**«ОБЛАЧНЫЕ ТЕХНОЛОГИИ»**

**ЛАБОРАТОРНАЯ РАБОТА № 10**

**«Запуск пользовательского контейнера Windows в MS Azure»**

**Цель** – развернуть приложение ASP.NET в образе контейнера Windows на сайте [Docker Hub](https://hub.docker.com/) из Visual Studio и запустить его в пользовательском контейнере в Службе приложений Azure.

В ходе выполнения работу необходимо выполнить:

1. [Создать веб-приложения ASP.NET](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#create-an-aspnet-web-app)

2. [Опубликовать на сайте Docker Hub](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#publish-to-docker-hub)

[Вход в Azure](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#sign-in-to-azure)

[Создание приложения-контейнера Windows](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#create-a-windows-container-app)

[Переход к контейнеру приложения](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#browse-to-the-container-app)

[Просмотр журналов запуска контейнера](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#see-container-start-up-logs)

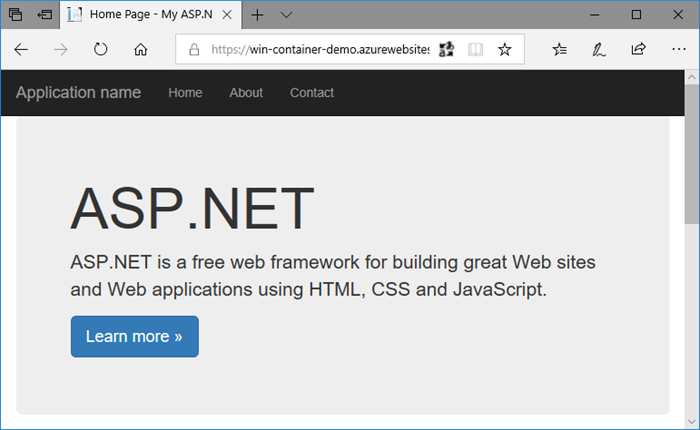
[Обновление на локальном компьютере и повторное развертывание](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#update-locally-and-redeploy)

[Использование другого родительского образа](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#use-a-different-parent-image)

[Дополнительная информация](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#next-steps)

## Краткие теоретические сведения

[Служба приложений Azure](https://docs.microsoft.com/ru-ru/azure/app-service/overview) предоставляет предопределенные стеки приложений на платформе Windows, например ASP.NET или Node.js, выполняющиеся в IIS. Предварительно настроенная среда Windows блокирует в операционной системе возможность административного доступа, установки программного обеспечения, изменений в глобальном кэше сборок и т. д. (см. раздел [Функциональные возможности операционной системы для службы приложений Azure](https://docs.microsoft.com/ru-ru/azure/app-service/operating-system-functionality)). Если приложению требуется более высокий уровень доступа, чем предусмотрено в предварительно настроенной среде, можно развернуть пользовательский контейнер Windows. В этом кратком руководстве показано, как развернуть приложение ASP.NET в образе Windows на сайте [Docker Hub](https://hub.docker.com/) из Visual Studio и запустить его в пользовательском контейнере в Службе приложений Azure.



## Порядок выполнения

### Замечания по назначению имен и других параметров используемых при выполнении работы.

При выполнении данной работы используйте следующие рекомендации по назначению имен различных ресурсов и других параметров (имена приложений, групп ресурсов, серверов, баз данных и т.п.):

1. При назначении имен, которые должны быть уникальными, используйте имена указанные в тексте или их модификации и обязательно включайте индивидуальный префикс: - xxyysssss

Где: xx – номер учебной группы;

yy – номер по журналу;

sssss – сокращенное наименование специальности (например, poit, isit, и т.п.).

2. При назначении паролей следуйте подсказкам, выдаваемым при редактировании соответствующих полей.

### Предварительные требования

Для выполнения этой лабораторной работы необходимо предварительно выполнить следующие действия:

A. [Установить Visual Studio 2017](https://www.visualstudio.com/downloads/), а также следующие рабочие нагрузки: **ASP.NET и веб-разработка** и **разработка Azure**. Если у вас уже установлена версия Visual Studio 2017, сделайте следующее:

1. Установите последние обновления для Visual Studio, выбрав **Справка** > **Проверить наличие обновлений**.

2. Добавьте рабочие нагрузки в Visual Studio, выбрав **Инструменты** > **Get Tools and Features** (Получить инструменты и компоненты).

