



An Executive's Guide to Database Management ROI







Table of Contents

1. Executive Summary	4
1.1. The Value of ClusterControl	4
1.2. Introduction to Severalnines & ClusterControl	7
2. Why Open Source	9
2.1. The Value of Community Innovation	10
2.2. Understanding the Open Source Database World	11
2.3. Understanding the Needs of Open Source Databases	11
2.4. The Downsides of Open Source	12
3. Understanding the Costs	13
3.1. Understanding the Cost of Free	13
3.2. Understanding the Cost of Downtime	15
3.3. The Cost of New Knowledge	15
3.4. Managing Open Source Databases - A Comparison	16
3.5. Database Total Cost of Ownership (TCO)	17
4. Where ClusterControl Fits	22
4.1. How Many Tools to do What ClusterControl Does	23
5. About ClusterControl	26
6. About Severalnines	26
7. Related Resources from Severalnines	27

Executive Summary

1.1. The Value of ClusterControl

In this whitepaper we will demonstrate to you how you can deploy and manage your open source databases in multiple different ways... either with high levels of manual management expertise or third party support to bring the needed level of expertise into your organization.

ClusterControl, an all-inclusive database management system, allows you to achieve the same end result of both scenarios without the costly internal or external labor costs all while reducing downtime. How does it do this?



Automation of Daily Tasks - Basic daily tasks are easily performed but are time consuming. Management tasks like backups and restore verification, upgrades, and failover/recovery are also very manual. Typically automation is introduced into the organization with the person who manages the databases writing scripts (or downloading them from the internet) and using them to deploy and manage the infrastructure. It is often hard to know to what extent these manual scripts are reliable, as good scripts take time to write and require a lot of testing - especially if you are going to use them on your production databases. At Severalnines, we are experts at databases. ClusterControl has these daily tasks automated, using proven methodologies and expertise, with a point-and-click or command line interface.

Automation of Complex Tasks - It takes a lot of work to operate a database in production, especially when it comes to high availability setups with multiple servers and data centers. ClusterControl automates the deployment, monitoring, management

and scaling of these distributed systems, and makes configuration easy with multiple-choice options through a point-and-click interface.

Deployment Using Proven Methodologies - The management features found in ClusterControl follow best practices that our team of database experts have learnt over many years working in the database industry. They are thoroughly tested and used in production by thousands of companies.

Data Integrity - It is not hard to deploy a high availability database, but is it production ready? Especially when it comes to replication or clustering, where data is distributed across multiple servers, it is important to manage the system as a whole and understand if and when the data is getting out of sync. Server failures or breakdown in replication can compromise your data. For recovery purposes, it is also important that data is continuously backed up and the backups are regularly verified.

Reducing or Removing Downtime - ClusterControl's unique combination of monitoring and alerts coupled with automated failover and scaling functionality make it highly unlikely for your database to go down. In addition, by using its proven methods for deployment and configuration you will experience less errors in production.

Reduction in Staffing Levels & Skill-set Requirements - With the automation found in ClusterControl you don't have to have a specialized database administrator to monitor query behaviour, manually deploy nodes or clusters, manage configurations, detect anomalies in performance, repair nodes or clusters, make backups or restore them or load balance your queries. ClusterControl does all that through either a GUI or CLI that can be easily operated by a SysAdmin. In addition, by automating a large part of the maintenance, we save you (over)time.

Enterprise-Grade at a Fraction of the Cost - While the majority of this document compares using ClusterControl to using other open-source databases with support or cobbled together tools, the reality is that ClusterControl is a very complete tool with nothing else like it on the market. We are on par with other enterprise software systems from the likes of Oracle and Microsoft, but at a greatly reduced cost.

Severalnines increased our streaming speed by 76% and this has greatly improved the delivery of content to our customers. The implementation took only two months to complete and saved us 12% in costs.

Thom Holliday, Marketing Maanger at StreamAMG

A Patchwork of Tools & Scripts

Most likely today your database infrastructure looks like a patchwork of tools with some things managed through scripts, others through various tools. Often times a tool is introduced to solve problem "x" but without considering problem "y." It is also not straightforward to integrate different tools as very often, they have not been designed to work with each other. Since every tool has its own way of doing something, there is a training aspect for the team or organization that will rely on it for collaboration. In addition it is often the case that a tool that works for a developer on a smaller development environment might no longer work as part of a larger production infrastructure.

When introducing new tools, it is important that they can be integrated with existing ones that are already in use by the ops team. Such is the nature of infrastructure nowadays that it consists of many different components, and it would be impossible to

find one tool to manage everything. Databases, as the source of truth for applications, are specialized components with their own management requirements. ClusterControl handles the entire deploy-monitor-manage-scale lifecycle of the database, but it can be also integrated with other management software.

If your organization is using a monitoring tool like Nagios, ClusterControl can integrate with it so that all alerting is channeled via Nagios. If Puppet is used for configuration management and deployments, the ClusterControl CLI can be used to manage what is already deployed. If the DevOps team has a particular chatbot used to alert when something goes wrong, that's fine... ClusterControl integrates with Slack, VictorOps, Telegram and others.

With Severalnines' help we've been able to deploy a centralised system across Europe... it is the database management life-saver for a fast-paced business like ours.

Stefan Rehlegger, System Architect at Eurofunk

Driving Digital Transformation on a Budget

The rise of new technologies and different IT approaches (like containers, cloud deployments, automation and DevOps) all have an impact on databases and how they are managed. The database, however, is usually the last thing you touch due to the high risk of affecting data integrity.

While your data is valuable and the stakes are high, you may end up with a database infrastructure that may be badly out of date. An antiquated infrastructure leads to technical debt and long-term slowness that sink transformation projects.

The challenge is that your company's IT budgets is most likely flat, and they're not going up anytime soon. Of the budget that you do have, 80% remain allocated, as they have been for decades, to support functions that don't move the revenue needle-functions like system maintenance, upgrades, and security patches.

That leaves you only 20% or less of your flat IT budgets for the innovative work that helps your company take business from your competitors. That percentage isn't nearly enough, especially as digital upstarts-free of the legacy IT systems that require so much operational heavy lifting-snap at the incumbents' businesses, sometimes upending their business models.

Driving Collaboration Within the Organization

Collaboration between development and operations teams are indispensable when building a modern and agile IT infrastructure. Can your application team spin up a development environment in minutes? Can the team leverage cloud based platforms and shut down idle infrastructure? Can the team react to a database issue, without having to rely on the person who set it up?

ClusterControl is designed for use by DBAs, SysAdmins, and Developers. Operations personnel with no database-specific experience can use ClusterControl to spin up entire database environments and perform day to day management tasks like backups, troubleshooting performance, recovery from failures, and so on. Eliminate dependency on the individual, instead empower your entire team so everyone can contribute to moving their projects forward.

The Bottom Line

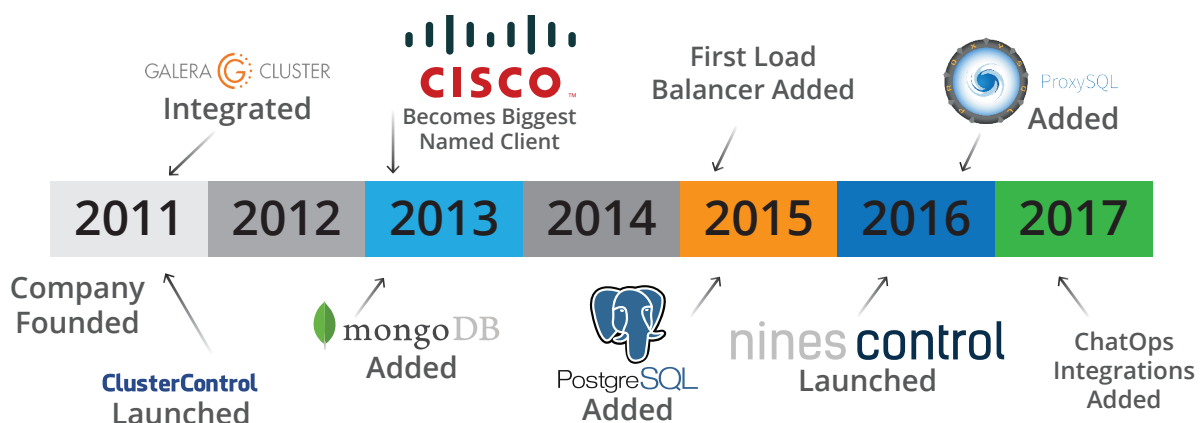
Up-front costs for software licenses are sometimes difficult to justify in your organization, especially when they are used to manage free open-source technologies. However when you consider the costs of headcount, support, cobbled-together tools, and downtime, the value of investment in a system like ClusterControl becomes apparent.

