Инструкция по ручному развертыванию Kafka для Wisla

В данной инструкции описано как установить Kafka в ручном режиме без использования скрипта предварительной настройки системы.

Этап 1. Переход под пользователя wisla:

На первом этапе необходимо перейти под пользователя Wisla используя команду:

```
sudo su - wisla
```

Если пользователь wisla еще не существует в системе его необходимо создать используя следующие команды пошагово.

1.Команда создания пользователя:

```
sudo useradd -d /home/wisla -m -s /bin/bash wisla
```

2.Команда установки пароля:

```
sudo passwd wisla
```

3.Добавление пользователя в файлы sudoers:

```
echo "wisla    ALL=(ALL:ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/wisla
```

4.Настройка системных лимитов для пользователя wisla:

```
# Создаем файл лимитов
cat > /etc/security/limits.d/wisla << 'EOF'
wisla   soft      nofile  32768
wisla   hard      nofile  32768
wisla   soft      nproc   32768
wisla   hard      nproc   32768
EOF
```

5. Добавляем в PAM (если нет):

```
echo "session required pam_limits.so" >> /etc/pam.d/common-session
```

Этап 2. Подготовка каталога kafka.

После перехода под пользователя wisla вам необходимо создать каталог где будет располагаться портативная Java и бинарный дистрибутив Kafka использую следующие команды:

1.Создание директории в /opt/:

```
sudo mkdir -p /opt/kafka
```

2.Проверить кому принадлежит каталог kafka:

```
ls -l /opt/
```

3.Сменить пользователя для каталога kafka если принадлежит не wisla используя следующую команду:

```
sudo chown wisla:wisla /opt/kafka
```

Этап 3. Копирование архива Java**:**

После создания каталога необходимо подготовить портативную Java для распаковки, предполагаем, что архив Java уже скачан и находится в доступном месте.

1.Скопируем его в домашнюю директорию wisla используя следующую команду:

```
cp OpenJDK17U-jdk_x64_linux_hotspot_17.0.17_10.tar.gz /home/wisla/
```

2.Переходим в каталог kafka используя следующую команду:

```
cd /opt/kafka
```

3.Распаковка Java:

```
tar -xzf ~/OpenJDK17U-jdk_x64_linux_hotspot_17.0.17_10.tar.gz
```

4.Проверяем результат распаковки Java:

```
ls -la /opt/kafka/
```

5.Устанавливаем переменную окружения:

```
export JAVA_HOME="/opt/kafka/jdk-17.0.17+10"
export PATH="$JAVA_HOME/bin:$PATH"
```

6.Проверяем версию Java:

```
java -version
```

Этап 4. Копирование архива Kafka**:**

После подготовки портативной Java, необходимо подготовить бинарный дистрибутив Kafka, предполагаем, что Kafka уже скачана и находится в доступном месте.

1.Копируем архив Kafka в домашнюю директорию:

```
cp kafka_2.13-4.1.1.tgz /opt/kafka/
```

2.Переходим в каталог kafka используя следующую команду:

```
cd /opt/kafka
```

3.Распаковываем Kafka в /opt/kafka:

```
tar -xzf ~/kafka_2.13-4.1.1.tgz --strip-components=1 -C /opt/kafka
```

4.Проверяем результат распаковки Kafka:

```
ls -la /opt/kafka/bin/kafka-server-start.sh
```

5.Создаем дополнительные рабочие директории для Kafka:

```
mkdir -p /opt/kafka/logs
mkdir -p /opt/kafka/config
mkdir -p /opt/kafka/config/kraft

sudo mkdir -p /var/lib/kafka
sudo chown wisla:wisla /var/lib/kafka
```

Этап 5. Конфигурация Kafka**:**
Теперь нам необходимо сконфигурировать Kafka.

1.Для начала определим IP адрес который будем использовать для конфигурации Kafka:

```
IP_ADDRESS=$(ip route get 1 2>/dev/null | awk '{print $7; exit}')
```

2.Генерируем Cluster ID:

```
CLUSTER_ID=$(/opt/kafka/bin/kafka-storage.sh random-uuid)
echo "Сгенерированный Cluster ID: $CLUSTER_ID"
```

ЗАПИШИТЕ ЭТОТ ID - он понадобится позже! (Пример моего ID : IdLVHil7RvCVrR9U2DEvKQ)

