

# Kubernetes – dla programistów i nie tylko

Podstawowe informacje o kursie

# Co musisz wiedzieć aby zacząć

- Podstawowa znajomość Docker-a (kurs nie zawiera wprowadzenia do Docker-a)
- Podstawowa znajomość YAML-a
- Środowisko do edycji kodu (darmowe Visual Studio Code lub może być np. Intellij IDEA)
- Komputer (najlepiej z dostępem administratora) aby zainstalować klaster kubernetesa

# Docker od podstaw

Sprawdź mój inny kurs na Udemy  
„Docker od podstaw – dla programistów i nie tylko”

[Dostępny tutaj](#)

# Licencja edukacyjna JetBrains (6 miesięcy)

Link rejestracyjny:

<https://www.jetbrains.com/store/redeem/>

Kod:

<dostępny-w-materiałach-na-udemy-po-zakupie-kursu>

Ponieważ jest to licencja edukacyjna i negocjowana z JetBrains dzięki czemu jest to pojedynczy kod na wiele licencji będę wdzięczny za użytek na własne potrzeby 😊 Jako produkt sugeruję wybrać IntelliJ IDEA Ultimate lub All products pack

# Kod źródłowy

Repozytorium z materiałami: <https://github.com/pnowy/kubernetes-course>

Komendy używane w kursie: <https://github.com/pnowy/kubernetes-course/blob/main/komendy.md>

# Dane kontaktowe

Kontakt bezpośrednio przez platformę

LinkedIn: <https://www.linkedin.com/in/przemeknowak/>

GitHub: <https://github.com/pnowy>

Email: [kurskubernetesa@przemeknowak.com](mailto:kurskubernetesa@przemeknowak.com)

WWW: <https://przemeknowak.com/>

# Kubernetes – dla programistów i nie tylko

Agenda - omówienie

# Linki

- <https://github.com/pnowy/kubernetes-course/blob/main/agenda.md>
- <https://github.com/pnowy/kubernetes-course/blob/main/komendy.md>
- <https://ossinsight.io/analyze/kubernetes/kubernetes>



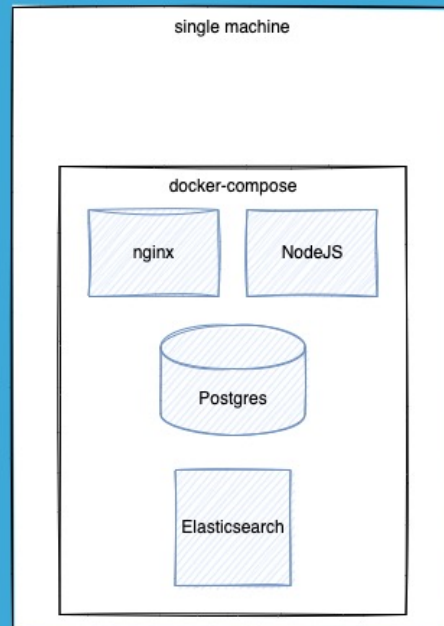
# Kubernetes – dla programistów i nie tylko