1.2. Introduction to Severalnines & ClusterControl

The Severalnines team members, mostly ex-MySQL AB staff, have dedicated over a decade of their careers helping telcos and high-traffic web properties deploy and manage high availability database clusters. These were mainly based on MySQL Replication or MySQL NDB Cluster setups. Managing high availability required that the solution was well configured in the first place, that monitoring was robust and provided enough information about workload and anomalies. Production incidents, like server failures, had to be handled in an automatic way so as to minimize downtime. A failover that required an administrator to troubleshoot the issue and execute commands to bring up a failed database or move services to another functioning database server, would just take too long - much longer than the 5 minutes of annual downtime that can be afforded to IT systems claiming 99.999% availability.

There has been a boom of open source databases in the past decade, and while many new technologies have emerged, from clustering frameworks for MySQL/MariaDB to NoSQL databases and database-aware load balancers, there has been a lack of tools out there to make these technologies easy to use. ClusterControl is an all-inclusive database management system that provides advanced deployment, management, monitoring, and scaling functionality to get open source databases up-and-running using proven methodologies.

Since its creation ClusterControl has strived to achieve its vision to make open-source database technologies stable, secure and easy-to-use for everyone. The timeline below details some of the major achievements of ClusterControl to date.



Automating database management saves us time in writing new code for checks and maintenance tools.

Andy McCulloch, IT Manager at ILL

Currently ClusterControl offers a full-suite of enterprise-grade features to help organizations deploy, manage, monitor, and scale their open source database environments. It provides full-lifecycle functionalities for database clusters and single instances, whether on premise or in the cloud - picture it like a 'virtual team of database administrators'.

Database Deployment (Technologies Supported)

- Databases - MySQL, MariaDB, MongoDB & PostgreSQL
- MySQL-based: Galera Cluster, MySQL NDB Cluster, Oracle Group Replication, GTID-based Master-Slave Replication, MariaDB, & Percona
- MongoDB: ReplicaSets, Sharded Clusters from MongoDB Inc. and Percona
- PostgreSQL: Streaming Replication
- Load Balancers - HAProxy, ProxySQL, Keepalived, MaxScale

Key Management Features

- Automated failover & recovery
- Backup, Restore and Verification
- Advanced security
- Topology management
- Array of automation features
- Developer Studio for advanced orchestration
- Operational Reports

Key Monitoring Features

- Unified view across data centers with ability to drill down into individual nodes
- Full stack monitoring, from load balancers to database instances down to underlying hosts
- Query Monitoring
- Database Advisors

Key Scaling Features

- Replication architectures including MySQL Group Replication
- Point-and-Click load balancing deployment and configuration
- Workload distribution features
- Database cloning

Why Open Source

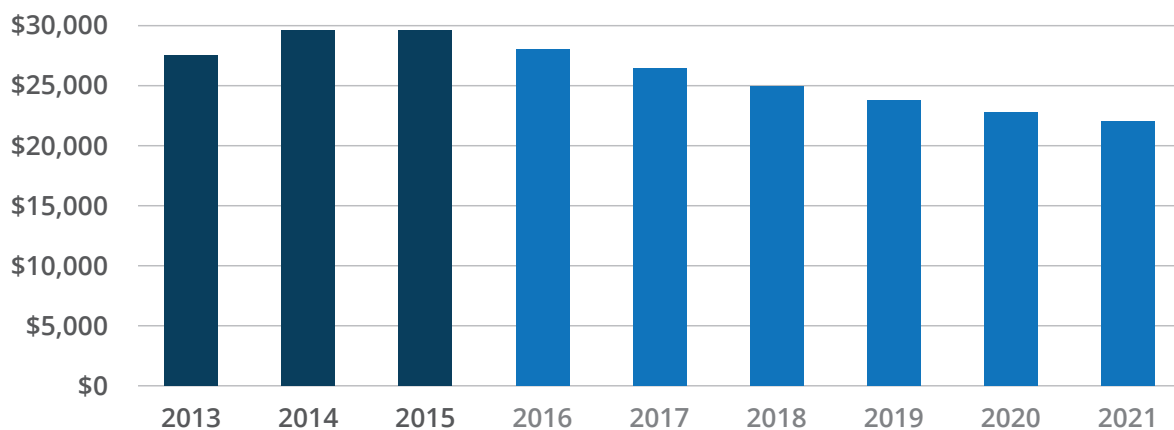
"By 2018, more than 70% of new in-house applications will be developed on an OSDBMS, and 50% of existing commercial RDBMS instances will have been converted or will be in process," predicted Gartner in their research paper, State of Relational Open Source RDBMSs.

And according to db-engines.com, MySQL is the number two database in the world according to its "mindshare" calculation. Three of the top five databases used in the world are open source.

The enterprise commercial database market is expected to contract by 20% to 30% by 2021 according to Gartner.

COMMERCIAL DATABASE MARKET SIZE (\$m)

(Source: Gartner 2013-2015, Samadhi Partners 2016-2021 estimates)



Why is this happening? The reason is simple - the open source database market is mature. Open source databases have caught up in their ability to support business-critical enterprise workloads. Open source technologies are quickly becoming the new standard and this is especially true in the database world where the adoption of open source technologies are following a similar path as Linux, which now dominates its space.

Scaling from thousands to millions of products is a giant leap and that will require us to have a strong infrastructure foundation. Our back-end is reliant on different databases to tackle different tasks. Using several different tools, rather than a one-stop shop, was detrimental to our productivity. Severalnines is that "shop" and we haven't looked back. It's an awesome solution like no other.

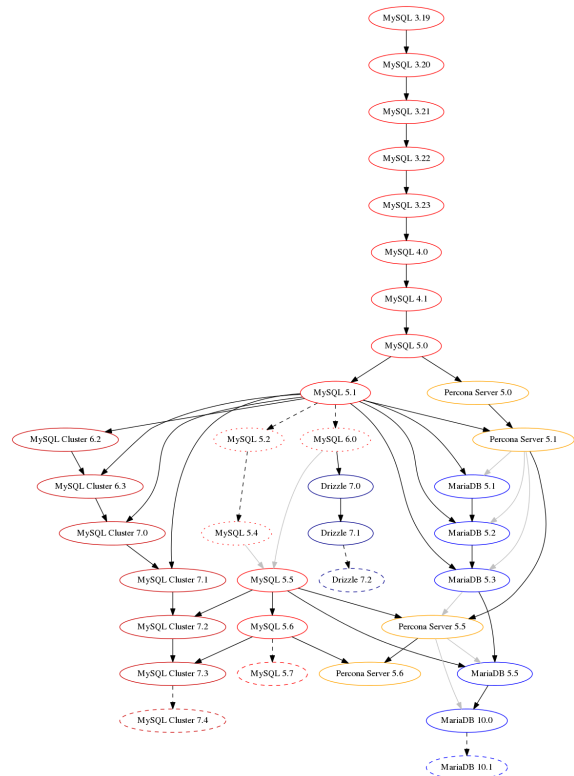
Zeger Knops, Head of Business Technology at vidaXL

