

УТВЕРЖДАЮ

\_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_

## **Программное обеспечение EveryTag**

### **Руководство по установке программного обеспечения на операционную систему ALT OS**



# Содержание

Термины и определения.....	4
Аннотация.....	5
1. Системные требования.....	5
2. Установка системы.....	8
2.1. Добавление репозиторий.....	8
2.2. Установка внешних компонентов Системы.....	9
2.2.1. Установка Tesseract.....	9
2.2.2. Установка Open JDK 11.....	9
2.2.3. Установка MongoDB.....	10
2.2.4. Установка RabbitMQ.....	11
2.2.5. Установка OpenSearch.....	12
2.2.6. Установка и настройка NGINX.....	15
2.2.7. Установка и настройка Keycloak.....	20
2.3. Установка компонентов ядра Системы.....	22
2.3.1. Установка и настройка FRONT.....	22
2.3.2. Установка и настройка программного продукта ILD.....	22
2.3.3. Запуск программного обеспечения.....	23
2.4. Проверка работы сервиса.....	24

## Термины и определения

№ п/п	Термин/сокращение	Определение/расшифровка
1	CLI	Command Line Interface
2	DNS	Domain name system
3	HTTPS	HyperText Transfer Protocol Secure (расширение протокола HTTP, поддерживающее шифрование)
4	ILD	Information leaks detection
5	JDK	Java development kit
6	JSON	JavaScript Object Notation
7	OCR	Optical character recognition
8	SSL	Secure sockets layer
9	SSO	Single sign-on
10	SQL	Structured query language
11	URL	Uniform resource locator
12	ПО	Программное обеспечение
13	Система	ПО EveryTag
14	СУБД	Система управления базами данных

## Аннотация

В данном документе приведено описание процесса установки компонентов программного обеспечения EveryTag (далее – ПО EveryTag или Системы).

После разработки и передачи данного документа, Система может быть усовершенствована, поэтому возможны некоторые не являющиеся критичными несоответствия, касающиеся описания архитектуры системы.

### 1. Системные требования

Для работы Системы, в случае развертывания решения на одном сервере, необходимо аппаратное и программное обеспечение, соответствующее следующим требованиям:

Минимальные требования к аппаратному обеспечению:

- Процессор с тактовой частотой не ниже 2.67 ГГц.
- Количество ядер процессора: 12.
- Оперативная память 32 ГБ.
- Дисковое пространство 100 ГБ.

Рекомендуется размещать компоненты Системы на четырех серверах. Такая реализация подразумевает размещение на четырех отдельных виртуальных или физических серверах следующих компонентов: http-сервера Nginx, системных компонентов, компонентов ядра ПО EveryTag и базы данных с файловым хранилищем.

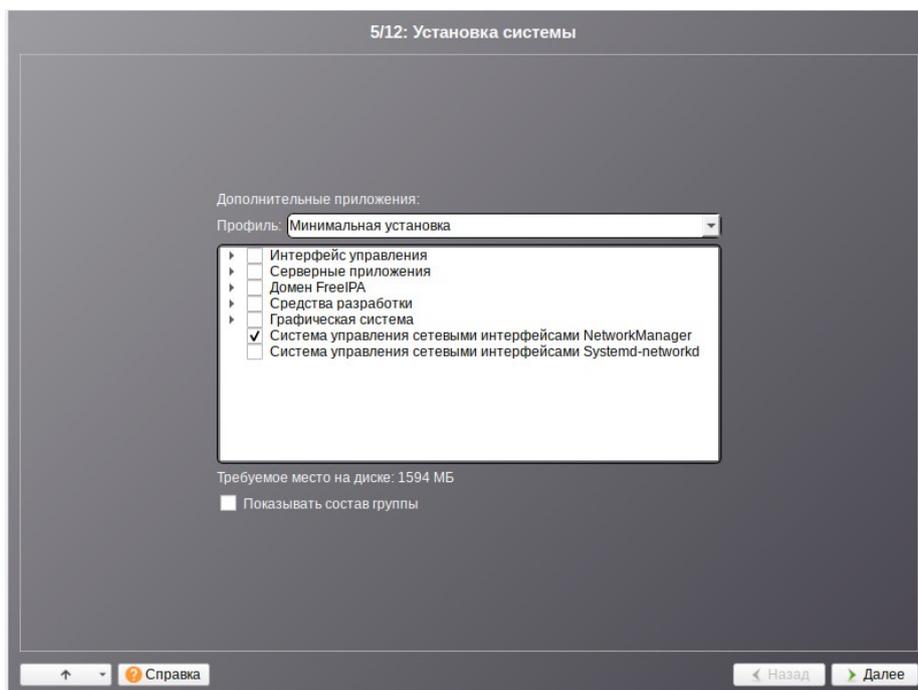
Данная схема размещения обеспечивает оптимальное распределение нагрузки между http-сервером, системными компонентами и компонентами ядра Системы в целях повышения производительности и быстродействия, а также минимизацию необходимого объема памяти для хранения резервных копий виртуальных машин. Выделение отдельного сервера для базы данных обусловлено большим объемом информации и высокой частотой ее резервного копирования по сравнению с прочими компонентами Системы.

