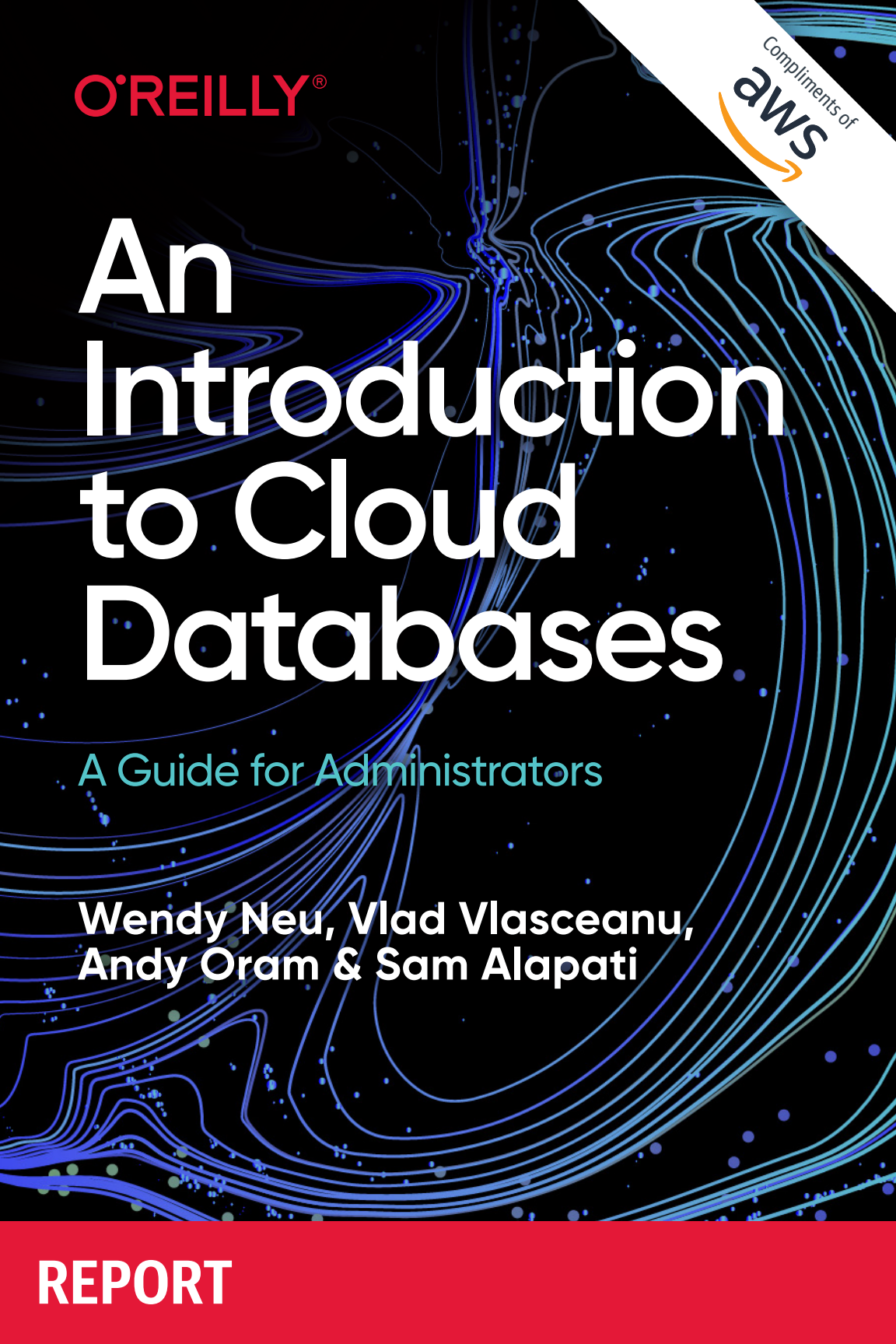
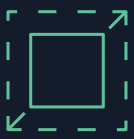

O'REILLY®
Compliments of
aws
An Introduction to Cloud Databases
A Guide for Administrators
Wendy Neu, Vlad Vlasceanu, Andy Oram & Sam Alapati
REPORT

Break free from old guard databases

AWS provides the broadest selection of purpose-built databases allowing you to **save, grow,** and **innovate faster**

Enterprise scale at 1/10th the cost of commercial databases

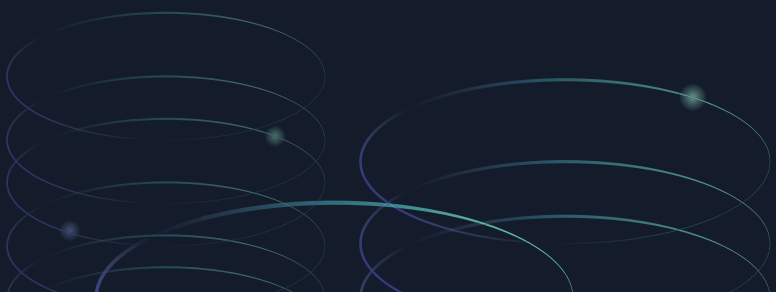


3-5x the performance vs popular alternatives

14+ database engines - more than any other provider

aws databases

Learn more: aws.amazon.com/databases



An Introduction to Cloud Databases

A Guide for Administrators

Wendy Neu, Vlad Vlasceanu, Andy Oram, and Sam Alapati

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

An Introduction to Cloud Databases
by Wendy A. Neu, Vlad Vlasceanu, Andy Oram, and Sam Alapati

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Development Editor: Jeff Bleiel
Acquisitions Editor: Jonathan Hassell
Production Editor: Katherine Tozer
Copyeditor: Octal Publishing, LLC

Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

September 2019: First Edition

Revision History for the First Edition
2019-08-19: First Release

This work is part of a collaboration between O'Reilly and AWS. See our statement of editorial independence.

978-1-492-04482-6

[LSI]

Table of Contents

CHAPTER 1
Database Options in the Cloud

The rush to the cloud is often measured in pure business terms—and its growing popularity is widely recognized. For instance, a recent Gartner report (requiring the submission of business information to view) finds that databases are growing at a rate of 68% in the cloud, whereas there is little on-premises growth outside of price increases and what they call "forced upgrades."

But this dramatic shift in computing and database access also shows up in the types of services offered and the evolution of professional jobs in computing. Managed databases of many sorts are now part of every major cloud vendor's offerings. The use of these databases will remove many tasks performed by a database administrator (DBA) in traditional environments where an organization owns its own hardware, which we'll call *on-premises* environments. Moving to the cloud will add new tasks, change some existing tasks, and provide a subtly different context for understanding many of the tasks.

Cloud vendors doggedly keep up with changes in the database space and vary their database offerings to meet the diverse needs of their clients. Relational databases (in both transactional and data warehouse form) appear along with nonrelational databases, such as key/value and document stores.

This report helps the DBA and related staff—such as data scientists, data architects, and application developers—choose among cloud offerings. It explains on a general level what responsibilities the DBA should expect to perform in the cloud environment. Finally, it offers guidelines for migration.

The report does not cover arguments for or against moving to the cloud, because there are other resources to support this decision and because the decision is tied in so tightly with the particular traits of your databases and how you use them. Furthermore, we don't review or recommend particular cloud offerings, although we refer to the offerings of the currently dominant cloud vendors: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform.

This chapter includes the following:

- A brief recap of the differences between on-premises computing and cloud computing, with an eye toward what is relevant to DBAs
- Definition of a managed database
- Types of managed databases offered by cloud providers
- Overview of a DBA's role in the cloud

Cloud offerings often mirror the software available for traditional, on-premises deployment. This means that you can move data from your own equipment to a comparable database offered by a cloud vendor. For instance, you can run MySQL or Oracle in the cloud instead of in your own datacenter, or you can use a database native to the cloud such as Amazon Aurora.

