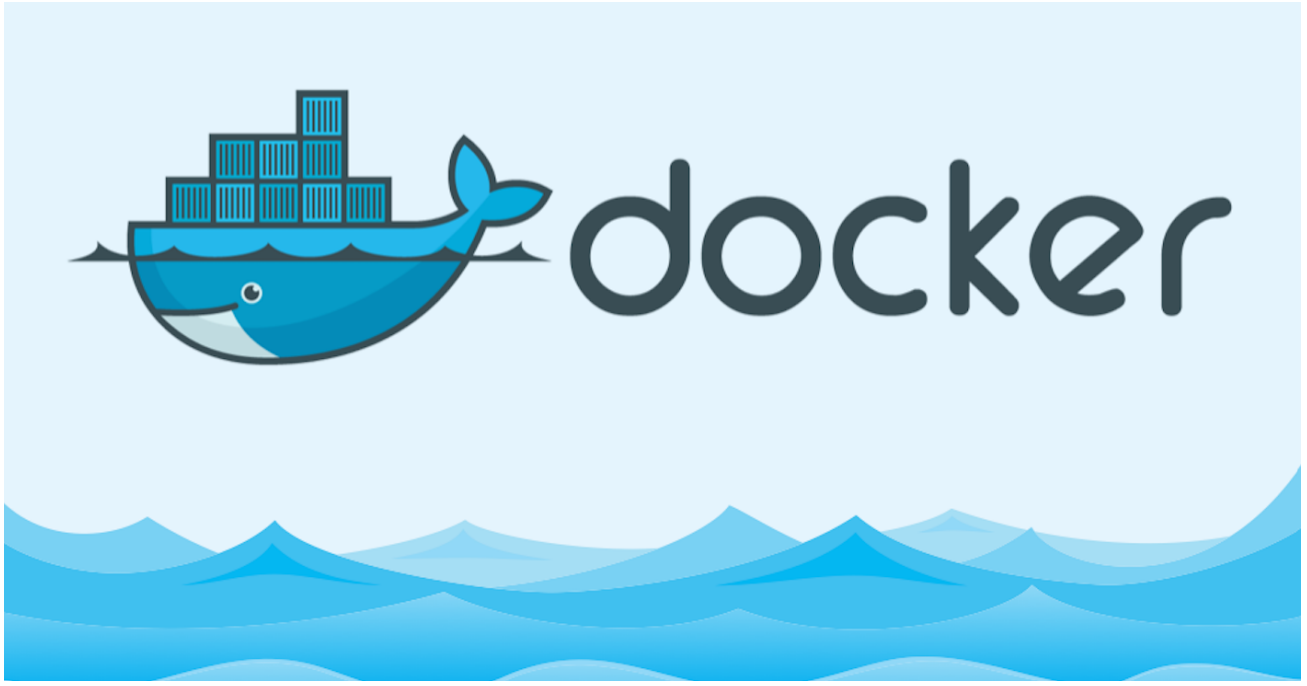


12: What is docker?

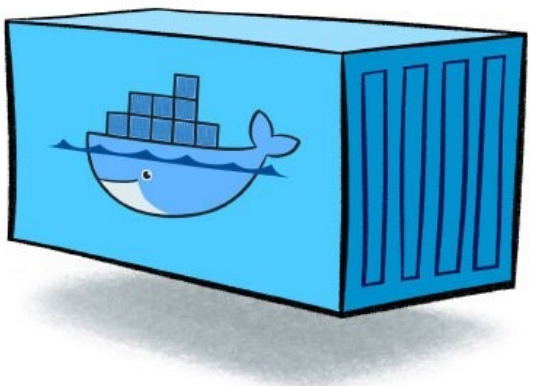
https://github.com/matthiaskoenig/itbtechtalks

Dr Matthias König
Humboldt University Berlin,
Institute for Theoretical Biology

What is docker?

