
Kubernetes
Performance
with k6

Almudena Vivanco
(@MrsDaehin)

```
import random
import time
import requests

class Speaker(object):
        def __init__(self):
                born = "Avilés - Asturias"
                studies = "Applied Maths and Computability"
                jobdescription = "Performance Jedi"
                company = "SCRM"
                project = "LidlPlus"
                talks = [Commit Conf, VLC Testing, Devops BCN, DevopsDays,
                WebPerfDays, Velocity]
        def talk(self):
                start_timer = time.time()
                r = requests.get('http://www.slideshare.net/almudenavivanco')
                r.raw.read()
                latency = time.time() - start_timer
                self.custom_timers[.net Conf] = latency
 if __name__ == '__main__':
        speech = Speaker()
        speech.talk()
        print trans.custom_timers
```

Success story: Lidl Plus rollout

Rollout timeline

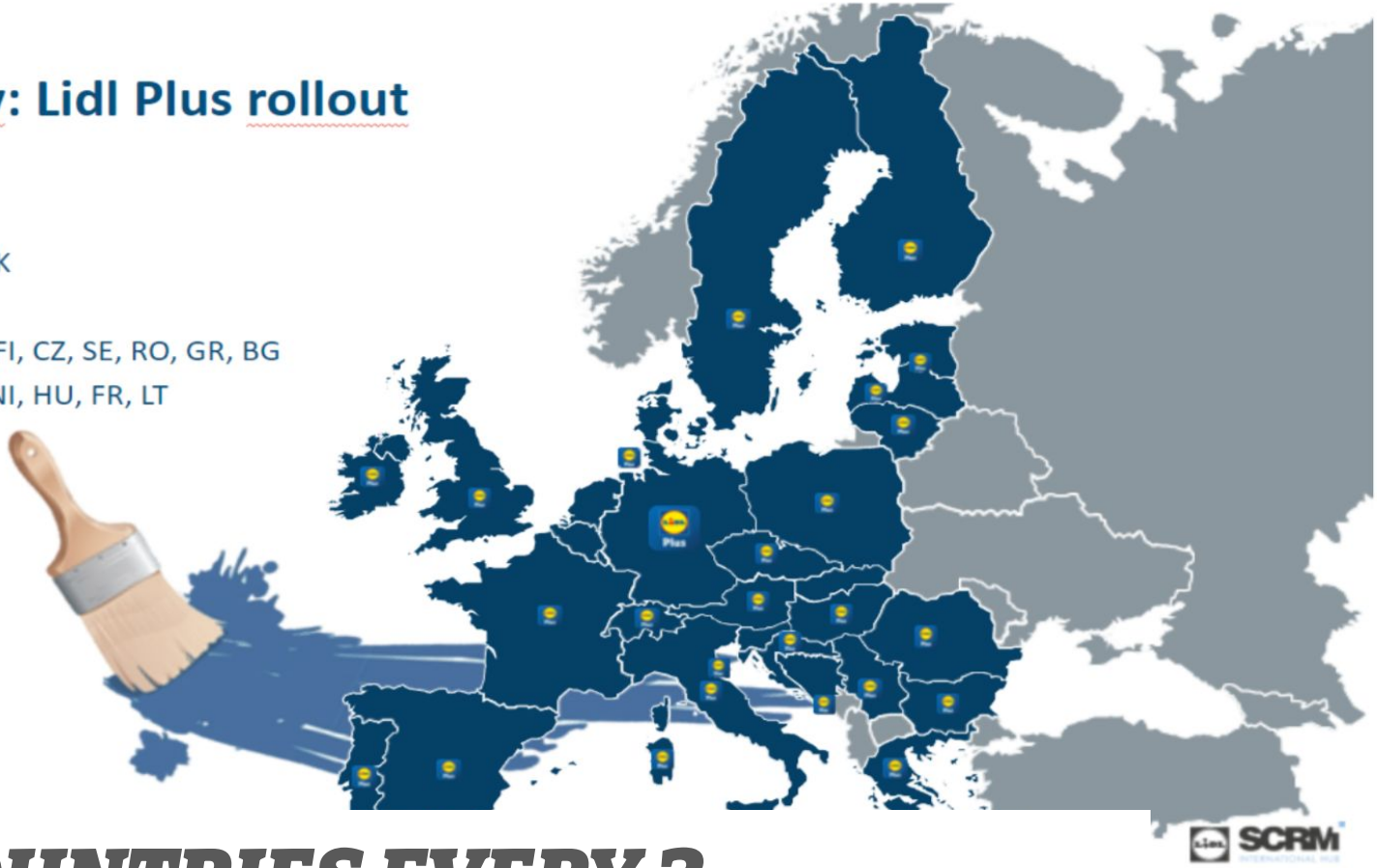
Pilot countries: ES, AT, DK

2019: PL, NL, DE

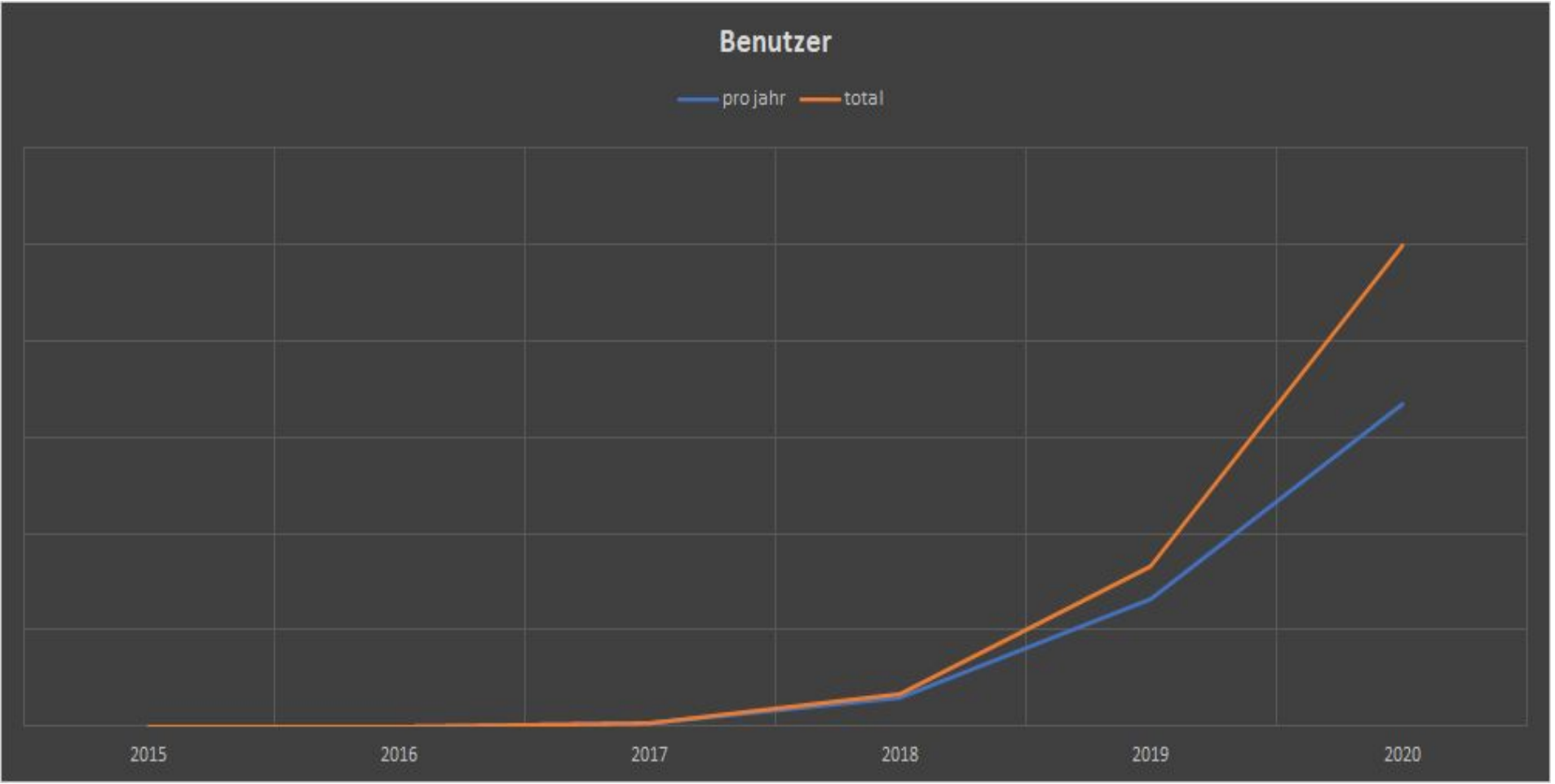
2020: SK, BE, LU, GB, IE, FI, CZ, SE, RO, GR, BG

