E4 COMPUTER ENGINEERING

WHEN PERFORMANCE MATTERS

DECICE

Introduction to key technologies I: Containers and container orchestration systems

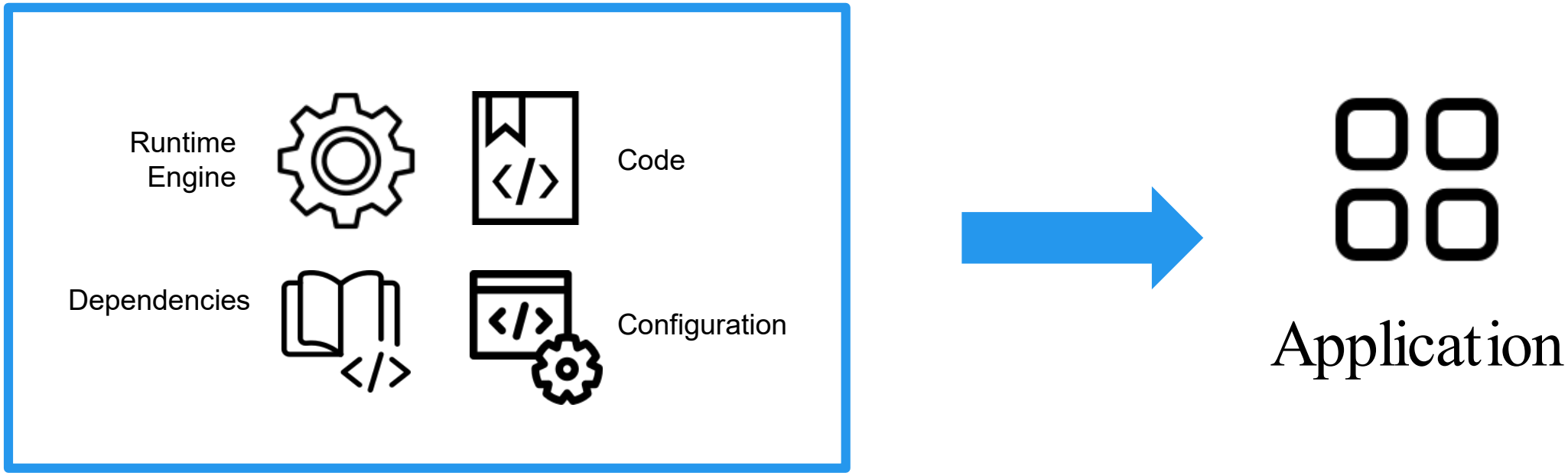
Riccardo Cavadini – riccardo.cavadini-ext@e4company.com

DECICE Consortium Event – November 06, 2023

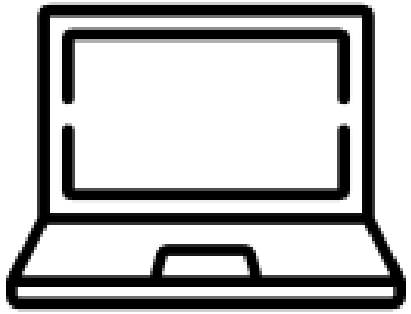
OUTLINE

- **Software deployment** problems
- **Containers** a solution
- **Containers** examples
- **Container orchestration systems**
- **Container orchestration systems** examples

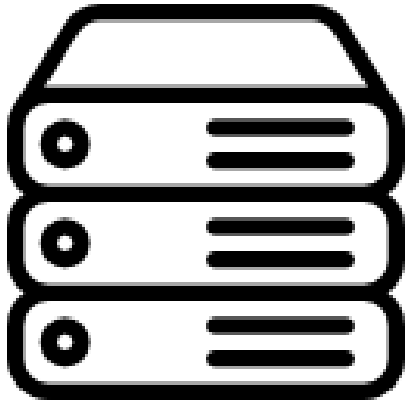
SOFTWARE DEPLOYMENT: APPLICATION COMPONENTS



