

Dev для Ops

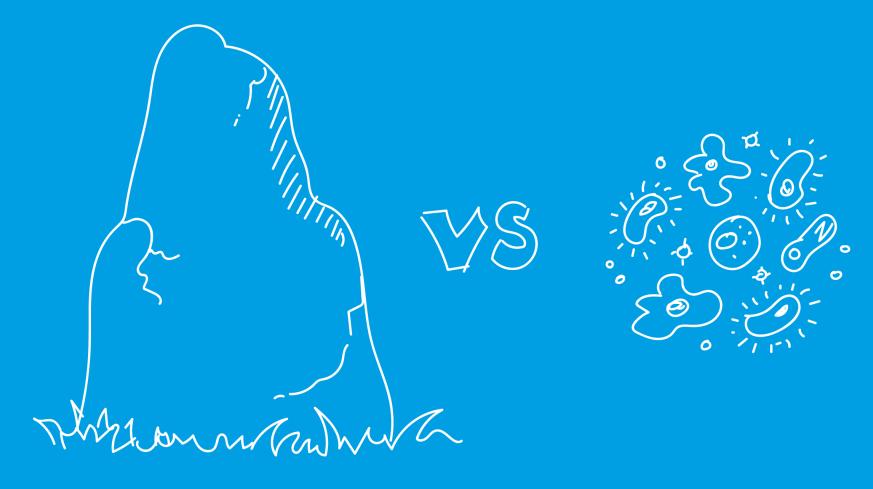
Павел Селиванов инженер Southbridge



済 План

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





Монолиты

Микросервисы





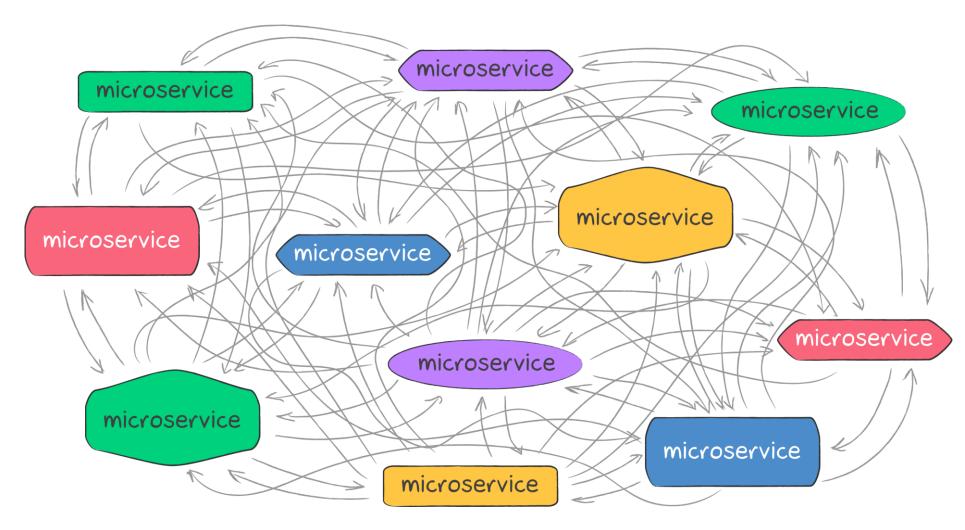
Монолиты







Микросервисы

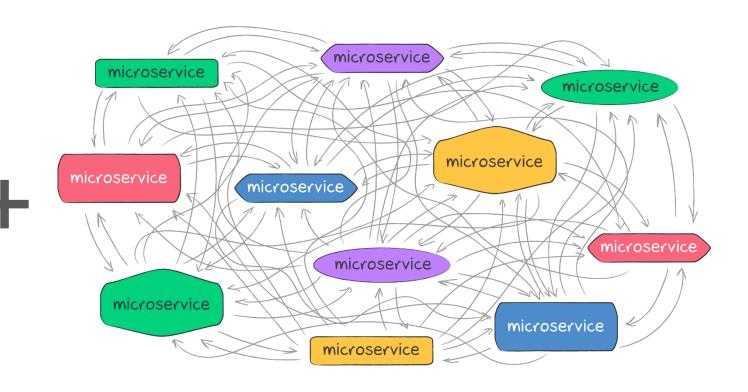






Макросервисы















В коде





В коде

Переменные окружения





- В коде
 - Переменные окружения
- Конфиги





- В коде
- Переменные окружения
- **—** Конфиги
- --- Свой формат





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



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



Тестирование



Зачем?

