

```
mysql_1 | 2020-10-02 10:36:56+00:00 [Note] [Entrypoint]: Temporary server stopped
mysql_1 |
mysql_1 | 2020-10-02 10:36:56+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.
mysql_1 |
mysql_1 | 2020-10-02T10:36:56.625333Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.21) starting as process 1
mysql_1 | 2020-10-02T10:36:56.633084Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql_1 | 2020-10-02T10:36:56.827180Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql_1 | 2020-10-02T10:36:56.910618Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld
/mysqlx.sock
mysql_1 | 2020-10-02T10:36:56.969173Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
mysql_1 | 2020-10-02T10:36:56.969326Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported f
or this channel.
mysql_1 | 2020-10-02T10:36:56.971715Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is acces
sible to all OS users. Consider choosing a different directory.
mysql_1 | 2020-10-02T10:36:56.989064Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.21' socket: '/var/run/mysqld/my
sqld.sock' port: 3306 MySQL Community Server - GPL.
wordpress_1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally
to suppress this message
wordpress_1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally
to suppress this message
wordpress_1 | [Fri Oct 02 10:36:59.459356 2020] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.38 (Debian) PHP/7.4.11 configured -- resuming normal operations
wordpress_1 | [Fri Oct 02 10:36:59.459411 2020] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

Это будет в основном техническая статья, так как постоянно хожу за этой информацией в поиск, а сейчас решил упростить себе задачу. Я подробно расскажу, как установить систему контейнеризации docker и docker-compose на CentOS 7 и 8. Тема распространенная и популярная, регулярно приходится устанавливать, настраивать и запускать контейнеры.

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, рекомендую познакомиться с **онлайн-курсом «DevOps практики и инструменты»** в OTUS. Курс не для новичков, для поступления нужно пройти .

Установка Docker Centos и docker-compose

Содержание

Введение

Установка Docker на Centos 8

Установка Docker в Centos 7

Установка docker-compose на Centos

Заключение

Помогла статья? Подписывайся на telegram канал автора

Введение

Думаю, не нужно каких-то подробных пояснений на тему того, что такое Docker и зачем он нужен. Сейчас этот продукт у всех на слуху. В частности, он используется по дефолту в популярном кластерном продукте по автоматическому развертыванию и управлению контейнерами - Kubernetes. По сути он

стал стандартом отрасли, хотя поддерживает работу не только на базе docker, но по факту в основном используют его. Я это все к тому, что docker с нами надолго :)

Пару тройку лет назад считалось, что docker лучше всего разворачивать на Ubuntu. С Centos были некоторые проблемы, но где-то в середине развития 7-й ветки они все были решены. Подробности я уже не помню, но по факту сейчас нет принципиальной разницы. Я запускаю контейнеры как на ubuntu, так и на centos. Обычно разработчики просят убунту, так как она им привычна. Лично я сам предпочитаю Centos. Там еще пока нет рекламы в syslog.

С установкой Docker на Centos 7 нет никаких проблем. С 8-й версией есть нюансы. Расскажу обо всем по порядку. Если у вас еще нет готовой системы, то вот мои статьи по теме - установка и настройка centos.

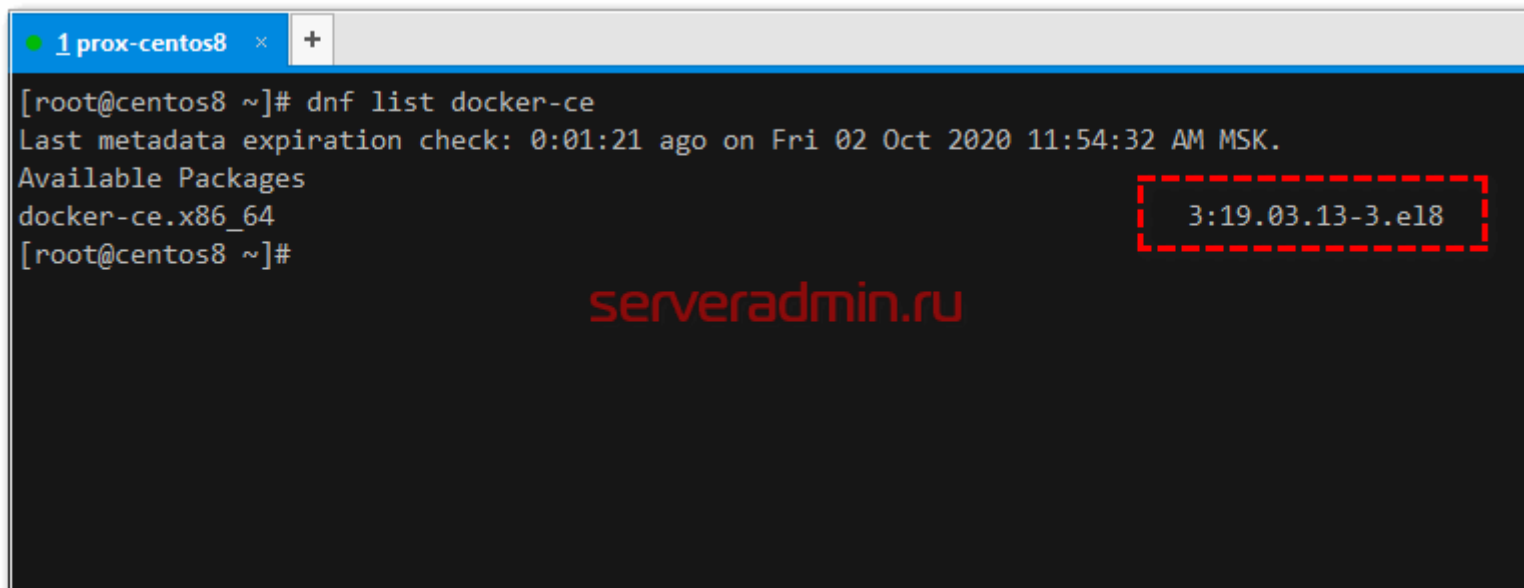
Установка Docker на Centos 8

Начинаем установку системы управления контейнерами Docker на Centos 8. Первым делом подключаем официальный репозиторий.

