



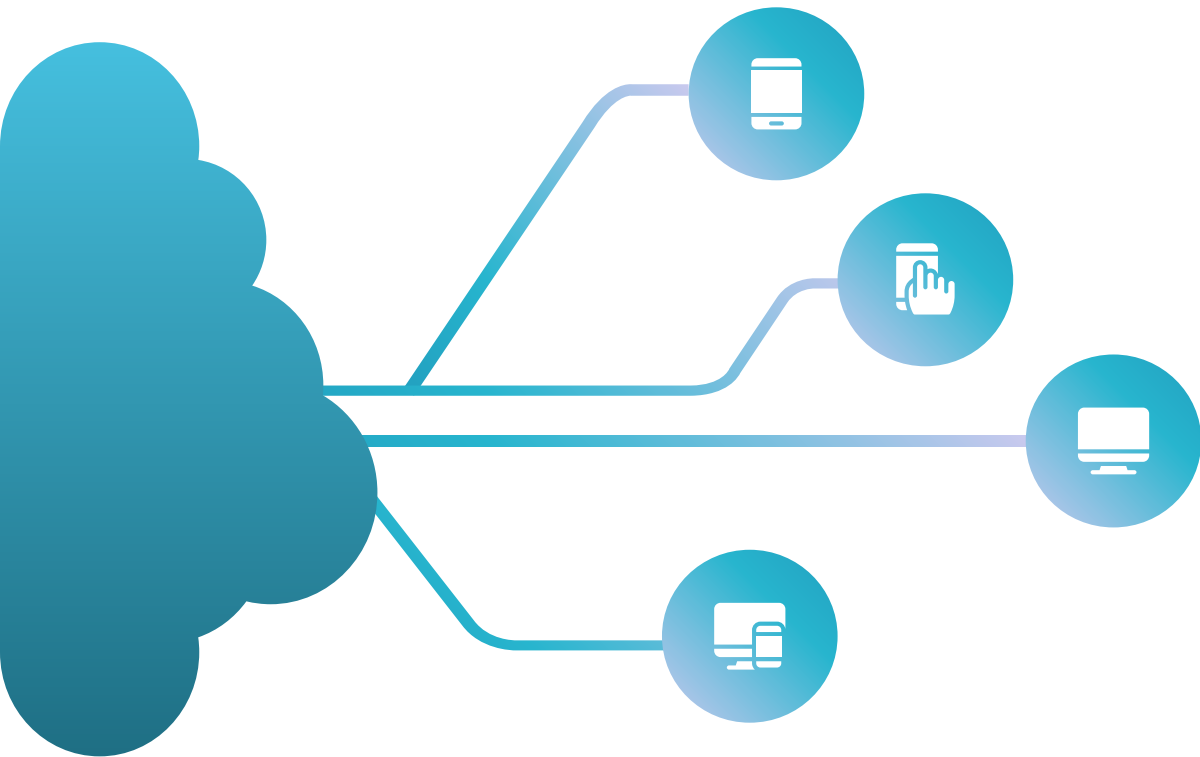
Виртуализация и контейнеризация в облачных системах

Преподаватель: Нартя Никита

Группа: I2302

Авторы: Чобану Станислав
 Мамалига Артур

Введение



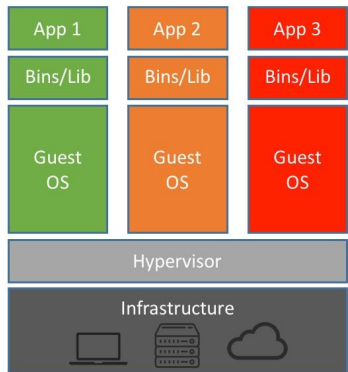
Облачные вычисления

становятся ключевой технологией современной IT-инфраструктуры.

