# Using Amazon Web Services and DFS Replication for Disaster Recovery of File Servers

July 2015

# Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

# Contents

# Abstract

Businesses of all sizes maintain file server infrastructure for storage and sharing of corporate documents. Although many businesses have recovery plans in place, they are often rarely tested or rely on traditional backup solutions that may not always meet the recovery time objectives (RTO) or recovery point objectives (RTO), particularly for large file servers.

This paper describes a step-by-step approach to implementing a proven and cost-effective disaster recovery solution for Windows-based file servers that can minimize data loss and provide fast, automatic recovery of file services running on the AWS cloud.

# Introduction

Businesses are using the AWS cloud to enable faster recovery of their critical IT systems without incurring the expense or operational overhead of a second physical site.

AWS offers a highly flexible platform, where resources like corporate file servers can be provisioned on demand, on a pay-as-you-go basis, and scaled to meet the needs of businesses over time.

For businesses that do not have a disaster recovery strategy in place or have found disaster recovery to be cost-prohibitive, AWS is a cost-effective, simple, and flexible solution.

For business that already have disaster recovery facilities and strategies in place, AWS can lower your costs; improve recovery time through scripting, automation and managed service offerings; and, by providing access to more than ten AWS regions, add geographic redundancy for your systems and data.

# Planning the Deployment

This section describes requirements, dependencies, and design and architecture limitations.

## Identify the Requirements

Here are some things to consider before you start the implementation:

- **RPO and RTO** – This architecture implements asynchronous or near real-time data replication and fast failover (generally 90 seconds) using Microsoft DFS Namespaces. If you require zero data loss or instant failover, then you may need to consider other options.
- **Data staging** – The size of your file server data, your available network bandwidth, and your timeframe will determine whether you need to either stage (pre-seed) most of the data on physical media and ship it to AWS using AWS Import/Export or stage the data across the network using a copy operation. [1]

  You can use the formula below to guide you:

  - Where,
    c = estimated convergence time (hours),
    a = total data size (GB),
    b = average usable network bandwidth (Mbps) assuming 60% efficiency

$$c^t = \frac{a/\left(\frac{b \times 0.6}{8000}\right)}{3600}$$

  If convergence time is greater than three days, then you may want to consider using AWS Import/Export to ship your data on physical hard drive devices.

## Distributed File System Replication

Distributed File System Replication (DFSR) is a Microsoft multi-master file replication engine designed to keep files synchronized across servers. DFSR uses

RPC to communicate; it implements Remote Differential Compression (RDC) to efficiently update files over limited bandwidth networks.

A DFS namespace provides a virtual or abstracted view of files and folders hosted on multiple servers for improving availability and performance.

Both DFSR and DFS Namespaces will be used in this paper.

Microsoft Windows Server 2012 R2 introduces two major feature enhancements to DFSR:

- Database cloning to improve initial sync performance.
- Cross-file RDC to improve WAN data transfer performance.

## Dependencies

In addition to the following technical requirements, you should also have some experience with Windows PowerShell, AWS Identity and Access Management (IAM), Amazon Elastic Compute Cloud (Amazon EC2), and Amazon Virtual Private Cloud (Amazon VPC).

### Technical Requirements

- File servers running Windows Server 2012 R2.
- An Active Directory domain with at least one domain controller.
- Active Directory schema updated to Windows Server 2008 R2 or later.

**Note** If your file servers are running Windows Server 2008 R2, you can do a simple in-place upgrade to Windows Server 2012 R2.

