Quality Attributes

Docker & Kubernetes

Related work

Since 2017-2018

The Present

- NSWI150 - Virtualizace a cloud computing
- NSWI177 - Úvod do Linuxu
- NSWI153 - Pokročilé programování webových aplikací
- …

Quality Attributes

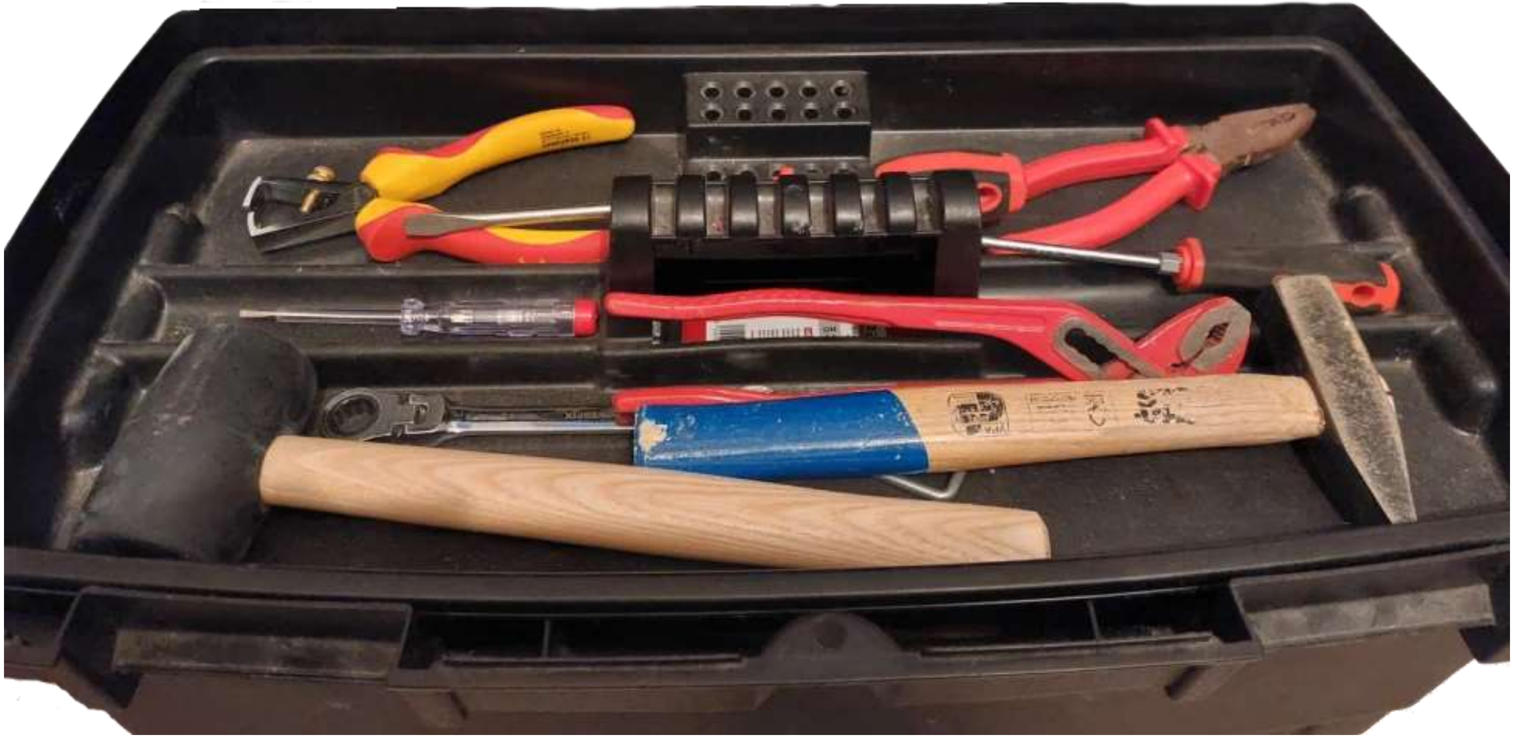
Run-Time

- Availability (ANSI/TIA-942, …)
- Performance
- Security
- …

Non-Run-Time

- Modifiability
- Portability
- Testability
- Reusability
- Integrability
- …

Toolbox – what is missing?



Managing expectations

I want to learn Docker / Kubernetes ...

