Getting Started w/Azure & PowerShell

March 23, 2014
Azure & PowerShell

Getting Started!
Ron Bokleman ron.bokleman@bluemetal.com Senior Architect Cloud & Services.
The Cloud & Services team are practitioners of Lean Engineering, a high velocity product development process that applies Lean methodology, service oriented patterns and practices and cloud platform capabilities for the design and development of modern applications for the enterprise.

Ron Bokleman has been in the software industry for 30 years having worked for both ISVs and large corporations such as Microsoft as a Senior Consultant and Technical Architect in the Northeast. Ron has deep industry experience in Banking, Financial Services and Construction.
Before we get going...

How many consider themselves “Developers”?

How many consider themselves more of an “Infrastructure / IT Pro” person?
Before we get going...

How many did your pre-session homework and either created or have access to a non-Production “Sandbox” Azure subscription?

Reminder: 101 Level Session, however, if you want more we can plan follow on sessions!
Agenda

• Provision (homework) an Azure Subscription
• Assign Administrators
• Preparing Your Desktop
• Getting Connected
• Executing Your First PowerShell Commands
• Azure IaaS Building Blocks
• Keeping up to date
• Resources
Provision an Azure Subscription

Free Trial!
Provision an Azure Subscription

- Azure IaaS/PaaS
  - In a Software Defined Data Center, all elements of the infrastructure — networking, storage, CPU and security — are virtualized and delivered as a set of services.
    - Software Defined & Virtualized Compute Resources (Virtual Machines)
    - Software Defined Networking (SDN)
    - Software Defined Storage (SDS)
    - Accessible API Layer & Tools for management and automation software, enabling an administrator to provision, control, and manage each software-defined data center component
      - Rest
      - PowerShell
      - Node.js
  - Azure itself, is a Software Defined Data Center (SDDC)!
  - A SDDC that can be utilized by establishing one or more subscriptions optionally as an extension of existing physical Data Center(s).
  - Sign Up
    - http://azure.microsoft.com/
Azure Management Console

- [https://manage.windowsazure.com](https://manage.windowsazure.com)
- Sign In using the MicrosoftID used to create the subscription.

- Question: How many like the Azure Management Console?
Azure *Preview* Management Console

- [https://portal.azure.com](https://portal.azure.com)
- Preview
Assign Administrators

Free Trial!
Assign Administrators

- Scroll down to the bottom to locate Settings
- Select Administrators
- Add:
  - Microsoft ID
  - Outside your organization
  - OrgID (O365/Azure AD)
  - Azure Active Directory Account
  - Unique to your subscription
Prepping Your Desktop
For Azure & PowerShell
Prepare Desktop

• How to install and configure Azure PowerShell
  • http://azure.microsoft.com/en-us/documentation/articles/install-configure-powershell/#Install

• Web Platform Installer
  • http://go.microsoft.com/fwlink/p/?linkid=320376&clcid=0x409
  • $PSVersionTable.PSVersion

• Windows 8.1 w/PowerShell v4.0 recommended.
• Windows Server 2012 or R2 recommended.
  • OS X with node.js

• Locate and launch the PowerShell ISE
Getting Connected To Azure
Using PowerShell v4.0 and above...
Getting Connected To Azure Subscription

• Start PowerShell ISE as an Administrator
  • Hint: You can pin it to the Windows taskbar for easy access.

• Get-ExecutionPolicy / Set-ExecutionPolicy
  
  Get-ExecutionPolicy -Scope LocalMachine
  Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope LocalMachine
  Update-Help

• Create C:\PSScripts folder...
• Create C:\PSScripts\Azure folder...
Getting Connected To Azure Subscription

• Establish Subscription Session

Get-AzurePublishSettingsFile
  • Opens browser…
  • Select subscription…
  • Download file [SubscriptionName-MM-DD-YYYY-credentials.publishsettings] to local disk C:\PSScripts\Azure…

Import-AzurePublishSettingsFile
  ‘C:\PSScripts\Azure\SubscriptionName-MM-DD-YYYY-credentials.publishsettings'

• Hint: If the importation doesn't work, or you simple want to start over, you can delete the files in C:\Users\<username>\AppData\Roaming\Windows Azure Powershell\*.*
Executing Your First...
PowerShell Commands
Executing Your First Azure PowerShell Commands

• Subscription?
  # List Current Subscription
  Get-AzureSubscription -Current

• Azure PowerShell Version?
  (Get-Module Azure).Version

• Change between subscriptions?
  # Select Subscription
  Select-AzureSubscription -SubscriptionName $Subscription
  # List Subscription
  Get-AzureSubscription -Current
Azure IaaS Building Blocks
**Azure IaaS Building Blocks**

- **Azure IaaS**
  - Subscription(s)
    - Billing Container
  - Region(s)
    - The **Region** (Location) is the physical location on a larger scale than an affinity group.
  - Affinity Group(s)
    - Affinity groups are a way you can group your cloud services by proximity to each other in the Azure datacenter in order to achieve optimal performance. When you create an affinity group, it lets Azure know to keep all of the services that belong to your affinity group as physically close to each other as possible. For example, if you want to keep the services running your data and your code together, you would specify the same affinity group for those cloud services. They would then run on hardware that is located close together in the datacenter. This can reduce latency and increase performance, while potentially lowering costs.
    
