Understanding Azure Data Factory

Operationalizing Big Data and Advanced Analytics Solutions

Sudhir Rawat
Abhishek Narain

Apress®

Understanding Azure Data Factory

Operationalizing Big Data and Advanced Analytics Solutions

Sudhir Rawat
Abhishek Narain

Apress®

Understanding Azure Data Factory: Operationalizing Big Data and Advanced Analytics Solutions

Sudhir Rawat
Bangalore, India

Abhishek Narain
Shanghai, China

ISBN-13 (pbk): 978-1-4842-4121-9
ISBN-13 (electronic): 978-1-4842-4122-6
https://doi.org/10.1007/978-1-4842-4122-6

Library of Congress Control Number: 2018965932

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Smriti Srivastava
Development Editor: Laura Berendson
Coordinating Editor: Shrikant Vishwakarma

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-4121-9. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Authors

Sudhir Rawat is a senior software engineer at Microsoft Corporation. He has 15 years of experience in turning data to insights. He is involved in various activities, including development, consulting, troubleshooting, and speaking. He works extensively on the data platform. He has delivered sessions on platforms at Microsoft TechEd India, Microsoft Azure Conference, Great India Developer Summit, SQL Server Annual Summit, Reboot (MVP), and many more. His certifications include MCITP, MCTS, MCT on SQL Server Business Intelligence, MCPS on Implementing Microsoft Azure Infrastructure Solutions, and MS on Designing and Implementing Big Data Analytics Solutions.

Abhishek Narain works as a technical program manager on the Azure Data Governance team at Microsoft. Previously he worked as a consultant at Microsoft and Infragistics, and he has worked on various Azure services and Windows app development projects. He is a public speaker and regularly speaks at various events, including Node Day, Droidcon, Microsoft TechEd, PyCon, the Great India Developer Summit, and many others. Before joining Microsoft, he was awarded the Microsoft MVP designation.

About the Technical Reviewer

Zain Asif is a freelance senior developer specializing in Microsoft technologies (C#, ASP.NET, ASP.NET MVC, ASP.NET Core, Azure Data Lake, Azure Data Factory, SQL Server and Power BI). He is passionate about new technologies, both software and hardware ones.

He is the founder of Falcon Consulting, and with it, he has had the opportunity to work with the biggest companies around the world such as Microsoft, Canon, and Accor. His aim in the future is to make his company an IT engineering company and work as a freelance software architect and Microsoft expert.

When not working, Zain can be seen on the ground playing cricket or football or in front of a PC geeking and gaming.

Introduction

Azure Data Factory is the de facto tool for building end-to-end advanced analytics solutions on Azure. It can handle complex ETL data workflows and integrates natively with all Azure services with enterprise-grade security offerings.

For ease of authoring and to make you more productive, it offers a drag-and-drop user interface with rich control flow for building complex data workflows, and it provides a single-pane-of-glass monitoring solution for your data pipelines.

Something that really stands out is the low price-to-performance ratio, being cost effective and performant at the same time. Its data movement capabilities with more than 75 high-performance connectors are extremely helpful when dealing with Big Data coming from various sources. To give you an example, 100GB data movement would cost you less than $0.40 (that is correct, 40 cents). ADF is an Azure service and bills you in a pay-as-you-go model against your Azure subscription with no up-front costs.

ADF also supports operationalizing existing SSIS packages on the cloud, which is helpful if you are modernizing your data warehouse solution over time with a lot of existing SSIS packages.

CHAPTER 1

Introduction to Data Analytics

The demand for Big Data analytics services is greater than ever before, and this trend will only continue—exponentially so—as data analytics platforms evolve over time. This is a great time to be a data engineer or a data scientist with so many options of analytics platforms to select from.

The purpose of this book is to give you the nitty-gritty details of operationalizing Big Data and advanced analytics solutions on Microsoft Azure.

