

Data Distribution and Exploitation in a Global Microservice Arteract Observatory

Panagiotis Gkikopoulos,
PhD student and research assistant, ZHAW and UNINE, Switzerland

The Microservice Pattern

- An increasingly prevalent approach to cloud-native service engineering.
- Comprises of constructing a larger application from loosely coupled, fine-grained component services.
- Grants advantages in terms of fault-tolerance, scalability and continuous integration.

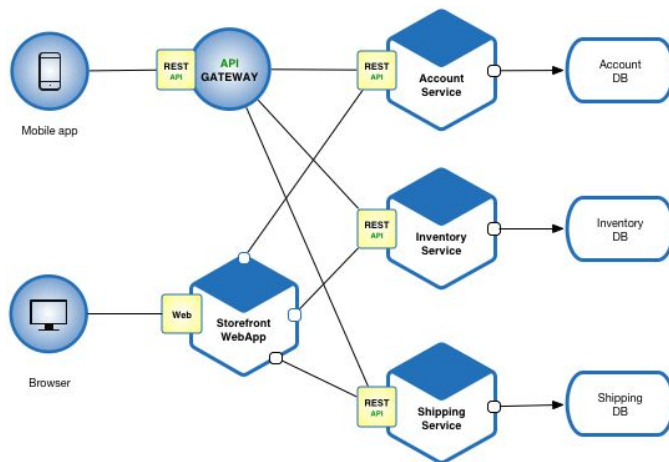


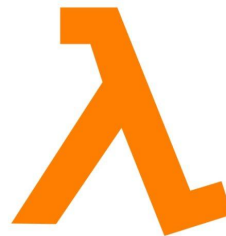
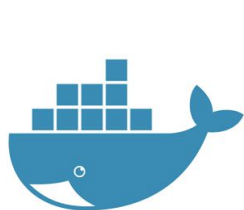
Image source: microservices.io

Microservice Artefacts

Any software artefact retrieved from a public repository and deployed directly, defining a single microservice or set of microservices.

e.g.

docker images, docker-compose files, helm charts, serverless applications, kubernetes operators, CNAB bundles



Microservice Artefact Repositories

Public distribution happens often through marketplaces or hubs, where users can browse artefacts submitted and maintained by various developers and submit their own.

e.g.

Kube Apps Hub, OperatorHub, DockerHub and the AWS Serverless Application Repository

Distributed applications (Dapps) ~ debatable.

Motivation

Rapid expansion of repositories



Wealth of data:

- Development trends
- Quality control issues
- Evolution of microservices



Need for orchestrated effort to collect and analyze the data

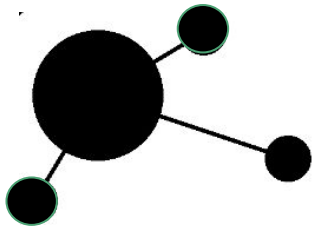
Benefits:

- Historical archive on microservices
- Establishment of better quality standards
- Detection of pattern and anti-patterns

Goals

- 1) Creation of a historical archive for the preservation of data on microservices
- 2) Setting a reference point for microservice development and the creation of quality standards.
- 3) The improvement of microservice artefacts through feedback to developers.
- 4) Understanding the metadata-runtime relationship
- 5) Contribution to a global observatory for microservice artefacts

MAO-MAO



- Stands for Microservice Artefact Observatory - Distributed
- Aims to promote a collaborative approach to the study of microservice artefacts
- Informally predated this doctoral study
- Encompasses early efforts to document and analyze the evolution of the microservice pattern in use

The current operating philosophy is...

