

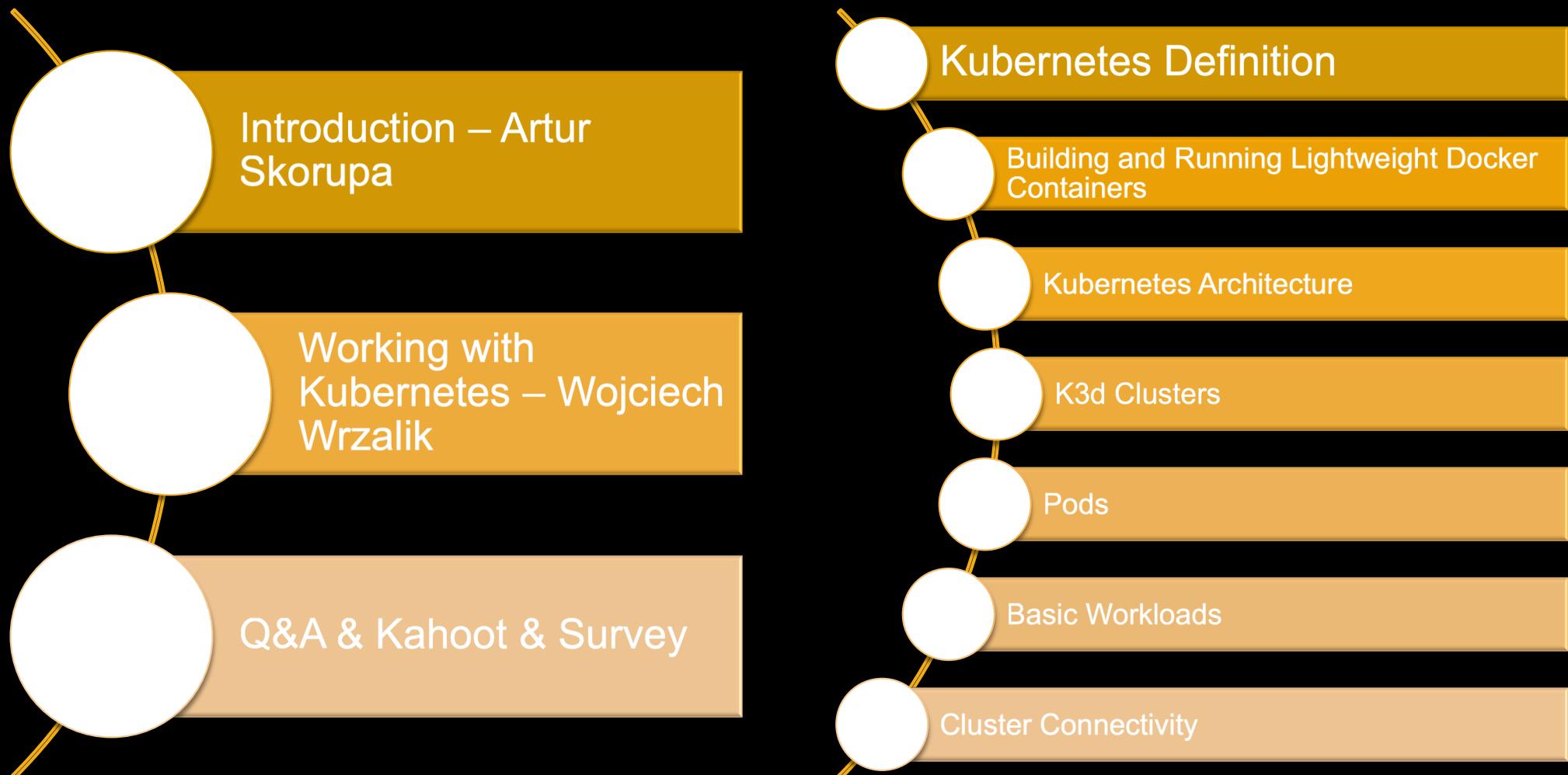
Working with Kubernetes

Wojciech Wrzalik, SAP

25 May, 2021

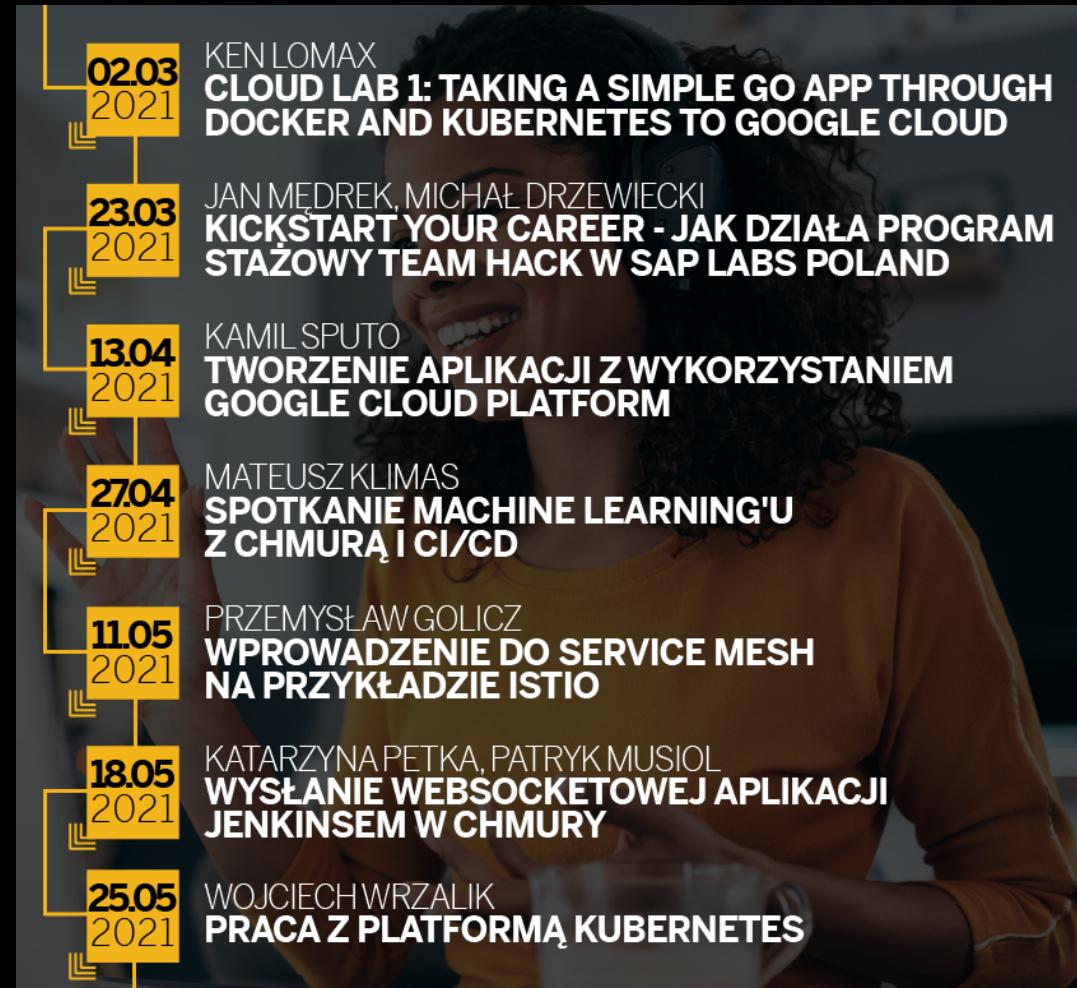


Agenda



Hack Your Career

HAYO
CKUR
CAREER



<https://hackyourcareer.github.io/>

<https://www.facebook.com/Hack.your.Career/>

SAP Labs Poland

Różnorodność domen

E-commerce, social commerce, cloud platform, open source,...

Development: Go, Java, Angular, Cloud Native solutions, ...

