amadeus

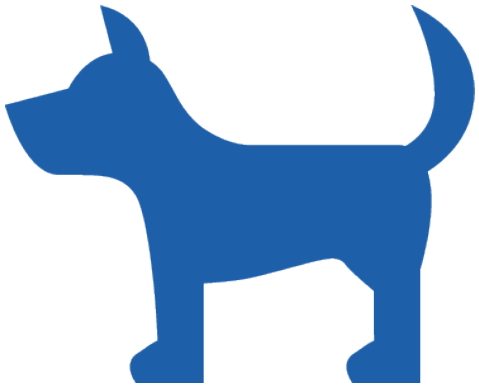
Next Generation Containerized Application Platform

From a collection of servers to the cloud

Polo-Francois Poli
Cloud & Application Services

Server Management

Cattle vs Pets



amadeus

Server Management

Decouple **what** should run
from **where** it should run

Automation

Application centric deployment

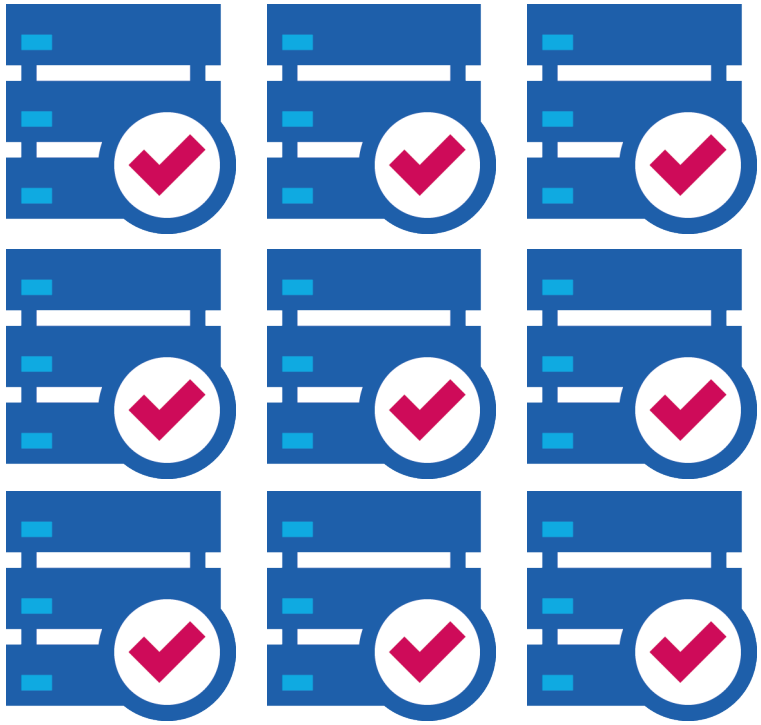
- Deploy the application **as a whole**
  - With **all** its dependencies

- Reproducibility
- Technology agnostic
- Homogeneity
- Simplify operations

Automation

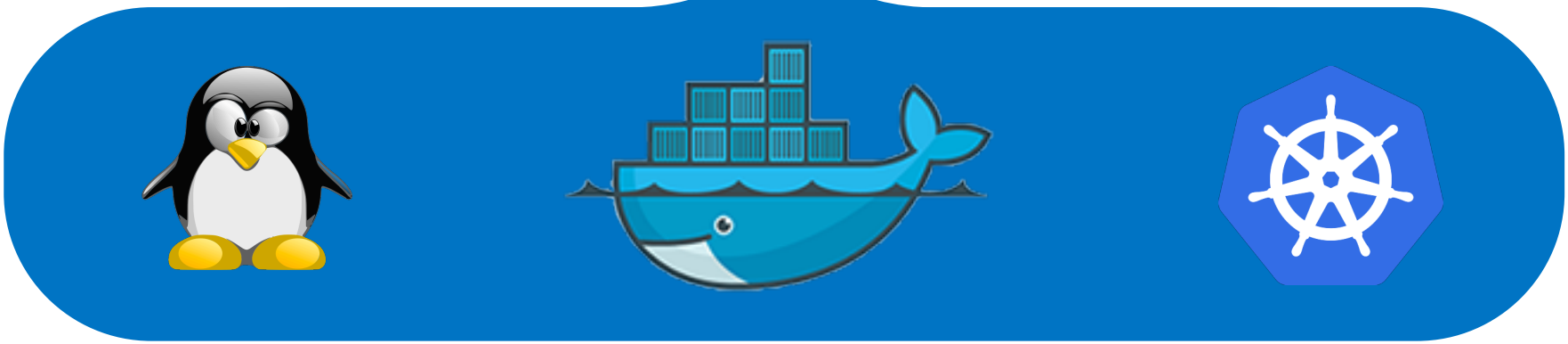
Instances anywhere

- Manage a cluster **as a whole**
  - Any task on any server
  - Start tasks in the cluster



Let the system manage machines and assign tasks to machine automatically

Technological Stack



amadeus

OPENSHIFT

Open-Source

- Linux container technology
- Container image format

kubernetes by Google

- Orchestration of Docker containers
- Cluster management

amadeus

Linux Containers

Kernel features only

cgroups (Control Groups)

- Limit, account isolate resource usage
- CPU, Memory, Disk I/O, Network
- Think ‘ulimit’ for group of processes

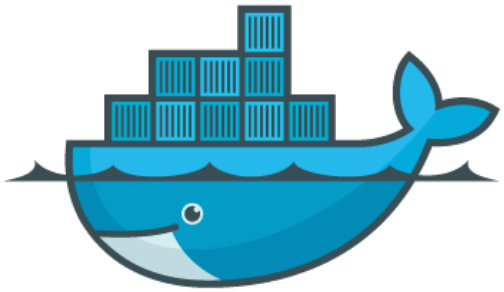
Kernel namespaces

- pid (processes)
- net (network interfaces, routing...)
- ipc (System V IPC)
- mnt (mount points, filesystems)
- uts (hostname)
- user (UIDs)

Docker Linux Containers

Image Format

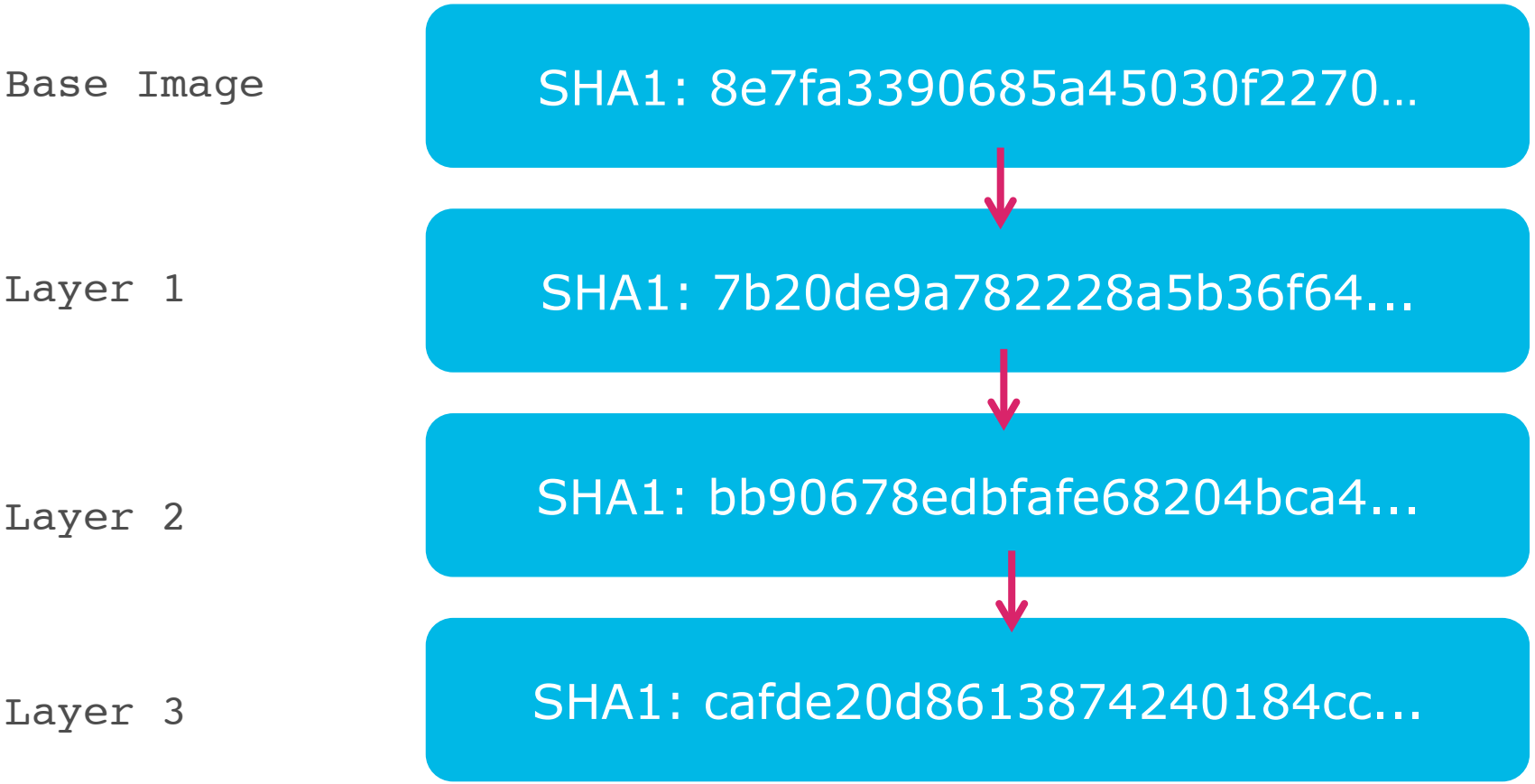
- Kernel front-end
  - Docker daemon & client
  - Easily create containers



- Container Images
  - Present a filesystem to the container
  - Build, version, deploy images

