

AUTOMATE IT

Кирилл Тихонов

DINS

Disclaimer

Все это используется в WEB

Под Linux

Примеры будут показывать один
инструмент

Задача доклада - сподвигнуть



Исходные данные

Есть сервер и клиент

Сервер посылает сообщение клиенту

Ответ

Если все хорошо: 200 ok

Если нет: 500 internal error

Проблемы?

НЕТ

Проблемы?

- Пришел новый человек и ему нужно объяснить как собирается проект
- Надо обновить библиотеку т.к. старая версия небезопасна
- Билд сервер упал со скалы
- За билд сервером со скалы прыгнул единственный человек который знает как это собирать для продакшена

Проблемы?

Перефразируем:

Окружение и зависимости это состояние, а не код

Package manager

Наиболее распространенные:

- Conan
- vcpkg
- nuget

Package manager conan find cmake api



Package manager Conanfile.txt

```
[requires]  
boost/1.69.0@conan/stable
```

```
[generators]  
cmake_find_package
```

Проблемы? Решены только две

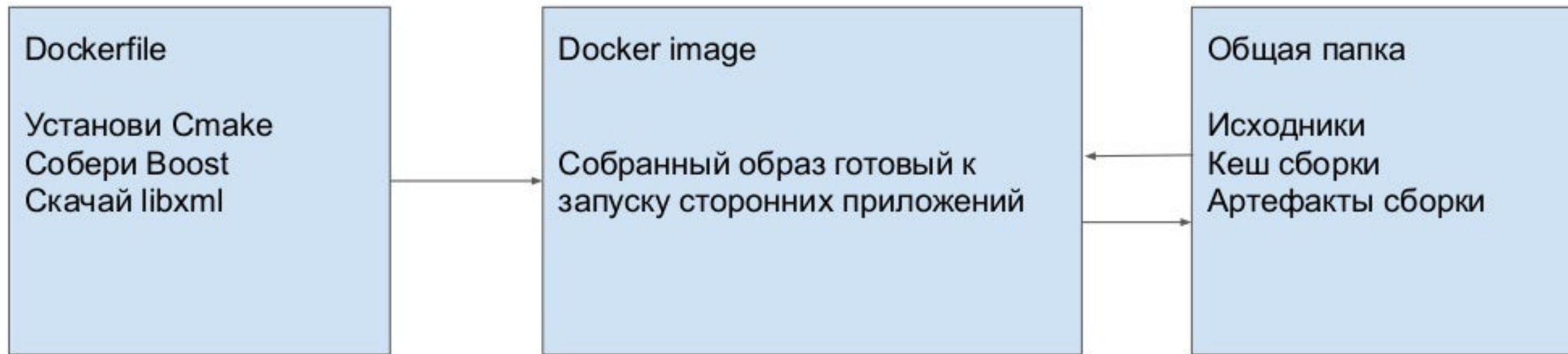
- Пришел новый человек и ему нужно объяснить как собирается проект
- Надо обновить библиотеку т.к. старая версия небезопасна
- Билд сервер упал со скалы
- За билд сервером со скалы прыгнул единственный человек который знает как это собирать для продакшена
- Версия GCC, Conan, Cmake

Docker

Что такое докер?

- Это не виртуальная машина.
- Это песочница
- способ описывать окружение в виде файлов
- возможность иметь везде одинаковое окружение
- способ изолировать ваше приложение от неявных зависимостей

Docker



Docker

Dockerfile для сборки

```
FROM centos:7.6.1810
```

```
COPY yum_packages.txt /tmp/yum_packages.txt
```

```
RUN xargs -a /tmp/yum_packages.txt yum install -y
```

```
ENV BOOST_ROOT=/home/devel/boost_1_60_0
```

```
USER=devel
```

Docker

Для работы с контейнером нужно 2 простых команды

```
docker build -t pcf_docker ./docker
```

```
docker run --rm -v $(pwd):/home/devel/build_dir -v /tmp:/tmp pcf_docker  
./buildscript.sh
```


Docker

Но иногда приходится писать так

```
docker run --privileged --dns=1.1.1.1 --rm -v $(pwd):/home/devel/build_dir -v /tmp:/tmp pcf_docker bash -c "snmpd; rsyslogd; ./docker/copy_mibs.sh; cd /home/devel/build_dir/pcf; (su devel -c './buildscript.sh $command;')"
```


Docker

Главные трудности которые мы получили:

- Дебаг
- Подсветка синтаксиса

Дебаг

Можно:

Присоединяться к запущенному контейнеру и выполнять параллельно gdb
через docker exec

Запускать GDB в контейнере и открывать порт

Главное дать нужные параметры при запуске контейнера
`--cap-add=SYS_PTRACE --security-opt seccomp=unconfined`

Подсветка

CLion: удаленное подключение к контейнеру по ssh, закачивание исходников, скачивание хедеров для быстрого локального резолва

VSCode: запуск серверной части внутри контейнера

Docker

Пример из VSCode: devcontainer.json

```
{  
  "name": "kam",  
  "dockerFile": "../docker/dockerfile",  
  
  "runArgs":  
    ["-u", "1000", "--cap-add=SYS_PTRACE", "--security-opt", "seccomp=unconfined"],  
  
  "extensions": ["ms-vscode.cpptools"]  
}
```

Проблемы?

