Opowieść

Historia orkiestratorów od Borga do października 2026 i mapa książki.

- Orkiestracja kontenerów: od Borga do wojny orkiestratorów i tego, co z niej zostało

Orkiestracja kontenerów: od Borga do wojny orkiestratorów i tego, co z niej zostało

Poziom: podstawy

Dotyczy: Borg i Omega, Apache Mesos i Marathon, fleet, Docker Swarm (klasyczny i tryb Swarm), Kubernetes, k3s, HashiCorp Nomad, Podman z Quadlet, stan na 18 października 2026

Czas: ~25 min

Ta strona otwiera książkę o orkiestracji kontenerów. Opowiada, jak z wewnętrznego systemu Google wyrosła cała branża, jak w latach 2015 do 2017 kilka projektów walczyło o to, kto będzie układał kontenery na serwerach świata, i co z tej walki zostało w październiku 2026. Na końcu jest mapa książki, która porównuje Swarm, k3s i Kubernetesa, opisuje ich wysoką dostępność i kończy się trzema scenariuszami dla różnych rodzajów aplikacji Zielonej.

Jeden serwer z Dockerem i plikiem Compose to dla wielu firm koniec drogi i nie ma w tym nic złego. Kłopot zaczyna się wtedy, gdy serwer jest jeden, a aplikacje mają działać także w dniu jego awarii. Trzeba wtedy kogoś, kto pilnuje, ile kopii aplikacji działa, na której maszynie i co zrobić, gdy maszyna zniknie.

Tym kimś jest orkiestrator. Książka o Kubernetesie opowiada historię jednego z nich (Kubernetes: sternik, który przypłynął z Google), a książka o Dockerze historię samego kontenera (Docker: kontenery, które zmieniły serwerownie). Ta strona opowiada o wszystkich naraz, bo dopiero obok siebie widać, dlaczego jedne przetrwały, a inne nie.

> Zakres: nie podajemy cen licencji ani usług chmurowych, bo zależą od kraju, umowy i dnia. Daty i wersje sprawdziliśmy w źródłach wymienionych na końcu, a tam, gdzie źródła się różnią, piszemy to wprost w nocie metodycznej.

Skróty użyte na tej stronie (15)

Skrót	Co znaczy
AKS	Azure Kubernetes Service: Kubernetes w chmurze Microsoftu, w którym płaszczyznę sterowania utrzymuje dostawca
BSL	Business Source License: licencja, na ktora HashiCorp zmienil w 2023 roku Terraform, ograniczajaca prawo do oferowania narzedzia jako konkurencyjnej platnej uslugi
CNCF	Cloud Native Computing Foundation: fundacja pod Linux Foundation, formalny właściciel projektu Kubernetes od lipca 2015 roku.
CSI	Container Storage Interface: wtyczka, która daje dyski podom w Kubernetesie
DC/OS	Datacenter Operating System: platforma firmy Mesosphere zbudowana na Apache Mesos, z planistą usług Marathon
EL	Enterprise Linux: rodzina RHEL i jej klonów (Rocky, Alma, Oracle Linux)
etcd	etc distributed: rozproszony magazyn par klucz-wartość, w którym Kubernetes trzyma cały stan klastra, a zmiany uzgadnia protokołem Raft
GLPI	Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01
IBM	International Business Machines: firma, w której laboratorium powstał model relacyjny i pierwszy działający silnik SQL
MPL	Mozilla Public License: otwarta licencja, na której HashiCorp wydawał swoje programy do sierpnia 2023 roku i na której jest dziś Talos Linux
Raft	protokół uzgadniania stanu w klastrze: zmiana jest przyjęta, gdy zgodzi się większość węzłów; na nim stoją menedżerowie Swarm i etcd
SQLite	mała baza danych zapisana w jednym pliku; Home Assistant trzyma w niej historię w pliku home-assistant_v2.db
SUSE	nazwa niemieckiej firmy i jej dystrybucji Linuksa, od niemieckiego skrótu Software und System Entwicklung
WMS	Warehouse Management System: system zarządzania magazynem
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS

Skąd się wzięło

> **Pojęcie: orkiestrator kontenerów.** Program, który zarządza kontenerami na wielu maszynach naraz. Administrator opisuje, co ma działać (obraz, liczba kopii, porty, zasoby), a orkiestrator sam wybiera maszyny, uruchamia kontenery, odtwarza je po awarii i wymienia na nowe wersje bez zatrzymywania całości.

> **Pojęcie: planista (scheduler).** Część orkiestratora, która decyduje, na której maszynie uruchomić kolejny kontener. Patrzy na wolny procesor i pamięć, na reguły (na przykład „nie dwie kopie na jednym hoście”) i na to, które maszyny w ogóle działają.

Google: Borg i Omega

Zanim słowo „kontener” stało się modne, Google od lat uruchamiał prawie wszystkie swoje usługi na wewnętrznym systemie Borg, działającym według książki o Kubernetesie od około 2003 roku. Świat zewnętrzny poznał go dopiero z dwóch publikacji naukowych: w 2013 roku Google opisał eksperymentalnego następcę o nazwie Omega, z wieloma planistami działającymi równolegle, a w 2015 roku na konferencji EuroSys opublikował wreszcie artykuł o samym Borgu.

Te dwie publikacje stały się podręcznikiem dla wszystkich, którzy chcieli zbudować coś podobnego poza Google. HashiCorp, ogłaszając w 2015 roku swojego Nomada, napisał wprost, że jego zarządzanie klastrem opiera się na Borgu i Omedze. Kubernetes, pisany przez inżynierów Google od zera, wziął z nich pomysły, ale nie kod.

