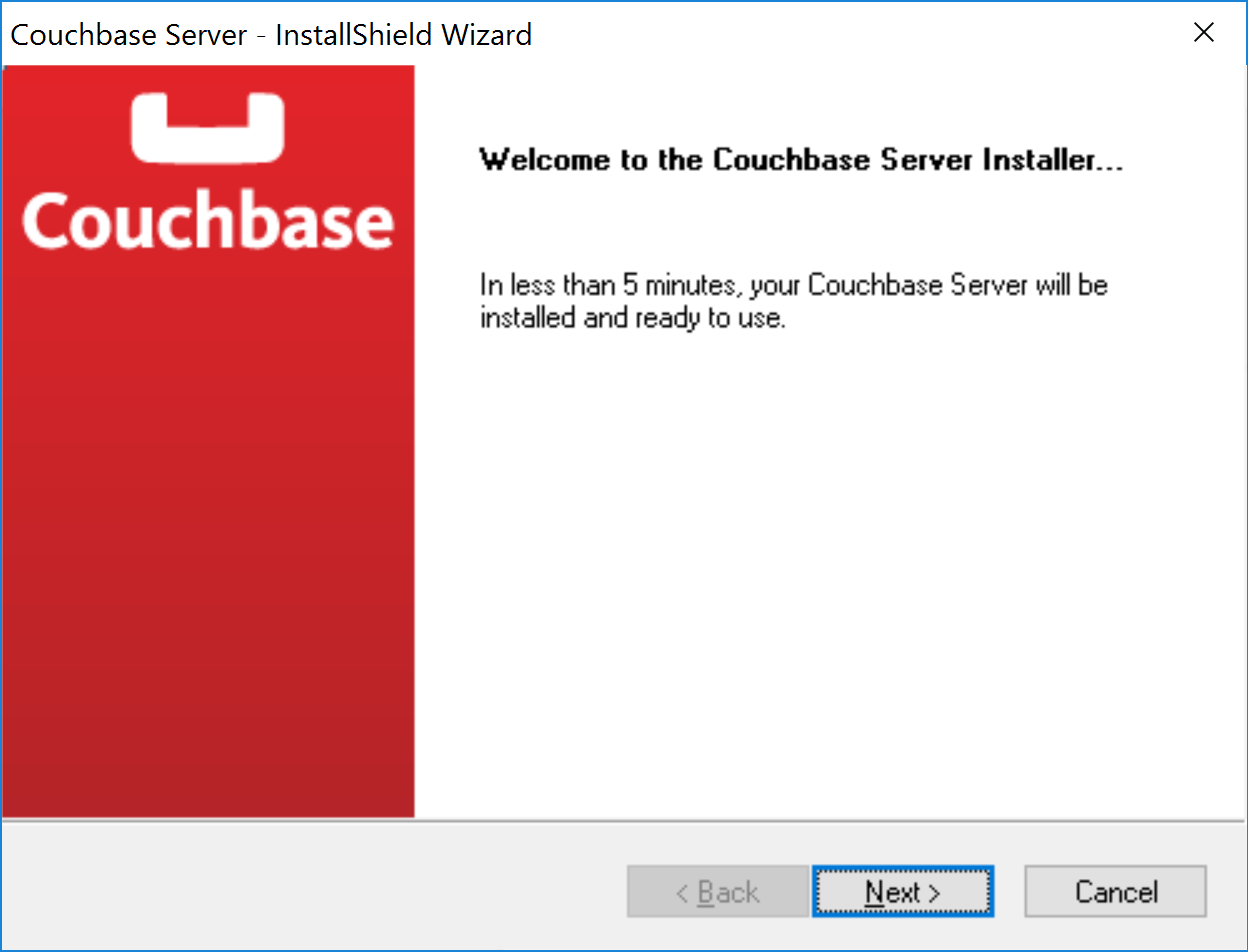
**Installing and Using Couchbase Caching Server**:

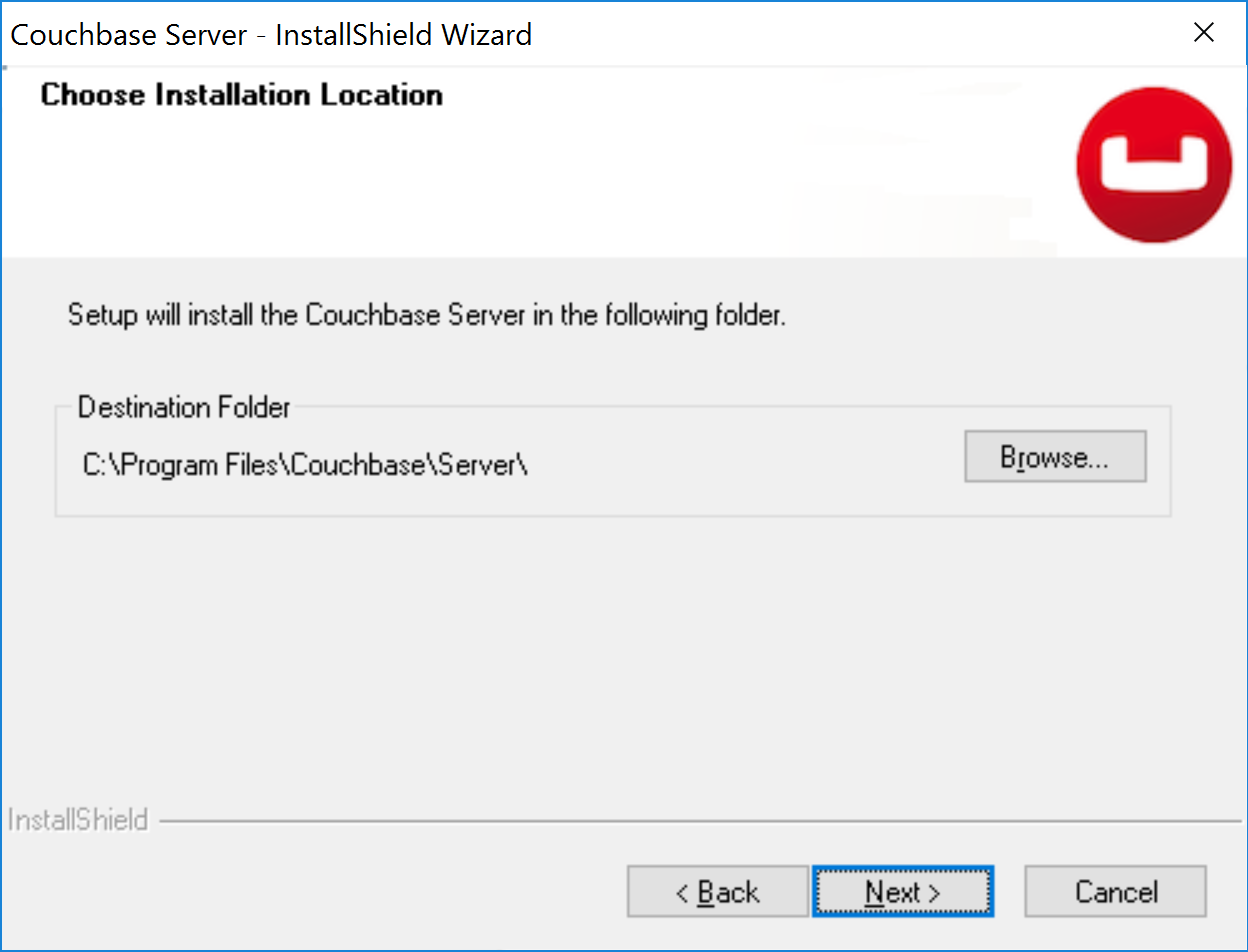
Download the latest Couchbase server from the following url: <http://www.couchbase.com/>

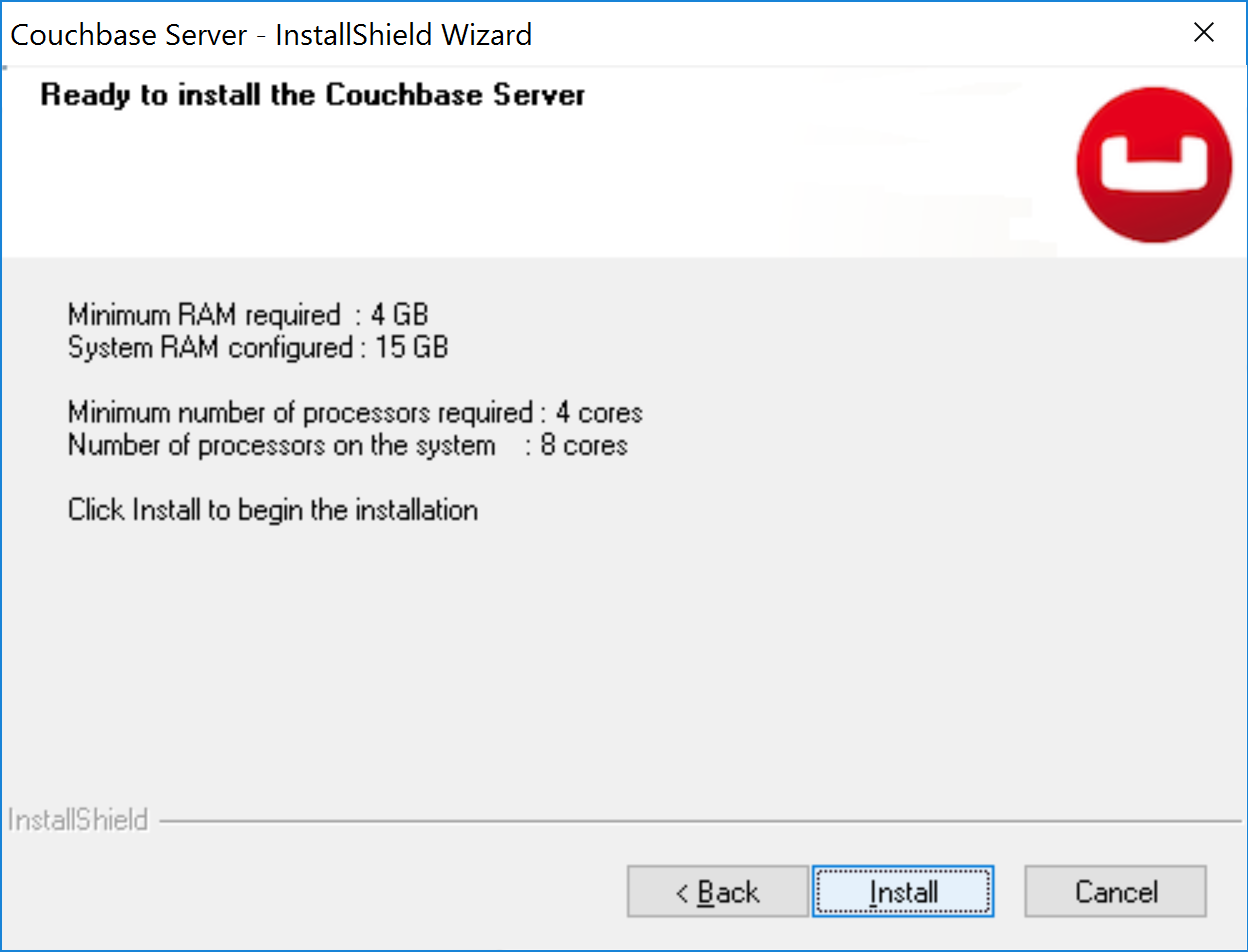
There will be a button on the right hand side for download. Once you click on the download link, choose the appropriate version e.g., if you are using windows, then click on the button next to windows as shown below. The community edition is free so select it before downloading Couchbase.

