

Container Orchestration



Outline

- Kenapa diperlukan container orchestration?
- Container Orchestration
- Tools Container Orchestration

**Kenapa diperlukan
container orchestration ?**

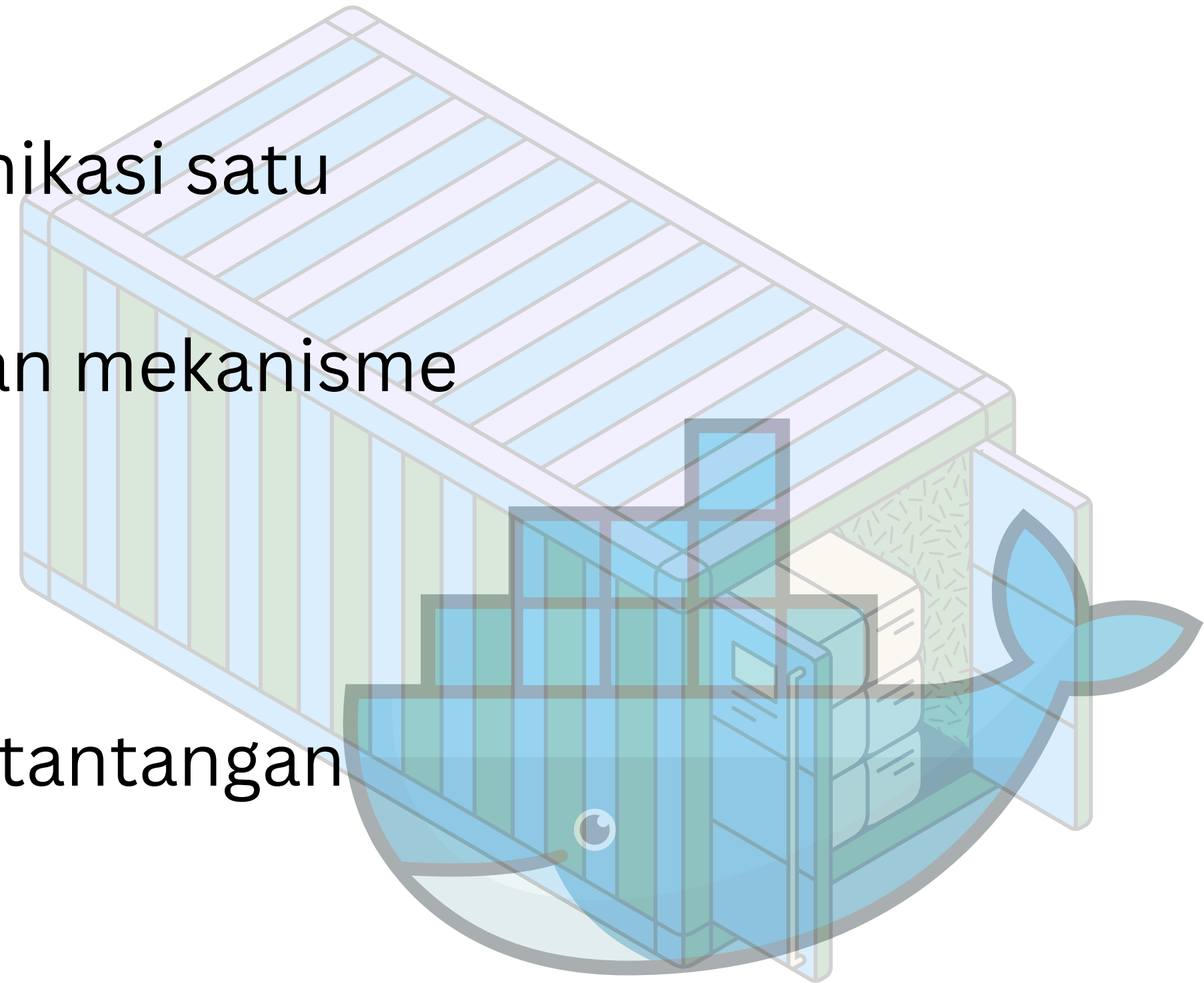
Baik linux container ataupun docker container memiliki kelebihan dapat mengisolasi aplikasi dari host



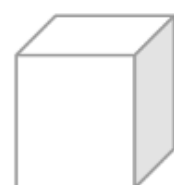
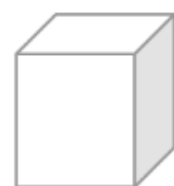
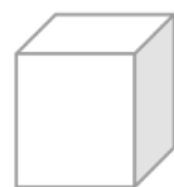
Namun container tidak mudah di scalable

Masalah scaling pada container

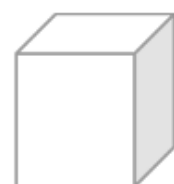
- Container tidak dapat berkomunikasi satu dengan yang lain
- Container perlu di deploy dengan mekanisme tepat
- Container perlu di manage
- auto scaling tidak dimungkinkan
- Distribusi traffic masih menjadi tantangan tersendiri



run
Services
in
Containers



⋮



without **container orchestration**

scaling up services
→ manual work
increases

fixing crashing nodes
→ manual work
increases

complexity for running
new stuff in production
increases

human cost for
running services
increases

scaling becomes
more and more
difficult

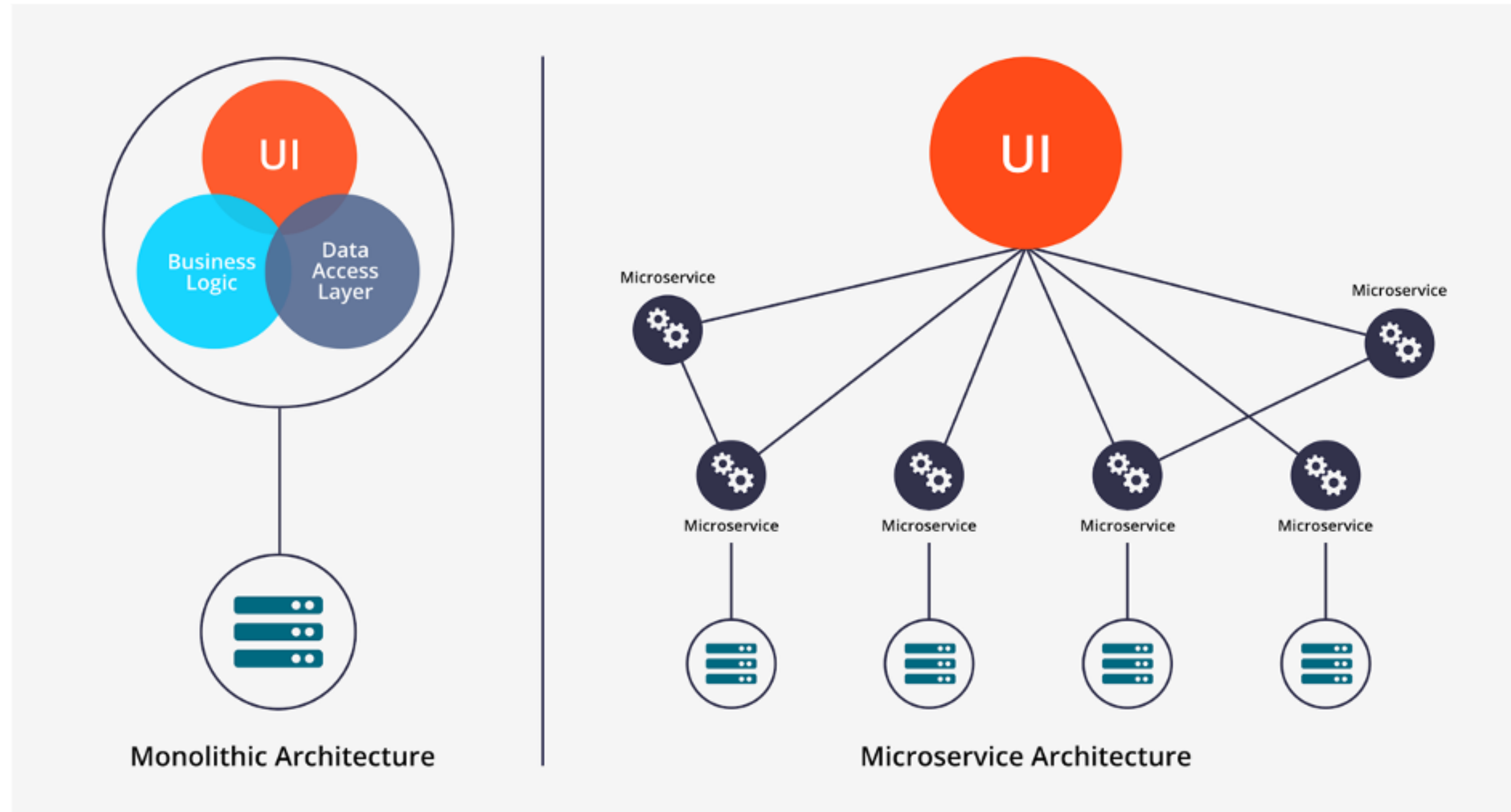
public cloud providers
bill more and more
expensive

