Dystrybucje Kubernetesa: kubeadm, k3s, RKE2, k0s, MicroK8s, Talos i OpenShift

Poziom: średnio

Dotyczy: Kubernetes 1.37.1 z kubeadm, k3s v1.37.1+k3s1, RKE2 v1.37.1+rke2r1, k0s v1.36.4+k0s.1, MicroK8s 1.36, Talos Linux 1.14.4, OpenShift 4.22 i OKD, stan na 18 października 2026

Czas: ~45 min

Kubernetes to projekt, a nie gotowy program do zainstalowania: do działającego klastra trzeba dobrać instalator, sieć, wejście z zewnątrz i sposób aktualizacji. Dystrybucje robią to za nas, każda inaczej. Ta strona porównuje siedem z nich, mówi, dla kogo jest każda, i pokazuje sprawdzoną w dokumentacji komendę instalacji na jednym węźle.

> Licencje i wsparcie podajemy wyłącznie jako informację, bez cen. Ceny subskrypcji zależą od umowy, liczby rdzeni i dnia, więc pytamy o nie producenta przed decyzją.

Skróty użyte na tej stronie (15)

Skrót	Co znaczy
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
CIS	Center for Internet Security: organizacja, która wydaje zalecenia utwardzenia systemów i Kubernetesa (CIS Benchmark), sprawdzane przez audytorów
CNI	Container Network Interface: wtyczka, która daje sieć podom w Kubernetesie
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
EPEL	Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów
etcd	etc distributed: rozproszony magazyn par klucz-wartość, w którym Kubernetes trzyma cały stan klastra, a zmiany uzgadnia protokołem Raft
FIPS	Federal Information Processing Standards: amerykańskie normy, z których FIPS 140 opisuje certyfikowane moduły kryptograficzne

Skrót	Co znaczy
HA	High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu
OKD	społecznościowa, bezpłatna wersja OpenShifta budowana z tego samego kodu, bez subskrypcji i wsparcia Red Hat
RHCOS	Red Hat Enterprise Linux CoreOS: niezmienny system operacyjny węzłów OpenShifta, aktualizowany razem z klastrem
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
RKE2	Rancher Kubernetes Engine 2: dystrybucja Kubernetesa od SUSE nastawiona na bezpieczeństwo, z profilem CIS i kryptografią zgodną z FIPS
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
SQLite	mała baza danych zapisana w jednym pliku; Home Assistant trzyma w niej historię w pliku home-assistant_v2.db
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera

Fiszka

Komendy instalacji pojedynczego węzła są przepisane z dokumentacji producentów (odczyt 14 do 18 października 2026), bez dodawania własnych opcji. Wykonujemy je na świeżej maszynie laboratoryjnej z Rocky Linux 9 jako użytkownik z uprawnieniami sudo, nigdy na `k3s01`, `docker01` ani innym serwerze z usługami. Talos i OpenShift Local nie instalują się na Rocky Linux: pierwszy jest osobnym systemem operacyjnym, drugi uruchamia maszynę wirtualną na stacji administratora. Pełne laboratoria HA z k3s i kubeadm opisują strony or2-03 i or2-04, a tu chodzi o pierwsze spotkanie z każdą dystrybucją.

◫◫ ZMIENIA SYSTEM

Inicjalizacja jednowęzłowego klastra kubeadm: kubeadm init

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

> (gdzie `10.244.0.0/16` to zakres adresów podów, który musi zgadzać się z ustawieniem wtyczki sieci CNI; ta wartość jest domyślna dla Flannel)

Przed tą komendą na maszynie musi działać środowisko uruchomieniowe kontenerów (na przykład containerd) oraz pakiety kubelet, kubeadm i kubectl, zainstalowane według dokumentacji Kubernetesa. Po `kubeadm init` węzeł zostaje `NotReady`, dopóki nie

zainstalujemy wtyczki CNI.

