Introduction to Windows Fabric

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:34 PM PDT

Contents

Introduction

Windows Fabric is a distributed systems platform that makes it easy to build scalable, reliable, and easily-managed services and applications for both cloud and on-premises through a set of public APIs. It is the platform on which SQL Azure is built for the Azure and Lync Server for on premise/O365 and can be used to make applications run reliably at cloud scale and be patched and managed without downtime.

Windows Fabric composes with the Windows Azure Service model to provide key capabilities of reliably running applications composed from both stateless and stateful services running at a higher density than is possible only with VMs, service-level load balancing to optimize resource usage, name resolution to dynamically determine current location of services, and full application lifecycle management for provisioning, deployment, monitoring, and live patching without downtime. Higher level application services such a SQL Azure, Service Bus and Cache Services can be rapidly built on Window Fabric.

Windows Fabric is a distributed systems platform that makes it easy to build and manage scalable, reliable applications for both the cloud and on premise.

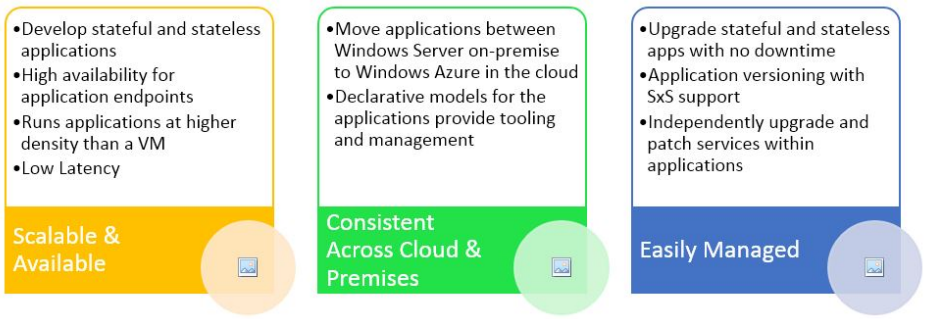
The motivation to run Sterling in Windows Fabric if we have a Fabric running SAWA v1 is that we need to use a consistent distributed systems fabric for building HA applications. We build blackbird to run the SAWA v1

blackbird. Black bird has other functions but a Fabric is a complex system and SQL Server does not have the resources to maintain and evolve a proprietary fabric software.

Sterling requirements:

- **Need:** Many workloads need a consistent distributed systems fabric
  - Distributed systems programming model
  - Solving hard distributed systems problems repeatedly
  - Reason about own system correctness
  -
- **Challenge:** Solving at scale in a distributed system is hard
  - Communication is asynchronous
  - Failures are common
  - Very large scale (100s of 1000s of machines)
- **Solution:** "Windows Fabric" is the software that stiches machines into a logical, scale-free, and consistent distributed systems fabric
  - Self-sufficient (helps with independence)
  - Self-healing (helps with cost)
  - Decentralized (helps with massive scale)
  - Layered and componentization (helps with correctness reasoning)

Windows Fabric Value Propositions



Picture 1

SAWA v1 fabric is tightly coupled into WA SQL DB. We need a solution of decoupling the Fabric services from SQL DB. This can lead to massive failures initiated by SQL components

Windows Fabric will allow us to run multiple services isolated on the same VM.

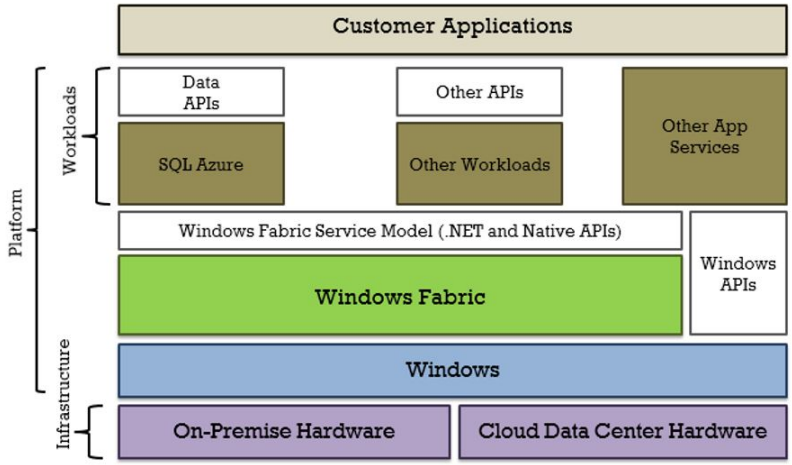
Fabric should be concise between running on-premises or private clouds and public clouds, only changing configuration settings.