This book guides you through using Azure Data Factory to coordinate data movement; to perform transformations using technologies such as Hadoop (HDInsight), SQL, Azure Data Lake Analytics, Databricks, files from different kinds of storage, and Cosmos DB; and to execute custom activities for specific tasks (coded in C#). You will learn how to create data pipelines that will allow you to group activities to perform a certain task. This book is hands-on and scenario-driven. It builds on the knowledge gained in each chapter.

The focus of the book is to also highlight the best practices with respect to performance and security, which will be helpful while architecting and developing extract-transform-load (ETL), extract-load-transform (ELT), and advanced analytics projects on Azure.

This book is ideal for data engineers and data scientists who want to gain advanced knowledge in Azure Data Factory (a serverless ETL/ELT service on Azure).

S. Rawat and A. Narain, *Understanding Azure Data Factory*,
https://doi.org/10.1007/978-1-4842-4122-6_1

What Is Big Data?

Big Data can be defined by following characteristics:

- *Volume*: As the name says, Big Data consists of extremely large datasets that exceed the processing capacity of conventional systems such as Microsoft SQL, Oracle, and so on. Such data is generated through various data sources such as web applications, the Internet of Things (IoT), social media, and line-of-business applications.
- *Variety*: These sources typically send data in a variety of formats such as text, documents (JSON, XML), images, and video.
- *Velocity*: This is the speed at which data is generated is by such sources. High velocity adds to Big Data. For example a factory installed sensor to keep monitor it's temperature to avoid any damage. Such sensors sends E/Sec (event per second) or sometime in millisecond. Generally IoT enable places has many such sensors which sends data so frequently.
- *Veracity*: This is the quality of data captured from various sources. System also generates bias, noise and abnormal data which adds to Big Data. High veracity means more data. It not only adds to big data but also add responsibility to correct it to avoid presenting wrong information to the business user.

Let's think about a fictious retail company called AdventureWorks, which has a customer base across the globe. AdventureWorks has an e-commerce web site and mobile applications for enabling users to shop online, lodge complaints, give feedback, apply for product returns, and so on. To provide the inventory/products to the users, it relies on a business-

to-business (B2B) model and partners with vendors (other businesses) that want to list their products on AdventureWorks e-commerce applications. AdventureWorks also has sensors installed on its delivery vans to collect various telemetry data; for example, it provides customers with up-to-date information on consignment delivery and sends alerts to drivers in the case of any issue, for example a high temperature in the delivery van's engine. The company also sends photographers to various trekking sites. All this data is sent back to the company so it can do image classification to understand the gadgets in demand. This helps AdventureWorks stock the relevant items. AdventureWorks also captures feeds from social media in case any feedback/comment/complaint is raised for AdventureWorks.

To get some valuable insights from the huge volume of data, you must choose a distributed and scalable platform that can process the Big Data. Big Data has great potential for changing the way organizations use information to enhance the customer experience, discover patterns in data, and transform their businesses with the insights.

Why Big Data?

Data is the new currency. Data volumes have been increasing drastically over time. Data is being generated from traditional point-of-sale systems, modern e-commerce applications, social sources like Twitter, and IoT sensors/wearables from across the globe. The challenge for any organization today is to analyze this diverse dataset to make more informed decisions that are predictive and holistic rather than reactive and disconnected.

Big Data analytics is not only used by modern organizations to get valuable insights but is also used by organizations having decades-old data, which earlier was too expensive to process, with the availability of pay-as-you-go cloud offerings. As an example, with Microsoft Azure you can easily spin up a 100-node Apache Spark cluster (for Big Data analytics) in less than ten minutes and pay only for the time your job runs on those clusters, offering both cloud scale and cost savings in a Big Data analytics project.

Big Data Analytics on Microsoft Azure

Today practically every business is moving to the cloud because of lucrative reasons such as no up-front costs, infinite scale possibilities, high performance, and so on. The businesses that store sensitive data that can't be moved to the cloud can choose a hybrid approach. The Microsoft cloud (aka Azure) provides three types of services.

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)

It seems like every organization on this planet is moving to PaaS. This gives companies more time to think about their business while innovating, improving customer experience, and saving money.

