: 1h

Query took: 7103ms Compare with Now

```
sum(rate(request_duration_seconds_count{service="orders",route!="/health"}[5m]))
```

Time: 2017-03-16T16:28:56Z Now Interval: 1h

Query took: 398ms Compare with Now

How do you do monitoring?

How do you do microservices?

Where are the examples?

What can I use to test this out?

How can I try against different orchestrators?

How do I do this in <insert language here>?

Etc.

Etc.

SOCK SHOP

An open source reference microservices architecture

Icon by http://www.flaticon.com/authors/freepik

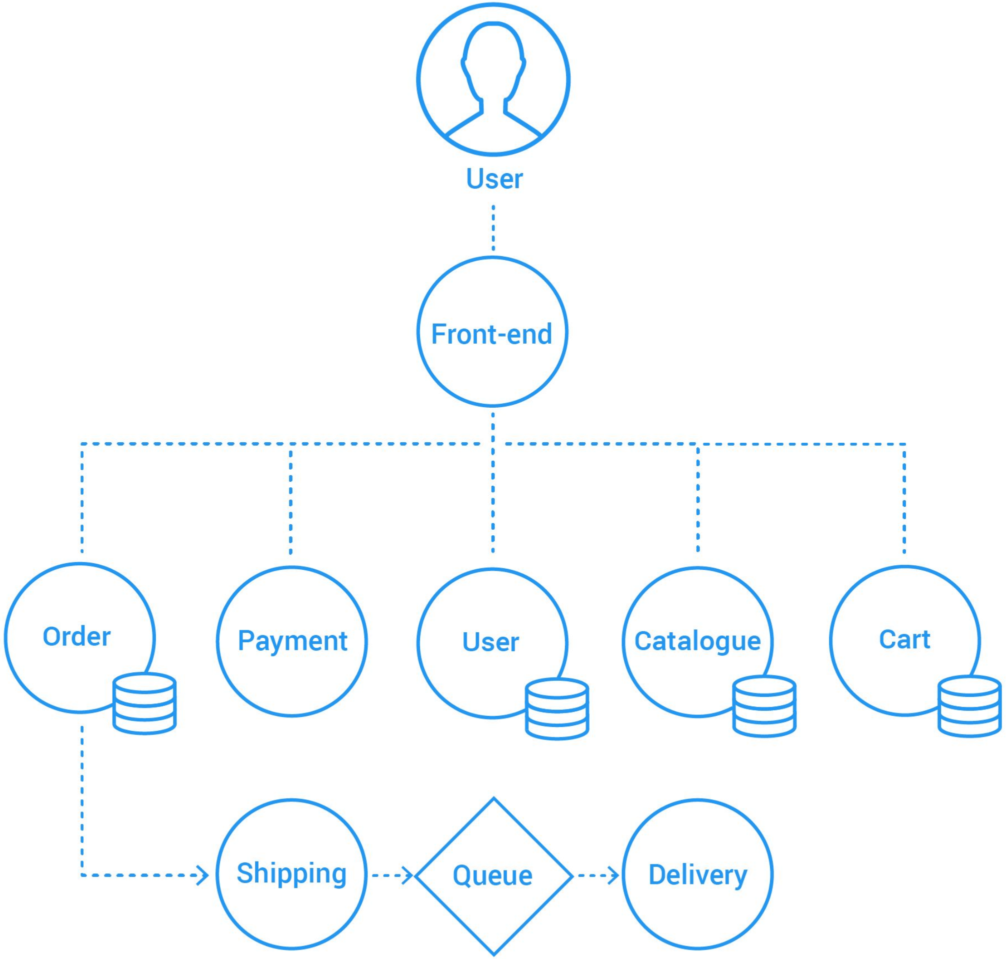
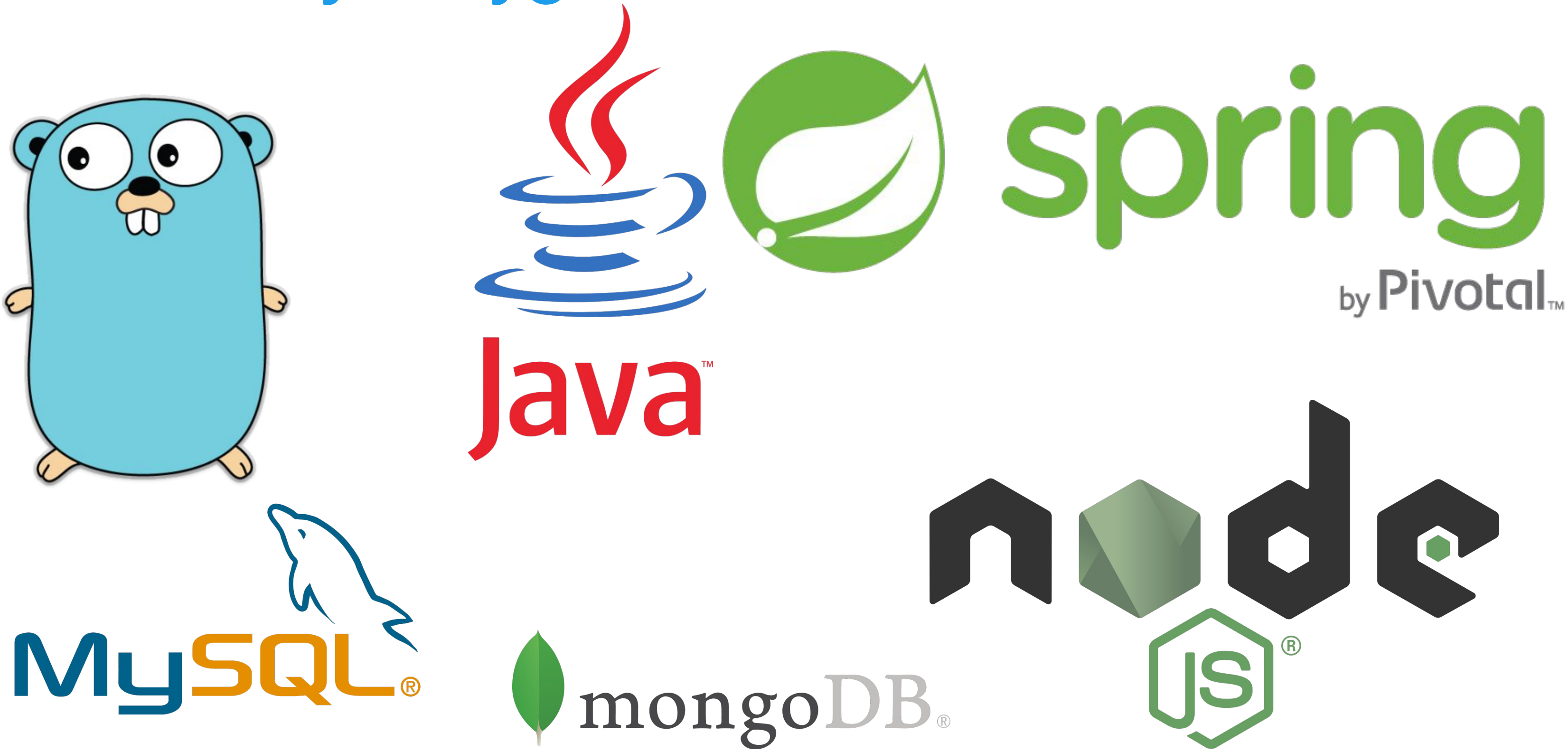
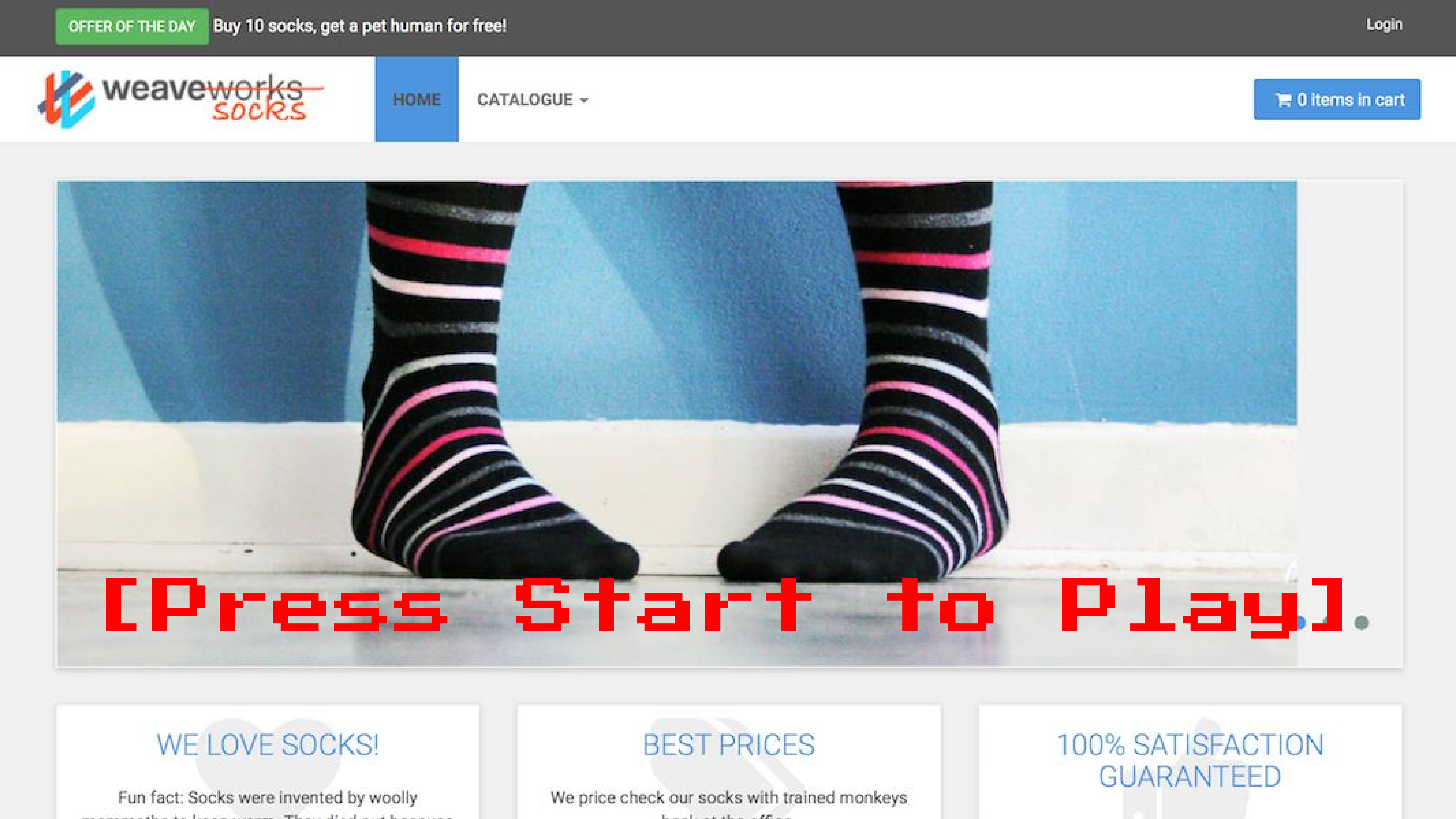