НЕТ

Полностью изолированное детерминированное окружение для сборки билда как на машине разработчика, так и на билдсервере, билдсервер и есть докер контейнер собранный из рецепта который находится рядом с исходниками

Проблемы?

Помимо сборки девелоперы запускают тесты

Добавим в контейнер:

Юнит тесты при сборке

Скрипт с несколькими приложениями внутри одного контейнера

Проблемы?

Программисты ошибаются

Не запускают все тесты

Считают что их изменение точно ничего не ломает. Я ломал

Пайплайн Тесты




















Решение

Добавляем пайплайн в jenkins/Gitlab

Внутри него будет:

- Статический анализ
- Сборка
- Прогон тестов
- Деплой (иногда)

Как это выглядит

Status	Pipeline	Triggerer	Commit	Stages		Run Pipeline
	#149043 latest detached		 abf3085f  UP-13326 ./	 	 00:12:34  1 month ago	
	#149041 detached		 4988ca2d  UP-13326 chmod ux	 	 00:00:02  1 month ago	

Как это выглядит

build:

stage: build

script: - ./buildscript.sh docker make

test:

stage: test

script: - bash ./buildscript.sh docker run_tests

only: - merge_requests

Второе решение - Гит хуки

В них обычно проверяют:

- Валидность коммита (jira task)
- Форматирование

Git Hook
clang-format

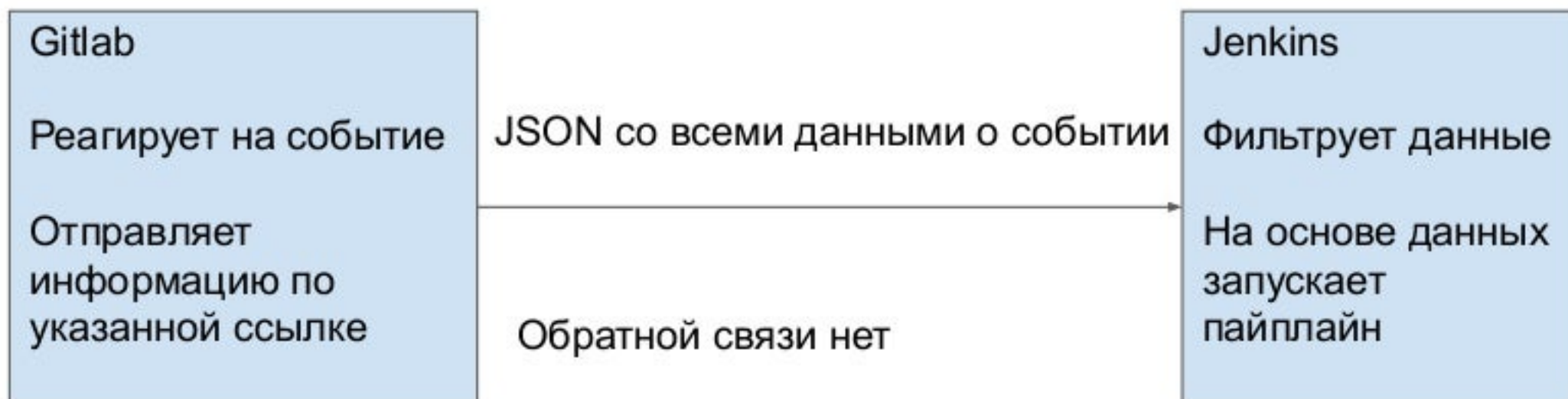


Евенты

В них обычно используют:

- Статический анализ
- Форматирование
- Деплой
- Прогон тестов если это занимает много времени

Евенты



Проблема?

Не все ошибки находятся на предыдущем этапе

Решение

Избыточность



Решение

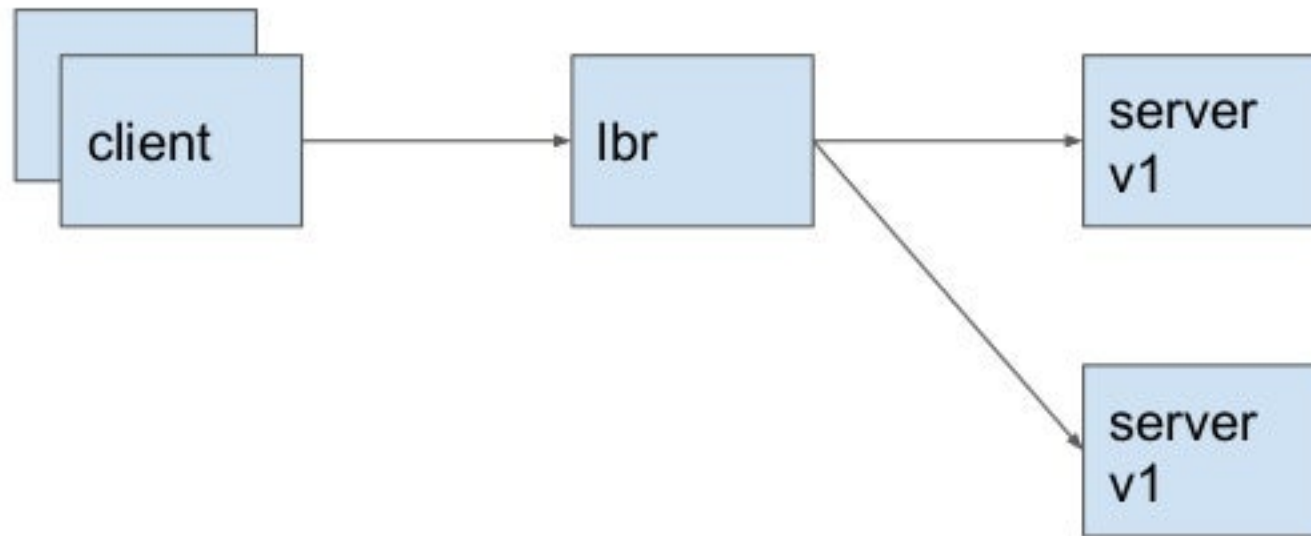
Клиента теперь два.