CNI = Container Network Interface: wtyczka, która daje sieć podom w Kubernetesie · **kubeadm** = Kubernetes admin: narzędzie, które inicjuje płaszczyznę sterowania, dołącza węzły i aktualizuje czysty klaster Kubernetesa

ZMIENIA SYSTEM

Kopia pliku administratora dla kubectl: admin.conf

```
mkdir -p $HOME/.kube && sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config && sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Trzy kroki z dokumentacji kubeadm w jednej linii: katalog, kopia pliku, zmiana właściciela. Od tej chwili `kubectl` działa bez sudo.

mkdir = make directory: utwórz katalog

ZMIENIA SYSTEM

Dopuszczenie zwykłych podów na jedynym węźle: kubectl taint

```
kubectl taint nodes --all node-role.kubernetes.io/control-plane-
```

Myślnik na końcu usuwa skazę (taint), która trzyma zwykłe pody z dala od płaszczyzny sterowania. Robimy to tylko w klastrze jednowęzłowym; w laboratorium HA pody idą na `kwk01` i `kwk02`.

HA = High Availability: wysoka dostępność, czyli działanie mimo awarii jednego elementu · **kubectl** = Kubernetes control: główne narzędzie Kubernetesa

TYLKO ODCZYT

Instalacja k3s na jednym węźle: get.k3s.io

```
curl -sfL https://get.k3s.io | INSTALL_K3S_VERSION='v1.37.1+k3s1' sh -
```

Instaluje k3s jako usługę systemd z bazą SQLite, Flannelem, Traefikiem, serwerem metryk i provisionerem lokalnych woluminów. Plik administratora leży w `/etc/rancher/k3s/k3s.yaml`, a token do dołączania węzłów w `/var/lib/rancher/k3s/server/node-token` [1].

Przykład:

```
[INFO]  Using v1.37.1+k3s1 as release
[INFO]  systemd: Starting k3s
```

SQLite = mała baza danych zapisana w jednym pliku; Home Assistant trzyma w niej historię w pliku home-assistant_v2.db · **curl** = client URL: pobieranie i wysyłanie przez HTTP z terminala

ZMIENIA SYSTEM

Instalacja RKE2 na jednym węźle: get.rke2.io

```
curl -sfL https://get.rke2.io | sudo sh -
```

Instaluje usługę `rke2-server` i pliki binarne, ale jej nie uruchamia [2]. Na Rocky Linux z włączonym NetworkManagerem dokumentacja każe najpierw ustawić go tak, żeby pomijał interfejsy tworzone przez CNI (opis w sekcji Known Issues dokumentacji RKE2).

RKE2 = Rancher Kubernetes Engine 2: dystrybucja Kubernetesa od SUSE nastawiona na bezpieczeństwo, z profilem CIS i kryptografią zgodną z FIPS · **CNI** = Container Network Interface: wtyczka, która daje sieć podom w Kubernetesie · **curl** = client URL: pobieranie i wysyłanie przez HTTP z terminala

ZMIENIA SYSTEM

Uruchomienie usługi RKE2: systemctl enable --now rke2-server

```
sudo systemctl enable --now rke2-server.service
```

Pierwszy start trwa kilka minut, postęp widać w `journalctl -u rke2-server -f`. Plik administratora to `/etc/rancher/rke2/rke2.yaml`, a `kubectl` leży w `/var/lib/rancher/rke2/bin/` i nie jest w zmiennej `PATH` [2].

RKE2 = Rancher Kubernetes Engine 2: dystrybucja Kubernetesa od SUSE nastawiona na bezpieczeństwo, z profilem CIS i kryptografią zgodną z FIPS · **systemctl** = system control: sterowanie usługami systemd

ZMIENIA SYSTEM

Pobranie k0s: get.k0s.sh

```
curl --proto '=https' --tlsv1.2 -sSf https://get.k0s.sh | sudo sh
```

Pobiera jeden plik binarny do `/usr/local/bin/k0s` [3].

curl = client URL: pobieranie i wysyłanie przez HTTP z terminala

ZMIENIA SYSTEM

Instalacja k0s jako jednowęzłowego klastra: k0s install controller --single

