Operating Systems and Distributed Systems

2021


+1!
LEVEL
UP

Welcome

People:

- Leon Bonde Larsen (Lecturer)
- Ahmad Rzgar Hamid (Lecturer)
- Mikkel Brinchs Larsen (Instructor)
- Oliver Lind Nordestgaard (Instructor)
- Hala Rahim Fadhel Al-Janabi (Instructor)
- Mads Møller Hansen (Instructor)
- Amalie Sander Jensen (Instructor)
- Patrick Hansen (Instructor)

Course contents

You will learn:

- what the OS does for you
  - scheduling
  - memory
  - storage
- how network and the internet works
  - protocols
  - security
  - distributed systems
- to use linux and the command line
- to build cloud native solutions

Philosophy

- Do practical work in class
- Read at home
- Discuss in study groups

It is all about what YOU do!

Structure

- Basic lecture (Thursday 13:00-13:45)
  - Basic topic
- Lab exercises (Thursday 14:00-15:45)
  - In study groups
  - With instructors and teachers
- Advanced lecture (Thursday 16:00-16:45)
  - Advanced stuff
  - Inspiration
  - Future directions

Materials

- No book
- All materials in english
- Laptop
- Ethernet port
- Know your machine

Study groups

- Why study groups?
  - Practical work
  - Discussions
  - Motivation

Study technique

- Time

Schedule

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
8-10							
10-12							
12-14							
14-16							
16-18							
18-20							

Schedule

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
8-10				DES			
10-12	PRO						
12-14	PRO	CCM	OPN	DES			
14-16	PRO	CCM	OPN				
16-18							
18-20							

Schedule

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
8-10				DES			Netflix
10-12	PRO						Netflix
12-14	PRO	CCM	OPN	DES			Netflix
14-16	PRO	CCM	OPN				Netflix
16-18					Bar	Party	Netflix
18-20	Karate		Karate		Bar	Party	Netflix

Schedule

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
8-10	Read	Read	Read	DES	PRO		Netflix
10-12	PRO			Labs	PRO		Netflix
12-14	PRO	CCM	OPN	DES	PRO		Netflix
14-16	PRO	CCM	OPN		PRO		Netflix
16-18		Labs		Labs	Bar	Party	Netflix
18-20	Karate	Labs	Karate	Labs	Bar	Party	Netflix

Schedule

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
8-10		Read	Read	DES	PRO		Netflix
10-12	PRO	Read	Read	Labs	PRO		Netflix
12-14	PRO	CCM	OPN	DES	PRO		Netflix
14-16	PRO	CCM	OPN	Labs	PRO		Netflix
16-18	Labs	Labs	Labs		Bar	Party	Netflix
18-20	Karate		Karate		Bar	Party	Netflix

Schedule

Work done = Time spent x Focus

Exam

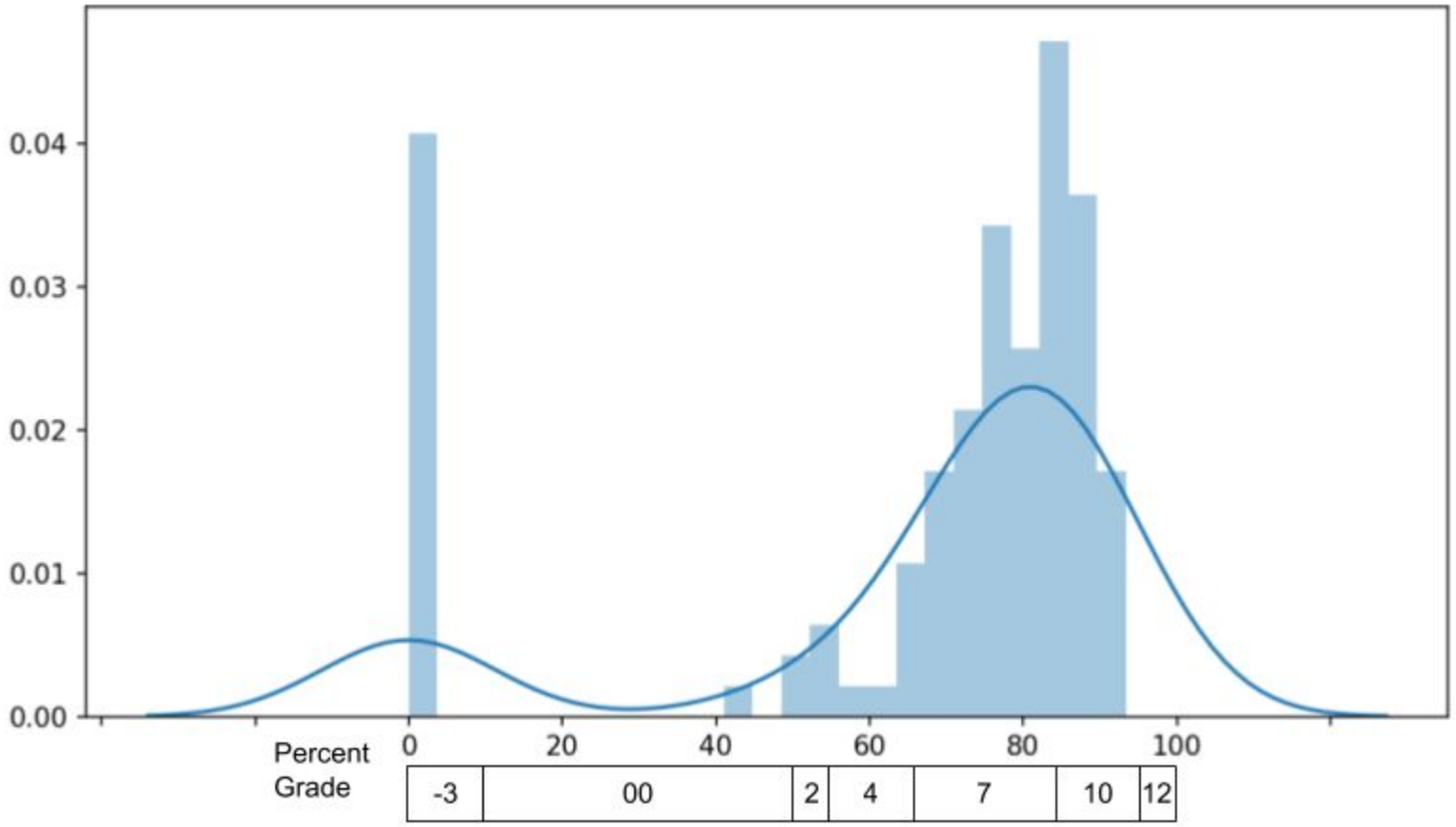
- Take home assignment - individual - pass/fail
- Written exam - 2 hours - multiple choice - grade

Multiple answer

What is the purpose of the DNS protocol?

A. Connect machines on the data link layer
B. Associate domain names with various information
C. Assign IP addresses to hosts in a network
D. Access distributed file system information

Multiple answer



Lecture plan

General Course Information

Description
This topic contains general information about the course.

For student guides and information regarding itslearning, mitsdu and digital exams, please follow this link.

2 plans

Introduction

Description
Introduction to the course, containers (Docker) and Container Orchestration (Kubernetes).

2 plans - from 01-09-2021 to 02-09-2021

Linux

Description
Introduction to the Linux Operating System, Scalable Systems and Network Models.

2 plans - from 02-09-2021 to 09-09-2021

Summary

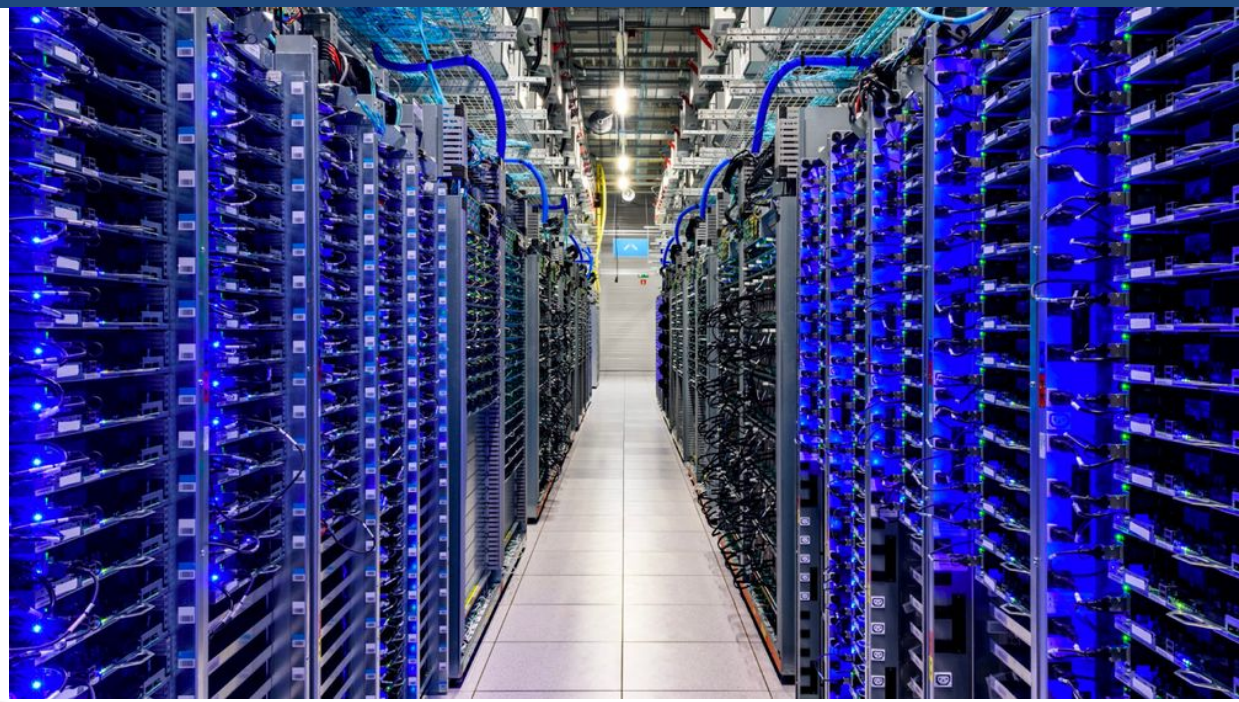
- Lectures and labs on Thursdays
- Be in control of your machine
- Make use of your study groups
- Develop good study habits (schedule + focus)
- Easy take home assignment (if you complete all labs)
- Written exam with lots of kahoot practise
- Check out the plan on itslearning

Introduction to containers

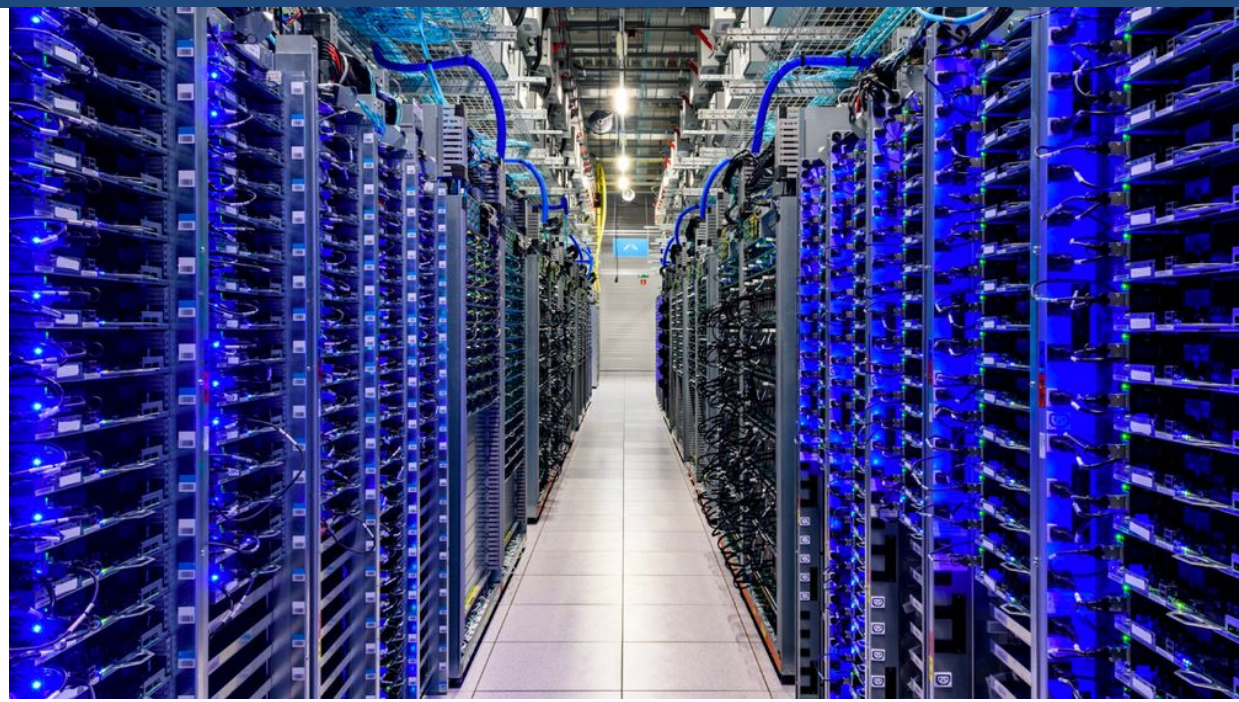
Leon Bonde Larsen et al.

Server

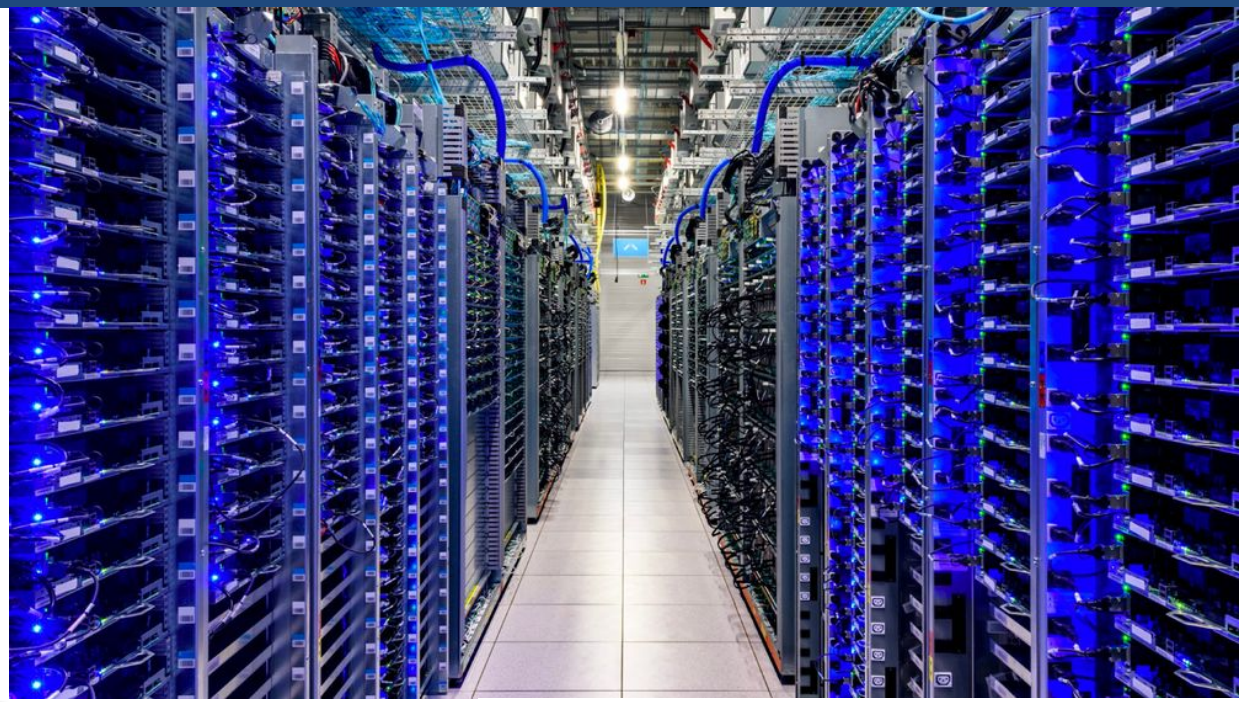
Server



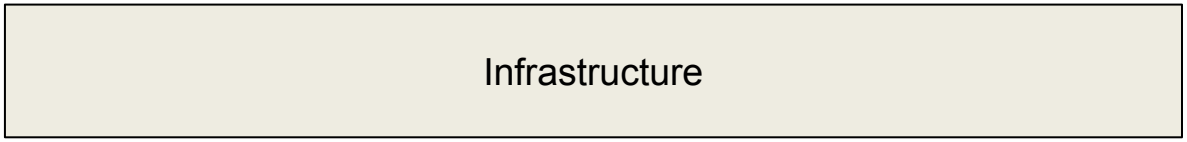
Server



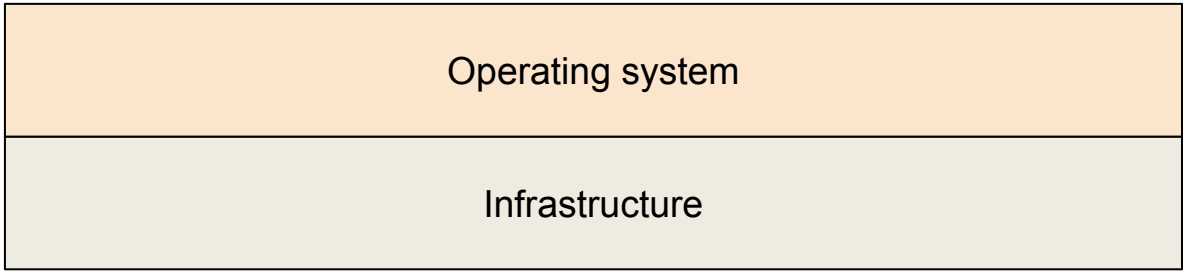
Server



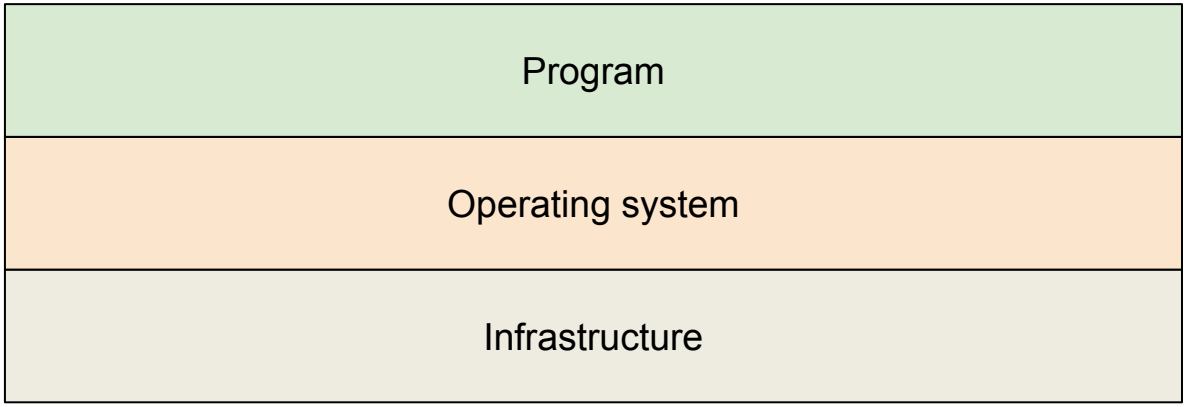
Server



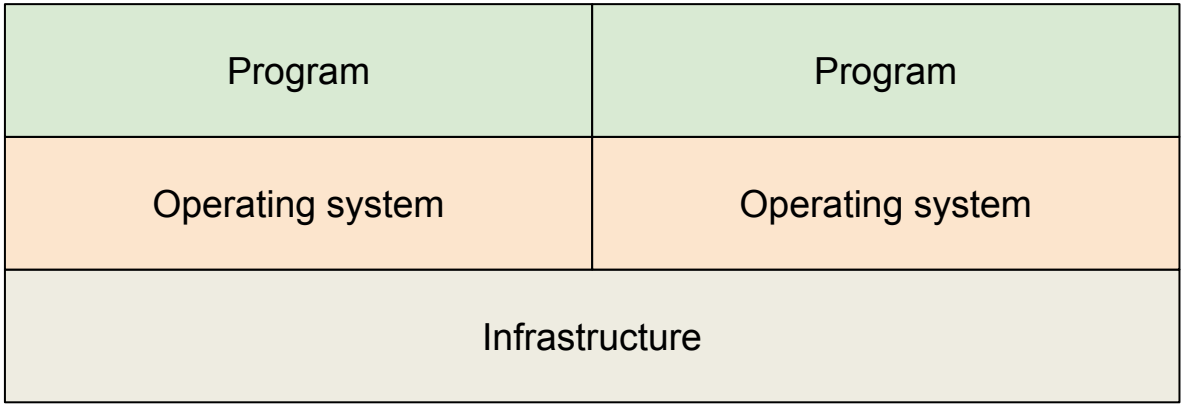
Server



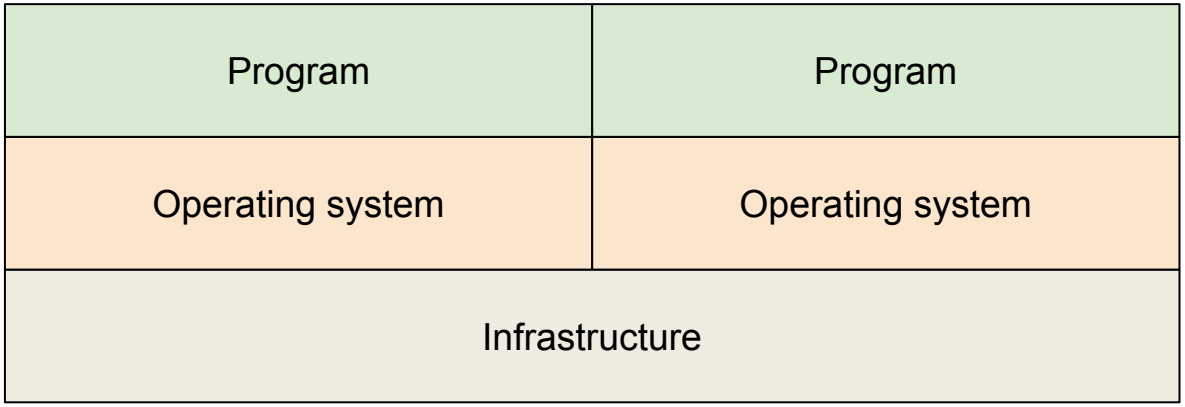
Server



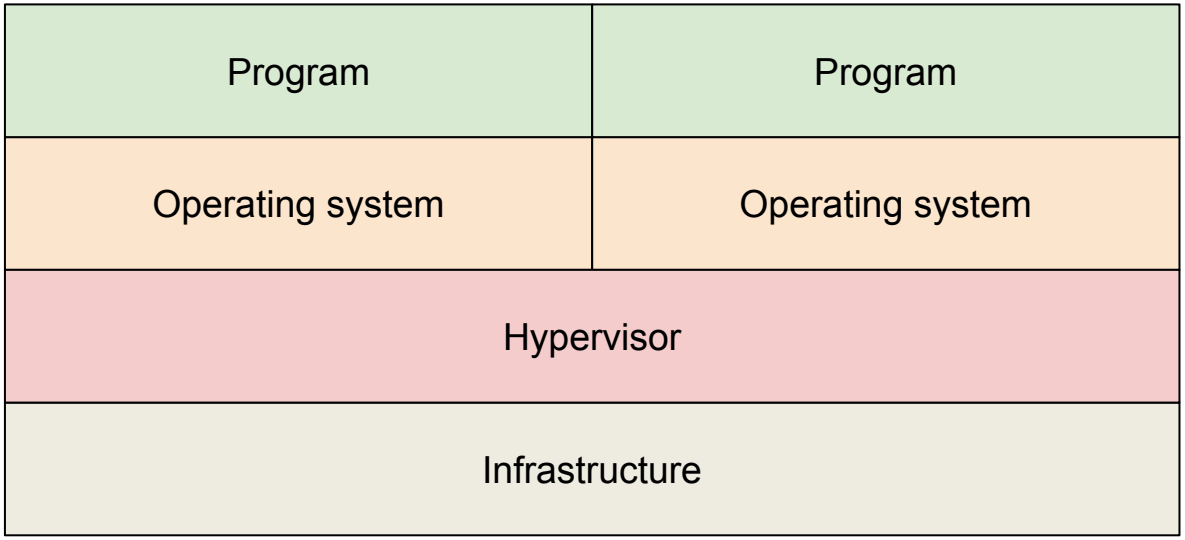
Virtual machines



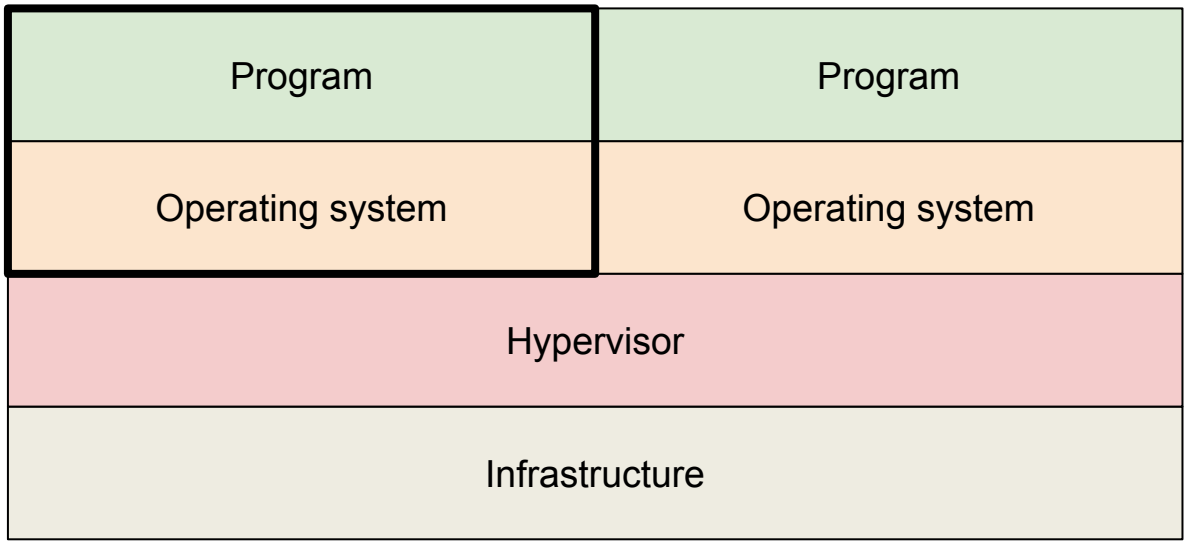
Virtual machines



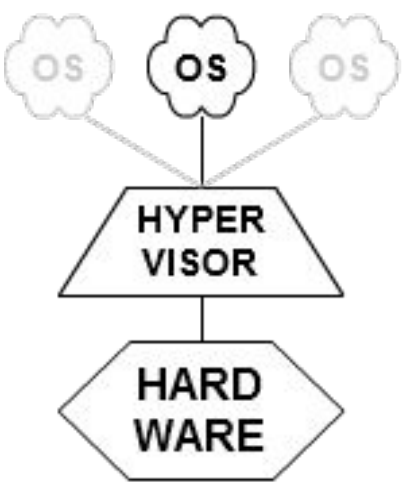
Virtual machines



Virtual machines

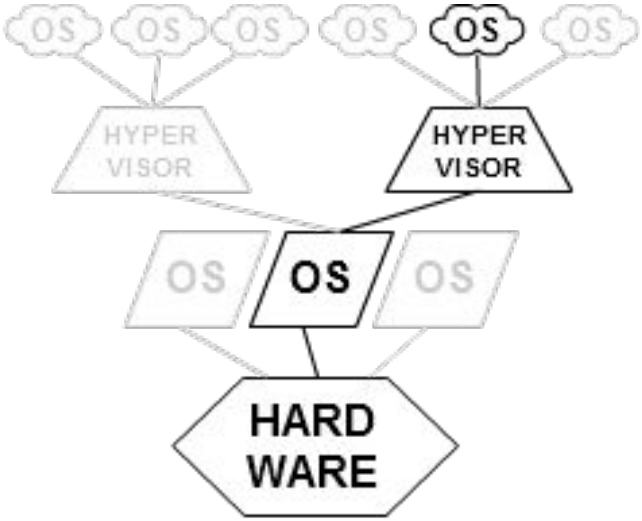


Virtual Machines



TYPE 1

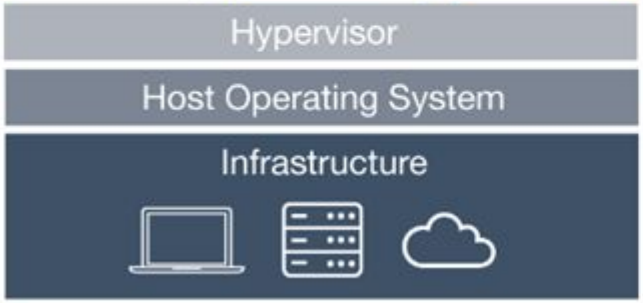
native
(bare metal)



