

Je oplossing ook in de wolken blijven volgen

WINDOWS AZURE DIAGNOSTICS MAAKT HET MOGELIJK

Pieter de Bruin

Organisaties denken steeds meer na over de optie om applicaties in de cloud te maken en bestaande applicaties er heen te migreren. Microsoft maakt het met Windows Azure en haar tools heel eenvoudig voor .NET ontwikkelaars om die oplossingen te maken. Maar als je oplossing eenmaal in de wolken staat, wil je om verschillende redenen ook weten hoe het ermee gaat. We bekijken de mogelijkheden om diverse informatiebronnen toegankelijk te maken.

Zoals je weet is Windows Azure Microsoft's basis voor het maken van oplossingen in de wolken. De beschikbare diensten bieden in feite rekenkracht en opslagcapaciteit met een schaalbaar operating systeem, database, service bus en beveiligingsdienst, waarmee je ieder systeem kunt maken. Voor organisaties is de cloud relevant als ze bijvoorbeeld een beheerafdeling hebben die maanden nodig heeft om een server beschikbaar te stellen of als ze een website hebben met een sterk wisselende load. En natuurlijk zijn er nog veel meer redenen zoals kostenvoordelen, outsourcing, etcetera.

Een belangrijk punt bij de overweging of de cloud ook voor jouw organisatie en systemen geschikt is, is het ongrijpbare van de cloud. Op dit moment heb je een serverruimte of een datacenter waar een machine staat die je websites en webservices host en als je je oplossing in de cloud zet, verdwijnt dat tastbare; Je kunt er niet even langslopen om te controleren of hij aan staat, inloggen met remote desktop om te kijken hoe druk hij is en of er nog fouten zijn opgetreden. En Windows Azure oplossingen schalen automatisch, dus je weet eigenlijk niet eens hoeveel instanties van Windows Azure jouw website hosten. Het goede nieuws is dat je op een nieuwe manier wel aan deze informatie kunt komen. En die informatie is beschikbaar van bekende bronnen als eventlog, IIS en performance monitor, zodat je je ervaring met Windows en .NET kunt blijven gebruiken.

Overigens zijn er meer redenen waarom je wilt weten wat er met je applicatie gebeurt; denk maar eens aan het controleren van een deployment of het oplossen van een bug. Maar wellicht gaat het je om minder code-gerelateerde zaken als het meten van de beschikbaarheid van je website, het meten van het aantal requests naar een webservice dat je in rekening wilt brengen of om vast te leggen wie gegevens benaderd heeft.

We beginnen met te controleren of Windows Azure wel aan staat. In principe is Windows Azure altijd actief. Het kan echter gebeu-

ren dat er een update is of een andere reden waarom een onderdeel van Windows Azure niet beschikbaar is. Microsoft heeft hiervoor onderstaand servicedashboard, waarop je direct ziet hoe het met de diverse onderdelen per continent gaat.

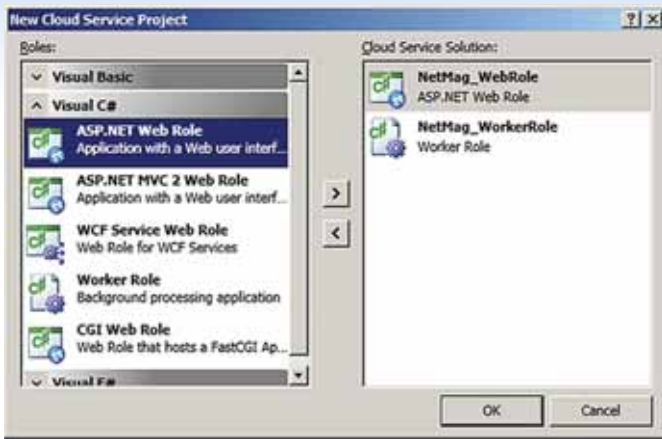
Current Status
Current status of the health of the Windows Azure platform is shown in the table below. If you wish to receive notifications for interruptions to any of the services, please subscribe to the respective RSS feeds. To view a detailed incident report for a service that is not running normally, mouse over the status icon or the status description for that service.

