

Василков Арсений Николаевич

# «Firebird в Kubernetes: от прототипа до внедрения»

# Обо мне

Студент Университета “Сириус”

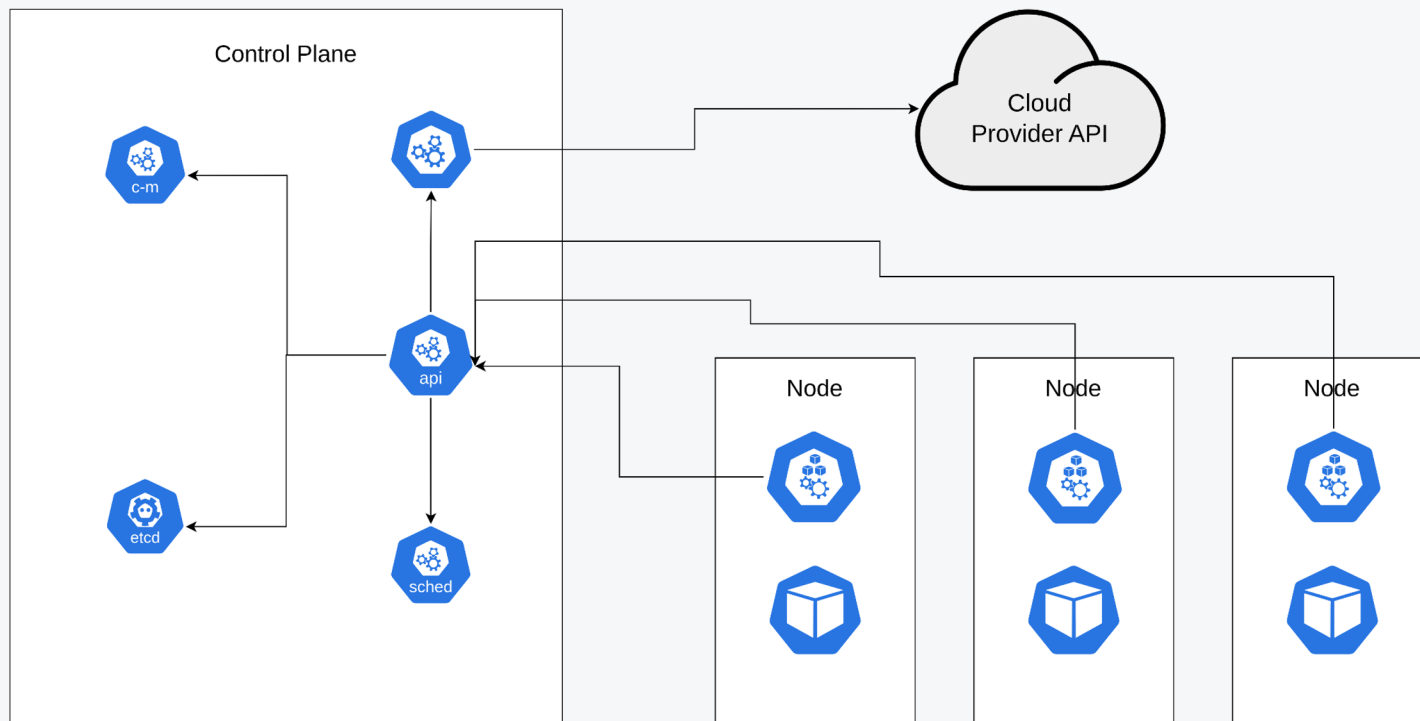
3 курс, Компьютерные системы  
и комплексы

