



Contents:

- 1 Managing data for the digital experience era
 - 1 Cloudant's DNA
 - 2 Cloudant performance profile
 - 2 How the Cloudant Data Layer works
 - 3 Create a Cloudant Managed DBaaS account
 - 3 Store JSON data
 - 4 Index your data
 - 5 Access JSON data via RESTful API
 - 5 Replicate your data
 - 5 Monitor and visualize
 - 6 Scaling Cloudant
 - 6 Deployment options
 - 6 Private Cloud requirements
 - 7 Getting Started with Cloudant

Technical Overview: Anatomy of IBM Cloudant DBaaS

How Cloudant DBaaS delivers massive scalability, high availability, and performance

Managing data for the digital experience era

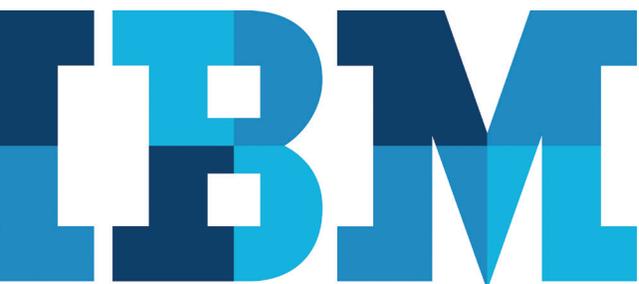
Today's applications are expected to manage a variety of structured and unstructured data, accessed by massive networks of users, devices, and business locations, or even sensors, vehicles and Internet-enabled goods.

IBM® Cloudant® is more than a database. It is your application's entire data layer, available as a fully-managed database as a service (Cloudant Managed) or as a system you install and run yourself IBM® Cloudant® Data Layer Local Edition (Cloudant Local). Cloudant helps developers create richer digital experiences for users by scaling elastically and making it easy to build advanced features like text search and geo-location services, while delivering data to devices non-stop, wherever and whenever users need it.

This paper describes what the Cloudant Data Layer is, what makes it distinct, and how it is used.

Cloudant's DNA

Cloudant was born and raised in the cloud era. It was built to be elastic, highly available, and to manage popular web and mobile data types like JSON, full-text, and geospatial data. Cloudant was created by combining the best open source code and thought leadership into an innovative NoSQL database platform that powers some of the biggest web and mobile apps in the world. Cloudant regularly contributes code back to these open source projects as well.



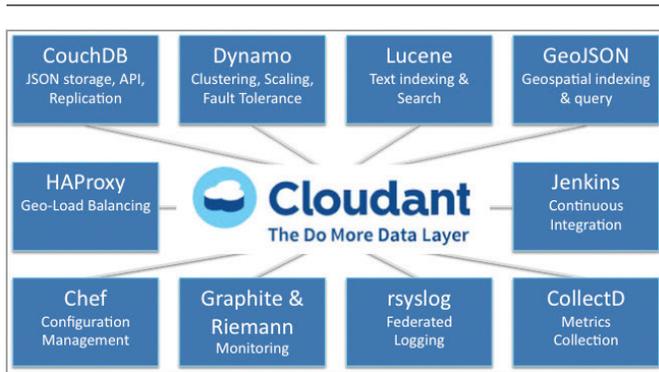


Figure 1: Cloudant leverages top open source technology providing compatibility with a variety of development tools, platforms and app infrastructures

Cloudant performance profile

Cloudant is best suited for applications that require an operational data store to:

- Handle a massively concurrent mix of low-latency reads and writes.
- Replicate and synchronize readable, writeable copies of data across multiple data centers or devices to enable continuous data availability and off-line data access for remote or mobile users with unreliable network connections.
- Manage multi-or un-structured data.
- Enrich applications with location-based (geo-spatial) services, full-text search, and real-time analytics.

