

Step-by-Step Guide: BigFix Root Server & SQL Server High Availability with Always On

Phase 1: Windows Server Failover Cluster (WSFC) Foundation

The WSFC is the underlying technology that enables high availability for both your BigFix Root Servers and SQL Servers. You will create a single WSFC encompassing all 6 servers (3 BigFix, 3 SQL).

1. Prerequisites for all Cluster Nodes (3 BigFix Root Servers, 3 SQL Servers)

Before creating the cluster, ensure all your servers meet these fundamental requirements:

- **Operating System:** All servers must be running the **same version of Windows Server**.
- **Active Directory Domain:** All servers must be **joined to the same Active Directory domain** and be in the same time zone as your local domain controller.
- **Domain Controller Location:** The **domain controller should not be hosted on any of the cluster machines**.
- **Account Permissions:** The user account used to create the cluster must be a **domain user with administrator rights on all servers** that will become cluster nodes. This user also needs the **Create Computer objects** permission in the Active Directory OU or container where the servers are located. Alternatively, a domain administrator can **prestage a cluster computer object (CNO)** in AD DS for the cluster name.
 - **Prestaging CNO (Optional but Recommended):**
 1. In **Active Directory Users and Computers**, create an Organizational Unit (OU) for cluster objects (recommended best practice).
 2. Right-click the OU, select **New > Computer**. Enter the **name you want to assign to the cluster** (this will be the WSFC's administrative name, distinct from BigFix masthead and SQL AG listener names).
 3. Right-click the newly created computer account, select **Disable Account**, and confirm.
 4. Grant **Full Control** permissions on this CNO to the domain user account that will be used to create the cluster. Ensure **Advanced Features** is enabled in **View** menu to see **Security** tab.
 5. (If CNO is in an OU) Grant the CNO permissions to create computer objects in the OU by adding the CNO as a principal to the OU's security settings and giving it **Create Computer objects** permission.
- **Network Configuration:** Ensure consistent network links for communications between WSFC nodes and between availability replicas. Dedicated network adapters for Always On availability groups are recommended for best performance.
- **Firewall Rules:** Configure firewalls to allow necessary network traffic for cluster communication and management.

Core Cluster Communication (Between Cluster Nodes)

These rules are fundamental for the cluster to maintain health, manage state, and handle failover operations.

Source	Destination	Port	Port Type	Description
Cluster Nodes	Cluster Nodes	3343	TCP/UDP	Cluster Heartbeat and Join: Used for inter-node health monitoring and for new nodes to join the cluster.
Cluster Nodes	Cluster Nodes	135	TCP	RPC Endpoint Mapper: Initial communication for Remote Procedure Calls (RPC).
Cluster Nodes	Cluster Nodes	49152-65535	TCP	Dynamic RPC: The high-port range used for subsequent RPC communication between nodes for various cluster operations. This range can sometimes be customized.

Cluster Nodes	Cluster Nodes	445	TCP	SMB (Server Message Block): Required for Cluster Shared Volumes (CSV) and for interacting with a File Share Witness.
Cluster Nodes	Cluster Nodes	-	ICMPv4/ICMPv6	ICMP: Allows for basic connectivity testing and is used during node join operations.

Active Directory and DNS Communication

A Failover Cluster is heavily reliant on Active Directory for authentication and DNS for name resolution. These rules should be configured to allow cluster nodes to communicate with domain controllers and DNS servers.

Source	Destination	Port	Port Type	Description
Cluster Nodes	Domain Controllers	53	TCP/UDP	DNS: For resolving hostnames of cluster nodes and other network resources.
Cluster Nodes	Domain Controllers	88	TCP/UDP	Kerberos: For authentication of cluster computer objects.
Cluster Nodes	Domain Controllers	389	TCP/UDP	LDAP: For querying Active Directory for cluster-related objects.
Cluster Nodes	Domain Controllers	464	TCP/UDP	Kerberos Password Change: Used for password management of cluster computer objects.
Cluster Nodes	Domain Controllers	3268/3269	TCP	Global Catalog: For searching the Global Catalog in a multi-domain environment.

Management and Optional Services

These rules facilitate the remote management of the cluster and support optional configurations.

Source	Destination	Port	Port Type	Description
Management Workstation	Cluster Nodes	135, 49152-65535 (Dynamic RPC), 445	TCP	Failover Cluster Manager: Allows for remote administration of the cluster using the Failover Cluster Manager snap-in.
Management Workstation	Cluster Nodes	5985	TCP	Windows Remote Management (WinRM): For remote management using PowerShell.

Important Considerations:

- **Source and Destination IP Addresses:** For enhanced security, it is highly recommended to specify the exact IP addresses or subnets of the cluster nodes, domain controllers, and management workstations in the firewall rules rather than using "Any."
- **Network Profiles:** Ensure that these firewall rules are applied to the correct network profiles (Domain, Private, or Public) that the cluster network interfaces are using.
- **Stateful Firewall:** When using a stateful firewall, it is generally sufficient to create inbound rules, as the firewall will automatically manage the corresponding outbound traffic for established connections.
- **Application-Specific Ports:** If you are running specific applications on your failover cluster, such as SQL Server or Hyper-V, you will need to open additional ports required by those applications. The rules outlined above are for the core functionality of the Windows Server Failover Cluster itself.

