Q #1) Explain DevOps.
Ans:

- DevOps promotes communication and collaboration between business, development & operations teams. In a way, we are talking about unification.
- DevOps is not only about tools but also about involving people and process.
- DevOps is inspired by the Agile process.
- DevOps is about automating the development, release and operation processes.
- DevOps helps in the speed of delivering applications to the end users.
- DevOps is also about continuous improvement which helps in learning through feedback.

Q #2) Explain a few prerequisites that are useful for DevOps implementation.
Ans: Typically the pre-requisites would include:

- Commitment at the senior level in the organization.
- Need for change to be communicated across the organization.
- Version control software.
- Automated tools for compliance to process.
- Automated Testing
- Automated Deployment

Q #3) What are the best practices for DevOps implementation?
Ans: DevOps implementation varies from one organization to other organization. Today organizations are looking to deliver the software faster and from that point of view here are my thoughts on how DevOps can bring about efficient communication with all the stakeholders including the operations team.

- Every organization has certain business goals and DevOps implementation should align with it. There should be a certain need for change.
- Encouraging communication and collaboration especially between development and operations.
- Automation is the key and should be done and carried out wherever possible within the SDLC stages to facilitate the working of DevOps.
- The tools used in DevOps should integrate very well. From the point of view of any commercial tools like IBM Rational CLM or Microsoft TFS, implementation of DevOps is good as it is from a single vendor. Similarly, one needs to look at open source tools which integrate to give the similar desired results.
- DevOps implementation is not successful without CI and CD practices. So continuous integration of your code with rigorous testing and doing continuous delivery is the key for it.
- It is very important for the Ops team to ensure that the applications are working very well at appropriate levels. If required they will need to work with the development

teams to build any tools that would help to build the right monitoring capabilities into the applications.

- Finally, encourage feedback from end users to enable continuous improvement which provides the key for improving the process and delivering quality software.

Q #4) What are the key components of DevOps?
Ans:

- Continuous Integration
- Continuous Testing
- Continuous Delivery
- Continuous Monitoring

Q #5) Explain Continuous Integration.
Ans: Continuous Integration is a very important component of the Agile process. Typically developers work on features or user stories within a sprint and commit their changes to the version control repository.
Once the code is committed, then the entire work of developers is well integrated and the build is performed on a regular basis based on every check-in or schedule. Hence, Continuous Integration as a practice forces the developer to integrate their changes with the others so as to get an early feedback.

Q #6) Explain Continuous Delivery.
Ans: Continuous Delivery is an extension of Continuous Integration which primarily helps to get the features which the developers are developing out to the end users as soon as possible. During this process, it goes through various stages of QA, Staging etc., and then for delivery to the PRODUCTION system.

Q #7) Explain Continuous Testing.
Ans: From the above goal of Continuous Integration which is to get the application out to end users are primarily enabling continuous delivery. This cannot be completed without a sufficient amount of unit testing and automation testing.
Hence, we need to validate that the code produced and integrated with all the developers who perform as required.

Q #8) Explain Continuous Monitoring.
Ans: As the application is developed and deployed, we do need to monitor its performance. **Monitoring** is also very important as it might help to uncover the defects which might not have been detected earlier.

Q #9) How will you approach when a project needs to implement devops?
Ans: Following approaches can be used but that will vary based on the actual project scenarios or from organization to organization

Stage 1: For any DevOps implementation an assessment of the existing process and implementation for around 2 to 3 weeks taking an average of 5 applications is a must to gauge the AS-IS process, identifies the improvement areas and provides a roadmap for the implementation.
Typically a senior profile like a DevOps architect should be involved to conduct this assessment.

Stage 2: A pilot POC can be done to showcase the end to end working of DevOps. Once this has been accepted and approved by the end users only then the actual implementation, handover and rolls out the plan for the projects involved should be carried out.
Any pieces of training required will also need to be imparted to the process or tools. DevOps architects, engineers, and testers would need to be involved at this stage.

Stage 3: The projects should now be in DevOps mode with all the components of continuous integration/delivery/testing and monitoring that is being followed.

