

Руководство по разворачиванию бота для MS Teams

Оглавление

Введение	3
1. Развертывание веб приложения для бота MS Teams	3
1.1. Установка среды выполнения .Net	3
1.2. Создание каталога для приложения.....	3
1.3. Автозапуск приложения.....	4
1.4. Установка Nginx.....	5
1.5. Конфигурирование Nginx.....	5
1.6. Получение сертификатов.....	7
2. Создание бота Teams	11
2.1. Создание бота.....	11
2.2. Связывание приложения Teams и веб приложения	21
2.3. Добавление приложения пользователем.....	23
3. Настройка Teams Addon	25

Введение

Бот для Microsoft Teams реализован в виде .Net Core Web API приложения. Одно из условий работы с платформой MS Teams – использование протокола https. Для работы по данному протоколу необходимо наличие сертификата.

1. Развертывание веб приложения для бота MS Teams

Инструкция по развертыванию веб приложения, а также получение бесплатного сертификата Let's Encrypt, будет описана для ОС Ubuntu 20.10 с использованием Nginx в качестве прокси сервера. Приложение также может быть развернуто на любом другом веб сервере.

1.1. Установка среды выполнения .Net

Перед установкой .NET выполните приведенные ниже команды, чтобы добавить ключ подписывания пакета Майкрософт в список доверенных ключей и добавить репозиторий пакетов.

```
sudo wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
```

```
sudo dpkg -i packages-microsoft-prod.deb
```

Установка среды выполнения

```
sudo apt-get update
```

```
sudo apt-get install -y apt-transport-https
```

```
sudo apt-get update
```

```
sudo apt-get install -y aspnetcore-runtime-3.1
```

1.2. Создание каталога для приложения

```
sudo mkdir /var/www
```

```
sudo mkdir /var/www/teamsbot
```

Скопируйте содержимое архива Qwerta.Pilot.Teams.Bot.zip в созданный каталог

1.3. Автозапуск приложения

Для автоматического запуска приложения необходимо создать юнит system.

Создаем и открываем для редактирования файл

```
sudo nano /etc/systemd/system/teamsbot.service
```

Содержимое файла:

```
[Unit]
Description=Pilot ICE Teams Bot

[Service]
WorkingDirectory=/var/www/teamsbot
ExecStart=/usr/bin/dotnet /var/www/teamsbot/Qwerta.Pilot.Teams.Bot.dll
Restart=always
RestartSec=10
SyslogIdentifier=teamsbot
User=root
Environment=ASPNETCORE_ENVIRONMENT=Production

[Install]
WantedBy=multi-user.target
```

После сохранения необходимо включить и запустить сервис при помощи следующих команд:

```
sudo systemctl enable teamsbot.service
```

```
sudo systemctl start teamsbot.service
```

Проверим статус сервиса:

```
sudo systemctl status teamsbot.service
```

1.4. Установка Nginx

В качестве обратного прокси сервере для обработки https запросов будем использовать Nginx

Для установки Nginx выполните команду:

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

Программа установки самостоятельно пропишет сценарии автозапуска systemd.

Так как Nginx устанавливается впервые, для запуска сервиса необходимо выполнить команду:

```
sudo service nginx start
```

1.5. Конфигурирование Nginx

В качестве адреса хоста в инструкции будет рассмотрен адрес example.ru. Вместо этого адреса вам необходимо указать свой адрес хоста.

Создайте файл конфигурации Nginx для вашего хоста:

```
sudo nano /etc/nginx/conf.d/example.ru.conf
```

Содержимое файла:

```
#server {
# listen 443 ssl http2;
# server_name example.ru;
# include SSL/example.ru.SSL.inc;
# include conf.d/example.ru.inc;
#}
server {
listen 80;
server_name example.ru;
location / {
include conf.d/example.ru.inc;
}
location ~ "^/\.well-known/acme-challenge/([-_a-zA-Z0-9]+)$" {
default_type text/plain;
return 200 "$1.[account key thumbprint]";
}
}
```

Содержимое файла не окончательное. После получение отпечатка ключа и самих ключей, содержимое файла будет изменено.

```
sudo nano /etc/nginx/conf.d/ example.ru.inc
```

Содержимое файла:

```
location / {  
    proxy_pass http://localhost:5000;  
    proxy_http_version 1.1;  
    proxy_set_header Connection keep-alive;  
    proxy_set_header Host $host;  
    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 $scheme;  
    proxy_pass_header Sec-WebSocket-Extensions;  
}
```

Создайте каталог для хранения сертификатов:

```
sudo mkdir /etc/nginx/SSL
```

Создайте дополнение конфигурации:

```
sudo nano /etc/nginx/conf.d/ SSL/example.ru.SSL.inc
```

Содержимое файла:

```
ssl_certificate SSL/example.ru.fullchain.pem;  
ssl_certificate_key SSL/example.ru.key.pem;  
ssl_certificate SSL/example.ru.ecc.fullchain.pem;  
ssl_certificate_key SSL/example.ru.ecc.key.pem;
```

1.6. Получение сертификатов

В качестве адреса хоста в инструкции будет рассмотрен адрес example.ru. Вместо этого адреса вам необходимо указать свой адрес хоста.

Необходимо установить клиент Let's Encrypt с <https://acme.sh/>

```
wget -O - https://get.acme.sh | sh
```

Далее необходимо выполнить шаги указанные на <https://github.com/acmesh-official/acme.sh/wiki/Stateless-Mode>, указать полученный account key thumbprint в блоке "`^\.well-known/acme-challenge/([_a-zA-Z0-9]+)$`" выше в конфиге веб-сервера.

```
acme.sh --register-account
```

В качестве результата команды будет получен результат вида

```
root@ed:~# acme.sh --register-account
[Mon Feb 6 21:40:18 CST 2017] Registering account
[Mon Feb 6 21:40:19 CST 2017] Already registered
[Mon Feb 6 21:40:21 CST 2017] Update success.
[Mon Feb 6 21:40:21 CST 2017]
ACCOUNT_THUMBPRINT='6fXAG9VyG0IahirPEU2ZerUtItW2DHZDzD9wZaEKpqd'
```

После этого необходимо внести полученный ACCOUNT_THUMBPRINT в файл конфигурации Nginx

```
sudo nano /etc/nginx/conf.d/example.ru.conf
```

```
#server {  
# listen 443 ssl http2;  
# server_name example.ru;  
# include SSL/example.ru.SSL.inc;  
# include conf.d/example.ru.inc;  
#}  
server {  
listen 80;  
server_name example.ru;  
location / {  
include conf.d/example.ru.inc;  
}  
location ~ "^/\.well-known/acme-challenge/([-_a-zA-Z0-9]+)$" {  
default_type text/plain;  
return 200 "$1.6fXAG9VyG0lahirPEU2ZerUtlW2DHZDzD9wZaEKpqd ";  
}  
}
```

После снесения изменений необходимо перезапустить Nginx для применения конфигурации:

```
sudo service nginx start
```

Для получения ключей выполните следующие команды:

```
acme.sh --issue --stateless -d example.ru
```

```
acme.sh --issue --stateless -d example.ru --keylength ec-384
```

для установки ключей выполните следующие команды:

```
acme.sh --install-cert -d example.ru --key-file /etc/nginx/SSL/example.ru.key.pem --cert-file  
/etc/nginx/SSL/example.ru.cert.pem --fullchain-file /etc/nginx/SSL/example.ru.fullchain.pem --  
reloadcmd "service nginx force-reload"
```

```
acme.sh --install-cert -d example.ru --ecc --key-file /etc/nginx/SSL/example.ru.ecc.key.pem --cert-file /etc/nginx/SSL/example.ru.ecc.cert.pem --fullchain-file /etc/nginx/SSL/example.ru.ecc.fullchain.pem --reloadcmd "service nginx force-reload"
```

После установки ключей, необходимо раскомментировать соответствующий раздел в конфигурации Nginx.

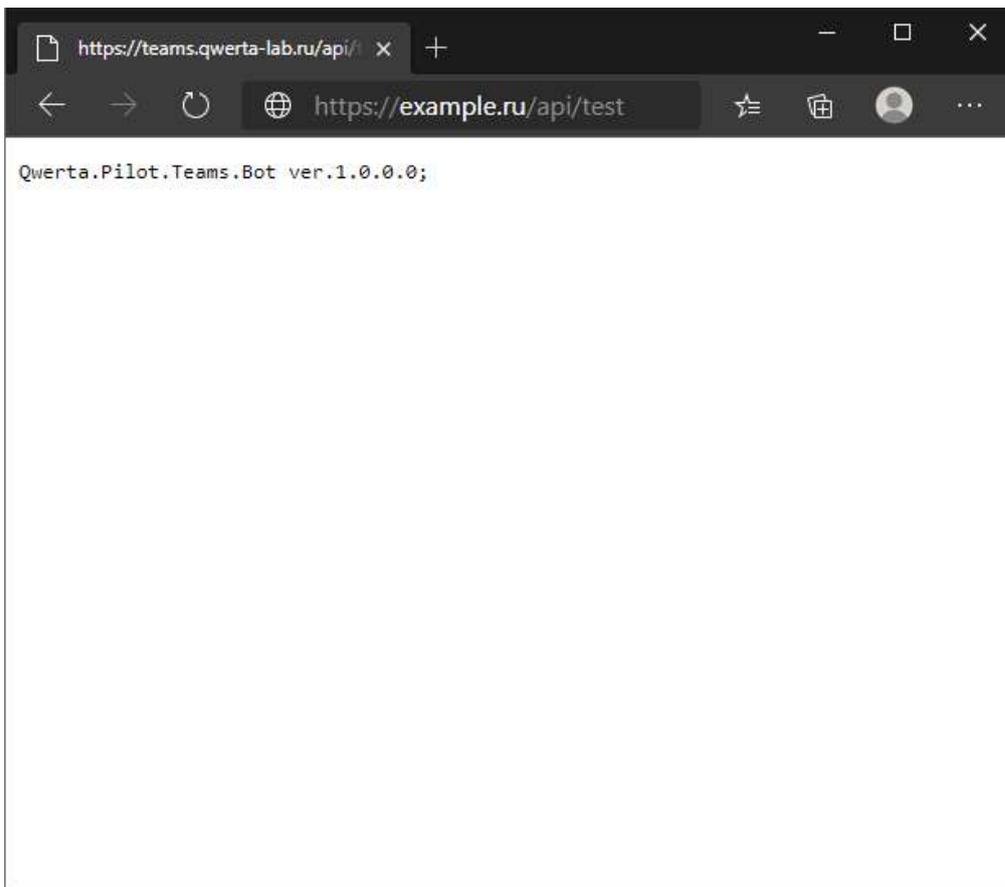
```
server {
    listen    443 ssl http2;
    server_name example.ru;
    include SSL/example.ru.SSL.inc;
    include conf.d/example.ru.inc;
}
server {
    listen    80;
    server_name example.ru;
    location / {
        include conf.d/example.ru.inc;
    }
    location ~ "^/\.well-known/acme-challenge/([-_a-zA-Z0-9]+)$" {
        default_type text/plain;
        return 200 "$1.6fXAG9VyG0IahirPEU2ZerUtlW2DHzDzD9wZaEKpqd ";
    }
}
```

После внесения изменений необходимо перезапустить Nginx для применения конфигурации

```
sudo service nginx start
```

Для проверки работоспособности развернутого приложения в адресной строке браузера введите адрес <https://example.ru/api/test>

В результате должна отобразиться информация о названии и версии модуля.