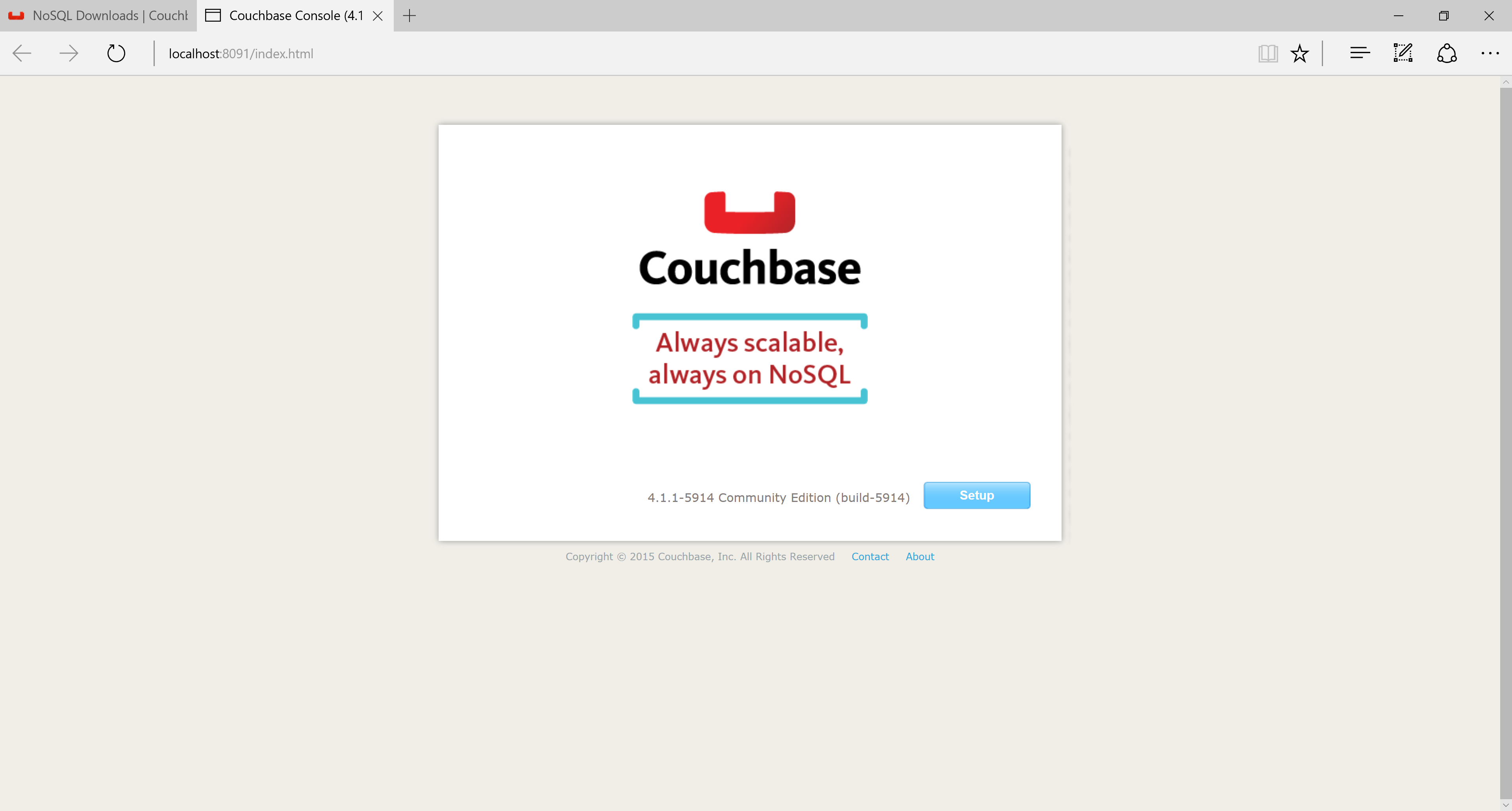


Once download is complete, it will start the installation as shown below. Follow the following steps.

