

Инструкция по развертыванию ПО Системный Геном

Данная инструкция подготовлена для развертывания ПО Системный Геном на выделенных серверах с доступом к системе через балансировщик. В зависимости от нагрузки возможно горизонтальное увеличение компонент системы, создание кластера СУБД и RabbitMQ (создание кластера в данной инструкции не описывается).

Системные требования

Вычислительная платформа комплекса должна удовлетворять следующим аппаратным требованиям для установки ПО:

- Для серверной части
 - процессор (CPU) с 4 и более ядрами, с частотой не ниже 1,5 ГГц;
 - оперативная память объемом не менее 16 Гб;
 - SSD диск объемом не менее 256 Гб.
- Для клиентской части
 - компьютер соответствующий рекомендованным требованиям Яндекс браузера

Установка СУБД и создание базы данных

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Произвести установку postgresql последней версии путем выполнения следующего набора команд:
`sudo apt install postgresql postgresql-contrib`
4. Настроить локализацию системы
`sudo locale-gen en_US.UTF-8`
5. Заменить в файле /etc/postgresql/14/main/postgresql.conf строку
Раскомментировать ее если требуется
`#listen_addresses = 'localhost'`
на строку
`listen_addresses = '*'`
6. Добавить в конец файла /etc/postgresql/14/main/pg_hba.conf строки

host	all	all	0.0.0.0/0	md5
host	all	all	:::0	md5
7. Заменить в файле /etc/postgresql/14/main/pg_hba.conf строку
`local all all peer`
на строку
`local all all md5`
8. Перезагрузить сервис postgresql
`sudo service postgresql restart`
9. Перейти в системного пользователя postgres командой
`sudo -i -u postgres`
10. Запустить оболочку psql командой
`psql`
11. Создать заготовку БД ПО Системный Геном
`CREATE DATABASE genome WITH TEMPLATE = template0 ENCODING = 'UTF8'
LC_COLLATE = 'en_US.UTF-8';`

12. Перейти в БД genome
`\connect genome`
13. Создать пользователя для подключения к БД
`CREATE ROLE genome login;`
`\password genome`
14. Назначить владельца БД
`ALTER DATABASE genome OWNER TO genome;`
15. Выйти из оболочки psql командой `exit`
16. Выйти из сессии пользователя postgresql командой `exit`
17. Передать на сервер файл дампа БД ПО Системный Геном `genome.20231217.dmp` из дистрибутива (методом на ваш выбор SCP/FTP и пр)
18. Выставить права доступа на файл дампа БД ПО Системный Геном
`sudo chmod a+x genome.20231217.dmp`
19. Развернуть дампы БД ПО Системный Геном в заготовку БД при помощи команды
`psql -U genome -d genome -f genome.20231217.dmp 1> import.log 2> import.error`
20. Убедиться что в созданном файле `import.error` отсутствуют ошибки