## Limitations

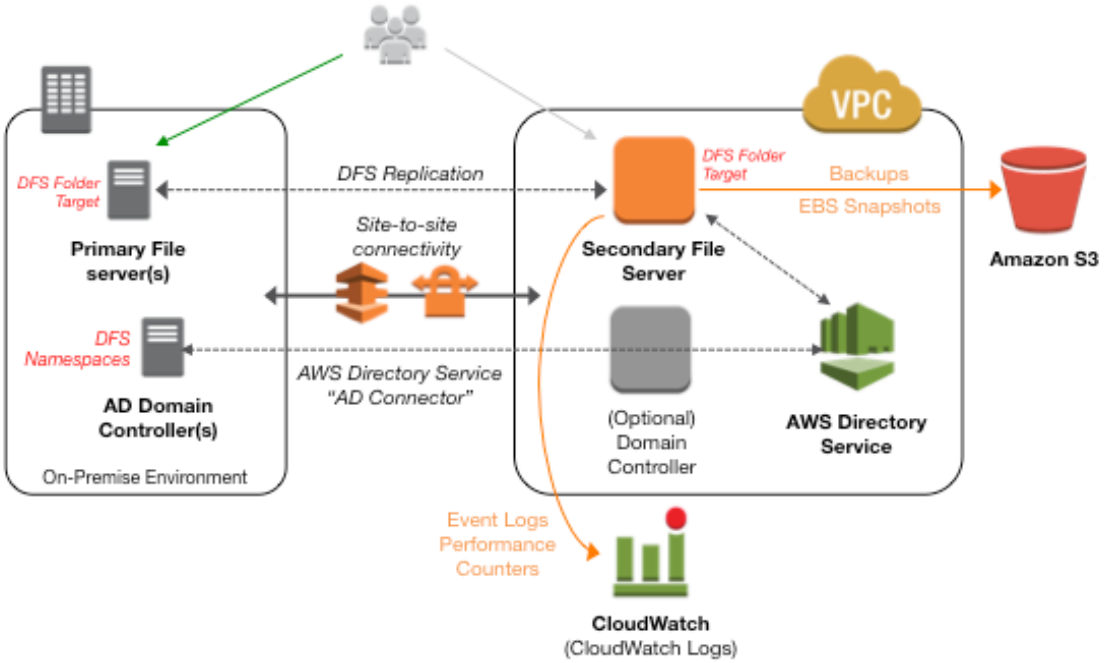
DFS Replication has been tested to support volumes up to 100 TB and 70 million files [2].

Amazon Elastic Block Store (Amazon EBS) supports up to 16 TB volumes. If your dataset is larger than 16 TB, then you can either partition the data into individual volumes (recommended) or create one contiguous volume using a software-based

RAID stripe across several Amazon EBS volumes. If you choose the second option, Amazon EBS snapshots should not be used for backups.

# Implementation

The following diagram shows the proposed architecture:




This section describes the implementation steps. You will start by establishing network and directory services connectivity; then launch and configure a file server instance; then configure the file server instance for DFSR; and then finally set up a DFS namespace for automatic failover.

## Step 1: Connect Corporate Networks

AWS provides two options for seamlessly, reliably, and securely extending your on-premises, private networking environments into the cloud.

- Site-to-site VPN: encrypted VPN tunnels across the Internet.
- AWS Direct Connect: private connectivity over a fixed or leased line.

For an outline of the concepts and implementation requirements for each, see the AWS VPC Connectivity Options [3] white paper.

VPC subnet network ACLs and on-premises firewalls must be configured to enable communication between the on-premises servers (including Active Directory and file servers) and AWS instances. The following ports must be open:

| UDP | TCP |
|---|---|
| 53 | 53 |
| 88 | 88 |
| 135 | 135 |
| 137-139 | 137-139 |
| 389 | 389 |
| 445 | 445 |
| | 636 |
| | 5985 |
| | 3268-3269 |
| | 49152-65535 |

## Step 2: Extend Directory Services

This step is optional, but recommended.

AWS Directory Service is a fully managed domain controller-compatible service. You can use the AD Connector feature in AWS Directory Service to simply, securely, and easily enable administrative single sign-on (SSO) to the AWS Management Console and automatically join an Amazon EC2 for Windows instance to your on-premises Active Directory Domain Services (AD DS) environment.

- Set up the AWS Directory Service AD Connector [4] feature.

**Note** AD Connector does not replicate any of your directory data. Instead, it acts as a proxy to directory requests such as authentications, domain joins, and directory lookups to your on-premises directory. AD Connector cannot be used as a disaster recovery solution for Active Directory environments.

