

М.ВидеоЭльдорадо

Переход в публичное облако

Ярослав Иссинский

2022 Q3



М.ВидеоЭльдорадо

Бэкграунд

Разработка частных облаков – 7 лет

Разработка гибридных облаков – 3 года

Сейчас

СТО/СРО домена

Техническая платформа



Домен:
Техническая платформа

ИТ-продукт:
IaaS, PaaS и Managed Services в Облаке

- IaaS (Яндекс Облако)
- Kubernetes
- PaaS (Kafka, PostgreSQL, Mongo, Couchbase, Vault, Redis, Яндекс S3 Object Storage и т.д.)
- Observability (Prometheus, Grafana, Loki, Jaeger, Sentry)
- Поддержка IaaS и PaaS
- Замещающие сервисы на open source решениях
- Консультации

ИТ-продукт:
Производственный процесс (CI/CD)

- Jira
- Библиотека шаблонов (Terraform, Helm)
- GitLab, Stash/Bitbucket (код репозиторий)
- Консультации и поддержка инструментов CI/CD, Jira, Confluence
- GitLab CI (пайплайны)
- Управление стандартами и методологиями пайплайнов
- Хранилище артефактов
- Confluence (wiki)
- Измерение метрик производственного процесса в командах

DevOps / SRE

Методология и Техрадар производственного процесса

FinOps

Единая поддержка технической платформы

Шаг 0

Зачем переходить в облако?



Правда

- Быстрее получаешь ресурсы
- Удалил → не платишь
(нет денег → удалил)
- Независимость от
инфраструктурной команды
(ну, почти...)
- Делает разработчиков и
DevOps счастливее
- Вам нужна новая экспертиза

Неправда

- Облако дешевле
- Облако надежнее
- Не нужны компетенции

Pattern

- Центр компетенций
- Нанять эксперта, который уже это делал
- Сформулировать цели и критерии успешности

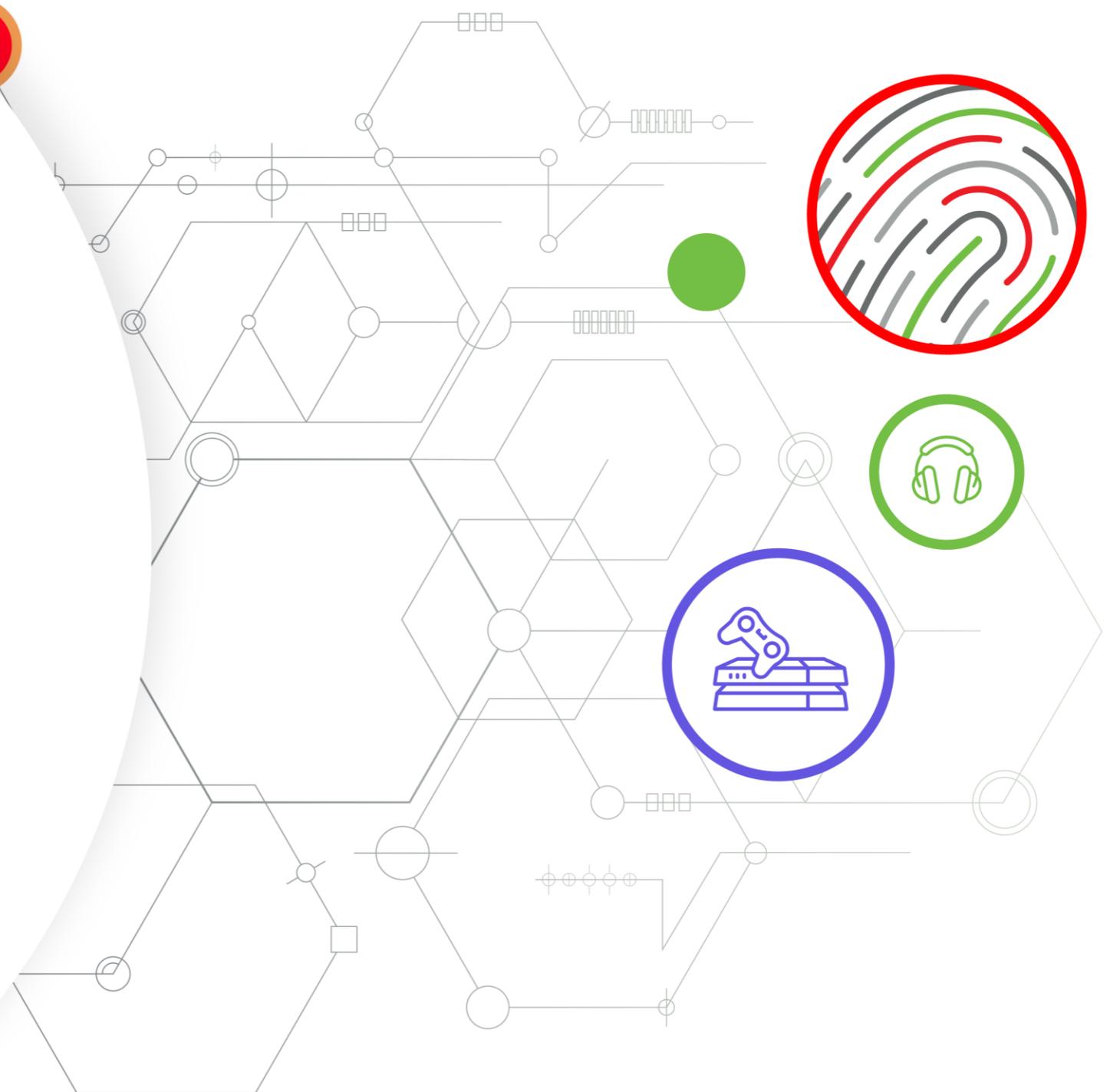
Antipattern

- Отдать облака команде легаси инфраструктуры
- Делать все силами подрядчиков

Использование подрядчиков – это прокачивание чужой экспертизы за свои деньги

Шаг 1

Что отправить в облако?



