The Emergence of DevOps

Written for Rootstrap, Inc.

By Ravi Das

Introduction

In our last whitepaper, we took a closer examination as to what DevSecOps is all about. Essentially, it is the union, or formation of three different departments within an organization:

> ➢ The software team;
> ➢ The IT security team;
> ➢ The operations team.

One of the ultimate goals of DevSecOps is to instill a proactive mindset to creating source code that is secure, and free from any vulnerabilities, gaps, or weaknesses. A theme here as well is in automation, especially when it comes to the various phases of the Software Development Lifecycle (SDLC).

We provided a formal definition of what DevSecOps, and focused on three main areas:

> ➢ How the concept of security can be implemented at each stage of the SDLC;
> ➢ How compliance controls can be deployed in an effort to make sure that the source code which is developed is up to speed with compliance regulations of the GDPR, HIPAA, CCPA, etc., especially when it comes to data privacy;
> ➢ The trends of DevSecOps;
> ➢ The benefits of DevSecOps.

In this whitepaper, we sort of continue with the same theme, but instead we take it out the security aspect of it, and focus primarily on DevOps, and why it is growing in popularity. It is broken down the into the following sections:

> ➢ The latest statistical trends;
> ➢ The trends of DevOps;
> ➢ The benefits of DevOps;
> ➢ Three use cases of DevOps:

>> *Databases;

>> *The job market surrounding DevOps;

>> *The usage of DevOps tools.

The Latest Statistical Trends

Here are some compelling reasons why DevOps is here to stay, and why it will explode into the future:

*60% less time is devoted to resolving support cases;

*There has been an over increase of 33% in terms of infrastructure improvements when DevOps is used.

According to a recent study conducted by Atlassian discovered the following:

*99% of the respondents felt that DevOps had made an overall, positive impact on their organization;

*There was an overall increase of 61% in terms of the quality of the projects that were delivered to the clients;

*The delivery to market increased by 49%;

*49% of the respondents said that using DevOps helped increase the productivity of source code creation at the modular level;

*Of those polled, 53% of those organization have been using DevOps now for at least three years or more, with no plans of changing course.

(SOURCE: 1).

According to a New Harvard Business Review Analytic Services Study:

*86% see the importance of delivering software projects on time, and this can only be done with DevOps;

*77% of those organizations that were surveyed claimed that that they are already using DevOps to varying degrees, or will plan to in the very near future;

*51% of the companies that were involved in this survey claim that they DevOps is currently being used not only on just existing projects, but new ones as well.

(SOURCE: 2).

According to a recent study by Anitian:

*90% of software development projects will make use DevOps in 2022, which is a sharp increase of 40% when compared to 2019;

*60% of software development teams will be adopting the practices of DevOps, when compared to only 20% back in 2019;

*70% of the companies surveyed will have incorporated automation into their SDLC as a result of adopting DevOps;

(SOURCE: 3).

Another survey that was launched by CDNetworks demonstrates the value that DevOps brings to the table:

*There was a 46% increase in the delivery rate of software projects to the client;

*There was a 96% increase in the speed to recover from any vulnerabilities, gaps, or holes that were discovered in the source code;

*Organizations would be twice as likely to meet KPIs and other quantitative metrics from both a financial and nonfinancial standpoint

(SOURCE: 4).

VAR Insights provided a more global view into the popularity of DevOps, and here is how it will grow:

*Overall, DevOps will grow at least by 18.3% to become a $20.3 billion market by the year 2028;

*The time to release new software projects from the development to production environments will be 63% faster.

(SOURCE: 5).

The Trends For DevOps

Here are some of the more noteworthy trends for DevOps in 2022:

1) A bigger move to the Cloud:

    Although the benefits of the Cloud were realized before COVID-19, the pandemic has catalyzed the movement to it even more. For instance, the days of maintaining an On Premises Infrastructure will soon be gone, and it is expected that just about everything will be stored and accessed on some sort of Private, Public, or even Hybrid Cloud platform either in the AWS or Microsoft Azure. The DevSecOps model will be even more needed now than ever before, especially when it comes to preventing data leakages, and software development that will now fully evolve on Virtual Machines (VMs) and Virtual Desktops (VDs).