```
# dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
Adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
```

Система проинформирует, что репозиторий успешно подключен. Смотрим, какие версии докера доступны для установки.

```
# dnf list docker-ce
```



```
[root@centos8 ~]# dnf list docker-ce
Last metadata expiration check: 0:01:21 ago on Fri 02 Oct 2020 11:54:32 AM MSK.
Available Packages
docker-ce.x86_64                               3:19.03.13-3.el8
[root@centos8 ~]#
```

Устанавливаем Docker на Centos 8.

```
# dnf install docker-ce
```

```
[root@centos8 ~]# dnf install docker-ce
Last metadata expiration check: 0:02:25 ago on Fri 02 Oct 2020 11:54:32 AM MSK.
Dependencies resolved.
=====
Package                Architecture          Version               Repository            Size
=====
Installing:
docker-ce              x86_64               3:19.03.13-3.el8     docker-ce-stable     24 M
Installing dependencies:
container-selinux      noarch               2:2.124.0-1.module_e18.2.0+305+5e198a41 AppStream             47 k
containerd.io          x86_64               1.3.7-3.1.el8        docker-ce-stable     29 M
docker-ce-cli          x86_64               1:19.03.13-3.el8     docker-ce-stable     38 M
libcgroup              x86_64               0.41-19.el8          BaseOS                70 k
Enabling module streams:
container-tools        rhel8
Transaction Summary
=====
Install 5 Packages

Total download size: 92 M
Installed size: 385 M
Is this ok [y/N]: y
```

Я хотел рассказать о некоторых нюансах установки на Centos 8, а по факту оказалось, что их больше нет. Еще недавно не получалось просто взять и установить пакет с докером. Была проблема с зависимостями, связанная с тем, что 8-я версия системы не поддерживала необходимую версию **container.id**. Но сейчас этой проблемы нет, так что можно про нее забыть.

Проверим, что Docker нормально запустился. И сразу добавим его в автозагрузку.

