Pro Database Migration to Azure

Data Modernization for the Enterprise

Kevin Kline
Denis McDowell
Dustin Dorsey
Matt Gordon

Foreword by Bob Ward

Apress®

Pro Database Migration to Azure

Data Modernization for the Enterprise

Kevin Kline
Denis McDowell
Dustin Dorsey
Matt Gordon

Foreword by Bob Ward

Apress®

Pro Database Migration to Azure: Data Modernization for the Enterprise

Kevin Kline
Mount Juliet, TN, USA

Dustin Dorsey
Murfreesboro, TN, USA

Denis McDowell
Raleigh, NC, USA

Matt Gordon
Lexington, KY, USA

ISBN-13 (pbk): 978-1-4842-8229-8
ISBN-13 (electronic): 978-1-4842-8230-4
https://doi.org/10.1007/978-1-4842-8230-4

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover image photo by Ferdinand Stöhr on Unsplash

Distributed to the book trade worldwide by Springer Science+Business Media LLC, 1 New York Plaza, Suite 4600, New York, NY 10004. 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 booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

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 http://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.

Printed on acid-free paper

Love is what we have, against age and time and death, against all the powers ranged to push us down. You gave me so much— a new hope, a new heart, a new future — all far better than anything I had before. I dedicate this to you, Rachel, my jewel more precious than rubies.
—Kevin Kline

Dedicated to Antoinette, who has been my cheerleader and biggest supporter, and to Christopher, Bella, Katherine, and Josh, who tolerated geek dad lessons for so many years.
—Denis McDowell

Dedicated to my wife Sarah and our three wonderful kids Zoey, Bennett, and Ellis who put up with me through the countless hours it took to make this a reality.
—Dustin Dorsey

This is dedicated to every member of #sqlfamily that has helped me through the years. As technical professionals, I feel like we have a responsibility to "pay it forward" to help those who come after us just as those who came before us helped us. I am immensely grateful for all who have helped me through the years, and I hope my efforts to pay that forward have proved similarly helpful to others in our community.
—Matt Gordon

Table of Contents

About the Authors

Kevin Kline is a noted database expert and software industry veteran. A long-time Microsoft Data Platform MVP, an AWS Data Community Ambassador, and respected community leader in the database industry, Kevin is a founder and former president of the Professional Association for SQL Server, as well as the author of popular IT books such as *SQL in a Nutshell*. Kevin is a top-rated speaker at industry trade shows worldwide and has a monthly column at *Database Trends and Applications* magazine. He tweets at `@kekline`.

Denis McDowell has been designing and implementing technology solutions with Microsoft Data Platform technologies for over 25 years. Denis' ten years leading the Application Management practice for a managed services provider and subsequent experience consulting in financial technologies led him to develop broad and deep expertise architecting requirements-driven cloud solutions to meet the business objectives of his customers. Denis is a certified Microsoft Azure Data Platform Engineer and speaks regularly at industry events and conferences around the world. Denis is a consultant at QBE, LLC, a leading management and technology consulting organization for the federal government and defense and intelligence communities, and is currently the Principal Cloud Architect for the US Army's Enterprise Cloud Management Agency (ECMA).

Dustin Dorsey has been architecting and managing SQL Server solutions for healthcare and technology companies for well over a decade. While he has built his career in database administration, he has also spent significant time working in development and business intelligence. During this time, Dustin has gained a keen interest and specialization in cost management around the data platform both on-premises and in the cloud that he has used to save organizations millions of dollars. Dustin is an international speaker and can be seen writing articles on popular SQL websites as well as on his own blog at `http://dustindorsey.com`. He is also active in the community both as a local user group leader and event organizer.

