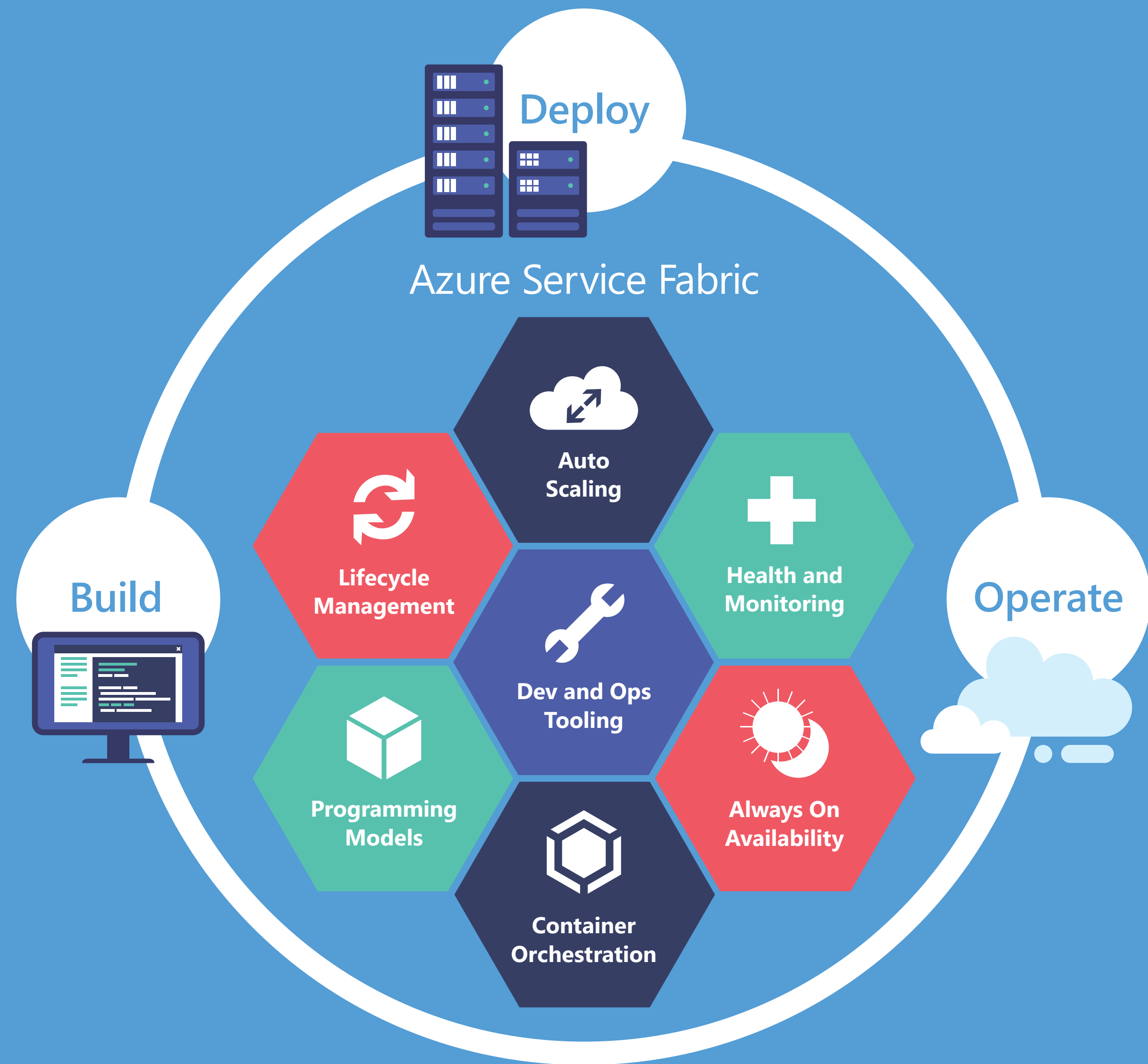




Azure Service Fabric

Build and operate always-on, scalable, distributed applications



Azure Service Fabric Capabilities



Lifecycle Management

Application development, deployment, management, and maintenance, including no downtime upgrades.



Auto Scaling

Start small and scale in/out with demand. Dynamic resource balancing with support for partitions and instances makes it easy to evolve with usage. Plus, scale each microservice independently, whether it is stateful or stateless.



