

# Lab 7. Мониторинг приложений

**Упражнение 1. Опционально.** На VM Linux собираем образ:

**Создайте директорию проекта и перейдите в нее**

```
mkdir work/metrics
```

```
cd work/metrics
```

**Создайте файл с кодом приложения**

```
vim server.js
```

```
const client = require('prom-client');
```

```
const express = require('express');
```

```
const app = express();
```

```
const register = new client.Registry();
```

```
client.collectDefaultMetrics({
```

```
  app: 'node-application-monitoring-app',
```

```
  prefix: 'node_',
```

```
  timeout: 10000,
```

```
  gcDurationBuckets: [0.001, 0.01, 0.1, 1, 2, 5],
```

```
  register
```

```
});
```

```
app.get('/metrics', async (req, res) => {
```

```
  res.setHeader('Content-Type', register.contentType);
```

```
  res.send(await register.metrics());
```

```
});
```

```
app.listen(8888, () => console.log('Server is running on http://localhost:8888, metrics are exposed on http://localhost:8888/metrics'));
```

**Создайте файл с декларацией зависимостей**

```
vim package.json
```

```
{
```

```
  "name": "metrics",
```

## Building Kubernetes Applications

```
"version": "1.0.0",
"description": "",
"main": "server.js",
"scripts": {
  "test": "echo \\\"Error: no test specified\\\" && exit 1"
},
"keywords": [],
"author": "Your Name",
"license": "no",
"dependencies": {
  "express": "^4.17.1",
  "prom-client": "^13.2.0"
}
}
```

### Создайте Dockerfile

#### vim Dockerfile

```
FROM node:14
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 8888
CMD [ "node", "server.js" ]
```

## Building Kubernetes Applications

`docker build -t testapp .`

`docker images`

`docker image tag testapp:lastest yorrepo/testapp:lastest`

`docker login -u yourname`

`docker push yorrepo/testapp:lastest`

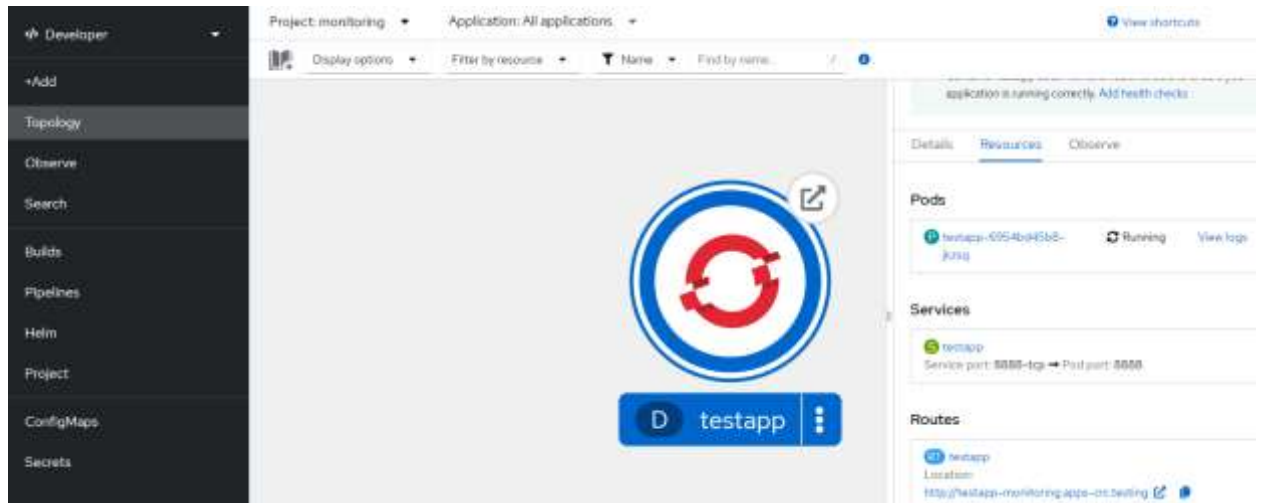
## Упражнение 2. Разверните тестовое приложение

Развертываем приложение:

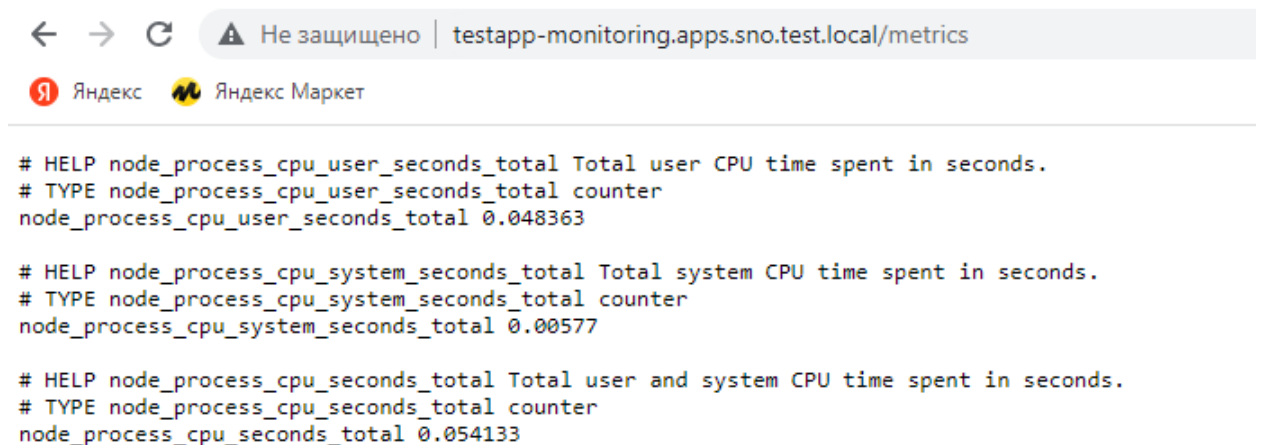
`oc new-project monitoring`

`oc new-app drrabbit/testapp:lastest`

`oc expose service/testapp`



Подключитесь из браузера хоста на <http://testapp-monitoring.apps.sno.test.local/metrics>



## Building Kubernetes Applications

### Упражнение 3. Развертываем Prometheus:

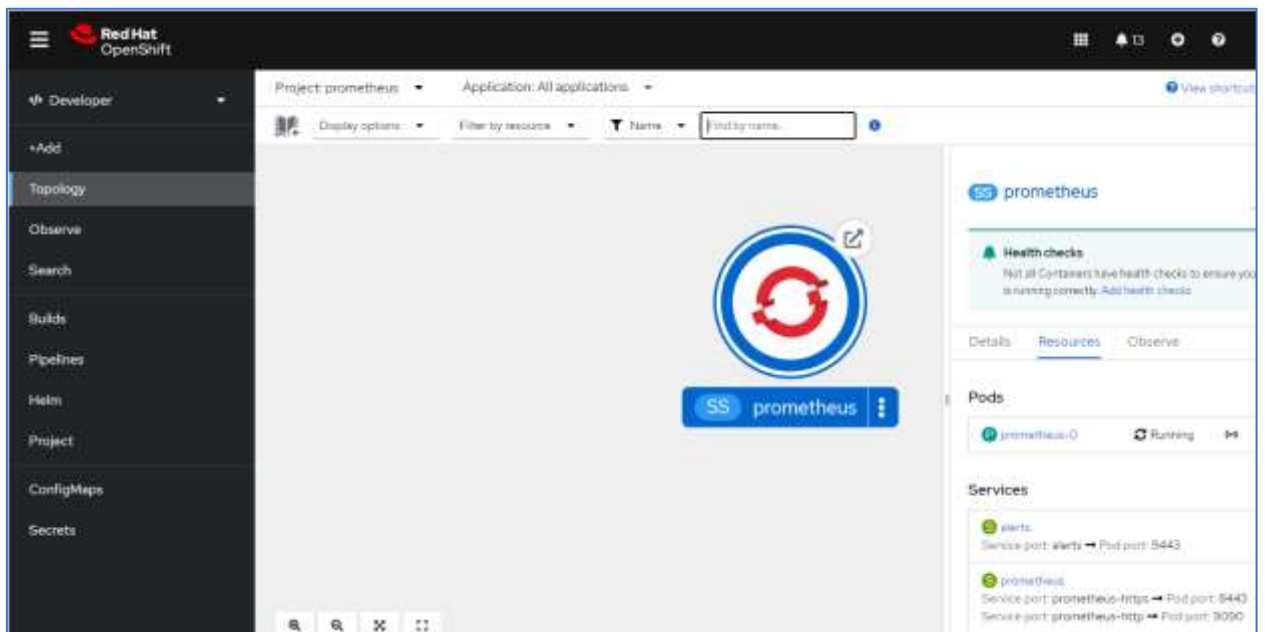
ос new-project prometheus

Используйте файл YAML prometheus.yaml отсюда:

[https://raw.githubusercontent.com/ConSol/springboot-monitoring-example/master/templates/prometheus3.7\\_without\\_clusterrole.yaml](https://raw.githubusercontent.com/ConSol/springboot-monitoring-example/master/templates/prometheus3.7_without_clusterrole.yaml)

Замените в нем API apps/v1beta1 на apps/v1

ос new-app -f prometheus.yaml -p NAMESPACE=prometheus



Подключитесь к Prometheus из браузера хоста на <http://192.168.50.31:9090>

## Building Kubernetes Applications

Создайте проект:

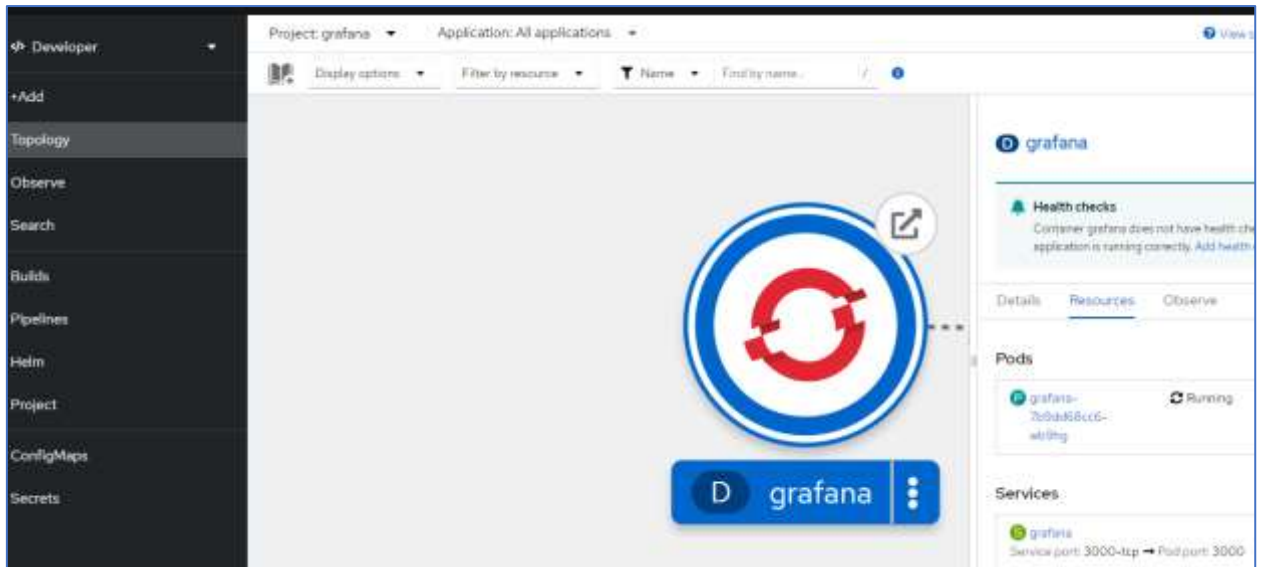
ос new-project grafana

Разверните контейнер Grafana:

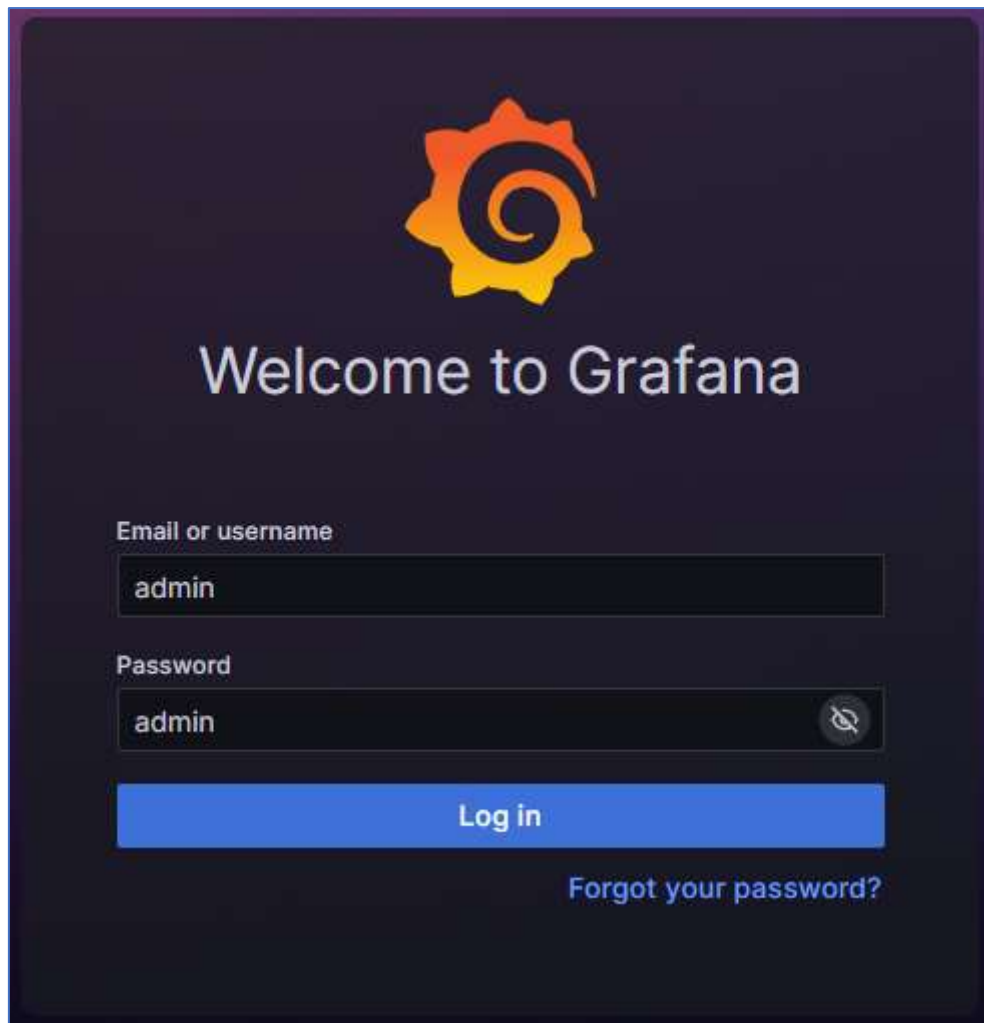
ос new-app drrabbit/grafana

Откройте маршрут:

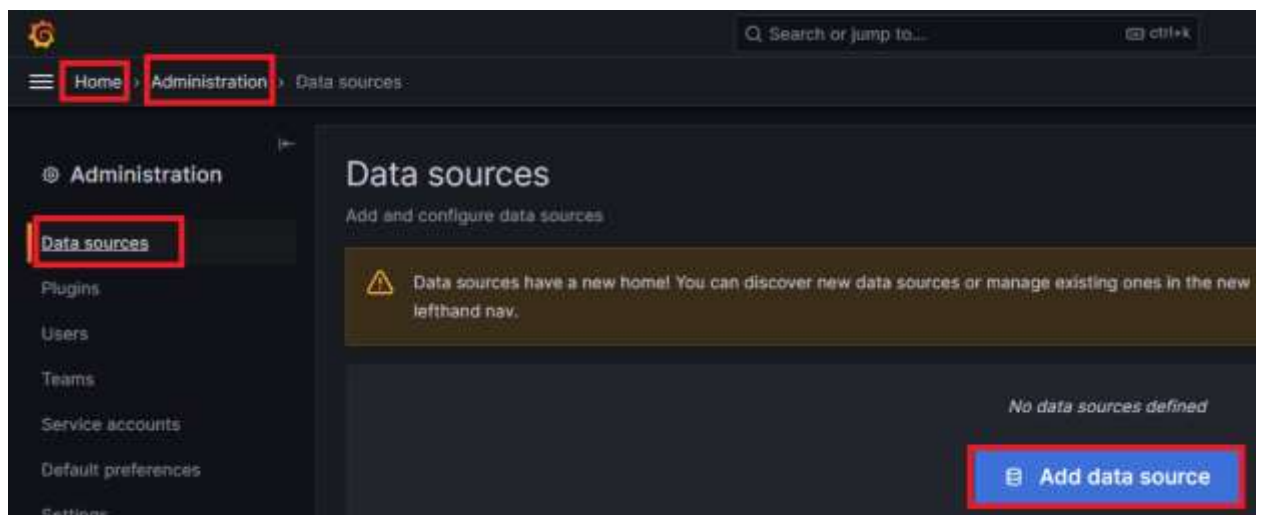
ос expose service/grafana



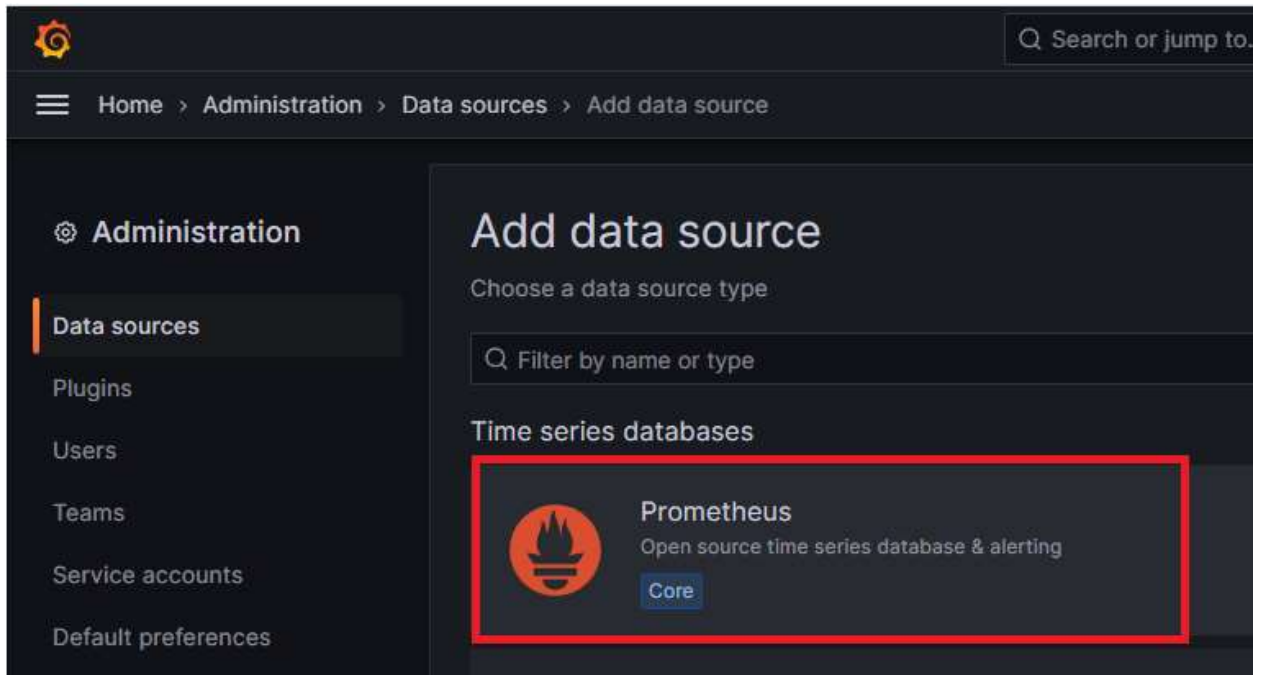
Используйте логин и пароль admin для первого входа. Затем задайте новый пароль:



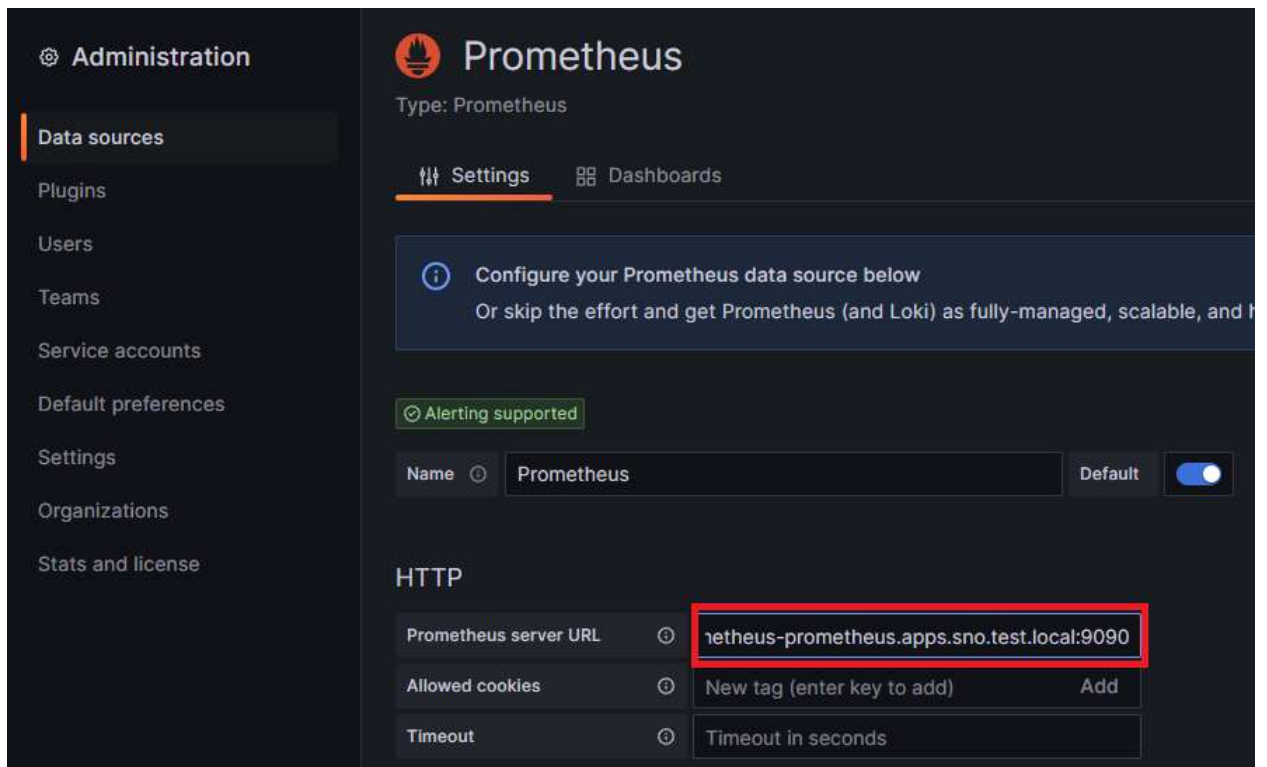
Настройте получение данных:



Выберите Prometheus в качестве источника данных:



Введите адрес и порт Prometheus-сервера

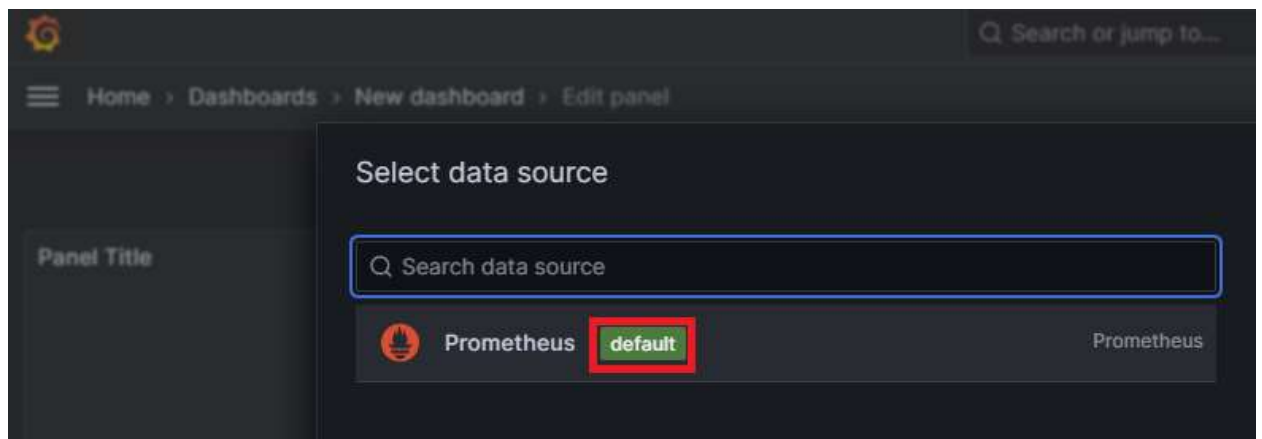
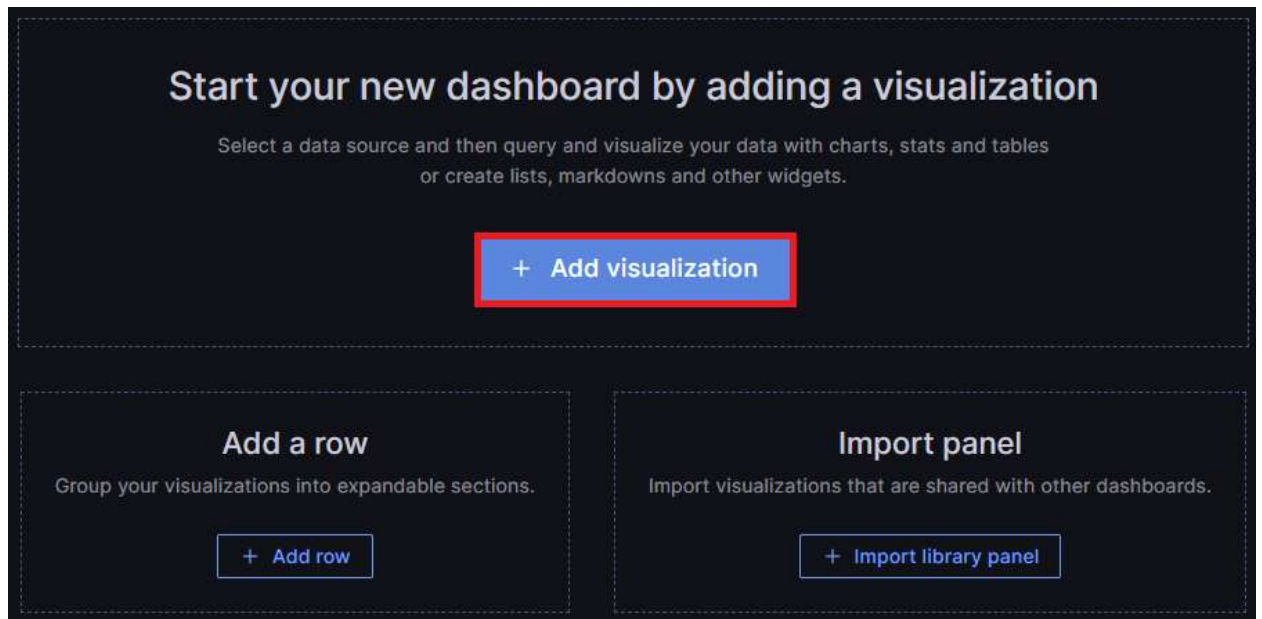


