

Очумелые ручки. Делаем свой Helm Chart Repository из подручных средств

Романов А.С., 2003

Что мы хотим решить?



Service A

- Web
- Java
- Database



Service B

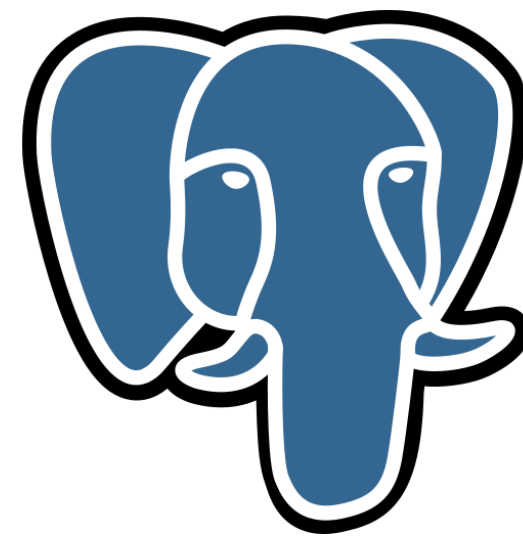
- Web
- Java
- Database
- Kafka



Service N

- Web
- Java
- No database

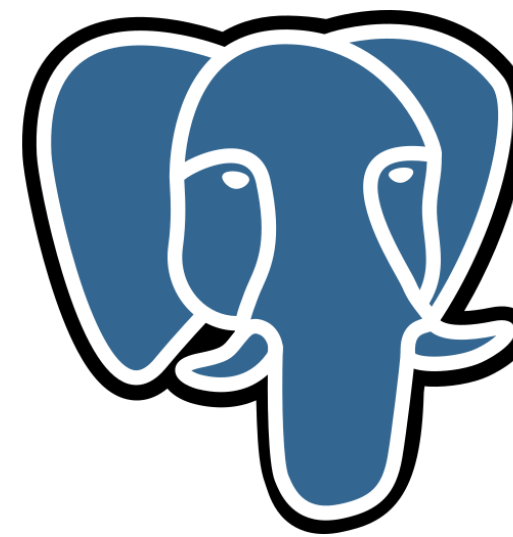
+



Database A

- Postgres
- 2Gb, 4cpu

+

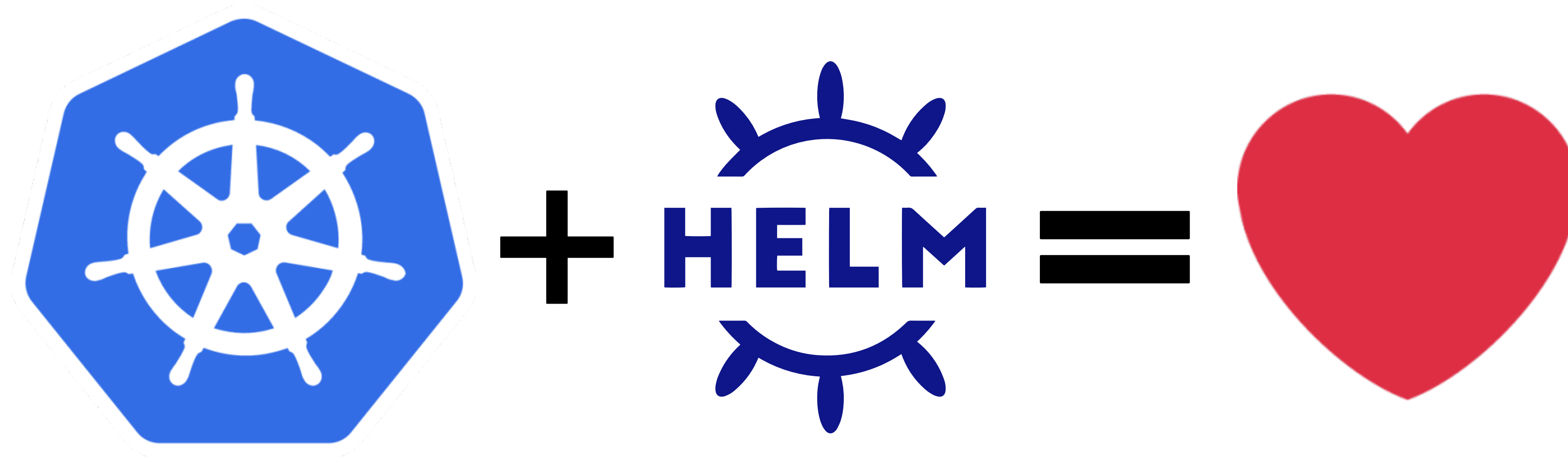


Database B

- Postgres
- 4Gb, 4cpu
- Master-Slave



Kubernetes + Helm

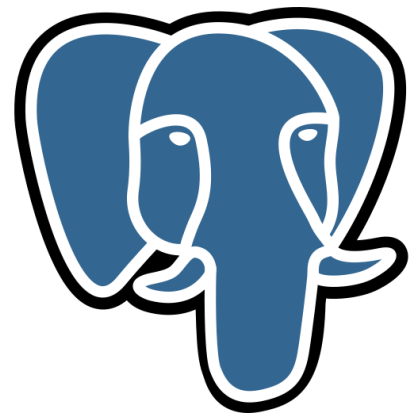


Что мы хотим решить?