You should also be able to keep a foot in each territory (on-premises and cloud), moving data into the cloud at a pace that's comfortable for your organization. You can also maintain two deployments forever, one on-premises and the other in the cloud. The cloud is often a good place to try something totally new, such as a big data project that you've never created before.

Although we expect most readers to be familiar with the cloud, its benefits, and various reasons to use or avoid it, the next section summarizes traits of the cloud that underlie the ideas in this report. After that, we can focus on databases.

High-Level Effects of Moving to the Cloud

As you move data to the cloud, you will encounter differences from on-premises deployments that affect your decisions. Following are the key changes of interest to a DBA:

- You pay as you go, in hourly or monthly increments, to run each server (called an *instance* in the lingo of virtual computing). Thus, if you have a business intelligence (BI) application that you run every night, you can fire up an instance of a database for 8 hours during the night and avoid paying for the other 16 hours. With some services (such as AWS Reserved Instances) you can also run a database at a discount for a limited time.
- Hardware and related infrastructure are handled by the vendor. This can take an enormous amount of work off the shoulders of system and database administrators. Instead of provisioning your own hardware, you choose the amount of CPU power, memory, and storage you want in your database instance. Vendors often have special offerings that can improve performance in ways related to strategies used on-premises: for instance, Microsoft Azure's SQL Data Warehouse offers solid-state drive (SSD) storage, and Amazon DynamoDB has a special in-memory caching version.
- The major cloud vendors (AWS, Azure, and Google Cloud) offer not only the isolation of your workloads in virtual machines (VMs), but the isolation of complete corporate networks in the cloud, a service called a virtual private cloud (VPC). Just as VMs allow each of your servers to use computing resources efficiently and securely, a VPC allows you to set up your corporate network in the cloud, potentially saving money while letting the cloud vendor manage a good deal of your network security. Database servers can participate in these VPCs. You can also set up a virtual private network (VPN) to communicate between the cloud's VPC and your on-premises computers.
- Cloud vendors offer multiple locations that are important for several purposes: to offer servers that are geographically close to the user bases of clients, provide redundancy in case of disaster, allow load balancing, and more. Some countries, particularly in the European Union, require that some types of data be stored within their legal jurisdictions, for privacy or other legal reasons. The concepts of *regions*, which may cover a large part of a continent, and *availability zones* (AZs) within those regions appear in AWS, Azure, and Google Cloud. Thus, you can distribute your workloads across several AZs in a region, replicating data and services, and set up load balancing across

redundant copies of data to provide resilience. Some cloud vendors offer regions with special features to meet regulatory needs, particularly around security. All major cloud providers also offer content delivery network (CDN) services.

- Cloud vendors offer sophisticated ways to scale services, provide fault tolerance, and perform load balancing. If your database instance fails and your configuration is set up for high availability, a new instance might be available. The client might need to restart sessions that were running on the old instance when it failed, but your administrative efforts go into configuring fault tolerance instead of getting up at 2:00 A.M. to restart the database. Similarly, if the load on a database becomes high, a new instance can start up to share the load, and then shut down when it is no longer needed. This is called *elastic scaling*.
- Vendors normally offer command-line options, graphical interfaces, and APIs, each with strengths and weaknesses that are familiar to computer professionals. APIs permit automation and thus take work off your hands.
- Some security that you need to worry about on-premises is handled by the vendor. Obviously, they control physical access to the servers. Depending on how you use their services, they can also protect you against low-level internet attacks such as port scans and denial-of-service (DoS) attacks. Later chapters lay out the aspects of security that are still the DBA's responsibility.
- Cloud vendors build in monitoring and performance tools that you can hook into your databases fairly easily, and that are always sprouting new features. Third-party monitoring tools might also be able to tap into and extend the built-in monitoring capabilities of the platform.

Many criteria for choosing a provider are general, not database specific: pricing, the ecosystem of third-party developers around each offering, stability, support for legal compliance regimes, added value in high-end tools for artificial intelligence, and more.

As you evaluate cloud offerings, think about future directions for your organization. For instance, will streaming and real-time data processing be a bigger part of your future? Which cloud offerings will support you a couple of years from now?

Self-Managed Versus Managed Databases

We now look specifically at databases. Your options in the cloud divide into two major categories:

Self-managed databases
: In this cloud offering, the vendor provides just the hardware, the hypervisor to run your VM, and the API or other tools to manage deployment. You need to create a VM running an operating system and your application. Obviously, you can run anything you want in the VM, including a database engine of your choice. You need to perform most of the administrative tasks yourself, such as installing updates and configuring all of the networking options.

Managed databases
: In this cloud offering, the vendor provides not only hardware but also the server software itself. Most vendors offer both traditional databases (such as Oracle and MySQL) and cloud native databases that are specific to that vendor.

Self-managed databases are much more like running your own database on-premises; managed databases change the way you work a lot more. Here are the key differences:

- If you use a managed database, you don't need to manually download, install, update, configure, or back up the database; the cloud provider does all of that. You can still modify parameters that instruct the database how to run (collation, cursors, connections, etc.) and control decisions such as whether to install new releases and how long to retain backups.
- Sometimes, the cloud vendor can provide a license for a proprietary database and include the cost with the cost of using the cloud service. In other cases, you are still responsible for obtaining a license. If you already purchased a license from a proprietary database vendor, you can sometimes apply that to a managed database.
- Because the vendor installs and runs the managed database, you are limited to the choices the vendor makes. You can't ask for a database engine that isn't supported by the vendor, or even a version that isn't already supported. But the three big vendors

(AWS, Azure, and Google) offer several proprietary and open-source databases as well as their own cloud native databases.

- Self-managed and managed databases offer different parameters for configuration.
- With managed databases, much of the security is handled by the vendor. But even a managed database leaves important security decisions up to you, such as to whom to give accounts and what limitations to place on tables or columns. Monitoring and auditing access attempts is usually still your job. You can also control security settings such as which IP addresses have access.

Managed databases therefore offer many advantages over on-premises deployments and self-managed databases, but you might have unique reasons to run your own database. The rest of this report covers managed databases because they offer a unique opportunity for a DBA to focus on activities that have more long-term impact on the business.

Cloud Native Databases

Managed databases can also be divided into two categories: traditional and cloud native. Traditional databases such as Oracle, SQL Server, MySQL, and PostgreSQL are frequently offered as managed databases by cloud providers. If you have built your organization around one of these databases on-premises, moving to the same database in the cloud simplifies migration. You are less likely to need to alter your applications and can use familiar tools to manage the databases. You can also mix offerings from different vendors or maintain your on-premises version of the database.

But the vendors have invested great effort into new offerings of their own, sometimes called *cloud native* databases. Vendors provide evidence that the cloud native databases perform better, scale more easily, and are cheaper in the long run. Thus, testimonials from Autodesk and InfoScout suggest that AWS engineers have solved many of the scaling and efficiency problems of managing relational databases in the cloud with their own database, Amazon Aurora. Cloud native databases are also designed to scale enormously, a task that has historically been difficult for relational databases.

Types of Managed Databases

Most types of databases you find out in the field are also offered as managed databases. Additionally, cloud vendors have also developed their own cloud native databases, following common industry trends and offering performance benefits. The major types of supported databases include the following:

Relational databases
: As mentioned earlier, some cloud databases are managed versions of popular databases in widespread use. For instance, Azure offers Microsoft's traditional SQL Server. Amazon supports MariaDB, MySQL, Oracle, PostgreSQL, and SQL Server through its Amazon Relational Database Service (RDS). Such offerings help you to move databases more easily from on-premises installations.