3.Создание конфигурационного файла:

```
sudo cat > /opt/kafka/config/kraft/server.properties << EOF
process.roles=broker,controller
node.id=1
controller.quorum.voters=1@${IP_ADDRESS}:9093
listeners=PLAINTEXT://${IP_ADDRESS}:9092,CONTROLLER://${IP_ADDRESS}:9093
advertised.listeners=PLAINTEXT://${IP_ADDRESS}:9092
log.dirs=/var/lib/kafka
num.partitions=1
num.recovery.threads.per.data.dir=1
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
log.retention.hours=168
log.segment.bytes=1073741824
log.retention.check.interval.ms=300000
controller.listener.names=CONTROLLER
sasl.mechanism.controller.protocol=PLAINTEXT
```

```
auto.create.topics.enable=true
EOF
```

4.Проверяем созданный файл:

```
cat /opt/kafka/config/kraft/server.properties
```

5.Форматирование хранилища

```
if [ ! -f /var/lib/kafka/meta.properties ]; then
    echo "Инициализируем хранилище Kafka..."
    /opt/kafka/bin/kafka-storage.sh format \
        -t $CLUSTER_ID \
        -c /opt/kafka/config/kraft/server.properties

    echo "Хранилище инициализировано с Cluster ID: $CLUSTER_ID"
else
    echo "Хранилище уже инициализировано"
    cat /var/lib/kafka/meta.properties
fi
```

6.Создание файла службы:

Создаем файл службы

```
sudo cat > /etc/systemd/system/kafka.service << EOF
[Unit]
Description=Apache Kafka Service (KRaft mode)
After=network.target

[Service]
Type=simple
User=wisla
Group=wisla
WorkingDirectory=/opt/kafka
Environment="JAVA_HOME=/opt/kafka/jdk-17.0.17+10"
Environment="PATH=/opt/kafka/jdk-17.0.17+10/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
ExecStartPre=/bin/bash -c "if [ ! -f /var/lib/kafka/meta.properties ]; then /opt/kafka/bin/kafka-storage.sh format -t $CLUSTER_ID -c /opt/kafka/
config/kraft/server.properties; fi"
ExecStart=/opt/kafka/bin/kafka-server-start.sh /opt/kafka/config/kraft/server.properties
SuccessExitStatus=143
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
EOF
```

Создание скрипта запуска:

```
sudo cat > /usr/local/bin/kafka-start << 'EOF'
#!/bin/bash
KAFKA_DIR="/opt/kafka"
KAFKA_LOG_DIR="/var/lib/kafka"
CLUSTER_ID="$CLUSTER_ID"  # ВСТАВЬТЕ ВАШ Cluster ID!

if [ ! -f $KAFKA_LOG_DIR/meta.properties ]; then
```

```
    $KAFKA_DIR/bin/kafka-storage.sh format -t $CLUSTER_ID -c $KAFKA_DIR/config/kraft/server.properties
fi
systemctl start kafka
echo "Kafka запущен в режиме KRaft"
EOF
```

Создаем скрипт остановки:

```
sudo cat > /usr/local/bin/kafka-stop << 'EOF'
#!/bin/bash
systemctl stop kafka
echo "Kafka остановлен"
EOF
```

Создаем скрипт статуса:

```
sudo cat > /usr/local/bin/kafka-status << 'EOF'
#!/bin/bash
echo "Kafka статус:"
systemctl status kafka --no-pager
EOF
```

Создаем скрипт сброса:

```
sudo cat > /usr/local/bin/kafka-reset << 'EOF'
#!/bin/bash
echo "Сброс Kafka хранилища..."
systemctl stop kafka
KAFKA_DIR="/opt/kafka"
KAFKA_LOG_DIR="/var/lib/kafka"
CLUSTER_ID="$CLUSTER_ID"  # ВСТАВЬТЕ ВАШ Cluster ID!

rm -rf $KAFKA_LOG_DIR/*
$KAFKA_DIR/bin/kafka-storage.sh format -t $CLUSTER_ID -c $KAFKA_DIR/config/kraft/server.properties
echo "Хранилище сброшено. Запустите: systemctl start kafka"
EOF
```

Выдача прав на исполнение:

```
chmod +x /usr/local/bin/kafka-*
```

Этап 6. Запуск и проверка Kafka:

Обновляем systemd:

```
sudo systemctl daemon-reload
```

Включаем автозагрузку:

```
sudo systemctl enable kafka
```

Запускаем Kafka

```
sudo systemctl start kafka
```

Проверяем статус

```
sudo systemctl status kafka
```

Смотрим логи:

```
sudo journalctl -u kafka -f
```

Проверяем, что порты открыты

```
netstat -tlnp | grep -E "9092|9093"
```

Проверяем метаданные

```
ls -la /var/lib/kafka/
```

Установка завершена! Kafka готова к использованию.
