

Corso di Sistemi Distribuiti

Corso di Laurea Magistrale in Ingegneria Informatica A.A. 2014/2015
DIMES - Università degli Studi della Calabria



DEVELOPMENT SERVICES ON WINDOWS AZURE

Microsoft Azure emulator

- The Azure SDK includes the **Microsoft Azure compute emulator**.
- The compute emulator is a local emulator of Azure, so that you can build and test your application before you deploy it.
- To use the compute emulator, you must download and install the **Azure SDK**.
 - Download it at <https://www.microsoft.com/windowsazure/sdk/>
- Microsoft .NET Framework 3.5 SP1 or .NET Framework 4.
- Microsoft SQL Server™ Express 2005 or SQL Server 2008 R2 Express for running applications that use the storage emulator.
- If you are intending to use a web role in your application, you must enable Internet Information Services (IIS) 7.0 with ASP.NET.

Useful link:

<https://msdn.microsoft.com/en-us/library/azure/ff687127.aspx>

<https://msdn.microsoft.com/en-us/library/azure/hh403990.aspx>

Step by step

- Install Visual Studio 2013 or 2015
- Install Azure SDK
- Launch Azure and install additional packages (if required).
- Create an Azure cloud service
- Build, run and debug your application locally.
- Publish your cloud code to Azure (if required).

Azure SDK installation

The screenshot shows a web browser window with the address bar displaying `azure.microsoft.com/it-it/downloads/`. The page header includes the Microsoft Azure logo, navigation links (Funzionalità, Prezzi, Documentazione, Download, Marketplace, Blog, Community, Supporto), and a search bar. A banner for 'Download' states 'Scarica gli SDK e gli strumenti da riga di comando necessari'. The main section is titled 'SDK' and includes the instruction 'Scarica e installa SDK per linguaggi specifici e strumenti per la tua piattaforma preferita.' Below this, there are four columns for different languages: .NET, Java, Node.js, and PHP. Each column lists links for installation on various operating systems (Windows, Mac, Linux) and documentation. The bottom of the browser window shows the Windows taskbar with the Start button, several application icons, and the system tray displaying the date and time as 18:26 on 28/04/2015.

Download di SDK e strum x

azure.microsoft.com/it-it/downloads/

VENDITE 800-788-741 | ACCOUNT PERSONALE | PORTALE | Ricerca

Microsoft Azure

Funzionalità | Prezzi | Documentazione | Download | Marketplace | Blog | Community | Supporto

VERSIONE DI VALUTAZIONE GRATUITA >

Download

Scarica gli SDK e gli strumenti da riga di comando necessari

SDK

Scarica e installa SDK per linguaggi specifici e strumenti per la tua piattaforma preferita.

.NET	Java	Node.js	PHP
Visual Studio 2015 - Installazione	Windows - Installazione	Windows - Installazione	Windows - Installazione
VS 2013 - Installazione	Mac - Installazione	Mac - Installazione	Mac - Installazione
VS 2012 - Installazione	Linux - Installazione	Linux - Installazione	Linux - Installazione
Librerie client	Documentazione	Documentazione	Documentazione
Documentazione			
Versioni precedenti			

go.microsoft.com/fwlink/p/?linkid=323510&clcid=0x410

uby

Dispositivi mobili

Contenuti multimediali

Mostra tutti i download...

WVDOVs2013Azure....exe

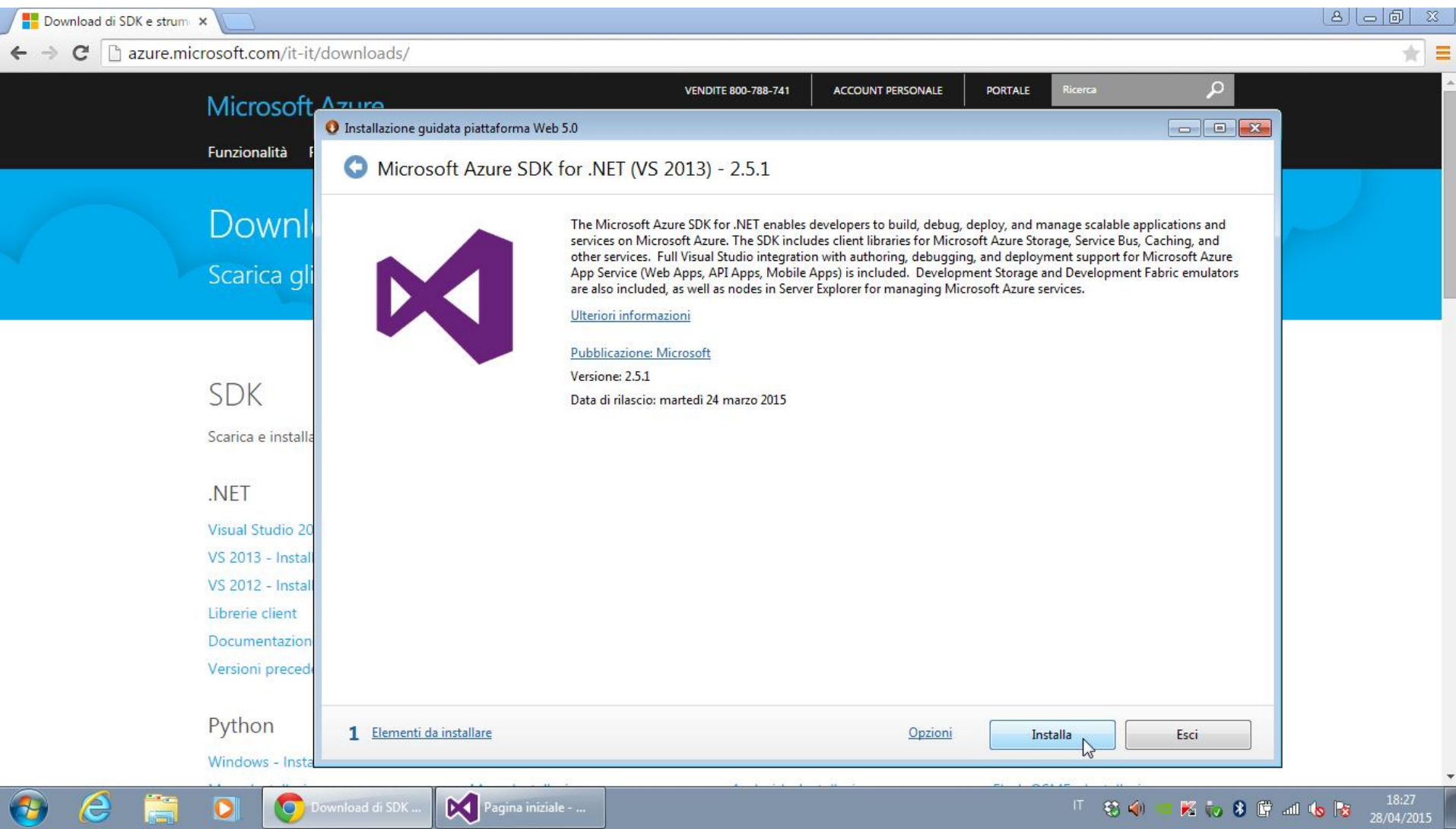
Download di SDK ...

Pagina iniziale - ...

