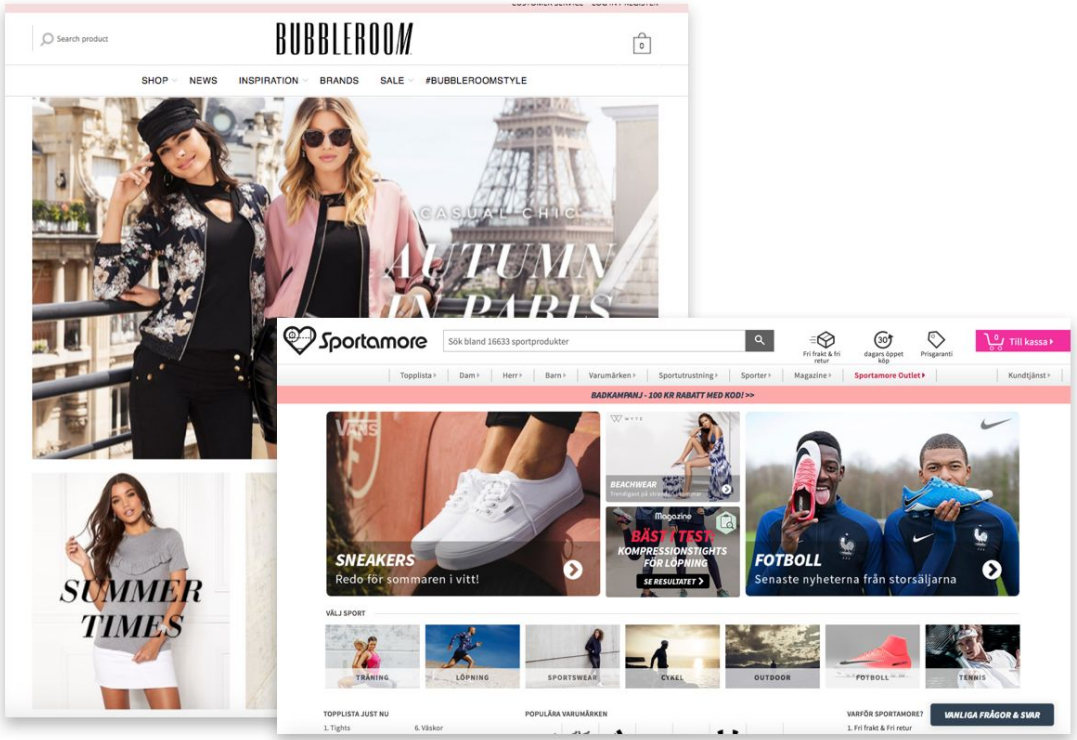
Дизайн систем

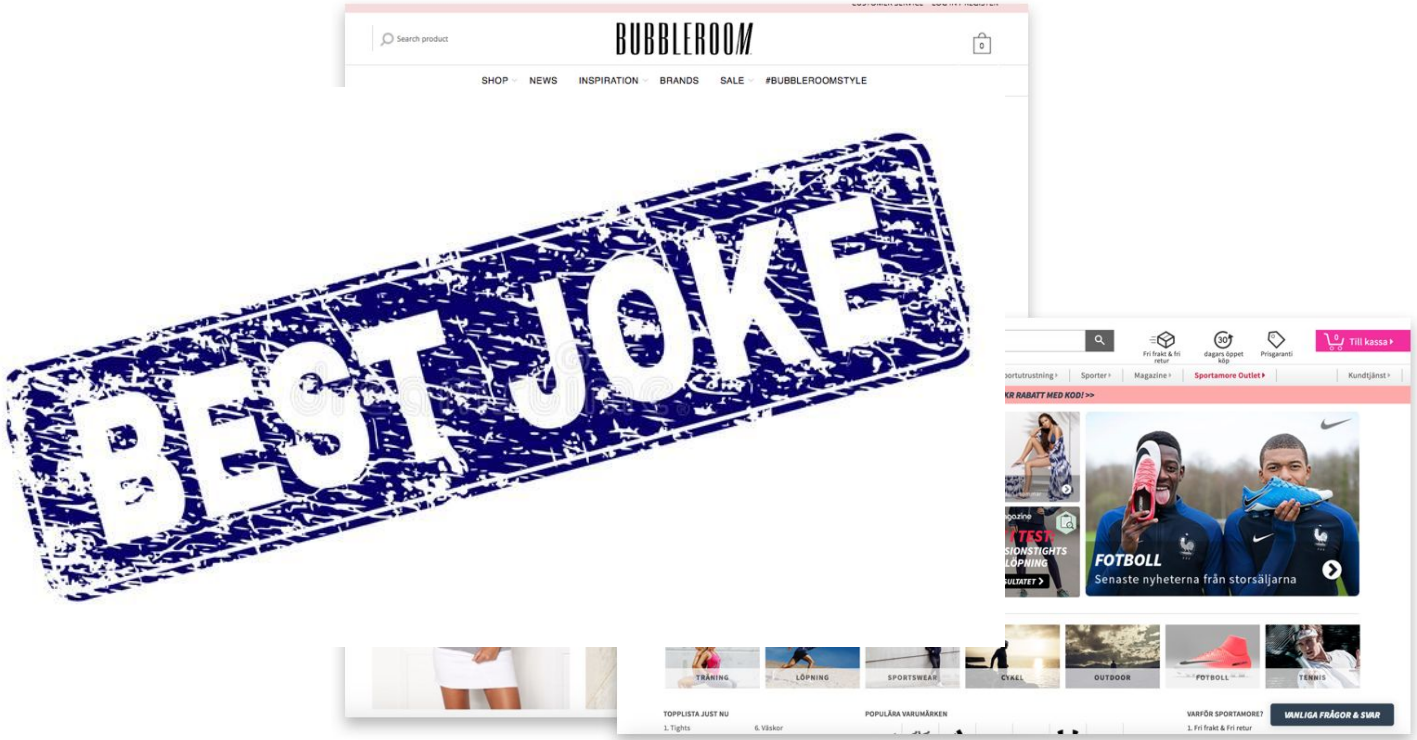
О чем курс

- Проектирование макетов дизайна сложных систем
- Верстка динамических web-страниц
- Современная типография
- Паттерны композиции - от лендингов до супер-аппов