Image by Remember To Play

Obnoxiously Polyglot





Buy 10 socks, get a pet human for free!

[Press Start to Play]

WE LOVE SOCKS!

Fun fact: Socks were invented by woolly mammoths to keep warm. They died out because

BEST PRICES

We price check our socks with trained monkeys back at the office

100% SATISFACTION GUARANTEED

git.io/sock-shop

github.com/microservices-demo/**microservices-demo**

ContainerSolutions

How to do Prometheus

Prometheus Libraries

https://prometheus.io/docs/instrumenting/clientlibs/

Officially accepted as proper languages:

Go, Java/Scala, Python, Ruby

Unofficial libraries:

Bash, C++, Common Lisp, Elixir, Erlang, Haskell, Lua, .NET / C#, Node.js, PHP, Rust

Helper libraries for frameworks

https://github.com/prometheus/client_java

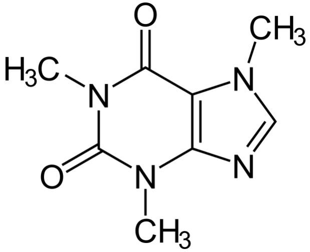
HIBERNATE

spring
by Pivotal™

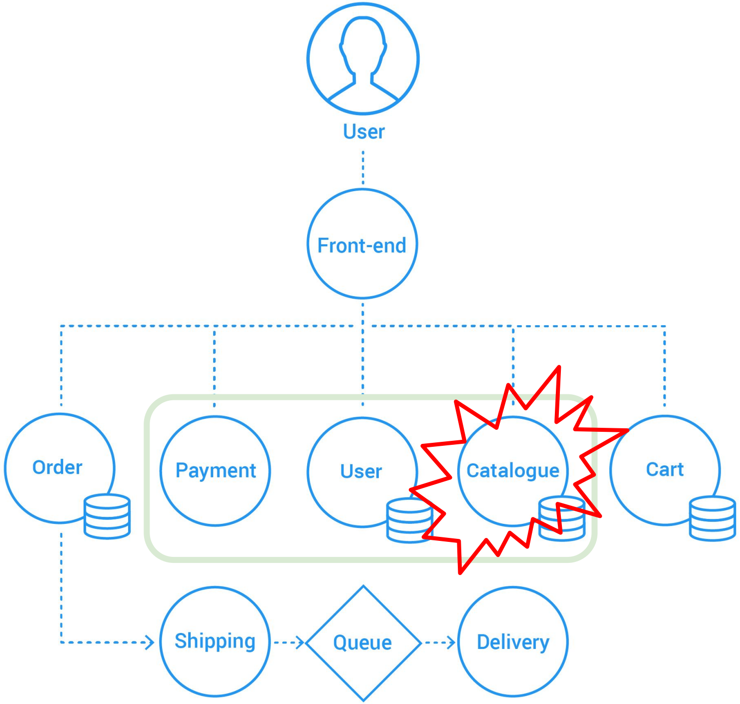
jetty://

Apache LOG4J ™

LOGBack™
The Generic, Reliable Fast & Flexible Logging Framework



go Go GO!


User
Front-end
Order
Payment
User
Catalogue
Cart
Shipping
Queue
Delivery

Metric definition

https://github.com/microservices-demo/catalogue/blob/master/cmd/cataloguesvc/main.go

```
var (
    HTTPLatency = prometheus.NewHistogramVec(prometheus.HistogramOpts{
        Name:    "request_duration_seconds",
        Help:    "Time (in seconds) spent serving HTTP requests.",
        Buckets: prometheus.DefBuckets,
    }, []string{"method", "route", "status_code", "isWS"})
)


func init() {
    prometheus.MustRegister(HTTPLatency)
}
```

Metric definition

https://github.com/microservices-demo/catalogue/blob/master/cmd/cataloguesvc/main.go

```
var (
    HTTPLatency = prometheus.NewHistogramVec(prometheus.HistogramOpts{
        Name:    "request_duration_seconds",
        Help:    "Time (in seconds) spent serving HTTP requests.",
        Buckets: prometheus.DefBuckets,
    }, []string{"method", "route", "status_code", "isWS"})
)


func init() {
    prometheus.MustRegister(HTTPLatency)
}
```

Request rate (count)
Error rate (count)
Duration (buckets)
@tom_wilkie

Middleware definition

https://github.com/weaveworks/common/blob/master/middleware/middleware.go

```
// Interface is the shared contract for all middlesware, and allows middlesware
// to wrap handlers.
type Interface interface {
    Wrap(http.Handler) http.Handler
}
```

Instrumentation definition

https://github.com/weaveworks/common/blob/master/middleware/instrument.go

```
// RouteMatcher matches routes
type RouteMatcher interface {
     Match(*http.Request, *mux.RouteMatch) bool
}

// Instrument is a Middleware which records timings for every HTTP request
type Instrument struct {
     RouteMatcher RouteMatcher
     Duration     *prometheus.HistogramVec
}
```

Instrumentation

https://github.com/weaveworks/common/blob/master/middleware/instrument.go

```
// Wrap implements middleware.Interface
func (i Instrument) Wrap(next http.Handler) http.Handler {
    return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
        begin := time.Now()
        isWS := strconv.FormatBool(IsWSHandshakeRequest(r))
        interceptor := &interceptor{ResponseWriter: w, statusCode: http.StatusOK}
        route := i.getRouteName(r)
        next.ServeHTTP(interceptor, r)
        var (
            status = strconv.Itoa(interceptor.statusCode)
            took   = time.Since(begin)
        )
        i.Duration.WithLabelValues(r.Method, route, status,
isWS).Observe(took.Seconds())
    })
```

Main Wiring

https://github.com/microservices-demo/catalogue/blob/master/cmd/cataloguesvc/main.go

```
// Service
var service catalogue.Service
…

// Endpoint
endpoints := catalogue.MakeEndpoints(service, tracer)

// HTTP router
// Prom handler in here
router := catalogue.MakeHTTPHandler(ctx, endpoints, *images, logger, tracer)
```

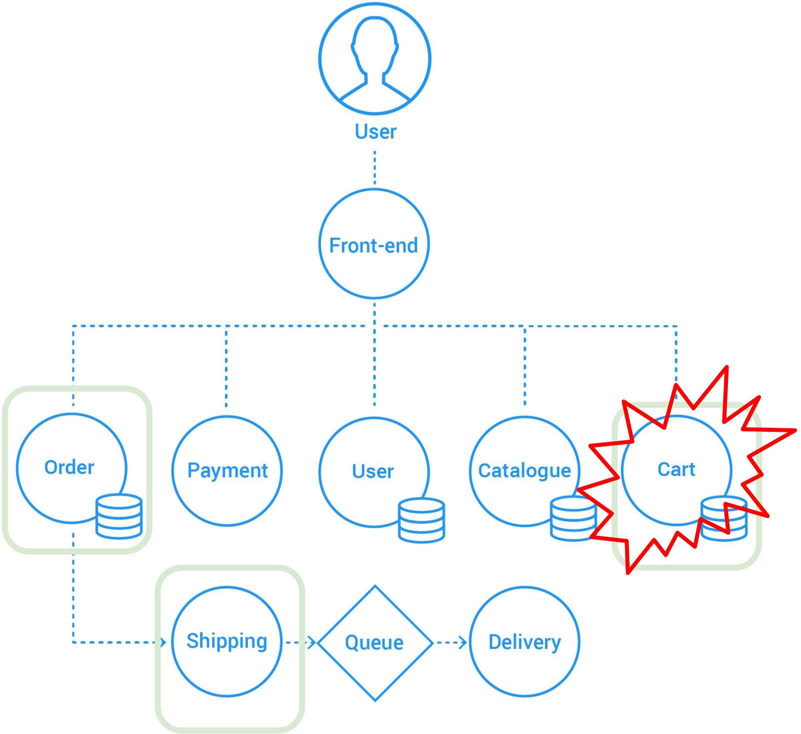
Main Middleware

https://github.com/microservices-demo/catalogue/blob/master/cmd/cataloguesvc/main.go

