Лабораторная #6.

Azure cache for Redis

В этой работе мы познакомимся с тем как создать Redis Cache и обратиться к нему из клиента.

Что конкретно мы сделаем:

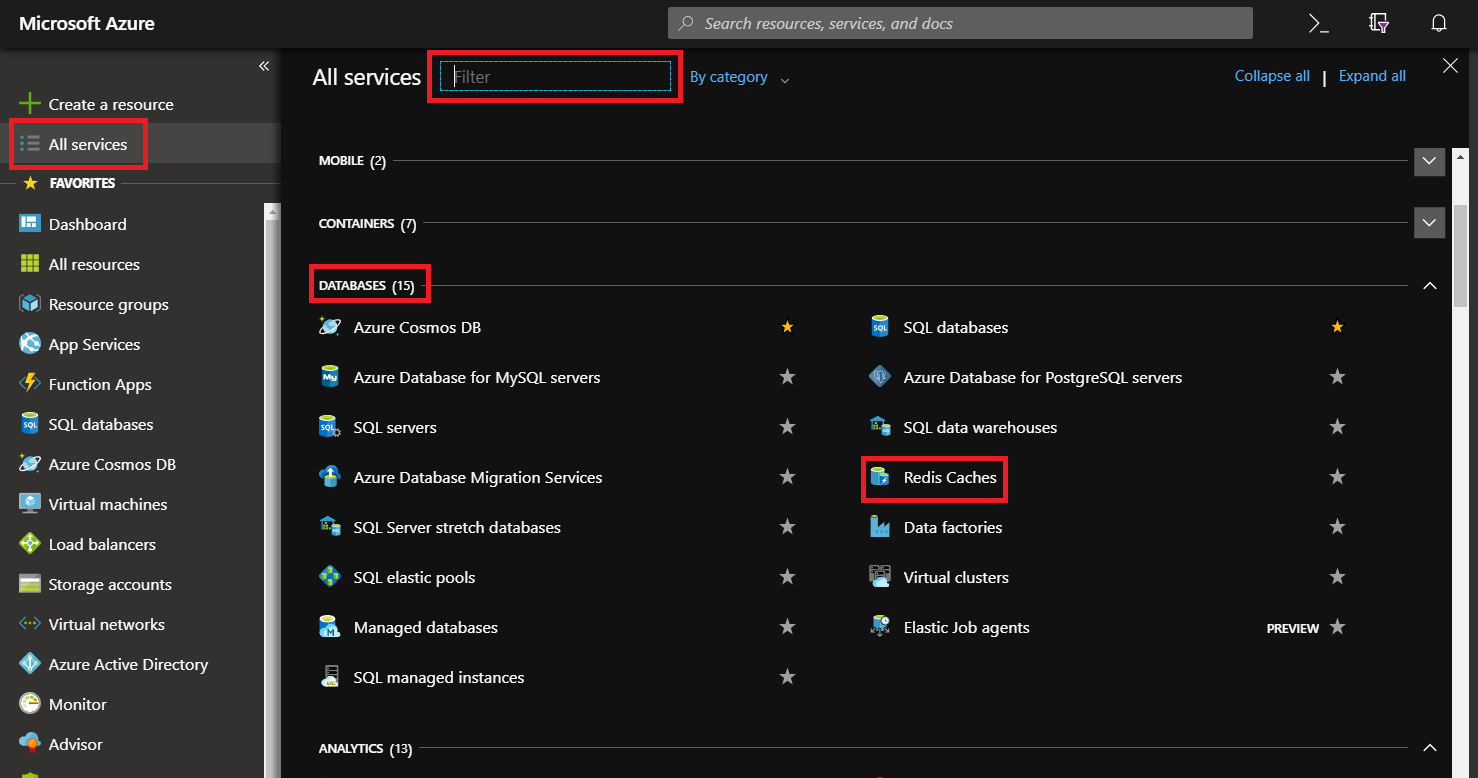
* Создадим Redis Cache используя портал Azure
* Посмотрим на консоль для управления Redis
* Создадим простое консольное клиентское приложение используя .Net Core
* Подключим его к Azure Redis Cache
* Добавим несколько записей в кэш и прочитаем их через приложение
* Проверим через консоль Redis данные в кэше

Под конец мы подчистим ресурсы – удалим ресурсную группу со всеми элементами.

