АЛКО-ИНСПЕКТОР

Руководство по установке

Серверная часть программного обеспечения «Алко-Инспектор»

Windows / Linux

ООО «Градиент»

Оглавление

1. Общие сведения 3
2. Системные требования 3
3. Установка ПО на операционные системы семейства Windows 4
4. Установка ПО на операционные системы семейства Linux 7
5. Настройка обратного прокси-сервера Nginx 9
6. Настройка разрешенных источников запросов (CORS) 11
7. Настройка сетевого доступа 12
8. Проверка работоспособности программного обеспечения 12

1. Общие сведения

Настоящее руководство содержит описание порядка установки, настройки и запуска серверной части программного обеспечения «Алко-Инспектор» (далее — ПО) на операционных системах семейства Windows и Linux.

Серверная часть ПО включает следующие компоненты:

- Backend — серверное приложение, обеспечивающее обработку запросов, взаимодействие с базой данных и комплексами «Алко-Инспектор»;
- Frontend — приложение, обеспечивающее работу пользовательского веб-интерфейса Мониторингового центра;
- PostgreSQL — система управления базами данных;
- Nginx — обратный прокси-сервер, обеспечивающий маршрутизацию запросов и подключение пользователей по протоколу HTTPS.

Порядок развертывания компонентов зависит от используемой операционной системы.

2. Системные требования

2.1. Минимальные системные требования

Параметр	Требование
Процессор	Intel или AMD, 2 ядра, 2,8 ГГц
Оперативная память	4 ГБ
Свободное дисковое пространство	50 ГБ
Сетевое подключение	Ethernet

2.2. Рекомендуемые системные требования

Параметр	Требование
Процессор	Intel или AMD, 4 ядра, 2,8 ГГц
Оперативная память	8 ГБ
Свободное дисковое пространство	100 ГБ
Сетевое подключение	Ethernet

Объем дискового пространства может быть увеличен в зависимости от количества подключенных комплексов, интенсивности поступления событий и установленного срока хранения фото- и видеоматериалов.

2.3. Поддерживаемые операционные системы

- Microsoft Windows Server 2016 и новее;

- Microsoft Windows 10;
- Ubuntu 24.04 LTS;
- ALT Linux.

3. Установка ПО на операционные системы семейства Windows

3.1. Подготовка операционной системы

Перед установкой ПО необходимо установить следующие программные компоненты:

- PostgreSQL версии 16 и новее;
- Nginx версии 1.28 и новее;
- .NET SDK 8.0;
- NSSM (Non-Sucking Service Manager).

Установка и настройка ПО выполняются пользователем, обладающим правами администратора операционной системы.

3.2. Размещение файлов программного обеспечения

Получить комплект поставки ПО у представителей ООО «Градиент».

Распаковать файлы Backend и Frontend в отдельные директории.

Пример структуры размещения файлов:

```
C:\AlcoInspector\
    Backend\
    Frontend\
```

Разместить исполняемый файл NSSM в отдельной директории. Например:

```
C:\Tools\nssm\nssm.exe
```

Указанные пути являются примерами и могут быть изменены администратором.

3.3. Настройка конфигурации ПО

Перед запуском программного обеспечения необходимо внести изменения в конфигурационные файлы Backend и Frontend.

В конфигурационных файлах необходимо указать:

- параметры подключения Backend к базе данных PostgreSQL;
- сетевой адрес и порт Backend;
- сетевой адрес и порт Frontend;
- адрес Backend, используемый Frontend для взаимодействия с серверной частью, при наличии соответствующего параметра конфигурации;
- параметры HTTPS, если они предусмотрены конфигурацией приложений.

В качестве примера далее используются следующие адреса:

Компонент	Адрес
Backend	https://localhost:5001

Компонент	Адрес
Frontend	https://localhost:5002
Внешний адрес ПО	https://monitoring.example.com

Адреса и порты должны соответствовать параметрам, указанным в конфигурации приложений и обратного прокси-сервера Nginx.

