

A vertical blue bar with rounded ends, positioned to the left of the text.

Платформа
Eltex SC

Установка платформы
Версия: 1.30

Содержание

1 Рекомендуемые характеристики сервера.....	3
2 Установка платформы Eltex SC.....	4
3 Установка платформы Eltex SC с внешней БД MongoDB.....	27
4 Работа с контейнерами.....	30
5 Расположение конфигурационных файлов.....	31
6 Проверка работоспособности платформы.....	32
7 Переход с http на https.....	34
8 Настройка CAPTCHA.....	37

1 Рекомендуемые характеристики сервера

Система Eltex SC 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;
- 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 Eltex SC

В инструкции описан процесс установки платформы Eltex SC версии 1.30 на операционную систему Ubuntu 20.04. Информацию по установке более ранних версий платформы Eltex SC (1.29 и ниже) можно найти по ссылке: [Архив Eltex SC](#).

⚠ 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 Eltex SC 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 Eltex SC.

Для установки платформы выполните следующие шаги:

1. Установите Ansible на сервер Ubuntu 20.04.

Пример установки через консоль:

Установка Ansible

```

1 apt update
2 apt install --install-recommends linux-generic-hwe-20.04-edge
3 apt install software-properties-common
4 add-apt-repository --yes --update ppa:ansible/ansible
5 apt install ansible

```

✔ Более подробная информация по установке Ansible доступна по ссылке.

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

Проверка версии Ansible

```
ansible --version
```

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

Пример добавления коллекций:

Установка	
1	<code>ansible-galaxy collection install community.general</code>
2	<code>ansible-galaxy collection install community.crypto</code>
3	<code>ansible-galaxy collection install community.docker</code>

4. Подготовьте конфигурацию.

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

Пример распаковки архива:

Распаковка архива с конфигурацией
<code>tar -C /etc -xvf ansible-iot-1.30.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-1.30`

5. Отредактируйте файл `/etc/ansible-iot-1.30/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 <code>inventory</code>	
1	<code>[iot]</code>
2	<code>localhost ansible_connection=local ansible_sudo_pass=rootpasswd</code>
3	<code>[elk]</code>
4	<code>localhost ansible_connection=local ansible_sudo_pass=rootpasswd</code>
5	<code>[monitoring]</code>
6	<code>localhost ansible_connection=local ansible_sudo_pass=rootpasswd</code>

6. Далее необходимо настроить параметры доступа к платформе Eltex SC.

- ❗ 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-1.30/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 # Параметры MongoDB.
35 mongodb:
36   # Версия MongoDB. На старом железе, не поддерживающем оптимизацию, нужно выставить
значение `4`.
37   version: 6
38   external:
39     # Если выставлен в true, будет использоваться внешняя MongoDB.
40     # ВАЖНО!!! MongoDB должна быть настроена, а параметры подключения нужно указать
в 'addr' и 'port'.
41     enable: false
42     # Адрес внешней MongoDB.
43     addr: "{{ iot.serverName }}"
44     # Порт внешней MongoDB.
45     port: 27017
46
47 # Параметры WEB.
```

