

---

# Introduction to Docker & Kubernetes

Ondrej Sika

Freelance DevOps Consultant & Lecturer

[ondrej@sika.io](mailto:ondrej@sika.io)

@ondrejsika

---

# About me

---

My name is **Ondrej Sika**, I am an IT & DevOps consultant, architect and lecturer.

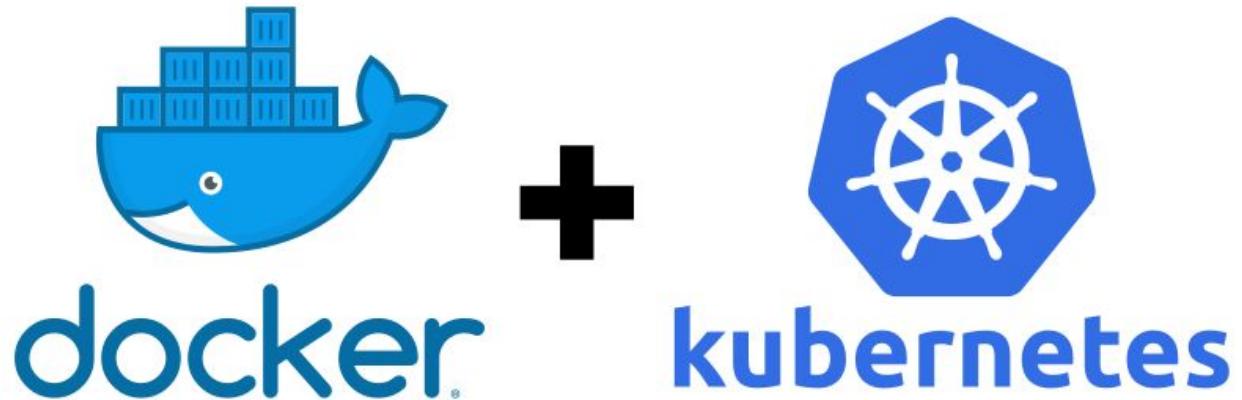
I'm boosting effectiveness & productivity of software development teams by using right tools and techniques which lead to faster development and reliable operation of software products.

I help companies to set up whole DevOps pipeline using training, consulting and short term project work.



# Agenda

- DevOps
- Docker
- Kubernetes
- Alternatives
- Summary



---

# DevOps

---

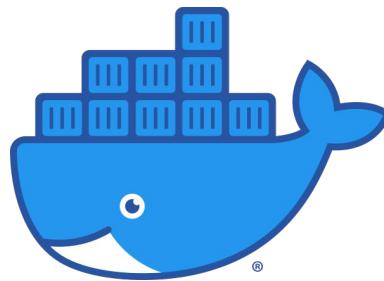
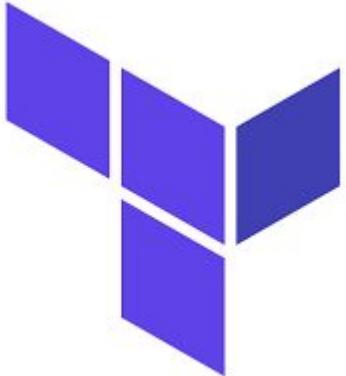
# What is DevOps?

---

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

Source: <https://aws.amazon.com/devops/what-is-devops/>

---



---

# What does it mean?

- "Agile Infrastructure" or "Agile Operations"
  - Rapid Delivery - Deliver changes automatically into production (staging, ...)
  - Reliability - People do mistakes, script don't.
  - Scaling - Easy scaling using Clouds, Kubernetes, Serverless, ...
  - Infrastructure as a Code - Treat your Infrastructure like a code (Terraform, ...)
  - Security - Security policy as a code
-

---

# Rapid Delivery (CI/CD)

- Continuous Integration
  - Integrate every
- Continuous Delivery
  - Deliver changes automatically into production (staging, ...)
- Tools for CI/CD:
  - Gitlab CI
  - Jenkins

---

# Reliability

- People make mistakes (especially under pressure), scripts don't
- HA Infrastructure (no single point of failure)
- Easy investigation using Git
- Automatic rollbacks in case of fail after deployments

# Infrastructure as a Code

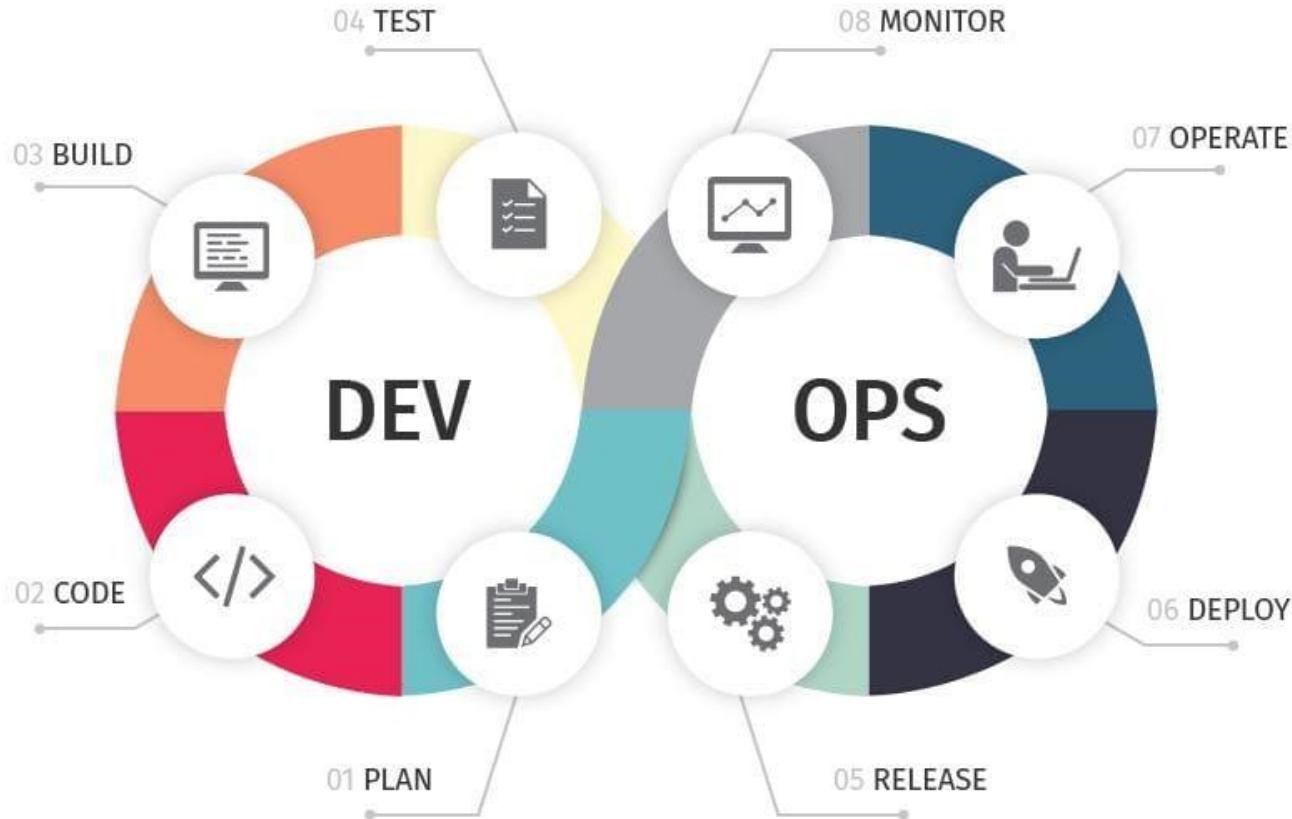
---

- Git Versioned
  - You can treat your infrastructure as other code - merge requests, CI, ...
- Automatic documentation
  - You can generate docs from the code
  - `terraform graph -type=refresh | dot -Tsvg > infrastructure.svg`
- Simple Scaling
  - In infrastructure definition code
  - Auto scaling (Kubernetes, Auto Scaling Groups)
- Reliable Upgrades
  - Review (merge requests) upgrades before applies
  - Rollbacks of infrastructure changes

---

# Easy & Secure Scaling

- Infrastructure as a Code
  - Scaling is easy and secure in Infrastructure as a Code
  - Terraform, Cloud Formation
- Autoscaling
  - Applications in Kubernetes
  - Nodes of Clusters (AWS, Azure, ...)
  - Auto Scaling Groups



---

# Modern Open Source Tools for DevOps

- SCM - Git
  - CI/CD - Gitlab CI / Jenkins
  - Container Engine - Docker
  - Orchestrator - Kubernetes, Swarm
  - Metrics & Monitoring - Prometheus
  - Logging - ELK, EFK
  - Provisioning - Ansible, Puppet
  - Infrastructure - Terraform
-

# 12 Factor Apps

---

- 12 rules how to write modern application
- Rules are about:
  - Sustainable development & operation
  - Shipping your code (product)
  - Configuration
  - Scaling
  - Operations - Logs, Admin process, ..
- Your Dev & Ops should read it
- <https://12factor.net/>

---

Source: <https://12factor.net/>

---

# Why Docker & Kubernetes?

---

---

# Why Docker & Kubernetes?

- Unify your environment
  - You need just Kubernetes Cluster (or machines with Docker) to run any application
  - Simple CI stack - Unified test, staging & production env
  - Solid role separation (but on shared codebase)
    - Devs: Dockerfile & Kubernetes manifest, ...
    - Ops: Kubernetes Clusters, Terraform manifests, ...
- Bulk deployments & management
  - Treat your deployments like a cattle, not a pets
- Deploy desired state
  - Declarative approach (instead of imperative)

---

# Docker

---

# What is Docker

---

Docker is a set of platform-as-a-service products that use OS-level virtualization to deliver software in packages called containers.