```
# systemctl enable --now docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
# systemctl status docker
```

```
[root@centos8 ~]# systemctl enable --now docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@centos8 ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-10-02 12:03:18 MSK; 7s ago
     Docs: https://docs.docker.com
   Main PID: 42921 (dockerd)
    Tasks: 12
   Memory: 46.3M
   CGroup: /system.slice/docker.service
           └─42921 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.273423685+03:00" level=error msg="Failed to built-in GetDriver graph btrfs /var/lib/docker"
Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.297683673+03:00" level=warning msg="Your kernel does not support cgroup blkio weight"
Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.297716044+03:00" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.297885877+03:00" level=info msg="Loading containers: start."
Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.630115656+03:00" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.1"
Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.867576052+03:00" level=info msg="Loading containers: done."
Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.889416284+03:00" level=info msg="Docker daemon" commit=4484c46d9d graphdriver(s)=overlay2 version=20.10.1"
Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.889695072+03:00" level=info msg="Daemon has completed initialization"
Oct 02 12:03:18 centos8 dockerd[42921]: time="2020-10-02T12:03:18.909085744+03:00" level=info msg="API listen on /var/run/docker.sock"
Oct 02 12:03:18 centos8 systemd[1]: Started Docker Application Container Engine.
```

Все нормально, Docker успешно стартовал. Проверим его работу, запустив какой-нибудь контейнер. Например, официальный образ nginx. Сначала загрузим его.

```
# docker pull nginx
```

```
[root@centos8 ~]# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
d121f8d1c412: Pull complete
ebd81fc8c071: Pull complete
655316c160af: Pull complete
d15953c0e0f8: Pull complete
2ee525c5c3cc: Pull complete
Digest: sha256:c628b67d21744fce822d22fdcc0389f6bd763daac23a6b77147d0712ea7102d0
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

serveradmin.ru

Проверяем список образов.

```
# docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
nginx                latest      7e4d58f0e5f3     3 weeks ago     133MB
```

Теперь запускаем контейнер на основе этого образа и сразу включаем ему автозапуск после старта системы. Частенько разработчики забывают это делать и потом, после ребута сервера, контейнеры не стартуют.

```
# docker run -d -p 80:80 --restart=always --name nginx-proxy nginx
```

Смотрим список запущенных контейнеров.

```
# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
1d6555ea4726	nginx	"/docker-entrypoint..."	3 minutes ago	Up 3 minutes	
0.0.0.0:80->80/tcp	nginx-proxy				

Все в порядке, контейнер слушает 80-й порт. Посмотрим список открытых портов в системе.

```
# netstat -tulnp
```

```
[root@centos8 ~]# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:5355           0.0.0.0:*                LISTEN      1050/systemd-resolv
tcp        0      0 0.0.0.0:22            0.0.0.0:*                LISTEN      841/sshd
tcp        0      0 127.0.0.1:5432        0.0.0.0:*                LISTEN      863/postgres
tcp6       0      0 :::5355               :::*                    LISTEN      1050/systemd-resolv
tcp6       0      0 :::80                 :::*                    LISTEN      44741/docker-proxy
tcp6       0      0 :::22                 :::*                    LISTEN      841/sshd
tcp6       0      0 :::1:5432             :::*                    LISTEN      863/postgres
udp        0      0 127.0.0.53:53         0.0.0.0:*                LISTEN      1050/systemd-resolv
udp        0      0 127.0.0.1:323        0.0.0.0:*                LISTEN      760/chronyd
udp        0      0 0.0.0.0:5355         0.0.0.0:*                LISTEN      1050/systemd-resolv
udp6       0      0 :::1:323              :::*                    LISTEN      760/chronyd
udp6       0      0 :::5355               :::*                    LISTEN      1050/systemd-resolv
[root@centos8 ~]#
```

Несмотря на то, что указан ipv6 адрес, контейнер nginx запущен в том числе и на ipv4. Можете зайти браузером на страницу с ip адресом сервера и

увидите там стартовую страницу nginx. Причем, даже если у вас запущен firewalld, docker автоматически добавит разрешающее правило для 80-го порта.

На этом по установке Docker на Centos 8 все. Мы настроили и проверили работу. Собственно, даже на таком простом примере наглядно видно, в чем удобство докера, хотя тут он сработал аналогично пакетному менеджеру. Но если у вас что-то посложнее с кучей зависимостей, поставится оно так же быстро и легко.

Установка Docker в Centos 7

В Centos 7 Docker устанавливается так же штатно, через официальный репозиторий. Прежде чем его подключить, убедитесь, что у вас установлен пакет **yum-utils**.

