

# Методы и средства сборки и развертывания ПО, 25

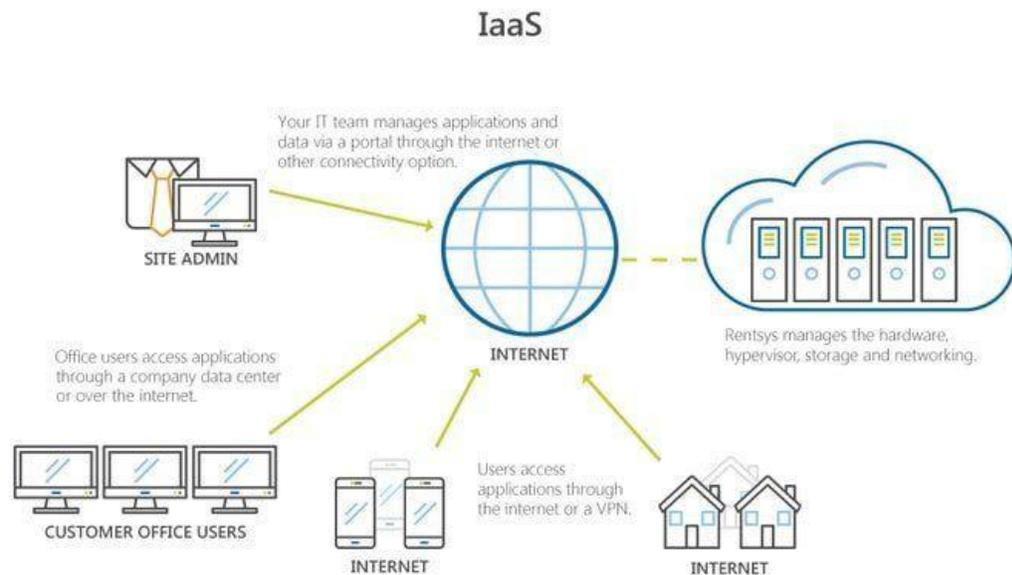
ФИО преподавателя: Смирнов Михаил Вячеславович

e-mail: [smirnovmgupi@gmail.com](mailto:smirnovmgupi@gmail.com)

## Лекция 6

# IaaS. Свойства и характеристики IaaS. IaaS в конвейере CI/CD

# Определение IaaS (Infrastructure-as-a-Service)



Один из вариантов модели облачных вычислений, где провайдер предоставляет заказчику фундаментальные вычислительные ресурсы через интернет: серверы, хранилища и сети.

Удобство в том, что вся инфраструктура арендуется и частично передается на аутсорс, а не покупается и поддерживается самостоятельно.