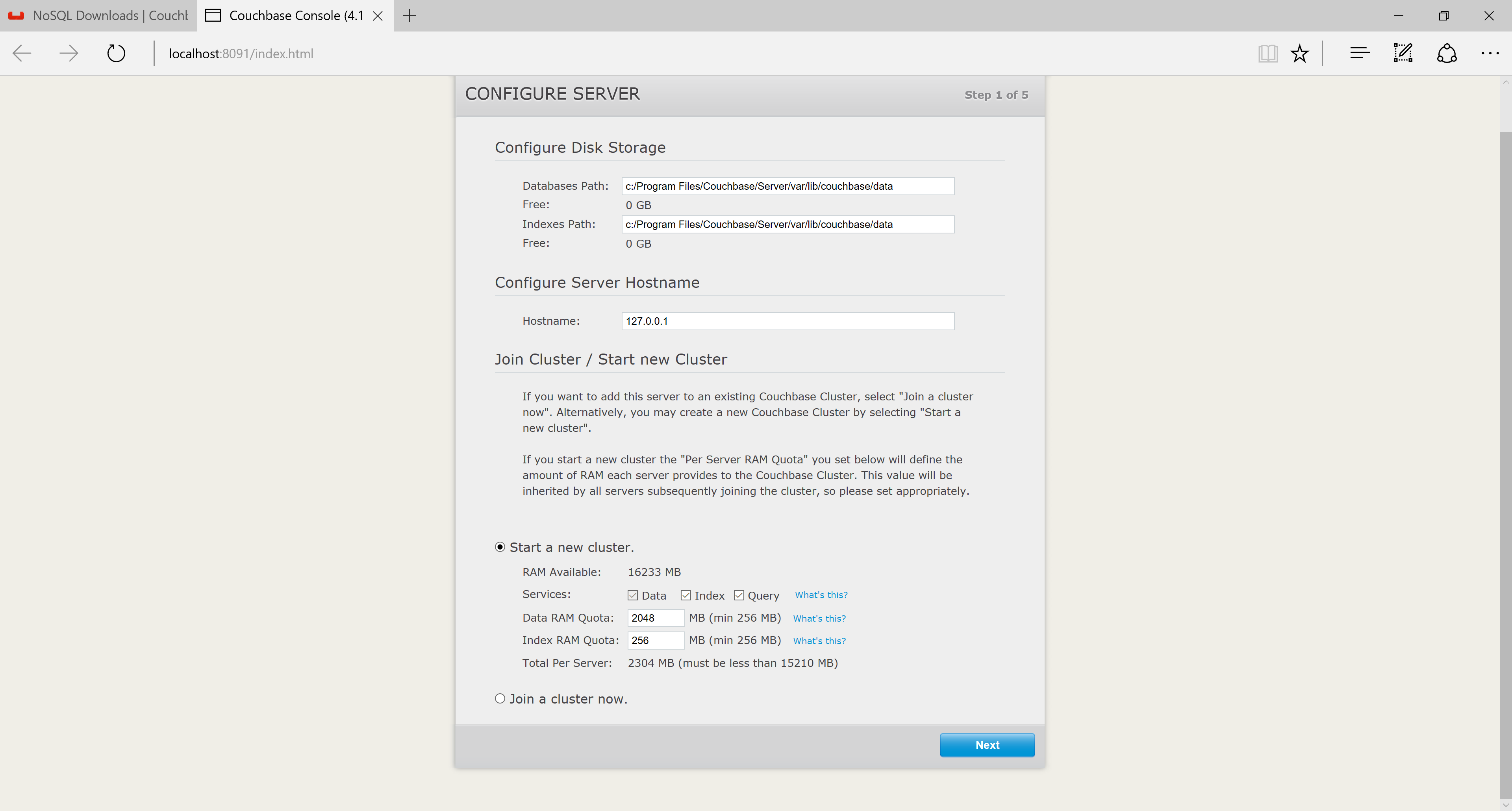


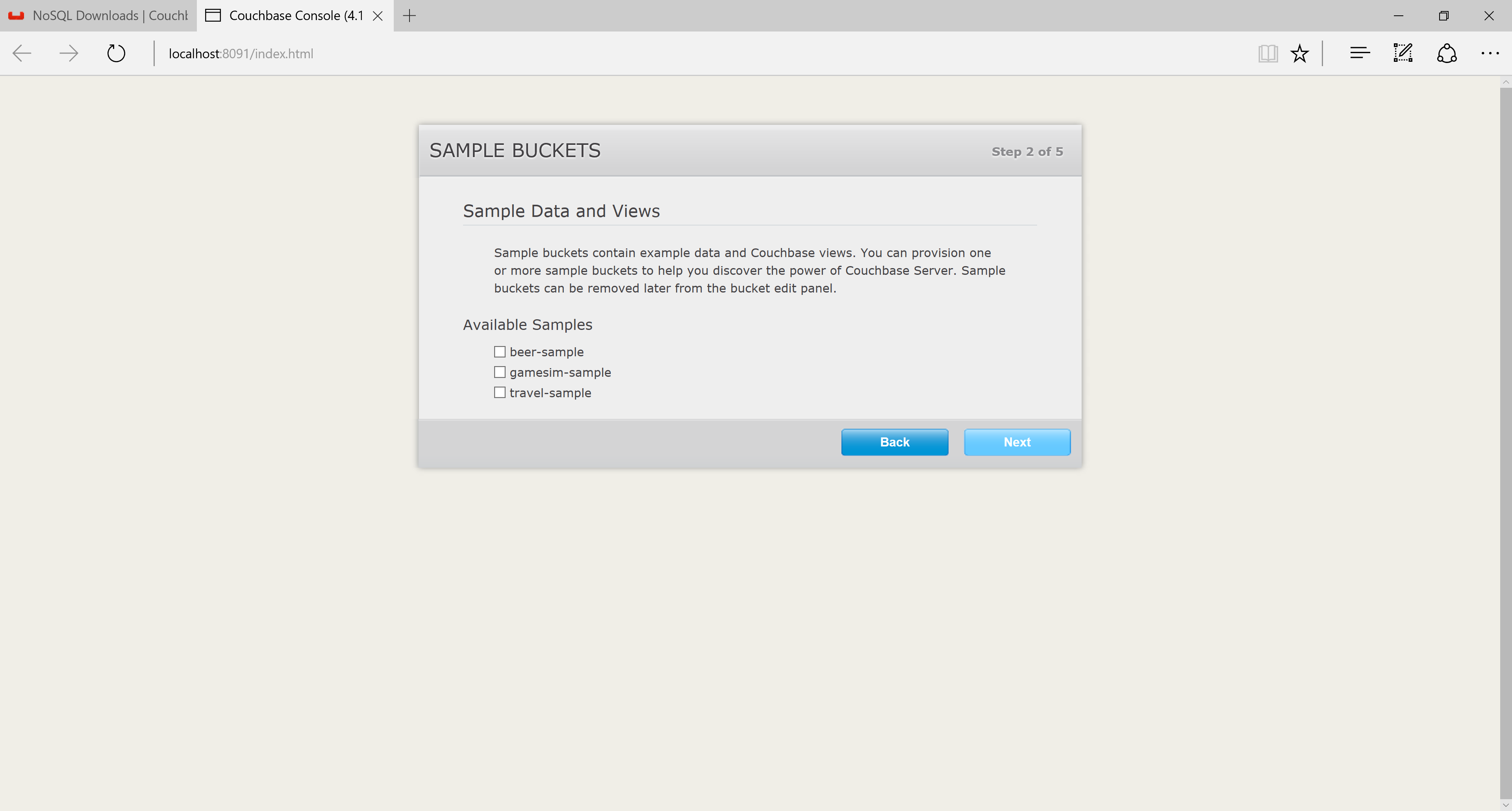