B. [Зарегистрировать учетную запись центра Docker](https://hub.docker.com/).

C. [Установить Docker для ОС Windows](https://docs.docker.com/docker-for-windows/install/).

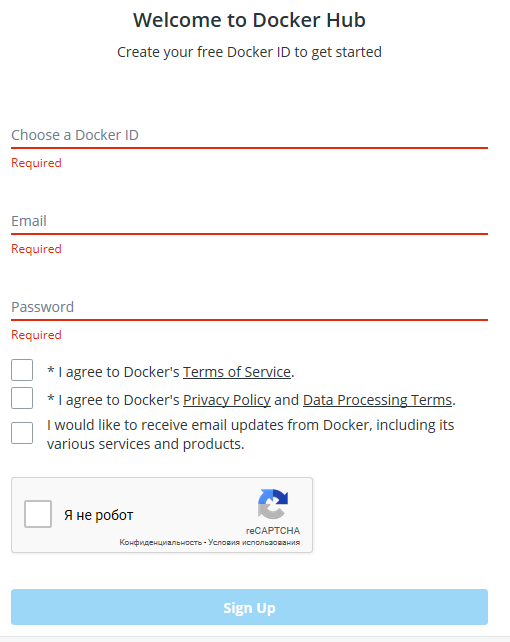
D. [Переключить Docker для запуска контейнеров Windows](https://docs.microsoft.com/virtualization/windowscontainers/quick-start/quick-start-windows-10).

## Задание 1. Установка и настройка Docker в ОС Windows

### Шаг 1. Создайте бесплатную учетную запись в центре Docker

1. Перейдите на сайт Docker Hub, следуя по ссылке: <https://hub.docker.com/>

2. Кликните ссылку "Sign up for Docker Hub". Откроется окно создания бесплатного аккаунта на Docker Hub.



3. Введите параметры акаунта, следуя рекомендациям по вводу имен и других параметров при заполнении форм, приведенным выше:

- идентификатор: - от 4 до 30 знаков (цифры и строчные буквы);

- адрес действующей эл.почты;

- пароль длиной от 6 до 128 символов;

- установите отметку в первых двух боксах "I agree …";

- кликните бокс в поле "Я не робот".

### Шаг 2. Установить Дocker в ОС Windows

1. Перейдите на сайт Docke, следуя по ссылке: <https://docs.docker.com/docker-for-windows/install/>

2. Кликните ссылку "Download from Docker Hub"

### Шаг 3. Переключение Docker для запуска контейнеров Windows

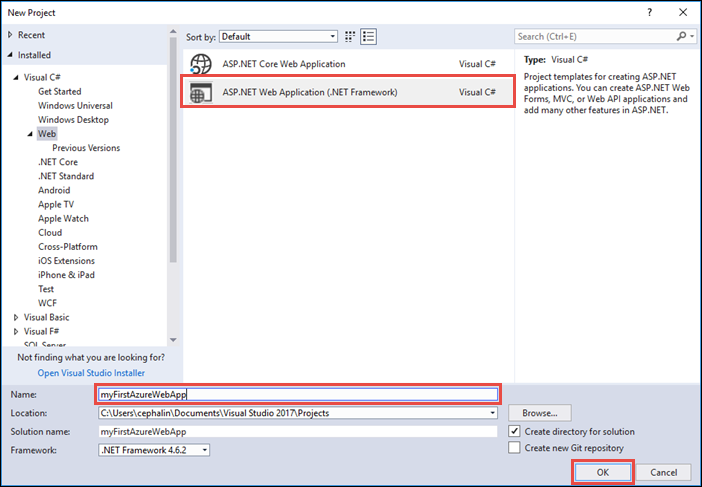
## Задание 2. Создание веб-приложения ASP.NET

### Шаг1. Создайте проект в Visual Studio.

1. Находясь в окне Visual Studio последовательно выберите пункты меню **Файл > Создать > Проект**.

2. В диалоговом окне **Новый проект** последовательно выберите пункты **Visual C# > Интернет > Веб-приложение ASP.NET (.NET Framework)**.