Microsoft Azure offers a wide range of cloud services for data analysis. We can broadly categorize them under storage and compute.

- Azure SQL Data Warehouse, a cloud-based massively parallel-processing-enabled enterprise data warehouse
- Azure Blob Storage, a massively scalable object storage for unstructured data that can be used to search for hidden insights through Big Data analytics
- Azure Data Lake Store, a massively scalable data store (for unstructured, semistructured, and structured data) built to the open HDFS standard
- Azure Data Lake Analytics, a distributed analytics service that makes it easy for Big Data analytics to support programs written in U-SQL, R, Python, and .NET

- Azure Analysis Services, enterprise-grade data modeling tool on Azure (based on SQL Server Analysis Service)
- Azure HDInsight, a fully managed, full-spectrum open source analytics service for enterprises (Hadoop, Spark, Hive, LLAP, Storm, and more)
- Azure Databricks, a Spark-based high-performance analytics platform optimized for Azure
- Azure Machine Learning, an open and elastic AI development tool for finding patterns in existing data and generating models for prediction
- Azure Data Factory, a hybrid and scalable data integration (ETL) service for Big Data and advanced analytics solutions
- Azure Cosmos DB, an elastic and independent scale throughput and storage tool; it also offers throughput, latency, availability, and consistency guarantees with comprehensive service level agreements (SLAs), something no other database service offers at the moment

What Is Azure Data Factory?

Big Data requires a service that can help you orchestrate and operationalize complex processes that in turn refine the enormous structure/semistructured data into actionable business insights.

Azure Data Factory (ADF) is a cloud-based data integration service that acts as the glue in your Big Data or advanced analytics solution, ensuring your complex workflows integrate with the various dependent

services required in your solution. It provides a single pane for monitoring all your data movements and complex data processing jobs. Simply said, it is a serverless, managed cloud service that's built for these complex hybrid ETL, ELT, and data integration projects (data integration as a service).

Using Azure Data Factory, you can create and schedule data-driven workflows (called *pipelines*) that can ingest data from disparate data stores. It can process and transform the data by using compute services such as Azure HDInsight Hadoop, Spark, Azure Data Lake Analytics, and Azure Machine Learning (Figure 1-1).

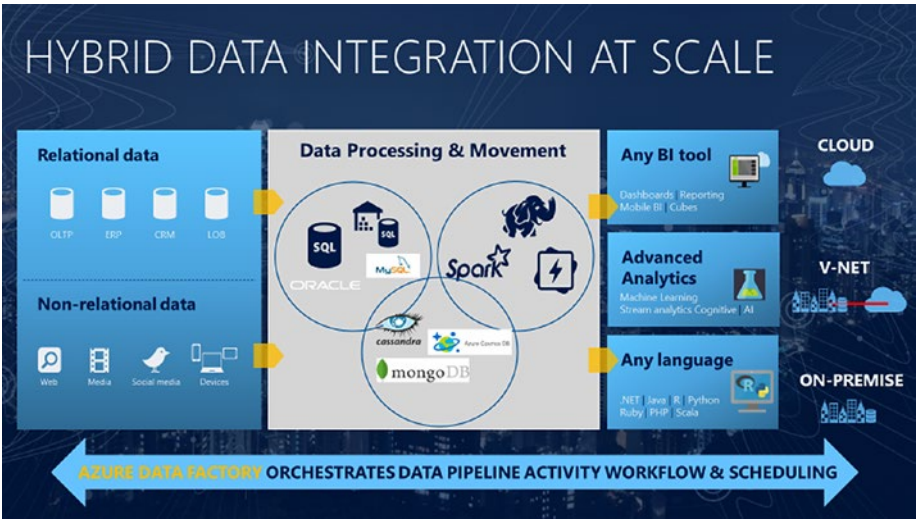

Figure 1-1. *Azure Data Factory*

High-Level ADF Concepts

An Azure subscription might have one or more ADF instances. ADF is composed of four key components, covered in the following sections. These components work together to provide the platform on which you can compose data-driven workflows with steps to move and transform data or execute custom tasks using custom activity that could include

deleting files on Azure storage after transforms or simply running additional business logic that is not offered out of the box within Azure Data Factory.

