



Платформа
ELIS

Установка платформы
Версия: 1.36

Содержание

1 Рекомендуемые характеристики сервера.....	3
2 Установка платформы ELIS	4
3 Установка платформы ELIS с внешней БД MongoDB	35
4 Работа с контейнерами	38
5 Расположение конфигурационных файлов	39
6 Проверка работоспособности платформы	40
7 Переход с http на https	42
8 Настройка CAPTCHA.....	45

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 сервера¹:

- 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.

 ¹Минимально необходимая конфигурация для запуска ядра платформы с микросервисами.

2 Установка платформы ELIS

В инструкции описан процесс установки платформы ELIS версии 1.36 на операционную систему Ubuntu 20.04. Информацию по установке более ранних версий платформы (1.35 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 разворачиванием 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 разворачивания 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

```
1 apt update
2 apt install --install-recommends linux-generic-hwe-20.04-edge
3 apt install software-properties-common
4 add-apt-repository --yes --update ppa:ansible/ansible
5 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	
1	<code>ansible-galaxy collection install community.general</code>
2	<code>ansible-galaxy collection install community.crypto</code>
3	<code>ansible-galaxy collection install community.docker</code>

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
<code>tar -C /etc -xvf ansible-iot-1.36.tar.gz</code>

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			
1	[iot]		
2	localhost	ansible_connection=local	ansible_sudo_pass=rootpasswd
3	[mqtt_broker]		
4	localhost	ansible_connection=local	ansible_sudo_pass=rootpassword
5	[elk]		
6	localhost	ansible_connection=local	ansible_sudo_pass=rootpasswd
7	[monitoring]		
8	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

```

1  ---
2  # Параметры установки платформы.
3  iot:
4      # Имя (IP-адрес) сервера, на котором будет производиться развертывание платформы
      IoT.
5      # ВАЖНО!!! В 'serverName' нужно прописывать то имя (IP-адрес), по которому будет
      доступны платформа.
6      serverName: "external.iot.address"
7      # Содержит путь до директории, в которую будет произведена установка.
8      installDir: /storage/iot
9      # Ссылка на политику конфиденциальности
10     privacyPolicyUrl: "my.privacy.policy"
11     # Ссылка на базу знаний
12     knowledgeBaseUrl: "https://docs.eltex-co.ru/display/EKB/IoT"
13
14     # Параметры MongoDB.
15     mongodb:
16         # Версия MongoDB. На старом железе, не поддерживающем оптимизацию, нужно
      выставить значение `4`.
17         version: 6
18         external:
19             # Если выставлен в true, будет использоваться внешняя MongoDB.
20             # ВАЖНО!!! MongoDB должна быть настроена, а параметры подключения нужно
      указать в 'uri'.
21             enable: false
22             # URI внешней MongoDB.
23             uri: "mongodb://external.mongodb.address:27017"
24
25     # Параметры WEB.
26     web:
27         # Порт HTTP, по которому будет осуществляться доступ в WEB.
28         httpPort: 80
29         # Порт HTTPS, по которому будет осуществляться доступ в WEB.
30         httpsPort: 443
31         # Автоматически перенаправлять запросы по порту HTTP на порт HTTPS
32         redirectHttpToHttps: true
33         # Нужно ли использовать HTTPS при формировании ссылок к WEB ('true' по
      умолчанию, при этом будет использован порт,
34         # указанный в 'iot.web.httpsPort'). Если поставить в 'false', будет использован
      HTTP и порт,
35         # указанный в 'iot.web.httpPort'.
36         useHttpsByDefault: true
37         nginx:
38             # Максимальное число соединений, которые одновременно может открыть рабочий
      процесс
39             worker_connections: 2048
40             # Ограничение скорости обработки запросов модулем Nginx Rate Limiting
41             rateLimit:
42                 enable: true
43         certbot:
44             # Использовать ли certbot для получения сертификатов Let's Encrypt.
45             enable: false
46             # Email владельца домена. Необходим для подтверждения валидности домена при
      получении сертификата Let's Encrypt.
47             email: test@email.com
48             fail2ban:

```

```

49     enable: true
50
51 # Параметры сервера отправки email.
52 mail:
53     smtp:
54         submitter: test@email.com
55         password: "password"
56         senderPrefix: "Сервер Eltex-SC"
57         auth: "true"
58         host: email.com
59         port: 587
60     # Протокол шифрования, используемый при подключении к серверу. Допустимые
        значения: none, starttls, ssl.
61     protection: starttls
62     push:
63         firebase:
64             enabled: false
65         apns:
66             enabled: false
67         hms:
68             enabled: false
69
70
71 # Параметры установки сервиса платежей для тарифов.
72 payment:
73     logLevel: INFO
74
75 # Параметры authorization server.
76 authorization:
77     # Уровень отладки внутри IoT Authorization Server.
78     logLevel: INFO
79
80     # Уровень сложности капчи: easy, medium, hard
81     captchaLevel: "easy"
82
83 # Параметры для управления доступностью саморегистрации.
84 selfRegistration:
85     allow: true
86     allowDemo: true
87     allowSocialNetworks: false
88
89 # Параметры клиентских регистраций (через соцсети).
90 clientRegistrations:
91     google:
92         enable: true
93         clientId: "GoogleClientIdChangeMe"
94         clientSecret: "GoogleClientSecretChangeMe"
95     microsoft:
96         enable: true
97         clientId: "MicrosoftClientIdChangeMe"
98         clientSecret: "MicrosoftClientSecretChangeMe"
99     apple:
100         enable: true
101         clientId: "AppleClientIdChangeMe"
102         keyId: "AppleKeyIdChangeMe"
103         teamId: "AppleTeamIdChangeMe"
104     yandex:
105         enable: true
106         clientId: "YandexClientIdChangeMe"
107         clientSecret: "YandexClientSecretChangeMe"

```