2021: IT, HR, CY, PT, CH, NI, HU, FR, LT

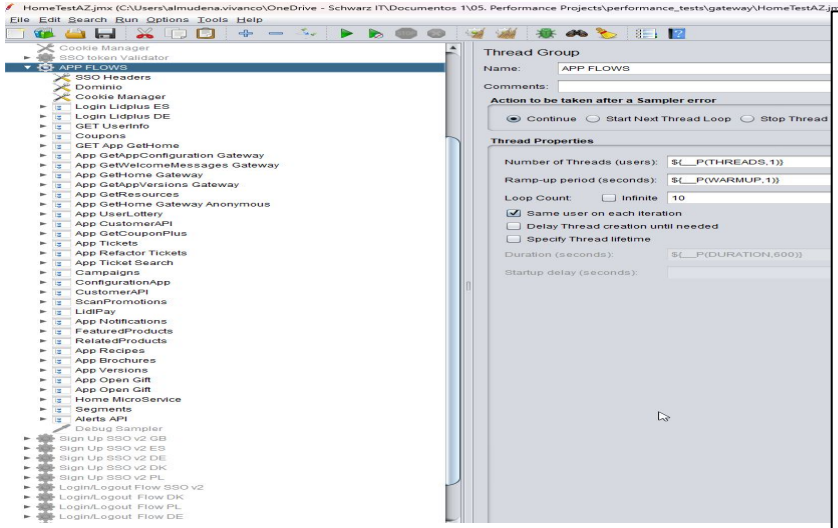
2022: RS, SI, MT, EE, LV

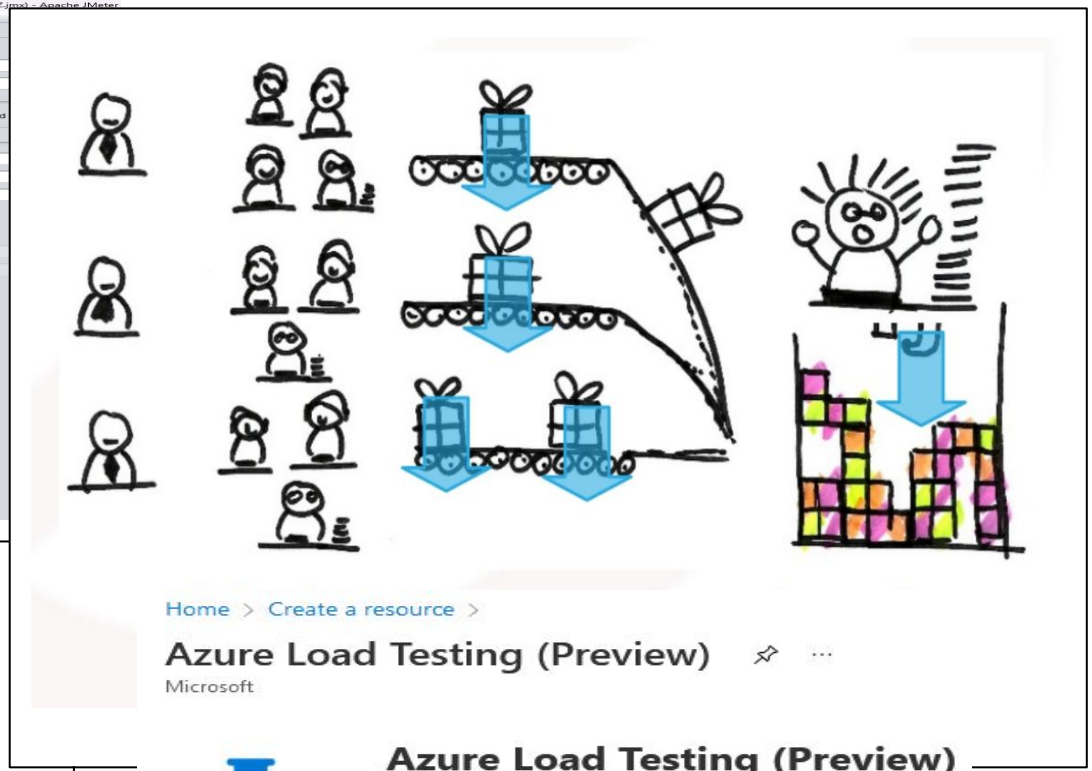


3 COUNTRIES EVERY 3 MONTHS



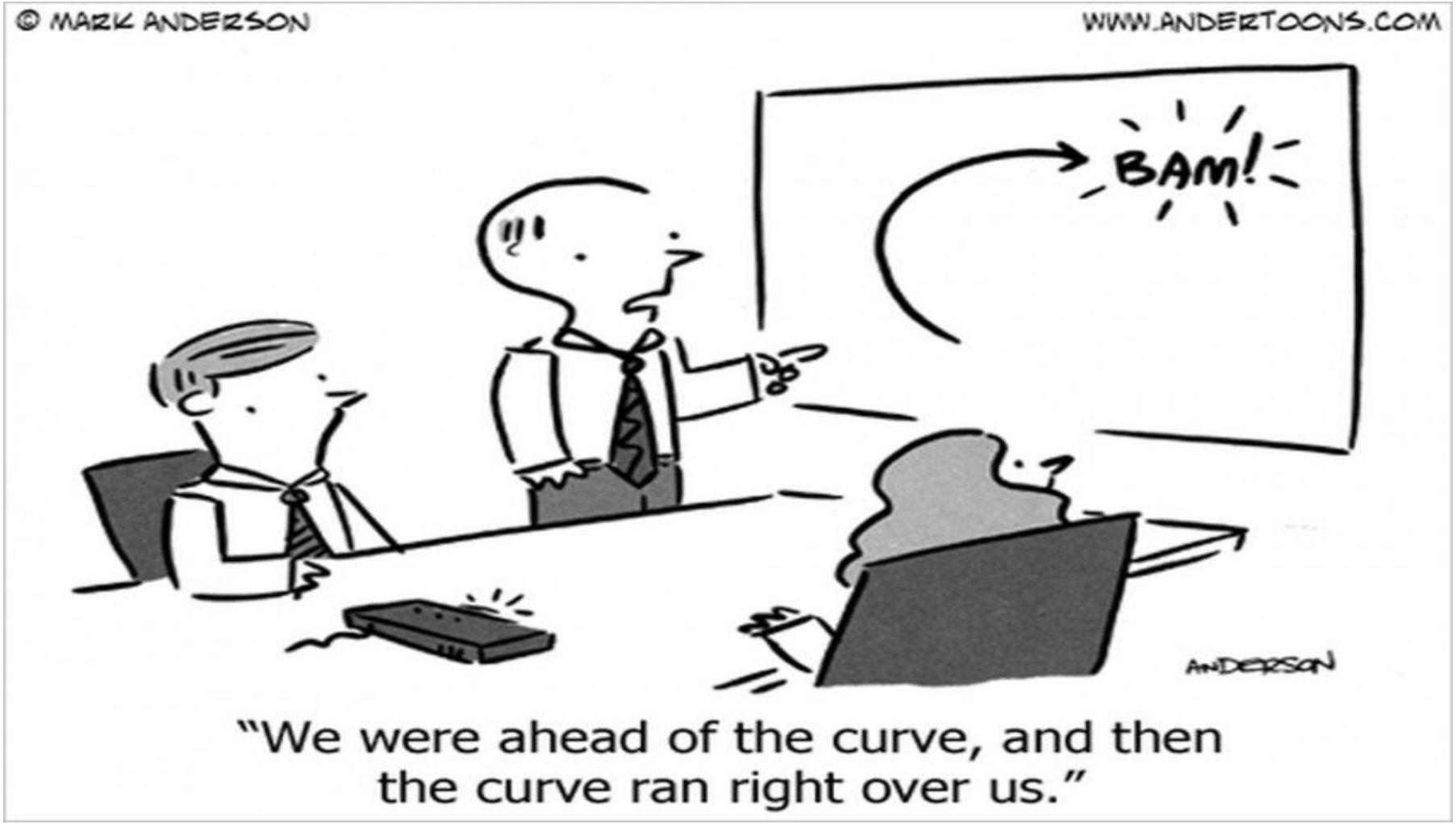
"For tomorrow" is an eternity





- Each microservice in Jmeter
- 3 Virtual Machines in Azure
- Everything centralized in 1 person

That was possible while we were less than 9 squads …


© MARK ANDERSON
WWW.ANDERTOONS.COM
BAM!
ANDERSON
"We were ahead of the curve, and then the curve ran right over us."

Why should you use k6?