Activity

An *activity* represents an action or the processing step. For example, you copy an activity to copy data between a source and a sink. Similarly, you can have a Databricks notebook activity transform data using Azure Databricks. ADF supports three types of activities: data movement, data transformation, and control flow activities.

Pipeline

A *pipeline* is a logical grouping of activities. Typically, it will contain a set of activities trying to achieve the same end goal. For example, a pipeline can contain a group of activities ingesting data from disparate sources, including on-premise sources, and then running a Hive query on an on-demand HDInsight cluster to join and partition data for further analysis.

The activities in a pipeline can be chained together to operate sequentially, or they can operate independently in parallel.

Datasets

Datasets represent data structures within the data stores, which simply point to or reference the data you want to use in your activities as inputs or outputs.

Linked Service

A *linked service* consists of the connection details either to a data source like a file from Azure Blob Storage or a table from Azure SQL or to a compute service such as HDInsight, Azure Databricks, Azure Data Lake Analytics, and Azure Batch.

Integration Runtime

The *integration runtime* (IR) is the underlying compute infrastructure used by ADF. This is the compute where data movement, activity dispatch, or SSIS package execution happens. It has three different names: Azure, self-hosted, and Azure SQL Server Integration Services (Figure 1-2).

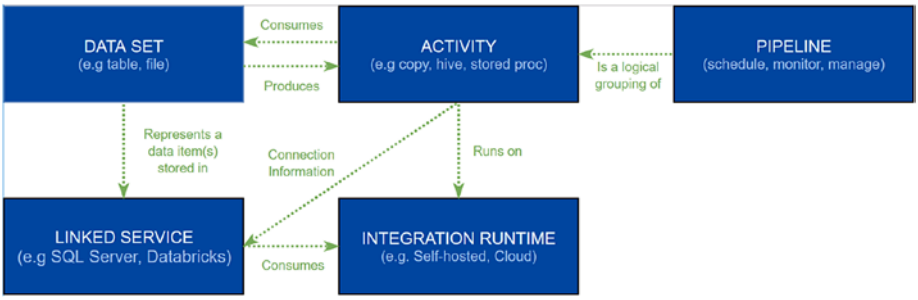

Figure 1-2. *Relationship between ADF components*

When to Use ADF?

The following are examples of when you should use ADF:

- Building a Big Data analytics solution on Microsoft Azure that relies on technologies for handling large numbers of diverse datasets. ADF offers a way to create and run an ADF pipeline in the cloud.
- Building a modern data warehouse solution that relies on technologies such as SQL Server, SQL Server Integration Services (SSIS), or SQL Server Analysis Services (SSAS); see Figure 1-3. ADF provides the ability to run SSIS packages on Azure or build a modern ETL/ELT pipeline letting you access both on-premise and cloud data services.

- Migrating or coping data from a physical server to the cloud or from a non-Azure cloud to Azure (blob storage, data lake storage, SQL, Cosmos DB). ADF can be used to migrate both structured and binary data.

You will learn more about the ADF constructs in Chapter 2.

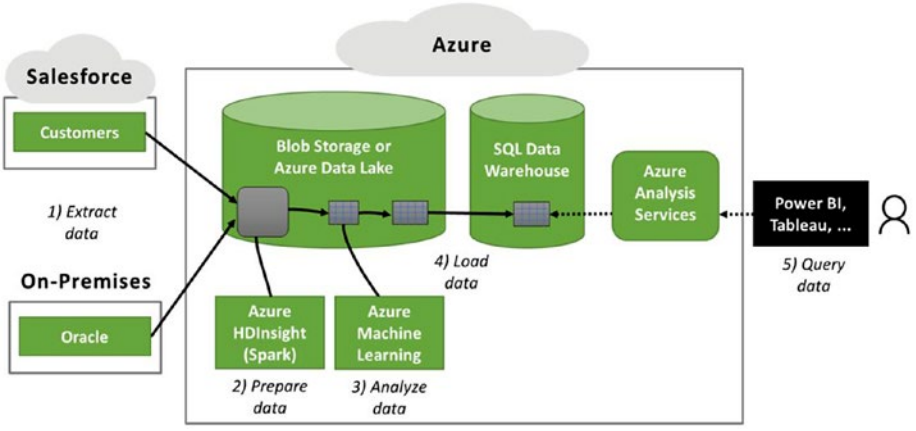

Figure 1-3. *A typical modern data warehouse solution*

Why ADF?

