



Платформа

ELIS

Установка платформы

Версия: 1.34

Содержание

1 Рекомендуемые характеристики сервера.....	3
2 Установка платформы ELIS	4
3 Установка платформы ELIS с внешней БД MongoDB	31
4 Работа с контейнерами	34
5 Расположение конфигурационных файлов	35
6 Проверка работоспособности платформы	36
7 Переход с http на https	38
8 Настройка CAPTCHA.....	41

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*:

- 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.

 Минимально необходимая конфигурация для запуска ядра платформы с микросервисами.

2 Установка платформы ELIS

В инструкции описан процесс установки платформы ELIS версии 1.34 на операционную систему Ubuntu 20.04. Информацию по установке более ранних версий платформы (1.33 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

```
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 Элтэкс. 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-1.34.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-1.34`

5. Отредактируйте файл `/etc/ansible-iot-1.34/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=password
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-1.34/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 **server_name**:

- ❌ 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 'addr' и 'port'.
21      enable: false
22      # Адрес внешней MongoDB.
23      addr: "external.mongodb.address"
24      # Порт внешней MongoDB.
25      port: 27017
26
27  # Параметры WEB.
28  web:
29    # Порт HTTP, по которому будет осуществляться доступ в WEB.
30    httpPort: 80
31    # Порт HTTPS, по которому будет осуществляться доступ в WEB.
32    httpsPort: 443
33    # Автоматически перенаправлять запросы по порту HTTP на порт HTTPS
34    redirectHttpToHttps: true
35    # Нужно ли использовать HTTPS при формировании ссылок к WEB ('true' по
   умолчанию, при этом будет использован порт,
36    # указанный в 'iot.web.httpsPort'). Если поставить в 'false', будет использован
   HTTP и порт,
37    # указанный в 'iot.web.httpPort'.
38    useHttpsByDefault: true
39    nginx:
40      # Максимальное число соединений, которые одновременно может открыть рабочий
   процесс
41      worker_connections: 2048
42      # Ограничение скорости обработки запросов модулем Nginx Rate Limiting
43      rateLimit:
44        enable: true
45    certbot:
46      # Использовать ли certbot для получения сертификатов Let's Encrypt.
47      enable: false
48      # Email владельца домена. Необходим для подтверждения валидности домена при
   получении сертификата Let's Encrypt.

```

```
49     email: test@email.com
50     fail2ban:
51       enable: true
52
53     # Параметры сервера отправки email.
54     mail:
55       smtp:
56         submitter: test@email.com
57         password: "password"
58         senderPrefix: "Сервер Eltex-SC"
59         auth: "true"
60         host: email.com
61         port: 587
62         # Протокол шифрования, используемый при подключении к серверу. Допустимые
        значения: none, starttls, ssl.
63         protection: starttls
64
65     # Параметры authorization server.
66     authorization:
67       # Уровень отладки внутри IoT Authorization Server.
68       logLevel: INFO
69
70       # Уровень сложности капчи: easy, medium, hard
71       captchaLevel: "easy"
72
73     # Параметры для управления доступностью саморегистрации.
74     selfRegistration:
75       allow: true
76       allowDemo: true
77       allowSocialNetworks: false
78
79     # Параметры клиентских регистраций (через соцсети).
80     clientRegistrations:
81       google:
82         enable: true
83         clientId: "GoogleClientIdChangeMe"
84         clientSecret: "GoogleClientSecretChangeMe"
85       microsoft:
86         enable: true
87         clientId: "MicrosoftClientIdChangeMe"
88         clientSecret: "MicrosoftClientSecretChangeMe"
89       apple:
90         enable: true
91         clientId: "AppleClientIdChangeMe"
92         keyId: "AppleKeyIdChangeMe"
93         teamId: "AppleTeamIdChangeMe"
94       yandex:
95         enable: true
96         clientId: "YandexClientIdChangeMe"
97         clientSecret: "YandexClientSecretChangeMe"
98       vk:
99         enable: true
100        clientId: "VkClientIdChangeMe"
101        clientSecret: "VkClientSecretChangeMe"
102      mailRu:
103        enable: true
104        clientId: "MailRuClientIdChangeMe"
105        clientSecret: "MailRuClientSecretChangeMe"
106
107     skills:
```

```

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