Kenapa diperlukan orchestration?

- Karena Container ringan, menjalankan di lingkungan production akan membutuhkan effort lebih.
- Ketika menggunakan micro services, akan ada ratusan atau ribuan container berjalan dan dioperasikan secara skala besar
- Hal ini dapat menimbulkan kompleksitas yang signifikan jika dikelola secara manual.
- Orkestrasi container digunakan untuk menyederhanakan operasional dan development yang kompleks menjadi lebih ter-otomatisasi.

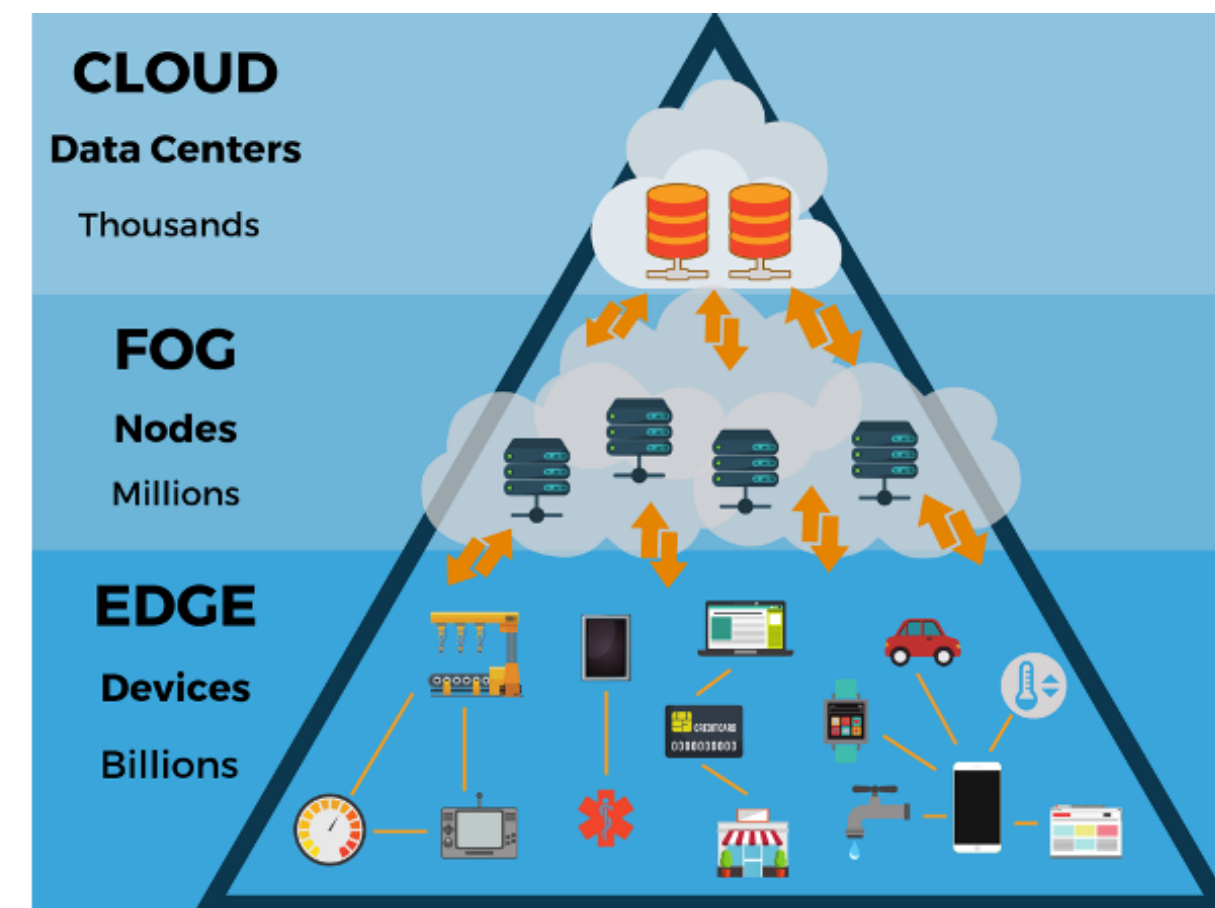
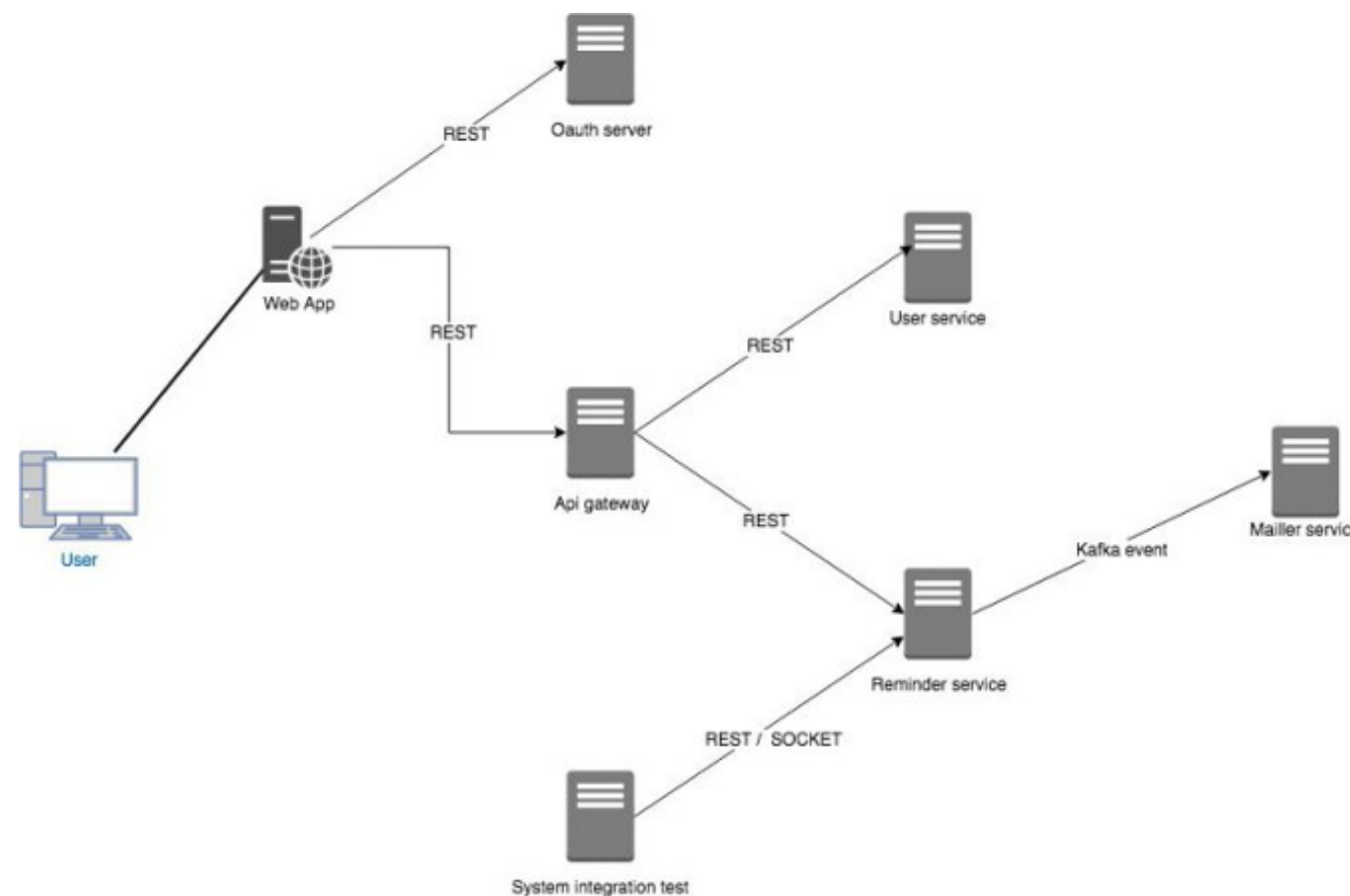


