

Введение в МОНИТОРИНГ

Метрики

Системы мониторинга



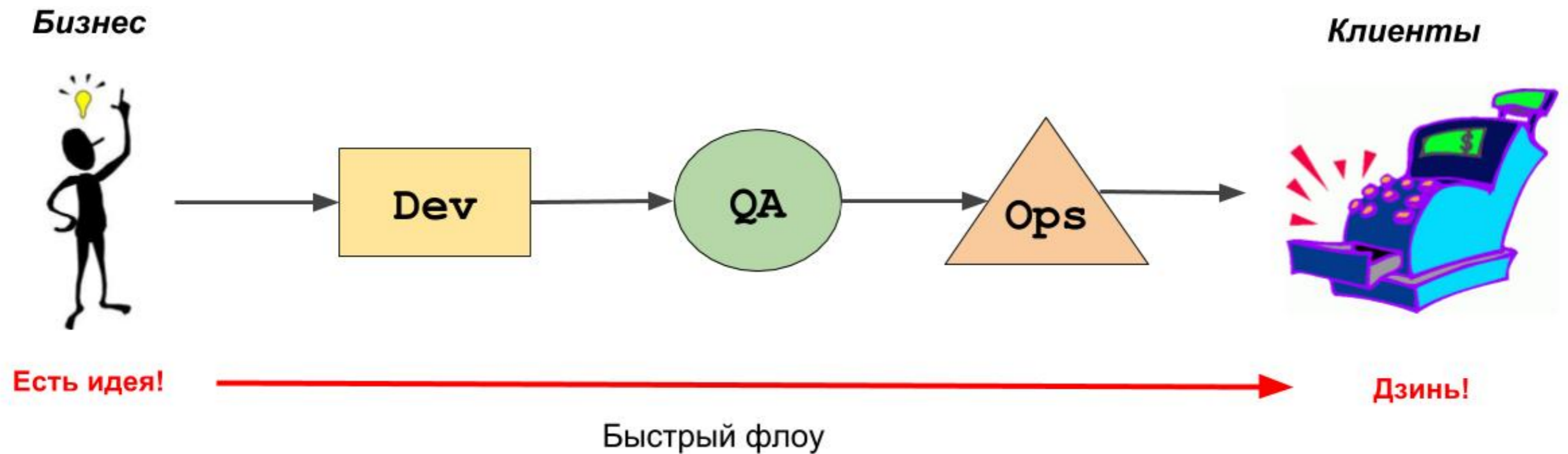
План

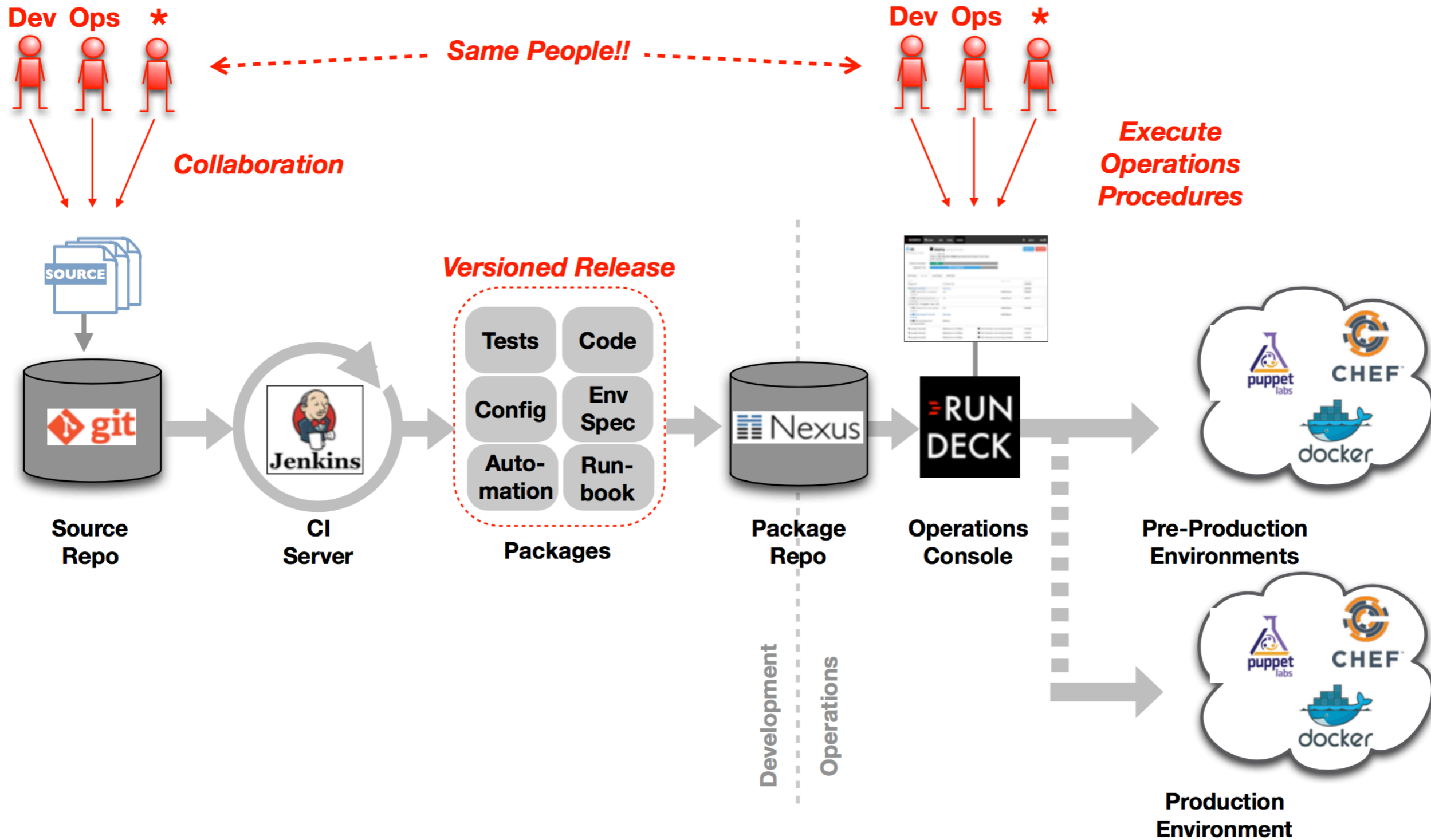
- Три пути DevOps
- Мониторинг
- Практики построения мониторинга и антипаттерны
- Модели мониторинга
- Метрики
- Системы мониторинга
- Prometheus

