Cloud service models and cloud deployment models

Before we start on the actual topic (the Azure platform), we should clarify some terms related to cloud computing. Knowing these concepts, we will then be in a position to identify individual parts of the Azure platform.

Let's start.

Cloud service models

The first term we will look at is **cloud service models**.

All workloads in a cloud scenario use resources from an extremely large resource pool that is operated (managed) by you or a cloud service provider. These resources include servers, storage, networks, applications, services, and much more.

The cloud service models describe to what extent your resources are managed by yourself or by your cloud service providers.

Let's look at the available service models. In the following diagram, you will find a comparison of the models and the existing management responsibilities. Areas that are colored in blue are managed by you: all others are the responsibility of your provider:

Chart

Description automatically generated

The offers are mainly categorized into the following service models:

* **On-premises**: On-premises describes a model in which the user manages all resources alone.
* **Infrastructure as a Service** (**IaaS**): IaaS describes a model in which the cloud provider gives the consumer the ability to create and configure resources from the computing layer upwards. This includes virtual machines, containers, networks, appliances, and many other infrastructure-related resources.
* **Platform as a Service** (**PaaS**): PaaS gives the consumer an environment from the operating system upwards. So the consumer is not responsible for the underlying infrastructure.
* **Software as a Service** (**SaaS**): SaaS is the model with the lowest levels of control and required management. A SaaS application is reachable from multiple clients and consumers, and the owning consumer doesn't have any control over the backend, except for application-related management tasks.

Cloud deployment models

The second term we will look at is **cloud deployment models**.

Cloud deployment models describe the way in which resources are provided in the cloud.

Which cloud deployment models are available?

Let's look at the following diagram first:

Chart

Description automatically generated

The deployment model based on the on-premises service model is called the **private cloud**. A private cloud is an environment/infrastructure, built and operated by a single organization, which is only for internal use.

In the context of this book, you should know that the Windows Azure Pack (a free add-on for the Windows server) gives you the opportunity to deploy Azure technologies in a private cloud environment.

The deployment model based on the IaaS and the PaaS service model is called the **public cloud**. A public cloud is an offer from a service provider (for example, Microsoft Azure), that can be accessed by the public. This includes individuals as well as companies.

Note

Note: When we talk about Azure in this book, it always means the public cloud model.

There is still a third deployment model available, which is the **hybrid cloud**. A hybrid cloud combines parts of the private and public clouds. It is defined as a private cloud environment at the consumer's site, as well as the public cloud infrastructure that the consumer uses.

In the context of this book, you should know that Azure Stack (a new offering from Microsoft) gives you the opportunity to build a hybrid cloud environment:

Chart

Description automatically generated

Azure execution models

Now we have acquired some basic knowledge about cloud computing, the question arises: Where do we go Each of these models can be used independently, but also in combination. from here?

I think that when we start talking about the Microsoft Azure platform, we should first talk about running applications in the cloud. For this, Microsoft Azure provides five deployment models (also known as **Azure execution models**), that are outlined in the following diagram:

Diagram

Description automatically generated

The five execution models are:

* **Cloud services**: Cloud services are the oldest part of the Azure platform and they have been available since its first preview (announced at the Microsoft Professional Developers Conference 2008). Cloud services are a PaaS offering from Azure and even though there are now some alternatives, they are still the leading solution. Cloud services are highly available, scalable, and multi-layered web apps hosted on a Windows Virtual Machine with an installed IIS.
* **Virtual machines** (**VM**): This model is the IaaS offer from Azure. With VMs (based on Windows or Linux OS) you have the flexibility to realize your own workloads. In order to work with VMs as easily as possible, over 3,000 prefabricated images are available in the Azure Marketplace.
* **Batch**: Azure Batch is a platform service for running large-scale parallel and **highperformance computing** (**HPC**) applications efficiently in the cloud.
* **Websites**: With this model, you can quickly create and deploy your websites.
* **Mobile services**: With this model, you can quickly create and host a backend for any mobile device.

Each of these models can be used independently, but also in combination.

As I mentioned earlier, there are hundreds of updates every year on the Azure platform and the Azure execution models have also been affected. The execution models' websites and mobile services are now out of date and have been replaced by the execution model, App Services (Azure App Services):

Diagram

Description automatically generated

Note

Azure App Services is not just the replacement for the outdated execution models but much more.

