

Что мы знаем о микросервисах?

Вадим Мадисон (Авито)

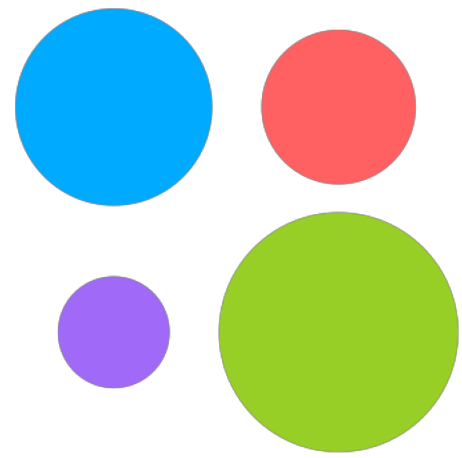


HighLoad⁺⁺

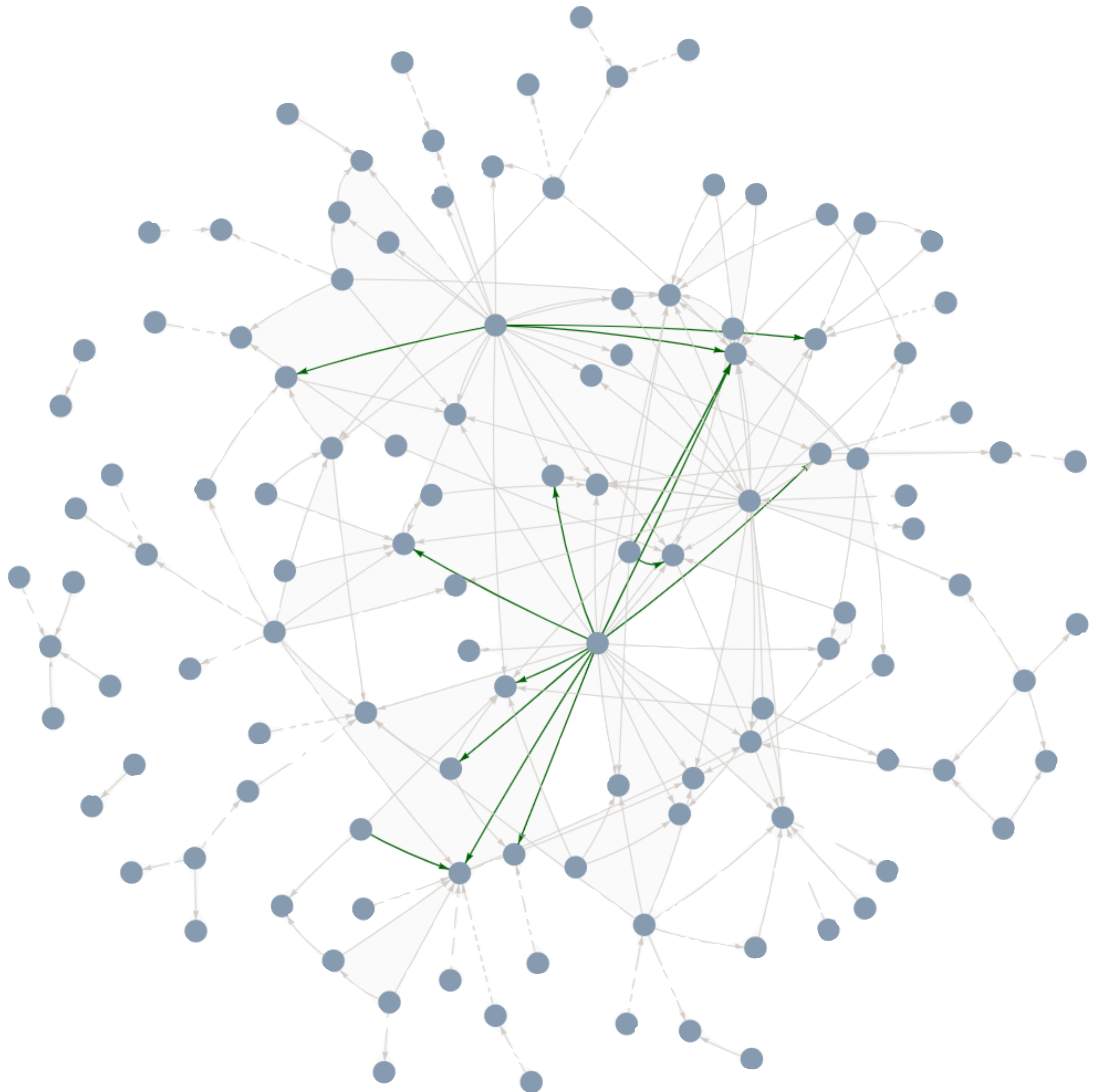
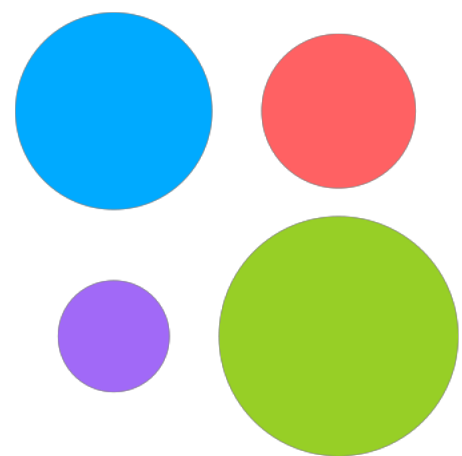
Профессиональная конференция
разработчиков высоконагруженных
систем

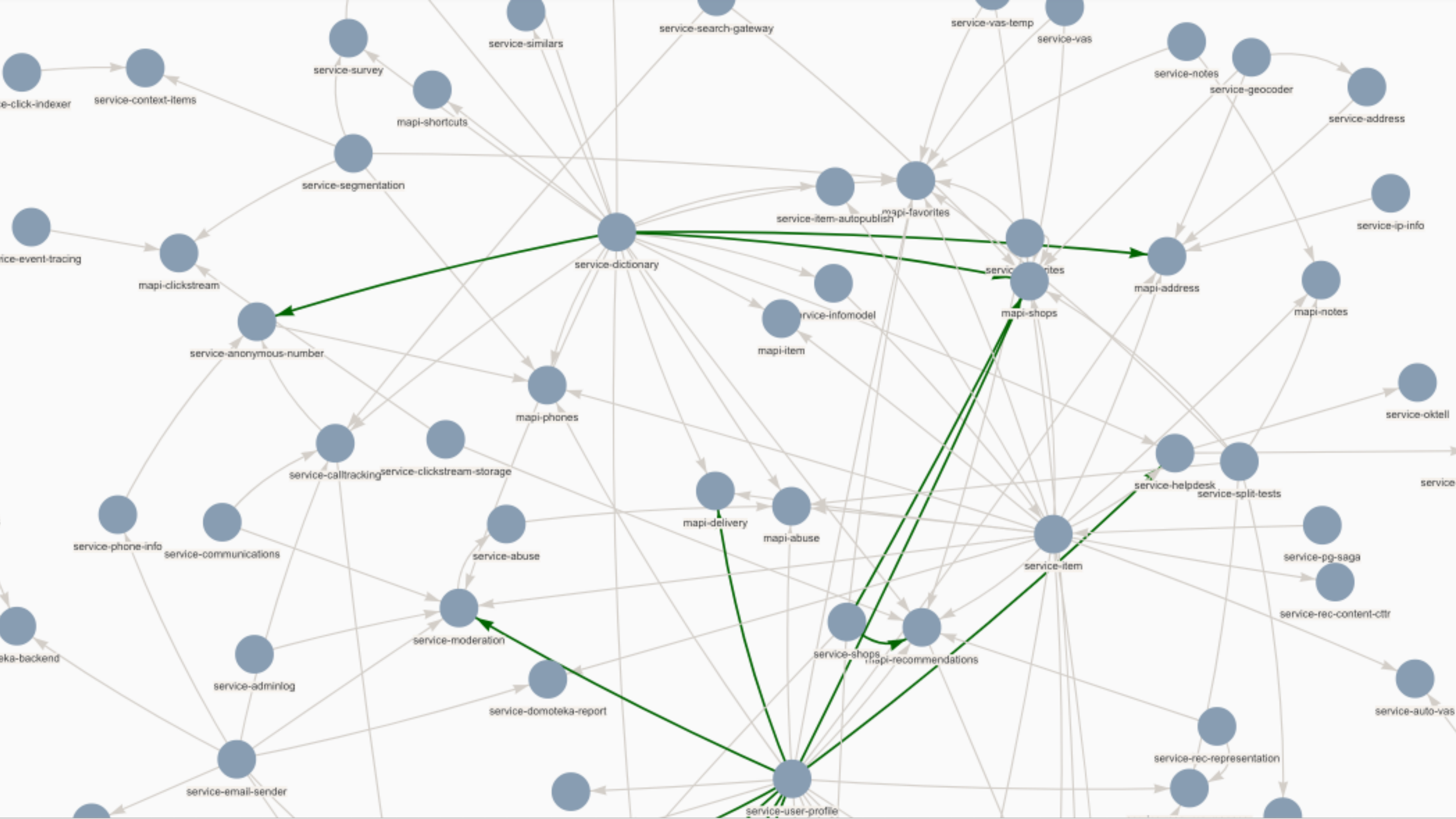


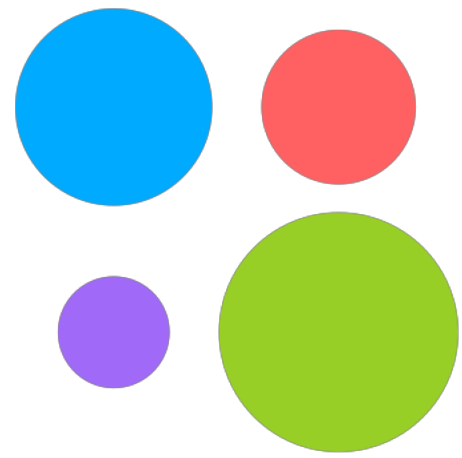
- TOP-2 крупнейших досок объявлений в мире
- 1 5000 000 объявлений в сутки
- 20 000+ rps к бэкенду
- 800+ серверов



Жизнь после Kubernetes?