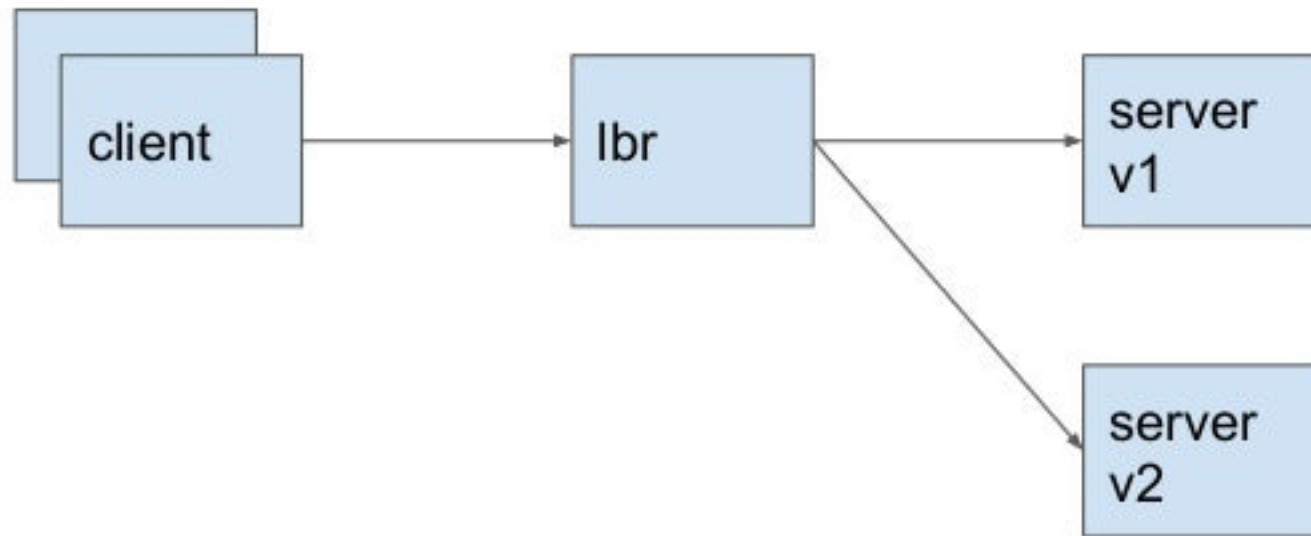
Если клиент получил невалидный ответ, он пробует еще раз

Два сервера, распределение нагрузки между ними через лoad балансер

Бонус







Проблемы?

Теперь ошибки не влияют на систему, но мы слишком поздно понимаем что все плохо

Решение?