Рекомендуемые системные требования для серверов рассчитываются на основании модели нагрузки, исходя из количества пользователей, параметров документооборота и других показателей.

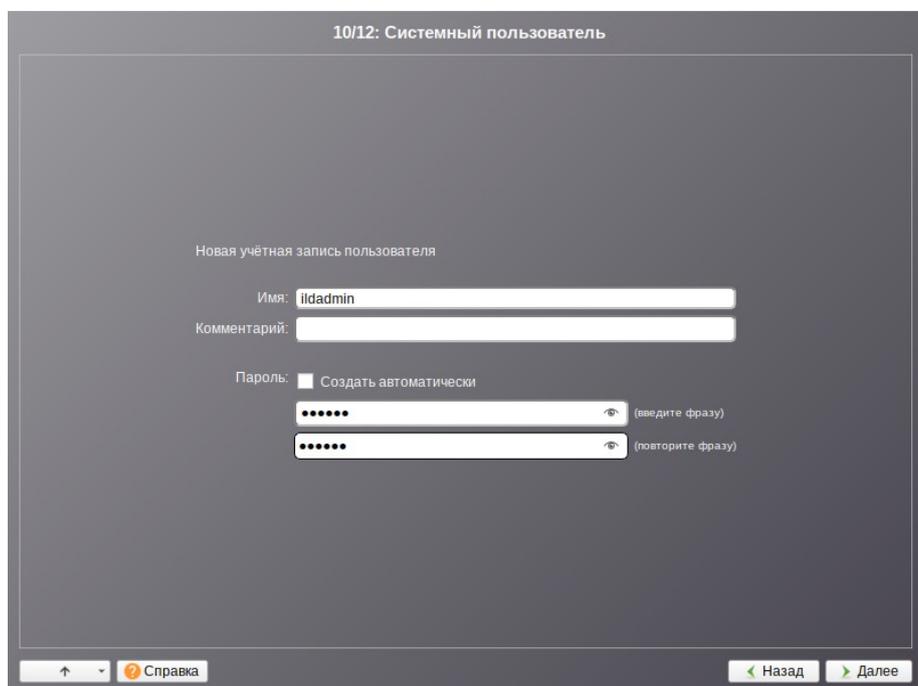
Необходимые и достаточные требования к программному обеспечению на серверах:

- Операционная система ALT Server 10.0 (FalcoRusticolus);

- При установке ОС на шаге "5/12: Установка системы", необходимо выбрать "Минимальная установка" и "Система управления сетевыми интерфейсами NetworkManager".



- При установке ОС на шаге "10/12: Системный пользователь", необходимо в поле "Имя" указать имя пользователя, от которого будет производиться запуск программного продукта, "ildadmin".



Необходимое и достаточное требование к программному обеспечению на рабочих станциях:

- Предварительно на рабочих станциях пользователей должны быть отключены блокировщики контента в браузере (AdBlock, Kaspersky Security, и т.п.) или адрес сервера с ПО EveryTag должен быть добавлен в исключения блокировщика.

