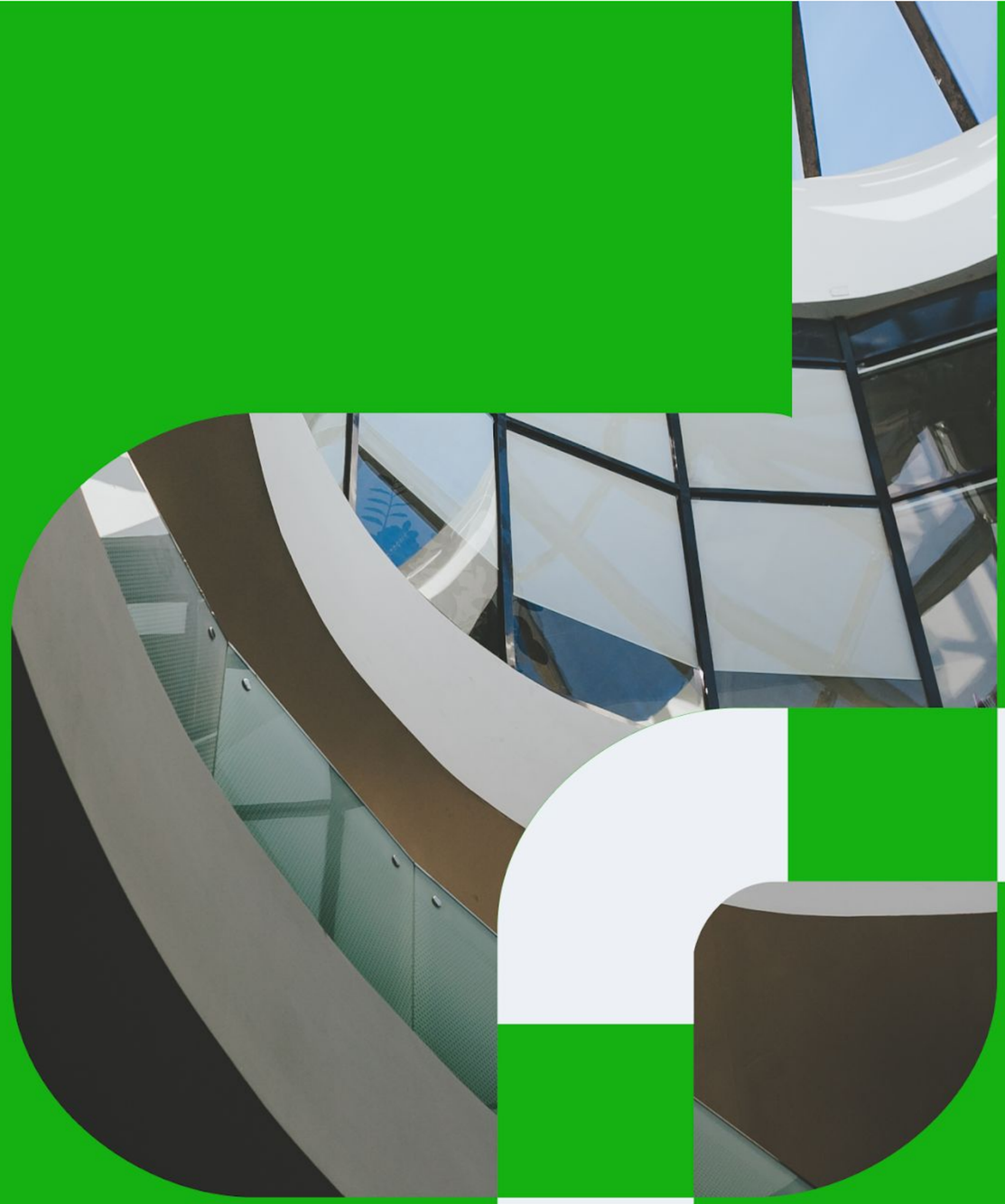
ИНСТИТУТ
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС

# k8s

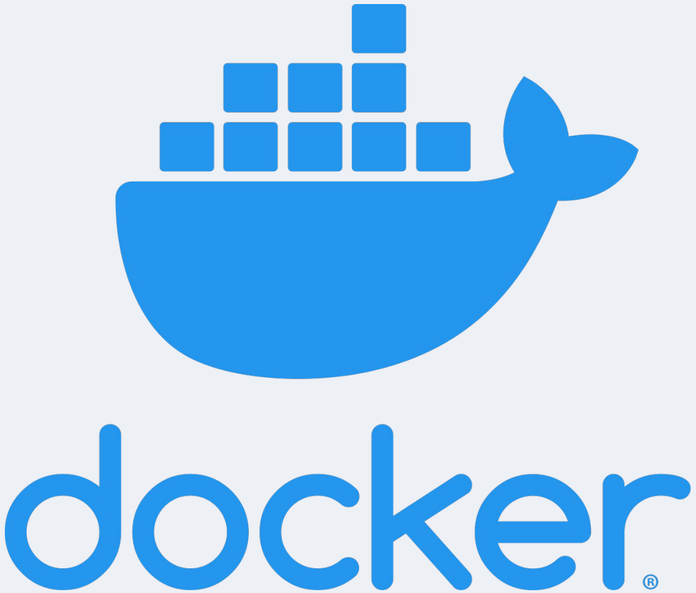


Мухамадуллин Ильяс
ilyasmukh@icloud.com

# Что такое контейнеризация?

УНИВЕРСИТЕТ ИННОПОЛИС

**Контейнеризация** – технология, которая позволяет упаковывать и запускать приложения и их зависимости в изолированных окружениях, называемых контейнерами.