Developer Friendly

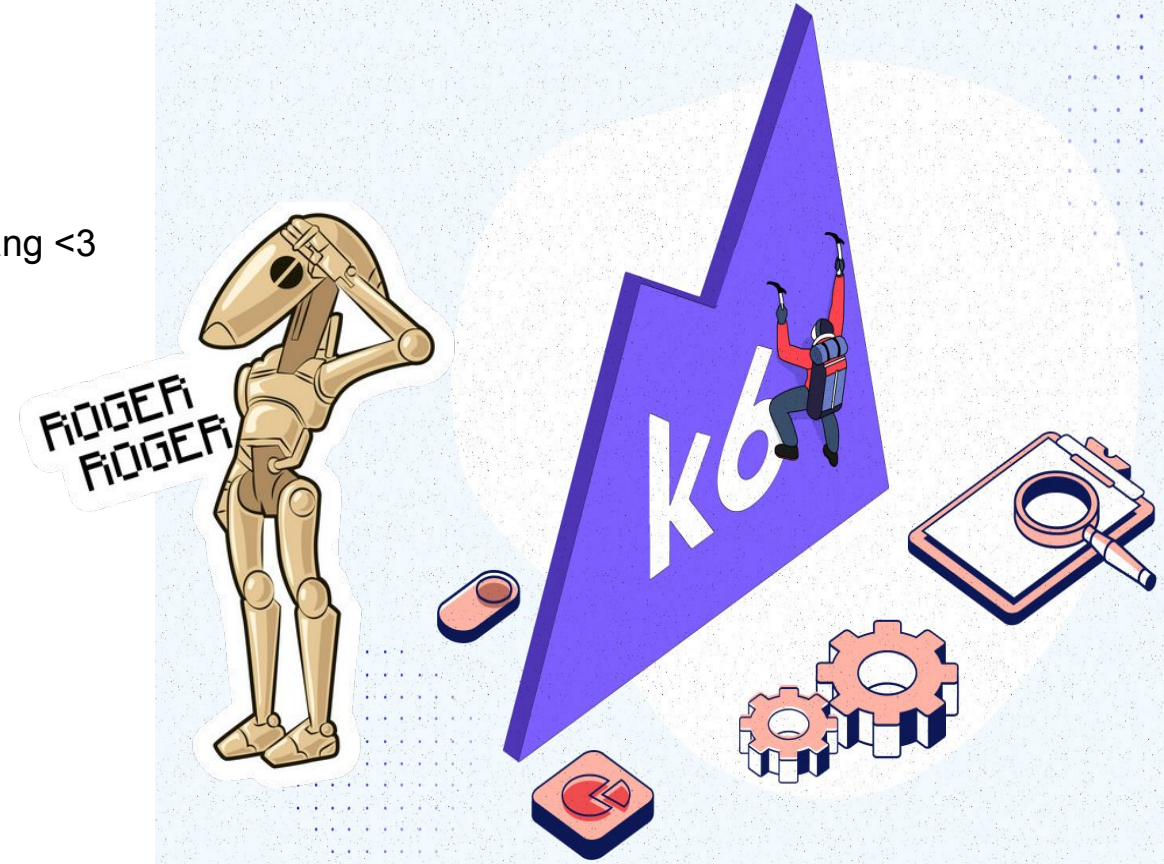
Native Concurrency Golang <3

CI/CD Integration

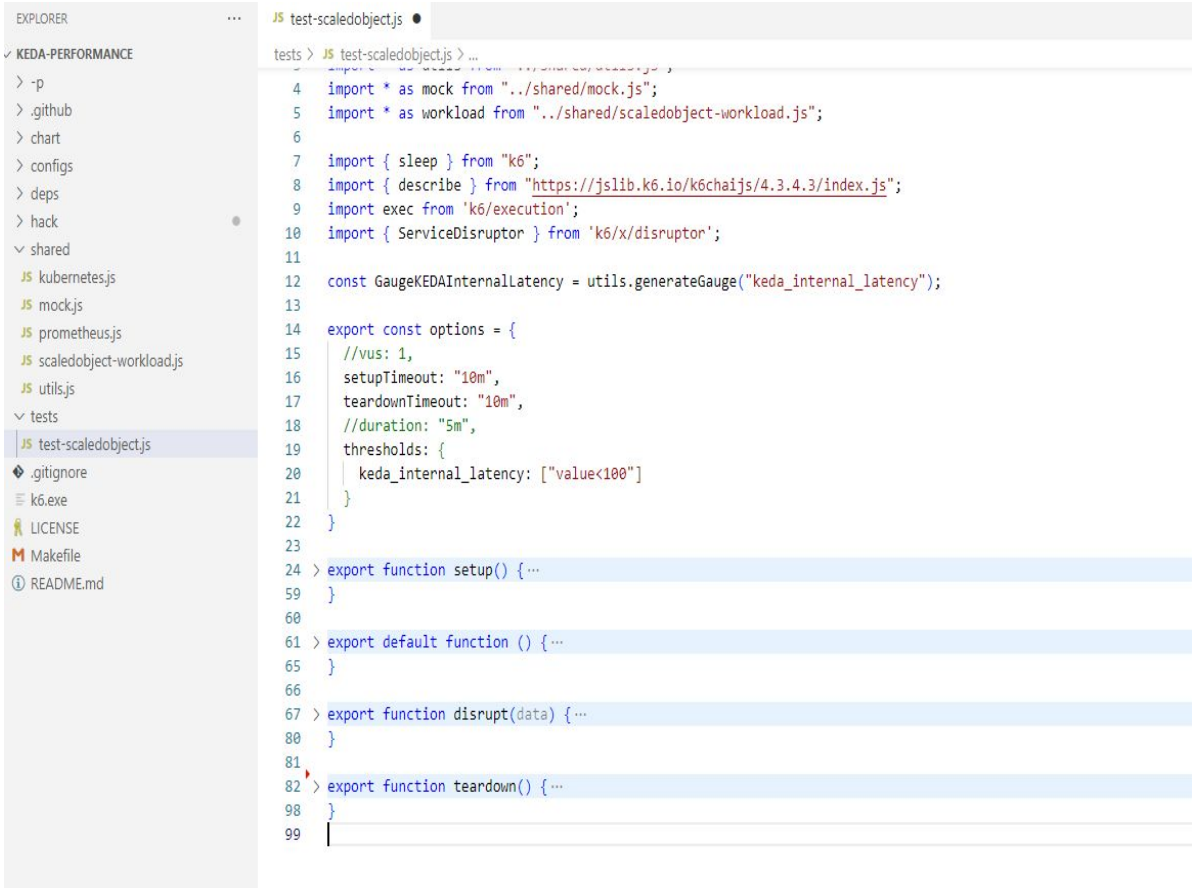
Analysis and monitoring

Horizontally Scaling

Extensibility



“Load” Test Structure




```
// 1 . Init Code
Options {
}

export function setup() {

// 2.- Setup code
}

export default function (data) {

// 3 .- VU code
}

export function teardown() {

// 4.- teardown code
}

Export function disrupt() {

// 5.- Chaos Behaviour
}
```

“Load” Test Structure

idp-performance-k6 > src > workloads > JS load.js > ...

```
55  export default function (data) {
70      group('Create a new user', () => {
73          newUserId = createUserResponse.json()['id'];
74          check(createUserResponse, {···
80          });
81      });
82
83      group('Create a new client', () => {···
94      });
95
96      if (__ITER % 2 === 0) {
97          group('List users', () => {···
107         });
108     } else {
109         group('List clients', () => {···
119         });
120     }
121
122     group('Delete user', () =>{
123         let deleteUserResponse = user.delete(newUserId)
124         let deleteUserTimingDuration = deleteUserResponse.timings.duration;
125         check(deleteUserResponse, {···
131         });
132     });
133
134     group('Delete client', () => {
135         let deleteClientResponse = client.delete(newClientId);
136         let deleteClientTimingDuration = deleteClientResponse.timings.duration;
137         check(deleteClientResponse, {···
143         });
144     });
145 }
146
147 export function teardown(data){
148     let user = new User(data.token);
149     for (const userId of data.usersAdminIds) {
150         group('Delete admin user', () =>{
151             let deleteUserResponse = user.delete(userId)
152             check(deleteUserResponse, {···
158             });
159         });
160     }
161 }
```

Logic is in the main.js

All checks are in the main.js

Project Organization in K6

EXPLORER

QA.IDL.PLUS.COUPONS.PERFORMANCE.TESTS

- bff-app
  - activate-promotion.js
  - deactivate-promotion.js
  - get-promotionshome.js
  - get-promotionslist.js
- configs
  - 1reqpersecondconf.json
  - 1reqs-with-faults.json
  - 100reqpersecond.json
  - 375reqpersecond.json
  - 600reqpersecondconf.json
  - 600reqs-with-faults.json
  - 750reqpersecond.json
  - 750reqs-with-faults.json
- constants
  - baseurls.js
  - mockserverBaseUrls.js
- mock-server
  - gk
  - upe
- pipeline
  - pipeline.yaml
- shared
  - jwt-generator.js
- use-cases
  - app-full-workflow-get-promotion.js
  - app-full-workflow.js
- .gitignore
- package-lock.json
- package.json
- README.md