  In addition, vendors have created databases of their own. For instance, Azure provides Azure Cosmos DB, Google Cloud offers Cloud Spanner, and AWS offers Amazon Aurora.

Data warehouses
: Although these are typically relational databases, they differ from the transaction-oriented databases internally and in their offerings. For instance, transaction-oriented databases usually store all the columns of a single row together so that you can quickly retrieve multiple columns about a customer or product. In contrast, data warehouses in the cloud tend to be columnar, meaning that they store data by column instead of by row. This greatly speeds up common warehouse queries like, "give me the ages of all customers who live in California." The tools offered by cloud vendors with data warehouses focus on fast ingestion and extraction, facilitating their use in big data applications. Amazon Redshift, Google's BigQuery, and Azure's SQL Data Warehouse are examples of these offerings.

Nonrelational databases
: This term is commonly used to cover a variety of different data stores that, unlike traditional relational databases, are built for special-application use cases. Cloud vendors offer a variety of these for different purposes:

  - *Key/value* databases offer quick storage and retrieval of values without support for more sophisticated operations.

- *Document* databases store data as JSON documents with a flexible schema, allowing data to be stored and queried in the same format used in applications.
- *Graph* databases store relationships among objects, making it easy to run algorithms such as finding the object with the most connections.
- *Search* databases optimize the location of documents containing particular words.
- *Time series* databases record events with timestamps and are optimized for time-related tasks such as graphing occurrences of events over time, useful for tracking events such as device readings or web postings.
- *Ledger* databases are like enhanced, secure log files that record activities such as financial transactions, signing them, and making them immutable.

Other options, including in-memory caching databases, might also be available. Cloud vendors offer services for less structured storage (i.e., BLOBs or files): examples include Amazon Simple Storage Service (Amazon S3), Google Cloud Storage, and Azure Blob Storage. The services offer even cheaper options for archival storage, along with tools to move data in and out of storage easily.

This chapter, necessarily, has stated database features in general terms, with lots of hedging words such as "some" or "might." This reluctance to be pinned down stems from the subtle differences that can be seen after you examine vendor offerings carefully. Some offerings have evolved further than others. Some databases offer easier administration or migration than others. You must take all of these factors into account when you choose a cloud offering.

When a Managed Database Might Not Be a Good Fit

A managed database service isn't appropriate in a couple scenarios.

Your database size and IOPS are greater than the database limits
: Cloud native databases are designed to scale to meet extreme demands. But managed databases based on more traditional database engines often come with some upper limits on their

database engines often come with some upper limits on their size and the input/output operations per second (IOPS) they can support. Many organizations can fit within these limits. But if your database is truly enormous, and you have greater IOPS needs than what the provider can support, you might find your chosen managed database services to be infeasible. You should run a cloud native database or run your own instances in the cloud by renting the VMs and storage that you need.

You need full control over your databases
: If it's important for you to have maximum control over your databases, you must manage your own databases in the cloud. If you need a specific database version that's not supported by the cloud vendor, or you need to use database features or options that aren't supported by the provider, you're back on your own.

Role of the DBA in a Managed Database

As mentioned at the start of this report, moving to the cloud will change your job, sometimes in subtle ways. Chapter 2 covers this in detail, but we'll just say here that after you relinquish hardware maintenance, system configuration, and other things a managed database takes care of, there is plenty of work left for you. Here are some tasks that the DBA often performs on a managed database:

- Determining requirements, such as CPU power and memory.
- Configuring database runs, including the automation of scaling, failure recovery, and taking snapshots of the database.
- Choosing which regions and AZs to host the database instances.
- Creating and manipulating databases and tables through the command line, console, or API offered by the vendor.
- Authorizing user accounts, setting up groups, and controlling access to databases and parts of the databases. Some databases offer this access through identity and access management (IAM) and others use the traditional interfaces offered by the database in on-premises deployments.
- Determining how to divide data into shards when you need to distribute it across multiple systems.
- Setting database parameters that affect performance or resiliency.

- Auditing access and monitoring database activity for the purposes of security, performance, and resilience.
- Performing ingestion and data transfers through traditional extract, transform, and load (ETL) or more recent streaming data tools. These data transfers can involve accepting data from outside sources, providing data to big data tools such as Spark or Kafka, storing data in a data warehouse either in the cloud or on-premises, and any other relationship you need to establish with another tool to get data processing done.
- Determining schemas or formats, together with data architects and application developers, and implementing them in relational or nonrelational data stores.

Most exciting, you'll have more time to look at the business context for the database and think of ways to improve the value of the data you manage for your organization. We go into more detail about these topics in Chapter 2.

CHAPTER 2

The Changing Role of the DBA in the Cloud

Traditionally, a database administrator (DBA) was entrusted with pretty much the entire gamut of operations pertaining to the storage and use of an organization's data. The DBA's task list included the following:

- Installing the racks and cabling
- Installing, patching, and upgrading database software
- Creating and configuring database instances
- Managing users, roles, and permissions
- Backup and recovery
- Securing the data, including the encryption of critical data
- Data migration, ingestion, and export
- Performance tuning
- Database monitoring and troubleshooting
- Handling high availability
- Assisting database developers and analysts in their database-related tasks

As you can tell from this list, the DBA's job has typically revolved around database operations, disaster management, user management, and performance tuning. Monitoring and troubleshooting have mostly been manual, and usually reactive instead of planned.

Not much time has been left for performing tasks that are higher along the value chain, such as designing databases and optimizing the applications that rely on the databases.

The lack of automation and a reactive approach to problems has left DBAs struggling to keep up with the various databases under their watch. Most of their workdays have devolved into a triage model in which they always need to attend to the most urgent tasks.

As cloud computing profoundly changes job roles and the ways organizations run their businesses, it is restructuring the traditional work of a DBA. In the cloud, many of the traditional tasks of a DBA simply vanish or are significantly reduced. The cloud provider, of course, handles all of the infrastructure work, such as racking and stacking the servers, networking, and storage. Backups and security are also farmed off, mostly to the cloud provider's domain. So what do DBAs do in the cloud?

If you're running your own databases in the cloud, you still will be doing a lot of the traditional DBA tasks such as installing and patching software, backing up the databases, and so on. This report focuses on managed databases, as defined in "Self-Managed Versus Managed Databases" on page 5, because they offer several extra benefits over self-managed databases as well as more substantial changes in the DBA's role. This chapter also focuses on relational databases, which typically require a greater variety of management tasks.

When you use a managed database in the cloud, you can spend more time on data architecture and the applications that the database supports. Anticipating needs and making enhancements to the database now become the primary concerns of the DBA. You'll be in a far better situation to derive more value from the organization's data assets, by helping teams deliver new features faster and proactively tuning application performance.

Other tasks include helping set up batch and real-time data pipelines to ingest and transform batch and streaming data. Building, maintaining, and tuning highly distributed data pipelines is an important function of administrators working with cloud-based databases. You can also devote more time to strengthening database security and assuring adherence to compliance requirements imposed by governments and industry bodies.

When you use a managed database service in the cloud, the traditional DBA role is inverted. Figure 2-1 summarizes the difference in the way DBAs spend their time on-premises and with a managed database service.

