

## ДОДАТКОВИЙ МЕХАНІЗМ МЕРЕЖЕВОЇ ВЗАЄМОДІЇ IN KUBERNETES

<sup>1</sup>Маньківський В.Б., <sup>2</sup>Конахович Г.Ф., <sup>2</sup>Романов А.О.

<sup>1</sup>Навчально-науковий інститут телекомунікаційних систем

КІП ім. Ігоря Сікорського, Україна

<sup>2</sup>Національний авіаційний університет, Київ, Україна

E-mail: tks@nau.edu.ua; v.b.mankivskiy@gmail.com; anton3329@gmail.com

### ADDITIONAL NETWORK INTERCONNECTION IN KUBERNETES

Containerization of applications and their deployment in a cloud environment is a key factor in ensuring the scalability, elasticity and availability of network applications. Kubernetes, a state-of-the-art container management platform, provides many capabilities for automatically scaling resources to ensure high availability and application performance. However, there are limitations that lead to the need to improve the mechanisms of network interaction, especially in conditions of uncertainty and variability of the environment.

Контейнеризація програм і їх розгортання в хмарному середовищі є ключовим фактором забезпечення масштабованості, еластичності та доступності мережеских програм. Kubernetes, найсучасніша платформа керування контейнерами, надає багато можливостей для автоматичного масштабування ресурсів для забезпечення високої доступності та ефективності програм. Однак існують обмеження, які призводять до необхідності вдосконалення механізмів мережескої взаємодії, особливо в умовах невизначеності та мінливості середовища.

Мережа є критично важливим компонентом будь-якої розподіленої системи, включно з Kubernetes. Однак проблеми з мережею можуть вплинути на доступність, надійність і продуктивність програм, розгорнутих у Kubernetes. У цій роботі ми досліджуємо вплив мережеских проблем у Kubernetes і аналізуємо найкращі методи виявлення та вирішення цих проблем.

Kubernetes - це популярна платформа оркестровки контейнерів з відкритим вихідним кодом, яка дає змогу організаціям масштабно розгорнути та керувати контейнерними програмами. Однією з ключових переваг Kubernetes є його здатність абстрагуватися від основної інфраструктури, що полегшує розробникам зосередження на створенні та розгортанні своїх програм. Проте мережа є критично важливим компонентом будь-якої розподіленої системи, включаючи Kubernetes, і проблеми з мережею можуть вплинути на доступність, надійність і продуктивність програм, розгорнутих на Kubernetes.

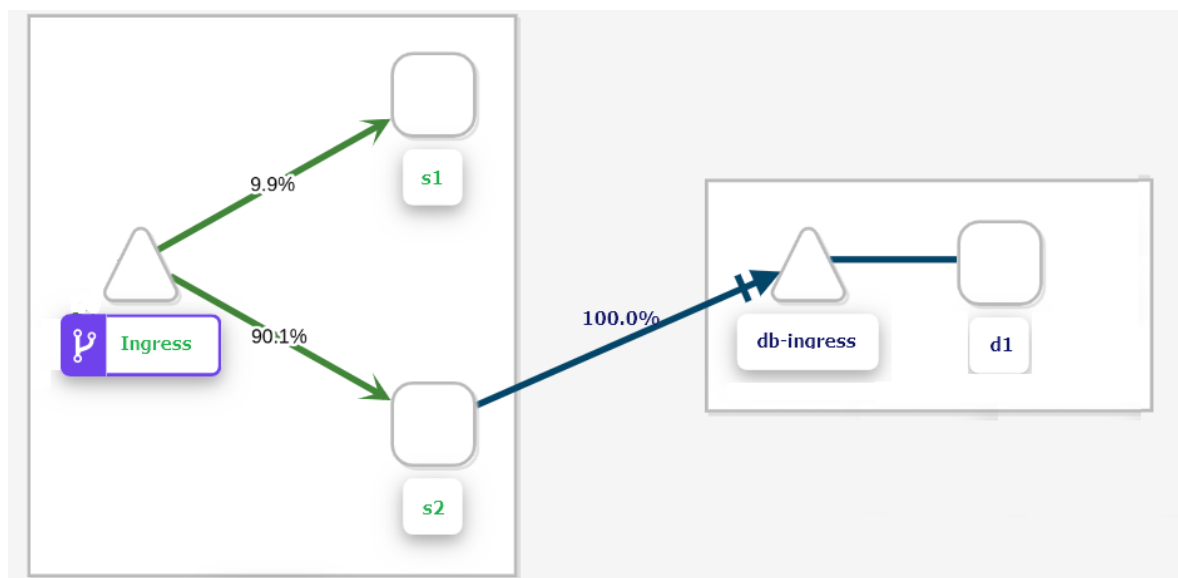
**Вплив мережеских проблем.** Проблеми з мережею в Kubernetes можуть проявлятися різними способами, включаючи проблеми з підключенням, проблеми з вирішенням DNS, збої балансувальника навантаження тощо. Ці проблеми можуть спричинити простій програми, повільний час відгуку та інші проблеми, які можуть негативно вплинути на роботу кінцевого користувача. Наприклад, якщо контейнер не може підключитися до бази даних через

проблеми з мережею, програма може вийти з ладу або перестати відповідати, що призведе до простою для кінцевих користувачів.