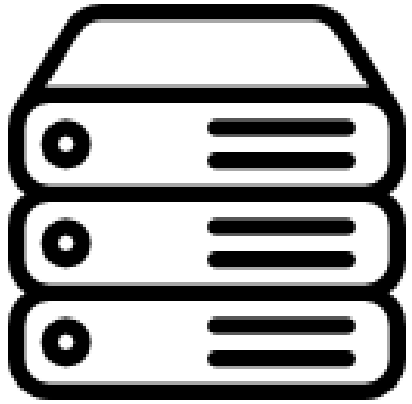
SOFTWARE DEPLOYMENT: ENVIRONMENTS



Developer Laptop

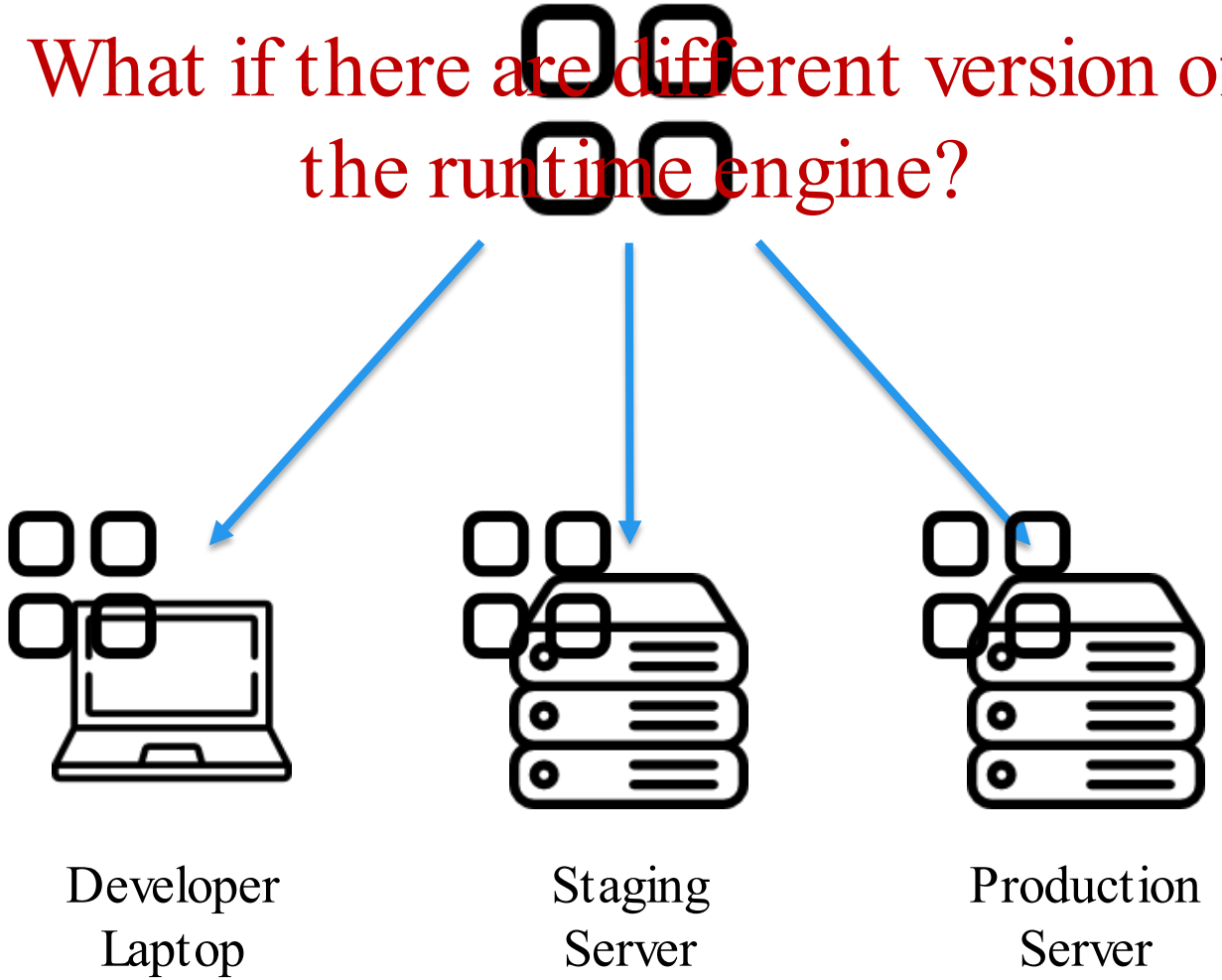


Staging Server



Production Server