    In fact, this further adoption to the Cloud will fuel another trend that will strongly correlate to DevSecOps: Infrastructure as Code, also known as "IaC" for short. This is where Cloud platform is not managed by manually or even by automated processes, but rather it is done through specific configuration files which contain the specifications of your infrastructure.

2) Edge Computing will take off:

    With Remote Workforce, pretty much all employees are scattered hundreds, if not thousands of miles apart from one another. Trying to gain access to the needed shared resources from a central VM can still take more time and processing power. As a result, you will see that VMs will be placed much closer to the points of where the remote employees reside at, geographically. The end result is that not only access, but data processing times will be greatly reduced. This is also technically known as "Edge Computing". When this trend starts to occur in greater momentum, it will be one of the primary responsibilities of the DevSecOps teams to make sure that they are deployed and configured properly.

3) A greater dependency on others:

    As the world becomes even more digital, the reliance upon using external, third-party vendors will grow even more. But gone are the days of implicit trust, now business entities are shifting towards a mindset of zero trust, with one of the primary drivers of this being the recent Solar Winds attack. Suppliers will now have to be carefully vetted to even greater degrees, and a fair share of this process will fall onto the shoulders of the DevSecOps team. In the end they will have to make sure that the security protocols implemented at third party are at least in par with the levels that you have set forth for your organization.
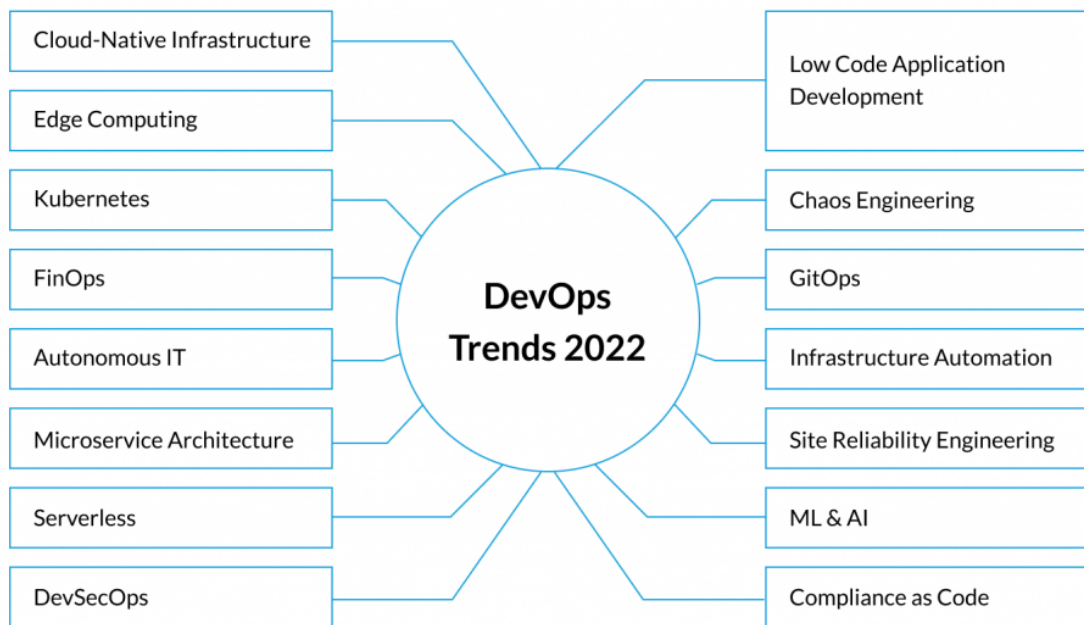
4) The use of AI:

    Although there is really nothing new about the concepts of AI, its market applicability has only happened recently. A key industry that is making use of this is the manufacturing and logistics/supply chain industries, especially when it comes to overseeing the use of robotic

processes. But in the world of software development, it is anticipated that its main use will primarily grow for the automation of routine tasks.

5) A greater need for measuring and observability:

This is where the role of Application Performance Monitoring Software will come into play. With this, software developers can get immediate from their peers about their source code, This will also be very critical in reducing the Mean Time To Repair (MTTR). APM tools will also give software developers the ability isolate and prioritize issues in a quicker fashion.