Matt Gordon is a Microsoft Data Platform MVP and has worked with SQL Server since 2000. He is the leader of the Lexington, KY, Data Technology Group and a frequent domestic and international community speaker. He's an IDERA ACE alumnus and Friend of Redgate. His original data professional role was in database development, which quickly evolved into query tuning work that further evolved into being a DBA in the healthcare realm. He has supported several critical systems utilizing SQL Server and managed dozens of 24/7/365 SQL Server implementations. Following several years as a consultant, he is now the Director of Data and Infrastructure for Rev.io, where he is implementing data governance, DevOps, and performance improvements enterprise-wide.

About the Technical Reviewer

Joseph D'Antoni is a Principal Consultant at Denny Cherry and Associates Consulting. He is recognized as a VMWare vExpert and a Microsoft Data Platform MVP and has over 20 years of experience working in both Fortune 500 and smaller firms. He has worked extensively on database platforms and cloud technologies and has specific expertise in performance tuning, infrastructure, and disaster recovery.

Acknowledgments

First, I would like to thank my family who has put up with me and supported me throughout my professional journey as I know it has been challenging at times. A very special thanks to Jim St. Clair and the IT leadership team at Corizon who took a chance on me as a support person and hired me to my first DBA position. That chance set my career down the path it is today, and I will be forever grateful. Thank you to my former managers Justin Steidinger, Jon Buford, Waylon Hatch, and Adam Murphy who challenged and supported me in growth and learning. Thank you to Mark Josephson for being such an amazing person and for all your help, inspiration, and motivation. You played a big role in giving me the confidence to start presenting and becoming more involved in the community that has opened so many doors. Thank you to my fellow coauthors and everyone that helped make this book possible, but a special thanks to Kevin Kline. You not only presented me with the opportunity to write this book, but you have also been a great mentor and friend. Thank you to everyone in the SQL community for all that you do and letting me be a part of you. And lastly, I would like to thank God who has put me in this position and made all this possible.

—Dustin Dorsey

I would like to acknowledge my Dad, Denis Sr., for being the original geek dad and instilling an insatiable curiosity about how things work. Also the technical wizards I have been fortunate to work with and who have helped me in immeasurable ways throughout my career. Scott Brooks and Greg Gonzalez for their laser-focused attention to detail, quality, and excellence. Ken Seitz for his leadership and sponsorship of my pet projects, some of which even worked. Frank Shepherd for continually forcing me to up my engineering game. David Jones and Monty Blight for showing me how leadership should look and taking me to customer visits so I could learn how to talk both to engineers and business stakeholders. Ben DeBow for convincing me that I could make it in the consulting world. Brian Swenson, Keith Lowry, and Michael Lowry for giving me the opportunity to lead a truly transformational effort at the Army. Paul Puckett, Director of the US Army Enterprise Cloud Management Agency for showing me that one can be an agent for transformational change and understand the technical details at the same

time. And finally, I would like to thank my coauthors, in particular Kevin Kline, who quite literally started me on my SQL Server journey all those years ago when I saw him speak about performance tuning at a hotel in Charlotte, NC. I left that session hooked and dove head first into SQL Server technologies. The rest is history.

—Denis McDowell

Foreword

More than ever, companies are looking to get an edge. They are looking to modernize their use of technology to respond to events in their industry and across the globe. Whether it is cybersecurity threads or the need to drive innovation at a rapid pace, modernization, especially for data, is critical to maintain or gain market share or just build great products for all of us. Modernize means to do something different, something transformative. One solution to achieve these goals is to migrate your data platform to the cloud. Microsoft Azure provides an ideal destination for cloud migration, especially when it comes to SQL Server.

In this book, the authors give you an end-to-end experience on how to migrate to Azure SQL, a family of cloud services based on SQL Server. Many today look at migration as a very technical exercise to simply move databases and objects from one location to another. This book certainly fulfils those fundamentals but goes further. I like to think of this book as a "setup for success" when it comes to migration to Azure. The authors do such a great job of mixing up practical advice and strategy with technical instructions, resources, and references. For example, this book explores the concept of team dynamics and how it can affect the success of migration. The technical steps of migration are all in this book, but it takes such a more holistic approach including budgeting, cost management, and the use of Azure Hybrid Benefit, a key tool to save money with Azure.