Mesos i Marathon: pierwszy duży gracz spoza Google

W 2009 roku na Uniwersytecie Kalifornijskim w Berkeley powstał projekt o roboczej nazwie Nexus, przemianowany potem na Mesos. Miał dzielić jedną pulę serwerów między wiele systemów naraz, na przykład Hadoopa i usługi WWW. Od 2010 roku używał go Twitter, który budował na nim swoje nowe usługi, a 24 lipca 2013 roku Mesos został pełnoprawnym projektem fundacji Apache.

Mesos był raczej „jądrem systemu operacyjnego dla serwerowni” niż gotowym orkiestratorem. Do uruchamiania długo działających usług potrzebował nakładki, najczęściej Marathona, rozwijanego przez firmę Mesosphere, która sprzedawała całość jako DC/OS. W 2015 roku to Mesos, a nie Kubernetes, uchodził za rozwiązanie sprawdzone w największej skali.

fleet: systemd na wielu maszynach

Równolegle CoreOS, twórca lekkiego systemu pod kontenery i bazy etcd, w 2014 roku wydał fleet: narzędzie, które rozsyłało jednostki systemd po wielu maszynach. Pomysł był prosty i dla administratora Linuksa naturalny, bo kontener był po prostu usługą systemd. W 2017 roku CoreOS zapowiedział wygaszenie fleet na rzecz Kubernetesa, a w styczniu 2018 roku samego CoreOS kupił Red Hat.

Ciekawostka: ten pomysł wrócił po latach w innym kształcie. Podman z Quadlet, opisany niżej, też robi z kontenera usługę systemd, tyle że na jednej maszynie, a pilnowanie kilku maszyn zostawia klastrowi takiemu jak Pacemaker.

Docker i dwa Swarmy

W marcu 2013 roku Solomon Hykes pokazał Dockera na konferencji PyCon i kontenery trafiły pod strzechy. Szybko okazało się, że jeden host to za mało, więc 4 grudnia 2014 roku na DockerCon Europe w Amsterdamie Docker ogłosił trzy narzędzia: Machine, Compose i Swarm. Ten pierwszy Swarm był osobnym programem, który do wykrywania węzłów potrzebował zewnętrznej bazy, na przykład Consula albo etcd.

> **Pojęcie: tryb Swarm.** Orkiestrator wbudowany w sam silnik Dockera, włączany jedną komendą `docker swarm init`. Węzły zarządzające uzgadniają stan protokołem Raft, a usługa opisana plikiem w formacie zbliżonym do Compose rozkłada się na wiele maszyn. Opisujemy go w książce o wysokiej dostępności na stronie Docker Swarm: klaster z trzema menedżerami, na którym przenosimy kolorowy nginx.

W czerwcu 2016 roku na DockerCon Docker pokazał wersję 1.12, wydaną 28 lipca 2016 roku, z trybem Swarm wbudowanym w silnik: bez dodatkowych baz, z szyfrowaną komunikacją między węzłami i z routing mesh, czyli portem otwartym na każdym węźle. Stary, osobny Swarm, nazywany odtąd klasycznym, przestał być rozwijany, a jego repozytorium zostało zarchiwizowane 1 lutego 2021 roku.

Kubernetes i fundacja CNCF

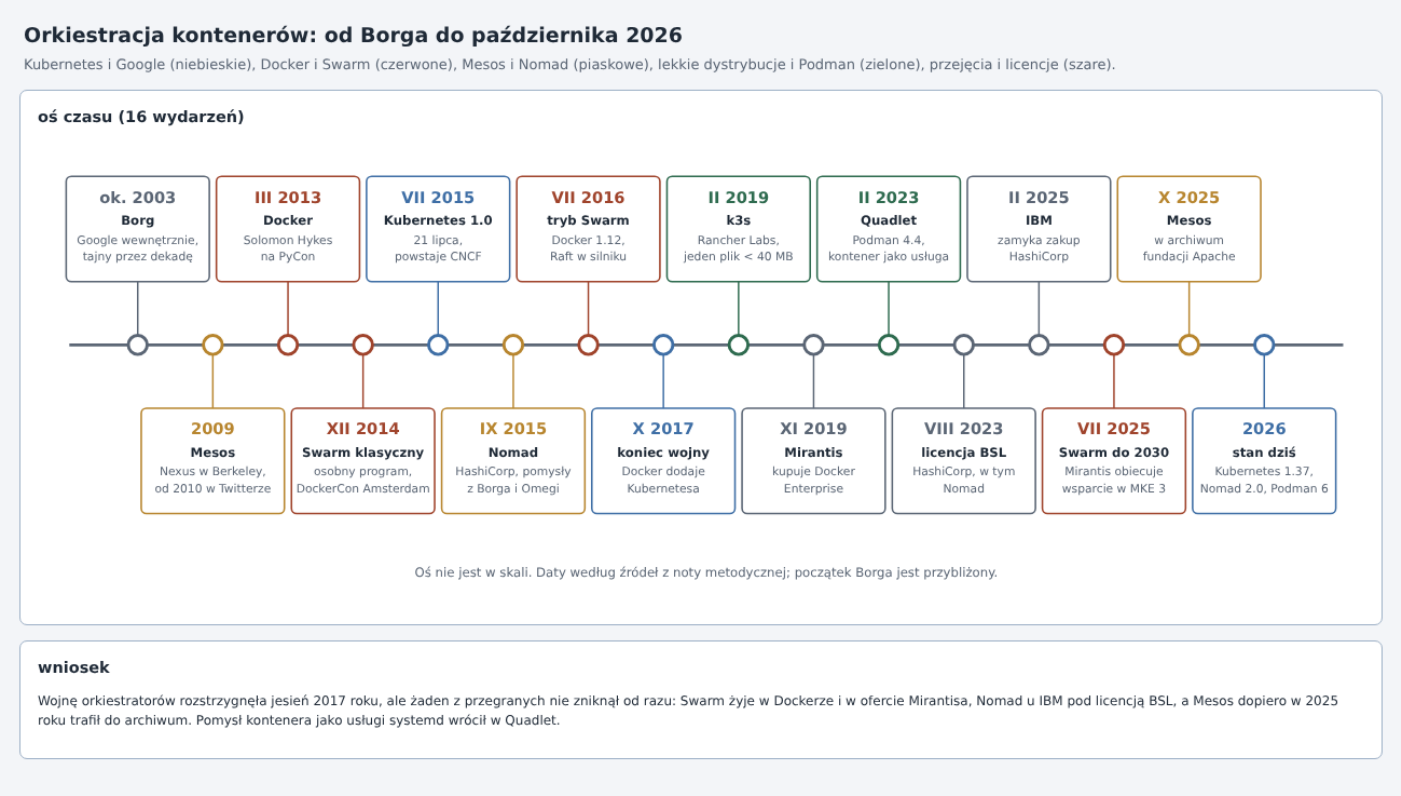
Pierwszy commit Kubernetesa trafił na GitHub 6 czerwca 2014 roku. 21 lipca 2015 roku ukazała się wersja 1.0, a Google tego samego dnia oddał projekt nowo założonej fundacji CNCF przy Linux Foundation. Miesiąc wcześniej, 24 czerwca 2015 roku, Red Hat wydał OpenShift Enterprise 3 zbudowany na Kubernetesie, co od początku dało projektowi poważnego partnera spoza Google. Szczegóły tej drogi (Borg, Omega, dockershim) opisuje książka o Kubernetesie, więc tu ich nie powtarzamy.