Pattern

- В облако едут cloud native системы
- Остальные трансформируются и догоняют или остаются

Antipattern

- В облако едет всё, включая legacy
- Lift-and-shift подход

Почему

**В облаке нет
100% доступности**

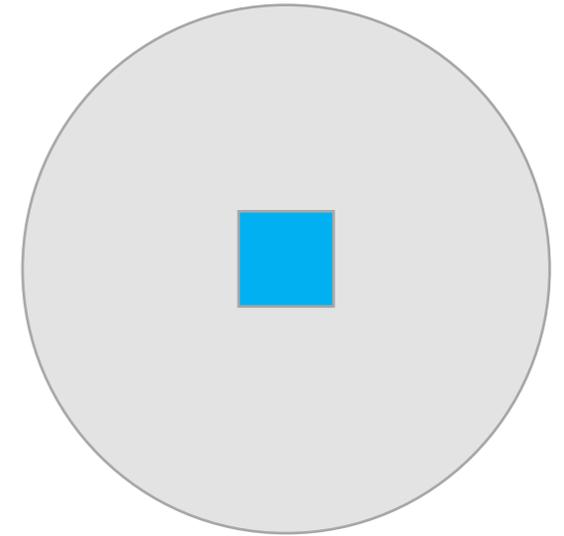


Начальные условия

Доступность в Облаке: 99,95%

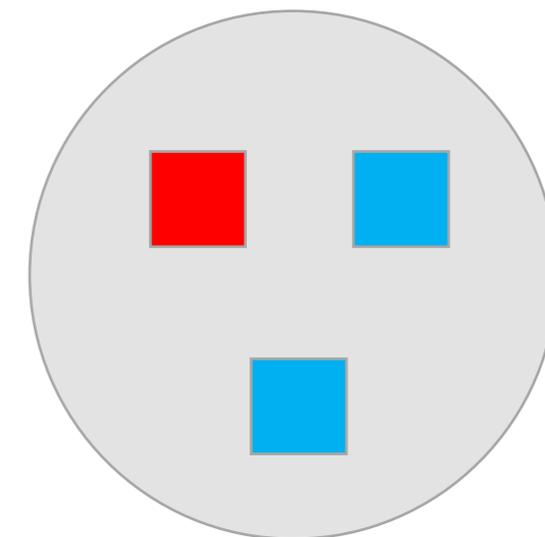
Система из 1 вм

Доступность: 99,95%



Начальные условия

Доступность в Облаке: 99,95%



Система из 3 вм, жесткая зависимость от каждой вм

Вероятность отказа суммируется:

$$3 \times 0,05\% = 0,15\%$$

Доступность: $100\% - 0,15\% = 99,85\%$ ▼

Начальные условия

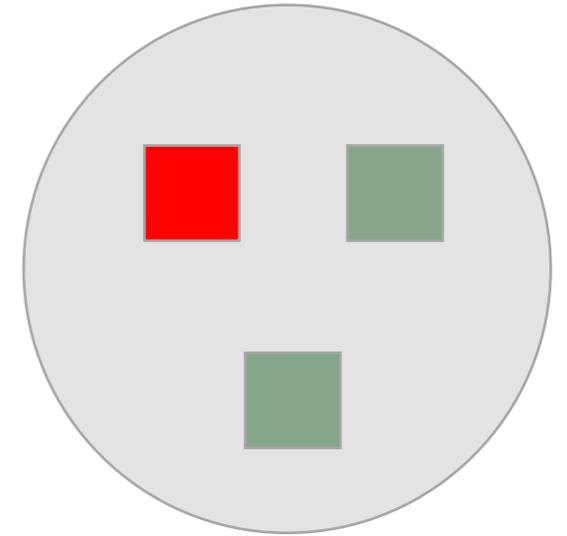
Доступность в Облаке: 99,95%

Система из 3 вм, работает пока жива хотя бы 1 вм

Вероятность отказа перемножается:

$$0,05\% \times 0,05\% \times 0,05\% = 0,000125\%$$

Доступность: 99,999875% ▲



Вывод

Если надежности одного Облака недостаточно,
вам нужно +1 Облако

или как-то обеспечить независимость отказов в одном Облаке

War 2

Infrastructure as Code



Pattern

- Terraform
- Ansible
- Kubernetes для микросервисов

Antipattern

- Накликивание мышкой
- «Service Desk API»

Шаг 3

Техрадар или техстек/тулстек анархия



Pattern

- Золотая середина между диктатом и анархией
- Есть Техрадар
- Диктат на уровне домена
- Все архитекторы знают как новая сущность попадает в Техрадар

Antipattern

- Нет единого понимания «что такое хорошо и что такое плохо»
- Каждая команда окуклилась и занимается велосипедостроением

Техрадар – диаграмма, на которой визуализированы IT технологии и инструменты с разбивкой по сегментам и кольцам

Шаг 4

**Разделите
продукты, проекты
и процессы**



Pattern

- ЗНИ/ЗНО – одна команда
- Разработка – другая команда
- Платформа – это ограниченный набор сервисов (см. Техрадар)
- Предоставлять готовые продукты, а не кубик за кубиком

Antipattern

- «А мы всё делаем по спринтам»
- Ресурсы отдельно, мониторинг отдельно, бэкап отдельно

NB: There is no one ring to rule them all

Шаг 5

Культура SRE



Pattern

- SRE подход
- Observability
(Метрики, логи, трейсы)
- Postmortem

Antipattern

- Облако всегда работает, а когда что-то не работает, то во всём виноват провайдер

<https://sre.google/books/>

Шаг 6

Агрегация метрик



Pattern

- Агрегация метрик доступности
- Необходим слой абстрагирования, который отделил бы индивидуальность сервиса.
У нас Grafana и Sloth.
- Status pages

Antipattern

- 100 продуктов –
100 индивидуальных
дашбордов

<https://sloth.dev/introduction/dashboards/>

Шаг 7

Постоянное улучшение



Pattern

- Сначала делать MVP
- Вносить улучшения по мере необходимости: не видно проблемы на status page во время инцидента → триггер к улучшению

Antipattern

- Включить перфекциониста и упороться на год в сложные метрики
- Никогда не пересматривать SLI/SLO

Шаг 8

Кто всё
это делает?



Pattern

- Люди внутри продуктовой команды
- Критерий: ходят на планирование и ретро

Antipattern

- Мы наняли подрядчика на суппорт и они – наш DevOps as a Service

<https://devopstopologies.ru/>
<https://teampologies.com/>

Шаг 9

Кто за это платит?



