1. integration (when developers integrate code from multiple sources)

Problem: Code integration presents compatibility issues and inconsistent configurations. This can result in broken or unexpected code, causing prolonged development time and decreased efficiency.

Solution: Docker can help address these issues by providing a standardized and isolated environment for developers to work in. Docker containers allow developers to package their applications, dependencies, and configurations into a single, self-contained unit that can be run consistently across different environments.

1. Testing

Problem: Software testing can be a complex and time-consuming task, especially when testing multiple configurations and environments. Ensuring that software is tested the same in all environments can be challenging, leading to inconsistency and bugs.

Solution: Docker can help address these issues by providing a standardized environment for testing. Docker containers can be used to create identical testing environments for each configuration.

1. Deployment

Problem: It can be difficult to ensure that the software is deployed consistently across different environments, resulting in compatibility issues and mistakes.

Solution: Using containers, we can wrap software, its needed components, and settings into one package. This helps make sure it can be deployed the same way in different surroundings.

1. customer support

Problem: Providing software support to customers can be challenging, especially when managing varying configurations and environments. Reproducing customer issues in the development setting can be difficult, causing longer resolution times and lowering customer satisfaction.

Solution: Docker can help address these issues by providing a consistent environment for customer support. Docker containers can be used to create identical environments for each customer configuration, enabling developers to easily reproduce customer issues in their development environment.

5. internal and external hackers who might attack the system

Problem: Insecure configurations and environments can lead to vulnerabilities that can be exploited by attackers, resulting in data breaches and other security incidents.

Solution: Docker containers lay out a safer environment, protecting the host and other containers from attack. Docker images undergo security scans, preventing exploitation of known vulnerabilities.

Cite: <https://www.docker.com/>

<https://cloudacademy.com/blog/what-is-docker/>

<https://docs.docker.com/build/ci/>

<https://www.suse.com/c/rancher_blog/top-5-challenges-with-deploying-docker-containers-in-production/>

<https://www.sitepoint.com/what-is-docker/>

<https://apiumhub.com/tech-blog-barcelona/top-benefits-using-docker/>

<https://www.section.io/engineering-education/best-practices-to-secure-a-docker-container/>

<https://beaglesecurity.com/blog/article/docker-container-security.html>