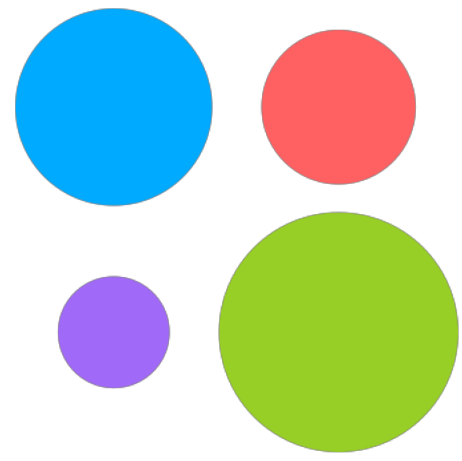






В чем проблема?

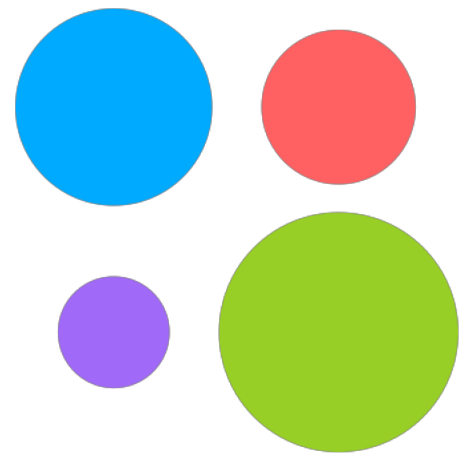
Большое количество сервисов



В чем проблема?

Большое количество сервисов

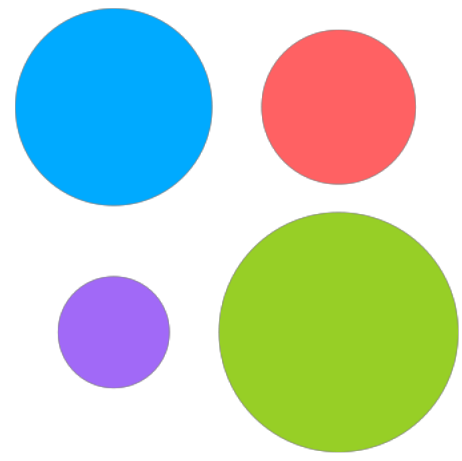
- Много разных репозиториев



В чем проблема?

Большое количество сервисов

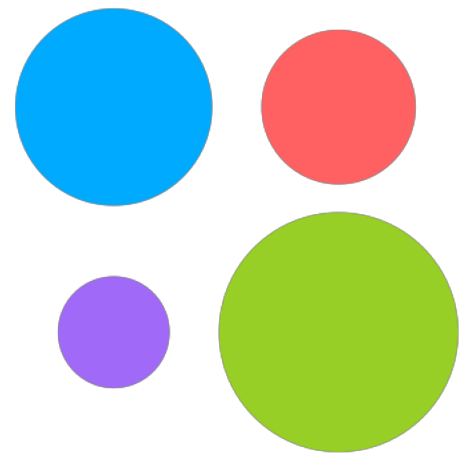
- Много разных репозиторийев
- Много команд, которые пишут код, не пересекаясь с другими



В чем проблема?

Большое количество сервисов

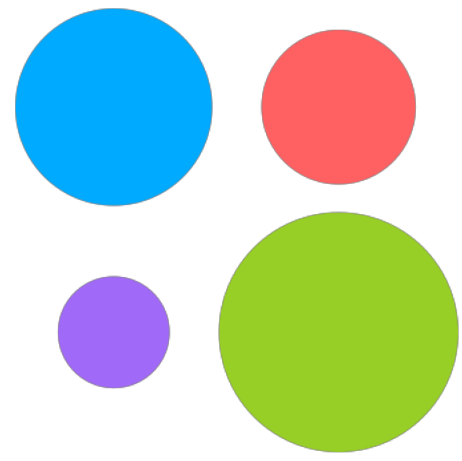
- Много разных репозиторийев
- Много команд, которые пишут код, не пересекаясь с другими
- Непересекающиеся контексты



В чем проблема?

Большое количество сервисов

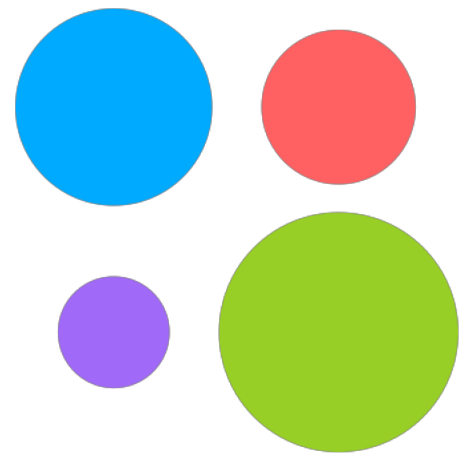
- Много разных репозиторийев
- Много команд, которые пишут код, не пересекаясь с другими
- Непересекающиеся контексты
- Фрагментарность данных
- ...



В чем проблема?

Большое количество инфраструктурных инструментов

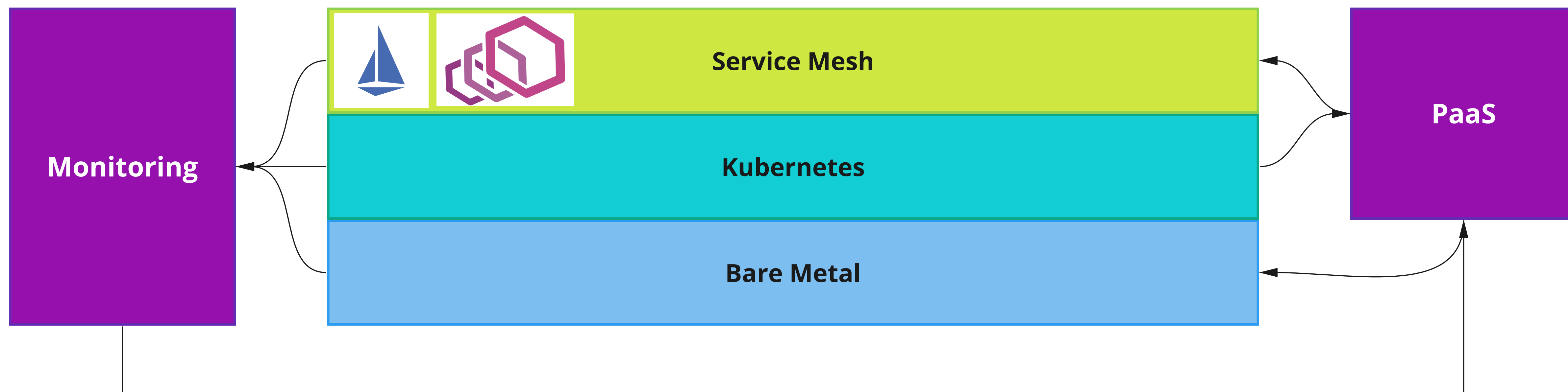
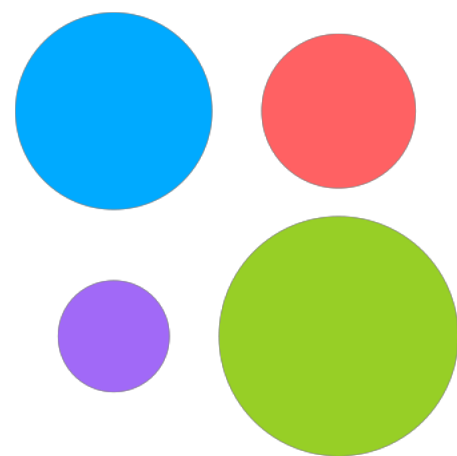
- Логирование
- Трассировка запросов (Jaeger)
- Агрегация ошибок (Sentry)
- Статусы / сообщения / события из Kubernetes
- Race Limit / Circuit Breaker (Hystrix)
- Связность сервисов (Istio)
- Сборка (Teamcity)
- Мониторинг (Grafana)
- ...

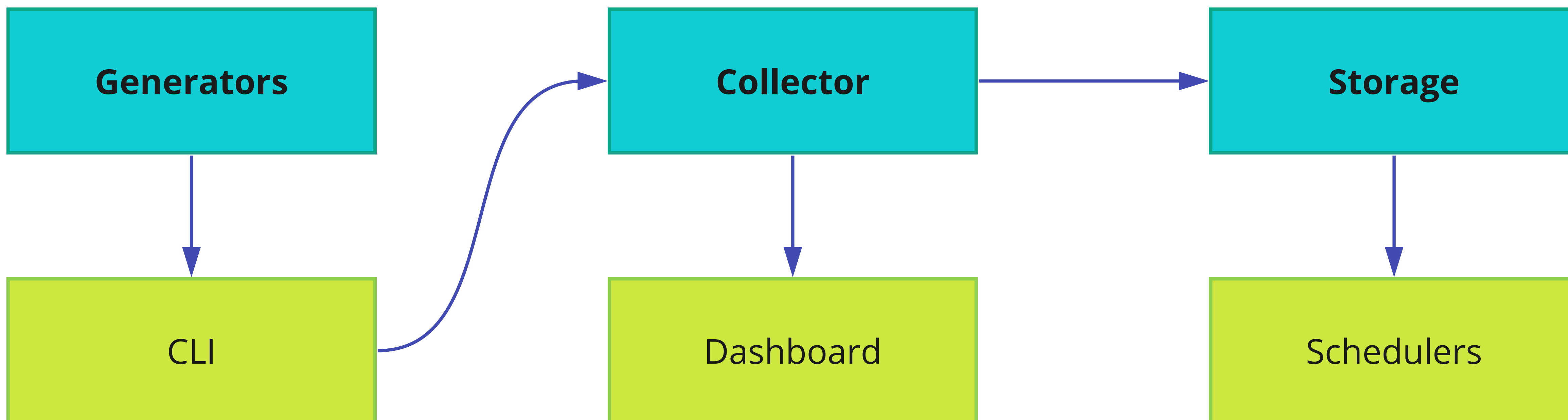
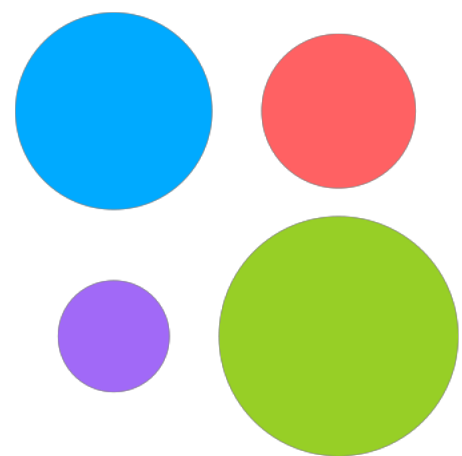


В чем проблема?

