Deploy App Orchestration 2.6 for High Availability and Disaster Recovery

Qiang Xu, Cloud Services Nanjing Team

Last Updated: Mar 24, 2015
Introduction

App Orchestration provides high availability and failover capability for app and desktop services using a multi-datacenter design. To ensure availability of app and desktop services, delivery site and storefront infrastructure is replicated in multiple datacenters. Deployment of replica or backup sites is handled by App Orchestration using a multi-tenant configuration model, a relational database hosted on Microsoft SQL.

When considering App Orchestration high availability and disaster recovery, there are two scenarios to plan for:
- Availability and failover capability of end-user app and desktop services – making sure end users can always connect to the apps and desktops
- Availability and failover of service provider management tools like App Orchestration – ensuring that service providers can use App Orchestration to manage app and desktop delivery during a disaster event or datacenter outage. App Orchestration features align with disaster planning and recovery use cases, so it make sense to include App Orchestration when failing over business critical infrastructure.

This document focuses on preparing and configuring an App Orchestration environment for high availability and fault tolerance.

SQL server 2012/2014 Enterprise provides a new high availability solution for SQL databases with “AlwaysOn” availability groups. AlwaysOn allows you to create a group of databases which failover together as a unit from one replica/instance of SQL Server to another replica/instance of SQL Server in the same availability group. Placing replicas in different datacenters provides datacenter level redundancy of critical databases.

To support “AlwaysOn” availability group, no special configurations or modifications are needed for App Orchestration. SQL Server provides replication capabilities at the transaction level. The design of AlwaysOn allows for important database management tasks, such as changing or upgrading the DB schema of App Orchestration to later versions. This document will provide the step-by-step guide to enable this feature with App Orchestration databases.

Process Overview

We recommend the following installation sequence and configuration steps to deploy an App Orchestration environment using Microsoft SQL AlwaysOn Availability Groups.

- Select or create a Windows Server failover cluster
- Install SQL Server 2012 on each cluster node
- Create and configure an availability group
- Install and configure App Orchestration multi-datacenter
- Add the App Orchestration databases to availability group
- Test failover for availability group

Before you begin

Microsoft SQL Server 2012 AlwaysOn Availability Groups are an emerging technology for ensuring high availability of critical databases. AlwaysOn replaces traditional database mirroring and database clustering techniques that have historically provided similar functions. Availability groups enable strategies for disaster recovery scenarios that compare to options such as mirroring and log shipping. Before you begin deployment, review the following information about SQL Server AlwaysOn, the technologies that support AlwaysOn, and App Orchestration:
- Knowledge and skill requirements
- AlwaysOn Availability Group concepts

© 2015 Citrix Systems, Inc. All rights reserved
Knowledge and skill requirements
To implement SQL Server AlwaysOn Availability Groups as a high availability and disaster recovery solution, several technologies interact and are required to be installed and configured correctly. We recommend that the team responsible for setting up an AlwaysOn environment for App Orchestration products has a working knowledge of, and hands-on skills with the following technologies:

- Windows Server Failover Clustering (WSFC) services
- SQL Server 2012
- App Orchestration multi-datacenter deployment

High Availability concepts for App Orchestration
- **Multi-Controller Delivery Sites** – several controllers working together to host a single delivery site. Provides high availability of critical functionality such as app and desktop access for end users, session brokering and site configuration.
- **Multi-Server StoreFront Server Groups** – several servers working together to host a single Citrix StoreFront Server Group. Provides high availability of StoreFront sites used for end user app and desktop access.
- **Multi-Configuration Server App Orchestration Deployment** - configure multiple configuration servers to provide high availability of configuration, orchestration and automation capabilities.
- **Multi-Datacenter** – Deliver app and desktop offerings to a tenant configured with a primary and backup datacenter. Delivery Site and StoreFront infrastructure is replicated in primary and backup datacenters, allowing for failover in the event of a datacenter outage.
- **Database Availability** – In this document we ensure high availability and datacenter level replication of SQL databases using Microsoft SQL Server 2012 AlwaysOn Availability Groups.