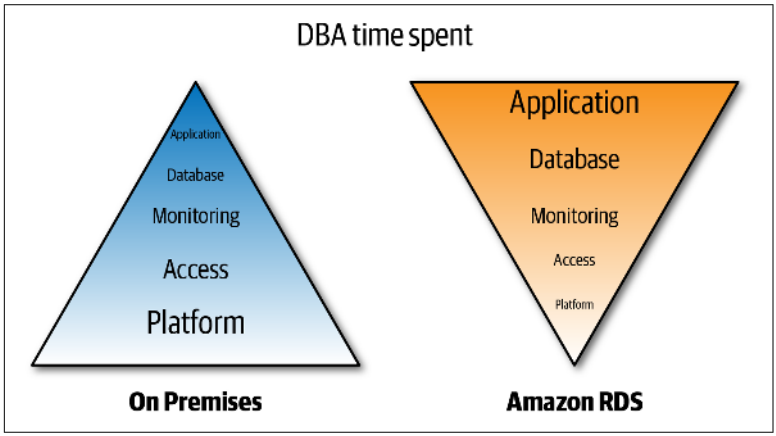


Figure 2-1. The inverted role of the DBA in the cloud

In short, moving databases to the cloud frees up administrator and developer time that you can use for other, more important tasks. Because routine maintenance and backups and such are taken care of by the cloud provider, you can focus more on your business goals such as the streamlining of key business processes.

The most critical factor determining your success in the new cloud milieu is to familiarize yourself with all the relevant cloud provider services. These include the various storage options, monitoring tools (which you can often use without extra cost), cloud security features, and other built-in cloud services that you can exploit to run your cloud databases in a cost-effective, high-performing, secure fashion.

How DBA Tasks Change in the Cloud

When databases move over to the cloud, many traditional operational tasks vanish, some tasks remain in a modified fashion, and some new tasks appear on the DBA's plate. However, virtually everything you have done on-premises will change somewhat because new tools and options will become available.

DBA Tasks That Are Taken Over by the Vendor

A key difference when using a managed database service in the cloud, as compared to an on-premises database, is that DBAs perform administrative tasks without physical access to servers or direct control over installations. At first, it might take some DBAs time to get used to this new reality. It's important for DBAs to take this opportunity to expand into the new tasks discussed in this chapter.

DBA Tasks That Remain but with Changes

Several conventional DBA responsibilities remain necessary in the cloud, but with subtle differences, regardless of whether you use a managed database service or run your own databases.

Provisioning

Managed database services such as Amazon RDS offer a wide range of instance types for your database servers. Instead of ordering a hardware system with particular memory and CPU requirements for your datacenter, you choose one or more instance types that offer the memory and CPU power that meet your requirements. Matching the instance type to your workloads can help reduce the possibility of paying for resources that you won't use.

Scaling is another provisioning task. Autoscaling, which you can do at the level of VMs and at the level of databases, lets the system add an instance when the loads become high and shut down an instance so as not to waste excess capacity when the loads diminish. Although autoscaling is an extremely valuable form of automation, you can fall victim to a DoS attack or configuration error that spins up a huge number of instances and jacks up your costs.

You can also choose regions near your database users for optimal response time. Using the offerings of a global cloud provider wisely is a key aspect of provisioning in the cloud.

As we have discussed, traditional databases, especially the online transaction processing (OLTP) databases, are notoriously difficult to scale. As the number of users, the size of the datasets, and the complexity of database operations increase, query response times grow longer. Business growth can be reflected in all those factors, as well as the growth of applications that place more of a burden on the

database. A skilled SQL developer and DBA can ameliorate these issues to a certain extent by setting optimal database configuration parameters and optimizing the SQL code.

Code optimizations and optimal database settings go only so far, however. Thus, scaling the database remains an intractable problem in many cases because you can't easily change the number of servers, RAM, or CPUs on the fly to match changing workload requirements. All of these resources involve capital budget spending, which needs to be approved well ahead of time. Adding a single physical server takes weeks in many traditional datacenters.

ETL, data ingestion, and data export

One of the most important tasks for a DBA is moving data around in some form, often transforming it in some way to extract important fields or massage the data into another format. Although this field has been characterized for some 40 years by ETL tools, recent forms of stream processing such as Hadoop and Spark have led to many new tools for ingesting data. Message brokers such as Apache Kafka and Amazon Kinesis direct the streams of data to multiple consumers in a *publish-and-subscribe* model. Most sites use different generations of these tools for different purposes, and your role is to master each tool as well as teaching them to coexist peacefully.

Streaming data imposes new burdens on traditional databases, and thus forces DBAs to tune parameters differently. Streaming usually introduces more processes that make many frequent, small updates instead of a few larger ones. Concurrency controls are also strained.

General tasks are the same whether your data is going into an on-premises database or cloud storage. But cloud vendors offer their own versions of streaming data, as well as tools for ingesting and exporting data. For instance, if you use Microsoft's cloud-based data warehouse, SQL Data Warehouse, you can then make use of both existing software in your toolbox or the tools offered in Microsoft Azure Data Factory (ADF). Amazon offers numerous tools for ingestion, such as Kinesis Firehose and Snowball. The Google toolkit for ingestion combines options from many sources.

Backups and snapshots

Cloud databases offer two basic methods for backing up and restoring your databases. Automated backups are usually turned on by default and can be used for point-in-time database recovery.

In contrast, snapshots are backups that you initiate. In on-premises environments, DBAs routinely take snapshots—by dumping the data from the database while it is in a consistent state—to do backups or replicate the data store. In the cloud, you can still use a snapshot for long-term backups (such as to tape storage), replication to other regions in the cloud, and retention for compliance to regulations.

Managed database services in the cloud, such as Amazon RDS, make snapshotting easy. You can make storage volume snapshots of a database instance (which can consist of multiple databases) and create a database instance by restoring the database snapshot.

Access

In the cloud, you are responsible for everything unique to your data and users: handing out accounts, forming groups or roles to combine user accounts, and designating access to various parts of your databases. We cover this topic more in "Security for Data and Applications in the Cloud" on page 19.

Maintenance of development and testing environments

One common DBA task is setting up development and testing environments. Often, this involves a lot of work on the part of the DBA. If you take database snapshots on a regular basis, you can create a development or test environment that reflects the state of the database at a given point in time. You can find examples of such services for Amazon, Google Cloud, and Azure.

Logs

DBAs need database logs for troubleshooting, compliance, and auditability; this is still true in the cloud. Figure 2-2 shows how you can easily view logs for a database instance through the AWS console. When you have many databases that you must monitor and manage, the usefulness of this feature becomes clear.

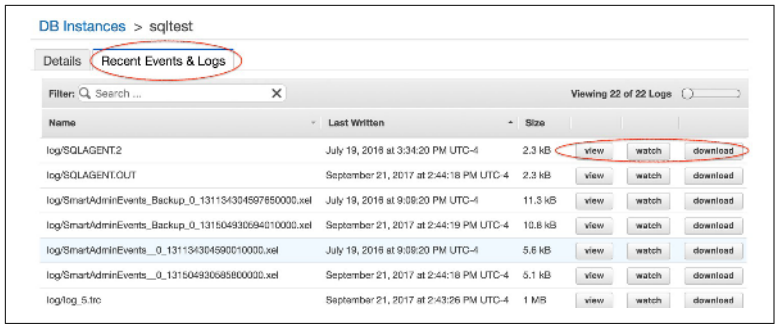


Figure 2-2. Viewing database logs through the Amazon RDS console

Monitoring and notifications

Cloud vendors provide copious information on metrics such as CPU utilization, query latency, query throughput, and I/O and memory usage. A number of built-in monitoring capabilities are offered, some free and some for a fee. You can also set alerts to quickly let you know if such measures move outside of predicted bounds.

Many cloud platforms expose events such as when a failover occurs, when a backup is taken, and when a database is stopped or reconfigured. The platforms provide hooks to let you set notifications or take other automatic actions when events occur.

Performance optimization

Tuning database performance is one of the key tasks of a traditional DBA; it separates the good DBAs from run-of-the-mill administrators. Good performance tuning requires an in-depth understanding of storage, network, memory, and application code. Placing the databases in the cloud doesn't relieve the DBA from complete responsibility for the database. Performance tuning, both reactive and proactive, is still going to be primarily the DBA's responsibility. However, cloud vendors provide some easy ways to obtain information, such as AWS's RDS Performance Insights.

