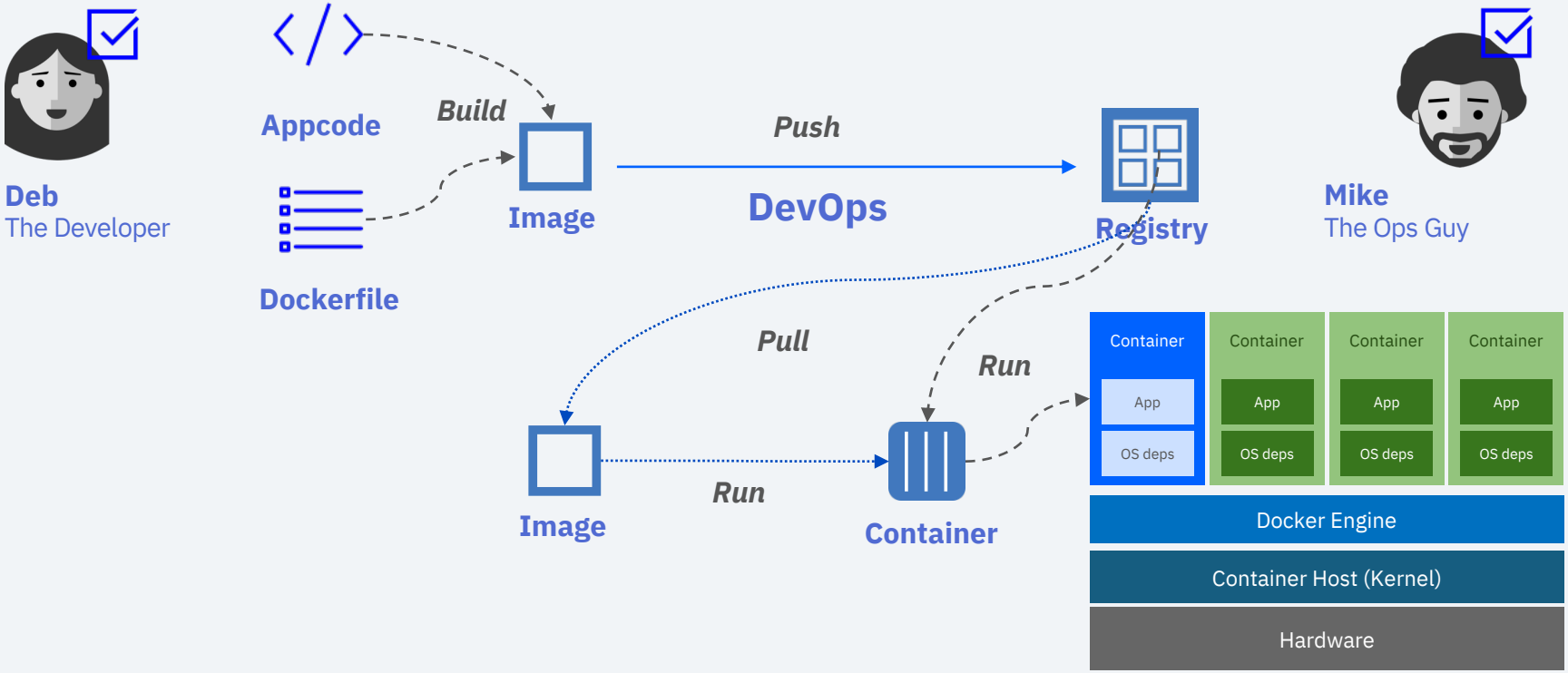
Kubernetes 101

Tal Neeman
Developer Advocate, IBM

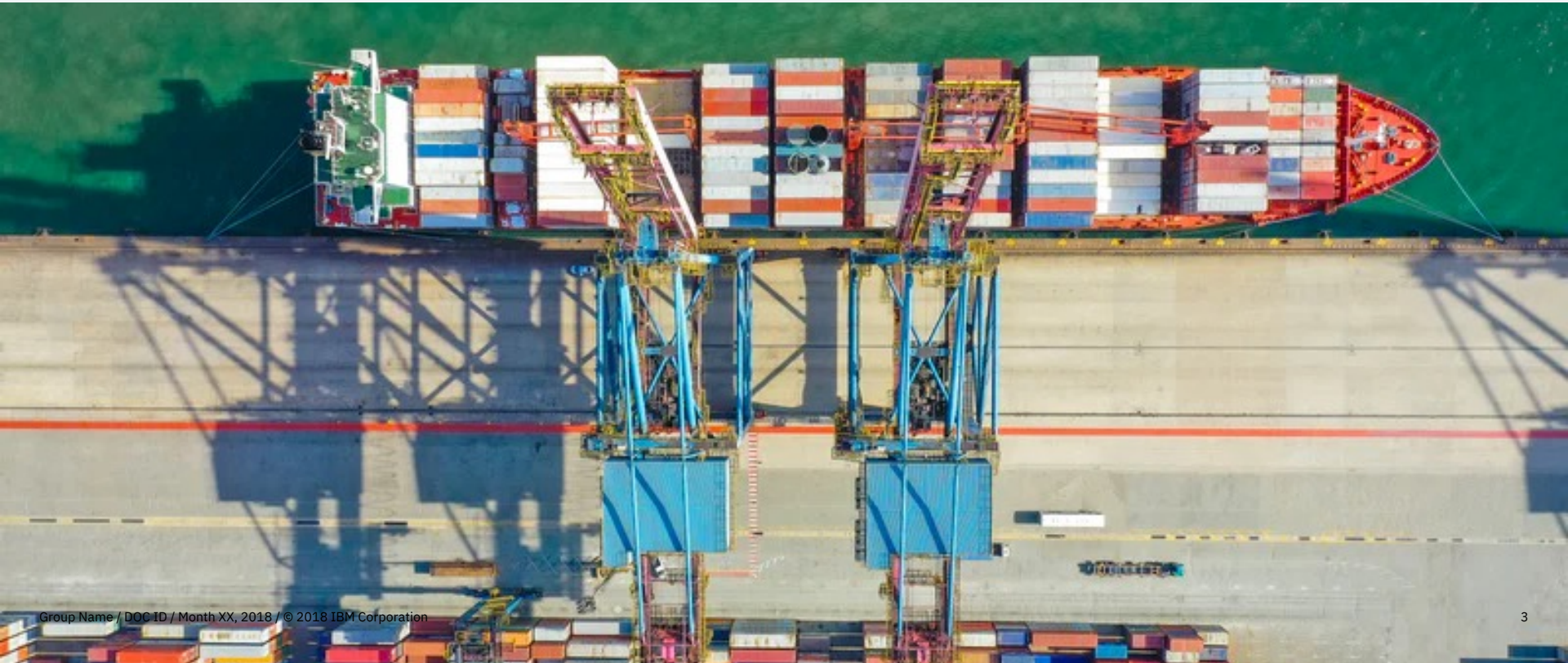
IBM **Developer**

IBM