2.1. The Value of Community Innovation

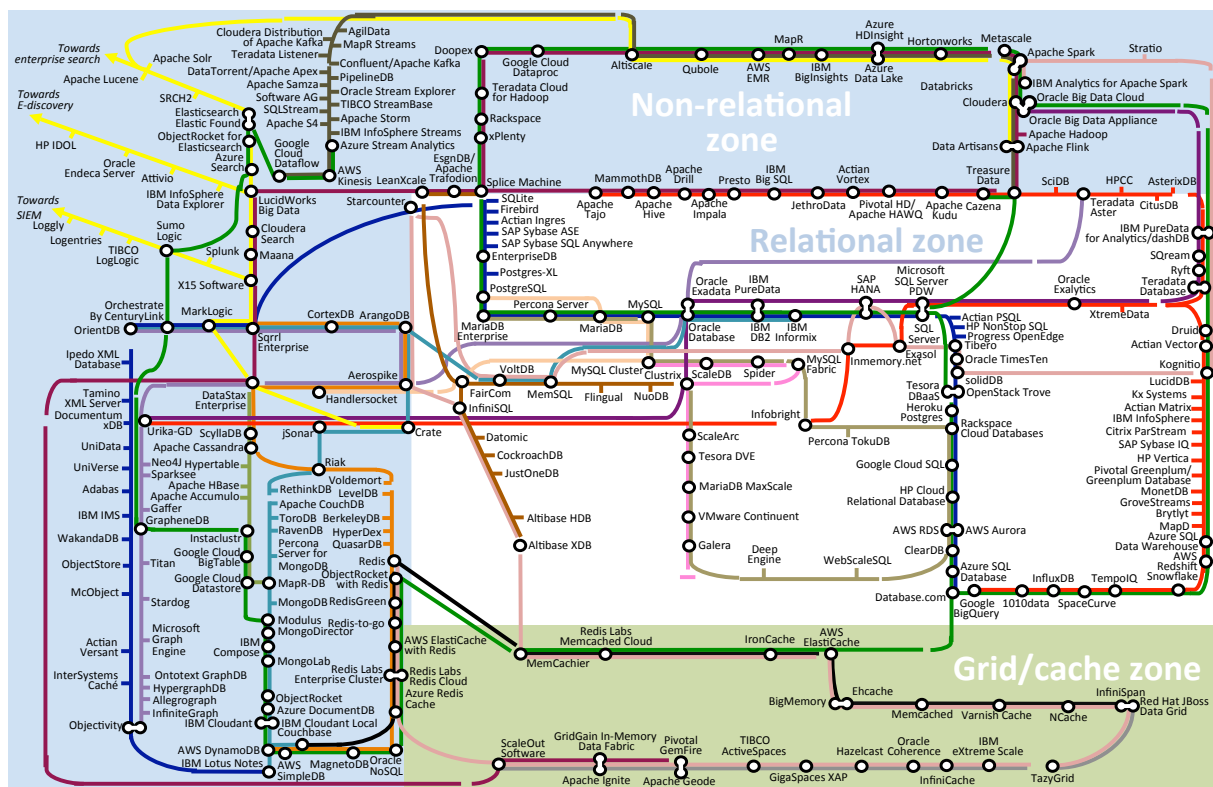
The open source community thrives through the innovation of its contributors. There are dozens of forks of MySQL alone (see chart on right), each offering their own unique improvements or modifications to the original version.

The downside of this technology reality is that the IT leader may be the last to know which open source technologies (see chart below) are actually getting used within the enterprise. This widespread usage of open source technologies can't really be measured by analysts or market experts.

It is also estimated by Gartner that DevOps teams will have a larger influence on the data infrastructure, up to 30% by the end of 2018.



Source: <http://imgur.com/HMxj2ZW>



Source: Updated Data Platforms Map – January 2016 (451 Research)

So far, it looks like a small miracle and we are now leveraging this solution in other parts of our infrastructure.

Václav Adamec, System Engineer at AVG

2.2. Understanding the Open Source Database World

While the variations and forks of specific open source databases is an exciting topic, the reality is that “92.1% of DMBS revenue comes from the top 5 vendors” according to Gartner Analyst Merv Adrian.

This means as an IT leader you are going to, most likely, only be looking at a handful of technology choices when determining the future of your database environments.

While MySQL, PostgreSQL, MongoDB, Cassandra, Redis & SQLite dominate the open source database technology market it's disrupted by companies like Percona and MariaDB that offer technology / support power-plays to enable applications powered by open source.

Rank			DBMS	Database Model	Score		
May 2017	Apr 2017	May 2016			May 2017	Apr 2017	May 2016
1.	1.	1.	Oracle +	Relational DBMS	1354.31	-47.68	-107.71
2.	2.	2.	MySQL +	Relational DBMS	1340.03	-24.59	-31.80
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1213.80	+9.03	+70.98
4.	4.	↑ 5.	PostgreSQL +	Relational DBMS	365.91	+4.14	+58.30
5.	5.	↓ 4.	MongoDB +	Document store	331.58	+6.16	+11.36
6.	6.	6.	DB2 +	Relational DBMS	188.84	+2.18	+2.88
7.	7.	↑ 8.	Microsoft Access	Relational DBMS	129.87	+1.69	-1.70
8.	8.	↓ 7.	Cassandra +	Wide column store	123.11	-3.07	-11.39
9.	9.	9.	Redis +	Key-value store	117.45	+3.09	+9.21
10.	10.	10.	SQLite	Relational DBMS	116.07	+2.27	+8.81

Source: db-engines.com

2.3. Understanding the Needs of Open Source Databases

Open source databases, especially new ones brought into an organization, often come from a developer on a project team. It's chosen because it's free (doesn't impact the project's external spend), is documented, and meets the technical requirements of the moment. But an open source database is often like bringing a puppy into your home, while it's fun to play with at first and fits your needs...eventually there are longer term considerations.

Here are some things to consider when bringing an open source database into your organization.

Expertise: Do you have people throughout the different facets of your team who know how to manage and troubleshoot the database when things go wrong?

Maintenance & Security: Is your data protected? Security vulnerabilities tend to pop-up from time to time with any open source technology. Does your team know what's happening out there? Do you have an upgrade plan?

Backups / Restorations: Do you have copies of your database? How often? Where are they stored? Do they exist on different hardware? Different datacenters? Do you test the backup files to make sure the data is not garbage?

Downtime: What will you do if the database crashes? What is your procedure for recovery? Does your team know how to troubleshoot problems?

Time: How long does this database need to stay active? Is it a short term project or a long term investment? If it is long term, do you have the people in place to handle the items listed above? What happens if the person who set up the database leaves?

2.4. The Downsides of Open Source

While utilizing open source technology means that you have many people from all over the world identifying bugs and providing fixes, you also have many people around the world trying to exploit the technology to take advantage of its end users.

Increased Security Threats

In early 2017 it was estimated that more than 33,000 MongoDB installations had been hacked and their data held for ransom. Within a month after this attack the hackers shifted their target to unsecured MySQL databases.

The largest upside of open source technology (the fact that everything is known with no secrets or vendor locked-in surprises) is also its biggest weakness. With full knowledge of how a database functions, developing software to attack it is simpler. Because of this, a new kind of "open source race" has been created, creating and deploying security patches as fast as the hackers are developing new threats.

Deployment of these patches and staying aware of the threats requires time and database infrastructure competence. The vulnerability which allowed the massive MongoDB hack in early 2017 was known as far back as 2015, but those who were unaware of this issue did not take the time to remedy it resulting in two years of compromised data integrity and lots of ransom money down the tubes.