Overall, the impacts that DevOps will have on different market segments and applications is illustrated in the diagram below:

Cloud-Native Infrastructure		Low Code Application Development
Edge Computing		Chaos Engineering
Kubernetes	DevOps Trends 2022	GitOps
FinOps		Infrastructure Automation
Autonomous IT		Site Reliability Engineering
Microservice Architecture		ML & AI
Serverless		Compliance as Code
DevSecOps		

Design by **dp** DEV.PRO

(SOURCE: 6).

The Strategic Benefits of DevSecOps

Some of the benefits of DevOps in 2022 include the following:

1) Long term cost savings:

By implementing security at every step of the way, you will in some fashion or another realize cost savings. Just imagine if you have never tested the web application, and it was released to the customer, and over a short period of time, they have discovered a number of bugs in the functionality of it. All fingers will be pointed back at you, and the cost will come out of your own

bottom line to repair them. But by implementing security into your S-SDLC at the beginning stages, this probability from this from actually happening will be greatly minimized.

2) <u>You will be able to deliver on time</u>:

As it has been discussed in this whitepaper, many software development teams often wait until the very end to security test the source code. Unless this has already been factored into the timeline, this can create a huge delay in the expected time of deliver. For instance, if there are many bugs that are discovered, then it simply means that much more time and extra will have to be devoted to remediating them completely. Also, by waiting until the very end, the chances are greater that a more haphazard job will be done, creating even more problems for your client in the end. But by testing on a modular basis, and in a parallel fashion, you can pretty much guarantee that the project should be delivered on time.

3) <u>Everybody will feel a sense of responsibility</u>:

One of the biggest goals in Cybersecurity is to have companies take a much more proactive mindset, not only internally, but externally as well. In other words, the thinking that it is entirely the responsibility of the IT Security to make sure that all is protected needs to quickly disappear. The silos that have separated employees in this regard need to come down. Everybody has a responsibility to security, and by having the DevSecOps work as one harmonious unit, this will foster that kind of proactive mindset by showing that it can be accomplished, for the long term. In other words, a sense of "federation" will emerge, which will be a solution where everyone benefits.

4) <u>An increased sense of perceived value</u>:

When customers look to purchase products and services, they not only look at the features and benefits of it, but in today's world, they are now looking at just how the secure the product actually is. A good example of this are the Internet of Things (IoT) that are designed for home use. Prospects are now looking at just how secure they are when it comes to connecting with other devices, whether it is a tangible product or even a software application. For example, by pointing out the security process and checking that have been put into the creation of the web application will not only lead to an increased perceived value of it, but it will also mean increased sales.

5) <u>Greater chances of being compliant</u>:

It is important to keep in mind that the concepts of DevSecOps does not apply to just the world of the S-SDLC. It can also be used in other departments as well, most notably that of Accounting, Human Resources, and Finance. By using the same kind of framework, you should be able implement controls to help safeguard any confidential information and data that you have stored in your database in an efficient and prompt manner. By doing so, you will become compliant with the tenets as set forth the data privacy laws of the GDPR, CCPA, HIPAA, etc.

The illustration below depicts some of the other benefits of DevOps:

208x MORE frequent code deployments

106x FASTER lead time from commit to deploy

2,604x FASTER time to recover from incidents

7x LOWER change failure rate (changes are $^1/_7$ as likely to fail)

(SOURCE: 7).

The Market Applications of DevOps – The Database

In our last whitepaper, we reviewed how quickly the uptick is for adopting DevOps as it relates to creating source code that is secure. There are other areas where DevOps is growing, especially when it comes to databases. This can be considered the heard of any Web based application, as it contains not only the Personal Identifiable Information (PII) datasets, but it also consists of all of the transactions that customers make from the front end.

The DBA obviously has a vested interest in the creation and optimization of any database, as everybody wants to have as much of a seamless experience as possible. But the role of the DBA has often taken a backseat until the application is almost ready to launch. In fact, they have even been chastised for being the bottleneck in the entire process.