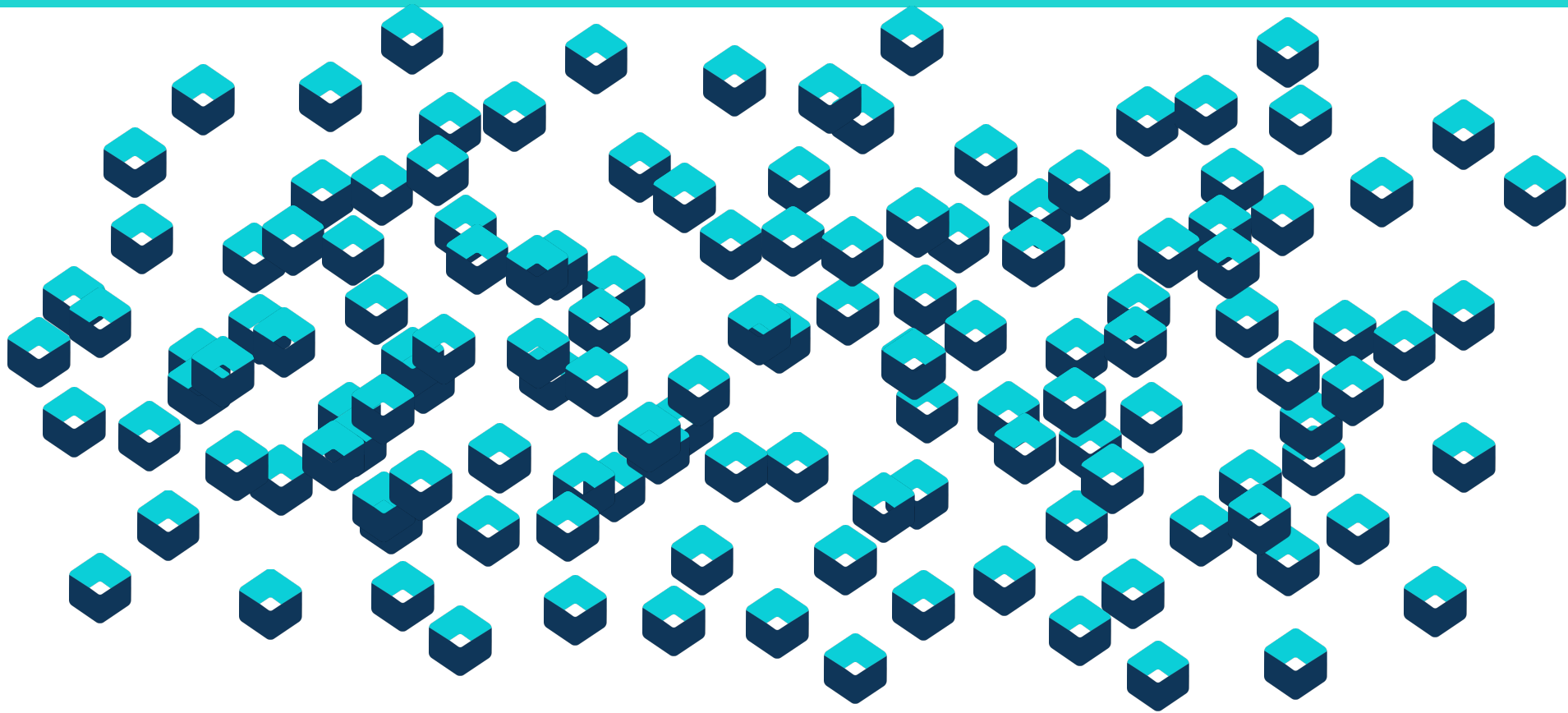
Basic deployment process



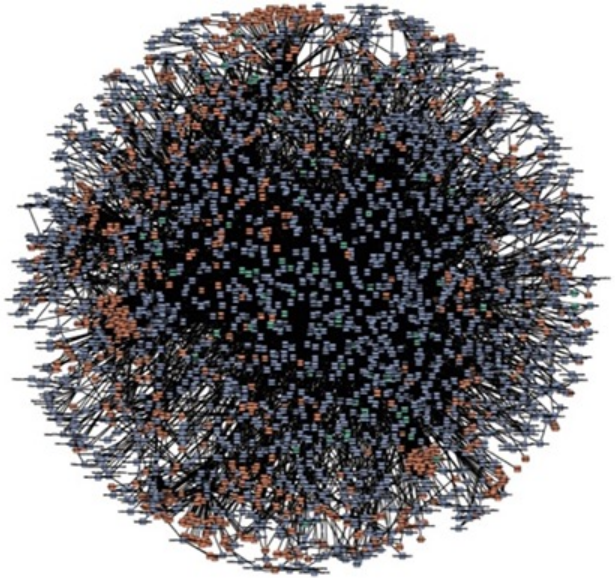
Orchestration



How to manage containers?



