

Над текстом нажмите кнопку **"Создать стенд"**. До 10 минут создается стенд. Каждый стенд создан под определенный модуль курса, поэтому при выполнении нижеследующих инструкций, нужно запустить отсюда стенд.

Где получить логин/пароль? В настройках профиля или над текстом по кнопке **"Доступы"**.

Практический стенд этого урока автоматически запускает кластер Apache Kafka, состоящий из 3 брокеров, установленных на 3 виртуальных машинах.

В этой практике мы воспользуемся встроенными утилитами бенчмаркинга Кафки для замера производительности кластера. Тестовый стенд этого урока автоматически поднимает Prometheus и Grafana с дашбордом для мониторинга доступным по адресу **<http://grafana.<ваш номер студента>.edu.slurm.io>** — вы можете наблюдать за нагрузкой кластера на нем!

Обратите внимание: так как на стенде нет TLS-сертификата, для открытия URL вам потребуется открыть его в режиме Инкогнито. Для входа используйте те же доступы, как и для входа на sbox - ваш номер студента и пароль от него.

Producer Benchmark

1. Заходим на наш стенд, подключаться можно к любой ноде в кластере.

Не забудьте заменить плейсхолдер `<ваш номер студента>` на ваш личный номер студента:

```
ssh <ваш номер студента>@sbox.slurm.io
ssh node-{1-3}.<ваш номер студента>.slurm.io
export USER=<ваш номер студента>
```

2. Переходим в папку со скриптами Кафки

```
cd /opt/kafka_2.13-2.7.0/bin/
```

3. Скрипт бенчмарка продюсера называется **kafka-producer-perf-test.sh**, давайте ознакомимся с набором его опций

```
./kafka-producer-perf-test.sh
```

```
usage: producer-performance [-h] --topic TOPIC --num-records NUM-RECORDS
[--payload-delimiter PAYLOAD-DELIMITER] --throughput THROUGHPUT
[--producer-props PROP-NAME=PROP-VALUE
[PROP-NAME=PROP-VALUE ...]] [--producer.config CONFIG-FILE] [--print-metrics]
[--transactional-id TRANSACTIONAL-ID]
[--transaction-duration-ms TRANSACTION-DURATION]
(--record-size RECORD-SIZE | --payload-file PAYLOAD-FILE)
...
```

Нас в первую очередь интересуют следующие конфигури:

- **producer-props** позволяет нам передавать любые опции экземпляру продюсера (см. [полный набор](#) в официальной документации)
- **topic** указывает название топика в который бенчмарк будет писать данные
- **num-records** устанавливает лимит на общее число отправленных сообщений
- **throughput** устанавливает лимит на число отправляемых сообщений в секунду
- **record-size** конфигурирует размер одного сообщения в байтах
- **print-metrics** указывает бенчмарку вывести на экран все финальные метрики продюсера

4. В первую очередь нам нужно создать топик для теста. Создадим топик с одной партицией и RF 3:

```
./kafka-topics.sh --bootstrap-server node-1.$USER:9092 --topic producer-benchmark
--replication-factor 3 --partitions 1 --create
```

5. Запустим бенчмарк продюсера и укажем ему записать 50000 сообщений, с пропускной способностью 1000 сообщений в секунду размером 100 байт:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=node-1.$USER:9092 --topic producer-benchmark --throughput 1000
--num-records 50000 --record-size 100 --print-metrics
4963 records sent, 991.2 records/sec (0.09 MB/sec), 75.9 ms avg latency, 1178.0
ms max latency.
5027 records sent, 1004.2 records/sec (0.10 MB/sec), 34.0 ms avg latency, 72.0 ms
max latency.
5013 records sent, 1002.4 records/sec (0.10 MB/sec), 29.4 ms avg latency, 85.0 ms
max latency.
5014 records sent, 1002.4 records/sec (0.10 MB/sec), 22.0 ms avg latency, 95.0 ms
max latency.
5001 records sent, 999.0 records/sec (0.10 MB/sec), 18.6 ms avg latency, 38.0 ms
max latency.
5010 records sent, 1001.8 records/sec (0.10 MB/sec), 24.3 ms avg latency, 58.0 ms
max latency.
5018 records sent, 1002.4 records/sec (0.10 MB/sec), 18.8 ms avg latency, 59.0 ms
max latency.
5004 records sent, 1000.8 records/sec (0.10 MB/sec), 17.8 ms avg latency, 58.0 ms
max latency.
5002 records sent, 1000.0 records/sec (0.10 MB/sec), 14.1 ms avg latency, 40.0 ms
max latency.
50000 records sent, 999.540212 records/sec (0.10 MB/sec), 26.57 ms avg latency,
1178.00 ms max latency, 21 ms 50th, 52 ms 95th, 191 ms 99th, 205 ms 99.9th.
```