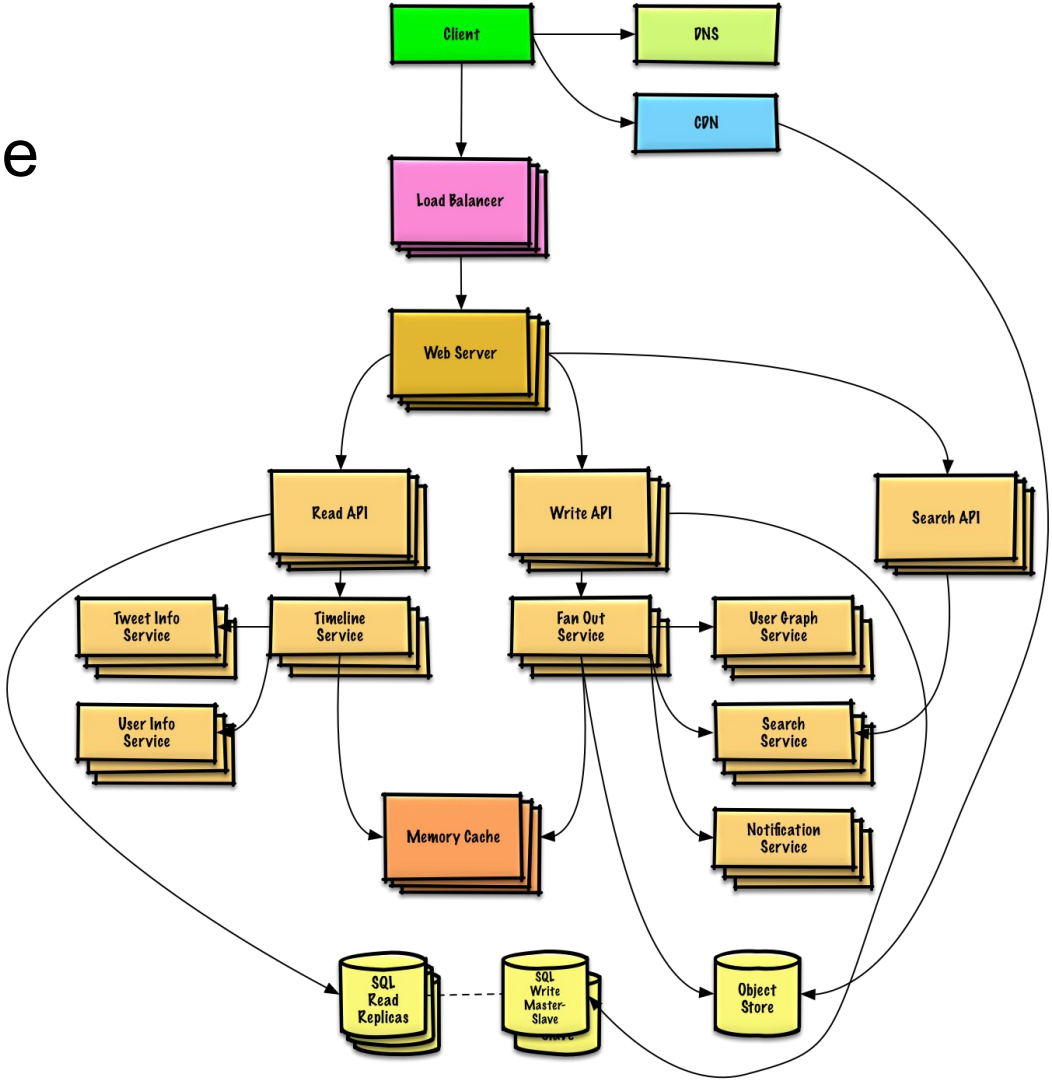
О чем курс

- Проектиров
  дизайна сло
- Верстка дин
  web-страни
- Современна
- Паттерны ко
  лендингов д



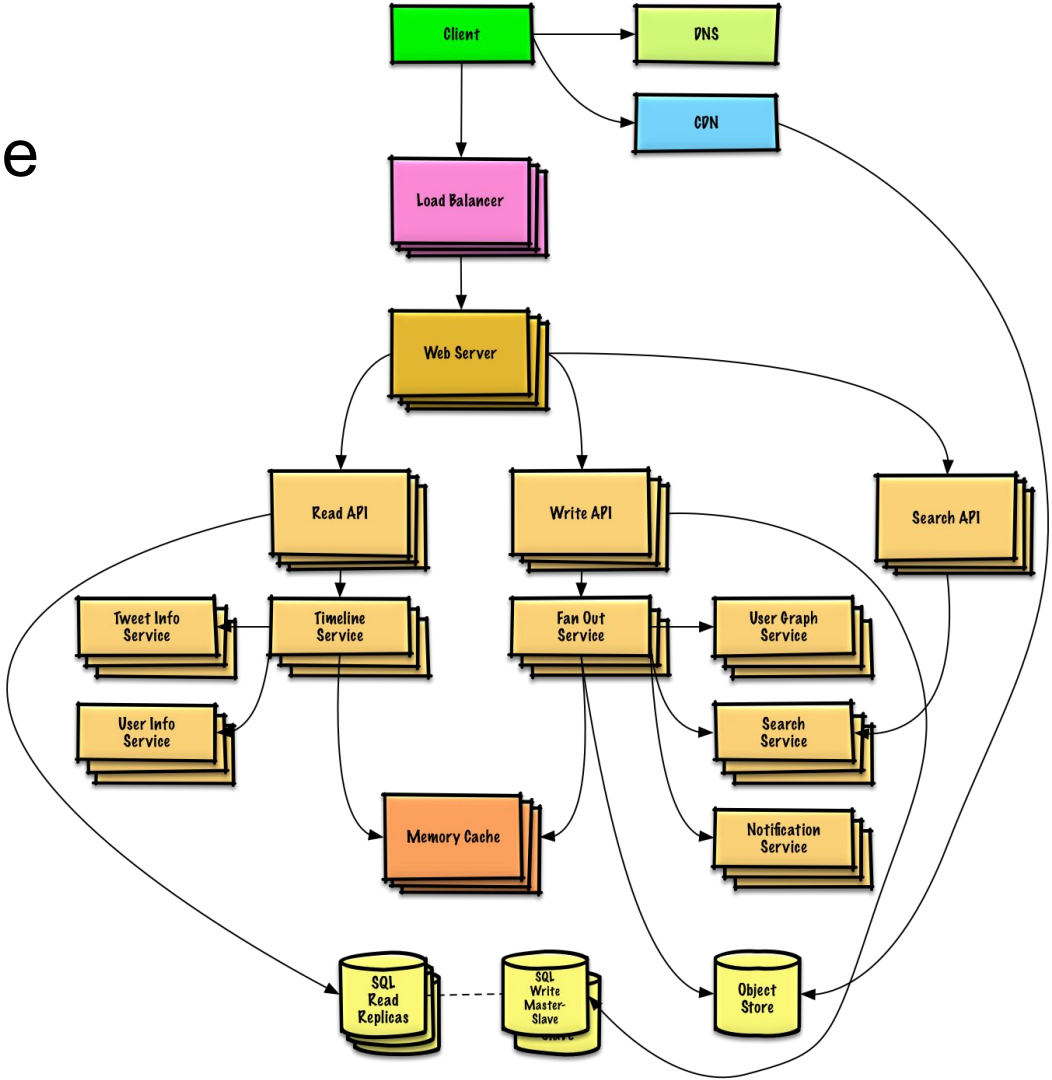
О чем курс на самом деле

Про то, как программировать работающие распределенные системы, которые решают бизнес-задачи