In the cloud, you won't be spending a lot of time worrying about typical server- and storage-related performance issues, because you can easily scale up or scale out your databases based on your workloads. You're more likely to spend your time analyzing database waits, so as to improve database response times.

In traditional on-premises environments, some performance tools are provided by the database itself, such as the EXPLAIN command and MySQL slow query log. Other performance monitoring tools run outside the database, and you can set up a system in the cloud to run these. But cloud vendors have been assiduously developing performance tools that run natively in the cloud, usually for no extra charge. You should definitely make use of what the vendor offers.

Databases that are performing poorly on-premises won't magically start performing well just because you moved them to the cloud! But the move can be helpful in some cases, such as when your on-premises database environment didn't have the capability to support a demand from the application for a large number of IOPS.

Handling high availability

This task is critical nowadays, when people will abandon a website if it fails to load within two or three seconds. On-premises, this task requires the advance purchase of adequate processing power, plus the configuration of load balancers. In the cloud, you can choose autoscaling and configure your database to run in multiple zones. A multizone database can automatically fail over to another AZ if the first zone fails. Similarly, you can offer your data in multiple regions for both high availability and local storage.

New Tasks for the DBA in the Cloud

When you run your databases in a cloud environment, you encounter tasks that, in most cases, you never had to worry about in a datacenter. These tasks include a new orientation toward costs and tracking your database licenses in the cloud.

Planning operational costs

The fundamental calculations for total cost of ownership (TCO) and return on investment (ROI) change radically in the cloud. As mentioned in "Provisioning" on page 14, you usually pay for particular instances and particular amounts of time in the cloud, instead of buying fixed infrastructure up front. However, you need to factor in many other things that are not directly related to databases. For instance, locating the database in regions closer to the users can reduce networking costs and perhaps even generate new business.

Vendors offer numerous options to match your usage to your needs and significantly reduce your spending. For instance, by purchasing reserved instances in AWS, you might be able to dramatically reduce costs. In general, careful planning can let you predict your needs at any particular time and eliminate the overprovisioning that on-premises datacenters need to do.

Effects on architecture and application design

The architecture of cloud-based databases is inherently quite different from that of a traditional, on-premises database in a datacenter. You must take into account multiple geographic regions and AZs, different types of network designs, and other cloud-specific features when planning your database deployments. The DBA must keep in close contact with application developers to understand their needs and provide the best fit in the cloud at the lowest possible cost.

Tracking license compliance

Proprietary databases are offered in the cloud through two basic licensing models: license included or bring your own license (BYOL). The latter model is valuable if you already have a license and the model allows BYOL. You might need to consult a licensing specialist or the vendor to find out whether your existing license applies to the cloud.

Security for Data and Applications in the Cloud

Introductions to cloud computing typically arrange the functions of an administrator on a stack, with the installation of physical servers and cabling at the bottom of the stack and application management at the top. Managed databases assign responsibility for the bulk of these tasks to the cloud provider, leaving only application-level tasks in the hands of you the client. Self-managed databases keep most tasks in the hands of the client, leaving the vendor mostly to provide physical servers and basic networking.

Security tasks also fit this stack concept, leading to what is commonly called a *shared security model*. The vendor takes on more security tasks in managed databases than in self-managed databases. Briefly put, the cloud provider oversees infrastructure security, but you continue to be responsible for the security of your data and user

information in the cloud. Thus, the tasks performed by the cloud vendor include the following:

- Physical securing of the datacenter and equipment, including the overseeing of staff
- Ensuring redundancy through replication and backups, following guidelines from the client
- Patching and updating the software, including the database itself if you choose a managed database
- Running tools that monitor and audit access
- Providing network and application firewalls
- Protection against DoS attacks
- Identity management services, which you configure

Client security tasks include:

- Setting up users and roles through the vendor's identity and access management (IAM) system
- Assigning access rights to users in the database, which includes defining accounts and roles, and keeping the accounts up to date by removing users after their access is no longer needed.
- Specifying firewall rules, backups, and other parameters offered by the cloud vendor
- Setting up alerts and reviewing logs for unauthorized access
- Operating system security if you use your own VM instead of a managed database
- Application-level security such as preventing SQL injection attacks
- Data encryption, at rest and in transit
- Proper authentication and authorization of users trying to access the databases or applications
- Log gathering and inspection
- Setting up event alerts and monitoring events for anomalies

Familiarity with the entire gamut of cloud security features is critical for running databases in the cloud. In the following subsections, we cover the key cloud security features on which DBAs should focus

their study: access control and IAM, network isolation, and data encryption.

Access Control and Identity and Access Management

Cloud providers generally excel in strong access control mechanisms. The vendors mentioned in this report rely on centralized IAM to manage users, security credentials (passwords, access keys, and permissions), and authorization policies that control which resources and services users can access. Administrators need to master IAM just to get access to the cloud for themselves and their users. In addition, cloud native databases are sometimes integrated with the general cloud IAM tools. Linking your database protections to the cloud's IAM provides the easiest and most secure access policies.

You will use IAM to define user accounts and then add database-specific access rules on top. Using IAM, you can grant different user permissions to perform different database operations. You also can institute fine-grained access controls—which most databases offer, such as restrictions on particular rows or columns—through IAM.

Traditional databases use the same tools in the cloud that you use on-premises, such as GRANT statements in SQL. But you might be able to hook them into IAM so that you can use the same user accounts in both the cloud and the database, and benefit from the extra security and convenience provided by that integration.

Network Isolation

Some common cloud features to protect your systems on the network are VPCs, firewalls, and network access control lists (ACLs).

As we discuss in "High-Level Effects of Moving to the Cloud" on page 2, a VPC is a private network within the cloud for communication between your servers. Within a VPC, you can isolate database instances by specifying the IP range that is allowed access to each database. The organization that creates a VPC has full control over its virtual networking environment and can select its own IP address ranges, create subnets, and configure its own route tables and network gateways.

You can, in addition, set up a virtual private gateway that extends your corporate network into your VPC, and permits access to the database instances in that VPC through a VPN of your choice.

Most cloud providers offer built-in firewalls that help you control network access to the computing instances. The cloud provider might also offer a private or dedicated connectivity option to connect the cloud consumer's office and on-premises environments with the cloud environment. You can set up database security groups to secure database instances within a VPC. Security groups are firewall rules that control network access to your cloud databases. You can also allow or deny network traffic entering and exiting a subnet via network ACLs. Any network traffic that enters or exits your VPC via your VPN can be inspected by your on-premises security infrastructure (such as a firewall) and by intrusion detection systems.

NOTE

Direct Connections

Instead of a VPN, you can connect systems outside the cloud to systems within the cloud through Direct Connect in AWS or ExpressRoute in Azure. They exploit private network links provided by telecommunications carriers to create a direct connection. End-to-end encryption is still recommended, and is generally done through standard Secure Sockets Layer (SSL)/Transport Layer Security (TLS).

Data Encryption

In the cloud, it's easy to encrypt your data to provide additional protection to data at rest. For example, when you enable encryption in an Amazon RDS cluster, the database stores all data in the tables that you create in an encrypted format. The encryption also applies to the database backups. Encryption is also easy to set up when you transmit data to and from the cloud.

Encryption is particularly necessary for organizations that must meet industry compliance requirements such as Health Insurance Portability and Accountability Act (HIPAA) compliance for health care, the Sarbanes-Oxley Act (SOX) for financial reporting, and the Payment Card Industry Data Security Standard (PCI DSS) standards for ecommerce and retail business. If you protect your encryption

keys, the loss of the encrypted data itself isn't normally considered a reportable security event.