The following tables provide real-life performance data characteristics for actual customers:

Large Single-Tenant Dedicated (single-tenant) Customer

Read:Write ratio	20 : 1
Data volume	130 TB
Transactional throughput	Over 2 billion requests per day
Latency	Under 10 milliseconds
User base	Global (mobile app)
Cluster growth	From 6 servers to over 200 in 12 months, spread across two geographically distinct data centers

Note: : read:write ratios vary from customer to customer. Cloudant handles mixed workloads that are read heavy, write heavy, or evenly balanced.

Large Multi-Tenant Customer

Read:Write ratio	3 : 1
Data volume	215 GB
Transactional throughput	1 million requests per day (42,000 per hour)

These are general guidelines to help scope Cloudant’s data layer performance. As application requirements change and workloads grow, Cloudant’s single-tenant (dedicated) clusters can grow along with them by adding nodes to scale up performance. Billed at a recurring monthly rate, many customers use this option to “pay as you grow.”

How the Cloudant Data Layer works

The Cloudant Data Layer is massive; your databases grow within it, not out of it, and we continually monitor and grow Cloudant to meet our customers’ collective storage and performance demands.

Application data resides in logical databases that you create. Cloudant physically stores the data on server nodes, which are joined into multi-tenant or single-tenant clusters that Cloudant manages as an on-line service, or that you manage yourself on your own private cloud using Cloudant Local. Cloudant clusters are easy to distribute across private or public-cloud data centers, enabling you to create a global “data delivery network” that pushes data to the network edge so that it’s always available and fast to access.

For Cloudant Managed customers, Cloudant automatically handles load balancing, clustering, growing/shrinking the clusters, and high availability of the system. As your database size and user load grow, Cloudant scales the data layer for you to keep up with the workload. Cloudant Local customers are responsible for all of these management tasks, but gain the deployment flexibility to install Cloudant on their own infrastructure behind their own firewalls.

The Cloudant team provides streamlined dev-ops tooling and training to help Cloudant Local customers keep their clusters running and growing.

Getting started with IBM Cloudant

Getting started and using Cloudant is very straightforward:

1. Create a Cloudant managed DBaaS account

The Cloudant Managed DBaaS saves you the trouble of choosing and provisioning hardware, installing and configuring DBMS software and load balancers. You simply sign up for a no charge account at Cloudant.com, and your data layer is ready. You're provisioned a custom URL against which you can immediately begin creating databases and working with your data.

When you activate your Cloudant account, you choose a physical platform (such as IBM® SoftLayer®) and geographic zone (e.g., US, Europe, Asia, etc.) as the primary location for your data layer. Typically, the choice is based on where your application code is hosted in order to get the benefits of app and data layer co-location. You can move your data layer from one hosting provider or location to another at any time and sleep more knowing Cloudant data experts are monitoring your data 24x7.

Creating multi-tenant versus single-tenant accounts

By default, your data layer will reside on a multi-tenant cluster that is also used by other multi-tenant account holders. Databases within multi-tenant clusters are secure and not accessible by other Cloudant users unless you explicitly set sharing permissions with others to enable team development. Cloudant includes special IO elevators that help ensure your application performance isn't impacted by "noisy neighbors." For those who prefer dedicated data layer resources, Cloudant can set up single-tenant clusters on dedicated hardware that can span multiple data centers or even cloud providers to guarantee that your distinct performance and availability requirements are met.

Database-as-a-Service and Service Level Agreements

Database-as-a-Service (DBaaS) solutions from Cloudant enable you to buy into a robust data management "Service Level Agreement" (SLA) instead of merely buying into a database technology. This greatly simplifies application development and delivery.

2. Store JSON Data

Cloudant is designed to store, index and report on flat collections of self-describing JSON documents. If you are familiar with relational SQL databases, then JSON documents are analogous to rows, and the fields within them are analogous to columns. A "database" in Cloudant is a collection of JSON documents. The notion of schema is not enforced within a database; therefore, the structure of one JSON document can vary widely from another document within the same database.

