Мониторинг Распределённых Систем

Практический опыт

О спикере

Алексей Иванов
Dropbox
Traffic Team

О спикере

Алексей Иванов

Dropbox

Traffic Team

О спикере

Алексей Иванов

Dropbox

Traffic Team

О спикере

Алексей Иванов
Dropbox
Traffic Team

О спикере

Алексей Иванов
Dropbox
Traffic Team

Disclaimer*

*Подробно о мониторинге и алёртинге можно почитать в Google SRE Book

Legal Disclaimer*

*Моё мнение не отражает мнение Dropbox

Фокус на выборе и тюнинге систем мониторинга*

*Недостаточно внимания уделяется удобству использования

TL;DR

Не важно какая у вас система мониторинга,
важно какие интерфейсы она
предоставляет и как они используются

UX важен*

*UX — это не морда Grafana’ы

Пользователи систем мониторинга — Инженеры и их Роботы

UX систем мониторинга — API, Библиотеки и Framework’и*

*Лёгкий способ для ввода и вывода данных в/из системы мониторинга

UX это интеграции*

*Deployment, Service Discovery, Logging, Exception Reporting, etc…

UX это UI*

*Важно, но не критично

Цель мониторинга:
уменьшить MTTR при факапе*

*Mean-Time-To-Recover

Жизненный цикл Факапа

Start
Detection
Triage
Workaround
Permanent Fix

Start

Detection

Triage

Workaround

Permanent Fix

Start

Detection

Triage

Workaround

Permanent Fix

Start
Detection
Triage
Workaround
Permanent Fix

Start
Detection
Triage
Workaround
Permanent Fix

Start
Detection
Triage
Workaround
Permanent Fix

Start
Detection
Triage
Workaround
Permanent Fix

Start

Detection

Triage

Workaround

Permanent Fix

Метрики*

*Должны быть у каждого сервиса и библиотеки в production’е

Метрики должны быть частью базового Framework’а

*Экспорт статистики через, например, /metrics с любого (микро-)сервиса

Добавить метрику должно быть максимально просто*

*В любом из поддерживаемых языков

Все внутренние библиотеки должны быть обмазаны метриками*

*Особенно клиентские библиотеки e.g.: SQL, memcache, http, etc.

RPC или Proxy Mesh*

*Оба подхода позволяют инструментировать код единожды

Annotation и Attribution*

*Всё от HTTP до SQL запросов может быть аннотировано

Exception’ы и Traceback’и*

*Сбор и агрегация для всех поддерживаемых языков

Агрегация структурированных логов (событий)*

*Используется для долгосрочного хранения, аудита и аналитики

Пайплайн для неструктурированных логов*

*В идеале не хранить их на диске, а слать сразу в сеть

Introspection APIs*

Внутренний стейт сервиса по /debug/ эндпоинтам

Start
Detection
Triage
Workaround
Permanent Fix

Start

Detection

Triage

Workaround

Permanent Fix

Алёрты*

*Каждый факап должен приводить к page’у

Задача алёрта:
уменьшить MTTD*

*Mean-Time-To-Detect

Тип алёрта:
Web или Page*

*Каждый раз когда вы создаёте Email алёрт в мире умирает котёнок

Гигиена Web-алёртов: TMoA*

*Too-Many-Old-Alerts: page который приходит oncall’у в случае большого кол-ва “долговесящих” web-алёртов

Алёрты на основе SLI и SLA*

*Подробнее в SRE Book от Google и “Reliability When Everything Is a Platform: Why You Need to SRE Your Customers”, SRECon 2017, by Dave Rensin

Симптомы
vs
Причины*

*page’ы должны быть на симптомы, подробнее в “My Philosophy on Alerting”, by Rob Ewaschuk

Внешние End-To-End тесты*

*Blackbox тестирование всех (микро-)сервисов

Интеграция системы алёртинга с остальным сервисами в production*

*Ещё раз о том, что UX важен

Автоматическое создание алёртов для сервиса на основе:

Конфигов deployment системы
Тегов сервиса в timeseries базе данных
Конфигов самого сервиса

Автоматическое создание алёртов
для сервиса на основе:

Конфигов deployment системы
Тегов сервиса в timeseries базе данных
Конфигов самого сервиса

Автоматическое создание алёртов
для сервиса на основе:

Конфигов deployment системы
Тегов сервиса в timeseries базе данных
Конфигов самого сервиса

Автоматическое создание алёртов
для сервиса на основе:

Конфигов deployment системы
Тегов сервиса в timeseries базе данных
Конфигов самого сервиса

Автоматическое создание алёртов для сервиса на основе:

Конфигов deployment системы
Тегов сервиса в timeseries базе данных
Конфигов самого сервиса

Создание алёртов для сервиса через API:

Позволяет хранить алёрты в репозитории
Генерировать их скриптом из yaml’а
Интегрировать с SD/Deploy/SCM/1C/etc

Создание алёртов для сервиса
через API:

Позволяет хранить алёрты в репозитории
Генерировать их скриптом из yaml’а
Интегрировать с SD/Deploy/SCM/1C/etc

Создание алёртов для сервиса
через API:

Позволяет хранить алёрты в репозитории
Генерировать их скриптом из yaml’а
Интегрировать с SD/Deploy/SCM/1C/etc

Создание алёртов для сервиса
через API:

Позволяет хранить алёрты в репозитории
Генерировать их скриптом из yaml’а
Интегрировать с SD/Deploy/SCM/1C/etc

Создание алёртов для сервиса через API:

Позволяет хранить алёрты в репозитории
Генерировать их скриптом из yaml’а
Интегрировать с SD/Deploy/SCM/1C/etc

Start
Detection
Triage
Workaround
Permanent Fix

Start
Detection
Triage
Workaround
Permanent Fix

Что в себя включает алёрт?

Имя алёрта:

nginx_5xx
nginx_api-fe_sfo_5xx_high_critical

Имя алёрта:

nginx_5xx
nginx_api-fe_sfo_5xx_high_critical

Имя алёрта:

nginx_5xx
nginx_api-fe_sfo_5xx_high_critical

Имя алёрта:

nginx_5xx
nginx_api-fe_sfo_5xx_high_critical

Имя алёрта:

nginx_5xx
nginx_api-fe_sfo_5xx_high_critical

Имя алёрта:

nginx_5xx
nginx_api-fe_sfo_5xx_high_critical

Имя алёрта:

nginx_5xx
nginx_api-fe_sfo_5xx_high_critical

Имя алёрта:

nginx_5xx

nginx_api-fe_sfo_5xx_high_critical

Имя алёрта:

nginx_5xx
nginx_api-fe_sfo_5xx_high_critical

Исторические данные:

График метрики за последние N часов
Кто, когда и как менял алёрт
История Downtime’ов с комментариями

Исторические данные:

График метрики за последние N часов

Кто, когда и как менял алёрт

История Downtime’ов с комментариями

Исторические данные:

График метрики за последние N часов

Кто, когда и как менял алёрт

История Downtime’ов с комментариями

Исторические данные:

График метрики за последние N часов
Кто, когда и как менял алёрт
История Downtime’ов с комментариями

Исторические данные:

График метрики за последние N часов
Кто, когда и как менял алёрт
История Downtime’ов с комментариями

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)
Ссылки на Dashboard’ы и Log View’er
Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)

Ссылки на Dashboard’ы и Log View’er

Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)
Ссылки на Dashboard’ы и Log View’er
Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)

Ссылки на Dashboard’ы и Log View’er

Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)

Ссылки на Dashboard’ы и Log View’er

Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)

Ссылки на Dashboard’ы и Log View’er

Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)

Ссылки на Dashboard’ы и Log View’er

Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)

Ссылки на Dashboard’ы и Log View’er

Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)
Ссылки на Dashboard’ы и Log View’er
Shell команды

Playbook встроен в алёрт

Описание сервиса и метрики (md/rst)
Ссылки на Dashboard’ы и Log View’er
Shell команды

Start
Detection
Triage
Workaround
Permanent Fix

Start
Detection
Triage
Workaround
Permanent Fix

Autoremediation*

*Роботы чинящие продакшн, подробнее в:
“Bridging the Safety Gap from Scripts to Full Auto-Remediation”, SRECon’16 Europe, by David Mah

Использование ремедиаций по назначению:

Автоматизация рутины и хаков
Перезапуск серверов
Замена железа

Использование ремедиаций по назначению:

Автоматизация рутины и хаков

Перезапуск серверов

Замена железа

Использование ремедиаций по назначению:

Автоматизация рутины и хаков

Перезапуск серверов

Замена железа

Использование ремедиаций по назначению:

Автоматизация рутины и хаков

Перезапуск серверов

Замена железа

Использование ремедиаций по назначению:

Автоматизация рутины и хаков

Перезапуск серверов

Замена железа

Использование ремедиаций по назначению:

Автоматизация рутины и хаков
Перезапуск серверов
Замена железа

Использование ремедиаций по назначению:

Автоматизация рутины и хаков
Перезапуск серверов
Замена железа

Start
Detection
Triage
Workaround
Permanent Fix

Start
Detection
Triage
Workaround
Permanent Fix

Злоупотребление autoremediation’ами:

Масштабирование сервисов
Миграции между датацентрами
Апгрейды ядер

Злоупотребление
autoremediation’ами:

Масштабирование сервисов
Миграции между датацентрами
Апгрейды ядер

Злоупотребление
autoremediation’ами:

Масштабирование сервисов
Миграции между датацентрами
Апгрейды ядер

Злоупотребление
autoremediation’ами:

Масштабирование сервисов
Миграции между датацентрами
Апгрейды ядер

Злоупотребление autoremediation’ами:

Масштабирование сервисов
Миграции между датацентрами
Апгрейды ядер

Start
Detection
Triage
Workaround
Permanent Fix

Start
Detection
Triage
Workaround
Permanent Fix

Дашборды*

*У каждого сервиса должен быть как минимум один дашборд

Story time*

*Troubleshooting интервью на SRE в Dropbox

Хороший troubleshooting:

Опирается на SLO/SLI/SLA
Методичен и Иерархичен
Использует графики и дашборды

Хороший troubleshooting:

Опирается на SLO/SLI/SLA
Методичен и Иерархичен
Использует графики и дашборды

Хороший troubleshooting:

Опирается на SLO/SLI/SLA
Методичен и Иерархичен
Использует графики и дашборды

Хороший troubleshooting:

Опирается на SLO/SLI/SLA
Методичен и Иерархичен
Использует графики и дашборды

Хороший troubleshooting:

Опирается на SLO/SLI/SLA
Методичен и Иерархичен
Использует графики и дашборды

Дашборды должны отражать процедуру troubleshooting’а*

*SLA-based, методичны и опираться на топологию

Методология USE*

*Utilization, Saturation, Errors
by Brendan Gregg

Пример USE:

E: 5xx’es
S: Задержка / Размер очереди
U: Количество запросов

Пример USE:

E: 5xx’es
S: Задержка / Размер очереди
U: Количество запросов

Пример USE:

E: 5xx’es

S: Задержка / Размер очереди

U: Количество запросов

Пример USE:

E: 5xx’es
S: Задержка / Размер очереди
U: Количество запросов

Пример USE:

E: 5xx’es
S: Задержка / Размер очереди
U: Количество запросов

Топология*

*Upstream’ы и Downstream’ы сервиса на дашбордах

Автоматическое определение зависимостей на основе метрик

RPC
Proxy Mesh
Клиентских Библиотек

Автоматическое определение зависимостей на основе метрик

RPC

Proxy Mesh

Клиентских Библиотек

Автоматическое определение зависимостей на основе метрик

RPC

Proxy Mesh

Клиентских Библиотек

Автоматическое определение зависимостей на основе метрик

RPC

Proxy Mesh

Клиентских Библиотек

Автоматическое определение зависимостей на основе метрик

RPC
Proxy Mesh
Клиентских Библиотек

Drilldown’ы*

*Группировки и Фильтры по тегам метрик

Чем сложнее сервис —
тем больше drilldown’ов

Если на графике одна метрика — вы неэффективно расходуете место на дашборде

Если вы видите spike на графике 5xx’ок то вы всегда начинаете разбираться откуда они

Примеры Drilldown’ов для 5xx’ок:

По типу: 500, 501, 502, 503, 504, …
По пути: /, /admin, /login, /log, …
По ДЦ: SFO, JFK, FRA, LED, …
По клиенту: Desktop, Mobile, API, …

Примеры Drilldown’ов для 5xx’ок:

По типу: 500, 501, 502, 503, 504, …
По пути: /, /admin, /login, /log, …
По ДЦ: SFO, JFK, FRA, LED, …
По клиенту: Desktop, Mobile, API, …

Примеры Drilldown’ов для 5xx’ок:

По типу: 500, 501, 502, 503, 504, …

По пути: /, /admin, /login, /log, …

По ДЦ: SFO, JFK, FRA, LED, …

По клиенту: Desktop, Mobile, API, …

Примеры Drilldown’ов для 5xx’ок:

По типу: 500, 501, 502, 503, 504, …
По пути: /, /admin, /login, /log, …
По ДЦ: SFO, JFK, FRA, LED, …
По клиенту: Desktop, Mobile, API, …

Примеры Drilldown’ов для 5xx’ок:

По типу: 500, 501, 502, 503, 504, …
По пути: /, /admin, /login, /log, …
По ДЦ: SFO, JFK, FRA, LED, …
По клиенту: Desktop, Mobile, API, …

Примеры Drilldown’ов для 5xx’ок:

По типу: 500, 501, 502, 503, 504, …
По пути: /, /admin, /login, /log, …
По ДЦ: SFO, JFK, FRA, LED, …
По клиенту: Desktop, Mobile, API, …

Версия сервиса как тег*

*Позволяет создать Canary vs Production “daily deploy” дашборд

Drilldown’ы должны быть легко доступны*

*Ссылки для быстрого переключения с одного на другой не уходя с дашборда (slice&dice)

Если на графике больше 10 метрик —
вы неэффективно расходуете
человеческое время

Statistics 101
для системных администраторов*

*Или как отобразить тысячи метрик на одном графике

Статистические приёмы:

Перцентили
Среднее + Top N / Bottom N
Std. Dev
Heatmap

Статистические приёмы:

Перцентили
Среднее + Top N / Bottom N
Std. Dev
Heatmap

Статистические приёмы:

Перцентили

Среднее + Top N / Bottom N

Std. Dev

Heatmap

Статистические приёмы:

Перцентили
Среднее + Top N / Bottom N
Std. Dev
Heatmap

Статистические приёмы:

Перцентили
Среднее + Top N / Bottom N
Std. Dev
Heatmap

Статистические приёмы:

Перцентили
Среднее + Top N / Bottom N
Std. Dev
Heatmap

Интеграция дашбордов с остальным сервисами в production

Дашборды не только для графиков!

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Что можно положить на дашборд:

Горящие Alert’ы у сервиса
История деплоев
N строк error log’а
N последних RPC ошибок
Лог авторемедиаций
Top N эксепшнов
Кто oncall

Резюмируя

Не важно какая у вас система мониторинга,
важно какие интерфейсы она
предоставляет и как они используются

Резюмируя

Задача команды мониторинга: предоставлять интерфейсы,через которые инженеры смогут сделать эффективный мониторинг

Резюмируя

Задача инженеров компании: расширять возможности системы мониторинга интегрируя её со всеми остальными инфраструктурами сервисами через предоставленные интерфейсы

Q&A

@SaveTheRbtz
rbtz@dropbox.com

Backup Slides*

*не уж-то до сюда долистали?!

Метрики

Уникальный Request-ID*

*Генерируется на входе в сеть

Tracing*

*Должен поддерживаться Framework’ами или RPC

Tracing для бедных

Через метрики RPC/Routing Mesh
…или ручную инструментарию кода

Tracing для бедных

Через метрики RPC/Routing Mesh

…или ручную инструментарию кода

Tracing для бедных

Через метрики RPC/Routing Mesh

…или ручную инструментарию кода

Tracing для бедных

Через метрики RPC/Routing Mesh
…или ручную инструментарию кода

Алёрты

Переизбыток алёртов —
это хуже, чем их недостаток!

Дашборды

Service Discovery имя как тег*

*Позволяет автоматом создавать дашборды для новых инсталляций сервиса

Q&A

@SaveTheRbtz
rbtz@dropbox.com