Debugging microservices applications on Red Hat OpenShift

Red Hat Summit 2019 - Workshop

Mitch Kelley - Solo.io - Software Engineer
Didier Wojciechowski - Red Hat - Senior Specialist Solution Architect
Madou Coulibaly - Red Hat - Senior Specialist Solution Architect

Tuesday, May 7th, 2019
What You Will Learn

● How to visualize and monitor a network of microservices using Istio/Kiali

● How to perform trace analysis using CNCF Jaeger/OpenTracing in order to discover, analyse and understand issues/problems

● How to execute run-time debugging across multi-microservices using squash
How You Will Learn

1. “You Are Not Alone“ (Introduction)
2. “Bird Box”… Not Today! (Kiali)
3. “Dream Within a Dream“ (Jaeger)
4. “Mr Robot”, Please Help Me! (Squash)
What You Will Use

Developer Workspace 1

Developer Workspace ...

Developer Workspace 75
*SNDF* I just feel so... useless...

IDE TAVERN

Don't say that...

But it's true! I spend my time highlighting syntax errors, but my developer doesn't even care...

Yeah, that sucks...

I even try to make myself useful! Yesterday I autocompleted a closing brace, and do you know what he did?

Don't tell me... he...

He deleted it! And then he typed it out again later!!

What a d**k...
Hands-on-Labs:
bit.ly/summit-workshop-debugging

Collaboration Link:
bit.ly/104C

User: userX
Password: r3dh4t1!
Projects: coolstoreX / infraX
Squash Architecture Overview

Red Hat Summit 2019
Boston, MA
How Squash Works

• User specifies debug intent
  - Namespace, Pod, Container
• Squash finds the pid of the target process
• Squash creates a pod with the corresponding debugger (dlv, gdb, etc) and triggers a debug session on the running process
• Squash forwards the debugger’s port to the user’s local environment
• Squash interface (IDE or CLI) connects to debug port

https://squash.solo.io/overview/
How Squash Secure-mode Works

• Squash runs in the cluster
• User indicates debug intent by writing a CRD in the target namespace
• Squash reads the debug intent and creates the required debugger

https://squash.solo.io/secure_mode/
Squash is part of a family of open-source, extensible resilience engineering tools

**Mesh**

with SuperGloo

Debug any service mesh environment through a common API

**Chaos**

with Glooshot

Coordinate chaos experiments and trigger debug sessions

**Replay**

with Loop

Record, replay, and debug erroneous requests outside production

**Extend**

make a plugin!

Squash is highly extensible for IDE, Debugger, and other tooling support