```
sudo k0s install controller --single --start
```

Instaluje usługę systemd z płaszczyzną sterowania i węzłem roboczym na jednej maszynie i od razu ją uruchamia. Opcja `--single` zamyka drogę do dołożenia kolejnych węzłów; gdy chcemy ją zostawić otwartą, dokumentacja podaje `--enable-worker --no-taints` zamiast `--single` [3].

k0s = polecenie k0s: instaluje i uruchamia klaster k0s jako usługę systemd, ma wbudowany kubectl

TYLKO ODCZYT

Sprawdzenie węzła k0s: k0s kubectl get nodes

```
sudo k0s kubectl get nodes
```

k0s ma wbudowany `kubectl`, więc nie trzeba niczego doinstalowywać. Węzeł bywa gotowy dopiero po minucie lub dwóch.

k0s = polecenie k0s: instaluje i uruchamia klaster k0s jako usługę systemd, ma wbudowany kubectl

ZMIENIA SYSTEM

Przygotowanie snapd na Rocky Linux pod MicroK8s: dnf install snapd

```
# Komendy można wykonać całościowo jako blok tekstu. Potem wylogowanie i ponowne
logowanie.
sudo dnf install -y epel-release
sudo dnf install -y snapd
sudo systemctl enable --now snapd.socket
sudo ln -s /var/lib/snapd/snap /snap
```

MicroK8s jest dystrybuowany jako snap, a Rocky Linux nie ma snapd w podstawowym repozytorium, więc bierzemy go z EPEL [4]. Instrukcja Snapcrafta jest napisana dla Rocky Linux 8, więc na Rocky 9 sprawdzamy ją w laboratorium.

EPEL = Extra Packages for Enterprise Linux: dodatkowe repozytorium pakietów dla RHEL i klonów · **dnf** = Dandified YUM: menedżer pakietów RHEL/Rocky · **systemctl** = system control: sterowanie usługami systemd · **ln** = link: utwórz dowiązanie do pliku, twarde albo symboliczne z flagą -s

ZMIENIA SYSTEM

Instalacja MicroK8s: snap install microk8s

```
sudo snap install microk8s --classic --channel=1.36
```

> (gdzie `1.36` to kanał z wersją Kubernetesa; strona „Get started” w dniu odczytu pokazywała kanał 1.35, a najnowsze noty wydań opisują MicroK8s 1.36)

Potem dokładamy użytkownika do grupy `microk8s` (`sudo usermod -a -G microk8s $USER`), logujemy się ponownie i czekamy na gotowość komendą `microk8s status --wait-ready` [5].

snap = menedżer pakietów snap od Canonical, którym instalujemy MicroK8s

ZMIENIA SYSTEM

Klaster Talos do prób w Dockerze na stacji administratora: talosctl cluster create docker

```
talosctl cluster create docker
```

Tworzy klaster Talos w kontenerach Dockera na lokalnej maszynie, wyłącznie do nauki [6]. Prawdziwy węzeł Talos to maszyna wirtualna uruchomiona z obrazu Talos i skonfigurowana przez `talosctl`; tę procedurę bierzemy z dokumentacji instalacji na maszynach wirtualnych, bo strona szybkiego startu jej nie opisuje. Klaster usuwa `talosctl cluster destroy`.

talosctl = Talos control: jedyne narzędzie do zarządzania węzłami Talos Linux przez ich API, bo węzły nie mają SSH

TYLKO ODCZYT

Przygotowanie OpenShift Local na stacji administratora: crc setup

```
crc setup
```

OpenShift Local (dawniej CodeReady Containers) to jednowęzłowy OpenShift w maszynie wirtualnej na komputerze administratora. Komenda sprawdza i przygotowuje środowisko, a do startu potrzebny jest plik pull secret z konta Red Hat.

□□ TYLKO ODCZYT

Uruchomienie OpenShift Local: crc start