Cloud providers typically offer managed services to simplify the creation, control, and management of your encryption keys. Thus, AWS Key Management Service (KMS) provides a centralized view of all the key users in the organization. It integrates with AWS CloudTrail to provide a log that shows key usage across the organization, thus satisfying several key regulatory and compliance requirements.

Infrastructure as Code: Making the Most of the Cloud

When you move your databases to the cloud, there's a temptation to keep management changes to a minimum because you're used to doing things a certain way. However, this strategy means that you're simply pouring old wine into new bottles and are failing to fully capitalize on the immense benefits the cloud places at your doorstep.

A key technological advance that distinguishes cloud-based systems to those in local datacenters is the easy availability of configuration systems that treat your infrastructure as code. Treating infrastructure as code enables many DevOps practices, which in turn facilitate close collaboration between developers and operations so that they can automate application delivery at scale. On-premises, you certainly can install automation tools such as configuration management systems (Puppet and Chef being popular examples) and continuous integration (CI) tools (Jenkins, for instance). But very sophisticated tools of this sort are built in to the cloud.

For instance, templates provided with AWS CloudFormation help you to model your entire infrastructure as code. You can define your cloud infrastructure by creating and configuring resources such as database tables and storage (Amazon S3) buckets, and treat these resources as code. You can check the AWS CloudFormation templates into your source control system and manage them the same way developers manage their code files.

A tool like CloudFormation offers the following benefits:

- It helps establish a single source of truth for all of your cloud resources.

- It can be integrated with code management tools such as a source control system.
- It helps automate development and test deployments.
- It supports your disaster recovery plans.

By using template files to create your resources programmatically, you achieve repeatability and consistency across your environments.

CHAPTER 3
Moving Your Databases to the Cloud

Like most organizational transformations, moving to the cloud is not performed overnight. You should choose one project to try out in your chosen cloud provider: either an existing project that provides a useful test case, or a new project that is unencumbered by legacy practices.

Almost all organizations that move to the cloud do so by first doing a proof-of-concept type migration of a noncritical database. After you've picked the low-hanging fruit, or successfully moved a small project that reaps benefits from using the cloud, you can extend what you've learned to other databases throughout your organization.

This chapter helps you understand the criteria that are usually associated with a successful first migration. Much of our material applies to relational databases. Although each migration is unique, you will probably carry out most or all of the following steps:

Planning

- Requirements gathering
- Determining capabilities to address the requirements
- Assessment of which databases to move and what changes might be required to the database or the applications using it
- Establishing success criteria and rollback criteria (fail-safes)

Data movement

- Replication
- Incorporation of changes since the replica was created
- Application testing
- Cutover
- Post-migration checks

Optimization

- Performance tuning
- Designing high availability
- Determining what events to log and monitor
- Creating a disaster recovery plan

We look at each major stage in the following sections.

Planning

This phase helps an organization assess the following issues:

- Relevant elements of the current environment: applications, databases, and critical workloads
- Whether the application or workload will run properly in the cloud provider's environment
- How the migration will help meet business objectives
- Requirements for the end state of the system
- The cost of running the current computing environment in the cloud, which should be a comprehensive return on investment, as explained in "Planning operational costs" on page 18

Factors in a Migration

When you want to move a database that you are running on-premises to a managed service in the cloud, you need to deal with physical, software, and organizational issues. It requires attention to several levels of change:

Physical data movement
: You need to get the data itself into the cloud. Data transfer for large databases can be both time-consuming and costly, but there are migration services to help.

Infrastructure compatibility
: Check every aspect of the cloud service against your legacy on-premises systems. A case study on the AWS blog offers an interesting example of snags that might occur during a test migration by unexpected software incompatibilities and minor errors. This particular migration was held up for some time just because of differences in SSL implementations by the on-premises systems and the cloud provider.

Database compatibility
: The cloud database is likely to differ from your on-premises database in some features that affect migration. Subtleties such as database character sets and permissions on stored procedures can trip you up. A *homogeneous* migration (such as from one Oracle database to another Oracle database) will probably go more smoothly than a *heterogeneous* migration (such as from an Oracle database to a PostgreSQL database, or one of the proprietary cloud vendors' databases). "Checking for Incompatibilities" on page 31 explains what you can do to reduce friction between databases. For certain kinds of migrations, cloud vendors might provide tools that take some of the work off your shoulders.

  If you're currently running a very old version of a database on-premises, it might be time to upgrade. Instead of merely choosing a newer version of the same relational database management system (RDBMS), which will involve a certain amount of pain, you might use this opportunity to try a heterogeneous migration to a new RDBMS that shows promise for the future.

Organizational changes
: People from several teams, especially DBAs and application developers, will need to devote time to learning the cloud tools and managing the new instances while keeping the legacy systems going on-premises until you are ready to move everything to the cloud. You might choose to spend months or even years

doing a full migration. You might also keep some of the data (or a copy of the data) permanently on-premises.

Major Migration Tasks

We suggest the following steps for a successful migration.

Create a cloud migration plan
: The cloud migration plan should list all the databases and applications in the order in which you want to move them to the cloud. The final assessment plan should delineate the migration plan for all databases. If you need new resources—in software, finance, or personnel—in the new database environment, that should also be in the plan.

Determine who performs the migration
: DBAs and developers should work together on the migration effort because each team has something to offer. While choosing staff, you can also decide whether to employ a migration service provided by the cloud vendor or a third-party company.

Educational tasks
: These include broad brush tasks such as understanding the use of the cloud vendor's tools, the specific features of the target database engine, the scope of the database migration, and the architecture of the cloud databases.

Create the cloud database architecture
: Select the type of databases that you want to use, as well as whether to use a managed or self-managed database. When making this choice, weigh all relevant factors such as cost, performance, reliability, and scalability.

Choose a migration process
: "Migrating the Database" on page 32 lays out the criteria for choosing among replication, backup/restore, or a dedicated migration service.

For a self-managed database, create your computing infrastructure
: If you choose to manage your own databases in the cloud, create virtual instances and deploy them through the vendor's compute service, such as Amazon Elastic Compute Cloud (Amazon EC2).

Determine opportunities for architectural changes
: You might find that choices you made on-premises are no longer right for the new environment in the cloud. For instance, you might be able to consolidate shards.

Set up the cloud database accounts
: Set up general accounts for users in the cloud through IAM before you set up the users, roles, and groups in the database.

Upgrade and test applications
: You might need to rewrite code to work with the new database. Some of the migration of applications may be automatable, just as with the database itself.

Readiness Assessment

A *readiness assessment* helps you to estimate the costs, the architecture of the cloud database, migration plans, and the impact of the move to the cloud on compliance regulations. The result of a cloud readiness assessment is a detailed report of your company's readiness to move its databases to the cloud.

Following are brief descriptions of the key steps during a cloud readiness assessment.

Stakeholder interviews
: Talking to application developers, business users, and others concerned with the use of data within your organization team will help you to determine requirements pertaining to performance, high availability, and the features and capabilities of the cloud-based databases.

Analysis of current on-premises databases
: Go back over your current databases to determine growth patterns of data, backup and recovery strategies, ongoing data exports and imports, and so on. Understanding the current database usage patterns will help you when it's time to decide what databases to use in the cloud, and whether you'll need a particular database offering in order to get the capabilities you need.

Prioritization of the databases to move
: Pick the databases that you'd like to move to the cloud first. One important criterion is the extent of changes to pass:[<span

class="keep-together">applications that will be required, which in turn calls for coordination with developers.

Migration cost analysis
: Perform a thorough analysis of the costs of migrating to the cloud versus staying in the on-premises datacenters so that you fully understand the TCO and ROI of the cloud migration.