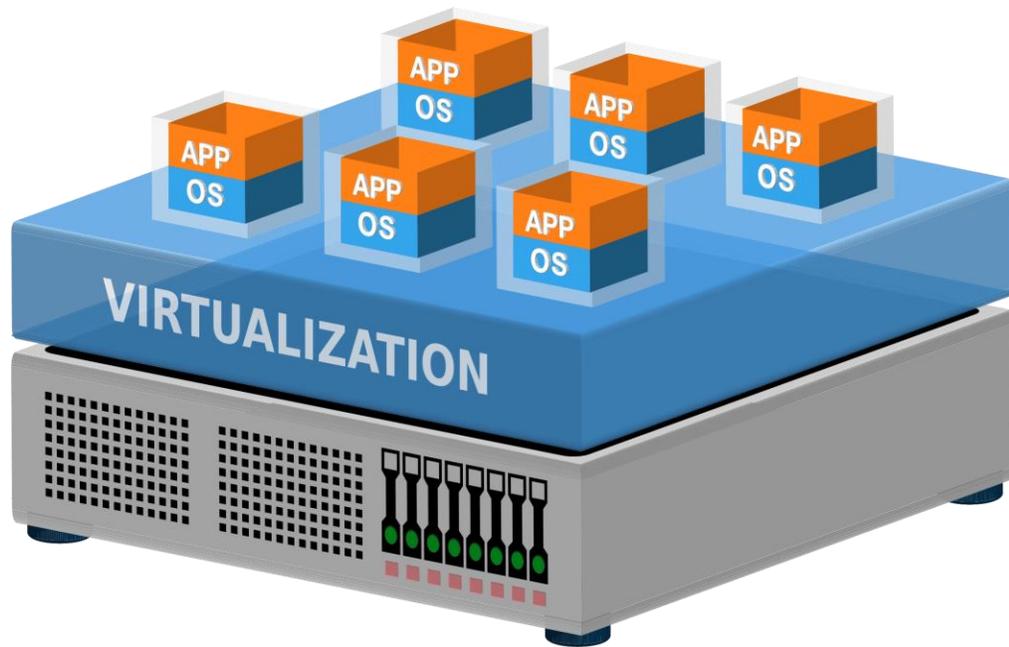
# Такси vs. Собственная машина



**Собственный Data-Center (On-Premise):** Это как личный автомобиль. Вы платите полную стоимость заранее, несете расходы на страховку, бензин, техобслуживание и ремонт. Машина всегда ваша, даже когда стоит в гараже.

**IaaS:** Это как такси или каршеринг. Вы платите только за поездку (время и расстояние). Вам не нужно покупать машину, менять масло или чинить двигатель. Вызвали, доехали, забыли. За парк, бензин и исправность машин беспокоится компания.

# Основные компоненты IaaS: вычислительные мощности



## **Виртуальные серверы (Виртуальные машины, VM)**

Это основа IaaS. С помощью технологии виртуализации один мощный физический сервер делится на множество изолированных виртуальных машин.

**vCPU (виртуальные процессоры):** Вычислительная мощность.

**RAM (оперативная память):** Объем памяти для работы приложений.

**ОС (Операционная система):** Windows Server, Ubuntu, CentOS и т.д.

**Тип инстанса:** Оптимизированные для вычислений, памяти, GPU или дискового I/O.

# Основные компоненты IaaS: хранилище

**Блочное хранилище:** Аналог виртуального жесткого диска (SSD/HDD). Подключается к VM и обеспечивает высокую скорость операций ввода-вывода. Идеально для ОС, баз данных и высокопроизводительных приложений.

**Файловое хранилище:** Сетевые файловые шарды. К одному тому могут получить доступ несколько VM одновременно. Подходит для общих документов.

**Объектное хранилище:** Данные как объекты в «ведрах» (buckets). Каждый объект имеет уникальный ID. Легко масштабируемо, надежно и дешево. Бэкапы, медиафайлы, статический контент сайтов.



# Основные компоненты IaaS: сети

**VPC (Virtual Private Cloud):** Изолированный сектор в облаке, где происходит развертывание ресурсов.

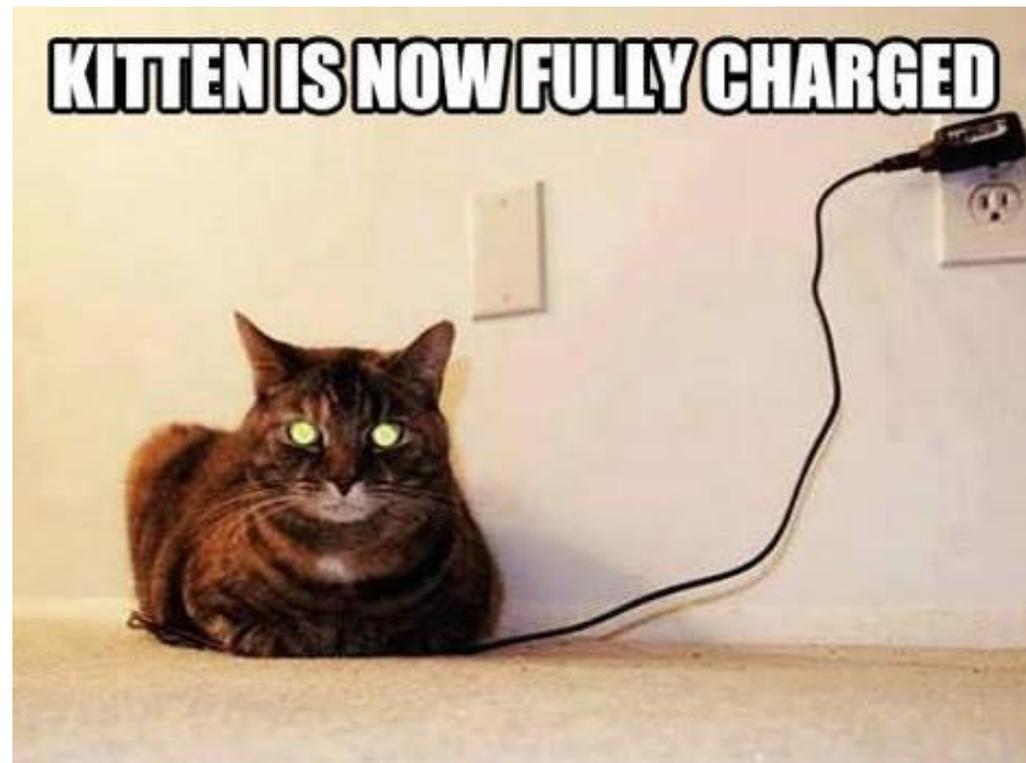
**IP-адресация:** Назначение публичных и частных IP-адресов.