SQL Server AlwaysOn Availability Group Concepts
A SQL Server Availability Group enables you to specify a set of databases that you want to fail over together as a single entity. When an availability group fails over to a target instance or target server, all the databases in the group also fail over. Because SQL Server 2012 can host multiple availability groups on a single server, you can configure AlwaysOn to fail over to SQL Server instances on different servers. This reduces the need to have idle high performance standby servers to handle the full load of the primary server, which is one of the many benefits of using availability groups.

An availability group consists of the following components:
• Replicas, which are a discrete set of user databases called availability databases that fail over together as a single unit. Every availability group supports one primary replica and up to four secondary replicas.
• A specific instance of SQL Server to host each replica and to maintain a local copy of each database that belongs to the availability group.

For the details about the benefit of AlwaysOn Availability Groups and overview of AlwaysOn Availability Groups terminology, see AlwaysOn Availability Groups(SQL Server).

Windows Server Failover Clustering
To create and use SQL Server 2012 AlwaysOn Availability Groups, you must install SQL Server 2012 on a Windows Server Failover Clustering (WSFC) cluster. For more information, see Windows Server Failover Clustering (WSFC) with SQL Server.

Although configuring a WSFC cluster is out of the scope for this article, you should be aware of the following requirements before you install and configure a cluster:
• All the cluster nodes must be in the same Active Directory Domain Services (AD DS) domain.
• Each availability replica in an availability group must reside on a different node of the same Windows Server Failover Clustering (WSFC) cluster.
• The cluster creator must have the following accounts and permissions:
  o Have a domain account in the domain where the cluster will exist
  o Have local administrator permissions on each cluster node
  o Have Create Computer objects and Read All Properties permissions in AD DS
For more information, see Failover Cluster Step-by-Step Guide: Configuring Accounts in Active Directory

An important aspect of configuring failover clustering and AlwaysOn is determining the quorum votes that are needed for the cluster nodes. Failover clustering is based on a voting algorithm where more than one half of the voters, or quorum, must be online and able to communicate with each other. Because a given cluster has a specific number of nodes and a specific quorum configuration, the cluster service is able to determine what constitutes a quorum. The cluster service will stop on all the nodes if the number of voters drops below the required majority.
For more information, see WSFC Quorum Modes and Voting Configuration (SQL Server) and Configure Cluster Quorum NodeWeight Settings.

Configuring AlwaysOn Availability Groups for App Orchestration
At least, this configuration requires two SQL Server instances to form a single group, and one availability group listener for the database connection.

Prepare the Windows Server Cluster environment
Obtain access to or create a three node Windows Server Failover Clustering (WSFC) cluster that you can use to install SQL Server 2012 on each cluster node. The following reference material provides guidance and detailed steps to configure a Windows Server failover cluster:
• Failover Clusters in Windows Server 2008 R2.
Prepare the SQL Server environment
Before you can create an Availability Group for App Orchestration, you must prepare the SQL Server 2012 environment. To prepare the environment, complete the following tasks:

- Configure iSCSI failover cluster for SQL Server data files (Optional)
- Install the SQL Server prerequisites
- Install SQL Server
- Enable AlwaysOn

Configure iSCSI failover cluster for SQL Server data files (Optional)
By default, the data files for SQL Servers is located on local machine, if you want to provide centralized access for the data files storage and have the failover capability in the cluster, below is one sample:

1. Prepare One iSCSI Target Server On W2K12, add iSCSI Target Server role
2. On the Target server, create iSCSI Virtual Disk file from Server Manager->iSCSI
3. On each SQL Servers, add File Server role, open Server Manager, Tools->iSCSI Initiator, in the Target field, input FQDN/IP address of Target Server configured in step 1 and connect, switch to “Volumes and Devices” Tab, click Auto Configure.
4. Open the Failover Cluster Manager, follow the steps for Configure Role for File Server, right click on the Disks, Add Disk to the cluster
5. Share the disk on Failover Cluster Manager by right click on the role, click “Add File Share”
6. On the each SQL Servers, map the shared folder to network Drive, now this drive can be used later for the SQL Server data files

