

Оглавление

Установка Sherlock Platform на Linux	2
(на примере ОС Ubuntu 22.04)	2
Требования к оборудованию и программному обеспечению:.....	2
Оборудование.....	2
Для сервера приложений	2
Для сервера баз данных	2
Для сервера очередей	2
Установка базы данных Sherlock Platform на Linux (на примере ОС Ubuntu 22.04)	3
Установка Web приложений Sherlock Platform.....	3
Установка лицензии Sherlock Platform	6
Установка сервиса Entry Point на отдельный Linux сервер.....	7
Необходимые для работы порты.....	8
Проверка работы Sherlock Platform	8
Приложение 1. Пример конфигурации nginx.....	9

Установка Sherlock Platform на Linux

(на примере ОС Ubuntu 22.04)

В данном разделе приведено детальное описание шагов по установке и настройке Sherlock Platform, начиная с версии backend 2.4.x. Скриншоты и пояснения приведены для Microsoft Windows Server 2016. Для других операционных могут незначительно отличаться детали.

Требования к оборудованию и программному обеспечению:

Оборудование

1. ОП: не менее 4Gb

Для сервера приложений

1. Операционная система и компоненты
 - a. Операционная система: Linux-совместимая ОС (Ubuntu)
 - b. Nginx
 - c. NetCore
 - d. Redis
2. Для сервера должен быть открыт в публичную сеть 443 (https) порт (если иное не согласовано в архитектуре развёртывания)

Для сервера баз данных

1. Операционная система: Linux-совместимая ОС (Ubuntu)
2. СУБД: PostgreSQL (лучше последней версии)

Примечание: При малой нагрузке допустимо совмещение сервера БД с сервером приложений

Для сервера очередей

1. Операционная система: Linux-совместимая ОС (Ubuntu)
2. Kafka 3.5

Примечание: При малой нагрузке допустимо размещение Kafka прямо на сервере приложений

Установка базы данных Sherlock Platform на Linux (на примере ОС Ubuntu 22.04)

База данных должна быть развернута на сервере в защищенной части корпоративной сети или на отдельном сервере СУБД (**рекомендуется**). В случае планируемой низкой нагрузки возможно размещение СУБД на едином сервере вместе с сервером приложений

1. Установить postgresql

```
#apt install postgresql postgresql-contrib
```

2. Создать отдельного пользователя PostgreSQL и БД sherlock и дать ему полные права только на данную базу

```
$sudo -u postgres psql
```

```
CREATE USER sherlock WITH PASSWORD 'your_password';
```

```
CREATE DATABASE sherlock OWNER sherlock;
```

```
\q
```

3. Восстановить БД из архива.

```
$pg_restore -U sherlock -W -d sherlock /tmp/sherlock.dump
```

4. Выполнить скрипт pg_update.sql для обновления базы, чтобы гарантировать соответствие версии базы и приложения

```
psql -U sherlock -W -d sherlock < pg_update.sql
```

5. Настроить резервное копирование базы данных или включить в существующий план резервного копирования (**Рекомендуется**)

Важно: для нагруженных платформ рекомендуется установка пула соединений. Подробнее смотрите <https://habr.com/ru/companies/otus/articles/798267/> или <https://selectel.ru/blog/tutorials/how-to-install-pgbouncer-connection-pooler-for-postgresql/>

Установка Web приложений Sherlock Platform

1. Установить все необходимые компоненты

- Установить redis

```
#apt install redis
```

- Установить nginx

```
#apt install nginx
```

- Установить dotnet 8.0

<https://learn.microsoft.com/ru-ru/dotnet/core/install/linux-scripted-manual#scripted-install>

2. Подготовить папку web приложения папку /opt/sherlock
 - a. Поместить содержимое архива в папку /opt/sherlockВ папке sherlock будут находиться 4 подпапки:
 - api
 - endpoint

- evalservice
 - frontend
- b. Переименовать файлы /opt/sherlock/frontend/index_new.html в /opt/sherlock/frontend/index.html соответственно
 - c. Переименовать папку /opt/sherlock/frontend/scripts/chatplatformsample в /opt/sherlock/frontend/scripts/chatplatform

3. Настроить конфигурационные файлы приложений:

- api/appsettings.json
 - i. [SherlockCRMDB](#) - параметры доступа к базе.

