



Azure SQL Revealed

A Guide to the Cloud for SQL Server
Professionals

—

Bob Ward

Foreword by Scott Guthrie

apress®

Azure SQL Revealed

A Guide to the Cloud for SQL Server Professionals

Bob Ward

Foreword by Scott Guthrie

Apress®

Azure SQL Revealed: A Guide to the Cloud for SQL Server Professionals

Bob Ward
North Richland Hills, TX, USA

ISBN-13 (pbk): 978-1-4842-5930-6
<https://doi.org/10.1007/978-1-4842-5931-3>

ISBN-13 (electronic): 978-1-4842-5931-3

Copyright © 2021 by Bob Ward

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

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

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

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 via the book's product page, located at www.apress.com/9781484259306. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

This book is dedicated to all at Microsoft who have worked tirelessly over the last decade to make Azure the world's computer.

Table of Contents

About the Authorxv

About the Technical Reviewerxvii

Forewordxix

Acknowledgmentsxxi

Introductionxxiii

Chapter 1: SQL Server Rises to the Clouds 1

 CloudDB 2

 The Red Dog..... 3

 The Azure Services Platform..... 4

 The Road to SQL Azure..... 8

 SQL Data Services 9

 SQL Azure Is Born..... 13

 The SAWA Project..... 16

 The Virtual Machine Initiative..... 16

 Becoming Azure SQL Database..... 19

 The Sterling (SAWA2) Project..... 20

 New Editions, the DTU, and Previews 25

 Intelligent Performance and the MDCS 26

 Advanced Data Security and the ILDC Team..... 27

 A Pane for the Future Called Ibiza 28

 A New Engineering Model for Azure..... 29

 Bending Azure SQL Database..... 30

 Lifting Customers to the Cloud..... 31

TABLE OF CONTENTS

Project Socrates Goes Hyper..... 32

Azure SQL Today 35

Chapter 2: What Is Azure SQL? 39

 The Azure Ecosystem..... 40

 Azure Accounts and Subscriptions 40

 The Azure Portal 42

 The Azure Marketplace..... 45

 Azure API and CLI 46

 Azure Resource Manager (ARM)..... 47

 Azure Monitor 50

 Azure Regions and Datacenters 50

 Azure Service-Level Agreement (SLA)..... 51

 What Is the Azure SQL?..... 52

 IaaS vs. PaaS..... 52

 The Azure SQL Lineup..... 53

 Azure SQL Managed Instance 55

 Managed Instance Capabilities 56

 Managed Instance Options and Limits 57

 Managed Instance Pools 59

 Managed Instance vs. SQL Server on Azure Virtual Machine 60

 Customers Using Managed Instance 61

 Azure SQL Database..... 62

 Azure SQL Database Capabilities 62

 Azure SQL Database Options and Limits 63

 Azure SQL Database vs. Azure SQL Managed Instance 70

 Customers Using Azure SQL Database 71

 Interfaces for Azure SQL 72

 Azure Portal 72

 az CLI 74

 PowerShell 75

 REST API 75

TDS and T-SQL.....	76
SQL CLI	77
SQL Server Management Studio (SSMS).....	77
Azure Data Studio (ADS)	78
Visual Studio Tools.....	79
Summary.....	80
Chapter 3: SQL Server on Azure Virtual Machine	81
Deploying	82
Pricing	82
SQL Server Gallery Images	83
Resource Group, Region, and Availability	86
Virtual Machine Sizes.....	89
Account, Port, and OS Licensing.....	93
Making Configuration Choices As Part of Deploy	94
Deploy!	108
Navigating in the Portal.....	111
Connecting to Your VM.....	113
Exploring the SQL Server Installation	115
Deploy on Your Own.....	117
Using a CLI and ARM Template	118
Reserved Instances and Dedicated Hosts	120
Migrate Using Azure Migrate	121
Deploying SQL Server on Linux with Azure Virtual Machine.....	122
Deploying SQL Server Containers.....	123
SQL Virtual Machine Resource Provider.....	124
Configuration.....	126
Stopping vs. Deallocating.....	126
Resizing	127
Security	127
Other Config Options	128
Maximizing Storage Performance.....	128

TABLE OF CONTENTS

- Performance Monitoring 130
 - Azure Metrics 130
 - Logs 132
 - Insights 133
- Networking..... 134
- HADR..... 135
 - Azure Storage 136
 - Backups..... 136
 - Always On Failover Cluster Instance 137
 - Always On Availability Groups 137
 - Go Further with Azure Availability..... 138
 - SQL Server and Linux Availability 138
- Summary..... 139
- Chapter 4: Deploying Azure SQL 141**
 - Pre-deployment Planning..... 142
 - New Deployment or Migration..... 142
 - Making Deployment Choices 143
 - Consider Resource Limits..... 148
 - Deploying Azure SQL Managed Instance..... 149
 - Deployment and Options 150
 - Deploying with a CLI 159
 - Implementation Details 159
 - Connecting and Verifying Deployment..... 161
 - Migrating to Azure SQL Managed Instance 168
 - Assessment and Planning 168
 - Migration 170
 - Application Changes..... 170
 - Post Migration 171
 - Deploying an Azure SQL Database 171
 - Deployment and Options 171
 - Deploying Business Critical 183

