# Performance Testing Microservices

Experience Report

TOOP, TAV 45, 06.11.2020

David Faragó, EclipseSource & QPR

# Contents

# Performance

## KPIs

- service-oriented
  - response time (s)
  - availability (%)
  - error-rate (%)
- efficiency-oriented
  - throughput (Hz)
  - utilization (%)

1 second more
⇒ 11% fewer page views
⇒ 7% fewer conversion

most important: perception by user

# Performance Tests

Stress

Load

Endurance

## Web Load Test

| Scope | Inside Out | Middle In | Outside In |
|---|---|---|---|
| Entry Point | 1.0 | 1.5 | 2.0 |

# Test Execution

- create (increasing) load
- measure KPIs
- compute statistics

- manual (UX) tests
- run other tests
- observe tests, logs, monitoring

# Test Execution Demo

- create (increasing) load

Via Locust (Python)

- Layer 1 User profiles
- Layer 2: Journeys
- Layer 3: Transactions

- measure KPIs

Infrastructure:

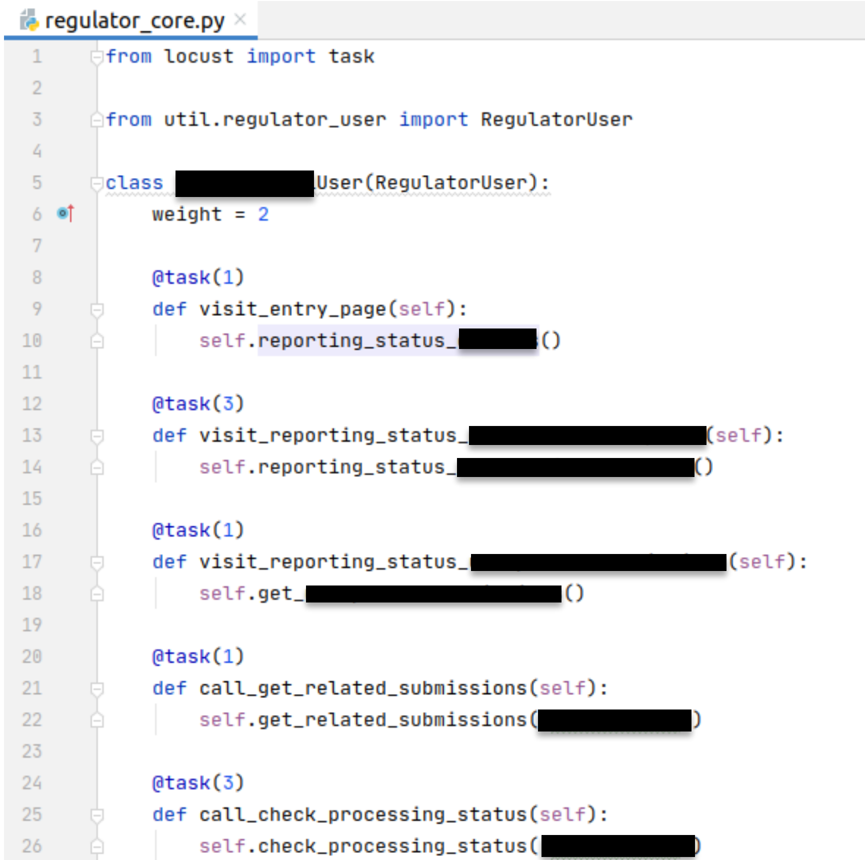
- Docker Locust master
- Docker Locust exporter
- Helm Service (Monitor)

- compute statistics

via Grafana and Prometheus:

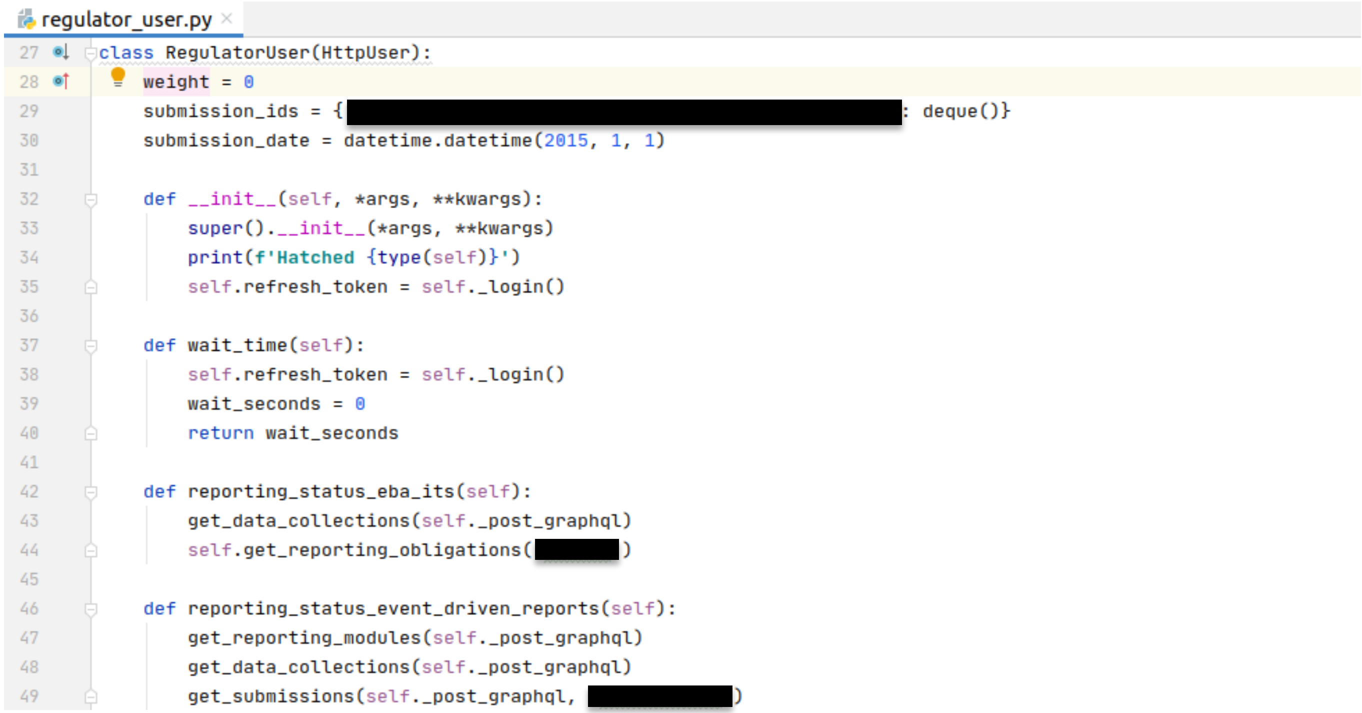
- store
- compute
- visualize

# Create Load via Locust: User Profiles



```
from locust import task

from util.regulator_user import RegulatorUser

class [redacted]User(RegulatorUser):
    weight = 2

    @task(1)
    def visit_entry_page(self):
        self.reporting_status_[redacted]()

    @task(3)
    def visit_reporting_status_[redacted](self):
        self.reporting_status_[redacted]()

    @task(1)
    def visit_reporting_status_[redacted](self):
        self.get_[redacted]()

    @task(1)
    def call_get_related_submissions(self):
        self.get_related_submissions([redacted])

    @task(3)
    def call_check_processing_status(self):
        self.check_processing_status([redacted])
```