Wprowadzenie do orkiestracji kontenerów

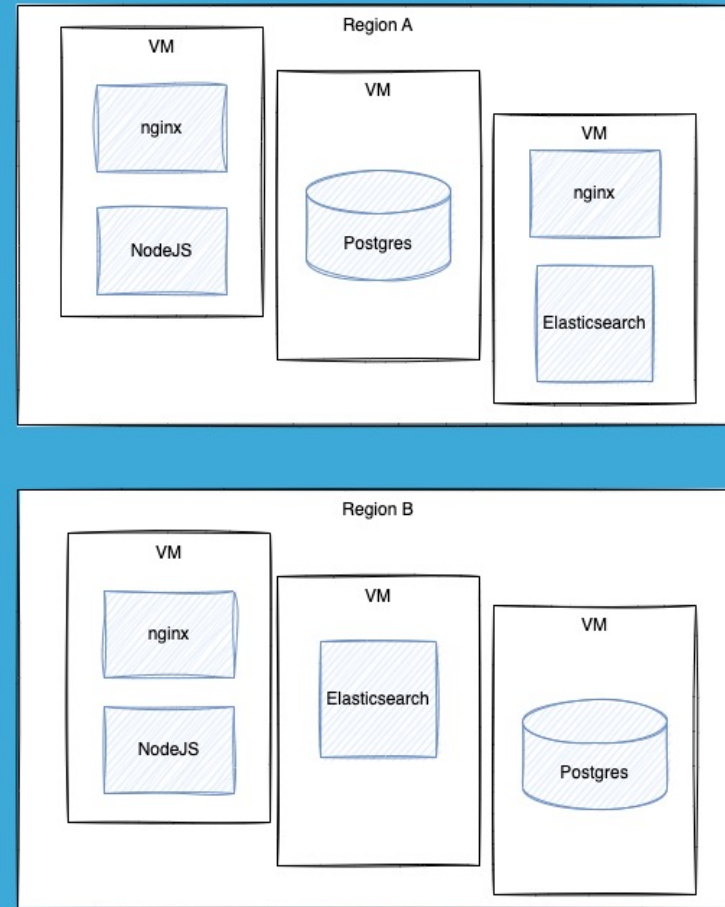
# Orkiestracja kontenerów

Proces automatyzacji, zarządzania oraz skalowania skonteneryzowanych aplikacji. Dotyczy (najczęściej) aplikacji o dużej skali.

Development



Production



# Orkiestracja kontenerów – obejmuje:

- Konfigurację i planowanie (scheduling) kontenerów
- Odpowiednie rozmieszczenie kontenerów (np. na odpowiednich maszynach)
- Zapewnienie dostępności kontenerów i monitorowanie ich stanu
- Konfigurację aplikacji uruchamianej na kontenerze
- Skalowanie kontenerów (np. w celu obsługi większego ruchu czy też równoważenia obciążenia w klastrze)
- Alokacja zasobów pomiędzy kontenerami
- Load balancing (zarówno dla grup kontenerów jak i całego klastra)
- Zabezpieczenie komunikacji pomiędzy kontenerami

	Kontenery	Orkiestracja kontenerów
Przeznaczenie	Zapewnienie separacji dla naszej aplikacji w postaci pojedynczej „paczki”	Zapewnienie i definicja relacji pomiędzy kontenerami, ich uruchamianiem, skalowaniem, połączeniem ze światem zewnętrznym
Alternatywy	<ul style="list-style-type: none"><li>• Wirtualne maszyny</li><li>• Bezpośrednia instalacja</li></ul>	<ul style="list-style-type: none"><li>• Własne skrypty</li><li>• Manualna konfiguracja</li></ul>
Narzędzia	<ul style="list-style-type: none"><li>• Docker</li><li>• Podman</li></ul>	<ul style="list-style-type: none"><li>• Kubernetes</li><li>• Docker Swarm</li><li>• Amazon ECS</li><li>• Mesos</li><li>• Rancher</li></ul>

# Kubernetes – dla programistów i nie tylko

## Kubernetes (k8s) - podstawowe informacje

# Infrastruktura - zmiany

Maszyny fizyczne → Wirtualizacja (2000)

