Executive Summary

Enterprises today are looking at various cloud offerings to understand how they can help augment on-premise solutions and provide IaaS or PaaS to business units. There are many different components when thinking about using a cloud connected solution; economics, security, governance, compliance, resiliency, scalability and performance to name but a few.

Pure Storage® Open Connect for Microsoft Azure is an open source PowerShell module that provides an easy way to establish connectivity with dedicated and private circuits utilizing Microsoft Azure ExpressRoute, Microsoft Azure and cloud exchange co-location sites such as Equinix Cloud Exchange data centers with Pure Storage FlashArrays. What makes this possible is Microsoft Azure ExpressRoute which provides private, reliable, high-speed connectivity between your enterprise and the cloud. With ExpressRoute a dedicated circuit is isolated using industry standard VLANs to allow private, secure access to resources deployed in Microsoft Azure Virtual Networks and also to provide connectivity to Microsoft Azure public services.

To compliment the ExpressRoute infrastructure, Pure Storage FlashArrays provide resilient, secure, high performance and scalability to meet the demands of enterprise data. The capabilities of a Pure Storage FlashArray are no different when using it within a cloud solution. Taking advantage of snapshots, data-at-rest encryption, data reduction, low-latency performance, resiliency and replication all work just as if the solution was on-premise. In addition to all of the core capabilities of the Pure Storage platform the simplicity of management still apply.

Goals and Objectives

This paper provides an overview of the Pure Storage All-Flash Cloud with Microsoft Azure solution and the deployment steps to setup and configure connectivity between Microsoft Azure and Equinix using ExpressRoute. The different components that will be explained are:

- **Microsoft Azure** – Subscription, virtual machines and virtual networking.
- **Pure Storage FlashArray** – Host iSCSI connectivity and volume management.
- **Open Connect for Microsoft Azure** – PowerShell module that provides simple custom configuration settings for connectivity.
- **Equinix Cloud Exchange Connectivity** – Networking and switch connectivity.
- **Microsoft Azure ExpressRoute** – Dedicated circuit, Border Gateway Protocol peering, virtual network and gateways.
- **Microsoft SQL Server** – Performance testing methods using a TPC-E like benchmark tool and Microsoft diskspd.

With the different components and processes to connect the various end points the main goal of this paper is to simplify those processes through the use of Windows PowerShell, Azure PowerShell and the Pure Storage PowerShell SDK as well as showcase all-flash performance in a hybrid cloud solution for enterprise workloads such as SQL Server.
Audience

This paper is written for database administrators, storage administrators, network administrators, cloud architects and DevOps teams that are interested in understanding the configuration of Azure ExpressRoute and the possibilities of leveraging the dynamic scalability of Azure cloud services combined with the Pure Storage FlashArray platform. Familiarity with Microsoft SQL Server, Pure Storage and Windows PowerShell is suggested.

Pure Storage Introduction

Who knew that moving to all-flash storage could help reduce the cost of IT? FlashArray//m makes server and workload investments more productive, while also lowering storage spend. With FlashArray//m, organizations can dramatically reduce the complexity of storage to make IT more agile and efficient, accelerating your journey to the cloud.

FlashArray//m’s performance can also make your business smarter by unleashing the power of real-time analytics, driving customer loyalty, and creating new, innovative customer experiences that simply weren’t possible with disk. All by transforming your storage with FlashArray//m.

FlashArray//m enables you to transform your data center, cloud, or entire business with an affordable all-flash array capable of consolidating and accelerating all your key applications.

Table 1. Pure Storage FlashArray//m Series. Full specifications can be reviewed here.
Solution Overview

The Pure Storage All-Flash Cloud for Microsoft Azure provides a cloud solution that allows enterprises to create a secure, scalable, and on-demand infrastructure using Microsoft Azure services with the security, performance and resiliency of the Pure Storage FlashArray platform. The solution uses Microsoft Azure ExpressRoute with the Equinix Cloud Exchange to connect Azure virtual machine services with co-located Pure Storage FlashArray equipment at an Equinix data center.

Figure 1 illustrates the different components that comprise the Pure Storage All-Flash Cloud with Microsoft Azure solution.

Figure 1. Pure Storage All-Flash Cloud with Microsoft Azure solution components.

There are a variety of use cases that can be deployed into the Pure Storage All-Flash Cloud with Microsoft Azure solution that include:

- Microsoft SQL Server®
- Development and test
- Oracle
- Disaster Recovery (DR)
- Data governance
As illustrated in Figure 1 both public and private connectivity can be used to access the equipment located in the enterprise private cage. This capability allows for use of Pure Storage FlashRecover Replication from on-premise Pure Storage FlashArray’s to Equinix data center co-located Pure Storage FlashArray’s. With Pure Storage FlashRecover Replication’s ability to preserve on-wire deduplication and compression this provides the ability to easily move data between on-premise applications and the cloud. In addition to being able to replicate data FlashRecover Replication also provides the underpinnings for Disaster Recovery options for various applications.