**Группы безопасности (Firewalls):** Виртуальные фаерволы для контроля входящего и исходящего трафика на уровне виртуальной машины.

**Балансировщики нагрузки:** Равномерно распределяют входящий трафик между несколькими виртуальными машинами для обеспечения отказоустойчивости и высокой доступности.



А хороший вопрос кстати.. А как быть с облаком, есть отвалится соединение Интернет?



# Дополнительные компоненты IaaS



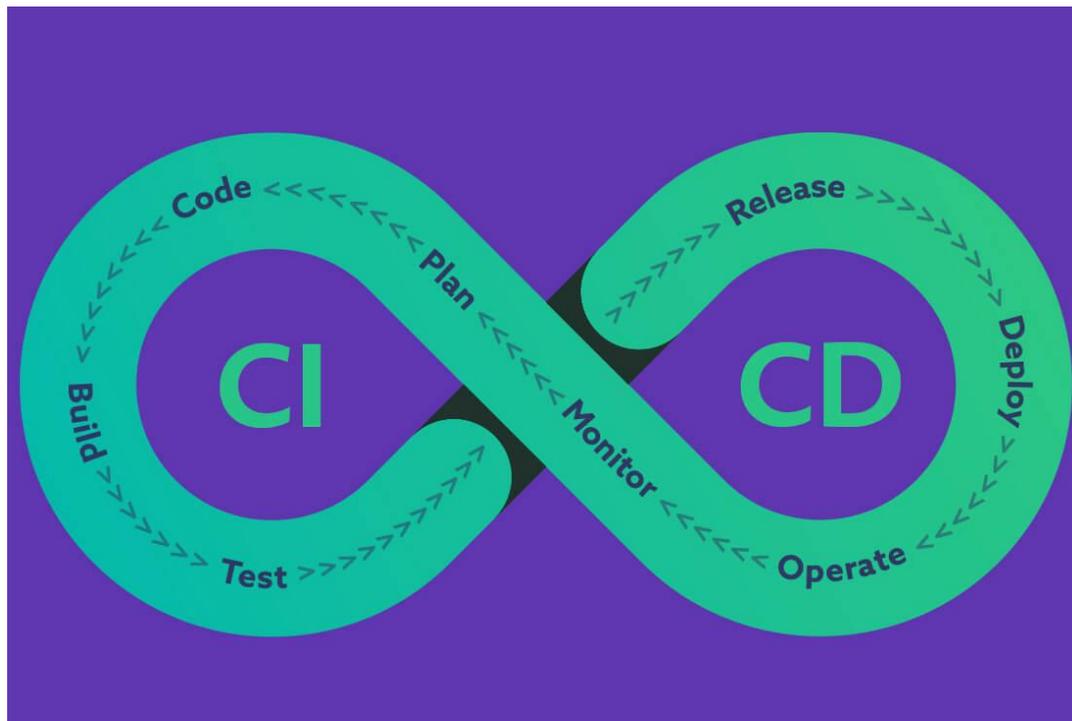
**Сопутствующие услуги, которые стали стандартами:**

**Резервное копирование и восстановление:** Автоматическое создание снимков (snapshots) виртуальной машины и данных в определенные моменты времени.

**Мониторинг и алертинг:** Встроенные инструменты для отслеживания загрузки CPU, потребления памяти, дискового пространства и сетевого трафика. Уведомления при превышении установленных порогов.

**Идентификация и управление доступом (IAM):** Система, которая контролирует, кто (пользователи или службы) и к каким ресурсам имеет доступ, и что они могут с ними делать.

