### **Windows Failover Cluster, SQL Server Failover Cluster & SQL Server Always On in Detail**

High availability (HA) and disaster recovery (DR) are critical for businesses that rely on SQL Server to run mission-critical applications. Several solutions in SQL Server address these needs, with **Windows Failover Clustering**, **SQL Server Failover Clustering**, and **SQL Server Always On** offering distinct methods to achieve high availability.

In this article, we'll dive deep into each of these technologies, their key features, differences, and use cases.

### **1. Windows Failover Clustering (WSFC)**

**Windows Server Failover Clustering (WSFC)** is a feature in Microsoft Windows Server that enables multiple servers (called nodes) to work together as a single system. The main goal is to ensure high availability for critical applications, like SQL Server, by providing automatic failover in case one of the nodes fails.

#### **Key Concepts of WSFC:**

* **Cluster Nodes**: A failover cluster consists of two or more server nodes (computers), connected by a shared storage system (SAN or NAS). If one node fails, another node takes over the workload.
* **Shared Storage**: The cluster nodes share access to the same storage system that holds the application data (e.g., SQL Server database files). This shared storage ensures that the application can access the same data on any node.
* **Failover**: When a node in the cluster fails, the services running on that node automatically move to another node without user intervention. This process is called failover.
* **Quorum**: The quorum defines how many nodes need to be active in order for the cluster to remain operational. A quorum prevents "split-brain" scenarios where two nodes independently think they are the primary.

#### **How WSFC Works:**

1. **Primary (Active) Node**: The active node runs the application (e.g., SQL Server). Clients connect to the virtual name (DNS alias) of the cluster, and the active node processes the queries.
2. **Secondary (Passive) Node**: The passive node does not run the application but stands by as a backup. If the primary node fails, the passive node becomes active and takes over the application without interruption to service.
3. **Shared Storage**: Both nodes have access to shared disk arrays (SAN/NAS), which contain the SQL Server databases and transaction logs.
4. **Failover**: When a failure occurs on the active node, the service (SQL Server) and associated resources fail over to the passive node.

#### **Benefits:**

* Provides **high availability** for SQL Server instances.
* **Automatic failover** ensures minimal downtime.
* Can be used for a **single SQL Server instance** running on a cluster of nodes.

#### **Limitations:**

* **Single instance clustering**: It only supports a single SQL Server instance per cluster.
* **Shared storage**: Requires a shared storage system, which could become a bottleneck.
* Not as flexible for **read scaling** or geographical distribution as SQL Server Always On.

### **2. SQL Server Failover Cluster Instance (FCI)**

**SQL Server Failover Cluster Instance (FCI)** is a high-availability solution that relies on **Windows Server Failover Clustering (WSFC)** for its core functionality. FCIs are essentially SQL Server instances that run on a failover cluster. FCIs provide high availability at the SQL Server instance level and allow for **automatic failover** of SQL Server services from one node to another.

#### **Key Concepts of SQL Server Failover Cluster Instances (FCI):**

* **SQL Server Instance Clustering**: The SQL Server instance is the resource that is clustered, not the database. This means that the entire SQL Server instance, including all databases within it, will failover to another node in case of a failure.
* **Shared Storage**: Like WSFC, FCI relies on shared storage where the SQL Server databases reside. This shared storage is accessible by all nodes in the cluster.
* **Automatic Failover**: If the active node goes down, the FCI will automatically failover to the passive node, ensuring that the SQL Server instance remains available.
* **Clustered Network Name**: A **virtual network name** (DNS alias) is used for the SQL Server instance. Clients connect to the instance using the virtual network name, and the failover process ensures that the virtual name always points to the active node.

#### **How SQL Server FCI Works:**

1. **Shared Storage**: The SQL Server databases are stored on a shared disk array that can be accessed by all nodes in the cluster.
2. **Active and Passive Nodes**: One node is active, running the SQL Server instance. If the active node fails, the passive node takes over by accessing the shared storage and running the SQL Server instance.
3. **Failover**: During failover, the virtual network name (SQL Server instance name) moves to the active node, and clients can still connect without interruption.

#### **Benefits:**

* **Automatic failover** ensures continuous availability of the SQL Server instance.
* Ensures high availability for the entire **SQL Server instance**, including all databases and services.
* Uses **shared storage**, ensuring the databases are always available to the active node.

#### **Limitations:**

* **Single instance per cluster**: Only one SQL Server instance can run on the cluster at a time.
* **Shared storage requirement**: Requires access to a shared disk array, which could introduce a single point of failure for the storage.
* **Cannot scale for read workloads**: Unlike Always On Availability Groups, FCIs do not provide a solution for scaling read workloads.

### **3. SQL Server Always On**

**SQL Server Always On** is a high-availability and disaster recovery solution introduced in SQL Server 2012. Unlike SQL Server Failover Clustering, Always On provides more flexibility by supporting multiple high-availability options, such as **Always On Availability Groups (AGs)** and **Always On Failover Cluster Instances (FCIs)**.

Always On provides a more flexible and feature-rich solution for high availability and disaster recovery than traditional failover clustering.