```

```

162         enabled: false
163
164     # Параметры для работы с видеосерверами
165     video:
166         # Параметры Flussonic.
167         flussonic:
168             url: ""
169             apiKey: ""
170             operatorId: ""
171             adminLogin: ""
172             motion:
173                 enabled: false
174         # Параметры видеосервера EVI
175         evi:
176             url: ""
177             apiKey: ""
178             operatorId: ""
179             adminLogin: ""
180
181     acquiring:
182         # Период после завершения действия последней подписки, в течение которого
183         # услуга продолжает (ограниченно) действовать
184         advancePeriod: 3d
185         paykeeper:
186             url: "CHANGE_ME"
187             secret: "PaykeeperSecretChangeMe"
188             user: "PaykeeperUsernameChangeMe"
189             password: "PaykeeperPasswordChangeMe"
190
191     # Настройки ИК-пульта.
192     irc:
193         # Время ожидания ИК команды от пользователя
194         recTimeout: 15s
195         # Таймаут записи команды (отсутствия фронтов)
196         cmdTimeout: 100ms
197
198     # Настройки охраны.
199     guard:
200         # Время, которое дается на включение FLIRS устройств (ждем подтверждение от
201         # контроллера),
202         # при постановке на охрану.
203         deviceRequestDelay: 15s
204         # Время, которое прибавляется ко времени задержки на очистку охранного кэша
205         # при постановке/снятия с охраны,
206         # на случай если охрана не завершила процесс постановки/снятия.
207         # Время задержки формируется как количество охранных устройств умноженное на
208         # deviceRequestDelay.
209         clearContextExtraCacheDelay: 1m
210
211     # Параметры установки сервисов логирования (Elasticsearch + Logstash + Kibana).
212     elk:
213         # Нужно ли добавлять в платформу appender, отправляющий логи в logstash.
214         # В нем нет необходимости, если ELK не развернут или не настроен; это лишь
215         # спровоцирует сообщения об ошибках отправки
216         # в логах платформы.
217         enable: false
218         # Имя (IP-адрес) сервера, на котором будет развернут ELK.
219         # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
220         # платформой (на том же хосте).

```

```
215 # В таком случае хосты в инвентаре в группах [iot] и [monitoring] должны
совпадать.
216   serverName: "{{ iot.serverName }}"
217 # Директория для установки системы логирования.
218   installDir: /storage/elk
219
220 # Параметры установки сервисов мониторинга (Prometheus + Grafana).
221 monitoring:
222   # Имя (IP-адрес) сервера, на котором будет развернуты сервисы мониторинга
(Prometheus + Grafana).
223   # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
платформой (на том же хосте).
224   # В таком случае хосты в инвентаре в группах [iot] и [elk] должны совпадать.
225   serverName: "{{ iot.serverName }}"
226   # Директория для установки системы мониторинга.
227   installDir: /storage/monitoring
228
229 slgate:
230   # Имя сервера, на котором будет развернут SLGATE.
231   # ВАЖНО!!! Нельзя использовать IP-адрес, т.к. такая схема не будет работать!
232   # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
платформой (на том же хосте).
233   # В таком случае хосты в инвентаре в группах [iot] и [slgate] должны совпадать.
234   serverName: "{{ iot.serverName }}"
235   # Директория для установки SLGATE.
236   installDir: /storage/slgate
```

