Azure Service Fabric Capabilities

Lifecyle Management

Auto Scaling

Health and Monitoring

Dev and Ops Tooling

Always On Availability

Programming Models

Container Orchestration

Build

Deploy

Operate

Azure Service Fabric

Build and operate always-on, scalable, distributed applications

Tools

Your choice of tools; use Visual Studio, PowerShell or Windows or bash. Hover, browse, share, or execute code.

Programming Models and Languages

White box stateful and stateless microservices. Native C# or .NET or with built-in programming models; code as you wish.

Deep and Diverse Tooling

With a rich set of integrated tooling in Visual Studio and PowerShell, you can extend your development stack with tools like Eclipse, Yeoman, Jenkins, or Azure CLI on Linux.

Always On

Stateful microservices on cluster溢价 available to meet latency and ensure performance while providing reliability through replication and persistence.

Debugging and Diagnostics

In addition to monitoring with monitoring the cluster you can get detailed from using ETW on Windows or co-slice in cases of severity. This allows you to create custom traces, extend them to create custom application traces, and then upload, search, and visualize using your favorite logging utilities.

One-Box Environment

Developers can develop, test, and debug applications, including multi-container applications, in a single sandbox environment that runs on the host machine or a virtual machine.

Service Discovery and Communication

Integrated system services enable easy service discovery that allows you to extend your applications to new services. The programming models provide easy to use, built-in communication classes as well as APIs to build custom communication components.

For More Information

Service Fabric Website: http://servicefabric.com
Service Fabric Blog: http://servicefabricblog.com
Service Fabric Documentation: https://aka.ms/servicefabricdocs
Try Service Fabric for Free: http://aka.ms/servicefabricvideo

Learn More

aka.ms/servicefabric

Build

Service Fabric Cluster

A Service Fabric Cluster provides a network-connected set of virtual or physical machines in which your microservices are deployed and managed. Scaling to thousands of machines is built-in reliability to ensure your demands. Connect to your node to the cluster to deploy and update your microservice-based applications.

Create Clusters Anywhere

Create and run your Service Fabric Clusters on Azure or premiers, or on a cloudless cloud where you can create isolated clusters through ARM on the Azure portal to run your code anywhere with maximum control, including support for both Windows and Linux.

No Downtime Upgrades

Upgrade applications with no downtime, as the upgrade rolls from one upgrade domain to another. With application health monitored throughout the upgrade, Service Fabric also automatically rolls back to the application to the previous version if it encounters any issues.

Operate

Health Monitoring

Azure Service Fabric includes an extensible health model for health reporting. Visualize and monitor application and cluster health using your favorite logging utilities.

Resilience and Failure Recovery

Service Fabric makes your application resilient through system services that detect and recover from failures.

Dynamic Resource Balancing

Service Fabric users services from one to another for improving resource allocation. Based on failure telemetry by monitoring CPU and network metrics, you can re-route failures from failing VMs to other VMs within the cluster.

Container Orchestration

Relying on a container management tool or application, application running in Docker containers are just another form of deployed microservice in Service Fabric.

Always On Availability

The Service Fabric platform includes self-healing capabilities so that your microservice-based applications can continue to be available in the event of failures.

Deployment

Auto Scaling

Rolling upgrade from V1 to V2

Update from one version of Service Fabric to another. Connect to any node in the cluster to deploy and update your microservice-based applications.

Stateful Microservices

Stateful microservices with fast built-in, fully-consistent state application stacks make it easy to keep your data in sync over the whole site.

Stateful or Stateless

Stateful microservices with stateless or stateful capabilities enable you to design and scale your applications using native C# or .NET programming models. Stateful microservices co-locate compute and data to reduce latency and enhance performance while providing reliability through replication and persistence.

Fault Tolerance

Replication across nodes protect your data from loss in the event of machine or service failures.

Replication and Persistence

Replication and persistence ensure that your applications are always available and durable, improving resource utilization. Extend this feature easily by specifying custom load metrics (and affinities) for your services, so optimal performance is maintained.

Container Orchestration

Relying on a container management tool or application, applications running in Docker containers are just another form of deployed microservice in Service Fabric.

Rolling upgrade from V1 to V2

Update from one version of Service Fabric to another. Connect to any node in the cluster to deploy and update your microservice-based applications.

Stateful Microservices

Stateful microservices with fast built-in, fully-consistent state application stacks make it easy to keep your data in sync over the whole site.

Stateful or Stateless

Stateful microservices with stateless or stateful capabilities enable you to design and scale your applications using native C# or .NET programming models. Stateful microservices co-locate compute and data to reduce latency and enhance performance while providing reliability through replication and persistence.

Fault Tolerance

Replication across nodes protect your data from loss in the event of machine or service failures.

Replication and Persistence

Replication and persistence ensure that your applications are always available and durable, improving resource utilization. Extend this feature easily by specifying custom load metrics (and affinities) for your services, so optimal performance is maintained.

Container Orchestration

Relying on a container management tool or application, applications running in Docker containers are just another form of deployed microservice in Service Fabric.