```
# yum install yum-utils
```

После этого подключаем репозиторий докера.

```
# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
```

Теперь можно приступить к установке Docker на Centos 7.

```
# yum install docker-ce docker-ce-cli containerd.io
```

```
=====
Package                                Arch                                Version                                Repository                                Size
=====
Installing:
containerd.io                          x86_64                              1.3.7-3.1.e17                         docker-ce-stable                         29 M
docker-ce                               x86_64                              3:19.03.13-3.e17                      docker-ce-stable                         24 M
docker-ce-cli                           x86_64                              1:19.03.13-3.e17                      docker-ce-stable                         38 M
Installing for dependencies:
audit-libs-python                      x86_64                              2.8.5-4.e17                           base                                     76 k
checkpolicy                             x86_64                              2.5-8.e17                               base                                     295 k
container-selinux                       noarch                              2:2.119.2-1.911c772.e17_8             extras                                    40 k
libcgroup                               x86_64                              0.41-21.e17                            base                                     66 k
libsemanage-python                     x86_64                              2.5-14.e17                             base                                     113 k
policycoreutils-python                 x86_64                              2.5-34.e17                             base                                     457 k
python-IPy                             noarch                              0.75-6.e17                             base                                     32 k
setools-libs                           x86_64                              3.3.8-4.e17                           base                                     620 k

Transaction Summary
=====
Install 3 Packages (+8 Dependent packages)

Total download size: 93 M
Installed size: 390 M
Is this ok [y/d/N]: █
```

Запускаем докер и добавляем в автозагрузку.

```
# systemctl enable --now docker
```

Проверяем работу:

```
# systemctl status docker
```

Запускаем контейнер с nginx для теста.

```
# docker run -d -p 80:80 --restart=always --name nginx-proxy nginx
```

Убеждаемся, что все запущено и корректно работает.

```
# docker ps  
# ss -tulnp
```

```
[root@prox-centos7-kvm ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
2f31820697cc       nginx              "/docker-entrypoint..." 32 seconds ago     Up 29 seconds      0.0.0.0:80->80/tcp   nginx-proxy
[root@prox-centos7-kvm ~]# ss -tulnp
Netid  State      Recv-Q Send-Q           Local Address:Port           Peer Address:Port
udp    UNCONN    0      0                *:68                          *:*
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
udp	UNCONN	0	0	127.0.0.1:323	*:*
udp	UNCONN	0	0	:::1:323	:::*
tcp	LISTEN	0	128	:::22	*:*
tcp	LISTEN	0	100	127.0.0.1:25	*:*
tcp	LISTEN	0	128	:::22	:::*
tcp	LISTEN	0	100	:::1:25	:::*
tcp	LISTEN	0	128	:::80	:::*

```
users:(("dhclient",pid=728,fd=6))
users:(("chronyd",pid=524,fd=5))
users:(("chronyd",pid=524,fd=6))
users:(("sshd",pid=785,fd=3))
users:(("master",pid=1232,fd=13))
users:(("sshd",pid=785,fd=4))
users:(("master",pid=1232,fd=14))
users:(("docker-proxy",pid=8825,fd=4))
[root@prox-centos7-kvm ~]#
```

На этом установка Docker на Centos 7 закончена. Для верности можете в браузере проверить, что nginx запущен и работает.

Установка docker-compose на Centos

Зачастую для работы с докером требуется также docker-compose. Он позволяет быстро запускать проекты, состоящие из нескольких контейнеров. По своей сути docker-compose просто скрипт на python. Так что для его работы нужен собственно сам скрипт и некоторые компоненты python. Последнюю версию скрипта можно посмотреть в репозитории на github - <https://github.com/docker/compose/releases/>. В моем случае это 1.27.4.

