The whale is dead - containers without Docker
Timo Heinrichs

Cloud Platform Engineer
› in second generation
› but with identity crisis

When I grow up, I want to be a Developer :)

Here’s inovex!

Look mum! I’m on the CNCF landscape!!!
Container technology timeline

1979

1979

2005

2005

2008

2008

2013

2013

2014/2015

2014/2015

chroot

openvz

lxc

docker

libcontainer/runc

Docker 0.6 with lxc

```
718 // Program
719 params = append(params, "--", container.Path)
720 params = append(params, container.Args...)
721 container.cmd = exec.Command("lxc-start", params...)
```
Container basics

› OS-level “virtualization”
› It’s a process
› Isolation works with (kernel) namespaces
  - pids, ipc, time, etc.
  - cgroups
  - network namespaces
  - overlay fs
› It’s not a VM!
Repeat after me:
IT’S NOT A VM!
Virtual Machines

Containers

https://www.weave.works/blog/a-practical-guide-to-choosing-between-docker-containers-and-vms
“The Open Container Initiative is an open governance structure for the express purpose of creating open industry standards around container formats and runtimes.”
Original docker image format is now industry standard
It’s JSON and .tar.gz ... Don’t be sad!
Media Types between Docker and OCI different but compatible
runtime-spec

- Original docker runtime is now industry standard

Defines:
- On-Disk format “FS Bundle”
- lifecycle verbs
  - create, start, kill, delete, state
- runc “reference” implementation

https://www.docker.com/blog/docker-0-9-introducing-execution-drivers-and-libcontainer/
Build, ship and run...but not anywhere!

- **Build**
  - `docker build`
  - Works fine on my machine.
  - But not great in pipelines!

- **Ship**
  - `docker push / pull`
  - Works fine on my machine.
  - But needs central registry & automation

- **Run**
  - `docker run`
  - Works fine on my machine.
  - But not in kubernetes!
› From Google
› Build (OCI) images
› Running in containers and without root possible

docker run \
  -v "$HOME"/.config/gcloud:/root/.config/gcloud \
  -v /path/to/context:/workspace \
gcr.io/kaniko-project/executor:latest \
  --dockerfile /workspace/Dockerfile \
  --destination "gcr.io/$PROJECT_ID/$IMAGE_NAME:$TAG" \
  --context dir://workspace/
From “containers” organization (RedHat)
Focus on building OCI images
Drop-in replacement with `buildah bud -f Dockerfile`

```bash
ctr1=$(buildah from "${1:-fedora}")

## Get all updates and install our minimal httpd server
buildah run "$ctr1" -- dnf update -y
buildah run "$ctr1" -- dnf install -y lighttpd
```
From “containers” organization (RedHat)
Drop-in replacement for docker client
rootless
daemonless
REST API
podman-compose
knows the “pod” concept
Additional tools

› Skopeo
  - Copy / Sync images between registries
  - List image tags, mark for deletion

› img
  - Image builder, built on top of **BuildKit**
  - `build, tag, push, pull, login, logout, save`

› reg
  - download layers from images
  - list all repos in a registry
  - Get vulnerability with clair image scanning

img & reg don’t have a cool logo. They probably suck...
The role of Kubernetes

- First docker alternative rkt already in 1.3
- Needs no `build,tag,push,volume,network`
- Container-Runtime-Interface for pluggable runtime
- `dockershim` component is deprecated now
- Removed in Kubernetes after 1.22 (later this year)

https://kubernetes.io/blog/2016/12/container-runtime-interface-cri-in-kubernetes/
Graduated CNCF project
“industry-standard container runtime”
Probably the thing you’ve been using all the time!
Use `ctr` or `crictl` to manage pods/containers

https://kubernetes.io/blog/2018/05/24/kubernetes-containerd-integration-goes-ga/
You are here

https://www.docker.com/blog/introducing-docker-engine-18-09/
Made by RedHat

Default in OpenShift

Purpose built as a kubernetes CRI

Use **cri-ctl** to manage pods/containers

https://cri-o.io/
Kata Containers

Additional isolation with a lightweight VM and individual kernels

Isolation by namespaces, cgroups with shared kernel
gVisor is an application kernel for containers that provides efficient defense-in-depth anywhere.
Docker is dead, right?

› Not yet!
› Docker Enterprise -> Mirantis k8s Engine
› Focus is shifted towards developers
› Default for poor Mac Users
› Irrelevant for running large scale containers
“I only understand train station!”

Kubernetes Administration Training

This course covers the core concepts typically used to create and manage a Kubernetes cluster in a productive environment.

Target audience: IT Engineers with Linux knowledge

Length: 4 days

Dates:
01.03.-04.03.2021 (remote, EN)

Times: 9 am - 5 pm

Number of participants: min. 3, max. 12

https://www.inovex.de/de/leistungen/trainings-workshops/kubernetes-administration-training/
https://www.inovex.de/en/our-services/training-workshops/kubernetes-administration-training