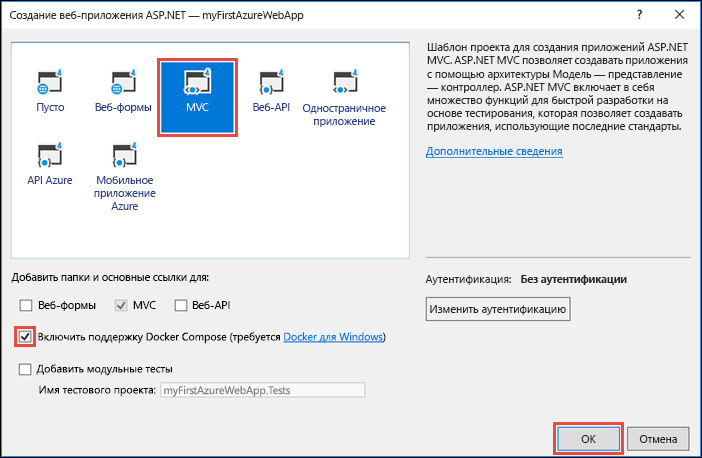
3. Присвойте приложению имя myFirstAzureWebApp и нажмите кнопку **ОК**.



4. Можно развернуть любой тип веб-приложения ASP.NET в Azure. Для примера в этом руководстве выберите шаблон **MVC** и задайте для аутентификации значение **Без аутентификации**.

5. Выберите **Включить поддержку Docker Compose**.

6. Нажмите кнопку **ОК**.



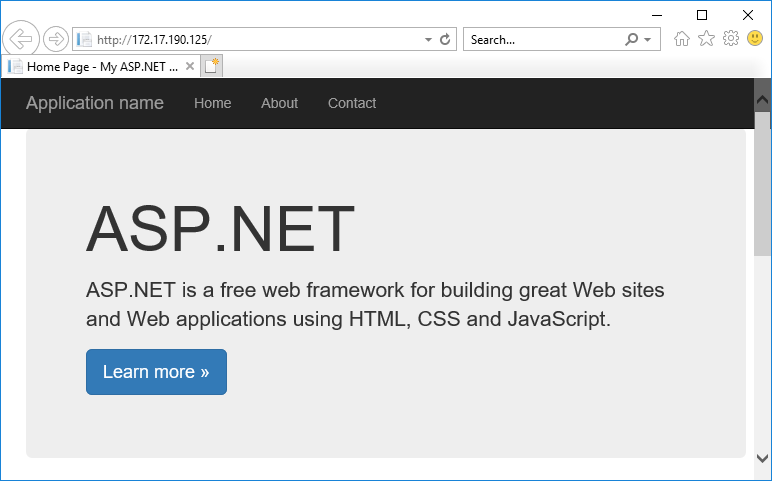
Если файл Dockerfile не открылся автоматически, откройте его в **обозревателе решений**.

Необходимо использовать [поддерживаемый родительский образ](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#use-a-different-parent-image). Измените родительский образ, заменив строку FROM приведенным ниже кодом. Затем сохраните файл.

Dockerfile

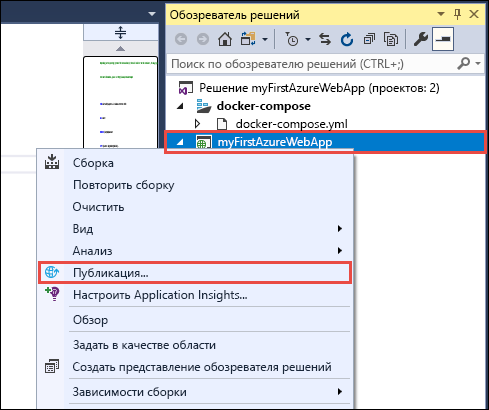
FROM microsoft/aspnet:4.7.1

В меню последовательно выберите пункты **Отладка > Запуск без отладки**, чтобы запустить веб-приложение локально.

