

Improved Deployment Strategies for Preventing Global Disruptions

July 19, 2024

The recent disruption caused by a [CrowdStrike](#) update and impacting [Microsoft](#)'s systems highlights the critical need for robust deployment strategies. None of these practices are new; they could have been applied to prevent such widespread issues:

Blue-Green Deployment: CrowdStrike and Microsoft could have used blue-green deployment to ensure that the new update was fully tested in a production-like environment before shifting traffic to it. Running two identical environments—one with the current version and one with the new update—allows for seamless transitions and quick rollbacks if issues arise. If any issues were detected, traffic could easily revert to the stable environment without causing disruptions to the entire user base.

Canary Releases: Rolling out the update to a small subset of machines initially could have allowed for close monitoring of the update's impact. This approach enables early detection of any issues before they affect a larger audience. Problems could have been identified and resolved at an early stage, minimizing the risk of widespread disruption.

Feature Flags: Using feature flags to gradually enable new features for a small group of users (or servers in this case) while keeping the rest of the system unchanged allows for dynamic control over which features are active. Any problematic features could have been quickly disabled without requiring a full rollback or redeployment, reducing the impact on end users. This technique is not always possible, but as we don't know the technical details (yet?) I had to mention it.

Rolling Updates: Deploying updates incrementally across different servers or regions helps in identifying potential issues in a controlled manner. By isolating the update's impact to smaller segments, it would have been easier to address issues without affecting the entire planet and provoking a global disruption as we all suffered today.

Automated Testing and CI/CD Pipelines: Comprehensive automated testing and Continuous Integration/Continuous Deployment (CI/CD) pipelines could have ensured that only thoroughly tested code reaches production. Early detection of bugs and issues in the development cycle would significantly reduce the likelihood of problematic updates being deployed. I am sure CrowdStrike's DevOps team uses CI/CD pipelines, so, what did they miss?

These strategies are well-established best practices in the industry and could have been implemented to avoid such a significant incident. As I said, the lack of technical details is still significant, but at some point, during the next weeks or months, we will know what the

failure was and why it was not detected until it made thousands or hundreds of thousands of servers crash and show the Blue Screen Of Death.

Who's at Fault?

From the end users' perspective, the outage is Microsoft's fault. They are the face of the service, and users rely on Microsoft to ensure stability and reliability. When systems crash, end users are unlikely to know or care about the involvement of CrowdStrike; they hold Microsoft responsible.

From Microsoft's perspective, the fault lies with the CrowdStrike update. CrowdStrike's failure to follow professional rollout procedures led to the disruption. However, Microsoft is also accountable for not having sufficient safeguards in place to prevent such third-party issues from causing widespread disruptions.

Regarding their agreement, most software service agreements include clauses about service continuity and responsibilities for outages. These agreements typically outline that vendors must adhere to certain standards and practices to avoid disruptions. In this case, it would be crucial to examine the specific terms between Microsoft and CrowdStrike to determine liability and responsibility for the outage. Such agreements often include indemnity clauses, service level agreements (SLAs), and specific protocols for managing updates and deployments to minimize risks.

Ultimately, both companies share the blame—CrowdStrike for the faulty update and Microsoft for not mitigating the risk effectively. This incident should prompt a reevaluation of their deployment strategies and inter-company agreements to ensure higher resilience against such disruptions in the future.