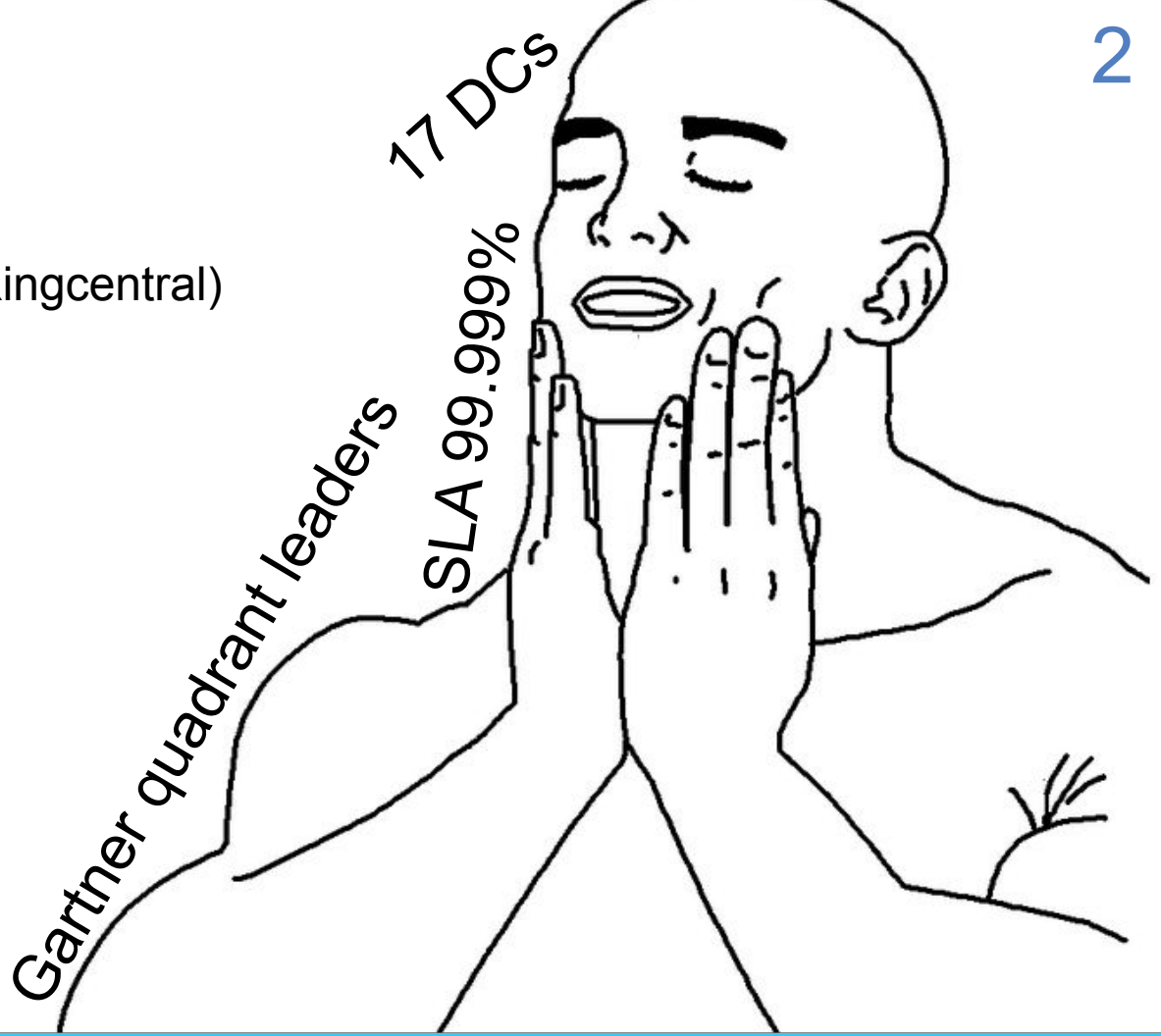
Автоматизация сборки, тестов и выкатки

Кирилл Тихонов

Team lead @ DINS (R&D of Ringcentral)

Exp:

Embedded 7y +
Highload 2y +



Agenda

Билд:
Детерминированный билд
Проверка на этапе сборки

Система:
Моментальная обратная связь
Контроль выкатки
Проверка на этапе выкатки

Исходные данные

Есть сервер и клиент

Клиент посылает сообщение серверу

Ответ
Если все хорошо: 200 ok
Если нет: 500 internal error





Проблемы?

Нам надо это собирать






Пайплайн - программный конвейер

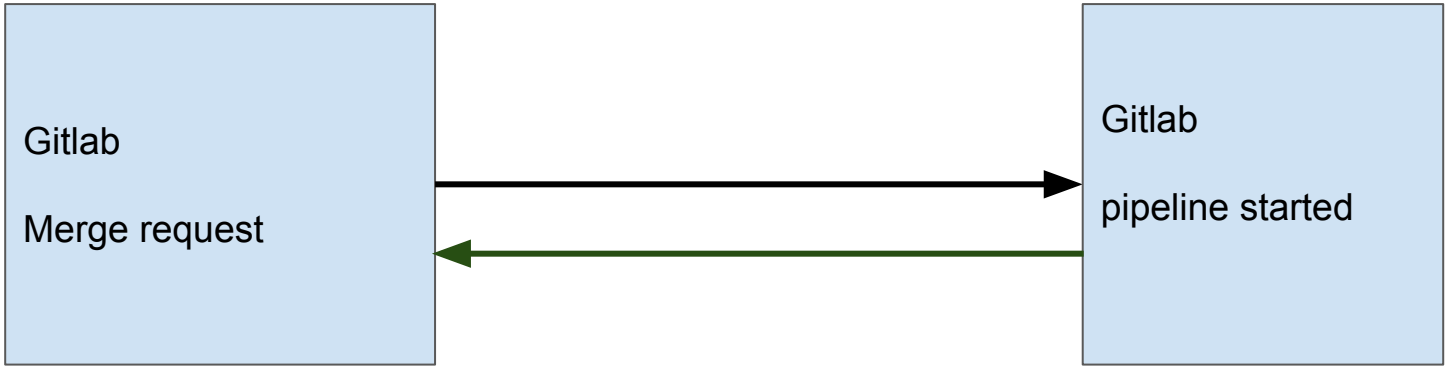
Примеры автоматизаций

- script
- client git-hook
- gitlab
- travis
- jenkins
- spinnaker (netflix)

