

Рекомендательная платформа MOODREC

Документация по установке

Документация для администраторов

Данная документация предназначена для персонала, производящего установку и обслуживание сервиса MoodRec на собственном оборудовании. Эта документация содержит информацию по развертыванию сервиса, его конфигурации и дальнейшему обновлению.

Предварительная подготовка

- Перед началом установки настоятельно рекомендуем ознакомиться с техническими требованиями к оборудованию и персоналу. Они подробно описаны в функциональных требованиях к MoodRec.
- Все дальнейшие действия по установке требуют наличия привилегий пользователя `root`.
- Для паролей и секретных комбинаций используется набор из символов – `abcdefghijklmnopqrstuvwxyzABCDEF`. Обязательно сгенерируйте более сложный пароль, состоящий из не менее 32-х символов.

Установка

В данном руководстве показан процесс установки для Debian-подобной ОС (например, Astra Linux или Ubuntu). Для non-Debian ОС процесс выглядит похожим образом, кроме установки некоторых зависимостей.

Загрузка архива

Всем клиентам предоставляется архив с расширением `.tar.gz`, распакуйте его и переместите содержимое в каталог `/opt`:

```
{
  tar xzf <архив>.tar.gz
  mv moodrec /opt/moodrec
}
```

Проверка времени на сервере

Синхронизация времени между серверами критична: различия могут привести к нарушению работы и ошибочной статистике. Настройте [NTP-сервер](#) и проверьте синхронизацию:

```
timedatectl status
```

Установка системных лимитов

Добавьте следующие лимиты в файл `/etc/security/limits.conf`:

```
*          -          nofile          1048576
root       -          nofile          1048576
root       -          memlock        unlimited
```

Затем переподключитесь и проверьте, что лимиты применились:

```
ulimit -n
```

Установка зависимостей

Установите следующие зависимости:

```
{
  apt update
  apt install --no-install-recommends gcc g++ libc6-dev
}
```

Установка необходимых компонентов

Установка Kafka

Kafka используется как основной брокер сообщений для работы сервиса.

Установка происходит согласно инструкциям:

- [Быстрый старт](#)
- [С чего начать](#)

Установка MongoDB

TIP

MongoDB используется для хранения системных настроек и пользовательских данных. Ниже приведена инструкция по установке из архива.

Дополнительные материалы:

- [Установка MongoDB Community Edition на Ubuntu](#)
- [Описание файла конфигурации](#)
- [Конфигурация базы данных](#)

Перейдите на [страницу загрузки](#), выберите версию 8.0, вашу версию ОС и архив `.tgz`. Скопируйте ссылку и скачайте архив:

```
wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-ubuntu2404-8.0.17.tgz
```

Распакуйте архив в `/usr/local` и создайте символическую ссылку:

```
{
  tar xzf mongodb-linux-x86_64-ubuntu2404-8.0.17.tgz -C /usr/local
  ln -s /usr/local/mongodb-linux-x86_64-ubuntu2404-8.0.17 /usr/local/mongodb
}
```

Установите зависимости:

```
{
  apt update
  apt install -y gnupg curl
}
```

Создайте пользователя, группу и рабочие каталоги:

```
{
  groupadd mongodb
  useradd -r -g mongodb -s /bin/false mongodb
  mkdir -p /var/{lib,log}/mongodb
  chown -R mongodb:mongodb /var/{lib,log}/mongodb
}
```

INFO

По умолчанию в MongoDB отключена аутентификация. Для повышения безопасности вы можете включить аутентификацию с использованием механизма SCRAM (аутентификация по паролю).

Создайте файл конфигурации `/etc/mongod.conf` :

```
# mongod.conf

# for documentation of all options, see:
# http://docs.mongodb.org/manual/reference/configuration-options/

storage:
  dbPath: /var/lib/mongodb
  wiredTiger:
    engineConfig:
      directoryForIndexes: true

systemLog:
  destination: file
  logAppend: true
  logRotate: reopen
  path: /var/log/mongodb/mongod.log

net:
  port: 27017
  bindIp: 127.0.0.1
```

INFO

Вы можете включить поддержку TLS для подключения к MongoDB. Для этого добавьте в файл `/etc/mongod.conf` следующие параметры:

```
net:
  tls:
    mode: requireTLS
    certificateKeyFile: <путь до файла с приватным ключом и сертификатом сервера (PEM)>
    CAFile: <путь до сертификата центра сертификации (CA)>
```

После перезапуска сервиса, MongoDB будет принимать подключения только от клиентов, использующих TLS.

