Платформа

ELIS

QUICK START. Быстрая установка платформы

Версия: 1.36

Содержание

1 Минимальные системные требования 3
2 Установка платформы ELIS 4
3 Первый вход на платформу ELIS 18

1 Минимальные системные требования

Платформа ELIS (Eltex IoT System) разработана для организации единой системы управления, конфигурирования и мониторинга устройств Интернета вещей (датчиков, Wi-Fi реле, камер и т. п.). Использование данной системы с графическим веб-интерфейсом и мобильным приложением позволяет:

- управлять учетными записями пользователей (добавление, изменение, удаление);
- осуществлять мониторинг состояния устройств умного дома;
- получать уведомления о событиях;
- создавать сценарии работы устройств в доме.

Минимальные системные требования сервера (минимально необходимая конфигурация для запуска ядра платформы с микросервисами):

- число аппаратных серверов — 1;
- процессор — i5 3,0 ГГц с поддержкой AVX;
- оперативная память — 8 ГБ;
- место на диске — 1000 ГБ;
- производительность дискового массива (чтение/запись) — 2000 IOPS.

2 Установка платформы ELIS

В инструкции приведена установка платформы ELIS версии 1.36 на операционную систему Ubuntu 20.04. Информацию по установке более ранних версий платформы ELIS (1.35 и ниже) можно найти по ссылке: Архив. ELIS.

⚠ С версии 1.19.4 установка через deb-пакеты не осуществляется.

❌ Установка ELIS на сервер, не отвечающий минимальным системным требованиям, может привести к неработоспособности платформы или увеличить время обработки запросов.

Перед развертыванием платформы необходимо установить Ansible и необходимые для ее работы компоненты. Ansible рекомендуется устанавливать из официального репозитория проекта.

✅ Ansible — система управления конфигурациями, написанная на языке программирования Python с использованием декларативного языка разметки для описания конфигураций. Система используется для автоматизации настройки и развертывания программного обеспечения, в частности для ПО ELIS.

Для установки платформы выполните следующие шаги:

1. Установите Ansible на сервер Ubuntu 20.04.

Пример установки через консоль:

Установка Ansible

```
apt update
apt install --install-recommends linux-generic-hwe-20.04-edge
apt install software-properties-common
add-apt-repository --yes --update ppa:ansible/ansible
apt install ansible
```

✅ Более подробная информация по установке Ansible доступна по ссылке.

2. Выполните проверку версии (должна быть не ниже **v2.9**):

Проверка версии Ansible

```
ansible --version
```

3. После установки Ansible добавьте необходимые для ее работы коллекции.

Пример добавления коллекций:

Установка

```
ansible-galaxy collection install community.general
ansible-galaxy collection install community.crypto
ansible-galaxy collection install community.docker
```

4. Подготовьте конфигурацию.

Для получения файлов конфигурации обратитесь с запросом в Коммерческий отдел Элтекс.
Файлы конфигурации будут направлены вам в виде архива tar.gz, который необходимо распаковать в директорию **/etc** с правами **root**.

Пример распаковки архива:

Распаковка архива с конфигурацией

```
tar -C /etc -xvf ansible-iot-1.36.tar.gz
```

После распаковки архива все пакеты и зависимости будут развернуты в директории на текущем сервере.

Файлы конфигурации и плейбуки (скрипты/конфигурации) Ansible будут расположены в директории **/etc/ansible-iot-1.36**.

5. Отредактируйте файл **/etc/ansible-iot-1.36/inventory**.

Откройте файл в любом доступном текстовом редакторе, например **nano**. Укажите пароль пользователя **root** в переменной **ansible_sudo_pass**:

⚠ Далее в примере для пользователя **root** используется пароль **rootpasswd**.
При установке задайте свой пароль.

Пример задания пароля:

Содержимое файла inventory

```
[iot]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
[mqtt_broker]
localhost   ansible_connection=local    ansible_sudo_pass=rootpassword
[elk]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
[monitoring]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
```

6. Далее необходимо настроить параметры доступа к платформе ELIS.

MongoDB версии 5 и выше работает только на процессорах с поддержкой AVX. Узнать, поддерживает ли ваш процессор AVX, можно с помощью команды:
lscpu | grep avx
Если ответ оказался пустым, ваш процессор не поддерживает AVX. Используйте MongoDB версии 4.
Если в ответе вернулся список флагов, можно использовать MongoDB версии 5 и выше.