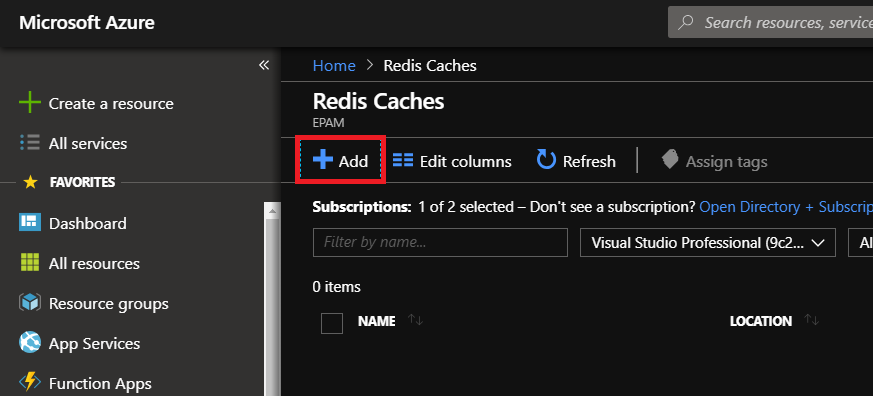
# Задание 01. Создание экземпляра Redis Cache

В этом задании мы создадим экземпляр Redis Cache на портале Azure, который в дальнейшем будет использоваться для хранение данных переданных с клиента.

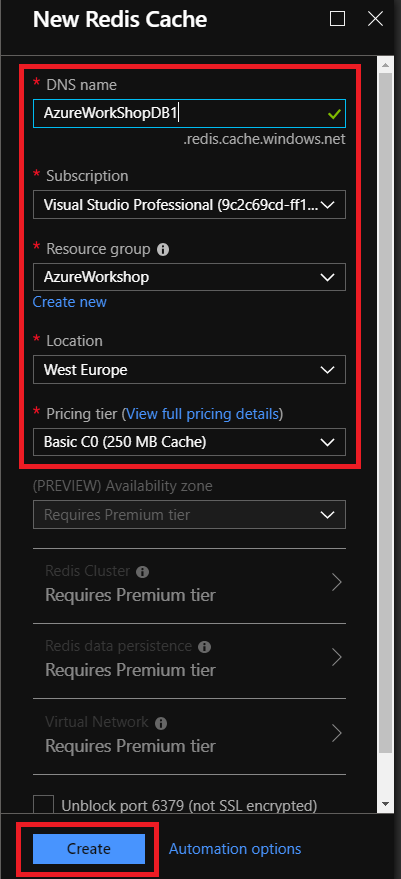
1. На Azure Portal нажмите кнопку **All services** и в разделе **Databases** выберите **Redis Caches** или воспользуйтесь фильтром для поиска



1. В открывшемся списке кэшей нажмите кнопку **Add**



1. Укажите настройки нового кэша в соответствии с таблицей ниже и нажмите кнопку **Create**