The following are reasons why you should use ADF:

- *Cost effective*: ADF is serverless, and the billing is based on factors such as the number of activities run, the data movement duration, and the SSIS package execution duration. You can find the latest pricing details at `https://aka.ms/adfpricing`.

  For example, if you run your ETL/ ELT pipeline hourly, which also involves data movement (assuming 100GB data movement per hourly run, which should take around 8 minutes with 200MBps

bandwidth), then ADF would bill you not more than $12 for the monthly execution (720 pipeline runs).

Note: The charges for any other service (HDInsight, Azure Data Lake Analytics) are not considered in this calculation. This is solely for the ADF orchestration and data movement cost. On the contrary, there are non-Microsoft ETL/ELT tools that may offer similar capabilities with a much higher cost.

- *On-demand compute*: ADF provides additional cost-saving functionality like on-demand provisioning of Hindsight Hadoop clusters. It takes care of the provisioning and teardown of the cluster once the job has executed, saving you a lot of additional cost and making the whole Big Data analytics process on-demand.
- *Cloud scale*: ADF, being a platform-as-a-service offering, can quickly scale if need be. For the Big Data movement, with data sizes from terabytes to petabytes, you will need the scale of multiple nodes to chunk data in parallel.
- *Enterprise-grade security*: The biggest concern around any data integration solution is the security, as the data may well contain sensitive personally identifiable information (PII).

  Since ADF is a Microsoft-owned service (or as I call it a *first-party citizen* on Azure), it follows the same security standards as any other Microsoft service. You can find the security and compliance certification information online.

A common challenge when building cloud applications is to manage the credentials that need to be in your code/ADF pipeline for authenticating to cloud services. Keeping these credentials secure is an important task. Ideally, they never appear on developer workstations or get checked into source control. ADF supports Azure Key Vault, which provides a way to securely store credentials and other keys and secrets, but your code/ADF pipeline needs to authenticate to Key Vault to retrieve them. Managed Service Identity (MSI) makes solving this problem simpler by giving Azure services such as ADF an automatically managed identity in Azure Active Directory (Azure AD). ADF supports MSI and uses this identity to authenticate to any service that supports Azure AD authentication, including Key Vault, without having any credentials in your code/ADF pipeline, which probably is the safest option for service-to-service authentication on Azure.

- *Control flow*: You can chain activities in a sequence, branch based on certain conditions, define parameters at the pipeline level, and pass arguments while invoking the pipeline on-demand or from a trigger. ADF also includes custom state passing and looping containers, that is, for-each iterators.
- *High-performance hybrid connectivity*: ADF supports more than 70 connectors at the time of writing this book. These connectors support on-premise sources as well, which helps you build a data integration solution with your on-premise sources.

- *Easy interaction*: ADF's support for so many connectors makes it easy to interact with all kinds of technologies.
- *Visual UI authoring and monitoring tool*: It makes you super productive as you can use drag-and-drop development. The main goal of the visual tool is to allow you to be productive with ADF by getting pipelines up and running quickly without requiring you to write a single line of code.
- *SSIS package execution*: You can lift and shift an existing SSIS workload.
- *Schedule pipeline execution*: Every business have different latency requirements (hourly, daily, monthly, and so on), and jobs can be scheduled as per the business requirements.
- *Other development options*: In addition to visual authoring, ADF lets you author pipelines using PowerShell, .NER, Python, and REST APIs. This helps independent software vendors (ISVs) build SaaS-based analytics solutions on top of ADF app models.

