Azure Cloud

In past all the tech giants and organizations (companies) have two teams

1. Development team and
2. Operation team.

**Development team**: receives the Clint’s requirements and according to the client requirement development team develops the code

The functions involved in it are:

1. planning
2. coding
3. building
4. Testing
5. Releasing

**Operation team:** here the released code is managed or maintained by the Operating team and it identifies the errors need of upgradation of code etc...in it.

The functions involved in it are:

1. Deploying
2. Operating
3. Monitoring

This process is called the software development life cycle (SDLC).

In earlier the organizations and MNC multinational company’s they were used many models for SDLC like

1. Water fall model
2. V- model
3. Increment model
4. RAD model
5. Spiral model
6. Prototype model etc.

This models have some limitations that once the code or project is build we cannot modify or update it. That means once requirements of the Clint are fixed we cannot change or upgrade it.

The biggest problem is that in the middle of software development, the customer asks to make changes in the software. It takes a lot of time and money to make these changes.

And also it is a time taking process and costly for building a client project by using separate development team and the operation team. And no flexibility in changing of software.

To overcome these limitations every organizations and tech giant are using agile model. And the agile model was proposed in the 1990s*.*

**Agile model:**

Agile model is a combination of iterative and incremental models, that is, it is made up of iterative and incremental models.

In Agile model, focus is given to process adaptability and customer or client satisfaction.

The agile model was created mainly to make changes in the middle of software development, so that the software project can be completed quickly.

This agile model have different phases in building a software they are:



1. Requirements / plan
2. Design
3. Development
4. Testing
5. Deployment
6. Review
7. Launch

**Advantages of Agile Model:-**

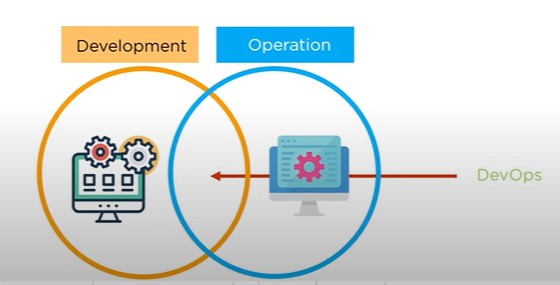
* In this, two programmers work together due to which the code is error free and there are very few mistakes in it.
* In this the software project is completed in a very short time.
* In this the customer representative has an idea of ​​each iteration so that he can easily change the requirement.
* This is a very realistic approach to software development.
* In this, focus is given on teamwork.
* There are very few rules in this and documentation is also negligible.
* There is no need for planning in this.
* It can be managed easily.
* It provides flexibility to developers.

**Disadvantages of Agile Model:-**

* It cannot handle complex dependencies.
* Due to lack of formal documentation in this, there is confusion in development.
* It mostly depends on the customer representative, if the customer representative gives any wrong information then the software can become wrong.
* Only experienced programmers can take any decision in this. New programmers cannot take any decision.
* In the beginning of software development, it is not known how much effort and time will be required to create the software.

DEVOPS

The term “DevOps” was introduced by combining software “development” (Dev) and “operations” (Ops.) The aforesaid term was coined by Patrick Debois in 2009 to make way for quick and effective delivery of software updates, bug fixes, and features.



Different people have different versions of the definition of DevOps. To some, it is a standard or a method. It is an integrated “culture” in the IT world. No matter how you choose to define DevOps, it is imperative to understand how to go about the DevOps journey to pick its benefits.

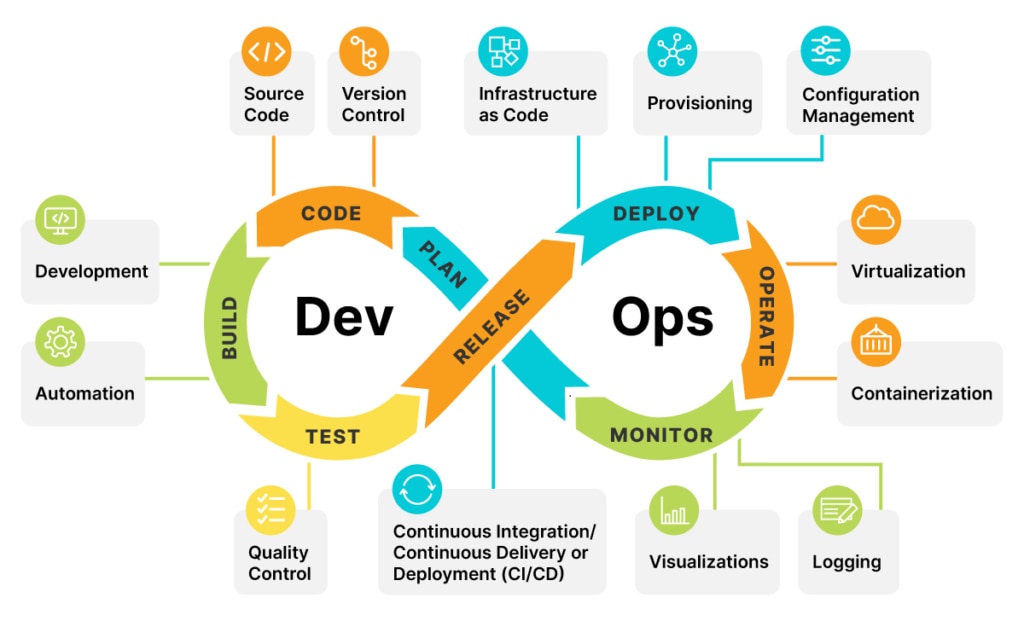
Devops is a combination of development and operation team (Dev + ops). It is a method that implements the agile model.