Установка брокера сообщений RabbitMQ

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Установить RabbitMQ
 - a. Выполнить установку подготовительных пакетов
`sudo apt-get install curl gnupg apt-transport-https -y`
 - b. Добавить ключи подписи репозитория RabbitMQ
`curl -1sLf`
`"https://keys.openpgp.org/vks/v1/by-fingerprint/0A9AF2115F4687BD29803A206B73A36E6026DFCA" | sudo gpg --dearmor | sudo tee`
`/usr/share/keyrings/com.rabbitmq.team.gpg > /dev/null`
`curl -1sLf`
`"https://keyserver.ubuntu.com/pks/lookup?op=get&search=0xf77f1eda57ebb1cc" | sudo gpg --dearmor | sudo tee`
`/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg > /dev/null`
`curl -1sLf "https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey" | sudo gpg --dearmor | sudo tee /usr/share/keyrings/io.packagecloud.rabbitmq.gpg > /dev/null`
 - c. Создать файл `/etc/apt/sources.list.d/rabbitmq.list` с следующим содержанием
`deb [signed-by=/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg] http://ppa.launchpad.net/rabbitmq/rabbitmq-erlang/ubuntu jammy main`
`deb-src [signed-by=/usr/share/keyrings/net.launchpad.ppa.rabbitmq.erlang.gpg] http://ppa.launchpad.net/rabbitmq/rabbitmq-erlang/ubuntu jammy main`
`deb [signed-by=/usr/share/keyrings/io.packagecloud.rabbitmq.gpg] https://packagecloud.io/rabbitmq/rabbitmq-server/ubuntu/ jammy main`
`deb-src [signed-by=/usr/share/keyrings/io.packagecloud.rabbitmq.gpg] https://packagecloud.io/rabbitmq/rabbitmq-server/ubuntu/ jammy main`
 - d. Выполнить последовательно команды для непосредственной установки
`sudo apt-get update -y`
`sudo apt-get install -y erlang-base \`
`erlang-asn1 erlang-crypto erlang-eldap erlang-ftp erlang-inets \`
`erlang-mnesia erlang-os-mon erlang-parsetools erlang-public-key \`
`erlang-runtime-tools erlang-snmp erlang-ssl \`
`erlang-syntax-tools erlang-tftp erlang-tools erlang-xmerl`

- ```
sudo apt-get install rabbitmq-server -y --fix-missing
```
- e. Установить плагин веб консоли администрирования
- ```
sudo rabbitmq-plugins enable rabbitmq_management
```
4. Создать пользователя для администрирования rabbitmq. **Важно:** обязательно измените имя пользователя и пароль в production среде

```
sudo rabbitmqctl add_user genome genome
```
 5. Добавить права администратора для созданного пользователя

```
sudo rabbitmqctl set_user_tags genome administrator
```
 6. Добавить vhost

```
sudo rabbitmqctl add_vhost genome
```
 7. Добавить пользователя в vhost

```
sudo rabbitmqctl set_permissions -p genome genome ".*" ".*" ".*"
```
 8. Зайти в панель администрирования RabbitMQ по адресу <http://ip-адрес-сервера:15672>
 9. Ввести имя пользователя и пароль заданные в шаге 5
 10. Перейти на вкладку Exchanges, внизу нажать Add a new exchange
 11. Заполнить поля
Virtual host: genome
Name: genome.backend
Type: direct
 12. Нажать Add exchange. Запись добавится в список выше
 13. Снова заполнить поля
Virtual host: genome
Name: genome.generator
Type: direct
 14. Нажать Add exchange. Запись добавится в список выше
 15. Снова заполнить поля
Virtual host: genome
Name: genome.logs
Type: fanout
 16. Нажать Add exchange. Запись добавится в список выше
 17. Перейти на вкладку Queues, внизу нажать Add a new queue
 18. Для создания очереди сообщений с именем Database заполнить поля:
Virtual host: genome
Type: classic
Name: Database
Durability: Durable
Auto delete: No
 19. Нажать Add queue. Запись добавится в список выше
 20. Повторить шаги 18-20 для создания очередей сообщений с именами GenerationStatus, ProjectDeployment, ProjectGeneration
 21. Перейти на вкладку Exchanges, выбрать в списке имя точки обмена genome.backend
 22. В открывшемся окне деталей точки обмена нажать на Bindings
 23. Заполнить поля:
To queue: GenerationStatus
Routing key: generation_status
 24. Нажать Bind
 25. Перейти на вкладку Exchanges, выбрать в списке имя точки обмена genome.generator
 26. В открывшемся окне деталей точки обмена нажать на Bindings

27. Заполнить поля:
To queue: ProjectDeployment
Routing key: generator_deploy
28. Нажать Bind
29. Нажать на Bindings
30. Заполнить поля:
To queue: ProjectGeneration
Routing key: generator_request
31. Нажать Bind
32. Перейти на вкладку Exchanges, выбрать в списке имя точки обмена genome.logs
33. В открывшемся окне деталей точки обмена нажать на Bindings
34. Заполнить поля:
To queue: Database
Routing key: logs
35. Нажать Bind

