I DevOps :-

It is a compound of development (Dev) and operations (Ops). DevOps is the union of people, process and technology to continually provide value to customers.

– development, IT operations quality engineering and security to coordinate and collaborate to produce better, more reliable products. By adopting a DevOps culture along with DevOps practices and tools, teams gain the ability to better respond to customer needs

DevOps allows a single team to handle entire application lifecycle from development to testing, deployment, and operations. DevOps helps to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrator

DevOps promotes collaboration between development and operations team to deploy code to production faster in an automated & repeatable way.

* DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, quality and speed of the application delivery has improved to great extent

DevOps architecture

Development and operations both play essentia roles in order to deliver applications.

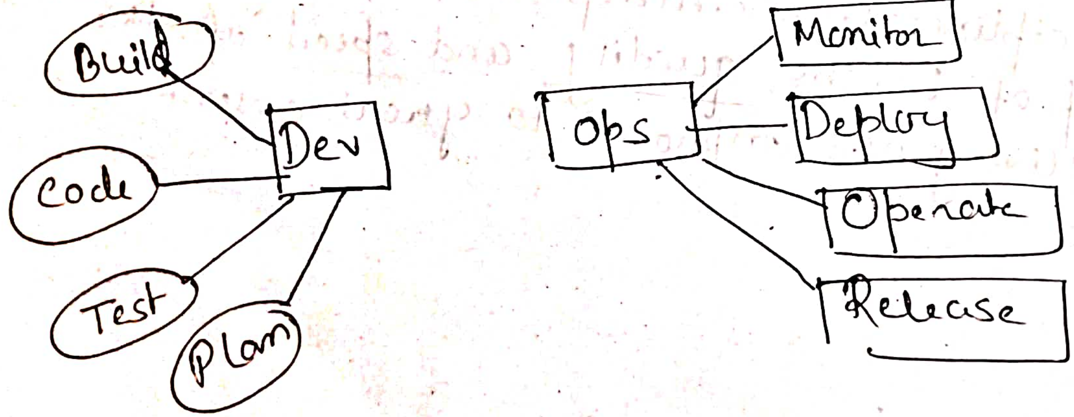
The development comprises analyzing the requirements, designing, developing, and testing of software components or frameworks.

The operation consists of administrative process, services, and support for the software.

When both development and operations are combin with collaborating, then the DevOps architecture is the solution to fix the gap between deployment and operation teams, therefore delivery can be faster.

* DevOps architecture is used for the application hosted on cloud platform and large distributed applicat

* If development and operations team works separately from each other, then it is time & consu to design, test and deploy. And if the teams are not in sync with each other, then it may cause a de in the delivery. So DevOps enables the teams to chang their shortcomings and increases productivity.



Build :- Without DevOps, the cost of consumption of resources was evaluated based on pre-defined individual usage with fixed hardware allocation.

With DevOps, using cloud, the resources are shared, build is dependent on user's need, which can control usage of resources.

(2) **Code** :- Git enables the codes to be used. The code can be appropriately arranged in files, folders and they can be reused

(3) **Test** :- The application will be ready for production after testing. In case of manual testing, it consumes more time in testing. and The testing can be automated which decreased time for testing.

(4) **Plan** :- As the operations and development teams are in sync, it helps in organizing the work to plan accordingly to increase productivity.

(5) **Monitor** :- Continuous monitoring is used to identify any risk of failure. Also, it helps in tracking the system accurately so that the health of the application can be checked.

(6) **Deploy** :- Many systems can support the scheduler for automated deployment.

(7) **Operate** :- DevOps changes the way traditional approach of developing and testing separately. The teams operate in a collaborative way where both the teams actively participate throughout the service lifecycle.

(8) Release

Deployment to an environment can be done by automation

DevOps lifecycle :-

The DevOps lifecycle includes seven phases

① Continuous development :-

This phase involves planning and coding of software. The vision of the project is decided during the planning phase.. The developers begin developing the code for the application. There are no DevOps tools that are required for planning, but there are several tools for maintaining the code.

② Continuous integration :-

Developers commit changes to the source code more frequently. This may be on a daily or weekly basis. It allows early detection of problems if they are present. Building code is not only involved compilation, but it also includes unit testing, integration testing, code review, and packaging. The updated code needs to be integrated continuously and smoothly with the systems to reflect changes to end users.

Jenkins is a popular tool used in this phase.

③ Continuous testing :-

In this phase, the developed software is continuously testing for bugs.

For constant testing, automation testing tools such as TestNG, JUnit, Selenium etc are used. These tool allow QAs to test multiple code-bases thoroughly in parallel to ensure that there is no flaw in the functionality. In this phase, Docker Containers can be used for simulating the test environment

Selenium does automation testing.

TestNG generates the reports.

The entire testing phase can automate with the help of a continuous integration tool called Jenkins

Automation testing saves a lot of time and effort for executing the tests instead of doing it manually

④ Continuous monitoring

Monitoring is a phase that involves all the operational factors of the entire DevOps process, where important information about the use of software is recorded and carefully processed to find out trends and identify problem areas.