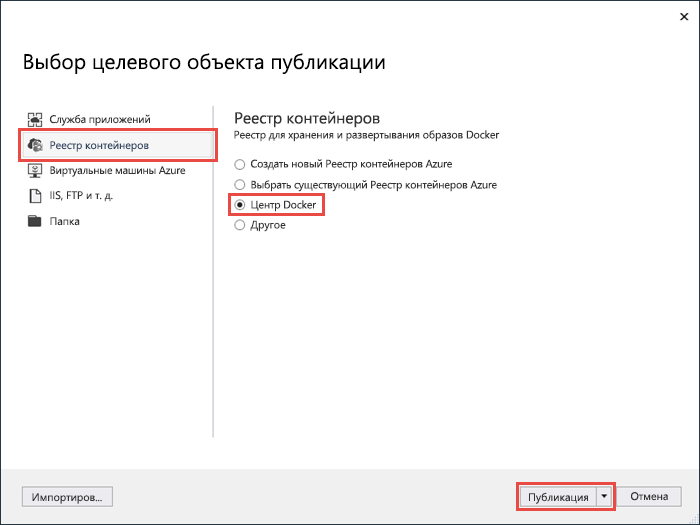


Публикация на сайте Docker Hub

Щелкните правой кнопкой мыши проект **myFirstAzureWebApp** в **обозревателе решений** и выберите **Опубликовать**.

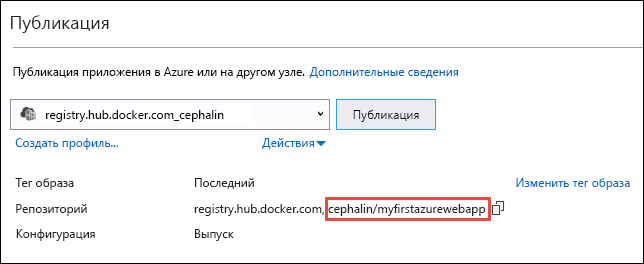


Мастер публикации запустится автоматически. Выберите **Реестр контейнеров** > **Docker Hub** > **Опубликовать**.



Укажите данные учетной записи Docker Hub и щелкните **Сохранить**.

Дождитесь завершения развертывания. Теперь на странице **Публикация** отображается имя репозитория, который будет использован позже в службе приложений.



Скопируйте имя этого репозитория для последующего использования.

Вход в Azure

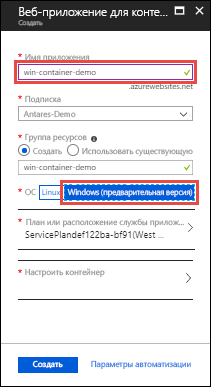
Войдите на портал Azure по адресу <https://portal.azure.com>.

Создание приложения-контейнера Windows

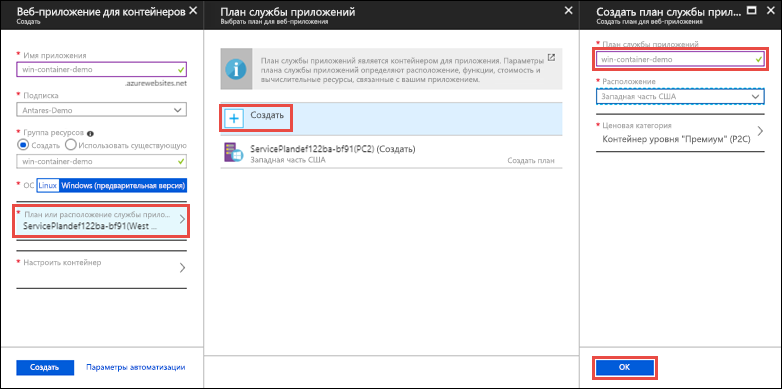
Выберите **Создать ресурс** в верхнем левом углу окна портала Azure.

В поле поиска над списком ресурсов Azure Marketplace найдите и выберите **Веб-приложение для контейнеров**.

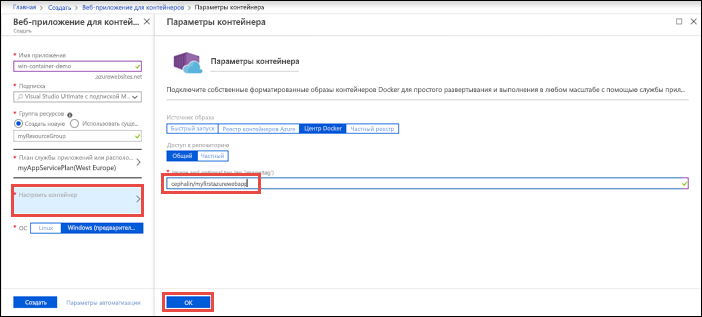
Укажите имя приложения, например win-container-demo, примите значения по умолчанию, чтобы создать группу ресурсов, а затем щелкните **Windows (предварительная версия)** в поле **ОС**.



