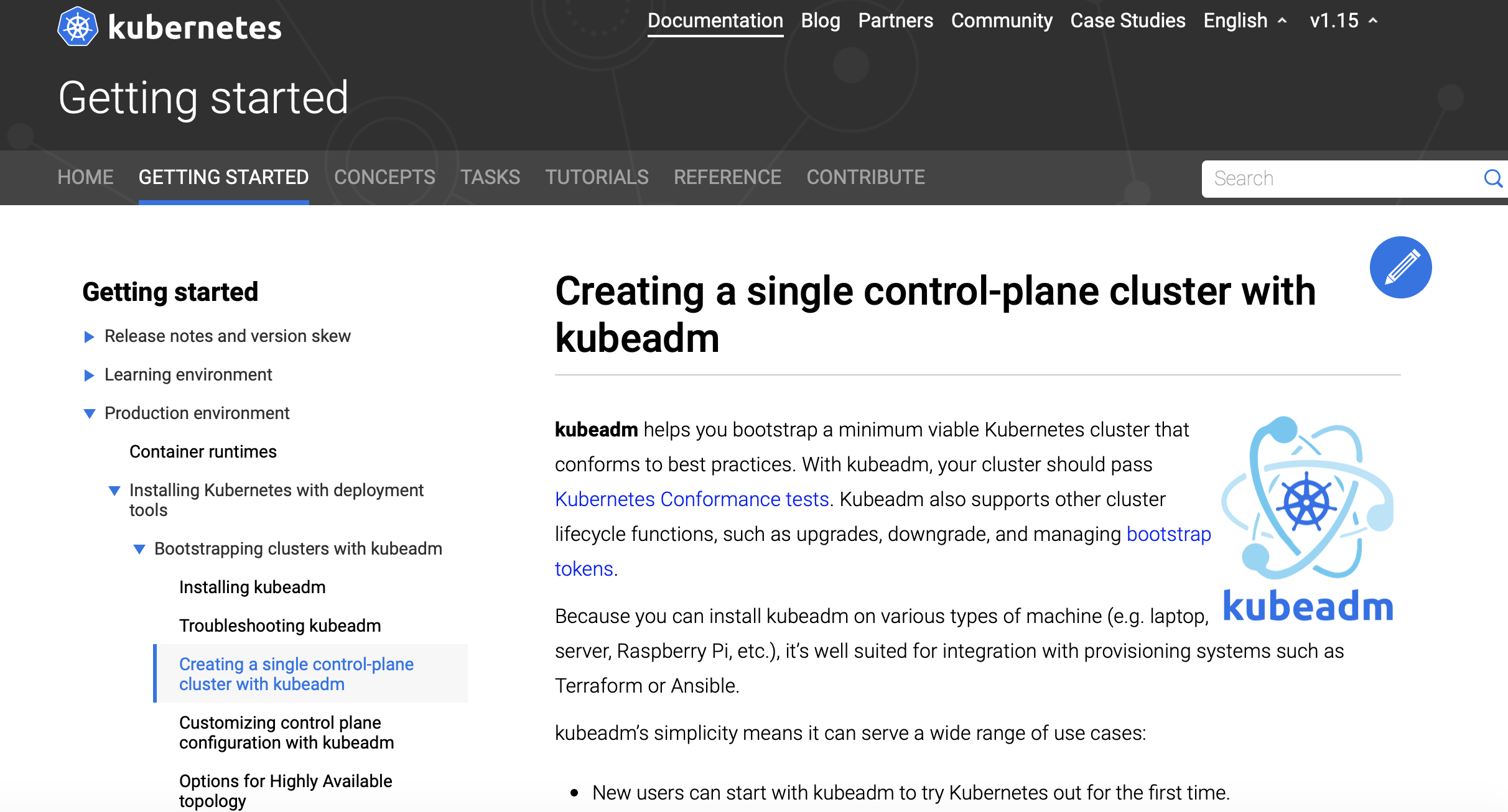
**Установка k8s**

Обзор:

Есть несколько инструментов установки Kubernetes, предоставляемых различными поставщиками. В этой лабораторной работе мы научимся использовать kubeadm. Планируется, что как независимый инструмент, поддерживаемый сообществом, он станет основным способом построения кластера Kubernetes.