Docker Image format

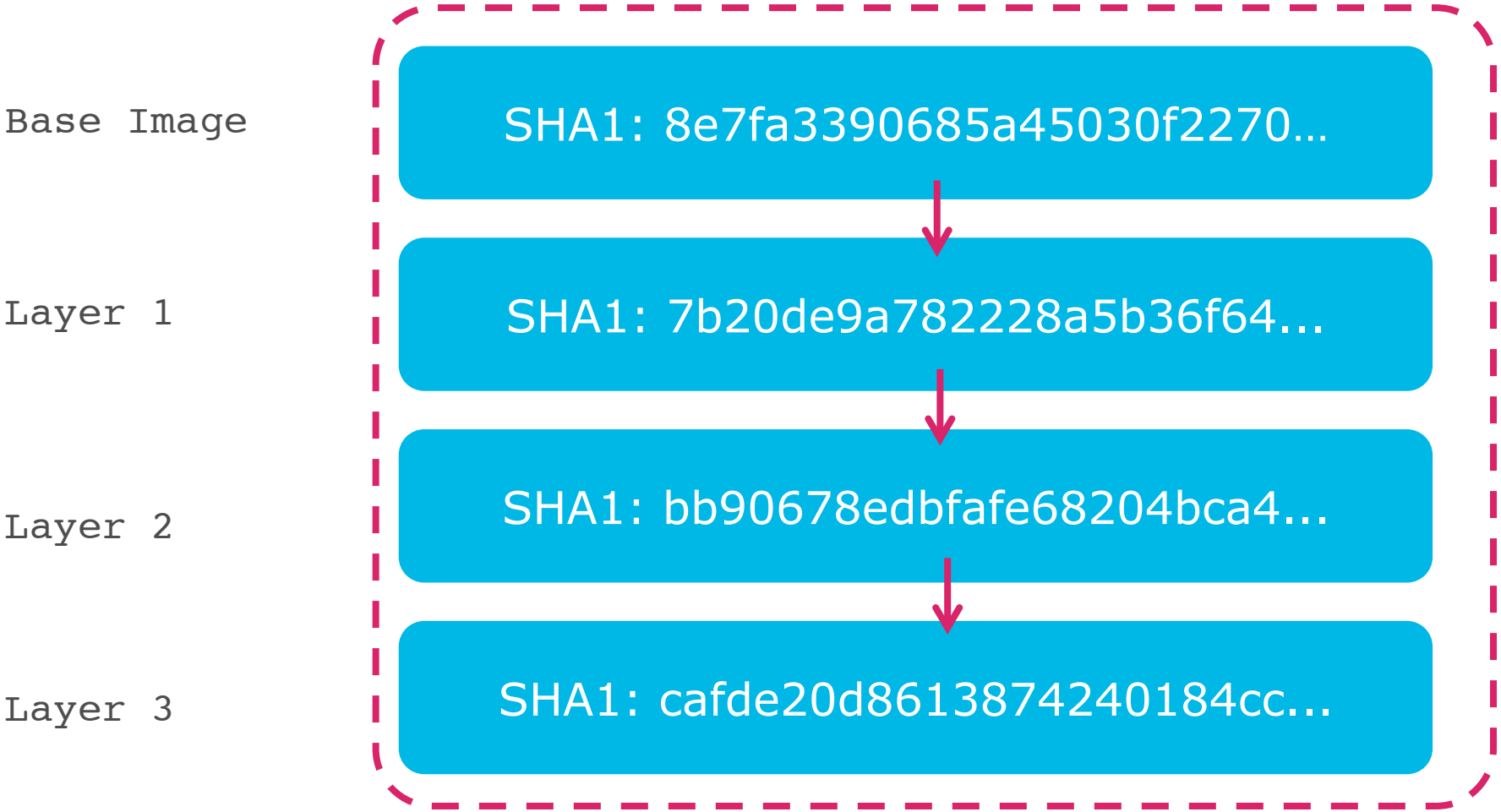
Tree structure



amadeus

Docker Image format

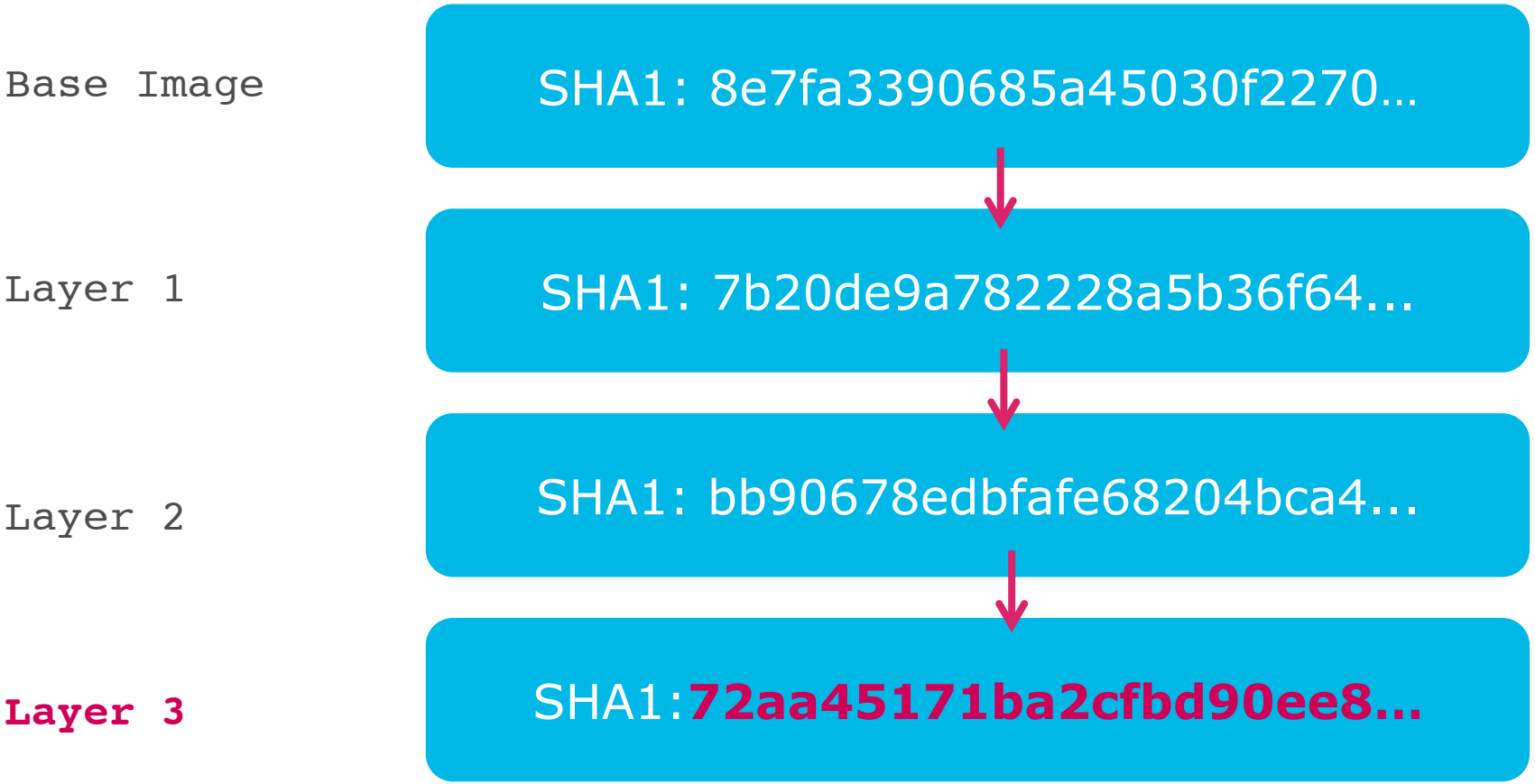
Tree structure



amadeus

Docker Image format

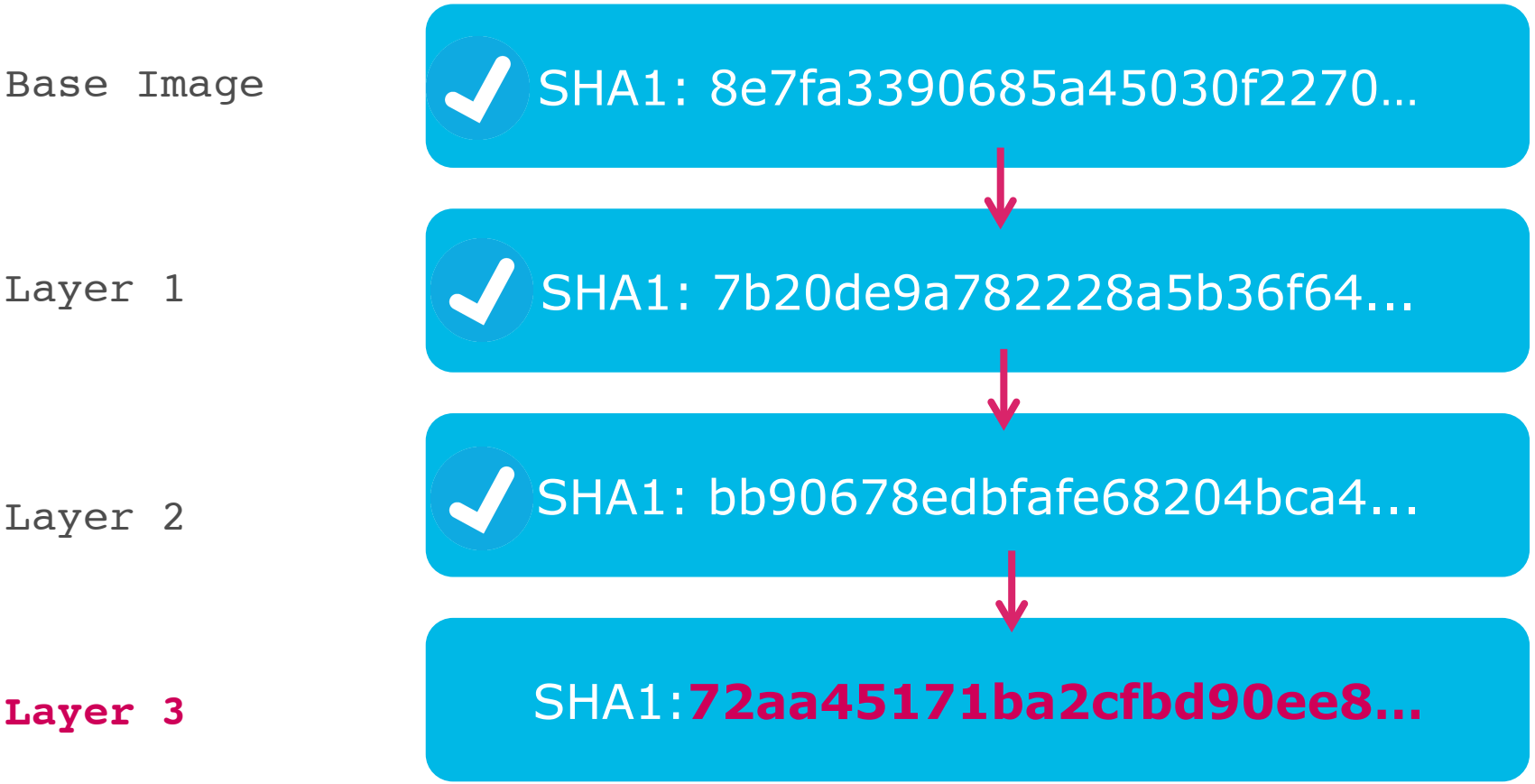
Tree structure



amadeus

Docker Image format

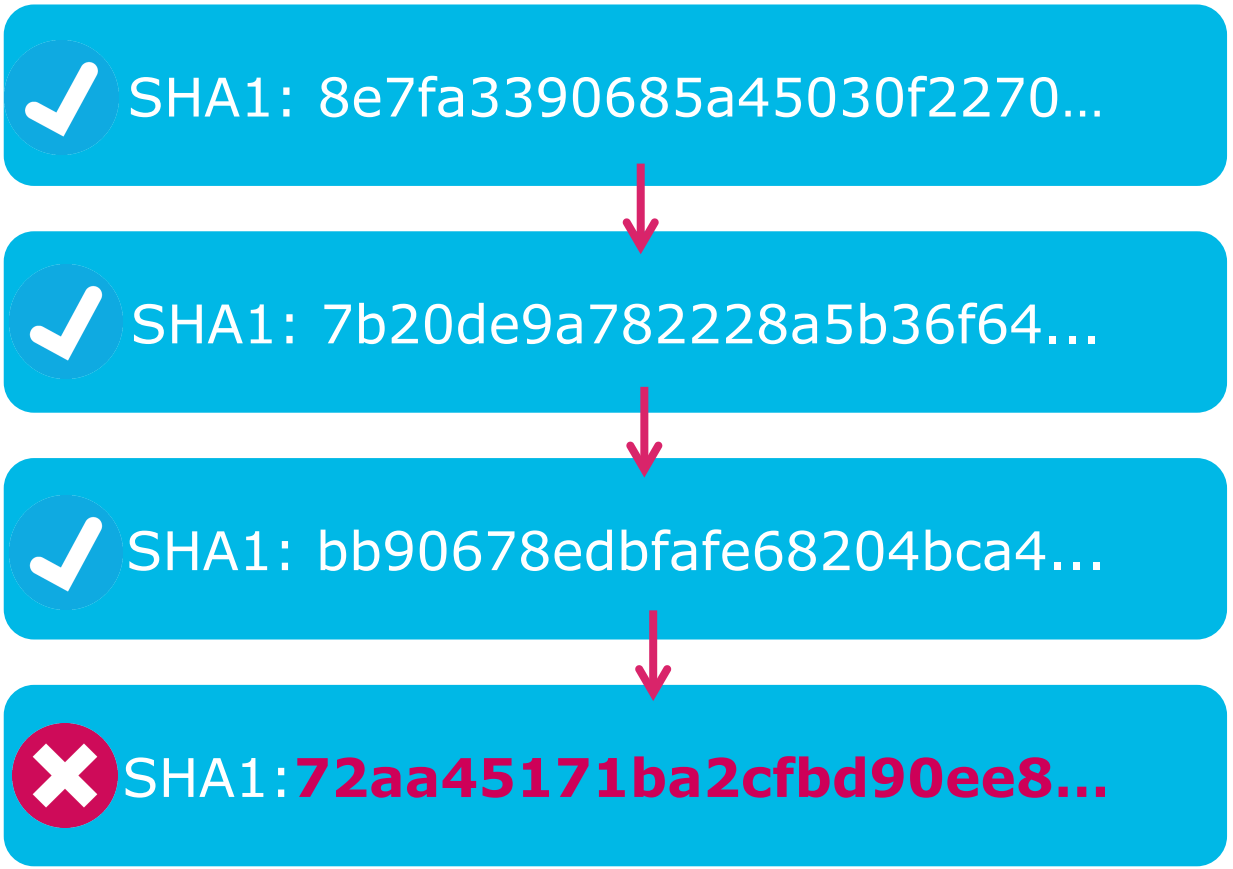
Tree structure



amadeus

Docker Image format

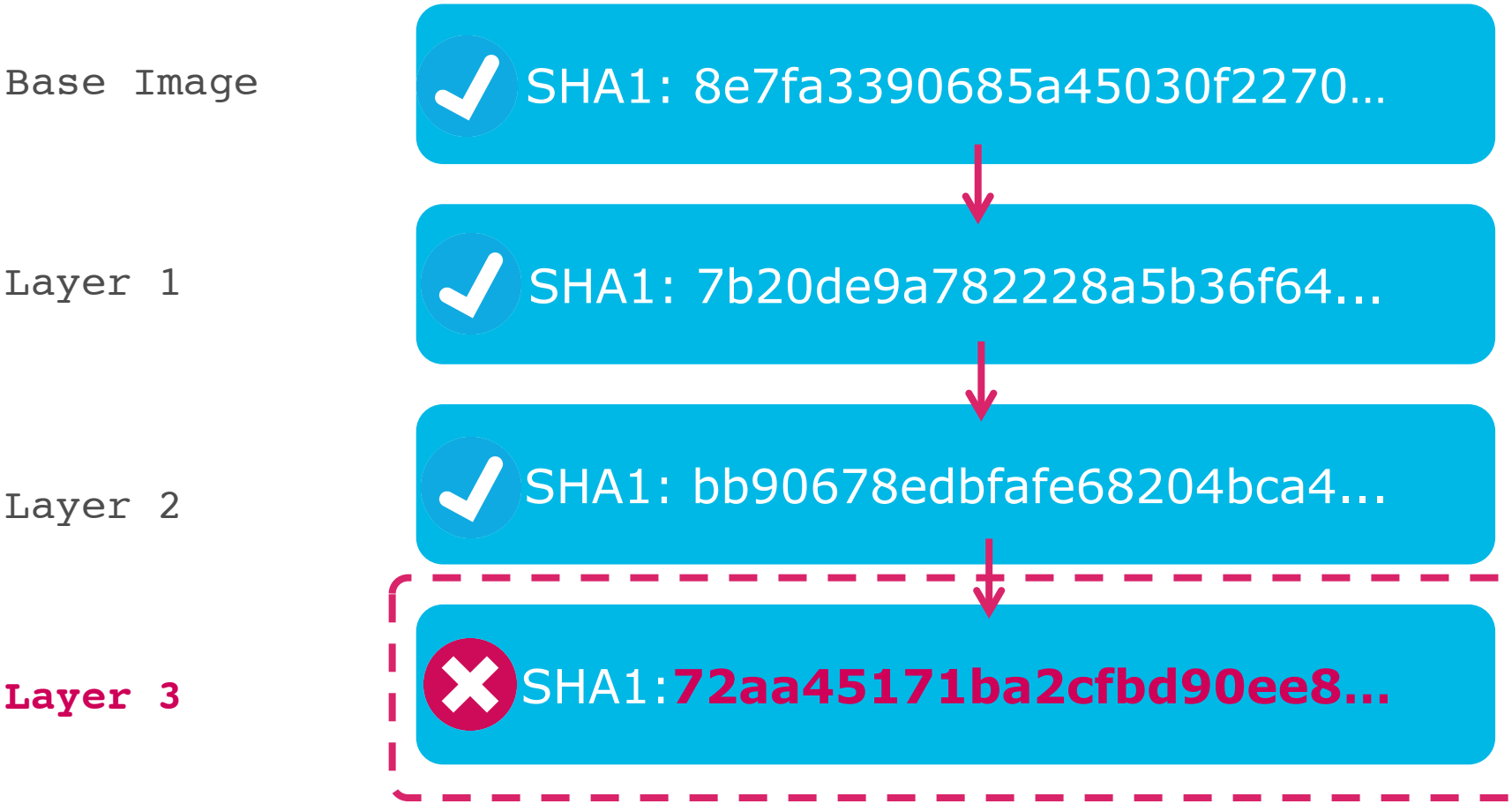
