

- Контейнеризация
  - Задания
    - Задание 1
    - Задание 2\*
  - Решение
    - Задание 1 (docker-compose.yml)
    - Задание 2 (docker swarm)
    - Заключение

# Контейнеризация

---

Урок 5. Docker Compose и Docker Swarm

## Задания

---

### Задание 1

1. создать docker compose файл, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД
2. запустить docker compose файл
3. по итогу на БД контейнере должно быть 2 реплики, на *adminer* должна быть 1 реплика. Всего должно получиться 3 контейнера
4. выводы зафиксировать

### Задание 2\*

1. создать кластер и мастер и слейв ноды
2. задеплоить на ноду несколько экземпляров какого-нибудь контейнера, например nginx
3. обязательно проверить и зафиксировать результаты, чтобы можно было выслать преподавателю для проверки

## Решение

## Задание 1 (docker-compose.yaml)

Создадим папку проекта *adminer* и в ней *docker-compose.yml*

```
vim docker-compose.yml
```

```
version: '3.9'

services:
  adminer:
    image: adminer:latest
    container_name: adminer
    ports:
      - "8080:8080"
    environment:
      - ADMINER_DEFAULT_SERVER=mariadb
      - ADMINER_DEFAULT_USERNAME=root
      - ADMINER_DEFAULT_PASSWORD=test123
  mariadb:
    image: mariadb:latest
    environment:
      - MYSQL_ROOT_PASSWORD=test123
    networks:
      - mynetwork
    command: ["--server-id=100", "--log-bin=ON"]
    deploy:
      mode: replicated
      replicas: 2

networks:
  mynetwork:
```

Внутри файла указываем инструкции в разделе *services* две службы: *adminer*, *mariadb*

В свою очередь каждая служба будет содержать другие инструкции:

- **image** - образ и версия службы (системы)
- **container\_name** - имя, которое будет присвоено контейнеру
- **ports** - пробрасываемые порты с контейнера на хостовую систему
- **environment** - переменные окружения контейнера
- **deploy** опции развертывания такие, как *mode* и *replicas*
- **replicas** - количество реплик (контейнеров) для данной службы

Когда файл готов, запускаем, в папке проекта *adminer*, команду

```
docker compose up
```

Загрузятся необходимые образы и произойдет разворачивание служб в соответствии с инструкциями

```
root@fen:/mnt/backup/github/docker/05.Tasks/adminer# vim docker-compose.yaml
root@fen:/mnt/backup/github/docker/05.Tasks/adminer# docker compose up
[+] Running 1/17
  ✓ adminer 7 layers [#####] 0B/0B Pulled 22.8s
    ✓ 5dea071bb978 Pull complete 15.5s
    ✓ 0d84864d5d25 Pull complete 12.2s
    ✓ 792bf43687eb Pull complete 0.5s
    ✓ 37ef5d269982 Pull complete 1.1s
    ✓ 81b9b963ca6d Pull complete 1.6s
    ✓ e57581023c92 Pull complete 2.4s
    ✓ e97ed40bb1db Pull complete 3.0s
  ✓ mariadb 8 layers [#####] 0B/0B Pulled 22.7s
    ✓ 44ba2882f8eb Already exists 0.0s
    ✓ 08b8223d0cb6 Pull complete 3.5s
    ✓ ef2696fb09d6 Pull complete 4.9s
    ✓ 6ae32c298a0d Pull complete 5.4s
    ✓ 5dc97cb97b44 Pull complete 5.9s
    ✓ a3e4bee69a58 Pull complete 16.7s
    ✓ b29c582204c9 Pull complete 12.9s
    ✓ f05405b8aaed Pull complete 20.4s
[+] Running 3/3
  ✓ Container adminer Created 3.0s
  ✓ Container adminer-mariadb-2 Created 3.0s
  ✓ Container adminer-mariadb-1 Created 3.0s
Attaching to adminer, adminer-mariadb-1, adminer-mariadb-2
adminer | touch: cannot touch '/adminer-init': Permission denied
adminer-mariadb-2 | 2023-09-09 09:57:14+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:11.1.2+maria-ubu2204 started.
adminer | [Sat Sep 9 09:57:14 2023] PHP 7.4.33 Development Server (http://[::]:8080) started
adminer-mariadb-1 | 2023-09-09 09:57:14+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:11.1.2+maria-ubu2204 started.
adminer-mariadb-2 | 2023-09-09 09:57:14+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
adminer-mariadb-2 | 2023-09-09 09:57:14+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:11.1.2+maria-ubu2204 started.
adminer-mariadb-1 | 2023-09-09 09:57:14+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
adminer-mariadb-2 | 2023-09-09 09:57:14+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:11.1.2+maria-ubu2204 started.
adminer-mariadb-1 | 2023-09-09 09:57:15+00:00 [Note] [Entrypoint]: Initializing database files
adminer-mariadb-2 | 2023-09-09 09:57:15+00:00 [Note] [Entrypoint]: Initializing database files
```

