Создание Windows Azure Virtual Machine для хостинга web-приложений

[Microsoft Azure](https://habrahabr.ru/hub/azure/)\*



В сегодняшней статье я постараюсь рассказать, как можно расширить границы использования Azure на примере создания в нем виртуальной машины и хостинга на ней веб-приложения.

Шаг 0. Получения доступа.

Делается это на странице [Предварительный просмотр компонентов](http://account.windowsazure.com/PreviewFeatures/), где можно выбрать то, что вы хотите попробовать. В данном случае выбираем *Virtual Machine*.

Шаг 1. Создание виртуальной машины

После того, как мы получили доступ к компонентам Windows Azure, самое время начать их использовать. Для этого в [портале управления](https://manage.windowsazure.com/) выберите пункт Virtual Machines и внизу страницы нажмите большую кнопку плюс.



Будет доступно два пункта: Quick Create и From Gallery. На самом деле разница между ними невелика, поэтому рассмотрим более полный путь. Выбираем *From Gallery*:



В данном диалоговом окне у нас есть несколько вариантов на выбор. Начнем с того, что помимо готовых образов ОС (видно на скриншоте), нам также предоставляется возможность загрузить свой собственный образ или VHD-диск, из которого впоследствие создастся виртуальная машина. Это будет удобно для крупных проектов, где нужно создать ферму одинаковых машин, не заморачиваясь с настройкой каждой из них. Нас же сейчас это не интересует, поэтому обратим свой взгляд на доступные готовые образы.

На выбор нам предлагается несколько вариантов. Есть Windows Server 2008 R2 (двух версий), линуксовые машинки, специализированный сервер под SQL Server и герой нашего сегодняшнего эксперимента — Windows Server 2012. Почему именно он? Поскольку я использую Windows Azure больше в ознакомительных целях, то я решил, что было бы неплохо попробовать и «новинку сезона» — новый сервер 2012/2016.

Итак, выбираем нужную нам ОС и жмем стрелку снизу. Попадаем на следующий экран мастера создания виртуальной машины:



Здесь все довольно понятно. Надо указать имя VM, пароль администратора и выбрать одну из предложенных конфигураций будущего «железа». Выбор довольно неплохой и может покрыть большинство требований по производительности для разных типов проекта. Я выбрал Small, что вполне меня устраивает. Идем дальше.

На третьем шаге мастера все тоже довольно понятно:



Для начала надо определиться с назначением будущей машины. Если мы хотим использовать ее как самостоятельный ресурс (а в данный момент мы именно этого и хотим), то выбираем *Standalone*. Режим *Connect To*, как написано в подсказке, позволит нам подключить данную ВМ в некую существующую сеть для разделения нагрузки. Нас это пока не интересует.

Затем надо придумать имя хоста, по которому будет доступна машина. Тут надо не путать имя машины на предыдущем шаге и данное имя хоста. Имя машины — это имя компьютера в Windows. Имя хоста же — это субдомен в домене cloudapp.net, по которому будет осуществляться доступ и управление сервером. Это имя должно быть уникально, поэтому нам тут же в мастере предоставляется подсказка о доступности или занятости того или иного имени.

Остальные поля довольно понятны и особо заострять внимание на них не стоит. Там надо выбрать аккаунт хранилища (либо существующий, либо создать новый), расположение ВМ и подписку, в рамках которой данная машина будет тарифицироваться.

На четвертом шаге вообще ничего делать не нужно, оставляем все как есть:



Нажимаем на галку внизу, и виртуальная машина начнет создаваться.

Шаг 2. Управление виртуальной машиной

После того, как вы создали виртуальную машину, она будет доступна в соответствующем списке:



Обязательно дождитесь, когда статус машины сменится с *Provisioning* на *Running*. Это будет означать, что машина готова к использованию. Если нажать на ее имя, то откроется панель управления виртуальной машиной:



Внизу страницы есть кнопка *Connect*. Она поможет вам получить правильный файл rdp для подключения к этой машине с помощью удаленного рабочего стола. Настало самое время зайти на свеженький сервер и поуправлять им.

Шаг 3. Управление сервером

На самом деле, в этой статье я не буду подробно рассказывать об администрировании Windows Server 2012. Поскольку в названии данной статьи упоминаются web-приложения, вот именно этим мы и займемся — установим IIS.