Установка и настройка подсистемы безопасности

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Установить docker выполнив инструкцию с сайта компании производителя <https://docs.docker.com/engine/install/ubuntu/>
4. Произвести первоначальную настройку keycloak согласно инструкции с сайта компании производителя <https://www.keycloak.org/getting-started/getting-started-docker>
 - a. В секции Create a realm указать имя нового пространства genome
 - b. В секции Secure the first application указать Client ID клиента genome-ide
 - c. В секции Valid redirect URIs указать *
Важно: обязательно задайте вместо * конечный адрес URL вашего приложения при настройке в production среде

Установка и настройка подсистемы хранения кода (GitLab)

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
- Установить GitLab выполнив инструкцию с сайта компании производителя https://docs.gitlab.com/ee/install/install_methods.html

Frontend

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Установить docker выполнив инструкцию с сайта компании производителя <https://docs.docker.com/engine/install/ubuntu/>
4. Передать на сервер docker образ pzge-fe-image.tar
5. Загрузить на сервер docker образ при помощи команды `sudo docker load < pzge-fe-image.tar`
6. Произвести запуск контейнера при помощи команды `sudo docker run -d --name genome-frontend --restart always -p 80:8080 pzge-fe-image`

Backend

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Установить docker выполнив инструкцию с сайта компании производителя <https://docs.docker.com/engine/install/ubuntu/>
4. Передать на сервер папку дистрибутива backend
5. Создать папку /srv/genome с подпапкой backend
6. Скопировать данные дистрибутива в папку backend
7. Загрузить на сервер docker образ при помощи команды `sudo docker load < pzge-be-image.tar`
8. Модифицировать файл appsettings.json указав строку подключения к СУБД DbConnection, настройки подключения к брокеру сообщений RabbitMQConfigurationOptions
9. Добавить в файл appsettings.json адреса подключения к серверу KeyCloak в секциях JWT и Swagger

```
"JWT": {  
  "Authority": "https://ip-адрес-сервера-авторизации/realms/genome",  
  "Issuer": "https://ip-адрес-сервера-авторизации/realms/genome",  
  "Audience": "genome-ide"  
},  
"Swagger": {  
  "AuthorizationUrl":  
"https://ip-адрес-сервера-авторизации/realms/genome/protocol/openid-connect/auth",  
  "TokenUrl":  
"https://ip-адрес-сервера-авторизации/realms/genome/protocol/openid-connect/token",  
  "OAuthClientId": "genome-ide",  
  "OAuthAppName": "Genome API"  
}
```

10. Произвести запуск контейнера при помощи команды

```
docker run -d --name genome-backend --restart always -p 8080:80 -v  
/srv/genome/backend/appsettings.json:/app/appsettings.json pzge-be-image
```

Установка и настройка подсистемы логирования

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Установить docker выполнив инструкцию с сайта компании производителя <https://docs.docker.com/engine/install/ubuntu/>
4. Передать на сервер папку дистрибутива logger
5. Создать папку /srv/genome с подпапкой logger
6. Скопировать данные дистрибутива в папку logger
7. Загрузить на сервер docker образ при помощи команды `sudo docker load < pzge-logger-image.tar`
8. Модифицировать файл appsettings.json указав настройки подключения к брокеру сообщений RabbitMQConfigurationOptions
9. Добавить в appsettings.json настройки подключения к БД подсистемы логирования в поле ConnectionString раздела SerilogConfiguration
10. Произвести запуск контейнера при помощи команды

```
sudo docker run -d --name genome-logger --restart always -v
/srv/genome/logger/appsettings.json:/app/appsettings.json pzge-logger-image
```