TYPE 2

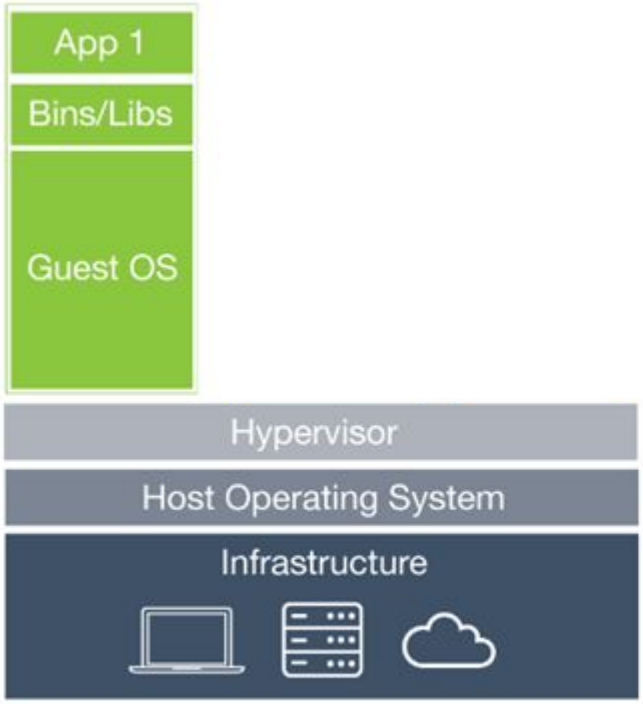
hosted

Virtual Machines



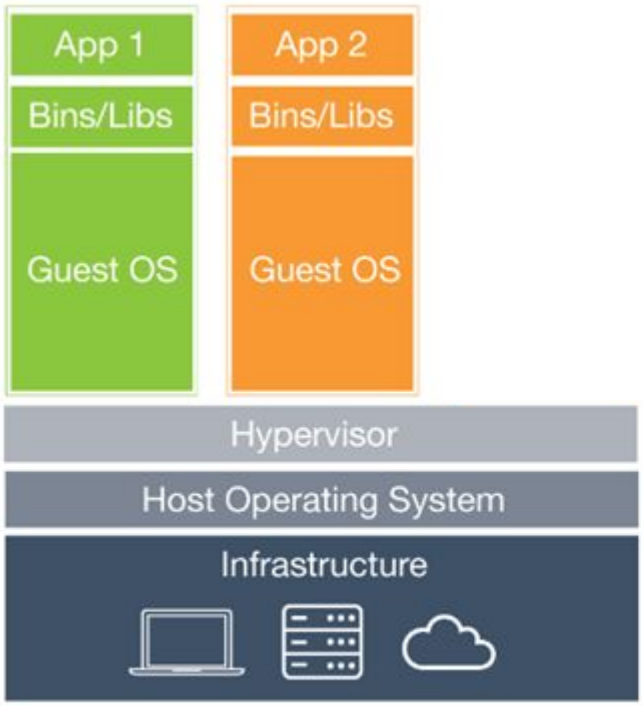
Virtual Machines

Virtual Machines



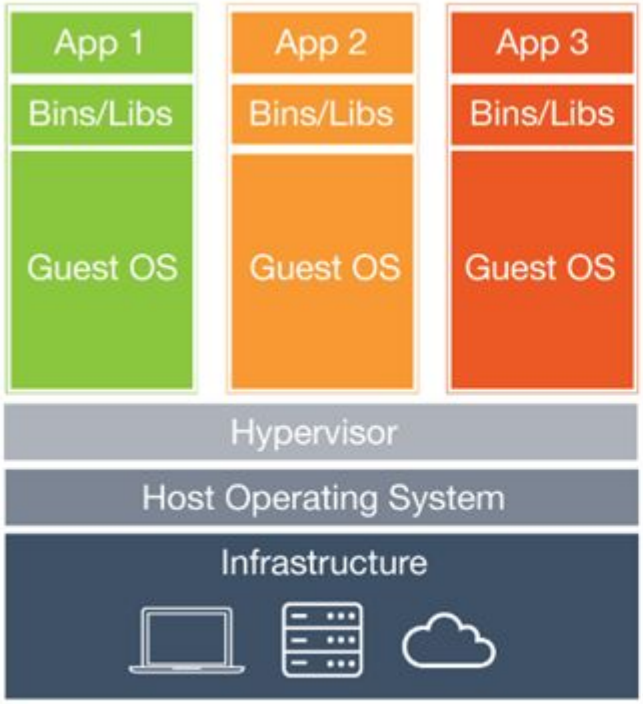
Virtual Machines

Virtual Machines



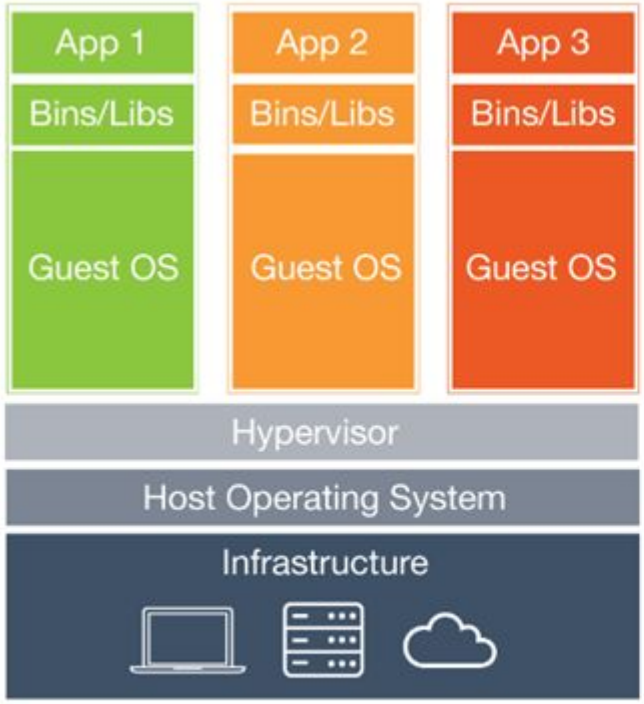
Virtual Machines

Virtual Machines



Virtual Machines

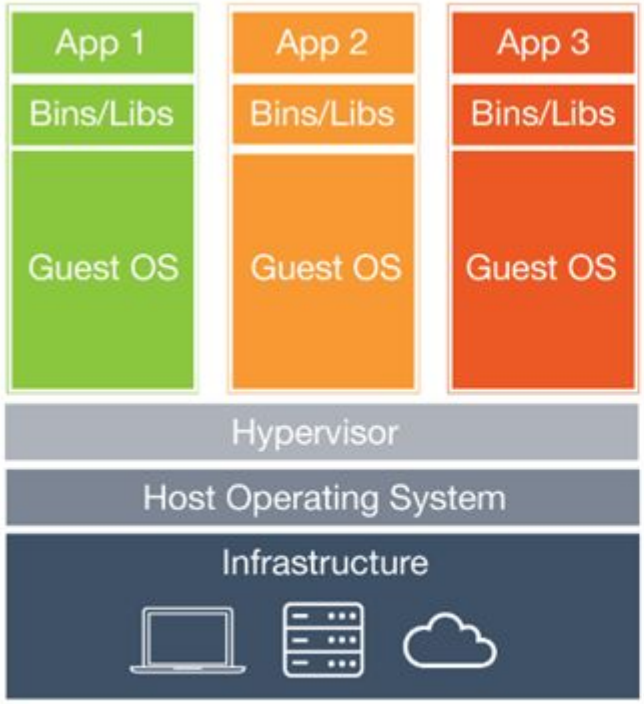
Containers



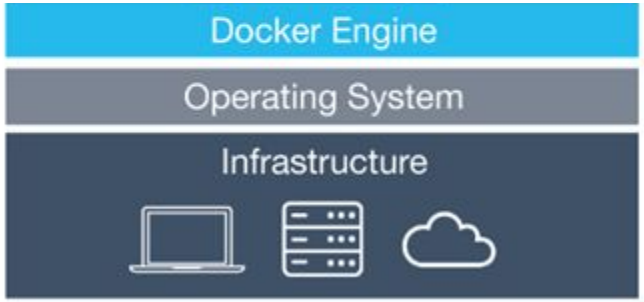
Virtual Machines

Containers

Containers

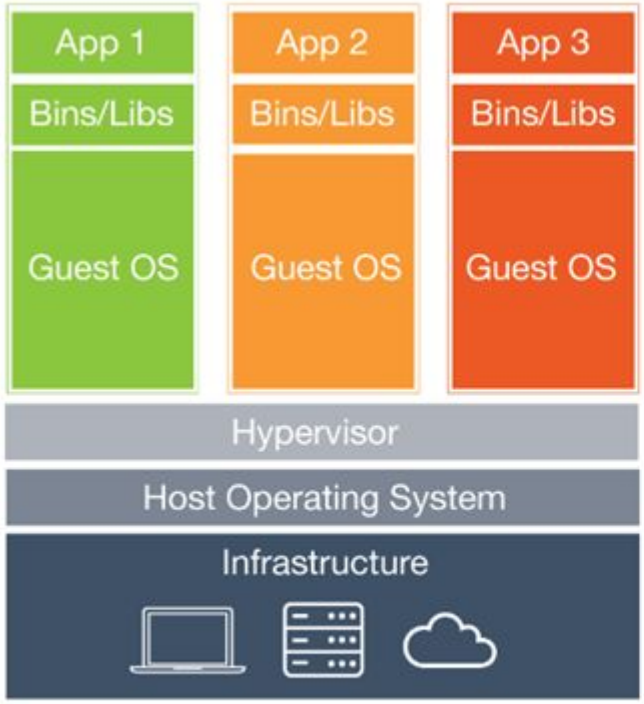


Virtual Machines

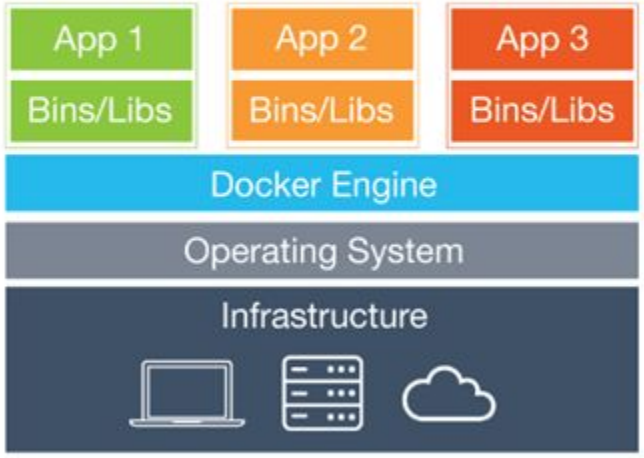


Containers

Containers

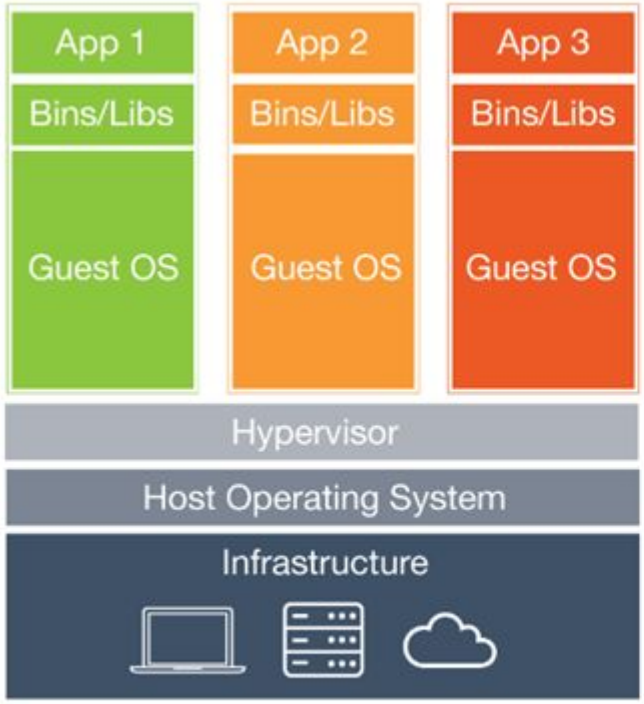


Virtual Machines

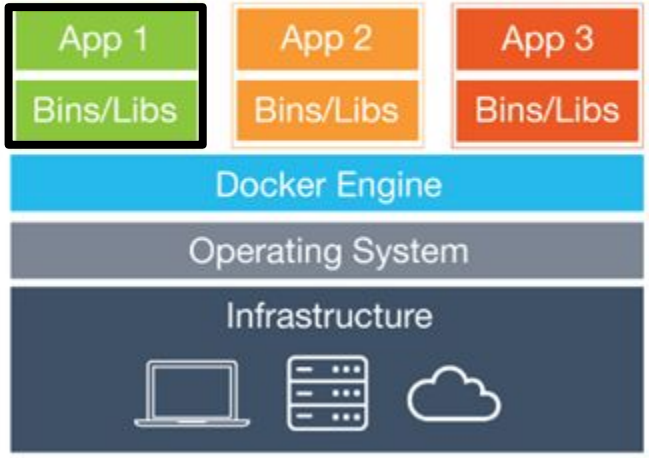


Containers

Containers



Virtual Machines

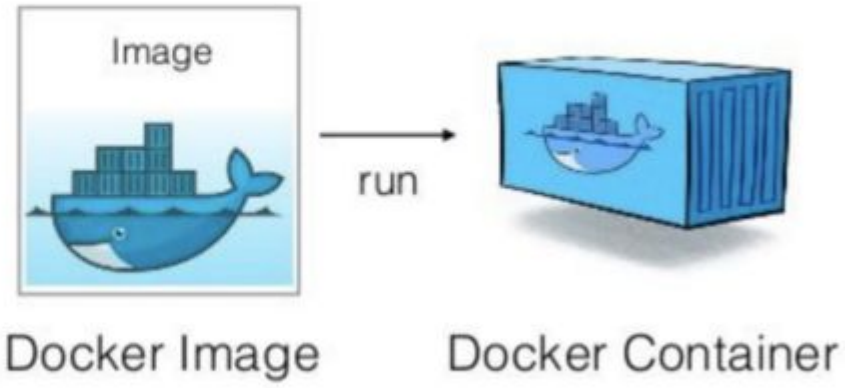


Containers

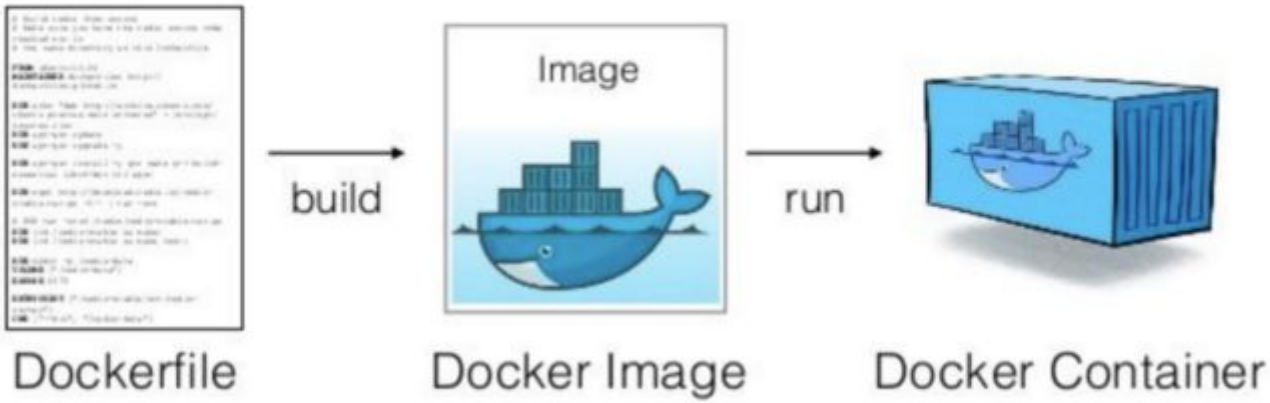
Concepts

- Containers
- Images
- Dockerfile
- Docker registry
- Data volumes
- Network

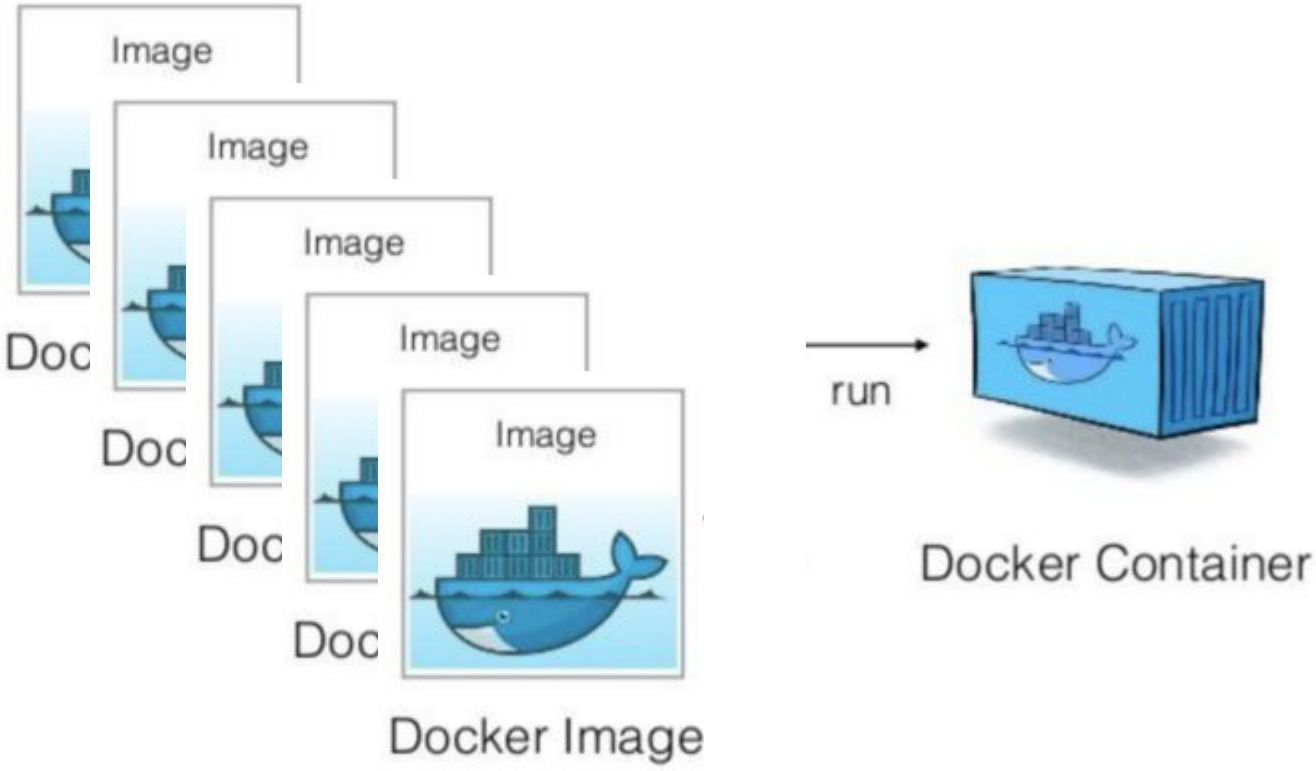
Docker - Images



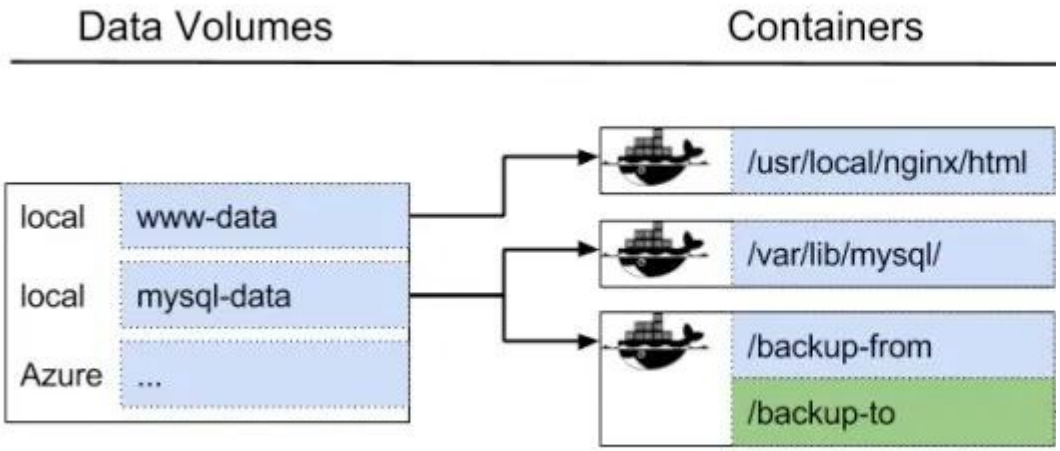
Docker - Dockerfile



Docker - Docker registry



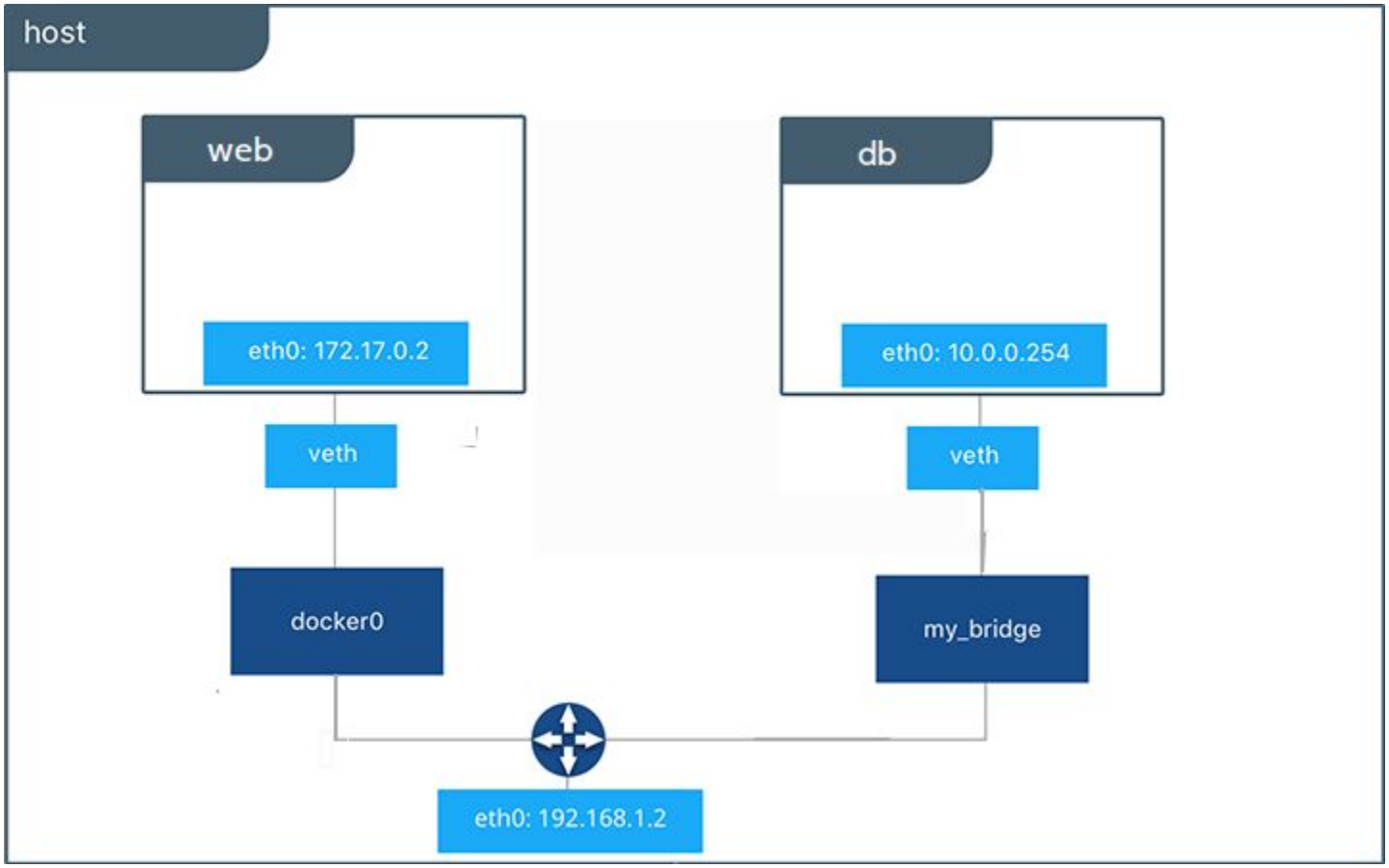
Docker - Data volumes



Docker - network

Docker - network

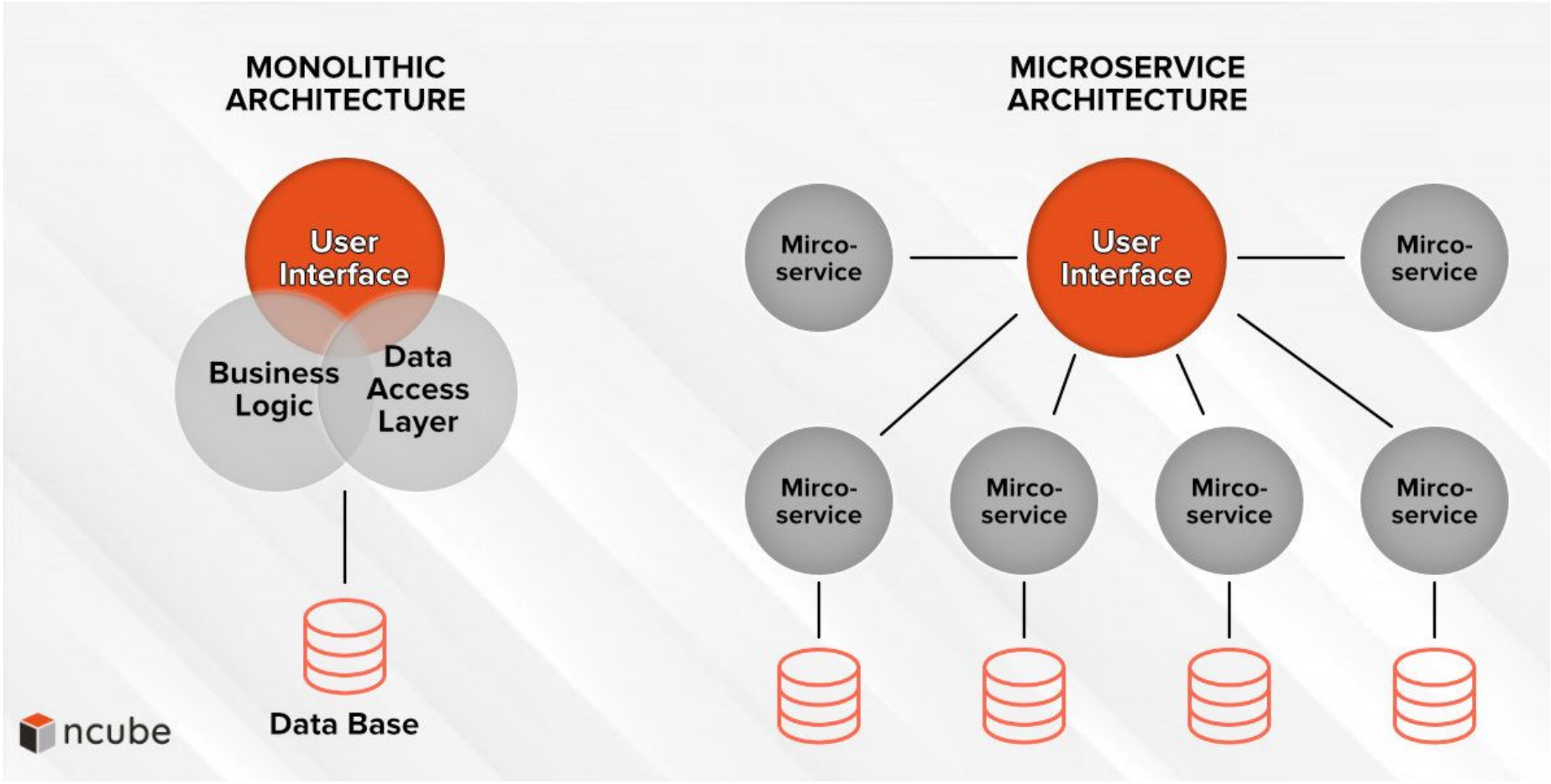
Docker - network



Docker commands

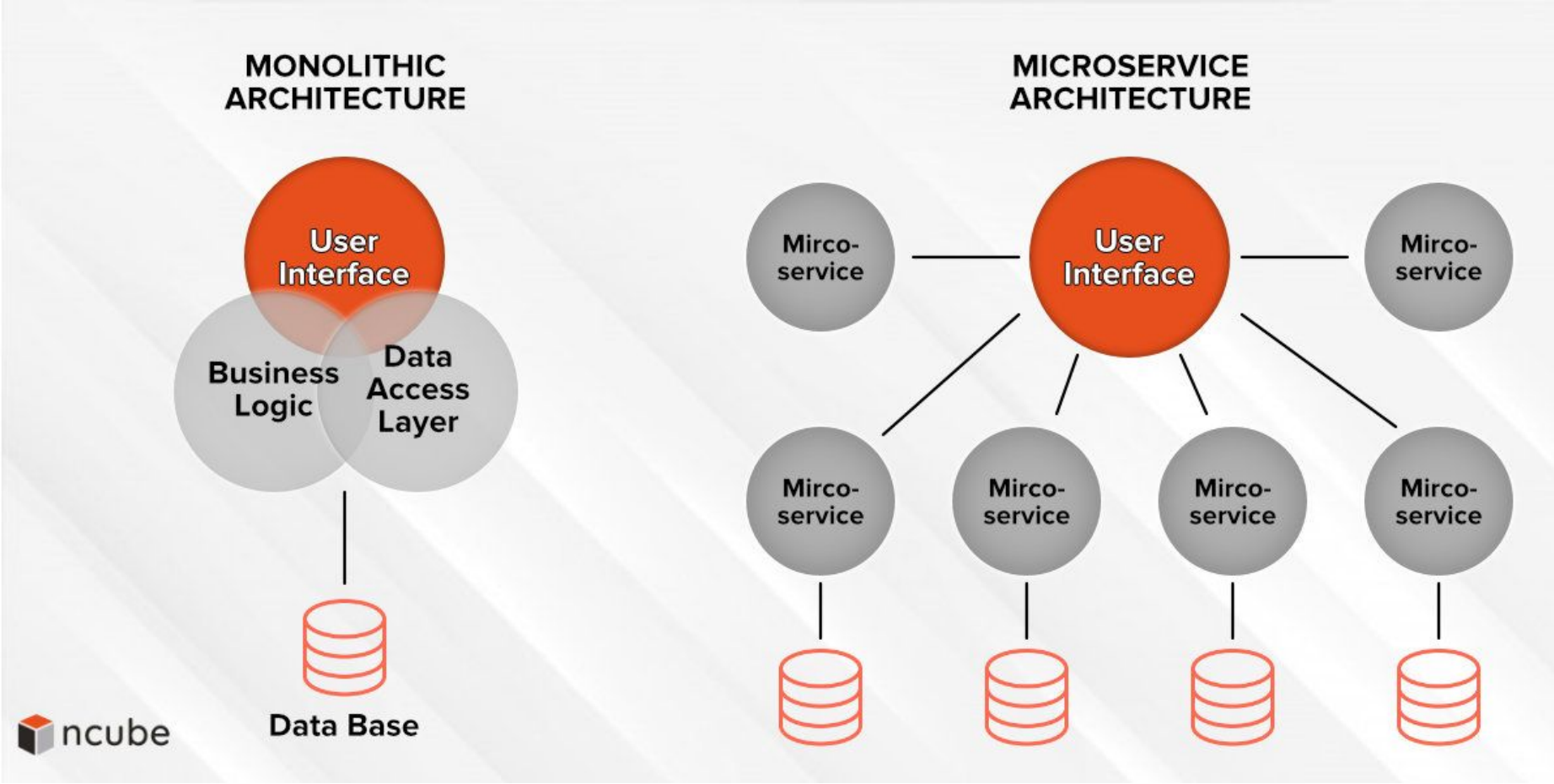
```
$ docker run                                    # run container
$ docker pull <image>                           # download image
$ docker build -t <username/imagename> .        # build image
$ docker start <containername>                  # start container
$ docker stop <containername>                   # stop container
```

Diskussion


MONOLITHIC ARCHITECTURE
User Interface
Business Logic
Data Access Layer
Data Base
ncube
MICROSERVICE ARCHITECTURE
Mirco-service
User Interface
Mirco-service
Mirco-service
Mirco-service
Mirco-service
Mirco-service

Diskussion

- Hvad kunne være fordele/ulemper ved microservice arkitektur?