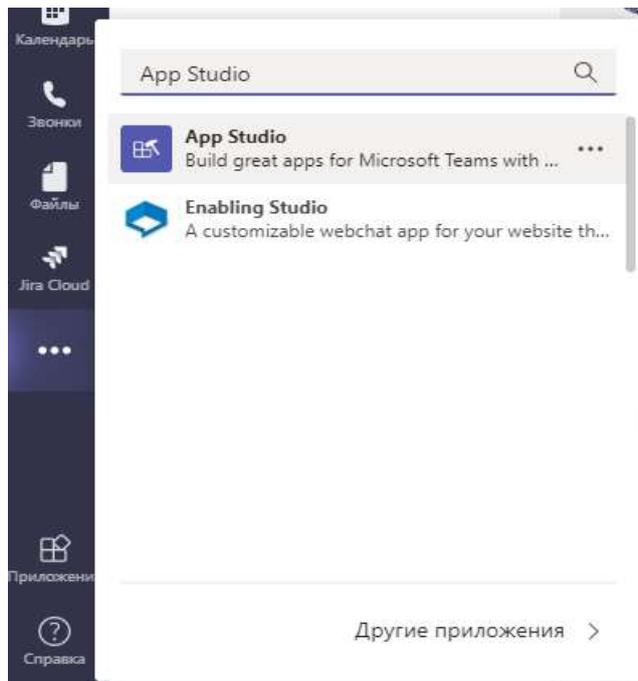


2. Создание бота Teams

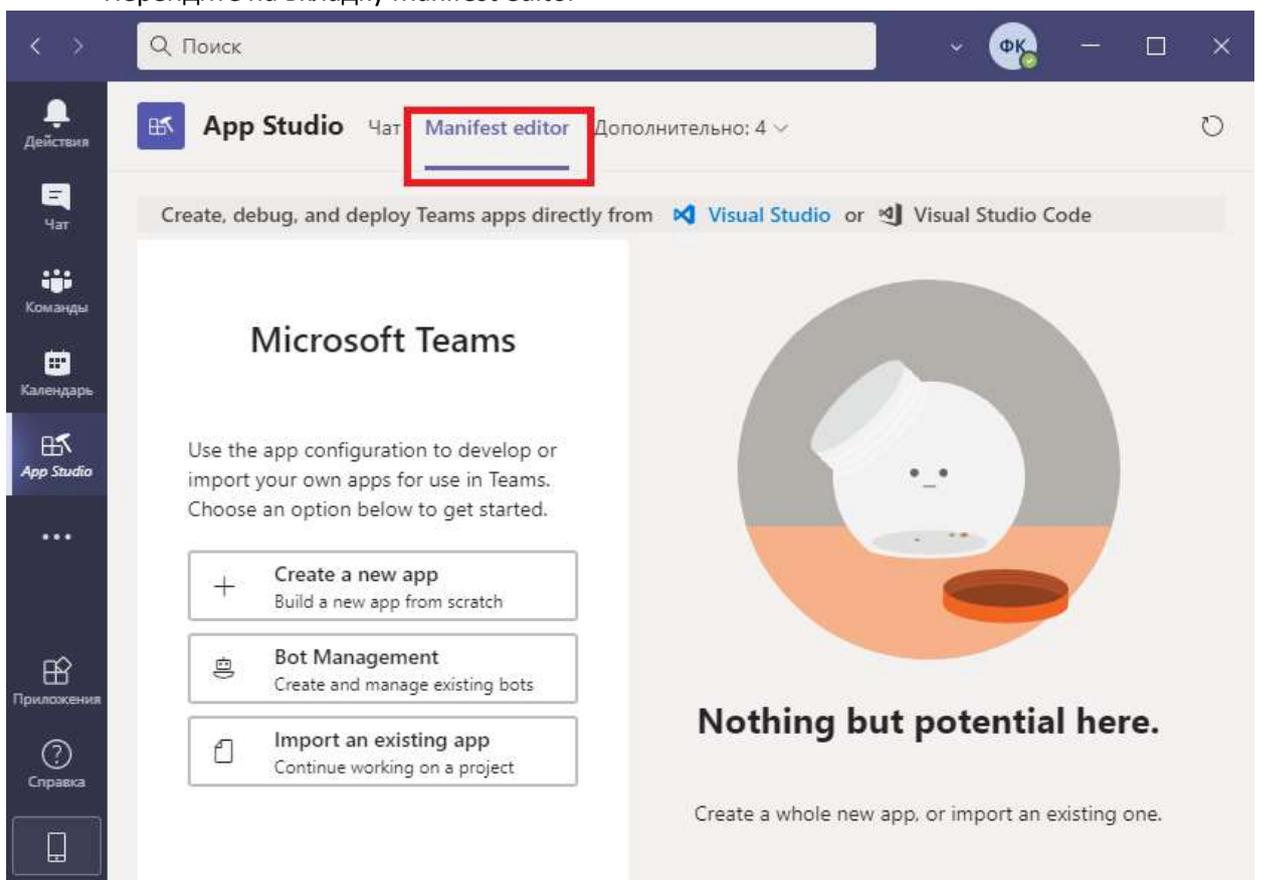
2.1. Создание бота

Создание бота будет происходить через клиентское приложение Microsoft Teams.

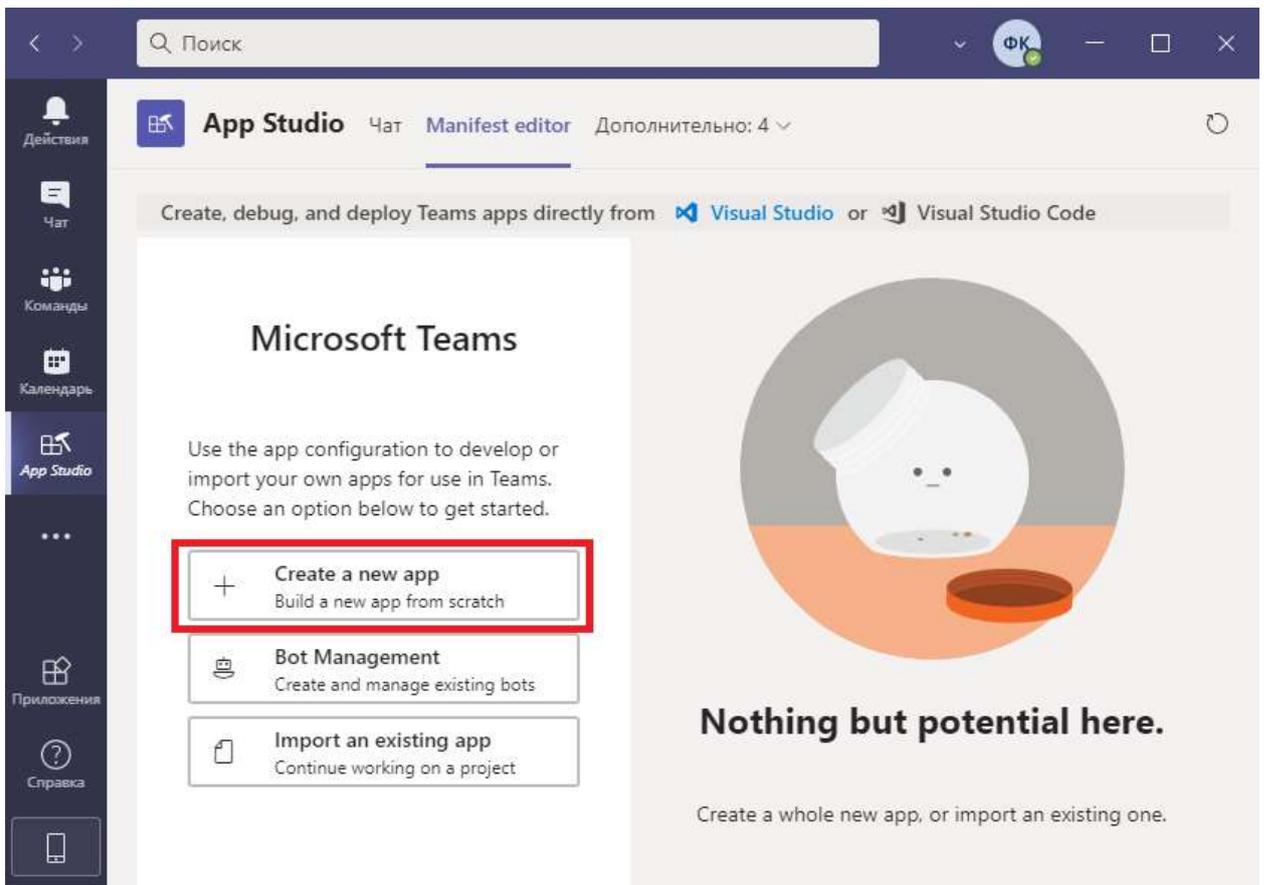
Необходимо запустить приложение App Studio из списка доступных приложений:



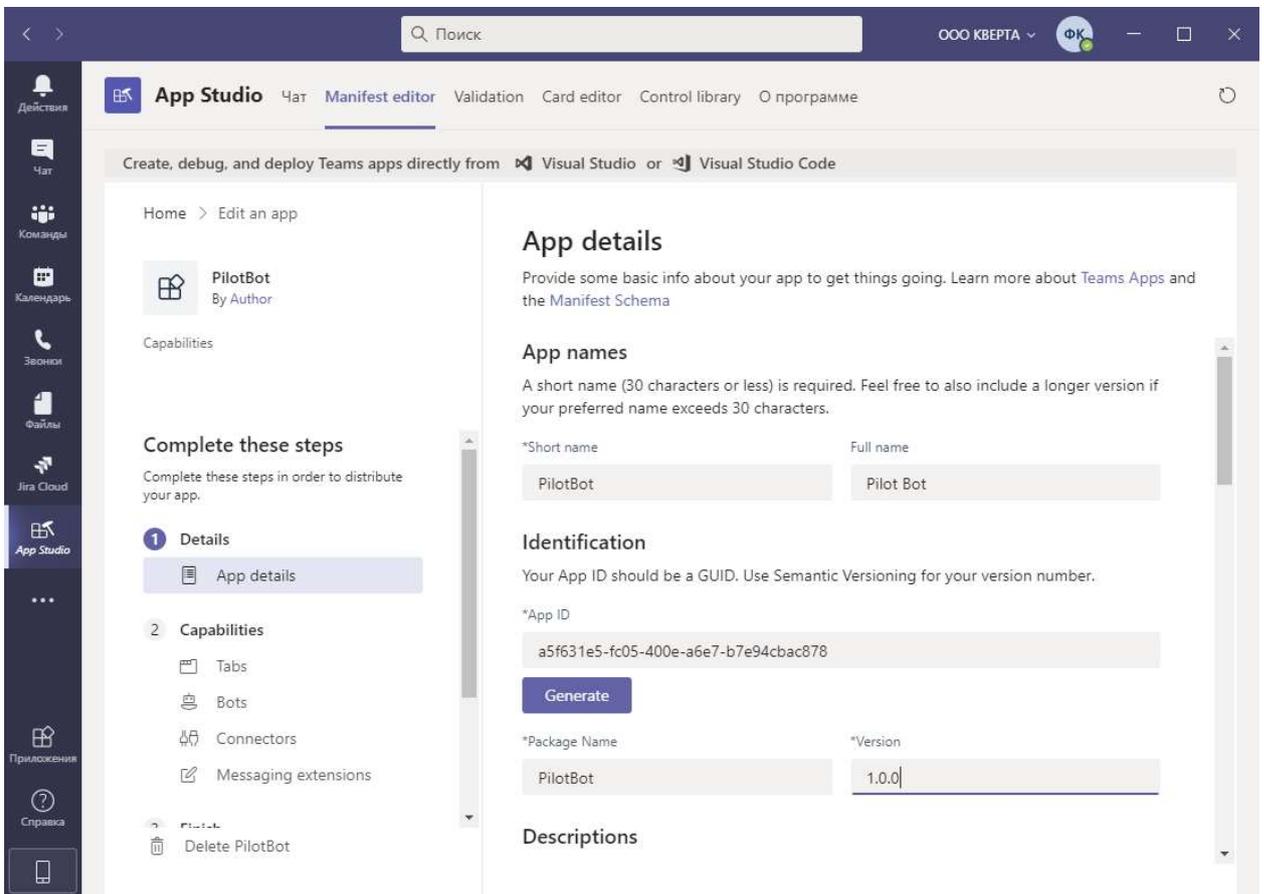
Перейдите на вкладку Manifest editor

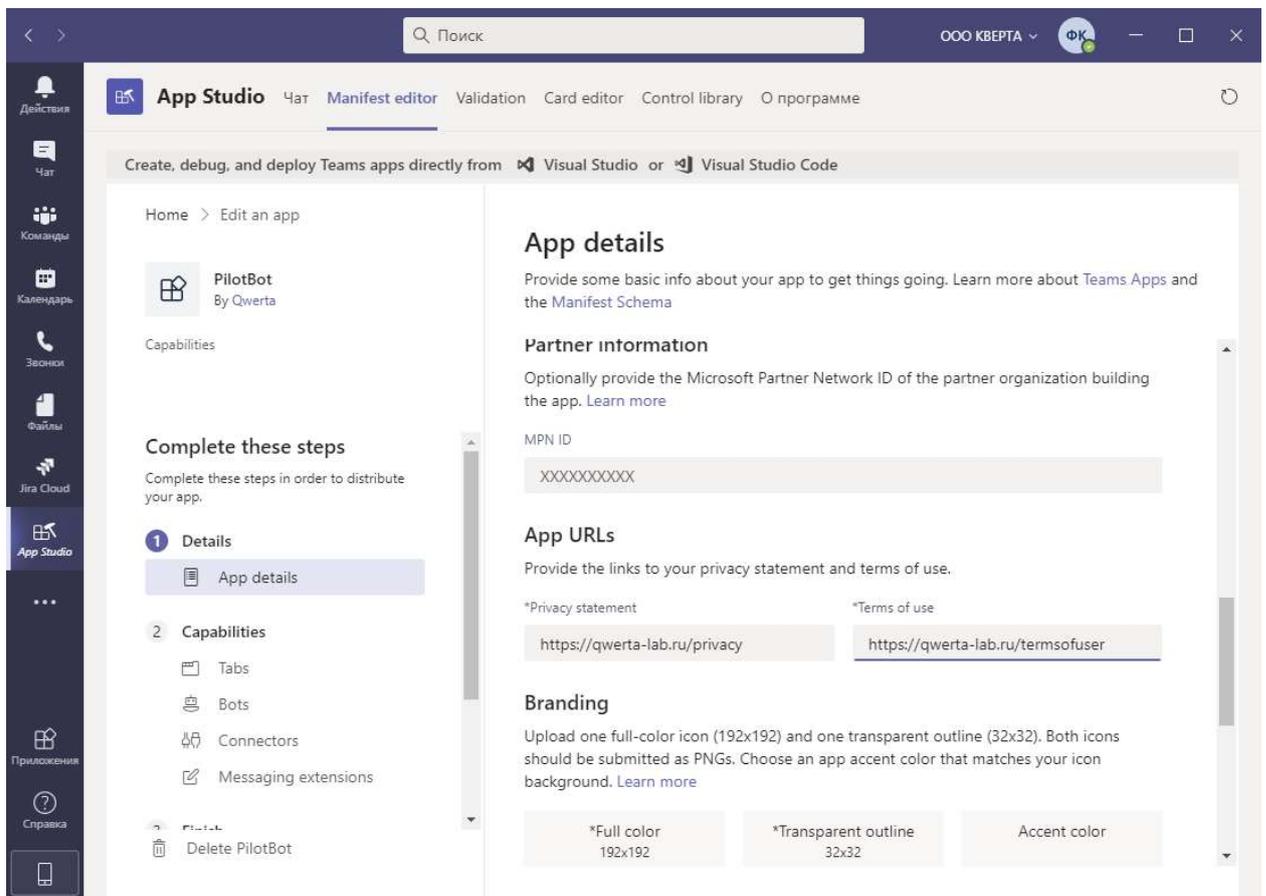
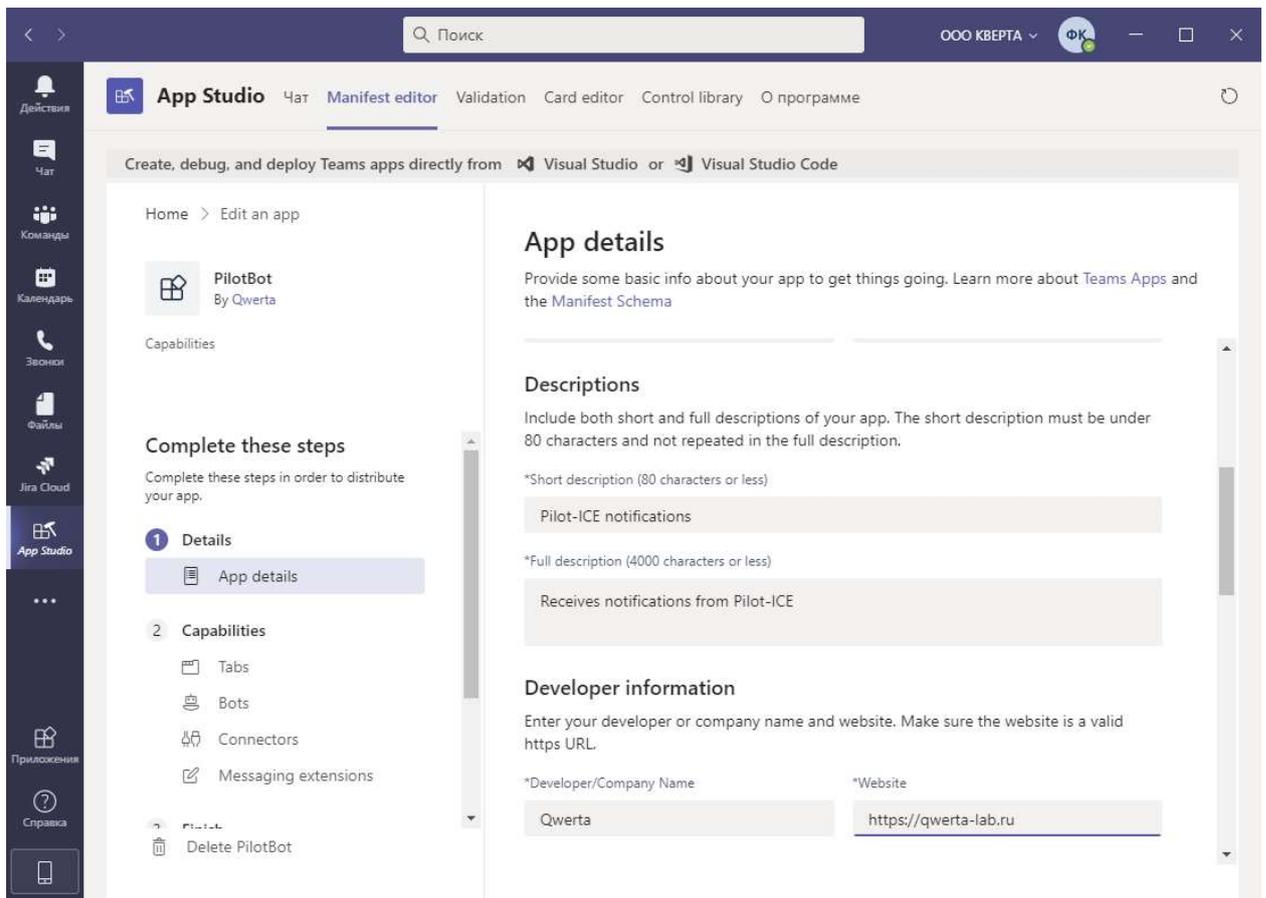