## 2. Установка системы

В данном разделе описана процедура установки ПО EveryTag на операционную систему ALT Server 10.0. Для установки и настройки ПО EveryTag администратору необходимо иметь в наличии:

- Архив с ядром ПО EveryTag, содержащий необходимые модули и конфигурационные файлы.
- Руководство по установке программного обеспечения (настоящий документ).

### 2.1. Добавление репозиториев

Перейдите в режим суперпользователя:

```
su -
```

В первую очередь удалите все доступные репозитории, выполнив команду:

```
apt-repo rm all
```

Затем добавьте следующий репозиторий [ftp.altlinux.org](http://ftp.altlinux.org):

```
tee /etc/apt/sources.list.d/alt.list <<EOF
rpm http://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/x86_64 classic
rpm http://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/noarch classic
rpm http://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/x86_64-i586
classic
EOF
```

После чего обновите список репозиториев:

```
apt-get update
```

## 2.2. Установка внешних компонентов Системы

Перейдите в режим суперпользователя

```
su -
```

Внешние компоненты Системы устанавливаются для обеспечения работоспособности функционала Системы и организации взаимодействия всех компонентов. Перед тем, как приступить к установке компонентов, необходимо установить JSON-процессор командной строки, выполнив команду:

```
apt-get install jq curl htop wget net-tools -y
```

### 2.2.1. Установка Tesseract

Компонент Tesseract — сервис, предназначенный для индексирования и обеспечения полнотекстового поиска в документах.

Перейдите в режим суперпользователя:

```
su -
```

Для установки Tesseract необходимо ввести следующую команду:

```
apt-get install tesseract tesseract-devel -y
```

Для проверки установленной версии введите команду:

```
tesseract -v
```

### 2.2.2. Установка Open JDK 11

Java Development Kit (сокращенно JDK) — система разработки приложений на языке Java, включающая в себя компилятор, стандартные библиотеки классов, исполнительную систему Java - Java Runtime Environment (JRE), набор утилит и документацию. Пакет предназначен для исполнения инструкций сервисов Системы и других ее приложений, написанных на языке Java.

Перейдите в режим суперпользователя:

```
su -
```

Для установки JDK необходимо ввести следующую команду:

```
apt-get install java-11-openjdk -y
```

Для проверки установленной версии введите команду:

```
java -version
```

### 2.2.3. Установка MongoDB

MongoDB — СУБД, используемая в составе платформы Системы для хранения иерархических структур данных (документов). Данная СУБД реализована с помощью подхода NoSQL, не требующего описания схемы таблиц.

Перейдите в режим суперпользователя:

```
su -
```

Установка MongoDB осуществляется с помощью следующей команды:

```
apt-get install mongo mongo-server-mongod -y
```

Разрешите автоматический запуск сервиса при старте системы:

```
systemctl enable mongod
```

Запустите сервис:

```
systemctl start mongod
```

Список команд для управления сервисом MongoDB:

- запуск сервиса

```
systemctl start mongod
```

- остановка сервиса

```
systemctl stop mongod
```

- проверка статуса сервиса

```
systemctl status mongod
```

Для проверки установленной версии введите команду:

```
mongod -version | grep "db version"
```

## 2.2.4. Установка RabbitMQ

RabbitMQ — это приложение для управления очередями сообщений, реализующее гарантированный обмен сообщениями между произвольными процессами, приложениями и/или серверами. Причём, каждый сервер, приложение или процесс могут выступать одновременно и отправителем, и получателем сообщений от других серверов, приложений и процессов.

Перейдите в режим суперпользователя:

```
su -
```

Установите программные пакеты:

```
apt-get install rabbitmq-server -y
```

Установите плагины:

```
rabbitmq-plugins enable rabbitmq_management  
rabbitmq-plugins enable rabbitmq_amqp1_0
```

Создайте сервис:

```
tee /etc/systemd/system/rabbitmq-server.service <<EOF  
[Unit]  
Description=RabbitMQ broker  
After=network.target epmd@0.0.0.0.socket  
Wants=network.target epmd@0.0.0.0.socket  
  
[Service]  
Type=notify  
User=rabbitmq  
Group=rabbitmq  
UMask=0027  
NotifyAccess=all  
TimeoutStartSec=600  
LimitNOFILE=32768  
Restart=on-failure  
RestartSec=10  
WorkingDirectory=/var/lib/rabbitmq  
ExecStart=/usr/lib/rabbitmq/rabbitmq-server
```

```
ExecStop=/usr/lib/rabbitmq/rabbitmqctl shutdown
SuccessExitStatus=69
```