Deploying Serverless	185
Deploying Hyperscale	186
Deploying with a CLI	187
Implementation Details	188
Connecting and Verifying Deployment.....	194
Migrating to Azure SQL Database	200
Summary.....	202
Chapter 5: Configuring Azure SQL	203
Configuring Azure SQL Managed Instance.....	204
sp_configure	204
Trace Flags	205
Tempdb.....	205
Master and Model.....	206
Configuring Edition	206
Networking Configuration.....	206
Configuring Databases	208
Configuring Azure SQL Database	210
Creating New Databases	210
Altering Databases	212
Network Configuration.....	213
Configuration Restrictions.....	214
Azure SQL Managed Instance Restrictions.....	214
Azure SQL Database Restrictions	218
Azure SQL Space Management.....	219
Azure SQL Managed Instance Space Management.....	219
Azure SQL Database Space Management	220
Loading Data	221
Keep These in Mind	221
bcp.....	222
BULK INSERT and OPENROWSET	223
SQL Server Integration Services (SSIS)	224

TABLE OF CONTENTS

BACPAC.....	226
Database Copy.....	227
RESTORE to Managed Instance	228
Spark Connector	229
Azure Data Factory (ADF).....	229
SQL Data Sync	230
Replication Subscriber	230
Updating Azure SQL	231
Maintenance of Azure SQL	231
New Features and Capabilities in Azure SQL.....	233
Summary.....	234
Chapter 6: Securing Azure SQL.....	235
Security Capabilities and Tasks	237
Security Capabilities.....	237
Security Tasks	238
Network Security	239
Azure SQL Managed Instance Network Security	240
Azure SQL Database Network Security	243
Authentication and Access.....	256
Azure Role-Based Access Control (RBAC)	256
Authentication for Azure SQL Managed Instance	257
Authentication for Azure SQL Database.....	259
Set Up and Configure Access	266
Protecting Your Data	266
Encrypting Connections.....	266
Transparent Data Encryption (TDE).....	267
Always Encrypted	272
Dynamic Data Masking (DDM).....	272
Monitoring Security.....	274
Monitoring the Azure Ecosystem	274
Auditing Azure SQL Managed Instance.....	276

Auditing Azure SQL Database	277
Advanced Data Security	286
Data Classification	288
Vulnerability Assessment	295
Advanced Threat Protection (ATP)	299
Azure Security Center	301
Summary	303
Chapter 7: Monitoring and Tuning Performance for Azure SQL	305
Performance Capabilities	306
Max Capacities	307
Indexes	310
In-Memory OLTP	311
Partitions	311
SQL Server 2019 Enhancements	312
Intelligent Performance	312
Configuring and Maintaining for Performance	313
Tempdb	313
Database Configuration	314
Files and Filegroups	314
Max Degree of Parallelism	315
Resource Governor	315
Maintaining Indexes	316
Maintaining Statistics	316
Monitoring and Troubleshooting Performance	317
Monitoring Tools and Capabilities	317
Dive into DMVs and Extended Events	319
Performance Scenarios	324
Azure SQL Specific Performance Scenarios	343
Accelerating and Tuning Performance	346
Scaling CPU Capacity	346
I/O Performance	356

TABLE OF CONTENTS

Increasing Memory or Workers	356
Improving Application Latency	357
Tune Like It Is SQL Server.....	358
Intelligent Performance	363
Intelligent Query Processing.....	363
Automatic Plan Correction	364
Automatic Tuning.....	365
Summary.....	372
Chapter 8: Availability for Azure SQL	373
HADR Capabilities	375
Automatic Backups and Point-In-Time restore.....	375
Built-In High Availability	375
Azure Redundancy.....	376
Geo-replication and Auto-failover Groups	376
Database Availability and Consistency	376
SQL Server Replication.....	377
Backup and Restore.....	377
Automatic Backups.....	378
Point-In-Time restore	381
Long-Term Retention Backups	388
Geo-restore of Databases.....	391
Restore Backups from Deleted Databases	392
Restore in Azure SQL Managed Instance.....	393
Built-In High Availability	394
General Purpose High Availability	395
Business Critical High Availability.....	398
Hyperscale High Availability	404
Go Further with Azure	410
Zone Redundancy.....	410
Geo-replication	413