    - Affinity groups are defined at the subscription level and the name of each affinity group must be unique within the subscription. Each affinity group you create is tied to a Region (which is the Location). Specify the same region when creating your affinity group and your virtual network.
Azure IaaS Building Blocks

- Storage Account(s)
  - Containers
- Virtual Network(s)
  - DNS – Azure provided or your own, for external you either need your own or ISP hosted or even Route53 in AWS
  - Subnets – Your own design using private network address space
    - Includes Multi-Site
- Load Balancing
- Cloud Service(s)
  - The Cloud Service DNS Name is the global DNS name that becomes part of the URI that's used to contact the virtual machine. You'll need to come up with your own cloud service name because it must be unique in Azure.
- VM Image(s)
  - Licensing
  - Endpoints
PowerShell – VM Images?

• Get all VM images for Windows Server 2012 R2?

```
(Get-AzureVMImage | where {$_.Label -like "Windows Server 2012 R2*"} | Sort PublishedDate -Descending)[0].ImageName
```

• A word about licensing...
• VM sizing...
PowerShell VM Role Sizes?

- PowerShell Exercise
Keeping Updated
Keeping Updated

- What version of Microsoft Azure PowerShell do we have?
- Updates?

Or

- `(Get-Module Azure).Version`
Resources

TechNet PowerShell Script Center
http://go.microsoft.com/fwlink/p/?linkid=320211&clcid=0x409

Scripting with Windows PowerShell
http://go.microsoft.com/fwlink/p/?linkid=320210&clcid=0x409

Windows Azure Management Cmdlets

Running PowerShell Scripts
http://go.microsoft.com/fwlink/p/?linkid=320627&clcid=0x409

Hey, Scripting Guy!
http://blogs.technet.com/b/heyscriptingguy/
Create an Azure VM (Properly)

- Step 0: Standardize on naming conventions!
- Step 1: Create an Affinity Group
- Step 2: Create a Network
- Step 3: Create a Storage Account and Container
- Step 4: Create a Cloud Service
- Step 5: Create a VM
  - Optionally...Create an Availability Set
# Create new Affinity Group in "East US" Region.

New-AzureAffinityGroup -Name "RonBok-P-East-US" -Location "East US" -Description "Affinity group for Production in East US."

New-AzureAffinityGroup -Name "RonBok-T-East-US" -Location "East US" -Description "Affinity group for Test in East US."

New-AzureAffinityGroup -Name "RonBok-D-East-US" -Location "East US" -Description "Affinity group for Development in East US."

• NOTE: Naming conventions?
# Create Virtual Networks

```powershell
Set-AzureVNetConfig -ConfigurationPath "C:\PSScripts\<Subscription>RonBok-North-Central-US-NetworkConfig.xml"
```

- **NOTE:** This is - DESTRUCTIVE – of any existing networks, so make a backup FIRST. You can even export the existing, network, edit the .XML accordingly and then import it.
Create a Storage Account and Container

• # Create new Storage Account in "East US" Region.
• New-AzureStorageAccount -AffinityGroup "RonBok-P-East-US" -StorageAccountName "ronbokpeastus"
• New-AzureStorageAccount -AffinityGroup "RonBok-T-East-US" -StorageAccountName "ronbokteastus"
• New-AzureStorageAccount -AffinityGroup "RonBok-D-East-US" -StorageAccountName "ronbokdeastus"

• NOTE: Naming conventions?
• NOTE: \vhds
• NOTE: \backups?
Create a Cloud Service

• # Create new Cloud Service in "East US" Region.
  
  • New-AzureService -AffinityGroup "RonBok-P-East-US" -ServiceName "RonBok-P-East-US" -Description "Cloud Service for Production in East US."

  • New-AzureService -AffinityGroup "RonBok-T-East-US" -ServiceName "RonBok-T-East-US" -Description "Cloud Service for Test in East US."

  • New-AzureService -AffinityGroup "RonBok-D-East-US" -ServiceName "RonBok-D-East-US" -Description "Cloud Service for Development in East US."

• NOTE: You may want a Cloud Service per server pairing...or by project
Create a VM or two, or ...

- Get a list of VM images

```
# Get a list of existing Azure VM Images from the Gallery.
Get-AzureVMImage | Where {$_..ImageName -like "*Windows-Server-2012-R2*"} | Format-Table -AutoSize
```

- Get a single, specific VM image

```
(Get-AzureVMImage | where {$_..Label -like "Windows Server 2012 R2*"} | Sort PublishedDate -Descending)[0].ImageName
```
Select the proper storage account

#Set-AzureSubscription -SubscriptionName "Visual Studio Ultimate with MSDN" -CurrentStorageAccountName "ronbokwestus"

Set-AzureSubscription -SubscriptionName "Visual Studio Ultimate with MSDN" -CurrentStorageAccountName "ronbokeastus"

• NOTE: If you don’t do this...your VM will create it’s own default storage account with a nice long, random name. 😞
...finally we can create our VM

- # Create RonBok-DC01/DC02 Virtual Machine in East US!
- New-AzureQuickVM -ImageName $AzureVMImage -ServiceName "RonBok-DC-East-US" -Windows -AdminUsername "ronbok" -AvailabilitySetName "DomainControllers" -InstanceSize "Small" -Name "RonBok-DC01" -Password "Password.1" -SubnetNames "Subnet-1" -VNetName "RonBok-East" #-AffinityGroup "RonBok-East-US"
- New-AzureQuickVM -ImageName $AzureVMImage -ServiceName "RonBok-DC-East-US" -Windows -AdminUsername "ronbok" -AvailabilitySetName "DomainControllers" -InstanceSize "Small" -Name "RonBok-DC02" -Password "Password.1" -SubnetNames "Subnet-1" -VNetName "RonBok-East" #-AffinityGroup "RonBok-East-US"
- Done!