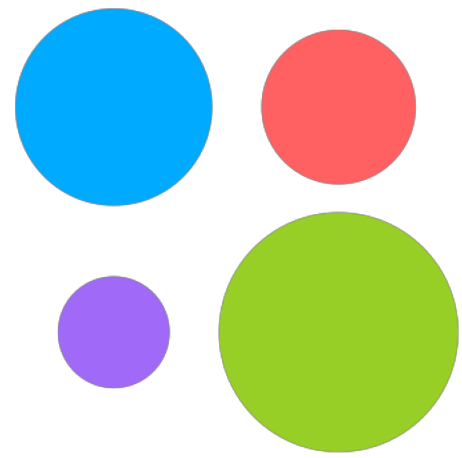
Большое количество инфраструктурных инструментов

- Логирование
 - Трассировка запросов (Jaeger)
 - Агрегация ошибок (Sentry)
 - Статусы / сообщения / события из Kubernetes
 - Race Limit / Circuit Breaker (Hystrix)
 - Связность сервисов (Istio)
 - Сборка (Teamcity)
 - Мониторинг (Grafana)
 - ...
- Slack
 - Email
 - Confluence
 - Stash
 - ...

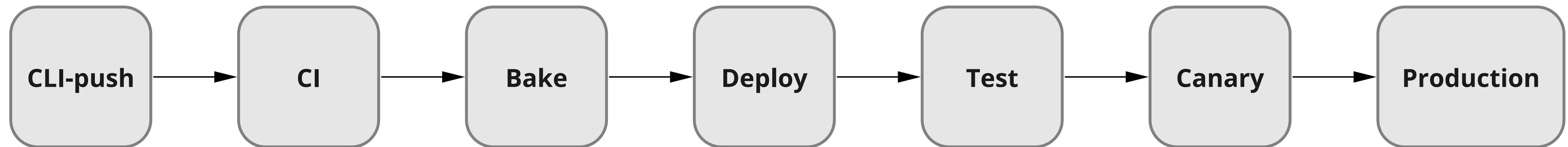




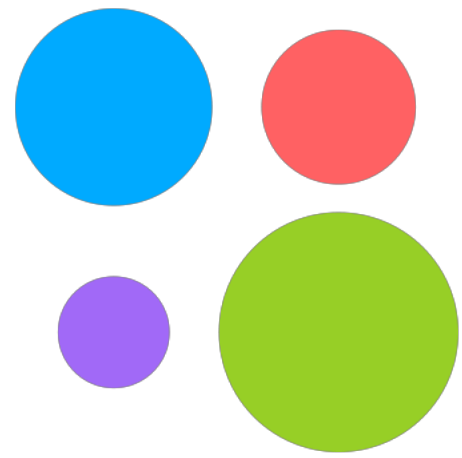
Как это поддерживать?



Стандартный Pipeline

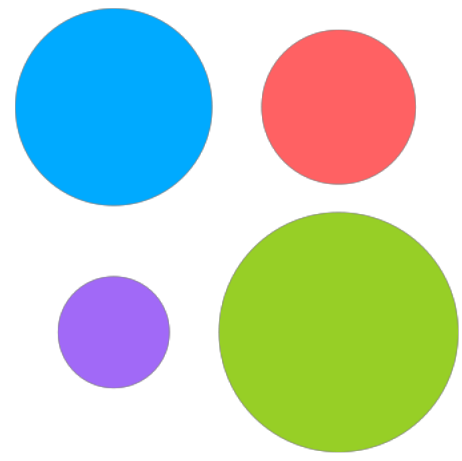


CLI-push



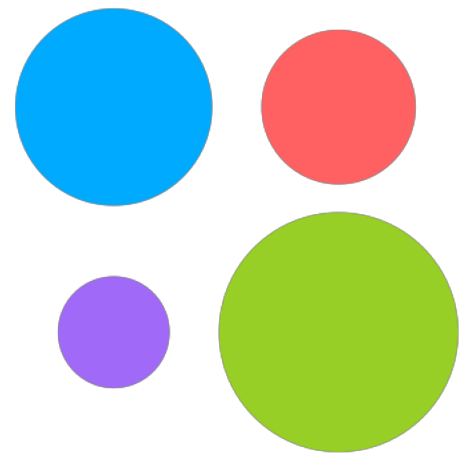
- Создание сервиса из шаблона

Что дает?



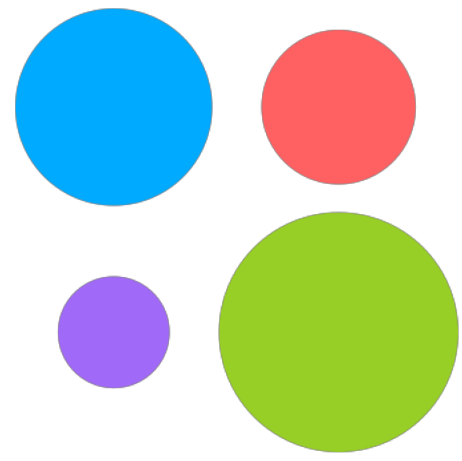
Что дает?

- Создание сервиса из шаблона
- Развертывание инфраструктуры для локальной разработки одной командой



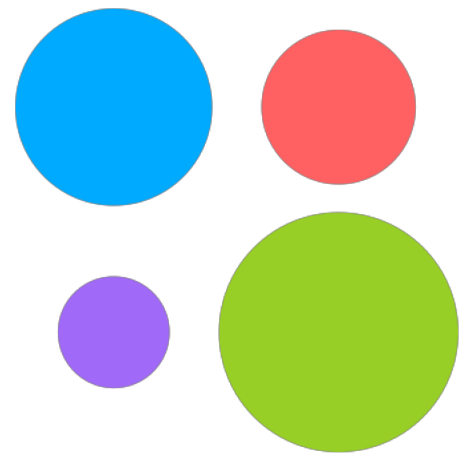
Что дает?

- Создание сервиса из шаблона
- Развертывание инфраструктуры для локальной разработки одной командой
- Подключение БД одной командой (не требует конфигурирования)



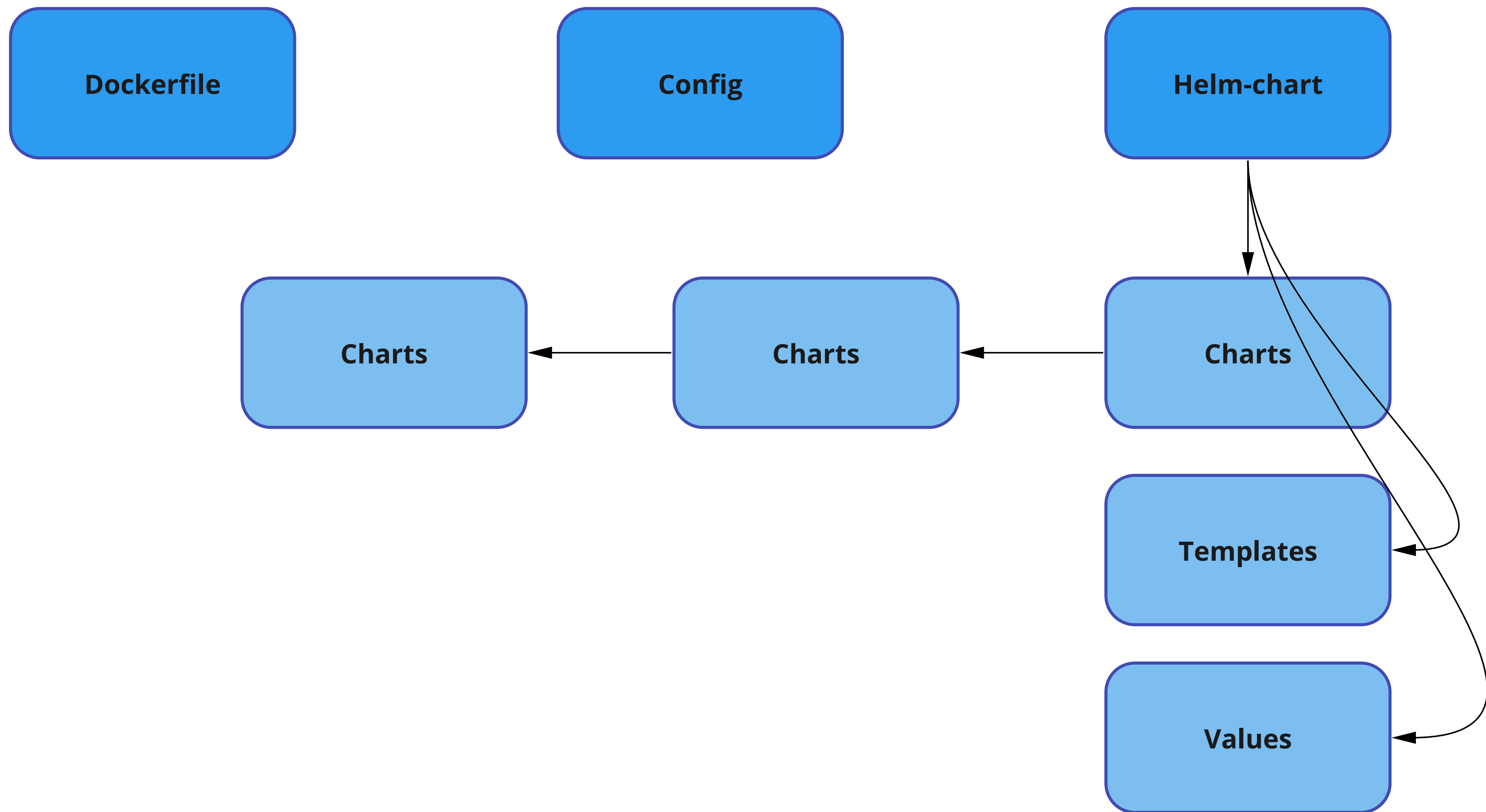
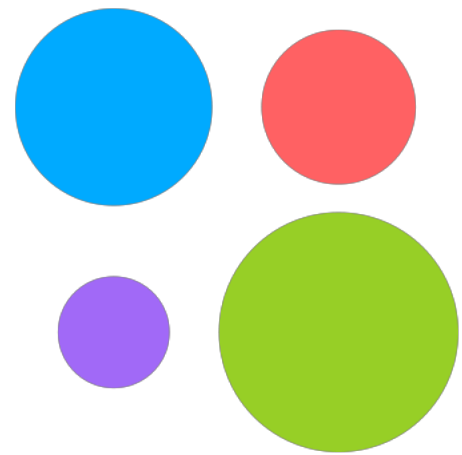
Что дает?

