

Думай как SRE *[v2.1]*

Михаил Кучеренко,
Аспирант кафедры ИУ5, МГТУ им. Н.Э. Баумана,
SRE, Руководитель команды CorpCloud, VK


08.11.2022

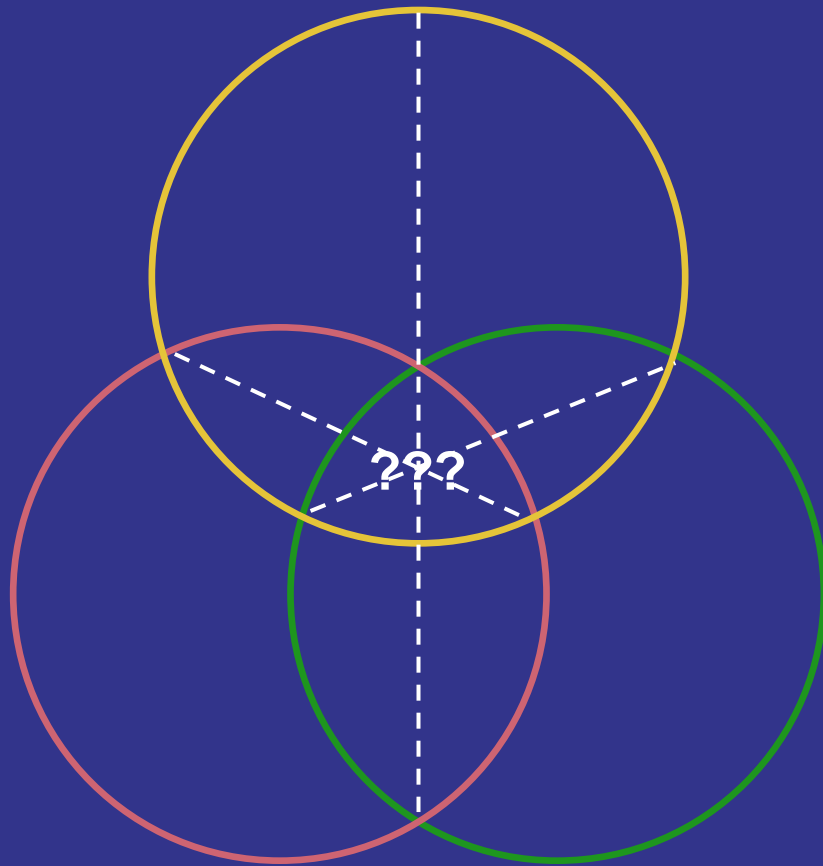
Встречаются два DevOps-евангелиста на конференции:

- Как сделать, чтобы релизить быстро и качественно?*
- Могу рассказать.*
- Рассказать я и сам могу, а как сделать-то?*

(C) [@ibaruch](#)

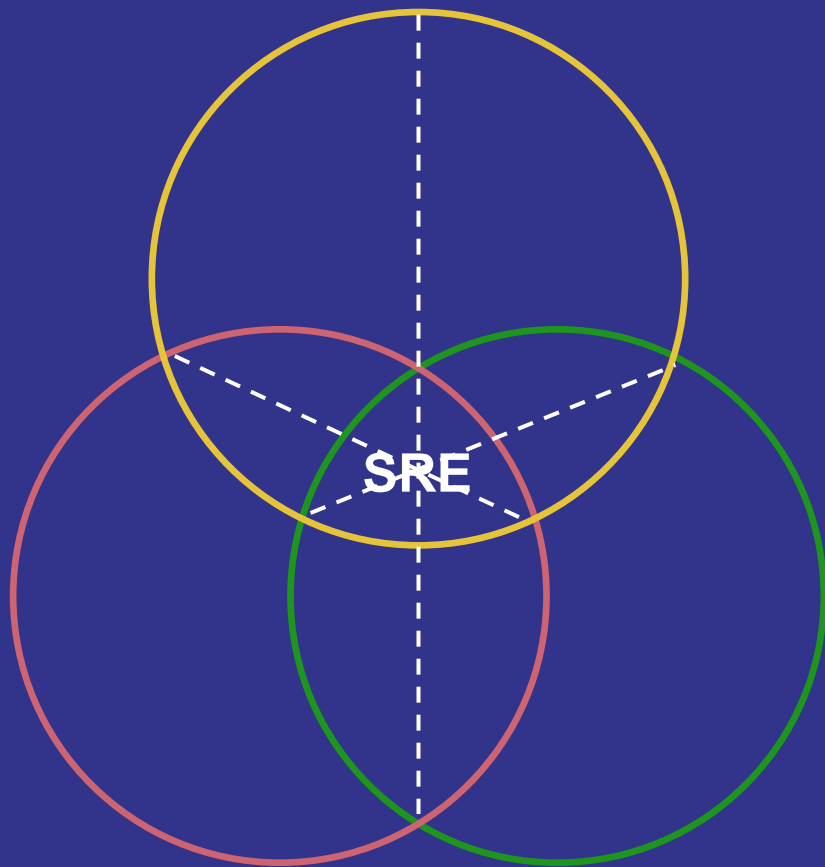
Обсудим:::

- 
1. Введение
 2. Определение
 3. Философия
 4. Архитектура
 5. Автоматизация
 6. Мониторинг
 7. Итоги



Что такое DevOps?

- Философия и методология
- Активное взаимодействие разработки и эксплуатации
- Сократить time-to-market
- Реализацией и расширением можно считать подход SRE