Три пути DevOps



Первый путь



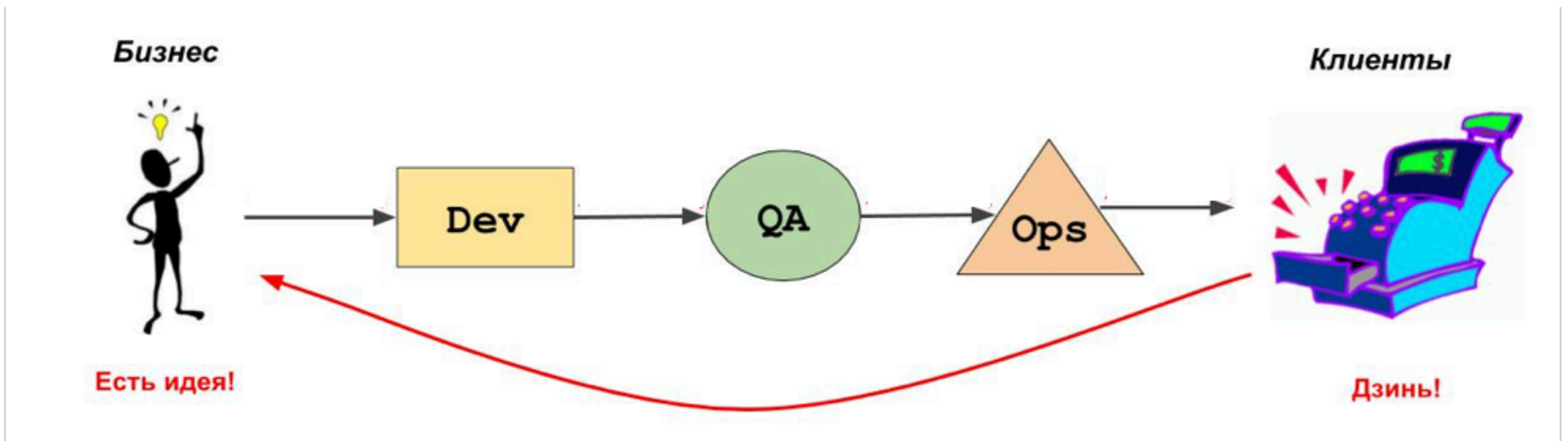


Что дальше?

- ...
- ...
- ...

Второй путь

Обеспечить быструю обратную связь справа налево



Плюсы быстрой обратной связи

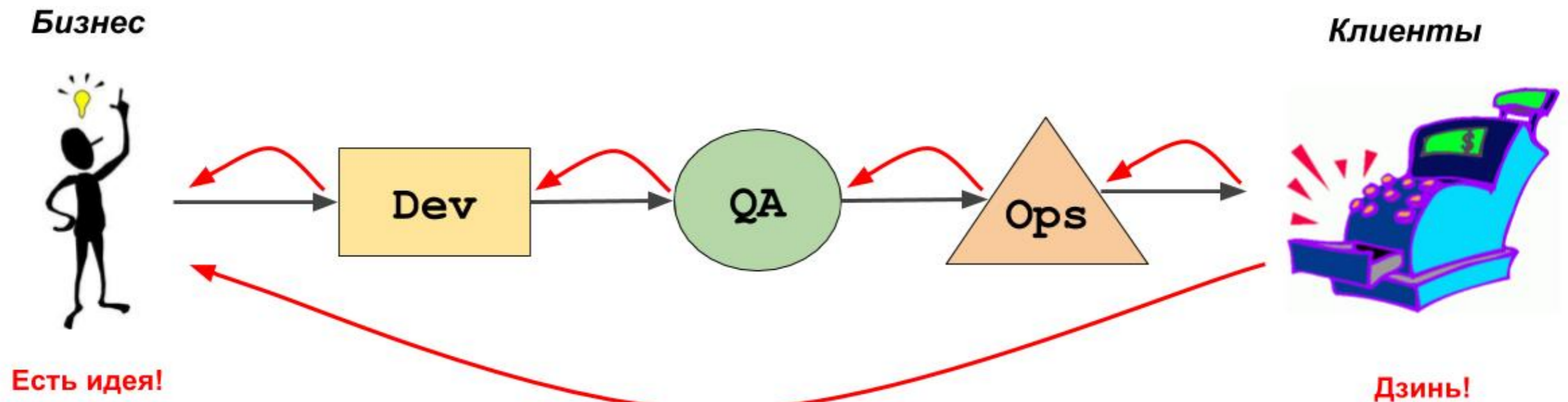
- Повышает качество продукта
- Позволяет оперативно реагировать на значимые события/изменения
- Повышается скорость обучения внутри организации

Примеры быстрой обратной связи

- Мониторинг окружений и работы приложения
- Мониторинг поведения пользователей
- Автоматизация сборки кода и прогона тестов
- ChatOps

Третий путь

- Создание культуры постоянного эксперимента и обучения внутри организации



МОНИТОРИНГ



Зачем?



Надежность имеет значение

Никто не будет пользоваться продуктом, если он:

- не работает
- работает медленно
- работает с ошибками

Надежность работы становится залогом его успеха.