- print(“Hello, World!”)
- C++ in 21 days

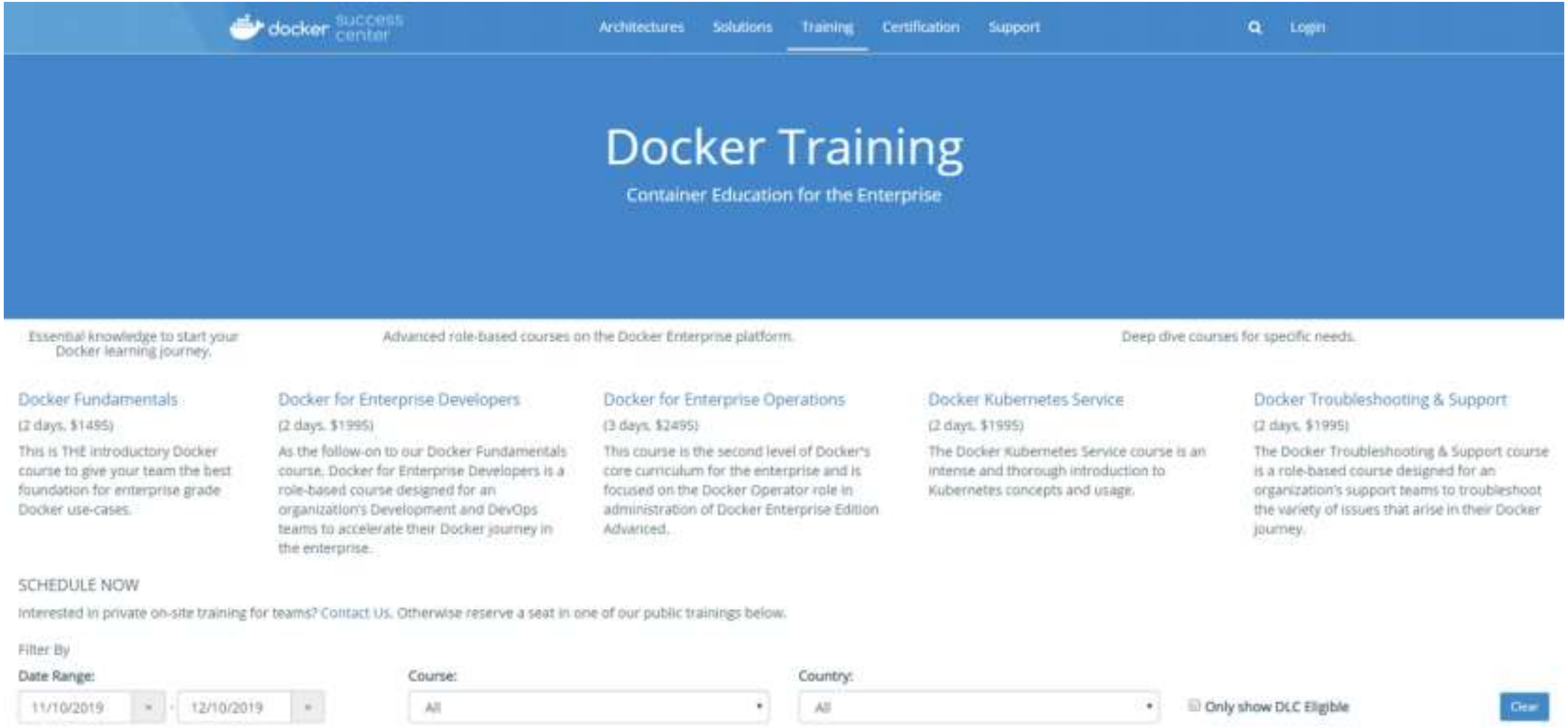
Software architect

A software architect is a software developer expert who makes high-level design choices and tries to enforce technical standards, including software coding standards, tools, and platforms.

Source https://en.wikipedia.org/wiki/Software_architect

Docker

Learning Docker …



Docker Training

Container Education for the Enterprise

Essential knowledge to start your Docker learning journey.

Advanced role-based courses on the Docker Enterprise platform.

Deep dive courses for specific needs.

Docker Fundamentals
(2 days, $1495)
This is THE introductory Docker course to give your team the best foundation for enterprise grade Docker use-cases.

Docker for Enterprise Developers
(2 days, $1995)
As the follow-on to our Docker Fundamentals course, Docker for Enterprise Developers is a role-based course designed for an organization's Development and DevOps teams to accelerate their Docker journey in the enterprise.

Docker for Enterprise Operations
(3 days, $2495)
This course is the second level of Docker's core curriculum for the enterprise and is focused on the Docker Operator role in administration of Docker Enterprise Edition Advanced.

Docker Kubernetes Service
(2 days, $1995)
The Docker Kubernetes Service course is an intense and thorough introduction to Kubernetes concepts and usage.

Docker Troubleshooting & Support
(2 days, $1995)
The Docker Troubleshooting & Support course is a role-based course designed for an organization's support teams to troubleshoot the variety of issues that arise in their Docker journey.

SCHEDULE NOW

Interested in private on-site training for teams? Contact Us. Otherwise reserve a seat in one of our public trainings below.

Filter By

Date Range: 11/10/2019 - 12/10/2019

Course: All

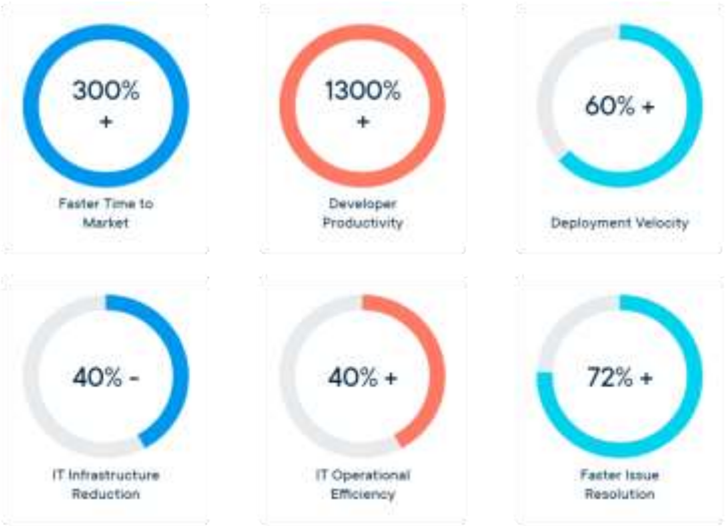
Country: All

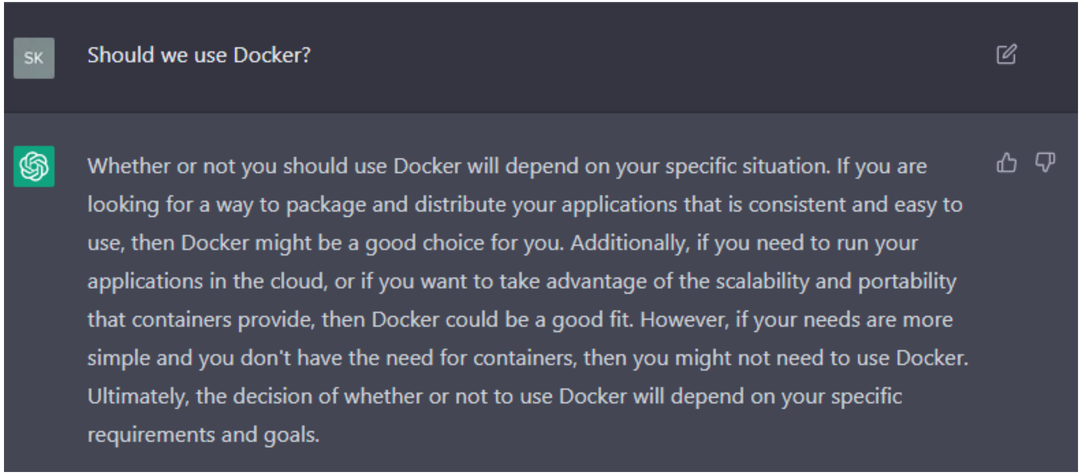
□ Only show DLC Eligible

Why Docker ?

In **2013**, Docker introduced what would become the industry standard for containers. Containers are a standardized unit of software that allows developers to isolate their app from its environment, solving the “it works on my machine” headache.

For millions of developers today, Docker is the de facto standard to build and share containerized apps – from desktop, to the cloud.