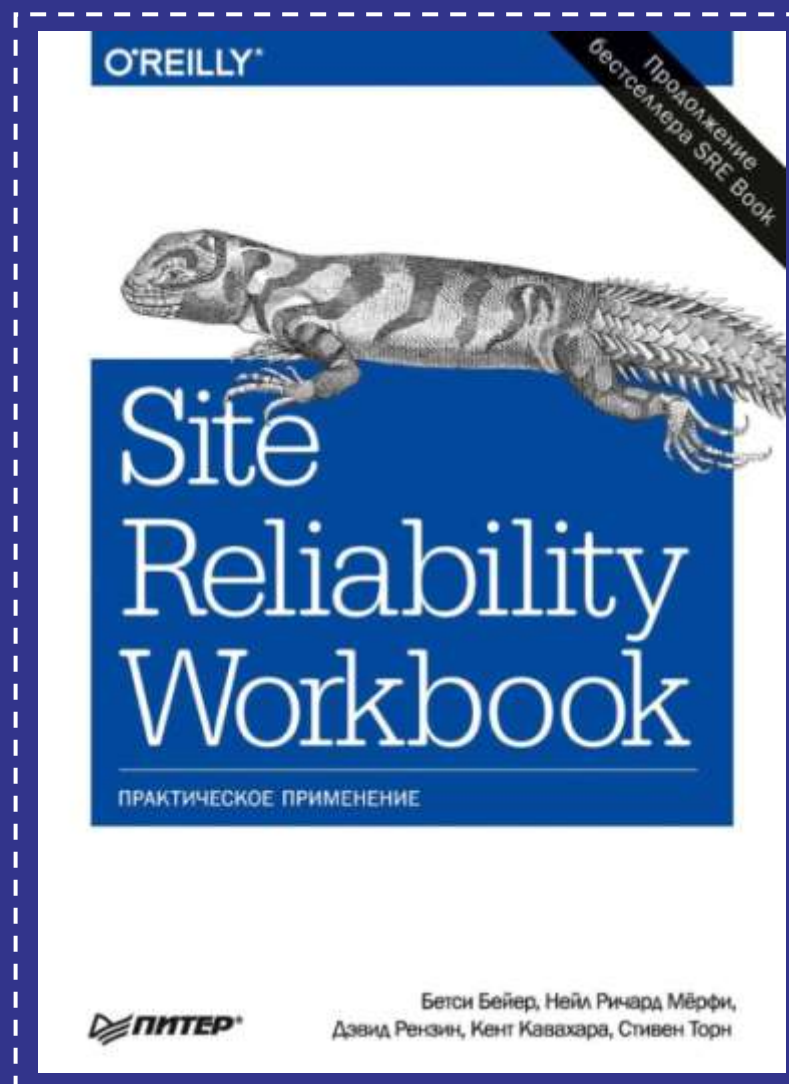
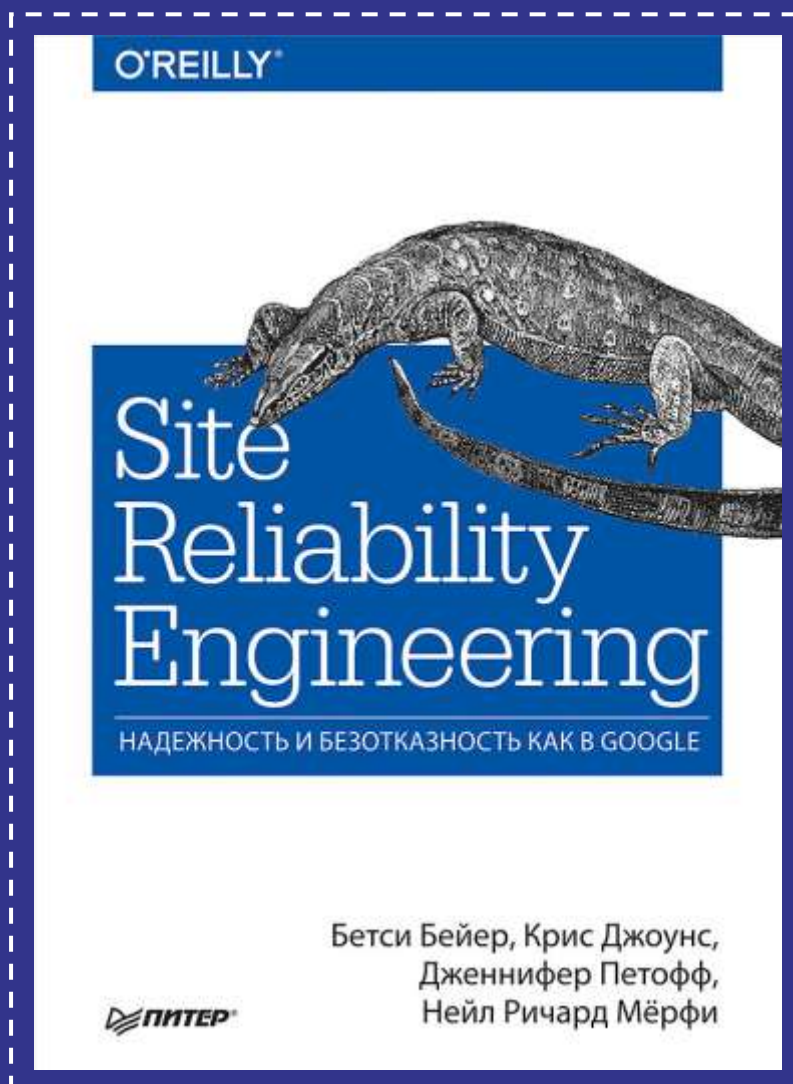
Что такое SRE?

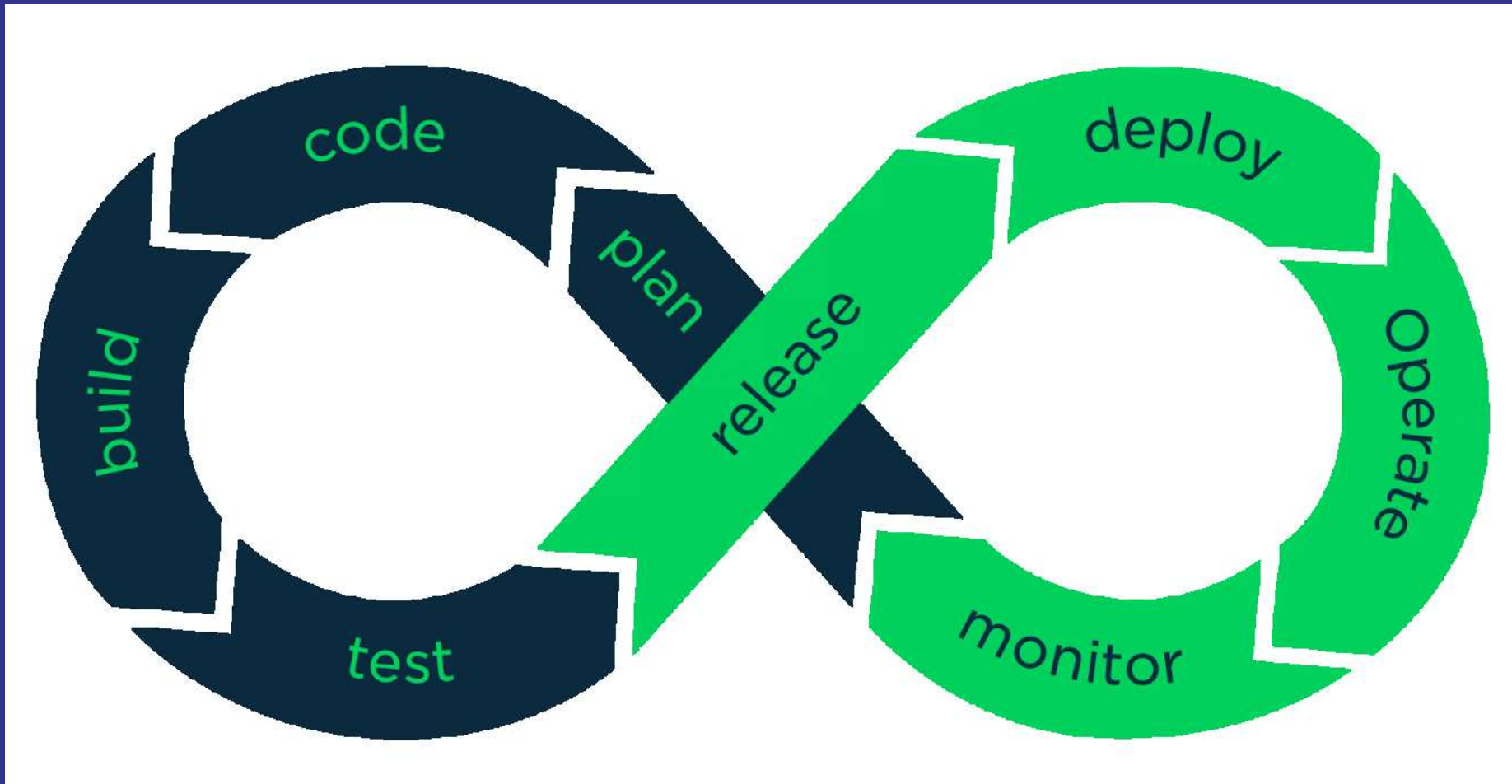
Site Reliability Engineering
Реализация философии DevOps