# IaaS, как «полигон» для CI/CD



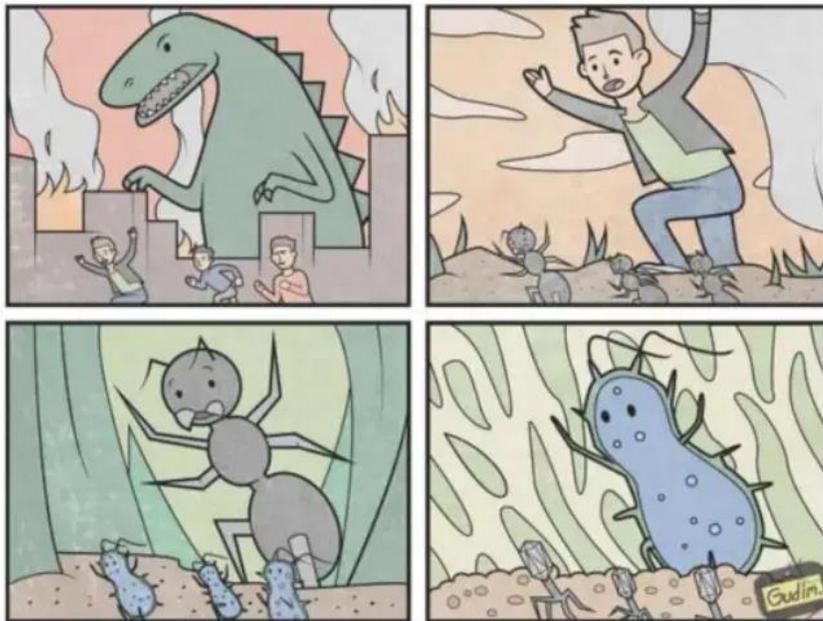
**Согласованность сред от разработки до продакшена.**

Проблема: "На моем ноутбуке код работал, а на сервере - нет".

**Решение с IaaS:** Инфраструктура для продакшена (серверы, ОС, софт) в виде кода. Этот же код используется для создания **идентичных** сред для разработки, тестирования и релиза. Это устраняет "эффект дрифта" конфигураций и делает развертывание предсказуемым.

# Масштабирование IaaS в CI/CD

## Проблема масштабирования



### Эластичность для сложных стратегий развертывания

IaaS позволяет реализовать продвинутые стратегии деплоя без простоев:

**Сине-зеленое развертывание:** Новая версия приложения ("зеленая" среда) разворачивается параллельно со старой ("синяя"). После тестирования трафик переводится на "зеленую" среду.

**Canary-релизы:** Новая версия разворачивается для небольшой группы пользователей. IaaS позволяет легко направить лишь часть трафика на новые экземпляры.

**Автомасштабирование:** Количество серверов автоматически увеличивается при росте нагрузки и уменьшается при ее спаде.

# IaC, infrastructure as code

**Ключевая технология: IaC (Infrastructure as Code)** - это практика управления и предоставления инфраструктуры с помощью конфигурационных файлов, а не вручную через интерфейс.

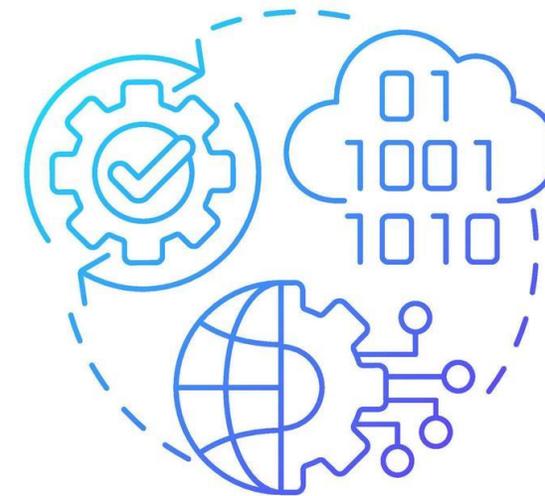
**Популярные инструменты:** Terraform, AWS CloudFormation, Ansible.

**Версионность:** Конфигурации можно хранить в Git, отслеживать изменения и откатываться.

**Повторяемость:** Идентичная среда разворачивается каждый раз.

**Документирование:** Код сам является документацией о том, из чего состоит ваша инфраструктура.

**Скорость:** Развертывание сложных сред одной командой.



**INFRASTRUCTURE-  
AS-CODE**

# Примеры IaaS: Amazon Web Services

## **AWS (Amazon Web Services)**

Один из первых массовых IaaS сервисов.

