Инструкция по развертыванию X-Control

2021

1 Подготовка

Для развертывания системы потребуется

1. Сервера с установленной CentOs 7 (https://www.centos.org/download/)
2. Дистрибутив системы, состоящий из следующих пакетов:
    a. AOKZ.WebApi.x.y.z.rhel.7-x64.rpm
    b. clisa.x.y.z.rpm
    c. (опционально) AOKZ.IntegrationService.WebApi.x.y.z.rhel.7-x64.rpm
    d. (опционально) AOKZ.PoibSobi.Connector.x.y.z.rhel.7-x64.rpm
    e. Архив с Web компонентом (aokz.site.tar.gz)
3. Доступ к сети интернет для загрузки и установки дополнительного ПО на сервера.

В случае отсутствия доступа к сети Интернет в инфраструктуре, в которой производится развертывание системы X-Control, необходимо заранее подготовить (загрузить) пакеты, которые потребуются для развертывания (pgsql, nginx и т.д.)

2 Развертывание по схеме с 3 серверами - DB, App, Web

2.1 Развертывание сервера СУБД (DB)

На сервере БД (AOKZ-DB) выполните следующие действия:

1. Разверните сервер PostgreSQL 11 в соответствии с официальной документацией https://www.postgresql.org/download/linux/redhat/

```
yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm

yum install -y postgresql11-server
```

2. Проинициализируйте бд

```
/usr/pgsql-11/bin/postgresql-11-setup initdb
```

3. Настройте автоматический запуск pg.

```
systemctl enable postgresql-11
systemctl start postgresql-11
```

4. Настройка пользователя и проверка подключения

   Задайте пароль для пользователя postgres:

```
passwd postgres
```

   Зайдите в систему под данной учетной записью:

```
su – postgres
```

   Подключитесь к командой оболочке psql:

```
psql
```

   Создайте пользователя БД (предварительно сгенерируйте и укажите пароль):

```
=#CREATE USER aokz WITH PASSWORD 'insert_user_password_here';
```

   Выдайте пользователю права на создание новый БД:

```
=# ALTER USER aokz CREATEDB;
```

   Отключитесь от psql :

```
 =# \q
```

Отключитесь от системы пользователем postgres:

```
exit
```

5. Откройте сетевой порт для DB

```
firewall-cmd --add-port=5432/tcp --permanent
firewall-cmd --reload
```

6. Настройте необходимые методу аутентификации и параметры удаленного подключения к PostgreSQLОпределите путь расположения БД

```
ps aux | grep postgres | grep -- -D
```

```
postgres  1951  0.0  0.2 396836 17000 ?        Ss   11:08   0:00
/usr/pgsql-11/bin/postmaster -D /var/lib/pgsql/11/data/
```

Code Block 1 Пример вывода

`/var/lib/pgsql/11/data/` – это путь расположения баз и конфигурационных файлов сервера БД

7. Измените адрес, на котором PostgreSQL принимает запросы:

   Откройте файл /db/pgsql/postgresql.conf

   Найдите и отредактируйте строку:

```
listen_addresses = '*'
```

> По умолчанию параметр закомментирован и настроен на прослушивание запросов только с локального сетевого интерфейса. В данном примере мы разрешили прослушивание запросов на всех IP-адресах (*), но, если требуется более безопасная настройка, можно просто перечислить последние через пробел.

8. Настройте аутентификации клиентов PostgreSQL:

   Откройте на редактирование конфигурационный файл /db/pgsql/pg_hba.conf и внизу добавьте следующую строку:

```
host     all     all     192.168.0.10/32     md5
```

В данном примере мы разрешаем удаленные подключения к серверу с компьютера 192.168.0.10. Доступ предоставляется всем учетным записям и всем базам (значение all). При необходимости, вместо all можно указать конкретные данные для повышения безопасности.

Как минимум, доступ должен быть разрешен с сервера приложений. Подробнее про структуру файла pg_hba.conf можно прочитать здесь: https://postgrespro.ru/docs/postgrespro/11/auth-pg-hba-conf

9. Перезапустите службу PostgreSQL

```
systemctl restart postgresql-11
```

Проверить подключение можно с удаленного компьютера следующей командой:

psql -h 192.168.0.2 -U usersql

* где 192.168.0.2 — IP-адрес сервера баз данных; usersql — имя учетной записи, от которой идет подключение.