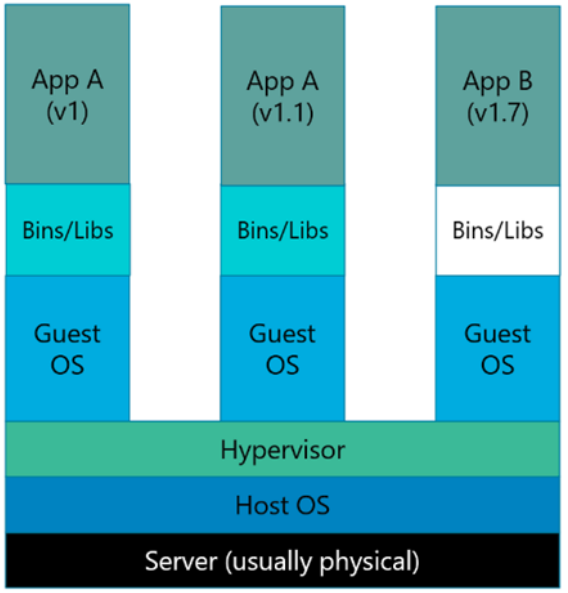
Docker is a **platform for developing, deploying, and running applications**

- **package and run an application in a loosely isolated environment** called a **container**.
- Containerization is increasingly popular because containers are:
  - **Flexible**: Even the most complex applications can be containerized.
  - **Lightweight**: Containers leverage and share the host kernel.
  - **Interchangeable**: You can deploy updates and upgrades on-the-fly.
  - **Portable**: You can build locally, deploy to the cloud, and run anywhere.
  - **Scalable**: You can increase and automatically distribute container replicas.
  - **Stackable**: You can stack services vertically and on-the-fly.
- Docker enables you to **separate your applications from your infrastructure** so you can deliver software quickly.

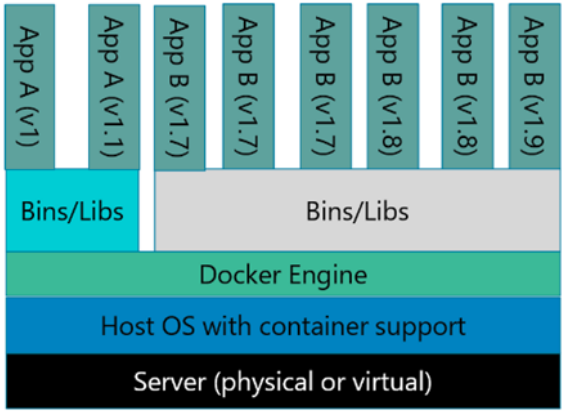


Virtual machines vs. Containers

Server Virtualisation: Each app and each version of an app has dedicated OS



Containers: All containers share host OS kernel and appropriate bins/libraries



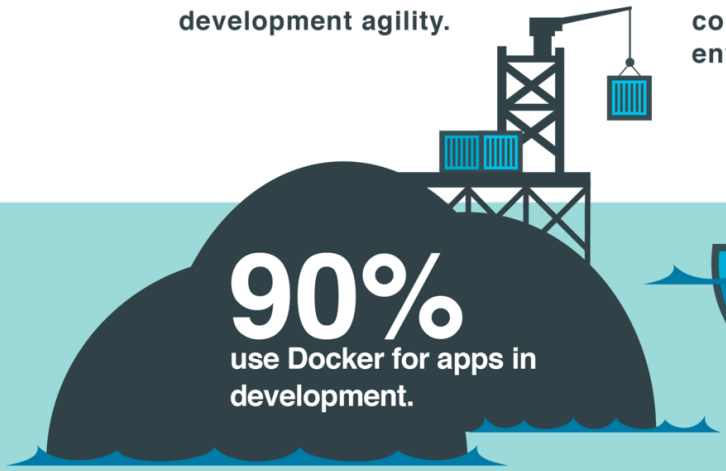
- A **virtual machine (VM) runs a full-blown "guest" operating system** with virtual access to host resources through a hypervisor.

