OBSERVABILITY AND RELIABILITY

1. Observability and Reliability

00:00 - 00:12

DevOps is as much about security and reliability as it is about productivity. Let's look at how DevOps helps ensure observability, quality, and reliability.

2. Observability

00:12 - 00:33

Observability is the ability to monitor the inner parts of the software. Complex software systems consist of different components such as microservices, APIs, databases, and other parts. Observing all of these parts is essential to see if they are up and behaving correctly.

1. [1] https://www.istockphoto.com/nl/vector/het-lijnpictogram-van-de-verrekijker-oceaanconcept-verrekijkerteken-op-witte-gm1245624425-363092800?phrase=field%20glass

3. Why is observability needed?

00:33 - 00:56

Modern software consists of many distinct parts that are tied together. When a problem is identified in the software, it can be tough to detect its source because all components are connected to each other. Observability allows us to detect the root cause of the problems because it monitors specific parts.

4. Alerting

00:56 - 01:15

Teams set up alerts that would let them know as soon as there was a problem with their software. For example, when setting up a data pipeline, a data engineer can configure it to send them an alert in case the pipeline fails to run. This alert enables the engineer to react fast, solve the problem with the data pipeline and make it run again.

1. [1] https://www.istockphoto.com/nl/vector/notification-bells-icon-gm1364077374-435399123?phrase=alerting

5. Why is reliability important?

01:15 - 01:59

Reliability is one of the essential characteristics of software. It refers to how many times software fails to do its operations. Some software requires exceptionally high reliability. Reliability in government services and defense systems is paramount because a reliability problem in this software might have dire consequences. Furthermore, banking systems also require the highest levels of reliability because a disruption in banking services can harm the banks. High reliability is not limited to these examples, and much other software also requires high reliability.

6. How to increase reliability?

01:59 - 02:08

There are two critical principles to increase reliability: Testing and Observability. DevOps is significant for reliability because it automates testing and enforces observability in change management.

7. Testing vs. observability

02:08 - 02:33

Testing enables teams to detect the problems before they go to market and impact users. In DevOps, testing ensures the software is reliable enough to face the users. Observability detects problems after the software is released to the market and interacts with users. Observability is needed even if the software is tested enough because the software might behave differently against users.

8. Reliability in microservices

02:33 - 02:58

Microservices architecture helps increase reliability because it handles different software parts separately. Microservice A is upgraded from version one to version two in this example, getting through the DevOps cycle. DevOps is very beneficial for reliability because even if Microservice A suffers a reliability problem through this change, Microservice B will continue to function well.

9. Let's practice!

02:58 - 03:08

Observability and reliability are essential concepts of DevOps. Let's hop into the exercises and start practicing.