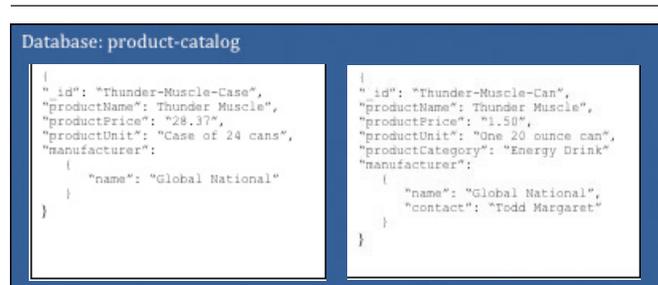


Figure 2: JSON documents are analogous to rows in relational databases, and they contain field types like numbers, strings, dates, Booleans, sub-structures, arrays, and attachments

Cloudant assigns each JSON document a distinct identifier (`_id`). Document fields can be numbers, strings, Booleans, objects, arrays, dates (JSON strings serialized/deserialized as JavaScript dates), or attachments. There is no limit to the number of elements or to the size of a document. You can index and query on any field in any JSON document in a database.

You can safely add new document types to a database or change the content structure of existing documents as needed. No data migration or schema updates are required as in a relational database. This flexibility is popular for supporting agile development practices, eliminating the complexity of database schema changes to support changing codebases. It also makes Cloudant an excellent fit for handling variably structured data such as product catalogs, experiment results, content management information, and other kinds of records that don't fit particularly well within relational tables.

Documents are accessed via HTTP using a RESTful web services API. Database access will be explained in more detail later in this white paper.

3. Index your data

In relational databases, an “index” is generally thought of as a performance optimization. In Cloudbant’s NoSQL system, you can’t get data out of the database without one.

In Cloudbant or CouchDB, an index is more akin to a materialized view. In fact, an index is the primary means of accessing data within a database. All databases automatically have a primary index based on each document’s reserved `_id` field. Users can build secondary indexes on any other fields within a document, and the system incrementally updates all indexes as JSON documents change. As with CouchDB, Cloudbant users can define secondary indexes as MapReduce “view” functions written in JavaScript. Indexes are typically exposed as part of the RESTful API’s URL path, against which queries are run.

It is worth noting that distinct to Cloudbant are additional indexing systems built into the service to organize different data types and to abstract the complexity of the traditional CouchDB MapReduce view system:

Cloudbant Query

Cloudbant Query is Cloudbant’s declarative query system that borrows syntax from MongoDB’s own declarative querying system. Cloudbant Query creates a data access language that is familiar across multiple NoSQL document-oriented databases.

Cloudbant Query is more performant than the traditional JavaScript-based views system because it is implemented in Erlang. It runs closer to Cloudbant’s core database engine, which is also implemented in Erlang. The system also allows users to easily define one index against which they can run many different types of queries, instead of defining multiple views to support complex queries.

Chainable, incremental MapReduce

Cloudbant includes several ways to index and query your data. The traditional way is to create special “design documents” in the database that contain MapReduce functions written in JavaScript. The Map function defines which JSON documents belong in the index and which fields to include from each member JSON doc. The optional Reduce function defines an aggregate operation to perform against the data in the index such as sums, averages, counts, etc. Every design document has a distinct URL, and accessing it from the client app returns the data in the index (or some subset you define using an optional query parameter in the URL) to the client as JSON.

MapReduce indexes are updated incrementally. Cloudbant distributes the MapReduce functions to all the nodes in your database cluster. Changes to JSON documents trigger incremental index updates rather than having to rebuild the whole index. This reduces IO overhead for the Cloudbant Data Layer, and also helps to enable you to perform faster, more near real-time analyses of your data.

To further improve performance and make development easier, Cloudbant helps remove the restriction that there can be only one map phase and one reduce phase. “Chainable” MapReduce queries allow the output of one MapReduce job to be persisted and fed to a chain of subsequent MapReduce jobs. This makes it easier and faster to create more advanced analytics on operational data.