Go to group rooms

```
Room A : U165 : Mikkel
Room B : U166 : Oliver
Room C : U167 : Hala og Mads
Room D : U171 : Amalie
Room E : U172 : Lasse
Room F : U176 : Patrick
```

- Work on Lab 1
- Go back at 15:45 and be ready here at 16:00
- Have fun!

Advanced lecture Kubernetes

Ahmad Rzgar Hamid et al.

Kubernetes

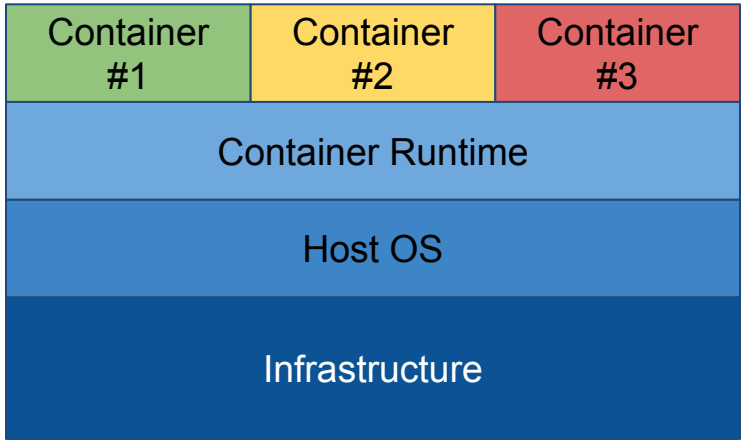
- Kubectl
- Manifest files
- Nodes
- Pods
  - Containers
    - Evt. side-car pattern
- Services
  - ClusterIp
  - Nodeport
  - LB
- Deployments
  - Stateful deployments
  - Cron jobs
  - etc
- Storage
- (Logging?)
- Overview of components
  - Controllers
  - API servers
  - Schedulers
  - ETCD
  - Kubelet
  - Kube-proxy
  - cloud-manager
- Semester project system
  - Ingress controller (nginx)
  - Nginx frontend LB
- Evt. Opsummering af containers samt k8s ?

Containers in Production

- Containers propose several advantages
  - Isolation
  - Portable
  - Maintainable
  - Deployable
- Is that good enough in a Production Environment?

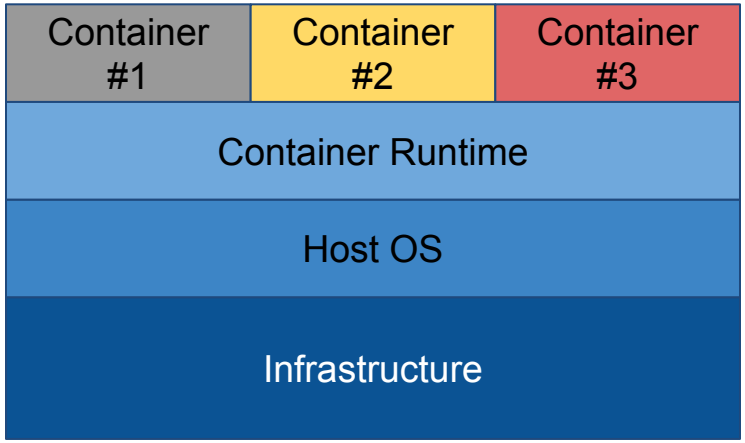
Single Machine Setup

How to handle a dead application or container?



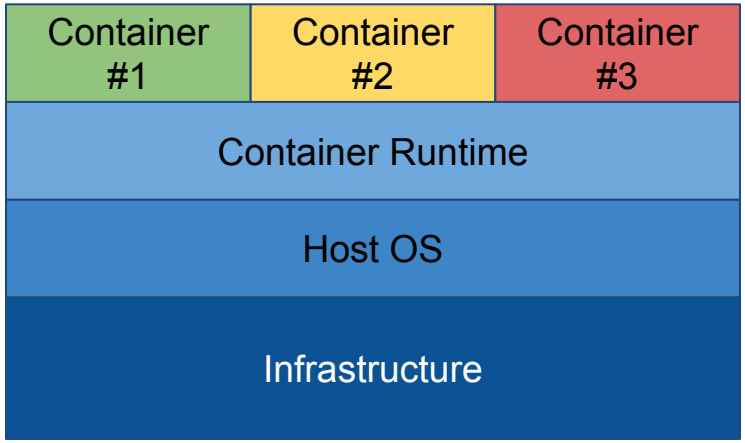
Single Machine Setup

How to handle a dead application or container?



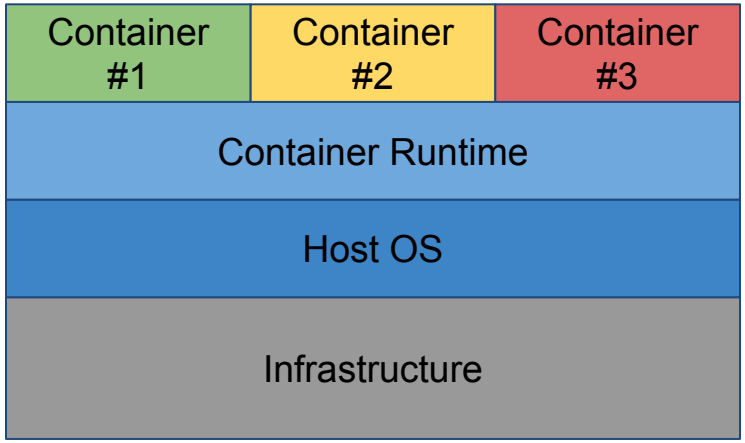
Single Machine Setup

How to handle a dead host machine?



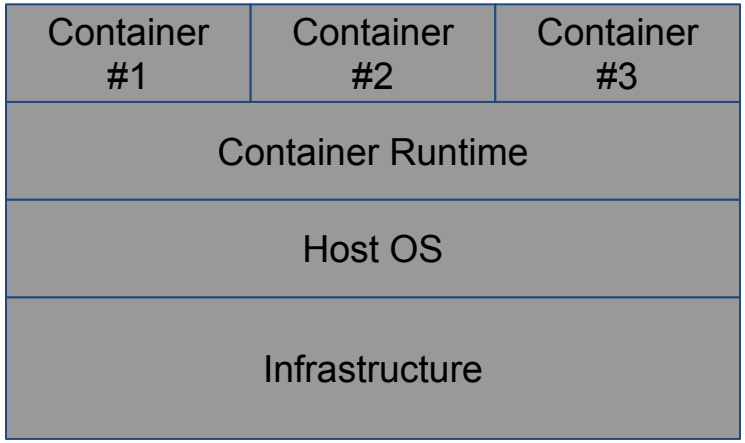
Single Machine Setup

How to handle a dead host machine?



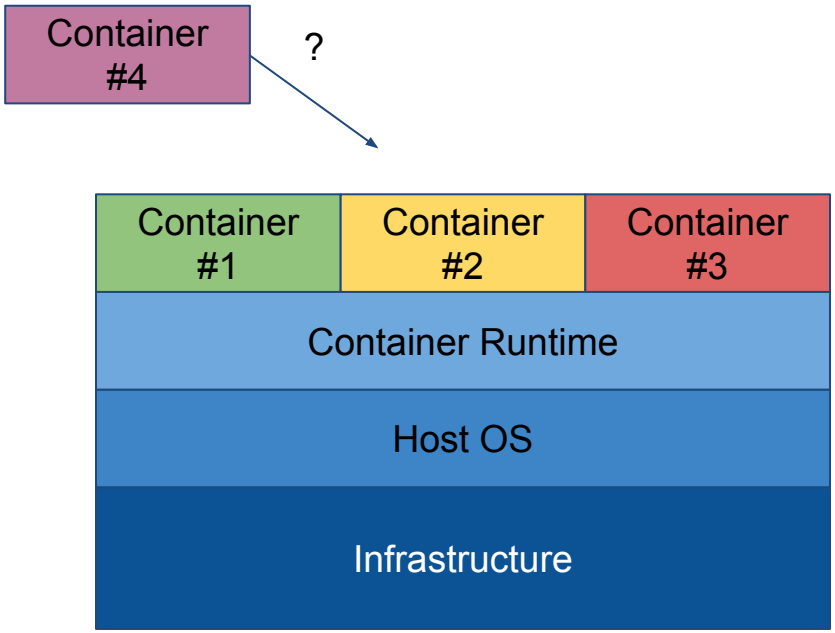
Single Machine Setup

How to handle a dead host machine?

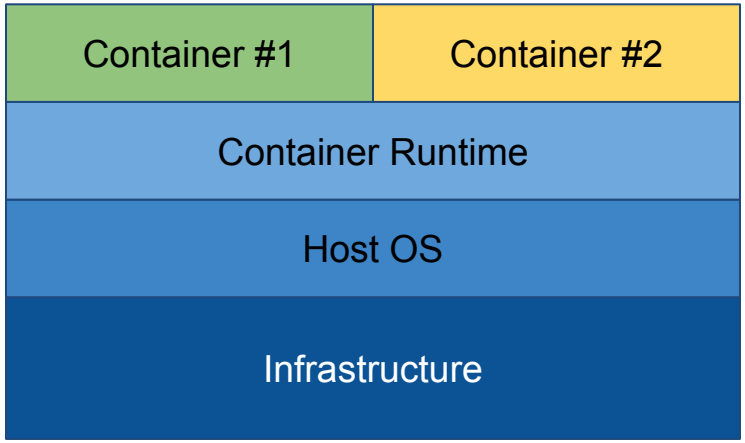


Single Machine Setup

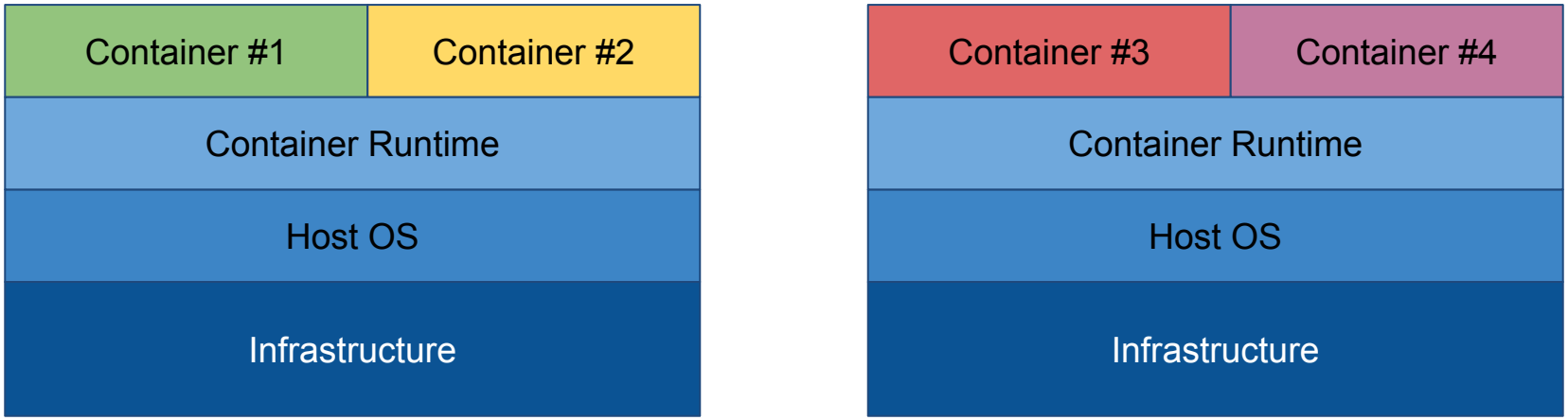
How to handle a full host machine, with no free resources?



Multi Machine Setup



Multi Machine Setup



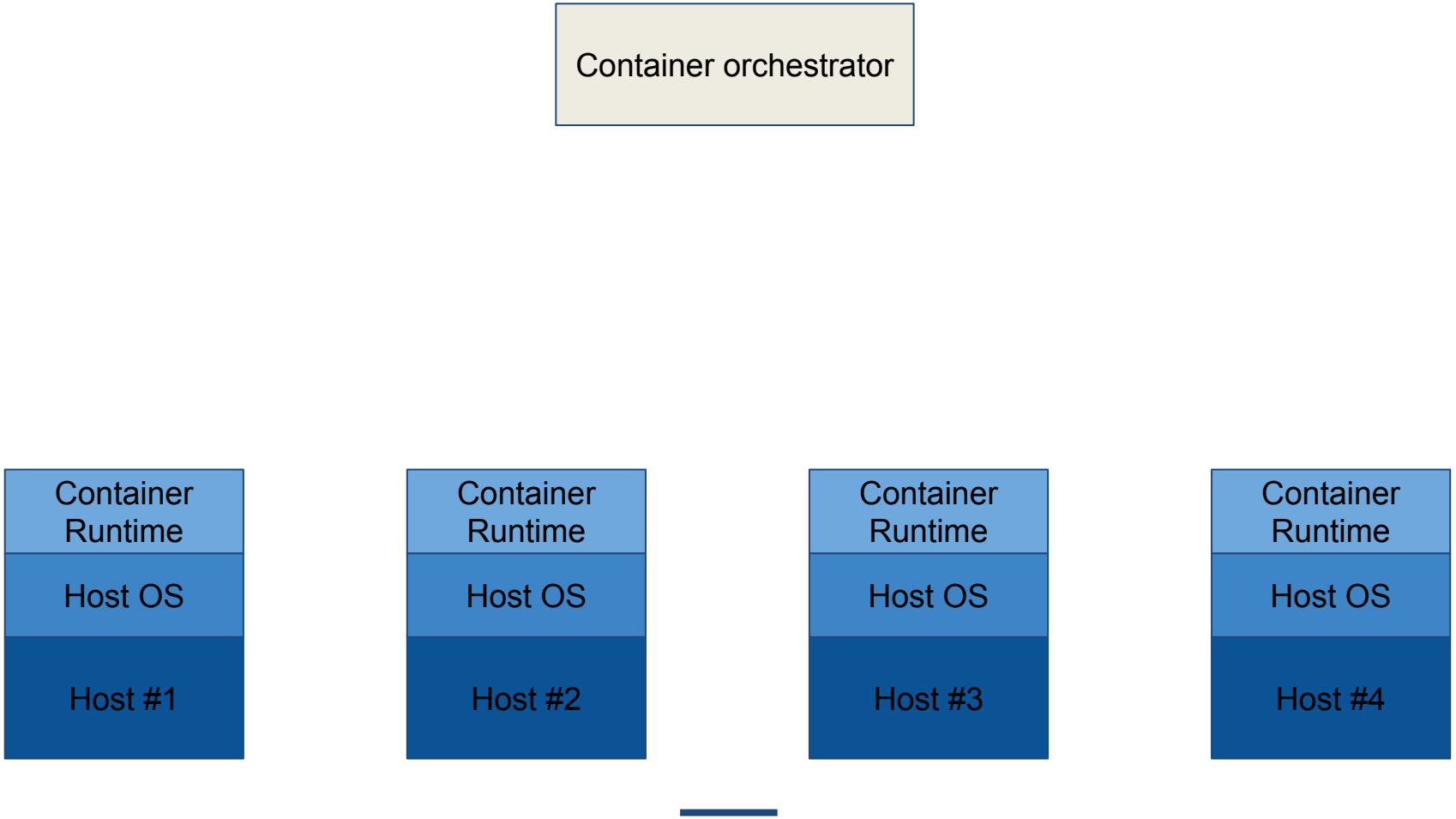
What is Missing?

- Scalability
- Availability
- Reliability
- Recoverability
- Robustness

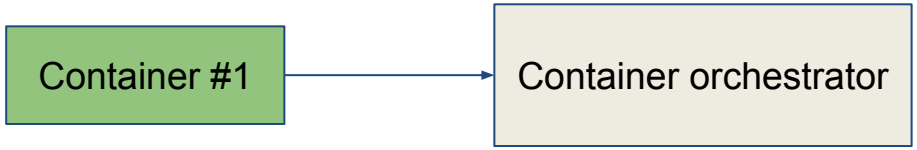
Container Orchestration

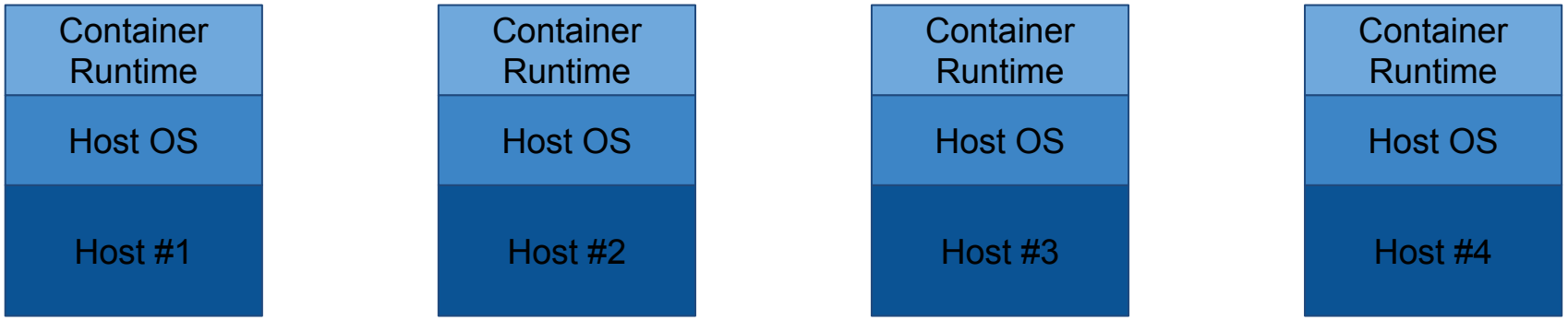
- Program that manages
  - Configuration
  - Coordination
  - Container Life Cycle
- Consist of a Cluster all running a Container Runtime (Read: Docker)*