Q #10) Can DevOps be applied to a Waterfall process? Explain the significance of the Agile process in DevOps implementation.
Ans: In the waterfall process, as all of us are aware initially complete Requirements are gathered, next the System is designed, Implementation of the System is then done followed by System testing and deployed to the end users. In this process, the problem was that there was a huge waiting time for build and deployment which made it very difficult to get the feedback.
The solution to the above problem was that the Agile process has to bring in agility in both development and operations. The agile process could be the principle or a certain pre-requisite may be required for DevOps implementation. DevOps goes hand in hand with the Agile process.

The focus area is to release the software in a very timely manner with shorter release cycles and quick feedback. So the agile process focus will mainly be on speed and in DevOps, it works well with the automation of various tools

Q #11) What is the difference between Continuous Delivery and Continuous Deployment?
Ans: In an Agile Sprint for **E.g.** there are many features or user stories that are developed, tested and ready for deploying. But based on customer scenarios and priorities not all would be deployed. So here in continuous delivery, it is very important to keep the code readily available for deployment.
In Continous Deployment, all the changes developed by the developer goes through various stages to be deployed to the PRODUCTION environment in an automated way.

Q #13) What are the top 10 devops tools that are used in the industry today?
Ans: The list goes as follows:

- Jira
- GIT/SVN
- Bitbucket
- Jenkins

- Bamboo
- SonarQube
- Artifactory/Nexus
- Docker
- Chef / Puppet /Ansible
- IBM Urbancode Deploy / CA-RA
- Nagios / Splunk

Q #14) Can you explain the uses of the tools just mentioned and how they connect to give a DevOps model (CI/CD)?

Ans:

Planning

Jira – Used for Project Planning and Issue management

Continuous Integration

Git – Version Control

Jenkins – Open Source Continuous Integration tool which can also help in Continuous Delivery.

SonarQube – Code analysis

JFrog Artifactory – Binary Repository Manager

Continuous Delivery

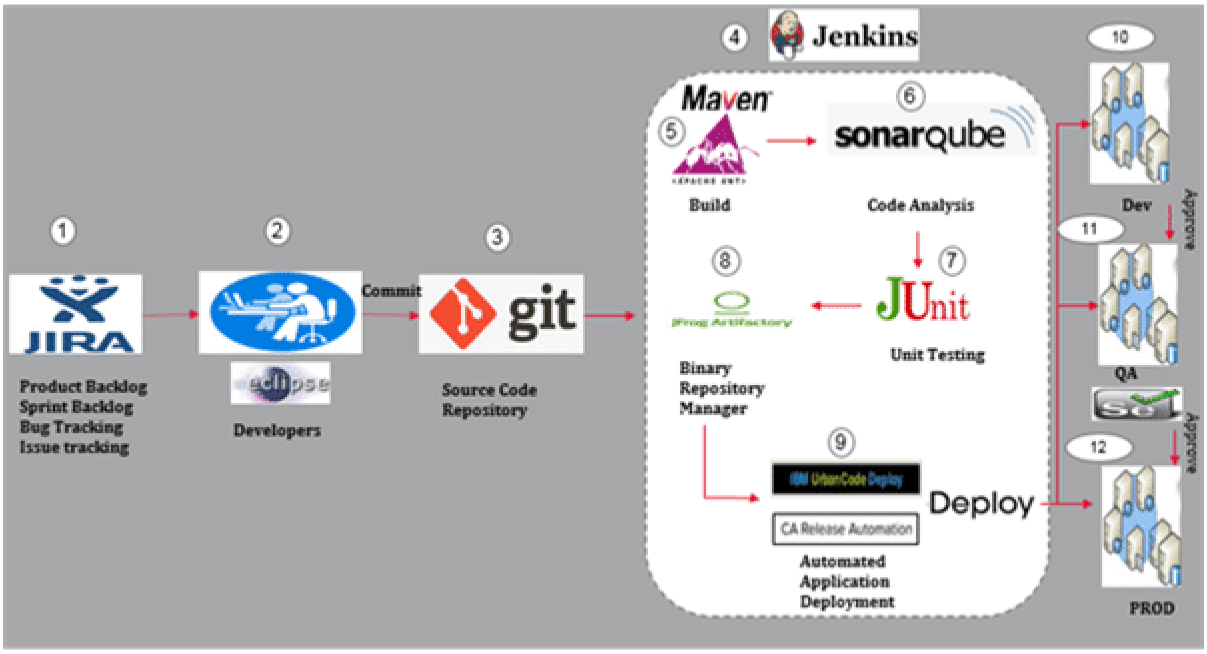
Chef / Puppet / Ansible – Configuration Management and Application Deployment