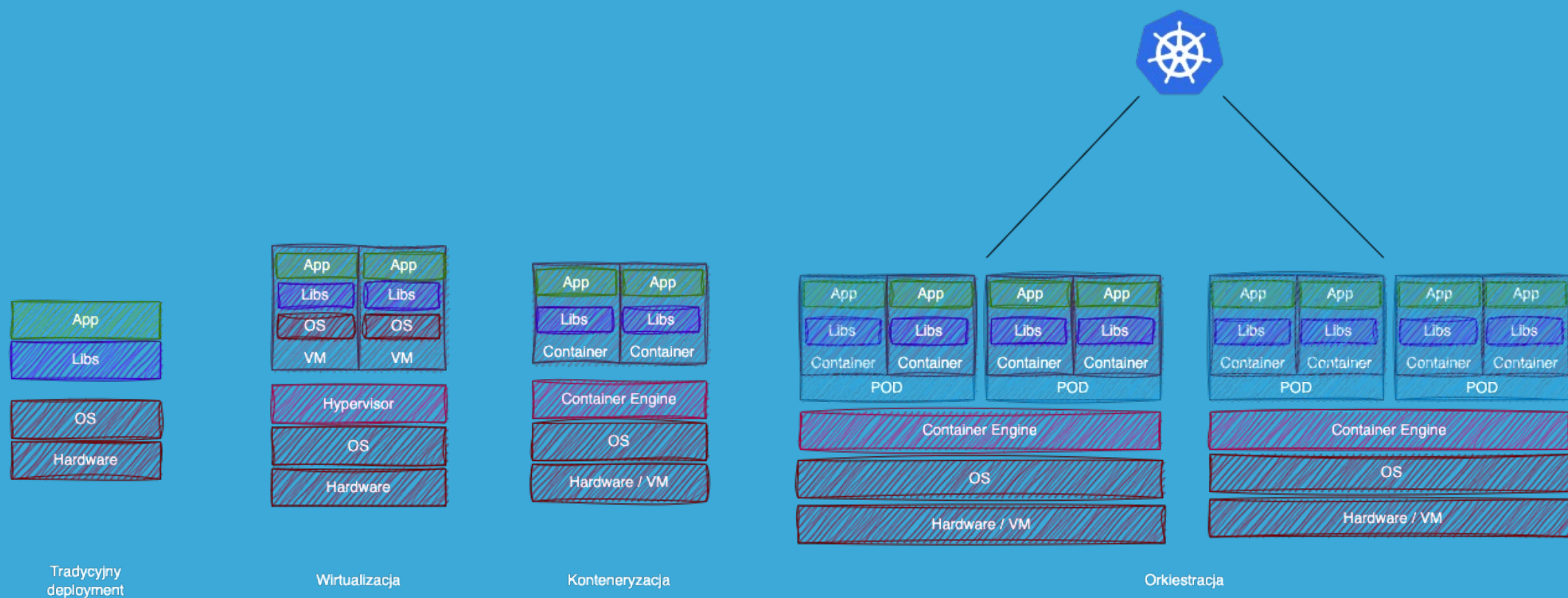


Datacenters → Cloud (2010)



Host → Containers (2014)

# Infrastruktura - zmiany





# Kubernetes - nazwa

Pochodzi z greckiego (κυβερνήτης *kubernētēs*)

Oznacza sternika / pilota

k8s  
kubernetes  
8



# Historia

- 2003 – 2004 Borg system (wewnętrzny projekt Google)
- 2014 Google przedstawia Kubernetes-a jako otwartą wersję Borg-a, w tym samym roku Microsoft, RedHat, IBM, Docker dołączają do projektu
- 2015 Kubernetes 1.0, utworzenie Cloud Native Computing Foundation (CNCF)
- 2016 Rozwój całego ekosystemu (Helm, Minikube, Kops)
- 2017+ Coraz większa popularność i migracja przedsiębiorstw do Kubernetes-a
- 2019+ "Eksplozja" popularności Kubernetesa

# Wady

- Rozwiązanie stosowane (zazwyczaj) w dużych organizacjach o określonym poziomie skomplikowania
- Wymagana znajomość konkretnych koncepcji (pod, deployment, etc.) przez co większe organizacje mogą mieć trudności z adaptacją
- Specyfikacja (manifesty) to YAML i aby sensownie nimi zarządzać często wymaga znajomości dodatkowych narzędzi (Helm, Kustomize)
- Administracja Kubernetesem jest skomplikowana

# Zalety

- Wprowadził standaryzację do systemów rozproszonych (deployment, zarządzanie konfiguracją, sieć, uprawnienia, monitoring)
- Open source! Duże i otwarte community.
- Przenośny i wspierany przez największych dostawców cloud-a (GCP, AWS, Azure)
- Rozszerzalny
- Praktycznie „standard” którego znajomość jest jeżeli nie wymagana to „nice to have” w dzisiejszych czasach

# Dlaczego warto się uczyć K8S

- Praktycznie „standard” którego znajomość jest jeżeli nie wymagana to „nice to have” w dzisiejszych czasach
- Znajomość koncepcji może być przydatna w przypadku konieczności debugowania aplikacji czy tej jej odpowiedniego przygotowania do deploymentu na Kubernetesie (zero—downtime deployment, self-healing)
- Coraz więcej firm, które używają k8s używa go także przy lokalnym developmencie (zamiast docker-compose, unifikacja manifestów, tunelowanie portów)

