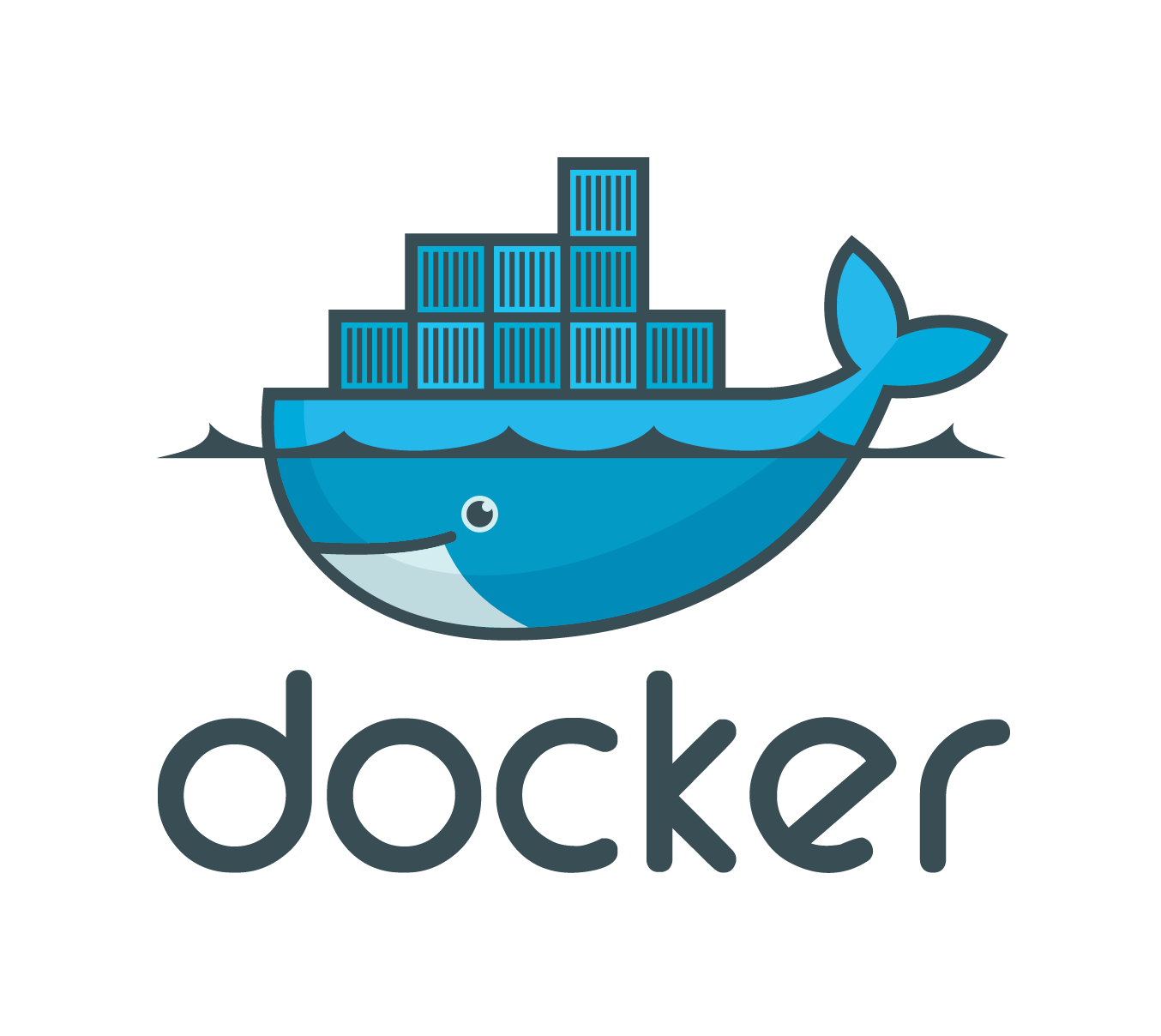
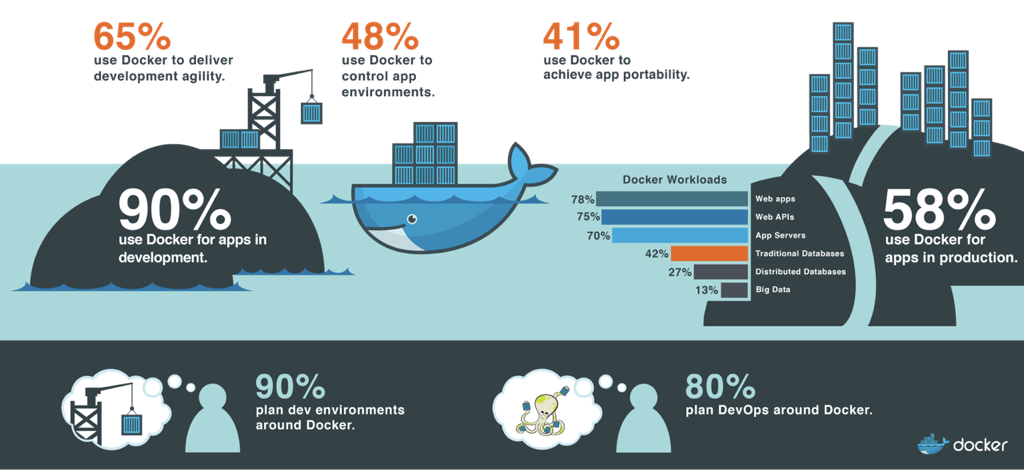
[: Docker :]



