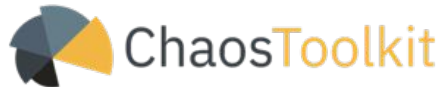


Chaos Engineering

From One to Many

Sylvain Hellegouarch

ChaosIQ CTO and Chaos Toolkit Lead Developer



Demo!

This talk is demo driven so, if you are reading these slides, I encourage you to visit:

<https://github.com/chaostoolkit-incubator/community-playground/tree/master/from-local-to-automated-chaostoolkit>

or

<http://tiny.cc/kubecon-chaos>

What is the context?

One service users can query over HTTP to fetch a simple increment counter.

Counter is managed by the service itself.

Decision is made to delegate the counter to another service the first service will fetch data from. Microservice architecture style.

Our application runs on a Kubernetes cluster.

What do we want to learn?

We want to collect evidence on how impactful a rollout is of a new or older version of the new backend service.

What is the protocol followed?

To do this, we'll devise and run experiments that look at the impact on users, and our SLO, under these conditions:

- Rollout the same version of a backend service (null hypothesis)
- Rollout a new version of the backend service
- Rollout a new version, which is reported unhealthy, of the backend service
- Rollout a new version, healthy but with a bug, of the backend service

Throughout, we'll measure the number of errors reported, via Prometheus, and we'll monitor the lifecycle of a request with OpenTracing.

Chaos Toolkit as a protocol for Chaos Engineering

We rely on the Chaos Toolkit support for the Open Chaos definition of a Chaos Engineering experiment to support our effort in collecting evidence.

