DATA ENGINEERING IMPLICATIONS OF THE DEVOPS ARCHITECTURE

1. Data Engineering Implications of the DevOps Architecture

00:00 - 00:19

DevOps is used to manage the change management of software securely and productively. This software could be online products or internal systems. Data Engineering manages the pipelines that help with data flows. Let's take a look into Data Engineering's role in the DevOps Change Management System.

2. Microservices architecture

00:19 - 00:52

Microservices are small-scale software programs deployed as independent units. Each microservice is like a single part of a honeycomb. Large organizations may have hundreds to thousands of microservices. Each microservice takes care of a specific functionality and has its own data and logic. They have their service databases that are used to store data. Microservice logic transforms the data and API's are used to send and receive data with other system components.

3. Monolithic architecture

00:52 - 01:18

A monolithic architecture is the opposite of a microservices architecture. In monolithic architecture, there are no microservices, and the software is deployed as a single big unit. Monolithic architecture is much simpler than microservices architecture; however, its maintenance is challenging and risky as the software grows. Even a single problem in monolithic software can bring everything down.

4. Databases in monolithic vs. microservices architecture

01:18 - 02:15

In the monolithic architecture, a limited number of databases are used by the whole application. The data engineering applications for the monolithic architecture are much simpler than a microservices architecture. Monolithic architectures might be a viable option for small-scale applications like a school grading system. Data Engineering of such a system can be as simple as ingesting students' names and grades and storing them in a database. In the microservices architecture, each service has a database of its own. The microservice can use its own database and also inject data from other databases. Microservices must do API calls to reach another service database. This approach creates a lot of complexity, but it is a viable option for large organizations. Because in large organizations, different teams can specialize in different microservices.

5. Microservices private databases

02:15 - 02:49

Due to the independent nature of the microservices architecture, they are handled separately by different teams. When a microservice is changed, the others are usually not impacted. But although they are maintained differently, some product functionalities require collaboration from two or more microservices. Microservices can not freely access each other's databases. So they continuously communicate with each other through their APIs.

6. Data Engineering applications in microservices

02:49 - 03:00

Although the microservices manage their data, they still use centralized databases to record and replicate their records. Records on the private databases are sent to centralized databases to make the data available for analytics purposes.

7. Data Engineering applications in microservices

03:00 - 03:17

A data pipeline is a group of tools and methods to ingest, transform, and move data. In the example shown here, the private service databases of the microservices are replicated to the independent databases.

8. Let's practice!

03:17 - 03:33

Data is the bread and butter of software and needs to be well managed. Data engineering empowers organizations to use it to its full advantage. Now let's practice how data engineering is employed in a microservices architecture.