You may have other types of storage solution based on your business requirement, it will not be covered in this article

Install SQL Server 2012
To install SQL Server 2012

1. Install SQL Server 2012 prerequisites on each cluster node.
   For more information, see Prerequisites, Restrictions, and Recommendations for AlwaysOn Availability Groups (SQL Server).
2. Install SQL Server on each cluster node.
   Choose the 2nd option “New SQL Server failover cluster installation”, For more information, see Installation for SQL Server 2012.
During Database Engine Configuration, specify the Data Directories to iSCSI clustered share folder (mapped network drive mentioned in “Configure iSCSI failover cluster for SQL Server data files (Optional)” or leave it as the default.

Enable AlwaysOn
You must enable AlwaysOn for each database servers in the cluster.

To enable AlwaysOn
1. Your logon account must have the permission levels to create an availability group. The account must have membership in the **db_owner** fixed database role and either CREATE AVAILABILITY GROUP server permission, CONTROL AVAILABILITY GROUP permission, ALTER ANY AVAILABILITY GROUP permission, or CONTROL SERVER permission, it recommended to change the logon account to the **same domain user account** for each instance.
2. Log on to the server that will host the primary replica and start SQL Server Configuration Manager.
3. In Object Explorer, select **SQL Server Services**, right-click **SQL Server (<instance name>)**, where <instance name> is the name of a local server instance for which you want to enable AlwaysOn Availability Groups, and then click **Properties**.
4. Select the **AlwaysOn High Availability** tab.
5. Select the **Enable AlwaysOn Availability Groups** check box, and then click **OK**.
6. Although the change is saved you must manually restart the SQL Server service (MSSQLSERVER) to commit the change. The manual restart enables you to choose a restart time that is best for your business requirements.
7. Repeat the previous steps to enable AlwaysOn for SQL Server on the other cluster nodes.

For more information, see [Enable and Disable AlwaysOn Availability Groups (SQL Server)](http://go.microsoft.com/fwlink/p/?LinkId=267140).

Create and configure the availability group
Use the following procedure to create an availability group on the primary replica, which is AO-SRV1 in our example.

If there is no user databases are on the instance of connected server, which is true in our case, since we have not created App Orchestration database yet in this step, you need to create empty AO database before creating the availability group.

To create the empty AO databases
1. Make sure that your logon account has the correct permissions for this task. You require one of the following permissions in the master database to create the new database:
   o CREATE DATABASE
   o CREATE ANY DATABASE
   o ALTER ANY DATABASE
2. Log on to the server that will host the primary replica, which is AO-SRV1 in our example.
4. In Object Explorer, right-click **Databases** and then click **New Database**.
5. In the New Database dialog box, type the Database name, which is "AppOrchestration" for this example. Options -> Collation, in the drop box, select "Latin1_General_100_CI_AS_KS" , Click OK.
Because the New Availability Group Wizard will not create an availability group unless the user database was backed up, you have to back up the database and set the Recovery model to "Full", From Options->Recovery model.
6. In Object Explorer expand Databases and right-click the database that you just created. Pick Tasks and then choose Back Up.
7. In the Back Up Database dialog box, click OK to accept all the default settings and create the back up.
8. Repeat above steps to create another database named "AppOrchestrationLogging"