Microservices



Kenapa microservices ?

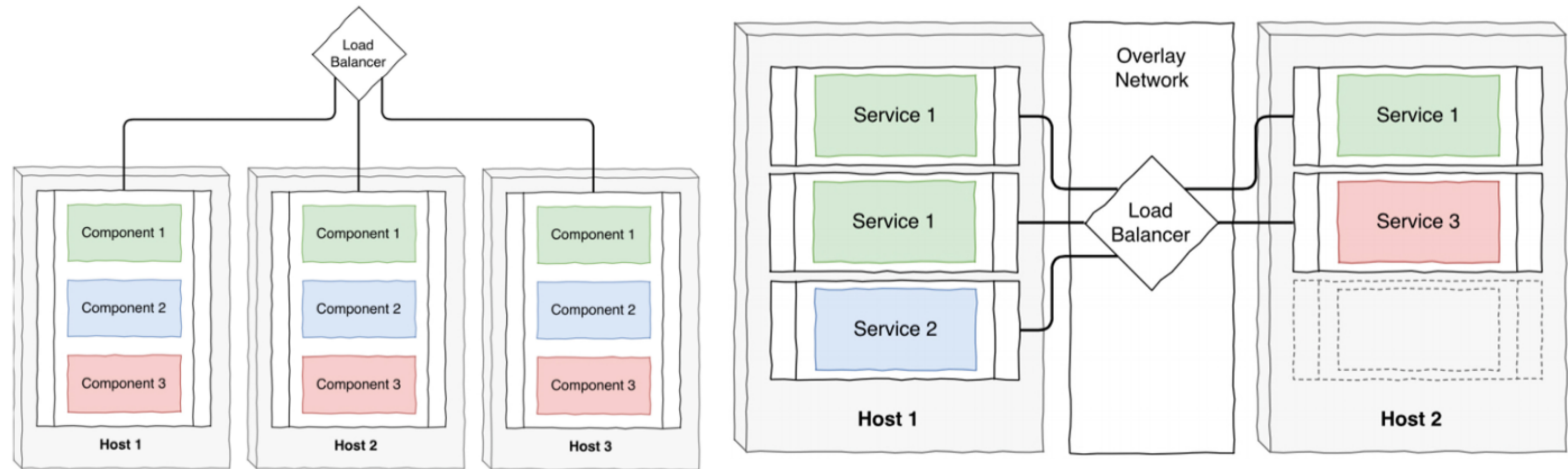
Skalabilitas Distribusi



Pada microservice anda tidak memerlukan server yang sangat powerfull. anda dapat menggunakan beberapa server dengan spesifikasi yang lebih rendah dan menjalankan services dari masing-masing aplikasi. System akan memiliki mekanis untuk memberikan performa dan membagi task secara terdistribusi.

Kenapa microservices ?

