

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Інформаційна технологія побудови
обчислювальної платформи для розміщення
і підтримки серверних додатків з
мікросервісною архітектурою

Виконав: студент групи ІКН-19м
Ткачук А.С.
Керівник д.т.н. проф.
Яровий А.А.

Мета та задачі роботи

- **Метою** магістерської кваліфікаційної роботи є зменшення показника часу, який витрачається на розгортання нових додатків або їх підсистем та версій на платформі.
- Для досягнення мети розробки необхідно виконати наступні **задачі**:
 - здійснити обґрунтування доцільності розробки інформаційної технології побудови обчислювальної платформи для розміщення і підтримки серверних додатків з мікросервісною архітектурою;
 - здійснити проектування обчислювальної платформи для розміщення і підтримки серверних додатків з мікросервісною архітектурою;
 - обґрунтувати вибір програмного інструментарію для реалізації інформаційної технології обчислювальної платформи для розміщення і підтримки серверних додатків з мікросервісною архітектурою;
 - здійснити програмну реалізацію інформаційної технології обчислювальної платформи для розміщення і підтримки серверних додатків з мікросервісною архітектурою;
 - здійснити тестування інформаційної технології обчислювальної платформи для розміщення і підтримки серверних додатків з мікросервісною архітектурою.

Об'єкт, предмет та методи дослідження

Об'єктом дослідження є процес підтримки роботи серверних додатків з мікросервісною архітектурою.

Предметом дослідження є програмні засоби побудови обчислювальної платформи для розміщення і підтримки серверних додатків з мікросервісною архітектурою.

Для досягнення мети дослідження застосовувалися методи моделювання UML, модель представлення хмарних обчислень PaaS, концепція керування інфраструктурою IaaS, підхід до адміністрування DevOps, розглядались патерни проектування мікросервісів, такі як Aggregator, Proxy, Chained, Shared Data.

Актуальність

Із активним розвитком комп'ютерних технологій не менш активно розвивається і Інтернет.

Все більше додатків створюється на основі поєднання двох архітектур. А саме, на основі клієнт-серверної архітектури, для зменшення навантаження на пристрої користувачів, та мікросервісної архітектури для спрощення розробки великих додатків.

При використанні клієнт-серверної архітектури безперечно виникає потреба у наявності серверу, на якому буде розміщено серверну частину додатку. Однак, з використанням мікросервісної архітектури сервісний додаток може бути розділений на декілька логічних частин, кожна з яких зазвичай розміщується на різних серверах або і взагалі на групі серверів.

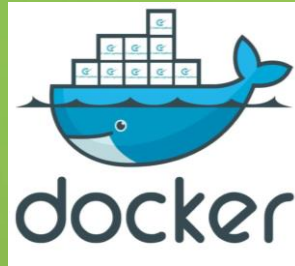
Проте, частина розробників може просто не мати власного серверу. І навіть просто наявності серверу чи групи серверів недостатньо. Самі сервери потрібно вміти налаштувати, потрібно з'єднати їх в мережу, налаштувати процеси автоматизації роботи з серверами і так далі.

Саме для випадків, коли у команди розробників програмного забезпечення недостатньо знань та вмінь для самостійного проведення таких робіт були створені обчислювальні платформи для розміщення і підтримки серверних додатків.

Вибір подібної платформи дуже важливий, так як він буде впливати на ціну, якість та надійність роботи, швидкість оновлення та зручність розробки додатку на протязі значної частини життєвого циклу програмного забезпечення. Обчислювальні платформи для розміщення і підтримки серверних додатків дозволяють значно спростити процес розгортання, оновлення та моніторингу додатку на сервері для розробників.