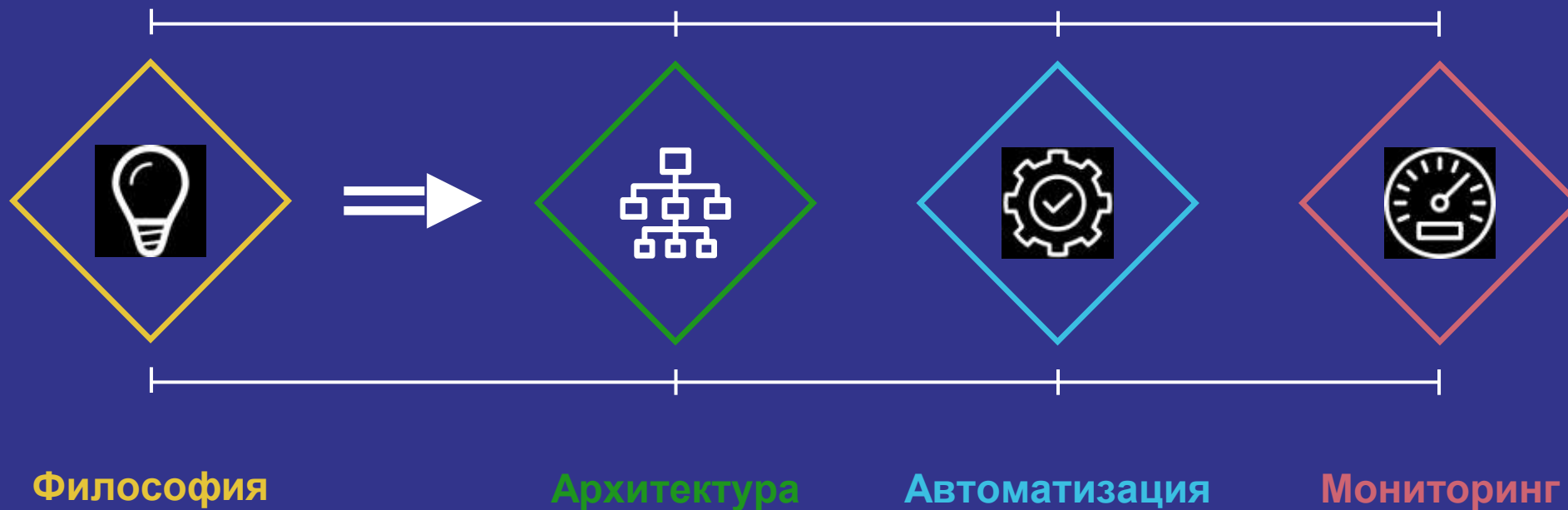
Все понимают по-разному

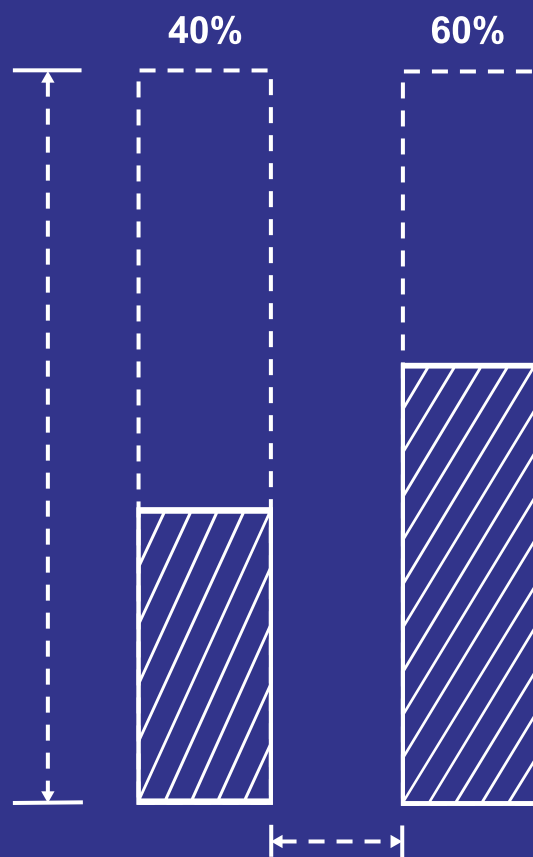
||

Никто не понимает







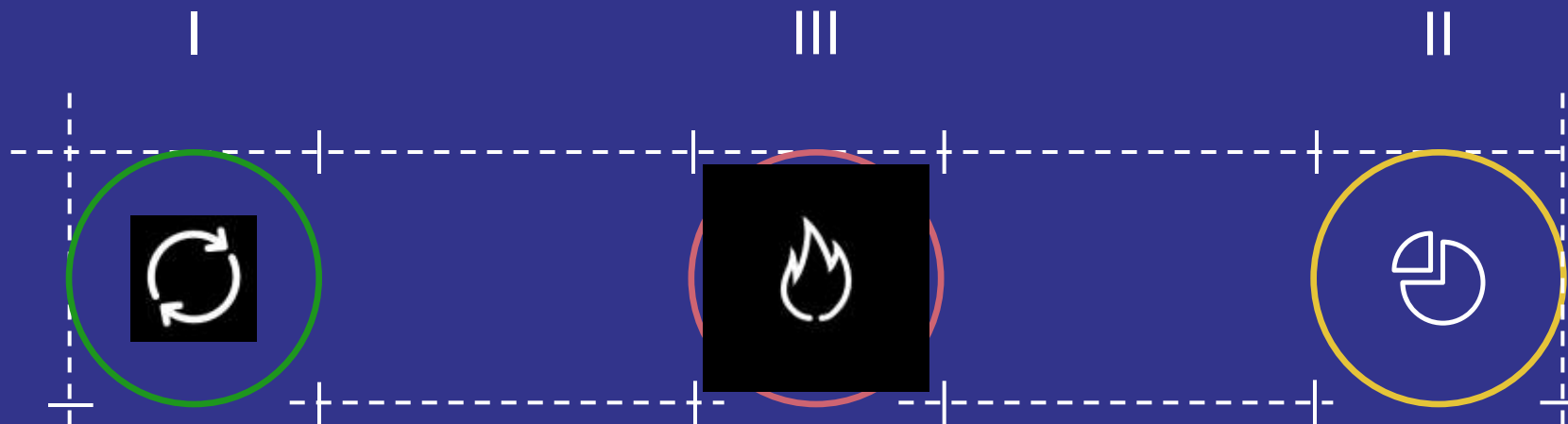


< 50% Операционные задачи

Реакция на инциденты, поддержка,
рутинные и ручные действия

> 50% Инженерные задачи

Развитие сервиса и автоматизация



Развертывание

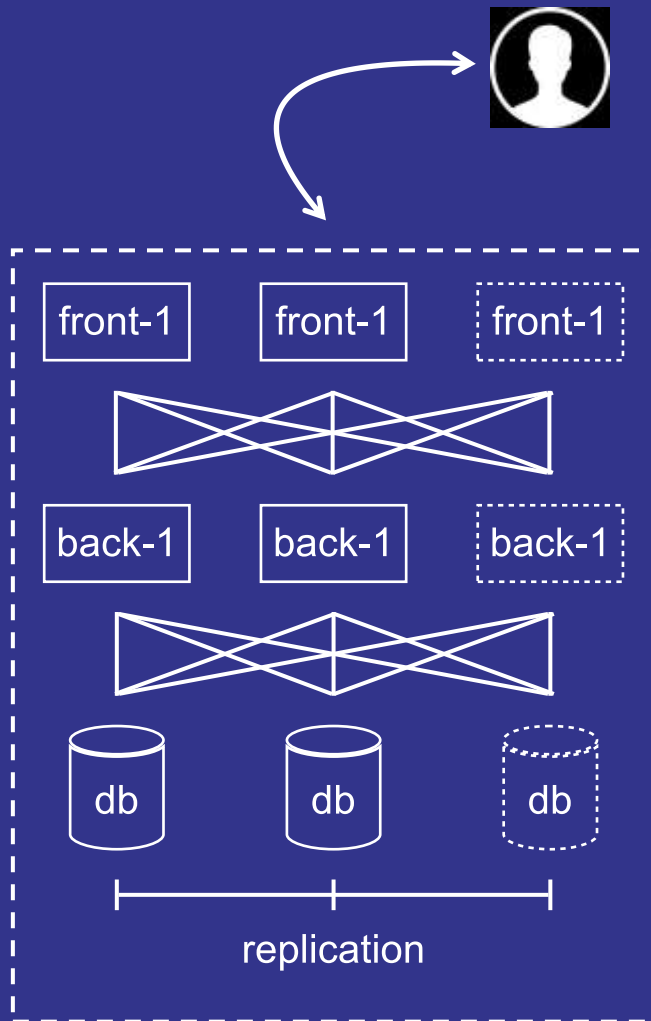
Поэтапное обновление
Выявление проблем
Безопасно откатывать

Отказы неизбежны

Резервирование
Масштабируемость
«Graceful degradation»