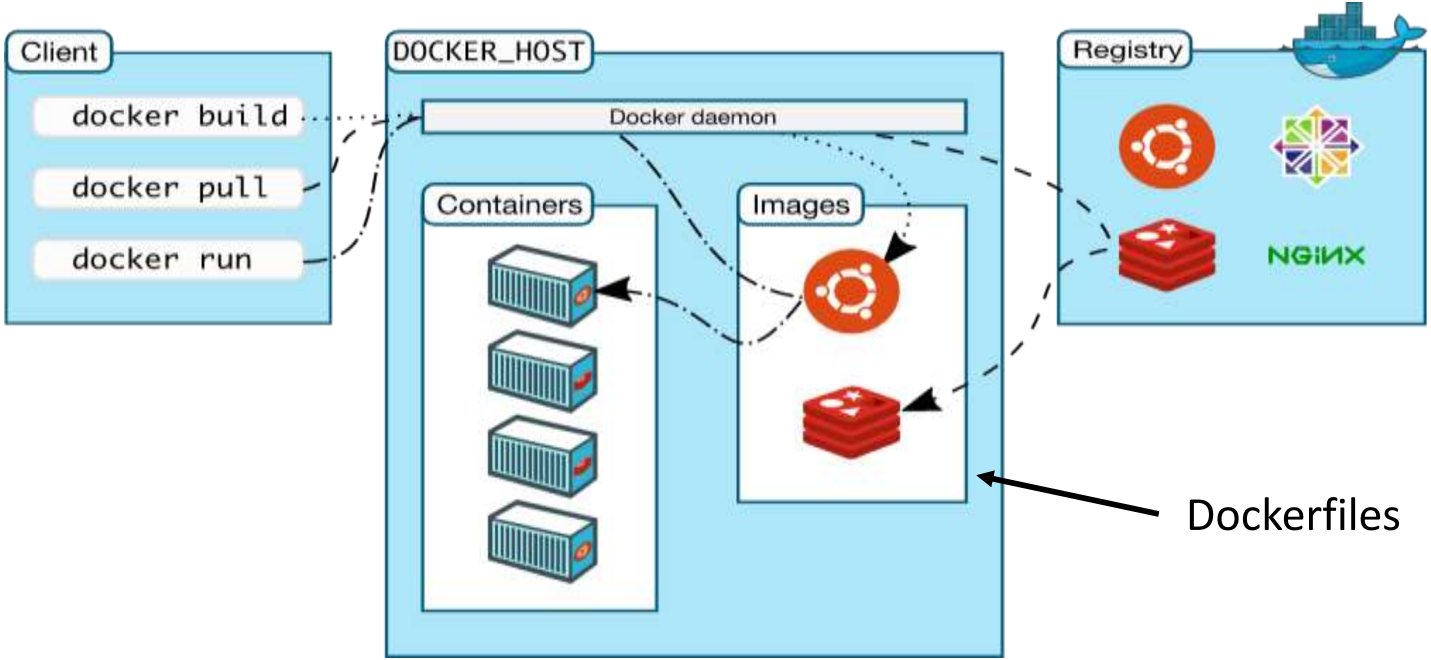
Should we use Docker?

Whether or not you should use Docker will depend on your specific situation. If you are looking for a way to package and distribute your applications that is consistent and easy to use, then Docker might be a good choice for you. Additionally, if you need to run your applications in the cloud, or if you want to take advantage of the scalability and portability that containers provide, then Docker could be a good fit. However, if your needs are more simple and you don't have the need for containers, then you might not need to use Docker. Ultimately, the decision of whether or not to use Docker will depend on your specific requirements and goals.

Choose a plan that is right for you …

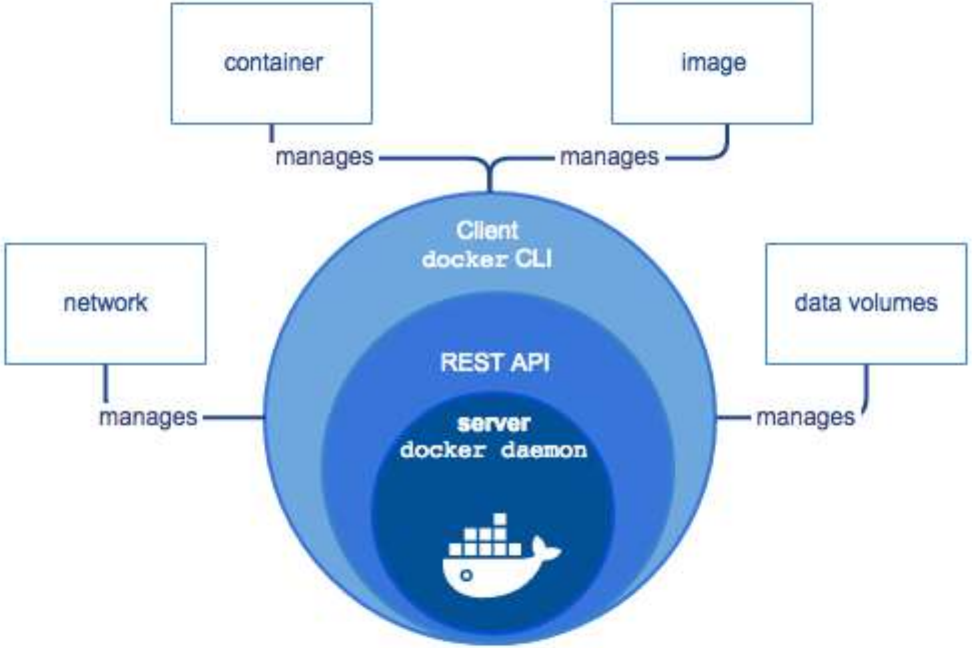
What is Docker?

Containerization is a lightweight virtualization technology alternative to hypervisor virtualization.



Dockerfiles

What is Docker?



Docker implementation

Linux namespaces provide isolated environments:

- pid namespace - process isolation
- net namespace - network interface
- ip namespace - inter-process communication
- mnt namespace - filesystem mounting points
- uts namespace - isolation of kernel and version identifiers

Control groups (cgroups) to limit resources.

Union file system (UnionFS) variants to manage file system.

….

Demo : Docker

- Docker install
- Docker commands
- Hello world
- Dockerfile
- Node.js application

There is more …

- Docker Captain Program
- Docker Volume
- ADD / COPY / ~~MAINTAINER~~ / ENTRYPOINT/…
- Multi-stage builds
- Docker Network
- API (REST, /var/run/docker.sock)
- cAdvisor
- Podman
- ….