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

iot:	Параметры установки платформы.
serverName: "my.test.server"	<p>Имя (IP-адрес) сервера, на котором будет производиться развертывание платформы IoT. Возможно использование 'localhost', если все манипуляции производятся локально.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p> В 'serverName' нужно прописывать то имя (IP-адрес), по которому будет доступна платформа. Если указать 'localhost', то платформа будет доступна только через 'localhost'.</p> </div>
installDir: /storage/iot	Путь до директории, в которую будет произведена установка.
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.
monitoring:	Параметры установки сервисов мониторинга (Prometheus + Grafana).
serverName: "{{ iot.serverName }}"	Имя (IP-адрес) сервера, на котором будут развернуты сервисы мониторинга (Prometheus + Grafana). 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.
installDir: /storage/monitoring	Директория для установки системы мониторинга.
mongodb:	Параметры MongoDB.
version: 6	Версия MongoDB.
external	
enable: false	<p>Параметр для использования внешней MongoDB. Если выставлен в true, будет использоваться внешняя MongoDB.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p> MongoDB должна быть настроена, а параметры подключения — указаны в 'addr' и 'port'.</p> </div>
addr: "{{ iot.serverName }}"	Адрес внешней MongoDB.
port: 27017	Порт внешней MongoDB.

web:	Параметры WEB.
serverName: "{{ iot.serverName }}"	Имя (IP-адрес) сервера, на котором будет развернут WEB. По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с платформой (на том же хосте).
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.
nginx:	
worker_connections: 1024	Максимальное число соединений, которое одновременно может открыть рабочий процесс.
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 e-mail.
password: "password"	Пароль от учетной записи e-mail.
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.
core:	Параметры платформы ELIS.
privacyPolicyUrl: "my.test.privacy"	Ссылка на политику конфиденциальности.
knowledgeBaseUrl: "IoT"	Ссылка на базу знаний.
logLevel: INFO	Уровень отладки внутри ELIS.
ctlGate:	Порты платформы для подключения Z-Wave-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.
api:	Порты API платформы.
port: 8071	HTTP-порт API-платформы.
sslPort: 8073	HTTPS-порт API-платформы.
captchaLevel: "easy"	Уровень сложности CAPTCHA: easy, medium, hard.
server:	
useHttpsForUi: true	Параметр, позволяющий использовать HTTPS при формировании ссылок к WEB ('true' по умолчанию, при этом будет использован порт, указанный в 'web.httpsPort'). Если поставить в 'false', будет использован HTTP и порт, указанный в 'web.httpPort'.
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.
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.
push:	Включение/выключение push-сообщений.
firebase: enabled: false	Включение/выключение push-сообщений для Android.
apns: enabled: false	Включение/выключение push-сообщений для iOS.
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.
eltex_server:	Параметры видеосервера Eltex.
url: ""	URL сервера Eltex.
apiKey: ""	Ключ API.
operatorId: ""	ID оператора.
adminLogin: ""	Логин администратора.
acquiring:	
advancePeriod: 3	Период после завершения действия последней подписки, в течение которого услуга продолжает (ограниченно) действовать.
paykeeper:	Настройки сервиса оплаты paykeeper.
url: "CHANGE_ME"	URL сервиса paykeeper.
secret: "PaykeeperSecretChangeMe"	Секрет сервиса paykeeper.
user: "PaykeeperUsernameChangeMe"	Пользователь сервиса paykeeper.
password: "PaykeeperPasswordChangeMe"	Пароль сервиса paykeeper.
clientRegistrations:	Параметры клиентских регистраций (через соцсети).
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.
yandexSkill:	Параметры навыка Яндекс для интеграции с Умным домом (Алисой). Отображается 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.
sberSkill:	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.
marusyaSkill:	Параметры проекта умного дома Mail.ru для интеграции с Марусей. Отображаются в карточке проекта/приложения.
clientId: "MarusyaClientIdChangeMe" password: "PasswordChangeMe"	Параметры для Basic Authentication.
appId	App ID, который был назначен приложению VK при создании.
oauthToken: ""	OAuth-токен, который необходимо указывать при отправке уведомлений.
irc:	Настройки ИК-пульта.
recTimeout: 15000	Время ожидания ИК-команды от пользователя в мс.
cmdTimeout: 100	Таймаут записи команды (отсутствия фронтов) в мс.
irdbPath: "CHANGE_ME"	Путь до базы ИК-сигналов IRDB.