Создайте systemd-сервис по пути `/etc/systemd/system/mongod.service` :

```
[Unit]
Description=MongoDB Database Server
Documentation=https://docs.mongodb.org/manual
After=network.target

[Service]
User=mongodb
Group=mongodb
ExecStart=/usr/local/mongodb/bin/mongod --config /etc/mongod.conf
PIDFile=/run/mongod.pid
Restart=always
LimitFSIZE=infinity
LimitCPU=infinity
LimitAS=infinity
LimitNOFILE=256000
LimitNPROC=256000
LimitMEMLOCK=infinity
TasksMax=infinity
TasksAccounting=false

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

Добавьте сервис в автозагрузку, запустите и проверьте статус:

```
{
  systemctl enable mongod.service
  systemctl start mongod.service
  systemctl status mongod.service
}
```

Скачайте и установите [MongoDB Shell](#), подключитесь к базе данных и создайте пользователя в базе `admin` :

```
use admin;
db.createUser({
  "user": "moodrec",
  "pwd": "abcdefghijklmnopqrstuvwxyABCDEF",
  "roles": [{
    "role": "root",
    "db": "moodrec"
  }]
})
```

Обновите параметры подключения в конфигурации `main_config.yaml`:

```
mongo: &mongo_config
database: "moodrec"
username: "moodrec"
password: "abcdefghijklmnopqrstuvwxyABCDEF"
auth_source: "admin"
host: "127.0.0.1"
port: "27017"
```

Установка ClickHouse

TIP

База данных ClickHouse используется для хранения истории, статистики и формирования пользовательских отчетов. Ниже представлена инструкция по установке из DEB-пакетов.

Установите зависимости и добавьте официальный репозиторий:

```
{
  apt update
  apt install -y apt-transport-https ca-certificates dirmngr gnupg
  apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 8919F6BD2B48D754
  echo "deb https://packages.clickhouse.com/deb stable main" | tee
  /etc/apt/sources.list.d/clickhouse.list
}
```

Установите ClickHouse:

```
{
  apt update
  apt install -y clickhouse-server clickhouse-client
}
```

Добавьте сервис в автозагрузку и запустите:

```
{
  systemctl enable clickhouse-server.service
  systemctl start clickhouse-server.service
  systemctl status clickhouse-server.service
}
```

Обновите параметры подключения в конфигурации `configs/release/main_config.yaml`:

```
clickHouse: &clickHouse_config
host: "127.0.0.1"
port: 9000
user: "moodrec"
password: "moodrec"
db_name: "moodrec"
```

Установка Kvrocks

ТИП

Kvrocks — альтернатива Redis, построенная на RocksDB. В отличие от Redis, он сохраняет данные на диск, что делает его подходящим для хранения больших объемов данных.

Установите зависимости для сборки:

```
{
  apt update
  apt install -y git build-essential cmake libtool python3
}
```

Перейдите на [страницу релизов](#) проекта на GitHub, скачайте последнюю версию и распакуйте архив:

```
{
  wget https://github.com/apache/kvrocks/archive/refs/tags/v2.12.1.tar.gz
  tar xzf v2.12.1.tar.gz
}
```

Выполните сборку и установку:

```
{
  cd kvrocks-2.12.1
  ./x.py build -DPORTABLE=1 -DCMAKE_BUILD_TYPE=Release -j $(nproc)
  mv build/kvrocks /usr/local/bin/
}
```

Создайте пользователя и каталоги:

```
{
  groupadd kvrocks
  useradd -r -g kvrocks -s /bin/false kvrocks
  mkdir -p /var/{lib,log}/kvrocks /etc/kvrocks
  chown -R kvrocks:kvrocks /var/{lib,log}/kvrocks
}
```

Создайте файл конфигурации `/etc/kvrocks/kvrocks.conf`:

```
# kvrocks.conf
bind 0.0.0.0
port 6666
db-name moodrec.db
dir /var/lib/kvrocks
log-dir /var/log/kvrocks
log-retention-days 7
backup-dir /var/lib/kvrocks/backup
requirepass abcdefghijklmnopqrstuvwxyzABCDEF
supervised systemd
workers 8
```

Создайте systemd-сервис по пути `/etc/systemd/system/kvrocks.service`:

```
[Unit]
Description=kvrocks SSD key-value database
Documentation=https://github.com/apache/kvrocks
Wants=network-online.target
After=network-online.target

[Service]
User=kvrocks
Group=kvrocks
Type=notify
ExecStart=/usr/local/bin/kvrocks -c /etc/kvrocks/kvrocks.conf
WorkingDirectory=/var/lib/kvrocks
Restart=on-failure
ExecStop=/bin/kill -s TERM $MAINPID
RestartSec=10s
LimitNOFILE=100000
LimitNPROC=4096
TimeoutSec=300
NoNewPrivileges=yes

[Install]
WantedBy=multi-user.target
Alias=kvrocks.service
```

Добавьте сервис в автозагрузку и запустите:

```
{
  systemctl enable kvrocks.service
  systemctl start kvrocks.service
  systemctl status kvrocks.service
}
```

Обновите параметры подключения в конфигурации `configs/release/main_config.yaml`:

```
redis: &redis_config
  host: "127.0.0.1"
  port: 6666
  user: "moodrec"
  password: "abcdefghijklmnopqrstuvwxyzABCDEF"
  read_timeout: 10
  write_timeout: 10
  pool_size: 100
```

Запуск сервиса

Перед запуском сервиса убедитесь, что вы заполнили файл конфигурации `main_config.yaml`. Описание всех параметров доступно в разделе **Параметры конфигурации**.

Для управления микросервисами MoodRec используется исполняемый файл `moodrec`. Справка по командам:

Usage:

```
moodrec [flags]
moodrec [command]
```

Available Commands:

```
list          List platform information
start         Start platform services
stop          Stop platform services
restart       Restart platform services
reload-nginx  Reload nginx configuration
help          Help about any command
completion    Generate the autocompletion script for the specified shell
```

Flags:

```
--appname      show application full name
--config string config path (default
"/opt/configs/release/main_config.yaml")
--debug        debug mode (for development purposes)
-h, --help     help for moodrec
-v, --version  version for moodrec
```

Use "moodrec [command] --help" for more information about a command.

На этом установка завершена, сервис готов к запуску и дальнейшей настройке. Дальнейшая настройка происходит в интерфейсе MoodRec.

Работу MoodRec обеспечивают следующие микропроцессы.

Сервисы

Сервис	Назначение
vectors	Работа с векторным хранилищем и поиск
trans	Преобразование описаний товаров в векторные представления
tasks	Выполнение отложенных заданий
offers	Управление продуктами
gateway	Проксирование HTTP (JSON) → gRPC (Proto)
events	Обработка событий
catboss	Работа с рекомендациями
auth	Авторизация
audience	Управление аудиториями

Проверка статуса всех сервисов:

```
moodrec status
```

Проверка статуса конкретного сервиса:

```
moodrec status vectors
moodrec status trans
moodrec status tasks
moodrec status offers
moodrec status gateway
moodrec status events
moodrec status catboss
moodrec status auth
moodrec status audience
```

Если сервис запущен корректно, вывод содержит `active (running)`. Если сервис остановлен — `inactive` или `failed`.

Управление конкретным сервисом выполняется отдельно для каждого микропроцесса.

Запуск:

```
moodrec start vectors
```

Остановка:

```
moodrec stop vectors
```

Перезапуск:

```
moodrec restart vectors
```

Аналогично для остальных сервисов:

```
moodrec start <service>  
moodrec stop <service>  
moodrec restart <service>
```

где `<service>` — один из: `vectors`, `trans`, `tasks`, `offers`, `gateway`, `events`, `catboss`, `auth`, `audience`.

После выполнения операций рекомендуется проверить состояние:

```
moodrec status <service>
```