Создайте план службы приложений, щелкнув **Расположение или план службы приложений** > **Создать**. Укажите имя нового плана, примите значения по умолчанию и нажмите кнопку **ОК**.



Щелкните **Настроить контейнер**. На странице **Образ и дополнительный тег** укажите имя репозитория, скопированное при [публикации на сайте Docker Hub](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#publish-to-docker-hub), затем нажмите кнопку **ОК**.

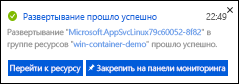


Если у вас есть пользовательский образ для веб-приложения в другом расположении, например [реестре контейнеров Azure](https://docs.microsoft.com/ru-ru/azure/container-registry/) или любом другом частном репозитории, его можно настроить здесь.

Нажмите кнопку **Создать** и подождите, пока Azure создаст необходимые ресурсы.

Переход к контейнеру приложения

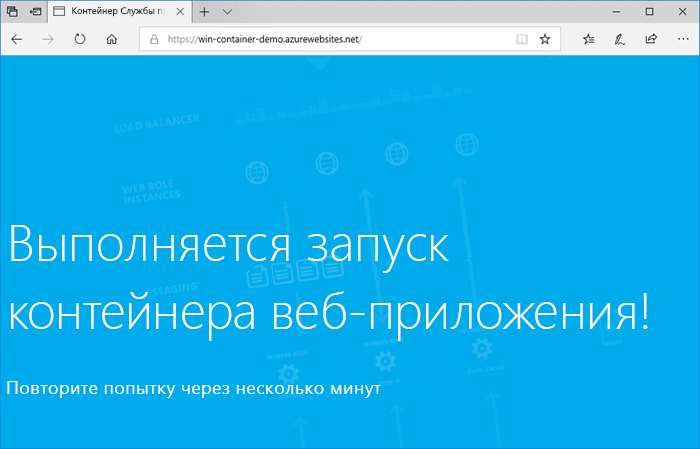
По завершении операции Azure отображается окно уведомления.



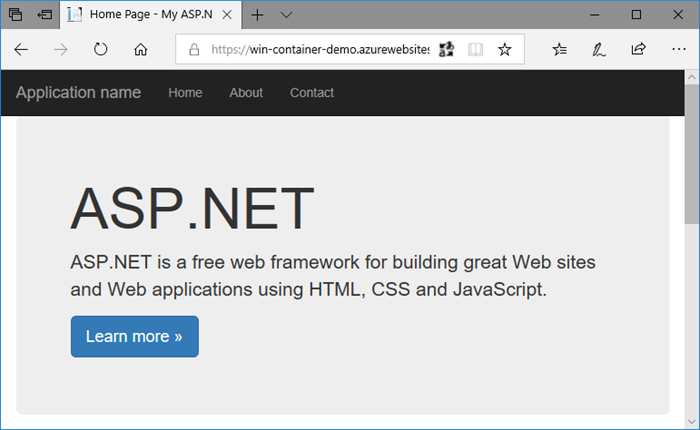
Щелкните **Перейти к ресурсу**.

На странице приложения щелкните ссылку в разделе **URL-адрес**.

В браузере откроется следующая страница:



Подождите несколько минут и повторите попытку, пока не отобразится домашняя страница ASP.NET по умолчанию.



**Поздравляем!** Вы запустили свой первый пользовательский контейнер Windows в службе приложений Azure.

Просмотр журналов запуска контейнера

Загрузка контейнера Windows может занять некоторое время. Чтобы просмотреть ход выполнения, перейдите по следующему URL-адресу, указав вместо <app\_name > имя приложения.

https://<app\_name>.scm.azurewebsites.net/api/logstream

Потоковые журналы выглядят следующим образом:

2018-07-27T12:03:11 Welcome, you are now connected to log-streaming service.

27/07/2018 12:04:10.978 INFO - Site: win-container-demo - Start container succeeded. Container: facbf6cb214de86e58557a6d073396f640bbe2fdec88f8368695c8d1331fc94b

27/07/2018 12:04:16.767 INFO - Site: win-container-demo - Container start complete

27/07/2018 12:05:05.017 INFO - Site: win-container-demo - Container start complete

27/07/2018 12:05:05.020 INFO - Site: win-container-demo - Container started successfully

Обновление на локальном компьютере и повторное развертывание

В **обозревателе решений** откройте Views\Home\Index.cshtml.

