

OTUS

Онлайн-образование

Не забыл включить запись

Меня хорошо видно && слышно?

Ставьте плюсы, если все хорошо
Напишите в чат, если есть проблемы

Правила вебинара

- Активно участвуем
- Задаем вопросы в чат или голосом
- Off-topic обсуждаем в Slack #канал группы или #general
- Вопросы вижу в чате, могу ответить не сразу

План

- Задачи мониторинга
- Типы мониторинга
- Метрики
- Хранение данных
- Сбор данных
- Алертинг
- Инструменты (Zabbix, NetData, etc...)

Ожидание от мониторинга

- Алертинг о сбоях в системе => Сбор информации о текущем состоянии
- Предупреждения о возможных сбоях и проблемах
- Отражение состояния системы/сервера/сервиса/компонента сервиса
- Сбор статистики и визуализация
- Агрегация и группировка
- Предоставление информации для мониторинга “второго порядка”
- Отчеты
- Дашборды

Типы мониторинга

WHITEBOX

BLACKBOX

Whitebox

Мониторинг, базирующийся на метриках, которое дает само приложение/сервис:

- через логи
- интерфейсы
- API
- SNMP

Blackbox

Мониторинг, оценивающий внешнее состояние сервиса/системы:

- ping
- HTTP request
- Открытый порт
- Наличие процессов

Метрики

Основные метрики системы:

- LoadAvg, CPU, Net (bps/pps), DISK Load
- потребление Mem/DISK
- “чистота” системных логов: dmesg, messages
- Актуальность состояния резервных копий

Метрики

Метрики процесса

- Наличие процесса и правильное количество этих процессов
- Открытый сокет/порт процесса
- возможность получить статус процесса (где применимо)
- некоторые параметры статуса

Метрики

Метрики сервиса

- “диагностический запрос”, который задействует все или большинство компонентов системы.
- время отклика/обработки запроса
- количество обращений в единицу времени
- количество одновременных обращений

Исторические данные (TimeSeries)

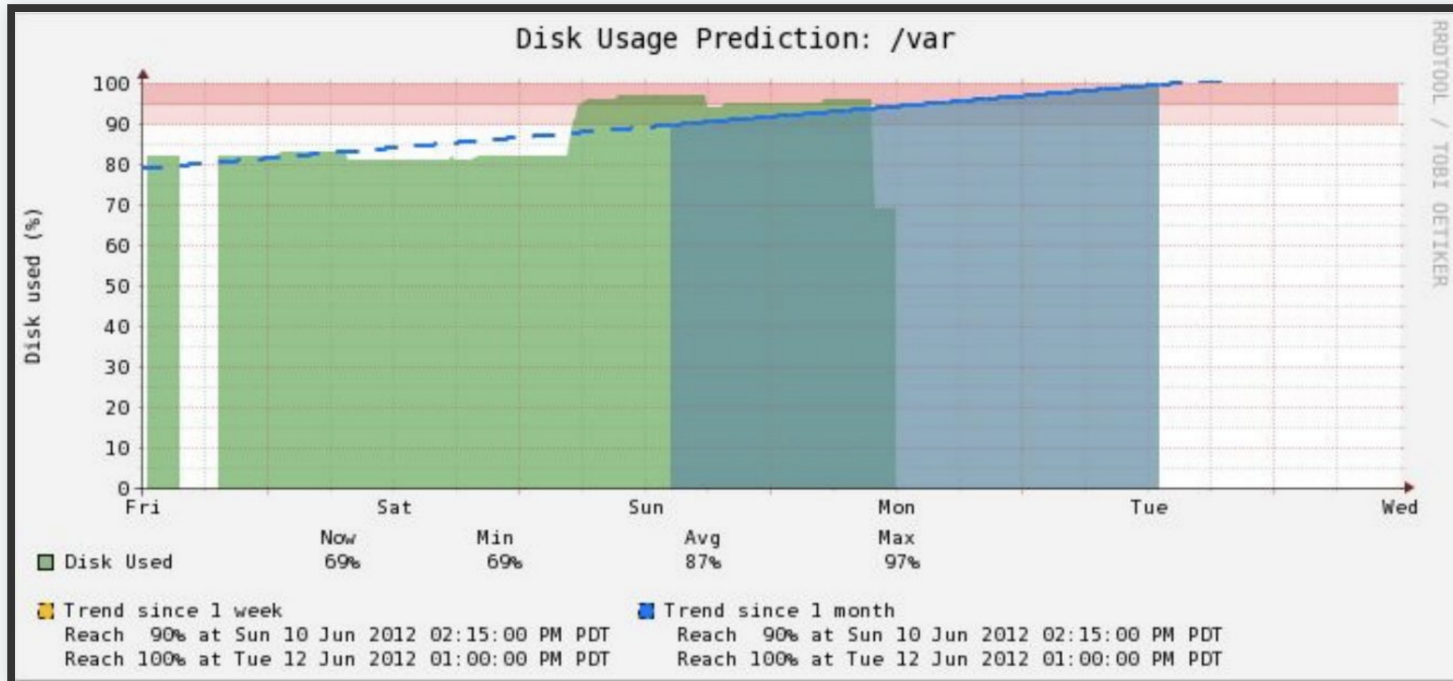
С историческими данными возникает несколько вопросов:

1. Как их собирать?
2. Где их хранить?
3. Сколько их хранить?
4. Как с ними работать?

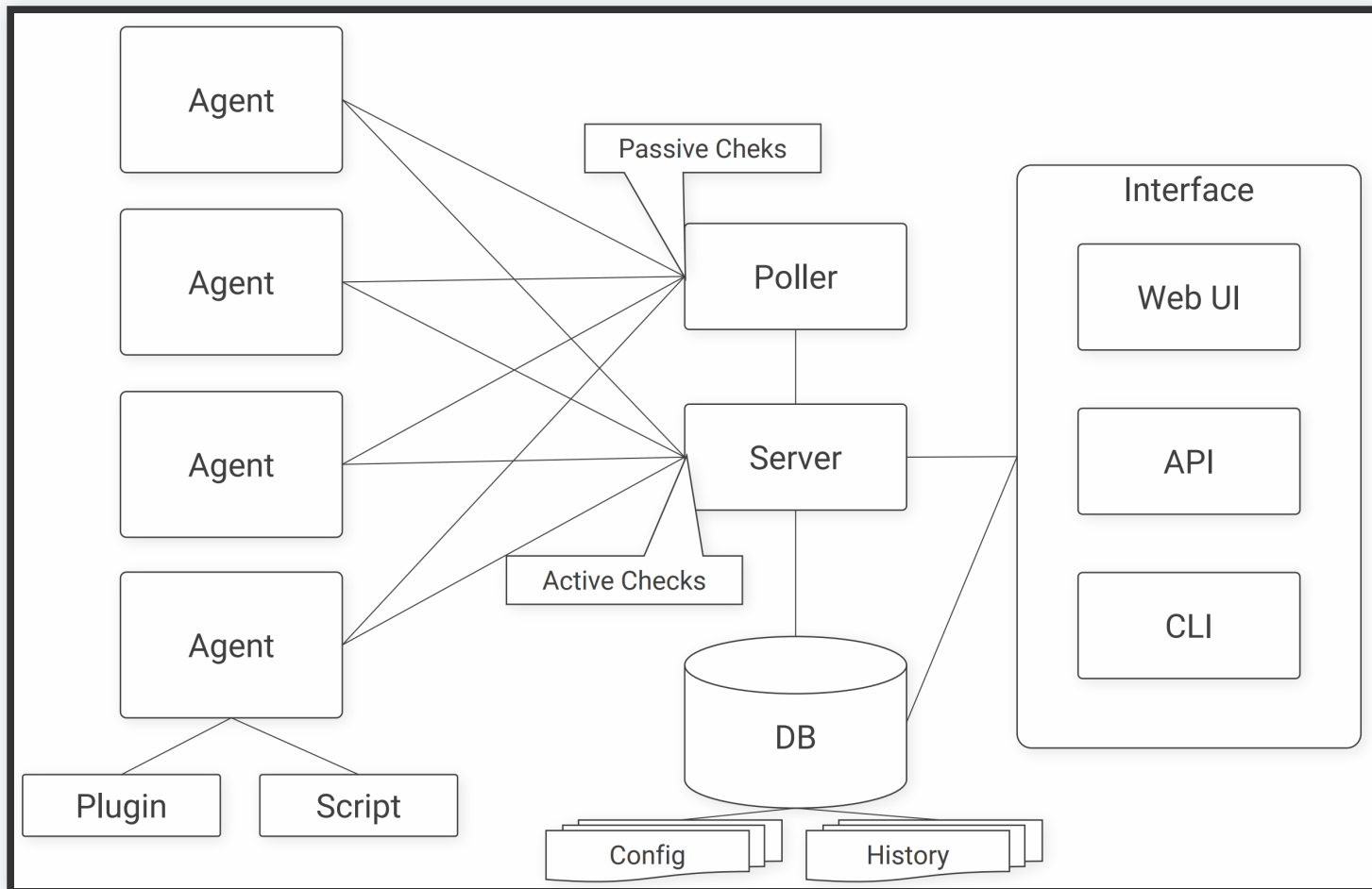
Для чего нужны исторические данные

1. Анализ проблем
2. Гадание на графиках или Trend monitoring. Тут высокое разрешение не важно, т.к. временные рамки в которых идет “гадание” это недели, месяца и годы.