Tree structure



amadeus

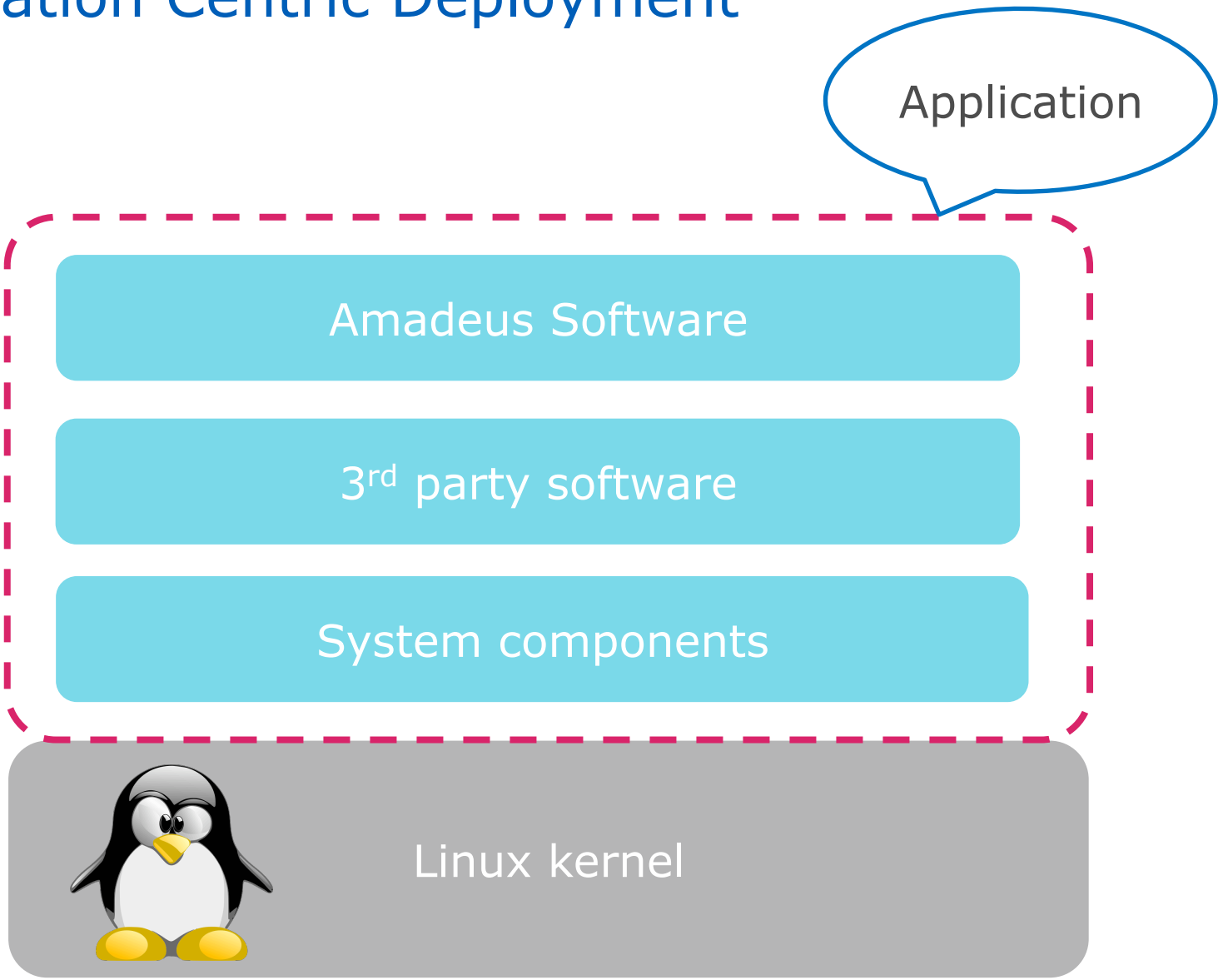
Docker Image format

Tree structure



amadeus

Application Centric Deployment

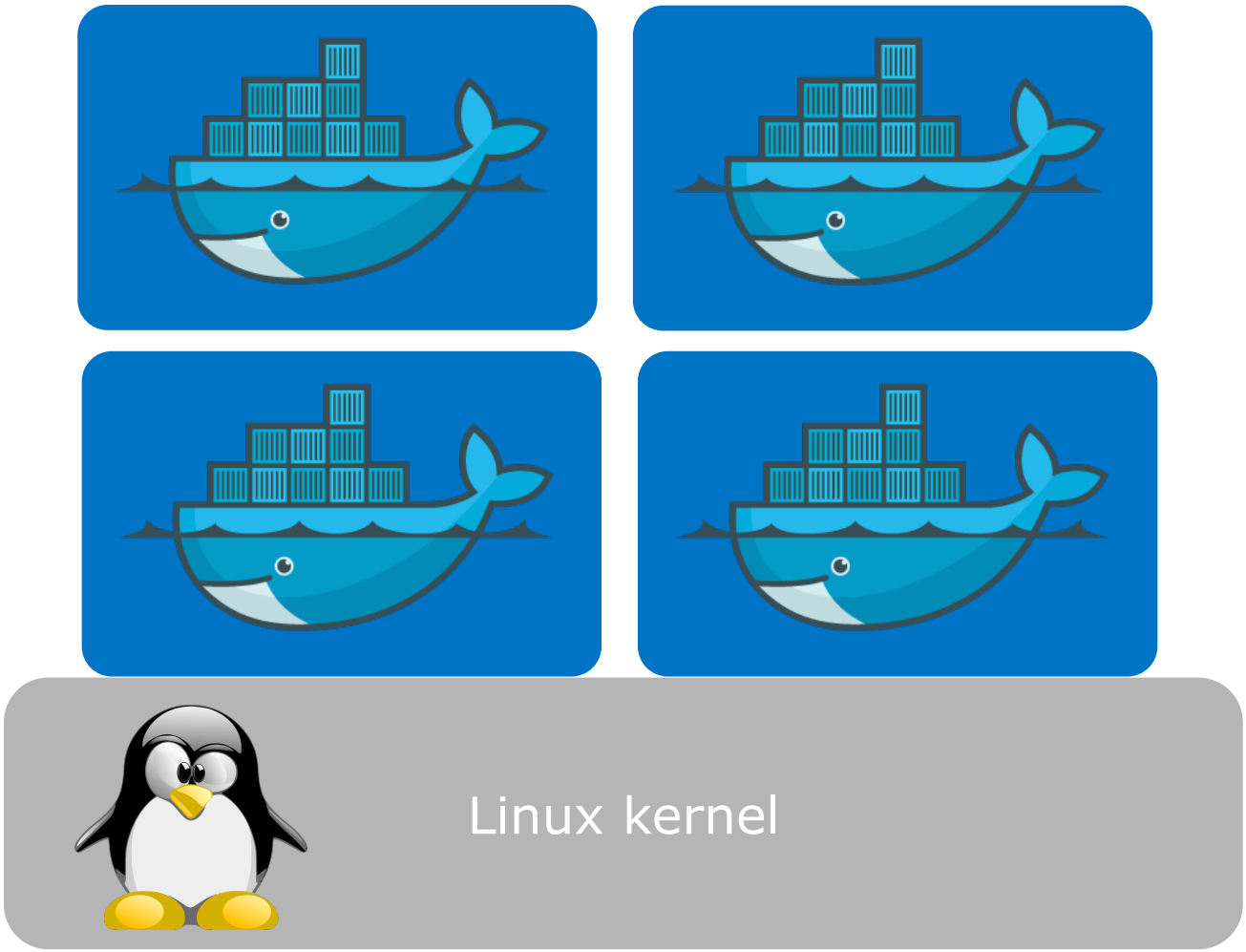


amadeus

Application Centric Deployment

amadeus

Application Centric Deployment



amadeus

Kubernetes

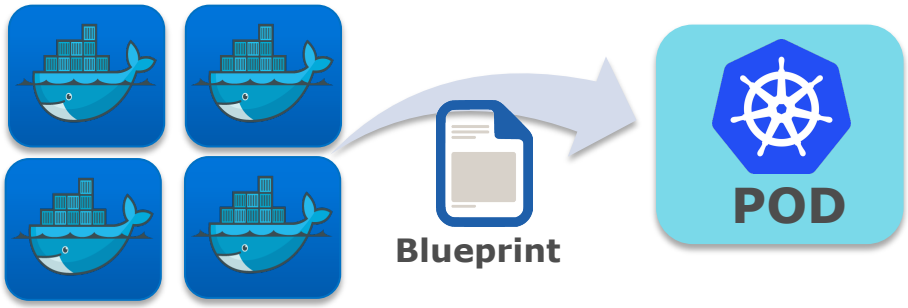
"Manage a cluster of Linux containers as a single system"

