

Własny kubernetes dla śmiertelników

Co jest pomiędzy Minikube a EKS?

<https://horasoft.eu/slides/k8s-for-mortals/>



Michał Borkowski
wielki.borsuk@gmail.com

Niniejszy utwór ma charakter wyłącznie fikcyjny.

Wszelkie imiona, nazwiska, postacie oraz wydarzenia są
wytworem wyobraźni autora.

Jakiegolwiek podobieństwo do rzeczywistych osób,
żywych, zmarłych, a także tych, które mają się dopiero
narodzić, jest całkowicie przypadkowe.

Autor nie ponosi odpowiedzialności za fakt, że
czyjekolwiek życie może przypominać powyższą treść.

Nowy wsłspaniały świat

```
kubectl apply -f deployment.yaml
```

Tak po prostu?

```
helm install nginx
```



I one znowu działają?

```
apiVersion: v1
kind: Pod
metadata:
  name: invincible
spec:
  restartPolicy: Always
  containers:
  - name: run-whatever-happens
    image: busybox:1.27.2
    command:
    - 'sh'
    - '-c'
    - 'echo "restart" && sleep 1000'
```



Tyle ile trzeba?

```
kubectl autoscale deployment my-app \  
--cpu-percent=70 --min=2 --max=10
```



I podmienia się po kolei?

```
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  name: zk-pdb
spec:
  minAvailable: 50%
  selector:
    matchLabels:
      app: zookeeper
```



Gdzie jestem teraz?

-  Umiem pisać YAML-e
-  Kubectl apply działa
-  EKS/GKE "po prostu działa"
-  Co się dzieje pod maską?

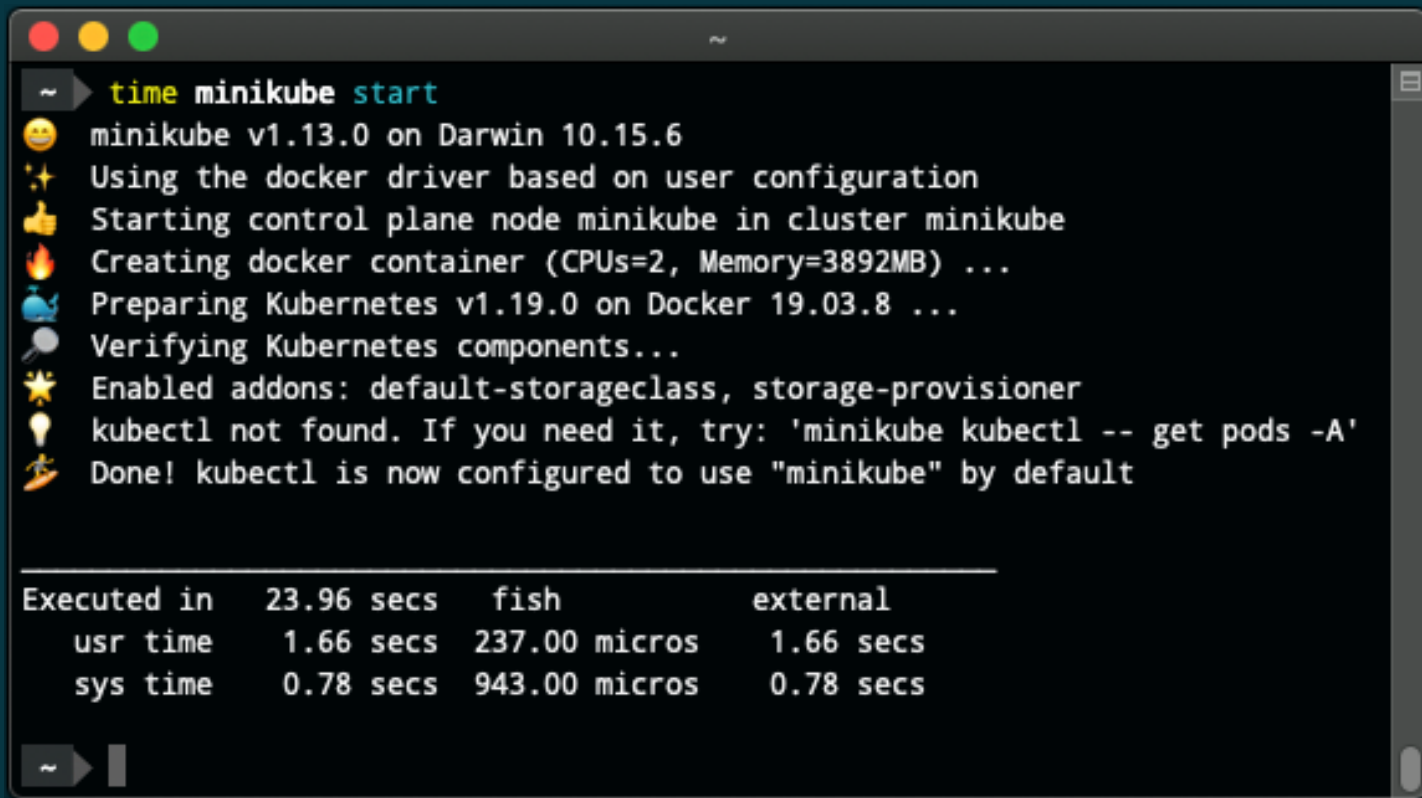
Gdzie chcę być?

