

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут» імені Ігоря Сікорського
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота № 6

З дисципліни: «Компоненти програмної інженерії – 2.
Моделювання та аналіз програмного забезпечення»

Виконав:

студент групи ІТ-73

Старовойтов Руслан Олександрович

Перевірив:

Галушко Д. О

Київ 2020

Тема: Разработка конструктора Telegram-ботов.

Цель: настройка операционной системы и установление пакетов для .net

Поскольку собственных серверов нет, то приложение будет запускаться в облаке. В качестве облака рассматривались два кандидата Amazon Web Services и Microsoft Azure. Оба имеют бесплатный тариф для студентов, но AWS не позволяет выбрать регион для запуска виртуальной машины, поэтому выбор пал на Azure.

В качестве os выбрана Ubuntu 18.04 LTS так раньше был опыт работы с ней и никаких серьёзных проблем не возникало.

Создадим виртуальную машину на сайте Microsoft Azure.

Microsoft Azure

Поиск по ресурсам, службам и документам (G+)

starovoytov.ruslan@gm...
КАТАЛОГ ПО УМОЛЧАНИЮ

Службы Azure

+

Создать ресурс

Azure Cosmos DB

Балансировщик нагрузки

Серверы Базы данных Azure...

Подписки

Виртуальные машины

Управление затратами ...

Политики конечной...

Службы приложений

Больше служб

Последние ресурсы

Имя	Тип	Последний просмотр
Monolith2	Виртуальная машина	месяц назад
(отключено) Azure для учащихся	Подписка	месяц назад
Подписка Azure 1	Подписка	месяц назад
Control	Виртуальная машина	2 месяцев назад
control	Группа ресурсов	4 месяцев назад
Monolith2_OsDisk_1_5e74479a840a4963a4d377f036c08c8a	Диск	4 месяцев назад