Отслеживание

Оценка SLA, SLO, SLI
Бюджет ошибок
Мониторинг



GNU/Linux

Ядро Linux + ПО GNU, открытое ПО

Резервирование и масштабирование

На уровнях приложения, сети, железа и даже отдельных ЦоД

Резервирование

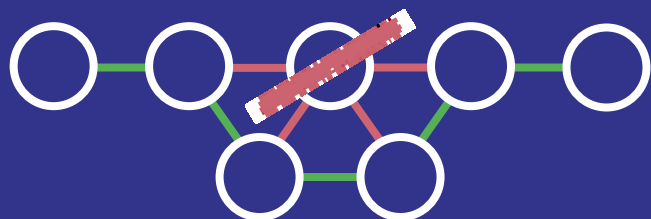
На уровнях приложения, сети, железа и даже ЦоД

Сеть

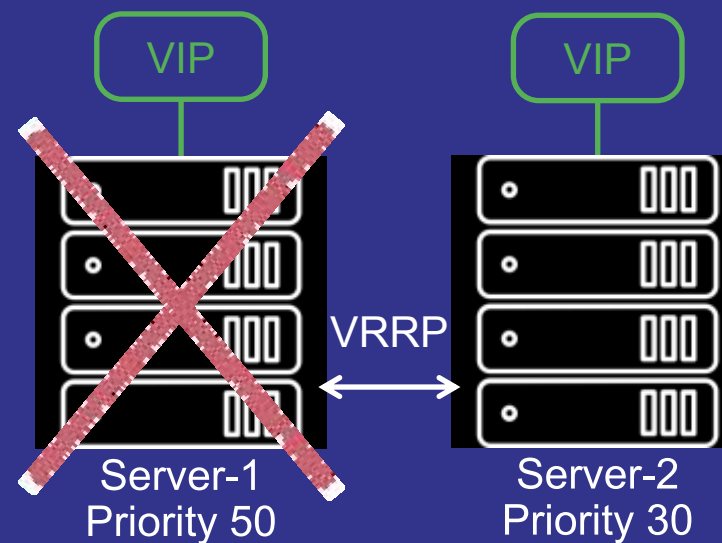
DNS:

```
mail.ru has address 94.100.180.200  
mail.ru has address 217.69.139.202  
mail.ru has address 217.69.139.200  
mail.ru has address 94.100.180.201
```