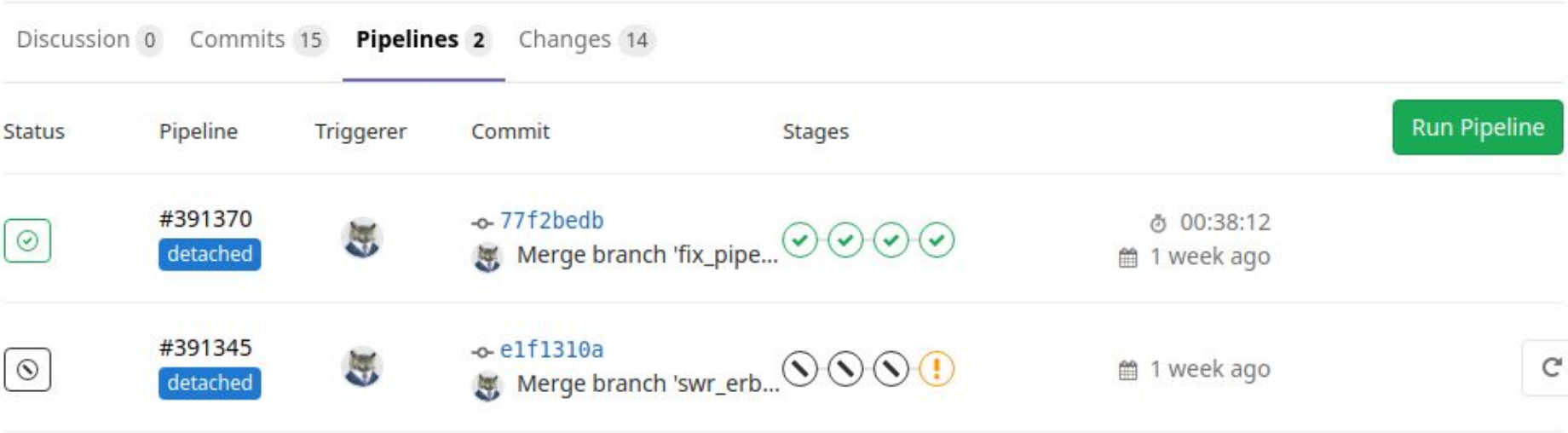




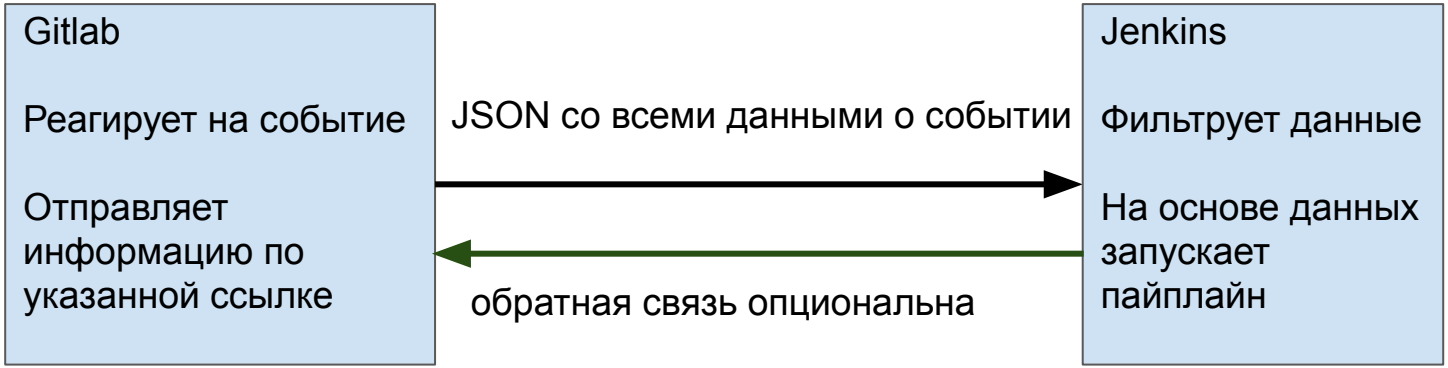
Встроенный пайплайн



Как это выглядит



Евенты


Gitlab
Реагирует на событие
Отправляет информацию по указанной ссылке
JSON со всеми данными о событии
обратная связь опциональна
Jenkins
Фильтрует данные
На основе данных запускает пайплайн

Гит хуки

Форматирование как пре commit hook

До:

```
SHA256_CTX sha256;
    SHA256_Init(&sha256); SHA256_Update(&sha256,
      str.c_str(), str.size());
  SHA256_Final(hash, &sha256);
```

После:

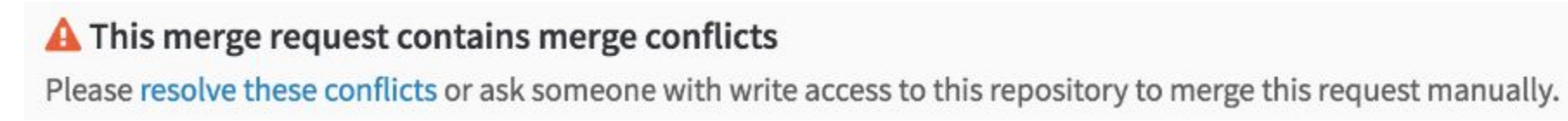
```
SHA256_CTX sha256;
SHA256_Init(&sha256);
SHA256_Update(&sha256,str.c_str(), str.size());
SHA256_Final(hash, &sha256);
```

Есть одно но






Гит хуки


This merge request contains merge conflicts
Please resolve these conflicts or ask someone with write access to this repository to merge this request manually.






DINS

Гит хуки

В них обычно проверяют:

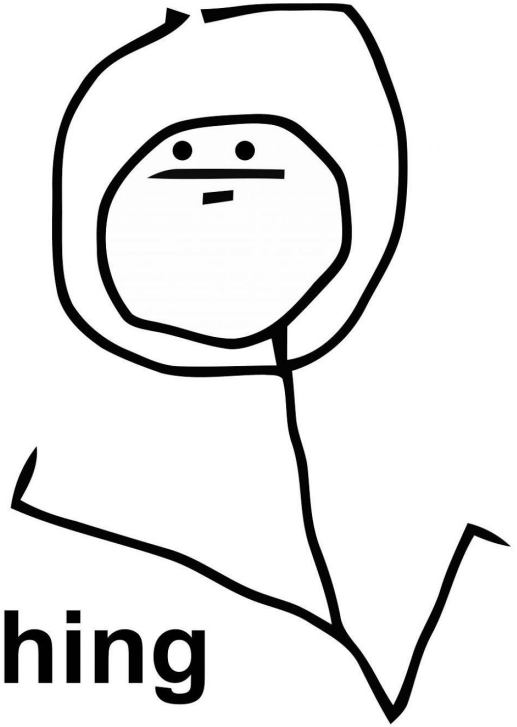
- Валидность коммита (jira task)
- Проверка статическая
- Проверка синтаксическая





Пайплайн?

build



Проблемы?

- Появилась бага, но воспроизводится только на билде из одного билдера

Два билдера, собирают по разному
Вы собираете на рабочей машине получаете третий результат
Девелопер получает четвертый





Решение

Управление зависимостями

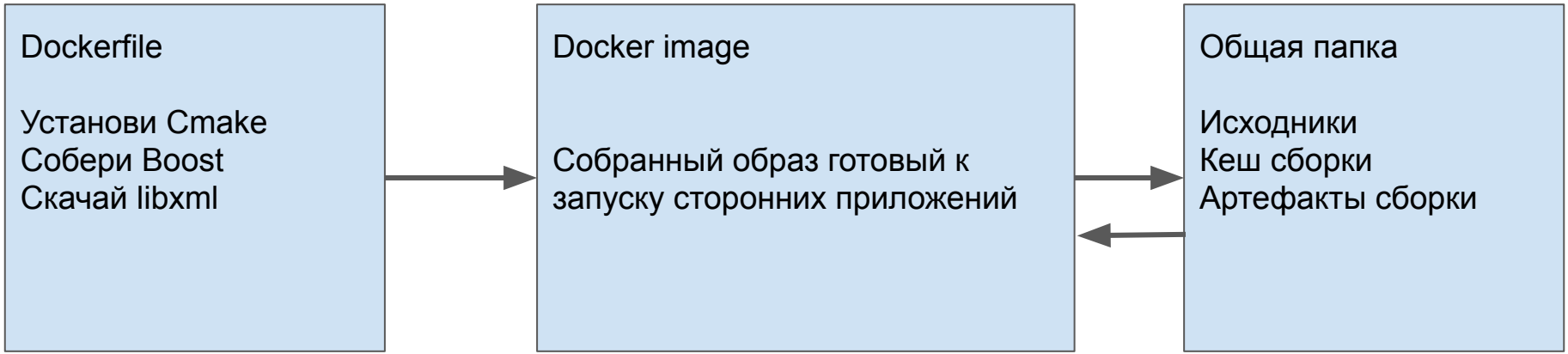
Менеджеры пакетов

Docker






Docker


Dockerfile
Установи Cmake
Собери Boost
Скачай libxml
Docker image
Собранный образ готовый к запуску сторонних приложений
Общая папка
Исходники
Кеш сборки
Артефакты сборки






