Building a resilient IaaS architecture
Hands-on lab unguided
June 2018



**Contents**

- Building a resilient IaaS architecture hands-on lab unguided
  - Abstract and learning objectives
  - Overview
  - Solution architecture
  - Requirements
    - Help references
  - Exercise 1: Prepare connectivity between regions
    - Task 1: Create a VNET in the second region

        - Tasks to complete
        - Exit criteria
    - Exercise 2: Build the DCs in for resiliency
        - Task 1: Create Resilient Active Directory Deployment
            - Tasks to complete
            - Exit criteria
        - Task 2: Create the Active Directory deployment in the second region
            - Tasks to complete
            - Exit criteria
        - Task 3: Add data disks to Active Directory domain controllers (both regions)
            - Tasks to complete
            - Exit criteria
        - Task 4: Format data disks on DCs and configure DNS settings across connection
            - Tasks to complete
            - Exit criteria
        - Task 5: Promote DCs as additional domain controllers
            - Tasks to complete
            - Exit criteria
        - Summary
    - Exercise 3: Build web tier and SQL for resiliency
        - Task 1: Deploy SQL Always-On Cluster
            - Tasks to complete
            - Exit criteria
        - Task 2: Convert the SQL deployment to Managed Disks
            - Tasks to complete
            - Exit criteria
        - Task 3: Build a scalable and resilient web tier
            - Tasks to complete
            - Exit criteria
        - Summary
    - Exercise 4: Configure SQL Server Managed Backup
        - Task 1: Create an Azure Storage Account
            - Tasks to complete
            - Exit criteria
        - Task 2: Configure managed backup in SQL Server
            - Tasks to complete
            - Exit criteria
    - Exercise 5: Validate resiliency
        - Task 1: Validate resiliency for the CloudShop application
            - Tasks to complete
            - Exit criteria
        - Task 2: Validate SQL Always On
            - Tasks to complete
            - Exit criteria
        - Task 3: Validate backups are taken

        - Tasks to complete
        - Exit criteria
    - After the hands-on lab
        - Task 1: Delete the resource groups created


# Building a resilient IaaS architecture hands-on lab unguided

## Abstract and learning objectives

In this hands-on lab, you will deploy a pre-configured IaaS environment and then redesign and update it to account for resiliency and in general high availability. Throughout the hands-on lab you will use various configuration options and services to help build a resilient architecture.

At the end of this workshop, you will be better able to design and use the following services:

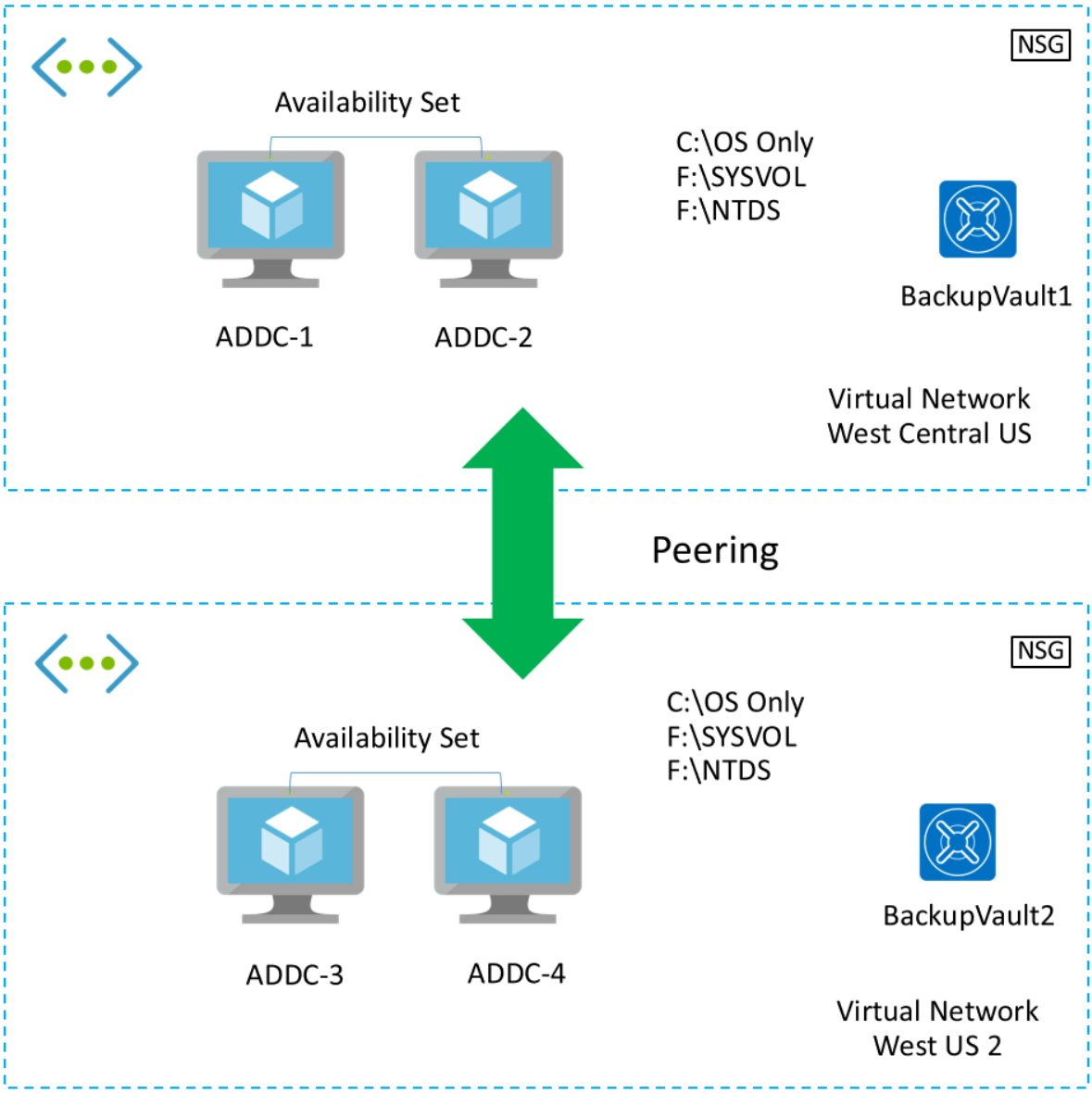
- The use of availability sets
- The use of Managed Disks
- Design principles when provisioning storage to VMs
- Effective employment of Azure Backup to provide point-in-time recovery
- SQL Server Always On Availability Groups