  Lowering costs by using a cloud database service might be one of your key goals. But a poor choice of cloud services will defeat this purpose.

Security and compliance
: Sometimes, you need special regions or AZs to comply with standards such as Service Organization Control 2 (SOC 2), PCI DSS, or the US HIPAA. Fortunately, cloud providers often match particular regions and AZs to these legal requirements. In addition, there are specialized regions, such as AWS's GovCloud (US), which is an isolated AWS region subject to FedRAMP High and Moderate baselines. Finally, you need to check whether you can meet your organization's service-level agreements (SLAs) in your chosen cloud setting. These SLAs typically include metrics for planned maintenance, backups, recovery point objectives (RPOs), and recovery time objectives (RTOs).

If you've done your cloud assessment correctly, you should also have a good idea of the scope of your cloud migration. Some of your applications might be so old that they would need to be completely redesigned to move to the cloud. If you don't currently have the resources to rewrite the applications, you can keep them in on-premises databases or move the entire servers into the cloud.

Migration steps
: You should plan exactly when and how you will do the data migration, along with preparations for rolling back the change if later testing shows that the database is not working properly. Establish tests and the success criteria for each step.

Application changes
: You need to do this in coordination with the development team. This includes scheduling of the application changes into the development process and testing.

Automating the migration
: This is valuable because it lets you repeat the migration as often as you need to, fixing errors as you encounter them or changing parameters. You also can adapt and reuse the automation framework for further migrations.

Checking for Incompatibilities

If you are using the same RDBMS (such as MySQL, for instance) for both databases, check the versions to determine whether you are using features that are unsupported on the cloud version. These checks are obviously even more likely to turn up incompatibilities if you are moving to a new database engine.

The conversion issues that you encounter when performing a heterogeneous migration to a cloud database are the same as when you're performing such a migration on-premises; however, you have access to a wider range of solutions.

The database vendors themselves generally offer tools that help move data to a cloud database of the same type. For example, Oracle comes with data migration tools such as Datapump, RMAN, and SQL Developer that help you to move your Oracle databases to the cloud. In addition, cloud providers and third parties offer services such as AWS's Schema Conversion Tool to move schemas. These tell you whether the schemas can be translated, and provide suggestions for workarounds if necessary.

Still, most heterogeneous database migrations require both automated tools and manual action by the DBA. If you're moving to an AWS database, you can draw on specific recommendations on the AWS website.

Some migration tools employ a translator to convert one database's objects, such as Oracle stored procedures, to a non-Oracle database such as PostgreSQL. These tools, however, might do an incomplete job, because each database engine employs unique coding practices.

This is the reason for performing stringent tests on the target database for accuracy and performance after migrating to the cloud. Some programming techniques employed by a specific RDBMS might not have an exact equivalent in the target database. The migration tool might flag these types of conversion issues during the migration process. In many cases, it can also show you the code that

you might need to run to overcome these issues and successfully complete the database migration.

Data Movement

Cloud providers are highly alert to the requirements their potential clients have about moving their on-premises databases to the cloud with minimal cost and disruption. Each vendor offers tools to expedite the move. AWS, for instance, offers numerous papers laying out in detail how to migrate from various on-premises databases to AWS, including procedures for Oracle, MySQL, and PostgreSQL.

Even so, migrations to the cloud can involve disruptive and costly downtimes. A migration tool, whether provided by the cloud vendor or by a third-party provider, must be able to handle all aspects of the database, such as schemas, user permissions, triggers, and stored procedures.

We recommend that you keep a journal during your early migrations, because the lessons you learn along the way will inform you and your colleagues as you pursue further migrations. Should enough bad things happen that you decide to use another vendor—or not to migrate at all—the journal will provide important evidence to support that decision.

Migrating the Database

Many cloud providers and third parties offer migration services, such as Microsoft Azure Database Migration Service and AWS Database Migration Service, that are well worth consideration. These are advertised as being fast, smooth, and easy. However, you might prefer to use existing processes with which you are familiar, such as restoring from a backup to a new database environment, and using replication.

If you use a database migration service, you can keep your source databases fully operational during the migration to the cloud and minimize the downtime for applications that rely on the database. This facilitates the key goal of minimal downtime, cited by many organizations migrating to the cloud. Figure 3-1 illustrates how AWS Database Migration Service creates the tables, loads data, and keeps the tables synchronized with the source database tables.

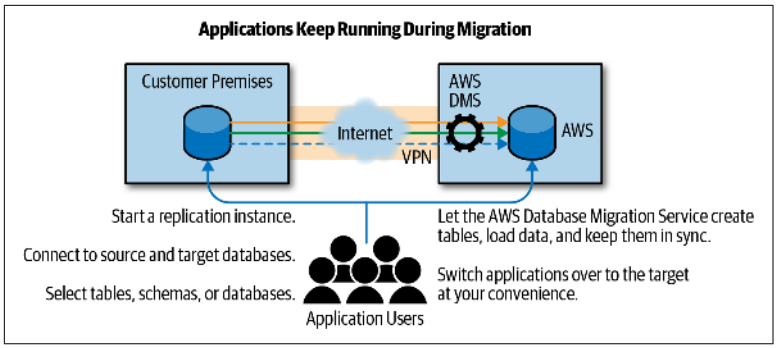


Figure 3-1. Keeping applications up during the database migration

The migration process usually revolves around a typical backup/restore process. The tasks depend on whether you can afford to take the database down during the transfer. If you must leave the database running, here's the basic sequence to follow:

1. Do a hot backup of the original database while it is running.
2. Copy and restore the backup in the cloud.
3. Enable replication to incorporate into the cloud version any changes made to the original database after you created the backup.

Of course, backup/restore is simpler if you can take the database out of operation. Migration then follows this pattern:

1. Shut down the original database.
2. Do a backup.
3. Copy and restore the backup in the cloud.
4. Start the new version in the cloud.

But you are likely to spend time testing the new cloud version before you are ready to put it into production to make sure that applications work, that security is in place, and that you are getting the performance you need. So you still might want to restart the original version before you are ready to trust the new version, and then you will use replication to synchronize the new one with the original one.

Migrating Applications

There are two basic strategies for moving your databases and related applications:

Lift and shift
: Move an entire database as is, with all of the applications it supports, to the new database in the cloud. This is the "old wine in a new bottle" application migration strategy given that you make little or no use of cloud native features.

Redesign the databases
: Rethink how your on-premises databases deal with administrative tasks such as scaling and high availability, and investigate the cloud provider's offerings in those areas. They are getting more and more sophisticated. More work and effort is required for this strategy, of course. You might also need to recode the applications to adjust to the changes in database strategy.

Often, customers start with "lift and shift" and follow up with redesigning and other modernizations.

Post-Migration Checks

After the data is completely copied over to the cloud database, validate the target database to ensure that all of the database objects are present on the cloud database. Several tests determine whether the migration was successful:

Validate data
: This can be as simple as checking the number of rows, or running a checksum to indicate that there was no loss or corruption. More complex validation processes factor in the schema changes inherent in heterogenous migrations. Unfortunately, this basic validation is often forgotten.

Basic functionality
: Perform end-to-end testing to ensure that you performed the migration successfully and that the system is operating as it is supposed to. Test the use of features that tend to differ between database engines and versions, such as triggers and stored procedures.

Performance testing
: Stress-test the new systems and benchmark their performance to get an idea of where you can enhance response time and throughput.

Security assessment
: Ensure that the cloud systems are secure by performing vulnerability assessments and penetration tests.

These steps also form a transition to the next phase, optimization. This phase follows the successful migration of your on-premises databases and applications to the cloud. The crucial elements in this phase are performance management and cost optimization. Careful, continuous monitoring of the new systems is critical so that you can quickly revert to the old systems in case you run into unexpected glitches.