На вкладке Manifest editor, нажмите кнопку Create a new app

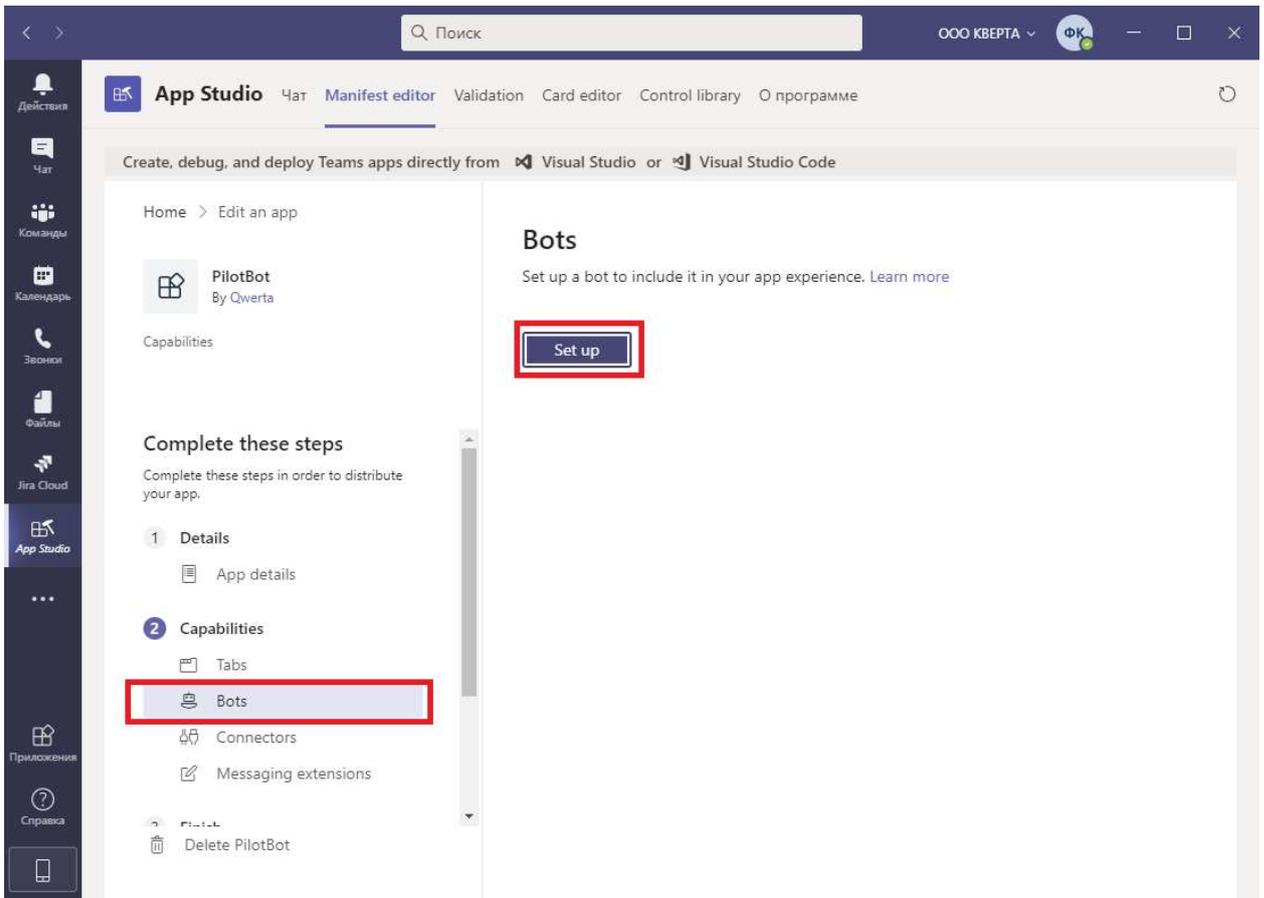


В разделе App details заполните поля в соответствии со скриншотами. Для заполнения поля App Id необходимо нажать кнопку Generate

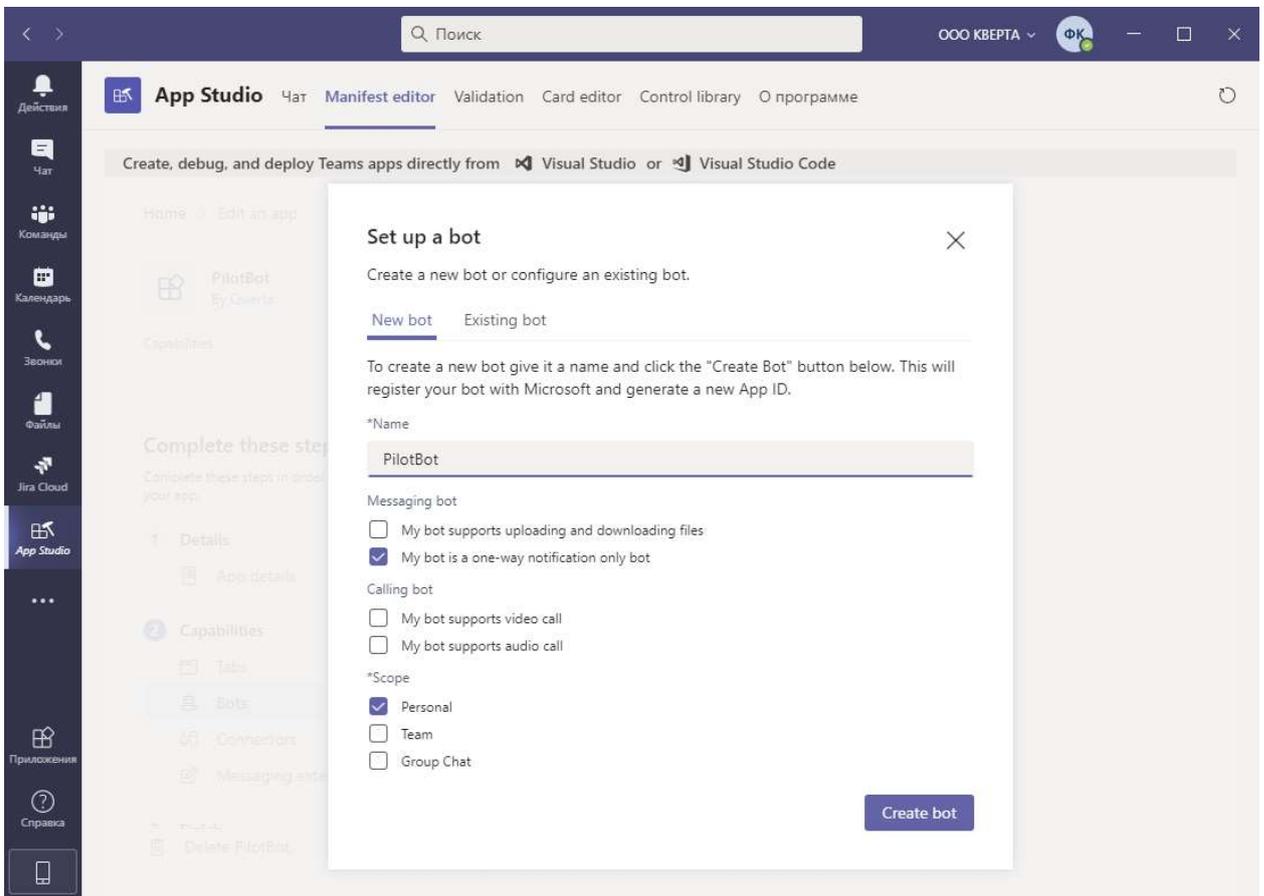




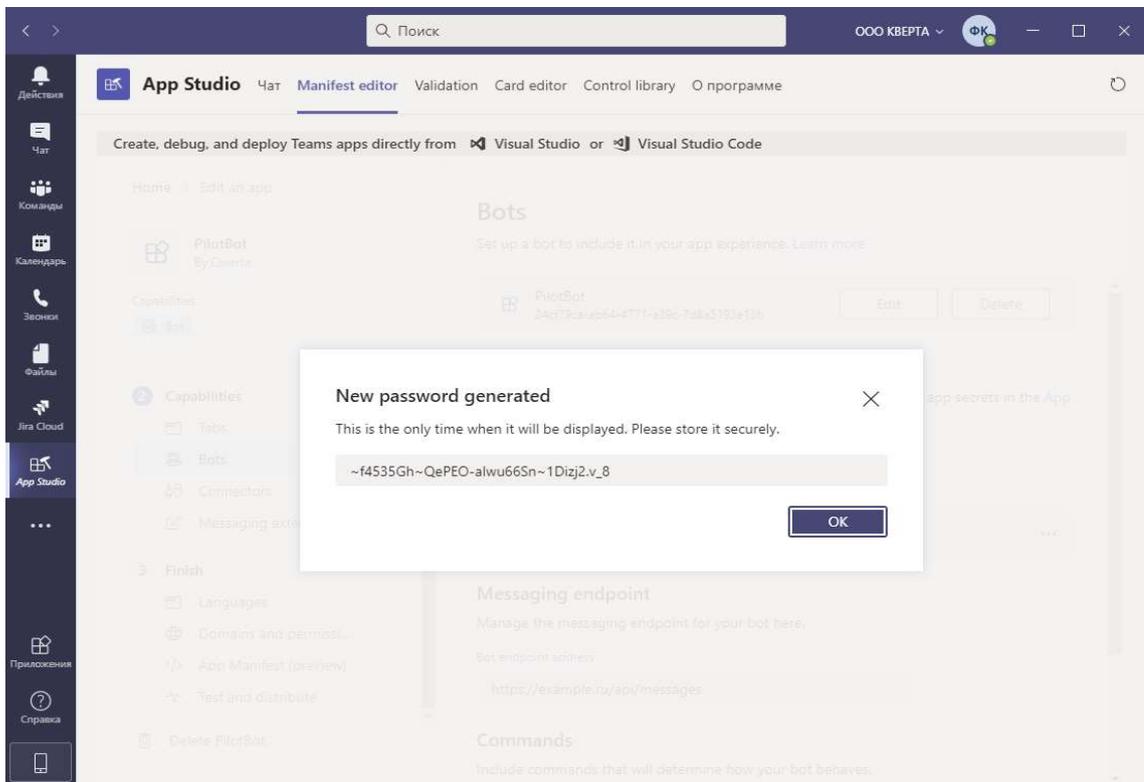
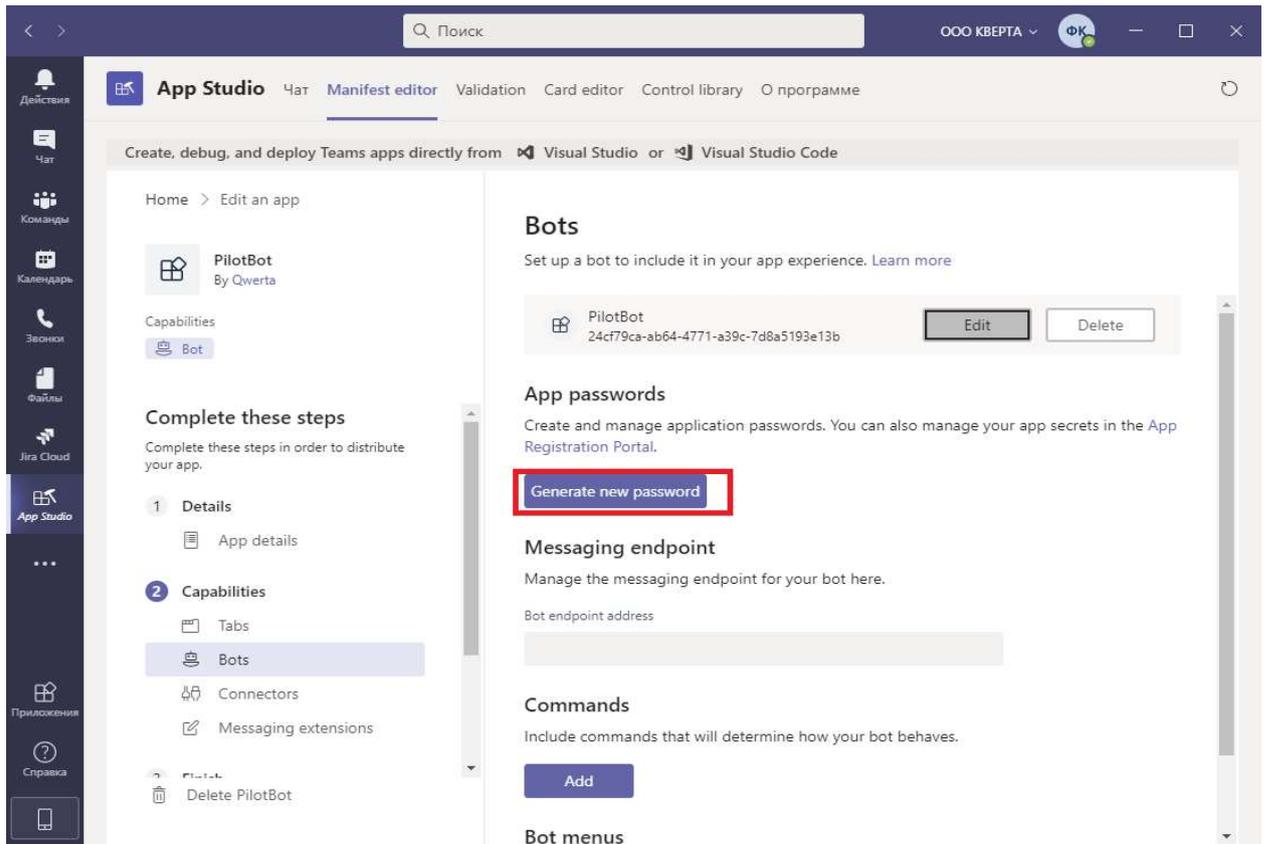
После заполнения обязательных полей, перейдите в раздел Bots и нажмите кнопку Set up