2. Install Failover Clustering Feature

- On all 6 servers (3 BigFix Root Servers, 3 SQL Servers), install the "Failover Clustering" Windows Server feature.
 - **PowerShell:** `Install-WindowsFeature -Name Failover-Clustering -IncludeManagementTools`
 - **Server Manager:** Add Roles and Features Wizard.
- It is suggested to restart the computers after installing this feature.

3. Validate Cluster Configuration

- **Strongly recommended** to validate the configuration before creating the cluster to ensure compatibility. Microsoft only supports cluster solutions that pass all validation tests.
- **Using Failover Cluster Manager:**
 1. Open **Failover Cluster Manager**.
 2. Under **Management** node, select **Validate Configuration**.
 3. Add all 6 servers (3 BigFix, 3 SQL) to the validation test.
 4. Choose **Run all tests (recommended)**.
 5. Review the report and correct any warnings or failures.
- **Using PowerShell:** `Test-Cluster -Node Server1, Server2, Server3, Server4, Server5, Server6` (replace with your server names).

4. Create the Windows Server Failover Cluster (WSFC)

- This step creates the foundation for both your BigFix and SQL HA configurations.
- **Using Failover Cluster Manager:**
 - Open **Failover Cluster Manager**, right-click on **Failover Cluster Manager** in the tree, and select **Create Cluster...**
 - On the **Select Servers** page, add all 6 servers (3 BigFix, 3 SQL) that passed validation.
 - On the **Access Point for Administering the Cluster** page, specify the **"Cluster Name"** for the cluster itself.
 - **Important:** This cluster name **must be distinct** from the hostname/IP you'll use for the BigFix masthead and the SQL Server Availability Group Listener. This name will be registered as the Cluster Name Object (CNO) in AD DS.
 - Follow the wizard until completion.
- **Quorum Configuration:** For a cluster with 3 or 4 nodes, a witness is **strongly recommended**. Given you have 6 nodes initially, and the cluster might operate with fewer nodes after failures, a witness provides additional resiliency.
 - Consider a **File Share Witness** or **Cloud Witness**. A Disk Witness is **not supported** if you are using Storage Spaces Direct.

- Dynamic quorum automatically adjusts the number of votes needed for the cluster to stay online.

Phase 2: SQL Server Always On Availability Groups (AGs) Configuration

You will configure SQL Server Always On Availability Groups for your BigFix databases across your 3 SQL Servers. This provides HA for your BigFix databases.

1. SQL Server Prerequisites for Always On AGs

- **SQL Server Edition:** You need **SQL Server Enterprise edition** on all 3 SQL Servers.
- **WSFC Node:** Each SQL Server instance must be a node in the WSFC you just created.
- **Version and Collation:** All server instances must be running the **same version of SQL Server** and use the **same SQL Server collation**.
- **Enable Always On Feature:** The "Always On Availability Groups" feature must be enabled on each SQL Server instance via SQL Server Configuration Manager.
 1. Open **SQL Server Configuration Manager > SQL Server Services**.
 2. Right-click the **SQL Server** entry, then **Properties**.
 3. In the **Always On Availability Groups** tab, ensure **Enable Always On Availability Groups** is checked.
- **Service Account:** The SQL Server service should run under an **Active Directory domain user account**. Use the **same domain user account** for all SQL Server instances in the AG.
 1. In **SQL Server Configuration Manager > SQL Server Services**, go to the **Log On** tab for the **SQL Server** service.
 2. Configure it to run under the chosen Active Directory user account.
 3. Restart the SQL Server instance if prompted.
- **Database Mirroring Endpoint:** Each server instance requires a database mirroring endpoint. The New Availability Group Wizard can create this automatically if the service account is a domain user; otherwise, you must create it manually using certificates.
- **TDE Protected Databases (If applicable):** If using Transparent Data Encryption (TDE), the certificate or asymmetric key must be the same on every server instance hosting an availability replica.

Essential Firewall Configurations for Microsoft SQL Server Always On

The following table outlines the essential firewall rules required for a successful and secure SQL Server Always On implementation. These rules encompass communication for the SQL Server Database Engine, the Windows Server Failover Cluster (WSFC) services, and the database mirroring endpoints.

Source	Destination	Port	Port Type	Description
All SQL Server Nodes in the Cluster	All SQL Server Nodes in the Cluster	135	TCP	RPC Endpoint Mapper: Used by the failover cluster for remote procedure calls and client-to-node and node-to-node communication.
All SQL Server Nodes in the Cluster	All SQL Server Nodes in the Cluster	445	TCP	SMB: Required for the File Share Witness, if one is configured for the cluster quorum.
Application Servers / Clients	Availability Group Listener IP	1433 (Default)	TCP	SQL Server Database Engine: The default port for client applications to connect to the SQL Server instance or the Availability Group Listener. If a named instance or a different port is used, that port should be opened instead.