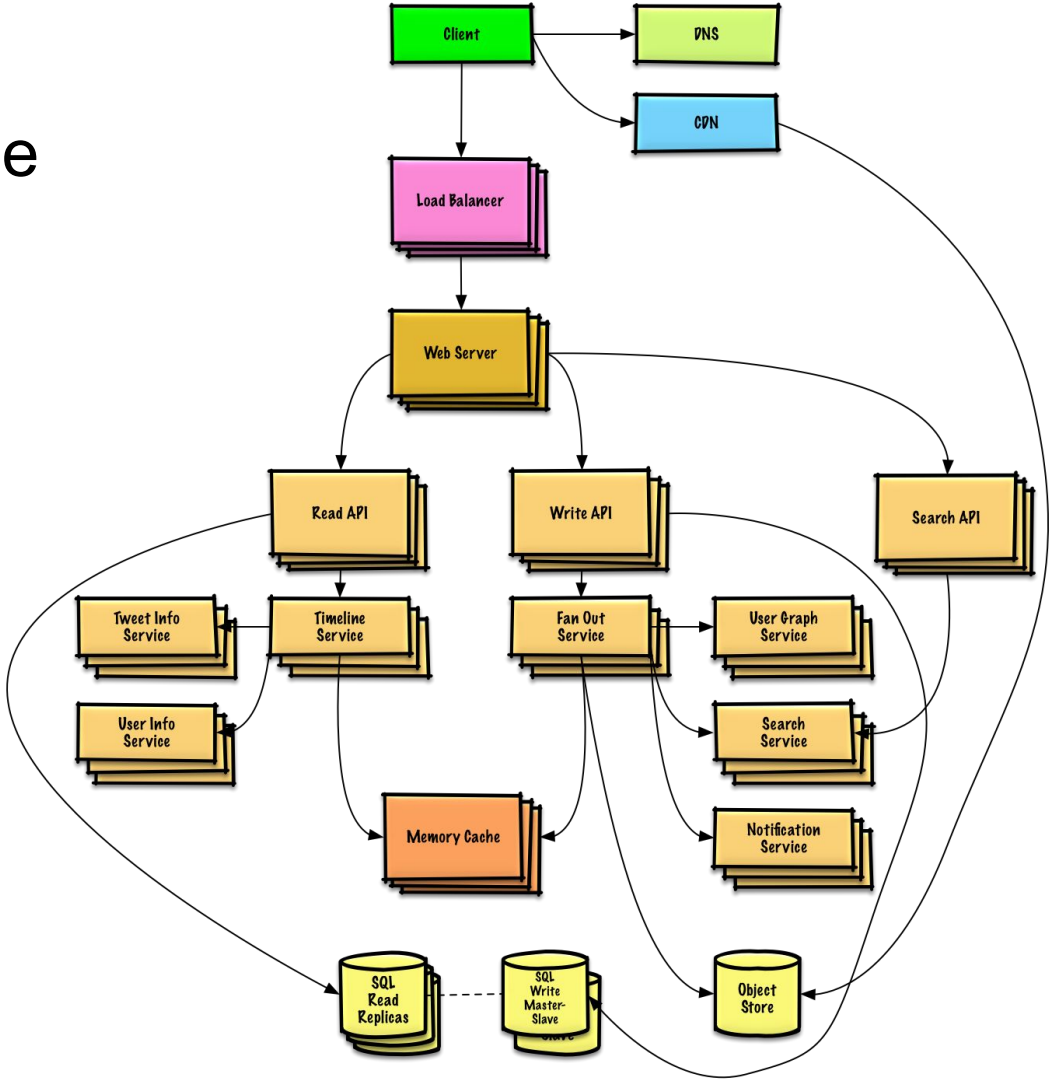
О чем курс на самом деле

Про то, как **программировать** работающие распределенные системы, которые решают бизнес-задачи



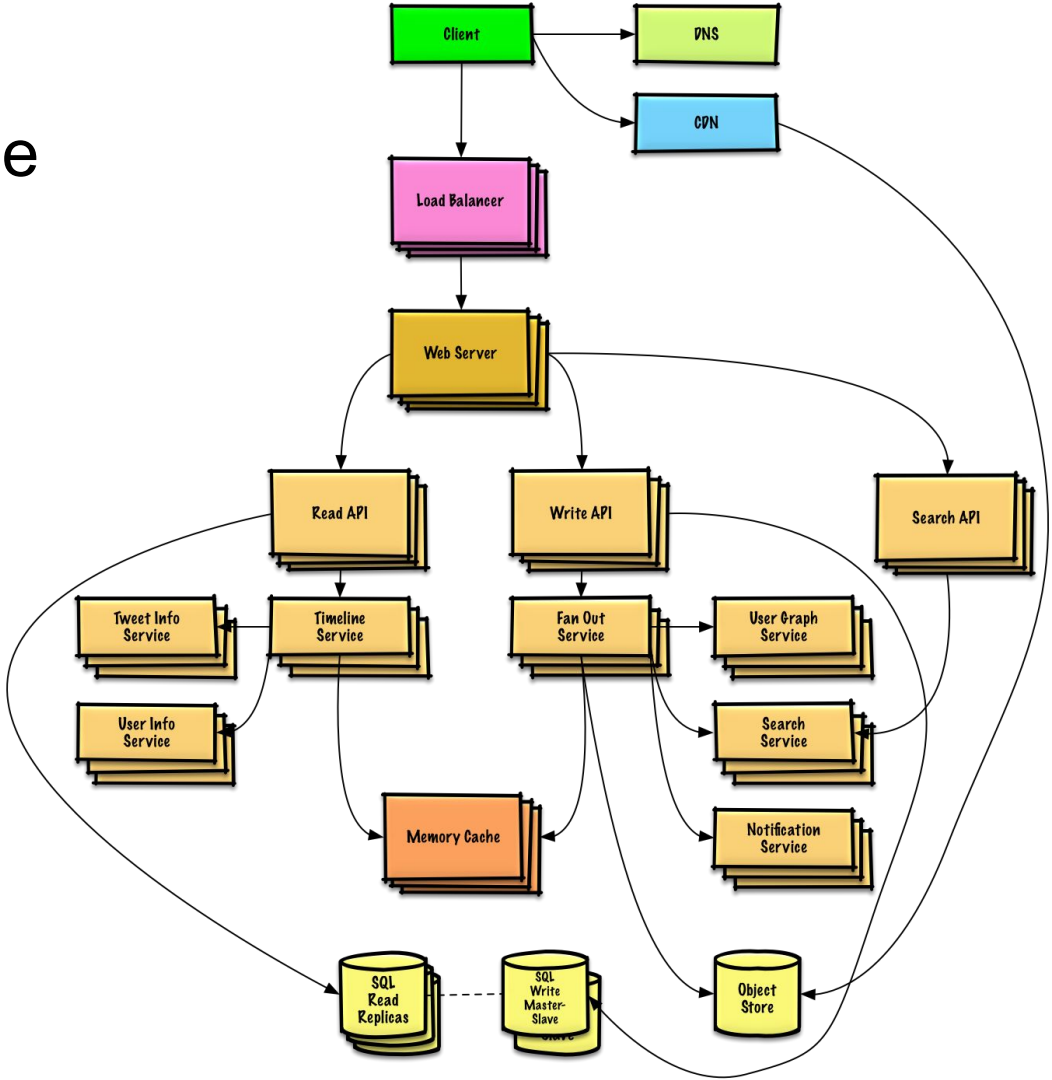
О чем курс на самом деле

Про то, как программировать **работающие распределенные системы**, которые решают бизнес-задачи



