

# Prometheus

Или когда надо было вчера

# Зачем нужен мониторинг?

- Модно рисовать какие-то графики
- Начальник обещал уволить
- Мазохизм

# Prometheus - эстетика античности



# И зачем нам это?

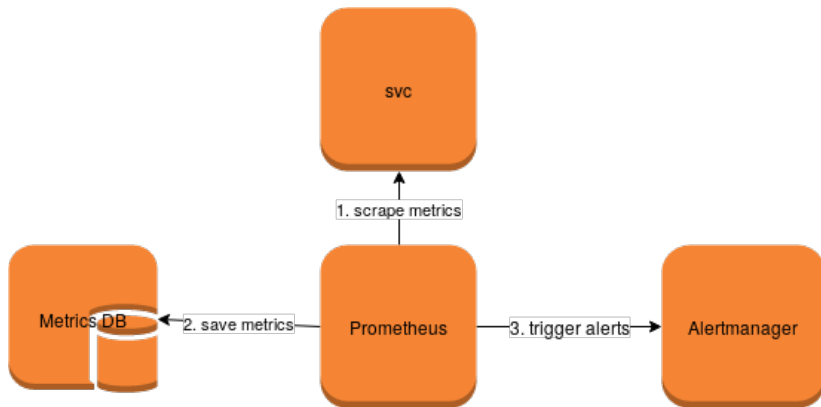
- Полное отсутствие авторизации
- Исключительно pull-метрики
- Интеграция с service discovery
- CNCF incubator

Поддержка проектов, которые имеют ряд признаков

- Open source стэк технологий
- Нативная поддержка контейнеризации
- Поддержка оркестрации
- Microservice orientation

- Широкая поддержка community
- Легкая интеграция между проектами
- Де-факто стандарт индустрии

# Prometheus в картинках



- 1. scrape metrics
- 2. save metrics
- 3. trigger alerts



# 1. scrape metrics

GET `http://service:port/metrics`

- Сервис дискавери облачных провайдеров
- Сервис дискавери Kubernetes
- Сервис дискавери Consul
- Статическая конфигурация

```
scrape_configs:
  - job_name: 'kubernetes-apiservers'
    kubernetes_sd_configs:
      - role: endpoints
    scheme: https
    tls_config:
      ca_file: .../serviceaccount/ca.crt
      insecure_skip_verify: true
    bearer_token_file: .../serviceaccount/token
```

# Real world scrape result

## kubernetes-nodes (12/12 up) [show less](#)

Endpoint	State	Labels	Last Scrape
<a href="https://kubernetes.default.svc:443/api/v1/nodes/ip-10-0-197-214.ec2.internal/proxy/metrics/cadvisor">https://kubernetes.default.svc:443/api/v1/nodes/ip-10-0-197-214.ec2.internal/proxy/metrics/cadvisor</a>	UP	<pre>beta_kubernetes_io_arch="amd64" beta_kubernetes_io_instance_type="m3.large" beta_kubernetes_io_os="linux" failure_domain_beta_kubernetes_io_region="us-east-1" failure_domain_beta_kubernetes_io_zone="us-east-1a" instance="ip-10-0-197-214.ec2.internal" kubernetes_io_hostname="ip-10-0-197-214.ec2.internal" kubernetes_io_role="node" node_role_kubernetes_io_node=""</pre>	2.376s ago
<a href="https://kubernetes.default.svc:443/api/v1/nodes/ip-10-0-198-39.ec2.internal/proxy/metrics/cadvisor">https://kubernetes.default.svc:443/api/v1/nodes/ip-10-0-198-39.ec2.internal/proxy/metrics/cadvisor</a>	UP	<pre>beta_kubernetes_io_arch="amd64" beta_kubernetes_io_instance_type="m3.large" beta_kubernetes_io_os="linux" failure_domain_beta_kubernetes_io_region="us-east-1" failure_domain_beta_kubernetes_io_zone="us-east-1a" instance="ip-10-0-198-39.ec2.internal" kubernetes_io_hostname="ip-10-0-198-39.ec2.internal" kubernetes_io_role="node" node_role_kubernetes_io_node=""</pre>	1.121s ago
<a href="https://kubernetes.default.svc:443/api/v1/nodes/ip-10-0-200-6.ec2.internal/proxy/metrics/cadvisor">https://kubernetes.default.svc:443/api/v1/nodes/ip-10-0-200-6.ec2.internal/proxy/metrics/cadvisor</a>	UP	<pre>beta_kubernetes_io_arch="amd64" beta_kubernetes_io_instance_type="m3.medium" beta_kubernetes_io_os="linux" failure_domain_beta_kubernetes_io_region="us-east-1" failure_domain_beta_kubernetes_io_zone="us-east-1a" instance="ip-10-0-200-6.ec2.internal" kubernetes_io_hostname="ip-10-0-200-6.ec2.internal" kubernetes_io_role="master" node_role_kubernetes_io_master=""</pre>	3.894s ago