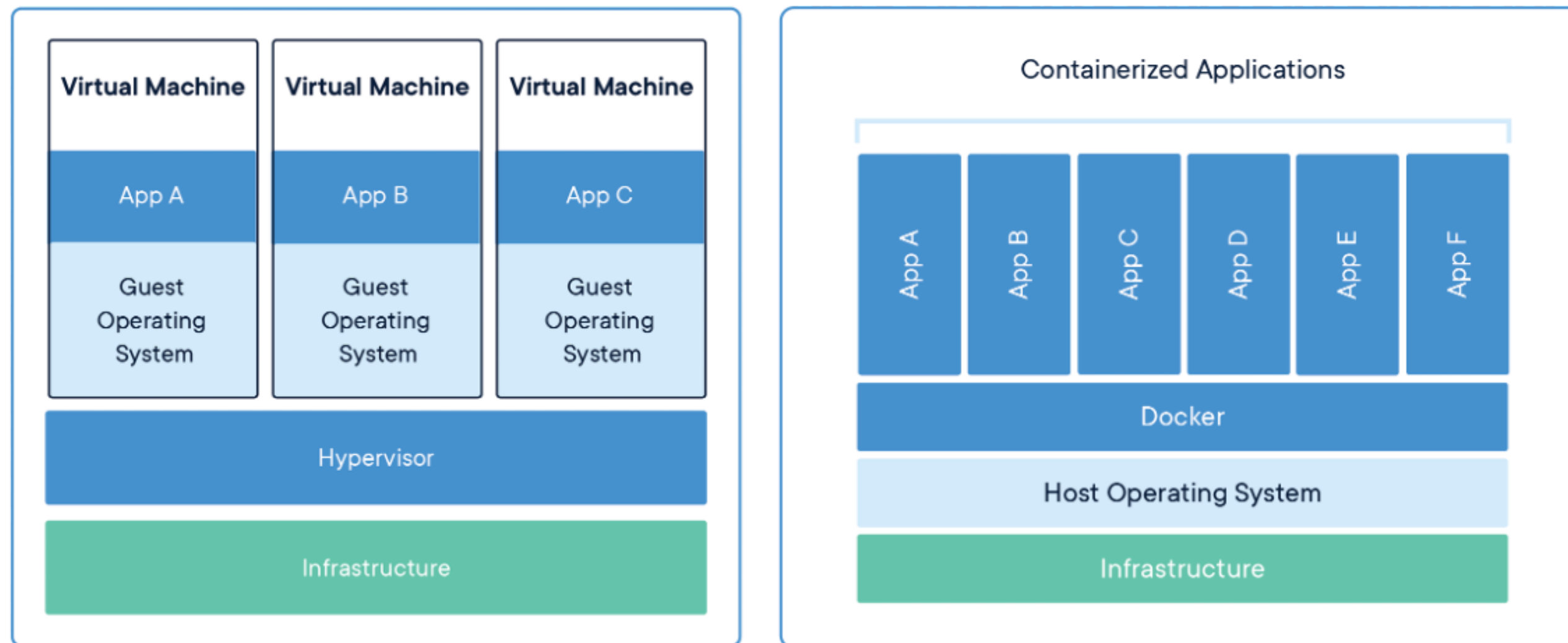
Skalabilitas



Dalam arsitektur Microservice, kita dapat melakukan non-uniform scaling dengan menggunakan beberapa instance dari layanan tertentu sehingga kita bisa mendapatkan utilisasi yang lebih tinggi dengan sumber daya server yang lebih sedikit.

Kenapa microservices ?

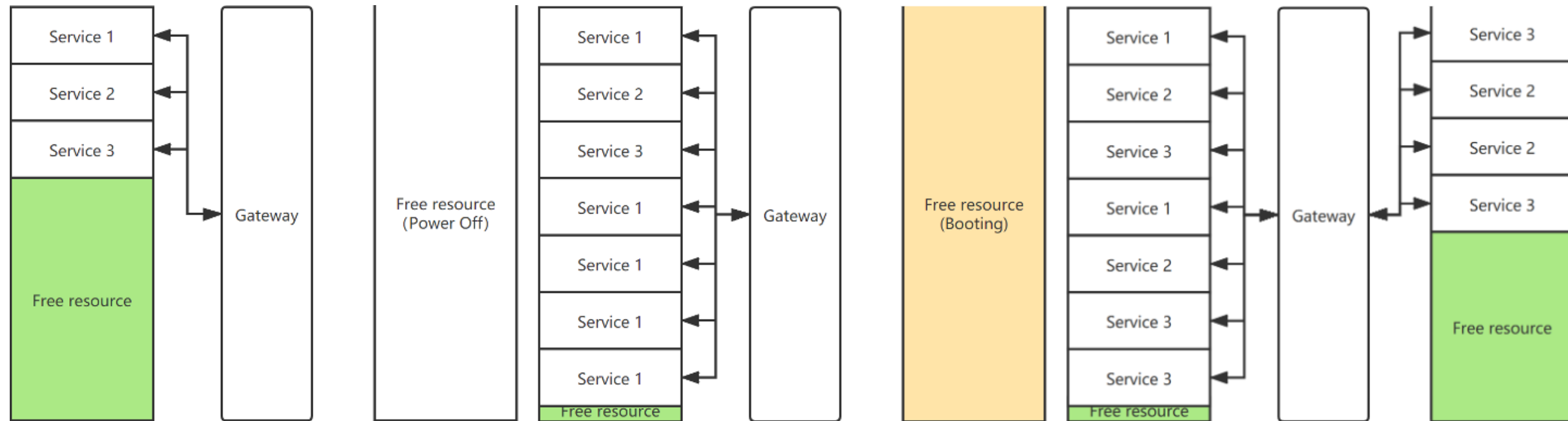
Portable



Dalam arsitektur Microservice, kita dapat melakukan non-uniform scaling dengan menggunakan beberapa instance dari layanan tertentu sehingga kita bisa mendapatkan utilisasi yang lebih tinggi dengan sumber daya server yang lebih sedikit.

Kenapa microservices ?

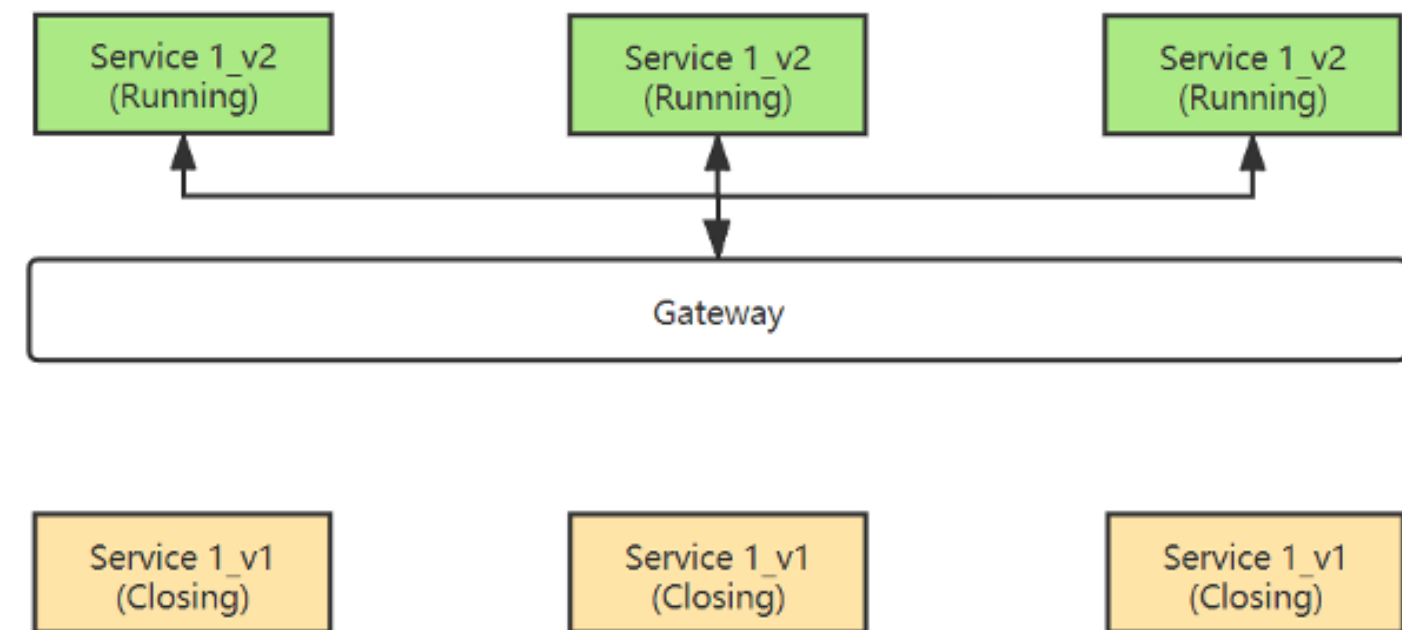
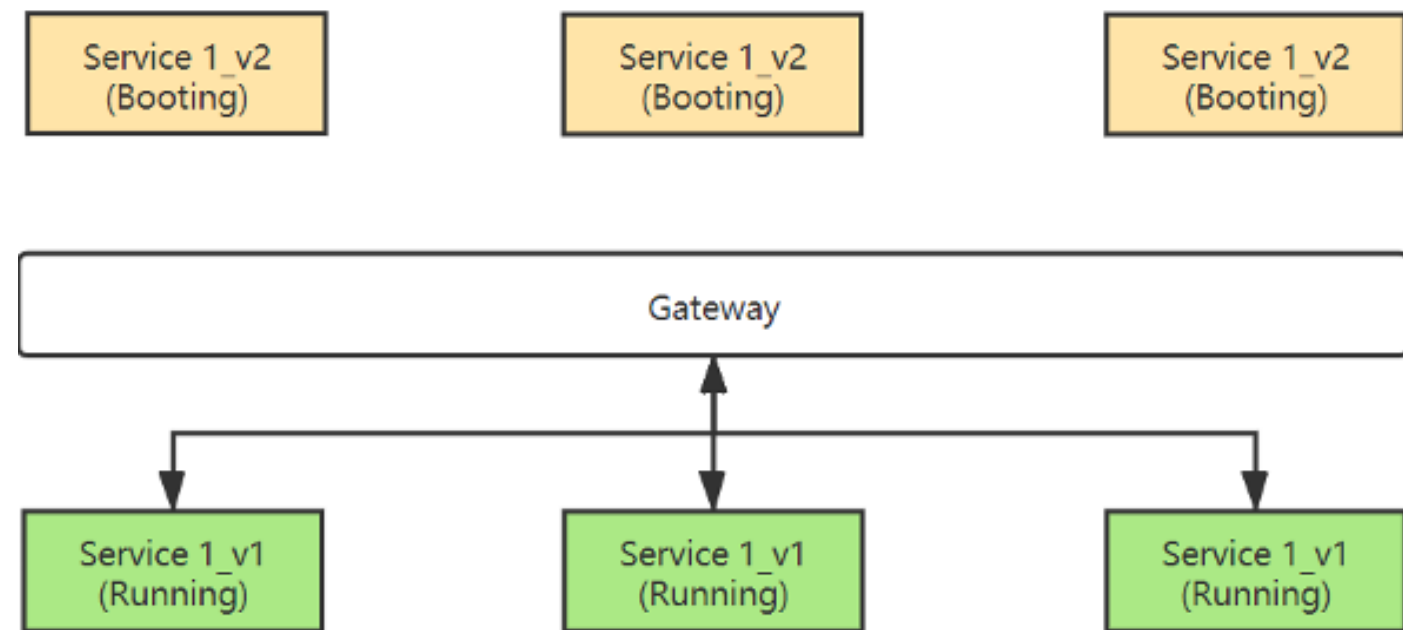
Elastis