О чем курс на самом деле

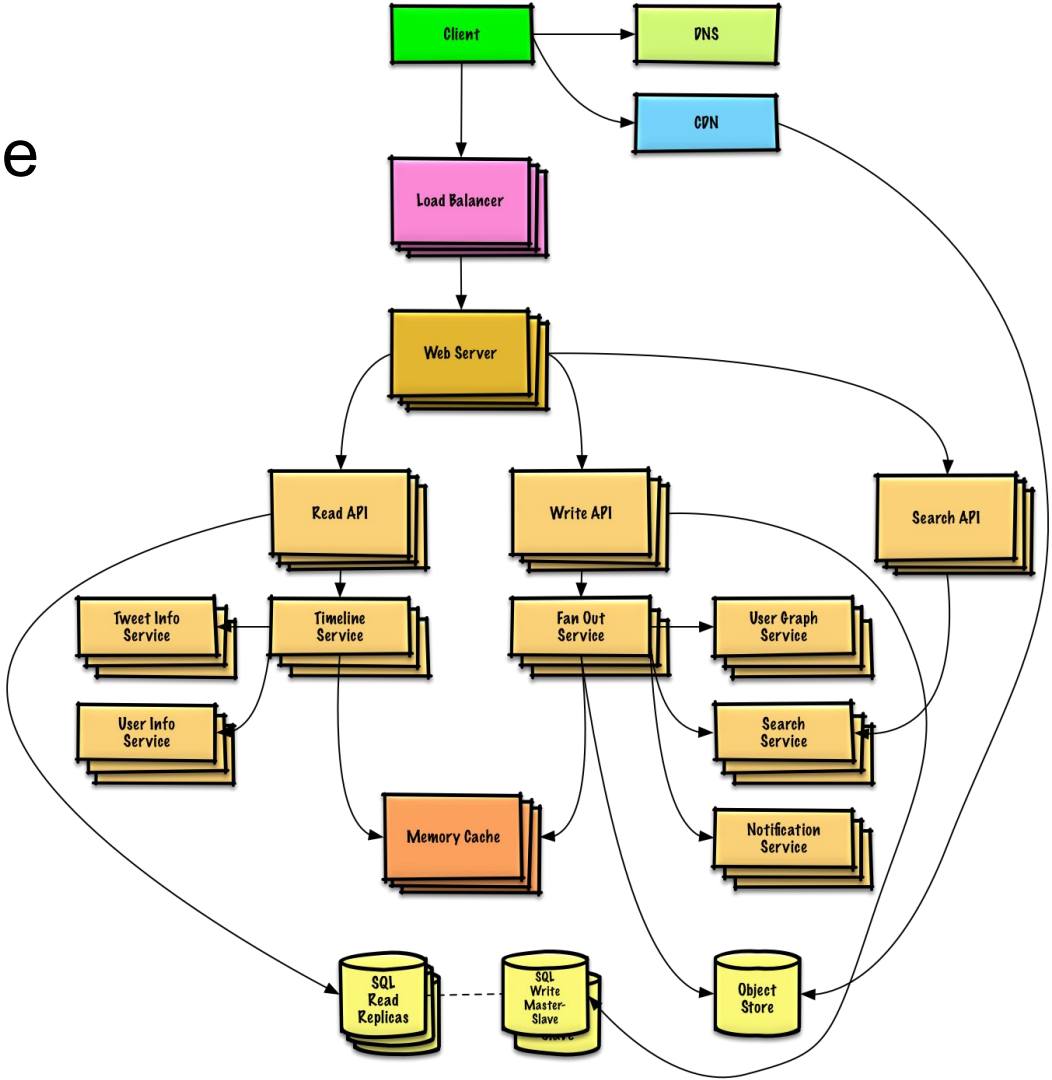
Про то, как программировать работающие распределенные системы, **которые решают бизнес-задачи**



О чем курс на самом деле

Про то, как программировать работающие распределенные системы, которые решают бизнес-задачи

А также про то, как проходить секцию “Дизайн” на собеседованиях :)