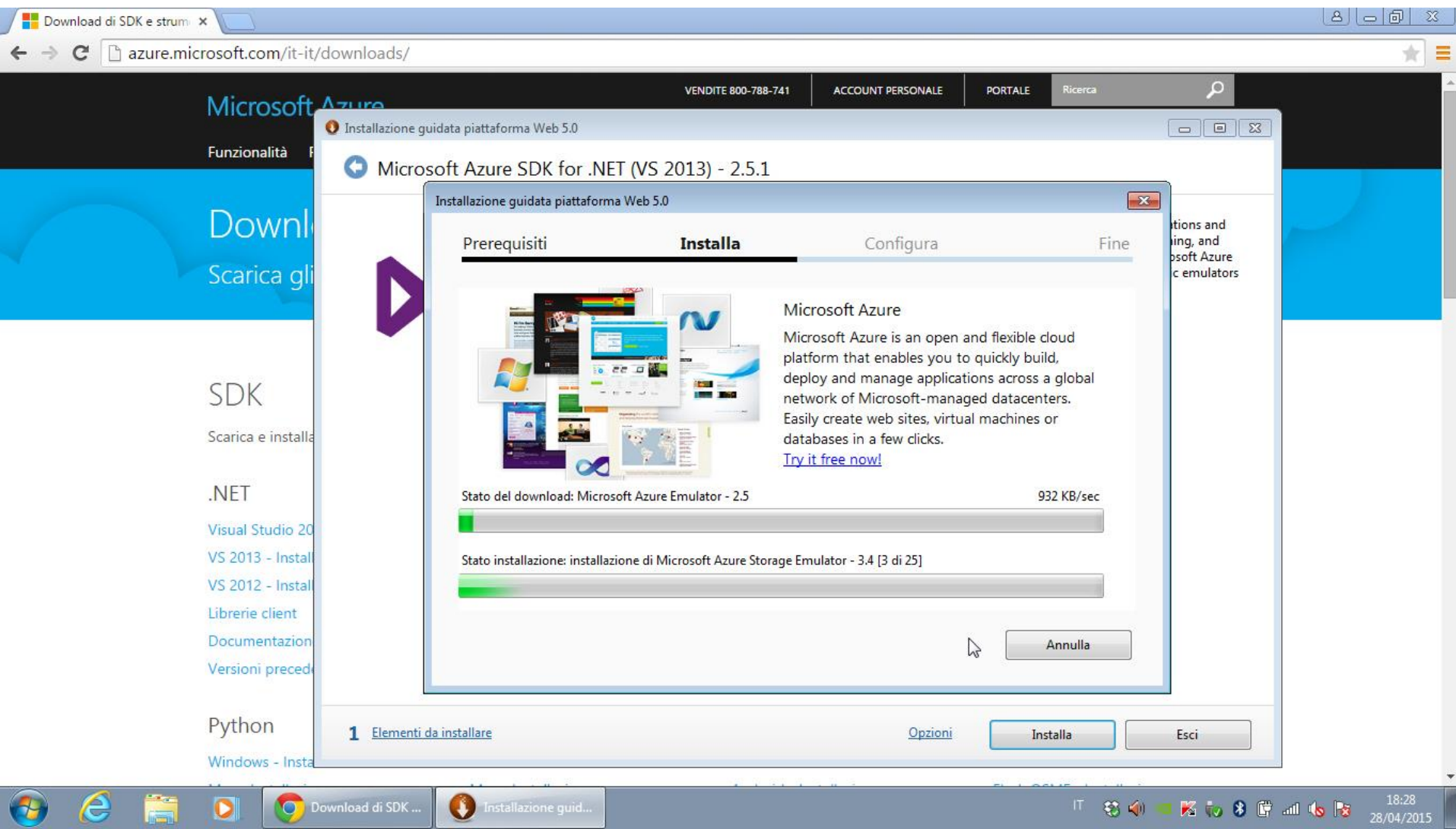
IT

18:26
28/04/2015

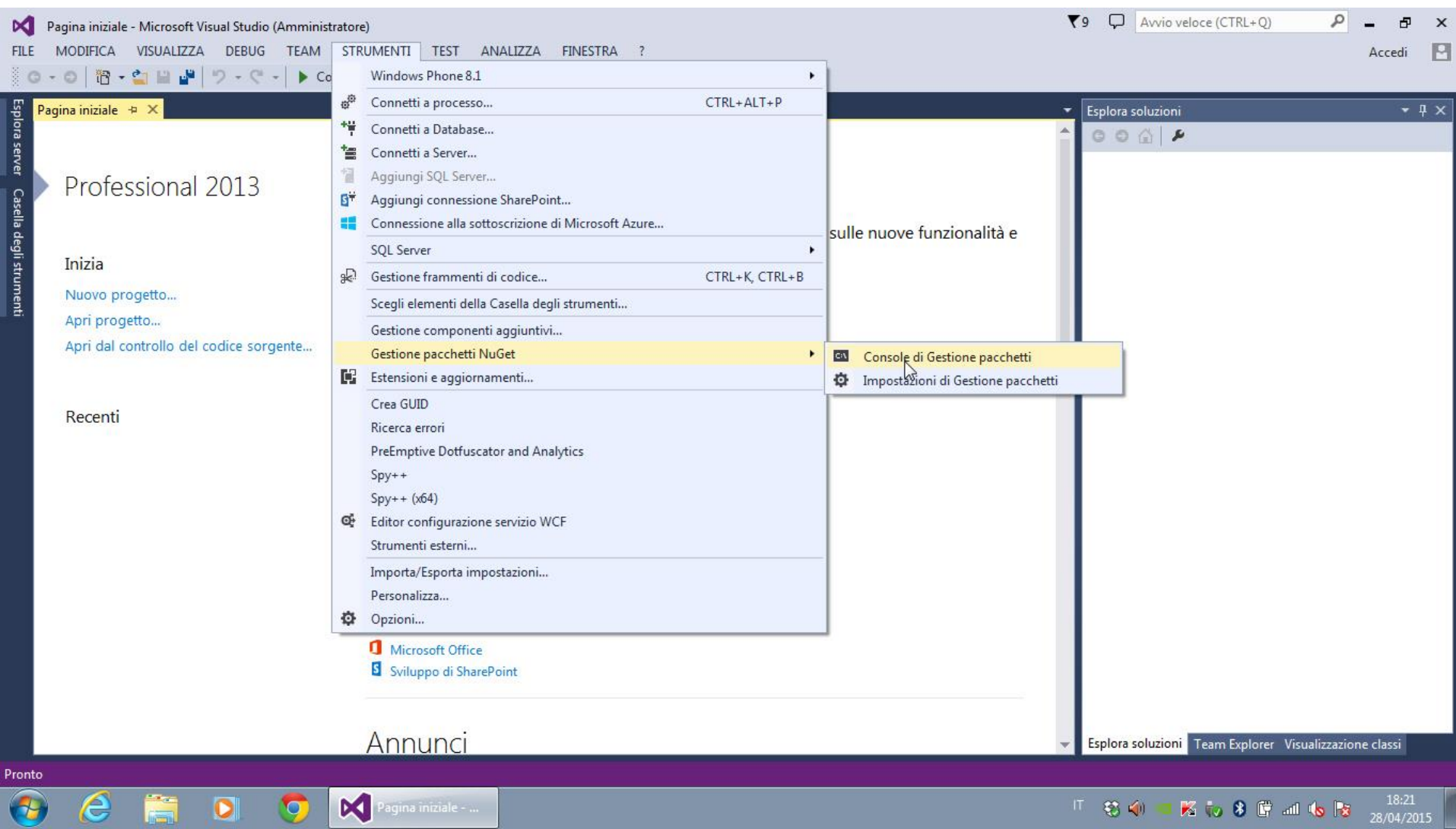
Azure SDK installation



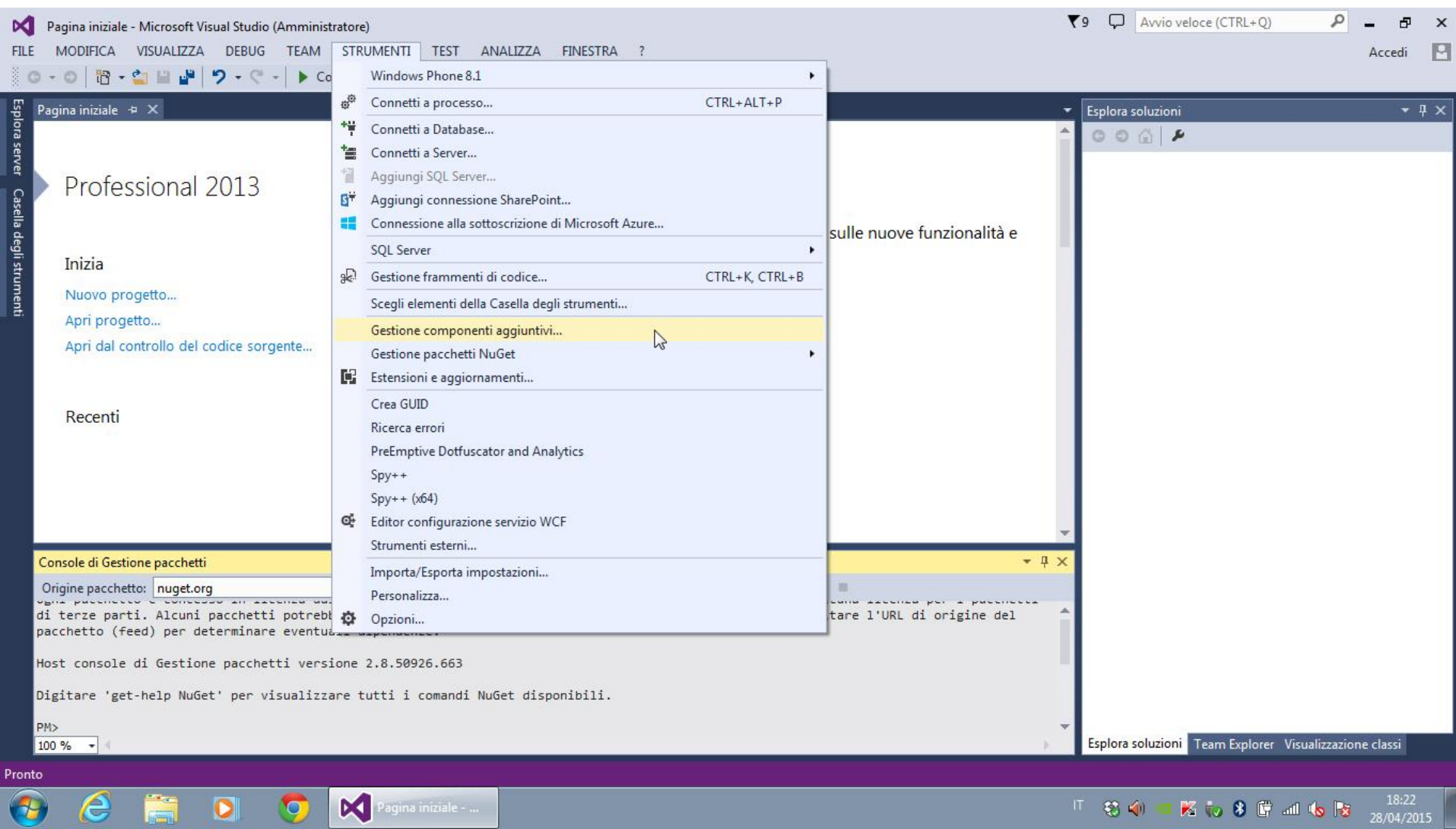
Azure SDK installation



Visual Studio additional components



Visual Studio additional components



Summary

- Hello World example
- Working with Queues (RdStorage)
- Working with Queues (RdStorage-MultipleMessages)

All the code has been tested by exploiting:

- MS Visual Studio 2013
- Windows Azure Compute Emulator – Version 3.4.0.0

Summary

- Hello World example
- Working with Queues (RdStorage)
- Working with Queues (RdStorage-MultipleMessages)

Create a simple Web 'HelloWorld' application