```
crc start
```

Dla ustawienia `openshift` dokumentacja podaje domyślnie 4 procesory wirtualne i 10752 MiB pamięci jako minimum [7]. Jednowęzłowy OKD (bez subskrypcji) instalujemy innym instalatorem (`openshift-install`), według dokumentacji OKD, i tej procedury tu nie przepisujemy.

□□ **OKD** = społecznościowa, bezpłatna wersja OpenShifta budowana z tego samego kodu, bez subskrypcji i wsparcia Red Hat

Z życia wzięte

□□ Z ŻYCIA WZIĘTE

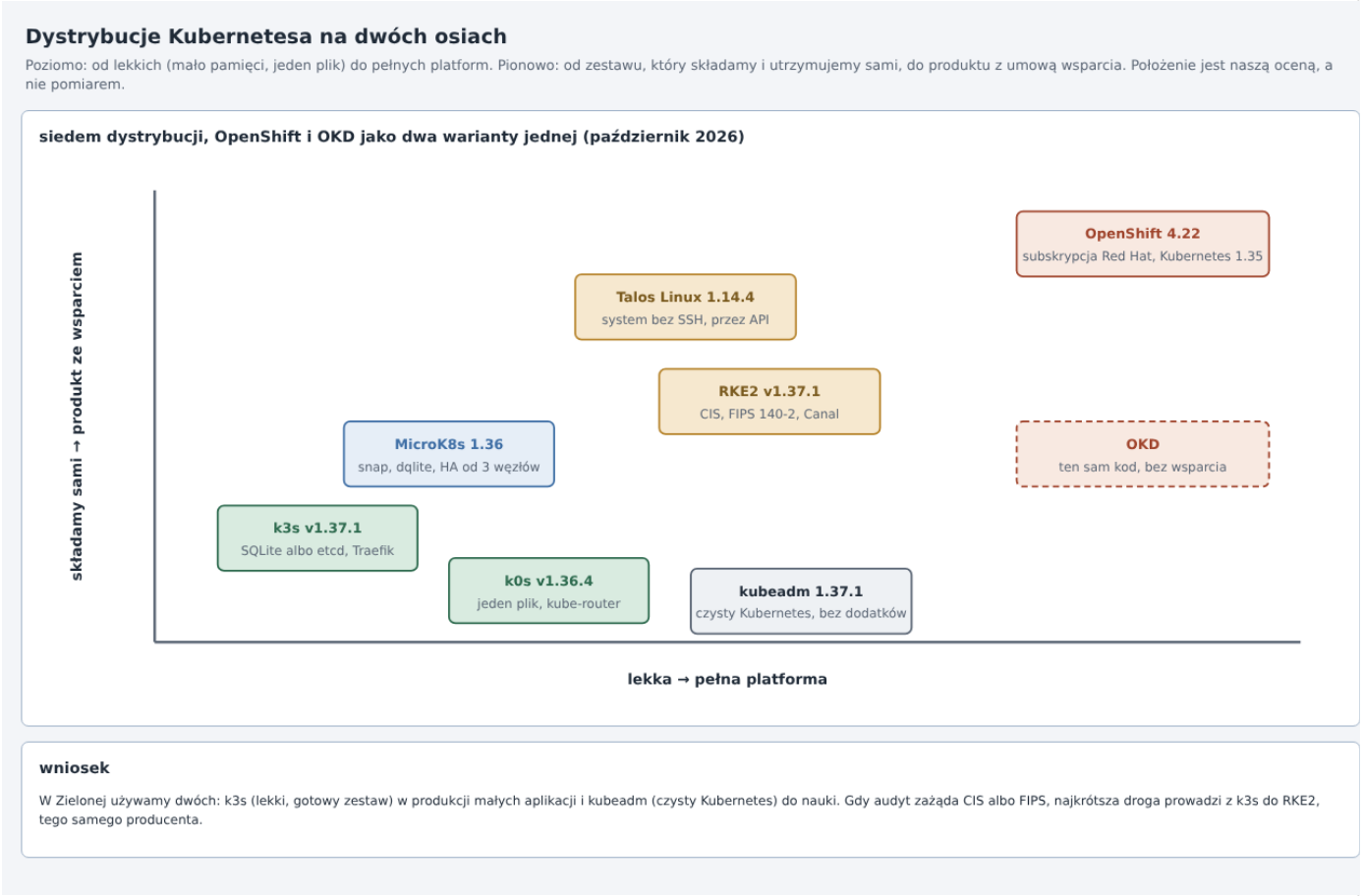
Czwartek, 15:20. Tomek wraca z konferencji z czterema naklejkami i pytaniem, czemu na `k3s01` mamy „okrojonego Kubernetesa”, skoro jest RKE2 z certyfikatami FIPS i Talos bez SSH. Ola odpowiada pytaniem: ile z tych funkcji sprawdzi u nas pan Wojtek przy audycie i kto będzie aktualizował trzy różne dystrybucje. Po godzinie z tabelą niżej zostają na stole dwie: k3s do produkcji małych aplikacji i kubeadm do nauki, a reszta trafia do notatki „gdyby audyt zażądał”.

Jak to działa

□□ Rozwiń: jak to działa pod spodem

Co właściwie dokłada dystrybucja

Czysty Kubernetes to kilka programów: serwer API, planista, menedżer kontrolerów, kubelet na każdym węźle i baza etcd. Żeby klaster przyjął pierwszą aplikację z zewnątrz, trzeba jeszcze dołożyć sieć podów (CNI), DNS w klastrze, kontroler wejścia, magazyn i sposób aktualizacji. Dystrybucja to decyzja, kto te części wybiera i utrzymuje: my czy producent.



Rysunek 1. Siedem dystrybucji na dwóch osiach: od lekkich do pełnych i od zestawu do samodzielnego złożenia do gotowego produktu z umową wsparcia (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: kine.** Warstwa tłumacząca, która udaje przed Kubernetesem bazę etcd, a dane zapisuje w zwykłej bazie SQL: SQLite, MySQL, MariaDB albo PostgreSQL. Dzięki niej k3s z jednym serwerem nie potrzebuje etcd, a k0s może trzymać stan w bazie, którą już mamy.

Pojęcie: dqlite. Rozproszona wersja SQLite od Canonical, z własnym uzgadnianiem stanu. MicroK8s trzyma w niej stan klastra i przy trzech węzłach sam włącza wysoką dostępność.

Pojęcie: CIS Benchmark. Zestaw zaleceń utwardzenia od organizacji Center for Internet Security, osobny dla systemów i dla Kubernetesa. Audytor pyta zwykle, czy klaster jest z nim zgodny, a sprawdza to narzędzie kube-bench.

Pojęcie: FIPS 140. Amerykańska norma dla modułów kryptograficznych (wersje 140-2 i 140-3). Dystrybucja „zgodna z FIPS” używa tylko szyfrowania z certyfikowanych bibliotek, co bywa wymogiem w administracji i u ich dostawców.