## 2. save metrics

- On-disk
- InfluxDB
- PostgreSQL
- SignalFx
- Custom adapter

### 3. trigger alerts

- Алерт прометея показывает текущую ситуацию
- Основаны на PromQL
- Не алертинг в полном смысле слова
- Для группировки событий - alertmanager

```
groups:
- name: example
  rules:
- alert: InstanceDown
  expr: up == 0
  for: 5m
  labels:
    component: hardware
    severity: error
  annotations:
    summary: "Instance {{ $labels.instance }} down"
    description: "{{ $labels.instance }} is down."
```

# Ну а где же Alertmanager?

- Статический адрес
- Сервис дискавери

prometheus.yml:

```
alerting:
  alertmanagers:
    - consul_sd_configs:
      - server: consul.dev:8500
        datacenter: us-east-1
      services:
        - alertmanager
```

- Группировка алертов
- Временное отключение алертов
- Доставка получателям
- Правила эскалации



# Alertmanager is fine

Alertmanager Alerts Silences Status

Filter

Group

Custom matcher, e.g. `env="production"`

No alerts found

```
receivers:  
- name: default-receiver  
  slack_configs:  
    - channel: '#dev-env'  
      title: Prometheus {{ .CommonLabels.component }}  
      send_resolved: false  
      api_url: https://hooks.slack.com/services/EXAMPLE  
      text: "{{ .CommonAnnotations.description }}"  
route:  
  receiver: default-receiver  
  group_by: [component, severity]
```