```

108     vk:
109         enable: true
110         clientId: "VkClientIdChangeMe"
111         clientSecret: "VkClientSecretChangeMe"
112     mailRu:
113         enable: true
114         clientId: "MailRuClientIdChangeMe"
115         clientSecret: "MailRuClientSecretChangeMe"
116
117     skills:
118         # Параметры навыка Яндекс для интеграции с Умным домом (Алисой). Отображается
119         в карточке навыка.
120         yandex:
121             enabled: false
122             # Параметры для Basic Authentication.
123             clientId: "YandexClientIdChangeMe"
124             password: "PasswordChangeMe"
125             # Id навыка, который необходимо указывать при отправке уведомлений.
126             skillId: ""
127             # OAuth-токен, который необходимо указывать при отправке уведомлений.
128             oauthToken: ""
129
130         # Параметры проекта умного дома Сбера для интеграции с Салютом. Отображается в
131         карточке проекта.
132         sber:
133             enabled: false
134             # Параметры для Basic Authentication.
135             clientId: "SberClientIdChangeMe"
136             password: "PasswordChangeMe"
137             # Bearer-токен, который необходимо указывать при отправке уведомлений.
138             bearerToken: ""
139
140         # Параметры проекта умного дома Mail.ru для интеграции с Марусей. Отображается
141         в карточке проекта/приложения.
142         marusya:
143             enabled: false
144             # Параметры для Basic Authentication.
145             clientId: "MarusyaClientIdChangeMe"
146             password: "PasswordChangeMe"
147             # App ID, который был назначен приложению VK при создании.
148             appId: "MarusyaAppIdChangeMe"
149             # OAuth-токен, который необходимо указывать при отправке уведомлений.
150             oauthToken: ""
151
152     # Параметры платформы IoT core.
153     core:
154         # Уровень отладки внутри IoT Core.
155         logLevel: INFO
156
157     # Порты платформы для подключения zway-контроллеров.
158     ctlGate:
159         port: 8070
160         tcpPort: 8069
161         sslPort: 8072
162
163     links:
164         # Нужно ли использовать HTTPS при формировании ссылок к ресурсам самой
165         платформы (например, прошивки).
166         useHttpsForApi: false

```

```

163     # Нужно ли использовать HTTPS при формировании ссылок на фото с камер
наблюдения.
164     useHttpsForCameraLinks: true
165     # Нужно ли использовать 'iot.web.httpPort'/'iot.web.httpsPort' при
формировании ссылок к API.
166     useUiProxyForApi: true
167
168     # Параметры для работы с видеосерверами
169     video:
170     # Параметры Flussonic.
171     flussonic:
172         url: ""
173         apiKey: ""
174         operatorId: ""
175         adminLogin: ""
176         motion:
177             enabled: false
178     # Параметры видеосервера EVI
179     evi:
180         url: ""
181         apiKey: ""
182         operatorId: ""
183         adminLogin: ""
184
185     # Настройки ИК-пульта.
186     irc:
187     # Время ожидания ИК команды от пользователя
188         recTimeout: 15s
189     # Таймаут записи команды (отсутствия фронтов)
190         cmdTimeout: 100ms
191
192     # Настройки охраны.
193     guard:
194     # Время, которое дается на включение FLIRS устройств (ждем подтверждение от
контроллера),
195     # при постановке на охрану.
196         deviceRequestDelay: 15s
197     # Время, которое прибавляется ко времени задержки на очистку охранного кэша
при постановке/снятии с охраны,
198     # на случай если охрана не завершила процесс постановки/снятия.
199     # Время задержки формируется как количество охранных устройств умноженное на
deviceRequestDelay.
200         clearContextExtraCacheDelay: 1m
201
202     # Параметры установки сервисов логирования (Elasticsearch + Logstash + Kibana).
203     elk:
204     # Нужно ли добавлять в платформу appender, отправляющий логи в logstash.
205     # В нем нет необходимости, если ELK не развернут или не настроен; это лишь
спровоцирует сообщения об ошибках отправки
206     # в логах платформы.
207         enable: false
208     # Имя (IP-адрес) сервера, на котором будет развернут ELK.
209     # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
платформой (на том же хосте).
210     # В таком случае хосты в инвентаре в группах [iot] и [elk] должны совпадать.
211         serverName: "{{ iot.serverName }}"
212     # Директория для установки системы логирования.
213         installDir: /storage/elk
214
215     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.

