

УДК 004.03

## ТЕХНОЛОГИИ ОРКЕСТРОВКИ DOCKER SWARM И KUBERNETES

**Чаплыгин Никита Алексеевич**

**Гридчин Владимир Сергеевич**

**Балаев Владислав Алексеевич**

магистранты

ФГАОУ ВО «Северо-Кавказский федеральный университет»

**Аннотация:** в статье рассматриваются технологии оркестровки контейнеров Docker Swarm и Kubernetes, продемонстрированы преимущества и недостатки данных систем в различных условиях, проведен сравнительный анализ, приведены примеры использования технологий для решения различных задач бизнеса.

**Ключевые слова:** Docker, Swarm, Kubernetes, контейнеры, микросервисы, оркестровка, архитектура

## DOCKER SWARM AND KUBERNETES ORCHESTRATION TECHNOLOGIES

**Chaplygin Nikita Alekseevich**

**Gridchin Vladimir Sergeevich**

**Balaev Vladislav Alekseevich**

**Abstract:** the article discusses the technologies of orchestration of Docker Swarm and Kubernetes containers, demonstrates the advantages and disadvantages of these systems in various conditions, conducts a comparative analysis, provides examples of the use of technologies to solve various business problems.

**Key words:** Docker, Swarm, Kubernetes, containers, microservices, orchestration, architecture

Компании по всему миру все больше полагаются на преимущества контейнерных технологий, чтобы облегчить развертывание сложных приложений и управление ими. Контейнеры группируют все необходимые зависимости в одном пакете. Они портативные, быстрые, безопасные, масштабируемые и простые в управлении, что делает их предпочтительным

выбором по сравнению с традиционными виртуальными машинами. Но для масштабирования контейнеров необходим инструмент оркестровки – платформа для управления множеством контейнеров.

Сегодня наиболее известными платформами оркестровки контейнеров являются Docker Swarm и Kubernetes. У обоих есть свои преимущества и недостатки, и оба они созданы для достижения определенных целей. В этой статье исследуются обе технологии, чтобы определить, какой инструмент оркестровки контейнеров лучше всего подходит под конкретную задачу.

Docker Swarm – это платформа оркестровки контейнеров с открытым исходным кодом, созданная и поддерживаемая Docker. Под капотом Docker Swarm преобразует несколько экземпляров Docker в один виртуальный хост. Кластер Docker Swarm обычно состоит из трех элементов:

- Узлы
- Услуги и задачи
- Балансировщики нагрузки

Узлы – это отдельные экземпляры механизма Docker, которые контролируют кластер и управляют контейнерами, используемыми для запуска сервисов и задач. Кластеры Docker Swarm также включают балансировку нагрузки для маршрутизации запросов между узлами.

Docker Swarm прост в установке, особенно для тех, кто только начинает заниматься оркестровкой контейнеров. Он легкий и простой в использовании. Кроме того, для понимания Docker Swarm требуется меньше времени, чем для понимания более сложных инструментов оркестровки. Он обеспечивает автоматическую балансировку нагрузки в контейнерах Docker, тогда как другие инструменты оркестрации контейнеров требуют ручных усилий.

Docker Swarm работает с Docker CLI, поэтому нет необходимости запускать или устанавливать весь новый CLI. Кроме того, он без проблем работает с существующими инструментами Docker, такими как Docker Compose.

Docker Swarm не требует изменения конфигурации, если ваша система уже работает внутри Docker.

Несмотря на свои преимущества, у использования Docker Swarm есть несколько недостатков, о которых вам следует знать.

Он легкий и привязан к Docker API, что ограничивает функциональность Docker Swarm по сравнению с Kubernetes. Также возможности автоматизации Docker Swarm не так надежны, как у Kubernetes.

Kubernetes – это платформа оркестровки контейнеров с открытым исходным кодом, которая изначально была разработана Google для управления своими контейнерами. Kubernetes имеет более сложную кластерную структуру, чем Docker Swarm. Обычно он имеет архитектуру компоновщика и рабочих узлов, разделенных на модули, пространства имен, карты конфигурации и так далее.

Kubernetes предлагает широкий спектр преимуществ для команд, которым нужен надежный инструмент оркестровки контейнеров:

- У него большое сообщество разработчиков ПО с открытым исходным кодом, и Google его поддерживает.
- Он поддерживает все операционные системы.
- Он может поддерживать и управлять крупными архитектурами и сложными рабочими нагрузками.
- Он автоматизирован и обладает способностью самовосстановления, которая поддерживает автоматическое масштабирование.
- Он имеет встроенный мониторинг и широкий спектр доступных интеграций.
- Его предлагают все три ключевых облачных провайдера: Google, Azure и AWS.

