Skillbox

Cloud native. Cloud agnostic. The Twelve-Factor App

Максим Чернухин

Software Architect

АО «Альфа-Банк»

Skillbox

На этом уроке мы узнаем о

- The Twelve-Factor App
- Cloud native
- Cloud agnostic

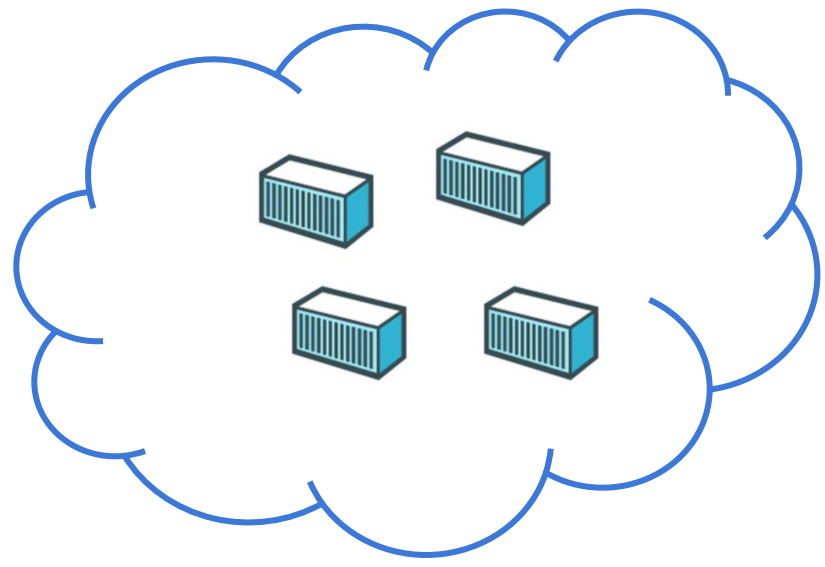
Тренды

- Cloud native
- Cloud agnostic / Multicloud

Skillbox

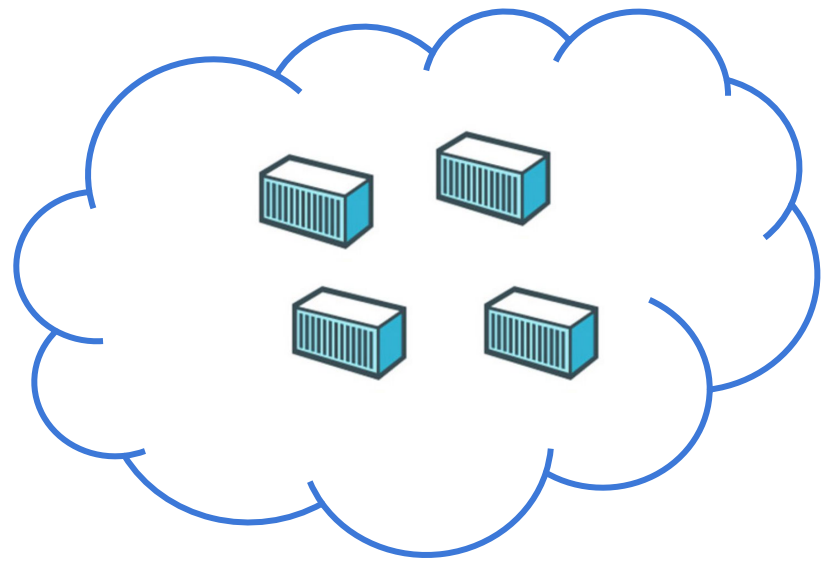
