Папка с Kafka находится на виртуальной машине, ОС Ubuntu, в директории home. Всё разворачивается локально, кроме kafka-exporter

1. Скачиваем Kafka отсюда: <https://kafka.apache.org/downloads> (у меня версия kafka\_2.13-3.2.0)
2. Скачиваем JMX-exporter отсюда: <https://github.com/prometheus/jmx_exporter> . Скачиваем jar файл (у меня jmx\_prometheus\_javaagent-0.17.0.jar). Кладём в kafka/kafka\_2.13-3.2.0/libs/ .
3. В папке kafka\_2.13-3.2.0 🡪 config создаём jmx\_prometheus\_javaagent-0.17.0.yaml . В него записываем следующее (по ссылке из пункта 2, example\_configs 🡪 kafka-2\_0\_0.yml):

lowercaseOutputName: true

rules:

# Special cases and very specific rules

- pattern : kafka.server<type=(.+), name=(.+), clientId=(.+), topic=(.+), partition=(.\*)><>Value

name: kafka\_server\_$1\_$2

type: GAUGE

labels:

clientId: "$3"

topic: "$4"

partition: "$5"

- pattern : kafka.server<type=(.+), name=(.+), clientId=(.+), brokerHost=(.+), brokerPort=(.+)><>Value

name: kafka\_server\_$1\_$2

type: GAUGE

labels:

clientId: "$3"

broker: "$4:$5"

- pattern : kafka.coordinator.(\w+)<type=(.+), name=(.+)><>Value

name: kafka\_coordinator\_$1\_$2\_$3

type: GAUGE

# Generic per-second counters with 0-2 key/value pairs

- pattern: kafka.(\w+)<type=(.+), name=(.+)PerSec\w\*, (.+)=(.+), (.+)=(.+)><>Count

name: kafka\_$1\_$2\_$3\_total

type: COUNTER

labels:

"$4": "$5"

"$6": "$7"

- pattern: kafka.(\w+)<type=(.+), name=(.+)PerSec\w\*, (.+)=(.+)><>Count

name: kafka\_$1\_$2\_$3\_total

type: COUNTER

labels:

"$4": "$5"

- pattern: kafka.(\w+)<type=(.+), name=(.+)PerSec\w\*><>Count

name: kafka\_$1\_$2\_$3\_total

type: COUNTER

- pattern: kafka.server<type=(.+), client-id=(.+)><>([a-z-]+)

name: kafka\_server\_quota\_$3

type: GAUGE

labels:

resource: "$1"

clientId: "$2"

- pattern: kafka.server<type=(.+), user=(.+), client-id=(.+)><>([a-z-]+)

name: kafka\_server\_quota\_$4

type: GAUGE

labels:

resource: "$1"

user: "$2"

clientId: "$3"

# Generic gauges with 0-2 key/value pairs

- pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.+), (.+)=(.+)><>Value

name: kafka\_$1\_$2\_$3

type: GAUGE

labels:

"$4": "$5"

"$6": "$7"

- pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.+)><>Value

name: kafka\_$1\_$2\_$3

type: GAUGE

labels:

"$4": "$5"

- pattern: kafka.(\w+)<type=(.+), name=(.+)><>Value

name: kafka\_$1\_$2\_$3

type: GAUGE

# Emulate Prometheus 'Summary' metrics for the exported 'Histogram's.

#

# Note that these are missing the '\_sum' metric!

- pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.+), (.+)=(.+)><>Count

name: kafka\_$1\_$2\_$3\_count

type: COUNTER

labels:

"$4": "$5"

"$6": "$7"

- pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.\*), (.+)=(.+)><>(\d+)thPercentile

name: kafka\_$1\_$2\_$3

type: GAUGE

labels:

"$4": "$5"

"$6": "$7"

quantile: "0.$8"

- pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.+)><>Count

name: kafka\_$1\_$2\_$3\_count

type: COUNTER

labels:

"$4": "$5"

- pattern: kafka.(\w+)<type=(.+), name=(.+), (.+)=(.\*)><>(\d+)thPercentile

name: kafka\_$1\_$2\_$3

type: GAUGE

labels:

"$4": "$5"

quantile: "0.$6"

- pattern: kafka.(\w+)<type=(.+), name=(.+)><>Count

name: kafka\_$1\_$2\_$3\_count

type: COUNTER

- pattern: kafka.(\w+)<type=(.+), name=(.+)><>(\d+)thPercentile

name: kafka\_$1\_$2\_$3

type: GAUGE

labels:

quantile: "0.$4"

# Generic gauges for MeanRate Percent

# Ex) kafka.server<type=KafkaRequestHandlerPool, name=RequestHandlerAvgIdlePercent><>MeanRate

- pattern: kafka.(\w+)<type=(.+), name=(.+)Percent\w\*><>MeanRate

name: kafka\_$1\_$2\_$3\_percent

type: GAUGE

- pattern: kafka.(\w+)<type=(.+), name=(.+)Percent\w\*><>Value

name: kafka\_$1\_$2\_$3\_percent

type: GAUGE

- pattern: kafka.(\w+)<type=(.+), name=(.+)Percent\w\*, (.+)=(.+)><>Value

name: kafka\_$1\_$2\_$3\_percent

type: GAUGE

labels:

"$4": "$5"

1. Говорим Kafka Broker использовать JMX exporter:

* cd /home/irina/kafka/kafka\_2.13-3.2.0/bin – необходимо поменять путь на то, где у вас находятся файлы
* sudo vim kafka-server-start.sh