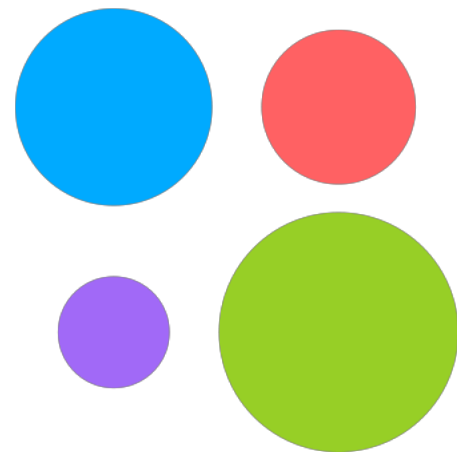
- Создание сервиса из шаблона
- Развертывание инфраструктуры для локальной разработки одной командой
- Подключение БД одной командой (не требует конфигурирования)
- Live-сборка



Что дает?

- Создание сервиса из шаблона
- Развертывание инфраструктуры для локальной разработки одной командой
- Подключение БД одной командой (не требует конфигурирования)
- Live-сборка
- Генерация болванок автотестов



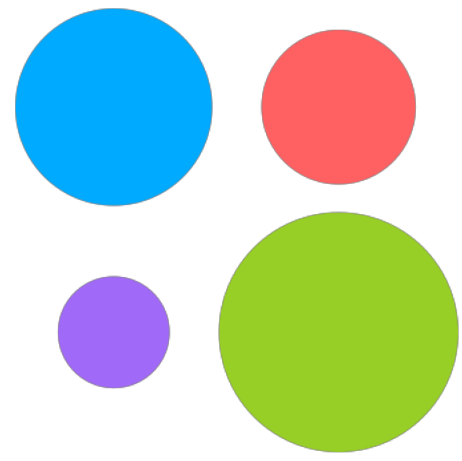


Dockerfile

**Config
(app.toml)**

File: app.toml

```
1 name = "paas-gateway"
2 [engine]
3 name = "golang"
4 version = "1.10"
5 size = "small"
6
7 replicas = 1
8
9 [envs.staging]
10 replicas = 3
11
12 [envs.prod]
13 replicas = 5
14
15 [env_vars]
16 GRAPHITE_URL = "http://graphite.msk.avito.ru"
17
18 [[dependencies]]
19 name = "service-service-events"
20
21 [[dependencies]]
22 name = "service-atlas"
```

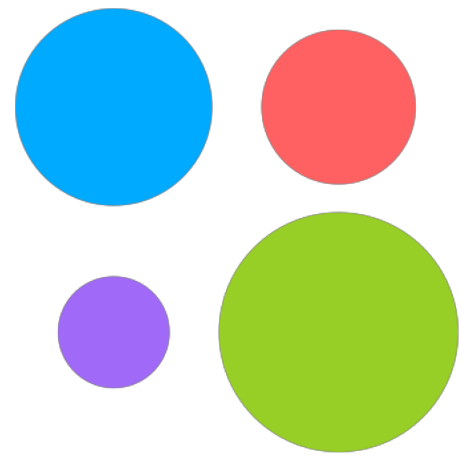


CLI-push

Базовая валидация

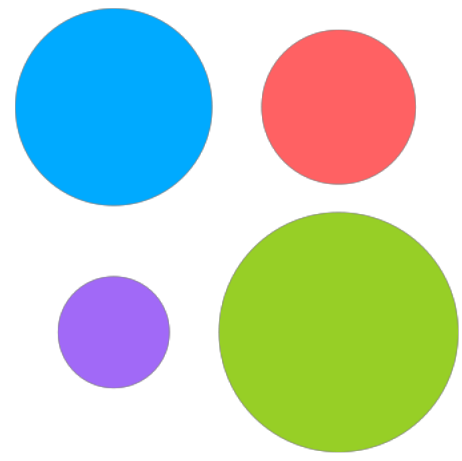
- Проверка наличия Dockerfile
- Проверка app.toml
- Наличие документации
- Зависимости
- Указаны правила алертов (владелец сервиса сам настраивает метрики которые нужно мониторить)
- ...

Документация



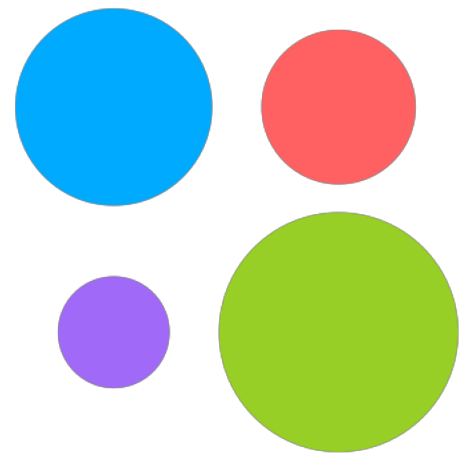
Что должно быть?

- Описание сервиса – несколько предложений текста, описывающих, что делает сервис



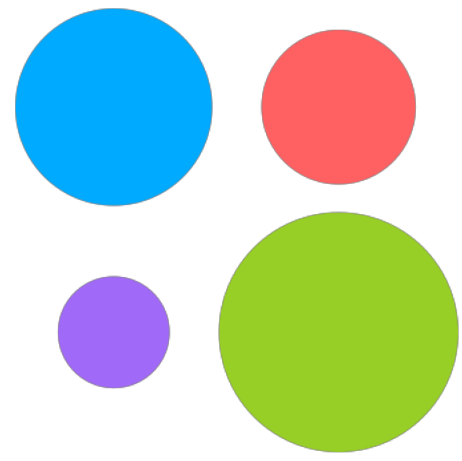
Что должно быть?

- Описание сервиса – несколько предложений текста, описывающих, что делает сервис
- Ссылка на диаграмму архитектуры



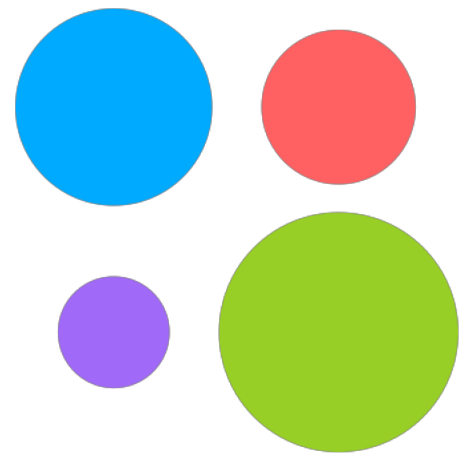
Что должно быть?

- Описание сервиса – несколько предложений текста, описывающих, что делает сервис
- Ссылка на диаграмму архитектуры
- Runbook



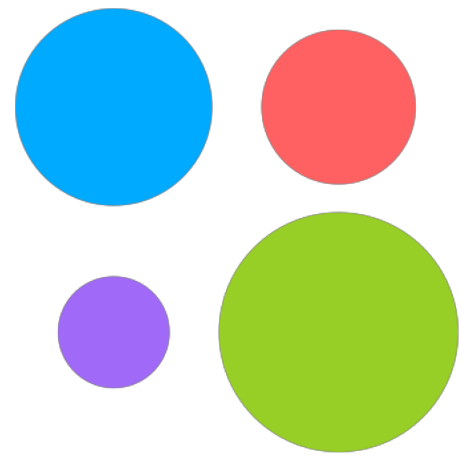
Что должно быть?