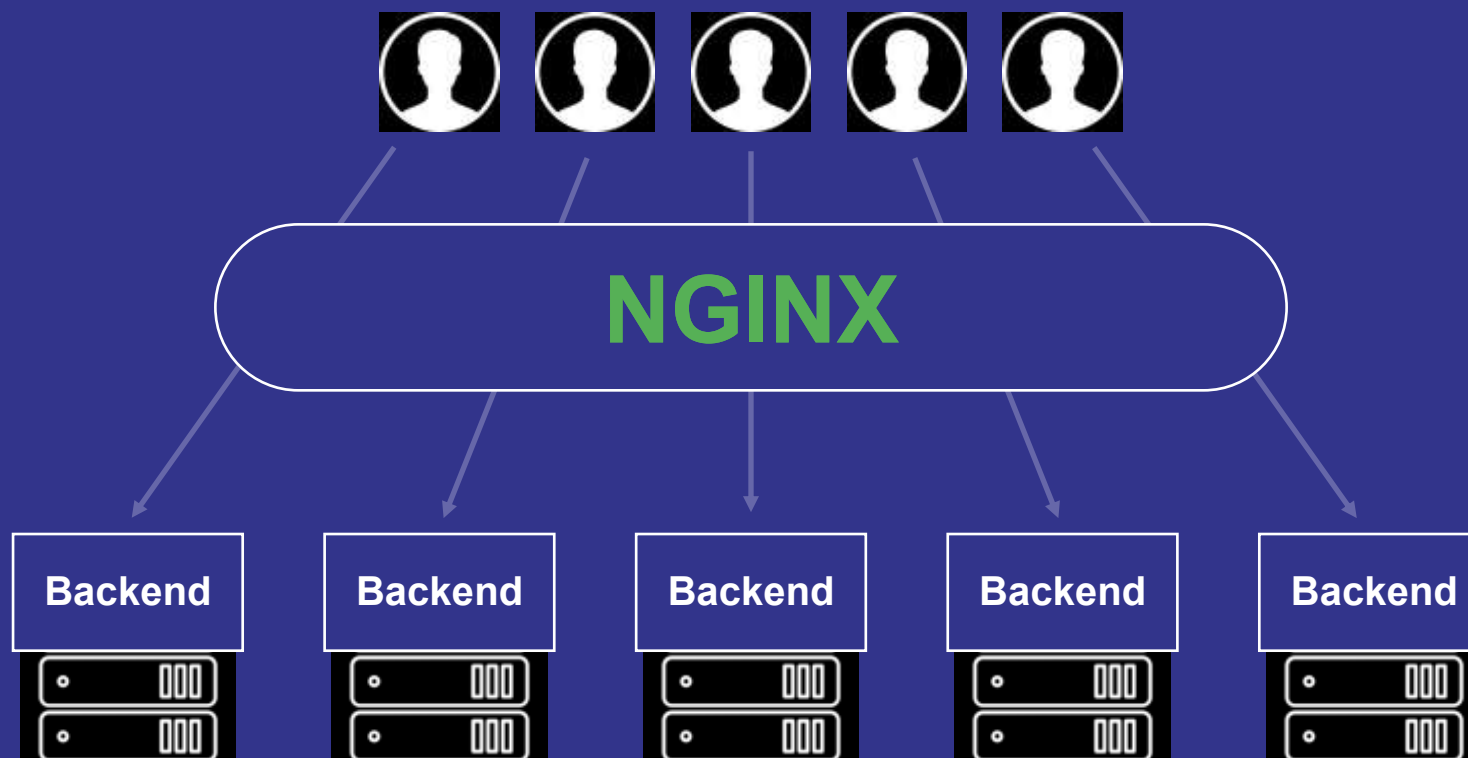
RIP & OSPF & BGP



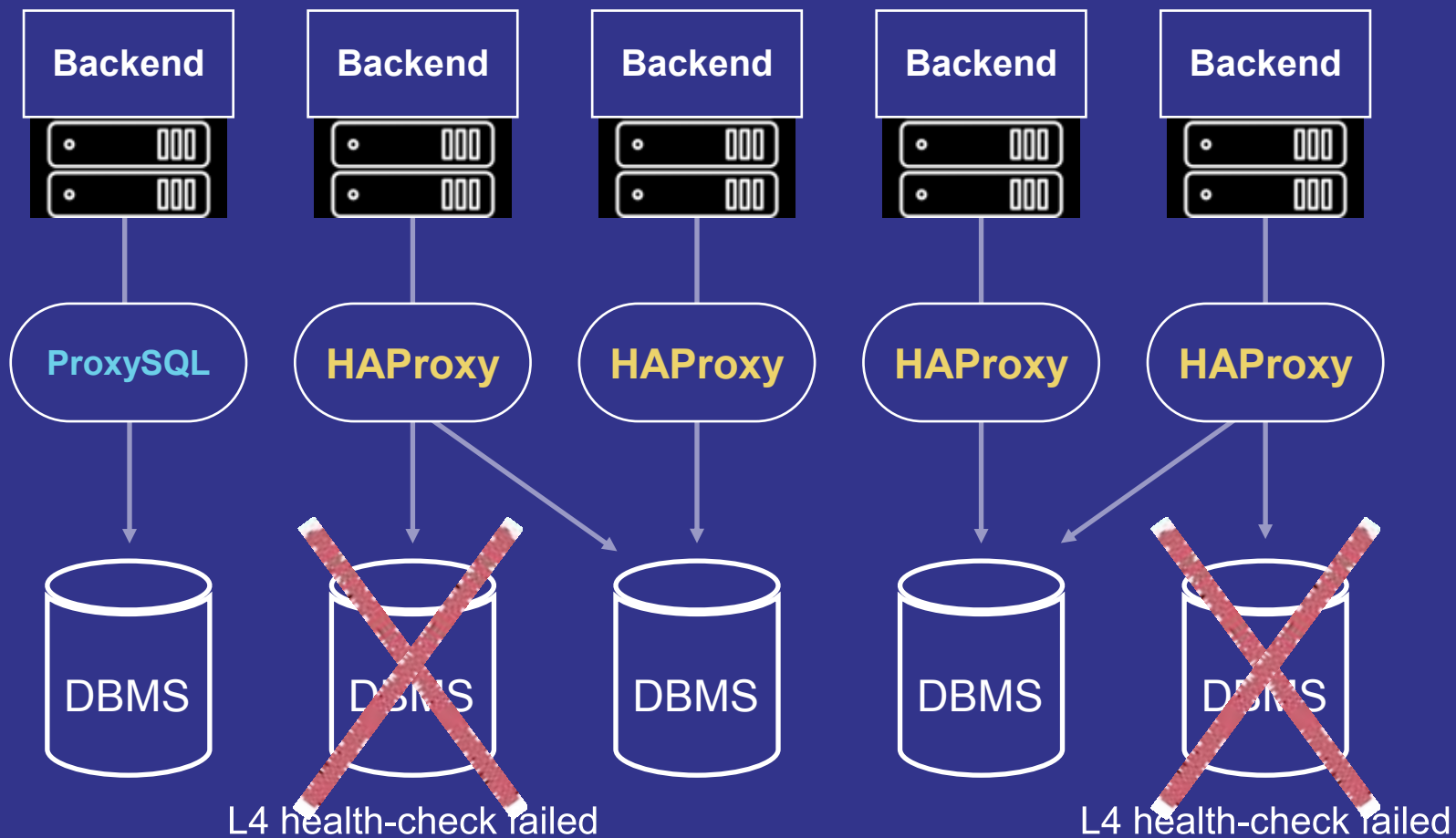
Keepalived



Балансировка, ограничения, защита



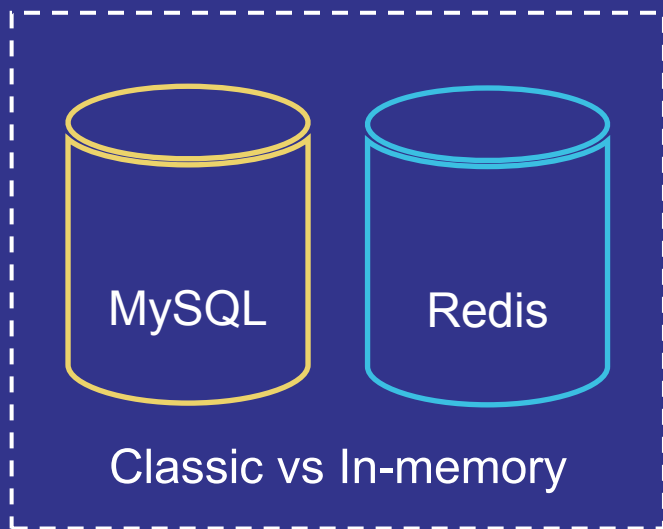
Легковесная балансировка



Управление



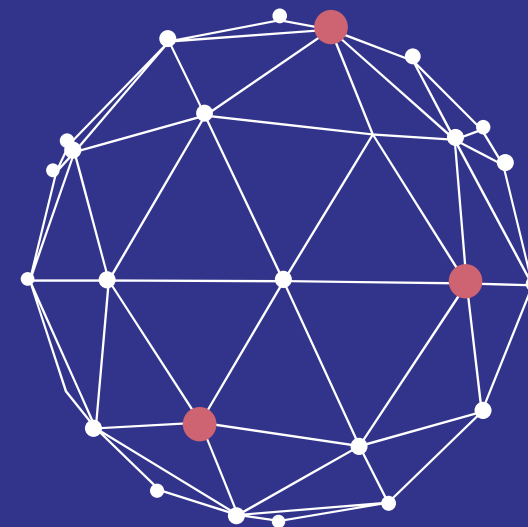
Хранение



Репликация - синхронизация копий объекта между узлами

Master-Master - можно читать и записывать на всех узлах

Master-Slave - можно записывать только на основном узле, но читать можно со всех





Ansible

Система управления конфигурациями

Infrastructure as Code (IaC)

```
1  ---
2  - name: Install chronyd
3    package:
4      name: chrony
5      state: latest
6
7  - name: Delete old config
8    file:
9      path: /etc/chronyd.conf
10     state: absent
11     notify: restart chronyd
12
13 - name: Create config for chronyd
14   template:
15     src: chronyd.conf