после if [ "x$KAFKA\_LOG4J\_OPTS" = "x" ]; then

    export KAFKA\_LOG4J\_OPTS="-Dlog4j.configuration=file:$base\_dir/../config/log4j.properties"

fi

где «EXTRA\_ARGS=${EXTRA\_ARGS-'-name kafkaServer -loggc'}» меняем на

EXTRA\_ARGS=${EXTRA\_ARGS-"-name kafkaServer -loggc -javaagent:/home/irina/kafka/kafka\_2.13-3.2.0/libs/jmx\_prometheus\_javaagent-0.17.0.jar=7075:/home/irina/kafka//kafka\_2.13-3.2.0/config/jmx\_prometheus\_javaagent-0.17.0.yaml"} ---- пути ставим свои

1. может понадобится предоставления разрешений скаченным файлам, можно через правую кнопку 🡪 разрешения, их предоставить
2. с доп ошибками, которые возникали, помогли справиться сл ссылки:

[**https://stackoverflow.com/questions/36970622/kafka-unrecognized-vm-option-printgcdatestamps**](https://stackoverflow.com/questions/36970622/kafka-unrecognized-vm-option-printgcdatestamps) **-- этот ответ помог :** Delete the $KAFKA\_GC\_LOG\_OPTS option. Might be a hack, but at least it gets the kafka zookeeper server to start! – в самом низу kafka/bin/kafka-run-class.sh

возможно помог ещё вот этот комментарий: <https://stackoverflow.com/questions/62275073/bash-bin-kafka-topics-sh-no-such-file-or-directory-installed-via-ambari> - A quick solution is to open **kafka-run-class.sh** and enable #!/bin/bash, then add "set -x" without the quotation to the next line. run the command again

**PROMETHEUS**

<https://computingforgeeks.com/install-prometheus-server-on-debian-ubuntu-linux/> - устанавливала по этой ссылке:

sudo groupadd --system prometheus

sudo useradd -s /sbin/nologin --system -g prometheus Prometheus

sudo mkdir /var/lib/Prometheus

for i in rules rules.d files\_sd; do sudo mkdir -p /etc/prometheus/${i}; done

sudo apt update

sudo apt -y install wget curl vim

mkdir -p /tmp/prometheus && cd /tmp/prometheus

curl -s https://api.github.com/repos/prometheus/prometheus/releases/latest | grep browser\_download\_url | grep linux-amd64 | cut -d '"' -f 4 | wget -qi –

tar xvf prometheus\*.tar.gz

cd prometheus\*/

sudo mv prometheus promtool /usr/local/bin/

prometheus –version

promtool –version

sudo mv prometheus.yml /etc/prometheus/prometheus.yml

sudo mv consoles/ console\_libraries/ /etc/prometheus/

cd $HOME

sudo vim /etc/prometheus/prometheus.yml

sudo tee /etc/systemd/system/prometheus.service<<EOF

[Unit]

Description=Prometheus

Documentation=https://prometheus.io/docs/introduction/overview/

Wants=network-online.target

After=network-online.target

[Service]

Type=simple

User=prometheus

Group=prometheus

ExecReload=/bin/kill -HUP \$MAINPID

ExecStart=/usr/local/bin/prometheus \

--config.file=/etc/prometheus/prometheus.yml \

--storage.tsdb.path=/var/lib/prometheus \

--web.console.templates=/etc/prometheus/consoles \

--web.console.libraries=/etc/prometheus/console\_libraries \

--web.listen-address=0.0.0.0:9090 \

--web.external-url=

SyslogIdentifier=prometheus

Restart=always

[Install]

WantedBy=multi-user.target

EOF

for i in rules rules.d files\_sd; do sudo chown -R prometheus:prometheus /etc/prometheus/${i}; done

for i in rules rules.d files\_sd; do sudo chmod -R 775 /etc/prometheus/${i}; done

sudo chown -R prometheus:prometheus /var/lib/prometheus/

sudo systemctl daemon-reload

sudo systemctl start prometheus

sudo systemctl enable prometheus

systemctl status prometheus

sudo ufw allow 9090/tcp ---- если firewall

**KAFKA-EXPORTER**

sudo docker run -d --name kafka-exporter -p 9308:9308 danielqsj/kafka-exporter --kafka.server=172.17.0.1:9092

Порт можно узнать запустив команду hostname -I (<https://pingvinus.ru/note/determine-ip-address>)

Если возникает конфликт образов, то можно запустить команду sudo docker system prune . Осторожно если есть нужные образы (Remove all unused containers, networks, images (both dangling and unreferenced), and optionally, volumes. – оф документация)

Подробнее об экспортёре - <https://github.com/danielqsj/kafka_exporter>

**PROMETHEUS/ KAFKA-EXPORTER/JMX-EXPORTER**

sudo vim /etc/prometheus/prometheus.yml

внизу где job\_name: 'prometheus' меняем на:

scrape\_configs:

# The job name is added as a label `job=<job\_name>` to any timeseries scraped from this config.

- job\_name: "kafka"

# metrics\_path defaults to '/metrics'

# scheme defaults to 'http'.

static\_configs:

- targets: ["localhost:7075"]

- job\_name: "kafka-exporter"

# metrics\_path defaults to '/metrics'

# scheme defaults to 'http'.

static\_configs:

- targets: ["localhost:9308"]

tls\_config:

insecure\_skip\_verify: true

На порту localhost:9090 должны появиться в status 🡪 Targets должно появится 2 endpoint

Помогли статьи:

<https://computingforgeeks.com/monitor-apache-kafka-with-prometheus-and-grafana/>

<https://computingforgeeks.com/install-prometheus-server-on-debian-ubuntu-linux/>

<https://kamaok.org.ua/?p=3345>

DASHBOARDS:

KAFKA-EXPORTER – 11962

JMX-EXPORTER – 721