



**ООО «Стек-ИТ»**

ОКПО 33441104  
ИНН 7604258887

ОГРН 1147604004426  
КПП 760401001

---

150002, г.Ярославль, ул. Малая Химическая, 7А  
info@stack-it.ru +7 (4852) 59-45-00  
support@stack-it.ru www.stack-it.ru



**ООО Компания «Стек»**

ОКПО 88731504  
ИНН 7602069740

ОГРН 1087602004885  
КПП 761001001

---

152912, Ярославская обл., г. Рыбинск, ул. Новая, д.21, пом.3.  
info@stack-it.ru +7 (4852) 59-45-00, 23-00-03  
support@stack-it.ru www.stack-it.ru

ПРОГРАММА ДЛЯ ЭВМ

**Прикладной программный комплекс «Стек-PCO»  
(ПК «Стек-PCO»)**

Инструкция по установке экземпляра программного обеспечения

© Общество с ограниченной ответственностью «Стек-ИТ», 2024 год

© Общество с ограниченной ответственностью Компания «Стек», 2024 год

2024 г.

## Оглавление

Введение.....	3
Установка комплекса СТЕК-PCO.....	4
1.Установка Python.....	4
2.Установка СУБД PostgreSQL.....	5
3.Подключение к серверу БД и его настройка.....	13
4.Восстановление базы данных.....	16
5.Установка Docker Desktop.....	21
6.Настройка конфигурационных файлов и каталогов.....	23
6.1.Настройка файла .env.....	24
6.2.Настройка файла docker-compose.yml.....	24
6.3.Каталог workspace.....	25
6.4.Настройка файла client.ini.....	25
6.5.Итоговый каталог с конфигурационными файлами ...	26
Запуск программного комплекса СТЕК-PCO.....	27
Дополнительная информация.....	29

## Введение

Все новые версии комплекса программ СТЕК-PCO устанавливаются с использованием Docker.

Docker позволяет упаковать приложение со всем его окружением и зависимостями в контейнер, который может быть развернут на любой Linux или Windows системе.

Комплекс СТЕК-PCO поделен на несколько контейнеров:

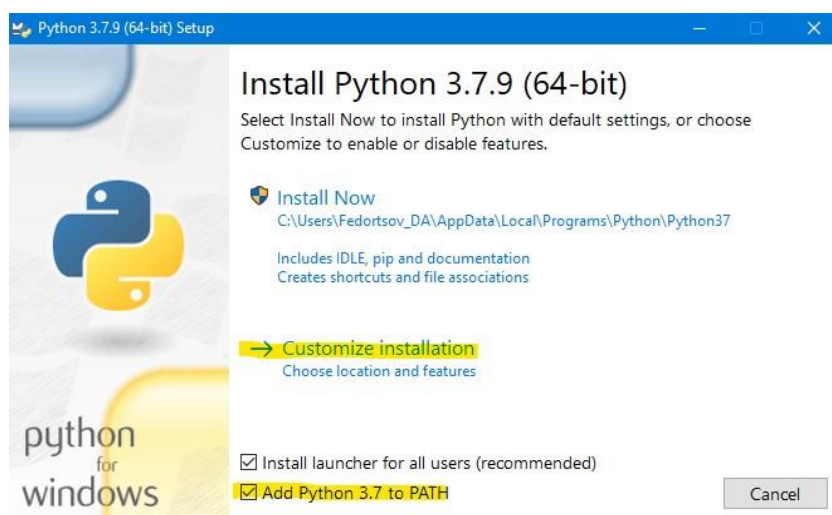
- Nginx - контейнер с WEB-сервером nginx, работающий в роли реверс-прокси;
- RabbitMQ - контейнер с брокером сообщений;
- Gateway - контейнер со StackGateway, который отвечает за аутентификацию и авторизацию пользователей;
- BirtReporter - контейнер со службой построителя отчетов;
- DotNetCore - контейнер с построителем отчетов в MSOffice;
- Frontend - контейнер со статикой, он же stack-front;
- AppHost - набор контейнеров с бинарными файлами и открытой частью бизнес-логики (workspace). Данные контейнеры поднимаются как минимум по одному на каждую из задач (ФЛ, ЮЛ и т.д.) + контейнеры с шедулером (отдельно для каждой из задач) + при необходимости контейнеры с дебагом (так же позадачно).
- Storage - объектное хранилище, сюда попадают файлы с построенными отчетами
- Consul - инструмент для обеспечения сервис-меш сети, предоставляет функциональность для обнаружения, конфигурации и управления сервисами в распределенной среде.

## Установка комплекса СТЕК-PCO

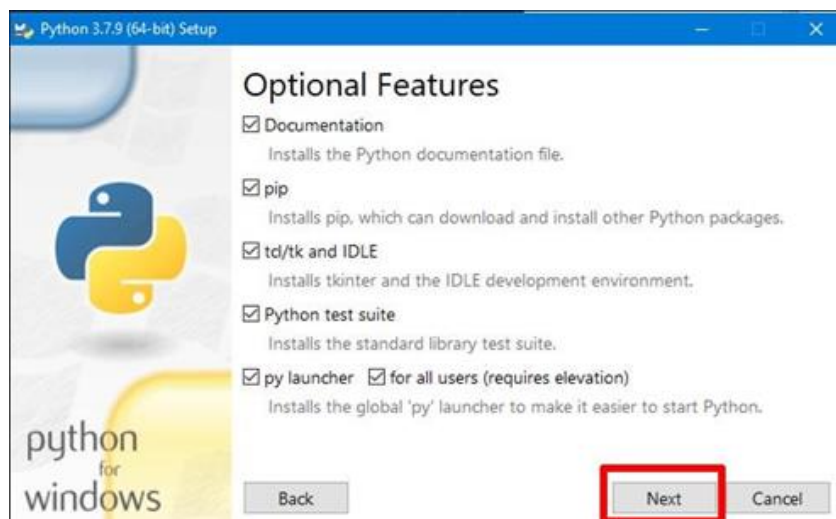
### 1. Установка Python

Необходимо скачать и установить [Python3.7.9](#)(он необходим для работы расширений, которые будут установлены в базу данных при её восстановлении)

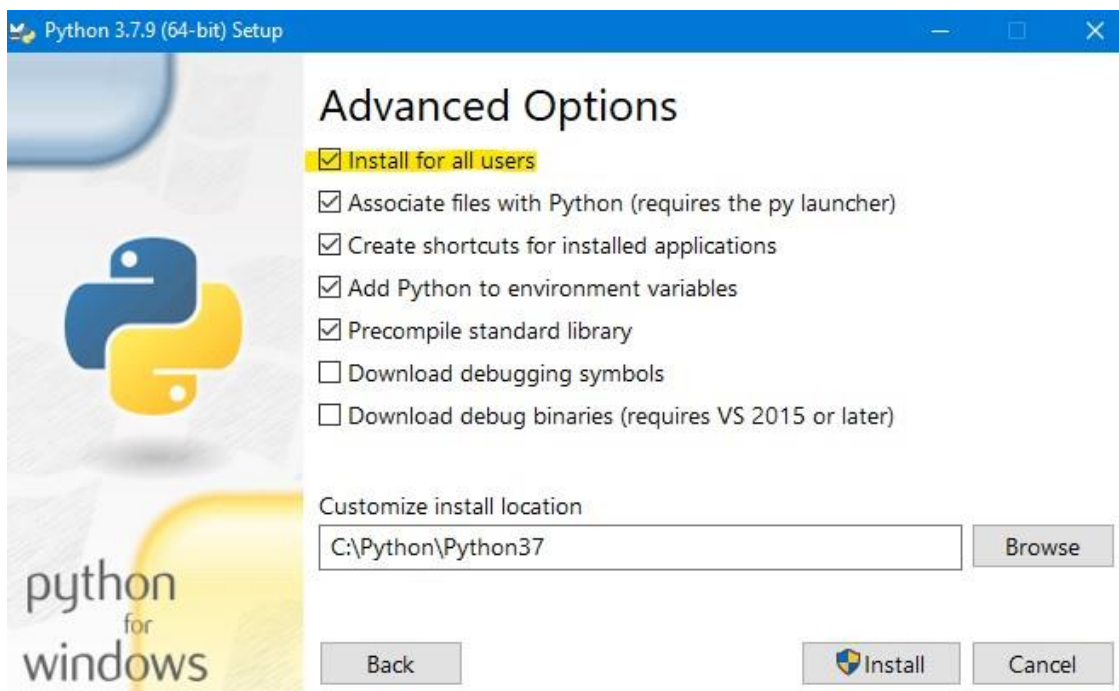
На первом шаге выбираем Add Python to PATH и Customize installation.



Дальше нажимается Next (Далее).



Выбираем отметку Install for all users, каталог установки и нажимаем Install, после окончания установки нажать Close.



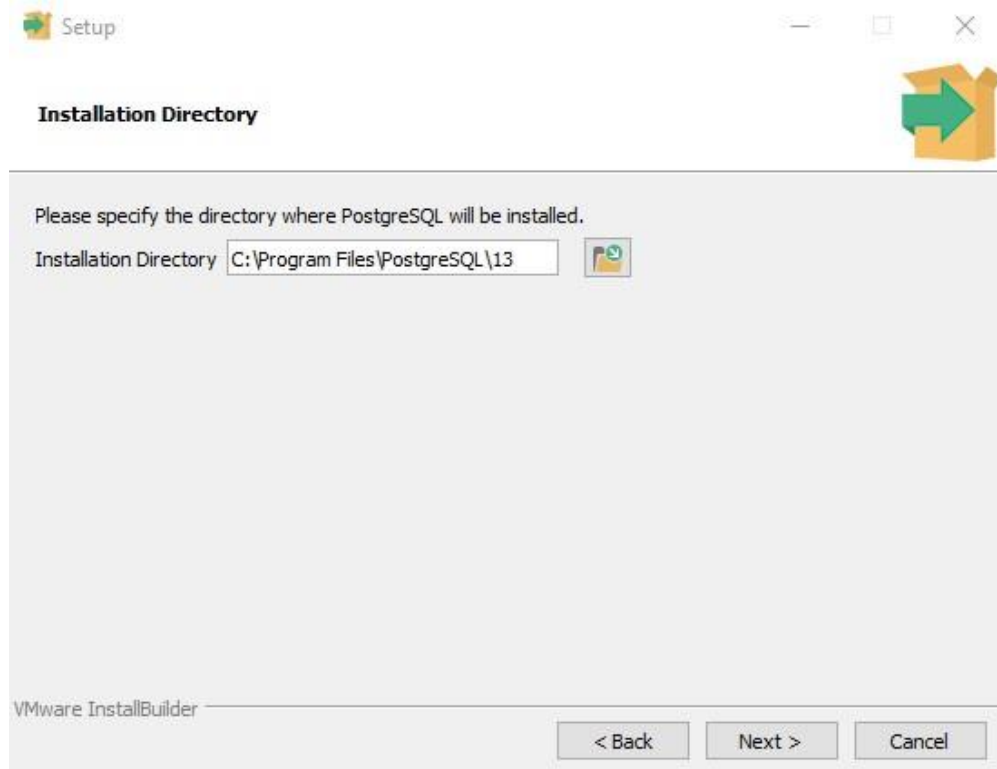
## 2. Установка СУБД PostgreSQL

Скачайте [PostgreSQL15](#) и запустите установщик.