- ✓ Для 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 # Версия контейнеров.
3 release: "1.34"
4
5 swarm:
6   enabled: false
7   # Имя stack-а для запуска в docker-swarm.
8   stack: swarm_iot
9
10 # Имя репозитория docker registry, содержащего docker-образы для развертывания.
11 registry: hub.eltex-co.ru
12
13 # Список сервисов для перезапуска (при запуске плейбуков restart_*.yml).
14 # Можно оставить пустым, а при запуске передавать параметром командной строки.
15 services: []
16
17 # Нужно ли выполнять подготовку дистрибутива к установке. Этот шаг полезен при
18 # "чистой" установке
19 # на только что созданный сервер. Если ранее уже была выполнена установка
20 # компонентов IoT через ansible,
21 # то такая подготовка не требуется и этот шаг можно пропустить для экономии времени.
22 withDistroPreparingStep: true
23
24 # Суффикс, добавляемый к имени каждого контейнера (помогает избежать конфликты имен
25 # контейнеров).
26 containerNameSuffix: ""
27
28 # Суффикс, добавляемый к имени создаваемой сети docker (помогает избежать конфликты
29 # имен сетей docker).
30 networkNameSuffix: ""
31
32 # Параметры сервисов IoT (для docker-compose), сгруппированные по именам.
33 # 'enable' - должен ли присутствовать сервис в docker-compose.yml.
34 # 'port.map' - номер порта сервиса в сети хоста.
35 # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
36 # 'db.name' - имя БД, используемой сервисом (связкой сервисов).
37 iot:
38   # Параметры мониторинга использования дискового пространства. Должны
39   # соответствовать требованию:
40   # warnThreshold > criticalThreshold > 0, иначе мониторинг дискового пространства
41   # будет отключен.
42   diskUsage:
43     # Порог дискового пространства (в %), при достижении которого все логи
44     # микросервисов переводятся в режим WARN
45     # (отображаются сообщения с тегами WARN и ERROR). Количество бэкапов баз
46     # уменьшается пропорционально приближению к
47     # порогу criticalThreshold.
48     warnThreshold: 20
49     # Порог дискового пространства (в %), при достижении которого все логи
50     # микросервисов переводятся в режим ERROR
51     # (отображаются только сообщения с тегом ERROR). Бэкапы баз не выполняются.
52     criticalThreshold: 10
53   # Нужно ли установить лимиты на сервисы в docker compose
54   limits:
55     enable: false
56   replication:

```

```

48     core:
49         enabled: false
50         replicaCount: 1
51     mqttBroker:
52         enabled: false
53         replicaCount: 2
54     zwayproxy:
55         enabled: false
56         replicaCount: 1
57     wsproxy:
58         enabled: false
59         replicaCount: 1
60     mongodb:
61         limits:
62             enable: false
63             cpus: 1.0
64             memory: 4G
65         port:
66             map: 27017
67             export: false
68     broker:
69         logLevel: INFO
70         limits:
71             enable: true
72             cpus: 1.0
73             memory: 4G
74         external:
75             # Будет ли развернут брокер на стороннем хосте.
76             enable: false
77             # Нужно ли устанавливать брокер на сторонний хост средствами ansible.
78             install: false
79             # Если установка брокера на сторонний хост будет выполняться средствами
80             ansible, то в какую директорию.
81             installDir: /storage/broker
82             # Внешний адрес стороннего хоста, по которому к брокеру будут подключаться
83             платформа и устройства.
84             host: "external.broker.address"
85             port:
86                 map: 8883
87         internal:
88             port:
89                 map: 8083
90                 export: false
91     db:
92         name: iot-broker
93     jconsole:
94         enabled: false
95         port: 32002
96         # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
97         # При развертывании докера - здесь указывается адреса хоста, где работает
98         докер.
99         host: "10.20.30.40"
100        hivemq:
101            rootFolder: /hivemq/
102    olapService:
103        logLevel: INFO
104        limits:
105            enable: true
106            cpus: 1.0
107            memory: 4G

```