Optimization

At this point, you are presumably running smoothly and can congratulate your team all around for a job well done. This section summarizes the main tasks required of the DBA and other team members as you move forward.

Availability

This, of course, is the fundamental requirement that you must meet in any setting. Moving to the cloud, fortunately, removes some causes of failure (if you properly configure the restarting of failed services). Furthermore, the cloud vendor provides numerous tools for predicting failures and alerting you to failures. To take advantage of these, carry out the planning in this section (many of these activities are also useful for performance optimization):

Disaster recovery (DR) plan
: With a large and robust cloud vendor, you can fail over to a new AZ if the current one fails. But you need both a plan and an automated process for failover. The plan should address the RTO, RPO, and geographic redundancy. As much as possible, exploit the failover capabilities provided by the vendor instead of handcrafting your own.

Logs and system monitoring
: Determine the events that can indicate imminent problems as well as problems that have already occurred. These can be incorporated into automated tracking. The tracking should convey enough information to tell you the source of failure: for instance, whether they stem from a user action such as a reboot of a service, from an attack, or from other changes in the environment. Some failures can be considered normal and can be addressed by your automated tools; these should be recorded but do not need to issue alerts to the administrator.

Change monitoring
: Administrators should always know what changes to database configuration, instance sizing, or cluster topology can affect availability. Modern development environments use robust processes for change tracking and version control so that every change goes through a vetting process and can be reversed.

System testing
: Try to determine the weak points in your system and anticipate failure. Some teams go through "pre-mortem" exercises to identify and remove potential sources of failure. Large sites can afford to bring down systems deliberately and watch whether recovery is adequate; this kind of testing, called *chaos engineering*, was popularized through Netflix's Chaos Monkey. Just as you do regular restores to make sure backups are working, you should test your recovery procedures.

Performance Optimization

Performance can benefit from the practices in the preceding section, notably monitoring. Performance monitoring should allow you to determine the relationships between events and changes in the database metrics, as well as to see discrepancies between the predicted and actual performance trends. Performance can additionally be maintained and improved through additional processes.

Workload testing
: Growth in the size and complexity of data, as well as application behavior changes, affect performance. Test performance regularly so that you learn of degradation before your customers tell you about it; you can then scale or make other changes to adapt. It can take a while for the database cache and table statistics to

update after a major change, which means that query plans and overall performance might take a while to return to desired levels.

Distributed optimization
: In fast-changing application environments, the developers understand the database queries better than the DBA. If they take on query testing and optimization, they can achieve good performance faster than the DBA can, freeing the DBA to do long-term optimizations and planning.

Knowing the importance of performance, cloud vendors have inserted traces and exposed a large number of statistics to users. AWS's RDS Performance Insights, for instance, lets you pull up historical data on a dashboard and view charts showing the load placed by particular operations, users, or SQL statements on the database. You can consult the dashboards to answer simple questions such as "When are my hosts overloaded?" or more complex ones such as "What is making this query produce suboptimal performance?" Azure offers both performance monitoring and automatic tuning. Third-party vendors also produce tools for monitoring and performance in the cloud.

Adapting to Differences in the Cloud

Finally, the vendor themselves will offer new services, new VM options, new hardware (such as SSDs), and other changes that can offer you big benefits. Always be aware of what the vendor can do for you, so that you can offload work onto its well-tested and standardized processes. As you do so, update your documented operational procedures. Finally, you can report abnormal events to the vendor's customer support, indicating the urgency of the event.

Conclusion

A migration to the cloud is a long-term process. Start small, because you will find you have a lot to learn along the way. Keep a journal and honestly record all the mistakes and problems encountered. Don't be embarrassed if legacy systems harbor poor practices or outright bugs that turn up during the migration—that will happen in virtually every organization. Documenting the problems is the best thing you can do for your company.

Hopefully, one or more of your early migrations will go well and you will be ready for a major move to the cloud. Doing so can reap benefits in costs, flexibility, and security. Not least in importance, moving to the cloud will provide an up-to-date computing environment that helps to draw leading staff, who want to be on the cutting edge, to your company.

CHAPTER 4
Conclusion

For many years after Amazon.com opened the first major cloud offering, the trade press presented the question facing system administrators and DBAs as "Cloud or not cloud?" Soon, on-premises clouds and hybrid offerings joined pure cloud solutions as options to consider. But the choices were always more complicated. As this report has shown, offerings have multiplied rapidly. DBAs must simultaneously evaluate databases along all the following axes:

- Third-party vendor, on-premises, or hybrid
- Relational or one of the many nonrelational varieties
- Managed or self-managed
- Cloud native (e.g., Amazon Aurora) or cross-platform (e.g., MySQL)
- Whether to take advantage of performance enhancements such as solid-state drives or caches
- Physical locations of cloud regions and availability zones
- Ease of migration
- Relevant skills needed and possessed by your staff
- Other aspects of vendor support and reputation

It is not a good idea to prematurely tie yourself to a choice in one area before looking at all options. It can well be that you can save a lot of money and improve customer experience by taking on some extra training or making a leap into an unfamiliar technology.

In addition to laying out the basic criteria for choosing databases, this report has tried to help you prepare a move to the cloud by preparing you for the changes that will likely occur in your responsibilities and tasks. Some responsibilities and tasks are simplified or removed by moving to the cloud, but you will also have new technologies to learn and will need to begin thinking in new ways about goals such as high availability and optimization.

You will learn a lot during your first migration, or by starting a new project in the cloud. Each project will clarify the landscape of cloud databases for you, and give you ideas for your next project. And hopefully, this report will alert you to what you need to look out for along the way.

About the Authors

Wendy A. Neu is a principal consultant with AWS Professional Services working on customers' most difficult problems, building high-quality, scalable, and architecturally sound systems. She is a regular contributor to the AWS Database Blog and is certified in AWS, Oracle, and Microsoft SQL Server. Prior to joining Amazon, she worked as a consultant in Cincinnati, OH, helping customers translate business needs into workable technology solutions.

Vlad Vlasceanu is a principal DB specialist solutions architect at AWS, based out of the Santa Monica, California office. Vlad helps customers adopt cloud native database solutions, like Amazon Aurora, and deploy large-scale, high-performance database architectures on AWS. His focus is on designing and implementing sustainable, cost-effective, and scalable database workloads that take advantage of the latest best practices and capabilities that the AWS platform offers. Prior to joining AWS, Vlad's career included more than 15 years of designing and developing both consumer focused web-based applications as well as data-driven applications for the energy industry. Vlad holds a Master of Science in Information Systems from Baylor University.

Andy Oram brought to publication O'Reilly's Linux series, the groundbreaking book *Peer-to-Peer*, and the best seller *Beautiful Code*. Andy has also authored many reports on technical topics such as data lakes, web performance, and open source software. His articles have appeared in *The Economist*, *Communications of the ACM*, *Copyright World*, the *Journal of Information Technology and Politics*, *Vanguardia Dossier*, and *Internet Law and Business*. Conferences where he has presented talks include O'Reilly's Open Source Convention, FISL (Brazil), FOSDEM, DebConf, and LibrePlanet. Andy participates in the Association for Computing Machinery's policy organization, USTPC. He also writes for various websites about health IT and about issues in computing and policy.

Sam R. Alapati is a data administrator at Solera Holdings in Westlake, Texas. He is part of the Big Data and Hadoop team. Sam is an Oracle ACE, a recognition conferred by Oracle Technology Network. He is the author of *Modern Linux Administration* (O'Reilly, 2018) as well as more than 20 database and system administration

books. Sam has experience working with all three major cloud providers: AWS, Microsoft Azure, and Google Cloud Platform.