Cloud native

- PaaS готовая инфраструктура



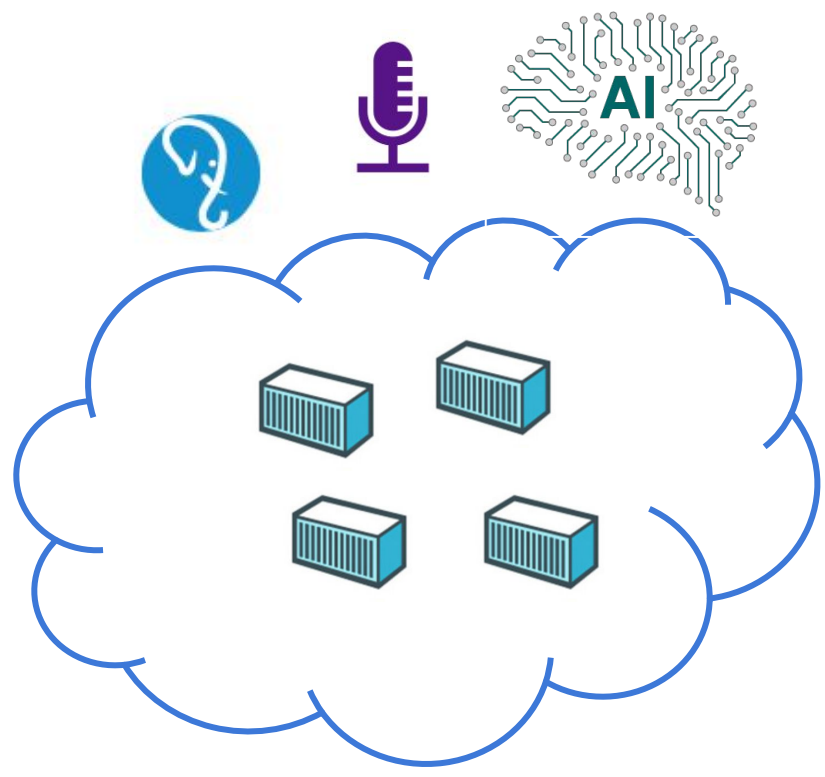
Cloud native

- PaaS готовая инфраструктура
- Настройки передаёт среда исполнения



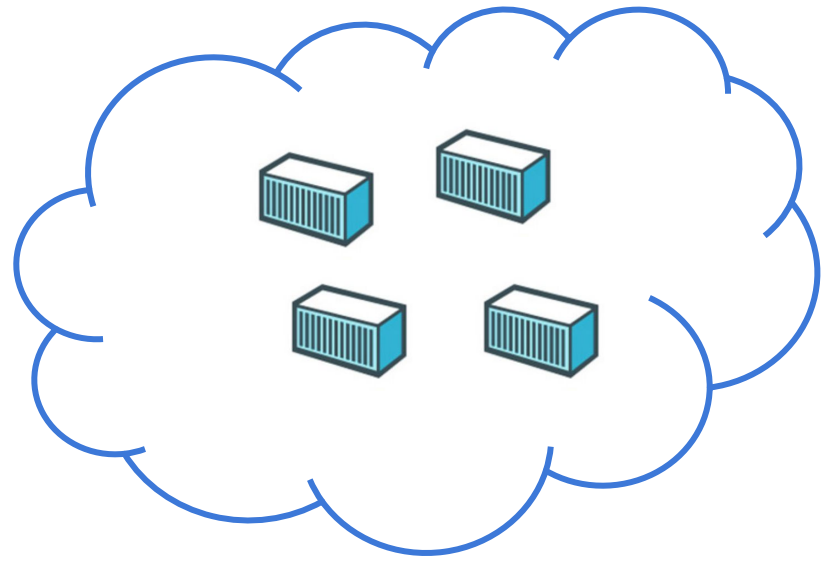
Cloud native

- PaaS готовая инфраструктура
- Настройки передаёт среда исполнения
- Готовые облачные сервисы