hms : enabled: false	Включение/выключение push-сообщений для Huawei.
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

```

1  ---
2  # Экземпляр платформы с параметрами по умолчанию.
3  instance: "IoT-instance-default"
4
5  # Версия контейнеров.
6  release: 1.36
7
8  # Имя репозитория docker registry, содержащего docker-образы для развертывания.
9  registry: hub.eltex-co.ru
10
11 # Список сервисов для перезапуска (при запуске плейбуков restart_*.yml).
12 # Можно оставить пустым, а при запуске передавать параметром командной строки.
13 services: []
14
15 # Нужно ли выполнять подготовку дистрибутива к установке. Этот шаг полезен при
16 # "чистой" установке
17 # на только что созданный сервер. Если ранее уже была выполнена установка
18 # компонентов IoT через ansible,
19 # то такая подготовка не требуется и этот шаг можно пропустить для экономии времени.
20 withDistroPreparingStep: true
21
22 # Суффикс, добавляемый к имени каждого контейнера (помогает избежать конфликты имен
23 # контейнеров).
24 containerNameSuffix: ""
25
26 # Суффикс, добавляемый к имени создаваемой сети docker (помогает избежать конфликты
27 # имен сетей docker).
28 networkNameSuffix: ""
29
30 # Параметры сервисов IoT (для docker-compose), сгруппированные по именам.
31 # 'enable' - должен ли присутствовать сервис в docker-compose.yml.
32 # 'port.map' - номер порта сервиса в сети хоста.
33 # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
34 # 'db.name' - имя БД, используемой сервисом (связкой сервисов).
35 iot:
36   swarm:
37     enabled: false
38     # Имя stack-а для запуска в docker-swarm.
39     stack: swarm_iot
40   # Параметры мониторинга использования дискового пространства. Должны
41   # соответствовать требованию:
42   # warnThreshold > criticalThreshold > 0, иначе мониторинг дискового пространства
43   # будет отключен.
44   diskUsage:
45     # Порог дискового пространства (в %), при достижении которого все логики
46     # микросервисов переводятся в режим WARN
47     # (отображаются сообщения с тегами WARN и ERROR). Количество бэкапов баз
48     # уменьшается пропорционально приближению к
49     # порогу criticalThreshold.
50     warnThreshold: 20
51     # Порог дискового пространства (в %), при достижении которого все логики
52     # микросервисов переводятся в режим ERROR
53     # (отображаются только сообщения с тегом ERROR). Бэкапы баз не выполняются.
54     criticalThreshold: 10
55   # использовать сервис Notification для отправки пушей, вместо старой логики
56   useNotification: true

```

```

48 # Нужно ли установить лимиты на сервисы в docker compose
49 limits:
50     enable: false
51 mongodb:
52     replication:
53         label: has-mongodb
54     limits:
55         enable: false
56         cpus: 1.0
57         memory: 4G
58     port:
59         map: 27017
60         export: false
61 broker:
62     logLevel: INFO
63     replication:
64         label: has-broker
65     limits:
66         enable: true
67         cpus: 1.0
68         memory: 4G
69     external:
70         # Будет ли развернут брокер на стороннем хосте.
71         enable: false
72         # Нужно ли устанавливать брокер на сторонний хост средствами ansible.
73         install: false
74         # Если установка брокера на сторонний хост будет выполняться средствами
75         # ansible, то в какую директорию.
76         installDir: /storage/broker
77         # Внешний адрес стороннего хоста, по которому к брокеру будут подключаться
78         # платформа и устройства.
79         host: "external.broker.address"
80         port:
81             map: 8883
82     internal:
83         port:
84             map: 8083
85             export: false
86 db:
87     name: iot-broker
88 jconsole:
89     enable: false
90     port: 32002
91     # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
92     # При развертывании докера - здесь указывается адреса хоста, где работает
93     докер.
94     host: "10.20.30.40"
95     hivemq:
96         rootFolder: /hivemq/
97     olapservice:
98         logLevel: INFO
99         replication:
100             label: has-olap
101             limits:
102                 enable: true
103                 cpus: 1.0
104                 memory: 4G
105     external:
106         # Будет ли развернут olapservice на стороннем хосте.
107         enable: false

```

```

105     # Нужно ли устанавливать olapservice на сторонний хост средствами ansible.
106     install: false
107     # Если установка olapservice на сторонний хост будет выполняться средствами
108     # ansible, то в какую директорию.
109     installDir: /storage/olapservice
110     # Внешний адрес стороннего хоста, по которому платформа будет подключаться к
111     # olapservice'y.
112     host: "external.olapservice.address"
113     port:
114     map: 8023
115     export: false
116     db:
117     name: iotcore
118     # Нужно ли устанавливать clickhouse (полезно для разработки olapservice).
119     install: true
120     replication:
121     label: has-clickhouse
122     limits:
123     enable: false
124     cpus: 1.0
125     memory: 4G
126     port:
127     map: 8123
128     export: false
129     captcha:
130     enable: true
131     replication:
132     label: has-captcha
133     limits:
134     enable: true
135     cpus: 1.0
136     memory: 4G
137     port:
138     map: 8088
139     export: false
140     caseSensitive: true
141     allowedSizes:
142     - "312x45"
143     - "270x40"
144     instance: "captcha:8088"
145     proportion: 100
146     zscaptcha:
147     enable: false
148     replication:
149     label: has-captcha
150     limits:
151     enable: true
152     cpus: 1.0
153     memory: 4G
154     port:
155     map: 8089
156     export: false
157     caseSensitive: true
158     instance: "zs-captcha:8089"
159     proportion: 0
160     authorization:
161     # Развертывание окружения без auth server, полезно для разработки auth server.
162     enable: true
163     # API-ключ для авторизации HTTP-запросов в сервис авторизации от внешних систем,
164     # таких как Core.