SOFTWARE DEPLOYMENT: PROCESS



SOFTWARE DEPLOYMENT: PROBLEMS

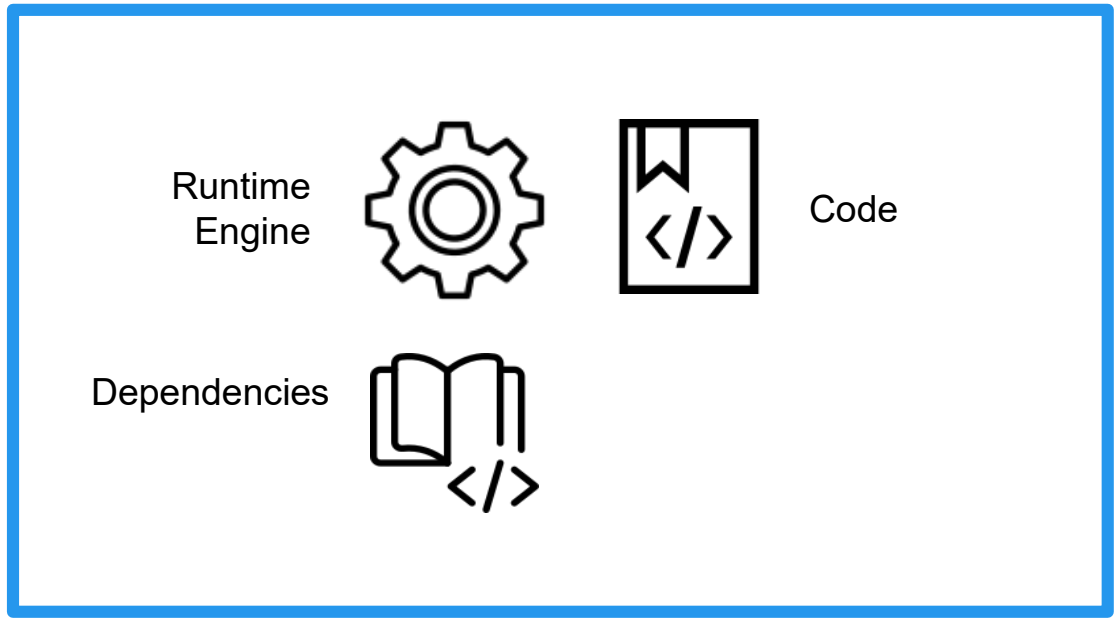
Complex environments
Increasing complexity in software dependencies

