

# K8S 48: Kubernetes. Helm. Введение

## Описание:

В процессе работы и доставки кода в инфраструктуру kubernetes инженеры сталкиваются с типовым набором проблем, которые необходимо решить, чтобы указанные процессы были надежны и вариативны. Вы уже представляете себе, как задеплоить приложение в k8s, но давайте попробуем представить, какие проблемы нам также потребуется решить, когда мы начнем обслуживать приложение, которое мы разместили в k8s:

1. Возможность отката. Самый первый и острый вопрос, который мы должны рассмотреть. Если мы настраиваем инструмент для доставки кода, то вопрос возможности максимально быстрого отката до определенного состояния возникает сам собой. Да, разумеется в kubectl есть возможность откатить состояние допустим deployment'a, но попробуйте это автоматизировать, и вы сами все поймете.
2. Вариативность. Как следствие первого пункта и как отдельная проблема. В случае внесения изменений в функционирование контейнера с приложением, изменения переменных окружения или каких-либо иных аспектов, связанных с работой нашего кода, мы должны иметь удобный интерфейс, с помощью которого можно было произвести эти изменения. Самый базисный вопрос — версионирование tag'a image контейнера с кодом, из container registry. Sed'ы?
3. Удобство использования. Конечно, можно выкрутиться и kubectl apply -f a lot of files, но чем больше элементов, с которыми мы работаем, тем больше потенциальных возможностей для ошибок.

Helm позволяет решить указанные проблемы. Мы можем собрать требуемое описание ресурсов в едином месте, шаблонизировать необходимые аспекты для удобной работы (совсем как ansible templates) и работать с полученной группой ресурсов как с единым элементом.

Примечание Программа практикума умышленно подразумевает использование версии helm 2.x.x, так как работа с 3.x.x после изучения второй версии не вызовет вопросов после прохождения данного курса, а при обслуживании реальных проектов вы можете столкнуться с legacy.

## Полезные ссылки:

- [Helm Docs](#)

## Задание:

1. Установите helm command line utility на ваш рабочий ПК (версии 2.x.x).

2. Произведите установку helm-tiller в kube-system, согласно официальной документации - [installing helm](#).
3. Продемонстрируйте набором команд успешное выполнение предыдущих пунктов (команды и выводы сохранить)
4. На проверку отправьте все сохраненные команды и выводы.