3.4. Регистрация Backend в качестве службы Windows

Запустить PowerShell от имени администратора.

Перейти в директорию, содержащую исполняемый файл NSSM:

```
cd C:\Tools\nssm
```

Выполнить команду:

```
.\nssm.exe install AlcoInspector.Backend
```

В открывшемся окне NSSM на вкладке Application указать следующие параметры:

Параметр	Значение
Path	Полный путь к исполняемому файлу Backend
Startup directory	C:\AlcoInspector\Backend
Arguments	Дополнительные аргументы запуска при необходимости

На вкладке Details указать:

Параметр	Значение
Display name	Alco-Inspector Backend
Startup type	Automatic

Нажать кнопку Install service. В результате в операционной системе будет зарегистрирована служба Backend с автоматическим запуском.

3.5. Регистрация Frontend в качестве службы Windows

Выполнить команду:

```
.\nssm.exe install AlcoInspector.Frontend
```

В открывшемся окне NSSM на вкладке Application указать следующие параметры:

Параметр	Значение
Path	Полный путь к исполняемому файлу Frontend
Startup directory	C:\AlcoInspector\Frontend
Arguments	Дополнительные аргументы запуска при необходимости

На вкладке Details указать:

Параметр	Значение
Display name	Alco-Inspector Frontend
Startup type	Automatic

Нажать кнопку Install service. В результате в операционной системе будет зарегистрирована отдельная служба Frontend с автоматическим запуском.

Если приложение поставляется в виде DLL-файла, в поле Path необходимо указать путь к исполняемому файлу dotnet.exe, а в поле Arguments — путь к соответствующему DLL-файлу приложения.

3.6. Запуск служб

Перед запуском служб необходимо убедиться, что PostgreSQL запущен и доступен для подключения.

Для запуска зарегистрированных служб выполнить следующие команды:

```
Start-Service AlcoInspector.Backend
Start-Service AlcoInspector.Frontend
```

Для проверки состояния служб выполнить команду:

```
Get-Service AlcoInspector.Backend, AlcoInspector.Frontend
```

При успешном запуске обе службы должны находиться в состоянии Running.

Для управления службами также допускается использование стандартной оснастки Windows «Службы». Для ее открытия выполнить команду:

```
services.msc
```

3.7. Проверка работоспособности ПО

Для проверки доступности Backend выполнить команду:

```
curl.exe -k https://localhost:5001
```

При корректной работе Backend ожидается ответ:

```
Server is running
```

Для проверки доступности Frontend выполнить команду:

```
curl.exe -k https://localhost:5002
```

Также допускается проверка доступности Frontend посредством открытия соответствующего адреса в веб-браузере.

Параметр -k используется исключительно для локальной диагностики при наличии самоподписанных сертификатов.

После проверки работоспособности компонентов необходимо выполнить настройку Nginx в соответствии с разделом 5 настоящего руководства.

4. Установка ПО на операционные системы семейства Linux

4.1. Подготовка операционной системы

Для развертывания ПО на операционных системах семейства Linux используется технология контейнеризации Docker.

На сервере должны быть установлены:

- Docker Engine;
- Docker Compose V2;
- Nginx.

Backend, Frontend, PostgreSQL и необходимые для их работы программные зависимости поставляются в составе контейнерной инфраструктуры.

Дополнительная установка PostgreSQL, .NET SDK и других зависимостей приложений непосредственно в операционную систему не требуется.

4.2. Получение файлов инфраструктуры

Получить у представителей ООО «Градиент»:

- комплект конфигурационных файлов контейнерной инфраструктуры;
- файл docker-compose.yaml;
- данные для получения образов программного обеспечения.

Разместить полученные файлы в отдельной директории на сервере. Например:

```
/opt/alcoinspector/
```

Перейти в указанную директорию:

```
cd /opt/alcoinspector
```

При использовании закрытого реестра контейнерных образов необходимо предварительно выполнить авторизацию с использованием предоставленных учетных данных:

```
sudo docker login <registry-address>
```

4.3. Настройка конфигурационных файлов

Перед запуском контейнерной инфраструктуры необходимо внести изменения в предоставленные конфигурационные файлы.