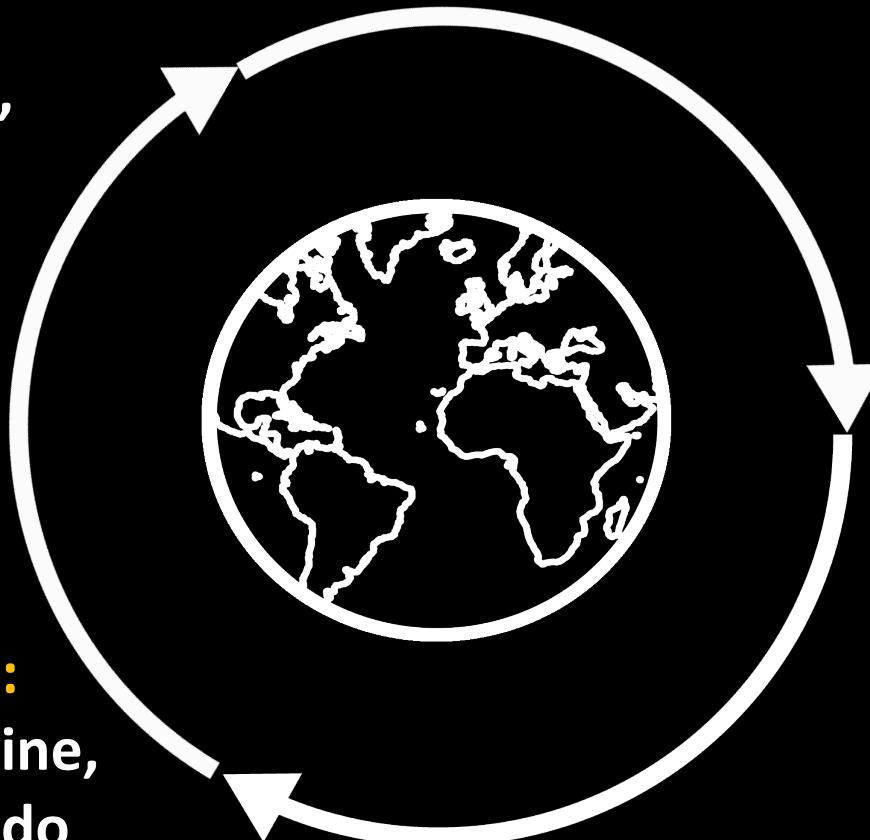
Możliwość nauki i rozwoju:

Platformy do edukacji on-line, certyfikacje, łatwy dostęp do ekspertów na całym świecie

> 350 pracowników

Różnorodność specjalizacji
Product teams, project teams, innovation projects teams, support teams

Jedno z 21 centrów SAP's Labs Network



Readme



Programmer with 8 years of experience.
Interested in **Java**, **Go** and **Kubernetes**.
Whenever possible enjoys **chess** and **good music**.

Goal

- Basic information about Kubernetes
- Useful for Hack Team
- Based on examples from
Kubernetes docs & YAML [Creative Commons Attribution 4.0 International](#)