16     dest: /etc/chrony/chrony.conf
17     owner: root
18     group: root
19     mode: 0644
20     backup: yes
21     notify: restart chronyd
22
23 - name: Start and enable chronyd
24   systemd:
25     name: chronyd
26     state: started
27     enabled: yes
28     daemon_reload: yes
```

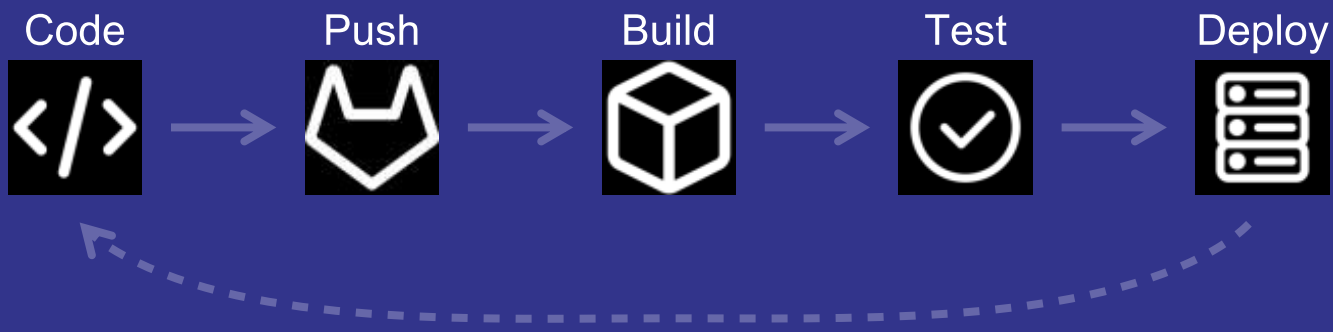


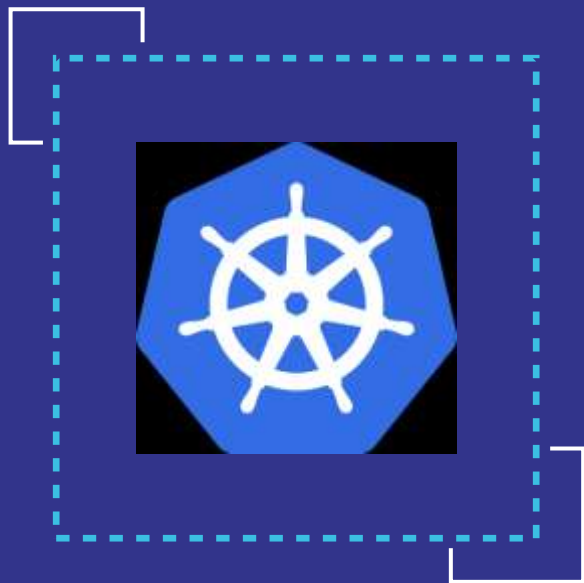
```
1  stages:
2    - build
3    - test
4
5  build-code-job:
6    stage: build
7    script:
8      - echo "Build our code"
9      - make
10
11 test-code-job1:
12   stage: test
13   script:
14     - echo "Test our code"
15     - go test
16
17 test-code-job2:
18   stage: test
19   script:
20     - echo "Test our code more"
21     - test.sh
22
```

.gitlab-ci.yml

CI/CD

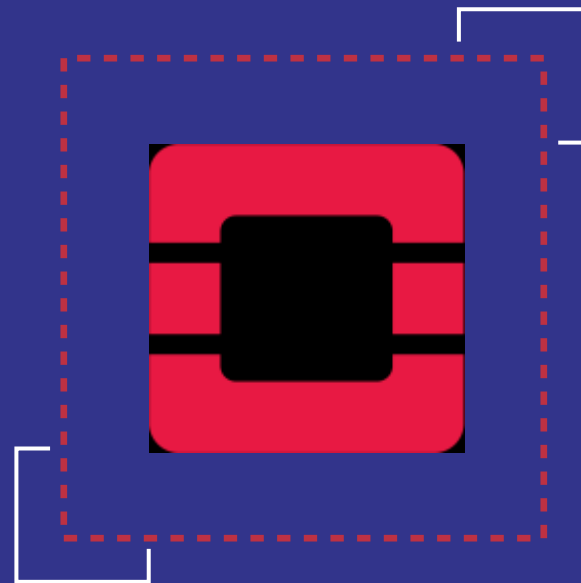
Непрерывная интеграция и непрерывное развертывание