Easy management and deployment, the fabric takes care of deployment of individual services and support application versioning.

Windows Fabric, Value for SAWA

- **Evolve:** SQL Azure on Windows Azure (SAWA)
    - Using WinFab as native-code, public-API “distributed systems platform”
    - Replacing logical replication with physical
    - Longevity of the solution. Blackbird was being discontinued when SQL started using.
    -
- **Value:** Service resiliency and availability
    - Easier to build distributed critical services, hence except fewer failures and better availability
    - Native code for better performance and optimization of resource use, higher density
- **Value:** Service Supportability and Platform Development Agility
    - Simpler service through several layers of functionality now abstracted in WinFab
    - Better quality, supportability
    - Physical replication: Faster future delivery of features for SQL Azure
    -
- **Value:** Service COGS
    - Reduced to no overhead of “management” nodes/environment due to distributed nature of the design
    - Use the same hardware as other services.
- **Value:** Better Azure Platform
    - As a customer of WinFab/WA, SQL Azure will help improve and ultimately benefit any customer using our Microsoft platform

Windows Fabric



Picture 2

Hardware layer can be on Premiers or a Datacenter, standalone VMs or VMs is a datacenter

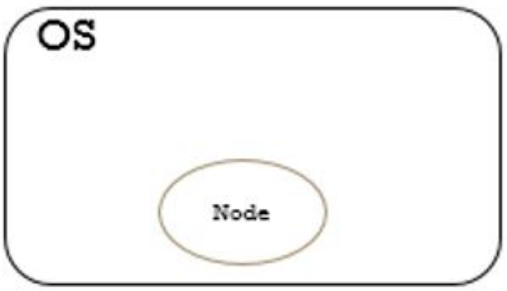
Workloads use the Fabric API to use the Fabric Services.

Services like Lync and Office 365 are already using Windows Fabric.

Windows Fabric Concepts

Node

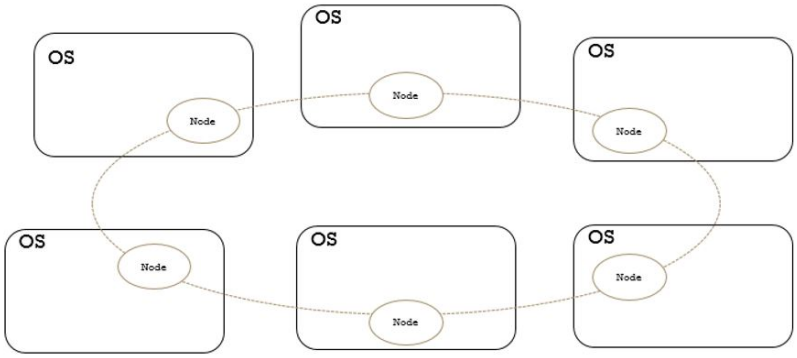
A node is an instance of Windows Fabric running on Windows Host OS (Physical or Virtual). Windows Fabric runs in a process called fabric.exe



Picture 3

Cluster

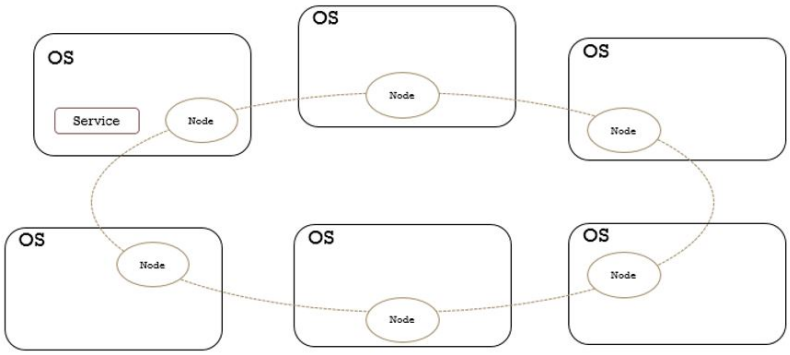
A cluster is a set of nodes stitched together to form a pool of resources. From a Fabric perspective a cluster is a single entity.



Picture 4

Services

Represent an entity (code and state) that Windows Fabric makes highly scalable, available, and reliable.



