Chapter

1

Gaining the Azure Data Engineer Associate Certification

WHAT YOU WILL LEARN IN THIS CHAPTER:

- ✓ The journey to certification
- ✓ How to pass Exam DP-203
- ✓ Azure product name recognition

The Azure Data Engineer Associate certification is one of several data-related accreditations offered by Microsoft. When compared to other available data-oriented certifications, as seen in Table 1.1, the Azure Data Engineer Associate certification is more complex. The primary reason for the added complexity relates to the dependencies a data scientist and a data analyst have on an Azure data engineer.

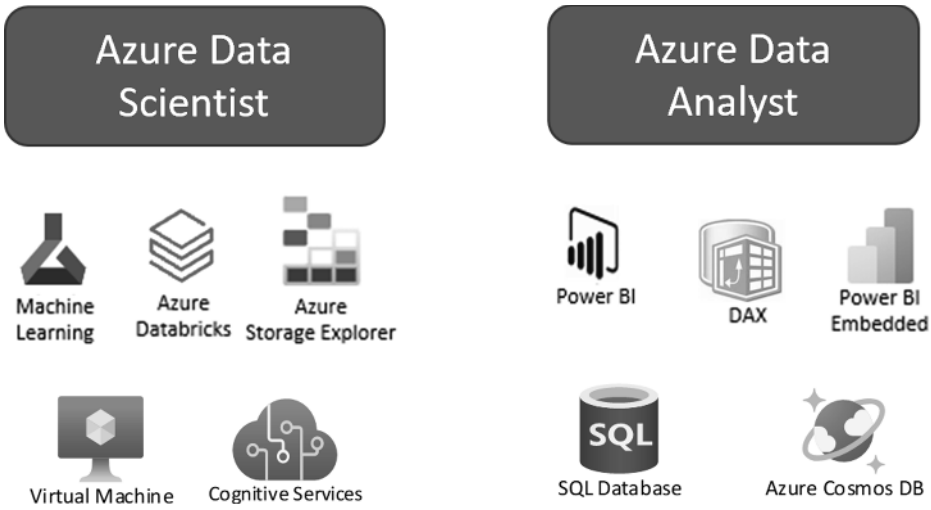
One primary responsibility of an Azure data engineer is to design a companywide platform on which data scientists or data analysts perform their experiments and analyses. For them to run these experiments and perform such analyses, the data needs to be hosted in a compliant, secured, performant, organized, and reliant manner. Although an Azure data scientist and an Azure data analyst are expected to understand and implement suitable analytical workloads, an Azure data engineer is expected to have a much broader range of responsibilities and reach.

TABLE 1.1 Azure certifications

Azure certification	Description
Azure Data Fundamentals	DP-900: Microsoft Data Fundamentals; basic knowledge of data concepts and Azure data products
Azure Data Scientist Associate	DP-100: Designing and Implementing a Data Science Solution on Azure; knowledge of predictive models and machine learning
Azure Data Analyst Associate	DA-100: Analyzing Data with Microsoft Power BI; proficiency to clean and transform data, provide data visualizations
Azure Data Engineer Associate	DP-203: Data Engineering on Microsoft Azure; design and support the platform for data scientists and data analysts to achieve their objectives
Azure Database Administrator Associate	DP-300: Administering Relational Databases on Microsoft Azure; ability to administer SQL Server database on-premises and in the cloud

I'm not suggesting that the other certifications are easy; my intention is to point out how the certifications link together. To better explain the expectations and responsibilities for each role, begin by reviewing Figure 1.1.

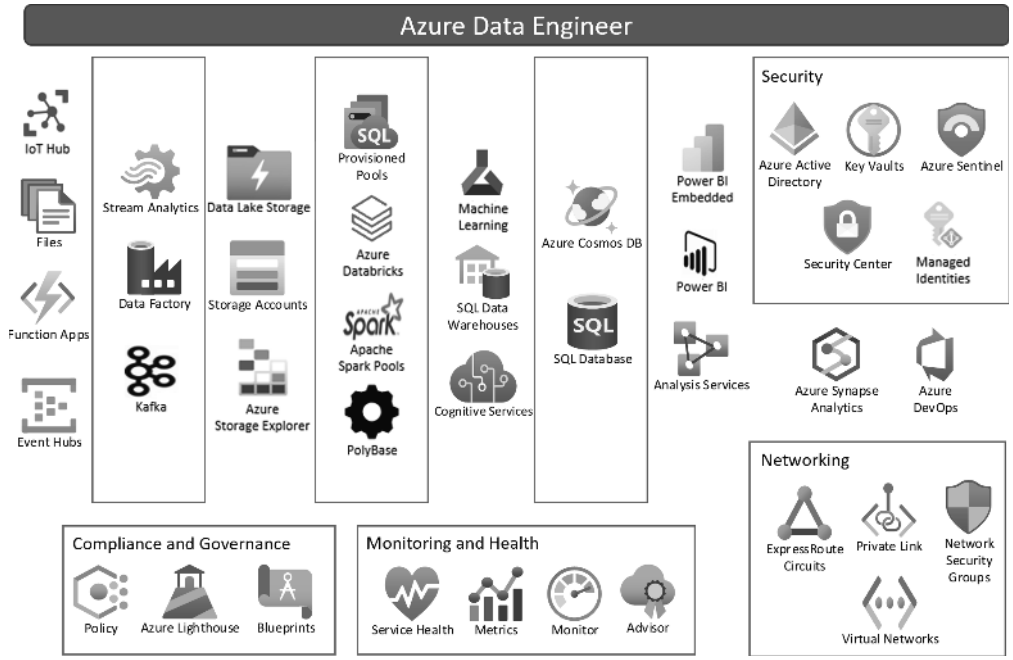
FIGURE 1.1 Comparing the Azure data scientist and analyst roles



Notice that an Azure data scientist primarily focuses on Azure products that concentrate on the preparation, training, and experimentation of data, whereas an Azure data analyst focuses on making proper connections to data sources, extracting data from those sources, and then creating easy-to-comprehend data visualizations. It is certainly true that some individuals who have either of those job titles perform duties and have responsibilities other than these. In general, however, both roles focus heavily on the data and not so much on the end-to-end platform used for placing, accessing, manipulating, and consuming the data. That end-to-end responsibility falls to the Azure data engineer; the scope of that role is partially illustrated in Figure 1.2.

Considering all the products and features within the scope of responsibility of an Azure data engineer, it is clear that this role exceeds, in both scale and scope, that of data management, preparation, and analysis. In addition to those responsibilities typically given to an Azure data scientist and an Azure data analyst, the Azure data engineer role expands into aspects of the provisioning, securing, and management of a company's data services solution.

FIGURE 1.2 The Azure data engineer role

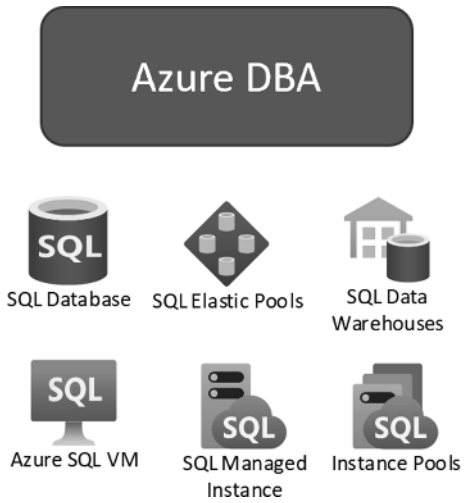


At this point in your career, many of the products in Figure 1.2 should already be somewhat familiar. You will be learning in more depth what these Azure products do, how they work together, and in what scenarios to use them. Understanding these topics is necessary to pass the Data Engineering on Microsoft Azure exam and become a great Azure data engineer.

Before moving on, there is one more certification we need to review: the Azure Database Administrator Associate certification. Figure 1.3 shows the context of the role and responsibilities.

This role focuses on *Structured Query Language (SQL)* Server and other Azure data-related products and services. There are many different implementations and operational aspects of running SQL Server solutions, such as on-premises, in the cloud, or using a hybrid model. It should be without any doubt that performing Big Data analytics running the SQL Server *relational database management system (RDBMS)* is possible and the resulting insights realize great value. Those who have been using SQL to *create, read, update, and delete (CRUD)* data stored in relational data structures will find this certification valuable. However, remaining isolated in the context of structured, or "relational," data would reduce and possibly prevent the realization of what other products and concepts can offer. *Not Only SQL (NoSQL)*, or "unstructured" or "nonrelational," data is not necessarily the substitute but rather a technique that must be added to anyone working in the *information technology (IT)* data management field. NoSQL and the tools commonly used in this discipline are part of a skillset that is no longer optional.

FIGURE 1.3 The Azure database administrator associate role



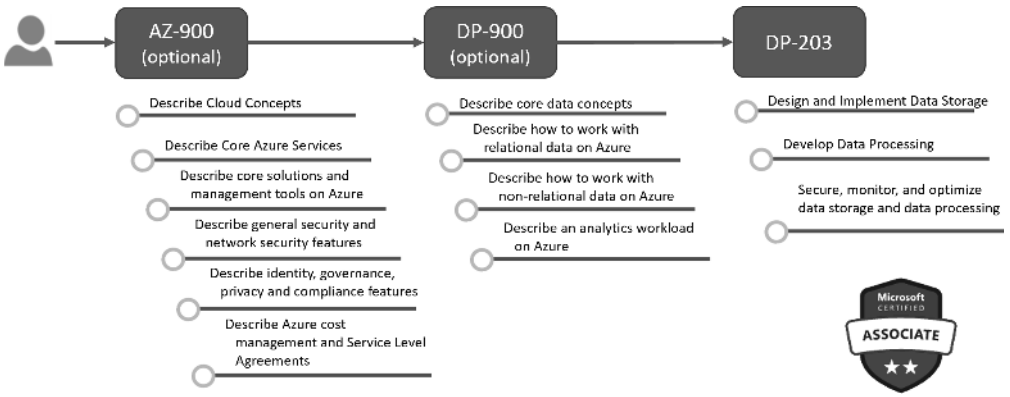
The Journey to Certification

When you set a goal—in this case, to attain the Azure Data Engineer Associate certification—make sure to take a moment to acknowledge the journey. Although the objective of this book is to help you earn a passing score on the test, the effort you make and the knowledge you retain are the critical components involved in performing the role. The certification is a validation of what you have learned. What you will learn in this book will not only help you study for the exam, but will provide you with opportunities to apply and use what you learn. Then you can take that skillset into your daily job.

Unlike numerous other Microsoft Azure certifications, there is no prerequisite to take the DP-203: Data Engineering on Microsoft Azure exam, but this doesn't mean you shouldn't consider preparing for and procuring some additional exam experience. Figure 1.4 provides a viable path to gradually build confidence and pertinent knowledge before taking the exam.

The first recommended exam on the path toward Azure Data Engineer Associate certification is AZ-900, Azure Fundamentals. This exam will help you understand basic cloud concepts, Azure product descriptions, and use case scenarios of core Azure products, features, and services. AZ-900 also covers Azure security, identity, networking, privacy, and governance constructs, as well as the products used for applying them. I will cover each of those concepts in this book, so if you choose not to pursue AZ-900, you will still learn enough about all those products to pass the exam. If you do prepare and pass the AZ-900, then most of what you read in this book about those topics will be a review, but with perhaps a different approach and context.

FIGURE 1.4 A path to the Azure Data Engineer Associate certification



The other optional exam on the path to the Azure Data Engineer Associate certification is DP-900, Azure Data Fundamentals (see Table 1.1). This exam builds on the AZ-900, as it assumes you know the basics of Azure and therefore focuses mostly on data-oriented Azure products and concepts. If you are reading this book and have a zero cognizant comparative description between relational and nonrelational data structures, then you might want to take a step back and seriously consider the DP-900 before proceeding further. Those two concepts are covered in detail in this book, but you should be familiar with those two terms. The DP-900 also introduces the concept of data analytics, which should render a visualization of the Azure Synapse Analytics product into your brain. If it doesn't, then that is another reason to consider DP-900.

Remember that the journey, the acquired skillset, and the experience you gain along the way will make you successful in your career and job. The certification is only one endorsement of your skills. So, two things you should know before moving on: First, consider preparing for and taking both the AZ-900 and DP-900 exams before DP-203. The knowledge you gain is 100 percent related to the DP-203 exam and your job. Second, feel confident that although those two exams are recommended yet optional, the content in this book will be enough for you to be successful in obtaining the Azure Data Engineer Associate certification.

How to Pass Exam DP-203

There is no secret process or shortcut for passing the exam, at least using publicly available sources. You must invest the time and effort necessary to learn the required content to be successful. Consider this book as a source to guide you along that process. The following list contains ways to help you learn the required information and gain the skills for passing the Data Engineering on Microsoft Azure exam:

- Have a clear understanding of the exam expectations and requirements.
- Use Azure daily.

- Read articles on Azure to stay current.
- Understand all Azure products.

Before proceeding, it is important to note that before attempting any Microsoft certification exam, you must commit to a *nondisclosure agreement (NDA)*. In other words, you agree not to capture and/or share notes concerning the content contained within the actual exam. To protect the integrity of the exam and the Azure Data Engineer Associate certification itself, I will make no references to any actual question found within the exam. When you pass the exam using the content of this book, it is because you have learned the required information and can therefore wear the badge and share the credential with true pride.

Understanding the Exam Expectations and Requirements

It doesn't take much imagination to realize that it is helpful to know what an exam is testing before you take it. The more you know about the tested topics, the higher probability you have of passing it. There is a list of the skills measured on the DP-203 exam on the Microsoft certification website; go to `https://aka.ms/certification` and search for **data engineer**. Click the Exam DP-203: Data Engineering on Microsoft Azure link and then click the Download Exam Skills Outline link toward the middle/bottom of the page. In the skills outline, it clearly states that it is neither a definitive nor a complete list, but it is a helpful summary for setting boundaries of knowledge expectations. Take particular note of the four primary categories of skill expectations. This division of skills is how this book is organized, beginning with Chapter 3, "Data Sources and Ingestion":

- Part II: Design and Implement Data Storage
- Part III: Develop Data Processing
- Part IV: Secure, Monitor, and Optimize Data Storage and Data Processing

If you currently have the same amount of knowledge and experience for each of the four topics, your exam preparation should match the percentage of the exam the topic represents. If this is not the case, then use your own best judgment to decide which areas you need to focus most on.

Design and Implement Data Storage

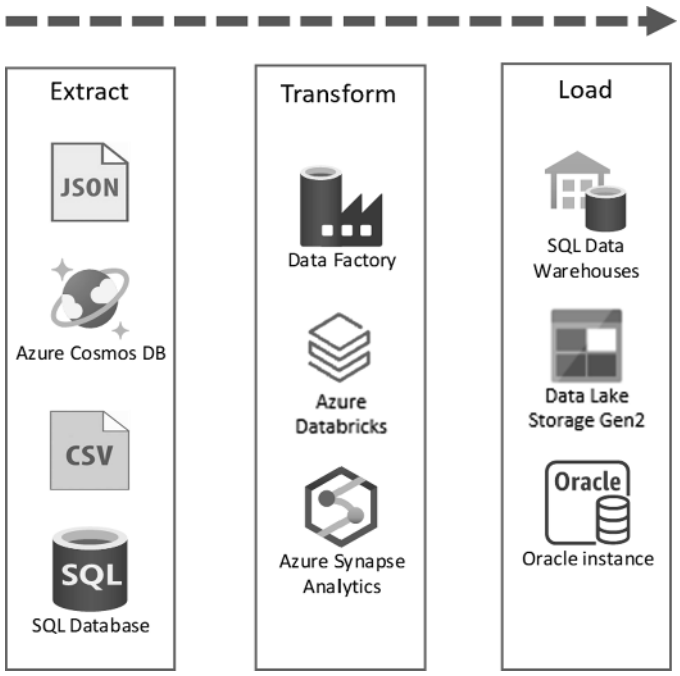
The first topic is *Design and Implement Data Storage*, which as its name implies, involves designing a data storage solution and then implementing that solution in Azure. This topic covers 15 – 20% of the exam content. This topic mostly focuses on the formats in which data can be stored, which kind of storage product is best for each format, and how you place and manage the data on that storage product. Data is stored in many formats, such as documents, blobs, parquet, and relational databases. Some relevant, in-scope services might be *Azure Data Lake Storage (ADLS)*, Azure Blob Storage, or Azure Cosmos DB. Finally, some related terms are *schema*, *changing dimensions*, *temporal data*, and *incremental loading*.

Don't worry if you aren't familiar with any of these formats, products, or concepts. Part II will cover everything you need to know.

Develop Data Processing

The next section is *Develop Data Processing*, which covers 40 - 45% of the exam making it the largest part of the exam. Here, the primary focus is on the transformation of the data that has been ingested and/or stored either on the Azure platform or on-premises. An important term relating to data management is *extract, transform, and load (ETL)*, as illustrated in Figure 1.5.