Quality Attributes with Docker

Run-Time

- Availability
- Performance
- Security
- …

Non-Run-Time

- Modifiability
- Portability
- Testability
- Reusability
- Integrability
- …

What if …

```
FROM maven:3-jdk-10-slim
RUN apt-get update; apt-get upgrade –y
RUN apt-get install -y curl software-properties-common gnupg; \
  curl -sL https://deb.nodesource.com/setup_10.x | bash -
RUN apt-get install -y nodejs
RUN node -v; npm –v

WORKDIR /etl
COPY . .
RUN mvn install –DskipTests
RUN chmod a+x start.sh


CMD ["/bin/bash", "/etl/start.sh"]
```

```
exec ./executor.sh & exec ./executor-monitor.sh & exec ./storage.sh & exec ./frontend.sh
```

https://github.com/linkedpipes/etl/pull/627

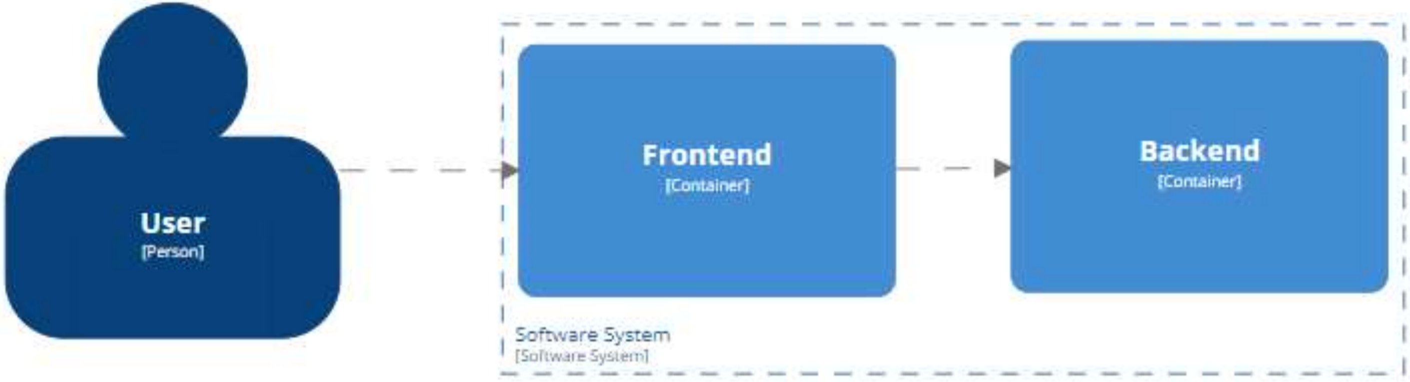
Docker Compose

Docker Compose

- Docker Compose vs. Docker-Compose
- Declarative
- Definition in *docker-compose.yml*
- Multiple isolated environments on a single host
- Service discovery

Demo : Docker Compose

- Two services Frontend / Backend
- Scale Frontend
- Reverse proxy
- Showcase Prankweb



Meme Time

Quality Attributes with Docker Compose

Run-Time

- Availability
- Performance
- Security
- …

Non-Run-Time

- Modifiability
- Portability
- Testability
- Reusability
- Integrability
- …

There is more …

- Multiple networks
- Service dependencies
- Development vs Production docker files
- Docker Compose File
- ….

What is Container Orchestration?

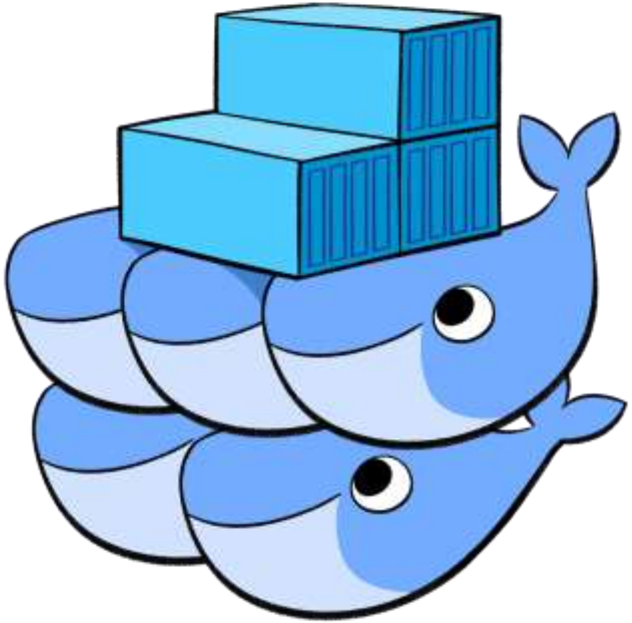
Container orchestration is the automation of much of the operational effort required to run containerized workloads and services.

This includes a wide range of things software teams need to manage a container's lifecycle, including provisioning, deployment, scaling (up and down), networking,
load balancing and more.

… a containerized application might translate into operating hundreds or thousands of containers, especially when building and operating any large-scale system.

Source: https://www.vmware.com/topics/glossary/content/container-orchestration.html

Docker Swarm



Docker Swarm is Dead. Long Live Kubernetes!

11 October, 2017 | by Kendall Miller

https://www.fairwinds.com/blog/docker-swarm-is-dead-long-live-kubernetes

Mirantis is committed to providing an excellent experience to all Docker Enterprise platform customers and currently expects to support Swarm for at least two years, depending on customer input into the roadmap. 11.2019

https://www.mirantis.com/blog/mirantis-acquires-docker-enterprise-platform-business/

The firm wanted to “to simplify container usage at enterprise scale with freedom of choice for orchestrators…and our customers can rest assured that Mirantis will continue to support their Swarm investments.” 02.2020

https://devclass.com/2020/02/25/mirantis-to-keep-docker-swarm-buzzing-around-pledges-new-features/

Classic Swarm has been archived and is no longer actively developed. You may want to use the Swarm mode built into the Docker Engine instead, or another orchestration system. 10.2022

https://github.com/docker-archive/classicswarmt

Docker Swarm

A swarm is a group of (physical/virtual) machines (nodes) that are running Docker and joined into a cluster.