```

```

162     # Специальное значение 'autogenerated' означает, что apiKey будет сгенерирован в
процессе установки.
163     apiKey: "autogenerated"
164     replication:
165         label: has-authorization-server
166     limits:
167         enable: true
168         cpus: 1.0
169         memory: 4G
170     external:
171         host: "external.authorization.address"
172     port:
173         map: 8091
174         export: false
175     ssl:
176         map: 8093
177         export: false
178     db:
179         name: iot-authorization-server
180     jconsole:
181         enable: false
182         port: 32003
183         # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
184         # При разворачивании докера - здесь указывается адреса хоста, где работает
докер.
185         host: "10.20.30.40"
186     email:
187         blacklist:
188             # "0 * * * * ?" раз в минуту
189             # "0 ${random.int[60]} ${random.int[1,5]} 1 * ?" раз в неделю
190             cron: "0 ${random.int[60]} ${random.int[1,5]} 1 * ?"
191     swagger:
192         enable: false
193     rabbit:
194         eventQueue:
195             name: platform-rpc-event-queue
196             # Количество консьюмеров на стороне платформы
197             platformConsumers:
198                 # Данный параметр не может быть больше чем maxCount.
199                 count: 16
200                 # Максимальное количество одновременных консьюмеров очереди.
201                 maxCount: 16
202             # Настройки сервиса уведомлений
203             notification:
204                 # Количество консьюмеров
205                 count: 1
206                 #Максимальное количество консьюмеров
207                 maxCount: 5
208                 #Наименования обменника
209                 exchange: notification-exchange
210                 #Ключ маршрута для уведомлений
211                 envelopeRoutingKey: notification
212                 #Ключ маршрута событий входа/выхода
213                 userEventRoutingKey: userevent
214     payment:
215         enable: true
216         # ApiKey используется для доступа к API платежного сервиса.
217         apiKey: "CHANGE_ME"
218     replication:
219         label: has-payment-server

```

```

220     limits:
221         enable: true
222         cpus: 1.0
223         memory: 4G
224     port:
225         map: 9091
226         export: false
227     ssl:
228         map: 9093
229         # Если экспорт false, то запросы будут проксироваться через Nginx. Для
        тестирования флаг можно включать.
230         # Если флаг true, то запросы должны ходить на прямую в payment service
231         export: false
232     internal:
233         # Максимальный временной период, через который будет повторяться запрос статуса
        платежа на платежный сервис.
234         orderCheckTimePeriodMax: "30m"
235         # Максимальное время, через которое сервис перестает опрашивать ордер.
236         orderCheckTimeTotal: "24h"
237         # Промежуток времени, через который повторяется отправка сообщения на
        платформу об успешной оплате,
238         # чтобы платформа создала подписку. Повторная отправка отменяется если
        платформа прислала сообщение о создании
239         # подписки или если количество попыток передать сообщение превысило параметр
        checkSubscriptionCreatedMaxAttempts
240         checkSubscriptionCreatedPeriodTime: "5m"
241         # Количество попыток передать платформе сообщение об успешной оплате.
242         checkSubscriptionCreatedMaxAttempts: 10
243     db:
244         name: iot-payment-server
245     jconsole:
246         enable: false
247         port: 32004
248         # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
249         # При разворачивании докера - здесь указывается адреса хоста, где работает
        докер.
250         host: "10.20.30.40"
251     swagger:
252         enable: false
253     paySystem:
254         # Необходимо указать эндпоинт на стороне IoT-Core куда будут
        переадресовываться запросы с ЮКассы.
255         # МП приложения могут сами распознать, что вебвью с ЮКассой закрыта, и
        кидать необходимый запрос на Iot-core
256         # и переходить на нужный экран. Веб так не умеет, поэтому создан этот
        параметр. Когда веб поддержит покупку
257         # тарифов, нужно будет уточнить на какую именно страницу нужно делать
        редирект.
258         redirectUrlAfterPay: "https://eltexhome.ru"
259     yookassa:
260         url: "https://api.yookassa.ru/v3"
261         # Идентификатор магазина.
262         shopId: ""
263         secretKey: ""
264         # Список доступных IP адресов запросов от Юкассy. Взято из официально
        документации.
265         # https://yookassa.ru/developers/using-api/webhooks#ip
266         availableHosts:
267             - "185.71.76.0/27"
268             - "185.71.77.0/27"