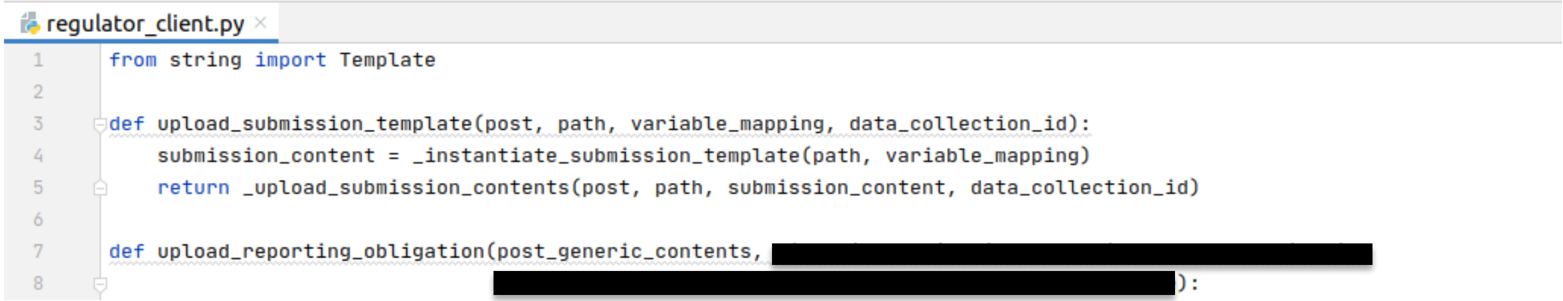
# Create Load via Locust: Journeys



```python
class RegulatorUser(HttpUser):
    weight = 0
    submission_ids = {[redacted]: deque()}
    submission_date = datetime.datetime(2015, 1, 1)

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        print(f'Hatched {type(self)}')
        self.refresh_token = self._login()

    def wait_time(self):
        self.refresh_token = self._login()
        wait_seconds = 0
        return wait_seconds

    def reporting_status_eba_its(self):
        get_data_collections(self._post_graphql)
        self.get_reporting_obligations([redacted])

    def reporting_status_event_driven_reports(self):
        get_reporting_modules(self._post_graphql)
        get_data_collections(self._post_graphql)
        get_submissions(self._post_graphql, [redacted])
```

...

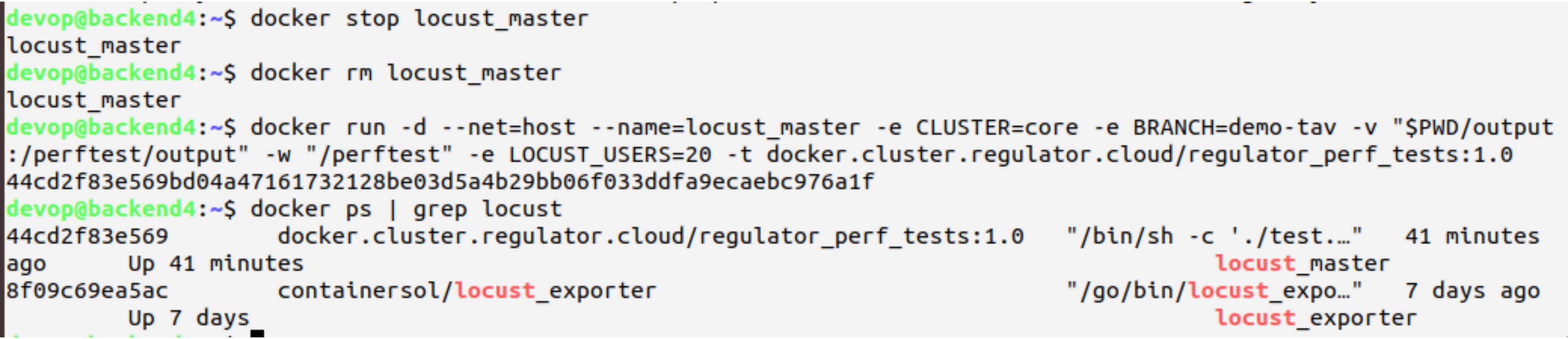
# Create Load via Locust: Transactions



```python
from string import Template

def upload_submission_template(post, path, variable_mapping, data_collection_id):
    submission_content = _instantiate_submission_template(path, variable_mapping)
    return _upload_submission_contents(post, path, submission_content, data_collection_id)

def upload_reporting_obligation(post_generic_contents, [redacted]
                                ):
```