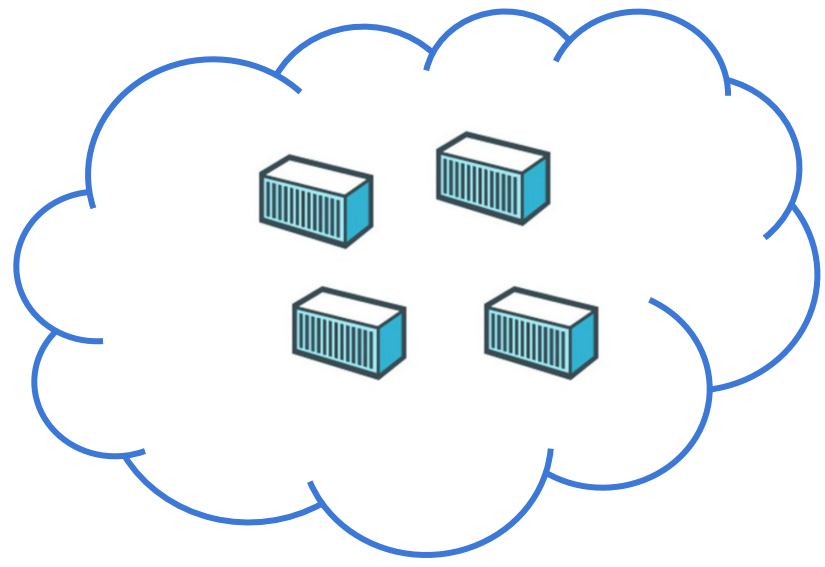
Cloud agnostic

- PaaS готовая инфраструктура



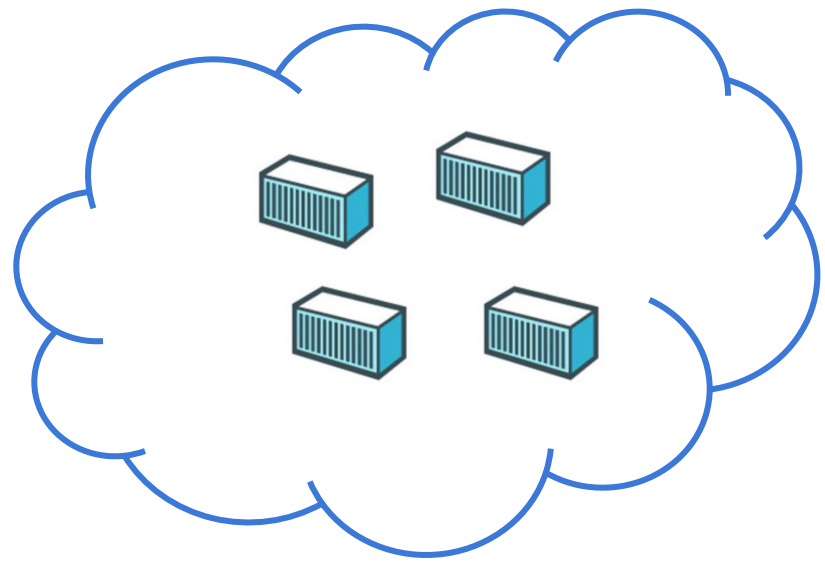
Cloud agnostic

- PaaS готовая инфраструктура
- Настройки передаёт среда исполнения



Cloud agnostic

- PaaS готовая инфраструктура
- Настройки передаёт среда исполнения
- Остальное реализовываем сами



Skillbox

The Twelve-Factor App

Методология для создания SaaS-приложений (By Heroku)

Skillbox

The Twelve-Factor App

Методология для создания SaaS-приложений

Принципы:

- декларативный формат для описания процесса установки и настройки

Skillbox

The Twelve-Factor App

Методология для создания SaaS-приложений

Принципы:

- декларативный формат для описания процесса установки и настройки
- максимальная переносимость между средами выполнения

Skillbox

The Twelve-Factor App

Методология для создания SaaS-приложений

Принципы:

- декларативный формат для описания процесса установки и настройки
- максимальная переносимость между средами выполнения
- развёртывание на современных облачных платформах

Skillbox

The Twelve-Factor App

Методология для создания SaaS-приложений

Принципы:

- декларативный формат для описания процесса установки и настройки
- максимальная переносимость между средами выполнения
- развёртывание на современных облачных платформах
- минимум расхождений между средой разработки и средой выполнения

