РУКОВОДСТВО АДМИНИСТРАТОРА BP-MODULE

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

1.1. Область применения

Данный документ предназначен для сотрудника эксплуатирующей организации, обладающего функциональной ролью «Администратор системы» (на уровне ОС), который выполняет функции по установке и настройке программного обеспечения Системы (**далее – администратор системы**).

1.2. Перечень инсталляционных файлов Системы

Пакет инсталляционных файлов Системы содержит (Таблица 1):

- подготовленные docker образы модулей Системы;
- скрипты развертывания базы данных;
- конфигурационные файлы для запуска docker образов.

Таблица 1

№	Имя файла	Назначение
1.	images.zip/client.tar	docker образ модуля клиентской части приложения
2.	images.zip/gis-client.tar	docker образ модуля клиентской части приложения для отображения картографической информации
3.	images.zip/api.tar	docker образ модуля серверной части приложения
4.	images.zip/loader.tar	docker образ модуля экспорта и импорта пространственных данных
5.	images.zip/gisapi.tar	docker образ модуля серверной части приложения
6.	images.zip/redis.tar	docker образ сервиса redis
7.	images.zip/openresty.tar	docker образ веб сервера nginx с модулем Lua
8.	images.zip/wso2is.tar	docker образ сервера аутентификации
9.	images.zip/geoserver.tar	docker образ сервера пространственных данных
10.	init-schema.zip	SQL скрипты создания модели в БД приложений.
11.	init-schema-wso2is.zip	SQL скрипты создания модели в БД сервера идентификации
12.	geoserver_data.zip	Данные и конфигурационные файлы геосервера
13.	configs.zip	Действующие конфигурационные файлы

1.3. Требования к программно-аппаратному комплексу

Для инсталляции программного обеспечения Системы необходимо, чтобы на аппаратном комплексе было установлено следующее базовое программное обеспечение третьих сторон:

- РЕД ОС Ver. 7.3 – операционная система серверов приложений и баз данных.
- PostgreSQL 12.7 (PostgreSQL 11 для WSO2 IS) – программное обеспечение с открытым исходным кодом, система управления

реляционными базами данных, включая автономные службы. Является частью прикладного ПО.

- Nginx 1.18 (официально поддерживаемая РЕД ОС) – программное обеспечение с открытым исходным кодом, используемое для создания и управления web сервером, обрабатывающим запросы пользователей. Является частью прикладного ПО.
- Tomcat 9.0.53 – программное обеспечение с открытым исходным кодом для запуска web приложений, реализует спецификацию java сервлетов. Устанавливается в docker контейнерах. Является частью прикладного ПО.
- Docker CE 20.10 – программное обеспечение с открытым исходным кодом для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации, платформа контейнеризации приложений, предоставляющая общеязыковую среду исполнения кода, разработанных на различных языках программирования. Для определения настроек и запуска приложений в контейнерах используется утилита Docker Compose, являющаяся инструментом из состава Docker для работы с конфигурационными файлами формата YAML. Является частью прикладного ПО.
- OpenJDK 17 (OpenJDK 11 для WSO2 IS и для GeoServer) – программное обеспечение с открытым исходным кодом для запуска кода на языке Java. Является частью прикладного ПО. Устанавливается в docker контейнерах.
- WSO2 IS 5.11 – программное обеспечение с открытым исходным кодом для централизованного управления доступом к различным веб-приложениям, сервисам и API. Используется для интеграции с MS Active Directory. Устанавливается в docker контейнере, на сервере аутентификации. Является частью прикладного ПО.
- Redis 6.0 – программное обеспечение с открытым исходным кодом, резидентная система управления базами данных класса NoSQL с открытым исходным кодом, работающая со структурами данных типа «ключ – значение». Используется для хранения сессионных данных пользователей. Устанавливается в docker контейнере, на сервере контейнеризации. Является частью прикладного ПО.
- GDAL 3.4 – программное обеспечение с открытым исходным кодом, библиотека для чтения и записи растровых и векторных геопространственных форматов данных, выпускаемая под Open Source лицензией X/MIT организацией Open-Source Geospatial Foundation (OSGeo). Библиотека предоставляет вызывающим приложениям единую абстрактную модель данных для всех поддерживаемых форматов. При сборке можно также включить дополнительные утилиты. С помощью этих утилит можно выполнять конвертацию и обработку данных, используя интерфейс командной строки. Устанавливается в docker контейнере, на сервере контейнеризации. Является частью прикладного ПО.
- Geoserver 2.20 – программное обеспечение с открытым исходным кодом, написанное на Java, предоставляющее возможность администрирования и публикации геоданных на сервере. Соответствует стандартам Open Geospatial Consortium: WMS, WFS, WMTS, WPS. Устанавливается в docker контейнере, на сервере ГИС. Является частью прикладного ПО.

2. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Установка программного обеспечения выполняется в следующей последовательности:

- устанавливается базовое ПО согласно официальной документации на программное обеспечение;
- выполняется конфигурация базового ПО;
- устанавливается и настраивается специализированное ПО;
- проводится настройка контроля функционирования программно-аппаратного комплекса Системы.

2.1. Конфигурация и развертывание базового и специализированного ПО

2.1.1. Конфигурация и развертывание PostgreSQL

Для установки необходимых пакетов PostgreSQL необходимо выполнить команды:

dnf install armadillo-9.900.3-3.el7 arpack-3.1.5-4.el7 atlas-3.10.1-14.el7 blas-3.9.0-1.el7 boost-serialization-1.73.0-4.el7 cfitsio-3.360-5.el7 cpptest-1.1.1-11.el7 freexl-1.0.4-3.el7 gdal-libs-3.3.2-2.el7 geos-3.7.2-1.el7 hdf5-1.10.6-1.el7 hdf-libs-4.2.15-1.el7 lapack-3.9.0-1.el7 libaec-1.0.4-2.el7 libdap-3.20.4-1.el7 libgeotiff-1.6.0-3.el7 libgfortran-8.3.1-6.el7 libgta-1.0.9-3.el7 libkml-1.3.0-3.el7 libpq-13.4-1.el7 libquadmath-8.3.1-6.el7 libspatialite-5.0.0-3.el7 netcdf-4.7.3-2.el7 ogdi-4.1.0-1.el7 openblas-0.3.9-2.el7 openblas-openmp-0.3.9-2.el7 openblas-serial-0.3.9-2.el7 postgis-3.1.4-2.el7 postgresql-12.7-1.el7 postgresql-server-12.7-1.el7 proj-8.0.0-1.el7 protobuf-c-1.3.1-2.el7 SFCGAL-1.3.10-1.el7 SuperLU-5.2.1-4.el7 unixODBC-2.3.7-2.el7 uriparser-0.7.5-11.el7 xerces-c-3.2.3-1.el7

systemctl enable postgresql

systemctl start postgresql

systemctl status postgresql

Основные настройки PostgreSQL находятся в двух файлах: postgresql.conf и pg_hba.conf. Эти файлы находятся в директории расположения БД, например /var/lib/pgsql/data. Уточнить расположение можно командой psql -U postgres -c 'SHOW data_directory;'.

В файле postgresql.conf содержатся параметры базы данных, а в pg_hba.conf – настройки доступа и идентификации пользователей.

Для прикладного программного обеспечения BP-MODULE должно быть развернуто два экземпляра БД. Один используется для обеспечения работы сервера аутентификации, другой для обеспечения работы остальных приложений.

Настройка PostgreSQL для обеспечения работы приложений

Для обеспечения работы приложений BP-MODULE на сервере должна быть установлена СУБД PostgreSQL версии 12.7 с расширением PostGIS версии не менее 3.0.

Конфигурационные файлы СУБД PostgreSQL должны иметь следующие настройки.

Конфигурационный файл postgresql.conf:

```
log_timezone = 'Etc/UTC'
datestyle = 'iso, mdy'
timezone = 'Etc/UTC'
listen_addresses = '*'
port = 10265
wal_buffers = 1MB
max_connections = 300
shared_buffers = 6GB
effective_cache_size = 18GB
maintenance_work_mem = 1536MB
checkpoint_completion_target = 0.9
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 4
effective_io_concurrency = 2
work_mem = 6990kB
min_wal_size = 1GB
max_wal_size = 4GB
max_worker_processes = 6
max_parallel_workers_per_gather = 3
max_parallel_workers = 6
max_parallel_maintenance_workers = 3
shared_preload_libraries = '/usr/lib64/pgsql/passwordcheck.so'
authentication_timeout = 1min
password_encryption = scram-sha-256
log_destination = 'syslog, stderr'
syslog_facility = 'LOCAL2'
log_min_messages = warning
log_min_error_statement = warning
log_checkpoints = on
log_connections = on
log_disconnections = on
log_duration = off
log_hostname = on
log_statement = 'all'
log_line_prefix = '|%m|%d|%u|%h|'
log_lock_waits = off
log_error_verbosity = verbose
log_temp_files = -1
log_replication_commands = off
log_filename = postgresql-12-main.log
log_directory = '/var/log/postgresql/'
logging_collector = on
```

В конфигурационном файле pg_hba.conf необходимо разрешить доступ модулям BP-MODULE. Настройки конфигурационного pg_hba.conf:

```
# Database administrative login by Unix domain socket
local   all             postgres                                peer
# TYPE  DATABASE        USER            ADDRESS                 METHOD
# "local" is for Unix domain socket connections only
local   postgres        postgres                                peer
local   bpmodule        postgres                                peer
# IPv4 local connections:
```

```
host    all             all             0.0.0.0/24              md5
```

После запуска сервиса БД необходимо выполнить процедуру создания базы данных, включить расширение PostGIS и создать пользователей (роли) для подключения к базе данных. Данные процессы выполняются командами:

1. Создание базы данных осуществляется следующей SQL командой:

   CREATE DATABASE “наименование БД”;

Примечание: *в BP-MODULE используется наименование базы данных bpmodule*

2. Включение расширения в БД осуществляется следующей SQL командой:

   CREATE EXTENSION postgis;

3. Создание пользователей (роли) осуществляется следующей SQL командой:

```
CREATE ROLE имя_роли WITH
  LOGIN
  NOSUPERUSER
  NOCREATEDB
  NOCREATEROLE
  INHERIT
  NOREPLICATION
  CONNECTION LIMIT -1
  PASSWORD 'пароль';
```

Необходимо выполнить команду для каждой из 3-х ролей (приведены ниже):

- eservice_gis – используется картографическим сервером geoserver
- eservice_tis – основная роль для всех модулей BP-MODULE

После выполнения настроек БД, приведенных выше, необходимо выполнить скрипты первоначального наполнения БД.

Скрипты init-data.sql, init-schema-routing.sql находятся в архиве init-schema.zip, который передается в составе инсталляционных файлов Системы.

Для выполнения скриптов необходимо выполнить команду:

psql -d bpmodule -p 10265 -f имя_файла.sql

Настройка PostgreSQL для обеспечения работы сервера идентификации WSO2 IS

Необходимо выполнить процедуру создания базы данных и создать пользователя (роль) для подключения к базе данных. Данные процессы выполняются командами:

1. Создание базы данных под сервер идентификации осуществляется следующей SQL командой:

   CREATE DATABASE “wsodb”;

2. Создание пользователя (роли) осуществляется следующей SQL командой:

```
CREATE ROLE wso2isadmin WITH
  LOGIN
```

NOSUPERUSER
NOCREATEDB
NOCREATEROLE
INHERIT
NOREPLICATION
CONNECTION LIMIT -1
PASSWORD 'пароль';

После выполнения настроек БД, приведенных выше, необходимо выполнить скрипты первоначального наполнения БД.

Скрипты uma.sql, shared.sql, metrics.sql, identity.sql, consent.sql, bps.sql находятся в архиве init-schema-wso2is.zip, который передается в составе инсталляционных файлов Системы.

Для выполнения скриптов необходимо выполнить команду:

psql -d wsodb -p 10265 -f имя_файла.sql

2.1.2. Конфигурация платформы контейнеризации Docker CE

Платформа контейнеризации приложений Docker CE используется для упаковки сервисов ППО BP-MODULE и всех его зависимостей вместе, и запуска их в изолированных контейнерах.