Для базовой установки достаточно отредактировать файл конфигурации **/etc/ansible-iot-1.36/vars/default.yml**.

Откройте файл в любом доступном текстовом редакторе, например **nano**. Укажите корректный **IP-адрес** или **доменное имя** для доступа к платформе в переменной **iot.serverName:**

При переходе с MongoDB 4 на MongoDB 6 требуется сначала перейти на MongoDB 5 и только потом перейти на MongoDB 6.
Или в файле /vars/default.yml для параметра version задать значение «5», запустить ansible-playbook install_iot.yml, затем задать значение «6» и снова запустить ansible-playbook install_iot.yml

Содержимое конфигурационного файла vars/default.yml

```
---
# Параметры установки платформы.
iot:
  # Имя (IP-адрес) сервера, на котором будет производиться развертывание платформы IoT.
  #  ВАЖНО!!! В 'serverName' нужно прописывать то имя (IP-адрес), по которому будет доступны платформа.
  serverName: "external.iot.address"
  # Содержит путь до директории, в которую будет произведена установка.
  installDir: /storage/iot
  # Ссылка на политику конфиденциальности
  privacyPolicyUrl: "my.privacy.policy"
  # Ссылка на базу знаний
  knowledgeBaseUrl: "https://docs.eltex-co.ru/display/EKB/IoT"

  # Параметры MongoDB.
  mongodb:
    # Версия MongoDB. На старом железе, не поддерживающем оптимизацию, нужно выставить значение `4`.
    version: 6
    external:
      # Если выставлен в true, будет использоваться внешняя MongoDB.
      # ВАЖНО!!! MongoDB должна быть настроена, а параметры подключения нужно указать в 'uri'.
      enable: false
      # URI внешней MongoDB.
      uri: "mongodb://external.mongodb.address:27017"

  # Параметры WEB.
  web:
    # Порт HTTP, по которому будет осуществляться доступ в WEB.
    httpPort: 80
    # Порт HTTPS, по которому будет осуществляться доступ в WEB.
    httpsPort: 443
    # Автоматически перенаправлять запросы по порту HTTP на порт HTTPS
    redirectHttpToHttps: true
    # Нужно ли использовать HTTPS при формировании ссылок к WEB ('true' по умолчанию, при этом будет использован порт,
    # указанный в 'iot.web.httpsPort'). Если поставить в 'false', будет использован HTTP и порт,
    # указанный в 'iot.web.httpPort'.
    useHttpsByDefault: true
    nginx:
      # Максимальное число соединений, которые одновременно может открыть рабочий процесс
      worker_connections: 2048
      # Ограничение скорости обработки запросов модулем Nginx Rate Limiting
      rateLimit:
        enable: true
    certbot:
      # Использовать ли certbot для получения сертификатов Let's Encrypt.
      enable: false
      # Email владельца домена. Необходим для подтверждения валидности домена при получении сертификата Let's Encrypt.
      email: test@email.com
    fail2ban:
```

```
      enable: true

  # Параметры сервера отправки email.
  mail:
    smtp:
      submitter: test@email.com
      password: "password"
      senderPrefix: "Сервер Eltex-SC"
      auth: "true"
      host: email.com
      port: 587
      # Протокол шифрования, используемый при подключении к серверу. Допустимые значения: none, starttls, ssl.
      protection: starttls
  push:
    firebase:
      enabled: false
    apns:
      enabled: false
    hms:
      enabled: false


  # Параметры установки сервиса платежей для тарифов.
  payment:
    logLevel: INFO

  # Параметры authorization server.
  authorization:
    # Уровень отладки внутри IoT Authorization Server.
    logLevel: INFO

    # Уровень сложности капчи: easy, medium, hard
    captchaLevel: "easy"

    # Параметры для управления доступностью саморегистрации.
    selfRegistration:
      allow: true
      allowDemo: true
      allowSocialNetworks: false

    # Параметры клиентских регистраций (через соцсети).
    clientRegistrations:
      google:
        enable: true
        clientId: "GoogleClientIdChangeMe"
        clientSecret: "GoogleClientSecretChangeMe"
      microsoft:
        enable: true
        clientId: "MicrosoftClientIdChangeMe"
        clientSecret: "MicrosoftClientSecretChangeMe"
      apple:
        enable: true
        clientId: "AppleClientIdChangeMe"
        keyId: "AppleKeyIdChangeMe"
        teamId: "AppleTeamIdChangeMe"
      yandex:
        enable: true
        clientId: "YandexClientIdChangeMe"
        clientSecret: "YandexClientSecretChangeMe"
```

```
      vk:
        enable: true
        clientId: "VkClientIdChangeMe"
