Платформа
ELIS

Установка платформы
Версия: 1.35

Содержание

1 Рекомендуемые характеристики сервера 3
2 Установка платформы ELIS 4
3 Установка платформы ELIS с внешней БД MongoDB 34
4 Работа с контейнерами 37
5 Расположение конфигурационных файлов 38
6 Проверка работоспособности платформы 39
7 Переход с http на https 41
8 Настройка CAPTCHA 44

1 Рекомендуемые характеристики сервера

Система ELIS строится по клиент-серверной архитектуре. Серверную часть рекомендуется устанавливать на многопроцессорный компьютер под управлением OS Ubuntu 20.

Производительность сервера зависит от числа пользователей, которые будут зарегистрированы на платформе.

Минимальные системные требования сервера[1]:

- 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.

[1]Минимально необходимая конфигурация для запуска ядра платформы с микросервисами.

2 Установка платформы ELIS

В инструкции описан процесс установки платформы ELIS версии 1.35 на операционную систему Ubuntu 20.04. Информацию по установке более ранних версий платформы (1.34 и ниже) можно найти по ссылке: Архив. ELIS.

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

⚠ С версии 1.25 сервис eltex-notification-gw(ngw) не требует дополнительной настройки через файл **notification.properties.j2.** Все настройки вынесены в файл **vars/default.yml**

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

Перед развертыванием платформы необходимо установить систему 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-1.35.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-1.35**

5. Отредактируйте файл **/etc/ansible-iot-1.35/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-1.35/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

  # 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:
  # Имя сервера, на котором будет развернут SLGATE.
  # ВАЖНО!!! Нельзя использовать IP-адрес, т.к. такая схема не будет работать!
```

```
215   # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
      платформой (на том же хосте).
216   # В таком случае хосты в инвентаре в группах [iot] и [slgate] должны совпадать.
217   serverName: "{{ iot.serverName }}"
218   # Директория для установки SLGATE.
219   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 работы функций самостоятельной регистрации, регистрации демонстрационных учетных записей, а также для процедуры восстановления пароля может потребоваться активация почтовых оповещений через email.

Также в **vars/service_parameters.yml** задаются дополнительные параметры:

Содержимое конфигурационного файла /vars/service_parameters.yml

```yaml
---
# Экземпляр платформы с параметрами по умолчанию.
instance: "IoT-instance-default"

# Версия контейнеров.
release: 1.35

# Имя репозитория docker registry, содержащего docker-образы для развертывания.
registry: hub.eltex-co.ru

# Список сервисов для перезапуска (при запуске плейбуков restart_*.yml).
# Можно оставить пустым, а при запуске передавать параметром командной строки.
services: []

# Нужно ли выполнять подготовку дистрибутива к установке. Этот шаг полезен при "чистой" установке
# на только что созданный сервер. Если ранее уже была выполнена установка компонентов IoT через ansible,
# то такая подготовка не требуется и этот шаг можно пропустить для экономии времени.
withDistroPreparingStep: true

# Суффикс, добавляемый к имени каждого контейнера (помогает избежать конфликты имен контейнеров).
containerNameSuffix: ""

# Суффикс, добавляемый к имени создаваемой сети docker (помогает избежать конфликты имен сетей docker).
networkNameSuffix: ""

# Параметры сервисов IoT (для docker-compose), сгруппированные по именам.
# 'enable' - должен ли присутствовать сервис в docker-compose.yml.
# 'port.map' - номер порта сервиса в сети хоста.
# 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
# 'db.name' - имя БД, используемой сервисом (связкой сервисов).
iot:
  swarm:
    enabled: false
    # Имя stack-а для запуска в docker-swarm.
    stack: swarm_iot
  # Параметры мониторинга использования дискового пространства. Должны соответствовать требованию:
  # warnThreshold > criticalThreshold > 0, иначе мониторинг дискового пространства будет отключен.
  diskUsage:
    # Порог дискового пространства (в %), при достижении которого все логгеры микросервисов переводятся в режим WARN
    # (отображаются сообщения с тегами WARN и ERROR). Количество бэкапов баз уменьшается пропорционально приближению к
    # порогу criticalThreshold.
    warnThreshold: 20
    # Порог дискового пространства (в %), при достижении которого все логгеры микросервисов переводятся в режим ERROR
    # (отображаются только сообщения с тегом ERROR). Бэкапы баз не выполняются.
    criticalThreshold: 10
  # использовать сервис Notification для отправки пушей, вместо старой логики
  useNotification: true
```

```
  # Нужно ли установить лимиты на сервисы в docker compose
  limits:
    enable: false
  mongodb:
    replication:
      label: has-mongodb
    limits:
      enable: false
      cpus: 1.0
      memory: 4G
    port:
      map: 27017
      export: false
  broker:
    logLevel: INFO
    replication:
      label: has-broker
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    external:
      # Будет ли развернут брокер на стороннем хосте.
      enable: false
      # Нужно ли устанавливать брокер на сторонний хост средствами ansible.
      install: false
      # Если установка брокера на сторонний хост будет выполняться средствами
 ansible, то в какую директорию.
      installDir: /storage/broker
      # Внешний адрес стороннего хоста, по которому к брокеру будут подключаться
платформа и устройства.
      host: "external.broker.address"
      port:
        map: 8883
    internal:
      port:
        map: 8083
        export: false
    db:
      name: iot-broker
    jconsole:
      enable: false
      port: 32002
      # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
      # При развертывании докера - здесь указывается адреса хоста, где работает
докер.
      host: "10.20.30.40"
    hivemq:
      rootFolder: /hivemq/
  olapservice:
    logLevel: INFO
    replication:
      label: has-olap
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    external:
      # Будет ли развернут olapservice на стороннем хосте.
      enable: false
```

```
      # Нужно ли устанавливать olapservice на сторонний хост средствами ansible.
      install: false
      # Если установка olapservice на сторонний хост будет выполняться средствами ansible, то в какую директорию.
      installDir: /storage/olapservice
      # Внешний адрес стороннего хоста, по которому платформа будет подключаться к olapservice'y.
      host: "external.olapservice.address"
    port:
      map: 8023
      export: false
    db:
      name: iotcore
      # Нужно ли устанавливать clickhouse (полезно для разработки olapservice).
      install: true
      replication:
        label: has-clickhouse
      limits:
        enable: false
        cpus: 1.0
        memory: 4G
      port:
        map: 8123
        export: false
  captcha:
    enable: true
    replication:
      label: has-captcha
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    port:
      map: 8088
      export: false
    caseSensitive: true
    allowedSizes:
      - "312x45"
      - "270x40"
    instance: "captcha:8088"
    proportion: 100
  zscaptcha:
    enable: false
    replication:
      label: has-captcha
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    port:
      map: 8089
      export: false
    caseSensitive: true
    instance: "zs-captcha:8089"
    proportion: 0
  authorization:
    # Развертывание окружения без auth server, полезно для разработки auth server.
    enable: true
    # API-ключ для авторизации HTTP-запросов в сервис авторизации от внешних систем, таких как Core.
```

```
    # Специальное значение 'autogenerated' означает, что apiKey будет сгенерирован в процессе установки.
    apiKey: "autogenerated"
    replication:
      label: has-authorization-server
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    external:
      host: "external.authorization.address"
    port:
      map: 8091
      export: false
      ssl:
        map: 8093
        export: false
    db:
      name: iot-authorization-server
    jconsole:
      enable: false
      port: 32003
      # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
      # При развертывании докера - здесь указывается адреса хоста, где работает докер.
      host: "10.20.30.40"
    email:
      blacklist:
        # "0 * * * * ?" раз в минуту
        # "0 ${random.int[60]} ${random.int[1,5]} 1 * ?" раз в неделю
        cron: "0 ${random.int[60]} ${random.int[1,5]} 1 * ?"
    swagger:
      enable: false
    rabbit:
      eventQueue:
        name: platform-rpc-event-queue
      # Количество консьюмеров на стороне платформы
      platformConsumers:
        # Данный параметр не может быть больше чем maxCount.
        count: 16
        # Максимальное количество одновременных консьюмеров очереди.
        maxCount: 16
      # Настройки сервиса уведомлений
      notification:
        # Количество консюмеров
        count: 1
        #Максимальное количество консюмеров
        maxCount: 5
        #Наименования обменика
        exchange: notification-exchange
        #Ключ маршртута для уведомлений
        envelopeRoutingKey: notification
