universität wien

Faculty of Computer Science
Workflow Systems and Technology Group

Docker Tutorial

VU052400 Information Management & System Engineering

Ralph Vigne (ralph.vigne@univie.ac.at)

Agenda

Why should I use containers?

Popularity and Benefits of Docker

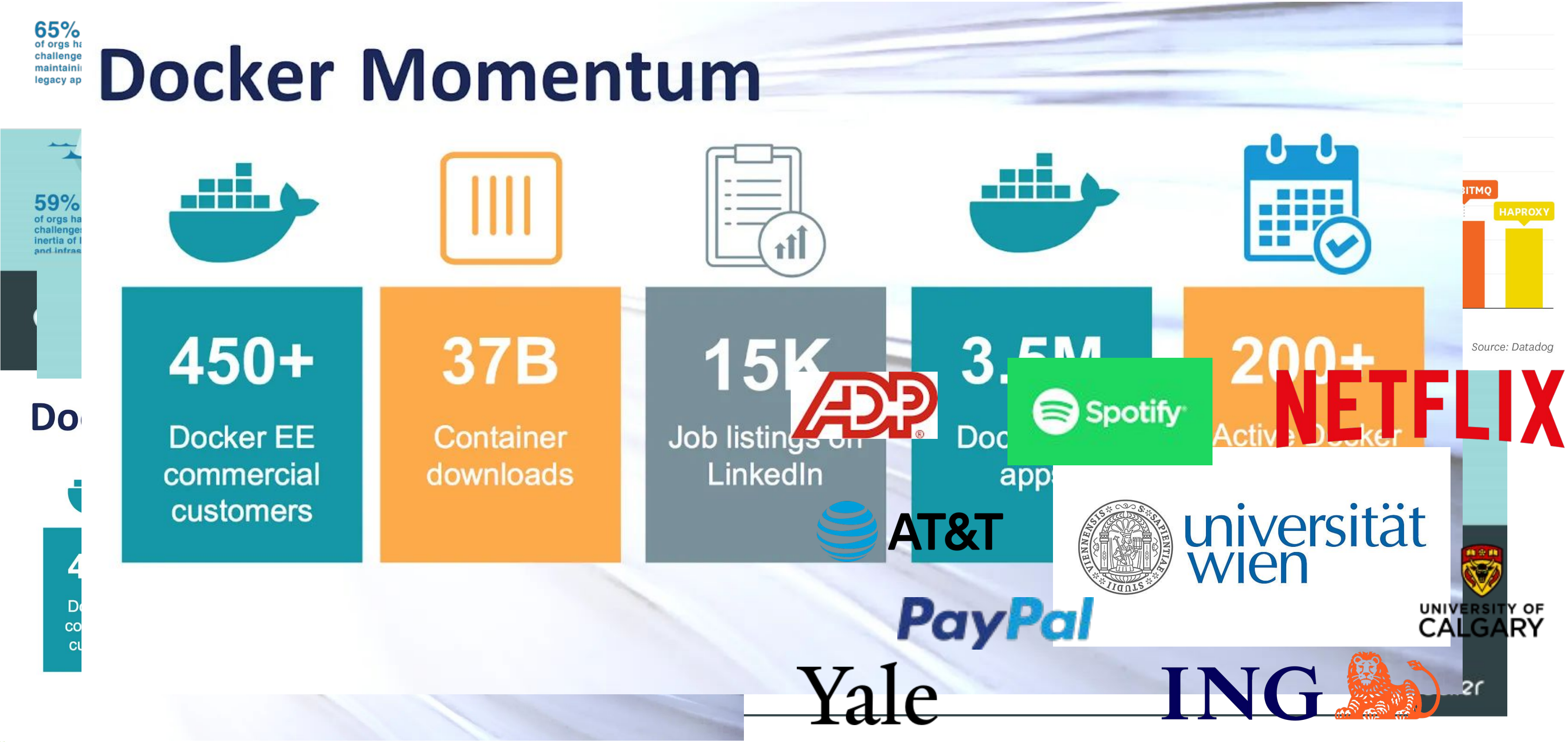
Docker Architecture & Concepts

Comparison to Traditional VMS
Architecture and Terminology of the Docker Engine

Examples

Why Should I use Containers?

