



Платформа
ELIS

Установка платформы
Версия: 1.33

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 1.33 на операционную систему Ubuntu 20.04. Информацию по установке более ранних версий платформы (1.32 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 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
<pre>tar -C /etc -xvf ansible-iot-1.33.tar.gz</pre>

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-1.33**

5. Отредактируйте файл **/etc/ansible-iot-1.33/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	<code>[iot]</code>
2	<code>localhost ansible_connection=local ansible_sudo_pass=rootpasswd</code>
3	<code>[mqtt_broker]</code>
4	<code>localhost ansible_connection=local ansible_sudo_pass=password</code>
5	<code>[elk]</code>
6	<code>localhost ansible_connection=local ansible_sudo_pass=rootpasswd</code>
7	<code>[monitoring]</code>
8	<code>localhost ansible_connection=local ansible_sudo_pass=rootpasswd</code>

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:
lscri | 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-1.33/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   # Возможно использование 'localhost', если все манипуляции производятся локально.
6   # ВАЖНО!!! В 'serverName' нужно прописывать то имя (IP-адрес), по которому будет
   # доступны платформа.
7   # Если указать 'localhost', то будет доступ только через 'localhost!'
8   serverName: "my.test.server"
9   # Содержит путь до директории, в которую будет произведена установка.
10  installDir: /storage/iot
11
12 # Параметры установки сервисов логирования (Elasticsearch + Logstash + Kibana).
13 elk:
14   # Нужно ли добавлять в платформу appender, отправляющий логи в logstash.
15   # В нем нет необходимости, если ELK не развернут или не настроен; это лишь
   # спровоцирует сообщения об ошибках отправки
16   # в логах платформы.
17   enable: false
18   # Имя (IP-адрес) сервера, на котором будет развернут ELK.
19   # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
   # платформой (на том же хосте).
20   # В таком случае хосты в инвентаре в группах [iot] и [monitoring] должны
   # совпадать.
21   serverName: "{{ iot.serverName }}"
22   # Директория для установки системы логирования.
23   installDir: /storage/elk
24
25 # Параметры установки сервисов мониторинга (Prometheus + Grafana).
26 monitoring:
27   # Имя (IP-адрес) сервера, на котором будут развернуты сервисы мониторинга
   # (Prometheus + Grafana).
28   # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
   # платформой (на том же хосте).
29   # В таком случае хосты в инвентаре в группах [iot] и [elk] должны совпадать.
30   serverName: "{{ iot.serverName }}"
31   # Директория для установки системы мониторинга.
32   installDir: /storage/monitoring
33
34 slgate:
35   # Имя сервера, на котором будет развернут SLGATE.
36   # ВАЖНО!!! Нельзя использовать IP-адрес, т.к. такая схема не будет работать!
37   # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
   # платформой (на том же хосте).
38   # В таком случае хосты в инвентаре в группах [iot] и [slgate] должны совпадать.
39   serverName: "{{ iot.serverName }}"
40   # Директория для установки SLGATE.
41   installDir: /storage/slgate
42
43 # Параметры MongoDB.
44 mongodb:
45   # Версия MongoDB. На старом железе, не поддерживающем оптимизацию, нужно выставить
   # значение `4`.
46   version: 6
47   external:

```