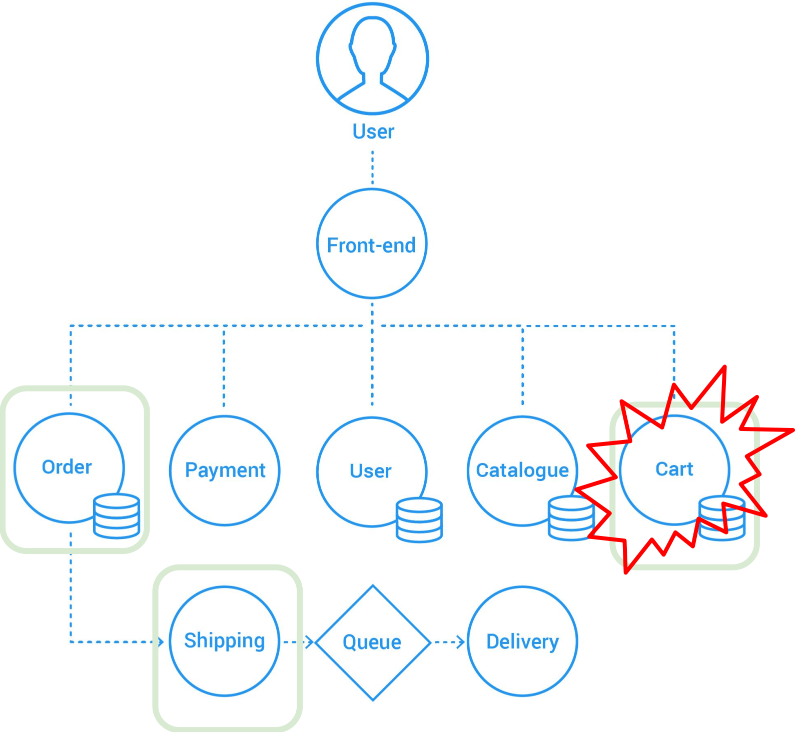
```go
// Middleware
httpMiddleware := []middleware.Interface{
    middleware.Instrument{
        Duration:     middleware.HTTPLatency,
        RouteMatcher: router,
    },
}
// Handler
handler := middleware.Merge(httpMiddleware...).Wrap(router)

…
// Standard server stuff
    errc <- http.ListenAndServe(":"+*port, handler)
```

j for Java


User
Front-end
Order
Payment
User
Catalogue
Cart
Shipping
Queue
Delivery

j for Java


User
Front-end
Order
Payment
User
Catalogue
Cart
Shipping
Queue
Delivery

j
Java™

First - the easy bit

https://github.com/microservices-demo/carts/blob/master/src/main/java/works/weave/socks/cart/CartApplication.java

```
@SpringBootApplication
@EnablePrometheusEndpoint
public class CartApplication {
   public static void main(String[] args) {
       SpringApplication.run(CartApplication.class, args);
   }
}
```

…

```
# In applications.properties
# Disable actuator metrics endpoints
endpoints.metrics.enabled=false
endpoints.prometheus.id=metrics
```

Timing - Easy option

```
@PrometheusTimeMethod(name = "my_method_seconds", help = "The number of seconds taken by the main handler")
public Object handleRequest {
     // Do stuff, this will be timed.
}
```

Timing - Easy option

```
@PrometheusTimeMethod(name = "my_method_seconds", help = "The number of seconds taken by the main handler")
public Object handleRequest {
     // Do stuff, this will be timed.
}
```

Uh oh. Timed methods are recorded as a Summary type, with no quantile information. Quite possible that this isn’t right for you application.

Timing - Intermediate option

```
// Stub. Insert new code for Histogram annotation here.
```

Timing - Current implementation

https://github.com/microservices-demo/carts/blob/master/src/main/java/works/weave/socks/cart/middleware/HTTPMonitoringInterceptor.java

```java
@Configuration
public class WebMvcConfig {
   @Bean
   HTTPMonitoringInterceptor httpMonitoringInterceptor() {
       return new HTTPMonitoringInterceptor();
   }

   @Bean
   public MappedInterceptor myMappedInterceptor(HTTPMonitoringInterceptor interceptor) {
       return new MappedInterceptor(new String[]{"/**"}, interceptor);
   }
}
```

Timing - The nasty bit

https://github.com/microservices-demo/carts/blob/master/src/main/java/works/weave/socks/cart/middleware/HTTPMonitoringInterceptor.java

```
public class HTTPMonitoringInterceptor implements HandlerInterceptor {
   static final Histogram requestLatency = Histogram.build()
           .name("request_duration_seconds")
           .help("Request duration in seconds.")
           .labelNames("service", "method", "route", "status_code")
           .register();
   @Override
   public boolean preHandle(HttpServletRequest httpServletRequest, HttpServletResponse
           httpServletResponse, Object o) throws Exception {
       httpServletRequest.setAttribute(startTimeKey, System.nanoTime());
       return true;
   }
...
```

Timing - The nasty bit

https://github.com/microservices-demo/carts/blob/master/src/main/java/works/weave/socks/cart/middleware/HTTPMonitoringInterceptor.java

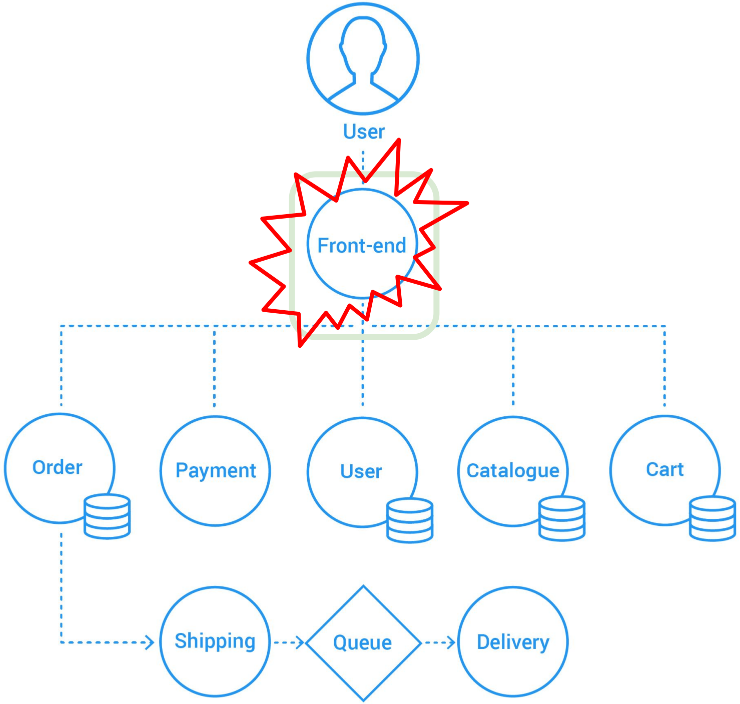
```java
@Override
public void postHandle(HttpServletRequest httpServletRequest, HttpServletResponse
        httpServletResponse, Object o, ModelAndView modelAndView) throws Exception {
    long start = (long) httpServletRequest.getAttribute(startTimeKey);
    long elapsed = System.nanoTime() - start;
    double seconds = (double) elapsed / 1000000000.0;
    String matchedUrl = getMatchingURLPattern(httpServletRequest); // ← Key part
    if (!matchedUrl.equals("")) {
        requestLatency.labels(
                serviceName,
                httpServletRequest.getMethod(),
                matchedUrl,
                Integer.toString(httpServletResponse.getStatus())
        ).observe(seconds);
    }
}
```

Recommendation?

Write your own annotation. Annotate each method you want to monitor.

Becomes tricky with spring-boot-data types, due to auto-magic.