Начнем собирать метрики

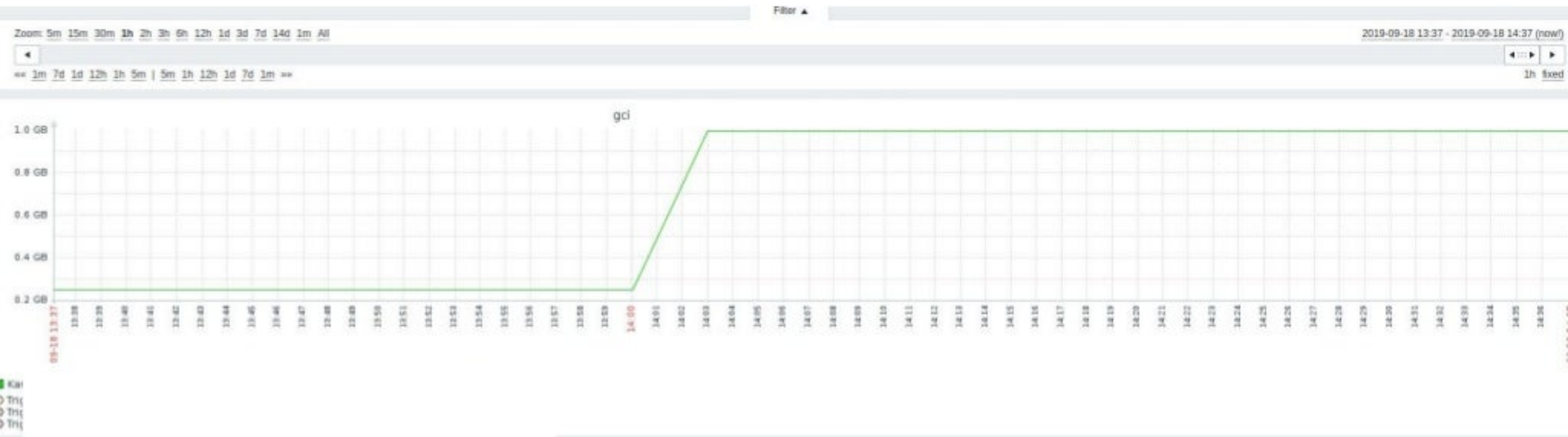
Метрики

Собираем с пода/конкретного сервера через:
Http, snmp

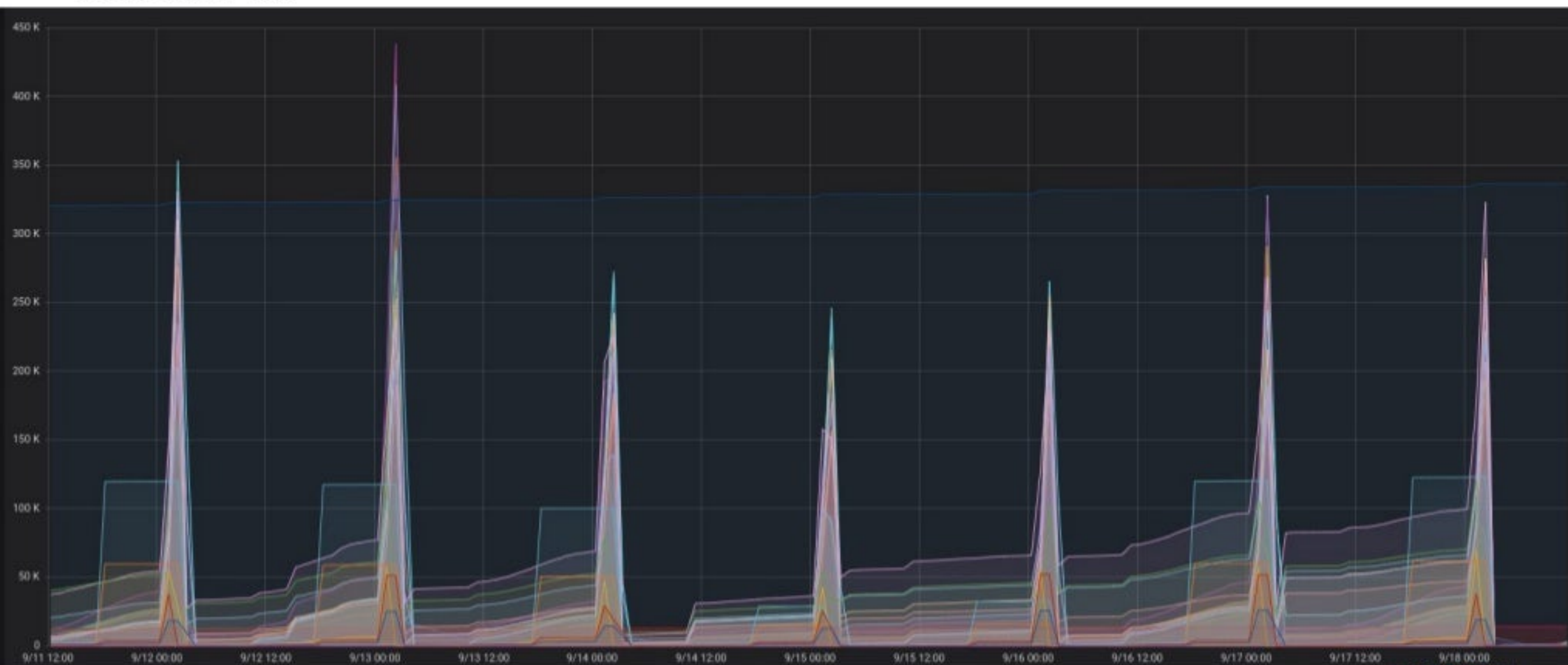
Рисуем красивые графики:
Zabbix, grafana

Настраиваем триггеры

Zabbix



Grafana



Метрики

Пишем письмо человеку который сломал билд
Последнему автору мерж реквеста

Дублируем письмо команде



Проблемы?

