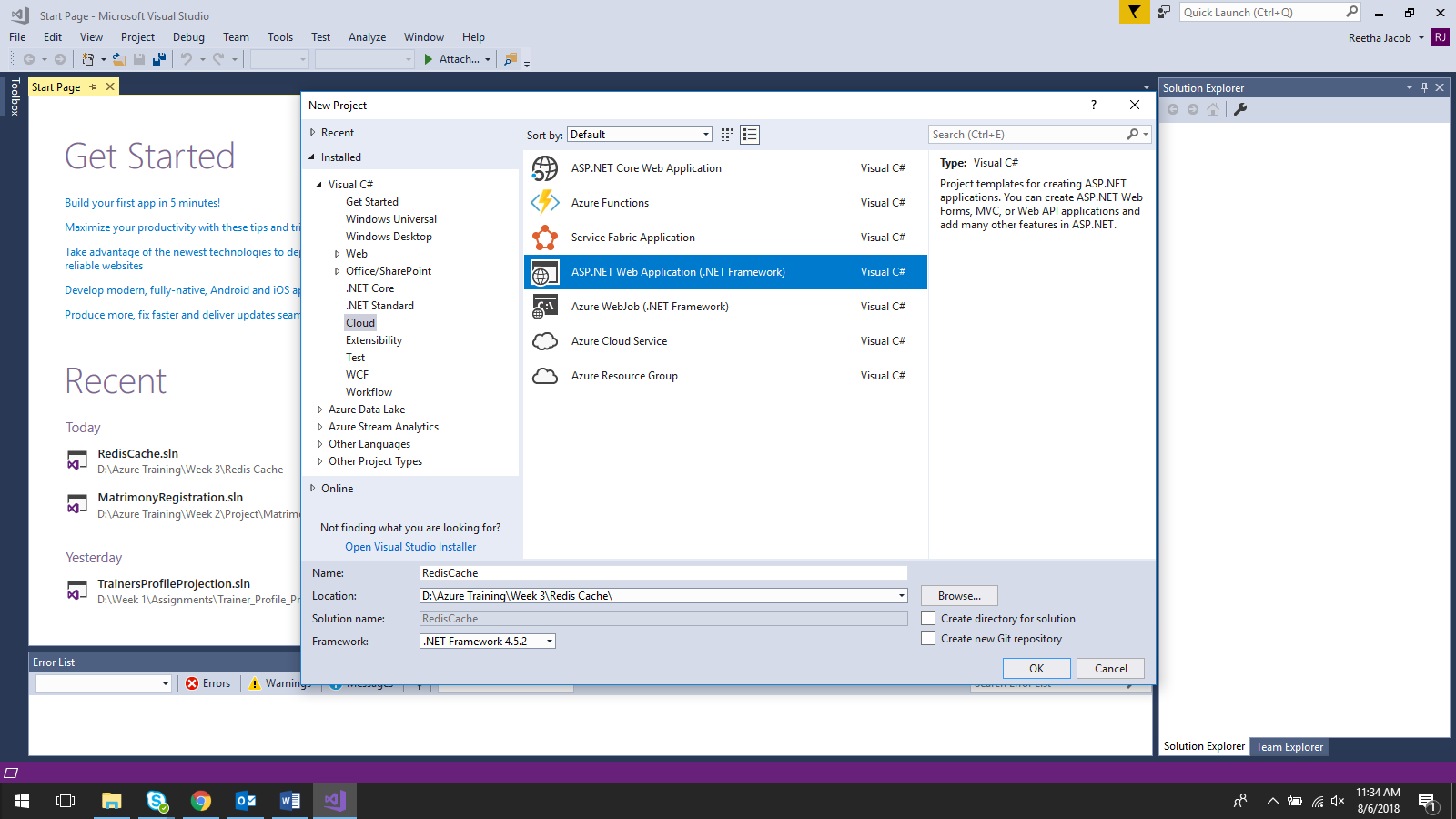
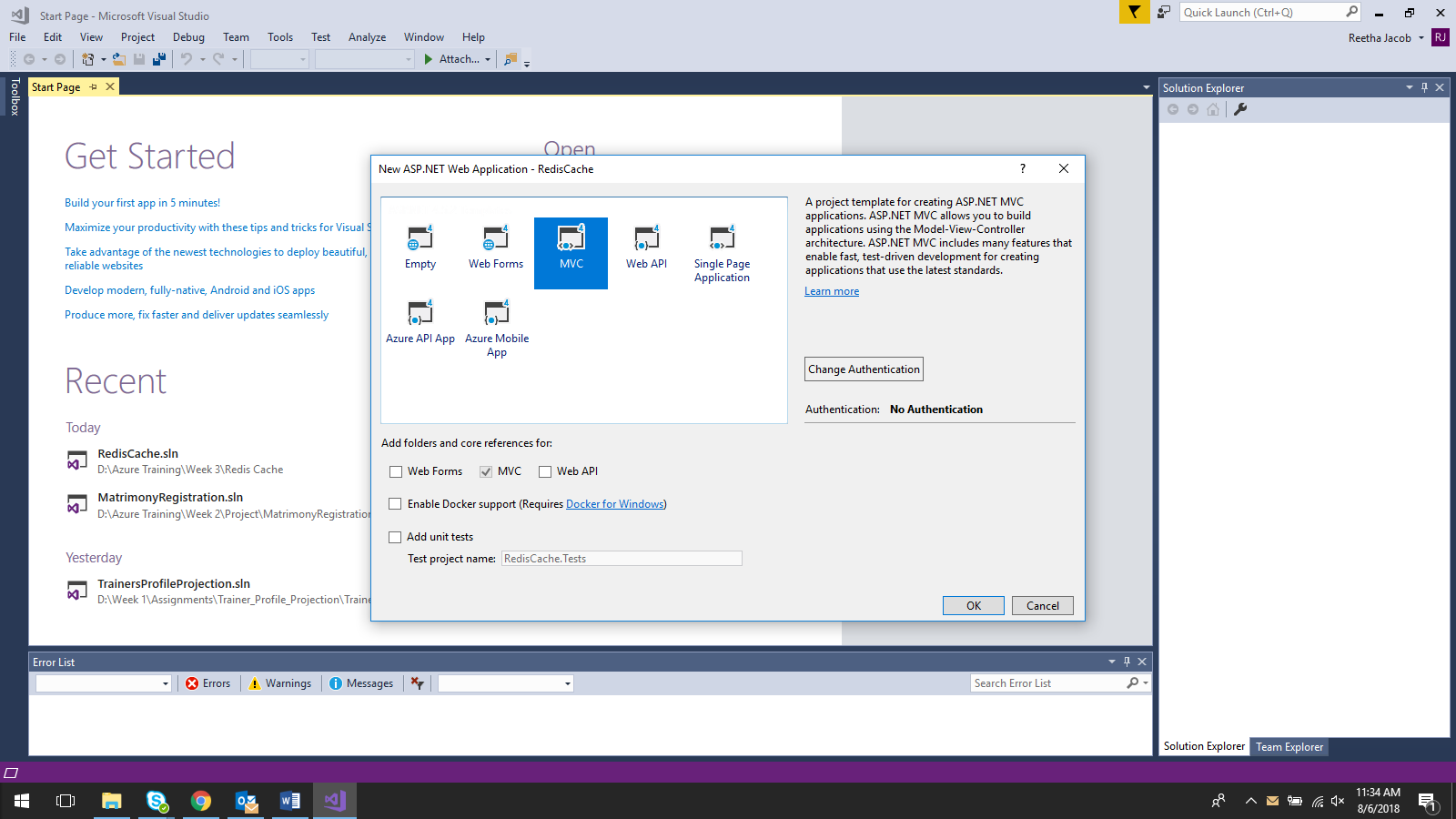
<https://docs.microsoft.com/en-us/azure/redis-cache/cache-overview>