Create the availability group

1. Make sure that your logon account has the required permissions to create an availability group. This requires membership in the db_owner fixed database role and either CREATE AVAILABILITY GROUP server permission, CONTROL AVAILABILITY GROUP permission, ALTER ANY AVAILABILITY GROUP permission, or CONTROL SERVER permission.
2. Log on to the server that will host the primary replica and start SQL Server Management Studio.
3. To start the New Availability Group Wizard, right-click AlwaysOn High Availability and then click New Availability Group Wizard.
4. Click Next to advance to the Specify Name page. Enter AO-AG1 as the name of the new availability group in the Availability group name: box.
This name must be: a valid SQL Server identifier, unique on the Windows Server Failover Clustering cluster and unique on the domain.
5. On the Select Databases page, all user databases that are eligible to become the primary database for the new availability group are listed on the User databases on this instance of SQL Server grid. Select the database which is "AppOrchestration" and "AppOrchestrationLogging", and then click Next.
6. On the Specify Replicas page, use the following tabs to configure the replicas for AO-AG1: Replicas, Endpoints, and Backup Preferences.
7. An availability group listener is a virtual network name that provides client connectivity to the database a given availability group. Availability group listeners direct incoming connections to the primary replica or to a read-only secondary replica. The listener provides fast application failover after an availability group fails over. For more information, see Availability Group Listeners, Client Connectivity, and Application Failover (SQL Server).
On the Listener tab, configure an availability group listener for our example, use the name AOListener, if your deployment span multi-subnet, you can add multiple IP address to the AOListener’s DNS records to provide fast connection
8. Select the desired configuration for each instance in the Selected instances grid, and then click Next.
9. Click Finish to create the availability group.
10. The **Select Initial Data Synchronization** page lets you select a synchronization preference and **specify the shared network location that all replicas can access**. For our environment accept the default, **Full**, which performs full database and log backups. Click **Next**.

11. The **Validation** page of the wizard displays the results of six checks before it lets you continue with availability group creation. If all checks pass, click **Next** to continue. If any tests fail, you cannot continue until you correct the error and then click **Re-run Validation** to run the validation tests again. When all the tests pass, click **Next** to continue.

12. On the **Summary** page, verify the configuration of the replica that you are adding and then click **Finish** to save it. To change the configuration, click **Previous** to return to previous wizard pages.

### Availability Modes and Failover Modes consideration

AlwaysOn Availability Groups supports two availability modes, asynchronous-commit mode and synchronous-commit mode, as follows:

- **Asynchronous-commit** mode is a disaster-recovery solution that works well when the availability replicas are distributed over considerable distances
- **Synchronous-commit** mode emphasizes high availability over performance, at the cost of increased transaction latency

To balance the database access performance and connectivity, it's recommended to have 3 nodes in the availability group at least, named Node 01, Node 02, Node 03, Node 01 and Node 02 in the main datacenter, have fast local network connection to provide high availability, Node 03 is deployed in remote disaster recovery site, with the below Availability Modes and Failover Modes:

- Node 01 and Node 02 – Synchronous-commit with automatic failover
- Node 03 – Asynchronous-commit with only forced manual failover, if you have dedicate network link between main datacenter and remote site, it’s recommended to set **Synchronous-commit** mode also, since asynchronous-commit may cause potential data loss while the failure happen

Refer to below diagram for example:
Primary replica
Secondary replica
Synchronous-Commit and automatic Failover
Availability Group

See Availability Modes for more details

**Recover from one failure**

If the hardware/network failure happened on the main datacenter (Node 01 and Node 02), below are the steps suggested:

1. Administrator connect to the SQL instance of Node 03 (remote site) and perform a forced failover, right click on the availability group **Failover...**, select the new primary replica “Node 03”, now it will become the primary replica to provide the database services, since it’s Asynchronous-commit mode, there is potential risk for the data loss here, for App Orchestration, it may cause the ongoing workflow fail while the failure happen, the possible mitigation is to **Cancel** and **Retry** the workflow or configure and repeat the workflow again, we only tested the limited scenarios for the potential impact of the data loss in this situation, administrator need to keep the data consistent
2. Once the Node 01 and Node 02 comes back online and re-establish the communication with WSFC cluster, administrator manually resume the database, **Resume an Availability Database (SQL Server)**

To resume a secondary database

1. In Object Explorer, connect to the server instance that hosts the availability replica on which you want to resume a database, and expand the server tree.
2. Expand the AlwaysOn High Availability node and the Availability Groups node.
3. Expand the availability group.
4. Expand the Availability Databases node, right-click the database, and click Resume Data Movement.
5. In the Resume Data Movement dialog box, click OK.