FIGURE 1.5 The extract, transform, and load (ETL) approach



Data used for analytics is extracted from numerous sources. Each source has a unique format, for example, *JavaScript Object Notation (JSON)* or *comma-separated values (CSV)*. The transformation process occurs when data in different formats are changed from their original source format to a standard destination format. This data transformation is necessary so that intelligence can be gathered from a single location using a standard method, such as *Transact Structured Query Language (T-SQL)*.

There are numerous methods for transforming data. The most common methods occur during either *batching* or *streaming*. A simple definition of *batching* is the incremental

processing of an existing set of data, in bulk. The "batch job" performs, for example, filtering, sorting, aggregation, and deduplication, all of which are based on business requirements. The batch processing logic must reflect the format of the data at the source, the format in which the destination is expecting it, and how to convert the data between the two formats.

The batch processing can occur in near real time, which means the data is transformed very shortly after it is generated, or it can be processed periodically, such as every hour, day, or month. Again, the schedule depends on the purpose of the data and business requirements. This is by no means an easy exercise, and data transformation is most often a distinct scenario for each person or company performing the processing. Azure products like *Azure Data Factory (ADF)*, *Azure Databricks*, and *Azure Synapse Analytics* are useful for handling batch processing of data.

The illustrations in this chapter are meant to introduce you to the relevant Azure data products. The intention is for you to get comfortable with the products and begin to visualize where they fit into an Azure data services solution and strategy. More product details, how the products work together, and their roles are described in detail in the remainder of this book.

Streaming, on the other hand, is not static (Figure 1.6). The data is not processed in bulk and is not pulled from an existing set of data. If you visualize a stream, you most likely think of flowing water moving effortlessly from a source to a destination. The same visualization applies for data streaming in that data is being generated from a source, such as via a *brain computer interface (BCI)*, a weather monitor, or a location tracker. Streamed data is received and transformed, in real time, then stored. That data could be reprocessed again offline via batching, or depending on the streaming logic, the data might be ready for loading into a destination data source for intelligence gathering. *Apache Kafka for HDInsight*, *Azure Stream Analytics*, *Azure Event Hubs*, and *Azure Databricks* are useful products for implementing a highly available and performant streaming solution.

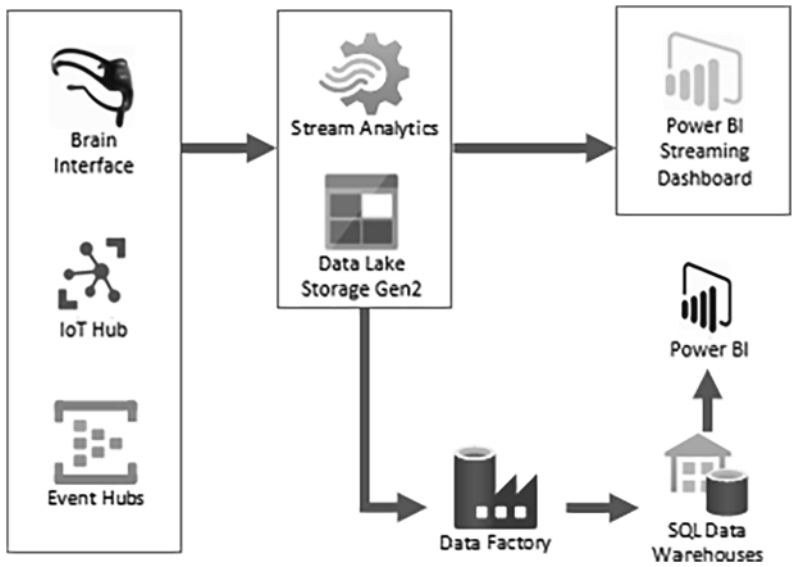
Secure, Monitor, and Optimize Data Storage and Data Processing

Secure, monitor, and optimize data storage and data processing covers —30–35% of the exam. Protecting the data you store on the Azure platform is of extreme importance. That statement is generally accepted but should be acknowledged often so that it remains front and center of all Azure data service project deployments. In many cases, the unintentional exposure of sensitive data has a catastrophic impact on both the company and on the individuals to whom the data belongs. The recovery from such an exposure will be costly, so it is something that must avoided. You can approach the security aspects of Azure from these four perspectives:

- The Azure management portal
- Accessing and securing data

- Protecting passwords and identities
- Privacy and governance

FIGURE 1.6 A data streaming pipeline



The Azure management portal (`https://portal.azure.com`) is where subscription owners, administrators, and contributors build and manage their applications. There are numerous roles and responsibilities within a team that contribute to an IT solution running on the Azure platform. Not all of those roles require access to the cost reports or need the ability to delete an Azure SQL database. *Role-based access control (RBAC)* limits individual access to portal-related and resource management activities. There are numerous components to this concept, as shown in Figure 1.7. A crucial product to achieve this aspect of security is *Azure Active Directory (Azure AD)*.

FIGURE 1.7 Azure portal security product and feature security hierarchy

The Azure AD is where access credentials, group affiliations, and permission information are stored. The privilege to administer the Azure AD should be limited and tightly controlled.

Anyone with administrator access to your company's Azure AD will have access to all your Azure assets contained with it, including access to all user, group, and application information in the Azure AD, plus management rights to some Microsoft 365 services. By default, they will not have access to Azure resources, but any global admin in Azure AD can elevate themselves to the User Access Administrator role. That way, they gain visibility to all Azure resources in subscriptions that are linked to that Azure AD and can then elevate themselves to any role in any of these subscriptions. Protect this privilege by all means necessary.

Azure data products tend to provide more management capabilities via the Azure portal when compared to other compute resources. This means you may need to spend extra time defining what tasks each person on your team requires and then map them to the RBAC roles provided by the Azure data product in the portal. For example, does a single person need access to both the Synapse Apache Spark Administrator role and the Synapse SQL Administrator role? Perhaps, perhaps not, but that decision needs to be made based on requirements.

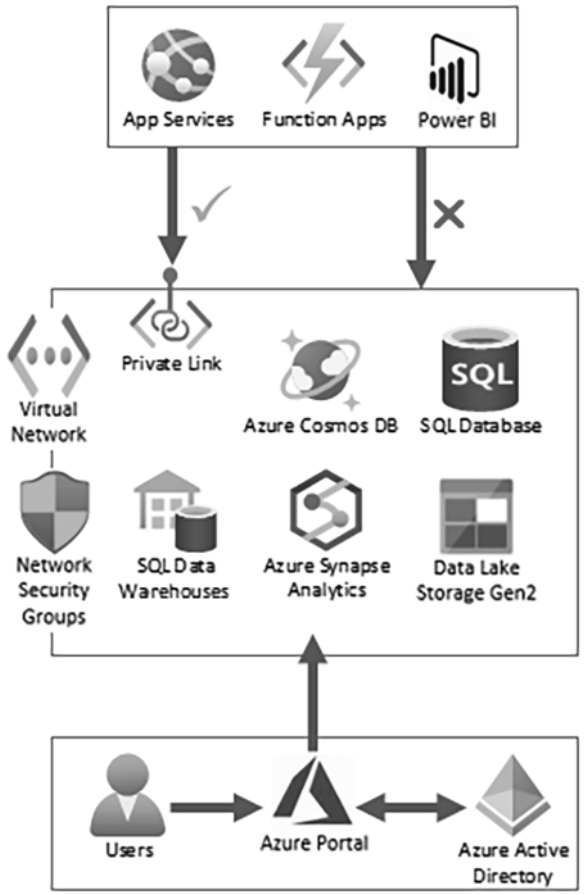
Once the Azure resources are provisioned and configured, accessing and manipulating that data comes into focus. There are two primary methods used to gain access to data. The first one is through an application. In Figure 1.8, an Azure app service, a function app, and Power BI are attempting to access resources that store data. Some products to help secure access are *private links*, *virtual network (VNet)*, and *network security groups (NSGs)*. Many Azure products have a global discoverable endpoint enabled by default. For example, an Azure Cosmos DB has a *Uniform Resource Identifier (URI)* of `https://*.documents.azure.com`, where * is the name of the Azure Cosmos DB account. This might not be a desired scenario; in that case, the solution is to implement a private link that results in the endpoint no longer being globally discoverable. Additionally, implementing a VNet allows you to control the ports and protocols used for accessing the resource that houses your data via NSGs.

The second means of accessing data is illustrated toward the bottom of Figure 1.8. The diagram shows how users, via the Azure portal, can access the resources and the data residing on those resources. This is related to the previous RBAC topic. Two additional concepts that relate to this type of data access are *access control lists (ACLs)* and data encryption. ACLs are used to control access to files existing in a hierarchical directory structure. ACLs are applied within the context of ADLS. Data encryption is focused on how the data is stored and transferred: data-at-rest or data-in-transit. By default, all data on Azure storage products are encrypted by the data-at-rest feature. There are some additional details about how to implement data-in-transit which I'll discuss later in this book, but the main concept with data-in-transit has to do with *Transport Layer Security (TLS)*.

Next, consider the requirements around managing passwords and identities. It is considered bad practice to store user IDs, passwords, and connection strings in application configuration files because anyone who has access to the application code has access to the data. Instead, you store sensitive credentials in an *Azure Key Vault (AKV)*, which can be accessed at runtime to retrieve the necessary credentials to access a database. The identity used to access AKV is known as a *Managed Identity (MI)*. Consider that from an end-user perspective, user identities are managed via Azure AD; however, there is also the concept of a *service*

principle. A service principal, also called MI, is an account that is assigned to a service instead of a human and can only be used with an Azure resource. Figure 1.9 describes the flow where an application is assigned an MI, which is then used to get a connection string from Azure Key Vault, which in turn is used to connect to a datastore.

FIGURE 1.8 An Azure data security diagram with products and features



The last remaining topic for this portion of the exam has to do with privacy and governance. Azure has some of the best tools for complying with industry standards, some of which are shown in Figure 1.10.

FIGURE 1.9 Using Azure Key Vault and MI

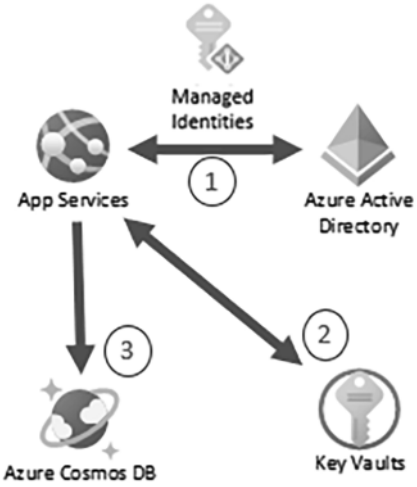
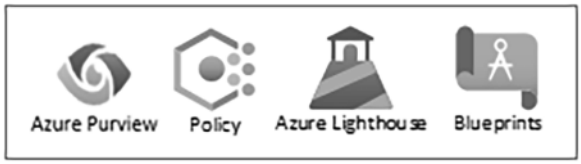
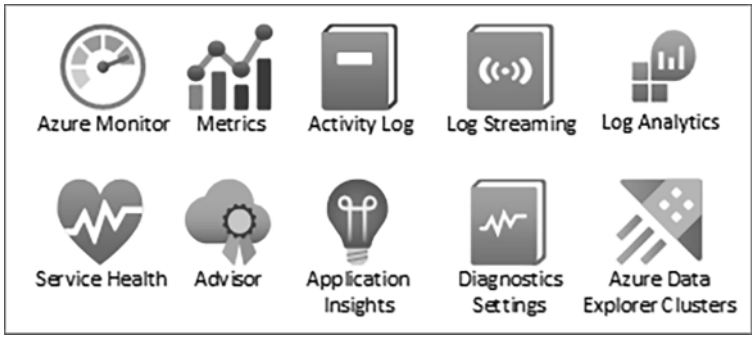


FIGURE 1.10 Azure privacy and governance products



These tools help a company manage the provisioning of Azure products so that they conform to a certain level of security; for example, making sure that any resource with a globally discoverable endpoint prohibits access via an unsecured channel. The privacy and governance features can also walk you through the process of marking the sensitivity levels of your data. Then, you can restrict data based on those levels and monitor who is accessing that data, when, and how often. Companies often overlook this aspect of managing an application running in the cloud. In some cases, there can be legal ramifications and potential loss of a contract if certain criteria are not met. You must be able to identify and use the tools and features available to adhere to these privacy and governance requirements in order to pass the exam. Monitoring and optimizing data storage and data processing are concerned with monitoring, performance tuning, and debugging failed pipeline executions or jobs. Numerous products are available for monitoring the health of your data analytics products, some of which are shown in Figure 1.11. The Azure product listed most prominently in the online skill requirements for the Azure Data Engineer Associate certification is *Azure Monitor*. Without a logging capability of some kind, you'd have trouble identifying the source and extent of compromise in the event of an incident.

FIGURE 1.11 Azure health and monitoring products

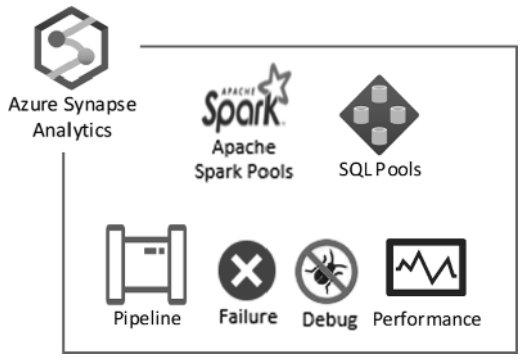


The configuration and alignment of ordered tasks, also called a *pipeline*, is an administrative activity, whereas the actual execution, or carrying out of those tasks, is what manipulates the data. Exceptions can occur during the configuration of a pipeline, but those exceptions wouldn't normally have any impact on the data. Pipeline tasks that manipulate data by performing inserts, updates, and deletes can result in data corruption, exceptions, or failures. For that reason, it is important to have information about the success and failure of each task throughout the pipeline run. That information should be logged into Azure Monitor. These logs are useful for identifying the specific task that caused the unexpected outcome. Using that information, you can review that specific task to find out what happened and implement a fix to prevent it in the future.

Scenarios consisting of failures and exceptions are not the only kind of problem you might face. A pipeline may simply hang, stop unexpectedly, or perform very slow. Capturing and analyzing performance metrics also help to pin down the causes for latent behavior.

Performance metrics, once analyzed, are useful for improving the performance of, for example, *Apache Spark Pools* or *SQL Pools* running within Azure Synapse Analytics (Figure 1.12). Partitioning, indexing, caching, and *user-defined function (UDF)* optimizations can be used to improve the performance of the pipeline or job. Keep in mind that the longer the job runs, the greater the incurred cost since the compute pool is consumed for longer periods of time. Therefore, it is prudent to get the jobs and pipelines to run as quickly and efficiently as possible.

FIGURE 1.12 Azure Synapse pools, performance, and debugging



Having a solid skillset and some applicable experience relating to monitoring, gathering logs, and analyzing performance data and troubleshooting failed Azure Spark jobs and SQL Pool pipeline runs is a necessity to pass the exam. This is achieved by provisioning, consuming, and experiencing the Azure data service products. In this book you will gain this experience through detailed descriptions and extensive discussions covering each key Azure data product and feature. Now that you know the sections of the exam, read on for more tips about not only passing the exam, but how to become a great Azure data engineer.

Use Azure Daily

If you do not have an Azure subscription, this is the first action you need to take. Navigate to `https://azure.microsoft.com` and open a free account. The subscription currently offers $200 of free credits and many other free services for the first 12 months. If you have Azure but you are not actively using it, consider increasing your consumption and exposure to it immediately. You need to have a good understanding of how products and features are organized in the Azure portal. You need significant exposure to the many Azure data services, such as analytics, databases, and storage. The more of those services you regularly work with and the more often you work with them, the better.

The best-case scenario for gaining the Azure Data Engineer Associate certification is that you are already working on a solution running on Azure. Perhaps you were part of the design team that implemented a data solution for your company, or perhaps you are using an existing set of tools to perform analytics. Either way, your chances of passing the exam are increased the more you use the product. There is a term, *catch-22*, that describes a dilemma of being trapped between mutual conflicting circumstances. This can come into play here because what happens if you are motivated to earn this certification to gain the experience but you need the experience to gain the certification? There exists a similar scenario when you're looking for a job that requires three years of experience, but you need a job to get that experience. What you will learn in this book will be helpful in getting you out of the catch-22. Read the chapter contents and perform the exercises to gain some hands-on experience, which is how you learn the most.

Read Azure Articles to Stay Current

To write that the amount and velocity of change in the IT industry is "moving rapidly" would greatly understate reality. The fact is, change is happening constantly in real time. Companies are now able to serve customers 24 hours a day, 7 days a week thanks to, for the most part, globalization. It is always daytime someplace on Earth, and that fact allows a company to strategically place subsidiaries around the globe. With global processes in place, a single task can be passed from location to location and effectively worked on, from beginning to end, without any pause.