Inconsistencies
Discrepancies between development and production environments

Scaling issues
Difficulty in scaling applications seamlessly

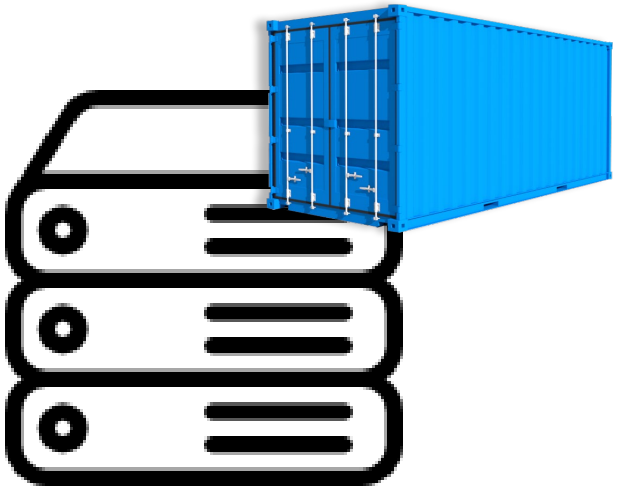
"It worked on my computer" problem

CONTAINERS: A SOLUTION

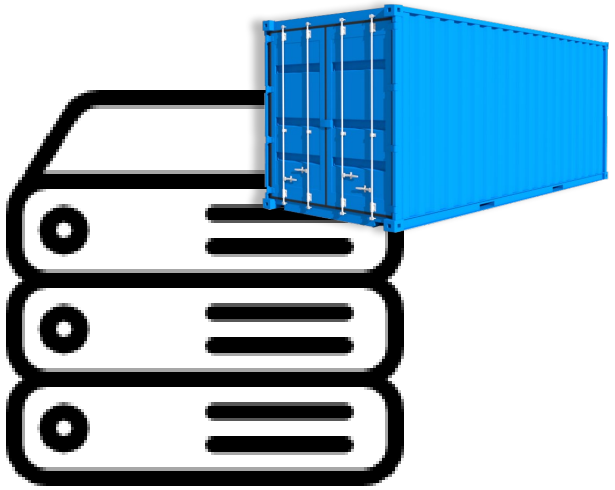


SOFTWARE DEPLOYMENT: ENVIRONMENTS

Developer Laptop

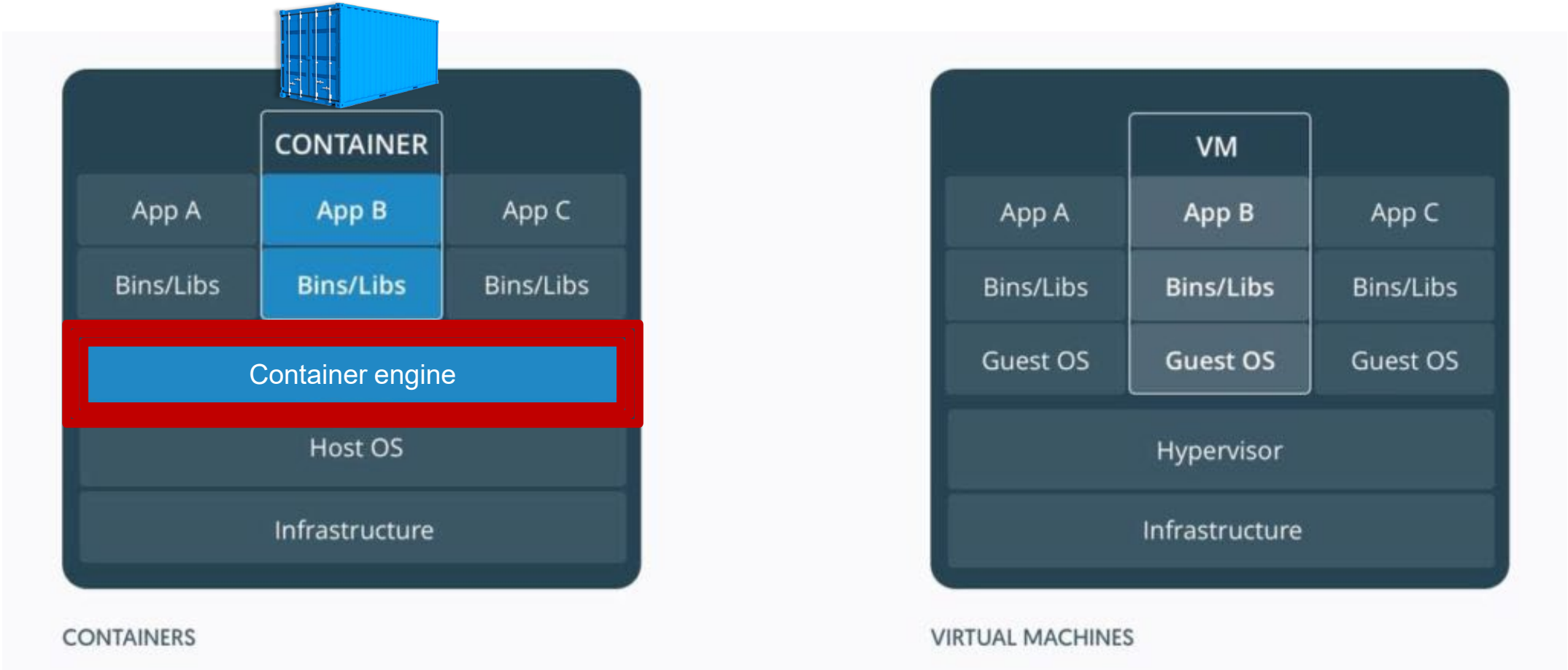


Staging Server



Production Server

CONTAINERS VS VIRTUAL MACHINES



DOCKER

Dockerfile

Docker Image

Docker Container

DOCKER: COMMANDS

- **docker build**: to build an image from a Dockerfile
- **docker run**: to start a new container from an image
- **docker ps**: to list all the running containers
- **docker stop**: to stop a running container
- **docker rm**: to remove a stopped container