Контейнер (Docker Container) представляет собой окружение для выполнения какого-либо одного процесса. Контейнер создается на основе образа (Docker Image). Образ Docker – это неизменяемый пакет файлов, содержащий библиотеки, зависимости, инструменты и другие файлы, необходимые для запуска приложения. Сервисы ППО BP-MODULE поставляются в виде готовых образов, упакованных в tar архив.

Для базовой установки необходимых пакетов Docker нужно выполнить следующие команды:

dnf install -y containerd-1.4.3-1.20201221.el7.x86_64 python3-attrs-20.3.0-1.el7.noarch python3-docopt-0.6.2-8.el7.noarch container-selinux-2.152.0-1.el7.noarch python3-cached_property-1.5.2-1.el7.noarch python3-jsonschema-3.2.0-1.el7.noarch docker-ce-20.10.1-2.el7.x86_64 python3-chardet-3.0.4-5.el7.noarch python3-rpm-4.13.90-11.el7.x86_64 docker-ce-cli-20.10.1-1.el7.x86_64 python3-docker-4.4.1-1.el7.noarch python3-texttable-1.6.2-1.el7.noarch docker-ce-rootless-extras-20.10.1-1.el7.x86_64 python3-dockerpty-0.4.1-10.el7.noarch python3-websocket-client-0.57.0-2.el7.noarch docker-compose-1.28.2-1.el7.noarch python3-docker-pycreds-0.4.0-1.el7.noarch runc-1.0.0-1.dev.git5efdc0c.el7.x86_64 fuse-overlayfs-1.3+dev-1.dev.gitbccc81e.el7.x86_64 slirp4netns-1.1.8-1.el7.x86_64 python3-pyrsistent-0.14.2-3.el7.x86_64 python3-dotenv-0.14.0-1.el7.noarch libslirp-4.4.0-1.el7.x86_64 python3-click-7.0-3.el7.noarch python3-pynacl-1.2.0-3.el7.x86_64 python3-bcrypt-3.1.4-3.el7.x86_64 python3-paramiko-2.4.2-2.el7.noarch

systemctl enable docker

systemctl enable containerd.service

systemctl start docker

Перед созданием контейнеров в локальное хранилище docker необходимо загрузить все необходимые файлы образов. Для этого файлы архивов копируются на сервер.

Файлы образов, находятся в архиве images.zip, который передается в составе инсталляционных файлов Системы.

Загрузка с образов из переданного архива в локальное хранилище docker на сервере осуществляется командой docker load:

```
docker load -i имя_файла.tar
```

Для создания, конфигурирования и запуска контейнеров используется утилита Docker Compose, являющаяся инструментом из состава Docker. Утилита Docker Compose использует конфигурационный файл docker-compose.yml (формат YAML). Расположение файла может быть любое, но все команды docker-compose должны выполняться в той же директории, где расположен файл docker-compose.yml. На сервере ППО BP-MODULE файл docker-compose.yml располагается по пути /opt/bpmodule/.

***Примечание**: В параметрах конфигурационного файла можно использовать переменные окружения. Используется синтаксис: ${VARIABLE}. Кроме переменных окружений операционной системы утилита Docker Compose считывает переменные в файле ".env". Этот файл должен располагаться в рабочей директории вместе с файлом docker-compose.yml.*

Конфигурирование приложений внутри контейнеров осуществляется путем передачи параметров через переменные окружения или путем монтирования конфигурационных файлов или директорий в контейнер.

Предварительно конфигурационные файлы должны быть загружены из папок архива на советующие сервера в папку каталога /opt/bpmodule/. Конфигурационные файлы контейнеров находятся в архиве config.zip, который передается в составе инсталляционных файлов Системы.

Далее необходимо провести процедуры донастройки и запуска контейнеризации в соответствии со следующим порядком:

1. Процедуры проведения предварительной донастройки конфигурации сервера ГИС BP-MODULE приведены в разделе 2.1.3.
2. Процедуры проведения предварительной донастройки конфигурации сервера аутентификации BP-MODULE приведены в разделе 2.1.4.
3. Процедуры проведения предварительной донастройки конфигурации сервера контейнеризации BP-MODULE заключаются в последовательной настройке конфигурационных файлов контейнеров программного обеспечения:
   - Конфигурация REDIS приведена в разделе 2.1.5.
   - Конфигурация GDAL приведена в разделе 2.1.6.
   - Конфигурация Tomcat приведена в разделе 2.1.7.
   - Конфигурация OpenJDK приведена в разделе 2.1.8.
   - Конфигурация Nginx для сервера контейнеризации приведена в разделе 2.1.9.
   - Конфигурация специализированного ППО BP-MODULE для сервера контейнеризации приведена в разделе 2.1.10.

После выполнения процедур донастройки конфигурационных файлов сервера контейнеризации, для запуска контейнеров Docker на сервере последовательно необходимо выполнить следующие команды:

1. Команда создания сети:

```
docker network create bp-module-network
```

2. Команда запуска контейнеров:

```
docker-compose up -d
```

2.1.3. Конфигурация и запуск контейнера GeoServer

Программное обеспечение GeoServer разворачивается в docker контейнере на сервере ГИС BP-MODULE в соответствии настройками, приведенными в разделе 2.1.2.

***Примечание**: В служебном каталоге с общим именованием /opt/bpmodule (требования к созданию служебного каталога приведено в разделе 2.1.2) добавляется дополнительный каталог /geoserver/geoserver-data/*

Для конфигурирования программного обеспечения необходимо скопировать файлы из архива geoserver_data.zip (слои, стили, шрифты) в каталог /opt/bpmodule/*geoserver*/*geoserver-data*. Конфигурационные файлы находятся в архиве geoserver_data.zip, который передается в составе инсталляционных файлов Системы.

***Примечание**: Проверить конфигурационные файлы docker-compose.yml и .env., располагающиеся в каталоге* /opt/bpmodule, *в части пути монтирования.*

После выполнения процедур донастройки для запуска контейнеров Docker на сервере последовательно необходимо выполнить следующие команды:

1. Команда создания сети:

```
docker network create bpmodule-network
```

2. Команда запуска контейнеров:

```
docker-compose up -d
```

2.1.4. Конфигурация и запуск контейнера WSO2 IS

Программное обеспечение WSO2 IS разворачивается в docker контейнере на сервере аутентификации BP-MODULE в соответствии настройками, приведенными в разделе 2.1.2.

***Примечание**: В служебном каталоге с общим именованием /opt/wso2is (требования к созданию служебного каталога приведено в разделе 2.1.2) добавляются дополнительные каталоги /wso2is/conf и /wso2is/security.*

Необходимо проверить, что в каталоге для запуска сервера (/opt/wso2is) после выполнения работ в соответствии с п. 2.1.2. присутствуют следующие файлы, и при необходимости выполнить их донастройку:

./wso2is/conf/deployment.toml – файл конфигурации сервер WSO2 IS, дополнительные процедуры настройки/проверки приведены в разделе 2.1.4.1.

./wso2is/security/client-truststore.jks – хранилище доверенных сертификатов, процедура настройки хранилища приведены в разделе 2.1.4.2.
./wso2is/security/wso2carbon.jks – хранилище ключей, процедура настройки хранилища приведены в разделе 2.1.4.2.
./docker-compose.yml – конфигурационный файл docker

2.1.4.1. Подготовка конфигурации сервера при помощи файла deployment.toml

Для корректной работы сервера аутентификации BP-MODULE необходимо проверить/задать пароль администратора в файле deployment.toml.

За данные настройки отвечают следующие настроечные параметры в файле deployment.toml:

Раздел файла [super_admin], параметры:

```
username = "admin"
password = "СМЕНИТЬ ПАРОЛЬ!!!"
```

Раздел файла [database.identity_db], параметры:

```
username = "wso2isadmin"
password = "СМЕНИТЬ ПАРОЛЬ!!!"
```

***Примечание**: Данный пароль задается в процессе выполнения SQL команды при создании пользователя (роли), прописанной в подразделе «Настройки PostgreSQL для обеспечения работы сервера идентификации WSO2 IS» раздела настроек 2.1.1.*

Раздел файла [database.shared_db], параметры:

```
username = "wso2isadmin"
password = "СМЕНИТЬ ПАРОЛЬ!!!"
```

***Примечание**: Данный пароль задается в процессе выполнения SQL команды при создании пользователя (роли), прописанной в подразделе «PostgreSQL для обеспечения работы сервера идентификации WSO2 IS раздела настроек 2.1.1.*

Пример базового конфигурационного файла настроек deployment.toml приведен ниже:

```
[server]
hostname = "bpmodule.technocom.tech"
node_ip = "10.56.63.16"
base_path = "https://$ref{server.hostname}:${carbon.management.port}"

[cors]
allow_any_origin = true
supported_methods = [
    "GET",
    "POST",
    "HEAD",
    "OPTIONS"
]
support_any_header = true
supports_credentials = true

[transport.https.properties]
```

```
proxyPort=443

[admin_console.authenticator.saml_sso_authenticator]
enable=true
priority="1"
assertion_consumer_service_url = "https://bpmodule.technocom.tech/acs"
identity_provider_sso_service_url = "https://bpmodule.technocom.tech/samlsso"

[super_admin]
username = "admin"
password = "СМЕНИТЬ ПАРОЛЬ!!!"
create_admin_account = true

[user_store]
type = "database_unique_id"

[database.identity_db]
type = "postgre"
hostname = "10.56.63.16"
name = "wsodb"
username = "wso2isadmin"
password = "СМЕНИТЬ ПАРОЛЬ!!!"
port = "10265"

[database.identity_db.pool_options]
commitOnReturn="true"
maxActive = "80"
maxWait = "60000"
minIdle = "5"
testOnBorrow = true
validationQuery="SELECT 1; COMMIT"
validationInterval="30000"
defaultAutoCommit=false

[database.shared_db]
type = "postgre"
hostname = "10.56.63.16"
name = "wsodb"
username = "wso2isadmin"
password = "СМЕНИТЬ ПАРОЛЬ!!!"
port = "10265"


[database.shared_db.pool_options]
commitOnReturn="true"
maxActive = "80"
maxWait = "60000"
minIdle = "5"
testOnBorrow = true
validationQuery="SELECT 1; COMMIT"
validationInterval="30000"
defaultAutoCommit=false

[output_adapter.email]
from_address= "support@technocom.tech"
username= "support"
password= "pass"
hostname= "mx.technocom.tech"
port= 25
enable_start_tls= false
enable_authentication= true
```

```
[authentication.consent]
prompt = false

[monitoring.jmx]
rmi_server_start = false

[keystore.primary]
name = "wso2carbon.jks"
password = "wso2carbon"

[user.association]
enable_for_federated_users = true

[system_applications]
read_only_apps = []
```

2.1.4.2. Подготовка хранилища ключей

WSO2 Identity Server по умолчанию использует асимметричное шифрование для аутентификации и защиты данных. Шифрование используется по умолчанию для следующих целей:

- Аутентификация связи по протоколам Secure Sockets Layer (SSL)/Transport Layer Security (TLS).
- Шифрование конфиденциальных данных, таких как простые текстовые пароли, найденные в файлах конфигураций/конфигураций как на уровне продукта, так и на уровне функций продукта.

Хранилище ключей – это хранилище (защищенное паролем), в котором хранятся ключи и сертификаты цепочки доверия. По умолчанию это wso2carbon.jks.

Хранилище доверенных сертификатов – это еще один репозиторий, защищенный паролем (аналогично хранилищу ключей), в котором хранятся цифровые сертификаты. По умолчанию это client-truststore.jks. Эти сертификаты могут быть одним из следующих:

- Сертификаты доверенных третьих сторон, с которыми программная система намеревается взаимодействовать напрямую.
- Сертификаты известных центров подписи сертификатов (ЦС), которые можно использовать для проверки подлинности ненадежных третьих лиц, с которыми связываются.

Чтобы сформировать правильно файл хранилища ключей и хранилища доверенных сертификатов, необходимо выполнить процедуры добавления соответствующих сертификатов.

