



Helmwave

Yet another deployment tool



Дмитрий Жиляев
5 лет в DevOps'e
Maintainer helmwave

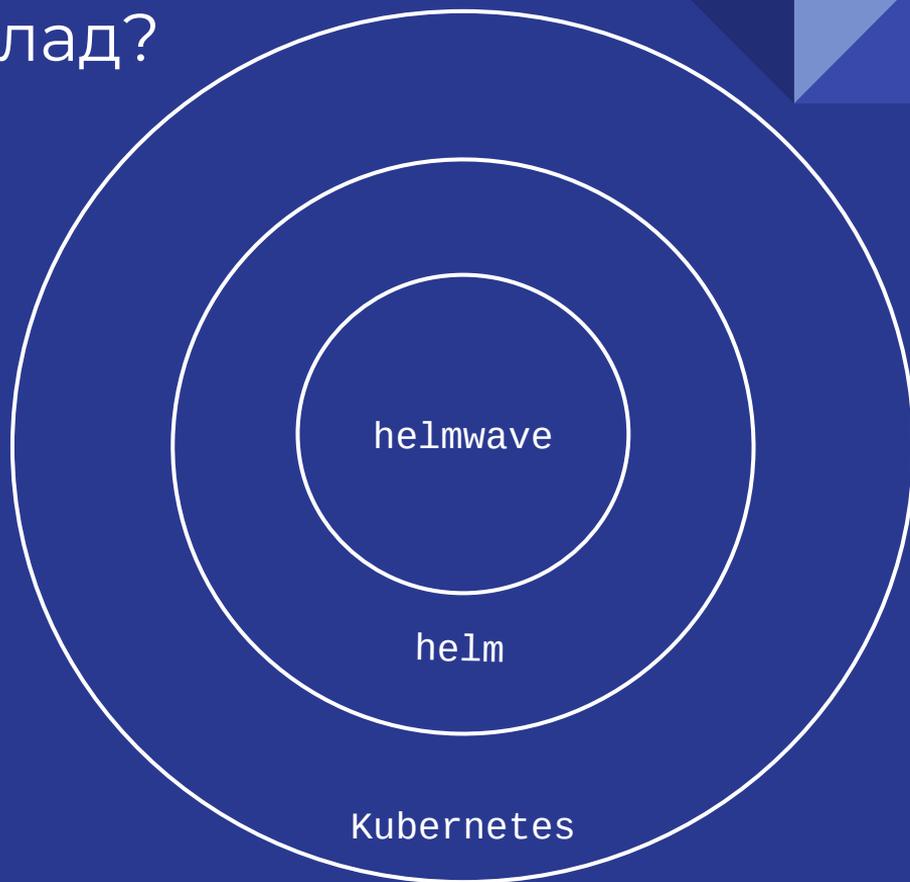


Роман Миронов
DevOps Engineer в Picnic

О чем доклад?

- А точно ли нужно делать еще один инструмент?
- Боли
- Практики, которые приносит helmwave
- Мини воркшоп
- Будущее инструментов развертывания (CD)

Для кого этот доклад?



Зачем создали helm?

Больно

6 проектов

*

10-20 сервисов

*

2-10 окружений

Сервисы и микросервисы

{...} configmap.yaml

{...} deployment-consumers.yaml

{...} deployment-import-feeds.yaml

{...} deployment.yaml

Год спустя

{...} cronjob-check-deleted-media.yaml

{...} deployment-admin-queue-moderation-a...