Об авторах

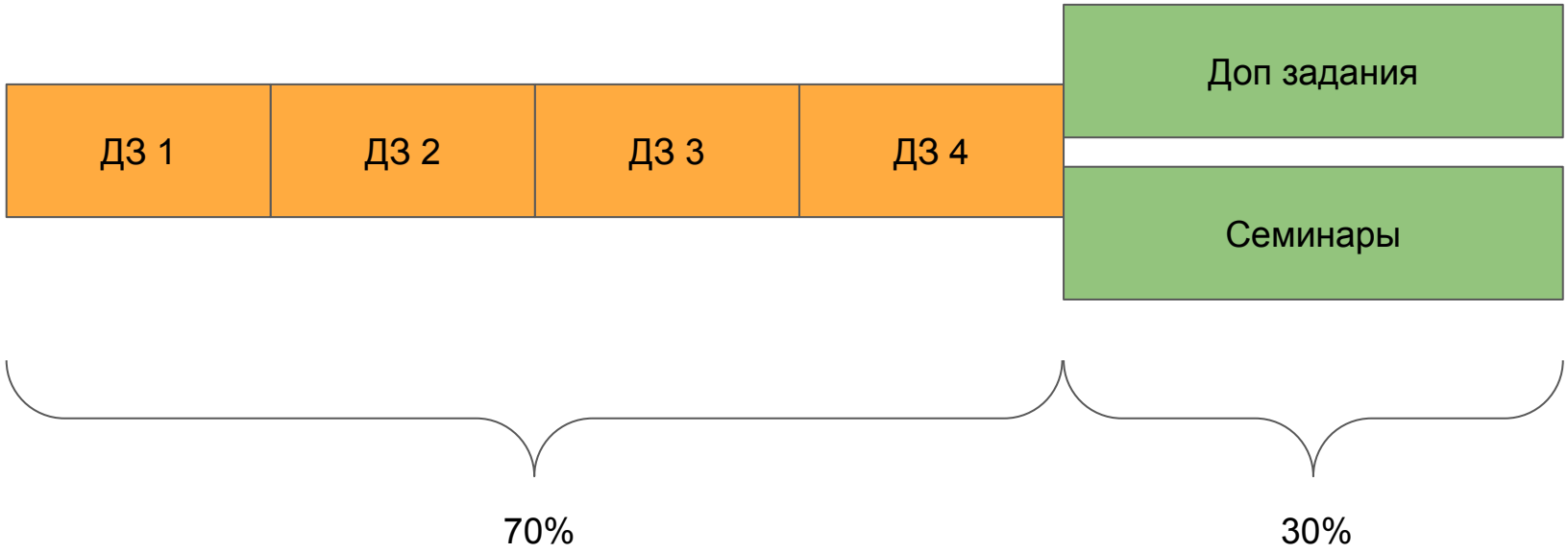
Лев Хотов
Старший разработчик в Joom

ex-Team lead в Bostongene
Выпускник ФКН ПМИ 2018

Алексей Космачев
Старший разработчик в Joom

ex-Architect в Neatsy Inc
ex-Team lead в Bostongene
Выпускник ФКН ПМИ 2018

Формула оценки

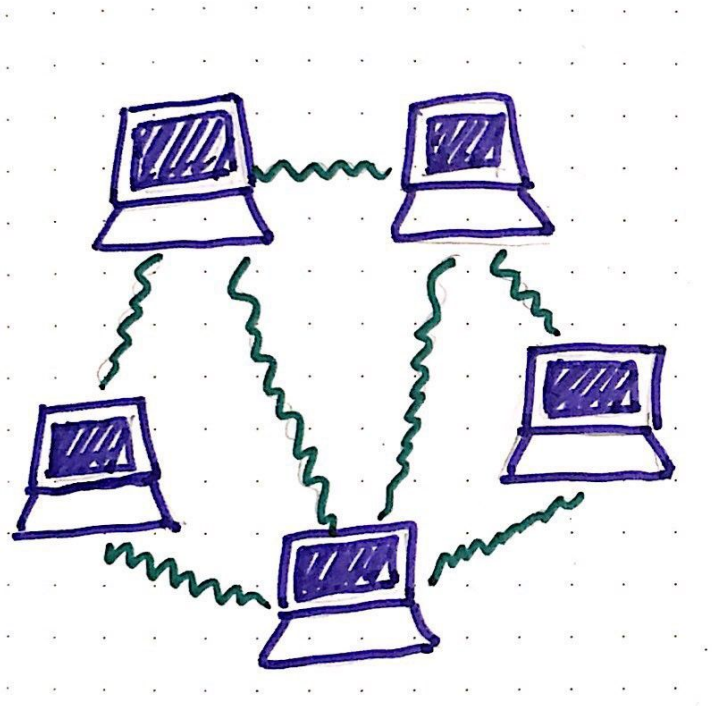


Домашние задания

Итог - законченный сервис

4 домашки - 4 этапа развития

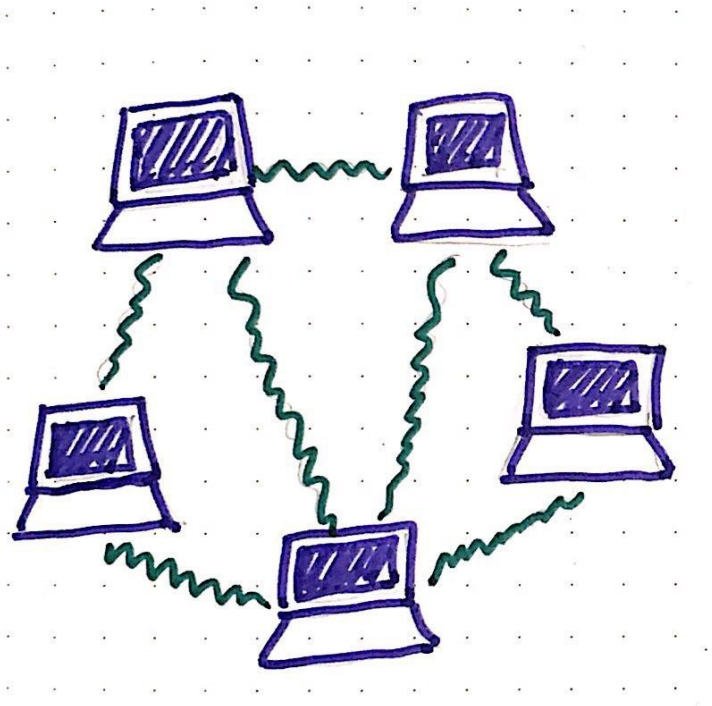
- Начальный MVP
- Масштабирование
- Добавление новой функциональности
- Доработка инфраструктуры



Домашние задания

Проверка:

- Автоматические тесты на корректную работу сервиса
- Нагрузочные тесты
- Выборочные code review



Доп задания и последние семинары

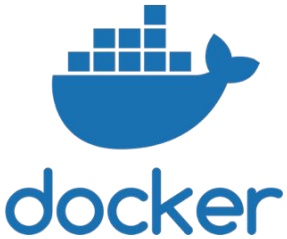
В процессе изучения материала - небольшие дополнительные задания