Porównanie w jednej tabeli

Dystrybucja	Dla kogo	Magazyn stanu	Sieć (CNI) domyślnie	Wejście domyślnie
kubeadm	nauka (CKA), zespoły, które chcą kontrolować każdy składnik	etcd na węzłach sterujących albo zewnętrzny	brak, wybieramy sami	brak
k3s	małe zespoły, brzeg sieci, laboratoria, maszyny z małą pamięcią	SQLite przez kine przy jednym serwerze, wbudowany etcd przy HA, zewnętrzna baza SQL przez kine; w 1.37 doszedł sterownik kine dla magazynu S3 [8]	Flannel (VXLAN)	Traefik i ServiceLB
RKE2	firmy z audytem bezpieczeństwa, administracja, środowiska odcięte od internetu	wbudowany etcd	Canal, do wyboru Cilium, Calico, Flannel [9]	Traefik dla nowych klastrów od 1.36, wcześniej ingress-nginx [9]
k0s	zespoły, które chcą jednego pliku binarnego i czystego systemu pod spodem	etcd domyślnie, kine jako opcja [10]	kube-router, do wyboru Calico [11]	brak, kontrolery jako rozszerzenia

Dystrybucja	Dla kogo	Magazyn stanu	Sieć (CNI) domyślnie	Wejście domyślnie
MicroK8s	stacje programistów, Ubuntu, małe klastry z automatycznym HA	dqlite, wysoka dostępność od trzech węzłów [12]	Calico (sprawdzamy w dokumentacji dla wersji 1.36)	dodatek `ingress` z Traefikiem od 1.35, wcześniej NGINX [13]
Talos Linux	zespoły, które chcą systemu operacyjnego tylko pod Kubernetesa, bez logowania na węzły	etcd na węzłach sterujących	Flannel [14]	brak
OpenShift i OKD	duże firmy, wsparcie Red Hat, platforma z konsolą, rejestrem i budowaniem obrazów	etcd na trzech węzłach sterujących (jeden w wariancie jednowęzłowym)	OVN-Kubernetes (wtyczkę domyślną dla 4.22 sprawdzamy w dokumentacji instalacji)	Ingress Operator z routerem opartym na HAProxy

Dystrybucja	Bezpieczeństwo	Aktualizacje	Licencja i wsparcie (informacyjnie)	Wersja, którą znaleźliśmy
kubeadm	ustawienia domyślne Kubernetesa, utwardzenie CIS robimy sami	`kubeadm upgrade`, jedna wersja podrzędna naraz	Apache 2.0, społeczność	Kubernetes 1.37.1, 15 września 2026 [15]
k3s	przewodnik utwardzenia CIS w dokumentacji, część zaleceń domyślnie	skrypt z `INSTALL_K3S_VERSION` albo `system-upgrade-controller`	Apache 2.0, projekt sandbox CNCF, wsparcie komercyjne SUSE	v1.37.1+k3s1, 30 września 2026 [16]
RKE2	profil CIS w `config.yaml` (`profile: "cis"`), kryptografia zgodna z FIPS 140-2 (BoringCrypto) [17]	skrypt instalacyjny albo pakiety RPM, `system-upgrade-controller`	Apache 2.0, wsparcie komercyjne SUSE	v1.37.1+rke2r1, 24 września 2026 [18]