The majority of our business is online, we are extremely reliant on the infrastructure our business proposition is built upon. It was crucial that we had a system in place to efficiently manage our databases. ClusterControl allowed us to implement a high availability database cluster and easily operate it, which helped the biggest teams in the world access player data anytime, anywhere. Severalnines technology was so easy to use we were able to implement it ourselves

Francesco Nassano, System Administrator at Wyscout

Understanding the Costs

3.1. Understanding the Cost of Free

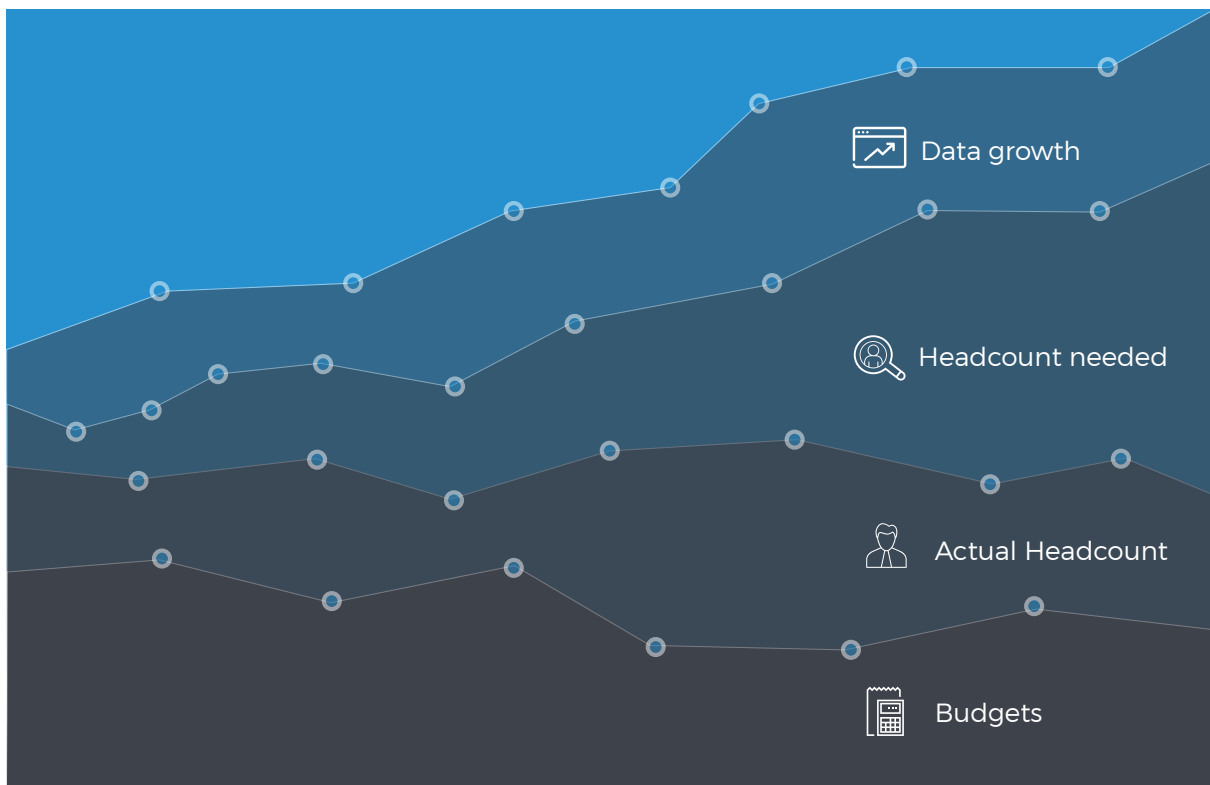
Michael Skok, Founding Partner of Underscore VC said in 2013 “Open source is eating the software world.” More and more enterprise organizations are both participating in the development of and using open source software as part of their mission critical infrastructures. And why not? There is something very attractive about what works great and is free. Budgets are ever shrinking, headcount ever disappearing and license costs of enterprise-grade software ever rising.

IT leaders are being forced to do more with less each and every year. While innovations and efficiency in new software technologies help with some of this, the gaps must be filled where the improvements lack.

In the database world, where the amount of data is expected to grow by three to five times every three years, costs are ever-increasing. Scaling and maintaining the database infrastructure continues to grow as well. This, coupled with reduced headcount and ever shrinking budgets, make managing the digital expansion even more of a challenge for the IT leader.

This increase in data, need for solutions, and lack of budget are some of the key things driving IT leaders to open source technologies.

But what is the **TRUE** cost of a free solution? With publicly available software, continually innovating with new forks, features and technologies, how can you decide what to do?

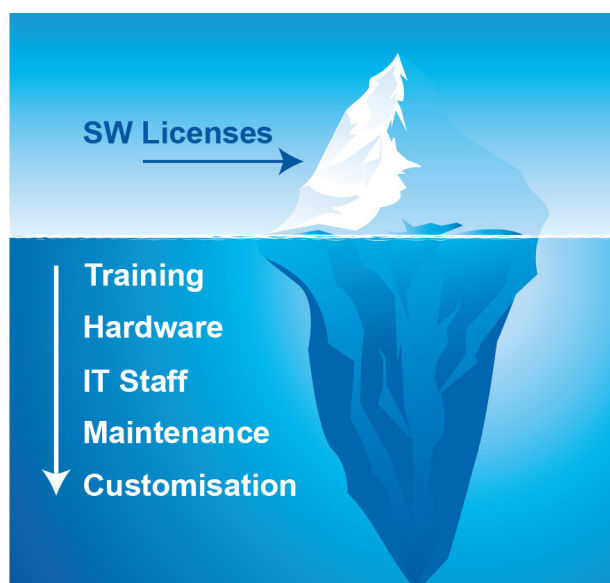


Homegrown or License.... You Pay Both Ways

According to IDC in their *Maximizing the Value of Enterprise Database Applications* report, it costs, on average, four times the purchase price to manage a database; IT leaders do not always take into account the “hidden” costs of bringing a new technology into the organization.

Knowledge - Learning a new open source database technology (see chart in section 2) takes time. Each fork, variation or specialized database comes with new documentation to read and understand, new operational procedures, new performance tuning and troubleshooting techniques to learn, new everything.

IT Staff - Noel Yuhanna, Analyst for Forrester Research, estimated that a typical DBA in a large enterprise can effectively manage up to 40 small (100GB) databases. As the size of the average database increases, so to does the decrease in the number of total number of databases which they can handle. But in small to medium sized organizations, having a DBA at all is a luxury so it often falls to the developers and system administrators to pick up the slack.



Maintenance - Maintaining a standard master-slave replication setup can be daunting to even a skilled database administrator. Achieving this maintenance can be done through two main avenues. One is through headcount, hiring a database administrator and equipping her with tools to maintain the environments. The other is through external consulting and remote DBA services.

Customization - There are also two avenues to take when it comes to ensuring your database performs with your application’s unique needs. You can architect the application for the database or customize the database for the application. With the first you quickly put yourself into a vendor lock-in situation which, once inside, it would be hard to move to another database vendor. The other takes you down the path of choosing a specialized/niche database that, although well suited for your use case and workload, might incur higher management/operational overhead. The open source database market is crowded with alternatives, as we saw in Figure 2.3 above. Therefore it is important to choose a technology that will be actively developed, maintained and supported for a foreseeable future.

It’s phenomenal software... I’m usually not impressed with vendors or the software we buy, because usually it’s over promised and under delivered.

Jordan Marshall, Manager of Database Administration at Black Hills Corporation

3.2. Understanding the Cost of Downtime

You are held to an SLA and only you can calculate the true cost of your downtime. In 2013, Amazon.com went down for “just” 40 minutes, costing the company an estimated \$4.8 million dollars in sales. It’s estimated today at their current revenue that they could lose up to \$250K per second for site downtime.

IHS did a study where they found that outages in 2015 cost those surveyed in excess of \$700 billion dollars. Gartner’s findings? On average \$5,600 per minute or well over \$300K per hour.

Return on investment is, largely, a simple calculation. Cost of doing nothing - cost of doing something. The reality is that databases do crash, data can get corrupted, and the hardware will fail at some point. Achieving true high availability requires, not just the implementation of a stable open source database deployed on reliable hardware, but also the introduction of database/query monitoring, automated failure detection, failover & repair, load balancing, scaling, a backup & restoration plan and more.

Developing all of those capabilities requires two things. Knowledge of how to do it and the time it takes to get it done. Each of those comes with a pricetag. The reality is that you are most likely reading this because a situation has occurred in your organization that resulted in downtime, coupled with high levels of employee overtime or external support spend to resolve the situation. While it is often difficult to justify an upfront cost to ensure a “security blanket” for your infrastructure. The reality is that in the long term, being proactive will ultimately save you money on your bottom line.

3.3. The Cost of New Knowledge

Open Source technologies innovate fast, but enterprises move slow. A new load balancing technology might be just the thing to take your application performance into the next level or reduce your downtime, but with reduced headcount and lack of knowledge, there is a real added cost of “knowing.” In a recent survey of DBAs, 62% said they were managing from two to five different database technologies in their organization. For each of these technologies there is a wave of new add-ons, tools, features, functions to improve, simplify or automate their environments. It is nearly a full-time job just keeping up with everything that is happening. Add to that the number of innovative technologies that your team probably never even hears about and before you know, your organization could be falling behind on essential technology advances.