## Overview

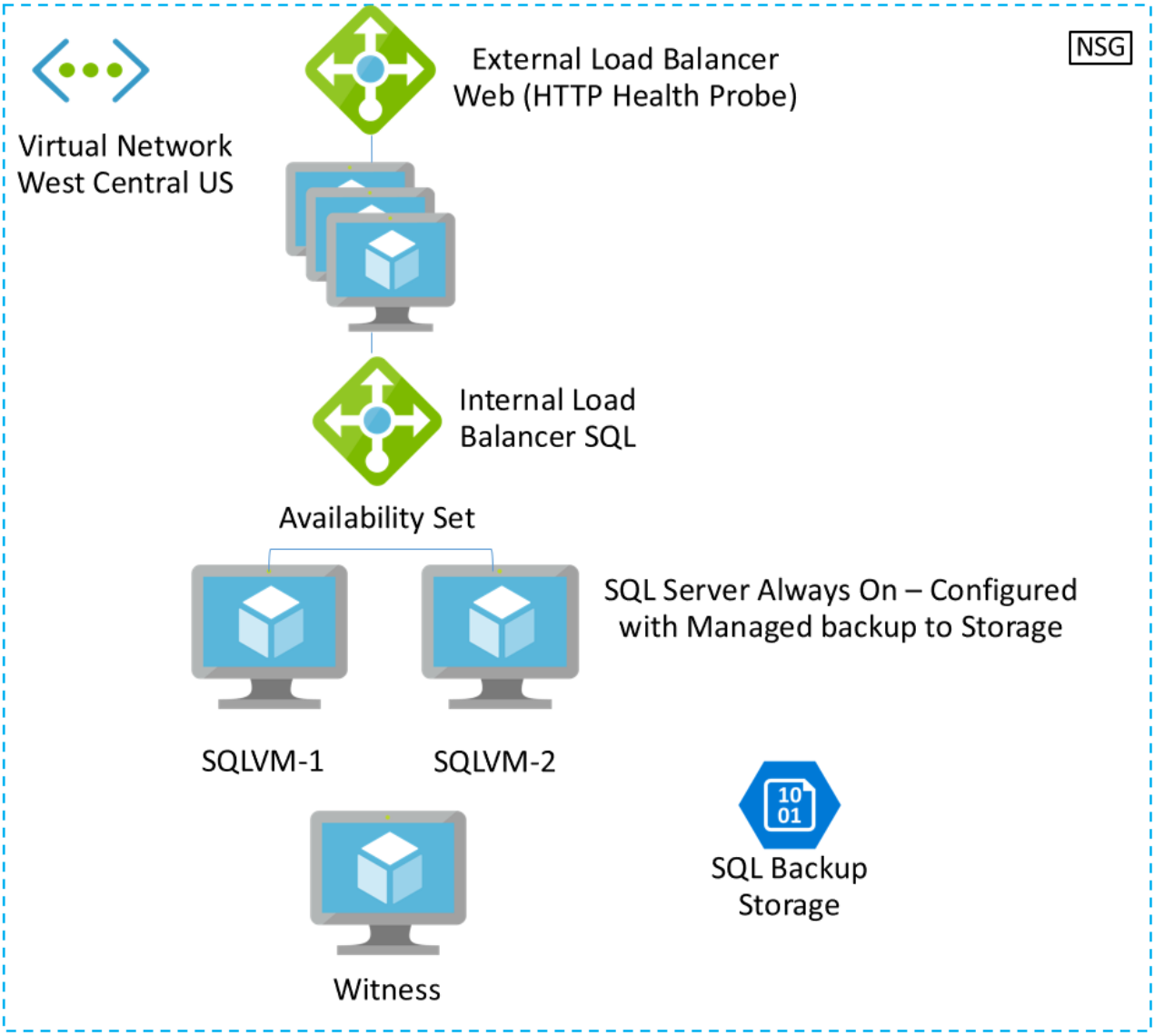
Contoso has asked you to deploy their infrastructure in a resilient manner to insure their infrastructure will be available for their users and gain an SLA from Microsoft.