- /service-chart
 - /templates
 - deployment.yaml
 - service.yaml
 - ingress.yaml
 - values.yaml
 - Chart.yaml

+

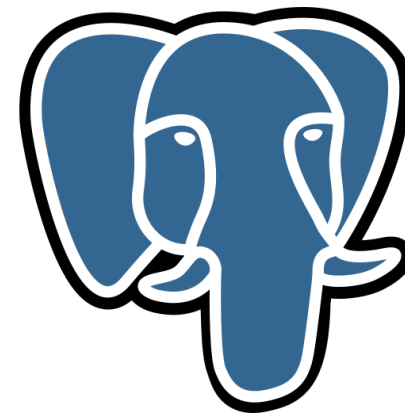


- /postgres-chart
 - /templates
 - deployment.yaml
 - config-map.yaml
 - ingress.yaml
 - values.yaml
 - Chart.yaml



- /service-chart
 - /templates
 - deployment.yaml
 - config-map.yaml
 - service.yaml
 - values.yaml
 - Chart.yaml

+



- /postgres-chart
 - /templates
 - deployment.yaml
 - config-map.yaml
 - ingress.yaml
 - values.yaml
 - Chart.yaml



- /service-chart
 - /templates
 - deployment.yaml
 - post-update-job.yaml
 - service.yaml
 - values.yaml
 - Chart.yaml



Романов Алексей

- Работаю в роли Software Архитектор.
- Co-founder в образовательном start-up **IT-Enduro**.
- 12 лет опыта в разработке Java/Kotlin.
- 7 лет в роли TeamLead/Архитектора.
- 6 лет опыт DevOps.
- 9 лет преподаю в Бауманском курс «Распределенные Системы Обработки Информации» aka Микросервисы.



План

- Best practice оформления манифестов.
- Использование subcharts для связанных сервисов.
- Создаем свой Helm Chart repo с помощью GitHub Pages.
- Создаем документацию наших чартов.
- Настраиваем правила валидации.
- Тестирование Helm Charts.



Helm

Helm – менеджер пакетов для Kubernetes. Helm развертывает пакетные приложения в Kubernetes и структурирует их в чарты (Helm Charts). Чарты содержат все предустановленные ресурсы приложения вместе со всеми версиями, которые помещены в один легко управляемый пакет.

```
helm-package/  
  |- .helmignore      # ignore  
  |- Chart.yaml       # meta информация о пакете  
  |- values.yaml      # default значения для шаблонов  
  |- charts/          # subcharts и dependencies  
  |- templates/       # шаблоны
```

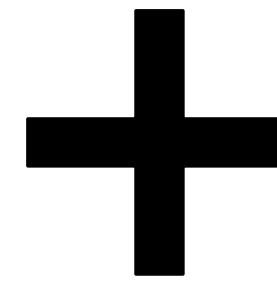
Задача Helm – предоставить инструмент для шаблонизации манифестов, чтобы их можно было переиспользовать для различных задач.



Kubernetes + Helm

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ $.Release.Name }}
spec:
  replicas: 1
  selector:
    matchLabels:
      app: {{ $.Release.Name }}
  template:
    metadata:
      labels:
        app: {{ .Release.Name }}
    spec:
      containers:
        - name: {{ .Release.Name }}
          image: "{{ .Values.repository }}:{{ .Values.tag }}"
          ports:
            - name: {{ .Chart.Name }}
              containerPort: {{ .Values.port }}
```

templates/deployment.yaml



```
# -- Image name and version
image:
  repository: romanowalex/simple-backend
  tag: v1.0

# -- Published port
port: 8080
```

templates/values.yaml

Kubernetes + Helm

```
$ helm upgrade --wait \  
  -f values.yaml \  
  --description "change resources" \  
  simple-backend \  
  k8s/java-service
```



Kubernetes + Helm

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simple-backend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: simple-backend
  template:
    metadata:
      labels:
        app: simple-backend
    spec:
      containers:
        - name: simple-backend
          image: "romanowalex/simple-backend:v1.0"
          ports:
            - name: simple-backend
              containerPort: 8080
```

manifest.yaml



Best Practice

- Переменные в camelCase (serviceName, serverPort).
- Имена chart в dash-case (java-service).
- Отступы в два пробела, tabs использовать нельзя.
- Версионирование charts через SemVer.
- Структуры предпочтительнее массивов.
- Не используем Sub Charts *.



Project ▾
Chart.yaml ×
values.yaml ×

- > .gradle
- > .idea
- > build
- > common-api
- > common-service
- > docker
- > gradle
- ▾ k8s
 - > elasticsearch-chart
 - > jaeger-chart
 - > logging-chart
 - > monitoring-chart
 - ▾ services-chart
 - ▾ charts
 - ▾ postgres-chart
 - > templates
 - Chart.yaml
 - values.yaml
 - service-chart
 - > templates
 - Chart.yaml
 - values.yaml
 - Chart.yaml
 - values.yaml
 - README.md

- > order-api
- > order-service
- > postman
- > scripts
- > store-api
- > store-service
- > warehouse-api
- > warehouse-service

```

1 apiVersion: v2 ✓
2 name: services-chart
3 type: application
4 version: 1.0.0
5 dependencies:
6   - name: postgres-chart
7     version: 1.0.0
8     alias: postgres
9   - name: service-chart
10    version: 1.0.0
11    alias: order
12   - name: service-chart
13     version: 1.0.0
14     alias: store
15   - name: service-chart
16     version: 1.0.0
17     alias: warranty
18   - name: service-chart
19     version: 1.0.0
20     alias: warehouse

```

```

1 order:
2   service:
3     port: 8380
4     image:
5       name: romanowalex/order-service
6       version: v1.0-lecture
7
8   store:
9     service:
10    port: 8480
11    image:
12      name: romanowalex/store-service
13      version: v1.0-lecture
14
15   warranty:
16     service:
17      port: 8180
18     image:
19      name: romanowalex/warranty-service
20      version: v1.0-lecture
21
22   warehouse:
23     service:
24      port: 8280
25     image:
26      name: romanowalex/warehouse-service
27      version: v1.0-lecture

```

Document 1/1 > apiVersion:
Document 1/1 > order:

Использование SubCharts

ПЛЮСЫ:

- Можно задеплоить все одной командой (Umbrella Chart).
- Бывают сервисы, которые не могут жить отдельно от своих зависимостей. (Например Kafka + Zookeeper, Patroni + Etcd + Nginx).