Figure 1 shows the Equinix Cloud Exchange in the center which provides the ability to connect Microsoft Azure compute resources with the enterprise private cage where Pure Storage FlashArray equipment is co-located. The connectivity between storage and compute resources is provided by Microsoft Azure ExpressRoute.

ExpressRoute provides dedicated high-bandwidth connectivity to the Microsoft Azure cloud with low-latency and secure communications. As a service provider to Microsoft Azure Equinix provides connectivity to meet any businesses requirements. Depending upon which connectivity model being used, Exchange Provider or Network Service Provider, the bandwidth options are: 10 MB/s, 50 MB/s, 100 MB/s, 200 MB/s, 500 MB/s, 1 GB/s, 2 GB/s, 5 GB/s and 10 GB/s.

Along with high-bandwidth Azure ExpressRoute provide connectivity using VLANs with provide private and secure access to the deployed resources in the Azure Virtual Network and public services.

There are several prerequisites to get started with the Pure Storage All-Flash Cloud with Microsoft Azure solution.

- **Pure Storage FlashArray/m** – Procure a FlashArray/m to be placed in an IBX® data center. Review more details here. The connectivity protocol for Azure ExpressRoute is iSCSI, so the FlashArray/m would need to be configured with iSCSI cards.

- **IBX® Data Center and Colocation** – Review colocation options for Equinix’s International Business Exchange here.

- **Networking Equipment** – Procure the proper networking gear to connect to the FlashArray/m for access to Azure ExpressRoute.

- **Deployment Planning** – Azure Regions provides a way for global presence that allows customers to provide services close to end-users based on geography, region or country. Review full details about Azure Regions here. In addition to understanding the various regions and what services are offered it is important to understand Azure Service Level Agreements (SLAs). Full details on the

---

1 Network Service Provider.

2 Exchange Provider.
various SLAs for services can be reviewed [here](#). For this paper the two SLAs of most concern are **ExpressRoute** and **Virtual Machines**.

There are several different access methods that can be used when configuring Azure ExpressRoute. This paper describes the connectivity of using a network service provider, Equinix. The other method that can be configured are direct connect to Azure from an existing WAN network (Eg. MPLS, Multi-Protocol Label Switching VPN or IPVN).

The remainder of this paper defines the different components, workflows, tools and configuration details which will help build an understanding of the scope. This knowledge can then be used to make an informed decision to deploy into Azure using either a direct connect or network service provider colocation model.

**Solution Management**

The Pure Storage® All-Flash Cloud with Microsoft Azure has several different ways to manage and develop for extensive scripts using the Open Connect for Microsoft Azure. Each mechanism below describes from the least complex to the most extensible.

- **Microsoft Azure Portal** – Figure 2 shows the new Microsoft Azure portal called Resource Manager. The portal can be used to create and manage all Azure services. Resource Manager provides a GUI to perform all management tasks for the different services. Resource Manager is the preferred version to manage Azure services over the original Classic portal. Both versions are available from Microsoft.
Windows PowerShell – Figure 3 shows the Windows PowerShell management experience through the use of the cmdlets which are part of the Azure PowerShell Module. Figure 3 shows adding an Azure account to the session using `Add-AzureAccount` and then query for the configured services under the account.
Figure 3. Managing Microsoft Azure services from Windows PowerShell.

- **Windows PowerShell Integrated Scripting Environment (ISE)** – Similar to the Windows PowerShell shell experience the ISE provides the same mechanism to issue commands shown in Figure 4. Additionally the ISE can be used for full script development, debugging and use of the Command Add-on for help with individual cmdlets. The ISE is a comprehensive editing and execution environment for PowerShell scripting.
• **Microsoft Visual Studio** – The ultimate management and development experience is using Visual Studio with PowerShell Tools for Visual Studio. Using the Integrated Development Environment (IDE) allows for developing PowerShell scripts, adding C# projects that use PowerShell along with debugging and viewing resources in Microsoft Azure using the Cloud Explorer.
Based on the different management and development options described above there are several software prerequisites to get started with the Pure Storage® All-Flash Cloud for Microsoft Azure solution.