Saat permintaan meningkat, server dapat secara otomatis deploy container baru untuk menyediakan layanan sesuai metrik sesuai waktu respons

Kenapa microservices ?

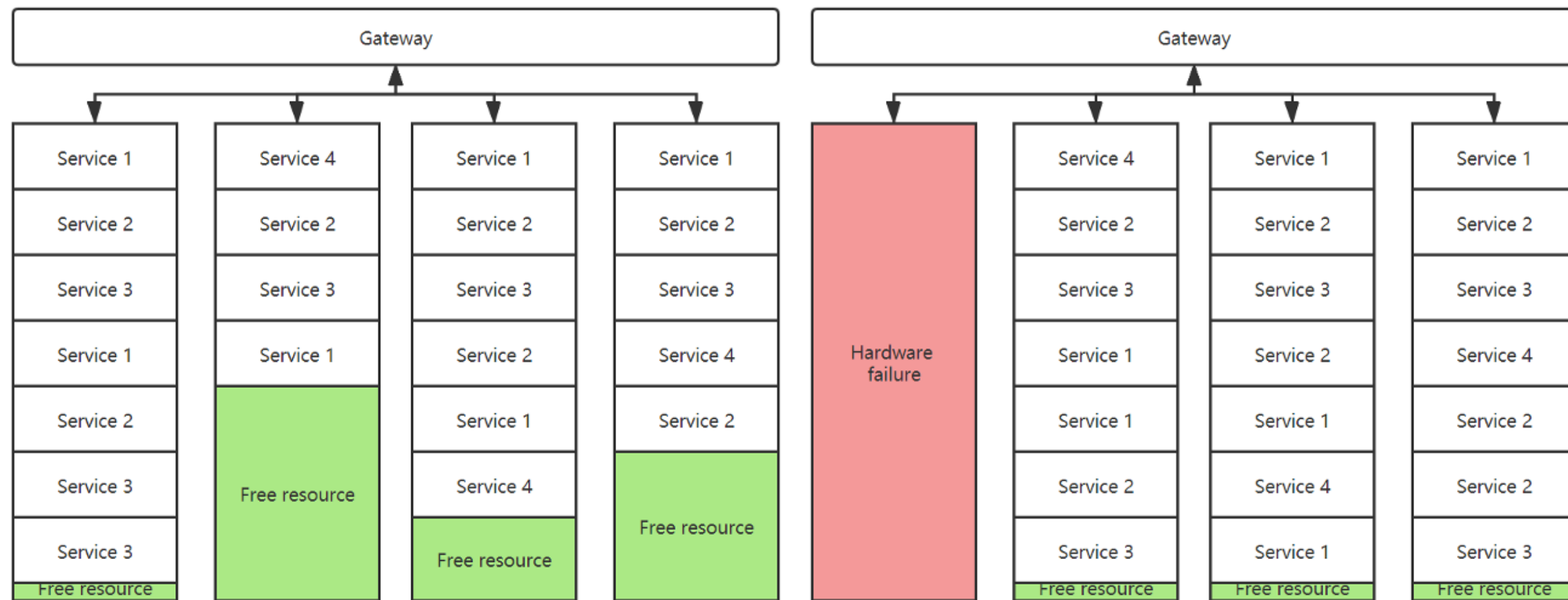
Availability



Saat memperbarui aplikasi, tidak perlu menghentikan seluruh aplikasi dan kemudian update versi yang baru.
dengan menggunakan arsitektur microservice, kita hanya membutuhkan sedikit resource untuk service tambahan yang di update

Kenapa microservices ?

Kehandalan (robust)



Jika satu server mengalami kegagalan perangkat keras seperti perangkat keras rusak atau mati listrik. server lain masih bisa menyediakan layanan

