



Azure Functions

Michael Weinand

About Me

- ▶ Michael Weinand
- ▶ Director of Software Engineering
Looking Glass Investments
- ▶ @mweinand



Background [Work] on Azure PaaS

How we got to Functions

Background Tasks

- ▶ Image processing or other CPU-intensive work
- ▶ Queue processing
- ▶ RSS aggregation
- ▶ File maintenance, such as aggregating or cleaning up log files
- ▶ Other long-running tasks that you want to run in a background thread, such as sending emails
- ▶ Data transformation

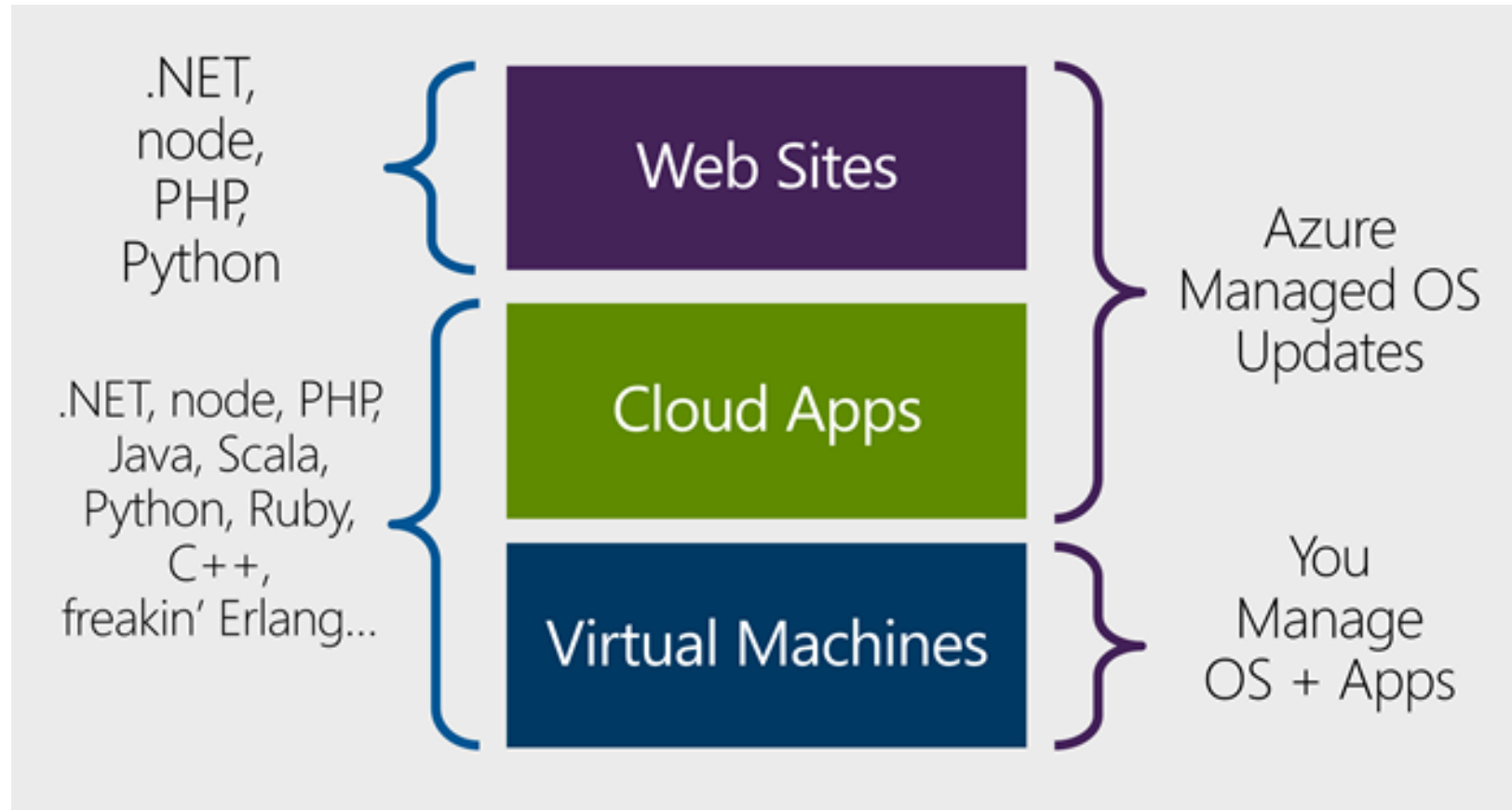
Azure Cloud Services

- ▶ Web and Worker roles
- ▶ Special Visual Studio Project Type (ccproj)
 - ▶ Only natively supports .NET, but can support other technologies
- ▶ Proprietary Deployment Package/Process (ccpkg)
 - ▶ 5+ minute deployments
 - ▶ Some configuration option changes require redeploy (VM size)
- ▶ Scale manually or based on CPU or Azure Storage Queue
- ▶ Runs on Virtual Machines
 - ▶ Can remote desktop
 - ▶ Microsoft handles hardware/updates/migration/failures

Windows Azure Web Sites

- ▶ Web Roles without all the hassle
- ▶ Platform as a Service (PaaS) for web applications
 - ▶ .NET, PHP, Node.js, [Python], [Java]
- ▶ Simplified deployment
 - ▶ MS Deploy, FTP, Git
- ▶ Scale manually or based on CPU/memory
- ▶ No access to underlying VMs
- ▶ Tiers (Hosting Plan) - free, shared, standard, premium
 - ▶ Lower cost options vs Cloud Services
 - ▶ Shared resources - multiple apps can share VM/payment plan