```

48     # Если выставлен в true, будет использоваться внешняя MongoDB.
49     # ВАЖНО!!! MongoDB должна быть настроена, а параметры подключения нужно указать
    в 'addr' и 'port'.
50     enable: false
51     # Адрес внешней MongoDB.
52     addr: "{{ iot.serverName }}"
53     # Порт внешней MongoDB.
54     port: 27017
55
56 # Параметры WEB.
57 web:
58     # Имя (IP-адрес) сервера, на котором будет развернут WEB.
59     # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
    платформой (на том же хосте).
60     serverName: "{{ iot.serverName }}"
61     # Порт HTTP, по которому будет осуществляться доступ в WEB.
62     httpPort: 80
63     # Порт HTTPS, по которому будет осуществляться доступ в WEB.
64     httpsPort: 443
65     # Автоматически перенаправлять запросы по порту HTTP на порт HTTPS
66     redirectHttpToHttps: true
67     nginx:
68         # Максимальное число соединений, которые одновременно может открыть рабочий
    процесс
69         worker_connections: 2048
70         certbot:
71             # Использовать ли certbot для получения сертификатов Let's Encrypt.
72             enable: false
73             # Email владельца домена. Необходим для подтверждения валидности домена при
    получении сертификата Let's Encrypt.
74             email: test@email.com
75             fail2ban:
76                 enable: true
77
78 # Параметры сервера отправки email.
79 mail:
80     smtp:
81         submitter: test@email.com
82         password: "password"
83         senderPrefix: "Сервер Eltex-SC"
84         auth: "true"
85         host: email.com
86         port: 587
87         # Протокол шифрования, используемый при подключении к серверу. Допустимые
    значения: none, starttls, ssl.
88         protection: starttls
89
90 # Параметры платформы IoT core.
91 core:
92     # Ссылка на политику конфиденциальности
93     privacyPolicyUrl: "my.test.privacy"
94     # Ссылка на базу знаний
95     knowledgeBaseUrl: "https://docs.eltex-co.ru/display/EKB/IoT"
96
97     # Уровень отладки внутри IoT Core.
98     logLevel: DEBUG
99
100 # Порты платформы для подключения zway-контроллеров.
101 ctlGate:
102     port: 8070

```



```
103     tcpPort: 8069
104     sslPort: 8072
105
106     # Уровень сложности капчи: easy, medium, hard
107     captchaLevel: "easy"
108
109     server:
110         # Нужно ли использовать HTTPS при формировании ссылок к WEB ('true' по
умолчанию, при этом будет использован порт,
111         # указанный в 'web.httpsPort'). Если поставить в 'false', будет использован HTTP
и порт, указанный в 'web.httpPort'.
112         useHttpsForUi: true
113         # Нужно ли использовать HTTPS при формировании ссылок к ресурсам самой платформы
(например, прошивки).
114         useHttpsForApi: false
115         # Нужно ли использовать HTTPS при формировании ссылок на фото с камер
наблюдения.
116         useHttpsForCameraLinks: true
117         # Нужно ли использовать 'web.serverName' вместо 'iot.serverName' и
'web.httpPort'/'web.httpsPort'
118         # при формировании ссылок к API.
119         useUiProxyForApi: true
120
121     # Параметры для управления доступностью саморегистрации.
122     selfRegistration:
123         allow: true
124         allowDemo: true
125         allowSocialNetworks: false
126
127     push:
128         firebase:
129             enabled: false
130         apns:
131             enabled: false
132
133     # Параметры для работы с видеосерверами
134     video:
135         # Параметры Flussonic.
136         flussonic:
137             url: ""
138             apiKey: ""
139             operatorId: ""
140             adminLogin: ""
141         motion:
142             enabled: false
143         # Параметры видеосервера eltex
144         eltex_server:
145             url: ""
146             apiKey: ""
147             operatorId: ""
148             adminLogin: ""
149
150     acquiring:
151         # Период после завершения действия последней подписки, в течение которого услуга
продолжает (ограниченно) действовать
152         advancePeriod: 3d
153         paykeeper:
154             url: "CHANGE_ME"
155             secret: "PaykeeperSecretChangeMe"
156             user: "PaykeeperUsernameChangeMe"
```