- Step 1: from Visual Studio create a new project Azure Cloud Service
- Step 2: add a Web Role ASP.NET and a Worker Role
- Step 3: run the application

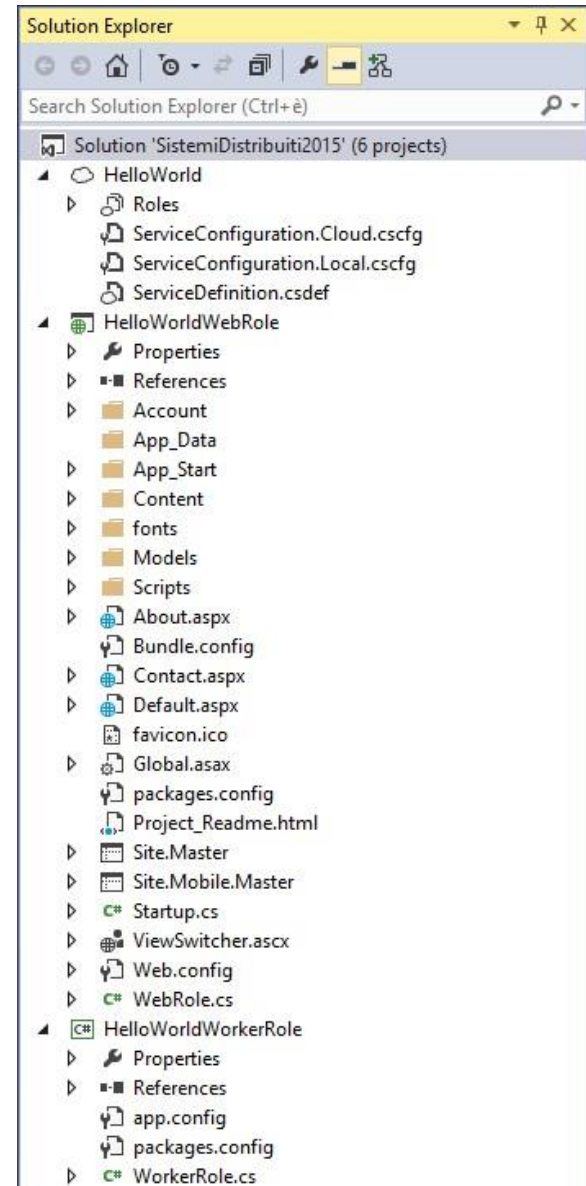
The solution created has 3 projects related to the Hello World example:

- **HelloWorld:** it includes the service definition file, ServiceDefinition.csdef, which contains metadata (i.e., which roles are used, their trust level, etc.).
- **HelloWorld_WebRole:** is a standard ASP.NET Web Application project template, with additional class to manage the initialization, starting, and stopping.
- **HelloWorld_WorkerRole:** it contains classes for the worker logic.