Пример исторических данных



Основные компоненты мониторинга



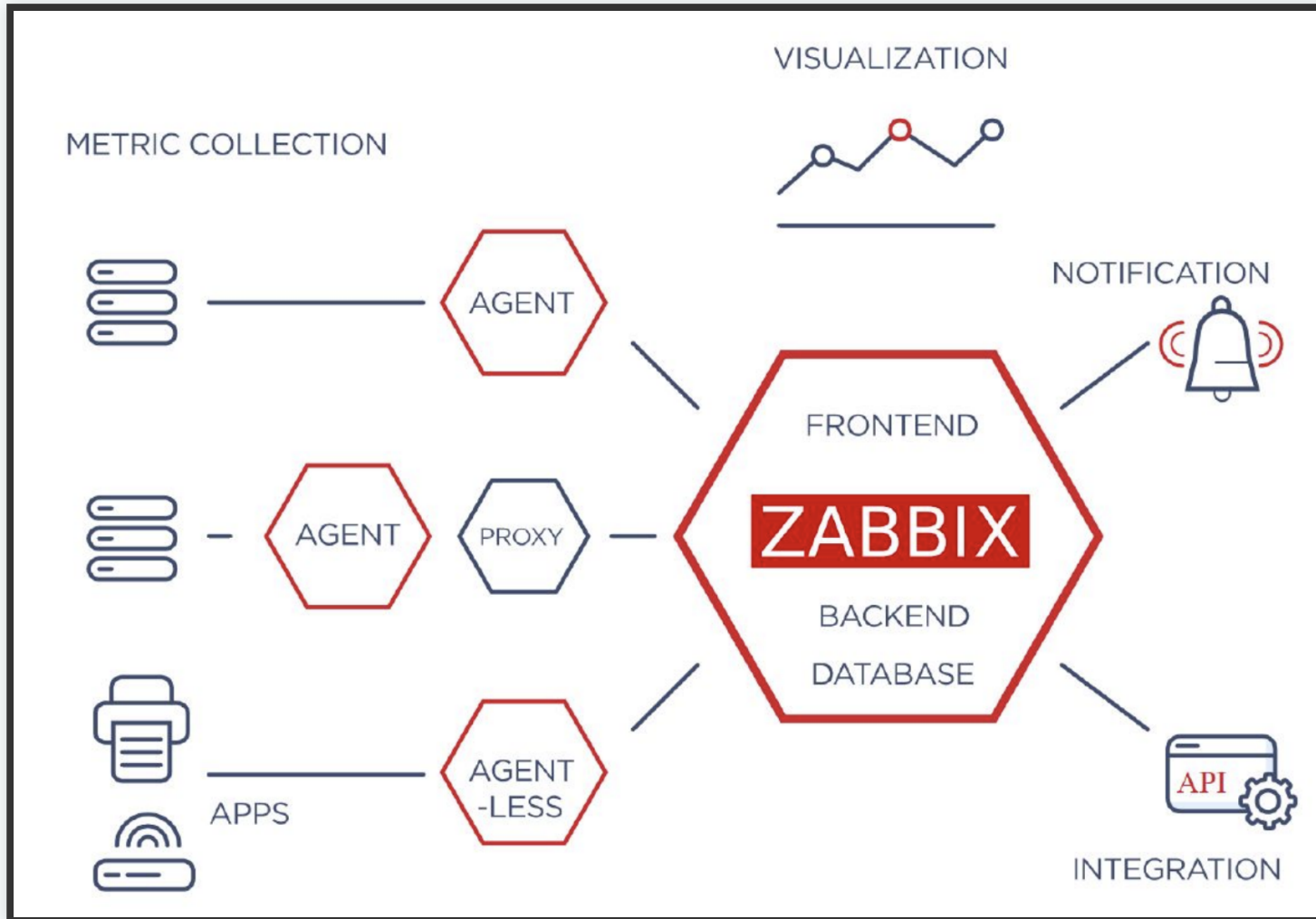
Инструменты мониторинга

- Zabbix - свой агент, свой веб, SQL database
- nagios/icinga - свой агент, свой веб, SQL database
- Graphite - carbon (агент), whisper (tsdb), graphite-web
- cacti, ganglia, collectd, mrtg, munin - база rrdtool
- Netdata - агент и веб
- Prometheus - node_exporter (агент), prometheus (база, веб), alertmanager
- InfluxData - telegraph (агент), influxdb, chronograph (web), kapacitor (alerts)
- StatsD - набор инструментов для сбора метрик
- Grafana - интерфейс для дашбордов и алертинга

Zabbix

ZABBIX

Zabbix



Zabbix взаимодействие с узлами

- Zabbix Agent
 - Собственно агент устанавливаемый на целевые хосты
- SNMP