Lucene Indexing and full-text search

In order to handle full-text search, Cloudbant integrated Apache Lucene text indexing and search into Cloudbant. This helps to enable you to enrich indexing and querying of your JSON data with:

- Ranked searching – search results can be ordered by relevance or by custom sort fields
- Powerful query types – including phrase queries, wildcard queries, proximity queries, fuzzy searches, range queries and more
- Language-specific analyzers
- Faceted search and filtering
- Bookmarking – paginate results in the style of Web search engines

GeoJSON-based 2D and 3D geospatial indexing and querying

GeoJSON support was also integrated into Cloudbant to enable the development of high-precision location-based services, which are a common requirement for mobile and sensor network applications. Unlike most other NoSQL databases, which only support limited geospatial processing such as proximity and bounding boxes, Cloudbant allows you to:

- Build queries using bounding polygons, rectangles, circles, or ellipses
- Perform nearest neighbor, route optimization, and predictive path analyses
- Store a wide range of data types, including complex geometries and metadata
- Use coordinate reference system (CRS) libraries since Cloudbant supports any CRS

4. Access JSON Data via a RESTful API

JSON documents can be retrieved, stored, or deleted individually or in bulk using GET, PUT, POST, DELETE statements over HTTP(s). Cloudant always encrypts data in flight. In Cloudant, JSON documents can store binary attachments right alongside other JSON data types, which makes it easy to manage media files, or even code.

Example: reading a document

Cloudant's RESTful API makes every document in a database accessible as JSON via a URL; this is one of the features that makes Cloudant so powerful for web and mobile applications. You can hit these endpoints using the curl tool on the command line, or even directly in your browser's address bar.

This example uses a sample database (called `crud`, short for create, read, update and delete) located at `https://<accountname>.cloudant.com/crud/`. It holds a single document with the primary document `_id` of `"welcome"`.

To access it via the API, simply append the document's `_id` to the URL of the database, as in:
`http://<accountname>.cloudant.com/crud/welcome/`. Access a working example at `http://broberg.cloudant.com/crud/welcome/`.

The Cloudant API is compatible with Apache CouchDB. Therefore you can access Cloudant from CouchDB-compatible client libraries, tools, and frameworks available for all major development platforms.

5. Replicate your data

Multi-master replication enables Cloudant to distribute readable – and writable – replicas of your data across multiple data centers, devices, and cloud providers and keep changes to them synchronized. This increases up time and reduces access latency by connecting users to the closest copy of the data. It does for mutable application data what a content delivery network (CDN) does for static content.

Compare this to the “master-slave” replication architecture in other popular database-as-a-service offerings and most relational databases, where only one replica can be updated (bottleneck); all others are read-only.

You can replicate between Cloudant Local databases in your data centers and Cloudant Managed DBaaS running on public cloud platforms. You can also replicate between Cloudant and Apache CouchDB (and many of its forks) to increase the size of your data cloud less expensively using free open source data stores in branch offices, points of sale, and others.

You have a lot of control over what gets replicated, how often it occurs, and how change conflicts are resolved if the same JSON document is changed more than one replica. You can define filters, which are rules that control what documents are replicated where. You can set replications to occur intermittently (and how often), or continuously so that data moves in real time. Cloudant has APIs that make it easy to view changes at a database or an account level, change conflicts, and to set rules for resolving conflicts

Cloudant customers use replication and sync to:

- To create copies (or replicas) of data in multiple locations for fault tolerance; if one source goes off line, there are others available to handle requests
- Push data to “Edge” databases – such as data marts or spreadsheets; making it great for independent analytic projects
- Enable off-line computing via hub-and-spoke databases and mobile sync
- Distribute subsets of data to specific partners, branch offices, etc. using replication filters. For example, if a country's data protection laws require data from or about its citizens remain stored within the country, you can set up Cloudant or CouchDB within that country to store local data and replicate changes to other data to servers outside the country.

6. Monitor and visualize

In order for Cloudant and its customers to monitor data layer performance, we have defined APIs and dashboards for collecting and reporting system metrics. Cloudant generates 50,000 metrics per second. You don't need to be an expert at interpreting this information because we do it for you to help keep your database running and growing smoothly.

If you want to view these metrics yourself, you can access your account metrics via the Cloudant dashboard and through APIs.