All SQL Server Nodes in the Cluster	All SQL Server Nodes in the Cluster	1434	UDP	SQL Server Browser Service: Allows clients to discover SQL Server instances on the network, especially when using named instances or non-default ports.
All SQL Server Nodes in the Cluster	All SQL Server Nodes in the Cluster	3343	TCP/UDP	Windows Server Failover Cluster: Used for cluster node communication, including heartbeats and cluster database updates.
All SQL Server Nodes in the Cluster	All SQL Server Nodes in the Cluster	5022 (Default)	TCP	Database Mirroring Endpoint: The default port for communication between SQL Server instances for database mirroring and Always On data synchronization. This port is configurable.

All SQL Server Nodes in the Cluster	All SQL Server Nodes in the Cluster	49152-65535 (Dynamic)	TCP	RPC Dynamic Ports: The failover cluster uses a range of dynamic ports for various communications. This range can be restricted to a smaller, more manageable set of ports if required for security policies.
-------------------------------------	-------------------------------------	-----------------------	-----	---

Important Considerations:

- **Source and Destination:** For enhanced security, it is highly recommended to restrict the source and destination IP addresses in your firewall rules to only the servers that are part of the Always On Availability Group and any necessary application servers. Using "Any" for the source or destination should be avoided in a production environment.
- **Default vs. Configured Ports:** The port numbers listed as "(Default)" can be changed. If you have configured your SQL Server instances or mirroring endpoints to use different ports, you must adjust your firewall rules accordingly.
- **Windows Firewall:** These rules can be implemented using the "Windows Defender Firewall with Advanced Security" snap-in. It is crucial to apply these rules to the correct network profile (Domain, Private, or Public) that your servers are using for inter-node communication.
- **Network Segmentation:** In a multi-subnet failover cluster, ensure that routing is correctly configured and that firewalls between the subnets allow the necessary traffic as detailed in the table.
- **Domain Controllers:** While not directly part of the SQL Server rules, ensure that all cluster nodes have the necessary access to domain controllers for authentication purposes. This typically includes ports for Kerberos (88/TCP & UDP) and LDAP (389/TCP & UDP).

By carefully implementing these firewall rules, you can help ensure the high availability and reliability of your Microsoft SQL Server Always On environment. Always consult your organization's security policies before making changes to firewall configurations.

2. BigFix Database Requirements for Always On AGs

- **User Databases:** BigFix databases (BFEnterprise, BESReporting) must be user databases (system databases are not supported).
- **Read-Write & Multi-User:** Databases must be read-write and multi-user.
- **No AUTO_CLOSE:** Databases must not use AUTO_CLOSE.
- **Full Recovery Model:** Databases must use the **full recovery model**.
 - Verify: `SELECT name, recovery_model_desc FROM sys.databases WHERE name = 'BFEnterprise' OR name = 'BESReporting'; GO.`
 - Set: `ALTER DATABASE BFEnterprise SET RECOVERY FULL; GO ALTER DATABASE BESReporting SET RECOVERY FULL; GO.`
- **Full Database Backup:** You **must perform at least one full database backup** of the BigFix databases before adding them to an AG.
 - Example: `BACKUP DATABASE BFEnterprise TO DISK = 'C:\Path\To\ExistingFolder\BFEnterprise.bak' WITH FORMAT, INIT; GO.`
- **Database User Privileges:** The database user (e.g., the BigFix Server's service account) must exist on all SQL Server instances with `sysadmin` privileges. If using SQL Server authentication, the login user should have the same name, password, and Security Identifier (SID) on all instances. **after the installation or upgrading of the databases, the privilege can be lowered to `db_owner`**

3. Creating the Availability Group (AG)

- Connect to the SQL Server instance hosting the BigFix database (this will be the primary replica initially).
- **Using SQL Server Management Studio (SSMS):**
 1. In Object Explorer, expand `Always On High Availability` node, right-click `Availability Groups` node, and select `New Availability Group Wizard`.
 2. On the `Specify Availability Group Options` page, enter a unique **"Availability group name"** (e.g., `BigFixAG`). This name is distinct from the WSFC name and the AG Listener name.
 3. Choose **"WSFC"** as the **"Cluster type"**.
 4. On the `Select Databases` page, select your BigFix databases (BFEnterprise, and optionally BESReporting).
 - **Note:** If BigFix Server and Web Reports are on the same machine, putting both databases in the same AG can prevent issues during upgrades.
 5. On the `Specify Replicas` page:
 - In the `Replicas` tab, add all 3 SQL Server instances as availability replicas.

- Set the **Availability Mode** for the primary replica and at least one secondary replica to "**Synchronous commit**" for High Availability. set the mode to "**Asynchronous commit**" for DR
No need to select Automatic Failover (Up to 5) at the moment
SQL Server 2019 supports up to five synchronous-commit replicas.