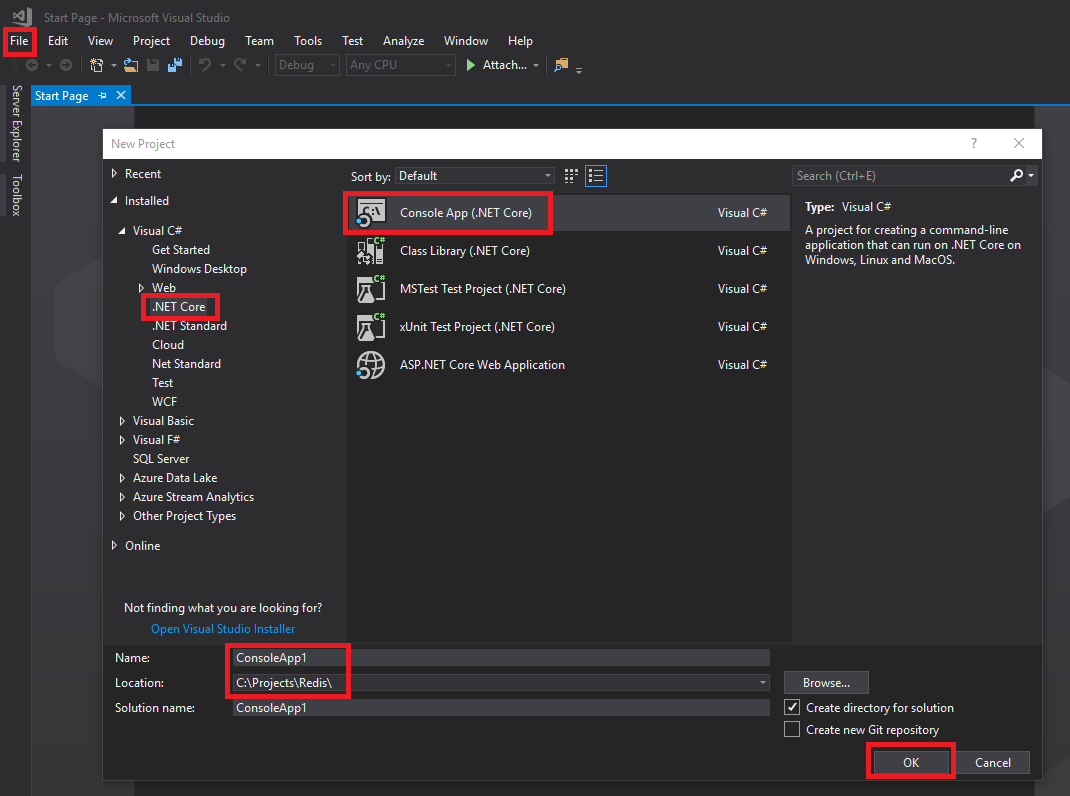
|  |  |
| --- | --- |
| Настройка | Значение |
| DNS name | AzureWorkshop<some numbers> |
| Subscription | Ваша подписка |
| Resource group | RedisCacheRG |
| Location | West Europe |
| Pricing tier | Basic C0 (250MB Cache) |

1. Нажмите на значок нотификаций в верхнем правом углу, дождитесь появления сообщения **Deployment succeeded** и нажмите на кнопку **Go to resource**.
2. В правом верхнем углу нажмите  для закрепления плитки кэша на рабочем столе портала.
3. Закройте все открытые окна и убедитесь, что плитка кэша видна на рабочем столе.

# Задание 02. Создание консольного приложения

Здесь мы создадим консольное приложение на .Net Core, используя для этого Visual Studio. В процессе создания приложения мы воспользуемся 2 библиотеками из репозитория NuGet: StackExchange.Redis – для доступа к Redis Cache, и Newtonsoft.Json – для сериализации и десириализации объектов в Json формат.

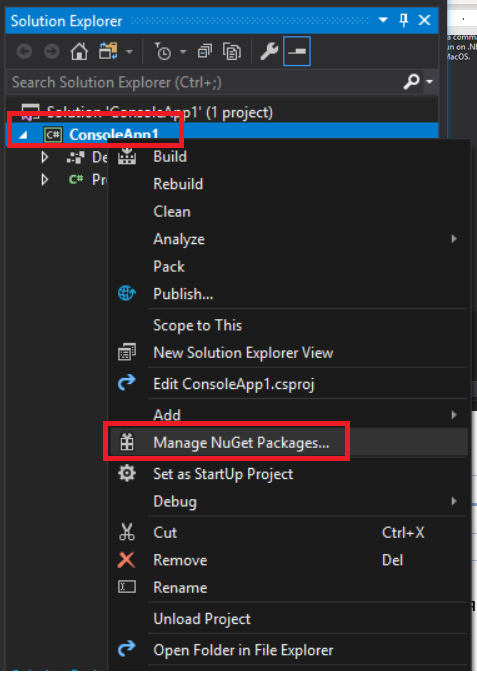
1. Запустите Visual Studio и создайте новый проект используя темплейт **Console App (.Net Core)**: **File > New > Project**.



