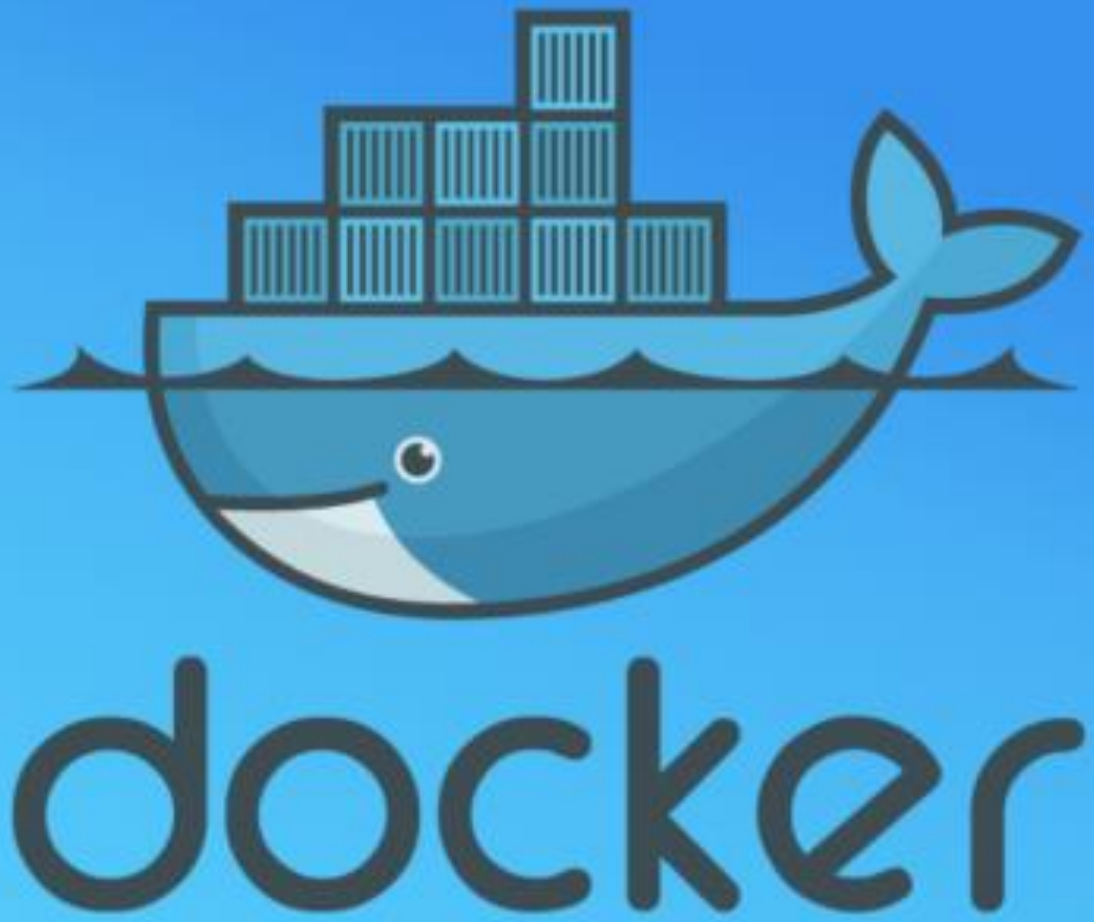


agilebridge
CONSIDERED
SOFTWARE SOLUTIONS

What You're Going to Learn Today

What is Docker, and why use it

The containerization mindset

Navigating the Basic Do’s & Don’ts

Hands-on demonstration with Docker commands and Docker Compose.

agilebridge
CONSIDERED SOFTWARE SOLUTIONS

docker

Why Docker? A quick glimpse

In the past few years, Docker has become a key player in both development and deployment processes, significantly changing the landscape of virtualization and application management.

One of Docker's standout features, especially compared to traditional virtual machines, is its speed. Docker containers can be started almost instantly—typically in less than a second—because they don't need to boot up an entire operating system each time. Instead, containers share the host system's kernel (the core part of the operating system).