При первом логине в новый сервер нам откроется окно Server Manager:



По умолчанию, в Windows Server 2012 службы IIS не включены, поэтому их надо добавить. Выбираем пункт *Add roles and features*, после чего открывается мастер добавления ролей. Три раза жмем Next (особо любопытные почитают, что там пишут, но сейчас нас это не интересует), и в открывшемся дереве выбираем *Web Server (IIS)*. Далее нам необходимо обязательно выбрать нужные компоненты IIS, которые позволят запускать на нем ASP.NET MVC приложения. Точные параметры настройки я писать не буду, каждый сам решит, что именно ему нужно. Обязательно надо включить ASP.NET, остальное по вкусу.

Жмем несколько раз Next, настраиваем будущую роль как нам надо и после завершения процесса установки наш сервер с радостью готов принимать гостей. Убедиться в этом можно, перейдя по [localhost](http://localhost/). Должно появиться что-то вроде этого:



Шаг 4. Открытие доступа

Все здорово, вот мы наконец наладили веб-сервер, разместили на нем какое-то приложение и оно открывается по localhost. Как же теперь открыть его извне? На самом деле все довольно просто. Наверняка вы еще на этапе создания вирутальной машины пытались нажимать на ссылку в панели управления, которая имеет вид <ваша\_вирут\_машина>.cloudapp.net. И наверняка у вас ничего не отобразилось. Все потому, что к виртуальной машине закрыт доступ извне, причем не на уровне фаерволла самой операционной системы, а на уровне инфраструктуры Windows Azure.

Если вы в панели управления виртуальной машиной посмотрите наверх, то увидите там три пункта меню: *Dashboard*, *Endpoints* и *Configure*. С первым мы разобрались в самом начале, с последним все понятно — там настраиваются параметры «железа», а вот что такое Endpoints давайте разберемся.

В закладке Endpoints мы можем задавать то, какие протоколы и какие порты будут открыты для данной виртуальной машины и на какие физические порты они будут перенаправлены:



По умолчанию в этом списке будет лишь один порт — тот, по которому вы соединяетесь через Remote Desktop. Причем публичный порт может быть любым, что, видимо, сделано для большей безопасности. Для того, чтобы открывать веб-приложения на данной виртуальной машине, необходимо открыть 80й порт. На скриншоте выше это показано. После этого действия по ссылке <ваша\_вирут\_машина>.cloudapp.net вы должны будете увидеть ваше собственное приложение. Маленькая победа достигнута.

Шаг 5. Присваиваем собственный домен

На предыдущем шаге мы сделали все, чтобы приложение заработало в облаке и было доступно извне. Однако согласитесь, что было бы совсем не круто, если бы мы могли обращаться к нему только по субдомену и никак бы не могли присвоить свой один или несколько доменов.

Конечно же это сделать возможно и довольно просто. Для этого официальная документация советует два разных пути и оба из них связаны с редактированием зоны DNS. Первый подразумевает под собой то, что вы создаете CNAME запись, в которой указываете, что, например, [www.mydomain.com](http://www.mydomain.com/) будет перенаправляться на mywinazure.cloudapp.net. Этот метод прост, но слишком уж прямолинеен. Если у вас на сервере размещено несколько веб-приложений, или же на основном домене может быть несколько субдоменов, то этот способ не сработает. Благо, что есть второй, более качественный способ.

В качестве решения, которым, кстати, воспользовался и я, предлагается редактирование A записи в зоне DNS. В панели управления виртуальной машиной, в области справа, вы могли заметить два IP адреса. Один называется приватным и является IP самой машины, а второй — публичный, виртуальный (Virtual IP, VIP). Вот именно его и надо указывать в зоне A для редактируемого домена. В данном случае весь трафик будет посылаться на вашу виртуальную машину, где IIS будет определять домен и выдавать соответствующее приложение (при условии правильной настройки Web Sites в самом IIS). Все просто и очевидно до неприличия, но именно так оно и работает.

* [windows azure](https://habrahabr.ru/search/?q=%5Bwindows%20azure%5D&target_type=posts),
* [azure](https://habrahabr.ru/search/?q=%5Bazure%5D&target_type=posts),
* [microsoft](https://habrahabr.ru/search/?q=%5Bmicrosoft%5D&target_type=posts),
* [asp.net](https://habrahabr.ru/search/?q=%5Basp.net%5D&target_type=posts)