Инструкция по установке Q.KeepeR

Exported on Jul 08, 2019

Инструкция составлена на примере развертывания Q.KeepeR на ОС Linux Ubuntu 18.04
Пользователь должен иметь возможность выполнять команды от имени суперпользователя (sudo + команда). Открыть терминал можно, нажав сочетание клавиш ctrl + alt + t
Все команды далее выполняются в этом терминале.

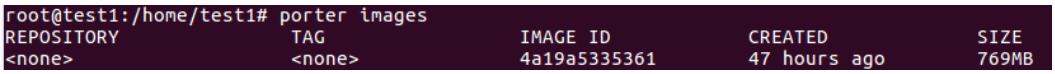
1. Установить пакет containerd.io из комплекта поставки ПО.
   sudo dpkg -i containerd.io_1.2.6_amd64.deb
   Проверить статус сервиса containerd
   systemctl status containerd
   Ожидаемое состояние: сервис запущен
   Установить пакеты Porter - клиентскую и серверную часть
   sudo dpkg -i porter-ce-cli_19.03.0~1.4.beta4-0~ubuntu-xenial_amd64.deb
   sudo dpkg -i porter-ce_19.03.0~1.4.beta4-0~ubuntu-xenial_amd64.deb
   Проверить статус сервиса porter
   systemctl status porter
   Ожидаемое состояние: сервис запущен
   Примечание:
   Если необходимо удалить установленные пакеты используйте команды:
   sudo dpkg -r porter-ce
   sudo dpkg -r porter-ce-cli

2. Выполнить команду в терминале:
   sudo sysctl -w vm.max_map_count=262144
   Примечание: данную команду необходимо выполнять всегда при перезагрузке машины, на которой разворачивается приложение Q.KeepeR, и при выходе пользователя из системы. Либо прописать настройку
   vm.max_map_count=262144
   в файле /etc/sysctl.conf

3. Скопировать из поставляемого комплекта ПО на машину, где предполагается развернуть приложение Q.KeepeR, следующие файлы:
   qscriber-oss-7.0.1-19051701.tar.gz
   qkeeper-oss-7.0.1-19051701.tar.gz
   qsaver-2.1.1-19062001.tar.gz
   qkeeper-logs.sh
   qkeeper-net-create.sh
   qkeeper-run.sh
   qkeeper-stop.sh
   qscriber-logs.sh
   qscriber-run.sh
   qscriber-stop.sh
   qsaver-run.sh

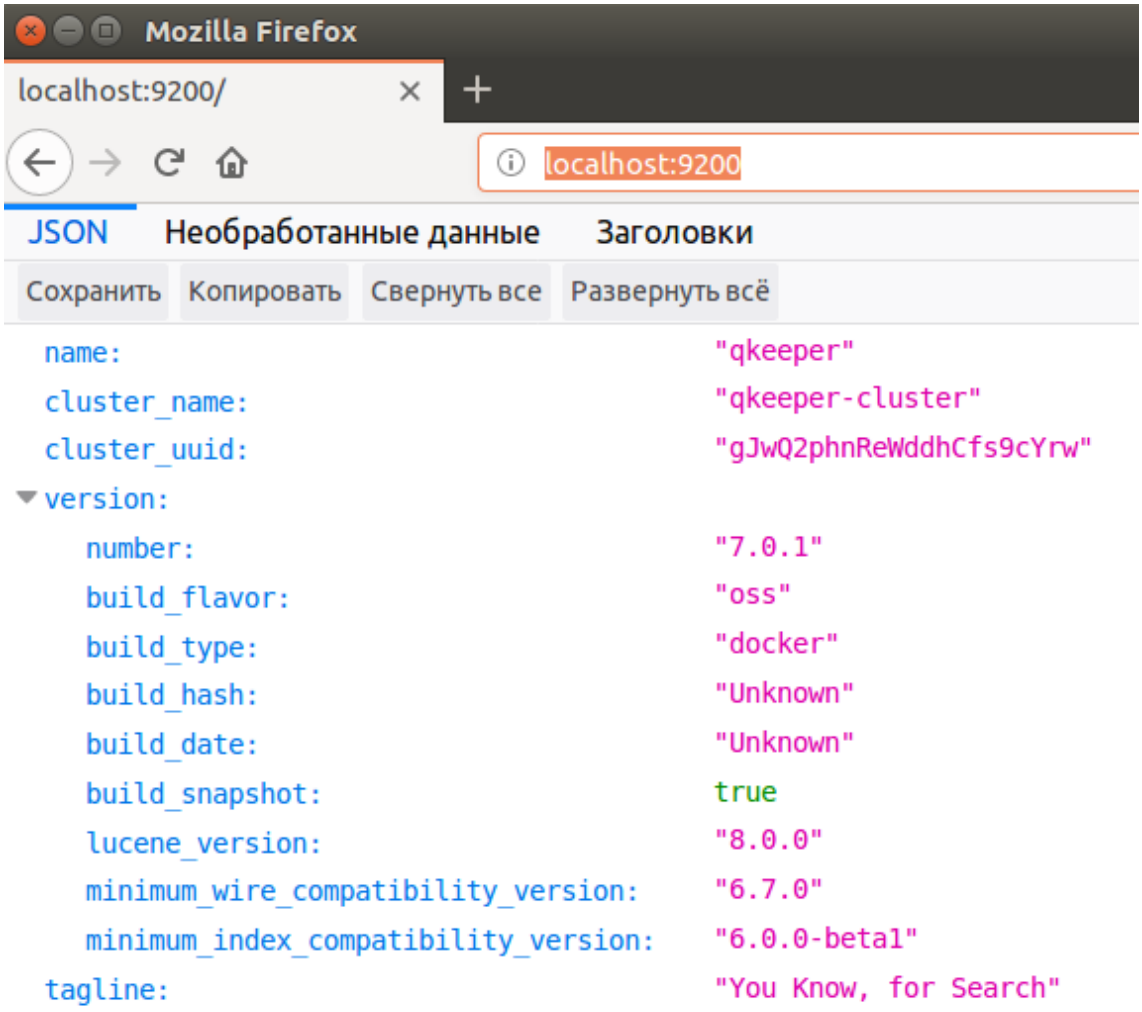
4. Перейти в директорию, куда скопированы файлы и выполнить в терминале следующую команду:
   sudo porter load -i qkeeper-oss-7.0.1-19051701.tar.gz
   Так загружается образ программы Q.KeepeR в репозиторий porter.