***Примечание**: Файлы ca.crt, ca.key и ca_bundle.crt берутся из предоставленного сертификата заказчиком сертификата.*

Процедура добавления выполняется следующим образом:

1. Создайте .pfx файл из сертификата (ca.crt), приватного ключа (ca.key) и цепочки серификатов ЦС (ca_bundle.crt) следующей командой:

   openssl pkcs12 -export -out bpmodule.pfx -inkey ca.key -in ca.crt -certfile ca_bundle.crt

2. Добавьте созданную пару ключей в хранилище ключей wso2carbon.jks следующей командой:

   keytool -importkeystore -srckeystore bpmodule.pfx -srcstoretype pkcs12 -destkeystore ./wso2is/conf/security/wso2carbon.jks -deststoretype JKS

3. Переименуйте добавленную пару ключей следующей командой:

   keytool -changealias -alias 1 -destalias wso2carbon -keystore ./wso2is/conf/security/wso2carbon.jks

4. Добавьте сертификат в хранилище доверенных сертификатов client-truststore.jks следующей командой:

   keytool -import -trustcacerts -alias wso2carbon -file ca.crt -keystore ./wso2is/conf/security/client-truststore.jks

***Примечание**: Утилита keytool входит в состав •OpenJDK и будет доступна только после установки OpenJDK на сервере. Установка OpenJDK выполняется в соответствии с общей инструкцией по развертыванию на операционной системе Linux.*

2.1.4.3. Запуск контейнера сервера WSO2 IS

Перед запуском контейнера сервера аутентификации WSO2 IS необходимо проверить конфигурационные файлы docker-compose.yml и .env., располагающиеся в каталоге /opt/wso2is, в части его настроек.

Конфигурационный файл docker-compose.yml для запуска контейнера сервера аутентификации:

```
version: '3.7'
services:
  wso2is:
    image: tkrepo.technocom.tech/technocom/wso2is:latest
    container_name: wso2is
    environment:
      - TZ= Asia/Yekaterinburg
    volumes:
      - ./wso2is/conf/deployment.toml:/home/wso2carbon/wso2is-5.11.0/repository/conf/deployment.toml
      - ./wso2is/conf/userstores:/home/wso2carbon/wso2is-5.11.0/repository/deployment/server/userstores/
      - ./wso2is/logs:/home/wso2carbon/wso2is-5.11.0/repository/logs/
      - ./wso2is/security:/home/wso2carbon/wso2is-5.11.0/repository/resources/security/
    networks:
      bpmodule-network:
        ipv4_address: 172.18.0.2
        aliases:
          - bpmodule.technocom.tech
    expose:
      - "9763"
      - "9443"
    ports:
      - 9763:9763
      - 443:9443
    restart: unless-stopped

networks:
  bpmodule-network:
```

```
external: true
```

После выполнения процедур донастройки для запуска контейнеров Docker на сервере аутентификации BP-MODULE последовательно необходимо выполнить следующие команды:

1. Команда создания сети:

   docker network create bpmodule-network --subnet 172.18.0.0/24

2. Команда запуска контейнеров:

   docker-compose up -d

При успешном выполнении запуска сервера выводится следующая информация:

В логах нет ошибок, по завершении запуска отображаются примерно следующие строки с URL консолей:

```
gaz-wso2is    | [2022-02-11 05:13:36,146] []  INFO
{org.wso2.carbon.ui.internal.CarbonUIServiceComponent} - Mgt Console URL  : https://
bpmodule.technocom.tech:443/carbon/
gaz-wso2is    | [2022-02-11 05:13:36,454] []  INFO
{org.wso2.identity.apps.common.internal.AppsCommonServiceComponent} - My Account URL : https://
bpmodule.technocom.tech/myaccount
gaz-wso2is    | [2022-02-11 05:13:36,455] []  INFO
{org.wso2.identity.apps.common.internal.AppsCommonServiceComponent} - Console URL :
https://bpmodule.technocom.tech/console
```

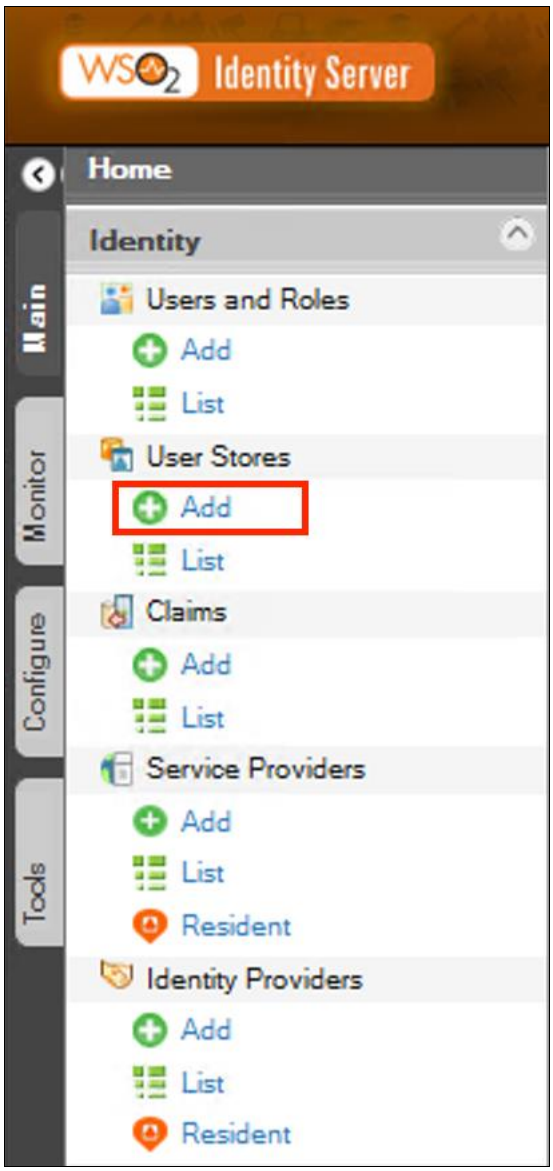
2.1.4.4. Настройка подключения к Active Directory

Сервер идентификации должен быть подключен к ldap каталогу Active Directory (далее AD). Для этого необходимо настроить хранилище пользователей (User Store). На каждый домен нужно создать отдельное хранилище. Для подключения к ldap каталогу необходима сервисная учетная запись в Active Directory. В действующей реализации для взаимодействия с хранилищем пользователей AD используется следующая учетная запись из каталога AD:

bpmodule

Создание хранилища пользователей осуществляется в веб-консоли сервера идентификации под учетной записью администратора. Веб-консоль будет доступна после настройки и запуска веб-сервера nginx (см раздел 2.1.11).

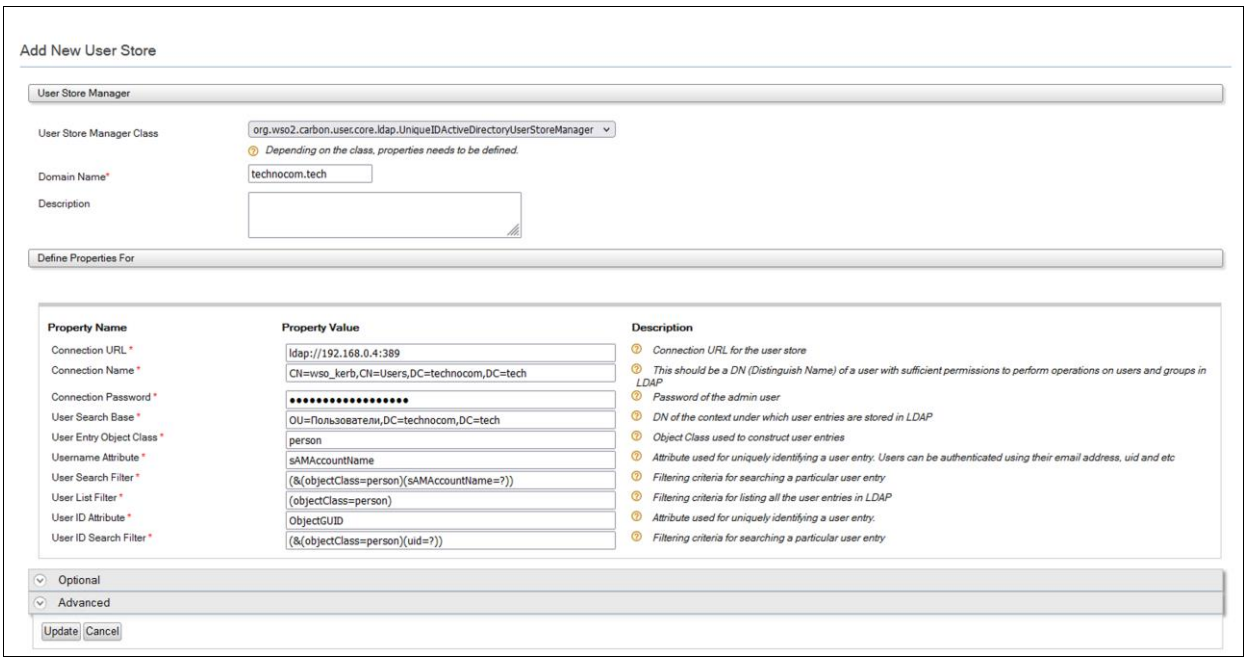
Перейдите к **Main> User Stores** и нажмите **Add**.

