

сеCloud. Руководство
администратора
ver.1.2

Для WINDOWS/LINUX

Оглавление

1.	Введение.....	3
1.1.	Назначение документа.....	3
1.2.	Термины.....	3
1.3.	Назначение продукта.....	4
1.4.	Основные функции программы.....	4
1.5.	Технические требования.....	5
2.	Установка.....	6
3.	Конфигурирование.....	12
3.1.	SNMP.....	12
3.2.	Modbus.....	15

1. Введение

1.1. Назначение документа

Этот документ является Руководством администратора seCloud.

Для комфортной работы с seCloud пользователям необходимо знать **ОСНОВЫ работы с электронной почтой и браузером.**

Руководство администратора предназначено для следующих целей:

- Помочь пользователю корректно установить и развернуть продукт.
- Ознакомить пользователя с процессом установки обновлений.
- Объяснить пользователю основные функции работы с SNMP.
- Объяснить пользователю основные функции работы с Modbus.

1.2. Термины

– **cEnergо** - программное обеспечение предназначенное для измерения и многотарифного коммерческого учета электрической энергии и мощности, автоматизированного сбора, хранения, обработки и отображения данных по энергопотреблению.

– **HesDLMS** - протокол, регламентирующий обмен данными между приборами учета и системами сбора данных, в основе которого лежит клиент-серверная архитектура.

– **СПОДЭС** - спецификация протокола обмена данными электронных счетчиков.

– **МЭК-104** - протокол информационного обмена, реализованный в соответствии с ГОСТ Р МЭК 60870-5-104-2004.

– **MKSP-1EE** - модуль управления и мониторинга для электропитающих установок постоянного тока типа.

– **IMEI** - международный идентификатор мобильного оборудования.

– **ModBus** - открытый коммуникационный протокол. Применяется в промышленности для организации связи между электронными устройствами.

– **CEA** - протокол обмена УСПД 164-01М, CE805 и CE805М.

1.3. Назначение продукта

seCloud - это облачный сервис, предназначенный для удаленной работы с приборами учёта. Сервис позволяет отслеживать данные о приборах с помощью графиков, таблиц и журналов событий с помощью стандартного web-браузера.

Клиент может использовать seCloud на своих серверах, либо воспользоваться услугой по размещению сервиса на серверах правообладателя.

Основными областями применения seCloud являются:

- Интеллектуальные системы учета электроэнергии (ИСУЭ).
- Розничный рынок электроэнергии для электросетевых компаний.
- Управляющие компании: СНТ, ДНТ, ТСЖ, УК и другие.

На объектах АСКУЭ «нетребовательных потребителей», с поддержкой приборов учёта по протоколу СПОДЭС.

К дополнительным областям применения относятся:

- Системы мониторинга электрохимзащиты (ЭХЗ) по протоколу MODBUS.
- Системы мониторинга электропитающих устройств (ЭПУ) по протоколу SNMP.

1.4. Основные функции программы

В seCloud существует три типа прав пользователей: абонент, менеджер, администратор. Каждому типу доступны свои программные функции:

– **Абонент.** Имеет доступ к учетной записи, где отображается информация об абоненте, заключенных договорах, и показание счетчиков.

– **Менеджер.** Имеет доступ к списку проектов, к информации о системе, к данным по энергопотреблению, к данным о параметрах сети, к данным о телеметрии, к данным о журналах событий. Также менеджеру доступны следующие функции:

- Управление устройствами: добавлять, редактировать, удалять, заменять, демонтировать, отключать.
- Импортирование данных.
- Управление реле устройств.
- Просмотр и редактирование информации на геокарте.
- Управление расписаниями сервисов.
- Работа с устройствами по протоколу Modbus.

- Работа с устройствами по протоколу SNMP.
- Работа с устройствами по протоколу IEC104.
- **Администратор.** Имеет те же права, что и менеджер, а также доступ к управлению системой: настройка сервера, управление пользователями, настройка импорта из систем сEnergo и HesDLMS, просмотр логов.

