

**РУКОВОДСТВО АДМИНИСТРАТОРА
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «NOVAGIS»**

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	3
2. УСТАНОВКА И НАСТРОЙКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	3
2.1. Развертывание и конфигурация PostgreSQL	3
2.2. Конфигурация платформы контейнеризации Docker CE.....	6
3. КОНФИГУРАЦИЯ И ЗАПУСК КОНТЕЙНЕРОВ NOVAGIS И ИХ ЗАВИСИМОСТЕЙ	7
4. КОНФИГУРАЦИЯ И ЗАПУСК NGINX ДЛЯ WEB-СЕРВЕРА	8

1. ВВЕДЕНИЕ

Данный документ предназначен для сотрудника эксплуатирующей организации, который выполняет функции по установке и настройке программного обеспечения «NOVAGIS».

2. УСТАНОВКА И НАСТРОЙКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1. Развертывание и конфигурация и PostgreSQL

В прикладном программном обеспечении NOVAGIS postgresql используется для двух контейнеров:

контейнер geoserver;

контейнер api.

Для установки необходимых пакетов PostgreSQL необходимо выполнить команды:

```
sudo apt install postgresql-all
```

```
sudo systemctl enable postgresql
```

```
sudo systemctl start 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 NOVAGIS должно быть развернуто два экземпляра БД. Один используется для обеспечения работы контейнера аутентификации, другой – для обеспечения работы остальных приложений.

Пример конфигурационного файла postgresql.conf:

```
ac_enable_maclabels_on_files = false
ac_ignore_maclabel          = false
ac_ignore_socket_maclabel   = true
ac_ignore_server_maclabel   = true
ac_enable_trusted_owner     = true
ac_enable_truncate          = true
ac_enable_sequence_ccr      = true
ac_enable_copy_to_file      = true
ac_auto_adjust_macs         = false
ac_enable_dblink_mac        = false
ac_enable_grant_options     = false
ac_enable_admin_options     = false
```

```
ac_caps_ttl          = 10          # rbt capabilities Time to live, in seconds
ac_debug_print       = false
ac_audit_mode        = 'internal'   # rbt audit mode. Possible values are
ac_audit_destination = 'all'        # rbt Audit destination. Possible values are
ac_audit_log_only_failures = false
ac_log_stmt_length   = 100         # rbt audit stmt length, 0 - unlimited
ac_wipe_buffers      = false
ac_wipe_pages        = false
ac_wipe_files        = false
data_directory = '/var/lib/postgresql/11/main'      # use data in another directory
hba_file = '/etc/postgresql/11/main/pg_hba.conf'    # host-based authentication
file
ident_file = '/etc/postgresql/11/main/pg_ident.conf' # ident configuration file
external_pid_file = '/var/run/postgresql/11-main.pid' # write an extra PID
file
listen_addresses = '*'      # what IP address(es) to listen on;
port = 5432                # (change requires restart)
max_connections = 500       # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of
directories
ssl = on
ssl_cert_file = '/etc/ssl/certs/ssl-cert-snakeoil.pem'
ssl_key_file = '/etc/ssl/private/ssl-cert-snakeoil.key'
shared_buffers = 128MB      # min 128kB
temp_buffers = 256MB        # min 800kB
work_mem = 64MB             # min 64kB
dynamic_shared_memory_type = posix # the default is the first option
max_files_per_process = 8000 # min 25
max_wal_size = 1GB
min_wal_size = 80MB
from_collapse_limit = 20
join_collapse_limit = 20    # 1 disables collapsing of explicit
log_destination = 'stderr'  # Valid values are combinations of
logging_collector = on      # Enable capturing of stderr and csvlog
log_directory = 'pg_log'    # directory where log files are written,
log_filename = 'postgresql-%a.log' # log file name pattern,
log_truncate_on_rotation = on # If on, an existing log file with the
```

```

log_rotation_age = 1d          # Automatic rotation of logfiles will
log_rotation_size = 0          # Automatic rotation of logfiles will
log_line_prefix = '%m [%p] %q%u@%d '          # special values:
log_timezone = 'Asia/Yekaterinburg'
cluster_name = '11/main'       # added to process titles if nonempty
stats_temp_directory = '/var/run/postgresql/11-main.pg_stat_tmp'
timezone = 'Asia/Yekaterinburg'
lc_messages = 'ru_RU.UTF-8'    # locale for system error message
lc_monetary = 'ru_RU.UTF-8'    # locale for monetary formatting
lc_numeric = 'ru_RU.UTF-8'     # locale for number formatting
lc_time = 'ru_RU.UTF-8'        # locale for time formatting
default_text_search_config = 'pg_catalog.russian'
shared_preload_libraries = 'online_analyze, plantuner' # (change requires restart)
max_locks_per_transaction = 150 # min 10
escape_string_warning = off
standard_conforming_strings = off
include_dir = 'conf.d'         # include files ending in '.conf' from
online_analyze.threshold = 50
online_analyze.scale_factor = 0.1
online_analyze.enable = off
online_analyze.verbose = off
online_analyze.local_tracking = on
online_analyze.min_interval = 10000
online_analyze.table_type = 'temporary'
plantuner.fix_empty_table = on

```

В конфигурационном файле `pg_hba.conf` необходимо разрешить доступ модулям NOVAGIS.

Пример конфигурационного файла `pg_hba.conf`:

local	all	postgres		peer
local	all	all		peer
host	all	all	0.0.0.0/0	md5
host	all	all	:::1/128	md5
local	replication	all		peer
host	replication	all	127.0.0.1/32	md5
host	replication	all	:::1/128	md5

После запуска сервиса БД необходимо выполнить процедуру создания базы данных и создать пользователей (роли) для подключения к базе данных.

Создание баз данных осуществляется следующей SQL командой:

```
create database novagis
```

Создание пользователя (роли) для NOVAGIS и назначение прав осуществляется следующей SQL командой:

```
CREATE ROLE novagis WITH  
    NOSUPERUSER  
    NOCREATEDB  
    NOCREATEROLE  
    INHERIT  
    LOGIN  
    NOREPLICATION  
    NOBYPASSRLS  
    CONNECTION LIMIT -1  
    PASSWORD 'пароль';  
CREATE SCHEMA novagis AUTHORIZATION novagis;  
GRANT ALL ON SCHEMA novagis TO novagis;
```

После выполнения создания роли NOVAGIS необходимо выполнить скрипты первоначального наполнения БД. Скрипт novagis.sql передается в составе инсталляционных файлов.

Для выполнения скрипта необходимо выполнить команду:

```
psql -U novagis -f novagis.sql -d novagis -h localhost
```

2.2. Конфигурация платформы контейнеризации Docker CE

Платформа контейнеризации приложений Docker CE используется для упаковки сервисов ППО NOVAGIS 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 NOVAGIS поставляются в виде готовых образов, упакованных в tar архив.

Для базовой установки необходимых пакетов Docker нужно выполнить следующие команды:

```
sudo apt install docker docker-compose  
sudo systemctl enable docker  
sudo systemctl start docker
```

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

на сервер в каталог /opt/images/. Краткое описание образов приведено ниже в таблице (**Ошибка! Источник ссылки не найден.1**).

Имя файла образа	Описание
auth.tar	docker образ сервера аутентификации
redis.tar	docker образ сервиса redis
gis-client.tar	docker образ модуля клиентской части приложения
api.tar	docker образ модуля серверной части приложения
mapfish.tar	docker образ модуля сервиса вывода пространственных данных на печать
loader.tar	docker образ модуля экспорта и импорта пространственных данных
geoserver.tar	docker образ сервера пространственных данных
routing.tar	docker образ модуля построения маршрутов

Делается это с помощью команд:

```
docker load -i имя_файла.tar
```

3. КОНФИГУРАЦИЯ И ЗАПУСК КОНТЕЙНЕРОВ NOVAGIS И ИХ ЗАВИСИМОСТЕЙ

Скопировать и распаковать архив novagis.zip в /opt/ и перейти в директорию /opt/novagis

Скорректировать параметры в файле docker-compose.yaml

DATASOURCE_HOST – имя сервера БД;

DATASOURCE_PORT – порт БД;

DATASOURCE_DB – имя БД;

DATASOURCE_SCHEMA – схема NOVAGIS в БД;

DATASOURCE_USER – имя пользователя БД;

DATASOURCE_PASSWORD – пароль пользователя БД;

MAIL_HOST – имя хоста исходящего почтового сервера;

MAIL_PORT – используемый порт исходящих подключений;

MAIL_PASS – пароль от учетной записи;

MAIL_USER – учетная запись исходящих подключений;

MAIL_SMTP_FROM – почта для исходящих подключений;

REDIS_HOST – имя сервера Redis;

REDIS_PORT – используемый Redis порт;

REDIS_PASSWORD – пароль для доступа к Redis;

KEYCLOAK_URL – URL сервера идентификации keycloak;

BASED_DOMAIN – внешнее имя NOVAGIS

KEYCLOAK_CLIENT_ID – публичный идентификатор OAuth клиента;
KEYCLOAK_CLIENT_SECRET – приватный токен OAuth клиента;
ROAD_DATASOURCE_HOST – имя сервера БД построителя;
ROAD_DATASOURCE_PORT – порт БД построителя;
ROAD_DATASOURCE_SCHEMA – схема построителя в БД;
ROAD_DATASOURCE_DB – имя БД построителя;
ROAD_DATASOURCE_USER – имя пользователя БД построителя;
ROAD_DATASOURCE_PASSWORD – пароль пользователя БД построителя.

Запускаем контейнеры:

```
Docker-compose up -d
```

4. КОНФИГУРАЦИЯ И ЗАПУСК NGINX ДЛЯ WEB-СЕРВЕРА

Прикладное программное обеспечение NOVAGIS 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.

Для установки nginx выполните следующие команды:

```
sudo apt install nginx
```

```
sudo systemctl enable nginx
```

Настойка nginx осуществляется с помощью двух конфигурационных файлов:

- nginx.conf – основной файл конфигурации;
- novagis.conf – дополнительный файл конфигурации.

Скопировать файл nginx.conf в /etc/nginx/nginx.conf

Скопировать файл novagis.conf /etc/nginx/conf/novagis.conf

Запустить nginx:

```
sudo systemctl start nginx
```