- Описание сервиса – несколько предложений текста, описывающих, что делает сервис
- Ссылка на диаграмму архитектуры
- Runbook
- FAQ



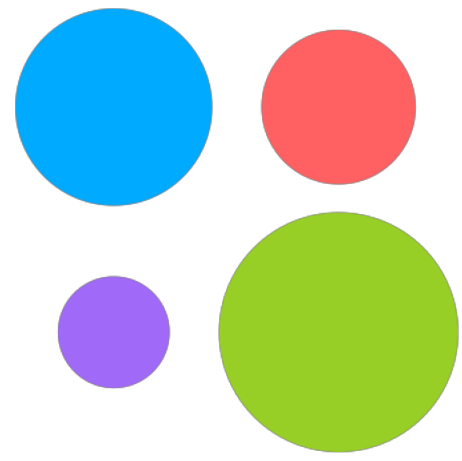
Что должно быть?

- Описание сервиса – несколько предложений текста, описывающих, что делает сервис
- Ссылка на диаграмму архитектуры
- Runbook
- FAQ
- Описание endpoint'ов API



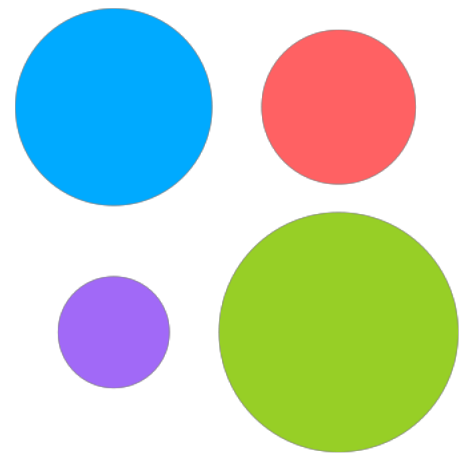
Что должно быть?

- Описание сервиса – несколько предложений текста, описывающих, что делает сервис
- Ссылка на диаграмму архитектуры
- Runbook
- FAQ
- Описание endpoint'ов API
- Labels – привязка к продукту, функциональности, структурному подразделению



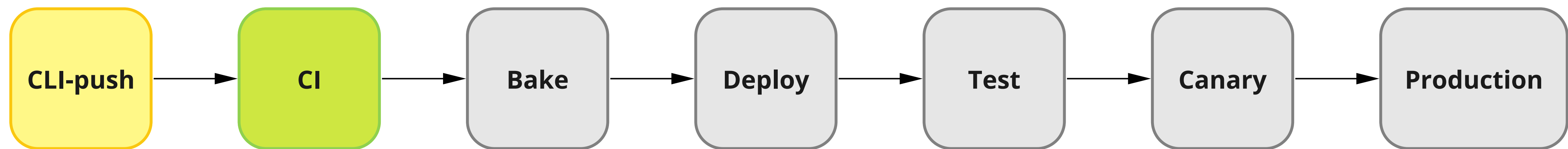
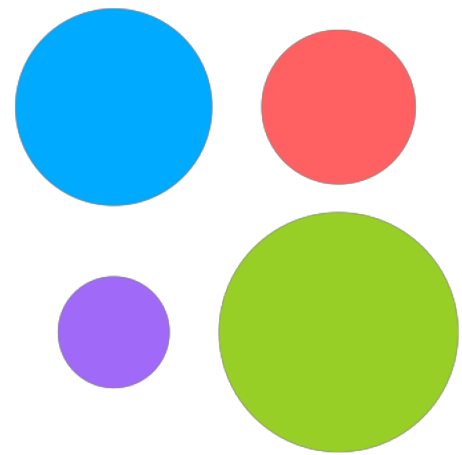
Что должно быть?

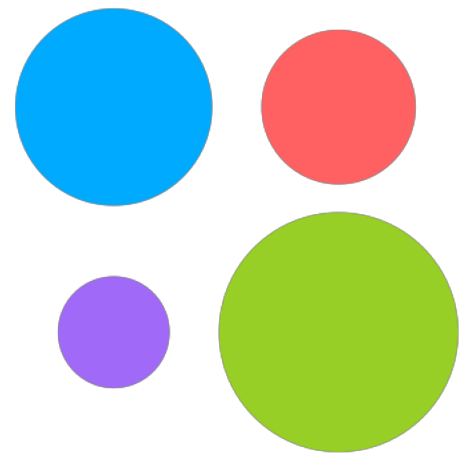
- Описание сервиса – несколько предложений текста, описывающих, что делает сервис
- Ссылка на диаграмму архитектуры
- Runbook
- FAQ
- Описание endpoint'ов API
- Labels – привязка к продукту, функциональности, структурному подразделению
- Владелец/владельцы – в большинстве своем определяем автоматически, но есть сложные случаи



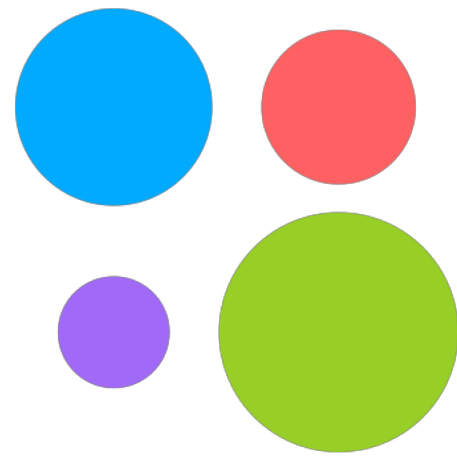
Что должно быть?

- Описание сервиса – несколько предложений текста, описывающих, что делает сервис
- Ссылка на диаграмму архитектуры
- Runbook
- FAQ
- Описание endpoint'ов API
- Labels – привязка к продукту, функциональности, структурному подразделению
- Владелец/владельцы – в большинстве своем определяем автоматически, но есть сложные случаи
- Ревью?

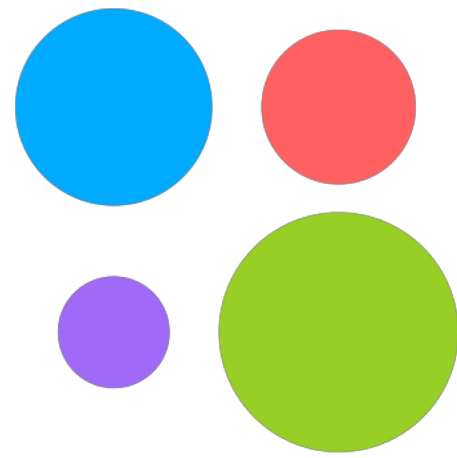




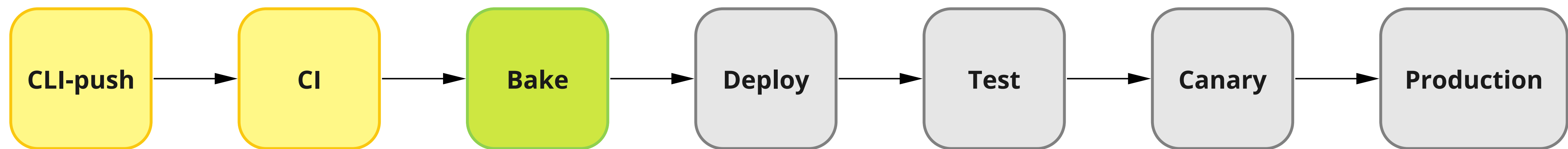
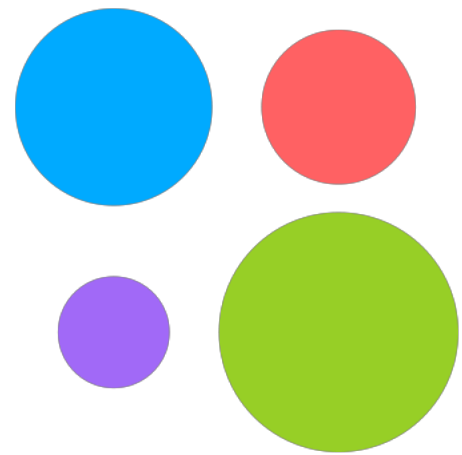
- Подготовка репозиториев
- Генерация пайплайна в Teamcity
- Выставление прав
- Поиск owner'ов сервиса по push'ам в репозиторий

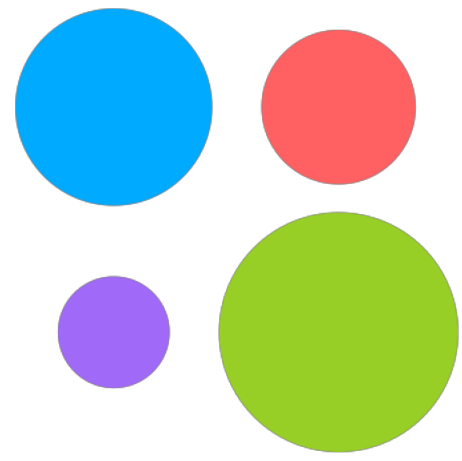


- Подготовка репозиториев
- Генерация пайплайна в Teamcity
- Выставление прав
- Поиск owner'ов сервиса по push'ам в репозиторий
 - Пытаемся найти 2-х овнеров
 - Считаем не только кол-во push'ей, но и кол-во кода в push'е



- Подготовка репозиториев
- Генерация пайплайна в Teamcity
- Выставление прав
- Поиск owner'ов сервиса по push'ам в репозиторий
 - Пытаемся найти 2-х овнеров
 - Считаем не только кол-во push'ей, но и кол-во кода в push'е
- Регистрируем сервис в Atlas – сервис, овнеров, зависимости, ...
- Проверяем миграции – если потенциально опасные – регистрируем триггер в Atlas и ставим в карантин