Source: [https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))

---

# What is a Container

---

Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels.

Source: [https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))

---

---

# Docker for Traditional Applications

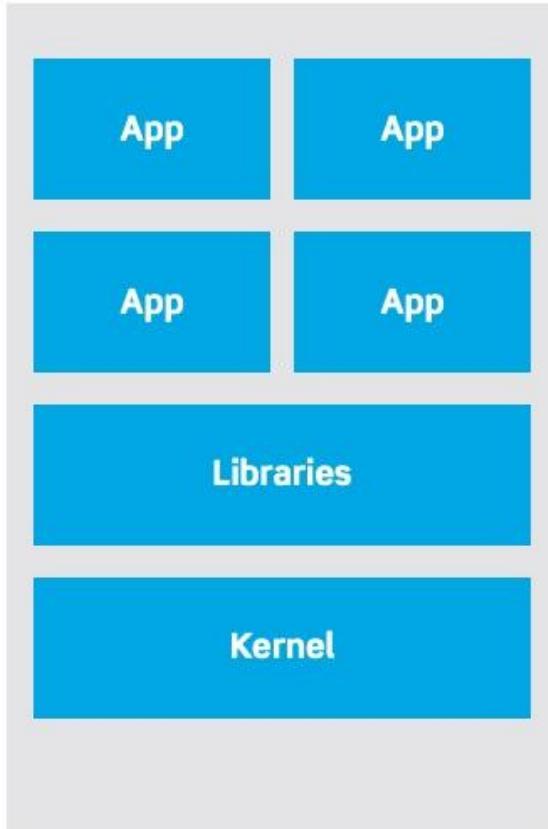
Make your application portable (able to  
deploy to new unified infrastructure)  
without touching code.

---

# Docker for Traditional Applications

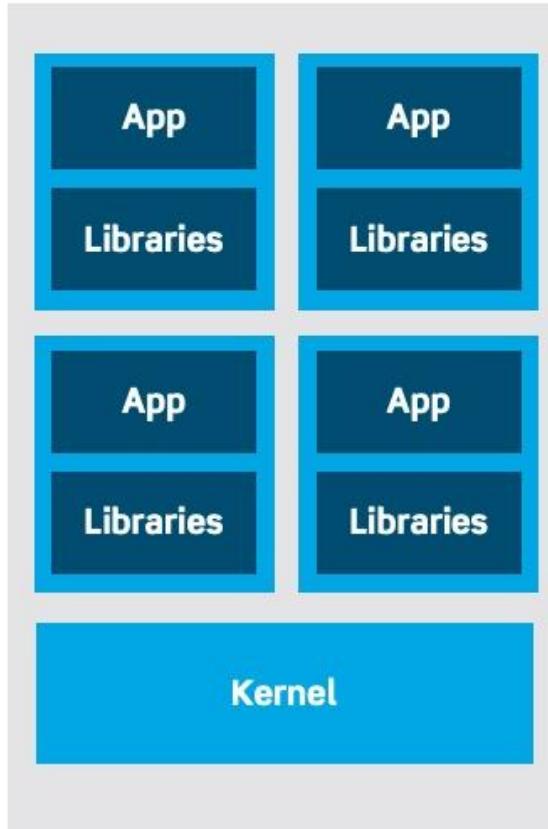
- Be able to add your current application into DevOps pipeline
- Be able to fast & easily deploy your current application to various unified environments (machines or clusters with Docker)
- Make environment (libraries, dependencies, ...) as part of application (source code)
- Deploy application with libraries & dependencies instead of installing dependencies on production servers. It's faster and more reliable approach.
  - Saves your production environment costs (resources) and minimize downtime

## The old way: Applications on host



Heavyweight, non-portable  
Relies on OS package manager

## The new way: Deploy containers



Small and fast, portable  
Uses OS-level virtualization

---

# Docker for Microservices

Docker is a simplest way how to build,  
ship & run microservices. In containers.

---

# Docker for Microservices & DevOps

- Simple integrations with various CI/CD tools
  - Fast, repeatable & cached builds
  - Simple application distribution through Registry and Docker Trusted Registry
  - Be able to deploy several times per day
  - Defines simple interface for communication between containers and underlying layer (kubernetes or hardware)
-

---

# Docker Quick Start

---

---

# Install Docker

Mac

```
brew cask install docker
```

Windows

```
choco install docker-desktop
```

Linux

<https://docs.docker.com/install/linux/docker-ce/debian/>

---

---

# System wide info

```
docker version # print version
```

```
docker info # system wide information
```

```
docker system df # docker disk usage
```

```
docker system prune # cleanup unused data
```

---

```
ondrej@sika-macbookpro:~$ docker version
```

```
Client: Docker Engine - Community
```

```
 Version: 19.03.4
```

```
 API version: 1.40
```

```
 Go version: go1.12.10
```

```
 Git commit: 9013bf5
```

```
 Built: Thu Oct 17 23:44:48 2019
```

```
 OS/Arch: darwin/amd64
```

```
 Experimental: true
```

```
Server: Docker Engine - Community
```

```
Engine:
```

```
 Version: 19.03.4
```

```
 API version: 1.40 (minimum version 1.12)
```

```
 Go version: go1.12.10
```

```
 Git commit: 9013bf5
```

```
 Built: Thu Oct 17 23:50:38 2019
```

```
 OS/Arch: linux/amd64
```

```
 Experimental: false
```

```
containerd:
```

```
 Version: v1.2.10
```

```
 GitCommit: b34a5c8af56e510852c35414db4c1f4fa6172339
```

```
runc:
```

```
 Version: 1.0.0-rc8+dev
```

```
 GitCommit: 3e425f80a8c931f88e6d94a8c831b9d5aa481657
```

```
docker-init:
```

```
 Version: 0.18.0
```

```
 GitCommit: fec3683
```

```
ondrej@sika-macbookpro:~$ █
```

```
ondrej@sika-macbookpro:~$ docker system info
```

Client:

Debug Mode: false

Plugins:

app: Docker Application (Docker Inc., v0.8.0)

buildx: Build with BuildKit (Docker Inc., v0.3.1-tp-docker)

Server:

Containers: 27

Running: 1

Paused: 0

Stopped: 26

Images: 216

Server Version: 19.03.4

Storage Driver: overlay2

Backing Filesystem: extfs

Supports d\_type: true

Native Overlay Diff: true

Logging Driver: json-file

Cgroup Driver: cgroupfs

Plugins:

Volume: local

Network: bridge host ipvlan macvlan null overlay

Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog

Swarm: inactive

Runtimes: runc

Default Runtime: runc

Init Binary: docker-init

containerd version: b34a5c8af56e510852c35414db4c1f4fa6172339

runc version: 3e425f80a8c931f88e6d94a8c831b9d5aa481657

init version: fec3683

Security Options:

seccomp

- - - - -

```
ondrej@sika-macbookpro:~$ docker system df
```

TYPE	TOTAL	ACTIVE	SIZE	RECLAIMABLE
Images	113	18	6.168GB	6.116GB (99%)
Containers	27	0	18.29kB	18.29kB (100%)
Local Volumes	21	10	392.2MB	1.005kB (0%)
Build Cache	89	0	36.32MB	36.32MB

```
ondrej@sika-macbookpro:~$ █
```

```
ondrej@sika-macbookpro:~$ docker system prune
```

WARNING! This will remove:

- all stopped containers
- all networks not used by at least one container
- all dangling images
- all dangling build cache