```
Metric Name
Value
app-info:commit-id:{client-id=producer-1}
: 62abe01bee039651
app-info:start-time-ms:{client-id=producer-1}
: 1615638113641
...
```

Во время своей работы бенчмарк выводит на экран метрики производительности продюсера — число посланных сообщений, пропускную способность, среднюю и максимальную задержку. В конце своей работы он так же выводит полный список JMX метрик инстанса продюсера благодаря переданной опции `--print-metrics`.

Из приведенного выше результата мы можем сделать вывод, что мы можем записать в 1 партицию 100 KB данных в секунду со средней задержкой в ± 27 мс.

6. Давайте посмотрим, какую максимальную пропускную способность мы можем ожидать от 1 партиции в нашем кластере при размере одного сообщения в 100

байт. Для этого передадим значение -1 в параметр `--throughput`, тем самым убрав ограничение на количество сообщений в секунду, а так же выставим `--num-records` в 500000:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=node-1.$USER:9092 --topic producer-benchmark --throughput -1
--num-records 500000 --record-size 100
77304 records sent, 15454.6 records/sec (1.47 MB/sec), 50.1 ms avg latency, 983.0
ms max latency.
116227 records sent, 23240.8 records/sec (2.22 MB/sec), 37.7 ms avg latency, 88.0
ms max latency.
121179 records sent, 24231.0 records/sec (2.31 MB/sec), 118.2 ms avg latency,
285.0 ms max latency.
110030 records sent, 22001.6 records/sec (2.10 MB/sec), 190.0 ms avg latency,
367.0 ms max latency.
500000 records sent, 20838.542969 records/sec (1.99 MB/sec), 99.21 ms avg
latency, 983.00 ms max latency, 55 ms 50th, 280 ms 95th, 348 ms 99th, 361 ms
99.9th
```

По результатам можно установить, что продюсер с дефолтными настройками может записать ± 2 МБ/сек данных со средней задержкой в 100 мс.

7. По умолчанию продюсер Кафки использует `acks=1`, т.е. ожидает ответа успешной записи от лидера партии. Давайте выставим это значение в 0, таким образом сказав продюсеру не ждать никаких ответов:

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=node-1.$USER:9092 acks=0 --topic producer-benchmark
--throughput -1 --num-records 500000 --record-size 100
81856 records sent, 16354.8 records/sec (1.56 MB/sec), 102.7 ms avg latency,
928.0 ms max latency.
125730 records sent, 25146.0 records/sec (2.40 MB/sec), 9.4 ms avg latency, 48.0
ms max latency.
151571 records sent, 30314.2 records/sec (2.89 MB/sec), 15.3 ms avg latency, 75.0
ms max latency.
123641 records sent, 24723.3 records/sec (2.36 MB/sec), 27.4 ms avg latency,
204.0 ms max latency.
500000 records sent, 24227.153794 records/sec (2.31 MB/sec), 32.98 ms avg
latency, 928.00 ms max latency, 10 ms 50th, 150 ms 95th, 391 ms 99th, 440 ms
99.9th.
```

Как видно, мы получаем более высокую производительность продюсера: 2.3 МБ/сек со средней задержкой в 32 мс.

8. Попробуйте изменить другие настройки продюсера влияющие на размер запросов, например `batch.size` или `compression.type` — какие результаты вы получили?