...Preservation First

- Research on artefact marketplaces, including metadata analysis, quality control and runtime testing
- Aims to prevent the loss of data through incomplete versioning by collecting as much data as possible

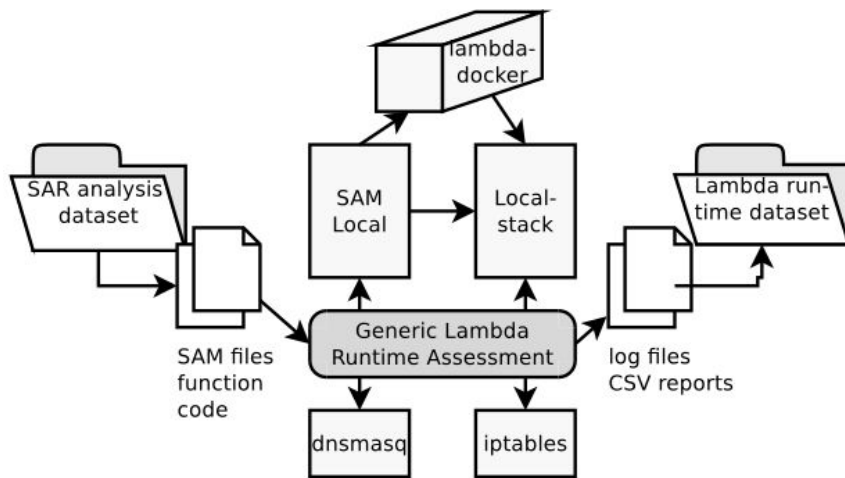
...Infrastructure Later

- Development of a distributed and decentralized research infrastructure, by expanding the network of researchers involved, and providing improved tools to collaborate and share data and experiment setups
- Includes the development of a software platform to automate the distribution and deployment of software testbeds and access to the datasets
- Began later than the collection of data for the informal consortium, but is a primary component of this PhD work

AWS Serverless Application Repository

Expanding the existing work on analyzing the AWS SAR by developing a testbed for generic testing of Serverless Application Model (SAM) applications.

The implementation of the testbed was based on the AWS **SAM local** testing suite and a **Localstack** private AWS emulation stack for providing the backend.



SAR experiment status

Out of **535 applications** in the SAR, the generic runner parsed and executed 270, invoking **283 Lambda functions**, out of which **109 (38,51%) exited successfully**.

Early insights:

- The difficulty of generic, automated execution, especially where AWS BaaS resources are needed
- Memory usage mismatch, indicating a possible need for even more fine-grained micro-billing strategies.

Avg. Allocated Memory	Avg. Max Memory Used
167.38 MB	23.31 MB

Concurrent Experiments

Research is being carried out on other artefact types.

Each individual project is **separate** but the goal is to **unify** them into a single research framework.

Current projects include:

- **Docker images:** (MSc student)
- **Dapps:** (Early stage)
- **HelmQA:** Providing feedback to the developers of Helm Charts and monitor the improvements and developer engagement

Summary

- Multiple experiments collecting data on microservice artefacts
- Understand the trends, patterns and anti-patterns
- Create and maintain a historical archive
- Experiments are still separate
- Effort is being made to integrate them into a single framework to aid in replication of experiments and availability of the data output

zh
aw

SPLAB



Thank you for your attention!

References:

- [1] Alberto Lluch Lafuente Manuel Mazzara Fabrizio Montesi Ruslan Mustafin Larisa Safina Nicola Dragoni, Saverio Giallorenzo. Microservices: yesterday, today, tomorrow. arXiv:1606.04036v4
- [2] Johannes Thones. Microservices. IEEE Software , 32(1):116, 2015.
- [3] AWS Serverless Application Repository. <https://aws.amazon.com/serverless/serverlessrepo/>. Accessed: 14.02.2019.
- [4] Josef Spillner. Quality Assessment and Improvement of Helm Charts for Kubernetes-Based Cloud Applications. arXiv:1901.00644v1
- [5] Josef Spillner. Quantitative Analysis of Cloud Function Evolution in the AWS Serverless Application Repository. arXiv:1905.04800
- [6] Josef Spillner. AWS-SAR Dataset. <https://github.com/serviceprototypinglab/aws-sar-dataset>. Accessed 13.05.2019.
- [7] Josef Spillner. Duplicate reduction in Helm Charts. <https://osf.io/5gkxq/>. Accessed 13.05.2019.
- [8] MAO-MAO: Microservice Artefact Observatory. <https://mao-mao-research.github.io/>. Accessed: 14.02.2019.
- [9] Localstack. <https://github.com/localstack/localstack>. Accessed 13.05.2019.