Auto-failover Group	419
Azure SQL SLA.....	428
Database Availability and Consistency.....	430
Database Availability	430
Accelerated Database Recovery (ADR).....	431
Database Consistency	431
Monitoring Availability.....	432
Instance, Server, and Database Availability	432
Backup and Restore History	433
Region, Data Center, and Service Availability	434
Replica Status	436
Failover Reasons	438
System Center Management Pack for Azure SQL.....	439
Summary.....	439
Chapter 9: Completing Your Knowledge of Azure SQL.....	441
Surface Area of Azure SQL	442
Linked Servers and Cross-Database Queries	443
External Tables	444
Database Mail.....	444
Service Broker	445
Full-Text Search.....	446
Machine Learning Services	446
What Is Missing?	447
Job Management	447
SQL Server Agent.....	448
Elastic Jobs	449
Azure Automation	457
Supporting Azure SQL	458
Handling Errors.....	458
Stack Dumps	460

TABLE OF CONTENTS

Troubleshooting Resources in the Azure Portal 460

UserVoice..... 465

Azure SQL Best Practices..... 465

 Security Playbook..... 465

 Best Practices for Performance..... 467

 Azure Advisor..... 467

 Stay in Touch with Our Team 469

Summary..... 470

Chapter 10: Go Big with the Cloud..... 471

 Integration with Azure Services 471

 Power Platform..... 472

 Azure Search 476

 Stream Analytics..... 481

 Azure Architectures and Solutions 481

 Azure Synapse 483

 Azure Arc..... 483

 Summary..... 484

Index..... 487

About the Author



Bob Ward is a Principal Architect for the Microsoft Azure Data team, which owns the development for all SQL Server versions. He has worked for Microsoft for more than 27 years on every version of SQL Server shipped from OS/2 1.1 to SQL Server 2019, including Azure SQL. Bob is a well-known speaker on SQL Server and Azure SQL, often presenting talks on new releases, internals, performance, and Azure SQL fundamentals at events such as PASS Summit, Red Hat Summit, Microsoft //build, SQLBits, SQLIntersection, Microsoft Inspire, and Microsoft Ignite. You can follow him at [@bobwardms](#) and [linkedin.com/in/bobwardms](#). Bob created and produced training for Azure SQL including Azure SQL Fundamentals, Azure SQL for Beginners, and the Azure SQL Bootcamp. Bob is the author of the Apress books *Pro SQL Server on Linux* and *SQL Server 2019 Revealed*.

About the Technical Reviewer



Joe Sack is a Principal Program Manager at Microsoft, focusing for the last four years on the Intelligent Query Processing feature family for Azure SQL Database and SQL Server. He has worked as a SQL Server professional since 1997 and has supported and developed for SQL Server environments in financial services, IT consulting, manufacturing, retail, and the real estate industry.

Over the years, Joe has published and edited several SQL Server books and white papers. His first book *SQL Server 2000 Fast Answers for DBAs and Developers* was published in 2003. He also started the T-SQL Recipe series, including *SQL Server 2005 T-SQL Recipes* and *SQL Server 2008 Transact-SQL Recipes*. He recorded 13 Pluralsight courses, including “SQL Server: Troubleshooting Query Plan Quality Issues,” “SQL Server: Transact-SQL Basic Data Retrieval,” and “SQL Server: Common Query Tuning Problems and Solutions.”

His Twitter handle is [@JoeSackMSFT](https://twitter.com/JoeSackMSFT), and you can find Joe speaking at most major SQL Server conferences.

Foreword

Cloud computing has become a pivotal part of business and our world. Azure, the world's computer, is at the forefront of delivering cloud computing at the scale the world needs. Azure SQL has been part of Microsoft's cloud lineup since the beginning. From SQL Services to SQL Azure to the powerful lineup of Azure SQL cloud services, databases have always been a core part of Azure.

In this book, Bob Ward, an architect on my team who for 27 years has been an integral part of the SQL Server story for Microsoft, leads you on a journey to learn Azure SQL starting with the compelling history of Azure and Azure SQL. Then he describes in detail the Azure SQL lineup, including SQL Server for Azure Virtual Machine, Azure SQL Managed Instance, and Azure SQL Database. A complete chapter is available for you to understand the details of using SQL Server in Azure Virtual Machine.

The heart of the book gives you what you need to deploy and configure Azure SQL to meet your requirements. You will then learn how to secure your data, maintain performance, and achieve high availability with your Azure SQL investments. I love how Bob compares and contrasts our SQL Server product you know so well to Azure SQL deployment options. But you will also learn the innovation of Azure SQL that can truly provide your business new value, including built-in High Availability, Automatic Tuning, Advanced Threat Protection, Serverless computing, and Hyperscale databases.

With 60+ Azure cloud regions around the world and more coming every quarter, Azure is everywhere you need it with scale, compliance, and resiliency for even the most demanding applications. Azure SQL services are a core element of our strategy to provide databases at scale. I believe this book will give you the knowledge you need to translate your skills with SQL Server to maximize the capabilities of the Azure SQL data platform and meet the needs of your migrations or new database applications.