Recommendation:

Scenario	Recommendation
Highest Availability Needed for a Stable Environment	Enable Automatic Failover. If your network is stable and your application performance is acceptable with synchronous commit, this setting provides the best protection against unplanned downtime. Start with one automatic failover target.
Performance-Sensitive Workload or Unstable Network	Use Synchronous-Commit with Manual Failover. This still guarantees zero data loss but gives you full control, preventing unwanted failovers due to transient network issues. It also avoids the performance hit of synchronous commit if you opt for asynchronous mode.
Disaster Recovery (DR) Site in a Different Location	Use Asynchronous-Commit with Manual Failover. For replicas over a long distance, synchronous commit is usually impractical. This DR setup allows for some data loss (the transactions not yet sent to the secondary) but provides a manual recovery option in a true disaster.

- You can configure **Readable Secondary** to allow read-only access to secondary databases for performance improvement.
6. On the **Select Initial Data Synchronization** page, choose a method:
- **Automatic seeding:** Preferred if data and log file paths are identical on all SQL Server instances.
 - **Full database and log backup:** Select this if automatic seeding is not possible. You'll need to specify a **shared network location** accessible by all replicas for backups.
 - **Join only:** If you manually prepared secondary databases already.

7. On the **Validation** page, ensure all entries show "Success".
8. Complete the wizard. Verify the AG state on the dashboard (right-click AG, **Show Dashboard**) should be "Healthy" and synchronization states should be "Synchronized".

4. Creating an Availability Group Listener

- It is **strongly recommended to create an Availability Group Listener** for client connectivity.
- The listener needs its **own hostname and IP address**, distinct from the cluster's and BigFix masthead's. It also needs a port.
If you select Static IP - Make sure to exclude the IP from Address Pool on the DHCP
If you select DHCP - Make sure to reserve the IP address after that on the DHCP
- The SQL Server service account needs permissions to create objects in Active Directory to register the listener's hostname/IP.
If the SQL Server service account does not have the permissions - follow the [Pre-stage the VCO procedure](#)
- **Using SSMS:**
 1. You can create the listener during the **New Availability Group Wizard** or after.
 2. If creating after: Right-click on the AG you created, then **Add Listener....**
 3. Enter a free **DNS Listener Name** and **Port**.
 4. Click **Add...** to specify a free IP address and its subnet (same as SQL Server instances).
 5. Complete the procedure.

Pre-stage the VCO procedure

SQL Administrator - Gather and Provide Information

1. **The Proposed Listener Name:** This is the DNS name clients will use to connect (e.g., **BigFix-Listener**).
2. **The Cluster Name Object (CNO):** This is the network name of your Windows Server Failover Cluster itself. You can find this in the "Failover Cluster Manager" in the main cluster view. Let's say your cluster is named **BF-Cluster**.
3. **The Proposed IP Address:** The static IP you plan to assign to the listener (e.g., **10.10.1.100**).

Step 2: Active Directory Administrator - Pre-stage the VCO

- **Create the Virtual Computer Object (VCO):**
 1. Navigate to the Organizational Unit (OU) where your server computer accounts reside (or the designated OU for cluster objects).
 2. Right-click within the OU and select **New > Computer**.
 3. In the "Computer name" field, enter the exact **Listener Name** you decided on (e.g., **BigFix-Listener**).
 4. Click **OK**.
- **Disable the New Computer Object:**
 1. Immediately find the newly created computer object (**BigFix-Listener**).
 2. Right-click it and select **Disable Account**.
 3. Confirm by clicking **Yes**. This is crucial because this account doesn't represent a physical computer and should not be enabled directly.
- **Grant Permissions to the Cluster Name Object (CNO):**
 1. Right-click the newly created, disabled computer object (**BigFix-Listener**) and select **Properties**.
 2. Go to the **Security** tab. (If you don't see the Security tab, go to **View** in the main console and enable **Advanced Features**).
 3. Click **Add...**
 4. In the object picker window, change the "Object Types..." to include **Computers**.
 5. Enter the name of your **Cluster Name Object (CNO)** (e.g., **BF-Cluster**) and click **Check Names**. It should resolve. Click **OK**.
 6. With the CNO selected in the list, grant it the **Full Control** permission. Granting "Full Control" ensures the cluster can manage the VCO, update its DNS records, and bring it online and offline as needed during a failover.
 7. Click **Apply**, then **OK**.

5. Update BigFix ODBC Data Sources

- After creating the AG and its listener, you must **update the BigFix ODBC data sources** on all BigFix Root Servers to connect to the AG listener's hostname (or IP) and port, rather than directly to a SQL Server instance.
- There are four or six ODBC data sources to update, depending on Web Reports installation.
- If BigFix was configured to use a local database initially, open **Registry Editor** to **HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix\Enterprise Server** and set **UseRemoteDB** (REG_SZ) to **1**.
- Update connection strings if not using DSN.