Wojna orkiestratorów (2015 do 2017)

Przez dwa lata administrator, który chciał klastra kontenerów, miał do wyboru co najmniej cztery drogi: Mesos z Marathonem, Swarm od Dockera, Kubernetesa i Nomada, którego HashiCorp ogłosił 28 września 2015 roku. Każda firma obiecywała, że to jej narzędzie zostanie standardem, a prasa branżowa nazwała ten okres wojną orkiestratorów.

> **Pojęcie: wojna orkiestratorów.** Potoczna nazwa lat 2015 do 2017, w których Docker, Mesosphere, Google z CNCF i HashiCorp rywalizowały o to, czyj orkiestrator stanie się standardem. Nie była to wojna na kod, tylko na partnerów, dostawców chmury i uwagę administratorów.

Rozstrzygnięcie przyszło jesienią 2017 roku. 6 września 2017 roku Mesosphere ogłosił obsługę Kubernetesa w DC/OS, a 17 października 2017 roku na DockerCon Europe w Kopenhadze sam Docker ogłosił, że jego produkty będą obsługiwać Kubernetesa obok Swarma, w wersji niezmienionej względem projektu. 6 marca 2018 roku Kubernetes został pierwszym projektem, który ukończył inkubację w CNCF.



Rysunek 1. Oś czasu orkiestracji: od Borga w Google, przez Mesosa, Swarma, Kubernetesa i Nomada w latach wojny orkiestratorów, po przejęcia, zmiany licencji i wersje z października 2026 (kliknięcie otwiera pełny rozmiar)

Po wojnie: przejęcia, lekkie dystrybucje i licencje

13 listopada 2019 roku Mirantis kupił biznes Docker Enterprise: platformę, około 750 klientów i około 300 z 400 pracowników. Prezes Mirantisa powiedział wtedy, że głównym orkiestratorem będzie Kubernetes, a Swarm będzie wspierany „co najmniej dwa lata”. Klienci zaprotestowali i 24

lutego 2020 roku Mirantis napisał, że Swarm zostaje bez daty końca, a 1 lipca 2025 roku ogłosił pełne wsparcie Swarma co najmniej do 2030 roku jako części Mirantis Kubernetes Engine 3.

Mesosphere zmienił 5 sierpnia 2019 roku nazwę na D2iQ i skręcił w stronę Kubernetesa, a jego zasoby kupił pod koniec 2023 roku Nutanix. Sam Apache Mesos w 2021 roku ledwie uniknął przeniesienia do „strychu" fundacji Apache, czyli archiwum projektów zakończonych, ale w sierpniu 2025 roku zarząd fundacji go wycofał, a w październiku 2025 roku trafił do archiwum na dobre.

> **Pojęcie: dystrybucja Kubernetesa.** Gotowy zestaw: Kubernetes z wybranymi dodatkami (sieć, magazyn, wejście z zewnątrz), instalatorem i sposobem aktualizacji. Tak jak Rocky Linux jest dystrybucją Linuksa, tak k3s czy OpenShift są dystrybucjami Kubernetesa.

W lutym 2019 roku Rancher Labs pokazał k3s: Kubernetesa w jednym pliku poniżej 40 MB, z bazą SQLite zamiast etcd dla pojedynczego węzła i z wymaganiem mniej niż 512 MB pamięci. 19 sierpnia 2020 roku k3s trafił do CNCF jako projekt sandbox (stan na październik 2026 bez zmian), a 1 grudnia 2020 roku SUSE zakończył zakup Ranchera. W ślad za nim poszli inni: k0s od Mirantisa (wydanie ogólnodostępne w kwietniu 2021, w CNCF od 2025 roku), MicroK8s od Canonical, Talos Linux od Sidero Labs. Porównuje je strona o dystrybucjach.