This is a key difference between an Enterprise-grade, commercial solution and an open source technology. In the commercial database space roadmaps are clear, highly-engineered and tested, rolling (slowly) on a regular basis. The open source space is different, sporadic, and sometimes quick-to-fail, but equally fast in progressing.

An unfortunate side effect of this is also that open source technologies tend to be less user-friendly. The incentive for a technology contributor or creator to provide anything more than the bare minimum of what is required is fairly low. This requires even more time taken by your teams to learn the new technologies and stay on top of the changes and innovations. It is also worth noting that the open source space is big, and comprises of everything from single person projects to software developed, maintained and supported by commercial organizations.

3.4. Managing Open Source Databases - A Comparison

Managing open source database environment takes three main forms...

- **Doing it in-house manually**
- **Utilizing external support/DBA services**
- **Using a combination of in-house with professional tools**

Each of these requires a point person on your side of the application who understands the unique needs and details of your infrastructure; however the level of knowledge and job position needed on your side will vary based on which option you select.

Database Administrator (DBA) - DBAs will naturally be the most knowledgeable when it comes to deploying and managing database setups. The advantages of having them on your team is that you can rely on them more to achieve a stable and performant environment with little downtime. The disadvantage of having one is that most budgets do not allow for such a specialized position. As we mentioned before, a DBA in a large enterprise can handle up to 40 100GB databases. Most businesses database setups are not this large and having this level of specialization would be inefficient and add unnecessary cost to the bottom line.

System Administrator (SysAdmin) - More and more SysAdmins are being tasked with the additional role of deploying and managing database environments. SysAdmins offer a broad range of knowledge from hardware and software to networking and security, but rarely have specialized knowledge in database management. While most can easily deploy a standard replication setup, as soon as there is a need for automating failover and recovery, or tuning database performance or queries, the skill set level drops off rapidly.

DevOps - Like SysAdmins, most DevOps personnel would not have specialized database knowledge. There is much about DevOps still being debated. One camp thinks that DBA-like person on the team is just another type of operations specialist which could run in conjunction with the standard DevOps teams, others feel that the database management function should dissolve into another Ops part of the overall team.

Developer - Developers have a track record of introducing new database into the organization, especially free open-source ones. While designing or architecting the application, a developer will be met with unique challenges that they will need a database to solve. Once the project is complete, however, the database is often "dropped" on the lap of the SysAdmin or DBA to manage. This has the unfortunate side effect of creating disorganized setups inside the overall infrastructure that require different tools and skill sets to operate.

Database Support Model

Managing open-source databases through external support is another option to handle the deployment and management of the database environments. Support is provided from database technology creators and third parties to add a level of skill that you may not currently have in your organization. Some database technology providers offer specialized versions of the databases with additional features or access to tools that are not available as part of the free database options. Support is the typical revenue model for open source technology creators.

In addition to providing database support and tools, these providers also offer consulting or ongoing remote DBA support for their technologies. This would allow you to “outsource” some of the more complex items that may come with your application needs.

For the sake of this comparison we will compare the basic paid versions and assume three nodes. Pricing is estimated based on information found on their website.

Technology	MySQL	MySQL	MariaDB MySQL	Percona MySQL
Plan Level	Standard	Enterprise	MariaDB Enterprise	Percona Server for MySQL
License Cost	\$0	\$0	\$0	\$0
Support Cost	\$6,000	\$15,000	\$7,500	Not Public
Support Level	24-7	24-7	24-7	24-7
Tools Provided	Workbench Only	Multiple Individual Tools: Monitor, Manager, Security, Scalability, HA	Management, monitoring, recovery tools, plugins, & connectors	PMM, Percona Toolkit, Monitoring Plugins

Managing MongoDB

MongoDB offers a similar support option to those listed above. For its Enterprise level a user gets access to the enterprise version of its database, access to its operations management tool, and 24-7 support for an estimated [\\$12K per node \(or \\$36K for three\)](#).

The Percona Server for MongoDB fork of MongoDB Community offers most of the features found in the enterprise version.

3.5. Database Total Cost of Ownership (TCO)

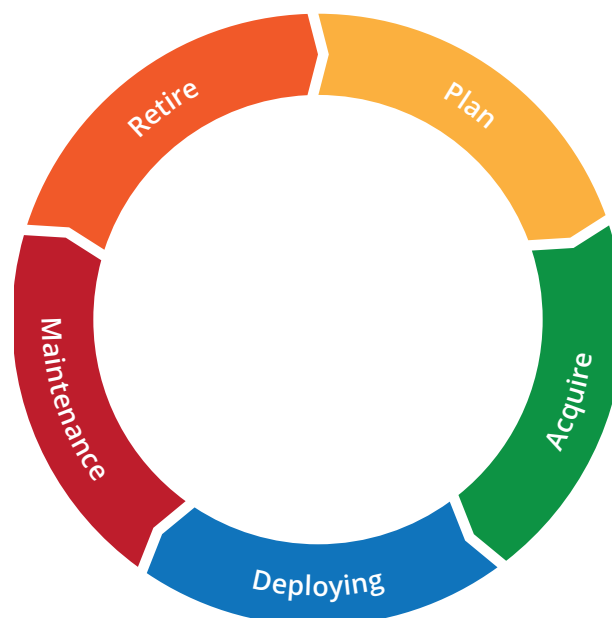
Cost analysis and cost effectiveness for databases are hardly ever performed, while it actually makes a lot of sense to perform such calculations. The easiest method of gaining insights into these is by performing a total cost of ownership (TCO) calculation. You might have theories on what your greatest cost factor is, but do you really know for sure?

Why would you perform such a TCO analysis? As with most research: prove your theory of the highest cost factor wrong. The TCO is a great tool to give a precise cost analysis and would give you some surprising insights!

Cost Factors for Databases

The cost factors for databases can be divided into two separate groups: capital expenses (CAPEX) and operational expenses (OPEX). Both cost factors are part of the infrastructure lifecycle.

Capital expenses are the costs you pay upfront during the **acquire** phase: hardware purchases, (non-recurring) licensing cost and any other one time cost factors like replacement parts. These expenses are a constant factor in the TCO and are spread out over the lifetime of your database servers. Most of these costs will happen in the acquire phase, however the replacement parts will obviously take place in the maintenance phase.



Operational expenses are the costs for running the database servers. As these costs are recurring, you pay them on a regular (e.g., yearly) interval and mostly during the maintenance phase. These costs include any ongoing support fees, data center/rack rental, power consumption, network usage and operational costs like (remote) hands and personnel. They also include sysops, DBAs and all costs made to facilitate them like desks, office space and training. Since these expenses are recurring, they will continue to grow during the lifetime of your database servers. The longer you operate these servers, the higher the operational expenses (OPEX) will be.

This means that the longer you use your database servers, the share between CAPEX and OPEX will shift towards a higher share of OPEX. The one time purchase of hardware may be considered a high cost upfront, but given that you will probably use the hardware for more than three years, it justifies the upfront cost.

For cloud hosting, the calculation will be similar. However, since you don't have hardware to purchase upfront, the CAPEX will be a lot lower. As cloud hosting has a recurring monthly cost, the OPEX will be higher. In some cases, your cloud provider may calculate some (setup) cost upfront and this should be treated as CAPEX.

Example Calculation for Hardware

In this example we will make a calculation of a small company (under 100 employees) that hosts on hardware in their own racks in a data center. This company has two dedicated sysops and one experienced DBA (1-4 years), where the DBA is managing around 20 databases and the sysops around 200 hosts. The average DBA salary for this is [\\$65,000](#), so the annual cost per database would be \$3,250. The sysops average around \$50,000 for the same experience, and cost \$250 per host per year. The sysops are also the people who manage the datacenter. We will not factor in the facilitation costs as this would get over complicated.

For our example cluster, we will make use of a three node MySQL replication setup: one master and two slave nodes. Hardware is based upon the Dell R730 with 64GB of memory and six 400GB SSDs, as this is a very popular model for this purpose. The price of a R730 with this configuration is currently \$7655.

Rental cost of a full rack is nowadays around \$350[1], so the colocation cost per U is roughly \$8 per month. Since the R730 is a 2U unit, the total cost for our databases would be \$48 per month.