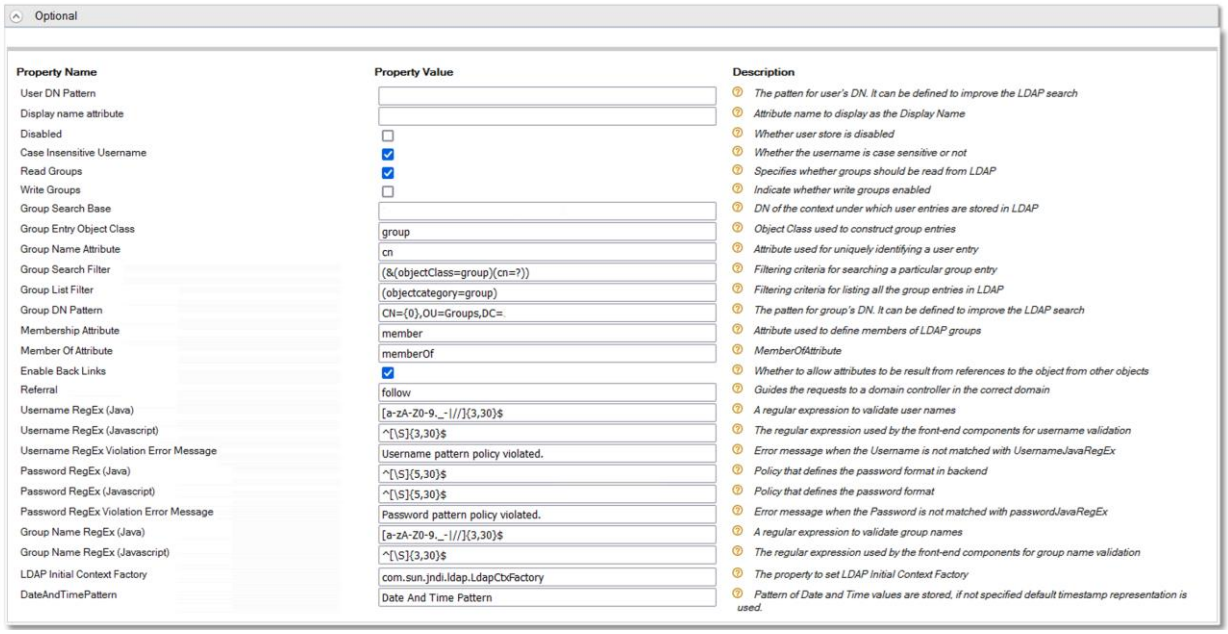


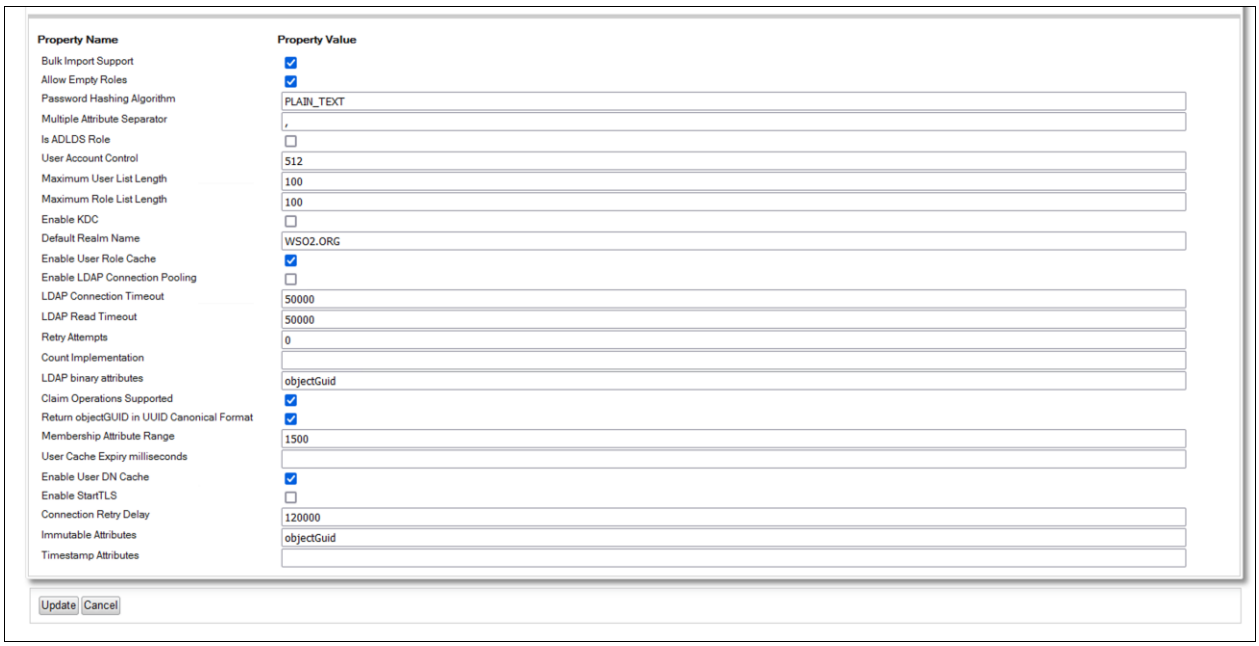
Замените значения параметров, указанные в фигурных скобках, на валидные для вашего AD.

Параметр	Значение	Описание
User Store Manager Class	org.wso2.carbon.user.core.ldap.UniqueIDActiveDirectoryUserStoreManager	Используется для настройки доменной службы Active Directory
Domain Name	Example.local	Домен
Description	любое	Описание подключения
Connection URL	ldap://dc1.example.local:389	URL-адрес подключения для хранилища пользователей
Connection Name	cn=bpmodule,ou=groups,ou=Service Accounts,ou=Datacenter,dc=example,dc=local	Это должно быть DN (отличительное имя) пользователя с достаточными разрешениями для выполнения операций с пользователями и

Параметр	Значение	Описание
		группами в LDAP
Connection Password	пароль	Пароль сервисной учетной записи, указаной в Connection Name
User Search Base	dc=dc1,dc=example,dc=local	DN контекст в LDAP, в котором записи осуществляется поиск пользователей
User Entry Object Class	Person	Класс объекта, используемый для создания пользовательских записей
Username Attribute	sAMAccountName	Атрибут, используемый для уникальной идентификации записи пользователя
User Search Filter	(&(objectClass=person)(sAMAccountName=?))	Критерии фильтрации для поиска определенной записи пользователя
User List Filter	(objectClass=person)	Критерии фильтрации для перечисления всех пользовательских записей в LDAP
User ID Attributer	ObjectGUID	Атрибут, используемый для уникальной идентификации записи пользователя
User ID Search Filter	(&(objectClass=person)(uid=?))	Критерии фильтрации для поиска определенной записи пользователя







2.1.4.5. Настройка WSO2 IS с IWA (Kerberos) в качестве аутентификатора

Добавьте запись в DNS, чтобы сопоставить IP-адрес веб сервера с именем хоста kproweb01-p(kproweb01-t).

Для подключения по протоколу kerberos необходима сервисная учетная запись в Active Directory. В действующей реализации используется следующая учетная запись из каталога AD:

bpmodule

Выполните следующие команды, чтобы зарегистрировать WSO2 IS в качестве субъекта-службы в Active Directory.

setspn -A HTTP/bpmodule.technocom.tech bpmodule

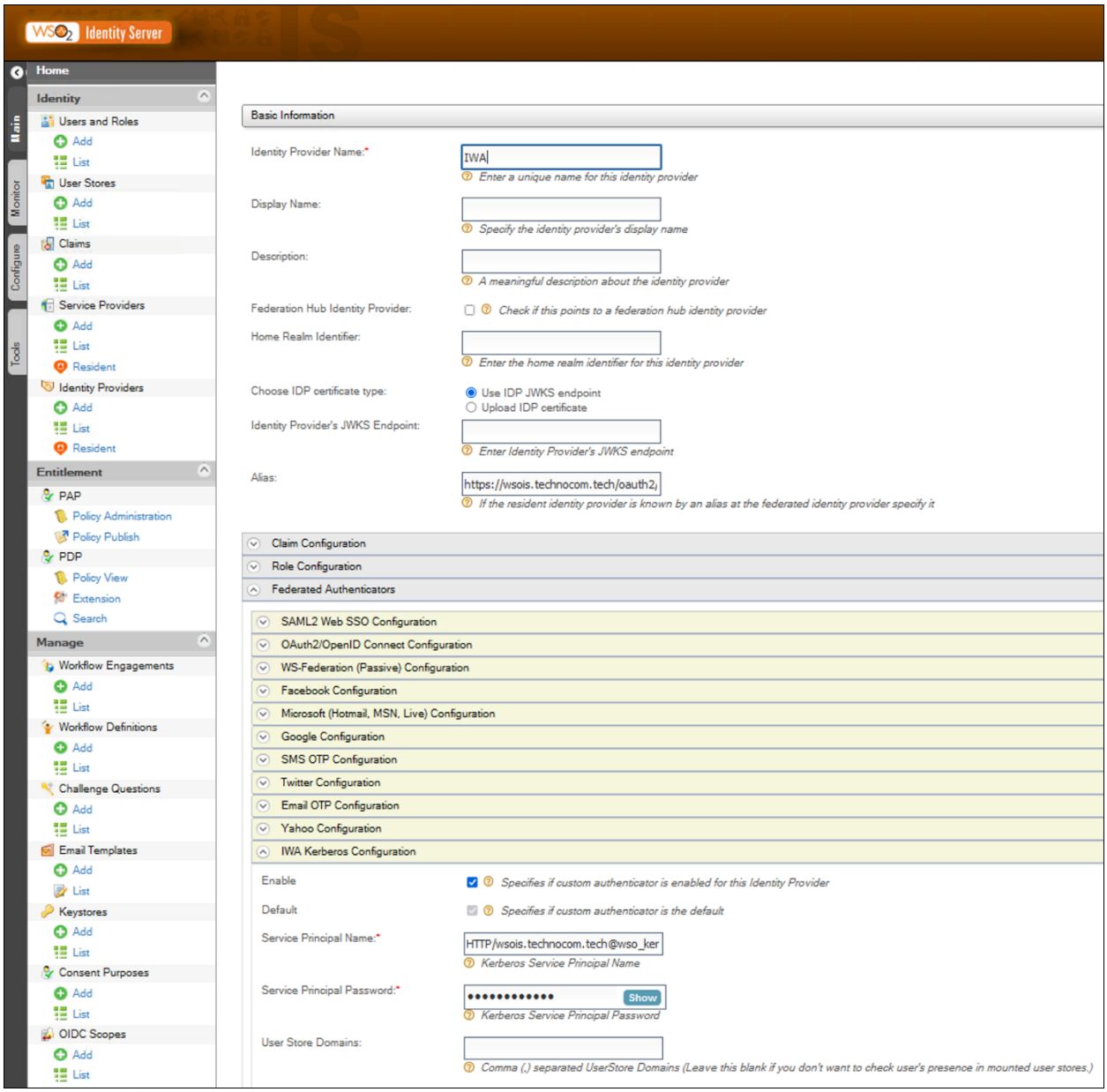
где HTTP – класс службы (не путать с HTTP протоколом),

bpmodule – логин сервисной учетной записи

Далее необходимо настроить провайдера идентификации (Identity Provider). Настройка осуществляется в веб-консоли сервера идентификации под учетной записью администратора. Веб-консоль будет доступна после настройки и запуска веб-сервера nginx (см раздел 2.1.11).

Войдите в консоль управления.

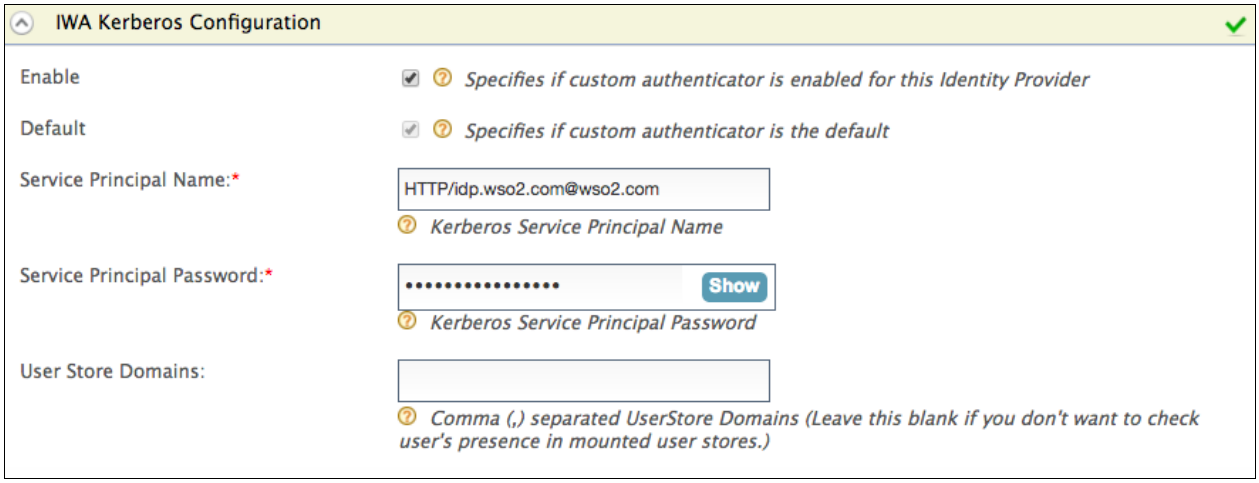
1. Перейдите к **Main> Identity Providers** и нажмите **Add** . Введите имя поставщика удостоверений.
2. Разверните раздел **Federated Authenticators** , а затем разверните **IWA Federated Configuration** .