```

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

```

```

161     limits:
162         enable: true
163         cpus: 1.0
164         memory: 4G
165     external:
166         host: "external.authorization.address"
167     port:
168         map: 8091
169         export: false
170     ssl:
171         map: 8093
172         export: false
173     db:
174         name: iot-authorization-server
175     jconsole:
176         enable: false
177         port: 32003
178     # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
179     # При разворачивании докера - здесь указывается адреса хоста, где работает
180 докер.
181     host: "10.20.30.40"
182     swagger:
183         enable: false
184     rabbit:
185         eventQueue:
186             name: platform-rpc-event-queue
187             # Количество консьюмеров на стороне платформы
188             platformConsumers:
189                 # Данный параметр не может быть больше чем maxCount.
190                 count: 16
191                 # Максимальное количество одновременных консьюмеров очереди.
192                 maxCount: 16
193     core:
194         # Разворачивание окружения без платформы, полезно для разработки core.
195         enable: true
196         limits:
197             enable: true
198             cpus: 1.0
199             memory: 4G
200         port:
201             map: 8071
202             export: false
203         ssl:
204             map: 8073
205             export: false
206     db:
207         name: iot-core
208         # Имена устаревших БД в MongoDB, используемые платформой.
209         deprecated:
210             fs:
211                 name: iot-fs
212             licenses:
213                 name: iot-licenses
214             events:
215                 name: iot-events
216     jconsole:
217         enable: false
218         port: 32001
219     # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.

```

```

219     # При разворачивании докера - здесь указывается адреса хоста, где работает
докер.
220     host: "10.20.30.40"
221     swagger:
222     enable: false
223     broker:
224     # Нужно ли платформе пытаться подключиться к MQTT Broker.
225     enable: true
226     threadPools:
227     main:
228     size: 16
229     olapervice:
230     # Нужно ли платформе пытаться подключиться к OlapService.
231     enable: true
232     mjollnir:
233     # URL для Mjollnir.
234     url: "http://lab3-test.eltex.loc:8878/api/v1"
235     slgate:
236     instances:
237     - slgate:4443
238     ignoreServerCertCheck: false
239     video:
240     evi:
241     ignoreServerCertCheck: false
242     checkEventSenderAddress: true
243     # Параметры WebRTC, которые необходимы камере для формирования своих ICE-
кандидатов.
244     webrtc:
245     # Параметры STUN-сервера.
246     stun:
247     # URL STUN-серверов.
248     urls:
249     - "stun.example1.com:3478"
250     - "stun.example2.com:3478"
251     # Параметры TURN-сервера.
252     turn:
253     # Параметры TURN:SFU
254     sfu:
255     # Принудительное включение
256     force: false
257     # Таймаут на ожидание инвента от EVI о начале webrtc-сессии между камерой
и TURN:SFU
258     eventTimeout: 10s
259     # Параметры P2P
260     p2p:
261     # Таймаут на ожидание SDP-answer от камеры
262     sdpAnswerTimeout: 60s
263     # Время ожидания проверки камерой типа своего NAT
264     checkCameraNatTypeTimeout: 10s
265     # Время выполнения ping к host candidate клиента
266     pingHostCandidateTimeout: 10s
267     web:
268     # Разворачивание окружения без WEB, полезно для разработки web.
269     enable: true
270     limits:
271     enable: true
272     cpus: 1.0
273     memory: 4G
274     hazelcast:
275     clusterName: iot-core

```

```

276     instanceName: iot-core-hazelcast-instance
277     limits:
278       enable: true
279       cpus: 1.0
280       memory: 4G
281     external:
282       port:
283         map: 5701
284         export: false
285   rabbitmq:
286     limits:
287       enable: true
288       cpus: 1.0
289       memory: 4G
290     admin:
291       user: guest
292       password: guest
293       port:
294         map: 15671
295         export: true
296       sslPort:
297         map: 15672
298         export: true
299     port:
300       map: 5672
301       export: false
302   zwayproxy:
303     enable: true
304     logLevel: INFO
305     external:
306       enable: false
307     limits:
308       enable: true
309       cpus: 1.0
310       memory: 4G
311     port:
312       map: 8070
313       export: true
314     sslPort:
315       map: 8072
316       export: true
317   rabbit:
318     queue:
319       # Идентификатор очереди для отправки сообщений из прокси в платформу.
320       platform: zway-proxy-platform
321     exchange:
322       # Идентификатор топик-коллектора.
323       proxy: zway-proxy-topic-exchange
324     # Количество консьюмеров на стороне платформы
325     platformConsumers:
326       # Данный параметр не может быть больше чем maxCount.
327       count: 16
328       # Максимальное количество одновременных консьюмеров очереди.
329       maxCount: 16
330     # Количество консьюмеров на стороне zwayProxy service
331     proxyConsumers:
332       # Количество одновременных консьюмеров очереди. Данный параметр не может
333       быть больше чем maxCount.
334       count: 1
335       # Максимальное количество одновременных консьюмеров очереди.

```

