



Chaos Monkey for Spring Boot

Manuel Wessner
John Fletcher

Live Demo

<https://github.com/fletchgqc/mediator>



Project History

Get Involved



Contributing to Chaos Monkey for Spring Boot

This document should help you get started if you want to contribute or hack on the code.
This is your first contribution to an open source project? You can learn how to fork a project and how to create your first Pull Request from the excellent [First Contributions](#) initiative.

If you are unsure where to start, please head over to our [Issue Tracker](#) where we have marked good beginner friendly issues.

Using GitHub Issues

We are using GitHub issues to track bugs, improvements and new features. If you have a general question on how to use Chaos Monkey for Spring Boot, please ask on Stack Overflow using the tag [#spring-boot--chaos-monkey](#).

Reporting Bugs

Please help us to speed up problem diagnosis by providing as much information as possible. A small sample project that reproduces the error would be optimal for us.

Requests for Improvements & New Features

New Pull Requests are highly welcome. However, before you provide us with a PR, please consider opening an GitHub issue tracker. This gives us the possibility to discuss the changes in details. If you're interested in a new feature, try to provide us with your use case.

Reporting Security Vulnerabilities

If you found a security vulnerability please do not disclosure it publicly. Please do not use our issue tracker for security vulnerabilities. You can use our public PGP key to contact us them responsibly. You can use our public PGP key to contact us if needed.

Project Setup

Fork and clone the repository

<https://github.com/codecentric/chaos-monkey-spring-boot>



Recent Changes

Scheduling Runtime Assaults Via Cron

```
POST /actuator/chaosmonkey/assaults
{
  "runtimeAssaultCronExpression": "0 0 * * * *",
  "killApplicationActive": "true"
}
```

Will kill our application every hour.

Memory Assault

`memoryActive: true`

`memoryMillisecondsHoldFilledMemory: 40000`

`memoryMillisecondsWaitNextIncrease: 1000`

`memoryFillIncrementFraction: 0.05`

`memoryFillTargetFraction: 0.95`

Will fill 5% of memory every second until 95% and then hold for 40 seconds.



Roadmap /
Upcoming

Roadmap / Upcoming

- Assaults on outgoing HTTP calls
 - RestTemplate
 - WebClient
- Assess use with reactive applications

When CSMB just
doesn't cut it...





Alternative Tools

Traffic Control (TC)

```
tc qdisc add dev eth0 root netem delay 100ms 20ms
```

Stress CPU

```
stress --cpu 2 --io 1 --vm 1 --vm-bytes 128M --timeout 10s --verbose
```

Pumba

- kill kill specified containers
- netem emulate the properties of wide area networks
- pause pause all processes
- stop stop containers
- rm remove containers