МИНУСЫ:

- Усложняется настройка, т.к. часть конфигурации становится внутренней.
- Усложняются обновления: для обновления версии Postgres нам надо передеплоить верхнеуровневый chart.
- А что делать, если одному сервису нужен Postgres, другому MongoDB, а еще нужен отдельный Postgres для аналитики.

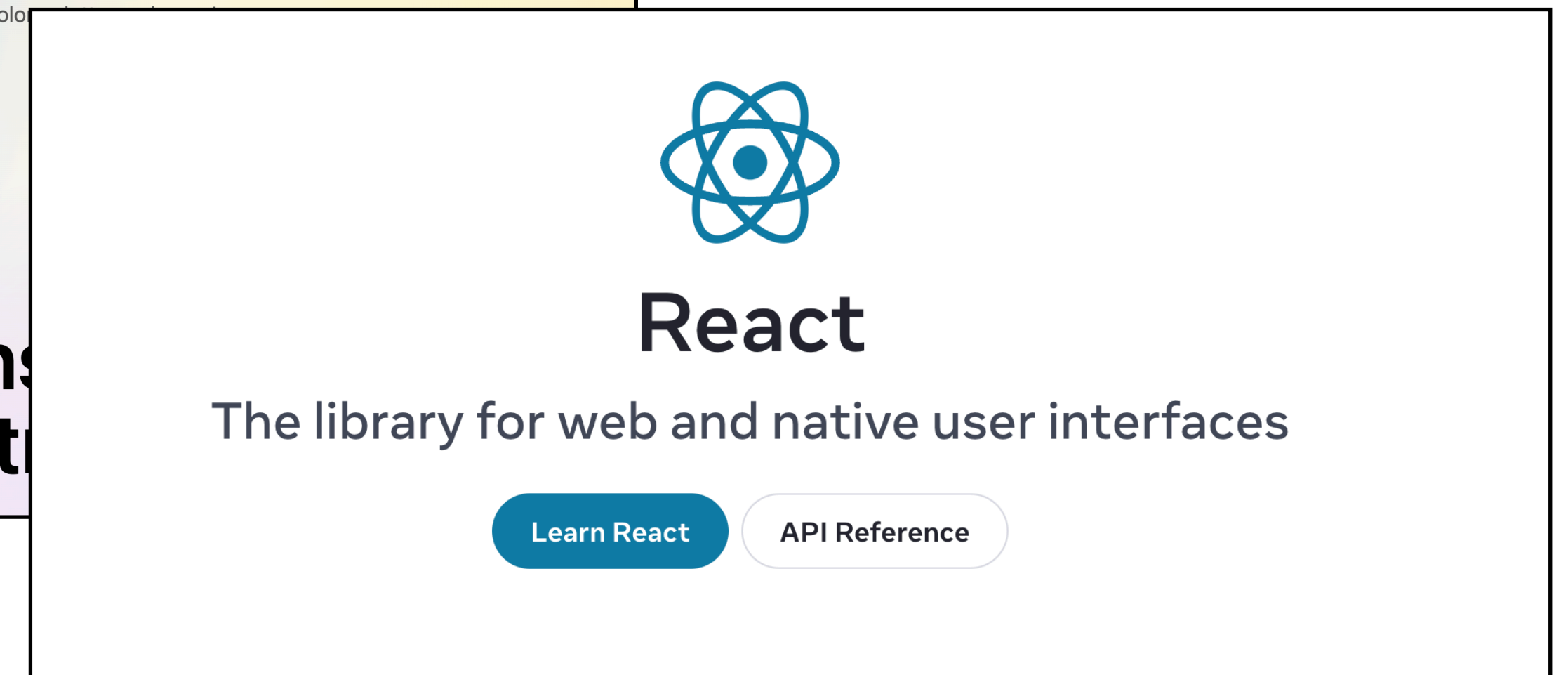
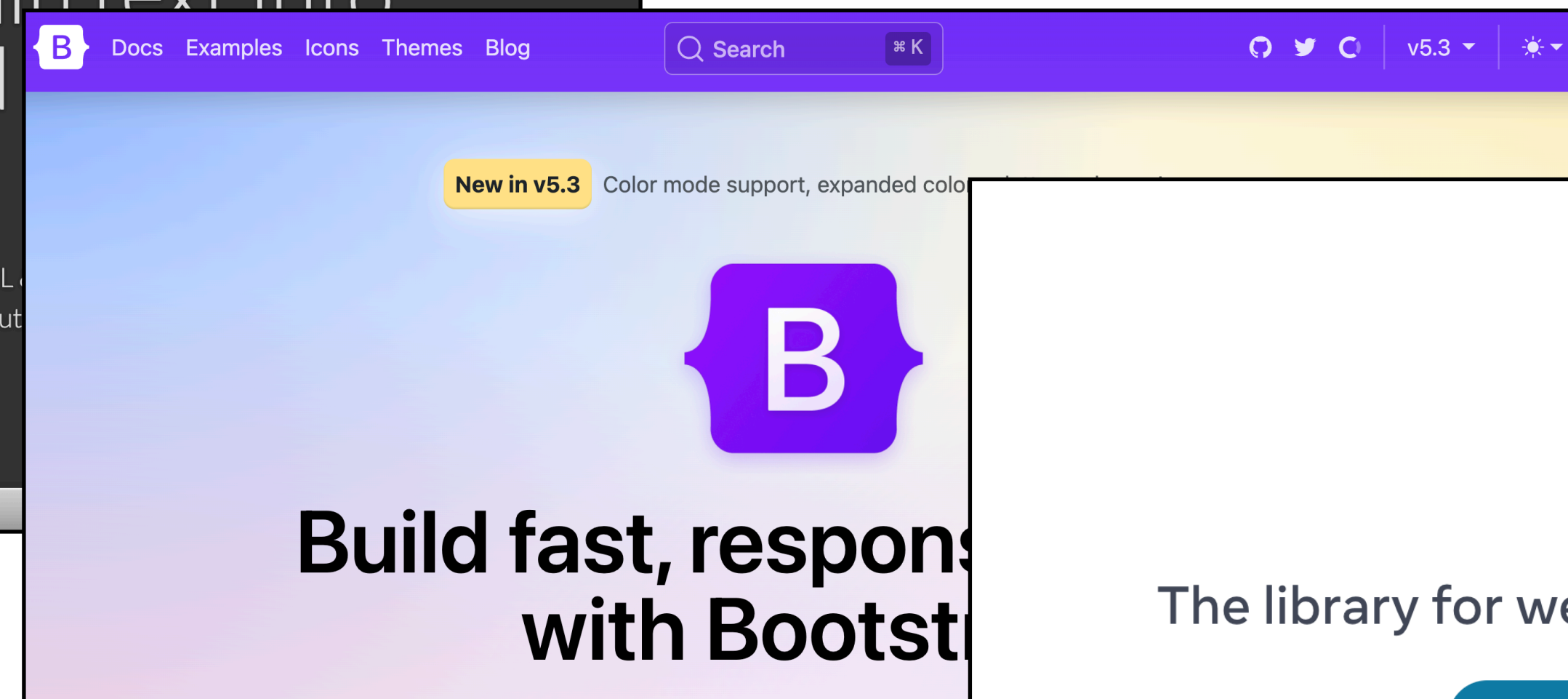
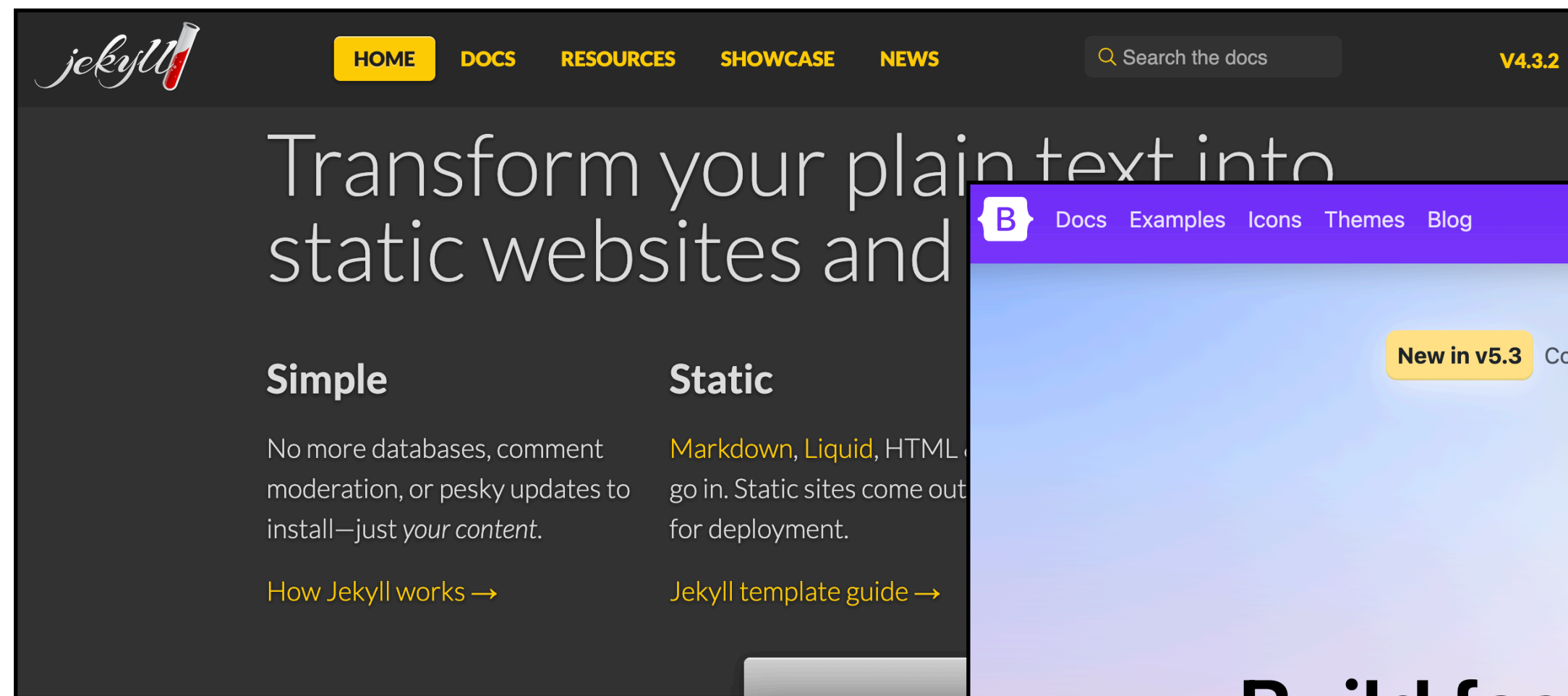
Оформление Chart.yaml

```
apiVersion: v2
name: java-service
type: application
description: Helm chart for Spring Boot application with logging and tracing
version: 1.3.0
appVersion: "v1.0"
home: https://.../Romanow/helm-charts/tree/master/charts/java-service
icon: https://.../Romanow/helm-charts/master/icons/spring-boot.png
sources:
  - https://spring.io/projects/spring-boot
  - https://hub.docker.com/_/amazoncorretto
keywords:
  - spring boot
  - java
  - kotlin
maintainers:
  - name: Romanov Alexey
    email: romanowalex@mail.ru
    url: https://romanow.github.io
```