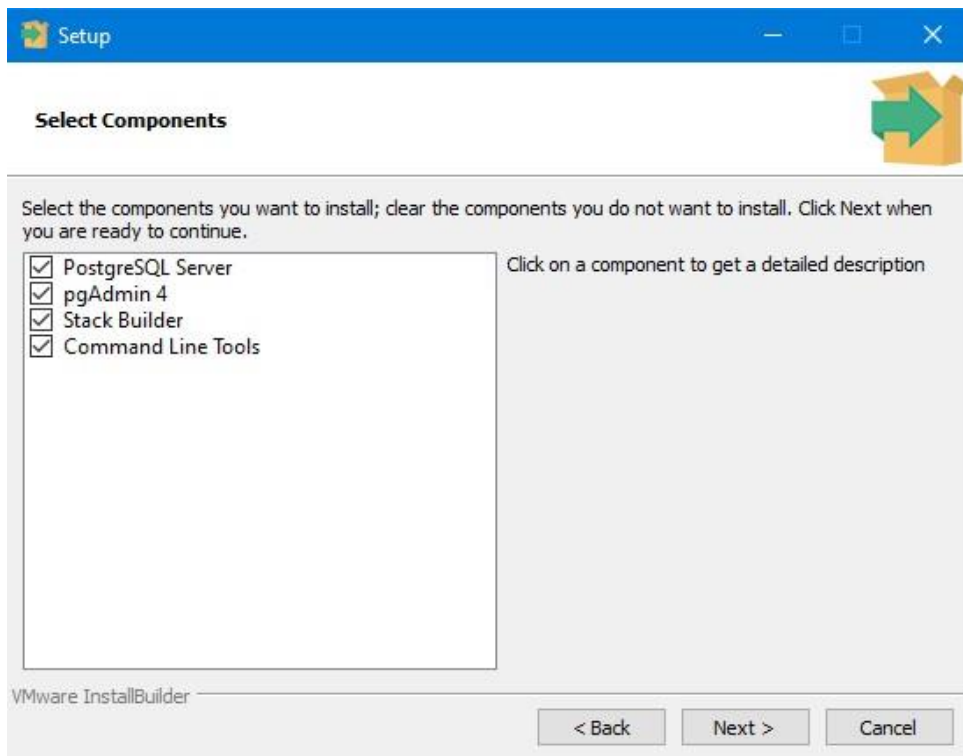
Нажмите Next>



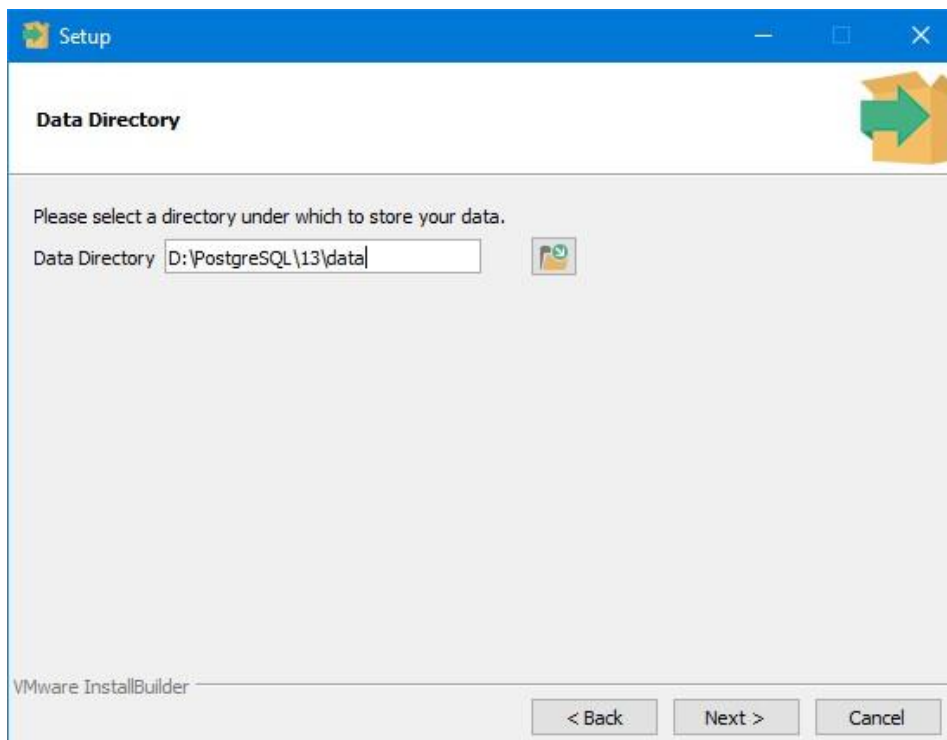
Выбираем директорию установки (рекомендуется оставить по умолчанию) и нажимаем Next>



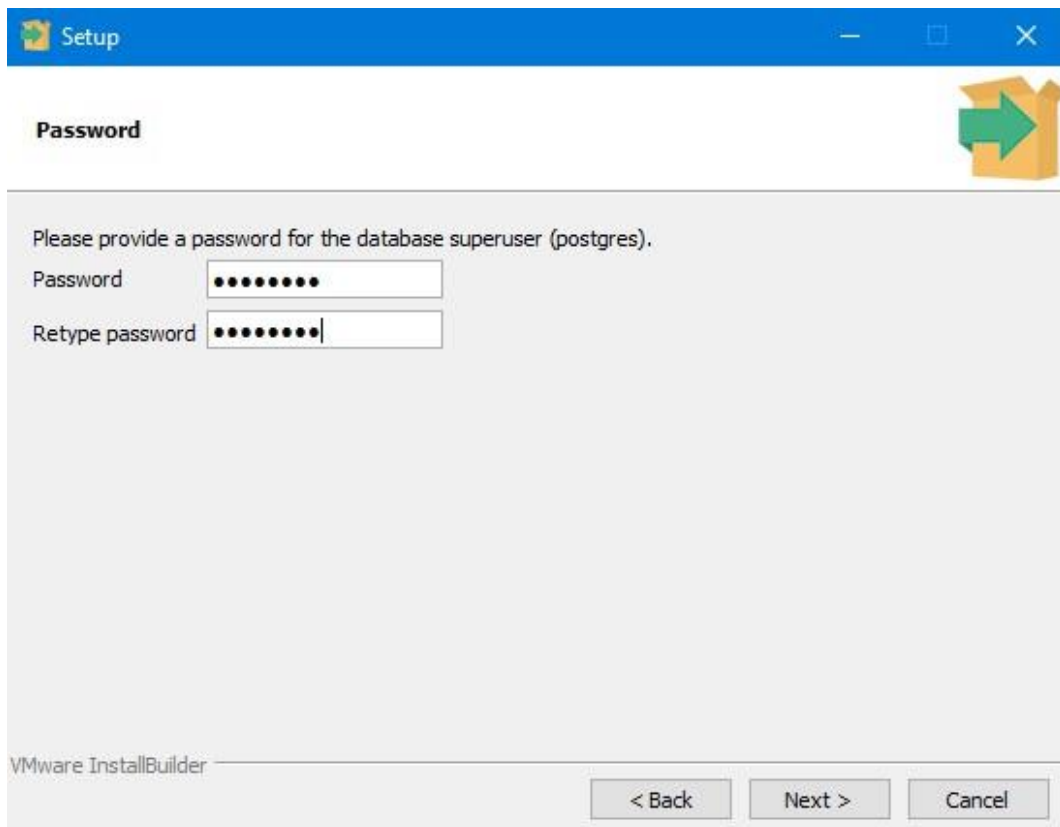
Выбираем компоненты для установки (можно поставить флаги как на изображении ниже) и нажимаем Next>



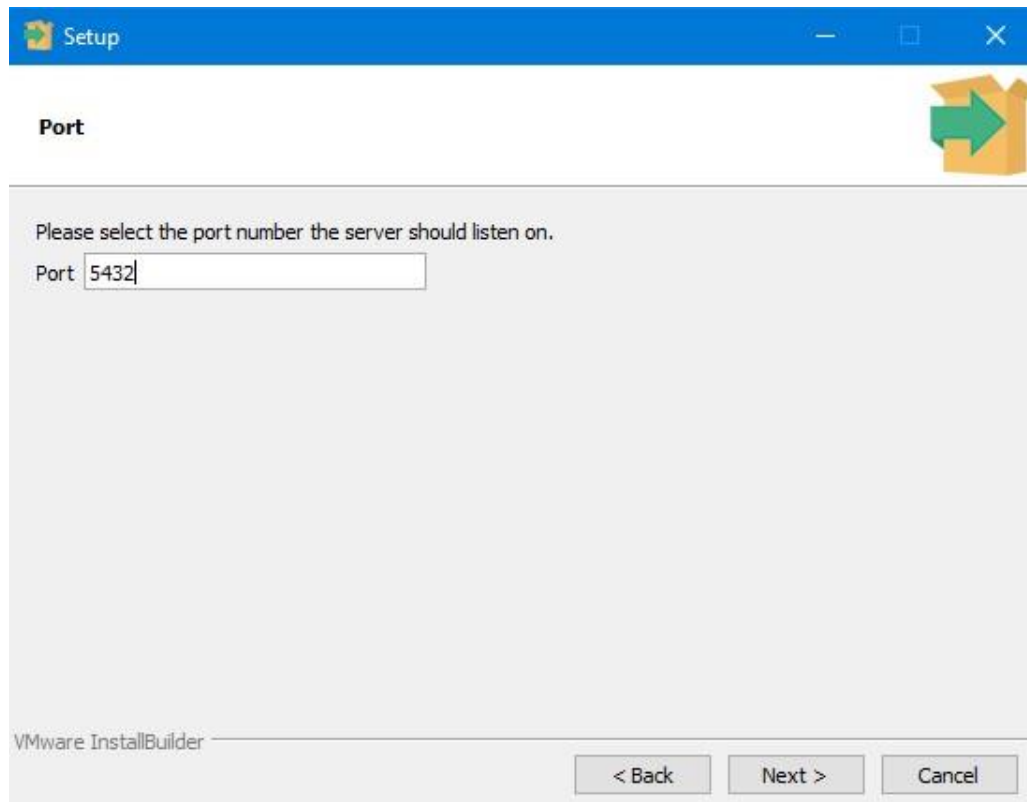
Выбираем директорию для хранения данных (рекомендуется оставить по умолчанию) и нажимаем Next>