Are you sure you want to continue? [y/N] y

Deleted Containers:

```
aa0c5aeea63927bf2270a7b30fffe753c779f00488b556d0bc72e48da9988a2
a830580dc9b8c51ab171ae0b3610e9b22528a894ac8b71c4c6057cf6b4c6146
8e51e938aba6def2b17660d9190f888ccf65c8a513106cc68fd906148af12f38
a87842ebe2fc659c16fb5845e8b24712e6fc7da052a5ff65cf8302b2a37ad4f0
7a6b9d3b2350e978c316457a899e5081fb85b81300f8bd2c117ecad78416e5ef
8c0f07124c0c30b4f9bc425158f578921a79ae5356c866a362414214c69dc91d
5f21f326961794babdb8687410cec6a2a8c2542751345c98cffd01bd3acb8ea1
6c476dbc088aa8cd9a538fad21e568d86f6accf54d654b6da5e46253073bb1d
2accdc4815af416c1955ed6a88f0aac55cc6921106ccb77ab65f187a7cef267
21dd8f1bc1e0f874aa2c64d9672692b5587c3ae0ea73aa013542d1d6b43cd48f
cc05bcfb288d9f61ed370008509b6bc2a81272056164db62f179829f878b1d24
6a426d00b1fd05ed9c41ee43eb76f29e88fd1f7432cc447950eaa72815971510
60d63960ba063151c52dab4ca846e06a69edc71e7328044a78d2a1ebf7eb19be
dd368942c604ce0a6a41285f9e3133b6296fbeec068caf46686535f061364bbd
02f64a7229a5392c6828f0251eb49eab9ed33f458790341625fb6dc10a0611b
6f1d012c3730685c64031fce7bf3042259c545b2763c065495ee744e6a3f3d79
51040efb94aaff31b708be6d3f9b1d5622ba253e46003b3bbcd61af689a07bcb
6cbfe50ef99d87b72bf13093cedd768fbac2b880e6f37ceea861ee253c6b4aed
f7829033ef0338eb4886ffb9a175d54eede376ef757c4e1f7dc6883578e3f830
b50a477216b3ffea6e0c2df94b2956ba064eca31f35c525002775f02a0988b10
f4353ecfff0e986366842dfba993599093c2fb9aee9b90f6a3fa375f402784e4
f6f7ddffbb3c6f8fe8097402364f669216acf9bd322b84646336eae942ea4aab
839d33e5bca6a006171d7bbad2935f50a0a1cdf43af34b1c2c86aed3a80d603
274291be097151ecaf466ca27b2cf9c0717126958dfddbc当地83572a31dc29d6
```

---

# Run Docker Container

Hello world

```
docker run hello-world
```

Simple web server

```
docker run -p 80:80 ondrejsika/hellojavadays2019
```

---

```
ondrej@sika-macbookpro:~$ docker run hello-world
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

```
https://hub.docker.com/
```

For more examples and ideas, visit:

```
https://docs.docker.com/get-started/
```

```
ondrej@sika-macbookpro:~$ █
```

```
ondrej@sika-macbookpro:~$ docker run -p 80:80 ondrejsika/hellojavadays2019
```

```
Server startded.
```

---

# Docker Image

```
docker image ls # list all images
```

```
docker image ls <image> # list all images
```

```
docker image ls -q # quiet output, just IDs
```

```
docker image rm <image> # remove image
```

---

```
ondrej@sika-macbookpro:~$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
postgres	9.6	afd8110f1813	3 weeks ago	211MB
postgres	11	5f1485c70c9a	3 weeks ago	293MB
postgres	12	f88dfa384cc4	3 weeks ago	348MB
debian	latest	8e9f8546050d	3 weeks ago	114MB
hello-world	latest	fce289e99eb9	10 months ago	1.84kB
ondrejsika/hellojavadays2019	latest	0162616d1d84	14 months ago	6.56MB

```
ondrej@sika-macbookpro:~$ █
```

```
ondrej@sika-macbookpro:~$ docker image ls postgres
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
postgres	9.6	afd8110f1813	3 weeks ago	211MB
postgres	11	5f1485c70c9a	3 weeks ago	293MB
postgres	12	f88dfa384cc4	3 weeks ago	348MB

```
ondrej@sika-macbookpro:~$ █
```

```
ondrej@sika-macbookpro:~$ docker image ls -q
```

```
af8110f1813  
5f1485c70c9a  
f88dfa384cc4  
8e9f8546050d  
fce289e99eb9  
0162616d1d84
```

```
ondrej@sika-macbookpro:~$ echo docker image rm `docker image ls -q`  
docker image rm af8110f1813 5f1485c70c9a f88dfa384cc4 8e9f8546050d fce289e99eb9 0162616d1d84
```

```
ondrej@sika-macbookpro:~$ █
```

```
ondrej@sika-macbookpro:~$ docker image rm hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
Deleted: sha256:fce289e99eb9bca977dae136fbe2a82b6b7d4c372474c9235adc1741675f587e
Deleted: sha256:af0b15c8625bb1938f1d7b17081031f649fd14e6b233688eea3c5483994a66a3
ondrej@sika-macbookpro:~$ █
```

---

# Docker Run

```
docker run [args..] <image> [<command>]
```

```
# Eg.:
```

```
docker run hello-world
```

```
docker run debian cat /etc/os-release
```

```
docker run ubuntu cat /etc/os-release
```

```
docker run -ti debian
```

---

```
ondrej@sika-macbookpro:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:c3b4ada4687bbbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

```
https://hub.docker.com/
```

For more examples and ideas, visit:

```
https://docs.docker.com/get-started/
```

```
ondrej@sika-macbookpro:~$ █
```

```
ondrej@sika-macbookpro:~$ docker run debian cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"

ondrej@sika-macbookpro:~$ docker run ubuntu cat /etc/os-release
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
7ddbc47eeb70: Pull complete
c1bbdc448b72: Pull complete
8c3b70e39044: Pull complete
45d437916d57: Pull complete
Digest: sha256:6e9f67fa63b0323e9a1e587fd71c561ba48a034504fb804fd26fd8800039835d
Status: Downloaded newer image for ubuntu:latest
NAME="Ubuntu"
VERSION="18.04.3 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.3 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
ondrej@sika-macbookpro:~$
```

```
ondrej@sika-macbookpro:~$ docker run -ti debian
root@5be969946991:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@5be969946991:/# cat /etc/apt/sources.list
# deb http://snapshot.debian.org/archive/debian/20191014T000000Z buster main
deb http://deb.debian.org/debian buster main
# deb http://snapshot.debian.org/archive/debian-security/20191014T000000Z buster/updates main
deb http://security.debian.org/debian-security buster/updates main
# deb http://snapshot.debian.org/archive/debian/20191014T000000Z buster-updates main
deb http://deb.debian.org/debian buster-updates main
root@5be969946991:/# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@5be969946991:/# apt
apt 1.8.2 (amd64)
Usage: apt [options] command
```

apt is a commandline package manager and provides commands for searching and managing as well as querying information about packages. It provides the same functionality as the specialized APT tools, like apt-get and apt-cache, but enables options more suitable for interactive use by default.

Most used commands:

- list - list packages based on package names
- search - search in package descriptions

---

# Common Docker Run Parameters

**--name <name>**

**--rm** - remove container after stop

**-d** - run in detached mode

**-ti** - map TTY a STDIN (for bash eg.)

**-e <variable>=<value>** - set ENV variable

---

```
ondrej@sika-macbookpro:~$ docker run --help
```

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

Options:

--add-host list	Add a custom host-to-IP mapping (host:ip)
-a, --attach list	Attach to STDIN, STDOUT or STDERR
--blkio-weight uint16	Block IO (relative weight), between 10 and 1000, or 0 to disable (default 0)
--blkio-weight-device list	Block IO weight (relative device weight) (default [])
--cap-add list	Add Linux capabilities
--cap-drop list	Drop Linux capabilities
--cgroup-parent string	Optional parent cgroup for the container
--cidfile string	Write the container ID to the file
--cpu-period int	Limit CPU CFS (Completely Fair Scheduler) period
--cpu-quota int	Limit CPU CFS (Completely Fair Scheduler) quota
--cpu-rt-period int	Limit CPU real-time period in microseconds
--cpu-rt-runtime int	Limit CPU real-time runtime in microseconds
-c, --cpu-shares int	CPU shares (relative weight)
--cpus decimal	Number of CPUs
--cpuset-cpus string	CPUs in which to allow execution (0-3, 0,1)
--cpuset-mems string	MEMs in which to allow execution (0-3, 0,1)
-d, --detach	Run container in background and print container ID
--detach-keys string	Override the key sequence for detaching a container
--device list	Add a host device to the container
--device-cgroup-rule list	Add a rule to the cgroup allowed devices list
--device-read-bps list	Limit read rate (bytes per second) from a device (default [])
--device-read-iops list	Limit read rate (IO per second) from a device (default [])
--device-write-bps list	Limit write rate (bytes per second) to a device (default [])
--device-write-iops list	Limit write rate (IO per second) to a device (default [])
--disable-content-trust	Skip image verification (default true)
--dns list	Set custom DNS servers