- **Pure Storage® Open Connect for Microsoft Azure** – This is a PowerShell script module that provides cmdlets for custom configuration of Azure ExpressRoute, FlashArray subnets, FlashArray hosts, iSCSI and volumes. The module is a wrapper for Azure PowerShell and ExpressRoute cmdlets. The intended use of Open Connect for Microsoft Azure is for configuration of connectivity between Microsoft Azure, Equinix Cloud Exchange and Pure Storage FlashArray.

- **Microsoft Azure PowerShell Module** – Azure PowerShell is a module that provides cmdlets to create, test, deploy, and manage services delivered through the Azure platform. In most cases, the cmdlets can be used for the same tasks as the Azure Management Portal, such as creating and configuring cloud services, virtual machines, virtual networks, and web apps. The Azure PowerShell module also includes cmdlets for configuring Azure ExpressRoute. Download the Azure PowerShell module [here](#).

- **Windows Management Framework (WMF)** – The minimum version of WMF to use is 4.0 which can be downloaded [here](#). There have been many enhancements with WMF 5.0 and it is highly recommended to use. WMF 5.0 is still in a Production Preview for Windows Server but is available by default on Windows 10. Download WMF 5.0 [here](#).

- **Windows PowerShell Integrated Scripting Environment (ISE)** – The PowerShell ISE comes as part of the Windows Server feature for Windows PowerShell. The ISE can be added using...
Windows PowerShell with `Add-WindowsFeature -Name PowerShell-ISE` or by using the Server Manager > Add Roles and Features.

![Remote Access module for Windows PowerShell](image)

**Figure 6.** PowerShell-ISE feature that can be added using `Add-WindowsFeature` cmdlet.

The Pure Storage® All-Flash Cloud for Microsoft Azure can be configured, managed and customized from a Windows 8.1, Windows 10 or Windows Server 2012 R2 or 2016 (Tech Preview 4) machine. Once the above software prerequisites are installed then Open Connect for Microsoft Azure can be used.

It is highly recommended to use Visual Studio 2013 or 2015 to use Open Connect for Microsoft Azure. Visual Studio has integration with GitHub, Cloud Explorer and PowerShell Tools for 2012, 2013 and 2015 for project creation and debugging. Open Connect for Microsoft Azure is an open source PowerShell module that can be further customized by developers or scripters.

**Deployment Guidance**

The Pure Storage Open Connect for Microsoft Azure simplifies the configuration between an Exchange service provider or direct connection. The configuration details provided in this paper use the Equinix International Business Exchange (IBX) as the service provider.

Prior to following the below deployment steps it is assumed that a Pure Storage FlashArray has been deployed into an exchange service provider facility and configured with management IP Addresses. In addition, all networking equipment is assumed to be configured. Testing access to the management ports is critical before proceeding to deployment.

The following steps describe the how to manually establish a connection between the Microsoft Azure cloud and an Exchange Server provider (Eg. Equinix) via Azure ExpressRoute.

1. Sign-up for a Microsoft Azure Subscription.
2. Download and install the following software packages:
   a. Microsoft Azure PowerShell Module
3. Configure Pure Storage iSCSI interface subnets.
4. Create a Microsoft Azure ExpressRoute circuit.
5. Connect the Azure ExpressRoute circuit with the Exchange Service Provider (Eg. Equinix).
6. Create and configure the Azure Virtual Network.


9. Connect the Azure Virtual Network to the Azure ExpressRoute circuit.

10. Create test virtual machine(s) into the Azure Virtual Network.

With the Open Connect for Microsoft Azure solution the above steps are simplified to the following:

1. Sign-up for a Microsoft Azure Subscription.

2. Download and install the following software packages:
   a. Microsoft Azure PowerShell Module (MSI)
   b. Pure Storage PowerShell SDK 1.0.15.0 (MSI)
   c. Pure Storage PowerShell Toolkit 3.0 (MSI)


4. Run New-PsOpenConenctConfiguration.

5. Run New-PsOpenConnectExpressRoute.

6. Run New-OpenConnectTestVM.

As previously noted to begin deployment of the Open Connect for Microsoft Azure solution a Windows 8.1, Windows 10 or Windows Server 2012 R2 client is required to install the software components noted in Step 2.

**Setup Deployment Environment**

The first task is to setup the individual software components. Download the different packages to a preferred folder and then install in the following order:

1. Microsoft Azure PowerShell Module

2. Pure Storage PowerShell SDK

3. Pure Storage PowerShell Toolkit

4. Pure Storage OpenConnect for Microsoft Azure
It is recommended to create a folder the following folder C:\MyAzure to download the above files for later use with the test virtual machine.

Once each of the software components has been installed using Windows Control Panel > Programs and Features each of the installations can be viewed.