- A **container runs natively within the host machine's kernel** and **shares the kernel** of the host machine with other containers.
- The isolation and security allow you to run many containers simultaneously on a given host.

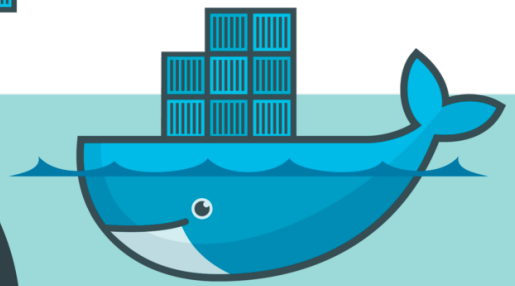
65%
use Docker to deliver development agility.

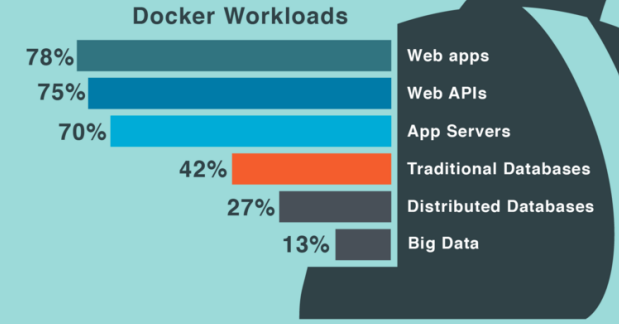
48%
use Docker to control app environments.

41%
use Docker to achieve app portability.



90%
use Docker for apps in development.





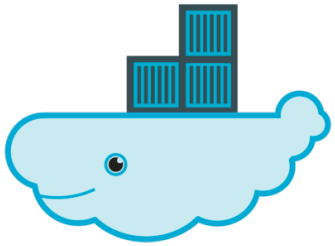
58%
use Docker for apps in production.

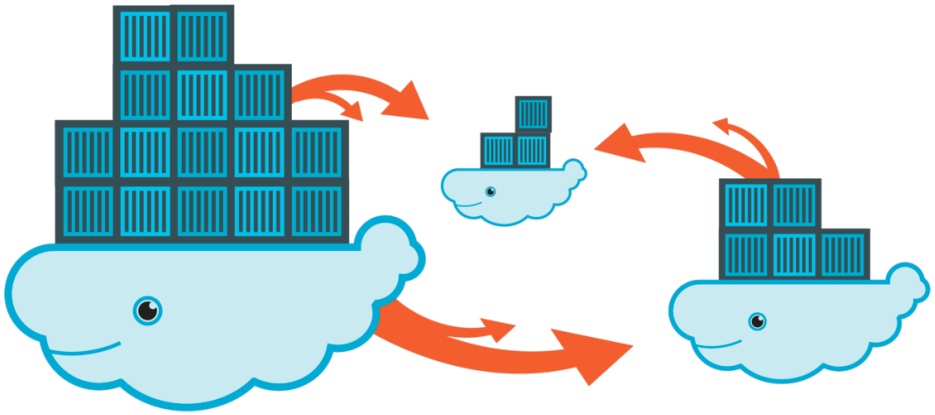
90%
plan dev environments around Docker.

80%
plan DevOps around Docker.

docker

80%
say Docker is part of cloud strategy





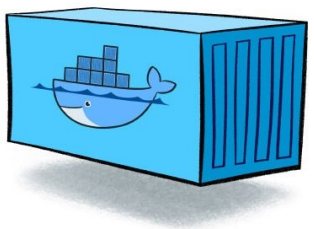
60%
plan to use Docker to migrate workloads to cloud