# Преимущества контейнеризации

УНИВЕРСИТЕТ ИННОПОЛИС

**Изоляция**:
Контейнеры изолируют приложение и его зависимости от хост-системы и других контейнеров, обеспечивая надежность и безопасность.

**Переносимость**:
Контейнеры можно развернуть на любой системе, которая поддерживает контейнерную платформу, что делает их переносимыми между различными окружениями.

# Преимущества контейнеризации

УНИВЕРСИТЕТ ИННОПОЛИС

**Масштабируемость**:
Контейнеры легко масштабируются для обработки увеличивающейся нагрузки, путем запуска дополнительных контейнеров.

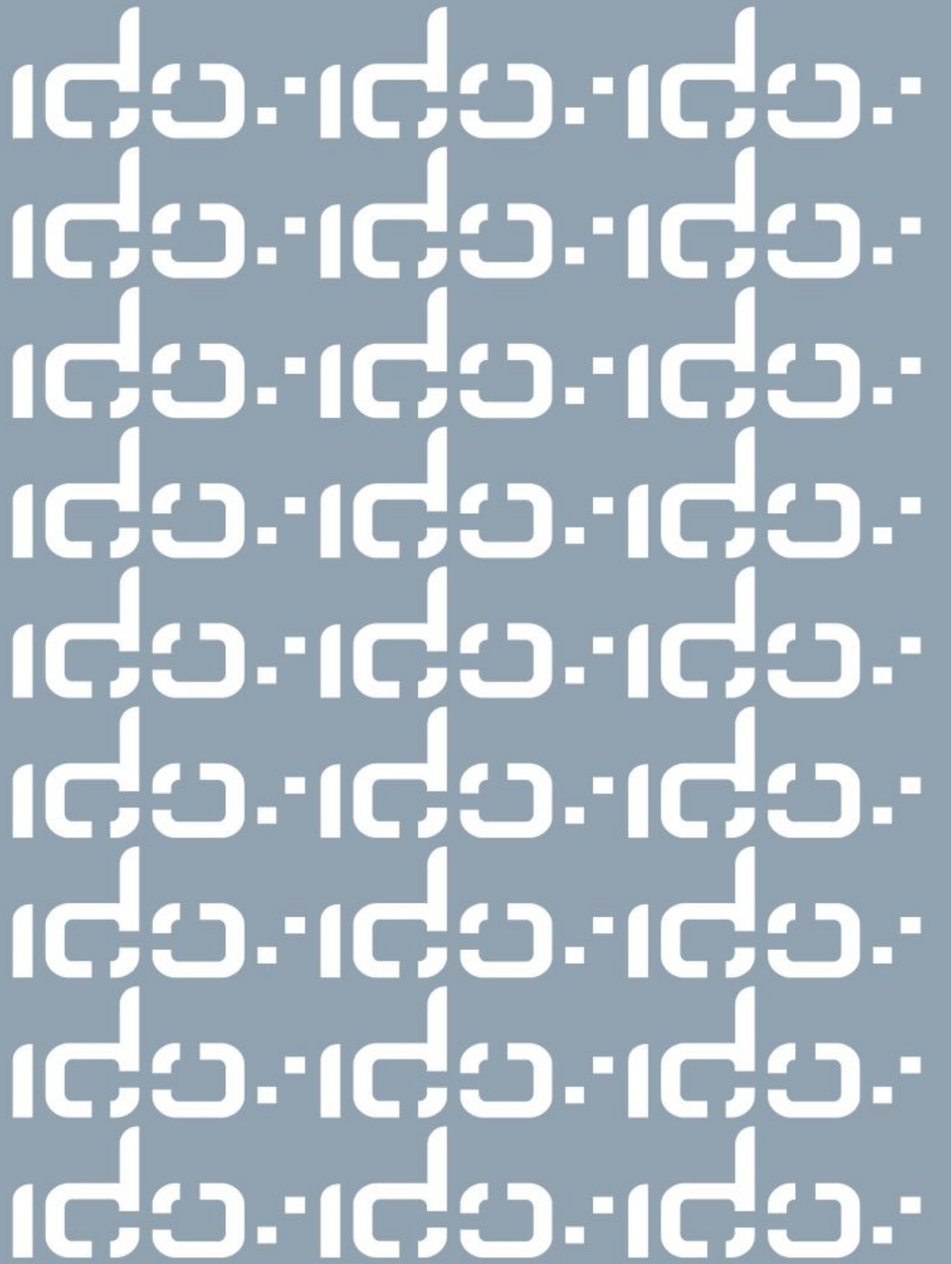
**Управление зависимостями**:
Все зависимости, необходимые для работы приложения, могут быть включены в контейнер, что упрощает управление зависимостями.