NodeJS - Express



Declarations

https://github.com/microservices-demo/front-end/blob/master/api/metrics/index.js

```
(function (){
 'use strict';

 var express = require("express")
   , client  = require('prom-client')
   , app     = express()

 const metric = {
   http: {
     requests: {
       duration: new client.Histogram('request_duration_seconds', 'request duration in
seconds', ['service', 'method', 'route', 'status_code']),
     }
   }
 }
```

Middleware

https://github.com/microservices-demo/front-end/blob/master/api/metrics/index.js

```
 function observe(method, path, statusCode, start) {
   var route = path.toLowerCase();
   if (route !== '/metrics' && route !== '/metrics/') {
       var duration = s(start); // Helper method to calculate duration
       var method = method.toLowerCase();
       metric.http.requests.duration.labels('front-end', method, route,
statusCode).observe(duration);
   }
 };

 app.use(middleware);
 app.get("/metrics", function(req, res) {
     res.header("content-type", "text/plain");
     return res.end(client.register.metrics())
 });
```

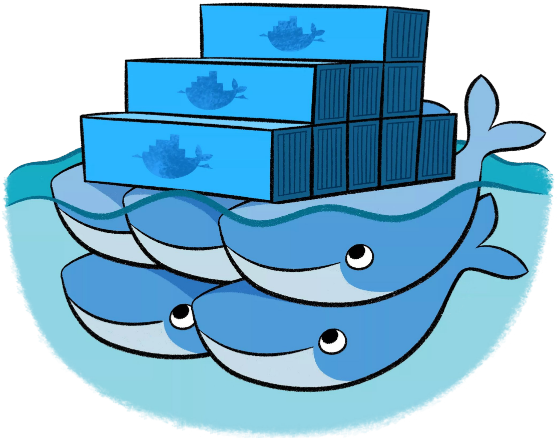
ContainerSolutions

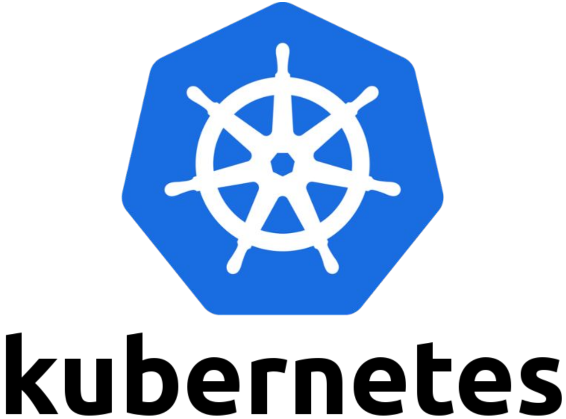
How to do Monitoring

MISUSE OF STUPID MEME
AS A MEANS OF SAYING "SHIP IT"
ABCMEMES.COM
CHUCK NORRIS APPROVED.
SHIP IT!
SHIP IT
T NOW!
memegenerator.net
IT COMPILES
SHIP IT
quickmeme.com
FUCK IT
BRILLIANT!
SHIP IT
IF YOU COULD GO AHEAD AND 'SHIP IT!'
THAT WOULD BE GREAT
LOOKS GOOD TO ME!
SHIP IT!

MISUSE OF STUPID MEME
AS A MEANS OF SAYING "SHIP IT"
ABCMEMES.COM
CHUCK NORRIS APPROVED.
SH
SHIP IT
OW!
emegenerator.net
IT COMPILES
SHIP IT
quickmeme.com
ONE DOES NOT SIMPLY
SHIP IT!
memegenerator.net
FUCK IT
SHIP IT
BRILLIANT!
IF YOU C
THAT WOULD BE GREAT
OOKS GOOD TO ME!
SHIP IT!

Deploying all the things





Deploying to Kubernetes

https://github.com/microservices-demo/microservices-demo/tree/master/deploy/kubernetes

```
...
   spec:
     containers:
     - name: prometheus
       image: 'prom/prometheus:v1.5.2'
       args:
         - '-storage.local.retention=360h'
         - '-storage.local.memory-chunks=1048576'
         -
'-config.file=/etc/prometheus/prometheus.yml'
         -
'-alertmanager.url=http://alertmanager:9093'
       ports:
       - name: web
         containerPort: 9090
...
```

```
...
       volumeMounts:
       - name: config-volume
         mountPath: /etc/prometheus
       - name: alertrules-volume
         mountPath: /etc/prometheus-rules
     volumes:
     - name: config-volume
       configMap:
         name: prometheus-configmap
     - name: alertrules-volume
       configMap:
         name: prometheus-alertrules
```

Deploying to Kubernetes

https://github.com/microservices-demo/microservices-demo/tree/master/deploy/kubernetes

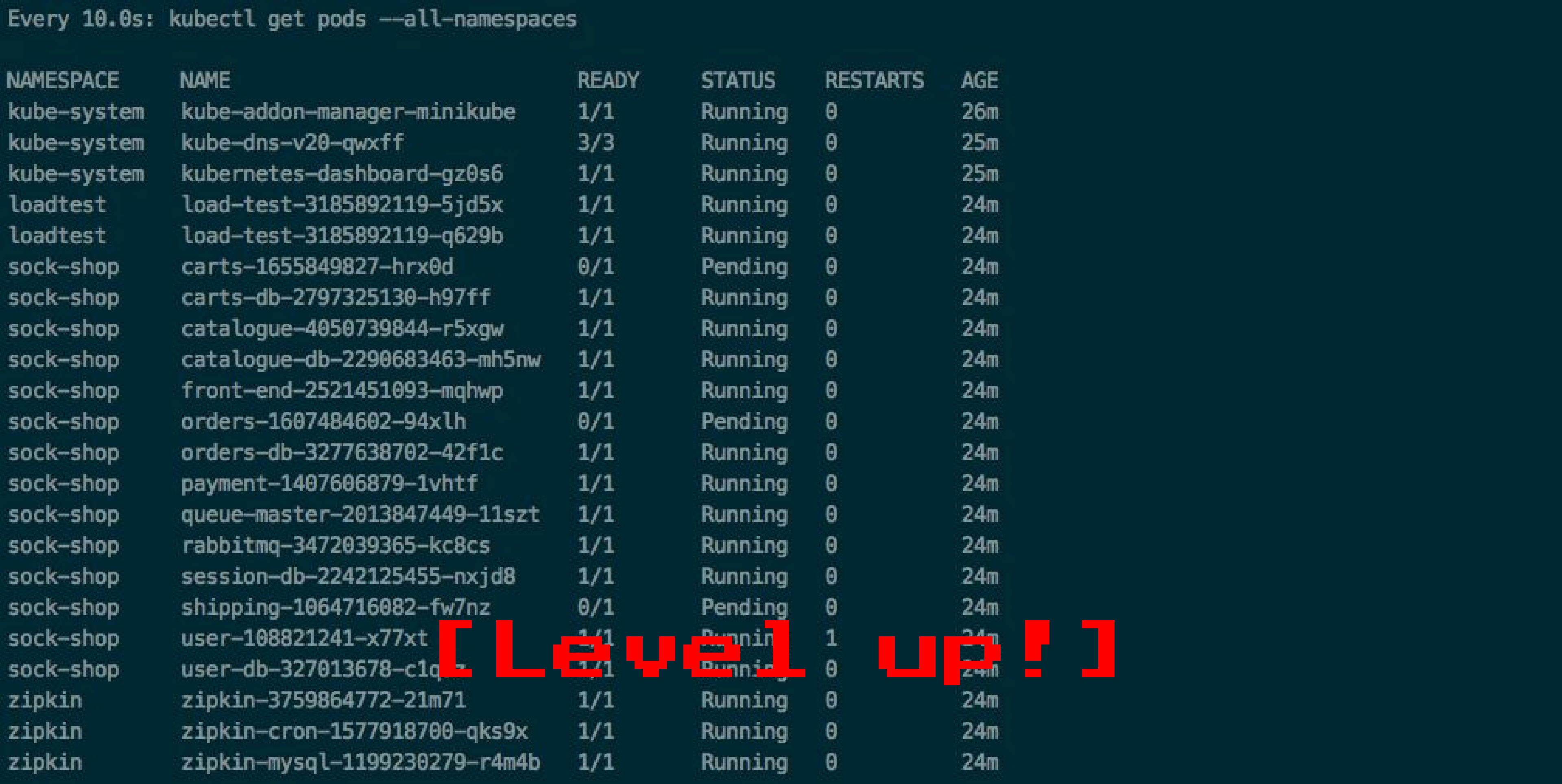
```
kind: Service
...
spec:
 selector:
   app: prometheus
 type: NodePort
 ports:
 - name: prometheus
   protocol: TCP
   port: 9090
   targetPort: 9090
   nodePort: 31090
```

```
kind: ConfigMap
metadata:
 name: prometheus-configmap
data:
 prometheus.yml: |
   global:
     scrape_interval: 15s
   scrape_configs:
     - job_name: kubernetes-service-endpoints
…
     - job_name: kubernetes-pods
…
     - job_name: kubernetes-nodes
```

Rock it with Minikube

```
$ minikube start  --memory 4096
$ git clone https://github.com/microservices-demo/microservices-demo.git
$ kubectl create -f ./deploy/kubernetes/complete-demo.yaml
$ kubectl create -f ./deploy/kubernetes/manifests-monitoring
```