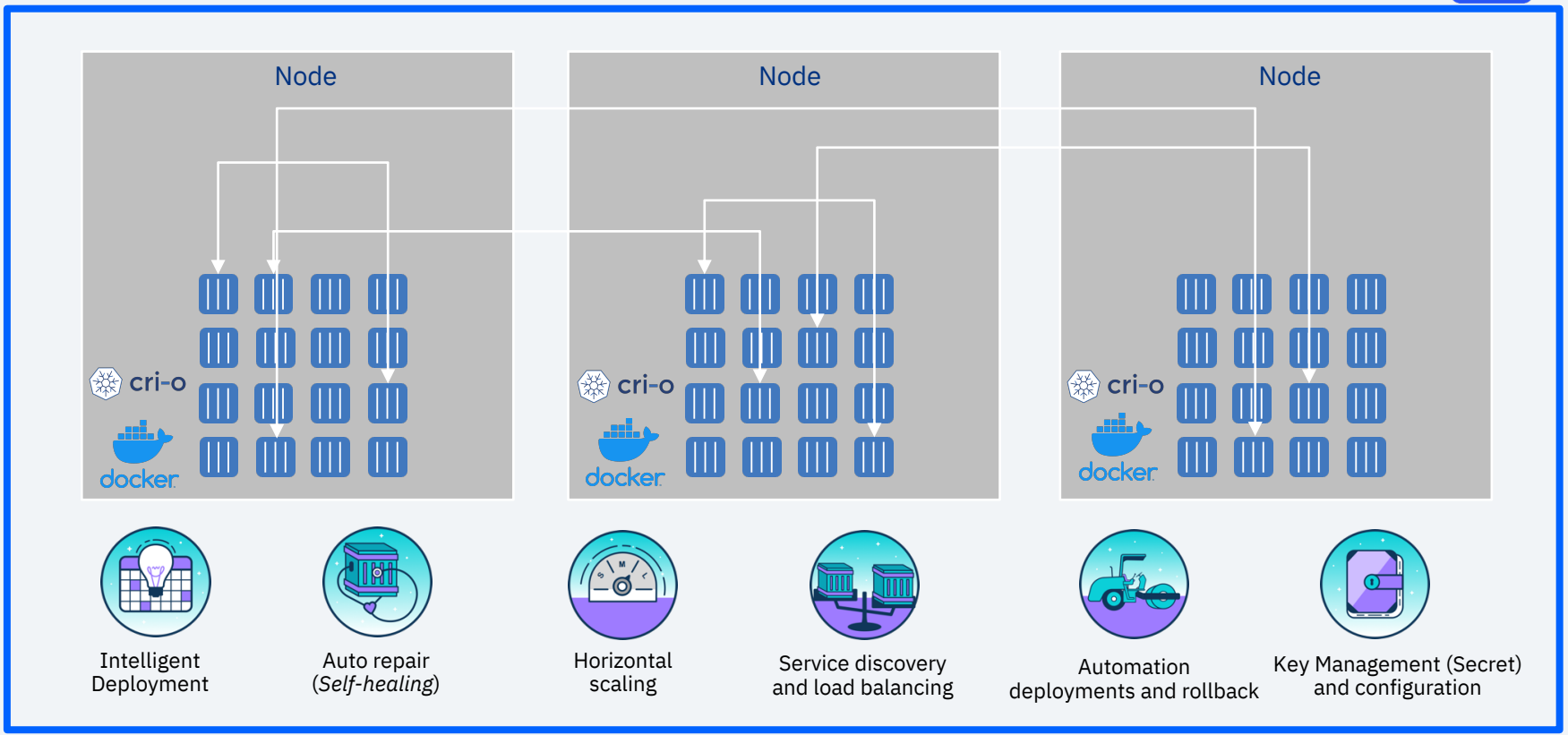
How to manage containers?

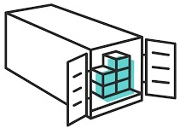


amazon.com

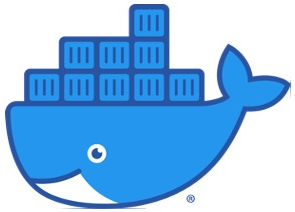
NETFLIX

Welcome to Kubernetes





Orchestration



Development

Production

Create multiple instances of containers: high availability and fault tolerance

Single host will become Single point of failure

Multi-host

Scaling/Replication

Automated scheduling of new containers: zero downtime deployments, instances/updates/recovery

