



DevCon School

Технологии будущего

Postgres Pro в облаке Microsoft Azure

Иван Панченко, Алексей Шишкин
Postgres Professional

Задача 1

Вы узнаете как установить
Postgres Pro в Microsoft
Azure

Задача 2

Вы научитесь выполнять
команды SQL в Postgres Pro

Задача 3

Вы познакомитесь с рядом
особенностей Postgres Pro,
научитесь работать с
массивами и JSON.

□ Отзывы

Помогите нам стать лучше!

На вашу почту отправлена индивидуальная ссылка на электронную анкету. 3 июня в 23:30 незаполненная анкета превратится в тыкву.

Заполните анкету и подходите к стойке регистрации за приятным сюрпризом!

#msdevcon

Оставляйте отзывы в социальных сетях. Мы все читаем. Спасибо вам! 😊

Postgres Pro в облаке Microsoft Azure



PROFESSIONAL
Postgres

postgrespro.ru

Установка БД Postgres Pro Standard 9.6.2 в облаке Microsoft Azure

- ◆ Для примера используется образ виртуальной машины (VM) Postgres Pro Standard 9.6.2, доступный на Microsoft Azure Marketplace
- ◆ Необходимо иметь доступ в Интернет и учетную запись в Microsoft Azure
- ◆ Требуемое ПО
 - Azure CLI 2.0
 - psql или Pgadmin

Состав VM Postgres Pro Standard 9.6.2 (1/4)

- ◆ ОС - Linux CentOS 7.3 (64-бит)
 - учетная запись - postgres
 - учетная запись - zabbix
 - учетная запись - mamonsu

- ◆ Размер ОС-диска - 50 ГБ
 - файловая система ext4 '/boot' (/dev/sda1) - 1 ГБ
 - файловая система ext4 '/' (/dev/sda2) - 49 ГБ

Состав VM Postgres Pro Standard 9.6.2 (2/4)

- ◆ Основная БД - Postgres Pro 9.6.2
 - порт: 5433 (открыт на firewall)
 - конфигурация: `/var/lib/pgpro/9.6/data/postgresql.conf`,
`/var/lib/pgpro/9.6/data/postgresql.auto.conf`
 - путь к `pg_config`: `/usr/pgpro-9.6/bin/pg_config`
 - путь к `psql`: `/usr/pgpro-9.6/bin/psql`
 - учетная запись - `postgres/azure` (при изменении пароля учетной записи необходимо актуализировать изменения в `/etc/mamonsu/agent.conf`)

Состав VM Postgres Pro Standard 9.6.2 (3/4)

◆ Мониторинг БД (сервер)

- zabbix-server 3.0
- порт: 80 (открыт на firewall)
- учетная запись - Admin/zabbix

◆ Мониторинг БД (агент)

- zabbix-agent
- агент matonsu 2.2.9 (в качестве агента zabbix, снимающего метрики с Postgres Pro)

Состав VM Postgres Pro Standard 9.6.2 (4/4)

- ◆ Вспомогательная БД - PostgreSQL 9.2 (в качестве БД для сервера zabbix)
 - порт: 5432
 - конфигурация: `/var/lib/pgsql/data/postgresql.conf`
 - путь к `pg_config`: `/usr/bin/pg_config`
 - путь к `psql`: `/usr/bin/psql`

Ссылки на документацию

◆ Postgres Pro Standard:

<https://postgrespro.ru/docs/postgrespro>

◆ Zabbix:

<https://www.zabbix.com/documentation/3.0/start>

◆ Mamonsu:

<https://github.com/postgrespro/mamonsu/blob/master/README.rst>

◆ Pgadmin

<https://www.pgadmin.org/docs>

Azure CLI 2.0

- ◆ Используем кросс-платформенную утилиту Azure CLI 2.0:
 - Также можно использовать Azure Portal <https://portal.azure.com> или Azure PowerShell.
 - Установка Azure CLI 2.0 описана в <https://docs.microsoft.com/ru-ru/cli/azure/install-azure-cli>
 - Основные команды Azure CLI 2.0 описаны в <https://docs.microsoft.com/ru-ru/cli/azure/get-started-with-azure-cli>
 - Полный список команд Azure CLI 2.0 можно найти на <https://docs.microsoft.com/ru-ru/cli/azure>
 - Установка и администрирование Linux Virtual Machine описаны в <https://docs.microsoft.com/ru-ru/azure/virtual-machines/linux>

Подключение к Azure и проверка окружения (1/4)

