Devops Unit Wise Important Questions

Unit 01

Short Questions

1. Explain in detail about Agile Development Model with a neat diagram?
2. DevOps and ITIL are not mutually exclusive? Justify.
3. Explain in detail about DevOps Continuous Delivery Pipeline?
4. List the different possible cases for Bottlenecks in CI/CD?
5. With an example explain DevOps Process?

Long Questions

1. Compare Agile and DevOps, and explain their complementary nature in achieving efficient software development and delivery.
2. Describe release management in DevOps, its challenges, benefits, and real-world examples of successful implementation.
3. Compare and contrast Scrum and Kanban as Agile methodologies, their support for DevOps, and contribution to software delivery.
4. Analyze the relationship between DevOps and ITIL, and how to effectively incorporate ITIL practices within a DevOps culture.
5. Explain the concept of a delivery pipeline in DevOps, its stages, and popular tools/technologies used.
6. Explore the role of automation in DevOps, its benefits, challenges, and examples of popular automation tools.
7. Identify common bottlenecks in DevOps, strategies to overcome them, and real- world examples of resolved bottlenecks.
8. Explain the significance of Continuous Delivery in DevOps and provide examples of organizations successfully implementing it.
9. Discuss the principles and practices of DevOps that improve collaboration and efficiency in IT operations.
10. Investigate the importance of monitoring and feedback loops in DevOps, and how organizations can leverage them for continuous improvement, with examples.

Unit 02

Short Questions

1. Explain DevOps Life-cycle for Business Agility?
2. Discuss about Continuous Testing in DevOps?

3. Explain in detail about Monolithic Architecture with a neat diagram?
4. Explain in detail about Microservices Architecture with a neat diagram?
5. How to handle Database Migrations?
6. Discuss about Resilience in DevOps?

Long Questions

1. Compare monolithic and microservices architectures in the context of DevOps, discussing their advantages and disadvantages and providing real-world examples.
2. Explore continuous testing in DevOps, its contribution to software quality, and the challenges and benefits of implementing it.
3. Explain the influence of DevOps on software architecture, focusing on factors like modularity, separation of concerns, and database migrations.
4. Discuss the impact of DevOps on achieving business agility and provide examples of companies that have adopted DevOps for faster software delivery and increased customer responsiveness.
5. Discuss how DevOps ensures the resilience and robustness of software systems and provide examples of organizations using DevOps for building resilient architectures.
6. Explain the concept of separation of concerns in software architecture and provide real- world examples of its implementation.
7. Explore the challenges and best practices for handling database migrations in DevOps and discuss available tools.
8. Investigate the relationship between software architecture and DevOps, focusing on how a well-designed architecture supports DevOps principles.
9. Discuss the impact of DevOps on software quality and reliability, providing examples of improvements achieved through DevOps practices.
10. Analyze the challenges and benefits of implementing microservices in the data tier, considering the alignment with DevOps principles and implications for data management, scalability, and maintenance.

Unit 03

Short Questions

1. What is the need of Source Code Control?
2. In DevOps, how Source code management is useful for different Roles?
3. Explain about Migrations of different Source Code Management Systems?
4. Write a short notes on Shared Authentication?
5. Explain in detail about Hosted Git Servers?
6. Write short notes on Docker Intermission?
7. Discuss about the Pull Request Model?
8. Explain in detail about GitLab?

Long Questions

1. Explain the significance of source code control in project management, its history, and role in version control and collaboration.
2. Discuss the roles of developers, testers, and release managers in source code management and how collaboration among them leads to project success.
3. Explore source code management systems, their importance, key features, and how they enable efficient code management and version control.
4. Investigate challenges and best practices for source code migrations, with examples of successful strategies.
5. Discuss shared authentication for accessing source code repositories, its implementation, benefits, and associated risks.
6. Compare hosted Git servers like GitHub, GitLab, and Bitbucket, discussing their features, advantages, and limitations.
7. Explore different Git server implementations (self-hosted, cloud-based, enterprise solutions), factors to consider when choosing one.
8. Analyze Docker's role in source code management, its benefits for creating, packaging, and deploying software applications.
9. Discuss Gerrit as a code review and collaboration tool, its features, and successful implementation examples.
10. Investigate GitLab as a comprehensive DevOps platform, its capabilities for source code management, CI/CD, issue tracking, and project management.