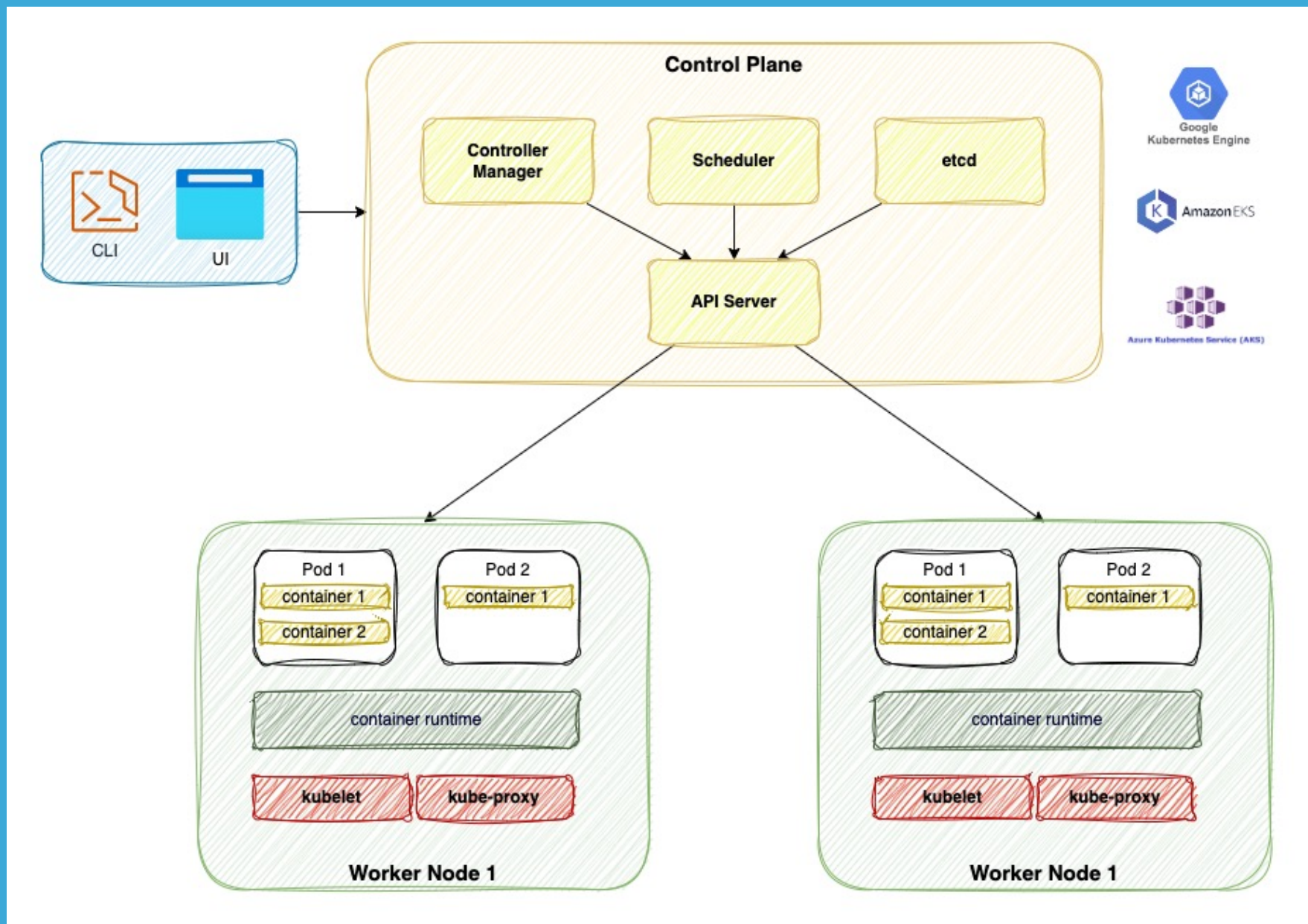
# Linki

- Borg
- Omega
- Kubernetes – pierwszy commit

# Kubernetes – dla programistów i nie tylko

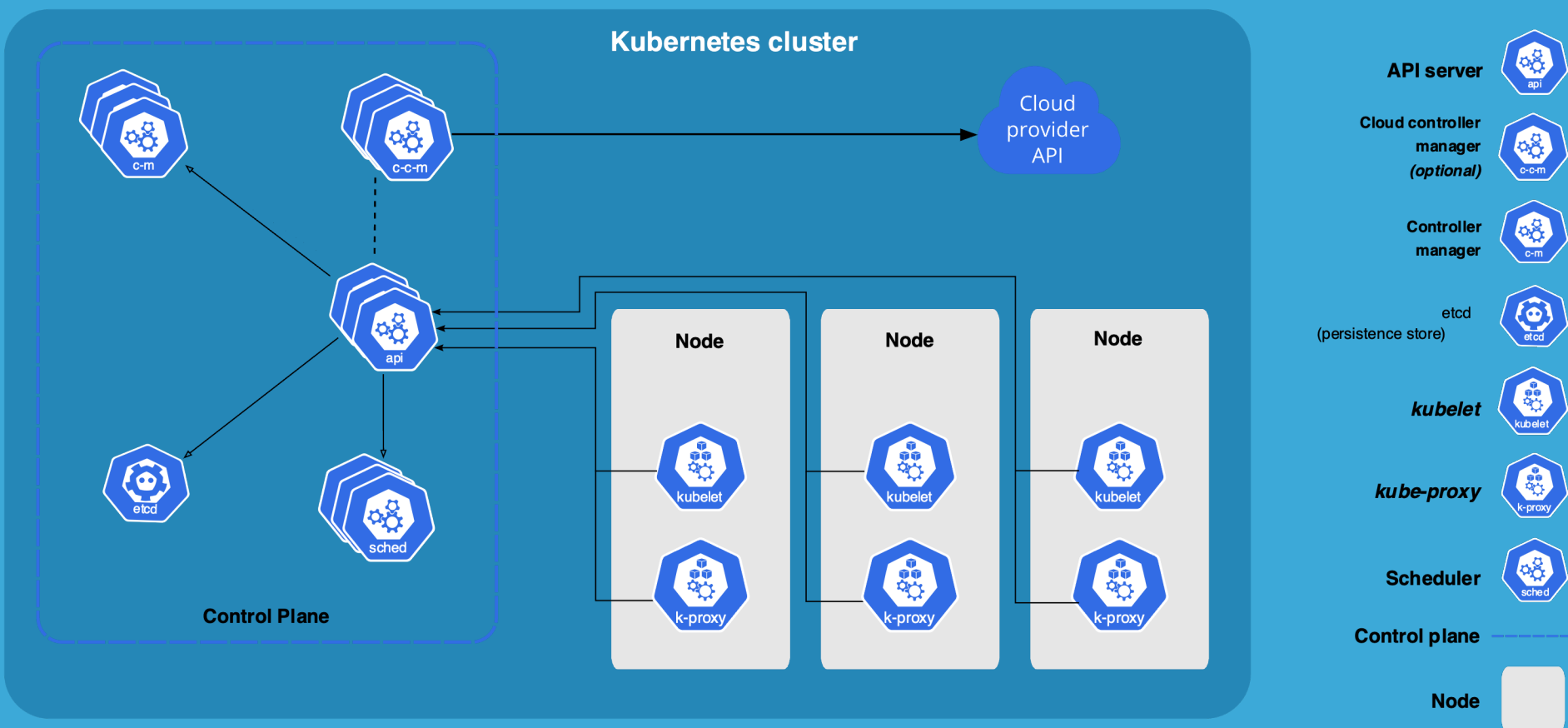
## Kubernetes - architektura

# Kubernetes - architektura





# Kubernetes - architektura



# Linki

- [Dokumentacja na oficjalnej stronie Kubernetes](#)

# Kubernetes – dla programistów i nie tylko

## Kubernetes – rodzaje instalacji

# Local development

- Minikube
- Docker Desktop
- Kind
- Rancher Desktop

# Cloud Providers

- GKE (Google Kubernetes Engine)
- EKS (Amazon Elastic Kubernetes Service)
- AKS (Azure Kubernetes Service)