Configure Pure Storage FlashArray Subnets

Open a browser and navigate to the virtual management port (vir0) that was use as part of the Pure Storage FlashArray setup. Once connected to the FlashArray navigate to the **System** tab > **Networking** as shown in Figure 9.
Figure 9. Pure Storage FlashArray Networking.

From the Networking section click the (gear icon) to **Create Subnet**, this will open up the **Edit Subnet** dialog to configure the new subnet. The example provided in Figure 10 uses a 192.168.255.0/24 address.

Figure 10. Creating a new subnet.
The networking setup is now complete with the FlashArray with management and virtual ports and subnets.

Before proceeding ensure that there is an active Microsoft Azure subscription. No details of the subscription are necessary except the username and password.

Customize Configuration

This is the most important configuration step for setting up the Pure Storage All-Flash Cloud for Microsoft Azure solution. It is very important to obtain the different details required.

Open up a text editor and navigate to C:\Program Files\Pure Storage\PowerShell\Modules\PureStorageOpenConnectMicrosoftAzure\. In this folder there are two XML configuration files that are used by the Open Connect for Microsoft Azure PowerShell Module.

5. MicrosoftAzureCustomConfiguration.xml – Contains all of the custom configuration properties to create the Azure ExpressRoute components and test virtual machines.

6. MicrosoftAzureVNetConfiguration.xml – Contains all of the detailed information for networking. This file should not be touched and is automatically generated from the CustomConfiguration.xml upon loading using the New-PsOpenConnectConfiguration cmdlet.

Modification of the XML schema provided in PsOpenConnectMicrosoftAzureCustomConfiguration.xml or PsOpenConnectMicrosoftAzureVNetConfiguration.xml should not be done.

Appendix A illustrates an example of the custom configuration XML file. Below are the different nodes and their use:

1. ProviderConfiguration – DnsServers that are to be used for the connectivity. This is optional.

2. FlashArrayConfiguration – Management ports and Subnets to be used for external and Azure connectivity.

3. AzureConfiguration – VirtualNetworkName (VNet), GatewayType and LocalNetworkSiteName that will be used to create a virtual network between Azure Cloud and the Equinix Cloud Exchange to access the Pure Storage FlashArray.
4. **AzureVirtualMachineConfiguration** – NetworkInterfaces, InstanceSize, VMTemplate, Admin, VMInstanceName and ServiceName. All of these settings are used to create the test virtual machine using the `New-PsOpenConnectTestVM` cmdlet.

5. **ExpressRouteConfiguration** – CircuitName, Bandwidth, Location, ProviderName, Sku and Border Gateway Protocol (BGP) peers. All of these details are used to create the dedicated circuit connection between Azure Cloud and Cloud Exchange known as ExpressRoute.

The following section contain Windows PowerShell cmdlet examples and in some cases the ‘tick’ symbol (`) is used for readability purposes.

**New-PsOpenConnectConfiguration**

To get started configuring the Azure solution start a Windows PowerShell session to begin the next steps of configuration. To view the different Pure Storage PowerShell modules, enter the following:

```
Get-Module -ListAvailable *PureStorage*
```

![Get-Module -ListAvailable *PureStorage*](figure11.png)

Next to see the exported module members, enter the following:

```
Get-Command -Module PureStorageOpenConnectMicrosoftAzure
```

Next to see the exported module members, enter the following:

```
Get-Command -Module PureStorageOpenConnectMicrosoftAzure
```
At first this might not seem like a very powerful module when in fact these four simple cmdlets wrap Azure, ExpressRoute, iSCSI and creating test virtual machines.

Using the previously customized configuration file start with the following:

```
New-PsOpenConnectConfiguration -ConfigurationFile 'C:\MyAzure\MicrosoftAzureCustomConfiguration.xml'
```

It is recommended to make a copy of the `MicrosoftAzureCustomConfiguration.xml` file and keep the one located in the Pure Storage PowerShell directory as a master copy. As illustrated above a folder was created C:\MyAzure and the xml file copied to this location for use.

After executing the `New-PsOpenConnectConfiguration` cmdlet the Microsoft Azure login will appear as shown in Figure 13.
The **New-PsOpenConnectConfiguration** cmdlet is a wrapper cmdlet that encapsulates the following tasks:

1. Retrieving custom configuration details  
2. Creating a new dedicated circuit  
3. Creating public and private Border Gateway Protocol (BGP) peering

**New-PsOpenConnectExpressRoute**

Now that the ExpressRoute connection is created and connected from the Microsoft Azure cloud to the Pure Storage FlashArray using the Exchange Cloud Exchange it is time to test the connectivity with a virtual machine.