Unit 04

Short Questions

1. List out the different Build Systems available today?
2. How to use Jenkins Build Server to create builds?
3. Discuss about Build Slaves?
4. Write short notes on Triggers?
5. Explain in detail about Job Chaining and Build Pipelines?
6. How to create builds by Dependency Order?

Long Questions

1. Discuss the role of build systems in DevOps, their key components, and how they automate the software build process. Provide examples of popular build systems.
2. Explore the features and capabilities of the Jenkins build server, its role in continuous integration and delivery, and the benefits and challenges of using Jenkins in DevOps.

3. Explain the importance of managing build dependencies in software development, common challenges, and effective strategies using build automation tools.
4. Discuss the significance of Jenkins plugins in extending its functionality for tasks like code analysis, testing, and deployment. Provide examples of popular Jenkins plugins.
5. Analyze the importance of file system layout in build server configurations, its impact on the build process and artifact management, and best practices for designing an efficient layout.
6. Explain the concept of build slaves in Jenkins, their role in distributed build execution, scalability, and performance improvement. Discuss strategies for configuring and managing build slaves effectively.
7. Investigate triggers in build automation, the types available in Jenkins, and how they initiate the build process based on various scenarios. 8. Explore job chaining and build pipelines in Jenkins, their role in automating complex build processes and deployment workflows, and the benefits of using them. Provide examples of successful implementations.
9. Discuss infrastructure as code (IaC) in the context of build servers, its facilitation of provisioning, configuration, and management. Explain the advantages of using IaC tools for build server infrastructure.
10. Compare alternative build servers like Bamboo, TeamCity, and CircleCI with Jenkins, discussing their features, advantages, limitations, and recommendations for choosing the appropriate one.
11. Discuss the importance of collecting and analyzing quality metrics during the build process, such as code coverage, static code analysis, and test results. Explain how integrating quality measures enhances software quality and continuous improvement in DevOps.

Unit 05

Short Questions

1. What are the Pros and Cons of Automated Testing?
2. Write short notes on Selenium? List out the features of it?
3. Discuss about JavaScript Testing?
4. Differentiate Test-driven development from REPL-driven development?
5. Write short notes on: i) Puppet ii) Chef iii) Ansible iv) SaltStack

Long Questions

1. Discuss the different types of testing in DevOps, their significance, and contributions to software quality. Provide examples of testing techniques and frameworks used.
2. Explore the benefits and challenges of test automation in software development, its
impact on efficiency and accuracy, and best practices for implementing it in DevOps.
3. Explain the features and capabilities of Selenium as a popular testing tool, including web application testing. Discuss its advantages and limitations in a DevOps context.
4. Discuss the challenges and approaches for testing backend integration points in software applications. Provide examples of tools used in testing backend integrations in DevOps.

5.Explore test-driven development (TDD) and its role in ensuring code quality and test coverage. Discuss the principles, benefits, and challenges of implementing TDD in DevOps.

6. Discuss REPL-driven development and its benefits for iterative testing and rapid code prototyping. Explain how it aligns with DevOps principles and facilitates faster feedback loops.

7. Explore deployment systems and strategies in DevOps, including continuous deployment and delivery. Provide examples of popular deployment systems.

8. Discuss the role of virtualization stacks, including hypervisors and containerization platforms like Docker, in efficient and scalable application deployment. Explain their alignment with DevOps principles.

9. Explain the concept of client-side code execution in application deployment, its benefits, and challenges. Discuss considerations for deploying client-side code in a DevOps environment.

10. Compare and contrast deployment tools like Puppet, Ansible, Chef, Salt Stack, and Docker.

11. Discuss their features, benefits, and use cases in automating deployment and infrastructure management in DevOps.