

# Spiderpool

RDMA network solution for the Kubernetes

Piotr Czarnik, Bartosz Kucharz  
Gabriela Piwar, Wojciech Szmelich

AGH Wydział Informatyki  
2024

# Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
<b>2</b>	<b>Opis technologii</b>	<b>2</b>
2.1	Kubernetes . . . . .	2
2.2	AWS . . . . .	2
2.3	Ansible . . . . .	3
<b>3</b>	<b>Case Study - opis projektu</b>	<b>3</b>
<b>4</b>	<b>Architektura rozwiązania</b>	<b>4</b>
<b>5</b>	<b>Konfiguracja środowiska</b>	<b>4</b>
<b>6</b>	<b>Instalacja</b>	<b>4</b>
<b>7</b>	<b>How to reproduce - steps</b>	<b>4</b>
7.1	Infrastructure as Code approach . . . . .	4
<b>8</b>	<b>Demo deployment steps</b>	<b>4</b>
8.1	Configuration set-up . . . . .	4
8.2	Data preparation . . . . .	4
8.3	Execution procedure . . . . .	4
8.4	Results presentation . . . . .	4
<b>9</b>	<b>Podsumowanie</b>	<b>4</b>

# 1 Wprowadzenie

Kubernetes to jedno z najpopularniejszych narzędzi do zarządzania aplikacjami kontenerowymi. Z tego też względu nieustannie wprowadzane są nowe moduły usprawniające jego działanie.

Jednym z nich jest Spiderpool - zaawansowane rozwiązanie zarządzania adresami IP (IPAM - IP Address Management) wykorzystujące technologię RDMA (Remote Direct Memory Access). Rozszerza on standardowe interfejsy sieciowe kontenerów (CNI - Container Network Interface) umożliwiając tworzenie interfejsów Macvlan, Ipvlan, oraz SR-IOV. Dzięki temu pozwala na większą dowolność w przypisywaniu adresów IP do kontenerów i w wykorzystaniu interfejsów sieciowych. Natomiast SR-IOV umożliwia kontenerowi na bezpośredni dostęp do fizycznego interfejsu sieciowego - szybszy transfer danych między węzłami w klastrze Kubernetesa, minimalizacja opóźnień i obciążenia procesora. Jest to szczególnie korzystne dla aplikacji wymagających wysokiej przepustowości i niskiego opóźnienia, jak aplikacje do przetwarzania dużych ilości danych, middleware, CNF (Container Network Functions) czy systemy baz danych.

## 2 Opis technologii

### 2.1 Kubernetes

Spiderpool działa na klastrach, czyli zestawie maszyn (węzłów) do uruchamiania skonteneryzowanych aplikacji. Kubernetes jest platformą open source do zarządzania takimi klastrami. Służy do zarządzania zadaniami i serwisami uruchamianymi w kontenerach, oraz umożliwia deklaratywną konfigurację i automatyzację. Najmniejsza i najprostsza jednostka w środowisku Kubernetes to pod, czyli grupa jednego lub wielu kontenerów aplikacji. W "czystym" k8s kontenery wewnątrz poda współdzielą adres IP i przestrzeń portów, zawsze są uruchamiane wspólnie w tej samej lokalizacji i współdzielą kontekst wykonawczy na tym samym węźle.

### 2.2 AWS

Spiderpool jest stworzone z myślą o działaniu na dowolnym środowisku chmurowym. Ułatwia również zarządzanie takimi rozwiązaniami jak multicloud

czy chmura hybrydowa.

Jedną z najbardziej znanych i używanych platform chmurowych jest Amazon Web Services (AWS), która zapewnia szeroki wybór usług oraz zasobów obliczeniowych, sieciowych i przechowywania danych. Usługi Amazona są znacznie rozbudowane i umożliwiają skonfigurowanie środowiska w taki sposób, aby było jak najbardziej dopasowane do danych potrzeb. Jedną z najważniejszych usług dostępnych w AWS jest Elastic Compute Cloud (EC2), która umożliwia elastyczne skalowanie zasobów obliczeniowych. Szerokie zastosowanie tej platformy oznacza również, że istnieje ogromna ilość informacji, dokumentacji i pomocy dostępnych dla użytkowników. Dlatego zdecydowano się wdrożyć projekt na tym środowisku.

## 2.3 Ansible

Ansible jest silnikiem orkiestracji pozwalającym na tworzenie oprogramowania w paradygmacie „infrastructure as a code”. Umożliwia automatyzację provisioningu, konfiguracji i deploymentu systemów oraz oprogramowania za pomocą Playbook-ów - zestawów tasków, które mają się wykonać na wcześniej zdefiniowanych node’ach.

Ansible udostępnia moduły dedykowane do konfiguracji AWSa i Kubernetesa. Wartą uwagi cechą playbooków jest idempotentność operacji - taski sprawdzają, czy dane zadanie już nie zostało wykonane, a jeśli tak to go nie powtarzają - umożliwia to wielokrotne ich uruchamianie bez konieczności czyszczenia całego środowiska.

## 3 Case Study - opis projektu

Za pomocą wymienionych uprzednio technologii, dążymy do uzyskania automatycznego deploymentu środowiska Spiderpool na chmurze AWS za pomocą Ansible’a. W miejscach, gdzie użycie natywnych Ansible’owych modułów nie będzie możliwe, wykorzystane zostaną skrypty bash-owe lub (co jest bardziej preferowane) skrypt bash-owy wewnątrz playbooka.

Po deploymentcie Spiderpoola konieczne jest zautomatyzowanie sprawdzenia poprawności działania środowiska:

1. test łączności sieciowej między kontenerami - za pomocą ping-ów lub curl-ów,

2. test działania funkcjonalności Spiderpoola - dodawanie i uwalnianie adresów IP,
3. uruchomienie kilku przykładowych aplikacji web-wych w kontenerach używających różnych adresów IP za pomocą Macvlan lub Ipvlan.

W ramach projektu zautomatyzowane zostaną więc:

1. konfiguracja AWS - VPC i EC2,
2. deployment K8s,
3. deployment Spiderpoola,
4. testy, a w tym deployment przykładowych aplikacji.

## **4 Architektura rozwiązania**

## **5 Konfiguracja środowiska**

## **6 Instalacja**

## **7 How to reproduce - steps**

### **7.1 Infrastructure as Code approach**

## **8 Demo deployment steps**

### **8.1 Configuration set-up**

### **8.2 Data preparation**

### **8.3 Execution procedure**

### **8.4 Results presentation**

## **9 Podsumowanie**