Реферат на тему “KUBERNETES”

# ВВЕДЕНИЕ

В последние годы контейнеризация стала стандартом в разработке и эксплуатации программного обеспечения. С ростом количества контейнеров возникла необходимость в инструменте, который мог бы управлять ими на уровне инфраструктуры. Таким решением стал **Kubernetes** — открытая система оркестрации контейнеров, разработанная компанией Google и переданная в сообщество **Cloud Native Computing Foundation (CNCF)**.

**Актуальность темы** заключается в том, что Kubernetes стал ключевым инструментом для DevOps и облачных технологий, обеспечивая автоматизацию развертывания, масштабирования и управления приложениями.

**Цель работы** — рассмотреть архитектуру, принципы работы и преимущества Kubernetes.  
**Задачи:**

1. Изучить историю возникновения и развития системы Kubernetes;
2. Рассмотреть её структуру и основные компоненты;
3. Проанализировать преимущества Kubernetes по сравнению с другими системами оркестрации;
4. Определить сферы применения технологии в современной IT-индустрии.

# 2. Понятие контейнеризации и предпосылки появления Kubernetes

В этом разделе можно описать:

* что такое контейнеры (Docker, LXC);
* какие проблемы они решают (изолированность, переносимость, автоматизация);
* как возникла потребность в управлении множеством контейнеров;
* кратко упомянуть появление Kubernetes в 2014 году и его открытый код.

# 3. Архитектура и основные компоненты Kubernetes

Раскрой следующие элементы:

* **Master Node** — управляющая часть кластера;
* **Worker Node** — рабочие узлы, где запускаются контейнеры;
* **Pod** — минимальная единица развертывания;
* **Deployment**, **ReplicaSet**, **Service**, **Ingress** — основные объекты управления;
* **etcd**, **API Server**, **Controller Manager**, **Scheduler** — компоненты управления.

Можно добавить простую схему или описание взаимодействия компонентов.

# 4. Преимущества и области применения

Опиши, почему Kubernetes стал стандартом:

* автоматическое масштабирование и самовосстановление приложений;
* возможность бесшовных обновлений (rolling updates);
* кроссплатформенность и поддержка разных облаков;
* удобство интеграции с CI/CD и DevOps-процессами.

Примеры применения:

* облачные сервисы (Google Cloud, AWS, Azure);
* крупные корпоративные системы;
* микросервисные архитектуры.

# Сравнение Kubernetes с альтернативами

Можно кратко сравнить Kubernetes с:

* **Docker Swarm** — проще, но менее гибкий;
* **Apache Mesos** — масштабнее, но сложнее в настройке;
* **Nomad (HashiCorp)** — лёгкий, но с меньшей экосистемой.

Вывод: Kubernetes — наиболее универсальное решение с широкой поддержкой и активным сообществом.

ЗАКЛЮЧЕНИЕ