SquareOps Tech

7 C's Of DevOps

Swipe next →

SquareOps Tech

1 Continuous Development

In Continuous Development, code is written in small parts instead of all at once. This method is key in DevOps because it makes the process more efficient. As soon as a new piece of code is written, it's tested, built, and put into use. This approach helps improve the quality of the code and makes it easier to fix any issues, like flaws or vulnerabilities. It also allows developers to focus more on writing high-quality code, enhancing the overall development process.

Swipe next ⟶

SquareOps Tech

Continuous Integration

Continuous Integration in DevOps can be broken down into four main steps:

1. **Getting the Source Code:** Developers write code on their local machines and then upload it to a source code management (SCM) system like GitHub. Here, others with access can view, download, and modify the code as needed.

Swipe next ⟶

2. Building the Code: Tools like Maven are used to compile the code into software packages, such as WAR, JAR, or EAR files, and to run unit tests to check for errors.

3. **Code Quality Review:** Tools like SonarQube review the code to assess its quality. SonarQube checks for coding standards and other quality metrics, and then produces a report in HTML or PDF format to highlight any issues.

4. **Storing Build Artifacts:** Once the code is built and tested, the resulting files (artifacts) are saved in a repository like Nexus, making them available for future use or deployment.

Swipe next ⟶

SquareOps Tech

3 Continuous Testing

Any company can implement continuous testing by using agile and DevOps practices. Depending on need, we can use automated testing tools like Testsigma, Selenium, and LambdaTest to test our code. These tools help us find and fix issues quickly and efficiently. With the help of a continuous integration platform like Jenkins, we can automate the entire testing process, which makes things even more streamlined and efficient.

Swipe next →

SquareOps Tech

Continuous Deployment

Continuous Deployment is the process of automatically deploying an application into the production environment when it has completed testing and the build stages.

Swipe next ⟶

SquareOps Tech

Continuous Monitoring

This is a crucial part of the DevOps lifecycle because it helps catch problems before they become serious. Using tools like Prometheus and Grafana, teams can keep an eye on important performance indicators such as CPU usage, memory usage, network traffic, how fast applications respond, and error rates. Prometheus collects this data, while Grafana helps visualize it, making it easy to monitor trends and issues over time.

Swipe next ⟶

SquareOps Tech

Continuous Feedback

Once an application is released, end users start using it and provide feedback on its performance and any issues they encounter. The DevOps team collects and analyzes this feedback, then coordinates with the developers to fix any identified problems. This process helps in reducing errors and improving the application’s efficiency. Continuous feedback from users not only enhances the application's performance but also streamlines the deployment process.

Swipe next ⟶

SquareOps Tech

7 Continuous Operations

We will maintain higher application uptime through continuous operation, which will help us reduce maintenance downtime that negatively affects user experience. Benefits of continuous operations include increased output, reduced production costs, and improved quality control.

Swipe next ⟶

SquareOps Tech

For more such post follow **SquareOps page** and for more informational blogs visit

squareops.com

SquareOps
TECHNOLOGIES