The **New-PsOpenConnectExpressRoute** cmdlet is a wrapper cmdlet that encapsulates the following tasks:

1. Retrieve custom test virtual machine configuration settings.  
2. Creates new Azure Virtual Network Gateway.  
3. Updates the Virtual Network Configuration XML file with the Virtual Network Gateway details.  
4. Creates the Azure Dedicated Circuit.

Example:

```
New-PsOpenConnectExpressRoute -ConfigurationFile 'C:\MyAzure\MicrosoftAzureCustomConfiguration.xml'
```

**New-PsOpenConnectTestVM**

Now that the ExpressRoute connection is created and connected from the Microsoft Azure cloud to the Pure Storage FlashArray using the Exchange Cloud Exchange it is time to test the connectivity with a virtual machine.

The **New-PsOpenConnectTestVM** cmdlet is a wrapper cmdlet that encapsulates the following tasks:

1. Retrieve custom test virtual machine configuration settings.  
2. Select the most recent image of Windows Server 2012 R2 Datacenter (Eg. January 2016).  
3. Creating a new Azure Storage Account to host the VHD.  
4. Configuring the virtual machines size and name.
5. Set the different provisioning options for the virtual machine (Eg. Subnet, Admin name, Admin password).


7. Create the new Azure virtual machine and retrieve the Remote Desktop (RDP) file and automatically launch the VM.

One this cmdlet has completed the following Remote Desktop Connection dialogs will be displayed as shown in Figure 14 and Figure 15. Choose to **Connect** and **Yes** to verify the identity of the remote computer. This is only a test virtual machine but select the **Don't ask me again for connections to this computer** can be checked.

Example:

```powershell
New-PsOpenConnect Test VM
  -ConfigurationFile 'C:\MyAzure\MicrosoftAzureCustomConfiguration.xml'
  -RDPPath 'C:\MyAzure\AccelerateVM5.rdp' -Launch $True
```

![Remote Desktop Connection](image1.png)

Figure 14. Remote Desktop Connection to AccelerateVM5.cloudapp.net.
Figure 15. Remote Desktop Connection identity verification for AccelerateVM5.cloudapp.net.

Figure 16 shows the downloaded AccelerateVM5.rdp file to the C:\MyAzure folder.

After logging into the test virtual machine (Eg. AccelerateVM5) native to the C:\MyAzure directory select the Pure Storage PowerShell SDK, Toolkit and Microsoft Azure modules and cut-n-paste into the test virtual machine.

New-PsOpenConnectFlashArrayiSCSISetup

After successfully creating a test virtual machine the final tasks are to setup the iSCSI connectivity. In order to perform this final setup a connection to the newly created test virtual machine that the New-PsOpenConnect FlashArray iSCSI Setup will perform are as follows:
1. Create & configure a new host on the Pure Storage FlashArray. This includes adding the iSCSI IQNs.

2. Create & connect a test volume to the newly created host from Step 1.

3. Add & configure the Windows Server feature for Multipath-IO.

4. Configure & start the Microsoft iSCSI Initiator service.

5. Establish connection to the host & volume on the Pure Storage FlashArray.

6. Online and format the new test volume.

Example:

```
New-PsOpenConnectFlashArrayIscsiSetup -ConfigurationFile 'C:\MyAzure\MicrosoftAzureCustomConfiguration.xml' -VolumeName 'TestVol-VM5' -VolumeSize 500GB -HostName 'AccelerateVM5'
```

Figure 18 illustrates once the `New-PsOpenConnectFlashArrayIscsiSetup` completes that the volumes shown from the FlashArray are connected to the test virtual machine, AccelerateVM5.
Figure 18. Windows Disk Management showing volumes from the FlashArray connected via iSCSI to test vm.

Figure 19 shows the newly created AccelerateVM5 from the Pure Storage FlashArray web management interfaces shows that the host is full redundant across the controllers with a 1 GB/s speed.
Verification & Testing

The next section will discuss the various tests that were performed to verify connectivity and performance.

Verification & Testing

The following section provides the verification and testing procedures that were used to validate the Open Connect for Microsoft Azure solution. The test results discussed below are based on a 1 GB/s connection between the Equinix International Business Exchange (IDX) and Cloud Exchange that is hosting a Pure Storage FlashArray that is accessible from the Microsoft Azure Cloud.

The different tests are defined as such:

1. **Disk Speed (diskspd)** – The replacement tool for SQLIO is a feature-rich and versatile storage testing tool, diskspd combines robust and granular IO workload definition with flexible runtime and output options, creating an ideal tool for synthetic storage subsystem testing and validation.