Люди которые это выкатывают, тоже ошибаются

Решение

Docker контейнер для приложения

Docker контейнер для приложения



Docker контейнер для приложения

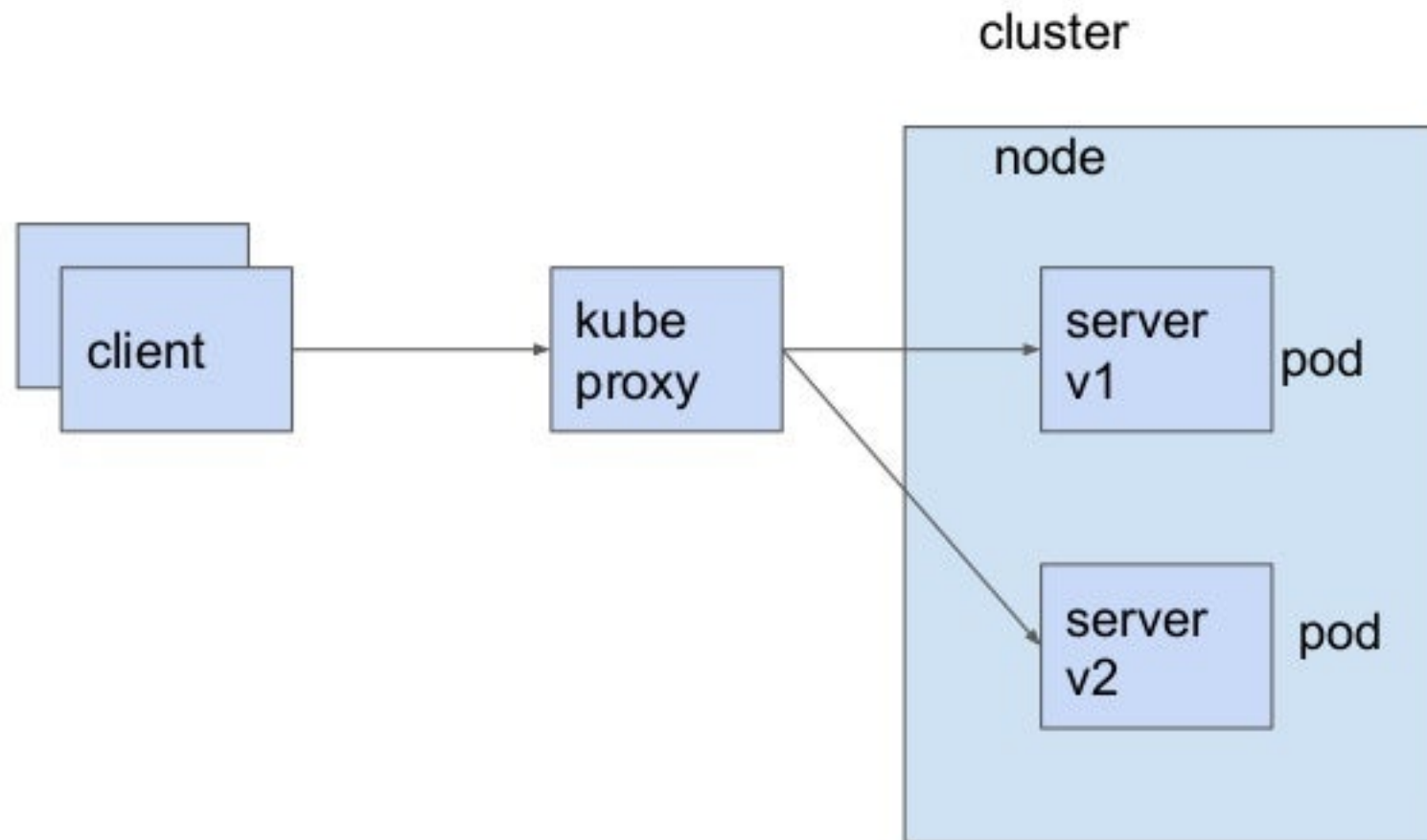
Отличия от контейнера для сборки:

- Минимальный размер
весь tmp остается за бортом контейнера
- Отсутствие лишнего
нет не используемых инструментов
- Простота использования
снаружи передается только несколько конфигурационных параметров

Disclaimer

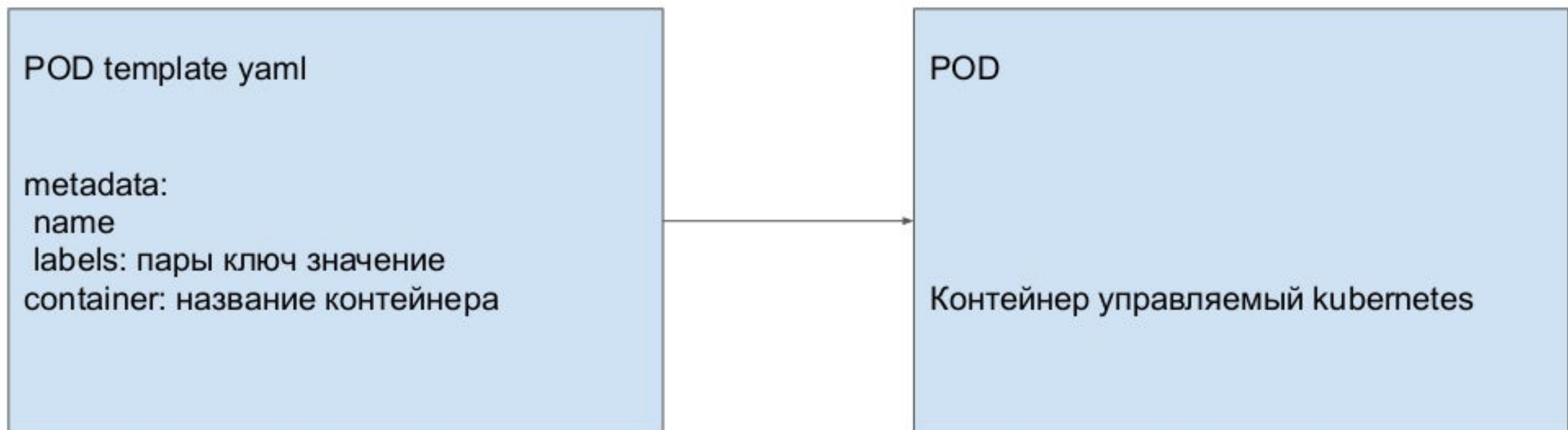
Kubernetes не использовался мной в продакшене

Kubernetes



Kubernetes

```
kubectl create -f podtemplate.yml
```



А как катить?