If you are unable to use the AWS Directory Service, or if you would like to have your Active Directory domain replicated and available on AWS for disaster recovery purposes, see the Implement Active Directory Domain Services in the AWS Cloud [5] white paper.

## Step 3: Provision a Windows File Server

In this step, you will use IAM, Amazon EC2, and Amazon EC2 Simple Systems Manager (SSM) to configure your Windows file server instance.

1. Follow the steps in the Amazon EC2 User Guide [6] to create an IAM role and associate a trust policy with the role. Use the AmazonEC2RoleForSSM template.
2. Select a region and launch a new Amazon EC2 instance for your file server.
    a. For **Instance type,** see the following table.

| | Small | Medium | Large |
|---|---|---|---|
| **EC2 Instance Type** | c4.large | c4.2xlarge | c4.8xlarge |
| **CPU*** | 2 vCPU | 8 vCPU | 36 vCPU |
| **CPU Type** | Intel Xeon E5-2666 v3 (Haswell) | | |
| **Memory (GiB)** | 3.75 | 15 | 60 |
| **OS** | Windows 2012 R2 | | |
| **Dedicated EBS Throughput** | 500 Mbps | 1,000 Mbps | 4,000 Mbps |
| **Recommended Storage Type** | EBS Magnetic or General Purpose SSD | EBS General Purpose SSD | EBS Provisioned IOPS (SSD-based) |
| **Baseline Performance** | 40-200 IOPS / Volume | 3 IOPS / GB (3,000 Burst <= 1TB) | 30 IOPS / GB |

    b. For **Domain Join Directory**, choose your directory, or skip this step.
    c. For **IAM Role**, choose the role you created in step 1.
    d. For **Network and Subnet**, specify your VPC and subnet.

e. For **Advanced Details – User Data**, copy and paste the following code to install the File Services role onto the instance and enable deduplication.

```
<powershell>
# Install Roles and Features
Add-WindowsFeature File-Services –IncludeManagementTools
Add-WindowsFeature Fs-Dfs-Namespace, Fs-Dfs-Replication,
Rsat-Dfs-Mgmt-Con, Fs-Data-Deduplication
# Enable De-duplication
Get-Volume |
Where-Object {$_.DriveLetter –ne “`0” –And $_.DriveLetter –
ne “C” } |
Enable-DedupVolume
</powershell>
```

f. For **EBS Storage**, using the preceding table for reference, create an additional Amazon EBS volume. For more information, see the Limitations section of this white paper.

g. For **EBS Encryption**, choose enabled. (Recommended)

h. **Security Group Settings**: Configure the following ports for inbound connection:

| UDP | TCP |
|---|---|
| 135 | 135 |
| 137-139 | 137-139 |
| | 445 |
| 3389 | 3389 |
| | 5985 |
| 49152-65535 | 49152-65535 |

**Note** For SSM to function and to join the domain, the instance must have Internet access.

3. Wait approximately 10 minutes for the instance to be launched and configured. Remote Desktop Protocol (RDP) access will be blocked until configuration is complete.
4. Log in to the instance and confirm:
    a. The instance is joined to your Active Directory domain.
    b. The Windows File Services and DFS features have been installed.
    c. Data volumes have been mounted and formatted.
    d. Deduplication has been configured.
5. On the **EC2Config Service Settings** dialog box, select the **Amazon CloudWatch Logs integration** check box.

Next, set up basic Window event log and performance counter monitoring using CloudWatch logs and Amazon EC2 SSM.

6. In the AWS Management Console, open CloudWatch and create a new CloudWatch log group. You will use this log group later.
7. Follow the steps in Sending Performance Counters to CloudWatch and Logs to CloudWatch Logs [7].
    a. For a sample Amazon EC2 SSM configuration document, see Appendix B.

## Step 4: Establish Replication Using DFSR

Now you will use Windows PowerShell to create a DFSR group on the primary file server.