        #Ключ маршртута событий входа/выхода
        userEventRoutingKey: userevent
  payment:
    enable: true
    # ApiKey используется для доступа к АПИ платежного сервиса.
    apiKey: "CHANGE_ME"
    replication:
      label: has-payment-server
```

```
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    port:
      map: 9091
      export: false
      ssl:
        map: 9093
        # Если експорт false, то запросы будут проксироваться через Nginx. Для тестирования флаг можно включать.
        # Если флаг true, то запросы должны ходить на прямую в payment service
        export: false
    internal:
      # Максимальный временной период, через который будет повторятся запрос статуса платежа на платежный сервис.
      orderCheckTimePeriodMax: "30m"
      # Максимальное время, через которое сервис перестает опрашивать ордер.
      orderCheckTimeTotal: "24h"
      # Промежуток времени, через который повторяется отправка сообщения на платформу об успешной оплате,
      # чтобы платформа создала подписку. Повторная отправка отменяется если платформа прислала сообщение о создании
      # подписки или если количество попыток передать сообщение превысило параметр checkSubscriptionCreatedMaxAttempts
      checkSubscriptionCreatedPeriodTime: "5m"
      # Количество попыток передать платформе сообщение об успешной оплате.
      checkSubscriptionCreatedMaxAttempts: 10
    db:
      name: iot-payment-server
    jconsole:
      enable: false
      port: 32004
      # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
      # При развертывании докера - здесь указывается адреса хоста, где работает докер.
      host: "10.20.30.40"
    swagger:
      enable: false
    paySystem:
      # Необходимо указать ендпоинт на стороне IoT-Core куда будут переадресовываться запросы с ЮКассы.
      # МП приложения могут сами распозновать , что вебвью с ЮКассой закрыта, и кидать необходимый запрос на Iot-core
      # и переходить на нужный экран. Веб так не умеет, поэтому создан этот параметр. Когда веб поддержит покупку
      # тарифов, нужно будет уточнить на какую именно страницу нужно делать редирект.
      redirectUrlAfterPay: "https://eltexhome.ru"
      yooKassa:
        url: "https://api.yookassa.ru/v3"
        # Идентификатор магазина.
        shopId: ""
        secretKey: ""
        # Список доступных IP адресов запросов от Юкассы. Взято из официально документации.
        # https://yookassa.ru/developers/using-api/webhooks#ip
        availableHosts:
          - "185.71.76.0/27"
          - "185.71.77.0/27"
```

```
          - "77.75.153.0/25"
          - "77.75.156.11"
          - "77.75.156.35"
          - "77.75.154.128/25"
          - "2a02:5180::/32"
    rabbit:
      queue:
        # Очередь для приема сообщений от платформы.
        paymentReceive: "paymentReceive"
        # Очередь для приема сообщений от платформы и синхронного ответа во временную очередь.
        paymentReceiveAndReply: "paymentReceiveAndReply"
        # Время ожидания синхронного ответа от PaymentServer на сообщение, отправленное в очередь
        # paymentReceiveAndReply
        paymentReceiveAndReplyQueueTimeOut: "30s"
        # Очередь для отправки сообщений на платформу.
        platformReceive: "platformReceive"
      consumers:
        # Количество одновременных консьюмеров в Iot-Core для очереди platformReceive.
        # Данный параметр не может быть больше чем maxCount.
        coreCount: 1
        # Максимальное количество одновременных консьюмеров в Iot-Core для очереди platformReceive.
        coreMaxCount: 1
        # Количество одновременных консьюмеров в Iot-Pqyment-Server для очереди paymentReceiveAndReply.
        paymentCount: 10
        # Максимальное количество одновременных консьюмеров в Iot-Pqyment-Server для очереди paymentReceiveAndReply.
        paymentMaxCount: 20
  core:
    # Развертывание окружения без платформы, полезно для разработки core.
    enable: true
    replication:
      replicaCount: 1
      label: has-core
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    port:
      map: 8071
      export: false
      ssl:
        map: 8073
        export: false
    db:
      name: iot-core
      # Имена устаревших БД в MongoDB, используемые платформой.
      deprecated:
        fs:
          name: iot-fs
        licenses:
          name: iot-licenses
        events:
          name: iot-events
    jconsole:
      enable: false
```

```
      port: 32001
      # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
      # При развертывании докера - здесь указывается адреса хоста, где работает докер.
      host: "10.20.30.40"
    swagger:
      enable: false
    broker:
      # Нужно ли платформе пытаться подключиться к MQTT Broker.
      enable: true
      threadPools:
        main:
          size: 16
    olapservice:
      # Нужно ли платформе пытаться подключиться к OlapService.
      enable: true
    mjollnir:
      # URL для Mjollnir.
      url: "http://smart.eltex-co.ru:8078/api/v1"
    slgate:
      instances:
        - slgate:4443
      ignoreServerCertCheck: false
    video:
      evi:
        ignoreServerCertCheck: false
        checkEventSenderAddress: true
      # Параметры WebRTC, которые необходимы камере для формирования своих ICE-кандидатов.
      webrtc:
        # Параметры STUN-сервера.
        stun:
          # URL STUN-серверов.
          urls:
            - "stun.example1.com:3478"
            - "stun.example2.com:3478"
        # Параметры TURN-сервера.
        turn:
          # Параметры TURN:SFU
          sfu:
            # Принудительное включение
            force: false
            # Таймаут на ожидание ивента от EVI о начале webrtc-сессии между камерой и TURN:SFU
            eventTimeout: 10s
        # Параметры P2P
        p2p:
          # Таймаут на ожидание SDP-answer от камеры
          sdpAnswerTimeout: 60s
        # Время ожидания проверки камерой типа своего NAT
        checkCameraNatTypeTimeout: 10s
        # Время выполнения ping к host candidate клиента
        pingHostCandidateTimeout: 10s
    rabbit:
      notification:
        exchange: notification-exchange
        envelopeRoutingKey: notification
        userEventRoutingKey: userEvent
        count: 1
        maxCount: 1
```

```
  web:
    # Развертывание окружения без WEB, полезно для разработки web.
    enable: true
    replication:
      label: has-web
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
  hazelcast:
    clusterName: iot-core
    instanceName: iot-core-hazelcast-instance
    replication:
      label: has-hazelcast
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    external:
      port:
        map: 5701
        export: false
  rabbitmq:
    replication:
      label: has-rabbitmq
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    admin:
      user: guest