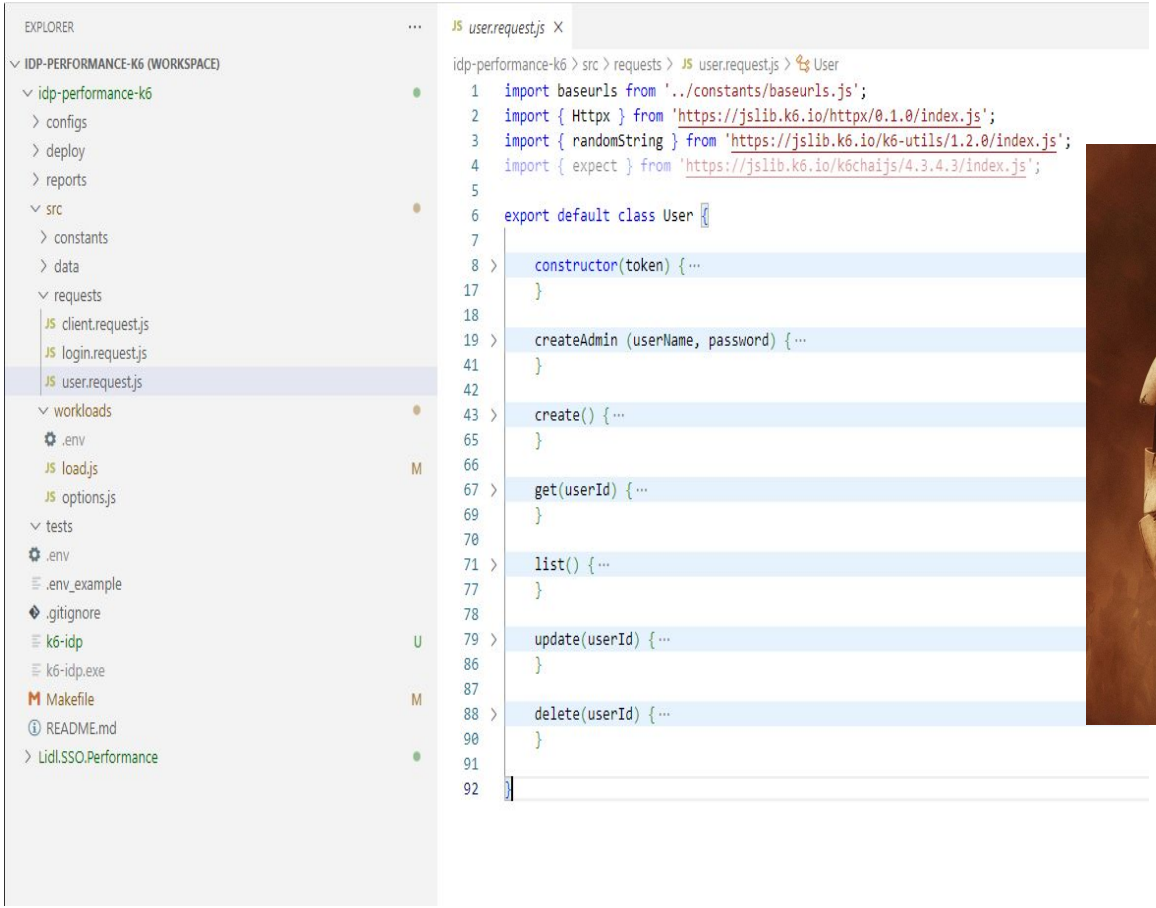
app-full-workflow-get-promotion.js

use-cases > app-full-workflow-get-promotion.js > disrupt

```
1  import { group, sleep } from "k6";
2  import { uuidv4, randomIntBetween, randomItem } from 'https://jslib.k6.io/k6-utils/1.4.0/index.js';
3  import generateJwt from "../shared/jwt-generator.js";
4  import { injectUpeGetPromotionsErrors } from "../mock-server/upe/index.js"
5  import { addGkMockExpectations, injectGkErrors } from "../mock-server/gk/index.js"
6  import getPromotionsHome from "../bff-app/get-promotionshome.js";
7  import getPromotionsList from "../bff-app/get-promotionslist.js";
8  import activatePromotion from "../bff-app/activate-promotion.js";
9  import deactivatePromotion from "../bff-app/deactivate-promotion.js";
10 import { ServiceDisruptor } from "k6/x/disruptor";
11
12 export const options = {
13   noVUConnectionReuse: false,
14   tags : {
15     testid : `coupons-${__ENV.CONFIG_FILE_NAME}_${new Date().toJSON()}`
16   }
17 }
18
19 export function setup() {
20   var environment = __ENV.ENVIRONMENT;
21   var onlineShopRatio = 10; //percentage
22   console.debug
23
24   addGkMockExpectations(environment);
25
26   sleep(4);
27
28   return {
29     environment,
30     onlineShopRatio
31   }
32
33 }
34
35 export default function (data) {
36   const LOG_PREFIX = "[app-full-workflow] ";
```

PROBLEMS TERMINAL DEBUG CONSOLE OUTPUT PORTS

Project Organization in K6



Project Organization in K6

EXPLORER

RECIPES.K6.PERFORMANCETESTS

- configs
  - combined.json
  - filters.json
  - landing.json
  - solobrowser.json
- constants
  - baseurls.js
  - recipesByCountry.js
- data
- deploy
- pages
  - allRecipesPage.js
  - homepage.js
  - landingPage.js
  - recipePage.js
- screenshots
- tests
  - backend.js
  - flows.js
  - landingDisplay.js
  - organic_traffic.js U
  - search.js
- tools
- workloads
  - almutest.js U
  - http.js
  - loadtest.js
- .gitignore
- .nvmrc
- Dockerfile
- Makefile
- README.md
- summary.html

OUTLINE

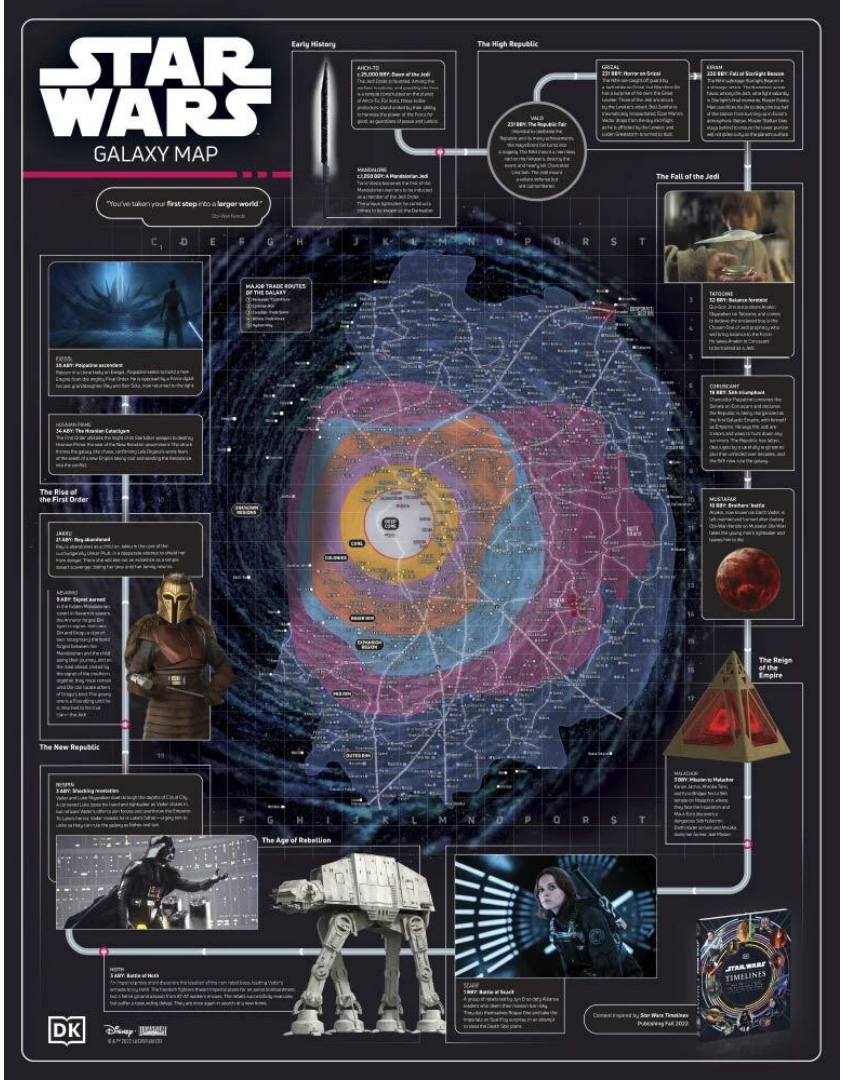
TIMELINE

SOLUTION EXPLORER

loadtest.js

workloads > loadtest.js > recipeDetailsFlow

```
31  const coursesShared = new SharedArray("courses" + __ENV.COUNTRY, function() {
35  });
36
37  export async function recipeDetailsFlow() {
38    const randomNumber = randomIntBetween(0, recipesSlugsShared.length);
39    const page = browser.newPage();
40
41    if (__ITER % 2 === 0) {
42      try {
43        console.info("go to recipeDetails");
44        await recipeDetails(page, recipesSlugsShared[randomNumber]);
45        if (__ITER % 4 === 0) {
46          console.info("go to recipeDetailsTag");
47          await recipeDetailsTag(page, recipesSlugsShared[randomNumber]);
48        } else {
49          console.info("go to allRecipes")
50          await allRecipes(page);
51        }
52      } finally {
53        page.close();
54      }
55    } else {
56      if (__ITER % 8 === 0) {
57        try {
58          console.info("entra en la main page el otro 50%");
59          await homepageMain(page);
60        } finally {
61          page.close();
62        }
63      } else {
64        try {
65          await landings(page, recipesListSlugsShared);
66        } finally {
67          page.close();
68        }
69      }
70    }
71  }
72
73  export function backendFlow() {
74    ssr_backend();
75  }
76
77  export async function landingFlow() {
78    const page = browser.newPage();
79    await landings(page, recipesListSlugsShared);
80  }
81
```



