Exploring Stateful Microservices in the Cloud Native World

Mary Grygleski - Sr. Developer Advocate @ IBM
For Dev<Talks/> Reimagined - Romania 2021
Who is Mary Grygleski?

- Senior Developer Advocate at IBM
  - Java, Open Source, Cloud, DevOps, Enterprise/Distributed System, etc.
  - Projects: Open Liberty, MicroProfile, Jakarta EE
  - Developer Relations Software Engineer (DevRel)

- Over 25 years of software engineering experience that span a wide spectrum from product development to enterprise applications and beyond

- President of Chicago Java Users Group (CJUG)

- Tech Community Co-organizers for several IBM-sponsored Meetup Groups in Chicago

- Active church volunteer and religious education teacher
Stateful Vs Stateless
“Computing” Life used to be simpler
Stateless Computing

- A communication protocol that does not retain any session information
  - State of the data does not get recorded between transactions
- Architecture, design and implementation is simpler
  - Scaling the system is easier
  - Recoverability from system failure is easier
Realistically, we live in a Stateful world.
Stateful Computing

- A communication protocol that would retain all session information
  - State of the data gets recorded at every step across all transactions
- Architecture, design and implementation is complex
  - Scaling the system is difficult
  - Recoverability from system failure involves a lot of efforts
Statefulness in the “cloud-less” days...
Client-Server Systems

- Stateful database systems on the server side
  - Database-styled transactions
  - 2-Phase Commit (2PC)
- Java
  - Enterprise Java Beans – Stateful EJB
  - Servlet – HTTPSession
- Client-side caching of server responses
  - Cookie-based authentication
  - Token-based authentication (JWT)
Stateful Microservices in Cloud Native environments
Cloud Native: Stateless containers?
Cloud Native Computing: An overarching approach

- Essentially an “extension” to Cloud Computing by addressing the true needs of enterprise-level distributed business application systems.
- What are the true needs? Netflix coined the term Cloud Native in the early 2010’s and essentially they wanted to leverage on the cloud to meet their goals for their systems to be:
  - Highly available
  - Scalable
  - Performant
The 12 Factor Application

- A methodology that was drafted by developers at Heroku.
- A set of guidelines or best practices for portable and resilient applications that are well suited to be in cloud environments.
- One of the factors indicates the need for self-contained services which are to be deployed as stateless processes. Microservices architecture (so far) is one that can satisfy such a requirement.
- It does not enforce the tools and libraries that the applications must use, but it provides the solid concepts that the applications must follow.
The 12 Factor Application

• I. Codebase
  • One codebase tracked in revision control, many deploys

• II. Dependencies
  • Explicitly declare and isolate dependencies

• III. Config
  • Store config in the environment

• IV. Backing services
  • Treat backing services as attached resources

• V. Build, release, run
  • Strictly separate build and run stages

• VI. Processes
  • Execute the app as one or more stateless processes

• VII. Port binding
  • Export services via port binding

• VIII. Concurrency
  • Scale out via the process model

• IX. Disposability
  • Maximize robustness with fast startup and graceful shutdown

• X. Dev/prod parity
  • Keep development, staging, and production as similar as possible

• XI. Logs
  • Treat logs as event streams

• XII. Admin processes
  • Run admin/management tasks as one-off processes
HOW to preserve state across session, transaction, and network boundaries?
Techniques / Mechanisms

- Cache
- Database-style transactions
- Cookies
- Sessions
- Tokens
Cloud Native Infrastructure

- Containers
  - Kubernetes
    - StatefulSet
    - PersistentVolume
    - Cookie affinity
Programming Design Pattern

- Saga
  - Transactions spanning across multiple services
  - Concept of compensation - no "rollbacks"
  - 2 ways of coordination sagas:
    - Choreography
    - Orchestration

- Long-Running Action / Saga Interaction Pattern
  - "Compensator" model
  - Example Library implementing LRA:
    - MicroProfile LRA
What about Reactive?
Reactive Systems and the Reactive Manifesto

![Diagram of reactive systems]

- Responsive
- Elastic
- Resilient
- Message-Driven
Code samples
Open Liberty session persistence using Jcache & Hazelcast

https://github.com/maryg-lab/guide-sessions

Interactive Lab version:
https://openliberty.io/guides/sessions.html
A Stateful Open Liberty application in Kubernetes

https://github.com/mgrygles/stateful-kube-demo
Resources and Links
Resources (Code samples, Design Patterns, Open Source Libraries)

- Code samples:
  - Open Liberty session persistence using JCache & Hazelcast: [https://github.com/maryg-lab/guide-sessions](https://github.com/maryg-lab/guide-sessions)
  - A Stateful Open Liberty application in Kubernetes: [https://github.com/mgrygles/stateful-kube-demo](https://github.com/mgrygles/stateful-kube-demo)

  Saga Design Pattern: [https://microservices.io/patterns/data/saga.html](https://microservices.io/patterns/data/saga.html)


- Open Liberty: [https://openliberty.io/](https://openliberty.io/)
  - Free tutorials/Guides: [https://openliberty.io/guides](https://openliberty.io/guides)


- Jakarta EE: [https://jakarta.ee/](https://jakarta.ee/)
Resources (Cloud, Cloud-Native, DevOps)


- Cloud Native Computing Foundation (CNCF): [https://github.com/cncf/](https://github.com/cncf/)
IBM Developer - Live Stream on Twitch

#Java/Cloud-Native Programmer’s diary: Every Wednesday 1:00pm (US-Central/GMT-5)
(*additional weekly and occasional streams by other advocates on many other topic areas!)

https://twitch.tv/IBMDriver
Get hands-on experience with a public cloud!

IBM Cloud Free-Tier sign-up  https://cloud.ibm.com/registration

No credit card is required
Join our Expert TV and Meetups (free training on many topics!)

IBM Expert TV:  https://techtv.bemyapp.com/

IBM Developer on Meetup.com  (includes all IBM Developer meetup groups in cities all over the world!)  https://www.meetup.com/pro/ibmdeveloper/

Highly recommended also:

IBM Developer SF Bay Area:  
https://www.meetup.com/IBM-Developer-SF-Bay-Area-Meetup/

Chicago Java Users Group (CJUG)  https://www.meetup.com/ChicagoJUG/
Help solve the world’s greatest challenge.
Answer the Call.

Put your coding skills to work in taking on the most pressing issue of our time — climate change. Start building your solution for the 2021 Call for Code Global Challenge for the chance to win $200,000 USD and receive comprehensive support to see your solution deployed.

Learn how to get started  See challenge details
Thank You!

Join me on Discord
https://discord.gg/RMU4Juw

@mgrygles
https://linkedin.com/in/mary-grygleski
https://github.com/mgrygles
https://dev.to/mgrygles