Програмний інструментарій що застосовувався при розробці підсистем платформи

Docker



Kubernetes



Ansible



Jenkins

Helm



Prometheus

Grafana



Elasticsearch

Logstash



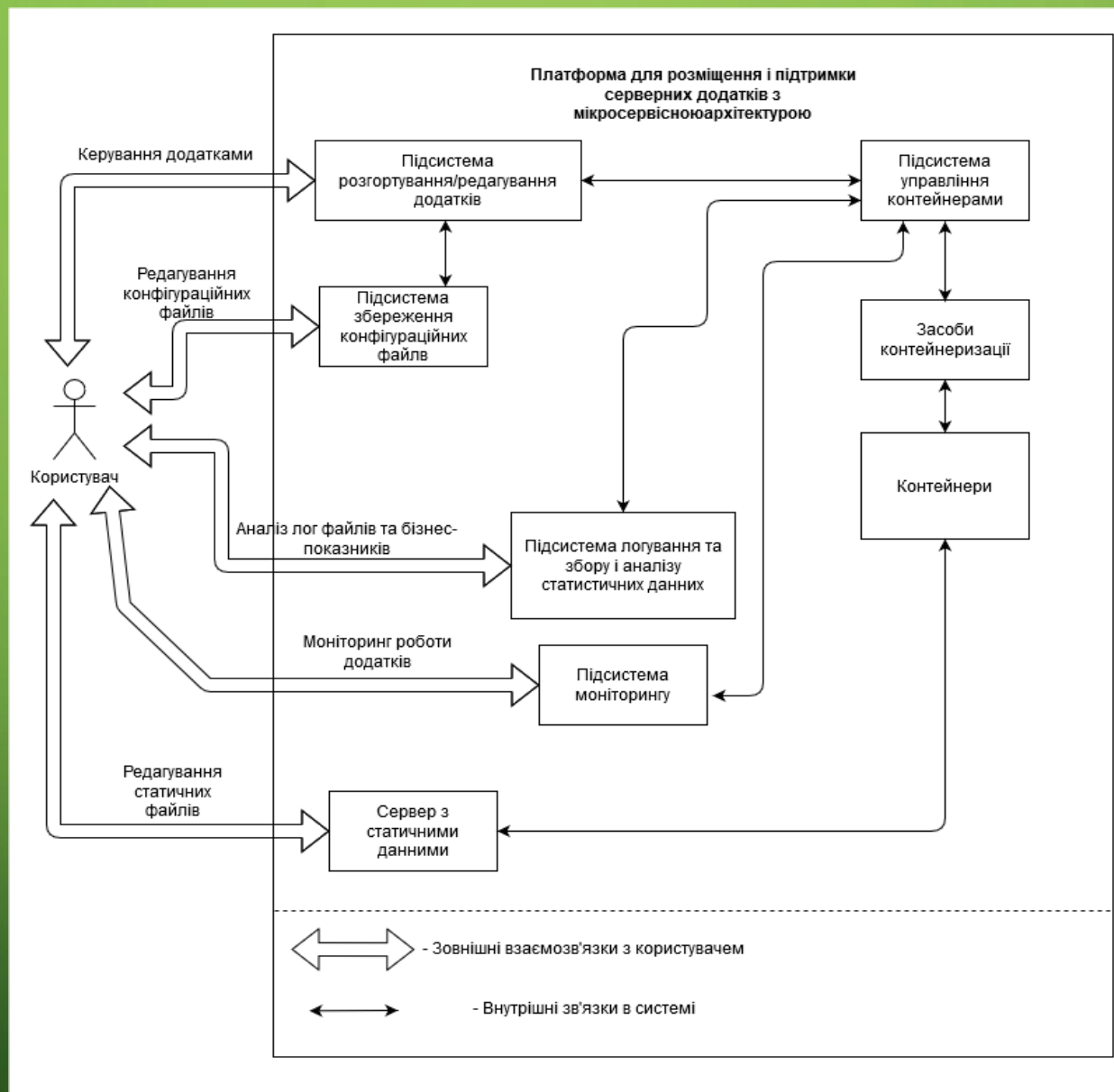
Kibana

Скріншоти аналогів

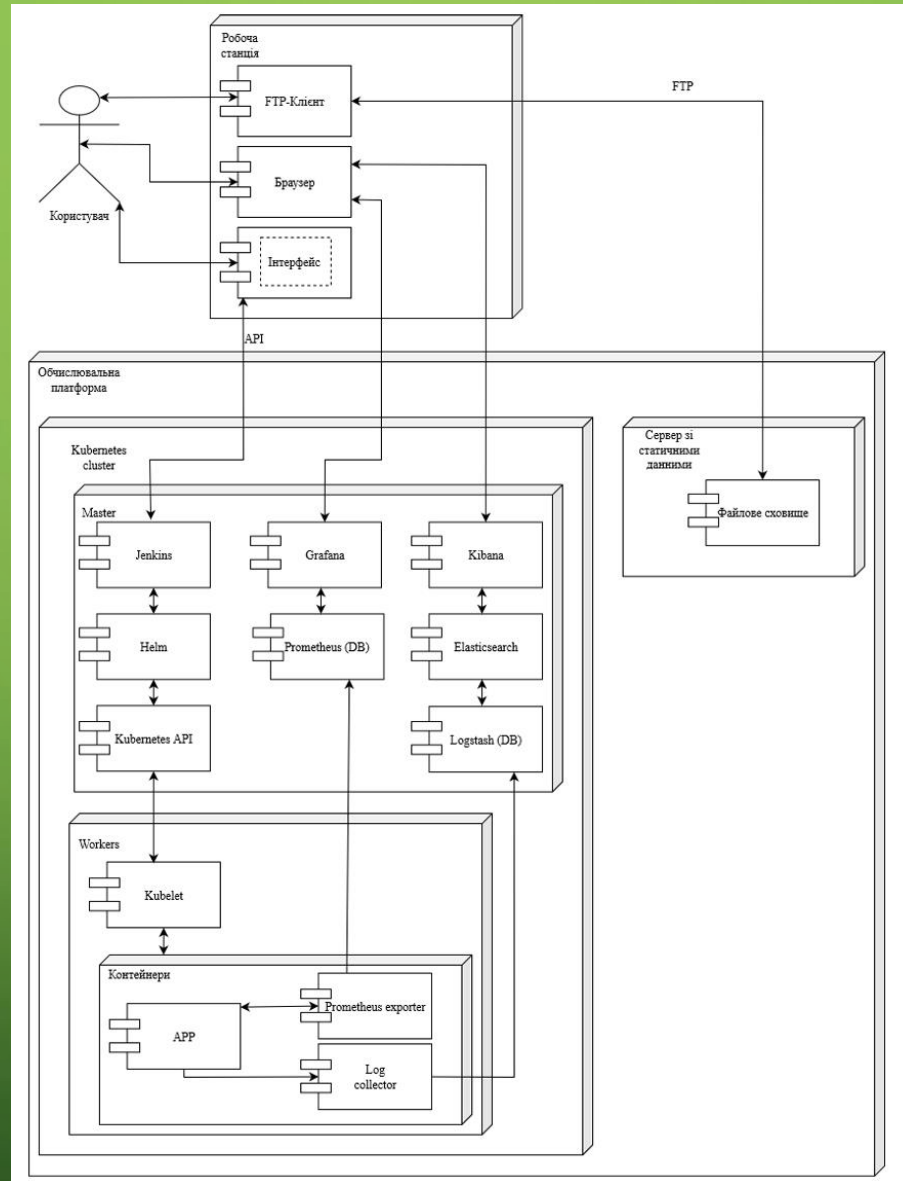
The screenshot shows the Red Hat OpenShift OperatorHub interface. The top navigation bar includes the OpenShift logo, a user profile for 'Jonny Appleseed', and a 'Project: default' dropdown. The left sidebar contains a navigation menu with categories like Home, Catalog, OperatorHub, Workloads, Networking, Storage, Builds, Monitoring, Compute, and Administration. The main content area is titled 'OperatorHub' and displays a grid of 34 operators. A filter sidebar on the left allows users to search by keyword and filter by install state, provider type, and provider. The operators shown include AMO Streams, Aqua Security Operator, Automation Broker Operator, Camel-K Operator, Cluster Logging, CockroachDB, Community Jaeger Operator, Couchbase Operator, and Crunchy PostgreSQL Enterprise.

