Important !!! Hyper-V Configuration. Genereation 1.

Общая информация.

- 1. RedOS Murom 7.3
- 2. Установка с настройками и пакетным содержимым "по умолчанию"
- 3. Установка дополнительных пакетов из стандартных репозиториев
- II. Создаем пользователя для SOICA
- \$ su (необходимо ввести пароль для root. Если root заблокирован, то переходим к шагу (4) текущий пользователь должен быть в sudo

\$ usermod -aG wheel <current_user_name> Выходим из учетной записи root (Ctrl^D)

- \$ sudo adduser soica
- \$ sudo usermod -aG wheel soica

III. Установка docker engine

- \$ sudo yum update (assume y-Yes for all options)
- \$ sudo yum install docker-ce (assume y-Yes for all options)
- $\$ sudo yum install docker-compose (assume y-Yes for all options)
- \$ sudo chkconfig docker on
- \$ sudo systemctl enable docker
- \$ sudo systemctl start docker
- \$ usermod -aG docker soica
- \$ sudo yum install wget

Выходим из терминала и заново заходим под пользователем soica (это необходимо для того, чтобы применились групповые политики)

\$ docker ps `(проверям, что docker daemon запущен)`

IV. Установка дистрибутивов SOICA

1.

- \$ cd ~
- \$ pwd (/home/soica)
- \$ mkdir dist, \$ cd dist (/home/soica/dist)

2. Выкачиваем/копируем с внешнего носителя образы docker контейнеров

- \$ wget http://195.19.176.194:10580/postgres.tar.gz
- \$ wget http://195.19.176.194:10580/soica.web_api.tar.gz
- \$ wget http://195.19.176.194:10580/soica.identity.tar.gz
- \$ wget http://195.19.176.194:10580/soica.control_panel.tar.gz
- \$ wget http://195.19.176.194:10580/soica.validation.tar.gz
- \$ wget http://195.19.176.194:10580/soica.recognition.tar.gz
- \$ wget http://195.19.176.194:10580/soica.license.tar.gz
- \$ wget http://195.19.176.194:10580/soica.ml-locator.tar.gz

3. Скачиваем начальную конфигурацию

\$ wget http://195.19.176.194:10580/soica-initial-configuration.tgz, создаем папку для локальных файлов, хранилище postgres, etc...

- \$ mkdir /home/soica/docker и распаковываем архив.
- \$ tar -xvzf /home/soica/dist/soica-initial-configuration.tgz -C /home/soica/docker

4. Восстанавливаем docker-images

\$	docker	load	<	./postgres.tar.gz
----	--------	------	---	-------------------

- \$ docker load < ./soica.web_api.tar.gz</pre>
- \$ docker load < ./soica.identity.tar.gz</pre>
- \$ docker load < ./soica.control_panel.tar.gz</pre>
- \$ docker load < ./soica.validation.tar.gz</pre>
- \$ docker load < ./soica.recognition.tar.gz</pre>
- \$ docker load < ./soica.license.tar.gz</pre>

\$ docker load < ./soica.ml-locator.tar.gz</pre>

!!! ВАЖНО: Перед следущими шагами проверяем/кастомизируем конфигурацию контейнеров с помощью docker-compose.yml. См. соотв. раздел ниже.

5. Поднимаем psostgres.

\$ docker-compose up -d postgres

!Важно!Следует поднять этот сервис первым.

6. Web.API

\$ docker-compose up -d soica.web-api

!Важно!Следует поднять этот сервис ВТОРЫМ. Он накатит на базу миграции.

7. Проверяем, что оба сервиса работают нормально:

\$ docker ps

Если все нормально - на экран выведется информация о двух запущенных контейнерах.

8. Запускаем остальные сервисы:

\$ docker-compose up -d

Кастомизация конфигурации контейнеров с помощью docker-compose.yml

1. Конфигурация по умолчанию для RedOS Murom 7.3:

\$ cd /home/soica/dist

\$ wget http://195.19.176.194:10580/docker-compose.yml.redos

Или копируем с флешки.

2. Заменяем docker-compose.yml:

\$ cp /home/soica/dist/docker-compose.yml.redos /home/soica/docker/docker-compose.yml

3. Узнаем/запоминаем основной IP для нашего сервера:

\$ ip addr

Команда выдаст все доступные сетевые интерфейсы. Ище смроку типа "inet 192.168.x.x". Обычно с 24-ой маской.

Или спрашиваем у админов компании :)

4. Все сервисы будут доступны по адресам:

http://IP_ADDR:10181 : Identity http://IP_ADDR:10188 : Web-API http://IP_ADDR:10182 : Control Panel http://IP_ADDR:10183 : Validation http://IP_ADDR:10185 : Recognition

service

Например, Control Panel: http://192.168.1.1:10182 (http://192.168.1.1:10182)

5. С помощью любого текстового редактора vim, nano, mcedit открываем на редактирование файл /home/soica/Docker/docker-compose.yml

Нас интересуют несколько параметров для ВСЕХ сервисов:

- a. ASPNETCORE_IdentityService__Uri
- b. ASPNETCORE_AppSettings_validation_url c) ASPNETCORE_AppSettings_service_address

Их значение необходимо изменить с <u>http://195.19.176.194:101xx (http://195.19.176.194:101xx)</u> на

a. http://IP_ADDR:10181/

- b. http://IP_ADDR:10183/ c) http://IP_ADDR:10188/
- 6. Остальные параметры контейнеров работают по конфигурации "из коробки". То есть изменять их не нужно. При условии, что все сервисы хостятся на одном физическом сервере.
- 7. Настройка и подключение удаленного Postgres сервера.
- Установка psql client

\$ sudo yum install -y postgresql15

• Проверка подключения к удаленному серверу/базе данных.

\$ psql -h <host> -p <port> -U <username>

- адрес или ір сервера - порт. Если не указан, то подключение производится по умолчанию на 5432 - имя пользователя

После ввода команды будет необходимо ввести пароль.

- Основные команды в консоли psql
 - \I вывести список доступных баз
 - \c переключиться на необходимую базу
 - \d вывести полный список таблиц/связей и т.д.
- Создание баз soica и soica-license

```
$ psql -h <host> -U <username> -c 'CREATE DATABASE soica';
$ psql -h <host> -U <username> -c 'CREATE DATABASE "soica-license";;
```

• Выполнить скрипт создания initial структуры базы soica

\$ psql -h <host> -U <username> -d soica -f <path_to_init_script>

<path_to_init_script> - путь до init.sql. Обычно /home/soica/docker/postgres/initdb/init.sql

• Конфигурация docker-compose.yml

Во всех сервисах(за исключением soica-license) изменить параметр ASPNETCORE_ConnectionStrings_SoicaDB:

 $\label{eq:asymptotic} ASPNETCORE_ConnectionStrings_SoicaDB=Host=; Parte; Database=soica; Username=; Password=; Include Error Detail=True; Password=; Pas$

- адрес или ір сервера - порт. Если не указан, то подключение производится по умолчанию на 5432 - имя пользователя - пароль

Для soica-license:

ASPNETCORE_ConnectionStrings__AppDatabase=Host=<host>;Port=<port>;Database=soica-license;Username=<username>;Password=<password>;Inc