DINS

Docker

Dockerfile для сборки

```
FROM centos:7.6.1810

COPY yum_packages.txt /tmp/yum_packages.txt

RUN yum --enablerepo=extras install epel-release -y

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






Docker

Для работы с контейнером нужно 2 простых команды

docker build -t imagename ./docker

docker run --rm -v $(pwd):/build_dir -v /tmp:/tmp imagename ./buildscript.sh





Docker

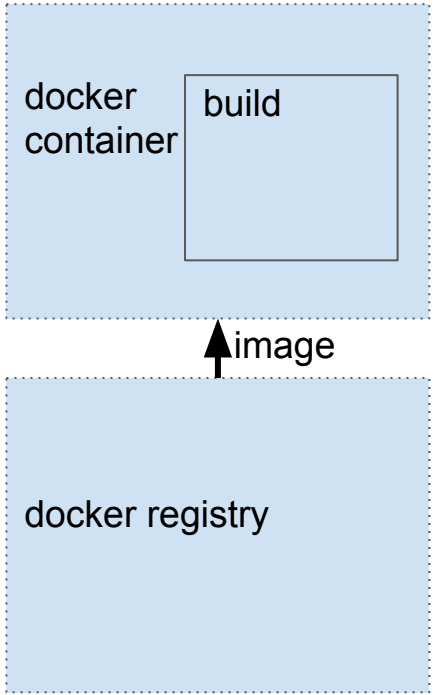
Но иногда приходится писать так:
пример из проекта

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





Пайплайн



Docker может использоваться не только билдером

Девелопмент и написание тестов на рабочей машине новым разработчиком

проблемы:

- Дебаг
- Подсветка синтаксиса





Дебаг

Можно:
Присоединяться к запущенному контейнеру и выполнять параллельно gdb через docker exec
Запускать дебагер 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






Дебаг

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 остались?

НЕТ

Изолированное окружение для сборки





Проблемы?

Помимо сборки в окружении надо запускать тесты

Модифицируем контейнер:
Юнит тесты при сборке
Интеграционные тесты - скрипт с несколькими приложениями внутри одного контейнера, docker compose





Пример

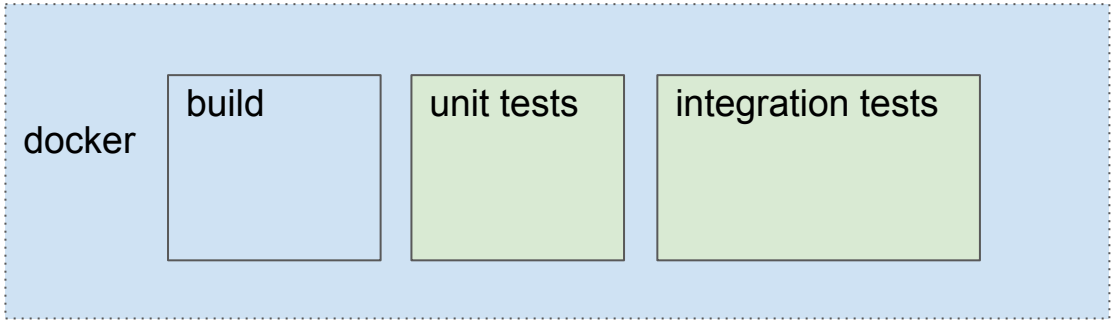
```
build:
  stage: build
  script:   - ./buildscript.sh docker make


