



Definition

In this lesson, we'll look at the definition of microservices platforms.

We'll cover the following



- Support for HTTP and REST
- Expenditure for installation and operation
- Migration to a microservices platform
- Influence on the macro architecture
- Specific platforms

The platforms in the next chapters differ from all other technologies presented so far in that they enable the communication of microservices and support aspects of operation such as deployment, monitoring, or log analysis.

Support for HTTP and REST

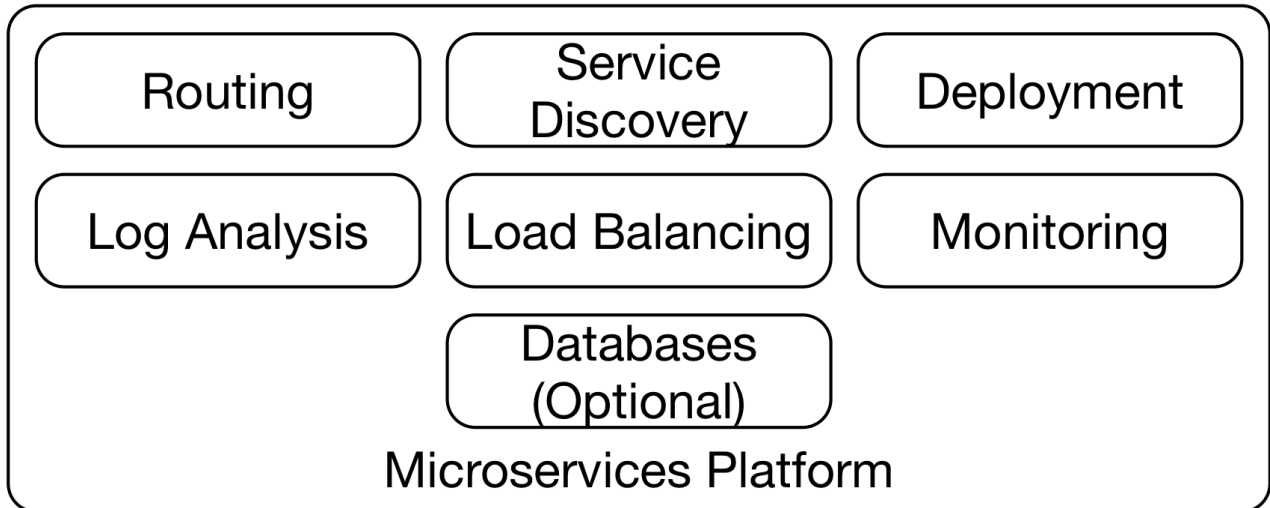
#

The platforms support HTTP and REST with load balancing, routing, and service discovery, but can be supplemented with other communication mechanisms. This makes it possible to use the platforms for setting up asynchronous systems.

However, this chapter considers synchronous communication.

Microservices on platforms that use this type of communication differ most from microservices on different infrastructures concerning their

mode of operation.



Features of Microservices Platforms

Expenditure for installation and operation

Microservices platforms are very powerful, but consequently very complex. They make it easier to work with microservices but installing and operating the platform itself can really be a challenge.

If the platform is running in the public cloud, the complex installation and operation does not play a role, because the operator of the public cloud has to take care of it. However, if an installation is carried out in the company's own data center, the operations team has to make the effort.

However, the effort for installing the platform has to be made only once. After that, deploying the microservices is much easier. This means that the cost of installing the platform will be amortized quickly.

Only limited operations support is necessary when installing the microservices. In this way, microservices can be deployed quickly and easily even if operations cannot support each deployment of each microservice.



Migration to a microservices platform

In contrast to the solutions shown so far, microservices platforms require a **fundamental change in the operation and installation of the applications**.

The other examples can be run on virtual machines or even physical servers and can be combined with existing deployment tools.

Since the microservices platforms also cover the deployment and operation of the microservices, they include features for which operations have already established tools in most cases. Therefore, the use of a microservices platform is **a bigger step than the use of other technologies**. Such a move can be a deterrent for conservative operations teams.

In addition, the introduction of microservices often leads to **organizational changes** and new technologies in addition to a new architecture.

It can be helpful if a complex microservices platform does not have to be introduced in this situation.

Influence on the macro architecture

Of course, a platform only supports certain technologies. In addition to the programming language, these are the technologies for deployment,

monitoring, logging, and so on. The platform therefore defines large parts of the technical infrastructure.



Therefore, microservices platforms have a relationship to the macro architecture.

- The macro architecture should state the **requirements of the platform** ensuring that the platform can run the microservices. The platform specifies deployment and logging in the macro architecture.
- The microservices platform **restricts the options** the team can choose in the macro architecture. After all, it is impossible to operate microservices that cannot be run on the platform. For example, the platform can define restrictions concerning the programming language.
- The microservices platform can **enforce** compliance with macro architecture rules. If developers disregard the rules, the microservices cannot run on the platform, ensuring that the rules are actually observed.

Specific platforms

The following two chapters show two different approaches for microservices platforms.

- **Kubernetes** (see chapter 13 (<https://www.educative.io/collection/page/10370001/5441945024331776/4922985196552192>)) can run Docker containers and solves challenges such as load balancing, routing, and service discovery at the network level. It is quite flexible since it is able to run arbitrary Docker containers. With Operators (<http://coreos.com/operators>) or Helm (<https://helm.sh/>), other services can be integrated into Kubernetes for monitoring.
- **Cloud Foundry** (see chapter 14

(<https://www.educative.io/collection/page/10370001/5441945024331776/5727581686988800>)) supports applications. All you have to do is provide Cloud Foundry with a Java application. Cloud Foundry creates a Docker container from it, which can then be executed. Cloud Foundry also solves load balancing, routing, and service discovery. In addition, Cloud Foundry includes additional infrastructure such as databases.

QUIZ

Z

1 Suppose at a company, the decision on whether to run your platform on a public cloud or to run it in your own data center lies with you. What is a factor that you will consider when making your decision?

- ☐ A) Running on a cloud requires a lot of complex installation.
- ☐ B) Running on a data center requires a lot of one-time but complex installation.
- ☐ C) Running on a cloud and running on your own data center don't have any significant advantages or

...can have any significant advantages or disadvantages over each other.



Submit Answer



Question 1 of 3
0 attempted



Reset Quiz 

In the next lesson, we'll look at some variations in the microservices platform approach that we have looked at so far.

← Back

Introduction

Next →

Variations



Mark as Completed



Report an Issue