—Scott Guthrie, Executive Vice President, Microsoft, Cloud and AI

Acknowledgments

I want to first thank God for giving me the endless gift of grace through his son Jesus Christ. God's grace blesses me every day, including the honor of being married to Ginger Ward. My wife Ginger continues to show me the example of what it is like to show kindness, grace, and respect to everyone she meets. Her support throughout my time writing this book kept me going. My sons, Troy and Ryan Ward, bless me every day as I see them carry on what I've taught them about integrity and responsibility.

Publishing a book is not easy, and without Jonathan Gennick and Jill Balzano from Apress, I could not have made it. Thank you both for your professionalism and kindness. I owe a huge thanks to my technical reviewer Joe Sack. If you have never met Joe Sack, I hope someday you do. Not only is he one of the nicest people I know, but a brilliant engineer. He brought the right balance of pointing out what could work better with compliments on many of the chapters.

I want to also thank my leadership at Microsoft for giving me this opportunity, including Scott Guthrie for writing the foreword, Rohan Kumar for all his guidance and wisdom, and to my manager Asad Khan who always supports me in everything I do at Microsoft.

They say "it takes a village" to build something big, but to write a book like this it takes an army. It is hard to call out every individual who helps you as you create a book like this, so I'll first start by thanking all the software engineers and program managers across the Azure SQL team in Redmond, Israel, and Serbia. I do want to mention a few specific individuals, and it all starts with Anna Hoffman. Anna and I worked together the entire calendar year of 2020 on Azure SQL, all remote. Her patience with what she calls my "hyperfocused" work amazes me every day. But more important, she possesses what I call an "old school" work ethic at a very young age. I find that hard to see these days. I also want to thank my colleagues Buck Woody and Marisa Brasile for all their guidance, encouragement, and help promoting our work in Azure SQL. I want to pay special thanks to the folks I interviewed for the first chapter of the book to tell you the history of Azure SQL, including Ted Kummert, Rohan Kumar, Ajay Kalhan, Peter Carlin, Brian Chamberlain, Conor Cunningham, Ron Matchoro, Guy Bowerman, Drazen Sumic, Morgan Oslake, Cristian Diaconu, Steve Lindell, Mark Russinovich, Evan Basalik, and Keith Elmore.

ACKNOWLEDGMENTS

There were many members of the Azure SQL and other engineering teams that helped me with quotes and questions that I want to thank, including Dimitri Furman, Denzil Ribeiro, Sanjay Mishra, Emily Lisa, Joachim Hammer, Andreas Wolter, Girish Mittur Venkataramanappa, Alain Dormehl, Abdul Sathar Sait, Mine Tokus, Ajay Jagannathan, Borko Novakovic, Hans Olav Norheim, Misha Kolianko, Pratyush Rawat, Scott Kim, Stojan Rakic, Venkata Raj Pochiraju, Yadi Reyes, Jovan Popovic, Sherif Mahmoud, Tomer Rotstein, Andrey Karpovsky, Rohit Nayak, Vassilis Papadimos, Dejan Krakovic, Craig Freedman, Silvano Coriani, Davide Mauri, Hanuma Kodavalla, Ebru Ersan, and Srini Acharya.

Our marketing team has always been very supportive of my work, so a big thanks to John “JG” Chirapurath, Wisam Hirzalla, Eric Hudson, Debbi Lyons, and Miwa Monji.

And finally thanks to you, the community, who supports and helps us promote the amazing story of SQL Server. Many of you are now engaging with Azure, so my hope is the timing of this book is perfect for you to begin your journey with Azure SQL and empower you to put the same energy and passion behind it as you have with SQL Server for so many years.

Introduction

In November of 2019, I had just released my second book on SQL Server called *SQL Server 2019 Revealed* (my first book was *Pro SQL Server on Linux* which was released in 2018). It was only one week after we had also just launched the SQL Server 2019 release. I had also delivered some eight sessions at the recent PASS Summit conference in Seattle. Therefore, near the end of calendar 2019, I wanted to reflect back on what was successful but also look toward the future.

I found myself sitting in a room with my manager Asad Khan reviewing the work I had done over the last year and what might be the next focus for me (that is how we will roll at Microsoft; yes, we were celebrating the release of SQL Server 2019, but we are always looking at what is our next big thing). We agreed that for at least the foreseeable future I needed to focus on delivering the message of SQL Server 2019, Containers, Linux, and Kubernetes to customers and our internal teams.