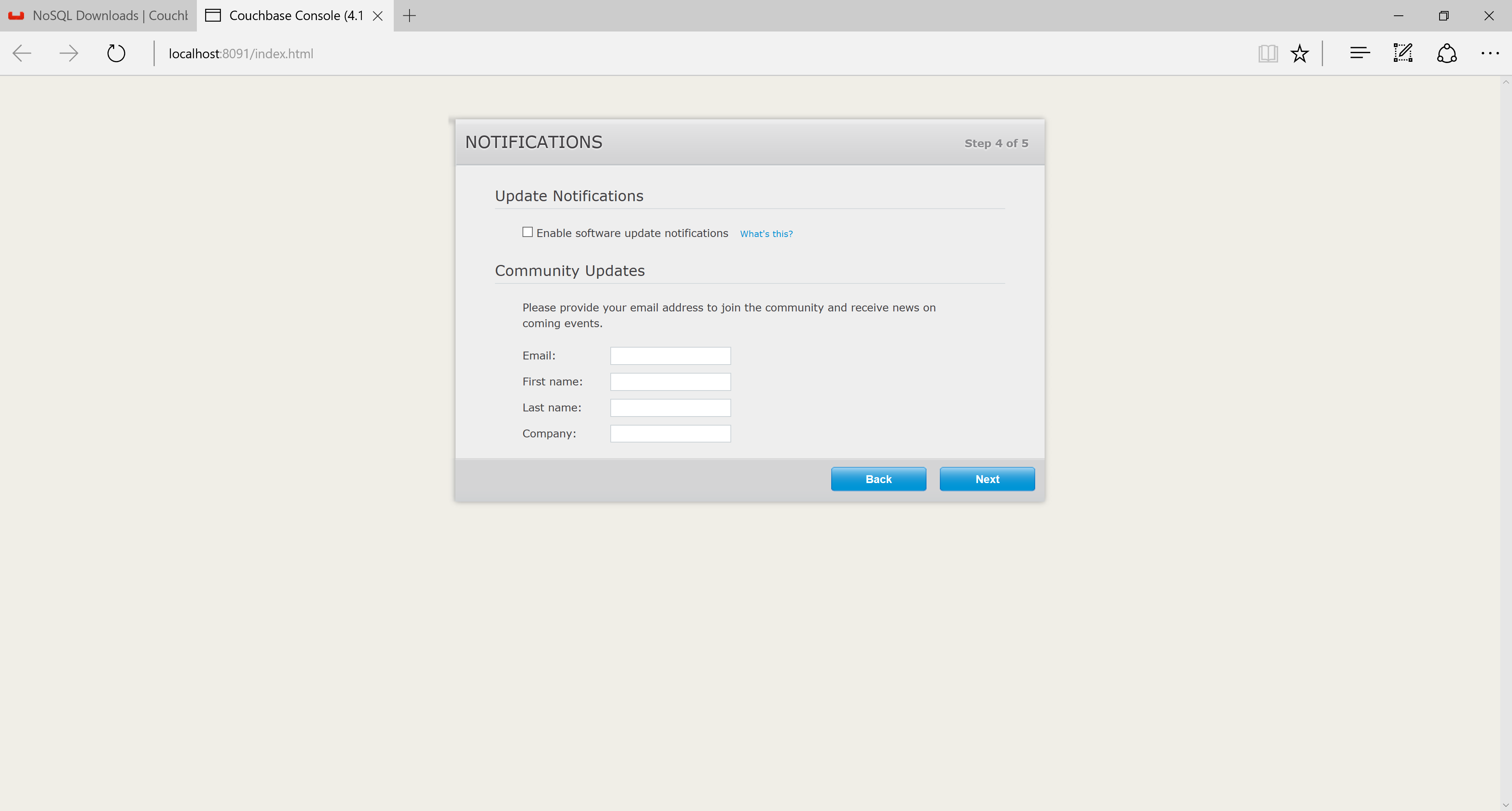
Change the Data Ram Quota to 2048 as show below.