All throughout the book, there is an emphasis on careful planning to save you time, money, and effort. But what I loved about this book is when there is a need to dive deep, the authors are not afraid to go there. Take for example the details they provide when it comes to security and trust which are key concepts to consider for migration. Moving data is a huge part of migration, and the authors don't just show you effective techniques for data migration but give you expertise on how to validate and execute key postmigration steps.

When I look toward a great book, I look at the experience of the authors. Kevin, Matt, Denis, and Dustin have almost 100 years of combined experience in enterprise IT. Between them, they've migrated more than 250 database applications to the Azure data platform. They've built and deployed more than 20 greenfield projects on the Azure

data platform in a mix of Azure SQL Database, Azure SQL Managed Instances, and SQL Server on Azure VMs.

To many, migration just means a "shift." This book provides you with a complete experience of making migration a journey toward modernization. Modernization to do something bigger and better. To transform how your business uses data to empower the future. The future you want. The future you need.

Bob Ward

Principal Architect, Microsoft

CHAPTER 1

The Azure SQL Data Platform

As any old timer in the IT industry can tell you, planning *always* helps when managing a project from inception to conclusion. Simple projects can go much more smoothly with an appropriately simple plan, such as a checklist. At the same time, projects that are complex and involve multiple teams are quite likely to need a commensurately more sophisticated plan with the input of many subject matter experts (SMEs) from both the business and IT side of your organization to ensure success.

That is not to say that most projects actually get the level of attention to planning that they need and deserve, considering their importance to the organization. But even simple projects with relatively few steps can often go astray. Many IT practitioners have thought "I know what I'm getting into and I know what I'm doing. This'll be three clicks of the mouse and I'm done," only to find that they now have to spend three days fixing what they thought was going to be a three-hour project.

In fact, many projects go astray even when there is a good plan in place, with the likelihood of difficulties increasing in proportion to the complexity of the IT project at hand. Yet the priority that most managers put on speedy action encourages the IT practitioners to play fast and loose with what should be an orderly, standardized process. Shortcuts are taken. Teams fail to communicate. Bad decisions are implemented with a passing thought to "We'll fix that later." Blind spots are missed. Requirements are misunderstood. Testing and quality assurance steps are shortchanged, with the expectation that the tests will come up roses without the need for any fixes.

The good news is that running mission-critical database workloads in the cloud is no longer a brave, new computing paradigm. It is a tried and true approach to enterprise computing with many advantages to other methods of operating your corporate information technology. But just because the cloud has been around long enough and grown to be popular enough to wear off the jagged surfaces of the bleeding edge, it is

K. Kline et al., *Pro Database Migration to Azure*, https://doi.org/10.1007/978-1-4842-8230-4_1

not a cakewalk, especially for those of us who have spent our careers focused on locally operated data centers. That's where this book comes in.

This book is for those who know the importance of their data assets, can envision the difficulties that they might encounter moving them to the cloud, and have the wisdom and foresight to learn as many lessons, both the failures and successes, as they can from those who have gone before and successfully migrated their on-premises databases to the Microsoft Azure cloud.

In subsequent chapters of the book, we will share best practices for the most important steps in a database migration project, covering everything from an on-premises environment to the Microsoft Azure cloud.

Azure Core Services and Concepts

Broadly speaking, Microsoft Azure provides hundreds of distinct services that enable you to build and manage powerful applications using Microsoft Azure cloud services. Covering all of those services and feature offerings is far beyond the scope of this book. However, we will take a few minutes here to make sure to provide you with an overview of key Azure services and concepts.

First, Azure provides four broad categories of services: Infrastructure, Platform, Security and Management, and Hybrid Cloud. Each of these categories contains many more point services and products. For example, the Security and Management category contains one product that you'll use on a daily basis, the Azure Portal.

