

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"**.

В этой практике мы развернем [Prometheus](#) и [Grafana](#) для мониторинга нашего тестового кластера, состоящего из 3 нод.

## Установка JMX Exporter

Для начала, нам нужно установить [JMX Exporter](#) на каждой ноде **Кафки**, который будет конвертировать JMX метрики Kafka и ZooKeeper в формат Prometheus.

1. Заходим на первую ноду, заменив плейсхолдер `<ваш номер студента>` на ваш личный номер студента:

```
ssh <ваш номер студента>@sbox.slurm.io  
ssh node-1.<ваш номер студента>  
Сразу становимся root:
```

```
sudo -i  
Склонируем репо с материалами для практики:
```

```
git clone git@gitlab.slurm.io:edu/kafka.git
```

2. Переходим в папку с Кафкой и создаем папку `metrics`

```
cd /opt/kafka_2.13-2.7.0 && mkdir metrics && cd metrics
```

3. Скачиваем JMX Exporter

```
wget  
https://repo1.maven.org/maven2/io/prometheus/jmx/jmx_prometheus_javaagent/0.15.0/  
jmx_prometheus_javaagent-0.15.0.jar
```

4. Скопируем конфигурационные файлы `jmx-exporter-kafka.yml` и

```
jmx-exporter-zookeeper.yml
```

```
cp ~/kafka/kafka-metrics/jmx-exporter-* /opt/kafka_2.13-2.7.0/metrics
```

5. Скажем Kafka и ZooKeeper запускаться вместе с нашим JMX Exporter агентом на порту 7071/7072 соответственно:

```
vim /etc/systemd/system/kafka.service
```

```
<<добавляем строку под уже имеющимся Environment>>
```

```
Environment=KAFKA_OPTS=-javaagent:/opt/kafka_2.13-2.7.0/metrics/jmx_prometheus_javaagent-0.15.0.jar=7071:/opt/kafka_2.13-2.7.0/metrics/jmx-exporter-kafka.yml
```

```
vim /etc/systemd/system/zookeeper.service
```

```
<<добавляем строку под уже имеющимся Environment>>
```

```
Environment=SERVER_JVMFLAGS=-javaagent:/opt/kafka_2.13-2.7.0/metrics/jmx_prometheus_javaagent-0.15.0.jar=7072:/opt/kafka_2.13-2.7.0/metrics/jmx-exporter-zookeeper.yml
```

6. Перезагружаем systemd, Kafka и ZooKeeper:

```
systemctl daemon-reload
```

```
systemctl restart kafka.service
```

```
systemctl status -l kafka.service
```

```
systemctl restart zookeeper.service
```

```
systemctl status -l zookeeper.service
```

7. Попробуем вызвать нашего агента внутри Kafka (port 7071), мы должны увидеть набор экспортированных метрик и их текущих значений:

```
curl localhost:7071
```

8. Аналогично, попробуем вызвать агента внутри ZooKeeper (port 7072):

```
curl localhost:7072
```

9. Повторяем шаги 1-9 для остальных нод: node-2.<ваш номер студента> и

```
node-3.<ваш номер студента>
```