Важно:

 - символ / в JSON надо экранировать!
 - ii. DatabaseType – тип базы данных. 1 – MS SQL, 2 - **PostgreSql**
 - iii. Адрес API и каталога загрузки (по умолчанию совпадает с адресом API)


```
"SherlockApiUrl": "https://<Адрес Sherlock Platform>/api/",
          "SherlockAttachments": "https://<Адрес Sherlock Platform>/api/"
```
 - iv. [SherlockWebSiteUrl](#) – указать адрес сайта. Настройка необходима, так как внутренний и внешний адрес могут отличаться
 - v. Путь к папке с файлами


```
"SherlockFilePath": "/opt/sherlock/files/"
```
 - vi. При размещении интерфейса V2 и API V2 на разных доменах необходимо выполнить настройки для поддержки кросс-доменных запросов. Для этого в настройке "CorsOrigins" надо через запятую указать все домены, с которых возможен доступ к API V2.
 - vii. если авторизация будет выполняться через ADFS в конфиге должен быть параметр "[AuthenticationType](#)" со значениями WsFederation, JwtBearer, FacebookAuthentication

Примечание: авторизация через ActiveDirectory (значение [AuthenticationType](#) = WindowsAuthentication) на Linux не поддерживается

Если авторизация будет выполняться через ActiveDirectory или ADFS, то надо добавить дополнительный параметр в тег app-root файла /opt/sherlock/frontend/index.html указать по необходимости дополнительные настройки.:

- authenticationType – возможные значения WindowsAuthentication, WsFederation, JwtBearer, FacebookAuthentication. Если не указано, используется JwtBearer
- adfsUrl – адрес url, применяемый для перехода к странице авторизации ADFS. Например

https://<Adfs_server_Url>/adfs/ls/?wa=wsignin1.0&wtrealm=<Auth_name>&wreply=<Sherlock_URL>

Важно: Дополнительно для ролей в базе надо указать коды групп ADFS.

- endpoint/appsettings.json
 - **SherlockCRMDB** - Параметры доступа к базе. **Важно:** символ / в JSON надо экранировать!
 - **SherlockApiUrl** – указать адрес Url https://<Адрес_Sherlock_Platform>/API/
- evalservice/appsettings.json – скопировать без изменений

4. Запустить сервисы api, endpoint, evalservice на отдельных портах(серверах). Пример конфигурации сервиса api:

```
# файл /etc/systemd/system.api.service
[Unit]
Description=Sherlock API

[Service]
ExecStart=/usr/bin/dotnet /opt/sherlock/api/SherlockPlatform.API.dll --urls "http://127.0.0.1:4001"
WorkingDirectory=/opt/sherlock/api
SyslogIdentifier=api
Environment=DOTNET_ROOT=/usr/lib64/dotnet
RemainAfterExit=yes
Restart=always
User=sherlock
[Install]
WantedBy=multi-user.target
```

Аналогичная конфигурация будет для endpoint и evalservice

5. Настроить Nginx для работы Sherlock Platform. Пример конфигурации приведен в приложении 1. Обязательно должны быть настроены пути для сервисов:
 - a. / - frontend приложения
 - b. /api/v2 – основное API приложения Sherlock
 - c. /endpoint – сервис endpoint
 - d. /evalservice – сервис Eval
6. Настроить SSL сертификат (по запросу может быть установлен сертификат 3-его уровня в домене sherlockcrm.ru). Сертификат должен быть выдан удостоверяющим центром, который соответствует требованиям Google.
7. Дополнительные настройки в файле api/appsettings.json указать
 - d. **SherlockApiPublicUrl** – публичный адрес API. Необходимо указывать, если решение установлено в DMZ

- e. CorsOrigins – список серверов с которых может идти доступ к API. Если значение не указано, то проверки CORS не осуществляются
- f. EntryPointUrl – адрес EntryPoint сервиса, если отличается от стандартной установки
- g. PublicEntryPointUrl – публичный адрес entry point. Необходимо указывать, если решение установлено в DMZ
- h. FileStoreUrl - адрес сервиса архивного хранения файлов
- i. EvalServiceUrl – адрес Eval сервиса, если отличается от стандартной установки
- j. EntryPointThreadCount – количество потоков обработки входящих сообщений. По умолчанию 50. Не рекомендуется устанавливать более чем $[\text{количество ядер}] * 15$. В среднем один поток позволяет обработать 1 сообщение в секунду
- k. BotScenarioThreadCount – количество потоков обработки сценариев ботов. По умолчанию 150. Рекомендуется устанавливать в диапазоне от $[\text{количество ядер}] * 25$ до $[\text{количество ядер}] * 50$. Для достаточной производительности данный параметр должен быть равен количеству активных диалогов