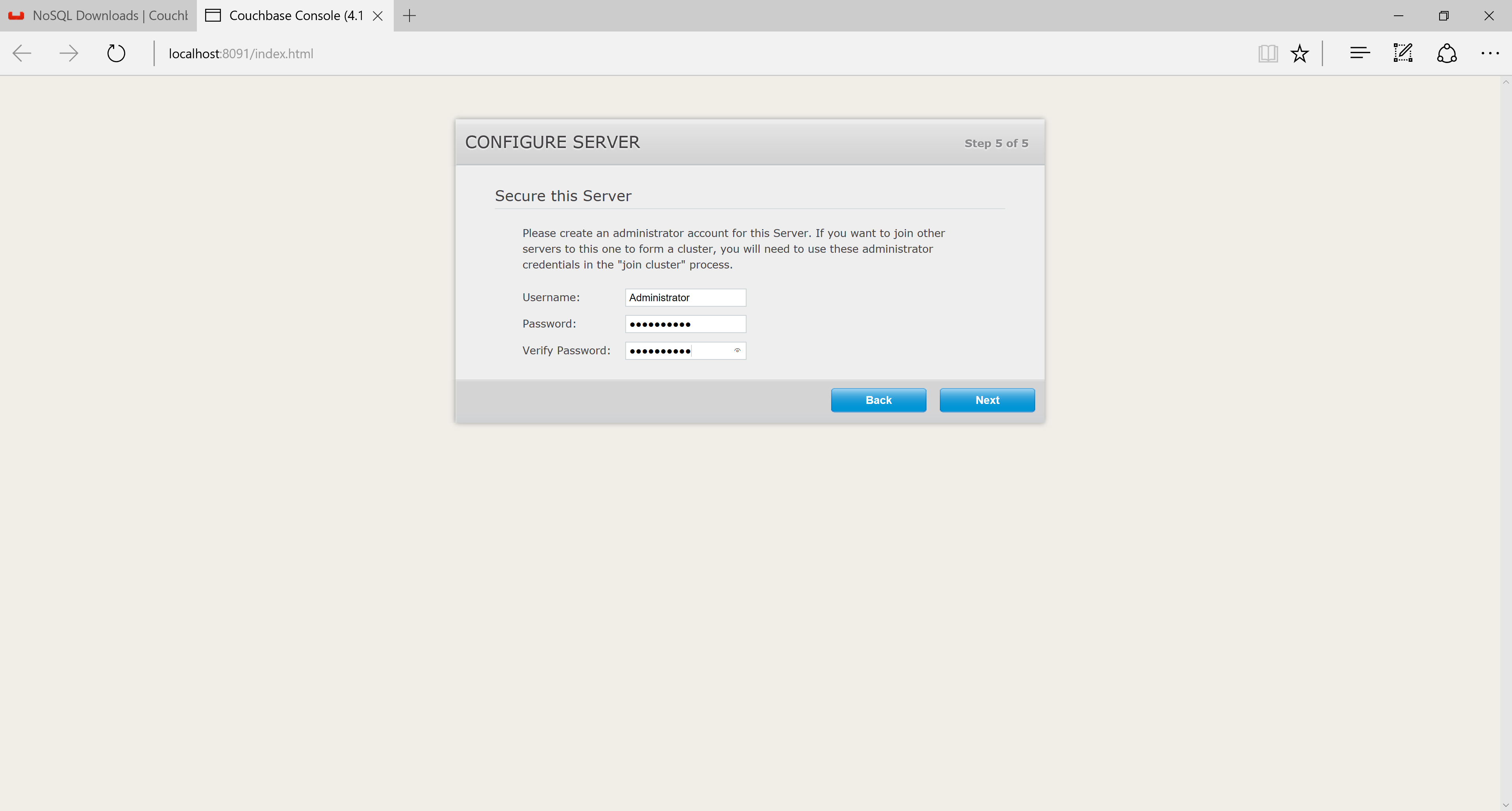


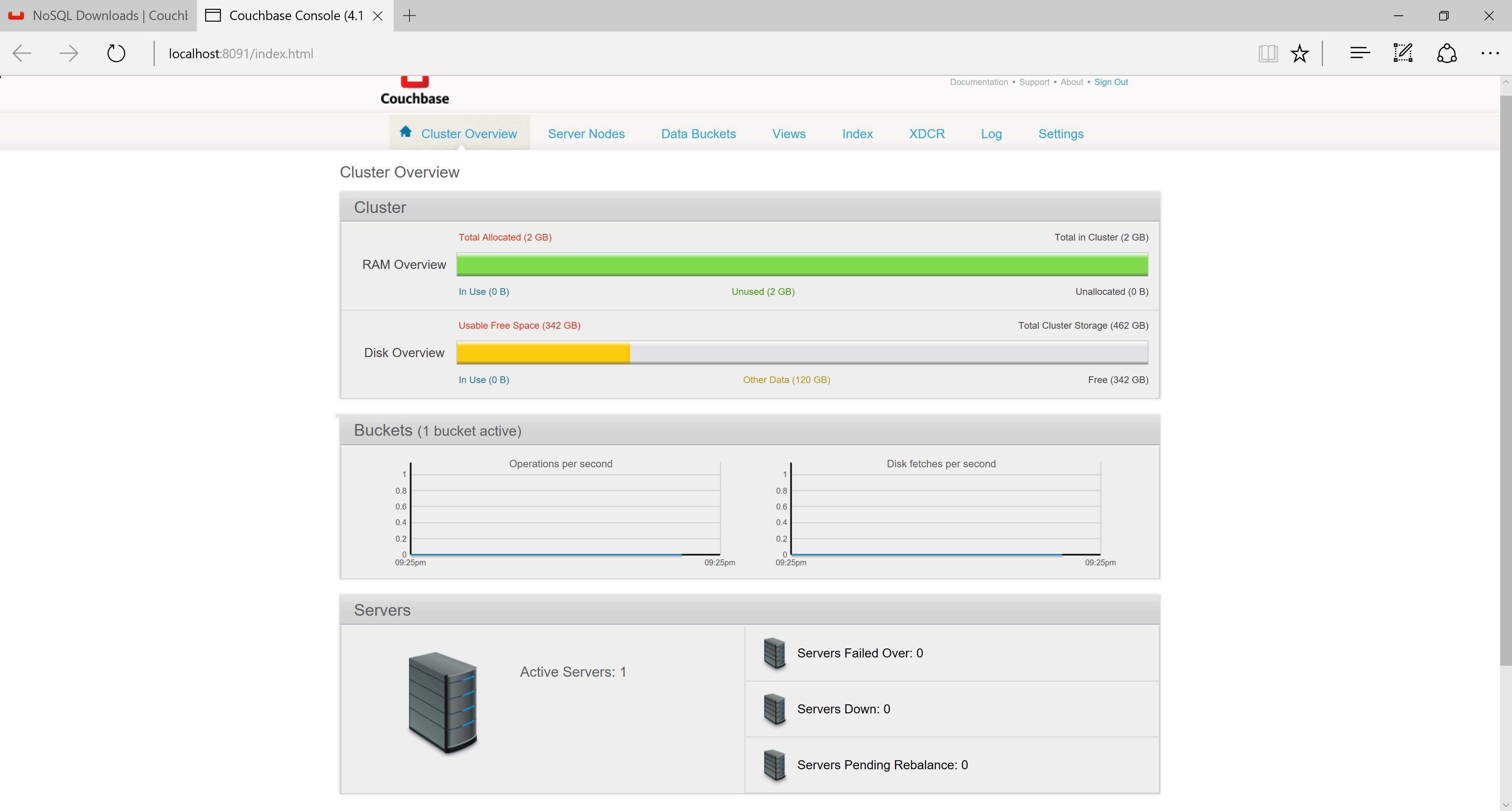
Change the bucket type to Memcached as shown below





Choose an administrator password in the following screen.