Picture 5

There are two types of Services:

- Stateless
- State full
  - In memory
  - Persisted

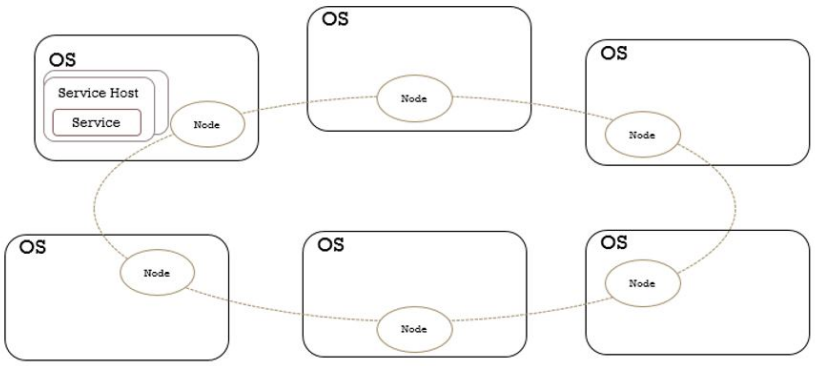
Services are:

- Discoverable
  - Have a unique name that is resolved anywhere in the cluster
- Highly Available
  - Replicate state into multiple copies that are rebuilt
- Scalable
  - Are partitioned and dynamically scale-up and scale-down

- Manageable
  - Can be deployed, created, updated, deleted and monitored
- Lightweight
  - Cheap to create/delete and densely packed
- Multi-Tenant
  - Higher density than VMs (process level) with resource and security boundaries

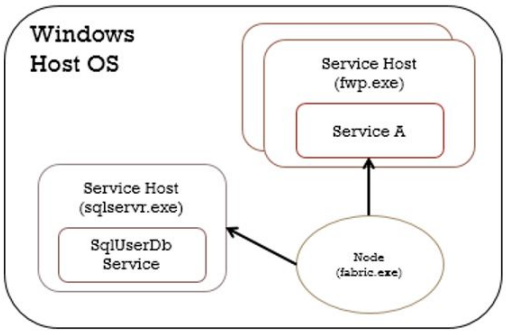
Service Host

Service host is the process in which a service runs. It has higher density than VM.



Picture 6

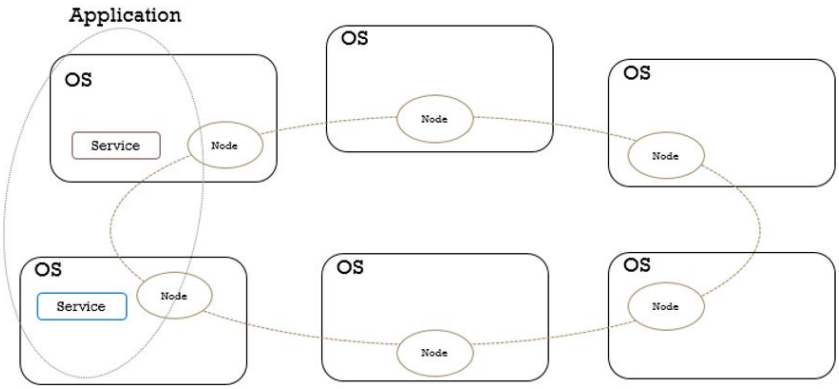
Different Services Types can be deployed to a node. We can have Multiple Services per Service Host and Multiple Service hosts per node.



Picture 7

Application

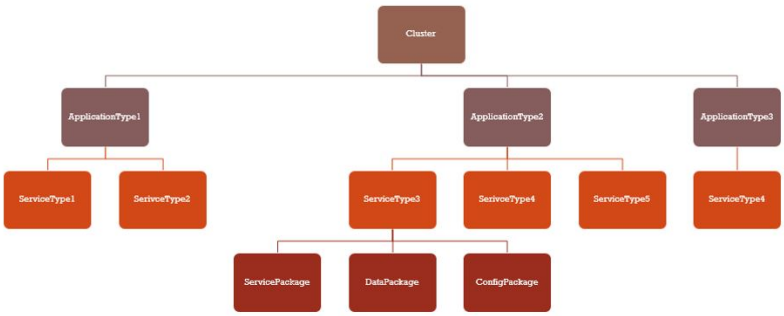
Application is a collection of services.



Picture 8

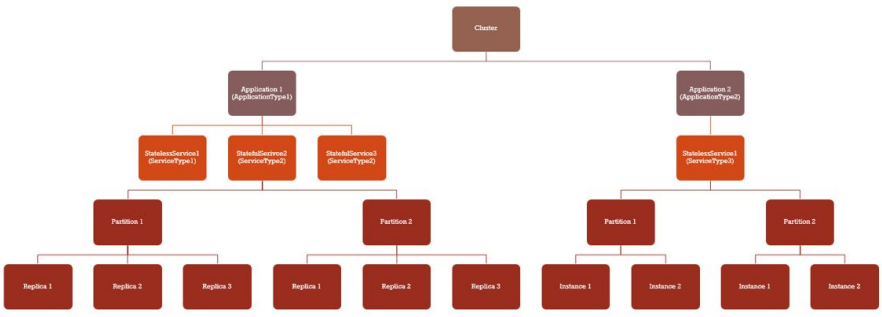
Terminology: Application and Service Types

We deploy a collection of services. Services can be of different types. Each service contains the code package, data package and configuration package. You can deploy code, data to support the code and configuration.