Найдите тег HTML <div class="jumbotron"> в верхней области и замените его следующим кодом:

HTML

<div class="jumbotron">

<h1>ASP.NET in Azure!</h1>

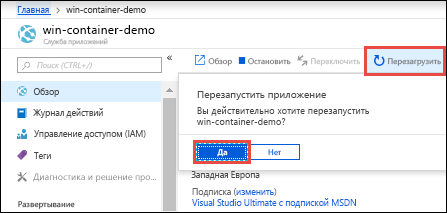
<p class="lead">This is a simple app that we’ve built that demonstrates how to deploy a .NET app to Azure App Service.</p>

</div>

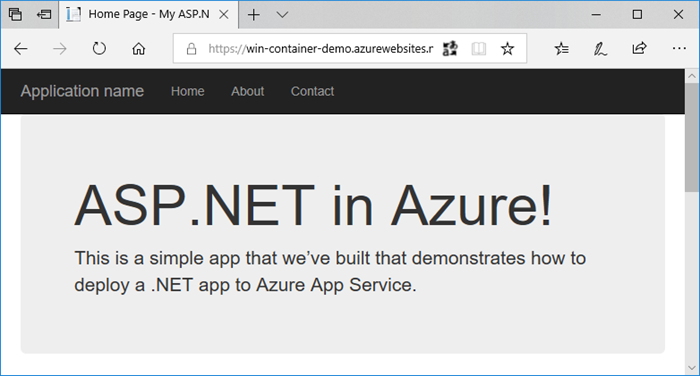
Чтобы выполнить повторное развертывание в Azure, щелкните правой кнопкой мыши проект **myFirstAzureWebApp** в **обозревателе решений**, а затем выберите **Опубликовать**.

На странице публикации выберите **Опубликовать** и дождитесь завершения публикации.

Чтобы указать службе приложений извлечь новый образ из Docker Hub, перезапустите приложение. На странице приложения на портале щелкните **Перезапустить** > **Да**.



Еще раз [перейдите к контейнерному приложению](https://docs.microsoft.com/ru-ru/azure/app-service/app-service-web-get-started-windows-container#browse-to-the-container-app). После обновления веб-страницы сначала должна появиться страница "Starting up" (Запуск) приложения, а через несколько минут должна отобразиться обновленная веб-страница.



Использование другого родительского образа

Вы можете использовать другой пользовательский образ Docker, чтобы запустить приложение. При этом необходимо выбрать правильный [родительский образ](https://docs.docker.com/develop/develop-images/baseimages/) для платформы, которую вы хотите использовать:

Для развертывания приложений .NET Framework используйте родительский образ на основе выпуска [канала долгосрочного обслуживания (LTSC)](https://docs.microsoft.com/windows-server/get-started/semi-annual-channel-overview#long-term-servicing-channel-ltsc) для Windows Server Core 2016.

Для развертывания приложений .NET Core используйте родительский образ на основе выпуска [канала долгосрочного обслуживания (LTSC)](https://docs.microsoft.com/windows-server/get-started/semi-annual-channel-overview#long-term-servicing-channel-ltsc) для Windows Server Nano 2016.

Скачивание родительского образа во время запуска приложения занимает некоторое время. Но вы можете ускорить запуск, используя один из следующих родительских образов, уже кэшированных в службе приложений Azure:

[microsoft/iis](https://hub.docker.com/r/microsoft/iis/):windowsservercore-ltsc2016, последняя версия

[microsoft/iis](https://hub.docker.com/r/microsoft/iis/):nanoserver-sac2016

[microsoft/aspnet](https://hub.docker.com/r/microsoft/aspnet/):4.7.2-windowsservercore-ltsc2016, 4.7.2, последняя версия

[microsoft/dotnet](https://hub.docker.com/r/microsoft/dotnet/):2.1-aspnetcore-runtime

[microsoft/dotnet](https://hub.docker.com/r/microsoft/dotnet/):2.1-sdk

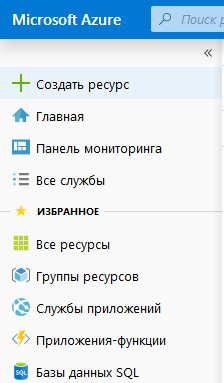
Задание 1. Создать облачный сервис Azure.

Шаг 1. Создать сервис Azure из шаблона ASP.NET MVC.

