Prateek Singh Chauhan
CWID: 20016291
Stevens Institute of Technology
SSW 567 WS

HW 02: Docker and Configuration Management

Problem:

Integration: Integration for commit was done manually, as hosted in single Operating system, during each integration all configurations and dependencies needs to check for every application compatibility in the same server.

Tester: Had to configure and install all dependencies and applications to test single module.

Deployment: In Traditional Solutions, organizations ran applications on physical servers and manually installed or used existing OS, tools, dependencies required for application. This in-premises physical servers provided isolation issues as there was no resource boundaries for an application. Also, it created Scaling issue as resources were under-utilized and was expensive to maintain or repeat for organizations. For multiple developer working in single feature

Customer support: Maintenance and Resolution of production incidents or customer incidents involved more time due to traditional integration, testing and deployment of application.

Internal and external hackers who might attack the system: Every individual involved has access to configurations and code causing security risk.

Solution by Docker:

Integration: Each application has its separate container and own set of dependencies and libraries providing process level isolation i.e., no dependency on other applications which facilitates Continuous Integration.

Testing: QA team only needs to run container and replicate the developer environment to test that specific application and need not install all the dependencies and code, saving time and energy, also provides consistency of environment with all individuals involved int eh process.

Deployment: Containerization provided allowed multiple instances to run in a single system, and single image can be considered as one unit and gave capability to host multiple applications containers with different configurations to be hosted in single servers. Also, resolving scalability and updating issue for an application by providing image versioning while being lighter than VMs and no hypervisors while increasing system performance. Easing the deployment capability.

Customer support: Docker provides fast integration, fast deployment, scalability, faster migrations, and ease in maintaining which provides faster resolution of production incidents or customer incidents.

Internal and external hackers who might attack the system: better security, less access needed to work with the code running inside containers, and fewer software dependencies. Also, to solve the security issue, Docker Content Trust enables IT ops team to sign images and ensure that only signed binary will run in production. Using multiple Docker Trusted Registries enables to build a progressive trust workflow for their applications development process.

Reference:

- https://www.linkedin.com/pulse/traditional-deployment-vs-virtualization-container-shazly-abozeid/
- https://www.edureka.co/blog/docker-tutorial?utm_source=youtube&utm_medium=description&utm_campaign=docker-tutorial-for-beginners-061216
- https://dzone.com/articles/top-10-benefits-of-using-docker
- https://www.simplilearn.com/tutorials/docker-tutorial/what-is-docker-container