Container Orchestration

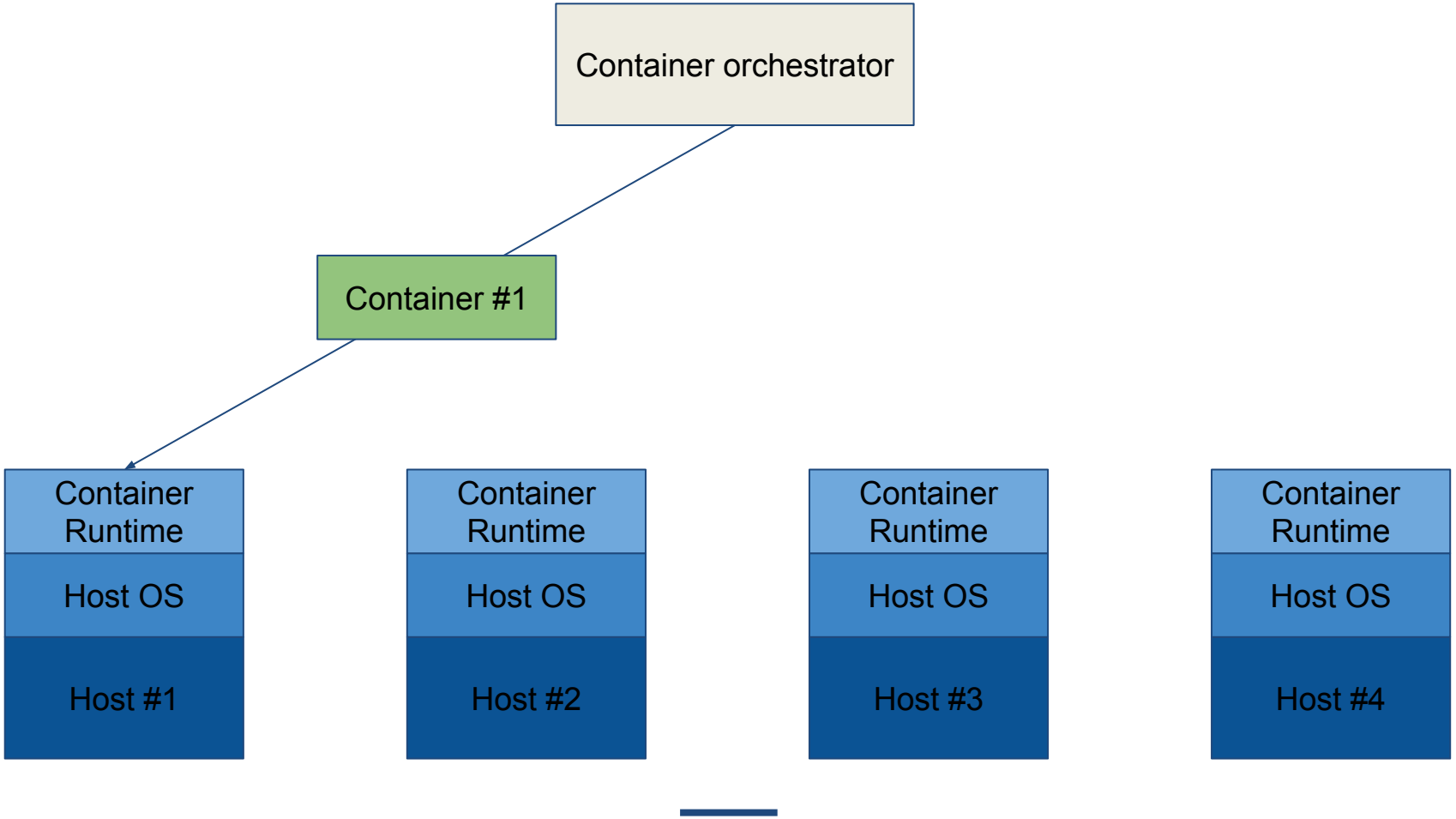


Container Orchestration

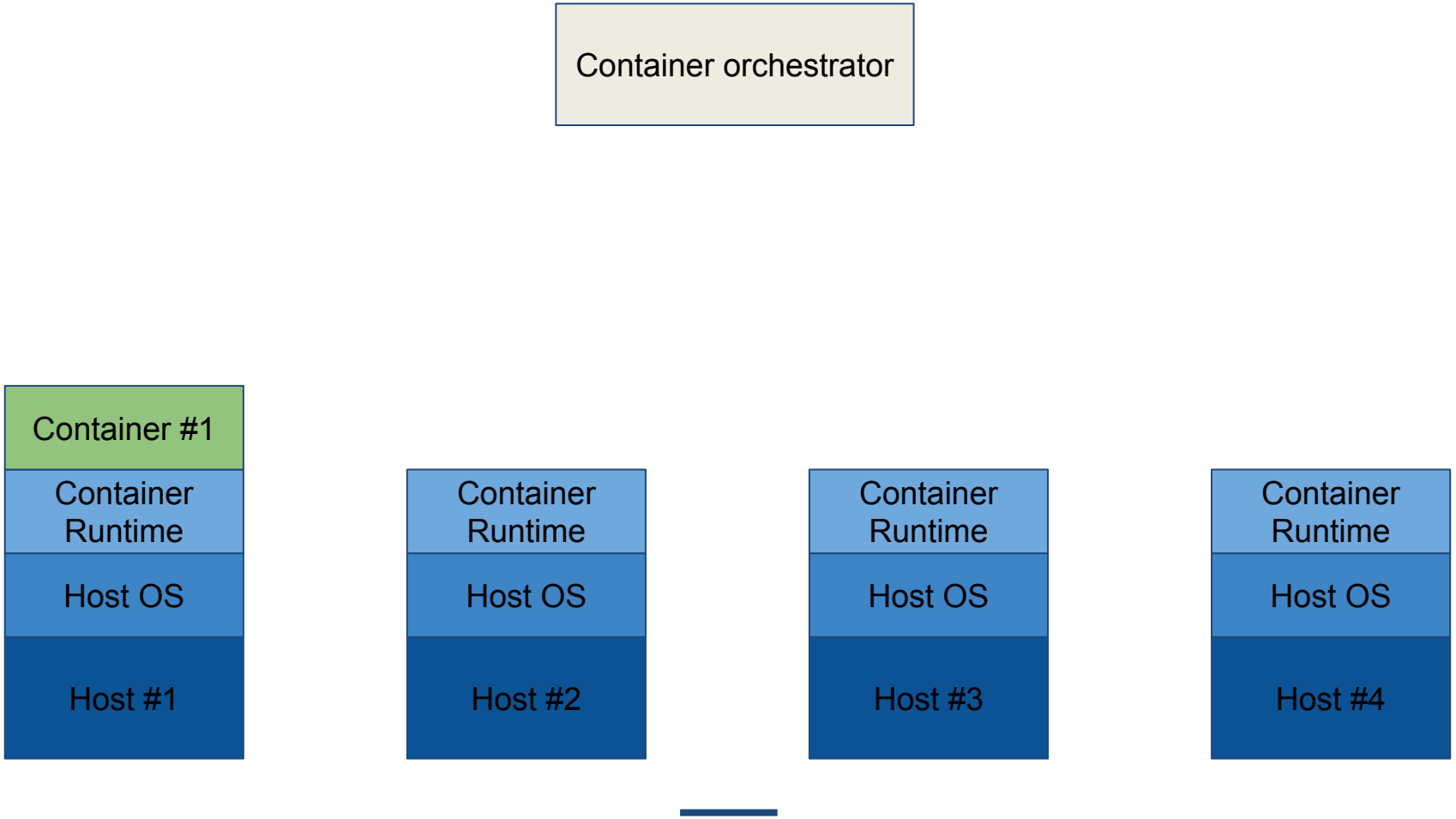




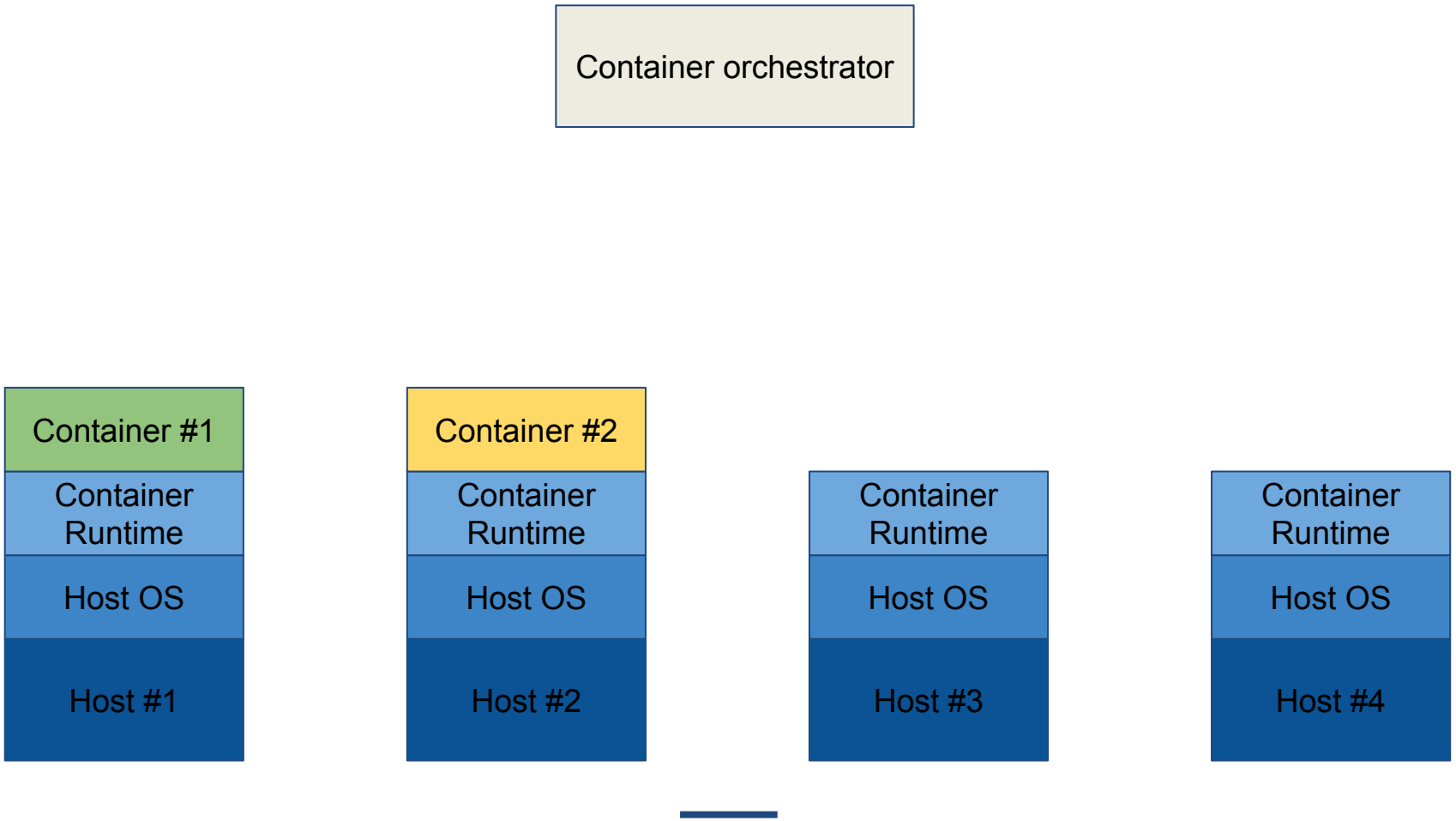
Container Orchestration



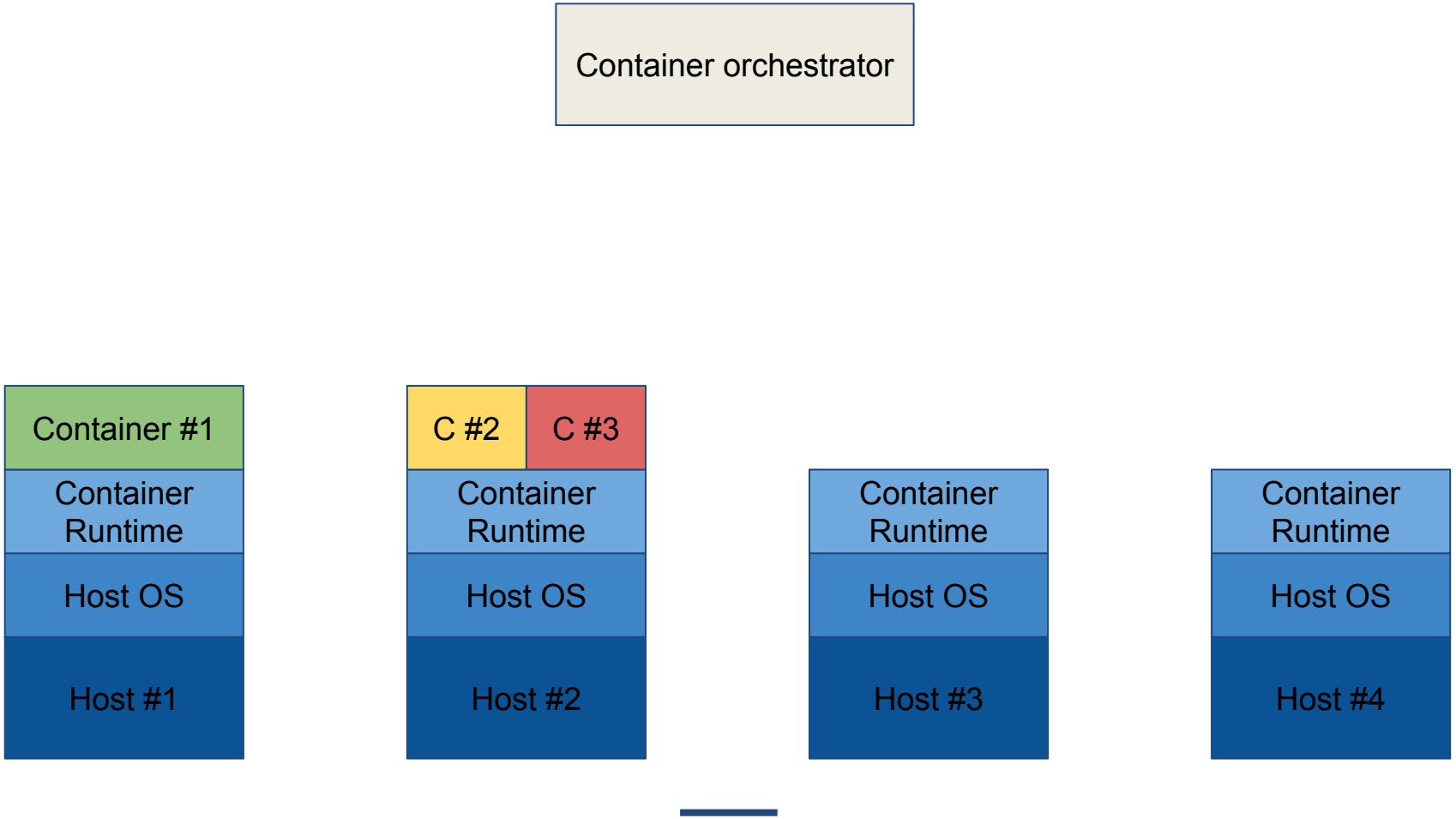
Container Orchestration



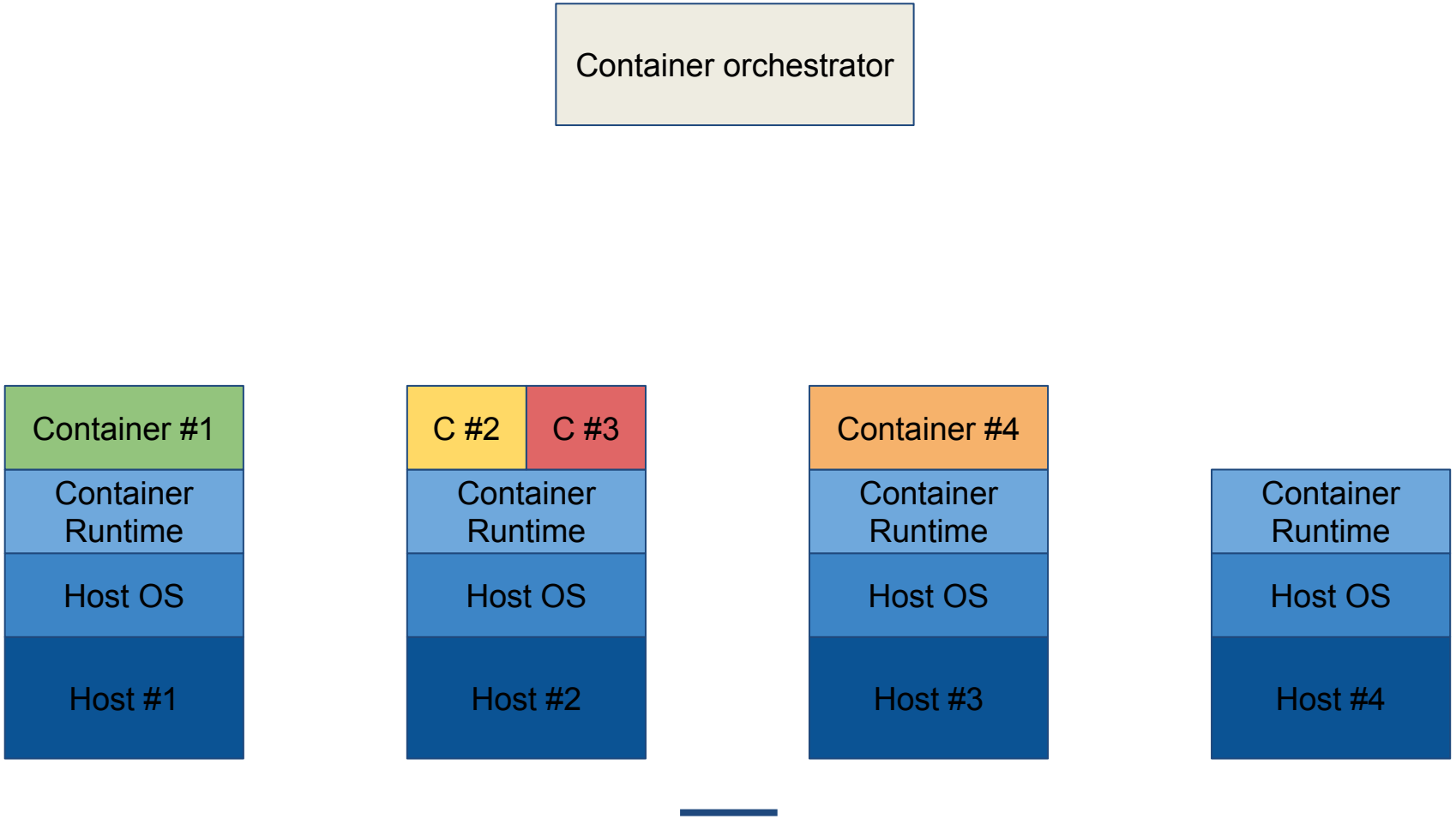
Container Orchestration



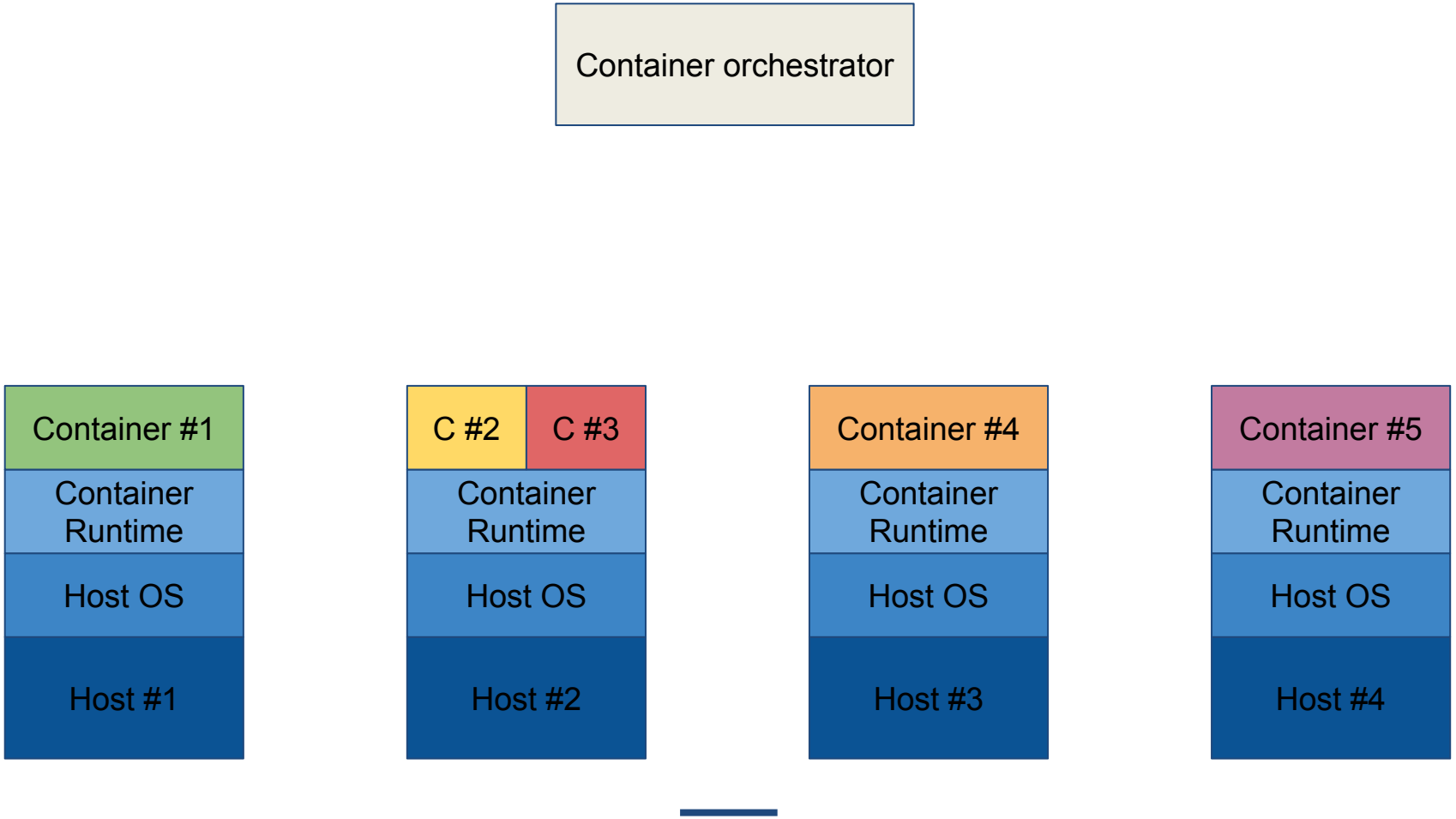
Container Orchestration



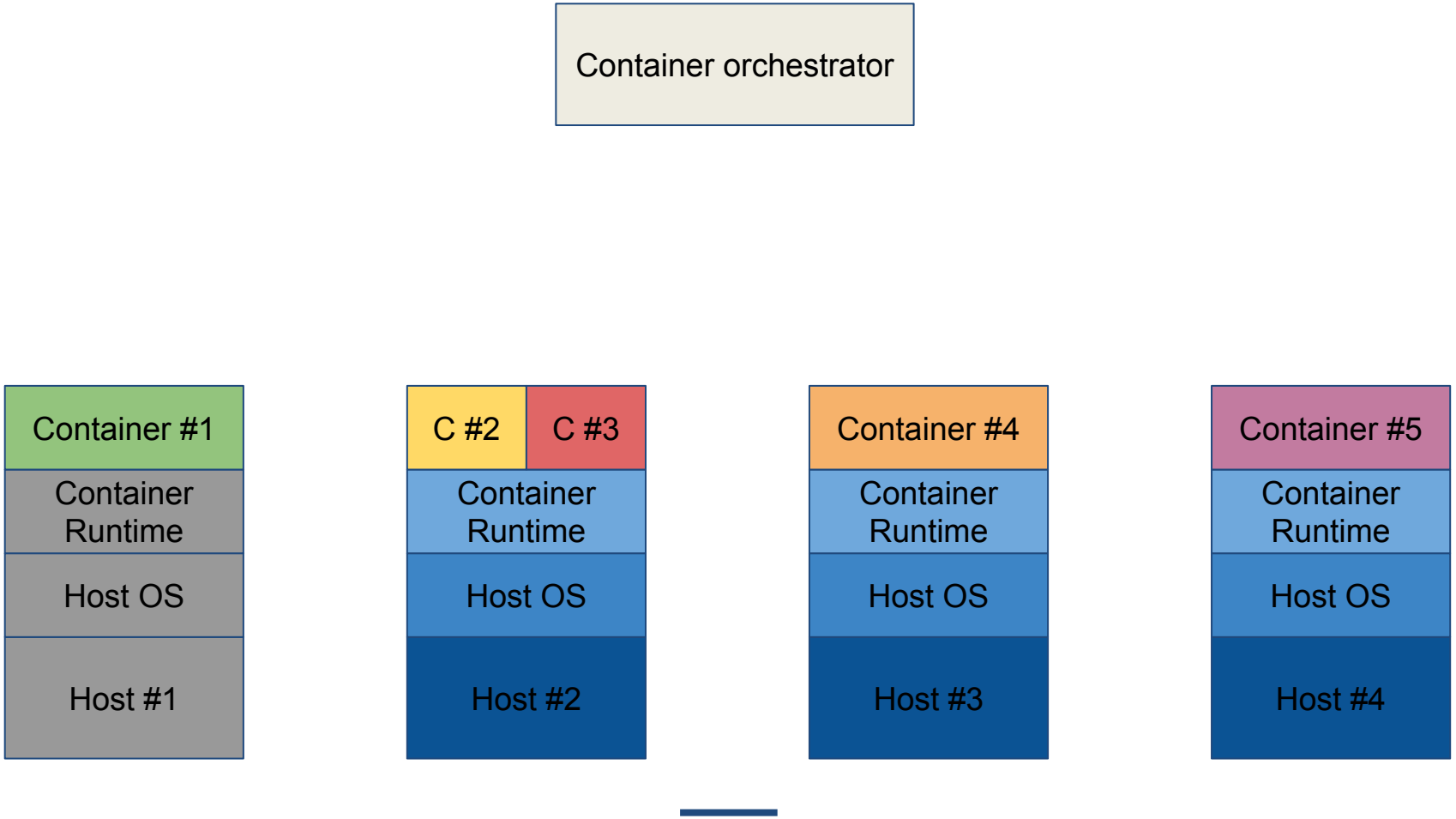
Container Orchestration



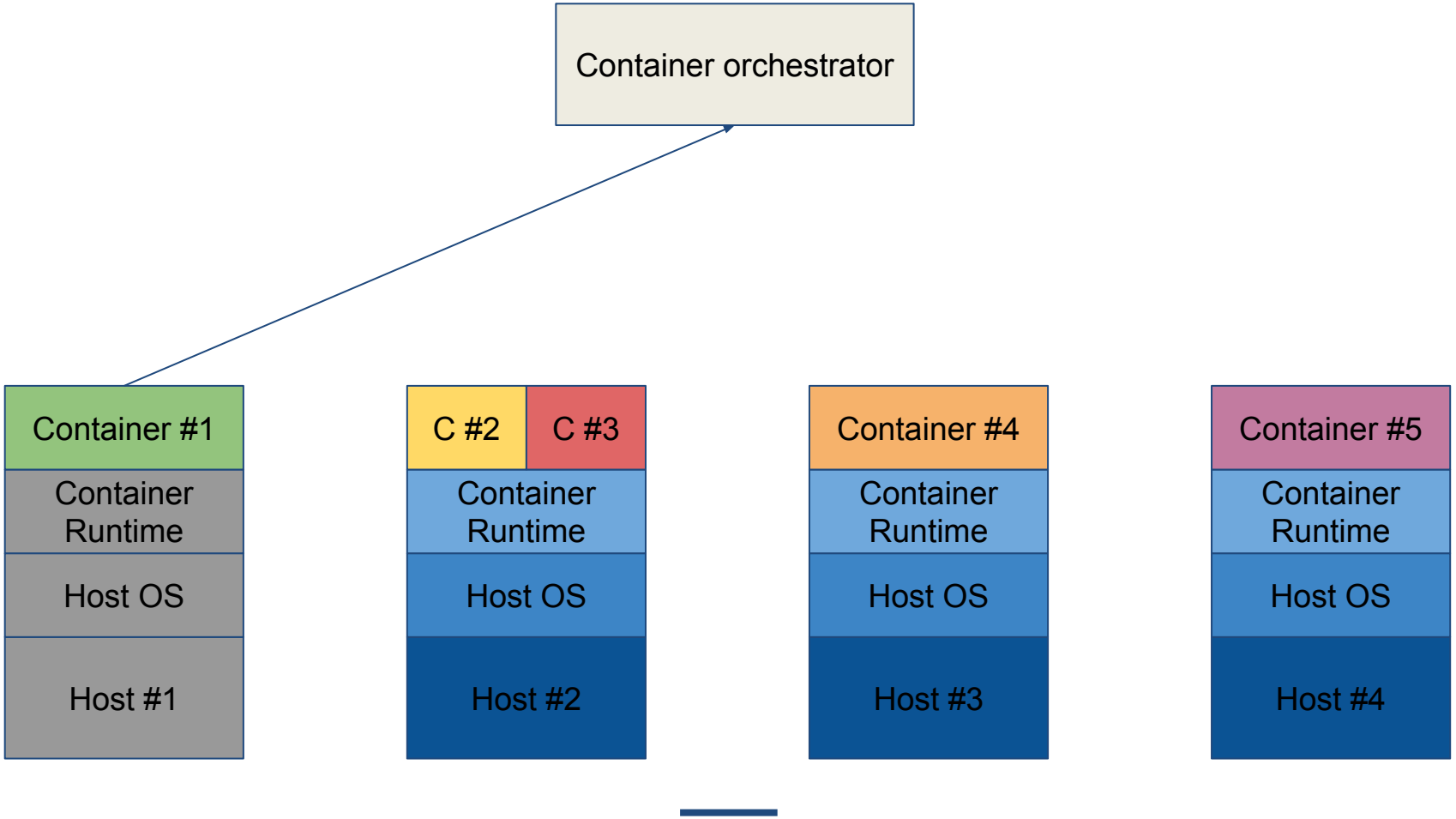
Container Orchestration



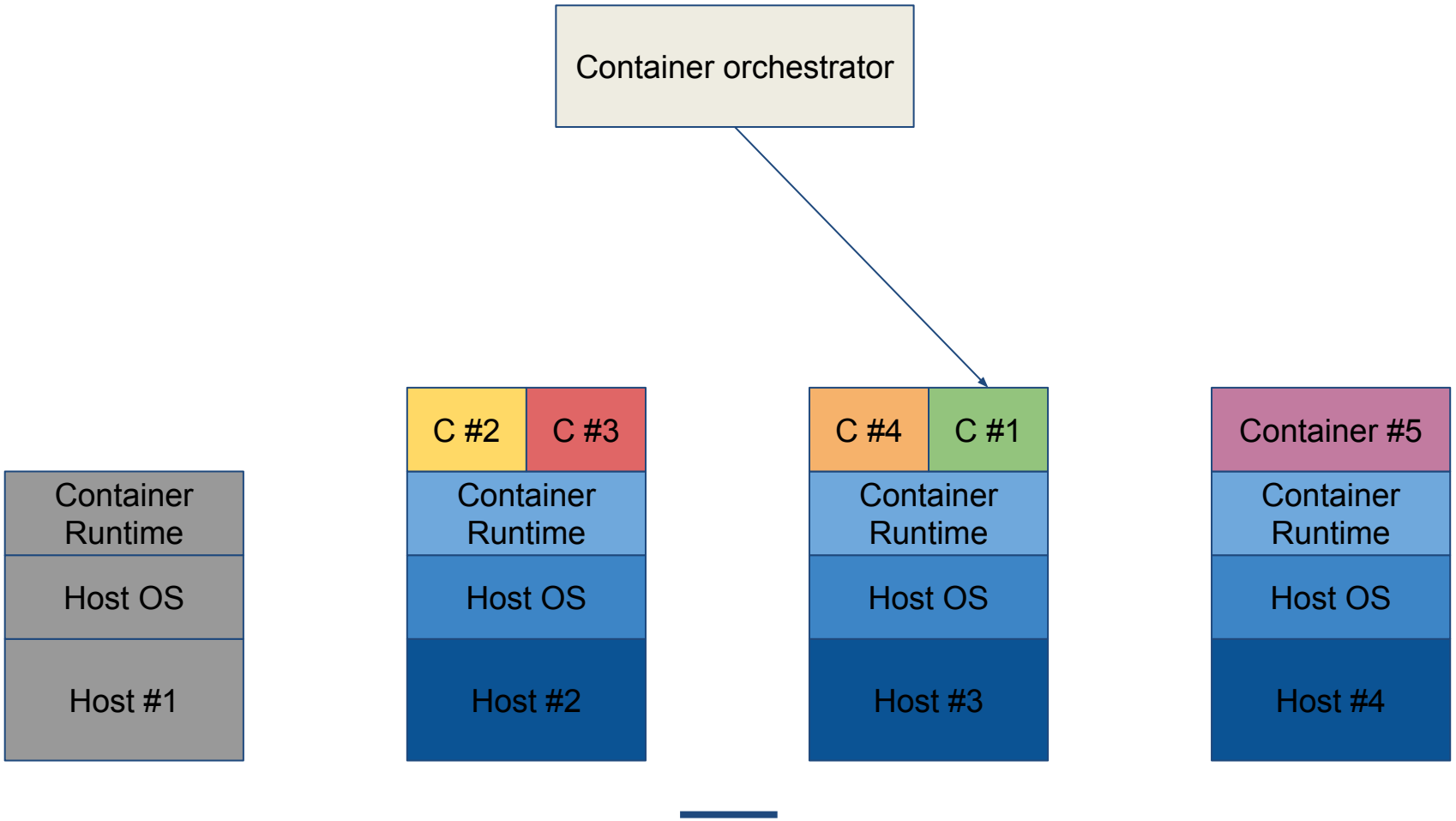
Container Orchestration



Container Orchestration



Container Orchestration



Kubernetes

A platform for managing containerised workloads and services.

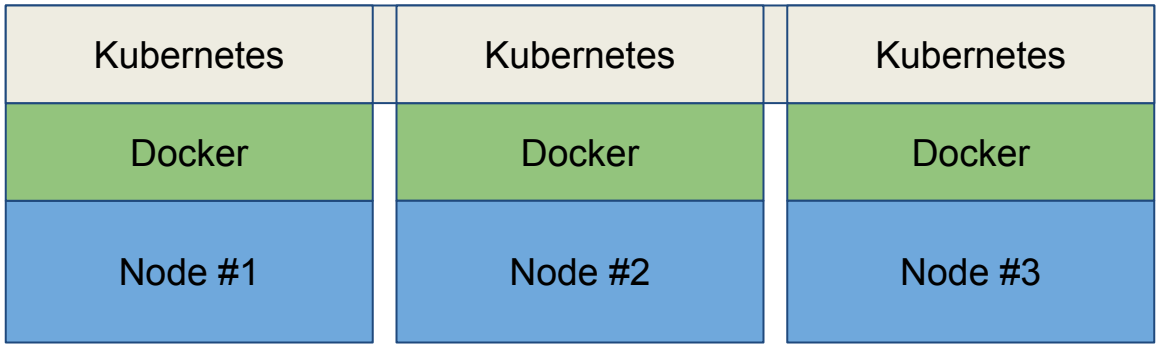
Runs a Container Runtime (Read: Docker)* underneath.

- Scalability
- Availability
- Reliability
- Recoverability
- Robustness

Kubernetes: Under the Hood

Kubernetes Nodes

A worker machine in the Kubernetes cluster.



Kubernetes Resources

Pods

Deployments

Services

Kubernetes Resources: Pods

Smallest deployable units.

Wrapper around container(s).

Popular model: “One-container-per-Pod”

Sidecar pattern.

Kubernetes Resources: Deployments

Resource that manages a replicated application, a Pod.

Each replica is represented by a Pod.

Distributes Pods among the cluster nodes.

Kubernetes Resources: Services

Way to expose an application running on a set of Pods as a network service.

Makes sure that network traffic can be directed to the current set of Pods for the workload.

Kubernetes Resources: Services

ClusterIP

- Exposes the Service on a internal IP.
- Makes the service reachable from within the cluster.

NodePort

- Kubernetes allocates a port in the 30.000 - 32.767 range.
- Each node proxies the (same) port into the Service.

Kubernetes Resources: Services

LoadBalancer

External load balancers that balance load between replicas.

Ingress

Not a Service type. Works with the “ClusterIP”-type.

Points to internal controllers that exposes the service.

Kubernetes Manifests

YAML or JSON files, defining creation, modification and deletion of Kubernetes resources.

Kubernetes Manifests: Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

Kubernetes Manifests: Service

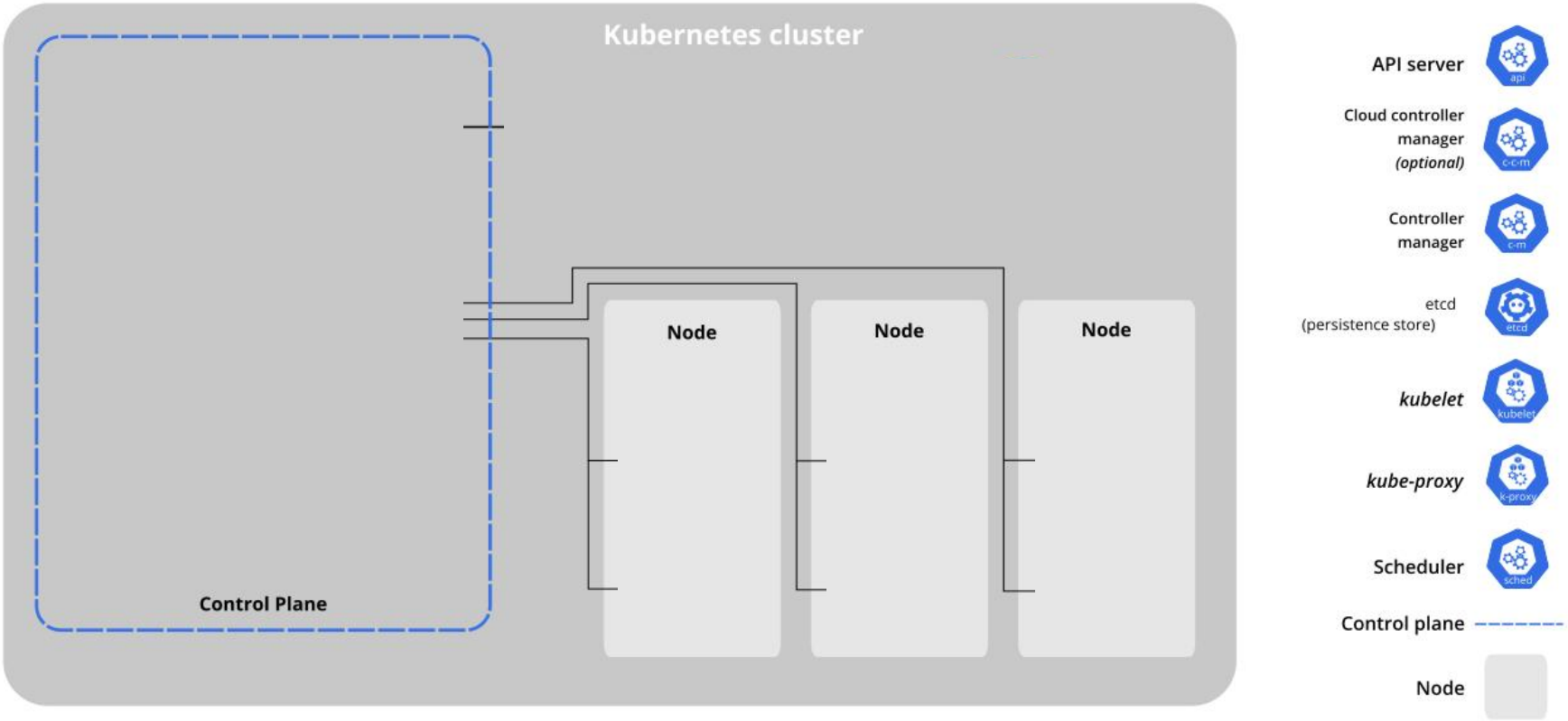
```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30007
```