Весь процесс установки и развертывания с возможными ошибками выводится в терминал

```

adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] InnoDB: Opened 3 undo tablespaces
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] InnoDB: 128 rollback segments in 3 undo tablespaces are active.
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] InnoDB: Setting file './ibtmp1' size to 12.000MiB. Physically writing the file full; Please wait ...
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] InnoDB: File './ibtmp1' size is now 12.000MiB.
adminer-mariadb-2 | 2023-09-09 09:57:22+00:00 [Note] [Entrypoint]: MariaDB init process done. Ready for start up.
adminer-mariadb-2 |
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] InnoDB: Log sequence number 47263; transaction id 14
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] Plugin 'FEEDBACK' is disabled.
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] Plugin 'wsrep-provider' is disabled.
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] InnoDB: Buffer pool(s) load completed at 230909 9:57:22
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] Server socket created on IP: '0.0.0.0'.
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] Server socket created on IP: '::'.
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] mariadb: Event Scheduler: Loaded 0 events
adminer-mariadb-1 | 2023-09-09 9:57:22 0 [Note] mariadb: ready for connections.
adminer-mariadb-1 | Version: '11.1.2-MariaDB-1:11.1.2+maria-ubu2204-log' socket: '/run/mysqld/mysqld.sock' port: 3306 mariadb.org binary distribution
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] Starting MariaDB 11.1.2-MariaDB-1:11.1.2+maria-ubu2204-log source revision 9bc25d98209df6810f7a7d5e7dd3ae677a313ab5 as process 1
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Number of transaction pools: 1
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Using generic crc32 instructions
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] mariadb: 0 TMPFILE is not supported on /tmp (disabling future attempts)
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Using liburing
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Initializing buffer pool, total size = 128.000MiB, chunk size = 2.000MiB
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Completed initialization of buffer pool
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: End of log at LSN=47279
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Opened 3 undo tablespaces
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: 128 rollback segments in 3 undo tablespaces are active.
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Setting file './ibtmp1' size to 12.000MiB. Physically writing the file full; Please wait ...
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: File './ibtmp1' size is now 12.000MiB.
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Log sequence number 47279; transaction id 14
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] Plugin 'FEEDBACK' is disabled.
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] Plugin 'wsrep-provider' is disabled.
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] InnoDB: Buffer pool(s) load completed at 230909 9:57:22
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] Server socket created on IP: '0.0.0.0'.
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] Server socket created on IP: '::'.
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] mariadb: Event Scheduler: Loaded 0 events
adminer-mariadb-2 | 2023-09-09 9:57:22 0 [Note] mariadb: ready for connections.
adminer-mariadb-2 | Version: '11.1.2-MariaDB-1:11.1.2+maria-ubu2204-log' socket: '/run/mysqld/mysqld.sock' port: 3306 mariadb.org binary distribution

```

После успешного поднятия служб в соответствии с *docker-compose.yml*, можно зайти в *adminer* и управлять *mariadb*

