

DEVOPS CULTURAL CONCEPTS

1. DevOps Cultural Concepts

00:00 - 00:09

DevOps is more than the tools and technologies used. It comes with a series of cultural concepts which are essential for DevOps.

2. DevOps culture

00:09 - 00:18

DevOps and all related systems depend on humans to operate —certain behaviors and concepts are required because they enable DevOps.

3. Concepts

00:18 - 00:30

The main concepts associated with DevOps are Collaboration, Autonomous Teams, Shared Responsibilities, and Post-Mortem. Let's look at them one by one.

4. Collaboration

00:30 - 00:56

Collaboration is at the heart of DevOps. The main principle of DevOps is the collaboration between the Development and Operations teams. Without collaboration, development and operations teams operate separately, which creates a lot of overhead and slows down innovation. DevOps want the same people to do the development and operation; therefore, no time and resources would be lost.

5. Autonomous teams

00:56 - 01:30

Autonomous teams mean that these teams have all of the required skillsets within team. Generally, the team can operate using its own resources. DevOps teams have various members with the necessary skills, making them autonomous and self-sufficient. Development teams are independent and have members specialized in different subjects like operations, product engineering, and data engineering.

6. Shared responsibilities

01:30 - 01:54

Autonomous teams are responsible for certain parts of the software. For example, they might be responsible for developing and maintaining a microservice. Since the teams are autonomous, they are responsible for developing and running their microservice and will be responsible for maintaining it. This principle is sometimes referred to as "You build it, you run it." The DevOps

teammates are responsible for developing and running the software and must rely on their own teams.

7. Automation

01:54 - 02:32

Automation is key to DevOps. DevOps, especially the CI/CD pipelines, helps us automate change management, testing, and deployment. Automation is a time saver; it helps the team focus on meaningful work and not worry about simple tasks. It also streamlines the change management process so all the outputs are of the same quality. For example, without the CI/CD pipelines, a developer should manually test the software and then have to manually deploy it, which is not the best use of their time.

8. Post-mortem

02:32 - 03:16

Mistakes are unavoidable. Most software suffers some problems regardless of how well-maintained they are. When a problem is spotted with software, it must be resolved. After it gets fixed, it is essential to document the issue and think about how it could have been prevented or better handled. This approach is called Post-Mortem. In Post-Mortem meetings and documents, the team reviews what happened and finds a way to prevent similar issues in the future. A most important point about Post-Mortems is that no team member should be called out or blamed. Instead, the team should approach it from a systematic control standpoint.

9. Outputs of a post-mortem

03:16 - 04:07

In a post-mortem, specific questions are very common. The first common question is: "How could this issue have been prevented?" Typical answers to this question could be: we should have conducted more testing; another standard solution could be reviewing the code better without deploying it. The second common question in a post-mortem is: "How could this issue have been better handled?" The typical answer to this question could be: "We should have implemented more observability and monitored our systems better." More observability helps catch the issues early on and identify their root cause. Identifying the root cause is very important to resolve the problems. As you can see, post-mortems are conducted in a blameless way.

10. Let's practice!

04:07 - 04:15

DevOps cultural concepts define how the teams will behave in good times and in bad. Let's hop onto the exercises and start putting them to use.