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

```
systemctl daemon-reload
```

Запустите сервис:

```
systemctl start rabbitmq-server
```

Разрешите автоматический запуск сервиса при старте системы:

```
systemctl enable rabbitmq-server
```

Список команд для управления сервисом RabbitMQ:

- запуск сервиса

```
systemctl start rabbitmq-server
```

- остановка сервиса

```
systemctl stop rabbitmq-server
```

- проверка статуса сервиса

```
systemctl status rabbitmq-server
```

Для проверки установленной версии введите команду:

```
rabbitmqctl status | grep "RabbitMQ version:"
```

## 2.2.5. Установка OpenSearch

Перейдите в режим суперпользователя:

```
su -
```

Установку OpenSearch необходимо производить с официального сайта <https://opensearch.org/downloads.html>.

Скачайте архив tar.gz версии 1.2.4.

```
wget https://artifacts.opensearch.org/releases/bundle/opensearch/1.2.4/opensearch-1.2.4-linux-x64.tar.gz
```

Создайте пользователя opensearch и установите пароль:

```
adduser opensearch  
passwd opensearch
```

Добавьте права на выполнение для скаченного архива:

```
chmod +x opensearch-1.2.4-linux-x64.tar.gz
```

Распакуйте архив, используя команду:

```
tar -xf opensearch-1.2.4-linux-x64.tar.gz
```

Создайте рабочий каталог «/opt/opensearch» для OpenSearch:

```
mkdir /opt/opensearch
```

Перенесите распакованные данные в рабочий каталог:

```
mv ./opensearch-1.2.4/* /opt/opensearch
```

Удалите каталог, оставшийся от распаковки:

```
rmdir ./opensearch-1.2.4
```

Сделайте пользователя opensearch владельцем рабочего каталога OpenSearch:

```
chown -R opensearch:opensearch /opt/opensearch
```

Запустите установочный скрипт от имени пользователя opensearch:

```
su opensearch  
/opt/opensearch/opensearch-tar-install.sh
```

Дождитесь сообщения «Node '<имя сервера>' initialized», нажмите «Ctrl+C» и выйдите из-под учетной записи пользователя opensearch.

Создайте сервис для работы OpenSearch:

```
tee /etc/systemd/system/opensearch.service <<EOF  
[Unit]  
Description=Opensearch  
Documentation=https://opensearch.org/docs/latest  
Wants=network-online.target  
After=network-online.target  
[Service]  
Type=simple  
RuntimeDirectory=opensearch
```

```
PrivateTmp=true
Restart=on-failure
RestartSec=60s
WorkingDirectory=/opt/opensearch
User=opensearch
Group=opensearch
ExecStart=/opt/opensearch/bin/opensearch
StandardOutput=journal
StandardError=inherit
# Specifies the maximum file descriptor number that can be opened by this process
LimitNOFILE=65535
# Specifies the maximum number of processes
LimitNPROC=4096
# Specifies the maximum size of virtual memory
LimitAS=infinity
# Specifies the maximum file size
LimitFSIZE=infinity
# Not use SWAP
LimitMEMLOCK=infinity
# Disable timeout logic and wait until process is stopped
TimeoutStopSec=0
# Allow a slow startup before the systemd notifier module kicks in to extend the
timeout
TimeoutStartSec=75
[Install]
WantedBy=multi-user.target
EOF

chown -R root:root /etc/systemd/system/opensearch.service
systemctl enable opensearch.service
systemctl daemon-reload
```

### Настройте конфигурационный файл:

```
mv /opt/opensearch/config/opensearch.yml /opt/opensearch/config/opensearch-
backup.yml

tee /opt/opensearch/config/opensearch.yml <<EOF
network.host: 0.0.0.0
http.port: 9200
cluster.name: docker-cluster
plugins.security.disabled: true
discovery.type: single-node
EOF
```