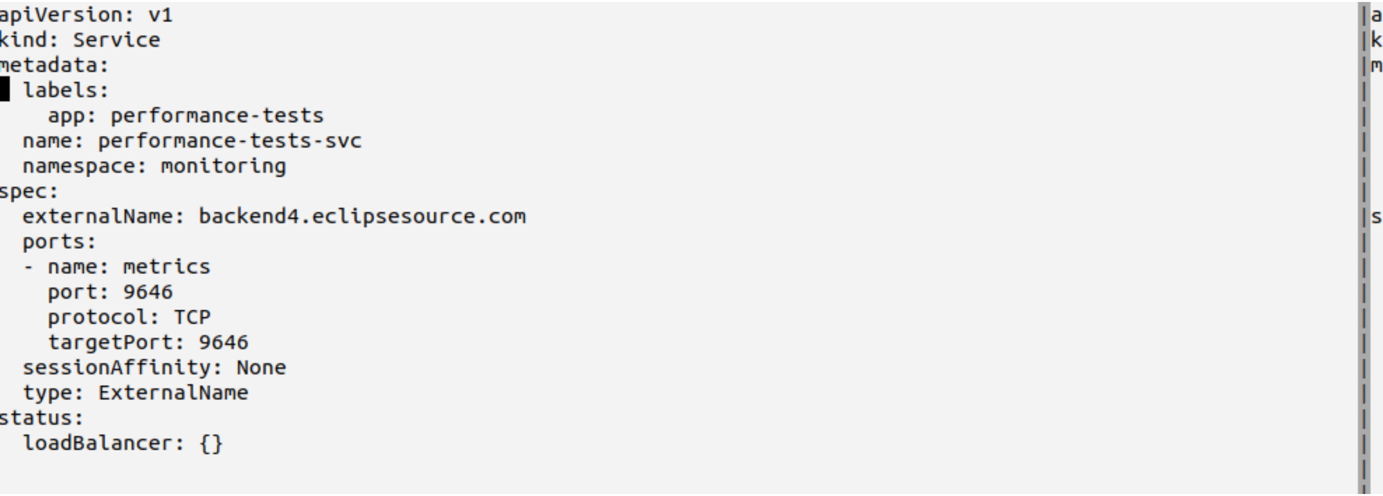
...

# Measure KPIs: Docker Infrastructure

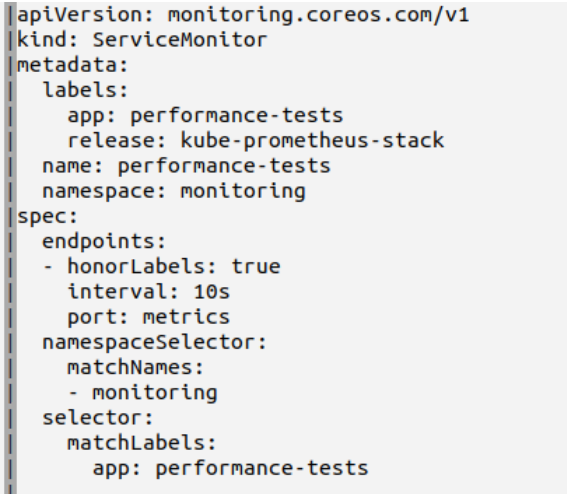


```
devop@backend4:~$ docker stop locust_master
locust_master
devop@backend4:~$ docker rm locust_master
locust_master
devop@backend4:~$ docker run -d --net=host --name=locust_master -e CLUSTER=core -e BRANCH=demo-tav -v "$PWD/output
:/perftest/output" -w "/perftest" -e LOCUST_USERS=20 -t docker.cluster.regulator.cloud/regulator_perf_tests:1.0
44cd2f83e569bd04a47161732128be03d5a4b29bb06f033ddfa9ecaebc976a1f
devop@backend4:~$ docker ps | grep locust
44cd2f83e569        docker.cluster.regulator.cloud/regulator_perf_tests:1.0   "/bin/sh -c './test.…"   41 minutes
ago      Up 41 minutes                                                                   locust_master
8f09c69ea5ac        containersol/locust_exporter                              "/go/bin/locust_expo…"   7 days ago
         Up 7 days                                                                       locust_exporter
```