Для настройки вывода логов Kestrel в файл смотрите <https://qna.habr.com/q/279902>

Установка лицензии Sherlock Platform

1. Необходимо направить запрос на лицензию с указанием имени SQL сервера и внутреннего url адреса API (https://<Адрес_Sherlock_Platform>/API/).
2. Полученный файл с лицензиями разместить в директории
/opt/sherlock/license/

Установка сервиса Entry Point на отдельный Linux сервер

В случае необходимости размещения основного сервера в защищенной зоне, в DMZ размещается отдельный сервер на базе Linux

Для его установки необходимо:

1. Установить все необходимые компоненты
 - Установить nginx
 - #apt install nginx**
 - Установить dotnet 8.0
 - <https://learn.microsoft.com/ru-ru/dotnet/core/install/linux-scripted-manual#scripted-install>
2. Подготовить папку web приложения папку /opt/sherlock. Для этого поместить содержимое папки /entrypoint архива в папку /opt/sherlock/entrypoint
3. В файле /opt/sherlock/entrypoint/appsettings.json указать
 - a. **SherlockCRMDB** - Параметры доступа к базе
 - b. **SherlockApiUrl** – указать адрес Url
4. Дополнительно надо настроить вывод метода

/api/v2/files

Маска для фильтрации:

~* /api/v2/files

Для работы чата на сайт необходимо настроить вывод дополнительных методов (по списку ниже) API основного сервера в Интернет на Nginx. Маска для фильтрации:

~* /api/v2/(token|chat|message|botcontext|bots|call|client|files|request|setting|userprofile|widget|webchathub)

```
/api/v2/BotContext
/api/v2/BotContext/Batch
/api/v2/Bots/RunBot
/api/v2/Call/CallByPhone?phoneNumber=
/api/v2/Chat/CreateChatAsync
/api/v2/Chat/GetUnreadMessageCount?externalId=${externalId}
/api/v2/Client
/api/v2/Files?objectId=${objectId}&systemDictionaryId=35df8dfe-d88a-4992-a4d4-54040166ca25&fileFolderType=1
/api/v2/Message/{id}
/api/v2/Message/GetById/{messageId}
/api/v2/Request/' + requestId + '/SetQualityGrade?grade...
/api/v2/Setting/EditDeleteClientTimeout
/api/v2/Token/TokenByAccessToken
/api/v2/UserProfile/GetUserProfilesByIds
/api/v2/Widget/...
/api/v2/webchathub/...
```

С сервера EntryPoint до основного сервера должны быть открыт 443 порт в обе стороны

Необходимые для работы порты

1. PostgreSQL – порт 5432
2. Kafka – порт 9092 (9093)

Проверка работы Sherlock Platform

1. Подключиться к интерфейсу https://<Адрес_Sherlock_Platform>/. Логин/пароль по умолчанию admin@sherlockcrm.ru / 1
2. Проверить работу API V2, набрав https://<Адрес_Sherlock_Platform>/API/V2/Swagger/. Должна отразиться страница с документацией по API
3. Проверить получение/отправку сообщений через любой канал. Например, через чат на сайте. Смотрите пункт Установка тестового чата на сайте

Приложение 1. Пример конфигурации nginx

```
server{
    server_name unix.sherlockcrm.ru;

    location / {
        root /opt/sherlock/frontend;
        index index.html;
        try_files $uri $uri/ /index.html;
    }

    location /api/v2 {
        rewrite (?i)^/api/v2/(.*)$ /$1 break;
        proxy_pass http://127.0.0.1:4001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection keep-alive;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location /entrypoint {
        rewrite (?i)^/entrypoint/(.*)$ /$1 break;
        proxy_pass http://127.0.0.1:4000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection keep-alive;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location /evalservice {
        rewrite (?i)^/evalservice/(.*)$ /$1 break;
        proxy_pass http://127.0.0.1:4002;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection keep-alive;
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location = /robots.txt {
        return 200 "User-agent: *\nDisallow: /\n";
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/unix.sherlockcrm.ru/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/unix.sherlockcrm.ru/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
```

```
}  
server{  
    if ($host = unix.sherlockcrm.ru) {  
        return 301 https://$host$request_uri;  
    } # managed by Certbot  
  
    server_name unix.sherlockcrm.ru;  
  
    listen 80;  
    return 404; # managed by Certbot  
}
```