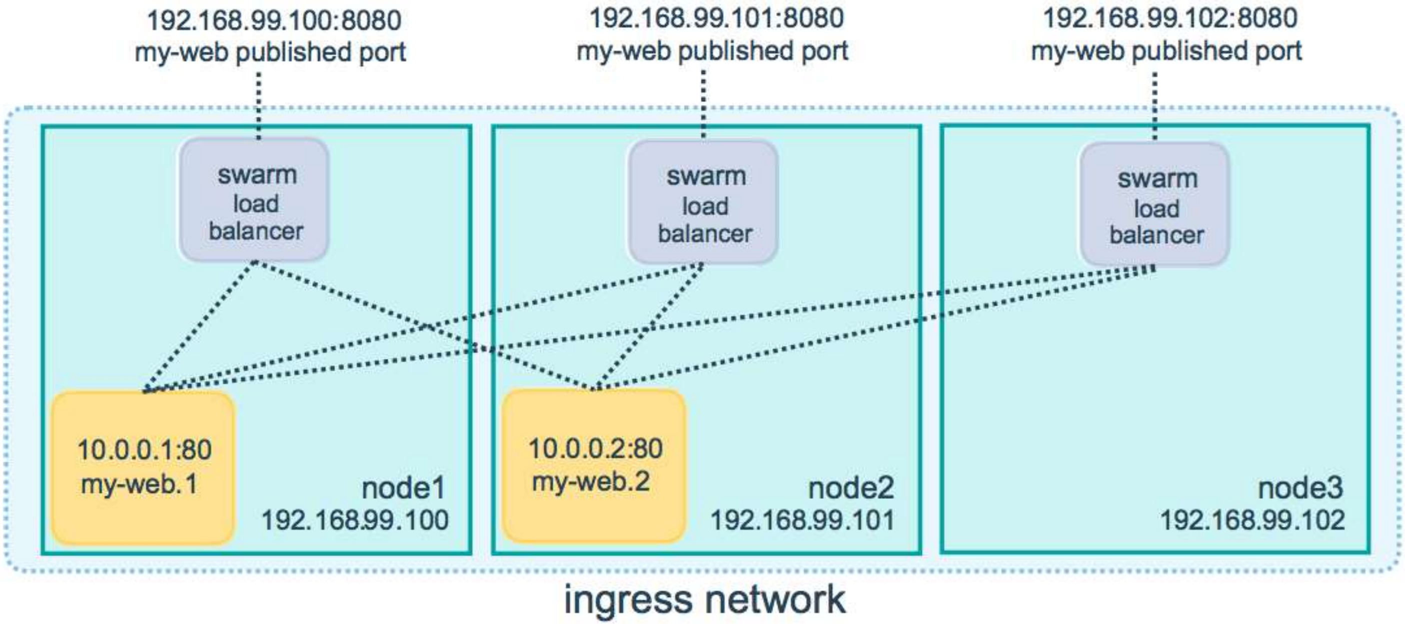
Orchestrate Docker containers which house stateless applications

- Easy to deploy
- Declarative service model / Desired state reconciliation
- Service discovery
- Secure by default
- Rolling updates
- Docker can be 'switched' into swarm mode
- …

Source: https://docs.docker.com/engine/swarm/

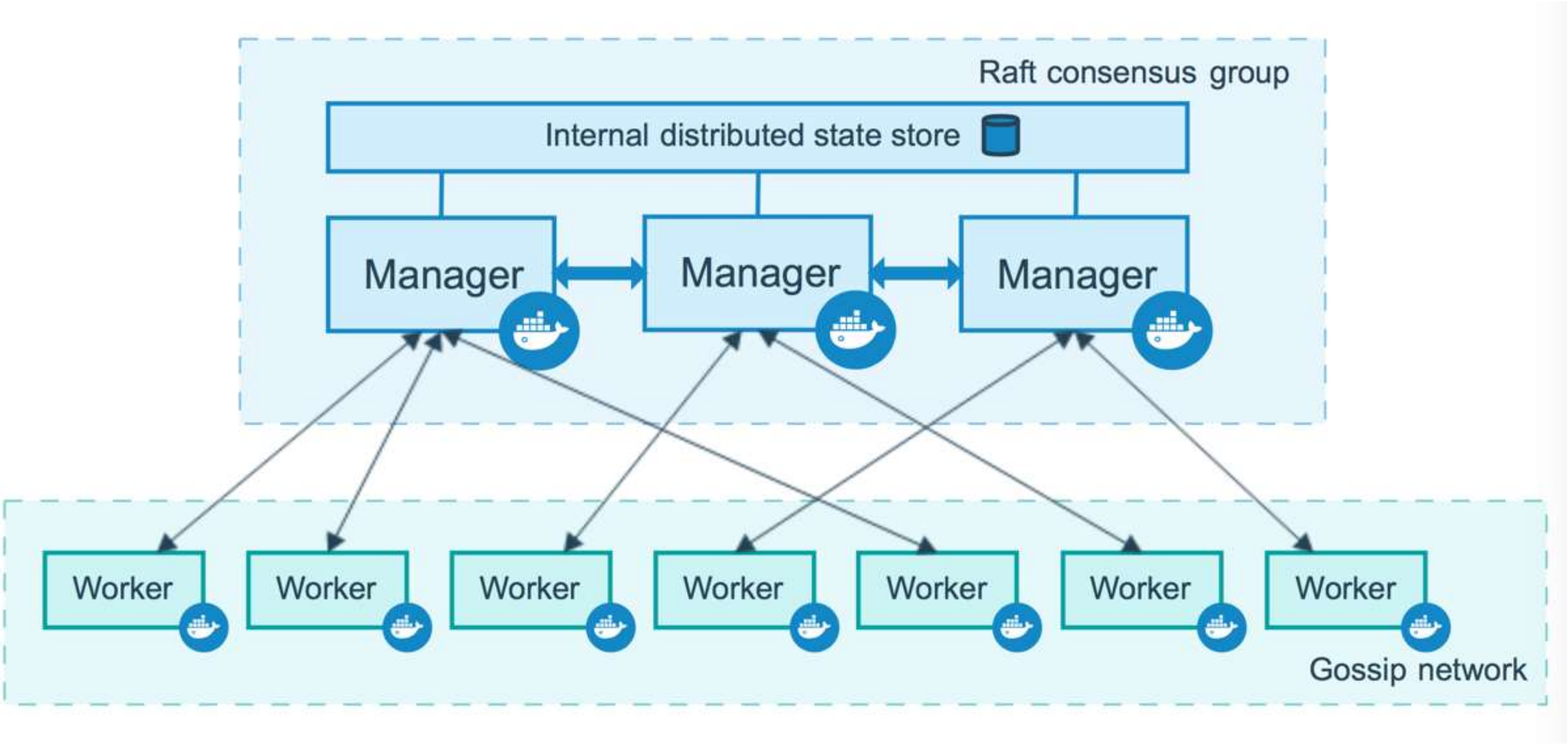
Docker Swarm

You can access the service by hitting any of the manager or worker nodes. It does not matter if the particular node does not have a container scheduled on it. That is the whole idea of the swarm.