      password: guest
      port:
        map: 15671
        export: false
      sslPort:
        map: 15672
        export: false
    port:
      map: 5672
      export: false
  zwayproxy:
    logLevel: INFO
    external:
      enable: false
      # Если установка zwayproxy на сторонний хост будет выполняться средствами ansible, то в какую директорию.
      installDir: /storage/zwayproxy
    replication:
      replicaCount: 2
      label: has-zwayproxy
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    port:
      map: 8070
      export: true
    sslPort:
      map: 8072
```

```
      export: true
    rabbit:
      queue:
        # Идентификатор очереди для отправки сообщений из прокси в платформу.
        platform: zway-proxy-platform
      exchange:
        # Идентификатор топик-коллектора.
        proxy: zway-proxy-topic-exchange
      # Количество консьюмеров на стороне платформы
      platformConsumers:
        # Данный параметр не может быть больше чем maxCount.
        count: 16
        # Максимальное количество одновременных консьюмеров очереди.
        maxCount: 16
      # Количество консьюмеров на стороне zwayProxy service
      proxyConsumers:
        # Количество одновременных консьюмеров очереди. Данный параметр не может
быть больше чем maxCount.
        count: 1
        # Максимальное количество одновременных консьюмеров очереди.
        maxCount: 1
  wsproxy:
    logLevel: INFO
    maxConnections: 25000
    external:
      enable: false
      # Если установка wsproxy на сторонний хост будет выполняться средствами
 ansible, то в какую директорию.
      installDir: /storage/wsproxy
      # Адрес хоста куда nginx будет проксировать запросы.
      host: "1.2.3.4"
      port: 8075
    replication:
      replicaCount: 2
      label: has-wsproxy
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    port:
      map: 8075
      export: true
    rabbit:
      queue:
        # Идентификатор очереди для отправки сообщений из прокси в платформу.
        platform: ws-proxy-platform
      exchange:
        # Идентификатор топик-коллектора.
        proxy: ws-proxy-topic-exchange
      # Количество консьюмеров на стороне платформы
      platformConsumers:
        # Данный параметр не может быть больше чем maxCount.
        count: 16
        # Максимальное количество одновременных консьюмеров очереди.
        maxCount: 16
      # Количество консьюмеров на стороне wsProxy service
      proxyConsumers:
        # Количество одновременных консьюмеров очереди. Данный параметр не может
быть больше чем maxCount.
        count: 16
```

```
        # Максимальное количество одновременных консьюмеров очереди.
        maxCount: 16
 # Параметры установки сервисов мониторинга (Prometheus + Grafana).
 monitoring:
   external:
     # Будет ли развернут monitoring на стороннем хосте.
     enable: true
     # Нужно ли устанавливать monitoring на сторонний хост средствами ansible.
     install: false
     # Если установка monitoring на сторонний хост будет выполняться средствами ansible, то в какую директорию.
     installDir: /storage/monitoring
   mongodbExporter:
     replication:
       label: has-mongodb-exporter
     limits:
       enable: false
       cpus: 1.0
       memory: 4G
     enable: true
     port:
       map: 9216
       export: false
   nginxExporter:
     replication:
       label: has-nginx-exporter
     limits:
       enable: false
       cpus: 1.0
       memory: 4G
     enable: true
     port:
       map: 9113
       export: false
   prometheus:
     replication:
       label: has-prometheus
     limits:
       enable: false
       cpus: 1.0
       memory: 4G
     port:
       map: 9090
       export: false
     # Период опроса сервисов со стороны Prometheus в секундах. Небольшие значения (менее десятков секунд) значительно
     # увеличат нагрузку на сеть.
     # При изменении параметра рекомендуется поменять переменную grafana.scrapeInterval на значение не ниже указанного
     # для Prometheus, чтобы не терять метрики.
     scrapeInterval: 60
   grafana:
     replication:
       label: has-grafana
     limits:
       enable: false
       cpus: 1.0
       memory: 4G
     port:
       map: 3000
```

```
      # Период опроса Prometheus со стороны Grafana в секундах.
      scrapeInterval: 60
  notification:
    enable: true
    logLevel: INFO
    replication:
      label: has-notification
    limits:
      enable: true
      cpus: 1.0
      memory: 4G
    jconsole:
      enable: false
      port: 32002
      # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
      # При развертывании докера - здесь указывается адреса хоста, где работает
докер.
      host: "10.20.30.40"
    port:
      export: false
      map: 9095
    rabbit:
      exchange: notification-exchange
      envelopeRoutingKey: notification
      userEventRoutingKey: userevent
    vault:
      uri: https://127.0.0.1:8200
      token: RootToken
      ssl:
        useTrustStore: true
        trustStore: file:///etc/vault.jks
        trustStoreType: jks
        trustStorePassword: vault0
      rootPath: credentials
    push:
      fcm:
        # Период между отправкой пушей, для достижения ограничения частоты отправки
пушей.
        delay: 250ms
    hazelcast:
      instance: iot-notification
      cluster: iot-notification
      addresses: "notification-hazelcast:5701"
      replication:
        label: has-notification-hazelcast
      limits:
        enable: true
        memory: 1G
        cpus: 1.0
      external:
        port:
          enable: false
          map: 15701

slgate:
  db:
    name: "activator"
    user: "activator"
    password: "activator"
    port:
```

```
      map: 5432
      export: false
  vpn:
    port:
      map: 1194
    protocol: udp
    maxClients: 1024
  easyrsa:
    req:
      country: "RU"
      province: "Novosibirsk"
      city: "Novosibirsk"
      organization: "Eltex"
      organizationUnit: "EMS"
      email: "eltex@eltex-co.ru"
  activator:
    port:
      map: 8899
      export: false
  proxy:
    scheme: "https://"
    useClientCert: false
    clientCertPassword: "CHANGE_ME"
    port:
      map: 4443

# Параметры сервисов ELK (для docker-compose), сгруппированные по именам.
# 'port.map' - номер порта сервиса в сети хоста.
# 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
elk:
  elasticsearch:
    rest:
      port:
        map: 9200
    nodes:
      port:
        map: 9300
  logstash:
    port:
      map: 5001
    api:
      port:
        map: 9600
  kibana:
    port:
      map: 5601
  fluentBit:
    port:
      map: 24224
    #Индексы, создаваемые в elasticsearch со стороны fluentBit
    targets:
      - core
      - authorization
      - notification
      - payment
      - zwayproxy
      - wsproxy
      - mqttbroker
      - olapservice
```