  test:
  stage: test
  script: - bash ./buildscript.sh  docker run_tests
  only:  - merge_requests
```





Пайплайн







Проблемы?

Маленький фикс сломал половину тестов, а девелопер их даже не запустил





tests

40 min

10 sec

pull request

approve





DINS

tests

40 min

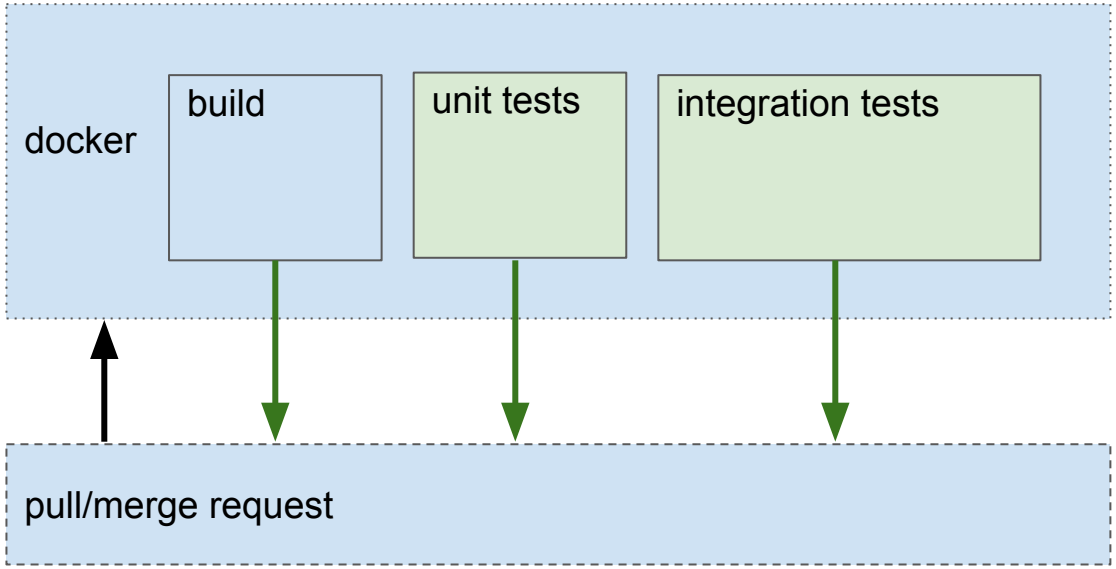
10 sec

pull request

approve

DINS

Пайплайн


docker
build
unit tests
integration tests
pull/merge request

Проблемы?

Не везде в рамках единого мерж реквеста нужна обратная связь

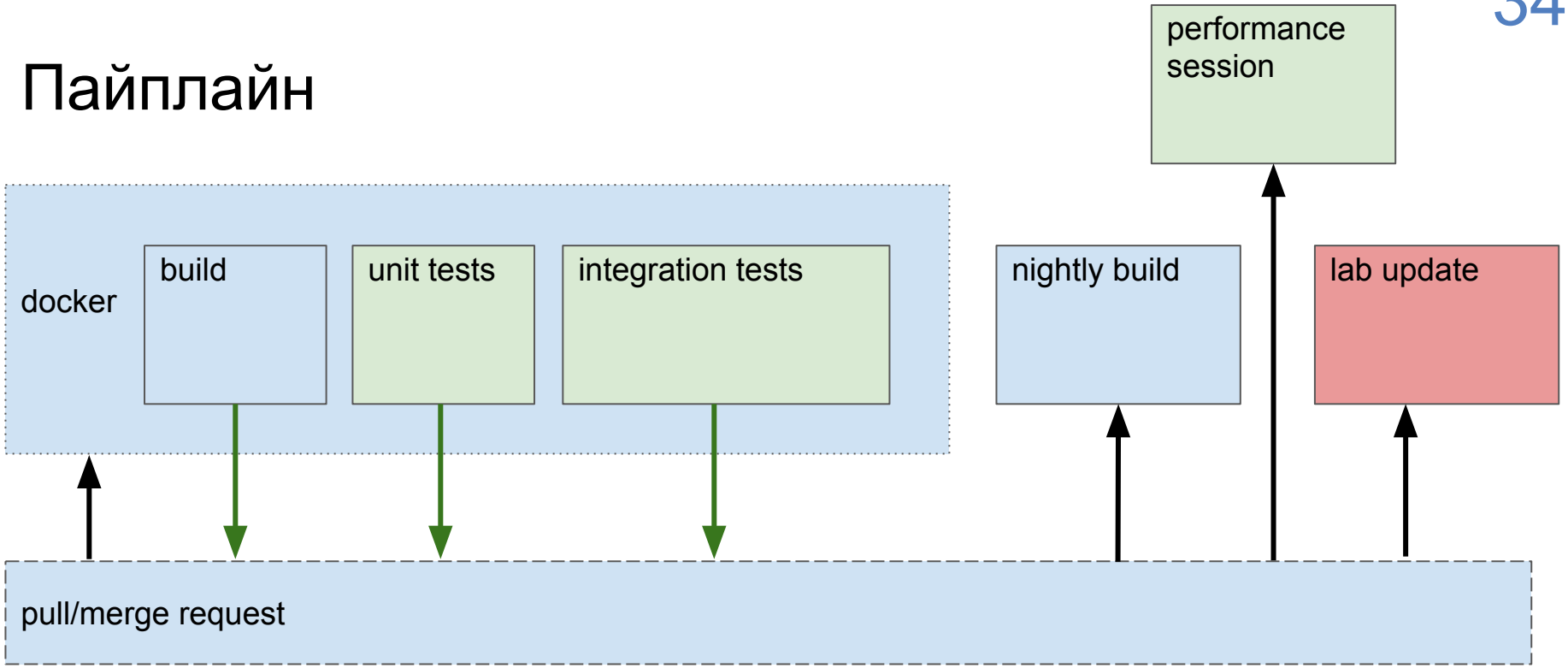
пример:

Проверка стабильности системы (3 дня)
Обновление лабы
Nightly build (он выполняется ночью)





Пайплайн

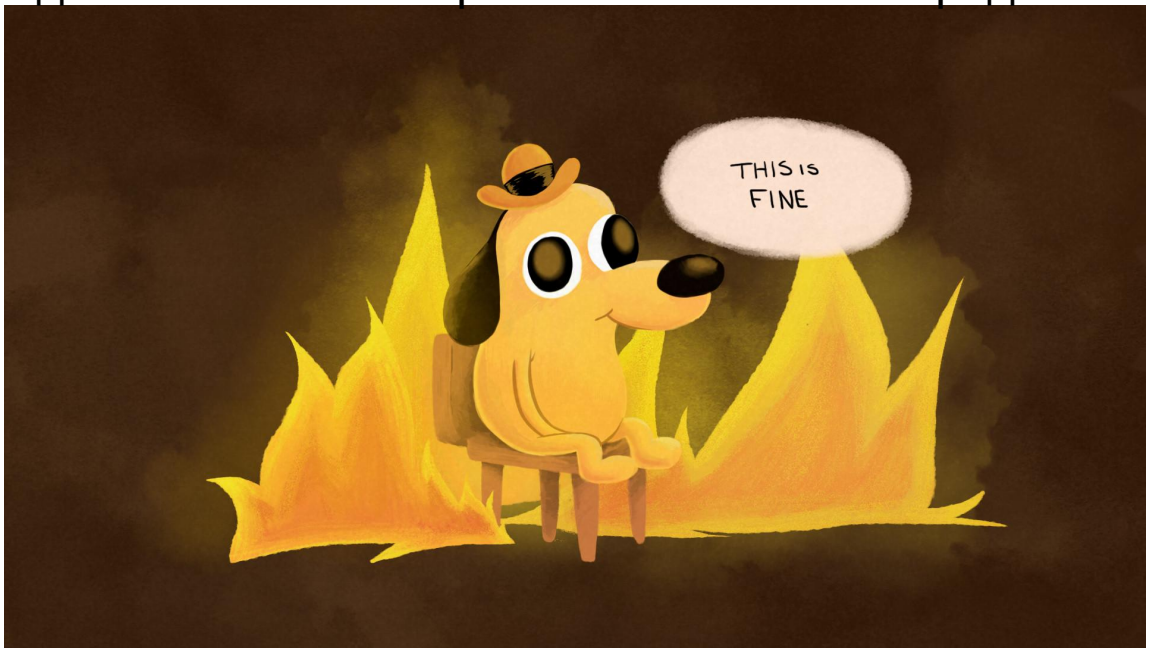


Встроенный пайплайн

Евенты

Проблема?

Баг не был найден на этапе тестирования и вылез на продакшен



Решение

Изменим систему

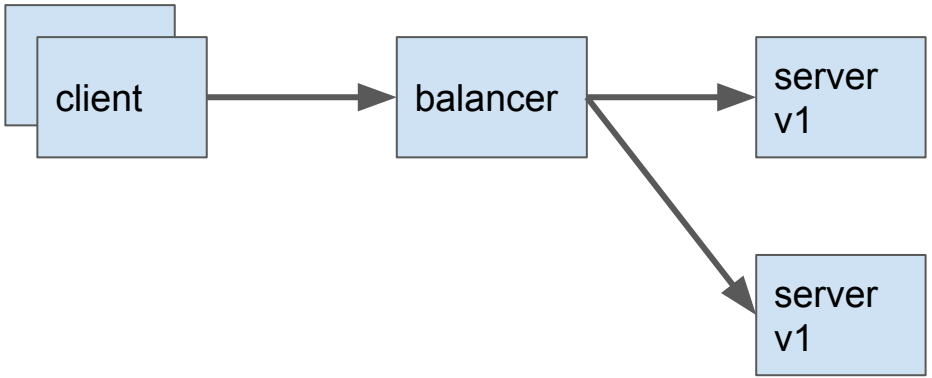
Клиента теперь два.
Если клиент получил невалидный ответ, он пробует еще раз

Два сервера, распределение нагрузки между ними через лоад балансер





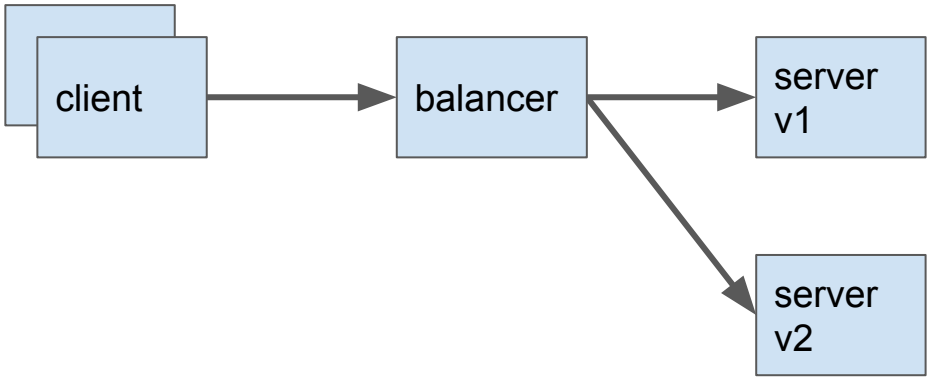
Обновление сервиса







Обновление сервиса







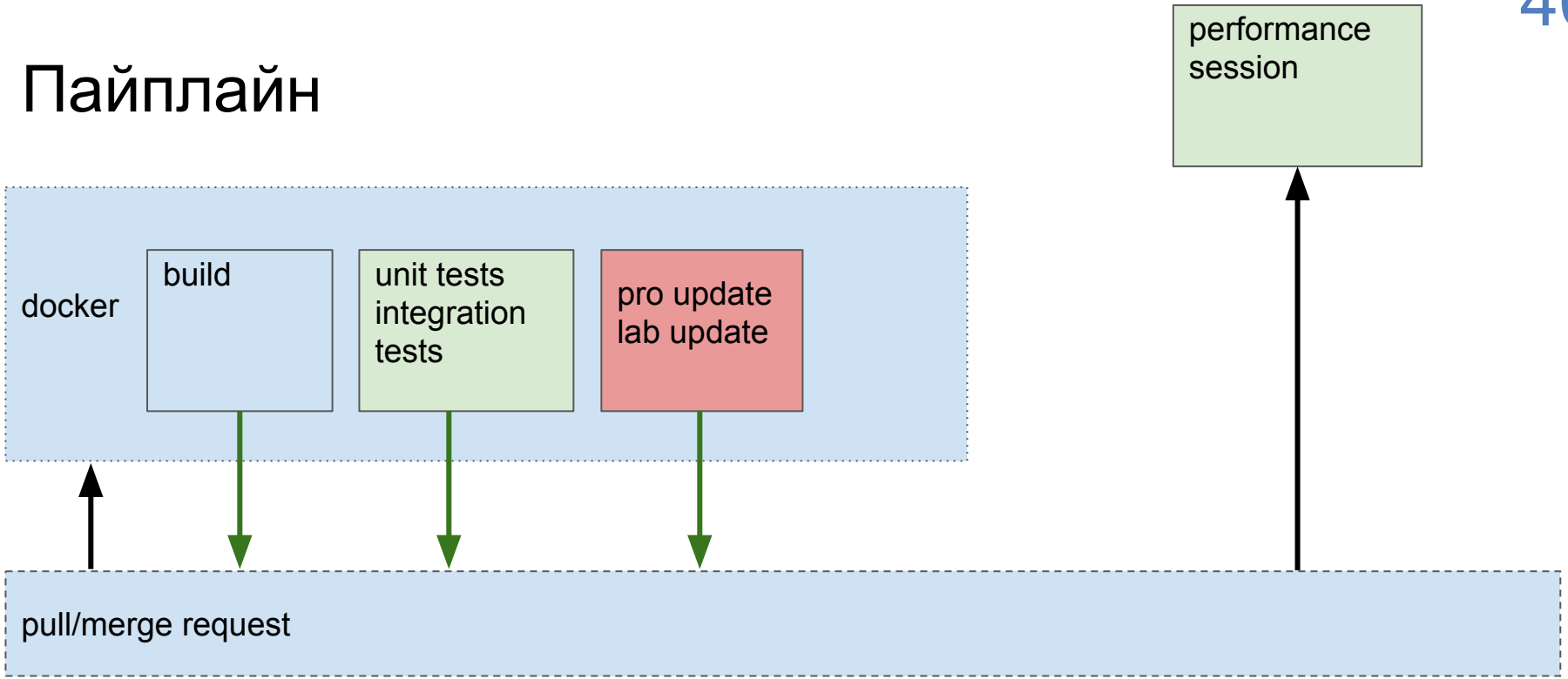
Обновим прод в пайплайне