- Участник Ред Проекта 2025
- Участник Ред Проекта 2026

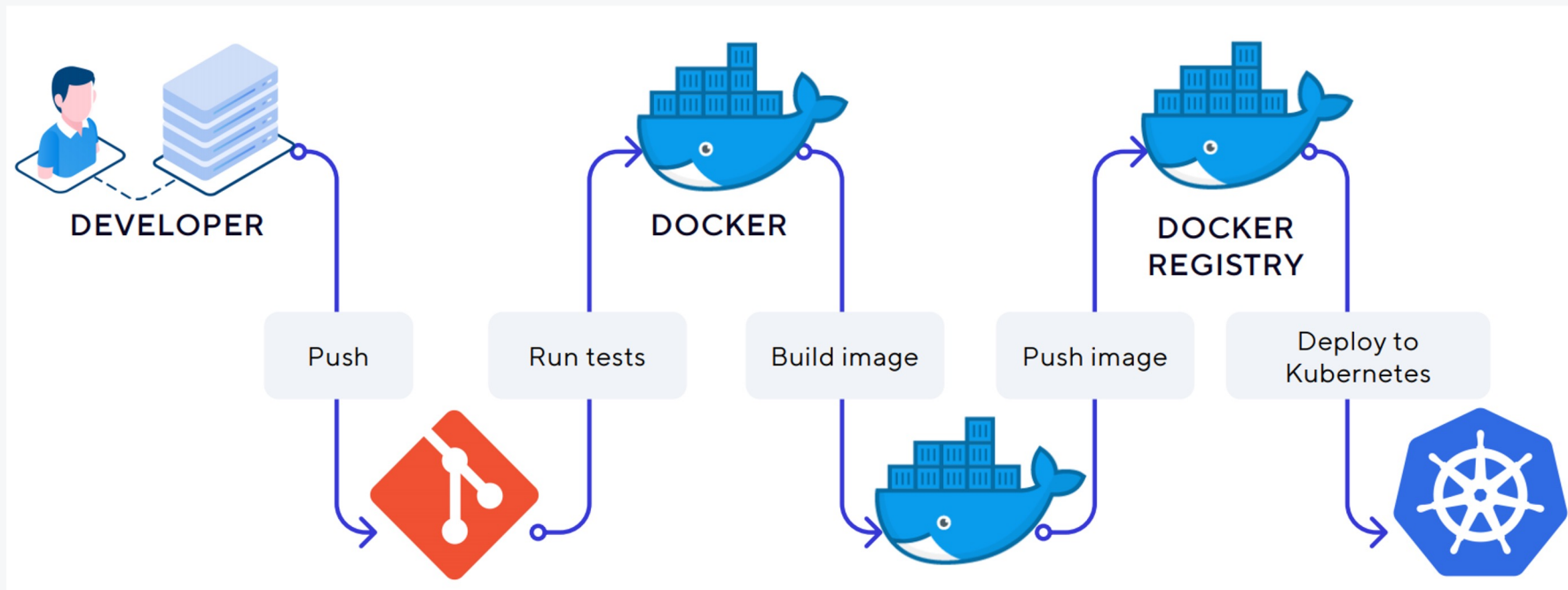
DevOps инженер  
в ООО “Бинетика”



# Kubernetes?



# Как происходит деплой



# Цель и проблематика

## Проблематика

Firebird в 2026 году живёт в мире Kubernetes, но без референсных практик:

Официальная документация Firebird не содержит рекомендаций по развертыванию в k8s — каждый админ изобретает велосипед.

## Цель

Создать воспроизводимый референсный паттерн развёртывания Firebird Server в Kubernetes. И дать готовое решение из коробки, которое можно развернуть

за 20 минут и получить:

1. Автоматическое управление экземплярами Firebird
2. Мониторинг из коробки (Prometheus + Grafana)
3. Автоматическую миграцию при сбоях

# Почему firebird?

## 1. Один файл – один PVC, не нужен оператор

```
@lendy kubectl exec -it data-base-0 -n firebird -- ls -lh /var/lib/firebird/data/  
  
total 1.9M  
-rw-rw----. 1 root root 1.9M Apr 18 07:08 databases.fdb
```

## 2. Рестарт без последствий

## 3. Легкий образ – быстрый failover

# Когда Kubernetes нужен?

## Много инстансов

Десятки баз по 10–100 ГБ  
(отделы, арендаторы,  
проекты).