Skillbox

The Twelve-Factor App

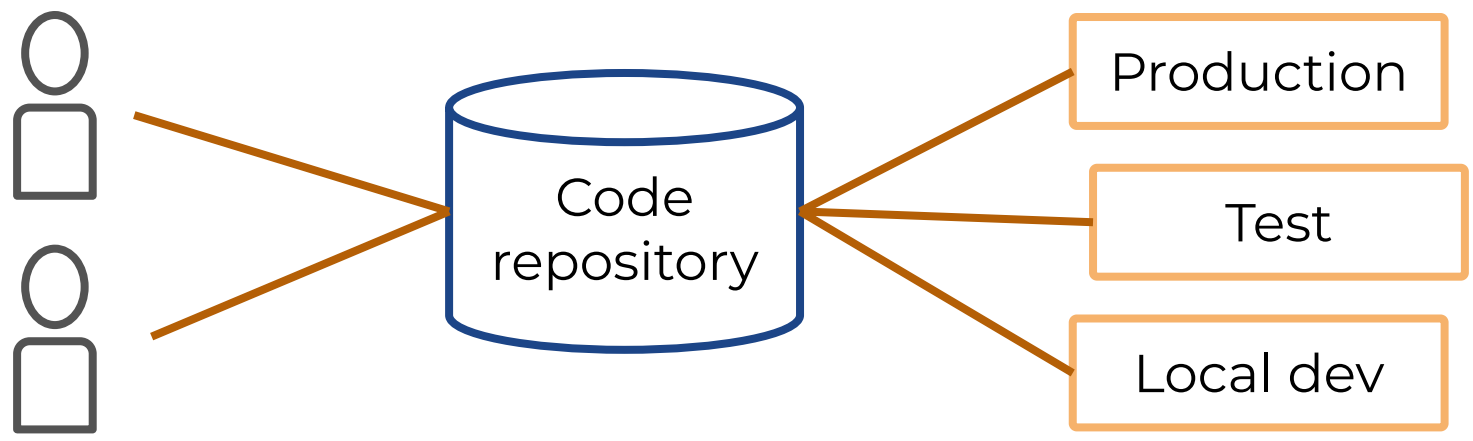
Методология для создания SaaS-приложений

Принципы:

- декларативный формат для описания процесса установки и настройки
- максимальная переносимость между средами выполнения
- развёртывание на современных облачных платформах
- минимум расхождений между средой разработки и средой выполнения
- масштабирование без существенных изменений в инструментах, архитектуре и практике разработки

Кодовая база

- Одно приложение (сервис) — один репозиторий
- Если надо переиспользовать код, необходимо создать отдельно библиотеку и подключить её через менеджера зависимостей
- Один репозиторий — множество развёртываний



Зависимости

- Все зависимости должны быть явно задекларированы
- Использование манифестов, где все зависимости отражены
- Декларируются и модули самой ОС, необходимые для работы приложения