Devops describes a culture and set of processes that brings development and operation team together to complete the software development. It allows organization to create and improve dropouts at a faster pace (quicker) than they can with the traditional development approaches like water fall model, v-model So on.

It can be also defined as DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.

That means Devops that enhancing the communication between development and operations team.it brings both development team and operations team together into a single team.

DevOps, essentially as an approach or a work culture, it is implemented by right fusion (combination) of collaboration, automation, integration, continuous delivery, testing, and supervising.



The key terminology’s used in Devops are continues integration (CI), continues deployments (CD) and infrastructure as code (IAC).

1. Cd: (ansible,chef,puppet)

It is all about continuously deploying new version of software into test environment. Continuous deployment (CD) is a software development strategy that uses automation to automatically release code changes into production. It's an extension of continuous delivery (CD) and is associated with DevOps and Agile methodologies.

The purpose of continuous deployment is to accelerate development and reduce the time it takes for changes to reach end users.

CD uses a continuous integration (CI) system to build and test code changes, and then automatically deploy them to production once they pass all tests.

CD can help teams accelerate releases, develop faster, and respond to changing market demands. Some benefits of CD include: Expedited software delivery, Minimal downtime, End users benefit from new features more quickly, and Releases are less risky and easier to fix.

CD also allows teams to deploy small batches of changes, which can make releases less risky and easier to fix if a problem occurs.

2. CI: (Jenkins tool)

Over the past few years, Continuous Integration has become one of the best practices for software development. The goal is to detect the errors early on without having to wait until the end of the project.

Continuous Integration is a software development practice that integrates code into a shared repository frequently. This is done by developers several times a day each time they update the codebase. Each of these integrations can then be tested automatically.

One of the main benefits of integrating regularly and testing each integration is that you can detect errors more quickly and locate them easily. Since each integration or update to codebase is usually small, pinpointing the exact change that causes the error can be done quickly.

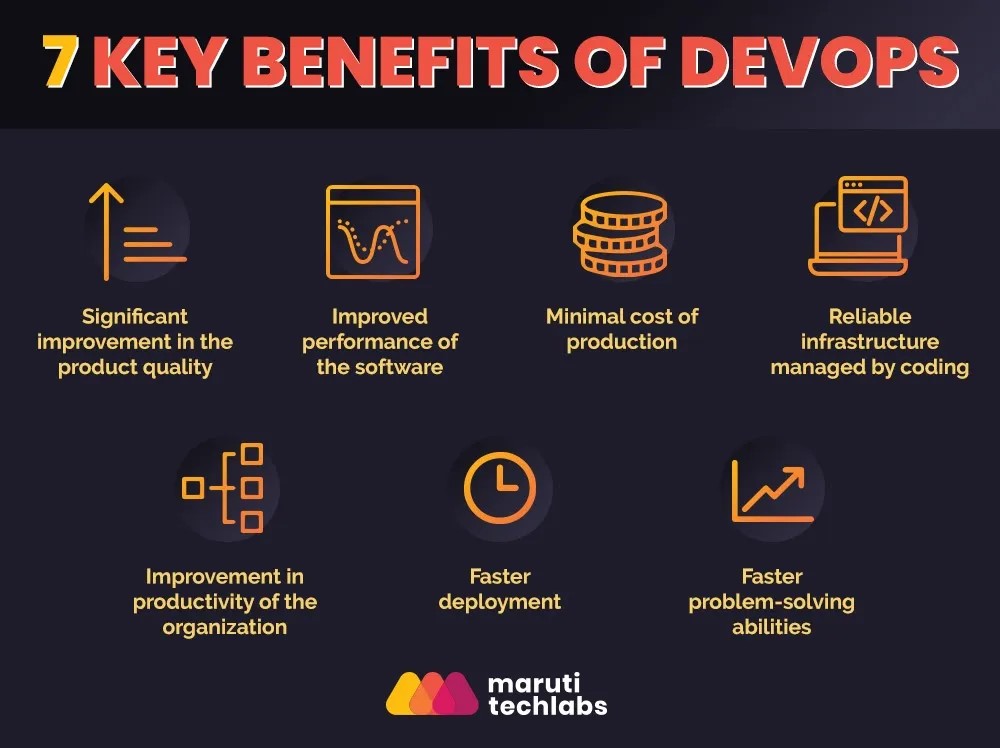
Continuous integration is a process where changes are merged into a central repository after which the code is automated and tested. The continuous integration process is a practice in software engineering used to merge developers' working copies several times a day into a shared mainline.

It refers to the process of automating the integration of code changes coming from several sources. The process comprises several automation tools that emphasize on the code’s correctness before Integration.