_Cluster

- Compute resources
- Virtual or physical machines

_Pods

- Collocated group of containers
- Smallest deployment unit
- **Scheduling and scaling unit**



amadeus

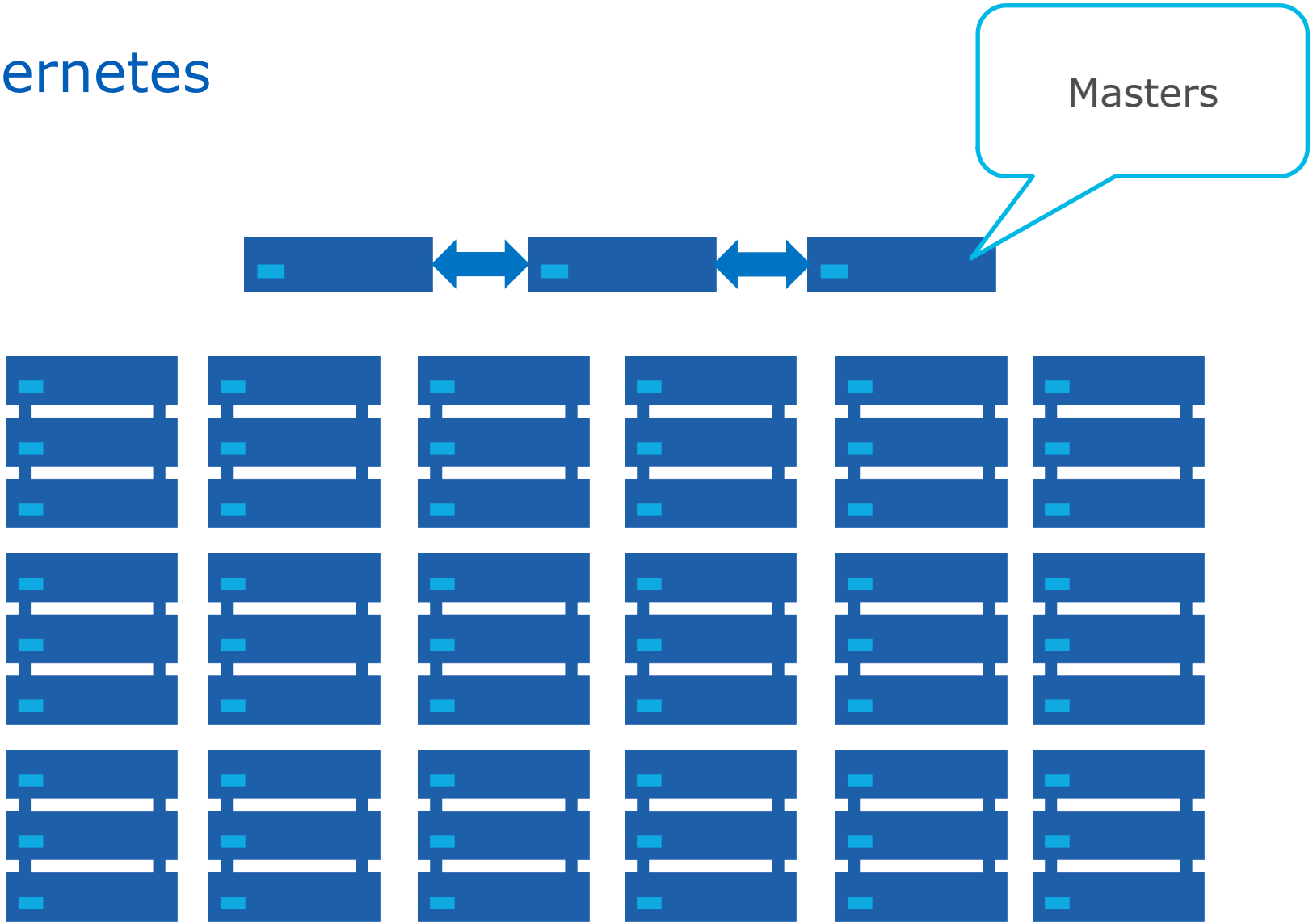
Kubernetes & Pods

Pods placement is automatic, resource based and decided by Kubernetes

Service discovery is used to know what is running where

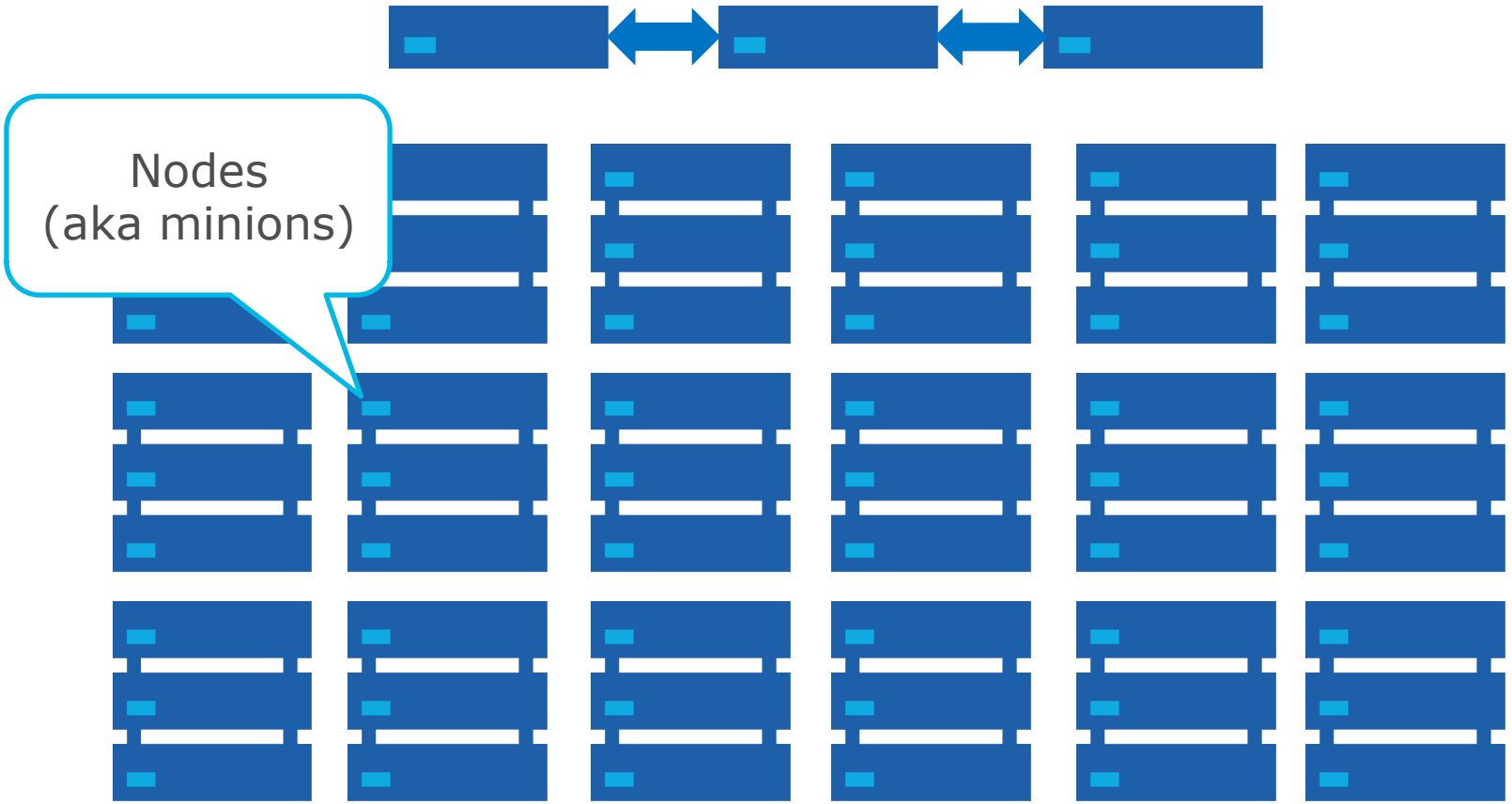
amadeus

Kubernetes