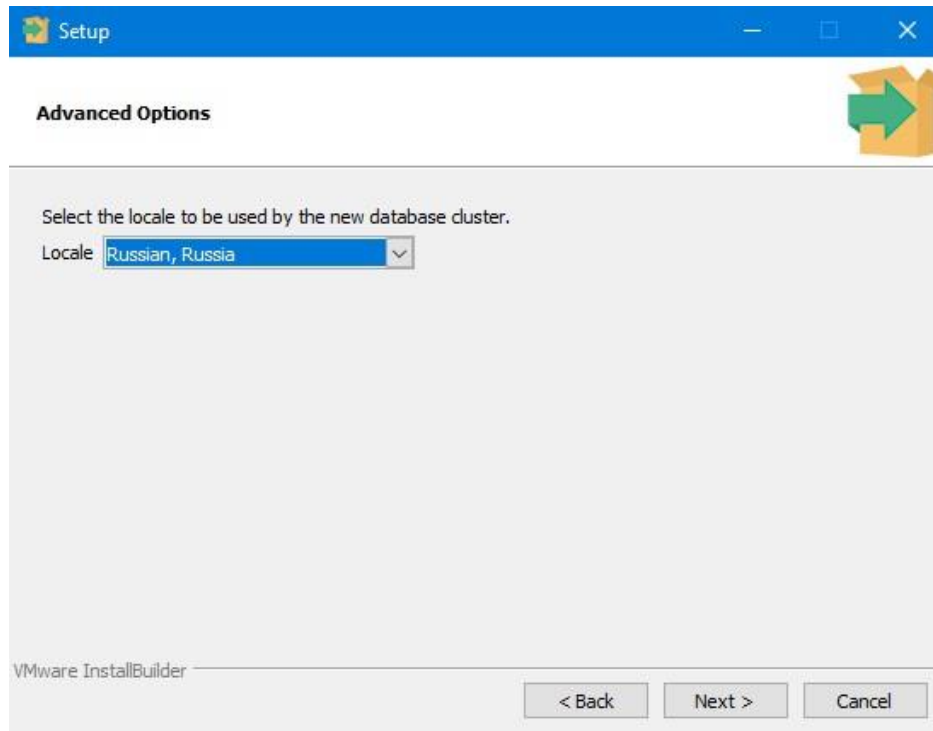
Задаем пароль для пользователя суперпользователя postgres и нажимаем Next>



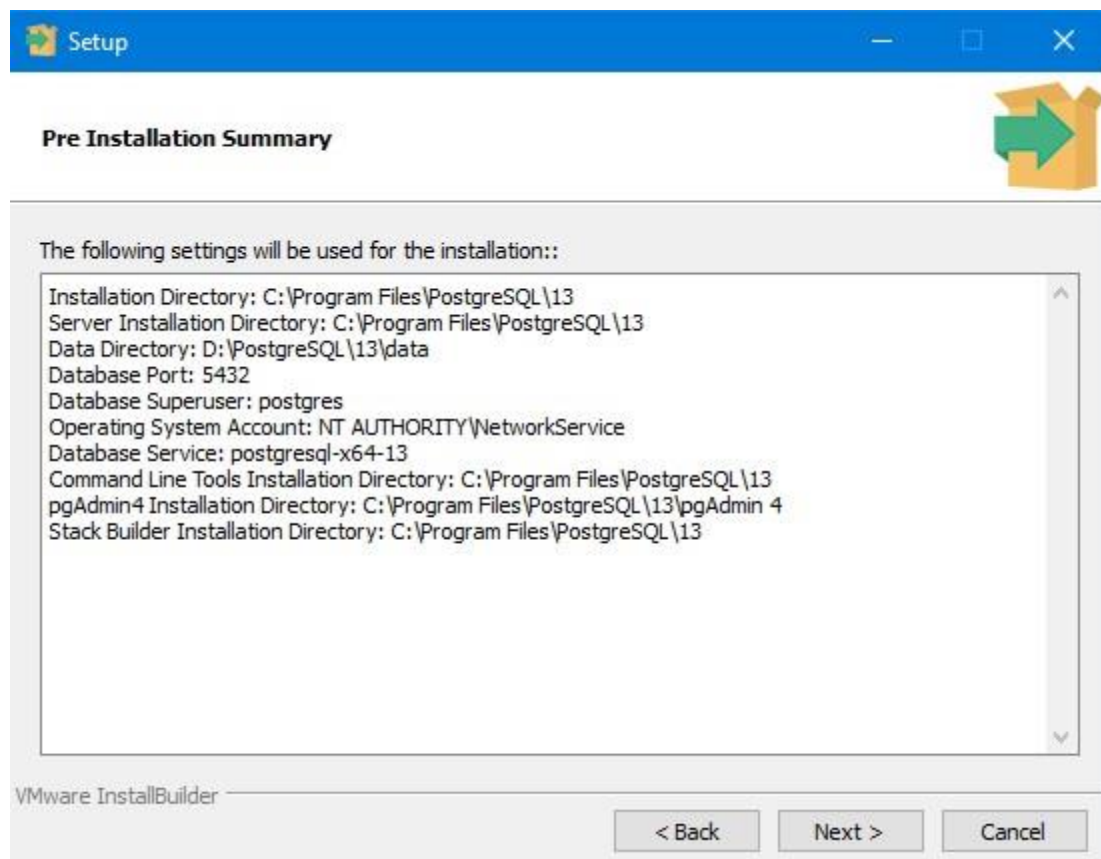
Выбираем порт (или задаем другой, если занят) и нажимаем Next>



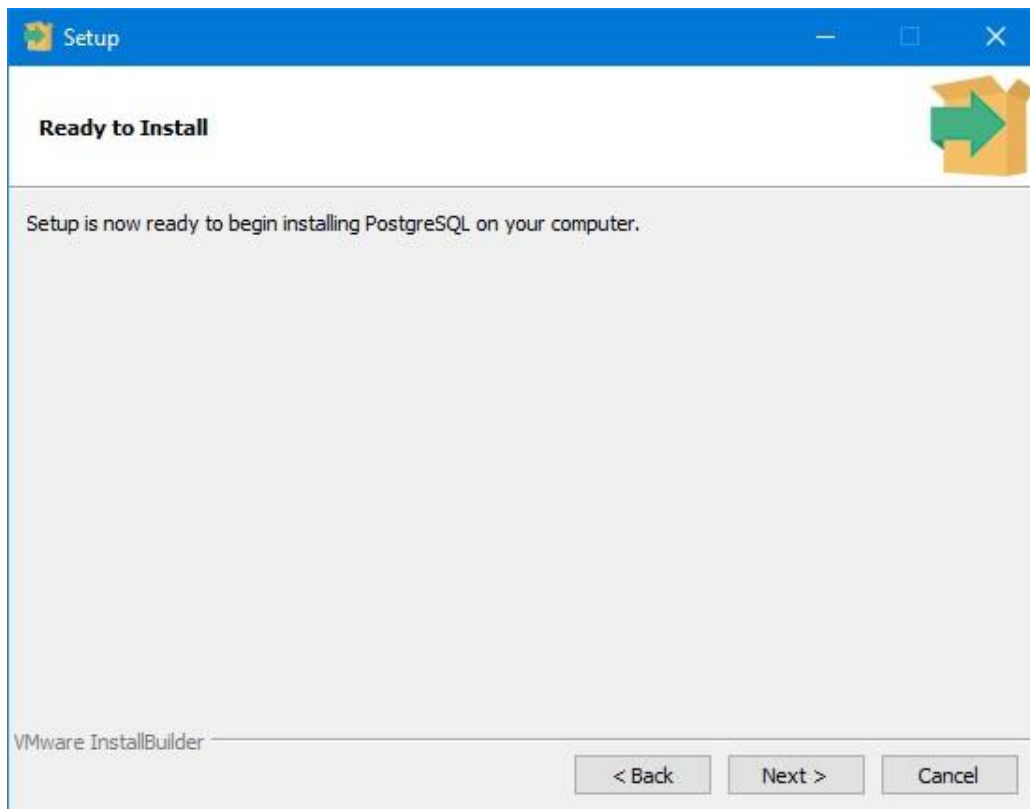
Выбираем локализацию Russian, Russia и нажимаем Next>



Проверяем данные установщика и нажимаем Next>



Запускаем установку, нажав на Next>

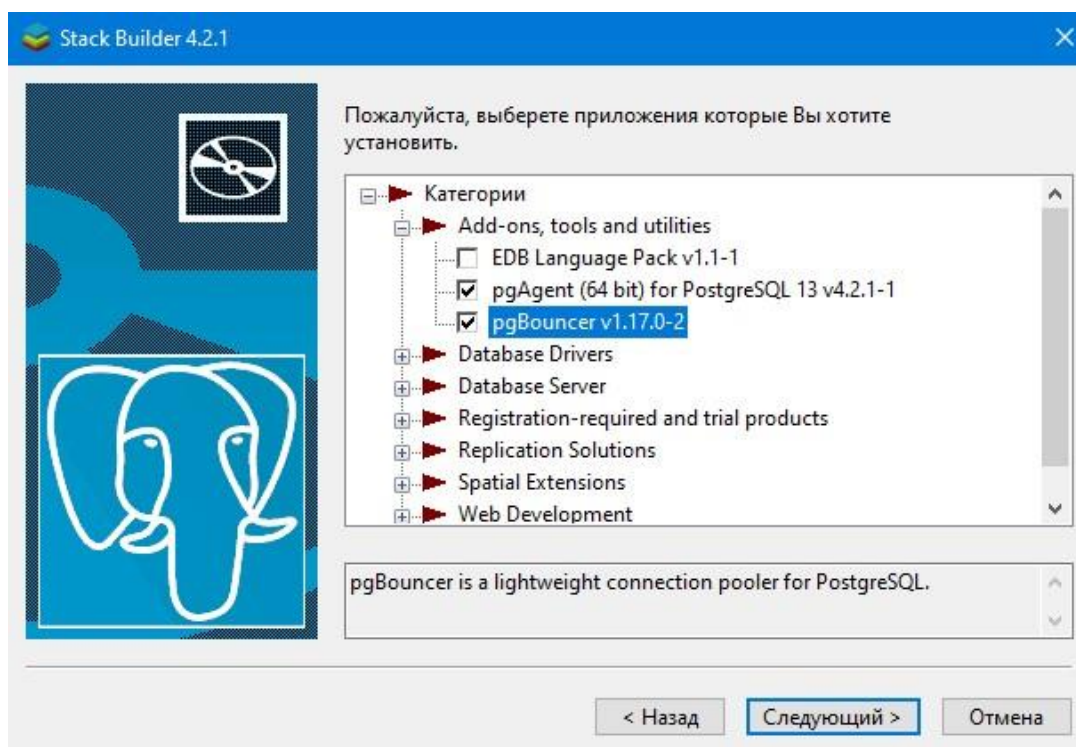


Отобразится окно с прогрессом установки и после завершения нажимаем Finish.