3. Заполните поля следующим образом:

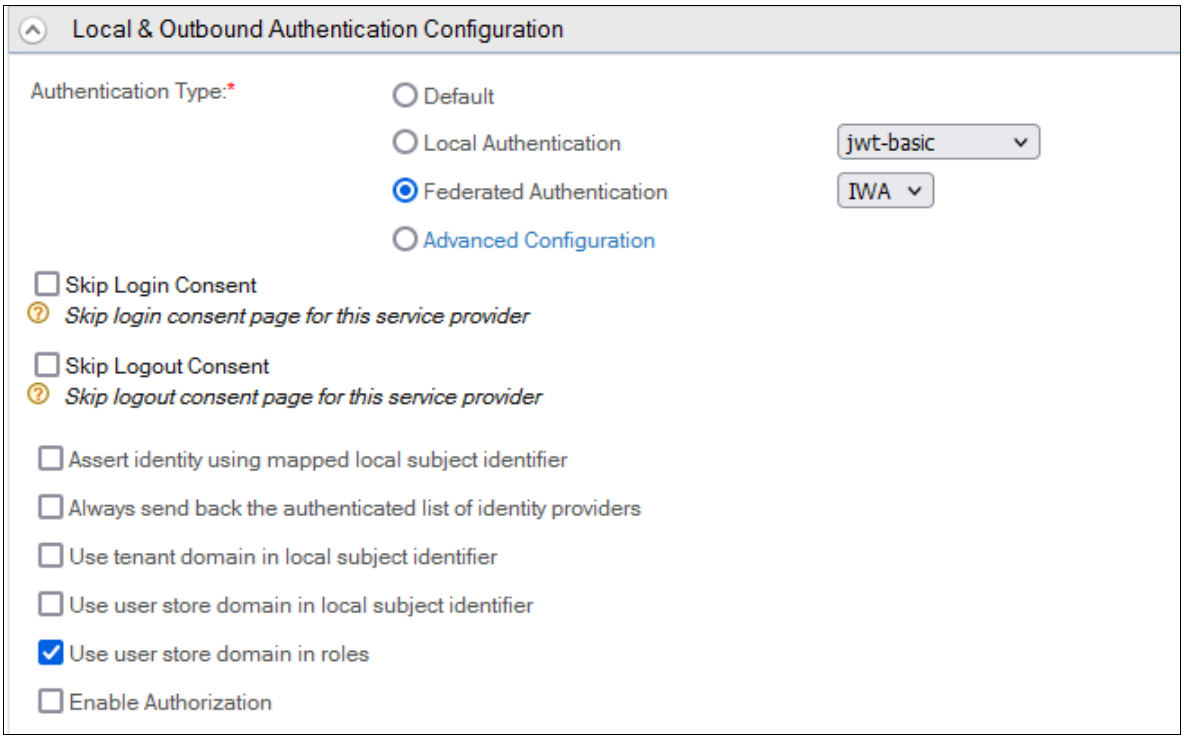
Поле	Описание	Образец значения
Enable	Enable this to enable a custom authenticator for the identity provider.	Выбрано

Поле	Описание	Образец значения
Server Principal Name	Имя участника-службы, которое вы зарегистрировали ранее. В формате: Имя_службы@логин	HTTP/bpmodule.technocom.tech @bpmodule
Server Principal Password	Пароль должен сервисной учетной записи	
User Store Domains	Имя хранилища пользователей. Если несколько, то перечисляются через запятую.	Dc1.example.local



Сохраните настройки провайдера.

Для использования IWA провайдера убедитесь, что он установлен в качестве **Federated Authentication** в Main -> Service provider -> <bpmodule> -> Local & Outbound Authentication Configuration.

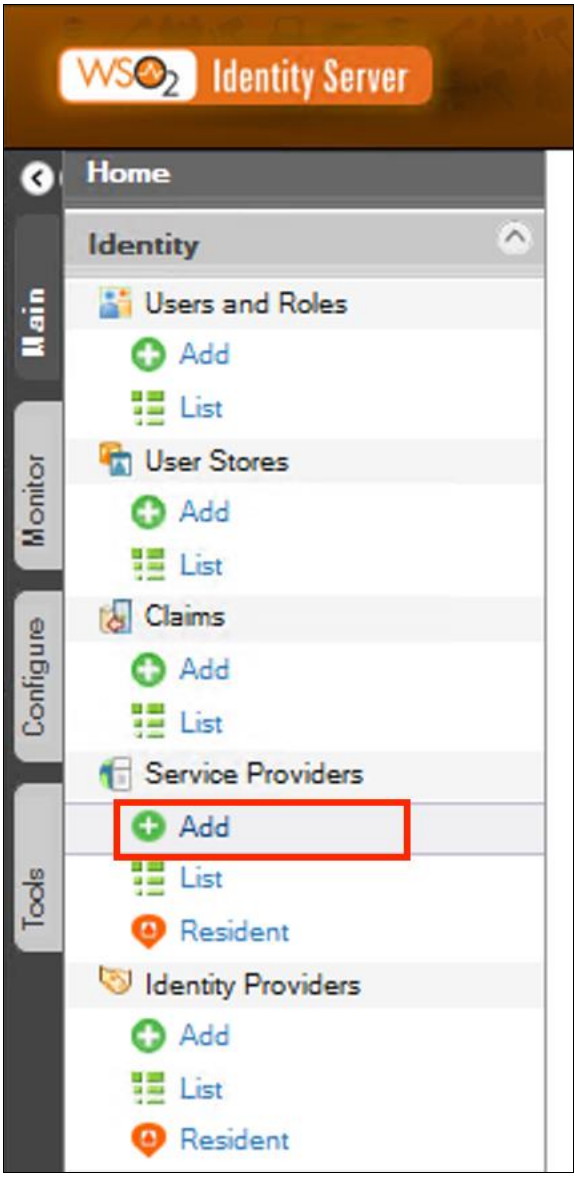


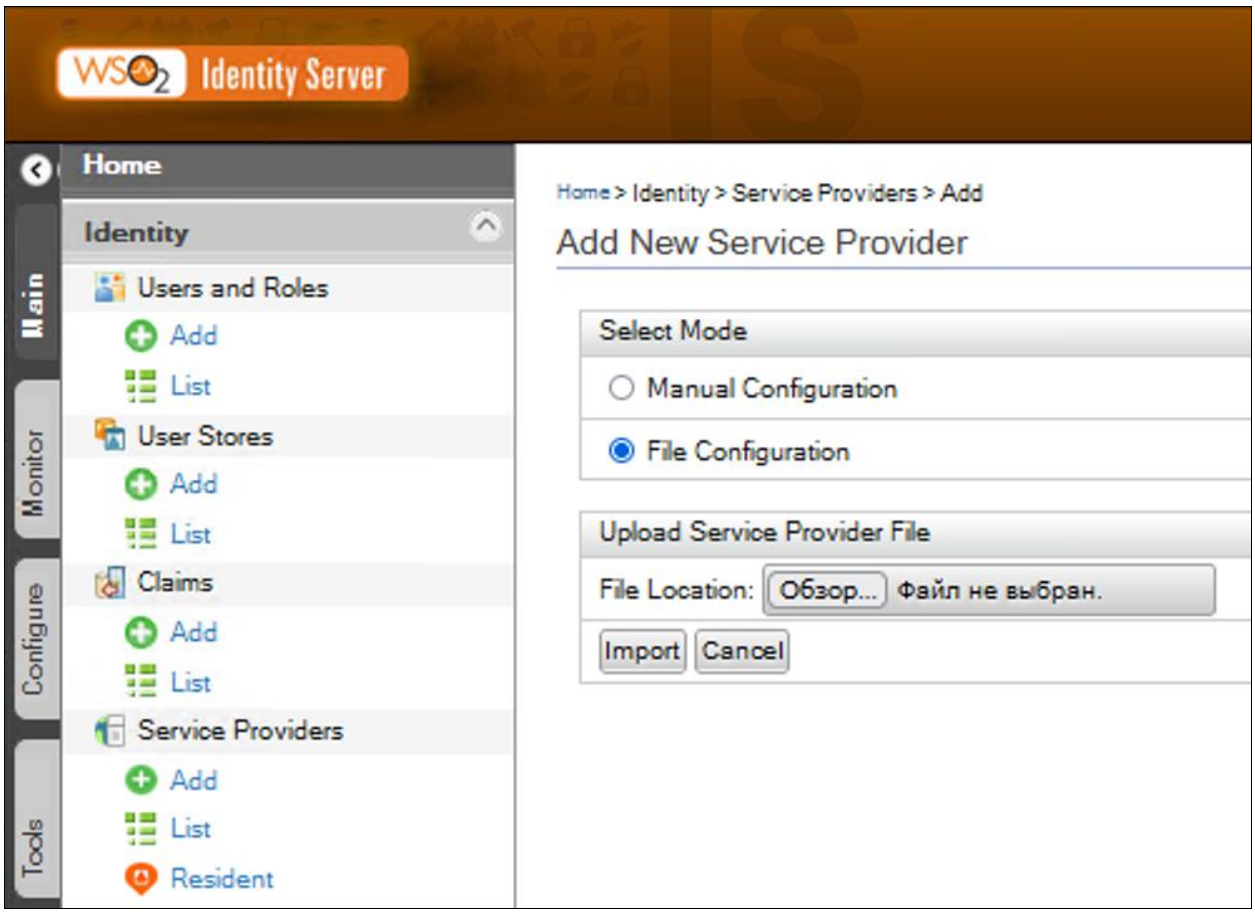
2.1.4.6. Настройка поставщика услуг аутентификации

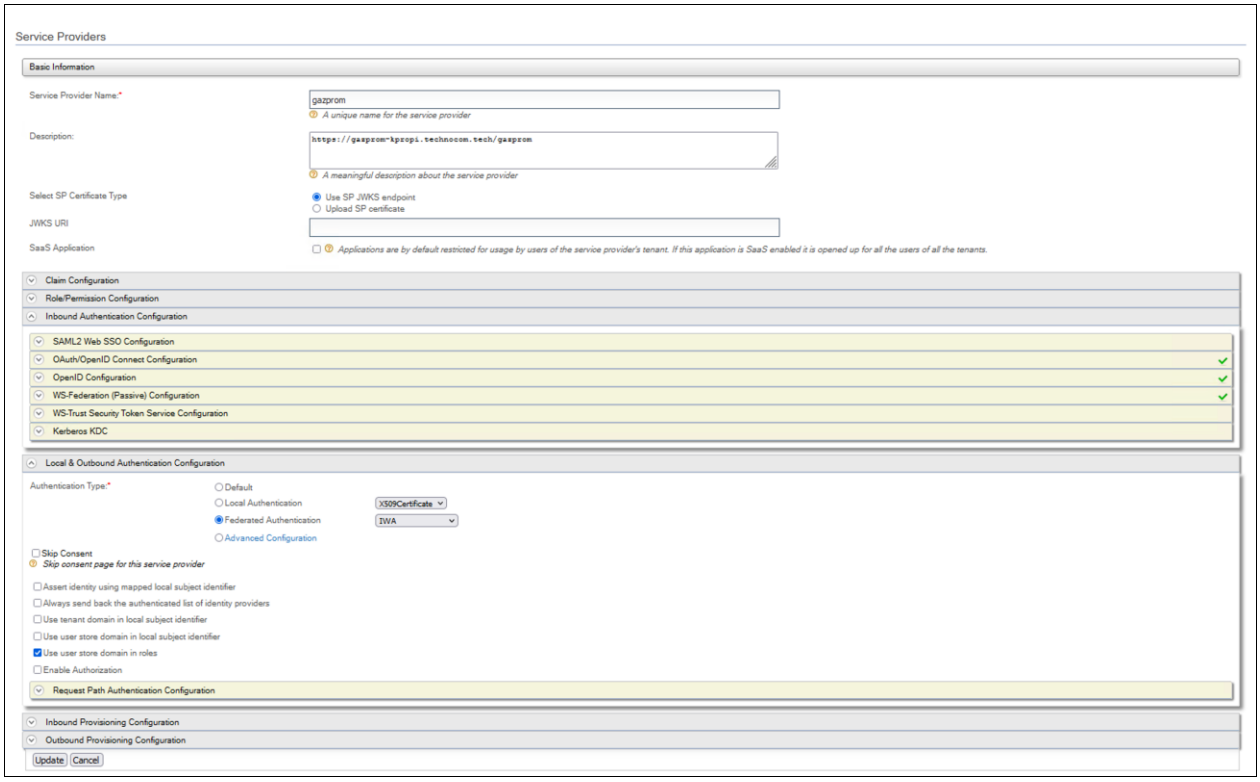
Раздел Service Providers содержит основные настройки о способе аутентификации пользователей в приложениях.

