

# KUB 02: Развертывание Kubernetes-кластера с помощью minikube

## Описание:

Абсолютно понятно, что первое задание, которое может быть в любом практикуме, — это установка используемого инструмента. Мы не станем нарушать традицию и начнем с рассмотрения того, как вы можете установить чистый кластер Kubernetes в своей инфраструктуре.

Мы предлагаем начать наш практикум именно с разворачивания кластера Kubernetes, хотя по логике может показаться, что нам стоит сначала разобрать его архитектуру. Это было сделано для того, чтобы вы смогли потрогать руками все компоненты, о которых мы будем говорить в архитектурной части. Иначе получилась бы голая теория, что не очень правильно на наш взгляд.

В целом, есть два основных варианта разворачивания кластера:

- Self Hosted — разворачивание кластера на своих серверах, виртуальных или железных;
- Cloud Based — разворачивание кластера в любом публичном облаке, которое предоставляет указанную возможность.

Ниже приведены критерии, которые следует учитывать при выборе одного из двух указанных вариантов. На данный момент вы можете не до конца понять всю глубину данных вопросов, поэтому мы советуем после прохождения практикума еще раз вернуться к ним и взглянуть на них свежим взглядом.

- Высокая доступность — развертывает ли ваше решение Kubernetes-кластеры высокой доступности с репликацией базовых метаданных для восстановления после сбоев?
- Обновления — Kubernetes обеспечивает мажорные обновления каждые 3-4 месяца. Какова ваша стратегия обновления Kubernetes? Какие периоды простоя нужны для обновления и приемлемо ли это для бизнеса?
- Поддержка гибридных систем — поддерживает ли ваше решение Kubernetes узлы, расположенные в частных дата-центрах и облаке, на которых должен работать Kubernetes? Предлагает ли он одинаковый или аналогичный уровень SLA и функциональность по ним?
- Поддержка федеративности — поддерживает ли ваше решение Kubernetes развертывание федеративных кластеров, которые могут расти в частных и публичных облаках для обеспечения надежности инфраструктуры и динамической пакетной совместимости?

- Дополнительные возможности — какие дополнительные функции необходимы вашей команде для запуска Kubernetes в масштабе и поддержки большого числа пользователей? Поддерживаются ли они вашим решением Kubernetes? Некоторые примеры включают поддержку единого входа, RBAC, изолированную сеть, постоянное хранилище.

По нашему опыту, можем сказать, что самый простой и надежный способ развертывания kubernetes — это использовать облака, поскольку облачный провайдер берёт на себя ответственность за многие компоненты системы, о которых вам придется думать в self-hosted решениях. Основные из них:

- Etcd — как мы узнаем дальше, kubernetes использует кластер etcd для хранения своего состояния. В случае использования self hosted решения вам будет необходимо поддерживать его доступность самостоятельно, а также проводить работы по резервному копированию и восстановлению в случае сбоев.
- Networking — в случае развертывания kubernetes на своих мощностях вы столкнетесь с необходимостью настройки сетевой инфраструктуры, которая может быть не совсем тривиальна в зависимости от используемых сетевых плагинов kubernetes. О сетевой модели мы поговорим отдельно в следующих заданиях.
- Persistent Volumes — если вы захотите использовать подключаемые постоянные хранилища для хранения данных, вам придется использовать инструменты вроде serph для создания блочных устройств. Serph — это отдельный инструмент, по которому можно выпустить несколько курсов или книг. Так что знайте, что это добавит вам головной боли.

Единственный приемлемый вариант, при котором разумно развернуть kubernetes локально на своих машинках, — это обучение. Ну и конечно, тестирование. Хотя иногда это может быть и политикой компании - "Ничего не отдаем наружу!". Если вы хотите погонять Kubernetes локально, то доступно достаточно много вариантов:

## Minikube

Minikube — это рекомендованный официальный метод для изучения работы Kubernetes. Он позволяет установить однонодовый кластер в виртуальной машине практически с любым поставщиком (virtualbox, kvm, hyperv, vmware), а также позволяет запустить и на хосте при помощи Docker. Данный вариант хорошо подходит для запуска кластера с целью ознакомления, для тестирования новых версий кластера и инструментов для него. Аналогом этого решения можно рассмотреть Minishift, который представляет собой инструмент для запуска кластера OpenShift в манере, подобной Minikube.

Итак, давайте запустим minikube на ОС ubuntu. Minikube потребует среду виртуализации для запуска кластера kubernetes. Использовать можно, как обычные решения для виртуализации, например kvm, так и в docker-контейнеры, но для этого потребуется установить docker-ce на сервер, используя [данную](#) инструкцию. После этого можем приступить к установке minikube. Кстати, хорошая инструкция для работы с minikube лежит [здесь](#).