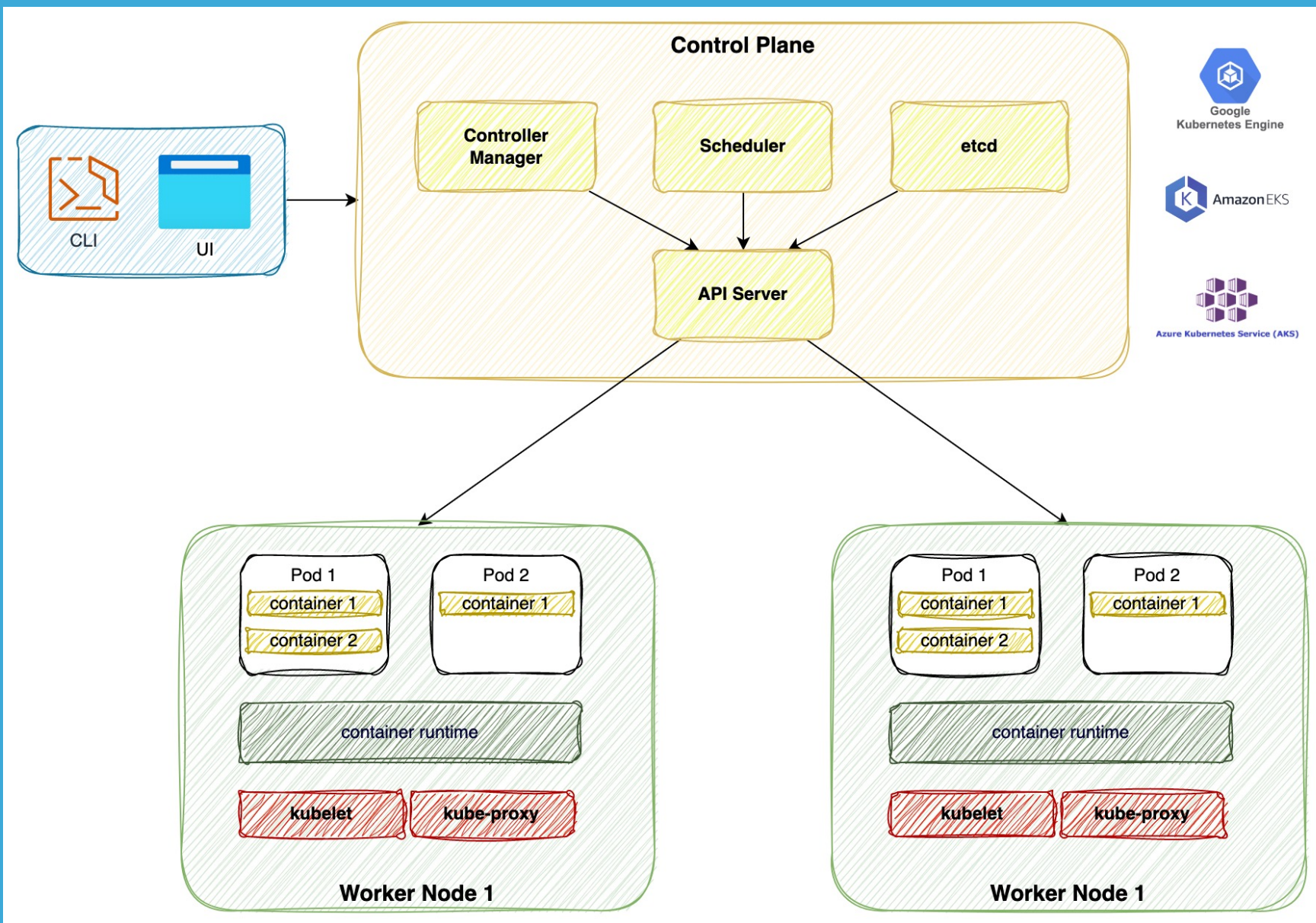
# Bare metal

- Kops
- K3S

# Kubernetes – dla programistów i nie tylko

## Kubectl – podstawowe informacje

# Kubectl





# Linki

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

# Kubernetes – dla programistów i nie tylko

## Minikube – instalacja Windows

# Linki

- [Minikube – dokumentacja](#)
- [Minikube - drivers](#)

# Kubernetes – dla programistów i nie tylko

## Minikube – instalacja Linux (Ubuntu 22)

# Linki

- [Minikube – dokumentacja](#)
- [Minikube - drivers](#)

# Kubernetes – dla programistów i nie tylko

## Minikube – instalacja MacOS

# Linki

- [Minikube – dokumentacja](#)
- [Minikube - drivers](#)

# Kubernetes – dla programistów i nie tylko

## Uruchamianie aplikacji (imperatywne – komendy)



# Kubernetes – dla programistów i nie tylko

Uruchamianie aplikacji (deklaratywnie - pliki)

# Kubernetes – dla programistów i nie tylko

Kubectl - alias

```
alias k="kubectl"
```

# Linki

<https://www.baeldung.com/linux/create-alias#2-permanent-alias> – jak ustawić alias w systemie Linux

<https://github.com/ahmetb/kubectl-aliases> - projekt z aliasami do Kuberneta

<https://ahmet.im/blog/kubectl-aliases/> - objaśnienie aliasów z projektu GitHub-a

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty - pod

# Linki

- <https://kubernetes.io/docs/concepts/workloads/pods/>
- <https://kubeyexample.com/learning-paths/istio/intro> - źródło diagramów z Istio

# Kubernetes – dla programistów i nie tylko

## Podstawowe koncepcje i obiekty – konfiguracja pod-ów

## Obrazy aplikacji dostępne na DockerHub

```
docker pull pnowy/kurs-kubernetesa-app:1.0.0
```

```
docker pull pnowy/kurs-kubernetesa-app:1.1.0
```

<https://hub.docker.com/r/pnowy/kurs-kubernetesa-app>



# Linki

- <https://hub.docker.com/r/pnowy/kurs-kubernetesa-app/tags>
- <https://kubernetes.io/docs/concepts/workloads/pods/>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – labels & selectors

# Linki

- <https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – annotations

# Linki

- <https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations/>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – deployments

# Linki

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

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – services



# Linki

- <https://kubernetes.io/docs/concepts/services-networking/service/>

# Kubernetes – dla programistów i nie tylko

## Podstawowe koncepcje i obiekty – narzędzia graficzne

# Linki

- <https://github.com/kubernetes/dashboard>
- <https://k9scli.io/>
- <https://k8slens.dev/>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – port-forward, proxy i  
dodatkowe komendy

# Linki

- <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#port-forward>
- <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#proxy>
- <https://github.com/johanhaleby/kubetail>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – kubectl explain

