1. **Azure**
   1. **Azure Types:**

Types of Azure Clouds

There are mainly three types of clouds in Microsoft Azure are:

* PAAS – (Platform as a service) is a computing platform which includes an operating system, programming language execution environment, database or web services. This Azure service is used by developers and application providers.
* SAAS – (Software as a service) is software which is centrally hosted and managed. It is a single version of the application is used for all customers. You can scale out to multiple instances. This helps you to ensure the best performance in all locations. The software is licensed through a monthly or annual subscription. MS Exchange, Office, Dynamics are offered as a SaaS
* IASS - (Infrastructure as a Service) is the foundational cloud platform layer. This Azure service is used by IT administrators for processing, storage, networks or any other fundamental computer operations.
  1. **Azure Functions:**

Azure Functions makes the app development process more productive, and lets you launch serverless applications on Microsoft Azure. It helps in processing data, coordinating with different systems for IoT, integrating various processes and systems and building simple APIs and microservices.

* + 1. **Features**

Here are some key features of Functions:

* Choice of language - Write functions using your choice of C#, Java, Javascript, Python, and other languages. See Supported languages for the complete list.
* Pay-per-use pricing model - Pay only for the time spent running your code. See the Consumption hosting plan option in the pricing section.
* Bring your own dependencies - Functions supports NuGet and NPM, so you can use your favorite libraries.
* Integrated security - Protect HTTP-triggered functions with OAuth providers such as Azure Active Directory, Facebook, Google, Twitter, and Microsoft Account.
* Simplified integration - Easily leverage Azure services and software-as-a-service (SaaS) offerings.
* Flexible development - Code your functions right in the portal or set up continuous integration and deploy your code through GitHub, Azure DevOps Services, and other supported development tools.
* Open-source - The Functions runtime is open-source and available on GitHub.
  + 1. **Types:**
* HTTPTrigger - Trigger the execution of your code by using an HTTP request. For an example, see Create your first function.
* TimerTrigger - Execute cleanup or other batch tasks on a predefined schedule. For an example, see Create a function triggered by a timer.
* CosmosDBTrigger - Process Azure Cosmos DB documents when they are added or updated in collections in a NoSQL database. For more information, see Azure Cosmos DB bindings.
* BlobTrigger - Process Azure Storage blobs when they are added to containers. You might use this function for image resizing. For more information, see Blob storage bindings.
* QueueTrigger - Respond to messages as they arrive in an Azure Storage queue. For more information, see Azure Queue storage bindings.
* EventGridTrigger - Respond to events delivered to a subscription in Azure Event Grid. Supports a subscription-based model for receiving events, which includes filtering. A good solution for building event-based architectures. For an example, see Automate resizing uploaded images using Event Grid.
* EventHubTrigger - Respond to events delivered to an Azure Event Hub. Particularly useful in application instrumentation, user experience or workflow processing, and internet-of-things (IoT) scenarios. For more information, see Event Hubs bindings.
* ServiceBusQueueTrigger - Connect your code to other Azure services or on-premises services by listening to message queues. For more information, see Service Bus bindings.
* ServiceBusTopicTrigger - Connect your code to other Azure services or on-premises services by subscribing to topics. For more information, see Service Bus bindings.
  1. **App Service:**

Azure App Service is a fully managed "Platform as a Service" (PaaS) that integrates Microsoft Azure Websites, Mobile Services, and BizTalk Services into a single service, adding new capabilities that enable integration with on-premises or cloud systems.