Phase 3: BigFix Root Server High Availability Configuration

This phase involves configuring your 3 BigFix Root Servers to leverage the WSFC and the shared volume. The provided guide primarily covers two BigFix servers, but the principles extend to three.

1. BigFix Masthead Hostname/IP Considerations

- The hostname (or IP) in your **BigFix masthead file is a critical resource**. It **cannot belong to any specific computer**. Instead, it will be associated with a "cluster role" you'll create in the WSFC.
- **Freeing the Masthead Hostname:** If the masthead hostname (e.g., **bfserver.domain.test**) is currently in use by a computer or DNS entry, you **must free it**.
 - Rename any computer using that FQDN.
 - Ensure no DNS entries map the masthead hostname to a specific computer.
 - Verify no leftover entries in Active Directory reference the old computer name.
- The masthead hostname/IP **must remain distinct** from the WSFC's administrative hostname/IP and the SQL Server Availability Group Listener's hostname/IP.

2. Create a Cluster Role for BigFix with Network Resources

- This cluster role will handle redirecting BigFix network traffic to the currently active BigFix Root Server.
- **Using Failover Cluster Manager:**
 1. Expand the cluster name in the tree, right-click on **Roles**, and click **Create Empty Role**.
 2. Rename the role (e.g., **BigFixRootServerRole**).
 3. Optionally, assign **Preferred Owner Nodes** (e.g., set your BigFix Root Servers as preferred owners).
 4. Ensure the **Status** of the new role is **Stopped**.

5. Assign the **BigFix masthead hostname (or IP)** to this cluster role. Right-click on the role, select **Add Resource > Client Access Point**. Specify the masthead hostname in **Name** and its IP in **Address**.

3. Redirect Traffic to the Installation Node (Temporary for Installation)

- Before installing BigFix, you must redirect the traffic meant for the masthead hostname/IP to the cluster node where you intend to install the first BigFix Root Server.
- **Using Failover Cluster Manager:**
 1. In the **Roles** view, right-click the BigFix cluster role you created.
 2. Select **Move > Select Node...**
 3. Select the cluster node where you will install the BigFix Server, then confirm.
 4. Ensure the cluster resources (hostname and IP) for this role are **Online** by starting the whole role or bringing its network resources online.

4. Install the First BigFix Root Server

First BigFix Root Server should be installed on a Remote SQL Server and not SQL Always on Server - Don't start services

- **Database Access Authentication:**
 - **Windows Authentication:** Run the installer as an Active Directory administrator that also has **sysadmin** privileges on the SQL Server instances. This AD user should also be the one running the BigFix Server services.
 - **SQL Server Authentication:** If using this, you may need to re-create the login user to ensure it has the same name, password, and SID on all SQL Server instances in the AG.
- After that make sure to configure the AG and Listener and Update the ODBC configuration

5. Move BigFix Data Files to Shared Storage (Cluster Shared Volume)

Review Options for [Shared Storage](#)

- A **shared storage solution** (preferably a **Cluster Shared Volume (CSV)**) is needed for BigFix Server data files. The BigFix Server program files (executables) should remain local.
- **Stop BigFix Services:** Ensure any programs accessing BigFix databases or data files are stopped.
- **On the first BigFix Root Server (where you just installed):**
 1. Run the `BESAdmin.exe` command with the `/setshareablesserverdatapath` and `/copydata` options to move the data files to the shared folder on the CSV.
 2. Example: `.\BESAdmin.exe /setshareablesserverdatapath /topath=<shared storage dir> /copydata /sitePvkFile=<path+license.pvk> /sitePassword=<password>.`
 3. Verify the file copy is successful by checking `BESAdminDebugOut.txt` log. You can delete local copies of data but keep log folders.

Folders to be copied:

None

`[BigFix Server folder]\Encryption Keys`

`[BigFix Server folder]\FillDBData\fastbufferdir"`

`[BigFix Server folder]\FillDBData\BufferDir"`

`[BigFix Server folder]\GatherDBData\gather"`

`[BigFix Server folder]\Mirror Server"`

`[BigFix Server folder]\TrustedDownloadCerts"`

`[BigFix Server folder]\wwwrootbes"`

`[BigFix Server folder]\UploadManagerData\BufferDir`

`[BigFix Server folder]\UploadManagerData\TransientDir`

- **On the other two BigFix Root Servers (future installations):**

1. After the Installation, Run the same **BESAdmin.exe** command **without the /copydata option**. This will configure them to use the shared path without copying data again.

6. Install Additional BigFix Root Servers (Second and Third Servers)

- Make [Server Backup](#) Procedure on the First BigFix Root Server
- Make [Server Recovery](#) Procedure on all of the Additional BigFix Root Servers
 1. **Redirect Traffic:** Before each installation, make the BigFix cluster role redirect traffic to the current BigFix Root Server being installed.
 2. **During BigFix Server Installer:**
 - At **Select Database Replication** dialog, choose **Single or Master Database**.
 - At **Select Database** dialog, choose **Use Remote Database**.
 - At **Database Server** dialog, specify the **hostname (or IP) and port of the SQL Server Availability Group listener**.
 - At **Choose Destination Location** dialog, specify a folder **local to the computer**.
 - At the last dialog, **do not start the BigFix Server services**.