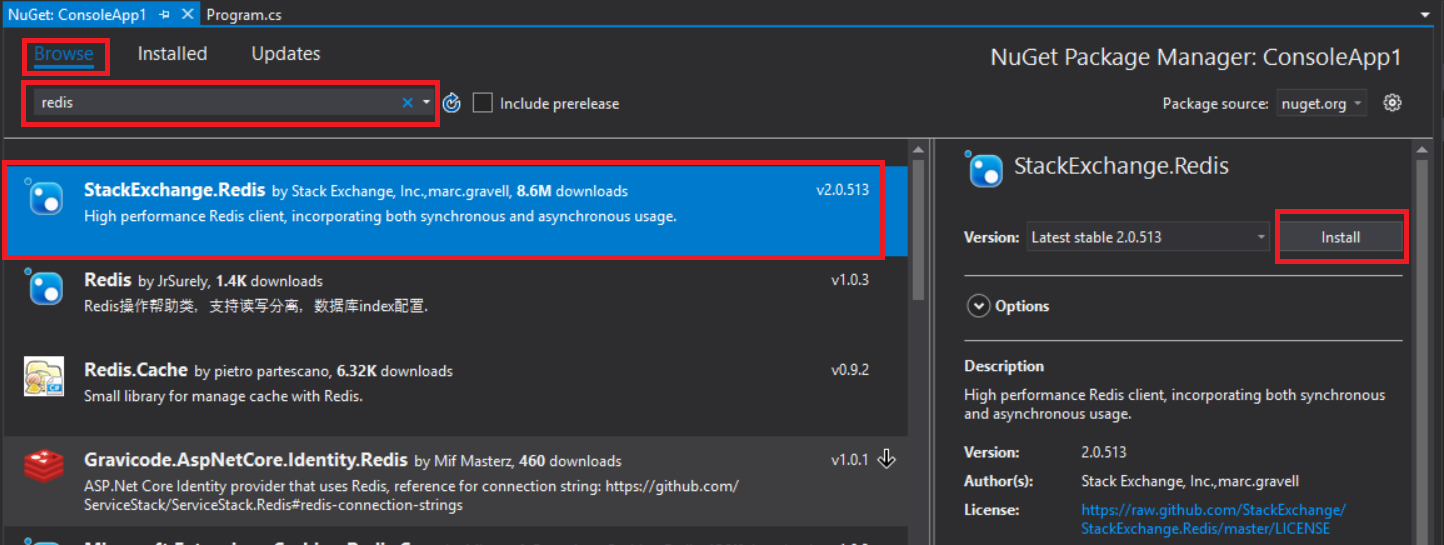
1. Введите имя проекта и выберите его место хранени на диске:

|  |  |
| --- | --- |
| Настройка | Значение |
| Name | ConsoleApp1 |
| Location | C:\Projects\Redis\ |

1. Добавьте в проект библиотеку **StackExchange.Redis** из NuGet репозитария. Для откройте локальное меню проекта в **Solution Explorer** и выберите там пункт **Manage NuGet Packages…**



1. В открывшемся окне перейдите на вкладку **Browse** и в поле фильтра введите **redis**, для ускорения поиска требуемой библиотеки **StackExchange.Redis,** выделите данную библиотеку и нажмите **Install.** Необходимы пакеты будут загружены из репозитория.