<https://chaostoolkit.org/>

<https://openchaos.io/>

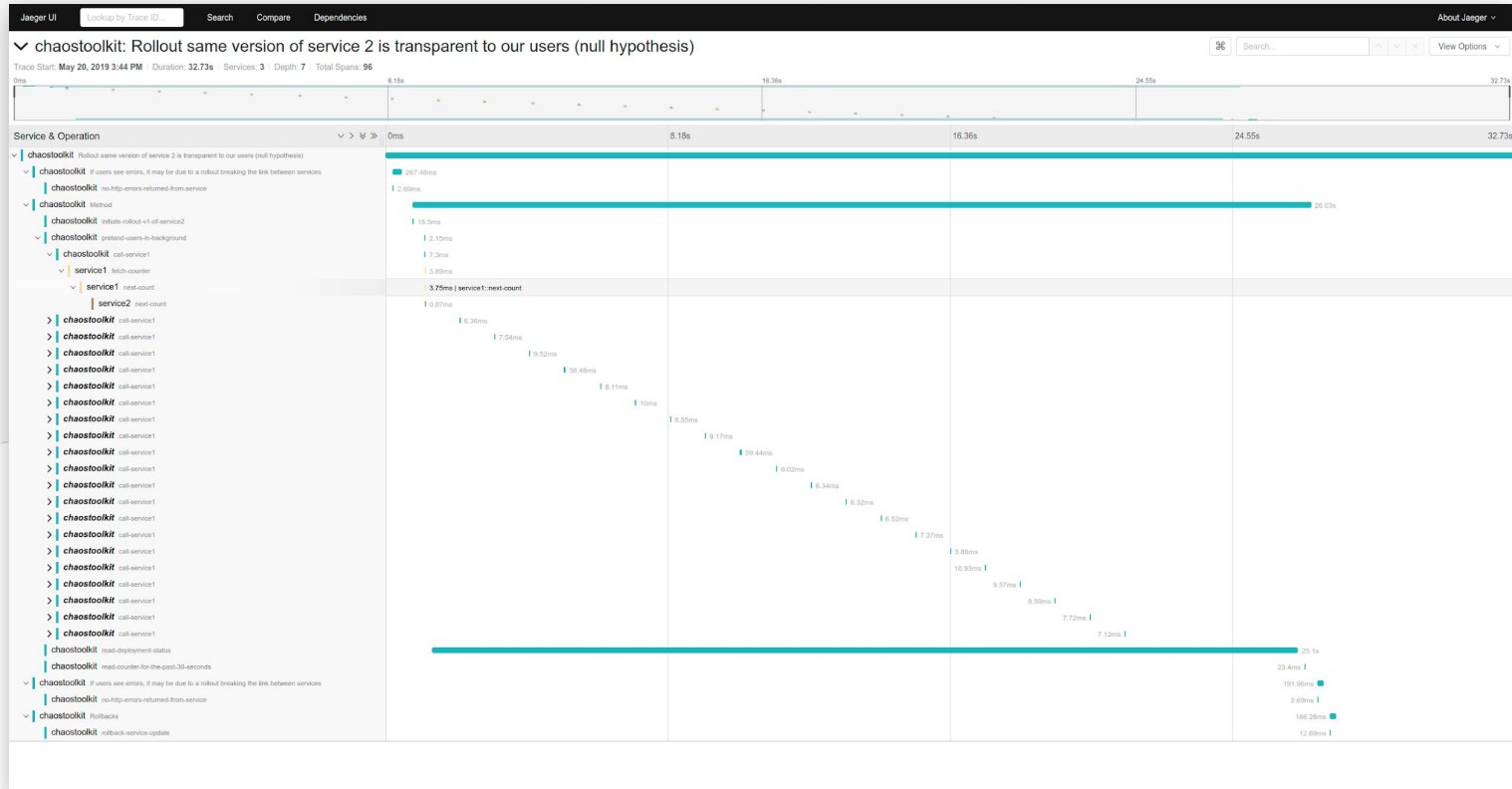
HYPOTHESIS 1

Same version does not impact users

```
(kubecorn) sylvain@sylvain-work:~/dev/demos/kubecorn-eu/experiments$ chaos run --journal-path=v1.json rollout-v1-service2.json
[2019-05-20 15:44:08 INFO] Validating the experiment's syntax
[2019-05-20 15:44:09 INFO] Experiment looks valid
[2019-05-20 15:44:09 INFO] Running experiment: Rollout same version of service 2 is transparent to our users (null hypothesis)
[2019-05-20 15:44:09 INFO] Steady state hypothesis: If users see errors, it may be due to a rollout breaking the link between services
[2019-05-20 15:44:09 INFO] Probe: no-http-errors-returned-from-service
[2019-05-20 15:44:09 INFO] Steady state hypothesis is met!
[2019-05-20 15:44:09 INFO] Action: initiate-rollout-v1-of-service2
[2019-05-20 15:44:10 INFO] Action: pretend-users-in-background
[2019-05-20 15:44:10 INFO] Calling 'http://counter.dev/' for 20s every 1.0s
[2019-05-20 15:44:10 INFO] Probe: read-deployment-status
[2019-05-20 15:44:35 INFO] Probe: read-counter-for-the-past-30-seconds
[2019-05-20 15:44:36 INFO] Steady state hypothesis: If users see errors, it may be due to a rollout breaking the link between services
[2019-05-20 15:44:36 INFO] Probe: no-http-errors-returned-from-service
[2019-05-20 15:44:36 INFO] Steady state hypothesis is met!
[2019-05-20 15:44:36 INFO] Let's rollback..
[2019-05-20 15:44:36 INFO] Rollback: rollback-service-update
[2019-05-20 15:44:36 INFO] Action: rollback-service-update
[2019-05-20 15:44:41 INFO] Experiment ended with status: completed
```