10 sierpnia 2023 roku HashiCorp ogłosił, że przyszłe wydania jego produktów, w tym Nomada, przechodzą z otwartej licencji MPL 2.0 na Business Source License. 24 kwietnia 2024 roku IBM ogłosił zakup HashiCorp, a 27 lutego 2025 roku zakończył transakcję wartą 6,4 miliarda dolarów. IBM, który od 2019 roku jest właścicielem Red Hata, ma więc dziś pod jednym dachem OpenShifta, Podmana i Nomada.

> **Pojęcie: licencja BSL (Business Source License).** Licencja, która pozwala czytać, zmieniać i używać kodu, ale zabrania oferowania go jako produktu konkurencyjnego wobec autora. Dla firmy, która uruchamia Nomada na własne potrzeby, zmienia niewiele, ale nie jest to już licencja otwarta w rozumieniu ruchu open source.

Podman i Quadlet: kontener jako usługa systemd

Red Hat od 2018 roku rozwijał Podmana, silnik zgodny z Dockerem, ale bez stale działającego demona. Wersja 1.0 ukazała się 16 stycznia 2019 roku, a w 2025 roku Podman trafił do CNCF. W lutym 2023 roku, w Podmanie 4.4, pojawił się Quadlet autorstwa Alexandra Larssona.

> **Pojęcie: Quadlet.** Mechanizm Podmana, w którym kontener opisujemy krótkim plikiem `.container` w składni systemd, a systemd sam zamienia go w usługę. Kontener startuje z

> systemem, restartuje się po awarii i ma dziennik w `journalctl`, jak każda inna usługa.

Quadlet nie jest orkiestratorem klastra i nie udaje nim. Dla małego zespołu bywa jednak rozsądną alternatywą: pojedyncza usługa w kontenerze na dwóch maszynach, przełączana przez Pacemakera, jest prostsza niż cały Kubernetes. Ten wariant porównujemy na stronie o Nomadzie, Podmanie i chmurze.