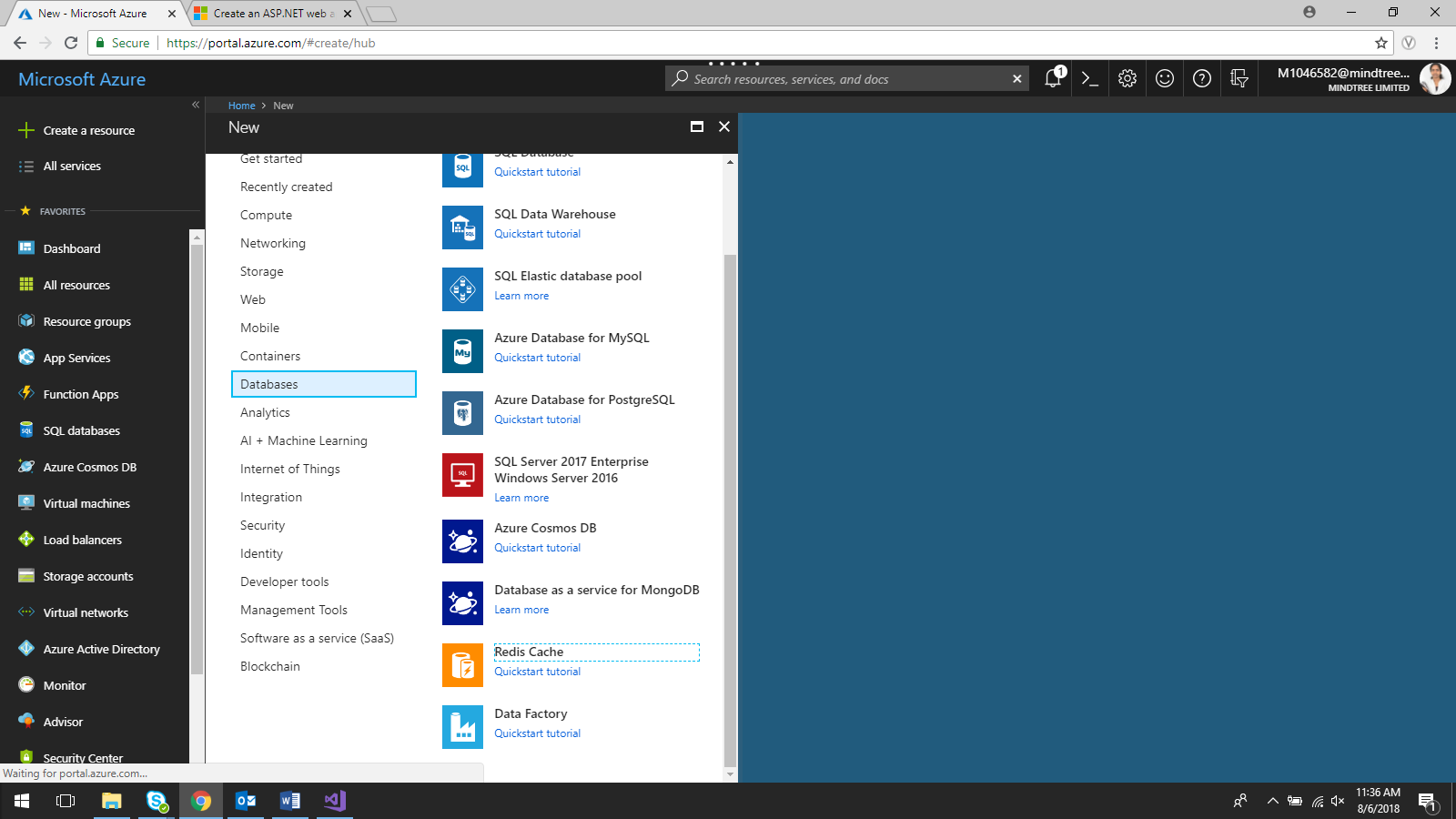
Create Empty MVC application

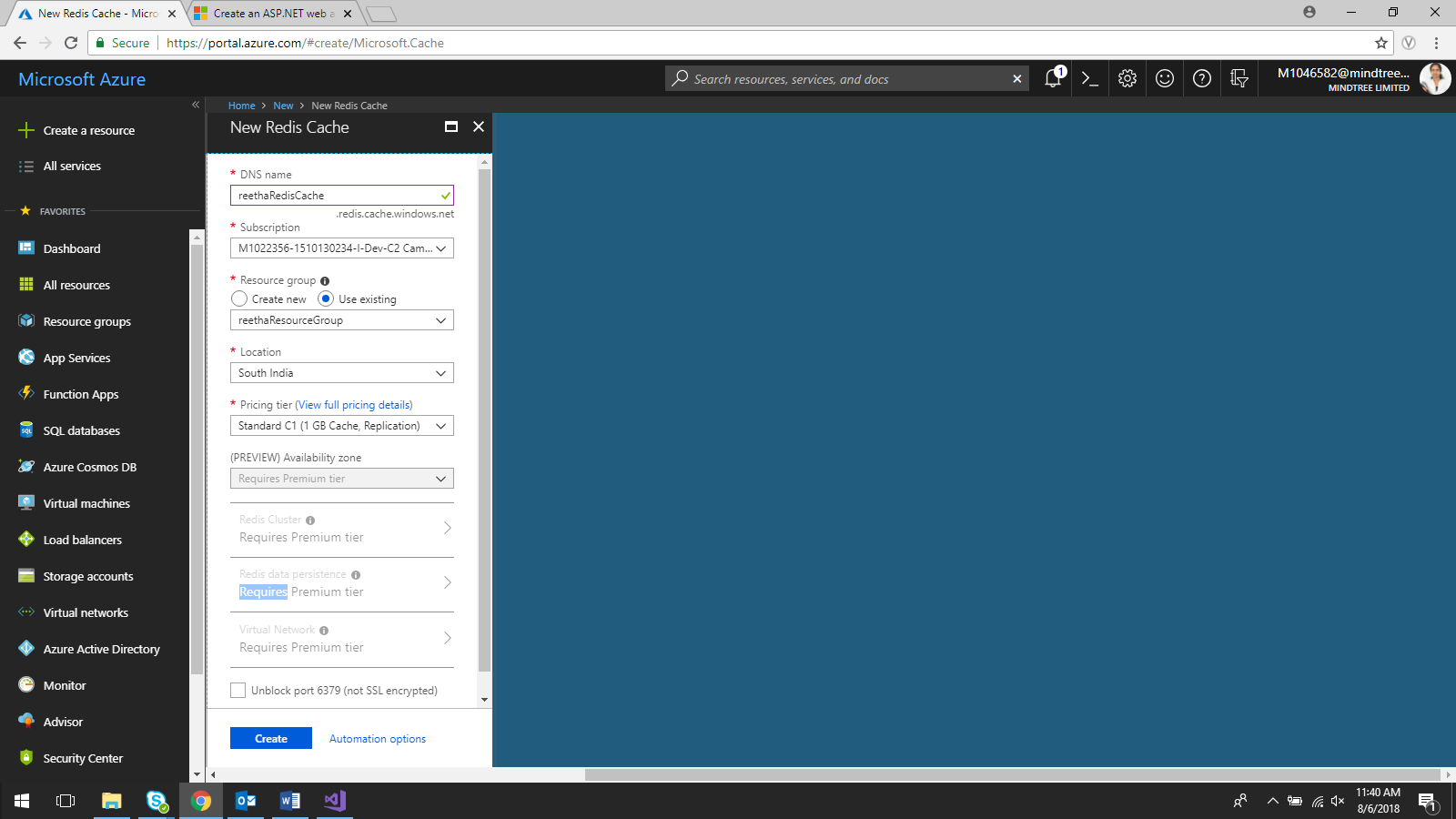




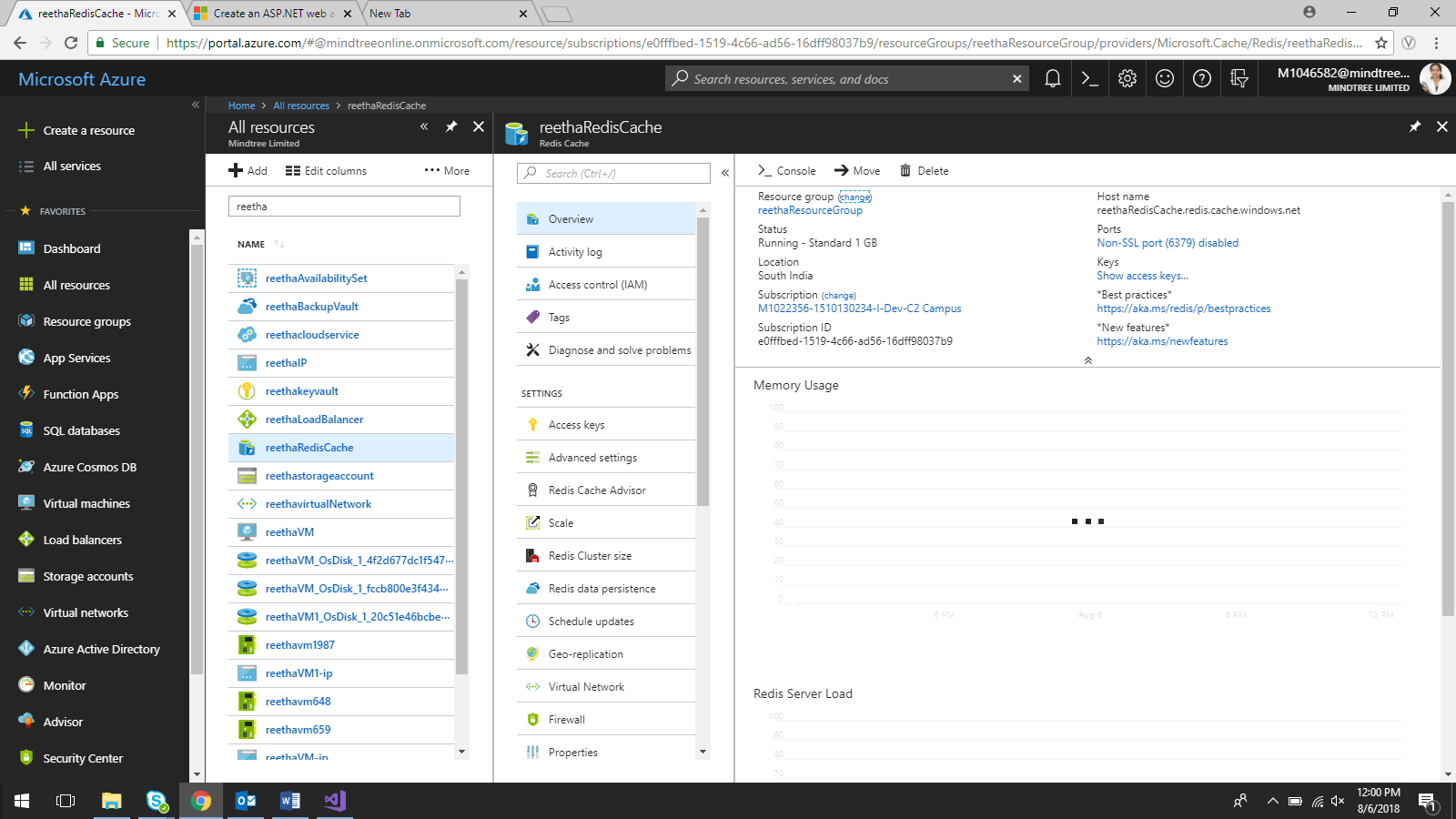
**Create a Cache**

Go to Azure Portal 🡪 Create a resource 🡪 Databases 🡪 Redis Cache





Click on Create



Retrieve host name, ports, and access keys by using the Azure portal

#### Retrieve the access keys and host name

#### 

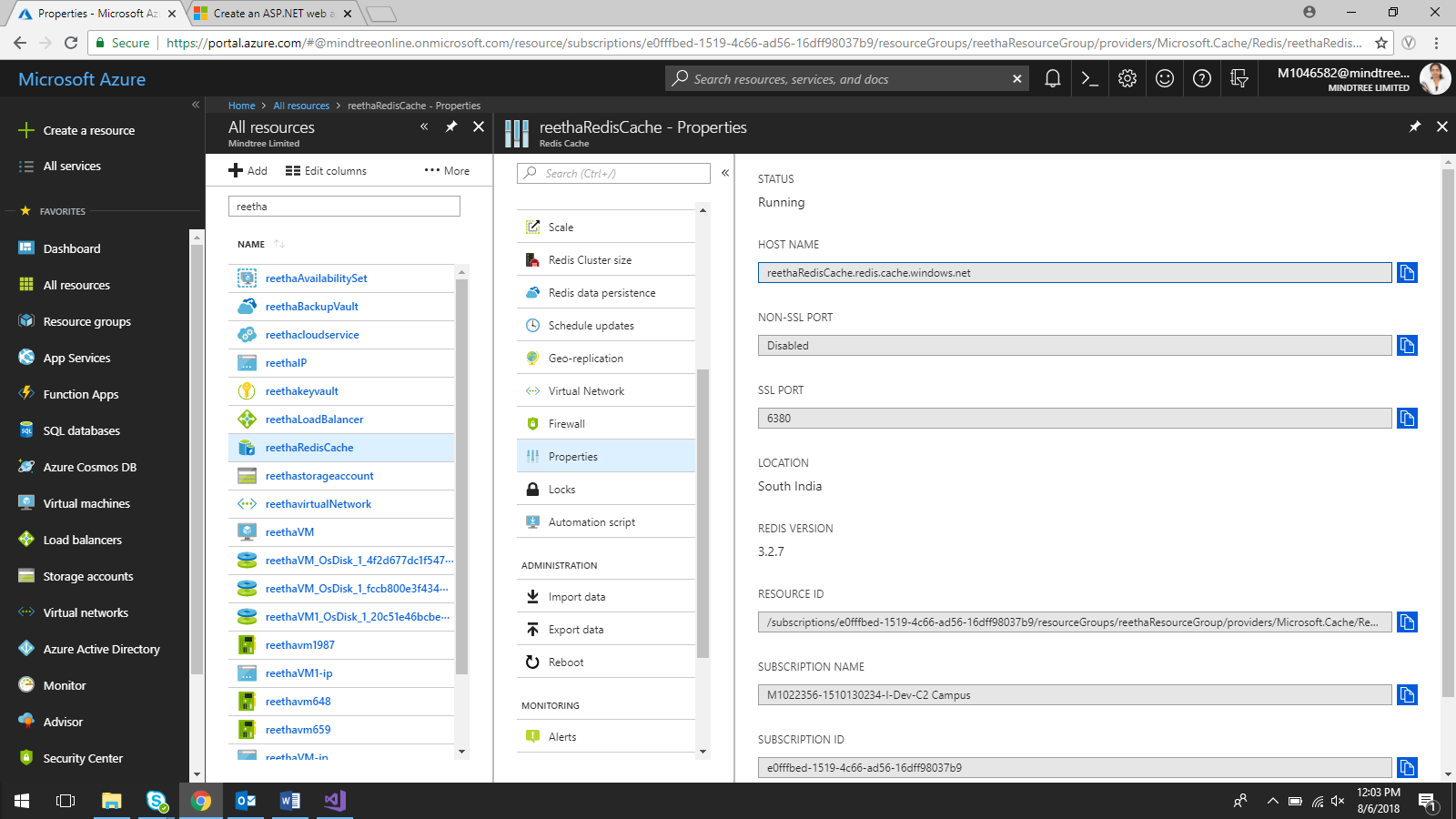
Primary: PYrQMnRaICy57s+IZZfnZ5Xv+SF3DZCVS4DVxqdON+Y=

Secondary: H7l9yHx4vdhRiTN0FZ8SIkvAuHHP9uKujWyu6h06G9c=

Primary connection string (StackExchange.Redis): reethaRedisCache.redis.cache.windows.net:6380,password=PYrQMnRaICy57s+IZZfnZ5Xv+SF3DZCVS4DVxqdON+Y=,ssl=True,abortConnect=False

Secondary connection string (StackExchange.Redis): reethaRedisCache.redis.cache.windows.net:6380,password=H7l9yHx4vdhRiTN0FZ8SIkvAuHHP9uKujWyu6h06G9c=,ssl=True,abortConnect=False