В появившемся окне, заполните поля согласно скриншоту:

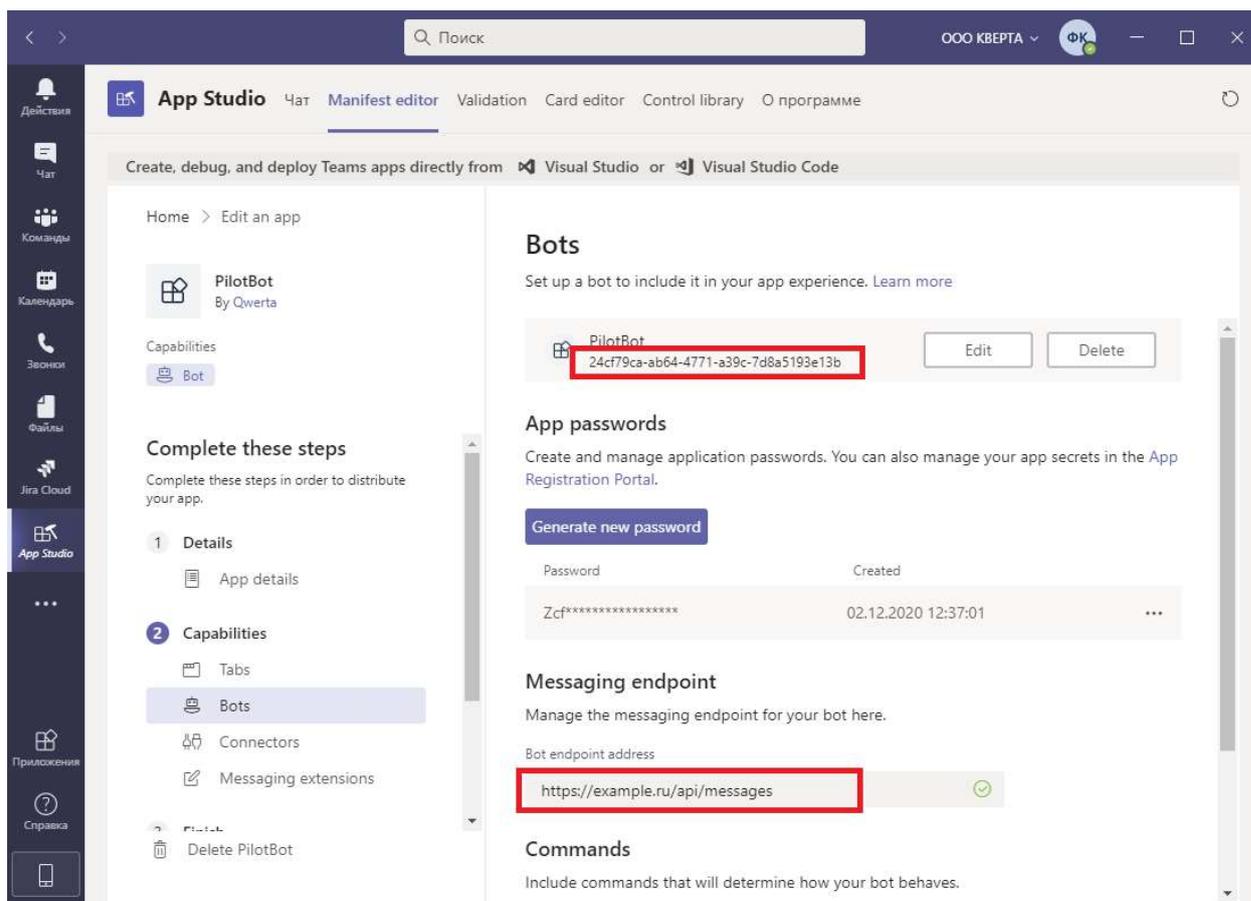


По завершению заполнения полей, нажмите кнопку Create bot. В открывшемся окне нажмите кнопку Generate new password

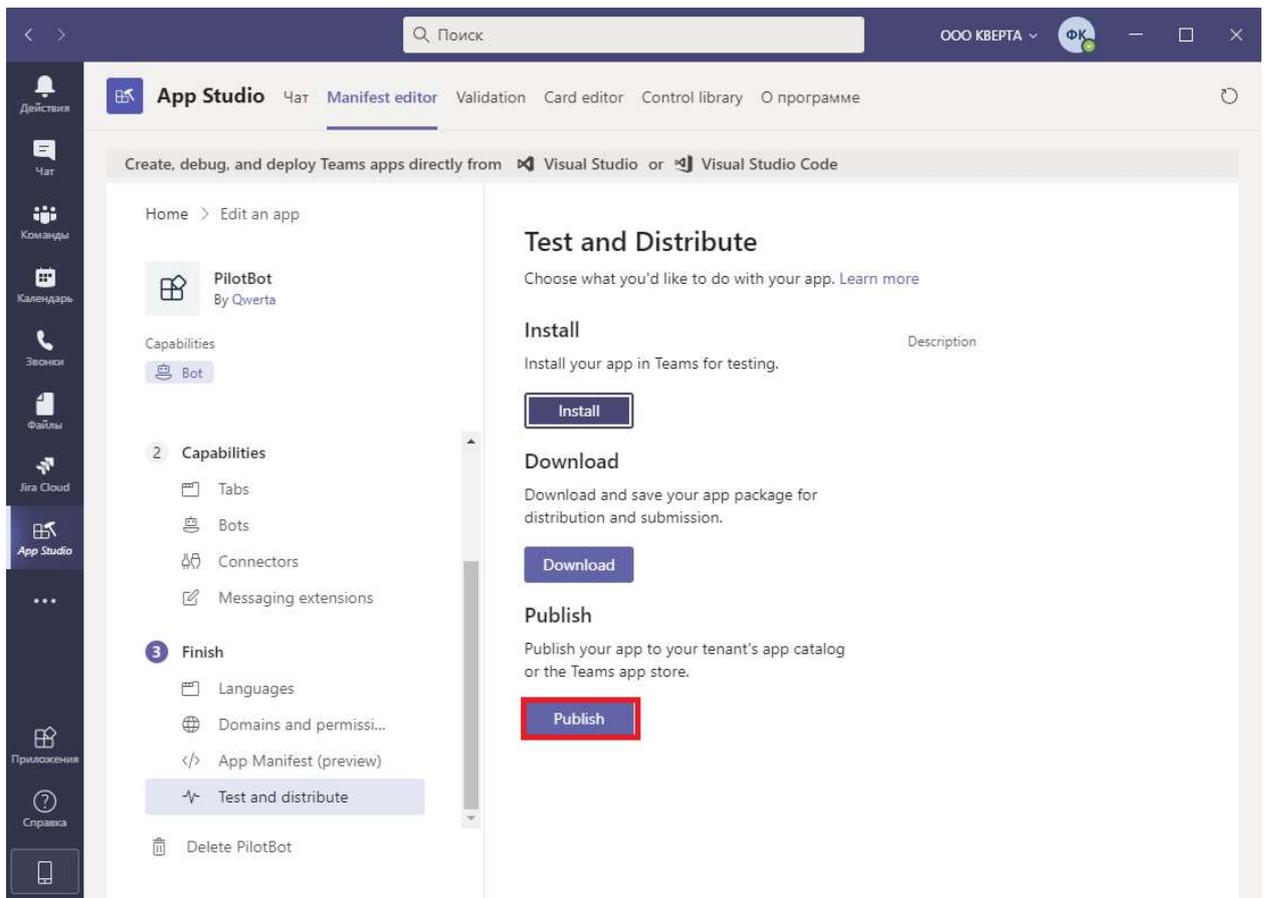


Сгенерированный пароль, а также идентификатор созданного бота нам далее понадобятся для связывания веб приложения и приложения Teams. В поле Bot endpoint address введите адрес развернутого ранее веб приложения в формате <https://example.ru/api/messages> и нажмите Enter.

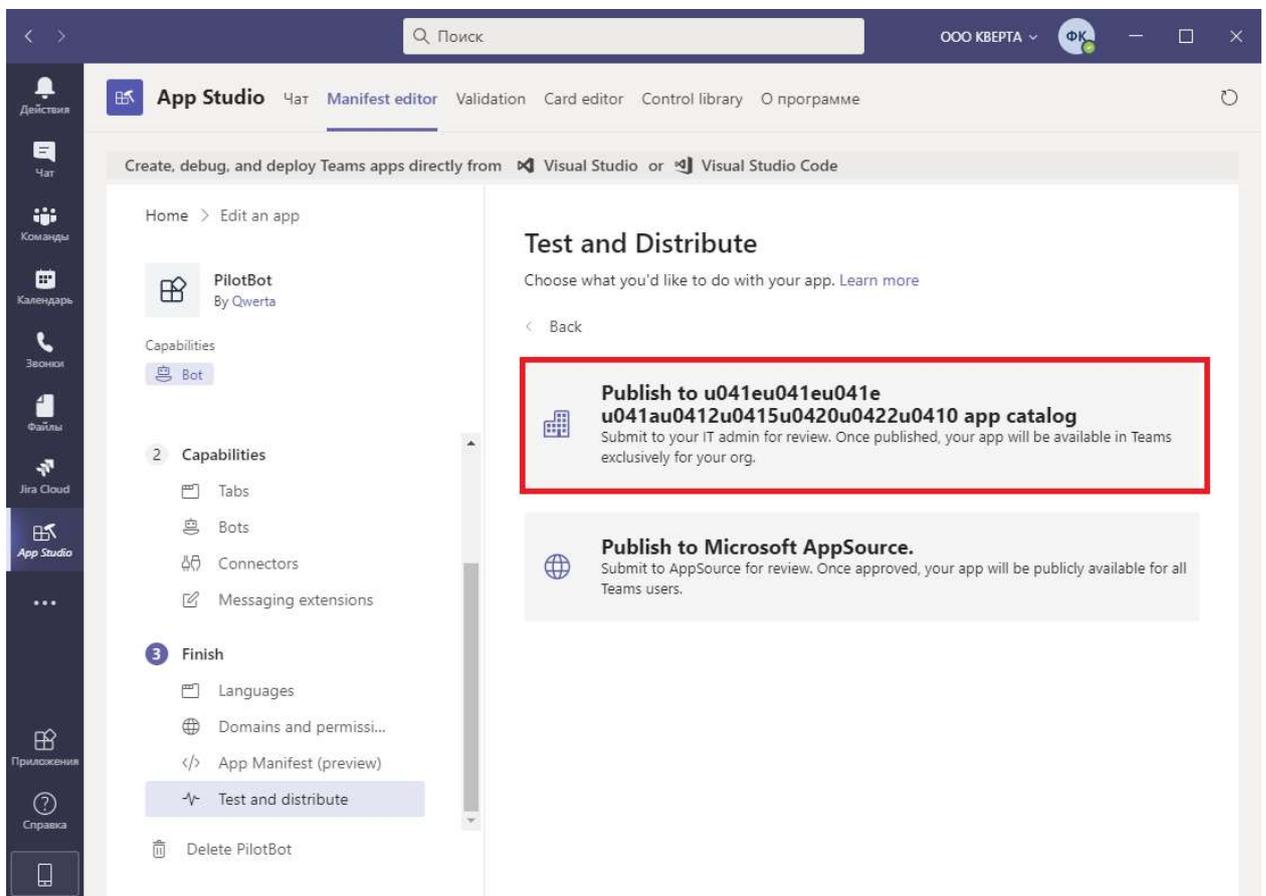
В качестве адреса хоста в инструкции рассмотрен адрес example.ru. Вместо этого адреса вам необходимо указать адрес своего хоста.