Scheduling

Resources need to be managed with respect to the scale and the underlying infrastructure

cluster management

Check health status of your applications and actions to recover or maintain

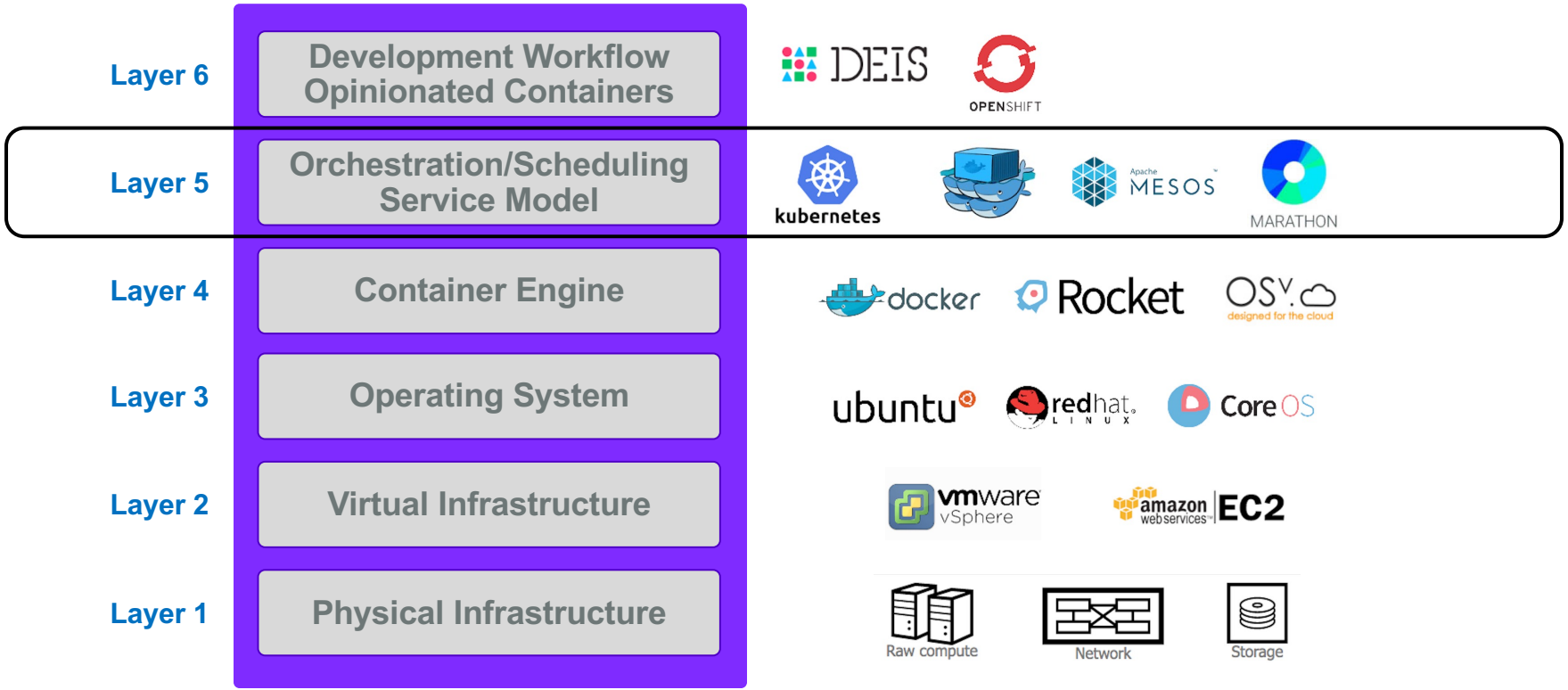
health management

@mohaghighi

What is Kubernetes?

- **Container Orchestrator**
  - Provision, manage, scale applications
- **Manage infrastructure resources needed by applications**
  - Volumes
  - Networks
  - Secrets
  - And many many many more...
- **Declarative model**
  - Provide the "desired state" and Kubernetes will make it happen
- **What's in a name?**
  - Kubernetes (K8s/Kube): "Helmsman" in ancient Greek

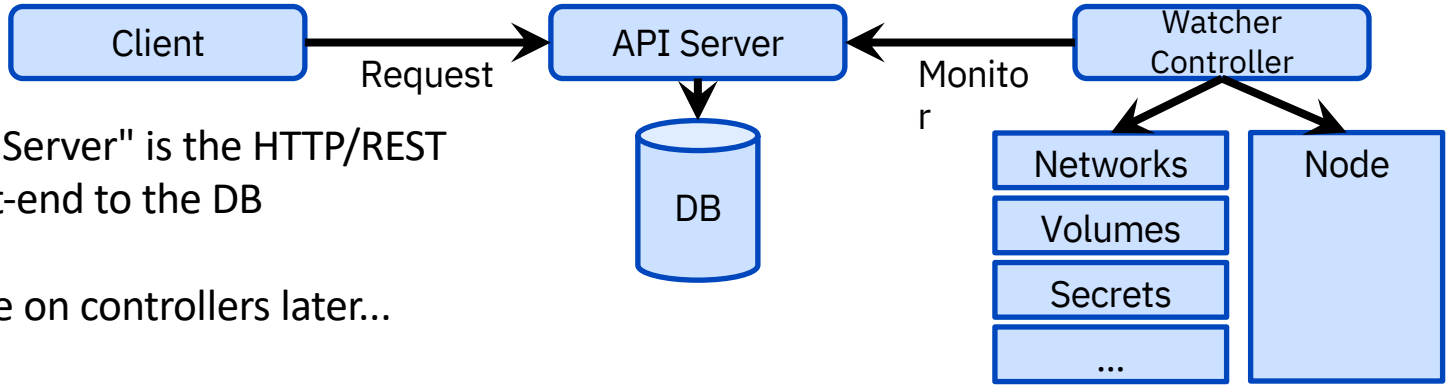
Container Ecosystem Layers



What is Kubernetes?

