

# K8S 45: Kubernetes. Manifests. Taints + tolerations

## Описание:

nodeAffinity позволяет повысить вероятность выбора узла для запуска pod. taint работает наоборот — позволяет исключить узел из возможных для запуска. taint и tolerations работают совместно, обеспечивая исключение определенных узлов из списка подходящих для запуска. Один или несколько taints вешаются на узел, чтобы убедиться, что pod, который не допускает этих ограничений, никогда не запускался на этом узле. tolerations применяются к Pod и разрешают (но не требуют) запуск на определенном узле с определенным taint. Посмотрите на taint мастера `kubectl describe node master`. taint формируется по правилу `<key>=<value>:<effect>`.

Пример назначенного на node taint:

```
node-role.kubernetes.io/master:NoSchedule
```

В этом примере value == null, effect == NoSchedule.

Это означает, что pod без описания tolerations для данного taint никогда не будет запущен на мастере. А это почти любой pod. Данное ограничение можно обойти:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: prod
spec:
  replicas: 5
  template:
    spec:
      ...
      tolerations:
      - key: node-role.kubernetes.io/master
        Operator: Exists
        effect: NoSchedule
```

В этом случае pod может попасть на узел мастера.

Назначение taint возможно как через `kubectl edit`, так и через `kubectl taint`. Пример:

```
kubectl taint node node1.k8s node-type=production:NoSchedule
```

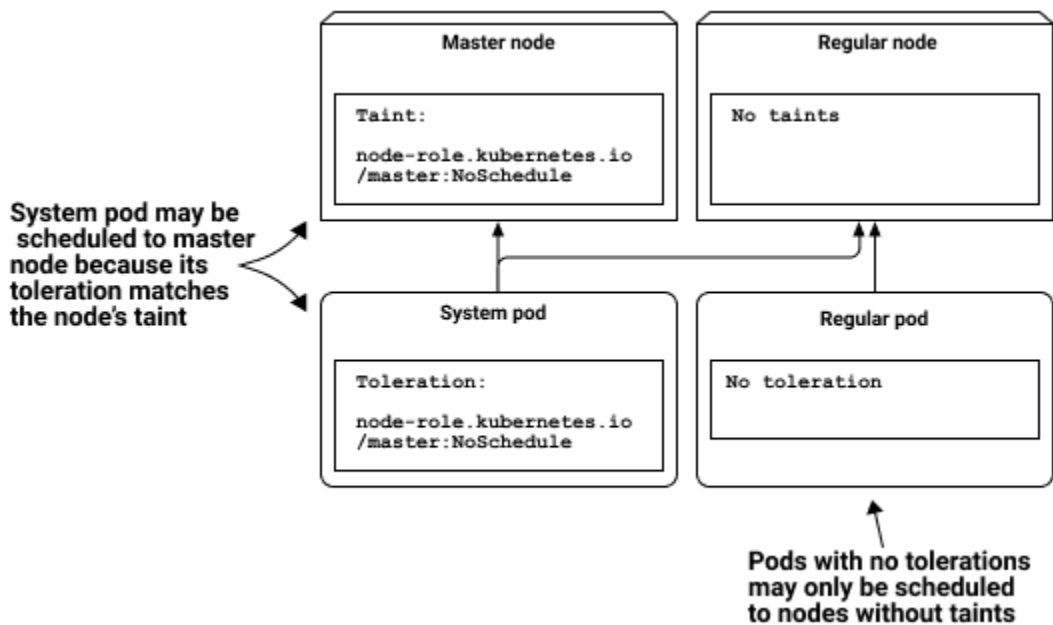
А затем обходить эти ограничения для определенных pod:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
```

```

name: prod
spec:
  replicas: 5
  template:
    spec:
      ...
  tolerations:
    - key: node-role.kubernetes.io/master
      Operator: Exists
      effect: NoSchedule

```



Есть три значения для effect:

- NoSchedule — поды никогда не будут назначены на данный узел.
- PreferNoSchedule — scheduler попытается избежать этого узла, но, если других доступных для запуска узлов не останется, будет выбран данный.
- NoExecute — не похож на два предыдущих ограничения, влияющих только на планировщик. Если повесить это ограничение на узел, то все pod, не обходящие данное ограничение, будут перемещены с узла.

Дополнительный параметр `tolerationSeconds` позволяет добавить время, в течение которого pod может обходить данное ограничение.

`tolerations:`

```

- effect: NoExecute
  key: node.alpha.kubernetes.io/notReady
  operator: Exists
  tolerationSeconds: 300

```

```
- effect: NoExecute
  key: node.alpha.kubernetes.io/unreachable
  operator: Exists
  tolerationSeconds: 300
```

В первом примере pod может 300 секунд находиться на узле с taint notReady, прежде чем будет перемещен на другой узел. Во втором примере это же правило применяется для узла с taint Unreachable.

## Полезные ссылки:

- [Taints and Tolerations \(official docs\)](#)
- [Herding pods: taints, tolerations and affinity in kubernetes](#)

## Задание:

1. Установите на ноде с меткой task: nodeAntiAffinity taint rbmTaint=tolerate:NoSchedule.
2. Перезапустите поды запущенного Deployment rbm-nodeAffinity (команды и вывод сохраните).
3. Выведите список подов, связанных с этим Deployment, с параметром -o wide (команды и вывод сохраните).
4. Добавьте к манифесту Deployment toleration, чтобы мы могли обойти taint на ноде, и уменьшите количество реплик до 1.
5. Примените манифест и выведите список подов, связанных с этим Deployment (команды и вывод сохранить).
6. На проверку отправьте манифест и все выводы.