```

335         maxCount: 1
336     wsproxy:
337         logLevel: INFO
338         maxConnections: 25000
339         external:
340             enable: false
341             # Адрес хоста куда nginx будет проксировать запросы.
342             host: "1.2.3.4"
343             port: 8075
344         limits:
345             enable: true
346             cpus: 1.0
347             memory: 4G
348         port:
349             map: 8075
350             export: true
351         rabbit:
352             queue:
353                 # Идентификатор очереди для отправки сообщений из прокси в платформу.
354                 platform: ws-proxy-platform
355             exchange:
356                 # Идентификатор топик-коллектора.
357                 proxy: ws-proxy-topic-exchange
358             # Количество консьюмеров на стороне платформы
359             platformConsumers:
360                 # Данный параметр не может быть больше чем maxCount.
361                 count: 16
362                 # Максимальное количество одновременных консьюмеров очереди.
363                 maxCount: 16
364             # Количество консьюмеров на стороне wsProxy service
365             proxyConsumers:
366                 # Количество одновременных консьюмеров очереди. Данный параметр не может
367                 # быть больше чем maxCount.
368                 count: 16
369                 # Максимальное количество одновременных консьюмеров очереди.
370                 maxCount: 16
371     slgate:
372         db:
373             name: "activator"
374             user: "activator"
375             password: "activator"
376             port:
377                 map: 5432
378                 export: false
379         vpn:
380             port:
381                 map: 1194
382             protocol: udp
383             maxClients: 1024
384         easysrsa:
385             req:
386                 country: "RU"
387                 province: "Novosibirsk"
388                 city: "Novosibirsk"
389                 organization: "Eltex"
390                 organizationUnit: "EMS"
391                 email: "eltex@eltex-co.ru"
392     activator:
393         port:

```