Kubectl

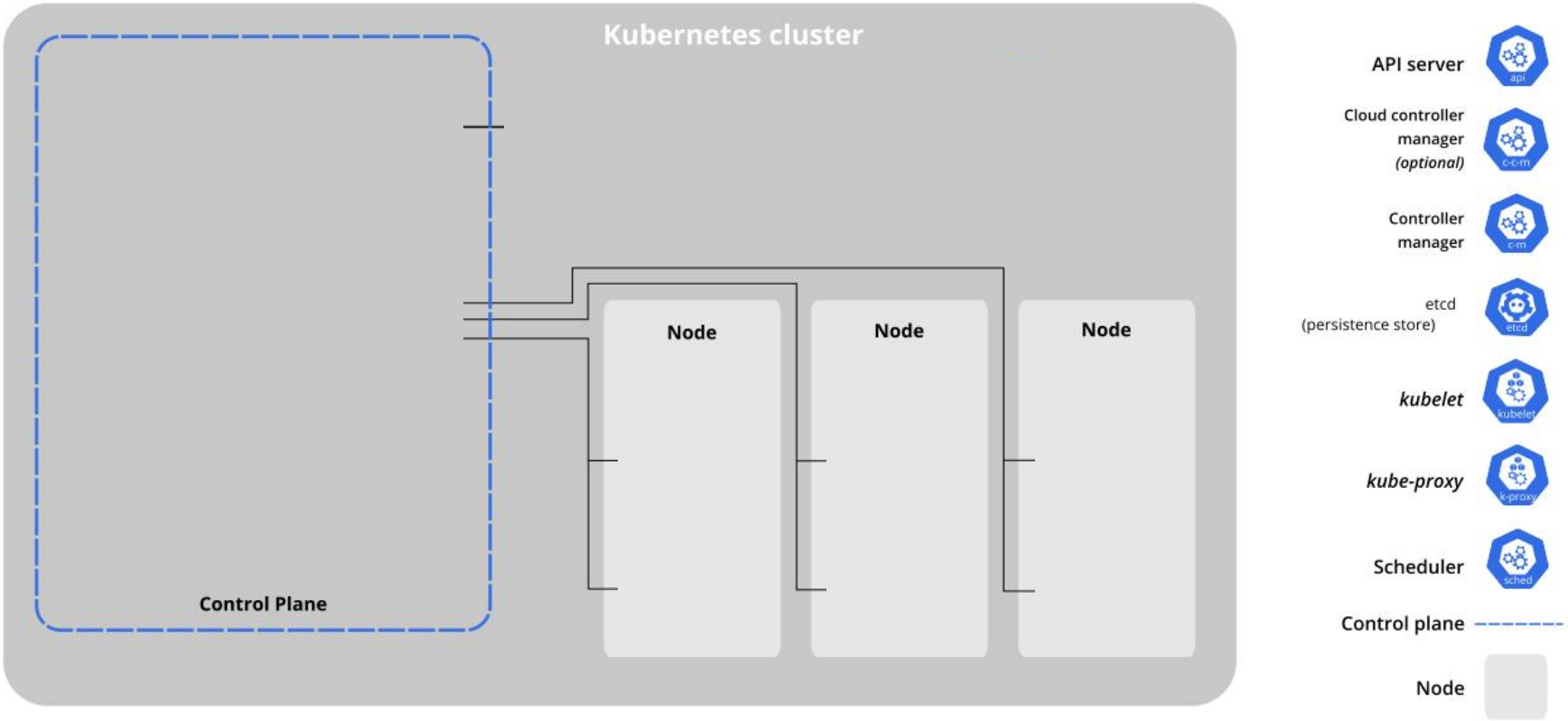
```
$ kubectl create -f [FILE]
$ kubectl apply -f [FILE]
$ kubectl delete -f [FILE]

$ kubectl get [RESOURCE] [NAME]
$ kubectl describe [RESOURCE] [NAME]

$ kubectl exec --stdin --tty [NAME] -- [COMMAND]
```

https://kubernetes.io/docs/reference/kubectl/cheatsheet/

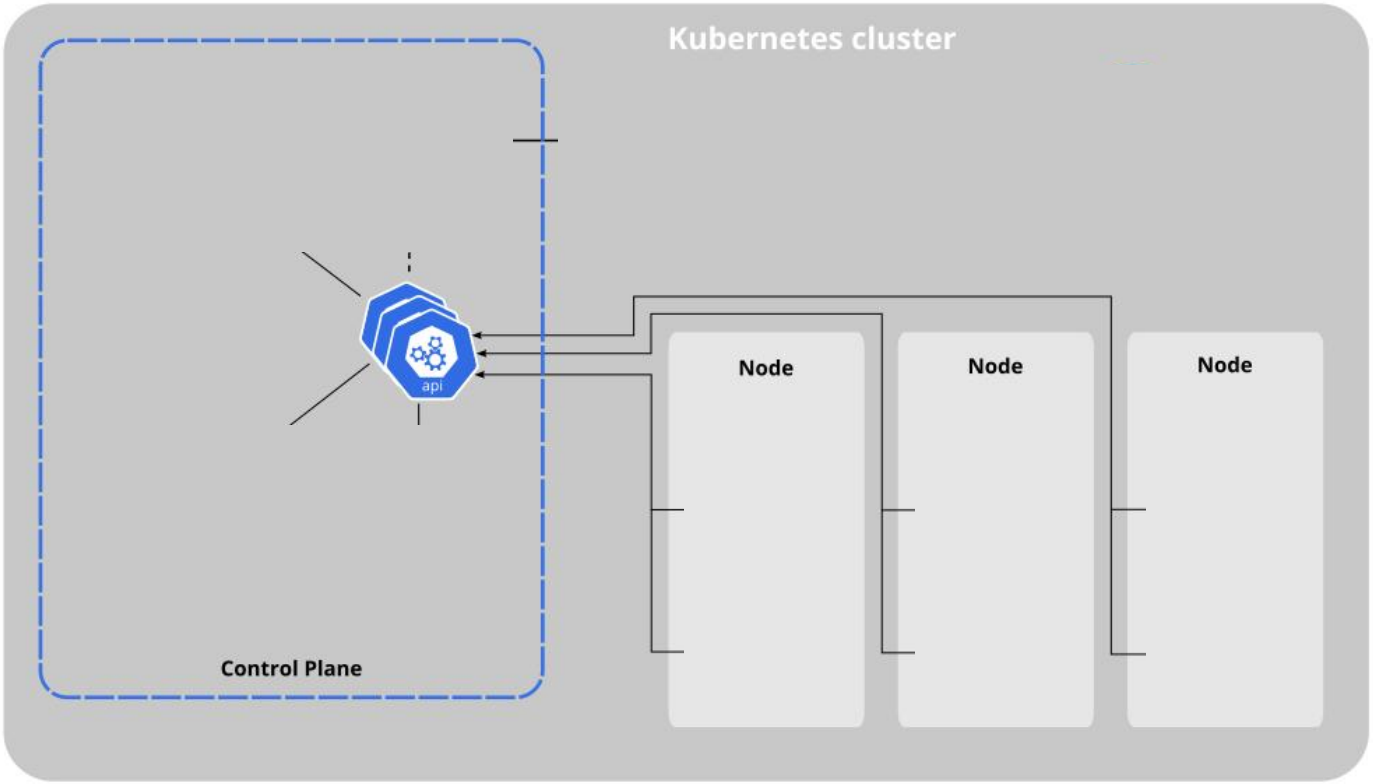
Kubernetes: Under the Hood



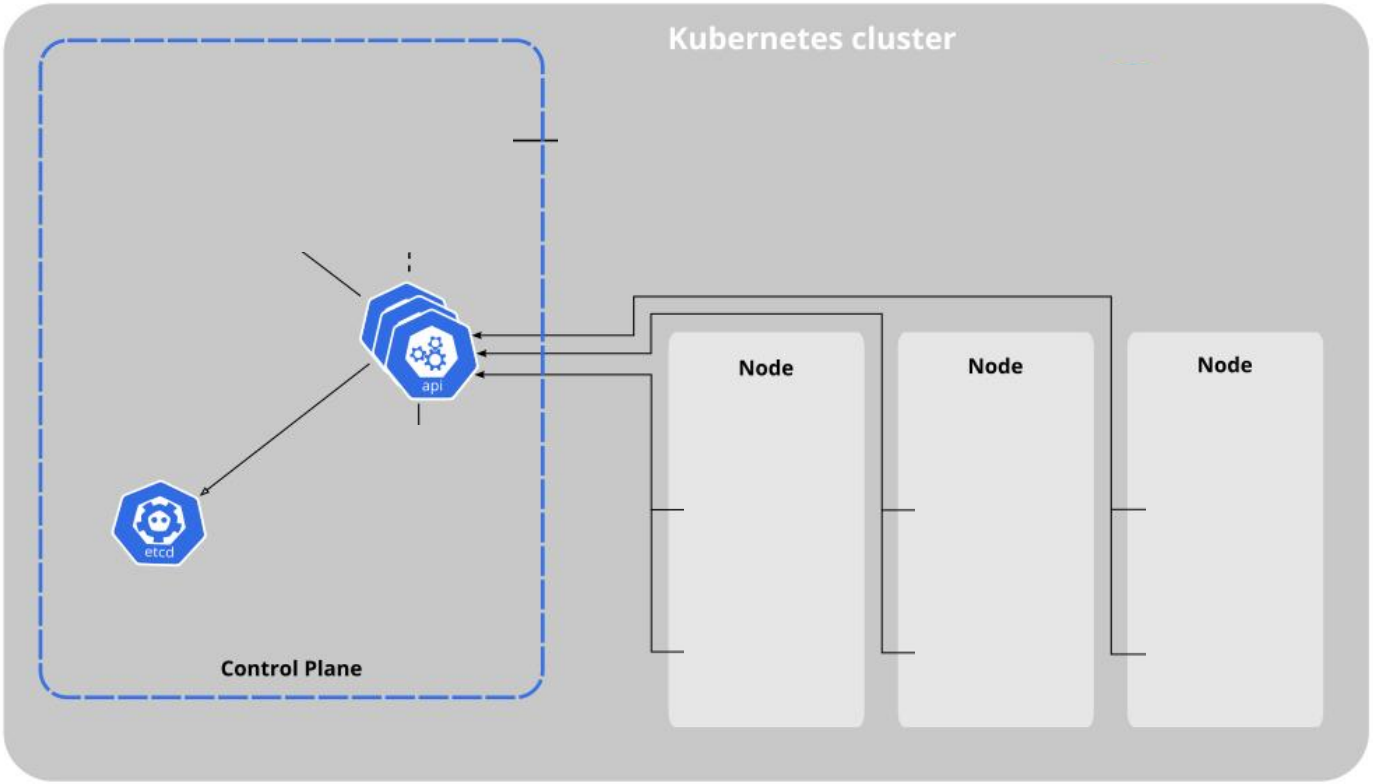
Kubernetes: Under the Hood



Kubernetes: Under the Hood



Kubernetes: Under the Hood



API server

Cloud controller manager *(optional)*

Controller manager

etcd (persistence store)

kubelet

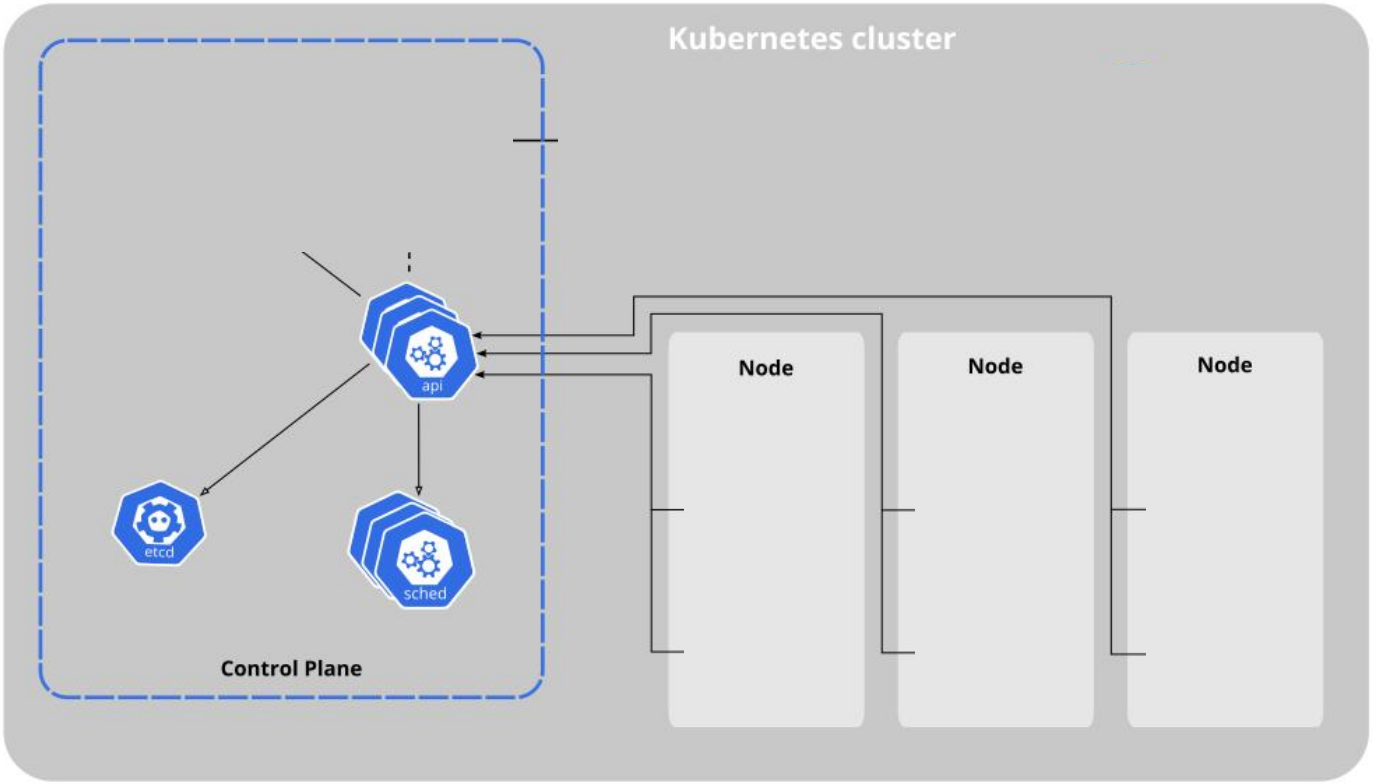
kube-proxy

Scheduler

Control plane

Node

Kubernetes: Under the Hood



API server

Cloud controller manager *(optional)*

Controller manager

etcd (persistence store)