Указать актуальные сетевые адреса Backend и Frontend, соответствующие инфраструктуре заказчика.

При необходимости изменить:

- внешний доменный адрес ПО;
- адрес Backend, используемый Frontend;
- публикуемые сетевые порты контейнеров;
- параметры подключения к базе данных.

При настройке взаимодействия между контейнерами необходимо учитывать, что адрес localhost внутри контейнера указывает на сам контейнер.

Для межконтейнерного взаимодействия следует использовать имена соответствующих сервисов Docker Compose, если такой способ подключения предусмотрен конфигурацией.

4.4. Запуск контейнерной инфраструктуры

Для развертывания и запуска программного обеспечения выполнить команду:

```
sudo docker compose up -d
```

Docker Compose выполнит получение необходимых образов при их отсутствии, создание контейнеров и запуск компонентов программного обеспечения.

Для проверки состояния контейнеров выполнить команду:

```
sudo docker compose ps
```

Для просмотра журналов работы компонентов выполнить команду:

```
sudo docker compose logs --tail=100
```

4.5. Проверка работоспособности ПО

Для проверки доступности Backend выполнить команду:

```
curl -k https://localhost:5001
```

При корректной работе Backend ожидается ответ:

```
Server is running
```

Для проверки доступности Frontend выполнить команду:

```
curl -k https://localhost:5002
```

Адреса, порты и протоколы подключения должны соответствовать параметрам, заданным в файле docker-compose.yaml и конфигурационных файлах приложений.

После успешного запуска контейнеров необходимо выполнить настройку Nginx.

4.6. Управление контейнерной инфраструктурой

Для остановки программного обеспечения выполнить команду:

```
sudo docker compose stop
```

Для повторного запуска:

```
sudo docker compose up -d
```

Для перезапуска контейнеров:

```
sudo docker compose restart
```

Для просмотра журналов в режиме реального времени:

```
sudo docker compose logs -f
```

Для применения изменений, внесенных в файл docker-compose.yaml, выполнить команду:

```
sudo docker compose up -d
```

5. Настройка обратного прокси-сервера Nginx

5.1. Назначение Nginx

Nginx обеспечивает доступ пользователей к программному обеспечению посредством единого доменного адреса.

Маршрутизация запросов осуществляется следующим образом:

Путь запроса	Назначение
/api	Backend, обработка API-запросов
/hub	Backend, постоянное соединение WebSocket
/	Frontend, пользовательский веб-интерфейс

Для обеспечения защищенного соединения рекомендуется использовать протокол HTTPS.

5.2. Подготовка TLS-сертификатов

Для настройки HTTPS необходимо подготовить:

- сертификат сервера;
- закрытый ключ сертификата;
- цепочку сертификатов удостоверяющего центра при необходимости.

Сертификат должен соответствовать доменному имени, используемому для доступа к ПО.

5.3. Пример конфигурации Nginx

Ниже приведен пример конфигурации Nginx с использованием условного доменного имени monitoring.example.com.

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}

server {
    server_name monitoring.example.com;

    listen 443 ssl;
    listen [::]:443 ssl;

    ssl_certificate     /etc/nginx/certs/fullchain.pem;
    ssl_certificate_key /etc/nginx/certs/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;

    client_max_body_size 5M;

    location /api {
        proxy_pass https://localhost:5001;

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
```

```
    }

    location /hub {
        proxy_pass https://localhost:5001;

        proxy_http_version 1.1;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_cache_bypass $http_upgrade;
    }

    location / {
        proxy_pass https://localhost:5002;