Устанавливаем docker-compose на Centos.

```
# curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-Linux-x86_64" -o /usr/local/bin/docker-compose
```

Делаем файл исполняемым и на всякий случай добавляем символическую ссылку еще и в */usr/bin*.

```
# chmod +x /usr/local/bin/docker-compose  
# ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Смотрим, все ли корректно установилось.

```
# docker-compose -v  
docker-compose version 1.27.4, build 40524192
```

Для теста можете запустить что-то с помощью *docker-compose*. Например, установить WordPress. Для этого создаем файл *docker-compose.yml* следующего содержания.

```
version: '3'  
  
services:  
  mysql:  
    image: mysql:8  
    command: --default-authentication-plugin=mysql_native_password  
    restart: always  
    environment:  
      MYSQL_ROOT_PASSWORD: root  
      MYSQL_DATABASE: wordpress  
    volumes:
```

```
- "./db:/var/lib/mysql"

wordpress:
  image: wordpress:php7.4-apache
  ports:
    - "80:80"
  environment:
    WORDPRESS_DB_HOST: mysql
    WORDPRESS_DB_USER: root
    WORDPRESS_DB_PASSWORD: root
    WORDPRESS_DB_NAME: wordpress
  volumes:
    - "./wp:/var/www/html/"
```

После этого запускаем проект.

```
# docker-compose up
```

Поднимутся два контейнера - один с mysql, другой с web сервером и исходниками wordpress. Если получите ошибку соединения с базой данных:

```
MySQL Connection Error: (2002) No route to host
```

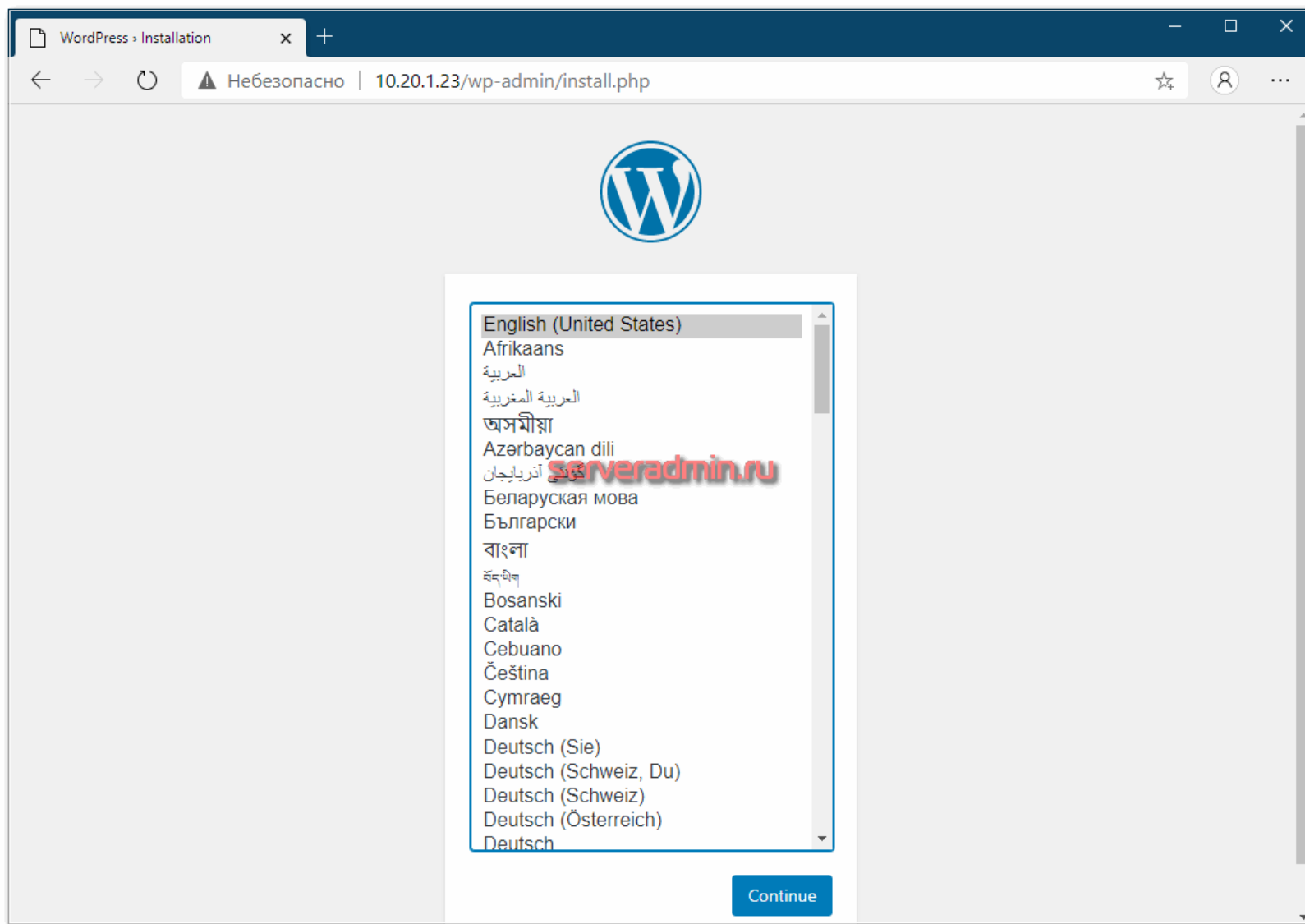
Значит у вас запущен firewalld. Он не дает нормально работать dns в докере. Вам необходимо отключить firewalld и перезапустить докер.