I then asked a very dangerous question: “What about Azure?” Asad owns the program management for SQL Server and Azure SQL. He paused for a few seconds and then asked me, “Bob, what do you mean? Do you mean you want to get involved in Azure?” I had been thinking about this for some time. Back in the fall of 2019, I did a roadshow with Buck Woody and Anna Hoffman called “Ground to Cloud” (see the workshop for yourself at <https://github.com/microsoft/sqlworkshops/tree/master/SQLGroundToCloud>). During the roadshow, I was also thinking ahead despite the fact that I was still finishing the *SQL Server 2019 Revealed* book and part of our overall launch work for SQL Server 2019. So I told Asad, “I think one thing I can help with the cloud is to do an assessment of Azure SQL from the perspective of a SQL Server expert.” Why not take my 27+ years of experience on SQL Server and see how it lines up with the cloud? Asad in his usual calm demeanor said that he thought this would be an excellent idea, but I think he was privately excited to see my involvement.

Roll forward to early January of 2020. I had spent time over the holiday break in December of 2019 talking to my wife Ginger and my sons Troy and Ryan. “Should I go for a third book?” Ginger replied with her usual wry smile “Why not? Don’t you have a hang for this book writing thing now?” I thought it over for a bit and then decided to dive in. After all, 2020 marks the tenth anniversary of Azure and Azure SQL. I thought

INTRODUCTION

this introduction would include all the places I travelled while writing this book, as I had with my previous two books. However, there was a historic worldwide event you may have heard of called COVID-19 which disrupted not only my travel but has also affected so many people. I grieve and pay respects to all of those reading this book for themselves and any of their loved ones affected. The time not travelling did allow for me to have more focus to write this book, so I hope you like the final result. All of this book was written in my home office in North Richland Hills, Texas, but I was connected to the world throughout the writing.

This book is the culmination of work for an assessment of Azure SQL from the perspective of SQL Server. However, even if you don't know SQL Server, I believe you will gain much knowledge to learn Azure SQL. The book has many examples embedded into the text you can use, but you can also download code examples from my GitHub repository at <https://github.com/Microsoft/bobsql>.

The book starts out with a history of Azure and Azure SQL. I think understanding the history of something can help you gain knowledge of current capabilities and what is possible. I had fun writing the first chapter as it involved some fairly intensive research and interviews with some names you will find familiar.

I then will take time to explain what Azure SQL means. This is a great chapter to get started so you understand what is possible with Azure SQL cloud services, including the Azure ecosystem. This chapter will also help guide your decisions as you choose which Azure SQL option best fits your requirements. I recommend anyone reading this book to review this chapter.

One of the choices for Azure SQL is Azure Virtual Machine. Since the experience of Azure Virtual Machine (VM) inside the guest OS is very much like SQL Server on-premises, I chose to dedicate a single chapter to the subject instead of including Azure VM in the rest of the book. This chapter is great though if Azure VM is your target as I talk about deployment, networking, storage, and performance. The rest of the book covers Azure SQL Managed Instance and Azure SQL Database as Azure SQL.

One of the first things you do with SQL Server is install the software and configure it. Therefore, that is where your Azure SQL journey begins: learning the details and options to deploy and configure Azure SQL. I think you will find this is a fairly thorough discussion of the topic, because in some cases, making the right decisions during deployment will save you time and money. Even if you are familiar with the basics of deploying Azure, I recommend you review these chapters. You might find some interesting details you didn't know about.

The heart of the book then takes you on a journey to learn the core of Azure SQL, which are security, performance, and availability. Every customer I've talked to that has mastered SQL Server knows these three areas, and they represent the core of the SQL Server engine. Therefore, when you read these chapters, you will understand the capabilities and tasks of Azure SQL compared to SQL Server. These chapters are the largest in the book and contain the most examples.

You will finish the book learning "What else." There are some topics like Job Management that don't fit into security, performance, and availability, so you will round off your knowledge in the second to last chapter. One of the goals of this book is to get the SQL Server professional comfortable with Azure SQL. However, I want readers to end their journey with the book in the last chapter to see what is possible beyond the fundamentals because you are now in the world's computer, Azure.

This book is intended to help you get to Azure SQL, stay on Azure SQL, and use Azure SQL to its maximum potential. So gear up and start learning. Welcome to the world's database: Azure SQL.

Bob Ward
North Richland Hills, Texas

CHAPTER 1

SQL Server Rises to the Clouds

In late 2005, Microsoft as a company was humming (I'm a little biased here) in the enterprise space and so was the SQL Server product. In October of 2005, we were close to releasing SQL Server 2005 (code name Yukon) which was unfortunately 5 years in the making (that is a story for another book; just ask Paul Randal). I was in Microsoft Support in those days, and despite the delay in getting SQL Server 2005 to market, I was very proud of the release. Windows, Windows Server, Office, and Xbox 360 were all popular products from Microsoft.