NB: Юнит-экономика – это классная штука

Pattern

- Каждый продукт имеет свой бюджет
- Все ресурсы аллоцированы на продукты
- Владельцы продуктов ежемесячно получают детализацию / счет

Antipattern

- За всё в облаке платит ДИТ

<https://www.finops.org/>

Шаг 10

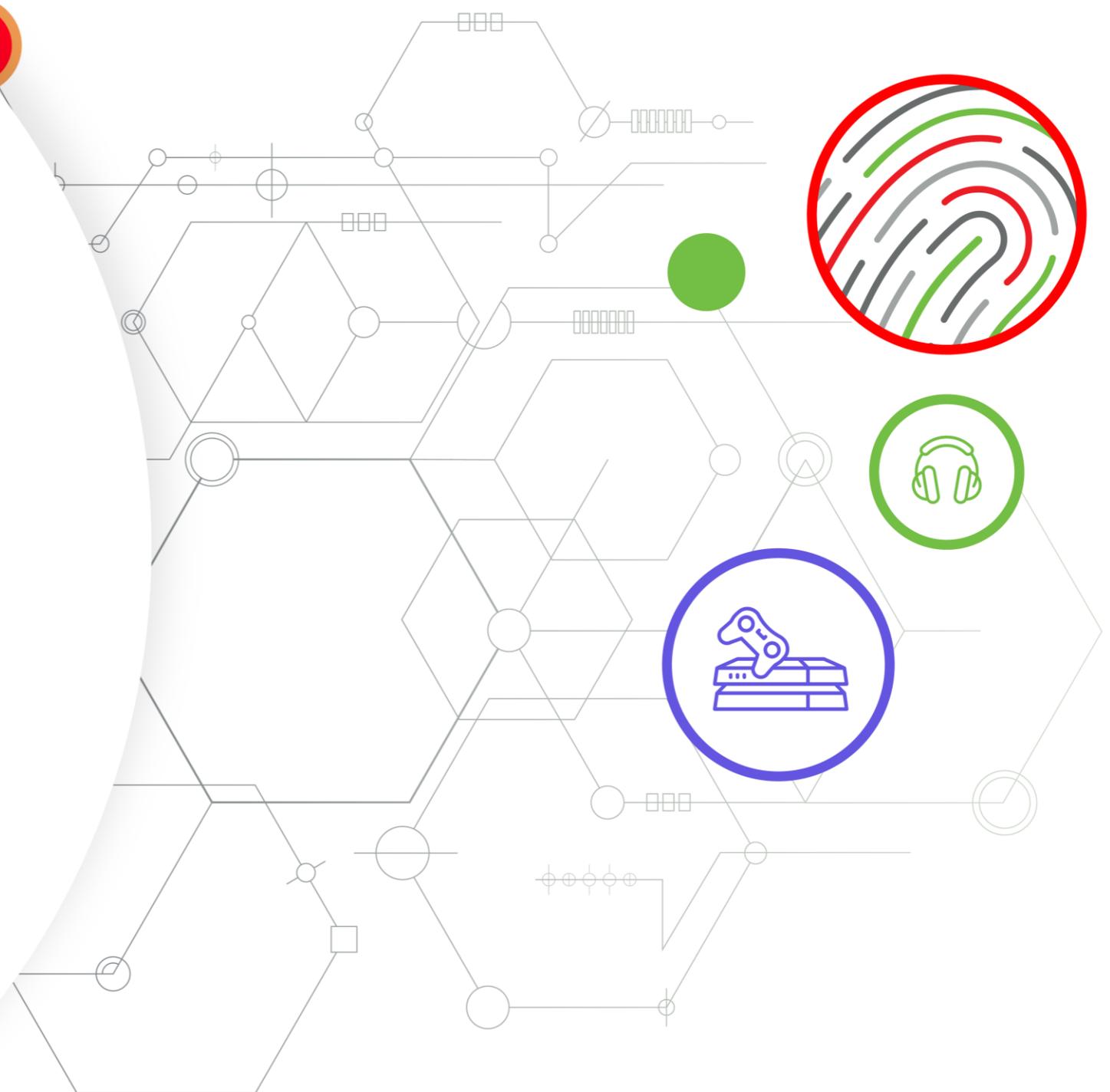
**Без чего всё это не
взлетит**



- Регулярные коммуникации между командами (например, кросс-командные викли синки DevOps/SRE)
- Публичное демо по результатам спринта
- Открытый микрофон
- Архкомы
- Круглый стол СТО/СРО