2. **Proprietary Benchmark Tool** – A TPC-E like benchmark tool was used to perform a real-world Online Transaction Processing application (OLTP).
3. **Dev/Test Scenario** – This is one of the primary workloads Pure Storage sees as an opportunity for businesses. The ability to leverage the elastic compute of Azure in combination with Pure Storage FlashRecover snapshots provides an ability to rapidly clone database instances.

### Disk Speed (diskspd)

Diskspd is the replacement tool for SQLIO and can be used to basic verification testing of IO paths to the Pure Storage FlashArray. For full details on how to use diskspd see the [TechNet Gallery](#).

**Example:**

```
PS C:\diskspd\amd64fre> .\diskspd.exe -c4096G -d3600 -r -w0 -t8a -o8 -h -L G:\AzureTest.dat
PS C:\diskspd\amd64fre> .\diskspd.exe -c4096G -d3600 -r -w10 -t8a -o8 -h -L G:\AzureTest.dat
PS C:\diskspd\amd64fre> .\diskspd.exe -c4096G -d3600 -r -w30 -t8a -o8 -h -L G:\AzureTest.dat
```

100% Read with Average 64KB IO Size

![Figure 20. Diskspd 100% Read test.](image)
10% Write / 90% Read with Average 64KB IO Size

Figure 21. Diskspd 10% Write / 90% Read test.

30% Write / 70% Read with Average 64KB IO Size

Figure 22. Diskspd 30% Write / 70% Read test.
Diskspd was used for basic verification testing of connectivity between the Microsoft Azure cloud and the Equinix International Business Exchange data center. Pure Storage does not recommend using synthetic testing tools for workload analysis. Pure Storage recommends deploying a real-world workload for testing analysis.

Microsoft SQL Server (OLTP)

The OLTP tests performed were done using Microsoft SQL Server 2014. The benchmark software is a proprietary tool provided by Microsoft to partners. The database was generated based on 10,000 users and the individual OLTP test was run using 100 users running simultaneous transactions.

Data Generation Results

Figure 23. Loading data for 10,000 users.
Index Creation

Figure 24. Creating indexes.

Database Load Results

Figure 25. Loading database with generated data.
Database Performance Results

Figure X illustrates the loading of the 10,000 user database from an Azure virtual machine across the Azure Cloud Cloud Exchange to a colocated Pure Storage FlashArray.

Microsoft SQL Server Dev/Test Scenario

Pure Storage provides FlashRecover

Scenario:

1. A copy of the Salsa POS (Prod) database is backed up and restored on a Azure VM that is connected to a Pure Storage FlashArray. The restore is called Salsa POS (Prod-Copy).

2. The development team need an instance of this database for feature enhancements coding.

3. Using the Pure Storage PowerShell SDK a volume snapshot is taken of the Salsa POS (Prod-Copy), and a new volumes is created for development.

4. Using a separate Azure virtual machine the newly create Salsa P is connected so development can begin work.

The above scenario and workflow can be repeated multiple times for different individual developers or teams of developers.

Step 1
Create FlashRecover Snapshot of the volume containing the SQL Server user data file(s).

1. Query FlashArray for volumes connected to AccelerateVM1.

   ```
   $Creds = Get-Credential
   $FlashArray = New-PfaArray -EndPoint 64.191.192.60 -Credentials $Creds -IgnoreCertificateError
   Get-PfaVolumes -Array $FlashArray | Select-Object name | Format-Table -AutoSize
   Results:
   name
   ----
   AccelerateVM1-SQLData
   AccelerateVM1-SQLSys
   AccelerateVM1-SQLTemp
   AccelerateVM2-SQLData
   AccelerateVM2-SQLSys
   AccelerateVM2-SQLTemp
   diskspd
   FlatFiles
   Test-Volume1
   Test-Volume2
   TestVol-VM2
   ```

2. Create FlashRecover Snapshot of AccelerateVM1-SQLData.

   ```
   New-PfaVolumeSnapshots -Array $FlashArray -Sources 'AccelerateVM1-SQLData' -Suffix 'Azure'
   Results:
   source : AccelerateVM1-SQLData
   serial  : 619826A62A08489F0001101C
   created : 2016-03-07T23:00:10Z
   name    : AccelerateVM1-SQLData.Azure
   size    : 1099511627776
   ```