In October of 2005, an architect new to Microsoft named Ray Ozzie sent an internal email to several executives at Microsoft (which eventually was sent to all employees including a 12-year veteran named Bob Ward) called *The Internet Services Disruption* (the email leaked to the Web fairly quickly which you can read at www.cnet.com/news/ozzie-memo-internet-services-disruption/). I remember hearing about the email leak and some of its contents as an employee but didn't pay much attention. Wasn't the Internet just for email and web browsing? In that email, Ray Ozzie painted a picture of Microsoft becoming a *cloud provider* vs. just a "traditional software company." Microsoft only really had a few "Internet services" offerings at the time which included the legendary Hotmail email service (which had existed since 1997), the Bing Search Service, and Xbox Live. The email from Ray Ozzie painted a picture for something far bigger.

One of the key statements from this email was "...All Business Groups have been asked to develop their plans to embrace this mission and create new service offerings that deliver value to customers and utilize the platform capabilities that we have today and are building for the future." Little did I know how much behind-the-scenes work would kick in within the SQL Server team to develop plans for this statement.

Ray Ozzie became the Chief Software Architect of Microsoft in the summer of 2006 (taking the role held by Bill Gates), and this email would set the stage for what would become known as *Azure*. SQL Server was destined to become a huge part of it.

CloudDB

In early 2006, Paul Flessner, Vice President of the Data Storage and Platform division of Microsoft, decided to step down as the leader of SQL Server and turn over the reins to Ted Kummert. When Ted took over to lead SQL Server, a project was already underway to look at cloud services led by Technical Fellow Peter Spiro, who was a chief architect for several SQL Server releases, including SQL Servers 7.0, 2000, and 2005. Peter formed a team which included several engineers. Among them were two architects still at Microsoft today: Ajay Kalhan and Tomas Talius. The team embarked on a project to build a cloud-based service to host databases. They called it **CloudDB**. As Ted tells it, “We needed to build a cloud version of SQL. Our goal was to build a *serverless* or Platform as a Service (PaaS) SQL. A customer wouldn’t worry about a server or VM, just a database.”

In order to build a cloud-based database service, the team needed to build out a robust design to support the concept of hosting multiple customers or “databases” isolated from each other using shared resources. This concept is called *multi-tenant*.

Note The term *tenant* can mean many things in the cloud. For CloudDB, in the beginning, a tenant referred to a database owned by a customer. You will see throughout this book the word *tenant*, but I’ll be clear about the scope of what I mean when using the term.

According to Ajay Kalhan, from the beginning the CloudDB team started working out designs to incorporate concepts such as failure detection, logical master (think of a “metadata” master, not physical), load balancing, and deployment. Early designs even looked at the idea of a “key-value store” instead of traditional relational database concepts. Not long after the team was building out the design for CloudDB, Ted assigned David Campbell to also work on the project and lead the team toward a true mission of “SQL Server in the Cloud.”

The team believe it needed an internal customer to help *dogfood* the project and prove they could host customers. That internal customer would become a public-facing cloud service called **Exchange Hosted Archive (EHA)** (an email archive solution in the cloud predating Office 365). For this internal customer, early designs to support multi-tenants (which in this case even though there was one internal customer, that customer serviced the needs of multiple customers) used a concept called *silos* where a SQL Server could host multiple databases, but tenants were partitioned within the database itself. EHA became one of the first **Software as a Service (SaaS)** services at Microsoft to use our cloud-based database service. Think of SaaS as purchasing software on a subscription basis and using the software from a hosted solution, like in Azure. You just focus on using an application hosted somewhere other than your computers. Since SQL Server hosted the back-end databases, the team forked the codebase of SQL Server 2005 to use for the service.

While the CloudDB team was working on their project with a goal to support EHA and other customers, another team at Microsoft was chartered by Ray Ozzie to look at how to host *compute services* in the cloud.

The Red Dog

In 2006, Ray Ozzie enlisted Microsoft veteran Amitabh Srivastava to lead a “Cloud OS” project in the attempt to move forward the “Internet services disruption” he had talked about a year ago. One of the first actions Amitabh took was to bring out of retirement Dave Cutler, the “father” of DEC VMS and Windows NT operating systems. As part of their initial project work, Srivastava and Cutler visited groups at Microsoft that were providing “cloud services,” including Xbox Live, Hotmail, and Bing. On one of the trips to visit Hotmail in San Jose, California, the team drove by a club called the Pink Poodle. It was Dave Cutler who famously said, “Maybe we should name our project the Pink Poodle?” The project team all agreed that would not go over well so named the project instead “Red Dog.” The name stuck (you can read more about the great history of the beginning of Red Dog at www.wired.com/2008/11/ff-ozzie/?currentPage=7 and www.zdnet.com/article/how-the-red-dog-dream-team-built-a-cloud-os-from-scratch/).