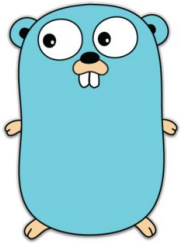
Проверка - в ручном режиме

Последние семинары - обсуждение самых частых вопросов из области дизайна систем

Проверка - в формате научного семинара

Набор инструментов



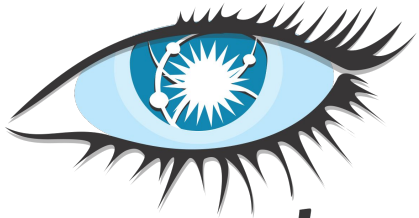


Go

redis

kubernetes

Prometheus

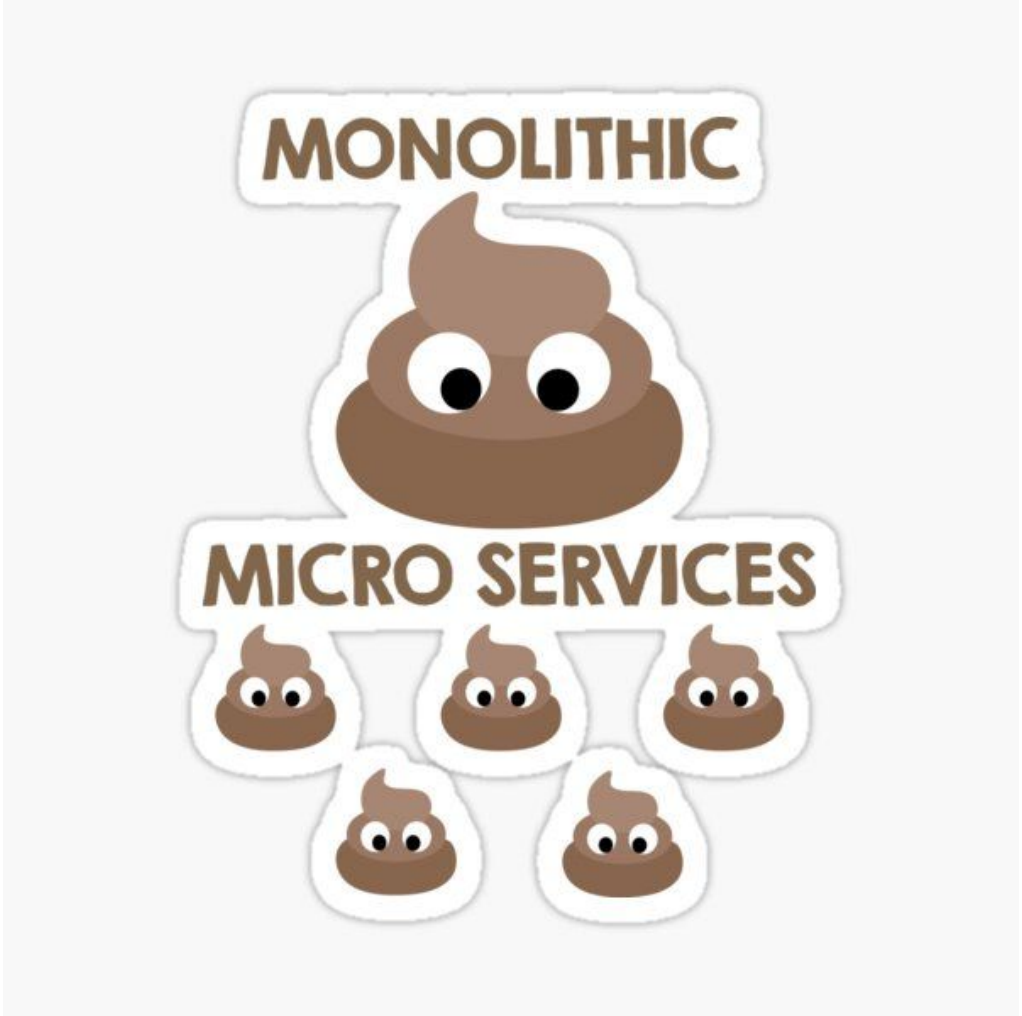


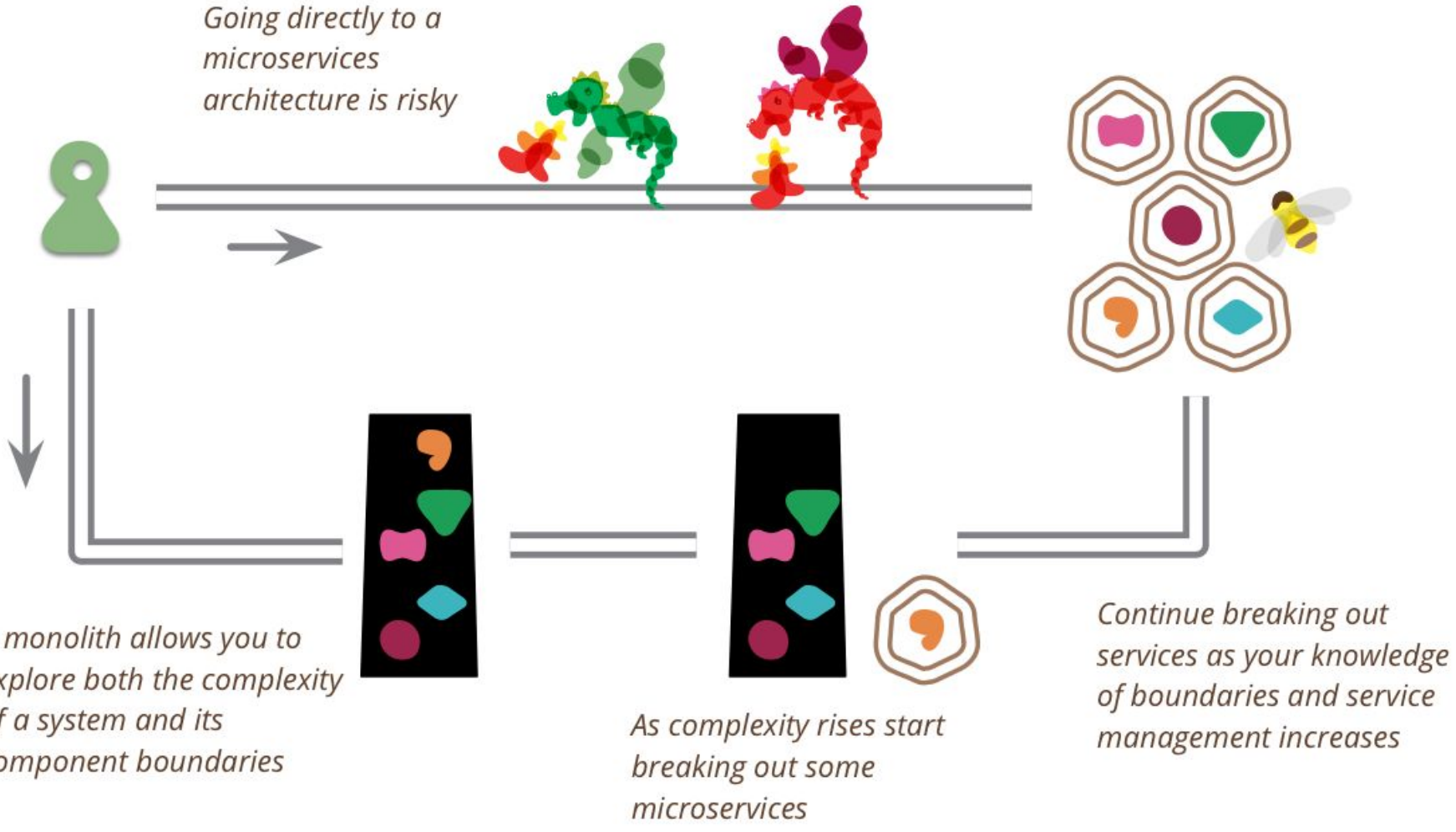
cassandra

Начнем сразу с микросервисов?

Начнем сразу с микросервисов?

НЕТ


MONOLITHIC
MICRO SERVICES


Going directly to a microservices architecture is risky
A monolith allows you to explore both the complexity of a system and its component boundaries
As complexity rises start breaking out some microservices
Continue breaking out services as your knowledge of boundaries and service management increases

Начинаем с монолита

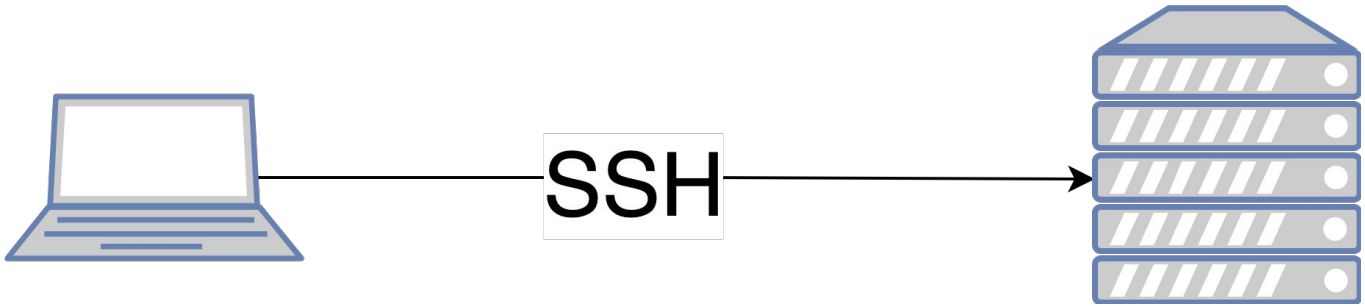
(но грамотно сделанного)

Начинаем с монолита

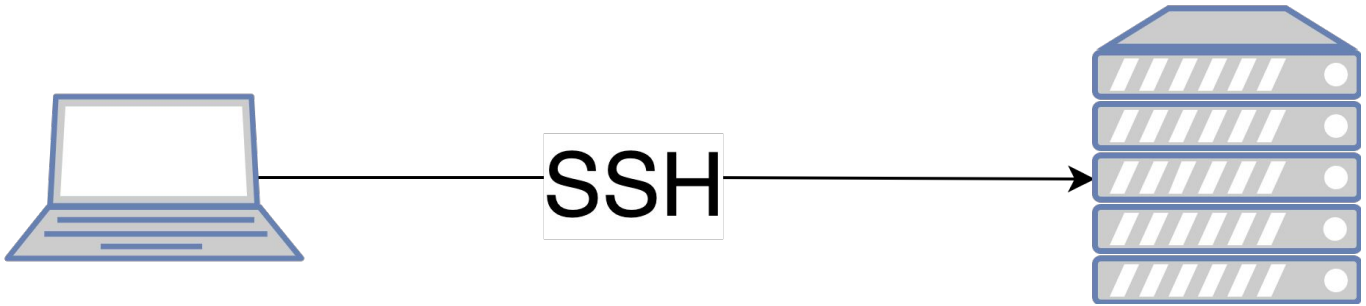
(но грамотно сделанного)

Как развернуть написанный монолит?