3. Administrator changes the new primary replica to synchronous-commit mode temporarily, it will enable resumed secondary databases to become SYNCHRONIZED,
After this step, Node 03’s database will overwrite the one in Node 01 and Node 02, skip this step if you want to keep Node 01’s database

4. Administrator perform a manual failover to original primary replica and change to failover modes and availability modes to original settings:
   - Node 01 and Node 02 – Synchronous-commit with automatic failover
   - Node 03 – Asynchronous-commit with only forced manual failover

Install and Configure App Orchestration for High Availability

We recommend reading the getting started guide and deploying a Multi-Datacenter Environment in App Orchestration 2.6 to get familiar with App Orchestration multi-datacenter features. When configuring App Orchestration for high availability, use the group listener DNS name for all database server connections during configuration. In this document, we’re using “AOListener” which was created in previous steps.

1. After the installation of configuration server, run “Citrix App Orchestration Server Configuration”->Create a new deployment. In the Database name field, input “AppOrchestration” which is the empty database created in the previous step, in the Database server field, input the availability group listener DNS Name.
2. Install and join the 2nd configuration server to the existing deployment right. After initial installation, launch “Citrix App Orchestration Server Configuration”->Join an existing deployment and input the first configuration server’s address to finish the wizard.
3. Repeat Step 2 to add more configuration servers to the deployment, it’s recommended to have 2 configuration servers in the main and remote datacenter individually to provide the high availability. If one configuration server is down, others will provide the capabilities and features of App Orchestration.

Notes: If you have already configured Delivery Sites and Storefront Server Groups before joining the 2nd configuration server, you’ll need to append the 2nd configuration server address on each App Orchestration Agent machine.

1. Open Registry editor, modify “ConfigurationServiceAddress” under “HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CloudAppManagement\Agent”, append 2nd configuration server information to it, like https://FQDN/cam/api
2. Restart the Agent services

In multi-datacenter deployments, the App Orchestration Agent talks to configuration server randomly, for better performance, check and modify HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CloudAppManagement\Agent, make sure only the configuration server in the same datacenter is listed, which will force Agent talk to the configuration server in the same datacenter.
Database Login Replication

App Orchestration database logins are not replicated in the availability group. This occurs because login information is stored in the master database, the workaround is to manually copy the App Orchestration database’s logins from the primary replica to the secondary replicas.

- Login to primary SQL instance by Management Studio
- Go to each App Orchestration databases->Security->Users, you will find the machine account information like “domain name\host name$”, record all App Orchestration servers information
- Execute the SQL query “CREATE LOGIN [domain name\host name$] FROM WINDOWS WITH DEFAULT_DATABASE = [master]” on each secondary SQL instance’s SQL query’s window, replace “domain name\host name$” by the machine account information from previous step

Repeat above steps if you add any new database to the availability group later or any new configuration servers to the deployment, keep the login account synced manually.

Configure Load Balancing using Netscaler for Configuration Server

To provide highly available access to the App Orchestration management interface, using NetScaler to load balance a single management address across multiple configuration servers.

1. Login to Netscaler, Traffic Management->Servers, add the configuration servers, specify the name and IP address
2. Go to Traffic Management->Services to configure the services, add services definition for https load balancing without SSL offload, in the Protocol, choose “SSL_BRIDGE” and Choose https as Monitors, for each servers
3. Go to Traffic Management->Virtual Servers, add one virtual server with Protocol “SSL_BRIDGE”, set Persistence to SOURCEIP to make sure the transaction is in one same session and activate the services
4. Add one DNS records for the virtual server, login the web console by this FQDN, like https://FQDN/cam/console
5. If failure happen, restart the browser, login again

Disaster Recovery Considerations for Multi-Datacenter

Setup two datacenters to provide ensure high availability and disaster recovery capability. Each datacenter is backed up each by the other, if failure happened on one datacenter, data and configuration will be kept and business continuity will not be interrupted.