The Second Age Of DevOps

But this is no longer the case as people are starting to realize the potential that DevOps has to offer in the database creation process, as we now approach what has been termed the "Second Age of DevOps". This can be broken into three distinct phases:

1) A central point of Source Control:

   All of the database related code will reside in one central repository (in fact, this can be viewed as a miniature database) so that everybody involved in the development process will have access to it, assuming that they have they have been authorized to do so and have been properly authenticated. This holding place will consist of the following items:

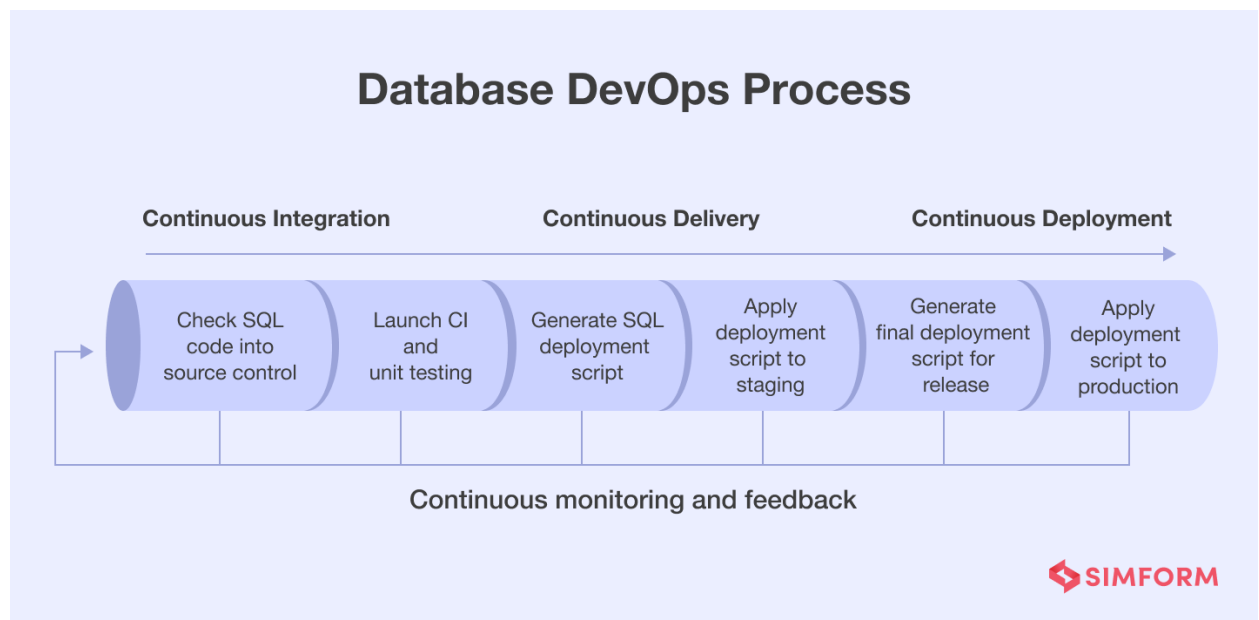   ➢ Keeping secure tabs on both static and dynamic datasets;

➢ All database configuration files, regardless of the platform that is being used (such as Oracle, MS SQL, MySQL, etc.);

➢ Any scripts for automation, again independent of coding language (such as PERL, PHP, JavaScript, etc.).

2) The automatic synchronization of other processes:

With this, the DBA will have the ability to test the database code very early on in the process, this eliminating the bottleneck issues just described. In other words, as other development work is happening, the creation and testing of the database can also evolve in a parallel fashion. This scenario is tailored after the following principles of database development:

➢ Continuous Integration;
➢ Continuous Delivery;
➢ Continuous Deployment.

This is illustrated in the diagram below:

Database DevOps Process

Continuous Integration   **Continuous Delivery**   **Continuous Deployment**

| Check SQL code into source control | Launch CI and unit testing | Generate SQL deployment script | Apply deployment script to staging | Generate final deployment script for release | Apply deployment script to production |

Continuous monitoring and feedback

⬥ SIMFORM

(SOURCE: 8).

3) Early security testing:

As with the case of testing of source code, the incorporation DevOps into the database process allows for the DBA to check for any vulnerabilities or gaps in a "Shift Left" fashion, this allowing for remediation to happen early on, when one build is done and the team is ready to move to the next build.