Анализ поведения пользователей

Без анализа поведения пользователей:

- нельзя понять насколько успешна бизнес идея
- нет понимания потребностей пользователей

На рынке выигрывает тот, кто умеет понимать потребности пользователей и реализовывать их в кратчайшие сроки.

Нужен процесс, который позволит...

- Обеспечить контроль над работой наших сервисов
 - Отслеживание состояния и качество работы сервисов и инфраструктуры
 - Система оповещения о проблемах и потенциальных проблемах
- Обеспечить механизм оценки работы бизнес идей

МОНИТОРИНГ

- совокупность инструментов и практик, позволяющая осуществлять контроль над работой IT систем и давать оценку качества работы этих систем

ЧТО МОНИТОРИТЬ?

- Инфраструктура
- Работа и состояние поддерживаемых сервисов
- Процесс деплоя
- Работа и состояние приложения
- Бизнес метрики, поведение пользователей

Мониторинг как сервис

Может быть:

- собственный сервис (Zabbix, Prometheus, ...)
- SaaS (New Relic, Datadog, okmeter.io)
- outsource (ITSumma и многие другие)

Должен удовлетворять потребности своих клиентов:

- dev
- ops
- business
- ...

Хорошие и плохие практики



Антипаттерны в мониторинге

1. Одержимость выбором инструментов
2. "Специальный" отдел мониторинга
3. Мониторинг для галочки
4. Использование мониторинга в качестве "костыля"
5. Ручное конфигурирование

Хорошие практики построения мониторинга

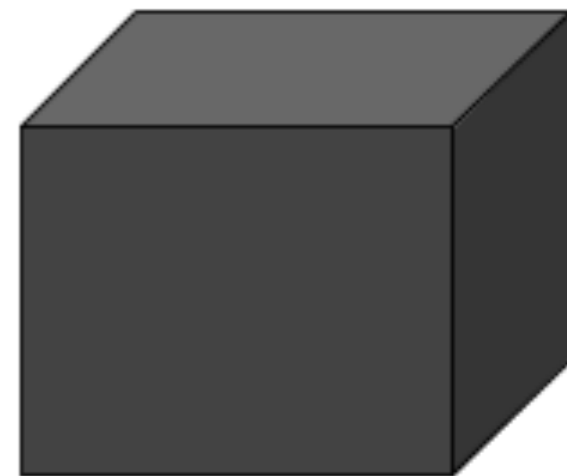
1. Мониторинг из многих компонентов как правило лучше, чем один монолит
2. Мониторинг это не только про оповещения
3. Построение мониторинга необходимо начинать с пользовательских метрик
4. Если есть возможность, начинать лучше с мониторинга как сервис (Datadog, New Relic, okmeter.io), чем разворачивать свою инфраструктуру
5. Постоянное улучшение всех компонентов

Модели МОНИТОРИНГА



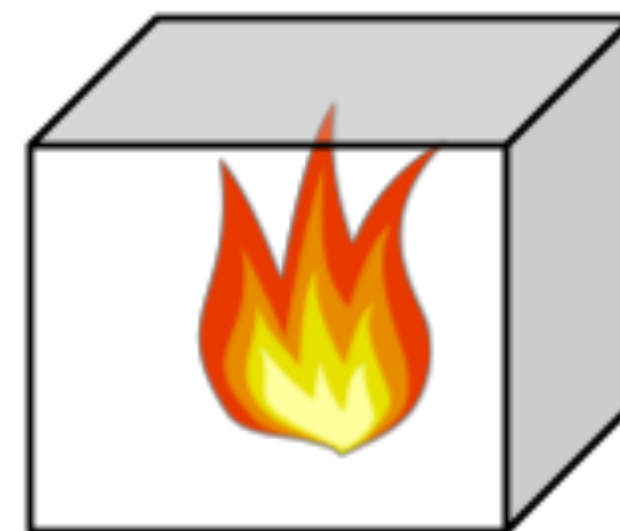
Blackbox МОНИТОРИНГ

- Мониторинг извне с точки зрения пользователя
- Не видим, как работает система внутри
- Примеры: проверка открытых портов, подсчет коннектов, наличие процесса