At its core, Kubernetes is a database (etcd).
With "watchers" & "controllers" that react to changes in the DB.
The controllers are what make it Kubernetes.
This pluggability and extensibility is part of its "secret sauce".

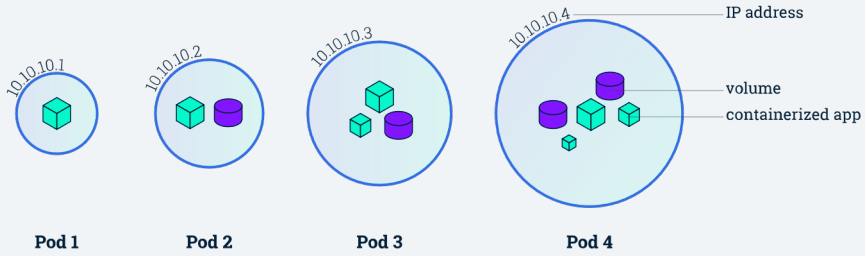
DB represents the user's desired state
Watchers attempt to make reality match the desired state

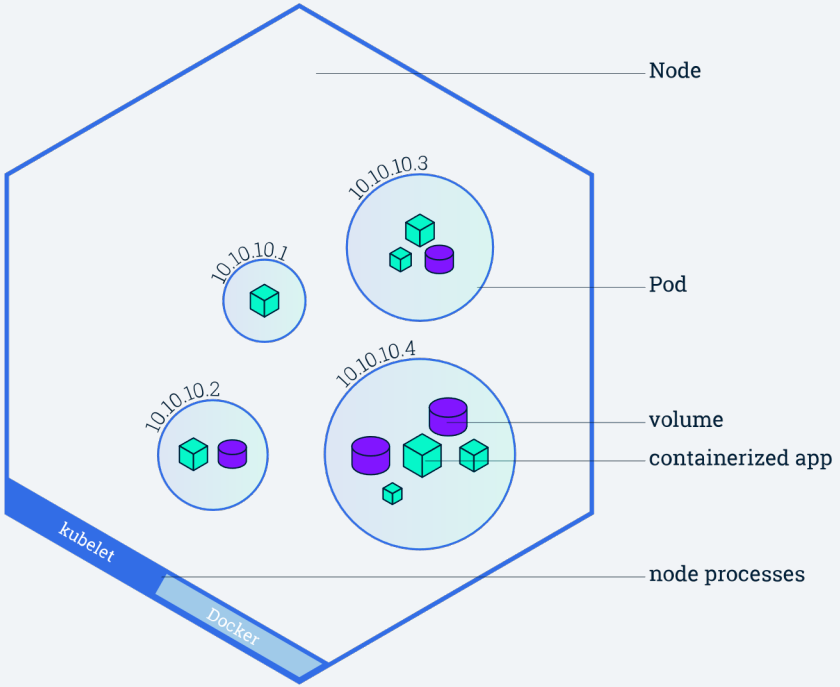


"API Server" is the HTTP/REST front-end to the DB

More on controllers later...

An application in Kubernetes





https://kubernetes.io/docs/tutorials/kubernetes-basics/explore/explore-intro/

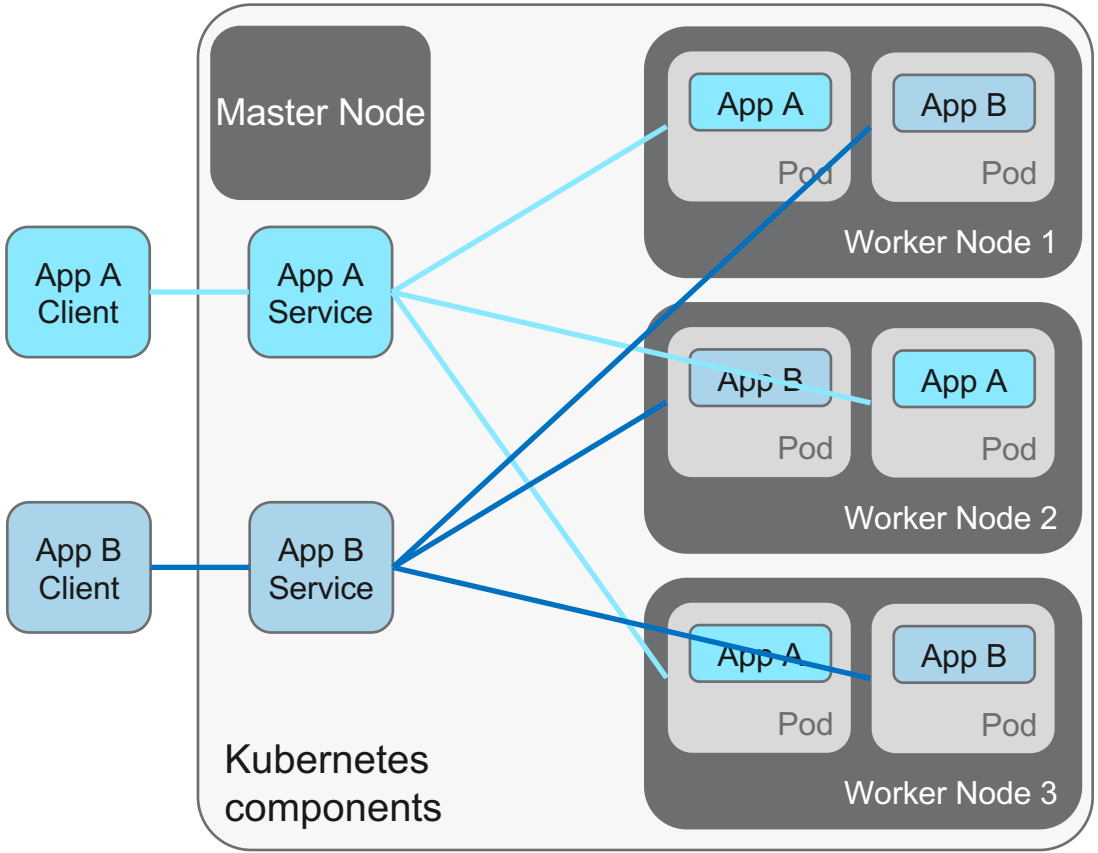
Kubernetes Architecture: How apps are accessed