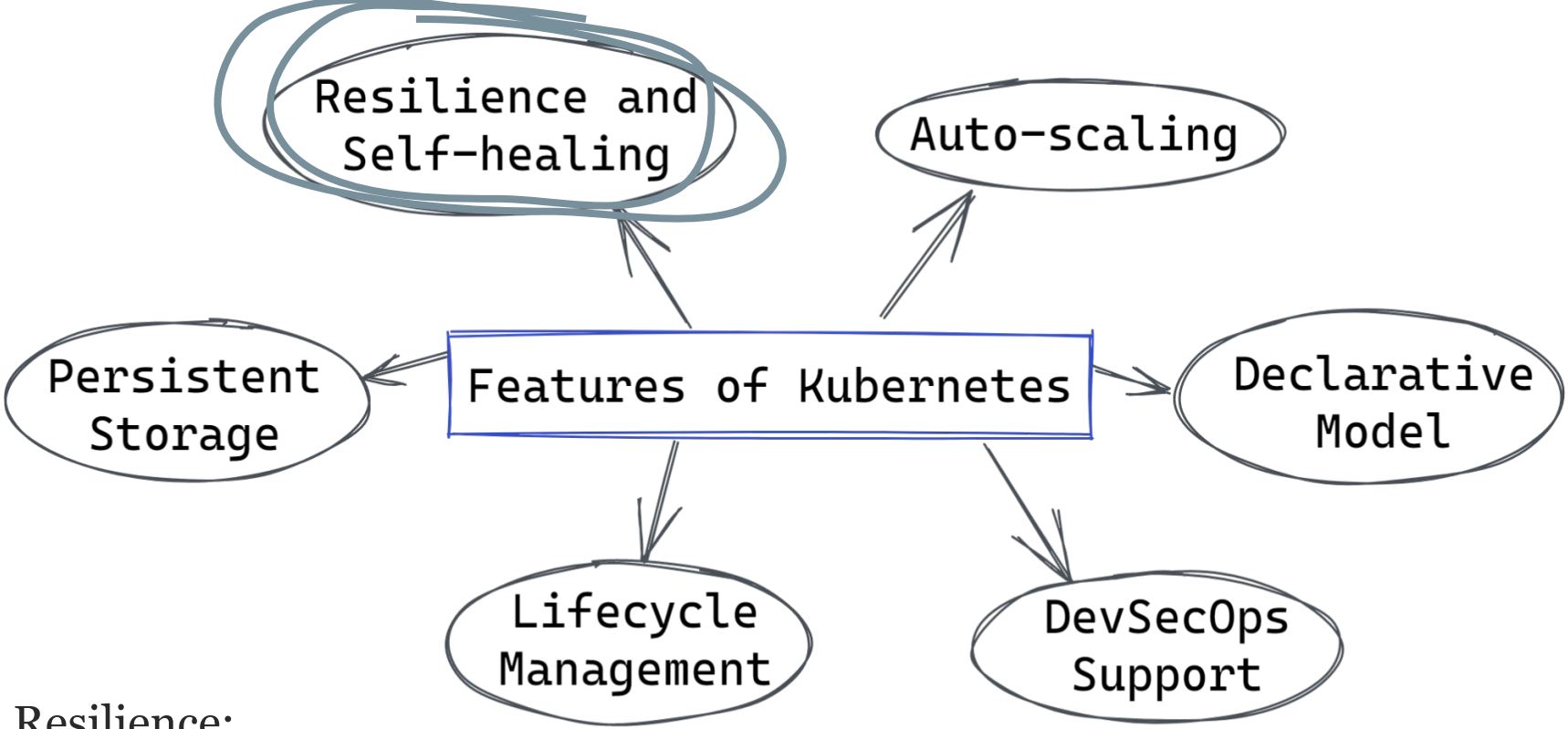
Browser Testing

homepage.js

```
pages > homepage.js > ...
1  import { getEnvironmentUrl } from "../constants/baseurls.js"
2
3  export class HomePage {
4    constructor(page) {
5      this.page = page
6      this.modalWindow = page.locator('button[id="onetrust-accept-btn-handler"]')
7      this.searchField =  page.locator('input[name="q"]')
8      //verifies that the h1 contains the searchToken
9      this.verificationMessage = page.locator('.text-3xl')
10   }
11
12   async goto() {
13     const country = __ENV.COUNTRY;
14       const env = __ENV.ENVIRONMENT;
15       let baseAddress = getEnvi
16       console.debug(`[homePage]
17       await this.page.goto(base
18   }
19
20   async clickCookiesConsent() {
21     //await this.cookiesConsent.c
22   }
23
24   getVerificationMessage() {
25     return this.verificationMessa
26   }
27 }
28
```

configs > {} combined.json > ...

```
1  {
2    "scenarios": {
3      "backend": {
4        "executor": "constant-arrival-rate",
5        "exec": "backendFlow",
6        "rate": 5,
7        "timeUnit": "1s",
8        "duration": "900s",
9        "preAllocatedVUs": 5
10     },
11     "recipeDetail": {
12       "executor": "constant-vus",
13       "exec": "recipeDetailsFlow",
14       "vus": 1,
15       "duration": "900s",
16       "options": {
17         "browser": {
18           "type": "chromium"
19         }
20       }
21     }
22   },
23   "thresholds": {
24       "http_req_duration": ["p(95)<500"],
25       "http_req_duration{scenario: browser}": ["p(95)<1000"],
26       "check_failure_rate": [
27         "rate<0.01",
28         { "threshold": "rate<=0.05", "abortOnFail": true }
29       ]
30     }
31   }
```



Resilience:

1.the capacity to withstand or to recover quickly from difficulties; toughness.

2.the ability of a substance or object to spring back into shape; elasticity.

1reqs-with-faults.json

configs > 1reqs-with-faults.json > ...

```json
{
  "scenarios": {
    "load": {
      "executor": "ramping-arrival-rate",
      "startRate": 20,
      "timeUnit": "1s",
      "preAllocatedVUs": 20,
      "maxVUs": 500,
      "stages": [
        { "target": 1, "duration": "2m" },
        { "target": 1, "duration": "10m" }]
    },
    "disrupt": {
      "executor": "shared-iterations",
      "exec": "disrupt",
      "env": {
        "INJECT_FAULTS": "1"
      },
      "iterations": 1,
      "vus": 1,
      "startTime": "1s"
    }
  }
}
```

Chaos Testing Integrated

k6

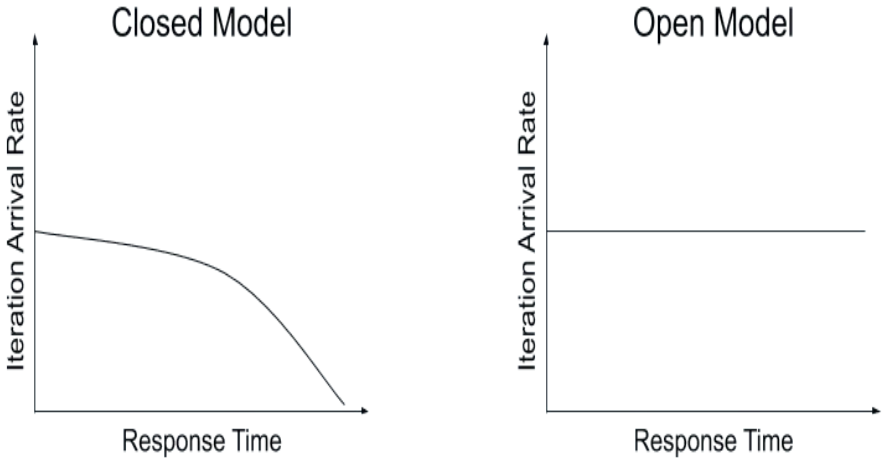
Grafana Labs

Improve application reliability with xk6-disruptor

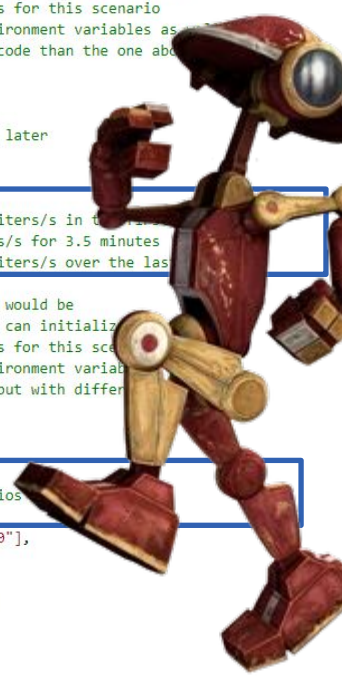
```
export function disrupt(data) {

  console.log("disrupt working");
  const fault = {
    averageDelay: "500ms",
  };

  const svcDisruptor = new ServiceDisruptor(
    "coupons-app-bff-uat-api",
    "coupons-uat"
  );
  svcDisruptor.injectHTTPFaults(fault, "600s", { ProxyPort: 8000 });
}
```

https://grafana.com/docs/k6/latest/testing-guides/injecting-faults-with-xk6-disruptor/