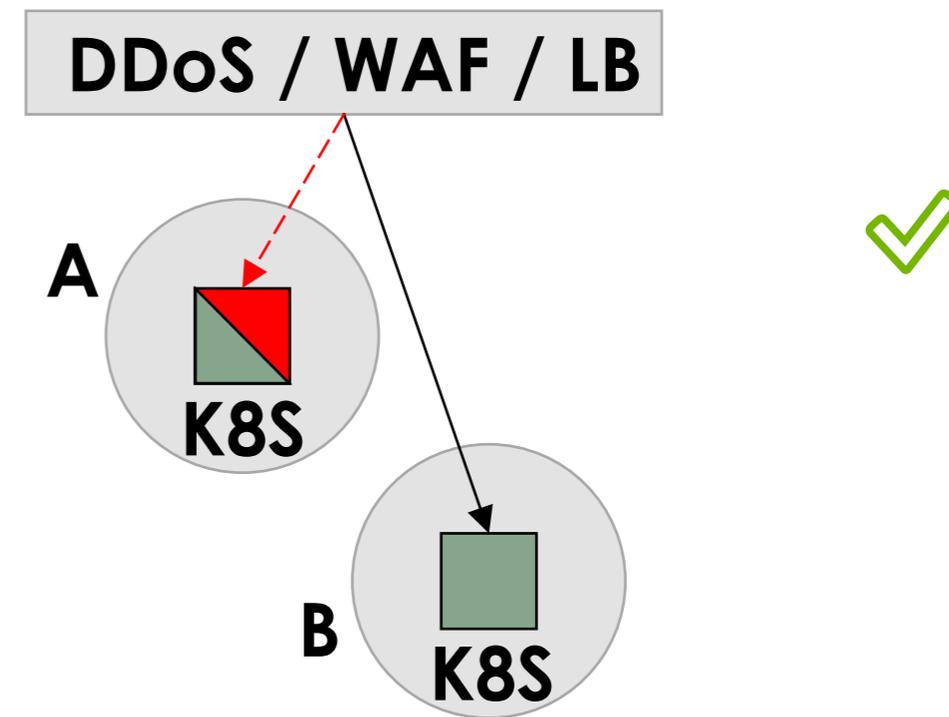
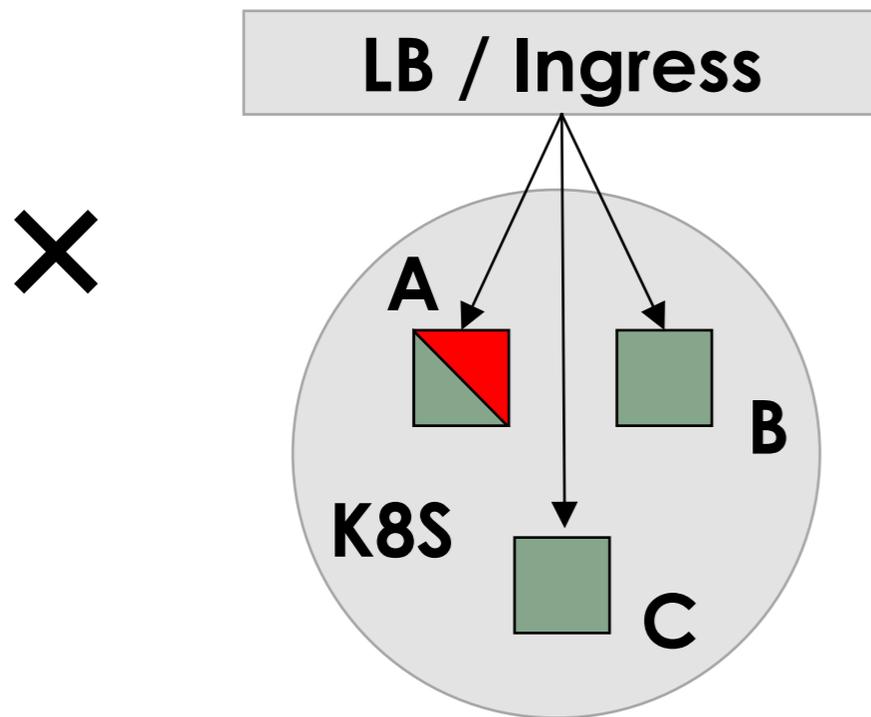
<https://www.atlassian.com/devops/frameworks/calms-framework>