{...} deployment-cache-invalidation.yaml

{...} deployment-chat-orders.yaml

{...} deployment-import-deleted-media.yaml

{...} deployment-import-feeds.yaml

{...} deployment-import-sc-propose.yaml

{...} deployment-worker-admin-queue.yaml

{...} deployment-worker-sbr-from-file.yaml

{...} deployment-worker-sbr-get-rating.yaml

{...} deployment.yaml

Окружения / Контуры

Статические

Prod

Stage

Именованные

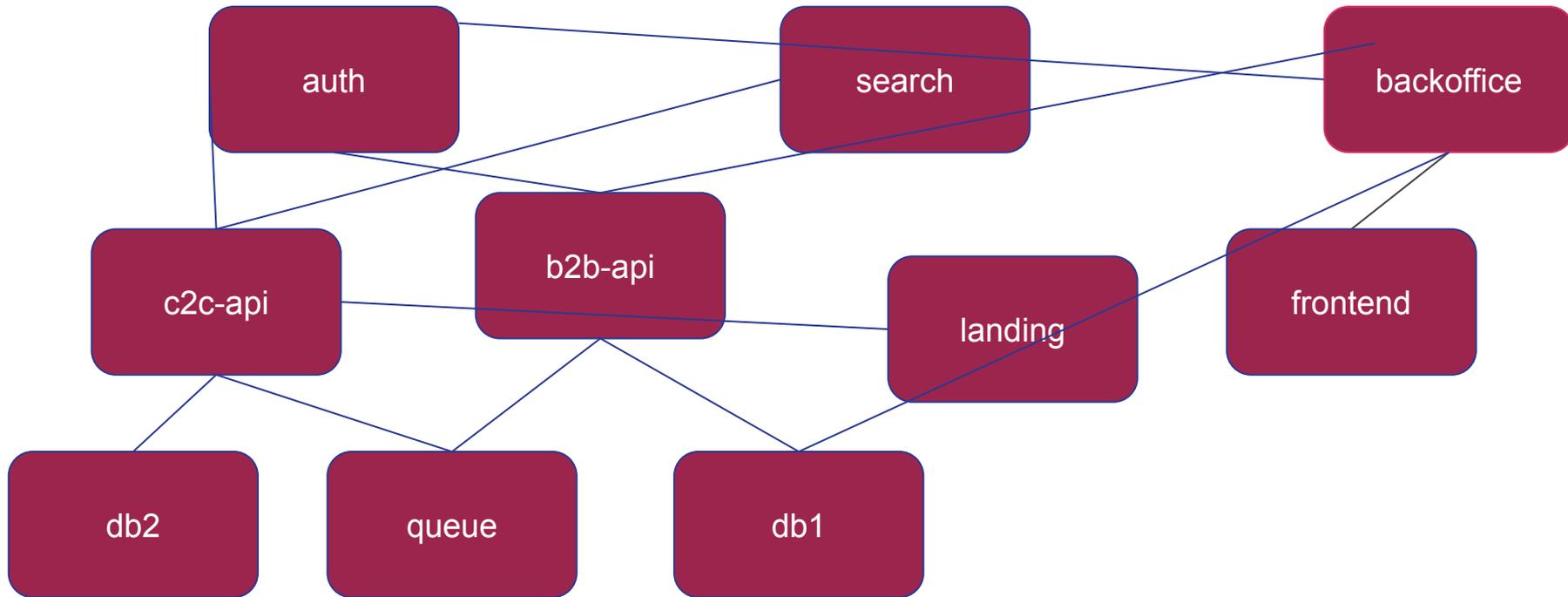
Динамические

review/feature-1

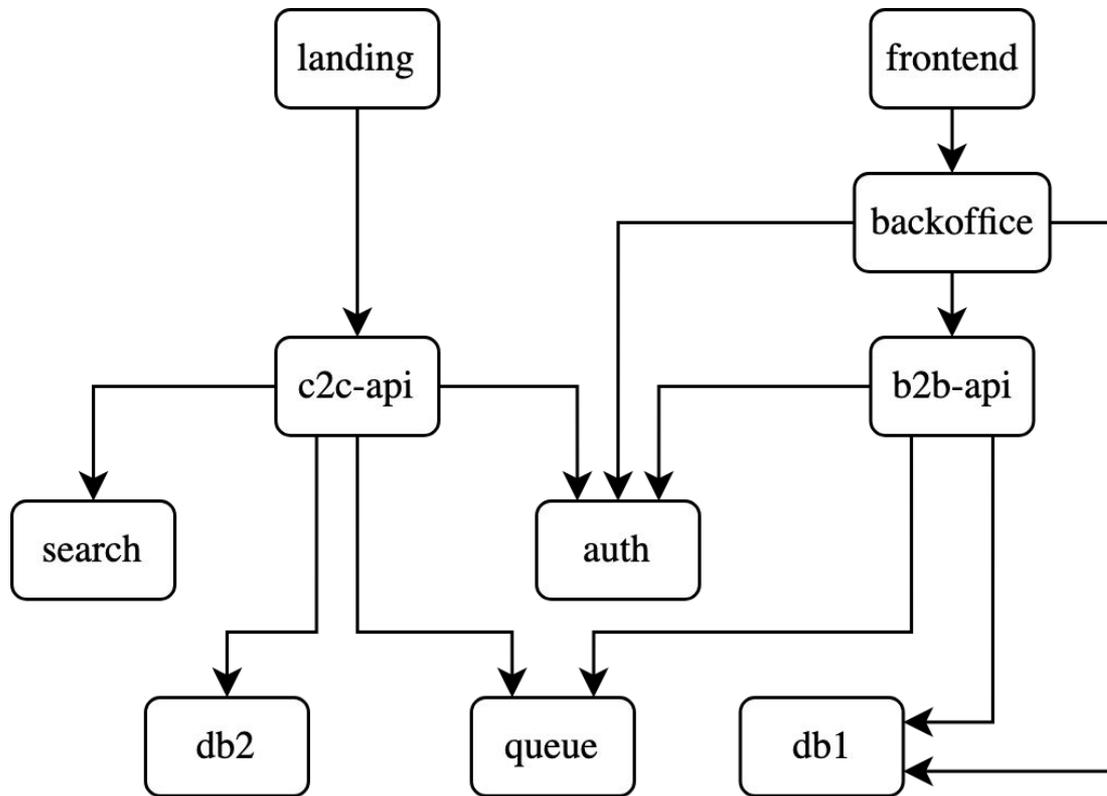
dev/feature-1

release/v1.2.3

Боль: Связи и зависимости