← → ↺

192.168.1.8:8080/?server=mariadb&username=root&db=information\_schema

☆

☆

Язык: Русский

MySQL » mariadb » База данных: information\_schema

Adminer 4.8.1

База данных: information\_schema

Изменить базу данных

Схема базы данных

Полномочия

SQL-запрос

Импорт

Экспорт

Создать таблицу

выбрать ALL\_PLUGINS

выбрать APPLICABLE\_ROLES

выбрать CHARACTER\_SETS

выбрать CHECK\_CONSTRAINTS

выбрать CLIENT\_STATISTICS

выбрать COLLATIONS

выбрать COLLATION\_CHARACTER\_SET

выбрать COLUMNS

выбрать COLUMN\_PRIVILEGES

выбрать ENABLED\_ROLES

выбрать ENGINES

выбрать EVENTS

выбрать FILES

выбрать GEOMETRY\_COLUMNS

выбрать GLOBAL\_STATUS

выбрать GLOBAL\_VARIABLES

выбрать INDEX\_STATISTICS

выбрать INNODB\_BUFFER\_PAGE

выбрать INNODB\_BUFFER\_PAGE\_LRI

выбрать INNODB\_BUFFER\_POOL\_ST

выбрать INNODB\_CMP

выбрать INNODB\_CMPMEM

выбрать INNODB\_CMPMEM\_RESET

выбрать INNODB\_CMPMEM\_RESET

Поиск в таблицах

Поиск

<input type="checkbox"/>	Таблица	Тип таблиц?	Режим сопоставления?	Объём данных?	Объём индексов?	Свободное место?
<input type="checkbox"/>	ALL_PLUGINS		Представление			
<input type="checkbox"/>	APPLICABLE_ROLES		Представление			
<input type="checkbox"/>	CHARACTER_SETS		Представление			
<input type="checkbox"/>	CHECK_CONSTRAINTS		Представление			
<input type="checkbox"/>	CLIENT_STATISTICS		Представление			
<input type="checkbox"/>	COLLATIONS		Представление			
<input type="checkbox"/>	COLLATION_CHARACTER_SET_APPLICABILITY		Представление			
<input type="checkbox"/>	COLUMNS		Представление			
<input type="checkbox"/>	COLUMN_PRIVILEGES		Представление			
<input type="checkbox"/>	ENABLED_ROLES		Представление			
<input type="checkbox"/>	ENGINES		Представление			
<input type="checkbox"/>	EVENTS		Представление			
<input type="checkbox"/>	FILES		Представление			

