

# Designing Scalable and Secure Microservices by Embracing DevOps-as-a-Service Offerings

Demetris Trihinas, Athanasios Tryfonos, Marios D. Dikaiiakos

Department of Computer Science  
University of Cyprus  
{ trihinas, a.tryfonos, mdd }@cs.ucy.ac.cy

## 1 Description

This tutorial introduces the microservices architectural paradigm, which arises from the decomposition of an application into smaller services that run as independent processes for each discrete business capability and interact using light communication mechanisms over platform-agnostic APIs [1]. By adopting the DevOps ideology, software teams are each responsible for different aspects of the end application allowing for the team to understand, alter and develop new code without knowing anything about the internals of other services. Although not a necessity, microservices are packaged and deployed through containerized runtimes for portability across private and public cloud offerings. Containerized applications share the host kernel and because they do not have the overhead of a guest OS, something absolutely required by VMs, their size is significantly smaller which makes them easier to migrate, faster to boot and less memory demanding [2]. This setting is ideal for data manipulation, content streaming and the Internet of Things, where software is compiled and offered as a cloud service from a set of smaller, loosely-coupled and independent features.

However, in this tutorial we also examine the fact that simply re-engineering applications into microservices does not remove the operational complexity and risk of failure. In the cloud era, as applications grow by adding more services, security enforcement and dynamic resource allocation become significant challenges [3]. At scale, these challenges can be addressed with autonomicity. Through automation, microservices are equipped with the ability to continuously control the underlying infrastructure, thus turning into services that can be programmatically harnessed. However, traditional monitoring is deemed ineffective for decomposed and highly dynamic microservices deployed over shared execution environments [4]. On the other hand, finer service granularity means more moving parts and hence an increased complexity of auto-scaling, more points of failure and more possibilities for severe privacy violations [5] [6].

To overcome these challenges and take full advantage of the rapidly evolving microservices landscape, we will introduce the Unicorn DevOps-as-a-Service platform<sup>1</sup>, which provides software teams of all sizes with a powerful toolset to simplify and speed up the development and

---

<sup>1</sup> <http://unicorn-project.eu>

management of scalable and secure applications over multi-cloud environments<sup>2</sup>. In respect to microservices, the tutorial will showcase how Unicorn facilitates microservice adoption by providing through its DevOps offerings both a cloud IDE plugin for application development and container packaging, and a dashboard for runtime management. In turn, going beyond the offerings of existing toolsets, we will also present how Unicorn puts particular emphasis on both security and auto-scaling enabled through design libraries that provide microservices with source code annotations to declare policies and constraints for dynamic resource allocation and continuous risk and vulnerability assessment. Finally, we will present how Unicorn complements its solution with advanced orchestration, network fabric and adaptive monitoring capabilities to support containerized multi-cloud deployments. This will showcase that Unicorn provides software teams with one collaborative and unified environment in the same place where they write code, so that they can easily ship, run and manage cloud-enabled microservices.

## 2 Intended Audience, Tutorial Outline and Learning Outcomes

This tutorial targets academic researchers and software developers and is intended to be accessible to audiences with a programming background and with some exposure to cloud environments, software version control and Unix operating systems.

The length of this tutorial is intended to be half a day and is divided into three parts:

- **Part I:** provides a comprehensive overview of the microservice architectural paradigm, the association of microservices with the DevOps culture principles and the enablement of microservices through containerized execution environments. A brief introduction to the concept of containerization is provided along with an overview of where containers sit on the virtualization stack and how they fit in the cloud computing paradigm. The container runtime of choice is Docker<sup>3</sup>, as the industry current golden standard, and Docker Compose<sup>4</sup> will also be used to ease microservice configuration and showcase service portability.
- **Part II:** lectures the audience on the design of applications following the microservice paradigm and enabled through the Docker container runtime on public cloud offerings (Google Cloud, Openstack). To ease microservice development, popular open-source frameworks (e.g., Spring Boot, nginx, Netflix OSS) for web applications will also be introduced. During this part light is shed on the open challenges that arise when designing and deploying secure microservices at scale that prevent developers from fully embracing continuous delivery in their software development pipeline. This part of the tutorial is also intended as a hand-on experience and welcomes any members of the audience to use their laptop and the provided software artifacts to run the presented examples.
- **Part III:** introduces the audience to the Unicorn DevOps-as-a-Service platform which is developed as a unified solution to overcome the open challenges previously presented. Specifically, we will extend in a few simple steps the exemplary microservices introduced in the previous section with the Unicorn feature set to show developers how they are able to simplify and speed up the design and management of scalable, secure and portable microservices over containerized cloud execution environments.