# Linki

- [https://jamesdefabia.github.io/docs/user-guide/kubectl/kubectl\\_explain/](https://jamesdefabia.github.io/docs/user-guide/kubectl/kubectl_explain/)

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – namespaces



# Linki

- <https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/>
- <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>
- <https://github.com/ahmetb/kubectx>
- <https://ohmyz.sh/>
- <https://github.com/romkatv/powerlevel10k>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – namespaces / DNS

# DNS

`[pod-ip].[namespace].pod.cluster.local`

`[svc-name].[namespace].svc.cluster.local`

`[port-name].[port-protocol].[svc].[namespace].svc.cluster.local`

# Linki

- <https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/>
- <https://kubernetes.io/docs/concepts/services-networking/dns-pod-service/>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – secrets

- Opaque
- Service account token secrets
- Docker config secrets
- Basic authentication secret
- SSH authentication secrets
- TLS secrets
- Bootstrap token secrets

# Linki

- <https://kubernetes.io/docs/concepts/configuration/secret/>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – secrets



- Opaque
- Service account token secrets
- Docker config secrets
- Basic authentication secret
- SSH authentication secrets
- TLS secrets
- Bootstrap token secrets

# Linki

- <https://kubernetes.io/docs/concepts/configuration/secret/>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – pod resources

# Linki

- <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>
- <https://kubernetes.io/docs/tasks/configure-pod-container/quality-service-pod/>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – probes

Readiness probe

Liveness probe

Startup probe

# Readiness probe

# Liveness probe



# Startup probe

# Handlers

ExecAction handler

TCPSocket Action handler

HTTPGetAction handler

gRPC handler

# Linki

- <https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/>
- <https://cloud.google.com/blog/products/containers-kubernetes/kubernetes-best-practices-terminating-with-grace>

# Kubernetes – dla programistów i nie tylko

Podstawowe koncepcje i obiekty – deployments konfiguracja

# Linki

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

# Kubernetes – dla programistów i nie tylko

## Zaawansowane koncepcje i obiekty – volumes

# Linki

- <https://kubernetes.io/docs/concepts/storage/persistent-volumes/>
- [https://minikube.sigs.k8s.io/docs/handbook/persistent\\_volumes/](https://minikube.sigs.k8s.io/docs/handbook/persistent_volumes/)

# Kubernetes – dla programistów i nie tylko

## Zaawansowane koncepcje i obiekty – storage classes



# Linki

- <https://kubernetes.io/docs/concepts/storage/storage-classes/>
- <https://kubernetes.io/docs/concepts/storage/persistent-volumes/>
- [https://minikube.sigs.k8s.io/docs/handbook/persistent\\_volumes/](https://minikube.sigs.k8s.io/docs/handbook/persistent_volumes/)

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
volumes – configmaps and secrets

# Linki

- <https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#populate-a-volume-with-data-stored-in-a-configmap>
- <https://kubernetes.io/docs/concepts/configuration/secret/#using-secrets-as-files-from-a-pod>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty – service types

# Linki

- <https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty – Ingress

# Linki

- <https://kubernetes.io/docs/concepts/services-networking/ingress/>
- <https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/>
- <https://github.com/stefanprodan/podinfo>
- <https://github.com/postmanlabs/httpbin>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty – Ingress TLS



# Linki

- <https://kubernetes.io/docs/concepts/services-networking/ingress/>
- <https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/>
- <https://github.com/FiloSottile/mkcert>

# Kubernetes – dla programistów i nie tylko

## Zaawansowane koncepcje i obiekty – init containers

# Główne cechy init containers

- Wykonują się do końca
- Uruchamiane sekwencyjnie
- Nie wspierają niektórych właściwości kontenerów aplikacyjnych

# Przykłady wykorzystania init containers

- Dodatkowe narzędzia dla kontenerów aplikacyjnych
- Weryfikacja stanu innych kontenerów
- Migracje (np. bazy danych), które w specyficznych przypadkach mogą trwać dłużej
- Pliki z config-mapy z możliwością edycji

# Linki

- <https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
StatefulSet - wprowadzenie

# Przykłady wykorzystania init containers

- Dodatkowe narzędzia dla kontenerów aplikacyjnych
- Weryfikacja stanu innych kontenerów
- Migracje (np. bazy danych), które w specyficznych przypadkach mogą trwać dłużej
- Pliki z config-mapy z możliwością edycji

# Linki

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



# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
headless service

# Linki

- <https://kubernetes.io/docs/concepts/services-networking/service/#headless-services>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
StatefulSet praktyka