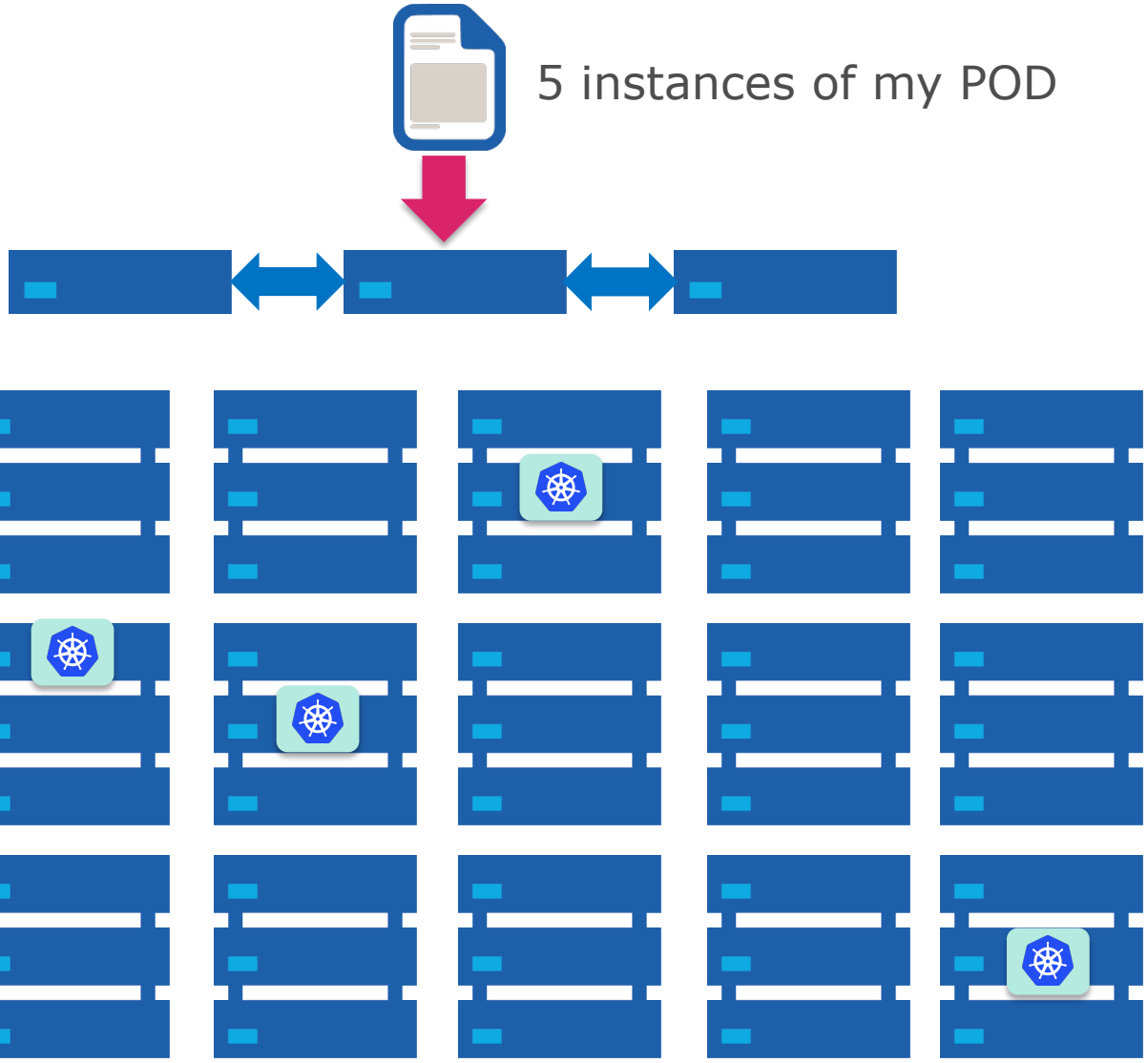
amadeus

Kubernetes



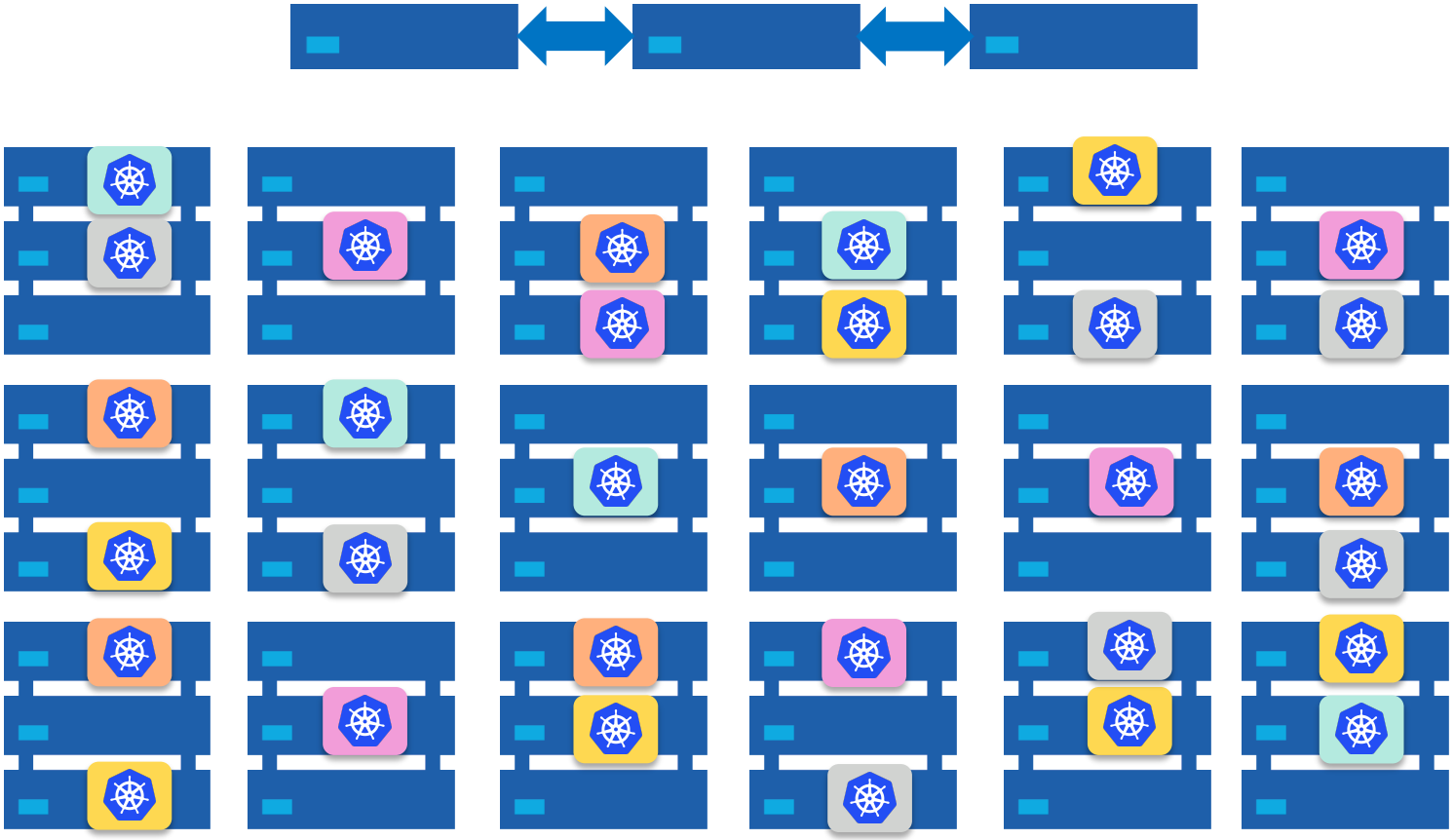
amadeus

Kubernetes



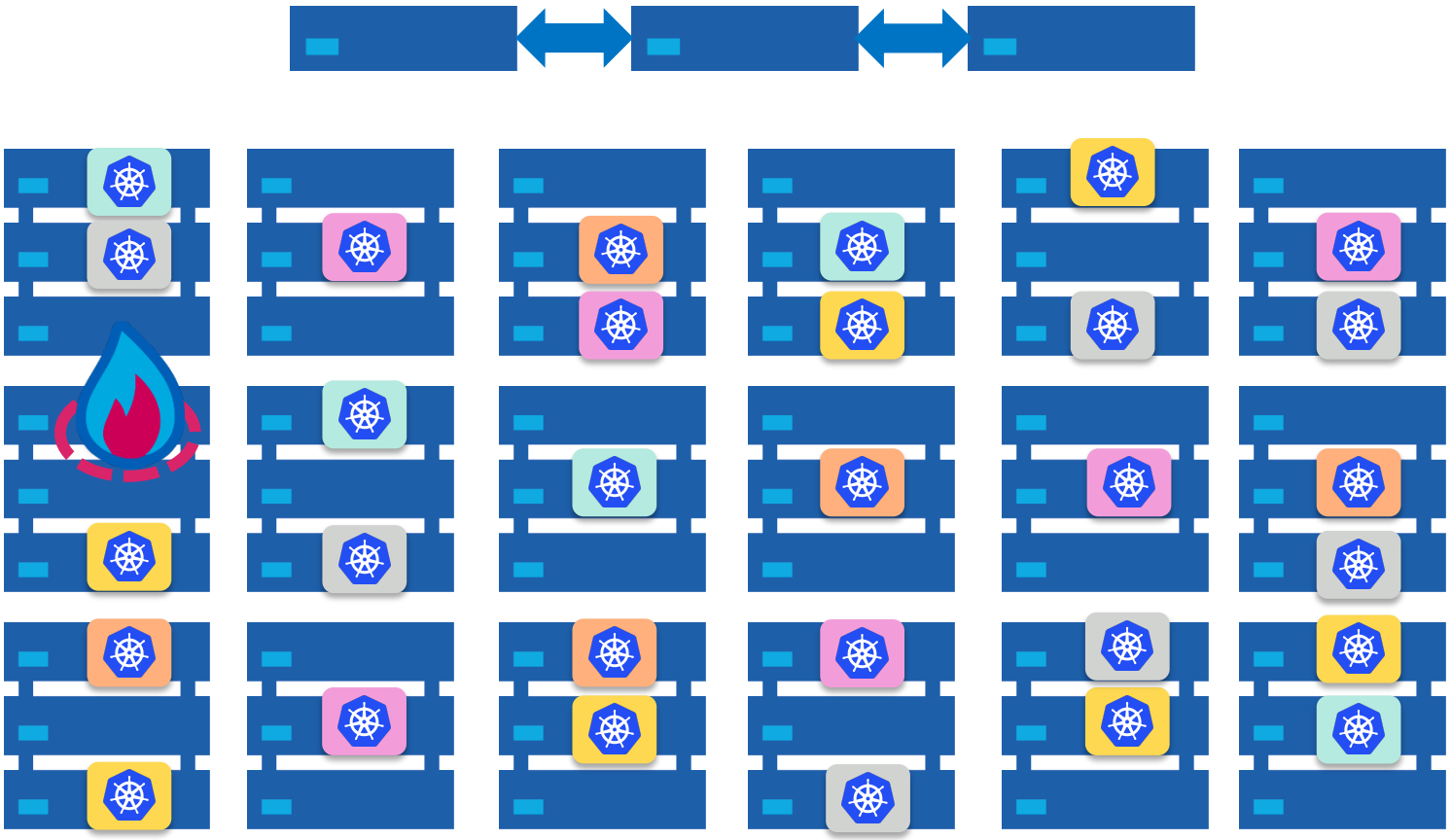
amadeus

Kubernetes



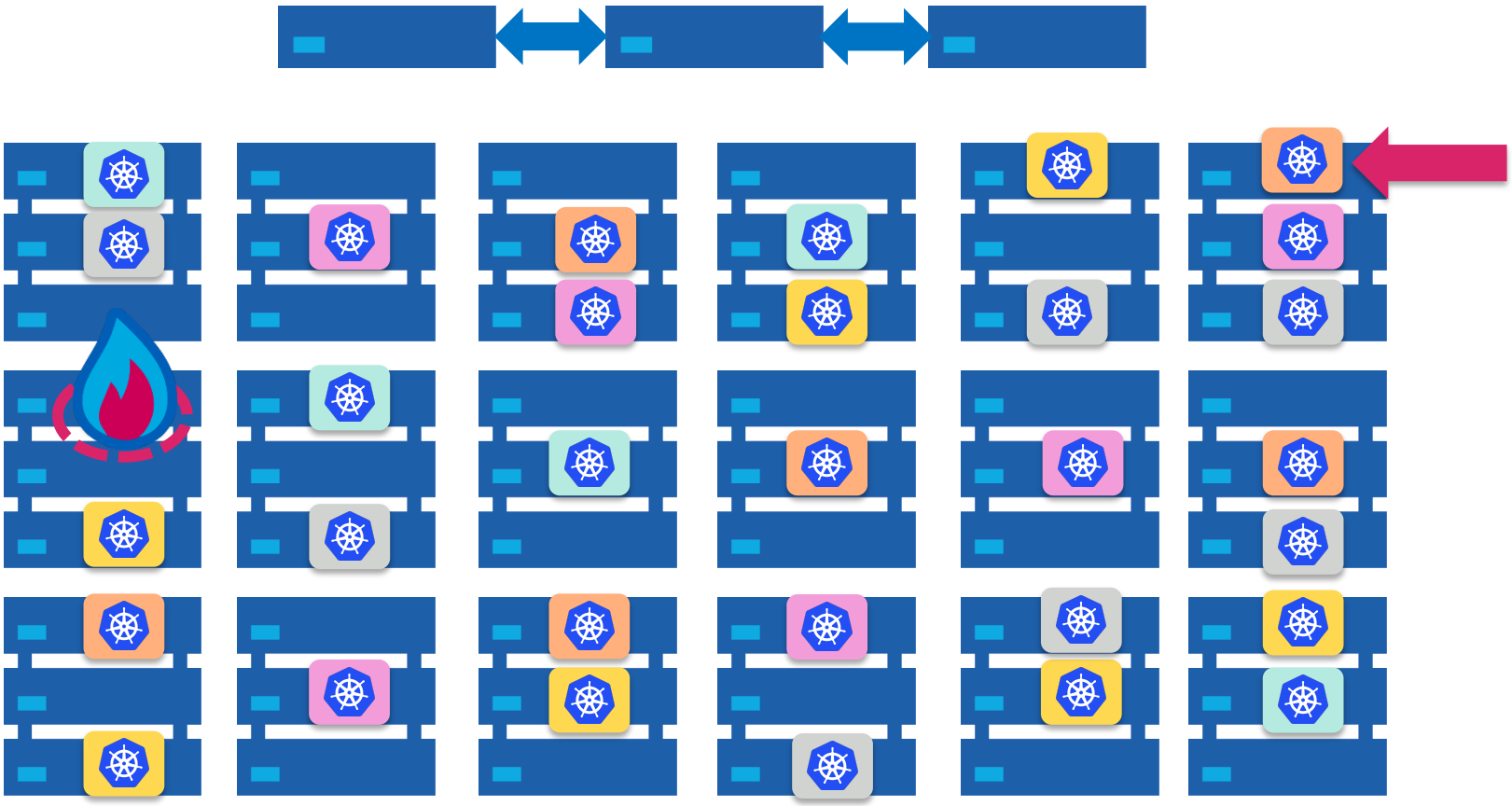
amadeus

Kubernetes



amadeus

Kubernetes



amadeus

Pods

Always same content

Always listening on same ports

amadeus

Pods

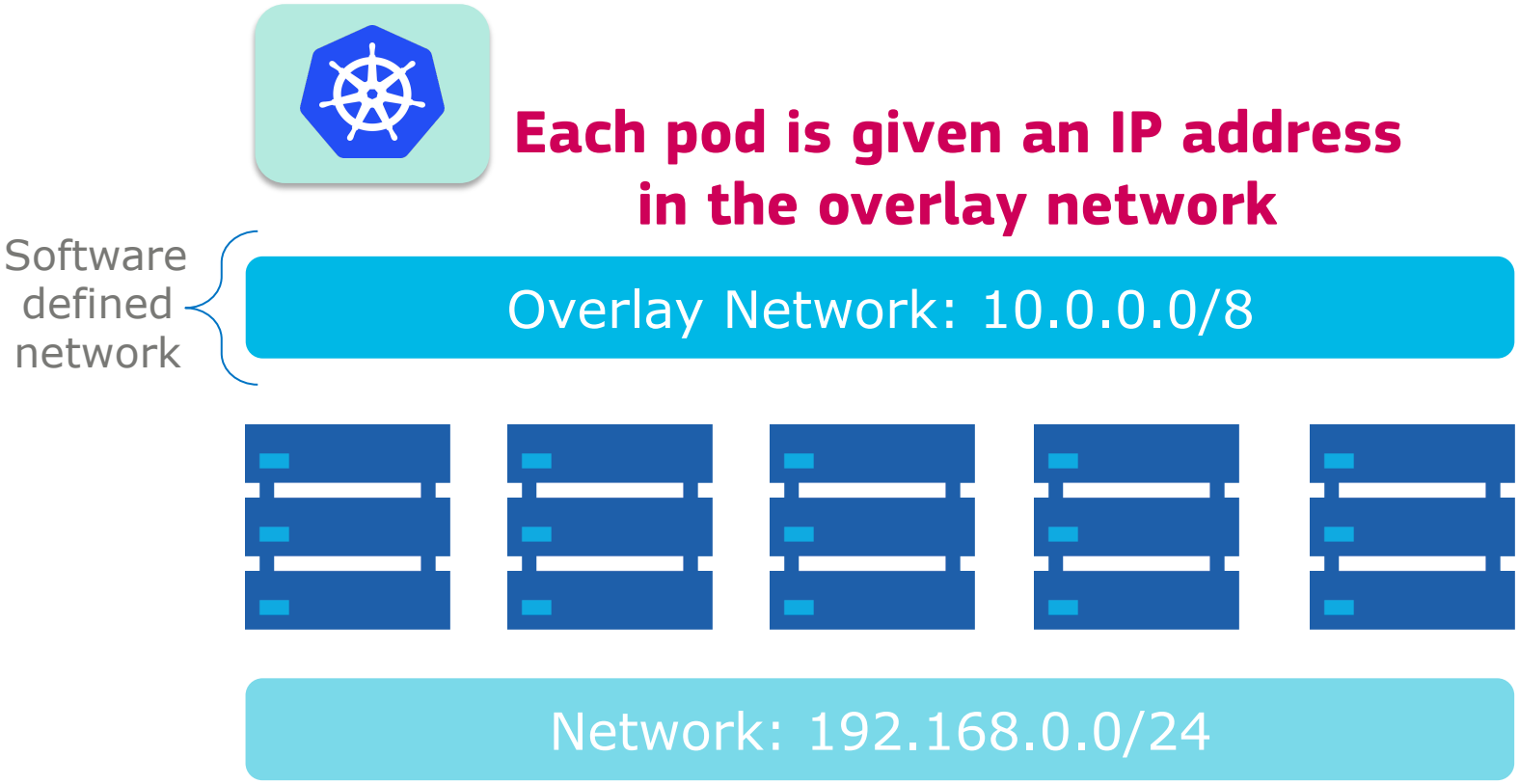
Always same content

Always listening on same ports