**AlertManager** APP 20:35

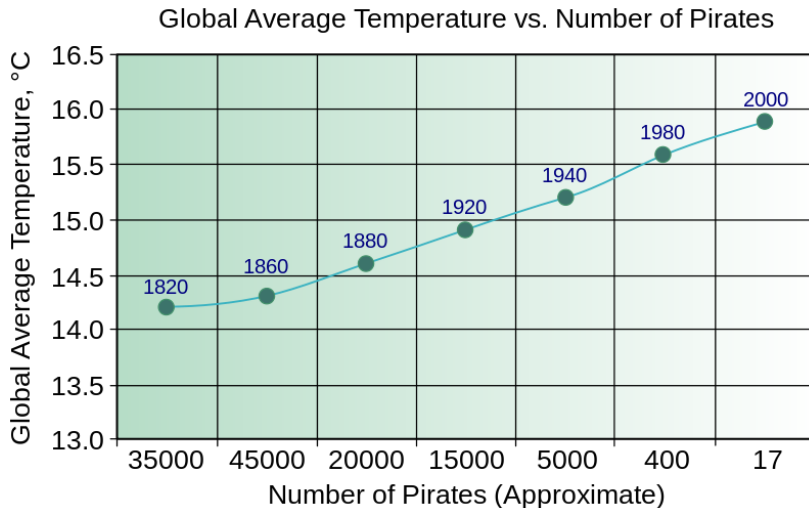
**Prometheus Redshift info**

@channel

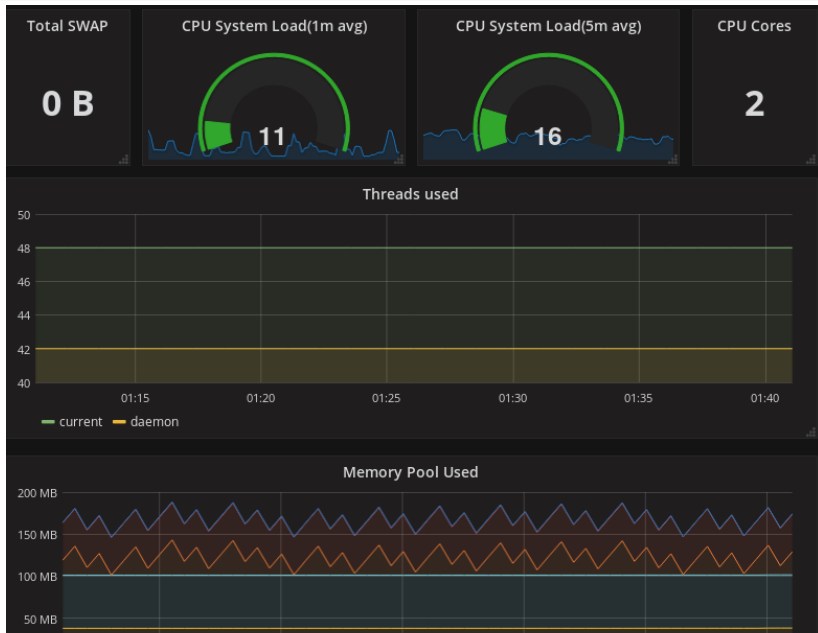
summary: Redshift health Norm

description: Redshift at us-abrau-prod: health status is HEALTHY

# Нарисуем какой-то график

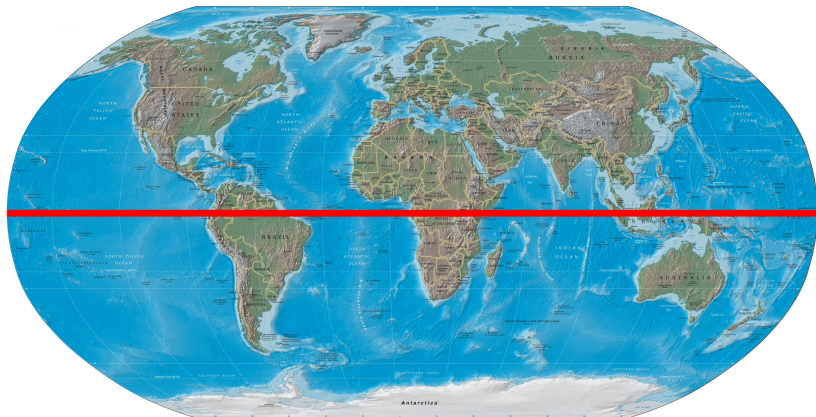


# Нет, другой

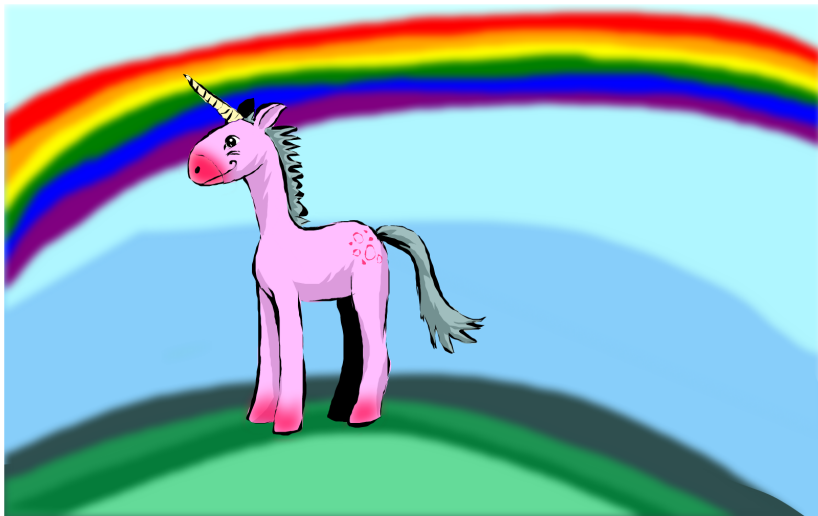


- Prometheus сам находит нужный сервис
- С помощью HTTP получает метрики
- Сохраняет их в хранилище
- По правилам формирует алерты
- Информировать алертменеджер об алертах
- Последний решает кого и как оповестить