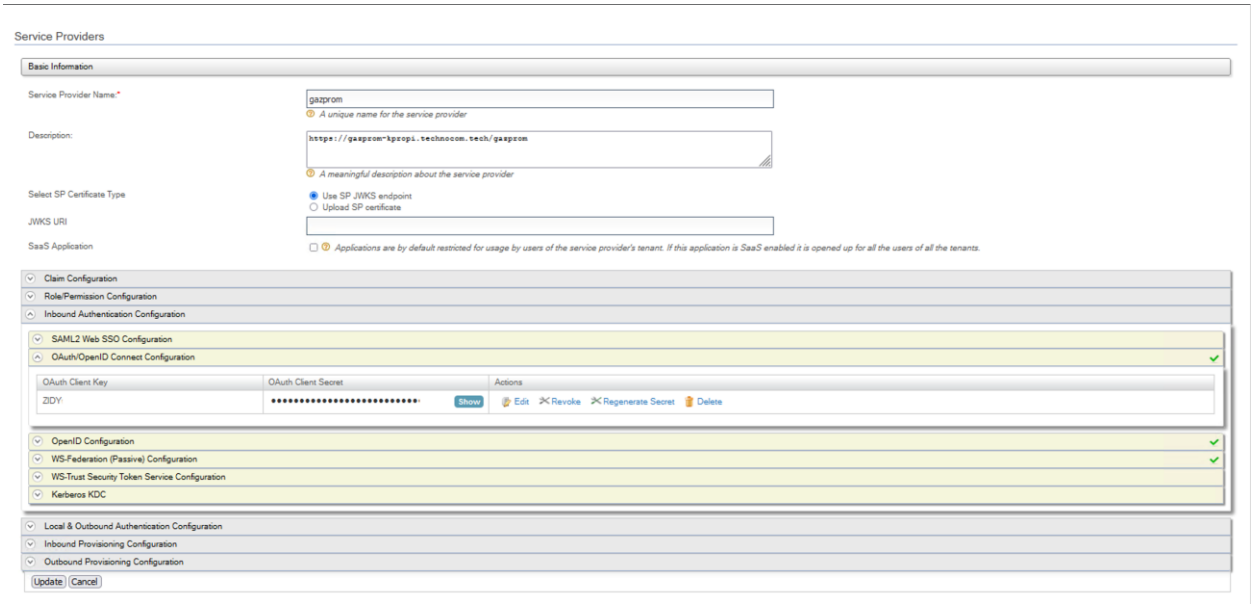
Настройка сервис провайдера осуществляется в веб-консоли сервера идентификации, под учетной записью администратора. Веб-консоль будет доступна после настройки и запуска веб-сервера nginx (см раздел 2.1.11).

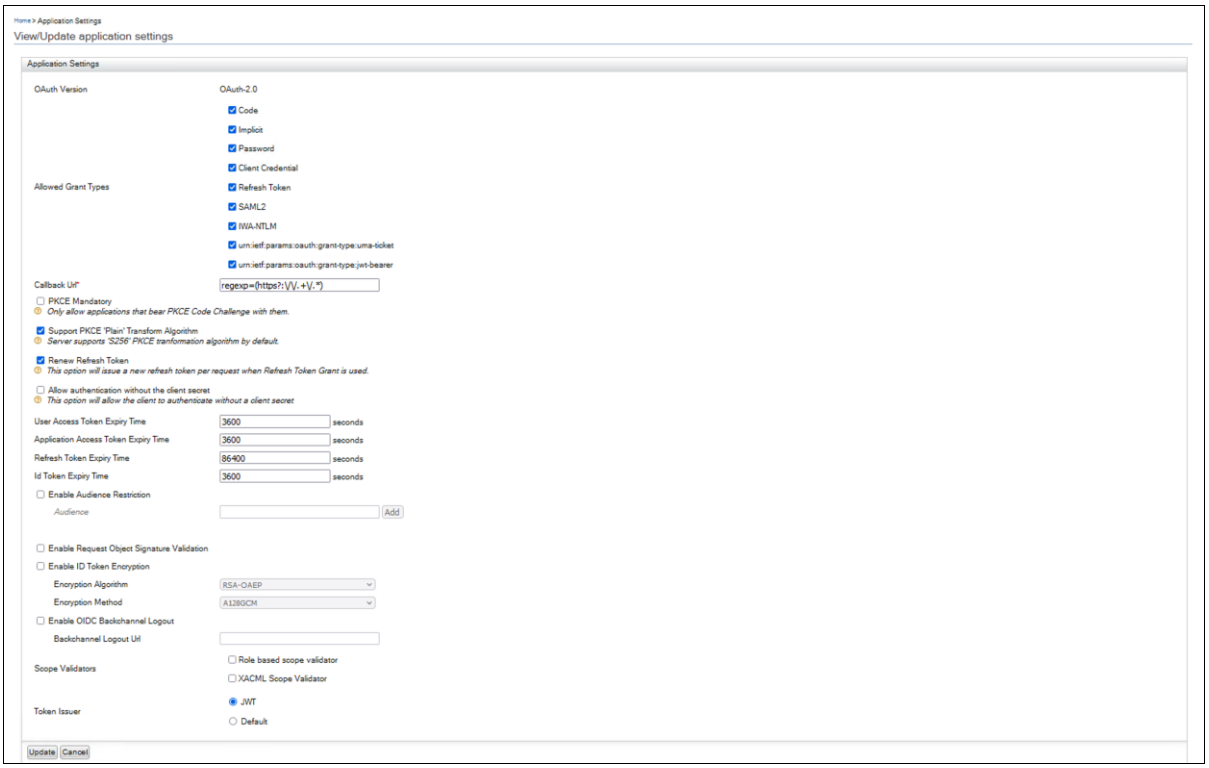
В меню Service Providers выберете Add и загрузите конфигурацию из файла kpropi.xml (находится в архиве configs.zip)




WSO2 Identity Server
Home
Identity
Users and Roles
Add
List
User Stores
Add
List
Claims
Add
List
Service Providers
Add
List
Resident
Main
Monitor
Configure
Tools
Home > Identity > Service Providers > Add
Add New Service Provider
Select Mode
Manual Configuration
File Configuration
Upload Service Provider File
File Location:
Обзор...
Файл не выбран.
Import
Cancel


Service Providers
Basic Information
Service Provider Name:*
gazprom
A unique name for the service provider
Description:
https://gazprom-kpropi.technocom.tech/gazprom
A meaningful description about the service provider
Select SP Certificate Type
Use SP JWKS endpoint
Upload SP certificate
JWKS URI
SaaS Application
Applications are by default restricted for usage by users of the service provider's tenant. If this application is SaaS enabled it is opened up for all the users of all the tenants.
Claim Configuration
Role/Permission Configuration
Inbound Authentication Configuration
SAML2 Web SSO Configuration
OAuth/OpenID Connect Configuration
OpenID Configuration
WS-Federation (Passive) Configuration
WS-Trust Security Token Service Configuration
Kerberos KDC
Local & Outbound Authentication Configuration
Authentication Type:*
Default
Local Authentication
X509Certificate
Federated Authentication
IWA
Advanced Configuration
Skip Consent
Skip consent page for this service provider
Assert identity using mapped local subject identifier
Always send back the authenticated list of identity providers
Use tenant domain in local subject identifier
Use user store domain in local subject identifier
Use user store domain in roles
Enable Authorization
Request Path Authentication Configuration
Inbound Provisioning Configuration
Outbound Provisioning Configuration
Update
Cancel



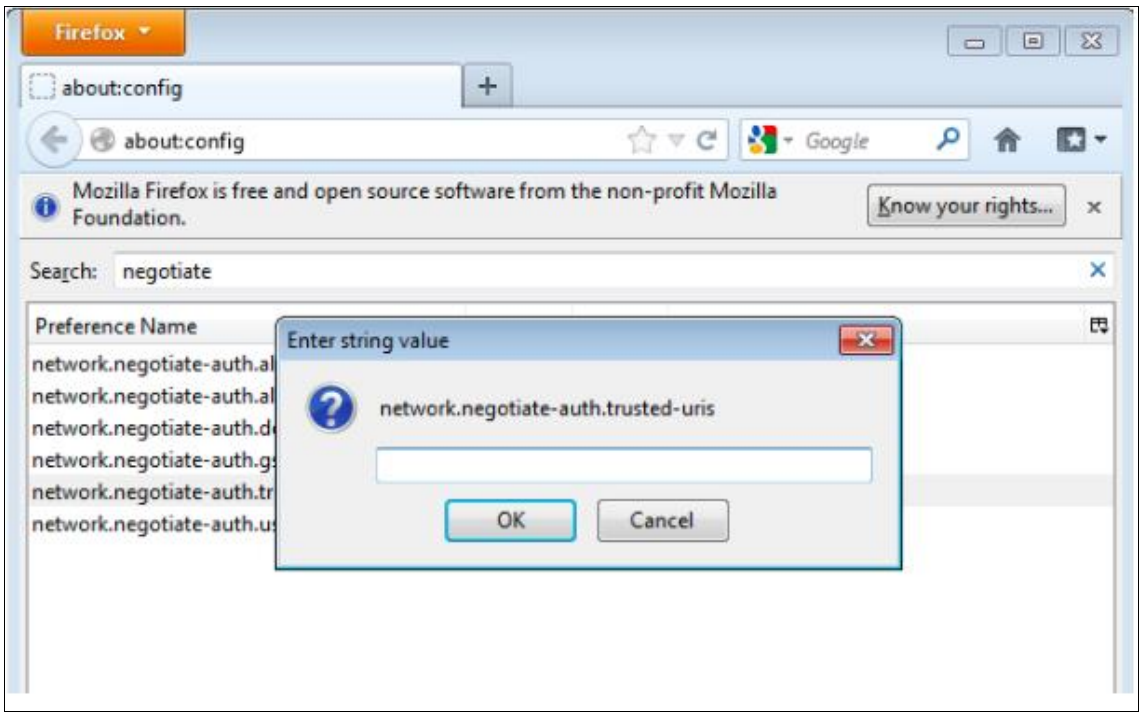


2.1.4.7. Настройка браузера для поддержки Kerberos и NTLM (настройка рабочих мест пользователей)

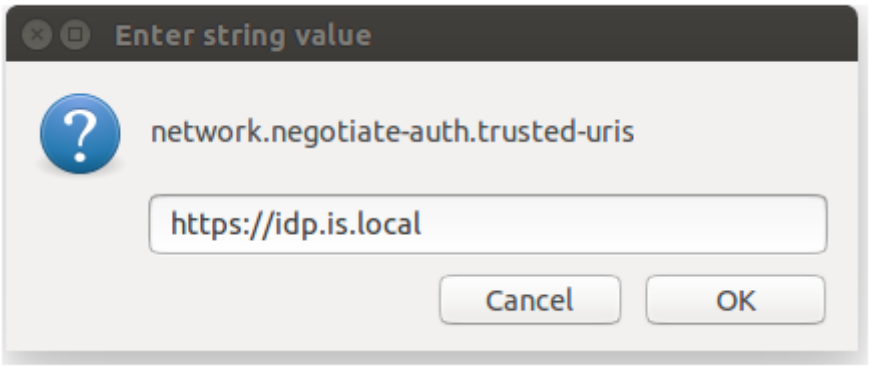
Настройка Firefox

Введите about:config адресную строку, проигнорируйте предупреждение и продолжите, это отобразит дополнительные настройки Firefox.

В строке поиска найдите ключ " network.negotiate-auth.trusted-uris.