1. To retrieve the host name and ports, select **Properties**.

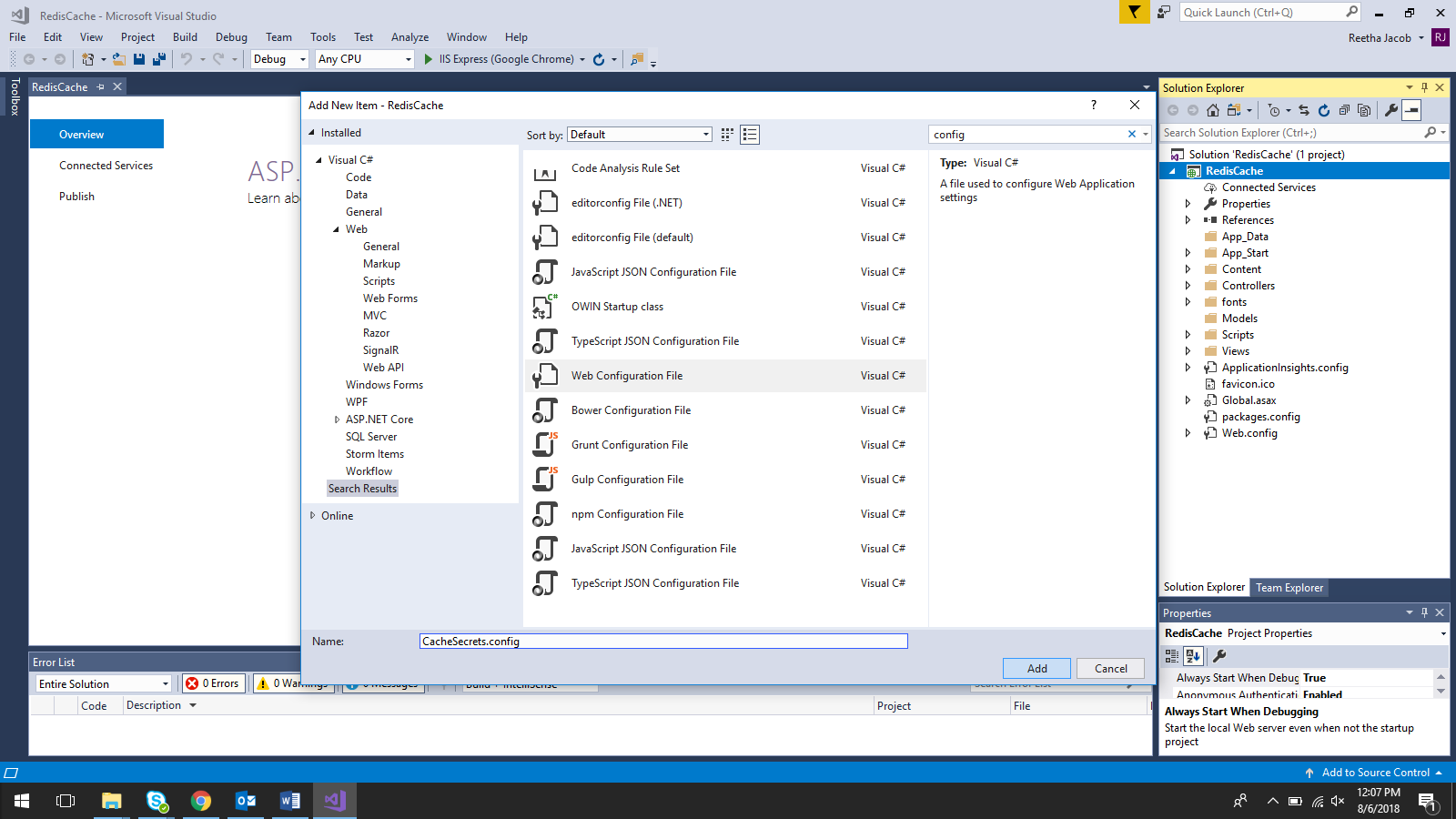


Host Name: reethaRedisCache.redis.cache.windows.net

Port: 6380

#### **To edit the CacheSecrets.config file**

Open the Project🡪 Add🡪New Item🡪 Select Web Configuration File



Edit the *CacheSecrets.config* file. Then add the following content:

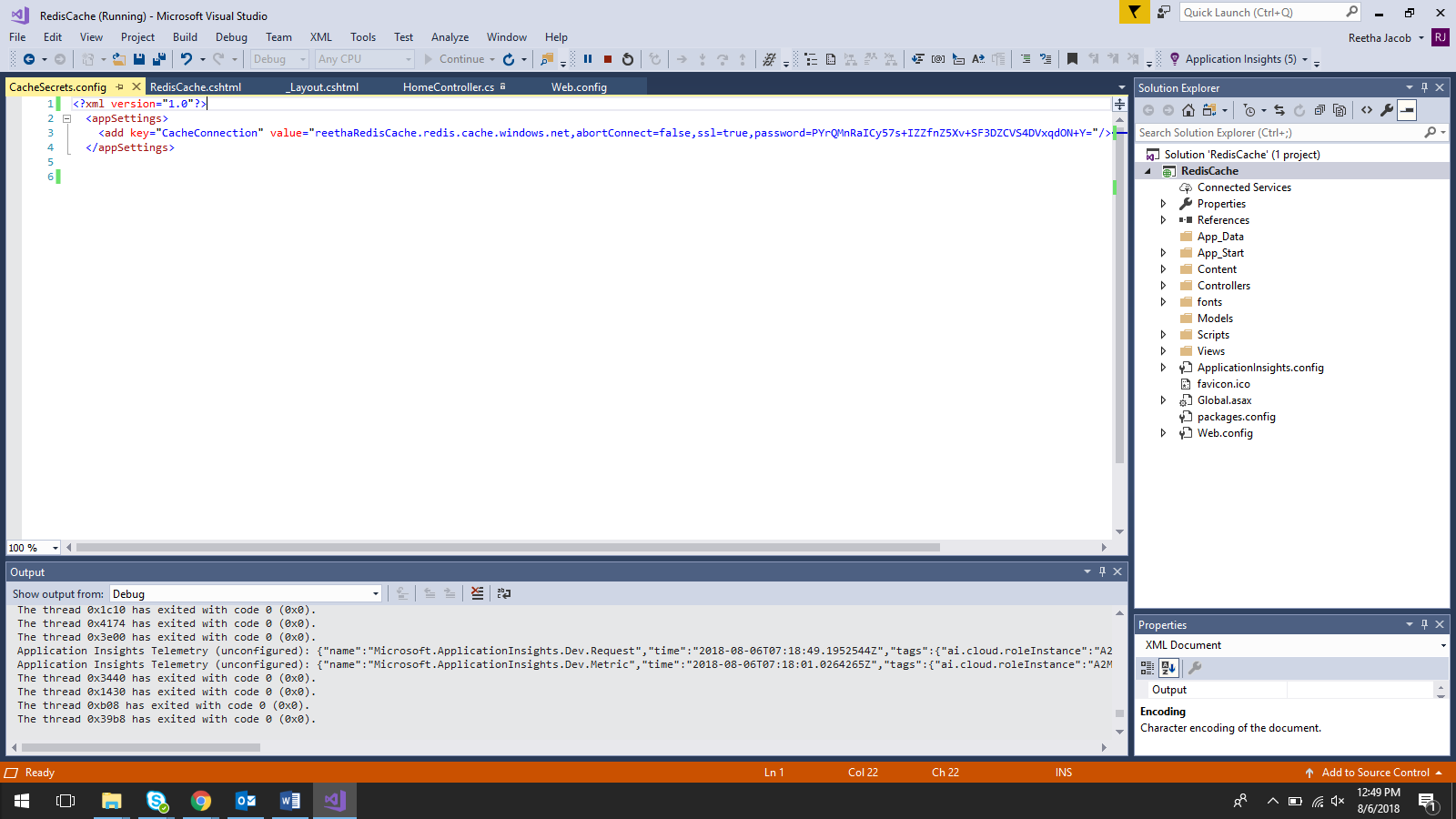
<appSettings>

<add key="CacheConnection" value="<cache-name>.redis.cache.windows.net,abortConnect=false,ssl=true,password=<access-key>"/>

</appSettings>

Replace <cache-name> with your cache host name.

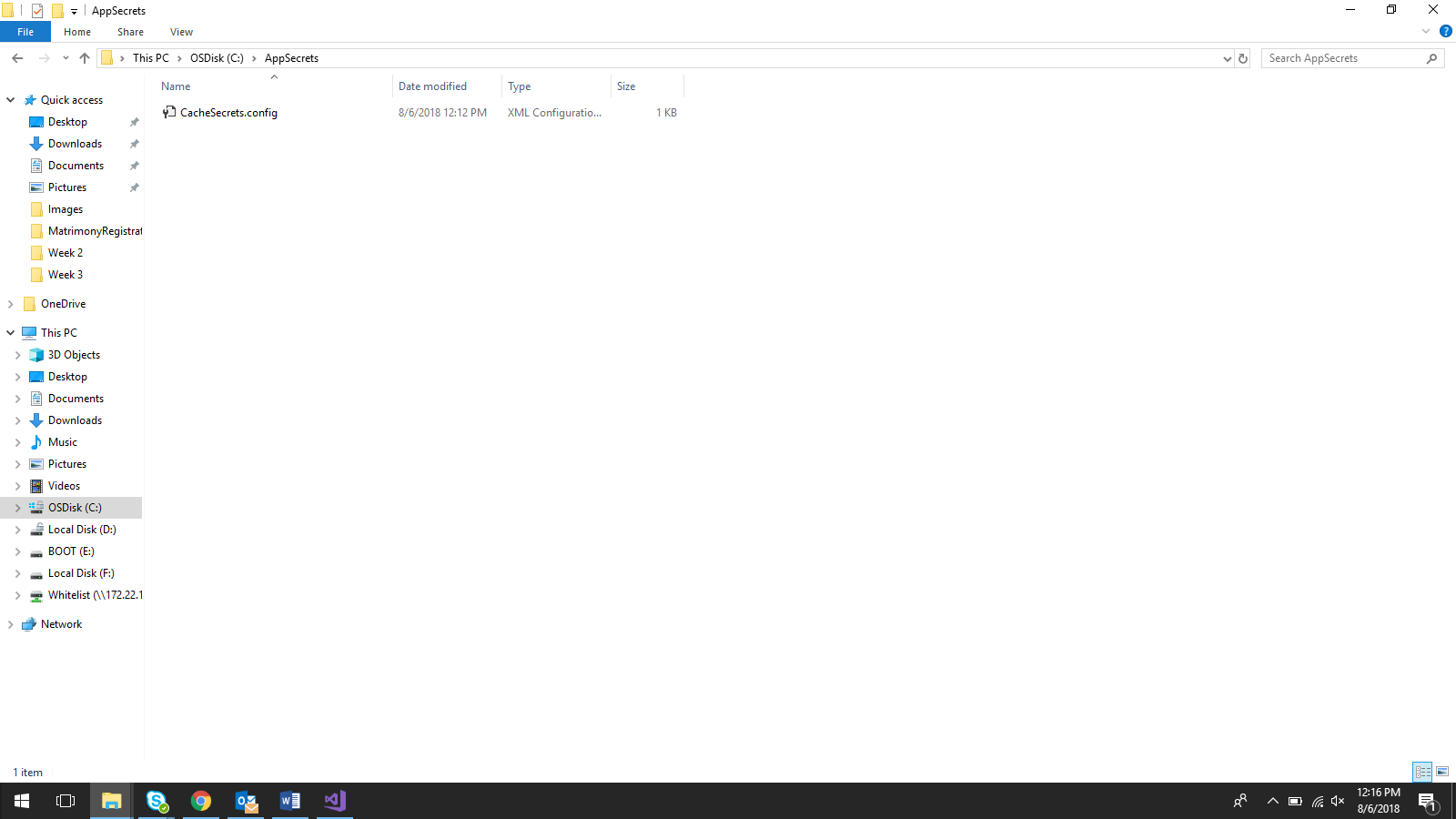
Replace <access-key> with the primary key for your cache.