Установим дополнительные компоненты pgAgent

(планировщик заданий) и pgBouncer (балансировщик нагрузки)



После установки необходимо изменить конфигурационные файлы postgresql.conf и pg\_hba.conf в директории с СУБД:

Имя	Дата изменения	Тип	Размер
base	30.01.2023 13:16	Папка с файлами	
global	30.01.2023 14:39	Папка с файлами	
log	30.01.2023 14:39	Папка с файлами	
pg_commit_ts	30.01.2023 13:16	Папка с файлами	
pg_dynshmem	30.01.2023 13:16	Папка с файлами	
pg_logical	30.01.2023 14:44	Папка с файлами	
pg_multixact	30.01.2023 13:16	Папка с файлами	
pg_notify	30.01.2023 13:16	Папка с файлами	
pg_replslot	30.01.2023 13:16	Папка с файлами	
pg_serial	30.01.2023 13:16	Папка с файлами	
pg_snapshots	30.01.2023 13:16	Папка с файлами	
pg_stat	30.01.2023 14:39	Папка с файлами	
pg_stat_tmp	30.01.2023 15:35	Папка с файлами	
pg_subtrans	30.01.2023 13:16	Папка с файлами	
pg_tblspc	30.01.2023 13:16	Папка с файлами	
pg_twophase	30.01.2023 13:16	Папка с файлами	
pg_wal	30.01.2023 13:16	Папка с файлами	
pg_xact	30.01.2023 13:16	Папка с файлами	
current_logfiles	30.01.2023 14:39	Файл	1 КБ
pg_hba.conf	30.01.2023 13:16	Файл "CONF"	5 КБ
pg_ident.conf	30.01.2023 13:16	Файл "CONF"	2 КБ
PG_VERSION	30.01.2023 13:16	Файл	1 КБ
postgresql.auto.conf	30.01.2023 13:16	Файл "CONF"	1 КБ
postgresql.conf	30.01.2023 13:17	Файл "CONF"	29 КБ
postmaster.opts	30.01.2023 14:39	Файл "OPTS"	1 КБ

В файле postgresql.conf изменить listen\_addresses = '\*' и max\_connections = 1000:

```

53 # -----
54 # CONNECTIONS AND AUTHENTICATION
55 # -----
56
57 # - Connection Settings -
58
59 listen_addresses = '*'          # what IP address(es) to listen on;
60                                # comma-separated list of addresses;
61                                # defaults to 'localhost'; use '*' for all
62                                # (change requires restart)
63 port = 5432                      # (change requires restart)
64 max_connections = 1000          # (change requires restart)

```

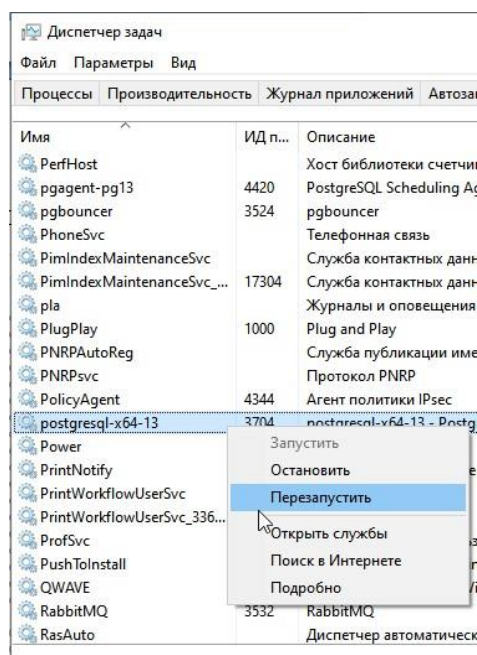
В файле pg\_hba.conf разрешаем подключение со всех адресов к любой базе, используя логин и пароль:

```

80
81 # TYPE DATABASE USER ADDRESS METHOD
82
83 # "local" is for Unix domain socket connections only
84 local all all scram-sha-256
85 # IPv4 local connections:
86 host all all 0.0.0.0/0 md5
87 # IPv6 local connections:
88 host all all ::0/0 md5

```

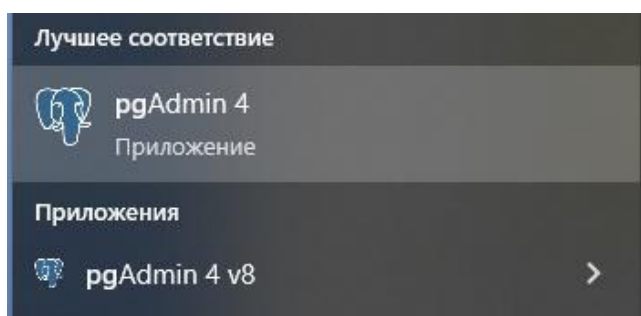
Сохраняем изменения в файлах и перезапускаем службу PSQL, для этого нажимаем сочетание клавиш CTRL + SHIFT + ESC, нажимаем на вкладку "Службы", находим службу PostgreSQL, нажимаем на неё правой клавишей мыши и, в открывшемся контекстном меню "Перезапустить":



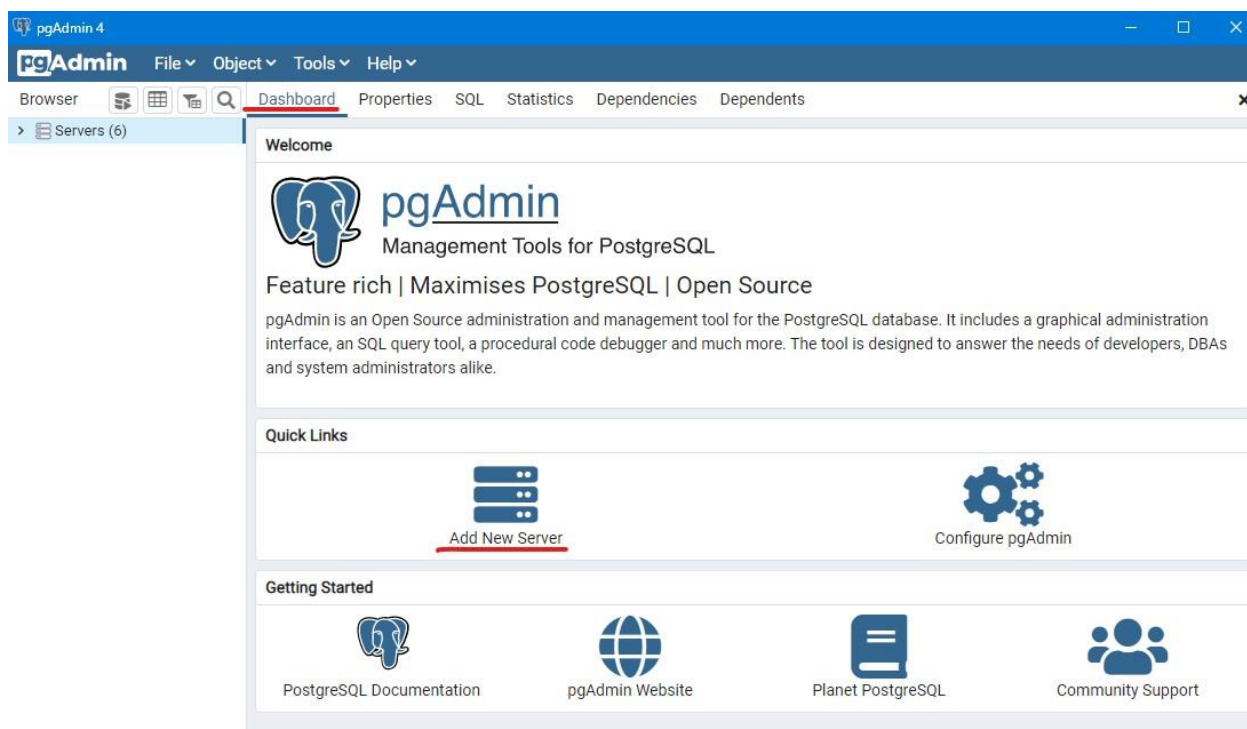
### 3.Подключение к серверу БД и его настройка

Для подключения к серверу БД используется, либо DBeaver, либо pgAdmin, в данном примере будет использоваться pgAdmin.

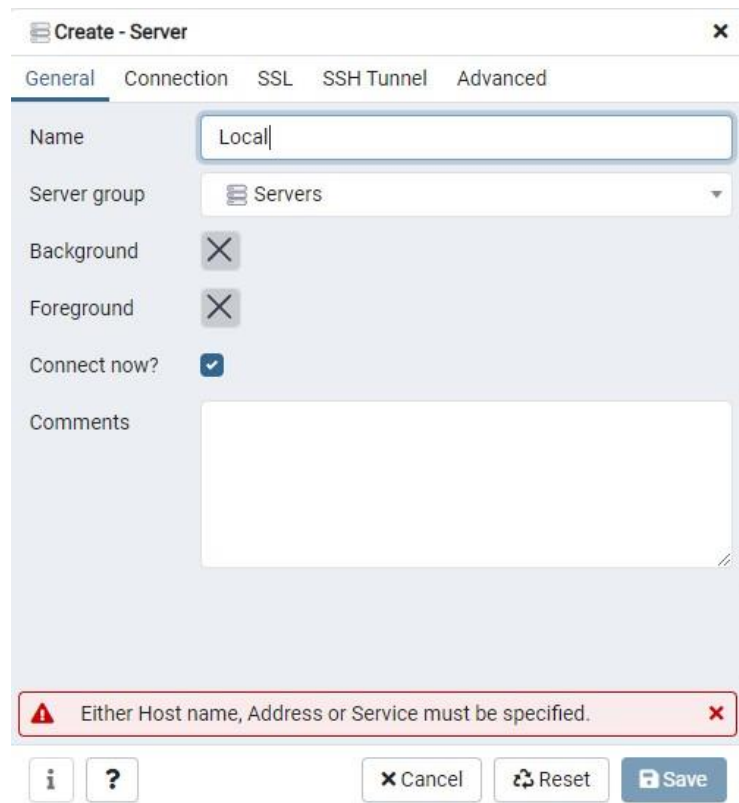
Открываем поиск Windows, набираем pgAdmin и запускаем его.