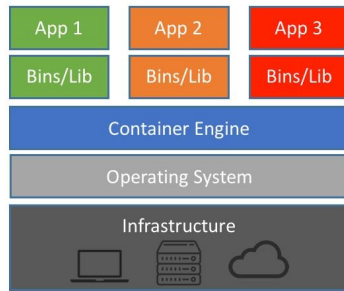
В основе лежит **виртуализация**, технология, что позволяет создавать логические (**виртуальные**) ресурсы поверх физических серверов.

Она предоставляет возможность создать изолированную, масштабируемую и надежную **инфраструктуру**, для обеспечения непрерывной работы сервисов.

Основные понятия



Machine Virtualization



Containers



Полная виртуализация - эмуляция оборудования гипервизором, на котором запускается полноценные ОС.
Пример: AWS EC2

Паравиртуализация - виртуальная ОС запущена на гипервизоре, но требует модифицированное ядро. Ранее применялась в EC2 PV инстансах. Сейчас почти не используется.

Контейнеризация - упаковка приложений и их зависимостей в изолированные среды поверх определенной ОС.
Пример: AWS ACS, Fargate

Ключевые характеристики

Изоляция ресурсов

Сбой или нагрузка одного сервиса не влияет на другие

Управляемость и автоматизация

Автоматическое развертывание, оркестрация, масштабирование.

Гибкость и масштабируемость

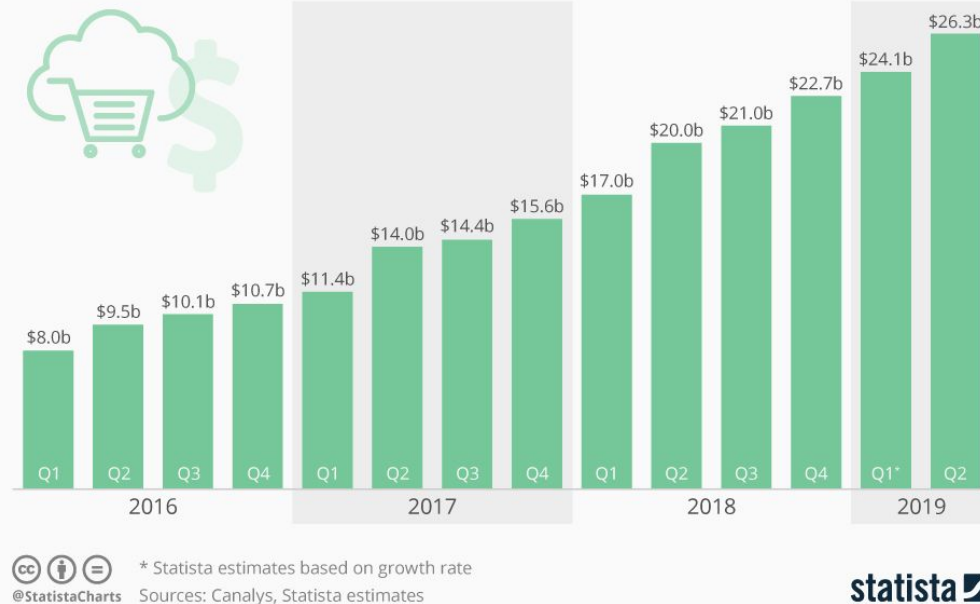
Способность быстро увеличивать, уменьшать ресурсы (CPU, RAM, storage)

Многоуровневая безопасность

Изоляция контейнеров, сетевые политики, шифрование данных, контроль доступа.

The Cloud Market Keeps Moving Upwards

Cloud infrastructure services market revenue worldwide from Q1 2016-Q2 2019



Теория: Виртуализация под капотом



Теория: Контейнеризация под капотом



chroot

Создает собственное файловое окружение. Со своим корневым каталогом.

PID namespaces

Изолирует процессы контейнера от процессов хостов и других контейнеров.