---

# Work with Containers

**docker ps** - list containers

**docker start** <container>

**docker stop** <container>

**docker restart** <container>

**docker logs** <container> - show STDOUT & STDERR

**docker rm** <container> - remove container

---

```
ondrej@sika-macbookpro:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
ondrej@sika-macbookpro:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
ondrej@sika-macbookpro:~$ docker run --name nginx -d nginx
```

```
79d0bf240497306eeacb31c1577098f90533284f65f7698d53ceed75b3451a8
```

```
ondrej@sika-macbookpro:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
79d0bf240497 nginx "nginx -g 'daemon of..." 4 seconds ago Up 3 seconds 80/tcp
```

```
ondrej@sika-macbookpro:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
79d0bf240497 nginx "nginx -g 'daemon of..." 11 seconds ago Up 9 seconds 80/tcp
```

```
ondrej@sika-macbookpro:~$ docker stop nginx
```

```
nginx
```

```
ondrej@sika-macbookpro:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
ondrej@sika-macbookpro:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
79d0bf240497 nginx "nginx -g 'daemon of..." 23 seconds ago Exited (0) 4 seconds ago
```

```
ondrej@sika-macbookpro:~$ docker start nginx
```

```
nginx
```

```
ondrej@sika-macbookpro:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
79d0bf240497 nginx "nginx -g 'daemon of..." 35 seconds ago Up 2 seconds 80/tcp
```

```
ondrej@sika-macbookpro:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
79d0bf240497 nginx "nginx -g 'daemon of..." 43 seconds ago Up 10 seconds 80/tcp
```

```
ondrej@sika-macbookpro:~$ docker stop nginx
```

```
nginx
```

```
ondrej@sika-macbookpro:~$ docker rm nginx
```

```
nginx
```

```
ondrej@sika-macbookpro:~$ █
```

```
ondrej@sika-macbookpro:~$ docker run --name postgres -d postgres  
b49001f1dfa08c0c5c45b169b2013bf84aa5bf05baf620563f644d5d3e253357
```

```
ondrej@sika-macbookpro:~$ docker logs postgres
```

```
The files belonging to this database system will be owned by user "postgres".  
This user must also own the server process.
```

```
The database cluster will be initialized with locale "en_US.utf8".
```

```
The default database encoding has accordingly been set to "UTF8".
```

```
The default text search configuration will be set to "english".
```

```
Data page checksums are disabled.
```

```
fixing permissions on existing directory /var/lib/postgresql/data ... ok  
creating subdirectories ... ok
```

```
selecting dynamic shared memory implementation ... posix
```

```
selecting default max_connections ... 100
```

```
selecting default shared_buffers ... 128MB
```

```
selecting default time zone ... Etc/UTC
```

```
creating configuration files ... ok
```

```
running bootstrap script ... ok
```

```
performing post-bootstrap initialization ... ok
```

```
syncing data to disk ... ok
```

```
initdb: warning: enabling "trust" authentication for local connections
```

```
You can change this by editing pg_hba.conf or using the option -A, or  
--auth-local and --auth-host, the next time you run initdb.
```

```
Success. You can now start the database server using:
```

```
pg_ctl -D /var/lib/postgresql/data -l logfile start
```

```
*****
```

```
WARNING: No password has been set for the database.
```

---

# Persistent Storage - Docker Volumes

Volumes are persistent data storage for containers.

Volumes can be shared between containers and data are written directly to host.

```
docker run -ti -v my-volume:/data debian
```

```
docker run -ti -v $(pwd)/my-data:/data debian
```

```
ondrej@sika-macbookpro:~$ docker run -ti -v my-volume:/data -w /data debian
root@7537ade4b254:/data# ls
root@7537ade4b254:/data# echo "Hello JavaDays" > hello.txt
root@7537ade4b254:/data# ls
hello.txt
root@7537ade4b254:/data# exit
ondrej@sika-macbookpro:~$ docker run -ti -v my-volume:/data -w /data debian
root@974504965e8e:/data# cat hello.txt
Hello JavaDays
root@974504965e8e:/data# exit
ondrej@sika-macbookpro:~$
```

---

# Port Forwarding

Docker can forward specific port from container to host.

```
docker run -p 80:80 ondrejsika/hellojavadays2019
```

---

```
ondrej@sika-macbookpro:~$ docker run --name hellojavadays -d -p 80:80 ondrejsika/hellojavadays2019  
f26cfe9316cef974ee84905e9065837be0932b0c055265c0642d855a5e59503a  
ondrej@sika-macbookpro:~$ curl localhost  
Hello JavaDays 2019 f26cfe9316ce  
ondrej@sika-macbookpro:~$ docker stop hellojavadays  
hellojavadays  
ondrej@sika-macbookpro:~$ docker rm hellojavadays  
hellojavadays  
ondrej@sika-macbookpro:~$ █
```

---

# Own Docker Images

---

# Dockerfile

**Dockerfiles** are used to produce docker images using reproducible builds.

Dockerfiles defines each layer for Docker Image Overlay2 filesystem

---

# .dockerignore

Ignore unnecessary files for docker build process. Speed up the build.

Same syntax as .gitignore

---

---

# Build Docker Image

```
docker build -t <image> <path>
```

```
docker build -f <dockerfile> -t <image> <path>
```

# Dockerfile

---

**FROM <image>** - define base image

**RUN <command>** - run command and save as layer

**COPY <local path> <image path>** - copy file or directory to image

**ENV <variable> <value>** - set ENV variable

**WORKDIR <path>** - change working directory

**VOLUME <path>** - define volume

**CMD <command>** - executable which you want to start in container

**EXPOSE <port>** - define port where container listen

---

---

# Example Dockerfile

```
FROM python:3.8-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY . .
CMD ["python", "app.py"]
EXPOSE 80
```

---

---

# Build

```
docker build -t ondrejsika/javadays2019-simple .
```

```
docker push ondrejsika/javadays2019-simple
```

```
ondrej@sika-macbookpro:~/javadays/simple$ docker build -t ondrejsika/javadays2019-simple .
Sending build context to Docker daemon 4.096kB
Step 1/7 : FROM python:3.8-slim
--> 6b52e13b1d32
Step 2/7 : WORKDIR /app
--> Running in 0f3f6ca397e3
Removing intermediate container 0f3f6ca397e3
--> 15b1cabf30fc
Step 3/7 : COPY requirements.txt .
--> a11a0e3ad428
Step 4/7 : RUN pip install -r requirements.txt
--> Running in c68c8adb93b8
Collecting flask
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-nor
Collecting itsdangerous>=0.24
  Downloading https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.
Collecting Werkzeug>=0.15
  Downloading https://files.pythonhosted.org/packages/ce/42/3aeda98f96e85fd26180534d36570e4d18108d62ae36f87694b476b83d6f/Werkzeug-0.16.0-py2.py3
Collecting click>=5.1
  Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abbc30d7257104c434fe0b07e5a195a6847506c074527aa599ec/Click-7.0-py2.py3-none-
Collecting Jinja2>=2.10.1
  Downloading https://files.pythonhosted.org/packages/65/e0/eb35e762802015cab1cce04e8a277b03f1d8e53da3ec3106882ec42558b/Jinja2-2.10.3-py2.py3-r
Collecting MarkupSafe>=0.23
  Downloading https://files.pythonhosted.org/packages/b9/2e/64db92e53b86efccfaea71321f597fa2e1b2bd3853d8ce658568f7a13094/MarkupSafe-1.1.1.tar.gz
Building wheels for collected packages: MarkupSafe
  Building wheel for MarkupSafe (setup.py): started
  Building wheel for MarkupSafe (setup.py): finished with status 'done'
  Created wheel for MarkupSafe: filename=MarkupSafe-1.1.1-cp38-none-any.whl size=12629 sha256=d8b8e075bd878cdfb66c7e4f238dfb989a9446cd7c9688d86c
  Stored in directory: /root/.cache/pip/wheels/f2/aa/04/0edf07a1b8a5f5f1aed7580ffb69ce8972edc16a505916a77
Successfully built MarkupSafe
Installing collected packages: itsdangerous, Werkzeug, click, MarkupSafe, Jinja2, flask
Successfully installed Jinja2-2.10.3 MarkupSafe-1.1.1 Werkzeug-0.16.0 click-7.0 flask-1.1.1 itsdangerous-1.1.0
Removing intermediate container c68c8adb93b8
--> 15b1cabf30fc
```

```
ondrej@sika-macbookpro:~/javadays/simple$ docker run --name hello -d -p 80:80 ondrejsika/javadays2019-simple  
0085f812479af03cbda6866e554f50180d9d763668b872cfb70984f9b3da9da7
```

```
ondrej@sika-macbookpro:~/javadays/simple$ curl localhost
```

```
Hello JavaDays 2019 0085f812479a
```

```
ondrej@sika-macbookpro:~/javadays/simple$ docker image ls python:3.8-slim
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
python	3.8-slim	6b52e13b1d32	3 weeks ago	194MB