**Упрощенное развертывание**:
Позволяет развертывать приложения быстрее и более

УНИВЕРСИТЕТ ИННОПОЛИС

# Контейнеризация

## Основные понятия



# Контейнер

УНИВЕРСИТЕТ ИННОПОЛИС

**Контейнер** - это запускаемый пакет, который включает в себя приложение и все его зависимости, включая библиотеки, настройки и другие файлы, необходимые для работы приложения.

Контейнеры изолируют приложение от окружающей системы и могут работать на любой системе, поддерживающей контейнерную платформу.

# Контейнерная платформа

УНИВЕРСИТЕТ ИННОПОЛИС

**Контейнерная платформа -** это программное обеспечение, которое управляет созданием, запуском и управлением контейнерами.

Примерами популярных контейнерных платформ являются Docker, Kubernetes и Podman.

# Образ

УНИВЕРСИТЕТ ИННОПОЛИС

**Контейнерный образ** - это стандартизированный шаблон, который содержит все необходимые компоненты для создания контейнера.

Образы могут быть разработаны и распространены другими пользователями. Они могут использоваться для создания контейнеров с минимальными усилиями.

# Содержание контейнера

УНИВЕРСИТЕТ ИННОПОЛИС

1. **Файлы приложения**
   Исполняемый код, библиотеки, конфигурационные файлы, шаблоны и тп.

2. **Зависимости**
   Библиотеки и зависимости, необходимые для выполнения вашего приложения.

3. **Операционная система**
   Базовая файловая система, ядро операционной системы и необходимые инструменты.

4. **Настройки и переменные среды**
   Конфигурации для запуска.

5. **Скрипты и команды**
   Команды для запуска вашего приложения, настройки среды и другие действия.

6. **Метаданные и информация о версии**
   Версии приложения, метаданные для идентификации и управления образом.

# Что такое k8s?

УНИВЕРСИТЕТ ИННОПОЛИС

**Kubernetes (кратко K8s)** - это современная и мощная система оркестрации контейнерами. Она разработана для автоматизации, масштабирования, и управления контейнеризированными приложениями.

kubernetes

# Особенности и преимущества k8s

УНИВЕРСИТЕТ ИННОПОЛИС

1. **Автоматизация и оркестрация**:
   Kubernetes обеспечивает автоматизацию развертывания, масштабирования и управления контейнеризированными приложениями. Он управляет циклом жизни приложения, начиная с развертывания и заканчивая его масштабированием и обновлением.

2. **Высокая доступность**
   K8s обеспечивает высокую доступность приложений и устойчивость к сбоям. Он способен автоматически восстанавливать контейнеры после сбоев и перераспределять нагрузку между рабочими узлами.

3. **Масштабирование**
   Kubernetes позволяет легко масштабировать приложения, добавляя или удаляя контейнеры в зависимости от нагрузки. Это делает его идеальным для микросервисных архитектур.

4. **Управление ресурсами**
   K8s предоставляет мощные средства управления ресурсами, включая возможность установки ограничений по использованию CPU и памяти для контейнеров.

# Особенности и преимущества k8s

УНИВЕРСИТЕТ ИННОПОЛИС

6. **Балансировка нагрузки**
   K8s предоставляет встроенные инструменты для балансировки нагрузки между контейнерами и сервисами.

7. **Секреты и конфигурации**
   Система управления секретами и конфигурациями в K8s позволяет хранить и управлять конфиденциальными данными и настройками приложений.

8. **Обновление и откат**
   K8s позволяет обновлять приложения без простоев.Можно быстро откатиться к предыдущей версии.