-  Niezależny od chmury
-  Zarządzać własnym klastrem
-  Kontrolować połączenia sieciowe
-  Rozumieć architekturę

Minikube



Minikube jest super



```
~ ➤ time minikube start
🐶 minikube v1.13.0 on Darwin 10.15.6
🌟 Using the docker driver based on user configuration
👍 Starting control plane node minikube in cluster minikube
🔥 Creating docker container (CPUs=2, Memory=3892MB) ...
🐳 Preparing Kubernetes v1.19.0 on Docker 19.03.8 ...
🔍 Verifying Kubernetes components...
🌟 Enabled addons: default-storageclass, storage-provisioner
💡 kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
🚀 Done! kubectl is now configured to use "minikube" by default
```

Executed in	23.96 secs	fish	external
usr time	1.66 secs	237.00 micros	1.66 secs
sys time	0.78 secs	943.00 micros	0.78 secs

```
apiVersion: v1
kind: Config
current-context: minikube
clusters:
- cluster:
    certificate-authority: /home/mborkowski-local/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Wed, 14 Jan 2026 12:19:25 CET
        provider: minikube.sigs.k8s.io
        version: v1.27.1
        name: cluster_info
    server: https://192.168.49.2:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Wed, 14 Jan 2026 12:19:25 CET
        provider: minikube.sigs.k8s.io
```

Minikube - the good parts

- ✓ Nauka podstaw kubectl
- ✓ Testowanie manifestów
- ✓ Szybkie prototypy
- ✓ `minikube service <nginx> --url`
- ✓ PersistentVolume

Minikube - the ugly parts

- ❌ **Abstrakcja:** VM w VM w Dockerze
- ❌ **Sieć:** Brak dostępu z zewnątrz (ingress?)
- ❌ **Persistence:** Restart = strata danych
- ❌ **Realność:** To nie jest prawdziwy klaster (?)
- ❌ **Skala:** Jeden node = brak nauki HA

Czy ma pan/pani
chwilę, by
porozmawiać o
usługach Systemd





Kubeadm: Ściana płaczu

Kiedy ambicja spotyka się z brutalną rzeczywistością

NETWORKING

FLANNEL

CALICO

CILIUM

DEPLOYMENT

HELM

KUSTOMIZE

KSOPS

BACKUP
SOLUTION?

```
kubeadm init...  
Choose your addons...  
...
```

STORAGE PLUGIN?

CEPH

OPENEBS

LONGHORN

METRICS?

PROMETHEUS

GRAFANA

INGRESS?

NGINX

TRAFKIK

HAPROXY

CSI
DRIVER?

CNI
PLUGIN?



Niby wiesz co Cię czeka

- **Certyfikaty**: Rotacja co 1 rok
- **ETCD**: Backup, recovery, HA setup
- **CNI**: Calico vs Flannel vs Weave
- **Load Balancer**: MetalLB konfiguracja
- **Storage**: Persistent volumes setup

Prawdziwa procedura

```
sudo kubeadm init
```

```
[init] Using Kubernetes version: v1.35.0
```

```
[preflight] Running pre-flight checks
```

```
    [WARNING Swap]: swap is supported for cgroup v2 only. The kubelet must be properly  
    configured to use swap. Please refer to  
    https://kubernetes.io/docs/concepts/architecture/nodes/#swap-memory,  
    or disable swap on the node
```

Prawdziwa procedura

```
sudo swapoff -a
```

```
sudo kubeadm init
```

```
[init] Using Kubernetes version: v1.35.0
```

```
[preflight] Running pre-flight checks
```

```
[WARNING Service-kubelet]: kubelet service is not enabled, please run  
'systemctl enable kubelet.service'
```

Prawdziwa procedura

```
sudo swapoff -a  
sudo pacman -S kubelet && sudo systemctl enable kubelet && sudo systemctl start kubelet  
  
sudo kubeadm init
```

```
[init] Using Kubernetes version: v1.35.0  
[preflight] Running pre-flight checks  
[ERROR CRI]: could not connect to the container runtime: failed to create new CRI  
runtime service: validate service connection: validate CRI v1 runtime API  
for endpoint "unix:///var/run/containerd/containerd.sock": rpc error:  
code = Unavailable desc = connection error: desc = "transport: Error while dialing:  
dial unix /var/run/containerd/containerd.sock: connect: no such file or directory"
```

Prawdziwa procedura

```
sudo swapoff -a
sudo pacman -S kubelet && sudo systemctl enable kubelet && sudo systemctl start kubelet
sudo pacman -S containerd && sudo systemctl enable containerd && \
    sudo systemctl start containerd

sudo kubeadm init
```

```
[init] Using Kubernetes version: v1.35.0
```

Your Kubernetes control-plane has initialized successfully!