Dev/stage/prod окружения,  
CI/CD стенды.

## Масштабирование

Гибкое масштабирование  
сервисов.

Создать единую точку  
контроля над всей  
инфраструктурой.

## Интеграция с новыми технологиями

GitOps, единый мониторинг,  
централизованные бэкапы.

Внедрение дополнительных  
плагинов.

# Возможности firebird в кубе

## Масштабирование



Один StatefulSet-шаблон  
– десятки инстансов  
Firebird. Новое окружение  
dev/stage/prod  
поднимается за минуты,  
а не часы.

## Миграция



При падении ноды  
Kubernetes автоматически  
переносит базу на другую  
ноду. Данные сохраняются  
– ручного вмешательства  
не требуется.

## Управление



Единая точка контроля  
для всех инстансов  
Firebird: мониторинг,  
обновления, бэкапы  
и секреты – всё через  
один инструмент.

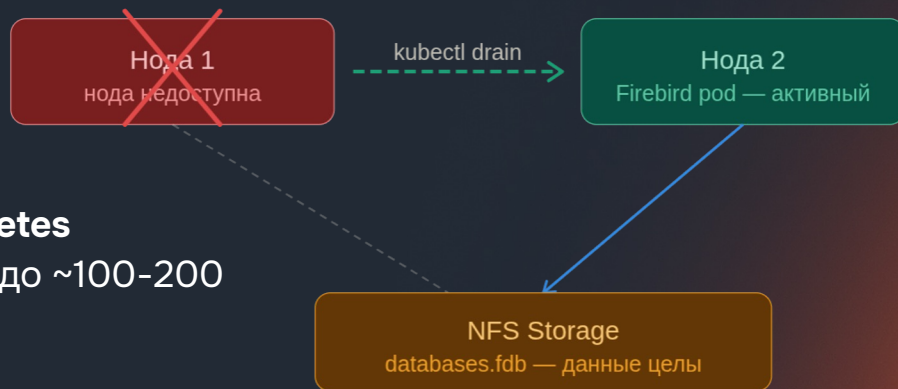
# Миграция

Для входа:

**NFS** как сетевое хранилище и **Kubernetes StatefulSet** как оркестратор пода с базой. (до ~100-200 ГБ комфортно)

Для продакшен разработки:

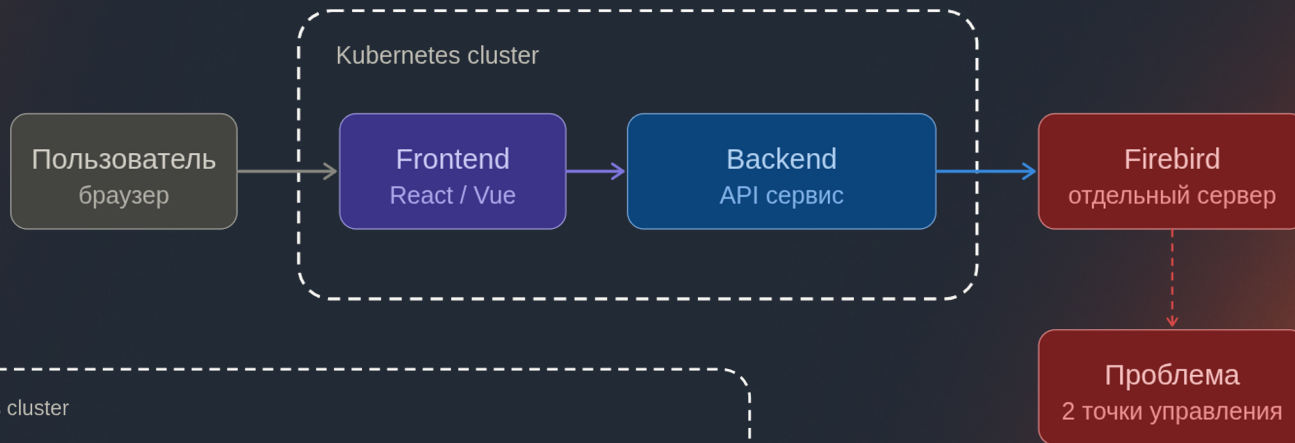
1. Longhorn - небольшие кластеры, on-premise, простая настройка, (до 1-2 ТБ)
2. Rook / Ceph - крупные кластеры, максимальная надёжность, (петабайтный масштаб)
3. AWS EFS / GCE Filestore - облачные кластеры, управляемый сервис, (практически без лимита)