Presumptions

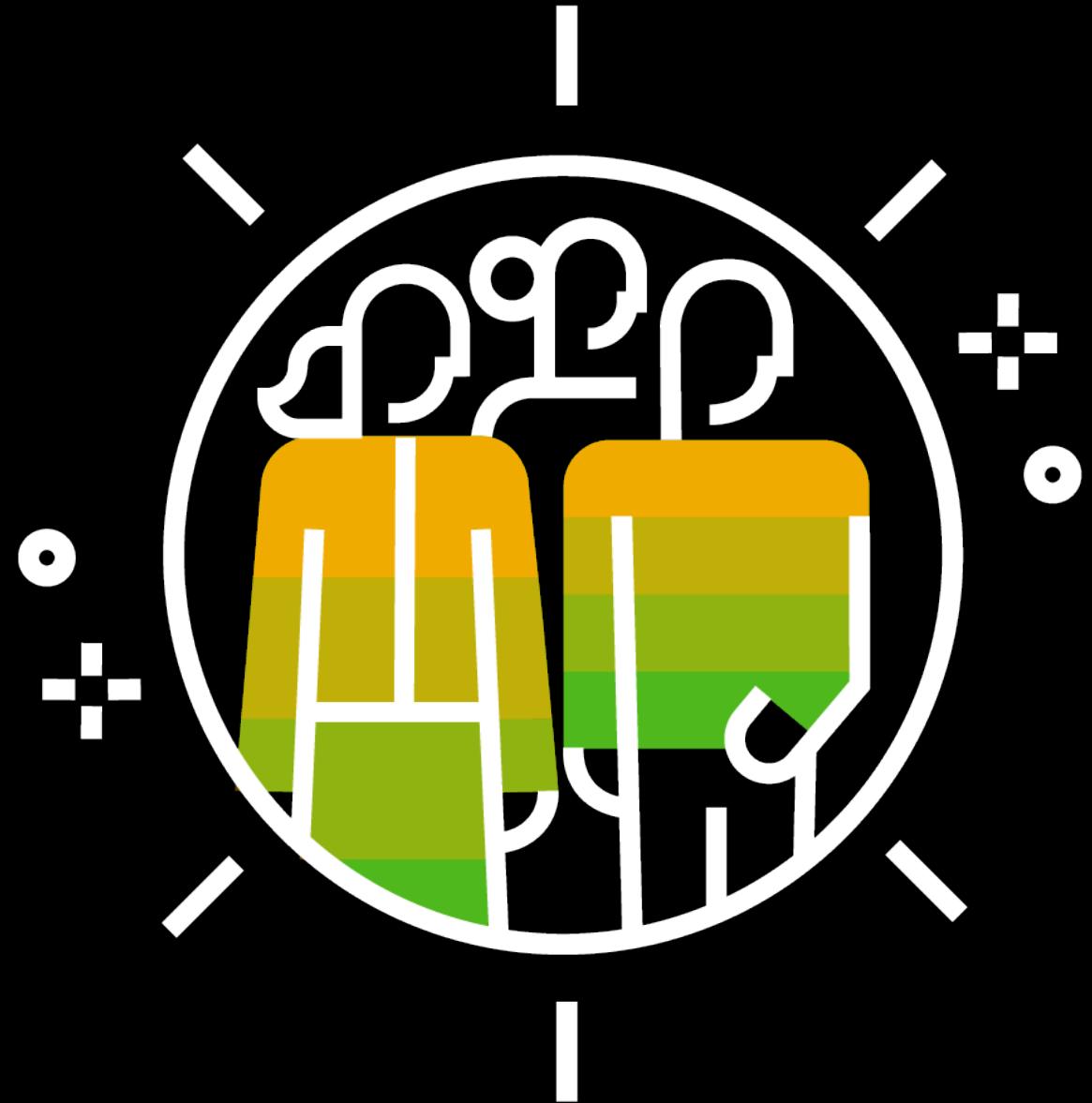
- kubectl installed
- Basic git knowledge
- Docker & k3d
- Bash

Git

```
→ ~ git clone https://github.com/rilikio/kubernetes-playground  
~/playground && cd playground
```

Team

- Programmers, DevOps and Administrators
- Simple iterations
- Minimum Viable Product (MVP) in Git
- Kubernetes



Requirements

- Simple clock application
- Web application suitable for cloud era



Cloud Era

- High Traffic
 - Fault Tollerance
 - Secure
-
- Distributed
 - Scalable

OS



Applications

Containers

- Ready-to-run packages
- Bound with OS and libraries
- Requires Container Runtime
- Containers != Containers != Virtualization



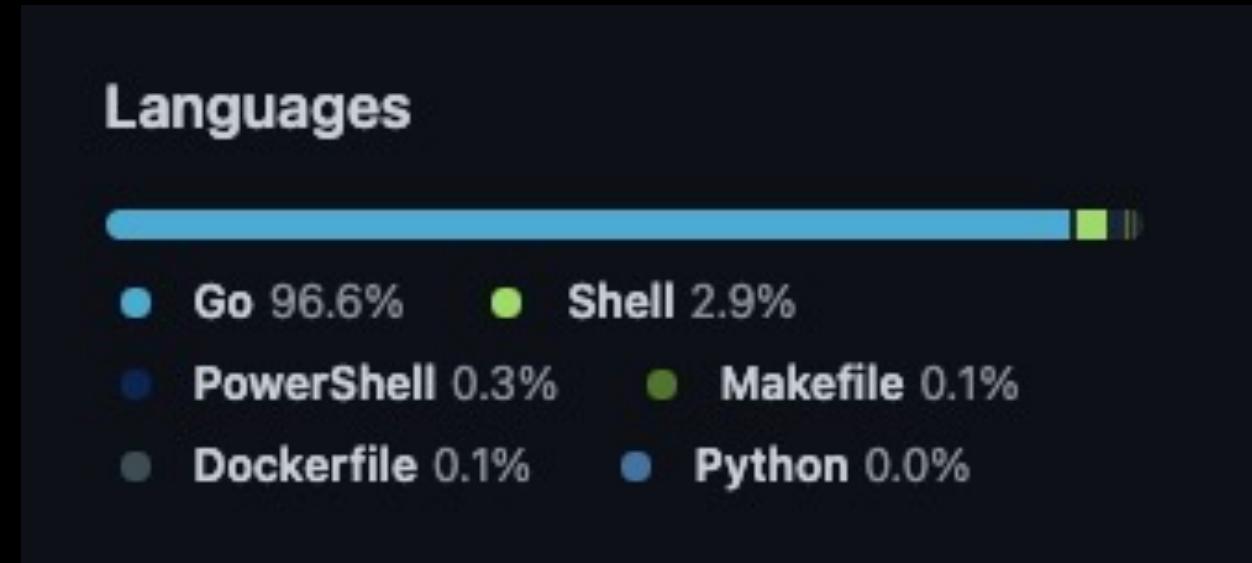
"Kubernetes is a **container** orchestration engine for automating deployment, scaling, and management of containerized application and workloads."