- **docker images**: to list all the images on the system
- **docker pull**: to download an image from a registry
- **docker exec**: to execute a command inside a running container
- **docker-compose**: to manage multi-container applications

DOCKER: LIMITATIONS

Needsprivileges

Docker daemon needs root privileges (possible security concern)

Enterprise-oriented

Allows for an easy micro-service virtualization, but is not compatible with traditional HPC

SINGULARITY: A (DIFFERENT) ALTERNATIVE

Doesnot needroot privileges
Doesnot usea daemon

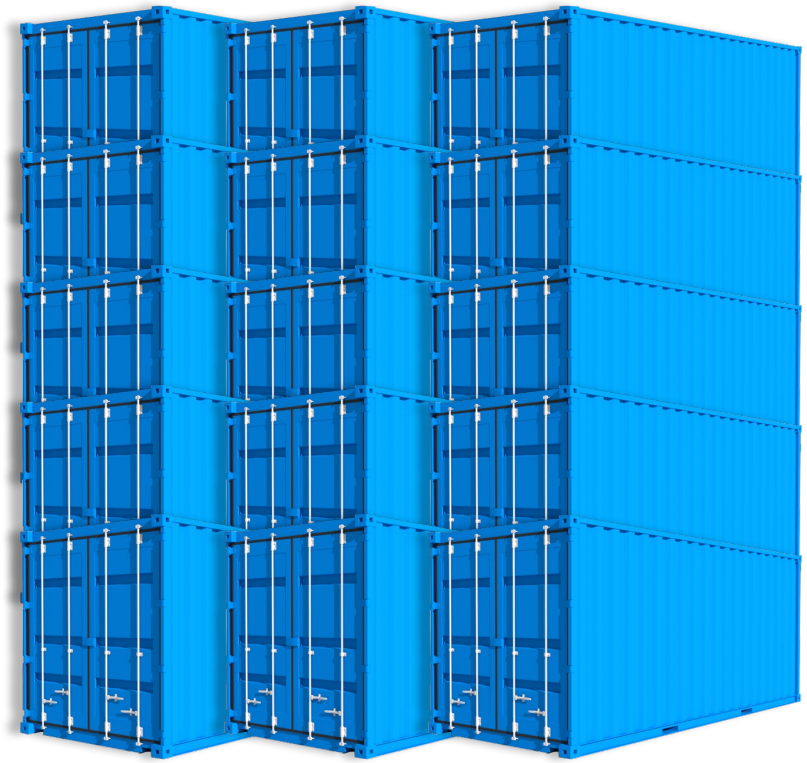
Born for scientific applications

Singularity is designed for general scientific use cases
The resource request matches the actual usage
It's easier to match resources requirements