ASP.NET vs ASP.NET MVC

based on ASP.NET (Active Server Page), ASP.NET MVC allows software developers to build a web application as a composition of three roles: Model, View and Controller.

- A model represents the state of a particular aspect of the application.
- A controller handles interactions and updates the model to reflect a change in state of the application, and then passes information to the view.
- A view accepts necessary information from the controller and renders a user interface to display that information.



- Demo in classroom
- Question: any interaction with the worker ?

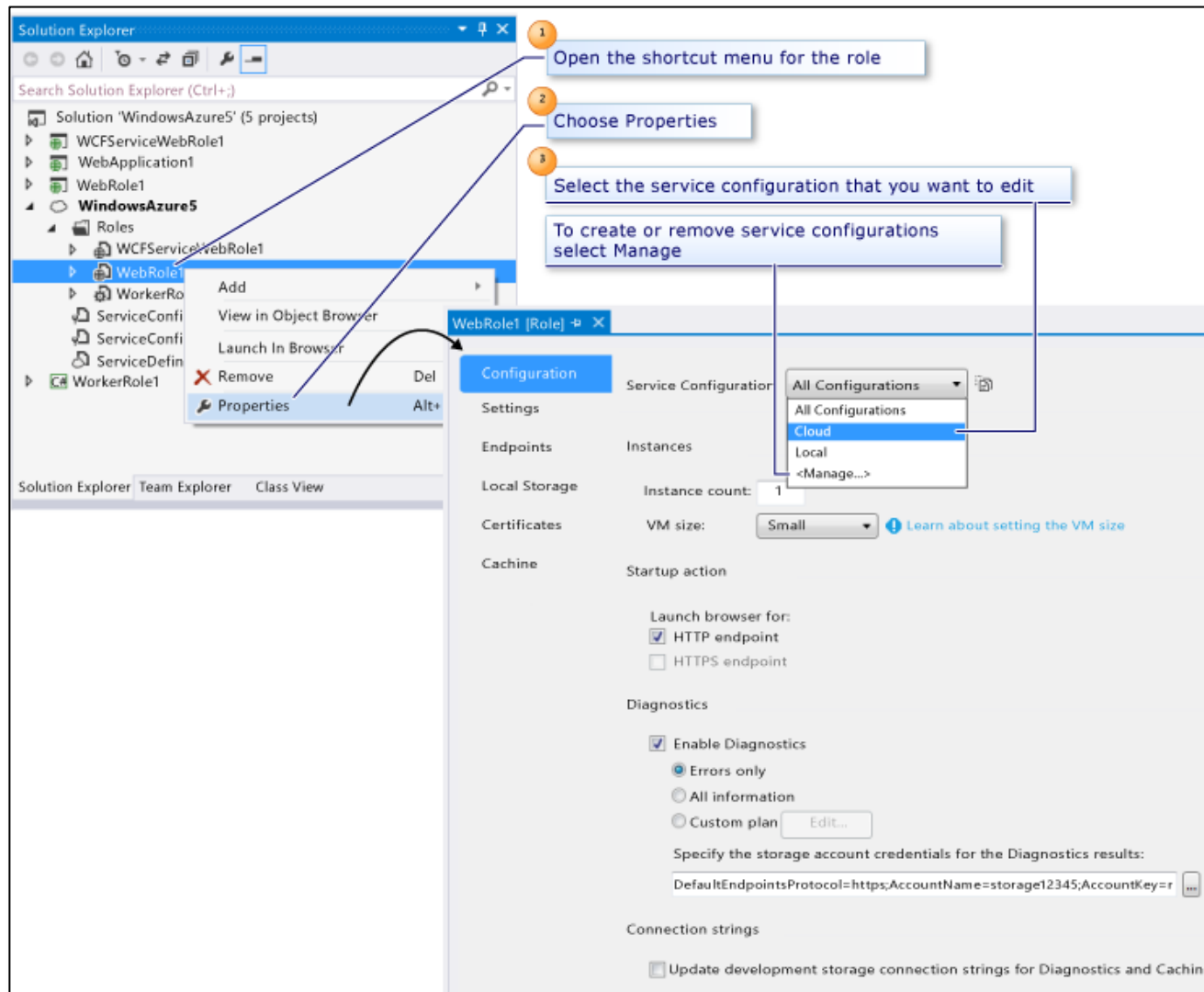
Summary

- Hello World example
- **Working with Queues (RdStorage)**
- Working with Queues (RdStorage-MultipleMessages)

Create a simple Web application to send messages to a Windows Azure queue

- A Worker role in the solution retrieves the messages and writes them to the compute emulator log.
- Queue service is a way to send messages between front-end roles and worker roles.
 - A queue can contain an unlimited number of messages, each of which can be up to 64 KB in size
 - Messages are pushed to the end of the queue and popped from the front of the queue.

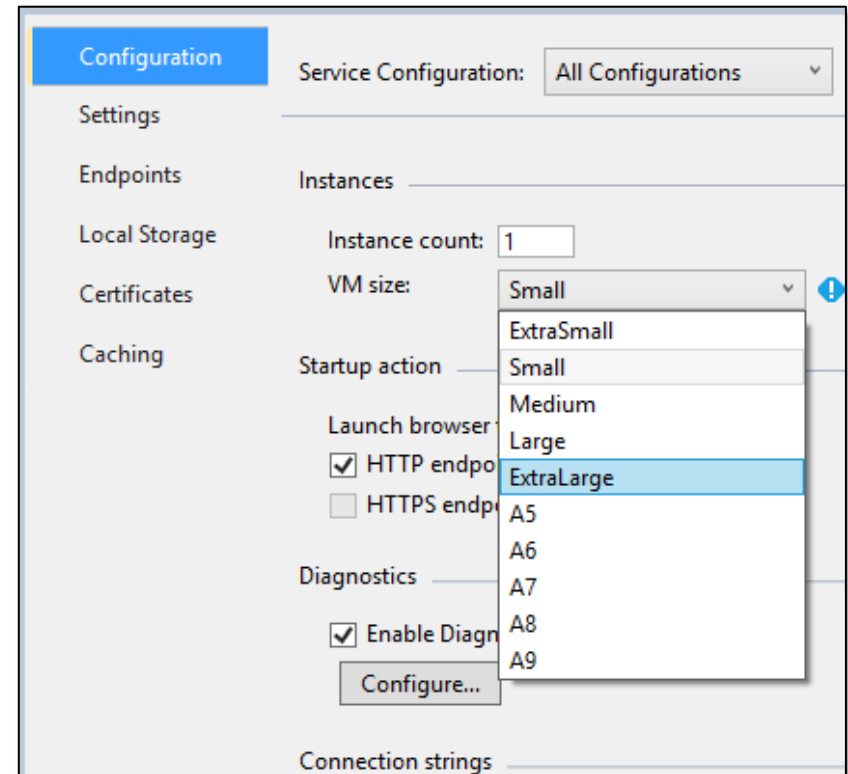
How to configure Roles for an Azure Cloud Service



More info at <https://msdn.microsoft.com/en-us/library/azure/hh369931.aspx>

How to change the number of instances for a role

1. Choose the Configuration tab.
2. In the Service Configuration list, choose the service configuration that you want to update.
 - ✓ **Note:** you can set the instance count for a specific service configuration or for all service configurations.
3. In the Instance count text box, type the number of instances that you want to start for this role.
 - ✓ **Note:** each instance is run on a separate virtual machine when you publish your cloud service to Azure.
4. Choose the Save icon in the toolbar to save these changes to the service configuration file.