Iteration #1 – Docker Image



Tools

<https://github.com/kubernetes/kubernetes>



- git checkout
- go run app.go

<https://golang.org/LICENSE>

DevOps Review

- Does not have to know how to run a program
 - Kubernetes enforces containers
 - Unified testing

Dockerfile

```
FROM golang:latest
COPY app.go app.go
CMD go run app.go
```

→ git checkout

Building

→ docker build --help

Usage: docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile

Options:

-t, --tag list Name and optionally a tag in the 'name:tag' format

→ docker build -t ghcr.io/rilikio/images/app:latest .

Push

```
→ docker push --help
```

Usage: docker push [OPTIONS] NAME[:TAG]

Push an image or a repository to a registry

```
→ docker push ghcr.io/rilikio/images/app:latest
```

DevOps Review

→ docker images --help

Usage: docker images [OPTIONS] [REPOSITORY[:TAG]]

List images

Options:

--format string Pretty-print images using a Go template

→ docker images ghcr.io/rilikio/images/app:latest --format "{{.Repository}}:{{.Tag}}\t{{.Size}}"

DevOps Review

→ docker history --help

Usage: docker history [OPTIONS] IMAGE

Show the history of an image

→ docker history ghcr.io/rilikio/images/app:latest

DevOps Review

→ docker run --help

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

Run a command in a new container

Options:

--name string
-p, --publish list

Assign a name to the container
Publish a container's port(s) to the host

→ docker run --name server -p 8080:8080
ghcr.io/ralikio/images/app:latest

→ docker exec --help

Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container

Options:

-i, --interactive
--privileged
-t, --tty

Keep STDIN open even if not attached
Give extended privileges to the command
Allocate a pseudo-TTY

→ docker exec server ps aux

→ docker ps --help

Usage: docker ps [OPTIONS]

List containers

Options:

-a, --all

Show all containers (default shows just running)

DevOps Review

→ docker stop --help

Usage: docker stop [OPTIONS] CONTAINER [CONTAINER...]

Stop one or more running containers

→ docker rm --help

Usage: docker rm [OPTIONS] CONTAINER [CONTAINER...]

Remove one or more containers

Multistage Builds

```
FROM <IMAGE> as builder
<OTHER_COMMADS>

FROM <IMAGE>
COPY --from=builder <SRC> <DST> .
```

PID Changes

- Shell wrapping
- Exec form

```
ENTRYPOINT /app    CMD /app  
ENTRYPOINT ["/app"] CMD【"/app"]
```