```

```

269         - "77.75.153.0/25"
270         - "77.75.156.11"
271         - "77.75.156.35"
272         - "77.75.154.128/25"
273         - "2a02:5180::/32"
274     rabbit:
275         queue:
276             # Очередь для приема сообщений от платформы.
277             paymentReceive: "paymentReceive"
278             # Очередь для приема сообщений от платформы и синхронного ответа во
временную очередь.
279             paymentReceiveAndReply: "paymentReceiveAndReply"
280             # Время ожидания синхронного ответа от PaymentServer на сообщение,
отправленное в очередь
281             # paymentReceiveAndReply
282             paymentReceiveAndReplyQueueTimeOut: "30s"
283             # Очередь для отправки сообщений на платформу.
284             platformReceive: "platformReceive"
285         consumers:
286             # Количество одновременных консьюмеров в Iot-Core для очереди
platformReceive.
287             # Данный параметр не может быть больше чем maxCount.
288             coreCount: 1
289             # Максимальное количество одновременных консьюмеров в Iot-Core для очереди
platformReceive.
290             coreMaxCount: 1
291             # Количество одновременных консьюмеров в Iot-Payment-Server для очереди
paymentReceiveAndReply.
292             paymentCount: 10
293             # Максимальное количество одновременных консьюмеров в Iot-Payment-Server для
очереди paymentReceiveAndReply.
294             paymentMaxCount: 20
295     core:
296         # Развертывание окружения без платформы, полезно для разработки core.
297         enable: true
298         replication:
299             replicaCount: 1
300             label: has-core
301         limits:
302             enable: true
303             cpus: 1.0
304             memory: 4G
305     port:
306         map: 8071
307         export: false
308     ssl:
309         map: 8073
310         export: false
311     db:
312         name: iot-core
313         # Имена устаревших БД в MongoDB, используемые платформой.
314         deprecated:
315             fs:
316                 name: iot-fs
317             licenses:
318                 name: iot-licenses
319             events:
320                 name: iot-events
321     jconsole:
322         enable: false

```

```

323     port: 32001
324     # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
325     # При разворачивании докера - здесь указывается адреса хоста, где работает
докер.
326     host: "10.20.30.40"
327     swagger:
328       enable: false
329     broker:
330       # Нужно ли платформе пытаться подключиться к MQTT Broker.
331       enable: true
332       threadPools:
333         main:
334           size: 16
335     olapservice:
336       # Нужно ли платформе пытаться подключиться к OlapService.
337       enable: true
338     mjollnir:
339       # URL для Mjollnir.
340       url: "http://smart.eltex-co.ru:8078/api/v1"
341     slgate:
342       instances:
343         - slgate:4443
344       ignoreServerCertCheck: false
345     video:
346       evi:
347         ignoreServerCertCheck: false
348         checkEventSenderAddress: true
349       # Параметры WebRTC, которые необходимы камере для формирования своих ICE-
кандидатов.
350       webrtc:
351         # Параметры STUN-сервера.
352         stun:
353           # URL STUN-серверов.
354           urls:
355             - "stun.example1.com:3478"
356             - "stun.example2.com:3478"
357         # Параметры TURN-сервера.
358         turn:
359           # Параметры TURN:SFU
360           sfu:
361             # Принудительное включение
362             force: false
363             # Таймаут на ожидание ивента от EVI о начале webrtc-сессии между камерой
и TURN:SFU
364             eventTimeout: 10s
365           # Параметры P2P
366           p2p:
367             # Таймаут на ожидание SDP-answer от камеры
368             sdpAnswerTimeout: 60s
369             # Время ожидания проверки камерой типа своего NAT
370             checkCameraNatTypeTimeout: 10s
371             # Время выполнения ping к host candidate клиента
372             pingHostCandidateTimeout: 10s
373       rabbit:
374         notification:
375           exchange: notification-exchange
376           envelopeRoutingKey: notification
377           userEventRoutingKey: userEvent
378           count: 1
379           maxCount: 1

```

```

380 web:
381     # Развертывание окружения без WEB, полезно для разработки web.
382     enable: true
383     replication:
384         label: has-web
385     limits:
386         enable: true
387         cpus: 1.0
388         memory: 4G
389     hazelcast:
390         clusterName: iot-core
391         instanceName: iot-core-hazelcast-instance
392         replication:
393             label: has-hazelcast
394         limits:
395             enable: true
396             cpus: 1.0
397             memory: 4G
398         external:
399             port:
400                 map: 5701
401                 export: false
402     rabbitmq:
403         replication:
404             label: has-rabbitmq
405         limits:
406             enable: true
407             cpus: 1.0
408             memory: 4G
409         admin:
410             user: guest
411             password: guest
412             port:
413                 map: 15671
414                 export: false
415             sslPort:
416                 map: 15672
417                 export: false
418         port:
419             map: 5672
420             export: false
421     zwayproxy:
422         logLevel: INFO
423         external:
424             enable: false
425         # Если установка zwayproxy на сторонний хост будет выполняться средствами
426         ansible, то в какую директорию.
427         installDir: /storage/zwayproxy
428         replication:
429             replicaCount: 2
430             label: has-zwayproxy
431         limits:
432             enable: true
433             cpus: 1.0
434             memory: 4G
435         port:
436             map: 8070
437             export: true
438         sslPort:
439             map: 8072