Прокрутите вниз и нажмите **Save & test**

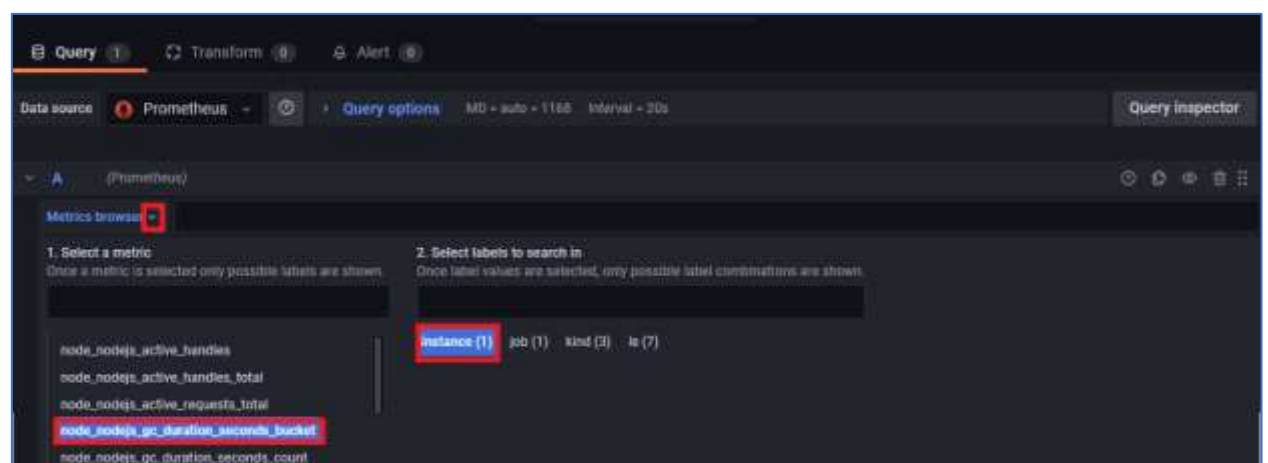


## Building Kubernetes Applications

Создайте панель:



Создайте запрос к источнику данных:



Нажмите применить (Apply) вверху справа