HPC-oriented

Singularity containers offer native support for GPUs, InfiniBand, MPI

MULTIPLE CONTAINERS: A THREAT TO SCALABILITY



Manual work increases

To scale up services, to fix crashing nodes, to run services

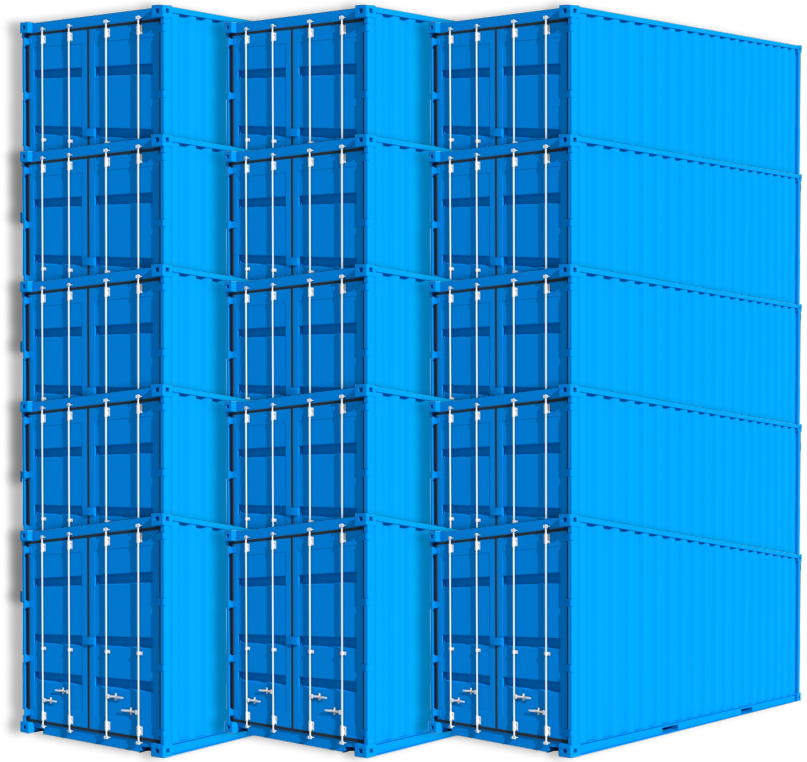
Complexity increases

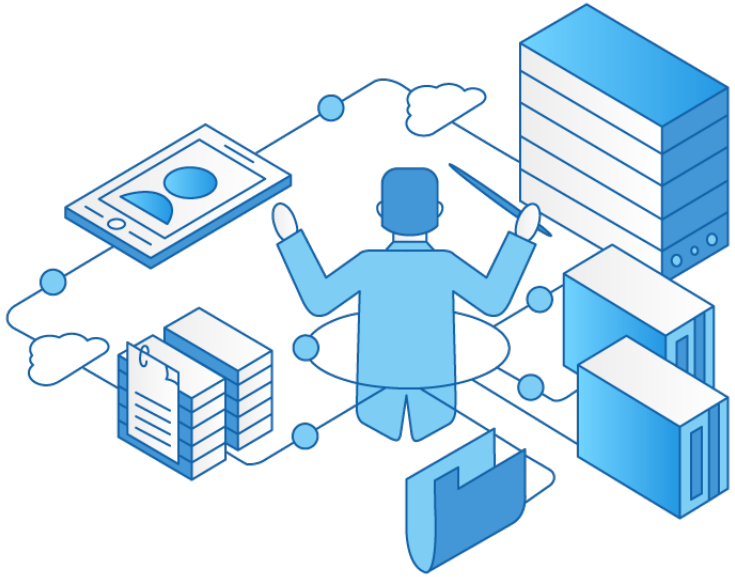
To run new elements in production, to correctly scale with many containers