WorkLoad

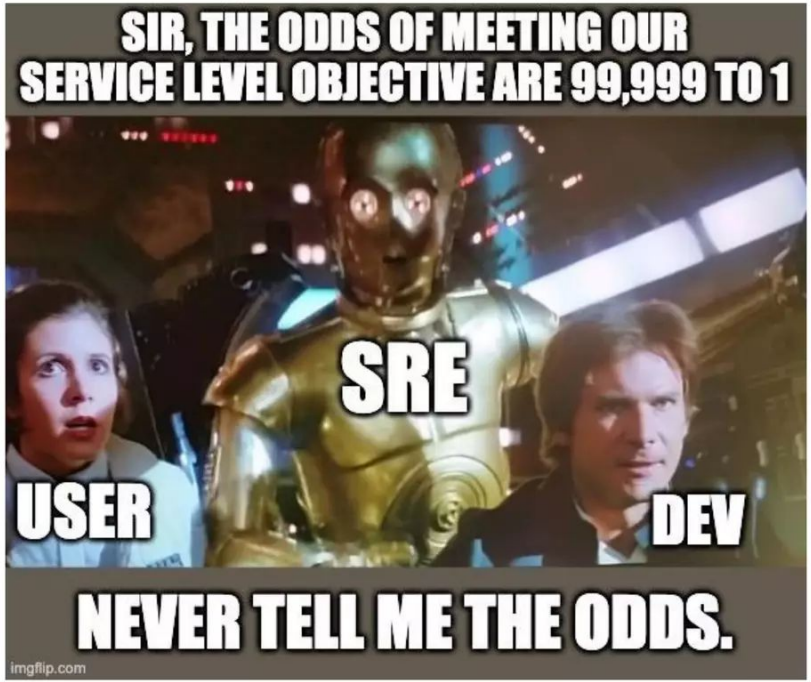



```
export const options = {
  scenarios: {
    my_web_test: {
      // some arbitrary scenario name
      executor: "constant-vus",
      vus: 50,
      duration: "5m",
      gracefulStop: "0s", // do not wait for iterations to finish in the end
      tags: { test_type: "website" }, // extra tags for the metrics generated by this scenario
      exec: "webtest", // the function this scenario will execute
    },
    my_api_test_1: {
      executor: "constant-arrival-rate",
      rate: 90,
      timeUnit: "1m", // 90 iterations per minute, i.e. 1.5 RPS
      duration: "5m",
      preAllocatedVUs: 10, // the size of the VU (i.e. worker) pool for this scenario
      tags: { test_type: "api" }, // different extra metric tags for this scenario
      env: { MY_CROC_ID: "1" }, // and we can specify extra environment variables as
      exec: "apitest", // this scenario is executing different code than the one ab
    },
    my_api_test_2: {
      executor: "ramping-arrival-rate",
      startTime: "30s", // the ramping API test starts a little later
      startRate: 50,
      timeUnit: "1s", // we start at 50 iterations per second
      stages: [
        { target: 200, duration: "30s" }, // go from 50 to 200 iters/s in t
        { target: 200, duration: "3m30s" }, // hold at 200 iters/s for 3.5 minutes
        { target: 0, duration: "30s" }, // ramp down back to 0 iters/s over the las
      ],
      preAllocatedVUs: 50, // how large the initial pool of VUs would be
      maxVUs: 100, // if the preAllocatedVUs are not enough, we can initializ
      tags: { test_type: "api" }, // different extra metric tags for this sc
      env: { MY_CROC_ID: "2" }, // same function, different environment variab
      exec: "apitest", // same function as the scenario above, but with differ
    },
  },
  discardResponseBodies: true,
  thresholds: {
    // we can set different thresholds for the different scenarios
    // of the extra metric tags we set!
    "http_req_duration{test_type:api}": ["p(95)<250", "p(99)<350"],
    "http_req_duration{test_type:website}": ["p(99)<500"],
    // we can reference the scenario names as well
    "http_req_duration{scenario:my_api_test_2}": ["p(99)<300"],
  },
};
```



https://k6.io/docs/using-k6/scenarios/advanced-examples/

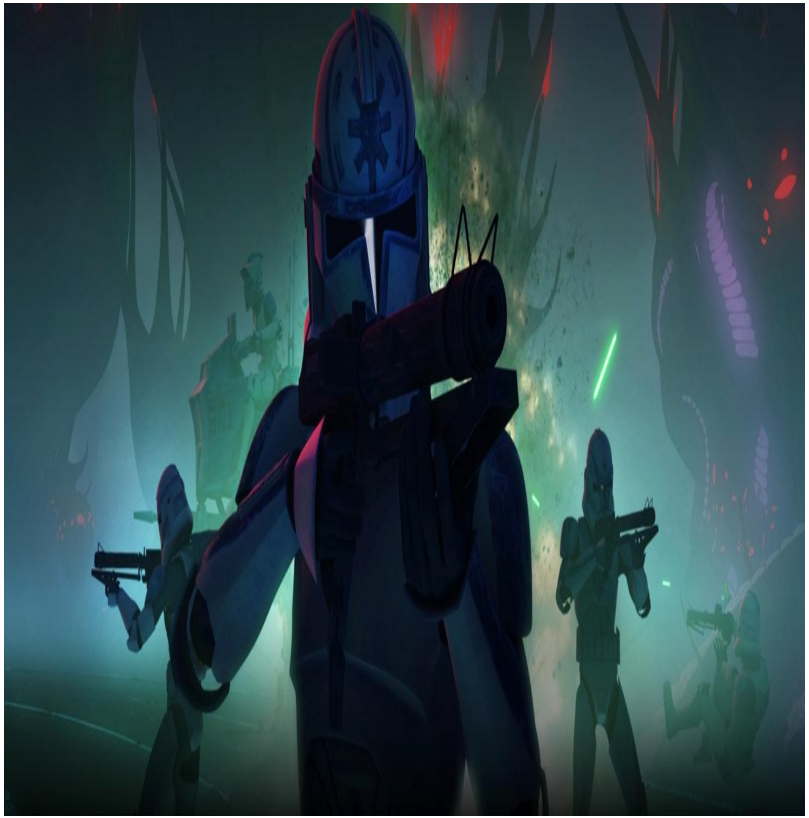
Never tell me the odds!!!



https://twitter.com/nobl9inc/status/1272974728584663049/photo/1

https://www.youtube.com/watch?v=wahlHeyHWKQ&ab_channel=Se%C3%B1orPerformoENG

Thresholds and Metrics



```
import * as utils from "../shared/utils.js";
import * as mock from "../shared/mock.js";
import * as workload from "../shared/scaledobject-workload.js";

import { sleep } from "k6";
import { describe } from "https://jslib.k6.io/k6chaijs/4.3.4.3/index.js";
import exec from "k6/execution";
import { ServiceDisruptor } from "k6/x/disruptor";

const TrendKEDAInternalLatency = utils.generateTrend(
  "keda_internal_latency",
  true,
);

export const options = {…
};

export function setup() {…
}

export default function () {
  workload.setExecutionPrefix(
    utils.generatePrefix(exec.test.options.ext.loadimpact.name),
  );
  var lags = prometheus.getLags(workload.getNamespaceName());
  lags.forEach((lag) => {
    TrendKEDAInternalLatency.add(lag.value, { resource: lag.resource });
  });
  sleep(5);
}

export function disrupt(data) {…
}

export function teardown() {…
}
```

```
"thresholds": {
  "keda_internal_latency": ["p(95) < 7", "p(99) < 15", "max < 20"]
}
}
```

K6 Extensions

JorTurFer / xk6-input-prometheus

Issues Pull requests Actions Security Insights

xk6-input-prometheus Public

Watch 1

main 1 branch 1 tag Go to file Add file Code

JorTurFer add readme ... 369679f on Aug 3 8 commits

.github/workflows	Add client	4 months ago
utils	Add support to QueryRange	4 months ago

xk6-azservicebus Public

Watch 1 Fork 0 Star 1

main 1 branch 1 tag Go to file Add file Code

fernandoescolar fixing readme e272790 3 weeks ago 3 commits

examples	Initial commit	3 weeks ago
.dockerignore	Initial commit	3 weeks ago
.gitignore	Initial commit	3 weeks ago
Dockerfile	Initial commit	3 weeks ago
LICENSE	Initial commit	3 weeks ago