Задание 4. Удалить облачную службу из облака Azure

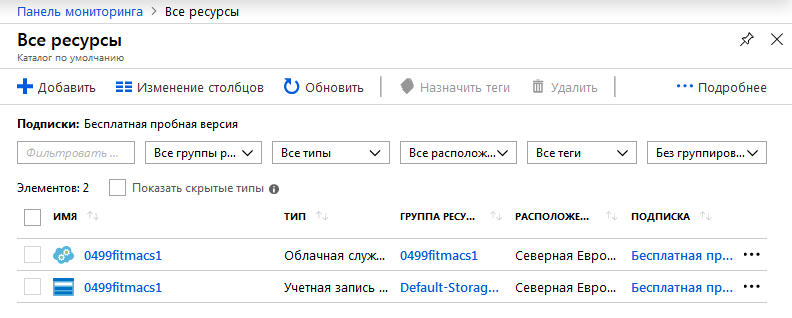
Шаг 1. Удалить облачную службу.

1. Войти на портал Azure

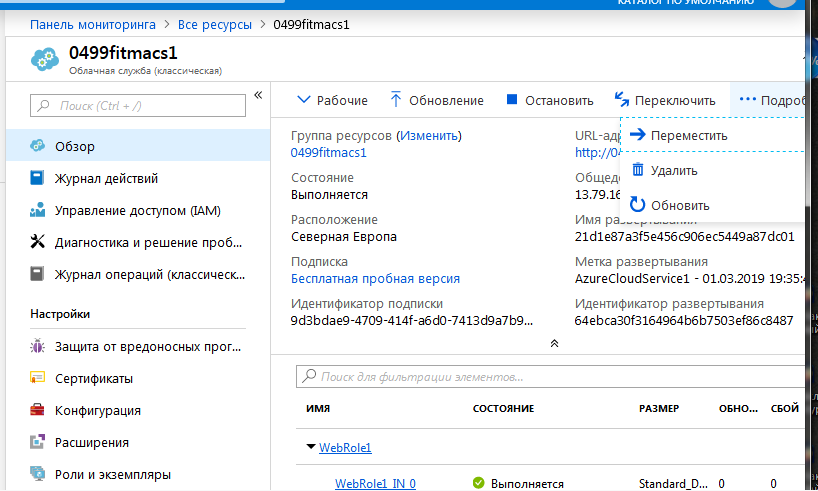


2. В области выбора служб кликнуть «Все ресурсы»

Откроется страница Ваших ресурсов.

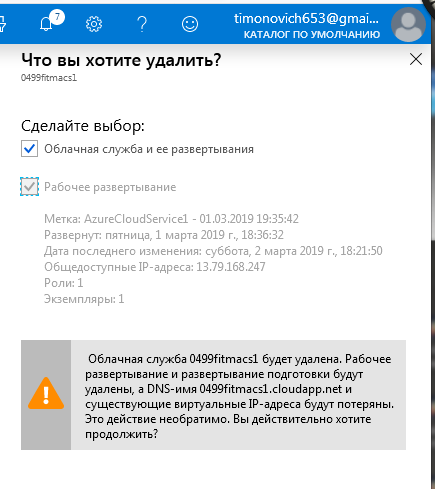


3. Отметить строку облачной службы и кликнуть по ее имени, откроется страница сервиса.



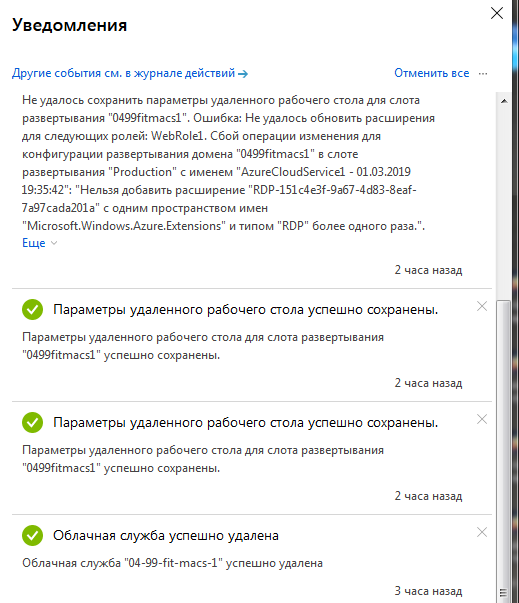
4. В верхней части этой страницы кликнуть ссылку «Подробнее», откроется меню в котором следует выбрать «Удалить».