More info at <https://msdn.microsoft.com/en-us/library/azure/hh369931.aspx>

How to manage connection strings for storage accounts

- You can add, remove or modify connection strings for your service configurations.
 - For example, a local connection string for a local service configuration has a value of *UseDevelopmentStorage=true*.
 - You might also want to configure a cloud service configuration that uses a storage account in Azure.
1. Choose the Settings tab.
 2. In the Service Configuration list, choose the service configuration that you want to update.
 - ✓ **Note:** you can update connection strings for a specific service configuration, but if you need to add or delete a connection string you must select All Configurations.
 3. To add a connection string, choose the **Add Setting** button -> A new entry is added to the list.
 4. In the field **Name** text box, type the name for the connection string (e.g. *DataConnectionString*).
 5. In the **Type** drop-down list, choose **Connection String**.
 6. To change the value for the connection string, choose the ellipsis button. -> The Create Storage Connection String dialog box is displayed.
 - ✓ To use the **local storage** account emulator, choose **Microsoft Azure storage emulator**.
 - ✓ To use a storage account in Azure, choose the **Your subscription** radio button, and select the desired storage account.
 7. Choose the **Save** icon.

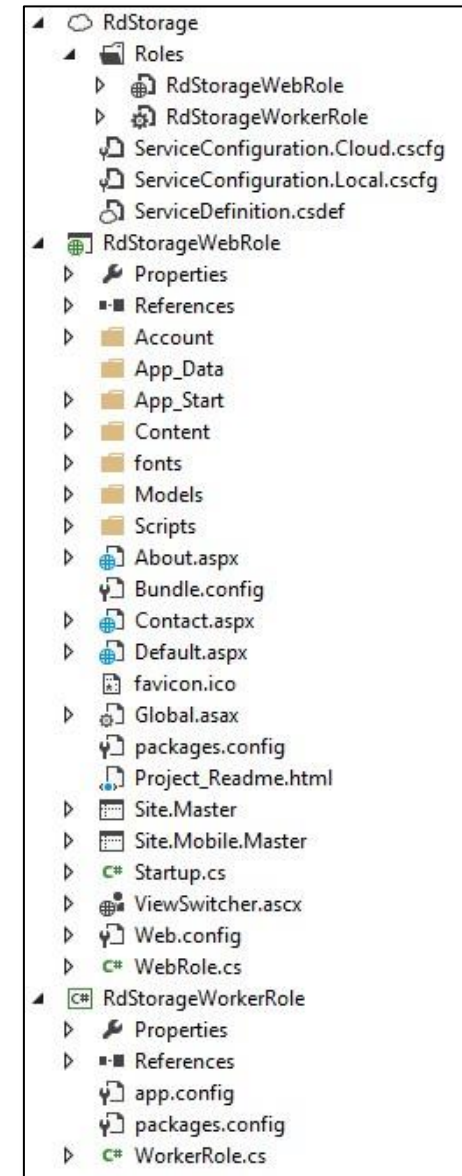
More info at <https://msdn.microsoft.com/en-us/library/azure/hh369931.aspx>

Task 1: Creating the initial solution

- Create a new Windows Azure Project (**RdStorage**)
- Create a new ASP.NET Web Role (**RdStorage_WebRole**)
- Create a new Worker Role (**RdStorage_WorkerRole**)
- Create a Storage Connection String
 - Since the Web and Worker Roles use Windows Azure storage services, the user has to specify the storage account settings for each role
 - A “DataConnectionString” is specified for each role in the project, each one using the Storage Local Emulator

The solution has 3 projects related to RdStorage:

- **RdStorage:** it includes the service definition file, **ServiceDefinition.csdef**, which contains metadata (i.e., which roles are used, their trust level, etc.).
- **RdStorage_WebRole:** is a standard ASP.NET Web Application project template, with additional class to manage the initialization, starting, and stopping.
- **RdStorage_WorkerRole:** it contains classes for the worker



Task 2: Sending messages to the Queue

- The **RdStorage_WebRole** web application is developed, to send messages to the queue.
- Create the ASP Web Page of the Web Role (**Default.aspx** file of the **RdStorage_WebRole**)

```
...  
<asp:TextBox ID="txtMessage" runat="server"></asp:TextBox>  
<asp:Button ID="btnSend" runat="server" Text="Send message"  
OnClick="btnSend_Click"/>  
...
```

- Implement the **btnSend_Click** event (**Default.aspx.cs** file of the **RdStorage_WebRole**)

```
...  
protected void btnSend_Click(object sender, EventArgs e)  
{  
    // initialize the account information  
    ...  
    // retrieve a reference to the messages queue  
    ...  
    // Create a message and add it to the queue.  
    CloudQueueMessage message = new CloudQueueMessage(txtMessage.Text);  
    queue.AddMessage(message);  
    //Clear text box value  
    txtMessage.Text = string.Empty;  
} //btnSend_Click  
...
```

For a better solution, these parts can be moved to another method of **Default.aspx.cs** (see next slide for the full code)

Full code 1/2 – Web Role

```
using Microsoft.WindowsAzure;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Queue;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace RdStorageWebRole
{
    public partial class _Default : Page
    {
        CloudQueue queue = null;

        protected void Page_Load(object sender, EventArgs e)
        {
            // initialize the account information
            CloudStorageAccount storageAccount =
            CloudStorageAccount.Parse(CloudConfigurationManager.GetSetting("DataConnectionString"));
            // Create the queue client
            CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient();
            // Retrieve a reference to a queue
            queue = queueClient.GetQueueReference("messagequeue");
            // Create the queue if it doesn't already exist
            queue.CreateIfNotExists();
        }
    }
    ...
}
```

Full code 2/2 – Web Role

```
...
protected void btnSend_Click(object sender, EventArgs e)
{
    // Create a message and add it to the queue.
    CloudQueueMessage message = new CloudQueueMessage(txtMessage.Text);
    queue.AddMessage(message);

    //Clear text box value
    txtMessage.Text = string.Empty;
} //btnSend_Click
```


The Web interface

[Sistemi distribuiti](#) [Home](#)

Corso di sistemi distribuiti

Development Services on Windows Azure

RdStorage example

- Create a simple Web application to send messages to a Windows Azure queue
- A Worker role in the solution retrieves the messages and writes them to the compute emulator log

Insert here your message:

© 2015 - My ASP.NET Application

Task 3: Retrieving messages from the Queue (who? ... the worker!!!)

- The **RdStorage_WorkerRole** is implemented, to retrieve messages from the queue and show them in the compute emulator log.
- Implement the logic of the Worker Role (**WorkerRole.cs** file of the **RdStorage_WorkerRole** project).
 1. Initialize some account information (in **onStart** or **Run** method of the **WorkerRole.cs** file).
 2. Create and retrieve a reference to the queue (in **onStart** or **Run** method of the **WorkerRole.cs** file).
 3. Retrieve messages from the queue and write them to the compute emulator log (**Run** method of the **WorkerRole.cs** file).

