

Role Of DevOps in minimizing downtime on Applications using Continuous Deployment

Renugadevi Anbarasan, Yuxin Meng

1. Introduction of Downtime

As we all know, time is money. And for the computer industry, time is even more precious, however, the occasional network failure leading to downtime can result in considerable financial losses for the company. Downtime is a computer industry term that refers to the time when a computer or IT system is unavailable, offline, or not working. The Business Dictionary describes the downtime definition as: "Period during which an equipment or machine is not functional or cannot work. It may be due to technical failure, machine adjustment, maintenance, or non-availability of inputs such as materials, labor, power." [1] In addition, downtime in an industrial environment may also refer to a failure of production equipment. In this article, we focus on downtime in a computer environment.

There are many reasons for downtime, including maintenance downtime (called scheduled downtime), human error, and software or hardware failure. It can be divided into planned downtime and unplanned downtime depending on the causes.

Planned downtime is the time when the IT department needs to do the necessary maintenance and upgrades to the system to shut down the network. Although the network is down at this time, will affect the use of the system, but the planned downtime is inevitable, it can make the network function to maintain a better continuity. There are several reasons for the planned downtime: hardware replacements, system diagnostics, configuration updates and so on. Taking advantage of planned downtime can minimize losses, and taking advantage of demand cycles can enable employees to perform other tasks efficiently even when network conditions are affected.

Unplanned downtime is what people don't want to happen, and it does more harm to the business. Due to an unexpected system failure, an unexpected network outage can occur at any time. If the enterprise is not ready to deal with it in time, then this will be the most costly type of downtime. As for the causes of network downtime, there are many reasons that can lead to unexpected network failures. For example, 97% of IT personnel stated that human error is the cause or a contributing factor in at least some network outages. [2] Besides, hardware failures, disaster or crisis situations and other situations could also cause the unplanned downtime.

In IT environments, downtime can be one of the metrics used for system availability. There is a "Five-9s" usability criterion, which means that the usability is 99.999%. Service level agreements (SLAs) often use monthly downtime or availability percentages for billing calculation. [3] As mentioned above, planned downtime due to system maintenance, etc. is usually not included in the availability percentage of the SLA.

2. Impacts of downtime on deployment

When the system failure, there will be downtime, which will bring great loss and impact. While companies can schedule planned network downtime to minimize these costs, unplanned downtime can lead to a number of unexpected costs that can have an impact on deployment.

Customer loss. One of the effects of downtime is losing customers due to downtime issues. This problem is particularly acute in business-to-business entities. There are a lot of transactions going on the Internet now. If the network service is interrupted, then these transactions will be forced to interrupt. If the service is down for a long time, it means that the customer has to wait for a long time. This can lead to impatient customers looking for other platforms, which can eventually lead to fewer customers.

Data Security. The deployment process needs to secure the data. And modern Enterprises survive and die because of their applications, many of which are located in data centers. If the data center fails, the application crashes with IT, paralyzing IT and paralyzing the business. [4] So IT must be able to withstand the impact of a data center outage, and downtime is bound to have an impact on the data center. In addition, data loss and damage to customers may result in other costs.

Loss of productivity. It takes a lot of energy to update a system at once, and many companies choose to shut down at night when usage is low, which makes employees work all night and leads to low productivity. And the probability of error increases with each deployment operation, so you need as little downtime as possible. Sometimes employees and third parties need some kind of web service to get the job done. When these services are unavailable, affected parties will be unable to perform certain tasks, which reduces overall productivity and makes deployment difficult.

SLA penalty. As reliance on data, IT, and network services increases, more and more companies require formal assurances from service providers, and as service-level agreements (SLA) become more likely to deliver services at their desired level. As described in the previous section, the SLA is an agreement that covers downtime. One indicator used in the SLA is availability. Failure to ensure uptime will result in financial penalties. Long downtime can affect the availability of their systems, which in turn can affect deployment. If the agreement cannot be fulfilled as a result, then there will be financial penalties.

3. Use of DevOps to minimize the downtime

To reduce IT downtime and its negative impact, companies need to find solutions that address the needs of their customers. At the same time, the solution needs to have a certain degree of security and convenience. As a result, many people opt for a DevOps strategy to meet demand and avoid downtime.[5] When applied to reduce downtime, DevOps is designed to provide a faster and more efficient response to changing business needs.

Simply put, the main benefits that DevOps can bring to an organization are increased communication, collaboration, integration, and automation. Speed, quality, control, and cost are the four key factors. DevOps accelerates deployment by focusing on speed and automating as much as possible. DevOps also minimizes downtime by reducing the time required to solve application problems & changes the software architecture so that updates can be seamlessly integrated and implemented quickly.

For planned downtime, the deployment strategy needs to be improved if downtime is to be reduced. For example, the more widely used zero downtime deployment. Zero-down deployment (ZDD) allows you to deploy new versions of your system without disrupting existing services. From the user's and the company's point of view, this should be the best way to deploy, as new features can be introduced and bugs fixed without any disruption. There are a number of strategies to implement a zero-downtime deployment, and the following are some of the more widely used deployment methods. Blue/Green Deployment. Blue/green deployment is about building another complete production environment for the next release. The development and operations teams can safely build the next version of the product in this separate production environment. When the next release is complete, the user is automatically redirected to the next release by modifying the configuration. Canary Release. Canary releases are very similar to blue/green deployments, but do not require an additional set of production environments. Before releasing a new version of the entire software to all users, test it with real customer traffic by releasing the new version to some users to ensure that the software does not have serious problems and reduce the risk of release. Rolling deployment. Rolling deployment is the process of selecting one or more service units from a service cluster, performing version updates after the service is stopped, putting them back into service, and repeating them until all service instances in the cluster are updated to the new version, instead of updating all the service instances at once.