The cloud computing sector is no exception and is where most of the change is happening. This is due to two main reasons:

- Cloud computing is "relatively" new.
- Competition for customers is fierce.

Although cloud computing has been around for a few decades, it has only started to kick into high gear in the last few years. Originally, running IT operations in the "cloud" was focused very much on *infrastructure as a service (IaaS)* product offerings. IaaS offers the compute power for running IT operations without having to pay for, set up, or administer the networking or datacenter architecture. Table 1.2 lists popular cloud service offerings.

TABLE 1.2 Popular cloud service offerings

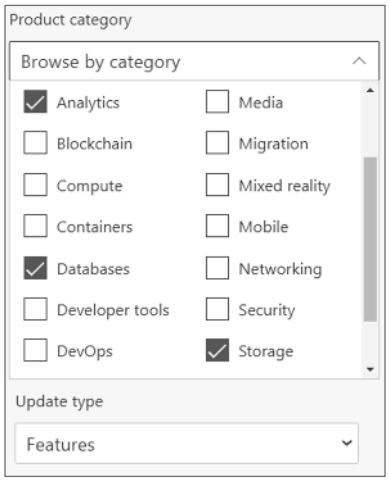
Cloud Service	Acronym	Description
Infrastructure as a service	IaaS	Not responsible for network or datacenter; complete control of the operating system.
Platform as a service	PaaS	IaaS but without control of the operating system
Software as a service	SaaS	Online, subscription-based software
Function as a service	FaaS	PaaS while platform manages compute scaling
Database as a service	DBaaS	PaaS while platform provides DB software, backups, and management
Data analytics as a service	AaaS	SaaS but platform provides infrastructure, storage, and compute for Big Data analytics

The volume and velocity of change in cloud computing has a lot to do with its relative newness. There is so much to create and the demand for it so intense that customers often choose a cloud provider solely based on which one brings a single feature to market first. The cost a company can realize by no longer needing to manage a datacenter drives the consumption of cloud supplier resources and the extinction of the concept of on-premises IT. Here are some ideas you should consider to speed up enhancements and changes within the context of the Azure data engineer associate role and scope.

Consider making it a habit to review official Azure feature updates at `https://azure.microsoft.com/updates`. Filter on product categories such as Analytics, Databases, and Storage. Additionally, as seen on Figure 1.13, restrict the article results to new or updated features only.

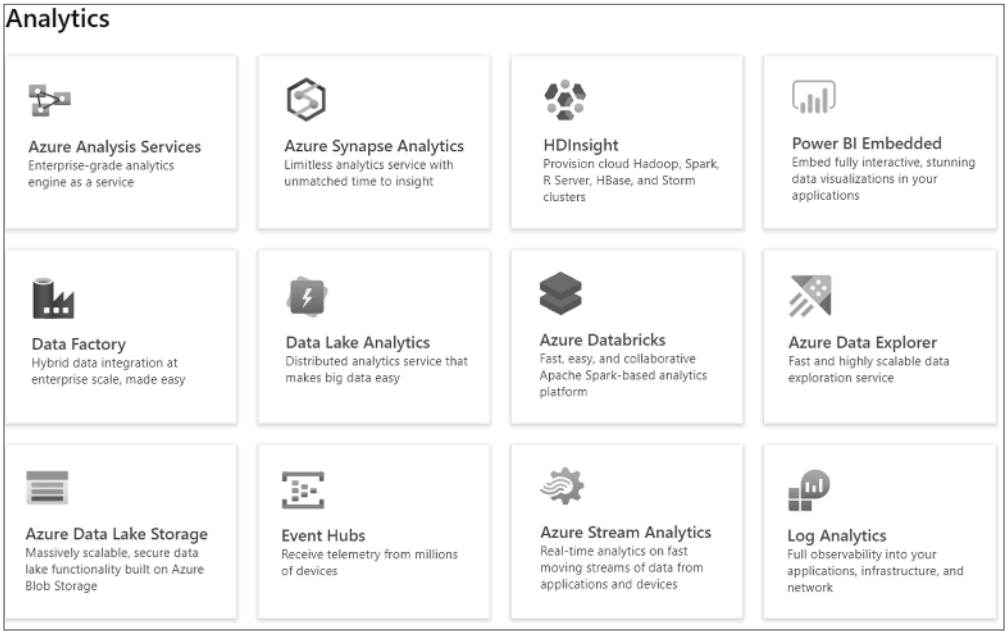
There is another very nice tool, Azure Heat Map (`https://aka.ms/azheatmap`), which is an online graphical representation of all Azure products, features, and much more.

FIGURE 1.13 Azure feature updates



There is also a massive amount of Azure product documentation available here: `https://docs.microsoft.com/azure`. You can search for a specific term or navigate to a product grouping, like `https://docs.microsoft.com/azure/?product=analytics`, which displays all Azure products related to analytics (see Figure 1.14).

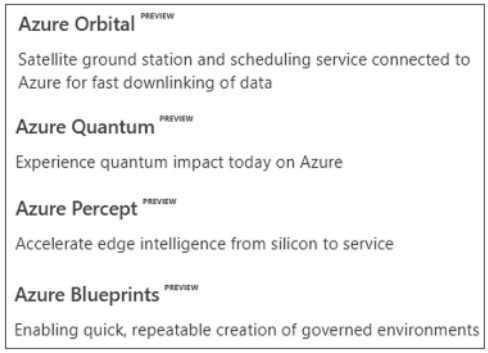
FIGURE 1.14 Azure Analytics product documentation



There are also many skilled and experienced Microsoft employees who post blogs about technical concepts and practices. Many of those who contribute to the Azure blog site are the ones who wrote the code for the product or who actively support the customers of the product. Access the public Azure blogs at `https://azure.microsoft.com/blog`, where you can filter on a date as well as on specific topics.

Actively using the products available on Azure is the best way to stay current with what is happening. Use the tips just provided to keep up on changes and updates to the products you use. Consider paying special attention to Azure products that have the word PREVIEW next to the name. These are new products that are in beta testing and usually contain new features that can make your technical life easier to manage, while also growing the capabilities of your solution running on the platform. Figure 1.15 illustrates how PREVIEW products may appear online.

FIGURE 1.15 Azure products in preview



It is no easy task to perform your day job and keep up with the changes in the products and industry. My final tip is to simply learn a little bit every day. Over the years you will be amazed at how much you have learned with just a little bit of consistent effort.

Have an Understanding of All Azure Products

This book is intended to guide you through the requirements for attaining the Azure Data Engineer Associate certification. The certification primarily focuses on the Azure products directly related to data analytics, storage, security, and streaming, such as the following:

- Azure Synapse Analytics
- Azure Databricks
- Apache Spark

- Azure Data Factory
- Azure Data Lake Storage
- Azure Active Directory
- Azure Key Vault
- Azure Managed Identity
- Azure Stream Analytics
- Azure Event Hubs

You should develop a deep level of knowledge and experience for these products; otherwise, you have a limited chance of passing the exam. However, the exam does sometimes expand into other Azure products from an integration point of view. These products are not directly related to data analytics; rather they are products that are required to implement a data analytics solution on Azure. Given that, you should expand your knowledge of Azure products to include a large majority of them. You do not need to know the technical complexities, in depth, of all Azure products, but you should have a general idea of what the products are used for. Read on for an introduction to many Azure products, all of which you must know to pass the exam.

Azure Product Name Recognition

There are currently over 200 Azure products. Thankfully they are categorized into approximately 20 different categories, which makes it easier to locate products directly related to the solution you need. Analytics, Compute, Databases, Networking, Management and Governance, Security, and Storage are a few of the most popular and most relevant categories. You can find the complete list here: `https://azure.microsoft.com/services`.

The following content provides product summaries and use cases of many Azure products. Products that the exam targets heavily will have additional content in upcoming chapters. The content includes product hands-on exercises and additional descriptions of specific capabilities and limitations in more detail.

Let's get right to it and summarize the Azure products anyone with a desire to become a world-class Azure data engineer should know. Review Table 1.3 for terms and their associated definitions, which are referred to in the following sections. You may need to return to this table if you come across an unfamiliar term.

TABLE 1.3 Technical terms and definitions

Term	Definition
Data lake	A repository of datastored in its natural, unformatted, unmodified, or raw form. Additionally, cleaned and enriched data. Can include structured, semi-structured, unstructured, and binary data. A successor to data mart or data warehouse.
Cluster, pool	Two or more compute machines bound together to collaborate on a shared compute task. See *orchestrator*.
Container	An isolated package of software that contains all application dependencies.
DataFrame	An in-memory compendium of data organized into specified columns. Commonly referred to in the context of Apache Spark.
Data model	An organization of elements in a standardized form. A data model may define a brainwave, which has elements such as frequency and channel. Also called *semantic model*.
Engine, runtime	A combination of source code and compute power that manages the transformation of data.
Ingest	A stage in the data analytics process where data is received from the originating source.
Job, batch job	A unit of work, often executed offline due to its high compute requirements.
Node	A compute machine that performs computations by following coded instructions. Also called *Worker node*.
Notebook	An organizational unit focused on unifying processes, experiments, and deployments. Contains visualizations and runnable commands.
Model and Serve	A stage in the data analytics process where conflated data is stored for the gathering of data intelligence.
Orchestrator	One compute machine, part of the cluster, provisioned to administer the worker nodes also contained in the cluster. See *node*.
Pipeline	A series of tasks that occur in serial succession from beginning to end.
Prep and Train	A stage in the data analytics process where data is pulled from numerous sources. Computer algorithms are then run on the data with the intention of merging the data into a single source.
Workspace	A web page or user interface containing quick, easy access to relevant tools, experiments, and information.

Azure Data Analytics

The products in this section are ones that you absolutely must know and have experience with to pass the exam. Consider the following content an introduction, just so you can get your head around what these products can do. Upcoming chapters provide more product details and hands-on exercises.

An important point to keep in mind as you read through these Azure data analytics services is that Azure Databricks, Azure HDInsight, and Azure Analysis Services are cloud-based implementations of Databricks, HDInsight, and Analysis Services. The latter three products are very popular data analytics services that run in on-premises datacenters. Therefore, if you are currently using one of them already and you want to migrate to the cloud, you can choose the one that you are already using and easily move your existing workloads to the cloud. Remember that when you move your workload to the cloud, you are outsourcing the infrastructure and platform management to the cloud service provider. This means you can focus all your efforts on data analytics, business insights, and *business intelligence (BI)* gathering.

It is important to know which of these objectives are in scope so that you can choose the correct Azure data analytics product:

- You want to create a new data analytics project that will run on Azure.
- You want to redesign an existing on-premises data analytics project and run it on Azure.
- You want to migrate an existing data analytics project to Azure.

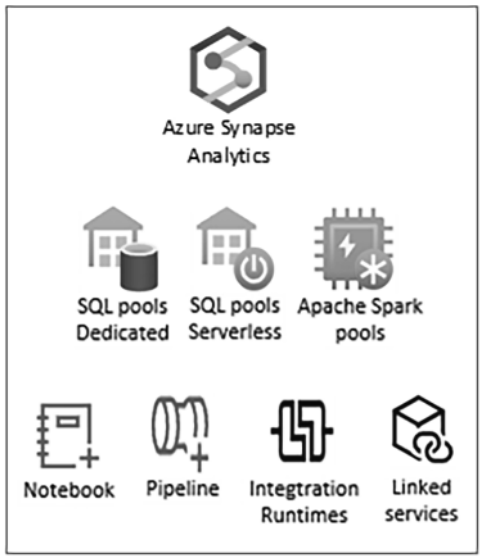
If your objective is either of the first two, then those projects should target Azure Synapse Analytics. This workspace is where Microsoft is building connectivity and configuration capabilities to the other popular data analytics products. Otherwise, the Azure platform supports an implementation of the most popular data analytics frameworks. As you read through these first descriptions, how best to target a creation or migration will become clearer because you will learn which features are provided by each product.

Azure Synapse Analytics

There is a term used to describe projects that fit nicely into the use case for Azure Synapse Analytics (ASA). The term is *greenfield*. A *greenfield project* means exactly what you might visualize if you close your eyes and think of a green field of grass. It's open, no fences, no constraints, a foundation that will support any idea, bound only by the limits of your imagination. Bringing it back into context, Azure Synapse Analytics is the workspace you need when you are starting a new project that will run on Azure. This isn't to say you can't migrate existing data analytics projects; it depends on the size, scale, and maturity of the project. Preexisting analytics solutions with limited complexity and already existing solutions with relatively small datasets might also be a good fit for Azure Synapse Analytics. Note that Azure Synapse Analytics is the product Microsoft wants all new data analytics customers to use. You should therefore expect many questions about it on the exam.

Azure Synapse Analytics includes a workspace management console as well as other Azure data service offerings, including pipelines and SQL pools. Figure 1.16 illustrates some of those services. Azure Synapse Analytics can also be integrated with many other Azure data services like Cosmos DB and Azure SQL. A very important feature is that you can run your analytics on either SQL pools or Apache Spark pools using Azure Synapse Analytics.

FIGURE 1.16 Azure Synapse Analytics services



If you need to perform analytics with SQL-based analytics, then use a SQL pool. If Scala, PySpark, or Spark SQL is required, then an Apache Spark pool would be your choice. The next several sections will introduce these and other important terms related to Azure Synapse Analytics.

SQL Pool

For individuals or companies that typically run SQL Server or other relational database solutions, the SQL pool makes the most sense. This pool provides two options: serverless and dedicated. A serverless SQL pool ensures that enough compute resources (CPU and memory) will be available, as required, on-demand, in real time. Once that compute is no longer required, it is deallocated so that the cost is optimized, and you are billed for only what you use. The other option is a dedicated SQL pool. Although there is scaling, you typically choose a compute size targeting a predetermined number of CPUs and memory. When more capacity is required, another instance (aka node) of that same size of compute (the same number of CPUs and amount of memory) is added to the pool and used.

Apache Spark Pool

Apache Spark is explored in the context of both Azure Databricks and Azure HDInsight in the next two sections. One benefit of running Apache Spark pools via Azure Synapse Analytics is that this approach uses Azure Data Lake Storage Gen2 as the primary filesystem, whereas the other products mentioned here do not fully support Gen2 in that capacity. Some of the frameworks provided by default are Spark SQL, GraphX, MLlib, Anaconda, and Apache Livy. There are numerous others; additional frameworks are added on a regular basis. It is the intent that all capabilities found in other Azure data analytics products that offer Apache Spark have the same features and capabilities in Azure Synapse Analytics. If there are any notable missing features, they will be called out in later chapters.

Pipelines

Table 1.3 offered a brief definition of a pipeline. From an Azure Synapse Analytics perspective, the definition is the same in that a pipeline groups activities together to perform a task. The most common type of pipeline in this context has to do with data integration. Data integration involves pulling data from numerous sources, manipulating it into a similar form, and storing that final data in a place for intelligence gathering or machine learning. A few related terms here are SQL scripts, Notebook, and Apache Spark job. Those terms are units where logic is organized and strung together to perform specific activities that execute during a pipeline run.

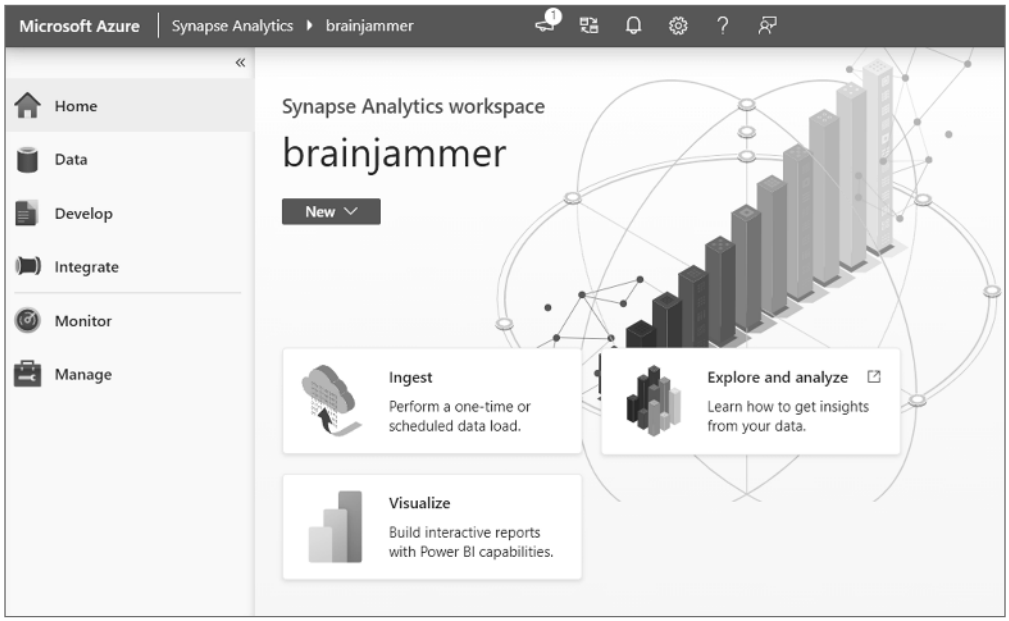
Integration Runtimes

There is also a brief description of a runtime in Table 1.3. In the context of integration runtimes (IRs), it is simply some provisioned compute power to manage and perform tasks and activities configured within a pipeline. There are three types of integration runtimes: Azure, self-hosted, and Azure-SSIS (SQL Server Integration Services). Only Azure and self-hosted are currently supported in Azure Synapse Analytics. If you require Azure-SSIS, then you need to use Azure Data Factory (ADF) discussed later in this book. The Azure integration runtime uses compute resources that exist on Azure, whereas self-hosted compute would come from hardware hosted on-premises. Think of the integration runtime as a compute machine that manages the movement of data between datastores that may exist either on Azure or within an on-premises private network. Another responsibility of the IR is to trigger and monitor the status of data transformations that run on your provisioned SQL or Apache Spark pools.