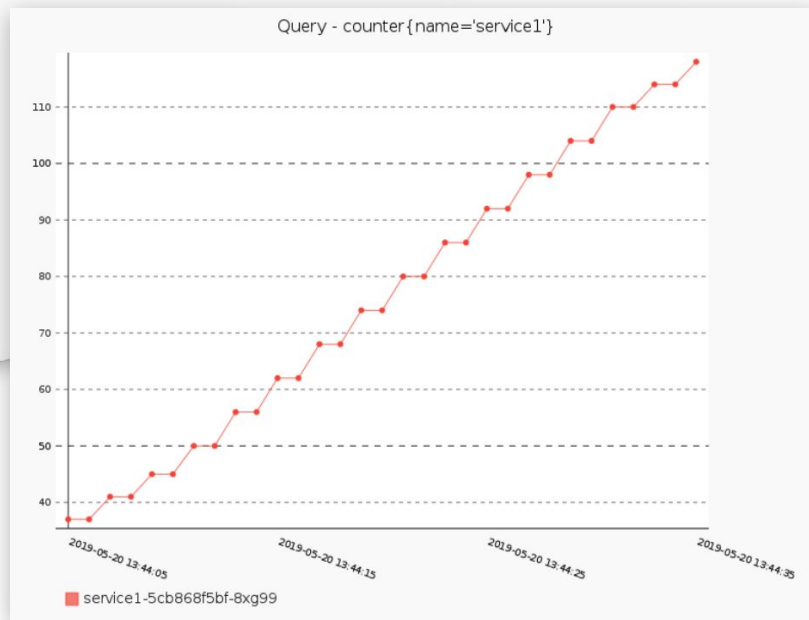
HYPOTHESIS 1

Same version does not impact users



HYPOTHESIS 1

Same version does not impact users



Findings: No impact.

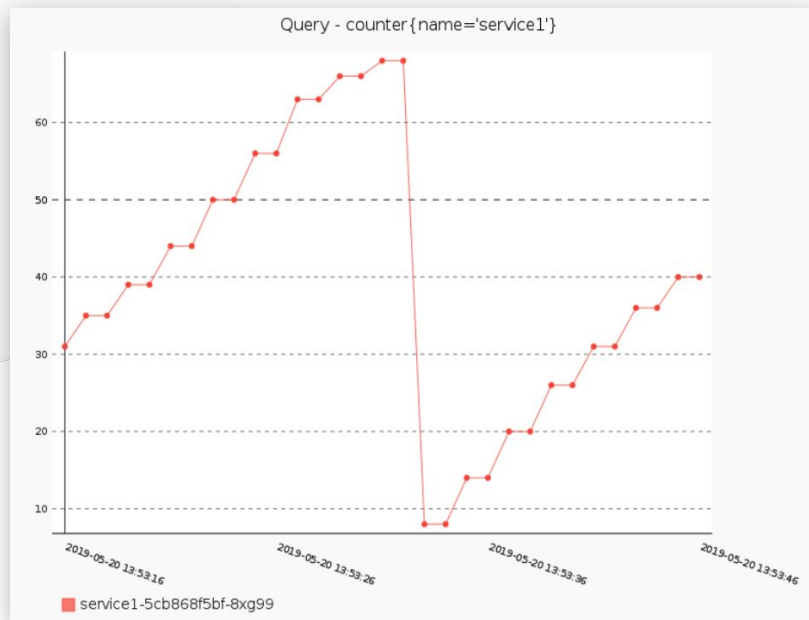
HYPOTHESIS 2

New version does not impact users

```
(kubecon) sylvain@sylvain-work:~/dev/demos/kubecon-eu/experiments$ chaos run --journal-path=v2.json rollout-v2-service2.json
[2019-05-20 15:53:19 INFO] Validating the experiment's syntax
[2019-05-20 15:53:19 INFO] Experiment looks valid
[2019-05-20 15:53:19 INFO] Running experiment: Rollout a new working version of service 2 is transparent to our users
[2019-05-20 15:53:20 INFO] Steady state hypothesis: If users see errors, it may be due to a rollout breaking the link between services
[2019-05-20 15:53:20 INFO] Probe: no-http-errors-returned-from-service
[2019-05-20 15:53:20 INFO] Steady state hypothesis is met!
[2019-05-20 15:53:20 INFO] Action: initiate-rollout-v2-of-service2
[2019-05-20 15:53:20 INFO] Action: pretend-users-in-background
[2019-05-20 15:53:20 INFO] Calling 'http://counter.dev/' for 20s every 1.0s
[2019-05-20 15:53:20 INFO] Probe: read-deployment-status
[2019-05-20 15:53:46 INFO] Probe: read-counter-for-the-past-30-seconds
[2019-05-20 15:53:46 INFO] Steady state hypothesis: If users see errors, it may be due to a rollout breaking the link between services
[2019-05-20 15:53:46 INFO] Probe: no-http-errors-returned-from-service
[2019-05-20 15:53:46 INFO] Steady state hypothesis is met!
[2019-05-20 15:53:46 INFO] Let's rollback...
[2019-05-20 15:53:46 INFO] Rollback: rollback-service-update
[2019-05-20 15:53:46 INFO] Action: rollback-service-update
[2019-05-20 15:53:46 INFO] Experiment ended with status: completed
```


HYPOTHESIS 2

New version does not impact users



Findings: No impact on availability but reset of counter value.