```
394     map: 8899
395     export: false
396 proxy:
397     scheme: "https://"
398     useClientCert: false
399     clientCertPassword: "CHANGE_ME"
400     port:
401     map: 4443
402
403 # Параметры сервисов ELK (для docker-compose), сгруппированные по именам.
404 # 'port.map' - номер порта сервиса в сети хоста.
405 # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
406 elk:
407     elasticsearch:
408     rest:
409     port:
410     map: 9200
411     nodes:
412     port:
413     map: 9300
414     logstash:
415     port:
416     map: 5001
417     api:
418     port:
419     map: 9600
420     kibana:
421     port:
422     map: 5601
423
424 # Параметры сервисов мониторинга (для docker-compose), сгруппированные по именам.
425 # 'port.map' - номер порта сервиса в сети хоста.
426 # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
427 monitoring:
428     prometheus:
429     port:
430     map: 9090
431     # Период опроса сервисов со стороны Prometheus в секундах. Небольшие значения
432     # (менее десятков секунд) значительно
433     # увеличат нагрузку на сеть.
434     # При изменении параметра рекомендуется поменять переменную
435     grafana.scrapeInterval на значение не ниже указанного
436     # для Prometheus, чтобы не терять метрики.
437     scrapeInterval: 60
438     grafana:
439     port:
440     map: 3000
441     # Период опроса Prometheus со стороны Grafana в секундах.
442     scrapeInterval: 60
443     nginxExporter:
444     enable: false
445     port:
446     map: 9113
```

Таблица описания значений в файле настроек vars/service_parameters.yml

release: 1.34	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.
limitsEnable: 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	
1	<code>cd /etc/ansible-iot-1.34</code>
2	<code>ansible-playbook install_iot.yml</code>

⚠ 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
<code>docker ps</code>

Вывод команды docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
36c21b863cc9	hub.eltex-co.ru/iot-double-web:1.34	/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:1.34	/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:1.34	/extras/app-starter	2 minutes ago	Up 2 minutes		iot-authorization-server
07d1f93831bd	hub.eltex-co.ru/iot-mqttbroker-mongo:1.34	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:1.34	"java --add-modules ..."	2 minutes ago	Up 2 minutes		iot-ngw-sc
228d41c96cba	hub.eltex-co.ru/iot-olapervice:1.34	java -cp @/app/jib-...	2 minutes ago	Up 2 minutes		iot-olapervice
e8e2899f2c8d	hub.eltex-co.ru/iot-captcha:1.34	java -jar LibreCapt...	2 minutes ago	Up 2 minutes	8888/tcp	iot-captcha
57c02941cc4f	hub.eltex-co.ru/iot-mongo6:1.34	/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:1.34	"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:1.34	"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:1.34	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:1.34	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://\[Адрес вашего сервера ELIS\]](http://[Адрес вашего сервера ELIS])
Адрес сервера был ранее указан в переменной **server_name** файла конфигурации **/etc/ansible-iot-1.34/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](https://www.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-1.34.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-1.34**.

5. Отредактируйте файл **/etc/ansible-iot-1.34/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=password
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-1.34/vars/default.yml** и **/etc/ansible-iot-1.34/vars/service_parameters.yml**.

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

Для параметра **enable** установите значение true, настройте параметры подключения в 'addr' и 'port'.

Содержимое конфигурационного файла vars/default.yml

```

1  mongodb:
2  # Версия MongoDB. На старой аппаратной платформе, не поддерживающей оптимизацию,
  выставить значение `4`
3  version: 6
4  external:
5  # Если выставлен в true, будет использоваться внешняя MongoDB
6  # ВАЖНО!!! MongoDB должна быть настроена, а параметры подключения нужно указать
  в 'addr' и 'port'
7  enable: false
8  # Адрес внешней MongoDB
9  addr: "{{ iot.serverName }}"
10 # Порт внешней MongoDB
11 port: 27017

```

8. После этого можно запустить установку:

```

cd /etc/ansible-iot-1.34
sudo ansible-playbook install_iot.yml

```

- ✓ Платформа будет доступна по адресу: `http://[Адрес вашего сервера ELIS]`
Адрес сервера был ранее указан в переменной **server_name** файла конфигурации `/etc/ansible-iot-1.34/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", "olapervice"]}'
```

В квадратных скобках нужно перечислить сервисы, которые необходимо перезапустить (в примере это "web", "core", "broker", "olapervice"). 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", "olapervice"]}'
```

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-1.34/templates` располагаются конфигурации для ядра: `/etc/ansible-iot-1.34/templates/iot/docker-compose/base/docker-compose.yml.j2` и веб-сервера: `/etc/ansible-iot-1.34/templates/iot/web/base_config`

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-1.34/vars/default.yml`. При запуске платформы параметры, заданные в файле конфигурации `/etc/ansible-iot-1.34/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-ая колонка — очередь принятых сервером пакетов;
- 3-я колонка — очередь пакетов, отосланных на контроллеры;
- 5-я колонка — 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" : "1.34-3477",
3   "api" : "1.0",
4   "currentTime" : "2022-07-25T09:24:12.544842Z[Etc/UTC]"
5 }
```

7 Переход с http на https

1. Открыть файл `base_config.yml.j2`:

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

2. Добавить код в блок WEB:

Содержимое конфигурационного файла `/docker-compose/base_config.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. Внести 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. Добавить свои сертификаты/ключи в /etc/ansible-iot/ssl/

5. Определить 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-1.34/vars/service_params.yml**

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

Настройки уровня сложности представлены тремя возможными значениями и доступны в файле **/etc/ansible-iot-1.34/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>