Практика



Поломка

Зона А выборочно тупит

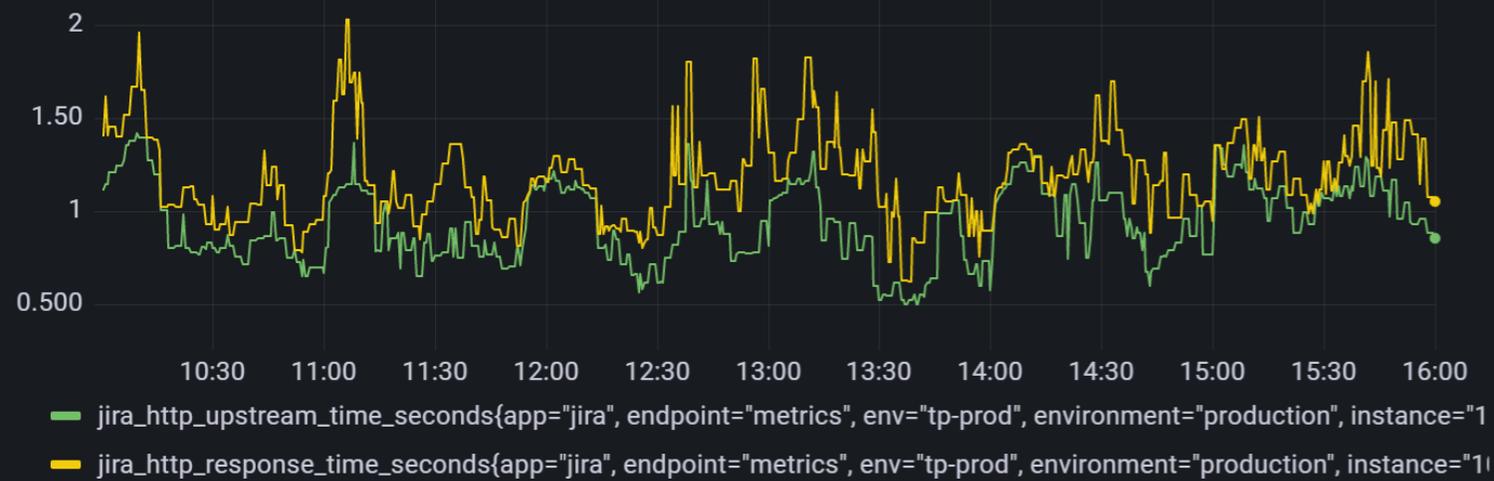


Service metrics – Квантили

Atlassian / Jira response time ★ 🔗

📊 📄 ⚙️ 🕒 Last 6 hours 🔍 🔄 30s 🗨️

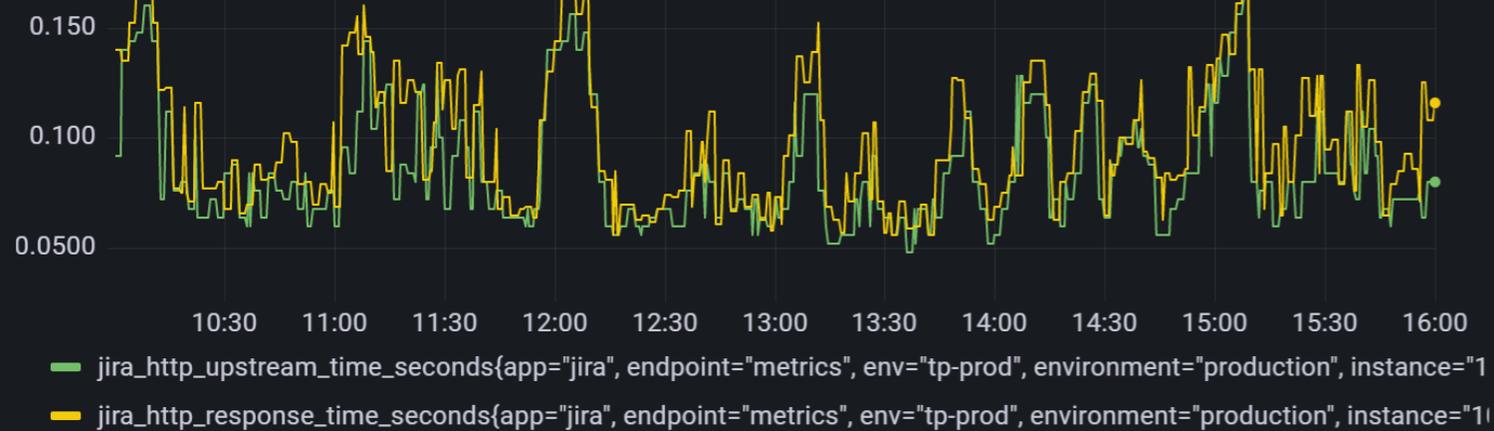
Jira response time: 99 quantile (200 GET)



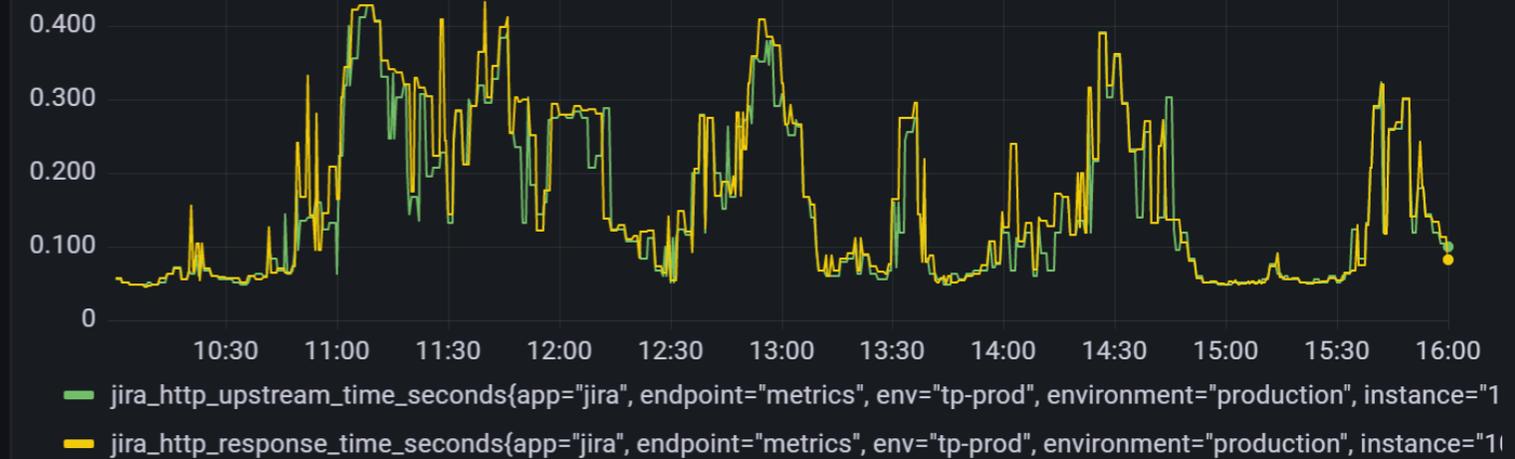
Jira response time: 99 quantile (200 POST)



Jira response time: 90 quantile (200 GET)

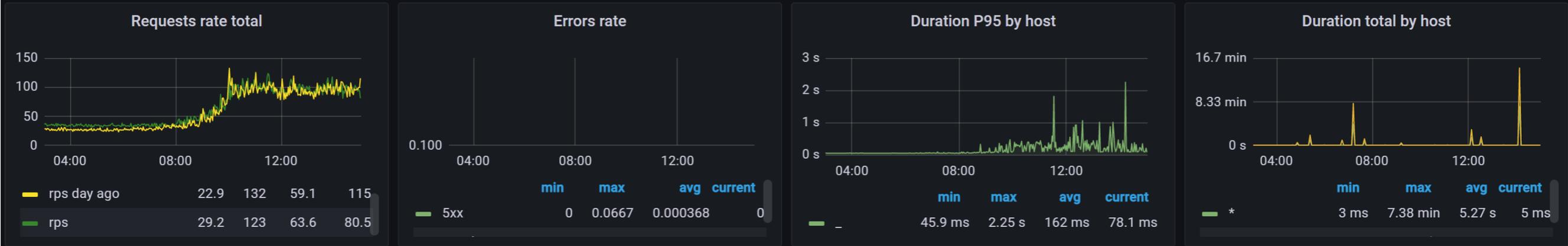


Jira response time: 90 quantile (200 POST)