Перезапустите сервис:

```
systemctl restart opensearch.service
```

Список команд для управления сервисом OpenSearch:

- запуск сервиса

```
systemctl start opensearch.service
```

- остановка сервиса

```
systemctl stop opensearch.service
```

- проверка статуса сервиса

```
systemctl status opensearch.service
```

Для проверки установленной версии введите команду:

```
/opt/opensearch/bin/opensearch -V
```

## 2.2.6. Установка и настройка NGINX

Перейдите в режим суперпользователя:

```
su -
```

Чтобы установить NGINX, выполните следующие команды:

```
apt-get install nginx -y
```

Создайте директорию для конфигурационных файлов:

```
mkdir /etc/nginx/conf.d  
sed -i 's|sites-enabled|conf|g' /etc/nginx/nginx.conf
```

Создайте конфигурационный файл:

```
tee /etc/nginx/conf.d/vdr.conf <<EOF  
upstream file-service {  
    server 127.0.0.1:17082;  
}  
upstream document-service {  
    server 127.0.0.1:17081;  
}  
upstream investigation-service {
```

```

    server 127.0.0.1:17085;
}
upstream micro.ppdf {
    server 127.0.0.1:19002;
}
upstream micro.word {
    server 127.0.0.1:19004;
}
upstream micro.image {
    server 127.0.0.1:19010;
}
upstream micro.notify {
    server 127.0.0.1:19003;
}
upstream micro.marking {
    server 127.0.0.1:19008;
}
upstream micro.workspaces {
    server 127.0.0.1:19011;
}

server {
    listen 80;
    server_name altlinux.everytag.ru;
    return 301 https://\${host}\${request_uri};
    client_max_body_size 20G;
}

server {
    listen 443 ssl http2;
    server_name altlinux.everytag.ru;

    client_max_body_size 20G;
    proxy_temp_file_write_size 64k;
    proxy_connect_timeout 10080s;
    proxy_send_timeout 10080;
    proxy_read_timeout 10080;
    proxy_buffer_size 64k;
    proxy_buffers 16 32k;
    proxy_busy_buffers_size 64k;
    proxy_redirect off;
    proxy_request_buffering off;
    proxy_buffering off;
}

```

```

ssl_certificate /etc/nginx/cert/altlinux.everytag.ru.cer;
ssl_certificate_key /etc/nginx/cert/altlinux.everytag.ru.key;
ssl_prefer_server_ciphers On;
ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-
RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-
DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-
ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-
SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-
AES256-SHA:DHE-RSA-AES256-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!3DES:!MD5:!PSK;
add_header Strict-Transport-Security max-age=15768000;

root /var/www/front.ild;

index index.html;

access_log /var/log/nginx/front.access.log;
error_log /var/log/nginx/front.error.log;

charset utf-8;
source_charset utf-8;

charset_types text/plain text/css application/json application/javascript
text/xml application/xml application/xml+rss text/javascript application/font-woff;
error_page 404 401 403 500 502 /index.html;

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_headers_hash_max_size 512;
proxy_headers_hash_bucket_size 128;

proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade";

location / {
    try_files $uri $uri/ /index.html;
}

location /ndas {
    root /var/www/ndas;

```

```
}

location /auth {
    access_log /var/log/nginx/auth.access.log;
    error_log /var/log/nginx/auth.error.log;
    proxy_pass https://127.0.0.1:8543;
}

location /files {
    proxy_pass http://file-service;
}

location /api/v1 {
    proxy_pass http://document-service;
}

location /api/v1/files {
    rewrite ^/api/v1/files(.*)$ /files/$1;
}

location /api/v1/investigations {
    proxy_pass http://investigation-service;
}

location /api/v1/calibration {
    proxy_pass http://investigation-service;
}

location /api/v1/ppdf {
    proxy_pass http://micro.ppdf;
}

location /api/v1/markings {
    proxy_pass http://micro.marking;
}

location /api/v1/word {
    proxy_pass http://micro.word;
}

location /api/v1/confidential {
    proxy_pass http://micro.workspaces;
}
```

```

location /api/v1/workspace/ {
    proxy_pass http://micro.workspaces;
}

location /api/v1/raster {
    proxy_pass http://micro.image;
}

location /microws {
    rewrite /microws(.*) /api/v1/websockets/\$1;
}

location /api/v1/websockets {
    proxy_http_version 1.1;
    proxy_set_header Upgrade \$http_upgrade;
    proxy_set_header Connection "Upgrade";
    proxy_set_header X-real-ip \$remote_addr;
    proxy_set_header X-Forwarded-For \$remote_addr;
    proxy_pass http://micro.notify;
}

location /api/v1/version {
}

location /api/v2 {
    proxy_pass http://document-service;
}

location /amplitude {
    return 200 "success";
}
}
EOF

