

Windows Azure for Research—Training Description

August 2013 – DRAFT FOR INFORMATION PURPOSES ONLY – ACTUAL CONTENT MAY CHANGE

This document outlines the default Windows Azure training course contents for presentation at Microsoft Research workshops to select research audiences. The course is especially designed for practicing research scientists with at least basic software development skills.

The course includes hands-on labs (“HOLs”) – these are practical exercises that attendees complete during the training. Where applicable, course materials will be provided for certain sections of the course.

Day 1: focuses on acquiring a general understanding of Windows Azure. Day 2: explores how to use Windows Azure at scale. There is flexibility in the course structure and tailoring is possible to emphasizing or de-emphasizing selected topics, depending on the specific audience and context.

Contents

Windows Azure for Research—Training Description	1
Target Audience	2
Prior Knowledge/Pre-requisites	2
Learning Objectives	2
Teaching Context.....	3
Windows Azure	3
Day 1: Windows Azure for Research Introduction	4
0. Welcome and Introductions [15 minutes].....	4
1. Overview of Cloud Computing and Windows Azure [1.5 hours]	4
2. HOL: Getting Started with Windows Azure Websites and Virtual Machines [2 hours]	4
3. HOL: Creating a Virtual Machine and Running Simple Data Analysis with IPython or Wakari.io [1 hour]	5
4. Windows Azure Storage, SQL Databases, and Data Management [1 hour]	5
5. DEMO: Building a Classic Scalable Application with Visual Studio and the Windows Azure SDK (BLAST) [45 min]	6
6. Day 1 Summary.....	6
Day 2: Windows Azure for Research: Compute and Data Beyond the Desktop	6
7. Recap and Welcome to Day 2 [15 minutes]	6
8. From Desktop to Cloud—Visualization Using Excel, Data Explorer, Power Map for Data Analysis on Windows Azure [1 hour]	6
9. Introduction to Big Data and Big Compute on Windows Azure [1 hour]	6
10. Data Acquisition for Devices and Always-on Services [1 hour]	6
11. Demo: Running Parametric Sweeps Using R, Python, and HPC [1 hour].....	7
12. HOL: IPython Cluster [1 hour].....	7
13. Creating an HDInsight Cluster (Demo) [30 minutes]	7
14. Demo: Architecture Patterns for the Cloud STORM, Messaging, and Data Caching [30 minutes]	7
15. Run Spark and Shark in a Linux Cluster on Windows Azure (optional/ demo only) [30 minutes]	8
16. HOL: Putting it Together, Build a Service Using Live Data Acquisition and Analysis [1 hour]	8
17. Discussion and Wrap-up [1 hour]	8

Target Audience

- Active scientist and eScience researchers who can code, who will soon code; and
- Active computer science (CS) researchers who are working with such scientists and handling the software for those scientists
 - Not for educators specifically, though also welcome (if space) if they don't have access to cloud/Windows Azure education classes elsewhere
- Practising researchers
 - Priority given to faculty, research assistants, post-docs
 - Suitable for grad students, master's students to fill extra places (if there is space)
 - Undergraduates could be optional to attend if the course is not full
 - If they are doing undergraduate research—but typically not general CS undergraduates

This class is targeted by design. Microsoft/others offer various training options for other audiences.

Prior Knowledge/Pre-requisites

- Can program in languages such as Java, Python, C#, or C. Course examples will be predominantly in Python and/or Java. You don't need to be an expert programmer, but if you are programming in your science research today, that is ideal and preferred for getting most from the class.
- We prefer basic exposure to cloud computing ("101" level), but certainly no real expertise required. No prior cloud usage experience required. We'll cover that, as well as our particular cloud technology. We'll take people who have heard of cloud computing and are curious or unsure about its value for their research, or equally those who are experienced in using and developing to another cloud system.
- No prior experience of Windows Azure is required—though some exposure can help us to incorporate more advanced topics. In general, we assume attendees have no prior experience with Windows Azure.