About

Azure ServiceBus Client for xk6

azure-servicebus xk6 xk6-extension

Readme
MIT license
Activity
1 star
1 watching
0 forks

https://k6.io/docs/extensions/get-started/explore/

Running tests is simple …

K6 run test.js

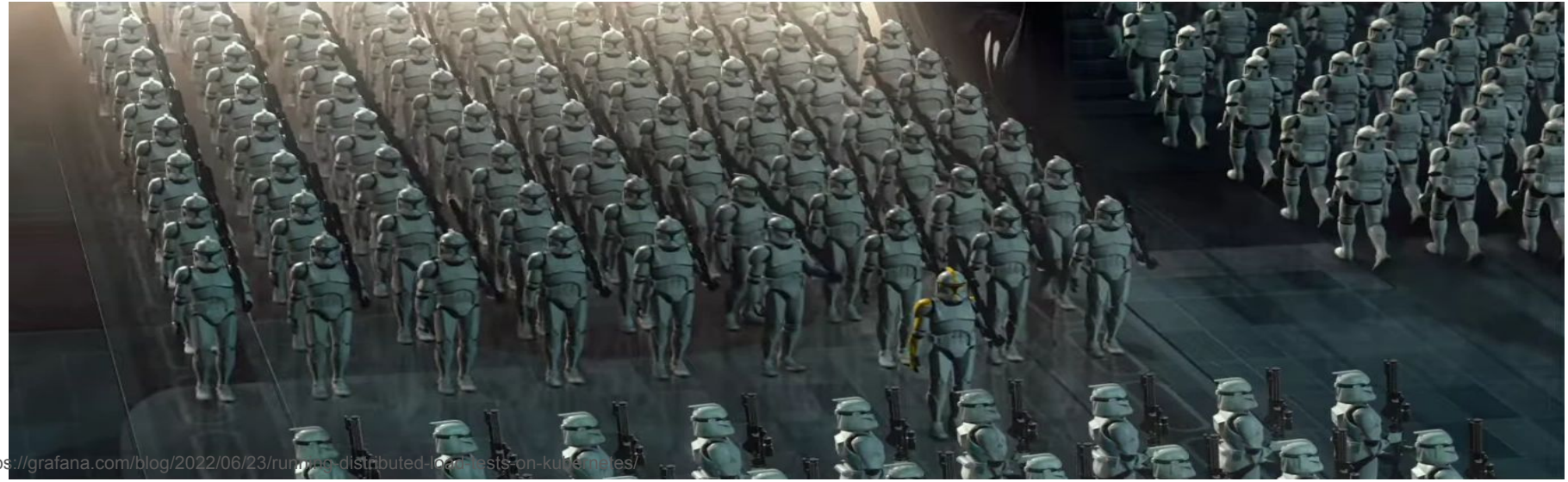
K6 OPERATOR

Why the Operator

Your system under test (SUT) should be accessed from multiple IP addresses.

A fully optimized node cannot produce the load required by your extremely large test.

Kubernetes is already your preferred operations environment.

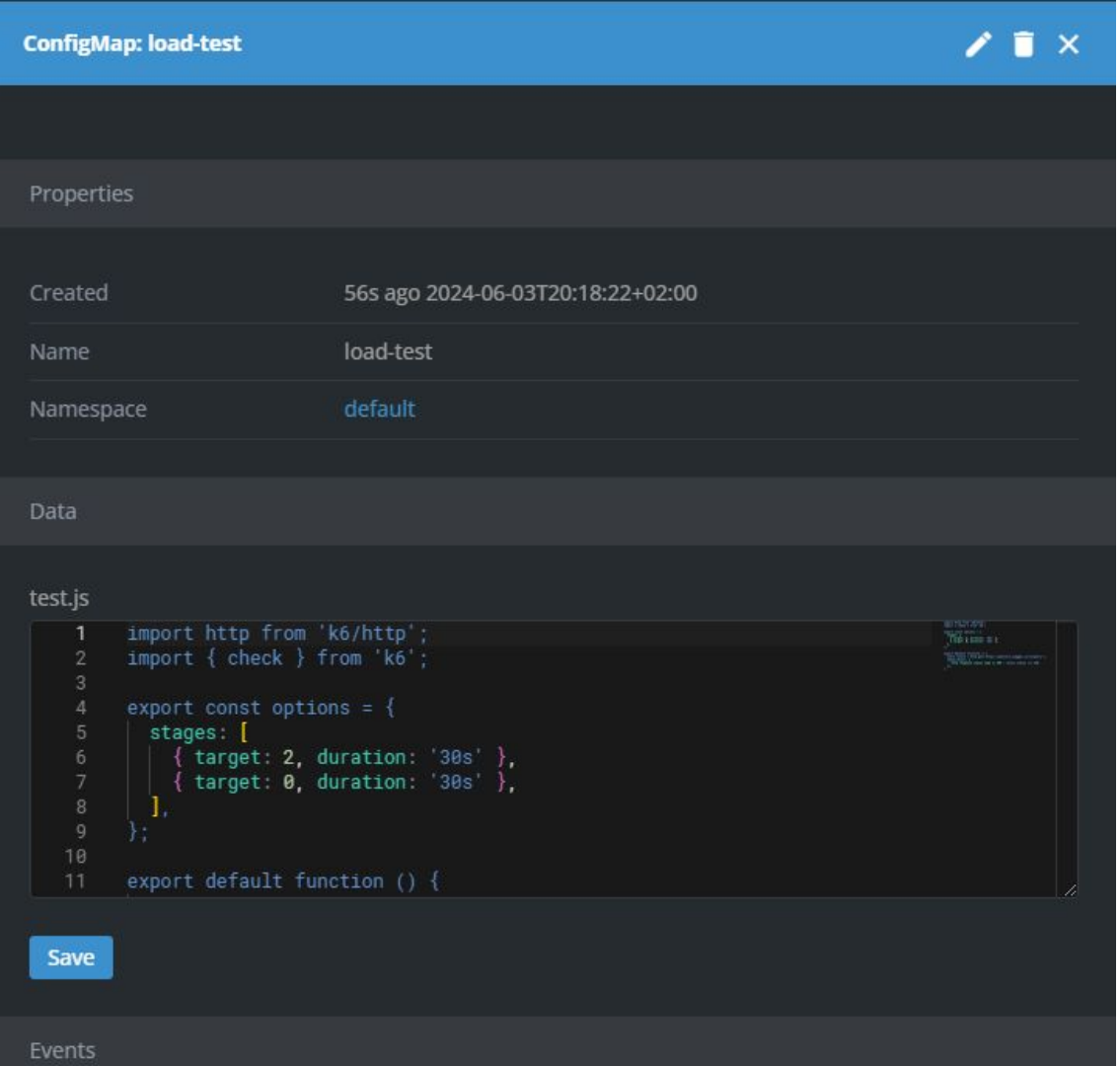


https://grafana.com/blog/2022/06/23/running-distributed-load-tests-on-kubernetes/

Config Map


```
kubectl create configmap my-test --from-file test.js
```

Using a `ConfigMap` is a quick and straightforward mechanism for adding your test scripts to Kubernetes. The `kubectl` tool provides a convenient method to create a new `ConfigMap` from a local script.

