



# **MongoDB NoSQL database on Cloud-A**

**CLOUD-A**

**GEOFF SULLIVAN**

**09.15.2015**

[Introduction](#)

[Intro to NoSQL \(via mongodb.com\)](#)

[NoSQL Database Types](#)

[The Benefits of NoSQL](#)

[NoSQL on Cloud-A](#)

[Compute](#)

[SSD Volumes](#)

[Database Management via Cloud 66](#)

[Best Practises](#)

[Instance Sizing](#)

[Monitoring](#)

[Security](#)

[Backups](#)

[Duplicity](#)

[mongodump](#)

[Moving Forward](#)

## Introduction

Cloud-A provides a flexible, scalable cloud infrastructure platform to allow users to select the right tool for each job. Cloud-A has recently announced its technology partnership with [MongoDB](#), to provide a supported, scalable NoSQL solution for Cloud-A users.

MongoDB is a next generation database for **modern application architecture** that accelerates time-to-market with less resources, reduces risk for mission-critical deployments all with a lower total cost of ownership when compared to other NoSQL databases.

MongoDB has many use cases but works particularly well for [Big Data & Operational Intelligence Solutions](#), [Product Data Management](#), [Mobile Application Backend](#), [IoT applications](#) and [Content Management Systems \(CMS.\)](#)

This whitepaper will outline how you can deploy and manage your own mongoDB solutions on Cloud-A..

## Intro to NoSQL (via [mongodb.com](#))

NoSQL encompasses a wide variety of different database technologies that were developed in response to a rise in the volume of data stored about users, objects and products, the frequency in which this data is accessed, and performance and processing needs. Relational databases, on the other hand, were not designed to cope with the scale and agility challenges that face modern applications, nor were they built to take advantage of the cheap storage and processing power available today.

### NoSQL Database Types

- Document databases pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.
- Graph stores are used to store information about networks, such as social connections. Graph stores include Neo4J and HyperGraphDB.
- Key-value stores are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or "key"), together with its value. Examples of key-value stores are Riak and Voldemort. Some key-value stores, such as Redis, allow each value to have a type, such as "integer", which adds functionality.
- Wide-column stores such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows.

## The Benefits of NoSQL

When compared to relational databases, NoSQL databases are [more scalable and provide superior performance](#), and their data model addresses several issues that the relational model is not designed to address:

- Large volumes of structured, semi-structured, and unstructured data
- Agile sprints, quick iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Efficient, scale-out architecture instead of expensive, monolithic architecture

## NoSQL on Cloud-A

Generally, the public cloud provides some unique benefits for running large data sets when compared to on-premise infrastructure. Cloud-A's infrastructure-as-a-service will behave very similarly to running an NoSQL deployment on your own, physical hardware, but with the added benefit of usage-based pricing, powerful APIs and the ability to scale on demand.

### Compute

Cloud-A [compute](#) provides the SSD backed performance required for NoSQL deployments. With over 18 pre-configured and optimized VM "flavours" including our **Big Data High Memory®** and **Big Data High Compute®** flavours, users have the ability to select the right configuration for each job, and pay for them as they use them.

### SSD Volumes

Cloud-A [volumes](#) allow users to directly attach SSD block storage to their VM instances to instantly scale storage as required. This ensures that the right amount of storage is allocated at the right time so no longer will you have over utilized or underutilized storage resources.



### Database Management via Cloud 66

Managing a scalable MongoDB deployment will require a deep understanding of how MongoDB replica sets work and how to use them. While some people will want to undertake this learning curve with the great resources available [online](#), there is an alternative.

Our Partner [Cloud 66](#) provides everything you need to deploy, manage and protect your applications on Cloud-A. Cloud 66 provides [database support](#) for the MongoDB with no need for additional configuration after deployment.

Learn more about how Cloud 66 supports MongoDB replica sets [here](#).

## Best Practises

### Instance Sizing

When starting out, Cloud-A 1GB or 2GB flavours might be the right fit for getting comfortable with mongoDB and for the development stage. As you scale up towards production, it will likely be required that you scale up the resources for your deployment. Cloud-A instances can simply be scaled using our snapshotting functionality. Learn more about scaling up your servers [here](#).

Check out our [Cloud Server Pricing](#) to create a budget for your project.

### Monitoring

As with any database server, monitoring is extremely important for your MongoDB instances. MongoDB has some built in functionality that will provide you with statistics and information about data set and index sizes, but additional resources may be required to provide you with the analytics required to properly manage a healthy MongoDB deployment.

MongoDB provides [MongoDB Cloud Manager](#), which is a cloud-based deployment monitoring tool and backup solution. The tool provides custom dashboards and reporting for your MongoDB instances.

### Security

It is important to properly lock down your Cloud-A instances with Security Groups so that MongoDB is only accessible from known safe systems. By default, a raw instance on Cloud-A comes with port 22 (SSH) open which will accept any incoming TCP connections. All other ports will reject incoming TCP connections.

Once you are set to deploy multiple instances you can open up specific ports so that your MongoDB can communicate with other known safe servers. You can [configure your security groups](#) to open up port 27017 (for mongod) to specific IP addresses.

Eventually, once you are past the point of setting up a single node and replica set, you will need to open the ports required for sharding (27018 and 27019.) [Sharding](#) allows you to store data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

## Backups

In addition to the aforementioned [MongoDB Cloud Manager](#), which provides a fully managed backup service for MongoDB, there are several other, open source backup tools that can be used for backing up your MongoDB deployments.

## Duplicity

[Duplicity](#) is our go-to open source backup tool, mostly because it is simple to use, it is compatible with Cloud-A Bulk Storage and it is free!

Check out our docs on How to [use Duplicity with Cloud-a Bulk Storage](#).

### **mongodump**

To backup MongoDB to Cloud-A Bulk Storage you can use the `mongodump` command to create a backup of that database:

```
mongodump --host 127.0.0.1:28017 --db <DB_NAME> --user <USER_NAME>
--password <PASSWORD> --out /var/backups/mongo/
```

Note that if you want to backup all MongoDB databases you can omit the `--db` option.

Again replace the corresponding options with your values, and make sure the `/var/backups/mongo` folder exists and is writable to you. Once you have your local database backups working, you can configure Duplicity to backup the `/var/backups/mongo` directory to Cloud-A's [Bulk Storage](#).

## Moving Forward

Looking for more information about MongoDB and how best to use it on Cloud-A? We have access to MongoDB technical resources. Reach out to our [team](#) today.

FOR MORE INFORMATION CONTACT: GEOFF SULLIVAN  
GEOFF@CLOUDA.CA