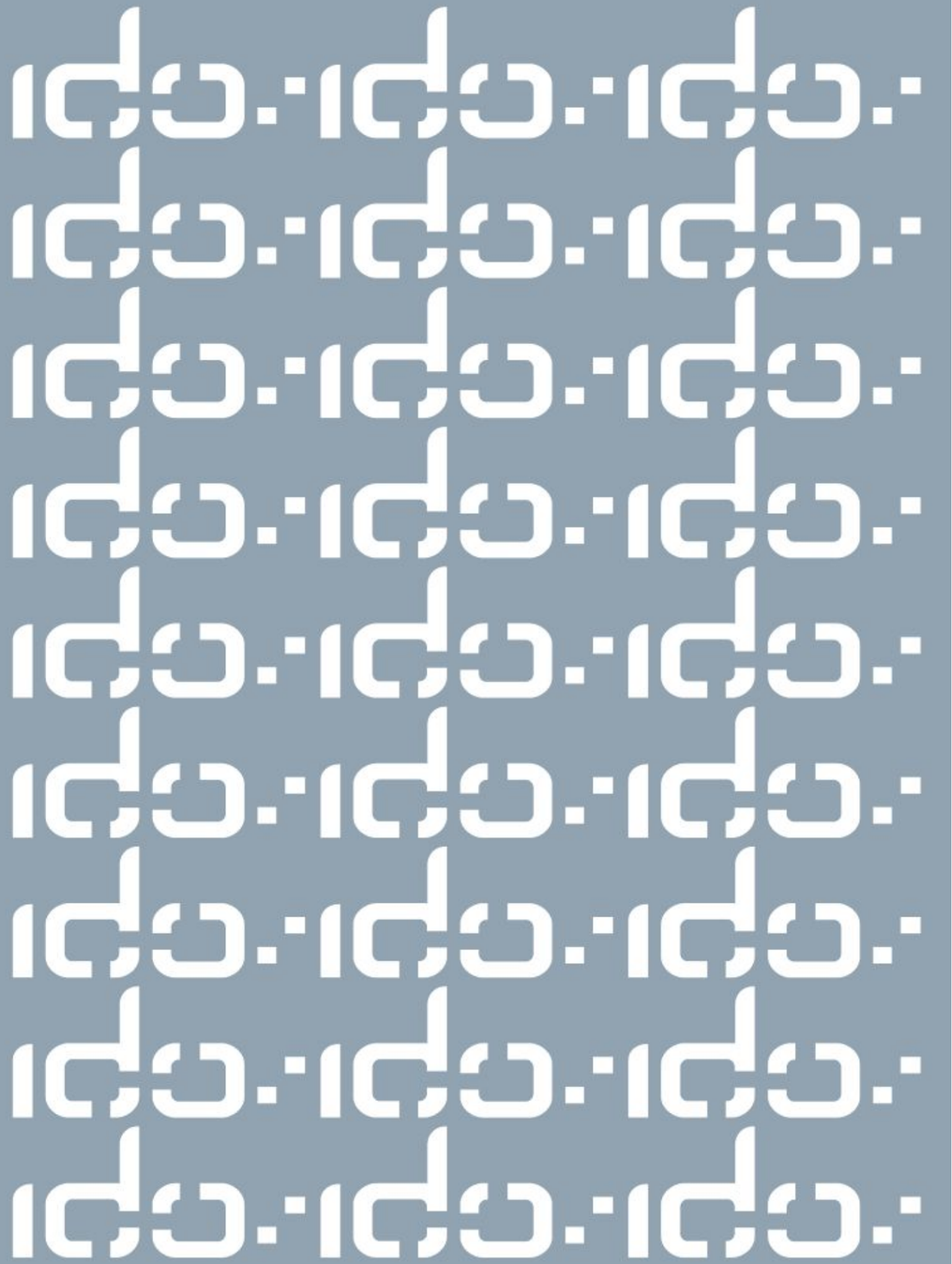
9. **Мультиоблако**
   Kubernetes поддерживает развертывание приложений в различных облаках (мультиоблачные стратегии)

10. **Расширяемость и плагины**
    Система поддерживает множество плагинов для доп. функций, таких как мониторинг и логирование.