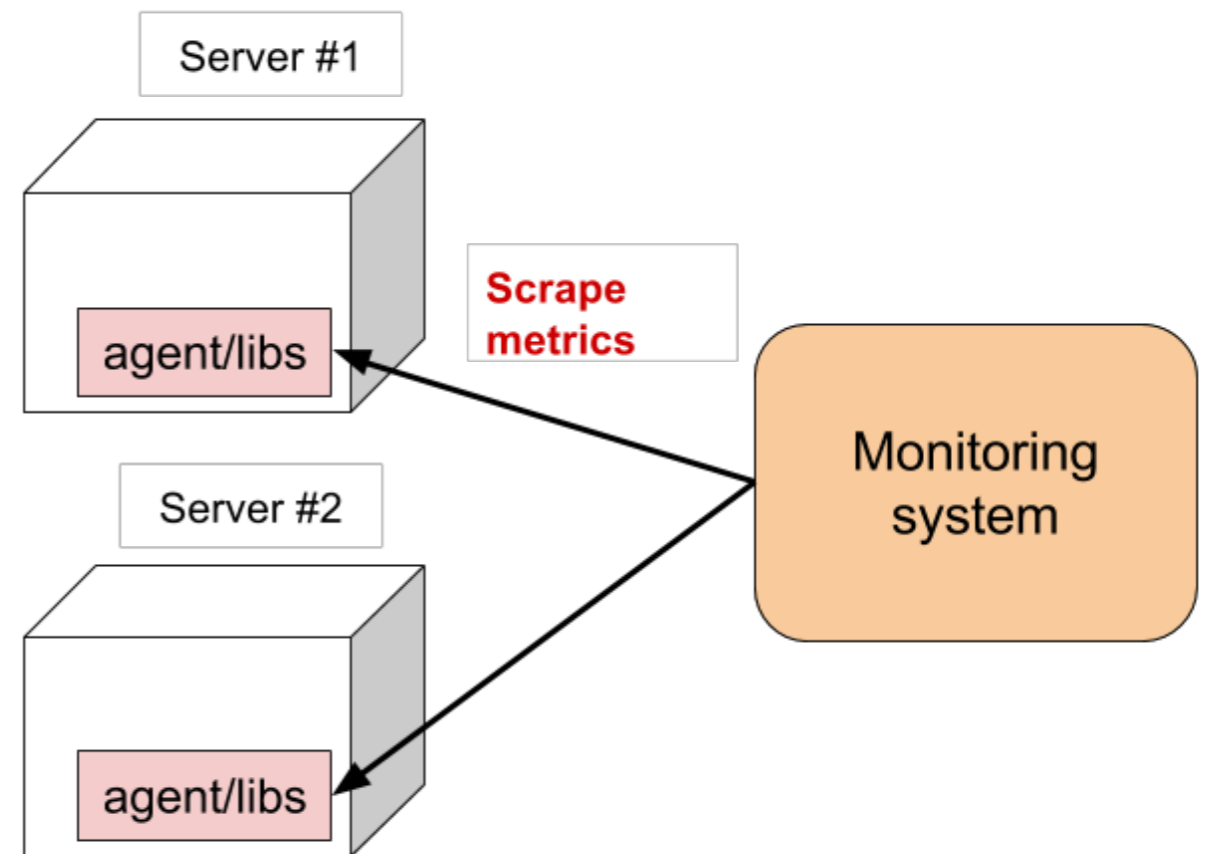
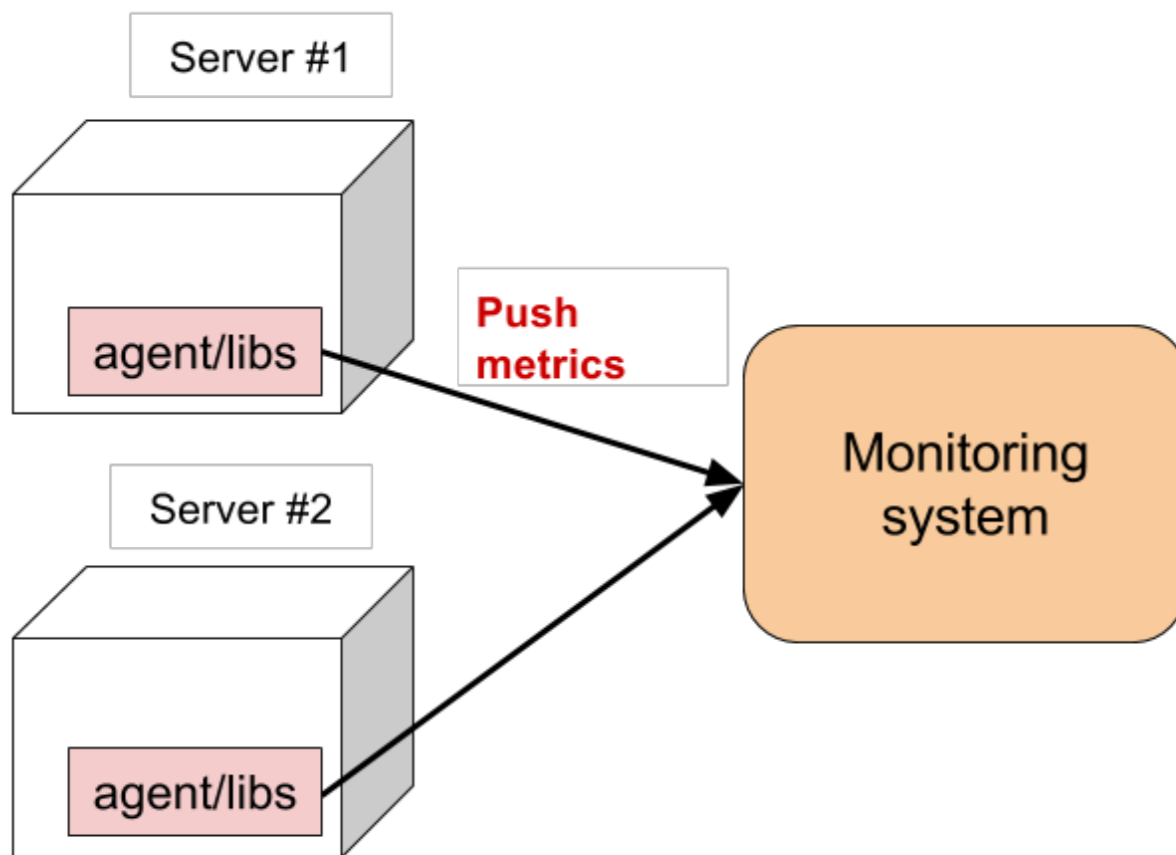


Whitebox мониторинг

- Дополняет модель blackbox
- Мониторинг на основе информации о внутренней работе системы
- Примеры: метрики приложений (время запроса к БД, количество пользователей и т.д.)