```



```

439     export: true
440 rabbit:
441     queue:
442         # Идентификатор очереди для отправки сообщений из прокси в платформу.
443         platform: zway-proxy-platform
444     exchange:
445         # Идентификатор топик-коллектора.
446         proxy: zway-proxy-topic-exchange
447     # Количество консьюмеров на стороне платформы
448     platformConsumers:
449         # Данный параметр не может быть больше чем maxCount.
450         count: 16
451         # Максимальное количество одновременных консьюмеров очереди.
452         maxCount: 16
453     # Количество консьюмеров на стороне zwayProxy service
454     proxyConsumers:
455         # Количество одновременных консьюмеров очереди. Данный параметр не может
456         # быть больше чем maxCount.
457         count: 1
458         # Максимальное количество одновременных консьюмеров очереди.
459         maxCount: 1
460 wsproxy:
461     logLevel: INFO
462     maxConnections: 25000
463     external:
464         enable: false
465         # Если установка wsproxy на сторонний хост будет выполняться средствами
466         # ansible, то в какую директорию.
467         installDir: /storage/wsproxy
468         # Адрес хоста куда nginx будет проксировать запросы.
469         host: "1.2.3.4"
470         port: 8075
471     replication:
472         replicaCount: 2
473         label: has-wsproxy
474     limits:
475         enable: true
476         cpus: 1.0
477         memory: 4G
478     port:
479         map: 8075
480         export: true
481 rabbit:
482     queue:
483         # Идентификатор очереди для отправки сообщений из прокси в платформу.
484         platform: ws-proxy-platform
485     exchange:
486         # Идентификатор топик-коллектора.
487         proxy: ws-proxy-topic-exchange
488     # Количество консьюмеров на стороне платформы
489     platformConsumers:
490         # Данный параметр не может быть больше чем maxCount.
491         count: 16
492         # Максимальное количество одновременных консьюмеров очереди.
493         maxCount: 16
494     # Количество консьюмеров на стороне wsProxy service
495     proxyConsumers:
496         # Количество одновременных консьюмеров очереди. Данный параметр не может
497         # быть больше чем maxCount.
498         count: 16

```

```

496         # Максимальное количество одновременных консьюмеров очереди.
497         maxCount: 16
498     # Параметры установки сервисов мониторинга (Prometheus + Grafana).
499     monitoring:
500         external:
501             # Будет ли развернут monitoring на стороннем хосте.
502             enable: true
503             # Нужно ли устанавливать monitoring на сторонний хост средствами ansible.
504             install: false
505             # Если установка monitoring на сторонний хост будет выполняться средствами
506             # ansible, то в какую директорию.
507             installDir: /storage/monitoring
508         mongodbExporter:
509             replication:
510                 label: has-mongodb-exporter
511             limits:
512                 enable: false
513                 cpus: 1.0
514                 memory: 4G
515             enable: true
516             port:
517                 map: 9216
518                 export: false
519         nginxExporter:
520             replication:
521                 label: has-nginx-exporter
522             limits:
523                 enable: false
524                 cpus: 1.0
525                 memory: 4G
526             enable: true
527             port:
528                 map: 9113
529                 export: false
530         prometheus:
531             replication:
532                 label: has-prometheus
533             limits:
534                 enable: false
535                 cpus: 1.0
536                 memory: 4G
537             port:
538                 map: 9090
539                 export: false
540         # Период опроса сервисов со стороны Prometheus в секундах. Небольшие значения
541         # (менее десятков секунд) значительно
542         # увеличат нагрузку на сеть.
543         # При изменении параметра рекомендуется поменять переменную
544         grafana.scrapeInterval на значение не ниже указанного
545         # для Prometheus, чтобы не терять метрики.
546         scrapeInterval: 60
547     grafana:
548         replication:
549             label: has-grafana
550         limits:
551             enable: false
552             cpus: 1.0
553             memory: 4G
554         port:
555             map: 3000

```

```

553     # Период опроса Prometheus со стороны Grafana в секундах.
554     scrapeInterval: 60
555   notification:
556     enable: true
557     logLevel: INFO
558     replication:
559       label: has-notification
560     limits:
561       enable: true
562       cpus: 1.0
563       memory: 4G
564     jconsole:
565       enable: false
566       port: 32002
567     # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
568     # При разворачивании докера - здесь указывается адреса хоста, где работает
докер.
569     host: "10.20.30.40"
570     port:
571       export: false
572       map: 9095
573     rabbit:
574       exchange: notification-exchange
575       envelopeRoutingKey: notification
576       userEventRoutingKey: userevent
577     vault:
578       uri: https://127.0.0.1:8200
579       token: RootToken
580       ssl:
581         useTrustStore: false
582         trustStore: file:///etc/vault.jks
583         trustStoreType: jks
584         trustStorePassword: vault0
585       rootPath: credentials
586     push:
587       fcm:
588         # Период между отправкой пушей, для достижения ограничения частоты отправки
пушей.
589         delay: 250ms
590     hazelcast:
591       instance: iot-notification
592       cluster: iot-notification
593       addresses: "notification-hazelcast:5701"
594       replication:
595         label: has-notification-hazelcast
596       limits:
597         enable: true
598         memory: 1G
599         cpus: 1.0
600     external:
601       port:
602         enable: false
603         map: 15701
604   slgate:
605     db:
606       name: "activator"
607       user: "activator"
608       password: "activator"
609       port:

```

```

611     map: 5432
612     export: false
613   vpn:
614     port:
615       map: 1194
616     protocol: udp
617     maxClients: 1024
618   easysrsa:
619     req:
620       country: "RU"
621       province: "Novosibirsk"
622       city: "Novosibirsk"
623       organization: "Eltex"
624       organizationUnit: "EMS"
625       email: "eltex@eltex-co.ru"
626   activator:
627     port:
628       map: 8899
629     export: false
630   proxy:
631     scheme: "https://"
632     useClientCert: false
633     clientCertPassword: "CHANGE_ME"
634     port:
635       map: 4443
636
637   # Параметры сервисов ELK (для docker-compose), сгруппированные по именам.
638   # 'port.map' - номер порта сервиса в сети хоста.
639   # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
640   elk:
641     elasticsearch:
642       rest:
643         port:
644           map: 9200
645       nodes:
646         port:
647           map: 9300
648     logstash:
649       port:
650         map: 5001
651       api:
652         port:
653           map: 9600
654     kibana:
655       port:
656         map: 5601
657     fluentBit:
658       port:
659         map: 24224
660     #Индексы, создаваемые в elasticsearch со стороны fluentBit
661     targets:
662       - core
663       - authorization
664       - notification
665       - payment
666       - zwayproxy
667       - wsproxy
668       - mqttbroker
669       - olapservice