```

48 web:
49   # Имя (IP-адрес) сервера, на котором будет развернут WEB.
50   # По умолчанию совпадает с 'iot.serverName', что предполагает установку рядом с
   платформой (на том же хосте).
51   serverName: "{{ iot.serverName }}"
52   # Порт HTTP, по которому будет осуществляться доступ в WEB.
53   httpPort: 80
54   # Порт HTTPS, по которому будет осуществляться доступ в WEB.
55   httpsPort: 443
56   # Автоматически перенаправлять запросы по порту HTTP на порт HTTPS
57   redirectHttpToHttps: true
58   nginx:
59     # Максимальное число соединений, которые одновременно может открыть рабочий
   процесс
60     worker_connections: 1024
61     certbot:
62       # Использовать ли certbot для получения сертификатов Let's Encrypt.
63       enable: false
64       # Email владельца домена. Необходим для подтверждения валидности домена при
   получении сертификата Let's Encrypt.
65       email: test@email.com
66
67   # Параметры сервера отправки email.
68   mail:
69     smtp:
70       submitter: test@email.com
71       password: "password"
72       senderPrefix: "Сервер Eltex-SC"
73       auth: "true"
74       host: email.com
75       port: 587
76       # Протокол шифрования, используемый при подключении к серверу. Допустимые
   значения: none, starttls, ssl.
77       protection: starttls
78
79   # Параметры платформы IoT core.
80   core:
81     # Ссылка на политику конфиденциальности
82     privacyPolicyUrl: "my.test.privacy"
83
84     # Уровень отладки внутри IoT Core.
85     logLevel: INFO
86
87     # Порты платформы для подключения zway-контроллеров.
88     ctlGate:
89       port: 8070
90       tcpPort: 8069
91       sslPort: 8072
92
93     # Порты API платформы.
94     api:
95       port: 8071
96       sslPort: 8073
97
98     # Уровень сложности капчи: easy, medium, hard.
99     captchaLevel: "easy"
100
101     server:
102     # Нужно ли использовать HTTPS при формировании ссылок к WEB ('true' по
   умолчанию, при этом будет использован порт,

```

```

103     # указанный в 'web.httpsPort'). Если поставить в 'false', будет использован HTTP
104     и порт, указанный в 'web.httpPort'.
105     useHttpsForUi: true
106     # Нужно ли использовать HTTPS при формировании ссылок к ресурсам самой платформы
107     (например, прошивки).
108     useHttpsForApi: false
109     # Нужно ли использовать HTTPS при формировании ссылок на фото с камер
110     наблюдения.
111     useHttpsForCameraLinks: true
112     # Нужно ли использовать 'web.serverName' вместо 'iot.serverName' и
113     'web.httpPort'/'web.httpsPort'
114     # вместо 'core.api.port'/'core.api.sslPort' при формировании ссылок к API.
115     useUiProxyForApi: false
116
117     # Параметры для управления доступностью саморегистрации.
118     selfRegistration:
119       allow: true
120       allowDemo: true
121       allowSocialNetworks: false
122
123     push:
124       firebase:
125         enabled: false
126       apns:
127         enabled: false
128
129     # Параметры для работы с видеосерверами.
130     video:
131       # Параметры Flussonic.
132       flussonic:
133         url: ""
134         apiKey: ""
135         operatorId: ""
136         adminLogin: ""
137         motion:
138           enabled: false
139       # Параметры видеосервера eltex.
140       eltex_server:
141         url: ""
142         apiKey: ""
143         operatorId: ""
144         adminLogin: ""
145
146     acquiring:
147       # Период после завершения действия последней подписки, в течение которого услуга
148       # продолжает (ограниченно) действовать
149       advancePeriod: 3
150       paykeeper:
151         url: "CHANGE_ME"
152         secret: "PaykeeperSecretChangeMe"
153
154     # Параметры клиентских регистраций (через соцсети).
155     clientRegistrations:
156       google:
157         enable: true
158         clientId: "GoogleClientIdChangeMe"
159         clientSecret: "GoogleClientSecretChangeMe"
160       microsoft:
161         enable: true
162         clientId: "MicrosoftClientIdChangeMe"

```