The Value Of DevOps In The Database

Implementing DevOps into the database world also brings in other value, such as the following:

➢ Automated processes for the provisioning of new databases (especially if you are making use of a Cloud based platform such as that of the AWS Microsoft Azure);

➢ The ability to back up databases in automated fashion per the requirements of your Security Policy;

➢ Allows for a wide margin of flexibility in case there are any last-minute database code changes that have to be made;

➢ Allows for a single, holistic view of all that is happening in real time to the development and testing of the database;

➢ Database queries can be modified, revised, edited, etc. on the fly without having to rewrite them scratch;

➢ It helps to greatly eradicate any silos that have may existed before. In other words, it fosters a sense of teamwork with all that are involved in the creation of the Web application, even the software developers themselves.

This can be seen in the diagram below:

The Ideal Database DEVOPS Capabilities

An automated database provisioning and configuration management.

An automated backup and recovery system.

CI/CD for database schemas and data migration.

A decoupled architecture that allows flexibility with code changes.

A holistic database testing and monitoring.

A centralized database with immediate visibility of queries during production.

Developer access to production data.

Collaboration and teamwork during troubleshooting

(SOURCE: 8).

<u>How To Implement DevOps Into The Database</u>

Ok, you might be asking how does one implement DevOps into the realm of the database? It can be done through this process:

1) <u>Get a grasp of the business requirements</u>:

   This is where you really have to get to know what the requirements of the client are, from the standpoint of the database. With this in mind, you have to carefully assess the following:

   ➢ The SDLC methodology that is going to be utilized;

   ➢ A preliminary view of what the database schema will look like based upon the discussions you have with the client;

   ➢ All of the SQL command lines that could potentially be used to answer any queries that could be posed to the database.

2) <u>Create the first cut of the database</u>:

   This will serve as the framework for what the final product will look like. As the database grows in size and complexity, there will of course be errors and omissions that will be made. But the beauty of DevOps is that you will be able to catch these early on, using automated tools.

3) <u>Start planning the integration</u>:

   One of the key things that DevOps emphasizes is collaboration with other teams. Thus, the reality here is that you will be working in close harmony with the software development team to refine the database from the last step just described. With this in mind, the integration of the database code into the primary source code should happen fairly quickly, without too many glitches.

6) <u>Implementing the database into a non-static environment</u>:
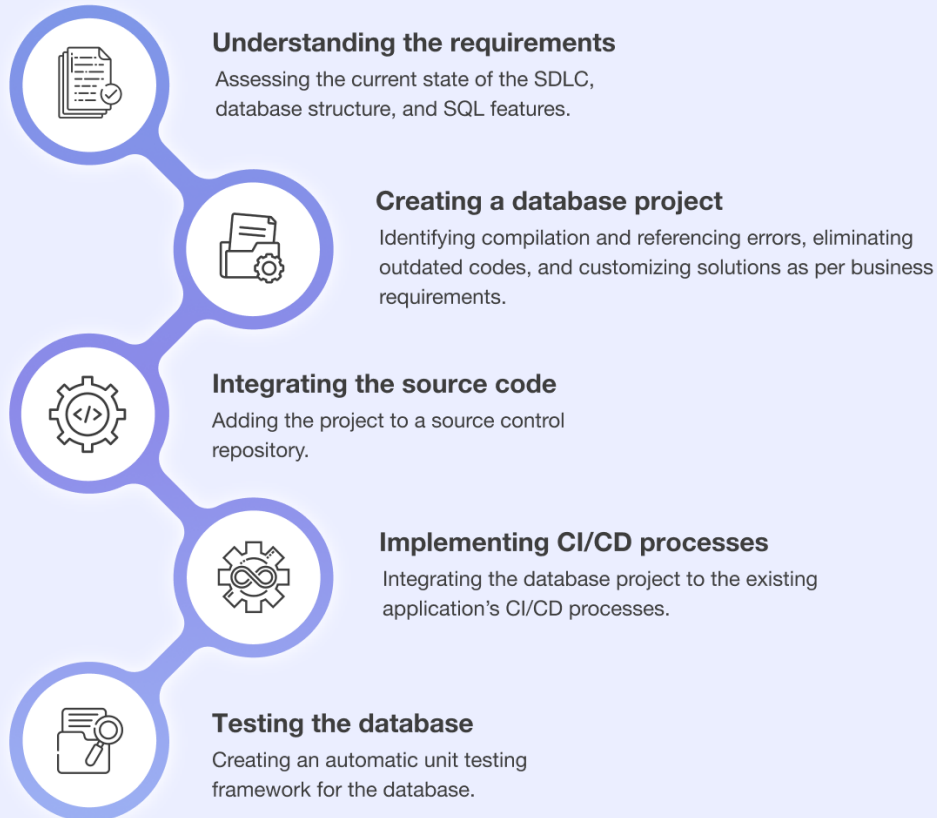
   Once the last step has been accomplished, you should be able to take the database product you have made and put it into the real time to processes of Continuous Delivery/Deployment (CD) and Continuous Integration (CI) , as described previously. But the beauty of using DevOps here is that you can use existing CD/CI flows, and not have to waste time in creating new ones.