IBM Urbancode Deploy / CA RA – Continuous Delivery

Continuous Monitoring

Nagios / Splunk

Sample DevOps workflow:



- Typically in an Agile process User Stories, Tasks, Defects etc., are all stored in JIRA and assigned to the Product Owners and Developers.
- Developers pick up the tasks assigned to them and work on the development. The source code is version controlled and stored in GIT. The developers commit their

changes to the source code in GIT. Eventually, the code is shared among the developers using GitHub.

- Jenkins which is the Continuous Integration tool pulls the code and on every check-in or based on a schedule the build is done using build tools like Maven or ANT.
- As the J2EE WAR files are produced, they are also version controlled and stored in a binary repository manager like Artifactory or Nexus.
- Unit Testing using JUnit and Code Analysis with SonarQube is also done and automated
- Once the above process is completed the Continuous Delivery is performed to different environments based on approvals using tools like IBM UrbanCode Deploy / CA RA Continuous Testing (Functional and Acceptance Testing) is invoked in the appropriate test environments using tools like Selenium
- Continuous Monitoring would be an ongoing activity in the PROD environment

Q #15) What is configuration management in terms of infrastructure and mention a few popular tools used?
Ans: Configuration management consists of practices and the various tools involved to automate the delivery and infrastructure operations. It is all about keeping the server ready (**E.g.** Installing system packages, network configuration settings) for application deployment once the application is developed.
So the Ops or the system admin needs to ensure parity in different environments (Dev, QA, PROD etc…) by provisioning the systems.

Tools Used in this space to automate the tasks of configuration management described above are **Chef / Puppet / Ansible**

Q #16) Which scripting tools are used in DevOps?
Ans: Python, Ruby

Q #17) Explain the typical roles involved in DevOps.
Ans:

- DevOps Architect – The leader who is responsible for the entire DevOps process.
- DevOps Engineer – The person should be experienced with Agile, SCM or Version Control, CI /CD and setting up automation tools for the same, Infrastructure automation and Database management skills. Any developer who has skills in coding or scripting and has the acumen to get into deployment or system admin can qualify for the role of a DevOps engineer.

Q #18) Have you been involved in DevOps implementation in the cloud? If yes which cloud computing platform?
Ans: **Examples of popular Cloud Computing Platforms:**

1. AWS – Amazon Web Services
2. Microsoft Azure
3. Google Cloud

Q #19) Explain some of the metrics that were followed for DevOps success.

Ans:

Some of the examples that could be mentioned are as follows:

- The first and most important factor is the speed of delivery which means time taken for any work item to get into the production environment.
- Next would be the deployment and how much time it would take once this process is automated.
- It is almost necessary to track how many defects are found in different environments with respect to the PRODUCTION environment. This is very important in considering the features that need to be released faster. Use of Agile methodologies helps a lot and the prime goal is to reduce PRODUCTION level defects.
- Normally deployments do not fail but it is very important to keep a track on this aspect and have a mechanism to roll back to the previous stable version.
- In any DevOps implementation, unit testing is the key as well as functional testing. Based on the code changes done, often we need to look at whether these test break and to what extent. It is imperative that the automated test is robust enough to sustain any code changes.
- It is very important to measure the actual or the average time that it takes to recover in case of a failure in the PRODUCTION environment. This is termed as Mean Time To Recover (MTTR) and it should be short. This also means that one needs to have proper monitoring tools to keep the recovery time short.
- Performance of the application is another key metric which should be monitored especially after any deployments are done.
- A very important factor for success is the number of bugs being reported by the customers which primarily depends on the quality of the application.