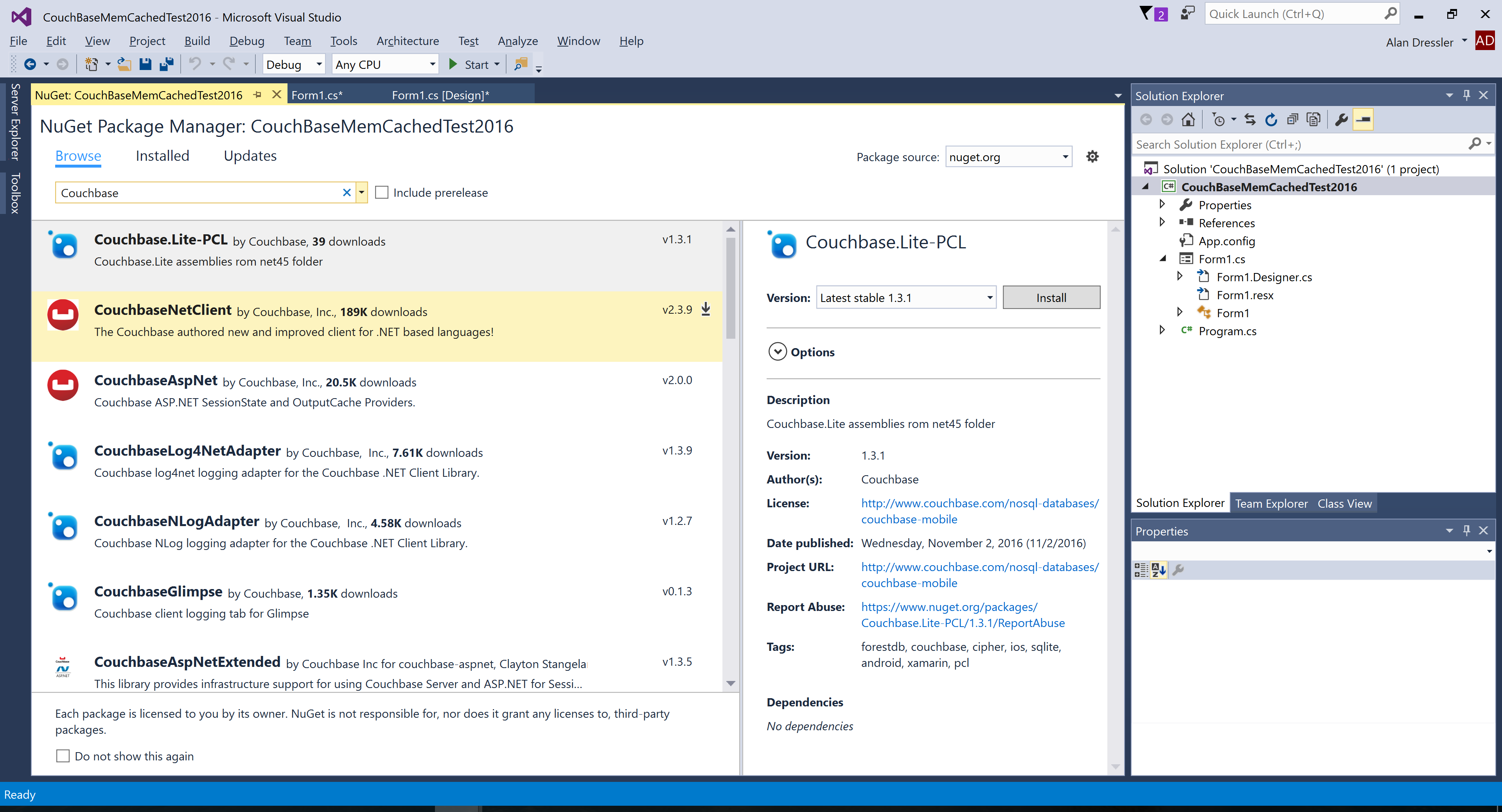


Now the Couchbase installation is complete, so we can test it.

**Creating a Test Application for Couchbase MemCached**:

Create a Windows project in Visual Studio. Name the project CouchBaseMemCachedTest2016.

From the Project menu, choose manage NuGet packages, then search on Couchbase as shown below. Select the CouchBaseNetClient to install.



Add a class called Student to the project with the following code:

**class Student**

{

public string FirstName { get; set; }

public string LastName { get; set; }

public int Id { get; set; }

}

Put two buttons on the form as shown and write the following code for the button handlers.

using Couchbase;

using Couchbase.Configuration.Client;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CouchBaseMemCachedTest2016

{

public partial class Form1 : Form

{

//private static readonly Cluster cluster = new Cluster();

static ClientConfiguration config = null;

private readonly Cluster cluster = null;

public Form1()

{

InitializeComponent();

config = new ClientConfiguration

{

Servers = new List<Uri>

{

new Uri("http://localhost:8091/pools"),

//new Uri("http://192.168.56.102:8091/pools"),

//new Uri("http://192.168.56.103:8091/pools"),

//new Uri("http://192.168.56.104:8091/pools"),

}

};

cluster = new Cluster(config);

}

private void btnMemCachedTest\_Click(object sender, EventArgs e)

{

// http://localhost:8091/index.html - couchbase admin url

try

{

//using (var bucket = cluster.OpenBucket())

{

using (var bucket = cluster.OpenBucket())

{

{

var doc = new Document<dynamic>

{

Id = "1234",

Content = new

{

fname = "Bill",

lname = "Baker"

}

};

var upsert = bucket.Upsert(doc);

if (upsert.Success)

{

var get = bucket.GetDocument<dynamic>(doc.Id);

doc = get.Document;

var msg = string.Format("{0} {1} {2}", doc.Id, doc.Content.fname, doc.Content.lname);

MessageBox.Show(msg);

}

}

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btnObjectStore\_Click(object sender, EventArgs e)

{

try

{

using (var cluster = new Cluster(config))

{

using (var bucket = cluster.OpenBucket())

{

Student s1 = new Student { FirstName = "Sally", LastName = "Simpson", Id = 1235 };

var upsert = bucket.Upsert<Student>("STData", s1);

if (upsert.Success)

{

var stData = bucket.GetDocument<Student>("STData");

var st = stData.Document.Content;

var msg = string.Format("{0} {1} {2}", st.Id, st.FirstName, st.LastName);

MessageBox.Show(msg);

}

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

cluster.Dispose();

}

}

}

Note that the line

**private static readonly Cluster cluster = new Cluster();**

creates a caching cluster on the localhost.

Build and test the storage and retrieval from Memcached.