Т.к. команда выполнялась без запуска в режиме демона, то вся служебная информации выводится в терминал, в частности мы можем наблюдать в режиме реального времени обращения к *adminer* через веб-сервер

```
adminer | [Sat Sep 9 10:14:04 2023] [::ffff:172.21.0.1]:47026 Accepted
adminer | [Sat Sep 9 10:14:04 2023] [::ffff:172.21.0.1]:47026 [200]: GET /
adminer | [Sat Sep 9 10:14:04 2023] [::ffff:172.21.0.1]:47026 Closing
adminer | [Sat Sep 9 10:14:43 2023] [::ffff:192.168.1.11]:51742 Accepted
adminer | [Sat Sep 9 10:14:43 2023] [::ffff:192.168.1.11]:51742 [200]: GET /
adminer | [Sat Sep 9 10:14:43 2023] [::ffff:192.168.1.11]:51742 Closing
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51748 Accepted
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51764 Accepted
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51748 [200]: GET /?file=default.css&version=4.8.1
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51748 Closing
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51764 [200]: GET /?file=functions.js&version=4.8.1
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51764 Closing
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51774 Accepted
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51774 [200]: GET /?file=favicon.ico&version=4.8.1
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51774 Closing
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51784 Accepted
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51784 [200]: GET /?file=favicon.ico&version=4.8.1
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51784 Closing
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51790 Accepted
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51790 [200]: POST /?script=version
adminer | [Sat Sep 9 10:14:44 2023] [::ffff:192.168.1.11]:51790 Closing
adminer | [Sat Sep 9 10:15:19 2023] [::ffff:192.168.1.11]:44988 Accepted
adminer | [Sat Sep 9 10:15:19 2023] [::ffff:192.168.1.11]:44988 [302]: POST /
adminer | [Sat Sep 9 10:15:19 2023] [::ffff:192.168.1.11]:44988 Closing
adminer | [Sat Sep 9 10:15:20 2023] [::ffff:192.168.1.11]:44990 Accepted
adminer | [Sat Sep 9 10:15:20 2023] [::ffff:192.168.1.11]:44990 [200]: GET /?server=mariadb&username=root
adminer | [Sat Sep 9 10:15:20 2023] [::ffff:192.168.1.11]:44990 Closing
adminer | [Sat Sep 9 10:15:20 2023] [::ffff:192.168.1.11]:45006 Accepted
adminer | [Sat Sep 9 10:15:20 2023] [::ffff:192.168.1.11]:45006 [200]: GET /?file=jush.js&version=4.8.1
adminer | [Sat Sep 9 10:15:20 2023] [::ffff:192.168.1.11]:45006 Closing
adminer | [Sat Sep 9 10:15:27 2023] [::ffff:192.168.1.11]:45012 Accepted
adminer | [Sat Sep 9 10:15:27 2023] [::ffff:192.168.1.11]:45012 [200]: GET /?server=mariadb&username=root&db=information_schema
adminer | [Sat Sep 9 10:15:27 2023] [::ffff:192.168.1.11]:45012 Closing
adminer | [Sat Sep 9 10:15:27 2023] [::ffff:192.168.1.11]:45022 Accepted
adminer | [Sat Sep 9 10:15:27 2023] [::ffff:192.168.1.11]:45022 [200]: GET /?server=mariadb&username=root&db=information_schema&script=db
adminer | [Sat Sep 9 10:15:27 2023] [::ffff:192.168.1.11]:45022 Closing
```

## Задание 2 (docker swarm)

Развернем кластерную систему на трех реальных системах

- **FEN** - ubuntu 22.04 (основной сервер)
- **XU2210** - ubuntu 22.10
- **XU2004** - ubuntu 20.04

Создадим на основном сервере **FEN** папку проекта *cluster* и поместим в нее *docker-compose.yaml* и *nginx.conf* для дальнейшего развертывания.

```
services:
  nginx-master:
    image: nginx:latest
    networks:
      - mynetwork
    deploy:
      placement:
        constraints: [node.role == manager]

  nginx-slave-1:
    image: nginx:latest
    networks:
      - mynetwork
    deploy:
      placement:
        constraints: [node.role == worker]

  nginx-slave-2:
    image: nginx:latest
    networks:
      - mynetwork
    deploy:
      placement:
        constraints: [node.role == worker]

  load-balancer:
    image: nginx:latest
    ports:
      - "80:80"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
    networks:
      - mynetwork
    depends_on:
      - nginx-master
      - nginx-slave-1
      - nginx-slave-2

networks:
  mynetwork:
```

"docker-compose.yml" 43L, 726B записано

Синтаксис, одинаковый, т.к. выполняется тем же **docker compose**, только присутствуют специфичные инструкции в директиве *deploy: placement* и *constraints*, которые указывают роль конкретного узла в кластере: *manager* или *worker*

```
events {}

http {
    upstream backend {
        server nginx-master;
        server nginx-slave-1;
        server nginx-slave-2;
    }

    server {
        listen 80;

        location / {
            proxy_pass http://backend;
        }
    }
}
```

"nginx.conf" 18L, 249B записано

Файл *nginx.conf* тоже будет иметь специфические настройки для распределения нагрузки между веб-серверами и переадресацией запросов в случае недоступности узлов.