Конфигурация

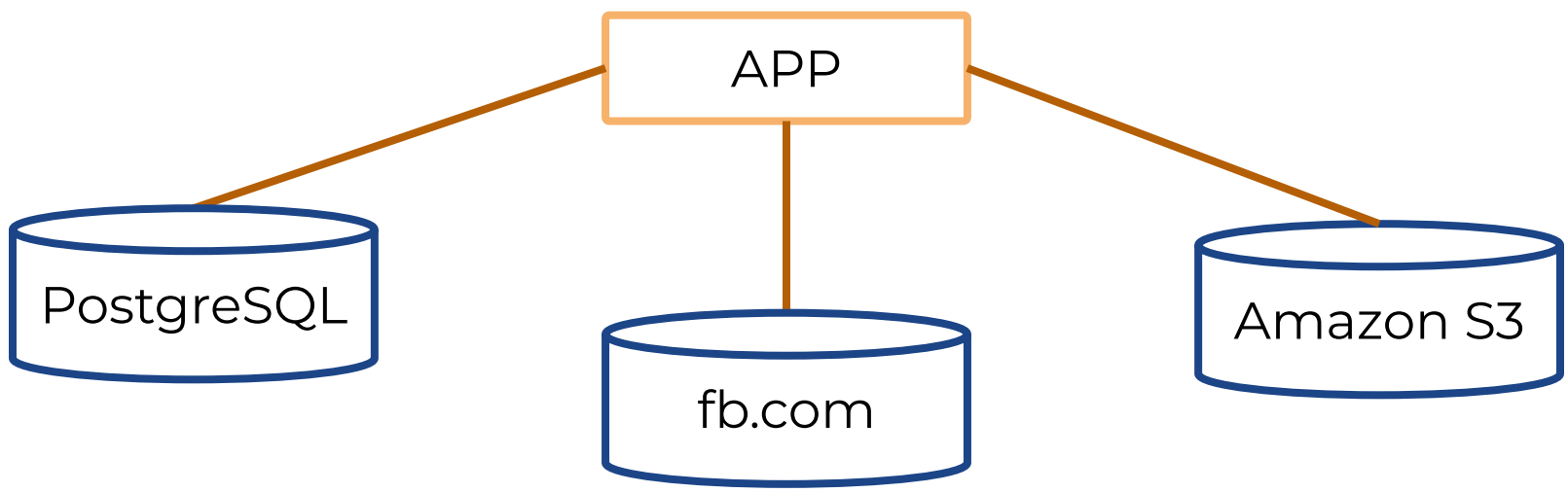
- Подключения к ресурсам (базы данных, удостоверяющие центры и т. д.)
- Регистрационные данные для подключения к внешним сервисам (Google, Amazon S3 и т. д.)
- Значения зависимы от среды исполнения (имя хоста, картинка бренда и т. д.)

Все конфигурации задаются через ENV-переменные.

Среда исполнения определяет, как будет работать приложение.

Сторонние службы (Backing Services)

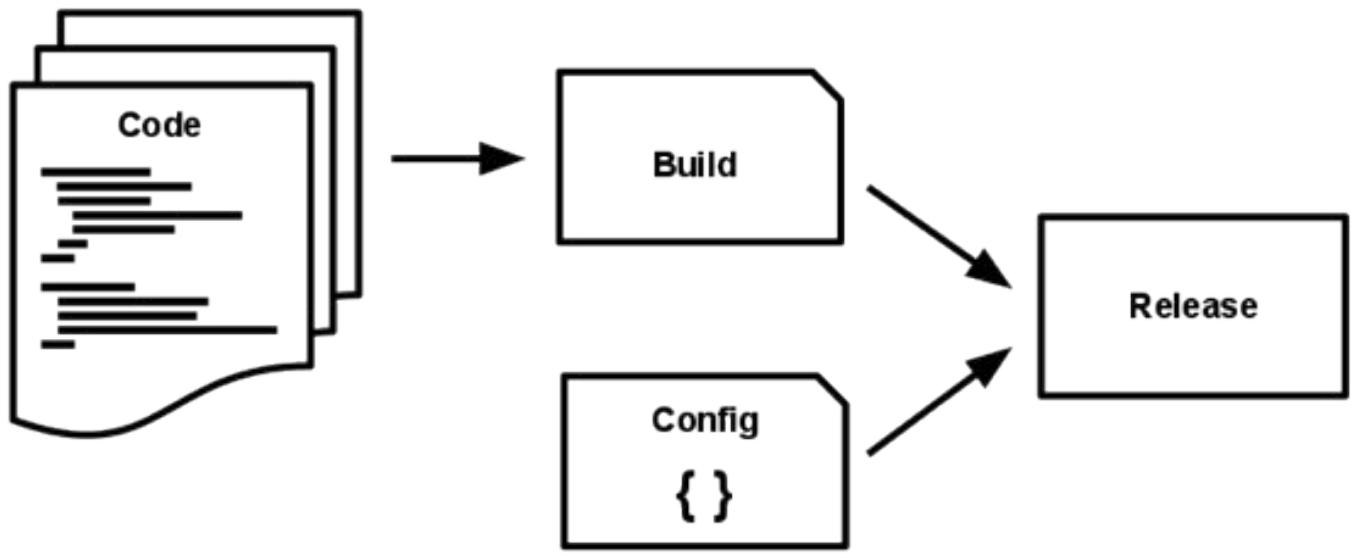
- Все сторонние службы, к которым надо обращаться по сети, рассматриваются как внешние ресурсы
- ENV-переменные для любого ресурса:
  - путь или расположение (URL и т. д.)
  - учётные данные