```
# systemctl stop firewalld
# systemctl restart docker
```

После этого запускайте docker-compose еще раз и проверяйте работу. Должно получиться примерно так.

```
mysql_1 | 2020-10-02 10:36:56+00:00 [Note] [Entrypoint]: Temporary server stopped
mysql_1 |
mysql_1 | 2020-10-02 10:36:56+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.
mysql_1 |
mysql_1 | 2020-10-02T10:36:56.625333Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.21) starting as process 1
mysql_1 | 2020-10-02T10:36:56.633084Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql_1 | 2020-10-02T10:36:56.827180Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql_1 | 2020-10-02T10:36:56.910618Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld
/mysqlx.sock
mysql_1 | 2020-10-02T10:36:56.969173Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
mysql_1 | 2020-10-02T10:36:56.969326Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported f
or this channel.
mysql_1 | 2020-10-02T10:36:56.971715Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is acces
sible to all OS users. Consider choosing a different directory.
mysql_1 | 2020-10-02T10:36:56.989064Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.21' socket: '/var/run/mysqld/my
sqld.sock' port: 3306 MySQL Community Server - GPL.
wordpress_1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally
to suppress this message
wordpress_1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally
to suppress this message
wordpress_1 | [Fri Oct 02 10:36:59.459356 2020] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.38 (Debian) PHP/7.4.11 configured -- resuming normal operations
wordpress_1 | [Fri Oct 02 10:36:59.459411 2020] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

```
# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
NAMES
f4edc21405e4       wordpress:php7.4-apache "docker-entrypoint.s..." 31 minutes ago     Up 52 seconds
0.0.0.0:80->80/tcp   root_wordpress_1
2c80b7811bb5       mysql:8            "docker-entrypoint.s..." 31 minutes ago     Up 52 seconds      3306/tcp,
33060/tcp          root_mysql_1
```



Docker-compose успешно установлен и проверен на Centos. Можно использовать по назначению.

Заключение

На этом по установке Docker и docker-compose на centos у меня все. Ссылка на официальную документацию - <https://docs.docker.com>.

Онлайн курс "DevOps практики и инструменты"

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, научиться непрерывной поставке ПО, мониторингу и логированию web приложений, рекомендую познакомиться с **онлайн-курсом «DevOps практики и инструменты»** в OTUS. Курс не для новичков, для поступления нужны базовые знания по сетям и установке Linux на виртуалку. Обучение длится 5 месяцев, после чего успешные выпускники курса смогут пройти собеседования у партнеров. Проверьте себя на вступительном тесте и смотрите программу подробнее по .

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.