---

<sup>2</sup> <https://gitlab.com/unicorn-project>

<sup>3</sup> <https://www.docker.com/>

<sup>4</sup> <https://docs.docker.com/compose/>

With the completion of the tutorial, the audience should:

- Have a clear sense of what are microservices and why this paradigm constitutes a revolutionary shift in software development.
- Acknowledge the principles that designate applications as microservices and understand the process flow in designing web applications as microservices with the use of popular frameworks (e.g., Spring Boot, nginx Netflix OSS) from the open-source ecosystem.
- Leverage Docker and its toolset (e.g., dockerfiles, docker hub) to containerize and run web applications as microservices.
- Understand how Docker orchestration tools (e.g., Docker Compose) can be used to automate microservice deployment on public cloud offerings.
- Acknowledge the existence of the Unicorn DevOps-as-a-Service platform offerings to speed up the design of secure and scalable microservices and ease the runtime management of large-scale and distributed deployments.

### 3 Materials

Course slides and short videos. For the audience interested in experiencing the tutorial “hands-on”, all software artifacts will be available for download. In this case, a laptop with a Unix OS and a good internet connection are required. The course slides, videos and software artifacts will remain accessible online after the conference completion.

### 4 Speakers Bio's

**Demetris Trihinas** is a Postdoctoral Researcher at the Department of Computer Science, University of Cyprus. His research interests include Distributed and Internet Computing with particular focus in Big Data Analytics for Cloud and Edge Computing. He received his PhD degree from the University of Cyprus in 2017 with his dissertation targeting developing low-cost probabilistic and adaptive learning models for approximate monitoring in order to improve energy-efficiency and reduce the volume of data generated by IoT services offered through the cloud. He additionally holds a MSc in Computer Science from the University of Cyprus and a Dipl-Ing in Electrical and Computer Engineering from the National Technical University of Athens (NTUA). Dr. Trihinas is also the developer of the JCatascopia cloud monitoring system, the AdaM framework for IoT devices and also an active member of the Cloud Application Management Framework (CAMF) which is an official Eclipse Foundation project. His work is published in IEEE/ACM journals and conferences such as Trans. Services Computing (TSC), Trans. Cloud Computing (TCC), ICDCS, INFOCOM, BigData, CCGrid, and ICSSOC.

**Athanasios Tryfonos** is a DevOps Engineer and Software Developer at the Laboratory of Internet Computing of the Department of Computer Science at University of Cyprus. His interests include Cloud Computing, Virtualization & Containerization Technologies, Microservices and modern DevOps Tools and Practices. He received his BSc from the Department of Computer Science at the University of Cyprus with his thesis focusing on artifact management for the Cloud Application Management Framework (CAMF) which is an official Eclipse Foundation project. He is currently member of the development teams of UNICORN H2020 Project and

ENEDH Interreg project. Athanasios has previously worked as a web application developer and systems administrator as well.

**Marios D. Dikaiakos** is Professor of Computer Science at the University of Cyprus and director of the University's Centre for Entrepreneurship. He served as Head of the Computer Science Department in 2010-2014. Dikaiakos received his Ph.D. in Computer Science from Princeton University (1994), an M.A. in Computer Science from Princeton (1991), and a Dipl.-Ing. degree in Electrical Engineering from the National Technical University of Athens (summa cum laude, 1988). Dikaiakos research focuses on the design, development and evaluation of large-scale distributed computing systems and tools. Prof. Dikaiakos has over 20 years of experience as principal investigator, principal institutional investigator or participant in research projects funded by European and American agencies. His research work includes over 160 original publications in books, journals and international conference proceedings, the editing of books and conference proceedings, and the direction of research projects at the University of Cyprus with a total budget exceeding 4 million Euros for the period 2000-2018. He has also served as program chair, co-chair and program committee member in numerous international scientific conferences; independent evaluator of European academic and research institutions; reviewer and evaluator for research proposals submitted to the E.U. and to several national European science foundations; and independent observer of proposal evaluation exercises of the E.U. under Framework Program 7. He is a Senior Member of the ACM, was a member of ACMs Distinguished Speakers Program (2012-2015), a member of the IEEE Computer Society and the Technical Chamber of Greece. More details can be found under: <http://www.cs.ucy.ac.cy/mdd/>.

## References

- [1] J. Thones. Microservices. *IEEE Software*, 32(1):116–116, Jan 2015.
- [2] R. Morabito, V. Cozzolino, A. Y. Ding, N. Bejar, and J. Ott. Consolidate iot edge computing with lightweight virtualization. *IEEE Network*, 32(1):102–111, Jan 2018.
- [3] D. Trihinas, A. Tryfonos, M. Dikaiakos, and G. Pallis. Devops as a service: Pushing the boundaries of microservice adoption. *IEEE Internet Computing*, 22(3):65–71, May 2018.
- [4] D. Trihinas, G. Pallis, and M. Dikaiakos. ADMIn: adaptive monitoring dissemination for the internet of things. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications (INFOCOM 2017)*, Atlanta, USA, May 2017.
- [5] Manuel Ramírez López and Josef Spillner. Towards quantifiable boundaries for elastic horizontal scaling of microservices. In *Proceedings of the 10th International Conference on Utility and Cloud Computing, UCC '17*, pages 35–40, New York, NY, USA, 2017. ACM.
- [6] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. *Microservices: Yesterday, Today, and Tomorrow*, pages 195–216. Springer International Publishing, 2017.