PERFORMANCE TESTING AUTOMATION FOR CI/CD PIPELINES.

Henrik Rexed
Partner Solution Evangelist
Neotys
Who is this speaker?

- Henrik Rexed (khnightloader)*
- 14 years of Performance engineering

- Technical Evangelist @Neotys
- Lego Fan
- Focus on technologies for Performance engineers
Agenda

01 The current challenge with Performance Testing

02 Make sure your tests are relevant

03 Let’s test earlier

04 What are the SLI I should focus on?

05 Demo
We waited 30 min NO SERVICE
Change of Methodology

- Big Bang
  - Waterfall
- Incremental
  - Rational Unified Process
- Agile
  - Scrum, Kanban
- Continuous
  - DevOps

Requirements → Design → Implementation → Testing → Acceptance → Deployment

Inception → Elaboration → Construction → Transition

Development → Operations

DevOps

Digital Singularity

Agile Event Horizon

time to react on change
Performance testing is like going to the Gym

- Over 1 in 5 Americans belong to at least one U.S. health club or studio
- 12% gym members sign up in January
- Most of the people quit or stop going after 24 weeks
- 50% is going to find the soulmate
Ok but let’s review our architecture first

- From Monolithic

  Presentation Layer

  Business Layer

  Data Layer

  One application per host
  Having the entire business logic
Ok but let's review our architecture first

- To New Microservices

Every service is managed by a dedicated team
Let’s move to the cloud

- Reduce the cost
- Increase the availability of the application
- Simplify, modernize

Yes, but...

SINCE WE ARE IN THE CLOUD...

OUR ARCHITECTURE SEEMS BIGGER...
How can we resolve all those challenges?
Agenda

01 The current challenge with Performance Testing
02 Make sure your tests are relevant
03 Let’s test earlier
04 What are the SLI I should focus on?
05 Q/A
Performance testing is like
Efficient and realistic workload model
Let’s reuse monitoring data

Traditional monitoring

Tracking system

APM
Let’s extract data from our APM to learn how our application is used

**Advantage**
- URL
- API
- Request /min
- Distribution during time
- User sessions
- User actions
- ....Etc

**Disadvantage**
- Aggregation
- You need APM solution
## Agenda

| 01 | The current challenge with Performance Testing |
| 02 | Make sure your tests are relevant |
| 03 | Let’s test earlier |
| 04 | What are the SLI I should focus on? |
| 05 | Q/A |
Architecture is Based on Components
Continuous Performance Testing

API Testing

Test a Component

Integration Testing

Test a System

Application Testing

Test Real World

Continuous Testing Embedded in CI/CD Pipelines

LTB 2020
Load Testing & Benchmarking
The complexity of the test depends on the project life cycle.

**Pre-Production Environment**
- Load testing using devices and browser to validate the UX
- Fail Over testing
- Cloud Testing

**QA Environment**
- Soak Test
- Limit

**Dev Environment**
- Component testing

**Production**
- RUM
- Synthetic
- Load in production
Agenda

01. The current challenge with Performance Testing
02. Make sure your tests are relevant
03. Let’s test earlier
04. What are the SLI I should focus on?
05. Demo
Process to analyse

- Test Failure
- Error rate
- Throughput + Response Time versus objectives
- Resources check
Quality Gate

Collecting the specification

File defining the indicators and the objectives

Retrieving the data From the datasource

Datasource

Quality Gate

Calculate the score
Not limit the evaluation only on response time

**Production**
- Rendering time
- Number of clicks to achieve an action
- Average visit time
- IO
- Number of Pods utilized

**User Experience**
- On a representative environment:
  - Response Time
  - Rendering time
  - Number of container Pods/utilized
  - The behavior of the network
  - ..etc

**Regression**
Detecting regression by looking at:
- Response time
- Hit/s
- Memory used
- CPU

**Scalability**
If it’s a cloud native architecture:
- Number of containers/Pods utilized
- The memory used
- CPU
- Response time
Agenda

01 The current challenge with Performance Testing

02 Make sure your tests are relevant

03 Let’s test earlier

04 What are the SLI I should focus on?

05 Demo
NeoLoad’s platform

NeoLoad Core
- Controller
- Load Generators

NeoLoad Web
- On-Prem Docker Stack
- RESTful APIs

NeoLoad SaaS
- Managed / Hosted
- Cloud Load Generators

NeoLoad Core
- NeoLoad Web
- NeoLoad SaaS

RESTful APIs

Neoload API®

LTB 2020
Load Testing & Benchmarking
What we prepared for you today

Kubernetes cluster

- Staging
- Production

Docker Registry with pre-built app images

GitHub Org

Keptn Repo for project files

Test Orchestration & Test History

Load generators

OneAgent Operator

dynatrace

SaaS Cluster

https://github.com/keptn-orders/
Demo time
Take aways

- Build realistic test based on real production situation.
- Test earlier by running continuous testing
- Automate the analysis by calculating a score of your release
- Select indicators that will validate:
  - Response time
  - The health of the infrastructure
  - The cost of the environment