1. Таким же образом добавьте другую библиотеку **Newtonsoft.Json**
2. Cкомпилируйте приложение: **Build > Build Solution**
3. И запустите его в отладке **Debug > Start Debugging…,** тем самым убедившись что приложение работоспособно.
4. Используя **Solution Explorer** откройте файл Program.cs
5. В секции подключаемых модулей using (в самом начале файла) добавьте 2 новых модуля, которые мы подключили к проекту:

using Newtonsoft.Json;

using StackExchange.Redis;

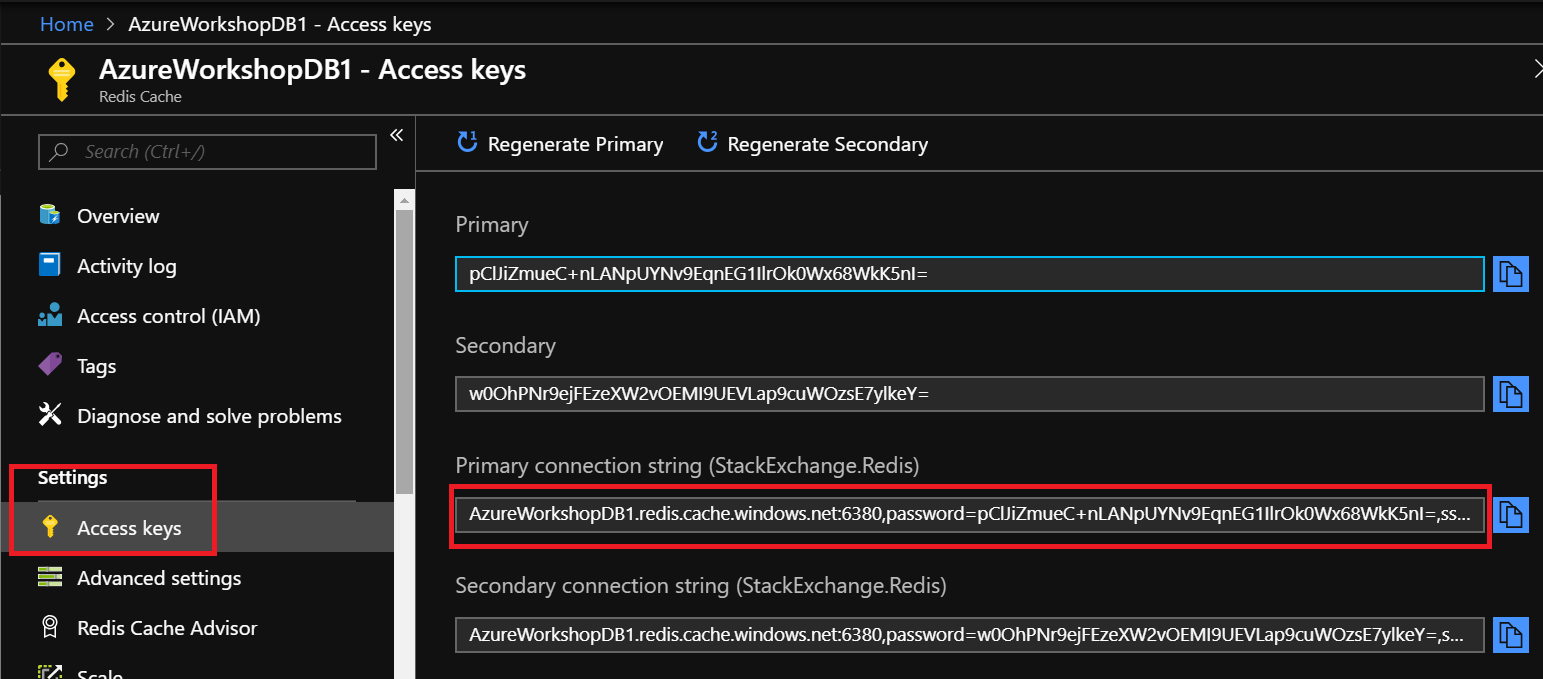
1. В начале метода Main класса Program добавьте следующий код:

ConnectionMultiplexer connectionMultiplexer = ConnectionMultiplexer.Connect("");

IDatabase cache = connectionMultiplexer.GetDatabase();

Данный код является заготовкой для создания подключения к кэшу. Как вы заметили метод Connect содержит пустую строку которую мы заполним строкой подключения к кэшу скопировав ее с портала Azure на следующем шаге.