Linked Services

A *linked service* allows you to connect to several external sources using a connection string. This is necessary if you want to pull the data from those sources and integrate it into your data model. This feature is accessible from the Manage menu in the Synapse Studio, as shown in Figure 1.17.

FIGURE 1.17 Azure Synapse Analytics Studio



Examples of the kinds of resource that can be configured using this service are Azure Cosmos DB, Azure Data Explorer, Azure SQL Database, Azure Blob Storage, and resources hosted at competing cloud service providers like Amazon. There are well over 30 possibilities, with more being added regularly.

Azure Databricks

Azure Databricks is an implementation of the Databricks data analytics platform optimized to run on the Azure platform. If your organization is currently using Databricks to perform data analytics on-premises and is considering migrating that workload to the cloud, this is the place to begin your investigation. A significant benefit of using Azure Databricks is the elimination of complexities required to manage the infrastructure necessary to operate an on-premises version of Databricks. Azure Databricks abstracts away that complexity, which frees up technical resources, thus allowing them to focus on data analytics instead of infrastructure management. Although the Azure Databricks platform can be useful for managing all phases of the data analytics process, its most foremost effect concerns the Prep and Train phase.

The Azure Databricks offering is divided into three environments:

- Databricks Data Science and Engineering
- Databricks Machine Learning
- Databricks SQL

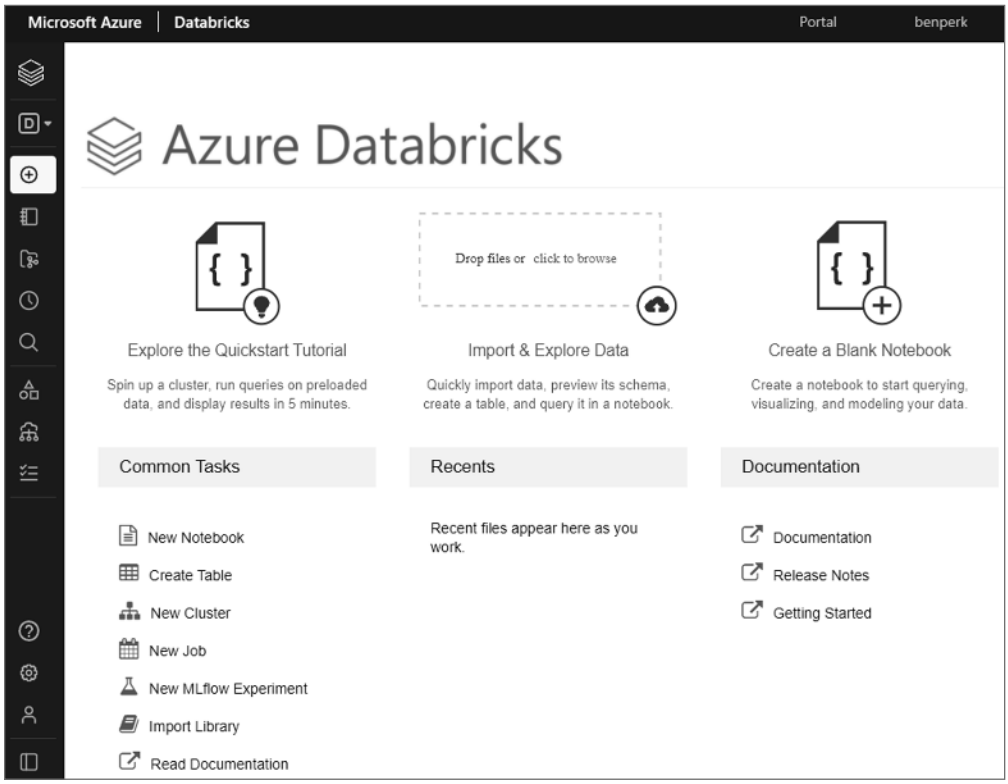
The default workspace environment is Data Science and Engineering, as shown in Figure 1.18. The workspace provides access to common tasks such as creating notebooks, clusters, jobs, and tables as well as seamless integration with Azure storage and security products.

The most popular technologies commonly associated with Databricks that are integrated with Azure Databricks are the following:

- Apache Spark
- Delta Lake
- MLflow
- PyTorch

They consist of a runtime, a datastore, and frameworks that are not limited to only Azure Databricks. These products can stand alone and be used from a multitude of workspaces and products.

FIGURE 1.18 Azure Databricks workspace



Apache Spark

A runtime is where the work happens; it is where the compute power is used the most. Azure Synapse Analytics Apache Spark pools represent a cluster of computers ready to perform the task at hand. The task is the merging of data stored in different locations and formats into a single construct. The merged data is then placed in a datastore for gaining insights in real time or later. The feature that sets Apache Spark apart from other similar products is the support for in-memory data processing. Performing high amounts of I/O, such as writing to disk, incurs unnecessary latency. Instead, the data is stored in-memory using a *DataFrame*.

Delta Lake

Delta Lake is a layer of capabilities, accessible using languages such as Python, R, Scala, or SQL to query and manipulate data pulled from your data lake. Delta lake tables are built on top of the *Azure Databricks File System (DBFS)*, which is useful for versioning data, managing indexes, and capturing statistics. Using that information, Delta Lake can deliver those capabilities in a highly performant, reliable, and secure manner. Consider a scenario in which data in your Data Lake is in both JSON and CSV formats. You can use a Delta Lake parameter to transform that data into delta format and place it into a DataFrame for further refining and aggregation. The following code snippet illustrates how to achieve that:

```
dataframe.write.format("delta").load("/brainjammer/meditation/*")
```

MLflow and MLlib

MLflow is an open source platform for managing the machine learning lifecycle. A lifecycle is a single or a series of experiments that results in a graphical visualization of the result of each run. There also exists the ability to compare and search results of numerous runs. MLlib is a scalable machine learning library for Azure Spark. Here is an example of some code that shows how to use these tools:

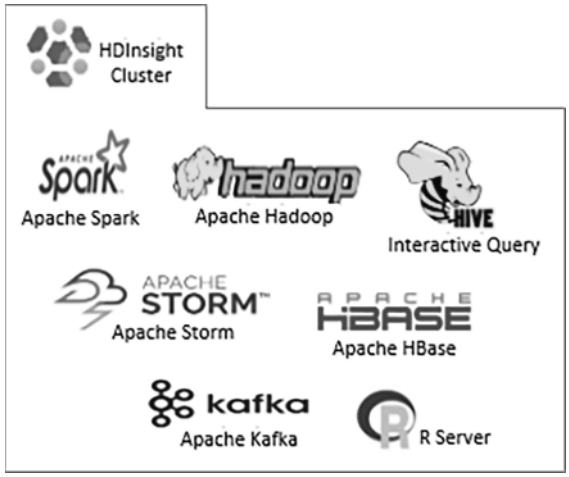
```
import mlflow.spark
df = spark.read.format("json") \
 .load("hdfs://brainjammer/classicalmusic/session101.json")
mlflow.spark.autolog()
with mlflow.start_run()
```

The code snippet is a summary; additional syntax is required to import libraries and complete machine learning runs.

Azure HDInsight

Azure HDInsight is a cloud installation of Hadoop components. If your company is already running on-premises versions of any open source framework shown in Figure 1.19, then Azure HDInsight is the product to provision on Azure.

FIGURE 1.19 Azure HDInsight most popular supported open source frameworks



The benefits of migrating your on-premises workloads to the cloud include scaling and outsourcing infrastructure management. Here are a few additional benefits you can expect from running your Hadoop component on Azure:

- HDInsight provides an end-to-end *service level agreement (SLA)*.
- Data engineers and scientists can use popular notebooks such as Jupyter and Zeppelin.
- Supported development tools include Visual Studio, Visual Studio Code, Eclipse, and .NET, IntelliJ, Java, R, and Python.
- There is seamless integration with Azure Monitor, Azure Active Directory, and Azure Virtual Networks.

The provisioning of these frameworks was achieved by first creating an HDInsight cluster. This can be performed using the Azure portal or Azure Data Factory and managed using Azure CLI, the .NET SDK, or PowerShell. Part of the process of creating a HDInsight cluster is selecting the cluster type. The various types of clusters are described in the following subsections and use the descriptions provided in the Azure portal.

Hadoop Petabyte-scale processing with Hadoop components like MapReduce, Hive (SQL on Hadoop), Pig, Sqoop, and Oozie.

Spark Fast data analytics and cluster computing using in-memory processing.

Kafka Lets you build a high throughput, low-latency, real-time streaming platform using a fast, scalable, durable, and fault-tolerant publish-subscribe messaging system. The Azure alternative to this is Azure Stream Analytics, discussed later in this chapter.

HBase Fast and scalable NoSQL database. Available with both standard and premium (SSD) storage options.

Interactive Query Build enterprise data warehouses with in-memory analytics using Hive (SQL on Hadoop) and low-latency analytical processing (LLAP). Note that this feature requires high memory instances.

Storm Reliably processes infinite streams of data in real time.

R Server Lets you analyze data at scale, build intelligent apps, and discover valuable insights across your business using both R and Python.

See Table 1.3 for descriptions of the terms cluster, orchestrator, and nodes. Once the cluster is provisioned, you can then decide how many nodes you need to perform the required work. You can do so manually by using Azure CLI, as the following snippet illustrates, or from within the Azure portal:

```
az hdinsight resize --resource-group <RESOURCEGROUP>`
 --name <CLUSTERNAME>`
 --workernode-count <NEWSIZE>
```

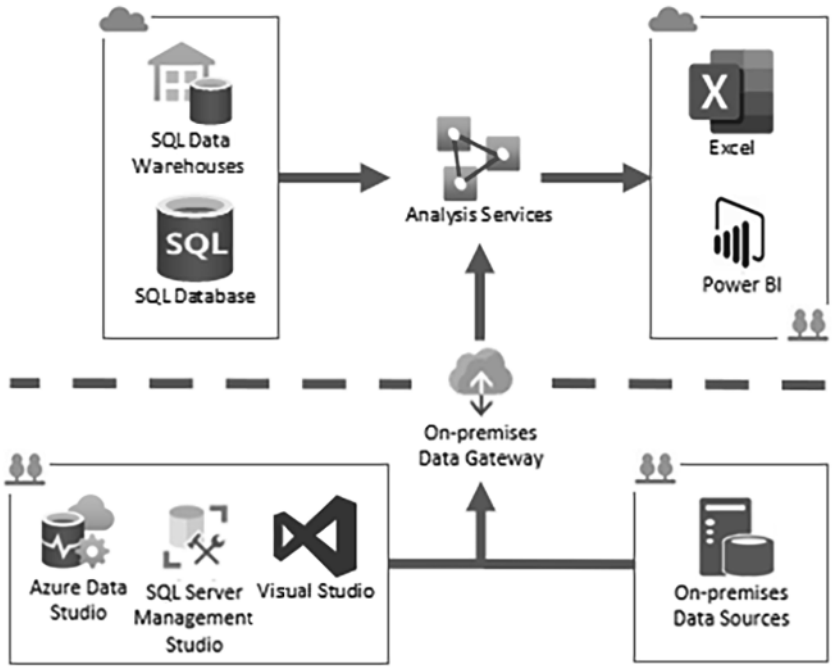
It is also possible to scale automatically based on CPU, memory, and number of containers on a node. Autoscaling is the most effective and efficient means for running your cluster. When the code in your container requires more compute, the platform adds more compute power to scale out. When that compute power is no longer needed, it scales in, which results in you being charged only for the compute power required to run your workloads.

Azure Analysis Services

For a company that already has an on-premises installation of SQL Server Analysis Services, Azure Analysis Services is the cloud implementation of this. Azure Analysis Services provides access to all the data you have in your on-premises databases and data warehouses, using a gateway, as well as any data source running on Azure. Figure 1.20 illustrates how the access is realized.

Azure Analysis Services is a PaaS cloud service offering, which means the magical aspect of this product is its scalability. As with most PaaS offerings, the product delivers numerous tiers to meet your needs based on *query processing units (QPU)* and memory. The tiers range from the Developer tier, which offers 20 QPUs and 3 GB of memory, up to over 1,000 QPUs and hundreds of gigabytes of memory. A QPU is the means for measuring how much query and data processing is used for performing computations. Instead of binding your compute requirements to a *central processing unit (CPU)*, which is a common approach, the QPU approach focuses more on the data processing needs versus the need to know how many CPUs are required to perform the work. Additionally, running on PaaS means that as your data grows you do not have to be concerned about buying, installing, configuring, and supporting extra hardware to store it; all of that comes at part of the PaaS offering.

FIGURE 1.20 Azure Analysis Services

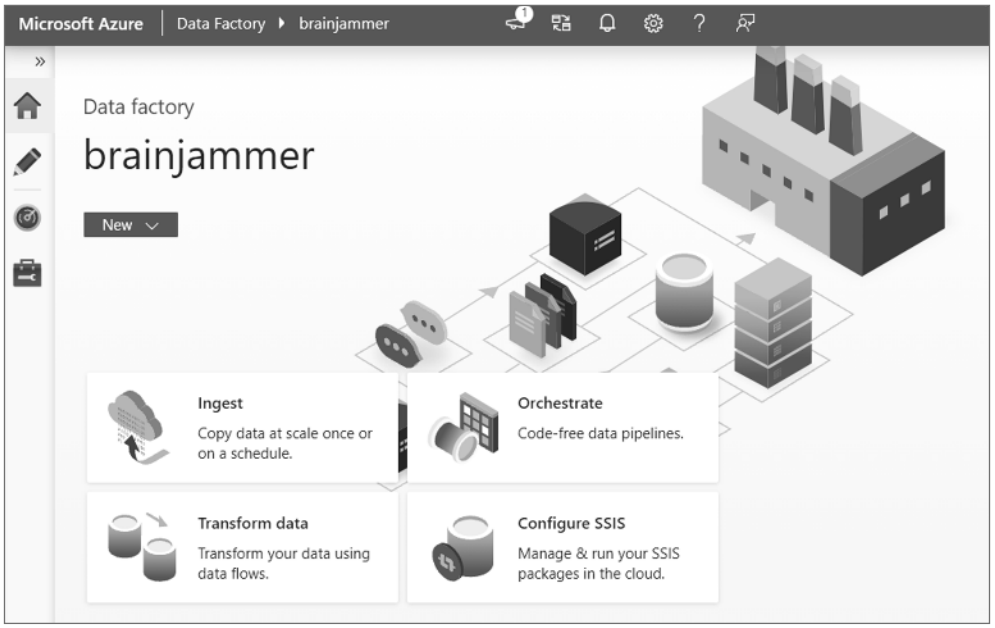


From a usability perspective *SQL Server Management Studio (SSMS)*, *SQL Server Data Tools (SSDT)*, Microsoft Excel, and Power BI all function the same as you would expect them to when working with SQL Server Analysis Services. From a performance perspective, Azure Analysis Services improves the performance of queries with an in-memory caching feature. This means that instead of making a call all the way to the data source again and again, the result of like queries can be stored in a cache, which reduces cost and latency. Another benefit of Azure Analysis Services is that you can continue using the existing business knowledge, business models, and business logic that you have invested much time and energy into tuning and perfecting. You can deploy these models to Azure using Visual Studio, for example, by changing the data source from your on-premises to Azure and choosing Deploy from the Solution pop-up menu.

Azure Data Factory

Azure Data Factory (ADF) is a data integration service accessible via Azure Data Factory Studio. Figure 1.21 shows an example of how that looks. You might recognize some similarities between it and Azure Synapse Analytics Studio, discussed earlier and shown in Figure 1.17.

FIGURE 1.21 Azure Data Factory Studio



As you saw in the earlier section "Azure Synapse Analytics," it is the service you should use if you are building new data analytic projects and workspaces on Azure. Azure Data Factory existed prior to ASA and there are many customers who currently use ADF. Many of the features of ASA are similar to those in ADF, such as integration runtimes, linked services, and pipelines, although there are some differences in the process of creating those features between the two studios. This will become more obvious later in this chapter when you work through the exercises and see for yourself.