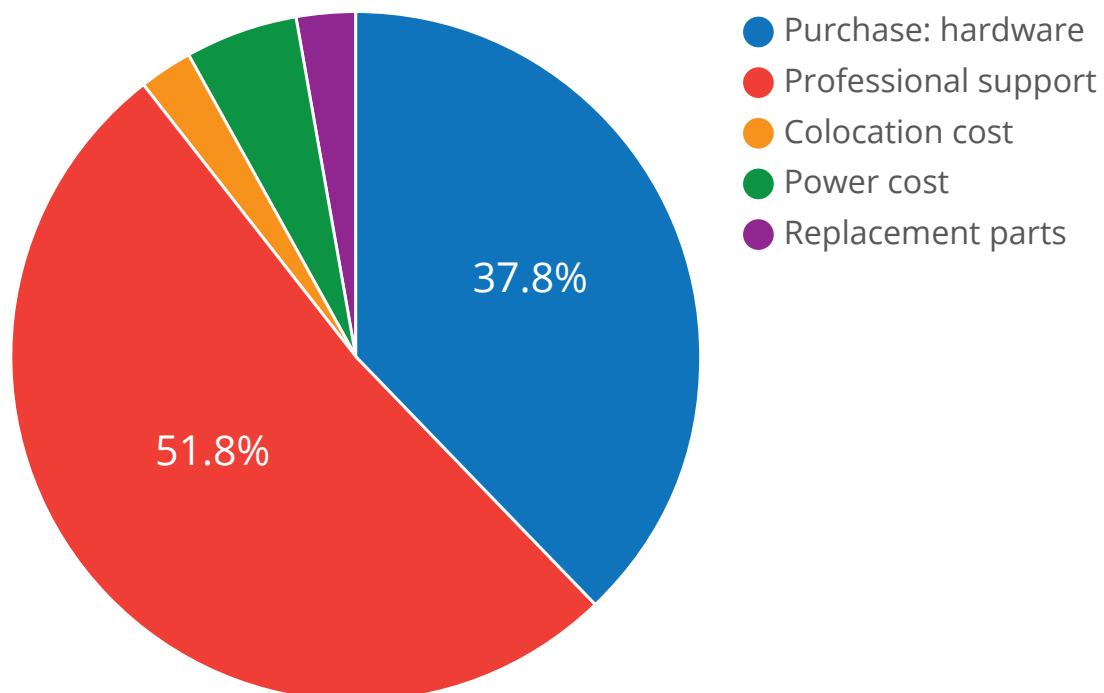
Modern colocation costs factor out the power consumption, as the power is a variable factor. Prices for power with colocation can vary a lot, but it currently averages around \$0.20 per kWh. The average database server consumes around 200 watts, which results in a 144kWh consumption per month per server. For our three database servers this would result in \$86 per month.

This results in the following TCO:

Cost item	CAPEX	OPEX (per year)	TCO (3 years)
Purchase: hardware	\$22,965		
Professional support (DBA / Sysop)		\$10,500	
Colocation cost		\$576	
Power cost (200W)		\$1,032	
Replacement parts	\$1,500		
Total	\$24,465	\$12,108	\$60,789

There are a couple of conclusions we can draw from this calculation. Cost for colocation, power and replacement parts are neglectable, compared to the other cost factors. Also during the lifetime of a database server, the support costs make up more than half of the total costs. And are far higher than the original purchase price of the servers.

TCO (3years)



Example Calculation Cloud Hosting

In this example we will make a calculation for a company that hosts in the cloud. To compare fairly, we will again make use of a three node MySQL replication setup on EC2. Amazon provides a nice [TCO calculator](#) for these purposes, so we made use of this as input for the calculations below.

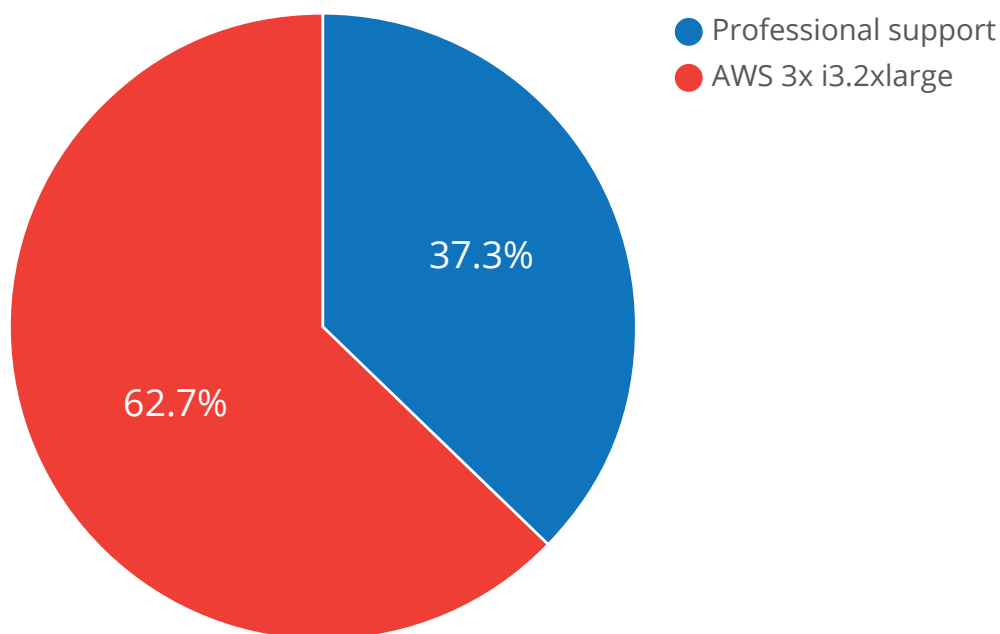
To make the database servers comparable, we chose the i3.2xlarge, which (currently) has 8 vCPUs, 61GB of RAM and 1900GB of SSD storage. This currently costs \$0.624 per hour, which is slightly below \$15 per day and \$5466 per year.

In the cloud the upfront investments (CAPEX) are not necessary. This is true in many cases, except if you make use of reserved instances like in AWS. With reserved instances, you make a claim on Amazon to reserve (performant) capacity for you, that you can use at will. In our calculation, we will not make use of reserved instances. Next to the lower CAPEX, our OPEX should be lower since our sysops don't have to go to the data center or install these servers.

This results in the following TCO:

Cost item	CAPEX	OPEX (per year)	TCO (3 years)
Professional support (DBA only)		\$9,750	
AWS 3x i3.2xlarge		\$16,399	
Total	\$0	\$27,149	\$78,447

TCO (3years)



Even though we have eliminated our upfront costs and capital investments (CAPEX), the OPEX is really high due to the premium we have to pay for high performance instances in AWS. Over a three year period, your TCO will be higher than having your own hardware.

OPEX Has a Large Influence

As you can see from these calculations, the influence of the operational costs (OPEX) during the lifetime of the servers is far greater than the initial large investment of the CAPEX. This is mostly due to running (and owning) these servers for multiple years.

In the case of owning your own hardware, we have shown that the operational costs even outweigh the initial costs for purchasing these servers. For the AWS example, the total costs of "owning" these servers is even higher than for the hardware example. This is the premium paid for flexibility, as with a cloud environment you are free to upgrade to a newer instance every year.

For both examples it is clear that the professional support for running these databases is relatively high. It looks like the sysops are clearly far more efficient when they are managing more than 200 hosts, but they don't have to bother with the additional tasks that the DBA is supposed to do. If you could only make the DBA more efficient.