Инициализация кластерной системы докера производится командой

```
docker swarm init
```

При наличии нескольких сетевых интерфейсов, либо адресов можно указать конкретный с ключом *--advertise-addr*. После успешной инициализации появится команда для присоединения других машин к нашему кластеру

```
FEN x XU2210 x XU2004
root@fen:/mnt/backup/github/docker/05.Tasks/claster# ls
docker-compose.yaml nginx.conf
root@fen:/mnt/backup/github/docker/05.Tasks/claster# vim docker-compose.yaml
root@fen:/mnt/backup/github/docker/05.Tasks/claster# vim nginx.conf
root@fen:/mnt/backup/github/docker/05.Tasks/claster# docker swarm init --advertise-addr 192.168.1.8
Swarm initialized: current node (ayvbur36tgs72gflt66lrmz0) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-2lvaivahg26d9lw18pox3e4w6e6arhj5c2spg2zhapxbne1nw1-4qjflqw39zf2pdkhz5ocuyqf 192.168.1.8:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

root@fen:/mnt/backup/github/docker/05.Tasks/claster# █
```

## Используем команду присоединения на XU2210

```
FEN x XU2210 x XU2004
root@xu2210:~# docker swarm join --token SWMTKN-1-2lvaivahg26d9lw18pox3e4w6e6arhj5c2spg2zhapxbne1nw1-4qjflqw39zf2pdkhz5ocuyqf 192.168.1.8:2377
This node joined a swarm as a worker.
root@xu2210:~# █
```

## Используем команду присоединения на XU2004

```
FEN x XU2210 x XU2004
root@xu2004:~# docker swarm join --token SWMTKN-1-2lvaivahg26d9lw18pox3e4w6e6arhj5c2spg2zhapxbne1nw1-4qjflqw39zf2pdkhz5ocuyqf 192.168.1.8:2377
This node joined a swarm as a worker.
root@xu2004:~# █
```

## Вывод информации об узлах выполняется командой

```
docker node ls
```

На выводе видим список узлов, кто является менеджером, кто рабочим их статусы и готовность

```
FEN x XU2210 x XU2004
root@fen:/mnt/backup/github/docker/05.Tasks/claster# docker node ls
ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION
ayvbur36tgs72gflt66lrmz0 * fen Ready Active Leader 24.0.6
jlz87mz5gu0llh9a84psduwu9 xu2004 Ready Active 20.10.21
m8jg7vss23yl5sempf8wgt9p xu2210 Ready Active 24.0.6
root@fen:/mnt/backup/github/docker/05.Tasks/claster# █
```

## На наших трех узлах разворачиваем (деплоим) веб-сервера *nginx* командой

```
docker stack deploy -c docker-compose.yaml gbnginx
```

В результате поднимается служебная сеть *gbnginx\_mynetwork* По одному контейнеру с веб-сервером на каждый узел, кроме мастера, на котором так же разворачивается балансировщик нагрузки



```

ot@fen:/mnt/backup/github/docker/05.Tasks/claster# docker stack deploy -c docker-compose.yaml gbnginx
Creating network gbnginx_mynetwork
Creating service gbnginx_nginx-master
Creating service gbnginx_nginx-slave-1
Creating service gbnginx_nginx-slave-2
Creating service gbnginx_load-balancer
ot@fen:/mnt/backup/github/docker/05.Tasks/claster# docker service ls

```

	NAME	MODE	REPLICAS	IMAGE	PORTS
si2pynoe5a	gbnginx_load-balancer	replicated	0/1	nginx:latest	*:80->80/tcp
6zc0d5hsbz	gbnginx_nginx-master	replicated	1/1	nginx:latest	
p7vq0eytey	gbnginx_nginx-slave-1	replicated	0/1	nginx:latest	
xm8n5o55zm	gbnginx_nginx-slave-2	replicated	0/1	nginx:latest	

```

ot@fen:/mnt/backup/github/docker/05.Tasks/claster# docker service ps gbnginx_nginx-slave-1