```
ondrej@sika-macbookpro:~/javadays/simple$ docker image ls ondrejsika/javadays2019-simple
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ondrejsika/javadays2019-simple	latest	9f84d7cbde64	57 seconds ago	204MB

```
ondrej@sika-macbookpro:~/javadays/simple$ █
```

---

# Multi-Stage Build

---

---

# Multi-Stage Dockerfile

```
FROM java-jdk:... as build
RUN gradle assemble

FROM java-jre:...
COPY --from=build /build/demo.jar .
```

---

# Example Multi-Stage Dockerfile

```
FROM golang as build
WORKDIR /build
COPY app.go .
ENV CGO_ENABLED=0
RUN go build -a -ldflags \
    '-extldflags "-static"' app.go
```

```
FROM scratch
COPY --from=build /build/app .
CMD ["/app"]
```

---

---

# Build

```
docker build -t ondrejsika/javadays2019-multi-stage .
```

```
docker push ondrejsika/javadays2019-multi-stage
```

```
ondrej@sika-macbookpro:~/javadays/multi-stage$ docker build -t ondrejsika/javadays2019-multi-stage .
Sending build context to Docker daemon 4.096kB
Step 1/9 : FROM golang as build
--> 54e71dcaf7c
Step 2/9 : WORKDIR /build
--> Running in 5264f2f048fb
Removing intermediate container 5264f2f048fb
--> 10723e2783f4
Step 3/9 : COPY app.go .
--> e7fec387124b
Step 4/9 : ENV CGO_ENABLED=0
--> Running in f25f90104560
Removing intermediate container f25f90104560
--> f5fe5ecfd6c5
Step 5/9 : RUN go build -a -ldflags '-extlflags "-static"' app.go
--> Running in b85ac10a200c
Removing intermediate container b85ac10a200c
--> ade3c00f8377
Step 6/9 : FROM scratch
-->
Step 7/9 : COPY --from=build /build/app .
--> 9521a2aab2d5
Step 8/9 : CMD ["/app"]
--> Running in 198be6ca1f42
Removing intermediate container 198be6ca1f42
--> dfecd3a32166
Step 9/9 : EXPOSE 80
--> Running in 9aed32af957a
Removing intermediate container 9aed32af957a
--> 5b5cdacdb9ed
Successfully built 5b5cdacdb9ed
Successfully tagged ondrejsika/javadays2019-multi-stage:latest
ondrej@sika-macbookpro:~/javadays/multi-stage$ █
```

```
ondrej@sika-macbookpro:~/javadays/multi-stage$ docker run --name hello -d -p 80:80 ondrejsika/javadays2019-multi-stage  
b6829f2074cc7dacc71db8dd55298a79cc44613d747f0707cfcc43f6c63aa5120  
ondrej@sika-macbookpro:~/javadays/multi-stage$ curl localhost  
Hello JavaDays 2019 b6829f2074cc  
ondrej@sika-macbookpro:~/javadays/multi-stage$ docker image ls golang  
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE  
golang          latest        54e71dcafb7c   10 days ago   803MB  
ondrej@sika-macbookpro:~/javadays/multi-stage$ docker image ls ondrejsika/javadays2019-multi-stage  
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE  
ondrejsika/javadays2019-multi-stage  latest        5b5cdacdb9ed   About a minute ago  7.39MB  
ondrej@sika-macbookpro:~/javadays/multi-stage$ █
```

---

# Docker BuildKit

---

# Docker BuildKit

---

Docker has new build tool called BuildKit which can speedup your builds. For example, it build multiple stages in parallel and more. You can also extend Dockerfile functionality for caches, mounts, ...

- [https://docs.docker.com/develop/develop-images/build\\_enhancements/](https://docs.docker.com/develop/develop-images/build_enhancements/)
  - <https://github.com/moby/buildkit/blob/master/frontend/dockerfile/docs/experimental.md>
-

---

# BuildKit Dockerfile Example

```
# syntax = docker/dockerfile:experimental
FROM openjdk:jre
RUN --mount=type=cache,target=/cache/.m2 \
    --mount=type=cache,target=/cache/.gradle \
    make
```

---

# Docker Without Kubernetes

If you run small application or just one server, you don't need Kubernetes.

Take a look for:

- Docker Compose
  - Docker Swarm
-

---

# Demo Time

---

---

# Kubernetes

---

---

# What is Kubernetes?

A Production-Grade Container  
Orchestration System

# What is Kubernetes?

---

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Source: <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

---

---

# What does Kubernetes do?

Abstract away the underlying hardware

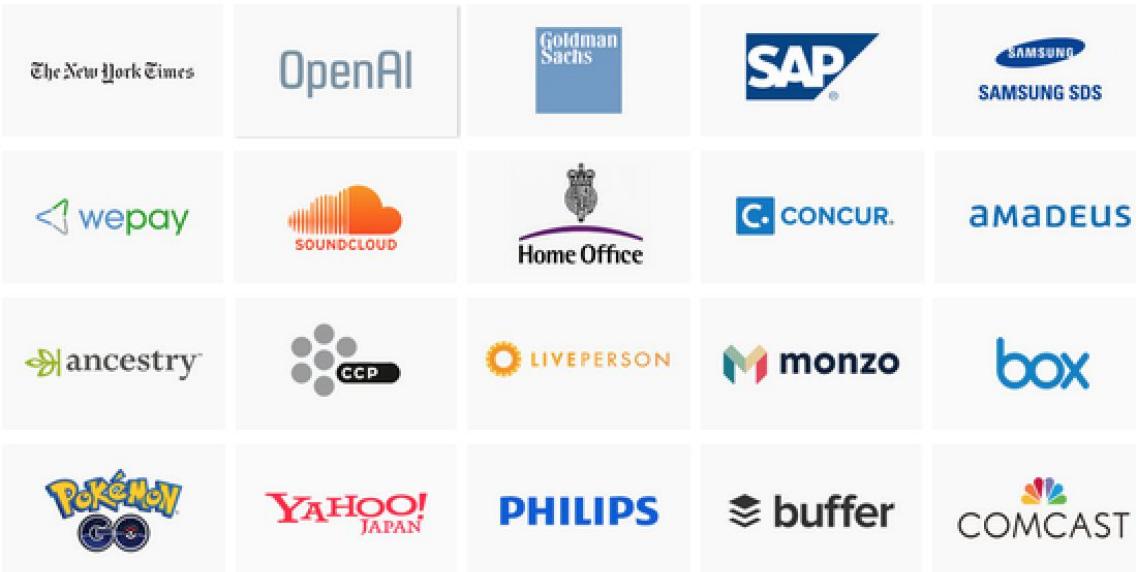
- Remove concept of nodes
- Manage your applications like cattle instead of like pets

Deploy your desired state

- You (admin) describe the desired state and kubernetes turn it into actual state
-

# Kubernetes Users

From small companies and startups to large enterprises



---

# No vendor lock

Kubernetes is no vendor lock to specific provider, you can run Kubernetes on:

- AWS
- GCP
- DigitalOcean
- Azure
- OpenStack
- or your private infrastructure

---

# Why (and when) you should use Kubernetes

- If you need HA
- If you have to manage applications on many servers
- If you don't want to care about servers (Kubernetes as a Service, IaaS)
- If you want easily deploy your Dockerized applications (IaaS)

---

# Which apps are suitable for Kubernetes?

- Stateless workers
- Batch processing
- Web Servers
- Mobile Backend

## Which not?

- Databases
  - Persistent data storages
-

# Core Concepts

---

**Pod** - The basic and atomically schedulable building block of Kubernetes, which is a single instance of app. Pods are mortal.

**Deployment** - Atomic update of Pods. Deployments contains Pod & ReplicaSet templates and keep running desired pods.

**Service** - Provide immortal IP address or DNS name for some selected pods.

**Ingress** - Provide external access to service using domain name.

Storage, Configuration, Monitoring, ...