Pod

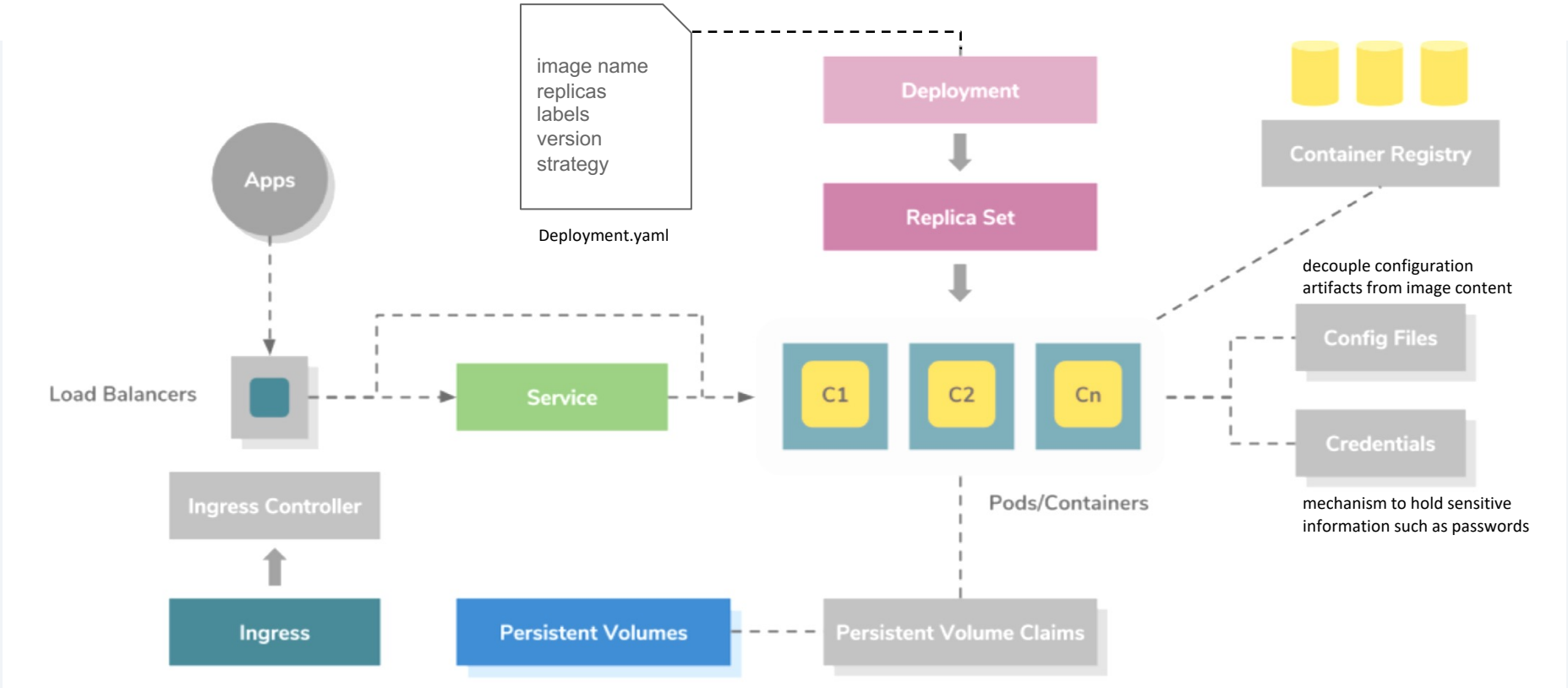
- Smallest deployment unit – runs containers
- Each pod has its own IP
- Shares a PID namespace, network, and hostname

Service

- Collection of pods exposed as an endpoint
  - state and networking info propagated to all worker nodes
- Types of service exposure
  - ClusterIP – Exposes cluster-internal IP
  - NodePort – Exposes the service on each Node's IP at a static port
  - LoadBalancer – Exposes externally using a cloud provider's load balancer
  - ExternalName – Maps to an external name (such as foo.bar.example.com)