Container Orchestration

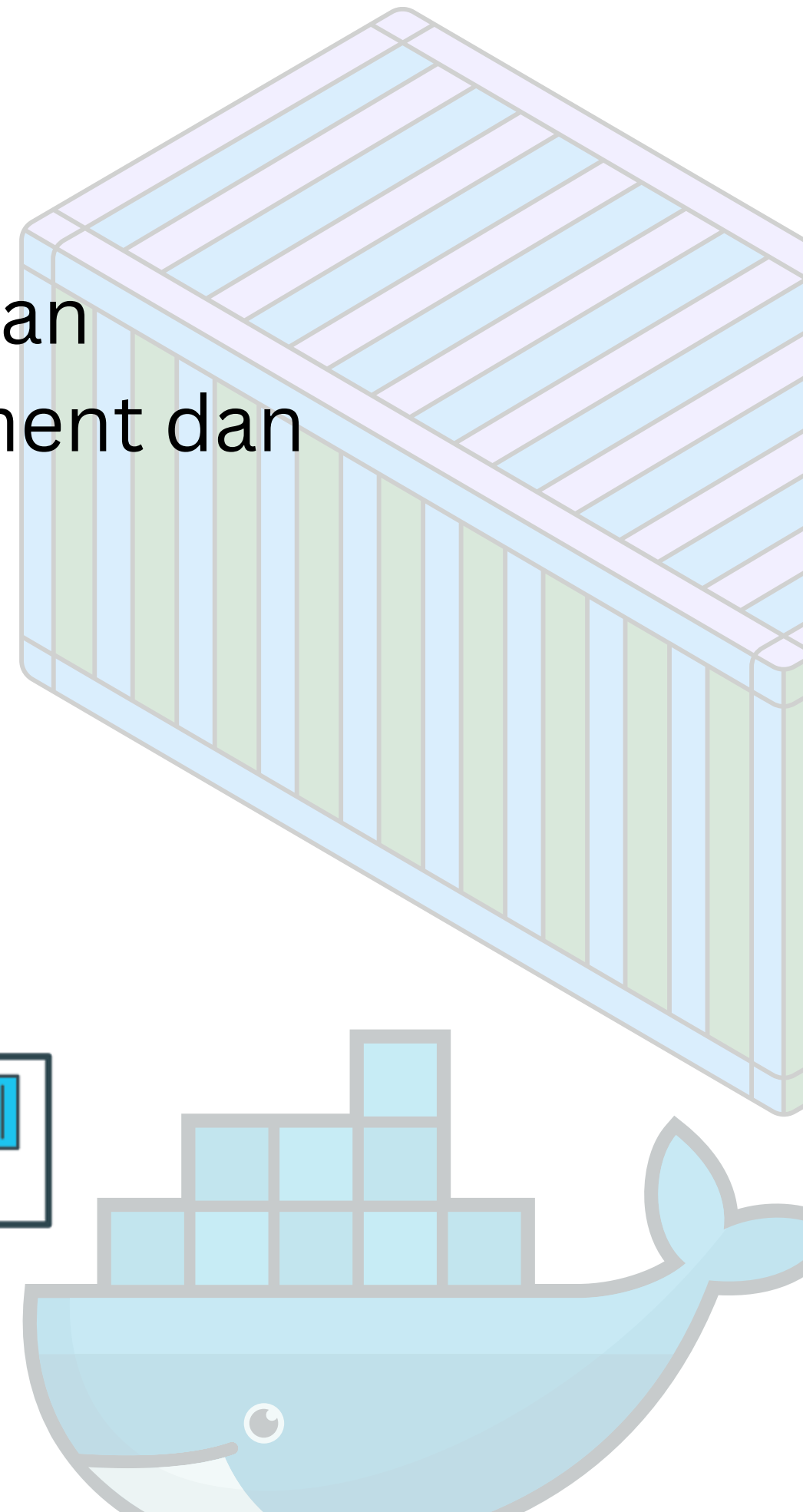
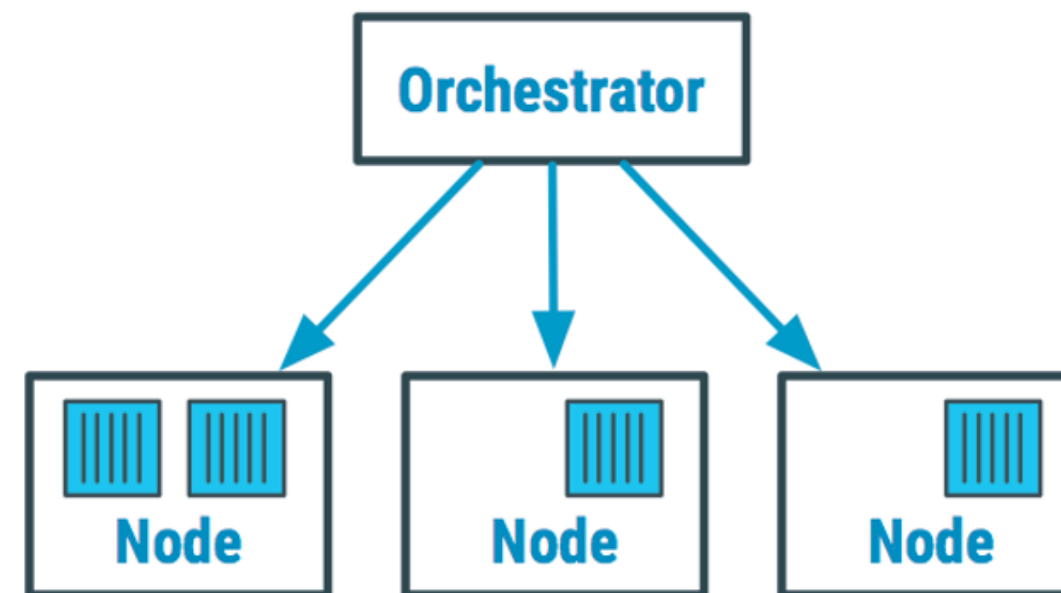
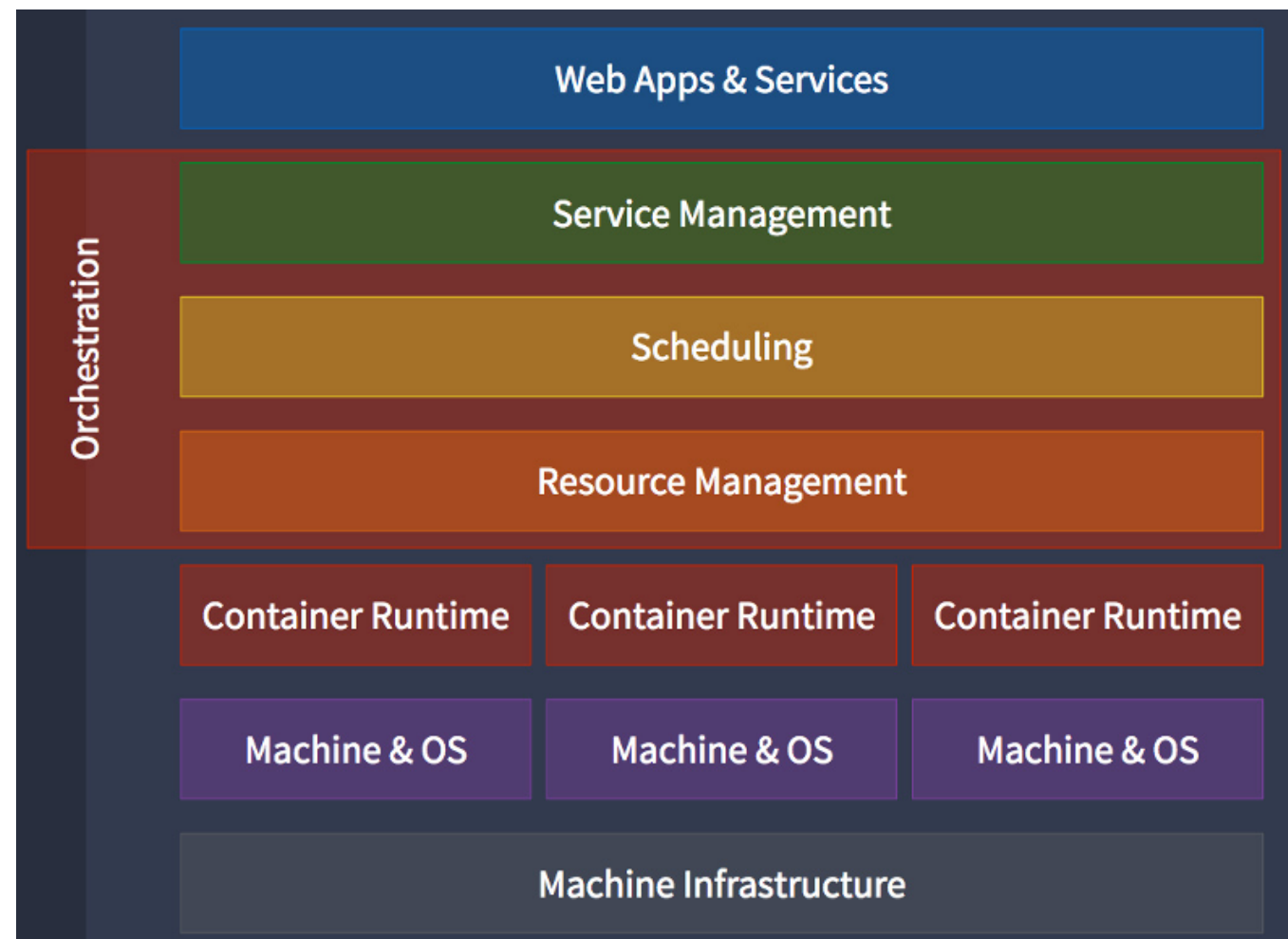
Apa itu Container Orchestration ?