```
157     password: "PaykeeperPasswordChangeMe"
158
159 # Параметры клиентских регистраций (через соцсети).
160 clientRegistrations:
161   google:
162     enable: true
163     clientId: "GoogleClientIdChangeMe"
164     clientSecret: "GoogleClientSecretChangeMe"
165   microsoft:
166     enable: true
167     clientId: "MicrosoftClientIdChangeMe"
168     clientSecret: "MicrosoftClientSecretChangeMe"
169   apple:
170     enable: true
171     clientId: "AppleClientIdChangeMe"
172     keyId: "AppleKeyIdChangeMe"
173     teamId: "AppleTeamIdChangeMe"
174   yandex:
175     enable: true
176     clientId: "YandexClientIdChangeMe"
177     clientSecret: "YandexClientSecretChangeMe"
178   vk:
179     enable: true
180     clientId: "VkClientIdChangeMe"
181     clientSecret: "VkClientSecretChangeMe"
182   mailRu:
183     enable: true
184     clientId: "MailRuClientIdChangeMe"
185     clientSecret: "MailRuClientSecretChangeMe"
186
187 # Параметры навыка Яндекс для интеграции с Умным домом (Алисой). Отображается в
188 # карточке навыка.
189 yandexSkill:
190   enabled: false
191   # Параметры для Basic Authentication.
192   clientId: "YandexClientIdChangeMe"
193   password: "PasswordChangeMe"
194   # Id навыка, который необходимо указывать при отправке уведомлений.
195   skillId: ""
196   # OAuth-токен, который необходимо указывать при отправке уведомлений.
197   oauthToken: ""
198
199 # Параметры проекта умного дома Сбера для интеграции с Салютом. Отображается в
200 # карточке проекта.
201 sberSkill:
202   enabled: false
203   # Параметры для Basic Authentication.
204   clientId: "SberClientIdChangeMe"
205   password: "PasswordChangeMe"
206   # Bearer-токен, который необходимо указывать при отправке уведомлений.
207   bearerToken: ""
208
209 # Параметры проекта умного дома Mail.ru для интеграции с Марусей. Отображается в
210 # карточке проекта/приложения.
211 marusyaSkill:
212   enabled: false
213   # Параметры для Basic Authentication.
214   clientId: "MarusyaClientIdChangeMe"
215   password: "PasswordChangeMe"
216   # App ID, который был назначен приложению VK при создании.
```

```
214     appId: "MarusyaAppIdChangeMe"
215     # OAuth-токен, который необходимо указывать при отправке уведомлений.
216     oauthToken: ""
217
218     # Настройки ИК-пульта.
219     irc:
220         # Время ожидания ИК команды от пользователя
221         recTimeout: 15s
222         # Таймаут записи команды (отсутствия фронтов)
223         cmdTimeout: 100ms
224
225     # Настройки охраны.
226     guard:
227         # Время, которое дается на включение FLIRS устройств (ждем подтверждение от
228         контроллера)
229         # при постановке на охрану.
230         deviceRequestDelay: 15s
231         # Время, которое прибавляется ко времени задержки на очистку охранного кэша при
232         постановке/снятии с охраны,
233         # на случай если охрана не завершила процесс постановки/снятия.
234         # Время задержки формируется как количество охранных устройств умноженное на
235         deviceRequestDelay.
236         clearContextExtraCacheDelay: 1m
```

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

iot:	Параметры установки платформы.
serverName: "my.test.server"	Имя (IP-адрес) сервера, на котором будет производиться развертывание платформы IoT. Возможно использование 'localhost', если все манипуляции производятся локально. <div style="border: 1px solid #ffc107; 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.server Name }}"	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.server Name }}"	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	Параметр для использования внешней MongoDB. Если выставлен в true, будет использоваться внешняя MongoDB. <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;"> <p>⚠ MongoDB должна быть настроена, а параметры подключения – указаны в 'addr' и 'port'.</p> </div>
addr: "{{ iot.server Name }}"	Адрес внешней MongoDB.
port: 27017	Порт внешней MongoDB.
web:	Параметры WEB.
serverName: "{{ iot.server Name }}"	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.
redirectHttp ToHttps: 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.
privacyPolicy Url: "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:	Доступность фиксации движения.
enabled: false	
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"	Параметры для Basic Authentication.
password: "PasswordChangeMe"	
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: "latest"
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: nexus.eltex.loc:9017
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 мониторинга использования дискового пространства. Должны соответствовать
33 # требованию:
34 # warnThreshold > criticalThreshold > 0, иначе мониторинг дискового пространства
35 # будет отключен.
36 diskUsage:
37   # Порог дискового пространства (в %), при достижении которого все логики
38   # микросервисов переводятся в режим WARN
39   # (отображаются сообщения с тегами WARN и ERROR). Количество бэкапов баз
40   # уменьшается пропорционально приближению к
41   # порогу criticalThreshold.
42   warnThreshold: 20
43   # Порог дискового пространства (в %), при достижении которого все логики
44   # микросервисов переводятся в режим ERROR
45   # (отображаются только сообщения с тегом ERROR). Бэкапы баз не выполняются.
46   criticalThreshold: 10
47
48 # Нужно ли установить лимиты на сервисы в docker compose
49 limitsEnable: false
50
51 # Параметры сервисов IoT (для docker-compose), сгруппированные по именам.
52 # 'enable' - должен ли присутствовать сервис в docker-compose.yml.
53 # 'port.map' - номер порта сервиса в сети хоста.
54 # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
55 # 'db.name' - имя БД, используемой сервисом (связкой сервисов).
56 iotServices:

```

```

48 db:
49   limits:
50     enable: false
51     cpus: 1.0
52     memory: 4G
53   port:
54     map: 27017
55     export: false
56 broker:
57   enable: true
58   # будет ли mqtt-broker работать на отдельном от iot сервисов хосте
59   separateHost: false
60   limits:
61     enable: true
62     cpus: 1.0
63     memory: 4G
64   external:
65     host: "{{ iot.serverName }}"
66     port:
67       map: 8883
68   internal:
69     port:
70       map: 8083
71     export: false
72   db:
73     name: iot-broker
74   jconsole:
75     enable: false
76     port: 32002
77     # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
78     # При разворачивании докера - здесь указывается адреса хоста, где работает
79 докер.
80     host: "10.20.30.40"
81   hivemq:
82     rootFolder: /hivemq/
83   olapservice:
84     enable: true
85     limits:
86       enable: true
87       cpus: 1.0
88       memory: 4G
89     port:
90       map: 8023
91       export: false
92     db:
93       enable: true
94       name: iotcore
95       limits:
96         enable: false
97         cpus: 1.0
98         memory: 4G
99     port:
100       map: 8123
101       export: false
102   ngw:
103     enable: false
104     limits:
105       enable: true
106       cpus: 1.0
107       memory: 4G

```

```
107     port:
108       map: 8040
109       export: false
110     db:
111       name: notification-gw
112       user: javauser
113       password: javapassword
114       limits:
115         enable: false
116         cpus: 1.0
117         memory: 4G
118       port:
119         map: 3306
120         export: false
121     captcha:
122       enable: true
123       limits:
124         enable: true
125         cpus: 1.0
126         memory: 4G
127       port:
128         map: 8088
129         export: false
130       caseSensitive: true
131       allowedSizes:
132         - "312x45"
133         - "270x40"
134       instance: "captcha:8088"
135       proportion: 100
136     zscaptcha:
137       enable: false
138       limits:
139         enable: true
140         cpus: 1.0
141         memory: 4G
142       port:
143         map: 8089
144         export: false
145       caseSensitive: true
146       instance: "zs-captcha:8089"
147       proportion: 0
148     core:
149       # Развертывание окружения без платформы, полезно для разработки core.
150       enable: true
151       limits:
152         enable: true
153         cpus: 1.0
154         memory: 4G
155       port:
156         map: 8071
157         export: false
158       ssl:
159         map: 8073
160         export: false
161     jconsole:
162       enable: false
163       port: 32001
164       # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
165       # При развертывании докера - здесь указывается адреса хоста, где работает докер.
```

```
166     host: "10.20.30.40"
167     broker:
168       threadPools:
169         main:
170           size: 16
171     web:
172     # Развертывание окружения без WEB, полезно для разработки web.
173     enable: true
174     limits:
175       enable: true
176       cpus: 1.0
177       memory: 4G
178     replication:
179     core:
180       enabled: false
181       replicaCount: 1
182     mqttBroker:
183       enabled: false
184       replicaCount: 2
185     zwayproxy:
186       enabled: false
187       replicaCount: 1
188     wsproxy:
189       enabled: false
190       replicaCount: 1
191     hazelcast:
192     clusterName: iot-core
193     instanceName: iot-core-hazelcast-instance
194     limits:
195       enable: true
196       cpus: 1.0
197       memory: 4G
198     external:
199     port:
200     map: 5701
201     export: false
202     rabbitmq:
203     limits:
204       enable: true
205       cpus: 1.0
206       memory: 4G
207     port:
208     map: 5672
209     export: false
210     zwayproxy:
211     enable: true
212     external:
213     enable: false
214     limits:
215       enable: true
216       cpus: 1.0
217       memory: 4G
218     port:
219     map: 8070
220     export: true
221     sslPort:
222     map: 8072
223     export: true
224     rabbit:
225     queue:
```

```

226     # Идентификатор очереди для отправки сообщений из прокси в платформу.
227     platform: zway-proxy-platform
228     exchange:
229     # Идентификатор топик-коллектора.
230     proxy: zway-proxy-topic-exchange
231     # Количество консьюмеров на стороне платформы
232     platformConsumers:
233     # Данный параметр не может быть больше чем maxCount.
234     count: 16
235     # Максимальное количество одновременных консьюмеров очереди.
236     maxCount: 16
237     # Количество консьюмеров на стороне zwayProxy service
238     proxyConsumers:
239     # Количество одновременных консьюмеров очереди. Данный параметр не может
    быть больше чем maxCount.
240     count: 1
241     # Максимальное количество одновременных консьюмеров очереди.
242     maxCount: 1
243     wsproxy:
244     enable: true
245     external:
246     enable: false
247     # Адрес хоста куда nginx будет проксировать запросы.
248     host: "1.2.3.4"
249     port: 8075
250     limits:
251     enable: true
252     cpus: 1.0
253     memory: 4G
254     port:
255     map: 8075
256     export: true
257     rabbit:
258     queue:
259     # Идентификатор очереди для отправки сообщений из прокси в платформу.
260     platform: ws-proxy-platform
261     exchange:
262     # Идентификатор топик-коллектора.
263     proxy: ws-proxy-topic-exchange
264     # Количество консьюмеров на стороне платформы
265     platformConsumers:
266     # Данный параметр не может быть больше чем maxCount.
267     count: 16
268     # Максимальное количество одновременных консьюмеров очереди.
269     maxCount: 16
270     # Количество консьюмеров на стороне wsProxy service
271     proxyConsumers:
272     # Количество одновременных консьюмеров очереди. Данный параметр не может
    быть больше чем maxCount.
273     count: 16
274     # Максимальное количество одновременных консьюмеров очереди.
275     maxCount: 16
276
277     slgateServices:
278     db:
279     name: "activator"
280     user: "activator"
281     password: "activator"
282     port:
283     map: 5432