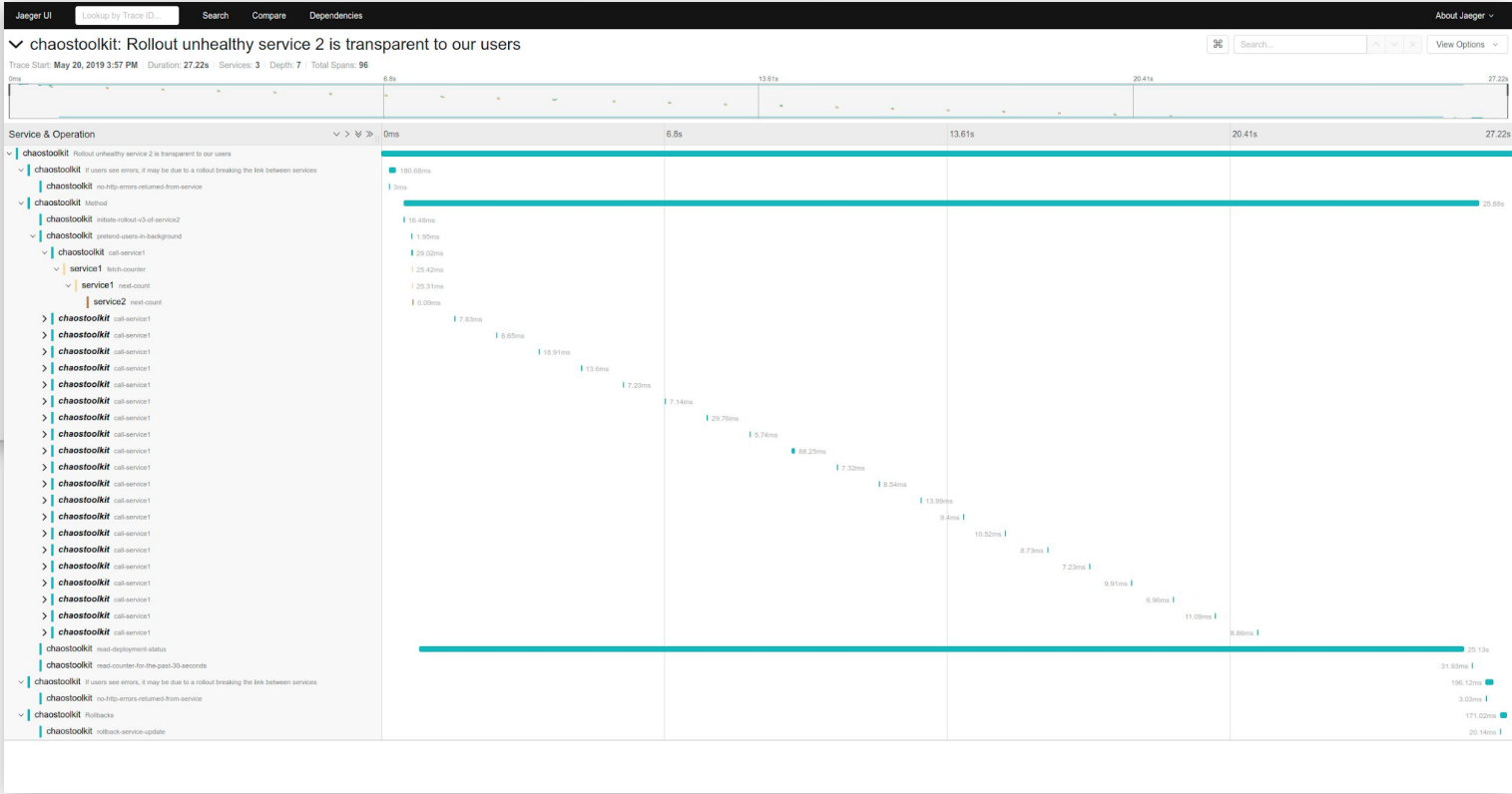
HYPOTHESIS 3

Unhealthy version does not impact users

```
(kubecorn) sylvain@sylvain-work:~/dev/demos/kubecorn-eu/experiments$ chaos run --journal-path=v3.json rollout-v3-service2.json
[2019-05-20 15:57:48 INFO] Validating the experiment's syntax
[2019-05-20 15:57:48 INFO] Experiment looks valid
[2019-05-20 15:57:48 INFO] Running experiment: Rollout unhealthy service 2 is transparent to our users
[2019-05-20 15:57:48 INFO] Steady state hypothesis: If users see errors, it may be due to a rollout breaking the link between services
[2019-05-20 15:57:48 INFO] Probe: no-http-errors-returned-from-service
[2019-05-20 15:57:49 INFO] Steady state hypothesis is met!
[2019-05-20 15:57:49 INFO] Action: initiate-rollout-v3-of-service2
[2019-05-20 15:57:49 INFO] Action: pretend-users-in-background