1.5. Технические требования

Для корректной работы seCloud компьютер должен соответствовать следующим минимальным требованиям:

- Минимальное разрешение экрана 1280x1024.
- Оперативная память от 4ГБ.
- Подключение к интернету.
- Браузер.

Рекомендованные браузеры:

- Google Chrome v.106.
- Firefox v.109.
- Opera v.92.

2. Установка

Для начала нужно подготовить систему установив в ней Docker и Docker-Compose.

Для работы с приборами учёта (связь с ними, сбор показаний, передача команд управления реле) в текущей версии 1.1 необходима установка сEnergo 4.8 (*только Windows*). Интеграция реализуется через сервис IntegratorCenergo путем взаимодействия с БД сEnergo напрямую (см. руководство пользователя).

Внимание! Для установки необходимо иметь права локального администратора.

2.1 Установка Docker

Актуальные системные требования, описание процесса установки и ссылки на загрузку Docker текущей версии приведены в официальной документации по ссылкам:

Для Windows: <https://docs.docker.com/desktop/windows/install>.

Для Linux: <https://docs.docker.com/desktop/linux/install>.

2.2 Установка Docker Compose

Для Windows он будет установлен в составе Docker Desktop.

Для Linux если установка производилась не пакетов Docker Desktop либо необходима отдельная установка возможна установка вручную.

Ниже приведен пример установки для Ubuntu 22.04.

Начнем с определения последнего выпуска Docker Compose на странице выпусков (<https://github.com/docker/compose/releases>). На момент написания настоящего документа наиболее актуальной стабильной версией является версия 2.14.0.

Запустите следующую команду для загрузки Docker Compose и предоставьте глобальный доступ к этому ПО в своей системе как docker-compose:

```
$ sudo curl -k -L  
"https://github.com/docker/compose/releases/download/v2.14.0/docker-compose-  
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Вариант запуска команды с указанием прокси сервера:

```
$ sudo curl -x 'http://10.5.0.9:3128' -L  
"https://github.com/docker/compose/releases/download/v2.14.0/docker-compose-  
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Вариант запуска команды с указанием прокси сервера и игнорирование SSL сертификата (параметр -k или --insecure):

```
$ sudo curl -k -x 'http://10.5.0.9:3128' -L  
"https://github.com/docker/compose/releases/download/v2.14.0/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Затем необходимо задать правильные разрешения, чтобы сделать команду docker-compose исполняемой:

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

Чтобы проверить успешность установки, запустите следующую команду:

```
$ sudo docker-compose --version
```

Вывод будет выглядеть следующим образом:

```
$ Docker Compose version v2.14.0
```

2.3 Развёртывание и запуск проекта

После того как Docker и Docker-Compose установлены достаточно запустить команду развёртывания проекта из репозитория DockerHub.

Для этого необходимо скопировать файл docker-compose.yml в любую папку и перейдя к ней в командной консоли выполнить команду:

Ubuntu:

```
$ sudo docker-compose -p cloud up -d
```

Windows:

```
docker-compose -p cloud up -d
```

Внимание! Для обеспечения безопасности рекомендуется в docker-compose.yml перед развёртыванием прописать пароль системного администратора СУБД PostgreSQL (заменить root на требуемый в следующих параметрах - POSTGRES_USER=root - POSTGRES_PASSWORD=root).

Внимание! Для Windows консоль управления необходимо открыть от имени администратора.

Внимание! Для успешного выполнения всех действий необходимо наличие интернета. В случае если интернет доступен через прокси-сервер, то необходимо настроить систему и Docker на работу через него (рекомендуется использовать интернет без прокси-сервера).

Внимание! Для исключения бесконтрольного расширения дискового пространства при внутренней процедуре логирования Docker консольного вывода контейнеров необходимо настроить ограничения на файлы логов Docker (официальная документация доступна по [ссылке](#)).

Например, добавив ограничения:

```
"log-opts": {  
  "max-file": "5",  
  "max-size": "10m"  
}
```

Это не более 5 файлов логов архива, с размером не более 10МБ.

Для Docker Desktop под Windows можно выполнить настройки, как на изображении ниже (Рисунок 1).

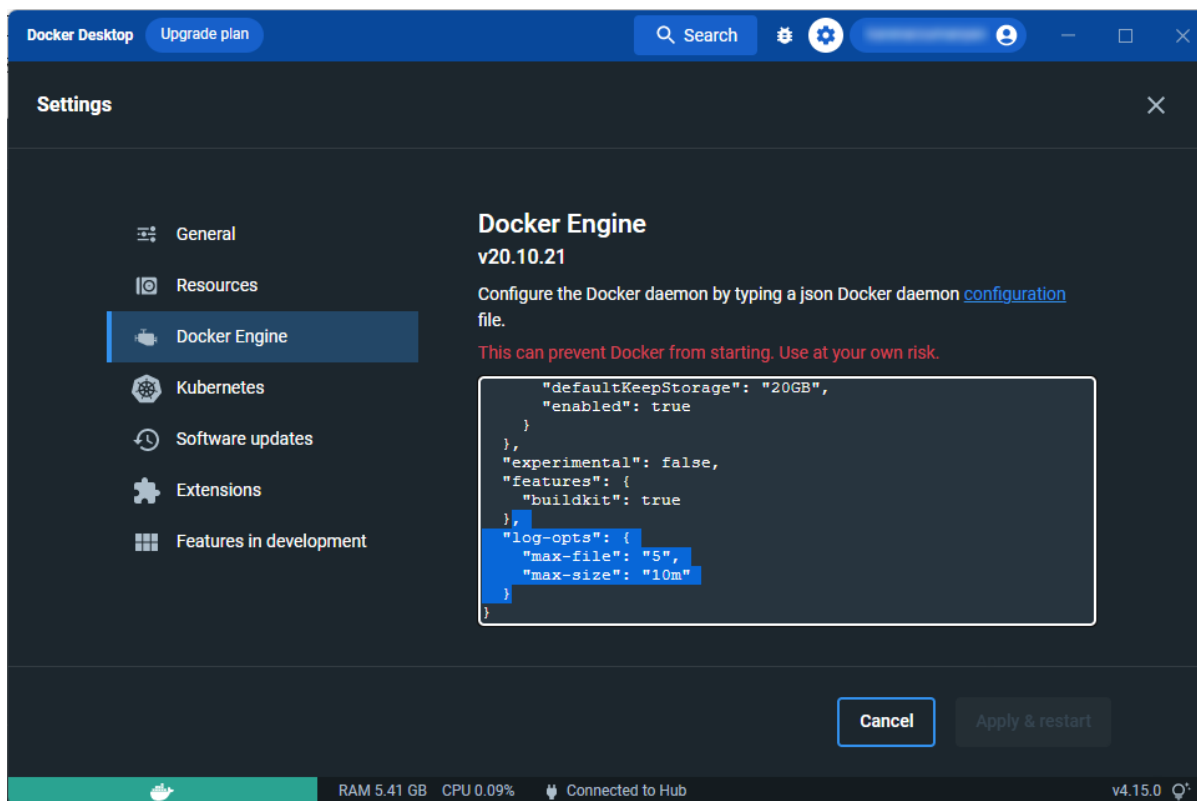


Рисунок 1