```

```

284     export: false
285   vpn:
286     port:
287       map: 1194
288     protocol: udp
289     maxClients: 1024
290   easysrsa:
291     req:
292       country: "RU"
293       province: "Novosibirsk"
294       city: "Novosibirsk"
295       organization: "Eltex"
296       organizationUnit: "EMS"
297       email: "eltex@eltex-co.ru"
298   activator:
299     port:
300       map: 8899
301     export: false
302   proxy:
303     scheme: "https://"
304     useClientCert: false
305     clientCertPassword: "CHANGE_ME"
306     port:
307       map: 4443
308
309 # Параметры сервисов ELK (для docker-compose), сгруппированные по именам.
310 # 'port.map' - номер порта сервиса в сети хоста.
311 # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
312 elkServices:
313   elasticsearch:
314     rest:
315       port:
316         map: 9200
317     nodes:
318       port:
319         map: 9300
320   logstash:
321     port:
322       map: 5001
323     api:
324       port:
325         map: 9600
326   kibana:
327     port:
328       map: 5601
329
330 # Параметры сервисов мониторинга (для docker-compose), сгруппированные по именам.
331 # 'port.map' - номер порта сервиса в сети хоста.
332 # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
333 monitoringServices:
334   prometheus:
335     port:
336       map: 9090
337     # Период опроса сервисов со стороны Prometheus в секундах. Небольшие значения
338     # (менее десятков секунд) значительно
339     # увеличат нагрузку на сеть.
340     # При изменении параметра рекомендуется поменять переменную
341     grafana.scrapeInterval на значение не ниже указанного
342     # для Prometheus, чтобы не терять метрики.
343     scrapeInterval: 60

```

```

342 grafana:
343   port:
344     map: 3000
345   # Период опроса Prometheus со стороны Grafana в секундах.
346   scrapeInterval: 60
347 nginxExporter:
348   enable: false
349   port:
350     map: 9113
351
352 coreInternal:
353   # Нужно ли включать swagger (описание API платформы).
354   swagger:
355     enable: false
356   # Нужно ли платформе пытаться подключиться к MQTT Broker.
357   mqttbroker:
358     enable: true
359   # Нужно ли платформе пытаться подключиться к OlapService.
360   olapService:
361     enable: true
362   # Имена БД в MongoDB, используемые платформой.
363   core:
364     db:
365       name: iot-core
366   fs:
367     db:
368       name: iot-fs
369   licenses:
370     db:
371       name: iot-licenses
372   events:
373     db:
374       name: iot-events
375   mjollnir:
376     # URL для Mjollnir.
377     url: "http://lab3-test.eltex.loc:8878/api/v1"
378   slgate:
379     instances:
380       - slgate:4443
381     ignoreServerCertCheck: false
382   video:
383     evi:
384       ignoreServerCertCheck: false
385       checkEventSenderAddress: true
386     # Параметры WebRTC, которые необходимы камере для формирования своих ICE-
387     кандидатов.
388     webrtc:
389       # Параметры STUN-сервера.
390       stun:
391         # URL STUN-серверов.
392         urls:
393           - "https://CHANGE_ME_1"
394           - "https://CHANGE_ME_2"
395       # Параметры TURN-сервера.
396       turn:
397         # Параметры TURN:SFU
398         sfu:
399           # Принудительное включение
400           force: false