kubelet

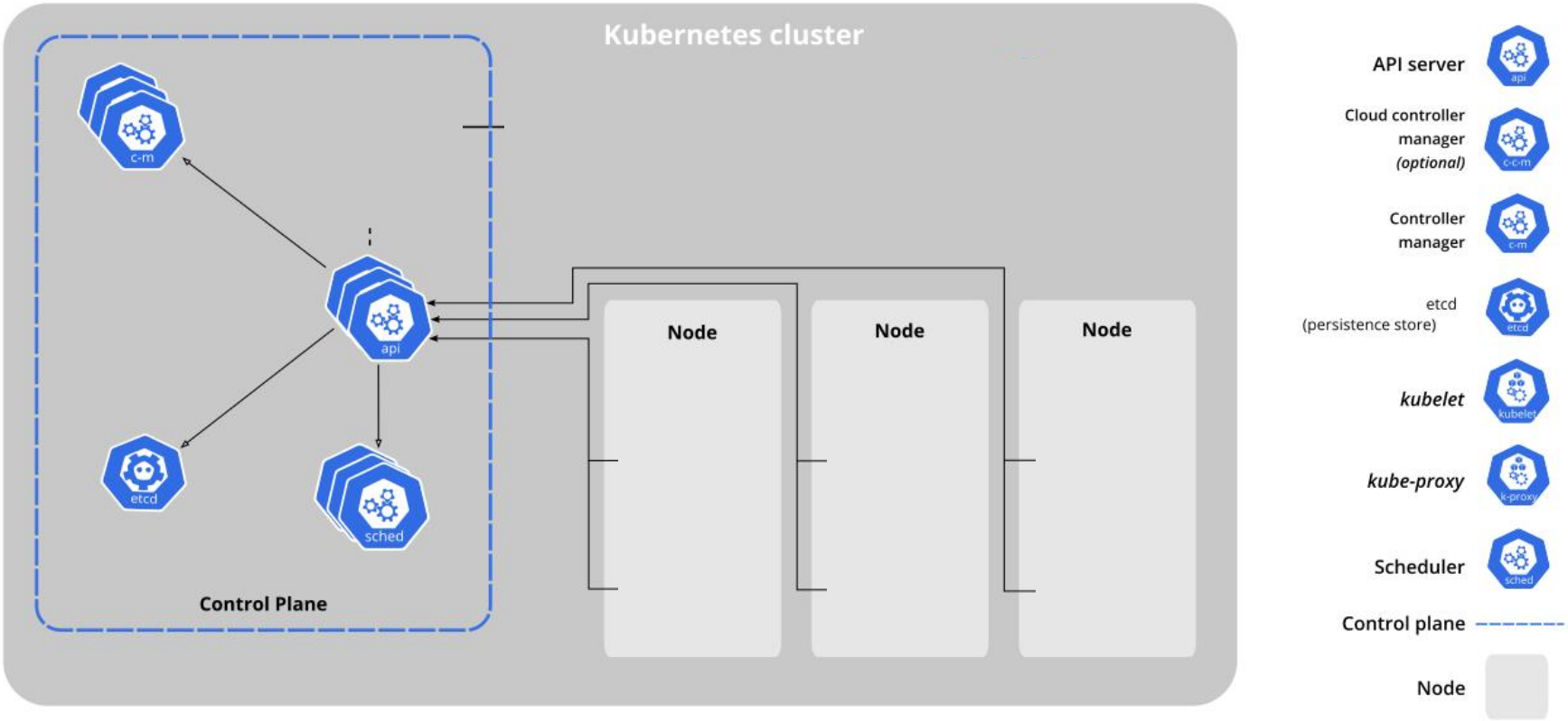
kube-proxy

Scheduler

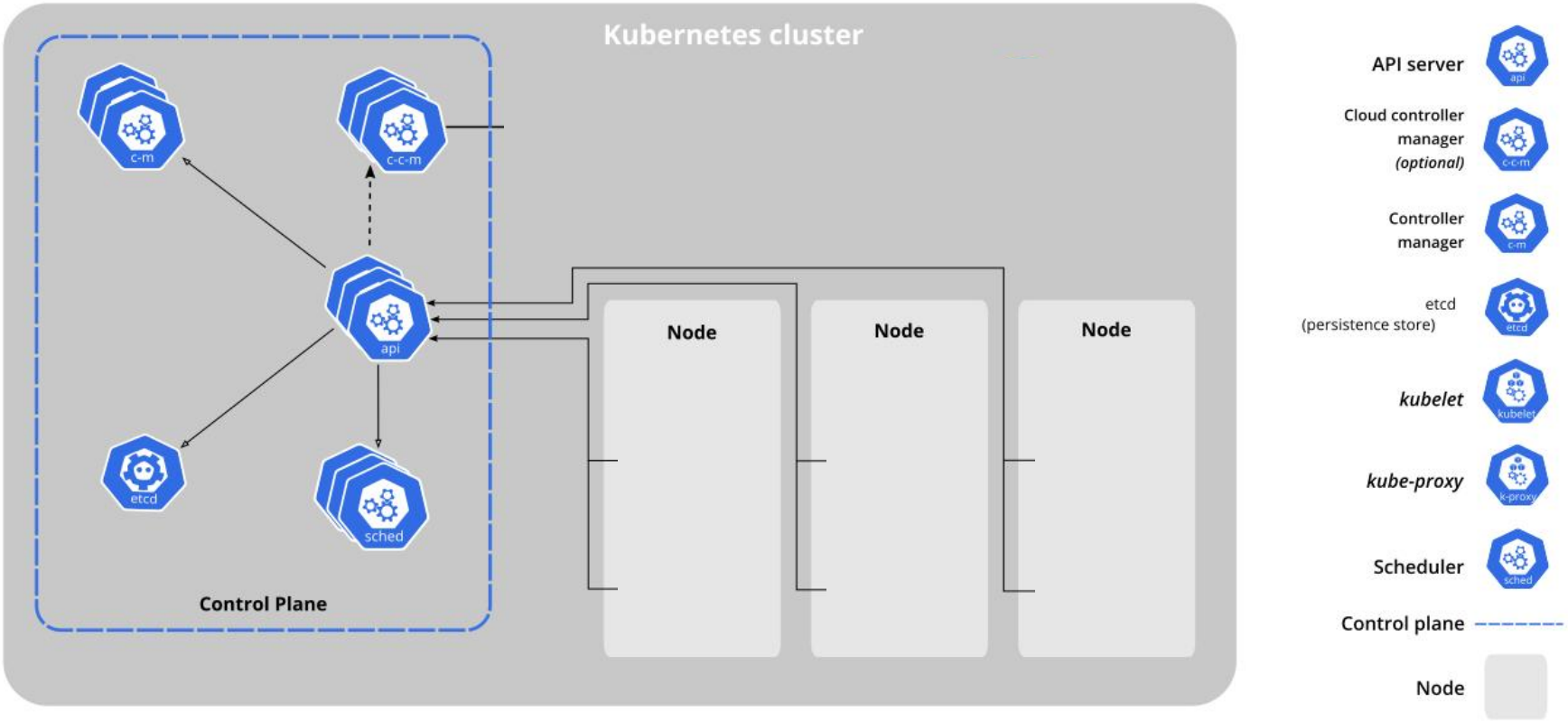
Control plane

Node

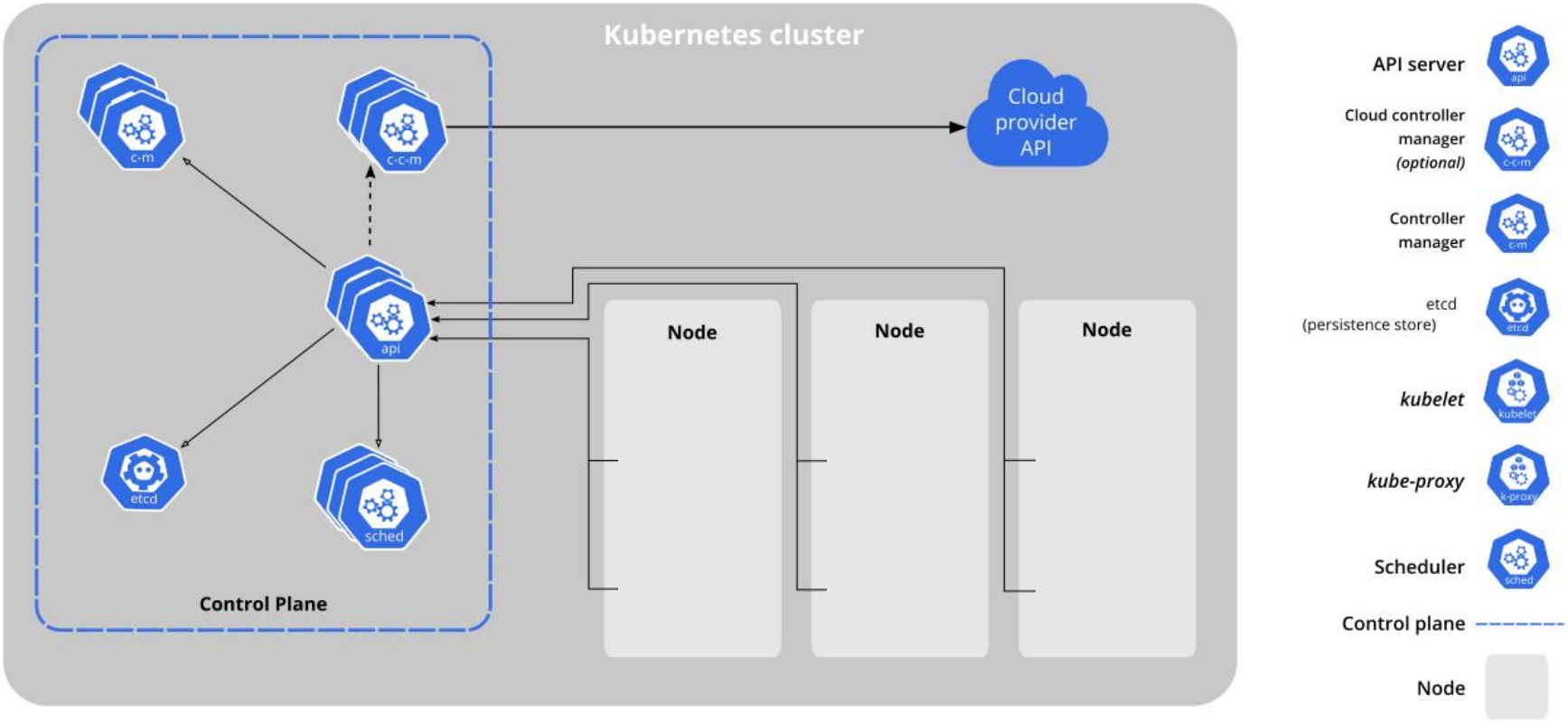
Kubernetes: Under the Hood



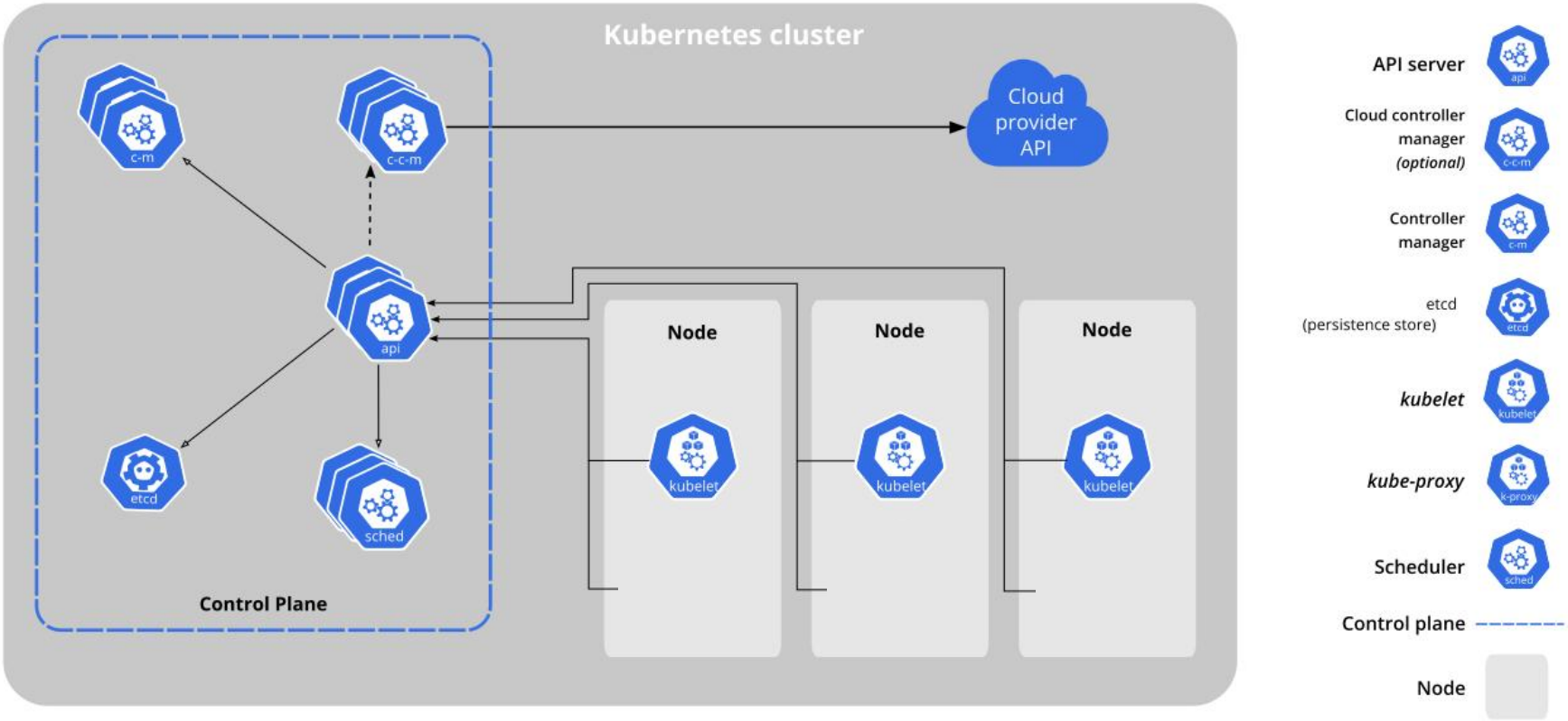
Kubernetes: Under the Hood



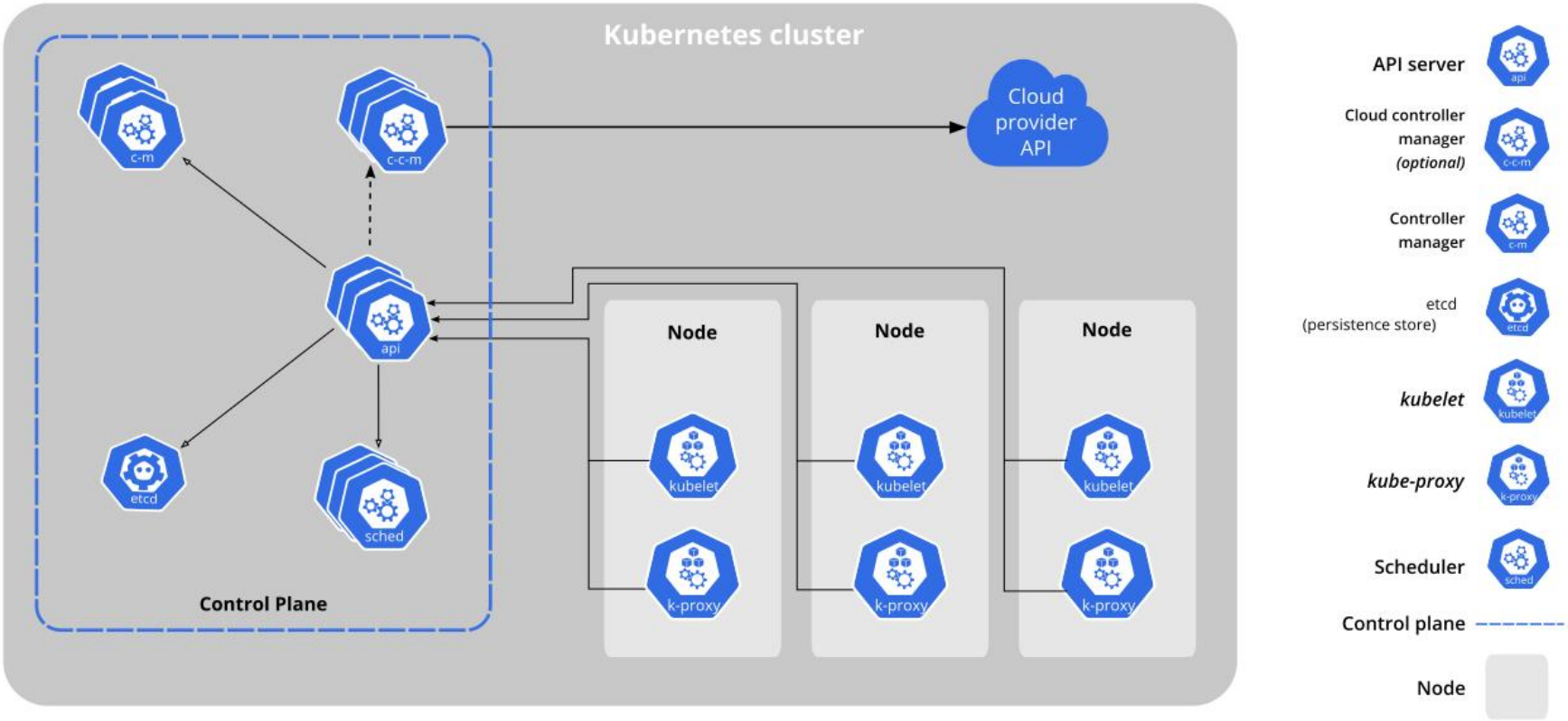
Kubernetes: Under the Hood



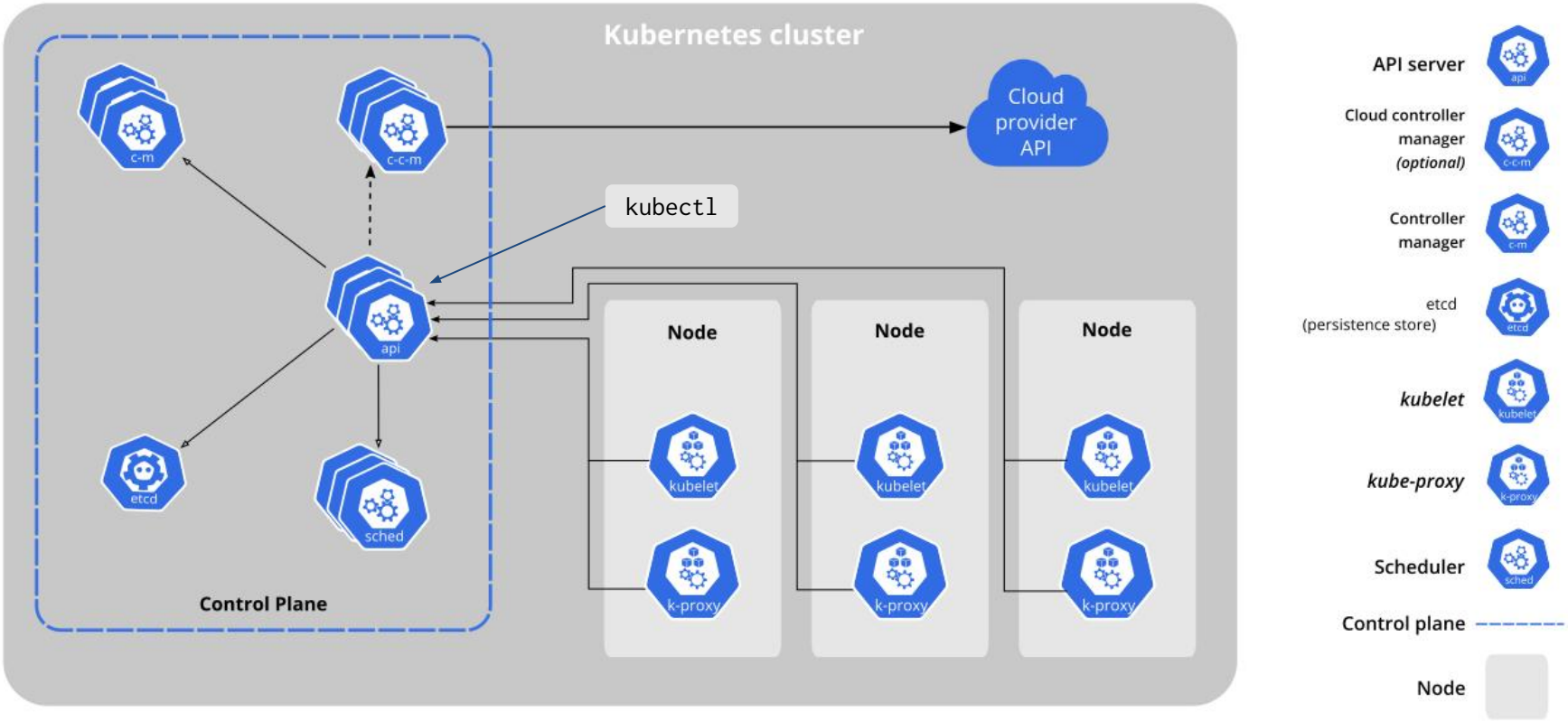
Kubernetes: Under the Hood



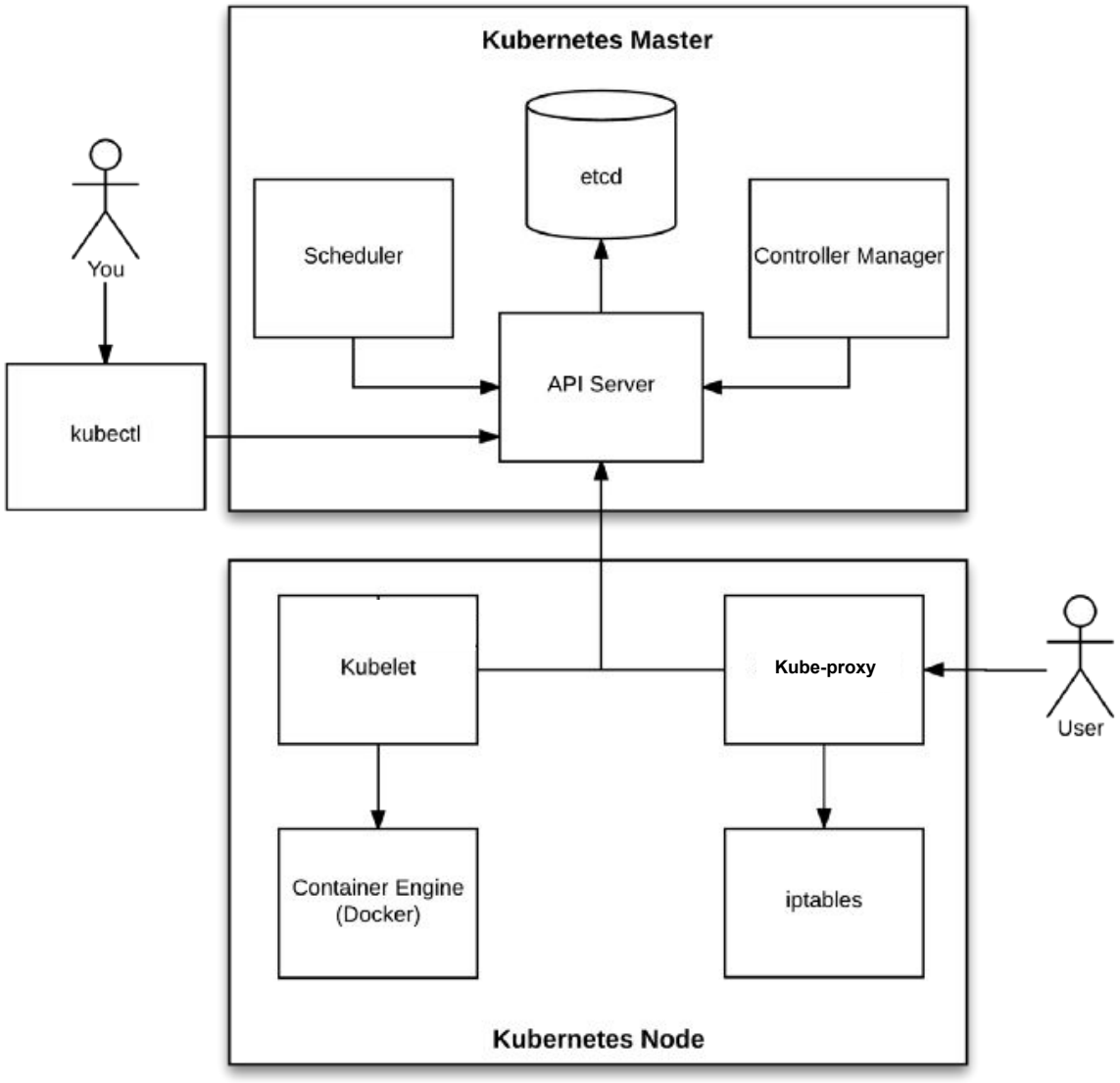
Kubernetes: Under the Hood



Kubernetes: Under the Hood



K8S


Kubernetes Master
etcd
Scheduler
Controller Manager
You
API Server
kubectl
Kubelet
Kube-proxy
User
Container Engine
(Docker)
iptables
Kubernetes Node

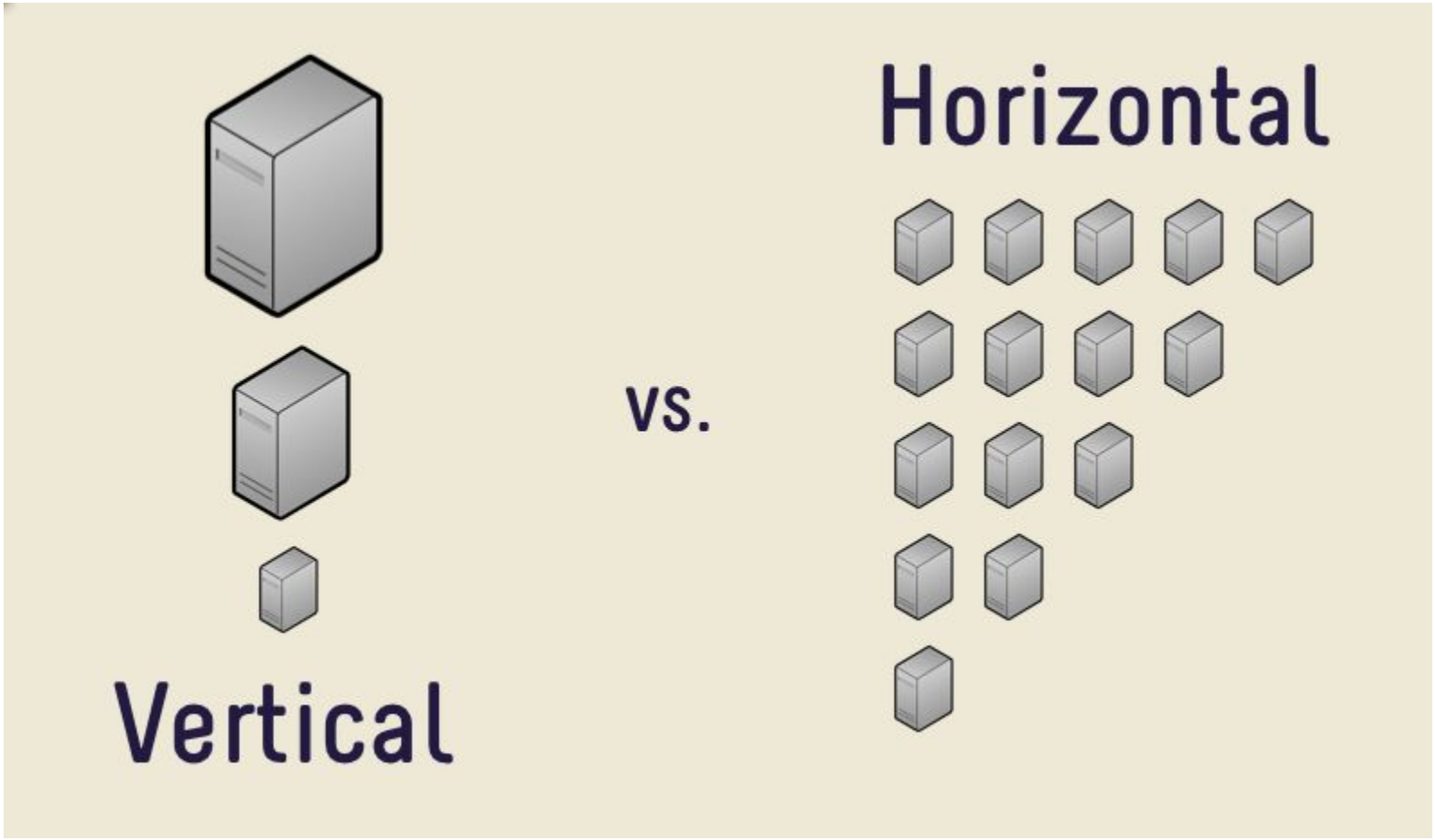
Kubernetes: Features

- Name spacing
- Load Balancing
- Auto Scaling
- Health-checks
- Rollout (and Rollback) Strategies

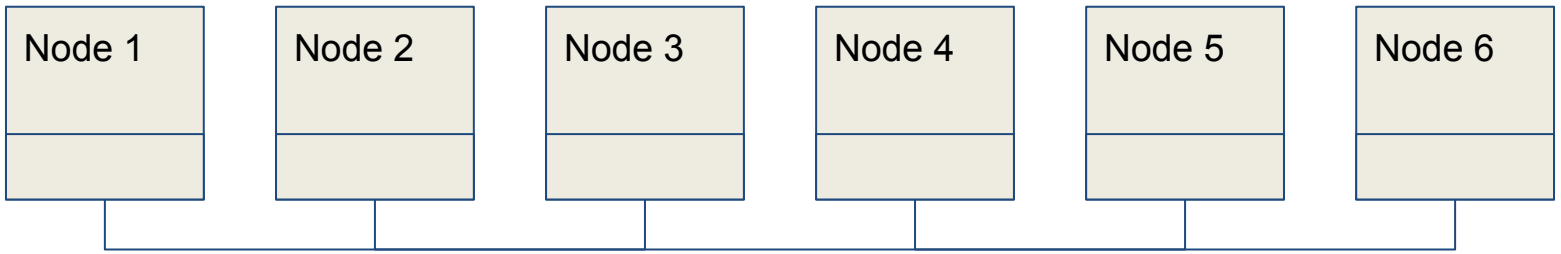
K8S

- On-premise
- Cloud
  - AWS
  - GCP
  - Azure
  - …
- Cloud features:
  - Load balancer
  - Monitoring
  - Scalability
  - …

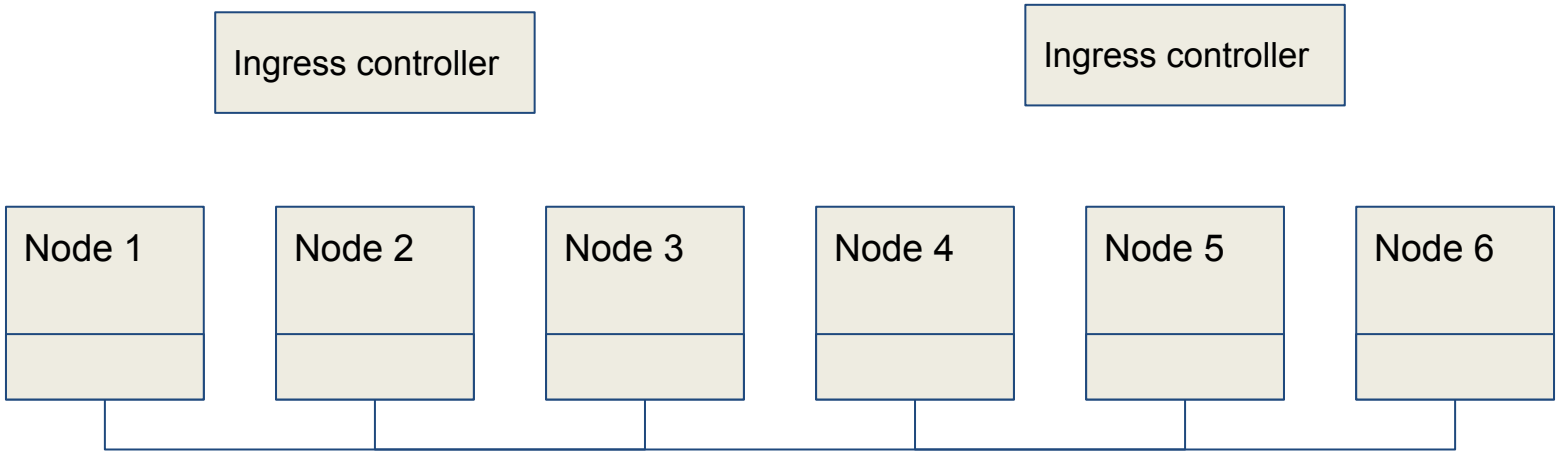
Scaling



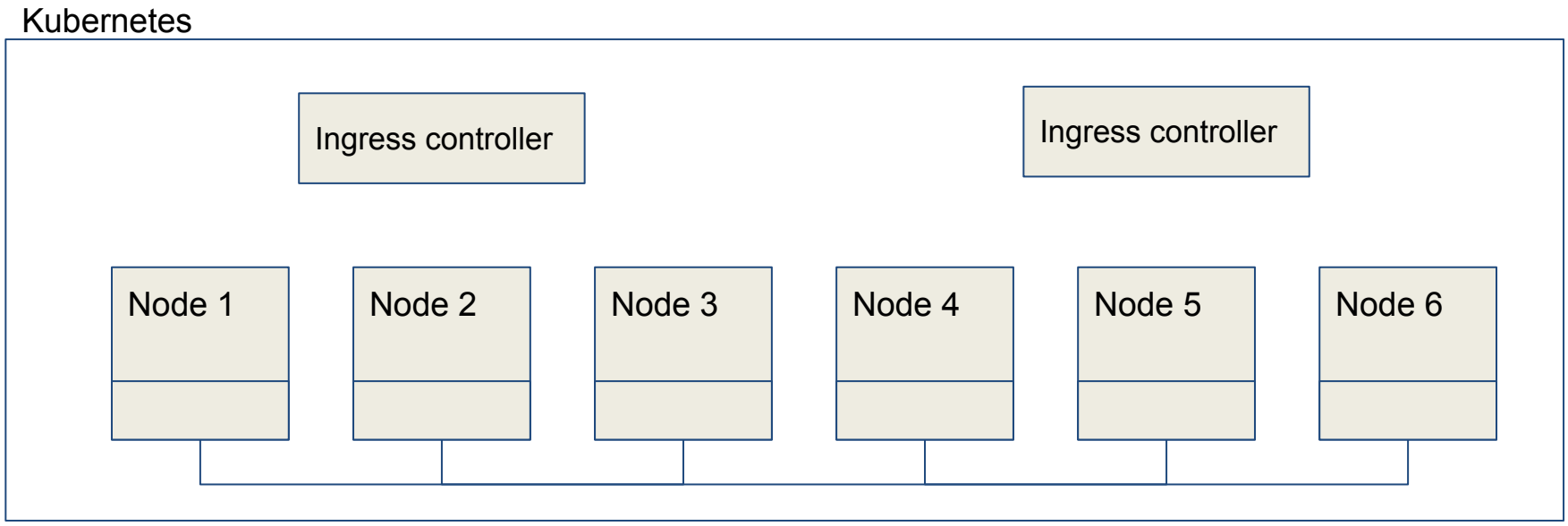
Kubernetes Setup



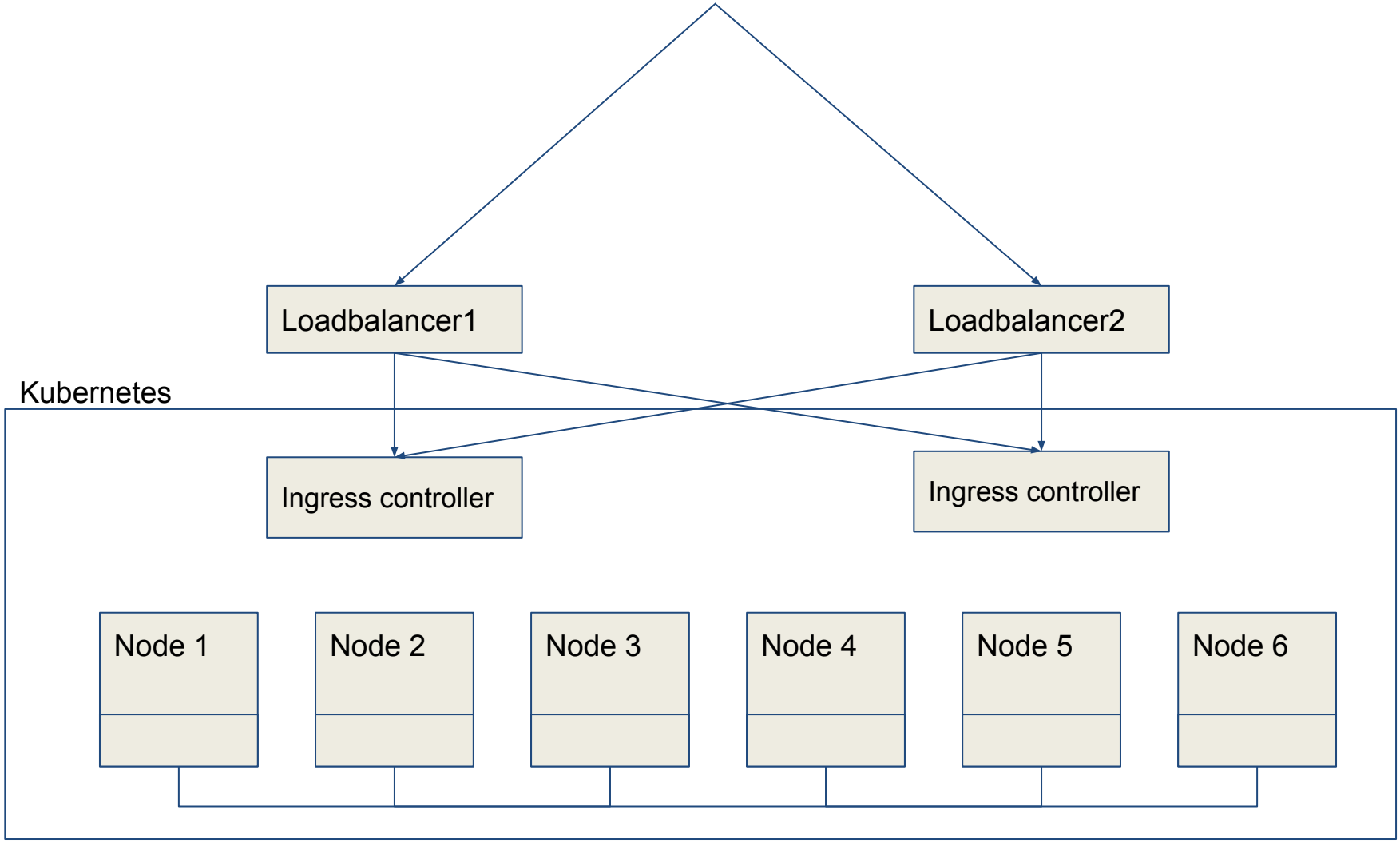
Kubernetes Setup



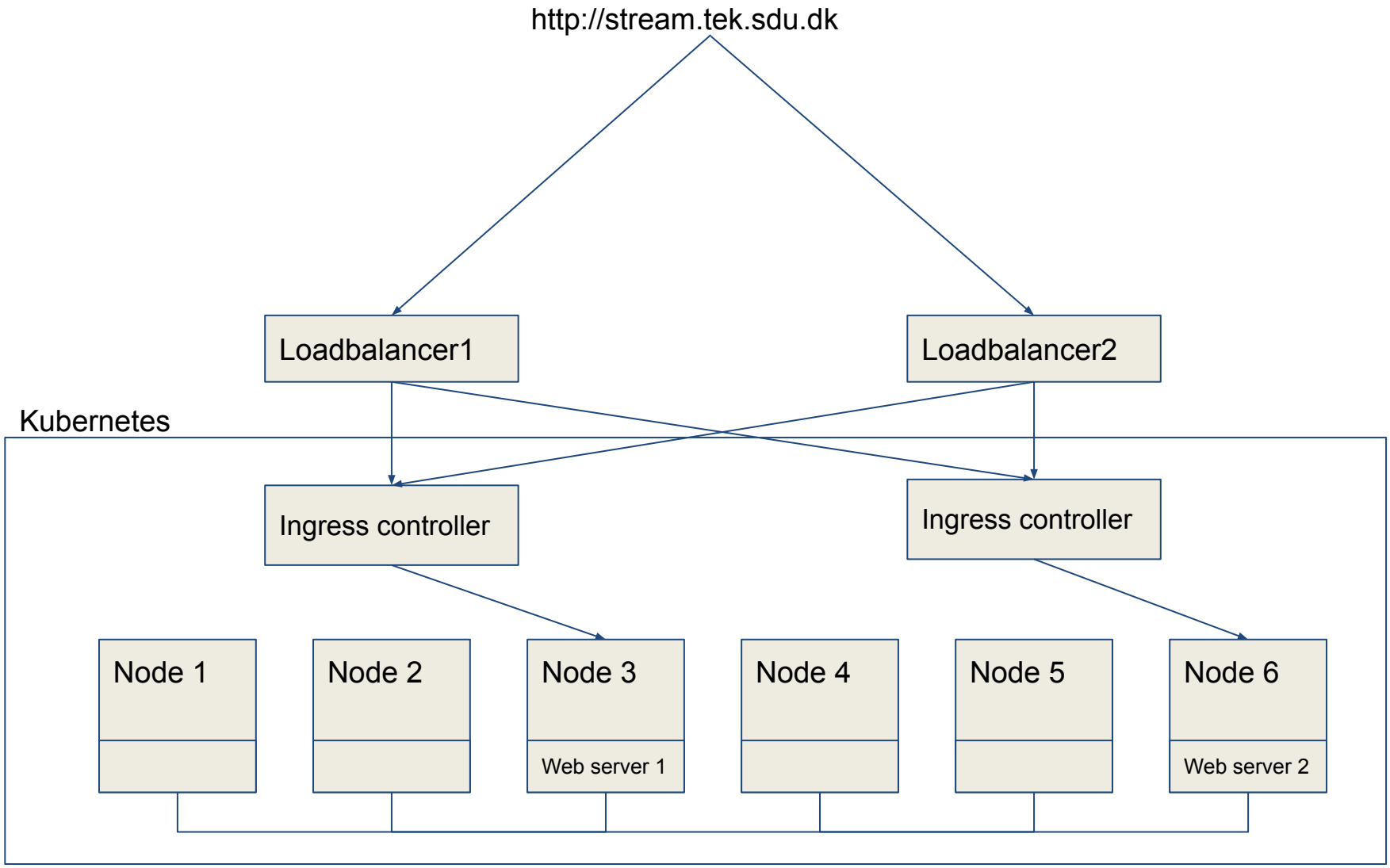
Kubernetes Setup



Kubernetes Setup



Kubernetes Setup



#random, #odw

Do read #random on slack!

- Nothing important
- Links
- Tips'n tricks

#odw

- A place for you to ask questions to us/students

  Link will be on blackboard later today

Group presentations

	IT3/SB3 Fall 2019: Assignments and mandatory activities						
Week	**CCM**	**DES**	**IAS**	**OPN**	**DM547**	**Project**	**Evaluation**
36	Announced Assignment 1 handed out					Group formation. Matching of expectations. Semester and project take off	
37	No class			Lab presentations by groups 1-4		Project proposal submission and approval, Friday Sep 13th	
38				Lab presentations by groups 5-8			
39				Lab presentations by groups 9-12	Assignment 1*, due Friday afternoon	Project foundation submission, Friday Sep 27th	
40		Assignment 1: Due in Class	Due	Lab presentations by groups 12-16		Project foundation review and approval. Reviews done by Oct. 4th. See blackboard for more.	

Group presentations

	IT3/SB3 Fall 2019: Assignments and mandatory activities						
Week	**CCM**	**DES**	**IAS**	**OPN**	**DM547**	**Project**	**Evaluation**
36	Announced Assignment 1 handed out					Group formation. Matching of expectations. Semester and project take off	
37	No class			Lab presentations by groups 1-4		Project proposal submission and approval, Friday Sep 13th	
38				[illegible] by [illegible]			
39				Lab [illegible] groups 9-[illegible]	Assignment 1*, due Friday [illegible]fternoon	Project foundation submission, Friday Sep 27th	
40		Assignment 1: Due in Class	Due	Lab presentations by groups 12-16		Project foundation review and approval. Reviews done by Oct. 4th. See blackboard for more.	

Group presentations

Lecture	Contents	Preparation before lecture
01 - Introductions Week 36 2019-09-04 13:00 - 17:00 Lecture: U45 Labs:	Basic lecture: • Introduction to the course • Containers Lab exercise: • Getting started with Docker Advanced lecture: • Docker swarm	• Go through both linux tutorials (shell and shell scripts) • Register your ambition level • Watch video about docker (12 min) • Read blogpost about virtual machines vs. containers Presentations • None
02 - Linux Week 37 2019-09-11 13:00 - 17:00 Lecture: U45 Labs:	Basic lecture: • Linux operating system Lab exercise: • Docker compose Advanced lecture: • Load balancing	• Read The Linux Command Line ch 2-4 • Read the Overview for docker-compose • Read the Quickstart ○ Make sure your understand the functionality docker-compose provides Presentations • Groups 1, 8, 15 and 23
03 - Network Week 38 2019-09-18 13:00 - 18:30 Lecture: U45 Labs:	Basic lecture: • Models, IP, routing Lab exercise: • Network setup, routing Advanced lecture: • Firewall	• Study and play with iproute2 (ip) html or pdf • Fx. ip addr show Presentations • Groups 2,9, 16 and 24
04 - Protocols Week 39 2019-09-25 13:00 - 17:00 Lecture: U45 Labs:	Basic lecture: • Protocols Lab exercise: • REST APIs Advanced lecture: • DNS, DNSSEC	• Study API Presentations • Groups 3, 10, 17 and 25

What is an operating system and what does it do?

Analogy

Analogy

Analogy

You = processor

Analogy

You = processor

Homework = task

Analogy

Analogy

Analogy

Door bell = interrupt

Analogy

Analogy

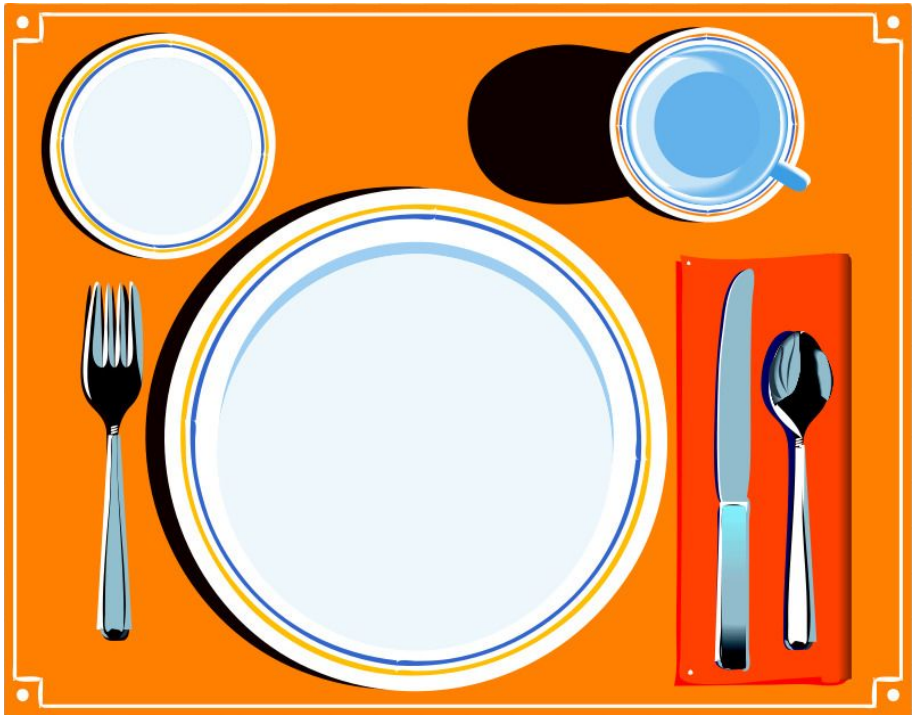
Pizza = dinner!