- Container orchestration adalah otomatisasi dari sebagian proses operasional yang diperlukan untuk menjalankan workload dan services yang ada pada container.
- Ini mencakup banyak hal yang diperlukan tim pengembang software untuk mengelola container life cycle, termasuk:
 - provisioning
 - deployment
 - scaling
 - networking
 - load balancing, dsb.



Container Orchestration

- Container orchestration melakukan otomatisasi dan menyederhanakan proses provisioning, deployment dan manajemen pada aplikasi container



Container Orchestration Software (Docker, Openshift & Kubernetes)



OPENSIFT

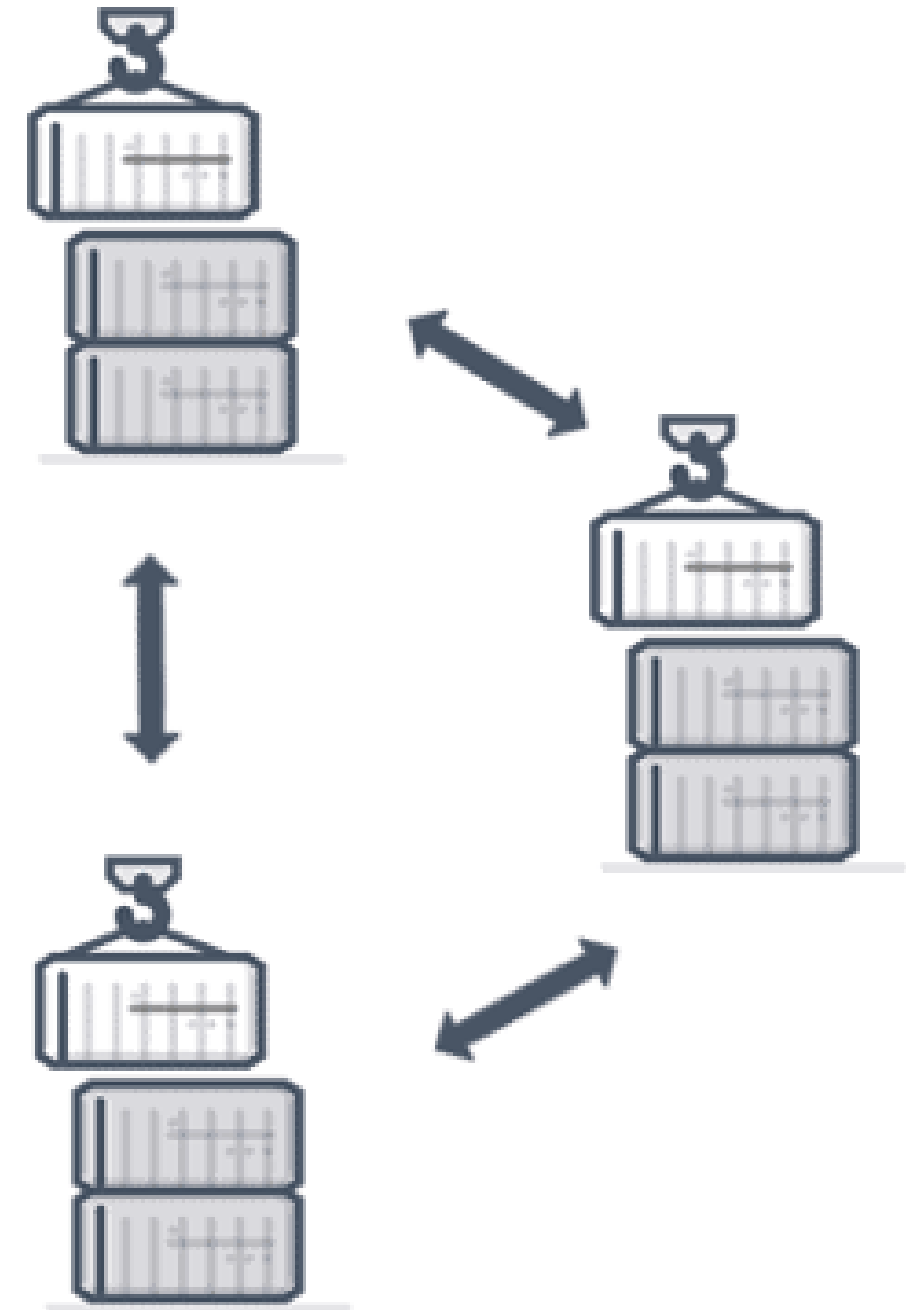


Automate:

- Configuration
- Provisioning
- Availability
- Scaling
- Security
- Resource allocation
- Load balancing
- Health monitoring