Windows Native Shared Storage Options:

These options are configured directly within the Windows operating system, either on physical servers or within virtual machines.

1. SMB/CIFS File Shares (Network Shares):

- **Concept:** Simple network folder sharing using the SMB protocol.
- **Use Case:** General file sharing, user home directories, basic application data.
- **Pros:** Easy to set up, widely compatible.
- **Cons:** Not high-performance for demanding applications, lacks inherent high availability (requires WSFC for that).

2. Windows Server Failover Clustering (WSFC) with Shared Storage:

- **Concept:** Multiple Windows servers (nodes) work together to provide high availability for applications and services by sharing access to the same storage.
- **Underlying Storage:** Can use Fibre Channel SAN, iSCSI SAN, or Storage Spaces Direct.
- **Key Feature:** Cluster Shared Volumes (CSVs) enable simultaneous access to volumes by multiple nodes.
- **Use Case:** Highly available file servers (Scale-Out File Servers), clustered databases (SQL Server), Hyper-V VMs.
- **Pros:** High availability, fault tolerance, minimizes downtime.
- **Cons:** More complex setup than simple shares.

3. Storage Spaces Direct (S2D):

- **Concept:** Software-defined storage that pools local drives from multiple servers in a WSFC into a highly available shared storage pool.
- **Use Case:** Hyper-converged infrastructure (compute and storage on same servers), large-scale file shares, virtual machine storage.
- **Pros:** Cost-effective (commodity hardware), scalable, resilient, high performance.
- **Cons:** Requires Windows Server Datacenter, can be resource-intensive.

4. Storage Spaces (for Single Servers):

- **Concept:** Pools physical drives within a *single* Windows server to create virtual disks with various resiliency (mirror, parity).
- **Use Case:** Data protection for individual servers, flexible storage expansion on a single machine.
- **Pros:** Protects against drive failure, easy to expand.

- **Cons:** Not a network shared storage solution, no protection against server failure.

VMware vSphere Shared Storage Options (for Windows VMs):

These methods provision shared storage to the ESXi hypervisors, which then present virtual disks to the Windows virtual machines running on them. Some options also allow *in-guest* shared storage for Windows clustering.

1. External Shared Storage (Traditional SAN/NAS):

- **Fibre Channel (FC) SAN:** High-speed, block-level storage network. ESXi uses **VMFS** (Virtual Machine File System) for datastores.
- **iSCSI SAN:** Block-level storage over Ethernet. ESXi also uses **VMFS**.
- **NFS (Network File System):** File-level storage from NAS devices. ESXi mounts NFS shares directly as datastores.
- **vSphere Virtual Volumes (vVols):** VM-centric storage management, abstracting array details, allowing granular policies.
- **Overall Use Case:** Provides the underlying shared datastores for all VMs, enabling vMotion, HA, DRS. Windows VMs consume virtual disks (VMDKs) from these datastores.
- **Pros:** Robust, proven, high performance (FC/iSCSI).
- **Cons:** Can be expensive (FC), NFS generally less performant for high I/O.

2. Software-Defined Shared Storage (within VMware):

- **VMware vSAN:**
 - **Concept:** Aggregates internal disks of ESXi hosts in a cluster to create a single, distributed shared datastore (hyper-converged).
 - **Use Case:** Highly scalable and flexible infrastructure, suitable for a wide range of virtualized workloads.
 - **Pros:** Simplified management, cost-effective, high performance, built-in HA.
 - **Cons:** Resource-intensive, proper sizing crucial.

3. In-Guest Shared Storage for Windows Server Failover Clustering (WSFC) within VMs:

- **Raw Device Mapping (RDM) - Physical Compatibility Mode:** Presents a raw LUN directly to a Windows VM for WSFC, allowing SCSI-3 Persistent Reservations.

- **Clustered VMDK (vSphere 7.0+):** Allows multiple VMs to share a single VMDK file, supporting SCSI-3 Persistent Reservations for WSFC in a more virtualized way.
- **In-Guest iSCSI Initiator:** Windows VM directly connects to an iSCSI target for shared storage, bypassing ESXi's storage stack.
- **In-Guest SMB File Shares:** Windows VMs use SMB to connect to a file share hosted by another Windows server (physical or virtual).
- **Overall Use Case:** Specific to enabling Windows Server Failover Clustering *inside* virtual machines, where shared disk access is required for the cluster service itself.
- **Pros:** Enables Windows native clustering features in a virtualized environment.
- **Cons:** Can add complexity, some options (RDM) limit vSphere features (e.g., snapshots).

Choosing the right option involves evaluating performance, scalability, high availability needs, budget, and the specific application requirements, both for the hypervisor layer and the Windows guest operating system.