From the beginning, the Red Dog team did things differently at Microsoft to build the “Cloud OS.” They built their own “data center” in the heart of the Microsoft campus, even taking reserve power from neighboring buildings. Their goals were ambitious and still resonate today. Their main overall goal was to *build a cloud service for developers to build scalable web applications*. They also had a massive theme from the beginning: *reliability*. As Dave Cutler said back in 2008, “One of the things you did not ask is why aren’t we saying more about Azure and in the process filling the marketplace with sterling promises for the future? The answer to this is simply that the RD group is very conservative, and we are not anywhere close to being done. We believe that cloud computing will be very important to Microsoft’s future and we certainly don’t want to do anything that would compromise the future of the product. We are hypersensitive about losing people’s data. We are hypersensitive about the OS or hypervisor crashing and having properties experience service outages. So, we are taking each step slowly and attempting to have features 100% operational and solidly debugged before talking about them. The opposite is what Microsoft has been criticized for in the past and the RD dogs hopefully have learned a new trick.”

The RedDog and CloudDB teams were marching together as separate projects (ironically on the same campus only buildings apart) to support cloud services for web applications and hosted databases in the cloud. These projects were on a path to come together in 2007 and 2008 for a launch of a unified cloud service.

The Azure Services Platform

In October of 2008 at the Microsoft Professional Developers Conference (PDC) in Los Angeles, California, Ray Ozzie announced **Windows Azure**. The PDC was the pre-cursor to today’s Microsoft //Build conference ([https://en.wikipedia.org/wiki/Build_\(developer_conference\)](https://en.wikipedia.org/wiki/Build_(developer_conference))). PDC was a huge event for Microsoft for developers.

Windows Azure was launched as part of the **Azure Services Platform**. Figure 1-1 shows a snapshot of the Azure Services Platform offerings.



Figure 1-1. *The Azure Services Platform in 2008*

The Red Dog team had been cranking away since 2006 with the goal of releasing a cloud service for developers. Ray Ozzie called Windows Azure a “new Windows offering at the web tier of computing” (watch the video for yourself at www.zdnet.com/article/ray-ozzie-announces-windows-azure/). He also called Azure “Windows in the cloud.” Microsoft now would offer customers Windows on your laptop (at that time, it was Windows Vista), servers for your enterprise (Windows Server), and Windows in the cloud (Azure).

Note I sought out many folks at Microsoft on why our cloud service was named Azure. As Buck Woody, who is my colleague now but worked on Azure in the early days, tells the story, “Azure means clear blue sky with no clouds. The name just seemed right without using the word cloud in our name.”

Like the goal of the CloudDB project, when Windows Azure first released, the goal was all about scalability and availability targeting web applications in the form of a *Platform as a Service (PaaS)*. Think of PaaS as purchasing a platform to host your application or database based on a subscription where the platform is *managed* by a

provider, like Azure. With PaaS, you are typically abstracted from a host computer or virtual machine. Therefore, *Cloud Services* was the first service in Windows Azure. This type of service was known internally as PaaS V1.

Note Cloud services is still offered today in Azure. You can read more about cloud services at <https://azure.microsoft.com/en-us/services/cloud-services/>. A new service for web applications has become popular today called Azure App Service which you can read more about at <https://azure.microsoft.com/en-us/services/app-service/>.

Even though a cloud service application ran in one or more Virtual Machines, the idea was to support easy-to-scale web applications in the cloud where developers didn't focus on the details of the virtual machine but more on the application. Developers at this time for Windows were used to the Internet Information Server (IIS) feature of Windows Server. While developers didn't have to worry as much about deploying and configuring IIS, they typically had to have an administrator within their organization. While developers had some access to the Virtual Machine native OS environment for cloud services, that access was limited. It would be a few years later that Microsoft would introduce the concept of *Infrastructure as a Service (IaaS)* through Azure Virtual Machine. Think of IaaS as purchasing an infrastructure to host your virtual machine based on a subscription. You worry about the guest VM and the provider manages the host, hardware, networking, and storage.

One of the other promises of PaaS and cloud services is to create an easy-to-use concept of application deployment, configuration, and updates. Furthermore, providing capabilities for scalability, built-in high availability, and load balancing made the concept of cloud services extremely appealing to web developers. These same concepts you will see are a part of the appeal as well for Azure SQL and databases.

In order to host PaaS cloud services, an *underlying hosting system* had to be built. The Windows Azure team took the designs from the RedDog project to build this hosting system to support deployment, networking, high availability, scale, and security, as cloud services abstracted all these details from the developer. This software hosting system was known as the *Windows Fabric*. Providing the underlying hosting system for services consumed by users is the *power of the cloud*. I found this interesting slide from a talk at the PDC 2008 conference that talks about all the details required for someone to run their own *fabric* in a data center as seen in Figure 1-2.

