1. What is DevOps?

* The DevOps is a combination of two words, one is Software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to testing, deployment, and operations. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.
* DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated and repeatable way.
* DevOps helps to increase organization speed to deliver applications and services.
* DevOps is nothing but a practice or methodology of making “Developers” and “Operations”

1. Why DevOps?

* The operation and development team worked in complete isolation.
* After the design-build, the testing and deployment are performed respectively. That’s why they consumed more time that actual build cycles.

1. DevOps Architecture Features
2. Automation can reduce time consumption, especially during the testing and deployment phase. The productivity increases, and release are made quicker by automation. This will lead in catching bugs quickly so that it can be fixed easily.
3. Collaboration : The Development and Operations team collaborates as a DevOps team, which improves the cultural model as the teams become more productive with their productivity. The teams share their responsibilities and work closely in sync, which in turn makes the deployment to production faster.
4. Intigration: Application need to be integrated with other components in the environment. The integration phase is where the existing code is combined with new functionality and then tested. Continuous integration and testing enable continuous development.
5. Configuration Management: it ensures the application to interact with only those resources that concerned with the environment in which it runs. The configuration files are not created where the external configuration to the application is separated from the source code. The configuration file can be written during deployment, or then can be loaded at the run time, depending on the environment in which it is running.
6. DevOps Advantages and Disadvantages
7. Advantages :
8. DevOps is an excellent approach for quick development and deployment of applications.
9. It responds faster to the market changes to improve business growth
10. DevOps Components – Architecture
11. Build, Code, Test, Plan, Monitor, Deploy, Operate, Release
12. Build: Without DevOps, the cost of the consumption of the resources was evaluated based on the pre-defined individual usage with fixed hardware allocation. The usage of cloud, sharing of resources comes into the picture, and the build is dependent upon the user’s need, which is a mechanism to control the usage of resources or capacity.
13. Code: Many good practices such as Git enables the code to be used, which ensures writing the code for business, helps to track changes, getting notified about the reason behind the difference in the actual the expected output, and if necessary reverting to the original code developed. The code can be appropriately arranged in files, folders, etc.
14. Test: The application will be ready for production after testing. In the case of manual testing, it consumes more time to testing and moving the code to the output. The testing can be automated, which decreases the time for testing so that the time to deploy the code to production can be reduced as automating the running of the scripts will remove many manual steps.
15. Plan: DevOps use Agile methodology to plan the development. With the operations and development team in sync, it helps in organizing the work to plan accordingly to increase productivity.
16. 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.
17. Deploy: Many systems can support the scheduler for automated deployment. The cloud management platform enables users to capture accurate insights and view the optimization scenario, analytics on trends by the deployment of dashboards.
18. Operate: DevOps changes the way traditional approach of developing the testing separately. The teams operate in a collaborative way where both the teams actively participate throughout the service lifecycle. The operation team interacts with developers, and they come up with a monitoring plan which serves the IT and business requirements.
19. Release: Deployment to an environment can be done by automation. But when the deployment is made to the production environment, it is done by manual triggering. Many processes involved in release management commonly used to do the deployment in the production environment manually to lessen the impact on the customers
20. DevOps Lifecycle:
21. Continuous Development: This phase involves the planning and coding of the software. The vision of the project is decided during the planning phase. And 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.
22. Continuous Integration: This stage is the heart of the entire DevOps lifecycle. It is a software development practice in which the developers require to commit changes to the source code more frequently. This may be on a daily or weekly basis. Then every commit is built, and this 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.
23. Jenkins is a popular tool used in this phase. Whenever there is a change in the Git repository, then Jenkins fetches the updated code and prepares a build of that code, which is an executable file in the form of war or jar. Then this build is forwarded to the test server or the production server.
24. Continuous Testing: this is where the developed software is continuously testing for bugs. For constant testing, automation testing tools such as TestNG, Junit, Selenium, etc are used, these tools allow QA 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 simulating the test environment.
25. DevOps Principles:
26. End to end responsibility: DevOps team need to provide performance support until they become the end of life. It enhances the quality of the products engineered
27. Continuous Improvement: DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the growth of products or services offered.
28. Automate Everything: Automation is an essential principle of the DevOps process. This is for software development and also the entire infrastructure landscape.
29. Custom Centric Action: DevOps team must take customer-centric for that they should continuously and testing procedures.
30. Work as one team: in the DevOps culture role of the designers, developers, and testers are already defined. All they need to do is work as one team with complete collaboration
31. Ansible: Ansible is a leading DevOps tool. Ansible is an open-source IT engine that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools. It makes it easier for DevOps team to scale automation and speed up productivity.
32. Ansible is easy to deploy because it does not use any agents or custom security infrastructure on the client-side, and by pushing modules to the clients. These modules are executed locally on the client-side, and the output is pushed back to the Ansible server
33. Features
34. It is easy to use to open source deploy applications.
35. It helps in avoiding complexity in the software development process.
36. It eliminated repetitive tasks.
37. It manages complex deployments and speeds up the development process
38. Docker: is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.