agilebridge
CONSIDERED SOFTWARE SOLUTIONS

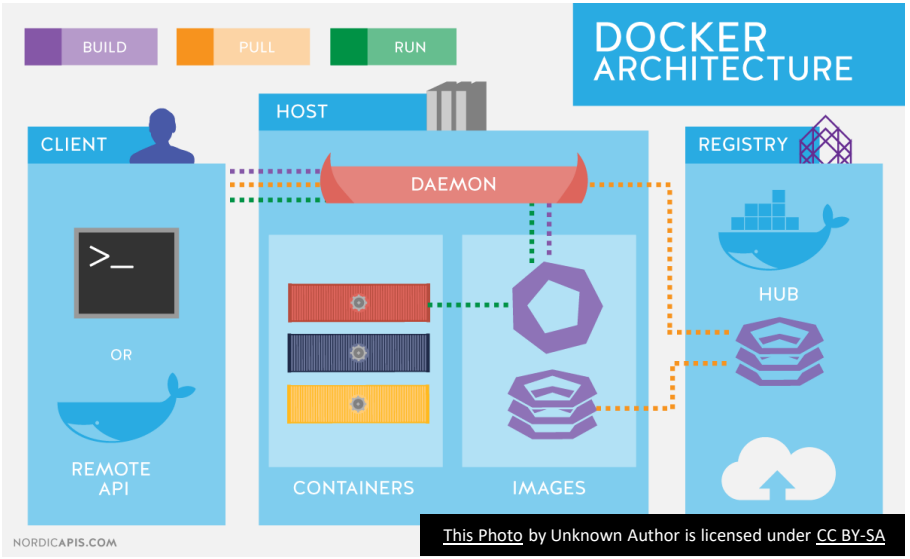
docker

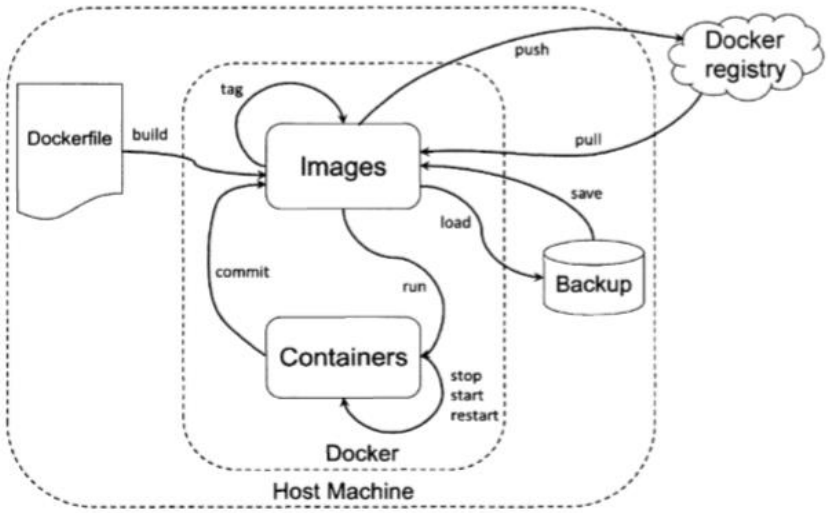
What is Docker?

- Docker is a platform that enables software to run in the same way on any system, by packaging it in containers with all the necessary components such as libraries, dependencies, and runtime. This approach eradicates the 'it works on my machine' dilemma.

But, How?

- Imagine having a sealed box (container) that can be sent anywhere, and no matter where it's opened, the content inside (software) operates the same way. Docker creates these uniform, predictable environments independent of the OS or infrastructure.





agilebridge
CONSIDERED SOFTWARE SOLUTIONS

docker

The Containerized Mindset

What you should always keep in mind

Because your container can take advantage of aut oscaling with Swarm OR Kubernetes, be aware tha t your apps run as single “EXEs” and take Distribut ed Computing into consideration.

Start thinking resiliency & auto recovery on infrastructure scale and think of containers as “Cattle”.

Some solutions

- Add a Redis cache to your tech stack to handle distributed storage overhead.
- Move workers & jobs to single responsibility CLI’s
- Take advantage of cloud patterns like circuit breaker and cache-aside. https://learn.microsoft.com/en-us/azure/architecture/patterns/

agilebridge
CONSIDERED
SOFTWARE SOLUTIONS

docker

The Do’s & Don'ts

- Always use linux based containers; Alpine, Ubuntu preferred.
- Special use cases need to be identified for windows containers
- Never use a dockerhub image in production, always define your own base images.
- Always tag your image with latest & release with semantic versioning
- When using databases, be mindful of logs/traces being committed to the image. Always keep it small.
- Always keep your docker compose core libraries updated and ensure security patches are applied when applicable.
- Always stick with project naming conventions when setting up compose files i.e. SQL should be {project-name}-database not some-sql-thing.
- Always define your subnet in your network

agilebridge
CONSIDERED
SOFTWARE SOLUTIONS

Visit The Git

https://github.com/JacquesBronk/docker-101