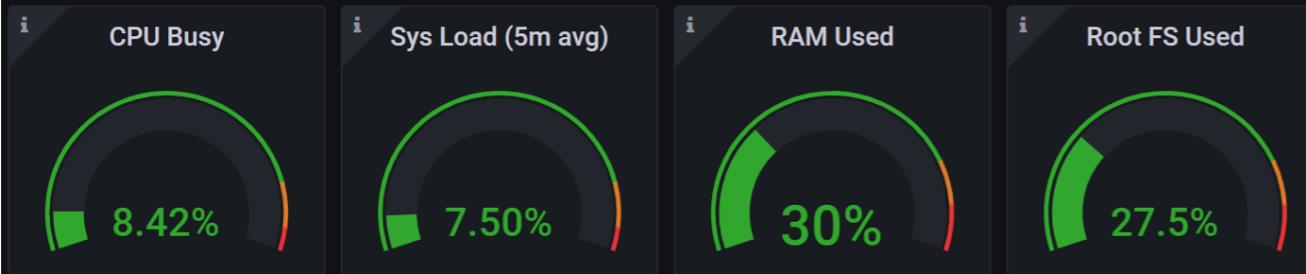


Service metrics – RED & App metrics

Rate, Errors, Durability

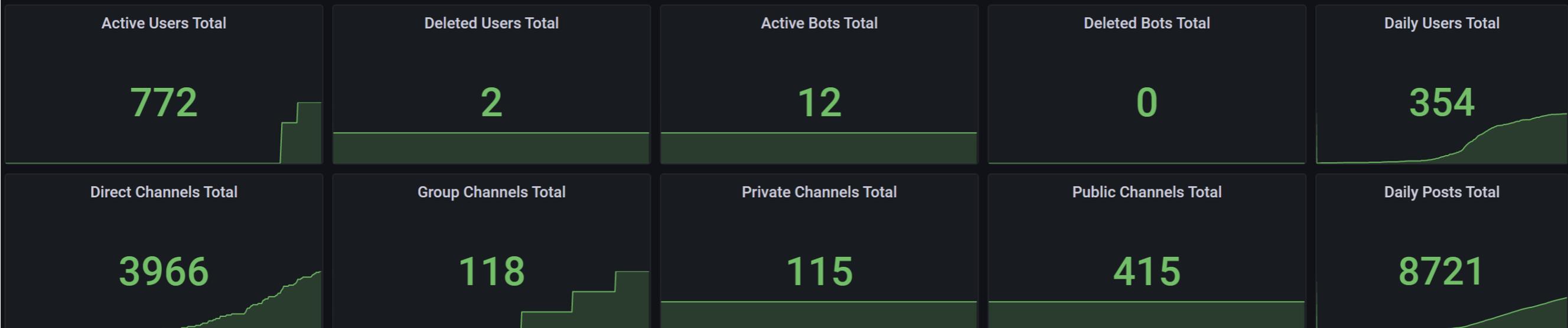


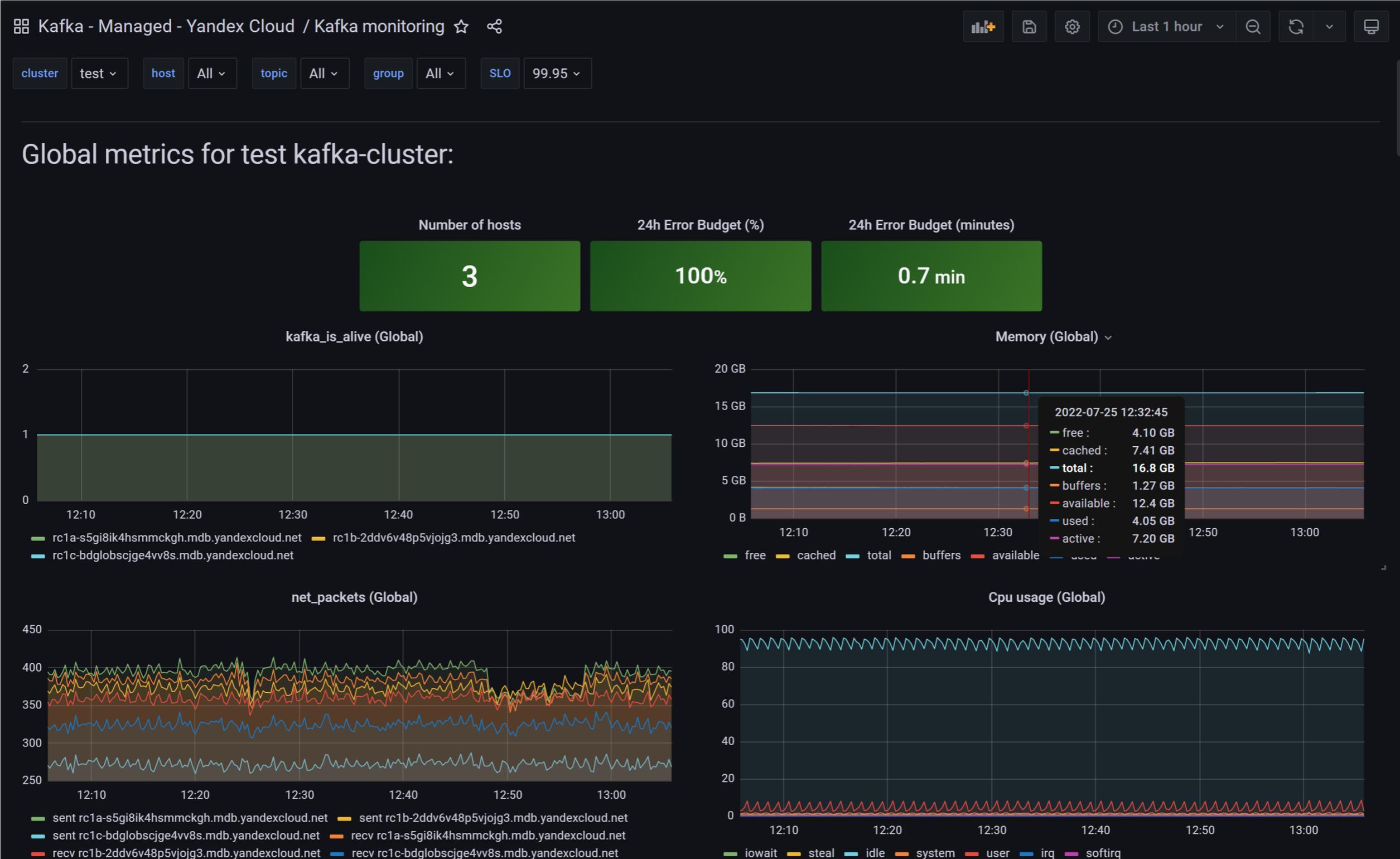