ADF contains mainly integration capabilities, though Dataflows has expanded it a bit more to the analytics side. ASA includes these capabilities (except Azure-SSIS IR, as mentioned later) and also full analytics capabilities with SQL pools and Apache Spark pools. ADF corresponds to the pipeline capability and associated components of ASA, plus improved integrations with several data services like Cosmos DB and Azure SQL.

ADF does not have integration with SQL or Apache Spark pools and therefore no notebooks, no Spark job definitions, and SQL pool stored procedure activities. Keep in mind these two important points:

- New features will be released to ASA Studio, meaning that ADF will slowly (over many years) decline into oblivion. If you are using ADF, consider moving your workspaces to ASA as soon as possible.
- Azure-SSIS integration runtimes currently only exist in ADF, and there hasn't been any mention about whether this capability will be ported to ASA.

Let's take a closer look at Azure-SSIS.

Azure-SSIS

SQL Service Integrated Services (SSIS) is described in greater detail later in the "Azure Databases" section. You can apply similarities to an SSIS package as you can to a pipeline. A package, like a pipeline, contains a series of steps, tasks, and activities that perform copying, transforming, and storing of the modeled data. An Azure-SSIS is an IR, which means that it is a compute machine that is allocated a CPU and memory. Azure-SSIS lets customers lift and shift existing SSIS packages onto Azure compute resources.

Azure Event Hubs

Azure Event Hubs is an endpoint for ingesting data at high-scale frequencies. There are numerous ways to move data into a datastore, such as by creating a connection to a database via code running on a website or within a client application. The problem with that approach is that databases typically have a limit on the number of concurrent connections that can be opened. When that limit is hit, clients that need a connection must wait, and the wait period has a timeout value. If the timeout is breached, then the connection is broken and the data possibly lost. The scenario in which a client connects directly to a database to store data is called a *coupled* or *real-time* solution. The alternative to a coupled solution is a *decoupled* solution.

Placing a service—for example, Azure Event Hubs—between a client and the database decouples the transaction involving those two entities. Think of Azure Event Hubs as a messaging queue, where the information that the client would normally send directly to a database instead places it into a queue. Once the message is in the queue, the client no longer has a connection and can continue with other work. The insertion into the queue (aka a *trigger*) notifies another resource, such as an Azure Function, to pull that data from the queue and perform the processing and insertion into the database. This decoupling reduces the chance of losing data and lowers the chance of experiencing an overload on the datastore side. Azure Event Hub is designed primarily for Big Data streaming scenarios that load billions of requests—for example, logging every stock trade happening globally per day. Event Hubs are also commonly used as a decoupler for Azure Databricks, Stream Analytics, Azure Data Lake Storage, or HDInsight products that are used to process the data to gather intelligence.

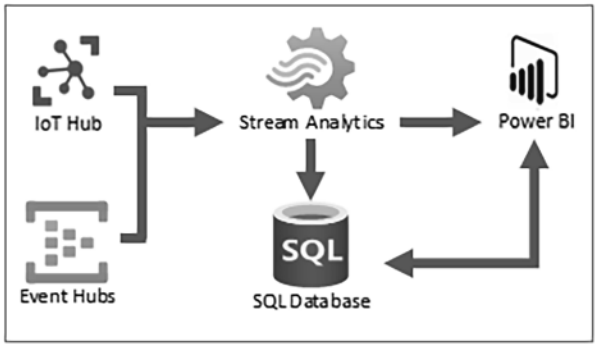
IoT Hub

IoT Hub is like Event Hub in that it is used to ingest large amounts of data with high reliability and low latency but the target data producers are Internet of Things (IoT) devices, such as weather readers, automobile trackers, brain–computer interfaces (BCIs), or streetlamps, to name a few. One option that is available via IoT Hub is the ability to send notifications to the IoT device in addition to receiving data from it. Consider the streetlamp that needs to be turned on at a certain time of day. IoT Hub can be configured to send that signal to the lamp at a certain time per day. The technology to achieve cloud-to-device messaging is called *WebSockets*. The primary difference between Event Hub and IoT Hub has to do with the number of messages received within a given time frame and the kind of message producer sending the message.

Azure Stream Analytics

Azure Stream Analytics can be used to analyze data flowing in from an IoT device or from any other numerous kinds of message producers. Event Hubs and IoT Hubs are what you use to ingest those messages at scale. Those messages are placed into a queue and then processed by a consumer that has subscribed to the hub. In other words, a binding exists between the queue where the messages arrive and the consumer that will process that message. When the message arrives at the queue, a notification that includes metadata is sent to the consumer. Then the consumer takes an action on that message. As you can see in Figure 1.22, a binding can exist between either an Event Hub or an IoT Hub and Azure Stream Analytics.

FIGURE 1.22 Azure Stream Analytics data flow



Azure Stream Analytics uses SQL query syntax to perform analysis of the streaming data, something like the following:

```
SELECT *
INTO Output
FROM Input
WHERE AF3Alpha BETWEEN 11 AND 15
```

Once the data is transformed, it can be streamed in real time to a Power BI dashboard or to a SQL Server database for offline or near-real-time reporting. Remember that ASA has the word "stream" in its name. If your solution requires a product to perform analytics on a stream of data in real time, this is the product you would choose. Alternatively, if your solution is event-driven or timer-based, then consider choosing Azure Functions or Logic Apps instead.

Other Products

There are other products related to data analytics that are available on the Azure platform. It is possible that the exam will ask a question about these products, but don't expect many. Regardless, if your desire is to be the best Azure data engineer possible, then the more you know, the better.

Anomaly Detector

Anomaly Detector is an *application programming interface (API)* used to find irregularities in data. Consider that a normal reading of an alpha brain wave is typically between 5.311 and 12.541 Hz. That range is determined by running data analytics on very large sets of recordings. If there is a reading outside of that range, it can be highlighted as an anomaly. Highlighting it ensures that the person who monitors such activities will be notified and take appropriate actions. Anomaly Detector can be used in real time, offline, or near real time.

Data Science Virtual Machines

A *Data Science Virtual Machine (DSVM)* is an Azure VM that comes preconfigured with lots of standard data sciences–related products. Instead of provisioning a default Azure VM, which would have nothing but the operating system on it, you would find the following, and more, installed by default:

- CUDA, Horovod, PyTorch, TensorFlow
- Python, R, C#, Node, Java
- Visual Studio, RStudio, PyCharm
- H2O, LightGBM, Rattle
- Apache Spark, SQL Server
- Azure CLI, AzCopy, Azure Storage Explorer

Azure Data Lake Analytics

Azure Data Lake Analytics is used to run on-demand data analytic jobs in parallel. The parallelism is achieved using Microsoft Dryad, which can compute data represented in *directed acyclic graphs (DAGs)*. A DAG is a model helpful in the calculation of the "traveling salesman" scenario, in which there can be sequential proposed directions that never form a complete loop. Behind the scenes, there is an implementation of Apache Hadoop, which uses Apache YARN to manage the resources across clusters. Azure Data Lake Analytics also support the U-SQL syntax, which combines SQL with C#. An example of U-SQL is shown in the following snippet:

```
@alphareading =
 EXTRACT
 AF3Alpha     decimal,
 T7Alpha      decimal,
```

```
 PzAlpha      decimal,
 T8Alpha      decimal,
 AF4Alpha     decimal
 FROM "/brainjammer/playingguitar/reading001.tsv"
 USING Extractors.Tsv();
OUTPUT @alphareading
 TO "/output/brainjammer/playingguitar.csv"
 USING Outputters.csv();
```

Note also that Azure Data Lake Analytics requires the Gen1 version of Azure Data Lake Storage; Gen2 is the most current one. If you need these options, then choose this product; otherwise, choose Azure Synapse Analytics to perform your data analytics on Azure since Azure Data Lake Analytics is retiring in the near future.

Power BI/Power BI Embedded

Power BI is a tool used to visualize data. The Power BI desktop application supports connecting to numerous datastores. Once connected, many visualizations can then be applied to the data. The following is a list of a few of the built-in visualizations:

- Area charts
- Stacked column charts
- Maps
- Matrices
- Key performance indicators

The data accessed from Power BI has typically already been through the data analytics and data modeling process. The result of those activities is then visualized using this product. Reporting and the creation of dashboards are also common uses for Power BI.

Power BI Embedded is an online SaaS service and is a means of delivering customer-facing analytics, reports, and dashboards via a web application or website. The benefit is that everyone who wants or needs to consume the results of your data analytics solutions is not required to have a Power BI license. Instead, the results can be placed online and accessed without any software installation requirements.

Azure Storage Products

Azure Storage is a group of products that provide a secure, scalable, and highly available solution for storing your data. This product grouping offers numerous capabilities and also serves as a place to store blobs and files rather than storing rows and columns of data into a *database management system (DBMS)*. An Azure Storage account provides a NoSQL store

and a messaging queue, both of which have higher scale alternatives. Read on to learn about each Azure Storage feature and its purpose and possible alternatives.

Azure Data Lake Storage

Azure Data Lake Storage (ADLS) is a fundamental piece of most enterprise data analytics solutions running on Azure. This product is optimized for Big Data analytics workloads. ADLS accomplishes this by providing storage capacity of up to multiple exabytes of data and supplying access to that data at a throughput of hundreds of gigabytes per second. ADLS Gen2 supports the open source platforms described in Table 1.4.

TABLE 1.4 ADLS-supported platforms

Platform	Supported version
Azure Databricks	5.1+
Cloudera	6.1+
Hadoop	3.2+
HDInsight	3.6+
Hortonworks	3.1.x+

ADLS Gen2 can also be easily integrated with many Azure products, such as Azure Data Factory, Azure Event Hub, Azure Machine Learning, Azure Stream Analytics, IoT Hub, Power BI, and Azure SQL databases. Additional information and capabilities include the following:

- Gen1 vs. Gen2
- Hadoop Distributed File System (HDFS)
- ACL and POSIX security model
- Hierarchical namespaces

Gen1 vs. Gen2

ADLS Gen1 will be retired as of February 29, 2024. Therefore, we don't recommended that you build any new solutions on that version. As mentioned earlier, Azure Data Lake Analytics uses Gen1; therefore, we also don't recommended building new data analytics solutions with that product either. ADLS Gen2 supports all the capabilities that exist in ADLS Gen1. The significant change is that Gen2 is now aligned with and built on Azure Blob

Storage. Building on top of Azure Blob Storage (described later) makes ADLS Gen2 more cost effective and provides diagnostic logging capabilities and access tiers.

Hadoop Distributed File System

If you have used HDFS in the past, you can expect the same experience when using ADLS. This has to do with how you and the operating system interact with data files. Reading, writing, copying, renaming, and deleting are most of the activities you would expect to be able to perform. The *Azure Blob Filesystem (ABFS) driver* is available on all Apache Hadoop environments such as Azure Synapse Analytics, Azure Databricks, and Azure HDInsight. ABFS has some major performance improvements over the previous *Windows Azure Storage Blob (WASB) driver* when it comes to renaming and deleting files. Examples of HDFS commands to create a directory, to copy data from local storage to a cluster, and to list the contents of a directory are shown here:

```
hdfs dfs -mkdir /brainjammer/
hdfs dfs -copyFromLocal meditation.json /brainjammer/
hdfs dfs -ls /brainjammer/
```

ACL and POSIX Security Model

When you're securing files and folders, the access permission level (aka Access ACL) is critical. The access permission level has to do with what actions a person or entity can perform on those files or folders. Table 1.5 describes those permissions.

TABLE 1.5 File/folder access permission levels

Permission	File permission	Directory permission
Execute (X)	No meaning in ADLS Gen2	Required to navigate child items in directory
Read (R)	Read contents of a file	R and X required to list directory contents
Write (W)	Write or append to a file	W and X required to create directory items

There is another kind of access control that pertains to ACLs and ADLS Gen2 called *Default ACLs*. An important distinction between Access ACLs and Default ACLs is that Default ACLs are associated with directories, not files. When you set an ACL on a directory, new files created within it take on the permission granted to the directory. A very important point to note is that if you change the Default ACL, it does not change the permissions on the files within it. The new ACL is only applied to files and directories created after the change. An additional step is required to apply those changes recursively.

Some tools you can use to set ACL permissions on files and directories hosted in ADLS Gen2 are

- Azure Storage Explorer
- Azure Portal
- Azure CLI, PowerShell
- Python, Java, .NET SDKs

Most distributions of Linux support *portable operating system interface (POSIX)* ACLs, which is a means for setting permissions on files and directories on that OS type. That activity is typically performed through a command window; the following code snippet shows a command and the result:

```
$ mkdir brainjammer
$ ls -la brainjammer
$ drwxr-xr-x 3 csharpguitar root 60 2022-05-02 16:19 ..
$ drwxr-xr-x 2 csharpguitar user 80 2022-05-02 16:52 brainjammer
$ -rw-r--r-- 1 csharpguitar user 131 2022-05-02 18:07 playingguitar.json
```

A directory named `brainjammer` is created using the `mkdir` command. The `ls` command lists the attributes of the directory and any other object within it. Notice that user `csharpguitar` has full access to the directory `brainjammer` and the groups `root` and `user` have read, execute, and read, respectively. Access can be further clarified by reading the contents in this list. We will break down the fourth line of the snippet `drwxr-xr-x` into its parts:

- `d` stands for directory.
- `rwx` means the user and group have read (r), write (w), and execute (x) permissions on that directory.
- `r-x` means the user and group have read (r) and execute (x) permissions.
- `r--` means "other" entities that are neither that user nor a user in that group.

Hierarchical Namespace

Hierarchical namespaces organize directories and blob files the same way the filesystem does on your personal computer. There exists a nested hierarchy of directories, each of which can include a file of some type. Hierarchical namespace keeps your data logically structured and by doing so renders better storage and retrieval performance. The following is an example of a hierarchical namespace on a filesystem:

```
brainjammer\
brainjammer\sessionjson\
brainjammer\sessionjson\metalmusic\
brainjammer\sessionjson\metalmusic\pow\
brainjammer\sessionjson\metalmusic\pow\brainjammer-pow-0904.json
brainjammer\sessionjson\worknoemail\pow\brainjammer-pow-1855.json
```

Object stores are typically flat, which results in less efficient behavior when moving, renaming, or deleting directories. Stating that a flat file structure is less efficient is

understating a bit; it is actually quite dramatic. When it comes to performing data analytics, it is common for part of the analysis process to copy, move, create, and delete files. In many cases, the time required to achieve such actions using flat structures can take longer than the analysis process itself. Running on Azure, you are charged by the amount of time you consume the product, so if this time can be reduced because your files are stored more optimally, then you save money. Using the hierarchical namespaces provided via ADLS Gen2 reduces the latency and therefore reduces costs.

Azure Storage

Azure Storage is a suite of modern storage solutions, all of which are massively scalable, secure, reliant, and accessible. When you need one of the offered Azure Storage services, you create what is called an Azure Storage Account. The storage account is where you can choose certain attributes that apply to all the services within it—for example, performance and redundancy details.

Performance

An Azure Storage Account has two performance options:

- *Standard* is recommended for most scenarios and is referred to as a general-purpose v2 account. When you visualize storage, a hard drive might pop into your mind. What you have in mind is likely a *hard disk drive (HDD)*. The speed of data access via an HDD is impacted by its proximity to the consumer and the magnetic heads reading from and writing to the platters. This is very fast and acceptable for most use cases, and it is what you get when you provision a Standard Azure Storage Account.
- *Premium* results in the provisioning of solid-state drives (SSDs), which are very low latency. Instead of physically aligning magnetic nibbles on a platter, the data is loaded into a memory chip, making it instantly accessible. If any of the following scenarios meet your requirements, then you should choose Premium which, is supported by a general-purpose v2 account:
  - Artificial intelligence/machine learning (AI/ML) processing
  - IoT or streaming scenarios
  - Data transformation solutions that require constant editing or modification that needs to be reflected immediately
  - Real-time applications that must write data quickly

Storage Redundancy

It is important to have your data stored in more than one place. For example, if you have your data in one datacenter and the datacenter loses power, you are out of business until the power is restored. If a storm destroys the datacenter and everything within it, then you are likely out of business for good if that is the only place you have stored your data.

Azure Storage provides six options for storing your data into different Azure datacenters. They are described in Table 1.6.

TABLE 1.6 Azure storage redundancy

Option	Acronym	Description
Locally redundant storage	LRS	Replicated 3 times within a single datacenter
Zone-redundant storage	ZRS	Replicated to 3 datacenters in the same region
Geo-redundant storage	GRS	LRS + replicated to another region
Geo-zone-redundant storage	GZRS	ZRS + replicated to another region
Read-access GRS	RA-GRS	Read access GRS
Read-access GZRS	RA-GZRS	Read access GZRS

LRS is straightforward in that you get three isolated copies of your data within a single datacenter. If there is any issue with a copy, Azure will take care that access is redirected to another copy. Most of the time, the redirection happens so fast you won't even notice. With ZRS, instead of having three copies of your data in the same datacenter, your three copies are in three different locations, but in the same region. A region may be an area like West Europe, South Central US, or UK South. Within each region, there are numerous datacenters (typically three) that service Azure customers, and those multiple datacenters are referred to as *zones*. Each region also has what is sometimes referred to as a *paired region*. For example, West Europe's paired region is North Europe, and South Central US is paired with North Central US. The regions exist in the same geography (or geo). GRS copies three times in one local datacenter, like LRS, and stores another copy in a different geographical location. With GZRS your data is copied into three datacenters in the first region and zones (ZRS) and then copied three times to one datacenter in another region. The read-access option means that only the primary copy of your data is writable and the other copies are read only.