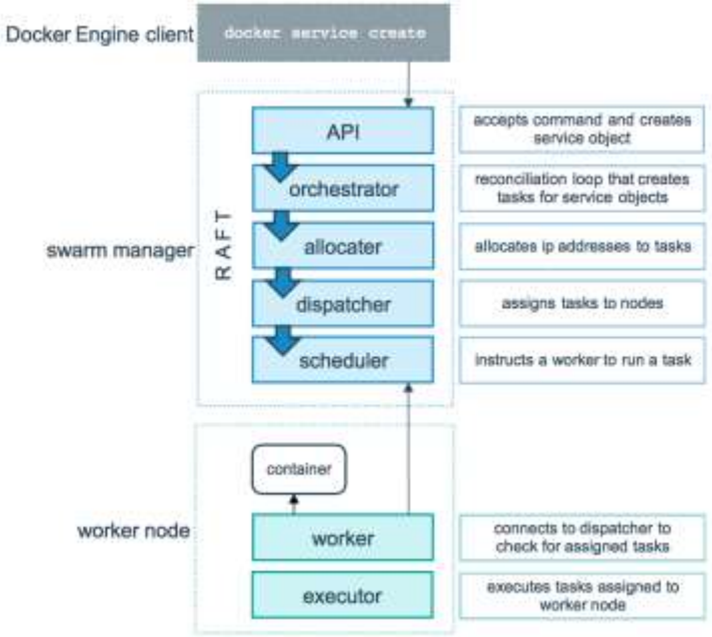
Source: https://docs.docker.com/engine/swarm/ingress/

Docker Swarm Infrasctructure



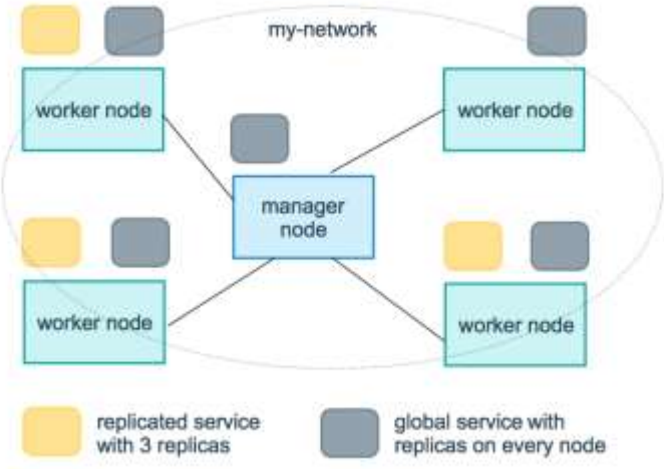
Source: https://docs.docker.com/engine/swarm/how-swarm-mode-works/nodes/

Docker Swarm Task



Docker Engine client
docker service create
swarm manager
RAFT
API
accepts command and creates service object
orchestrator
reconciliation loop that creates tasks for service objects
allocater
allocates ip addresses to tasks
dispatcher
assigns tasks to nodes
scheduler
instructs a worker to run a task
worker node
container
worker
connects to dispatcher to check for assigned tasks
executor
executes tasks assigned to worker node

The underlying logic of Docker swarm mode is a general purpose scheduler and orchestrator. The service and task abstractions themselves are unaware of the containers they implement.



my-network
worker node
worker node
manager node
worker node
worker node
replicated service with 3 replicas
global service with replicas on every node

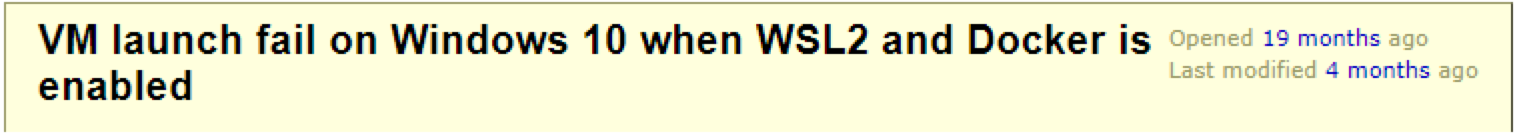
A task is a one-directional mechanism. It progresses monotonically through a series of states: assigned, prepared, running, etc. If the task fails the orchestrator removes the task and its container and then creates a new task to replace it according to the desired state specified by the service.

Source: https://docs.docker.com/engine/swarm/how-swarm-mode-works/services/

Demo : Docker Swarm

- Create two Docker virtual machines
- Create the Swarm
- List nodes
- Deploy Visualisation tool
- Deploy application from Docker compose example

Cancelled

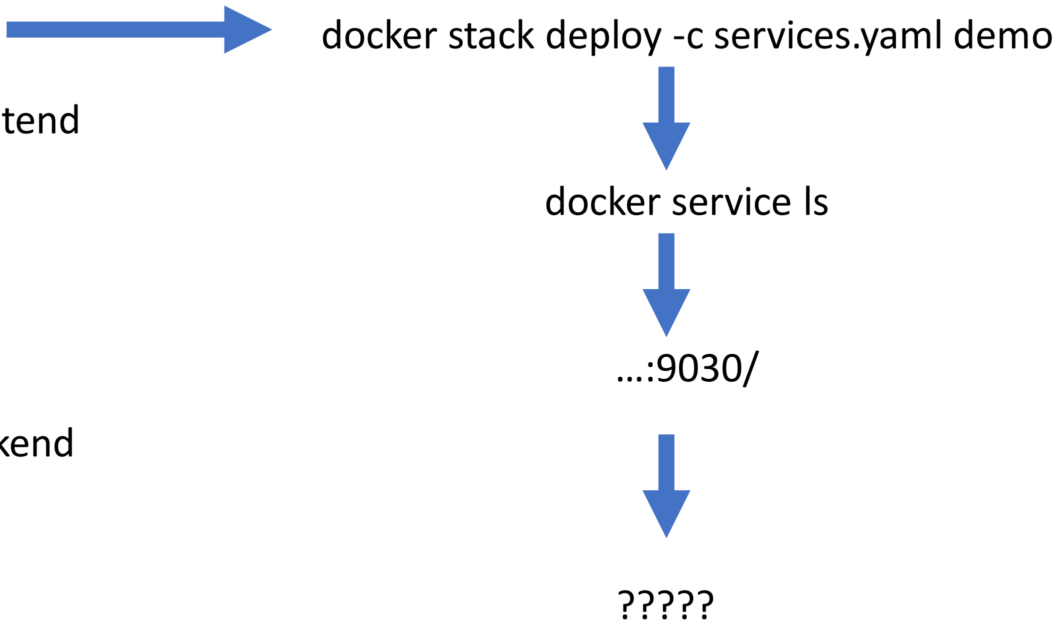


VM launch fail on Windows 10 when WSL2 and Docker is enabled

Opened 19 months ago
Last modified 4 months ago

Docker Swarm

```
version: '3'
services:
   frontend:
      image: skodapetr/tutorial-frontend
      ports:
         - 9030:8080
      networks:
         - public
         - local
   backend:
      image: skodapetr/tutorial-backend
      networks:
         - local
networks:
  public:
  local:
```



```
docker stack deploy -c services.yaml demo
```

```
docker service ls
```

…:9030/

?????

There is more … or is it?

- Rolling updates
- drain node (docker node update --availability drain worker)
- Configuration data (echo "This is a config" | docker config create my-config -)
- Secrets (printf "This is a secret" | docker secret create my_secret_data -)
- Persistent Volume Service
- Docker Swarm is Dead
- ….