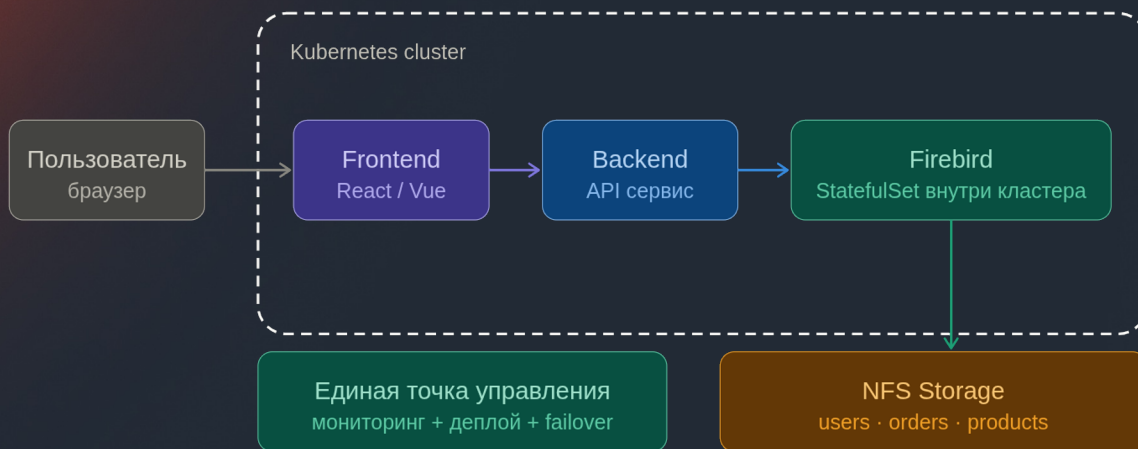
# Прикладное использование



# Как было



# Как хотим



# Исходные данные

1. **Репозиторий** GitHub, репозиторий с готовыми манифестами и конфигурами для работы – StatefulSet, PVC, Service, Secret, NFS конфигури.
  2. **Кластер** Kubernetes с тремя нодами.
  3. **Утилиты**
    - **kubectl** – управление кластером
    - **minikube** – локальный кластер
    - **task** – для быстрого запуска
1. **База с тестовыми данными** Firebird 5.0.3.
  2. **Мониторинг** Prometheus + Grafana + Firebird Exporter.

Вся инфраструктура разворачивается с нуля за 15-20 минут по инструкции из README.

# Создание кластера

```
@lendy minikube start --nodes=3 --cpus=3 --memory=8192
```

😊 minikube v1.36.0 on Fedora 42

✨ Using the docker driver based on existing profile

! You cannot change the number of nodes for an existing minikube cluster.

👍 Starting "minikube" primary control-plane node in "minikube" cluster

🚗 Pulling base image v0.0.47 ...