```

158     clientSecret: "MicrosoftClientSecretChangeMe"
159     apple:
160         enable: true
161         clientId: "AppleClientIdChangeMe"
162         keyId: "AppleKeyIdChangeMe"
163         teamId: "AppleTeamIdChangeMe"
164     yandex:
165         enable: true
166         clientId: "YandexClientIdChangeMe"
167         clientSecret: "YandexClientSecretChangeMe"
168     vk:
169         enable: true
170         clientId: "VkClientIdChangeMe"
171         clientSecret: "VkClientSecretChangeMe"
172     mailRu:
173         enable: true
174         clientId: "MailRuClientIdChangeMe"
175         clientSecret: "MailRuClientSecretChangeMe"
176
177     # Параметры навыка Яндекс для интеграции с Умным домом (Алисой). Отображается в
178     # карточке навыка.
179     yandexSkill:
180         # Параметры для Basic Authentication.
181         clientId: "YandexClientIdChangeMe"
182         password: "PasswordChangeMe"
183         # Id навыка, который необходимо указывать при отправке уведомлений.
184         skillId: ""
185         # OAuth-токен, который необходимо указывать при отправке уведомлений.
186         oauthToken: ""
187
188     # Параметры проекта умного дома Сбера для интеграции с Салютом. Отображается в
189     # карточке проекта.
190     sberSkill:
191         # Параметры для Basic Authentication.
192         clientId: "SberClientIdChangeMe"
193         password: "PasswordChangeMe"
194         # Bearer-токен, который необходимо указывать при отправке уведомлений.
195         bearerToken: ""
196
197     # Параметры проекта умного дома Mail.ru для интеграции с Марусей. Отображается в
198     # карточке проекта/приложения.
199     marusyaSkill:
200         # Параметры для Basic Authentication.
201         clientId: "MarusyaClientIdChangeMe"
202         password: "PasswordChangeMe"
203         # App ID, который был назначен приложению VK при создании.
204         appId: "MarusyaAppIdChangeMe"
205         # OAuth-токен, который необходимо указывать при отправке уведомлений.
206         oauthToken: ""
207
208     # Настройки ИК-пульта.
209     irc:
210         # Время ожидания ИК команды от пользователя (мс).
211         recTimeout: 15000
212         # Таймаут записи команды (отсутствия фронтов) (мс).
213         cmdTimeout: 100
214         # Путь до базы ИК сигналов IRDB.
215         irddbPath: "CHANGE_ME"

```

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 IoT Core.
privacyPolicy Url: "my.test.privacy"	Ссылка на политику конфиденциальности.
knowledgeBaseUrl: "IoT"	Ссылка на базу знаний.
logLevel: INFO	Уровень отладки внутри IoT Core.
ctlGate:	Порты платформы для подключения zway-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: 1.30
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 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
57 #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   limits:
59     enable: true
60     cpus: 1.0
61     memory: 4G
62   external:
63     port:
64       map: 8883
65   internal:
66     port:
67       map: 8083
68       export: false
69 db:
70   name: iot-broker
71 jconsole:
72   enable: false
73   port: 32002
74   # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
75   # При разворачивании докера - здесь указывается адреса хоста, где работает
76 докер.
77   host: "10.20.30.40"
78 olapervice:
79   enable: true
80   limits:
81     enable: true
82     cpus: 1.0
83     memory: 4G
84   port:
85     map: 8023
86     export: false
87 db:
88   name: iotcore
89   limits:
90     enable: false
91     cpus: 1.0
92     memory: 4G
93   port:
94     map: 8123
95     export: false
96 ngw:
97   enable: false
98   limits:
99     enable: true
100    cpus: 1.0
101    memory: 4G
102   port:
103     map: 8040
104     export: false
105 db:
106   name: notification-gw
107   user: javauser
```

```

107     password: javapassword
108     limits:
109         enable: false
110         cpus: 1.0
111         memory: 4G
112     port:
113         map: 3306
114         export: false
115 captcha:
116     enable: true
117     limits:
118         enable: true
119         cpus: 1.0
120         memory: 4G
121     port:
122         map: 8088
123         export: false
124     caseSensitive: true
125     allowedSizes:
126         - "312x45"
127         - "270x40"
128     instance: "captcha:8088"
129     proportion: 100
130 zscaptcha:
131     enable: false
132     limits:
133         enable: true
134         cpus: 1.0
135         memory: 4G
136     port:
137         map: 8089
138         export: false
139     caseSensitive: true
140     instance: "zs-captcha:8089"
141     proportion: 0
142 core:
143     # Развертывание окружения без платформы, полезно для разработки core.
144     enable: true
145     limits:
146         enable: true
147         cpus: 1.0
148         memory: 4G
149     jconsole:
150         enable: false
151         port: 32001
152     # Необходимо указать адрес хоста, куда будет подключаться jconsole-клиент.
153     # При развертывании докера - здесь указывается адреса хоста, где работает
154 докер.
155     host: "10.20.30.40"
156 web:
157     # Развертывание окружения без WEB, полезно для разработки web.
158     enable: true
159     limits:
160         enable: true
161         cpus: 1.0
162         memory: 4G
163 hazelcast:
164     embedded: true
165     clusterName: iot-core
166     instanceName: iot-core-hazelcast-instance