Push vs Pull



Метрики

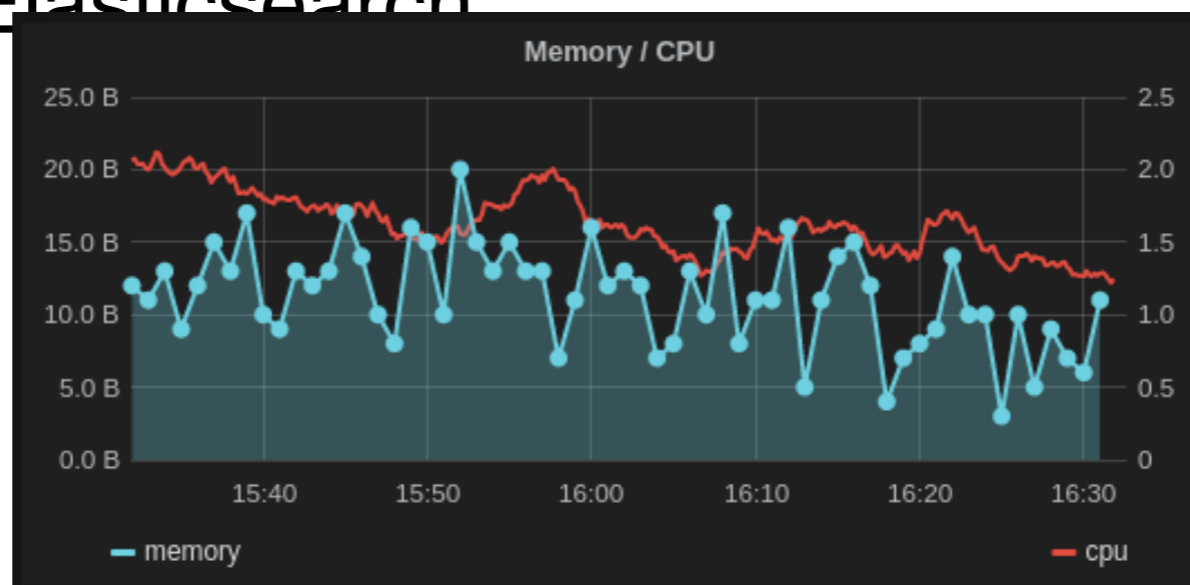


Метрика

- Описывает свойство наблюдаемой системы в определенный момент времени
- Представляет собой пару "ключ = значение"
- Пример: `node_load1 = 0.74`
- Собираются системой мониторинга с определенным временным интервалом
- К метрике может добавляться доп. информация. Например метки (labels) в Prometheus. `instance="node-exporter:9100",job="node"`

Временной ряд (time-series)

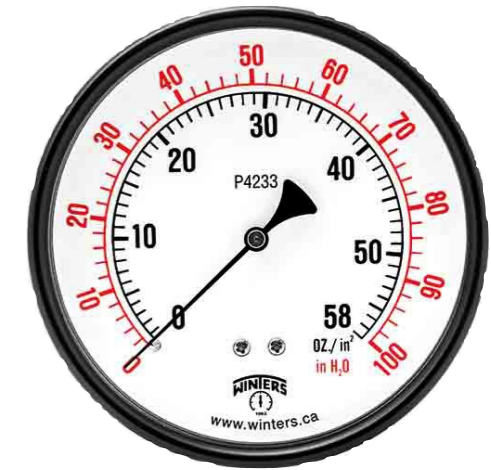
- Набор значений метрик(и) за определенный период времени - это временной ряд
- Для хранения временных рядов применяются TSDB (time series databases).
- Примеры TSDBs: InfluxDB, Whisper(Graphite), Prometheus, Elasticsearch



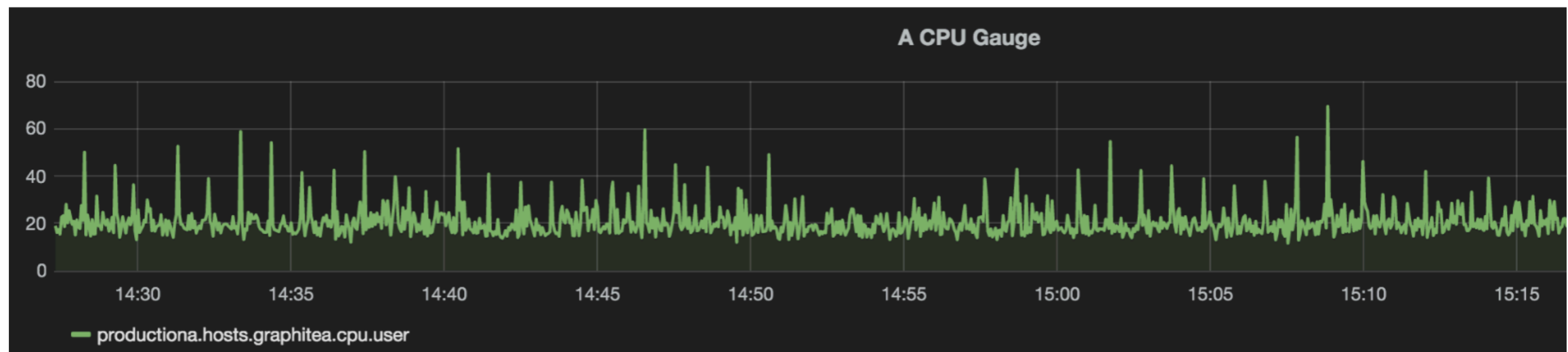
ОСНОВНЫЕ ТИПЫ МЕТРИК

- Gauge (шкала)
- Counter (счетчик)
- Histogram/Timer/Summary (гистограмма)

