

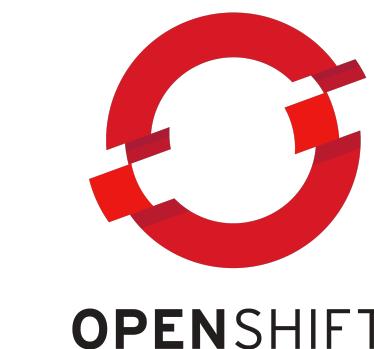


# Using Prometheus For Container Resource Consumption Estimates

Michael Hausenblas @mhausenblas

Developer Advocate at Red Hat

2017-08-17, PromCon, Munich



[github.com/openshift-demos/resorcerer](https://github.com/openshift-demos/resorcerer)

# The Problem



# Motivation

- Containers  $\cong$  cgroups + namespaces + CoW fs
- Resources a container consumes:  $\langle CPU, memory, GPU, disk, ports, \dots \rangle$
- Questions
  - How much memory does your app consume?
  - How much CPU time?
  - What about spikes in the traffic?

# Motivation

Unfortunately, Kubernetes has not yet implemented dynamic resource management, which is why we have to set resource limits for our containers. I imagine that at some point Kubernetes will start implementing a less manual way to manage resources, but this is all we have for now.

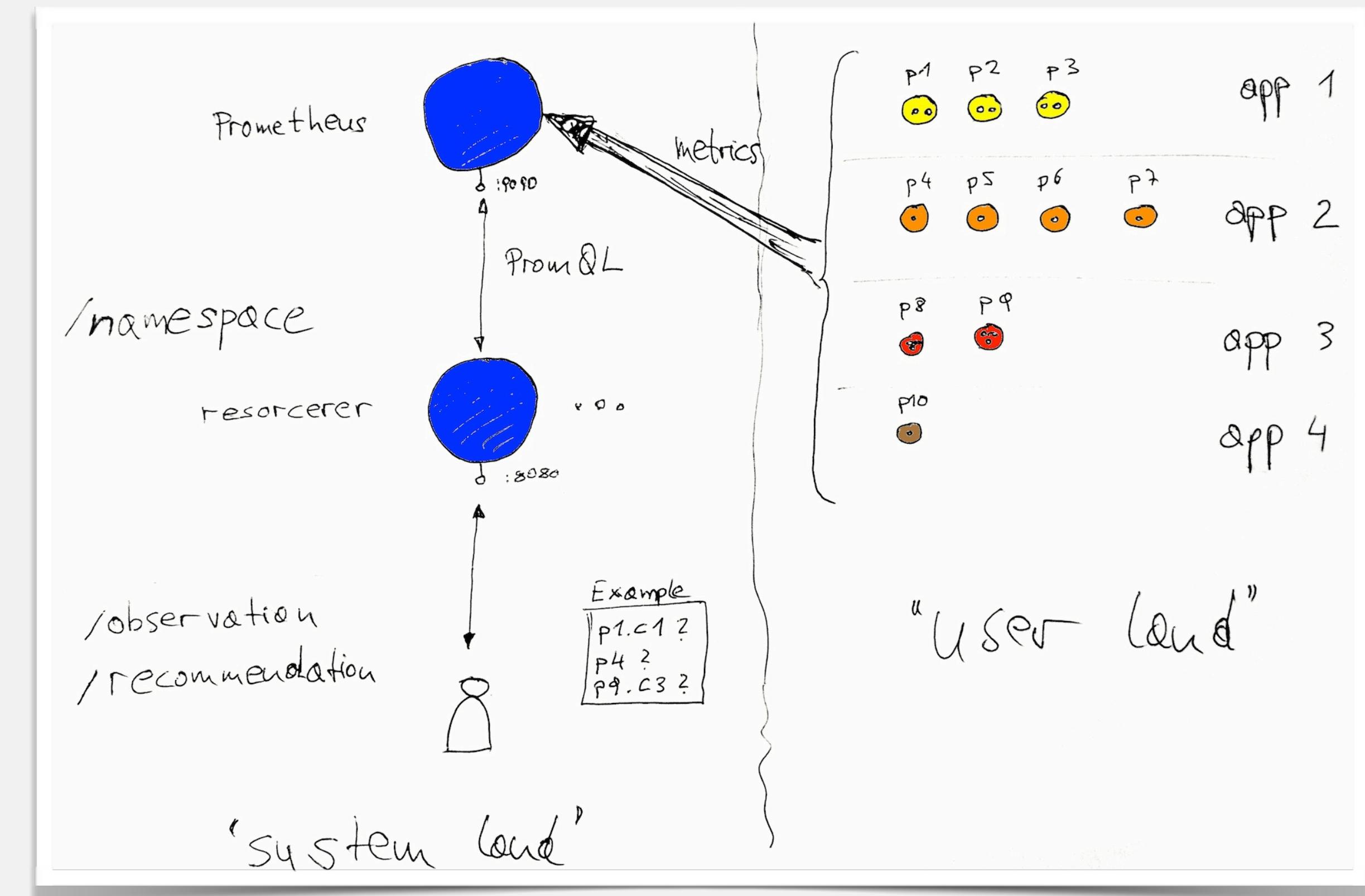
Ben Visser, 12/2016 in: [Kubernetes – Understanding Resources](#)