Full code 1/2 – WorkerRole.cs

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Net;
using System.Threading;
using System.Threading.Tasks;
using Microsoft.WindowsAzure;
using Microsoft.WindowsAzure.Diagnostics;
using Microsoft.WindowsAzure.ServiceRuntime;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Queue;

namespace RdStorageWorkerRole
{
    public class WorkerRole : RoleEntryPoint
    {
        public override bool OnStart()
        {
            // Set the maximum number of concurrent connections
            ServicePointManager.DefaultConnectionLimit = 12;
            bool result = base.OnStart();
            Trace.TraceInformation("RdStorageWorkerRole has been started");
            return result;
        }
    }
}
```

Full code 2/2 – WorkerRole.cs

...

```
public override void Run()
{
    Trace.TraceInformation("RdStorageWorkerRole is running");
    // initialize the account information
    CloudStorageAccount storageAccount =
    CloudStorageAccount.Parse(CloudConfigurationManager.GetSetting("DataConnectionString"));
    // Create the queue client
    CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient();
    // Retrieve a reference to a queue
    CloudQueue queue = queueClient.GetQueueReference("messagequeue");
    // retrieve messages and write them to the compute emulator log
    while (true)
    {
        Thread.Sleep(2000);
        if (queue.Exists())
        {
            var msg = queue.GetMessage();
            if (msg != null)
            {
                Trace.TraceInformation(string.Format("Message '{0}' processed.",
msg.AsString));
                queue.DeleteMessage(msg);
            }
        }
    } //while
} //Run end

} //class
} //namespace
```

The worker process will try to get a message from the queue every 2 seconds using the **GetMessage** method. If there are messages in the queue, it will show them in the Compute Emulator log.

Demo

Task 4: Execution of the application and watch by the Emulator UI

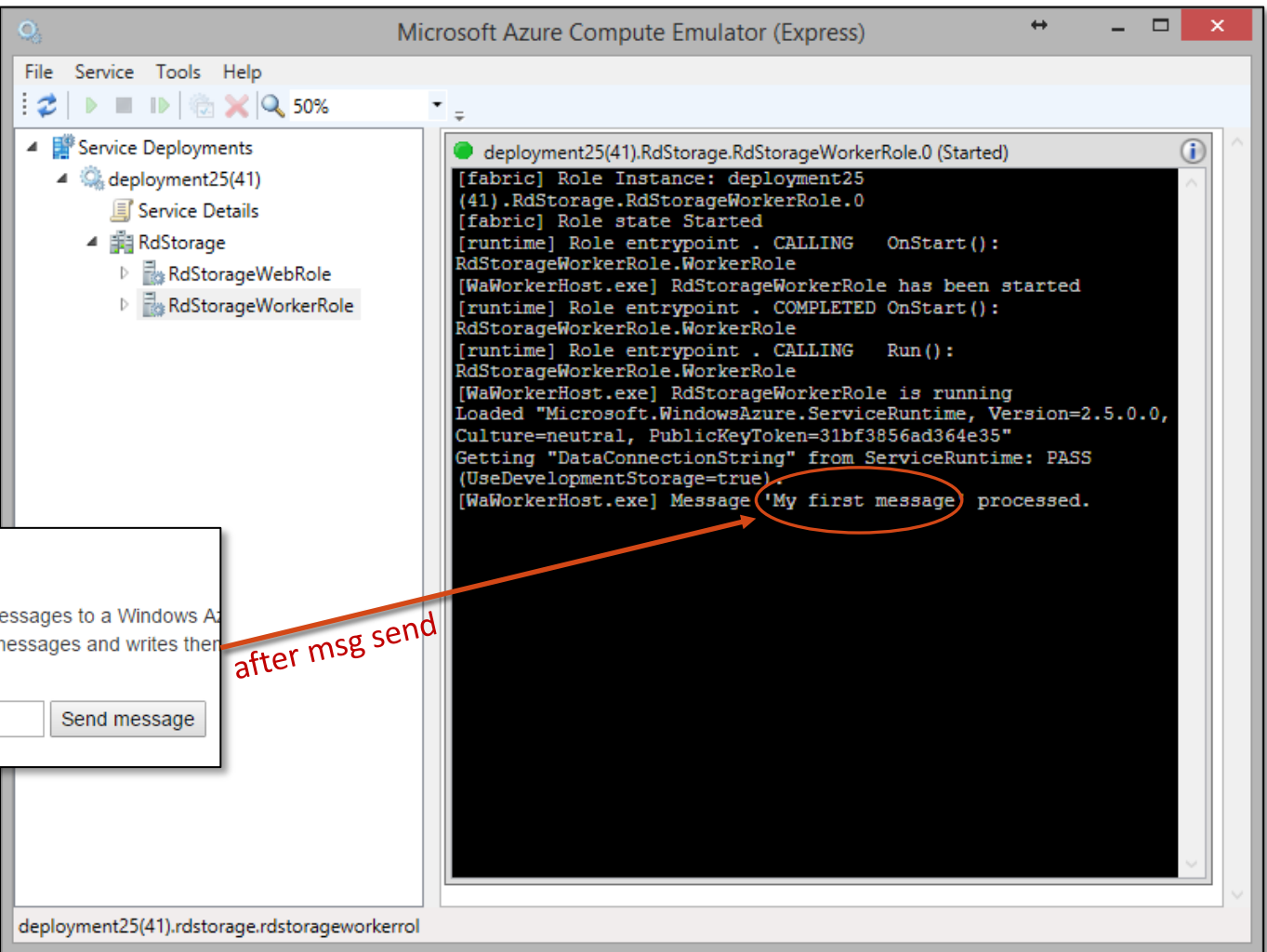
Browser

RdStorage example

- Create a simple Web application to send messages to a Windows Azure Service Bus
- A Worker role in the solution retrieves the messages and writes them to the console

Insert here your message:

after msg send



Microsoft Azure Compute Emulator (Express)

File Service Tools Help

50%

Service Deployments

- deployment25(41)
 - Service Details
 - RdStorage
 - RdStorageWebRole
 - RdStorageWorkerRole

deployment25(41).RdStorage.RdStorageWorkerRole.0 (Started)

```
[fabric] Role Instance: deployment25(41).RdStorage.RdStorageWorkerRole.0
[fabric] Role state Started
[runtime] Role entrypoint . CALLING OnStart():
RdStorageWorkerRole.WorkerRole
[WaWorkerHost.exe] RdStorageWorkerRole has been started
[runtime] Role entrypoint . COMPLETED OnStart():
RdStorageWorkerRole.WorkerRole
[runtime] Role entrypoint . CALLING Run():
RdStorageWorkerRole.WorkerRole
[WaWorkerHost.exe] RdStorageWorkerRole is running
Loaded "Microsoft.WindowsAzure.ServiceRuntime, Version=2.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
Getting "DataConnectionString" from ServiceRuntime: PASS
(UseDevelopmentStorage=true).
[WaWorkerHost.exe] Message 'My first message' processed.
```

deployment25(41).rdstorage.rdstorageworkerrol

Trace Information

- You can fix the Logging Level (**TraceInformation (...)** write at Information level)

- Example:

```
Trace.TraceInformation(string.Format(" Message '{0}' '{1}' processed.", msg.AsString, msg.AsString));
```

More info at <https://msdn.microsoft.com/en-us/library/system.diagnostics.trace.traceinformation.aspx>

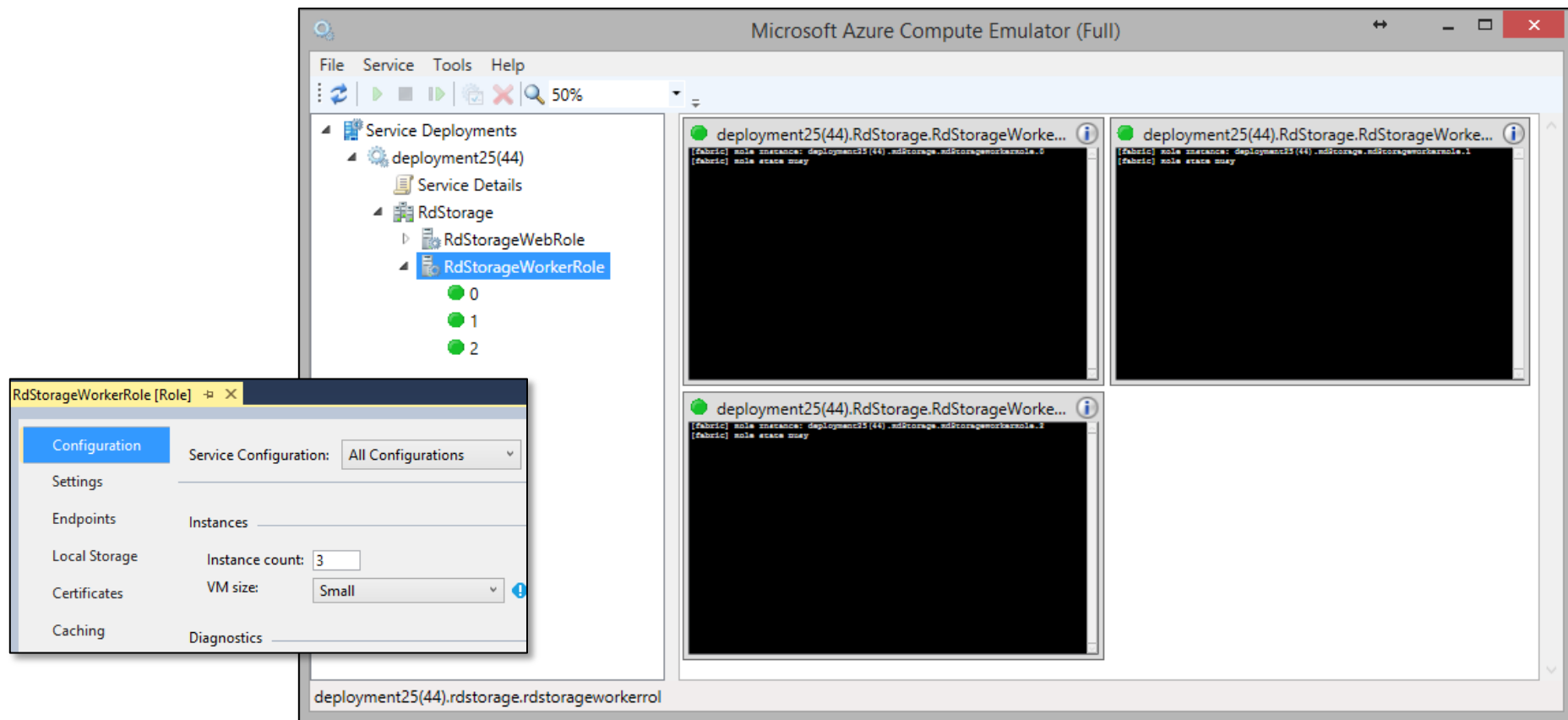
The screenshot shows the MSDN page for the **Trace.TraceInformation Method**. The page is part of the Microsoft Developer Network, with navigation links for Technologies, Downloads, Programs, Community, Documentation, and Samples. The left sidebar lists the navigation path: MSDN Library > .NET Development > .NET Framework 4.5 and 4.6 Preview > .NET Framework Class Library > System.Diagnostics Namespaces > System.Diagnostics > Trace Class > Trace Methods > **TraceInformation Method**. The main content area shows the method signature for **TraceInformation Method** in the .NET Framework 4.5. It describes the method as writing an informational message to the trace listeners in the **Listeners** collection. A note states that the member is overloaded and provides a link to the overload list. The **Overload List** contains two entries:

	Name	Description
	<code>TraceInformation(String)</code>	Writes an informational message to the trace listeners in the Listeners collection using the specified message.
	<code>TraceInformation(String, Object[])</code>	Writes an informational message to the trace listeners in the Listeners collection using the specified array of objects and formatting information.

Below the table, there is a **See Also** section with links to **Reference**, **Trace Class**, and **System.Diagnostics Namespace**.

Demo

- RdStorage example works with 1 worker.
- You can change the number of Worker instances by «RdStorage->Roles»
- More info at <https://msdn.microsoft.com/en-us/library/azure/dn339018.aspx>



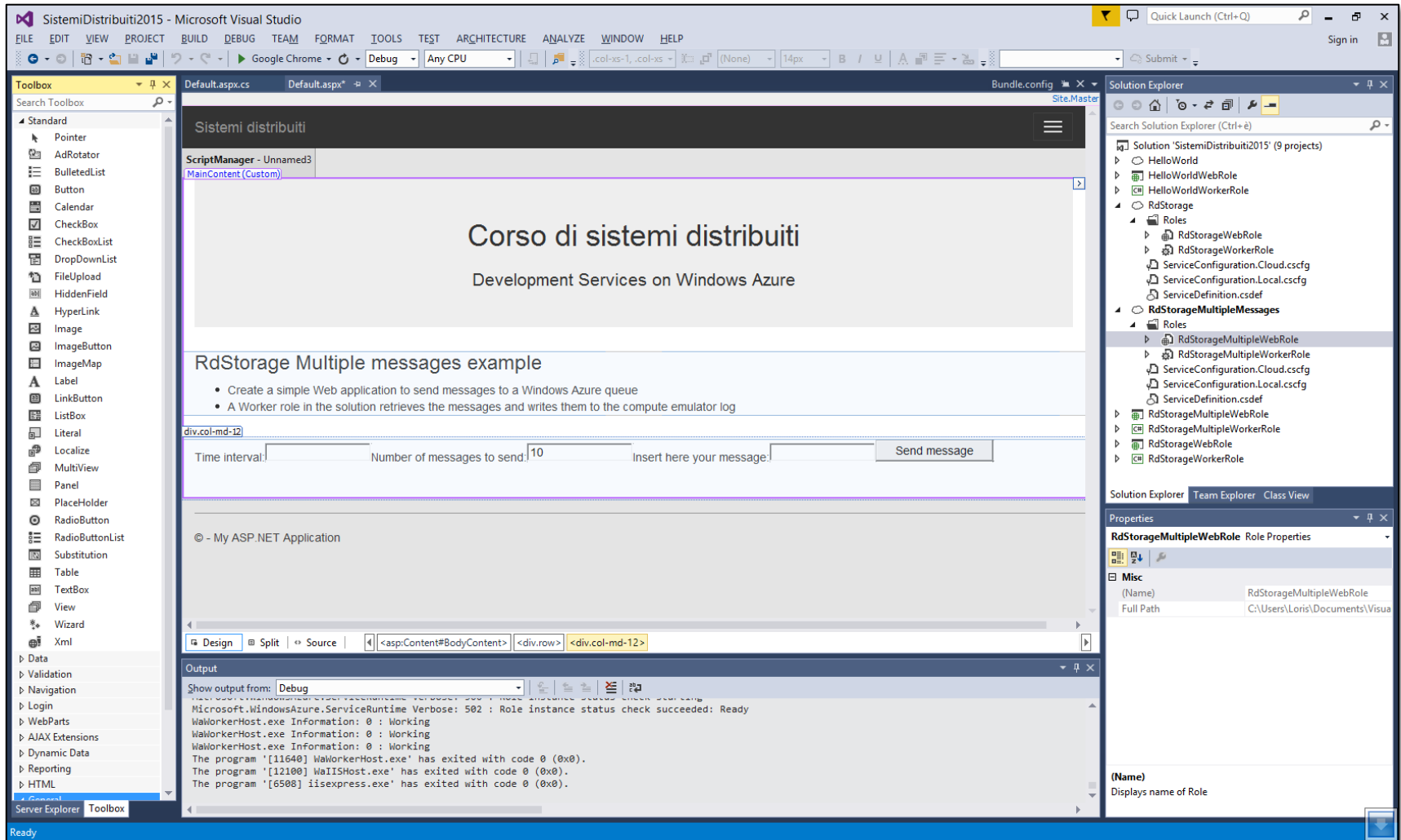
Summary

- Hello World example
- Working with Queues (RdStorage)
- Working with Queues (RdStorage-MultipleMessages)

- Let us modify the previous example, by automatically sending more messages in sequence.
- The user can fix:
 - String msg to be sent to the worker(s)
 - Period of repetition
 - Number N of messages to be sent (msg<i>, for i=1...N)

RdStorage-MultipleMessages

The ASP Web Page can be designed by graphical tools



Modified the Default.aspx file of the Web Role