Deploy via SSH

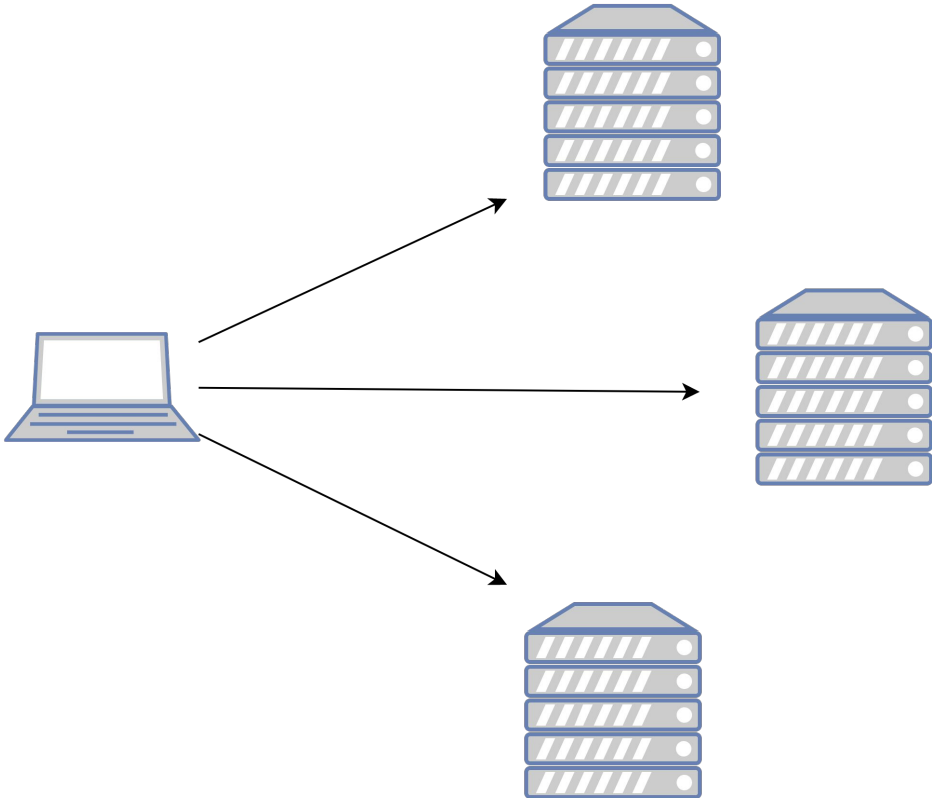


Deploy via SSH



Где собрать приложение?
Как установить зависимости?
Как выставить конфигурации?

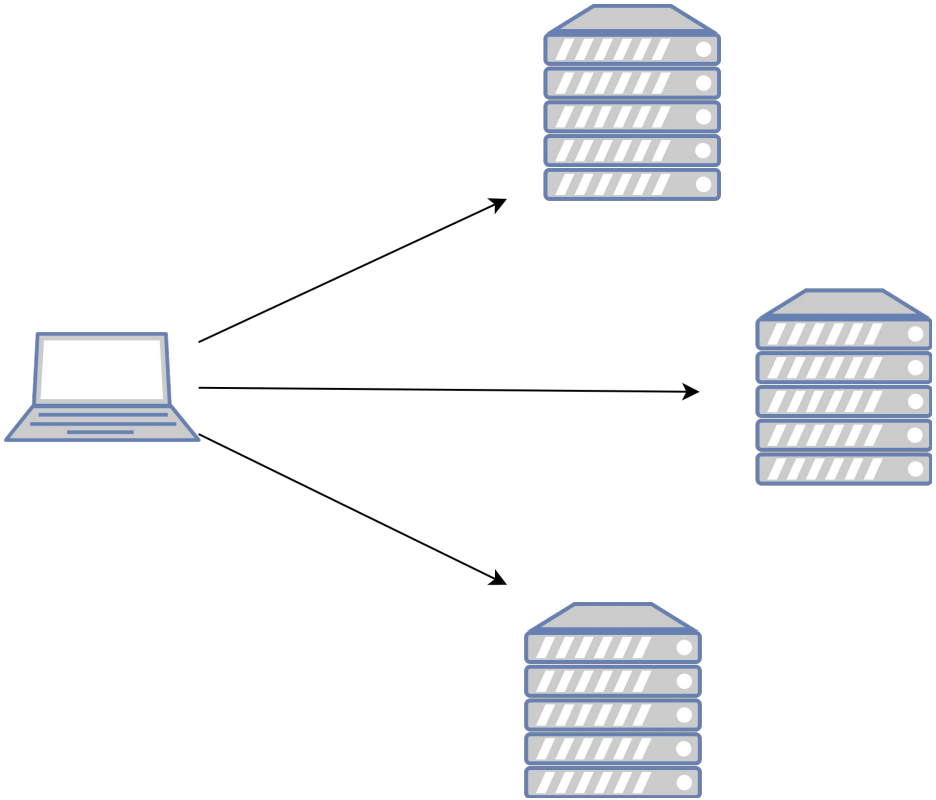
А если у нас много машин



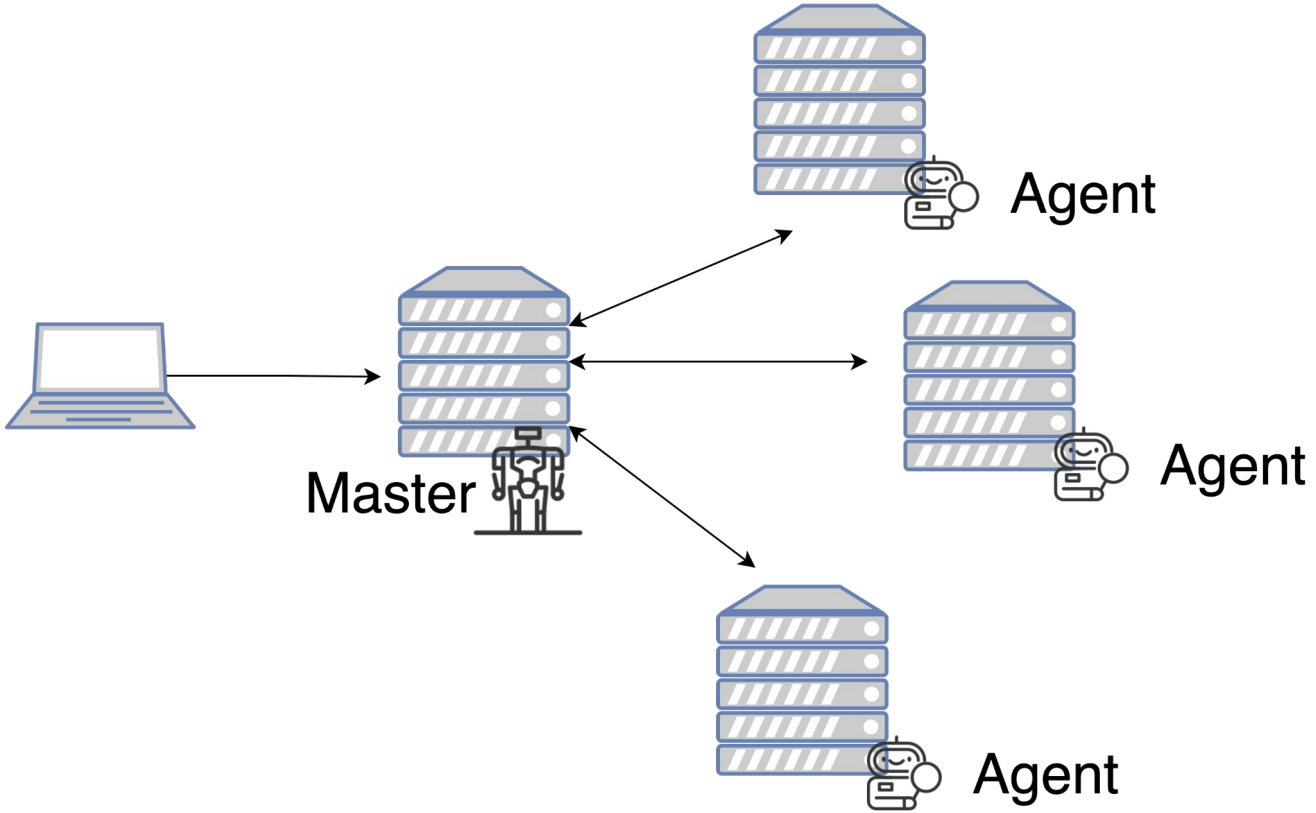
А если у нас много машин

Как поддерживать одинаковое окружение на всех машинах?

Как не запутаться при ручном управлении?