# Measure KPIs: Helm Service (Monitor) Infrastructure

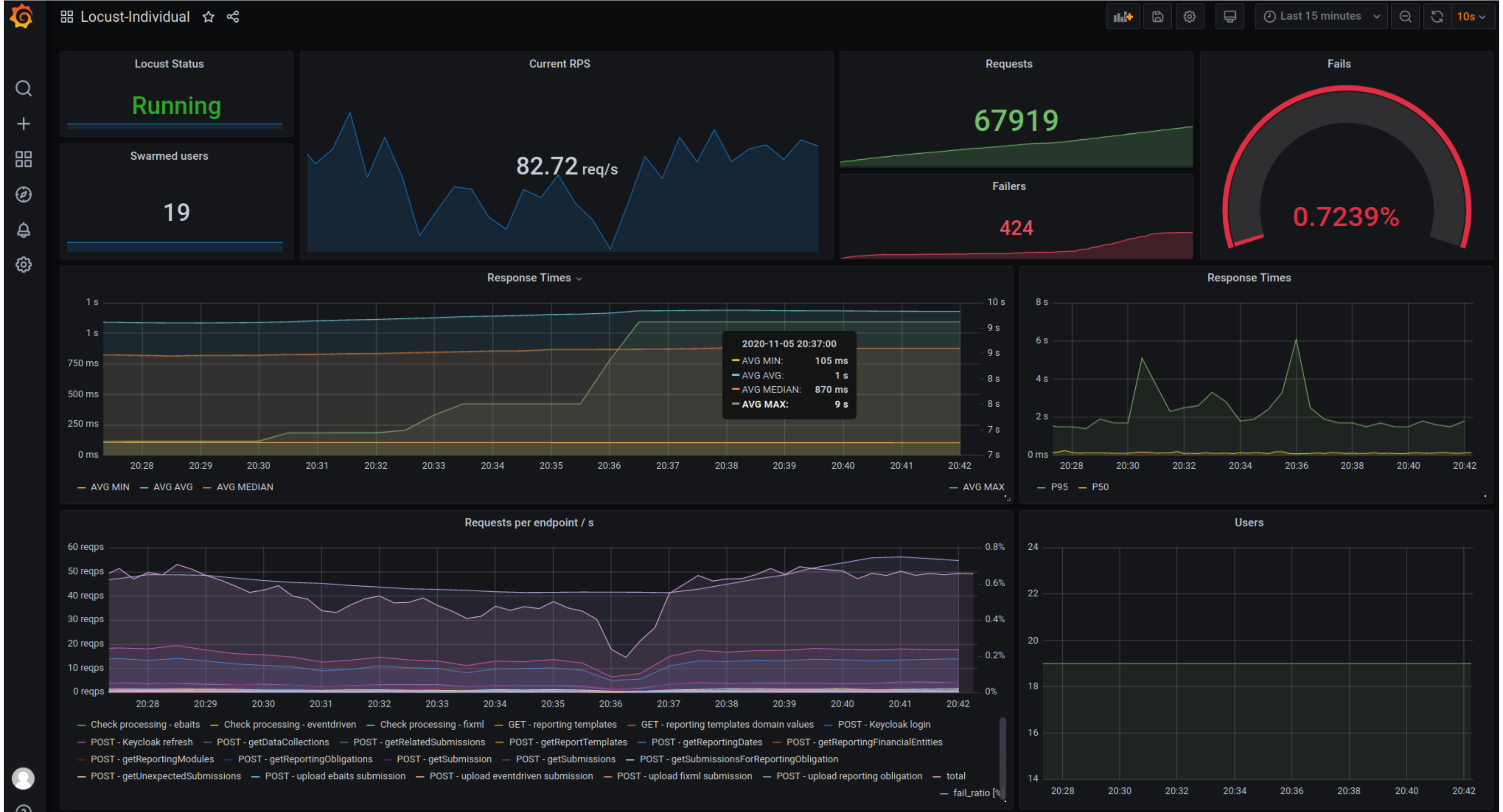


```yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app: performance-tests
  name: performance-tests-svc
  namespace: monitoring
spec:
  externalName: backend4.eclipsesource.com
  ports:
  - name: metrics
    port: 9646
    protocol: TCP
    targetPort: 9646
  sessionAffinity: None
  type: ExternalName
status:
  loadBalancer: {}
```