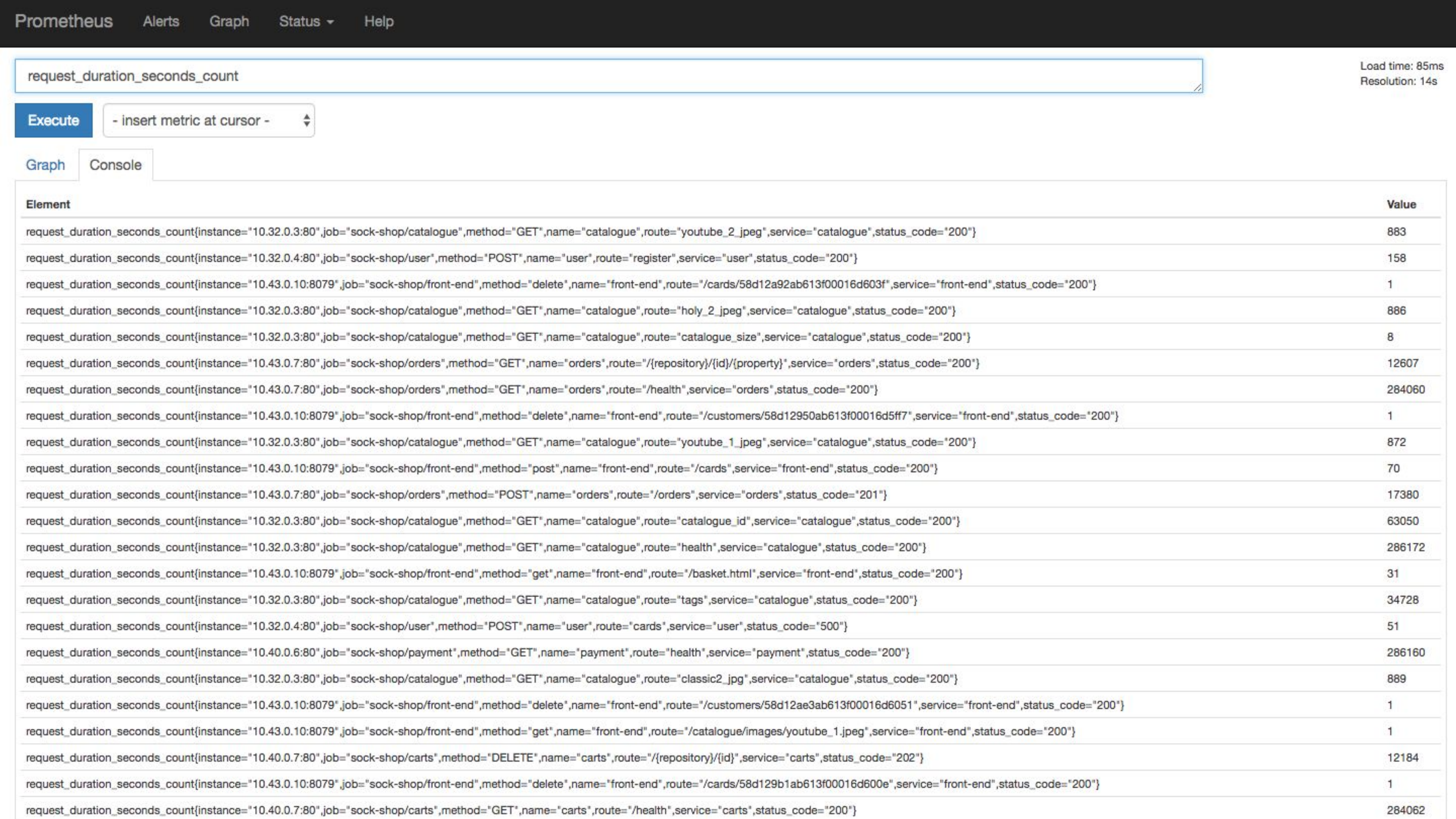
………… Then wait. (each image is approx 100MB = 20 mins to download images)



```
Every 10.0s: kubectl get pods --all-namespaces

NAMESPACE     NAME                                READY     STATUS    RESTARTS   AGE
kube-system   kube-addon-manager-minikube         1/1       Running   0          26m
kube-system   kube-dns-v20-qwxff                  3/3       Running   0          25m
kube-system   kubernetes-dashboard-gz0s6          1/1       Running   0          25m
loadtest      load-test-3185892119-5jd5x          1/1       Running   0          24m
loadtest      load-test-3185892119-q629b          1/1       Running   0          24m
sock-shop     carts-1655849827-hrx0d              0/1       Pending   0          24m
sock-shop     carts-db-2797325130-h97ff           1/1       Running   0          24m
sock-shop     catalogue-4050739844-r5xgw          1/1       Running   0          24m
sock-shop     catalogue-db-2290683463-mh5nw       1/1       Running   0          24m
sock-shop     front-end-2521451093-mqhwp          1/1       Running   0          24m
sock-shop     orders-1607484602-94xlh             0/1       Pending   0          24m
sock-shop     orders-db-3277638702-42f1c          1/1       Running   0          24m
sock-shop     payment-1407606879-1vhtf            1/1       Running   0          24m
sock-shop     queue-master-2013847449-11szt       1/1       Running   0          24m
sock-shop     rabbitmq-3472039365-kc8cs           1/1       Running   0          24m
sock-shop     session-db-2242125455-nxjd8         1/1       Running   0          24m
sock-shop     shipping-1064716082-fw7nz           0/1       Pending   0          24m
sock-shop     user-108821241-x77xt                1/1       Running   1          24m
sock-shop     user-db-327013678-c1q[illegible]    1/1       Running   0          24m
zipkin        zipkin-3759864772-21m71             1/1       Running   0          24m
zipkin        zipkin-cron-1577918700-qks9x        1/1       Running   0          24m
zipkin        zipkin-mysql-1199230279-r4m4b       1/1       Running   0          24m
```

[Level up!]