Появиться страница с запросом на подтверждение удаления.



5. Поставить отметку в боксе «Облачная служба и ее развертывание». Подтвердить кликнув кнопку «Удалить».

Кликнуть значек  в строке навигации для просмотра уведомлений:



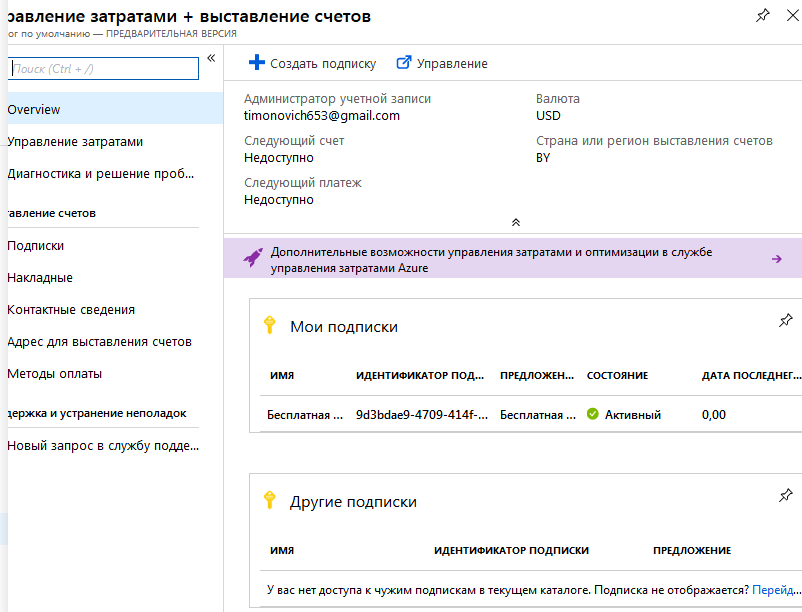
Шаг 2. Оценка затрат на услуги Azure

1. Перейдите на портал Amazon.

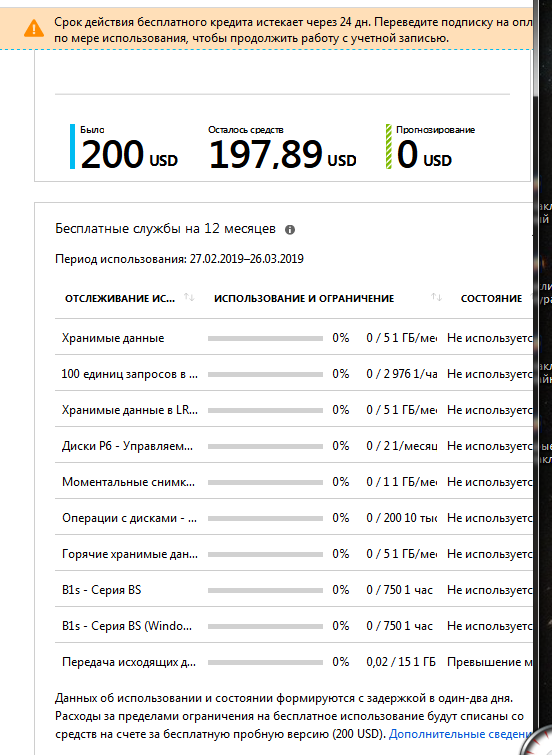
2. Просмотрите Ваши затраты связанные с выполнением данной работы, воспользовавшись сервисом «Управления затратами»



Откроется страница «Управление затратами + Выставление счетов»



На странице «Управление затратами + Выставление счетов» кликните по своей учетной записи для того, чтобы посмотреть Ваш баланс затрат.



3. Выйдите с портала Azure и закройте браузер.

Задание 5. Подготовить отчет о выполнении лабораторной работы

В отчете необходимо кратко описать все выполненные действия, привести скриншоты с результатами выполнения каждого из заданий. В отчет также должны быть включены ответы на контрольные вопросы.

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Приведите этапы создания и развертывания облачного сервиса в MS Azure.

2. Приведите последовательность создания облачного сервиса в MS Azure.

3. Приведите понятие ресурсов для развертывания облачного сервиса в MS Azure и дайте характеристику плана ресурсов.

4. При помощи каких инструментов можно проверить результаты развертывания облачного сервиса в MS Azure.