⑤ Continuous feedback :-

The application development is consistently improved by analyzing the results from the operating of the software. This is carried out by placing the critical phase of constant feedback between the operations and development of next version of current software application.

⑥ Continuous deployment :-

In this phase, the code is deployed to the production server. It is essential to ensure that the code is correctly used on all the servers.

⑦ Continuous Operations :- All ~~the~~ DevOps operations are based on continuity with complete automation of the release process, Client → ABC Company Manager

① Requirement Analysis

Govt

1 year

Developer will develop, then testing team will test, then deploy, monitor

On-site (where client sits)

offshore (where the project is developed)

Why DevOp needs ?

hat client needs

- Fast delivery
- High Quality (customer satisfaction is also needed)
- less cost to operate the softwar
- Available (when customer wants ~~they~~ ~~the~~ the software should be open)

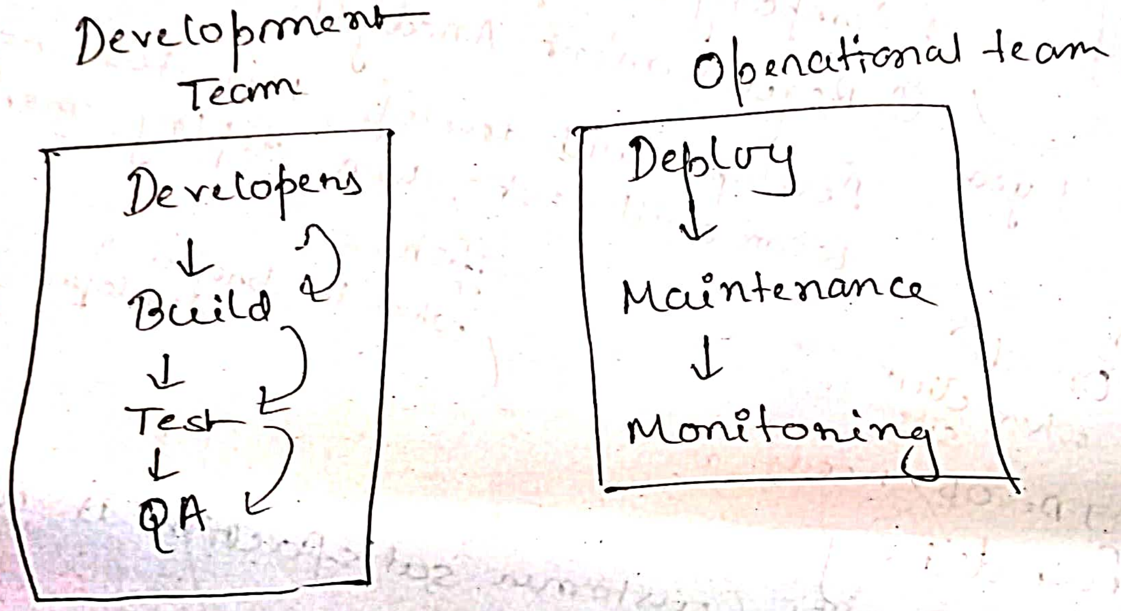
- Developer will write code for application
- Build :- Developer need to build the code (executable file will be generated)
- Test :- Test the executable file Testing team will do it {code is executed properly or not}
- Quality assurance :- After testing, it is done {whether it is fulfilling customer's requirements or not}
- Now it will deploy the application on on-site's Computer {whether the application is working in real time environment or not}

Maintenance, employee's training. How to solve the error

↓

Continuous monitoring after everything.

This total process is called <u>software development life cycle</u> (SDLC)



Without develop, after development, the software goes to operational team. So there are lot of conflicts. In the development
{ suppose the code is not properly working. In the development server, there may be some preconfigured software which is not present in operational team's server. Difference configuration, Different version of software }

Without devops, within the organization, there may be lot of problems. How one can deliver the software

Earlier method was waterfall method.

① Development team and operational team are not cooperative to each other

② Each team has to do everything manually (step by step)
{ After completing a task, we deliver it to next level

That's why we combine Development team and Operations team and made DevOPS.

After DevOps come, some jobs are ~~to~~ not required for deploy, earlier we needed server admin, whose job was to install the software, updation, server maintenance. But now we don't want these manual things. We want automation.

For automation, we use different tools which will work in different levels.

* When developer will write the code, he will
send the code in git. Jenkins will take this
tool ~~on~~ from git, and will send it to Build team.
Build team will send this to Maven. Now ~~B~~
suppose some error is there in Build team. They
will send this notification to Jenkins. Jenkins will
send this to developer team. He will update in
git

CI/CD = continuous integration / continuous delivery

DevOPs : Implementing automation at each and every stage

Devovs stages

- Version control :- Maintain different version of the code ⇒ git is used
- continuous integration :- compile, validate, code review, unit testing, integration testing ⇒ Jenkins
- Continuous delivery :- Deploying the build app to test server ⇒ Maven
- Continuous deployment :- Deploying the test app on the production server for release.

continuous monitoring