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

server {

    listen 80;
    listen [::]:80;

    server_name monitoring.example.com;

    return 301 https://$host$request_uri;
}
```

Данный фрагмент конфигурации необходимо разместить внутри блока http основного конфигурационного файла Nginx либо в отдельном файле, подключаемом внутри данного блока посредством директивы include.

5.4. Адаптация конфигурации

Перед применением конфигурации необходимо изменить следующие параметры:

Параметр	Назначение
monitoring.example.com	Доменное имя сервера заказчика
ssl_certificate	Путь к файлу сертификата
ssl_certificate_key	Путь к закрытому ключу сертификата
https://localhost:5001	Адрес Backend
https://localhost:5002	Адрес Frontend
client_max_body_size	Максимально допустимый размер тела запроса

При использовании Nginx на Windows необходимо указать пути к сертификатам в формате, соответствующем данной операционной системе. Например:

```
ssl_certificate     C:/nginx/certs/fullchain.pem;
ssl_certificate_key C:/nginx/certs/privkey.pem;
```

Если Backend и Frontend настроены на прием запросов по протоколу HTTP, необходимо изменить протокол в соответствующих директивах proxy_pass.

5.5. Применение конфигурации на Windows

Запустить PowerShell от имени администратора.

Перейти в директорию установки Nginx:

```
cd C:\nginx
```

Проверить корректность конфигурации:

```
.\nginx.exe -t
```

При отсутствии ошибок выполнить запуск Nginx:

```
Start-Process .\nginx.exe
```

Если Nginx уже запущен, выполнить перечитывание конфигурации:

```
.\nginx.exe -s reload
```

Для обеспечения доступности ПО после перезагрузки сервера необходимо настроить автоматический запуск Nginx.

5.6. Применение конфигурации на Linux

Проверить корректность конфигурации:

```
sudo nginx -t
```

При отсутствии ошибок выполнить запуск Nginx и включить его автоматический запуск:

```
sudo systemctl enable --now nginx
```

Применить изменения конфигурации:

```
sudo systemctl reload nginx
```

Проверить состояние Nginx:

```
sudo systemctl status nginx
```

6. Настройка разрешенных источников запросов (CORS)

Для обеспечения корректного взаимодействия Backend и Frontend необходимо проверить параметры разрешенных источников запросов.

Подключиться к базе данных PostgreSQL посредством PgAdmin или другого средства администрирования PostgreSQL.

В таблице AllowedCORS добавить разрешенный источник запросов, соответствующий внешнему адресу Frontend.

Пример:

```
https://monitoring.example.com
```

При использовании нестандартного порта необходимо также указать соответствующий порт.

После внесения изменений выполнить перезапуск Backend.

Для Windows:

```
Restart-Service AlcoInspector.Backend
```

Для Linux:

```
sudo docker compose restart
```

7. Настройка сетевого доступа

Для обеспечения доступа пользователей к программному обеспечению необходимо настроить сетевые правила и межсетевой экран сервера.

Порт	Протокол	Назначение
80	TCP	Перенаправление HTTP на HTTPS
443	TCP	Защищенный доступ к ПО
5001	TCP	Внутренний порт Backend
5002	TCP	Внутренний порт Frontend
5432	TCP	Подключение к PostgreSQL

При размещении всех компонентов на одном сервере не требуется предоставлять внешний доступ к портам Backend, Frontend и PostgreSQL.

Доступ к указанным портам рекомендуется ограничить внутренней инфраструктурой сервера.

Доменное имя ПО должно разрешаться в IP-адрес сервера, на котором размещен Nginx.

8. Проверка работоспособности программного обеспечения

После завершения установки и настройки ПО необходимо:

1. Убедиться в работоспособности PostgreSQL.
2. Проверить состояние Backend.
3. Проверить состояние Frontend.
4. Проверить состояние Nginx.
5. Открыть веб-браузер.
6. Перейти по настроенному доменному адресу ПО.
7. Убедиться в отображении страницы авторизации.
8. Выполнить авторизацию с использованием предоставленных учетных данных.
9. Проверить доступность данных Мониторингового центра.

10. Проверить отсутствие ошибок при взаимодействии Backend и Frontend.
11. Проверить установление соединения WebSocket.
12. Проверить корректность TLS-сертификата.

При успешном выполнении указанных действий установка серверной части программного обеспечения «Алко-Инспектор» считается завершенной.