Network namespaces

Создает отдельный сетевой стек для контейнера. Создает иллюзию отдельной сети

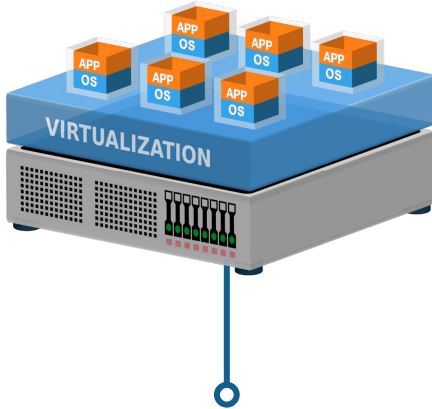
cgroups

Механизм управления ресурсами. Позволяет ограничивать доступные контейнеру ресурсы.

host kernel

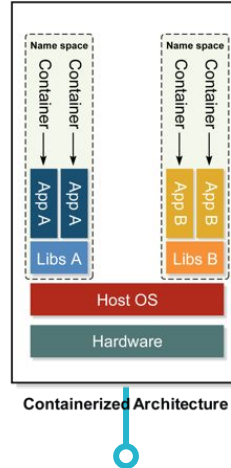
Все контейнеры работают в рамках общей операционной системы и используют одну общую ОС для системных вызовов.

Практическое применение



Виртуализация

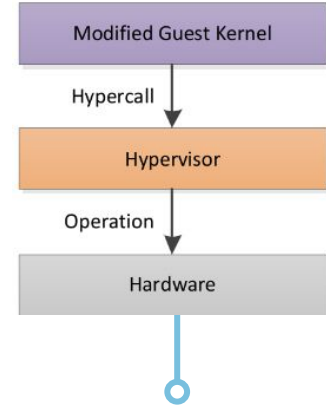
EC2, Azure Virtual Machines, VirtualBox, Qemu, VMware ESXi (bare metal hypervisor)



Контейнеризация

Docker, Podman, K8, LXC / LXD, Azure container app, AWS ECS

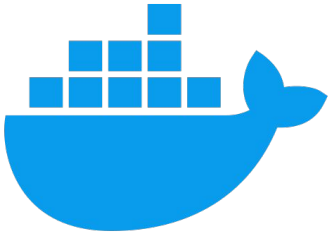
Para-virtualization



Паравиртуализация

Xen, KVM, VMware, Qemu

Когда использовать контейнеры



Контейнеры

Портативность

Нужно упаковать приложение и все его зависимости так, чтобы оно одинаково запускалось везде

Cloud-native и микросервисы

Cloud-native разработка, использование микросервисной архитектуры.

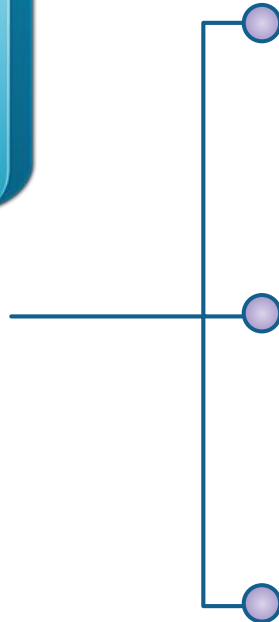
Интеграция с DevOps/CI/CD

Простота CI/CD процессов, автоматизация тестирования, сборки, развертывания.

Когда использовать виртуальные машины



Виртуалки



Legacy

Монолитные или требовательные приложения которые тяжело или невозможно разбить на микросервисы.

Безопасность

Нужна сильная изоляция между системами ради соблюдения критериев безопасности среды.

Зоопарк

Необходим запуск различных операционных систем или старых специфичных ОС.