# Motivation

Kubernetes doesn't have dynamic resource allocation, which means that requests and limits have to be determined and set by the user. When these numbers are not known precisely for a service, a good approach is to start it with overestimated resources requests and no limit, then let it run under normal production load for a certain time.

Antoine Cotten, 05/2016 in: [1 year, lessons learned from a 0 to Kubernetes transition](#)

# resorcerer—a proof of concept



[hackernoon.com/container-resource-consumption-too-important-to-ignore-7484609a3bb7](https://hackernoon.com/container-resource-consumption-too-important-to-ignore-7484609a3bb7)

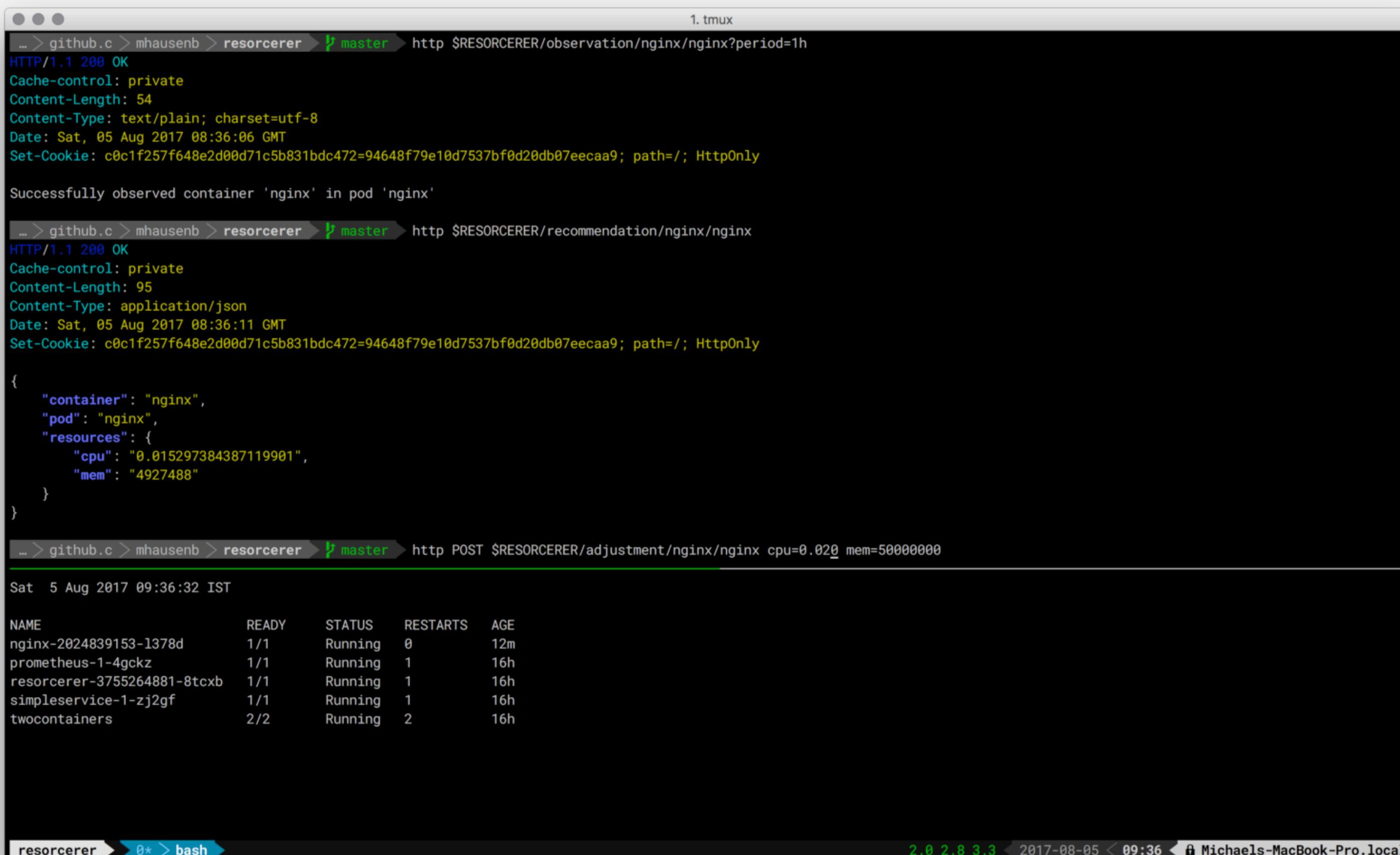
# PromQL examples used in resorcerer

```
sum(  
    rate(  
        container_cpu_usage_seconds_total{  
            container_name="nginx"} [30m]  
    )  
)
```

```
max_over_time(  
    container_memory_usage_bytes{  
        pod_name="nginx",  
        container_name="nginx"} [30m]  
)
```