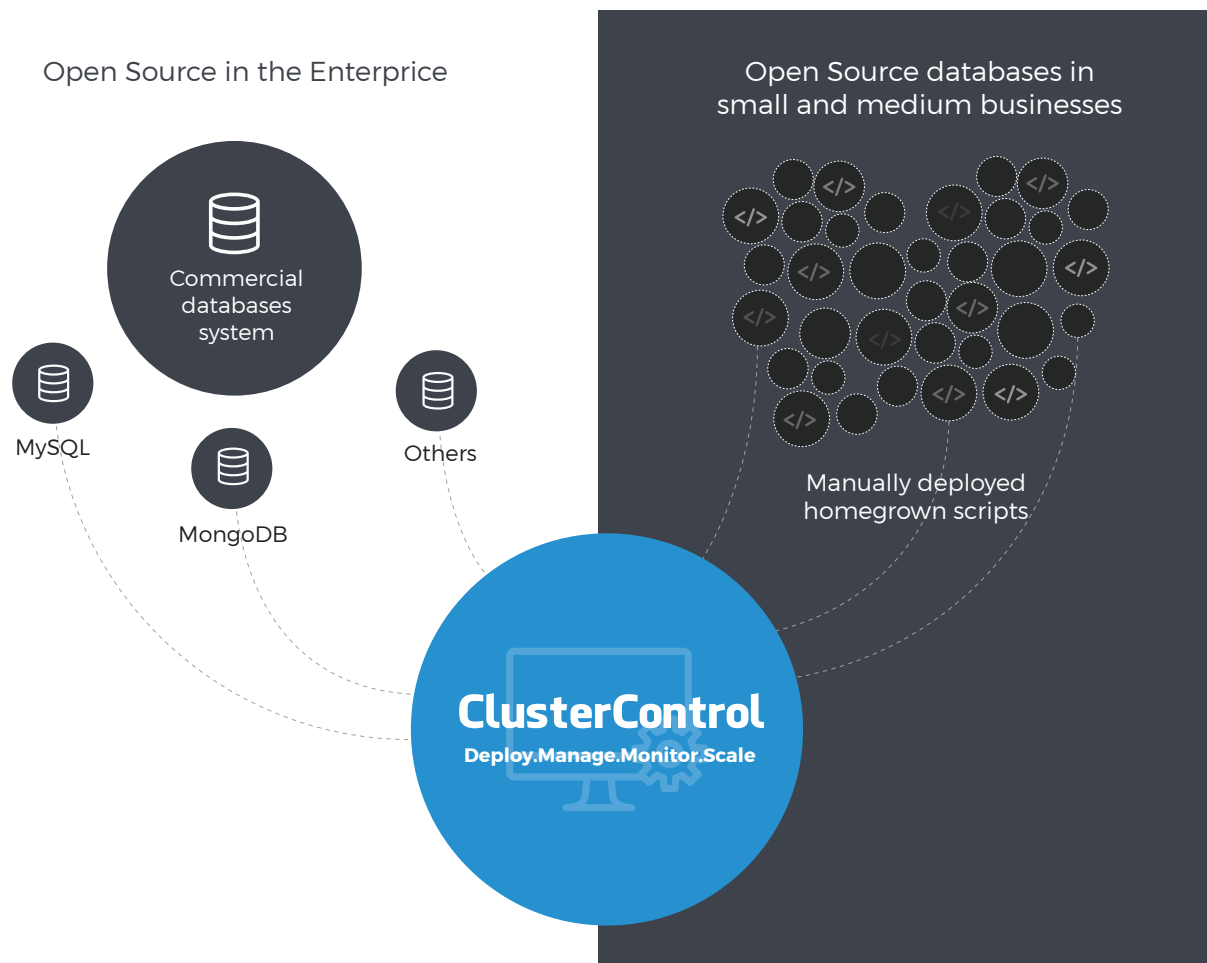
ClusterControl presents value to our operations team in several different ways, one of which is performance analysis. When first implementing Galera, we found a number of performance bottlenecks which we quickly found and resolved as a direct result of ClusterControl. One of those being 'Health Report' which provides insight into cache hit ratios, percentage of max connections, open file limits, table lock contention along with other valuable metrics.

Michael Ward, Site Reliability Manager at Ping Identity

Our database is mission critical for our business, it is where we store masked and encrypted credit card data, transaction data, merchant and end customer information. With the amount of transactions flowing through our systems, we cannot afford any downtime, performance problems or security glitches. Our databases are fully replicated to a separate DR site. Having a management tool like ClusterControl has helped us achieve our goals.

Idris Khanafi, Head of Infrastructure at Veritrans

Where ClusterControl Fits



So Where Does ClusterControl Fit in the Open Source World?

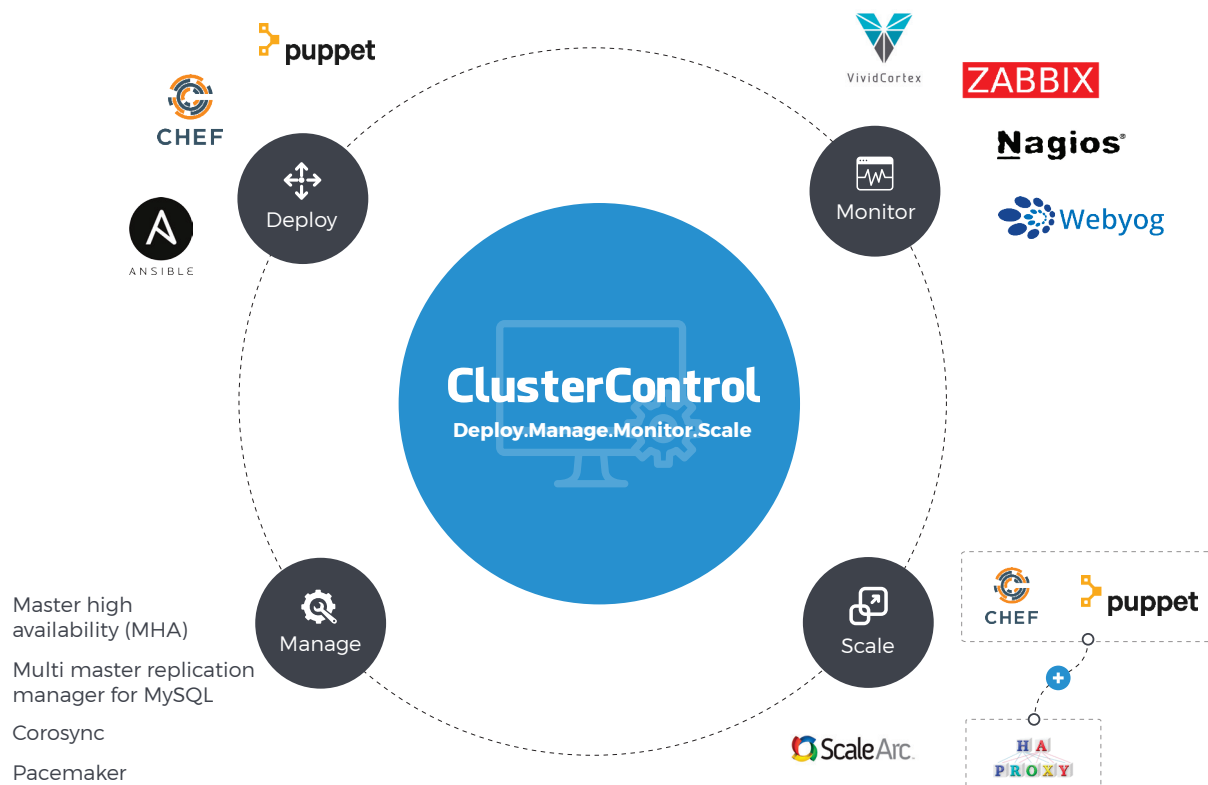
ClusterControl's mission is to enable its users to easily deploy, manage, monitor and scale open source databases.

Most enterprises will run Oracle or SQL Server, or a combination of both, with small pockets of isolated open source databases operating independently. Small to medium sized businesses would tend to deploy mainly open source databases, especially for new applications.

Most of these open source databases have been deployed on a shoestring budget. Deployment would often be done manually, with homegrown scripts to operate the environment. A variety of tools will be used to achieve different aspects of the system... some free, some on monthly payments, maybe even with a yearly license cost. The tools used are brought in by the Developers, DevOps, SysAdmins, or DBAs managing the projects, as selection of these tends to be largely based on personal preference. Over time these tools and homegrown solutions become "legacy" as contractors, agencies and employees turnover and the new ones come on board and continue the practices that were introduced before them.

4.1. How Many Tools Would it Take to Do What ClusterControl Does?

So what's the alternative to home-made scripts and cobbled-together solutions?