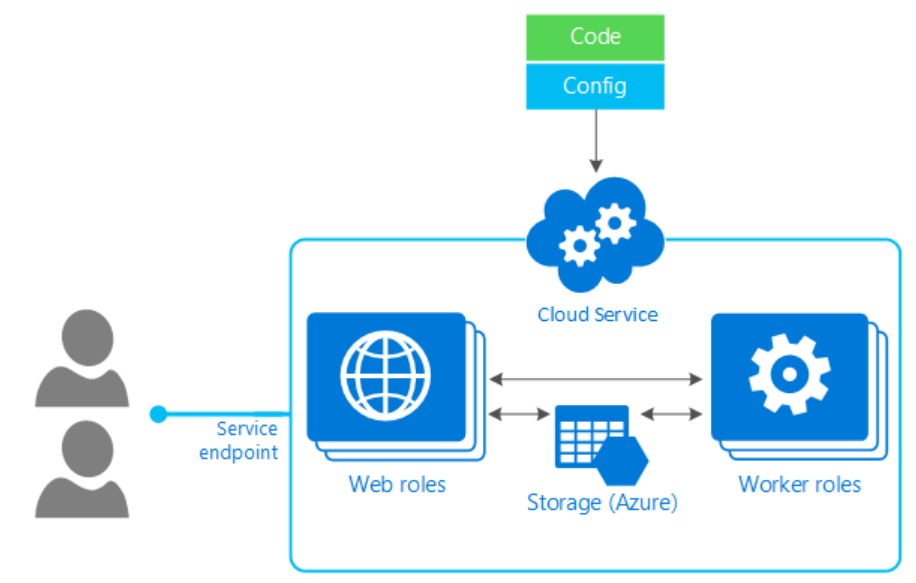


* 1. **Microsoft Service Fabric:**

Microsoft Azure Service Fabric is the next-generation cloud application platform for highly scalable, highly reliable distributed applications. It introduces many new features for packaging, deploying, upgrading, and managing distributed cloud applications.

* 1. **Cloud Service:**

Azure Cloud Services is an example of a platform as a service (PaaS). Like Azure App Service, this technology is designed to support applications that are scalable, reliable, and inexpensive to operate.



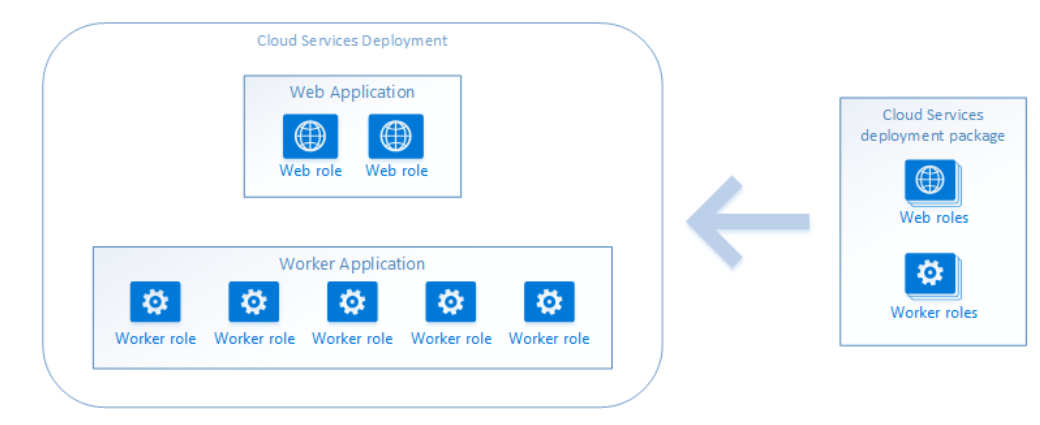
* + 1. Two types of Azure Cloud Services roles. The only difference between the two is how your role is hosted on the VMs:
* Web role: Automatically deploys and hosts your app through IIS.
* Worker role: Does not use IIS, runs your app standalone.

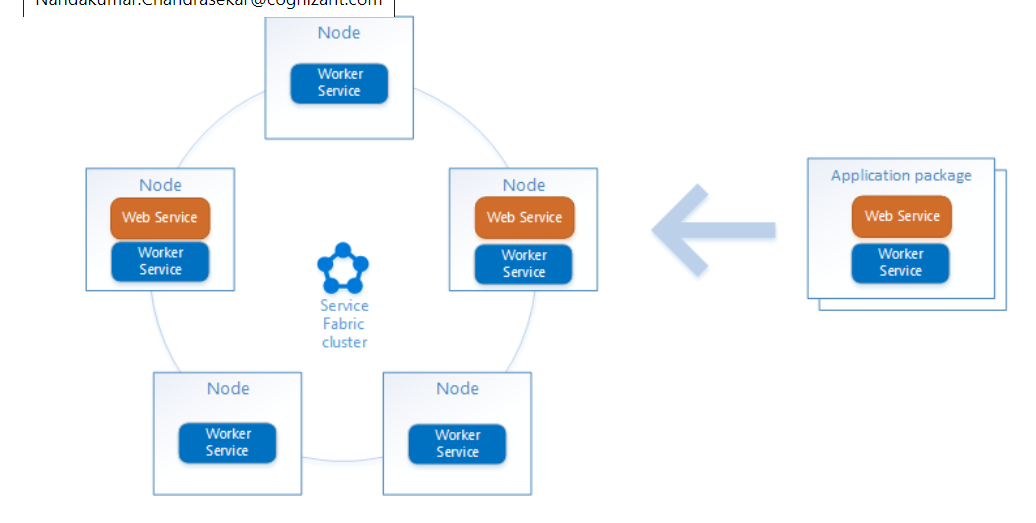
A cloud service is created from three components, the service definition (.csdef), the service config (.cscfg), and a service package (.cspkg). Both the ServiceDefinition.csdef and ServiceConfig.cscfg files are XML-based and describe the structure of the cloud service and how it's configured; collectively called the model. The ServicePackage.cspkg is a zip file that is generated from the ServiceDefinition.csdef and among other things, contains all the required binary-based dependencies. Azure creates a cloud service from both the ServicePackage.cspkg and the ServiceConfig.cscfg.