Создать

Поиск в Marketplace

Azure Marketplace

Показать все

Начало работы

Недавно созданные

Искусственный интеллект и машинное обучение

Аналитика

Безопасность

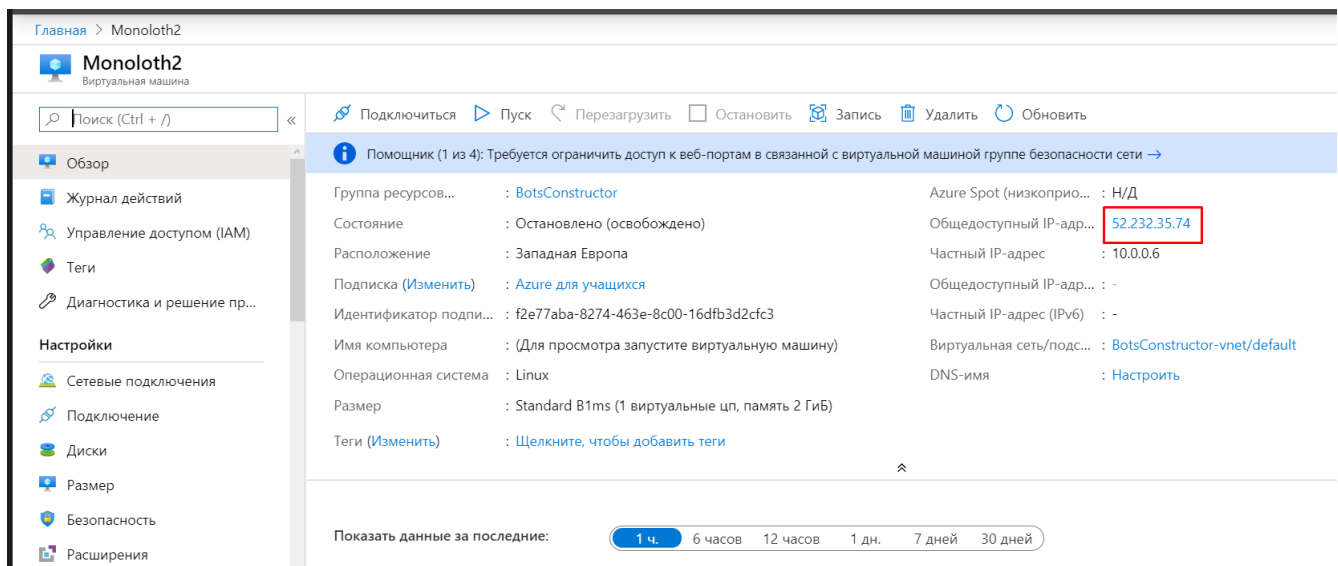
Популярные

Windows Server 2016 Datacenter

Краткие руководства и учебники

Ubuntu Server 18.04 LTS

Подробнее



После указания тарифного плана, пароля, региона нам будет выдан ip адрес машины. Зайдём на виртуальную машину через консоль. Для этого можно использовать Putty или Git bash, который идёт в комплекте с git. Выполним обновление системы командой.

```
Zenit@sharkan MINGW64 /d/Desktop
$ ssh ubuntu@51.144.49.111
ubuntu@51.144.49.111's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.0.0-1032-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Thu Apr 16 11:46:55 UTC 2020

System load:  0.01               Processes:    127
Usage of /:   14.0% of 28.90GB   Users logged in: 0
Memory usage: 18%               IP address for eth0: 10.0.0.4
Swap usage:   0%

 * Kubernetes 1.18 GA is now available! See https://microk8s.io for docs or
   install it with:

   sudo snap install microk8s --channel=1.18 --classic

 * Multipass 1.1 adds proxy support for developers behind enterprise
   firewalls. Rapid prototyping for cloud operations just got easier.

   https://multipass.run/

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

27 packages can be updated.
0 updates are security updates.

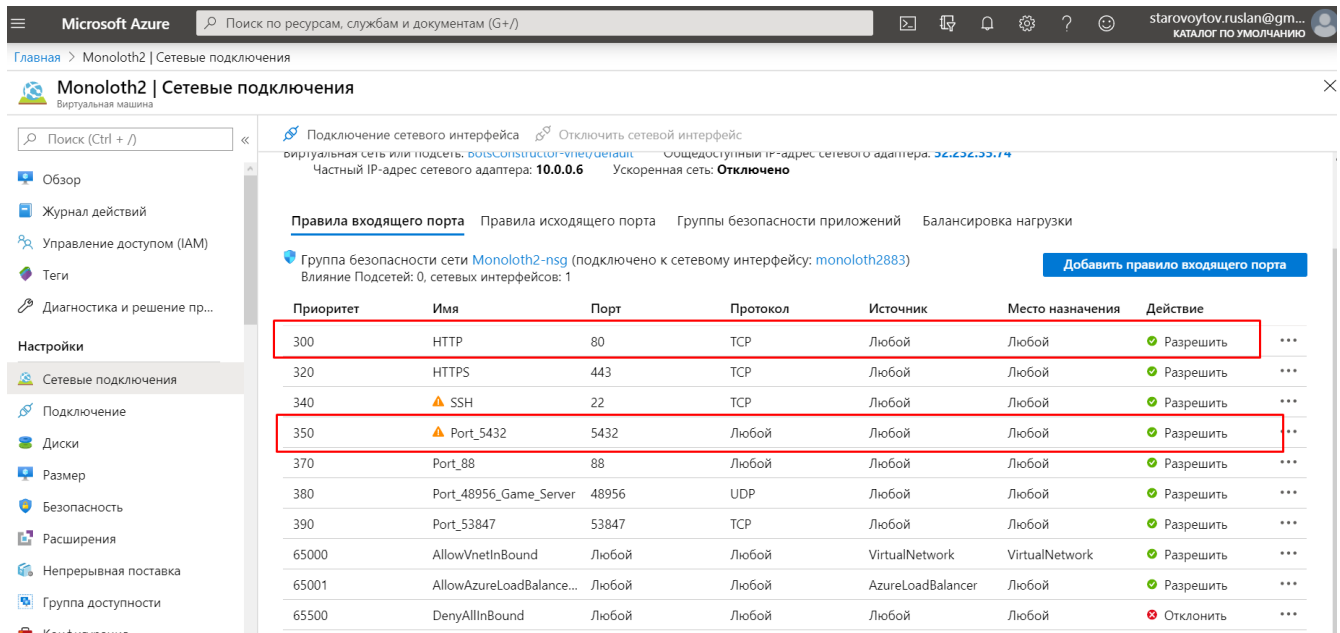
*** System restart required ***
Last login: Wed Apr 1 13:00:07 2020 from 193.161.14.235
ubuntu@MainServer:~$
```

```
apt-get update
```

Откроем порты для доступа к БД и нашему веб-приложению. PostgreSQL будет работать на порте 5432. Веб-сервер nginx будет слушать порт 80. Выполним команды на виртуальной машине.

```
iptables -I INPUT -p tcp -m tcp --dport 5432 -j ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Теперь порты открыты на ОС, но до сих пор не доступны извне. Нужно открыть порты машины на сайте Azure. Для этого в настройках виртуальной машины выберем «Сетевые подключения» → кнопка “Добавить правило безопасности для входящего трафика”. После добавления правила окно «Сетевые подключения» должно выглядеть так.



Теперь подключение к БД и серверу возможно извне.

Установим пакеты для того, чтобы наше серверное приложение смогло запускаться. Список команд можно найти на сайте документации Microsoft для каждой версии SDK и операционной системы.

<https://docs.microsoft.com/ru-ru/dotnet/core/install/linux-package-manager-ubuntu-1910>

Мы используем C# на SDK версии .Net Core 2. Выполним команды.

```
wget -q https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb  
sudo dpkg -i packages-microsoft-prod.deb
```

```
sudo add-apt-repository universe  
sudo apt-get install apt-transport-https  
sudo apt-get update  
sudo apt-get install dotnet-sdk-2.2=2.2.102-1
```

Для проверки правильности установки выполним команду.

```
ubuntu@MainServer:~$ dotnet --version  
3.1.102
```

Если сообщения об ошибке не возникло, то установка выполнена успешно.