

# Helm — шаблонизация и переиспользование

Цели:

- Понять, что такое Helm и зачем он нужен.
- Разобрать структуру Helm Chart.
- Установить Helm.
- Создать простой чарт для Nginx.

# Зачем нужен Helm

Факты:

- Kubernetes-манифесты быстро усложняются.
- Для одного сервиса может быть 5–10 YAML-файлов.
- Разные окружения требуют разных значений.

Идея Helm:

Управление Kubernetes-приложениями через шаблоны + версиями релизов.

# Что такое Helm Chart

## Определение:

Helm Chart — пакет, в котором лежат шаблоны Kubernetes-манифестов и значения для этих шаблонов.

## Chart включает:

- Структуру каталогов.
- Логику шаблонов.
- Настройки для разных окружений.

## Основные термины

- **Chart** — пакет приложения (аналог «проекта»).
- **Release** — установленный экземпляр чарта.
- **Values** — переменные, передаваемые в шаблоны.
- **Templates** — YAML с шаблонизацией.

# Структура Helm Chart

Helm создаёт базовый каркас:

```
mychart/
  Chart.yaml
  values.yaml
  templates/
    deployment.yaml
    service.yaml
    _helpers.tpl
```

# Назначение ключевых файлов

## Chart.yaml

Описание чарта: имя, версия, dependencies.

## values.yaml

Настройки по умолчанию, которыми управляет пользователь.

## templates/

Шаблоны, из которых Helm генерирует манифесты.

## \_helpers.tpl

Функции и шаблоны для переиспользования.

## Пример параметров в values.yaml

```
replicaCount: 2

image:
  repository: nginx
  tag: "1.25"
  pullPolicy: IfNotPresent

service:
  type: ClusterIP
  port: 80
```

# Пример шаблонизации в Deployment

```
spec:  
  replicas: {{ .Values.replicaCount }}  
  template:  
    spec:  
      containers:  
        - name: nginx  
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
```

## Преимущества Helm

- Переменные вместо копирования YAML.
- Готовые шаблоны, которые можно переиспользовать.
- Версионирование релизов.
- Установка и обновление в 1 команду.
- Простое управление окружениями.

# Установка Helm

Команды основаны на официальной инструкции.

Linux/macOS:

```
curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash
```

Проверка:

```
helm version
```

# Создание чарта Nginx

Создать новый чарт:

```
helm create nginx-demo
```

После этого в каталоге появится базовая структура.

# Настройка чарта под Nginx

1. В `values.yaml` установить нужный образ:

```
image:  
  repository: nginx  
  tag: "latest"
```

2. Проверить шаблоны:

```
helm template nginx-demo/
```

## Установка чарта в кластер

Команда установки:

```
helm install nginx-demo ./nginx-demo
```

Проверить релиз:

```
helm list
```

## Обновление релиза

Изменить tag в values.yaml => запустить:

```
helm upgrade nginx-demo ./nginx-demo
```

Отследить историю:

```
helm history nginx-demo
```

## Удаление релиза

Удалить установленный релиз:

```
helm uninstall nginx-demo
```

Удалить директорию чарта (опционально).

## Практика (структура занятия)

1. Установка Helm.
2. Создание чарта `nginx-demo`.
3. Настройка `values.yaml`.
4. Деплой чарта в Kubernetes.
5. Обновление и удаление релиза.

