

# KUB-05: Kubeconfig — подключаемся к кластеру

## Описание:

В прошлых заданиях мы с вами благополучно развернули три инсталляции кubernetes - две локальных и одну в облаке. Давайте сегодня разберемся с тем, как вы можете подключаться к различным кластерам Kubernetes, используя утилиту kubectl и файл kubeconfig.

Итак, Kubeconfig — это конфигурационный файл, в котором хранятся настройки доступа к кластеру k8s. По умолчанию, он находится по пути `$HOME/.kube/config`, однако вы можете указать команде kubectl другой путь к этому файлу при помощи флага `--kubeconfig` или используя переменную окружения `KUBECONFIG`.

Давайте рассмотрим структуру kubeconfig:

```
clusters:
- cluster:
  certificate-authority-data: ...
  server: IP
  name: cluster1

users:
- name: user1
  user:
    exec:
      args:
      - k8s
      - create-token
      - --profile=default
      command: /usr/local/bin/yc

- name: user2
  user:
    client-certificate-data: LS0tLS1CRUdJTi...
    client-key-data: LS0tLS1CRUdJTi...

- name: user3
  user:
    token: gjkrenjejjtenjwth
```

```
contexts:
- context:
  cluster: cluster1
  user: user1
  name: context1
```

В данном файле описываются три основных списка:

1. `clusters` — здесь описываются кластеры, к которым мы можем подключаться. Для них указывается адрес для подключения, включая порт, а также Certificate Authority, чтобы `kubectl` смог проверить, что мы действительно подключились к тому кластеру, используя TLS-соединение.
2. `users` — здесь описаны пользователи, которые будут использованы для подключения к серверу. У пользователя могут задаваться различные типы аутентификации:
  - сертификат и клиентский ключ - тогда пользователь будет проходить аутентификацию с помощью сертификатов;
  - указан параметр `exec`, которая запустит команду с аргументами и получит временный ключ доступа к кластеру;
  - указан `token`, который будет использован для аутентификации. Второй путь более безопасный (`exec`) и именно он часто используется в облачных установках `kubernetes`. Временный ключ чаще всего выдается с помощью утилит для работы с облаком - `aws-cli`, `gcloud`, `us` и других.
3. `contexts` — по сути, это профили. Здесь связывается кластер с пользователем, который будет использован для подключения.

Также можно посмотреть все настроенные контексты:

```
kubectl config get-contexts
CURRENT   NAME                                     CLUSTER
AUTHINFO                                     NAMESPACE
*         kubernetes-admin-cluster.local@cluster.local
cluster.local  kubernetes-admin-cluster.local
```

Звездочкой отмечается текущий активный контекст. Переключаться между ними можно, используя команду `kubectl config use-context <context_name>`.

Если вы работаете с `kubernetes` кластерами, то вы вполне часто будете использовать команды для переключения контекста - `use-context` & `get-contexts`, поскольку это вполне распространенный кейс - для продуктовой среды применять один кластер, а для тестовых сред другой. Таким образом вы сможете обкатать обновление мастер нод и групп узлов перед обновлением продуктовой среды.

## Полезные ссылки:

- [Accessing to multiple clusters](#)

## Задание:

1. На подготовленной виртуальной машине установите kubectl.
2. Создайте тестовую конфигурацию kubeconfig, содержащую (за основу можно взять конфигурацию, которую сгенерировал kubespray в 3ем задании):
  - кластер с именем cluster-old
  - кластер с именем cluster-new
  - пользователя с именем user
  - контекст с именем context-old и использующий кластер cluster-old и пользователя user
  - контекст с именем context-new и использующий кластер cluster-new и пользователя user
4. Переключитесь на использование контекста context-new.
5. Положите конфигурационный файл в директорию /root/.kube/ с именем config и отправьте задание на проверку.