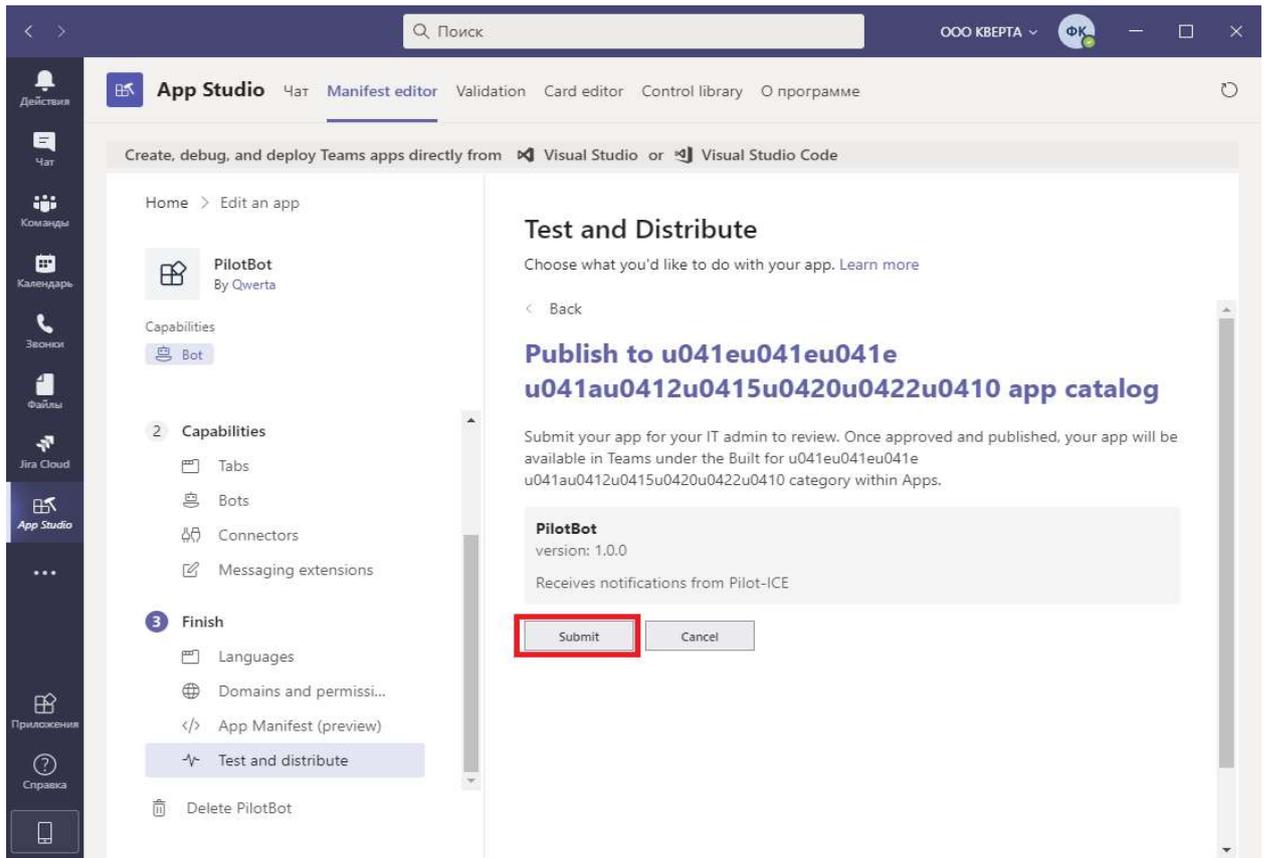
Перейдите на вкладку Test and distribute и нажмите кнопку Publish



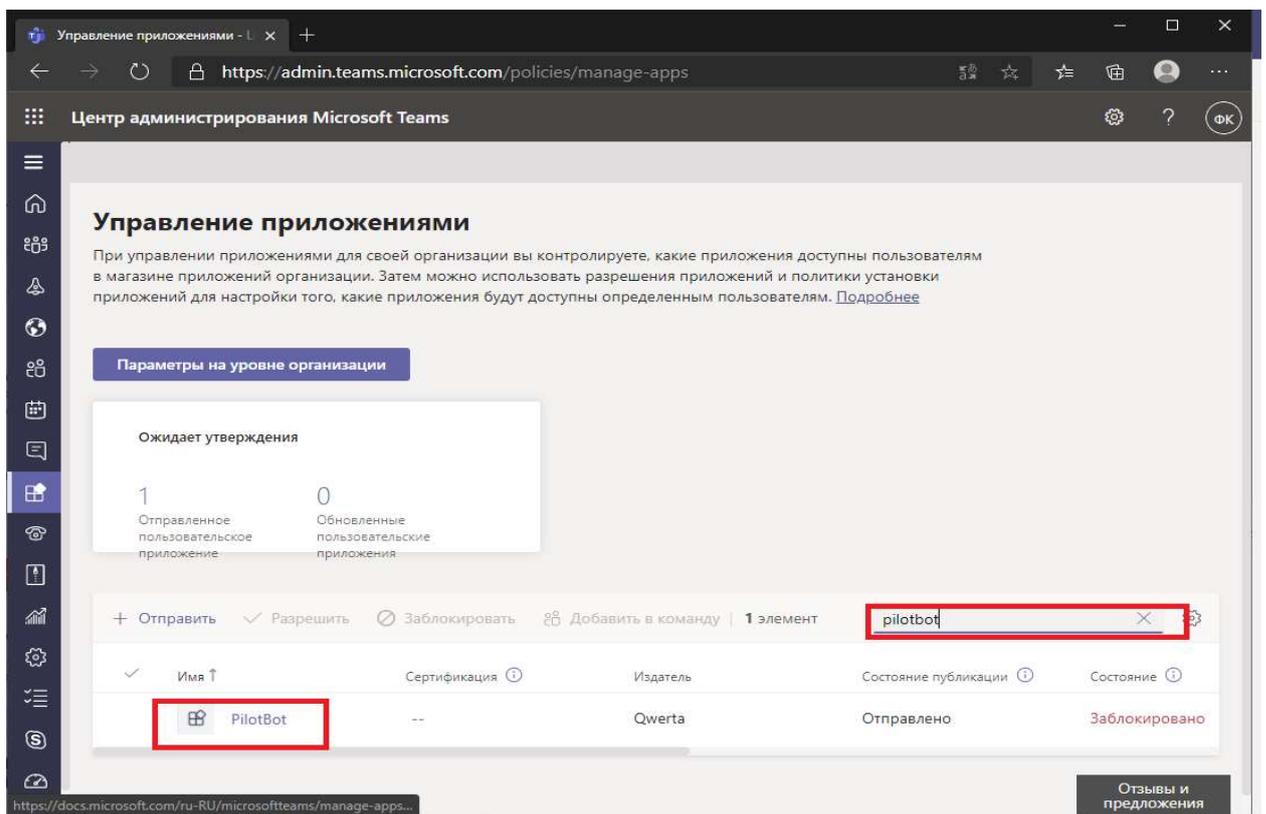
В появившемся окне нажмите кнопку публикации для вашей организации.



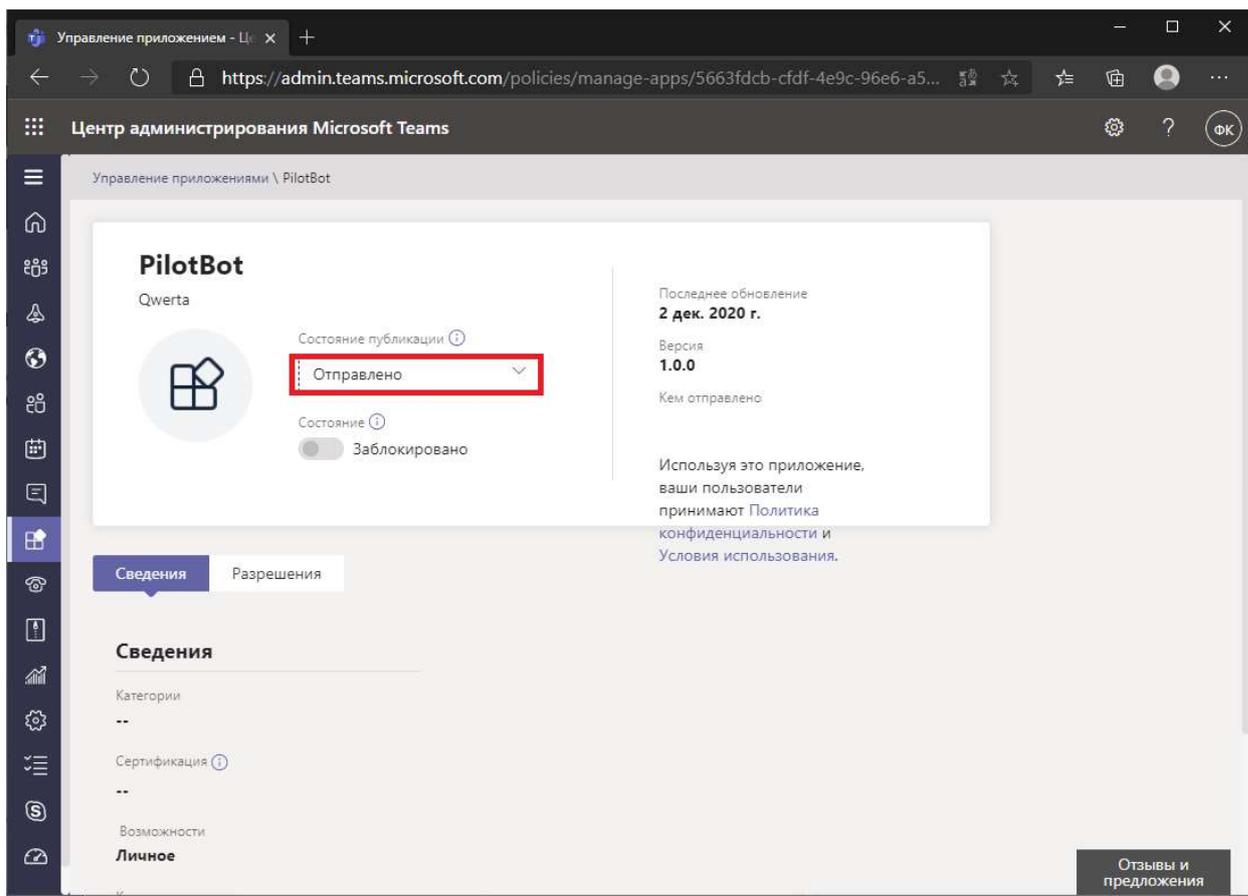
В появившемся окне нажмите кнопку Submit для отправки на рассмотрение приложения администратору Teams.