При первом запуске утилита запросит создание "мастер пароля", задайте любой и не забудьте его сохранить, затем настроим подключение к серверу, для этого переходим в Dashboard и выбираем Add New Server:



На первой вкладке открывшегося окна задаем имя сервера:



На вкладке Connection указываем адрес сервера. В нашем случае необходимо указать IP-адрес рабочего компьютера. Чтобы узнать IP-адрес компьютера, достаточно ввести команду ipconfig в командной строке или терминале VSCode.

Порт и пароль пользователя postgres, указывается такой же, который задавали при установке PostgreSQL.

Register - Server

General Connection Parameters SSH Tunnel Advanced

Host name/address: [свой ip]

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication?:

Password: ..

Save password?:

Role:

Service:

Close Reset Save

На вкладке Parameters устанавливаем значение Allow в поле SSLmode и нажимаем Save.

Register - Server

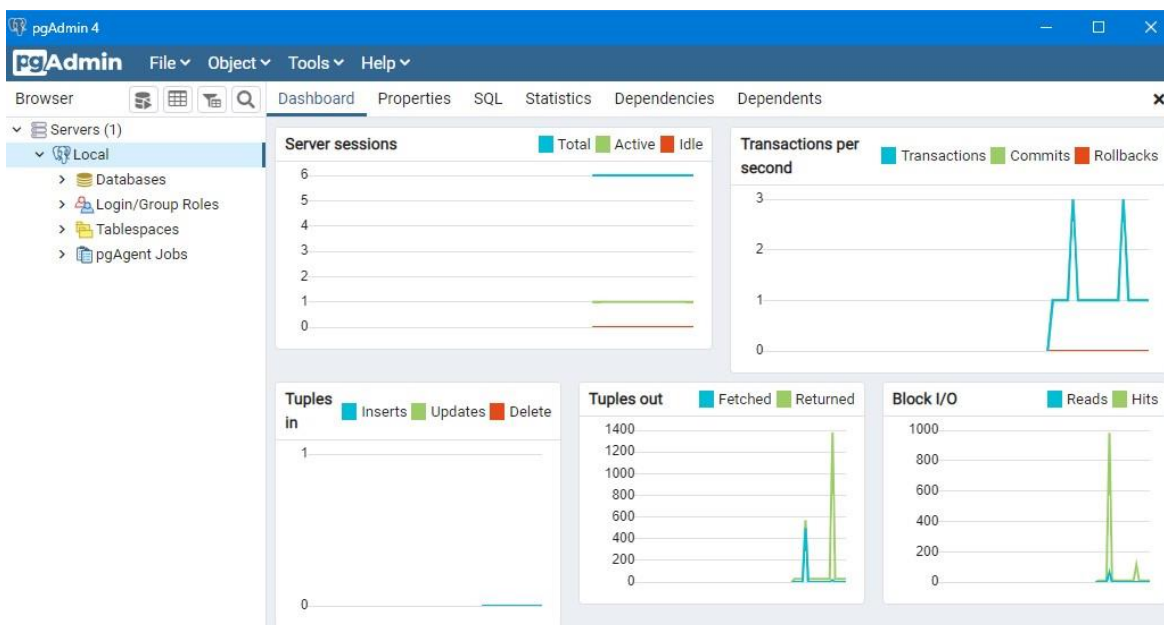
General Connection Parameters SSH Tunnel Advanced

Connection Parameters

Name	Keyword	Value
SSL mode	sslmode	allow
Connection timeo...	connect_timeout	10

Close Reset Save

В левой части окна pgAdmin, в обозревателе отобразится созданное нами подключение:



#### 4. Восстановление базы данных

На данном этапе необходимо иметь Backup базы данных, который будет использоваться с комплексом СТЕК-PCO. Обычно имеет формат demo\_pg\_[номер версии].backup Бэкап базы demo\_2404.backup можно скачать по ссылке:

<https://disk.yandex.ru/d/B68lcFq5OreP2g>

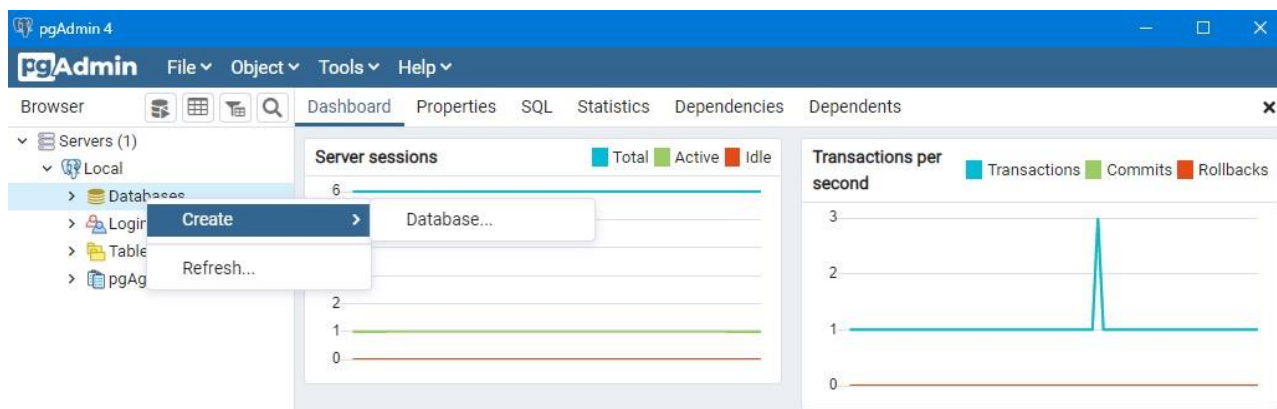
С помощью команд создаём необходимые роли:

```
create role "SA" login password 'Stack450194' superuser createdb createrole inherit
replication connection limit -1;
grant "postgres" to "SA";
```

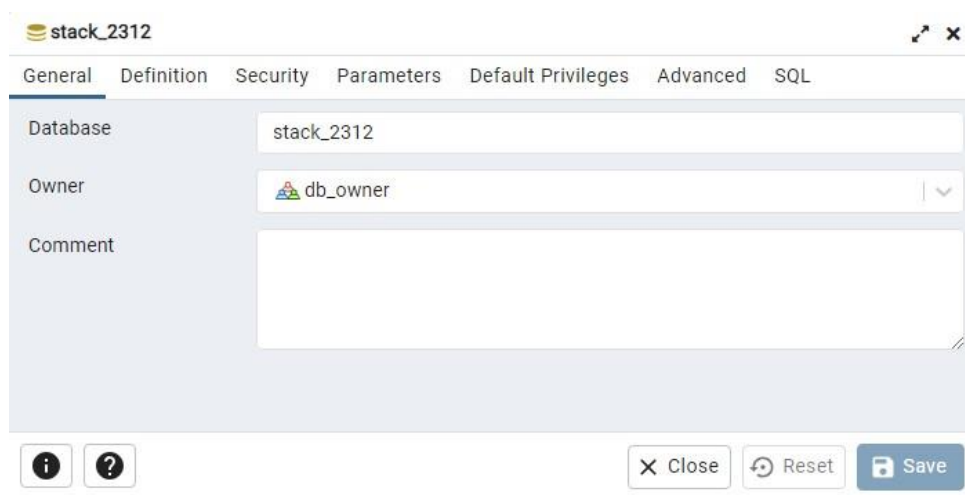
```
create role "SA_ASYNC" login password 'Stack450194' superuser createdb
createrole inherit replication connection limit -1;
grant "postgres" to "SA_ASYNC";
```

```
create role "report" with superuser login password 'report';
grant "postgres" to "report";
```

Создаем БД: для этого на вкладке Databases в обозревателе открываем правой кнопкой мыши меню и выбираем Create Database:



На первой вкладке задаем имя БД, владельца SA и нажимаем Save:



Даём пользователям все права на созданную базу (для примера база demo\_2404):

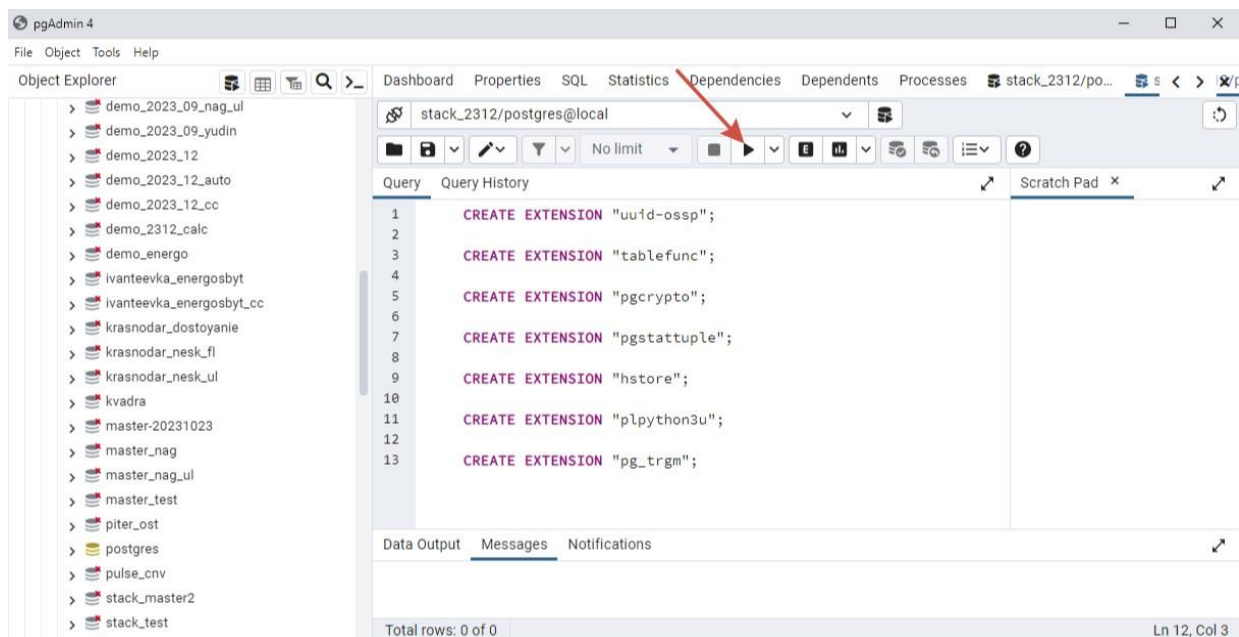
```
grant all privileges on database demo_2404 to "SA";  
grant all privileges on database demo_2404 to "SA_ASYNC";  
grant all privileges on database demo_2404 to "report";
```