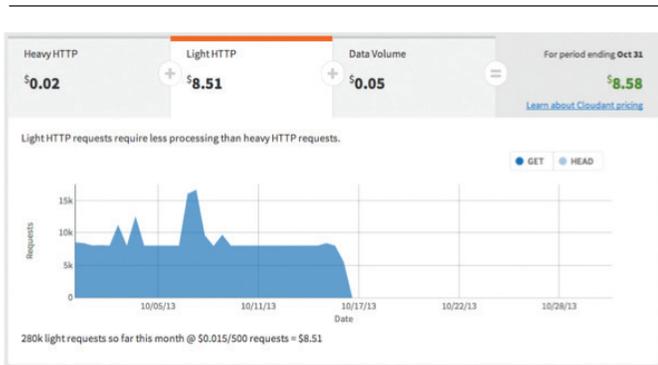


Figure 3: Data metrics are easily accessible through the Cloudant dashboard

Scaling Cloudant Elastic Cloud scalability

To meet the scaling requirements of today's applications, Cloudant implemented its own version of master-less, quorum-based clustering in Erlang (a language designed specifically for creating massively parallel applications), and it handles:

- Cluster membership
- Routing and coordination of database interactions
- Coordination of distributed queries
- Remote procedure calls to get better performance and better resilience to node failures running jobs on remote nodes.

Tune-able eventual consistency:

Quorum-based clustering helps enable you to specify the number of copies of data that you want to store (for high availability), and how many of them must be written to disk or how many must match in order for Cloudant to consider the data safely written or safely consistent. These quorum values are known as N, W, and R.

For example, on a 3-node cluster, you could decide to store N=3 copies of the data, once copy per node. W=2 would mean Cloudant would tell you the data is safely committed after 2 copies have been written. R=2 would mean that Cloudant would consider a copy of read data to be consistent with others if at least 2 copies match.

The quorum values can be changed in order to tune performance and consistency in a partitioned environment (i.e., CAP theorem). By default, Cloudant helps optimize for availability.

The IO Queue (IOQ)

Cloudant handles billions of database interactions every day for thousands of databases. The IOQ is a sophisticated IO prioritization layer that analyzes the priority of each database IO request to ensure that all clients get their fair share of the IO layer. It makes sure low-latency requests get higher priority, while also ensuring that performance is good for lower priority IOs such as database compaction.

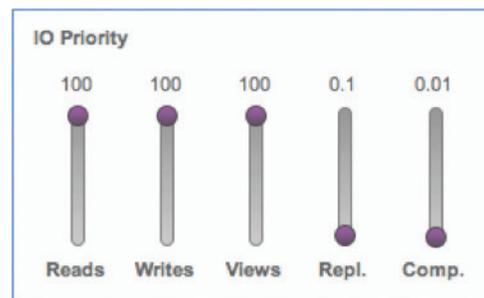


Figure 4: Cloudant's sophisticated IO prioritization layer (The IO Queue) analyzes the priority of each database IO request

Deployment options

Cloudant is available in several forms:

Cloudant Managed DBaaS

- A scalable, always-on, distributed NoSQL data layer that is managed 24 x 7 by the big systems experts at IBM Cloudant.
- Accounts can run on multi-tenant database clusters of database servers and load balancers, shared by many users, or on
- A single-tenant clusters running on dedicated hardware servers hosted within a top-tier cloud providers like IBM SoftLayer, Rackspace, Amazon AWS, or Microsoft Azure

Cloudant Local

- Puts the power of Cloudant DBaaS in the privacy of your data center and under your control
- Customers bring their own hardware and manage their own Cloudant clusters on private, public, or hybrid cloud platforms, using Cloudant dev-ops tooling informed by six years of managed DBaaS operation

Private Cloud requirements

Cloudant Local packages up the always-on technology of Cloudant with streamlined DBaaS operations tooling and all the institutional knowledge that Cloudant developer-operators have acquired over the years to put the power of

DBaaS in the privacy of your own data centers.

Cloudant local requirements overview

Server nodes

Cloudant Local requires five (5) machines to create a fully functional Cloudant cluster that ensures 24 x 7 availability.