```

```

166     limits:
167         enable: true
168         cpus: 1.0
169         memory: 4G
170     external:
171         port:
172             map: 5701
173             export: false
174 rabbitmq:
175     limits:
176         enable: true
177         cpus: 1.0
178         memory: 4G
179     port:
180         map: 5672
181         export: false
182 zwayproxy:
183     enable: false
184     external:
185         enable: false
186     limits:
187         enable: true
188         cpus: 1.0
189         memory: 4G
190     port:
191         map: 8070
192         export: true
193     sslPort:
194         map: 8072
195         export: true
196     rabbit:
197         queue:
198             # Идентификатор очереди для отправки сообщений из прокси в платформу.
199             platform: zway-proxy-platform
200         exchange:
201             # Идентификатор топик-коллектора.
202             proxy: zway-proxy-topic-exchange
203         # Количество консьюмеров на стороне платформы
204         platformConsumers:
205             # Данный параметр не может быть больше чем maxCount.
206             count: 1
207             # Максимальное количество одновременных консьюмеров очереди.
208             maxCount: 1
209         # Количество консьюмеров на стороне zwayProxy service
210         proxyConsumers:
211             # Количество одновременных консьюмеров очереди. Данный параметр не может
212             # быть больше чем maxCount.
213             count: 1
214             # Максимальное количество одновременных консьюмеров очереди.
215             maxCount: 1
216     wsproxy:
217         enable: false
218         external:
219             enable: false
220             # Адрес хоста куда nginx будет проксировать запросы.
221             host: "1.2.3.4"
222             port: 8075
223         limits:
224             enable: true
225             cpus: 1.0

```

```

225     memory: 4G
226   port:
227     map: 8075
228     export: true
229   rabbit:
230     queue:
231       # Идентификатор очереди для отправки сообщений из прокси в платформу.
232       platform: ws-proxy-platform
233     exchange:
234       # Идентификатор топик-коллектора.
235       proxy: ws-proxy-topic-exchange
236     # Количество консьюмеров на стороне платформы
237     platformConsumers:
238       # Данный параметр не может быть больше чем maxCount.
239       count: 1
240       # Максимальное количество одновременных консьюмеров очереди.
241       maxCount: 1
242     # Количество консьюмеров на стороне wsProxy service
243     proxyConsumers:
244       # Количество одновременных консьюмеров очереди. Данный параметр не может
245       # быть больше чем maxCount.
246       count: 1
247       # Максимальное количество одновременных консьюмеров очереди.
248       maxCount: 1
249   # Параметры сервисов ELK (для docker-compose), сгруппированные по именам.
250   # 'port.map' - номер порта сервиса в сети хоста.
251   # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
252   elkServices:
253     elasticsearch:
254       rest:
255         port:
256           map: 9200
257       nodes:
258         port:
259           map: 9300
260     logstash:
261       port:
262         map: 5001
263     api:
264       port:
265         map: 9600
266     kibana:
267       port:
268         map: 5601
269
270   # Параметры сервисов мониторинга (для docker-compose), сгруппированные по именам.
271   # 'port.map' - номер порта сервиса в сети хоста.
272   # 'port.export' - нужно ли выполнять маппинг порта из контейнера в сеть хоста.
273   monitoringServices:
274     prometheus:
275       port:
276         map: 9090
277     grafana:
278       port:
279         map: 3000
280     nginxExporter:
281       enable: false
282       port:
283         map: 9113