GitHub Pages

GitHub Pages — это общедоступные статические веб-страницы, размещенные и опубликованные через GitHub.



GitHub Pages

Для создания сайта нужно создать ветку **gh-pages** и добавить в нее файлы:

- index.md – содержимое страницы;
- _config.yml – тема и прочие настройки.

```
index.md x
1  [!pre-commit](https://img.shields.io/badge/pre--commit-enabled-brightgreen?l
2  [!Artifact Hub](https://img.shields.io/endpoint?url=https://artifacthub.io/b
3  [!Release Charts](https://github.com/Romanow/helm-charts/actions/workflows/cf
4
5  ## Installation
6
7  `shell
8  $ helm repo add romanow https://romanow.github.io/helm-charts/
9  $ helm search repo romanow
10 `
11
_config.yml x
1  title: Helm Charts repository
2  description: Helm Charts repository for education purpose
3
4  remote_theme: pages-themes/cayman@v0.2.0
5  plugins:
6  - jekyll-remote-theme
```


romanow.github.io/helm-charts-repo-lecture/



Helm Charts repository

Sample Helm Charts repository for public lecture

[View on GitHub](#)

pre-commit enabled | Artifact Hub romanow-helm-charts-repo | Release Charts failing

Installation

```
$ helm repo add romanow https://romanow.github.io/helm-charts-repo-lecture/  
$ helm search romanow
```

helm-charts-repo-lecture is maintained by Romanow.

This page was generated by [GitHub Pages](#).

Создаем свой собственный Helm Chart Repository

Для публикации используем `helm/chart-releaser-action`.

Он собирает charts в архив¹ и публикует их как Release² в GitHub.

После этого собираем файл `index.yaml`³ с информацией о charts и делаем коммит в ветку `gh-pages` (GitHub Pages).

```
$ cr package charts/* --package-path release 1  
$ owner=$(cut -d '/' -f 1 <<< "$GITHUB_REPOSITORY")  
$ repo=$(cut -d '/' -f 2 <<< "$GITHUB_REPOSITORY")
```

```
$ cr upload \  
  --package-branch gh-pages \  
  --package-with-index \  
  --package-path release \  
  --git-repo "$repo" \  
  --owner "$owner" 2
```

```
$ cr index \  
  --package-branch gh-pages \  
  --package-with-index \  
  --package-path release \  
  --git-repo "$repo" \  
  --owner "$owner" \  
  --push 3
```

```
apiVersion: v1
entries:
  elasticsearch:
    - apiVersion: v2
      appVersion: 7.17.8
      created: "2023-01-24T14:09:05.969526161Z"
      description: Elasticsearch is a distributed, RESTful search and analytics engine
        capable of addressing a growing number of use cases.
      digest: e2f050c7aeebb844e124b626aa82371e752b271640d02b96c11564b11508a4b4
      home: https://github.com/Romanow/helm-charts/tree/master/charts/elasticsearch
      icon: https://raw.githubusercontent.com/Romanow/helm-charts/master/images/elastic.png
      keywords:
        - elasticsearch
        - database
      maintainers:
        - email: romanowalex@mail.ru
          name: Romanov Alexey
          url: https://romanow.github.io
      name: elasticsearch
      sources:
        - https://www.elastic.co/elasticsearch/
        - https://hub.docker.com/_/elasticsearch
      type: application
      urls:
        - https://github.com/Romanow/helm-charts/releases/download/elasticsearch-1.3.0/elasticsearch-1.3.0.tgz
      version: 1.3.0
    - apiVersion: v2
      appVersion: 7.17.8
      created: "2023-01-24T13:39:31.461828394Z"
      description: Elasticsearch is a distributed, RESTful search and analytics engine
        capable of addressing a growing number of use cases.
      digest: 1d4932481108b180c65f8dc8501768300774502d354aa69f9df500c45ab6b99b
      home: https://github.com/Romanow/helm-charts/tree/master/charts/elasticsearch
      icon: https://github.com/Romanow/helm-charts/tree/master/images/elastic.jpeg
      keywords:
        - elasticsearch
        - database
```

```
aromanov ~ ➤ helm repo add romanow https://romanow.github.io/helm-charts/
```

```
"romanow" has been added to your repositories
```

```
aromanov ~ ➤ helm search repo romanow
```