Mattermost VM

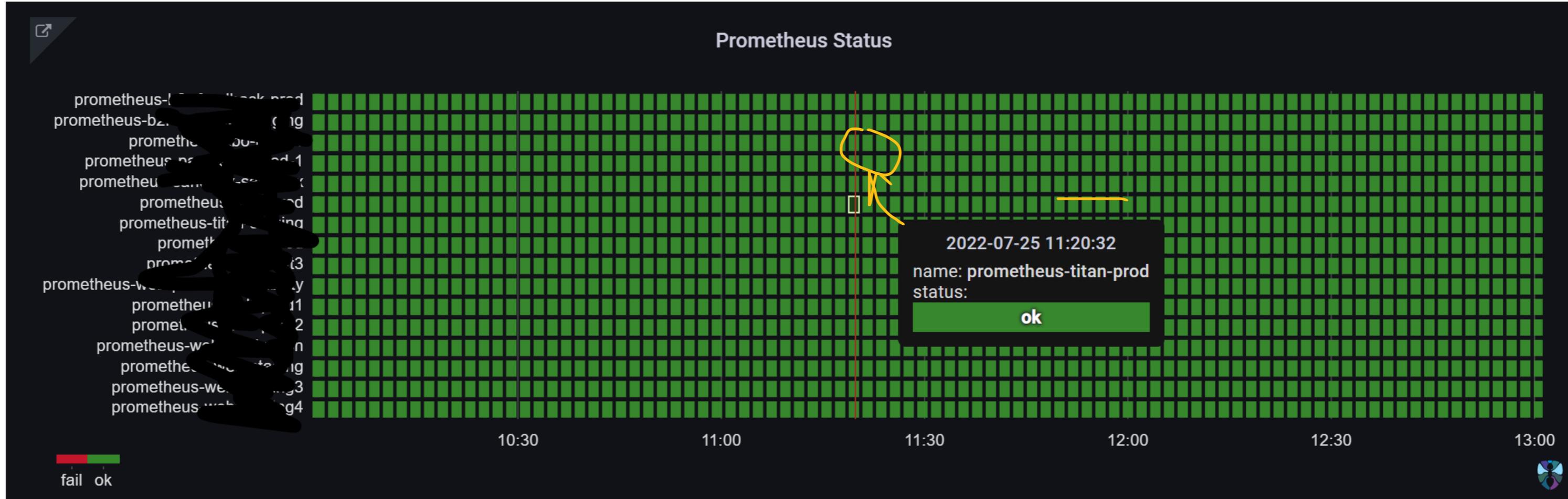


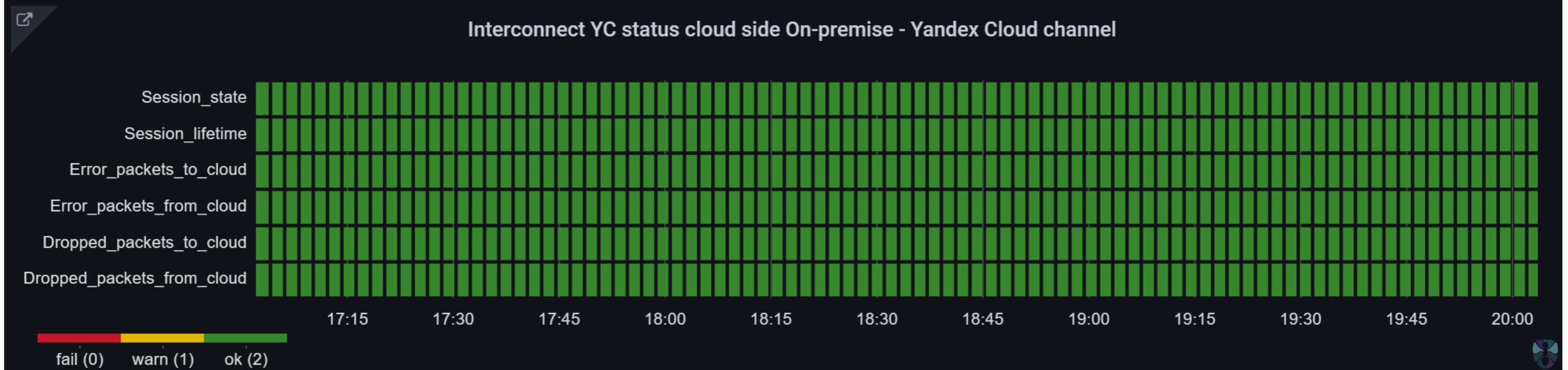
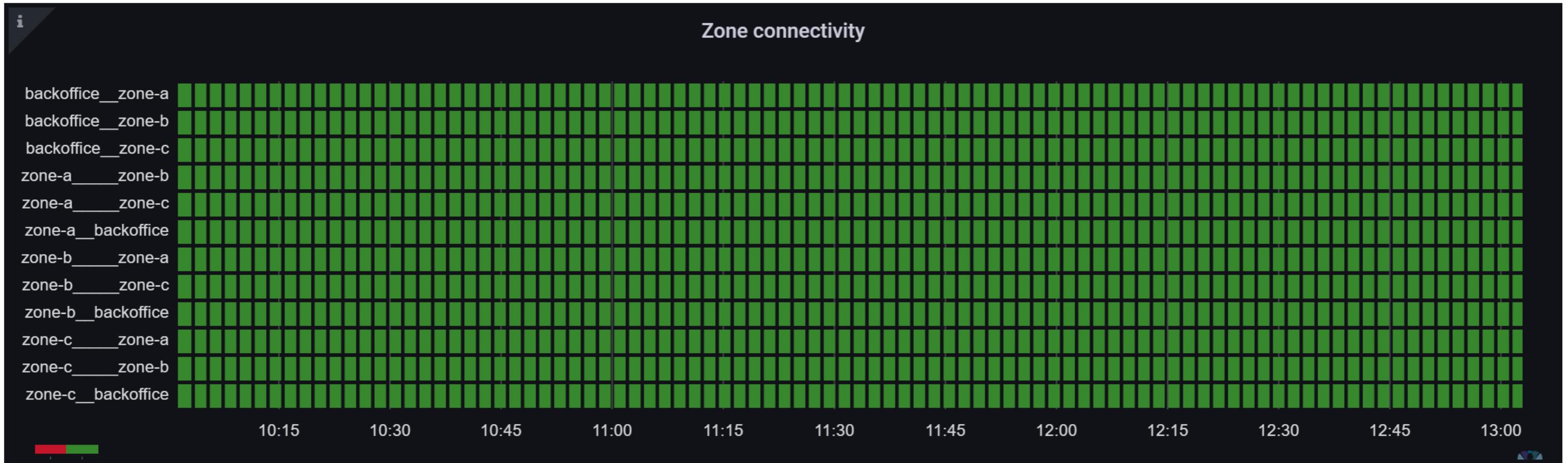
Mattermost DB (5 panels)