[2019-05-20 15:57:49 INFO] Calling 'http://counter.dev/' for 20s every 1.0s
[2019-05-20 15:57:49 INFO] Probe: read-deployment-status
[2019-05-20 15:58:15 INFO] Probe: read-counter-for-the-past-30-seconds
[2019-05-20 15:58:15 INFO] Steady state hypothesis: If users see errors, it may be due to a rollout breaking the link between services
[2019-05-20 15:58:15 INFO] Probe: no-http-errors-returned-from-service
[2019-05-20 15:58:15 INFO] Steady state hypothesis is met!
[2019-05-20 15:58:15 INFO] Let's rollback..
[2019-05-20 15:58:15 INFO] Rollback: rollback-service-update
[2019-05-20 15:58:15 INFO] Action: rollback-service-update
[2019-05-20 15:58:16 INFO] Experiment ended with status: completed
```

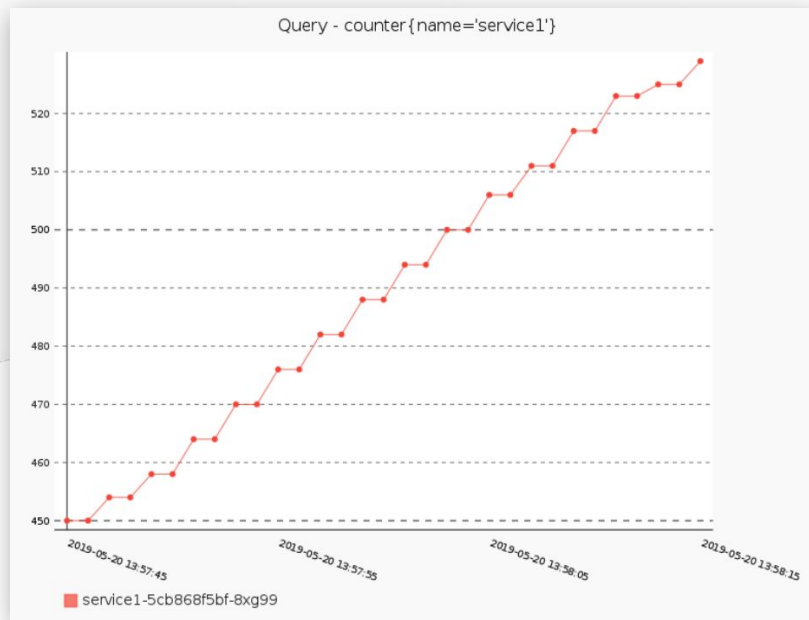
HYPOTHESIS 3

Unhealthy version does not impact users



HYPOTHESIS 3

Unhealthy version does not impact users



Findings: No impact.