Among these many individual services, we have the *core services*, which include *Compute*, *Network*, *Storage*, and *Database*. Compute is the bedrock foundation of Azure and includes services like Azure Virtual Machines, Azure Container Instances, Azure Kubernetes Service, and Azure Virtual Desktop. Azure networking services include Virtual Networks, VPN Gateways, Azure ExpressRoute, Azure Load Balancer, and more. Storage services include Azure Blob Storage, Azure Disk Storage, Azure File Storage, and other services. Finally, and perhaps most importantly for this book, we have a variety of data services like Azure Cosmos DB, Azure SQL, Azure SQL Managed Instance, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure's big data and analysis services.

Conceptually, there are a few important details you should learn about building and deploying applications to Azure, as shown in Figure 1-1. When you build an application in the Azure cloud, you'll provision the services and features you want by first creating

an Azure account, that is, your global unique credentials used to sign on Azure and access your subscriptions. (Same people refer to this as a *tenant*). An account/tenant is associated with a single entity, such as a person, company, or organization. You are also likely to manage your Azure account using Azure Active Directory (AAD) for authentication and authorization, providing granular permissions to the services you and others in your organization are using in the cloud.

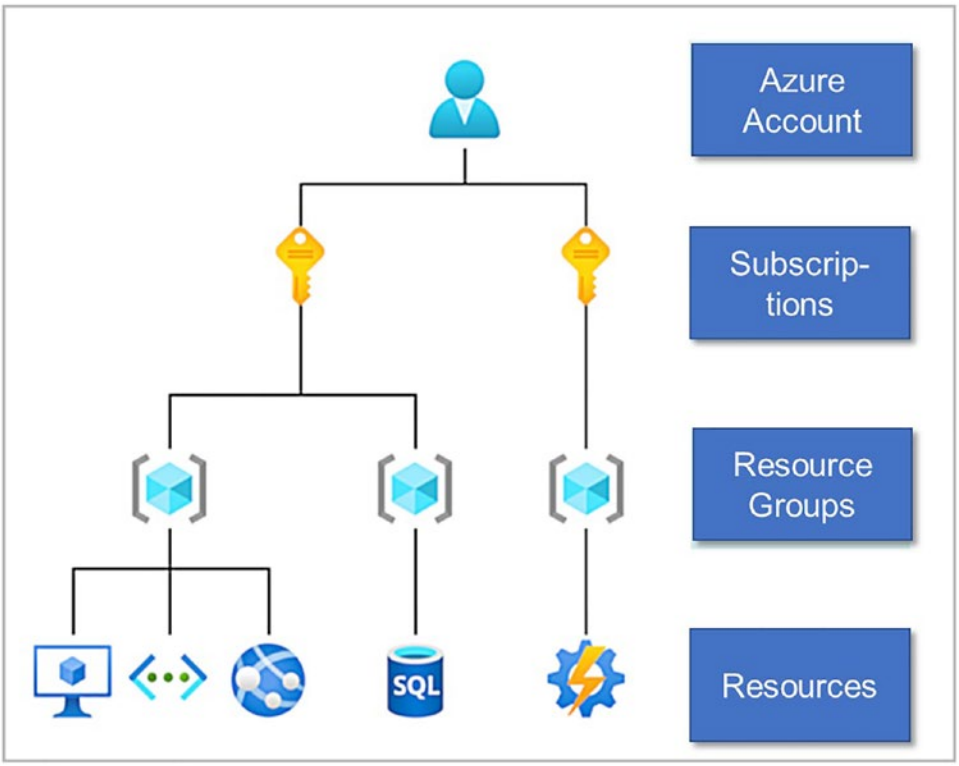

Figure 1-1. *Azure provisioning of services built atop Azure Core Services*

As we see in Figure 1-1, the hierarchy within Azure enforces a one-to-many relationship. One tenant/account can have multiple subscriptions, but subscriptions belong to only one tenant/account. A subscription can have many resource groups, but a resource group belongs to only one subscription. And so on.