Learning Objectives

After attending this course, successful attendees will be able to:

- Explain what cloud computing is
- Explain why and when to use cloud computing for scientific research:
 - Including examples of where it has been successfully applied in specific domains, such as life sciences, environmental sciences, and social sciences
- Understand and be practiced in the major design patterns for successful cloud applications, in particular being able to:
 - Deploy websites by using Wordpress or CMS (Content Management System)
 - Create virtual machines and deploy IPython
 - Manage storage by using Windows Azure blobs
 - Understand how to link client applications with Windows Azure
 - Create and develop SQL Azure databases
 - Create and deploy mobile services
 - Create and run scalable computing applications by using Visual Studio, Python, R, and MATLAB
 - Build an end-to-end cloud-based system for sensor data acquisition, data processing, and REST service for access from any device
- Run the researcher's own app/service on Windows Azure—with a positive training outcome directly relevant to researcher
- Know what next steps to take—being able to make a personalized action plan for using Windows Azure for the researcher's own work

Teaching Context

- You can expect approximately 50 attendees per class event for the global program of classes, though there may be regional/cultural variances in practice—the number is a balance of pedagogical, technical, and other considerations.
- Attendees will use their own laptop; this is primarily a “hands-on” experience. Attendees will do exercises on their laptop, accessing the cloud. Good connectivity is vital and this will require necessary set-up by the local host prior to the event.
- The attendee’s laptop does not need to have the Windows operating system installed—but they do need a modern Internet browser. Windows Azure is agnostic and does not require Windows—we will show both Windows and Linux virtual machines (VMs) in the cloud. In principle, this means the attendees could be remote, but the class is designed for an in-person learning experience.

Windows Azure¹

- Windows Azure is an open and flexible cloud platform that enables you to quickly build, deploy, and manage applications across a global network of Microsoft-managed data centers. You can build applications by using any language, tool, or framework. And you can integrate your public cloud applications with your existing IT environment.
- Windows Azure delivers a 99.95 percent monthly service level agreement (SLA) and enables you to build and run highly available applications without focusing on the infrastructure. It provides automatic operating system (OS) and service patching, built in network load balancing, and resiliency to hardware failure. It supports a deployment model that enables you to upgrade your application without downtime.
- Windows Azure enables you to use any language, framework, or tool to build applications. Features and services are exposed by using open REST protocols. The Windows Azure client libraries are available for multiple programming languages, and are released under an open source license and hosted on GitHub.
- Windows Azure enables you to scale your applications to any size easily. It is a fully automated self-service platform that allows you to provision resources within minutes. Elastically grow or shrink your resource usage based on your needs. You pay only for the resources your application uses. Windows Azure is available in multiple data centers around the world, enabling you to deploy your applications close to your customers.
- Windows Azure delivers a flexible cloud platform that can satisfy any application need. It enables you to reliably host and scale out your application code within compute roles. You can store data by using relational SQL databases, NoSQL table stores, and unstructured blob stores, and optionally use Hadoop and business intelligence services to mine your data. You can take advantage of Windows Azure’s robust messaging capabilities to enable scalable distributed applications, as well as deliver hybrid solutions that run across a cloud and on-premises enterprise environment. Windows Azure’s distributed caching and content delivery network (CDN) services allow you to reduce latency and deliver great application performance anywhere in the world.

¹ From: <http://www.windowsazure.com/en-us/overview/what-is-windows-azure/>

Day 1: Windows Azure for Research Introduction

0. Welcome and Introductions [15 minutes]

Brief introductions (name, institution and department, and field of study).

Introduce agenda for Day 1.

Verify everyone has connectivity and power.

1. Overview of Cloud Computing and Windows Azure [1.5 hours]

Overall introduction to the cloud, why you would use it for research, and types of workload you would use it for. Examples of how corporations and scientists are using the cloud. Describe where and how data is stored and services are provisioned within a data center by providing a virtual tour of the data center through video and commentary.

This introduction will focus on illustrating differences between platform as a service (PaaS) and infrastructure as a service (IaaS) and the different types of workloads that can be run in Windows Azure. It will describe details from the data center up and the layout to what services are available. It will describe exemplars such as the genomic study examples, weather simulations, and other highly scalable descriptions of workloads. Six demos:

- a) Managing storage with free tools like Cerebrata Azure Explorer
- b) Building a hello Azure application on a website, the simplest ASP.net, PHP, Node.js. Paper publishing/simple app that pulls data from storage (data file upload/viewer example), commercial CMS
- c) BLAST demo: application patterns with web role and worker roles
- d) Fetch Climate demo
- e) Weather demo: architecture illustrating ease of scalability and the value of a live service
- f) Mobile services: sending a notification, iOS example

Emphasis on showing broader topics on what Windows Azure can do for researchers.