        clientSecret: "VkClientSecretChangeMe"
      mailRu:
        enable: true
        clientId: "MailRuClientIdChangeMe"
        clientSecret: "MailRuClientSecretChangeMe"

    skills:
      # Параметры навыка Яндекс для интеграции с Умным домом (Алисой). Отображается в карточке навыка.
      yandex:
        enabled: false
        # Параметры для Basic Authentication.
        clientId: "YandexClientIdChangeMe"
        password: "PasswordChangeMe"
        # Id навыка, который необходимо указывать при отправке уведомлений.
        skillId: ""
        # OAuth-токен, который необходимо указывать при отправке уведомлений.
        oauthToken: ""

      # Параметры проекта умного дома Сбера для интеграции с Салютом. Отображается в карточке проекта.
      sber:
        enabled: false
        # Параметры для Basic Authentication.
        clientId: "SberClientIdChangeMe"
        password: "PasswordChangeMe"
        # Bearer-токен, который необходимо указывать при отправке уведомлений.
        bearerToken: ""

      # Параметры проекта умного дома Mail.ru для интеграции с Марусей. Отображается в карточке проекта/приложения.
      marusya:
        enabled: false
        # Параметры для Basic Authentication.
        clientId: "MarusyaClientIdChangeMe"
        password: "PasswordChangeMe"
        # App ID, который был назначен приложению VK при создании.
        appId: "MarusyaAppIdChangeMe"
        # OAuth-токен, который необходимо указывать при отправке уведомлений.
        oauthToken: ""

  # Параметры платформы IoT core.
  core:
    # Уровень отладки внутри IoT Core.
    logLevel: INFO

    # Порты платформы для подключения zway-контроллеров.
    ctlGate:
      port: 8070
      tcpPort: 8069
      sslPort: 8072

    links:
      # Нужно ли использовать HTTPS при формировании ссылок к ресурсам самой платформы (например, прошивки).
      useHttpsForApi: false
```

```
      # Нужно ли использовать HTTPS при формировании ссылок на фото с камер наблюдения.
      useHttpsForCameraLinks: true
      # Нужно ли использовать 'iot.web.httpPort'/'iot.web.httpsPort' при формировании ссылок к API.
      useUiProxyForApi: true

    # Параметры для работы с видеосерверами
    video:
      # Параметры Flussonic.
      flussonic:
        url: ""
        apiKey: ""
        operatorId: ""
        adminLogin: ""
        motion:
          enabled: false
      # Параметры видеосервера EVI
      evi:
        url: ""
        apiKey: ""
        operatorId: ""
        adminLogin: ""

    # Настройки ИК-пульта.
    irc:
      # Время ожидания ИК команды от пользователя
      recTimeout: 15s
      # Таймаут записи команды (отсутствия фронтов)
      cmdTimeout: 100ms

    # Настройки охраны.
    guard:
      # Время, которое дается на включение FLIRS устройств (ждем подтверждение от контроллера),
      # при постановке на охрану.
      deviceRequestDelay: 15s
      # Время, которое прибавляется ко времени задержки на очистку охранного кэша при постановке/снятии с охраны,
      # на случай если охрана не завершила процесс постановки/снятия.
      # Время задержки формируется как количество охранных устройств умноженное на deviceRequestDelay.
      clearContextExtraCacheDelay: 1m

# Параметры установки сервисов логирования (Elasticsearch + Logstash + Kibana).
elk:
  # Нужно ли добавлять в платформу appender, отправляющий логи в logstash.
  # В нем нет необходимости, если ELK не развернут или не настроен; это лишь спровоцирует сообщения об ошибках отправки
  # в логах платформы.
  enable: false
  # Имя (IP-адрес) сервера, на котором будет развернут ELK.
  # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с платформой (на том же хосте).
  # В таком случае хосты в инвентаре в группах [iot] и [elk] должны совпадать.
  serverName: "{{ iot.serverName }}"
  # Директория для установки системы логирования.
  installDir: /storage/elk

slgate:
```

```
216   # Имя сервера, на котором будет развернут SLGATE.
217   # ВАЖНО!!! Нельзя использовать IP-адрес, т.к. такая схема не будет работать!
218   # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
      платформой (на том же хосте).
219   # В таком случае хосты в инвентаре в группах [iot] и [slgate] должны совпадать.
220   serverName: "{{ iot.serverName }}"
221   # Директория для установки SLGATE.
222   installDir: /storage/slgate
```