Библиотека шаблонов

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Установить docker выполнив инструкцию с сайта компании производителя <https://docs.docker.com/engine/install/ubuntu/>
4. Передать на сервер папку дистрибутива templates
5. Создать папку /srv/genome с подпапкой templates
6. Скопировать данные дистрибутива в папку templates
7. Загрузить на сервер docker образ при помощи команды

```
sudo docker load < pzge-templates-image.tar
```
8. Модифицировать файл appsettings.json из папки generator дистрибутива указав настройки подключения к брокеру сообщений RabbitMQConfigurationOptions
9. Модифицировать переменные GIT_EMAIL, GIT_NAME, GIT_TOKEN, GIT_ADDRESS для подключения к Подсистеме хранения кода в файлах push_repository.sh и deploy.sh каждого шаблона в папке /srv/genome/templates/templates
10. Произвести запуск контейнера при помощи команды

```
sudo docker run -d --name genome-templates -v
/srv/genome/templates/templates:/templates -v
/srv/genome/templates/appsettings.json:/app/appsettings.json --restart always -p 80:80
pzge-templates-image
```

Генератор

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Установить docker выполнив инструкцию с сайта компании производителя <https://docs.docker.com/engine/install/ubuntu/>
4. Передать на сервер папку дистрибутива generator
5. Создать папку /srv/genome с подпапкой generator
6. Скопировать данные дистрибутива в папку generator
7. Загрузить на сервер docker образ при помощи команды

```
sudo docker load < pzge-generator-image.tar
```
8. Модифицировать файл appsettings.json указав строку подключения к СУБД DbConnection, настройки подключения к брокеру сообщений RabbitMQConfigurationOptions и URL адрес Библиотеки шаблонов в поле Host Секции TemplateLibraryOptions
9. Произвести запуск контейнера при помощи команды

```
sudo docker run -d --name genome-generator -v
/srv/genome/generator/appsettings.json:/app/appsettings.json --restart always
pzge-generator-image
```

Балансировщик

1. Создать виртуальную машину с ОС Ubuntu 22.04
2. Зайти на сервер под привилегированным пользователем
3. Установить nginx командой

```
sudo apt-get install nginx
```

4. Создать самоподписанный сертификат

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt  
sudo openssl dhparam -out /etc/nginx/dhparam.pem 4096
```
5. Создать файл `/etc/nginx/snippets/self-signed.conf` и вставить в него строки

```
ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;  
ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
```
6. Создать файл `sudo nano /etc/nginx/snippets/ssl-params.conf` и вставить в него строки

```
ssl_protocols TLSv1.3;  
ssl_prefer_server_ciphers on;  
ssl_dhparam /etc/nginx/dhparam.pem;  
ssl_ciphers EECDH+AESGCM:EDH+AESGCM;  
ssl_ecdh_curve secp384r1;  
ssl_session_timeout 10m;  
ssl_session_cache shared:SSL:10m;  
ssl_session_tickets off;  
ssl_stapling on;  
ssl_stapling_verify on;  
resolver 8.8.8.8 8.8.4.4 valid=300s;  
resolver_timeout 5s;  
# Disable strict transport security for now. You can uncomment the following  
# line if you understand the implications.  
#add_header Strict-Transport-Security "max-age=63072000; includeSubDomains;  
preload";  
add_header X-Frame-Options DENY;  
add_header X-Content-Type-Options nosniff;  
add_header X-XSS-Protection "1; mode=block";
```
7. Передать на сервер файл конфигурации балансировщика nginx `genome.conf` из дистрибутива
8. Внести изменения в файл `genome.conf` указав адрес размещения frontend, backend, и сервера авторизации в соответствующих полях
9. Скопировать файл конфигурации в папку `/etc/nginx/sites-available`
10. Создать ссылку на конфигурацию командой

```
ln -s /etc/nginx/sites-available/genome.conf /etc/nginx/sites-enabled/genome.conf
```
11. Применить настройки балансировщика командой `sudo nginx -s`