Scratch Image

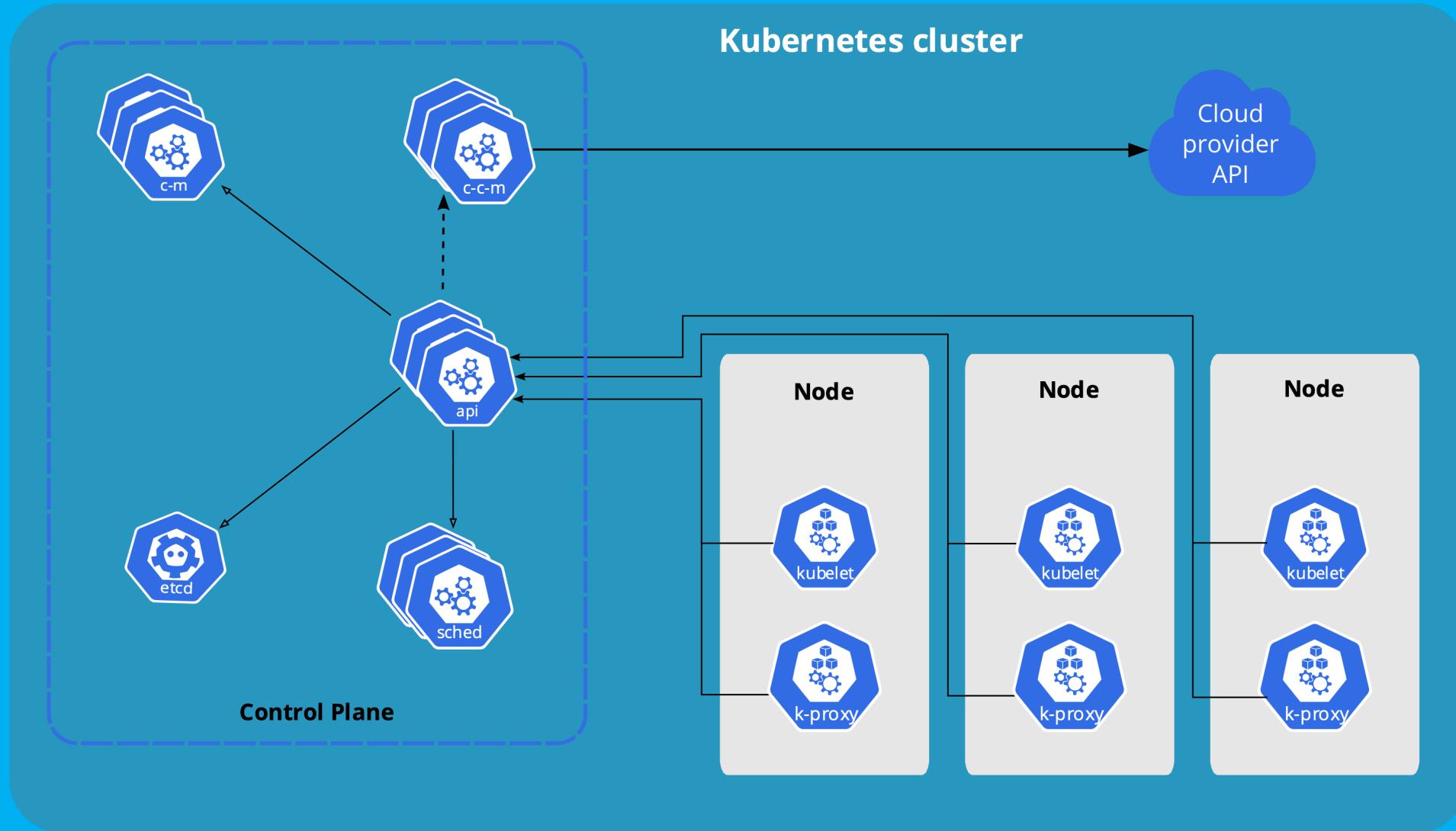
FROM ...as *builder*

...

FROM scratch

Verification & Review

- docker build -t ghcr.io/rilikio/images/app:latest .
- docker history ghcr.io/rilikio/images/app:latest
- docker images ghcr.io/rilikio/images/app:latest --format "{{.Repository}}:{{.Tag}}\t{{.Size}}"



* Image from <https://kubernetes.io/docs/concepts/overview/components/> - CC BY 4.0



Iteration #2 – Workloads in Kubernetes



Cluster Creation

→ `k3d cluster create --help`

Create a new k3s cluster with containerized nodes (k3s in docker).