Перед восстановлением в созданную базу резервной копии необходимо на БД создать расширения, для чего в обозревателе,

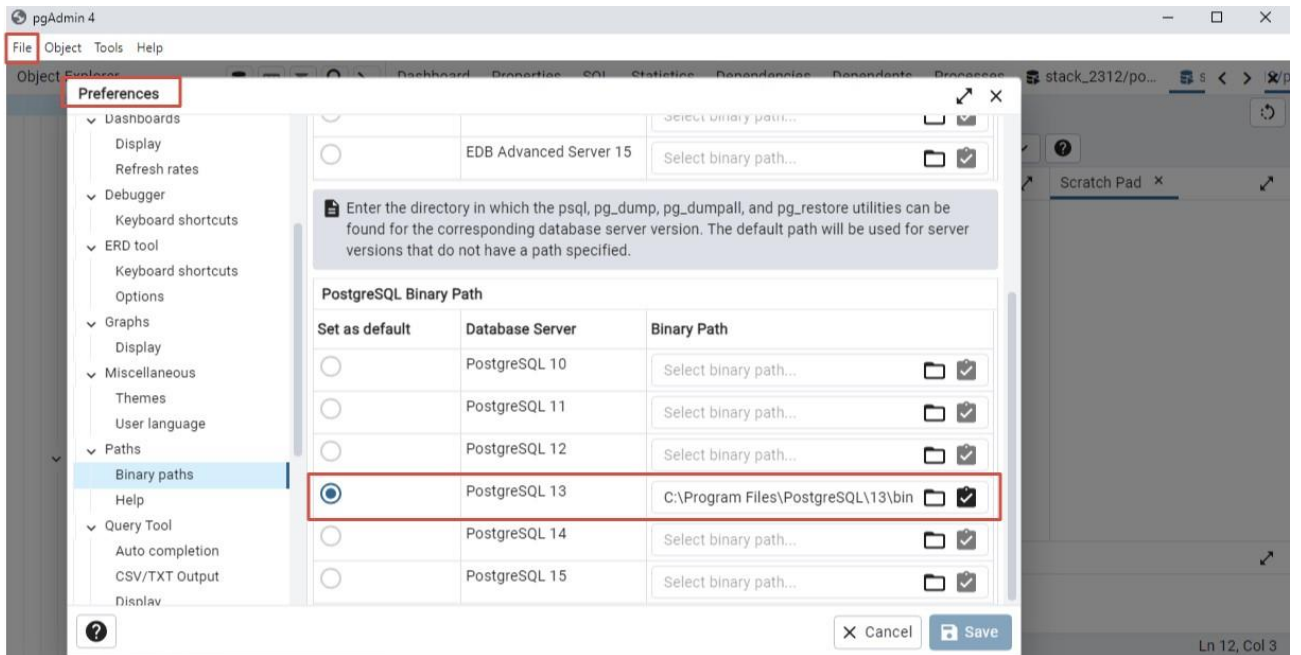
выбираем правой кнопкой мыши базу, в открывшемся меню нажимаем QueryTool:

В окно запроса вставляем расширения, как указано ниже и выполняем запрос:

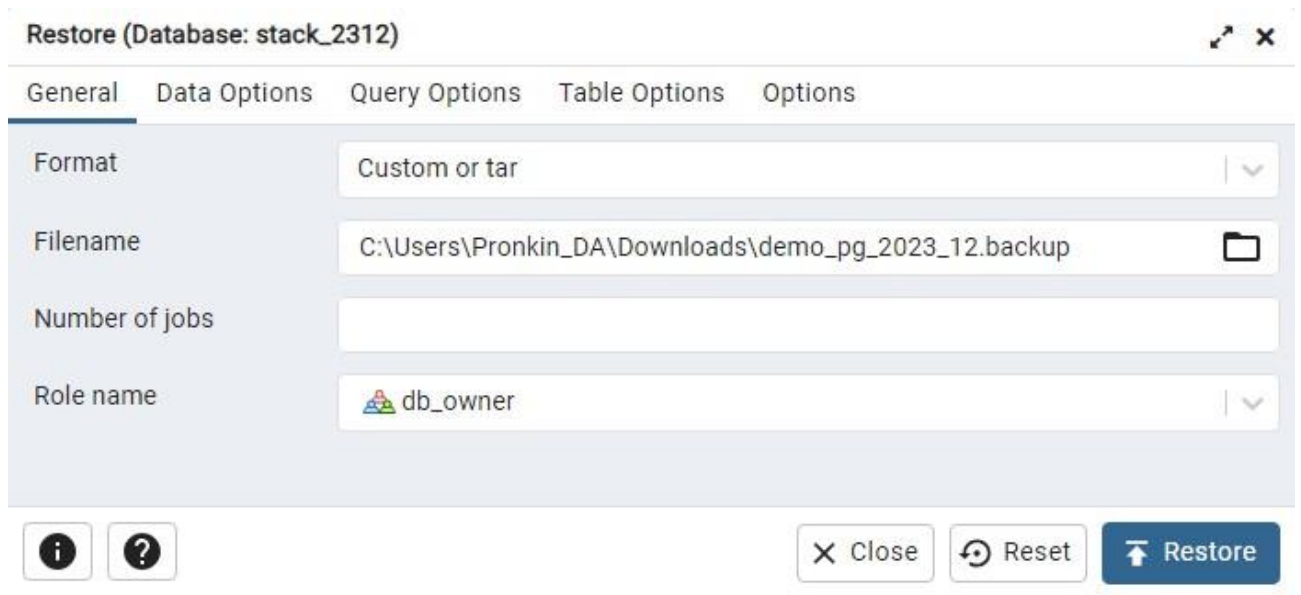
```
CREATE EXTENSION "uuid-oss" with schema public;  
CREATE EXTENSION "tablefunc" with schema public;  
CREATE EXTENSION "pgcrypto" with schema public;  
CREATE EXTENSION "pgstattuple" with schema pg_catalog;  
CREATE EXTENSION "hstore" with schema public;  
CREATE EXTENSION "plpython3u" with schema pg_catalog;  
CREATE EXTENSION "pg_stat_statements" with schema public;  
CREATE EXTENSION "pg_trgm" with schema pg_catalog;  
CREATE EXTENSION "pg_buffercache" with schema public;
```



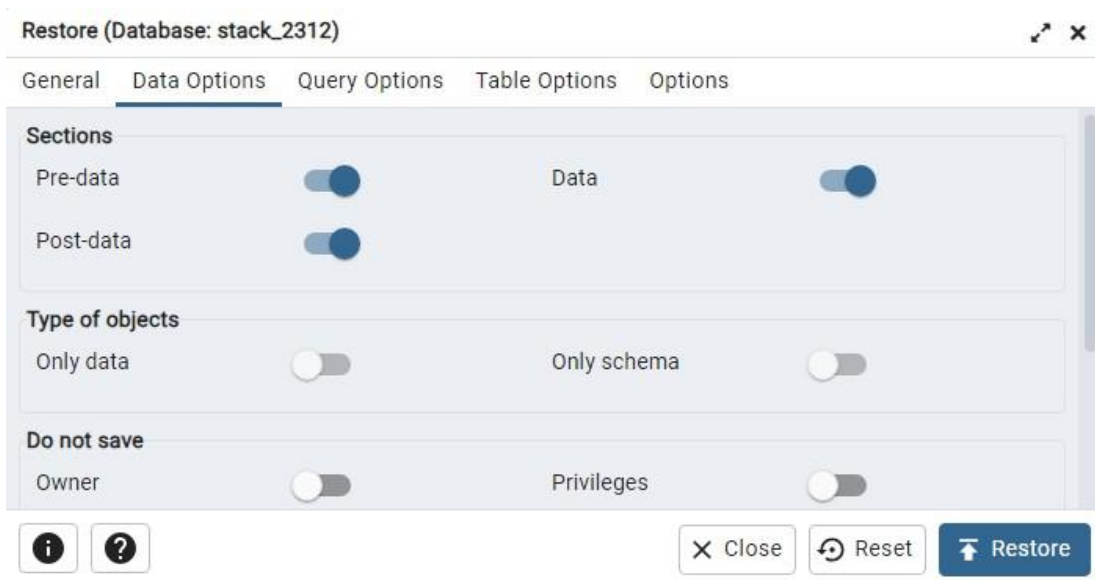
Перед восстановлением БД, необходимо в настройках pgAdmin указать путь к утилитам установленной версии PostgreSQL, для этого открываем File - Preferences-Binary paths и указываем путь до папки bin Postgres:



В открывшемся окне на первой вкладке указываем формат резервной копии, путь к файлу (каталогу) резервной копии и владельца SA:



На следующей вкладке Restore Options выставляем флаги как указано на скриншоте ниже:

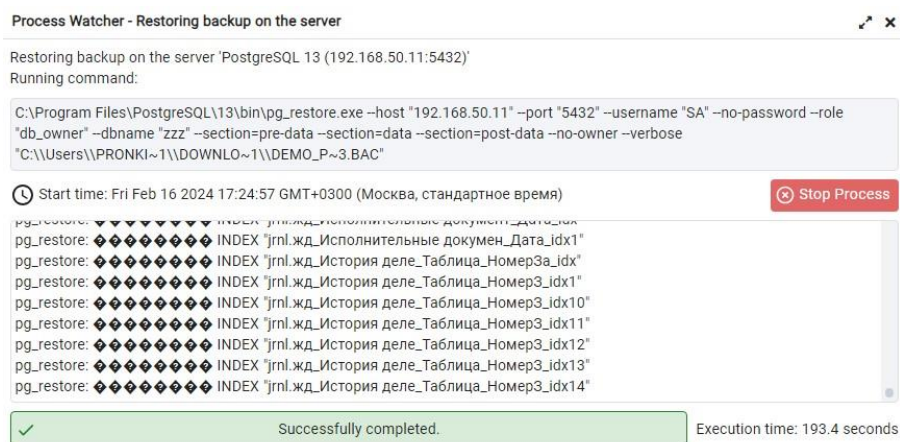


Нажимаем Restore, начнется восстановление БД из резервной копии. Это займет какое-то время.

