Windows Clustered Service Setup & Installation Guide

This document provides a step-by-step guide to install a Windows Service in a clustered environment where the service executable resides on a shared Cluster Disk (e.g., E:) and is managed by Windows Failover Clustering.

1. Environment Overview

- Service Type: Custom Windows Service
- Cluster Disk: E:\ (Shared between nodes)
- Cluster Nodes: NodeA, NodeB
- Cluster Manager: Windows Failover Cluster Manager
- Service Account: LocalSystem
- Startup Type: Manual (Cluster-managed)

2. Prerequisites

- Service binary and dependencies must be placed in a directory on Cluster Disk E:\ (e.g., E:\MyService\MyService.exe)
- Both nodes must have access to the same path once the disk is moved
- Windows Failover Clustering role already created

3. Service Registration Script (PowerShell)

This script should be run on each cluster node only after moving ownership of Cluster Disk E: to that node.

PowerShell Script:

```
$serviceName = "MyService"

$displayName = "My Clustered Service"

$exePath = "E:\MyService\MyService.exe"

$serviceDescription = "Custom clustered Windows Service"


if (Get-Service -Name $serviceName -ErrorAction SilentlyContinue) {

Write-Output "Service '$serviceName' already exists. Skipping registration."

} else {

sc.exe create $serviceName binPath= ""$exePath"" start= demand DisplayName= ""$displayName""

sc.exe description $serviceName "$serviceDescription"

Write-Output "Service '$serviceName' created successfully."

}
```

Note: Do NOT set recovery settings via sc failure, as the cluster manages recovery.

4. Cluster Role Configuration (Post Service Registration)

- Open Failover Cluster Manager

- Go to Roles Right-click the appropriate role Properties

- Under General tab:

- Verify name and assigned resources (Service + Disk)

- Under Resources:

- Add the new service (if not added automatically)

- Under Advanced Policies:

- Ensure both NodeA and NodeB are listed as Possible Owners

- Set Maximum Failures and Failover Period as required

- Set Startup Type of the service to Manual (important)

5. Azure DevOps Deployment Integration

Objective:

Deploy the service binary to Cluster Disk E:\ using Azure DevOps without impacting the overall cluster or other clustered services.

Deployment Steps:

1. Stop the specific clustered service:

Stop-ClusterResource -Name "MyService"

2. Deploy updated binaries to E:\MyService

3. Start the specific clustered service:

Start-ClusterResource -Name "MyService"

Recommendations:

- Only stop the specific resource during deployment

- Ensure Azure DevOps agent runs on the node owning E:

6. Key Best Practices

- Avoid using sc failure: Cluster manages service restart/failover.

- Run service as LocalSystem

- Startup type should be Manual

- Use cluster-safe PowerShell commands

7. Validation

- Move cluster disk to Node B Register service Validate

- Move cluster role to Node B Ensure service starts

- Test failover by shutting down Node A

- Use commands to verify:

Get-ClusterResource -Name "MyService"

Stop-ClusterResource -Name "MyService"

Start-ClusterResource -Name "MyService"

8. Cluster Monitoring and Identification

- Get Cluster Name:

Get-Cluster | Select-Object Name

- Find Active Node for Service:

Get-ClusterResource -Name "MyService" | Get-ClusterOwnerNode

- List All Resources and Their Owners:

Get-ClusterResource | Select-Object Name, OwnerNode, State

- Get Active Node for Role:

Get-ClusterGroup | Select-Object Name, OwnerNode, State

9. Troubleshooting Tips

| Issue | Resolution |

|---|---|

| Service fails to start | Check E: availability, dependencies |

| Cluster role wont failover | Check if Node B is set as possible owner |

| Service restarts endlessly | Confirm no sc failure recovery is set |

| Other services interrupted | Ensure only specific resource is stopped |