7) <u>The final testing</u>:

   Although it is assumed you will be testing the various components of the database as they evolve in a modular fashion (just like the source code), you will want to give the newly created database a final spin. One of the key advantages of DevOps here is that this will be purely an automatic process, which should not take long to complete (assuming all errors and vulnerabilities have been fully remediated in the previous stages).

This entire process can be seen in the diagram below:

Steps for Database DEVOPS Implementation

Understanding the requirements
Assessing the current state of the SDLC, database structure, and SQL features.

Creating a database project
Identifying compilation and referencing errors, eliminating outdated codes, and customizing solutions as per business requirements.

Integrating the source code
Adding the project to a source control repository.

Implementing CI/CD processes
Integrating the database project to the existing application's CI/CD processes.

Testing the database
Creating an automatic unit testing framework for the database.

SIMFORM

(SOURCE: 8).

<u>The Macro Values of DevOps In Databases</u>

While it is the DBA that benefits primarily the most from the use of DevOps, there are other key stakeholders as well that it brings value to, ranging the from the C-Suite to the Project Manager that is responsible for the ultimate delivery of the Web application project.

Here is a sampling of them:

➢ Operational and Technical Support is further optimized and enhanced – issues can be responded to very quickly;

➢ The remediation process will occur in a timelier fashion than ever before;

➢ Automation is a guaranteed processing, thus eliminating administrative costs on redundant tasks;

➢ A sense of being an all for one and one for all mindset is quickly adopted, especially with the software development and operations teams;

➢ The company can stay on the cutting edge of Research and Development, this giving it a competitive advantage in the marketplace;

➢ Effective resource allocation is realized;

➢ The Project Manager and other relevant stakeholders can get a birds eye view on the metrics and other relevant KPIs as they relate to the development process from a unified dashboard;

➢ Projects are delivered to the client right on time (if not before) and right on budget (or even under budget);

➢ Effective management of the datasets can be done in shorter cycles and iterations;

➢ The version control process is totally automated, so all stakeholders can see the changes and enhancements that are being made on a real time basis. The end result here is that there is now a greater sense of both responsibility and accountability.

Conclusions

Overall, this whitepaper has examined the emerging popularity of DevOps, especially as it relates to database design and development. In this aspect, there are numerous DevOps tools that are popular, and the following illustration demonstrates this:

(SOURCE: 9).

A future whitepaper will do a much deeper dive into the features and benefits of them.

Sources

1) https://www.strongdm.com/blog/devops-statistics#:~:text=IDC%20predicts%20that%20the%20DevOps,end%20customer%20experience%20in%202021
2) https://www.ahead.com/resources/devops-stats-growth-value/
3) https://www.anitian.com/20-statistics-that-todays-devsecops-teams-should-know/
4) https://www.cdnetworks.com/web-performance-blog/5-statistics-that-prove-devops-practices-improve-web-performance/
5) https://www.varinsights.com/doc/devops-trends-here-to-stay-in-0001
6) https://dev.pro/insights/15-devops-trends-2022-from-dev-pro-experts/
7) https://newrelic.com/devops/benefits-of-devops
8) https://www.simform.com/blog/database-devops/
9) https://blog.inedo.com/best-change-management-database-software