Health and Monitoring

Events reporting the health of the cluster and applications are integrated into the platform. Traces collected from the entire cluster can be uploaded into a variety of stores for visualization and diagnosis.



Dev and Ops Tooling

Integration with IDEs on Windows and Linux to help you develop applications, along with built-in tooling for management of deployed applications.



Programming Models

Develop distributed microservice-based applications with your choice of stateless and stateful programming models. Create stateful microservices co-locating data and compute, reducing complexity and dependencies and improving performance.



Container Orchestration

Orchestrate your container applications and mix and match with native Service Fabric microservices and guest executables.



Always On Availability

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

For More Information

Service Fabric Video
<http://aka.ms/servicefabricvideo>

Service Fabric Team Blog
<http://aka.ms/servicefabricblog>

Service Fabric Documentation
<http://akalm.s/servicefabricdocs>

Try Service Fabric for Free
<http://aka.ms/tryfabric>

Service Fabric Labs
<http://aka.ms/servicefabricslabs>

Build

Tools

Your choice of tools: use Visual Studio; PowerShell on Windows; or Eclipse, Yeoman, Jenkins, or Azure CLI on Linux.

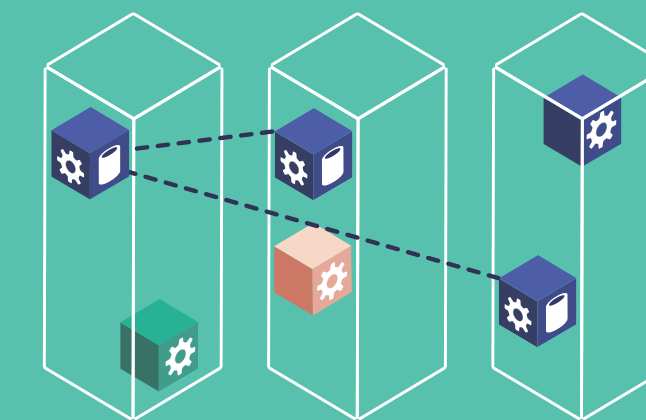
Programming Models and Languages

Write stateless and stateful microservices in Java and C# with built-in programming models such as Reliable Service and Reliable Actors. Or, use any framework or language.

Reliable Actors	Reliable Services	ASP.NET Core	Any Code/ Framework
C#, Java	C#, Java	C#	Node.js, Python, ...

Stateful Microservices

Stateful microservices with fast built-in, fully consistent state replication across nodes protect your data from loss in the event of process and machine failures.



Stateful microservices co-locate compute and data to reduce latency and enhance performance while providing reliability through replication and persistence

Debugging and Diagnostics

In addition to visualizing and monitoring the cluster, you can get detailed traces using ETW on Windows or LTTng on Linux. Easily 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, on their local machines with the same runtime that runs in production deployments. The Service Fabric one-box environment mimics multi-machine clusters so that developers get a fully functional environment without needing to set up a multi-machine cluster.

Service Discovery and Communication

Integrated system services enable easy service discovery that allows services (running in containers or outside) to find other services. The programming models provide ready-to-use, built-in communication stacks as well as APIs to build custom communication components.

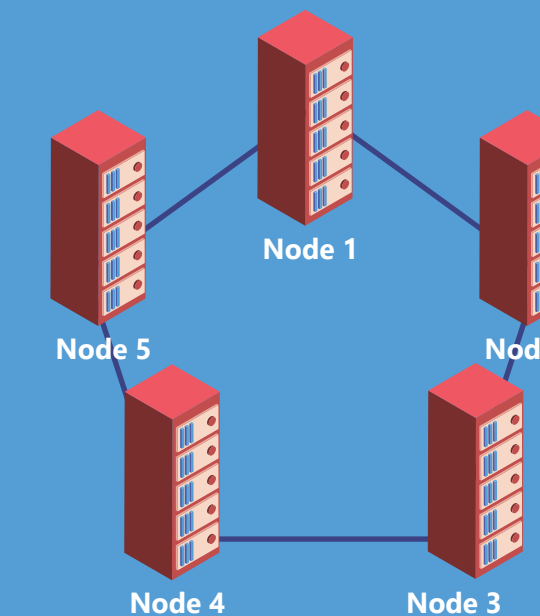


Learn More
aka.ms/servicefabric

Deploy

Service Fabric Cluster

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



Service Fabric Cluster

Create Clusters Anywhere

Create and run your Service Fabric Clusters on Azure, on-premises, or on other hosted clouds. In Azure, you can create managed clusters through ARM or the Azure portal. Run your code anywhere with maximum choice, 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 the application to the previous version if it encounters any issues.