- ◆ Подключаемся к Microsoft Azure с помощью команды:

az login

или

az login --username <myusername>

- ◆ Установим табличный формат вывода команд Azure CLI 2.0:

az configure --output table

- ◆ Проверяем установленную версию Azure CLI 2.0 (должна быть не ниже 2.0.4):

az --version

Подключение к Azure и проверка окружения (2/4)

- ◆ Проверим, зарегистрированы ли необходимые сервисы Microsoft.Storage, Microsoft.Compute и Microsoft.Network:

```
az provider show --namespace Microsoft.Storage
```

```
az provider show --namespace Microsoft.Compute
```

```
az provider show --namespace Microsoft.Network
```

- ◆ Если нет, то регистрируем их:

```
az provider register --namespace Microsoft.Storage
```

```
az provider register --namespace Microsoft.Compute
```

```
az provider register --namespace Microsoft.Network
```

Подключение к Azure и проверка окружения (3/4)

- ◆ Список географических регионов для размещения VM:

az account list-locations

Далее будем использовать географический регион 'northeurope'.

- ◆ Список доступных размеров VM в географическом регионе 'northeurope':

az vm list-sizes --location northeurope

Далее будем использовать размер VM 'Standard_DS1' (доступен для 'Free Trial' подписки).

Подключение к Azure и проверка окружения (4/4)

- ◆ Получаем название издателя образа VM Postgres Pro в Microsoft Azure Marketplace в географическом регионе 'northeurope':

```
az vm image list-publishers \  
--location northeurope \  
--query "[?starts_with(name, 'postgres')]"
```

Далее будем использовать название издателя образа VM 'postgres-pro'.

- ◆ Получаем название образа VM Postgres Pro, доступного на Microsoft Azure Marketplace от издателя 'postgres-pro' в географическом регионе 'northeurope':

```
az vm image list \  
--all \  
--location northeurope \  
--publisher postgres-pro
```

Далее будем использовать название образа VM 'postgres-pro:postgres-pro-standard:hrly:9.6.2'.

Создание VM (1/5)

- ◆ Настройка программного развертывания на портале <https://portal.azure.com>:
 - меню 'Создать'
 - меню 'Поиск по Marketplace' - 'Postgres Pro Standard 9.6.2'
 - 'Хотите выполнить развертывание программным способом? Начните работу ->'
 - Состояние подписки 'Включить'
 - 'Сохранить'

Создание VM (2/5)

◆ Создаем ресурсную группу:

```
az group create \
```

```
--location northeurope \
```

```
--name myresourcegroup
```

Создание VM (3/5)

- ◆ Создаем VM из образа, доступного на Microsoft Azure Marketplace:

```

az vm create \
--name myvm-xx \
--resource-group myresourcegroup \
--image postgres-pro:postgres-pro-standard:hrly:9.6.2 \
--location northeurope \
--size Standard_DS1 \
--admin-password Azure1234567890+ \
--admin-username azureuser \
--authentication-type password \
--public-ip-address-dns-name myvm-dnsname-xx \
--os-disk-name myosdisk-xx

```

- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Создание VM (4/5)

- ◆ (Опционально) Выполняем дополнительные настройки ОС VM:

```

az vm extension set \
--resource-group myresourcegroup \
--vm-name myvm-xx \
--name customScript \
--publisher Microsoft.Azure.Extensions \
--settings script-config-pgpro_profile.json
  
```

- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Создание VM (5/5)

- ◆ (Опционально) Содержимое файла 'script-config-pgpro_profile.json':

```

{
  "commandToExecute": "echo 'export PGHOME=/usr/pgpro-9.6' > /var/lib/pgsql/.pgpro_profile; \
echo 'export PATH=$PGHOME/bin:$PATH' >> /var/lib/pgsql/.pgpro_profile; \
echo 'export MANPATH=$PGHOME/share/man:$MANPATH' >> /var/lib/pgsql/.pgpro_profile; \
echo 'export PGDATA=/var/lib/pgpro/9.6/data' >> /var/lib/pgsql/.pgpro_profile; \
echo 'export PGDATABASE=postgres' >> /var/lib/pgsql/.pgpro_profile; \
echo 'export PGUSER=postgres' >> /var/lib/pgsql/.pgpro_profile; \
echo 'export PGHOST=127.0.0.1' >> /var/lib/pgsql/.pgpro_profile; \
echo 'export PGPORT=5433' >> /var/lib/pgsql/.pgpro_profile; \
chown postgres:postgres /var/lib/pgsql/.pgpro_profile; \
chmod 0640 /var/lib/pgsql/.pgpro_profile"
}
  
```

Подключение к VM

- ◆ В результате создается VM:

- с полным доменным именем 'myvm-dnsname-xx.northeurope.cloudapp.azure.com' (полное доменное имя формируется из короткого доменного имени, задаваемого при создании VM, названия географического региона и 'cloudapp.azure.com')

- и с ОС-пользователем VM 'azureuser' (по умолчанию с правами 'sudo') с паролем 'Azure1234567890+' (сложность пароля определяется требованиями)

- ◆ Подключаемся к VM:

```
ssh azureuser@myvm-dnsname-xx.northeurope.cloudapp.azure.com
```

- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Проверка статуса сервиса БД Postgres Pro

- ◆ Проверим статус сервиса БД Postgres Pro:

```
sudo systemctl -l status postgrespro-9.6.service
```

- ◆ Чтобы остановить/стартовать сервис БД Postgres Pro используем команды:

```
sudo systemctl stop postgrespro-9.6.service
```

```
sudo systemctl start postgrespro-9.6.service
```

Подключение к БД Postgres Pro

- ◆ Переключаемся на пользователя 'postgres':

```
sudo su - postgres
```

- ◆ Для подключения к БД Postgres Pro используем команду:

```
psql --port=5433
```

- ◆ Для выхода из 'psql' используем команду:

```
\q
```

Внешний доступ к VM

- ◆ При необходимости внешнего подключения к БД Postgres Pro открываем порт 5433 (команда выполняется с сообщением об ошибке, которое можно игнорировать):

```
az vm open-port \  
--port 5433 \  
--resource-group myresourcegroup \  
--name myvm-xx \  
--priority 1001
```

- ◆ При необходимости внешнего подключения к серверу мониторинга БД открываем порт 80 (команда выполняется с сообщением об ошибке, которое можно игнорировать):

```
az vm open-port \  
--port 80 \  
--resource-group myresourcegroup \  
--name myvm-xx \  
--priority 1002
```

- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Внешнее подключение к БД Postgres Pro

- ◆ Для внешнего подключения к БД Postgres Pro с помощью 'psql' используем команду:

```
psql --host=myvm-dnsname-xx.northeurope.cloudapp.azure.com --port=5433 --username=postgres --dbname=postgres
```

- ◆ Для внешнего подключения к БД Postgres Pro с помощью 'Pgadmin' при конфигурации сервера в меню 'Pgadmin' используем:
 - 'mydb-xx' для 'Name'
 - 'myvm-dnsname-xx.northeurope.cloudapp.azure.com' для 'Host'
 - '5433' для 'Port'
 - 'postgres' для 'Maintenance DB'
 - 'postgres' для 'Username'
- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Внешнее подключение к серверу мониторинга БД

- ◆ Внешнее подключение к серверу мониторинга БД по ссылке:

<http://myvm-dnsname-xx.northeurope.cloudapp.azure.com/zabbix>

- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Изменение конфигурации VM (1/6)

- ◆ В этом примере использовался образ VM Postgres Pro 9.6.2 и VM размера 'Standard_DS1'
- ◆ Эта конфигурация имеет следующие ограничения:
 - 1 ядро CPU
 - объем оперативной памяти - 3.5 ГБ
 - размер временного диска - 7 ГБ
 - максимальное количество подключаемых дисков данных - 2
 - размер ОС-диска - 50 ГБ
- ◆ Как изменить эти ограничения?

Изменение конфигурации VM (2/6)

◆ Изменение количества ядер

- Количество ядер в VM зависит от размера VM.
- Изменение возможно выполнить с помощью изменения размера VM:

```
az vm list-vm-resize-options
```

```
az vm resize
```

- Требуется перезагрузка VM.

Изменение конфигурации VM (3/6)

- ◆ Изменение объема оперативной памяти
 - Объем оперативной памяти в VM зависит от размера VM.
 - Изменение возможно выполнить с помощью изменения размера VM:

az vm list-vm-resize-options

az vm resize

- Требуется перезагрузка VM.

Изменение конфигурации VM (4/6)

- ◆ Изменение размера временного диска
 - Размер временного диска зависит от размера VM.
 - Изменение возможно выполнить с помощью изменения размера VM:

az vm list-vm-resize-options

az vm resize

- Требуется перезагрузка VM.

Изменение конфигурации VM (5/6)

- ◆ Изменение количества дисков данных
 - Количество дисков данных в VM зависит от размера VM.
 - Изменение возможно выполнить с помощью изменения размера VM:

az vm list-vm-resize-options

az vm resize

az vm disk attach

- Требуется перезагрузка VM.

Изменение конфигурации VM (6/6)

◆ Увеличение размера ОС-диска

- Выполняется с помощью команд:

az vm deallocate

az disk update

az vm start

- Увеличение размера раздела ОС-диска и его файловой системы выполняется вручную с помощью команд ОС (**fdisk** и **resize2fs**).