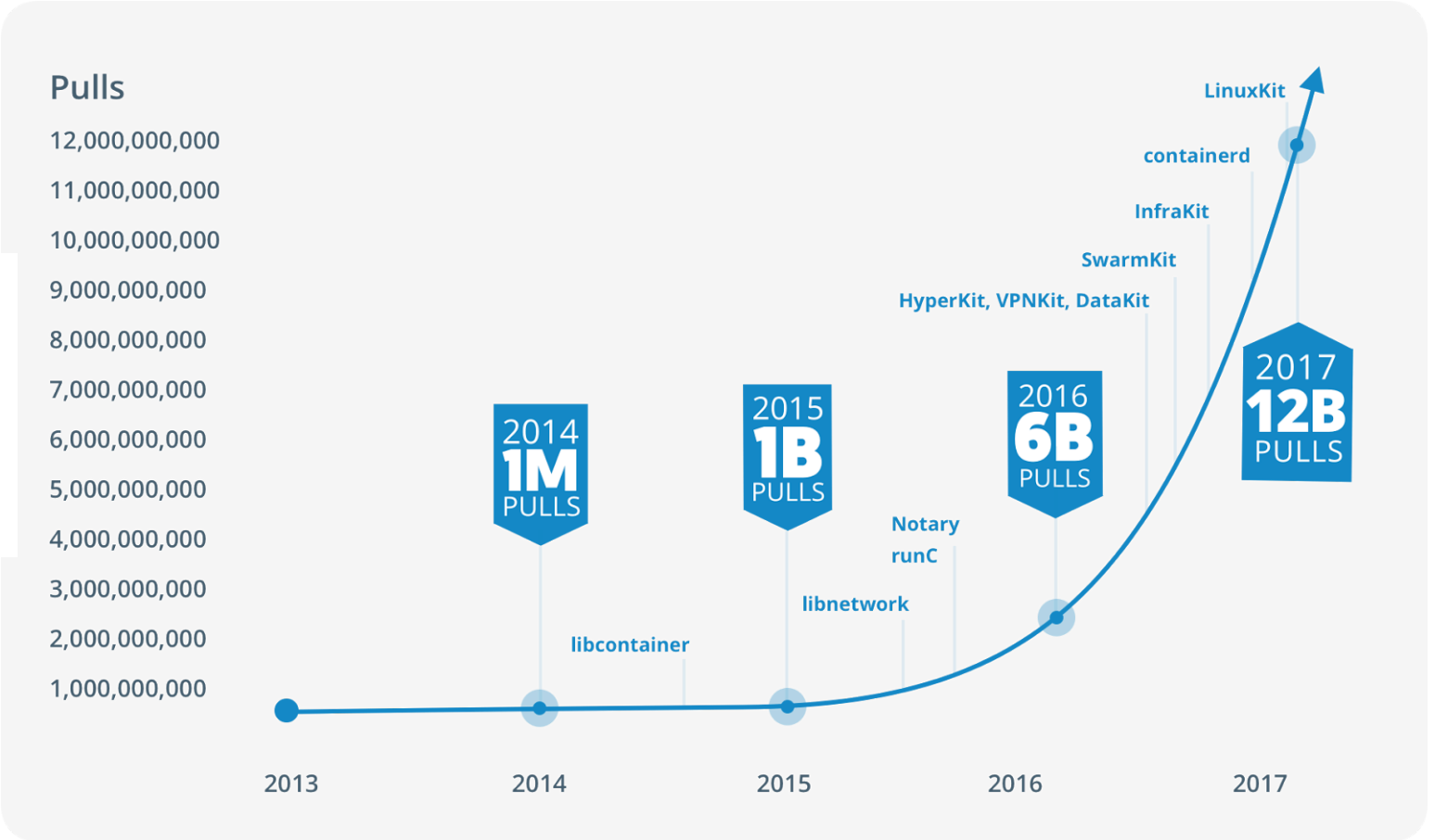
41%
want application portability across environments