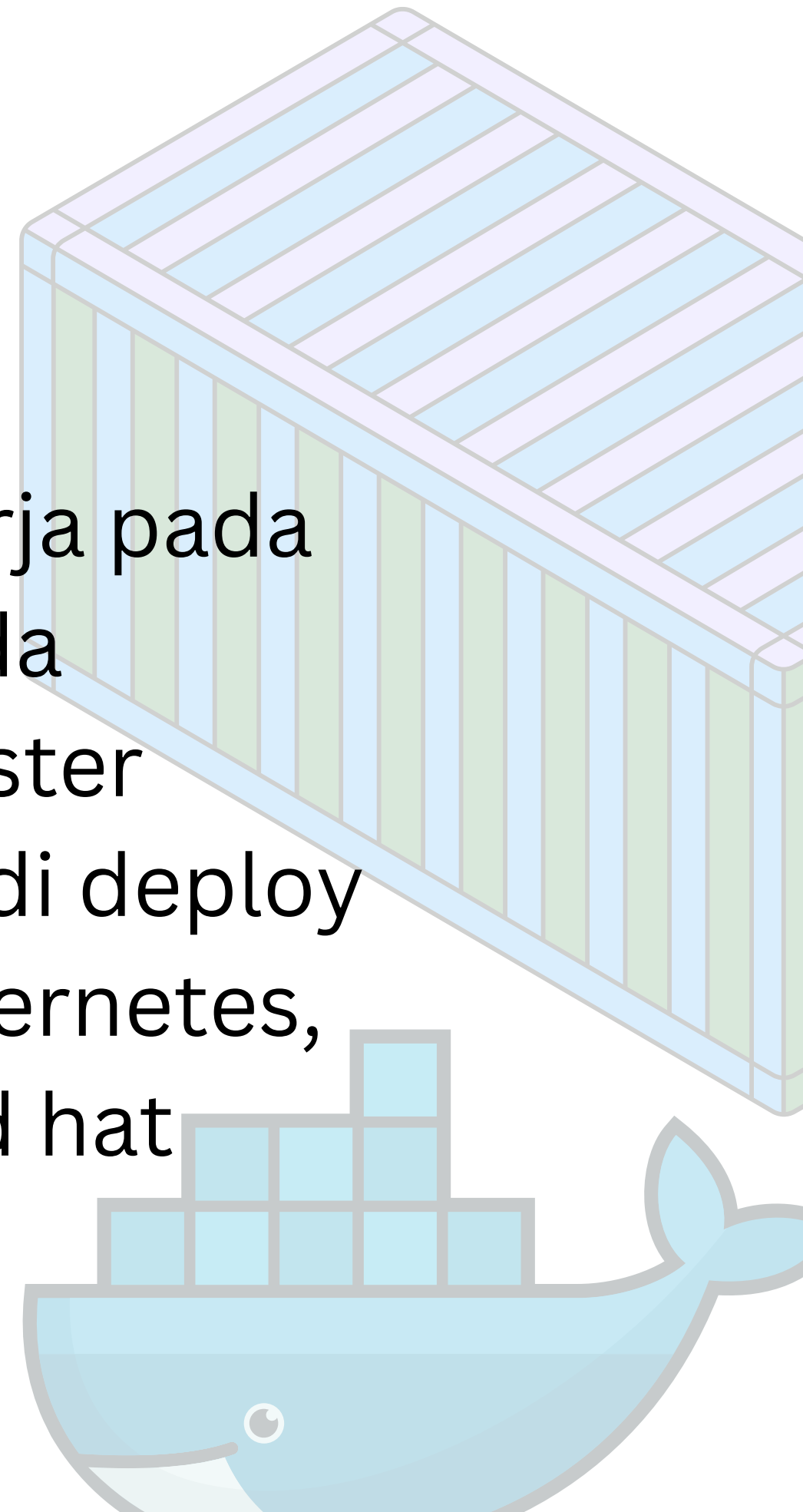


Application Environment w/ Multiple Containers



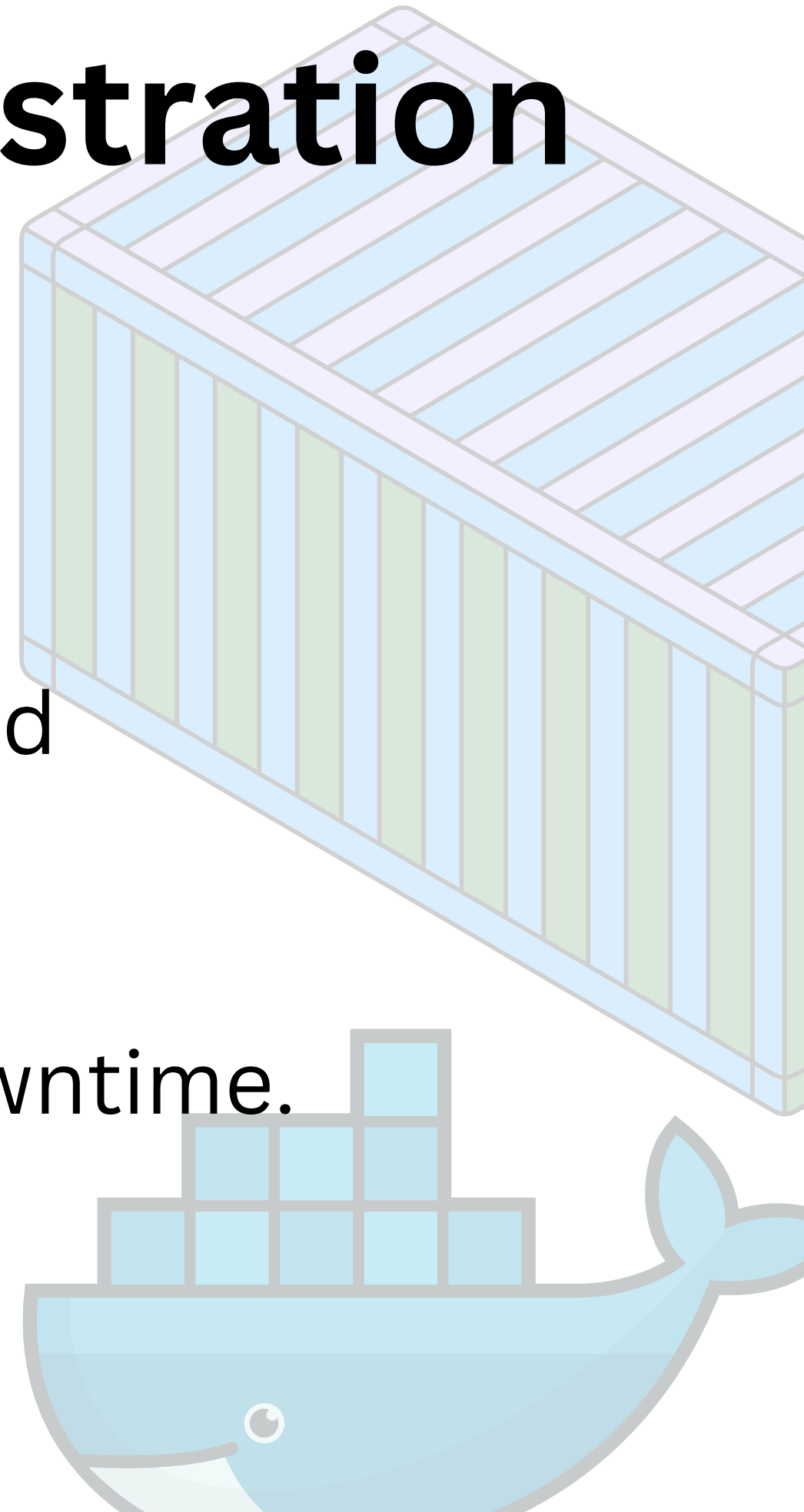
Container Orchestration

- Container orchestration melakukan proses manajemen atau penjadwalan dari beban kerja pada container secara otomatis untuk aplikasi pada microservice yang menggunakan banyak cluster
- Secara umum container orchestration yang di deploy berbasis pada versi open source seperti kubernetes, docker swarm atau versi komersil seperti red hat openshift



Kelebihan Container Orchestration

- Fault-tolerance
- On-demand scalability
- Optimal resource usage
- Auto-discovery to automatically discover and communicate with each other
- Accessibility from the outside world
- Seamless updates/rollbacks without any downtime.



Tools Container Orchestration