Rolling upgrade from V1 to V2

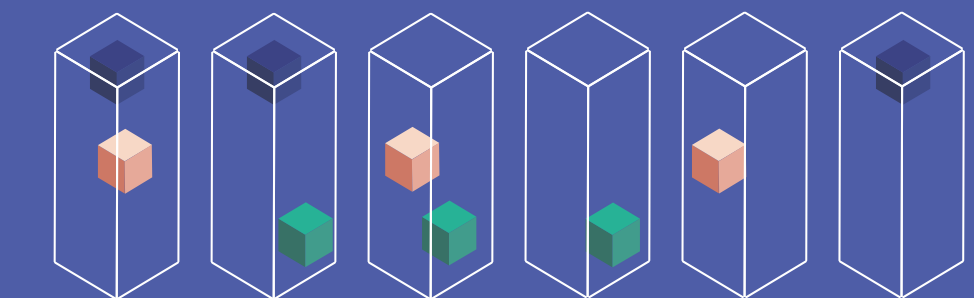
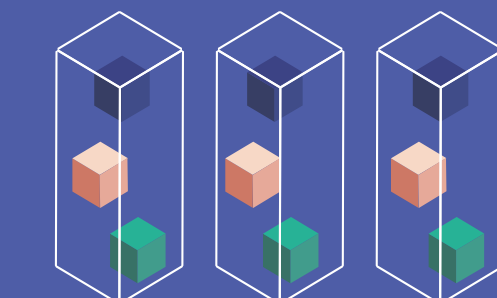
Operate

Health Monitoring

Azure Service Fabric includes an extensible health model for health reporting. Visualize and monitor application and cluster health using the browser-based Service Fabric Explorer.

Scaling

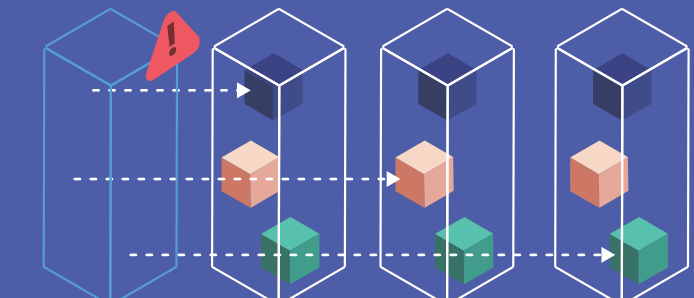
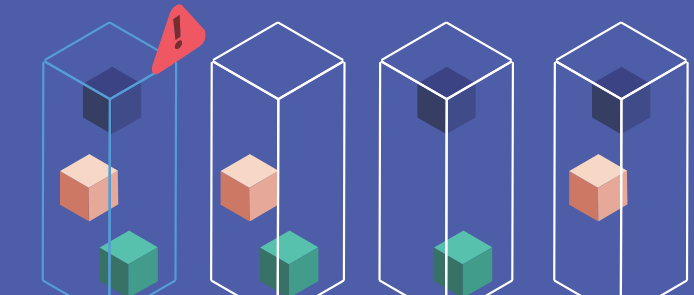
Scale stateless and stateful microservices to meet demand easily by using built-in features to partition your data and to add or remove service instances, as needed. Add or remove nodes to your cluster, and Service Fabric will dynamically scale your services out or in.



Redistributing microservices across added machines

Resiliency and Failure Recovery

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



Resiliency: how Service Fabric redistributes microservices when one machine fails

By default, VMs in a cluster are spread across logical groups known as fault domains (FDs), with no two FDs sharing host hardware (including power and network). Thus, an issue resulting in the failure of a VM in one FD doesn't affect VMs in other FDs, allowing Service Fabric to migrate services from the failed VM to other VMs within the cluster.

Dynamic Resource Balancing

Service Fabric moves services from one machine to another for improving resource utilization. Extend this feature easily by specifying custom load metrics (and affinities) for your services, so optimal performance is maintained.

Container Orchestration

Deploy, scale, and manage your containerized applications. Applications running in Docker containers are just another form of stateless microservices for Service Fabric.