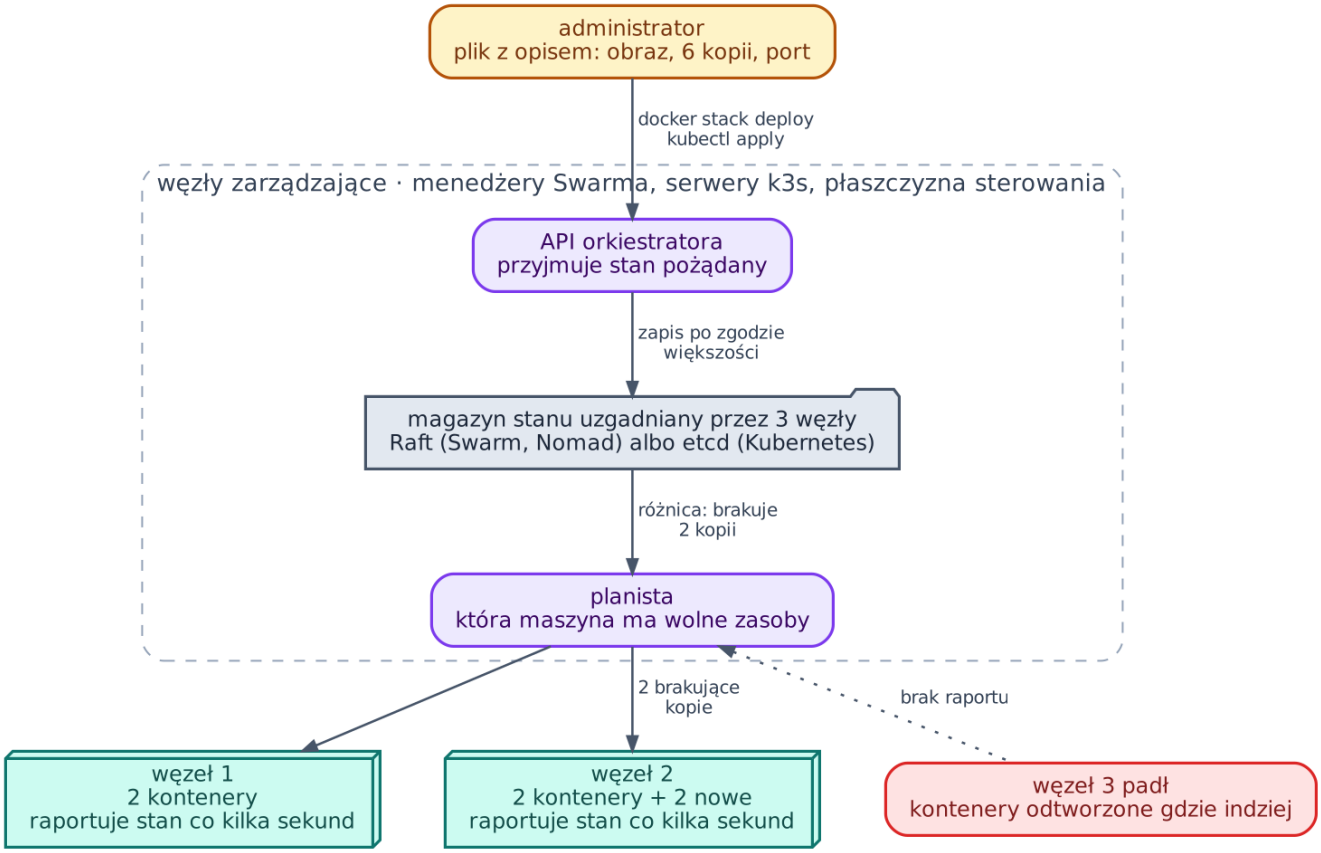
Stan na październik 2026

Projekt	Ostatnie wydanie, które znaleźliśmy	Co to znaczy dla nas
Docker Engine z trybem Swarm	29.9.1, 14 października 2026	tryb Swarm jest w silniku i w dokumentacji, ale informacji o zmianach w nim w ostatnich wydaniach nie znaleźliśmy; Mirantis obiecuje wsparcie co najmniej do 2030 roku w swoim produkcie komercyjnym
Kubernetes	1.37 (wydany 26 sierpnia 2026), poprawka 1.37.1 z 15 września 2026	nowa wersja podrzędna co około cztery miesiące, każda wspierana około 14 miesięcy
k3s	v1.37.1+k3s1, 30 września 2026	nadal projekt sandbox CNCF, rozwijany przez SUSE; nasze laboratoria z książki o wysokiej dostępności stoją na v1.36
HashiCorp Nomad	2.0.7, 17 września 2026 (2.0.0 z 21 kwietnia 2026)	licencja BSL, od wersji 2.0 numeracja i cykl wsparcia według zasad IBM
Podman	6.1.4, 12 października 2026 (6.0 z 24 czerwca 2026)	Rocky Linux 9.8 ma w repozytorium serię 5.8, więc na naszych serwerach pracujemy na niej
Apache Mesos	ostatnie stabilne wydanie 1.11.0 z 24 listopada 2020	w archiwum Apache od października 2025 roku

Czy Swarm jest rozwijany? Uczciwa odpowiedź brzmi: utrzymywany tak, rozwijany niewiele. Silnik Dockera nadal go zawiera, a Mirantis rozwija wokół niego obsługę magazynów CSI w swoim produkcie, ale w informacjach o wydaniach Dockera 29 nie znaleźliśmy nowości dotyczących Swarma. Stan repozytorium `moby/swarmkit` sprawdzamy sami przed decyzją, bo w tej sesji nie mogliśmy odczytać historii zmian.

Jak to działa w jednym obrazku

Każdy orkiestrator z tej książki, od Swarma po AKS w chmurze, działa według tej samej pętli. Administrator zapisuje stan pożądany, orkiestrator przechowuje go w uzgodnionej bazie na kilku maszynach, planista rozdziela kontenery na węzły, a węzły stale raportują, co naprawdę działa. Gdy rzeczywistość rozjeżdża się z opisem, orkiestrator poprawia rzeczywistość, nie opis.



Rysunek 2. Jedna pętla dla wszystkich orkiestratorów: plik z opisem, magazyn stanu uzgadniany przez trzy węzły, planista i węzły robocze, które raportują stan; po awarii węzła pętla odtwarza brakujące kontenery gdzie indziej (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: stan pożądany.** Opis tego, co ma działać, zapisany w pliku: obraz `nginx:1.30.5`, sześć kopii, port 8080. Orkiestrator porównuje go co kilka sekund ze stanem faktycznym i usuwa różnice.

Różnice między orkiestratorami leżą w szczegółach tej pętli: gdzie trzymają stan (Raft wbudowany w Dockera, etcd, SQLite przez warstwę kine, Raft w Nomadzie), ile maszyn potrzebują do kworum, jak wpuszczają ruch z zewnątrz i co robią z danymi aplikacji. Teorię kworum i podziału klastra wykłada strona or2-01, a my w tej książce wracamy do niej tylko odsyłaczem.

Dlaczego Zielona

> **Z ŻYCIA WZIĘTE**
>
> Środa, 13:30. Ola wraca z meetupu w Poznaniu z listą: „Swarm, bo umiemy Dockera, k3s, bo jest lekki, albo od razu prawdziwy Kubernetes, bo tak robią wszyscy”. Tomek dorzuca Nomada, bo przeczytał, że jest prostszy, a pan Zbigniew pyta tylko, czy któraś z tych rzeczy sprawi, że awaria `docker01` w listopadzie nie zatrzyma poczty i Vaultwardena. Nikt przy stole nie umie odpowiedzieć bez porównania, więc umawiamy się na nie przed 15 listopada, bo potem obowiązuje zakaz zmian do 6 stycznia.

Zielona ma dziś jeden `docker01` z Gitea, NetBoksem, Graylogiem, MailCow, Vaultwardenem i GLPI, a książka o wysokiej dostępności proponuje dla niego ciepłą rezerwę na `docker02` z adresem wirtualnym. Laboratorium Kubernetesa na `k3s01` służy zespołowi Oli do prób. Optima na `sql01` i WMS na `ora01` w ogóle nie stoją w kontenerach i żaden orkiestrator tego nie zmieni.

Pytanie nie brzmi więc „który orkiestrator jest najlepszy”, tylko „czy dwuosobowy zespół potrzebuje orkiestratora i jeśli tak, którego”. Ta książka buduje trzy laboratoria (Swarm, k3s z trzema serwerami, Kubernetes z kubeadm), psuje je celowo i na końcu pokazuje trzy scenariusze dla trzech rodzajów aplikacji, każdy do odtworzenia w laboratorium krok po kroku. Który scenariusz trafi do produkcji, zdecyduje autor razem z panem Zbigniewem.