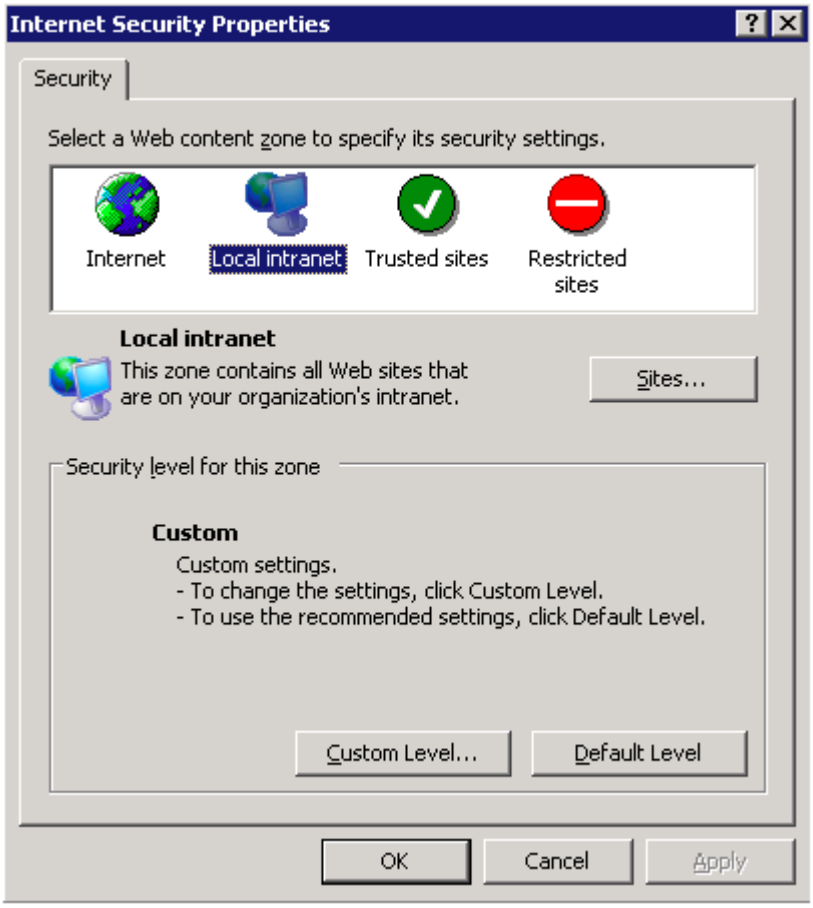
Добавьте URL-адрес WSO2 Identity Server и нажмите OK.



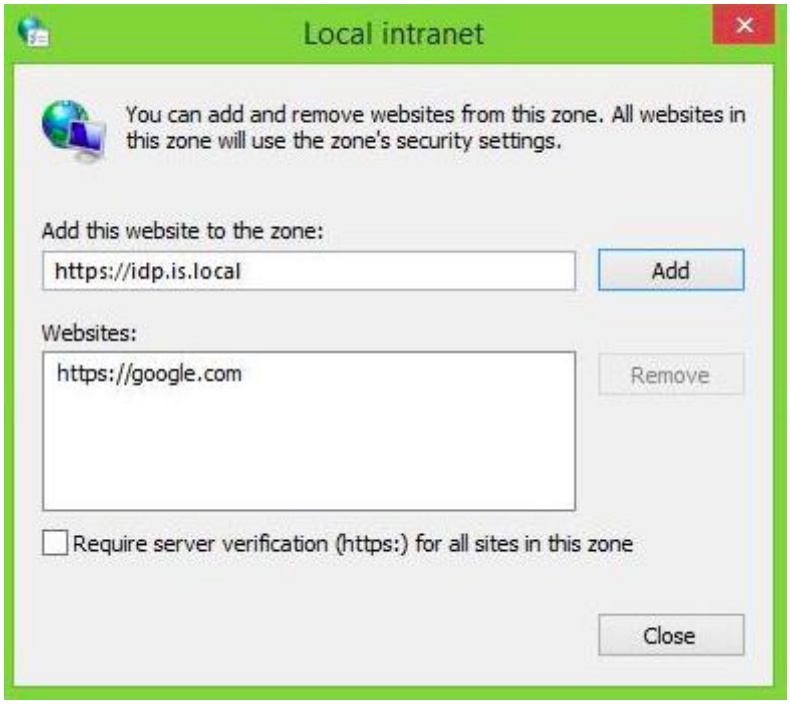
Настройка Internet Explorer/Chrome

Перейдите в Инструменты -> Свойства обозревателя.

На вкладке «Безопасность» выберите локальную интрасеть.



Нажмите кнопку **Сайты** . Затем добавьте туда URL-адрес WSO2 Identity Server.



Chrome просто наследует настройки от Internet Explorer, так что дополнительно ничего настраивать не нужно.

2.1.5. Конфигурация Redis

Программное обеспечение Redis разворачивается в docker контейнере на сервере контейнеризации BP-MODULE в соответствии настройками, приведенными в разделе 2.1.2 (файл образа - redis.tar).

Конфигурация сервиса Redis осуществляется путем проверки и, в случае необходимости, корректировки конфигурационных файлов docker compose (*docker-compose.yml и .env.*) в части следующих параметров:

Параметры, необходимые для корректного запуска сервиса:

- ALLOW_EMPTY_PASSWORD – разрешает использовать пустой пароль, необходимо указать значение NO
- REDIS_PASSWORD – задает пароль для подключения к сервису.

Актуальная конфигурация файлов docker compose представлена в Приложение 1.

2.1.6. Конфигурация GDAL

Используется в составе контейнеров: loader.tar (модуль экспорта и импорта пространственных данных) и way-loader.tar (модуль загрузки дорог для модуля построения маршрутов). Дополнительная конфигурация базового ПО не требуется.

Программное обеспечение GDAL разворачивается в docker контейнере на сервере контейнеризации BP-MODULE в соответствии настройками, приведенными в разделе 2.1.2.

2.1.7. Конфигурация Tomcat

Используется в составе контейнеров:

- wso2is
- geoserver
- api
- loader
- gisapi

Дополнительная конфигурация базового ПО не требуется.

2.1.8. Конфигурация OpenJDK

Используется в составе контейнеров:

- wso2is
- geoserver
- api
- gisapi
- loader

Дополнительная конфигурация базового ПО не требуется.

2.1.9. Конфигурация Nginx для сервера контейнеризации

Для аутентификации в nginx добавлен модуль Lua, который может запускать сценарии Lua с использованием компилятора LuaJIT. Программное обеспечение nginx разворачивается в docker контейнере на сервере контейнеризации BP-MODULE (файл образа - openresty.tar).

Конфигурация сервиса Nginx осуществляется путем проверки и, в случае необходимости, корректировки конфигурационных файлов docker compose (*docker-compose.yml u .env.*).

Параметры конфигурации файла настроек docker compose, в том числе для Nginx, представлено в Приложение 1.

В директории контейнера необходимо убедиться в наличии следующих файлов настроек:

- ./openresty/conf/nginx.conf – основной файл конфигурации nginx
- ./openresty/conf/kpropi.conf – дополнительный файл конфигурации nginx
- ./openresty/certs/ca-bundle.trust.crt – корневые сертификаты, необходимы при валидации JWT токенов.

Содержимое файла nginx.conf:

```
# Enables the use of JIT for regular expressions to speed-up their processing.
pcre_jit on;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    resolver 127.0.0.11;

    # See Move default writable paths to a dedicated directory (#119)
    # https://github.com/openresty/docker-openresty/issues/119
    client_body_temp_path /var/run/openresty/nginx-client-body;
    proxy_temp_path       /var/run/openresty/nginx-proxy;
    fastcgi_temp_path     /var/run/openresty/nginx-fastcgi;
    uwsgi_temp_path       /var/run/openresty/nginx-uwsgi;
    scgi_temp_path        /var/run/openresty/nginx-scgi;
```

```
    sendfile        on;
    #tcp_nopush     on;
    #tcp_nodelay    off;

    keepalive_timeout  65;

    gzip              on;
    gzip_http_version  1.0;
    gzip_comp_level    2;
    gzip_proxied       any;
    gzip_types         text/plain text/css application/javascript text/xml application/xml+rss;

    client_max_body_size 1G;

    lua_package_path "/usr/local/openresty/?.lua;;";
    lua_ssl_trusted_certificate /etc/ssl/certs/ca-bundle.trust.crt;
    lua_ssl_verify_depth 5;
    # cache for discovery metadata documents
    #lua_shared_dict discovery 1m;
    # cache for JWKs
    lua_shared_dict jwks 1m;

    include /etc/nginx/conf.d/*.conf;

    # Don't reveal OpenResty version to clients.
    server_tokens off;
}
```

Содержимое файла bpmodule.conf:

```
server {
    listen 80 default_server;
    server_name _;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl http2;

    proxy_read_timeout 25m;
    proxy_send_timeout 25m;
    proxy_buffer_size 64k;
    proxy_buffers 4 64k;
    proxy_busy_buffers_size 64k;

    client_max_body_size 2G;
    large_client_header_buffers 4 64k;

    ssl_certificate /etc/nginx/ssl/cert.pem;
    ssl_certificate_key /etc/nginx/ssl/key.pem;

    access_log logs/openresty-access.log;
    error_log logs/openresty-error.log debug;
```

```
proxy_redirect off; # new
proxy_set_header Host $host; # new
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-Host $http_host;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $http_scheme;

set $session_secret
    455cb000f8a77322c2991da04a119b872ccd494931e964453558efb827d25364;
set $session_storage redis;
set $session_redis_prefix sessions;
set $session_redis_host redis-service;
set $session_redis_port 6379;
set $session_redis_auth redispassword;
set $session_strategy regenerate;
set $session_cookie_httponly off;
set $session_cookie_samesite off;

set $auth_lua /etc/nginx/lua.d/authenticate.lua;
set $check_token /etc/nginx/lua.d/check_token.lua;

location / {
    rewrite ^ /bp-module/ permanent;
}

location /login {
    access_by_lua_file $auth_lua;
}

location /logout {
    access_by_lua_file $auth_lua;
}

location /bp-module-api/ {
    access_by_lua_file $check_token;
    proxy_pass http://bp-module-api:80;
}

location /bp-module-api/logout {
    access_by_lua_file $check_token;
    proxy_pass http://bp-module-api:80;
}

location /bp-module-api/bp-module-api/oauth2/callback {
    access_by_lua_file $auth_lua;
    proxy_pass http://bp-module-api:80;
}

location /favicon.ico {
    proxy_pass http://bp-module-client:80;
}
```

```
    location /bp-module/ {
        proxy_pass http://bp-module-client:80;
    }

    location /egis-bp-module/ {
        proxy_pass http://egis-bp-module:80;
    }

    location /bp-module-loader/ {
#       access_by_lua_file $auth_lua;
        proxy_pass http://loader:8080;
    }

    location /bp-module-gisapi/ {
        access_by_lua_file $auth_lua;
        proxy_pass http://gisapi:8070;
    }

    location /geoserver/ {
#       access_by_lua_file $auth_lua;
        proxy_pass http://geoserver:8080;
    }

    location ~* ^/(carbon|console|myaccount) {
        proxy_pass https://wso2is:9443;
        allow 192.168.0.0/23;
        deny all;
    }

    location /t/ {
        proxy_pass https://wso2is:9443;
    }

    location ~*
^/(authenticationendpoint|logincontext|commonauth|accountrecoveryendpoint|oauth2|api|oidc|scim|iwa-
kerberos|samlsso|acs|samlartresolve) {
        proxy_pass https://wso2is:9443;
    }
}
```

2.1.10. Конфигурация специализированного ППО BP-MODULE

Специализированное ППО BP-MODULE состоит из нескольких сервисов передаваемых в форме образов контейнеров:

- client – клиентская часть приложения (файл образа – client.tar)
- gis-client – клиентская часть приложения для отображения картографической информации (файл образа – gis-client.tar)
- api – серверная часть приложения (файл образа – api.tar)
- loader – сервис экспорта и импорта пространственных данных (файл образа – loader.tar)
- gisapi – серверная часть приложения картографии (файл образа – gisapi.tar)

Специализированное ППО BP-MODULE разворачивается в docker контейнере на сервере контейнеризации BP-MODULE в соответствии настройками, приведенными в разделе 2.1.2.

Образы сервисов client и gis-client не имеют настраиваемых конфигурационных параметров.

