

## Установка, настройка и запуск OpenSearch

OpenSearch является продуктом реакции на ~~не~~давнее изменение условий лицензирования Elasticsearch и прекращения работы в качестве open-source платформы. AWS, Logz.io и ряд других компаний-партнеров в течение нескольких месяцев работали над созданием не просто функциональной замены Elasticsearch, а самостоятельным и перспективным проектом.

### Установка OpenSearch

Сначала вам нужно будет загрузить OpenSearch для Docker (убедитесь, что Docker Compose уже имеется на вашем устройстве).

Запустите терминал на Mac или Linux. Скачайте Docker-образы для 1) OpenSearch и 2) дашборда OpenSearch (эквивалент Kibana).

```
docker pull opensearchproject/opensearch:1.0.0-rc1
```

```
docker pull opensearchproject/opensearch-dashboards:1.0.0-rc1
```

Чтобы перейти к следующему шагу, вы должны убедиться, что либо удалили Elasticsearch, либо деактивировали его. Это связано с тем, что OpenSearch по умолчанию работает на том же порту, что и Elasticsearch – 9200. Конфликт портов может помешать дальнейшей работе. То же самое касается дашбордов OpenSearch и Kibana; оба по умолчанию используют порт 5601.

Далее запустите образ:

```
docker run -p 9200:9200 -p 9600:9600 -e "discovery.type=single-node"
opensearchproject/opensearch:1.0.0-rc1
```

Должно получиться сообщение подобного вида:

```
[2021-06-28T10:19:29,483][INFO ][o.o.s.c.ConfigurationRepository] [71235t674gby] Node
'71235t674gby' initialized
[2021-06-28T10:20:27,525][INFO ][o.o.i.i.ManagedIndexCoordinator] [71235t674gby]
Performing move cluster state metadata.
```

Чтобы продолжить, откройте вторую вкладку в терминале. Отправляйте запросы, чтобы убедиться, что OpenSearch запущен:

```
curl -XGET https://localhost:9200 -u 'admin:admin' --insecure
```

В результате должно появиться подобная информация:

```
{  "name" : "9ae0601b601c",
    "cluster_name" : "docker-cluster",
    "cluster_uuid" : "KkExYRLvQ1CIPA_hoRBYIA",
    "version" : {
      "distribution" : "opensearch",
      "number" : "1.0.0-rc1",
      "build_type" : "tar",
      "build_hash" : "26d579287f50bb33e17c8fe1f05ea208d5c64d1f",
      "build_date" : "2021-05-28T18:18:49.848386Z",
      "build_snapshot" : false,
      "lucene_version" : "8.8.2",
      "minimum_wire_compatibility_version" : "6.8.0",
      "minimum_index_compatibility_version" : "6.0.0-beta1"
    }
}
```

Чтобы развернуть Node-приложения, создайте новый файл `docker-compose.yml`. Вы можете использовать шаблон `OpenSearch docker-compose.yml template`. Сохраните файл в удобном для вас месте. В качестве примера можно создать собственный каталог/директорию для файлов `docker-compose` и отдельную поддиректорию для каждого проекта отдельно взятого `yml`-файла, в котором он будет жить. Такой шаблон доступен в документах `OpenSearch`:

```
version: '3'
services:
  opensearch-node1:
    image: opensearchproject/opensearch:latest
    container_name: opensearch-node1
    environment:
      - cluster.name=opensearch-cluster
      - node.name=opensearch-node1
      - discovery.seed_hosts=opensearch-node1,opensearch-node2
      - cluster.initial_master_nodes=opensearch-node1,opensearch-node2
      - bootstrap.memory_lock=true # along with the memlock settings below, disables
swapping
      - "OPENSEARCH_JAVA_OPTS=-Xms512m -Xmx512m" # minimum and maximum Java heap size,
recommend setting both to 50% of system RAM
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536 # maximum number of open files for the OpenSearch user, set to at
least 65536 on modern systems
        hard: 65536
    volumes:
      - opensearch-data1:/usr/share/opensearch/data
      #- ./custom-opensearch.yml:/usr/share/opensearch/config/opensearch.yml
    ports:
      - 9200:9200
      - 9600:9600 # required for Performance Analyzer
    networks:
      - opensearch-net
  opensearch-node2:
    image: opensearchproject/opensearch:latest
```

```

container_name: opensearch-node2
environment:
  - cluster.name=opensearch-cluster
  - node.name=opensearch-node2
  - discovery.seed_hosts=opensearch-node1,opensearch-node2
  - cluster.initial_master_nodes=opensearch-node1,opensearch-node2
  - bootstrap.memory_lock=true
  - "OPENSEARCH_JAVA_OPTS=-Xms512m -Xmx512m"
ulimits:
  memlock:
    soft: -1
    hard: -1
  nofile:
    soft: 65536
    hard: 65536
volumes:
  - opensearch-data2:/usr/share/opensearch/data
  #- ./custom-opensearch.yml:/usr/share/opensearch/config/opensearch.yml
networks:
  - opensearch-net
opensearch-dashboards:
  image: opensearchproject/opensearch-dashboards:latest
  container_name: opensearch-dashboards
  ports:
    - 5601:5601
  expose:
    - "5601"
  environment:
    OPENSEARCH_HOSTS: '["https://opensearch-node1:9200","https://opensearch-
node2:9200"]'
  #volumes:
    #- ./custom-opensearch_dashboards.yml:/usr/share/opensearch-
dashboards/config/opensearch_dashboards.yml
  networks:
    - opensearch-net