Popularity



Docker Momentum

450+ Docker EE commercial customers

37B Container downloads

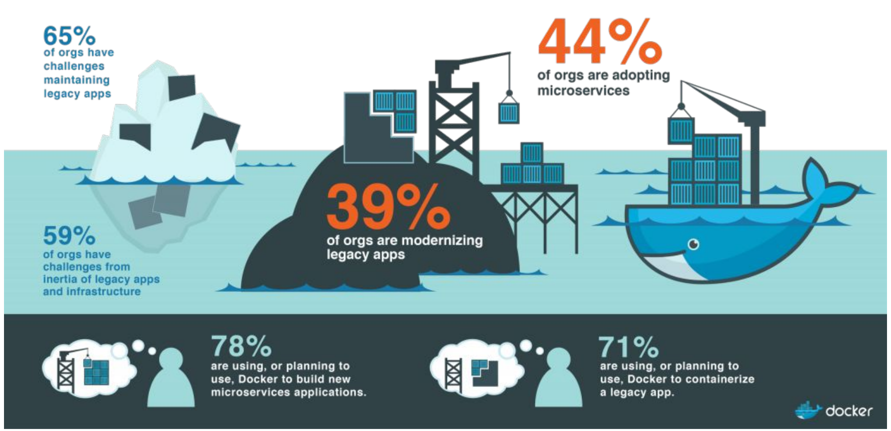
15K Job listings on LinkedIn

3.5M Doc… apps

200+ Active Docker…

What’s it all about?

universität wien



65% of orgs have challenges maintaining legacy apps

44% of orgs are adopting microservices

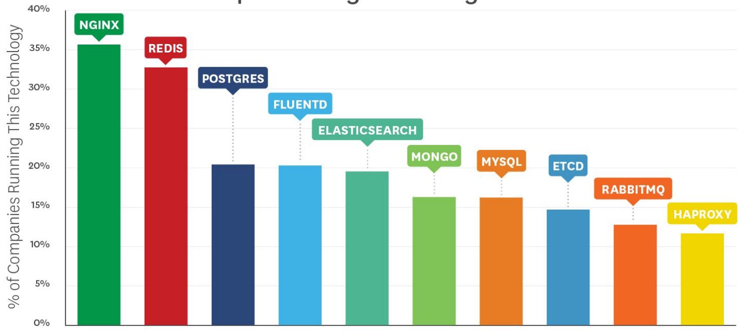
39% of orgs are modernizing legacy apps

59% of orgs have challenges from inertia of legacy apps and infrastructure

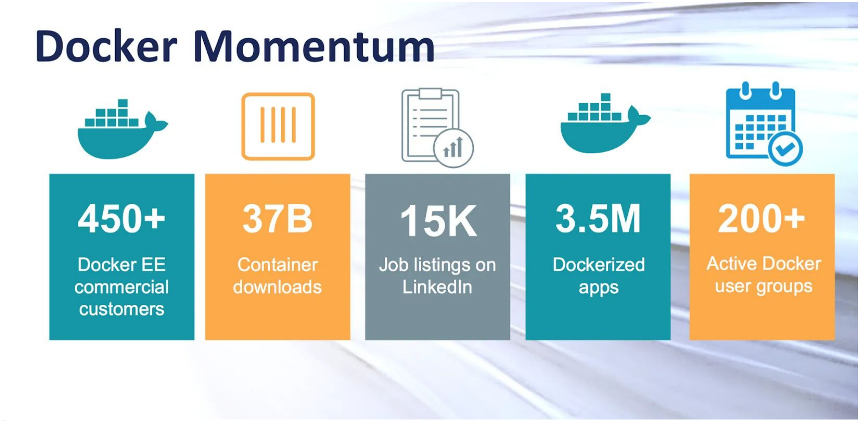
78% are using, or planning to use, Docker to build new microservices applications.

71% are using, or planning to use, Docker to containerize a legacy app.

Top Technologies Running on Docker



Source: Datadog



Docker Momentum

450+	37B	15K	3.5M	200+
Docker EE commercial customers	Container downloads	Job listings on LinkedIn	Dockerized apps	Active Docker user groups

ADP

Spotify

NETFLIX

AT&T

universität wien

PayPal

UNIVERSITY OF CALGARY

Yale

ING

Benefits of Docker

- Multi-Cloud Platforms
- Environment Standardization and Version Control
- Continuous Deployment and Testing
- Isolation
- Security