NAME	CHART VERSION	APP VERSION	DESCRIPTION
romanow/elasticsearch	1.3.0	7.17.8	Elasticsearch is a distributed, RESTful search ...
romanow/frontend	1.3.0	v1.0	Frontend application running in nginx, based on...
romanow/grafana	1.3.0	8.3.4	Grafana allows you to query, visualize, alert o...
romanow/influxdb	1.3.0	1.8.4	The Time Series Data Platform where developers ...
romanow/jaeger-collector	1.3.0	1.41.0	Jaeger is a distributed tracing system released...
romanow/jaeger-query	1.3.0	1.41.0	Jaeger is a distributed tracing system released...
romanow/java-service	1.3.0	v1.0	Helm chart for Spring Boot application with log...
romanow/kibana	1.3.0	7.17.8	Kibana is a free and open user interface that l...
romanow/logstash	1.3.0	7.17.8	Logstash is a free and open server-side data pr...
romanow/node-exporter	1.3.0	1.5.0	Prometheus exporter for hardware and OS metrics...
romanow/postgres	1.3.2	15	PostgreSQL is a powerful, open source object-re...
romanow/prometheus	1.3.0	2.40.0	Prometheus collects and stores its metrics as t...

```
aromanov ~ ➤
```

Публикуем репозиторий на ArtifactHub

ArtifactHub – это веб-приложение, которое позволяет находить, устанавливать и публиковать Helm charts.

ArtifactHub не хранит пакеты, а только связывает их с репозиторием.





Repositories

If you want your repositories to be labeled as **Verified** that you own or have control over the repository.

Helm Charts

🔒 Verified Publisher

ID: c8de8ab5-cfae-46e4-8a8b-bb3372a919db

URL: <https://romanow.github.io/helm-charts/>

LAST PROCESSED: 17 days ago ✓ (next check in ~ 5)

LAST SECURITY SCAN: a day ago ✓ (next scan in ~)

Update repository



Kind

Helm charts

Name (Required)

romanow-helm-charts

This name will appear in your packages' urls and **cannot be updated** once is saved.

Display name

Helm Charts

Url (Required)

<https://romanow.github.io/helm-charts/>

For more information about the url format and the repository structure, please see the [Helm charts repositories](#) section in the [repositories guide](#).

Disabled

Use this switch to disable the repository temporarily or permanently.

Security scanner disabled

Use this switch to disable the security scanning of the packages in this repository.

✎ UPDATE

🔄 REFRESH

🔗 CLAIM OWNERSHIP

➕ ADD

ID provided below. This label will let users know

PROJECT

Documentation

Blog

Statistics

COMMUNITY

🌐 GitHub

🗨 Slack

🐦 Twitter



Copyright © The Artifact Hub Authors

ЖДЕМ ПРИМЕРНО 30 МИНУТ

1 - 12 of 12 results

Show: 20

Filters: REPOSITORY: romanow-helm-charts

FILTERS

⊗ Reset

- Official
- Verified publishers

KIND

- Helm charts (10497)
- OLM operators (305)
- Tekton tasks (297)
- Krew kubectl plugins (216)
- Containers images (76)
- Keptn integrations (43)
- Gatekeeper policies (41)
- Helm plugins (36)
- Kubewarden policies (28)
- Falco rules (23)
- Tinkerbelle actions (11)
- OPA policies (7)
- KEDA scalers (4)
- Tekton pipelines (3)

CATEGORY



elasticsearch

Romanow Helm Charts

★ 1 Helm chart

Updated 17 days ago
Version 1.3.0

Elasticsearch is a distributed, RESTful search and analytics engine capable of addressing a growing number of use cases.

Verified Publisher Images Security Rating



frontend

Romanow Helm Charts

★ 1 Helm chart

Updated 17 days ago
Version 1.3.0

Frontend application running in nginx, based on create-react-app.

Verified Publisher



grafana

Romanow Helm Charts

★ 1 Helm chart

Updated 17 days ago
Version 1.3.0

Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored.

Verified Publisher Images Security Rating

ArtifactHub



[https://artifacthub.io/packages/
search?repo=romanow-helm-charts](https://artifacthub.io/packages/search?repo=romanow-helm-charts)

Генерация документации

Helm Docs – инструмент для автоматической генерации документации на базе информации из `Charts.yaml` и его параметрах (`values.yaml`).

Markdown генерируется с помощью `go templates` на базе шаблона, описанного в файле `README.md.gotmpl`.



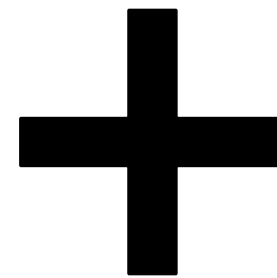
Генерация документации

```
# -- Image name and version
image:
  pullPolicy: IfNotPresent
  repository:
  tag:

ingress:
  # -- Enable ingress
  enabled: false
  # -- Ingress name
  # @default -- .Release.Name
  name:
  # -- Domain
  domain: romanow-alex.ru

# -- Count of replicas
replicas: 1
ports:
  # -- Application port
  internal: 8080
  # -- If define, create
  # -- NodePort for external usage
  external:
```

charts/java-service/values.yaml



```
{{- define "repository.url" -}}
https://github.com/romanow/helm-charts
{{- end -}}
```

```
{{- define "helm.url" -}}
https://romanow.github.io/helm-charts/
{{- end -}}
```

```
{{- define "install" -}}
### Installing the Chart
```