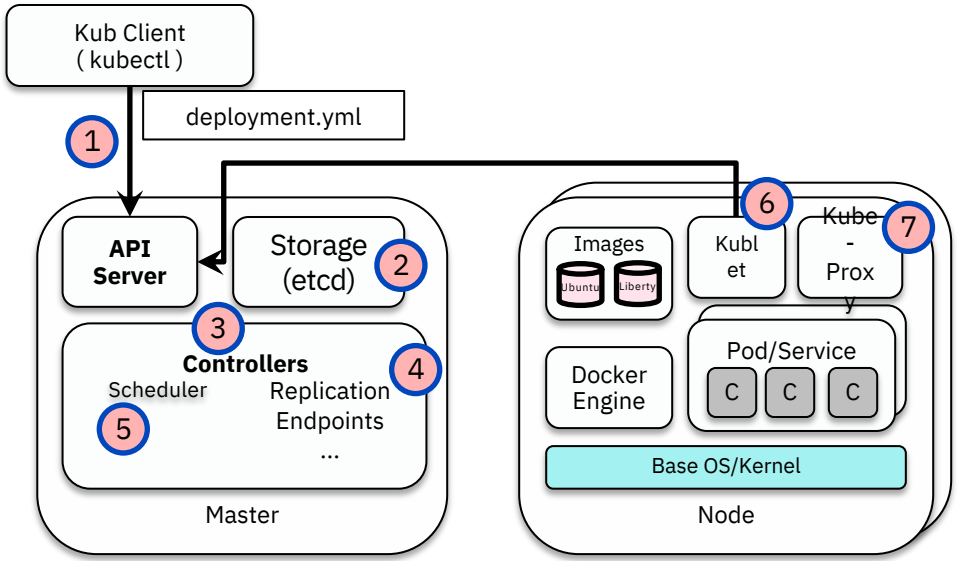
Components of Kubernetes



https://medium.com/containermind/a-beginners-guide-to-kubernetes-7e8ca56420b6

Kubernetes in Action!

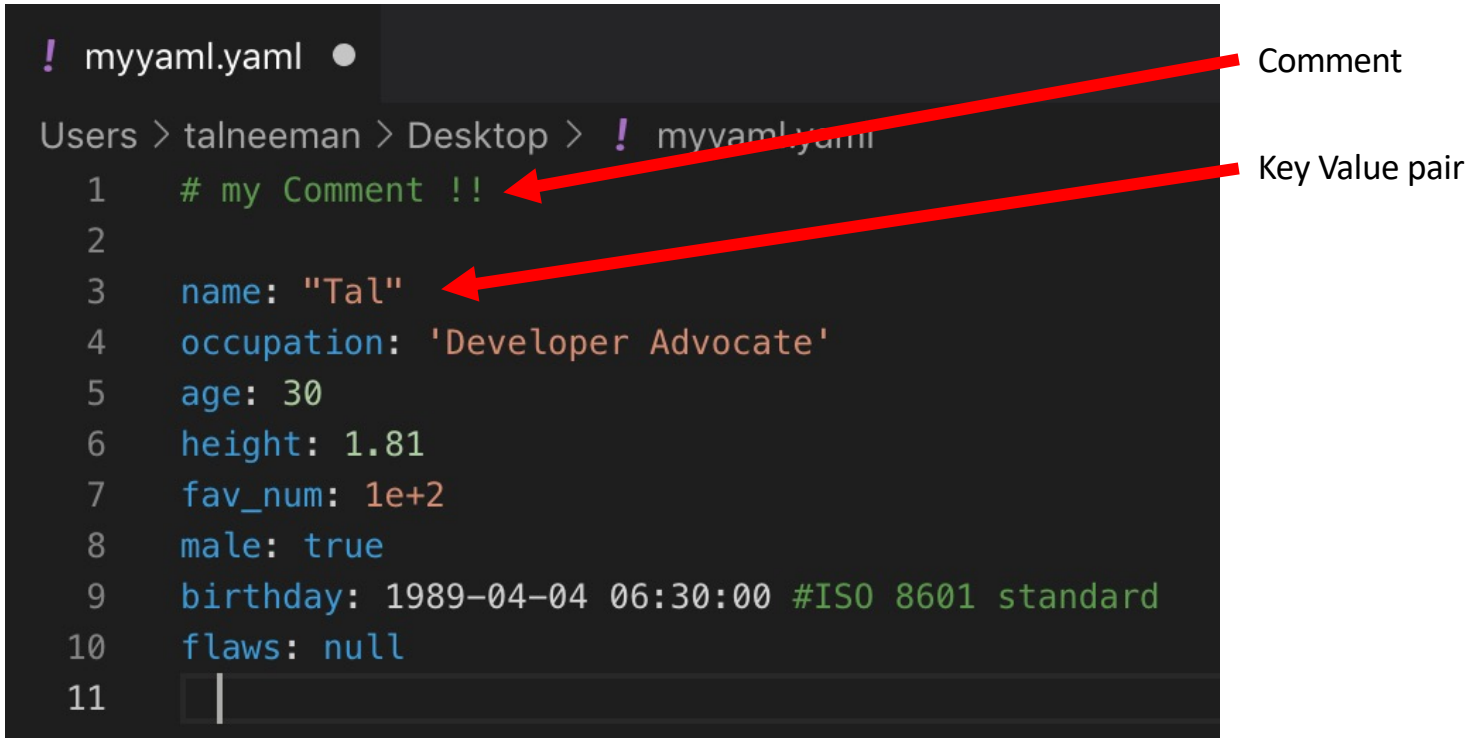
1. User via "kubectl" deploys a new application
2. API server receives the request and stores it in the DB (etcd)
3. Watchers/controllers detect the resource changes and act upon it
4. ReplicaSet watcher/controller detects the new app and creates new pods to match the desired # of instances
5. Scheduler assigns new pods to a kubelet
6. Kubelet detects pods and deploys them via the container runtime (e.g. Docker)
7. Kubeproxy manages network traffic for the pods – including service discovery and load-balancing