Deployment

ClusterControl takes care of the deployment of open source databases (in many flavors including MariaDB and Percona) through its point-and-click interface or through its command line. Those deploying databases without ClusterControl tend to rely on automation tools like Ansible, Chef or Puppet.

Each of these tools, including ClusterControl, offer free versions as well as enterprise options with support. It is fair to say that most deployments made using these tools are on the free versions, however, as we discussed in the previous section of this document, you often pay whether the solution is free using homegrown scripts on a free deployment tool or using the tool's enterprise version which automates many of these features.

For the sake of this comparison we will compare the basic paid versions of the tools and assume three nodes.

Costs	PUPPET Standard	CHEF Automate	ANSIBLE Standard
Cost Per Node	\$120	\$137	n/a
Minimum Node Requirement	11	50	Up to 100
Max Number of Tickets per month	5	unlimited	unlimited
First Year Cost (min)	\$1,320	\$6,850	\$10,000

Monitoring

ClusterControl provides a unified view of all database environments across your data centers and lets you drill down into individual nodes. Without ClusterControl most people would use a third party monitoring tool. Some of these monitoring tools focus on the database while others offer both database and server-level monitoring.

For the sake of this comparison we will also compare the basic paid versions of the tools and assume three nodes.

Costs	VividCortex Standard	Monyog Professional	Nagios XI Standard
Cost Per Node	\$2,988 per Instance	\$139	n/a
Minimum Node Requirement	n/a	n/a	Up to 100
Max Number of Tickets per month	Unlimited, chat & email only	\$60 Premium Support	5 Call Pack \$995
First Year Cost (min)	\$2,988	\$597	\$2,990

Management

ClusterControl offers a wide array of integrated management tools. Features like configuration management, backups and restores, patching, upgrades, automated failover and recovery and more make it a full-featured database management system. Because of the wide range of management options it is impossible to compare it to another tool, as there just aren't any on the market.

What is available to use are a variety of open-source, free to use, script-based tools that focus on a particular management function like backup or failover.

While none of these tools do the full-scope of management functions that ClusterControl does, each offer the ability to manage a particular aspect of the database lifecycle.

Tool Name	Function	Description
Master High Availability (MHA)	Failover	MHA performs automating master failover and slave promotion with minimal downtime, usually within 10-30 seconds.
Multi-Master Replication Manager	Failover	MMM (Multi-Master Replication Manager for MySQL) is a set of flexible scripts to perform monitoring/failover and management of MySQL master-master replication configurations (with only one node writable at any time).
Corosync	Cluster Mgmt	The Corosync Cluster Engine is an open source project derived from the OpenAIS project and licensed under the new BSD License. The mission of the Corosync effort is to develop, release, and support a community-defined, open source cluster.
Pacemaker	Failover	Pacemaker achieves maximum availability for your cluster services by detecting and recovering from node and service-level failures. It achieves this by utilizing the messaging and membership capabilities provided by preferred cluster infrastructure Corosync.

Scaling

At the heart of ClusterControl is its features and functions around reducing downtime, providing high availability, and scaling. Scaling in ClusterControl can both be performed manually and also automated. ClusterControl also helps you plan for capacity by letting you understand historical as well as current performance levels, so you can plan for future scaling and upgrades.

ClusterControl lets you deploy the very best in open-source load balancing technology like Maxscale, ProxySQL, HAProxy, and Keepalived. While these technologies are free, and can be deployed manually using script-based installations, ClusterControl automates the process with its point-and-click interface. ClusterControl also lets you deploy caching technologies as well.

Other companies like ScaleArc offer caching and load balancing for your MySQL databases through a GUI but do not offer the built-in automation options of ClusterControl. ScaleArc does not publish its pricing so a direct comparison is not possible.

We needed expert advice and help to optimise our database operations at a time when customers are asking for better streaming and content delivery services. Severalnines was the best fit and a lot of good reviews online gave us the confidence to work with them. Led by their CEO, Vinay Joosery, the Severalnines team offered excellent, personalised support and gave us practical advice on how to enhance our systems. Thanks to Severalnines, we can spend more time working with our customers to deliver the next generation of content services as our back end is completely protected.

Laurent Pythoud, IT Web Manager at European Broadcasting Union (EBU)

About ClusterControl

ClusterControl is the all-inclusive open source database management system for users with mixed environments that removes the need for multiple management tools. ClusterControl provides advanced deployment, management, monitoring, and scaling functionality to get your MySQL, MongoDB, and PostgreSQL databases up-and-running using proven methodologies that you can depend on to work. At the core of ClusterControl is its automation functionality that lets you automate many of the database tasks you have to perform regularly like deploying new databases, adding and scaling new nodes, running backups and upgrades, and more.

About Severalnines

Severalnines provides automation and management software for database clusters. We help companies deploy their databases in any environment, and manage all operational aspects to achieve high-scale availability.

Severalnines' products are used by developers and administrators of all skills levels to provide the full 'deploy, manage, monitor, scale' database cycle, thus freeing them from the complexity and learning curves that are typically associated with highly available database clusters. Severalnines is often called the "anti-startup" as it is entirely self-funded by its founders. The company has enabled over 12,000 deployments to date via its popular product ClusterControl. Currently counting BT, Orange, Cisco, CNRS, Technicolor, AVG, Ping Identity and Paytrail as customers. Severalnines is a private company headquartered in Stockholm, Sweden with offices in Singapore, Japan and the United States. To see who is using Severalnines today visit:

<https://www.severalnines.com/company>



Deploy



Manage



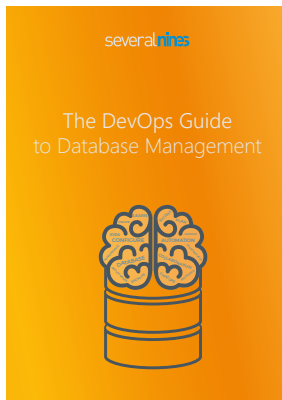
Monitor



Scale

Related Resources from Severalnines

Whitepapers



The DevOps Guide to Database Management

Relational databases are not very flexible by nature, while DevOps is all about flexibility. This creates many challenges that need to be overcome. This white paper discusses three core challenges faced by DevOps when it comes to managing databases. It also discusses how Severalnines ClusterControl can be used to address these challenges.

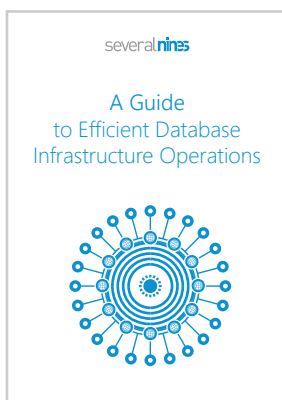
[Download here](#)



Management and Automation of Open Source Databases

Proprietary databases have been around for decades with a rich third party ecosystem of management tools. But what about open source databases? This whitepaper discusses the various aspects of open source database automation and management as well as the tools available to efficiently run them.

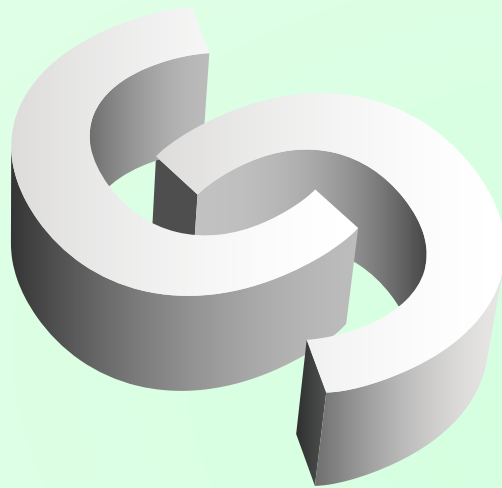
[Download here](#)



A Guide to Efficient Database Infrastructure Operations

Taking control of their data is every company's number one job. Database operations encompass a number of functions, including the initial deployment of a solution, configuration management, performance monitoring, SLA management, backups, patches, version upgrades and scaling. In this white paper, we will discuss the operational aspects of running database infrastructures, and how companies can make these more efficient.

[Download here](#)



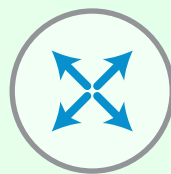
Deploy



Manage



Monitor



Scale