Мета

Планування і розгортання високодоступного кластеру Kubernetes використовуючи наявну локальну інфраструктуру, а також налаштування і інтеграція необхідних сервісів для роботи програм телемедичних досліджень.

[Slide 3]

Сучасна телемедицина включає в себе різноманітні сфери. До таких сфер відносяться, наприклад дистанційні консультації. Однієї із головних сфер телемедицини є телемедичні моніторингові дослідження, задача яких полягає у вчасному виявленні та попередженні хвороб. У випадках виявлення причини захворювання такі дослідження дозволять терміново вжити заходи із відтермінування переходу хвороби в активний стан, а також забезпечити персоніфікацію необхідних медичних послуг. Для підтримки системи телемедичних досліджень необхідно розгорнути таку інфраструктуру, яка би дозволяла би забезпечувати високу доступність, а також мати значні обчислювальні потужності.

Для вирішення цієї задачі якнайкраще підходить кластер серверів, який може бути розгорнутий у хмарному середовищі, але, за наявності локальної інфраструктури, а також персоналу, який буде підтримувати і налаштовувати, то значно дешевше буде розгортати саме локально.

[Slide 4]

Мета роботи була розбита на дві окремі задачі.

Узагальнена задача включає створення та конфігурацію ВМ (або ж налаштування реальної інфраструктури, за наявності), встановлення компонентів керування кластером, ініціалізація кластера та робочих вузлів, встановлення CNI та встановлення Container runtime.

Задачі розгортання сервісів для телемедичних задач, є специфікою, яка формує телемедичну систему, і включає дії, які необхідно зробити для підтримки саме системи телемедичних досліджень. Ця задача включає встановлення балансувальника навантаження на кластер, інтеграцію об’єктного сховища даних пацієнтів і результати досліджень, а також налаштування системи моніторингу стану кластера. Основною задачею, яка постає перед адміністратором, є налаштування взаємозв’язку між інтегрованими сервісами та розгорнутим телемедичним додатком.

[Slide 5]

Перед розгортанням був проведений аналіз існуючих підходів до розгортання кластерів. Наприклад, кластер, який керується ПЗ Kubernetes, використовує ВО Medrid, яка розгорнула розподілену інфраструктуру для виконання задачі обслуговування сервісу теле-ЕКГ досліджень. Іншим прикладом може послугувати компанія Spotify, яка у 2018р. перейшла із підходу віртуалізації всіх ресурсів на підхід кластеризації для підтримки власного сервісу прослуховування музики. Даний стрімінговий сервіс приймає загалом близько 10 мільйонів запитів на секунду.

Крім того, кластеризацію використовують у своїй роботі такі компанії, як NYT, Adidas та AirBnb.

[Slide6]  
 На даному слайді приведена ілюстрація архітектури побудованої системи. Розгорнутий кластер включає 3 вузла, один із яких вузол керування, Для ПЗ оркестрування, або ж керування кластером було обрано Kubernetes. Під час цієї кваліфікаційної роботі було розглянуто два найбільш популярних оркестратори – K8s та Docker Swarm. Порівняння було наведено у пояснювальній записці. Kubernetes був обраний через те, що надає значно ширший обсяг функцій ніж Docker Swarm, таких як доступність самовідновлення та автоматичного масштабування.

Перелік інших використаних технологій:

CNI – Flannel

Container runtime – CRI-O

Система моніторингу – стек Prometheus/Grafana

LB – MetalLB

Об’єктне сховище – MinIO

[Slide 7]

На цьому слайді зображена архітектура кластера Kubernetes. Серед основних компонентів вузла керування можна виділити etcd сховище конфігурації, API-server та менеджер контролерів.

До компонентів робочого вузла відноситься Kubelet та мережевий проксі. Які, у зв’язці з container runtime відповідають за підтримку розгорнутих на вузлі сервісів.

[Slide 8]

На цьому слайді наведено принцип мережевої складової кластера. За відсутності реальної інфраструктури було прийняте рішення використати середовище VirtualBox для моделювання кластера, який передбачається розгорнути на реальній інфраструктурі на базі кафедри в рамках наукової роботи. Враховуючи особливості Vbox розгортання, кожному вузлу було виділено два віртуальних інтерфейси. Перший, Bridged adapter відповідає за зв’язок вузлів і зовнішньою мережею (домашньою), в той же час, використовуючи NAT adapters створюється внутрішня мережа для передачі запитів, задач для вузлів, ітд. Крім того, CNI Flannel створює власну мережу для взаємодії вузлів із розгорнутими на ньому контейнерами.

[Video]