Либо с помощью любого графического клиента PostgreSQL (например, PgAdmin)

2.2 Развертывание сервера приложений (App)

1. Откройте порты на доступ снаружи:

```
firewall-cmd --add-port=80/tcp --permanent
firewall-cmd --reload
```

2. Разверните Redis

Официальный сайт: https://redis.io/download
Инструкция по развертыванию: https://computingforgeeks.com/how-to-install-latest-redis-on-centos-7/

Добавьте репозиторий Remi:

```
yum -y install http://rpms.remirepo.net/enterprise/remi-release-7.rpm
```

Установите последний Redis:

```
yum --enablerepo=remi install redis
```

Добавьте сервис в автозагрузку:

```
systemctl enable --now redis
```

Проверьте статус службы Redis:

```
systemctl status  redis
```

3. Разверните SFTP сервер (используется для хранения загружаемых пользователями файлов)

> SFTP сервер может быть развернут как на APP сервере (в случае малого предполагаемого объема хранимых файлов), так и на отдельном сервере (предпочтительно в случае больших объемов).

Создайте нового пользователя и задайте ему пароль

```
adduser aokz
passwd aokz
```

Создайте директорию, в которой будут храниться файлы SFTP сервера и дайте на нее права созданному пользователю

```
mkdir -p /var/sftp/uploads
chown root:root /var/sftp
chmod 755 /var/sftp
chown aokz:aokz /var/sftp/uploads
```

Отредактируйте файл **/etc/ssh/sshd_config** , добавьте в него следующий блок

```
Match User aokz
ForceCommand internal-sftp
PasswordAuthentication yes
ChrootDirectory /var/sftp
PermitTunnel no
AllowAgentForwarding no
AllowTcpForwarding no
X11Forwarding no
```

Перезапустите ssh сервер

```
systemctl restart sshd
```

4. Разверните AOKZ.WebAPI из пакета AOKZ.WebApi.1.0.0.rhel.7-x64.rpm
   a. Разверните сервис из rpm

```
rpm -i ./AOKZ.WebApi.1.0.0.rhel.7-x64.rpm
```

   b. Сервис будет установлен в папку /usr/share/AOKZ.WebApi/

При необходимости, укажите порт, который будет использоваться сервисом. (По-умолчанию используется 80-й порт.) Для этого в конфигурационном файле /usr/share/AOKZ.WebApi/appsettings.json в разделе "CommonOptions" укажите параметр “Port”:

```
"CommonOptions": {
   "Port":  80
 }, ...
```

   c. Проверьте наличие в папке /usr/share/AOKZ.WebApi/ конфигов, кроме appsettings.json по маске appsettings.*.json и удалите их:

```
# проверка
find /usr/share/AOKZ.WebApi/ -name "appsettings.*.json" -type f
# удаление
find /usr/share/AOKZ.WebApi/ -name "appsettings.*.json" -type f -delete
```

   d. Настройте подключение к БД.
   Для этого в файле /usr/share/AOKZ.WebApi/appsettings.json найдите раздел «ConnectionStrings», в нем строку подключения “AOKZ” и данной строке подключения укажите данные для подключения к развернутому ранее серверу БД:

```
"ConnectionStrings": {
    "AOKZ": "Host=aokz-
db;Port=5432;Database=aokz;Username=aokz;Password=insert_user_password_here"
},
...
```

где

Host=aokz-db – имя или ip-адрес нашего сервера баз данных

Port=5432 – порт сервера БД. По-умолчанию используется порт 5432. Если Вы при настройке сервера БД изменили номер порта – впишите сюда соответствующее значение.

Database=aokz – имя базы данных, которую будет использовать сервис (будет создана автоматически).

Username=aokz – имя пользователя, которого вы создали при развертывании сервера БД

Password=insert_password_here – пароль пользователя, который вы создали при развертывании сервера БД.

e. Настройте подключение к Redis. Для этого в файле /usr/share/AOKZ.WebApi/appsettings.json найдите или создайте раздел "RedisCache", в котором укажите настройки для подключения к установленному сервису Redis:

```
"RedisCache": {
    "ConnectionString": "127.0.0.1",
    "DatabaseNumber": 0,
    "DefaultKeyLifeTime": "0.00:05:00"
  },
...
```

f. Настройте подключение к почтовому серверу для отправки уведомлений. Для этого в файле /usr/share/AOKZ.WebApi/appsettings.json найдите или создайте раздел "EmailOptions", в котором укажите настройки для подключения к почтовому серверу (по SMTP):

```
"EmailOptions": {
    "Host": "mail.mycompany.ru",
    "Port": 25,
    "FormTitle": "Система АОКЗ",
    "FromEmail": aokz@mycompany.ru
  },
...
```

g. Включите MOCK-режим аутентификации, для этого в конфигурационном файле /usr/share/AOKZ.WebApi/appsettings.json укажите режим аутентификации «MOCK»:

```
"Auth": {
    "Mode": [ "MOCK" ],
...
}
```