После изменения настроек необходимо нажать кнопку «Apply&restart».

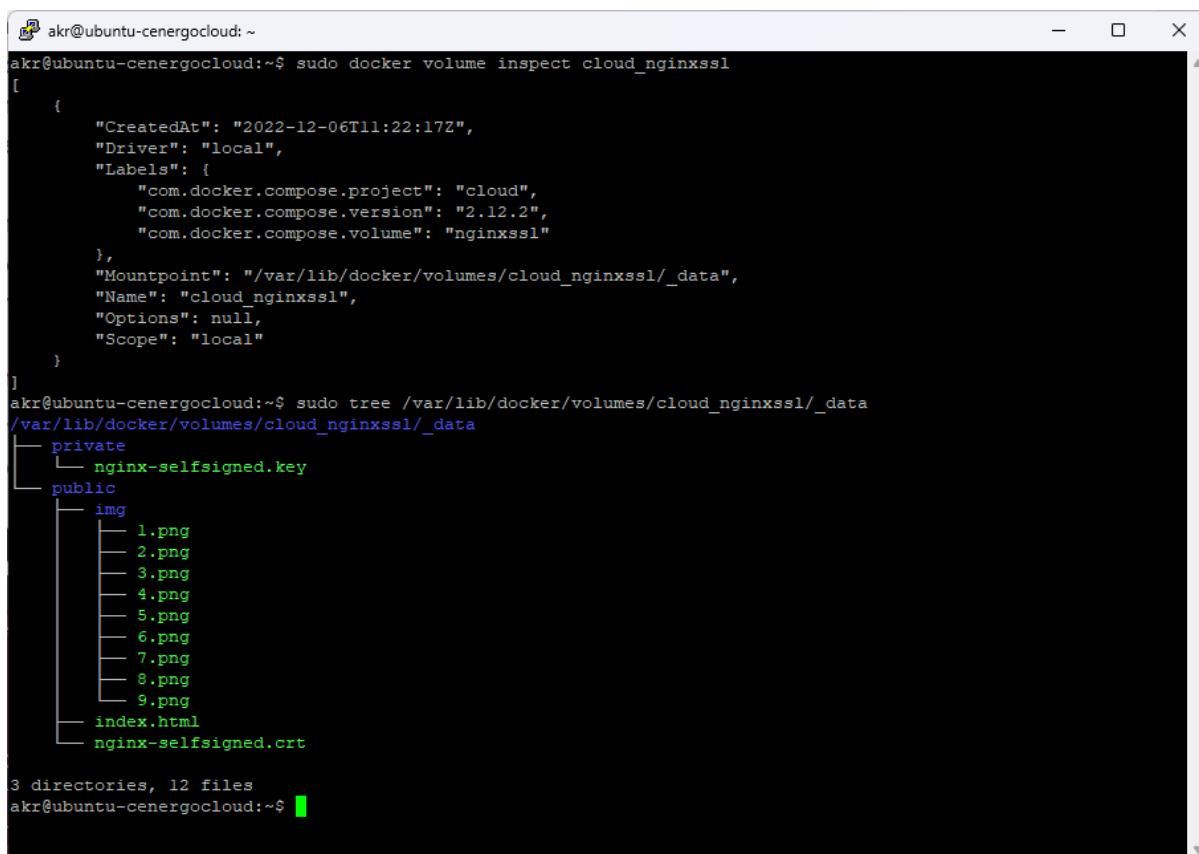
Внимание! Для настройки сертификата безопасности подключения по HTTPS возможна его установка.

При необходимости его можно установить по пути хранения тома cloud_nginxssl. Путь можно узнать в параметре Mountpoint, выполнив команду:

```
docker volume inspect cloud_nginxssl
```

Пример вывода (Рисунок 2):

```
[
  {
    "CreatedAt": "2022-11-29T13:24:59Z",
    "Driver": "local",
    "Labels": {
      "com.docker.compose.project": "cloud",
      "com.docker.compose.version": "2.12.2",
      "com.docker.compose.volume": "nginxssl"
    },
    "Mountpoint": "/var/lib/docker/volumes/cloud_nginxssl/_data",
    "Name": "cloud_nginxssl",
    "Options": null,
    "Scope": "local"
  }
]
```



```
akr@ubuntu-cenergocloud: ~
akr@ubuntu-cenergocloud:~$ sudo docker volume inspect cloud_nginxssl
[
  {
    "CreatedAt": "2022-12-06T11:22:17Z",
    "Driver": "local",
    "Labels": {
      "com.docker.compose.project": "cloud",
      "com.docker.compose.version": "2.12.2",
      "com.docker.compose.volume": "nginxssl"
    },
    "Mountpoint": "/var/lib/docker/volumes/cloud_nginxssl/_data",
    "Name": "cloud_nginxssl",
    "Options": null,
    "Scope": "local"
  }
]
akr@ubuntu-cenergocloud:~$ sudo tree /var/lib/docker/volumes/cloud_nginxssl/_data
/var/lib/docker/volumes/cloud_nginxssl/_data
├── private
│   └── nginx-selfsigned.key
├── public
│   └── img
│       ├── 1.png
│       ├── 2.png
│       ├── 3.png
│       ├── 4.png
│       ├── 5.png
│       ├── 6.png
│       ├── 7.png
│       ├── 8.png
│       └── 9.png
│   ├── index.html
│   └── nginx-selfsigned.crt
3 directories, 12 files
akr@ubuntu-cenergocloud:~$
```

Рисунок 2

По пути /var/lib/docker/volumes/cloud_nginxssl/_data в подпапках private public необходимо расположить файлы ключа (nginx-selfsigned.key) и сертификата (nginx-selfsigned.crt).

Пример подготовки самоподписанного сертификата см. далее 2.1.1 Регистрация в один клик, в разделе SNMP. Нужно обратить внимания на обязательное наличие в сертификате всех вариантов альтернативных имён DNS и IP в [alt_names].

2.4 Обновление

Рекомендуется следующая последовательность шагов:

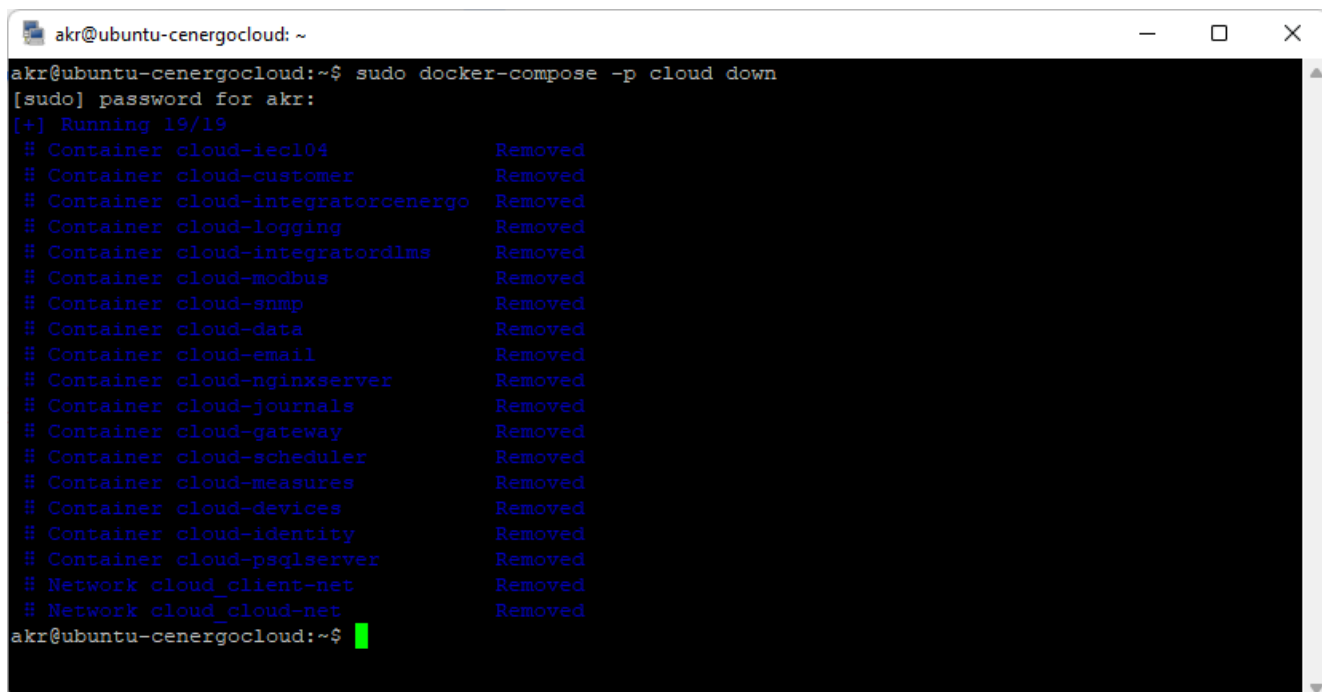
1) Остановить сервисы (Рисунок 3). Этот шаг является необязательным.

Ubuntu:

```
sudo docker-compose -p cloud down
```

Windows:

```
docker-compose -p cloud down
```



```
akr@ubuntu-cenergocloud: ~  
akr@ubuntu-cenergocloud:~$ sudo docker-compose -p cloud down  
[sudo] password for akr:  
[+] Running 19/19  
# Container cloud-iec104 Removed  
# Container cloud-customer Removed  
# Container cloud-integratorcenergo Removed  
# Container cloud-logging Removed  
# Container cloud-integratordlms Removed  
# Container cloud-modbus Removed  
# Container cloud-snmp Removed  
# Container cloud-data Removed  
# Container cloud-email Removed  
# Container cloud-nginxserver Removed  
# Container cloud-journals Removed  
# Container cloud-gateway Removed  
# Container cloud-scheduler Removed  
# Container cloud-measures Removed  
# Container cloud-devices Removed  
# Container cloud-identity Removed  
# Container cloud-psqlserver Removed  
# Network cloud_client-net Removed  
# Network cloud_cloud-net Removed  
akr@ubuntu-cenergocloud:~$
```

Рисунок 3

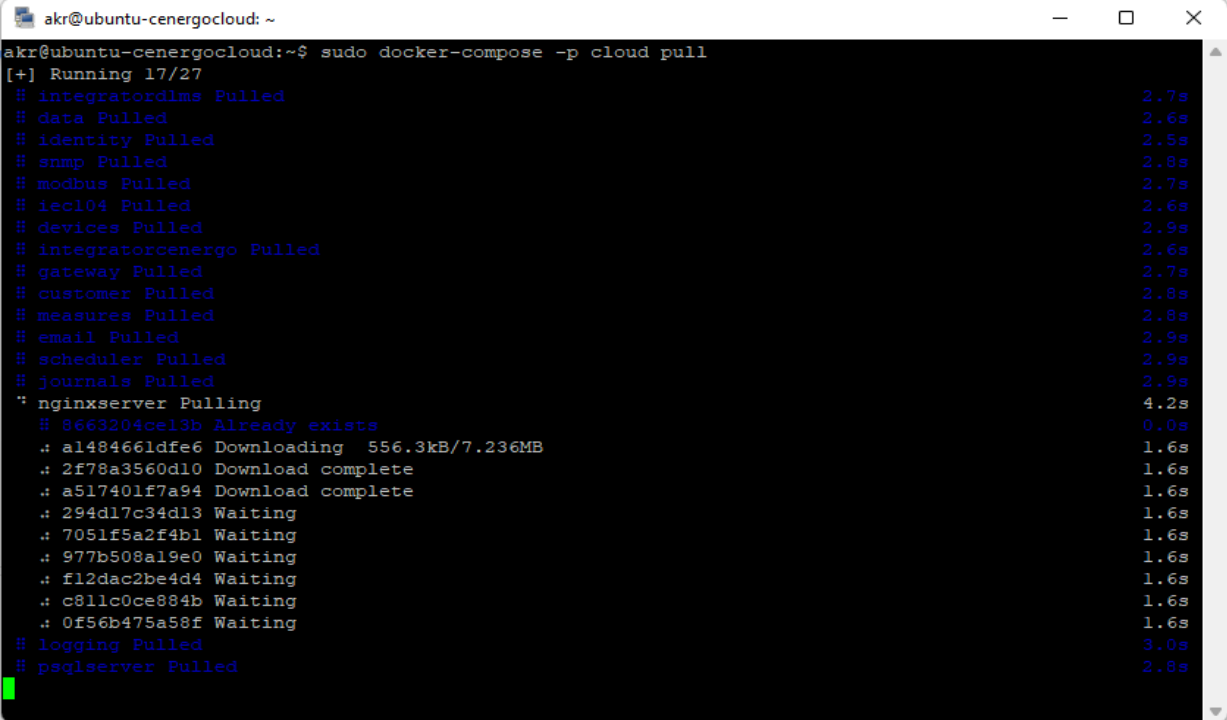
2) Обновить сервисы (Рисунок 4). Этот шаг можно выполнить повторно чтобы убедиться, что все обновления прошли успешно и более системе не обнаруживает новые версии образов.

Ubuntu:

```
sudo docker-compose -p cloud pull
```

Windows:

```
docker-compose -p cloud pull
```



```
akr@ubuntu-cenergocloud: ~  
akr@ubuntu-cenergocloud:~$ sudo docker-compose -p cloud pull  
[+] Running 17/27  
# integratordlms Pulled 2.7s  
# data Pulled 2.6s  
# identity Pulled 2.5s  
# snmp Pulled 2.8s  
# modbus Pulled 2.7s  
# iecl04 Pulled 2.6s  
# devices Pulled 2.9s  
# integratorcenergo Pulled 2.6s  
# gateway Pulled 2.7s  
# customer Pulled 2.8s  
# measures Pulled 2.8s  
# email Pulled 2.9s  
# scheduler Pulled 2.9s  
# journals Pulled 2.9s  
* nginxserver Pulling 4.2s  
# 8663204ce13b Already exists 0.0s  
.: a1484661dfe6 Downloading 556.3kB/7.236MB 1.6s  
.: 2f78a3560d10 Download complete 1.6s  
.: a517401f7a94 Download complete 1.6s  
.: 294d17c34d13 Waiting 1.6s  
.: 7051f5a2f4b1 Waiting 1.6s  
.: 977b508a19e0 Waiting 1.6s  
.: fl2dac2be4d4 Waiting 1.6s  
.: c811c0ce884b Waiting 1.6s  
.: 0f56b475a58f Waiting 1.6s  
# logging Pulled 3.0s  
# postgresqlserver Pulled 2.8s
```

Рисунок 4

2) Запустить сервисы:

Ubuntu:

```
sudo docker-compose -p cloud up -d
```

Windows:

```
docker-compose -p cloud up -d
```

```
akr@ubuntu-cenergocloud: ~
akr@ubuntu-cenergocloud:~$ sudo docker-compose -p cloud up -d
[+] Running 19/19
# Network cloud_cloud-net          Created           0.2s
# Network cloud_client-net        Created           0.2s
# Container cloud-psqlserver       Started           4.5s
# Container cloud-identity         Started           5.1s
# Container cloud-gateway          Started           14.0s
# Container cloud-journals         Started           12.6s
# Container cloud-devices          Started           14.3s
# Container cloud-email            Started           12.2s
# Container cloud-measures         Started           13.9s
# Container cloud-logging          Started           13.5s
# Container cloud-customer         Started           14.0s
# Container cloud-integratoroldms  Started           18.8s
# Container cloud-data             Started           18.4s
# Container cloud-integratorcenergo Started           18.6s
# Container cloud-scheduler        Started           18.9s
# Container cloud-nginxserver      Started           18.5s
# Container cloud-modbus           Started           22.4s
# Container cloud-iec104           Started           22.1s
# Container cloud-snmp             Started           22.1s
akr@ubuntu-cenergocloud:~$
```

Рисунок 5

3. Конфигурирование

3.1. SNMP

Сервис SNMP - использует ряд настроек, которые могут быть изменены в случае необходимости.

Для изменения настроек - необходимо переопределить ключи с помощью переменных среды перед запуском контейнера. Как это может быть сделано описано в документации docker-compose: <https://docs.docker.com/compose/environment-variables/>.

Внимание! Использование «:» для разделения иерархических ключей - не поддерживается рядом операционных систем, потому - лучшим вариантом будет использование «__» ([подробнее](#)).

Для настройки сервиса SNMP используются следующие ключи:

Автоматическая регистрация устройства:

Auth__EntryPoint - веб-сервер через который устройство МКSP-1EE сможет обратиться к ceCloud для проведения процедуры автоматической регистрации в системе.

Пример:

Auth__EntryPoint=<https://www.cecloud.ru>.

Внимание! Значение пути должно включать схему (<https://> или <http://> в зависимости от настроек сервера). По умолчанию используется `https`.

Внимание! В случае использования реверс-прокси, он должен быть настроен на редирект контейнеру `cloud.nginxserver` либо `cloud.gateway`.

3.1.1 Регистрация в один клик

Функция контроллера «Регистрация в один клик» – требует сертификат сервера, содержащего информацию о домене и IP-адресе сервера.

Файл сертификата должен быть расположен **по пути хранения тома `cloud_nginxssl`**. (см. Развёртывание и запуск проекта, в разделе 2. Установка): `/var/lib/docker/volumes/cloud_nginxssl/_data/public`. Имя файла должно быть `nginx-selfsigned.crt`

Для генерации самоподписанного сертификата, можно использовать `openssl`.

Пример создания самоподписанного сертификата для Linux.

1. Создайте новую папку для работы и перейдите в неё:

```
mkdir /new-certs
cd /new-certs
```

2. Сгенерируйте приватный ключ:

```
openssl genrsa -out nginx-selfsigned.key 2048
```

3. Создайте текстовый файл `ssl_conf` следующего содержания:

```
[req]
default_bits = 4096
prompt = no
default_md = sha256
x509_extensions = v3_req
distinguished_name = dn

[dn]
C = RU
ST = {край/область/штат}
L = {город/населенный пункт}
O = {организация}
CN = {доменное имя сервера}
emailAddress = {электронная почта}
```

```
[v3_req]
subjectAltName = @alt_names
```

```
[alt_names]
DNS.1 = {доменное имя сервера}
IP.1 = {ip-адрес сервера}
```

Внимание! Замените значения в фигурных скобках валидными в вашем случае.

4. Сгенерируйте сертификат командой:

```
openssl req -new -x509 -key nginx-selfsigned.key -days 730 -out nginx-selfsigned.crt -
config <(cat ssl_conf)
```

Альтернативный способ генерирования ключа без файла конфигурации:

```
sudo openssl req -x509 -nodes -days 730 -newkey rsa:2048 -out selfsigned.crt -keyout
nginx-selfsigned.key -addext "subjectAltName = DNS.1:{короткое доменное имя сервера},
DNS.2:{доменное имя сервера}, IP.1:{ip-адрес сервера}"
```

В данном случае все параметры описанные в файле конфигурации будут запрашиваться для ввода в процессе генерации ключа.

5. Переместите ключ `nginx-selfsigned.key` в папку `private`, которая была примонтирована к сервисам `nginxserver` и `snmp`.

6. Переместите сертификат `nginx-selfsigned.crt` в папку `public`, которая была примонтирована к сервисам `nginxserver` и `snmp`.

7. Перезапустите сервисы.

Подробности можно изучить в официальной документации OpenSSL

<https://www.openssl.org/docs/manmaster/man1/openssl-req.html>

3.2. Modbus

В сервисе Modbus используются настройки, которые могут быть изменены в случае необходимости.

Для соединения устройств с сервисом Modbus используется внешний порт 21050. В случае если данный порт занят, его можно заменить на необходимый в файле `docker-compose.yml`.

Для этого нужно открыть в любом тестовом редакторе файл `docker-compose.yml`. Затем в файле найти сервис «`modbus`», внутри сервиса найти раздел «`ports`».

Порты имеют следующий формат:

«-внешний_порт_docker:внутренний_порт_контейнера».

Чтобы изменить порта для работы с устройствами нужно найти прописанные порты `-21050:5050` и значение внешнего порта заменить на необходимый для вашей сети.

В случае если `seCloud` запущен чтобы применить изменённые настройки нужно его перезапустить для этого нужно остановить сервисы путем выполнения команды:

Ubuntu:

```
sudo docker-compose -p cloud down
```

Windows:

```
docker-compose -p cloud down
```

И после запустить сервисы:

Ubuntu:

```
sudo docker-compose -p cloud up -d
```

Windows:

```
docker-compose -p cloud up -d
```