---

---

# Kubernetes Cluster Components

**API Server** - Stateless API server backed by distributed **Etcd**

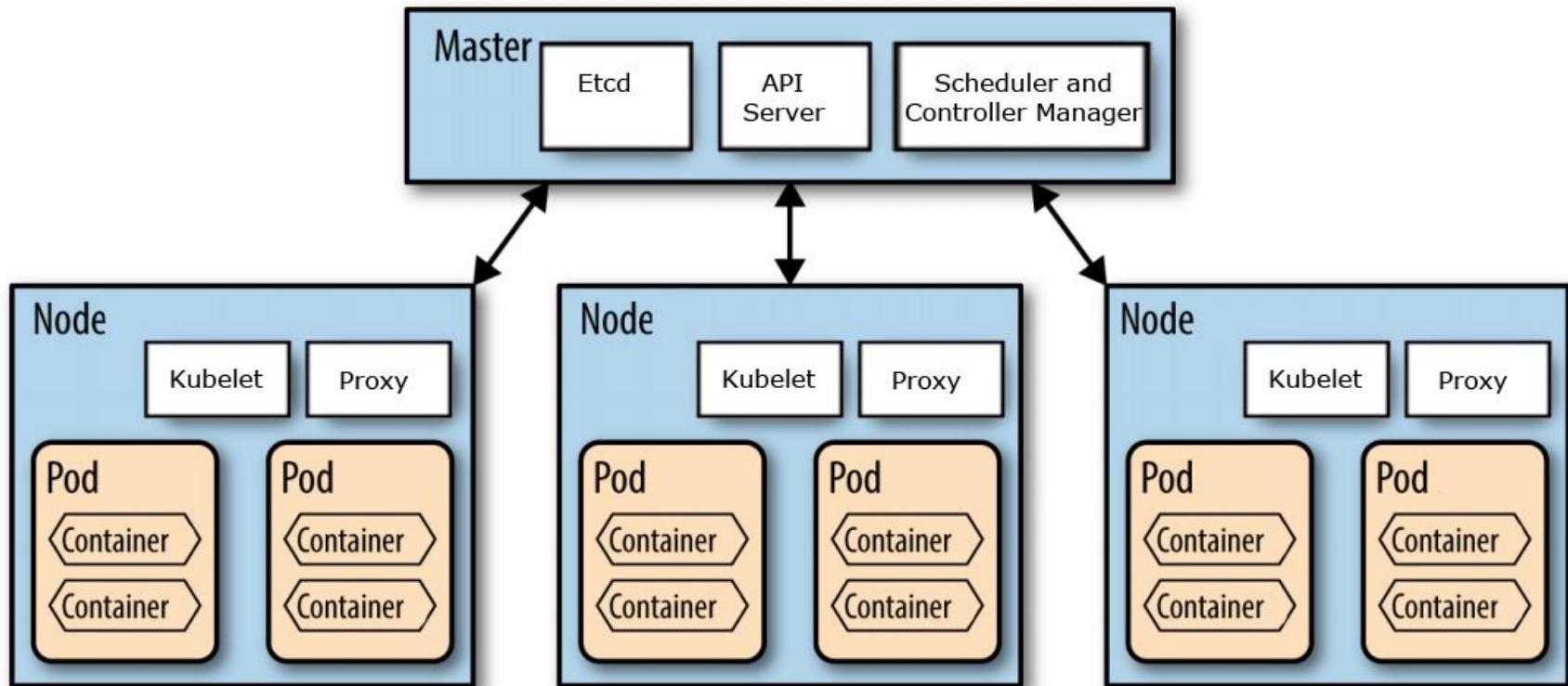
**Controller Manager** - ensure the actual state of the cluster equals the desired state

**Scheduler** - Schedule creations of Pods on a Nodes

**Kubelet** - Client for API Server, run Pods

**Kube Proxy** - Forward traffic into cluster

---



---

# Tools

**kubectl** - Kubernetes client (for CLI)

**helm** - Package manager for Kubernetes

**kubeadm** - Tool for Kubernetes cluster setup (on VMs)

**minikube** - Run Kubernetes locally for development

**kops** - Create Kubernetes cluster in cloud

---

---

# Kubernetes Cluster Components

**API Server** - Stateless API server backed by distributed **Etcd**

**Controller Manager** - ensure the actual state of the cluster equals the desired state

**Scheduler** - Schedule creations of Pods on a Nodes

**Kubelet** - Client for API Server, run Pods

**Kube Proxy** - Forward traffic into cluster

---

---

# Install Kubernetes Client

Mac

```
brew install kubernetes-cli
```

Windows

```
choco install kubernetes-cli
```

Linux

<https://kubernetes.io/docs/tasks/tools/install-kubectl/>

---

---

# Install Helm

Mac

```
brew install kubernetes-helm
```

Windows

```
choco install kubernetes-helm
```

Linux

<https://helm.sh/docs/install/>

---

---

# Setup Kubernetes Cluster

- Manually using **kubeadm**
- Using Ansible (Ansible use also **kubeadm**)
- On the Cloud using **kops** (creates EC2 instances & setup cluster there)
- Using **Terraform** or Cloud Formation

---

# Create Kubernetes cluster using Terraform

```
git clone git@github.com:ondrejsika/terraform-do-kubernetes-example.git  
cd terraform-do-kubernetes-example
```

```
terraform init  
terraform apply -auto-approve  
terraform output kubeconfig > kubeconfig
```

```
export KUBECONFIG=kubeconfig  
kubectl cluster-info  
kubectl get nodes
```

```
16 resource "digitalocean_kubernetes_cluster" "sikademo" {
17   name      = "sikademo"
18   region    = "fra1"
19   version   = "1.15.5-do.1"
20
21   node_pool {
22     name        = "sikademo"
23     size        = "s-2vcpu-2gb"
24     node_count = 3
25   }
26 }
27
28
29 resource "digitalocean_loadbalancer" "sikademo" {
30   name      = "sikademo"
31   region    = "fra1"
32
33   droplet_tag = "k8s:${digitalocean_kubernetes_cluster.sikademo.id}"
34
35   healthcheck {
36     port = 30001
37     protocol = "tcp"
38   }
39
40   forwarding_rule {
41     entry_port  = 80
42     target_port = 30001
43     entry_protocol = "tcp"
44     target_protocol = "tcp"
45   }
46
47   forwarding_rule {
48     entry_port  = 443
49     target_port = 30001
50     protocol = "https"
51   }
52 }
```

```
ondrej@sika-macbookpro:~/Projects/terraform/terraform-do-kubernetes-example (master)$ tf init
```

Initializing the backend...

Initializing provider plugins...

The following providers do not have any version constraints in configuration,  
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking  
changes, it is recommended to add version = "... constraints to the  
corresponding provider blocks in configuration, with the constraint strings  
suggested below.

```
* provider.cloudflare: version = "~> 1.16"  
* provider.digitalocean: version = "~> 1.4"
```

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see  
any changes that are required for your infrastructure. All Terraform commands  
should now work.

If you ever set or change modules or backend configuration for Terraform,  
rerun this command to reinitialize your working directory. If you forget, other  
commands will detect it and remind you to do so if necessary.

```
ondrej@sika-macbookpro:~/Projects/terraform/terraform-do-kubernetes-example (master)$
```

```
ondrej@sika-macbookpro:~/Projects/terraform/terraform-do-kubernetes-example (master)$ tf apply -auto-approve
```

```
digitalocean_kubernetes_cluster.sikademo: Creating...
```

```
cloudflare_record.k8s_wildcard: Creating...
```

```
cloudflare_record.k8s_wildcard: Creation complete after 1s [id=58af9124b9ead1f032bb2901493b6f3f]
```

```
digitalocean_kubernetes_cluster.sikademo: Still creating... [10s elapsed]
```

```
digitalocean_kubernetes_cluster.sikademo: Still creating... [20s elapsed]
```

```
ondrej@sika-macbookpro:~/Projects/terraform/terraform-do-kubernetes-example (master)$ terraform output kubeconfig > kubeconfig
ondrej@sika-macbookpro:~/Projects/terraform/terraform-do-kubernetes-example (master)$ export KUBECONFIG=kubeconfig
ondrej@sika-macbookpro:~/Projects/terraform/terraform-do-kubernetes-example (master)$ kubectl cluster-info
Kubernetes master is running at https://fb2abe02-d0b1-4853-927d-971f10173bba.k8s.ondigitalocean.com
CoreDNS is running at https://fb2abe02-d0b1-4853-927d-971f10173bba.k8s.ondigitalocean.com/api/v1/namespaces/kube-system/services/kube-dns:pr
```

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

```
ondrej@sika-macbookpro:~/Projects/terraform/terraform-do-kubernetes-example (master)$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
sikademo-e5c6	Ready	<none>	11m	v1.15.5
sikademo-e5ca	Ready	<none>	11m	v1.15.5
sikademo-e5ct	Ready	<none>	12m	v1.15.5