Before you begin, be sure you have values for each of the properties in the following table because they will be used in the scripts provided.

| | Property | Description | Example |
|---|---|---|---|
| **Replicated Folder Name** | <Folder> | Logical folder name | *CorporateShare* |
| **File Server Source** | <SourceHost> | Host name | *FS1* |
| **File Server Destination** | <DestHost> | Host name | *FS2* |
| **Content Path (Source)** | <ContentPath> | Actual file location | *D:\Corporate* |

| | Property | Description | Example |
|---|---|---|---|
| **Database Clone Path (Source)** | <ClonePath> | Temporary path used to export database | D:\DFSRClone |
| **Database Clone Drive (Source)** | <CloneDrive> | Volume where database will reside (above) | D: |
| **Content Path (Destination)** | <DestContentPath> | Location of files on secondary server or physical media | \\FS2\D$\Data or F:\Data (direct attached hard drive for AWS Import/Export) |
| **DFSR Database Clone Path (Destination)** | <DestClonePath> | Location of DFSR database export on secondary server or physical media | \\FS2\D$\Dfsrclone or F:\Dfsrclone (direct attached hard drive for AWS Import/Export) |
| **SMB Share** | <ShareName> | Share name for DFS Root | E.g. Corporate |
| **DFS Root Name** | <DFSRootName> | DFS Share Name | E.g. Corporate |
| **DFS Share Name** | <DFSShareName> | DFS Share Name | E.g. Corporate |
| **Domain Controller Host** | <DCHostName> | Domain Controller NetBIOS name | AD1 |

1. Log in to the primary file server.
2. Create and configure the replication group.

```
New-DfsReplicationGroup -GroupName "RG1"
New-DfsReplicatedFolder -GroupName "RG1" -FolderName <Folder>
Add-DfsrMember -GroupName "RG1" -ComputerName <SourceHost>
Set-DfsrMembership -GroupName "RG1" -FolderName <Folder> -ContentPath
<ContentPath> -ComputerName <SourceHost> -PrimaryMember $True
Update-DfsrConfigurationFromAD -ComputerName <SourceHost>
```

3. Wait for DFS Replication event ID 4112, which confirms that the service initialized the replicated folder.

```
Get-WinEvent "Dfs replication" -MaxEvents 10 | fl
```

Use DFSR database cloning to speed up the initial sync process.

4. Clone and export the DFSR database.

```
New-Item -Path "<ClonePath>" -Type Directory
Export-DfsrClone -Volume <CloneDrive> -Path "<ClonePath>"
```

5. Wait for DFS Replication event ID 2402, which indicates the export process was completed successfully.
6. Now copy the files either directly across the network to the secondary server, or to a physical media device to be used by AWS Import/Export.

```
Robocopy.exe "<ContentPath>" "<DestContentPath>" /E /B /COPYALL
/R:6 /W:5 /MT:64 /XD DfsrPrivate /TEE /LOG+:preseed.log
Robocopy.exe "<ClonePath>" "<DestClonePath>" /B
```

7. If you are staging (pre-seeding) the files to physical media, follow the steps in the AWS Import/Export Developer Guide [8] to ship the physical media and import data into an Amazon EBS snapshot.
8. (Optional) Verify the integrity of the data. Use the PowerShell `Get-DfsrFileHash` command to compare the signatures on a sample of files on each server.

Now that the data has been staged on AWS, you will import the DFSR database and establish DFSR connections between the primary and secondary servers so that any remaining files can be synchronized.

9. Log in to the secondary server running on AWS.
10. Import the DFSR database.

```
Import-DfsrClone -Volume D: -Path "<DestClonePath>"
```

11. Check CloudWatch logs for event ID 2404 in the DFS Replication log. This indicates the clone was imported successfully.
12. Establish the connection between the primary and secondary servers.

```
Add-DfsrMember -GroupName "RG1" -ComputerName "<DestHost>" | Set-
DfsrMembership -FolderName "<Folder>" -
ContentPath "<DestContentPath>"
Add-DfsrConnection -GroupName "RG1" -
SourceComputerName "<SourceHost>" -
DestinationComputerName "<DestHost>"
Update-DfsrConfigurationFromAD <SourceHost>,<DestHost>
New-SmbShare -Name <ShareName> -Path <DestContentPath>
```

13. Monitor replication progress.

```
# Unreplicated changes count
(Get-DfsrBacklog -GroupName "RG1" -FolderName "<Folder>" -
SourceComputerName "<SourceHost>" -DestinationComputerName
"<DestHost>" -Verbose 4>&1).Message.Split(':')[2]

# Detailed report
Get-DfsrState -ComputerName "<DestHost>" | Format-Table filename,
updatestate,inbound,source* -auto -wrap
```

## Step 5: Implement a DFS Namespace

In this section, you will create a domain-based DFS namespace.