Following the creation of your enterprise's Azure *account*, you can create one or more *subscriptions* for billing purposes and to provide separation and control when managing how you spend your budget in the cloud. For example, you might create separate subscriptions for your Dev and QA departments, with additional subscriptions

to support billing directly to the budgets of your organization's business units. You may also want to specify one or more Azure policies at the subscription level, for example, to send the Azure administrator a notification that a specific budget quota is close to reaching its limit. You can also associate one or more Azure subscriptions so that they trust your AAD tenant.

Most organizations have multiple subscriptions, but not all. Within your subscriptions, you can manage specific resources by defining and provisioning *Azure Resource Groups*. Resource groups are logical collections of resources belonging to the same application stack, environment, and life cycle. For example, a typical resource group for the purpose of database migration to Azure would include the Azure core services mentioned earlier, like database, storage, network, and compute resources. But your resource groups may also include many other Azure services so that you can access a variety of other specific capabilities when you build your Azure solution.

Resource groups from one subscription are isolated from other resources in other subscriptions. It's good practice to use a resource group to logically aggregate resources that share the same life cycle for easy deployment, updates, and deletions. Resource groups store metadata in a specific location, which can be very important if you must meet compliance regulations concerning the physical location where your data is stored. After creating a resource group, you can use the Azure Portal or Azure Resource Manager (ARM) templates to deploy one or more resources to the resource group. You might then also *lock* the resource group so that no critical resource can be altered or deleted.

If you want to further refine the management of your Azure environment with as many as 50 *tags,* composed of a name/value pair, you can apply tags to a resource, a resource group, or a subscription. Examples for tags might include "Environment=Production" or "Owner=Finance". Tags have size limits and are case-sensitive. Microsoft recommends managing tags in a hierarchy as described at Tag resources, resource groups, and subscriptions for logical organization - Azure Resource Manager | Microsoft Docs.

Within this construct, you can also assign *Azure role-based access control* (Azure RBAC) to one or more users, groups, service principals, or managed identities. Azure RBAC provides a wide selection of built-in roles, like *Owner* or *Contributor*, so that you can effectively manage access to the Azure resources in your subscriptions. These built-in roles are frequently all you will need for proper user authorization, but you can also create your own Azure *custom roles* to further refine resource authorization. A full discussion of these concepts is beyond the scope of this book, but check online for a full discussion at Azure RBAC documentation | Microsoft Docs.

Finally, this section would not be complete without a discussion around some basic concepts of designing your Azure architecture to support high availability and disaster recovery as required for your environment. An in-depth discussion of designing Azure infrastructure for high availability is beyond the scope of this book, but we did want to make sure some basic concepts and terms were covered in our introduction as they can, if needed, be a part of the Azure architecture you deploy to support the migrations discussed in depth in the rest of this book.

Understanding what an Azure region is will be fundamental to any deployment decisions you make going forward. Azure regions are data centers deployed, as Microsoft states it, "within a latency-defined perimeter." They are connected to each other via dedicated networks. That means that while a region may be a singular place in our minds, it can be comprised of multiple data centers. This leads us directly into the concept of availability zones in Azure. Azure availability zones are separate locations within each Azure region – meaning that local failures in a single data center should not impact the other physically separate locations within the availability zone. Microsoft guarantees that a minimum of three separate availability zones will be present in all availability zone–enabled regions. If a service you are provisioning offers you zone redundancy, that means it is taking advantage of the availability zones on offer in the region you've chosen for deployment.

An additional concept that is useful to understand is the concept of paired regions in Azure. While not all services support this cross-region replication, it is another layer to consider as you design your deployment for high availability. For services that support this concept, it means those services will be supported not just within the availability zones in a region but outside the region with another region as well. It further minimizes the possibility of a local (or regional) event impacting the availability of your Azure-deployed services.