Status	Service [Sub-Region]	Description	RSS
●	AppFabric Access Control [South Central US]	Service is running normally.	●
●	AppFabric Service Bus [South Central US]	Service is running normally.	●
●	SQL Azure Database [North Europe]	Service is running normally.	●
●	SQL Azure Database [South Central US]	Service is running normally.	●
●	SQL Azure Database [Southeast Asia]	Service is running normally.	●
●	Windows Azure CDN [Worldwide]	Service is running normally.	●
●	Windows Azure Compute [North Central US]	Service is running normally.	●
●	Windows Azure Compute [North Europe]	Service is running normally.	●
●	Windows Azure Compute [South Central US]	Service is running normally.	●
●	Windows Azure Compute [Southeast Asia]	Service is running normally.	●
●	Windows Azure Storage [North Central US]	Service is running normally.	●
●	Windows Azure Storage [North Europe]	Service is running normally.	●
●	Windows Azure Storage [South Central US]	Service is running normally.	●
●	Windows Azure Storage [Southeast Asia]	Service is running normally.	●

FIGUUR 1: HET SERVICE MANAGEMENT DASHBOARD.

Het dashboard kan ook historische informatie per onderdeel tonen, zodat je achteraf kunt zien of er een probleem was. Het Windows Azure servicedashboard benader je via: <http://www.microsoft.com/windowsazure/support/status/servicedashboard.aspx>

Om specifiekere informatie van de oplossing te bekijken, gaan we aan de slag met Visual Studio en doorlopen we drie stappen: je definieert welke gegevens je wilt verzamelen, je begint met verzamelen en tenslotte verplaats je ze naar een goede opslag om ze veilig te stellen en te benaderen. Windows Azure Diagnostics verzamelt automatisch bepaalde gegevens van je cloudoplossing en als je wilt kun je meer informatiebronnen configureren in je cloudoplossing zelf of in een applicatie buiten de cloud. Om te beginnen heb je Visual Studio 2008 of 2010 nodig. Vervolgens installeer je de Windows Azure SDK en Windows Azure



FIGUUR 2: EEN NIEUWE CLOUD SERVICE.

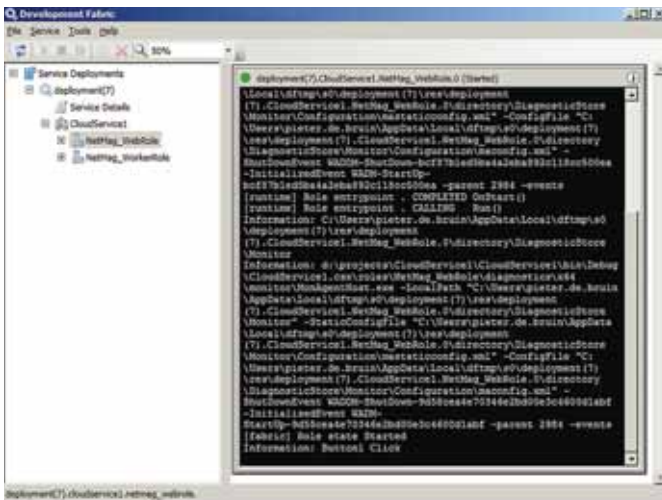
Tools for Microsoft Visual Studio. Nu maak je eenvoudig een nieuwe cloud applicatie: Start Visual Studio, klik File | New project, selecteer Cloud Service | Windows Azure Cloud Service.

Je ziet een scherm waar je rollen voor je cloud service kunt selecteren: selecteer een ASP.NET Web Role en een Worker Role. Nu maakt Visual Studio een nieuwe solution voor je met daarin de webrole, workerrole en een cloud service met configuratieinformatie. De properties van je clouddienst biedt twee tabs: build events en development. De Roles map bevat de rollen in de solution en hier kun je rollen toevoegen of verwijderen. Ook zie je bestanden met servicedefinitie en serviceconfiguratie. Het is nuttig om die bestanden te openen, zodat je ziet welke informatie er standaard in staat. In dit geval is dat een diagnostics connectionstring die verwijst naar je development storage. Als je de solution op Windows Azure wilt laten draaien, zul je deze connectionstring moeten aanpassen.