amadeus

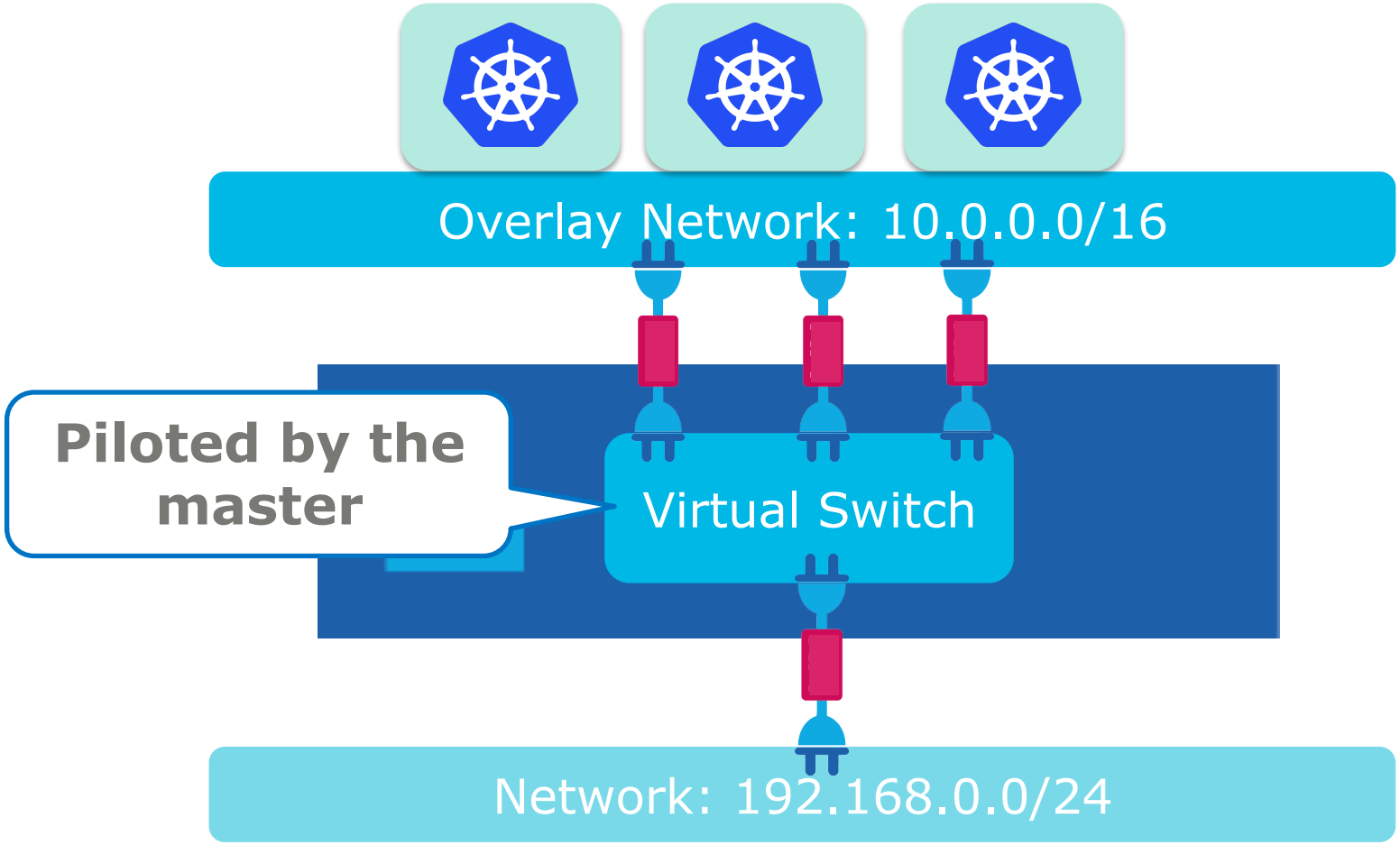
Pod addressing

SDN



amadeus

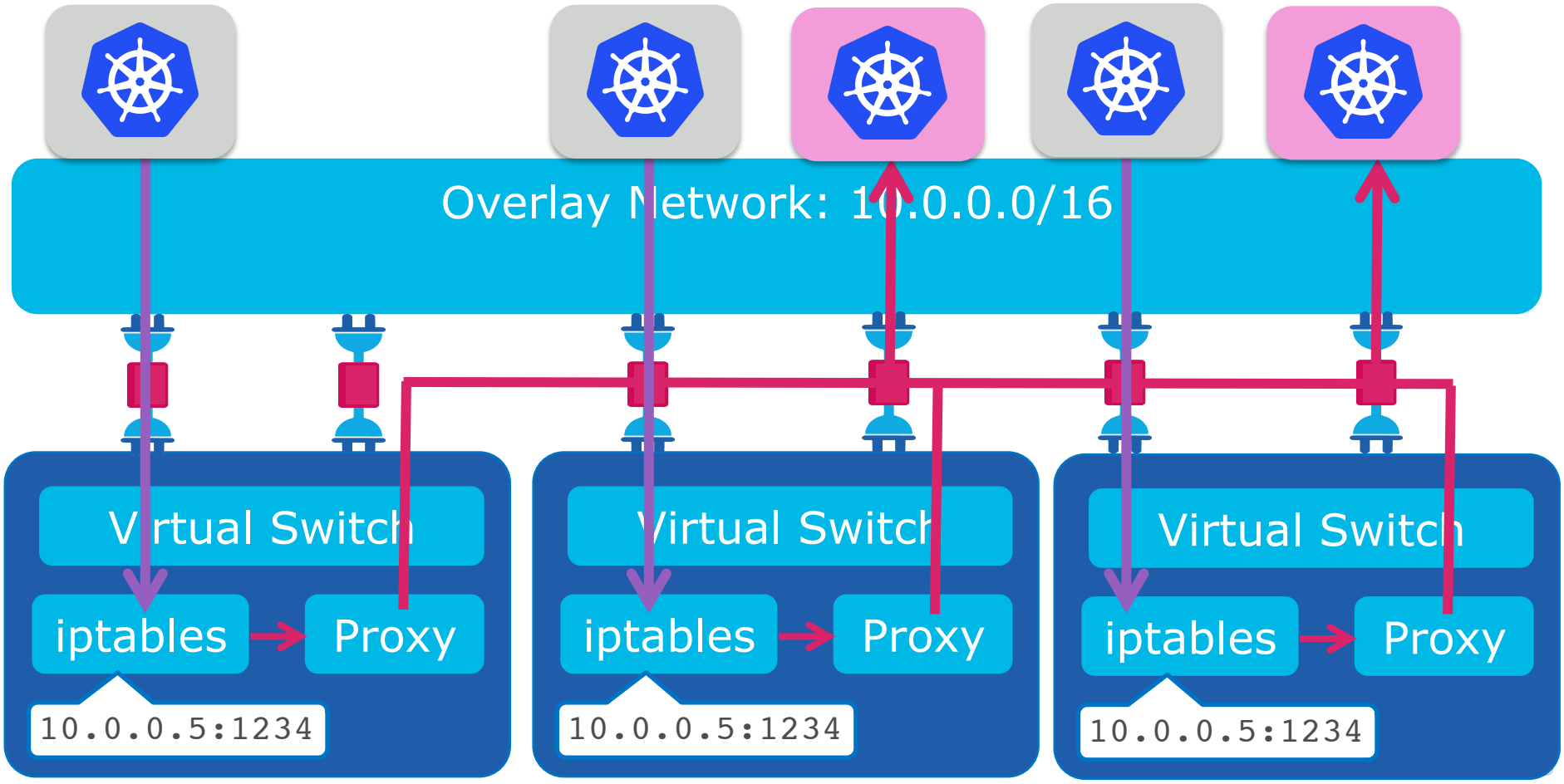
Pod addressing



amadeus

Service addressing

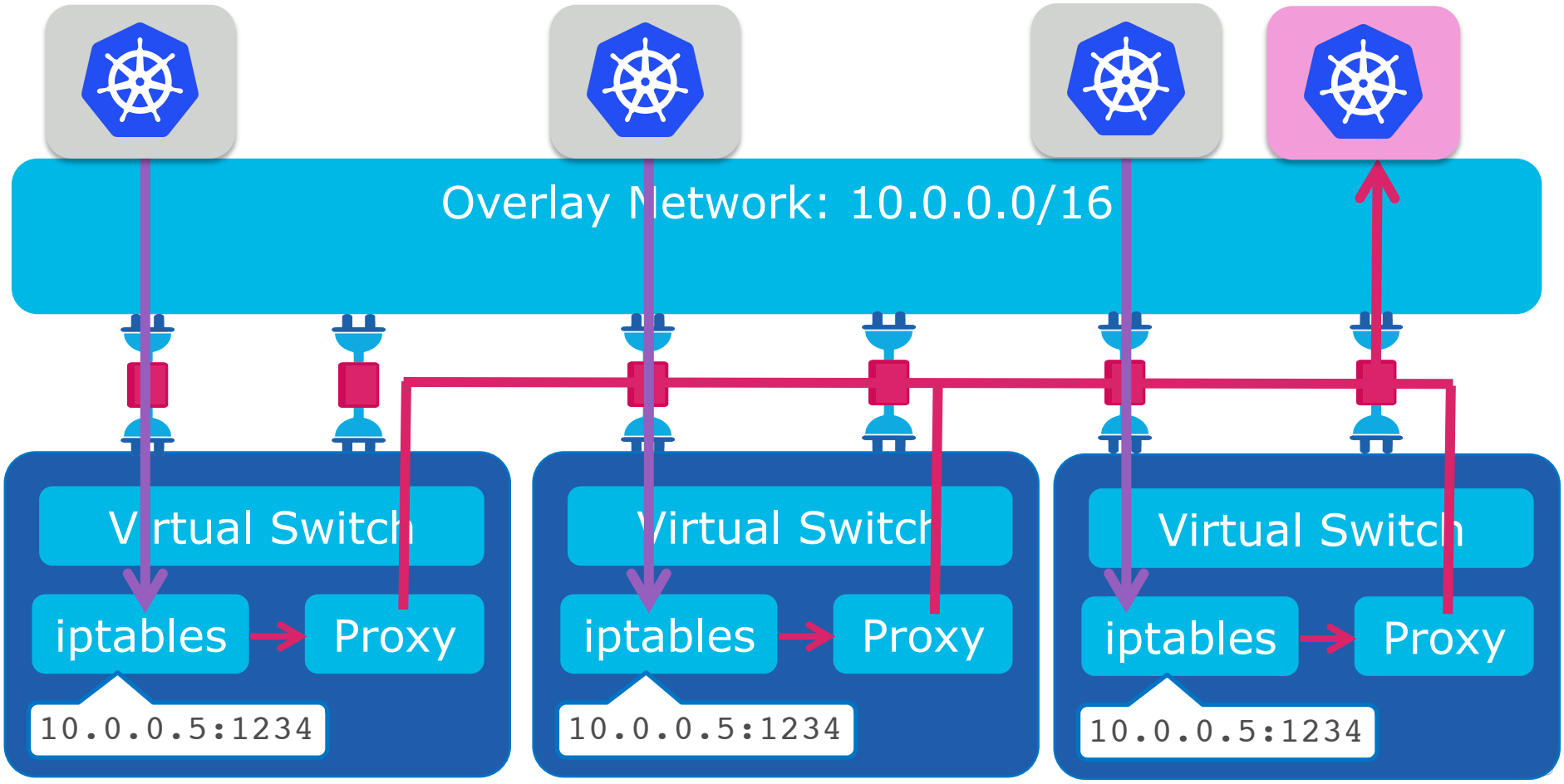
Service “S1”
IP/port: 10.0.0.5:1234



amadeus

Service addressing

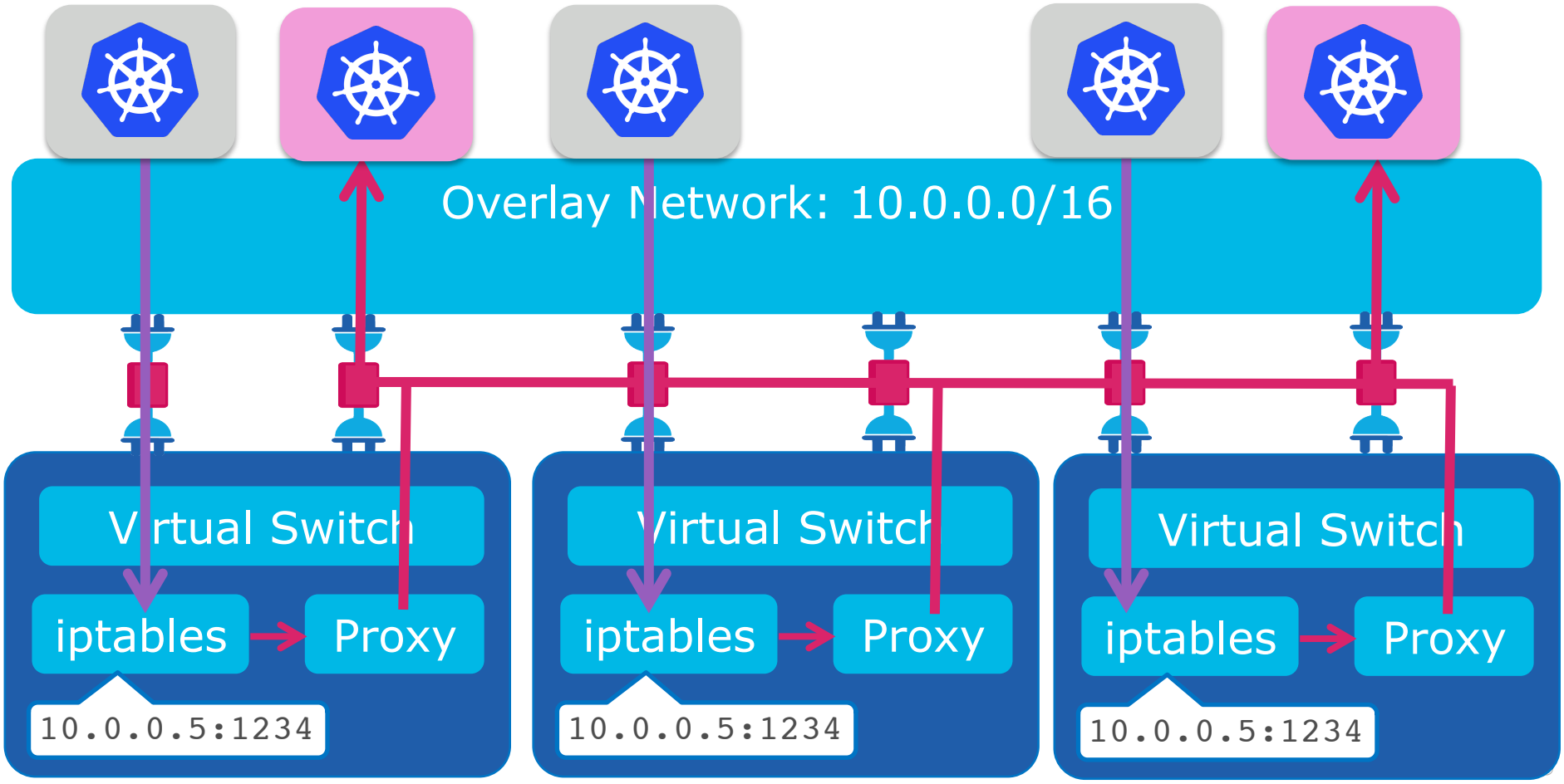
Service “S1”
IP/port: 10.0.0.5:1234



amadeus

Service addressing

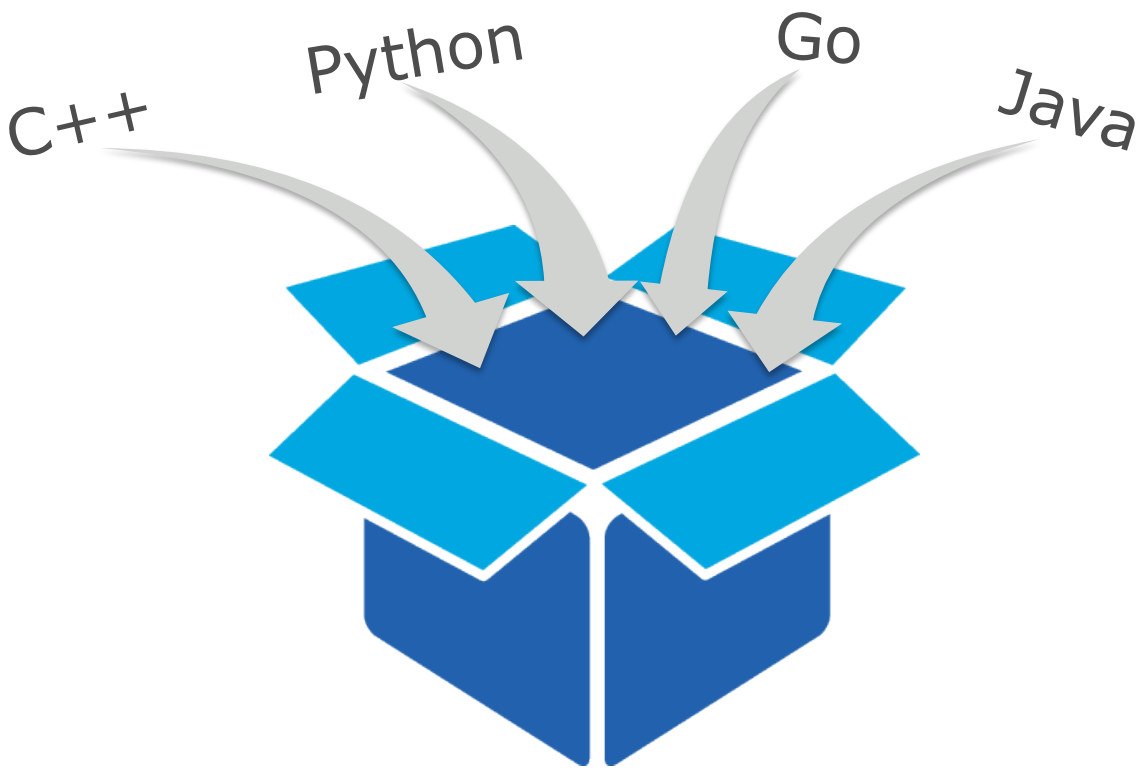
Service “S1”