Server Backup

How to back up the BigFix Server on Windows.

Consider backing up to a remote system to allow for higher fault tolerance. Perform the following steps:

1. Stop all the BigFix services, including running plug-ins if any. The main BigFix services should be stopped in the following order.

None

BES WebUI

BES Web Reports Server

BES Client

BES GatherDB

BES FillDB

BES Root Server

2. If you have remote processes that may access the BigFix databases, like a remote BigFix WebUI, stop them as well.
3. Back up the following files and folders used by the BigFix Server and Web Reports:
 - [BigFix Server folder]\BESReportsData. Web Reports data folder.
 - [BigFix Server folder]\BESReportsServer\wwwroot\ReportFiles. Support files for Web Reports custom reports.
4. Back up your license key, license certificate, and masthead file.
The `license.pvk` and `license.crt` files are critical to the security and operation of BigFix. If the private key (`pvk`) files are lost, they cannot be recovered.
The masthead (`afxm`) file is an important file that contains information about the BigFix Server configuration. To back it up, you can copy the [BigFix Server folder]\actionsite.`afxm` file and rename it to `masthead.afxm`.

5. Decrypt and save the encrypted keys.

The encrypted keys are files named like `Encrypted...Key` and, by default, they are located in the `[BigFix Server folder]`. Depending on the version of the BigFix Server and on what BigFix components are installed on your environment, you will have a slightly different set of encrypted keys. The possible key files are:

- `EncryptedAPIServerKey`
- `EncryptedClientCAKey`
- `EncryptedCustomClientCAKey`
- `EncryptedCustomServerTLSKey`
- `EncryptedExplorerCAKey`
- `EncryptedPlatKey`
- `EncryptedServerSigningKey`
- `EncryptedWebUICAKey`

6. Use the **ServerKeyTool.exe** tool and run the steps documented in [this page](#) to decrypt the keys. For example:

None

```
ServerKeyTool.exe /decrypt
```

```
  /dirIn:"C:\Path\To\BigFix Server folder"
```

```
  /dirOut:"C:\Path\To\Backup folder"
```

```
  /sitePvkLocation:"C:\Path\To\license.pvk"
```

```
  /sitePvkPassword:"*****"
```

7. All the existing encrypted keys stored in the input folder are backed up at once, and the files containing the decrypted keys are stored in the specified destination folder with the filename prefix `Decrypted*`.
8. Analyze the BigFix Server database tables named `DBINFO` and `REPLICATION_SERVERS`.

None

```
SELECT * FROM [BFEnterprise].[dbo].[DBINFO];
```

```
SELECT * FROM [BFEnterprise].[dbo].[REPLICATION_SERVERS];
```

9. Record all the rows in those tables for verification purposes.
If DNS aliases are being used instead of the actual hostnames of your BigFix Server computers, the field values of those tables should remain the same after a restore procedure. If they contain hostnames, and the hostnames of your BigFix Server computers change, those fields may need manual modification after the restore. If you want to update the CN on the BigFix internal certificates, see [How to change the Common Name \(CN\) on BigFix internal certificates](#).
10. Any configuration involving registry keys is neither saved nor restored. After completing a restore procedure, you will have to perform those configurations again. You should only edit a BigFix registry key if a documented configuration step explicitly calls for it. However, for verification purposes, you should export the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BigFix`.

Server Recovery

How to restore a BigFix Server on Windows.

1. Ensure that the new BigFix Server computer can be reached on the network using the same URL that is in the [masthead file](#). For example, if you look at the field `X-Fixlet-Site-Gather-URL`, it contains something like `http://192.168.10.32:52311/cgi-bin/bfgather.exe/actionsite` or `http://bigfixserver.company.com:52311/cgi-bin/bfgather.exe/actionsite`.

Note:

To avoid issues when the BigFix clients connect to the BigFix Server before it is fully restored, ensure that it is not available on the network until the recovery is complete.

2. Restore the backed up files and folders, creating the directory structure.
3. Encrypt the key files you decrypted during the backup procedure.
Use the **ServerKeyTool.exe** tool and run the steps documented in [this page](#) to encrypt the keys. For example:

None

```
ServerKeyTool.exe /encrypt  
  
/dirIn:"C:\Path\To\Backup folder"  
  
/dirOut:"C:\Path\To\BigFix Server folder"  
  
/sitePvkLocation:"C:\Path\To\license.pvk"  
  
/sitePvkPassword:"*****"
```

4. All the decrypted keys are encrypted at once, and the files containing the encrypted keys are stored in the specified destination folder with the filename prefix **Encrypted***.
5. Go to the [BigFix Download Center Platform Release](#) page, find the same patch level of the original installation and go to the corresponding download page. Then download the "Installer Generator" setup file.
6. Run the BigFix Installation Generator on the computer where you want to restore the BigFix Server.
7. When you reach the "Setup Type" dialog, select the option "I want to install with an existing masthead" and choose the Server masthead you previously backed up.
Do not re-use the folders of a previous Installation Generator installation, they likely contain an outdated masthead file.
8. Run the BigFix Installation Guide and, from there, start the BigFix Server installer.