```

Таблица описания значений в файле настроек vars/service_parameters.yml

release: 1.36	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 логи 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 логи 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.
payment: enable: true	Сервис платежей. 'enable' — параметр, отражающий необходимость присутствия сервиса в docker-compose.yml.
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.
notification: enable: true	Сервис для уведомлений. 'enable' — параметр, отражающий необходимость присутствия сервиса в docker-compose.yml.

vault: uri: https://127.0.0.1:8200 token: RootToken ssl: useTrustStore: false trustStore: file:///etc/vault.jks trustStoreType: jks trustStorePassword: vault0	Настройки хранилища секретов для отправки уведомлений.
rootPath: credentials	Путь до секретов в хранилище секретов.

7. После этого можно запустить установку:

Установка	
1	cd /etc/ansible-iot-1.36
2	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:1.36	/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:1.36	/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	hub.eltex.loc:9017/iot-authorization-server:1.36	/extras/app-starter	2 minutes ago	Up 2 minutes		iot-authorization-server
1131ade6b4e4	hub.eltex-co.ru/iot-notification:1.36	/extras/app-starter...	2 minutes ago	Up 2 minutes		iot-notification
07d1f93831bd	hub.eltex-co.ru/iot-mqttbroker-mongo:1.36	java -cp @/app/jib-...	2 minutes ago	Up 2 minutes	0.0.0.0:8883->8883/tcp, :::8883->8883/tcp	iot-mqtt-broker
c3f94517f770	hub.eltex-co.ru/iot-minio:1.36	/entrypoint.sh serv...	2 minutes ago	Up 2 minutes		iot-minio
ea794e1690c0	hub.eltex-co.ru/iot-payment:1.36	/extras/app-starter...	2 minutes ago	Up 2 minutes		iot-payment
8f8401f5ae0a	hub.eltex.loc:9017/iot-hazelcast:1.36	java --add-modules ...	2 minutes ago	Up 2 minutes		iot-notification-hazelcast
8f8401f5ae0a	hub.eltex.loc:9017/iot-hazelcast:1.36	java --add-modules ...	2 minutes ago	Up 2 minutes		iot-hazelcast
228d41c96cba	hub.eltex-co.ru/iot-olapervice:1.36	java -cp @/app/jib-...	2 minutes ago	Up 2 minutes		iot-olapervice
e8e2899f2c8d	hub.eltex-co.ru/iot-captcha:1.36	java -jar LibreCapt...	2 minutes ago	Up 2 minutes	8888/tcp	iot-captcha

57c02941cc4f	hub.eltex-co.ru/ iot-mongo6:1.36	/entrypoint.sh	2 minutes ago	Up 2 minutes	0.0.0.0:27017->27017/ tcp, :::27017->27017/tcp	iot-mongo
7c3d8d5c4137	hub.eltex.loc:901 7/iot-clickhouse- server:1.36	docker- entrypoint.s...	2 minutes ago	Up 2 minutes	8123/tcp, 9000/tcp, 9009/ tcp	iot-iot- mysql
0327972ab361	hub.eltex.loc:901 7/iot- rabbitmq:1.36	docker- entrypoint.s...	2 minutes ago	Up 2 minutes	4369/tcp, 5671-5672/tcp, 15691-15692/tcp, 25672/ tcp	iot- rabbitmq:
a5bcb7e27390	hub.eltex.loc:901 7/iot- wsproxy:1.36	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:901 7/iot- zwayproxy:1.36	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- zwayprox y

- ✓ 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.

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**):

```

1  sudo apt-get update
2  sudo apt-get upgrade
3  sudo apt-get install -y software-properties-common gnupg build-essential net-tools
   dkms
4  wget https://www.mongodb.org/static/pgp/server-4.4.asc
5  sudo apt-key add server-4.4.asc
6  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
7  sudo apt-get update
8  sudo apt-get install -y mongodb-org
9  sudo service mongod start
10 sudo service mongod status

```

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

```

1  port: 27017
2  bindIp: 0.0.0.0

```

И перезапустите сервис mongod:

```

1  sudo service mongod restart
2  sudo service mongod status

```

3. Установите Ansible на сервер:

```

1  sudo add-apt-repository --yes --update ppa:ansible/ansible
2  sudo apt install ansible
3  sudo ansible-galaxy collection install community.general
4  sudo ansible-galaxy collection install community.crypto
5  sudo ansible-galaxy collection install community.docker

```

4. Подготовьте конфигурацию:

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 от **root** в переменной **ansible_sudo_pass**:

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

Содержимое файла inventory

1	[iot]		
2	localhost	ansible_connection=local	ansible_sudo_pass=rootpasswd
3	[mqtt_broker]		
4	localhost	ansible_connection=local	ansible_sudo_pass=rootpassword
5	[elk]		
6	localhost	ansible_connection=local	ansible_sudo_pass=rootpasswd
7	[monitoring]		
8	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-1.36/vars/default.yml** и **/etc/ansible-iot-1.36/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`

```

1  # Параметры MongoDB.
2  mongodb:
3      # Версия MongoDB. На старой аппаратной платформе, не поддерживающей оптимизацию,
      # нужно выставить значение `4`.
4      version: 6
5      external:
6          # Если выставлен в true, будет использоваться внешняя MongoDB.
7          # ВАЖНО!!! MongoDB должна быть настроена, а параметры подключения нужно
      # указать в 'uri'.
8          enable: false
9          # URI внешней MongoDB.
10         uri: "mongodb://external.mongodb.address:27017"
```

8. После этого можно запустить установку:

```
cd /etc/ansible-iot-1.36
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-1.36/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-1.36/templates** располагаются 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**

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-1.36/vars/default.yml**. При запуске платформы параметры, заданные в файле конфигурации **/etc/ansible-iot-1.36/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-сокеты этого контроллера недоступны, 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

```
1 {
2   "version" : "1.36-3477",
3   "api" : "1.0",
4   "currentTime" : "2022-07-25T09:24:12.544842Z[Etc/UTC]"
5 }
```

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

```

1  {% if iotServices.web.enable %}
2      web:
3          image: "${REGISTRY_IOT}/iot-double-web:${RELEASE}"
4          container_name: iot-double-web${CONTAINER_NAME_SUFFIX}
5  {% if limitsEnable and iotServices.web.limits.enable %}
6      deploy:
7          resources:
8              limits:
9                  cpus: '${WEB_LIMITS_CPUS}'
10                 memory: ${WEB_LIMITS_MEMORY}
11  {% endif %}
12  {% if iotServices.core.enable %}
13      depends_on:
14          core:
15              condition: service_started
16      links:
17          - core
18  {% endif %}
19      environment:
20          - ENABLE_CERTBOT=${WEB_ENABLE_CERTBOT}
21          - SERVER_NAME=${WEB_SERVER_NAME}
22          - CERTBOT_EMAIL=${WEB_CERTBOT_EMAIL}
23          - HTTPS_PORT=${WEB_HTTPS_PORT}
24          - TZ=${TIMEZONE}
25      volumes:
26          - "${INSTALL_DIR}/web/etc/nginx/nginx.conf:/etc/nginx/nginx.conf"
27          - "${INSTALL_DIR}/web/etc/nginx/conf.d/default.conf:/etc/nginx/conf.d/
28          default.conf"
29          - "${INSTALL_DIR}/web/well-known:/usr/share/eltex-sc-web/www/.well-known"
30  {% if web.certbot.enable %}
31          - "${INSTALL_DIR}/web/etc/letsencrypt:/etc/letsencrypt"
32          - "${INSTALL_DIR}/web/var/log/letsencrypt:/var/log/letsencrypt"
33  {% endif %}
34          - "${INSTALL_DIR}/ssl/certs/eltex-sc-web.crt:/etc/ssl/certs/eltex-sc-web.crt"
35          - "${INSTALL_DIR}/ssl/private/eltex-sc-web.key:/etc/ssl/private/eltex-sc-
36          web.key"
37          - "/etc/ansible-iot/ssl/домен/fullchain.pem:/etc/ansible-iot/ssl/домен/
38          fullchain.pem"
39          - "/etc/ansible-iot/ssl/домен/privkey.pem:/etc/ansible-iot/ssl/домен/
40          privkey.pem"
41          - "/etc/ansible-iot/ssl/options-ssl-nginx.conf:/etc/ansible-iot/ssl/options-
42          ssl-nginx.conf"
43          - "/etc/ansible-iot/ssl/ssl-dhparams.pem:/etc/ansible-iot/ssl/ssl-
44          dhparams.pem"
45  {% include docker_compose_web_additional_volumes ignore missing %}
46  {% if not iotServices.core.enable %}
47          - "${INSTALL_DIR}/web/etc/dnsmasq.d:/etc/dnsmasq.d"
48      extra_hosts:
49          - core:host-gateway
50  {% endif %}
51      ports:
52          - "${WEB_HTTP_PORT}:${WEB_HTTP_PORT}"
53          - "${WEB_HTTPS_PORT}:${WEB_HTTPS_PORT}"
54  {% include docker_compose_web_additional_ports ignore missing %}

```

```

51     restart: unless-stopped
52 {% endif %}
53
54 networks:
55     default:
56         name: iot${NETWORK_NAME_SUFFIX}
57         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

```

1  server {
2      listen {{ web.httpsPort }} ssl;
3
4      ssl_certificate /etc/ansible-iot/ssl/домен/fullchain.pem;
5      ssl_certificate_key /etc/ansible-iot/ssl/домен/privkey.pem;
6      include /etc/ansible-iot/ssl/options-ssl-nginx.conf; # managed by Certbot
7      ssl_dhparam /etc/ansible-iot/ssl/ssl-dhparams.pem; # managed by Certbot
8      ....

```

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-1.36/vars/service_params.yml**

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

Настройки уровня сложности представлены тремя возможными значениями параметра **captchaLevel** и доступны в файле **/etc/ansible-iot-1.36/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>