k0s	jeden plik binarny bez zależności od systemu; zgodność z CIS i FIPS sprawdzamy w dokumentacji aktualnej wersji	podmiana pliku binarnego, narzędzie `k0sctl`	Apache 2.0, projekt sandbox CNCF od 2025 roku, wsparcie Mirantis	v1.36.4+k0s.1, 19 września 2026; wydania 1.37 nie znaleźliśmy [19]

Dystrybucja	Bezpieczeństwo	Aktualizacje	Licencja i wsparcie (informacyjnie)	Wersja, którą znaleźliśmy
MicroK8s	izolacja snapa; utwardzenie CIS i FIPS sprawdzamy w dokumentacji Canonical	`snap refresh` w obrębie kanału, kanał zmieniamy świadomie	Apache 2.0, wsparcie Canonical	1.36 (Kubernetes 1.36), dnia wydania noty nie podają [12]
Talos Linux	brak SSH, powłoki i menedżera pakietów, niezmienny system plików, zalecenia KSPP, wiele zaleceń CIS spełnionych domyślnie, wersja FIPS 140-3 tylko z umową [20]	`talosctl upgrade` dla systemu i `talosctl upgrade-k8s` dla Kubernetesa	MPL 2.0, wsparcie Sidero Labs	1.14.4, 16 października 2026 [21]
OpenShift i OKD	ograniczenia uprawnień podów (SCC) domyślnie, tryb FIPS i Compliance Operator opisane w dokumentacji Red Hat	operator wersji klastra, kanały aktualizacji	OpenShift w subskrypcji Red Hat, OKD Apache 2.0 i społeczność	OpenShift 4.22 na Kubernetesie 1.35, lipiec 2026 [22]; numer bieżącego wydania OKD do potwierdzenia

Jak wybieramy w praktyce

Wybór dystrybucji jest wyborem tego, kogo budzimy o 3:00 w nocy. Przy kubeadm budzimy siebie, bo każdy składnik wybraliśmy sami. Przy k3s, RKE2, k0s i MicroK8s budzimy nadal siebie, ale z gotową listą składników, które ktoś przetestował razem. Przy OpenShift i przy Talosie z umową dzwonimy do producenta, za co płacimy subskrypcją i tym, że robimy wszystko po jego myśli.

Sytuacja	Dystrybucja	Uzasadnienie
Dwuosobowy zespół, kilka aplikacji, maszyny wirtualne na Hyper-V	k3s	najmniej pracy, ten sam Kubernetes co wszędzie, znamy go z `k3s01`
Ten sam zespół, ale audytor wymaga zgodności z CIS i FIPS	RKE2	ten sam producent i podobna konfiguracja co k3s, utwardzenie w pakiecie
Nauka do CKA i zrozumienie składników	kubeadm	egzamin opiera się na kubeadm, a każdą część widać

Sytuacja	Dystrybucja	Uzasadnienie
Dużo węzłów bez administratora na miejscu, brzeg sieci	Talos albo k3s	Talos nie ma czego popsuć przez SSH, k3s mało waży
Programista na laptopie	MicroK8s, k3s albo OpenShift Local	szybki start, łatwe usunięcie