1. Create a share for the new DFS root on each domain controller, if required.

```
New-Item -Path \\<DCHost\c$\DFSRoots\<DFSRootName> -type
directory -Force
New-SmbShare -Name <DFSRootName> -Path C:\DFSRoots\<DFSRootName>
-ReadAccess everyone -CimSession <DCHost>
```

2. Create the DFS namespace on the domain controllers. (Repeat for each domain controller.)

```
New-DfsnRoot -TargetPath "\\<DCHost>\<DFSRootName>" -Type
DomainV2 -Path "\\corp.local\<DFSRootName>"
```

3. Finally, set the folder targets for the namespace, ensuring that the AWS file server instance is set to the lowest priority class so that it is activated only if the primary file server fails.

```
New-DfsnFolderTarget -Path
"\\<Domain>\<DFSRootName>\<DFSShareName>" -TargetPath
"\\<SourceHost>\<ShareName>" -ReferralPriorityClass GlobalHigh
New-DfsnFolderTarget -Path
"\\<Domain>\<DFSRootName>\<DFSShareName>" -TargetPath
"\\<DestHost>\<ShareName>" -ReferralPriorityClass GlobalLow
```

**Note** When you use PowerShell to manually create DFS folder targets, the DFS management console will not update the replication group publication status or the **Folder Targets** tab. If you would like these items to appear, then after you have completed step 2, open the DFS console, publish the replication group to your namespace, and set the referral ordering through the GUI.

## Step 6: Configure Backups

Point-in-time volume-based Amazon EBS snapshots can be used to quickly recover from Amazon EBS volume failures, data corruption, and the accidental or malicious erasure of data.

1. Update the existing IAM role to include the appropriate permissions to create snapshots. For an example IAM policy, see Appendix A.
2. Use Task Manager or a similar tool to create a scheduled task that executes the following PowerShell script. This script will automate the creation of Amazon EBS snapshots for the instance on which it is running.

The script will tag both the instance and the snapshots with metadata, such as the original device name, owner, and backup time and date.

```
# .NOTE Script excludes the boot volume
# .NOTE Set the default region first, Set-DefaultAWSRegion <awsregion>
$myInstanceID = (New-Object
System.Net.WebClient).DownloadString("http://169.254.169.254/latest/meta-
data/instance-id")
$volumes = (Get-EC2Volume).Attachment | where {$_.InstanceId -eq
$myInstanceID -and $_.Device -ne '/dev/sda1'} | Select VolumeId, Device
  foreach ($volume in $volumes)
  {
     $dt = (Get-Date).DateTime
     $snapshot = New-EC2Snapshot $volume.VolumeId -Description ("["+
$myInstanceID + "] backup at " + $dt)
     New-EC2Tag -Resource $myInstanceID -Tag @{ Key="last-backup"; Value=$dt
}
     New-EC2Tag -Resource $snapshot.snapshotId -Tag @( @{ Key="backup-
datetime"; Value=$dt }, @{ Key="owner"; Value=$myInstanceID}, @{ Key="from-
device"; Value=$volume.Device})
   }
```

**Note** Restoring snapshots of DFS replicated folders can cause DFS database corruption. To avoid this, create a new Amazon EBS volume from the snapshot, attach it to the instance, and then selectively copy the required files.
For more information, see this Microsoft Support Article. [9]

## Step 7: Test Failover/Failback

Now that your servers are configured, DFSR is functioning, and the namespace is active, perform some basic failover testing.