5. Теперь в этой же директории необходимо выполнить:
   sudo porter images
   В терминале должно появиться следующее сообщение:



6. Теперь, для удобства, необходимо переименовать образ в репозитории, используя значение 4a19a5335361 под надписью IMAGE ID из рисунка консоли выше (примечание: это значение всегда уникально):
   porter tag 4a19a5335361 ru.mdp/qkeeper-oss:7.0.1-19051701

7. Для корректной работы желательно создать отдельную сеть для контейнеров, выполнив sudo sh qkeeper-net-create.sh
8. Чтобы QKeepeR заработал, достаточно запустить скрипт следующей командой: sudo sh qkeeper-run.sh
9. Через 5 - 10 секунд, можно открыть в браузере (например, firefox) следующий адрес: http://localhost:9200
   Должна открыться страница со следующей информацией:



Для остановки приложения, нужно выполнить команду:
sudo qkeeper-stop.sh
ВНИМАНИЕ: в данном случае все данные, введенные ниже (все объекты типа сотрудник), удалятся. Повторный запуск программы Q.KeepeR произойдет в исходном состоянии.
Для того чтобы данные сохранялись, необходимо настроить монтирование папок на диске машины, где запущен Q.KeepeR.

Теперь можно пользоваться программой Q.KeepeR

Например, чтобы добавить объект сотрудник, достаточно вызвать HTTP метод PUT и указать в формате JSON необходимые данные:

Для вызова метода можно использовать утилиту curl в Linux:
Примечание: установить программу curl можно, выполнив команду в терминале:
sudo apt install curl (при наличии интернет соединения).
Либо установить командой:

sudo dpkg -i curl_7.58.0-2ubuntu3.7_amd64.deb
выполнив ее в директории с поставляемым файлом curl_7.58.0-2ubuntu3.7_amd64.deb.

```
curl -XPUT 'localhost:9200/organization/employee/1' \
-H 'Content-Type: application/json' \
-d '{
"first_name" : "Алексей",
"last_name" : "Иванов",
"age" : 25,
"about" : "Люблю путешествовать на автомобилях",
"interests": [ "музыка", "автомобили", "игра на гитаре" ]
}'
```

Данный запрос можно полностью скопировать и вставить в Linux терминале и нажать Enter.

Здесь важен путь по которому добавляется объект сотрудник:
organization — это имя индекса (аналог обычной базы данных)
employee — это имя типа объектами (аналог таблицы в обычной базе данных)
1 — порядковый номер конкретного сотрудника
Параметры в JSON формате можно сравнить с полями в обычной таблице с конкретными значениями.
Для понимания основ поиска и чтобы понять, как работает полнотекстовый поиск, необходимо добавить дополнительные данные, выполнив следующие команды:

```
curl -XPUT 'localhost:9200/organization/employee/2' \
-H 'Content-Type: application/json' \
-d '{
"first_name" : "Иван",
"last_name" : "Петров",
"age" : 30,
"about" : "Люблю езду на автомобилях и лыжи",
"interests": [ "покупки", "автомобили", "лыжи" ]
}'
```

нажать Enter

```
curl -XPUT 'localhost:9200/organization/employee/3' \
-H 'Content-Type: application/json' \
-d '{
"first_name" : "Владимир",
"last_name" : "Попов",
"age" : 27,
"about" : "Нравится рыбалка и охота,",
"interests": [ "автомобили", "рыбалка", "охота"]
}'
```

нажать Enter

```
curl -XPUT 'localhost:9200/organization/employee/4' \
-H 'Content-Type: application/json' \
-d '{
"first_name" : "Михаил",
"last_name" : "Иванов",
"age" : 41,
"about" : "Прогулки, чтение журналов об автомобилях",
"interests": [ "коньки", "компьютерные игры", "журналы"]
}'
```

нажать Enter
Таким образом будет добавлено всего 4 сотрудника.