## Solution architecture

Highly resilient deployment of Active Directory Domain Controllers in Azure.



Deployment of a web app using scale sets, and a highly available SQL Always On deployment.




## Requirements

1. Microsoft Azure Subscription
2. Virtual Machine Built during this hands-on lab or local machine with the following:
    - Visual Studio 2017 Community or Enterprise Edition
    - Latest Azure PowerShell Cmdlets
    - https://azure.microsoft.com/en-us/downloads/
    - Ensure you reboot after installing the SDK or Azure PowerShell will not work correctly

### Help references

| Description | Links |
| --- | --- |

| | |
|---|---|
| Authoring ARM Templates | https://azure.microsoft.com/en-us/documentation/articles/resource-group-authoring-templates/ |
| Virtual Machine Scale Set Samples | https://github.com/gbowerman/azure-myriad |
| Azure Quick Start Templates | https://github.com/Azure/azure-quickstart-templates |
| Network Security Groups | https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-nsg/ |
| Managed Disks | https://azure.microsoft.com/en-us/services/managed-disks |
| Always-On Availability Groups | https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/overview-of-always-on-availability-groups-sql-server?view=sql-server-2017 |
| SQL Server Managed Backup to Azure | https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/sql-server-managed-backup-to-microsoft-azure?view=sql-server-2017 |
| Virtual Network Peering | https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-peering-overview |
| Azure Backup | https://azure.microsoft.com/en-us/services/backup/ |

# Exercise 1: Prepare connectivity between regions

Duration: 30 minutes

Contoso is planning to deploy infrastructure in multiple regions in Azure to provide infrastructure closer to their employees in each region as well as the ability to provide additional resiliency in the future for certain workloads. In this exercise, you will configure connectivity between the two regions.

## Task 1: Create a VNET in the second region

**Tasks to complete**

- Create a new Virtual network in the West US 2 region named ContosoVNET2 that mirrors ContosoVNET except with a different address space

**Exit criteria**

- There should be a new virtual network with two subnets: Apps and Data in the West US 2 region

# Exercise 2: Build the DCs in for resiliency

Duration: 30 minutes

In this exercise, you will deploy Windows Server Active Directory configured for resiliency using Azure Managed Disks and Availability Sets in the primary region. You will then deploy additional domain controllers in a second region for future expansion of Contoso's Azure footprint.

## Task 1: Create Resilient Active Directory Deployment

**Tasks to complete**

- Create two Domain Controllers (DCs) in the first region: ContosoDC01, ContosoDC02

**Exit criteria**

- The DCs should be configured for resiliency in availability sets and with managed disks
- The DCs should also be configured for Azure Backup

## Task 2: Create the Active Directory deployment in the second region

**Tasks to complete**

- Create two Domain Controllers (DCs) in the second region: ContosoDC03, ContosoDC04

**Exit criteria**

- The DCs should be configured for resiliency in availability sets and with managed disks
- The DCs should also be configured for Azure Backup