УНИВЕРСИТЕТ ИННОПОЛИС

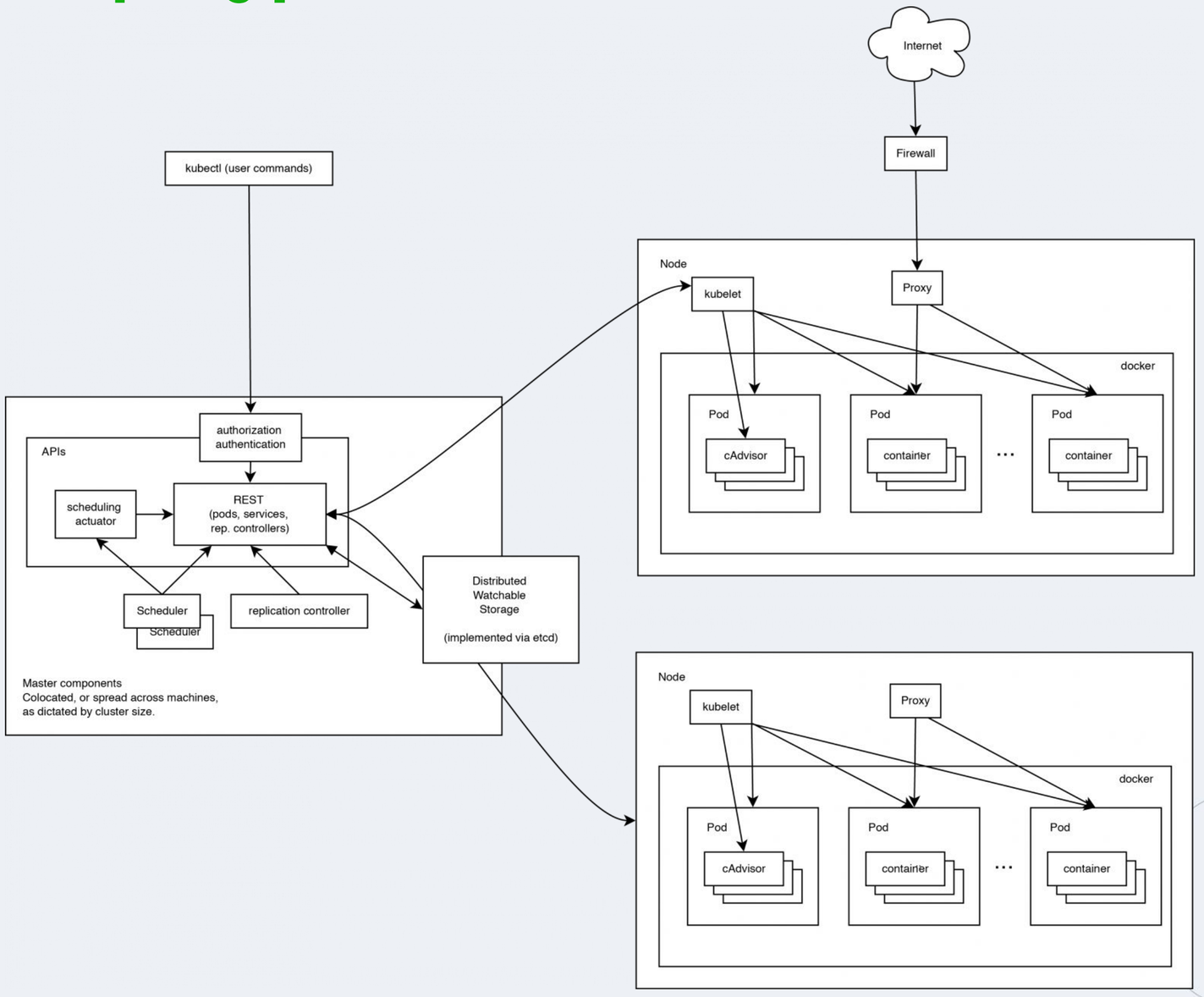
# Kubernetes

## Основные понятия



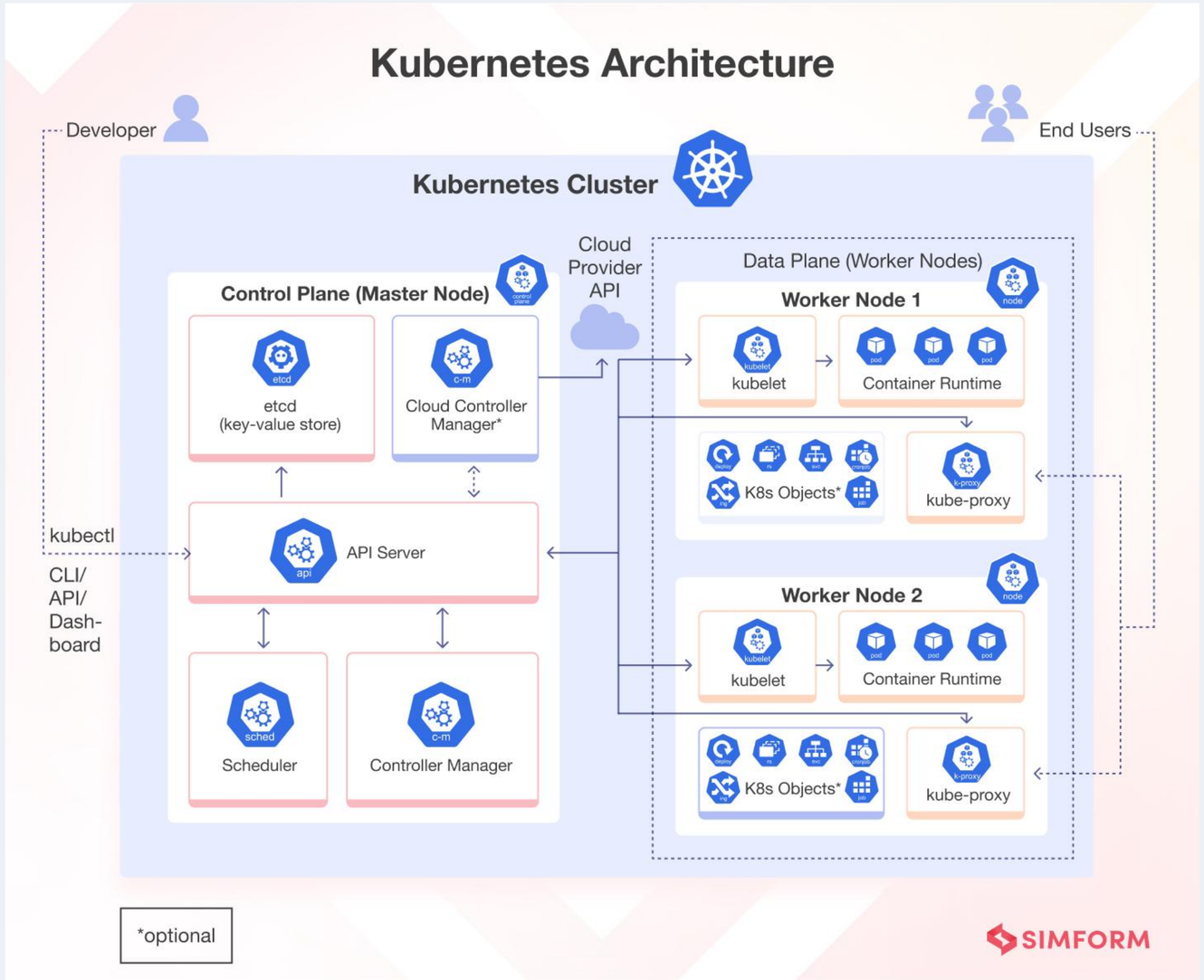
# Типовая конфигурация

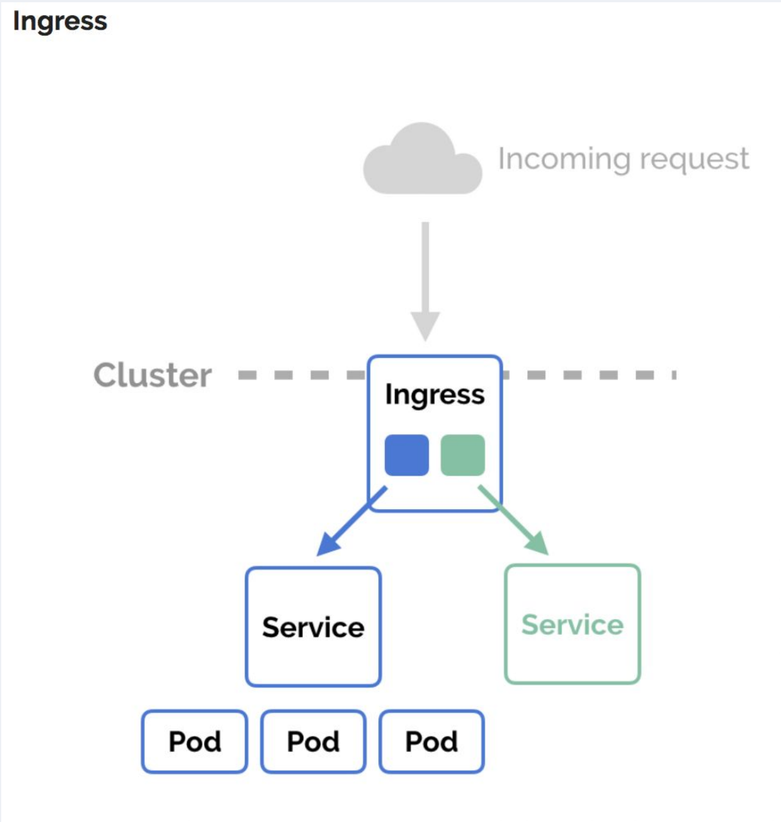
УНИВЕРСИТЕТ ИННОПОЛИС



# Типовая конфигурация

УНИВЕРСИТЕТ ИННОПОЛИС

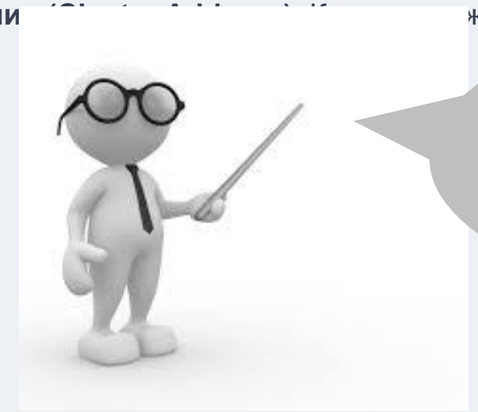


# Основные понятия

УНИВЕРСИТЕТ ИННОПОЛИС

1. **Мастер-сервер**: Мастер-сервер (или контрольная плоскость) - это главный управляющий узел Kubernetes. Он содержит несколько компонентов, включая:
    1. **API-сервер**: Он предоставляет API для управления кластером. Все операции с кластером выполняются через API-сервер.
    2. **etcd**: Распределенное и согласованное хранилище, в котором хранится конфигурация кластера и состояние всех объектов Kubernetes.
    3. **Компоненты управления (controller managers)**: Эти компоненты контролируют состояние кластера. Например, Replication Controller отвечает за масштабирование реплик подов.
    4. **Планировщик (scheduler)**: Планировщик решает, на какой ноде в кластере следует запустить каждый под, учитывая ресурсы и требования приложения.
2. **Ноды (Nodes)**: Ноды - это рабочие узлы кластера, на которых выполняются контейнеры. Каждая нода обычно содержит:
    1. **Kubelet**: Агент, который поддерживает связь между мастер-сервером и нодой. Он управляет контейнерами на ноде и отчитывается о состоянии контейнеров обратно мастеру.
    2. **Kube Proxy**: Этот компонент отвечает за сетевое проксирование на ноде. Он обеспечивает сетевую изоляцию между подами и маршрутизацию сетевого трафика.
    3. **Container Runtime**: Это программное обеспечение, которое выполняет контейнеры, например, Docker.
3. **Поды (Pods)**: Поды - это самые маленькие развертываемые объекты в Kubernetes. Они содержат один или несколько контейнеров и разделяют сетевой стек и хранилище. Поды используются для группировки и управления контейнерами вместе.