Picture 9

Terminology: Application and Service Instances, Partitioning and Replicas



Picture 10

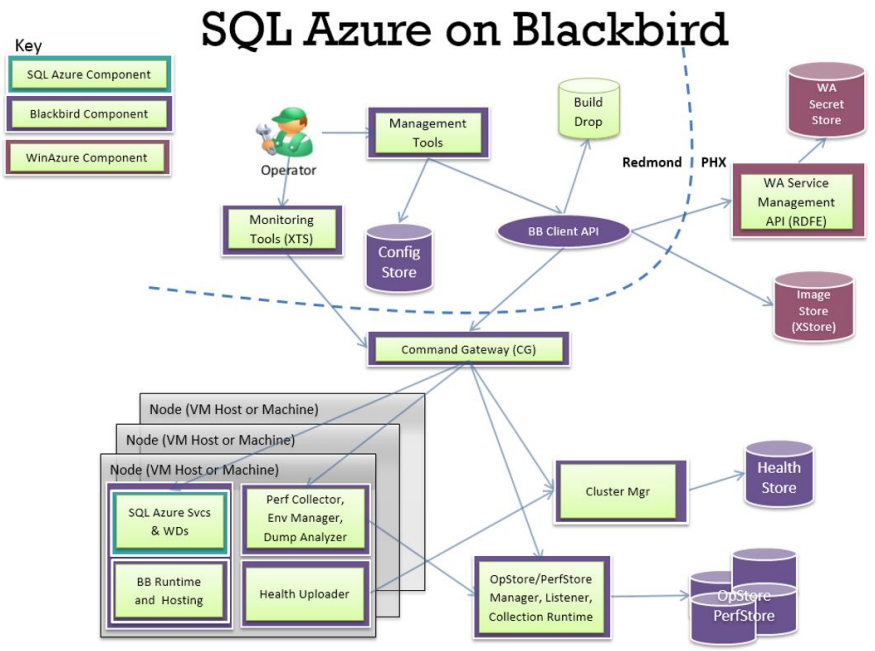
Service Model: Three Manifests

- Cluster Manifest

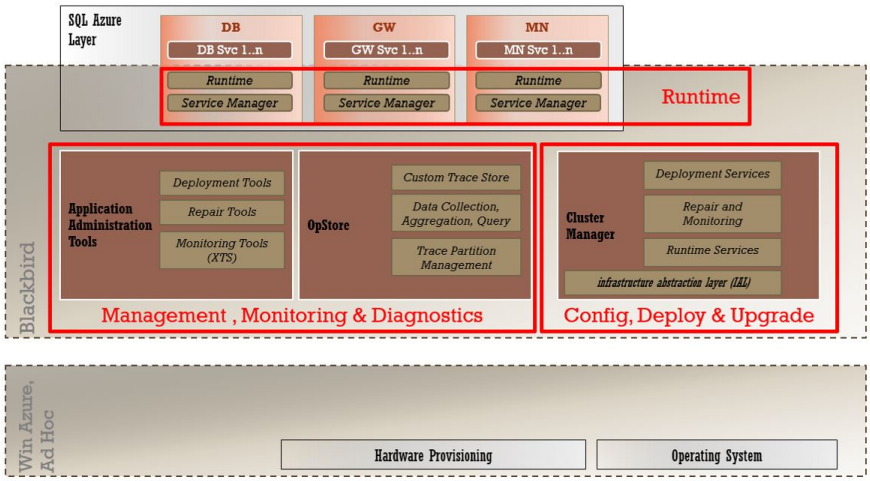
    - Machine types
        - Seed nodes
        - Security
        - Credentials
        - Security Descriptors
            - Fabric cluster check
            - Fabric client check
    - Policies
        - Trace aggregation
        - Machine repair
        - Load balancing
- Application Manifest
    - Service manifests references
        - Configuration overrides
        - RunAs credentials
        - Placement constraints
        - Isolation
    - Security
        - Principals
            - Users
            - Groups
        - Credentials
            - Certificates
        - Security Descriptors
    - Policies
        - Trace aggregation
- Service Manifest
    - Set of Service Types
        - Placement constraints
        - Load balancing metrics
        - Health properties
        - Alerts/Exceptions
    - Code
    - Configuration
    - Data
    - Endpoints
    - Security
        - Credentials
            - Certificates
        - Security Descriptors
            - Access Check