Analogy

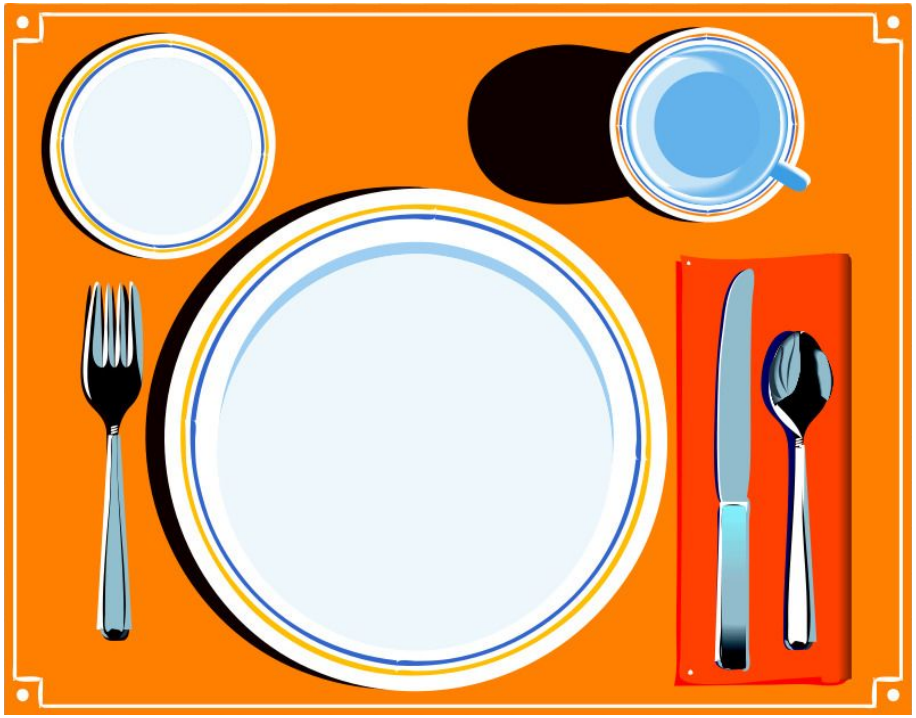
Pizza = dinner!
(actually it's a high priority task)

Analogy

Analogy



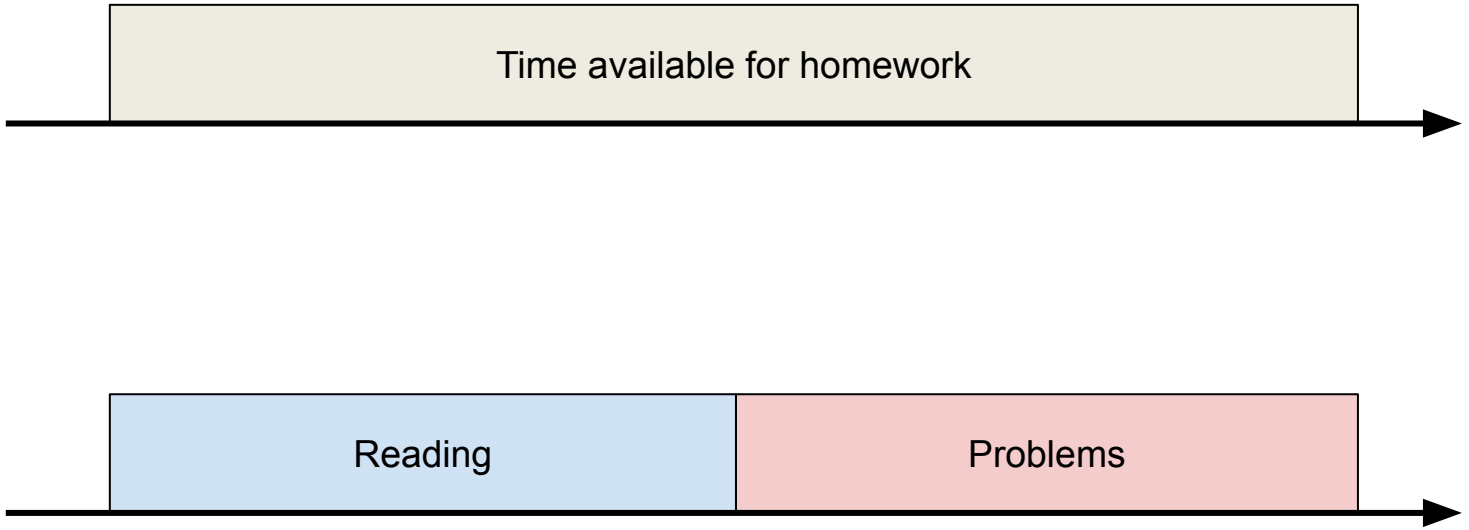
Analogy



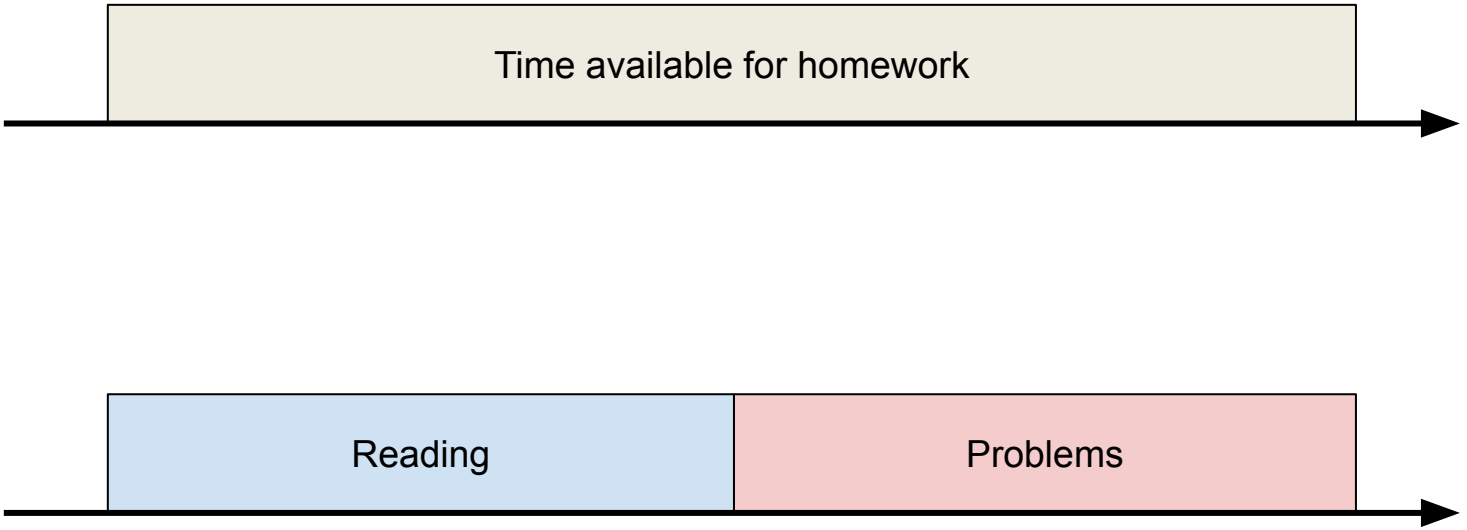
Context switch

Analogy

Analogy



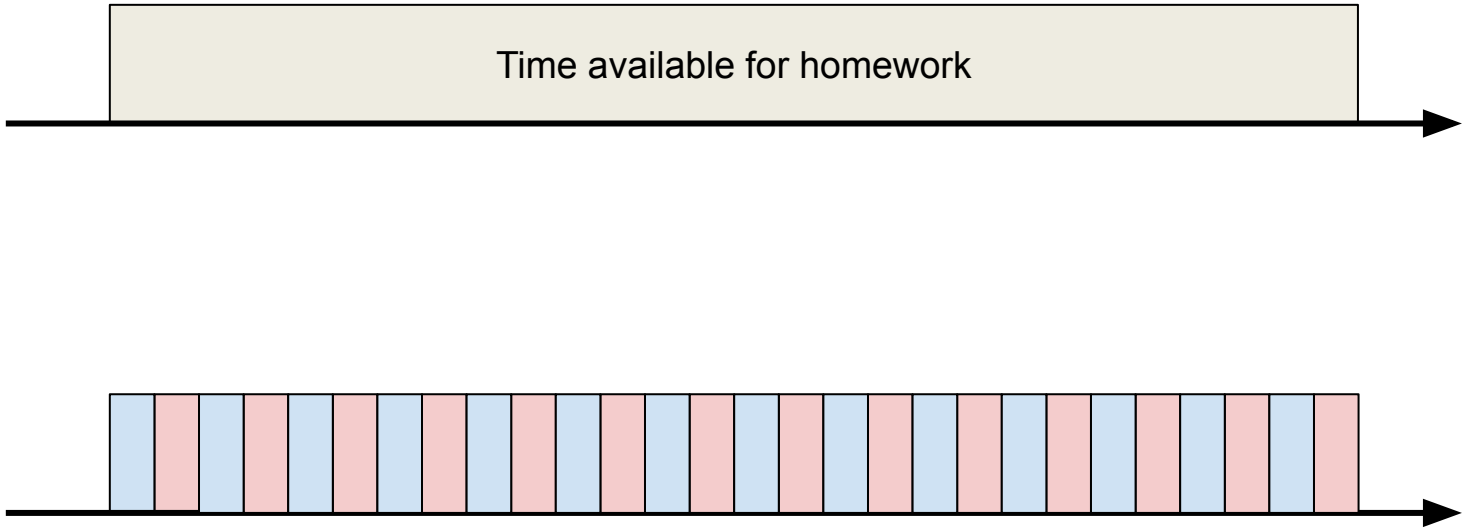
Analogy



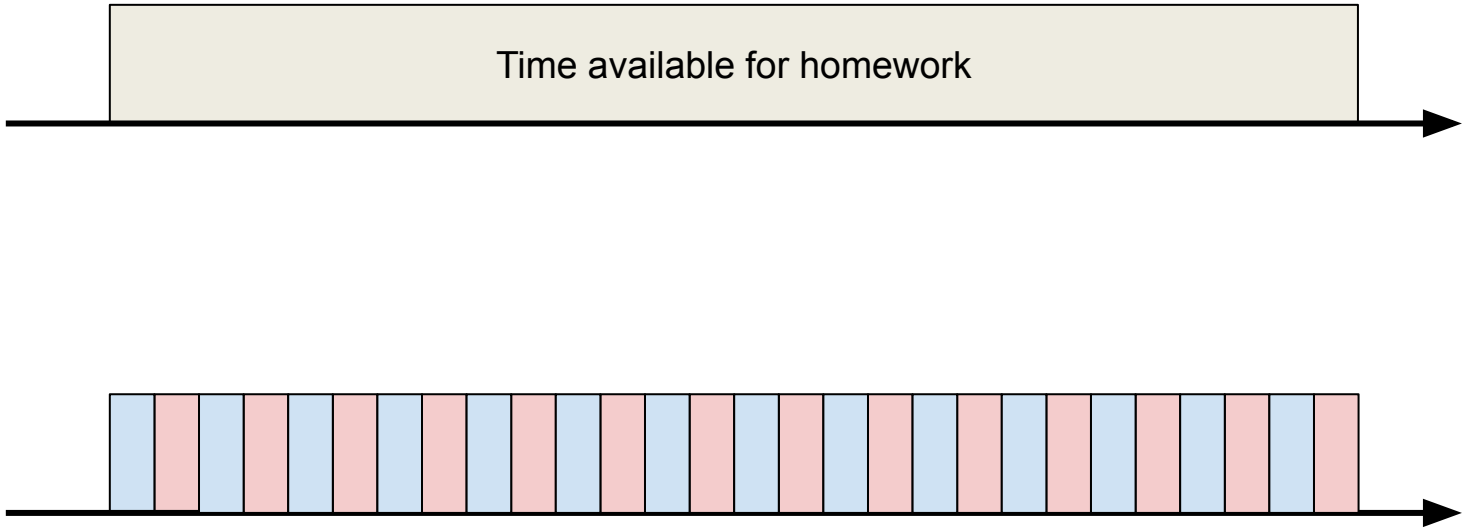
Scheduling

Analogy

Analogy

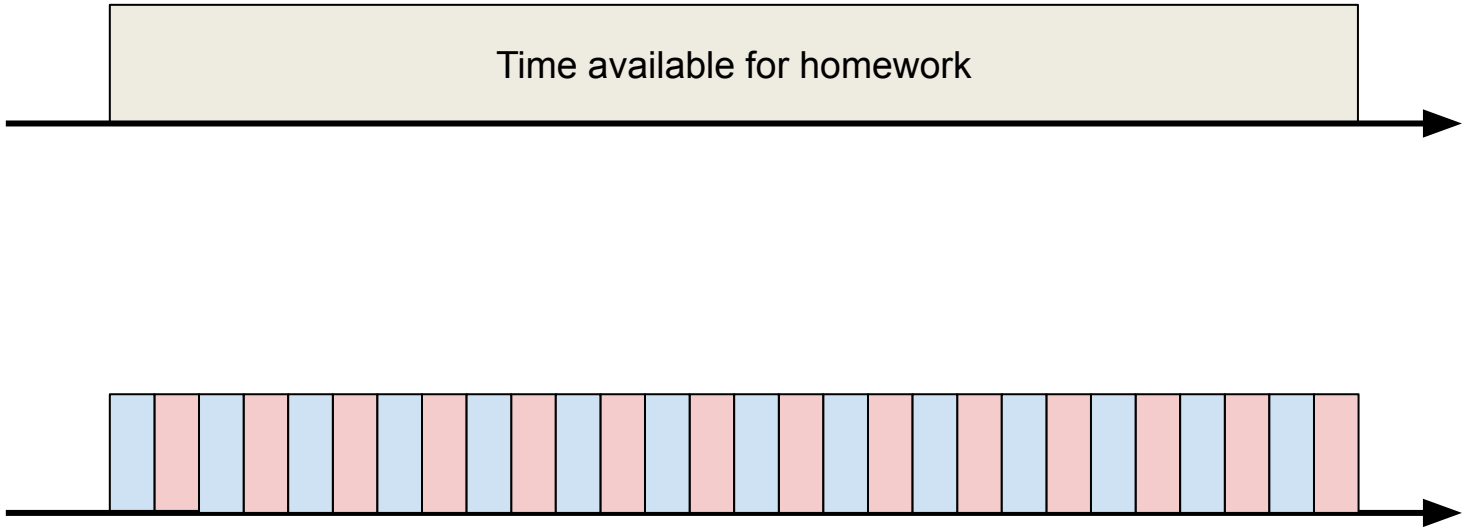


Analogy



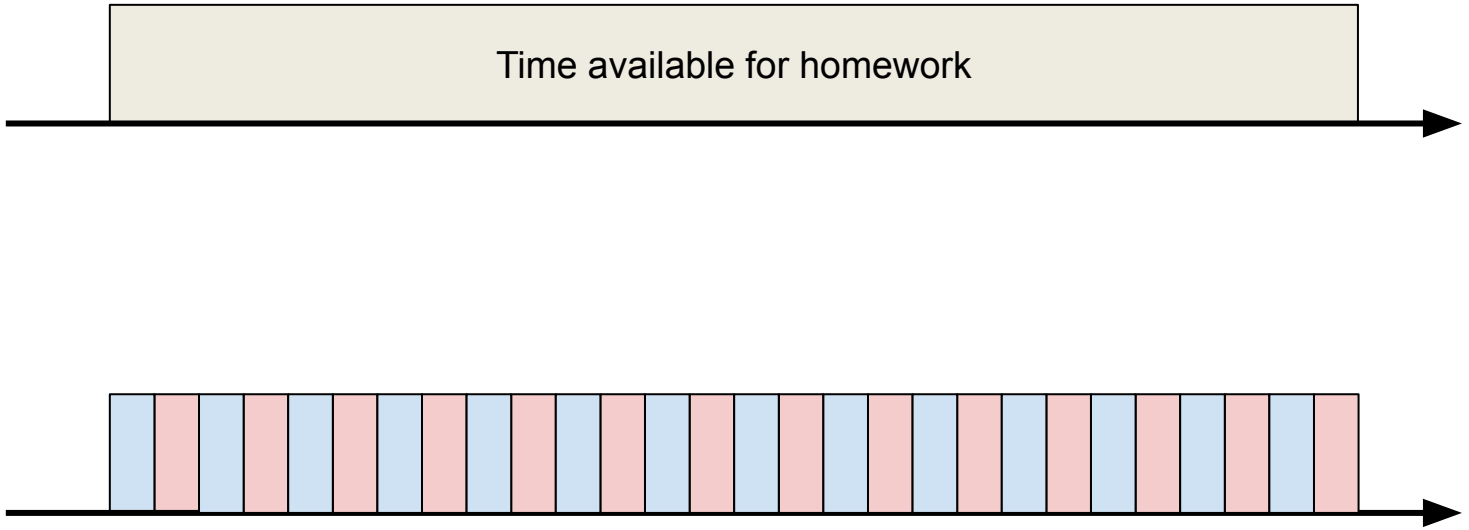
Time slicing

Analogy



Time slicing

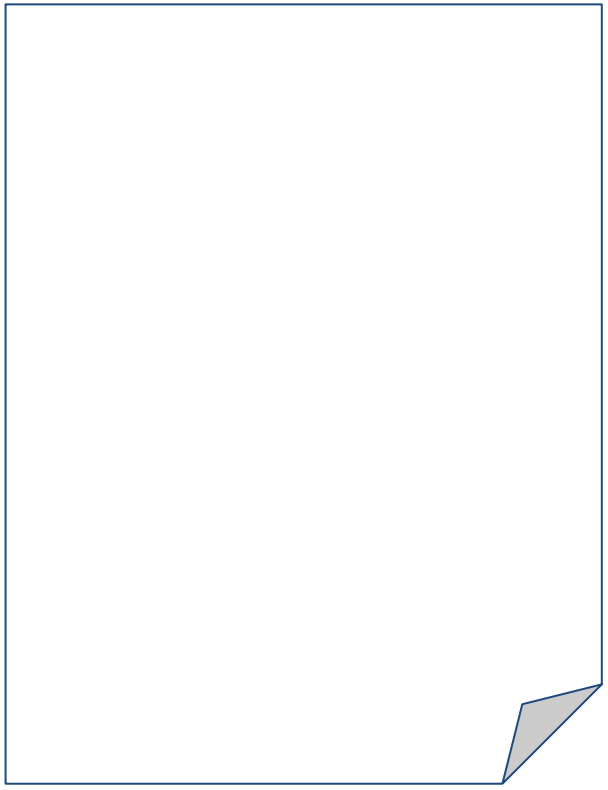
Analogy



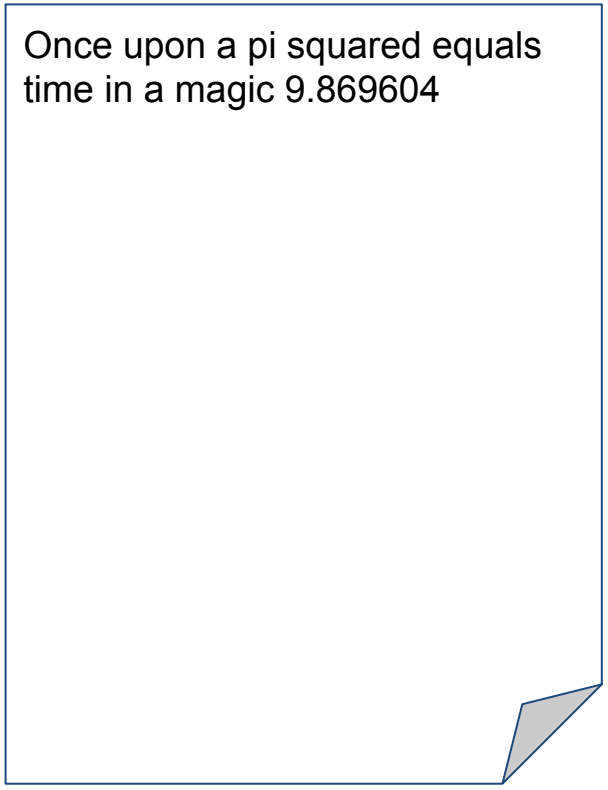
Time slicing
Overhead

Strategy

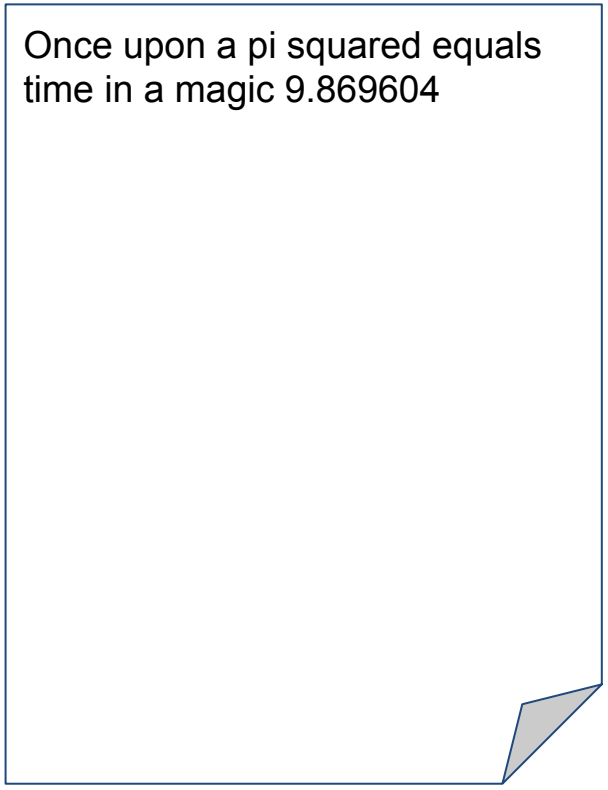
Analogy



Analogy



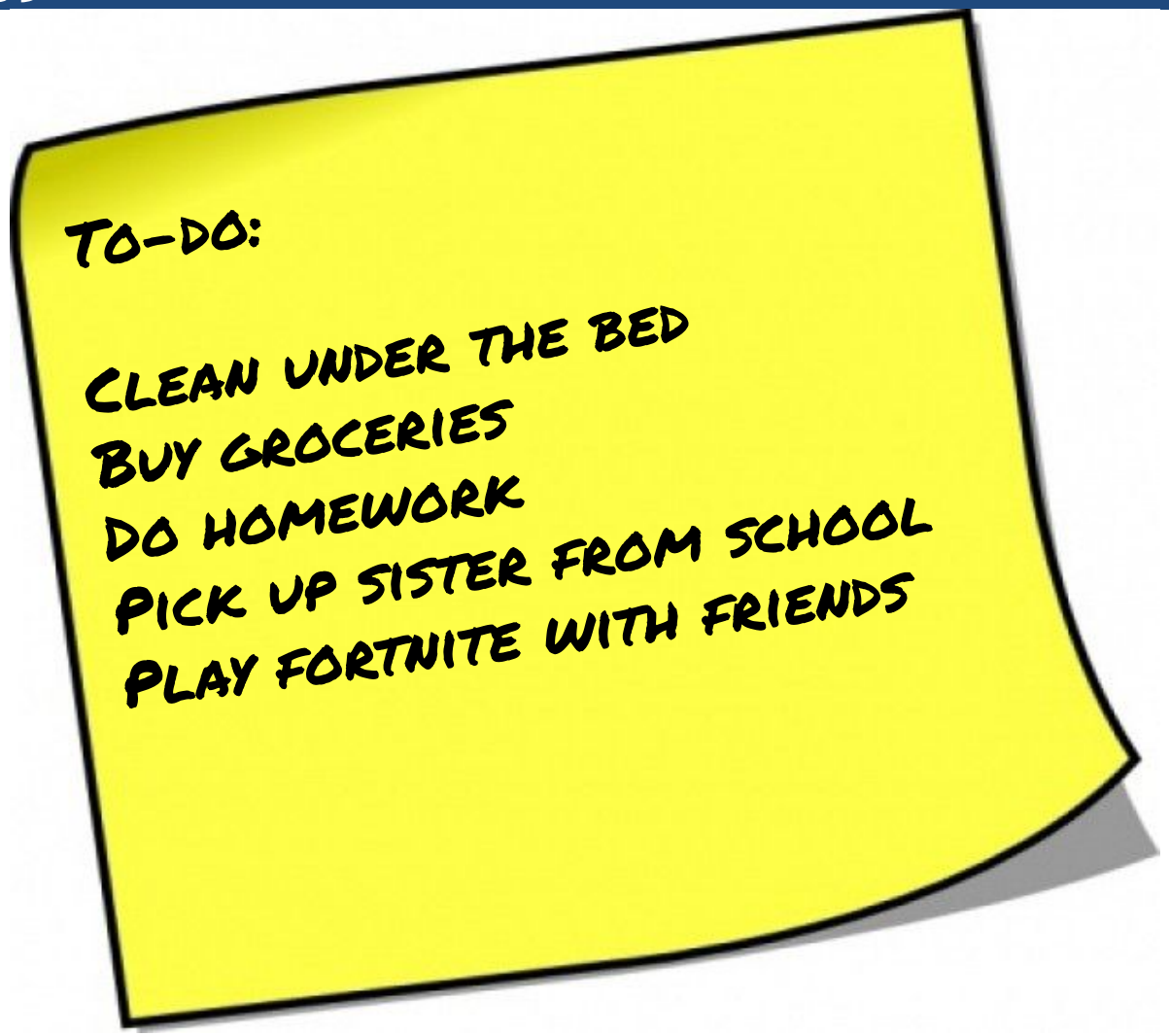
Analogy



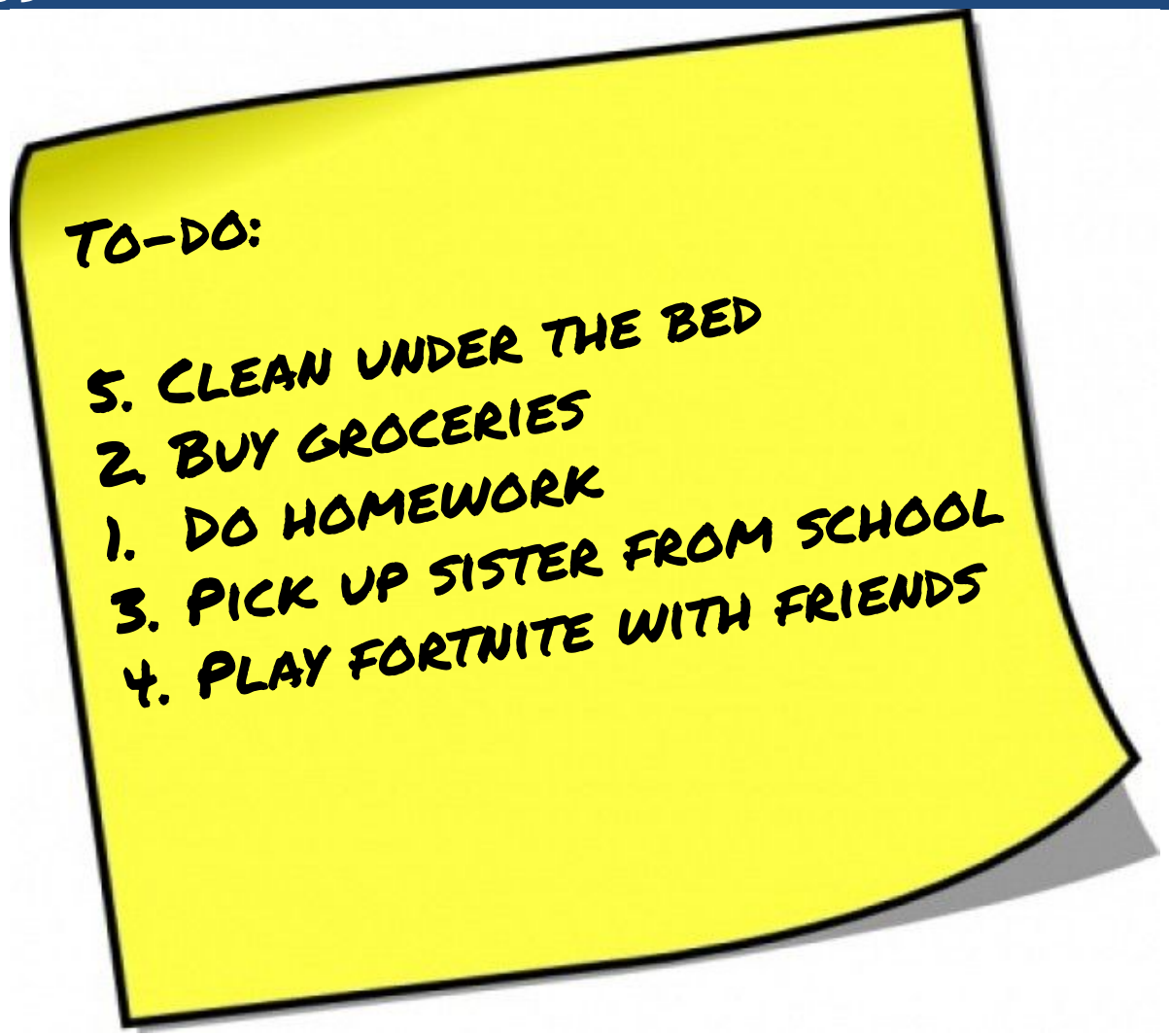
Mutual exclusion (mutex)