De default pagina van het webrole project is nog leeg, dus als je een debugger start zie je een wit scherm. Voor de eerste demonstratie voeg je daarom een button toe aan default.aspx. Daarnaast vul je de Button1_Click eventhandler met een trace regel.

```
System.Diagnostics.Trace.WriteLine("Button1 Click", "Information");
```

Start de debugger en je ziet het scherm met de knop. Klik een keer op de knop. Inmiddels zie je ook een Windows Azure icoon in je taakbalk. Je kunt hier met je rechtermuisknop op klikken en

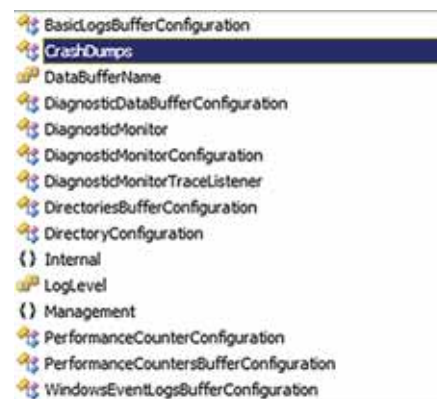


FIGUUR 3: DE DEVELOPMENT FABRIC.

dan zie je opties voor het starten, tonen en het afsluiten van de Development Fabric en Development Storage. De Development Fabric biedt je de Windows Azure Fabric op je ontwikkelmachine. Development Storage biedt Windows Azure Storage op je ontwikkelmachine voor het opslaan van blobs, queues en tabellen. Als je de Development Fabric opent, zie je een boomstructuur met details van de huidige deployment.

Van diverse items kun je details opvragen, maar het gaat ons om de webrole. In de console van de webrole zie je het deployment-proces en onderaan een item met de parameters die je hebt meegegeven aan de trace schrijfactie. Je kunt de informatie die in dit console komt filteren. Klik rechts op de webrole, selecteer Logging Level en kies een niveau dat de gegevens logt waar je in geïnteresseerd bent.

We hebben nu een standaard solution gemaakt met een regel traceinformatie. Je ziet dat de solution vanaf het begin informatie wegschrijft en dat je daar zelf informatie aan kunt toevoegen. En de solution ziet geen verschil of hij in een ontwikkelomgeving op één machine draait of met meerdere instanties op Windows Azure, terwijl daar wel een wereld van verschil tussen zit. De Windows Azure Diagnostics programmeerinterface zorgt hiervoor. Bij het aanmaken van een nieuwe cloud service solution bevatten de roles referenties naar Microsoft.WindowsAzure.Diagnostics.dll, Microsoft.WindowsAzure.ServiceRuntime.dll en Microsoft.WindowsAzure.StorageClient.dll die je standaard kunt vinden in C:\Program Files\Windows Azure SDK\v1.0\ref. De diagnostics namespace biedt de volgende functionaliteit.



FIGUUR 4: MICROSOFT.WINDOWSAZURE.DIAGNOSTICS NAMESPACE.

De DiagnosticsMonitor komen we later nog tegen. De andere klassen geven aan dat je met deze API vooral veel configuratie-informatie kunt benaderen en bewerken en ze bieden alvast een inzicht in de flexibiliteit die je per logging bron hebt. Je ziet hier bijvoorbeeld crashdumps, performance counters en eventlogs. En je hebt al trace meldingen gezien die we naar het Windows Azure log schrijven, of naar de Development Fabric als de solution op een ontwikkelmachine draait. Een groot voordeel van deze configureerbare opzet is dat je het kunt aanpassen terwijl de service beschikbaar blijft. Als je tijdelijk dus iets wilt meten, kun je gewoon de configuratie van een clouddienst in productie aanpassen en wordt de instelling actief.