Skillbox

Сборка, релиз, выполнение

- Сборка — этап трансформации кода и внешних зависимостей в исполняемые файлы
- Релиз — объединение сборки и конфигурации конкретной среды исполнения
- Выполнение — запуск приложения в среде исполнения



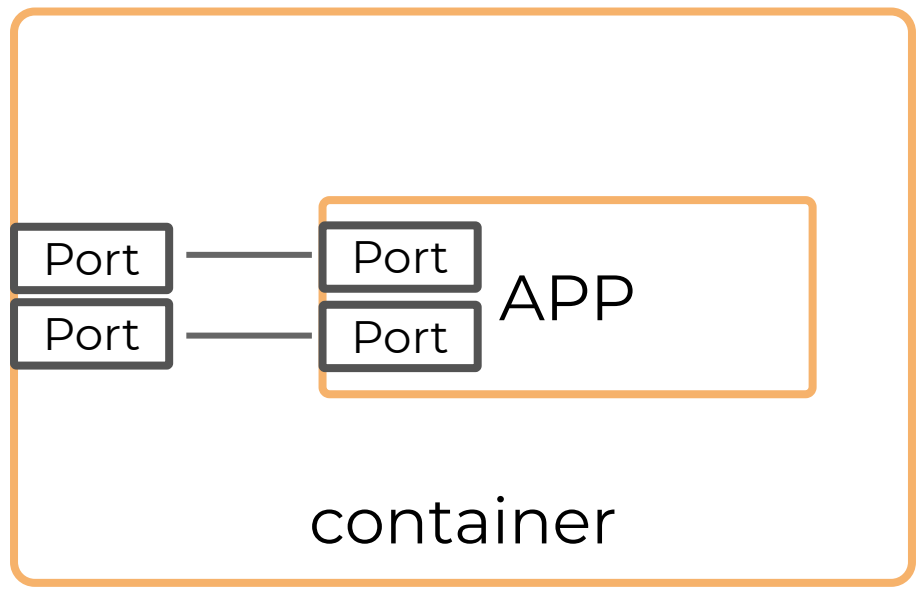
Процессы (Stateless)

- Запущенные процессы (приложения, сервисы) не должны хранить состояния
- Хранить состояние необходимо во внешних хранилищах (бд, кеш и т. д.)