Конфигурация образов сервисов ППО BP-MODULE осуществляется путем проверки и, в случае необходимости, корректировки конфигурационных файлов docker compose (*docker-compose.yml и .env.*) в части следующих параметров:

Параметры, необходимые для запуска образа сервиса api:

- DATASOURCE_HOST – имя сервера БД
- DATASOURCE_PORT – порт БД
- DATASOURCE_DB – имя БД
- DATASOURCE_SCHEMA – схема в БД
- DATASOURCE_USER – имя пользователя БД
- DATASOURCE_PASSWORD – пароль пользователя БД
- WSO2_URL – адрес сервера идентификации WSO2
- WSO2_CLIENT_ID – идентификатор OAuth клиента
- WSO2_CLIENT_SECRET – пароль OAuth клиента
- LDAP_URL – адрес подключения к серверу ldap
- LDAP_BASE – базовый адрес поиска в ldap
- LDAP_USERNAME – пользователь для подключения к ldap
- LDAP_PASSWORD – пароль пользователя для подключения к ldap

Параметры, необходимые для запуска образа сервиса loader:

- DATASOURCE_HOST – имя сервера БД
- DATASOURCE_PORT – порт БД
- DATASOURCE_DB – имя БД
- DATASOURCE_SCHEMA – схема пространственных данных в БД
- DATASOURCE_USER – имя пользователя БД
- DATASOURCE_PASSWORD – пароль пользователя БД

Параметры, необходимые для запуска образа сервиса gisapi:

- DATASOURCE_HOST – имя сервера БД
- DATASOURCE_PORT – порт БД
- DATASOURCE_DB – имя БД
- DATASOURCE_SCHEMA – схема пространственных данных в БД
- DATASOURCE_USER – имя пользователя БД
- DATASOURCE_PASSWORD – пароль пользователя БД
- GEOSERVER_URL – адрес подключения к геосерверу
- GEOSERVER_USER – пользователь для подключения к геосерверу

Содержание конфигурационных файлов docker compose представлено в Приложение 1.

2.1.11. Конфигурация и запуск Nginx для web-сервера

Прикладное программное обеспечение BP-MODULE использует два экземпляра Nginx. Один используется как прокси сервер для доступа ко всем сервисам. Второй используется для аутентификации пользователей при обращении к сервисам специализированного программного обеспечения.

Программное обеспечение nginx устанавливается на Web-сервере BP-MODULE.

Для установки nginx выполните следующие команды:

dnf install nginx-1.18.0-1.el7 nginx-mimetypes-2.1.48-4.el7 nginx-mod-http-xslt-filter-1.18.0-1.el7 gperftools-libs-2.6.1-7.el7 nginx-all-modules-1.18.0-1.el7 nginx-mod-http-image-filter-1.18.0-1.el7 nginx-mod-mail-1.18.0-1.el7 nginx-filesystem-1.18.0-1.el7 nginx-mod-http-perl-1.18.0-1.el7 nginx-mod-stream-1.18.0-1.el7

systemctl enable nginx

systemctl start nginx

Настойка nginx осуществляется с помощью двух конфигурационных файлов:

- /etc/nginx/nginx.conf – основной файл конфигурации.
- /etc/nginx/conf.d/bpmodule.technocom.tech.conf – дополнительный файл конфигурации.

Содержимое файла nginx.conf:

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;
```

```
    include /etc/nginx/conf.d/*.conf;

    server {
        listen       80;
        listen       [::]:80;
        server_name  _;
        root         /usr/share/nginx/html;l

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        error_page 404 /404.html;
            location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
        }
    }

}
```

Содержимое файла bpmodule.technocom.tech.conf:

```
upstream mincifry-uslugi-openresty-bk {
    server 192.168.55.139:443;
    keepalive 100;
}


server {
    listen 80;
    server_name bpmodule.technocom.tech;
    return 301
        https://$server_name$request_uri; # редирект обычных запросов на https
}

server {
    listen 443 ssl;
    http2 on;
    server_name bpmodule.technocom.tech;
    proxy_read_timeout 25m;
    proxy_send_timeout 25m;
    proxy_buffer_size 64k;
    proxy_buffers 4 64k;
    proxy_busy_buffers_size 64k;

    client_max_body_size 2G;
    large_client_header_buffers 4 64k;

    ssl_certificate /etc/nginx/ssl/certificate.crt;
    ssl_certificate_key /etc/nginx/ssl/private.key;
```

```
access_log /var/log/nginx/bpmodule.technocom.tech-ssl-access.log;
error_log /var/log/nginx/bpmodule.technocom.tech-ssl-error.log;

proxy_redirect ~^https?:\/\/((.+\.)*\w+)(:\d+)?(\/.*)$ https://$1$4;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header Host $http_host;
proxy_set_header X-Forwarded-Host $host;
proxy_set_header X-Forwarded-For $remote_addr;
proxy_set_header X-Forwarded-Proto $scheme;

location / {
    # rewrite ^ /bp-module/ permanent;
    proxy_pass https://mincifry-uslugi-openresty-bk;
}
```

Добавьте сертификат и приватный ключ для домена приложения в каталог /etc/nginx/ssl:

/etc/nginx/ssl/certificate.crt

/etc/nginx/ssl/private.key.

ПРИЛОЖЕНИЕ 1

Содержание файла docker-compose.yml:

```
version: '3.7'
services:

  redis:
    image: ${IMAGE_REDIS}
    container_name: redis
    environment:
      - ALLOW_EMPTY_PASSWORD=no
      - REDIS_PASSWORD=${REDIS_PASSWORD}
      - TZ=${TIME_ZONE}
    expose:
      - '6379'
    volumes:
      - redis-data:/bitnami/redis/data
    networks:
      bpmodule-network:
        aliases:
          - redis-service
    restart: unless-stopped

  openresty:
    image: ${IMAGE_OPENRESTY}
    container_name: openresty
    ports:
      - '80:80'
      - '443:443'
    volumes:
      - ./openresty/certs/ca-bundle.trust.crt:/etc/ssl/certs/ca-bundle.trust.crt
      - ./openresty/conf/nginx.conf:/usr/local/openresty/nginx/conf/nginx.conf:ro
      - ./openresty/conf/kpropi.conf:/etc/nginx/conf.d/kpropi.conf:ro
      - ./openresty/logs:/usr/local/openresty/nginx/logs
    environment:
      - TZ=${TIME_ZONE}
    networks:
      bpmodule-network:
    depends_on:
      - redis
    restart: unless-stopped

  client:
    container_name: client
    image: ${IMAGE_CLIENT}
    expose:
      - "80"
    networks:
      bpmodule-network:
        aliases:
          - client-service
    restart: unless-stopped
```

```
gis-client:
  container_name: gis-client
  image: ${IMAGE_GIS_CLIENT}
  expose:
    - "80"
  networks:
    bpmodule-network:
      aliases:
        - gis-client-service
  restart: unless-stopped

api:
  container_name: api
  image: ${IMAGE_API}
  expose:
    - "8080"
  volumes:
    - /tmp/bpmodule:/mnt/tmp
    - ./mnt/fs:/mnt/fs
  user : "${UID}:${GID}"
  environment:
    - DATASOURCE_HOST=${DATASOURCE_HOST}
    - DATASOURCE_PORT=${DATASOURCE_PORT}
    - DATASOURCE_DB=${DATASOURCE_DB}
    - DATASOURCE_SCHEMA=${DATASOURCE_SCHEMA}
    - DATASOURCE_USER=${DATASOURCE_USER}
    - DATASOURCE_PASSWORD=${DATASOURCE_PASSWORD}
    - WSO2_URL=${SECURITY_IS_URL}
    - WSO2_CLIENT_ID=${SECURITY_ID}
    - WSO2_CLIENT_SECRET=${SECURITY_SECRET}
    - LDAP_URL=${LDAP_URL}
    - LDAP_BASE=${LDAP_BASE}
    - LDAP_USERNAME=${LDAP_USERNAME}
    - LDAP_PASSWORD=${LDAP_PASSWORD}
    - LDAP_USER_DN=${LDAP_USER_DN}
    - LDAP_GROUP_DN=${LDAP_GROUP_DN}
    - TZ=${TIME_ZONE}
  networks:
    bpmodule-network:
      aliases:
      - api-service
  restart: unless-stopped

loader:
  container_name: loader
  image: ${IMAGE_LOADER}
  expose:
    - "8080"
  volumes:
    - ./mnt/loader:/mnt
  user : "${UID}:${GID}"
  environment:
```

```
    - DATASOURCE_HOST=${DATASOURCE_HOST}
    - DATASOURCE_PORT=${DATASOURCE_PORT}
    - DATASOURCE_DB=${DATASOURCE_DB}
    - DATASOURCE_SCHEMA=${DATASOURCE_SCHEMA}
    - DATASOURCE_USER=${DATASOURCE_USER}
    - DATASOURCE_PASSWORD=${DATASOURCE_PASSWORD}
    - LOAD_PATH=${LOAD_PATH}
  networks:
    bpmodule-network:
      aliases:
        - loader-service
  restart: unless-stopped

 gisapi:
  container_name: gisapi
  image: ${IMAGE_GISAPI}
  expose:
    - "8098"
  user : "${UID}:${GID}"
  environment:
    - DATASOURCE_HOST=${DATASOURCE_HOST}
    - DATASOURCE_PORT=${DATASOURCE_PORT}
    - DATASOURCE_DB=${DATASOURCE_DB}
    - DATASOURCE_SCHEMA=${DATASOURCE_SCHEMA}
    - DATASOURCE_USER=${DATASOURCE_USER}
    - DATASOURCE_PASSWORD=${DATASOURCE_PASSWORD}
    - GEOSERVER_URL=${GEOSERVER_URL}
    - TZ=${TIME_ZONE}
  networks:
    bpmodule-network:
      aliases:
        - gisapi-service
  restart: unless-stopped

volumes:
  redis-data:
networks:
 gisapi-network:
   external: true
```

Содержимое файла .env:

```
## Версии docker образов модулей
IMAGE_REDIS=redis:6.0
IMAGE_OPENRESTY=tkrepo.technocom.tech/technocom/openresty:latest
IMAGE_CLIENT= tkrepo.technocom.tech/bp-module:latest
IMAGE_API= tkrepo.technocom.tech/tis/bp-module-api:latest
IMAGE_GIS_CLIENT= tkrepo.technocom.tech/gis/bp-module:latest
IMAGE_LOADER= tkrepo.technocom.tech/gis/eservice-loader:latest
IMAGE_GISAPI= tkrepo.technocom.tech/gis/eservice-gisapi:latest

## Параметры подключения к БД
DATASOURCE_HOST=example.local
DATASOURCE_PORT=10265
```

```
DATASOURCE_DB=bpmodule
DATASOURCE_SCHEMA=bpmodule
DATASOURCE_USER=bpmodule
DATASOURCE_PASSWORD=ПАРОЛЬ

## Параметры сервера идентификации WSO2
#Адрес сервера идентификации
SECURITY_IS_URL=https://bpmodule.technocom.tech
#OAuth Client Id
SECURITY_ID= ПРОПИСАТЬ ID OAUTH КЛИНЕТА СЕРВИС ПРОВАЙДЕРА WSO2
#OAuth Client Secret
SECURITY_SECRET= ПРОПИСАТЬ SECRET OAUTH КЛИНЕТА СЕРВИС ПРОВАЙДЕРА WSO

## Параметры подключения к LDAP каталогу Active Directory
#Адрес подключения к серверу ladp
LDAP_URL=ldap://dc1.example.local:389
#Каталог начиная с которого производится поиск
LDAP_BASE=dc=example,dc=local
#Пользователь
LDAP_USERNAME= cn=bpmodule,ou=groups,ou=Service
Accounts,ou=Datacenter,dc=example,dc=local
#Пароль
LDAP_PASSWORD= ПРОПИСАТЬ ПАРОЛЬ СЕРВИСНОЙ УЧЕТНОЙ ЗАПИСИ
#Имя раздела в котором расположены пользователи
LDAP_USER_DN=
#Имя раздела в котором расположены группы
LDAP_GROUP_DN=

REDIS_PASSWORD= ПРОПИСАТЬ ПАРОЛЬ REDIS

#Параметры для gisapi
GEOSERVER_URL=http://geoserver:8080/geoserver/
DATASOURCE_HOST: www.example.local
DATASOURCE_PORT: 5432
DATASOURCE_DB: bpmodule
DATASOURCE_SCHEMA: eservice_gis
DATASOURCE_USER: eservice_gis
DATASOURCE_PASSWORD:
#Id пользователя под которым запускается процесс в контейнере
UID=1000
#Id группа пользователя под которым запускается процесс в контейнере
GID=988

TIME_ZONE=Asia/Yekaterinburg
```