Firma ze wsparciem Red Hat i zespołem platformowym	OpenShift	konsola, budowanie obrazów, rejestr i wsparcie w jednym

W Zielonej zostajemy przy k3s (produkcja małych aplikacji, laboratorium na `k3s11` do `k3s13`) i kubeadm (laboratorium do nauki na `kcp01` do `kcp03`). Gdyby pan Wojtek przy audycie zażądał zgodności z CIS, pierwszym kandydatem na zmianę jest RKE2, bo konfiguracja i narzędzia są bardzo podobne do k3s.

Częste błędy

Objaw	Przyczyna	Co robimy
Po `kubeadm init` węzeł `NotReady`	brak CNI	instalujemy wtyczkę zgodną z `--pod-network-cidr`
Pody nie startują na jedynym węźle kubeadm, stan `Pending`	skaza płaszczyzny sterowania	karta z `kubectl taint` z Fiszki, tylko dla jednego węzła
`kubectl` po instalacji RKE2 „nie istnieje”	plik leży w `/var/lib/rancher/rke2/bin/`, poza `PATH`	dopisujemy katalog do `PATH` albo podajemy pełną ścieżkę
Pody RKE2 nie dostają sieci na Rocky Linux	NetworkManager zarządza interfejsami CNI	ustawienie z sekcji Known Issues dokumentacji RKE2 i restart
Węzeł k0s nie przyjmuje drugiego węzła	instalacja z `--single`	instalujemy od nowa z `--enable-worker --no-taints`
`snap: command not found` po instalacji snapd	brak ponownego logowania, nieustawione ścieżki	wylogowanie i logowanie albo restart, potem komenda ponownie
Instrukcja z internetu instaluje ingress-nginx	projekt wycofany w marcu 2026	kontroler wejścia z pakietu dystrybucji (Traefik) albo Gateway API
Dwa klastry różnych dystrybucji w jednej firmie	każdy wybrał swoją	jedna dystrybucja na produkcję, druga tylko w laboratorium do nauki

□□ Ćwiczenia

1. Dlaczego k3s z jednym serwerem nie potrzebuje etcd, a k3s z trzema serwerami potrzebuje?

Odpowiedź

Przy jednym serwerze stan klastra trzyma SQLite, a warstwa kine tłumaczy zapytania Kubernetesa na SQL. SQLite jest plikiem na jednej maszynie, więc nie rozproszy się na trzy serwery. Przy trzech serwerach k3s używa wbudowanego etcd, które uzgadnia stan między nimi i znosi utratę jednego serwera, albo zewnętrznej bazy SQL przez kine.

1. Audytor pyta o zgodność klastra z CIS. Co odpowiadamy dla k3s, RKE2 i Talosa?

Odpowiedź

k3s: dokumentacja ma przewodnik utwardzenia CIS, który wykonujemy sami, a zgodność sprawdzamy kube-bench. RKE2: profil `cis` w `config.yaml` włącza utwardzenie przy instalacji, po przygotowaniu użytkownika etcd i ustawień sysctl. Talos: wiele zaleceń CIS jest spełnionych domyślnie, a raport robimy kube-bench z listą pominięć, którą podaje dokumentacja Talosa, bo narzędzie zgłasza na nim fałszywe alarmy.

