1. Kubernetes є одним з найбільш популярних засобів для

* розгортання та оркестрації контейнеризованих додатків
* декомпозиції контейнеризованих додатків
* реєстрації контейнеризованих додатків
* згортання та анігіляції контейнеризованих додатків
* форматування контейнеризованих додатків

2. Kubernetes базується на технології контейнерів, що дозволяють забезпечити

* ізольоване середовище для додатків
* нестабільне середовище для додатків
* аморфне середовище для додатків
* неізольоване середовище для додатків
* напівізольоване середовище для додатків

3. Kubernetes працює з об'єктами, такими як

* Pod
* Kod
* Lod
* Flood
* Temp

4. Kubernetes працює з об'єктами, такими як

* Service
* DataSet
* Logs
* Kood
* Pood

5. Kubernetes працює з об'єктами, такими як

* Deployment
* Realize
* Setting
* DataSet
* Docker

6. Kubernetes працює з об'єктами, такими як

* ReplicaSet
* DataSet
* MirrorSet
* CodeSet
* DBSet

7. Кожен Pod містить

* один або кілька контейнерів
* один контейнер
* два контейнери
* не менше двох контейнерів
* три контейнери

8. Кожен Pod містить

* спільний віртуальний мережевий простір
* незалежний віртуальний мережевий простір
* роздільний мережевий простір
* внутрішній віртуальний мережевий простір
* спільний мережевий простір

9. Кожен Pod має власну IP-адресу та може бути масштабований за допомогою

* ReplicaSet
* Service
* Docker
* Deployment
* DataSet

10. Service дозволяє створювати стабільний доступ до

* Групи Pod
* DataSet
* Docker
* Pod
* Різних за функціями Pod

12. Кожен ReplicaSet має визначену кількість

* Pod
* Контейнерів
* DataSet
* Service
* Job

13. Kubernetes працює з об'єктами, такими як

* Job
* Realize
* Setting
* DataSet
* Docker

14. Docker Image це

* образ контейнера, який містить код додатку та всі його залежності
* об'єкт, який дозволяє розгортати та оновлювати контейнеризовані додатки. Описує кількість копій додатку, що мають бути запущені, та спосіб їх розгортання
* механізм, який дозволяє забезпечити доступ до запущених копій додатку, незалежно від того, на якому вузлі вони запущені. Крім того, може забезпечувати балансування навантаження між копіями додатку
* механізм, який дозволяє забезпечити зовнішній доступ до додатку, використовуючи правила маршрутизації
* дозволяє зберігати конфігураційні дані та секретні дані, такі як паролі, ключі доступу та інші

15. Deployment це

* об'єкт, який дозволяє розгортати та оновлювати контейнеризовані додатки. Описує кількість копій додатку, що мають бути запущені, та спосіб їх розгортання
* образ контейнера, який містить код додатку та всі його залежності
* механізм, який дозволяє забезпечити доступ до запущених копій додатку, незалежно від того, на якому вузлі вони запущені. Крім того, може забезпечувати балансування навантаження між копіями додатку
* механізм, який дозволяє забезпечити зовнішній доступ до додатку, використовуючи правила маршрутизації
* дозволяє зберігати конфігураційні дані та секретні дані, такі як паролі, ключі доступу та інші

16. Service це

* механізм, який дозволяє забезпечити доступ до запущених копій додатку, незалежно від того, на якому вузлі вони запущені. Крім того, може забезпечувати балансування навантаження між копіями додатку
* об'єкт, який дозволяє розгортати та оновлювати контейнеризовані додатки. Описує кількість копій додатку, що мають бути запущені, та спосіб їх розгортання
* образ контейнера, який містить код додатку та всі його залежності
* механізм, який дозволяє забезпечити зовнішній доступ до додатку, використовуючи правила маршрутизації
* дозволяє зберігати конфігураційні дані та секретні дані, такі як паролі, ключі доступу та інші

17. Service це

* механізм, який дозволяє забезпечити зовнішній доступ до додатку, використовуючи правила маршрутизації
* механізм, який дозволяє забезпечити доступ до запущених копій додатку, незалежно від того, на якому вузлі вони запущені. Крім того, може забезпечувати балансування навантаження між копіями додатку
* об'єкт, який дозволяє розгортати та оновлювати контейнеризовані додатки. Описує кількість копій додатку, що мають бути запущені, та спосіб їх розгортання
* образ контейнера, який містить код додатку та всі його залежності
* дозволяє зберігати конфігураційні дані та секретні дані, такі як паролі, ключі доступу та інші

18. Service може забезпечувати балансування навантаження між

* Копіями додатку
* Групою Pod
* Різними Job
* контейнерами
* сервісами

19. Створення Ingress - механізму, який дозволяє забезпечити зовнішній доступ до додатку, використовуючи правила маршрутизації. Для цього необхідно визначити правила, які вказують, на який сервіс та порт має бути виконано запит

* із зовнішньої мережі
* із внутрішньої мережі
* із локальної мережі
* із глобальної мережі
* нема правильної відповіді

20. Kubernetes працює з об'єктами, такими як

* HPA
* PRF
* JOB
* POD
* JJK