Cost increases

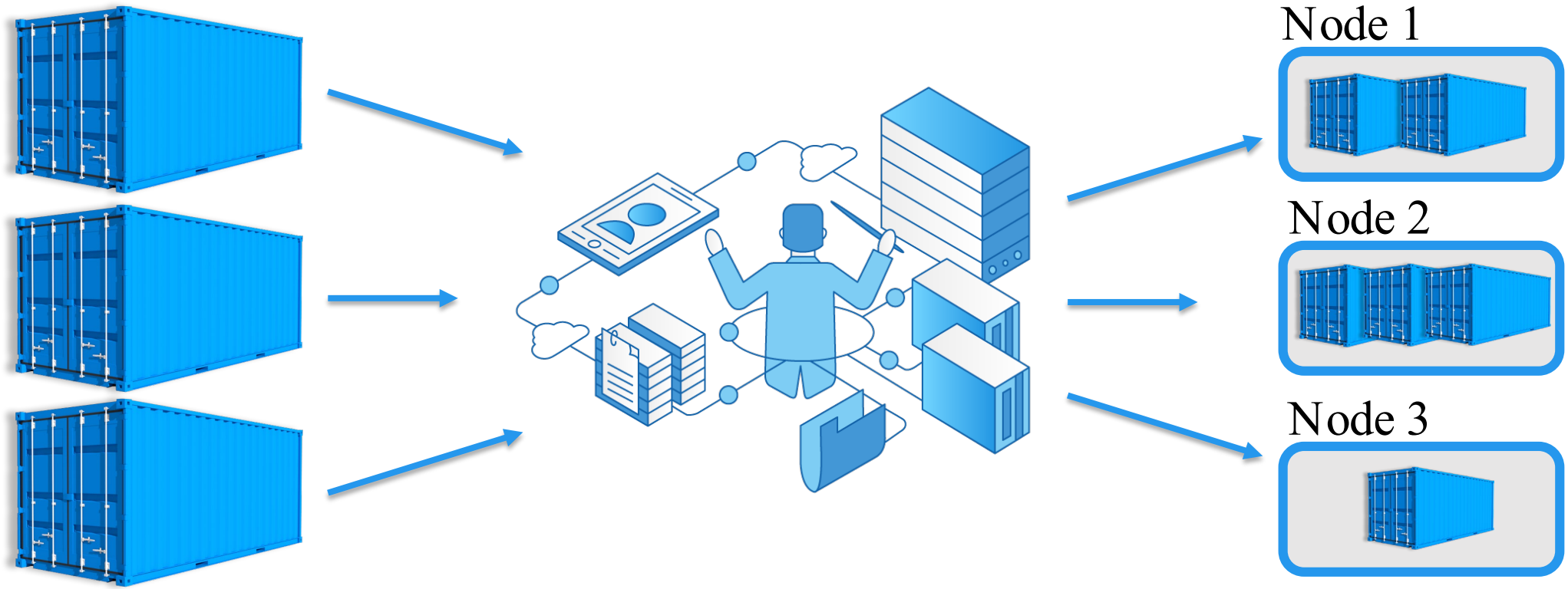
In terms of human cost and public cloud cost