Детальные инструкции доступны по ссылке:

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/>



1. сделайте sudo и обновите repo

apt-get update && apt-get upgrade -y

1. убедитесь что docker установлен

docker info

1. добавить k8s repo

vim /etc/apt/sources.list.d/kubernetes.list

deb http://apt.kubernetes.io/ kubernetes-xenial main

1. Добывить a GPG ключ для пакетов. Команда из 3 строк, можно удалить слеш и написать в одну строку. Ожидаемый вывод OK.

curl -s \

https://packages.cloud.google.com/apt/doc/apt-key.gpg \

| apt-key add -

OK

1. Обновите репо

apt-get update

1. Установите софт.

apt-get install -y \

kubeadm kubelet kubectl

Также добавьте запись в /etc/hosts где ntmk-ubuntu-ksm имя машины

127.0.1.1 ntmk-ubuntu-ksm

1. Решая, какую сеть подов использовать для сетевого интерфейса контейнера (CNI), следует учитывать ожидаемые требования к кластеру. В кластере может быть только одна сеть. Сеть должна обеспечивать связь контейнер-контейнер, под-под, под-сервис и внешний-сервис. Поскольку Docker использует частную сеть хоста, использование виртуального моста docker0 и интерфейсов veth потребуется на этом хосте для связи.
2. Мы будем использовать Calico в качестве сетевого плагина, который позволит нам использовать сетевые политики позже в ходе курса. В настоящее время Calico по умолчанию не выполняет развертывание с использованием CNI. Версия Calico 3.3 имеет более одного файла конфигурации для обеспечения гибкости с RBAC. Загрузите файлы конфигурации для. После загрузки найдите ожидаемый диапазон IPV4 для контейнеров.

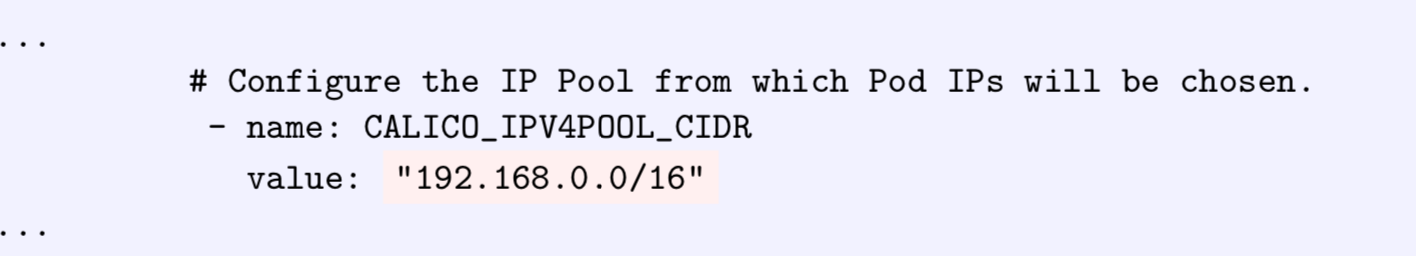
1. Скачайте файлы:

wget https://tinyurl.com/yb4xturm -O rbac-kdd.yaml

wget [https://tinyurl.com/y8lvqc9g -O calico.yaml](https://tinyurl.com/y8lvqc9g%20-O%20calico.yaml)

1. Просмотрите файл и найдите пул IPV4 адресов назначенный на контейнеры. Посмотрите различные параметры и проверьте CALICO\_IPV4POOL\_CIDR – он должен совпадать с тем идет на вход команде **kubeadm**

vi calico.yaml



1. Инициализируйте мастер. Прочитайте вывод. Обратите внимание на token .Его также можно в дальнейшем получить используя **kubeadm token list**. The output also directs you to create a pod network to the cluster, which will be our next step. Pass the network settings **Calico** has in its configuration file, found in the previous step.

kubeadm init \

--kubernetes-version 1.21.1 \

--pod-network-cidr 192.168.0.0/16 \

# This creates the cluster

# To avoid using newest possible

# The IP range we found in Calico file

| tee kubeadm-init.out # Save output for future review

Также параметры можно передать через конфиг файл, его можно инициализировать так.