Таблица описания значений в файле настроек /vars/default.yml

iot:	Параметры установки платформы.
serverName: "external.iot.address"	Имя (IP-адрес) сервера, на котором будет производиться развертывание платформы IoT. Возможно использование 'localhost', если все манипуляции производятся локально. ⚠ В 'serverName' нужно прописывать то имя (IP-адрес), по которому будет доступна платформа. Если указать 'localhost', то платформа будет доступна только через 'localhost'.
installDir: /storage/iot	Путь до директории, в которую будет произведена установка.
privacyPolicyUrl: "my.privacy.policy"	Ссылка на политику конфиденциальности.
knowledgeBaseUrl: "https://docs.eltex-co.ru/display/EKB/IoT"	Ссылка на базу знаний.
mongodb:	Параметры MongoDB.
version: 6	Версия MongoDB.
external	
enable: false	Параметр для использования внешней MongoDB. Если выставлен в true, будет использоваться внешняя MongoDB. ⚠ MongoDB должна быть настроена, а параметры подключения — указаны в 'uri'.
uri: "mongodb://external.mongodb.address:27017"	Адрес и порт внешней MongoDB.
web:	Параметры WEB.
httpPort: 80	Порт HTTP, по которому будет осуществляться доступ в WEB.
httpsPort: 443	Порт HTTPS, по которому будет осуществляться доступ в WEB.

redirectHttpToHttps: true	Параметр для перенаправления HTTP-запросов на HTTPS.
useHttpsByDefault: true	
nginx:	
worker_connections: 2048	Максимальное число соединений, которое одновременно может открыть рабочий процесс.
rateLimit:	
enable: true	Параметр, позволяющий включать/отключать ограничение скорости обработки запросов модулем Nginx Rate Limiting.
certbot:	
enable: false	Параметр, позволяющий использовать certbot для получения сертификатов Let's Encrypt.
email: test@email.com	Email владельца домена. Необходим для подтверждения валидности домена при получении сертификата Let's Encrypt.
mail:	Параметры сервера отправки email.
smtp:	
submitter: test@email.com	Учетная запись email.
password: "password"	Пароль от учетной записи email.
auth: "true"	Проверка подлинности SMTP (включена по умолчанию).
senderPrefix: "Сервер Eltex-SC"	Имя отправителя.
host: email.com	Адрес SMTP-сервера.
port: 587	SMTP-порт сервера.
protection: starttls	Протокол шифрования, используемый при подключении к серверу. Допустимые значения: none, starttls, ssl.
push:	Включение/выключение push-сообщений.
firebase: enabled: false	Включение/выключение push-сообщений для Android.
apns: enabled: false	Включение/выключение push-сообщений для iOS.
payment: logLevel: INFO	Параметры установки сервиса платежей для тарифов.
authorization:	Параметры authorization server.

logLevel: INFO	Уровень отладки внутри IoT Authorization Server.
captchaLevel: "easy"	Уровень сложности CAPTCHA: easy, medium, hard.
selfRegistration:	Параметры для управления доступностью самостоятельной регистрации.
allow: true	Доступность самостоятельной регистрации.
allowDemo: true	Доступность самостоятельной регистрации демо-аккаунтов.
allowSocialNetworks: false	Доступность самостоятельной регистрации через соцсети.
clientRegistrations:	Параметры клиентских регистраций (через соцсети).
google:	Наименование соцсети.
enable: true	Доступность самостоятельной регистрации через соцсеть.
clientId: "GoogleClientIdChangeMe"	ID клиента.
clientSecret: "GoogleClientSecretChangeMe"	Секрет клиента.
microsoft:	Наименование соцсети.
enable: true	Доступность самостоятельной регистрации через соцсеть.
clientId: "MicrosoftClientIdChangeMe"	ID клиента.
clientSecret: "MicrosoftClientSecretChangeMe"	Секрет клиента.
apple:	Наименование соцсети.
enable: true	Доступность самостоятельной регистрации через соцсеть.
clientId: "AppleClientIdChangeMe"	ID клиента.
keyId: "AppleKeyIdChangeMe"	
teamId: "AppleTeamIdChangeMe"	
yandex:	Наименование соцсети.
enable: true	Доступность самостоятельной регистрации через соцсеть.
clientId: "YandexClientIdChangeMe"	ID клиента.