4. **Сервисы (Services)**: Сервисы определяют сетевые правила для доступа к подам. Они обеспечивают стабильные точки доступа к подам, даже если поды пересоздаются или перемещаются.
5. **Конфиг-карты (ConfigMaps) и Секреты (Secrets)**: Эти ресурсы позволяют управлять конфигурациями и секретами, используемыми в приложениях.
6. **Подчиненные ресурсы (Controllers)**: Эти ресурсы управляют состоянием приложений и ресурсов. Например, Deployment позволяет управлять репликами подов.
7. **Хранилища и объекты управления состоянием**: Kubernetes позволяет управлять хранилищем и объектами управления состоянием, такими как Persistent Volumes (PV) и Persistent Volume Claims (PVC).
8. **Сети и политики (Network and Policies)**: Kubernetes имеет механизмы для настройки сетевых политик, обеспечения сетевой изоляции и управления трафиком.
9. **Кластерные дополнени** жет быть допо ми, такими как мониторинг, журналирование, панели управления и другие.



Этот текст такой мелкий для шпаргалки.

Данные понятия подробно разобраны на схеме на предыдущем слайде ;)

# Docker vs k8s

УНИВЕРСИТЕТ ИННОПОЛИС

1. **Масштабирование и оркестрация**:
    1. **Docker**: Создание и запуск контейнеров на отдельной машине.
    2. **Kubernetes**: Оркестрация - управляет разворачиванием и масштабированием контейнеров на кластере серверов, обеспечивая высокую доступность и отказоустойчивость.

2. **Управление состоянием**:
    1. **Docker**: Видно состояние контейнера.
    2. **Kubernetes**: Позволяет управлять состоянием приложений, что включает в себя масштабирование, восстановление после сбоев, обновления без простоев и балансировку нагрузки.

3. **Объединенное управление**:
    1. **Docker**: Локально на одной машине.
    2. **Kubernetes**: На кластере серверов, что позволяет унифицированное и масштабируемое управление множеством контейнеров.