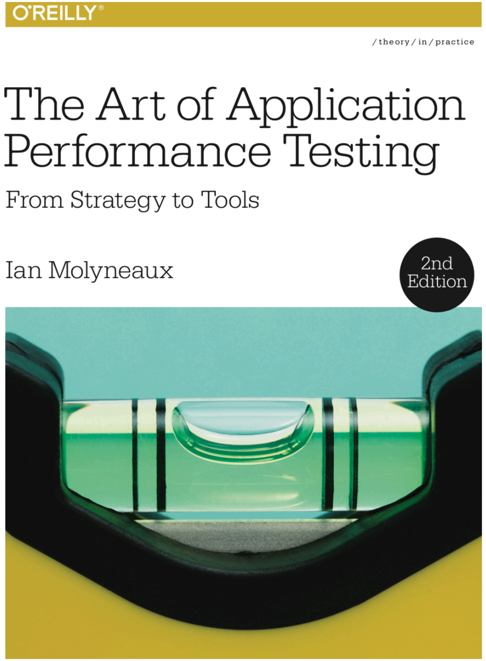


```yaml
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app: performance-tests
    release: kube-prometheus-stack
  name: performance-tests
  namespace: monitoring
spec:
  endpoints:
  - honorLabels: true
    interval: 10s
    port: metrics
  namespaceSelector:
    matchNames:
    - monitoring
  selector:
    matchLabels:
      app: performance-tests
```

# Compute Statistics via Grafana



# References



O'REILLY®

/ theory / in / practice

The Art of Application Performance Testing

From Strategy to Tools

Ian Molyneaux

2nd Edition

International Software Testing Qualifications Board

ISTQB®

**Certified Tester**

**Foundation Level Specialist Syllabus Performance Testing**

Version 2018

Provided by

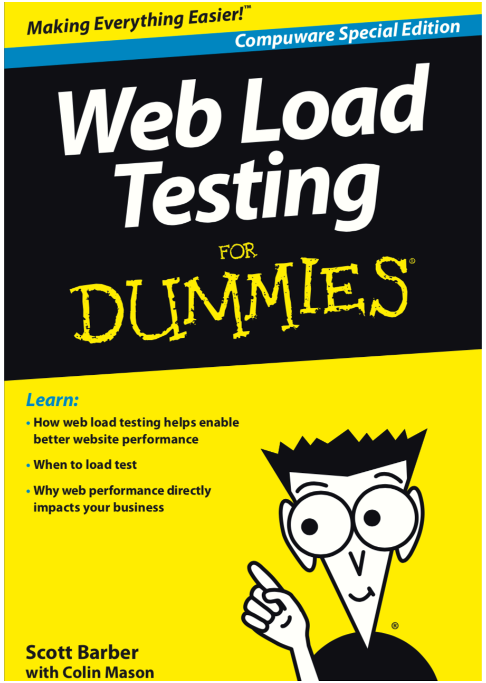
American Software Testing Qualifications Board

and

German Testing Board

ASTQB American Software Testing Qualifications Board, Inc.

GTB German Testing Board Software.Testing.Excellence.



Making Everything Easier!™

Compuware Special Edition

Web Load Testing FOR DUMMIES®

**Learn:**

- How web load testing helps enable better website performance
- When to load test
- Why web performance directly impacts your business

Scott Barber with Colin Mason