Практическое применение (Статистика)



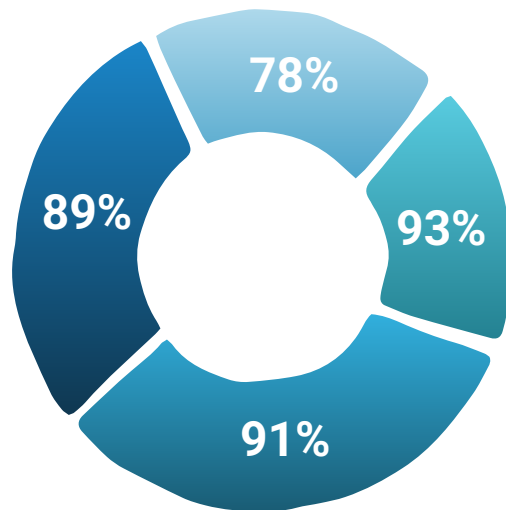
Cloud

Использование
облачных
технологий
достигло 89%



Containers

91% компаний
используют
контейнеры



Kubernetes

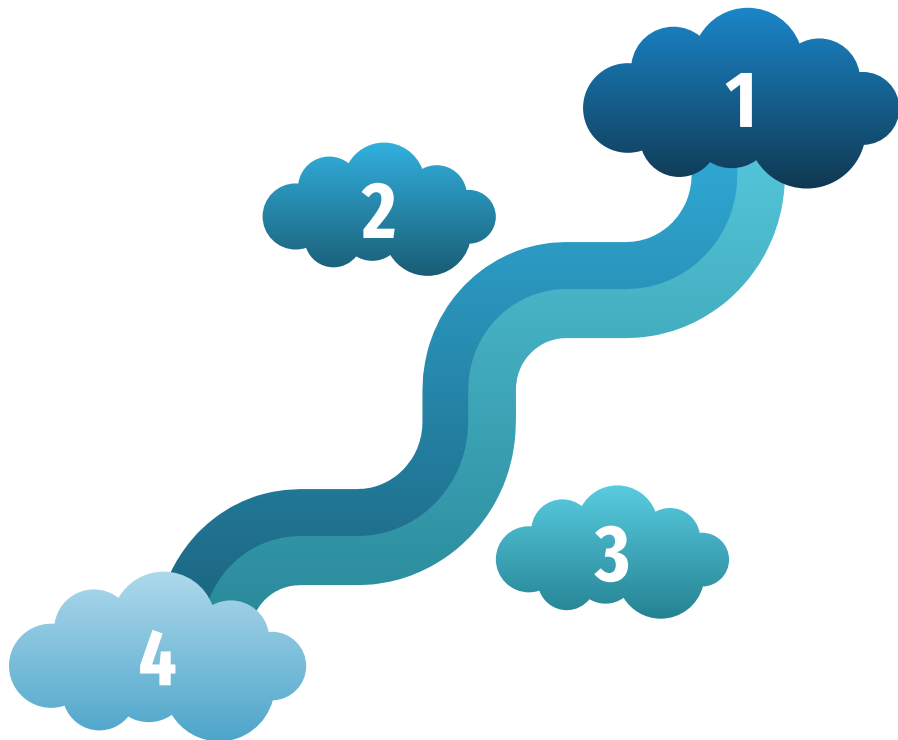
93% используют
или планируют
использовать
Kubernetes



Providers

Около 78%
компаний
используют
более одного
облачного
провайдера

Заключение



1

Виртуализация является ключевой технологией в современной IT инфраструктуре

2

Виртуализация позволяет выстраивать надёжную и изолированную среду для выполнения приложений

3

Виртуализация позволила развить сервисы для облачных вычислений и хостинга

4

Изобретение контейнеров позволило создать гибкую и масштабируемую инфраструктуру

Ссылки



<https://www.datadoghq.com/container-report-2021/>

https://www.cncf.io/wp-content/uploads/2025/04/cncf_annual_survey24_031225a.pdf

<https://habr.com/ru/articles/709268/>

<https://btholt.github.io/complete-intro-to-containers/namespaces>

<https://www.redhat.com/en/topics/containers/containers-vs-vms>