To install the chart with the release name
`{{ template "chart.name" . }}`:

```
```shell
$ helm repo add {{ template "repository.organization" . }} \
 {{ template "helm.url" . }}
$ helm repo update
$ helm install \
 {{ template "chart.name" . }} \
 {{ template "helm.path" . }}
```
{{- end -}}
```

charts/java-service/README.md.gotmpl

Генерация документации

```
$ helm-docs -chart-search-root charts/
```

Helm Chart for java-service

Version 1.4.2 Type application AppVersion v1.0

Helm chart for Spring Boot application with logging and tracing

Installing the Chart

To install the chart with the release name `java-service` :

```
➤ $ helm repo add romanow https://romanow.github.io/helm-charts/  
$ helm repo update  
$ helm install java-service romanow/java-service
```

Configuration

| Key | Type | Default | Description |
|-------------------|--------|------------|---|
| database.host | string | "postgres" | Database host (set `DATABASE_HOST` env) |
| database.name | string | null | Database name (if defined, set `DATABASE_NAME` env) |
| database.password | string | "test" | Database password (set `DATABASE_PASSWORD` env) |


charts/java-service/README.md

HELM DOCS



<https://github.com/norwoodj/helm-docs>

Валидация Helm Charts

 Pre-commit Framework – это менеджер скриптов для pre commit hooks. Вы задаете список проверок и действий, которые нужно выполнить, а pre-commit скачивает нужные ресурсы и выполняет эти скрипты. Тем самым он дает возможность легкого переноса pre-commit скриптов между проектами.



Валидация Helm Charts

repos:

- repo: `https://github.com/pre-commit/pre-commit-hooks`
hooks:
 - id: `trailing-whitespace` # *обрезаем конечные пробелы*
 - id: `end-of-file-fixer` # *добавляет пустую строку в конце файла*
 - id: `check-case-conflict` # *проверка имен файлов, которые могут быть некорректны в Windows или MacOS*
- repo: `https://github.com/Lucas-C/pre-commit-hooks`
hooks:
 - id: `remove-crlf` # *замена \r\n на \n*
 - id: `remove-tabs` # *замена tabs на whitespaces*

- repo: `https://github.com/norwoodj/helm-docs`
hooks:
 - id: `helm-docs` # *генерация README.md*

- repo: `https://github.com/gruntwork-io/pre-commit`
hooks:
 - id: `helmlint` # *валидация корректности helm chart*

`.pre-commit-config.yaml`

Валидация Helm Charts

```
$ pre-commit run --all \  
-config=.pre-commit-config.yaml
```

.pre-commit-config.yaml

PRE COMMIT



<https://pre-commit.com/>

Тестирование Helm Charts

- Запустить локальный кластер k8s.
- Установить chart:
`helm install my-release charts/my-chart`
- Проверить что все ресурсы запущены:
`kubectl wait --for=condition=ready pod --timeout=120s`
- Очистить ресурсы:
`helm uninstall my-release`



Тестирование Helm Charts

- Локальный кластер k8s – kind.
- Установка с помощью Chart-Test (ct) от Helm.



Тестирование Helm Charts

```
$ ct lint-and-install \  
  --validate-maintainers=false \  
  -charts=charts/java-service
```



Тестирование Helm Charts

```
name: "Release Charts"
on: [ push ]
jobs:
  ....

test:
  runs-on: ubuntu-latest
  needs: validate
  strategy:
    matrix: ${{ fromJson(needs.validate.outputs.matrix) }}
  steps:
    ....

    - name: Create kind cluster
      uses: helm/kind-action@v1.4.0

    - name: Run ct install
      run: |-
        ct lint-and-install \
          -config=.ct.yaml \
          -charts ${{ matrix.chart }}
```



← Release Charts

add whole test #109

Re-run all jobs

Summary

Jobs

- ✓ Validate charts
- ✓ Test charts (charts/elasticsearch)
- ✓ Test charts (charts/frontend)
- ✓ Test charts (charts/grafana)
- ✓ Test charts (charts/influxdb)
- ✓ Test charts (charts/jaeger-colle...)
- ✓ Test charts (charts/jaeger-query)
- ✓ Test charts (charts/java-service)
- ✓ Test charts (charts/kibana)
- ✓ Test charts (charts/logstash)
- ✓ Test charts (charts/node-export...)
- ✓ Test charts (charts/postgres)
- ✓ Test charts (charts/prometheus)
- ✓ Publish charts