Een role spreekt de diagnostics monitor dus aan door middel van configuratie en gebruikt local directory storage voor tijdelijke opslag van de gegevens. De role schrijft eigen gegevens zoals traces direct naar de local storage, terwijl de diagnostics monitor ge-

vens uit windowsbronnen zoals IIS Logs, performance counters en eventlogs kan halen en opslaan. De gegevens in local directory storage zijn tijdelijk en niet direct leesbaar. Je zal de informatie dus regelmatig moeten opslaan om het veilig te stellen en toegankelijk te maken voor bijvoorbeeld een website met een service dashboard. Hiervoor gebruik je Windows Azure Storage, de duurzame, schaalbare oplossing voor het opslaan van gegevens in Tables, Queues en Blobs. De Diagnostics Monitor regelt het verplaatsen van gegevens naar Windows Azure Storage voor je op een bepaald moment of ad-hoc. In WebRole.cs wordt standaard een Diagnostics Monitor gestart.

```
public override bool OnStart()
{
    // Create config and set transfers
    DiagnosticMonitorConfiguration diagConfig = DiagnosticMonitor.
    GetDefaultInitialConfiguration();
    diagConfig.Directories.ScheduledTransferPeriod = TimeSpan.From-
    Seconds(15);
    diagConfig.Logs.ScheduledTransferPeriod = TimeSpan.FromSeconds(15);

    DiagnosticMonitor.Start("DiagnosticsConnectionString", diagConfig);
}
```

Hier maak je een nieuwe configuratie voor de diagnostics monitor op basis van de standaard configuratie. Windows Azure kent veel vormen van loggen die je kunt vergelijken met Windows Server. Het loggen van informatie kost inspanning voor een server, dus standaard staan alleen logs aan van Windows Azure, Windows Azure Diagnostic Infrastructure en IIS. Als je ook geïnteresseerd bent in andere logs, zul je dat apart moeten aangeven. In de volgende regels geef je aan wanneer data van de local storage moeten worden verplaatst en je start de monitor met een periodieke gegevensoverdracht.

De cloud service template voegt ook automatisch een tracelister toe aan de web.config, zodat tracemeldingen, waar we eerder mee begonnen, in het Windows Azure log terecht komen.

```
<system.diagnostics>
  <trace>
    <listeners>
      <add type="Microsoft.WindowsAzure.Diagnostics.
      DiagnosticMonitorTraceListener, Microsoft.Windows-
      Azure.Diagnostics, Version=1.0.0.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35" name="Azure-
      Diagnostics">
        <filter type="" />
      </add>
    </listeners>
  </trace>
</system.diagnostics>
```

Dit is de basis van Windows Azure Diagnostics: De Visual Studio template voor een cloud service plaatst automatisch code in webrole.cs en web.config, zodat Azure diagnostics verzameld worden zonder dat je er iets voor hoeft te doen.

Nu gaan we verder met het configureren van bronnen die je wilt loggen. Dit is een uitgebreidere OnStart methode in WebRole.cs.

```
// Gather small Crash Dumps
CrashDumps.EnableCollection(true);

// Create config and set transfers
DiagnosticMonitorConfiguration diagConfig = DiagnosticMonitor.
GetDefaultInitialConfiguration();
diagConfig.Directories.ScheduledTransferPeriod = TimeSpan.From-
Seconds(15);
diagConfig.Logs.ScheduledTransferPeriod = TimeSpan.FromSeconds(15);

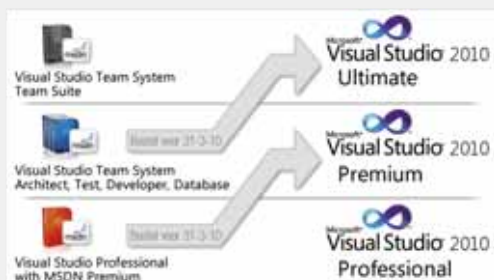
// Gather Performance Counters
PerformanceCounterConfiguration procTimeConfig = new Performance
CounterConfiguration();
```

(Advertentie)

computercollectief
computerbooks & software
comcol.nl

Nú is het moment om Visual Studio met MSDN te kopen

Visual Studio 2010 wordt begin april 2010 verwacht. U kunt veel voordeel krijgen door nu Visual Studio 2008 met MSDN te bestellen. U krijgt dan Visual Studio 2010 zodra deze leverbaar is en, vanaf MSDN Premium, een meer geavanceerde versie, die normaal honderden tot duizenden euro's duurder is. Bovendien krijgt u voor € 100,- boeken cadeau!