*Docker Logo : 컨테이너들을 싣고 나르는 선박을 위트있게 표현하였다.*

* **서론**

1. 컨테이너는 물류 분야에서 이용되면서 인류가 큰 발전을 할 수 있도록 기여하였다. 이 개념이 IT 분야에 적용되었고, 서버 관리 측면에서 혁신적인 패러다임으로 전 세계에 이름을 알렸으며 현재 놀라운 일들을 이룩하고 있는 것이 ‘**Docker’** 이다.
2. DevOps의 등장으로 개발주기가 짧아지면서 잦은 서비스 배포가 이루어지고 마이크로 서비스 아키텍처가 유행하면서 응용 프로그램을 이루는 기능들이 각각 독립적이지만 상호 교류 가능한 모듈로 쪼개어지면서 수많은 모듈들이 도출되었고 이것들을 관리하기가 복잡해졌다. 게다가 새로운 툴이 계속 개발되어 출시되고 Cloud의 발전으로 설치해야 할 서버가 수백, 수천 가지에 이르는 답이 없는 상황 속에서 **‘Docker’** 가 등장하였고 서버관리 방식이 완전히 바뀌게 된다.
3. 2016년 도커에서 진행한 설문조사에 의하면,

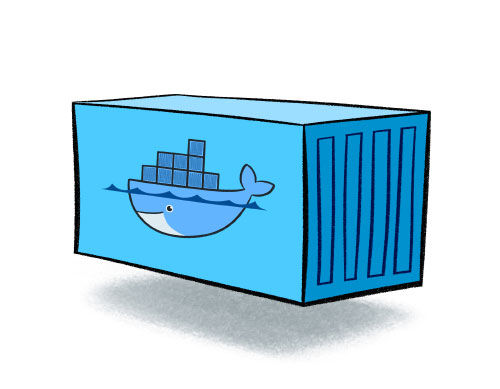


90%가 개발에 사용 중이고 80%가 DevOps에 사용할 예정이며 58%가 운영환경에서 사용 중 이라고 한다.

* **본론**

1. 본격적으로 **‘Docker’** 란 무엇인지 알아보자.

**Docker** 는 **컨테이너 기반의 오픈소스 가상화 플랫폼**이다. 본래 컨테이너는 규격화된 공간을 가지며, 안에 각종 화물들을 싣고 다양한 운송수단에 의해 옮겨진다. 서버에서 이야기하는 컨테이너도 이와 비슷하다. 다양한 프로그램, 실행환경을 컨테이너로 추상화하고 동일한 인터페이스를 제공하여 프로그램의 배포 및 관리를 단순하게 해준다.

.

*docker container*

1. **컨테이너는 격리된 공간에서 프로세스가 동작하는 기술**이다.