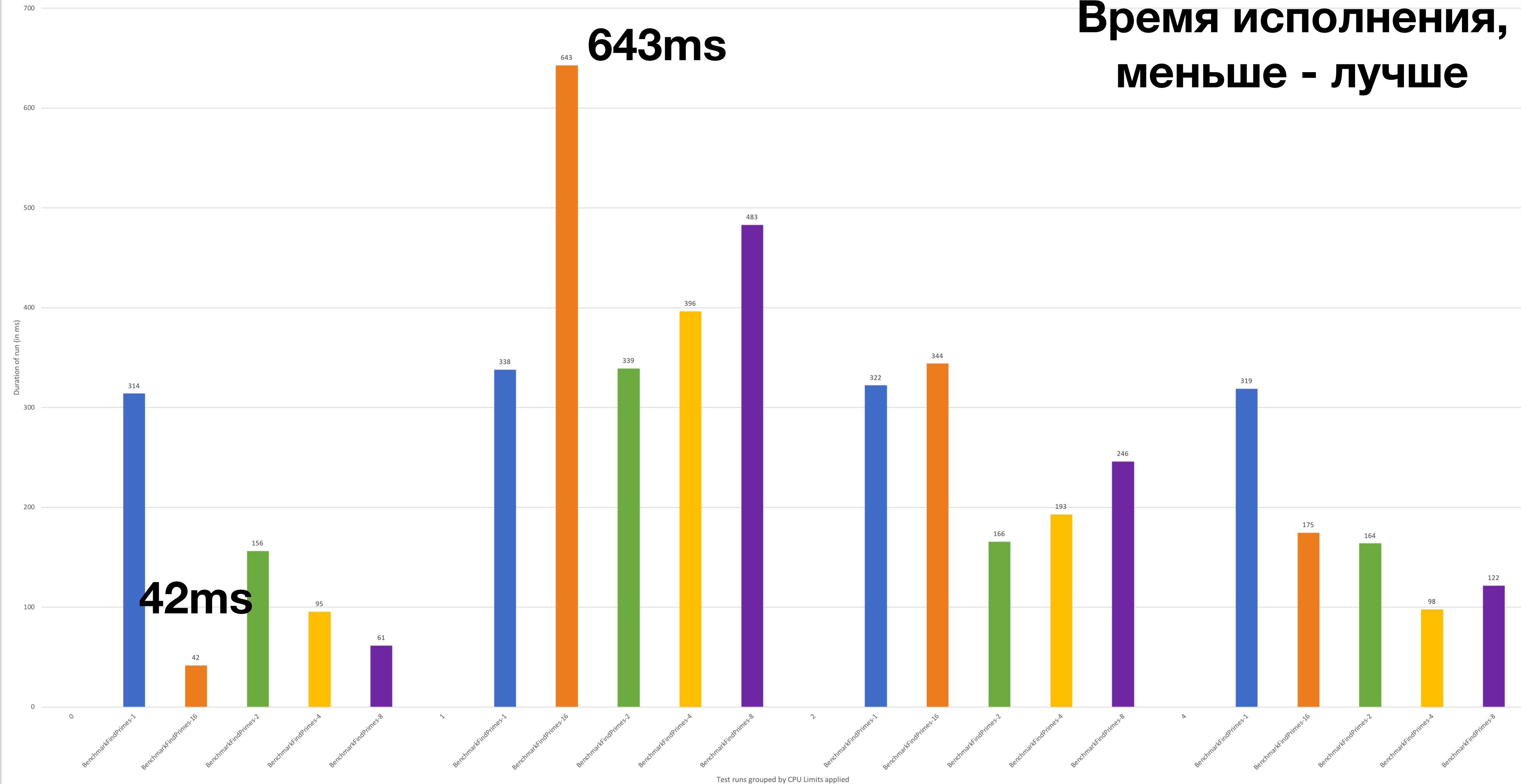




- Сборка приложения в docker image
- Генерация helm-чартов для самого сервиса и связанных ресурсов (БД, кэш, ...)
- Создание админских тикетов на открытие портов и т.п.
- Прогон unit-тестов, подсчет code-coverage
 - Кол-во тестов, покрытие регистрируются в Atlas
- Учет ограничений по CPU, RAM, ...

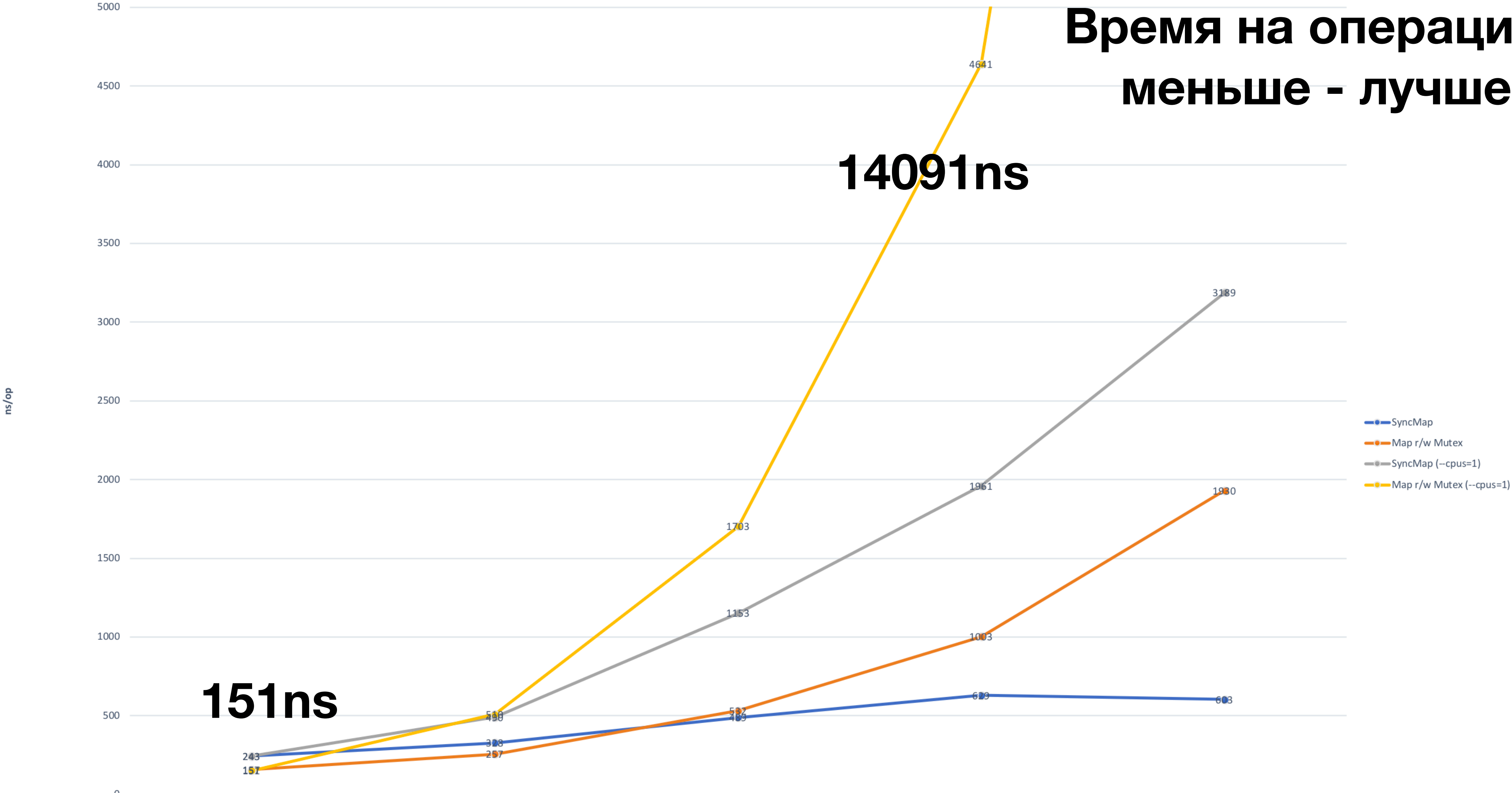
Golang + k8s + GOMAXPROCS

Время исполнения,
меньше - лучше



CPU CFS Quota and Mutexes

Время на операцию,
меньше - лучше



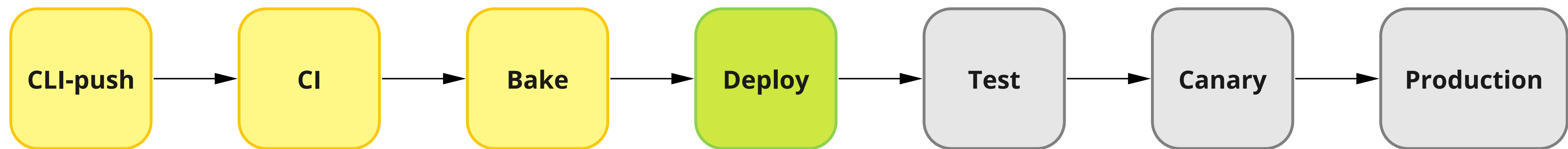
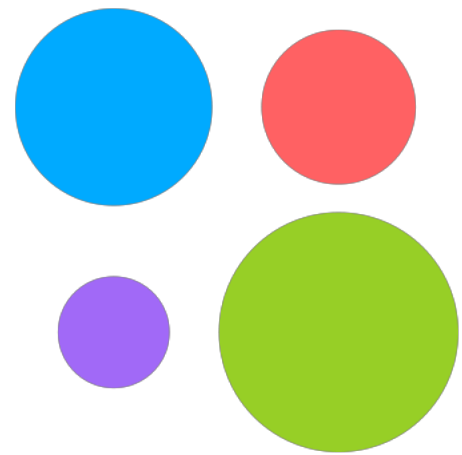
151ns