kubeadm config print init-defaults

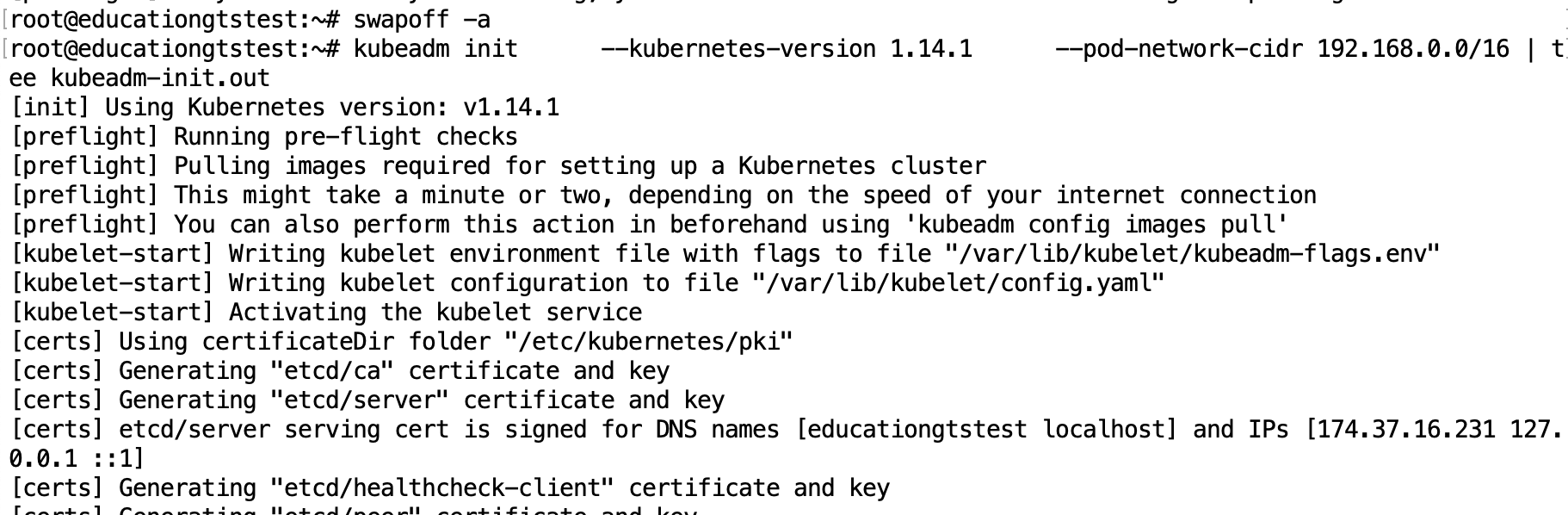
В случае с ошибкой cgroups driver используйте инструкцию отсюда

<https://kubernetes.io/docs/setup/production-environment/container-runtimes/>

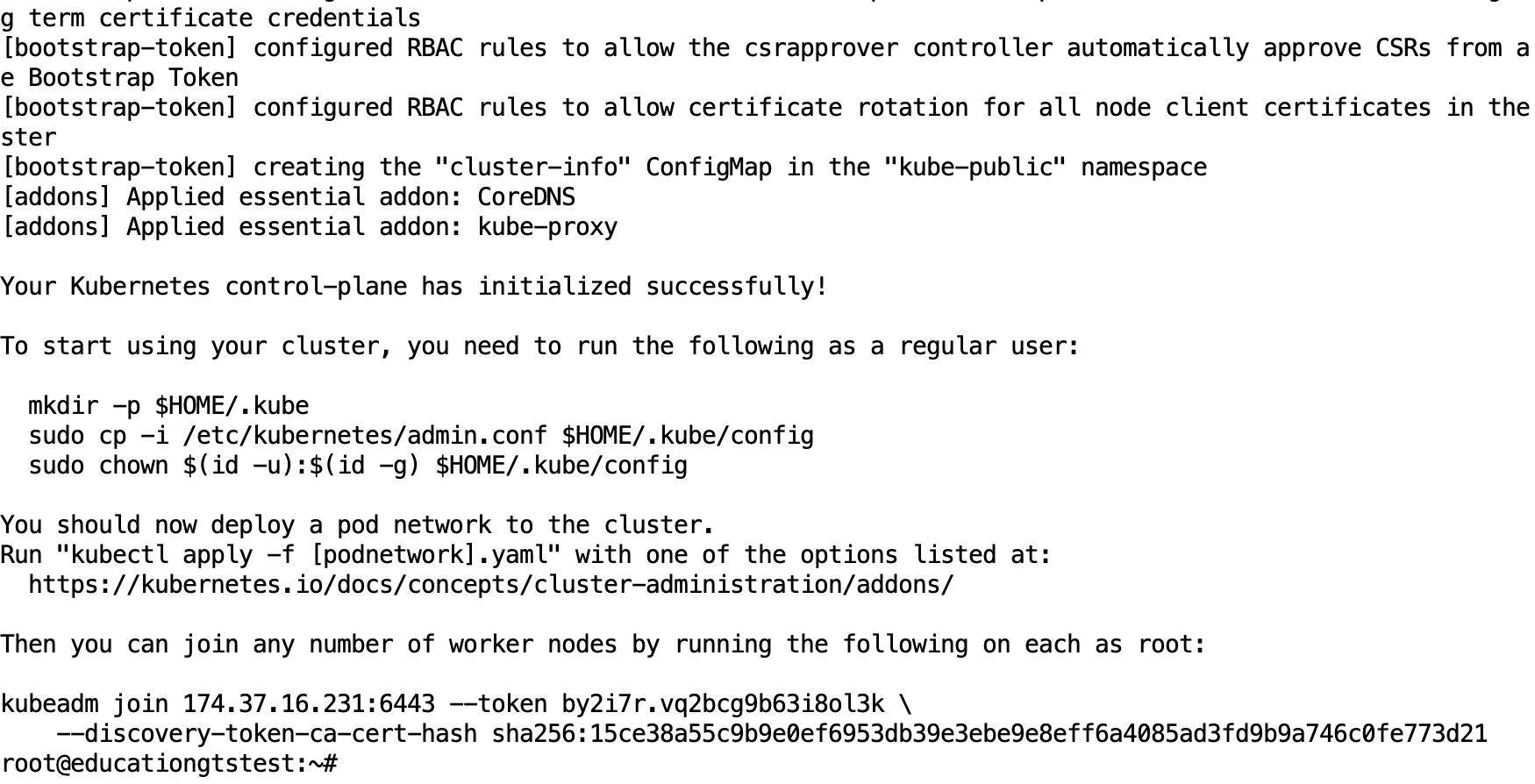
В случае с ошибкой swap error – сделайте