IP/port: 10.0.0.5:1234



amadeus

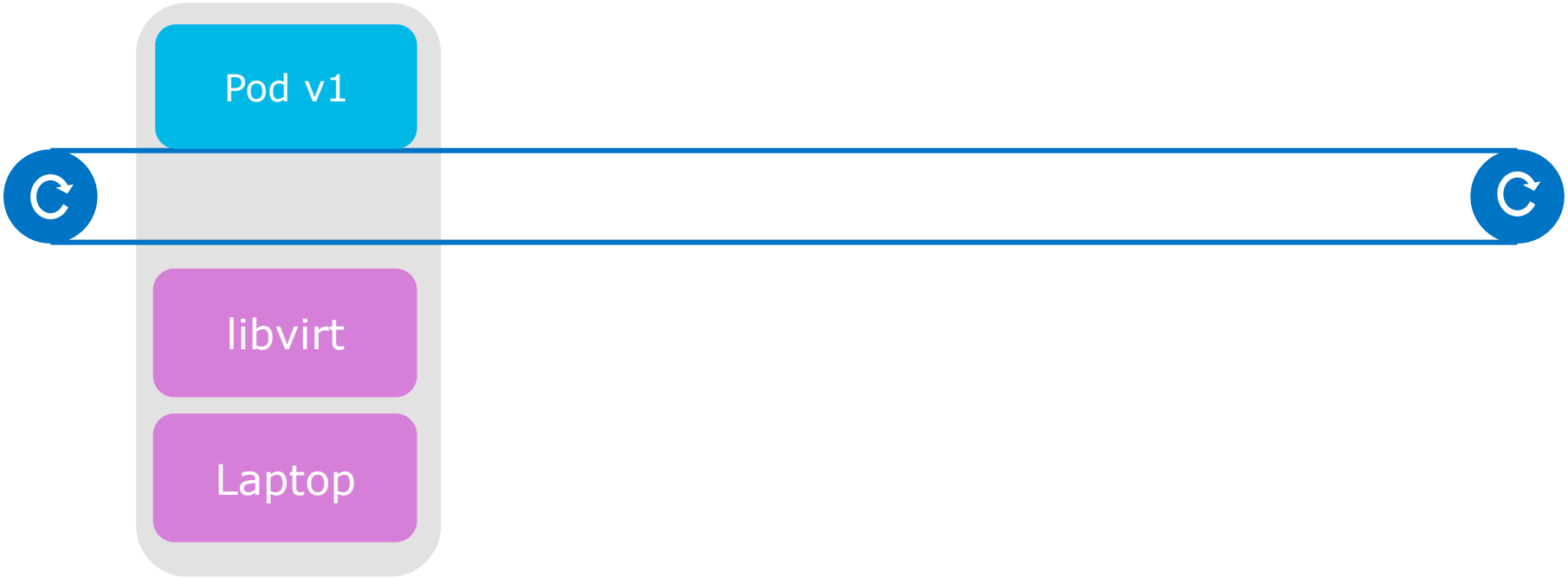
Operational Model

- Homogenization: everything looks the same



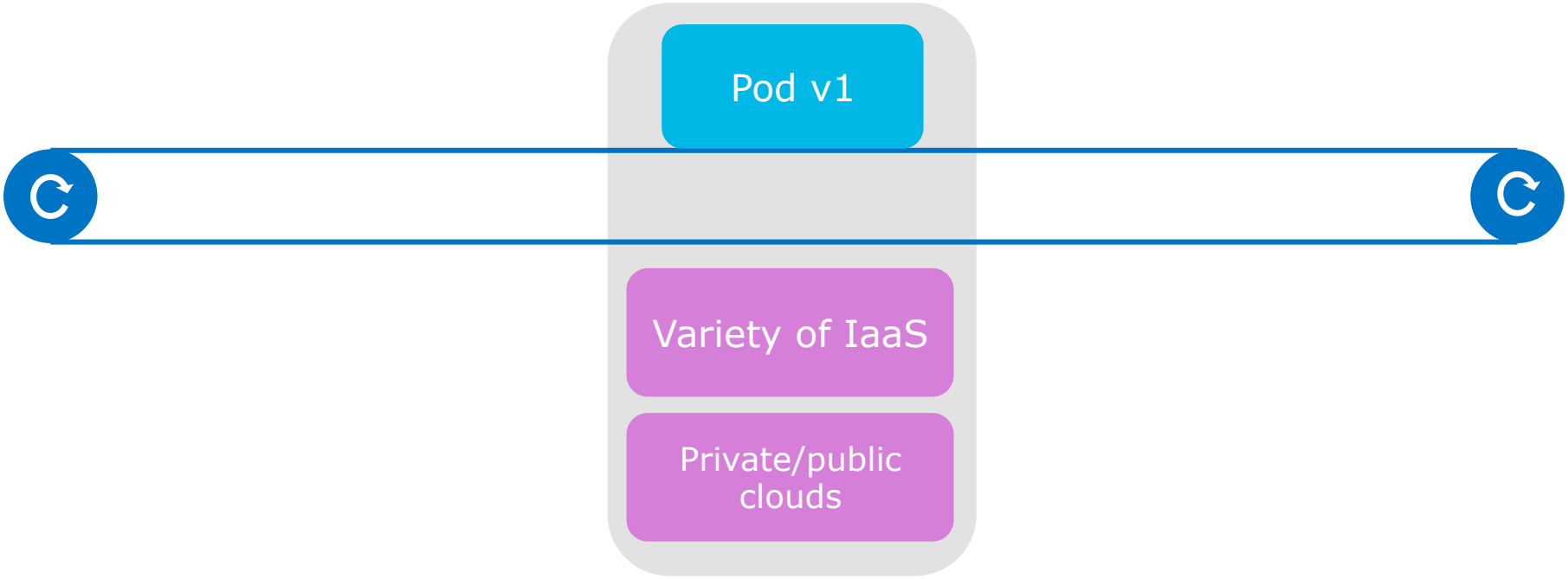
amadeus

Easy Deployment



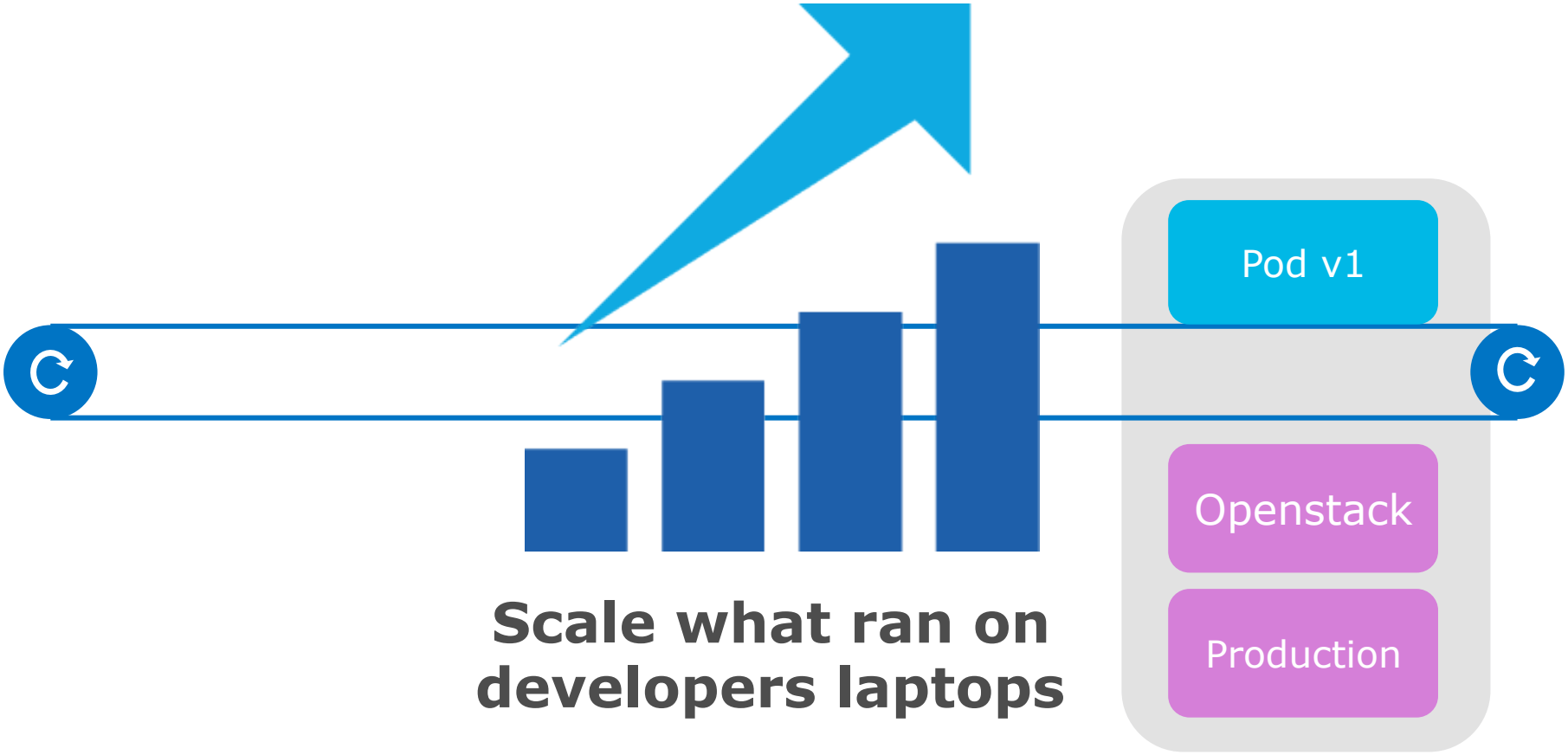
amadeus

Easy Deployment



amadeus

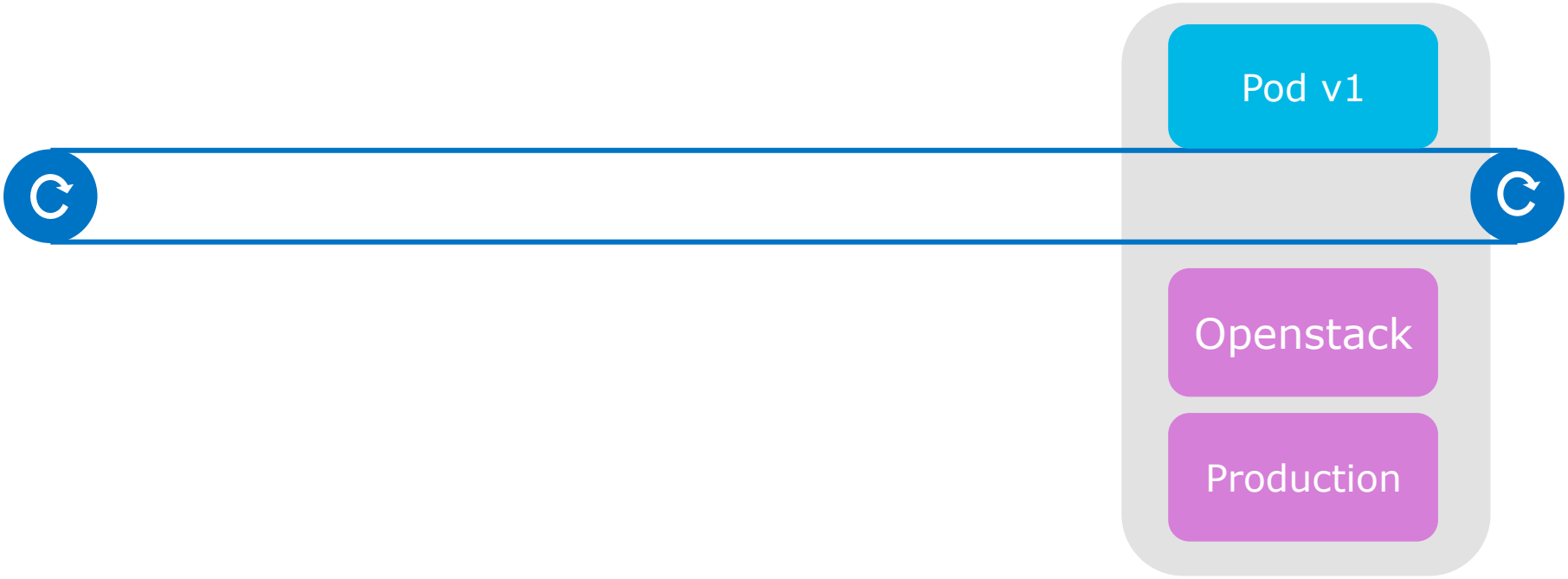
Easy Deployment



Scale what ran on developers laptops

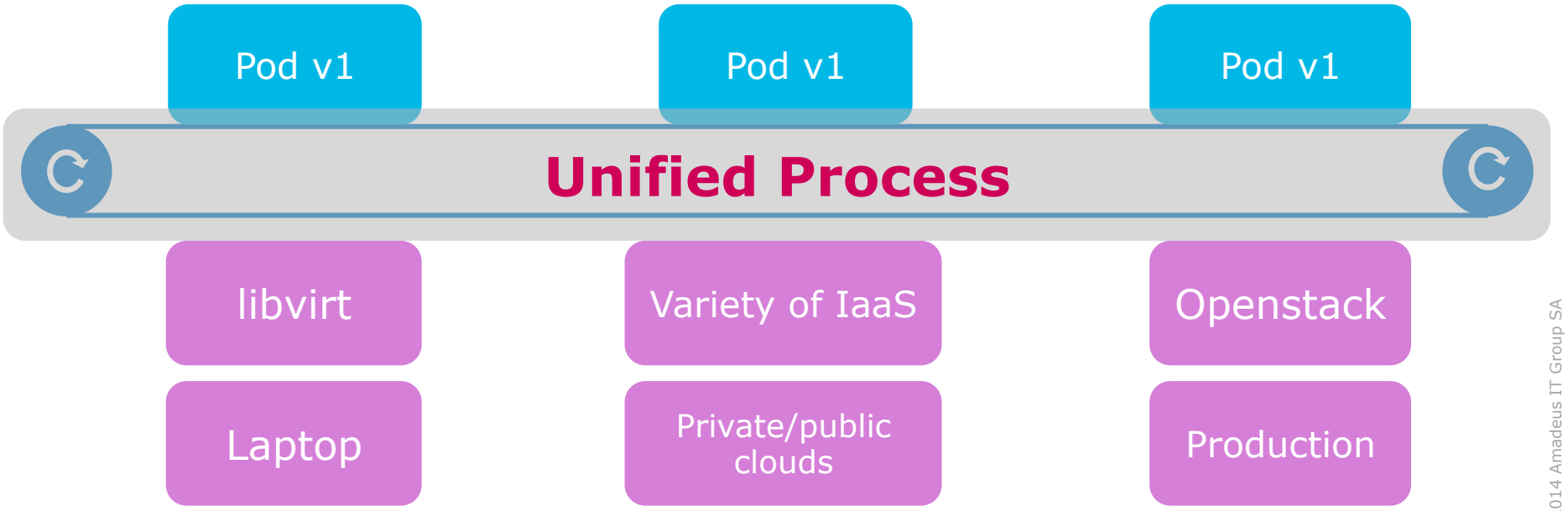
amadeus

Easy Deployment



amadeus

Easy Deployment



amadeus

Key Takeaways

- Building a cloud-ready system with **OPEN**SHIFT
  - Macro-management
  - Self-healing / resiliency
  - Dynamic system
  - Multi-cloud/data-center capability

- Uniform operations
  - Simplification
  - Reduced operational burden
  - Enable transition to DevOps model

amadeus

Thank you

You can follow us on:
AmadeusITGroup

amadeus.com/blog
amadeus.com

amadeus