



Dev для Ops

Павел Селиванов
инженер Southbridge



План

- Сервисы
- Инфраструктура и код
- Тестирование
- docker-compose



VS

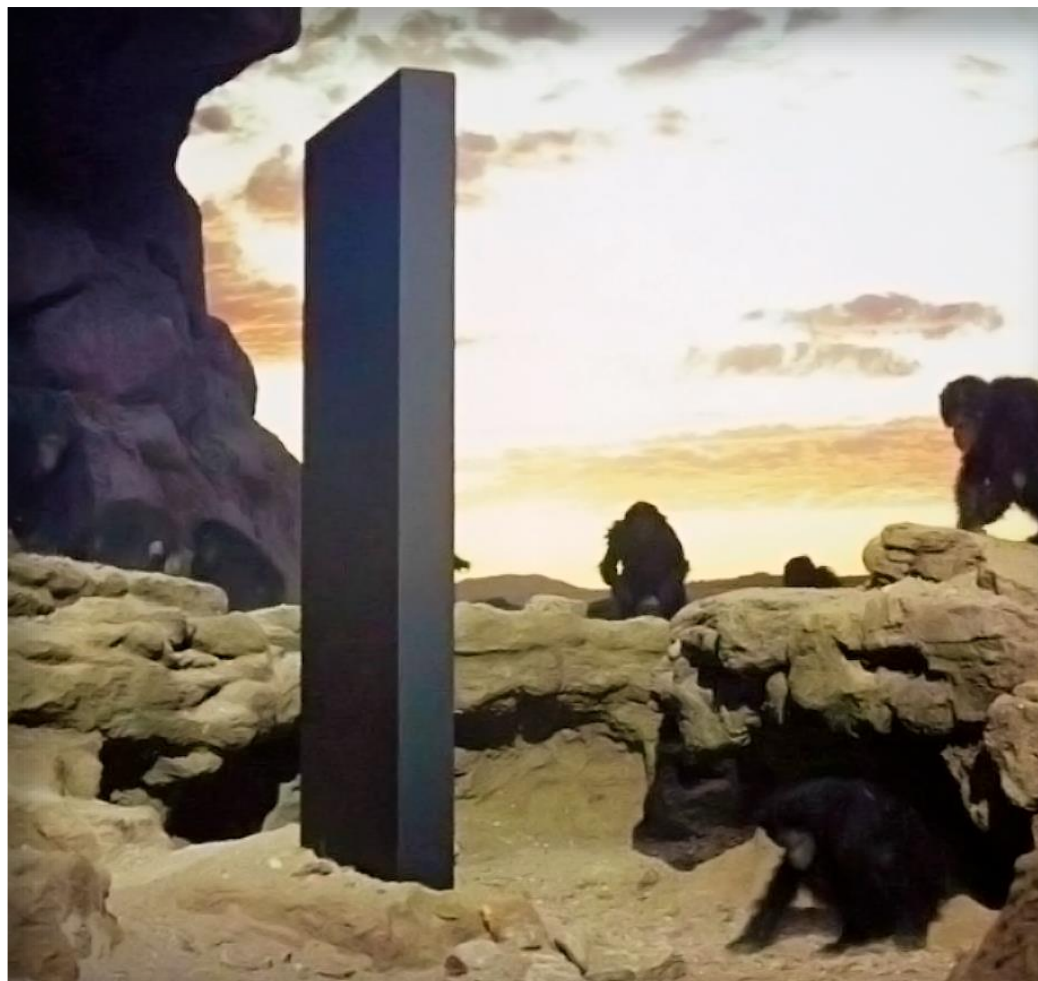


Монолиты

Микросервисы

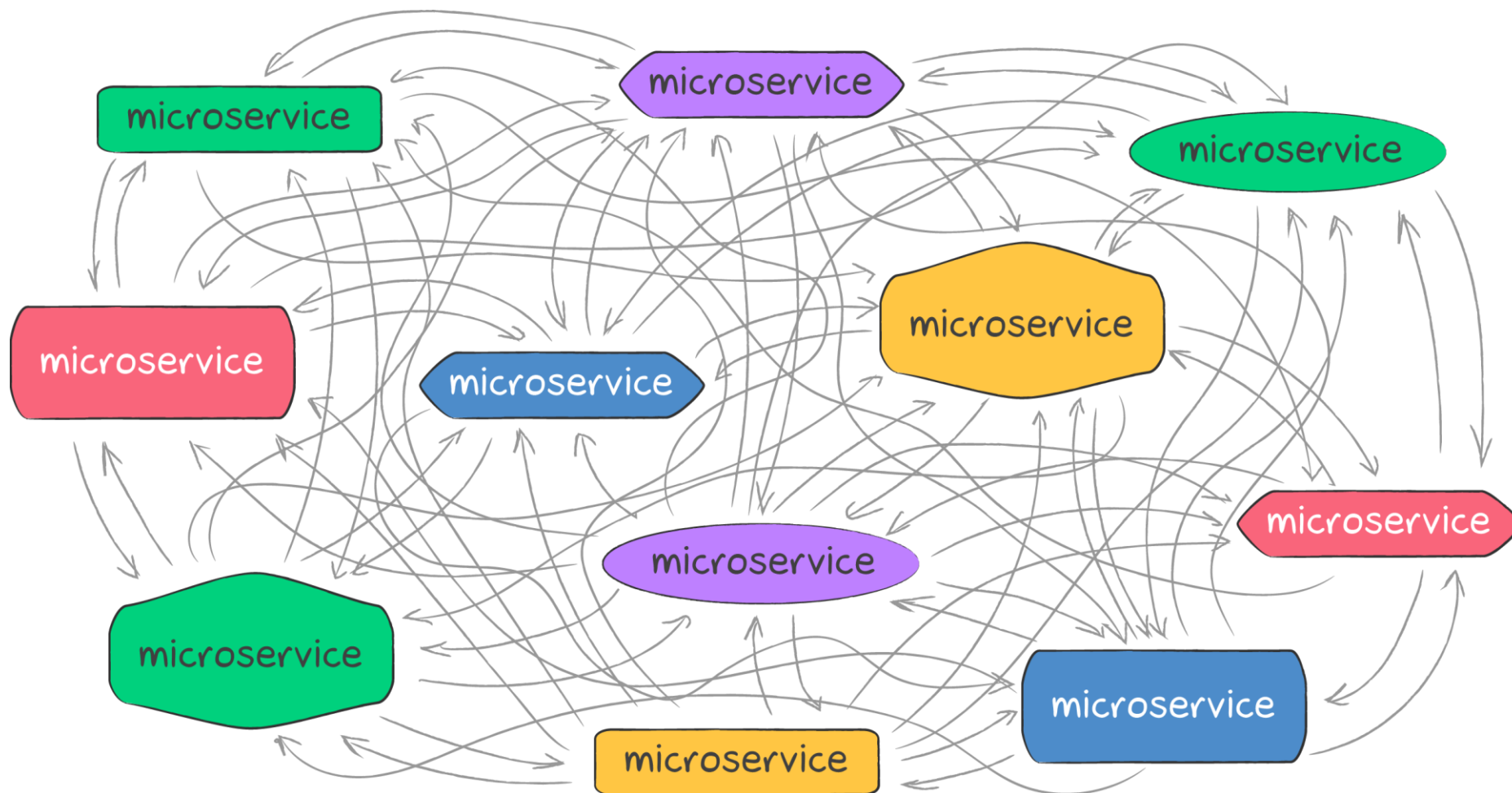