1. Вернитесь на портал Azure и откройте настройки экземпляра кэша созданного вами ранее, используя закрепленную плитку Redis Cache из задания 1. Вы перейдете на панель **Overview** вашего кэша.
2. В списке настроек кэша перейдите на элемент **Access keys**, и скопируйте сроку подключения к кешу из поля **Primary connection string**



1. Вставьте данную строку в метод Connect, класса ConnectionMultiplexer в проекте консольного приложения
2. Скомпилируйте и запустите приложения, убедившись что клиент успешно подключается к кешу Redis.
3. Добавьте следующий код ниже вместо строки Сonsole.WriteLine("Hello World!");

cache.StringSet("Key1", "Hello from Azure Redis!");

var foo1 = new Foo()

{

Age = 1,

Name = "Foo1"

};

var serializedFoo1 = JsonConvert.SerializeObject(foo1);

cache.StringSet("serializedFoo1", serializedFoo1);

var foo2 = new Foo()

{

Age = 2,

Name = "Foo2"

};

var serializedFoo2 = JsonConvert.SerializeObject(foo2);

cache.StringSet("serializedFoo2", serializedFoo2);

Console.WriteLine(cache.StringGet("Key1"));

Console.WriteLine(cache.StringGet("serializedFoo1"));

Console.WriteLine(cache.StringGet("serializedFoo2"));

Console.ReadKey();

Данный код последовательно добавляет и считывает из кэша занчения объектов с помощью методов StringSet и StringGet.

1. После класса Program добавьте код нового класса Foo:

internal class Foo

{

public Foo()