Для отображения подробной информации хода восстановления нажимаем на кнопку View details баннера:

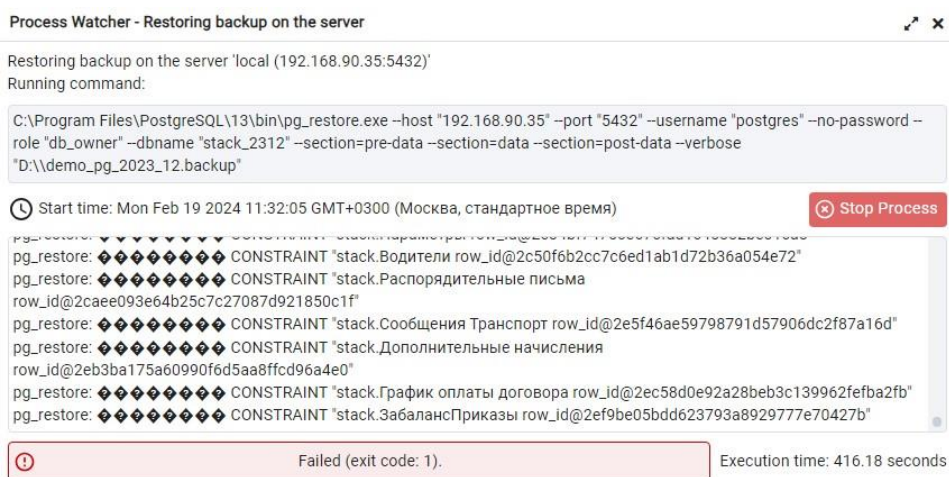


После восстановления появится сообщение о том, что восстановление прошло успешно.



Иногда, в случае восстановления базы перенесенной с другого сервера, операция может завершиться с сообщением об ошибке из-за отсутствия на сервере роли db\_user восстанавливаемой базы.

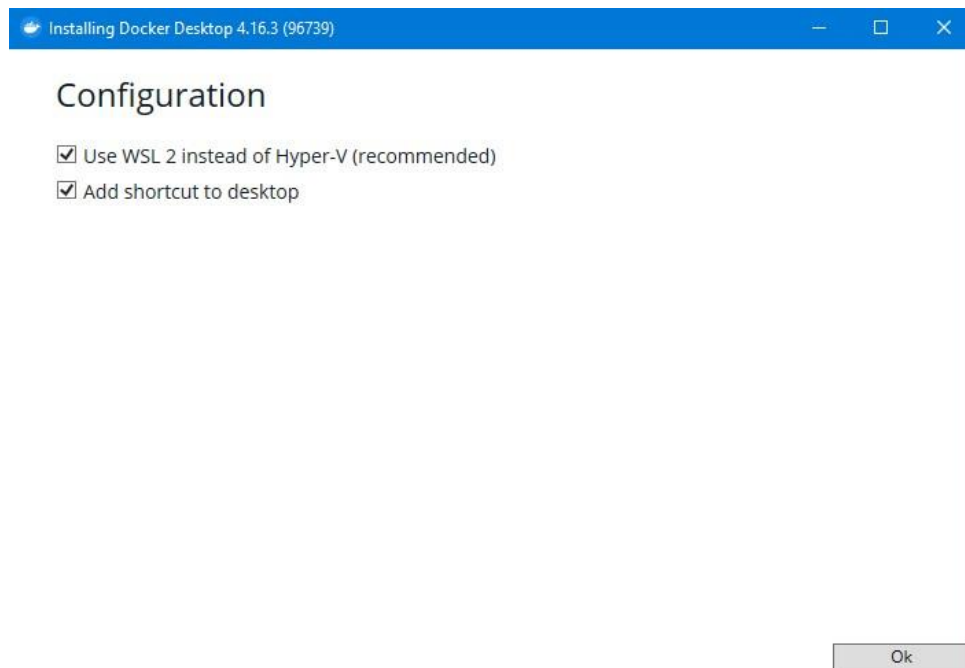
Никаких дополнительных действий в этом случае предпринимать не нужно, пользователь создастся после подключения к базе программным комплексом СТЕК-PCO.



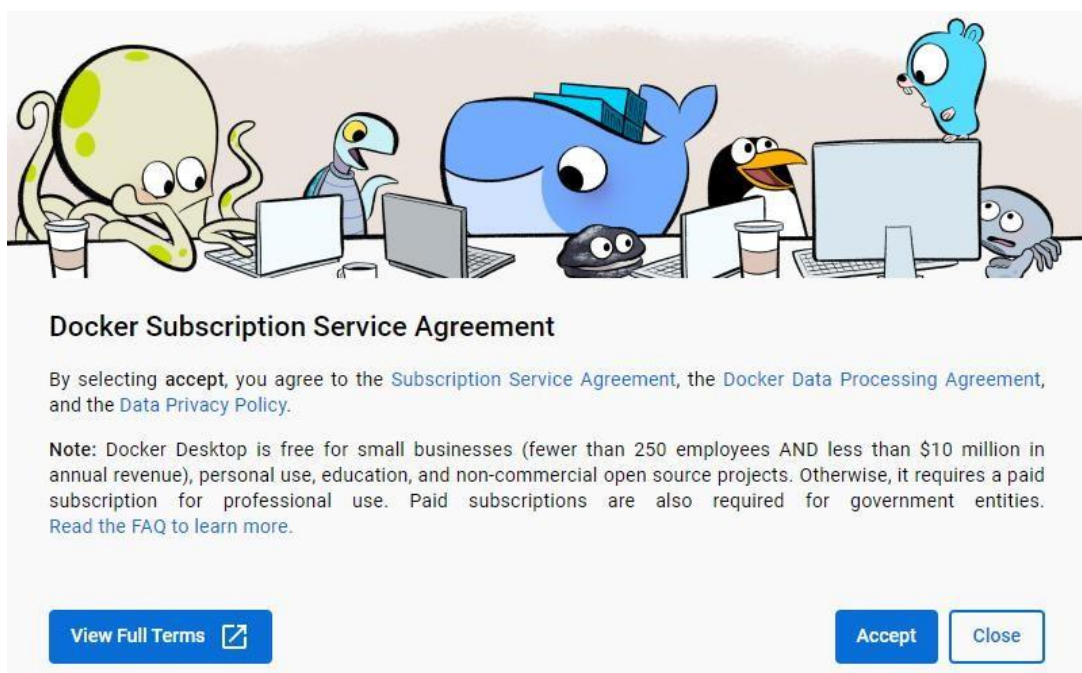
The screenshot shows a window titled "Process Watcher - Restoring backup on the server". It displays the command used for restoring a backup: `C:\Program Files\PostgreSQL\13\bin\pg_restore.exe --host "192.168.90.35" --port "5432" --username "postgres" --no-password --role "db_owner" --dbname "stack_2312" --section=pre-data --section=data --section=post-data --verbose "D:\demo_pg_2023_12.backup"`. The process started on Mon Feb 19 2024 11:32:05 GMT+0300 (Moscow, standard time). The output shows several successful constraint restoration messages, such as `pg_restore: CONSTRAINT "stack.Водители row_id@2c50f6b2cc7c6ed1ab1d72b36a054e72"`. However, the process ended with a failure: `Failed (exit code: 1).` The execution time was 416.18 seconds. A "Stop Process" button is visible in the interface.

## 5. Установка Docker Desktop

Скачиваем Docker Desktop с официального сайта и запускаем установщик. Все отметки оставляем по умолчанию. Нажимаем ОК. После окончания установки соглашаемся с перезагрузкой ОС.



После перезагрузки соглашаемся с лицензионным соглашением, нажимаем Ассепт.



Запускаем Visual Studio Code и устанавливаем следующие расширения:

Docker, Remote - Containers, Remote - SSH (если нужна отладка).

Настраиваем ограничения CPU и оперативной памяти для Docker. Предпочтительно выделять только 4 ядра CPU и максимум 4 ГБ памяти. Для этого:

- Откройте командную строку или PowerShell и выполните следующую команду: `wsl -shutdown` (выключить все экземпляры wsl, такие как Docker Desktop)

- В папке пользователя (C:\Users\[имя пользователя]) из под которого запускается Docker, необходимо создать файл `.wslconfig`

- Редактируем данный файл с помощью блокнота и заносим конфигурацию:

```
[wsl2]
memory=4GBprocessors=4
```

## 6. Настройка конфигурационных файлов и каталогов

На данном этапе необходимо иметь файлы `.env.example` и `docker-compose.yml`. Версия файлов должна быть такой же, какой и версия восстановленной БД.

Файл `.env.example` переименовываем в `.env`, размещаем их в произвольном каталоге.

### **Примечание:**

Можно скачать архив `stack.tar.xz` с файлами `.env` и `docker-compose.yml` можно по ссылке: <https://disk.yandex.ru/d/jxNtmqdmCWj8A> и распаковать архив в выбранную Вами папку. В файле `.env` нужно заменить `<your_ip_addr>` на ip адрес вашего компьютера.

## 6.1. Настройка файла .env

- COMPOSE\_PROJECT\_NAME - имя проекта (при запуске нескольких сборок одновременно, имена должны различаться).
- BUILD - номер сборки, подробнее описано в комментариях файла .env
- IP\_ADDR - IP-адрес рабочего компьютера
- TASK\_NAMES - список задач, перечисляются через пробел.
- SITE\_NAME - имя, по которому будет осуществляться доступ в программу (можно указать как localhost, так и имя, либо IP адрес компьютера)
- NGINX\_PORT - порт, по которому будет доступ в программу (должен быть уникальным для каждого из одновременно запущенных проектов).
- POSTGRES\_HOST - указать адрес корпоративного сервера (смотреть в pgAdmin)
- POSTGRES\_PORT - порт (такой же как в pgAdmin)
- DATABASE\_LOGIN - логин базы (такой же как в pgAdmin)
- DATABASE\_PASSWORD - пароль базы (такой же как в pgAdmin)
- DATABASE\_NAME - имя базы (такой же как в pgAdmin)

## 6.2. Настройка файла docker-compose.yml

В каждом из сервисов, описанных в данном файле, есть параметр `image:` `docker.stack-it.ru/stable/<имя-`

образа>:<версия>. По умолчанию выбрана ветка stable. Так же доступны test и client/<название клиентского проекта>

Если не требуется выбора определенных версий образов или подключения к внешним сервисам (например внешнему RabbitMQ или BirtReporter), то оставляем все как есть и переходим в конец файла.