```

	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
xwd3eurzpx	gbnginx_nginx-slave-1.1	nginx:latest	xu2210	Running	Running about a minute ago		

```

ot@fen:/mnt/backup/github/docker/05.Tasks/claster# docker service ps gbnginx_nginx-slave-2

```

	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
5cernmz5sk	gbnginx_nginx-slave-2.1	nginx:latest	xu2004	Running	Running about a minute ago		

```

ot@fen:/mnt/backup/github/docker/05.Tasks/claster# docker service ps gbnginx_nginx-master

```

	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
ts3mjzexgn	gbnginx_nginx-master.1	nginx:latest	fen	Running	Running 2 minutes ago		

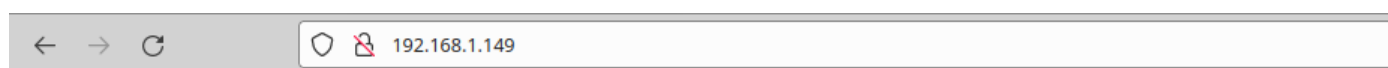
Более детальную информацию по каждой запущенной службе и узле размещения можно посмотреть командой

```
docker service ps gbnginx_nginx-master
```

В частности отобразится информация о ноде *fen* размещенной на основном сервере **FEN**:

- Id
- имя узла
- имя образа
- узел
- желаемое состояние
- текущее состояние
- ошибки
- порты

Обращение браузером к узлу XU2004



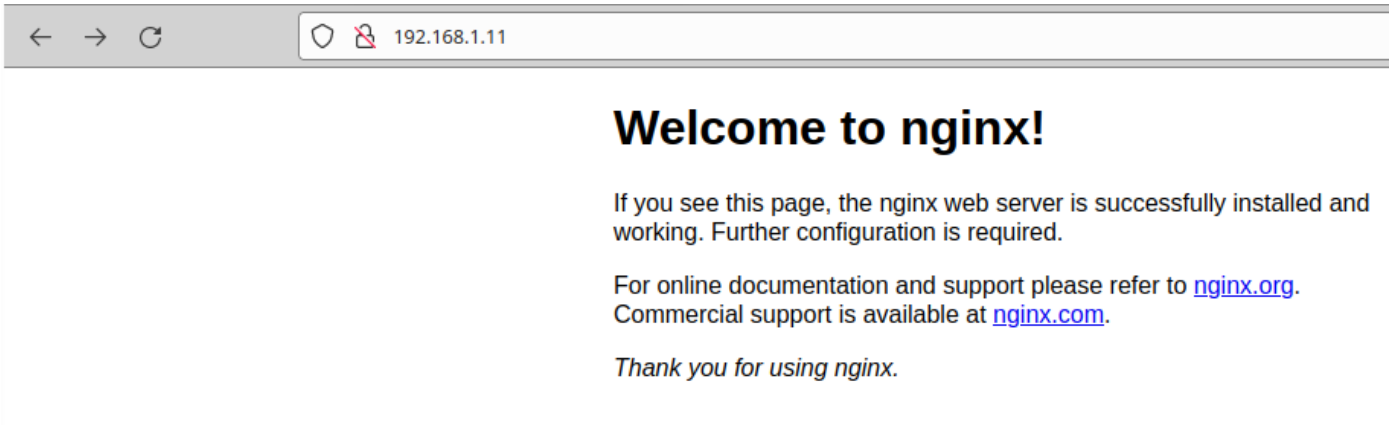
## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