Монолиты



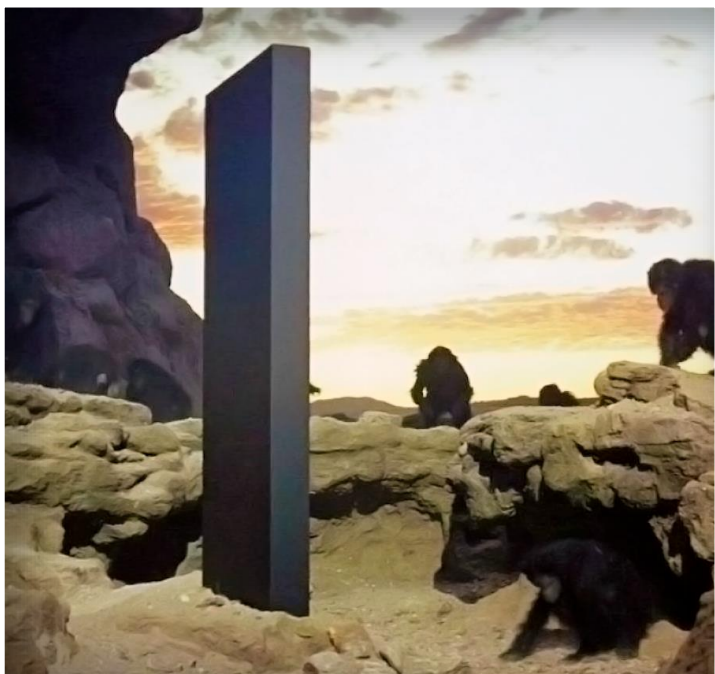


Микросервисы

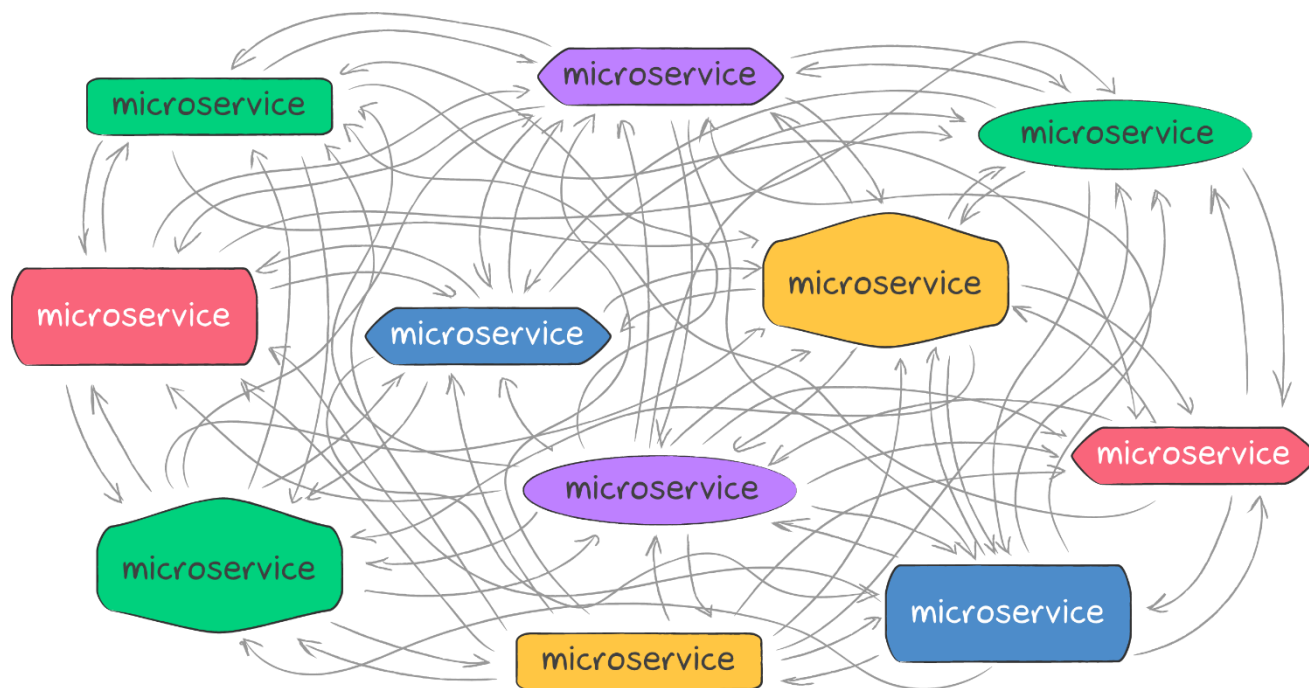




Макросервисы



+





Костыли





Конфигурация



В коде



Конфигурация



В коде



Переменные окружения



Конфигурация



В коде



Переменные окружения



Конфиги



Конфигурация

- В коде
- Переменные окружения
- Конфиги
- Свой формат



Конфигурация

- В коде
- Переменные окружения
- Конфиги
 - Свой формат
 - YAML / TOML / JSON / XML