In layman's terms, these concepts are important to understand because taking advantage of them can go a long way toward eliminating single points of failure in your deployed resources. Major cloud providers have designed amazing redundancy and resiliency into their systems, but we cannot rely solely on that when deploying resources. As always, understanding the trade-off between high availability, the cost to support it, and the need for it is critical. If that is a consideration for your project, understanding the concepts discussed in the last three paragraphs will be helpful for you.

Migration and Modernization

There are many drivers and motivations that push organizations to undertake a migration to the cloud. Sometimes, organizations might be motivated by strong internal considerations relating to their technology infrastructure, like

- Simplifying their business by no longer running their own data centers.
- Getting extended end-of-life support for important products. For example, SQL Server 2008 on-premises is no longer supported by Microsoft, except if it is housed in an Azure VM where it has extended support.
- Access to highly scalable infrastructure in cloud data centers so that they can provide much better support to requirements for a highly elastic application design.
- Easier management and IT operations for hybrid environments.

Other business-centric motivators often come into play with migrations to the cloud, like

- Ease of operations when expanding operations in new markets, particularly when those markets require data residency, that is, that data collected for their citizens is stored in data centers within their national jurisdiction. For example, Germany requires that international organizations that collect data on German citizens must process and store that data in Germany.
- Heightened security and compliance capabilities, such as those offered by Microsoft Azure, which far exceed those common to many self-run data centers.
- Business functionality improvements by taking advantage of cloud-native application designs and other developer and application innovations, like microservices and containers.

These are excellent reasons to move to the Azure cloud. We'll also spend some time later talking about failed migration projects and how they went wrong in an upcoming section entitled "Cautionary Tales." But first, let's take a quick look at the most common high-level migration strategies.

Migration and Modernization Strategies Made Simple

Jargon is always a big impediment for those who are diving into a new arena of knowledge, particularly in vocational and professional situations. Cloud migration and modernization are no different.

Before going too far, let's cover some of the terminology describing migration strategies at their highest level, since we will refer back to these terms frequently throughout the book. Let's take a look at Table 1-1 to see a quick description of the most prominent migration strategies.

Table 1-1. *Terms for cloud migration scenarios*

Term	Description
Rebuild	Take the existing stack and convert it to one of the Azure PaaS options, such as Azure SQL Database or Azure SQL Managed Instance. This strategy may represent a high level of new code, database designs, and application architectures to take advantage of the new capabilities of the PaaS environment.
Replace	Take an existing stack from your data center and convert it to a wholly SaaS solution. This is the most time-consuming and difficult strategy, since it involves an entire rewrite of the application, supporting services, and possibly database design and database programmable modules, like stored procedures and user-defined functions. When done well, the replace strategy (also known as "cloud native") can offer the greatest performance improvements and new features, but it also includes the greatest risk.
Rehost	Take an existing stack from your data center, when it is managed entirely as Virtual Machines (VMs), and then move them to an identical configuration of Azure VMs. Also known as "lift and shift" since this strategy represents no new code or design work, only new hosting servers.
Refactor	Take the on-premises stack, and then rewrite many of its components to take advantage of and optimize for the Azure cloud. This strategy represents a moderate level of new code, as well as new database and application designs. This is also sometimes called "rearchitect."

In addition to the common terms used to describe the high-level strategy employed to complete your cloud migration shown in Table 1-1, there are also a variety of important terms used to describe the types of cloud architecture we intend to use, as shown in Table 1-2.