Gauge



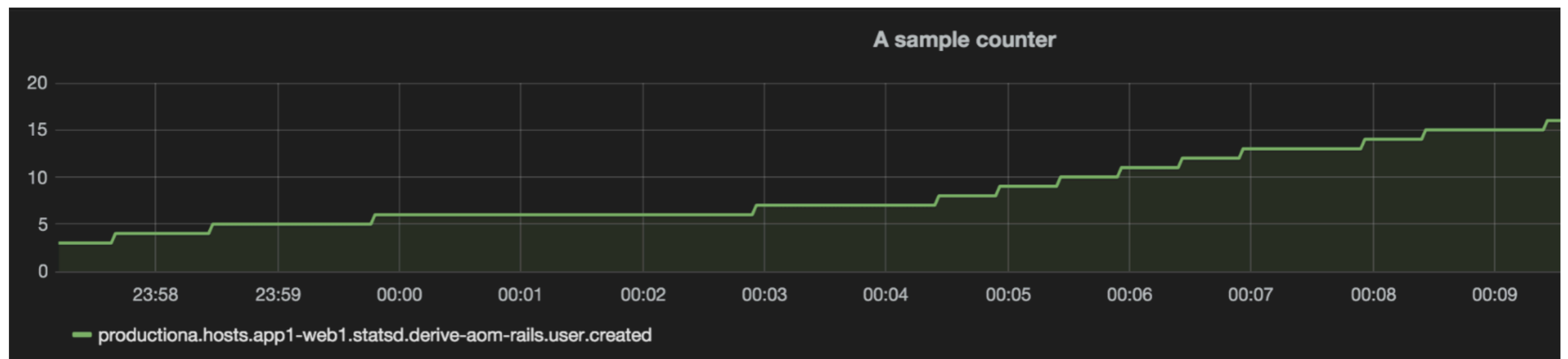
- Численное значение, которое может непредсказуемо меняться со временем
- Может увеличиваться и уменьшаться
- Примеры: использование CPU, memory, disk; количество пользователей онлайн



Counter



- Счетчик - численное значение, которое может расти со временем
- Может только увеличиваться, но возможен сброс на ноль
- Примеры: uptime, количество HTTP запросов, количество регистраций пользователей, число продаж

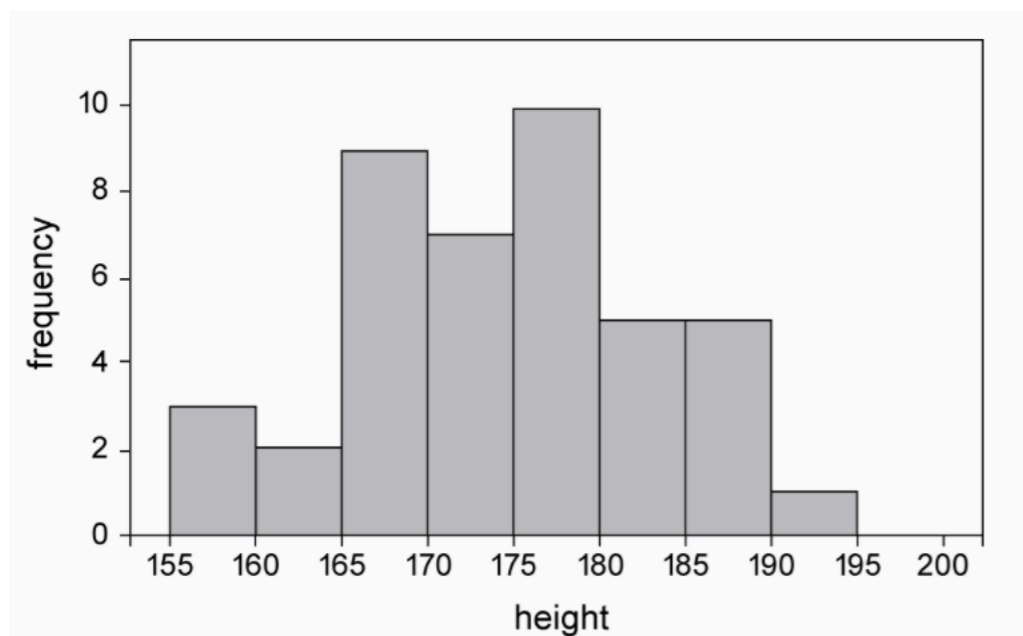


Histogram/Timer/ Summary



wind306

- Позволяют отследить длительность операции
- Примеры: время ответа веб сервера, длительность запроса к БД