- Предельный размер ОС-диска - 1023 ГБ.

- Требуется перезагрузка VM.

◆ Увеличение размера диска данных

- Так же как для ОС-диска

Пример: изменение размера VM

- ◆ В качестве примера изменим размер VM со 'Standard_DS1' на 'Standard_DS3'
- ◆ Список доступных размеров VM в географическом регионе 'northeurope' (выполнять команду 'az vm deallocate' не требуется):

```
az vm list-vm-resize-options \
```

```
--name myvm-xx \
```

```
--resource-group myresourcegroup
```

- ◆ Чтобы изменить размер VM, выполняем:

```
az vm resize \
```

```
--name myvm-xx \
```

```
--resource-group myresourcegroup \
```

```
--size Standard_DS3
```

- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Пример: увеличение размера ОС-диска (1/3)

- ◆ В качестве примера увеличим размер ОС-диска с 50 ГБ до 80 ГБ
- ◆ Получаем информацию об ОС-диске:

```
az disk list \  
--resource-group myresourcegroup \  
--query '[*].{Name:name,Gb:diskSizeGb,Tier:accountType}'
```

- ◆ Временно отключаем VM:

```
az vm deallocate \  
--resource-group myresourcegroup \  
--name myvm-xx
```

- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Пример: увеличение размера ОС-диска (2/3)

- ◆ Увеличиваем размер ОС-диска:

```
az disk update \  
--name myosdisk-xx \  
--resource-group myresourcegroup \  
--size-gb 80
```

- ◆ Проверяем обновленную информацию об ОС-диске:

```
az disk list \  
--resource-group myresourcegroup \  
--query '[*].{Name:name,Gb:diskSizeGb,Tier:accountType}'
```

- ◆ Стартуем VM:

```
az vm start \  
--resource-group myresourcegroup \  
--name myvm-xx
```

- ◆ Вместо 'xx' используем '01', '02', '03' и т.д.

Пример: увеличение размера ОС-диска (3/3)

- ◆ Подключаемся к VM:

```
ssh azureuser@myvm-dnsname-xx.northeurope.cloudapp.azure.com
```

- ◆ Увеличиваем размер раздела файловой системы '/':

```
(echo d; echo 2; echo n; echo p; echo 2; echo ; echo ; echo w) | sudo \
fdisk /dev/sda
```

- ◆ Выполняем перезагрузку VM:

```
sudo reboot
```

- ◆ Увеличиваем размер файловой системы '/':

```
sudo resize2fs /dev/sda2
```

- ◆ Выполняем перезагрузку VM:

```
sudo reboot
```

Соединение с БД (1/3)

- ◆ Разные версии PostgreSQL на Вашей машине в Azure

```
sudo -u postgres psql -p 5433
```

```
psql (9.2.18, server 9.6.2)
```

```
WARNING: psql version 9.2, server version 9.6.
```

Some psql features might not work.

- ◆ Удобнее работать с клиентом той же версии

```
/usr/pgpro-9.6/bin/psql -U postgres
```

Соединение с БД (2/3)

◆ Аутентификация пользователя

- См файл `/var/lib/pgpro/9.6/data/pg_hba.conf`

```
sudo -u postgres vim  
/var/lib/pgpro/9.6/data/pg_hba.conf
```

Добавим строчку:

```
host all all ::1/128 trust
```

```
sudo systemctl reload postgrespro-9.6.service
```

Соединение с БД (3/3)

- ◆ Можно идти по сети на порт 5433 (надо расковырять FW)
- ◆ Можно сделать SSH-тоннель

```
ssh azureuser@myvm-dnsname-  
xx.northeurope.cloudapp.azure.com \  
-L5000:localhost:5433
```

- ◆ Тогда локальный порт 5000 пробросится на Ваш сервер

```
psql -h localhost -p 5000 -U postgres
```

Выполняем SQL

◆ Простейший SQL

```
SELECT 1;
```

```
  ?column?
```

```
-----
```

```
         1
```

```
(1 row)
```

Демо-база

◆ Демо-база «авиаперевозки»

<https://postgrespro.ru/education/demodb>

```
wget https://edu.postgrespro.ru/demo_small.zip
```

```
sudo yum install unzip
```

```
unzip demo_small.zip
```

```
sudo -u postgres /usr/pgpro-9.6/bin/psql -p 5433 <  
demo_small.sql
```

```
sudo -u postgres /usr/pgpro-9.6/bin/psql -p 5433 -d demo
```