```
<div class="col-md-12">
    <asp:Label ID="Label2" runat="server" Text="Time interval:"></asp:Label>
    <asp:TextBox ID="txtPeriod" runat="server"></asp:TextBox>
    <asp:Label ID="Label3" runat="server" Text="Number of messages to send:"></asp:Label>
    <asp:TextBox ID="numMessages" runat="server" Text="10"></asp:TextBox>
    <asp:Label ID="Label1" runat="server" Text="Insert here your message:"></asp:Label>
    <asp:TextBox ID="txtMessage" runat="server"></asp:TextBox>
    <asp:Button ID="btnSend" runat="server" Text="Send message" OnClick="btnSend_Click" />
</div>
```

RdStorage-MultipleMessages

Modified the 'btnSend_Click' method in Default.aspx.cs of the Web Role

```
protected void btnSend_Click_Multiple(object sender, EventArgs e)
{
    // initialize the account information
    CloudStorageAccount storageAccount =
    CloudStorageAccount.Parse(CloudConfigurationManager.GetSetting("DataConnectionString"));
    // Create the queue client
    CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient();
    // Retrieve a reference to a queue
    CloudQueue queue = queueClient.GetQueueReference("messagequeue");
    // insert multiple messages into the queue
    int n, sleepTime;
    string txtMsg = txtMessage.Text;
    string txtP = txtPeriod.Text;
    int.TryParse(txtP, out sleepTime);
    string numMsg = numMessages.Text;
    int.TryParse(numMsg, out n);
    for (int i = 0; i < n; i++)
    {
        Thread.Sleep(sleepTime * 1000);
        string currMsg = txtMsg + i;
        var msg = new CloudQueueMessage(currMsg);
        queue.AddMessage(msg);
    }
    txtMessage.Text = string.Empty;
} //btnSend_Click
```

← append an 'id' to the string automatically built

- Demo in classroom

Documentation Azure

<http://azure.microsoft.com/en-us/documentation/>

The screenshot shows the Microsoft Azure Documentation Center homepage. The top navigation bar is dark blue with links for Features, Pricing, Documentation (highlighted), Downloads, Marketplace, Blog, Community, and Support. On the right, there are links for SALES 800-788-741, MY ACCOUNT, PORTAL, a search bar, and a FREE TRIAL button with a right arrow icon. The main content area has a blue background with the heading 'Documentation Center'. Below the heading, there are three bullet points: 'Build solutions with any language or development platform.', 'Target any browser, client or device.', and 'Run on Windows or Linux.' To the right of these points is a grid of eight dark blue buttons with white text and right arrows, representing different programming languages and platforms: .NET, Node.js, Java, iOS, Android, Windows, PHP, Python, Ruby, and Media. Below this grid, there is a section titled 'Documentation by service' with five light gray boxes. Each box contains a service category and a list of related services.

Microsoft Azure

SALES 800-788-741 | MY ACCOUNT | PORTAL | Search

Features Pricing **Documentation** Downloads Marketplace Blog Community Support

FREE TRIAL →

Documentation Center

Build solutions with any language or development platform.
Target any browser, client or device.
Run on Windows or Linux.

- .NET >
- Node.js >
- Java >
- iOS, Android, Windows >
- PHP >
- Python >
- Ruby >
- Media >

Documentation by service

Compute <ul style="list-style-type: none">Virtual MachinesCloud ServicesRemoteAppBatch	Web & Mobile <ul style="list-style-type: none">WebsitesMobile ServicesAPI ManagementPush Notifications	Data & Storage <ul style="list-style-type: none">SQL DatabaseDocumentDBCacheStorageStorSimpleAzure Search	Analytics <ul style="list-style-type: none">HDInsightMachine LearningStream AnalyticsData Factory	Networking <ul style="list-style-type: none">Virtual NetworkExpressRouteTraffic Manager
--	--	---	---	--

Documentation .NET

<http://azure.microsoft.com/en-us/develop/net/>

Microsoft Azure

SALES 800-788-741 | MY ACCOUNT | PORTAL | Search

Features Pricing Documentation Downloads Marketplace Blog Community Support

FREE TRIAL →

.NET Azure Documentation

- ✓ Use .NET and Visual Studio to deploy a new or existing application in seconds.
- ✓ Automate deployment from TFS or source code repositories like GitHub.
- ✓ Expand functionality with managed services like Storage and SQL Database.
- ✓ Learn how to run ASP.NET web apps and .NET programs in the cloud by using Azure Websites, WebJobs, Cloud Services, and VMs

Install the SDK for VS 2015 Preview »

Install the SDK for VS 2013 »

Install the SDK for VS 2012 »

Featured

Get started with Websites & ASP.NET

Start Tutorial >

Get started

- Get started with Azure Websites and ASP.NET
- Get started with the Azure SDK for .NET
- Introduction to Azure
- Azure architecture poster
- Websites, Cloud Services, or Virtual Machines?

What's new

- Azure Resource Manager tools preview
- Azure SDK for .NET 2.4
- Azure PowerShell release
- Event Hubs in public preview
- Azure SDK for .NET 2.5
- More »

API reference

- Blob Storage .NET | REST
- Table Storage .NET | REST
- Service Management .NET | REST
- More »

Virtual Machines | Websites | Cloud Services | Mobile

Jump to Select ...

Less More