 - Протокол использующийся для управления сетевыми устройствами
- JMX
 - Мониторинг JMX можно использовать для наблюдения за счетчиками JMX в Java приложениях.
- IPMI
 - интеллектуальный интерфейс управления платформой. В контексте Zabbix используется для мониторинга аппаратной платформы серверов

Zabbix Узлы сети и группы

- Старайтесь группировать хосты по типам и задачам (хост может быть включен в любое кол-во групп)
- Давать понятные имена для хостов
- Впоследствии разделение прав на просмотр и управление

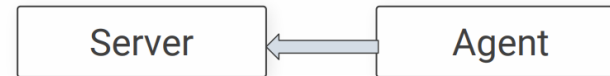
Zabbix Типы элементов данных

Опрашиваемые



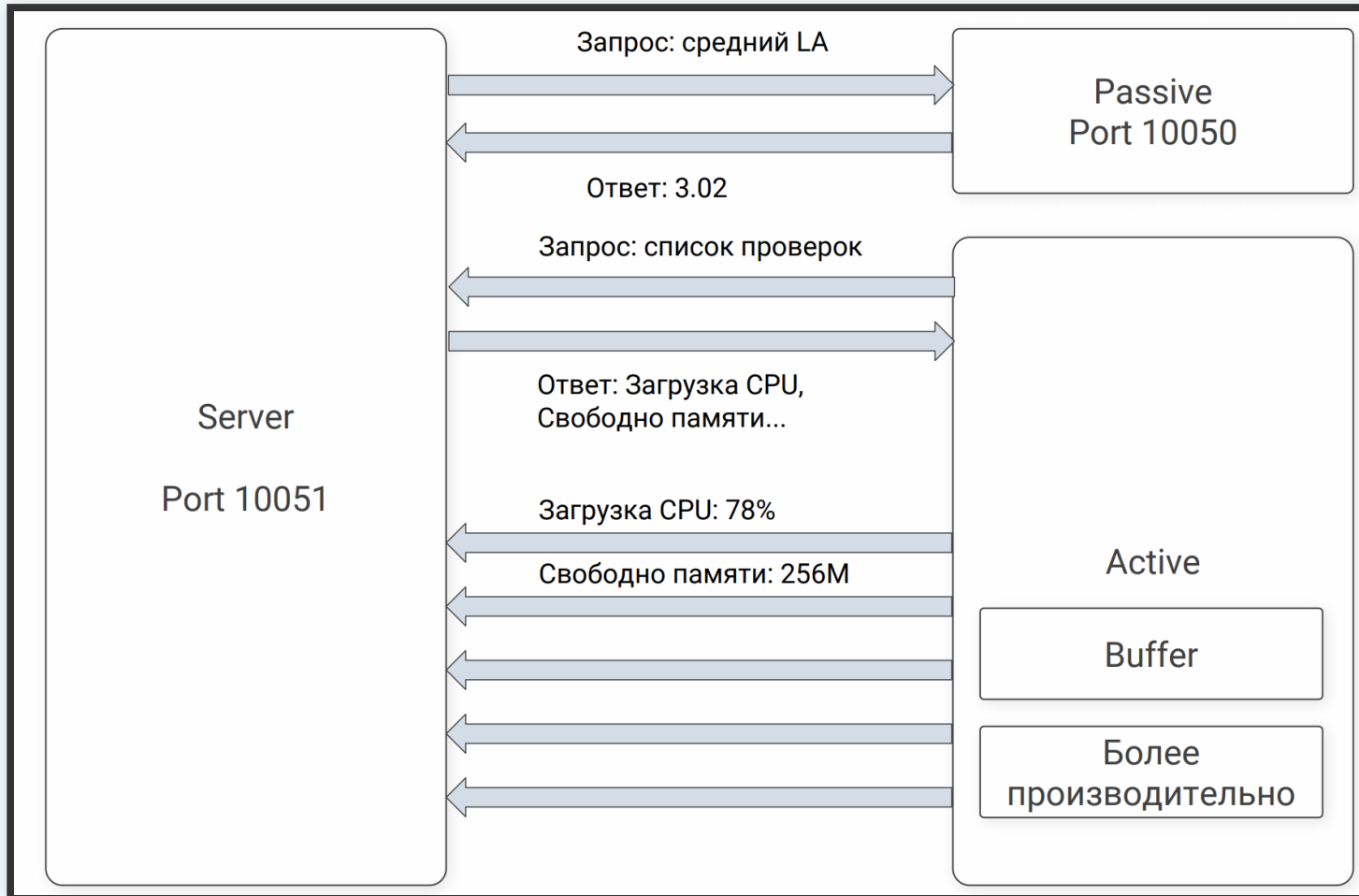
- Zabbix agent
- Безагентные проверки (simple checks)
- SNMP агент
- IPMI
- JMX
- HTTP
- SSH
- Telnet
- Базы данных