Co wniosła i co zabrała orkiestracja

Wniosła	Zabrała albo utrudniła
samonaprawę: kontener po awarii wstaje sam na innej maszynie	nową warstwę, która sama może się zepsuć (kworum, certyfikaty, sieć nakładkowa)
aktualizacje kroczące bez przerwy dla użytkownika	przepisanie istniejących plików Compose na nowy format
jeden opis aplikacji w repozytorium zamiast notatek z konsoli	złudzenie, że dane też przenoszą się same; nie przenoszą się
wspólny język i ogromny ekosystem wokół Kubernetesa	krzywą nauki, która dla dwóch osób oznacza tygodnie, a nie dni
przenośność między serwerownią a chmurą (AKS, EKS)	zależność od dostawców i ich zmian licencji, co pokazał HashiCorp

Mity i sprostowania

Obiegowa wersja	Jak było naprawdę
Kubernetes to otwarty Borg.	Kubernetes napisano od zera; wziął z Borga i Omegi pomysły, ale nie kod.
Docker Swarm umarł w 2019 roku.	Mirantis zapowiadał najpierw dwa lata wsparcia, po protestach klientów wycofał się z tego, a w 2025 roku obiecał wsparcie do co najmniej 2030 roku. Tryb Swarm jest w Dockerze 29.
Swarm i tryb Swarm to to samo.	Swarm klasyczny z 2014 roku był osobnym programem i jest zarchiwizowany od 2021 roku; tryb Swarm z Dockera 1.12 to inny, wbudowany mechanizm.
k3s to okrojony, „nieprawdziwy” Kubernetes.	Dokumentacja k3s opisuje go jako w pełni zgodną dystrybucję Kubernetesa, a przy premierze usunięto z niego głównie kod przestarzały i dodatki domyślnie wyłączone.
Nomad jest dalej otwarty, bo kod jest na GitHubie.	Kod jest publiczny, ale od sierpnia 2023 roku na licencji BSL, która nie jest licencją otwartą.
Wojnę wygrał Kubernetes, bo był najlepszy technicznie.	Zadecydowało w dużej mierze to, że oddano go neutralnej fundacji i poparli go dostawcy chmury oraz Red Hat; Swarm był prostszy, a Mesos sprawdzony w większej skali.

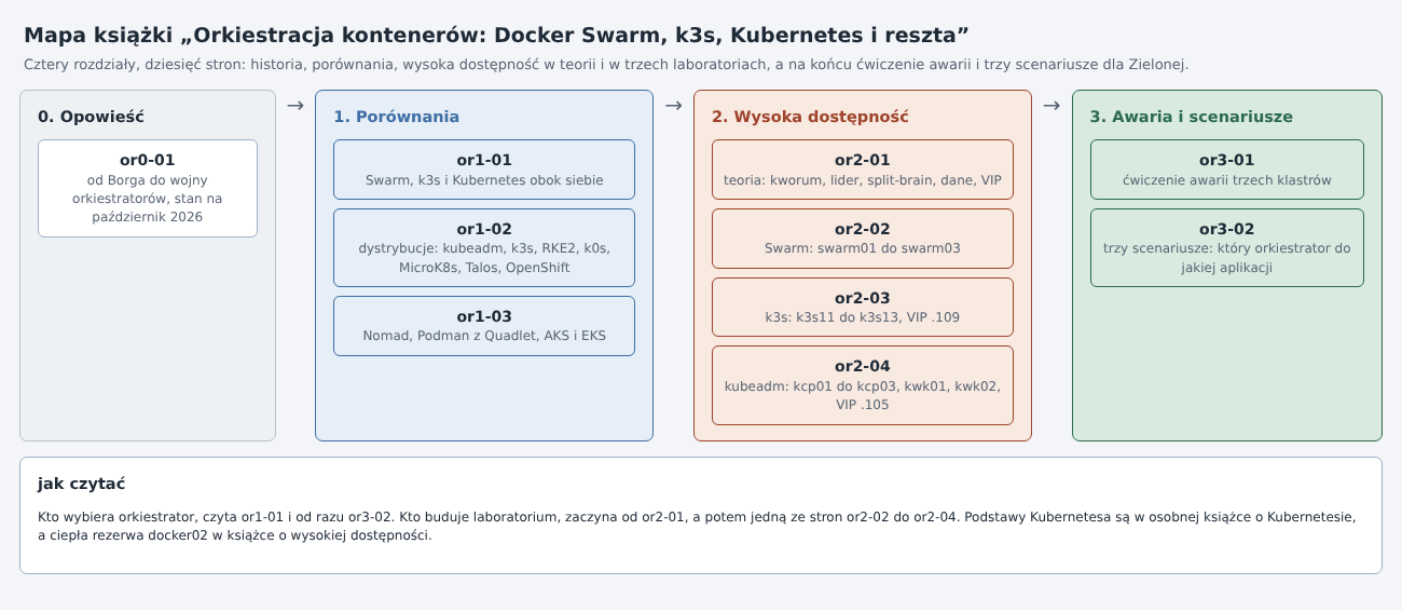
Oś czasu

Data	Wydarzenie
około 2003	Borg zaczyna pracę wewnątrz Google (data przybliżona, za książką o Kubernetesie)
2009	Nexus, później Mesos, na Uniwersytecie Kalifornijskim w Berkeley
marzec 2013	Docker pokazany na PyCon
2013	publikacja o Omedze (EuroSys)
24 lipca 2013	Mesos projektem najwyższego poziomu Apache
2014	fleet od CoreOS
6 czerwca 2014	pierwszy commit Kubernetesa
4 grudnia 2014	Docker ogłasza Swarm (klasyczny), Machine i Compose
2015	publikacja o Borgu (EuroSys)
24 czerwca 2015	OpenShift Enterprise 3 na Kubernetesie
21 lipca 2015	Kubernetes 1.0 i powstanie CNCF
28 września 2015	HashiCorp ogłasza Nomada