#### **Key Components of SQL Server Always On:**

1. **Always On Availability Groups (AGs)**:
   * **Availability Groups** allow you to group multiple databases together, making them failover as a single unit. Each availability group has one primary replica (active) and one or more secondary replicas (passive).
   * **Synchronous Commit Mode**: In this mode, the transaction is written to both the primary and secondary replicas, ensuring no data loss.
   * **Asynchronous Commit Mode**: In this mode, data is written to the primary replica and then asynchronously replicated to secondary replicas. This is used for geographically distributed replicas or high-latency environments.
   * **Automatic Failover**: If a failure occurs on the primary replica, one of the secondary replicas can automatically take over as the new primary.
   * **Read-Only Secondary Replicas**: Secondary replicas can be configured for **read-only** workloads (e.g., reporting), offloading read traffic from the primary replica.
2. **Always On Failover Cluster Instances (FCIs)**:
   * **FCIs** are similar to traditional SQL Server Failover Cluster Instances but are integrated into the Always On framework. They provide failover clustering at the SQL Server instance level with the added benefits of Always On features.
   * FCI supports **automatic failover** at the instance level but requires shared storage for database files.

#### 

#### 

#### 

#### **How SQL Server Always On Works:**

1. **Always On Availability Groups (AGs)**:
   * The primary replica handles all read and write operations, while the secondary replicas maintain copies of the database. If the primary replica fails, one of the secondary replicas can automatically take over as the primary.
   * Availability groups support **automatic failover** for synchronous replicas and **manual failover** for asynchronous replicas.
   * **Client connectivity**: Clients connect to the availability group through a virtual network name, which automatically directs the traffic to the primary replica.
2. **Always On Failover Cluster Instances (FCIs)**:
   * Like traditional SQL Server Failover Clustering, FCIs in Always On rely on **shared storage** and provide **automatic failover** of the entire SQL Server instance.
   * The difference is that Always On FCIs are integrated with **Windows Server Failover Clustering (WSFC)** to offer better high availability.

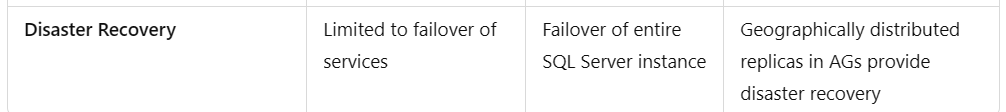
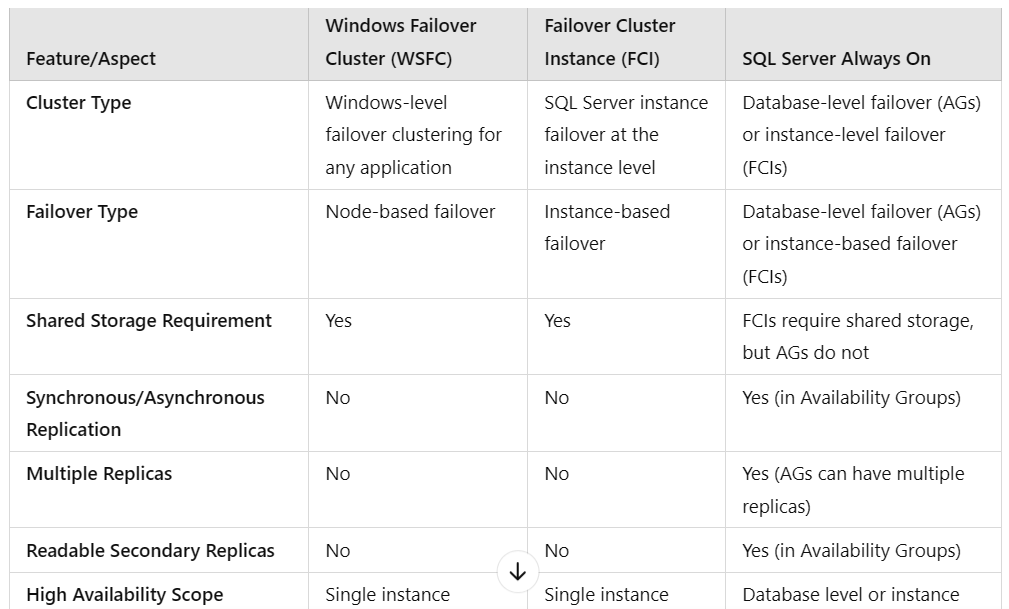
#### **Benefits of SQL Server Always On:**

* **Multiple Availability Options**: Always On provides both **Always On Availability Groups** for database-level failover and **Always On Failover Cluster Instances** for instance-level failover.
* **Automatic and Manual Failover**: Always On supports automatic failover in Availability Groups and also allows manual failover.
* **Scalability**: Always On Availability Groups allow **readable secondary replicas**, providing a way to offload read-only traffic and scale out read workloads.
* **Disaster Recovery**: Always On AGs can replicate data across geographically distributed locations, improving disaster recovery.

#### **Limitations:**

* **Requires Enterprise Edition**: Always On Availability Groups require SQL Server Enterprise edition, which could be costlier than Standard edition or other clustering solutions.
* **Complex Setup**: The setup and configuration of Always On AGs can be more complex, especially when dealing with synchronous and asynchronous replication modes, and requires good knowledge of both SQL Server and Windows Server Failover Clustering.

### **Key Diff B\w Windows Failover, SQL Server Failover Cluster & SQL Server Always On**



### **Summary:**

* **Windows Failover Clustering (WSFC)** provides high availability at the node level, ensuring that services like SQL Server continue running on a backup node if the primary node fails.
* **SQL Server Failover Cluster Instances (FCI)** extend the capabilities of WSFC to SQL Server, providing high availability at the SQL Server instance level, with automatic failover to a backup node.
* **SQL Server Always On** offers more flexibility and advanced features like Always On Availability Groups (for database-level failover) and Always On FCIs (for instance-level failover), allowing multiple replicas and read-only secondary databases for better scalability and disaster recovery.

Each solution has its specific use case, and the choice of which to use depends on the business requirements for **high availability**, **scalability**, and **disaster recovery**.