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 192.168.122.145:6443 --token x5y4s6.ms48d9u28km4albl \
    --discovery-token-ca-cert-hash sha256:0b9....
```

Prawdziwa procedura

```
kubectl apply -f \
  https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
```

```
[arch@archlinux ~]$ kubectl get pods -A
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
kube-flannel	kube-flannel-ds-pf464	0/1	Error	3 (33s ago)	61s
kube-system	coredns-7d764666f9-bwt2c	0/1	ContainerCreating	0	8m13s
kube-system	coredns-7d764666f9-fhg4t	0/1	ContainerCreating	0	8m13s

Prawdziwa procedura

```
sudo swapoff -a
sudo pacman -S kubelet && sudo systemctl enable kubelet && sudo systemctl start kubelet
sudo pacman -S containerd && sudo systemctl enable containerd && \
    sudo systemctl start containerd

sudo kubeadm init --pod-network-cidr 10.244.0.0/16

kubectl apply -f \
    https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
# sudo modprobe br_netfilter
```

A ingress i storage dopiero przed nami!



```
#!/bin/bash
#
# Common setup for all servers (Control Plane and Nodes)

set -euxo pipefail

# Kubernetes Variable Declaration
KUBERNETES_VERSION="v1.34"
CRICTL_VERSION="v1.34.0"
KUBERNETES_INSTALL_VERSION="1.34.0-1.1"

# Disable swap
sudo swapoff -a

# Keeps the swap off during reboot
(crontab -l 2>/dev/null; echo "@reboot /sbin/swapoff -a") | crontab - || true
sudo apt-get update -y

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
```

K3S: Droga środka

Bez niepotrzebnej złożoności, gotowy do użycia

```
curl -sfL https://get.k3s.io | sh -
```

Prawdziwy proces

```
curl -sfL https://get.k3s.io | sh -
```

```
# Na drugim nodzie:
```

```
curl -sfL https://get.k3s.io | \  
  K3S_URL=https://<server>:6443 \  
  K3S_TOKEN=<token> sh -
```



Co zyskujesz?

- **Jedna komenda:** Instalacja w 30 sekund
- **Mały footprint:** ~512MB RAM
- **Pełna funkcjonalność:** Certyfikowany K8s
- **Batteries included:** Traefik, ServiceLB
- **SQLite:** Zamiast ETCD (opcjonalnie)
- **Prawdziwe nody:** Multi-node clustering



Idealne dla:

- **Homelab:** Raspberry Pi, stare laptopy
- **Edge computing:** IoT, ograniczony dostęp
- **Nauka:** Zrozumienie architektury
- **Development:** testowanie lokalne
- **Production:** Małe/średnie środowiska

Anatomia uproszczenia



Storage Backend

Standardowy K8s:

`etcd:`

- Tworzony dla klastrów
- Raft consensus
- TLS certificates
- Wrażliwy na lagi

K3s:

`sqlite:`

- Jeden plik
- Zero konfiguracji
- Brak certyfikatów



Networking

Standardowy K8s:

`components (additional):`

- CNI plugin (Flannel)
- LBalancer (MetalLB)
- Ingress Controller
- DNS (CoreDNS)

K3s:

`built-in:`

- Flannel CNI
- ServiceLB (Klipper)
- Traefik Ingress
- CoreDNS



Binary Size

Standardowy K8s:

`# Komponenty:`

`kube-apiserver: ~120MB`
`kube-scheduler: ~50MB`
`kube-controller: ~110MB`
`kubelet: ~110MB`
`# Total: ~400MB+`

K3s:

`# Jeden plik:`

`k3s: ~60MB`
`# Wszystko w środku`

Nauka przez praktykę
nie przez ból

Twój plan na najbliższy weekend



Materiały dla cierpiących na bezsnność

- <https://docs.k3s.io/> - dokumentacja k3s
- <https://github.com/k3s-io/k3s> - repozytorium
- <https://github.com/awesome-selfhosted/awesome-selfhosted> - inspiracja do domowego klastra
- <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/> - instrukcja dla kubeadm
- <https://www.theodo.com/en-fr/blog/kubernetes-technologies-kubeadm-vs-minikube-kind-and-k3s> - zestawienie klastrów



<https://norasoft.eu/slides/k8s-for-mortals/>

wielki.borsuk@gmail.com