1. Log in to a Windows client.
2. Create the following PowerShell script and save it as a `.ps1` file. The script continuously performs a directory listing of your DFS share.

```
while ($true) {
```

```
 ls \\<Domain>\<DFSRootName>\<DFSShareName> | select -last 3 |
format-table (Get-Date).DateTime, FullName
 sleep  1
 }
```

3. Run the script.
4. Shut down the primary file server.
5. Observe the timing in the script to ensure the client successfully reconnects and the directory listings continue (typically around 90 seconds).
6. Test failback by turning on the primary server and shutting down the secondary server running on AWS.

# Alternatives

Amazon WorkDocs [10] is an alternative to traditional workgroup or corporate file servers. This fully managed, secure enterprise storage and sharing service provides strong administrative controls and feedback capabilities to improve user productivity. You can securely access files stored in Amazon WorkDocs from mobile and desktop devices using integrated single sign-on (SSO) to your existing Active Directory domain.

# Conclusion

File server disaster recovery environments can be built and run on AWS using familiar tools like Microsoft DFS Replication. This paper has outlined the steps and best practices for deploying a DFS-based solution while leveraging services like AWS Directory Service and Amazon CloudWatch to simplify administration and operations.

Companies of all sizes can benefit from AWS security, automation, global data center footprint, and low cost of operating disaster recovery environments.

# Document Revisions

June 2015 – Initial revision.

# Appendix A: IAM Policy for Amazon EBS Snapshots

Here is an example IAM policy for Amazon EBS snapshots.

```
{
        "Version": "2012-10-17",
        "Statement": [
          {
          "Sid": "Stmt1426256275000",
          "Effect": "Allow",
          "Action": [
            "ec2:CreateSnapshot",
            "ec2:CreateTags",
            "ec2:DeleteSnapshot",
            "ec2:DescribeSnapshotAttribute",
            "ec2:DescribeSnapshots",
            "ec2:DescribeVolumeAttribute",
            "ec2:DescribeVolumeStatus",
            "ec2:DescribeVolumes"
          ],
          "Resource": [
             "*"
          ]
          }
        ]
}
```

# Appendix B: SSM Configuration Document

This snippet sample sends DFS Replication event logs and performance counters to CloudWatch Logs and CloudWatch, respectively. Replace <NAME> and <UUID> with your DFS replicated folder name and UUID.

```
{
     "EngineConfiguration": {
         "PollInterval": "00:00:15",
         "Components": [
              …
              {
                  "Id": "DFSReplicationEventLog",
                  "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.C
loudWatch",
                  "Parameters": {
                  "LogName": "DFS Replication",
                  "Levels": "7"
                  }
              },
              {
                 "Id": "PerformanceCounterDFS",
                 "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInp
utComponent,AWS.EC2.Windows.CloudWatch",
                 "Parameters": {
                     "CategoryName": "DFS Replicated Folders",
                     "CounterName": "Total Files Received",
                     "InstanceName": "<NAME>-{<UUID>}",
                     "MetricName": "DFS-Files-Received",
                     "Unit": "Count",
                     "DimensionName": "",
                     "DimensionValue": ""
                  }
              },
         ],
 …
```

# Notes

DFS Replication: Frequently Asked Questions (FAQ)
https://technet.microsoft.com/en-US/library/cc773238.aspx

[1] http://blogs.technet.com/b/filecab/archive/2013/07/31/dfs-replication-in-windows-server-2012-r2-revenge-of-the-sync.aspx

[2] https://technet.microsoft.com/en-US/library/cc773238.aspx#BKMK_007

[3] https://media.amazonwebservices.com/AWS_Amazon_VPC_Connectivity_Options.pdf

[4] http://docs.aws.amazon.com/directoryservice/latest/adminguide/setting_up.html

[5] https://d0.awsstatic.com/whitepapers/Implementing_Active_Directory_Domain_Services_in_the_AWS_Cloud.pdf

[6] http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-configuration-manage.html

[7] http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-configuration-cwl.html

[8] http://docs.aws.amazon.com/AWSImportExport/latest/DG/createEBSimportjobs.html

[9] https://support.microsoft.com/en-us/kb/2517913

[10] http://aws.amazon.com/workdocs/