**Мануал:** <https://kubernetes.io/ru/docs/tutorials/hello-minikube/>



# Конфиги базы

Statefulset - основной конфиг базы

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: data-base
  namespace: firebird
spec:
  serviceName: "firebird-ser"
  replicas: 1
  selector:
    matchLabels:
      app: firebird-app
  template:
    metadata:
      labels:
        app: firebird-app
    spec:
      containers:
        - name: firebird-app
          image: firebirdsql/firebird:5.0.3-noble
```

PVC - хранилище данных

```
volumes:
  - name: firebird-data
    persistentVolumeClaim:
      claimName: pvc
```

Service - доступ к базе

```
apiVersion: v1
kind: Service
metadata:
  name: firebird-ser
  namespace: firebird
spec:
  type: ClusterIP
  ports:
    - port: 3050
      targetPort: 3050
  selector:
    app: firebird-app
```

# NFS сервер

Stateless - основной конфиг NFS сервера

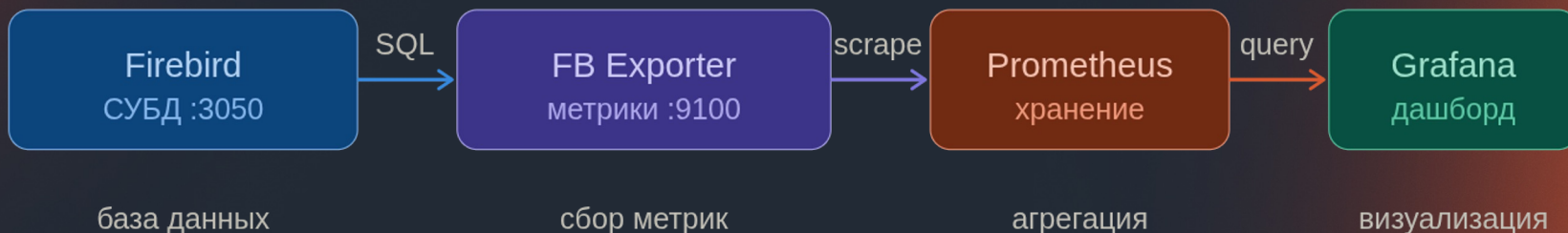
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nfs-server
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nfs-server
  template:
    metadata:
      labels:
        app: nfs-server
    spec:
      nodeSelector:
        kubernetes.io/hostname: minikube
      containers:
        - name: nfs-server
          image: itsthene트워크/nfs-server-alpine:lates
```

Service - для доступа NFS сервера

```
apiVersion: v1
kind: Service
metadata:
  name: nfs-server
  namespace: default
spec:
  clusterIP: 10.96.0.200
  selector:
    app: nfs-server
  ports:
    - port: 2049
      targetPort: 2049
```

# Мониторинг

Для мониторинга используется стандартный стек наблюдаемости – Prometheus + Grafana, который уже привычен большинству DevOps команд. К нему добавляется единственный специфичный компонент – Firebird Exporter, который умеет разговаривать с базой и переводить её внутренние метрики в формат понятный Prometheus.



# Быстрый старт

Для запуска необходимо использовать утилиту task и Taskfile

```
release:
  cmds:
    - task: build-app
    - task: build-db
    - task: apply-exporter
    - task: apply-service-monitor
    - task: start-forwarding
```