Учебник для начинающих

◆ Рекомендуем скачать на будущее

<https://postgrespro.ru/education/introbook>

Начинаем работать с тестовой БД

◆ Список таблиц

```
set search_path to "$user",bookings, public;
```

```
\d
```

```
List of relations
```

Schema	Name	Type	Owner
bookings	aircrafts	table	postgres
bookings	airports	table	postgres

```
.....
```

Начинаем работать с тестовой БД

◆ Поля одной таблицы

```
\d aircrafts
```

```
Table "bookings.aircrafts"
```

Column	Type	Modifiers
aircraft_code	character(3)	not null
model	text	not null
range	integer	not null

```
Indexes:
```

```
"aircrafts_pkey" PRIMARY KEY, btree (aircraft_code)
```

```
.....
```

Простейшие запросы

◆ Просто SELECT

```
SELECT * FROM aircrafts WHERE aircraft_code =
  '773';
```

◆ Стандартный LIKE

```
SELECT * FROM aircrafts WHERE model LIKE
  'Airbus%';
```

◆ Регулярное выражение POSIX

```
SELECT * FROM aircrafts WHERE model ~ '^ (A|Boe) ';
```

Запросы с группировкой

```
SELECT arrival_city, count( * )  
FROM routes  
WHERE departure_city = 'Москва'  
GROUP BY arrival_city  
ORDER BY count DESC;
```

Использование массивов (1/2)

```
SELECT ARRAY[1,2];
```

```
SELECT days_of_week FROM routes;
```

```
days_of_week
```

```
-----
```

```
{1}
```

```
{2,3}
```

Использование массивов (2/2)

- ◆ Откуда не летают во вторник ?

```
SELECT DISTINCT departure_city
FROM routes
WHERE NOT days_of_week && ARRAY[2];
```

Использование JSON (1/3)

◆ Хобби пилотов

```
CREATE TABLE pilots (last_name name, hobbies
  jsonb);
```

```
INSERT INTO pilots VALUES ('Ivanov', '{"trips":
  3, "sports": ["футбол", "плавание"],
  "home_lib": true}');
```

```
INSERT INTO pilots VALUES ('Petrov', '{"trips":
  0, "sports": ["футбол", "плавание", "теннис"],
  "home_lib": true}');
```

Использование JSON (2/3)

◆ Оператор извлечения по ключу

```
SELECT hobbies->'sports' FROM pilots;
```

-> извлекает JSON

->> извлекает текстовое значение

Использование JSON (3/3)

◆ Кто любит футбол?

```
SELECT * FROM pilots
WHERE hobbies @> '{ "sports": [ "футбол" ]
  }'::jsonb;
```

◆ Полюбил хоккей

```
UPDATE pilots
SET hobbies = hobbies || '{ "sports": [ "хоккей"
  ] }'
WHERE last_name = 'Petrov';
```

Рекурсивные запросы (1/2)

◆ Посчитаем факториал с помощью CTE

```

WITH RECURSIVE t(n, factorial) AS (
    VALUES (0, 1)
    UNION ALL
    SELECT t.n+1, t.factorial*(t.n+1) FROM t WHERE
    t.n < 5
)
SELECT * FROM t;
  
```

Рекурсивные запросы (2/2)

◆ Все маршруты из Усть-Кута в Нерюнгри

```

WITH RECURSIVE p(last_arrival, destination, hops, flights, found) AS (
  SELECT a_from.airport_code, a_to.airport_code, ARRAY[a_from.airport_code], ARRAY[]::char(6)[],
         a_from.airport_code = a_to.airport_code
  FROM airports a_from, airports a_to
  WHERE a_from.airport_code = 'UKX'
  AND a_to.airport_code = 'CNN'
  UNION ALL
  SELECT r.arrival_airport, p.destination, (p.hops || r.arrival_airport)::char(3)[], (p.flights || r.flight_no)::char(6)[],
         bool_or(r.arrival_airport = p.destination) OVER ()
  FROM routes r, p
  WHERE r.departure_airport = p.last_arrival
  AND NOT r.arrival_airport = ANY(p.hops)
  AND NOT p.found
)
SELECT hops, flights
FROM p WHERE p.last_arrival = p.destination;

```

Документация

- ◆ Основная документация

<https://postgrespro.ru/docs/>

- ◆ Книги

<https://postgrespro.ru/education/introbook>

Postgres Professional

<http://postgrespro.ru/>

+7 495 150 06 91

info@postgrespro.ru

The background is a collage of hexagonal tiles in various shades of blue, orange, and grey. Some tiles contain abstract patterns like splatters, silhouettes of buildings, wavy lines, and polka dots.

postgrespro.ru