Source: https://dzone.com/articles/5-key-benefits-docker-ci

Increase Return of Investment

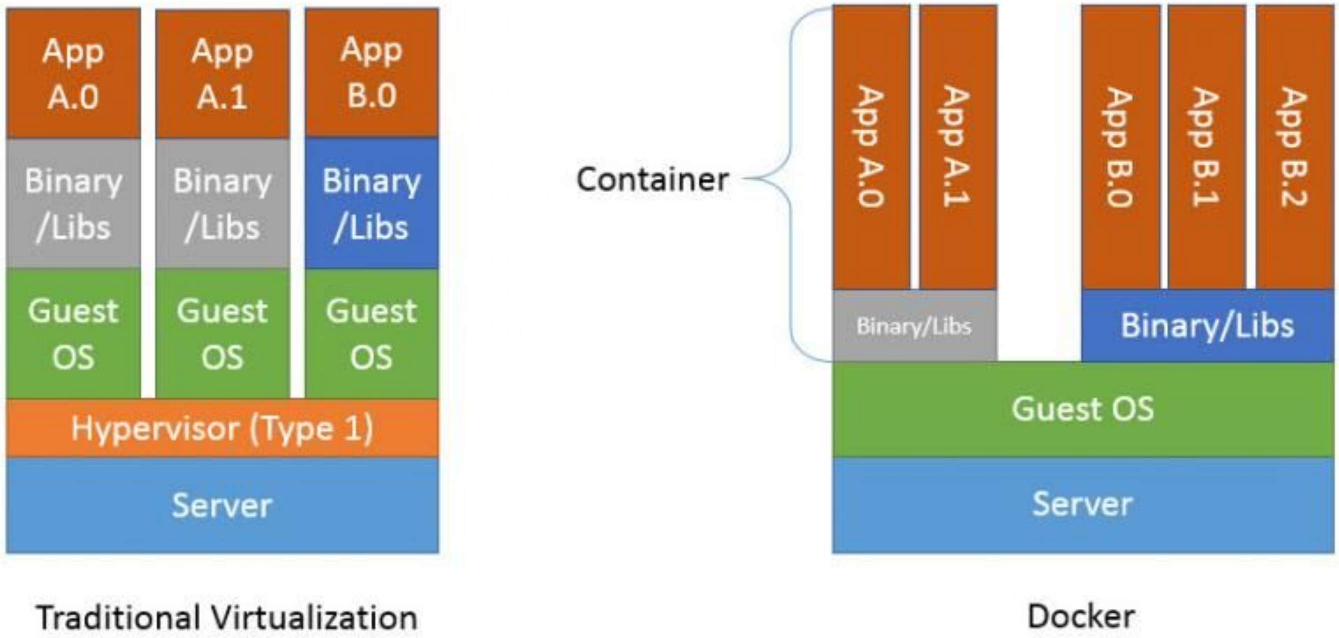
- Improved Resource Efficiency
  - No more dedicated “bare-metal” server
  - Reducing the number of physical hosts
  - Cloud bursting to compensate for peak loads

- Shorter Development Cycles
  - Rapid deployment and continuous integration processes
  - Faster ramp-up times of new employees
  - More stable releases

