

## **Инструкция по установке Q.TraceR**

Exported on Jul 08, 2019

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](https://github.com/porterhq/porter) 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.TraceR, 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.TraceR, следующие файлы:
4.
 

```
qscriber-oss-7.0.1-19051701.tar.gz
```

```
qkeeper-oss-7.0.1-19051701.tar.gz
```

```
qsaver-2.1.1-19062001.tar.gz
```

```
qtracer-2.12.12-19062801.tar.gz
```

```
qkeeper-logs.sh
```

```
qkeeper-net-create.sh
```

```
qkeeper-run.sh
```

```
qkeeper-stop.sh
```

```
qscriber-logs.sh
```

```
qscriber-run.sh
```

```
qscriber-stop.sh
```

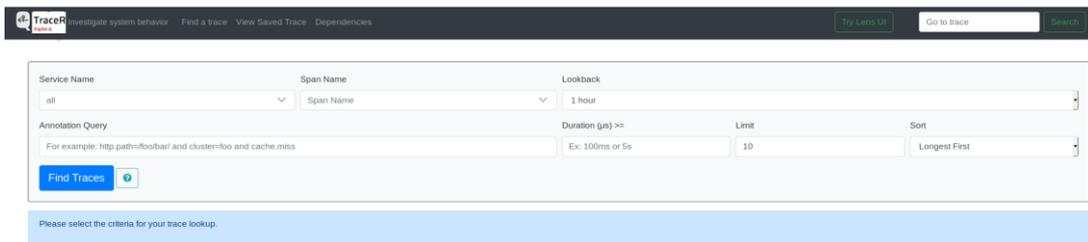
```
qsaver-run.sh
```

```
qtracer-run.sh
```
5. Перейти в директорию, куда скопированы файлы и выполнить в терминале следующую команду:  

```
sudo porter load -i qtracer-2.12.12-19062801.tar.gz
```

 Так загружается образ программы Q.TraceR в репозиторий porter.
6. Запустить Q.TraceR можно выполнив скрипт командой  

```
sudo sh qtracer-run.sh
```
7. Через 5-10 секунд можно зайти в браузере (например, firefox) по адресу <http://localhost:9411>
8. Откроется страница со следующей информацией:



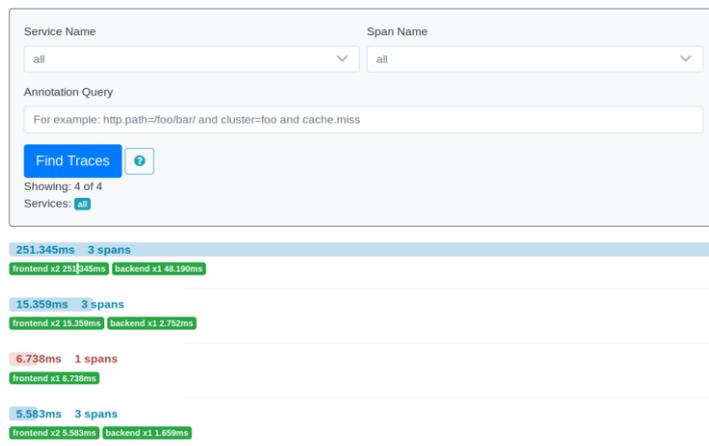
9. Для демонстрации отображения трассировки необходимо развернуть небольшое тестовое приложение. Для этого понадобится программа maven. Ее можно установить, выполнив в терминале команду
 

```
sudo apt install maven
```

 либо установить командой
 

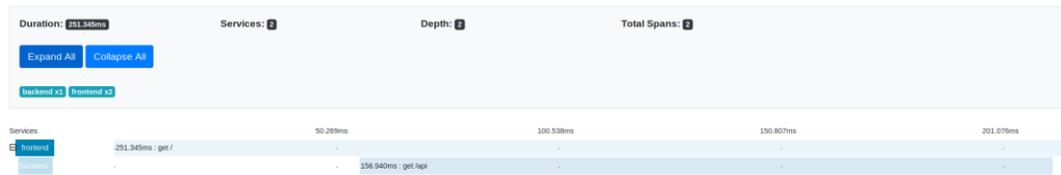
```
sudo dpkg -i maven_3.6.0-1~18.04.1_all.deb
```
10. Затем необходимо перейти в каталог, где находится папка brave-webmvc-example запустить скрипт следующей командой:
 

```
sh startQtracerExample.sh
```
11. Через некоторое время (30 - 50 секунд) будет запущено 2 сервера jetty (frontend и backend) с открытыми портами 8081 и 9000 соответственно. Суть этого тестового приложения - продемонстрировать трассировку всех вызовов от между frontend и backend. В исходном коде этого приложения внесены элементы трассировки, которые отправляют результаты при каждом запросе из frontend можно исследовать в Q.TraceR. Для этого необходимо:
  12. Открыть frontend в браузере по адресу <http://localhost:8081>  
Отобразится текущее время и дата в формате: Wed Jul 03 14:33:42 MSK 2019  
Можно дополнительно несколько раз обновить страницу в браузере (клавиша F5)  
Фактически из этой frontend части приложения будет послан запрос в backend, который в ответе возвращает текущее время и дату. При этом на каждом этапе происходит трассировка вызовов. Чтобы увидеть подробную информацию о вызовах необходимо:
  13. Открыть в браузере меню Q.TraceR (по адресу <http://localhost:9411>) и нажать на синюю кнопку с надписью Find Traces.  
Должна появиться следующая информация:



14. Видно, что появилась новая информация внизу (заголовки с указанием миллисекунд (ms) и вызовов (spans) и элементы, подсвеченные зеленым)
15. Для исследования вызовов необходимо кликнуть мышью на заголовок с миллисекундами и вызовами (на рисунке выше выделено голубой линией).

Отобразится следующее окно со сводной информацией по вызовам из frontend'a и ответу от backend'a



16. Для исследования вызовов и получения id трассировки нужно нажать на кнопку с надписью frontend (внизу слева) и нажать кнопку Show IDs. Откроется следующее окно:

frontend.get /: 251.345ms ×

Services: frontend

Date Time	Relative Time	Annotation	Address
03.07.2019, 14:33:38		Server Start	172.18.0.1 (frontend)
03.07.2019, 14:33:38	251.345ms	Server Finish	172.18.0.1 (frontend)

Key	Value
http.method	GET
http.path	/
mvc.controller.class	Frontend
mvc.controller.method	callBackend
Client Address	10.5.206.7:58174

Show IDs

traceId	7396549a09809a1d
spanId	7396549a09809a1d
parentId	

Важно отметить, что запросы выстраиваются в иерархическом виде, то есть ответ backend стоит под frontend. И при корректном уровне встраивания трассировки в код приложения можно выявить все узкие места, где происходят задержки вызовов или ответов и отследить последовательность вызовов, даже если приложение развернуто в кластере.