Боль: Связи и зависимости



Боли с Helm

- Никакой полезной информации во время деплоя
- Нужно оборачивать в bash
- Не IAC
- Низкий уровень DRY
- Секреты
- Umbrella chart
- Управление репозиториями

Предпосылки в 2020?

	Scripting	Helmfile	Helmsman	Terraform	Ansible	werf
Logs	~	-	-	~	-	+
Compose	-	+	+	+	+	-
Support	0	7	5	9	6	7
Fit	0	10	5	9	3	4

Нужен ли еще один инструмент?

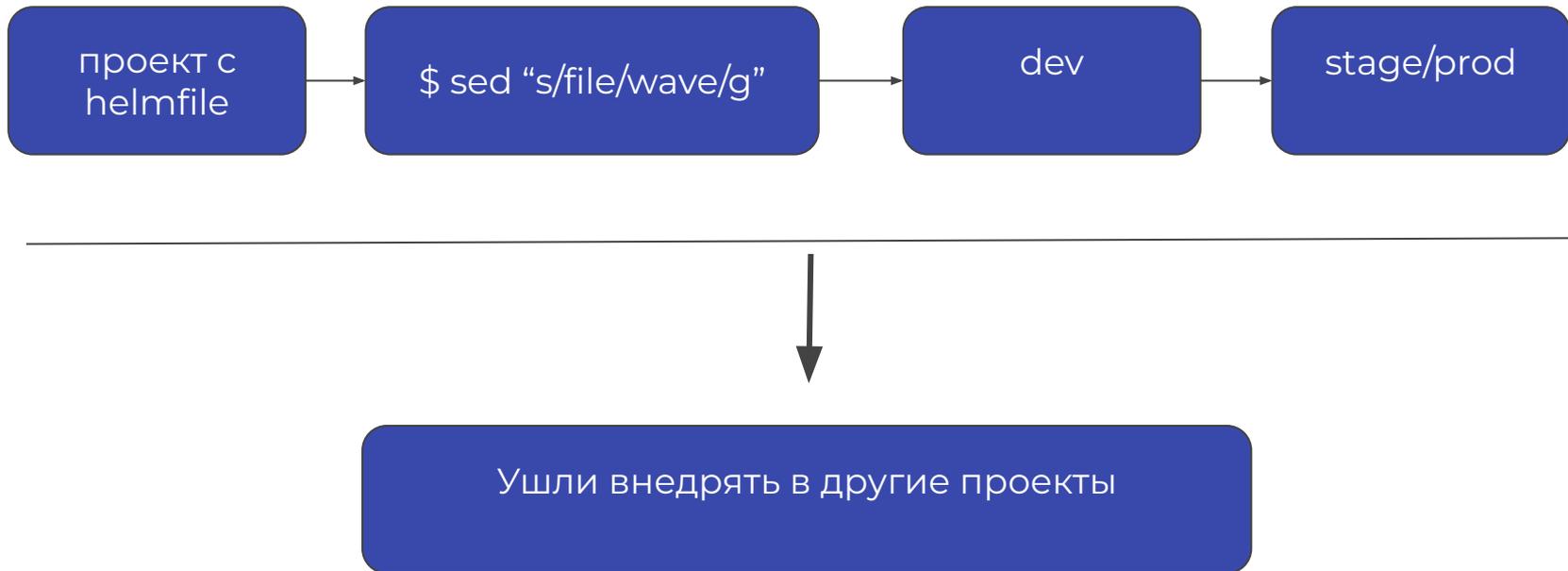
- Формулировка гипотезы: Быстрее? ~~Выше?~~ ~~Сильнее?~~
- MVP
- Тестовые запуски, сбор обратной связи
- Первые использования в проде
- Развитие