Data	Wydarzenie
27 lipca 2016	Mesos 1.0
28 lipca 2016	Docker 1.12 z trybem Swarm
6 września 2017	Mesosphere ogłasza Kubernetesa w DC/OS
2017	CoreOS zapowiada wygaszenie fleet
17 października 2017	Docker ogłasza obsługę Kubernetesa obok Swarma
30 stycznia 2018	Red Hat kupuje CoreOS
6 marca 2018	Kubernetes pierwszym absolwentem CNCF
16 stycznia 2019	Podman 1.0
26 lutego 2019	pierwsze doniesienia o k3s od Rancher Labs
5 sierpnia 2019	Mesosphere zmienia nazwę na D2iQ
13 listopada 2019	Mirantis kupuje Docker Enterprise
24 lutego 2020	Mirantis: Swarm zostaje bez daty końca
19 sierpnia 2020	k3s w CNCF (sandbox)
1 grudnia 2020	SUSE kończy zakup Ranchera
1 lutego 2021	archiwizacja repozytorium Swarma klasycznego
kwiecień 2021	k0s ogólnodostępny
luty 2023	Quadlet w Podmanie 4.4
10 sierpnia 2023	HashiCorp przechodzi na licencję BSL
grudzień 2023	Nutanix przejmuje zasoby D2iQ (formalnie ogłoszone w marcu 2024)
24 kwietnia 2024	IBM ogłasza zakup HashiCorp
21 stycznia 2025	Podman przyjęty do CNCF
27 lutego 2025	IBM kończy zakup HashiCorp
2025	k0s w CNCF (sandbox)
1 lipca 2025	Mirantis: Swarm wspierany co najmniej do 2030 roku
październik 2025	Apache Mesos w archiwum fundacji
21 kwietnia 2026	Nomad 2.0.0
24 czerwca 2026	Podman 6.0
26 sierpnia 2026	Kubernetes 1.37
14 października 2026	Docker Engine 29.9.1

Mapa tej książki

Książka ma dziesięć stron w czterech rozdziałach. Strony pierwszego i czwartego rozdziału piszemy od strony wyboru, a strony drugiego i trzeciego od strony awarii.



Rysunek 3. Mapa książki: opowieść, porównania (Swarm, k3s i Kubernetes, dystrybucje, Nomad, Podman i chmura), wysoka dostępność w teorii i w trzech laboratoriach, ćwiczenie awarii i trzy scenariusze dla Zielonej (kliknięcie otwiera pełny rozmiar)

Strona	Na jakie pytanie odpowiada
Orkiestracja kontenerów: od Borga do wojny orkiestratorów i tego, co z niej zostało	Skąd się wzięły orkiestratory i co zostało po wojnie
Swarm, k3s i Kubernetes: porównanie trzech orkiestratorów	Czym różnią się Swarm, k3s i Kubernetes z kubeadm w pojęciach, sieci, magazynie, aktualizacjach i pracy admina
Dystrybucje Kubernetesa: kubeadm, k3s, RKE2, k0s, MicroK8s, Talos i OpenShift	Którą dystrybucję wybrać, czym trzymają stan i jak je zainstalować na jednym węźle
Nomad, Podman z Quadlet i zarządzany Kubernetes: AKS i EKS	Co poza Kubernetesem: Nomad, kontenery jako usługi systemd z Pacemakerem i Kubernetes w chmurze
Wysoka dostępność orkiestratorów: kworum, etcd, dane i adres wirtualny	Jak działa kworum Raft i etcd, lider, podział klastra, dane trwałe i adres wirtualny
Docker Swarm w wysokiej dostępności: Raft, autolock, kopia i odzyskanie kworum	Jak zbudować i utrzymać Swarm z trzema menedżerami na `swarm01` do `swarm03`

Strona	Na jakie pytanie odpowiada
k3s w wysokiej dostępności: migawki etcd, wymiana węzła, odtworzenie i aktualizacja	Jak zbudować k3s z trzema serwerami na `k3s11` do `k3s13` z adresem wirtualnym API
Kubernetes z kubeadm w wysokiej dostępności: trzy węzły płaszczyzny sterowania, etcd, certyfikaty i aktualizacja	Jak zbudować płaszczyznę sterowania na `kcp01` do `kcp03` z węzłami `kwk01` i `kwk02`
Ćwiczenie awarii na trzech klastrach: Swarm, k3s i kubeadm równolegle	Co się dzieje, gdy wyłączamy węzły, lidera i adres wirtualny w każdym z trzech laboratoriów
Trzy scenariusze dla Zielonej: który orkiestrator do jakiej aplikacji	Który orkiestrator pasuje do jakiego rodzaju aplikacji, z trzema kompletnymi scenariuszami do odtworzenia w laboratorium

