

КЕРУВАННЯ ОБЧИСЛЮВАЛЬНИМИ РЕСУРСАМИ КОРИСТУВАЦЬКИХ ПРИСТРОЇВ У СКЛАДІ КЛАСТЕРУ KUBERNETES

Смаглюк В.О., Алексєєв М.О.

*Навчально-науковий інститут телекомунікаційних
систем КПІ ім. Ігоря Сікорського, Україна
E-mail: amxleclerk@gmail.com, alexeyev@its.kpi.ua*

MANAGEMENT OF COMPUTING RESOURCES OF USER DEVICES IN A KUBERNETES CLUSTER

The work is devoted to the topical issue of the effective use of computing resources in systems with a dynamic distributed architecture that use personal or public computing devices for the needs of distributed computing, specifically to the task of effectively managing the life cycle of working nodes of clusters of containerized applications. Unlike the known existing solutions, the systems considered in the work have a relatively short-term and non-deterministic lifetime of their computing agents and therefore must have specialized functionality for automatically connecting and disconnecting computing agents from the cluster, as far as the process of joining a new working node to a system takes time and requires a certain qualification and level of authorization. The approach proposed in the work is based on the designing of the process of managing working nodes in a form of Service Discovery in the Kubernetes cluster environment. The advantage of the proposed approach is the complete automation of the process based on prepared programs that include operations to check the user's operating environment, install the necessary software to connect to the computer system, or uninstall components and disable the agent at the user's request. The work considers the experimental environment, investigates the system's operation in possible use cases and justifies the effectiveness of the approach.

Для підвищення ефективності використання незайнятих ресурсів обчислювальних вузлів та зниження загального рівня їх простоювання і водночас, для вирішення нагальних завдань з розміщення та виконання програмних компонентів корпоративних інформаційно-обчислювальних систем у попередній роботі було вирішено створювати кластери із залученням персональних пристроїв користувачів, а також парку комп'ютерної техніки та загальнодоступних обчислювальних пристроїв у корпоративній мережі. Також було прийняте рішення щодо використання контейнеризації прикладних програм та програмної технології Kubernetes (далі – k8s) для оркестрації робочих навантажень на їх основі, оскільки цей підхід дозволяє використовувати апаратні і програмні обчислювальні ресурси різні за походженням і характеристиками, а також дає змогу підключати і від'єднувати їх у процесі експлуатації [1]. Вузол K8s (node, “нода”) або робочий вузол у даному контексті це фізична або віртуальна машина, яка бере участь у кластері

K8s, і яку можна використовувати для запуску робочих навантажень (pod, “под”) у вигляді контейнеризованих програм. Проте, процес долучення та від’єднання робочих вузлів потребує виконання певних операцій у ручному режимі. При невеликій кількості потенційних клієнтів адміністратор K8s кластеру може сам проконтролювати переходи між станами та виконувати конфігураційні задачі, а саме: ініціалізацію програмного забезпечення, аналіз відповідності наданого пристрою до вимог цільової системи або видалення його. Проте, зі збільшенням кількості робочих пристроїв N буде зростати операційне навантаження P на адміністративний вузол за певним експоненціальним законом $P = \exp(N)$. Якщо додатково врахувати, що система є динамічною, тоді час життя робочих пристроїв не обов’язково буде збільшуватися за лінійним законом і треба відслідковувати та вести звітність про кожен з клієнтів: у якому стані вони знаходяться, а також які дії потрібно виконати, щоб безпечно перевести пристрій з одного в інший стан.

Враховуючи це, в рамках роботи пропонується ведення актуальної таблиці станів робочих пристроїв у вигляді Service Discovery, де у ролі сервіса виступають ноди, з використанням бази даних etcd, яка є розподіленим сховищем у форматі ключ-значення і зазвичай поставляється разом із K8s. Цей компонент зберігає всю інформацію про робочі навантаження, які представлені у системі, враховуючи облік нод. За допомогою інтерфейсу K8s API ця інформація періодично оновлюється, щоб відповідати реальному станові системи. Якщо у etcd немає інформації про певну з нод, тоді вважається, що вона перебуває у стані готовності до ініціалізації. Навіть у випадку, коли пристрій користувача було вилучено із кластеру, але програмні компоненти досі встановлено на його машині, він знову має пройти етап ініціалізації запиту на приєднання до розподільнювальної системи. Для автоматизації актуалізації інформації про робочі навантаження з використанням etcd та K8s API у роботі був створений автоматизований пакет (скрипт), який перевіряє систему на відповідність вимог цільової кластерної мережі, інсталує необхідні програмні пакети та приєднає клієнтський пристрій до K8s або навпаки здійснює від’єднання робочих вузлів (див. рис. 1).