Every cluster will consist of one or more containers:

- 1 (or more) server node container (k3s)
- (optionally) 1 loadbalancer container as the entrypoint to the cluster (nginx)
- (optionally) 1 (or more) agent node containers (k3s)

Usage:

`k3d cluster create NAME [flags]`

Flags:

`-a, --agents int` Specify how many agents you want to create
`-p, --port [HOST:][:HOSTPORT:]CONTAINERPORT[/PROTOCOL][@NODEFILTER]` Map ports from the node containers to the host
(Format: [HOST:][:HOSTPORT:]CONTAINERPORT[/PROTOCOL][@NODEFILTER])

→ `k3d cluster create dev -p "8082:30080@agent[0]" --agents 1`

→ `docker ps`

Kubeconfig

```
→ k3d kubeconfig get -a  
  
→ export KUBECONFIG=$(k3d kubeconfig write dev)"  
  
→ kubectl config view  
  
→ k config use-context k3d-k3s-default  
  
→ kubectl config set-context --current --  
  namespace=default
```

```
---  
apiVersion: v1  
clusters:  
- cluster:  
    certificate-authority-data: ...  
    server: https://0.0.0.0:62670  
    name: k3d-k3s-default  
contexts:  
- context:  
    cluster: k3d-k3s-default  
    user: admin@k3d-k3s-default  
    name: k3d-k3s-default  
current-context: k3d-k3s-default  
kind: Config  
preferences: {}  
users:  
- name: admin@k3d-k3s-default  
  user:  
    client-certificate-data: ...  
    client-key-data: ...
```

Running

- `kubectl run debug --image=radial/busyboxplus:curl -i --tty -- sh`
- `kubectl run app --image=ghcr.io/ralkio/images/app:latest`
- `kubectl logs -f app`
- `kubectl port-forward app 8080`

Exploring Pods

```
→ kubectl run app --  
image=ghcr.io/rilikio/images/app:latest --dry-  
run=client -o yaml >> pod.yaml
```

```
→ kubectl get pods
```

```
→ playground git:(main) git checkout
```



Workloads

- Deployments
- StatefulSet
- DaemonSet
- Job/CronJob

Iteration #3 – Using Workloads



Deployment Migration

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-deployment
spec:
  selector:
    matchLabels:
      app: server
  template:
    metadata:
      labels:
        app: server
  spec:
    containers:
      - image: ghcr.io/rilikio/images/app:latest
        name: app
        resources: {}
```

- `kubectl delete pod app`
- `git checkout`
- `kubectl apply -f deployment.yaml`
- `kubectl get deployments -o wide`

Port Exposure

```
apiVersion: v1
kind: Service
metadata:
  name: app-service
spec:
  type: NodePort
  selector:
    app: server
  ports:
    - protocol: TCP
      port: 8080
      nodePort: 30080
```

- `kubectl apply -f service.yaml`
- `kubectl get svc`
- `kubectl describe service app-service`
- `kubectl get pods --output=wide`
- `kubectl cluster-info`
- `kubectl get nodes --output wide`
- `kubectl run debug --image=radial/busyboxplus:curl -i --tty`

Iteration #3 – Cluster Connectivity



Direct Access to API Server

```
TOKEN=$(kubectl describe secret $SECRET_NAME | grep -E '^token' |  
cut -f2 -d':' | tr -d " ")  
  
APISERVER=$(kubectl config view --minify | grep server | cut  
-f 2- -d ":" | tr -d " ")  
  
curl $APISERVER/api/v1/pods --header "Authorization:  
Bearer $TOKEN" --insecure
```

RBAC Roles & Program Example

→ git checkout

Go and Dockerfile Mods

- `docker build --build-arg APP=ghcr.io/rilikio/images/client:latest -t client .`
- `kubectl apply -f job.yaml`

Aktualne oferty pracy:

- <https://jobs.sap.com/search/?locationsearch=gliwice>
[\(https://url.sap/g32xtt\)](https://url.sap/g32xtt)

Śledź nas na:

- <https://hackyourcareer.github.io/>
- <https://www.facebook.com/Hack.your.Career/>

Thank you.



Used Tech

- <https://kubernetes.io/> - <https://github.com/kubernetes/website/blob/master/LICENSE>,
<https://github.com/kubernetes/kubernetes/blob/master/LICENSE>
- <https://golang.org/> - <https://golang.org/LICENSEE>
- <https://docs.docker.com/> - <https://www.docker.com/legal>
- <https://k3d.io/> - <https://github.com/rancher/k3d/blob/main/LICENSE>