9. In the "License credentials" dialog, make sure that the selected "Masthead file" is the one in the Installation Generator folder or select the masthead you backed up.
10. In the "Select Database Replication" dialog
 - select the "Single or Master Database" option
11. In the "BigFix Master Operator Credentials" dialog, in the "Username" field, specify the same BigFix master operator that you used for the installation you backed up.

Note: If, later on during the installation, you get a pop-up message saying that "The user name *MyUserName* already exists in the database", you can ignore it.

12. In the "Choose Destination Location" dialog, make sure that "Destination Folder" is the same folder where you restored the files you backed up.
13. In the "Server Properties" dialog, make sure the "WWW Path" leads to the folder where you restored the BigFix Server `wwwroot` folder. Make sure the "URL" matches the one in your masthead.
14. In the "Web Reports Properties" dialog, make sure the "WWW Path" leads to the folder where you restored the Web Reports `wwwroot` folder. Make sure the "Port" is the same as before.
15. Continue the installation procedure until completion.
16. If you customized your previous BigFix installation by performing additional configuration procedures, you will have to perform them again for this installation. For example, remember to re-apply your custom HTTPS configuration and to re-configure your Web Reports email server settings, if you had them in your previous installation. Only edit the BigFix registry keys directly if a documented configuration step calls for it. Moreover, direct modifications of those keys are better performed when the BigFix services are stopped. Do not import the BigFix registry settings from your previous installation. You can use the registry keys you backed up for verification purposes.
17. Reinstall the BigFix components and plugins that were installed on the original BigFix Server by re-deploying the appropriate Fixlets. For example, reinstall the BigFix WebUI, the BES Server Plugin Service, and any plugins you had.

7. Completing the BigFix Server Cluster Role (Adding Services)

- After installing all 3 BigFix Root Servers and configuring shared storage paths on all of them, you need to add the BigFix services to the BigFix cluster role you created.
- **Stop All BigFix Services:** Ensure all BigFix services on all BigFix Root Servers are stopped.
- **Using Failover Cluster Manager:**
 1. In the **Roles** view, right-click the BigFix cluster role.
 2. Select **Add Resource > Generic Service**.
 3. Select the **"BES Root Server"** service, and complete the procedure.
 4. Repeat for **"BES FileDB"** service.
 5. Repeat for **"BES GatherDB"** service.
 6. (Optional) If Web Reports is installed on the same servers, you can also add the **"BES Web Reports Server"** service to this same cluster role.
 - **Note:** The WebUI service, if installed, should not be mapped to this cluster role.

8. Managing Clustered BigFix Services

- **Crucial:** Once a BigFix service is part of a cluster role, you **must manage (start/stop) it only through the cluster manager**.
- Never start or stop BigFix services directly via Windows Services Manager.
- A BigFix service that is part of a cluster role should **never run on two computers simultaneously** to prevent data corruption.
- To stop a clustered service, stop its entire role or take it offline via Failover Cluster Manager.

9. Configuring the BigFix Client on BigFix Root Server Computers

- On **each computer where you installed the BigFix Root Server**, add the following client setting:
 - **Setting Name:** **_BESClient_RelaySelect_UseMastheadRootServer**
 - **Value:** **1**
- Start or restart the BigFix Client after applying this setting. This ensures the client on the Root Server uses the masthead hostname/IP to connect to the currently active BigFix Server, rather than assuming a local connection.

10. Upgrading a Clustered BigFix Environment

- Before upgrading, ensure all programs accessing BigFix databases or data files are stopped.
- When upgrading BigFix Server, Web Reports, or WebUI components in a cluster, you need to upgrade all components on all computers where they are installed. Since BigFix Platform version 11.0.4, you can choose not to start components at the end of installation/upgrade.

This comprehensive setup provides a robust high availability and disaster recovery solution for your BigFix environment.

Sources

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/prereqs-restrictions-recommendations-always-on-availability?view=sql-server-ver17>

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/use-the-availability-group-wizard-sql-server-management-studio?view=sql-server-ver17>

https://help.hcl-software.com/bigfix/11.0/platform/Platform/Installation/c_configuring_ha_dr_windows.html#c_configuring_ha_dr_windows_section_1

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/creation-and-configuration-of-availability-groups-sql-server?view=sql-server-ver17>

<https://learn.microsoft.com/en-us/windows-server/failover-clustering/create-failover-cluster?pivot=s=windows-admin-center#verify-the-prerequisites>

<https://learn.microsoft.com/en-us/windows-server/failover-clustering/prestage-cluster-adds>

<https://learn.microsoft.com/en-us/windows-server/storage/storage-spaces/quorum>

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/overview-of-always-on-availability-groups-sql-server?view=sql-server-ver17>

<https://4sysops.com/archives/vmware-vsphere-7-clustered-vmkd/>