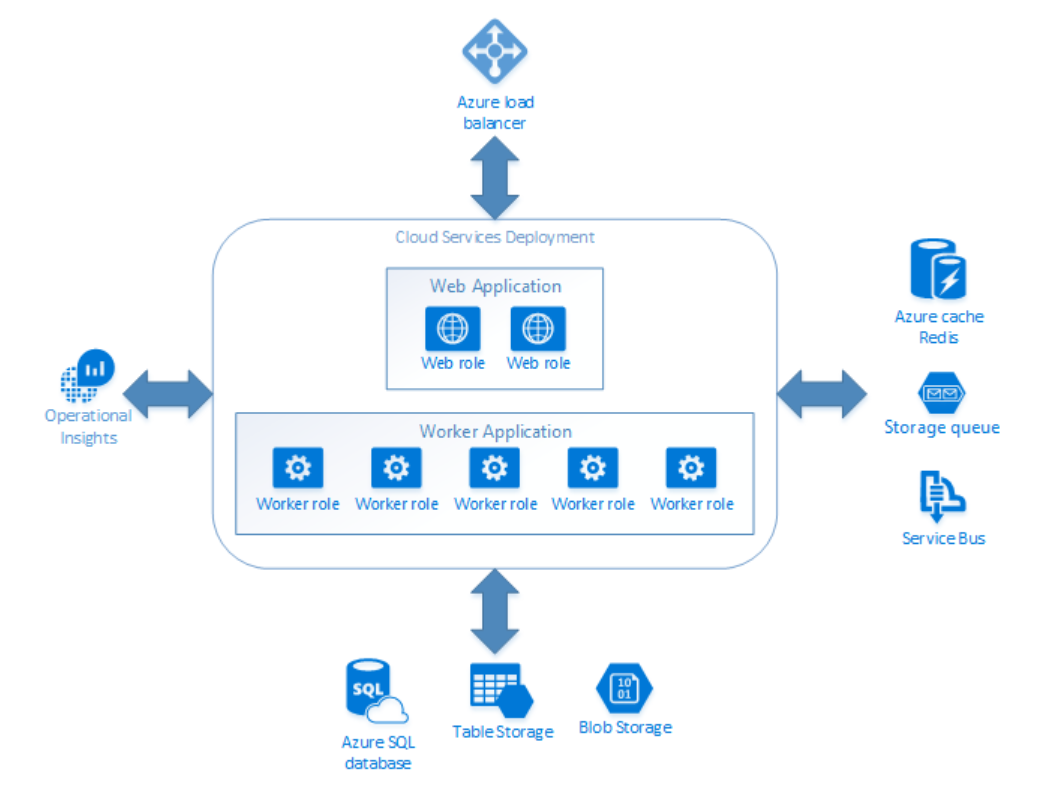
Once the cloud service is running in Azure, you can reconfigure it through the ServiceConfig.cscfg file, but you cannot alter the definition.

* 1. **Differences between Cloud Services and Service Fabric before**
* Cloud Services is about deploying applications as VMs.
* Service Fabric is about deploying applications to existing VMs or machines running Service Fabric on Windows or Linux

Cloud Service Sample below:

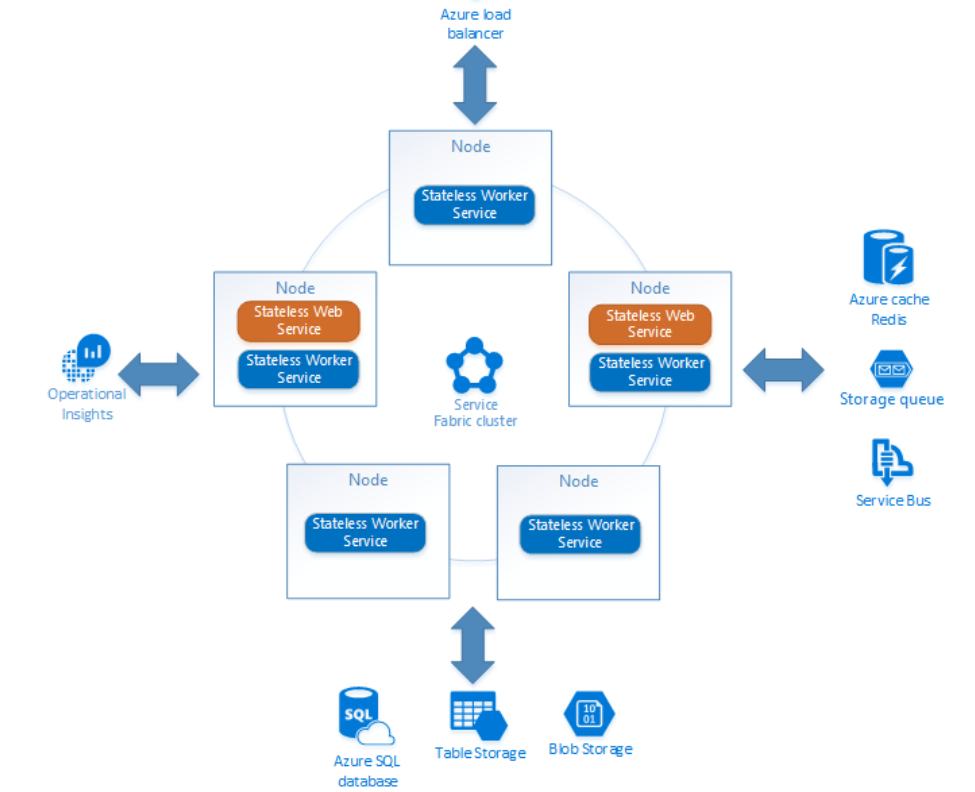


Service Fabric Sample Below:

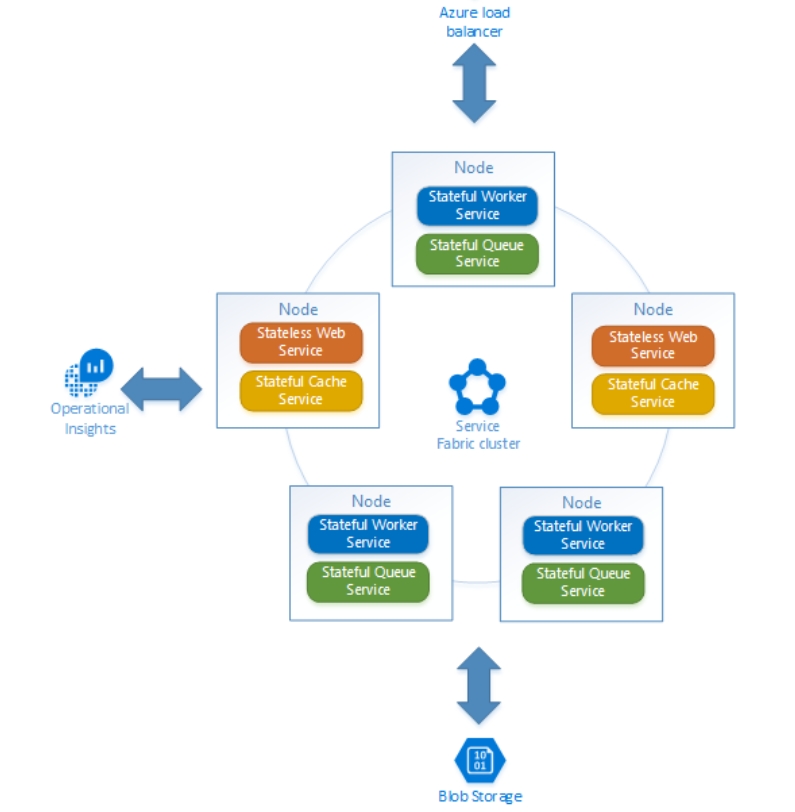
Cloud service Architecture:

Service Fabric Architecture -for migrated ones and new development ones

Service Fabric – can be stateful or stateless, but cloud services are stateless and need to have DB for maintaining connectivity and also connections to other items through queue - but in-service fabric we don’t need Queue for connectivity.

Service Fabric architectures: 

**Service Fabric without migration or without queues:**



Advantages of Service Fabric:

* Fast deployment times. Creating VM instances can be time consuming. In Service Fabric, VMs are only deployed once to form a cluster that hosts the Service Fabric application platform. From that point on, application packages can be deployed to the cluster very quickly.
* High-density hosting. In Cloud Services, a Worker Role VM hosts one workload. In Service Fabric, applications are separate from the VMs that run them, meaning you can deploy a large number of applications to a small number of VMs, which can lower overall cost for larger deployments.
* The Service Fabric platform can run anywhere that has Windows Server or Linux machines, whether it's Azure or on-premises. The platform provides an abstraction layer over the underlying infrastructure so your application can run on different environments.
* Distributed application management. Service Fabric is a platform that not only hosts distributed applications, but also helps manage their lifecycle independently of the hosting VM or machine lifecycle.