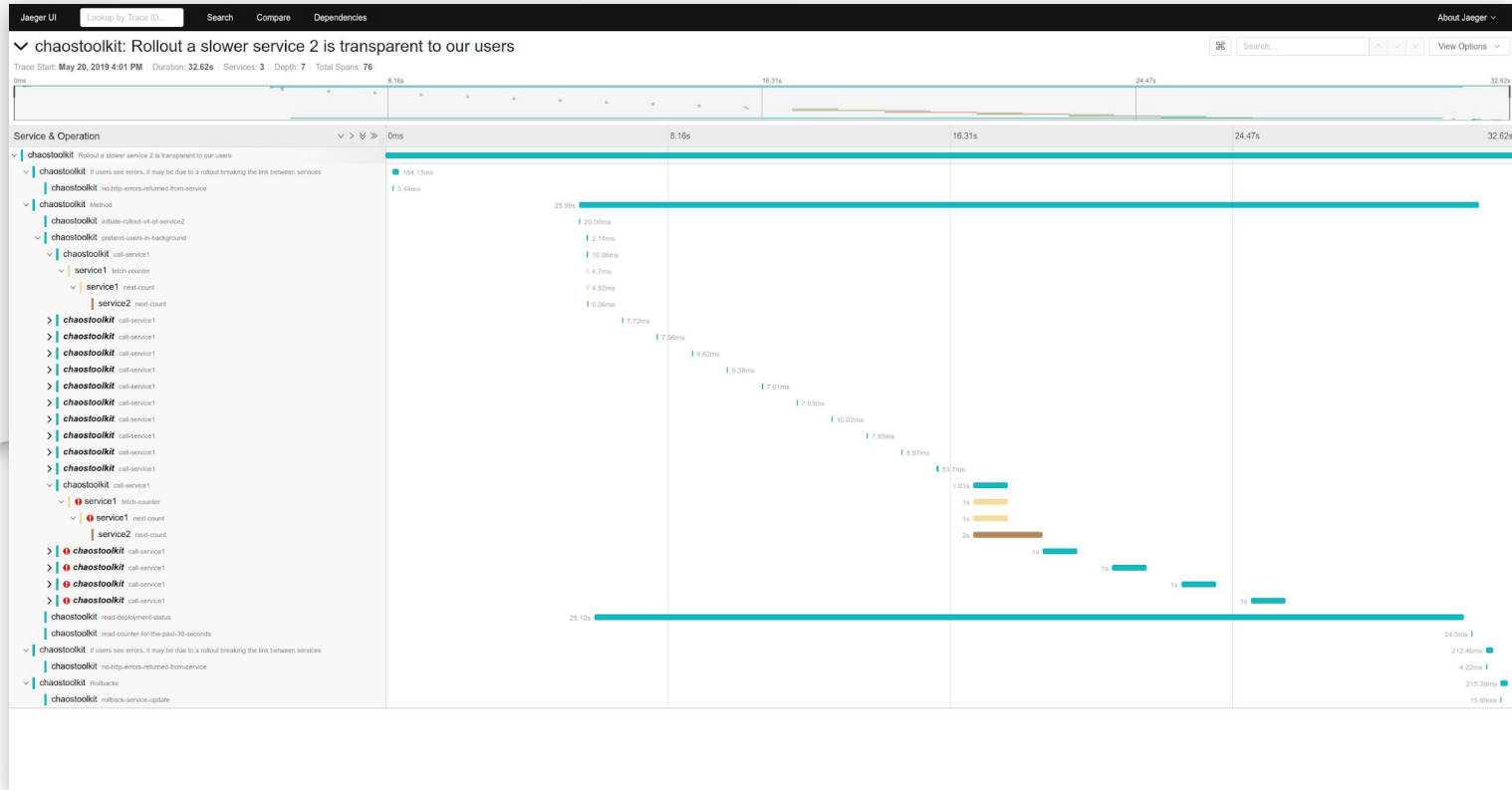
HYPOTHESIS 4

Slower version does not impact users

```
(kubecon) sylvain@sylvain-work:~/dev/demos/kubecon-eu/experiments$ chaos run --journal-path=v4.json rollout-v4-service2.json
[2019-05-20 16:01:05 INFO] Validating the experiment's syntax
[2019-05-20 16:01:05 INFO] Experiment looks valid
[2019-05-20 16:01:05 INFO] Running experiment: Rollout a slower service 2 is transparent to our users
[2019-05-20 16:01:05 INFO] Steady state hypothesis: If users see errors, it may be due to a rollout breaking the link between services
[2019-05-20 16:01:05 INFO] Probe: no-http-errors-returned-from-service
[2019-05-20 16:01:06 INFO] Steady state hypothesis is met!
[2019-05-20 16:01:11 INFO] Action: initiate-rollout-v4-of-service2
[2019-05-20 16:01:11 INFO] Action: pretend-users-in-background
[2019-05-20 16:01:11 INFO] Calling 'http://counter.dev/' for 20s every 1.0s
[2019-05-20 16:01:11 INFO] Probe: read-deployment-status
[2019-05-20 16:01:37 INFO] Probe: read-counter-for-the-past-30-seconds
[2019-05-20 16:01:37 INFO] Steady state hypothesis: If users see errors, it may be due to a rollout breaking the link between services
[2019-05-20 16:01:37 INFO] Probe: no-http-errors-returned-from-service
[2019-05-20 16:01:37 CRITICAL] Steady state probe 'no-http-errors-returned-from-service' is not in the given tolerance so failing this experiment