Summary

Azure Data Factory is a serverless data integration service on the cloud that allows you to create data-driven workflows for orchestrating and automating data movement and data transformation for your advanced analytics solutions. In the upcoming chapters, you will dig deeper into each aspect of ADF with working samples.

CHAPTER 2

Introduction to Azure Data Factory

In any Big Data or advanced analytics solution, the orchestration layer plays an important role in stitching together the heterogenous environments and operationalizing the workflow. Your overall solution may involve moving raw data from disparate sources to a staging/sink store on Azure, running some rich transform jobs (ELT) on the raw data, and finally generating valuable insights to be published using reporting tools and stored in a data warehouse for access. Azure Data Factory is the extract-transform-load (ETL)/extract-load-transform (ELT) service offered by Microsoft Azure.

Azure Data Factory (ADF) is a Microsoft Azure platform-as-a-service (PaaS) offering for data movement and transformation. It supports data movement between many on-premise and cloud data sources. The supported platform list is elaborate and includes both Microsoft and other vendors. It is a powerful tool providing complete flexibility for the movement of structured and unstructured datasets, including RDBMS, XML, JSON, and various NoSQL data stores. Its core strength is the flexibility of being able to use U-SQL or HiveQL.

This chapter will introduce you to Azure Data Factory basics (Figure 2-1). This knowledge will form the building blocks for the advanced analytics solution that you will build later in the book.

S. Rawat and A. Narain, *Understanding Azure Data Factory*,
https://doi.org/10.1007/978-1-4842-4122-6_2

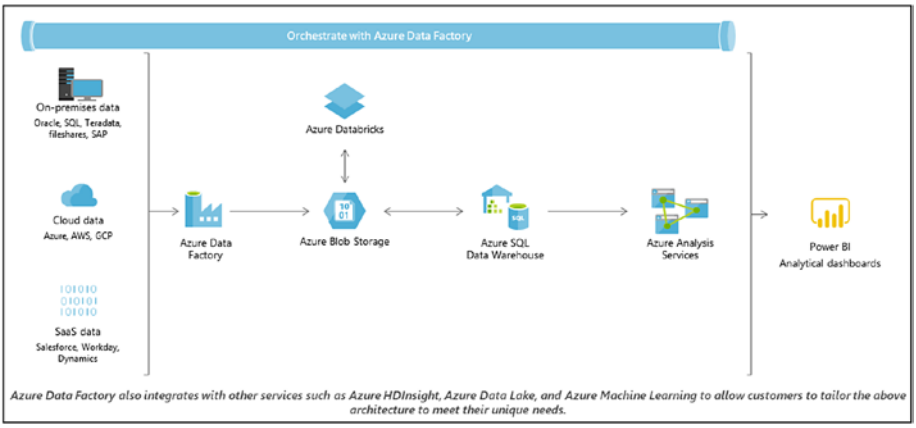

Figure 2-1. *Azure Data Factory basics*

Azure Data Factory v1 vs. Azure Data Factory v2

When you create an Azure Data Factory resource on your Azure subscription, the wizard will ask you to choose between Azure Data Factory v1 and Azure Data Factory v2. Azure Data Factory version 2 is generally available and being actively developed, which means regular feature updates. Azure Data Factory v1 is stabilized, but it's more limited than v2. ADF v2 adds the much needed control flow functionality, which lets data engineers define complex workflows. Monitoring is also an added enhancement in v2, making it much richer and natively integrating it with Azure Monitor and Microsoft Operations Management Suite for building single-pane-of-glass monitoring. One of the biggest features of v2 is the integration of SQL Server Integration Services (SSIS). Many Microsoft customers have been using SSIS for their data movement needs primarily involving SQL Server databases for many years because SSIS has been in existence for a long time. The integration of SSIS and Azure Data Factory

has been a key customer requirement for migrating to the PaaS platform for ETL without needing to rewrite the entire data transformation logic across the enterprise.