Voor meer informatie: bezoek de Visual Studio 2010 themapagina: comcol.nl/thema/visual_studio_2010



17652-A2
Introducing Windows Azure € 31,90 This book provides you with a rich set of example projects that cover the core skills you must learn to be a Windows Azure developer.



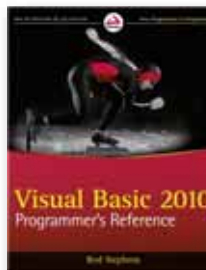
15529-B8
Pro Silverlight 3 in C# € 39,90 An in-depth exploration of Silverlight for professional developers. By the end of this book, you'll have a solid understanding of the complete Silverlight platform.



17289-A1
Introducing .NET 4.0: with Visual Studio 2010 € 31,90 Brings you up to speed on the latest changes in .NET 4.0 and VS2010 and takes a look at ASP.NET MVC, Azure & Silverlight.



17184-A1
Cloud Computing with the Windows Azure Platform € 31,90 Roger Jennings clears away the clouds and gets you started.



1424-I1
Visual Basic 2010 Programmer's Reference € 31,90 Includes the latest enhancements to the VB language. Expected Feb. 2010.



17363-A0
Microsoft Forefront Threat Management Gateway (TMG) Administrator's Companion € 46,90

```

procTimeConfig.CounterSpecifier = @"\\Processor(*)\\% Processor
Time";
procTimeConfig.SampleRate = System.TimeSpan.FromSeconds(5.0);
diagConfig.PerformanceCounters.DataSources.Add(procTimeConfig);

// Gather Windows Eventlog
diagConfig.WindowsEventLog.DataSources.Add("Application!*");
diagConfig.WindowsEventLog.ScheduledTransferPeriod = TimeSpan.
FromMinutes(5);

// Gather infrastructure logs
diagConfig.DiagnosticInfrastructureLogs.ScheduledTransfer-
LogLevelFilter = LogLevel.Error;
diagConfig.DiagnosticInfrastructureLogs.ScheduledTransferPeriod =
TimeSpan.FromMinutes(5);

// Start the monitor
DiagnosticMonitor.Start("DiagnosticsConnectionString", diagConfig);

```

De eerste regel geeft aan of je de hele crashdump wilt opslaan. Daarna voegen we een performance counter toe aan de Diagnostics Configuration. De counterspecifier is de naam van de performance counter zoals je die kunt vinden in de bekende Windows Performance Monitor. Zoals je ziet kun je de uploadfrequentie vrij invullen en is die hier van het windows eventlog anders dan van de standaard logs. De laatste datasources zijn het application eventlog van Windows en de infrastructure logs. Vervolgens geven we opdracht om te beginnen met de gegevensoverdracht naar Windows Azure Storage.

Natuurlijk kun je ook gegevens verplaatsen op een moment dat je dat zelf wilt.

```

private void Transfer()
{
    CloudStorageAccount account = CloudStorageAccount.Development-
StorageAccount;
    var deploymentdm = new DeploymentDiagnosticManager(account,
deploymentId);
    var roleinstancedm = deploymentdm.GetRoleInstanceDiagnostic
Manager(roleName, roleInstanceName);
    var dataBuffersToTransfer = DataBufferName.Logs;

    OnDemandTransferOptions transferOptions = new OnDemand-
TransferOptions();
    transferOptions.From = DateTime.MinValue;
    transferOptions.To = DateTime.UtcNow;
    transferOptions.LogLevelFilter = LogLevel.Critical;

    Guid requestID = roleinstancedm.BeginOnDemandTransfer(data-
BuffersToTransfer, transferOptions);
}