Analogy

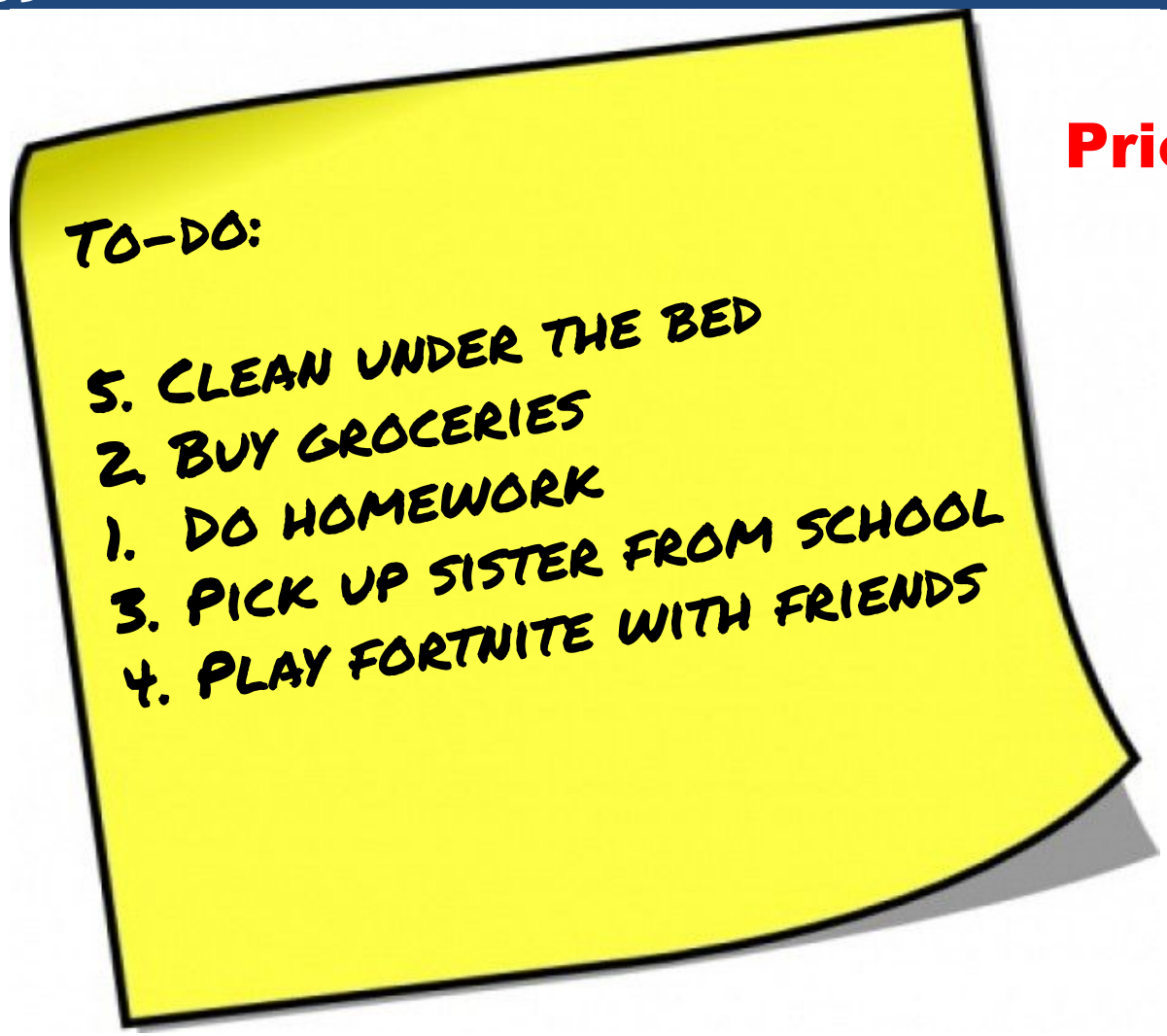
Analogy


TO-DO:
CLEAN UNDER THE BED
BUY GROCERIES
DO HOMEWORK
PICK UP SISTER FROM SCHOOL
PLAY FORTNITE WITH FRIENDS

Analogy

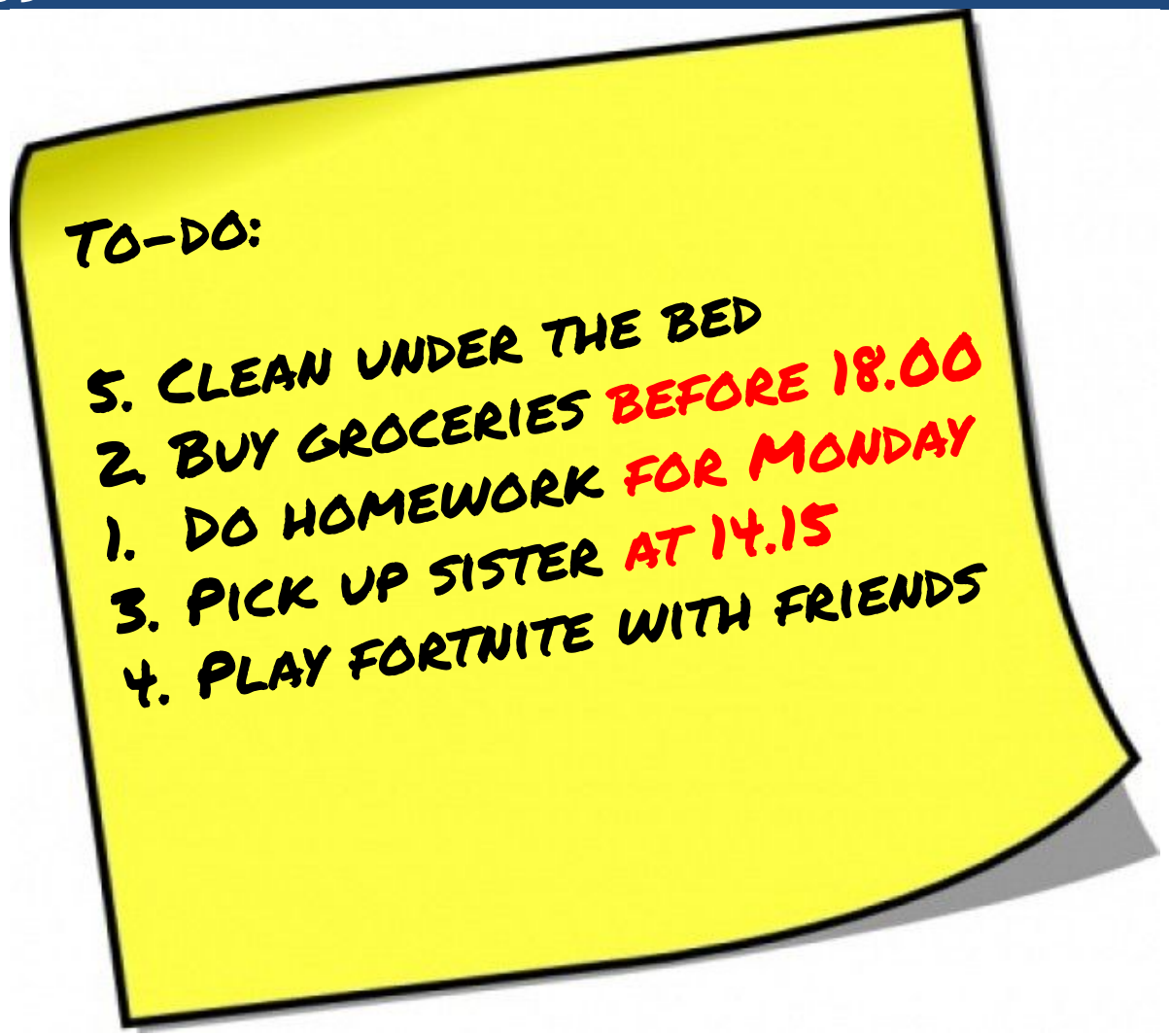

TO-DO:
5. CLEAN UNDER THE BED
2. BUY GROCERIES
1. DO HOMEWORK
3. PICK UP SISTER FROM SCHOOL
4. PLAY FORTNITE WITH FRIENDS

Analogy

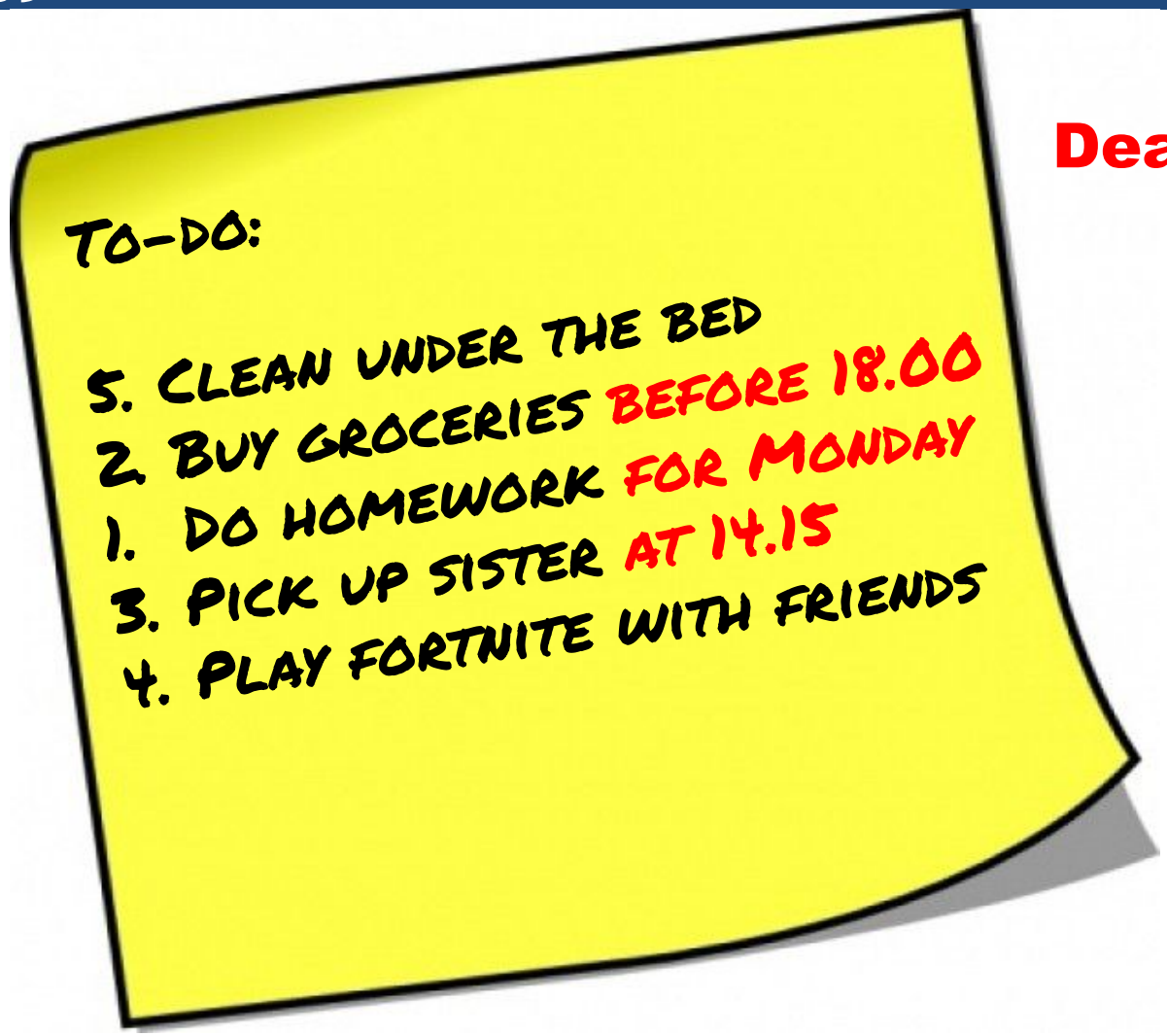


Priorities

Analogy


TO-DO:
5. CLEAN UNDER THE BED
2. BUY GROCERIES BEFORE 18.00
1. DO HOMEWORK FOR MONDAY
3. PICK UP SISTER AT 14.15
4. PLAY FORTNITE WITH FRIENDS

Analogy

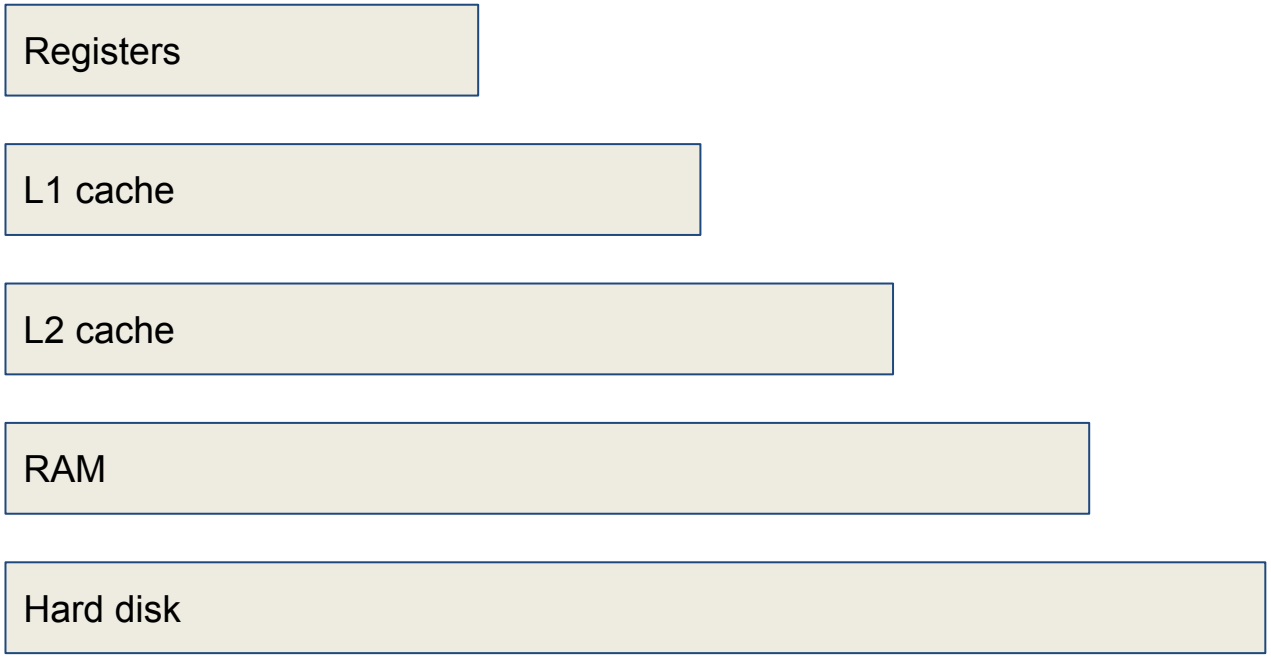


Deadlines

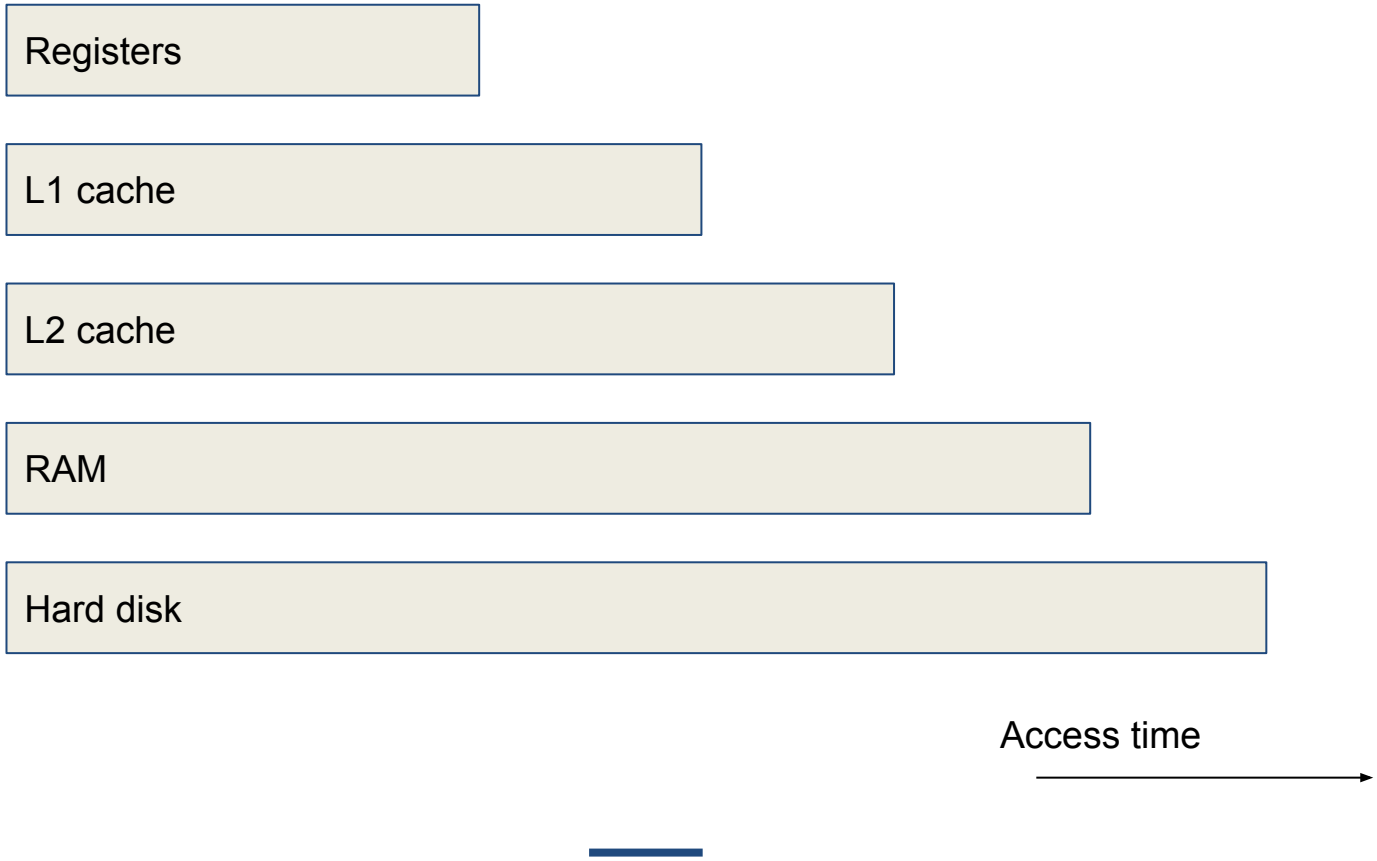
Strategy

Memory

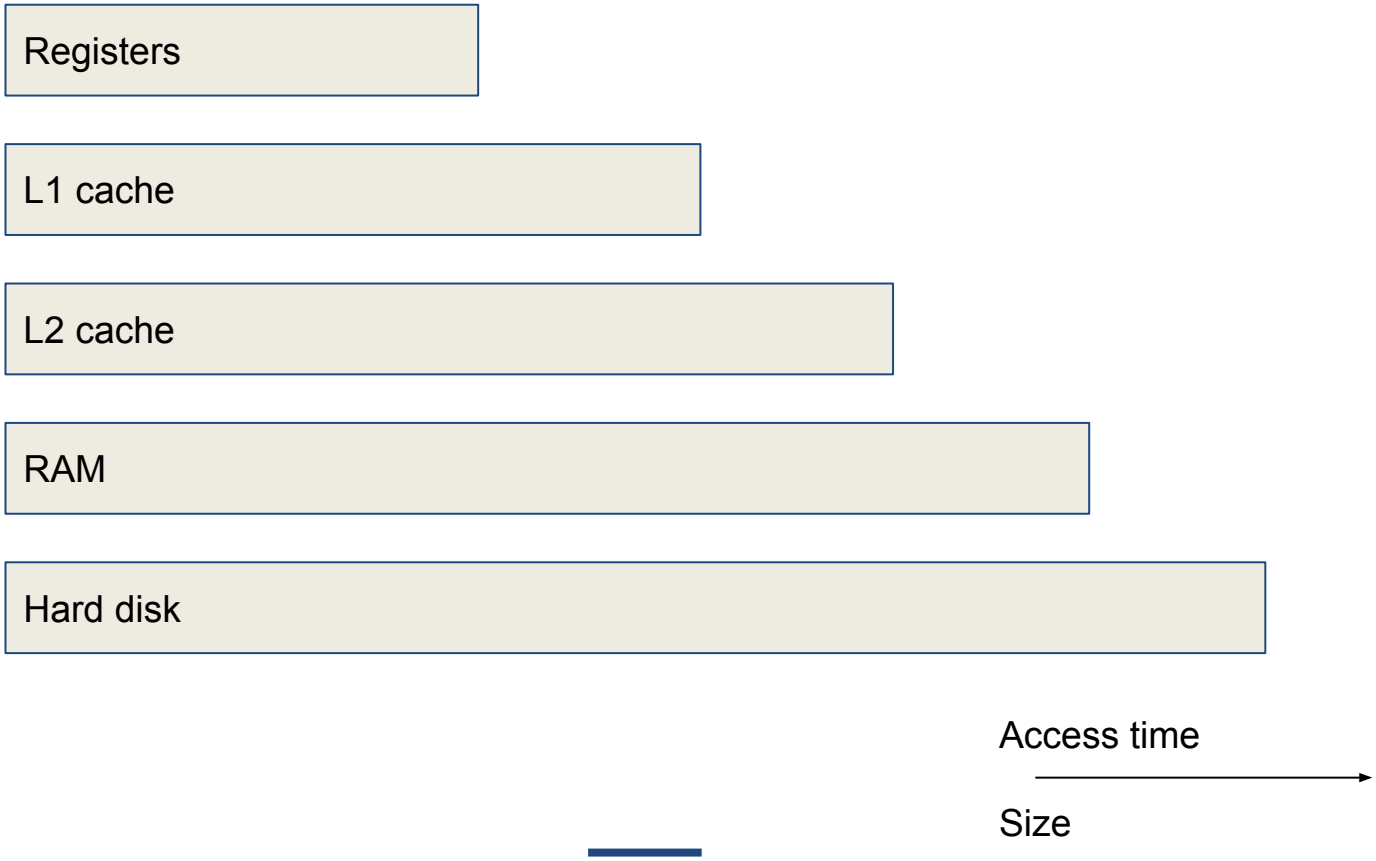
Memory



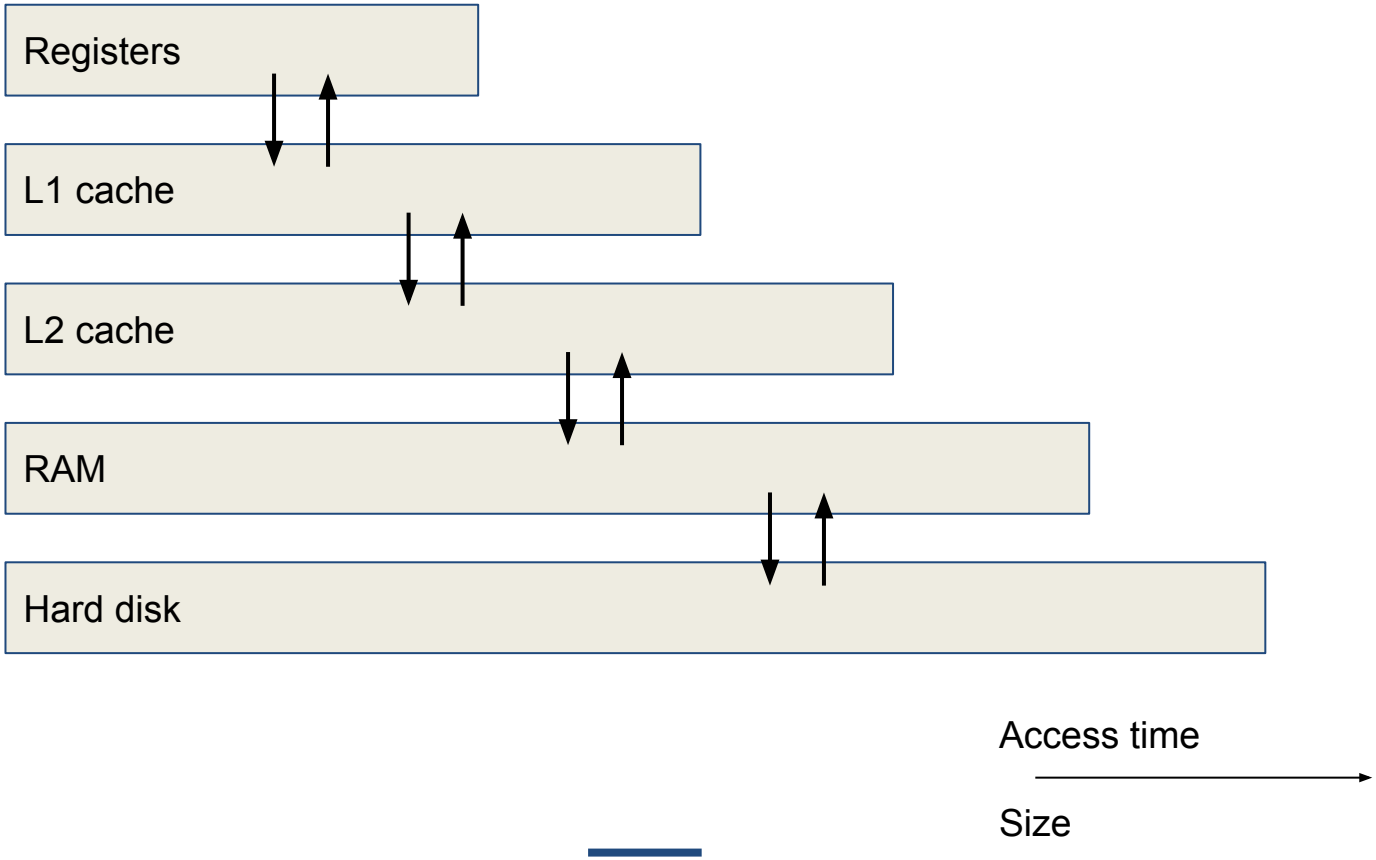
Memory



Memory



Memory



Strategy

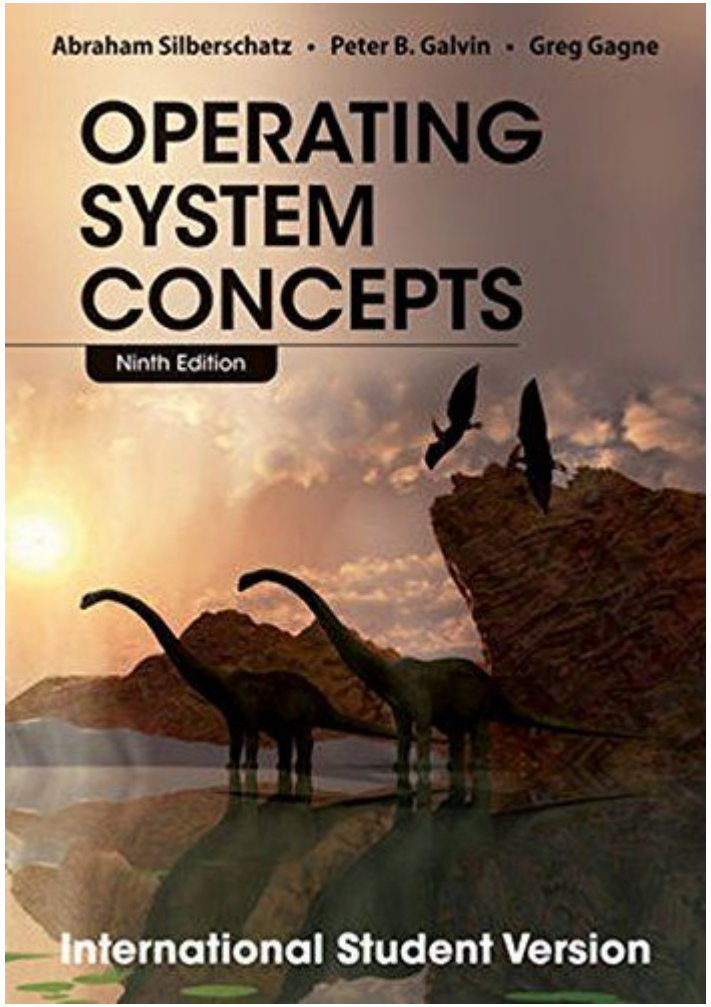
Storage

Summary

Primary responsibilities of the OS

- Resource management
  - Processing
  - Memory
  - Storage

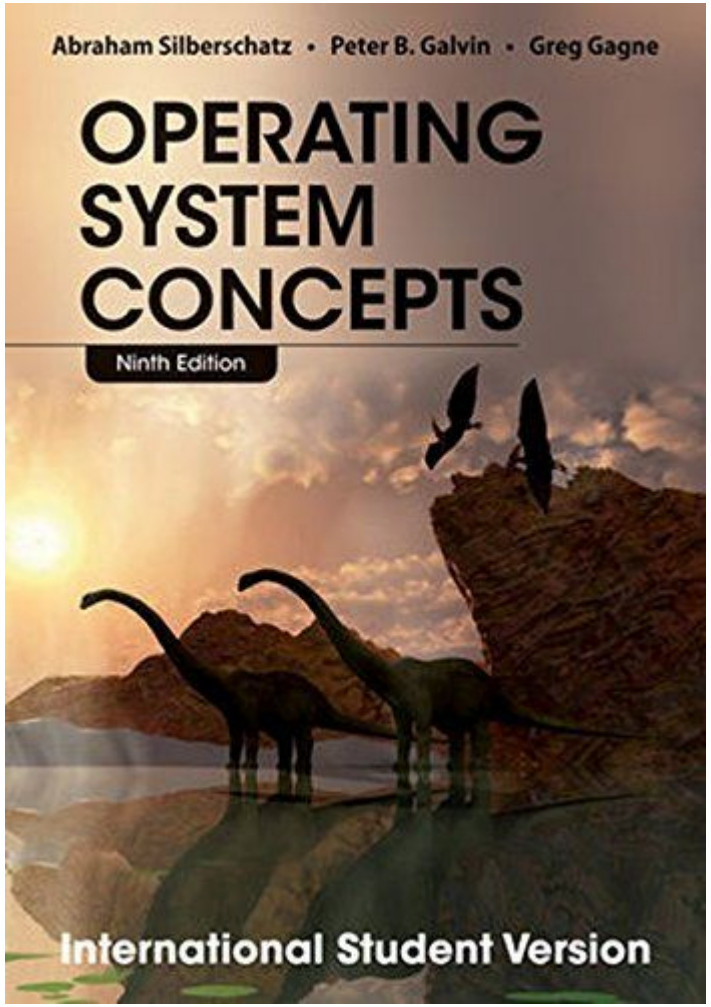
Homework



Study chapter 1 (43 pages)

in Silberschatz

Homework



Read chapter 1 (43 pages)

in Silberschatz

Have fun and see you next week!