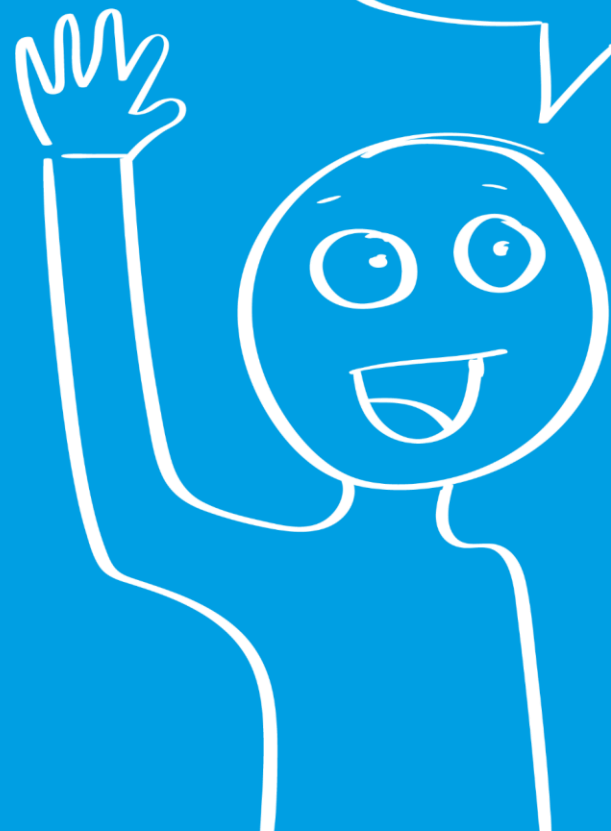


Конфигурация

- В коде
- Переменные окружения
- Конфиги
- Свой формат
- YAML / TOML / JSON / XML
- Конфиг сервера + автодискавери

Тестирование

Зачем?





Зачем

- Экономит время и ресурсы;
- Обеспечивает безопасность кода при командной работе;
- Улучшает качество кода.