Save the file

Copy the file from project path and paste to below path for security

C:\AppSecrets\CacheSecrets.config



## **Update the MVC application**

### Update the web.config file with an app setting for the cache

### Open Web.config file

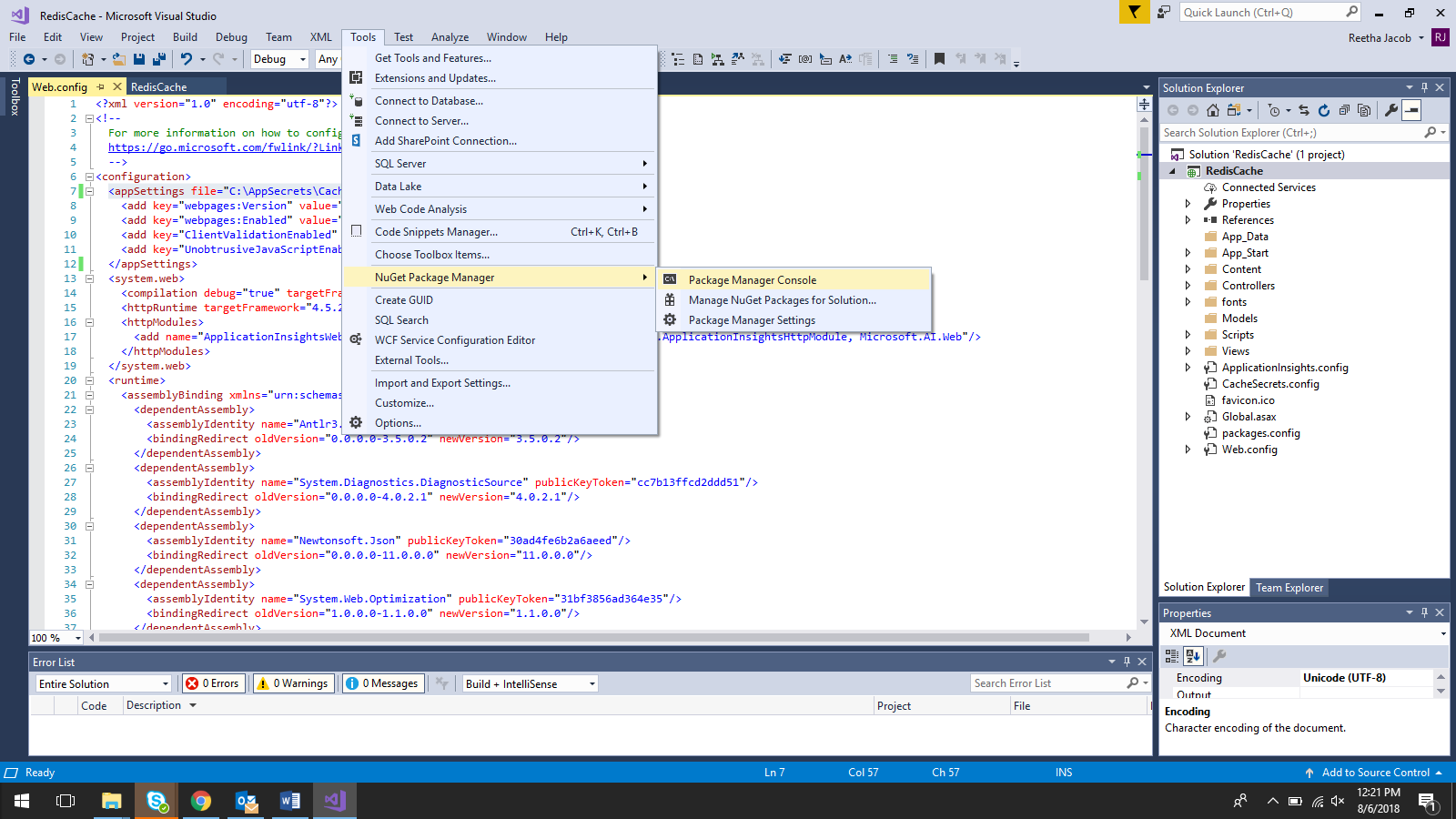
### Update the appSettings tag with the file path

### <appSettings file="C:\AppSecrets\CacheSecrets.config">

### 

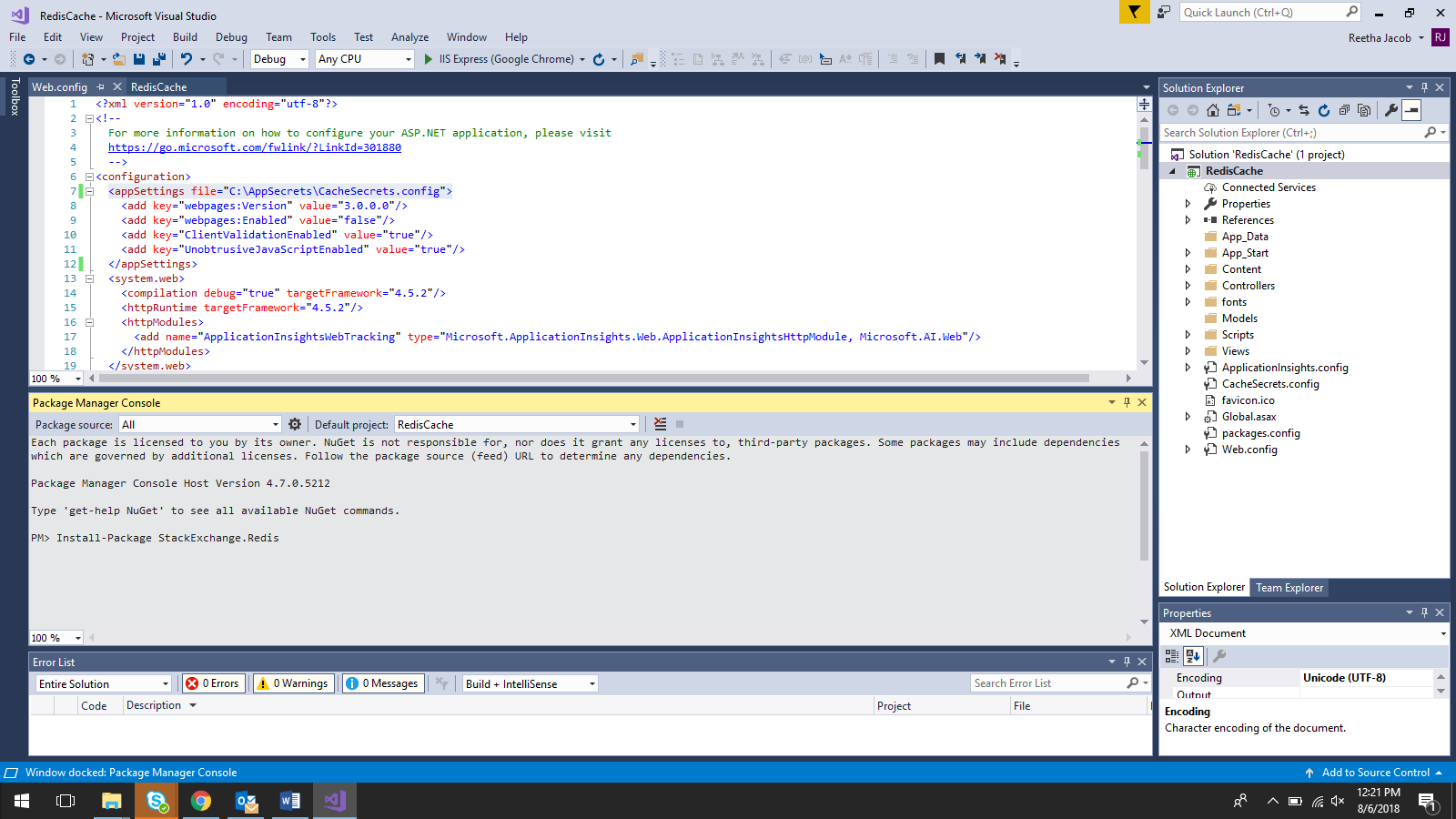
### To configure the application to use StackExchange.Redis

Open VS, select Tools > NuGet Package Manager > Package Manager Console.