```

Скопируйте сертификаты, предоставленные компанией EveryTag:

```

mkdir /etc/nginx/cert
cp /home/ildadmin/<путь до папки>/nginx/cert/altlinux.everytag.ru.*
/etc/nginx/cert/

```

Запустите NGINX:

```
systemctl restart nginx
```

Для проверки установленной версии введите команду:

```
nginx -v
```

## 2.2.7. Установка и настройка Keycloak

Keycloak — это решение для идентификации и управления доступом пользователей Системы.

Keycloak обеспечивает:

- регистрацию пользователей;
- аутентификацию пользователей;
- единый вход (SSO);
- клиентские адаптеры;
- интерфейс для администратора и управления учетными записями.

Перейдите в режим суперпользователя:

```
su -
```

Для установки Keycloak распакуйте архив `keycloak_altlinux.tar.gz`, предоставленный компанией EveryTag:

```
tar -xf /home/ildadmin/<путь до папки>/arch/keycloak_altlinux.tar.gz -C /opt/
```

Создайте сервис:

```
tee /etc/systemd/system/keycloak.service <<EOF
[Unit]
Description=Keycloak service
[Service]
Type=simple
WorkingDirectory=/opt/keycloak/
ExecStart=/opt/keycloak/bin/standalone.sh -Djboss.socket.binding.port-offset=100 -b
0.0.0.0 -Darchaius.configurationSource.additionalUrls=file:///opt/keycloak/
standalone/deployments/eureka-client.properties
[Install]
WantedBy = multi-user.target
EOF

systemctl daemon-reload
```

Перезапустите сервис:

```
systemctl restart keycloak.service
```

Список команд для управления сервисом Keycloak:

- запуск сервиса

```
systemctl start keycloak.service
```

- остановка сервиса

```
systemctl stop keycloak.service
```

- проверка статуса сервиса

```
systemctl status keycloak.service
```

## 2.3. Установка компонентов ядра Системы

Внешние компоненты Системы устанавливаются для обеспечения работоспособности функционала Системы и организации взаимодействия всех компонентов. Перед тем, как приступить к установке компонентов, необходимо установить JSON-процессор командной строки, выполнив команду

### 2.3.1. Установка и настройка FRONT

Перейдите в режим суперпользователя:

```
su -
```

Для установки Front распакуйте архив `front_altlinux.tar.gz`, предоставленный компанией EveryTag:

```
tar -xf /home/ildadmin/<путь до папки>/arch/front_altlinux.tar.gz -C /var/
```

### 2.3.2. Установка и настройка программного продукта ILD

Перейдите в режим суперпользователя:

```
su -
```

Для установки ILD распакуйте архив `ild_altlinux.tar.gz`, предоставленный компанией EveryTag:

```
tar -xf /home/ildadmin/<путь до папки>/arch/ild_altlinux.tar.gz -C /opt/
```

Скопируйте лицензионный файл, предоставленный компанией EveryTag:

```
cp /home/ildadmin/<путь до папки>/license.lic /opt/ild/license
```

Измените владельца для папки `ild`:

```
chown -R ildadmin:ildadmin /opt/ild
```

Для установки MICRO-SERVICE распакуйте архив `front_altlinux.tar.gz`, предоставленные компанией EveryTag:

```
tar -xf /home/ildadmin/<путь до папки>/arch/micro_altlinux.tar.gz -C /opt/
```

Измените владельца для папки `micro`:

```
chown -R ildadmin:ildadmin /opt/micro
```