docker-compose

docker-compose.yml

network

App

DB

Redis



docker-compose

docker-compose.yml

network

App

Redis

Поднятие любого проекта

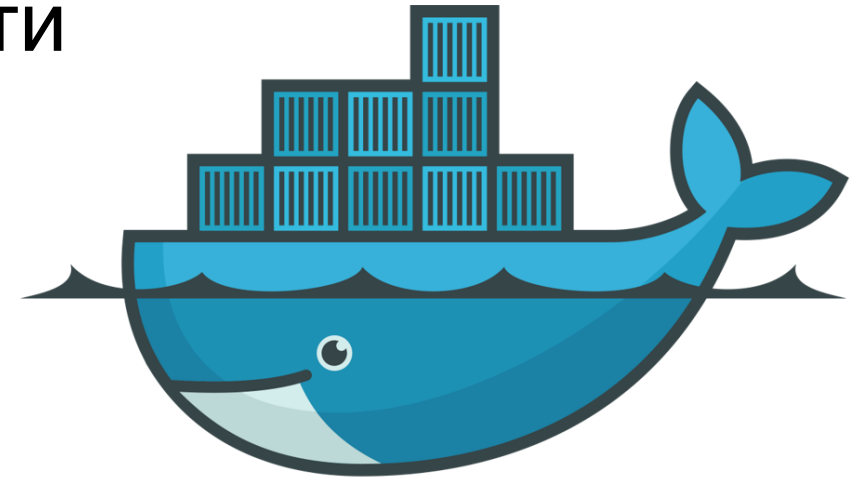
==

`docker-compose up -d`



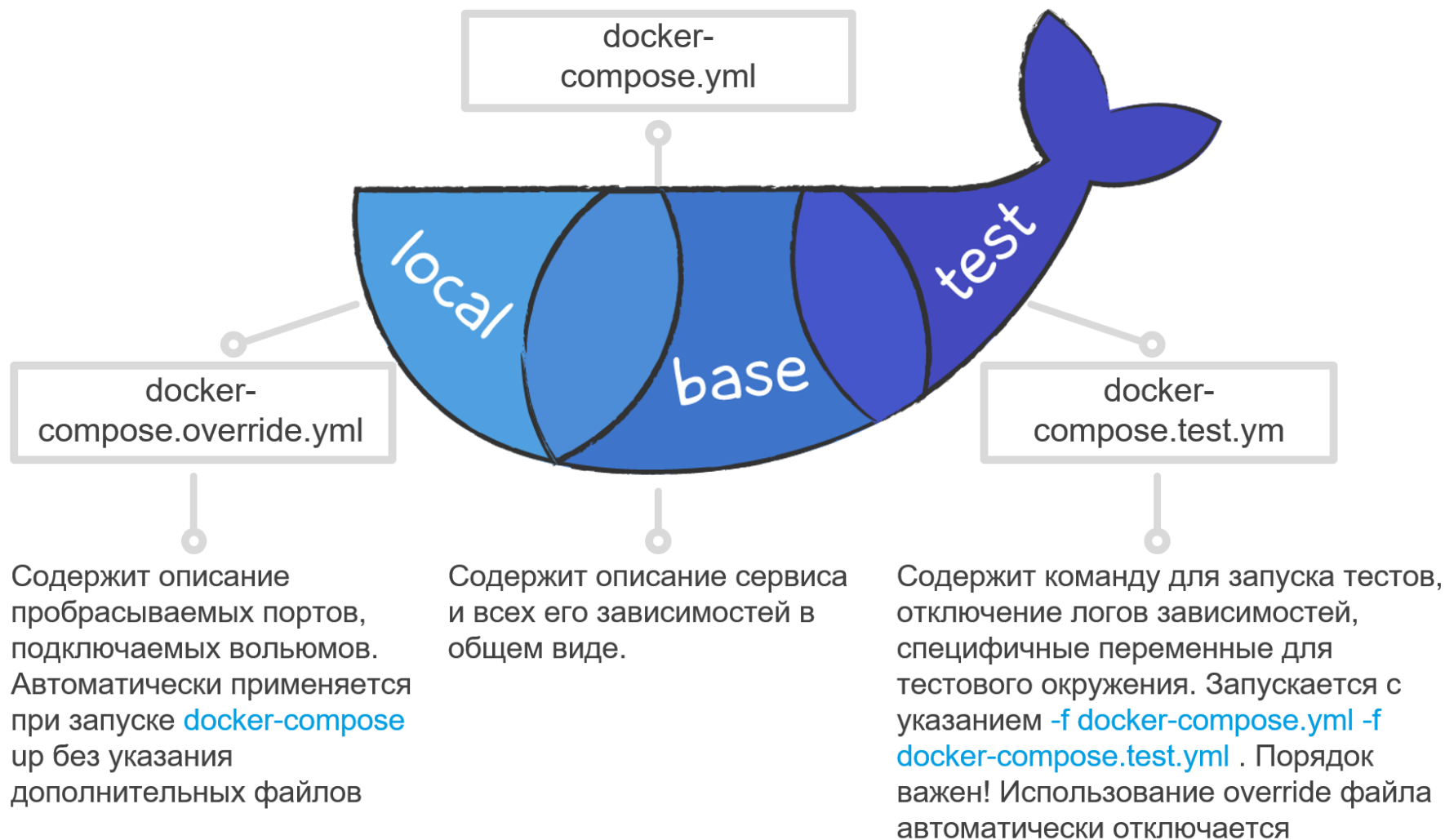
Разработка с Docker

- 1. Стандартный **Dockerfile**
- 2. Стандартный **docker-compose.yml**
- 3. Добавляем нужные зависимости
- 4. Собираем **образ**
- 5. Запускаем окружение
- 6. Пишем приложение
- 7. Повторяем с **п.3** до достижения успеха





Тестирование с Docker





slurm.io