The screenshot shows the OpenShift console interface for an application named 'test-php-tk'. The top navigation bar includes a search bar, user profile, and breadcrumb navigation. The main content area displays the application's status and configuration options. The 'Installed add-ons' section shows a price of '\$0.00/month' and a 'Configure Add-ons' link. The 'Latest activity' section shows 'All Activity'. The 'Dyna formation' section shows a price of '\$0.00/month' and a 'Configure Dynos' link. The 'Collaborator activity' section shows a 'Manage Access' link. The main content area contains three messages: 'There are no add-ons for this app', 'This app has no process types yet', and 'There is no recent activity on this app'.

Загальна структурна схема платформи



UML-діаграма компонентів



UML-діаграма прецедентів

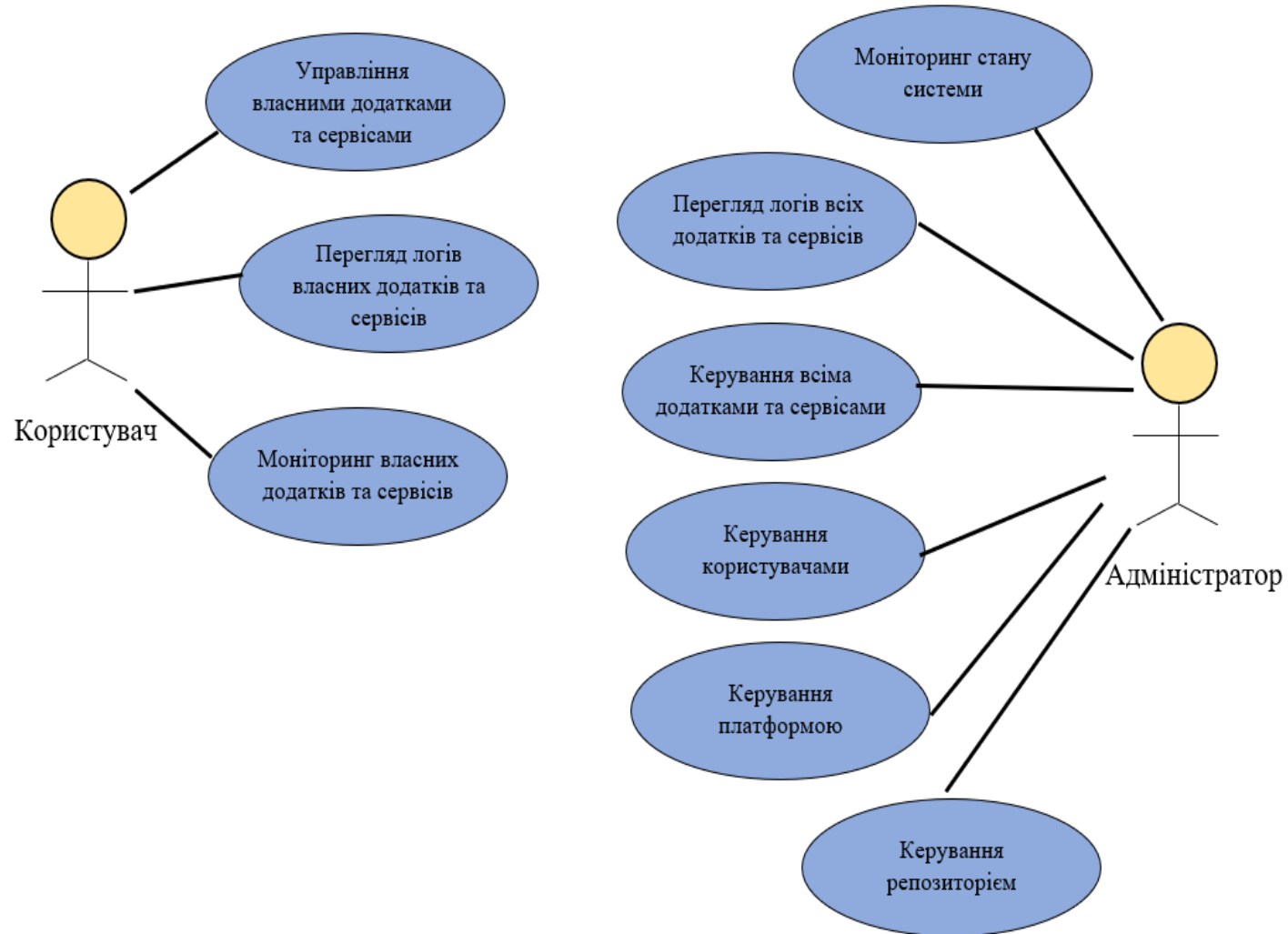


Схема алгоритму роботи плейбуку для встановлення кластера Kubernetes

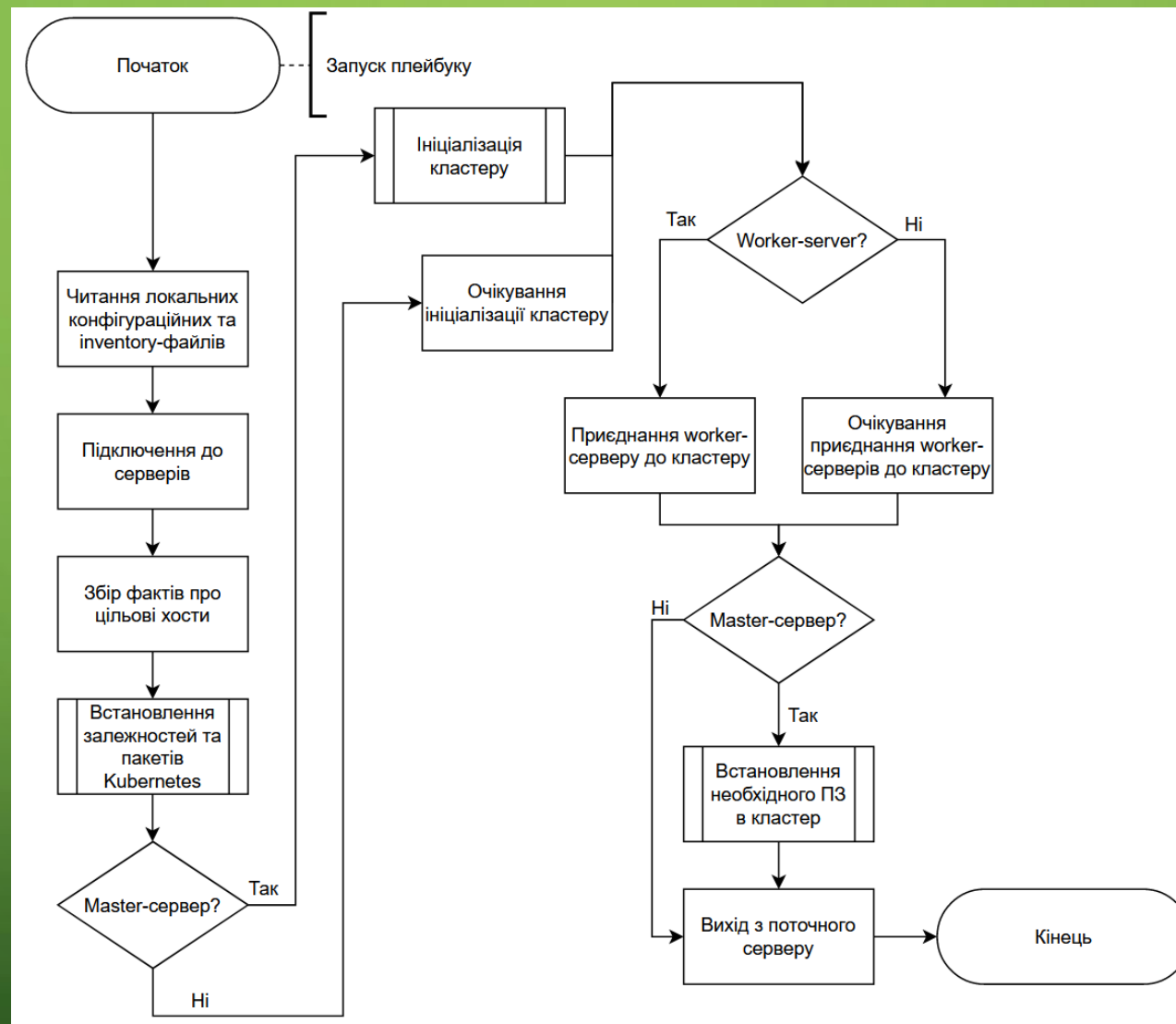
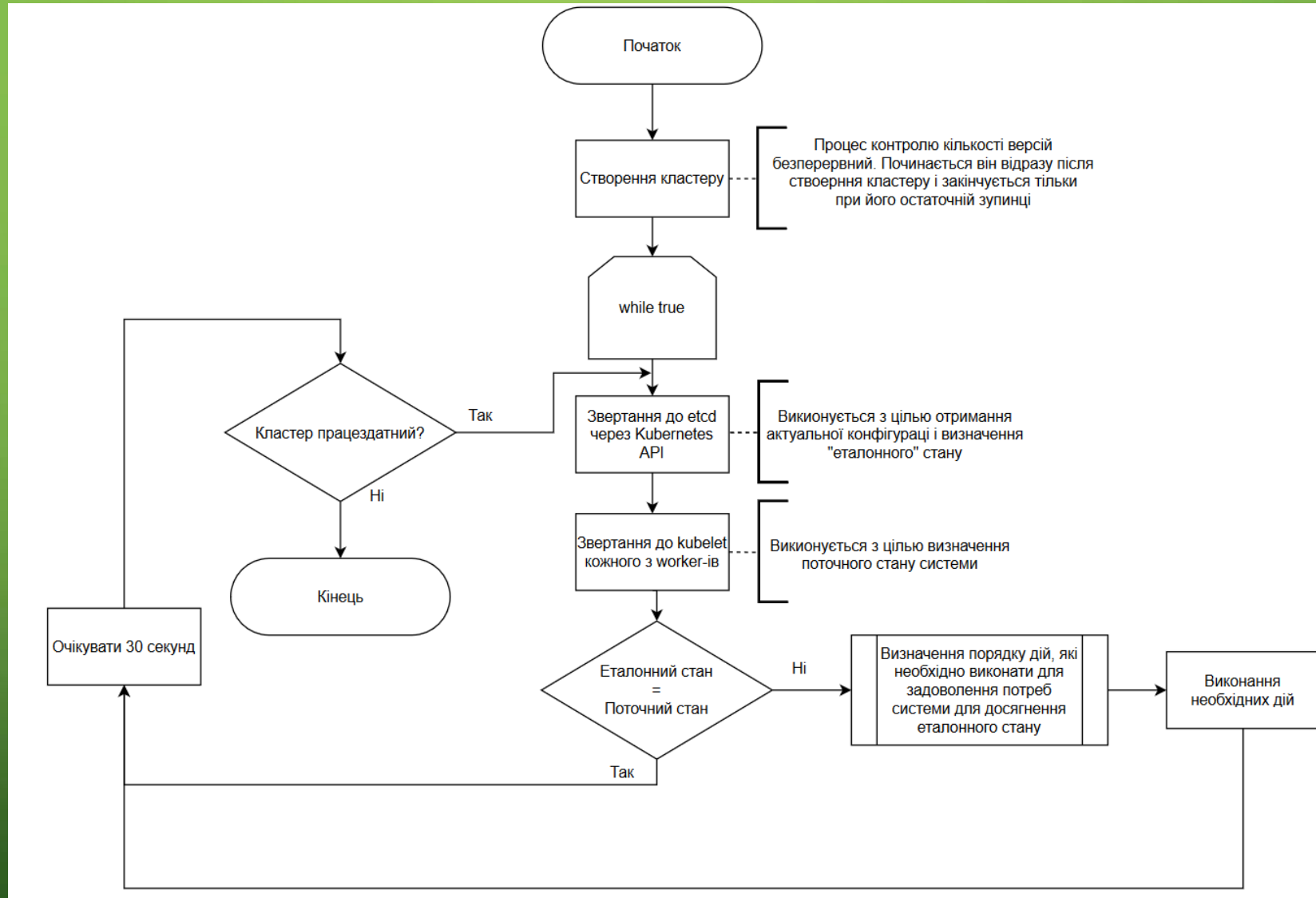