**Мануал:** <https://taskfile.dev/docs/guide>

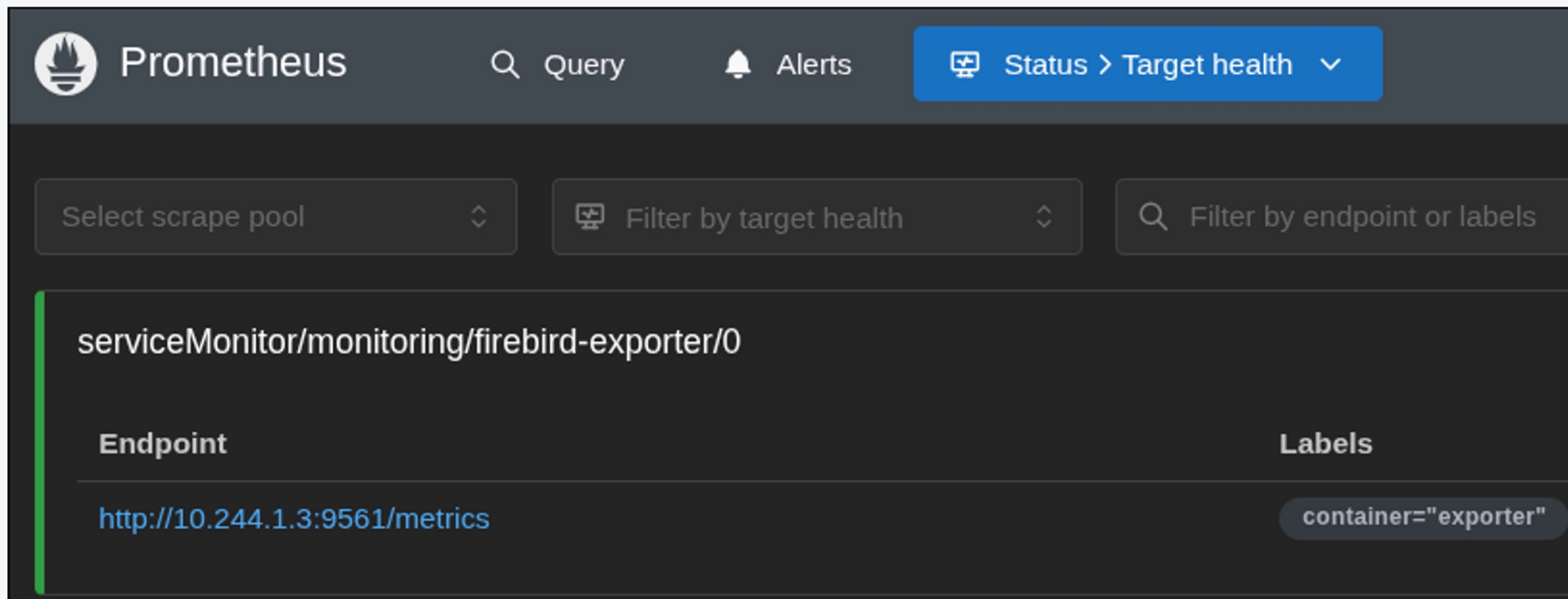


# Exporter

**Exporter** для базы данных нужен для сбора технических метрик (нагрузка, память, количество запросов) и преобразования их в формат, понятный системам мониторинга (обычно Prometheus)

```
firebird_up 1.0
# HELP firebird_connections_total Количество подключений к базе данных
# TYPE firebird_connections_total gauge
firebird_connections_total 3.0
# HELP firebird_transactions_active Количество активных транзакций
# TYPE firebird_transactions_active gauge
firebird_transactions_active 1.0
# HELP firebird_queries_active Количество активных запросов
```

# Prometheus



The screenshot displays the Prometheus web interface. At the top, there is a navigation bar with the Prometheus logo, the text "Prometheus", a search icon labeled "Query", a bell icon labeled "Alerts", and a blue button labeled "Status > Target health" with a dropdown arrow. Below the navigation bar, there are three filter boxes: "Select scrape pool" with a dropdown arrow, "Filter by target health" with a dropdown arrow, and "Filter by endpoint or labels" with a search icon. The main content area shows a path "serviceMonitor/monitoring/firebird-exporter/0". Below this, there is a table with two columns: "Endpoint" and "Labels". The "Endpoint" column contains the URL "http://10.244.1.3:9561/metrics". The "Labels" column contains a label "container=\"exporter\"".