Kubernetes

Платформа контейнеров



OpenStack

Платформа виртуальных машин

Всё еще достойны



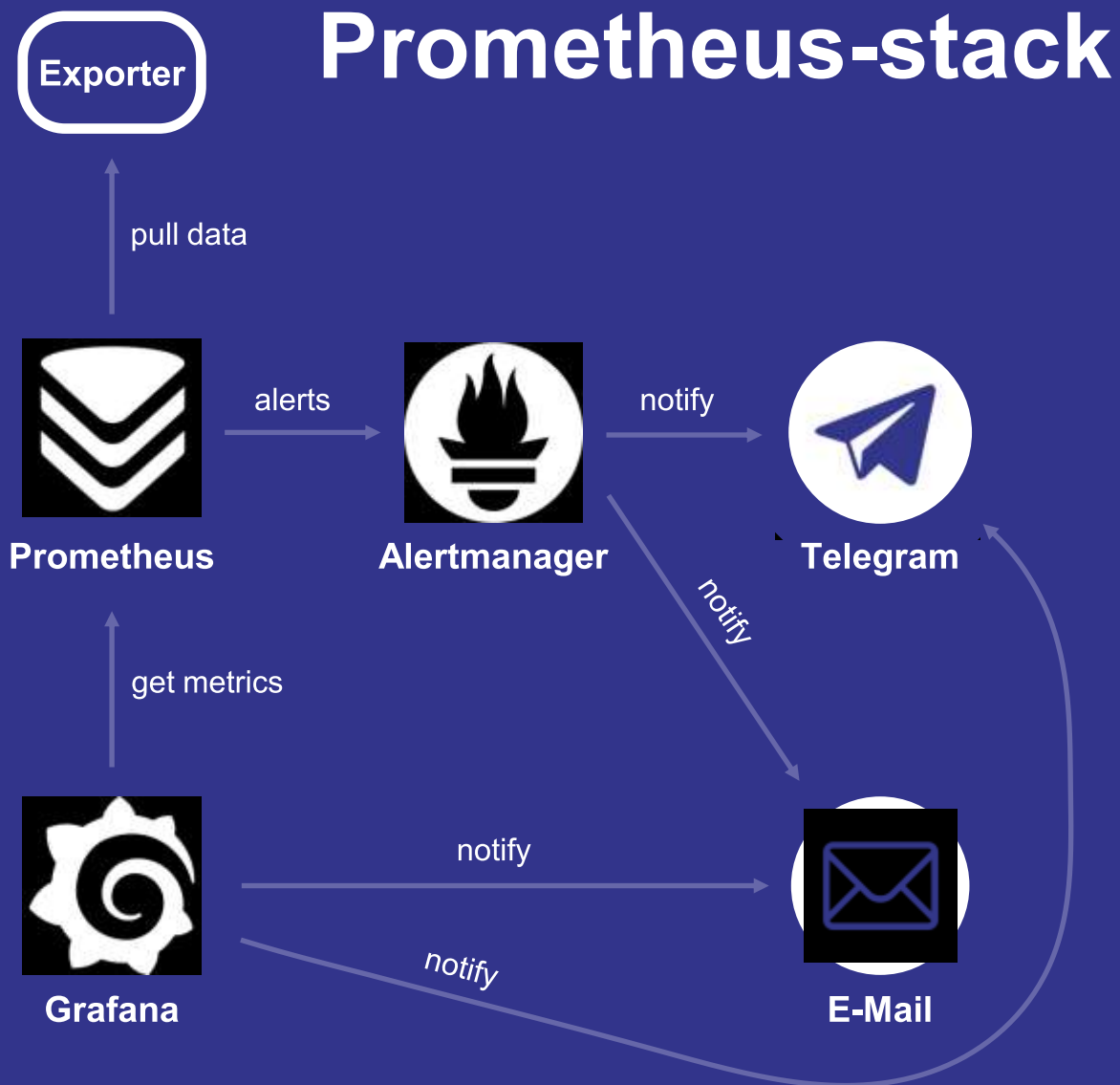
Системы мониторинга

Zabbix vs Prometheus

- + Многофункциональный
- Сложный (комплексный)
- Плохо масштабируется
- Устаревший интерфейс

- + Оптимизирован под ВР
- + Простой в эксплуатации
- + PromQL - отличный язык запросов
- + Легко масштабировать
- + Написан на Go
- + Prometheus-stack
- Нет встроенного интерфейса
- + Но есть Grafana
- Да, тут **слишком** много метрик

Prometheus-stack



Exporter - любая программа, которая отдает метрики
Prometheus - сердце системы, TSDB, собирает
Alertmanager - система управления оповещениями
Grafana - система построения графиков по метрикам



Алерты в Telegram

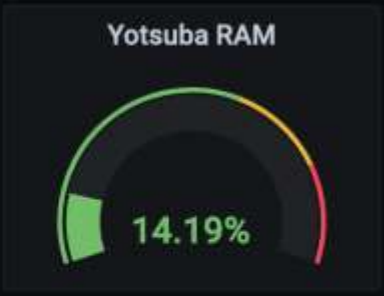
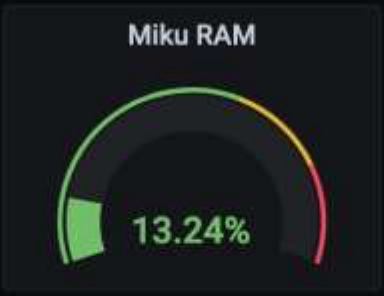
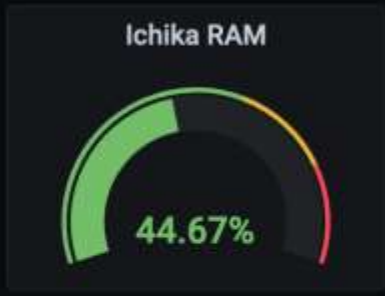


Raspberry Pi Temperatures



Interfaces status

miku:9100 eth0
nino:9100 eth0
asuna:9100 eth0
rikka:9100 eth0



Talk is cheap. Show me the code.

-- Linus Torvalds

Golang

```
1 package main
2
3 import (
4     "log"
5     "net/http"
6
7     "github.com/prometheus/client_golang/prometheus"
8     "github.com/prometheus/client_golang/prometheus/promhttp"
9 )
10
11 func main() {
12     // Register custom metric
13     usersOnline := prometheus.NewGauge(
14         prometheus.GaugeOpts{
15             Name: "users_online",
16         })
17     prometheus.MustRegister(usersOnline)
18
19     usersOnline.Set(float64(147)) // Set our gauge value
20
21     http.Handle("/metrics", promhttp.Handler())
22     log.Fatal(http.ListenAndServe(":8181", nil))
23 }
```

```
1 from prometheus_client import start_http_server, Gauge
2
3 g = Gauge('metrics_name', 'Description of gauge')
4 start_http_server(9191) # Handle prometheus requests
5
6 g.inc() # Increment by 1
7 g.dec(10) # Decrement by given value
8 g.set(4.2) # Set to a given value
```

Python

Формат отдаваемых prometheus-у метрик:

```
# HELP speed_test_max_latency Maximum latency in ms
# TYPE speed_test_max_latency gauge
speed_test_max_latency 7.212
# HELP speed_test_max_download Maximum download speed in bits/sec
# TYPE speed_test_max_download gauge
speed_test_max_download 5.553885017766232e+08
# HELP speed_test_max_upload Maximum upload speed in bits/sec
# TYPE speed_test_max_upload gauge
speed_test_max_upload 6.368826393779614e+08
```