[2019-05-20 16:01:37 INFO] Let's rollback...
[2019-05-20 16:01:37 INFO] Rollback: rollback-service-update
[2019-05-20 16:01:37 INFO] Action: rollback-service-update
[2019-05-20 16:01:38 INFO] Experiment ended with status: deviated
[2019-05-20 16:01:38 INFO] The steady-state has deviated, a weakness may have been discovered
```

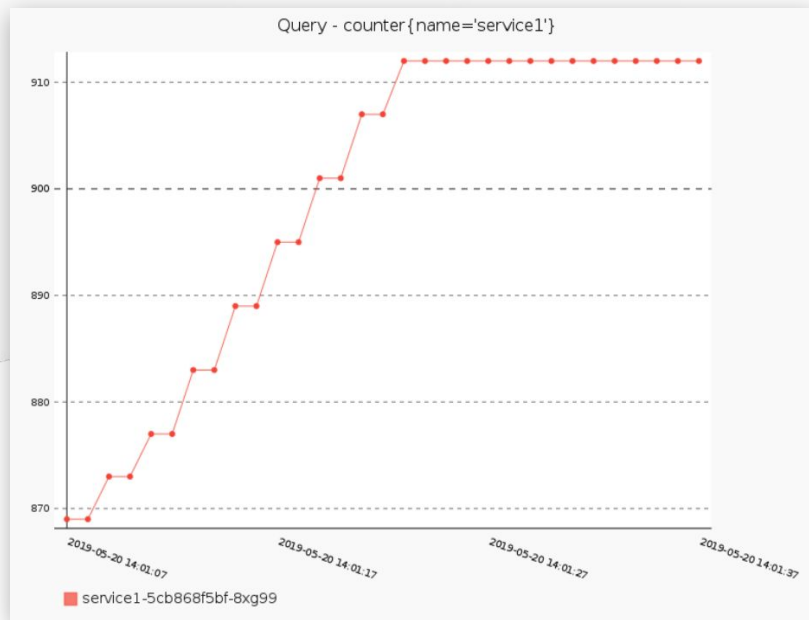

HYPOTHESIS 4

Slower version does not impact users



HYPOTHESIS 4

Slower version does not impact users



Findings: Impacts on availability and thus returned counter value too.

Findings

What we've learnt is that Kubernetes can certainly help us preventing downtime or impact on users when our service declare an appropriate health reporting. Indeed, in that case, the rollout will no complete.

Chaos Toolkit as a protocol for Chaos Engineering

Through the experiment protocol, you can automate not only impacting your system but also collecting data that can help you making a sound analysis afterwards.

Chaos Toolkit as a platform for Chaos Engineering

Chaos Toolkit is great tool to start on your own but can scale up in your organization by integrating naturally with your platform:

- Observability: sending metrics, logs and traces
- Automation: run as part of your CI/CD or as a Kubernetes Job...
- Report nicely for team analysis

Thank you!

@lawouach