sudo swapoff -a

Вывод будет примерно такой:



>>>>>>>>>>>



1. Примените конфигурацию сетевого плагина к вашему кластеру.

kubectl apply -f rbac-kdd.yaml

kubectl apply -f calico.yaml

1. При частом использовании kubectl удобно сделать -включить автозаполнение bash. Начните с добавления настроек в текущую оболочку. Затем обновите файл /.bashrc, чтобы он стал постоянным.

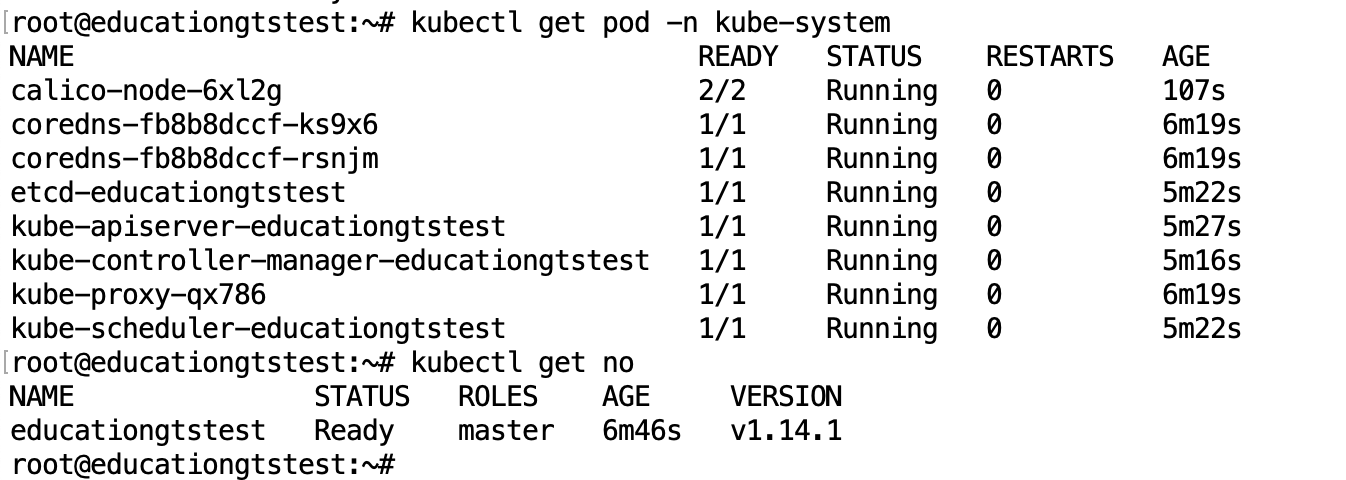
source <(kubectl completion bash)

echo "source <(kubectl completion bash)" >> ~/.bashrc

Теперь можно использовать табуляцию.

kubectl des<Tab> n<Tab><Tab> od<Tab>

1. Убедитесь что системные поды kube-system в статусе ready



1. Чтобы подключить другие узлы используйте команду kubeadm join вместе с токеном который был выведен в процессе установки kubeadm init

kubeadm join 174.37.16.231:6443 --token by2i7r.vq2bcg9b63i8ol3k \

--discovery-token-ca-cert-hash sha256:15ce38a55c9b9e0ef6953db39e3ebe9e8eff6a4085ad3fd9b9a746c0fe773d21

1. Посмотреть узлы кластера можно командой

kubectl get no

kubectl get no -o wide

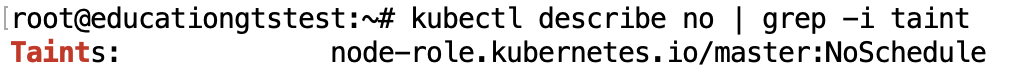
1. Детали можно найти используя команду.