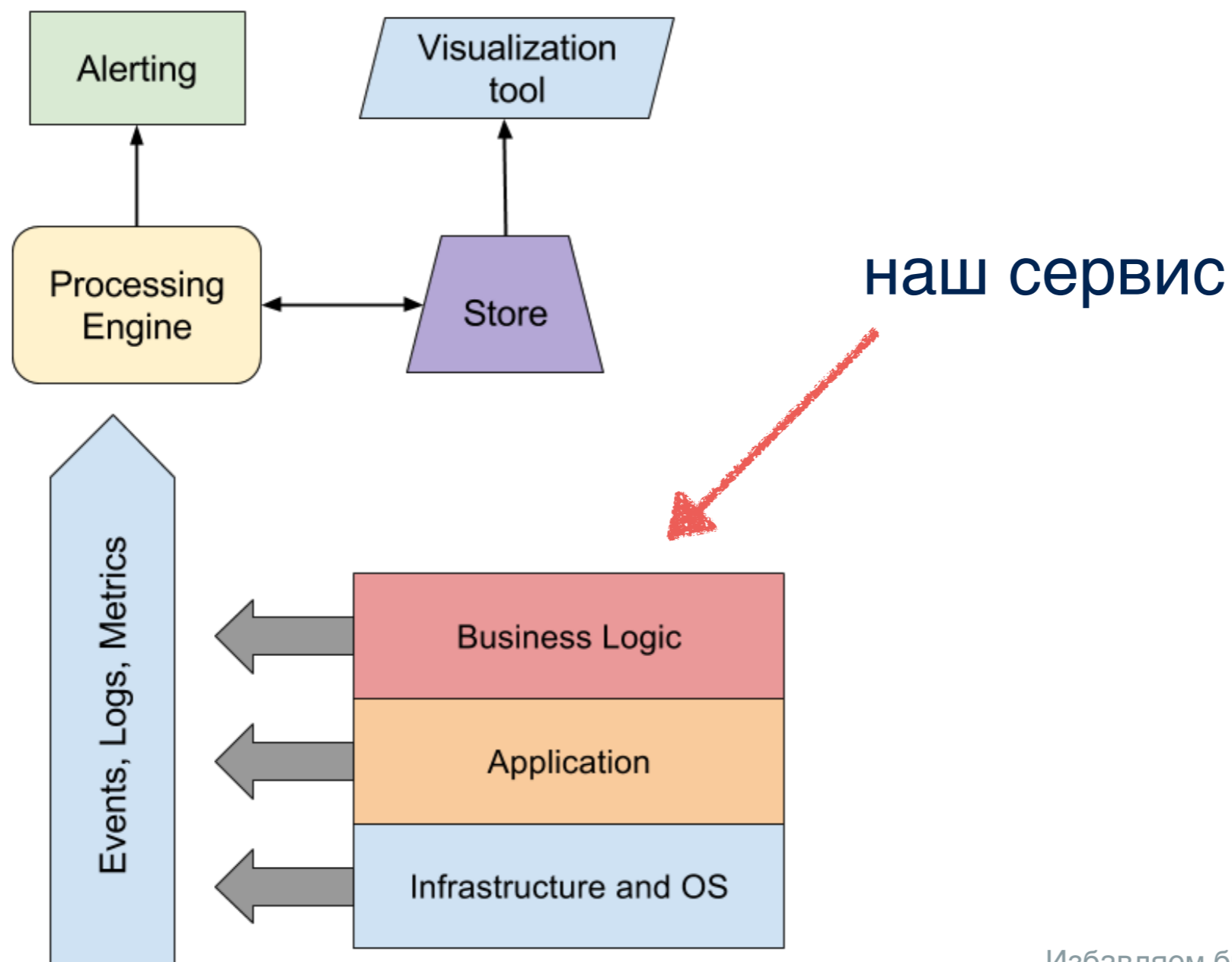


Element	Value
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="+Inf",path="/"}	3
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="0.005",path="/"}	0
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="0.01",path="/"}	0
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="0.025",path="/"}	1
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="0.05",path="/"}	2
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="0.1",path="/"}	3
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="0.25",path="/"}	3
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="0.5",path="/"}	3
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="1",path="/"}	3
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="10",path="/"}	3
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="2.5",path="/"}	3
ui_request_latency_seconds_bucket{instance="ui:9292",job="ui",le="5",path="/"}	3

СИСТЕМЫ МОНИТОРИНГА



Архитектура



ОСНОВНЫЕ КОМПОНЕНТЫ

- Средства сбора и отправки данных: агенты, API libs
- Хранилище данных
- Инструмент обработки и аналитики данных
- Инструмент визуализации
- Система оповещения

Примеры систем мониторинга

- Nagios/Icinga
- Zabbix
- TICK (Telegraf, InfluxDB, Chronograf, Kapacitor)
- Graphite
- Prometheus
- Sensu
- Riemann
-



