Section 1: Selling CI/CD to your Team/Organization

Explain the fundamentals and benefits of CI/CD to achieve, build, and deploy automation for cloud-based software products.

Fundamentals of CI

Continuous Integration is the practice of integrating all your code changes into the main branch of a shared source code repository early and often, automatically testing each change when you commit or merge them, and automatically kicking off a build. With continuous integration, errors and security issues can be identified and fixed more easily, and much earlier in the development process.

Fundamentals of CD

Continuous Deployment (CD) is the practice of automatically deploying code changes to production environments after they have been successfully built and tested. CD eliminates the need for manual intervention in the deployment process, reducing the risk of human error and enabling faster and more frequent releases.

Benefit of CD/CI

1

Reduce Costs and Boost Profits: CI/CD is also good for the bottom line. It standardizes deployment processes across all projects, and, done right, it enables teams to systematically test every change made to the source code.

As a result, this process stands to dramatically reduce the likelihood that any bugs or errors slip through the cracks and cause problems down the line. Done right, this practice can lower development costs by eliminating many of the costs incurred while building and testing code changes.

And because CI/CD also makes it easier to deliver high-quality products to market faster and respond to feedback as it comes in, organizations stand to see an increase in profits. Customers stick around longer and will likely recommend your products to others in their network.

2

Faster Time to Market: CI/CD enables faster and more frequent releases, allowing you to deliver new features, bug fixes, and improvements to your users more quickly. By automating the build, test, and deployment processes, you can significantly reduce the time it takes to go from code changes to production deployment.

3

Reduced Risk: Manual deployments are prone to human error, which can lead to issues and downtime in production environments. CI/CD automates the deployment process, reducing the risk of errors and ensuring consistency across deployments.

4

Increased Collaboration: CI/CD encourages collaboration among developers by providing a shared repository where code changes are integrated and tested. This promotes better communication, code reviews, and knowledge sharing among team members. It also helps identify and resolve conflicts or issues early on, leading to smoother development workflows.

5

Scalability and Flexibility: CI/CD pipelines can be easily scaled and adapted to handle different types of projects and environments. Whether you are working on a small application or a complex distributed system, CI/CD allows you to automate the deployment process and easily scale up or down as needed.