3. Create new volume from AccelerateVM1-SQLData snapshot.

   ```
   New-PfaVolume -Array $FlashArray -Source 'AccelerateVM1-SQLData.Azure' -VolumeName 'AccelerateVM1-DataDevCopy'
   Get-PfaVolumes -Array $FlashArray | Select-Object name | Format-Table -AutoSize
   Results:
   source : AccelerateVM1-SQLData
   serial  : 619826A62A08489F0001101D
   created : 2016-03-07T23:02:46Z
   name    : AccelerateVM1-DataDevCopy
   size    : 1099511627776
   name
   ----
   AccelerateVM1-DataDevCopy
   AccelerateVM1-SQLData
   AccelerateVM1-SQLSys
   AccelerateVM1-SQLTemp
   AccelerateVM2-SQLData
   AccelerateVM2-SQLSys
   AccelerateVM2-SQLTemp
disks
   FlatFiles
   Test-Volume1
   Test-Volume2
   ```
4. Connect new volume, AccelerateVM1-DataDevCopy, to AccelerateVM2 host.

```
Get-PfaHostVolumeConnections -Array $FlashArray -Name 'AccelerateVM2'
New-PfaHostVolumeConnection -Array $FlashArray -VolumeName 'AccelerateVM1-DataDevCopy' -HostName 'AccelerateVM2'
Get-PfaHostVolumeConnections -Array $FlashArray -Name 'AccelerateVM2'
```

Results:

<table>
<thead>
<tr>
<th>vol</th>
<th>name</th>
<th>lun</th>
<th>hgroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Vol - VM2</td>
<td>AccelerateVM2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>diskspd</td>
<td>AccelerateVM2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>AccelerateVM2-SQLData</td>
<td>AccelerateVM2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>AccelerateVM2-SQLSys</td>
<td>AccelerateVM2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>AccelerateVM2-SQLTemp</td>
<td>AccelerateVM2</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

```
Get-PfaHostVolumeConnections -Array $FlashArray -Name 'AccelerateVM2'
```

Results:

<table>
<thead>
<tr>
<th>vol</th>
<th>name</th>
<th>lun</th>
<th>hgroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Vol - VM2</td>
<td>AccelerateVM2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>diskspd</td>
<td>AccelerateVM2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>AccelerateVM2-SQLData</td>
<td>AccelerateVM2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>AccelerateVM2-SQLSys</td>
<td>AccelerateVM2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>AccelerateVM2-SQLTemp</td>
<td>AccelerateVM2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>AccelerateVM1-DataDevCopy</td>
<td>AccelerateVM2</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

5. Rescan AccelerateVM2 for newly connected volume, AccelerateVM1-DataDevCopy.

```
winrm quickconfig /q -force
Register-PfaHostVolumes -ComputerName localhost
```

For testing purposes `winrm quickconfig` was used to set the winrm service to auto-start, create a listener for all IPs requesting access and enabling firewall excepts for WS-Management traffic. This is not recommended in a production environment.
Attach database using SQL Server Management Studio.

1. Open SQL Server Management Studio
2. Select Databases, right-click, select Attach...
3. Navigate to volume and select the Salsa POS (Nightly-Dev) file. (eg. J:\MSSQLSERVER\Salsa POS (Nightly-Dev.mdf)
Summary

Pure Storage All-Flash Cloud for Microsoft Azure provides a combination of benefits provided by private, industry-leading all-flash storage coupled with highly scalable, on-demand public cloud compute resources from Microsoft Azure. The solution leverages the high-availability of Microsoft Azure virtual machine services, proprietary, low-latency connectivity with Equinix and the performance, data reduction, resiliency and simplicity of management with Pure Storage FlashArray.

The Pure Storage® Open Connect for Microsoft Azure PowerShell module automates connectivity, set up and management of Pure Storage to Azure with Azure ExpressRoute and Equinix Cloud Exchange. This solution is ideal for easily creating a hybrid cloud solution for workloads that require on-demand compute scaling, secure connectivity, performance and data security. The Pure Storage FlashArray couples low-latency performance, platform resiliency and data-at-rest encryption with scalable cloud compute resources.
References

The following are references used in the development of the Pure1 Connect for Microsoft Azure solution.

  - GitHub -- [https://github.com/Microsoft/diskspd](https://github.com/Microsoft/diskspd)
- Microsoft Azure Subscriptions -- [https://account.windowsazure.com/subscriptions/](https://account.windowsazure.com/subscriptions/)


• Diskspd Download -- https://gallery.technet.microsoft.com/DiskSpd-a-robust-storage-6cd2f223