```



```
400      # Таймаут на ожидание ивента от EVI о начале webrtc-сессии между камерой и
TURN:SFU
401      eventTimeout: 10s
402      # Параметры P2P
403      p2p:
404      # Таймаут на ожидание SDP-answer от камеры
405      sdpAnswerTimeout: 60s
406      # Время ожидания проверки камерой типа своего NAT
407      checkCameraNatTypeTimeout: 10s
408      # Время выполнения ping к host candidate клиента
409      pingHostCandidateTimeout: 10s
```

Таблица описания значений в файле настроек vars/service_parameters.yml

release:1.33	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-1.33</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						
docker ps						
Вывод команды docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
36c21b863cc9	hub.eltex-co.ru/iot-double-web:1.33	/docker-entripoint....	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:1.33	java -Dspring.profi...	2 minutes ago	Up 2 minutes	.0.0.0:8069-8073->8069-8073/tcp, :::8069-8073->8069-8073/tcp	iot-core
07d1f93831bd	hub.eltex-co.ru/iot-mqttbroker-mongo:1.33	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:1.33	"java --add-modules ..."	2 minutes ago	Up 2 minutes		iot-ngw-sc
228d41c96cba	hub.eltex-co.ru/iot-olapervice:1.33	java -cp @/app/jib-...	2 minutes ago	Up 2 minutes		iot-olapervice
e8e2899f2c8d	hub.eltex-co.ru/iot-captcha:1.33	java -jar LibreCapt...	2 minutes ago	Up 2 minutes	8888/tcp	iot-captcha
57c02941cc4f	hub.eltex-co.ru/iot-mongo6:1.33	/entripoint.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:1.33	"docker-entripoint.s..."	2 minutes ago	Up 2 minutes	8123/tcp, 9000/tcp, 9009/tcp	iot-iot-mysql
0327972ab361	hub.eltex.loc:9017/iot-rabbitmq:1.33	"docker-entripoint.s..."	2 minutes ago	Up 2 minutes	4369/tcp, 5671-5672/tcp, 15691-15692/tcp, 25672/tcp	
a5bcb7e27390	hub.eltex.loc:9017/iot-wsproxy:1.33	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:1.33	java -cp @/app/jib-...	2 minutes ago	Up 2 minutes	0.0.0.0:8070->8070/tcp, :::8070->8070/tcp, 0.0.0.0:8072->8072/tcp, :::8072->8072/tcp	iot-zwayproxy

- ✓ Платформа будет доступна по адресу: `http://[Адрес вашего сервера ELIS]`
Адрес сервера был ранее указан в переменной **server_name** файла конфигурации **/etc/ansible-
iot-1.33/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-1.33.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-1.33**.

5. Отредактируйте файл **/etc/ansible-iot-1.33/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-1.33/vars/default.yml** и **/etc/ansible-iot-1.33/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-1.33
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-1.33/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-1.33/templates` располагаются конфигурации для ядра: `/etc/ansible-iot-1.33/templates/iot/docker-compose/base_config.yml.j2` и веб-сервера: `/etc/ansible-iot-1.33/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-1.33/vars/default.yml`. При запуске платформы параметры, заданные в файле конфигурации `/etc/ansible-iot-1.33/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" : "1.33-3477",
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.

Список инстансов CAPTCHA доступен в файле **/etc/ansible-iot-1.33/templates/default-for-docker.yml.j2**

Настройки доступны в файле **/etc/ansible-iot-1.33/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-1.33/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>