Контроллеры



Kubernetes

```
kubectl create -f deployment.yml
```

Controller (deployment) yaml

selector
POD template

Количество



Kubernetes

```
kubectl create -f deployment.yml
```

Controller (deployment) yaml

selector
POD template

Количество



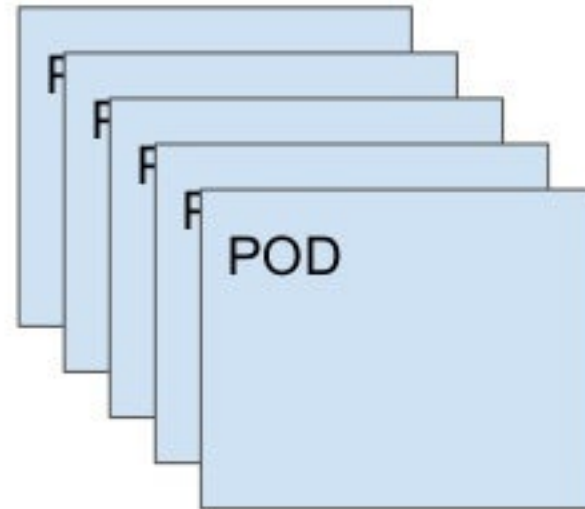
Kubernetes

```
kubectl apply -f thisdeployment.yaml
```

Controller (deployment) yaml

selector
POD template **НОВЫЙ**

Количество



Kubernetes

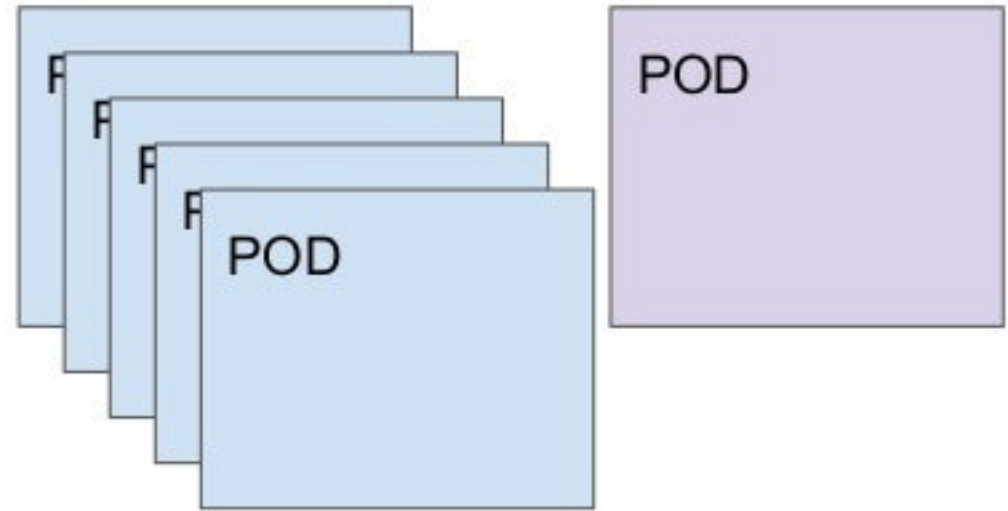
```
kubectl apply -f thisdeployment.yaml
```

Controller (deployment) yaml

selector

POD template **НОВЫЙ**

Количество



Kubernetes

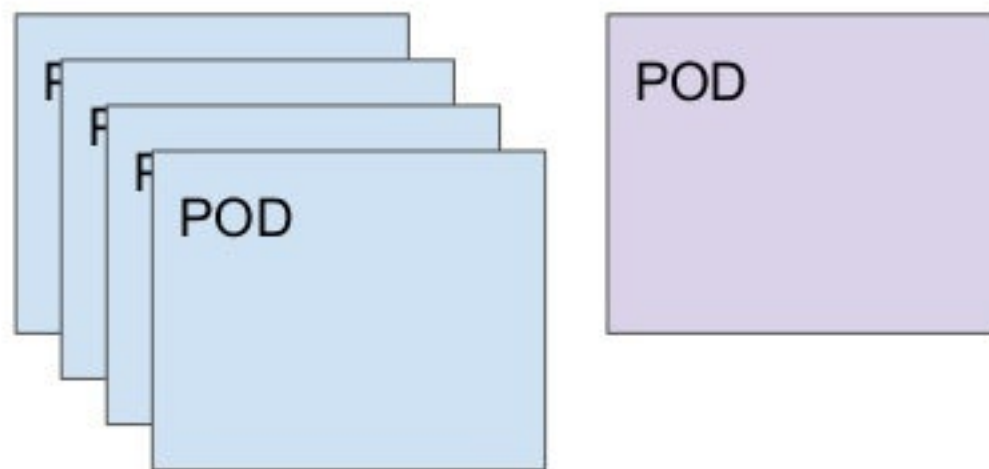
```
kubectl apply -f thisdeployment.yaml
```

