Day 43

DIY

DIY Questions:1.An organization wants to increase its competitive efficiency. The management team decides to release features on a daily or hourly basis. They also want to ensure cross-department coordination. Which of the following steps of CI/CD pipelines will the organization adapt in this scenario?

a)Continuous integration b)Continuous delivery c)Continuous deployment d)Continuous development

Ans:

a)Continuous integration

c)Continuous deployment

2.If Alice is required to integrate the Continuous Integration workflow in her software development process, then which of the following sequences of steps should she follow in order to complete this action:1.The Continuous Integration server detects changes in the version control repository, so the Continuous Integration server retrieves the latest copy of the code from the repository and then executes a build script, which integrates the software.2.The developer commits the code to the version control repository. Meanwhile, the Continuous Integration server on the integration build machine polls the source code repository for changes.3.The Continuous Integration server generates feedback of the build results to the specified project members.4.The Continuous Integration server continues to poll for changes in the version control repository, and the process repeats.5.Unit tests are then carried out if the build of that project passes. If the tests are successful, the code is ready to be deployed to either the staging or production server.

a)2 -> 3 -> 4 -> 1 -> 5 b)3 -> 2 -> 1 -> 5 -> 4 c)3 -> 1 -> 5 -> 2 -> 4 d)2 -> 1 -> 3 -> 5 -> 4

Ans:d)2 -> 1 -> 3 -> 5 -> 4

3.Which of the following statements represent the advantages of using Continuous Integration (CI) and Continuous Deployment (CD): a)Reduces overhead across the development and deployment process.b)Reduces the time and effort for integrations of different code changes c)Enables a quick feedback mechanism on every change

Ans:

4.Which of the following steps of CI/CD pipelines mainly focus on:•Instant responses to bugs•Reliable and stable releases a)Continuous Integration b)Continuous Delivery c)Continuous Deployment d)All of these

Ans:All of these

5.In the Software Development Life Cycle (SDLC), testing is a phase where developers check whether the software satisfies the specific requirements and expectations of the customer by testing for bugs and errors. Testing tools help testers find issues in their products before the users do, resulting in better quality software. You are using CI/CD pipelines in your software development process.

a) Wercker b) Terraform c) Google Chrome Developer d) Go continuous delivery

Ans:

6.What are the benefits of using version control?

Ans:

Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time. As development environments have accelerated, version control systems help software teams work faster and smarter. They are especially useful for DevOps teams since they help them to reduce development time and increase successful deployments.

Benifits :

1. Collaboration: Enables seamless collaboration among team members.

2. History and Auditing: Keeps a complete history of code changes for accountability and issue tracking.

3. Code Backup and Recovery: Provides reliable backup and recovery of code.

4. Code Branching and Forking: Supports simultaneous work on different versions of the codebase.

5. Code Revert and Rollback: Allows easy rollback to previous versions in case of mistakes or bugs.

6. Traceability and Bug Tracking: Helps track down the origin of bugs or issues in the code.

7. Continuous Integration and Deployment: Integrates with CI/CD pipelines for automated testing and deployment.

These benefits enhance collaboration, accountability, code management, stability, and efficiency in software development.

7.Which are the top DevOps tools?

Ans:

There are numerous DevOps tools available, each serving different purposes within the software development and delivery process. Here are some popular and widely used DevOps tools:

1. Jenkins: An open-source automation server for continuous integration and continuous delivery (CI/CD) pipelines.

2. Git: A distributed version control system that enables collaboration and code management.

3. Docker: A containerization platform that allows applications to be packaged into containers for easy deployment and scalability.

4. Ansible: An open-source automation tool that automates infrastructure provisioning, configuration management, and application deployment.

5. Kubernetes: An open-source container orchestration platform for automating the deployment, scaling, and management of containerized applications.

6. Terraform: An infrastructure as code tool that enables the provisioning and management of infrastructure resources across various cloud providers.

7. Puppet: An infrastructure automation tool that provides configuration management and automation capabilities.

8. Chef: A configuration management tool that allows for infrastructure automation and the management of server configurations.

9. Splunk: A log management and analysis tool that helps monitor and troubleshoot applications and infrastructure.

10. Grafana: A data visualization and monitoring tool used to monitor and analyze metrics and logs.

It's important to note that the choice of DevOps tools may vary depending on specific requirements, project needs, and personal preferences. Organizations often use a combination of these tools to establish efficient and streamlined DevOps practices.