

개발 환경 에서 Docker 활용

2020.06.11

Developing with Docker

Developing apps today requires so much more than writing code. Multiple languages, frameworks, architectures, and discontinuous interfaces between tools for each lifecycle stage creates enormous complexity. Docker simplifies and accelerates your workflow, while giving developers the freedom to innovate with their choice of tools, application stacks, and deployment environments for each project.

Developing with Docker

오늘날 앱을 개발하려면 코드 작성보다 훨씬 많은 것이 필요합니다. 각 라이프 사이클 단계에 대한 도구 간의 여러 언어, 프레임 워크, 아키텍처 및 불연속 인터페이스는 매우 복잡합니다. **Docker**는 워크 플로를 단순화하고 가속화하는 동시에 개발자가 각 프로젝트에 대해 선택한 도구, 응용 프로그램 스택 및 배포 환경을 자유롭게 혁신 할 수 있도록합니다.

그래서 결국 장점이 뭘까?

- 환경 격리, Virtual Machine, Python Virtual Env, java jenv, direnv ...
- 환경구축이 쉽다. (개발 환경 셋팅이 반)
- “나는 되는데 왜 여기서 안되지?”를 예방 할수 있다.
- 잘만 하면 개발 환경을 그대로 운영으로 쓸수도 있다. (?)

장점만 있을까?

- 또 배워야 해요...
- 잘 쓰려면 Docker만으로 안된다. Docker-compose, Docker-swarm, Kubernetes
- 레퍼런스가 있다곤 하지만 부족하다. 삽질의 가능성이 큼

샘플

결론

Docker, Git, Unit Test, CI-CD는
개발자를 괴롭히는 기술이 아닙니다.

운영에 적용은 당장 어렵겠지만
개발부터 조금씩 적용하면 됩니다.

혼자는 어렵겠지만 같이하면
즐겁지 않을까요?

감사합니다.