Не повторяйте это без правильно настроенного пайплайна





Пайплайн


performance session
docker
build
unit tests integration tests
pro update lab update
pull/merge request

Проблемы?

Теперь ошибки которые не нашли наши тесты не влияют на систему, но мы слишком поздно понимаем, что все плохо





Решение?

Начнем собирать метрики с лабы и продакшена






Метрики

Собираем с пода/конкретного сервера через:
Http, snmp, zabbix protocol

Рисуем красивые графики:
Zabbix, grafana

Настраиваем триггеры

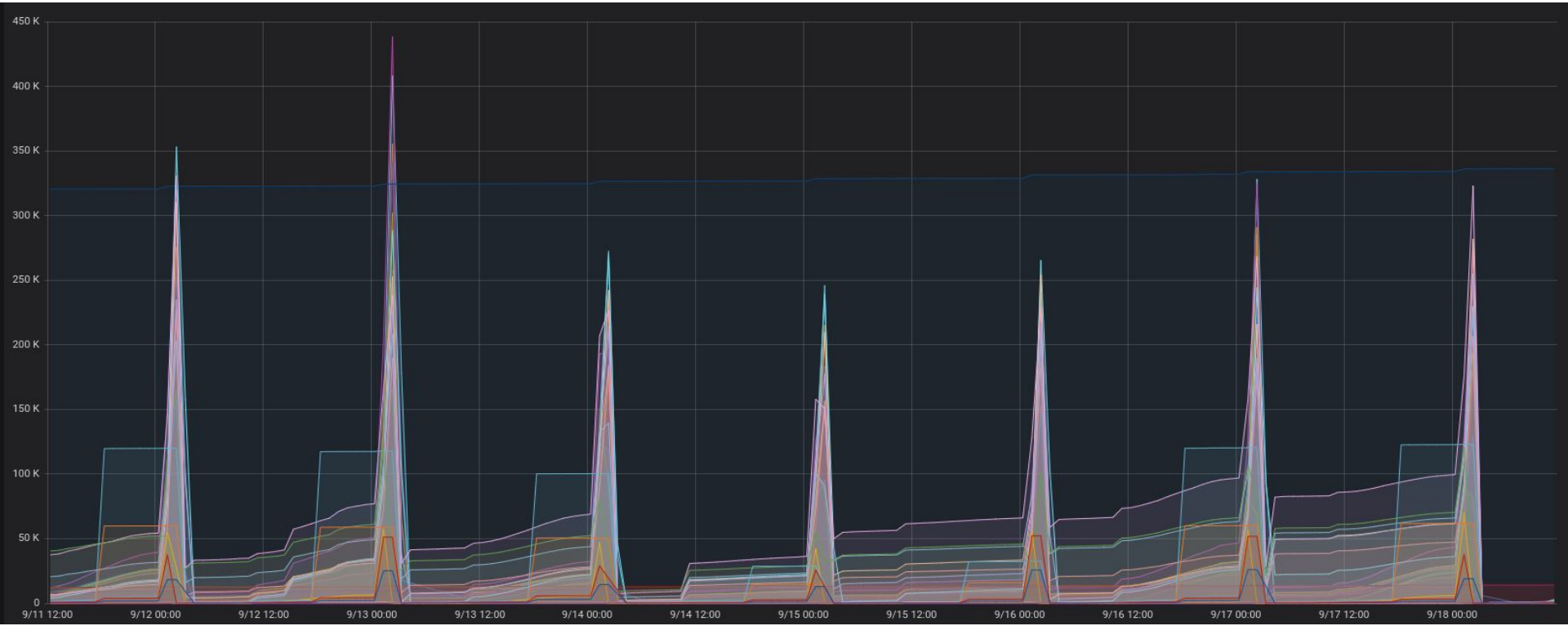




Zabbix


Filter ▲
Zoom: 5m 15m 30m 1h 2h 3h 6h 12h 1d 3d 7d 14d 1m All
2019-09-18 13:37 - 2019-09-18 14:37 (now!)
«« 1m 7d 1d 12h 1h 5m | 5m 1h 12h 1d 7d 1m »»
1h fixed
gci
1.0 GB
0.8 GB
0.6 GB
0.4 GB
0.2 GB
09-18 13:37
14:00
last min avg max
Kamailio shared memory free size [all] 1021.35 MB 253.34 MB 719.63 MB 1021.38 MB
Trigger: Free of kamailio shared memory less than 157286400 [< 157286400]
Trigger: Free of kamailio shared memory less than 100MB [< 100M]
Trigger: Free of kamailio shared memory less than 50MB [< 50M]

Grafana


450 K
400 K
350 K
300 K
250 K
200 K
150 K
100 K
50 K
0
9/11 12:00
9/12 00:00
9/12 12:00
9/13 00:00