2. HOL: Getting Started with Windows Azure Websites and Virtual Machines [2 hours]

The overview will describe how both Linux and Windows are first-class citizens of Windows Azure. In this session, users will first experience Windows Azure for themselves with a very quick hands-on lab (HOL) during which they sign in, see the management console, and build their first Windows Azure website.

We'll describe and demonstrate:

Websites

Introductions will be made to Windows Azure websites showing the Gallery Images for Windows Azure websites (WAWS) and what applications can be run. Make sure participants have Windows Azure accounts running for first HOL.

- a) Build a WordPress site for publishing blogs, etc.
- b) Build a website with a GitHub/Bitbucket source.
- c) Overview of Windows Azure websites showing how to collect metrics; configure source control; scale, set up connections to databases; and add handler mappings to run Python, Node.js, and PHP.

Virtual Machines

- a) The platform image gallery
- b) The vmdepot and how to contribute
- c) IPython VM creation demo for Portal and Windows Azure command-line interface (CLI)
- d) Different images available for science
- e) Demo: capturing a Windows Virtual Machine with sysprep
- f) Demo: capturing a Linux virtual machine with waagent
- g) Adding data disks to both Linux and Windows VMs
- h) We'll cover concepts in slides and demos, availability sets for resilience

3. HOL: Creating a Virtual Machine and Running Simple Data Analysis with IPython or Wakari.io [1 hour]

- a) Create an image from the VM depot using the CLI (Command Line Interface). Lab will walk through installing the Windows Azure CLI on Linux/Mac/Windows and the command to run to create a small instance VM Depot image of IPython
- b) Install the Python SDK on the VM and create a script to download and upload files and create containers in blob storage
- c) Run sample storage commands in IPython notebooks.
- d) Run data analysis example using Panda and Scikit-learn.
- e) Create an account on SUSE Studio and create a Python VM (optional)

4. Windows Azure Storage, SQL Databases, and Data Management [1 hour]

Introduction to the benefits of using cloud-based storage solutions for research applications.

Overview showing asynchronous nature of storage. Access speed performance guidelines due to flat network storage. (Blob Meta data versus SQL Azure) Discussion of database as a service. This will show Windows Azure Storage at a high level. It will describe volatile storage in PaaS and persistent storage in IaaS and what underlies both.

Introduction to Blob Storage:

- a) Differences between block blobs and page blobs
- b) Storage Services SDK with Python (Demos)
- c) Storage Services REST API and Shared Key Authentication (Demo optional or cover in slide)
- d) Securing storage with Shared Access Signatures (Demo)

Introduction to Table Storage and why you would use it. Flattening out the hierarchy and picking your Rowkeys and PartitionKeys correctly. Demos will include:

- a) Creating a table for audit records in code
- b) Show dynamic nature of tables and add additional columns

Description of Storage Queues and how and why you would use them. Illustrating that PaaS notes has volatile storage.

Demo:

- a) Setting up LocalStorage on a web role
- b) Setting up a CloudDrive on a web role and mounting

5. DEMO: Building a Classic Scalable Application with Visual Studio and the Windows Azure SDK (BLAST) [45 min]
 - a) Demo: Download and import of a .publishsettings file
 - b) Demo: Creating a cloud service project
 - c) Through the demo, describe elements of ServiceConfiguration.cscfg and .csdef files and how they describe cloud services and roles.
 - d) Demo: Adding diagnostics to service and consuming by using Cerebrata Azure Management Studio
 - e) Demo: Adding a startup task to the project, caching and/or remote desktop
 - f) Demo: Adding web and worker roles to a project and using Windows Azure Queue Storage

6. Day 1 Summary

Day 2: Windows Azure for Research: Compute and Data Beyond the Desktop

7. Recap and Welcome to Day 2 [15 minutes]

Refresh participants on what was covered the previous day.