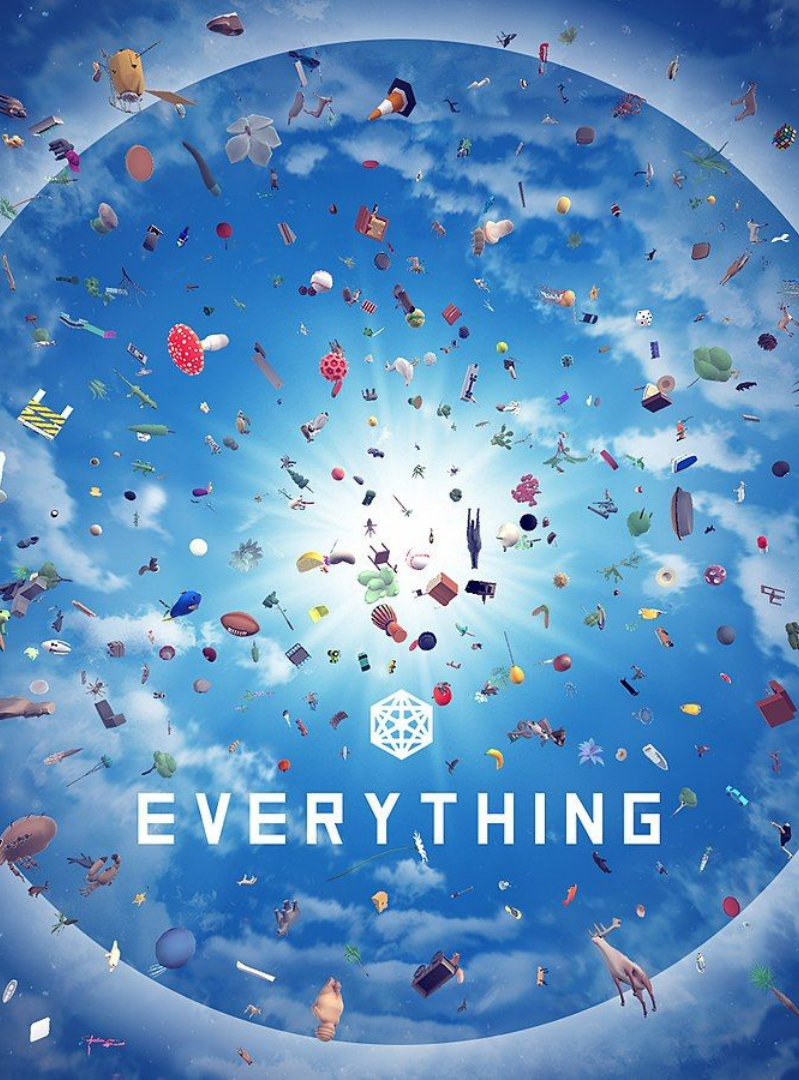


Pumba

```
pumba netem --duration 60s delay --time 1000 mediator_1
```



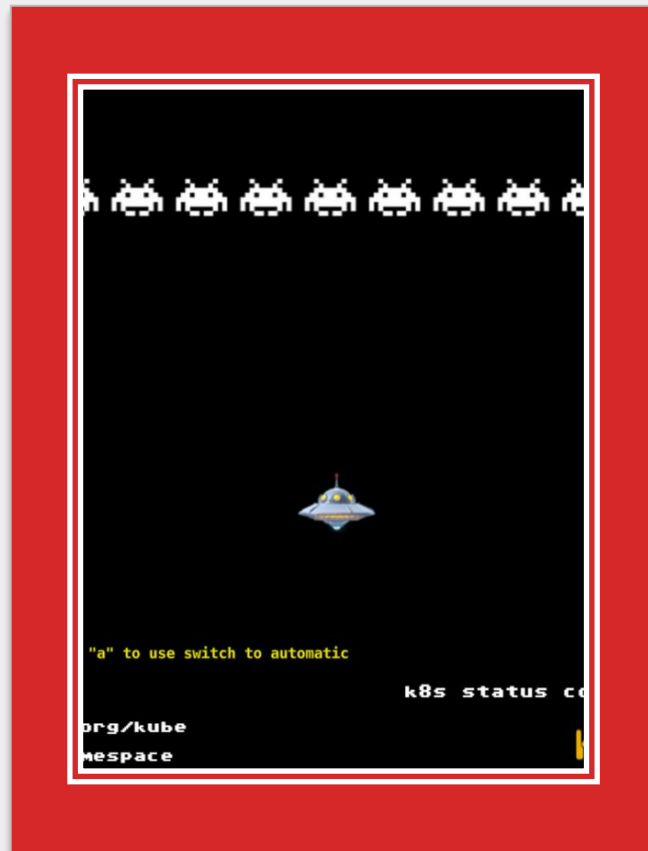
PUMBA

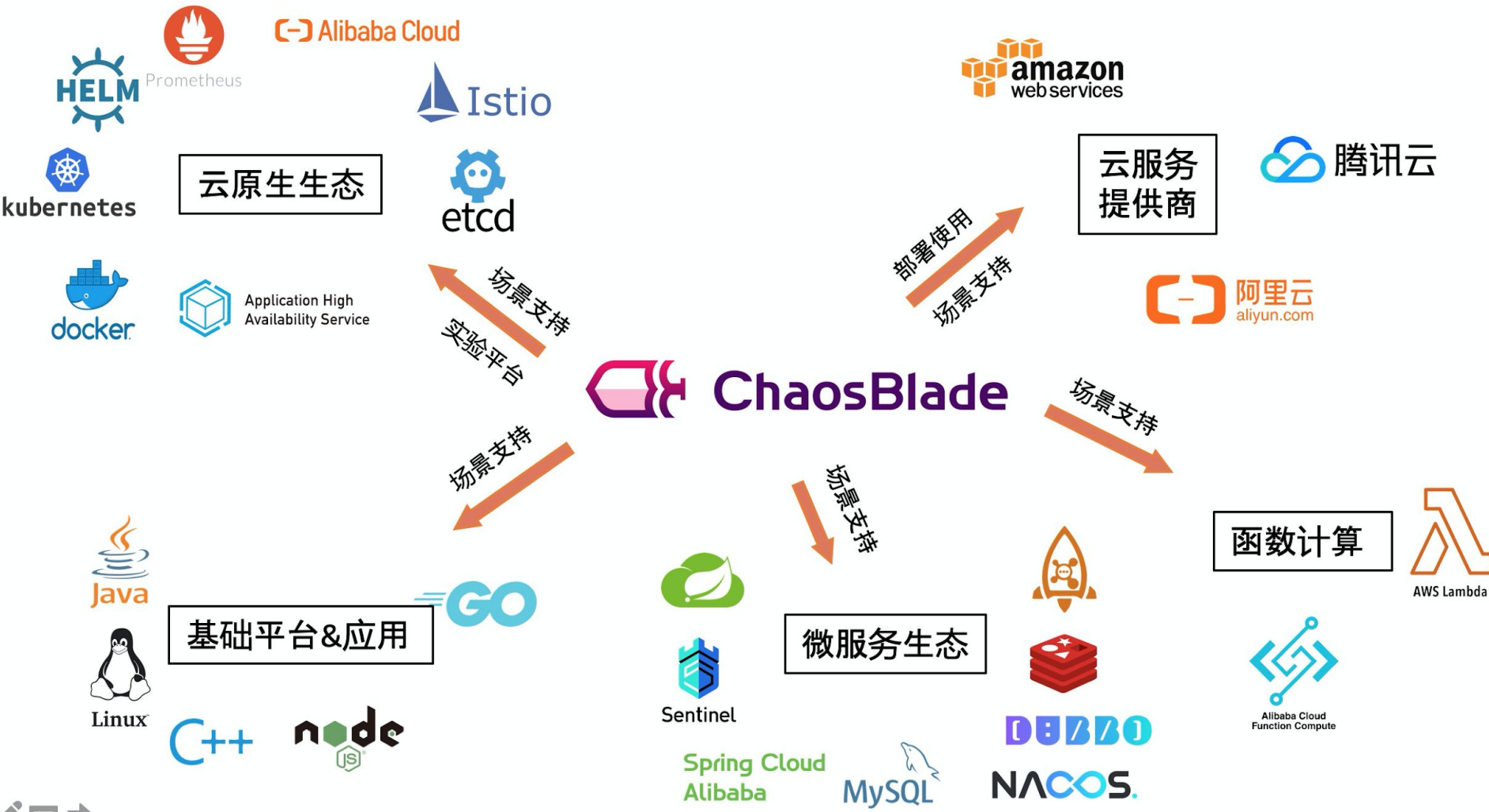


And everything
else...

KubeInvaders

- Space Invaders Clone with Kubernetes Pods







Chaos as a Service

Less system failure. Faster recovery.

The Enterprise-grade Chaos Engineering Platform

Getting started with Chaos Engineering can sometimes be intimidating. ChaosIQ offer the **tools, training and support** that will help you and your team bootstrap, accelerate and scale your journey towards building more resilient systems.

[DISCOVER](#)[START A FREE TRIAL](#)



chaosmesh

Attack

Network Resource State

Container



Stress CPU



Stress IO



Stress Memory

Host



Stress CPU



Stress IO



Stress Memory

Generate CPU load for one or more cores

CPU Cores

How many CPU cores should be consumed?

CPU Load

How much CPU should be consumed?

Activity

Stop all

Now Running



S1-1 Pause Containter fa... 00:19:54



S1-2 Kill k8s 50% of pods... 01:32:31

Next Up

S1-3 Inject latency on all pods 11:00 AM



John Fletcher

Humble Developer

codecentric AG

john.fletcher@codecentric.de

@fletchgqc



Manuel Wessner

Consultant Kind of Dude

codecentric AG

manuel.wessner@codecentric.de

<https://github.com/WtfJoke/>