Устанавливаем:

```
# Устанавливаем minikube
```

```
curl -LO
https://storage.googleapis.com/minikube/releases/latest/minikube
-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube

# Создаем пользователя, под которым будем создавать кластер
useradd -s /bin/bash -m -G docker kub

# Запускаем minikube
su - kub
minikube start
🤗 minikube v1.15.1 on Ubuntu 20.04
✨ Automatically selected the docker driver
👍 Starting control plane node minikube in cluster minikube
🚚 Pulling base image ...
📦 Downloading Kubernetes v1.19.4 preload ...
>
preloaded-images-k8s-v6-v1.19.4-docker-overlay2-amd64.tar.lz4:
256.00 MiB

... skipped ...
```

По окончании установки можете проверить, какие контейнеры у вас были запущены, используя docker ps.

Теперь совершим обзорную экскурсию по кластеру. Для доступа к кластеру из консоли потребуется утилита kubectl. Эта утилита доступна для всех современных ОС.

Инсталляционные пакеты есть на [официальном сайте](#). Но при использовании minikube он может установить ее за нас. Так что давайте выполним несколько команд и посмотрим на наш кластер:

```
$ minikube kubectl -- cluster-info
Kubernetes master is running at https://192.168.49.2:8443
KubeDNS is running at
https://192.168.49.2:8443/api/v1/namespaces/kube-system/services
/kube-dns:dns/proxy
```

To further debug and diagnose cluster problems, use `'kubectl cluster-info dump'`.

Также можем посмотреть на ноды, которые подключены к кластеру:

```
$ minikube kubectl -- get nodes
NAME           STATUS    ROLES    AGE      VERSION
minikube      Ready    master   3m11s   v1.19.4
```

Отлично, теперь на этом кластере можно будет потренироваться, но пока перейдем к другим вариантам установки кластера.

## Kubernetes с kops

[kops](#) — это еще один инструмент CLI, который обеспечивает развертывание кластера Kubernetes с помощью декларативной спецификации, предоставленной пользователем. В настоящее время единственной поддерживаемой платформой развертывания является Amazon Web Services, но kops предлагает и другие преимущества, включая высокую доступность для ваших кластеров и дополнения для сетей и мониторинга, в том числе Flannel, Calico и Dashboard.

Пользователю не нужно изначально настраивать инфраструктуру, поскольку kops развернули необходимые ресурсы AWS, такие как экземпляры EC2, хранилище EBS и сеть VPC.

## Kubernetes с kubespray

[Kubespray](#) — это набор доступных плейбуков, инвентори и дополнительных инструментов для общих задач управления конфигурацией кластера Kubernetes.

Kubespray обеспечивает:

- высокодоступный кластер;
- поддержку большинства популярных дистрибутивов Linux;
- интеграционные тесты.

## Kubernetes с kubeadm

Возможности CLI kubeadm позволяют настраивать кластеры Kubernetes в виртуальной или физической инфраструктуре.

По нашему опыту, kubeadm предоставил более простой способ развертывания Kubernetes по сравнению с kops и kubespray.

## Rancher

[Rancher](#) изначально не умел работать с Kubernetes, а работал с Docker-контейнерами напрямую, но, начиная с ветки версии 2.x, умеет взаимодействовать и разворачивать Kubernetes-кластер через Web UI, работает практически с любым облачным поставщиком, равно как и на железных серверах.

## Все с нуля

Вы также можете развернуть свой кластер Kubernetes с нуля, не используя ни один из этих инструментов.

Страница [документации](#) Kubernetes содержит инструкции о том, как это можно реализовать.

Если ваша цель состоит в том, чтобы получить глубокое понимание Kubernetes, настройка его без помощи вышеуказанных инструментов может быть полезна, и мы настоятельно рекомендуем использовать именно этот метод для развертывания своего первого кластера.

Мы также рекомендуем ознакомиться с [Kubernetes The Hard Way](#) — учебником от Келси Хайтауэр.

Естественно, этот список не претендует на полноту, но именно эти решения являются наиболее распространенными и удобными для работы с кластерами от однонодовых до многонодовых инсталляций.

## Полезные ссылки:

- [Overview of kubeadm](#)
- [CNCF Cloud Native Interactive Landscape](#)
- [Installing Kubernetes with Minikube \(official docs\)](#)
- [Getting started \(official docs\)](#)
- [K3s - Lightweight Kubernetes](#)
- [MicroK8s - Zero-ops Kubernetes for workstations and edge / IoT](#)

## Задание:

1. Установите на подготовленную виртуальную машину docker-ce.
2. Установите на подготовленную виртуальную машину minikube.
3. Создайте пользователя с именем kubernetes, из-под которого будет создаваться кластер kubernetes (добавьте его в группу docker).
4. Установите кластер kubernetes, который будет слушать порт 8443.
5. Найдите в выводе команды `kubectl cluster-info` адрес, на котором запущен kubernetes master, и сохраните его в файл `/tmp/master.txt`. Адрес должен быть вида: `X.X.X.X:YYYYY`, то есть сохранить надо IP адрес и порт для подключения.
6. Отправьте задание на проверку.