Наведені сценарії контролюються створеним у роботі програмним кодом, що отримує клієнт із відповідної адреси. Проте, може бути ситуація, коли було втрачено зв’язок із робочим вузлом. У такому випадку, робочий вузол переводиться у стан «неактивний режим». Коли вузол вимикається або виходить з ладу, він переходить у стан NotReady, тобто його не можна використовувати для запуску подів. Після цього всі поди, запущені на вузлі, стають недоступними. Робочі навантаження залишаються при цьому закріпленими за проблемним вузлом. Якщо K8s не може зв’язатися з вузлом,

він чекає за замовчуванням 40 секунд, а потім встановлює статус вузла на Unknown. Якщо вузол залишається недоступним відбувається ініціювання «виселення» за допомогою API для всіх подів на недоступному вузлі. За замовчуванням контролер вузла чекає 5 хвилин між позначенням вузла як невідомого та поданням першого запиту на вилучення [3]. Якщо недоступний вузол не повернувся до мережі, його буде видалено із внутрішньої бази, що у термінах даної роботи означає переведення у стан “готовності до ініціалізації”.

У випадку повторного приєднання втраченого робочого вузла після того, як його було видалено з K8s кластеру, клієнт повинен повністю пройти процес ініціалізації описаний вище (див. рис. 1,а).