Трапы



- Zabbix agent (активный)
- SNMP трапы
- Zabbix траппер

Zabbix Active/Passive checks



Zabbix Триггеры

- Понятное имя триггера
- Устанавливайте важность триггера
- Синтаксис:
 - `{host:key.function(param)}=0`
 - `vm.memory.size[available].last(0)}<20M`
- Тестируйте триггеры (встроенная функция в Zabbix)

Zabbix оповещения

- Mail
- SMS
- SIP
- Jabber
- Slack
- Telegram
- Discord

Zabbix Хранение истории

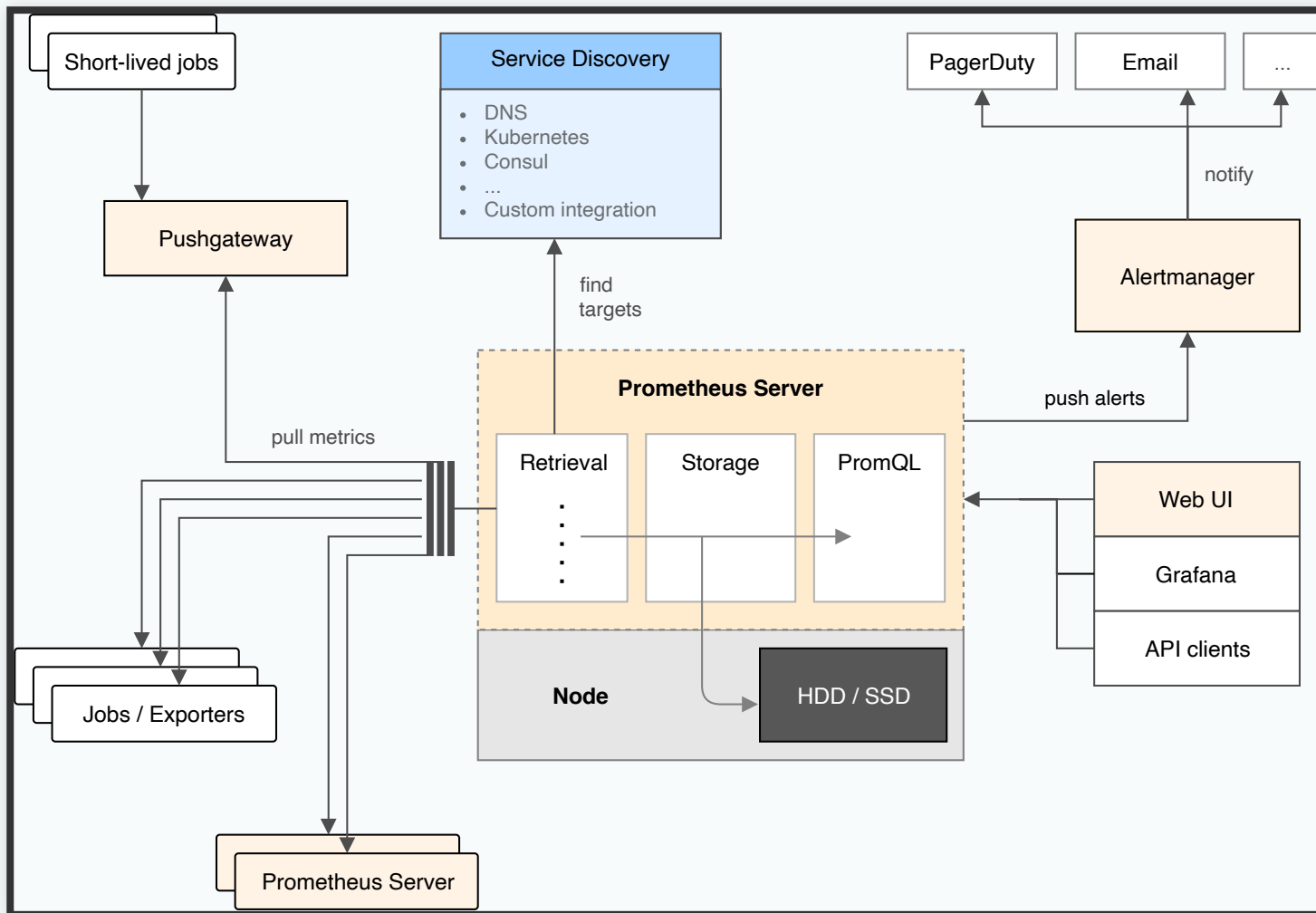
Вы можете указать как много дней историй̄ будет храниться:

- в диалоге свойств элемента данных
- при массовом обновлении элементов данных
- при настройке задач очистки истории Любые более старые данные будут удалены с помощью автоматической очистки базы данных (Housekeeper).

Prometheus



Prometheus архитектура



Prometheus конфигурация

PROMETHEUS.YML

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s

- job_name: 'prometheus'
  scrape_interval: 5s
  scrape_timeout: 5s
  static_configs:
    - targets: ['localhost:9090']

- job_name: node-exporter
  static_configs:
    - targets: ['localhost:9100']
```

Prometheus exporters

Сбор всех данных осуществляется за счет использования различных экспортеров.

Подробный список [тут](#)

Prometheus Хранение данных

- Prometheus TSDB
- Victoria Metrics
- Thanos
- Click House

Prometheus Alerting

RULES.YML

```
groups:
- name: example
  rules:

  # Alert for any instance that is unreachable for >5 minutes.
  - alert: InstanceDown
    expr: up == 0
    for: 5m
    labels:
      severity: page
    annotations:
      summary: "Instance {{ $labels.instance }} down"
      description: "{{ $labels.instance }} of job {{ $labels.job }} has been down for more than 5 minutes."

  # Alert for any instance that has a median request latency >1s.
  - alert: APIHighRequestLatency
    expr: api_http_request_latencies_second{quantile="0.5"} > 1
    for: 10m
    annotations:
      summary: "High request latency on {{ $labels.instance }}"
      description: "{{ $labels.instance }} has a median request latency above 1s (current value: {{ $value }}s)"
```

Prometheus Alertmanager

PROMETHEUS.YML

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s

rule_files:
- rules.yml

- job_name: 'prometheus'
  scrape_interval: 5s
  scrape_timeout: 5s
  static_configs:
    - targets: ['localhost:9090']

- job_name: node-exporter
  static_configs:
    - targets: ['localhost:9100']

# Alerting specifies settings related to the Alertmanager
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        # Alertmanager's default port is 9093
        - localhost:9093
```

Prometheus Alertmanager

ALERTMANAGER.YML

```
global:
  resolve_timeout: 1m
  slack_api_url: 'https://hooks.slack.com/services/TSUJTM1HQ/123123'

route:
  receiver: 'slack-notifications'

receivers:
- name: 'slack-notifications'
  slack_configs:
  - channel: '#monitoring-instances'
    send_resolved: true
```

Netdata

netdata



Statping

statping

Librenms



LibreNMS

Grafana