Схема алгоритму контролю реплік додатків Kubernetes-ом



Результат тестування

Спроба, №	Власна розробка	Платформа «Негоки»
1	21 с.	24 с.
2	19 с.	26 с.
3	19 с.	26 с.
4	16 с.	23 с.
5	15 с.	25 с.
6	20 с.	25 с.
7	17 с.	28 с.
8	17 с.	24 с.
9	16 с.	26 с.
10	18 с.	26 с.
11	20 с.	27 с.
12	17 с.	24 с.
13	20 с.	23 с.
14	21 с.	25 с.
15	17 с.	25 с.
16	20 с.	27 с.
17	16 с.	26 с.
18	15 с.	24 с.
19	18 с.	24 с.
20	18 с.	22 с.
21	17 с.	24 с.
22	19 с.	25 с.
23	17 с.	26 с.
24	17 с.	24 с.
25	18 с.	23 с.
Середнє значення	17,92 с.	24,88 с.



За результатами тестування, час що затрачується на розгортання додатку на платформі зменшився на 28% у порівнянні з аналогом

Економічна частина

- Прибуток: 1 204 165,35
- Абсолютний ефект: 1 081 465,35 (грн.)
- Відносний ефект: 76%
- Термін окупності: 1,31 року

Апробація та результати

- Результати досліджень було апробовано на XLIX науково-технічній конференції професорсько-викладацького складу, співробітників та студентів Вінницького національного технічного університету у 2020р.
- За основними результатами досліджень опубліковано одну публікацію та подано заявку про реєстрацію авторського права на твір (комп'ютерну програму)

Висновки

- В ході виконання магістерської кваліфікаційної роботи розроблено інформаційну побудову обчислювальної платформи для розміщення і підтримки серверних додатків з мікросервісною архітектурою.
- Під час аналізу архітектури серверних додатків та платформи для розміщення і підтримки серверних додатків з мікросервісною архітектурою було розглянуто наявні архітектури побудови серверних додатків, виконано їх порівняння.
- Проаналізовано платформи для розміщення і підтримки розподілених серверних додатків з мікросервісною архітектурою. Розглянуто і проаналізовано системи-аналоги і визначено їх переваги та недоліки.
- Виконано аналіз особливостей платформ для розміщення і підтримки серверних додатків з мікросервісною архітектурою та проведено аналіз інструментарію, що використовується для реалізації підсистем платформи.

Висновки

- За результатами аналізу було обрано наступний програмний інструментарій: Docker, Kubernetes, Ansible, Jenkins, Elastic Stack, Prometheus, Grafana, BitBucket, Nginx, ProFTPd. Розглянуто специфічні аспекти побудови платформ на базі оркестратора Kubernetes.
- Визначено структурну організацію інформаційної технології, розроблено алгоритми роботи основних вузлів інформаційної технології.
- Програмно реалізовано інформаційну технологію. Здійснено тестування програмної реалізації інформаційної технології, під час якого розроблена інформаційна технологія працювала відповідно до поставлених перед нею задач. Виявлено, що нова розробка дозволила зменшити час розгортання додатку, у порівнянні з аналогом, на 28%.
- Виконано економічне обґрунтування доцільності розробки.

The image features a dark green gradient background. In the four corners, there are decorative elements consisting of light green lines that resemble circuit traces or a network diagram, with small circles at the end of the lines. The text "Дякую за увагу!" is centered in the middle of the image in a white, sans-serif font.

Дякую за увагу!