[github.com/openshift-demos/resorcerer](https://github.com/openshift-demos/resorcerer)

# Demo time!



The screenshot shows a tmux session with multiple panes. The top pane displays the output of an curl command to check the observation endpoint for the 'nginx' container. It shows the response headers and a message indicating successful observation. The second pane shows the recommendation endpoint for the 'nginx' container, displaying a JSON object with resource usage information. The third pane shows a POST request to the adjustment endpoint, specifying a CPU limit of 0.020 and memory of 50000000. The bottom pane lists the status of various pods in the cluster.

```
... > github.c > mhausenb > resorcerer > master > http $RESORCERER/observation/nginx/nginx?period=1h
HTTP/1.1 200 OK
Cache-control: private
Content-Length: 54
Content-Type: text/plain; charset=utf-8
Date: Sat, 05 Aug 2017 08:36:06 GMT
Set-Cookie: c0c1f257f648e2d00d71c5b831bdc472=94648f79e10d7537bf0d20db07eecaa9; path=/; HttpOnly

Successfully observed container 'nginx' in pod 'nginx'

... > github.c > mhausenb > resorcerer > master > http $RESORCERER/recommendation/nginx/nginx
HTTP/1.1 200 OK
Cache-control: private
Content-Length: 95
Content-Type: application/json
Date: Sat, 05 Aug 2017 08:36:11 GMT
Set-Cookie: c0c1f257f648e2d00d71c5b831bdc472=94648f79e10d7537bf0d20db07eecaa9; path=/; HttpOnly

{
  "container": "nginx",
  "pod": "nginx",
  "resources": {
    "cpu": "0.015297384387119901",
    "mem": "4927488"
  }
}

... > github.c > mhausenb > resorcerer > master > http POST $RESORCERER/adjustment/nginx/nginx cpu=0.020 mem=50000000

Sat 5 Aug 2017 09:36:32 IST

NAME          READY   STATUS    RESTARTS   AGE
nginx-2024839153-1378d   1/1     Running   0          12m
prometheus-1-4gckz       1/1     Running   1          16h
resorcerer-3755264881-8tcxb 1/1     Running   1          16h
simpleservice-1-zj2gf     1/1     Running   1          16h
twocontainers            2/2     Running   2          16h

resorcerer > 0* > bash > 2.0 2.8 3.3 < 2017-08-05 < 09:36 < Michaels-MacBook-Pro.local
```

Video walkthrough of the resorcerer demo



redhat<sup>®</sup>

# openshift.com



[plus.google.com/+RedHat](https://plus.google.com/+RedHat)



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[twitter.com/RedHatNews](https://twitter.com/RedHatNews)