Just look at the following diagram:

Diagram

Description automatically generated

As you can see, the Azure App Services execution model currently consists of the following elements:

* **Web Apps**: Web Apps is simply the new name for the Azure websites' execution model and does not really differ in its functionality from the previous component (by combining it with other app service elements, but there are extended features).
* **Mobile Apps**: Mobile Apps is the new name for the Azure mobile services execution model and does not differ in its functionality from the previous component (by combining it with other app service elements, but there are more features).
* **Logic Apps**: With Logic Apps, you can easily build and deploy powerful integration solutions by automating business processes or integrating your SaaS and enterprise applications. A visual designer is available for creating the necessary workflows. Logic Apps are the next evolutionary step in the Integration Services and the BizTalk services, and starting from 2018 are the only offer in this area.
* **API Apps**: First of all, I must mention that the Azure API Apps are the only truly new component of this model. API Apps allows you to discover, host, manage, and market APIs and SaaS connectors in a modern, feature-rich, scalable, and globally available platform. API Apps is a complete solution for enterprise developers and system integrators, which extends the development of Web Apps (mobile applications) with numerous useful features.
* **Function Apps**: This is not really a component of the Azure App Services. Azure Functions uses the Azure App Services environment only to handle the functionality.
* **Container Apps**: This is not really a component of the Azure App Services. Azure Container Services uses the Azure App Services environment only to handle the functionality.
* **Microsoft PowerApps**: This is not really a component of the Azure App Services. Microsoft PowerApps is a SaaS variant of the Azure App Services and uses the Azure App Services and the Azure App Services environment only to handle the functionality.

A further change to the area of Azure execution models is the addition of another model, Azure Service Fabric (also known as Azure Microservice Architecture), in 2016:

Diagram

Description automatically generated

With Azure Service Fabric, you can focus on building applications and business logic, and let the Azure platform solve all other problems by analyzing reliability, scalability, management, and latency.

Azure application building blocks

Let's return to the last diagram in the previous section. In the bottom layer, you'll find two other components of the Azure platform:

* Azure application building blocks
* Azure data services

Both Azure application building blocks and Azure data services, are managed services that extend the platform with so-called common capabilities (shared functionalities).

In the following diagram, you will find an overview of the Azure application building blocks. Because of the high number of individual components on offer, they are only represented in categories:

Diagram

Description automatically generated

The service categories are as follows:

* **Media**: With the Azure Media Services, the Azure platform provides an extensive portfolio for on-demand and live video processing, video and audio encoding, and much more.
* **Cache**: The use of a cache allows you high throughput and consistent data access with low latency for fast and scalable Azure applications. The solution for the Azure platform called **Azure Redis Cache** is based on the popular open source Redis cache and has been realized as a fully managed service.
* **CDN**: The **Content Delivery Network** (**CDN**) offers a global solution for delivering high-bandwidth content that is hosted in Azure or any other location (for example any HTTP/HTTPS location).
* **Identity**: This category contains the identity services, such as the Azure **Active Directory** (**AD**), Azure AD B2C, Multi-Factor Authentication, and Azure Key Vault which is a safe place for your certificates.
* **Networking**: This category contains the basic networking services. For example, Azure ExpressRoute, VNet Peering, and VPN gateways.
* **Integration**: The integration services include interfaces for hybrid connections, **Enterprise Application Integration** (**EAI**) and **Electronic Data Interchange** (**EDI**) message processing, an easy-to-use administrative portal for trading partners as well as support for common EDI schemas and comprehensive EDI processing via X12 and AS2.
* **Messaging**: The messaging services include all interfaces from the Azure Service Bus not included in the integration category,  for example, Azure Service Bus topics and Azure Service Bus Notification Hubs.
* **Dev Services**: These are cloud-based development tools for version control, collaboration, and other development-related tasks, for example **Visual Studio Team Services** (**VSTS**) and the Azure DevTest Labs.
* **IoT**: IoT services include the fundamental tools needed to work with devices used for the IoT,  for example Azure IoT Hub, IoT Edge, and Azure Event Hubs.
* **Blockchain**: Blockchain is a way for businesses, industries, and organizations to make and verify transactions—streamlining business processes, and reducing the potential for fraud.

Azure data services

Azure data services are managed services that extend the platform with so-called common capabilities (shared functionalities). Because of the special importance of data in today's digital world, they were separated from the Azure application building blocks and represent a separate kind of service.

In the following diagram, you will find an overview of the Azure data services. Because of the high number of individual components on offer, these are only represented in categories:

Diagram

Description automatically generated

The service categories are as follows:

* **Storage**: This category includes a total of five very different services: Blob Storage (storage of unstructured data), Table Storage (NoSQL storage based on key-value pairs), Queue Storage (for message processing), File Storage, and Disk Storage (Premium Storage).
* **SQL Database as a Service**: This category includes three full managed Databases as a Service: SQL Server, MySQL and PostgreSQL. This category also includes some special offers: SQL Server DWH, SQL Server Stretch DB, SQL Server Elastic DB. All special offers are further developments of the SQL Server as a Service and cover specific cloud workloads.
* **NoSQL Database as a Service**: This category includes a fully managed NoSQL Database as a Service: Azure CosmosDB. A NoSQL database is used to store semi-structured data. A NoSQL database distinguishes between storing key-values, graphs, and document data. You can specify what type of storage you want to use when creating the database.
* **Big Data**: This category includes, along with Azure HDInsight, a fully managed implementation of Apache Hadoop. In addition, implementations (with varying levels of development) are available for Apache Storm, Apache Spark, Apache Kafka, and the Microsoft R Server.
* **Analytics**: This category includes tools to analyze and process data, such as Azure Stream Analytics, Azure Data Lake Analytics, and the Azure Data Factory.
* **AI**: This category includes a fully managed service, **Azure Machine Learning** (**Azure ML**), that enables you to easily build, deploy and share predictive analytics solutions, and also includes some prefabricated solutions for immediate use (Microsoft Cognitive Services).
* **Visualization**: This category is a special case because the offered service (Microsoft PowerBI) is strictly an Azure service but is only offered by Microsoft as an SaaS solution.

Azure platform services

And finally, we now come to a special component of the Azure application building blocks: the Azure platform services. Azure platform services are all services that are responsible for internal workflows on the platform itself.

Since Microsoft has an open strategy for the Azure platform, these services are usually also available to the end user.

Examples of these type of services are as follows:

* Azure Load Balancer
* Azure Traffic Manager and much  more

How is access to Azure organized?

In the previous sections of this chapter, we've had an overview of the Azure platform. Now let's dig a bit deeper.

The first point in our discovery journey— How is access to the Azure platform organized?

As long as we look at Azure from our personal point of view, the answer is simple—our entire world consists of an Azure account, a subscription and the direct or indirect handling of Azure resources, as shown here:

Diagram

Description automatically generated

But if we look at the situation from the perspective of an enterprise, it becomes much more complicated, as shown in the next figure. The reason for this is Microsoft's attempt to delineate your company with an Azure enrollment as precisely as possible.

Let's have a look—the high-level element in the figure is now the enterprise element linked to the Enterprise Administrator's role. Here, the most comprehensive and highest rights exist. An Enterprise Administrator is allowed to do everything within the Azure enrollment.

The rights of the Enterprise Administrator include:

* If necessary, he/she can appoint additional Enterprise Administrators
* It defines the so-called departments and appoints corresponding department administrators
* It can set up accounts as required

The Enterprise Administrator is also the only person who can access all consumption and cost data at every level of the Azure enrollment.

The next element is departments (linked to the Department Administrator's role)—with the creation of departments, you can subdivide your enrollment into logical units. Even if the term **department** suggests something different, you have given yourself flexibility in terms of how the elements are divided.

The decision on how elements are classified is actually made based on the following:

* Functional aspects (in fact, according to the organizational structure)
* Business interests (that is according to the project's business)
* Geographical aspects (different locations, branch offices, and so on)

Let's go to the Department Administrators—they have the ability to create accounts within their department and, if necessary, can create a cost center (for a complete cost control).

From the next level, everything is as usual (Azure account, subscriptions, Azure resources and so on):

Diagram

Description automatically generated

How is work with Azure organized?

Now we know how access to Azure is organized. The question arises— How do I work with Azure?

In general, all work is done through the Azure portal (also known as the Azure management portal).

The Azure portal is a web-based application that can be used to create, manage, and remove all types of Azure resources and services. It includes a customizable dashboard, the ability to create your own dashboards and tooling for managing or monitoring Azure resources. It also provides information for cost and usage management.

The Azure portal is located at [https://portal.azure.com.](https://portal.azure.com/)

A big thing, but wait, there's even more. Look at the following diagram:

Diagram

Description automatically generated

As you can see, there are four other ways to work with Azure, namely:

* **Azure PowerShell**: Azure PowerShell is a set of modules that provides cmdlets to manage Azure. You can use the cmdlets to create, manage, and remove Azure services. In most cases, you can use the cmdlets for the same tasks that you perform in the Azure portal. The cmdlets can help you achieve consistent, repeatable, and hands-off deployments. Azure PowerShell is an open source project and available at <https://github.com/Azure/azure-powershell>.
* **Azure CLI**:The Azure CLI is a tool that you can use to create, manage, and remove Azure resources from the command-line. The Azure CLI is an open source project and available for Linux, macOS, and Windows at <https://github.com/Azure/azure-cli>.
* **REST APIs**:All Azure services, including the Azure management portal, provide their own REST APIs for their functionality. They can therefore be accessed by any application that RESTful services can process.
* **Client libraries**:In order for software developers to write applications in the programming language of their choice, Microsoft offers wrapper classes for the REST APIs. These are available as a so-called Azure SDK for numerous programming languages (for example .NET, Java, Node.js) at <https://github.com/Azure>.