Журналирование - Logging

Rsyslog - сбор и отправка логов

```
1  if $programname == 'razumator-api' then {
2      action(type="omfile"
3          |   fileOwner="razumator"
4          |   fileGroup="razumator"
5          |   dirOwner="razumator"
6          |   dirGroup="razumator"
7          |   dirCreateMode="0750"
8          |   FileCreateMode="0640"
9          |   File="/var/log/razumator/api.log")
10 }
11
12 if $programname == 'razumator-core' then {
13     action(type="omfile"
14         |   fileOwner="razumator"
15         |   fileGroup="razumator"
16         |   dirOwner="razumator"
17         |   dirGroup="razumator"
18         |   dirCreateMode="0750"
19         |   FileCreateMode="0640"
20         |   File="/var/log/razumator/core.log")
21 }
```

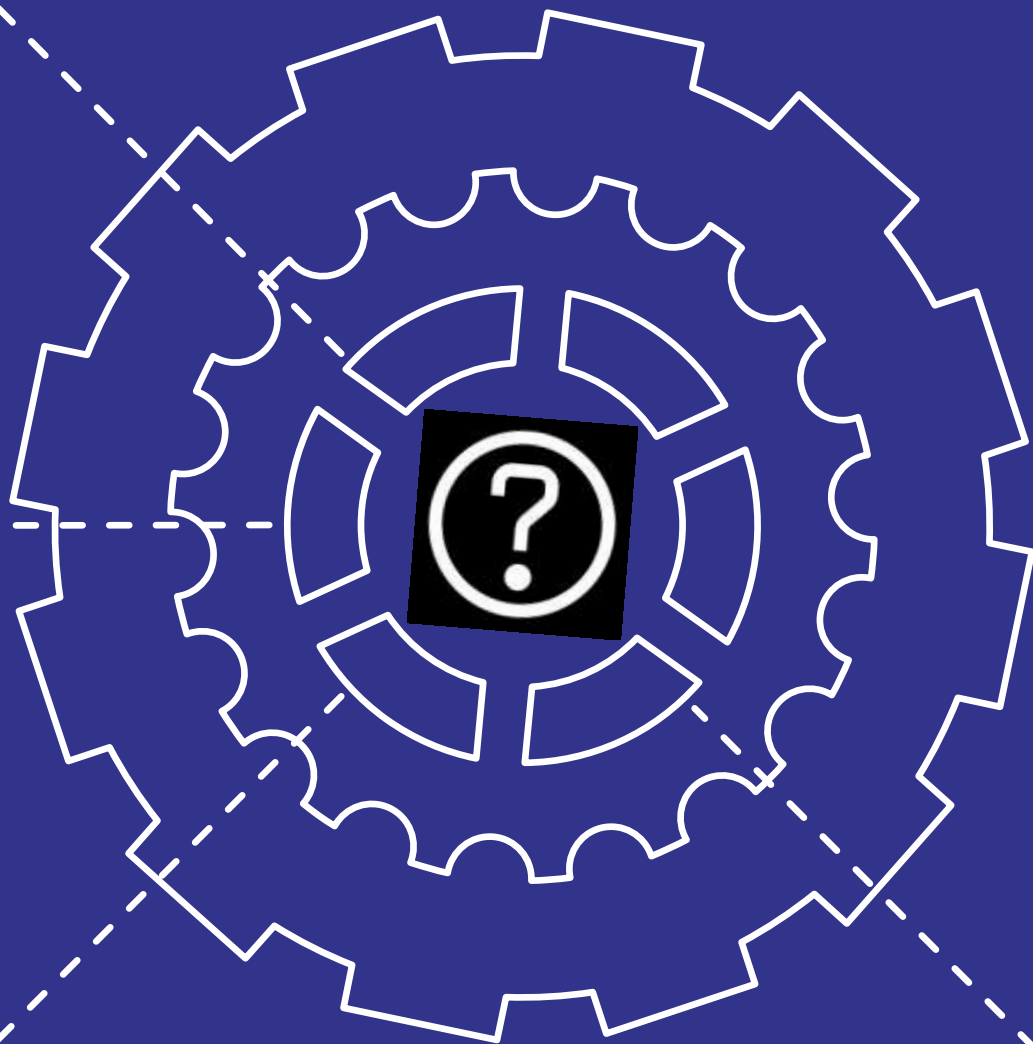
Logrotate - ротация логов

```
1  /var/log/razumator/*.log {
2      daily      # Ежедневно ротировать
3      rotate 7   # Хранить 7 версий
4      size 10M   # Максимальный размер
5      missingok  # Игнорировать отсутствие
6      compress   # Сжимать старые журналы
7      copytruncate # Обрезать, а не перемещать
8  }
```




ИСТОЧНИКИ

1. Бейер Б., Джоунс К., Петофф Д., Мёрфи Н. Site Reliability Engineering. Надежность и безотказность как в Google. – Спб.: Питер, 2019. – 592 с.: ил. - (Серия «Бестселлеры O`Reilly»).
2. Бейер Б., Рензин Д., Кавахара К., Торн С., Мёрфи Н. Site Reliability Workbook: практическое применение. - Спб.: Питер, 2021. – 544 с.: ил. - (Серия «Бестселлеры O`Reilly»).
3. Андрей Губа. Три дня, которые потрясли нас в 2013 [Электронный ресурс]: Хабр – веб-сайт. – Москва, 2015. – Режим доступа: <https://habr.com/ru/company/odnoklassniki/blog/268413/>, свободный. (дата обращения: 03.10.2021)
4. The .gitlab-ci.yml file [Электронный ресурс]: GitLab Docs – Электрон. документация. – Режим доступа: https://docs.gitlab.com/ee/ci/yaml/gitlab_ci_yaml.html, свободный. (дата обращения: 03.10.2021)
5. Клименко П. Краткое введение в etcd [Электронный ресурс]: Dots and Brackets. Размышления об эмиграции, жизни, и вообще - веб-сайт. - 2018. - Режим доступа: <https://dotsandbrackets.com/quick-intro-etcd-ru/>, свободный. (дата обращения: 03.10.2021)
6. Abdullin N. Schkn. Полное руководство по Prometheus в 2019 году [Электронный ресурс]: Хабр – веб-сайт, блог компании Southbridge. – Москва, 2019. – Режим доступа: <https://habr.com/ru/company/southbridge/blog/455290>, свободный. (дата обращения: 03.10.2021)
7. Алексеев А. Пишем метрики в Prometheus на языке Go [Электронный ресурс]: Записки программиста - веб-сайт. - 2019. – Режим доступа: <https://eax.me/golang-prometheus-metrics/>, свободный. (дата обращения: 03.10.2021)



Проектируй с умом!

Время для ваших вопросов...

Мои контакты:

Михаил Кучеренко
makucherenko@list.ru