Благодаря широкой поддержке сообщества и способности обрабатывать даже самые сложные сценарии развертывания, Kubernetes часто является выбором номер один для групп корпоративных разработчиков, управляющих приложениями на основе микросервисов.

Несмотря на обширный набор функций, Kubernetes также имеет несколько недостатков:

- Сложный процесс установки и крутая кривая обучения.
- Необходимость установки отдельных инструментов CLI и изучение каждого из них.
- Переход от Docker Swarm к Kubernetes может быть сложным.

В некоторых ситуациях Kubernetes может быть слишком сложным и привести к снижению производительности.

Платформы сравниваются с точки зрения требований к настройке, возможностей развертывания приложений, доступности и масштабирования, функций мониторинга, безопасности и балансировки нагрузки.

Docker Swarm прост в установке по сравнению с Kubernetes, а экземпляры обычно согласованы во всей ОС. Настроить кластер в Docker

Swarm проще, чем настроить Kubernetes. Его легче изучить, и он работает с существующим интерфейсом командной строки. По сравнению с Docker Swarm, Kubernetes имеет более сложную установку и требует ручных усилий. Инструкции по установке могут отличаться для каждой ОС. Его сложно изучить, и для него нужен отдельный инструмент CLI.

Приложения Docker Swarm – это службы или микросервисы, которые можно развернуть с помощью файлов YAML или Docker Compose. Kubernetes предоставляет более широкий набор опций, таких как сочетание пространств имен, модулей и развертываний.

Docker Swarm обеспечивает высокую доступность, поскольку вы можете легко дублировать микросервисы в нем. Кроме того, Docker Swarm имеет более быстрое время развертывания. С другой стороны, он не обеспечивает автоматического масштабирования. Kubernetes по своей природе высокодоступен, отказоустойчив и самовосстанавливается. Он также обеспечивает автоматическое масштабирование и при необходимости может заменить неисправные блоки.

Docker Swarm поддерживает мониторинг только через сторонние приложения. Встроенных механизмов мониторинга нет. Kubernetes имеет встроенный мониторинг и поддерживает интеграцию со сторонними инструментами мониторинга.

Docker Swarm полагается на безопасность транспортного уровня (TLS) для выполнения задач, связанных с безопасностью и контролем доступа. Kubernetes поддерживает несколько протоколов безопасности, таких как RBAC, SSL / TLS, управление секретами, политики.

Docker Swarm поддерживает автоматическую балансировку нагрузки и использует DNS под капотом. В Kubernetes нет механизма автоматической балансировки нагрузки. Однако Nginx Ingress может служить балансировщиком нагрузки для каждой службы в кластере.

И Kubernetes, и Docker Swarm реализуют определенные варианты использования. Какой из них стоит использовать, зависит от потребностей организации.

Для новичков Docker Swarm - простое в использовании решение для масштабного управления контейнерами. Подходит для компаний, которые начинают использовать технологии контейнеризации и не имеют сложных рабочих нагрузок для управления.

Если используются сложные приложения, необходим полный пакет, который включает в себя мониторинг, функции безопасности,

самовосстановление, высокую доступность и абсолютную гибкость, тогда стоит использовать Kubernetes.

В этой статье исследованы две основные технологии оркестровки контейнеров, Kubernetes и Docker Swarm. Docker Swarm – это легкий, простой в использовании инструмент оркестровки с ограниченными предложениями по сравнению с Kubernetes. Напротив, Kubernetes сложен, но мощный и предоставляет возможности самовосстановления и автоматического масштабирования из коробки.

То, какой инструмент оркестрации лучше, зависит от потребностей бизнеса. Прежде чем делать выбор, необходимо проанализировать цели и опыт своей команды.

### **Список литературы**

1. Fabrizio Soppelsa, Chanwit Kaewkasi. Native Docker Clustering with Swarm // Packt Publishing. – 2016. – 280 с.
2. Russ McKendrick. Mastering Docker: Enhance your containerization and DevOps skills to deliver production-ready application // Packt Publishing. – 2020. – 568 с.
3. Viktor Farcic. The DevOps 2.1 Toolkit: Docker Swarm: Building, testing, deploying, and monitoring services inside Docker Swarm clusters (The DevOps Toolkit Series) // CreateSpace Independent Publishing Platform. – 2017. – 403 с.
4. Brendan Burns, Joe Beda, Kelsey Hightower. Kubernetes: Up and Running: Dive into the Future of Infrastructure // O'Reilly Media. – 2019. – 278 с.
5. Nisarg Vasavada, Dhvani Sametriya. Cracking Containers with Docker and Kubernetes: The definitive guide to Docker, Kubernetes, and the Container Ecosystem across Cloud and on-premises (English Edition) // BPB Publications. – 2021. – 752 с.