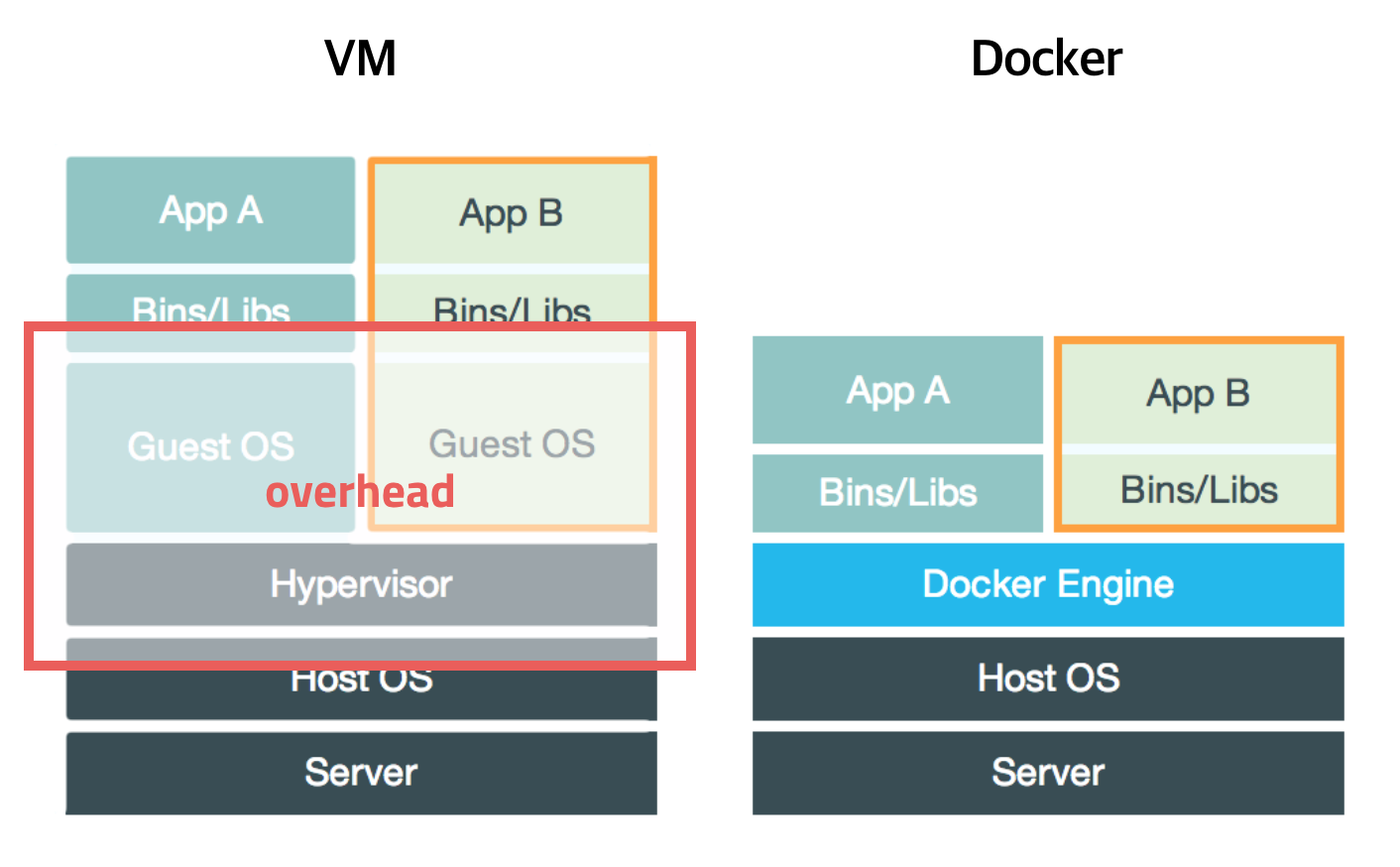
가상화 기술의 하나지만 기존 방식과 차이가 있다.

**[기존 방식 1 : 전가상화 방식 – 호스트 가상화 방식]**

기존의 가상화 방식은 주로 OS 를 가상화한 것이다. 기존의 가상머신(VMware, VirtualBox)은 호스트 OS 위에 게스트 OS 전체를 가상화하여 사용하는 방식이다. 이 방식은 여러 가지 OS를 가상화할 수 있고 비교적 사용법이 간단하지만 무겁고 느려서 운영환경에서는 사용할 수 없다.

**[기존 방식 2 : 반가상화 방식 – CPU 가상화 방식]**

이러한 상황을 개선하기 위해 CPU의 가상화 기술(HVM)을 이용한 KVMKernel-based Virtual Machine과 반가상화 Paravirtualization 방식의 Xen이 등장했다. 이러한 방식은 게스트 OS가 필요하긴 하지만 전체 OS 를 가상화하는 방식이 아니기 때문에 호스트 가상화 방식에 비해 성능이 향상되었다. 이러한 기술들은 OpenStack 이나 AWS, Rackspace 같은 클라우드 서비스에서 가상 컴퓨팅 기술의 기반이 되었다.



*Virtual Machine and Docker*

**[새로운 방식 : 프로세스 격리 방식]**

전가상화든 반가상화든 추가적인 OS를 설치하여 가상화하는 방법은 어찌됐든 성능문제가 있었고 이를 개선하기 위해 프로세스를 격리하는 방식이 등장했다. 리눅스에서 이 방식을 리눅스 컨테이너라고 하고 단순히 프로세스를 격리시키기 때문에 가볍고 빠르게 동작한다. CPU나 메모리는 딱 프로세스가 필요한 만큼만 추가로 사용하고 성능적으로도 거의 손실이 없다.