1. Setup two or more configuration servers in each datacenter, install and configure Compute resource, DDC, Storefront servers individually on each datacenter.
2. Setup SQL “AlwaysOn” availability group, Join the replicas on each datacenter to the same availability group, that make all the datacenter access the same database
3. Add the configuration servers to one load balancer, and create DNS records for the virtual server, this virtual server name will be the server name of AO web console
4. If failure happened on one datacenter, since database is synced to another datacenter’s SQL replica, load balancer will failover to another datacenter’s configuration server,
admin/end user can access the resource from live datacenter without any additional configuration

Using Failover Tests to Validate AlwaysOn Installations

After synchronizing App Orchestration data with the secondary replicas, the final step is to test failover. It is recommended to run extensive failover tests to ensure behavior of the AlwaysOn environment is as expected, and that operators completely understand the configuration requirements and procedures related to SQL Server 2012 Availability Groups.

Firstly, test availability group failover by using either the planned manual failover described in Perform a Planned Manual Failover of an Availability Group (SQL Server) or the forced manual failover described in Perform a Forced Manual Failover of an Availability Group (SQL Server).

You can perform either of the previous failovers by using the Failover Wizard in SQL Server Management Studio, Transact-SQL, or Windows PowerShell in SQL Server 2012.

Then test the database connectivity and App Orchestration functionality, these tests include and are not limited to the following:

- Verify that published App/Desktop from different datacenter are completely functional.
- Can login to configuration server web console, verify that App Orchestration data is preserved and not corrupted.

Migrate Existing Database Deployments to “AlwaysOn” Availability Groups

If you have already deployed the App Orchestration databases, connecting to the SQL instance directly and want to add HA/DR capability by enabling “AlwaysOn” availability group, you need to:

- Backup the databases (2 databases need to backed up, AppOrchestration and AppOrchestrationLogging)
- Restore the databases to the primary replica
- Add the App Orchestration databases to the availability group
- Modify the databases connection string pointing the connection to group listener

Restore the databases on the primary replica

After restore, you need create the login account manually on the primary replica and each secondary replicas in the group, detail steps, refer to Login Replication section

To modify the databases connection string

On the configuration servers, open the registry editor, find the key “DatabaseConnectionString”
Under "HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CloudAppManagement\Configuration", replace current SQL server FQDN with group listener's VNN, other connection string should be modified also in the follows:

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\C\loudAppManagement\Connections\ConnectionString

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XDservices\ADIdentitySchema\DataStore\Connections\ConnectionString

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XDservices\ConfigurationSchema\DataStore\Connections\ConnectionString

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XDservices\XDservices\DataStore\Connections\ConnectionString

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XDservices\ConfigurationSchema\DataStore\Connections\ConnectionString

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XDservices\DesktopUpdateManagerSchema\DataStore\Connections\ConnectionString

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XDservices\HostingUnitServiceSchema\DataStore\Connections\ConnectionString

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\XDservices\TrustServiceSchema\DataStore\Connections\ConnectionString

Upgrade App Orchestration 2.5 to later version

If you have already deployed AO 2.5 with “AlwaysOn” availability group, and want to upgrade to later version, for example AO 2.6, for the configuration server role upgrade, you need to:

- Make sure the databases are synced between main datacenter and remote datacenter before the upgrade
- No in progress workflow and other operation on the configuration server

Force the databases synced between replicas if there is asynchronous-commit mode

1. In Object Explorer, connect to the server instance that hosts the primary availability replica and expand the server tree.
2. Expand the AlwaysOn High Availability node and right click on the Availability Groups node->Properties, set all the replicas Availability Mode to “Synchronous commit”, which will force the databases are synced
3. Connect to each server instance and check if the databases are in “Synchronized” status, after all the databases are in synchronized status, go to next step

Upgrade AO configuration servers

1. From the first App Orchestration configuration server, browse to CloudAppManagement in the ProductMedia folder and run Setup, the installer detects that App Orchestration 2.5 is installed and prompts you to start the upgrade.
2. Close the setup dialog box, when prompted, click Yes to restart the machine

© 2015 Citrix Systems, Inc. All rights reserved
3. Run Citrix App Orchestration Server Configuration to upgrade the database schema and environment configuration
4. Repeat above steps on other configuration servers you want to upgrade
5. Click the **Close** button to launch the web console, check the settings
6. Revert the Availability Mode to original settings