Централизированное управление конфигурацией



Примеры

Конфигурирование напрямую

- Bash + SSH
- Ansible

Конфигурирование через агентов

- Puppet
- Chef
- Salt

Воспроизводимость

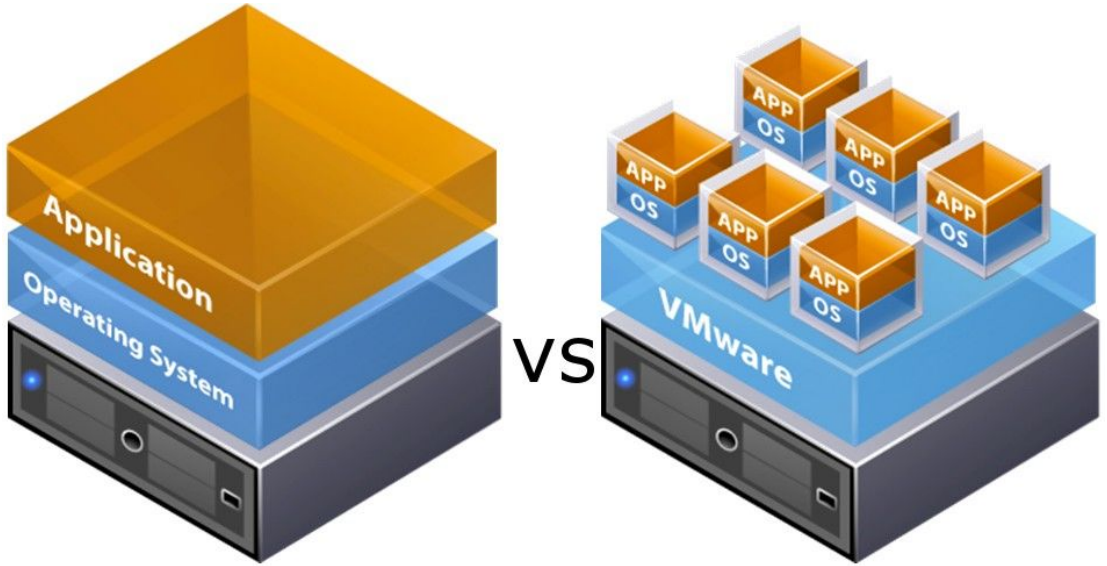
“ну не знаю, у меня такая же нога, а ничего не болит” (с)

Локально у вас на ноутбуке все запустилось, а в продакшене падает

Почему? И что делать?

Виртуальные машины

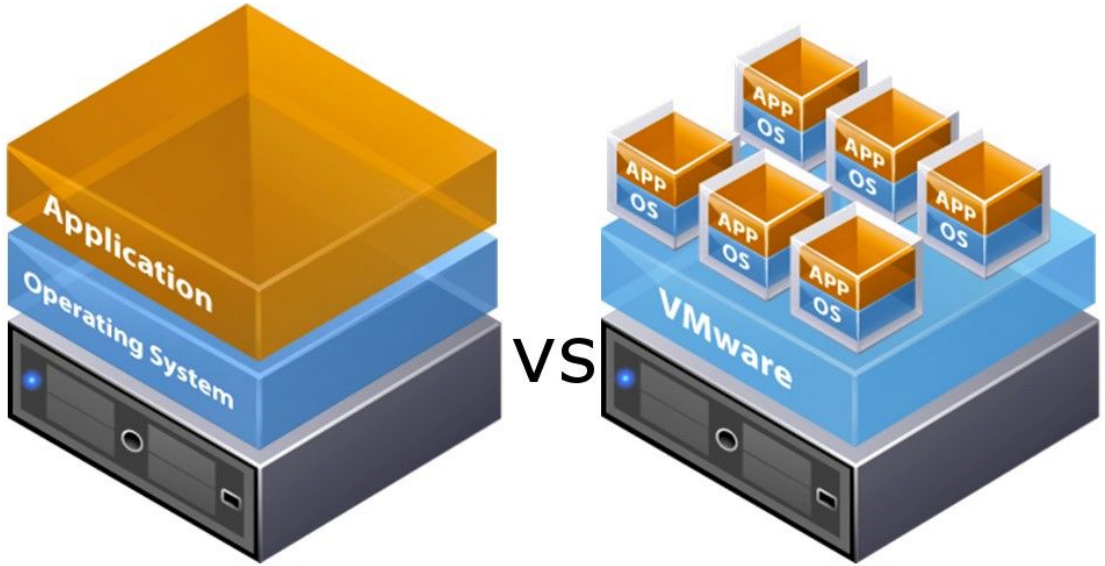
Виртуализация решает проблему окружения для запуска



Виртуальные машины

Виртуализация решает проблему окружения для запуска

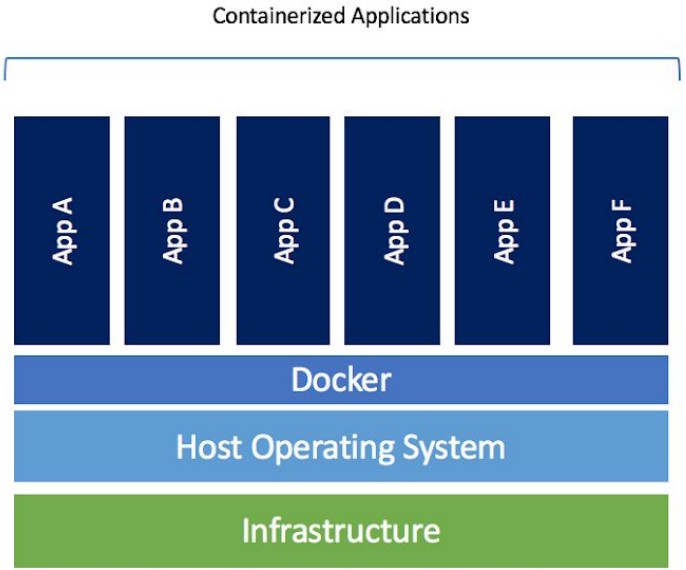
Но при этом влияет на производительность

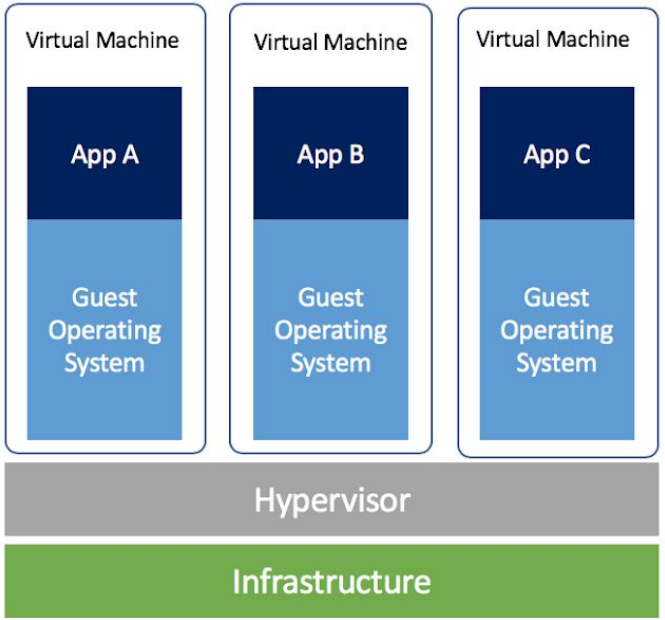


Traditional Architecture

Virtual Architecture

Контейнеризация





Docker