BENEFITS OF DEVOPS:

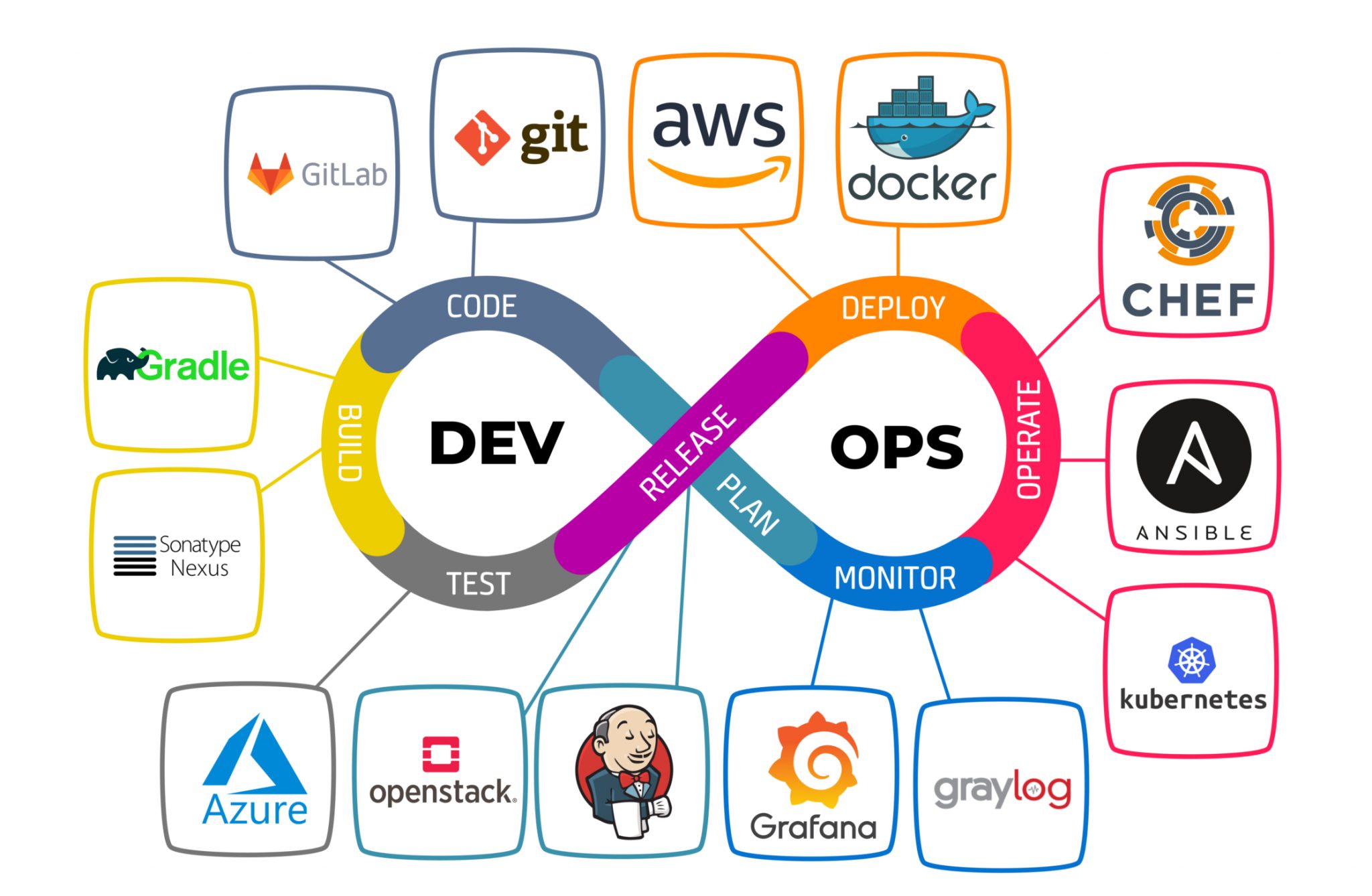
Here are some of the prime benefits a company can enjoy after adopting the DevOps way of working:

1. ensure faster deployment
2. stabilize work environment
3. continues delivery of software
4. faster and reliable problem solving techniques
5. minimal cost of production
6. Faster, better product delivery.
7. Faster issue resolution and reduced complexity.
8. Greater scalability and availability.
9. More stable operating environments.
10. Better resource utilization.
11. Greater automation.
12. Greater visibility into system outcomes.
13. Greater innovation.



The advantages of DevOps from an IT manager’s point of view. The benefits are:

* Lower volume of defects
* Lower cost of a release
* Improved software performance
* Lower cost of investment
* Frequent release of new features, fixes, and updates
* Improved MTTR (Mean Time To Recovery)



**What is the difference between the DevOps and Cloud engineer?**

**DevOps engineer:**

A DevOps engineer focuses on automation and streamlining of software development process, they handle CI/CD making sure everything runs smoothly from code to deployment.

**Cloud Engineer:**

A cloud engineer they design, build and manages the cloud infrastructure, they all about scalability, security and making everything running smoothly on a cloud.