```
ondrej@sika-macbookpro:~/Projects/terraform/terraform-do-kubernetes-example (master)$ █
```

---

# Kubernetes CLI - kubectl

`kubectl apply -f <file>`

`kubectl get -f <file>`

`kubectl get <resource>`

`kubectl describe -f <file>`

`kubectl delete -f <file>`

---

---

# Resources in Kubernetes

---

---

# Resources in Kubernetes

- Workload
    - Pods
    - Controllers - Deployments, StatefulSets, DaemonSets, Jobs, CronJobs
  - Service & Load Balancing
    - Services, Ingress
  - Storage
    - PersistentVolumes, PersistentVolumeClaims
  - Configuration
    - ConfigMaps, Secrets
  - RBAC
    - ServiceAccounts, Roles, RoleBindings
-

# Pod

---

- Minimal schedulable unit
- Contains one (or more) containers running in one IPC & network namespace
- Contains definition of Docker image, resource limits and other settings for containers
- Pods are not used directly, we use controllers like Deployments, ...

More: <https://kubernetes.io/docs/concepts/workloads/pods/pod/>

---

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat 01_pod.yml
```

File: 01\_pod.yml

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: simple-hello-world
5   labels:
6     app: simple-hello-world
7     svc: example
8 spec:
9   containers:
10    - name: simple-hello-world
11      image: ondrejsika/go-hello-world:2
12      ports:
13        - containerPort: 80
14      resources:
15        requests:
16          memory: 10Mi
17          cpu: 10m
18        limits:
19          memory: 20Mi
20          cpu: 20m
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl apply -f 01_pod.yml
pod/simple-hello-world created
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl get -f 01_pod.yml
NAME           READY   STATUS      RESTARTS   AGE
simple-hello-world  0/1   ContainerCreating  0          6s
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl get -f 01_pod.yml
NAME           READY   STATUS      RESTARTS   AGE
simple-hello-world  1/1   Running    0          14s
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl get po
NAME           READY   STATUS      RESTARTS   AGE
simple-hello-world  1/1   Running    0          20s
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl describe -f 01_pod.yml
Name:         simple-hello-world
Namespace:    default
Priority:    0
Node:        sikademo-e5c6/10.135.86.183
Start Time:  Tue, 12 Nov 2019 10:13:03 +0100
Labels:      app=simple-hello-world
              svc=example
Annotations: kubectl.kubernetes.io/last-applied-configuration:
              {"apiVersion":"v1","kind":"Pod","metadata":{"annotations":{},"labels":{"app":"simple-hello-world","svc":"example"},"name":"simple-hello-world","namespace":"default"}}
Status:      Running
IP:         10.244.2.165
IPs:        <none>
Containers:
  simple-hello-world:
    Container ID:  docker://2555c3884d7b8f58b1b7228d60309712b36f3fee020857678423d2e5bf86725d
    Image:        ondrejsika/go-hello-world:2
    Image ID:    docker-pullable://ondrejsika/go-hello-world@sha256:a7430a31465d86443b8c79497b08ec350d5cdc492e0fbfb4098928184bc8c29b
    Port:        80/TCP
    Host Port:   0/TCP
    State:       Running
    Started:    Tue, 12 Nov 2019 10:13:12 +0100
    -
```

# Deployment

---

- Used to maintain some specific Pods up and running in N instances
- Provide various deployment (upgrade) strategies
- Allow us to rollback deployment

More: <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

---

File: 04\_01\_deployment.yml

---

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: hello-world
5   labels:
6     app: hello-world-deploy
7 annotations:
8   kubernetes.io/change-cause: ver1
9 spec:
10   selector:
11     matchLabels:
12       app: hello-world-deploy
13   replicas: 2
14   template:
15     metadata:
16       labels:
17         app: hello-world-deploy
18         svc: example
19     spec:
20       containers:
21         - name: hello-world
22           image: ondrejsika/go-hello-world:1
23           ports:
24             - containerPort: 80
25           resources:
26             requests:
27               cpu: 10m
28               memory: 10Mi
29             limits:
30               memory: 20Mi
31               cpu: 20m
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl apply -f 04_01_deployment.yml
deployment.apps/hello-world created
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl get -f 04_01_deployment.yml
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
hello-world	2/2	2	2	17s

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl get deploy,rs,po
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.extensions/hello-world	2/2	2	2	29s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.extensions/hello-world-6b8f4bdb46	2	2	2	29s

NAME	READY	STATUS	RESTARTS	AGE
pod/hello-world-6b8f4bdb46-qzwtl	1/1	Running	0	29s
pod/hello-world-6b8f4bdb46-z6ls5	1/1	Running	0	29s

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl delete -f 04_01_deployment.yml
```

```
deployment.apps "hello-world" deleted
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

# StatefulSet

---

- StatefulSet is the workload API object used to manage stateful applications.
- Manages the deployment and scaling of a set of Pods, and provides guarantees about the ordering and uniqueness of these Pods.

More: <https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/>

---

# DaemonSet

---

- A DaemonSet ensures that all (or some) Nodes run a copy of a Pod.
- As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected.

Some typical uses of a DaemonSet are:

- running a cluster storage daemon, such as **glusterd**, **ceph**, on each node.
- running a logs collection daemon on every node, such as **fluentd** or **logstash**.

---

More: <https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/>

# Services

---

- ClusterIP
  - Internal service to connect resources inside Kubernetes cluster
- NodePort
  - Expose specific port on every node of cluster
  - Use ports from range 30000 - 32767
- LoadBalancer (cloud only)
  - Create new load balancer with new IP
  - Publish service on standard (defined) ports

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat 05_clusterip_service.yml
```

---

File: 05\_clusterip\_service.yml

---

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: hello-world-clusterip
5 spec:
6   type: ClusterIP
7   ports:
8     - port: 80
9       protocol: TCP
10      targetPort: 80
11 selector:
12   app: hello-world-deploy
13
```

---

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat 06_nodeport_service.yml
```

---

File: 06\_nodeport\_service.yml

---

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: hello-world-nodeport
5 spec:
6   type: NodePort
7   ports:
8     - port: 80
9       protocol: TCP
10      targetPort: 80
11      # nodePort is optional
12      nodePort: 31001
13   selector:
14     app: hello-world-deploy
15
```

---

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat loadbalancer.yml
```

---

File: `loadbalancer.yml`

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: hello-world-loadbalancer
5 spec:
6   type: LoadBalancer
7   ports:
8     - port: 80
9       protocol: TCP
10      targetPort: 80
11 selector:
12   app: hello-world-deploy
13
```

---

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl apply -f 05_clusterip_service.yml
service/hello-world-clusterip created
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl apply -f 06_nodeport_service.yml
service/hello-world-nodeport created
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl apply -f loadbalancer.yml
service/hello-world-loadbalancer created
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
hello-world-clusterip  ClusterIP  10.245.51.146 <none>       80/TCP        19s
hello-world-loadbalancer LoadBalancer 10.245.50.14  <pending>    80:30770/TCP  12s
hello-world-nodeport     NodePort    10.245.0.120  <none>       80:31001/TCP  17s
kubernetes          ClusterIP  10.245.0.1    <none>       443/TCP      40m
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
hello-world-clusterip  ClusterIP  10.245.51.146 <none>       80/TCP        6m25s
hello-world-loadbalancer LoadBalancer 10.245.50.14  138.68.114.73  80:30770/TCP  6m18s
hello-world-nodeport     NodePort    10.245.0.120  <none>       80:31001/TCP  6m23s
kubernetes          ClusterIP  10.245.0.1    <none>       443/TCP      47m
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl delete -f 05_clusterip_service.yml
service "hello-world-clusterip" deleted
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl delete -f 06_nodeport_service.yml
service "hello-world-nodeport" deleted
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl delete -f loadbalancer.yml
service "hello-world-loadbalancer" deleted
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

# Ingress

---

- Ingress allows you expose services on domains and web paths
- Easiest & cheapest way how to expose web services
- Requires Ingress Controllers
  - Traefik - <https://github.com/ondrejsika/kubernetes-ingress-traefik>
  - Nginx + Cert Manager

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl apply -f webservers.yml
```

```
deployment.apps/nginx created
```

```
service/nginx created
```

```
deployment.apps/apache created
```

```
service/apache created
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl apply -f 10_ingress.yml
```

```
ingress.extensions/webservers created
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl get deploy,svc,ingress
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.extensions/apache	1/1	1	1	17s
deployment.extensions/nginx	1/1	1	1	17s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/apache	ClusterIP	10.245.59.16	<none>	80/TCP	17s
service/kubernetes	ClusterIP	10.245.0.1	<none>	443/TCP	64m
service/nginx	ClusterIP	10.245.248.47	<none>	80/TCP	17s

NAME	HOSTS	ADDRESS	PORTS	AGE
ingress.extensions/webservers	nginx.k8s.sikademo.com,apache.k8s.sikademo.com	80	11s	

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ kubectl describe -f 10_ingress.yml
```

```
Name: webservers
```

```
Namespace: default
```

```
Address:
```

```
Default backend: default-http-backend:80 (<none>)
```

```
Rules:
```