В конце файла необходимо добавить контейнеры для каждой из задач, которые нужны в работе. По умолчанию будет запускаться admin и fl. Для добавления нужно раскомментировать последний блок и изменить в нем первую строку с именем сервиса и в параметр TASK\_NAME указать задачу (список возможных, fl ul admin dlг\_fl dlг\_ul commun avar passport pdu rsp kassa). Аналогично добавляются остальные задачи.

Необходимо раскомментировать строки:

- "./workspace:/root/stack/stack.srv/workspace"
- "./client76.ini:/root/stack/stack.srv/bin/0/settings/client76.ini"

### 6.3.Каталог workspace

На данном этапе должен быть каталог workspace с такой же версией, как и версии файлов .env и docker-compose.yml.

Помещаем каталог workspace в корень каталог, где расположены .env и docker-compose.yml

### 6.4.Настройка файла client.ini

Далее в нашем каталоге необходимо создать файл client[номер региона].ini и скопировать в него код из шаблона.

[номер региона] – регион использования ПК СТЕК-PCO, например в Ярославской области, файл называется client76.ini.

[AppPath]

RS=/root/stack/stack.srv/workspace/Common/RX

RS=/root/stack/stack.srv/workspace/Kvplata/RX

RS=/root/stack/stack.srv/workspace/Kvplata/RX/REGION/76

RS=/root/stack/stack.srv/workspace/CommonApi/RX

RS=/root/stack/stack.srv/workspace/CommonApi/RX/Ru

RS=/root/stack/stack.srv/workspace/Kvplata/Stack.API/rx

DB=/root/stack/stack.srv/workspace/Common/DB

DB=/root/stack/stack.srv/workspace/Kvplata/DB

DB=/root/stack/stack.srv/workspace/Kvplata/DB/REGION/76

DB=/root/stack/stack.srv/workspace/CommonApi/DB

DB=/root/stack/stack.srv/workspace/Kvplata/Stack.API/DB

PRG=/root/stack/stack.srv/workspace/Common/prg

PRG=/root/stack/stack.srv/workspace/Common/reporter

PRG=/root/stack/stack.srv/workspace/Kvplata/prg

PRG=/root/stack/stack.srv/workspace/CommonApi/prg/RabbitMQ

PRG=/root/stack/stack.srv/workspace/Kvplata/prg/REGION/76

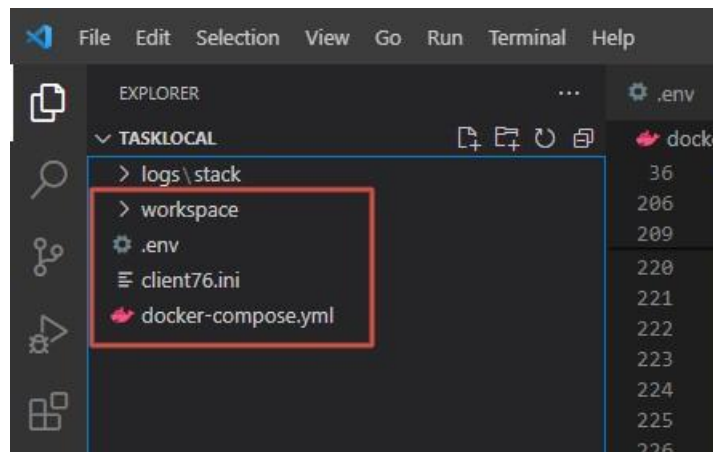
PRG=/root/stack/stack.srv/workspace/CommonApi/prg

PRG=/root/stack/stack.srv/workspace/Kvplata/Stack.API/prg

RPT=/root/stack/stack.srv/workspace/Kvplata/rpt

## 6.5.Итоговый каталог с конфигурационными файлами

В результате в нашем каталоге должен получиться такой набор файлов и папок:



## Запуск программного комплекса СТЕК-PCO

Скачиваем два архива dotnetcore.tar.xz и images.tar.xz с образами docker по ссылкам:

<https://disk.yandex.ru/d/G7ZBV7qj66lemA>

<https://disk.yandex.ru/d/trNAhodskHcslA>

Убедитесь, что Docker Desktop запущен.

Распакуем архивы в произвольную папку и из консоли установим образы:

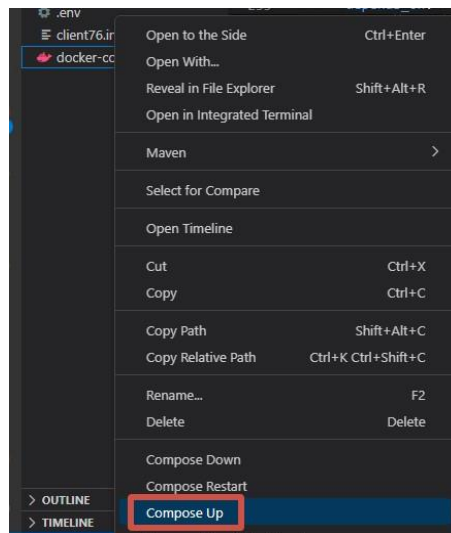
```
docker load < dotnetcore.tar
```

```
docker load < images.tar
```

В файле docker-compose.yml для сервиса app\_host\_admin в разделе environment нужно указать переменную:

```
APPHOST_PROGRAM_NAME: "Стек-PCO"
```

Откройте IDE VSCode. В VSCode нажимайте правой кнопкой мыши на файл docker-compose.yml и выберите Compose Up.

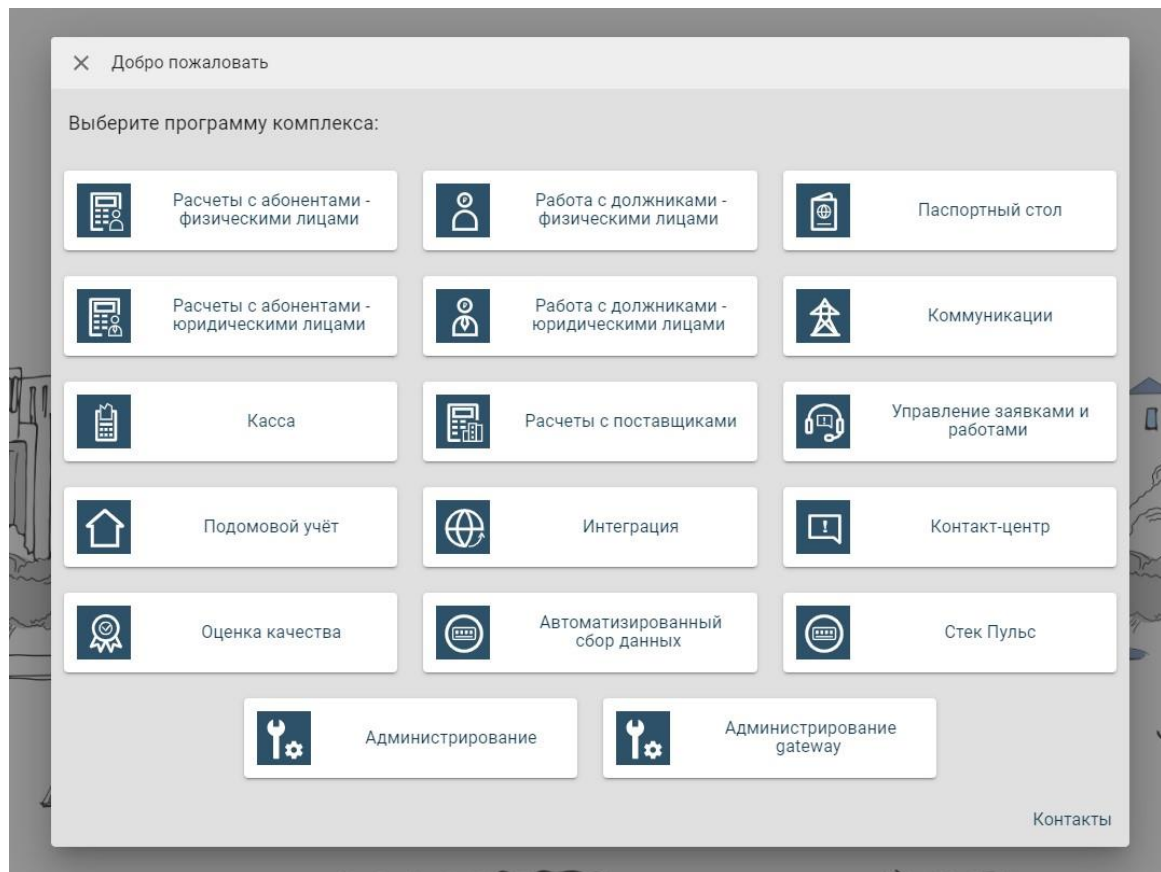


Первый запуск займет пару минут. После успешного запуска контейнеров, их статус можно проверить в расширении Docker (или в самом Docker Desktop).

Примечание:

Таким же образом можно запустить программу из папки stack (распакованного архива stack.tar.xz в папку stack по выше указанной ссылке).

Для входа программы необходимо в браузере ввести `http://` + значение переменной `IP_ADDR` (свой IP) + `NGINX_PORT` (по умолчанию 80). `IP_ADDR` и `NGINX_PORT` задается в файле `.env`.  
Например, `http://192.168.50.103:80`



## Дополнительная информация

### Команды для работы с Docker-compose:

Все команды выполняются в каталоге, где находится файл конфигурации `docker-compose.yml`

`docker-compose pull` — выкачивает изменения из хранилища образов, когда необходимо обновить сборку;

`docker-compose up -d` — запуск/перезапуск образов.

Команды для работы со сборкой внутри контейнера:

Конвертация БД.

Внутри админского контейнера `app_host` в консоли выполняем команду `/root/scripts/convert-db`

Типичные ошибки запуска Docker:

Команда `docker-compose pull` выдаёт ошибку: `services.app_host.environment.ONETIMEQUERY contains true, which is an invalid type, it should be a string, number, or a null`

Решение: в настройках Docker включить пункт Use Docker Compose V2