# Linki

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

# Kubernetes – dla programistów i nie tylko

Zarządzanie manifestami  
Helm - wprowadzenie

# Linki

- <https://helm.sh/>

# Kubernetes – dla programistów i nie tylko

Zarządzanie manifestami  
Helm – instalacja aplikacji

# Linki

- <https://helm.sh/>
- <https://helm.sh/docs/intro/cheatsheet/>
- <https://artifacthub.io/packages/helm/podinfo/podinfo>



# Kubernetes – dla programistów i nie tylko

Zarządzanie manifestami  
Helm – ćwiczenie - parametryzacja

# Linki

- <https://helm.sh/>
- <https://helm.sh/docs/intro/cheatsheet/>
- <https://artifacthub.io/packages/helm/bitnami/wordpress>

# Kubernetes – dla programistów i nie tylko

Zarządzanie manifestami  
Helm – własny chart

# Linki

- <https://helm.sh/>
- <https://helm.sh/docs/intro/cheatsheet/>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
Job

# Linki

- <https://kubernetes.io/docs/concepts/workloads/controllers/job/>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
CronJob

# Linki

- <https://kubernetes.io/docs/tasks/job/automated-tasks-with-cron-jobs/>
- <https://en.wikipedia.org/wiki/Cron>
- <https://crontab.guru/>



# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
API Groups

# Linki

- <https://jamesdefabia.github.io/docs/api/>
- <https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.26/>
- <https://kubernetes.io/docs/reference/using-api/deprecation-guide/>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
Service accounts

# Linki

- <https://kubernetes.io/docs/reference/access-authn-authz/authentication/>
- <https://kubernetes.io/docs/concepts/security/service-accounts/>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
RBAC (Role-based access control)

# Linki

- <https://kubernetes.io/docs/reference/access-authn-authz/rbac/>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
DaemonSet

# Linki

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



# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
Alokacja podów na konkretnych nodach

# Linki

- <https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes/>
- <https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes-using-node-affinity/>
- <https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/>

# Kubernetes – dla programistów i nie tylko

Zaawansowane koncepcje i obiekty  
Alokacja podów na konkretnych nodach

# Linki

- <https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes/>
- <https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes-using-node-affinity/>
- <https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/>

# Kubernetes – dla programistów i nie tylko

Środowiska produkcyjne  
Google Kubernetes Engine

# Linki

- <https://cloud.google.com/free>
- <https://cloud.google.com/kubernetes-engine>

# Kubernetes – dla programistów i nie tylko

Środowiska produkcyjne

AWS – konfiguracja wstępna

# Linki

- <https://aws.amazon.com/free>
- [Instalacja AWS CLI](#)
- [AWS CLI MFA](#)
- <https://dodov.dev/blog/aws-cli-access-denied-despite-having-full-permissions>



# Kubernetes – dla programistów i nie tylko

Środowiska produkcyjne

Elastic Kubernetes Service

# Linki

- [eksctl](#)
- <https://eksctl.io/installation/#prerequisite>
- [eksctl przykłady](#)
- <https://aws.amazon.com/eks/>

# Kubernetes – dla programistów i nie tylko

Dobre praktyki

**Liveness / readiness probes**  
**Fault tolerance**  
**Resource management**  
**Labels and annotations**  
**Namespaces**  
**Logging and monitoring**  
**CI / CD (GitOps)**  
**Secrets**

# Linki

- <https://spring.io/blog/2020/03/25/liveness-and-readiness-probes-with-spring-boot>
- <https://cloud.google.com/blog/products/containers-kubernetes/kubernetes-best-practices-terminating-with-grace>
- <https://blog.colinbreck.com/kubernetes-liveness-and-readiness-probes-how-to-avoid-shooting-yourself-in-the-foot/>
- <https://kubernetes.io/docs/tasks/run-application/configure-pdb/>
- <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/>
- <https://kubernetes.io/docs/concepts/policy/resource-quotas/>
- <https://prometheus.io/>
- <https://grafana.com/>
- <https://argo-cd.readthedocs.io/en/stable/>
- <https://fluxcd.io/>
- <https://external-secrets.io/latest/introduction/overview/>
- <https://sealed-secrets.netlify.app/>

# Kubernetes – dla programistów i nie tylko

Co dalej i gdzie szukać dodatkowych informacji

Dzięki ;)

# Linki

- <https://kubernetes.io/docs/home/>
- <https://github.com/tomhuang12/awesome-k8s-resources>
- <https://github.com/collabnix/kubetools>
- <https://www.cncf.io/>
- <https://github.com/kelseyhightower/kubernetes-the-hard-way>