h. Настройте подключение к SFTP серверу. Для этого в конфигурационном файле /usr/share/AOKZ.WebApi/appsettings.json, в секции "FilesSaving" укажите параметры подключения к SFTP серверу:

```
"FilesSaving": {
    "StorageType": "SFTP",
    "SFtpSettings": {
      "Host": "sftp_server_ip",
      "Username": "aokz",
      "Password": "user_password_here",
      "BaseFolder": "path_on_sftp_to_upload"
    }
  }
```

i. (опционально) Настройте подключения к Jinn. Для этого в конфигурационном файле /usr/share/AOKZ.WebApi/appsettings.json, в секции "JinnClientSettings" укажите параметры подключения к Jinn server:

```
"JinnClientSettings": {
    "BaseAddress": "http://jinn.server.address:8080",
    "ValidationServiceUrl": "/tccs/SignatureValidationService"
  },
```

j. Сконфигурируйте приложение /usr/share/AOKZ.WebApi/AOKZ.WebApi для работы в режиме фонового сервиса (systemd). Для этого создайте файл /etc/systemd/system/AOKZ_WebApi.service со следующим содержимым:

```
[Unit]
Description=AOKZ_WebAPI
Requires=redis.service

[Service]
Type=simple
PIDFile=/usr/share/AOKZ.WebApi/service.pid
WorkingDirectory=/usr/share/AOKZ.WebApi
User=root
Group=nobody
Environment=ASPNETCORE_ENVIRONMENT=Development
OOMScoreAdjust=-1000
ExecStart=/usr/share/AOKZ.WebApi/AOKZ.WebApi
TimeoutSec=300

[Install]
WantedBy=multi-user.target
```

k. Добавьте а автозагрузку и запустите сервис:

```
systemctl enable AOKZ_WebApi
systemctl start AOKZ_WebApi
```

l. Проверьте состояние сервиса и последние служебные сообщения:

```
systemctl -l status AOKZ_WebApi
```

В дальнейшем для управления службой можно использовать стандартные команды
systemctl start AOKZ_WebApi
systemctl stop AOKZ_WebApi systemctl restart AOKZ_WebApi

5. Разверните консоль супер-администратора (CliSa) из clisa.x.y.z.rpm

```
rpm -i ./clisa.x.y.z.rpm
```

Консоль супер-администратора будет установлена в папку /usr/share/clisa/
Запуск:

```
dotnet /usr/share/clisa/clisa.dll
```

2.3 Развертывание Web-сервера

1. Установите nginx (https://nginx.org/ru/linux_packages.html#RHEL-CentOS)
2. Распакуйте содержимое архива aokz.site.tar.gz в папку /usr/share/nginx/html/aokz
3. Удалите файл /etc/nginx/conf.d/default.conf, и вместо него создайте файл /etc/nginx/conf.d/aokz.conf со следующим содержимым, прописав в параметрах «proxy_pass» адрес развернутого ранее сервера приложений:

```
server {
    underscores_in_headers on;

    listen       80;
    server_name  localhost;

    client_max_body_size 500M;

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    underscores_in_headers on;
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name localhost;

    ssl_protocols TLSv1.2;
    ssl_certificate /etc/nginx/ssl/aokz-test.crt;
    ssl_certificate_key /etc/nginx/ssl/aokz-test.key;

    location / {
        root   /usr/share/nginx/html/aokz;
        index  index.html index.htm;
    }

    location /swagger/ {
        proxy_pass   http://aokz-back$request_uri;
    }

    location /api/ {
        proxy_pass   http://aokz-back$request_uri;
    }
}
```

Code Block 2 /etc/nginx/conf.d/aokz.conf

4. Сгенерируйте самоподписанный SSL сертификат (путь хранения сертификата и ключа должен совпадать с указанным в файле aokz.conf на предыдущем шаге)

```
mkdir -p /etc/nginx/ssl
openssl req -new -x509 -sha256 -newkey rsa:2048 -days 3650 -nodes -out
/etc/nginx/ssl/aokz.crt -keyout /etc/nginx/ssl/aokz.key -subj
"/C=RU/L=Moscow/O=Companyname/CN=localhost"
```

В данном случае генерируется самоподписанный сертификат сроком действия 10 лет. Параметры субъекта сертификата задаются в -subj "/C=RU/L=Moscow/O=Companyname/CN=localhost".

Для прохождения проверки валидности сертификата рекомендуется использовать сертификат, выданный коммерческим CA, либо Let's Encrypt (https://letsencrypt.org/ru/).

5. Откройте порты для доступа извне:

```
firewall-cmd --add-port=80/tcp --permanent
firewall-cmd --add-port=443/tcp --permanent
firewall-cmd --reload
```

6. Проверьте конфигурацию nginx:

```
nginx -t
```

7. Запустите nginx:

```
systemctl start nginx
```

Настройка завершена. Для проверки перейдите по адресу https://адрес_сервера_web/