

# *Pierwsze kroki z Dockerem i Kubernetes*

---

## **W tym rozdziale:**

- Tworzenie, uruchamianie i udostępnianie obrazu kontenera przy użyciu Dockera
- Uruchamianie lokalnego, jednowęzłowego klastra Kubernetes
- Konfigurowanie klastra Kubernetes w Google
- Kubernetes Engine
- Konfigurowanie i używanie klienta wiersza poleceń `kubectl`
- Wdrażanie aplikacji w Kubernetes i skalowanie jej w poziomie

Zanim zaczniemy szczegółowo poznawać koncepcje Kubernetes, przyjrzyjmy się, jak utworzyć prostą aplikację, opakować ją w obraz kontenera i uruchomić w zarządzanym klastrze Kubernetes (w Google Kubernetes Engine) albo w lokalnym jednowęzłowym klastrze. Powinno to dać Czytelnikowi nieco lepszy obraz całego systemu Kubernetes i ułatwi lekturę kilku następnych rozdziałów, w których zajmiemy się podstawowymi blokami konstrukcyjnymi i koncepcjami używanymi w Kubernetes.

## 2.1 Tworzenie, uruchamianie i udostępnianie obrazu kontenera

Jak wiemy z poprzedniego rozdziału, uruchamianie aplikacji w Kubernetes wymaga ich zapakowania w obrazy kontenerów. Pokażemy teraz podstawowe wprowadzenie do korzystania z Dockera na wypadek, gdyby Czytelnik jeszcze go nie używał. W kolejnych kilku podrozdziałach wykonamy następujące działania:

- 1 Zainstalujemy Dockera i uruchomimy swój pierwszy powitalny kontener „Hello world”.
- 2 Utworzymy trywialną aplikację Node.js, którą później wdrożymy w Kubernetes.
- 3 Opakujemy aplikację w obraz kontenera, tak by mogła zostać uruchomiona jako izolowany kontener.
- 4 Uruchomimy kontener oparty na tym obrazie.
- 5 Wypchniemy obraz do Docker Hub, tak by każdy mógł go uruchomić gdziekolwiek.

### 2.1.1 Instalowanie Dockera i uruchamianie powitalnego kontenera

Najpierw musimy zainstalować Dockera w swoim komputerze systemu Linux. Jeśli ktoś nie używa Linuksa, może uruchomić linuksową maszynę wirtualną (VM) i uruchomić Dockera wewnątrz tej maszyny wirtualnej. Jeśli ktoś używa Maca lub Windowsów i zainstaluje Dockera zgodnie z instrukcją, Docker skonfiguruje potrzebną maszynę wirtualną i uruchomi demona Docker wewnątrz tej maszyny wirtualnej. W macierzystym systemie operacyjnym dostępny będzie wykonywalny klient Dockera, który może komunikować się z demonem wewnątrz maszyny wirtualnej.

W celu zainstalowania Dockera należy postępować zgodnie z instrukcjami dla konkretnego systemu operacyjnego, dostępnymi pod adresem <http://docs.docker.com/engine/install/>. Po ukończeniu instalacji będzie można użyć wykonywalnego programu klienckiego w celu uruchamiania różnych poleceń Dockera. Możemy na przykład próbować pobrać i uruchomić istniejący obraz z Docker Hub, publicznego rejestru Dockera, który zawiera gotowe do użycia obrazy kontenerów dla wielu dobrze znanych pakietów oprogramowania. Jednym z nich jest obraz `busybox`, którego użyjemy do wykonania prostego polecenia `echo „Hello world”`.

#### URUCHAMIANIE POWITALNEGO KONTENERA

Dla osób, które nie znają narzędzia `busybox` – jest to pojedynczy plik wykonywalny, który łączy w sobie wiele standardowych uniksowych narzędzi wiersza poleceń, takich jak `echo`, `ls`, `gzip` i tak dalej. Zamiast obrazu `busybox` moglibyśmy użyć również innych w pełni wyposażonych kontenerów systemu operacyjnego, takich jak Fedora, Ubuntu lub inne podobne, jeśli tylko będą one zawierać plik wykonywalny `echo`.

Jak uruchomić obraz `busybox`? Nie trzeba pobierać ani instalować czegokolwiek. Wystarczy użyć polecenia `docker run` i wskazać, jaki obraz ma zostać pobrany i uruchomiony oraz (opcjonalnie), jakie polecenie ma być wykonane, co widać w poniższym listingu.