request_duration_seconds_count

Load time: 85ms
Resolution: 14s

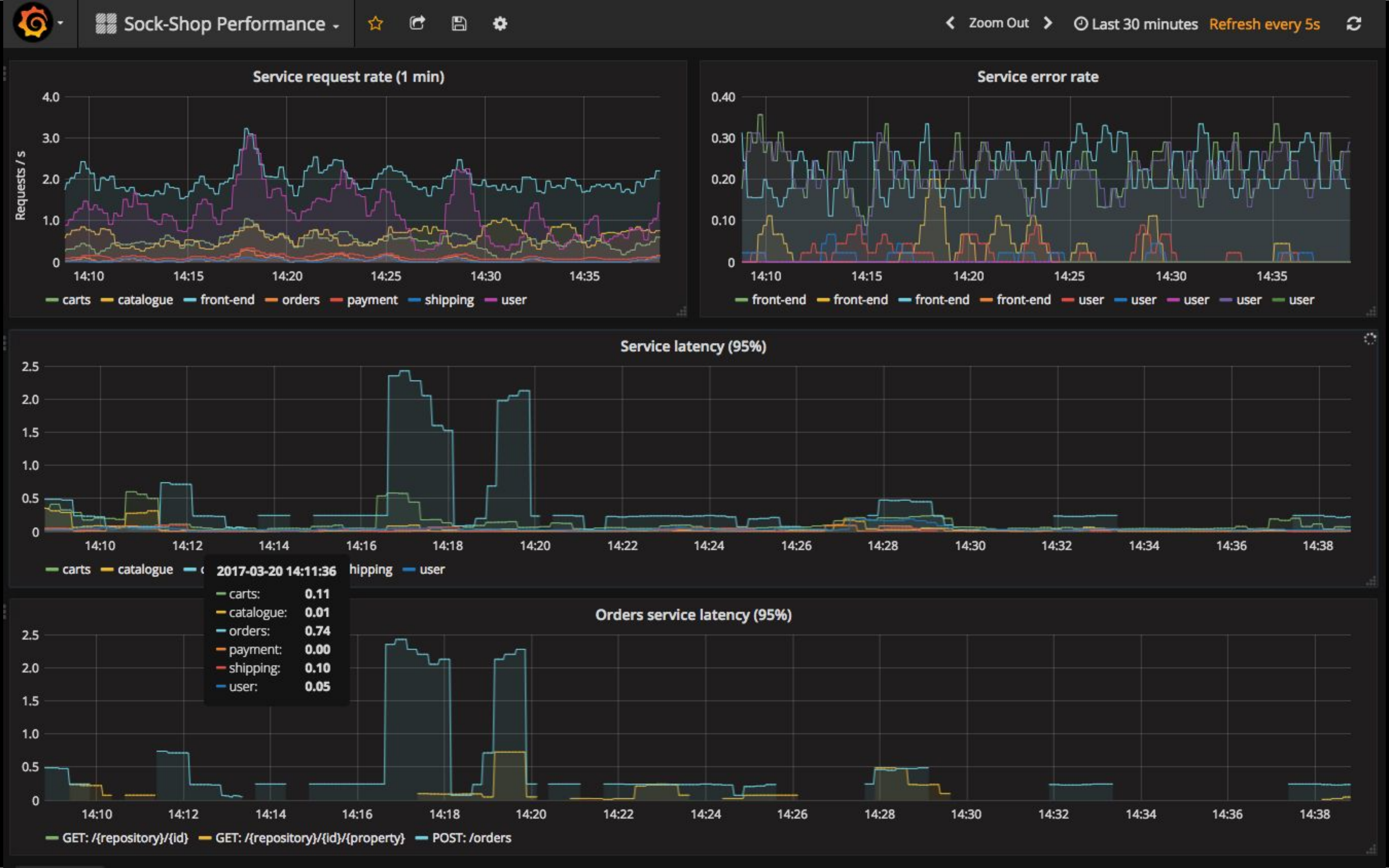
Element	Value
request_duration_seconds_count{instance="10.32.0.3:80",job="sock-shop/catalogue",method="GET",name="catalogue",route="youtube_2_jpeg",service="catalogue",status_code="200"}	883
request_duration_seconds_count{instance="10.32.0.4:80",job="sock-shop/user",method="POST",name="user",route="register",service="user",status_code="200"}	158
request_duration_seconds_count{instance="10.43.0.10:8079",job="sock-shop/front-end",method="delete",name="front-end",route="/cards/58d12a92ab613f00016d603f",service="front-end",status_code="200"}	1
request_duration_seconds_count{instance="10.32.0.3:80",job="sock-shop/catalogue",method="GET",name="catalogue",route="holy_2_jpeg",service="catalogue",status_code="200"}	886
request_duration_seconds_count{instance="10.32.0.3:80",job="sock-shop/catalogue",method="GET",name="catalogue",route="catalogue_size",service="catalogue",status_code="200"}	8
request_duration_seconds_count{instance="10.43.0.7:80",job="sock-shop/orders",method="GET",name="orders",route="/{repository}/{id}/{property}",service="orders",status_code="200"}	12607
request_duration_seconds_count{instance="10.43.0.7:80",job="sock-shop/orders",method="GET",name="orders",route="/health",service="orders",status_code="200"}	284060
request_duration_seconds_count{instance="10.43.0.10:8079",job="sock-shop/front-end",method="delete",name="front-end",route="/customers/58d12950ab613f00016d5ff7",service="front-end",status_code="200"}	1
request_duration_seconds_count{instance="10.32.0.3:80",job="sock-shop/catalogue",method="GET",name="catalogue",route="youtube_1_jpeg",service="catalogue",status_code="200"}	872
request_duration_seconds_count{instance="10.43.0.10:8079",job="sock-shop/front-end",method="post",name="front-end",route="/cards",service="front-end",status_code="200"}	70
request_duration_seconds_count{instance="10.43.0.7:80",job="sock-shop/orders",method="POST",name="orders",route="/orders",service="orders",status_code="201"}	17380
request_duration_seconds_count{instance="10.32.0.3:80",job="sock-shop/catalogue",method="GET",name="catalogue",route="catalogue_id",service="catalogue",status_code="200"}	63050
request_duration_seconds_count{instance="10.32.0.3:80",job="sock-shop/catalogue",method="GET",name="catalogue",route="health",service="catalogue",status_code="200"}	286172
request_duration_seconds_count{instance="10.43.0.10:8079",job="sock-shop/front-end",method="get",name="front-end",route="/basket.html",service="front-end",status_code="200"}	31
request_duration_seconds_count{instance="10.32.0.3:80",job="sock-shop/catalogue",method="GET",name="catalogue",route="tags",service="catalogue",status_code="200"}	34728
request_duration_seconds_count{instance="10.32.0.4:80",job="sock-shop/user",method="POST",name="user",route="cards",service="user",status_code="500"}	51
request_duration_seconds_count{instance="10.40.0.6:80",job="sock-shop/payment",method="GET",name="payment",route="health",service="payment",status_code="200"}	286160
request_duration_seconds_count{instance="10.32.0.3:80",job="sock-shop/catalogue",method="GET",name="catalogue",route="classic2_jpg",service="catalogue",status_code="200"}	889
request_duration_seconds_count{instance="10.43.0.10:8079",job="sock-shop/front-end",method="delete",name="front-end",route="/customers/58d12ae3ab613f00016d6051",service="front-end",status_code="200"}	1
request_duration_seconds_count{instance="10.43.0.10:8079",job="sock-shop/front-end",method="get",name="front-end",route="/catalogue/images/youtube_1.jpeg",service="front-end",status_code="200"}	1
request_duration_seconds_count{instance="10.40.0.7:80",job="sock-shop/carts",method="DELETE",name="carts",route="/{repository}/{id}",service="carts",status_code="202"}	12184
request_duration_seconds_count{instance="10.43.0.10:8079",job="sock-shop/front-end",method="delete",name="front-end",route="/cards/58d129b1ab613f00016d600e",service="front-end",status_code="200"}	1
request_duration_seconds_count{instance="10.40.0.7:80",job="sock-shop/carts",method="GET",name="carts",route="/health",service="carts",status_code="200"}	284062

Bonus: Grafana!

https://github.com/microservices-demo/microservices-demo/tree/master/deploy/kubernetes/manifests-monitoring

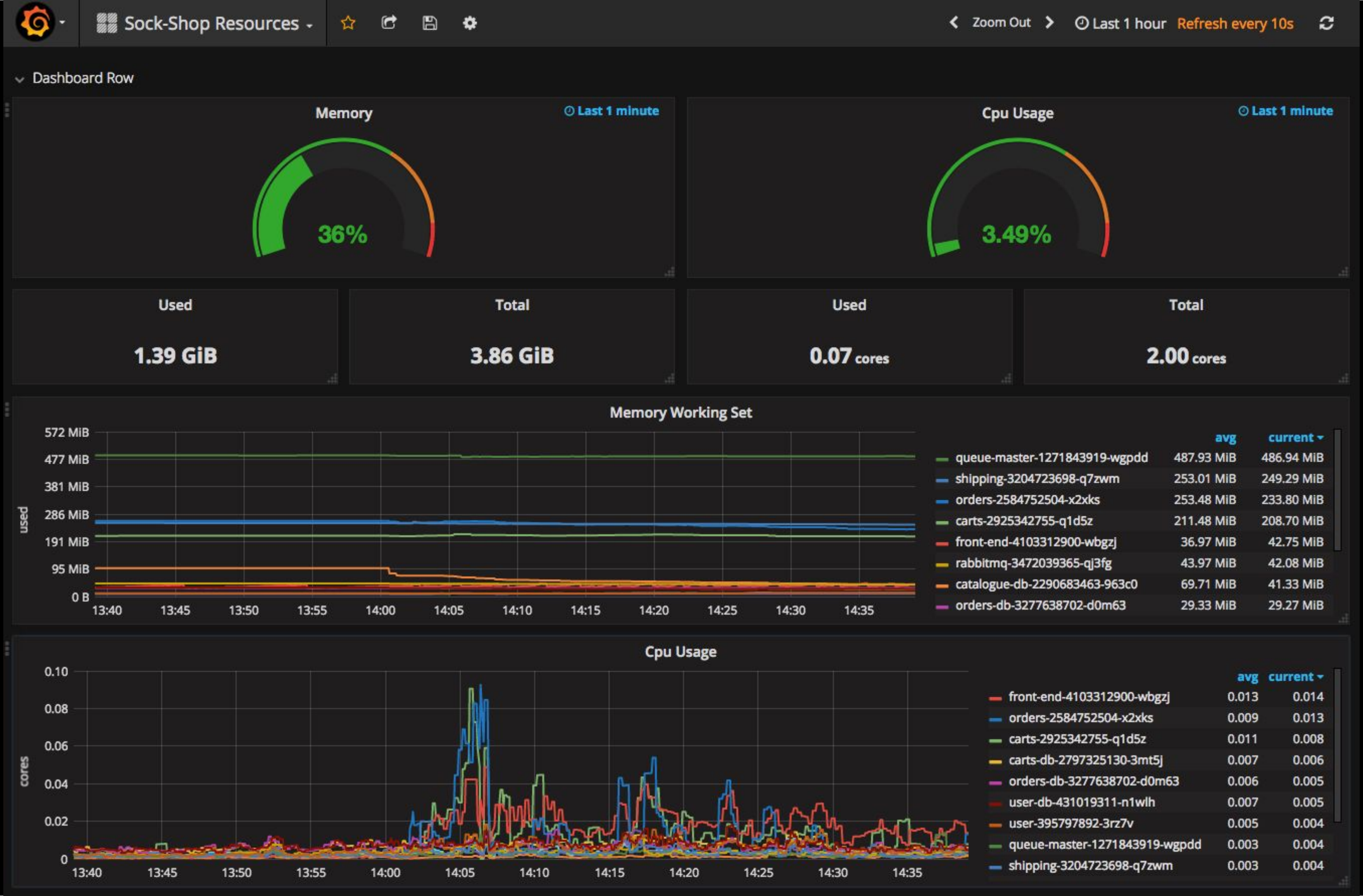
Comprises of:

- Grafana dependency & service
- ConfigMap with JSON Grafana dashboards inside
- Batch job to ingest and write the dashboards to a json file.
- Kubernetes state exporter
- Disk usage exporter