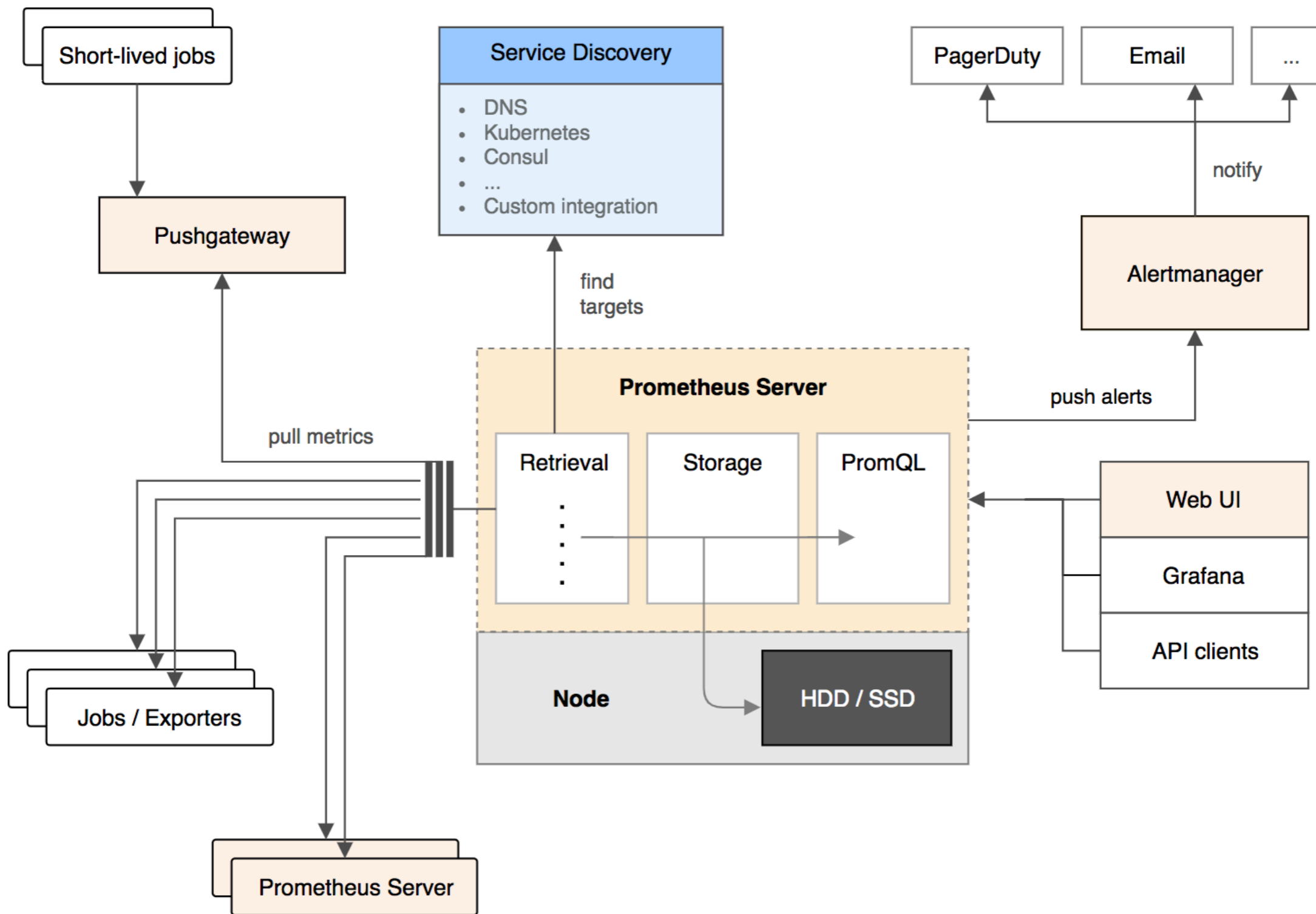
Prometheus



Prometheus



- Развитие проекта началось в 2012
- Создан на основе Borgmon (система мониторинга в Google), бывшими работниками Google
- Open source
- Написан на Go
- Whitebox, Pull система



Jobs & targets

- Targets (endpoint) - источник для сбора метрик
- Группы источников объединяют в jobs

Exporter

- Программа, которая делает метрики доступными для сбора Prometheus
- Дает возможность конвертировать метрики в нужный для Prometheus формат
- Используется когда нельзя поменять код приложения
- Примеры: PostgreSQL, RabbitMQ, Nginx, Node exporter, cAdvisor

Конфигурация

prometheus.yml

global:

scrape_interval: "5s"

scrape_configs:

- job_name: "prometheus"

static_configs:

- targets:

- "localhost:9090"

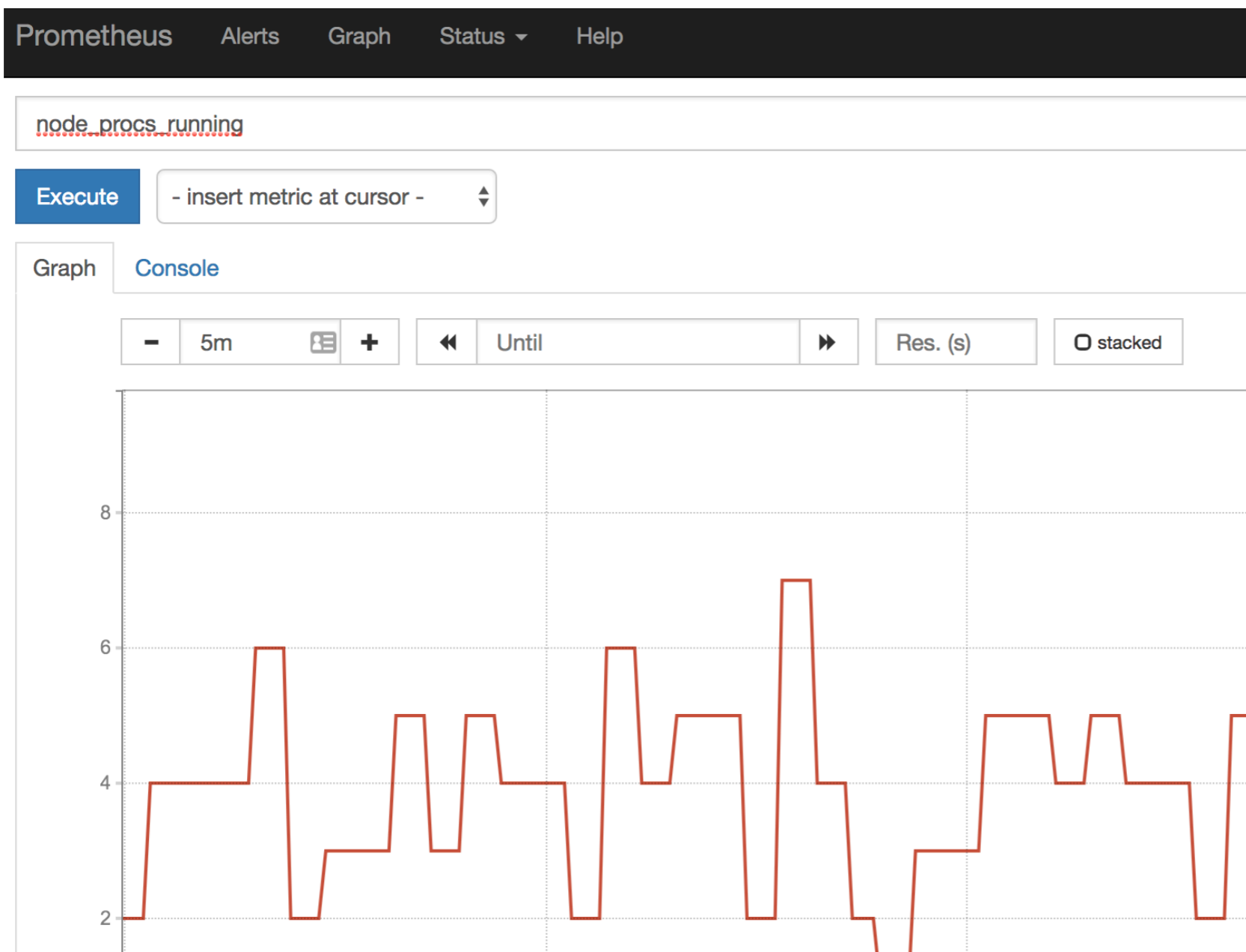
- job_name: "reddit"

static_configs:

- targets:

- "172.18.0.12:9292"

Пользовательский интерфейс



Полезные ссылки

- Три пути DevOps
- книги Practical monitoring, The Art of Monitoring
- Prometheus
- Перевод статьи про Prometheus, еще один
- Глава про мониторинг из книги Site Reliability Engineering
- Тематические чаты: Церковь метрик, Zabbix
- Видео с митапа про мониторинг. Слайды

To be continued....

Вопросы ?