Таблица описания значений в файле настроек vars/service_parameters.yml

release: 1.35	Версия контейнеров.
swarm: enabled: false	Включение/выключение оркестрации.
stack: swarm_iot	Имя stack-а для запуска в docker-swarm.
registry: hub.eltex-co.ru	Имя репозитория docker registry, содержащего docker-образы для развертывания.
services: []	Список сервисов для перезапуска (при запуске плейбуков restart_*.yml). Можно оставить пустым, а при запуске передавать параметром командной строки.
withDistroPreparingStep: true	Параметр, отражающий необходимость дистрибутива к установке. Этот шаг полезен при «чистой» установке на только что созданный сервер. Если ранее уже была выполнена установка компонентов IoT через Ansible, то такая подготовка не требуется и этот шаг можно пропустить для экономии времени.
containerNameSuffix: ""	Суффикс, добавляемый к имени каждого контейнера (помогает избежать конфликты имен контейнеров).
networkNameSuffix: ""	Суффикс, добавляемый к имени создаваемой сети docker (помогает избежать конфликты имен сетей docker).
diskUsage:	Параметры мониторинга использования дискового пространства. Должны соответствовать требованию:warnThreshold > criticalThreshold > 0, иначе мониторинг дискового пространства будет отключен.
warnThreshold: 20	Порог дискового пространства (в %), при достижении которого все логгеры микросервисов переводятся в режим WARN (отображаются сообщения с тегами WARN и ERROR). Количество бэкапов баз уменьшается пропорционально приближению к порогу criticalThreshold.
criticalThreshold: 10	Порог дискового пространства (в %), при достижении которого все логгеры микросервисов переводятся в режим ERROR (отображаются только сообщения с тегом ERROR). Бэкапы баз не выполняются.
limits enable: false	Параметр, позволяющий установить лимиты на сервисы в docker compose.
iotServices:	Параметры сервисов IoT (для docker-compose), сгруппированные по именам. В данной таблице описание всех параметров приведено на примере сервиса broker.
broker: enable: true	'enable' — параметр, отражающий необходимость присутствия сервиса в docker-compose.yml.
limits:	Лимиты ресурсов для конкретного сервиса.

enable: true	Включение/выключение лимитов ресурсов для конкретного сервиса.
cpus: 1.0	Лимит ядер процессора.
memory: 4G	Лимит выделяемой памяти.
external: port: map: 8883 internal: port: map: 8083	'port.map' — номер порта сервиса в сети хоста.
export: false	'port.export' — параметр, отражающий необходимость маппинга порта из контейнера в сеть хоста.
db: name: iot-broker	'db.name' — имя БД, используемой сервисом (связкой сервисов).
zwayproxy: enable: false	Включение/выключение сервиса проксирования запросов zway (снижает нагрузку на ядро).
rabbit: queue: platform: zway-proxy-platform	Идентификатор очереди для отправки сообщений из прокси в платформу.
exchange: proxy: zway-proxy-topic-exchange	Идентификатор топик-коллектора.
platformConsumers:	Количество консьюмеров на стороне платформы.
count: 1	Данный параметр не может быть больше чем maxCount.
maxCount: 1	Максимальное количество одновременных консьюмеров очереди.
proxyConsumers:	Количество консьюмеров на стороне zwayProxy service.
count: 1	Количество одновременных консьюмеров очереди. Данный параметр не может быть больше чем maxCount.
maxCount: 1	Максимальное количество одновременных консьюмеров очереди.
elkServices:	Параметры сервисов ELK (для docker-compose), сгруппированные по именам.
monitoringServices:	Параметры сервисов мониторинга (для docker-compose), сгруппированные по именам.

testdata: enable: false	Параметр, отражающий необходимость создания тестовых учетных записей на платформе.
swagger: enable: false	Параметр, отражающий необходимость включения swagger (описание API платформы).
mqttbroker: enable: true	Параметр, отражающий необходимость подключения к MQTT Broker.
olapservice: enable: true	Параметр, отражающий необходимость подключения к OlapService.
core: db: name: iot-core fs: db: name: iot-fs licenses: db: name: iot-licenses events: db: name: iot-events	Имена БД в MongoDB, используемые платформой.
mjollnir:	Сервис проверки лицензий.
url: "http://smart.eltex-co.ru:8078/api/v1"	URL для Mjollnir.

7. После этого можно запустить установку:

Установка

```
cd /etc/ansible-iot-1.35
ansible-playbook install_iot.yml
```

⚠ Если ранее использовалась версия платформы ELIS с подключением внешней БД MongoDB, необходимо удалить запись о репозитории mongodb.org из APT (например, **/etc/apt/sources.list.d/mongodb-org-4.4.list**).

8. Выполните проверку статуса контейнеров:

Проверка статуса контейнеров

```
docker ps
```

Вывод команды docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
36c21b863cc9	hub.eltex-co.ru/iot-double-web:1.35	/docker-entrypoint.…	2 minutes ago	Up 2 minutes	0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp	iot-double-web
01fe2697e5ff	hub.eltex-co.ru/iot-core:1.35	/extras/app-starter	2 minutes ago	Up 2 minutes	.0.0.0:8069-8073->8069-8073/tcp, :::8069-8073->8069-8073/tcp	iot-core
6ddca5a875a8	gub.eltex.loc:9017/iot-authorization-server:1.35	/extras/app-starter	2 minutes ago	Up 2 minutes		iot-authorization-server
07d1f93831bd	hub.eltex-co.ru/iot-mqttbroker-mongo:1.35	java -cp @/app/jib-…	2 minutes ago	Up 2 minutes	0.0.0.0:8883->8883/tcp, :::8883->8883/tcp	iot-mqtt-broker
8f8401f5ae0a	hub.eltex.loc:9017/iot-hazelcast:1.35	"java --add-modules …"	2 minutes ago	Up 2 minutes		iot-ngw-sc
228d41c96cba	hub.eltex-co.ru/iot-olapservice:1.35	java -cp @/app/jib-…	2 minutes ago	Up 2 minutes		iot-olapservice
e8e2899f2c8d	hub.eltex-co.ru/iot-captcha:1.35	java -jar LibreCapt…	2 minutes ago	Up 2 minutes	8888/tcp	iot-captcha
57c02941cc4f	hub.eltex-co.ru/iot-mongo6:1.35	/entrypoint.sh	2 minutes ago	Up 2 minutes	0.0.0.0:27017->27017/tcp, :::27017->27017/tcp	iot-mongo
7c3d8d5c4137	hub.eltex.loc:9017/iot-clickhouse-server:1.35	"docker-entrypoint.s…"	2 minutes ago	Up 2 minutes	8123/tcp, 9000/tcp, 9009/tcp	iot-iot-mysql