9/13 12:00
9/14 00:00
9/14 12:00
9/15 00:00
9/15 12:00
9/16 00:00
9/16 12:00
9/17 00:00
9/17 12:00
9/18 00:00







DINS

Триггеры

Пишем письмо человеку который сломал билд
Последнему автору мерж реквеста

Дублируем письмо команде





Проблема

Мы катим не то, что тестируем (интеграциоными тестам)

Катим нативное приложение
Тестируем контейнер





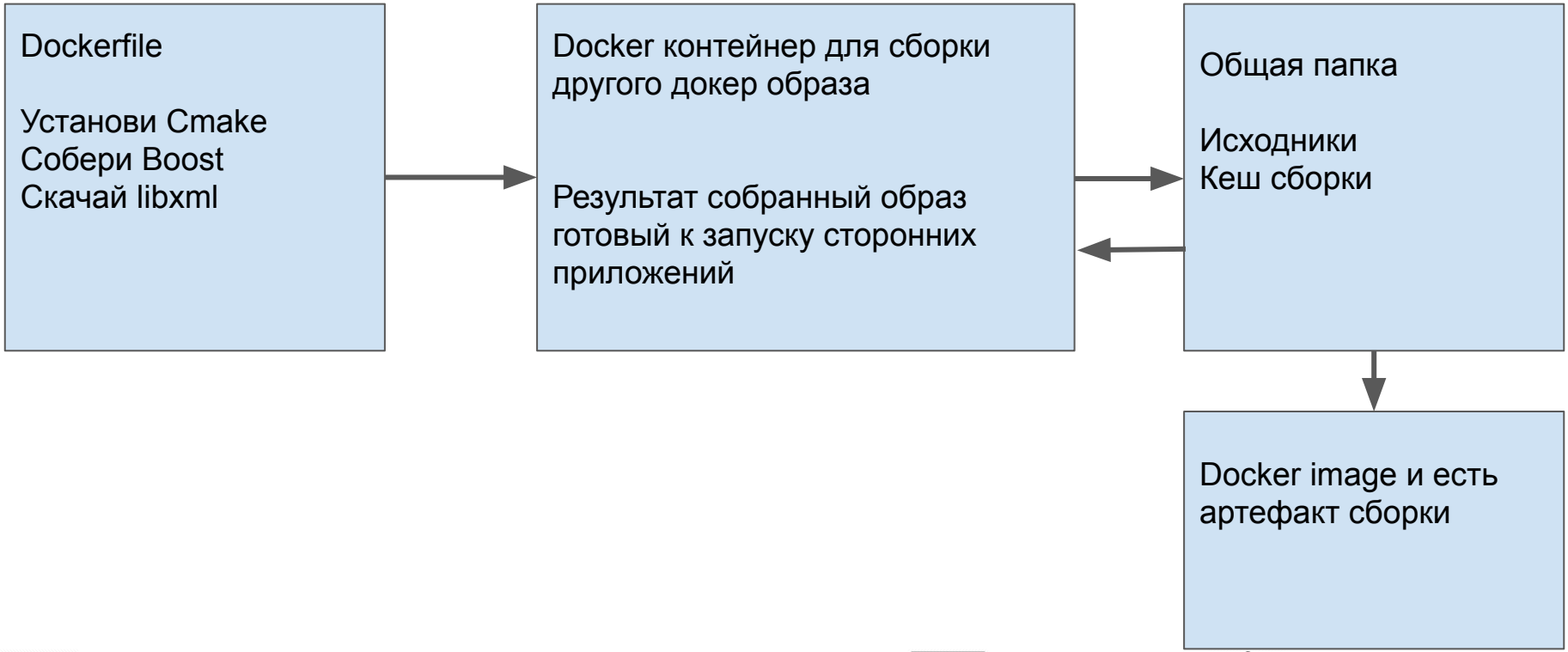
Решение

Docker контейнер для приложения






Docker контейнер для приложения



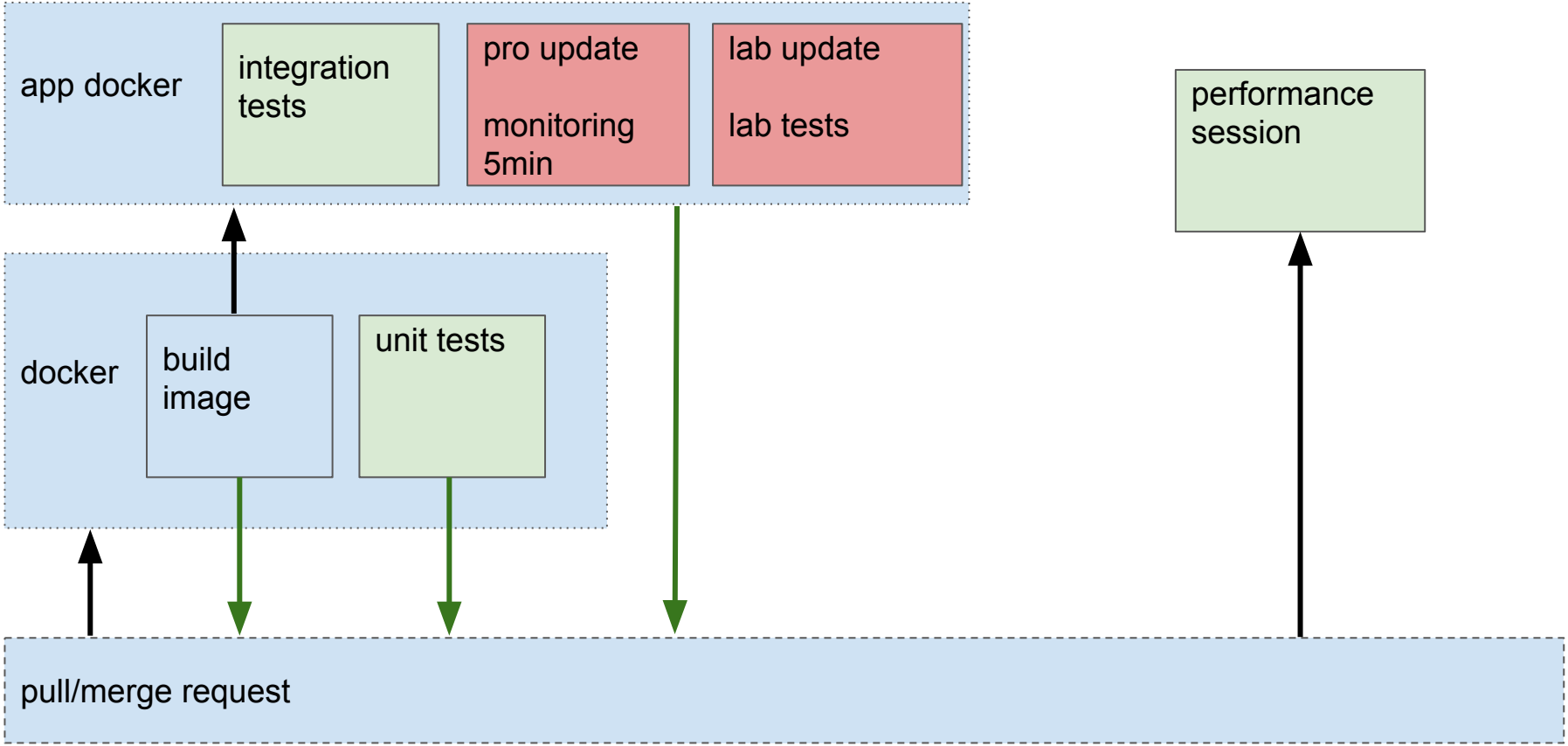
Docker контейнер для приложения

Build container	**App container**
Кэшируется все что можно	Минимальный размер
Дополнительные инструменты	Нет ничего кроме приложения
Максимальная конфигурируемость/гибкость	Простота использования





Пайплайн


app docker
integration tests
pro update
monitoring 5min
lab update
lab tests
performance session
docker
build image
unit tests
pull/merge request

Проблема

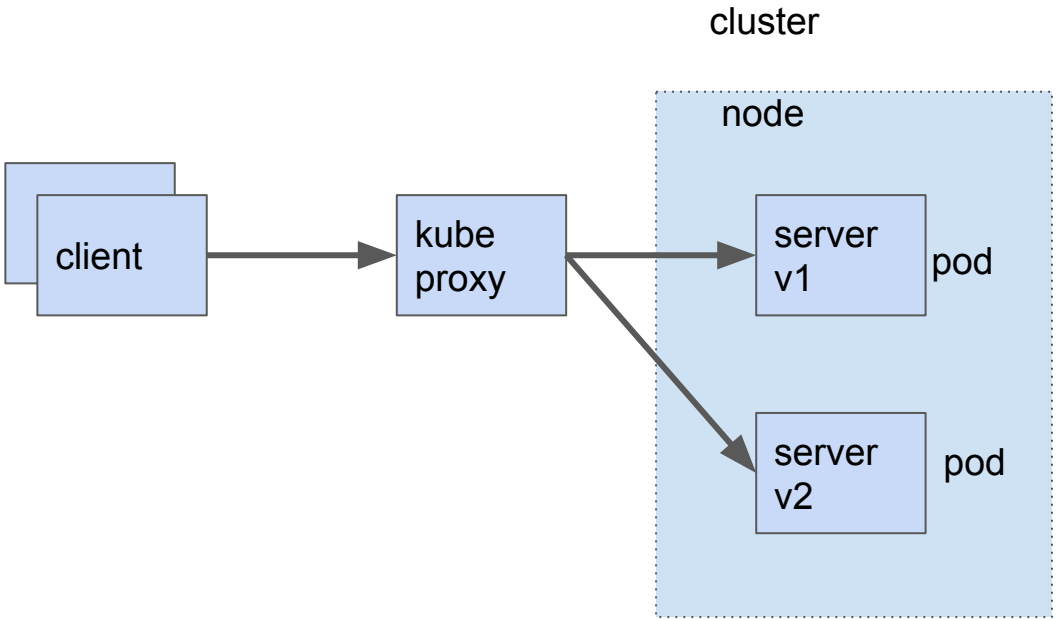
Мы не можем организованно управлять приложениями в продакшене

Каждый красный блок (update pro/build) это много работы



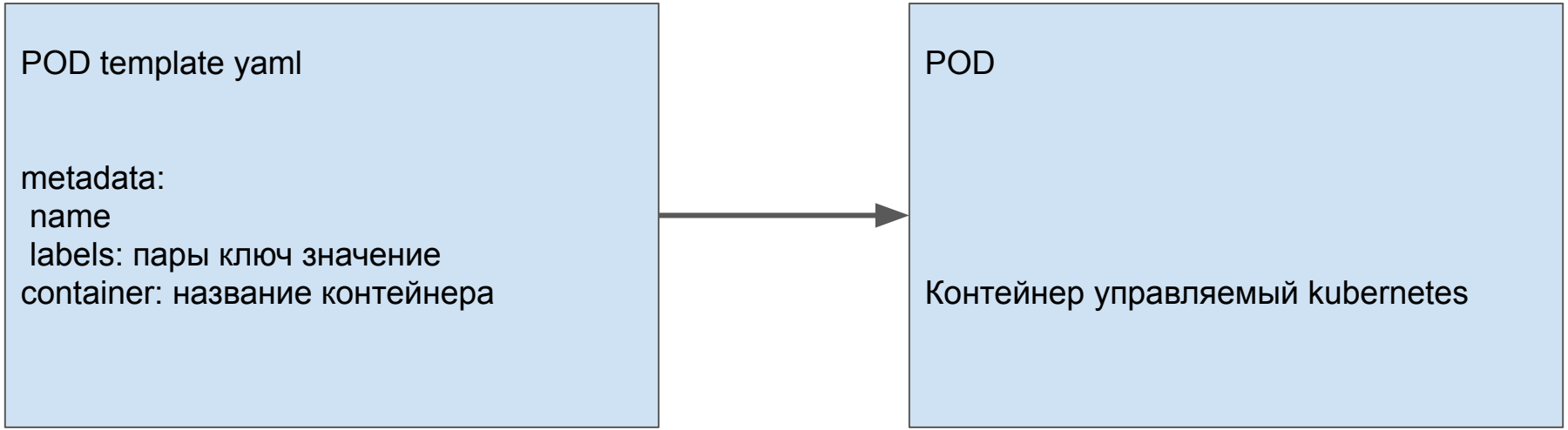


Kubernetes



Kubernetes