Добавьте запись в hosts:

```
tee -a /etc/hosts <<EOF  
127.0.0.1 altlinux.everytag.ru  
EOF
```

Скопируйте:

```
cp /home/ildadmin/<путь до папки>/service/*.service /etc/systemd/user
```

Отключите использование системой ipv6:

```
tee -a /etc/sysctl.conf <<EOF  
net.ipv6.conf.all.disable_ipv6=1  
net.ipv6.conf.default.disable_ipv6=1  
net.ipv6.conf.lo.disable_ipv6=1  
EOF  
  
sysctl -p
```

Выполните первоначальную настройку коллекции Mongo:

```
chmod +x /home/ildadmin/<путь до папки>/postinstall.sh  
/home/ildadmin/<путь до папки>/postinstall.sh
```

### 2.3.3. Запуск программного обеспечения

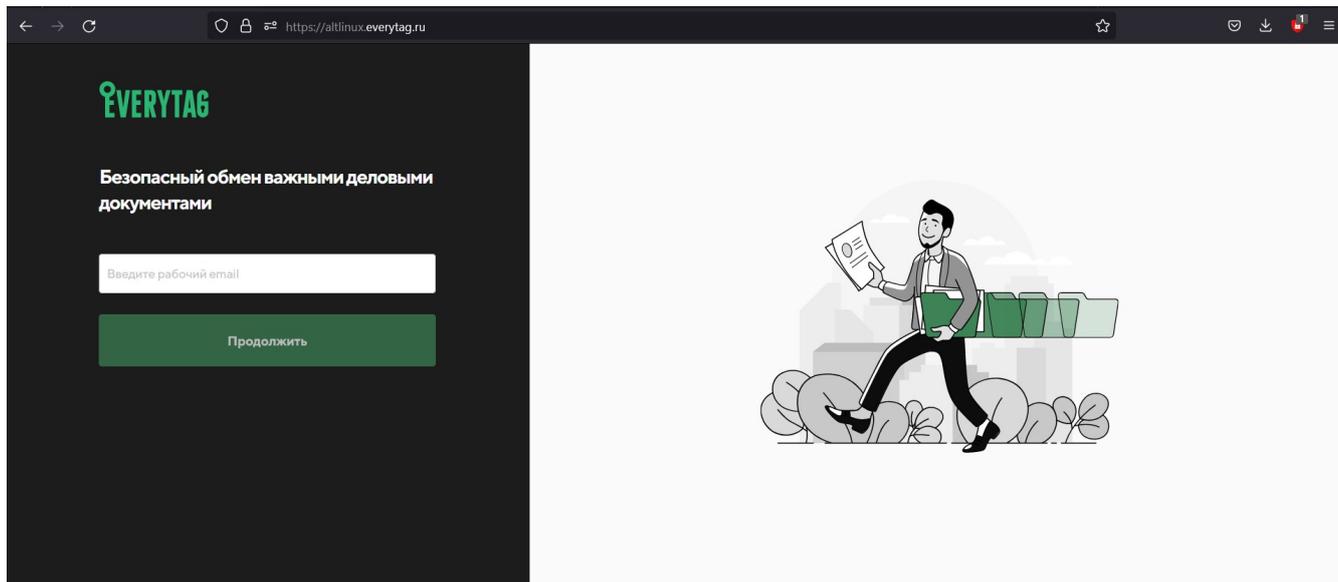
Все операции по запуску программного обеспечения ILD выполняется из-под учетной записи пользователя **ildadmin**

Во избежание неправильной последовательности запуска программных модулей воспользуйтесь скриптом:

```
chmod +x /home/ildadmin/<путь до папки>/START_ILD.sh  
/home/ildadmin/<путь до папки>/START_ILD.sh
```

## 2.4. Проверка работы сервиса

Перейдите по адресу <https://altlinux.everytag.ru>



При необходимости потребуется внести запись в hosts в формате:

```
<ip-адрес сервера с установленным ILD> altlinux.everytag.ru
```

- путь до файла на машинах с ОС Windows C:\Windows\System32\drivers\etc\hosts
- путь до файла на машинах с ALT OS /etc/hosts.