Prometheus

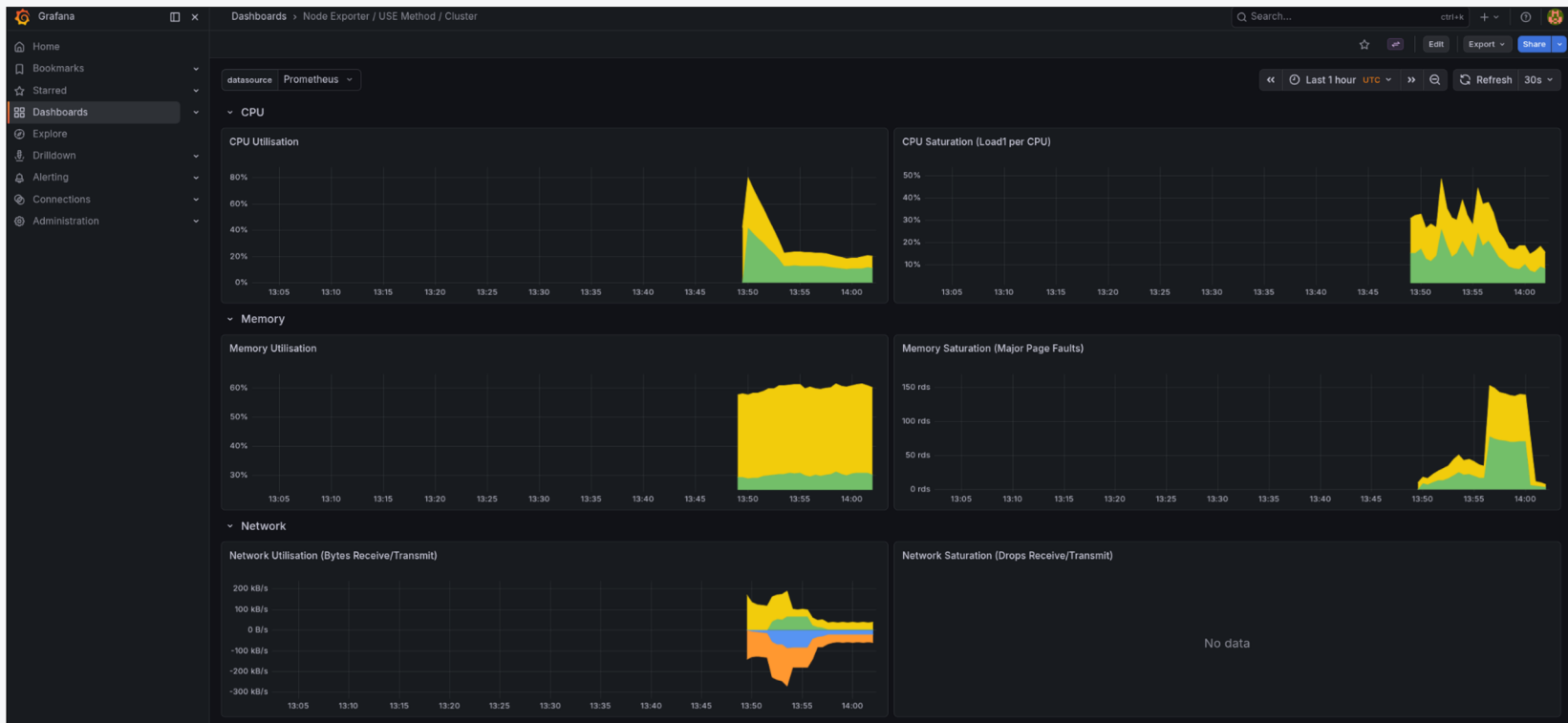
Query Alerts Status > Target health

Select scrape pool Filter by target health Filter by endpoint or labels

serviceMonitor/monitoring/firebird-exporter/0

Endpoint	Labels
<a href="http://10.244.1.3:9561/metrics">http://10.244.1.3:9561/metrics</a>	container="exporter"