14091ns

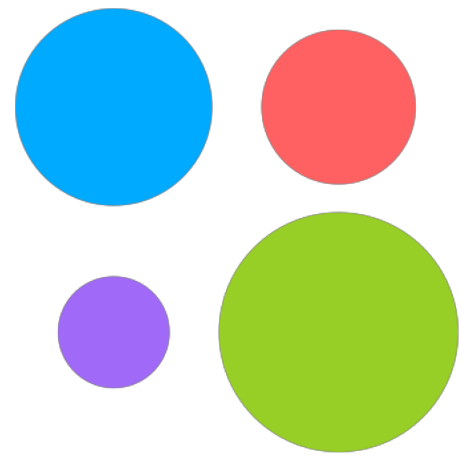
	1	2	4	8	16
SyncMap	243	328	489	629	603
Map r/w Mutex	157	257	532	1003	1930
SyncMap (--cpus=1)	243	490	1153	1961	3189
Map r/w Mutex (--cpus=1)	151	510	1703	4641	14079

CPU

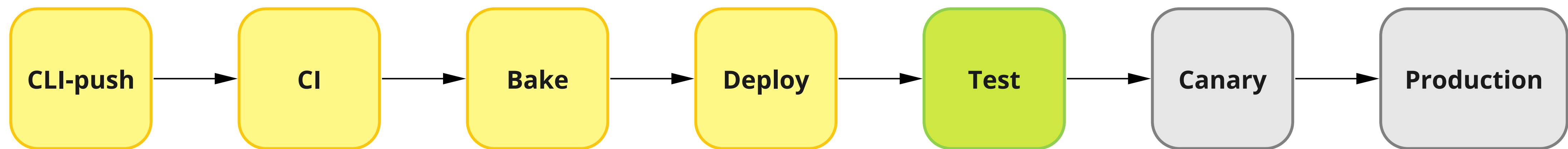
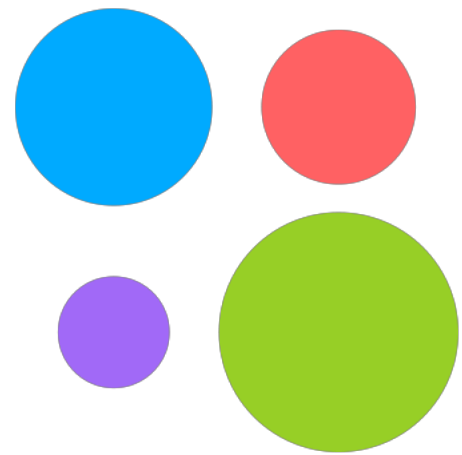
**[https://github.com/uber-go/
automaxprocs](https://github.com/uber-go/automaxprocs)**

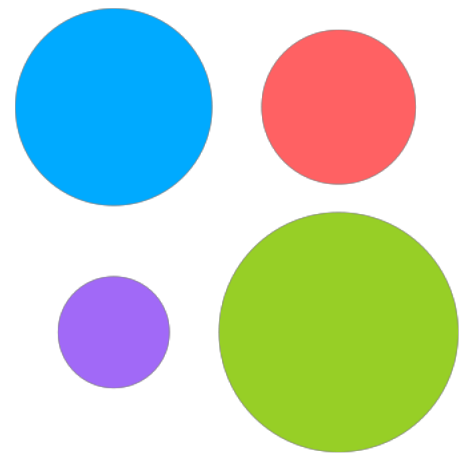


Проверка конвенций



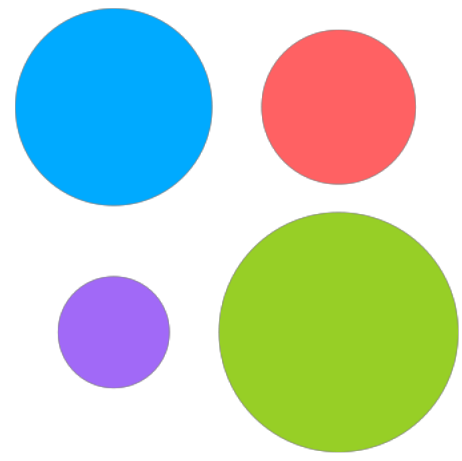
- Проверка сервисных endpoint'ов
- Проверка ответов сервисных endpoint'ов на соответствие схеме
- Проверка формата логов
- Проверка выставления заголовков при запросах
- Проверка выставления заголовков при отправке сообщений в шину (X-Source-ID) - отслеживание связности сервисов через шину



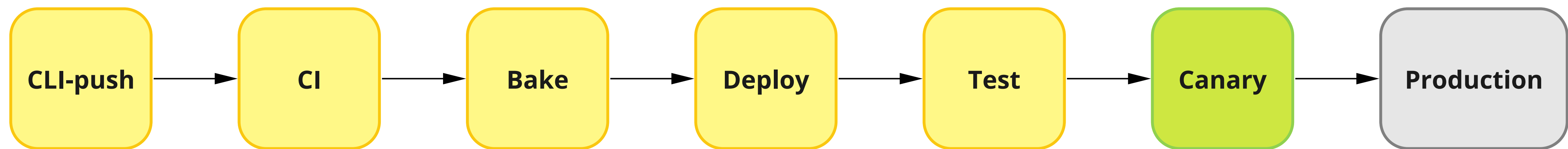
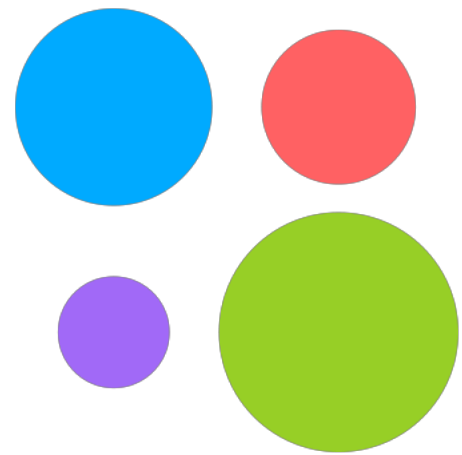


- Тестирование в закрытом контуре
 - Например: Hoverfly.io
- Нагрузочное тестирование

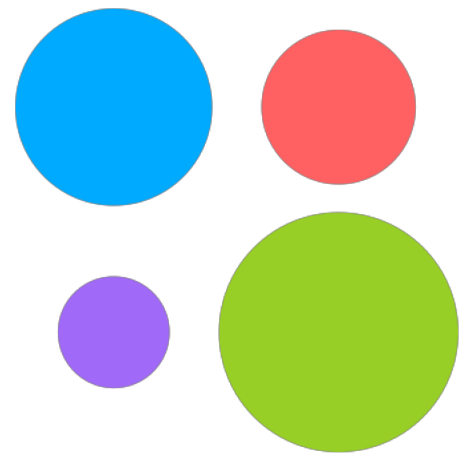
Нагрузочное тестирование



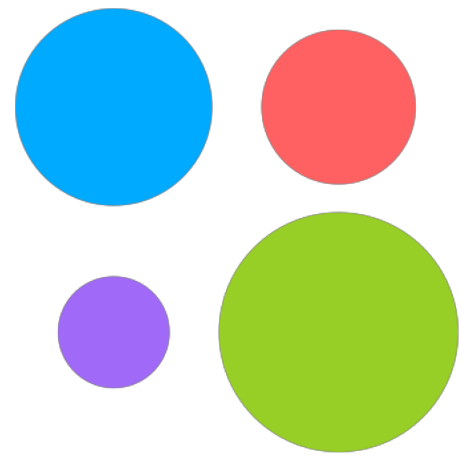
- Соответствие потребления выставленным ограничениям
 - Слишком мало – проверяем, не прибивается ли сервис OOM Killer
 - Слишком много – сервис не может адекватно утилизировать
- RPS – отсечка
 - Слишком мало – сервис супернеоптимальный или специфичный – требуется подтверждение выпуска в препрод/прод
 - Слишком много (выше разумного) – сервис пустой, потенциальная бага, что-то отвалилось уже на старте?



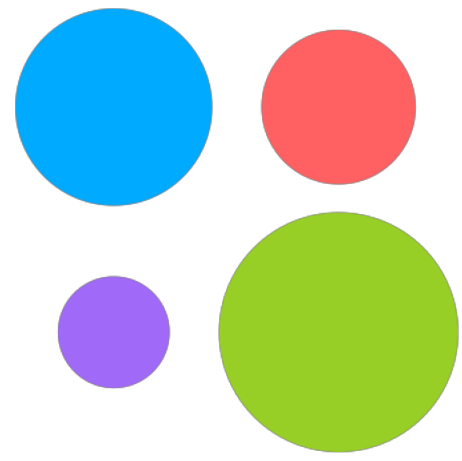
Запуск на малом количестве пользователей



- **Какой процент?**
- Начинаем с очень малого процента пользователей < 0.1

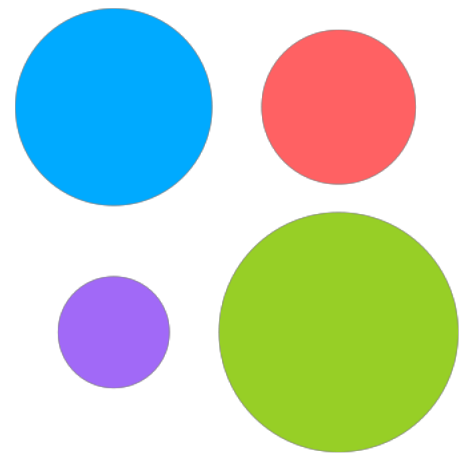


- **Какой процент?**
 - Начинаем с очень малого процента пользователей < 0.1
- **Сколько держим?**
 - Самая минимальная нагрузка – 5 мин
 - Основная – 2 часа
 - Для сложных сервисов выставляем в app.toml



Что смотрим?