**EC2 (Elastic Compute Cloud):** Виртуальные серверы. Основная вычислительная единица.

**EBS (Elastic Block Store):** Блочное хранилище для EC2.

**S3 (Simple Storage Service):** Объектное хранилище, отраслевой стандарт.

**VPC (Virtual Private Cloud):** Фундамент для сетевой изоляции и безопасности.



# Примеры IaaS: Microsoft Azure

## **Microsoft Azure**

Популярная IaaS с глубокой интеграцией в экосистему Microsoft.

**Virtual Machines:** Виртуальные серверы с поддержкой Windows и Linux.

**Managed Disks:** Блочное хранилище.

**Blob Storage:** Объектное хранилище, аналог S3.

**Virtual Network:** Аналог VPC.



# Примеры IaaS: Google Cloud Platform

## **Google Cloud Platform (GCP)**

Инновативный в области данных и искусственного интеллекта.

**Compute Engine:** Виртуальные серверы.

**Persistent Disks:** Блочное хранилище.

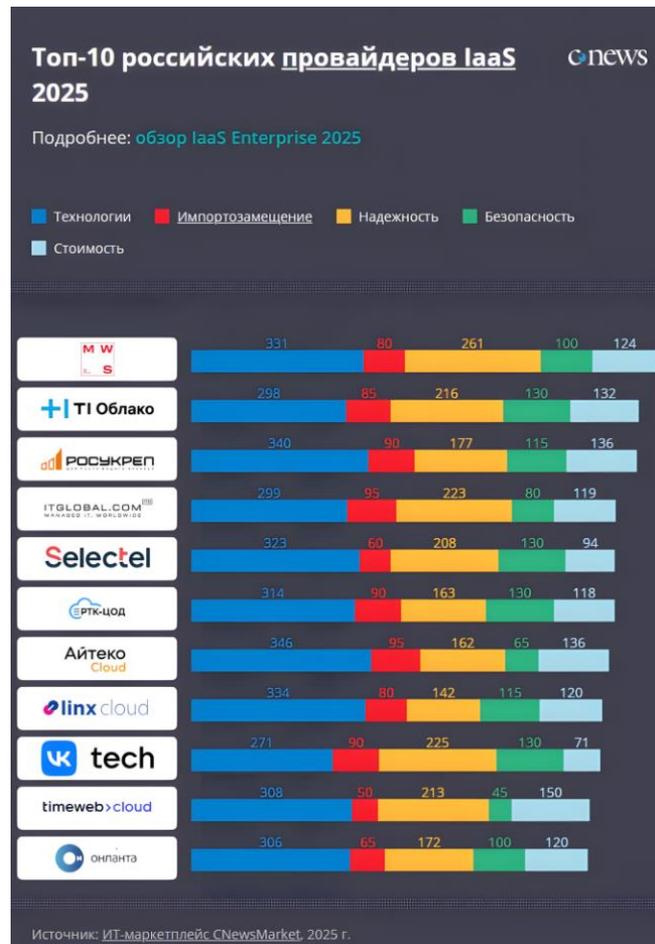
**Cloud Storage:** Объектное хранилище.