```

Этот образец демонстрирует нам возможность создания самого маленького “кластера” – всего два узла (Nodes) (конечно, технически один узел уже является кластером, но опустим лишние споры). Он также имеет один контейнер для запуска дашбордов OpenSearch (опять же, на порту 5601).

В завершение, необходимо запустить docker-compose для старта OpenSearch:

```
docker-compose up
```

## Настройка OpenSearch

Для настройки OpenSearch требуется отдельный файл yaml/yml: `opensearch.yml`. Вы можете либо 1) создать этот файл с помощью `-v command`, либо 2) в файле `docker-compose.yml`, упомянутом выше.

### Вариант 1:

```
docker run \
-p 9200:9200 -p 9600:9600 \
-e "discovery.type=single-node" \
-v /<full-path-to>/custom-opensearch.yml:/usr/share/opensearch/config/opensearch.yml
\ opensearchproject/opensearch:1.0.0-rc1
```

Вариант 2 (в файле `docker-compose.yml`; вам нужно будет настроить это для каждого узла): Option 2 (within the `docker-compose.yml` file; you will have to configure this for each node):

```
version: '3'
services:
  opensearch-node1:
    image: opensearchproject/opensearch:latest
    container_name: opensearch-node1
    environment:
      - cluster.name=opensearch-cluster
      - node.name=opensearch-node1
      - discovery.seed_hosts=opensearch-node1,opensearch-node2
      - cluster.initial_master_nodes=opensearch-node1,opensearch-node2
      - bootstrap.memory_lock=true # along with the memlock settings below, disables
swapping
      - "OPENSEARCH_JAVA_OPTS=-Xms512m -Xmx512m" # minimum and maximum Java heap
size, recommend setting both to 50% of system RAM
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536 # maximum number of open files for the OpenSearch user, set to
at least 65536 on modern systems
        hard: 65536
    volumes:
      - opensearch-data1:/usr/share/opensearch/data
      #- ./custom-opensearch.yml:/usr/share/opensearch/config/opensearch.yml
```

Чтобы настроить дашборды OpenSearch таким же образом, следуйте следующим инструкциям:

```
opensearch-dashboards:
  image: opensearchproject/opensearch-dashboards:latest
  container_name: opensearch-dashboards
  ports:
    - 5601:5601
  expose:
    - "5601"
  environment:
    OPENSEARCH_HOSTS: '["https://opensearch-node1:9200","https://opensearch-
node2:9200"]' #volumes:
    #- ./custom-opensearch_dashboards.yml:/usr/share/opensearch-
dashboards/config/opensearch_dashboards.yml
  networks:
    - opensearch-net
```

## Плагины

OpenSearch имеет встроенные плагины, которые являются продолжением открытого дистрибутива Open Distro (который изначально создавал уникальные плагины для ассимиляции с сервисом Elasticsearch, а теперь адаптирован для работы с OpenSearch).

Стоит отметить, что существует более минималистичная версия OpenSearch без встроенных открытых плагинов дистрибутива Open Distro, которые вы можете установить самостоятельно.

Чтобы настроить образ с помощью другого плагина, следуйте данному синтаксису:

```
FROM opensearchproject/opensearch:1.0.0-rc1
```

```
RUN /usr/share/opensearch/bin/opensearch-plugin install --batch <opensearch-plugin-
name>
```

А затем:

```
docker build --tag=opensearch-custom-plugin
```

```
docker run -p 9200:9200 -p 9600:9600 -v /usr/share/opensearch/data opensearch-custom-
plugin
```