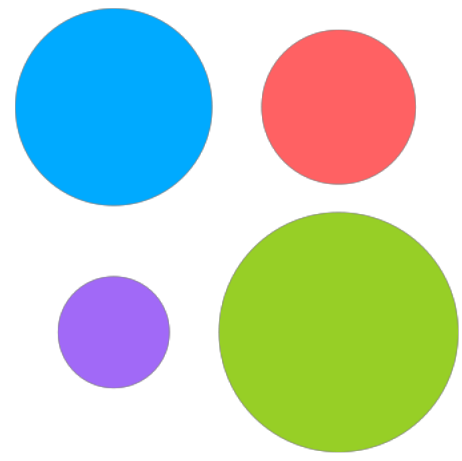
- Специфичные для языка метрики (воркеры для php-fpm)
- Ошибки в Sentry
- Статусы ответов
- Response Time (точное и среднее)
- Latency
- Исключения обработанные и не обработанные



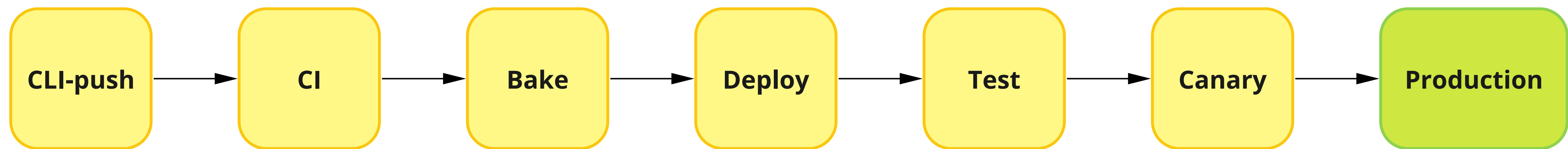
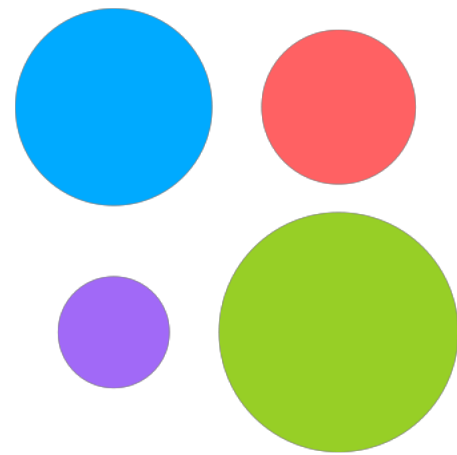
Что смотрим?

- Специфичные для языка метрики (воркеры для php-fpm)
- Ошибки в Sentry
- Статусы ответов
- Response Time (точное и среднее)
- Latency
- Исключения обработанные и не обработанные
- **Еще 100500 продуктовых метрик**

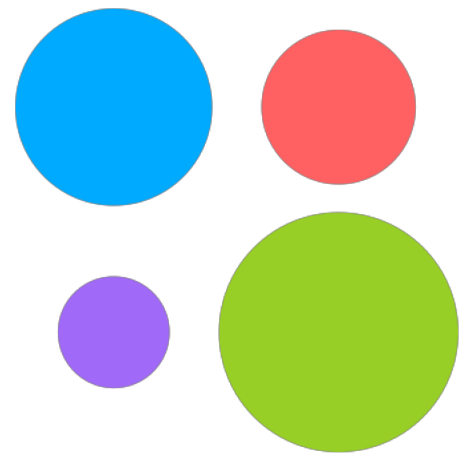
Squeeze testing



- Нагружаем один brick до точки отказа
- Добавлем +1, смотрим дельту
- Добавлем +1, ...
- ...
- Смотрим регрессию
- Обогащаем или заменяем данные из нагрузочного тестирования в Atlas



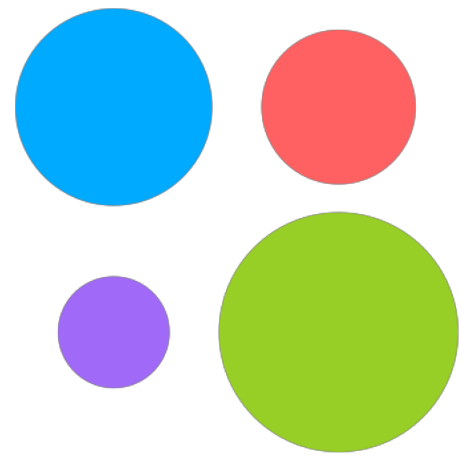
Масштабирование



По CPU

Только по CPU не очень эффективно

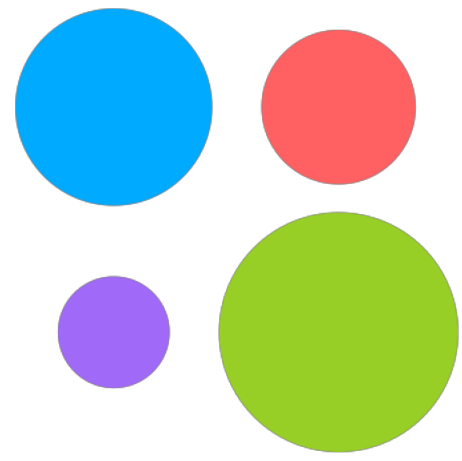
- прогрев
- шумные соседи
- cron-task с временной нагрузкой
- ...



App specific

По app-specific метрикам

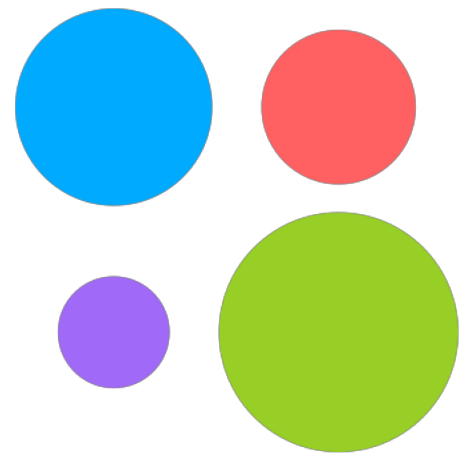
- кол-во запросов в очереди обслуживания
- время проведенное в очереди
- общая задержка ответа
- ...



Итоговая схема

- CPU + RAM
- Кол-во запросов
- Время ответа
- Прогноз по историческим данным

**Как быть с зависимостями
при масштабировании?**

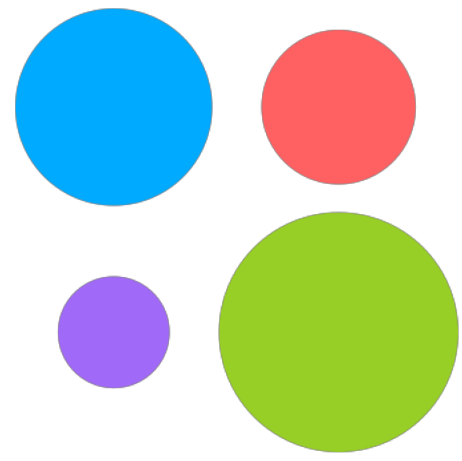


Каскад масштабирования

- +1 уровень
- На основе исторических данных относительно полученной комбинации CPU + Response Time + исторические данные инициализирующего сервиса

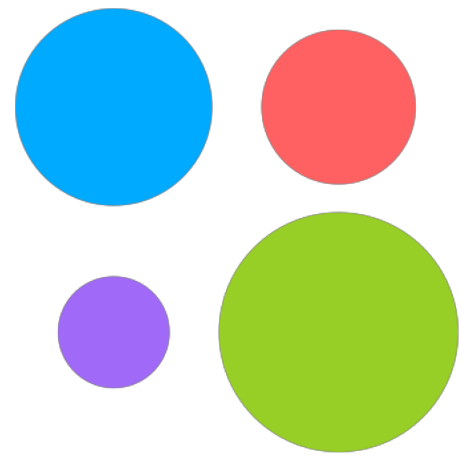
Обслуживание

Триггеры



- Миграции, если не осталось версий сервиса ниже X
- Security Updates (базовые образы, библиотеки)
- Давно не обновлялись
- Потребители очень старых версий сервиса, затрудняющие поддержку этого сервиса
- С очень низкой нагрузкой – раньше была, теперь – десяток запросов
- Карантин по невыполнению новых требований платформы

Дашборд



- Единая точка информации о сервисе - покрытие тестами, кол-во образов, подов, версий
- Фильтрация по сервису и лейблам
- Интеграция с трасировкой, логгированием, мониторингом
- Единая точка документации по сервисам
- Единая точка показа всех ивентов по сервисам

service-image-storage

staging ▼

334

Запросов в минуту



10738 ms

Время ответа



45.7%

ПОКРЫТИЕ ТЕСТАМИ



Последний деплой **beta**















Релизер: aasukhov

Время релиза: 2018-10-18 18:01:18

[Логи сборки](#)

Info: http://staging.k.avito.ru/service-image-storage/_info

События

Date	Type	Severity	Message
02.11.2018, 16:03:45	 CPU тротлинг image-storage-staging-rabbitr	warning	
02.11.2018, 16:03:16	 CPU тротлинг image-storage-staging-rabbitr	warning	
02.11.2018, 16:02:15	 CPU тротлинг image-storage-staging-rabbitr	warning	
02.11.2018, 16:01:45	 CPU тротлинг image-storage-staging-b45b4c	warning	
02.11.2018, 16:00:45	 CPU тротлинг image-storage-staging-rabbitr	warning	
02.11.2018, 15:59:45	 CPU тротлинг image-storage-staging-rabbitr	warning	
02.11.2018, 15:57:45	 CPU тротлинг image-storage-staging-rabbitr	warning	

События

[illegible]

Поды **beta**

Name	Status	Started	IP
image-storage-staging-b45b4db7f-tvlwk	Running	25.10.2018, 7:06:16	10.9.142.187
image-storage-staging-mongodb-6cd59d8585-h9lpw	Running	06.11.2018, 15:40:53	10.9.173.231
image-storage-staging-postgresql-7d555456cd-4hqr5	Running	17.10.2018, 12:16:54	10.9.157.234
image-storage-staging-rabbitmq-74b65cdb58-9hzhh	Running	24.10.2018, 12:30:55	10.9.169.19

Спасибо!

vamadison@avito.ru

vadim.madison@gmail.com