```

```

284
285 coreInternal:
286   # Нужно ли создавать тестовые учетные записи на платформе.
287   testdata:
288     enable: false
289   # Нужно ли включать swagger (описание API платформы).
290   swagger:
291     enable: false
292   # Нужно ли платформе пытаться подключиться к MQTT Broker.
293   mqttbroker:
294     enable: true
295   # Нужно ли платформе пытаться подключиться к OlapService.
296   olapservice:
297     enable: true
298   # Имена БД в MongoDB, используемые платформой.
299   core:
300     db:
301       name: iot-core
302   fs:
303     db:
304       name: iot-fs
305   licenses:
306     db:
307       name: iot-licenses
308   events:
309     db:
310       name: iot-events
311   mjollnir:
312     # URL для Mjollnir.
313     url: "http://smart.eltex-co.ru:8078/api/v1"

```

Таблица описания значений в файле настроек vars/service_parameters.yml

release:1.30	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).
zwayproxу: 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.
olapervice: 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-1.30</code>
2	<code>ansible-playbook install_iot.yml</code>

⚠ Если ранее использовалась версия платформы Eltex SC 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 <code>docker ps</code>						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
36c21b863cc9	hub.eltex-co.ru/iot-double-web:1.30	/docker-entypoint...	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:1.30	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:1.30	java -cp @/app/jib-...	2 minutes ago	Up 2 minutes	0.0.0.0:8883->8883/tcp, :::8883->8883/tcp	iot-mqtt-broker
d1c736dc27d0	hub.eltex-co.ru/eltex-ngw:1.29-602	/usr/sbin/ngw_start...	2 minutes ago	Up 2 minutes		iot-ngw-sc
228d41c96cba	hub.eltex-co.ru/iot-olapservice:1.30	java -cp @/app/jib-...	2 minutes ago	Up 2 minutes		iot-olapservice
e8e2899f2c8d	hub.eltex-co.ru/iot-captcha:1.30	java -jar LibreCapt...	2 minutes ago	Up 2 minutes	8888/tcp	iot-captcha
57c02941cc4f	hub.eltex-co.ru/iot-mongo6:1.30	/entypoint.sh	2 minutes ago	Up 2 minutes	0.0.0.0:27017->27017/tcp, :::27017->27017/tcp	iot-mongo
7c3d8d5c4137	hub.eltex-co.ru/iot-mysql:1.30	docker-entypoint.s...	2 minutes ago	Up 2 minutes	3306/tcp, 33060/tcp	iot-iot-mysql

- ✔ Платформа будет доступна по адресу: `http://[Адрес вашего сервера Eltex SC]`
Адрес сервера был ранее указан в переменной `server_name` файла конфигурации `/etc/ansible-iot-1.30/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 Eltex SC с внешней БД MongoDB

В случае развертывания внешней БД MongoDB, т.е. при установке платформы Eltex SC 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-1.30.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-1.30**.

5. Отредактируйте файл **/etc/ansible-iot-1.30/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	[elk]		
4	localhost	ansible_connection=local	ansible_sudo_pass=rootpasswd
5	[monitoring]		
6	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-1.30/vars/default.yml** и **/etc/ansible-iot-1.30/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-1.30
sudo ansible-playbook install_iot.yml

```

- ✓ Платформа будет доступна по адресу: `http://[Адрес вашего сервера Eltex SC]`
Адрес сервера был ранее указан в переменной **server_name** файла конфигурации `/etc/ansible-iot-1.30/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-1.30/templates` располагаются конфигурации для ядра: `/etc/ansible-iot-1.30/templates/iot/default-for-docker.yml.j2` и веб-сервера: `/etc/ansible-iot-1.30/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-1.30/vars/default.yml`. При запуске платформы параметры, заданные в файле конфигурации `/etc/ansible-iot-1.30/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 Eltex SC 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 Eltex SC>**.

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 контроллеров Z-Wave или датчиков Wi-Fi 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 Eltex SC>:<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" : "1.30-3477",
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-1.30/templates/default-for-docker.yml.j2**

Настройки доступны в файле **/etc/ansible-iot-1.30/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-1.30/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>