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



А как катить?

Контроллеры

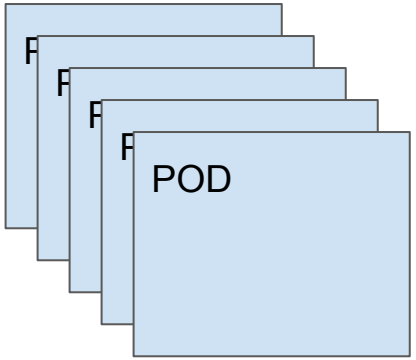
Kubernetes

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

Controller (deployment) yaml

selector
POD template

Количество



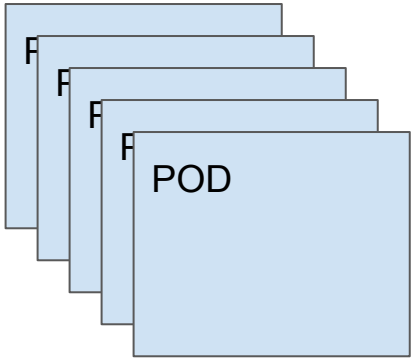
Kubernetes

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

Controller (deployment) yaml

selector
POD template

Количество



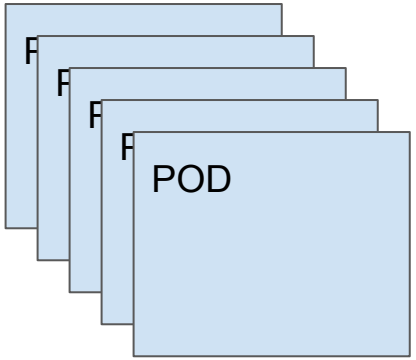
Kubernetes

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

Controller (deployment) yaml

selector
POD template НОВЫЙ

Количество



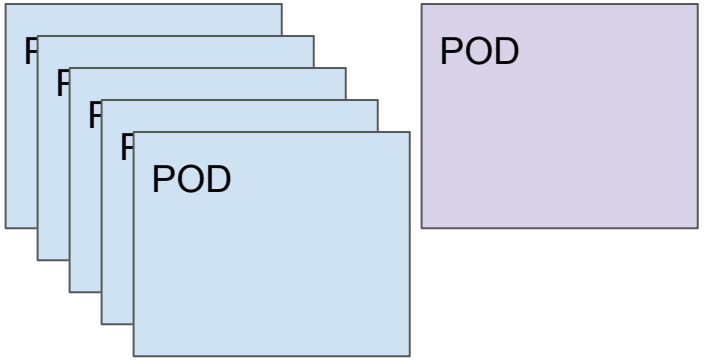
Kubernetes

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

Controller (deployment) yaml

selector
POD template НОВЫЙ

Количество



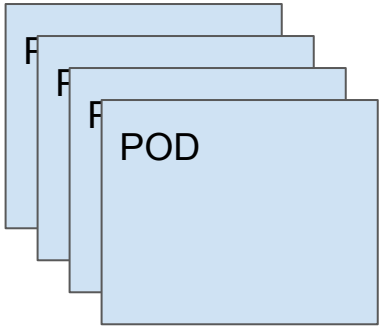
Kubernetes

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

Controller (deployment) yaml

selector
POD template НОВЫЙ

Количество







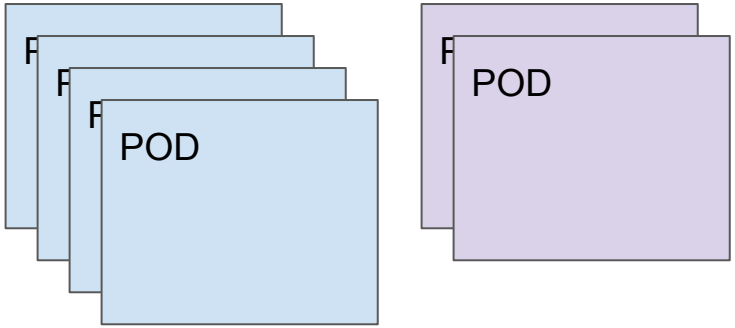

Kubernetes

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

Controller (deployment) yaml

selector
POD template НОВЫЙ

Количество



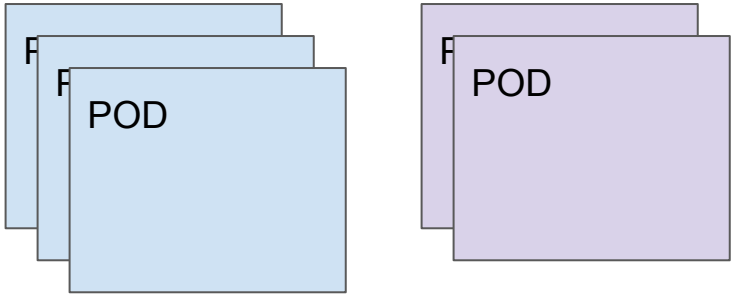