Run the below command

Install-Package StackExchange.Redis



The NuGet package downloads and adds the required assembly references for your client application to access Azure Redis Cache with the StackExchange.Redis cache client. If you prefer to use a strong-named version of the StackExchange.Redis client library, install the StackExchange.Redis.StrongName package.

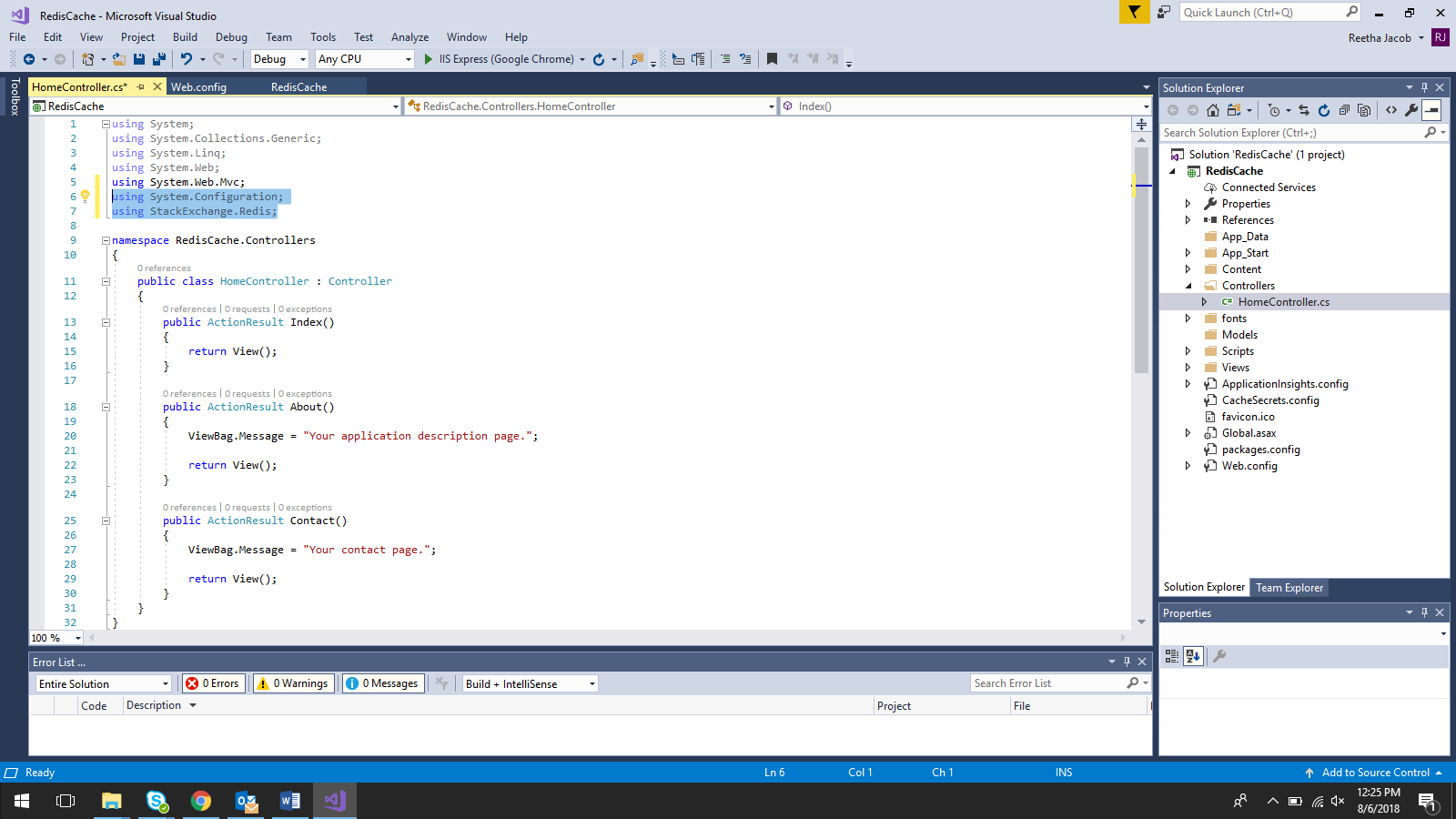
### To update the HomeController and Layout

Open HomeController.cs

Add the below namespaces

**using System.Configuration;**

**using StackExchange.Redis;**



Add the following method to the HomeController class to support a new RedisCache action that runs some commands against the new cache.

public ActionResult RedisCache()

{

ViewBag.Message = "A simple example with Azure Redis Cache on ASP.NET.";

var lazyConnection = new Lazy<ConnectionMultiplexer>(() =>

{

string cacheConnection = ConfigurationManager.AppSettings["CacheConnection"].ToString();

return ConnectionMultiplexer.Connect(cacheConnection);

});

// Connection refers to a property that returns a ConnectionMultiplexer

// as shown in the previous example.

IDatabase cache = lazyConnection.Value.GetDatabase();

// Perform cache operations using the cache object...

// Simple PING command

ViewBag.command1 = "PING";

ViewBag.command1Result = cache.Execute(ViewBag.command1).ToString();

// Simple get and put of integral data types into the cache

ViewBag.command2 = "GET Message";

ViewBag.command2Result = cache.StringGet("Message").ToString();

ViewBag.command3 = "SET Message \"Hello! The cache is working from ASP.NET!\"";

ViewBag.command3Result = cache.StringSet("Message", "Hello! The cache is working from ASP.NET!").ToString();

// Demostrate "SET Message" executed as expected...

ViewBag.command4 = "GET Message";

ViewBag.command4Result = cache.StringGet("Message").ToString();

// Get the client list, useful to see if connection list is growing...

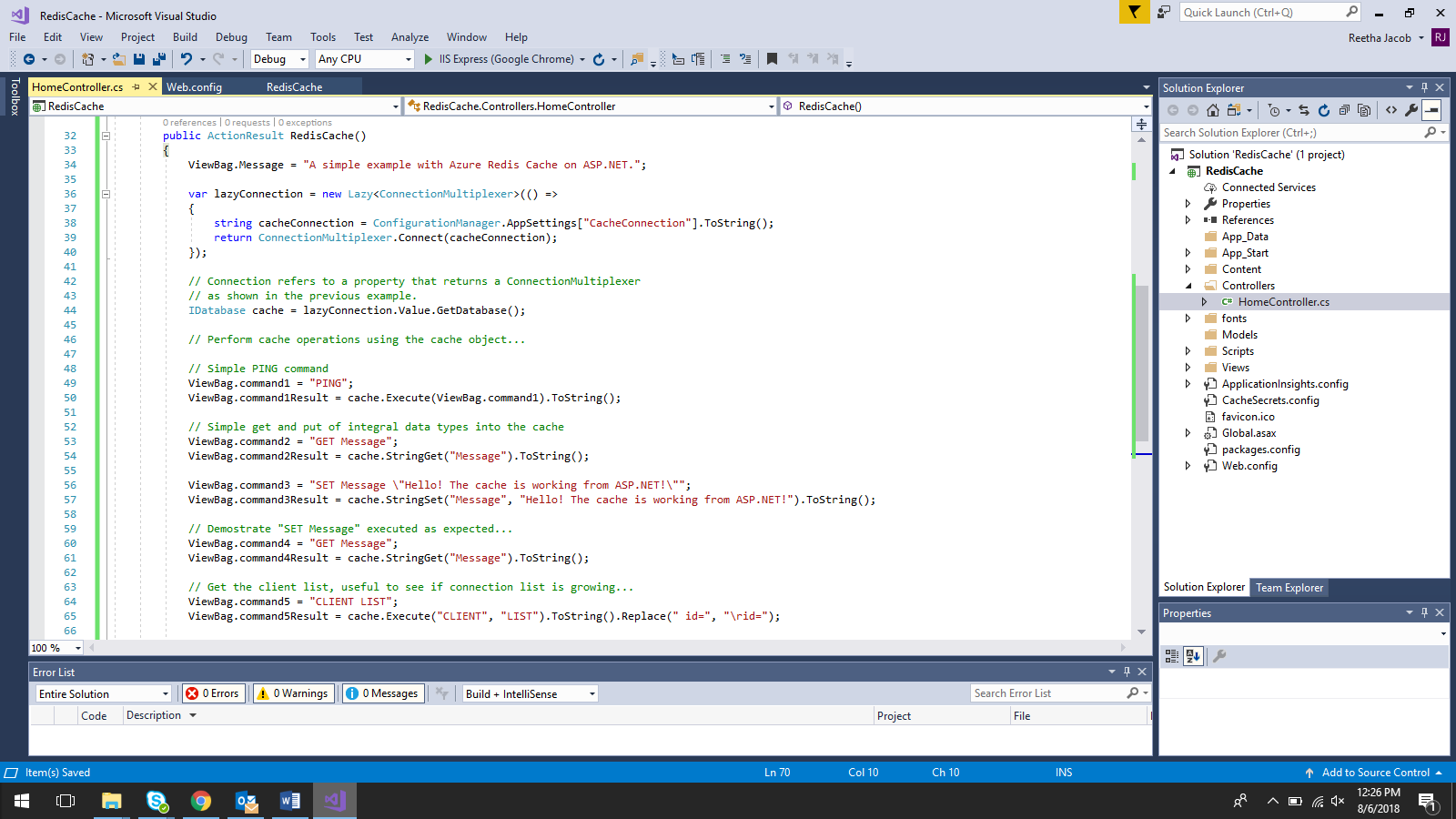
ViewBag.command5 = "CLIENT LIST";