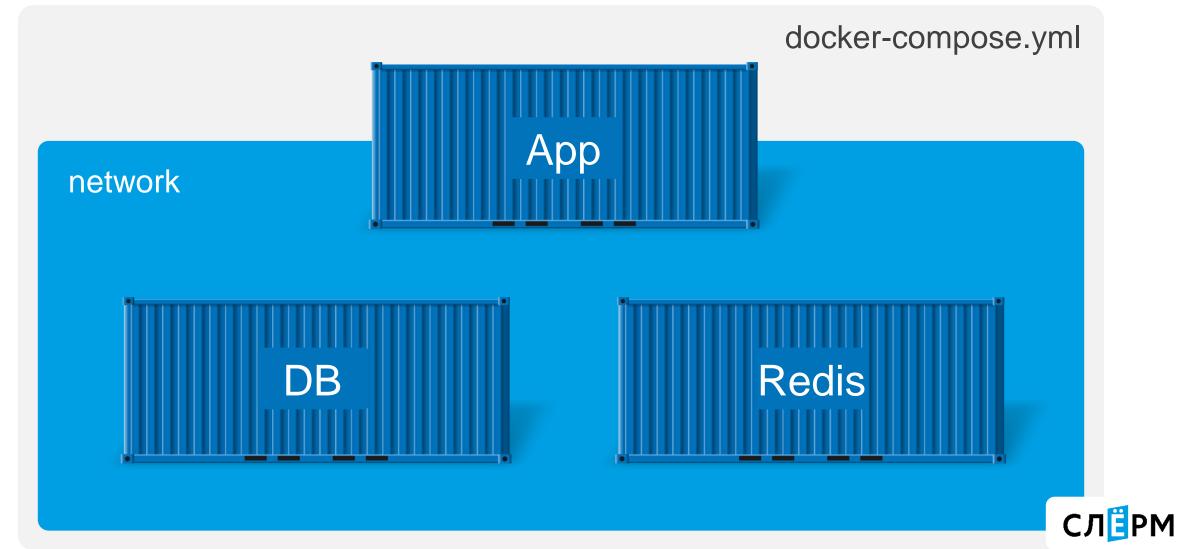


Зачем

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

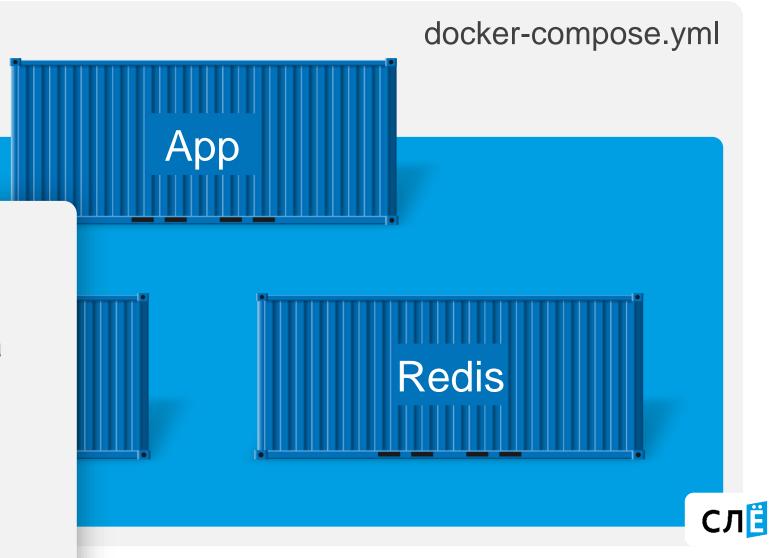


docker-compose





docker-compose



Поднятие любого проекта

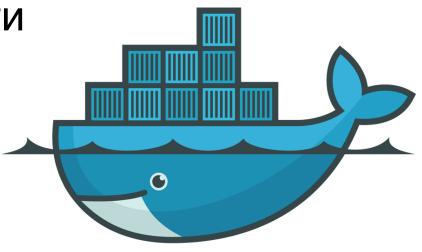
network

docker-compose up -d



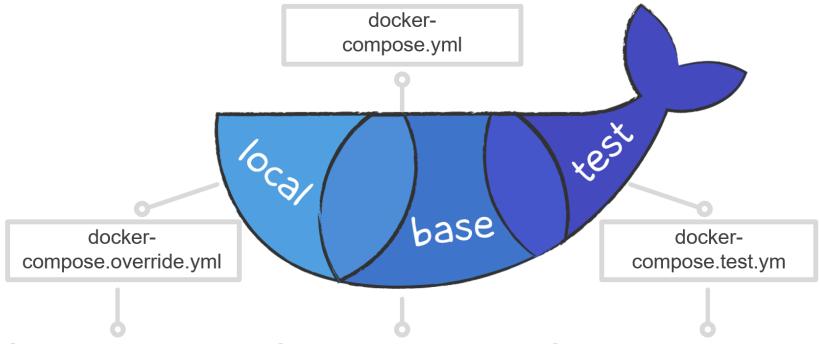
Разработка с Docker

- 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



Содержит описание пробрасываемых портов, подключаемых вольюмов. Автоматически применяется при запуске docker-compose ир без указания дополнительных файлов

Содержит описание сервиса и всех его зависимостей в общем виде.

Содержит команду для запуска тестов, отключение логов зависимостей, специфичные переменные для тестового окружения. Запускается с указанием -f docker-compose.yml -f docker-compose.test.yml . Порядок важен! Использование override файла автоматически отключается