0327972ab361	hub.eltex.loc:9017/iot-rabbitmq:1.35	"docker-entrypoint.s…"	2 minutes ago	Up 2 minutes	4369/tcp, 5671-5672/tcp, 15691-15692/tcp, 25672/tcp	
a5bcb7e27390	hub.eltex.loc:9017/iot-wsproxy:1.35	java -cp @/app/jib-…	2 minutes ago	Up 2 minutes	0.0.0.0:8075->8075/tcp, :::8075->8075/tcp	iot-wsproxy
984b90122ea9	hub.eltex.loc:9017/iot-zwayproxy:1.35	java -cp @/app/jib-…	2 minutes ago	Up 2 minutes	0.0.0.0:8070->8070/tcp, :::8070->8070/tcp, 0.0.0.0:8072->8072/tcp, :::8072->8072/tcp	iot-zwayproxy

Платформа будет доступна по адресу: http://[Адрес вашего сервера ELIS]
Адрес сервера был ранее указан в переменной **iot.serverName** файла конфигурации **/etc/ansible-iot-1.35/vars/default.yml**

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

9. Добавьте файл лицензий.

Подробное описание процесса добавления файла лицензий доступно в документации Интерфейс администратора в разделе **Лицензии.**

Время использования платформы и количество добавляемых устройств ограничивается типом лицензии, которая приобретается заказчиком.

3 Установка платформы ELIS с внешней БД MongoDB

В случае развертывания внешней БД MongoDB, т.е. при установке платформы ELIS с внешним процессом/службой mongodb:

> ⚠ Доступ к репозиториям mongodb.org может быть ограничен. В случае проблем с установкой пакетов необходимо использовать зеркало репозитория или вручную установить deb-пакеты (например, доступные в публичном репозитории).

1. Установите необходимые зависимости и сервис MongoDB (например **версии 4.4**):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install -y software-properties-common gnupg build-essential net-tools dkms
wget https://www.mongodb.org/static/pgp/server-4.4.asc
sudo apt-key add server-4.4.asc
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
sudo apt-get update
sudo apt-get install -y mongodb-org
sudo service mongod start
sudo service mongod status
```

2. В файле **/etc/mongod.conf** в секции **net** укажите:

```
port: 27017
bindIp: 0.0.0.0
```

И перезапустите сервис mongod:

```
sudo service mongod restart
sudo service mongod status
```

3. Установите Ansible на сервер:

```
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
sudo ansible-galaxy collection install community.general
sudo ansible-galaxy collection install community.crypto
sudo ansible-galaxy collection install community.docker
```

4. Подготовьте конфигурацию:

Распаковка архива с конфигурацией

```
tar -C /etc -xvf ansible-iot-1.35.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-1.35.**

5. Отредактируйте файл **/etc/ansible-iot-1.35/inventory.**

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

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

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. Настройте параметры доступа к платформе, в том числе параметры для работы с внешним сервисом БД. Для этого отредактируйте файлы конфигурации **/etc/ansible-iot-1.35/vars/default.yml** и **/etc/ansible-iot-1.35/vars/service_parameters.yml.**

7. Откройте файл **default.yml** в любом доступном текстовом редакторе, например **nano**. Укажите корректный IP-адрес или доменное имя в переменной **iot.serverName.**

Для блока параметров mongodb: параметр **enable** установите в значение **true**, настройте параметр подключения в '**uri**'.

Содержимое конфигурационного файла vars/default.yml

```
 # Параметры MongoDB.
  mongodb:
    # Версия MongoDB. На старой аппаратной платформе, не поддерживающей 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"
```

8. После этого можно запустить установку:

```
cd /etc/ansible-iot-1.35
sudo 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-1.35/vars/default.yml**

9. Добавьте файл лицензий.

Подробное описание процесса добавления файла лицензий доступно в документации Интерфейс администратора в разделе **Лицензии.**

Время использования платформы и количество добавляемых устройств ограничивается типом лицензии, которая приобретается заказчиком.

4 Работа с контейнерами

В процессе обслуживания можно осуществлять перезапуск контейнеров:

Перезапуск контейнеров

```
ansible-playbook restart_iot.yml --extra-vars '{"services":["web", "core", "broker",
"olapservice"]}'
```

В квадратных скобках нужно перечислить сервисы, которые необходимо перезапустить (в примере это "web", "core", "broker", "olapservice"). Если не указать параметр "services" через --extra-vars, то будут перезапущены все сервисы.
Аналогичным образом можно остановить часть контейнеров:

Остановка контейнеров