Table 1-2. *Important terms for cloud architectures*

Term	Description
IaaS	Infrastructure as a Service (IaaS) is a type of cloud computing service that offers essential compute servers, storage, and networking resources on demand using Azure VMs. IaaS frees you from buying and managing physical servers. But you must still purchase, install, configure, patch, and manage your own software – including operating systems, middleware, databases, and applications. Synonymous with the term "lift and shift."
PaaS	Platform as a Service (PaaS) is a superset of IaaS. It includes the compute servers, storage, and networking of IaaS. It also includes middleware, development tools, business intelligence (BI) services, databases, and more. For an Azure migration project, PaaS refers to systems built on Azure SQL Database and Azure SQL Managed Instance.
SaaS	Software as a Service (SaaS) gives you a complete cloud-centric software solution that you purchase on a pay-as-you-go basis from Microsoft Azure. Examples of SaaS products include email, travel booking applications, customer relationship management (CRM) applications, and ride-hailing services. Azure takes care of all of the underlying infrastructure, middleware, application software, and application data that are running in the cloud, as well as ensures the availability of the app, and secures the app and your data. SaaS applications are usually brand new, called *greenfield applications*, although we have seen some customers use Azure Service Fabric to rebuild older, monolithic applications (usually based on IIS/ASP.NET) into an elegant, cloud-based application using multiple, manageable microservices using domain-driven design principles.
Stack	The infrastructure + data and databases + application under consideration.
Landing Zone	A landing zone is your initial Azure environment for hosting the stack you plan to migrate, preprovisioned according to your requirements and strategy. Your landing zone will typically undergo many changes, expansions, and adjustments before it is ready to support a full production workload. More details at What is an Azure landing zone? - Cloud Adoption Framework | Microsoft Docs.

For a more comprehensive review of all potential terms and concepts, as well as recommendations for building a large, enterprise-scale Azure infrastructure, read more at https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/ready/enterprise-scale/architecture.

The Five Disciplines of Cloud Governance

The Microsoft Cloud Adoption Framework provides a comprehensive set of business and technology guidance to better organize and align your business and technical strategy, provide best practices, and offer tools to help you achieve a successful migration. Investing in strong cloud governance is worth every penny. Not only does a cloud governance framework help you envision business risks, it also helps you create well-considered policies and processes to support with and align to the needs of your organization. (Read more of the Cloud Adoption Framework at https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/.)

When discussing a cloud migration project, it is helpful to consider important aspects of the project outside of technology alone. As pointed out in the Cloud Adoption Framework, there are five major pillars or disciplines of governance when migrating to the cloud.

The five disciplines of cloud governance are as follows:

1. **Cost management**, in which organizations anticipate and monitor their costs in the cloud and, where needed, limit the total amount spent. When needed, the organization can also scale up resources to meet demand, in line with additional costs. In addition, cost management governance dictates that accountability is extremely important, so the migration team should also create cost accountability policies and processes and authorize staff. Chapters 3 and 4 go into cost management in great detail.

2. **Security baseline**, where the organization ensures compliance with IT security requirements by applying a baseline of security policies and technologies to all migration efforts. Chapter 8 discusses elements of building an Azure security baseline.

3. **Resource consistency**, which ensures consistent application of resource configurations for better enforcement of policies around on-boarding new migration projects, backup and recovery, and discoverability for Azure-based resources. Chapter 6 goes into the details about ensuring you have enough resources for your Azure cloud landing zone.
4. **Identity baseline**, whereby our organization ensures a baseline for identity management and accessibility to the Azure environment through consistently applied and enforced role definitions and assignments. Chapters 7 and 8 cover some elements of identity management applicable to an identity baseline.
5. **Deployment acceleration**, offering a centralized, consistent, and standardized migration and postmigration experience through the use of deployment templates, such as those provided by Azure Resource Manager (ARM) templates. Chapters 9 through 11 cover a variety of best practices that accelerate your Azure deployment.

The Cloud Adoption Framework and its operating model for governance is an expansive set of documentation with many valuable recommendations, guidelines, and best practices. We recommend that you study this information as soon as convenient or, at the latest, after your first migration pilot project but before you have many applications running in Azure. Read more at The Five Disciplines of Cloud Governance - Cloud Adoption Framework | Microsoft Docs.

Cautionary Tales

A smart person learns from their own mistakes, but the wise person learns from the mistakes of others. In the interest of wisdom, we want to make sure you are aware of the many reasons why database migration projects fail. Throughout the rest of this chapter, we will tell you a few cautionary tales to help you know what to avoid.