clientSecret: "YandexClientSecretChangeMe"	Секрет клиента.
vk:	Наименование соцсети.
enable: true	Доступность самостоятельной регистрации через соцсеть.
clientId: "VkClientIdChangeMe"	ID клиента.
clientSecret: "VkClientSecretChangeMe"	Секрет клиента.
mailRu	Наименование соцсети.
enable: true	Доступность самостоятельной регистрации через соцсеть.
clientId: "mailRuClientIdChangeMe"	ID клиента.
clientSecret: "mailRuClientSecretChangeMe"	Секрет клиента.
skills:	
yandex: enabled: false	Параметры навыка Яндекс для интеграции с Умным домом (Алисой). Отображаются в карточке навыка.
clientId: "YandexClientIdChangeMe" password: "PasswordChangeMe"	Параметры для Basic Authentication.
skillId: ""	ID навыка, который необходимо указывать при отправке уведомлений.
oauthToken: ""	OAuth-токен, который необходимо указывать при отправке уведомлений.
sber: enabled: false	Параметры проекта умного дома Сбера для интеграции с Салютом. Отображаются в карточке проекта.
clientId: "SberClientIdChangeMe" password: "PasswordChangeMe"	Параметры для Basic Authentication.
bearerToken: ""	Bearer-Token, который необходимо указывать при отправке уведомлений.
marusya: enabled: false	Параметры проекта умного дома Mail.ru для интеграции с Марусей. Отображаются в карточке проекта/приложения.
clientId: "MarusyaClientIdChangeMe" password: "PasswordChangeMe"	Параметры для Basic Authentication.

appId: "MarusyaAppIdChangeMe"	App ID, который был назначен приложению VK при создании.
oauthToken: ""	OAuth-токен, который необходимо указывать при отправке уведомлений.
core:	Параметры платформы ELIS.
logLevel: INFO	Уровень отладки внутри ELIS.
ctlGate:	Порты платформы для подключения ZWave-контроллеров.
port: 8070	WS-порт для подключения контроллеров к платформе.
tcpPort: 8069	Порт для подключения контроллеров Ethernet к платформе в режиме TCP-клиент.
sslPort: 8072	WSS-порт для подключения контроллеров к платформе.
links:	
useHttpsForApi: false	Параметр, позволяющий использовать HTTPS при формировании ссылок к ресурсам самой платформы (например, прошивки).
useHttpsForCameraLinks: true	Параметр, позволяющий использовать HTTPS при формировании ссылок на фото с камер наблюдения.
useUiProxyForApi: false	Параметр, позволяющий использовать 'web.serverName' вместо 'iot.serverName' и 'web.httpPort'/'web.httpsPort' вместо 'core.api.port'/'core.api.sslPort' при формировании ссылок к API.
video:	Параметры для работы с видеосерверами.
flussonic:	Параметры Flussonic.
url: ""	URL сервера Flussonic.
apiKey: ""	Ключ API.
operatorId: ""	ID оператора.
adminLogin: ""	Логин администратора.
motion: enabled: false	Доступность фиксации движения.
evi:	Параметры видеосервера EVI.
url: ""	URL сервера EVI.
apiKey: ""	Ключ API.
operatorId: ""	ID оператора.

adminLogin: ""	Логин администратора.
irc:	Настройки ИК-пульта.
recTimeout: 15s	Время ожидания ИК-команды от пользователя в секундах.
cmdTimeout: 100ms	Таймаут записи команды (отсутствия фронтов) в мс.
guard:	Настройки охраны.
deviceRequestDelay: 15s	Время, которое дается на включение FLIRS устройств (ожидание подтверждения от контроллера) при постановке на охрану.
clearContextExtraCacheDelay: 1m	Время, которое прибавляется ко времени задержки на очистку охранного кэша при постановке/снятии с охраны, на случай если охрана не завершила процесс постановки/снятия. Время задержки формируется как количество охранных устройств, умноженное на deviceRequestDelay.
elk:	Параметры установки сервисов логирования (Elasticsearch + Logstash + Kibana).
enable: false	Параметр, позволяющий добавить в платформу appender, отправляющий логи в logstash. В нем нет необходимости, если ELK не развернут или не настроен: это спровоцирует сообщения об ошибках отправки в логах платформы.
serverName: "{{ iot.serverName }}"	Имя (IP-адрес) сервера, на котором будет развернут ELK. По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с платформой (на том же хосте). В таком случае хосты в инвентаре в группах [iot] и [monitoring] должны совпадать.
installDir: /storage/elk	Директория для установки системы логирования.
slgate:	
serverName: "{{ iot.serverName }}"	Имя сервера, на котором будет развернут SLGATE. ⚠ Нельзя использовать IP-адрес, т. к. такая схема не будет работать. По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с платформой (на том же хосте). В таком случае хосты в инвентаре в группах [iot] и [slgate] должны совпадать.
installDir: /storage/slgate	Директория для установки SLGATE.