• Microsoft Azure Service Status -- https://azure.microsoft.com/en-us/status/

Appendix A – Custom Configuration XML File

```xml
<?xml version="1.0" encoding="utf-8"?>
<OpenConnectMicrosoftAzure xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ProviderConfiguration>
    <Dns>
      <DnsServers>
        <DnsServer />
      </DnsServers>
    </Dns>
  </ProviderConfiguration>

  <FlashArrayConfiguration>
    <Management>
      <ct0.eth0>x.x.x.x</ct0.eth0>
      <ct0.eth1>x.x.x.x</ct0.eth1>
      <vir0>x.x.x.x</vir0>
    </Management>
    <Subnet>
      <VLAN1>
        <IpAddress></IpAddress>
      </VLAN1>
      <VLAN2>
        <IpAddress></IpAddress>
      </VLAN2>
    </Subnet>
  </FlashArrayConfiguration>

  <AzureConfiguration>
    <VirtualNetworkName>Azure</VirtualNetworkName>
    <GatewayType>DyanmicRouting</GatewayType>
    <LocalNetworkSiteName>EQX</LocalNetworkSiteName>
  </AzureConfiguration>

  <AzureVirtualMachineConfiguration>
    <NetworkInterfaces>
      <Interface1></Interface1>
      <Interface2></Interface2>
    </NetworkInterfaces>
    <InstanceSize>ExtraLarge</InstanceSize> <!--Required for multiple NIC support-->
    <VMTemplate>Windows Server 2012 R2 Datacenter, January 2016</VMTemplate>
    <Admin>Barkz</Admin>
    <AdminPassword>Test1234!</AdminPassword>
    <VMInstanceName>AccelerateVM2</VMInstanceName>
    <ServiceName>AccelerateService2</ServiceName>
  </AzureVirtualMachineConfiguration>

  <ExpressRouteConfiguration>
    <CircuitName>PureStorageExpressRouteCircuit</CircuitName>
    <Bandwidth>1000</Bandwidth>
    <Location>Silicon Valley</Location>
    <ProviderName>Equinix</ProviderName>
  </ExpressRouteConfiguration>
</OpenConnectMicrosoftAzure>
```
<Primary>x.x.x.x/30</Primary>
<Secondary>x.x.x.x/30</Secondary>
<VLAN>3021</VLAN>
</Public>
</BorderGatewayProtocolPeering>
</ExpressRouteConfiguration>
</OpenConnectMicrosoftAzure>
Glossary of Terms

The following section defines terms used in the Pure1 Connect for Microsoft Azure solution.

**Microsoft Azure ExpressRoute**
Microsoft Azure ExpressRoute extends on-premises networks into the Microsoft cloud over a dedicated private connection facilitated by a connectivity provider. Connectivity can be from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a co-location facility. ExpressRoute connections do not go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet.

**Microsoft Azure Virtual Network**
An Azure virtual network (VNet) is a representation of a custom defined network in the cloud. It is possible to control Azure network settings and define DHCP address blocks, DNS settings, security policies, and routing.

**Microsoft Azure Virtual Network Gateway**
A virtual network gateway for Microsoft Azure is a virtual router used to connect a virtual network to other networks. For this solution the Azure virtual network gateway connects Microsoft Azure ExpressRoute to a Pure Storage FlashArray that has been deployed at an Equinix data center.

**Microsoft Azure Border Gateway Protocol**
BGP (Border Gateway Protocol) is a layer 3 network routing protocol for exchanging routing information between gateway hosts. For the Pure1 Connect for Microsoft Azure solution BPG is used between Microsoft Azure and the Equinix data center.

**Microsoft Azure Virtual Machines**
Azure Virtual Machines is one of several types of on-demand, scalable computing resources that Azure offers. An Azure virtual machine provides the flexibility of virtualization without having to buy and maintain the physical hardware that runs the virtual machine. The virtual machine(s) still need to be maintained -- configuring, patching, and maintaining the operating system and any other software that runs on the virtual machine.

**Microsoft Azure Region**
Azure services are hosted in data centers that are managed globally by Microsoft in different geographic locations. For a full list of Azure services by region see [https://azure.microsoft.com/en-us/regions/#services](https://azure.microsoft.com/en-us/regions/#services).
About the Author

As a Solutions Architect, Barkz is creating the foundation knowledgebase for implementing Microsoft server technologies on Pure Storage. Those core items include best practices, reference architectures, management tasks, automation scripts and examples for application extensibility. With more than 20 years of experience with Microsoft solutions, Barkz has been part of all aspects from architecture, user design, development, test, release and administration. Barkz has experience in Windows PowerShell, Windows Server, Microsoft SQL Server (Admin & Development), Microsoft SharePoint Server (Admin & Development), Microsoft Hyper-V, Visual Studio, C# and REST API.

Barkz blog: http://blog.purestorage.com/author/barkz/ or http://purepowershellquy.com

Twitter: @purepowershell

GitHub
http://github.com/barkz

Demonstration Videos
YouTube

Pure1 Community
Programming Interfaces Community Page

Pure Storage PowerShell Toolkit
https://github.com/PowerShell-Toolkit