## Task 3: Add data disks to Active Directory domain controllers (both regions)

**Tasks to complete**

- Add an additional data disk (managed) to each of the domain controllers

**Exit criteria**

- Each DC should have an additional SSD based 1023 GB managed disk attached

## Task 4: Format data disks on DCs and configure DNS settings across connection

**Tasks to complete**

- Format the disks as the F: drive on each of the VMs
- Configure the virtual networks in each region to reference the IPs of the new Domain Controllers
- Run the following script on the ADVM virtual machine:
- Set-DnsServerPrimaryZone -Name contoso.ins -DynamicUpdate NonsecureAndSecure

**Exit criteria**

- Each DC should have an additional SSD based 1023 GB managed disk attached and formatted
- The Virtual Networks in each region should reference the local DCs in each region
- The Set-DnsServerPrimary zone cmdlet should be executed on ADVM

### Task 5: Promote DCs as additional domain controllers

**Tasks to complete**

- Promote the four DCs to join the contoso.ins Active Directory domain

**Exit criteria**

- All the DCs should be Domain Controllers

### Summary

In this exercise, you will deploy Windows Server Active Directory configured for resiliency using Azure Managed Disks and Availability Sets in the primary and the failover region.

## Exercise 3: Build web tier and SQL for resiliency

Duration: 60 minutes

In this exercise, you will deploy resilient web servers using VM scale sets and a SQL Always-On Cluster for resiliency at the data tier.

### Task 1: Deploy SQL Always-On Cluster

In this task, you will deploy a SQL Always-On cluster using an ARM template that deploys to your existing Virtual Network and Active Directory infrastructure.

**Tasks to complete**

- Navigate to https://github.com/opsgility/cw-building-resilient-iaas-architecture-sql and click the **Deploy to Azure Button**. Deploy the template to the ContosoCloudShopRG resource group in the West Central US region.
- After the template is deployed, execute the following command on SQLVM-1:

```
New-Cluster -Name CLUST-1 -Node SQLVM-1,SQLVM-2,WITNESSVM -StaticAddress
10.22.1.8
```

- Enable SQL Server AlwaysOn on SQLVM-1 and SQL VM-2 and change the service login for both to Contoso\adadmin
- Copy the script from: **C:\HOL\CreateSQLAG.sql** on the LABVM to **C:\SQATA on SQLVM-1**. Execute the script in cmd mode.

**Exit criteria**

- SQL AlwaysOn Availability groups should be deployed

### Task 2: Convert the SQL deployment to Managed Disks

In this task, you will convert the disks of the SQL deployment to managed disks. This task could be automated as part of the template deployment; however, it is important to understand how to migrate existing infrastructure to managed disks.

**Tasks to complete**

- Execute the following script to convert the SQL disks to managed

```
    <#
    The following code converts the existing availability to aligned/managed
and then converts the disks to managed as well.

    Note: the PlatformFaultDomainCount is set to 2 - this is because the region
currently only supports two managed fault domains
    #>

    $rgName = 'ContosoCloudShopRG'

    $avSetName = 'SQLAVSet'

    $avSet = Get-AzureRmAvailabilitySet -ResourceGroupName \$rgName -Name
\$avSetName

    $avSet.PlatformFaultDomainCount = 2

    Update-AzureRmAvailabilitySet -AvailabilitySet $avSet -Sku Aligned

    foreach($vmInfo in $avSet.VirtualMachinesReferences)

    {
        $vm = Get-AzureRmVM -ResourceGroupName $rgName | Where-Object {$_.Id -
eq $vmInfo.id}

        Stop-AzureRmVM -ResourceGroupName $rgName -Name $vm.Name -Force

        ConvertTo-AzureRmVMManagedDisk -ResourceGroupName $rgName -VMName
$vm.Name
    }
```

**Exit criteria**

- All the disks for the SQL deployment should be managed

### Task 3: Build a scalable and resilient web tier

In this task, you will deploy a VM scale set that can automatically scale up or down based on the CPU criteria. The application the scale set deploys points to the new SQL AlwaysOn availability group created previously.

**Tasks to complete**

- Navigate to https://github.com/opsgility/cw-building-resilient-iaas-architecture-ss and click the **Deploy to Azure Button**. Deploy the template to the ContosoCloudShopRG resource group in the West Central US region.

**Exit criteria**

- The scale set should be deployed, and you should be able to browse the CloudShop application from the public IP address assigned to the load balancer

### Summary

In this exercise, you deployed resilient web servers behind a load balancer, and a SQL Always-On Availability Group for database resiliency.