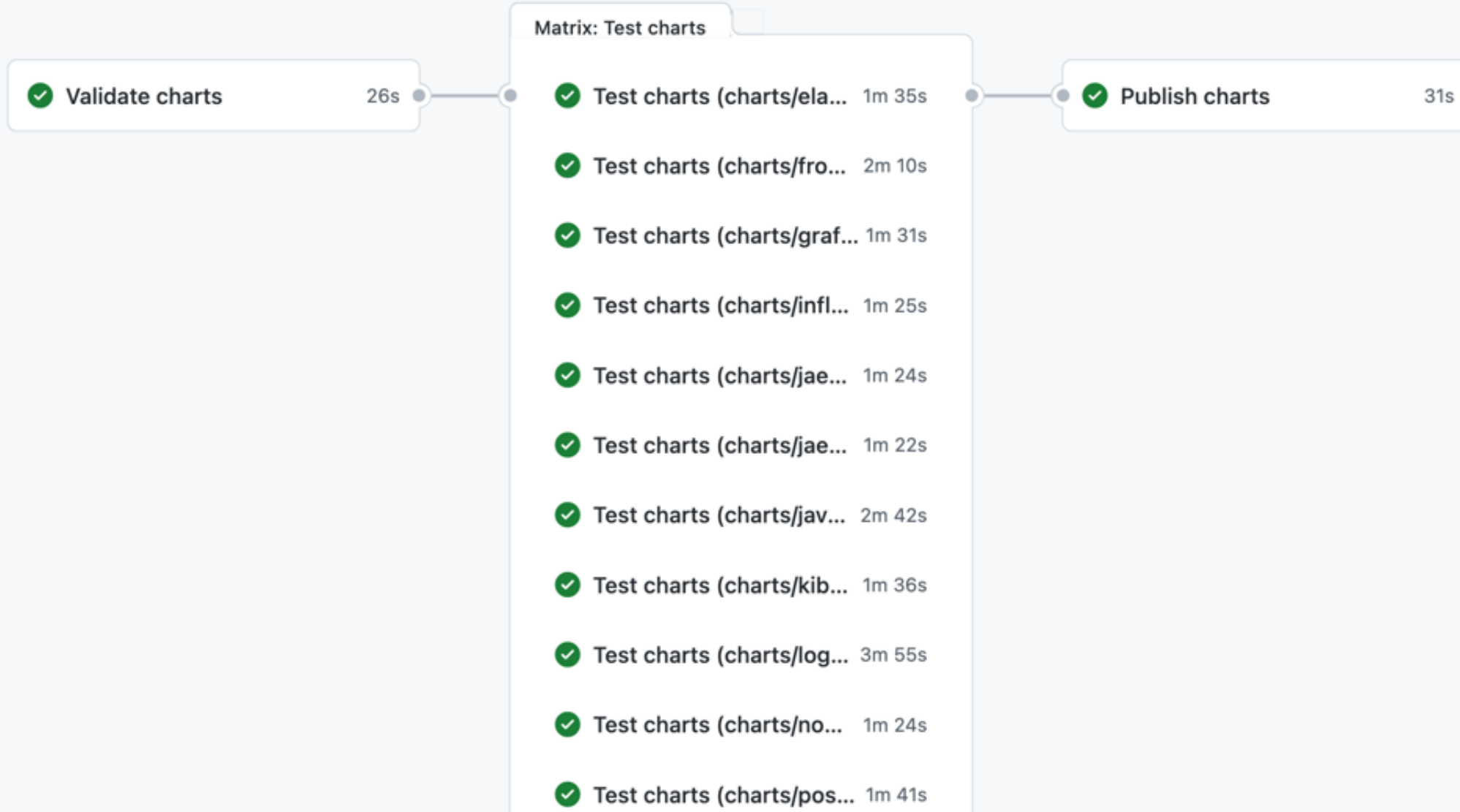
Run details

Triggered via push 11 hours ago Status Total duration Artifacts

Romanow pushed 75fa68a master Success 5m 18s -

charts-release.yaml

on: push



Тестирование Helm Charts

```
name: "Release Charts"
on: [ push ]
jobs:
  validate:
    runs-on: ubuntu-latest
    outputs:
      matrix: |
        {
          "chart": ${{ steps.set-matrix.outputs.charts }}
        }
    steps:
      .....
```

```
- name: Get changed files
  id: detect-changes
  uses: tj-actions/changed-files@v35
  with:
    json: true
    dir_names: true
    dir_names_max_depth: 2
    files: |
      charts/**
```



Тестирование Helm Charts

Если дефолтные значения в `values.yaml` не полные или не подходят для корректного запуска, то Chart Testing позволяет задавать дополнительные параметры для запуска теста в `charts/*/ci/*-values.yaml`.

```
Project
├── .idea
├── charts
│   ├── elasticsearch
│   ├── frontend
│   ├── grafana
│   ├── influxdb
│   ├── jaeger-collector
│   ├── jaeger-query
│   └── java-service
│       └── ci
│           └── test-values.yaml
│               ├── templates
│               │   ├── Chart.yaml
│               │   ├── README.md
│               │   ├── README.md.gotmpl
│               │   └── values.yaml
│               ├── kibana
│               ├── logstash
│               ├── node-exporter
│               ├── postgres
│               └── prometheus
└── test-values.yaml

values.yaml
1 # -- Image name and version
2 image:
3   pullPolicy: IfNotPresent
4   repository:
5   tag:
6
7 # Memory allocation inside container
8 memoryUsage: 75.0
9
10 # Additional Java arguments
11 javaOptions:
12
13 ingress:
14   # -- Enable ingress
15   enabled: false
16   # -- Ingress name
17   # @default -- .Release.Name
18   name:
19   # -- Domain
20   domain: romanow-alex.ru

test-values.yaml
1 image:
2   pullPolicy: IfNotPresent
3   repository: romanowalex/simple-backend
4   tag: v1.0
5
6 metrics:
7   enabled: false
8
9 memoryUsage: 25.0
10
```

CHART TESTING



<https://github.com/helm/chart-testing>

Выводы

- Если вы работаете с k8s, то helm – ваш лучший друг.
- Чтобы переиспользовать helm charts в других проектах, можно легко и просто сделать свой собственный репозиторий.
- Linters и прочие валидаторы помогают поддерживать код в чистоте и порядке.
- Helm docs позволяет поддерживать актуальную документацию без лишних усилий.
- Тестирование helm – one love ❤️, без него вы узнаете что что-то не работает только в момент деплоя.

Ссылки



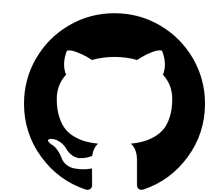
[https://github.com/
Romanow/helm-charts](https://github.com/Romanow/helm-charts)



[https://artifacthub.io/packages/
search?repo=romanow-helm-charts](https://artifacthub.io/packages/search?repo=romanow-helm-charts)

Ссылки

Романов Алексей



[@romanow](#)



[romanowalex](#)

IT Enduro



[@it_enduro](#)



[it-enduro.ro](#)