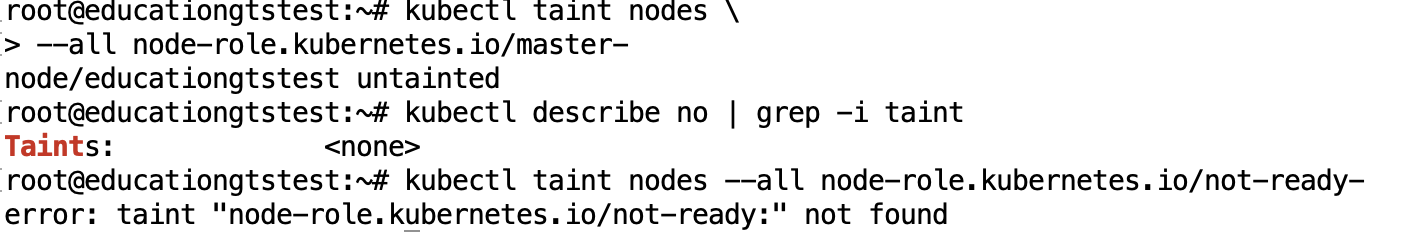
Kubectl describe no master\_name

Kubectle describe no worker\_name

1. Разрешить запускать pod на мастер сервере.

The master node begins tainted for security and performance reasons. Note the minus sign(-) at the end, which is the syntax to remove a taint. As the second node does not have the taint you will get a not found error.





1. Используйте команду, чтобы отменить изменения, сделанные командами kubeadm init или kubeadm join. Команда на рабочем узле:

kubadm reset

1. Можно удалить узел

kubectl get no

kubectl delete no worker\_node\_name

1. Add worker node to the cluster again by kubeadm join command
2. Worker node can be made so that is cannot be scheduled, which stops and new work from being dispatched to the node, but leaves running work intact. Use kubectl cordon command:

kubectl cordon worker\_node\_name

See the status

kubectl get nodes

When you ready to resume dispatching work on worker node use the kubectl uncordon command

kubectl uncordon worker\_node\_name

1. Temporarily remove a worker node for maintenance

It’s possible to temporarily remove a worker node so that the node can be patched. And so on, and then re-enable the node in the cluster. Use the kubectl drain command:

kubectl get no

kubectl drain worker\_node\_name --delete-local-data --ignore-daemonsets=true –force

kubectl get no

The use of the drain command marks the node as unavailable for scheduling new work, and then evicts any running pods that are on that node. Daemonsets cannot be evicted by using the kubectl drain command. Specifying –ignore-daemonsets=true tells the command it is safe to proceed with eviction and ignore DaemonSets.

Now, the node is no longer managed by the cluster and any tasks that were planned can be performed.

After the node is made operational again, run the kubectl uncordon command to reinstate the node.

kubectl uncordon worker\_name

1. Removing working node

When you no longer need particular node in your k8s cluster, this process remove the worker node is the same as removing node for a maintenance.

On master node issue:

kubectl drain (see command above)

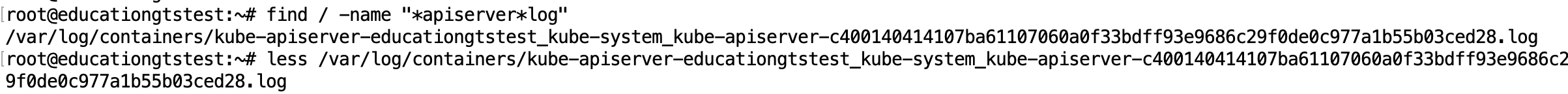
when all of the pods and services on the subject node are gone, the node can be deleted

kubectl delete node worker\_node\_name

kubectl get no

To clean Kubernetes information of the worker node that was removed, use kubeadm reset command.

1. See the logs



Search for and review log files for kube-dns, calico, and kube-proxy

find / -name "\*calico\*log"

find / -name "\*dns\*log"

find / -name "\*proxy\*log"

View the current Pods in the cluster.

kubectl get po --all-namespaces

kubectl -n kube-system logs <tab> <tab>

kubectl -n kube-system logs kube-apiserver-masternod\_name

kubectl get events