ViewBag.command5Result = cache.Execute("CLIENT", "LIST").ToString().Replace(" id=", "\rid=");

lazyConnection.Value.Dispose();

return View();

}



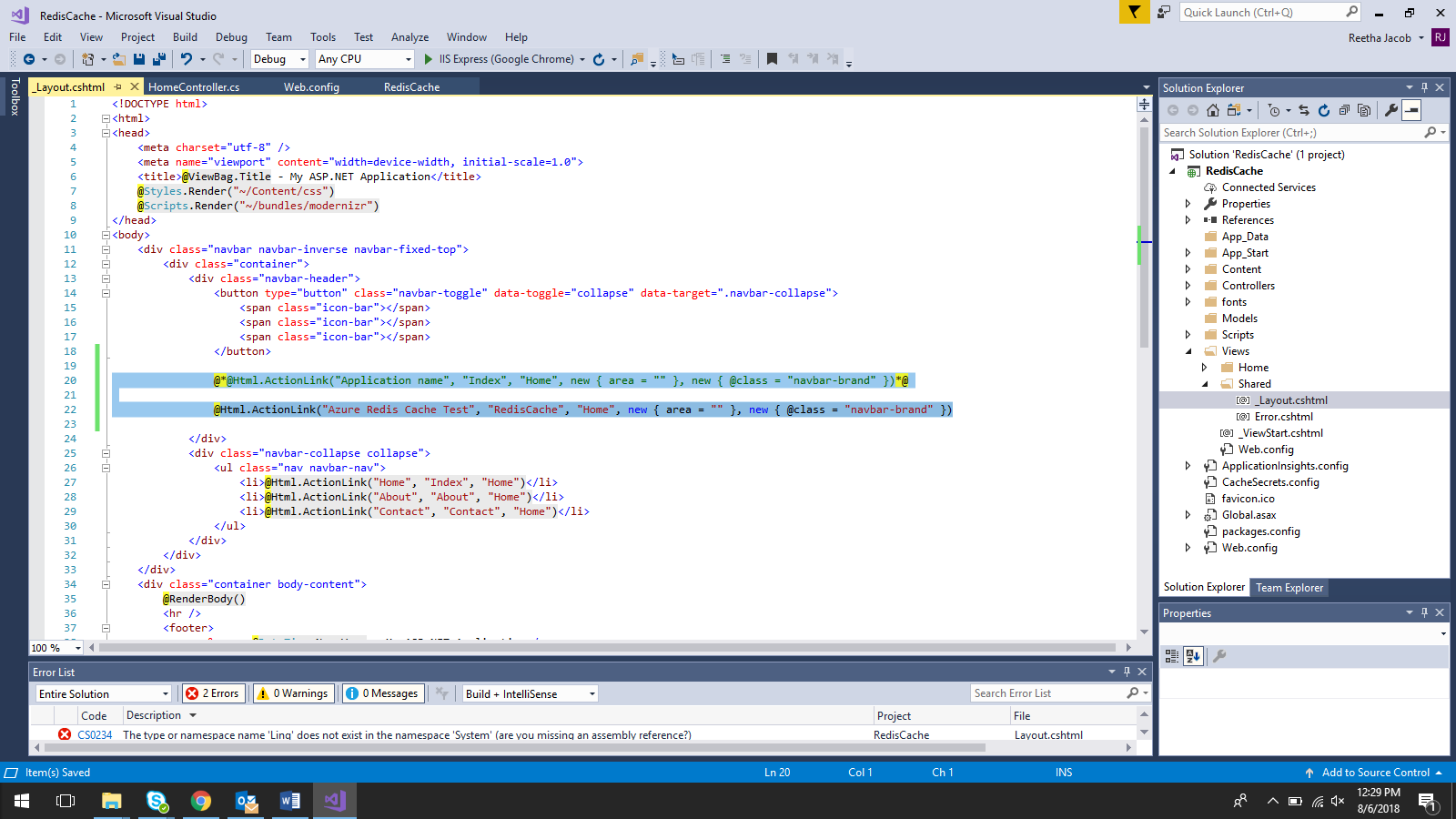
Go to View🡪Shared🡪\_Layout.cshtml

Replace

@Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })

With

@Html.ActionLink("Azure Redis Cache Test", "RedisCache", "Home", new { area = "" }, new { @class = "navbar-brand" })



### To add a new RedisCache view

### Add new View(RedisCache) in Home

### 

### Replace the code in the RedisCache.cshtml file with the following code:

@{

ViewBag.Title = "Azure Redis Cache Test";

}

<h2>@ViewBag.Title.</h2>

<h3>@ViewBag.Message</h3>

<br /><br />

<table border="1" cellpadding="10">

<tr>

<th>Command</th>

<th>Result</th>

</tr>

<tr>

<td>@ViewBag.command1</td>

<td><pre>@ViewBag.command1Result</pre></td>

</tr>

<tr>

<td>@ViewBag.command2</td>

<td><pre>@ViewBag.command2Result</pre></td>

</tr>

<tr>

<td>@ViewBag.command3</td>

<td><pre>@ViewBag.command3Result</pre></td>

</tr>

<tr>

<td>@ViewBag.command4</td>

<td><pre>@ViewBag.command4Result</pre></td>

</tr>

<tr>

<td>@ViewBag.command5</td>

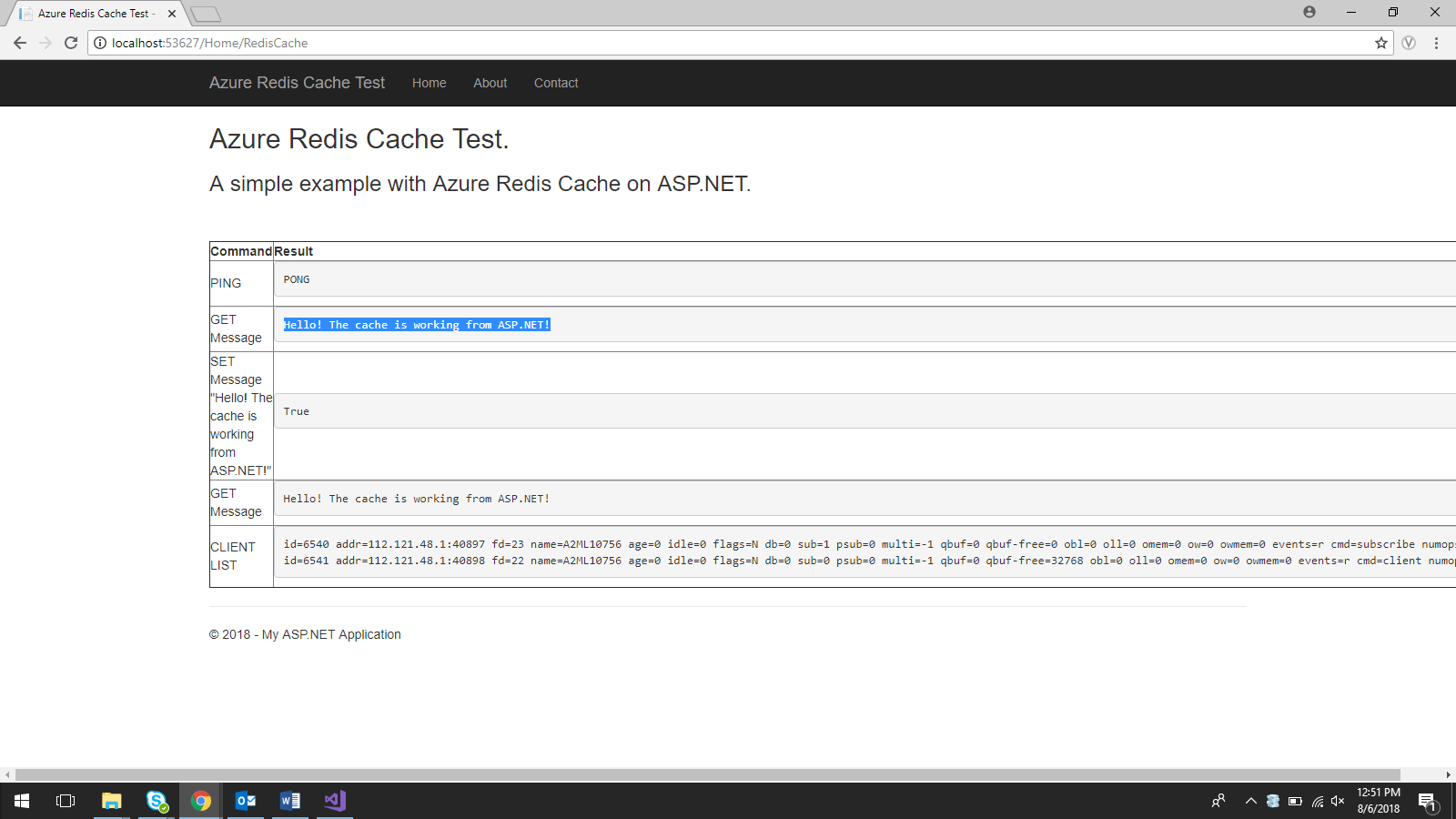
<td><pre>@ViewBag.command5Result</pre></td>