Introduce agenda for Day 2.

Short Q&A.

8. From Desktop to Cloud—Visualization Using Excel, Data Explorer, Power Map for Data Analysis on Windows Azure [1 hour]

Show how desktop application, such as Excel, can be used as a client for Cloud Services:

- a. Geoflow (Power Map) demo
- b. Data explorer demo.
- c. Power pivot demo (Xvelocity Column based DB).
- d. Python and Excel demo.

9. Introduction to Big Data and Big Compute on Windows Azure [1 hour]

Introduce where Big Data and Big Compute are applicable with compelling real-world scientific examples to set up Day 2. Highlight the benefits of Azure Big Compute versus “traditional” high performance computing (HPC) systems.

- a) Demos on scale out options
- b) MATLAB parametric sweep demo using HPC
- c) Demo running R and/or Python

10. Data Acquisition for Devices and Always-on Services [1 hour]

- a) Catalog of examples

- b) Internet of Things
- c) REST
- d) How to scale data acquisition
- e) Sensor
- f) Messaging using service bus, and service bus relays
- g) Demo: Raspberry Pi

11. Demo: Running Parametric Sweeps Using R, Python, and HPC [1 hour]

Describe how easy it is to run parametric sweep jobs using researchers' favorite tools: R, Python, and MATLAB. Explain the different types of HPC available including scheduler SDK, burst to Azure, and a full IaaS implementation. Show the power of HPC as a demo.

- a) MATLAB parametric sweep demo using pre-deployed HPC
- b) Running R and/or Python on Windows Azure
- c) Share results using websites

12. HOL: IPython Cluster [1 hour]

- a) Downloading and installing the CLI running commands to install an IPython cluster—deploy two nodes of a four-node cluster with the IPython VM Depot image
- b) As above, but using the Service Management API Python SDK wrapper to deploy the cluster
- c) Run Python script to download files from blob storage to the cluster
- d) Backup files between storage accounts using Copy Blob API in Python

13. Creating an HDInsight Cluster (Demo) [30 minutes]

Show how Microsoft Excel can be used to drive Big Data solutions. Explain what HDInsight is and how this is part of Microsoft's Big Data Strategy. Explain the architecture and interfaces to create an HDInsight cluster. Introduce Spark and explain the difference between this and HDInsight. Go through a use case of when each one would be used and what the major differences are.

- a) Create a four-node HDInsight cluster in subscription
- b) Create a simple map-reduce of an ecommerce dataset
- c) Write a Hive query against the data set
- d) Write a "pig Latin" expression against the dataset
- e) Using pre-deployed spark/shark instance, demo run a simple query against the same dataset as HDInsight first using Scala Map-Reduce then using Shark

14. Demo: Architecture Patterns for the Cloud STORM, Messaging, and Data Caching [30 minutes]

- a) How to deal with data in the cloud, advanced topics, performance optimization
- b) Prep for real-time data analysis
- c) The same demo using Raspberry Pi

15. Run Spark and Shark in a Linux Cluster on Windows Azure (optional/ demo only)
[30 minutes]

Introduce Spark and explain the difference between this and HDInsight. Go through a use case of when each one would be used and what the major differences are.

- a) Install Mesos on a new Linux virtual machine
- b) Install shark and spark
- c) Create another virtual machine and add to the Spark cluster
- d) Run a simple query against the same dataset as HDInsight first using Scala Map-Reduce then using Shark

16. HOL: Putting it Together, Build a Service Using Live Data Acquisition and Analysis [1 hour]

- a) End-to-end exercise
- b) GPS data from China, mobile devices as clients
Vendor to author; data feed, data analysis, and dashboarding; emulating the many devices; D3 for dashboarding

17. Discussion and Wrap-up [1 hour]

- a) Recap what participants have learned. Summarize more advanced topics to follow up individually with people afterwards (e.g., STORM), Q&A
- b) Next steps: Visiting researcher program, RFP applications, community activities/resources, etc.
- c) Ask audience the type of projects they would be interested in doing
- d) Course evaluation questionnaire
- e) Course completion certificate awarded upon handing in survey questionnaire!

End of course