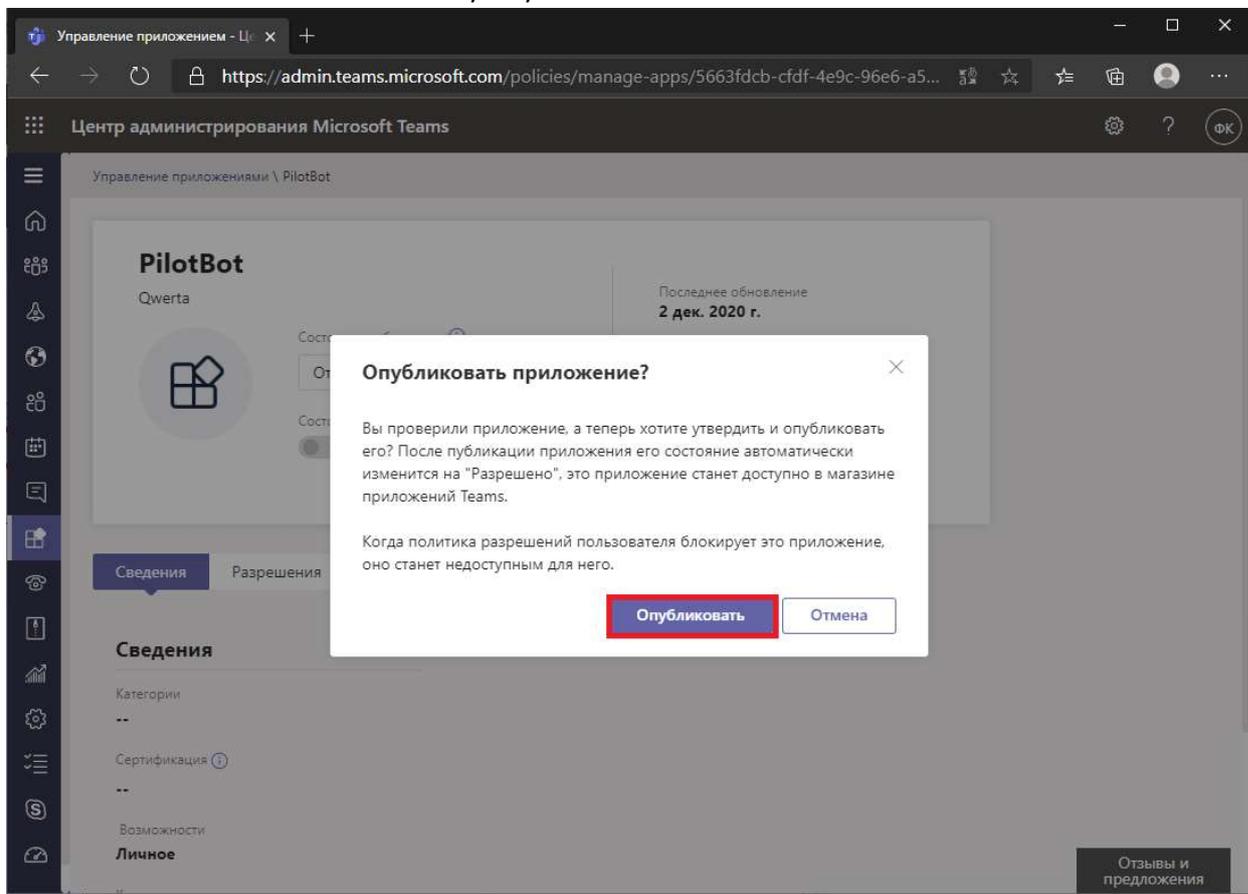
Зайдите по адресу <https://admin.teams.microsoft.com/policies/manage-apps> от учетной записи с правами администратора Teams. В поисковой строке введите название бота и нажмите на название в списке найденных приложений.

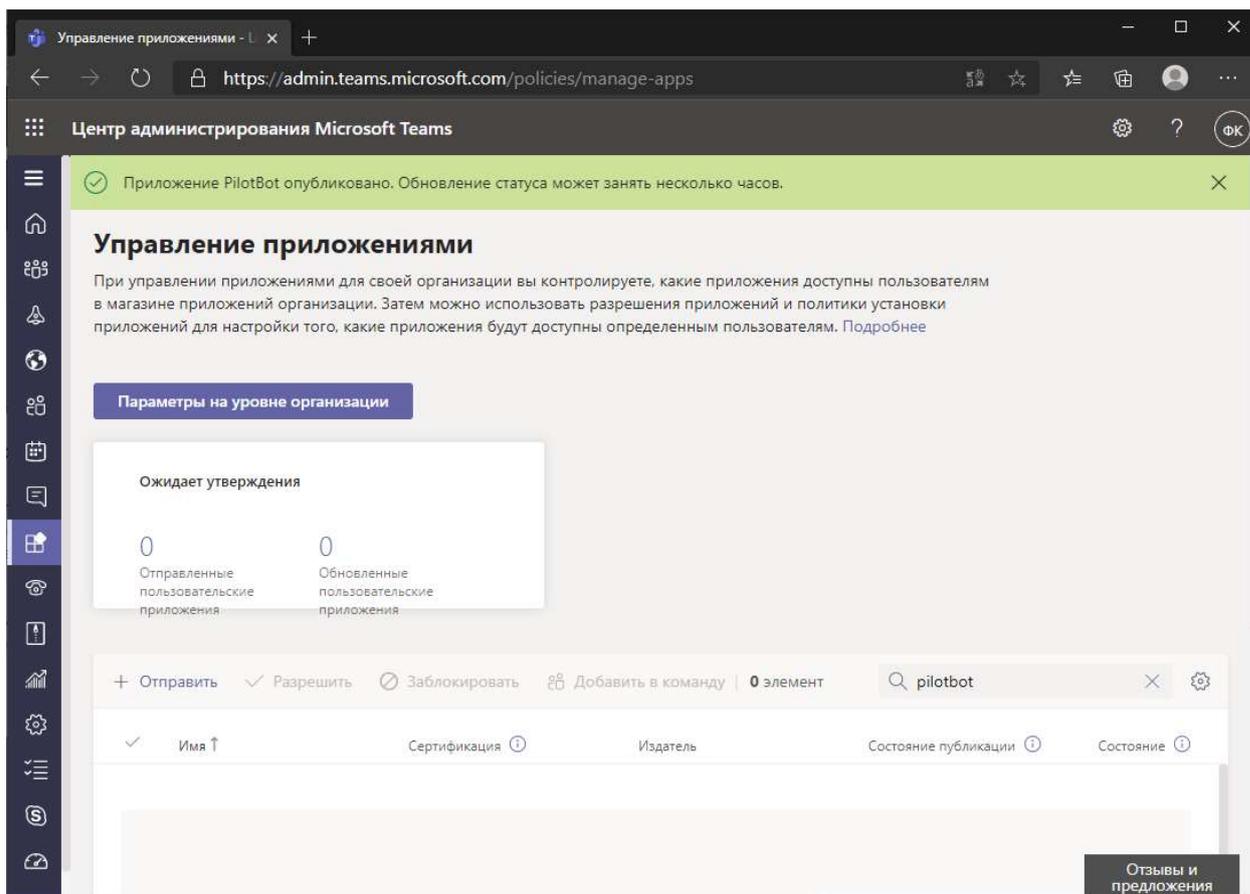


В выпадающем списке выберете пункт - Опубликовать



В появившемся окне нажмите кнопку Опубликовать





Приложение в списке доступных может появиться в течении нескольких часов.

2.2. Связывание приложения Teams и веб приложения

Для того чтобы связать развернутое ранее веб приложение с приложением Teams, необходимо внести идентификатор приложения и пароль в файл настроек appsettings.json веб приложения.