Host	Path	Backends
-----	-----	-----
nginx.k8s.sikademo.com	/	nginx:80 (10.244.1.231:80)
apache.k8s.sikademo.com	/	apache:80 (10.244.2.54:80)

```
Annotations:
```

```
kubectl.kubernetes.io/last-applied-configuration: {"apiVersion":"extensions/v1beta1","kind":"Ingress","metadata":{"annotations":{},"name":"we
```

# Persistent Storage

---

- **EmptyDir**
  - Simplest persistent storage
  - Chained to specific Pod (persistent only for that specific pod)
  - Stored on node
- **PersistentVolume (PV)**
  - Storage which can be attached to pods
- **StorageClass (SC)**
  - Dynamic provisioner of Persistent Volumes
- **PersistentVolumeClaim (PVC)**
  - allow a user to consume abstract storage resources

More: <https://kubernetes.io/docs/concepts/storage/volumes/>

---

File: emptydir.yaml

---

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: emptydir
5 spec:
6   initContainers:
7     - name: init
8       image: debian
9       command:
10         - /bin/sh
11         - -c
12         - date > /data/index.html.tmp; mv /data/index.html.tmp /data/index.html;
13   volumeMounts:
14     - mountPath: /data
15       name: data
16   containers:
17     - name: nginx
18       image: nginx
19       resources:
20         requests:
21           memory: 10Mi
22           cpu: 10m
23         limits:
24           memory: 20Mi
25           cpu: 20m
26   volumeMounts:
27     - mountPath: /usr/share/nginx/html
28       name: data
29   volumes:
30     - name: data
31       emptyDir: {}
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat pv_nfs.yml
```

---

	File: <b>pv_nfs.yml</b>
--	-------------------------

```
1 apiVersion: v1
2 kind: PersistentVolume
3 metadata:
4   name: my-nfs
5 spec:
6   capacity:
7     storage: 5Gi
8   volumeMode: Filesystem
9   accessModes:
10    - ReadWriteOnce
11   persistentVolumeReclaimPolicy: Retain
12   # persistentVolumeReclaimPolicy: Recycle
13   # persistentVolumeReclaimPolicy: Delete
14   storageClassName: nfs
15   mountOptions:
16    - hard
17   nfs:
18     path: /nfs
19     server: nfs.sikademo.com
```

---

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat pvc_default.yml
```

	File: pvc_default.yml
--	-----------------------

1	apiVersion: v1
2	kind: PersistentVolumeClaim
3	metadata:
4	name: default-pvc
5	spec:
6	accessModes:
7	- ReadWriteOnce
8	resources:
9	requests:
10	storage: 1Gi

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

```
5 spec:
6   selector:
7     matchLabels:
8       app: nginx
9   replicas: 1
10  template:
11    metadata:
12      labels:
13        app: nginx
14    spec:
15      containers:
16        - name: nginx
17          image: nginx:alpine
18          resources:
19            requests:
20              cpu: 20m
21              memory: 30Mi
22            limits:
23              cpu: 50m
24              memory: 50Mi
25      ports:
26        - containerPort: 80
27      volumeMounts:
28        - mountPath: /usr/share/nginx/html
29          name: data
30      volumes:
31        - name: data
32      persistentVolumeClaim:
33        claimName: nginx-data
34
35  ---
36  apiVersion: v1
37  kind: Service
38  metadata:
39    .
```

---

# ConfigMap & Secret

- Store Configuration & Secrets for Pods & Kubernetes components

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat 11_secret.yml
```

File: 11\_secret.yml

```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: my-secret
5 data:
6   password: c2VjcmV0a2V5
7   token: c2VjcmV0dG9rZW4=
8 stringData:
9   tokenFromString: secrettokenfromstring
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat 12_config_map.yml
```

File: 12\_config\_map.yml

```
1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: my-config
5 data:
6   config: hello
7   hello.html: |
8     <h1>Hello from ConfigMap</h1>
9     <p>This is stored in ConfigMap</p>
```

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

---

# **RBAC (Role Based Access Control)**

- ServiceAccount
  - User in Kubernetes
- ClusterRole, Role
  - Define permissions in Kubernetes
- ClusterRoleBinding, RoleBinding
  - Assigns Role to ServiceAccount

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ bat 14_admin.yml
```

---

File: 14\_admin.yml

```
1 apiVersion: v1
2 kind: ServiceAccount
3 metadata:
4   name: admin-user
5   namespace: kube-system
6 ---
7 apiVersion: rbac.authorization.k8s.io/v1
8 kind: ClusterRoleBinding
9 metadata:
10   name: admin-user
11 roleRef:
12   apiGroup: rbac.authorization.k8s.io
13   kind: ClusterRole
14   name: cluster-admin
15 subjects:
16 - kind: ServiceAccount
17   name: admin-user
18   namespace: kube-system
```

---

```
ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)$ █
```

ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)\$ bat 15\_read.yml

File: 15\_read.yml

```
1 apiVersion: v1
2 kind: ServiceAccount
3 metadata:
4   name: read-user
5   namespace: kube-system
6 ---
7 kind: ClusterRole
8 apiVersion: rbac.authorization.k8s.io/v1
9 metadata:
10  name: pod-reader
11  namespace: kube-system
12 rules:
13 - apiGroups: []
14   resources: ["pods"]
15   verbs: ["get", "list"]
16 ---
17 apiVersion: rbac.authorization.k8s.io/v1
18 kind: ClusterRoleBinding
19 metadata:
20   name: read-user
21   namespace: kube-system
22 roleRef:
23   apiGroup: rbac.authorization.k8s.io
24   kind: ClusterRole
25   name: pod-reader
26 subjects:
27 - kind: ServiceAccount
28   name: read-user
29   namespace: kube-system
```

ondrej@sika-macbookpro:~/Projects/kubernetes-training (master)\$ bat 16\_namespace\_admin.yml

File: 16\_namespace\_admin.yml

```
1 apiVersion: v1
2 kind: Namespace
3 metadata:
4   name: devel
5 ---
6 apiVersion: v1
7 kind: ServiceAccount
8 metadata:
9   name: devel-user
10  namespace: devel
11 ---
12 kind: Role
13 apiVersion: rbac.authorization.k8s.io/v1
14 metadata:
15   name: devel-user-full-access
16   namespace: devel
17 rules:
18 - apiGroups: ["*"]
19   resources: ["*"]
20   verbs: ["*"]
21 ---
22 kind: RoleBinding
23 apiVersion: rbac.authorization.k8s.io/v1
24 metadata:
25   name: devel-user-role-binding
26   namespace: devel
27 subjects:
28 - kind: ServiceAccount
29   name: devel-user
30
```

---

# Helm

---

---

# Helm - Package manager for Kubernetes

```
helm repo add ondrejsika https://helm.oxs.cz
```

```
helm install demo ondrejsika/one-image --set host=demo.k8s.sikademo.com
```

NAME: demo

LAST DEPLOYED: Tue Nov 12 11:46:09 2019

NAMESPACE: default

STATUS: deployed

REVISION: 1

TEST SUITE: None

NOTES:

See: <https://demo.k8s.sikademo.com>

---

---

# Demo Time

---

---

# Summary

---

---

# Summary

- DevOps helps you with faster & reliable deployments
  - Docker helps you separate applications & unify your environment
  - Kubernetes remove concept of nodes and provide you one large pool of resources
  - Kubernetes deploy desired state
  - Docker & Kubernetes help you with microservice architecture
  - IaaS (Terraform) provide simple & reproducible infrastructure (even on private cloud)
-

---

# Alternatives

## Docker

- RKT
- Containerd

## Kubernetes

- Docker Swarm
  - OpenShift
-

# Resources

---

<https://aws.amazon.com/devops/what-is-devops/>

<https://dev.to/ashokisaac/devops-in-3-sentences-17c4>

<https://devopsish.com/what-is-devops/>

<https://www.davidbegin.com/using-terraform-docs-to-automate-keeping-your-terraform-modules-documenting/>

<https://12factor.net/>

<https://www.youtube.com/watch?v=uMA7qqXIXBk>

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

<https://www.howtoforge.com/core-components-of-a-kubernetes-cluster/#the-kubeapiserver>

---

[sika.link/javadays2019](https://sika.link/javadays2019)

---

# Thank you & Questions

Ondrej Sika

email: [ondrej@sika.io](mailto:ondrej@sika.io)

www: <https://ondrejsika.io>

twitter: [@ondrejsika](https://twitter.com/ondrejsika)

linkedin: [/in/ondrejsika/](https://www.linkedin.com/in/ondrejsika/)

Slides: <https://sika.link/javadays2019>

---