4. **Балансировка нагрузки**:
    1. **Docker**: Нет.
    2. **Kubernetes**: Kubernetes имеет встроенные инструменты для балансировки нагрузки между контейнерами приложений.

5. **Планирование**:
    1. **Docker**: Нет.
    2. **Kubernetes**: Kubernetes включает средства для планирования и управления ресурсами, что позволяет оптимизировать использование аппаратного оборудования.

# Контейнеризация и Виртуализация

УНИВЕРСИТЕТ ИННОПОЛИС

## 1.Уровень изоляции:

1. **Контейнеризация**: Контейнеры работают на уровне операционной системы (ОС). Они используют общее ядро ОС, но изолируют приложение и его зависимости в отдельное пространство пользователя. Каждый контейнер имеет свою файловую систему и процессы, но использует общее ядро ОС.
2. **Виртуализация**: Виртуальные машины (ВМ) создают полную виртуальную копию физической аппаратной платформы. Они включают в себя свою собственную копию операционной системы, что делает их более изолированными от хост-системы.

## 2.Ресурсы и производительность:

1. **Контейнеризация**: Контейнеры более легковесны и делят ресурсы с хост-системой. Они обычно имеют более быстрое время запуска и меньшее потребление ресурсов. Однако, из-за общего ядра, они могут иметь ограничения в изоляции и безопасности.
2. **Виртуализация**: ВМ имеют более высокий уровень изоляции, но они более ресурсоемки и могут иметь более долгое время запуска.

## 3.Запуск приложений:

1. **Контейнеризация**: Контейнеры обычно используются для упаковки и развертывания одного приложения с его зависимостями. Они легко масштабируются и управляются, но не предоставляют такую высокую изоляцию.
2. **Виртуализация**: ВМ позволяют запускать несколько приложений и даже разные операционные системы на одном хосте. Они обеспечивают высокую изоляцию, но могут потреблять больше ресурсов.