Quality Attributes with Docker Swarm

Run-Time

- Availability +
- Performance +
- Security
- …

Non-Run-Time

- Modifiability
- Portability
- Testability
- Reusability
- Integrability
- …

What is Container Orchestration?

The million dollar question is, how do we build **reliable** and **highly available** systems from unreliable components?

The idea is to start with **redundancy**, **detect component failure**, and **replace** bad components quickly.

To achieve **high availability**, we must sacrifice consistency. But that doesn't mean that our system will have corrupt or arbitrary data. The keyword is **eventual consistency**.

Source: https://kubernetes.io/docs/tutorials/kubernetes-basics/scale/scale-interactive/

Kubernetes

Kubernetes

Kubernetes is an open-source system which allows you to run containers, manage them, automate deploys, scale deployments, create and configure ingresses, deploy stateless or stateful applications, and many other things.

… design to create a highly available applications.

Kubernetes Services and Swarm Services are very different! In Swarm, a service provides both scheduling and networking facilities, creating containers and providing tools for routing traffic to them. In Kubernetes, scheduling and networking are handled separately: deployments (or other controllers) handle the scheduling of containers as pods, while services are responsible only for adding networking features to those pods.

Kubernetes

Hackers Enlisted Tesla's Public Cloud to Mine Cryptocurrency

The attackers had apparently discovered that this particular Kubernetes console - an administrative portal for cloud application management—wasn't password protected and could therefore be accessed by anyone.

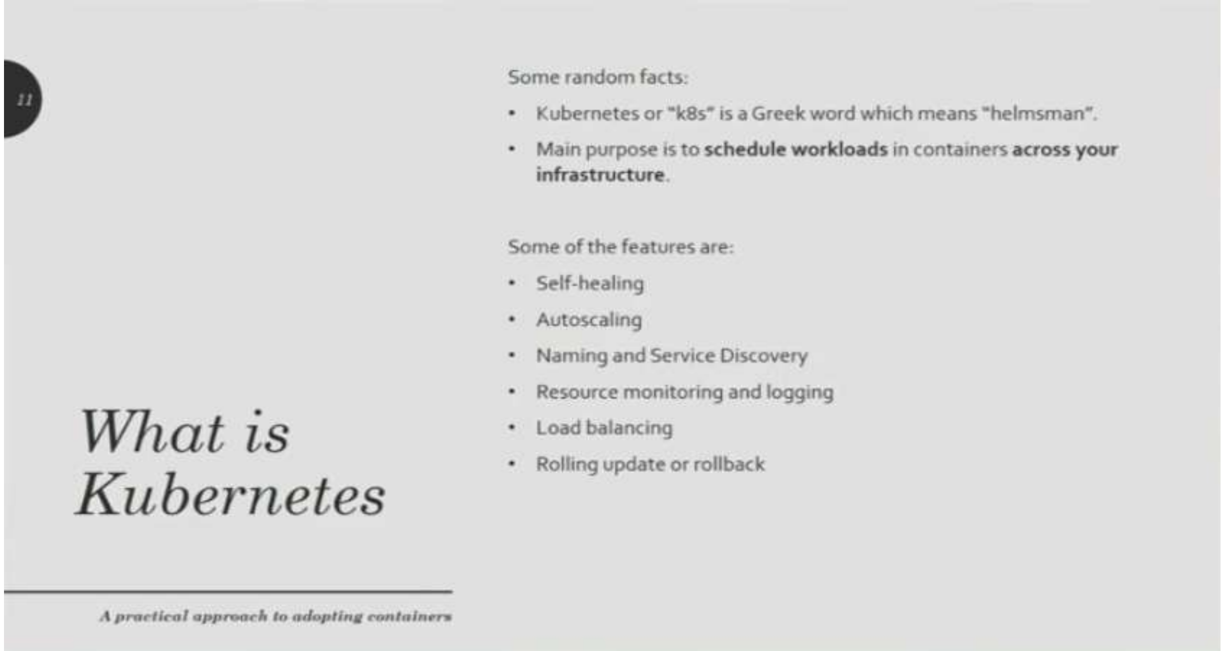
From there they would have found … that one of the console's "pods," or storage containers, included login credentials for a broader Tesla Amazon Web Services cloud environment.

Source: https://www.wired.com/story/cryptojacking-tesla-amazon-cloud/

Kubernetes doesn't know what images are doing. It's difficult to protect the image preparation and delivery pipeline (including image repositories). Speed of development and deployment of new images conflict with careful review changes.

Source: https://arstechnica.com/information-technology/2018/06/backdoored-images-downloaded-5-million-times-finally-removed-from-docker-hub/

Kubernetes



11

Some random facts:

- Kubernetes or "k8s" is a Greek word which means "helmsman".
- Main purpose is to **schedule workloads** in containers **across your infrastructure**.

Some of the features are:

- Self-healing
- Autoscaling
- Naming and Service Discovery
- Resource monitoring and logging
- Load balancing
- Rolling update or rollback

What is Kubernetes

A practical approach to adopting containers

Creates k8s clusters using voice.

Source: https://www.youtube.com/watch?v=LzEt5EYOguM

https://twitter.com/kelseyhightower

Kubernetes as a Service

- Abstract infrastructure
- Declare desired state and send to API (kubectl) ~ contract
- Easy to use difficult to master/manage/operate
- Kubernetes makes assumptions on how your application behaves
- Kubernetes operators (OperatorHub, Cassandra, …)

State primitives:

Pod

- Group of containers deployed to a single node (computer)
- Each have unique IP
- Data are lost without a persistent storage