Чтобы получить информацию о сотруднике с порядковым номером 1, достаточно выполнить такой HTTP запрос (примечание: параметр pretty используется для удобного представления данных для чтения):

curl -X GET "localhost:9200/organization/employee/1?pretty"

ответ будет следующим:

{
"_index" : "organization",
"_type" : "employee",
"_id" : "1",
"_version" : 7,
"_seq_no" : 9,
"_primary_term" : 2,
"found" : true,
"_source" : {
"first_name" : "Алексей",
"last_name" : "Иванов",
"age" : 25,
"about" : "Люблю путешествовать на автомобилях",
"interests" : [
"музыка",
"автомобили",
"игра на гитаре"
]
}
}

Чтобы произвести поиск всех сотрудников, достаточно выполнить такой запрос:
curl -X GET "localhost:9200/organization/employee/_search?pretty"

Для поиска всех сотрудников с фамилией Иванов, достаточно выполнить:

```
curl -X GET "localhost:9200/organization/employee/_search?pretty" \
-H 'Content-Type: application/json' \
-d '{
"query" : {
"match" : {
"last_name" : "Иванов"
}
}
}'
```

В данном случае Q.KeepeR найдет всех сотрудников, у которых в поле «last_name» указана фамилия Иванов.

Запрос для полнотекстового поиска будет выглядеть так:

```
curl -X GET "localhost:9200/organization/employee/_search?pretty" \
-H 'Content-Type: application/json' \
-d '{
"query" : {
"match" : {
"about" : "на автомобилях"
}
}
}'
```

Q.KeepeR отобразит всех сотрудников у которых есть точное совпадение фразы «на автомобилях», а также всех, у кого есть в поле «about» слова «автомобилях» и «на» по релевантности.
Ответ будет таким:

{
"took" : 10,
"timed_out" : false,

```
"_shards" : {
"total" : 1,
"successful" : 1,
"skipped" : 0,
"failed" : 0
},
"hits" : {
"total" : {
"value" : 3,
"relation" : "eq"
},
"max_score" : 0.9128574,
"hits" : [
{
"_index" : "organization",
"_type" : "employee",
"_id" : "1",
"_score" : 0.9128574,
"_source" : {
"first_name" : "Алексей",
"last_name" : "Иванов",
"age" : 25,
"about" : "Люблю путешествовать на автомобилях",
"interests" : [
"музыка",
"автомобили",
"игра на гитаре"
]
}
},
{
"_index" : "organization",
"_type" : "employee",
"_id" : "2",
"_score" : 0.7777306,
"_source" : {
"first_name" : "Иван",
"last_name" : "Петров",
"age" : 30,
"about" : "Люблю езду на автомобилях и лыжи",
"interests" : [
"покупки",
"автомобили",
"лыжи"
]
}
},
{
"_index" : "organization",
"_type" : "employee",
"_id" : "4",
"_score" : 0.67745006,
"_source" : {
"first_name" : "Михаил",
"last_name" : "Иванов",
"age" : 41,
"about" : "Нравится катание на коньках, чтение журналов об автомобилях",
```

```
"interests" : [
"коньки",
"компьютерные игры",
"журналы"
]
}
}
]
}
}
```

Для поиска точной фразы, вместо слова match используйте match_phrase.

В программном продукте Q.KeepeR используется понятие узлов или нод. Каждая нода — это экземпляр программы Q.KeepeR, запущенный на отдельной машине.
Для масштабирования производительности системы возможно использовать кластерную конфигурация из нескольких машин.
В папке /bin в конфигурационном файле каждой ноды можно указать следующие настройки:

Для кластерной конфигурации нужны следующие параметры в конфигах каждой ноды:
1. Для мастер нод
cluster.name: TestCluster
node.name: node1
discovery.zen.ping.unicast.hosts: ["ip_master_1", "ip_master_2"]
node.master: true
2.
Для нод слейвов
cluster.name: TestCluster
node.name: nodeUniqueName
discovery.zen.ping.unicast.hosts: ["ip_master_1", "ip_master_2"]
node.master: false

Внимание: Между значением параметра и двоеточием обязательно должен быть пробел!