The recent release of Azure Data Factory v2 has taken a major step toward meeting this requirement. SSIS packages can now be integrated with ADF and can be scheduled/orchestrated using ADF v2. The SSIS package execution capability makes all fine-grained transformation capabilities and SSIS connectors available from within ADF. Customers can utilize existing ETL assets while expanding ETL capabilities with the ADF platform.

ADF v2 allows SSIS packages to be moved to the cloud using the integration runtime (IR) to execute, manage, monitor, and deploy these packages to Azure. The IR allows for three different scenarios: Azure (a pure PaaS with endpoints), self-hosted (within a private network), and Azure-SSIS (a combination of the two).

The capability of SSIS package integration with ADF has led to the expansion of a core feature of the ADF platform. Specifically, there is now a separate control flow in the ADF platform. The activities are broken into data transformation activities and control flow activities; this is similar to the SSIS platform.

In addition to the SSIS integration, ADF v2 has expanded its functionality on a few other fronts. It now supports an extended library of expressions and functions that can be used in the JSON string value. Data pipeline monitoring is available using OMS tools in addition to the Azure portal. This is a big step toward meeting the requirements of customers with established OMS tools for any data movement activity.

There has also been a change in job scheduling in ADF v2. In the prior version, jobs were scheduled based on time slices. This feature has been expanded in ADF v2. Jobs can be scheduled based on triggering events, such as the completion of a data refresh in the source data store.

In this book, we will focus on Azure Data Factory v2, but most of the features are applicable to v1 too.

Data Integration with Azure Data Factory

Azure Data Factory offers a code-free, drag-and-drop, visual user interface to maximize productivity by getting data pipelines up and running quickly. You can also connect the visual tool directly to your Git repository for a seamless deployment workflow. Using Azure Data Factory, you can create and schedule data-driven workflows (called *pipelines*) that can ingest data from disparate data stores. ADF can process and transform the data by using compute services such as Azure HDInsight Hadoop, Spark, Azure Data Lake Analytics, Azure Cosmos DB, and Azure Machine Learning.

You can also write your own code in Python, .NET, the REST API, Azure PowerShell, and Azure Resource Manager (ARM) to build data pipelines using your existing skills. You can choose any compute or processing service available on Azure and put them into managed data pipelines to get the best of both the worlds.

Architecture

When you create an Azure Data Factory v2 resource on your Azure subscription, you create a data integration account. This is sort of a serverless workplace where you can author your data pipelines. You are not billed for this step. You pay for what you use, and that will happen only when you execute some pipeline.