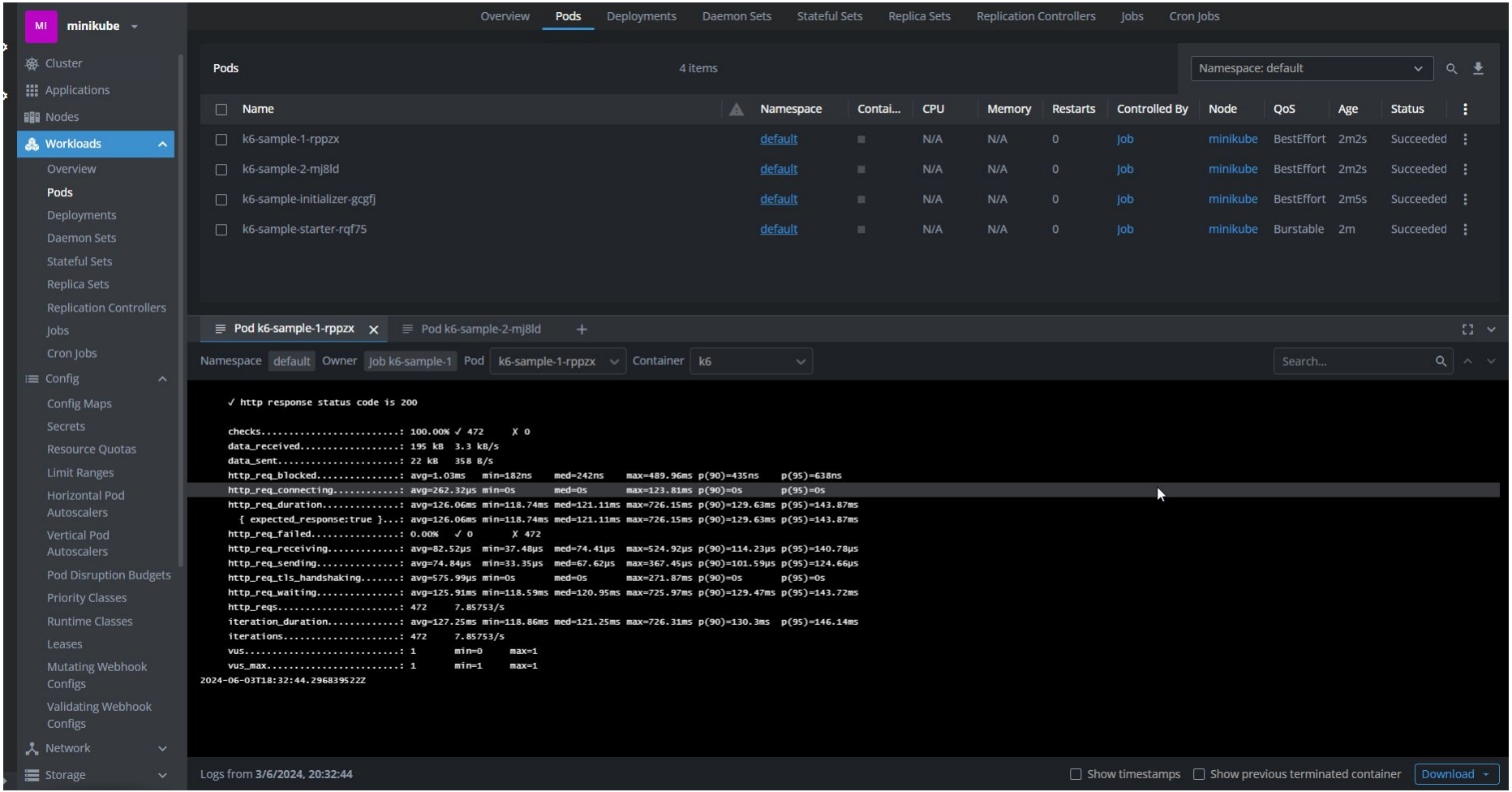


Create a configmap with the test itself

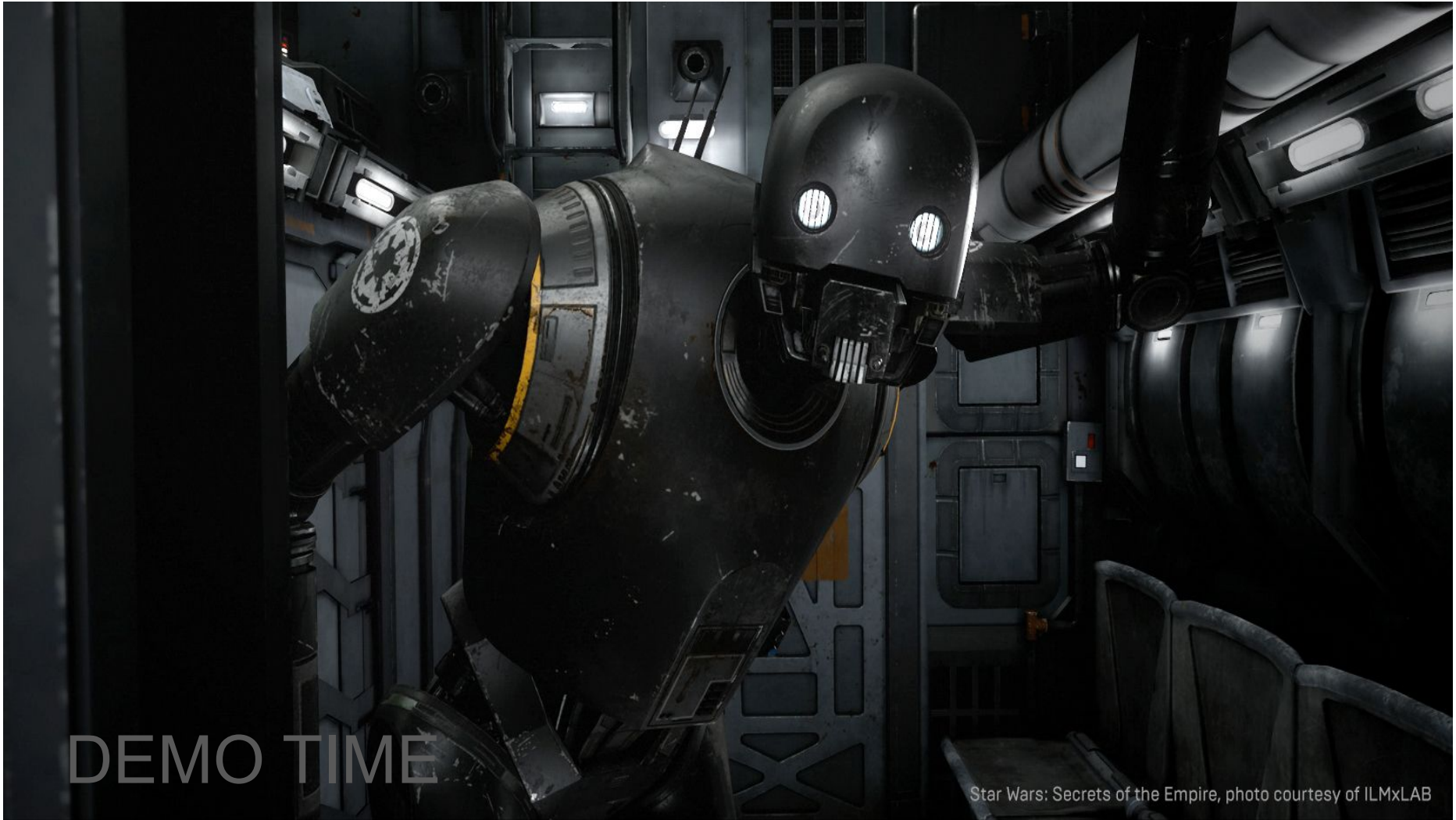
Create a Custom Resource to start the Execution

Running tests is still easy …


```
kubectl create configmap load-test --from-file test/test.js
kubectl apply -f custom-resource.yml
```


MI
minikube
Cluster
Applications
Nodes
Workloads
Overview
Pods
Deployments
Daemon Sets
Stateful Sets
Replica Sets
Replication Controllers
Jobs
Cron Jobs
Config
Config Maps
Secrets
Resource Quotas
Limit Ranges
Horizontal Pod Autoscalers
Vertical Pod Autoscalers
Pod Disruption Budgets
Priority Classes
Runtime Classes
Leases
Mutating Webhook Configs
Validating Webhook Configs
Network
Storage
Overview
Pods
Deployments
Daemon Sets
Stateful Sets
Replica Sets
Replication Controllers
Jobs
Cron Jobs
Pods
4 items
Namespace: default
Name
Namespace
Contai...
CPU
Memory
Restarts
Controlled By
Node
QoS
Age
Status
k6-sample-1-rppzx
default
N/A
N/A
0
Job
minikube
BestEffort
2m2s
Succeeded
k6-sample-2-mj8ld
default
N/A
N/A
0
Job
minikube
BestEffort
2m2s
Succeeded
k6-sample-initializer-gcgfj
default
N/A
N/A
0
Job
minikube
BestEffort
2m5s
Succeeded
k6-sample-starter-rqf75
default
N/A
N/A
0
Job
minikube
Burstable
2m
Succeeded
Pod k6-sample-1-rppzx
Pod k6-sample-2-mj8ld
Namespace
default
Owner
Job k6-sample-1
Pod
k6-sample-1-rppzx
Container
k6
Search...
✓ http response status code is 200
checks.....: 100.00% ✓ 472 X 0
data_received.....: 195 kB 3.3 kB/s
data_sent.....: 22 kB 358 B/s
http_req_blocked.....: avg=1.03ms min=182ns med=242ns max=489.96ms p(90)=435ns p(95)=638ns
http_req_connecting.....: avg=262.32µs min=0s med=0s max=123.81ms p(90)=0s p(95)=0s
http_req_duration.....: avg=126.06ms min=118.74ms med=121.11ms max=726.15ms p(90)=129.63ms p(95)=143.87ms
{ expected_response:true }...: avg=126.06ms min=118.74ms med=121.11ms max=726.15ms p(90)=129.63ms p(95)=143.87ms
http_req_failed.....: 0.00% ✓ 0 X 472
http_req_receiving.....: avg=82.52µs min=37.48µs med=74.41µs max=524.92µs p(90)=114.23µs p(95)=140.78µs
http_req_sending.....: avg=74.84µs min=33.35µs med=67.62µs max=367.45µs p(90)=101.59µs p(95)=124.66µs
http_req_tls_handshaking.....: avg=575.99µs min=0s med=0s max=271.87ms p(90)=0s p(95)=0s
http_req_waiting.....: avg=125.91ms min=118.59ms med=120.95ms max=725.97ms p(90)=129.47ms p(95)=143.72ms
http_reqs.....: 472 7.85753/s
iteration_duration.....: avg=127.25ms min=118.86ms med=121.25ms max=726.31ms p(90)=130.3ms p(95)=146.14ms
iterations.....: 472 7.85753/s
vus.....: 1 min=0 max=1
vus_max.....: 1 min=1 max=1
2024-06-03T18:32:44.296839522Z
Logs from 3/6/2024, 20:32:44
Show timestamps
Show previous terminated container
Download

K6 Grafana Dashboards



Star Wars: Secrets of the Empire, photo courtesy of ILMxLAB


Lidl
Thank you!
for shopping
with us!
Keep shopping on
Lidl.com
need assistance?
(+34) 962 58 78 03
Lidl
Thank you!
for shopping
with us!
Keep shopping on
Lidl.com
need assistance?
(+34) 962 58 78 03