2017-03-20 14:11:36

carts:	0.11
catalogue:	0.01
orders:	0.74
payment:	0.00
shipping:	0.10
user:	0.05



Memory — Last 1 minute

36%

Cpu Usage — Last 1 minute

3.49%

Used

1.39 GiB

Total

3.86 GiB

Used

0.07 cores

Total

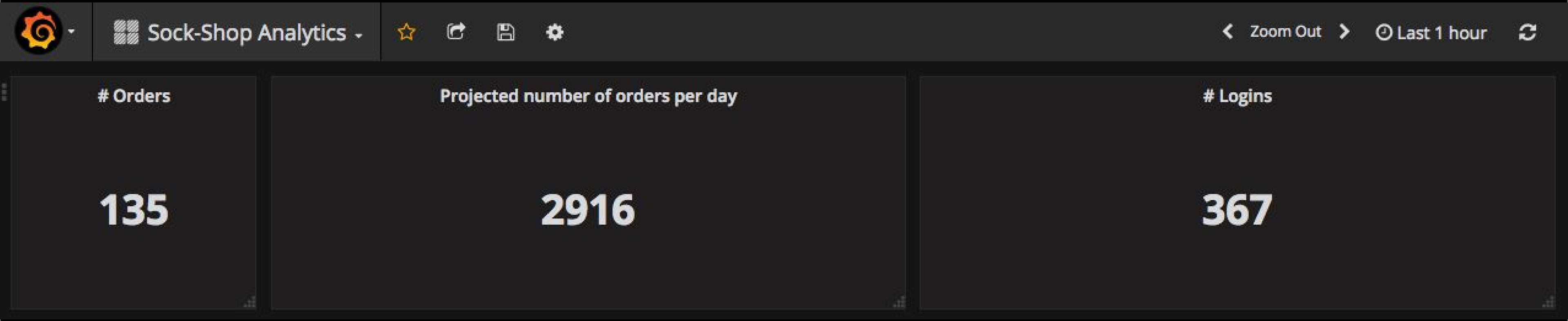
2.00 cores

Memory Working Set

	avg	current
🟩 queue-master-1271843919-wgpdd	487.93 MiB	486.94 MiB
🟦 shipping-3204723698-q7zwm	253.01 MiB	249.29 MiB
🟦 orders-2584752504-x2xks	253.48 MiB	233.80 MiB
🟩 carts-2925342755-q1d5z	211.48 MiB	208.70 MiB
🟥 front-end-4103312900-wbgzj	36.97 MiB	42.75 MiB
🟨 rabbitmq-3472039365-qj3fg	43.97 MiB	42.08 MiB
🟧 catalogue-db-2290683463-963c0	69.71 MiB	41.33 MiB
🟪 orders-db-3277638702-d0m63	29.33 MiB	29.27 MiB

Cpu Usage

	avg	current
🟥 front-end-4103312900-wbgzj	0.013	0.014
🟦 orders-2584752504-x2xks	0.009	0.013
🟩 carts-2925342755-q1d5z	0.011	0.008
🟨 carts-db-2797325130-3mt5j	0.007	0.006
🟪 orders-db-3277638702-d0m63	0.006	0.005
🟥 user-db-431019311-n1wlh	0.007	0.005
🟧 user-395797892-3rz7v	0.005	0.004
🟩 queue-master-1271843919-wgpdd	0.003	0.004
🟦 shipping-3204723698-q7zwm	0.003	0.004



Orders

135

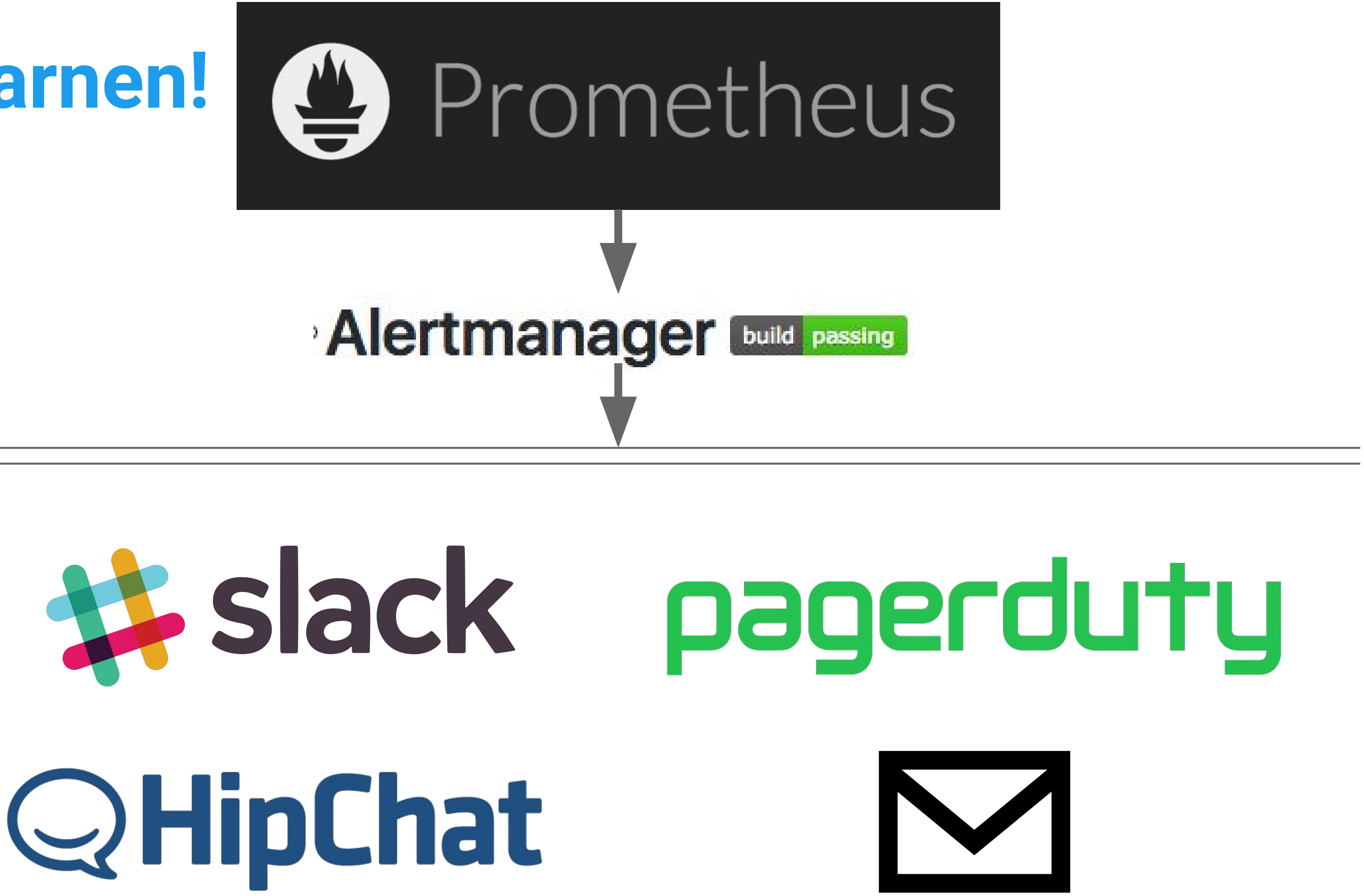
Projected number of orders per day

2916

Logins

367

Alert! Warnen!



Alertmanager manifest

https://github.com/microservices-demo/microservices-demo/tree/master/deploy/kubernetes/manifests-alerting