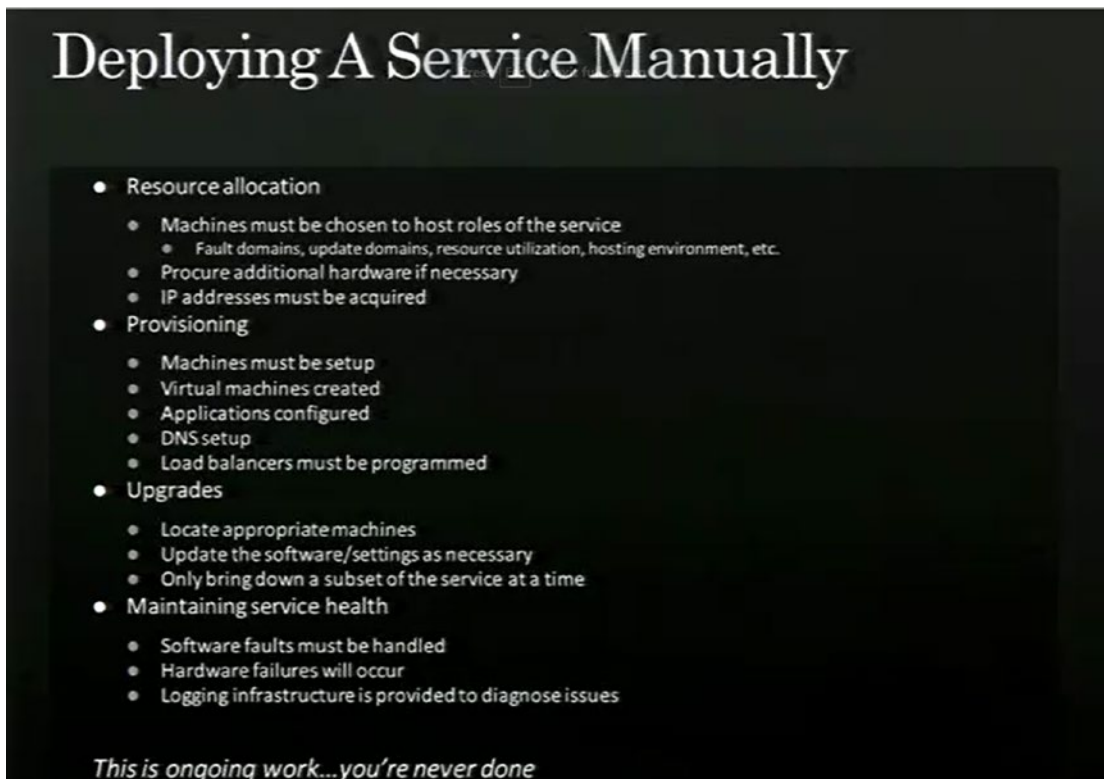


Figure 1-2. *Building your own fabric in a data center*

This slide speaks volumes for what a fabric must support for cloud services at scale. A highly available *fabric controller (FC)* is key to the system. The FC maintains a graph of the inventory of what it manages: computer, Virtual Machines, load balancers, and switches with edges being objects like network cables. One key to the fabric system is the use of a declarative model so the FC takes what you declare and implements it. Very early on, the Windows Fabric in Azure had concepts of high availability such as fault and update domains (I'll describe the importance of these in Chapters 2 and 3 of the book).

Tip The slide from Figure 1-2 comes from an excellent presentation from the PDC 2008 event which talks about Windows Fabric and the hosting environment of the original Windows Azure service. You can watch this presentation at <https://channel9.msdn.com/blogs/pdc2008/es19>. Another good resource I found on some basics of hosting and Windows Fabric comes from an interview with Azure CTO Mark Russinovich at <https://searchcloudcomputing.techtarget.com/blog/The-Troposphere/How-Azure-actually-works-courtesy-of-Mark-Russinovich>.

Windows Fabric is today known as **Service Fabric**. The usage of service fabric is also exposed to applications to host their own services in a Service Fabric cluster. You can read more about Azure Service Fabric at <https://azure.microsoft.com/en-us/services/service-fabric/>.

Note As you read more about service fabric in this chapter in the book, you will likely see some similarity to another *fabric system* called Kubernetes. If you want to read more about differences between these two systems, this blog post is a good place to start: <https://techcommunity.microsoft.com/t5/azure-developer-community-blog/service-fabric-and-kubernetes-community-comparison-part-1-8211/ba-p/337421>.

To round out the set of *Azure Services*, Microsoft announced the data platform or **SQL Services**, thus beginning the first public announcement of the journey that would become Azure SQL.

The Road to SQL Azure

A big part of the announcement for Windows Azure at PDC in 2008 involved data. Since the CloudDB project in 2006, Peter Spiro, David, Campbell, Ajay Kalhan, Tomas Talius, and the rest of the team had built out a set of cloud services for SQL Server to now host *multi-tenant databases* (or multiple customers in a shared set of SQL Servers).