Comparison to Existing System

Black Bird

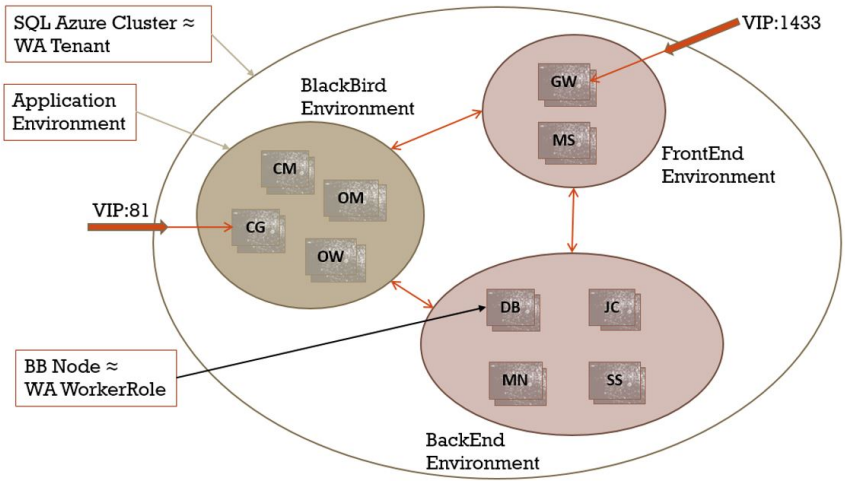


Picture 11



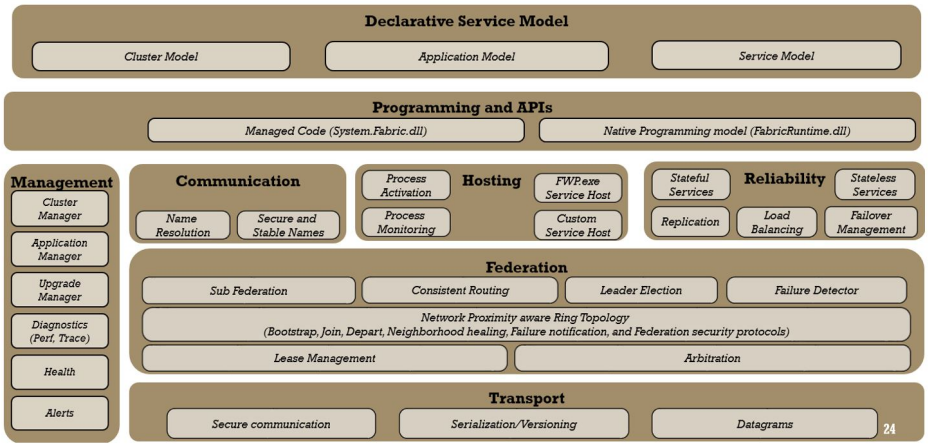
Picture 12

SAWA v1 Cluster Topology



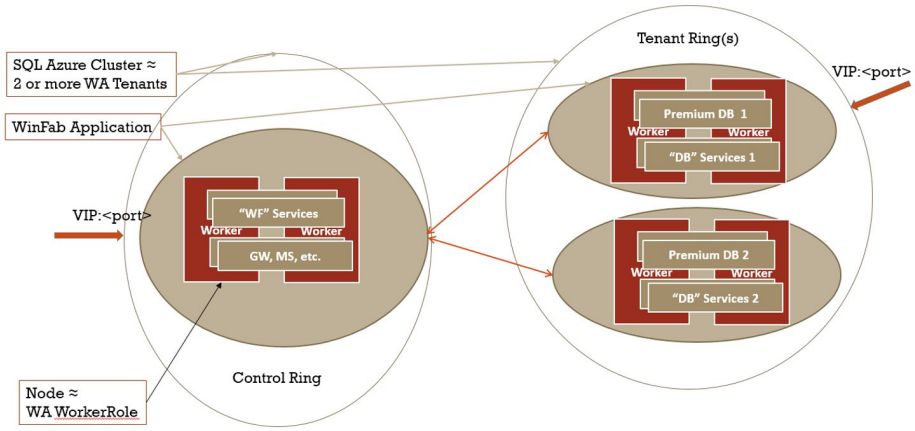
Picture 13

Windows Fabric Architecture



Picture 14

SAWA v2 (Sterling) Cluster Topology



Picture 15

How good have you found this content?