35+%
want to avoid cloud vendor lock-in

docker

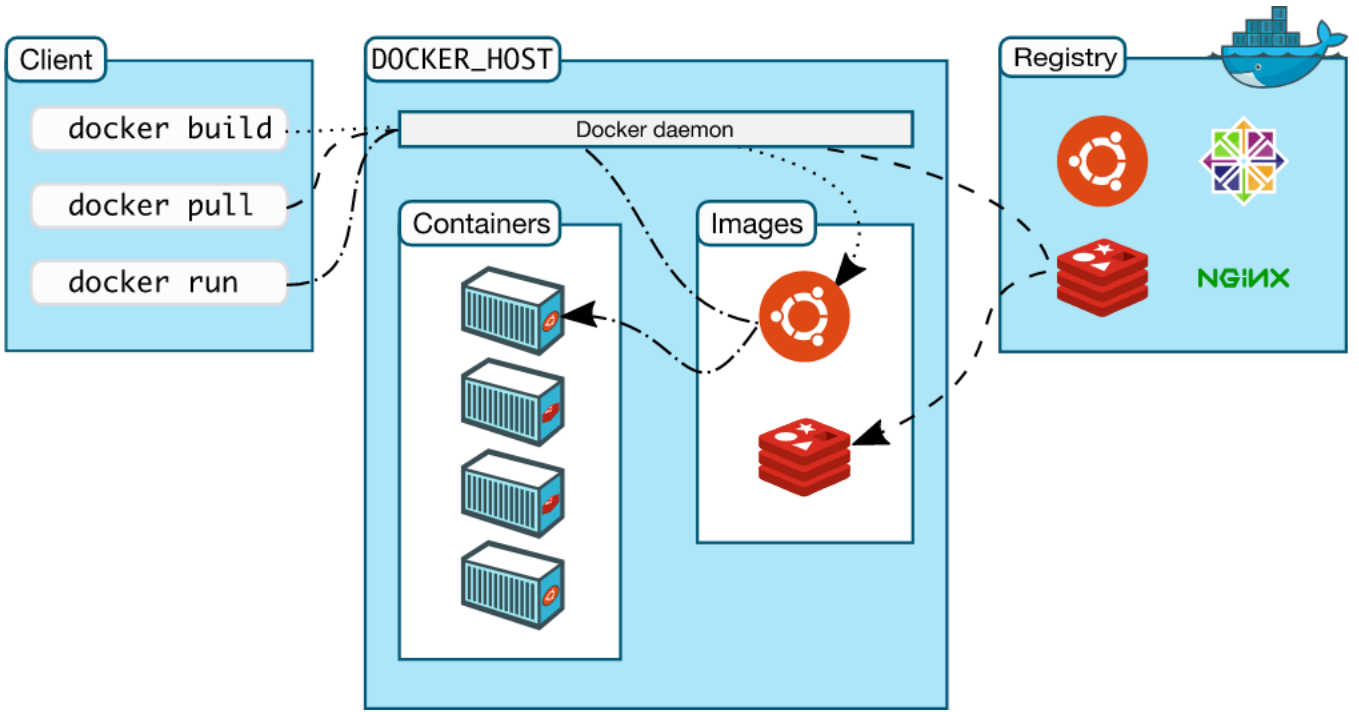
Docker containers (image pulls)





Docker architecture

- **images** are the **building plans for containers**
- images are available from **registry** (or local)
- **client-server model**
- containers are orchestrated using **docker-compose, docker swarm** or **kubernetics**