Blob Storage

If you need to store files such as JSON, XML, DOCX, MP3, PDF, or any other kind of file, then use Blob Storage. Blob Storage implements a flat namespace for organizing data, in contrast to the hierarchical namespace organization that comes with ADLS Gen2. Although you can create folders and place files within them when using Blob Storage, renaming files or deleting them must be performed on each file. In other words, you cannot use wildcard syntax, which involves performing a command on all the files within a directory using

a single command. This isn't possible with Blob Storage; you must instead execute one command per file. Therefore, ADLS is more performant when running data analytics.

Table Storage

Azure Table Storage is a key-value pair database product that stores nonrelational structured data. This product is very efficient if you have a large set of data objects or records that are retrievable using a key. The Azure Cosmos DB Table API is built on top of Azure Table Storage. The additional feature you get when using the Azure Cosmos DB Table API is that your data can be distributed to all supported Azure regions, which is currently around 30+, whereas with Azure Table Storage, your data is confined to a single region with an optional read-only version in another. Refer to Table 1.6 for additional details.

Azure Files

The Azure Files feature provides the same functionality as mapping a drive using Windows Explorer or mounting a drive to a network file share. Instead of the network share being on a server within your private network, the share is in the cloud. Azure Files supports both the *Server Message Block (SMB)* and *Network Files System (NFS)* protocols.

Storage Queues

Azure has three messaging services: Azure Service Bus, Azure Event Hubs, and Azure Storage Queues. Each can receive a short text message, store it for a short period of time, and notify a subscribed system that a message has arrived. Once notified, the system can retrieve the message and perform whatever the system has been programmed to do. Storage Queue messages have a size limit of 64 KB. Compared to the other two messaging services, Azure Storage Queues are the most cost-effective for most IT solutions that need to decouple transactions. Azure Service Bus provides *first in, first out (FIFO)* capabilities, and Azure Event Hubs are for high-frequency data streaming scenarios.

Azure Disks

An Azure Disk, also called an Azure managed disk, is a hard drive that can be mounted to an Azure VM that you provision. There are data disks that come in large sizes, up to around 32,767 GB. If you needed a place to store data that is used by a process running on an Azure VM, then this is the kind of disk you should select. Alternatively, there is an OS disk where you would store and configure the operating system. The cool thing about the OS disk is that you can make numerous copies of it and mount it to multiple Azure VMs, which means you can have multiple identical servers with minimal effort.

Other Products

Here are some additional storage products that may be included on the exam. Even if they do not appear on the exam, it is good practice to know most of the Azure products, what they do, and why you would use them.

Azure Data Explorer

Azure Data Explorer (ADE) has been referred to often as a Kusto cluster, which is a datastore, typically an RDBMS, that is query-able using Kusto Query Language (KQL) syntax, which is very similar to SQL. There is tight integration between Azure Data Explorer and Azure Monitor logs and Application Insights. Therefore, if you enable those features and place the data into the Kusto cluster, you can analyze the logs using KQL. ADE can also be hooked into a data analytics process, where you can learn behaviors and find opportunities that exist using the logs generated by your applications.

Azure Storage Explorer

Azure Storage Explorer is a graphical user interface (GUI) that provides features for managing your Azure Storage features like Blob Storage, Azure Files, Storage Queues, and Table Storage. You can also set ACLs on files within your ADLS Gen2 container.

Azure Data Share

This product lets you pull data from numerous sources, create a dataset, and then share that dataset with partners or suppliers. Supported datastores include the following:

- Azure Blob Storage
- Azure Data Lake Storage Gen1 and Gen2
- Azure SQL Database
- Azure Synapse Analytics

Once your datasets are added to the Azure Data Share, you then set up a snapshot schedule. The snapshot schedule will update the datasets with the most current data on the scheduled frequency. A Data Share Invitation is sent to the data consumer, which must be accepted to access the data. The invite contains the agreed terms and conditions of how the data can be used.

Azure Databases

Azure has data service products that support most of the popular RDBMS and NoSQL databases. There are many of them, and you need to know details about each. Read on to learn more.

Azure Cosmos DB

Azure Cosmos DB is a database administration feature. It will automatically administer DBMS updates and patching, storage capacity management, and autoscaling capabilities. From a storage perspective, the scale happens automatically and elastically. *Elastically* means that when the database needs more storage space, it gets allocated, and when it is no longer

needed, it gets deallocated. That is a very cost-effective approach where only what is needed is provided and paid for. In this scenario, capacity scaling has more to do than increasing the size or number of compute instances the database runs on. It means that the instances can be scaled into different Azure regions, globally. Then, not only do you have built-in isolated data failover capabilities spread worldwide, you can also get the data as close as possible to those who produce or consume it. The closer the data is to those who require or produce it, the faster the data can be retrieved and manipulated.

Another great feature is the ability to have write capabilities in different regions. In most multi-instance data service scenarios, there is a single-source database, often called the *single version of the truth*. The other instances are copies of that source database, are read only, and get updated on a scheduled basis. However, Azure Cosmos DB will support multiregion writes and will manage the synchronization of data among all the instances.

If Azure Cosmos DB is a data administration tool, where is the data stored? When you provision an Azure Cosmos DB account, the first step is to select which data storage service API you require. The list of possible APIs is shown in Figure 1.23.

FIGURE 1.23 Azure Cosmos DB and supported APIs

See Table 1.7 for a quick overview of the APIs.

TABLE 1.7 Azure Cosmos DB APIs

API	Data model	Container item
Core (SQL)	Document database	Document
Azure Cosmos DB API for MongoDB	Document database	Document
Cassandra	Relations (Apache/CQL)	Row
Table	Key/value pair	Item
Gremlin	Graph database	Node or edge

Core (SQL)

Choose this native API if you need to work with documents, especially when the documents are JSON. Using the familiar SQL query language supports searching the content within the document itself. Additionally, there is a very feature-rich Azure Cosmos DB SDK that lets you interface and query the data source using .NET, Python, JavaScript, and Java.

Azure Cosmos DB API for MongoDB

If you already have an existing on-premises version of MongoDB, then choose this API when moving the workload to Azure. It uses the SQL query language and the Azure Cosmos DB SDK, which supports several programming languages, such as Rust, Golang, Xamarin, .NET, and Python.

Apache Cassandra

If you have an on-premises NoSQL Cassandra database and want to migrate to the Azure platform, choose this service. When you migrate to Azure, you no longer need to manage nodes, clusters, or the OS. All of that is outsourced to Azure. Data is queried using the *Cassandra Query Language (CQL)*, and you can use the client drivers you are already familiar with for accessing via code.

Table API

The Azure Cosmos DB Table API is the same concept as an Azure Table, where it stores keys and their values. You can store a massive number of these pairs, and querying them is also very fast. There is a guarantee of <10 ms latency for reads and writes, 10 million operations per second, and the global distribution you would expect from an Azure Cosmos DB API. None of those mentioned features is available using the standard Azure Table feature found in the Azure Storage account. If you want those features, then it is easy to migrate between an Azure Table and an Azure Cosmos DB Table API.

Gremlin (Graph)

Gremlin is a graph database service useful for storing data as it appears in the real world. Using the Gremlin query language, based on Apache TinkerPop project, you can access and manipulate data. The Gremlin API on Azure Cosmos DB is recommended for new workloads you want to run on the Azure platform. A graph database represents data at the storage layer instead of at a table layer. For example, if you had a relational database and wanted to store information about people, the accounts they have, and their location, you would probably create three tables and link them together with primary and foreign keys. With a graph database, each of those three would be represented by a node, and the relationship is bound between nodes and not between the data that exists within them.

Azure SQL Server Products

Microsoft SQL Server has been around for some time. It is a RDBMS created by Microsoft and uses SQL queries to manage the data stored within it. All the common concepts that go along with an RDBMS apply, like primary/foreign keys, triggers, indexes, and stored procedures. There are a lot of customers who run very large workloads on this RDBMS on Azure and in private datacenters. Numerous SQL Server offerings are available on Azure. Azure SQL is a group of products that use the SQL Server database engine in the cloud. Each of the products is described in more detail next.

SQL Server on Virtual Machines

Running this product is identical to running a full version of Microsoft SQL Server in your private datacenter. When you migrate that to Azure, though, you no longer need to manage the hardware and networking aspects. Since an Azure VM is an IaaS cloud service, you are still required to manage updates to the operating system (both Windows and Linux are supported) and SQL Server version and patches. It is still a very cost-effective solution and a vast variety of VM sizes and locations are available to run your data service on.

Azure SQL Database

This product is on the other side of the cloud service spectrum when compared to SQL Server on Azure VMs. An Azure SQL Database (SQL DB) is a relational DBaaS that offers not only the outsourcing of network and hardware infrastructure, but also operating system and DBMS responsibilities. All you need to worry about is the data and innovation. There is also a serverless (i.e., elastic) version of this product, which means your compute and storage expands and contracts on demand. Notice that the name of this product is singular, meaning it is a single database. You can have numerous Azure SQL Databases running on Azure, but they are isolated from each other.

Azure SQL Managed Instances (SQL MI)

Azure SQL Managed Instances (SQL MI) is most relevant when you want to migrate an existing SQL Server workload from a private datacenter to Azure. It is optimized for lift-and-shift scenarios. It is a PaaS service offering and supports a collection of databases manageable from a single management console.

Additional Azure Databases

Here are a few other Azure databases that are not part of the Azure Cosmos DB suite or Azure SQL Server offerings.

MariaDB MariaDB is a relational database running on Azure based on the MariaDB community edition. You can choose to run this database as either IaaS or PaaS.

MySQL MySQL is a relational database running on Azure based on the MySQL Community Edition. You can choose to run this database as either IaaS or PaaS.

PostgreSQL PostgreSQL is a relational database running on Azure based on the PostgreSQL Community Edition. You can choose either the IaaS or PaaS offering.

Other Products

The following are database-related Azure products that play an important role. They might not be on the exam, but they are good to know about nonetheless.

Azure Migrate This is the recommended tool to use for planning migrations to Azure. Azure Migrate will analyze all your on-premises servers, data, applications, and infrastructure to assess the effort and complexities of the migration.

Azure Database Migration Service You can use Azure Database Migration Service (DMS) to migrate data from multiple on-premises database sources, with minimal downtime to Azure data services.

Azure Data Box Azure Data Box helps you migrate large amounts of data to Azure. This service is optimized for data larger than 40 TB, with a maximum of 80 TB. Moving that kind of data across a network of any kind would not be possible. You place your data on a secure storage device, ship it to an Azure datacenter, and someone loads the data onto a storage point.

SQL Server Integrated Services This tool is useful for building data transformation and data integration solutions. You might notice some similarities between SSIS and Azure Data Factory or Azure Synapse Analytics as you work through your exam prep. SSIS is useful for pulling data from numerous datastores, transforming the data, and storing it on a central datastore for analysis. The central datastore historically is a data warehouse.

SQL Server Management Studio SSMS is a GUI that provides tools and features for creating and managing SQL Server databases. It supports the editing and execution of queries, performing complex administrative tasks, security management, and vulnerability assessments. Using SSMS, a database administrator (DBA) can perform most tasks required to maintain a database. SSMS can be used on SQL Server databases running on the Azure platform as well as on-premises. This GUI runs only on Windows.

Azure Data Studio This product is a more feature-limited version of SSMS. It can be run cross-platform, meaning it can run on Windows, Linux, and macOS. It is useful for creating and running SQL queries but not useful if deep administrative actions are necessary. However, many admin actions can be performed through an integrated terminal using PowerShell or sqlcmd.

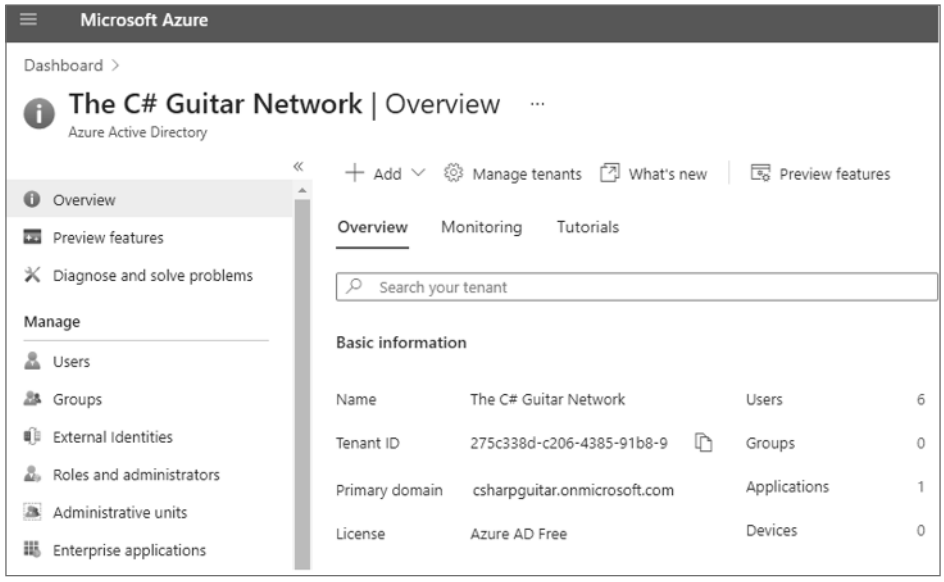
Azure Security

Security is the most important aspect of IT. Networks, compute machines, data storage, and so on are important, but a data breach is almost, if not worse, than losing all your data. Knowing about the security products, features, and services that exist on Azure is an absolute must. Although your job as an Azure data engineer might not make you responsible for implementing a security model, it is very important that you can see when something is not secure and know how to approach fixing it.

Azure Active Directory

There are numerous approaches when it comes to signing up for an Azure subscription. When you sign up for an Azure subscription, you must provide the name of your organization. That name becomes what is called an Azure Active Directory *tenant*. The tenant is assigned a Tenant ID, which is a unique GUID, and a directory endpoint in the format of `*.onmicrosoft.com`, where `*` is the tenant name. When you code applications to authenticate against the Azure Active Directory for your subscription, then that is the URL you use—for example, `csharpguitar.onmicrosoft.com`. Figure 1.24 illustrates how the Azure Active Directory blade renders in the Azure portal.

FIGURE 1.24 Azure Active Directory portal



Azure Active Directory offers many features, and we'll discuss them in the following subsections.

Remember that this chapter is an introduction and the information about many of these products have been summaries. There are many exercises in the following chapters where you will get hands-on exposure and gain more insight into the products.

Users and Groups

The most common types of accounts are the user account and the service principal. The first, as you might expect, is linked to a human being. A user account has properties like first name, last name, email address, telephone number, and most importantly, a user ID and password. The user ID and password represent the credentials and are used to confirm the user's identity. This is known as *authentication*. A service principal is also used for authentication reasons; however, this account type is not linked to a human but to a resource. A resource can be a server, a workstation, an application, or perhaps a mobile device. The point is a service principal is not linked to a person. A common reason for using a service principal is when you want to grant access to a database. Instead of using an account linked to a person, you use an account linked to the resource that needs access to the database. Then all requests from that resource to the database are rightly authenticated.

User accounts and service principals can be added to a group. It takes less administration to grant a group access to a resource versus individual accounts. The process is to create a group, then add accounts to the group. You then grant that group access to the resource and certain permissions, like read, write, and delete. If you need to add or remove someone's access to that resource, then you remove them from the group, instead of changing the properties in an individual account. If you need to remove the delete permission from everyone, then you make the change on the group, instead of on each account. When an account is used to authenticate against a resource, the permissions assigned to the group is what gets used to verify *authorization*. Since the identity has already been authenticated, which means they are who they say they are, the account is granted certain permissions that define what they are allowed to do, known as authorization.

Multifactor Authentication

Using only user IDs and passwords to protect a resource is no longer considered secure. People have a habit of using the same set of credentials on numerous locations, so if one location is compromised, then all of them are. This is where multifactor authentication (MFA) can help. User IDs and passwords are something a person knows, but when you implement MFA, it requires an object that a person has. This object is typically a mobile device, but it can also be a fingerprint, retina, thumb drive, or RFID card. When configuring your Azure Active Directory security policies, you enable this by simply selecting a check box; the platform does the rest for you.