Выполните команду

```
sudo nano /var/www/teamsbot/appsettings.json
```

Заполните поля

MicrosoftAppId – идентификатор приложения

MicrosoftAppPassword – пароль для приложения

Пример содержания файла:

```
{
  "Logging": {
    "IncludeScopes": false,
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",

  "ConnectionStrings": {
    "DefaultConnection": "Data Source=BotConversations.db",
    "SqliteConnection": "Data Source=BotConversations.db"
  },

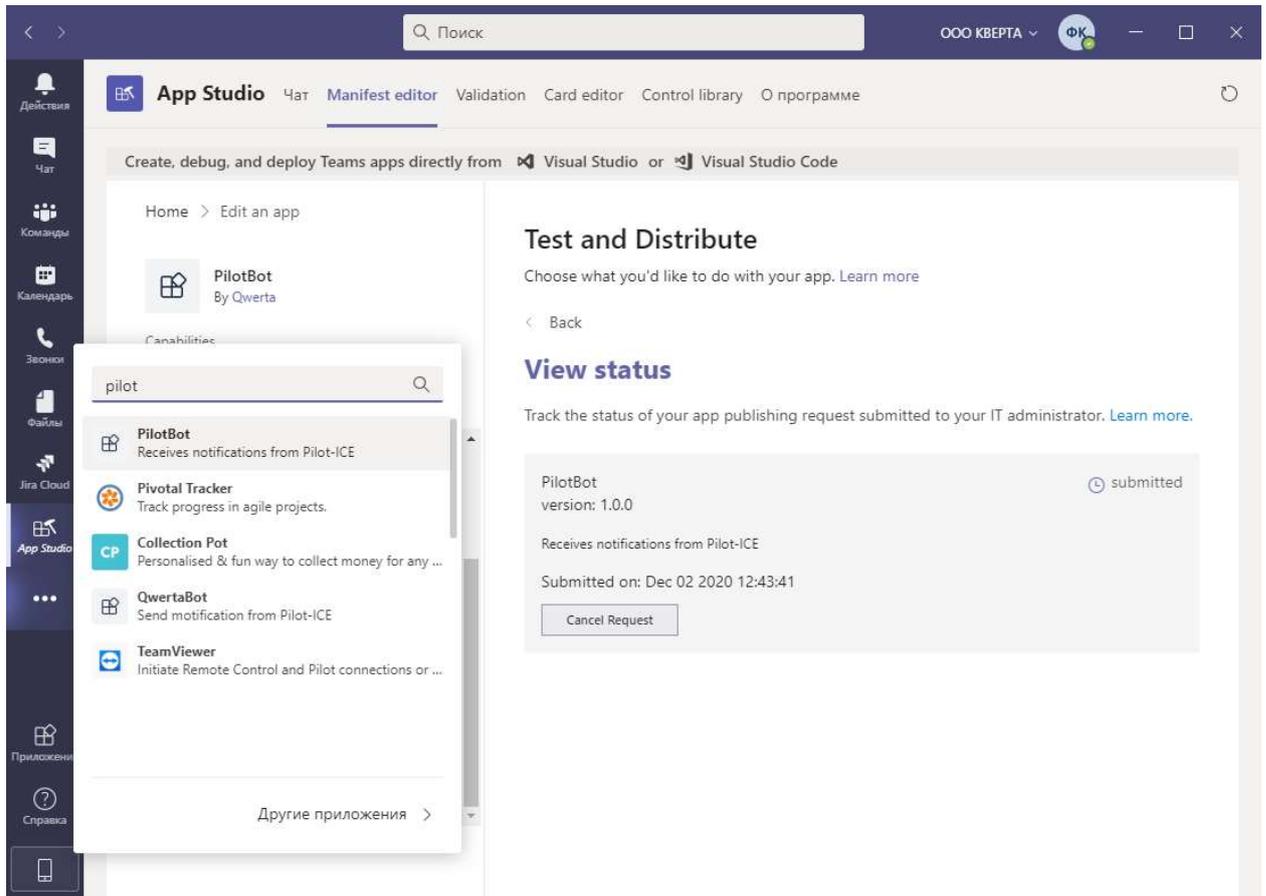
  "MicrosoftAppId": "24cf79ca-ab64-4771-a39c-7d8a5193e13b",
  "MicrosoftAppPassword": "~f4535Gh~QePEO-alwu66Sn~1Dizj2.v_8"
}
```

После сохранения настроек необходимо перезапустить сервис. Выполните команду:

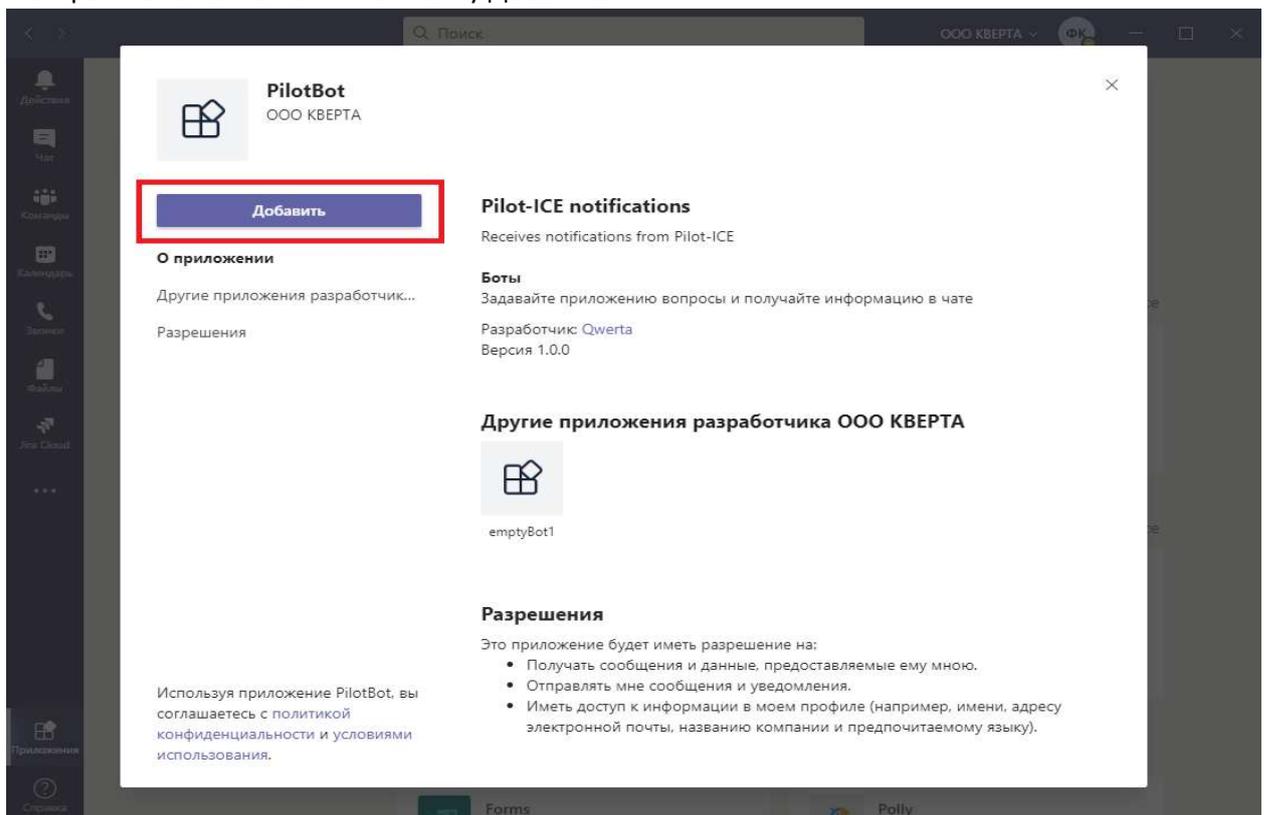
```
sudo service teamsbot restart
```

2.3. Добавление приложения пользователем

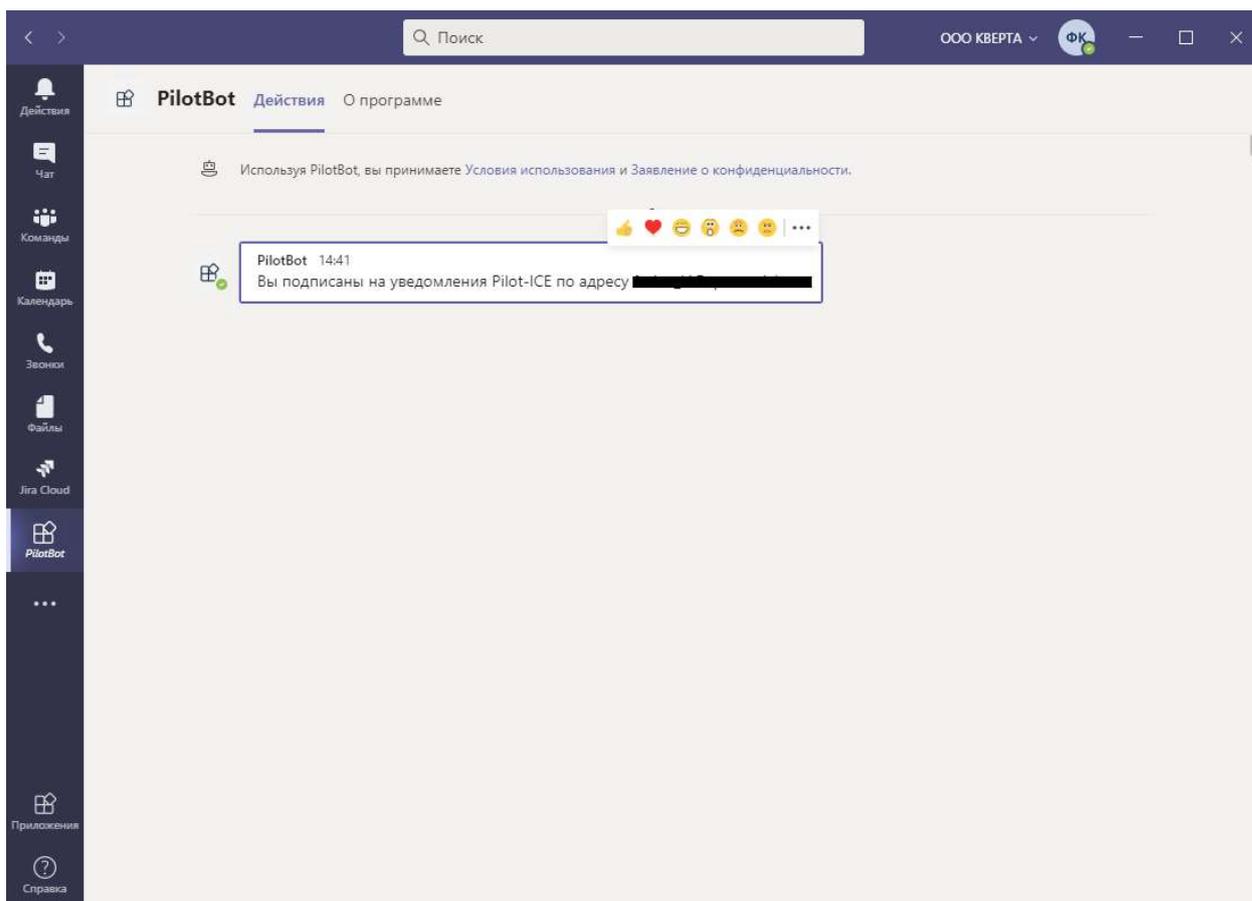
Найдите приложение в списке доступных по имени



В открывшемся окне нажмите кнопку **Добавить**



В случае успешной настройки связывания приложений, получите сообщение, что вы подписаны на уведомления Pilot-ICE с указанием адреса электронной почты



Указанный адрес электронной почты должен быть прописан в карточке пользователя в MyAdmin

✕

Редактирование пользователя

*** Имя пользователя**

Отображаемое имя пользователя

Почта

Номер телефона

*** Пароль**

*** Подтверждение пароля**

Администратор базы данных

Статус пользователя

Доступен Недоступен

3. Настройка Teams Addon

Для отправки уведомлений из Pilot-ICE, необходимо настроить расширение для работы Teams. Запустите файл `.\Extensions\Qwerta.Pilot.NotificationEx.Activity\Plugins\Teams Qwerta.Pilot.NotificationEx.Teams.Addon.Configuration.exe` с правами администратора из каталога базы данных. Укажите адрес развернутого веб приложения. При необходимости укажите параметры прокси сервера.

```
C:\ProgramData\ASCON\Pilot Server\Databases\pilot-ice_ru\Extensions\Qwerta.Pilot.NotificationEx.Activity\Plugins\Teams\Qwerta.Pilot.NotificationEx...  
Bot URL: https://example.ru  
Proxy required [y/n]: n  
Settings saved  
-
```