```
ansible-playbook stop_iot.yml --extra-vars '{"services":["web", "core", "broker",
"olapservice"]}'
```

Или обновить все контейнеры из репозитория:

Обновление контейнеров

```
ansible-playbook update_iot.yml
```

Применение новой или измененной конфигурации:

Обновление контейнеров

```
ansible-playbook install_iot.yml
```

5 Расположение конфигурационных файлов

В **/etc/ansible-iot-1.35/templates** располагаются конфигурации для ядра: **/etc/ansible-iot-1.35/templates/iot/docker-compose/base/docker-compose.yml.j2** и веб-сервера: **/etc/ansible-iot-1.35/templates/iot/web/base_config.j2**

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

Файлы журналов разбиваются на части по размеру. Каждая часть сохраняется в файл с именем: server-YYYY-MM-DD.NN.log, где YYYY-MM-DD — дата, а NN — номер части.

При изменении конфигурации необходимо перезапустить **install_iot.yml**:
ansible-playbook install_iot.yml
Некоторые конфигурационные параметры дублируют файл конфигурации **/etc/ansible-iot-1.35/vars/default.yml** . При запуске платформы параметры, заданные в файле конфигурации **/etc/ansible-iot-1.35/vars/default.yml,** имеют наивысший приоритет по отношению к другим файлам конфигурации, а также ведут к их перезаписи.

Порты, используемые платформой по умолчанию:

- **8069** — порт для подключения контроллеров Ethernet к платформе в режиме TCP-клиент;
- **8070** — WS-порт для подключения контроллеров к платформе;
- **8071** — HTTP-порт API-платформы;
- **8072** — WSS-порт для подключения контроллеров к платформе;
- **8073** — HTTPS-порт API-платформы;
- **8883** — порт для подключения к MQTT-брокеру;
- **8088** — порт для CAPTCHA.

По окончании установки и конфигурирования сервер ELIS будет готов к работе. Взаимодействие с платформой происходит через веб-интерфейсы пользователей и администратора, а также через мобильное приложение Eltex Home.

Во всех случаях для входа в платформу необходимо указать ее адрес и номер порта. В общем случае в адресную строку браузера вводится следующее: **<Адрес сервера ELIS>**.

6 Проверка работоспособности платформы

В данном разделе представлен чек-лист для проверки работоспособности платформы.

Проверить занятость ядер процессора и оперативной памяти

```
htop
```

⚠ Необходимо следить за *MEM%* и *CPU%* процессов mongo и docker-контейнеров.

Проверить занятость физического места на диске

```
df -h
```

⚠ Обратите внимание на использование диска всеми подразделами.

Проверить соединения устройств с платформой

```
netstat -na | grep 8070
netstat -na | grep 8883
```

Параметры, на которые следует обратить внимание:

- Состояние соединений:
  - **ESTABLISHED**, **LISTEN** — норма;
  - **LAST_ACK** — связь с контроллером потеряна, рядом должно быть переоткрытое соединение с того же IP;
  - **TIME_WAIT**, **CLOSE_WAIT** — соединение зависло, неудачное закрытие;
- 2-й столбец — очередь принятых сервером пакетов;
- 3-й столбец — очередь пакетов, отправленных на контроллеры;
- 5-й столбец — IP-адреса контроллеров.

Необходимо следить, чтобы пакеты не копились в очереди на контроллер. Если такое происходит, значит веб-сокет этого контроллера недоступен, и нужно проверить событие и время события в лог-файлах платформы.

Проверить наличие ошибок в логе

Лог-файлы платформы по умолчанию находятся в **/storage/iot/core/var/log/eltex-sc/server***. Они разбиваются на части по размеру. Чтобы проверить все журналы на наличие некоторого параметра за определенную дату (например за 15 января 2022 года) и записать результаты в файл, выполните команду:

```
grep <значение для поиска> server-2022-01-15* > <имя файла для вывода>
```

Если требуется, можно дописать следующие опции после значения для поиска:

- **-i** — не чувствителен к регистру;
- **-n** — номера строк;
- **-h** — подавляет запись имени файла перед каждой строкой в результирующем файле;
- **-A** — количество строк после совпадения;
- **-B** — количество строк до совпадения.

Общий вид команды:

```
grep -i -n -A 5 -B 2 error server-2022-01-15* > errors.log
```

Ключевые слова для поиска:

- *ERROR;*
- *ID* или *IP* контроллера;
- *PONG* — ключевое слово, которое записывается, если контроллер не отвечает.

Проверить доступность веб-интерфейса

Самый простой способ проверить доступность веб-интерфейса платформы — открыть его в браузере (по IP или доменному имени — в зависимости от настроек сети и сервера).

Можно проверить доступность API платформы. Для этого в адресной строке браузера к URL веб-интерфейса платформы допишите:

```
<Адрес сервера ELIS>:<порт API>/api/v1/version
```

где порт API — вышеуказанный порт доступа к API по HTTP (по умолчанию **8071/8073**). Платформа должна моментально вернуть данные об установленной версии платформы, версии API и текущем времени на сервере в формате JSON.

Пример вывода:

Ответ сервера на запрос

```
{
  "version" : "1.35-3477",
  "api" : "1.0",
  "currentTime" : "2022-07-25T09:24:12.544842Z[Etc/UTC]"
}
```

7 Переход с http на https

1. Откройте файл docker-compose.yml.j2:

```
ansible-iot/templates/iot/docker-compose/base/docker-compose.yml.j2
```

2. Добавьте код в блок WEB:

Содержимое конфигурационного файла /docker-compose/base/docker-compose.yml.j2