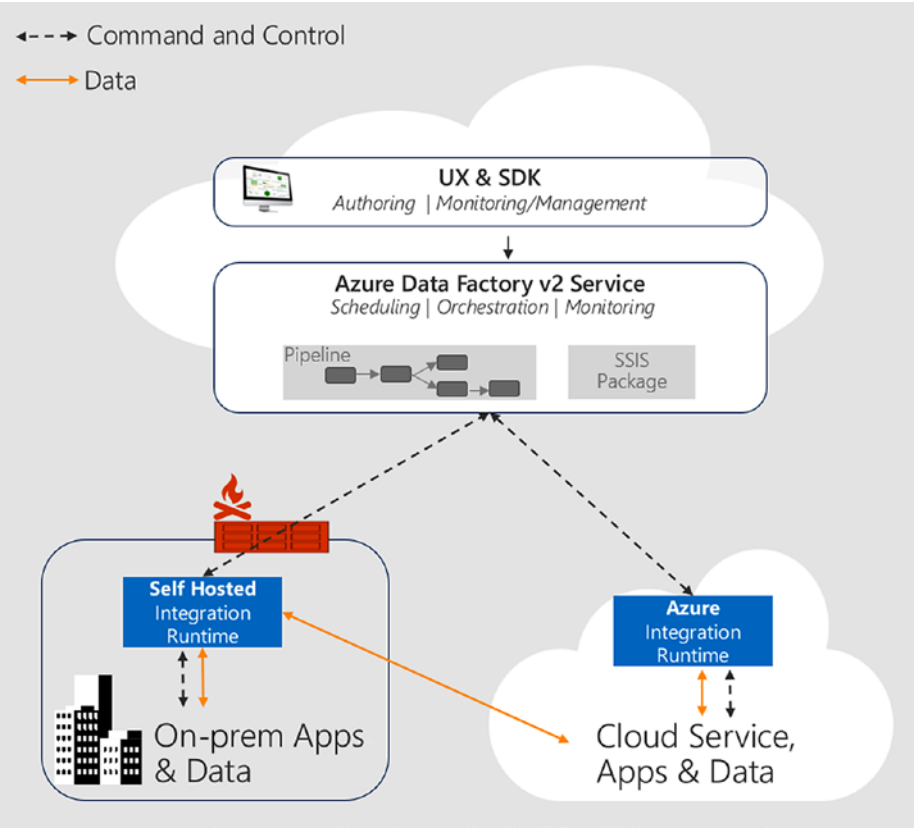

Figure 2-2. *ADF architecture showing the command/ control flow versus data flow during orchestration*

Once you start authoring the pipeline, the ADF service stores the pipeline metadata in the selected ADF region. When your pipeline is executed, the orchestration logic runs on some compute, in other words, the integration runtime. There are three types of IR used for different purposes, and I will talk about the use of each one of them in the upcoming sections.

Concepts

Azure Data Factory is composed of five key components. These components come together while you build data-driven workflows for transforming data.

Pipelines

A *pipeline* is a logical grouping of activities performing a set of processes such as extracting data, transforming it, and loading into some database, data warehouse, or storage. For example, a pipeline can contain a group of activities to ingest data from Amazon S3 (an on-premise file system to a staging store) and then run a Spark query on an Azure Databricks cluster to partition the data.

A data factory might have one or more pipelines.

An Azure Data Factory instance uses JSON to describe each of its entities. If you are using visual authoring, you will not need to understand this structure. But when writing code/script, you'll need to understand this JSON payload (see Table 2-1).

Here is how a pipeline is defined in JSON format:

```
{
    "name": "PipelineName",
    "properties":
    {
        "description": "pipeline description",
        "activities":
        [
        ],
        "parameters": {
         }
    }
}
```

Table 2-1. *Pipeline Properties*

Tag	Description	Type	Required
`name`	Specifies the name of the pipeline. Use a name that represents the action that the pipeline performs. Maximum number of characters: 140. Must start with a letter, number, or underscore (_). The following characters are not allowed: . + ? / < > * % & : \	String	Yes
`description`	Specifies the text describing what the pipeline is used for.	String	No
`activities`	The pipeline can have one or more activities defined within it.	Array	Yes
`parameters`	The `parameters` property can have one or more parameters defined within the pipeline, making your pipeline flexible for reuse.	List	No

Activities

Activities represent a processing step in a pipeline. These are specific tasks that compose the overall pipeline. For example, you might use a Spark activity, which runs a Spark query against Azure Databricks or an HDInsight cluster, to transform or analyze your data. Azure Data Factory supports three types of activities: data movement (copy activities), data transform (compute activities), and control activities.

Execution Activities (Copy and Data Transform)

The following are the execution activities:

- Copy supports 70+ connectors for copying data from the source to the sink, including binary copy. I will cover this in depth in Chapter 3.
- Data transform supports the transform activities in Table 2-2.

Table 2-2. *Transform Activities*

Data Transformation Activity	Compute Environment
Hive	HDInsight (Hadoop)
Pig	HDInsight (Hadoop)
MapReduce	HDInsight (Hadoop)
Hadoop streaming	HDInsight (Hadoop)
Spark	HDInsight (Hadoop)
Machine learning activities: batch execution and update resource	Azure VM
Stored procedure	Azure SQL, Azure SQL Data Warehouse, or SQL Server
U-SQL	Azure Data Lake Analytics
Cosmos DB	Azure Cosmos DB
Custom code	Azure Batch
Databricks notebook	Azure Databricks
Databricks JAR	Azure Databricks
Databricks Python	Azure Databricks