Гипотезы

Работа напрямую с объектами sdk helm/kubectl:

- Работает быстрее чем os.exec на 20%
- Параллелизация развертывания чартов
- Уменьшение оверлея при параллелизации
- Перехват helm событий
- Удобные логи
- Управление очередью релизов

Как внедряли MVP?



Features

Краткий список фичей

- Параллельность релизов / репозиториев
- Вывод логов
- Умеет ходить в vault / aws sm / etc
- Разделение values по окружениям
- Шаблонизация values
- Управление деплоем:
 - pending_release_strategy
 - depends_on
 - allow_failure
- И многое другое

Community 

Решаю проблему зависимости между чартами и разные наборы чартов для разных сред. Моя инициатива, других не знаю.

10:56

Через helmwave я запускаю создание джоба в кубере, а в нем уже запускается утилита для миграции наших приложений.

16:11

Ну у меня довольно базовая схема - просто собрать кучу чартов, сгруппировать их values и поставить. С этим вейв справился на 100%

14:42

Весной этого года внедрили у себя Helmwave и используем в CI/CD для разворота стендов где требуется поднимать несколько Helm чартов в разные неймспейсы.

Используется для деплоя всего стека. Пока только на новой платформе которая еще не доехала до внешних продов. Также для процессов разработки - так как позволяет использовать продовые конфиги в локальной разработке. В итоге покрывает и прод и в локальных стендах как полная замена докер-компоуза.

Стенды используем как для демо так и для тестирования. Т.е. одновременно в одном кластере может быть поднято от 1 до 5 стендов.

1) Отсутствие необходимости иметь и менеджить бинарь хелма и его плагины

2) классный и удобный рендеринг values

14:59

спасибо за проект, он должен очень сильно упростить мой CI :)

Ну мне понравилась хорошая раскраска в консоли, с нормальными акцентами и всё такое. Причем с первого запуска и из коробки, да

14:40

уже использую на нескольких проектах - полет нормальный. спасибо за развитие индустрии!)

09:40

спасибо за полезный проект.

Большое спасибо авторам за сей классный инструмент



08:36

Documentation

Требования к документации

- Наличие
- Красивая и удобная
- Базовые сценарии использования
- Актуальность
- Автогенерация новой версии
- Версионированность

Helmwave

Docker-Compose for helm

With helmwave you can:

- Templating values
- Select specific release or group of releases
- Parallel deploy
- Live tracking k8s resource while deploy
- ... and much more!



[telegram chat](#)

[Get started](#)

Workshop time!

releases:

```
{{ with readFile "vars/products.yaml" | fromYaml | get "releases" }}
```

```
{{ range $v := . }}
```

```
- name: {{ $v | get "name" }}
```

```
  chart: {{ $v | get "name" }}
```

```
<<: *options
```

```
tags:
```

```
- {{ $v | get "name" }}
```

```
- product
```

```
values:
```

```
# all products & all envs
```

```
- values/product/_/_ .yaml
```

```
# all products & an env
```

```
- values/product/_/{{ requiredEnv "CI_ENVIRONMENT" }}.yaml
```

```
# a product & all envs
```

```
- values/product/{{ $v | get "name" }}/_ .yaml
```

```
# a product & an env
```

```
- values/product/{{ $v | get "name" }}/{{ requiredEnv "CI_ENVIRONMENT" }}.yaml
```

```
{{ end }}
```

```
{{ end }}
```

variables:

```
HELMWAVE_TAGS: "ns-ready, $CI_PROJECT_NAME, rabbitmq, postgresql, minio, chat"
```