```
{% if iotServices.web.enable %}
  web:
    image: "${REGISTRY_IOT}/iot-double-web:${RELEASE}"
    container_name: iot-double-web${CONTAINER_NAME_SUFFIX}
{% if limitsEnable and iotServices.web.limits.enable %}
    deploy:
      resources:
        limits:
          cpus: '${WEB_LIMITS_CPUS}'
          memory: ${WEB_LIMITS_MEMORY}
{% endif %}
{% if iotServices.core.enable %}
    depends_on:
      core:
        condition: service_started
    links:
      - core
{% endif %}
    environment:
      - ENABLE_CERTBOT=${WEB_ENABLE_CERTBOT}
      - SERVER_NAME=${WEB_SERVER_NAME}
      - CERTBOT_EMAIL=${WEB_CERTBOT_EMAIL}
      - HTTPS_PORT=${WEB_HTTPS_PORT}
      - TZ=${TIMEZONE}
    volumes:
      - "${INSTALL_DIR}/web/etc/nginx/nginx.conf:/etc/nginx/nginx.conf"
      - "${INSTALL_DIR}/web/etc/nginx/conf.d/default.conf:/etc/nginx/conf.d/default.conf"
      - "${INSTALL_DIR}/web/well-known:/usr/share/eltex-sc-web/www/.well-known"
{% if web.certbot.enable %}
      - "${INSTALL_DIR}/web/etc/letsencrypt:/etc/letsencrypt"
      - "${INSTALL_DIR}/web/var/log/letsencrypt:/var/log/letsencrypt"
{% endif %}
      - "${INSTALL_DIR}/ssl/certs/eltex-sc-web.crt:/etc/ssl/certs/eltex-sc-web.crt"
      - "${INSTALL_DIR}/ssl/private/eltex-sc-web.key:/etc/ssl/private/eltex-sc-web.key"
      - "/etc/ansible-iot/ssl/домен/fullchain.pem:/etc/ansible-iot/ssl/домен/fullchain.pem"
      - "/etc/ansible-iot/ssl/домен/privkey.pem:/etc/ansible-iot/ssl/домен/privkey.pem"
      - "/etc/ansible-iot/ssl/options-ssl-nginx.conf:/etc/ansible-iot/ssl/options-ssl-nginx.conf"
      - "/etc/ansible-iot/ssl/ssl-dhparams.pem:/etc/ansible-iot/ssl/ssl-dhparams.pem"
{% include docker_compose_web_additional_volumes ignore missing %}

{% if not iotServices.core.enable %}
      - "${INSTALL_DIR}/web/etc/dnsmasq.d:/etc/dnsmasq.d"
    extra_hosts:
      - core:host-gateway
{% endif %}
    ports:
      - "${WEB_HTTP_PORT}:${WEB_HTTP_PORT}"
      - "${WEB_HTTPS_PORT}:${WEB_HTTPS_PORT}"
{% include docker_compose_web_additional_ports ignore missing %}

```

```
    restart: unless-stopped
{% endif %}

networks:
  default:
    name: iot${NETWORK_NAME_SUFFIX}
    external: true
```

3. Внесите новые пути до ssl, объявленные в шаге 2:

```
ansible-iot/templates/iot/web/base_config.j2
```

Содержимое конфигурационного файла /web/base_config.j2

```
server {
    listen {{ web.httpsPort }} ssl;

    ssl_certificate /etc/ansible-iot/ssl/домен/fullchain.pem;
    ssl_certificate_key /etc/ansible-iot/ssl/домен/privkey.pem;
    include /etc/ansible-iot/ssl/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/ansible-iot/ssl/ssl-dhparams.pem; # managed by Certbot
....
```

4. Добавьте свои сертификаты/ключи в /etc/ansible-iot/ssl/

5. Определите тип сертификата.

⚠ Ваш сертификат должен иметь формат PKCS#8.

Способ определения типа сертификата

```
# head -1 /tmp/eltex-sc-api.key
-----BEGIN PRIVATE KEY-----

# head -1 /storage/iot/ssl/private/eltex-sc-api.key
-----BEGIN RSA PRIVATE KEY-----
```

- BEGIN PRIVATE KEY — кодировка PKCS#8;
- BEGIN RSA PRIVATE KEY — кодировка PKCS#1.

6. Если ваш сертификат имеет кодировку PKCS#1, необходимо конвертировать его в PKCS#8.

Пример конвертирования сертификата из PKCS#1 в PKCS#8, letsencrypt:

```
openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in путь_до_ключа -out
имя_для_нового_ключа.
```

8 Настройка CAPTCHA

Сервис CAPTCHA используется для ограничения нежелательной активности в целях повышения устойчивости системы. В частности, снижается вероятность спам-атак платформы ботами.

Настройки доступны в файле **/etc/ansible-iot-1.35/vars/service_parametrs.yml**

Параметр **caseSensitive** отвечает за чувствительность к регистру.

Настройки уровня сложности представлены тремя возможными значениями параметра **captchaLevel** и доступны в файле **/etc/ansible-iot-1.35/vars/default.yml**

Уровень сложности	Описание
easy	Низкий уровень сложности. Обычно представляет из себя легкочитаемый текст.
medium	Средний уровень сложности. Обычно представляет из себя слегка искаженный текст.
hard	Высокий уровень сложности. Обычно представляет собой сильно искаженный текст.

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