```

Hier maak je een deployment diagnostics manager die verbindt met bijvoorbeeld de development fabric. Vervolgens kan die een role instance diagnostics manager maken om met een specifieke role instance te communiceren op basis van een specifieke naam of op basis van collecties. Daarna geef je aan wat je wilt verplaatsen, op welke moment en eventueel een filter op de data. Tenslotte start je de transfer en krijg je een requestID terug op basis waarvan je later de resultaten kunt correleren. Je kunt een transfer natuurlijk ook periodiek laten uitvoeren, maar dat ligt aan de functionele behoefte.

Net hebben we een filter gezet op de gegevens die we willen verplaatsen. Nu kun je altijd een filter zetten op transferoptions, omdat de API het ondersteunt, maar het wordt niet altijd toegepast. Zie het zo: een trace- of logmelding heeft een loglevel, waarbij je kunt aangeven of het een foutmelding is, een kritische melding of een informatieve melding. Deze meldingen worden gestructureerd opgeslagen in tabellen. Daarnaast heb je andere vormen van log-

ging: denk bijvoorbeeld aan IIS logs of je eigen log.txt. Deze bestanden zijn ongestructureerd en worden opgeslagen in blobs. Voor deze ongestructureerde data wordt het filter niet toegepast maar wordt het hele log verplaatst.

Zoals je begrijpt kan de hoeveelheid loggegevens snel groeien. Daarom is het belangrijk dat je goed nadenkt over het gebruik van je gegevens en je op basis daarvan bepaalt welke gegevens je wilt opslaan, verplaatsen en bewaren. Aan zowel het opslaan als het verplaatsen van gegevens zijn immers kosten verbonden in een cloud model.

Als je eenmaal gegevens verplaatst hebt, kun je op de gewone manier communiceren met Tables en Blobs om resultaten te visualiseren. De Windows Azure Training Kit biedt voorbeelden om gegevens uit Windows Azure storage te lezen en te tonen in bijvoorbeeld ASP.NET. Een tool die je zonder programmeren toegang geeft tot Windows Azure storage is Cloud Storage Studio van Cerebrata.



FIGUUR 5: CERABRATA CLOUD STORAGE STUDIO.

Dit artikel toont het begin van de mogelijkheden voor ontwikkelaars en integrators om Windows Azure tools te maken en handelingen te automatiseren. Microsoft.WindowsAzure.Diagnostics is een interessante API waar je mogelijkheden mee krijgt die nog niet beschikbaar zijn in een gebruikersinterface. Als je oplossingen voor Windows Azure maakt is de Windows Azure SDK onmisbaar met bijvoorbeeld de Service Management API waarmee je deployments automatiseert.

Links

- Windows Azure SDK
<http://msdn.microsoft.com/en-us/library/dd179367.aspx>
- Windows Azure for Developers
<http://www.microsoft.com/windowsazure/getstarted/>

.....
Pieter de Bruin, werkt voor Avanade, de wereldwijde samenwerking tussen Accenture en Microsoft. Hij focust zich op maatwerkontwikkeling en integratie. Daarnaast is hij Microsoft Most Valuable Professional in Visual Studio Team System. Hij is bereikbaar via pieter.de.bruin@avanade.com.



Jurriaan
Database admin

Edward
Marketinganalist



*Intelligente mensen.
Intelligente beslissingen.
Efficiënt zakendoen.*

Wat gebeurt er als u de vertrouwde Microsoft® Office functionaliteit combineert met de krachtige analysemogelijkheden van SQL Server® 2008 en de functies voor informatie-ontsluiting van SharePoint®?

Dan heeft u mensen die data van meerdere bronnen analyseren en zelfstandig in rapporten kunnen verwerken. Real-time, goed geïnformeerde beslissingen maken zonder dat IT eraan te pas hoeft te komen? Dat is méér dan de som der delen.

Lees meer over de efficiencyvoordelen van het maken van betere beslissingen. Ga naar itseverybodysbusiness.nl/beslissen