## Exercise 4: Configure SQL Server Managed Backup

Duration: 15 minutes

In this exercise, you will configure SQL Server Managed Backup to back up to an Azure Storage Account.

### Task 1: Create an Azure Storage Account

**Tasks to complete**

- Create a storage account for SQL server backup data by executing the following PowerShell script on your LABVM

```
$storageAcctName = "[unique storage account name]"

$resourceGroupName = "ContosoCloudShopRG"
$containerName= "backups"
$location = "West Central US"
$storageSkuName = "Standard_LRS"


"Creating Storage Account $storageAcctName"
$sa = New-AzureRmStorageAccount -ResourceGroupName $resourceGroupName  `
                                -Name $storageAcctName `
                                -Location $location `
                                -SkuName $storageSkuName
```

```
$storageKey = (Get-AzureRmStorageAccountKey -Name $storageAcctName -
ResourceGroupName $resourceGroupName )[0].Value
$context = New-AzureStorageContext -StorageAccountName $storageAcctName -
StorageAccountKey $storageKey


Write-Host "Creating New Storage Container  $containerName"
New-AzureStorageContainer -name $containerName -permission container -
context $context


$fullSasToken = New-AzureStorageContainerSASToken -Name $containerName -
Permission rwdl -FullUri -Context $context
$containerUrl = $fullSasToken.Substring(0,$fullSasToken.IndexOf("?"))
$sasToken = $fullSasToken.Substring($fullSasToken.IndexOf("?")+1)


$enableManagedBackupScript = @"
--------------------
---BEGIN TSQL Script
--------------------
CREATE CREDENTIAL [$containerUrl]
WITH IDENTITY = 'Shared Access Signature',
     SECRET = '$sasToken'

GO

EXEC msdb.managed_backup.sp_backup_config_basic
 @enable_backup = 1,
 @database_name = 'AdventureWorks',
 @container_url = '$containerUrl',
 @retention_days = 30

 --------------------
 ---END TSQL Script
 --------------------
"@


write-host $enableManagedBackupScript
```

- Copy the generated tSQL code to notepad for later use

**Exit criteria**

- A storage account for SQL Server managed back and code to create an identity in SQL Server should be ready

## Task 2: Configure managed backup in SQL Server

**Tasks to complete**

- Execute the following tSQL code on SQLVM-1 to enable the SQL Agent:

```
EXEC sp_configure 'show advanced options', 1
GO
RECONFIGURE
GO
EXEC sp_configure 'Agent XPs', 1
GO
RECONFIGURE
GO
```

- Execute the code from the previous task that was copied to notepad on SQLVM-1
- Execute the following code to create a custom backup schedule:

```
USE msdb;
GO
EXEC managed_backup.sp_backup_config_schedule
     @database_name =  'AdventureWorks'
    ,@scheduling_option = 'Custom'
    ,@full_backup_freq_type = 'Weekly'
    ,@days_of_week = 'Monday'
    ,@backup_begin_time =  '17:30'
    ,@backup_duration = '02:00'
    ,@log_backup_freq = '00:05'
GO
```

- Execute the following code to create a backup immediately:

```
EXEC msdb.managed_backup.sp_backup_on_demand
@database_name  = 'AdventureWorks',
@type ='Database'
```

**Exit criteria**

- SQL Server should be configured to backup to an Azure Storage account based on your custom schedule
- SQL Server backup data should in the backups container of the Azure Storage Account

## Exercise 5: Validate resiliency

### Task 1: Validate resiliency for the CloudShop application

**Tasks to complete**

- Spike the CPU of the Cloud Shop application by clicking the CPU spike button on the web apps home page

**Exit criteria**

- After 15-20 minutes, new instances should spin up automatically from the auto scale rules

### Task 2: Validate SQL Always On

**Tasks to complete**

- Within the Azure portal, click on Virtual Machines and open **SQLVM-1.** Click **Stop** at the top of the blade to shut the virtual machine off.

**Exit criteria**

- **The SQL AlwaysOn cluster should automatically failover to SQLVM-2**

### Task 3: Validate backups are taken

**Tasks to complete**

- Open the Azure Backup Vaults created earlier and ensure that backup data for the VMs is present
- Open the container for the SQL Server backup storage account and ensure backup data is present

**Exit criteria**

## After the hands-on lab

Duration: 10 minutes

### Task 1: Delete the resource groups created

- Within the Azure portal, click Resource Groups on the left navigation
- Delete each of the resource groups created in this lab by clicking them followed by clicking the Delete Resource Group button. You will need to confirm the name of the resource group to delete.

You should follow all steps provided *after* attending the hands-on lab.