- 1 primary load balance
- 1 failover load balancer
- 3 database servers

Software specifications

Cloudant Local runs on the following 64-bit operating systems:

Debian-derived Linux distributions:

- Debian 6.0.10
- Ubuntu Server 12.04.4

Red Hat-derived Linux distributions:

- Red Hat Enterprise Linux Server 6.5
- Community Enterprise Operating System (CentOS) 6.5

Best practice is for all server nodes to run the same OS.

Hardware Specifications

Database (DB) nodes

Minimum requirements:

Four (4) cores and eight (8) threads, such as Xeon E3-1270 V2, eight (8) GB of RAM and one (1) gigabit network.

Reference specifications:

For larger implementations, the minimum requirements are 12 cores and 24 threads, such as dual Xeon E5 2620, 64 GB of RAM, local SSD drives to meet data volume requirements of your usage, and a one (1) gigabit network.

Load Balancer (LB) nodes

Minimum requirements:

Dual core processor and 4 GB RAM, 500 GB local hard drive, and a one (1) gigabit network.

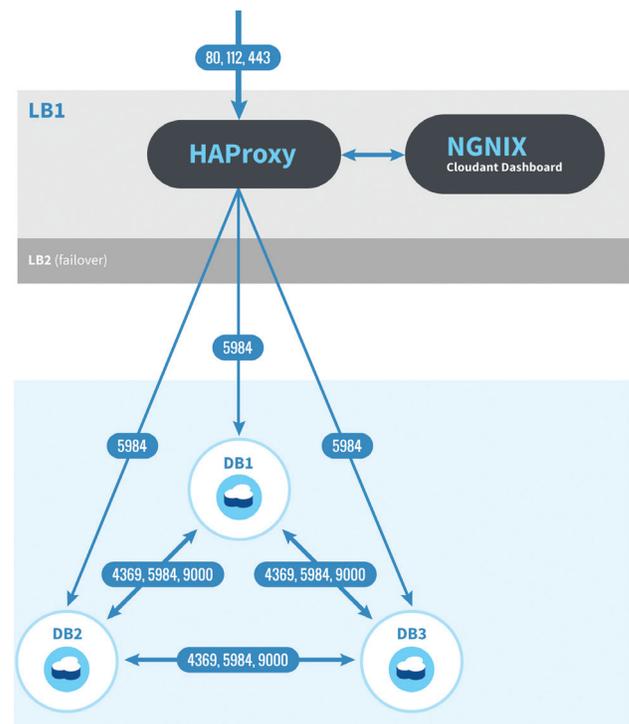
Reference specifications:

For larger implementations, the minimum requirements are a quad core processors and 8 GB RAM, 1 TB local hard drive, and a one (1) gigabit network.

Cloudant Local architecture diagram and ports

The following diagram shows the basic architecture of Cloudant Local, including two load balancers (lb1 and lb2), and three database nodes (db1, db2, and db3).

In the diagram, the lines and accompanying text identify the communication paths and ports that are used by the components of Cloudant Local.



Getting started with Cloudant

Cloudant is available as a fully managed NoSQL database as a service (DBaaS) for fast, turnkey provisioning, and worry-free data management. It is also available as Cloudant Local, which puts the power of the Cloudant DBaaS in the privacy of your data centers. You can even connect Cloudant Local and Cloudant Managed DBaaS databases together to form hybrid cloud databases for the greatest balance of cloud cost, reach, performance, and compliance control. Simply sign up for a no charge account and get started at <https://cloudant.com>

For more information

For more information, please contact your IBM representative or IBM Business Partner, or visit: cloudant.com/ or ibm.com/cloudant



© Copyright IBM Corporation 2015

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
February 2015

IBM, the IBM logo, ibm.com, and Cloudant are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at: ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other product, company or service names may be trademarks or service marks of others.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on the specific configurations and operating conditions. It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM product and programs. **THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.** IBM products are warranted according to the terms and conditions of the agreements under which they are provided.



Please Recycle