Production env

deploy to prod:

variables:

```
HELMWAVE_TAGS: $CI_PROJECT_NAME
```

Staging env

deploy to stage:

variables:

```
HELMWAVE_TAGS: "ns-ready, $CI_PROJECT_NAME"
```

```
$ tree helmwave/values/infrastructure
```

```
helmwave/values/infrastructure
```

```
├── adminer  
│   ├── _ .yml  
│   ├── dev.yml  
│   ├── release.yml  
│   └── stage.yml  
├── postgresql  
│   ├── _ .yml  
│   └── dev.yml  
└── rabbitmq  
    ├── _ .yml  
    ├── dev.yml  
    ├── release.yml  
    └── stage.yml
```

```
$ tree helmwave/values/product
├── product
│   ├── -
│   ├── _ .yml
│   ├── dev.yml
│   ├── prod.yml
│   ├── release.yml
│   └── stage.yml
│   ├── api
│   │   ├── _ .yml
│   │   ├── dev.yml
│   │   ├── perf.yml
│   │   ├── prod.yml
│   │   ├── release.yml
│   │   └── stage.yml
│   └── frontend
│       ├── dev.yml
│       ├── perf.yml
│       ├── prod.yml
│       ├── release.yml
│       └── stage.yml
```

```
$ echo $HELMWAVE_TAGS
ns-ready, api, rabbitmq, postgresql, minio, chat
$ cd /opt/helmwave
$ helmwave yml
[🤖 aka INFO]: 📄 YML is ready!
      build plan with next command: helmwave build -f helmwave.yml
$ helmwave build
[🤖 aka INFO]: Building releases...
[🤖 aka INFO]: Building graphs...
[🤖 aka INFO]: Depends On:

[ api@uprovid... ]

[ chat@uprovi... ]

[ postgresql@... ]
```

uprovider-2633, postgresql, Secret (v1) has changed:

...

app.kubernetes.io/instance: postgresql

app.kubernetes.io/managed-by: Helm

namespace: uprovider-2633

type: Opaque

data:

- postgresql-postgres-password: "YTREaFUyVW5FSw=="

+ postgresql-postgres-password: "cWR5NGZNRXQ2QQ=="

postgresql-password: "bG9jYWw="

```
$ helmwave up
```

```
[👹 aka WARNING]: this flag is used by autobuild (--build) but autobuild is disabled  
flag: plandir
```

```
[👹 aka WARNING]: this flag is used by autobuild (--build) but autobuild is disabled  
flag: tags
```

```
[🙄 aka INFO]: 🛠️ Plan
```

releases:

- api@uprovider-2633
- chat@uprovider-2633
- postgresql@uprovider-2633
- rabbitmq@uprovider-2633
- minio@uprovider-2633
- ns-ready@uprovider-2633

repositories:

- bitnami
- cetic
- sre

registries:

-

```
[👹 aka WARNING]: 🐶 kubedog is enable
```

```

└─ Status progress
  | JOB                                     ACTIVE          DURATION
SUCCEEDED/FAILED
  | api-migrate                             0              68s
0->1/0
  | |   POD                                READY          RESTARTS      STATUS
  | └─ migrate-ncvfz                       0/1            0             Running->Completed
  | api-optimize                             1              1s
0/0
  | |   POD                                READY          RESTARTS      STATUS
  | └─ optimize-hzb5q                       0/1            0             ContainerCreating
e terminated, succeeded 0->1
Waiting for: pods should b

```

```

└─ job/api-optimize service messages
  | event: JobIsComplete: Pod exits with success, the job is complete
  | [👉 aka INFO]: ✅
  |           release: api@uprovider-2633
  | [👉 aka INFO]: Success 6 / 6

```

Планы на будущее

- Major release 1.0.0
- Chart.lock в planfile
- Kubectx
- Behavior scenarios
- Логи для rollback / uninstall операций

<https://github.com/helmwave/helmwave/issues>

Каким должен быть CD инструмент в 2022?

- Накатываться по шагам
- Логи / события при развертывании
- Лок параллельного деплоя в том числе при развертывании зависимостей
- Поведенческие сценарии
 - Единый интерфейс поведений и статусов
 - Расширенные статусы:
uninstall/deploying/waiting/pending/scalingUp/scallingDown/?etc
 - Управление сценариями по статусу
- Зависимости
- Параллелизация
- Ваши предложения / замечания

411

<https://github.com/helmwave/helmwave/>



Star



Беспокойство о плохих практиках

- Нужен ли helmwave швейцарский нож?
- При проектировании мы озаботились, тем как людям избегать костылестроение?
- Как уложить их процессы в helmwave?
- Как helmwave должен интуитивно помочь людям их избегать?
- В чём именно заключается хорошая/плохая практика?
- Простота на “уровне пальцев”.

--set в yaml

Запрещён

Разрешён

Плагины