**Google Kubernetes Engine (GKE):** Управляемый сервис для оркестрации контейнеров.



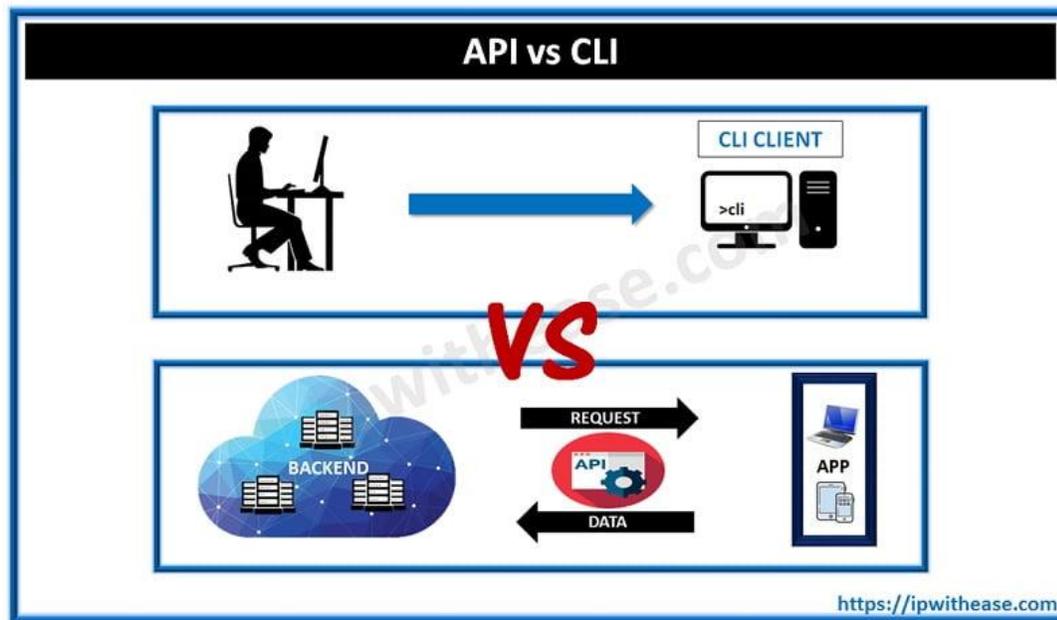
# Google Cloud Platform

# Сегмент IaaS в РФ



Источник: [https://www.cnews.ru/reviews/rejting\\_provajderov\\_iaas\\_enterprise\\_2025](https://www.cnews.ru/reviews/rejting_provajderov_iaas_enterprise_2025)

# Работа с IaaS (CLI - API)



**CLI (Command Line Interface):** Набор инструментов для управления ресурсами провайдера прямо из терминала. Позволяет выполнять серийные операции.

**API (Application Programming Interface):** Программный интерфейс, который позволяет коду (например, из скрипта на Python или из пайплайна CI/CD) напрямую взаимодействовать с облачной платформой: создавать, изменять, удалять ресурсы.

# Простенький IaC в Terraформе для AWS, для базового понимания

```
# Указание на язык конфигурации Terraform
Hcl
# Объявляем провайдера (AWS)
provider "aws" {
  region = "us-east-1"
}

# Описываем ресурс (виртуальный сервер)
resource "aws_instance" "my_web_server" {
  ami          = "ami-0c02fb55956c7d316" # ID образа ОС (напри
  мер, Ubuntu)
  instance_type = "t3.micro"           # Тип инстанса (1 vCPU, 1
  GiB RAM)
  tags = {
    Name = "My-Terraform-Server"
  }
}
```

После команды terraform apply Terraform свяжется с AWS API и создаст облачный сервер в указанной конфигурации. Чтобы изменить конфигурацию, вы меняете код и применяете снова. Чтобы уничтожить инстанс сервера используется команда terraform destroy

# Работа с IaaS(CLI - API)

- ✓ **Гибкость и масштабируемость:** Ресурсы можно увеличить или уменьшить за несколько кликов или автоматически в ответ на нагрузку.
- ✓ **Экономическая эффективность (OpEx):** Отсутствие капитальных затрат, оплата по факту использования.
- ✓ **Гибкость: Высокая доступность и отказоустойчивость:** Провайдеры имеют геораспределенные дата-центры. Встроенные механизмы репликации данных и восстановления после сбоев.
- ✓ **Скорость развертывания:** Новая IT-инфраструктура разворачивается за минуты, ускоряя выход на рынок.
- ✓ **Фокус на бизнесе:** IT-отделы перестают быть "ремонтной бригадой" для серверов и могут сосредоточиться на разработке и внедрении инноваций.
  
- ✗ **Зависимость от провайдера (Vendor Lock-in):** Миграция с одной облачной платформы на другую может быть сложной и дорогой из-за использования уникальных сервисов и API.
- ✗ **Непредсказуемые затраты:** При плохом управлении и отсутствии мониторинга "утечки" ресурсов могут привести к неожиданно большим счетам.
- ✗ **Требует новых навыков:** Командам необходимо изучать специфические сервисы провайдеров, инструменты IaC и DevOps-практики.

# Чтение на дом

- Д. Хамбл, Д. Фарли, Непрерывное развертывание ПО, стр. 259-265, 273-316.

Спасибо за внимание!