Skillbox

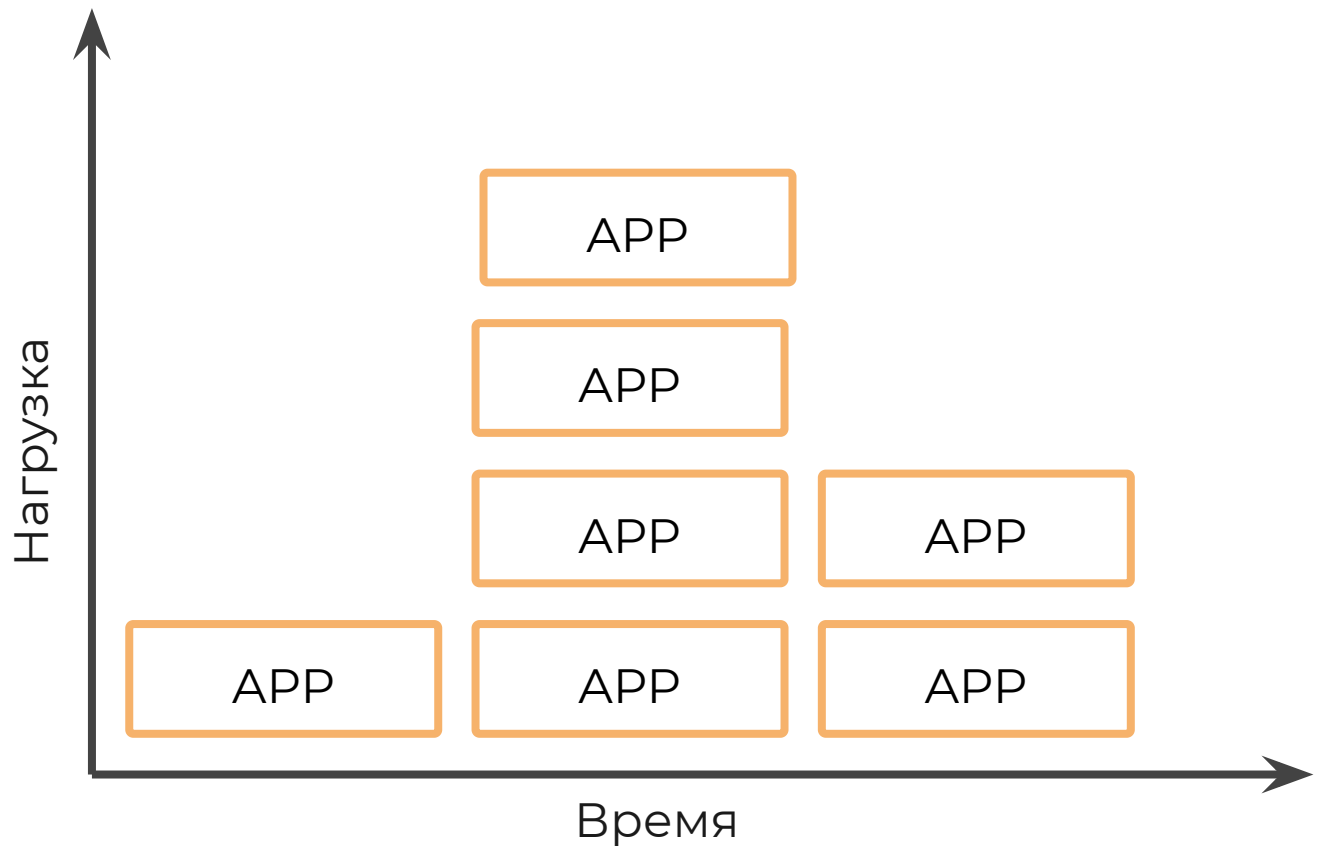
Привязка портов (Port binding)

Экспортируйте сервисы через привязку портов приложения к контейнеру.



Параллелизм

Приложение должно быть разработано так, чтобы при высокой нагрузке можно было запустить дополнительные экземпляры.



Утилизируемость (Disposability)

- Минимизация времени запуска приложения
- Корректная остановка приложения с остановкой прослушивания портов и завершением задач, которые были в работе (Graceful Shutdown)

Skillbox

Паритет окружений (Dev/prod)

Окружения разработки, промежуточного развёртывания (staging) и рабочего развёртывания (production) должны быть максимально похожими.

Типы различий:

- различие во времени
- различие персонала
- различие инструментов

Skillbox

Журналирование (Logs)

- Рассматривайте журнал логов как поток событий
- Все логи отправляются в stdout

Задачи администрирования

- Код задач должен лежать в репозитории
- Зависимости должны быть объявлены в манифесте
- Конфигурация задачи должна находиться в переменных окружения

Итоги

- Для того чтобы ваша система работала в облаках, она должна работать с инфраструктурой как с сервисом, который может меняться
- Инфраструктура определяет настройки вашего приложения
- Облака нужны для того, чтобы часть работ взять на себя, чаще всего это инфраструктура IaaS и PaaS, но бывают и другие сервисы
- Выбор надо принимать исходя из уникальности вашей задачи

Что дальше

- Поговорим о типах облаков
- Узнаем, какие бывают нюансы, если использовать разные подходы одновременно

Skillbox

Спасибо за внимание!