

# K8S 46: Kubernetes. Kubectl. Node management

## Описание:

В процессе обслуживания кластера может возникнуть необходимость в перезагрузке worker nodes (например, для обновления kernel, libc, восстановления оборудования и т.д.). Если при этом время простоя будет коротким, то после перезапуска Kubelet на ноде, Kubelet попытается запустить запланированную на этой ноде рабочую нагрузку (pods). Если же перезагрузка занимает продолжительное время (контролируемое параметром pod-eviction-timeout), тогда controller-manager завершит работу pods на указанной ноде. Если их состояние описано в replica set, то требуемые поды будут запущены на других узлах кластера. Таким образом, если мы описали требуемое количество подов, контроль за их состоянием происходит непрерывно, и в случае доступных нод controller manager будет инициировать их запуск.

Однако ряд проводимых работ может влиять на корректность работы пользовательской нагрузки на затронутых нодах. И в kubernetes есть специальные инструменты, отличные от taint and toleration, с помощью которых можно отметить ноду, на которой мы собираемся проводить работы, как unshedule, и очищать ее от пользовательской нагрузки. Taint and toleration требуются для описания постоянной логики функционирования кластера, но если нужно более простое, временное, решение для проведения запланированных работ, то на помощь приходят команды drain/cordon/uncordon node.

## Полезные ссылки:

[Cluster management](#)

## Задание:

1. Привяжите к ноде X label с произвольной парой key=value, отличающийся от уже имеющихся в кластере.
2. Уберите ноду из балансировки пользовательской нагрузки и очистите ее командой drain.
3. Попробуйте задеплоить под с NodeSelector: key=value из п.1 и приведите ошибку из describe состояния пода как ответ к текущему заданию.