Deployment

- Blueprint for pods
- Can scale up replicas of pods

Service

- An abstraction for pod identification

Volume

- Request for storage

Demo : Kubernetes

- LinkedPipes ETL
- K8s deployment
- kubectl apply –f ./k8s/

- Online playground
- Localhost minikube

Quality Attributes with Kubernetes

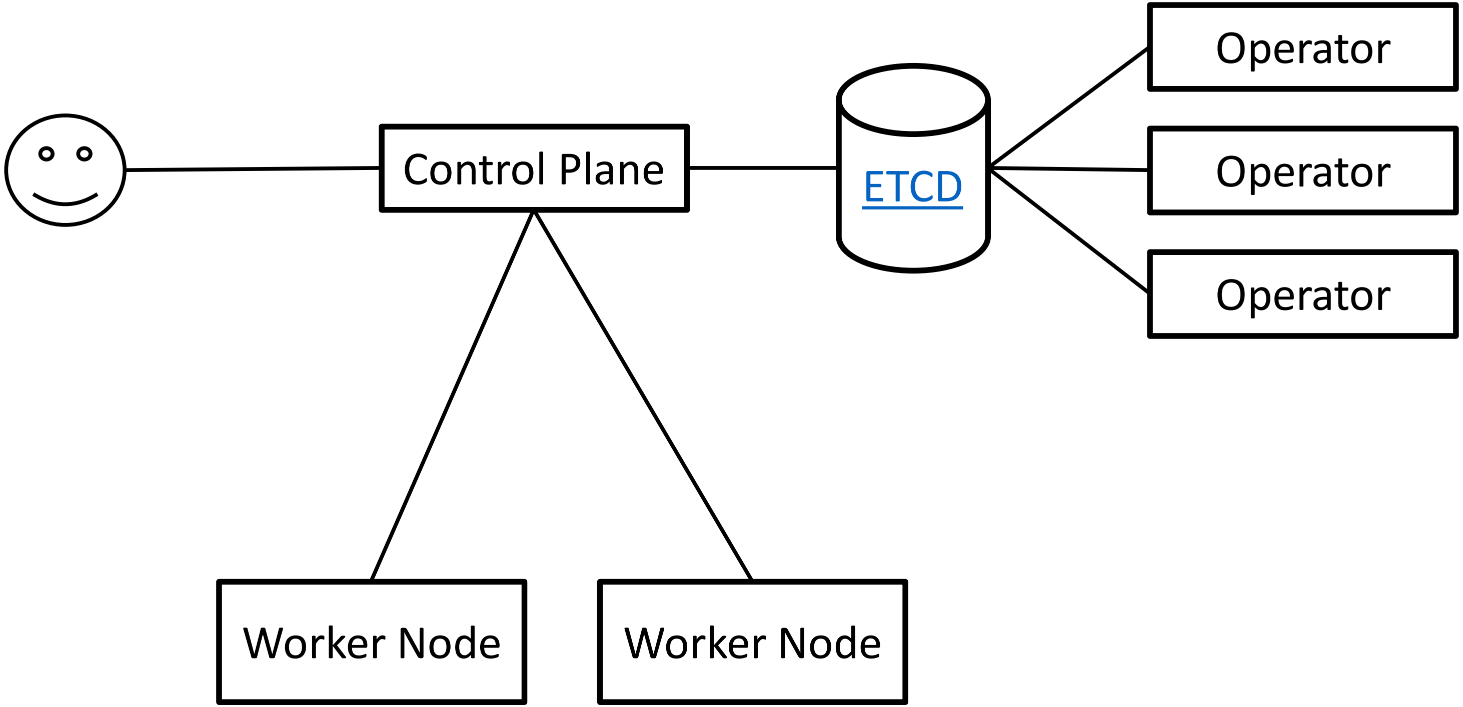
Run-Time

- Availability ++
- Performance ++
- Security
- …

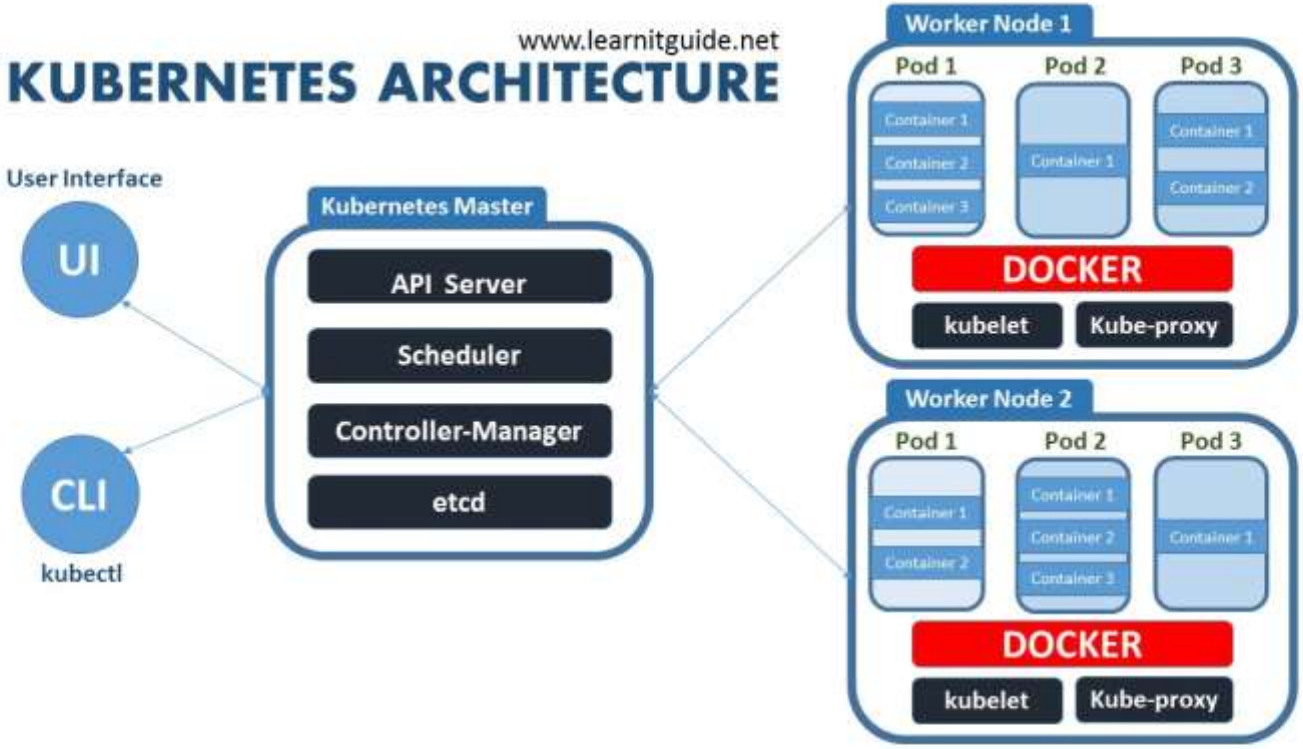
Non-Run-Time

- Modifiability
- Portability
- Testability
- Reusability
- Integrability
- …

Kubernetes Super-Simplified



Kubernetes Simplified



Source: https://www.learnitguide.net/2018/08/what-is-kubernetes-learn-kubernetes.html

There is a lot more …

- Highly customizable
- It's like a Linux distribution ….
  OpenShift (Learn, Deploy a Spring application on OpenShift, MiniShift, K8sToOpenShift)
- Package management: OperatorHub / Rancher / Helm / …
- Rolling updates / Canary deployments / …
- Auto scaling
- Federates clusters
- Secrets / Jobs / ….
- OpenFaaS
- ….

I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.
@ScottAdamsSays
Dilbert.com
YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE.
JUST PUT IT IN CONTAINERS.
11-08-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel
YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS.
KUBERNETES.

I can only …