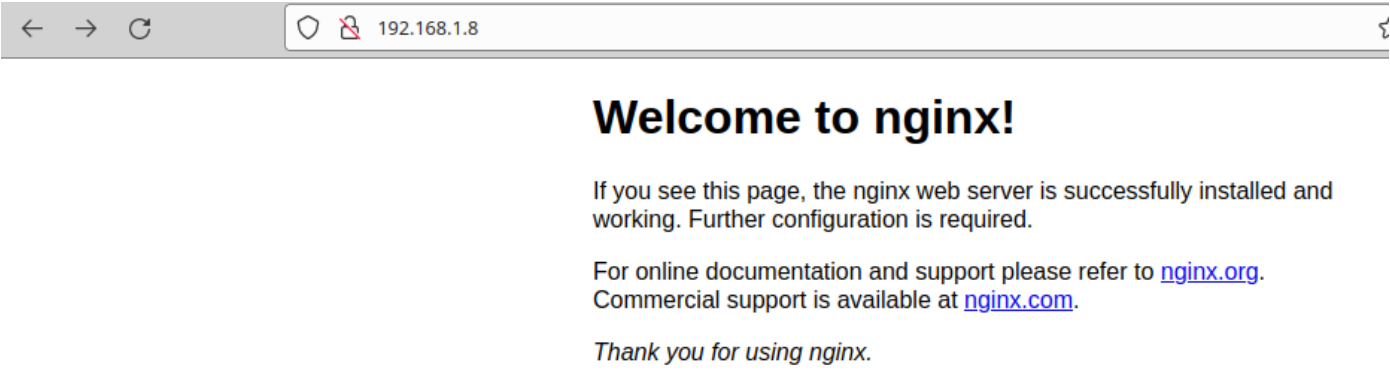
For online documentation and support please refer to [nginx.org](https://nginx.org). Commercial support is available at [nginx.com](https://nginx.com).

Thank you for using nginx.

Обращение браузером к узлу XU2210



Обращение браузером к узлу FEN



Статусы узлов можно повысить командой

```
docker node promote XU2004
```

Так статус менеджера можно присвоить сразу всем узлам

```
root@fen:~# docker node ls
ID                               HOSTNAME        STATUS        AVAILABILITY    MANAGER STATUS    ENGINE VERSION
ayvbur36tgs72gflt66lrmz0 *      fen             Ready         Active           Leader            24.0.6
j1lz87mz5gu01lh9a84psduwu9      xu2004          Ready         Active           Reachable         20.10.21
m8jg7vss23yl5semp8wgt9p         xu2210          Ready         Active           Reachable         24.0.6
root@fen:~# docker node promote xu2004
Node xu2004 promoted to a manager in the swarm.
root@fen:~# docker node promote xu2210
Node xu2210 promoted to a manager in the swarm.
root@fen:~# docker node ls
ID                               HOSTNAME        STATUS        AVAILABILITY    MANAGER STATUS    ENGINE VERSION
ayvbur36tgs72gflt66lrmz0 *      fen             Ready         Active           Leader            24.0.6
j1lz87mz5gu01lh9a84psduwu9      xu2004          Ready         Active           Reachable         20.10.21
m8jg7vss23yl5semp8wgt9p         xu2210          Ready         Active           Reachable         24.0.6
root@fen:~#
```

# Заключение

Данная связка работает хорошо, в случае отключения сразу двух любых узлов, третий берет на себя всю нагрузку. Но, в случае если умирает главный сервер, в моем случае FEN, остальные работают хорошо до их (worker) перезагрузки. Т.е. существует гипотетическая вероятность глобального блэкаута, при котором, старт не основного узла не возобновит работу веб-службы, до тех пор пока не стартует инициализирующая всех нода, даже если всем узлам присвоить статус менеджера.

Как вариант, нужно делать инициализацию кластера сразу на трех серверах и организовывать перекрестные контейнеры на каждом узле, но представляется сложным решением, в плане организации, обслуживания и дальнейшего масштабирования.

На данный момент, знаю решение вопроса блэкаута только без организации кластера, с помощью модуля nginx, а именно [http\\_health\\_check](#). С помощью него можно организовать независимые веб-сервера, но необходимо с помощью unison, либо rsync настроить синхронизацию контента, есть встроенные способы кластеризации и у mariadb (mysql). Но, это все требует больших временных трудозатрат.

Возможно есть более легкие решения на случай блэкаута, для docker swarm.