# Скучная часть позади!

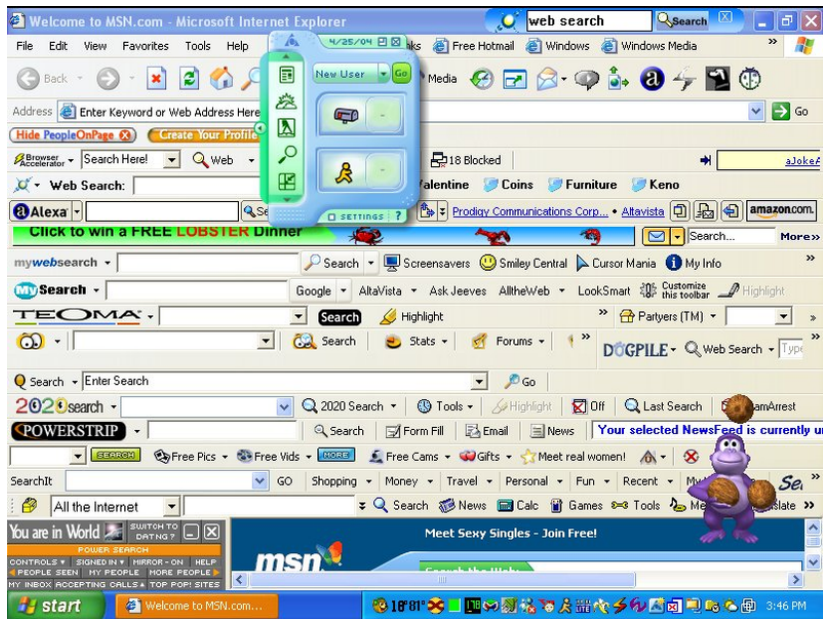


# Как обычно говорят на докладах о своём

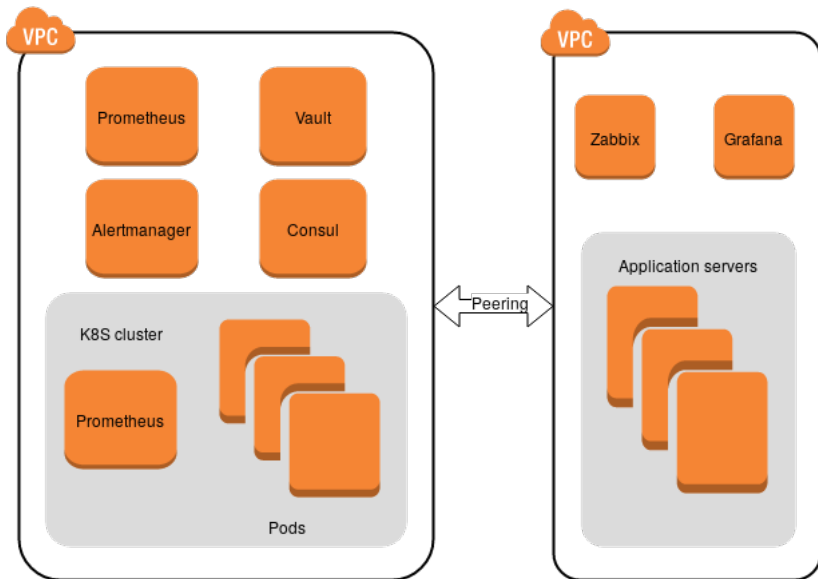




# Как на самом деле



# Самая честная схема



# Почему не оставить Zabbix?

- Нет нативной интеграции с Kubernetes
- Сложность настройки новичку
- Push-метрики

**fasten**