Source: https://dzone.com/articles/5-key-benefits-docker-ci

Docker Architecture & Concepts

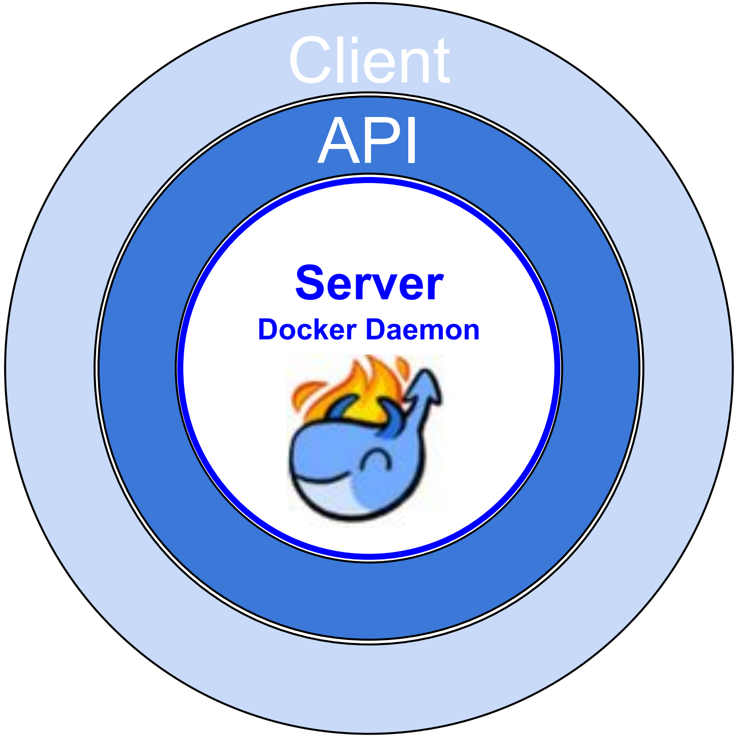
Comparison to VMs



Traditional Virtualization

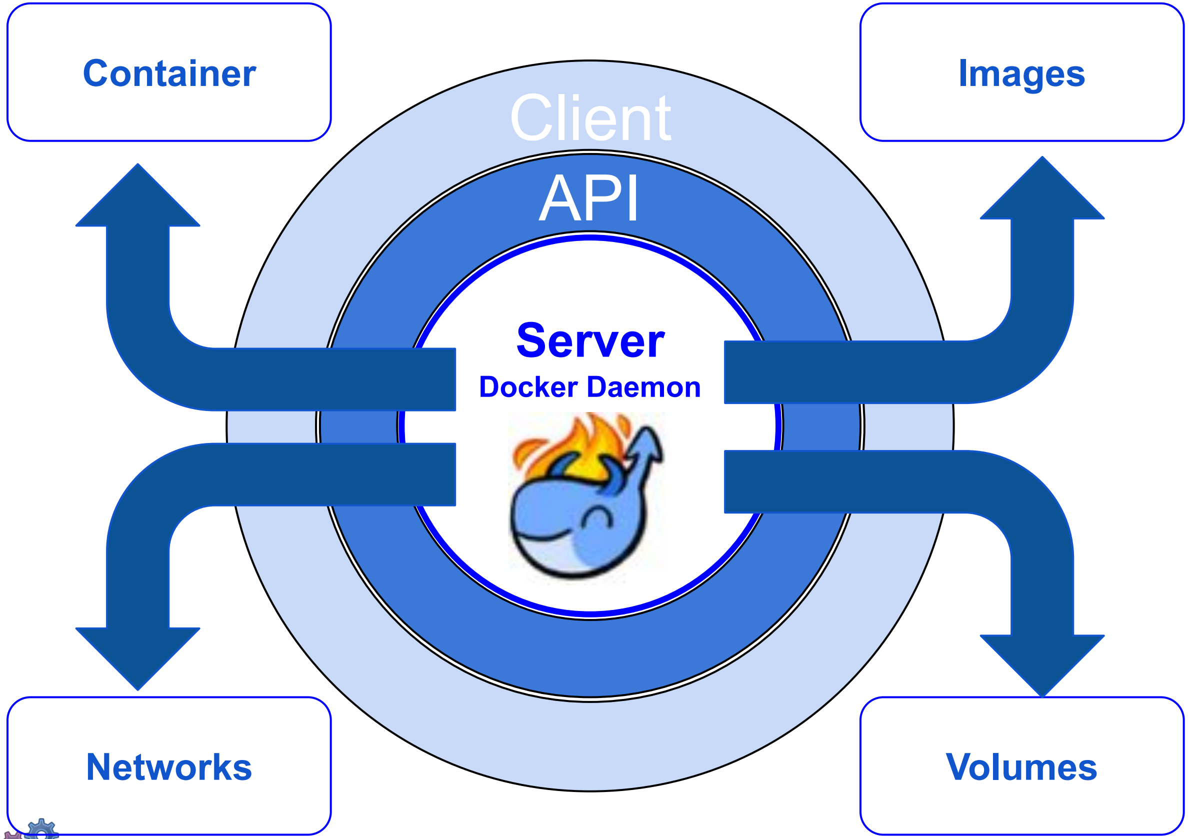
Docker

Docker Engine



- long-running program called a daemon process (the ***dockerd*** command)
- interfaces that programs use to talk to the daemon and instruct it
- A command line interface (CLI) client (the ***docker*** command).

Docker Engine



- An image is a read-only template with instructions for creating a Docker container.
- A container is a runnable instance of an image
- Connect a container to one or more isolated networks
- Volumes are storage to persist and share data between containers

Examples

Example 1: Getting Started

- Start the daemon
- Run “Hello World”
- Installing and Using 3rd party images

Example 2: Building and Stacking

1. Run the Application Locally
2. Building the Container Manually
3. Use Dockerfiles to Automate Build process
4. Use Docker-Compose to Define the Application Environment
5. Define a Stack with Multiple Services

Conclusion

- Companies want to get rid of **legacy apps**
- Docker plays nice in the **Cloud**
- Docker promises to **improve the ROI**
- Docker is a **distributed** software
- Images can be found on **DockerHub**
- A **Dockerfile** makes you image reproducible and **docker-compose** manages your complete stack

Resources

Docker Quickstart: https://docs.docker.com/get-started/

Develop with Docker: https://docs.docker.com/develop/

Docker Compose: https://docs.docker.com/compose/

Tutorial Video: https://youtu.be/Fuq22uaZhIU

Git Repository: https://github.com/vigne/docker-tutorial

FIN.

This tutorial is based on the Docker documentation available at: https://docs.docker.com/