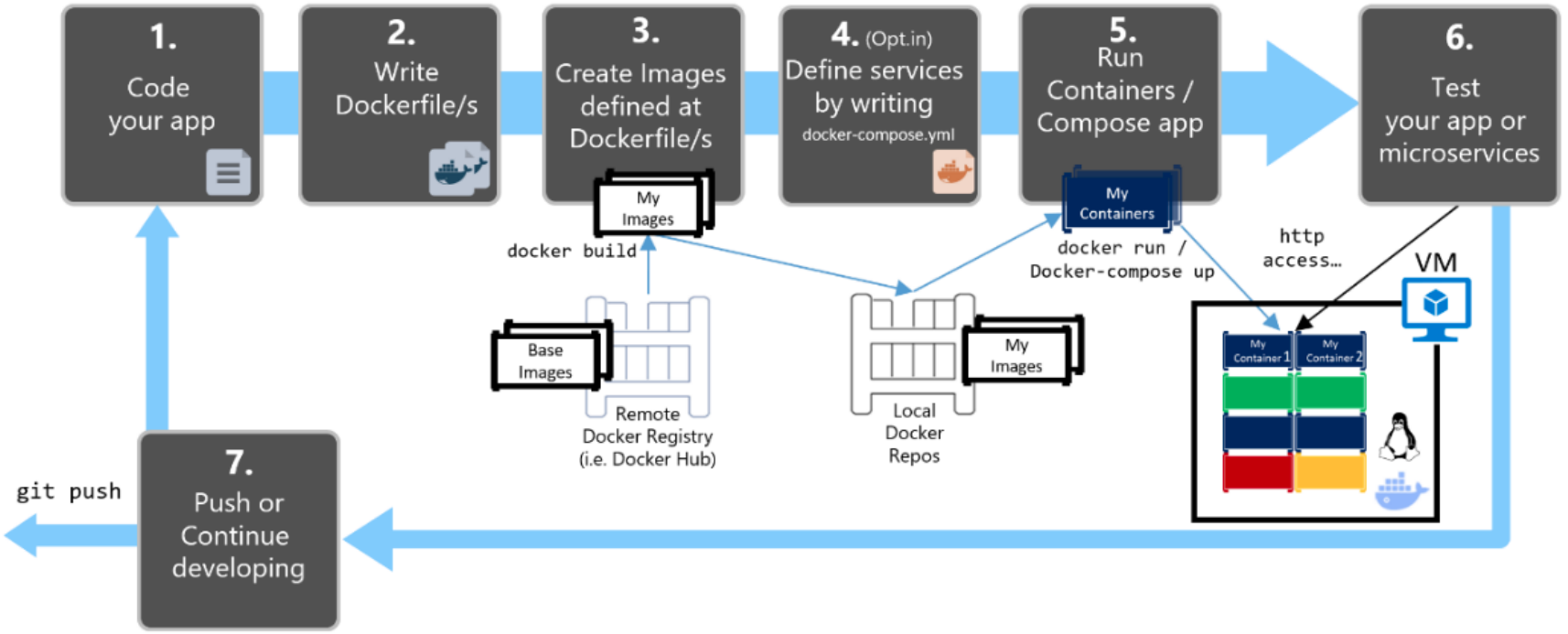


Dockerfile

- **Dockerfile defines** instructions for building an **image**

```
# Use an official Python runtime as a parent image
FROM python:2.7-slim

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed packages specified in requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

Typical docker workflow



References

- https://docs.docker.com/engine/docker-overview/
- https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/docker-application-development-process/docker-app-development-workflow
- https://github.com/fabianomenegidio/dugong-bioinformatics/blob/master/Learning.md
- https://www.esds.co.in/blog/docker-and-containers-technology/

What can I use docker for?

- Fast, consistent delivery of your applications
  - Docker streamlines the development lifecycle by allowing developers to work in standardized environments using local containers which provide your applications and services.
- Responsive deployment and scaling
  - Docker's container-based platform allows for highly portable workloads. Docker containers can run on a developer's local laptop, on physical or virtual machines in a data center, on cloud providers, or in a mixture of environments.
- Running more workloads on the same hardware
  - Docker is lightweight and fast. It provides a viable, cost-effective alternative to hypervisor-based virtual machines, so you can use more of your compute capacity to achieve your business goals. Docker is perfect for high density environments and for small and medium deployments where you need to do more with fewer resources.