1. Tomek chce postawić klaster Talos na `hv01` „jak k3s, skryptem po SSH”. Co jest nie tak z tym planem?

Odpowiedź

Talos jest osobnym systemem operacyjnym bez SSH i bez powłoki, więc nie instaluje się go skryptem na Rocky Linux. Maszynę uruchamiamy z obrazu Talos, a konfigurację wysyłamy przez API narzędziem `talosctl`. Na stacji administratora można najpierw poćwiczyć na klastrze w Dockerze (`talosctl cluster create docker`).

Powiązane: Orkiestracja kontenerów: od Borga do wojny orkiestratorów i tego, co z niej zostało · Swarm, k3s i Kubernetes: porównanie trzech orkiestratorów · Nomad, Podman z Quadlet i zarządzany Kubernetes: AKS i EKS · k3s w laboratorium: pierwszy deploy, nginx w kolorach · Architektura klastra: płaszczyzna sterowania i węzły · Utrzymanie klastra

Źródła: (odczyt 14 do 18 października 2026)

1. docs.k3s.io: Quick-Start Guide
2. docs.rke2.io: Quick Start
3. docs.k0sproject.io: Quick Start Guide
4. snapcraft.io: Installing snap on Rocky Linux
5. canonical.com: MicroK8s Get started
6. docs.siderolabs.com: Talos Quickstart v1.14

7. docs.redhat.com: Red Hat OpenShift Local 2.44, Getting Started Guide, rozdział „Configuring”
8. docs.k3s.io: K3s 1.37 release (10 września 2026)
9. docs.rke2.io: Basic Network Options
10. docs.k0sproject.io: Configuration options
11. docs.k0sproject.io: Networking
12. canonical.com: MicroK8s High Availability i Release notes
13. canonical.com: MicroK8s ingress addon
14. docs.siderolabs.com: Flannel
15. kubernetes.io: Releases
16. docs.k3s.io: v1.37.X release notes
17. docs.rke2.io: FIPS 140-2 Enablement i CIS Hardening Guide
18. releasealert.dev: rancher/rke2
19. github.com: k0sproject/k0s releases; cncf.io, k0s joins CNCF Sandbox (20 maja 2025)
20. docs.siderolabs.com: Talos default hardening and CIS compliance i System requirements
21. releasealert.dev: siderolabs/talos
22. docs.redhat.com: OpenShift Container Platform 4.22 release notes (Kubernetes 1.35, RHCOS na pakietach RHEL 9.8); dbta.com, „Red Hat OpenShift 4.22 is Now Available” (23 lipca 2026)
23. Licencje: github.com/siderolabs/talos (MPL-2.0), github.com/canonical/microk8s (Apache-2.0)

Revision #5
Created 2026-10-18 08:52:51 UTC by Tomasz Konieczny
Updated 2026-10-26 14:26:05 UTC by Tomasz Konieczny