Conditional Access

In addition to restricting access to your resources based on an authenticated identity or group affiliation, you can control access on the following common signals:

- IP location
- Device

In most corporate scenarios, you know where your employees will be connecting to your network from. Keep in mind that the context here is not a scenario where you have a website on the Internet being accessed all over the world; rather, the context is to an intranet or internal/private network. In this case you know all your users because they would have an account in the Azure AD. Every device connected to a network has an IP address, and IP addresses are grouped by country. Therefore, you can easily identify a range of IP addresses that would be requesting access to your resources, monitor them, and trigger an alert or action when not within the range. Many organizations also enforce which devices can connect to their networks. If the device is not recognized, again, an alert can be sent and an action taken.

If access is attempted and fails one or more of the conditions, you can configure the Azure AD to deny access by default. That is a bit restrictive, however, so you can instead apply an additional authentication step, such as MFA. If a user tried to authenticate from an unknown location or an unknown device, it makes sense not to, by default, deny their access. Instead, have Azure AD act on that information and ask for some additional information, like a code sent to a mobile device or email account. Once that information is successfully entered, the user can access as expected.

Azure Managed Identity

An Azure Managed Identity (MI) is the Azure AD implementation of a service principal. MI is an account that can be granted access to your Azure resources or any application that supports Azure AD authentication. It is like a service principal, but instead of using an account linked to a human, the account is linked to a resource.

Reporting and Monitoring

Since Azure AD is the place where most of your authentication and authorization activities will happen, it is very important to have some tools to make sure all is working. If you see in the audit logs that many of your employees are failing to be authenticated because of IP location, you might want to perform some analysis to find that IP range and consider changing the Conditional Access rule. Reporting and monitoring is also important to check if there have been any malicious attempts to exploit your directory. There are two reports that might come in handy here, which are available with an Azure AD Premium P2 license:

- The first is "user flagged for risk," which highlights an account that may have been compromised. It is possible for a person to lose a device that isn't protected by a PIN, and someone who finds the device can use a cached user ID and password to access the application without having to enter those credentials. The Azure platform captures

telemetry and can flag and alert security anomalies that might not be obvious to individuals. This report can help surface such events.

- The other report is called "risky sign-ins" or "risky users," which is similar to the first but instead targets compromised credentials versus a lost device. Either way, knowing what is going on in your Azure AD is extremely important.

Role-Based Access Control

RBAC is a feature that is used to control access to Azure resources in the Azure portal, Azure PowerShell, or Azure CLI. For example, you can grant someone the authority to provision and configure Azure VMs and at the same time deny them the authority to provision networking and data services. You can also grant some the authority to provision an Azure Synapse Analytics workspace and deny them authority to provision an Azure Cognitive service. It is very flexible, so you can come up with your scenarios, and they are likely supported using RBAC. There are two fundamental concepts you need to know about RBAC:

- Role definitions
- Scope

Role Definitions

Consider, for example, a role named Synapse User. When you think about that role, imagine what kind of permissions such a role would require. Does a person assigned that role need access to Azure Arc or the Azure Active Directory? Probably not. Would that role need access to Azure Synapse Analytics? That is very likely. This is what RBAC provides—it lets you define roles that have specific permissions to perform specific tasks on specific Azure resources. There are two important things to keep in mind:

- First, no permissions are granted by default, which is a good thing. It is best practice to grant a role the absolute minimum permissions required to perform their tasks.
- Second, assign a role to a group and not an individual. If you have a large organization, it is much more feasible to assign permissions to a group versus an individual. Then, if you need to change permissions, then you change it on a single group instead of potentially thousands of individual user accounts.

The following snippet illustrates an example Synapse User role in JSON format:

```
{
 "Name": "Synapse User",
 "Id": null,
 "IsCustom": true,
 "Description": "Read Synapse Analytics resources and create support cases",
 "Actions": [
              "Microsoft.Synapse/workspaces/read",
              "Microsoft.Synapse/workspaces/bigDataPools/read",
```

```
            "Microsoft.Synapse/workspaces/sqlPools/read",
            "Microsoft.Synapse/workspaces/integrationruntimes/read",
            "Microsoft.Support/*"
 ],
 "NotActions": [ ],
 "DataActions": [ ],
 "NotDataActions": [ ],
 "AssignableScopes": [
 "/subscriptions/#######-####-####-####-#########"
 ]
}
```

The first entry seen in the `Actions` property of the previous snippet is `Microsoft.Synapse/workspaces/read`, which gives the person the ability to read Azure Synapse Analytics workspaces. Notice that there is no ability to create or delete. Adding something like `Microsoft.Synapse/workspaces/*` would grant all permissions to the workspaces. The next three actions grant read access to Spark pools, SQL pools, and integration runtimes. The final entry is not part of the built-in Synapse User role but is added to allow the individual to create support cases with Microsoft, if required. This additional permission calls out the fact that if a built-in RBAC role doesn't match your needs, it is possible to create custom roles. Notice that the `IsCustom` attribute is `true`. To see a list of all built-in Azure Synapse Analytics roles, visit `https://docs.microsoft.com/en-us/azure/synapse-analytics/security/synapse-workspace-synapse-rbac-roles`.

Scope

The next concept that is important to RBAC is scope. Notice the hierarchical structure illustrated in Figure 1.25.

FIGURE 1.25 Role-based access control scope

A management group is the top level, and it can be used to define different departments within a company—for example, IT, Finance, HR, and Accounting. Each of those may be responsible for managing their own spending and therefore would want to have control over all the subscriptions for which they receive the bill. A subscription is the second level and is the bill unit; it is the place where Azure resources are provisioned into. When you provision any Azure product or feature, you must provide the associated subscription. You might imagine that a company would have numerous ongoing projects that require a set of different products and features. If you were to put them all directly into the subscription bucket, it would be hard to visually group each product into their respective projects. This is where resource groups come in. You can name a resource group based on a project and then place those resources into that group. It is an organizational unit that helps structure and categorize your Azure resources. Keep in mind that a resource can only be assigned to a single resource group.

So, what does all that have to do with scope? Well, when you place an individual into a group and then grant that group a role, the next step is to give that group access to Azure resources. You can do that at any level, as shown in Figure 1.25. Two locations are likely inefficient, such as at the top management group and at the bottom to a specific resource. Granting access at the top would give the group access to all Azure resources in all Azure subscriptions. Granting access at the resource level gives access to a specific resource and all members are constrained to only that resource. The other two, subscription or resource group, make the most sense. It depends on the role and the expectations the Azure administrator has of that role. Granting access at the subscription level gives the group access to all the included resource groups and all resources within them. Granting access to the resource group gives permissions on all the resources only within the resource group.

Attribute-Based Access Control

Attribute-based access control (ABAC) is in `PREVIEW` (or beta) as of this writing. ABAC builds on RBAC and uses an additional check based on attributes. For example, say you have provisioned an ADLS Gen2 and have loaded up thousands of files. Some of those files are highly confidential, whereas others are public. If you grant a group read access on that ADLS container, it means they can read all the files, regardless of sensitivity level. Instead, you can set an attribute on each file and add an ABAC condition to the permission based on that sensitivity attribute.

Azure Key Vault

It is unfortunately very common to store passwords and connection strings within application code. Sometimes they are stored in a configuration file along with the coded application. Even if the password or connection string is encrypted, it can still be compromised. This is because anyone with administrative access to the machine that decrypts the credential can also decrypt it. The solution to this is Azure Key Vault (AKV), which is a place to store secrets, keys, and certificates.

A secret is the place to store things like tokens, passwords, or API keys. A key is a tool that can generate and store highly sophisticated encryption keys used for encrypting your data. Certificates are *Transport Layer Security/Secure Sockets Layer (TLS/SSL)* certificates for use with your Azure web-accessible hosted resources. The most common and most secure procedure for attaining access to the information stored within the AKV is by using an MI. An application that historically used a connection string existing within a configuration file would have code that accesses and uses the value in that file. That code would need to be changed to access an AKV instead. The resource running the code—for example an Azure Function, Azure App Services Web App, or an Azure VM—would need an MI that is authenticated and authorized against the AKV. Notice in Figure 1.26 that there is a system-assigned identity configured for an Azure App Service. A user-assigned identity can be used across multiple resources, whereas a system-assigned identity is restricted to a single resource.

FIGURE 1.26 Azure App Service Managed Identity

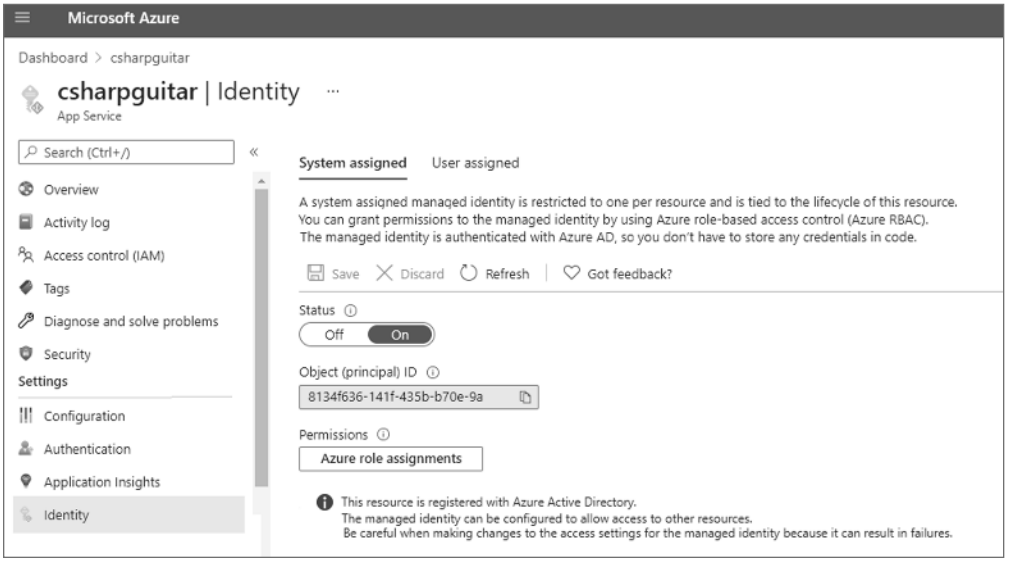


Figure 1.27 illustrates the application resource in Azure Active Directory. You would see the same on the Enterprise applications blade.

And finally, Figure 1.28 shows the granted permission to an AKV for the MI.

AKV is the go-forward solution for storing secrets, keys, and certificates. You should know this product very well.

FIGURE 1.27 Azure Managed Identity in Azure Active Directory

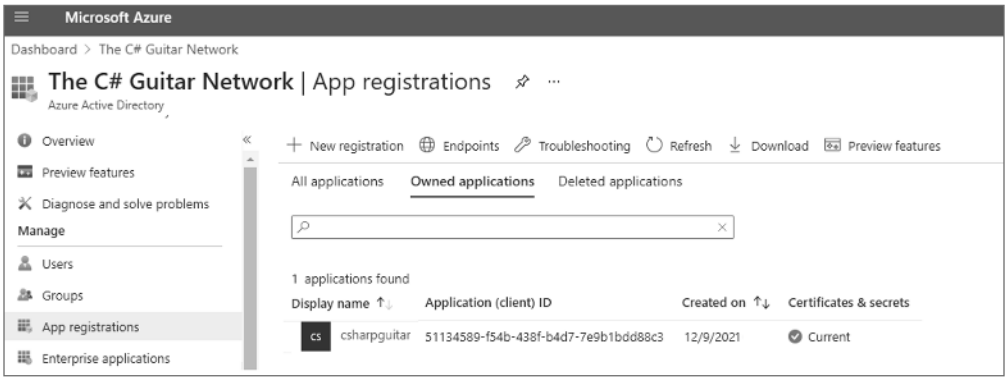
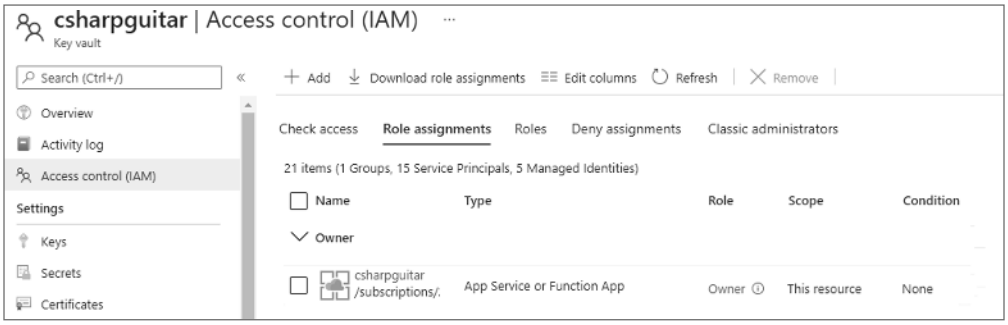


FIGURE 1.28 Azure Managed Identity in Azure Key Vault



Other Products

Security is very important. Here are some additional security-related Azure products and features you should know about.

Azure Sentinel

For those responsible for implementing a security solution, Azure Sentinel is the place to start, and it's helpful to become an expert at it. This product is an industry-leading *security information event management (SIEM)* and *security orchestration automated response (SOAR)* solution. In addition to data collections and analysis capabilities, Azure Sentinel has threat intelligence, detection, and hunting features. Founded on a collecting, detecting, investigating, and responding philosophy, Azure Sentinel protects not only all your cloud resources, but your on-premises, private resources as well.

Microsoft Defender for Cloud

Microsoft Defender for Cloud consists of two features: Azure Security Center and Azure Defender. Azure Security Center is focused on making sure your provisioned resources are in compliance with necessary regulatory standards. In addition, Azure Security Center will check to make sure all your resources follow cybersecurity industry best practice guidelines. While Azure Sentinel collects logs to detect threats, then responds to them, Azure Security Center uses checklists and industry guidelines to help enforce compliance.

Azure Defender monitors Azure VMs, SQL databases, web application, network, containers, and so forth. If a threat is found, an alert is sent to an administrator for action. Like Azure Security Center, Azure Defender can be used in a hybrid cloud scenario. This means you can monitor your on-premises resources running in your private network. These enhanced security features are also referred to under the new name of Microsoft Defender for Storage.

Azure Information Protection

If you want to have a chance at privacy and governance, the first step is to identify what assets you have that need protection, then what level of sensitivity those assets require—for example, public, internal, confidential, and highly confidential. That is not always an easy task. Azure Information Protection (AIP) can help you identify assets you may not know exist, and once classified, can help you enforce access policies and compliance.

Azure Sphere

This is a secure platform where devices can connect to via the Internet. When a device connects, it can be monitored, remotely maintained, controlled, and updated by an administrator. A scenario might be one in the IoT context where you want an isolated secure connection point for the devices and a means to manage and monitor them.

Azure Networking

The networking capabilities in Azure are in most cases abstracted away from customers. The Azure network capabilities that link the Azure resources within a datacenter and between datacenters is owned, operated, and administrated by Microsoft. However, there are some networking products and features that let customers use and configure virtual aspects of the Azure infrastructure to meet their specific requirements.

Virtual Networks

There are two scenarios you should visualize when you attempt to understand how VNets work on Azure. The first concerns IaaS, aka Azure VMs, which cannot be created without a VNet. In that case, consider that an Azure VM exists within the VNet. There are other

products like Azure Data Lake Storage, Azure SQL, and Azure Cosmos DBs that can be created without a VNet and then integrated with a VNet later. This means that the product is not created within the VNet; rather it can be bound to it and use some of its features. When you create a product without connecting it to a VNet, in most cases it means the product has a globally discoverable endpoint—that is, a public IP. That is not a security issue because the endpoint is protected, but some companies simply want to have that endpoint private. Changing an endpoint from public to private can be accomplished by binding the product to a VNet, then using the features of the VNet to control access. The most important features are as follows:

- Routing
- Network security group
- Peering
- Azure private link/endpoint
- Service endpoints
- Service tags

Routing

Once a product is connected to a VNet, you can force all outbound traffic into the VNet. Then you can use a *user-defined route (UDR)* to force that traffic wherever you would like. This is often referred to as *tunneling*. For example, many companies have on-premises software that tracks all outbound connections coming from within their networks. You can use a forced tunneling via UDRs to force the touring of traffic from Azure into your local networks and then out to the Internet via your company's network devices. Terms such as address prefix and next hop type are common in this context.

Network Security Group

A network security group (NSG) consists of numerous properties like protocol, action, direction, and port number. For example, if you wanted to allow only secured outbound Internet connectivity from machines within your VNet, you would create a rule that contains the properties shown in Table 1.8.

TABLE 1.8 NSG example

Property	Value	Possible values
Action	Allows	Allow or deny
Port	443	$1 - 2^{16}$
Direction	Outbound	Inbound or Outbound
Protocol	HTTPS	TCP, UDP, HTTP, or Any

It is also possible to allow or deny access based on the IP address of the machines. Classless Inter-Domain Routing (CIDR) is used to identify a group of allowed or denied IP address ranges. CIDR looks something like `10.0.0.0/24`.