# Grafana



# Процесс миграции

Мигрирование базы с одной ноды на другую в случае критической ошибки или замены оборудования

```
@lendy kubectl get pods -n firebird -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE          NOMINATED NODE   READINESS GATES
data-base-0   1/1    Running   0           4h25m 10.244.1.2   minikube-m02 <none>                 <none>
ES
```

```
Mon 13 Apr - 17:22 ~/firebird-k8s j origin main 1* 1*
@lendy kubectl get pods -n firebird -o wide -w
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE          NOMINATED NODE   READINESS GATES
data-base-0   1/1    Running   0           4h25m 10.244.1.2   minikube-m02 <none>                 <none>
data-base-0   1/1    Running   0           4h25m 10.244.1.2   minikube-m02 <none>                 <none>
data-base-0   1/1    Terminating 0       4h25m 10.244.1.2   minikube-m02 <none>                 <none>
data-base-0   1/1    Terminating 0       4h25m 10.244.1.2   minikube-m02 <none>                 <none>
data-base-0   0/1    Error       0       4h25m 10.244.1.2   minikube-m02 <none>                 <none>
data-base-0   0/1    Error       0       4h25m 10.244.1.2   minikube-m02 <none>                 <none>
data-base-0   0/1    Error       0       4h25m 10.244.1.2   minikube-m02 <none>                 <none>
data-base-0   0/1    Pending    0       0s      <none> <none>                 <none>                 <none>
data-base-0   0/1    Pending    0       0s      <none> minikube <none>                 <none>
data-base-0   0/1    ContainerCreating 0       0s      <none> minikube <none>                 <none>
data-base-0   1/1    Running   0           89s    10.244.0.10 minikube <none>                 <none>
one>
```

```
Mon 13 Apr - 17:22 ~/firebird-k8s j origin main 1* 1*
@lendy kubectl drain minikube-m02 --ignore-daemonsets --delete-emptydir-data --force
node/minikube-m02 cordoned
Warning: ignoring DaemonSet-managed Pods: kube-system/kindnet-rvngl, kube-system/kube-proxy-j2crz, monitoring/prometheus-prometheus-node-exporter-mn5j2
evicting pod monitoring/prometheus-kube-prometheus-kube-prometheus-prometheus-0
evicting pod monitoring/prometheus-kube-prometheus-operator-8678b74f54-rrjvg
evicting pod firebird/data-base-0
evicting pod monitoring/prometheus-grafana-644d588c9c-45mr5
evicting pod monitoring/prometheus-kube-state-metrics-7c568c7984-s4t2v
evicting pod monitoring/firebird-exporter-6894b64cc7-zlv57
pod/prometheus-kube-prometheus-operator-8678b74f54-rrjvg evicted
pod/prometheus-prometheus-kube-prometheus-prometheus-0 evicted
pod/prometheus-kube-state-metrics-7c568c7984-s4t2v evicted
pod/data-base-0 evicted
pod/prometheus-grafana-644d588c9c-45mr5 evicted
pod/firebird-exporter-6894b64cc7-zlv57 evicted
node/minikube-m02 drained
Mon 13 Apr - 17:23 ~/firebird-k8s j origin main 1* 1*
@lendy
```

# Видео

# Что дает бизнесу и разработчикам

## Для админов / DevOps

Меньше ручной рутины при создании/обновлении инстансов;

Единый шаблон вместо зоопарка самописных скриптов.

## Для разработчиков

Поднятие dev/stage баз;

Предсказуемое окружение для тестов и миграций.

## Для РЕД СОФТ / Firebird сообщества

Хороший «референс» по запуску Firebird в Kubernetes;

Основа для будущих Helm-чартов, операторов, мастер-классов.

# Дальнейшее развитие

## DB Security



Vault + network policies +  
TLS между сервисами

## Monitoring



Alertmanager –  
уведомления при сбоях  
и аномалиях

## Open Source



**open source**  
initiative®

Helm chart + Kubernetes  
Operator для Firebird

# Соцсети

GitHub repo



TG



Спасибо за внимание!