Controller (deployment) yaml

selector

POD template **НОВЫЙ**

Количество



Kubernetes

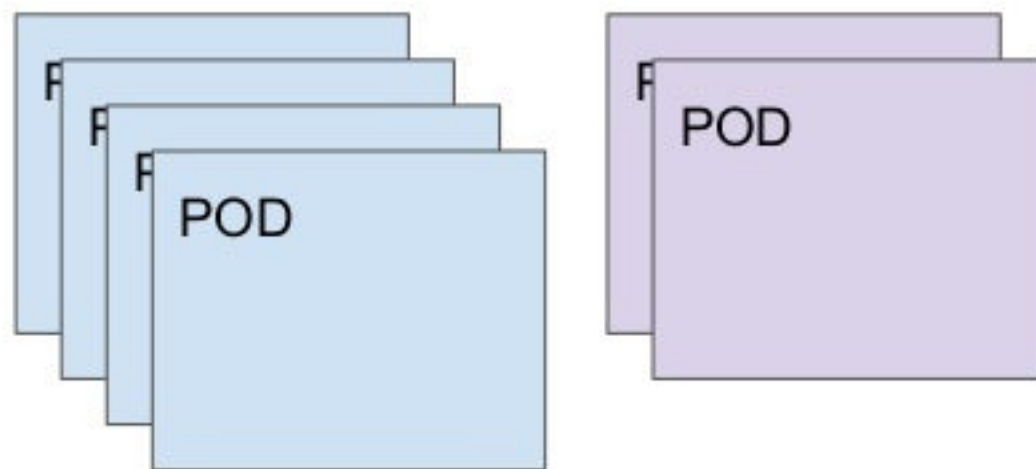
```
kubectl apply -f thisdeployment.yaml
```

Controller (deployment) yaml

selector

POD template **НОВЫЙ**

Количество



Kubernetes

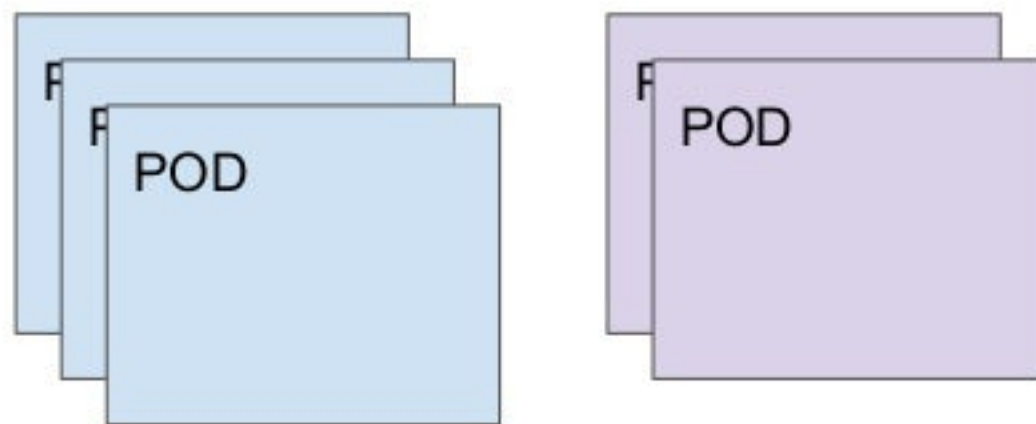
```
kubectl apply -f thisdeployment.yaml
```

Controller (deployment) yaml

selector

POD template **НОВЫЙ**

Количество



Kubernetes

```
kubectl apply -f thisdeployment.yaml
```

Controller (deployment) yaml

selector

POD template **НОВЫЙ**

Количество



Проблемы?

С таким подходом система стала сверхпопулярной, и в 2 часа ночи все начали ей пользоваться