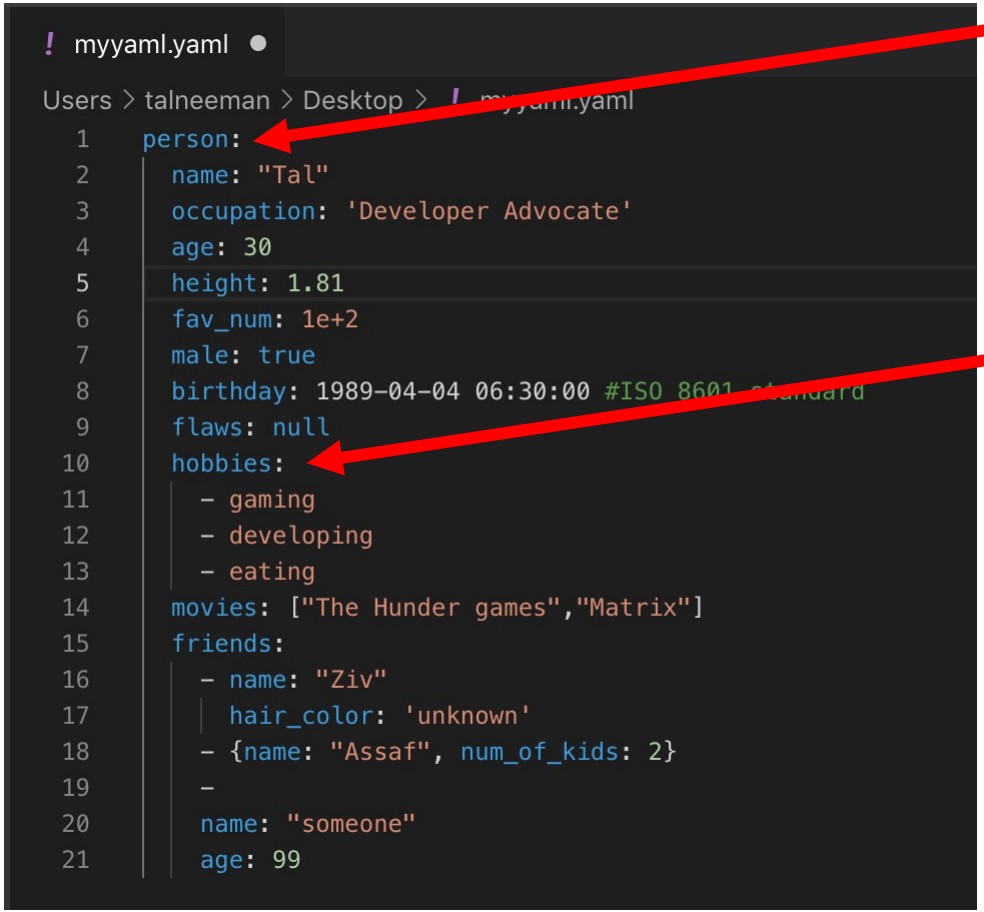


Yaml Syntax basics

XML	JSON	YAML

<Servers>
<Server>
<name>Server1</name>
<owner>John</owner>
<created>123456</created>
<status>active</status>
</Server>
</Servers>
	{
	Servers: [
	{
	name: Server1,
	owner: John,
	created: 123456,
	status: active
	}
	]
	}
		Servers:
		- name: Server1
		owner: John
		created: 123456
		status: active


myyaml.yaml
Users > talneeman > Desktop > myyaml.yaml
1 # my Comment !!
2
3 name: "Tal"
4 occupation: 'Developer Advocate'
5 age: 30
6 height: 1.81
7 fav_num: 1e+2
8 male: true
9 birthday: 1989-04-04 06:30:00 #ISO 8601 standard
10 flaws: null
11
Comment
Key Value pair



Dictionary / Map

Array / List

myyaml.yaml

Users > talneeman > Desktop > myyaml.yaml

```yaml
person:
  name: &name "Tal"
  occupation: 'Developer Advocate'
  age: 30
  height: 1.81
  fav_num: 1e+2
  male: true
  birthday: 1989-04-04 06:30:00 #ISO 8601 standard
  flaws: null
  hobbies:
    - gaming
    - developing
    - eating
  movies: ["The Hunder games","Matrix"]
  friends:
    - name: "Ziv"
      hair_color: 'unknown'
    - {name: "Assaf", num_of_kids: 2}
    -
     name: "someone"
     age: 99
  description: >
    a long description that you want to be readable
    so you split them into many lines
    but you prefer to get them as a 1 line at the end
    so you use > sign
  signature: |
    Tal
    I Want to save this format
    email - talne@il.ibm.com
  id: *name #"Tal"
```

Copy the value here to *name

Convert long text into single line string

Convert long text into single string
and keep the format

myyaml.yaml

Users > talneeman > Desktop > myyaml.yaml

```yaml
# my Comment !!

person:
  name: &name "Tal"
  occupation: 'Developer Advocate'
  age: !!float 30 #30.0
  height: !!str 1.81 #"1.181"
  fav_num: 1e+2
  male: true
  birthday: 1989-04-04 06:30:00 #ISO 8601 standard
  flaws: null
  hobbies:
    - gaming
    - developing
    - eating
  movies: ["The Hunder games","Matrix"]
  friends:
    - name: "Ziv"
      hair_color: 'unknown'
    - {name: "Assaf", num_of_kids: 2}
    -
    name: "someone"
    age: 99
  description: >
    a long description that you want to be readable
  so you split them into many lines
  but you prefer to get them as a 1 line at the end
  so you use > sign
  signature: |
    Tal
  I Want to save this format
  email - talne@il.ibm.com
  id: *name #"Tal"

---

base: &base
  var1: value1

foo:
  <<: *base #var1: value1
  var2: value2
```