</tr>

### </table>

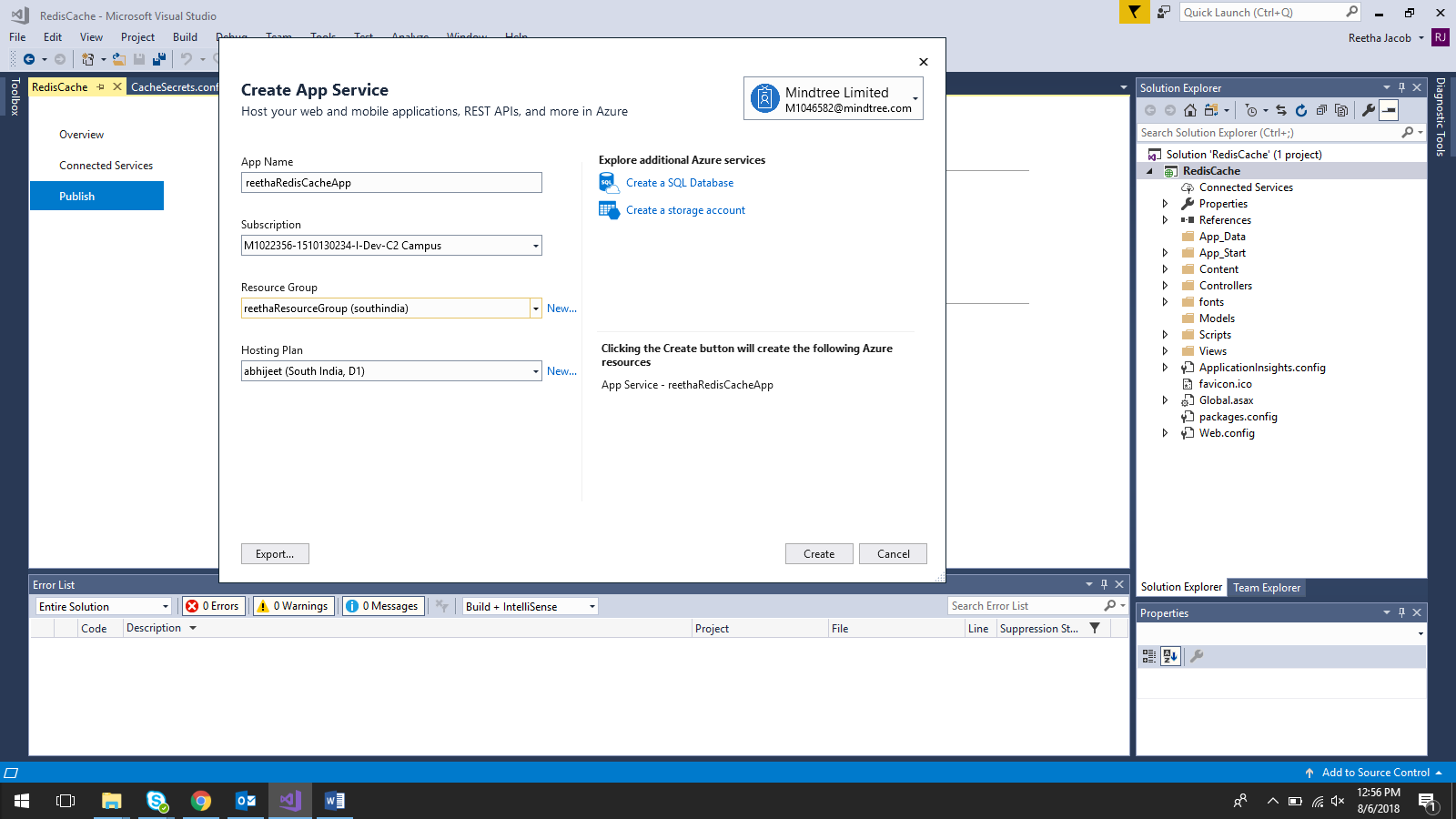
### 

Run the application

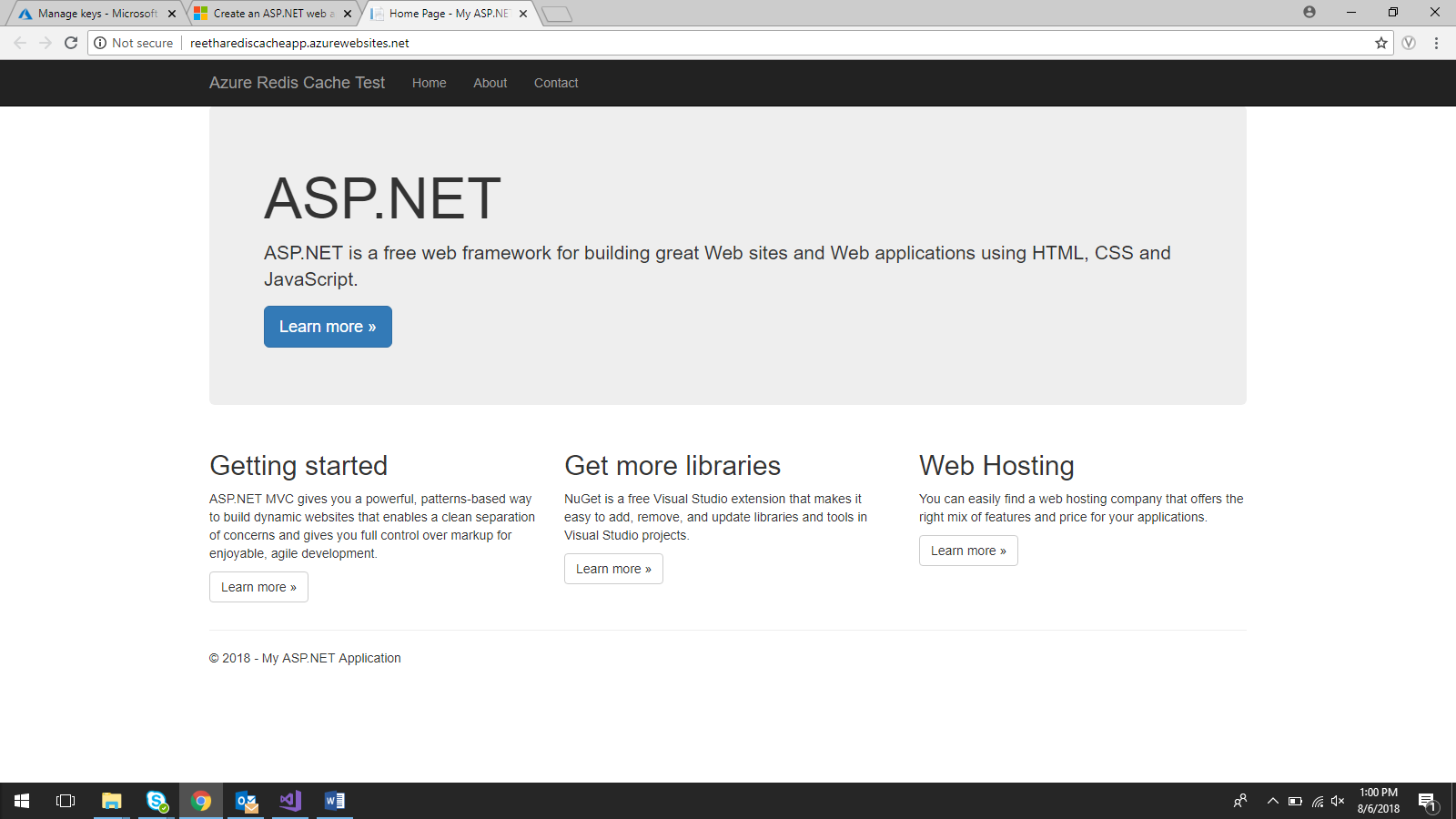


First GET Message is populating from Cache.

**Host the Application in AppService**



<http://reetharediscacheapp.azurewebsites.net/>



### Add the app setting for the cache

### Go to Azure Portal🡪 search for

### 

### 

### Go to AppSettings🡪Add new CacheConnection

### 

### Save and Run the application

### 