✓ Для функций самостоятельной регистрации, регистрации демонстрационных учетных записей, а также для процедуры восстановления пароля может потребоваться активация почтовых оповещений через email.

7. После этого можно запустить установку:

Установка

```
cd /etc/ansible-iot-1.36
ansible-playbook install_iot.yml
```

Платформа будет доступна по адресу: **http://[Адрес вашего сервера ELIS]**.
Адрес сервера был ранее указан в переменной **iot.serverName** файла конфигурации **/etc/ansible-iot-1.36/vars/default.yml**.

Порты доступа к API платформы можно изменить только в случае редактирования соответствующих настроек в файле конфигурации.

Содержимое файла конфигурации доступно в полной документации к ELIS.

Конфигурации для ядра: **/etc/ansible-iot-1.36/templates/iot/docker-compose/base/docker-compose.yml.j2** и веб-сервера: **/etc/ansible-iot-1.36/templates/iot/web/base_config.j2**

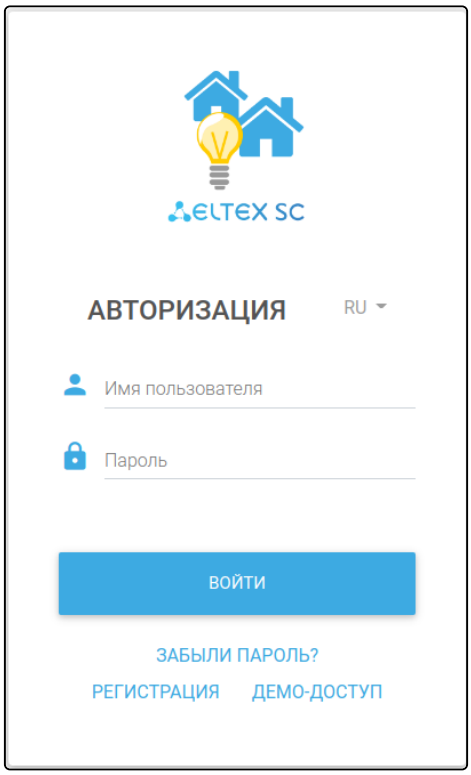
Директория хранения журналов работы платформы: **/storage/iot/core/var/log/eltex-sc/server.log**.

3 Первый вход на платформу ELIS

Для доступа к платформе через web-интерфейс используйте адрес сервера ELIS, указанный ранее в конфигурации **/etc/ansible-iot-1.36/vars/default.yml**.

1. В адресной строке вашего браузера введите: **http://[Адрес вашего сервера ELIS]**. Откроется страница авторизации.
2. Введите имя пользователя и пароль в соответствующие поля. Нажмите кнопку **Войти**.

Данные учетной записи «Администратор»:
Логин: **admin**
Пароль: **Test18plat34Form**



Более подробная документация доступна по ссылке Установка платформы или в разделе «Документы и файлы» карточки продукта на официальном сайте Элтекс.

Техническая поддержка

Для получения технической консультации по вопросам эксплуатации оборудования ООО «Предприятие «ЭЛТЕКС» вы можете обратиться в Сервисный центр компании:

Форма обращения в приложении Eltex Home: в настройках аккаунта перейдите в «Центр поддержки». Опишите проблему в форме обращения.

Электронная почта (при отсутствии учетной записи в Eltex Home): iot@eltex-co.ru

Форма обратной связи на сайте: https://eltex-co.ru/support/

Servicedesk: https://servicedesk.eltex-co.ru

На официальном сайте компании вы можете найти техническую документацию и программное обеспечение для продукции ООО «Предприятие «ЭЛТЕКС», обратиться к базе знаний или оставить интерактивную заявку:

Официальный сайт компании: https://eltex-co.ru/

База знаний: https://docs.eltex-co.ru/display/EKB/Eltex+Knowledge+Base