**Виявлення мережевих проблем.** Щоб виявити проблеми з мережею в Kubernetes, важливо відстежувати мережевий трафік між контейнерами та вузлами, а також справність мережевих компонентів, таких як DNS і балансувальників навантаження. Kubernetes надає кілька вбудованих інструментів для моніторингу мережі, зокрема сервер Kubernetes API, kube-proxy і core\_dns. Крім того, сторонні інструменти моніторингу, такі як Prometheus і Grafana, можна використовувати для моніторингу продуктивності мережі та виявлення проблем.

**Вирішення проблем мережі.** Вирішення мережевих проблем у Kubernetes може бути складним завданням, оскільки є багато потенційних причин і конфігурацій, які слід враховувати. В роботі ми розглянули одну з найкращих практик [2,3] з точки зору спорідненої технології, які можуть допомогти швидко виявити та вирішити проблеми з мережею. До неї належить використання сервісної мережі Istio, щоб керувати трафіком між контейнерами та вузлами.

Istio - це популярна сітка сервісу з відкритим кодом, яка надає комплексне рішення для керування трафіком між контейнерами та вузлами в Kubernetes. Istio використовує модель sidecar, де проксі (Envoy) розгортається поряд з кожним контейнером у модулі. Цей проксі-сервер перехоплює весь мережевий трафік до та з контейнера та дозволяє Istio керувати трафіком між контейнерами та вузлами.



**Рис. 1. Istio routing.**

Визначені правила маршрутизації достатньо прості для зміщення акценту на реалізацію (Рис. 1): після розгортання сервісів Istio (sidecar service) можна визначити правила маршрутизації за допомогою функцій керування трафіком

Istio. Це дає змогу контролювати, як трафік проходить між контейнерами та вузлами у вашому кластері Kubernetes. Наприклад, можна визначити правила, по яким маршрутизується трафік на основі конкретних заголовків, шляхів або агентів користувачів.

**Рекомендації.** Відстежуйте та захищайте трафік: Istio також надає функції моніторингу та безпеки, які дозволяють відстежувати мережевий трафік і захищати ваші програми Kubernetes. Ви можете використовувати Istio для застосування таких політик, як обмеження швидкості, контроль доступу та шифрування.

Масштабуйте Istio та керуйте ним. Нарешті, ви можете масштабувати Istio та керувати ним за допомогою таких інструментів Kubernetes, як `kubectl` і `Helm`. Ви можете використовувати ці інструменти для розгортання нових конфігурацій Istio, оновлення існуючих конфігурацій і керування такими компонентами Istio, як площина керування та площина даних.

Загалом, Istio надає потужне та гнучке рішення для керування трафіком між контейнерами та вузлами в Kubernetes. Використовуючи Istio, ви можете переконатися, що ваші програми Kubernetes надійні, продуктивні та безпечні.

**Висновок.** Мережа є критично важливим компонентом будь-якої розподіленої системи, включно з Kubernetes. Проблеми з мережею можуть вплинути на доступність, надійність і продуктивність програм, розгорнутих у Kubernetes. Тому важливо швидко виявляти та вирішувати ці проблеми, використовуючи передові практики, такі як моніторинг мережевого трафіку, використання сітки служби та обмеження трафіку за допомогою мережевих політик. Дотримуючись цих найкращих практик, організації можуть переконатися, що їхні програми надійні та продуктивні, навіть якщо виникають проблеми з мережею.

### Література

1. Xie, Z., Yuan, X., Wang, J., Ju, S. "Encoder-Decoder Network with Cross-Match Mechanism for Answer Selection" In: Sun, Chinese Computational Linguistics. CCL 2019. Lecture Notes in Computer Science(), vol 11856. Springer, Cham. [https://doi.org/10.1007/978-3-030-32381-3\\_6](https://doi.org/10.1007/978-3-030-32381-3_6)
2. Romanov O, Mankivskyi V. Optimal traffic distribution based on the sectoral model of loading network elements, IEEE Problems of Infocommunications. Science and Technology Trans. Multimedia, 2019, pp. 210–219.
3. Mankivskyi Volodymyr and Pedko Andrii Concept of edge computing in mobile operator network // Modern challenges in telecommunications, vol. 14, 2020, pp. 51–54.
4. M. Xiao et al., Millimeter Wave Communications for Future Mobile Networks // IEEE JSAC, vol. 35, no. 9, Sept. 2017, pp.1909–35.
5. Oleksandr I Romanov, Mykola M Nesterenko, Nadiia O Fesokha and Volodymyr B Mankivskyi. Evaluation of productivity virtualization technologies of switching equipment telecommunications networks. // National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 2020 - pp.: 132 – 138