Continuum of Control



Azure WebJobs

- ▶ Azure Web Sites version of Cloud Services Worker Roles
- ▶ Runs on same resources as Azure Web Sites
- ▶ Run [almost] any script or executable
 - ▶ exe/.NET console applications
 - ▶ Windows Cmd
 - ▶ Windows Batch
 - ▶ Bash
 - ▶ PHP
 - ▶ Python
 - ▶ NodeJS

Azure WebJobs SDK

- ▶ Azure WebJobs SDK provides .NET plumbing for responding to events
- ▶ Open Source: <https://github.com/Azure/azure-webjobs-sdk>
- ▶ Triggers and Bindings:
 - ▶ Azure Blob Storage
 - ▶ Azure Queue Service
 - ▶ Azure Service Bus
 - ▶ Timers
- ▶ Extensible: <https://azure.microsoft.com/en-us/blog/extensible-triggers-and-binders-with-azure-webjobs-sdk-1-1-0-alpha1/>

Azure App Service

- ▶ Evolution of Azure Web Sites
- ▶ Includes:
 - ▶ Web Apps
 - ▶ Mobile Apps
 - ▶ API Apps
 - ▶ Logic Apps
- ▶ All on the same basic infrastructure
- ▶ Web Hosting Plan => App Service Plan

Azure Functions

An event driven, compute-on-demand experience

Serverless Computing

- ▶ Cloud computing execution model where the service provider manages all of resources required to serve requests
 - ▶ No manual provisioning of VMs
- ▶ Evolution of Platform as a Service (PaaS) => Function as a Service (FaaS)
- ▶ Pay-per-execution billing model
 - ▶ GB of memory/s
 - ▶ Per 100 ms of CPU time
- ▶ Scales automatically to handle number of requests

What are Azure Functions?

- ▶ Azure's Serverless Computing option
- ▶ Built on Azure WebJobs/App Service
- ▶ Introduces
 - ▶ Function Apps
 - ▶ Dynamic Service Plans
 - ▶ HTTP Triggers

Demo

Azure Functions “Hello World!”

Triggers and Bindings

- ▶ Timer
- ▶ Webhook
- ▶ API Hub
- ▶ SendGrid
- ▶ Blob
- ▶ DocumentDB
- ▶ EventHub
- ▶ HTTP
- ▶ Queue
- ▶ ServiceBus

function.json

```
{{"bindings": [  
  {  
    "type": "queueTrigger",  
    "name": "message",  
    "direction": "in",  
    "queueName": "samples-csharp"  
    "connection": "...."  
  },  
  {  
    "type": "blob",  
    "name": "result",  
    "direction": "out",  
    "path": "samples-output-csharp/{Id}"  
  }  
]}
```

run.csx

```
public static void Run(Message message, out string  
result, TraceWriter log)  
{  
    log.Info($"C# Queue trigger function processed  
message: {message.Id}");  
    result = message.Value;  
}  
  
public class Message  
{  
    public string Id { get; set; }  
    public string Value { get; set; }  
}
```


run.csx?

- ▶ CSX files are C# scripts
- ▶ Executes the **public static void Run()** method
- ▶ Powered by Roslyn
- ▶ Can be executed locally by ScriptCS

Demo

ScriptCS Hello World

Dependency Resolution

- ▶ Function app can download dependencies:
 - ▶ NuGet (project.json) - Automatic
 - ▶ NPM (project.json) - KUDU npm install

Deployment and Tools

- ▶ Deployment is the same as Azure App Service
 - ▶ FTP
 - ▶ Git
 - ▶ TFS/VSTS
 - ▶ MSDeploy
 - ▶ Octopus Deploy
- ▶ Tools
 - ▶ Azure Functions Portal
 - ▶ Kudu

Demo

Function Portal/KUDU

Scaling & Pricing

- ▶ Pricing

- ▶ <https://azure.microsoft.com/en-us/pricing/details/functions/>

- ▶ Scaling

- ▶ Memory Tier Size

- ▶ # of instances controlled by Azure

Demo

Lots of Queue Messages!

Limitations/Other Notes

- ▶ Does not support PhantomJS/Selenium due to local address connection limitations
- ▶ Once you choose a Dynamic plan or a Standard plan, you cannot change
- ▶ CSX tooling/Testing Locally - possible but hoops
- ▶ **Still in Preview! Things are changing rapidly.**

Resources

- ▶ <https://github.com/mweinand/azurefunctionstalk>
- ▶ <https://azure.microsoft.com/en-us/blog/introducing-azure-functions/>
- ▶ <https://github.com/projectkudu/kudu/wiki/Azure-Web-App-sandbox>
- ▶ <https://github.com/Azure/azure-webjobs-sdk-script>
- ▶ <http://scriptcs.net/>
- ▶ <https://azure.microsoft.com/en-us/documentation/articles/functions-compare-logic-apps-ms-flow-webjobs/>
- ▶ <http://www.biztalk360.com/integrate-2016-resources/leveraging-azure-functions-azure-app-service-integration-scenarios/>
- ▶ <http://www.hanselman.com/blog/ProjectlessScriptedCWithScriptCSAndRoslyn.aspx>