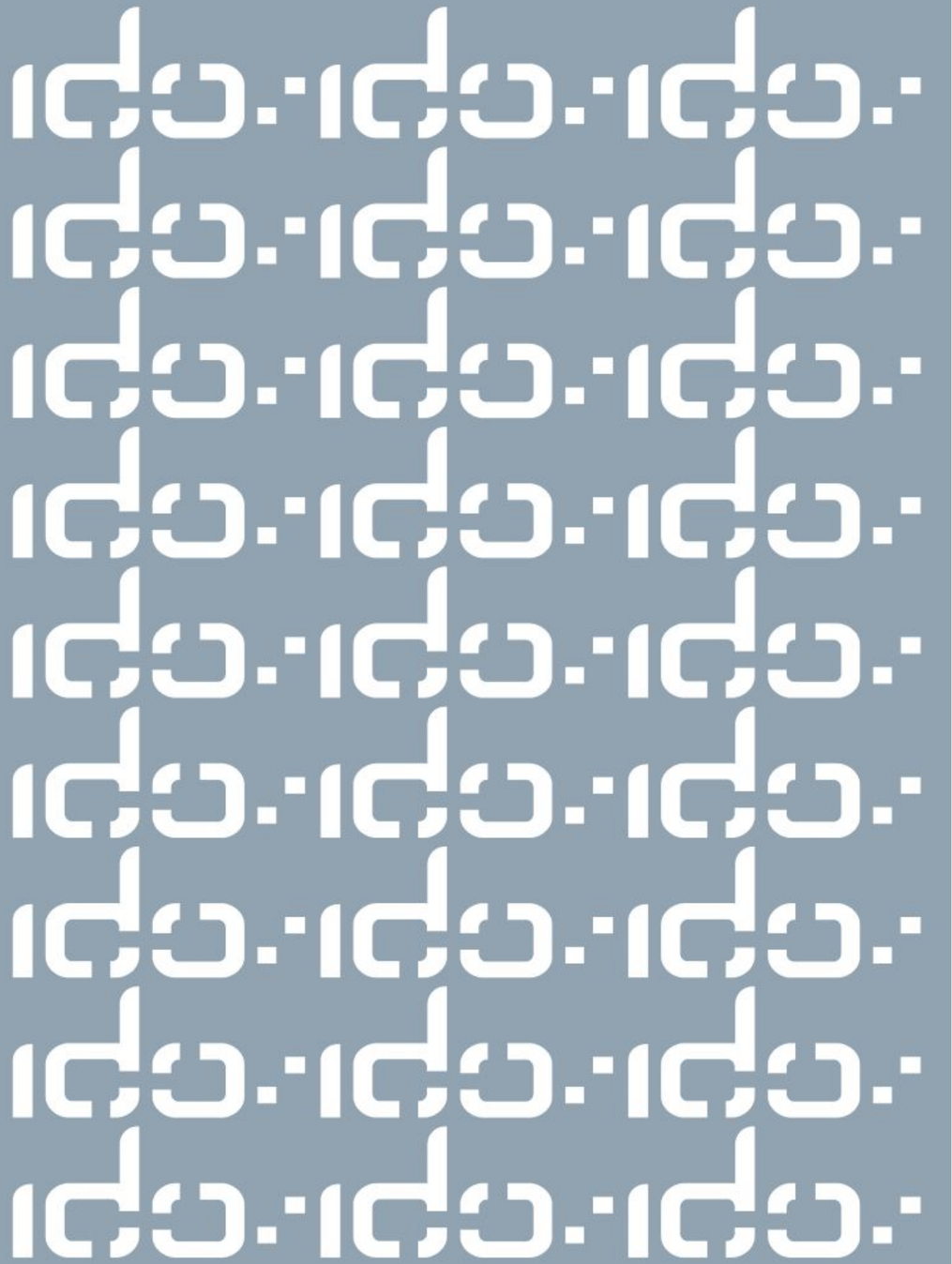
## 4.Скорость развертывания и масштабирования:

1. **Контейнеризация**: Контейнеры обычно разворачиваются быстрее и масштабируются легче благодаря своей легковесной природе.

УНИВЕРСИТЕТ ИННОПОЛИС

# Почему не всегда все приложения размещают в k8s?

## Еще немного особенностей



# Хранение данных

УНИВЕРСИТЕТ ИННОПОЛИС

СУБД требуют постоянного хранения данных.

Kubernetes имеет механизмы для работы с постоянными томами, но управление данными внутри контейнеров может быть сложной задачей.

Необходимо также заботиться о резервном копировании и восстановлении данных.

Некоторые организации предпочитают размещать СУБД вне Kubernetes для лучшего управления данными.

# Использование ресурсов

УНИВЕРСИТЕТ ИННОПОЛИС

СУБД и Kafka могут быть ресурсоемкими приложениями.

Это может потребовать тщательной настройки ресурсов в Kubernetes, чтобы избежать конфликтов между приложениями.

Неправильная конфигурация ресурсов может привести к снижению производительности всего кластера.

# Особенности настройки

УНИВЕРСИТЕТ ИННОПОЛИС

Настройка СУБД и Kafka, включая кластеризацию и репликацию, может быть сложной задачей.

Хотя Kubernetes предоставляет инструменты для оркестрации контейнеров, конфигурация и управление самими приложениями требует опыта и усилий.

УНИВЕРСИТЕТ ИННОПОЛИС

# Демонстрация

Архитектурный дизайн типовых решений

ИНСТИТУТ
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС

# Спасибо за внимание

Мухамадуллин Ильяс
ilyasmukh@icloud.com