Peering

This is a relatively easy concept to grasp. It means that you can connect two or more VNets together. Resources in VNet A can have access to resources in VNet B. You can make the access between the two VNets the same, or perhaps you wouldn't want resources in VNet B to have any or limited access to resources in VNet A.

Azure Private Link/Endpoint

So far, you have read about what is referred to as a global discoverable endpoint. Many Azure products are allocated these by default when provisioned. Here is an example of a few, where `*` represents the name of the provisioned product, which must be unique.

- `*.documents.azure.com`
- `*.azuresynapse.net`
- `*.azuredatabricks.net`
- `*.blob.core.windows.net`

To make the endpoint private, which means it is not visible on the Internet, you need to provision an Azure Private Link Service. Then, you can bind any of the endpoints shown in the list to that service, and it makes them invisible to anyone excluding the services you configure to allow discoverability. Keep in mind that this is different from simply configuring a firewall or IP restrictions. Those two actions restrict what can pass through to the resources, but the endpoint remains visible on the Internet. A private endpoint makes the endpoint invisible and then secured using a firewall or IP restriction.

Service Endpoints

This feature existed before Azure Private Links, but you should use Azure Private Links instead of Service Endpoints. If you use Service Endpoints, the global endpoint remains discoverable, but only resources that exist in the integrated VNet are allowed to access the bound resource, and vice versa. For example, if you have an Azure VM residing in a VNet and integrate an Azure App Service with the VNet, then enable Service Endpoints, that would mean that the Azure VM within the VNet can access the Azure App Service and nothing else. Of course, you have control over that—you could allow more or less access. The beauty of this product is the elevated level of security with very low configuration complexity.

Service Tags

A difficulty when setting up security that uses IP addresses is finding out which ones need to be allowed and which ones denied. When setting up these kinds of restrictions on Azure, instead of setting a restriction based on IP, you use a *service tag*, which will give all machines

that run a specific service access to your protected service. For example, `AzureKeyVault`, `AzureMonitor`, `Sql`, and `Storage` are service tags that give access to those services to your Azure resource instead of restricting to a single IP or range of IPs.

Other Products

Let's look at other networking-related Azure products you should be familiar with.

Azure Bastion

It is common practice to make a remote desktop connection to a Windows machine using the *Remote Desktop Protocol (RDP)*. The port is 3389. RDP may not be supported in some company scenarios due to security restrictions. Instead, you can use Azure Bastion to connect over port 443 from within the Azure portal. The connection between you and the Azure VM is confined to the Microsoft Azure internal network and does not traverse the Internet.

Azure ExpressRoute

Making a secured connection between your private datacenter and Azure requires a connection of some kind. You might be able to simply expose an endpoint on your local firewall and allow traffic into your intranet. This might not be the most secure approach, but it is possible. Another approach is a *virtual private network (VPN)*, which is a more secure connection between two entities across the public Internet. A VPN does have capacity constraints and the traffic does travers the Internet. An Azure ExpressRoute is useful for enterprises that need a secure, high-capacity connection from their private datacenters to the Azure platform that does not traverse the Internet.

Azure Compute

Renting computers was the original cloud service offering. The ability to provision a computer when you need it and then decommission it when it's no longer needed is very cost effective. Also, no longer having to worry about the networking and datacenter security is a very attractive part of running IT operations in the cloud.

Azure Virtual Machines

An Azure virtual machine (Azure VM) is an allocation of a CPU and memory from a larger host machine. This is the IaaS service offering from Azure. There is a great variety of sizes that are optimized for high CPU or high memory compute requirements.

Azure Virtual Machine Scale Sets

Once upon a time, it took some effort to scale out instances of Azure VMs. Historically, when you chose IaaS as your cloud solution, the code running on it was expected to run on a single instance. As time changes, so do expectations. Azure Virtual Machine Scale Sets (VMSS) is a solution that simplifies the scaling out of identical instances of a single Azure VM when required.

Azure App Service Web Apps

Azure App Service Web Apps is a PaaS cloud service offering. This product offers web application and website hosting services. The beauty of this is the scalability. When you need to add an additional server to run your code, click a button and it will be added and online within a few moments. This is also a great place to host REST APIs for your customers or partners to consume. Lastly, there is a service named WebJobs that supports the scheduling and execution of small- to middle-scale batch jobs.

Azure Functions

An Azure Function is a FaaS, aka serverless cloud service, offering on Azure. This product is useful for event-driven IT solutions, such as monitoring and processing messages that enter a messaging queue. The code that is executed is typically small, consisting of a single method or snippet of code. This product scales to the capacity that is required to perform the work automatically.

Azure Batch

Azure Batch is focused on large-scale batch execution workloads. The machines that can be selected are tuned for CPU- or memory-intensive activities. The ability to schedule and create dependencies between a sequence of jobs is also supported.

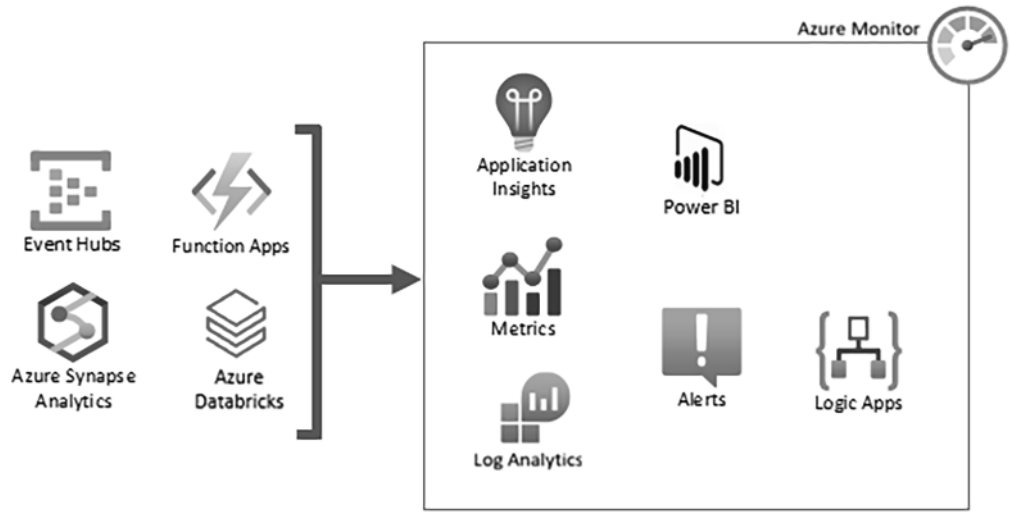
Azure Management and Governance

Once you have decided to run your workloads on Azure and identified the products you need, you should consider management and governance. Make sure you have control over spending, who can provision products, who can access data, and of course, how the products you use are performing. You also need to consider certifications like ISO, GDPR, and PCI DSS and what kind of compliance is necessary to fulfill those requirements.

Azure Monitor

You must log information about the performance, availability, and use of your workloads running on the Azure platform. This is admittedly an often overlooked aspect from IT organizations in general, not just of those that run some of their solutions in the cloud. Azure Monitor is shown in Figure 1.29.

FIGURE 1.29 Azure Monitor



One aspect that jumps out when looking at Figure 1.29 is that Azure Monitor groups together other logging and tracing products. Azure Monitor is a management console that helps you integrate and create a monitoring solution. Notice that many Azure products can integrate with Azure Monitor, like Event Hubs, Azure Functions, Azure Synapse Analytics, and Azure Databricks. The logs are stored in a datastore that can be analyzed by Azure Synapse Analytics, Metrics, and Log Analytics. The data can be visualized with Power BI, and if you find some data that is not as expected, it is possible to send alerts or trigger a Logic App so that the anomaly is surfaced.

Azure Purview

Use Azure Purview to help manage and govern your data landscape. You can use this tool with on-premises, SaaS, Azure, and other cloud service providers. Being able to identify sensitive data and where data has come from, and discovering new data sources are all features of Azure Purview. This tool is focused on your data real estate hosted on Azure. Data

Map, Data Catalog, and Data Insights are all capabilities you get via the Azure Purview Studio console.

Your company may have many datastores and databases that need to be monitored not only from an access perspective, but based on what kind of data is stored in each. Azure Purview provides an administrative overview of your data sources. Defining the reason for the data's existence, what the data's intended purpose is, the sensitivity classification of the data, and who can access are "need to know" pieces of information. Governance and manageability of your data sources is what this product is all about.

Azure Policy

After an Azure subscription is created and you begin giving rights to others to provision and configure Azure products, there needs to be control. For example, if you have specific locations where your IT must be or must not be, you might contract with a third party that requires their data be stored in a specific country. Or perhaps when an Azure VM is provisioned you need to make sure a monitoring application is installed on it. Maybe you want to restrict the sizes of Azure VMs and allow only specific ports to be opened on VNets. Most restrictions you want to place on the provisioning of Azure resources can be handled through an Azure Policy.

Azure Blueprints (Preview)

A blueprint in general is a computer-generated or hand-drawn image of some kind of architecture, in this case, IT architecture. From role assignments and implementing policies to the actual provisioning of Azure resources, the Azure Blueprints feature is a way to orchestrate the deployment of each declaratively.

Azure Lighthouse

There are many customers who run solutions on Azure that are used and/or resold by other companies. Azure Lighthouse supplies tools for governing, controlling scale, and automating the many tasks for managing such an offering.

Azure Cost Management and Billing

There are products on Azure that are very costly. Even having them provisioned for a single day, sitting idle, can cost hundreds or even thousands of dollars. The way you can manage this is by using Azure Cost Management. This toolset lets you set budgets and spending limits and configure alerts when spending is approaching that limit. You can create reports that identify which products are generating the most costs, which can help you control cash flow. One additional feature of this tool is the ability to report the cost on a group of products and features that make up a specific project. This helps you isolate and get more clarity on cost and profits on a per-project basis.

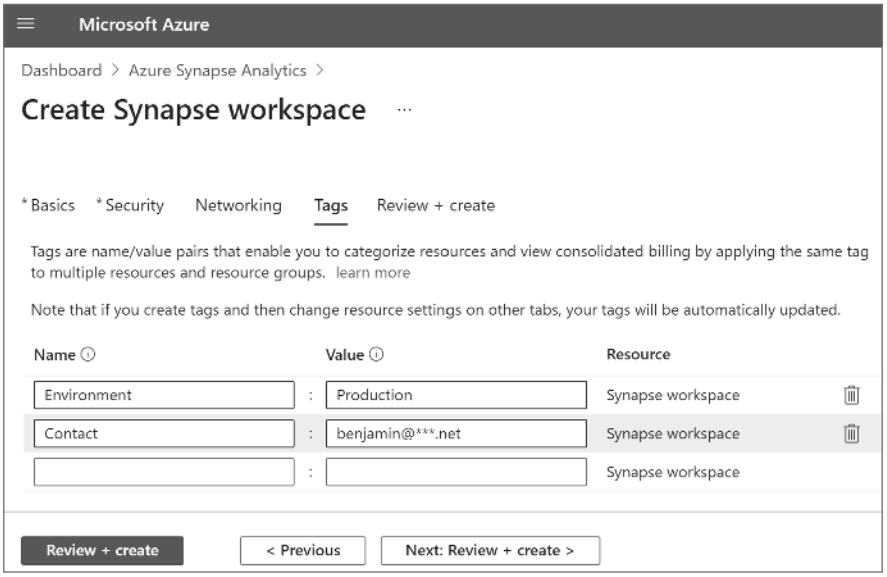
Other Products

Here are a few additional products and features that are good to know about.

Tags

When you provision an Azure resource in the Azure portal, one common step is requesting tags (see Figure 1.30). This gives you the option to add a query-able identifier to the resource. For example, you can mark it as production, test, or development, or perhaps identify a contact person for the given resource. Regardless, this is a valuable concept you might consider implementing to manage your Azure resources as your footprint grows.

FIGURE 1.30 Tags for Azure products



Azure DevOps

This product is based on application source code and the development process. A company's application source code is stored in Azure Repos that store history. Azure Pipelines can be configured to automatically perform a release into production when source code is checked in. That process is commonly referred to as *continuous integration/continuous deployment (CI/CD)*. Azure Boards and Azure Test plans are helpful in linking source code changes to specific work items and making sure the change works and meets the expectations of the requested work.

Summary

This chapter provided tips on preparing for the Data Engineering on Microsoft Azure exam. You learned about the importance of staying current and that you should make sure to work with Azure often, especially with Azure data service products. Working with these tools will keep the information fresh in your head and will make it much easier to pass the exam. Finally, this chapter introduced you to a large number of Azure products. Some of them you must know in great depth, whereas others you only need to know a little bit and the rest not so much. The point is that you know, or have at least some idea of, what most Azure products are and when you would use them.

Exam Essentials

Make sure you are on the right Azure path. There are numerous Azure certification paths. Make sure you are on the path that best fits your career's objectives. Azure Data Engineer Associate certification is for data specialists who want to work with data warehouses, data lakes, Big Data, machine learning, and data intelligence gathering. Being curious is a very important trait of a data engineer, and looking for clues and trends in massive amounts of data should be a passion.

Gain experience with Azure. To pass this exam, you will need experience with the platform. Do not expect the book or training to get you through this without having worked on the platform for some amount of time. If you do that, then this certification will have greater meaning to you and others.

Keep up-to-date by subscripting to online resources. The Azure platform is constantly changing, which requires daily actions to keep your skills up-to-date. Here are some of the most popular resources to read on a regular basis.

`https://docs.microsoft.com/azure`

`https://azure.microsoft.com/updates`

`https://aka.ms/azheatmap`

`https://azure.microsoft.com/services`

`https://docs.microsoft.com/azure/?product=analytics`

Know what products exist in Azure. Sometimes, possible answers to exam questions mention products that are not real Azure products. They read or sound like they are, but they are not. Knowing what really is available on the platform will help you remove at least one possible answer to a few questions.

Gain deep, some and base knowledge. No one knows everything, but you can be an expert in one or two Azure products and features. Those products and features usually have a

connection or dependency to other Azure features, which you could then learn some internals about. There are some products and features that are not related to any other directly. For example, there is no direct relationship between Azure Cognitive Services and a VNet. It would be good to have at least a base knowledge of what benefits they can provide.

Focus on certain products. The Azure Data Engineer Associate exam is heavy on Azure Synapse Analytics, SQL Pools, Azure Databricks, Azure Data Factory, Azure Active Directory, Azure Key Vault, and Stream Analytics products. Make sure you have a very good understanding of these products, their features, and their limits, as well as DataFrames, advanced SQL querying, clusters, and *slowly changing dimensions (SCDs)*. This knowledge gives you the best chance of passing the exam.

Review Questions

Many questions can have more than a single answer. Please select all choices that are true.

1. Which of the following are pools available in Azure Synapse Analytics?
   - **A.** SQL pools
   - **B.** Spark pools
   - **C.** Azure Cosmos DB pools
   - **D.** Integration runtime (IR) pools
2. Which of the following are components of Azure Synapse Analytics?
   - **A.** SQL Pools and Apache Spark Pools
   - **B.** Integration runtimes (IRs)
   - **C.** Pipelines
   - **D.** Linked Services
3. What is the *best* definition of a data lake?
   - **A.** A place where data that is no longer needed gets stored
   - **B.** A place to store data that has not yet been transformed
   - **C.** A logical grouping of all your data
   - **D.** The location where transformed data is stored
4. Why would a customer choose to use Azure HDInsight?
   - **A.** The solution must run on Apache Spark pools.
   - **B.** The solution requires the Linux operating system.
   - **C.** Their solution is dependent on Kafka.
   - **D.** The on-premises solution already runs HDInsight.
5. What is a common use of Azure Stream Analytics?
   - **A.** To analyze data existing on an ADLS Gen1 datastore
   - **B.** To analyze data streams and identify anomalies
   - **C.** To transform data existing on an Azure Cosmos DB
   - **D.** To update Power BI with real-time analytics
6. Which of the following are Azure SQL database offerings?
   - **A.** SQL Server on Virtual Machines
   - **B.** SQL on Elastic Pools (SQL EP)
   - **C.** Azure SQL Managed Instance (SQL MI)
   - **D.** Azure SQL Database (SQL DB)

7. What is the purpose of role-based access controls (RBACs)?
   - **A.** Defines the permissions for a group of individuals
   - **B.** Prevents users from deleting data from a database
   - **C.** Can enforce sensitivity level data access
   - **D.** Assigns the scope of permissions

8. An Azure Managed Identity is which of the following?
   - **A.** Similar to a service principal
   - **B.** An identity of a provisioned Azure resource
   - **C.** Can be either user- or system-defined identities
   - **D.** A permission constrained account for guests

9. Which of the following are features of a VNet?
   - **A.** A subnet
   - **B.** Azure Virtual Machine
   - **C.** Network security group (NSG)
   - **D.** Azure Firewall

10. Which tool is responsible for monitoring your Azure resources?
    - **A.** Azure Lighthouse
    - **B.** Azure Purview
    - **C.** Azure Monitor
    - **D.** Azure Sphere