{

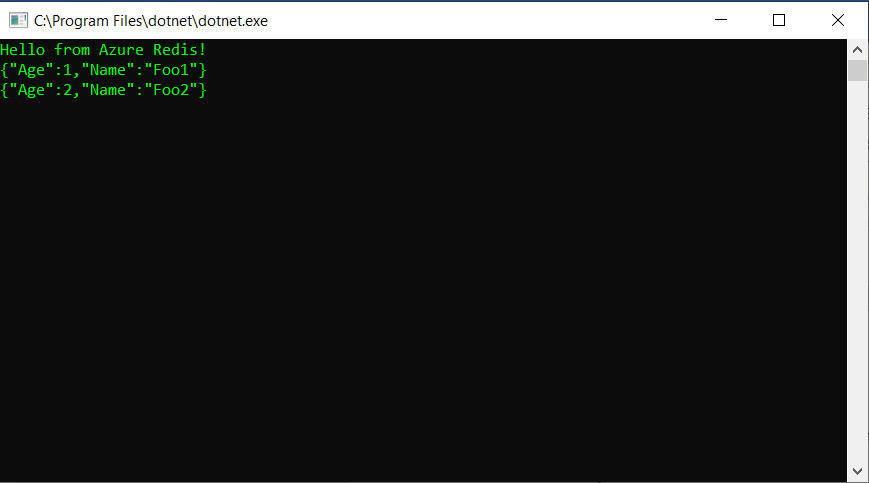
}

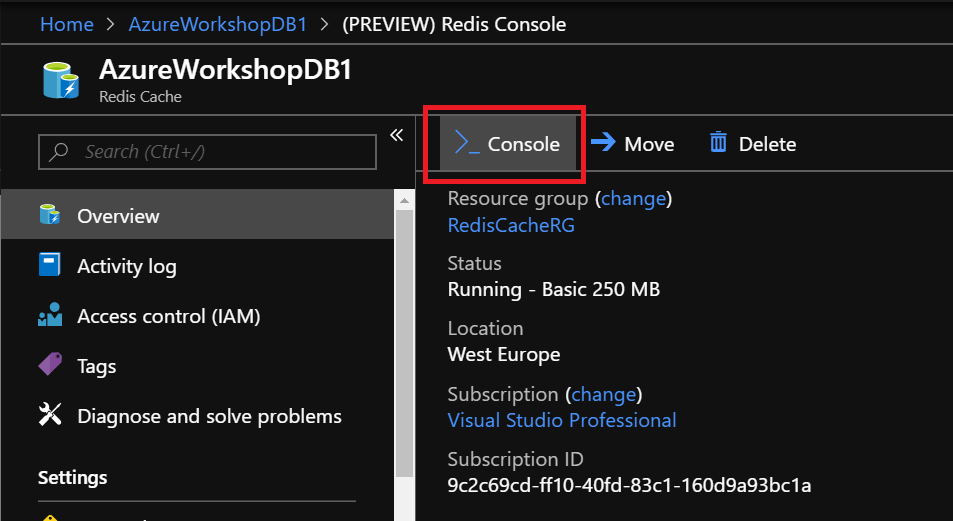
public int Age { get; set; }

public string Name { get; set; }

}

1. Сохраните весь Solution
2. Скомпилируйте и запустите приложение.
3. Приложение запуститься, подключится к кэшу, запишет в него объекты, считает их и выведет на экран



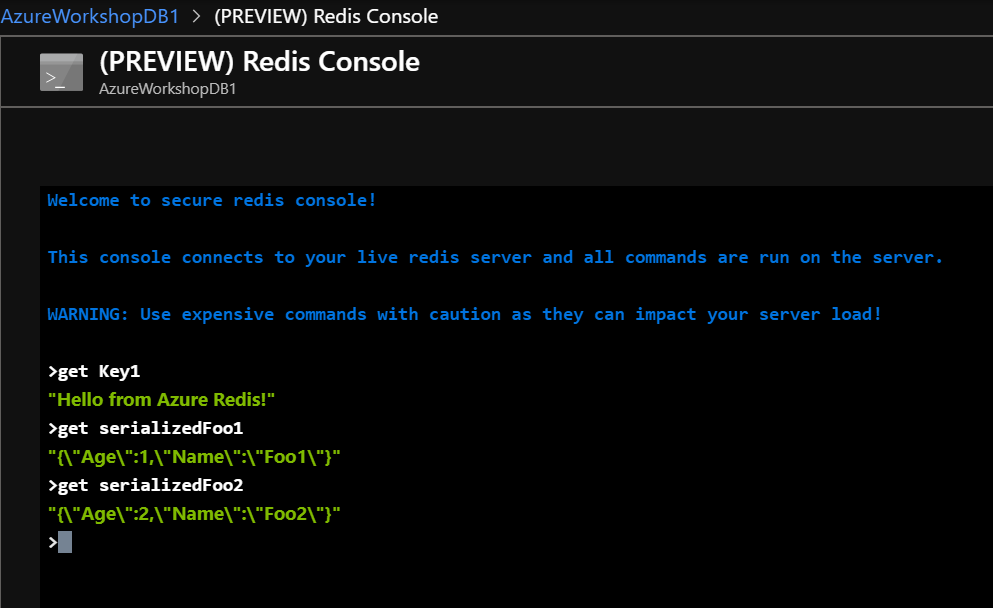
1. Для остановки работы клиента нажмите любую клавишу в консоли Windows.
2. Вернитесь на портал Azure в панель Overview вашего кэша
3. Запустите консоль Redis
4. Введите последовательно 3 команды

>get Key1

> get serializedFoo1

> get serializedFoo2

Данные команды получают значения по ключам из апшего кэша:

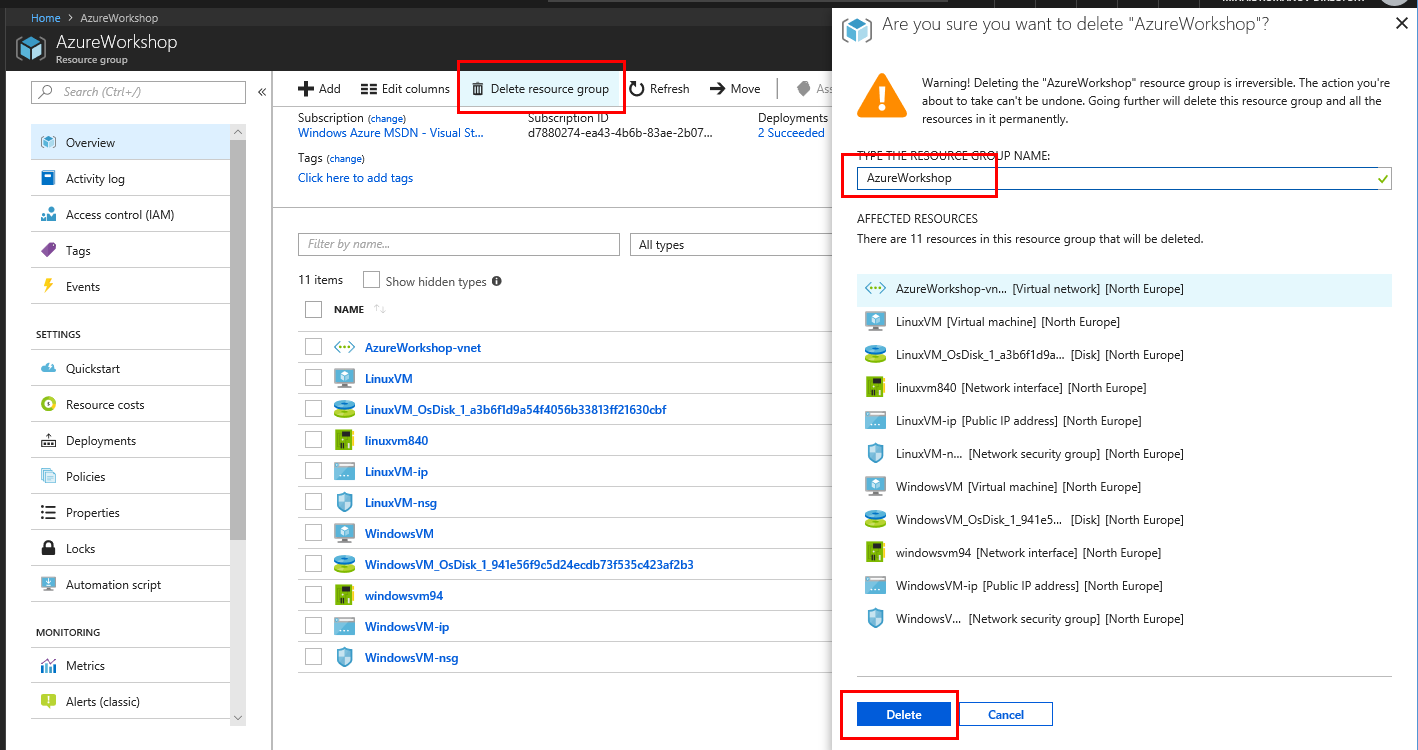


# Задание 03. Очистка ресурсов

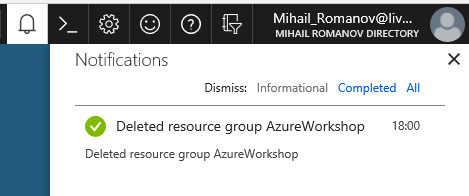
В этой задаче мы удалим все ране созданные ресурсы, включая кэш и его данные. Для этого мы просто удалим ресурсную группу.

Будьте внимательны – удаление ресурсной группы может быть достаточно продолжительным по времени действием!

1. Перейдите на Azure portal и найдите ресурсную группу содержащую экземпляр вашего кэша и откройте ее.
2. В окне группы нажмите **Delete resource group**, в открывшемся окне введите или скопируйте и вставьте имя группы и нажмите кнопку **Delete**



1. Откройте окно нотификаций и дождитесь окончания удаления ресурсной группы



1. Удалите с рабочего стола ярлыки всех ресурсов

Поздравляем! Ваша работа готова!