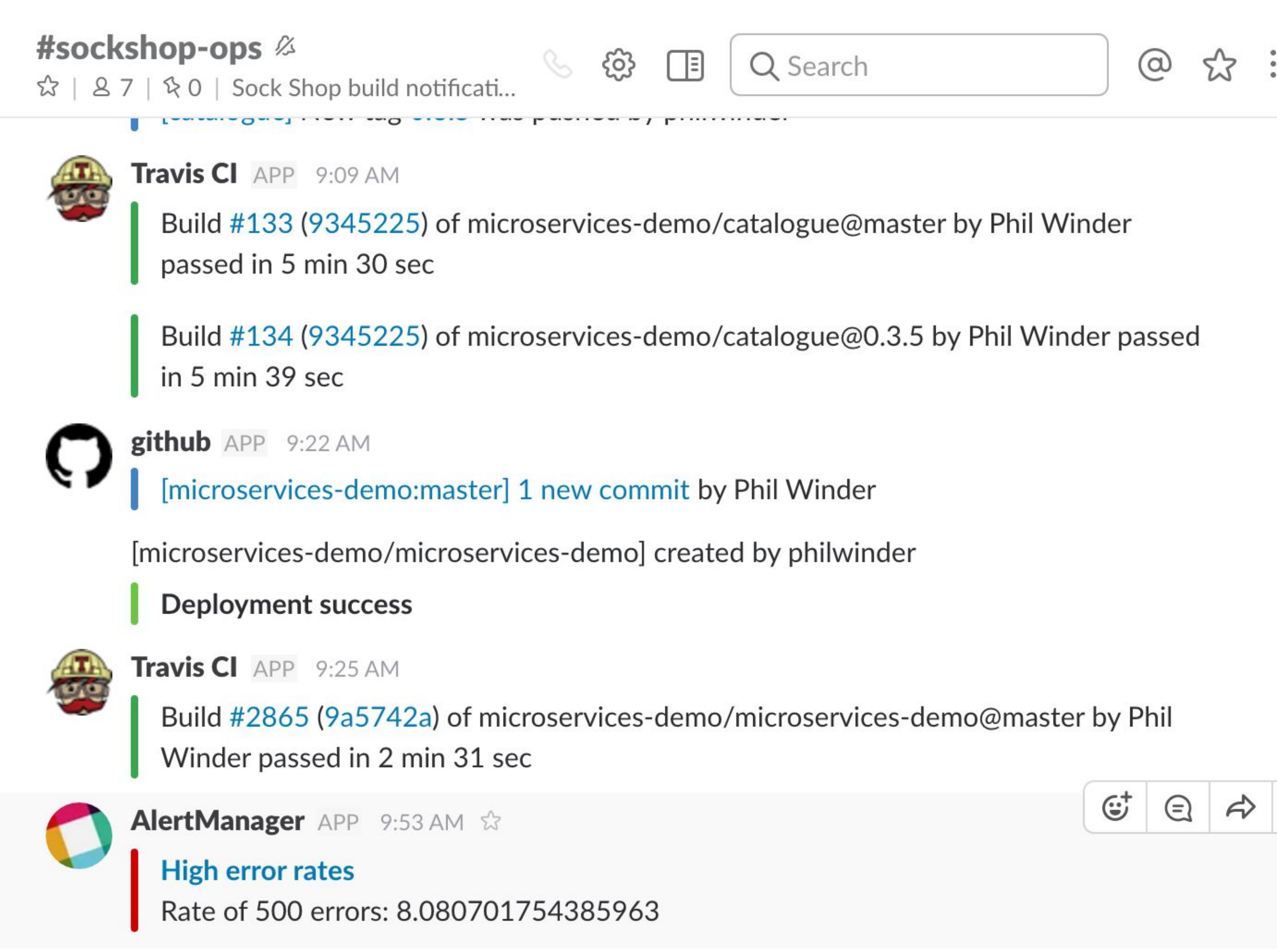
```
...
   spec:
     containers:
     - name: alertmanager
       image: prom/alertmanager:latest
       env:
         - name: SLACK_HOOK_URL
           valueFrom:
             secretKeyRef:
               name: slack-hook-url
              key: slack-hook-url
       command: ['/bin/sh',
'/etc/alertmanager/configure_secret.sh']
       args:
         -
'-config.file=/etc/alertmanager/config.yml'
         - '-storage.path=/alertmanager'
```

```
...
       ports:
       - name: alertmanager
         containerPort: 9093
       volumeMounts:
       - name: config-volume
         mountPath: /etc/alertmanager
     volumes:
     - name: config-volume
       configMap:
         name: alertmanager
```

Alert rules

https://github.com/microservices-demo/microservices-demo/blob/master/deploy/kubernetes/manifests-monitoring/prometheus-alertrules.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
 name: prometheus-alertrules
data:
 alert.rules: |
   # Alert for high error rate in the Sock Shop.
   ALERT HighErrorRate
     IF rate(request_duration_seconds_count{status_code="500"}[5m]) > 1
     FOR 5m
     LABELS { severity = "slack" }
     ANNOTATIONS {
       summary = "High HTTP 500 error rates",
       description = "Rate of HTTP 500 errors per 5 minutes: {{ $value }}",
     }
```



#sockshop-ops

☆ | 7 | 0 | Sock Shop build notificati...

[catalogue] New tag 0.3.5 was pushed by philwinder

Travis CI APP 9:09 AM

Build #133 (9345225) of microservices-demo/catalogue@master by Phil Winder passed in 5 min 30 sec

Build #134 (9345225) of microservices-demo/catalogue@0.3.5 by Phil Winder passed in 5 min 39 sec

github APP 9:22 AM

[microservices-demo:master] 1 new commit by Phil Winder

[microservices-demo/microservices-demo] created by philwinder

Deployment success

Travis CI APP 9:25 AM

Build #2865 (9a5742a) of microservices-demo/microservices-demo@master by Phil Winder passed in 2 min 31 sec

AlertManager APP 9:53 AM ☆

High error rates

Rate of 500 errors: 8.080701754385963

ContainerSolutions

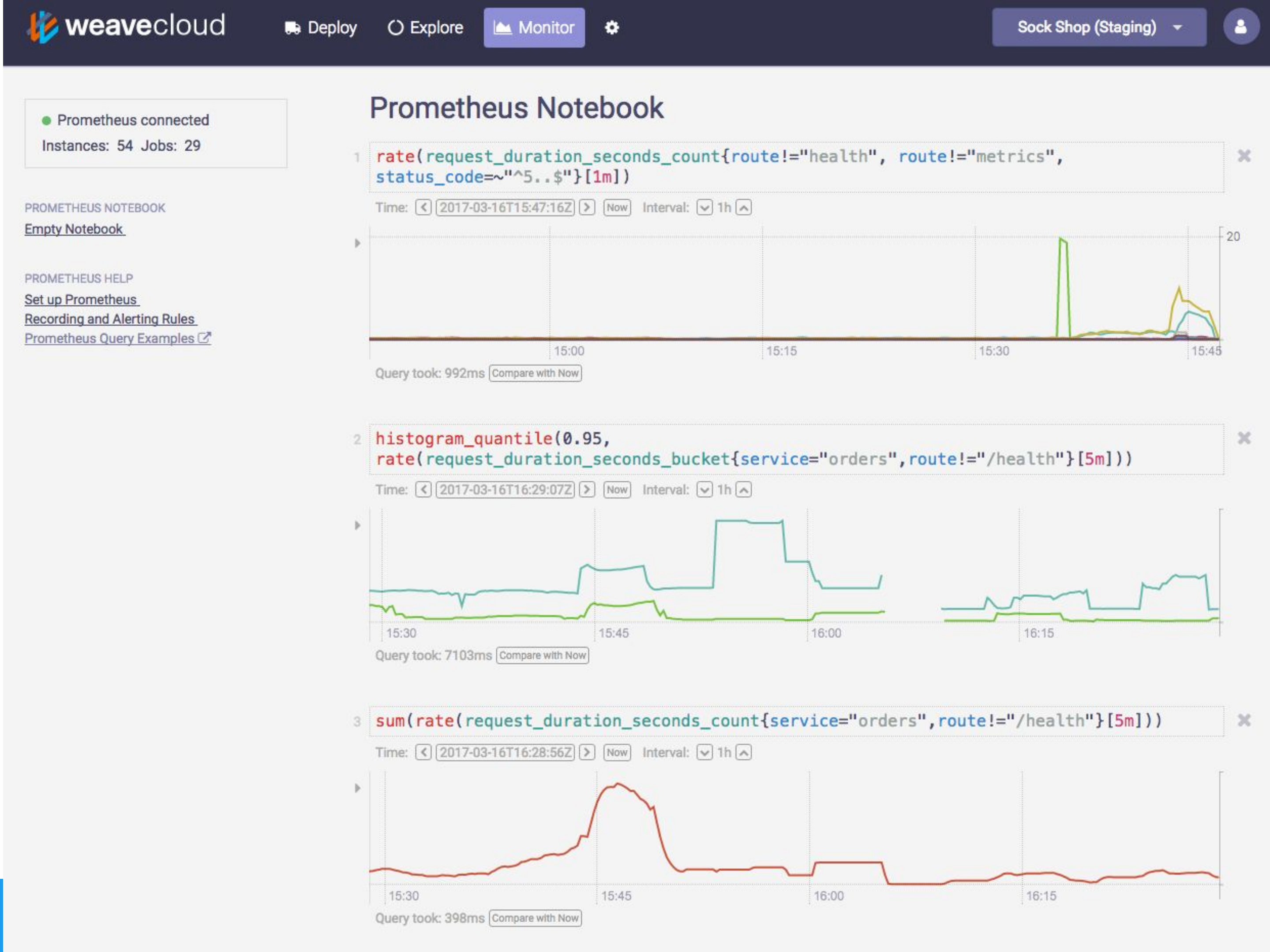
Summary

git.io/sock-shop

github.com/microservices-demo/**microservices-demo**

Weave Cloud

- Managed
- Multi-tenant
- Unlimited data retention
- WIP



● Prometheus connected

Instances: 54 Jobs: 29

PROMETHEUS NOTEBOOK

Empty Notebook

PROMETHEUS HELP

Set up Prometheus

Recording and Alerting Rules

Prometheus Query Examples

Prometheus Notebook

1

```
rate(request_duration_seconds_count{route!="health", route!="metrics",
status_code=~"^5..$"}[1m])
```

Time: 2017-03-16T15:47:16Z Now Interval: 1h

Query took: 992ms Compare with Now

2

```
histogram_quantile(0.95,
rate(request_duration_seconds_bucket{service="orders",route!="/health"}[5m]))
```

Time: 2017-03-16T16:29:07Z Now Interval: 1h

Query took: 7103ms Compare with Now

3

```
sum(rate(request_duration_seconds_count{service="orders",route!="/health"}[5m]))
```

Time: 2017-03-16T16:28:56Z Now Interval: 1h

Query took: 398ms Compare with Now

Phil Winder

Freelance Engineer
@DrPhilWinder
phil@WinderResearch.com

Winder Research

Container Solutions

weave~~works~~ socks