In the case of unplanned downtime, since these things happen at unexpected times, things we can do with DevOps are monitor metrics and implement effective maintenance strategies.[6] To monitor system information, you need to collect system data to help you understand the cause of the outage, such as the log, the error type, and when the outage occurred. With proper monitoring, it is possible to monitor and set up problems before they become larger and cause downtime, thus avoiding some of the downtime costs. Alerts will enable the team to keep an eye out for problems as they arise. Like most monitoring tools, alarm tools require a high degree of flexibility and are expected to be customized to needs. In addition to monitoring metrics to prevent further problems, it is important that the system has an efficient maintenance strategy when problems occur. You may solve problems that arise by periodically inspecting your software systems according to maintenance guidelines.

4. Pros & Cons in CI/CD pipeline

CI/CD being a subset of Continuous Integration, the developers and the deployment team ensure the update is releasable in the system. Continuous deployment, like Continuous integration is associated with quantifiable threats. Once they are in operation, the pipelines become deployable unit routes.

CI/CD is major pathbreaker in web services & SaaS application where the new features are being deployed constantly without downtimes. The development pipelines could well be in line with the latest deployment thus pushing a new feature without outage or suspension of the services rendered. Integrating the current technologies with legacy applications often maintain the believability of the service also providing new additional feature [7]. This Continuous delivery model is often appreciated hands down because of its nature in incorporating technologies & applications by making them usable to the vast majority of audience.

The Continuous deployment reduces the MTTR (Mean Time To Resolution) and makes it shorter and faster. It also isolates the fault so that the rest of the application/service works without being in a Stalemate. Increased and more streamlined release processes enable faster handling of issues towards resolution and make it available for deployment. It also improves the test reliability due to more minor & precise updates. Because of the said nature, it decreases the amount of errors in CI/CD pipeline thus making happy customers.

The major challenge in the CI/CD pipeline is choosing the right automation procedure. There is no defined rules to pick the right delivery model, but it's important that the chosen model stands for the organization's values without compromising the quality, speed and efficiency. Adopting a framework and an established way of working among the teams involved is always a challenge in large organizations. The developers of a dashboard are usually not the ones that will be using the dashboard. So, there are high chances we fall victim to logical fallacy. Therefore, getting the requirements right is as important as the development itself [8]. The DevOps plays a humongous role in bridging the gap between the IT & business customers and it is important that the DevOps team is built right.

Involving more teams in the CI/CD pipeline could be crucial because keeping them in sync and also autonomous requires human intervention. Some of the CI/CD between these teams could be connected and can only be deployed together where the said human element plays a big role.

5. Development Trend

DevOps aims to provide a collaborative environment where development and operations teams can design, validate, and deliver applications rapidly, efficiently, and regularly. The organizations meet their goals quicker and deploy new features, bug fixes, and security updates with high altitude, resulting in seamless implementations. Beyond simply adopting this strategy, it is essential to embrace the recent technology innovations to obtain the maximum return on investment [9].

When it comes to adopting the innovative technologies/solutions, a lot of organizations are open to use Infrastructure Automation (**IA**) tools, Application Release Orchestration (**ARO**) tools, Application Performance Monitoring (**APM**) software, etc.

6. Conclusion

IT teams often struggle with turning big business goals into projects and processes, like the development and deployment of applications. Reducing downtime while cutting costs is key for any business, therefore driving innovation by bringing on DevOps team needs to be a core focus for any organisation. DevOps teams are designed to enact cultural and technological changes within organisations, and are typically built around the sole aim of reacting faster and more effectively to changing business demands.

References

1. Alex. (2019, June 21). DevOps: How to Have Less Downtime. Alibaba Cloud Community. https://www.alibabacloud.com/blog/devops-how-to-have-less-downtime_594973.
2. Collins, E. (2020). The Cost of Downtime. Nutanix. <https://www.nutanix.com/blog/the-cost-of-downtime>
3. K, B. (2017, September 6). DevOps is the key to reducing IT downtime. Best DevOps. <https://www.bestdevops.com/devops-is-the-key-to-reducing-it-downtime/>
4. Markle, C. (2020, September 10). What Is Downtime? Definitions and Best Practices. Beekeeper. <https://www.beekeeper.io/blog/downtime/>
5. Suse. (2018, February 20). What is Downtime? Answer from SUSE Defines. SUSE Defines. <https://susedefines.suse.com/definition/downtime/>
6. What is Network Downtime? Unplanned; Planned Downtime. Worldwide Services. (2020, September 18). <https://worldwideservices.net/what-is-network-downtime/>
7. Resource Center - Genesys Cloud https://help.mypurecloud.com/articles/benefits-continuous-integration-continuous-deployment-ci-cd/?fbclid=IwAR1RxB3w11YQ-0ekmXKYJilRqeIyQWj8AK4YFD-OdpuSj_M_OfpFjz_1Gzc
8. Tarun Manrai. (2020, February 5). Pros and Cons of CI/CD Pipelines <https://manrai-tarun.medium.com/pros-and-cons-of-ci-cd-pipelines-1d9c1ec1d242>
9. Andreja Velimirovic (2020, December 16) Top 8 DevOps Trends for 2021. <https://phoenixnap.com/blog/devops-trends>