CONTAINER ORCHESTRATION





CONTAINER ORCHESTRATION



CONTAINER ORCHESTRATION

Deployment
Optimal resource usage, automatic scalability

Networking
Auto-discovery, accessibility from outside

Management
Load balancing, Fault tolerance, updates/rollbacks

KUBERNETES

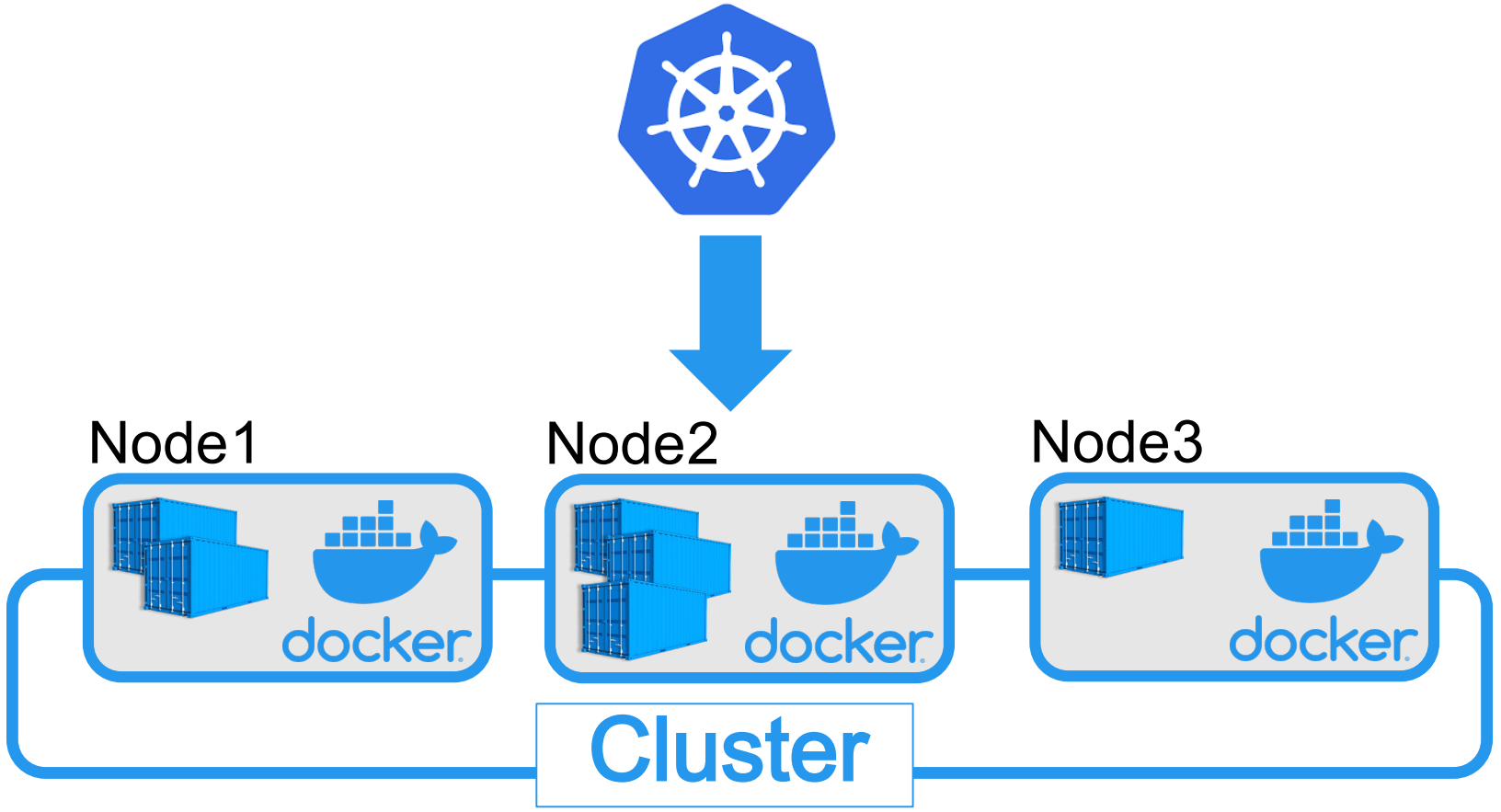
Open source

Launched by Google, now part of the Cloud Native Computing Foundation with a big community
Compatible with all cloud vendors, extensible and portable

Automatic

Automates container deployment, scaling, management and load balancing

KUBERNETES



E4
COMPUTER
ENGINEERING

GRAZIE
THANK YOU