Решение

Kubernetes поддерживает автомасштабирование

По метрикам

Решение

PODAutoscaler yaml

targerref: deployment

max

min

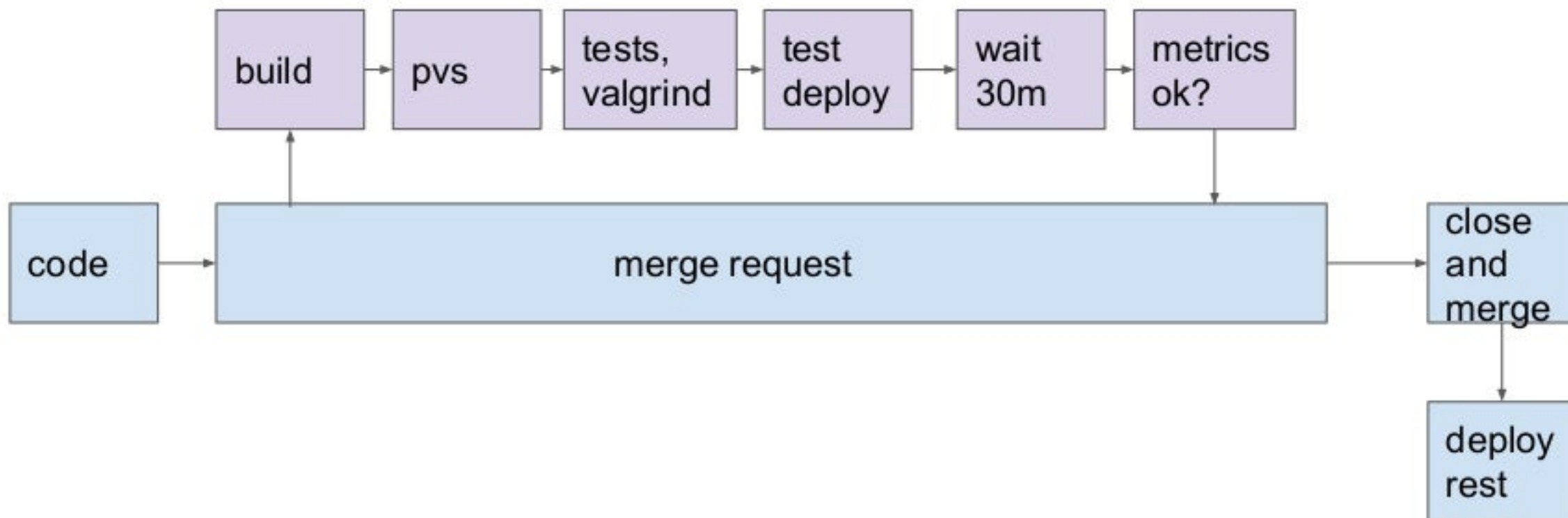
metrics

Проблемы?

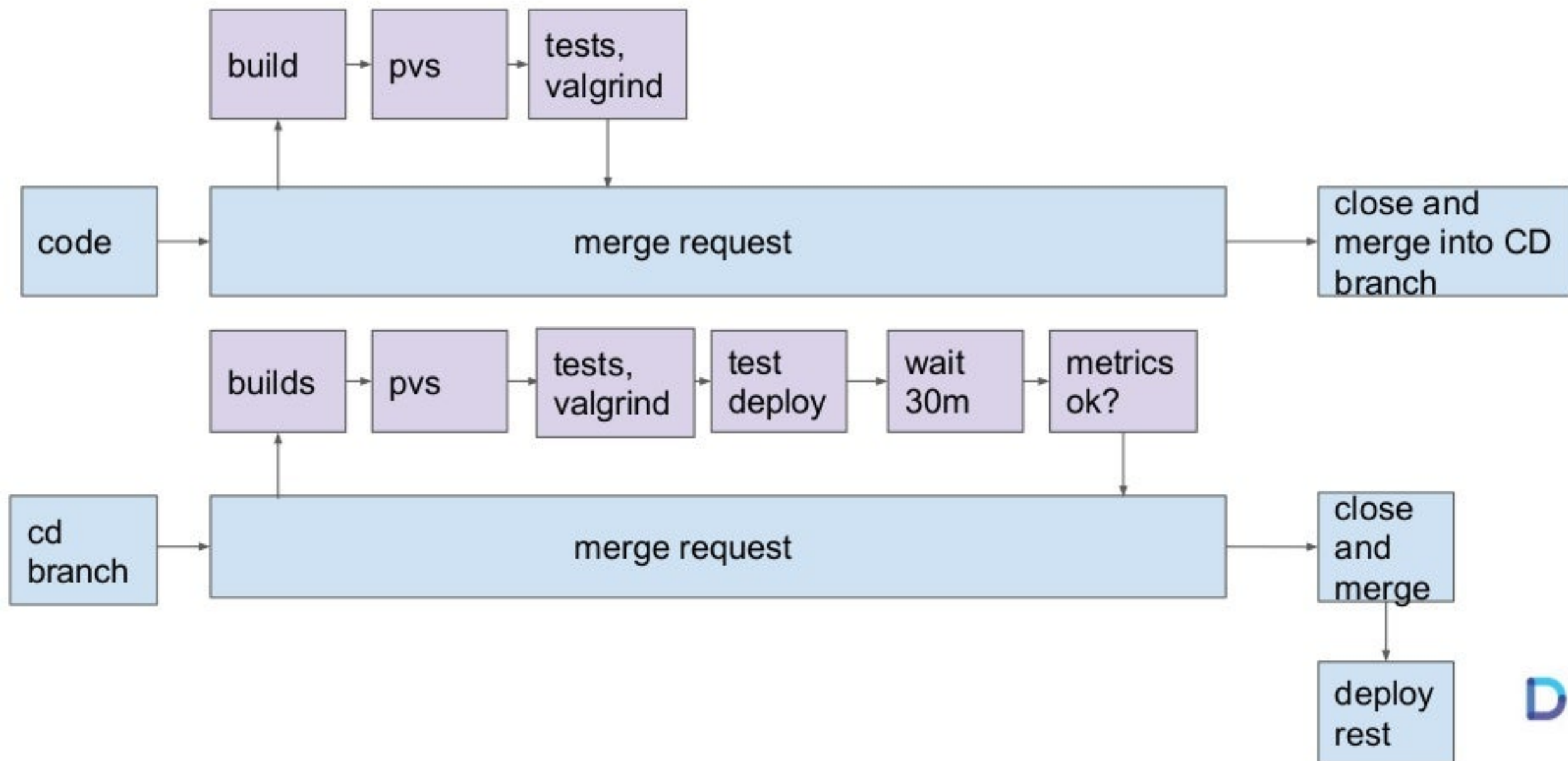
Кончились, придумаем сами



Объединим



Объединим



Проблемы?

А зачем это все?

Вася

Вася ошибся в коде, за него это нашел статический анализатор и отправил ему письмо. Почини вот тут.

Прошел месяц, он быстро фиксил баг и не запустил все тесты, а тесты сломались... все

Через месяц он изменил систему конфигурации приложения и опечатался. Проблема локализована во время замены одного тестового сервера и ни один клиент не был затронут.

Потом всю команду уволили, ведь они теперь не обладают тайными знаниями как все должно работать, знания документированы рядом с кодом.

Спасибо