Kubernetes

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

Controller (deployment) yaml

selector
POD template НОВЫЙ

Количество



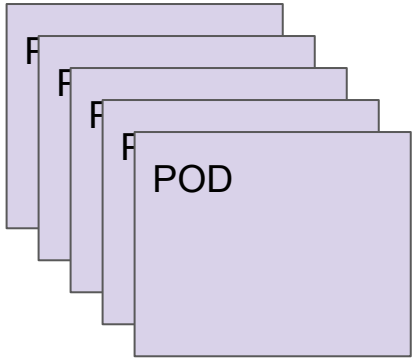
Kubernetes

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

Controller (deployment) yaml

selector
POD template НОВЫЙ

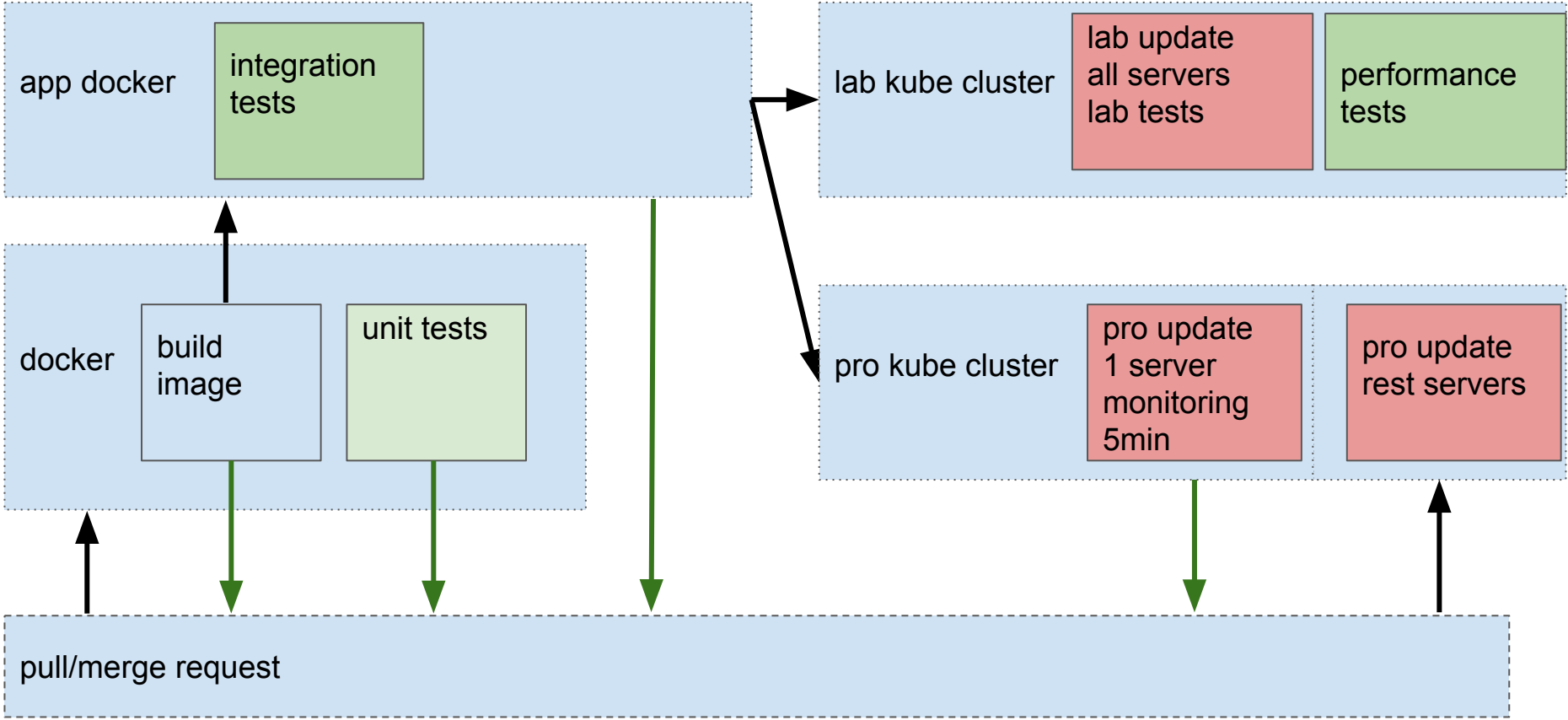
Количество



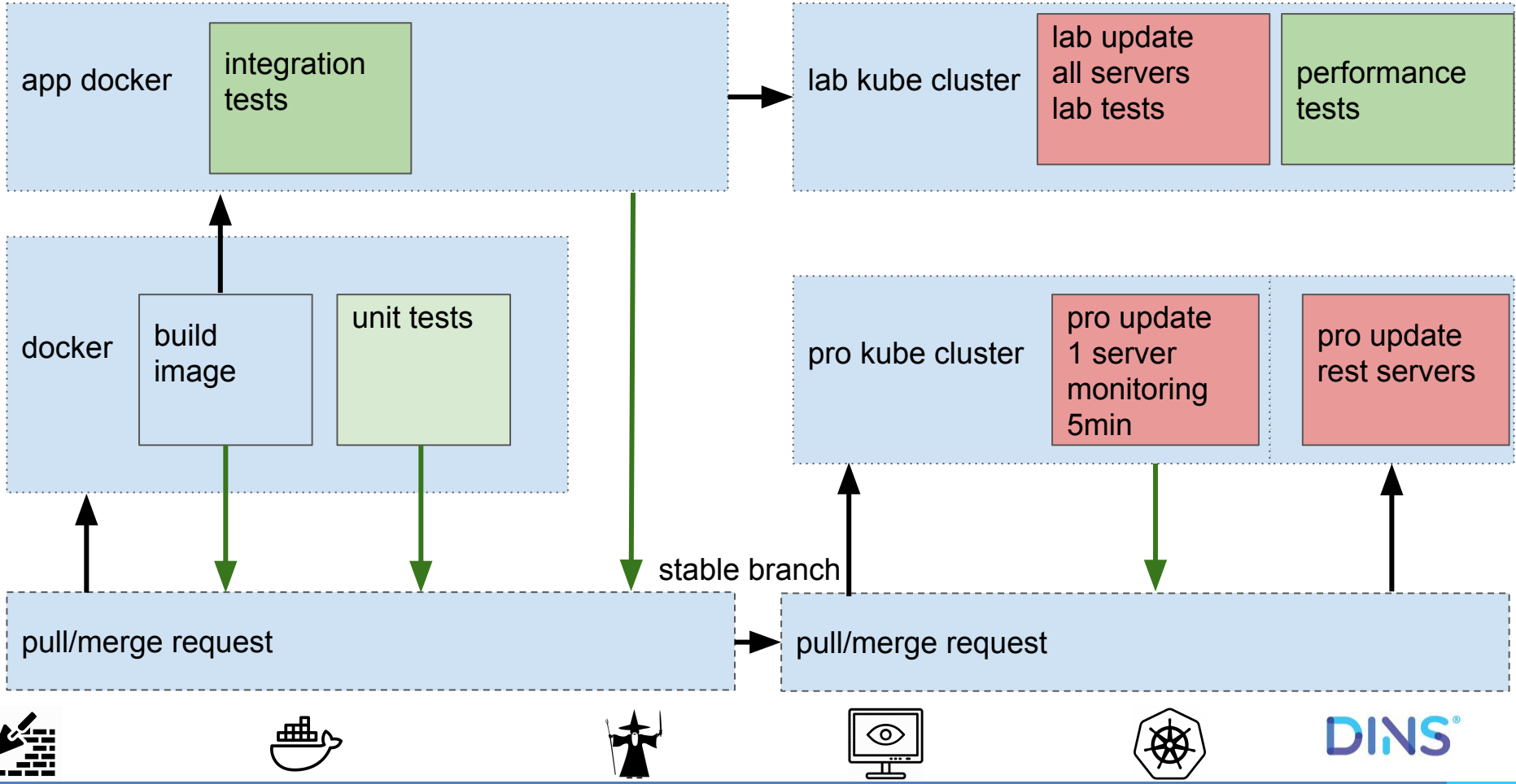





Пайплайн


app docker
integration tests
lab kube cluster
lab update all servers lab tests
performance tests
docker
build image
unit tests
pro kube cluster
pro update 1 server monitoring 5min
pro update rest servers
pull/merge request

Пайплайн


app docker
integration tests
lab kube cluster
lab update all servers lab tests
performance tests
docker
build image
unit tests
pro kube cluster
pro update 1 server monitoring 5min
pro update rest servers
stable branch
pull/merge request
pull/merge request

Что мы получили

Детерминированный билд
Контроль выкатки
Проверка на этапе сборки
Проверка на этапе выкатки
Моментальная точная обратная связь






Что **мы** получили

Факты:

Пайплайн не зависит от инструмента

Надо начинать с малого

Повторить это не так сложно





Спасибо