Nota metodyczna: czego nie wiemy

- Początku Borga nie podajemy co do roku: książka o Kubernetesie pisze o około 2003 roku, a publikacja Google z 2015 roku mówi tylko o ponad dekadzie pracy.
- Daty powstania Marathona i założenia Mesosphere nie znaleźliśmy w źródłach, do których dotarliśmy, więc jej nie podajemy.
- Daty zapowiedzi wygaszenia fleet nie podajemy co do miesiąca: artykuł ServerWatch z 31 maja 2017 roku opisuje już wygaszanie, ale nie podaje dnia zapowiedzi.
- Dnia przejścia HashiCorp na BSL nie podaje wprost InfoQ; The Register z 11 sierpnia 2023 roku pisze, że ogłoszenie padło „wczoraj”, czyli 10 sierpnia.
- Strona Wikipedii o Mesosie podaje wycofanie projektu w sierpniu 2025 roku, a strona archiwum Apache przeniesienie w październiku 2025 roku; podajemy obie daty jako dwa kroki.
- Książka o wysokiej dostępności pisze na stronie o Swarmie, że Swarm powstał w 2014 roku jako biblioteka SwarmKit. Według źródeł z tej strony w 2014 roku powstał Swarm klasyczny, a tryb Swarm (zbudowany na SwarmKit) wszedł do silnika w Dockerze 1.12 w 2016 roku.
- Czy w `moby/swarmkit` w 2026 roku trafiają nowe funkcje, czy tylko poprawki, nie sprawdziliśmy, bo historia zmian repozytorium była dla nas niedostępna.
- Liczby z mapy wersji (Kubernetes, k3s, Nomad, Podman, Docker) odczytaliśmy między 14 a 18 października 2026 roku i po tej dacie mogą już być nowsze.

Powiązane: Kubernetes: sternik, który przypłynął z Google · Po co Kubernetes, skoro mamy Dockera · Docker: kontenery, które zmieniły serwerownie · Docker Swarm: klaster z trzema menedżerami, na którym przenosimy kolorowy nginx · k3s z trzema węzłami serwerowymi: etcd,

adres wirtualny API i Longhorn · Wzorce projektowania wysokiej dostępności: jak dobrać sposób do rodzaju usługi · Macierz wysokiej dostępności usług Zielonej: poziom rezerwy każdej usługi, zależności i kolejność prac

Źródła

Odczyt 14 do 18 października 2026.

Google i Kubernetes: research.google/pubs/pub43438 („Large-scale cluster management at Google with Borg”, EuroSys 2015); kubernetes.io/blog/2024/06/06/10-years-of-kubernetes (pierwszy commit 6 czerwca 2014); cncf.io/news/2015/07/21 (Kubernetes 1.0 i CNCF); cncf.io/announcements/2018/03/06/kubernetes-first-cncf-project-graduate; kubernetes.io/releases i kubernetes.io/releases/1.37 (1.37.1 z 15 września 2026); docs.aws.amazon.com/eks/latest/userguide/kubernetes-versions.html (wydanie upstream 1.37 26 sierpnia 2026); redhat.com, komunikat „Red Hat Delivers OpenShift Enterprise 3” (24 czerwca 2015).

Mesos, Mesosphere i fleet: en.wikipedia.org/wiki/Apache_Mesos; attic.apache.org/projects/mesos.html; blog.x.com/engineering, „Mesos graduates from Apache incubation” (2013); thenewstack.io, „Apache Mesos narrowly avoids a move to the attic, for now”; techcrunch.com, „Mesosphere adds Kubernetes support to its data center operating system” (6 września 2017); techcrunch.com, „Mesosphere changes name to D2iQ” (5 sierpnia 2019); lemondeinformatique.fr, „Nutanix formalise le rachat des actifs de D2iQ” (12 marca 2024); serverwatch.com, „CoreOS Fleet Fades Away in Favor of Kubernetes and Tectonic” (31 maja 2017); cnbc.com, „Red Hat buys CoreOS for 250 million” (30 stycznia 2018).

Docker i Swarm: techcrunch.com, „Docker launches its first commercial product and gets into orchestration tools” (4 grudnia 2014); infoq.com, „DockerCon: Docker Swarm” (28 czerwca 2016); versionlog.com/docker-engine/1.12 (28 lipca 2016); github.com/docker/classicswarm (archiwum od 1 lutego 2021); thenewstack.io, „Docker fully embraces Kubernetes” (17 października 2017); thenewstack.io, „Mirantis acquires Docker Enterprise” (13 listopada 2019); mirantis.com/blog/mirantis-will-continue-to-support-and-develop-docker-swarm (24 lutego 2020); mirantis.com/blog/mirantis-guarantees-long-term-support-for-swarm (1 lipca 2025); docs.docker.com/engine/release-notes/29 (29.9.1 z 14 października 2026).

Rancher, k3s i inne dystrybucje: devclass.com, „Rancher throws out the old for small but edge-y Kubernetes distribution” (26 lutego 2019); cncf.io/projects/k3s (sandbox od 19 sierpnia 2020); siliconangle.com, „SUSE completes acquisition of Rancher Labs” (1 grudnia 2020); docs.k3s.io/release-notes/v1.37.X; mirantis.com/blog/k0s-zero-friction-kubernetes-now-in-general-release (27 kwietnia 2021); cncf.io/blog/2025/05/20/a-milestone-for-lightweight-kubernetes-k0s-joins-cncf-sandbox.

HashiCorp i Nomad: hashicorp.com/blog/nomad-announcement (28 września 2015); theregister.com/2023/08/11/hashicorp_bsl_licence; infoq.com/news/2023/08/hashicorp-adopts-bsl; newsroom.ibm.com, komunikaty z 24 kwietnia 2024 i 27 lutego 2025; developer.hashicorp.com/nomad/docs/release-notes/v2-0-x; developer.hashicorp.com/nomad/docs/ce-license-support; releases.hashicorp.com/nomad.

Podman i Quadlet: en.wikipedia.org/wiki/Podman; redhat.com/en/blog/quadlet-podman (17 lutego 2023); redhat.com/en/blog/red-hat-contribute-comprehensive-container-tools-collection-cloud-native-computing-foundation (14 listopada 2024); lwn.net/Articles/1079600 (Podman 6.0); endoflife.ai/podman/6.1; git.almalinux.org/rpms/podman (pakiet podman-5.8.2 dla EL 9.8).

Ilustracje: wszystkie rysunki na tej stronie są własne, narysowane dla tej bazy.