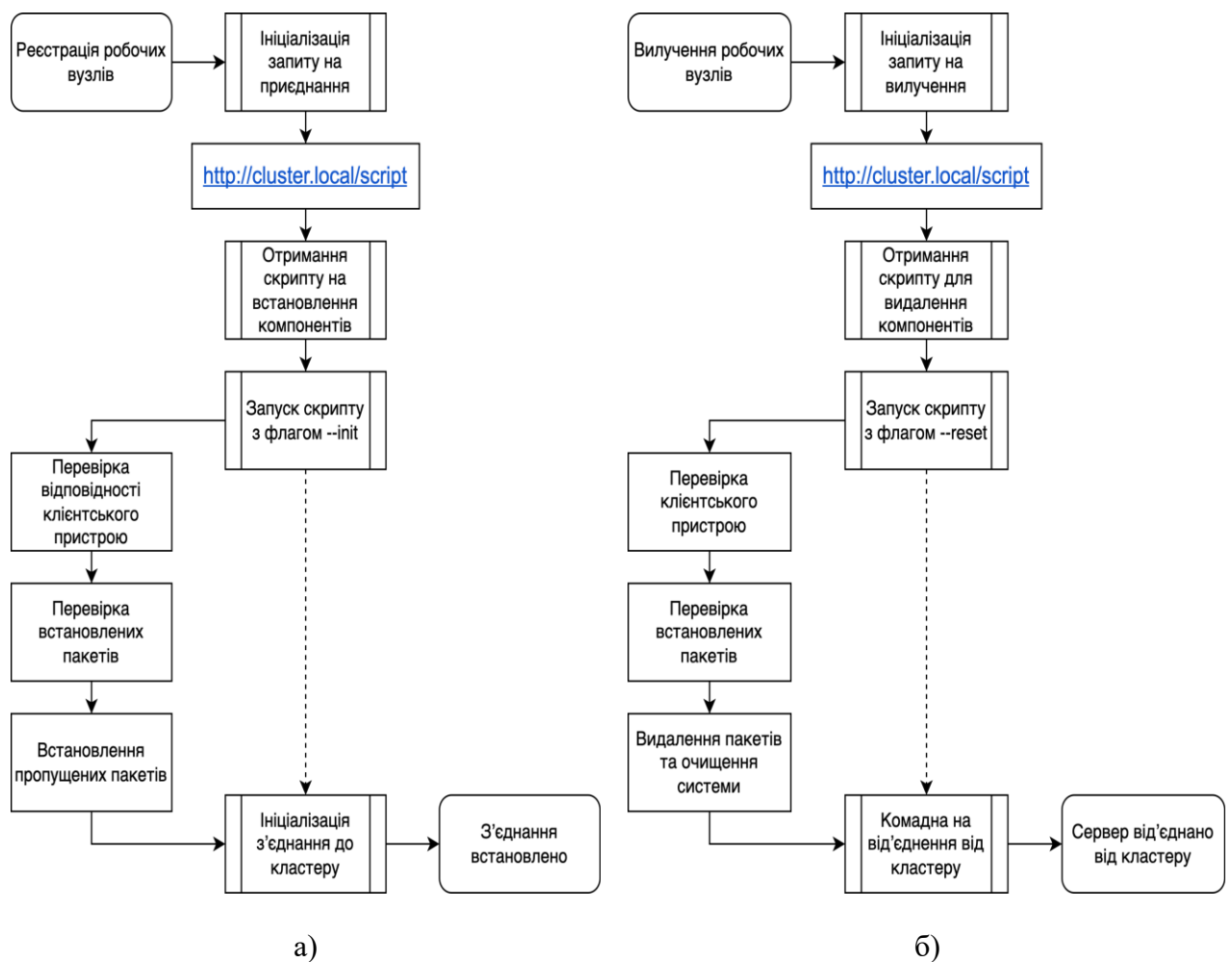


Рис. 1. Блок-схема процесу реєстрації а) від'єднання б) робочих вузлів

У роботі був створений експериментальний макет (див. рис. 2), який містить один головний вузол, два робочі статичні вузли, один динамічний вузол (у стані “активний режим”), потенційний динамічний вузол (у стані “готовність до ініціалізації”) та точку доступу (в рамках корпоративної мережі) з виконувальним програмним кодом. Результат роботи розробленого ПЗ показаний на рис. 2.

```

bash-3.2$ sh ./script.sh --init
# System information
Target OS: Ubuntu 22.04.1 LTS
System resources check: passed

# Kubernetes information
Target Kubernetes version: 1.23.13
Installation kubernetes packages: kubeadm, kubelet, kubectl
Successfully installed kubernetes packages.

# Docker information
Target Docker version: 20.10.8
Installation docker packages: docker-ce, docker-ce-cli, containerd.io
Successfully installed docker packages.

# Joining the cluster
Successfully joined the cluster.

```

a)

NAME	STATUS	ROLES	AGE	VERSION
master.static.com	Ready	master	12h	v1.23
node1.static.com	Ready	compute	12h	v1.23
node2.static.com	Ready	compute	12h	v1.23
node3.dynamic.com	Ready	compute	10h	v1.23



NAME	STATUS	ROLES	AGE	VERSION
master.static.com	Ready	master	13h	v1.23
node1.static.com	Ready	compute	13h	v1.23
node2.static.com	Ready	compute	13h	v1.23
node3.dynamic.com	Ready	compute	11h	v1.23
node4.dynamic.com	Ready	compute	6m	v1.23

б)

Рис. 2. Реєстрація нового вузла а) та оновлений перелік б) робочих вузлів.

Висновки.

1. Завдяки програмному коду вдалося мінімізувати оперативне навантаження на адміністратора K8s кластеру, стосовно управління життєвим циклом робочих вузлів та їх підготовкою до приєднання у систему.

2. Запропонований метод дозволяє автоматизувати процес реєстрації та вилучення робочих вузлів до K8s кластеру, що дає можливість побудувати динамічну обчислювальну систему з робочими вузлами з короткостроковим і недетермінованим часом життя своїх обчислювальних агентів.

Література

1. Смаглюк В.О., Алексеев М.О. Керування клієнтськими обчислювальними ресурсами з динамічним життєвим циклом у корпоративній мережі. ISSN 2788-5518. Інфокомунікаційні та комп'ютерні технології: Науковий журнал. № 2(04) 2022. К.: Університет "Україна", 2022. - с.134-142
2. Etcd Documentation. "What is etcd?": Онлайн: <https://etcd.io/docs/v3.5/faq/#what-is-etcd>: 25.03.2022.
3. Kubernetes Docs. Kubernetes Nodes: Conditions: Онлайн: <https://kubernetes.io/docs/concepts/architecture/nodes/#condition>: 22.03.2022 17:43.