Mattermost app

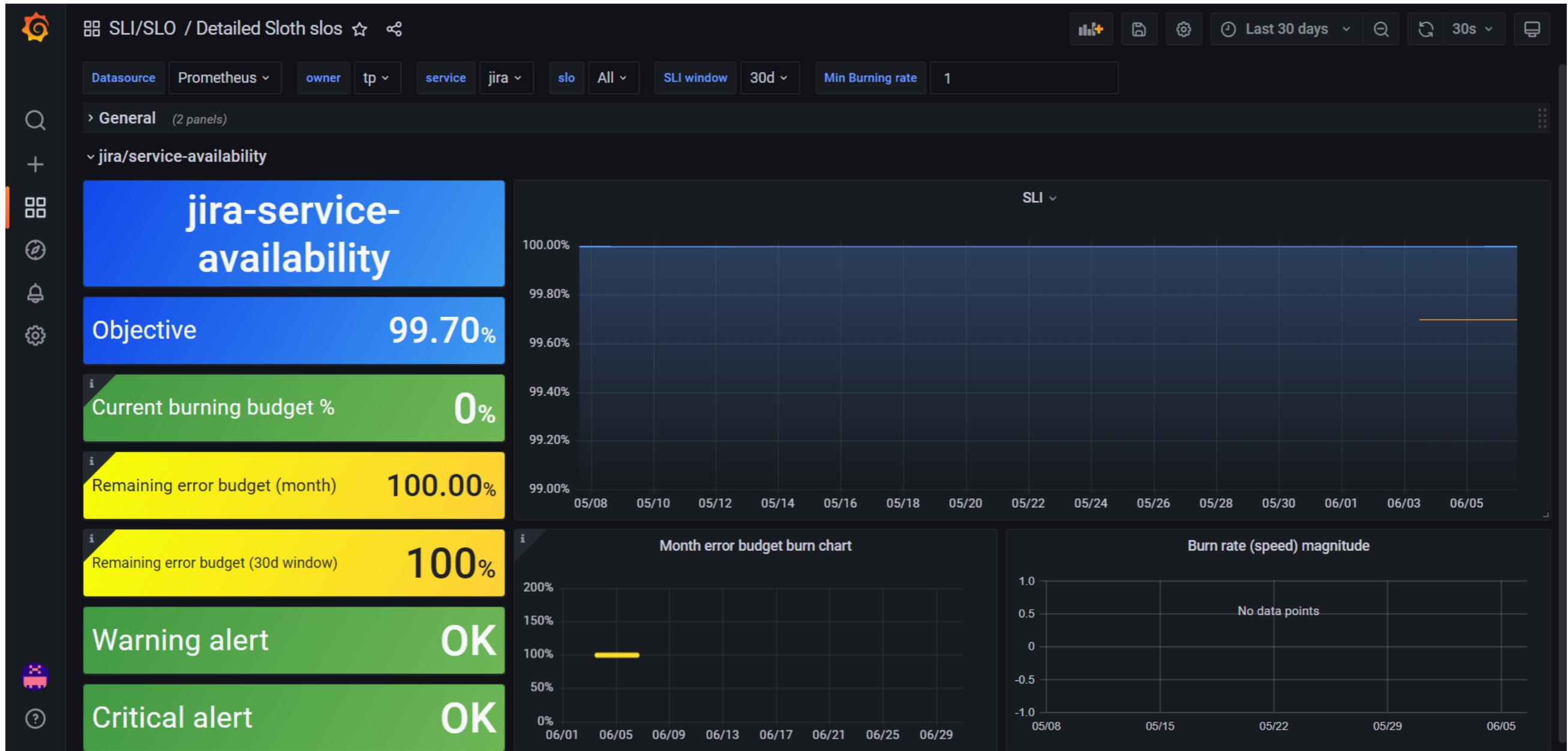








Sloth для конкретного продукта





Alertmanager BOT 01:51

[FIRING:1] ZoneConnectivityProblem for (instance="blackbox-monitoring-backoffice.tech.mvideo.ru:9115", prometheus="monitoring/prometheus-operator-prometheus", status="0", zone="zone-c__backoffice")

Alert: - critical

Description: Connectivity problem zone-c__backoffice. Check blackbox-monitoring-backoffice.tech.mvideo.ru:9115.

Dashboard: <http://grafana.tech.mvideo.ru/d/hN9eGp77z/sre-dashboard?orgId=1>

Runbook URL:

Details:

- alertname: ZoneConnectivityProblem
- instance: blackbox-monitoring-backoffice.tech.mvideo.ru:9115

Show more



1 reply | Follow



Alertmanager BOT 01:51

[RESOLVED] ZoneConnectivityProblem for (instance="blackbox-monitoring-backoffice.tech.mvideo.ru:9115", prometheus="monitoring/prometheus-operator-prometheus", status="0", zone="zone-c__backoffice")

Alert: - critical

Description: Connectivity problem zone-c__backoffice. Check blackbox-monitoring-backoffice.tech.mvideo.ru:9115.

Dashboard: <http://grafana.tech.mvideo.ru/d/hN9eGp77z/sre-dashboard?orgId=1>

Runbook URL:

Details:

- alertname: ZoneConnectivityProblem
- instance: blackbox-monitoring-backoffice.tech.mvideo.ru:9115

Show more



Aleks... 06:36

влияние работ на сети:

Уважаемые коллеги,

Информируем Вас, что 27.07.2022 будут проводиться работы по объединению фабрик SAN сети между ЦОД Волочаевская, ЦОД Новь и ЦОД Компрессор.

Время проведения работ - с 01:00 до 4:00.

Прерывания в работе сервисов не планируется.

С уважением, ДИТ

предварительная информация о сбое:

Работа 4 - Переключение блейд-корзин (серверов) DEBS01, DEBS04, HPBS08 в новые коммутаторы - выполнено за исключением DEBS04 переключение этой корзины вызвало недоступность части сервисов вернули обратно.

Edited

Reply to this thread...



Вопросы

