

severalnines

ClusterControl The Guide



Table of Contents

Introduction	4
What Makes ClusterControl Different	5
Comprehensive Management of Open Source DBs	5
The Need for Cluster Management Software	6
Not Just Monitoring	6
All-in-one Management Software	8
ClusterControl Architecture	9
Functionality	13
Backup Management	13
Schedule, Manage and Operate Backups	14
Define Backup Policies, Retention, History	14
Upload to the Cloud	15
Backup Encryption	16
Automatic Backup Verification	16
Monitoring and Alerting	18
Integrations	19
Agent Based Monitoring	21
Deployment and Scaling	23
Deployment on Cloud	24
Load Balancers	25
Integration with Configuration Management Systems via the CLI	26
Patches and Upgrades	27
Operational Reporting on Version Upgrades	27
Security and Compliance	29
Securing ClusterControl Traffic	29
ClusterControl Secure Backups	30
Encryption of Data in Transit (SSL)	31
Key Management	32
Security Advisors	32
User Roles	33
Audit Log for MySQL	34
Database Infrastructure Audit	34
Operational Reports	36
Configuration Management	38
Automatic Recovery & Repair	40
Make Sure the Master Is Really Dead Before You Failover	40
Failover Only Once	41
Do not Failover to an Inconsistent Slave	41
Only Write to the Master	41
Do not Automatically Recover the Failed Master	41
Performance Management	43
Automatic Performance Advisors	46
Custom Advisors	47
Developer Studio	48



Table of Contents

Command Line Interface (CLI)	49
Deployment	50
Monitoring	51
Scaling	52
Management	52
Conclusion	54
About Severalnines	55
Related Resources	56

Introduction

ClusterControl is the only management system you'll ever need to take control of your open source database infrastructure.

It's a bold statement, but one we feel we have earned the right to make, based on many years of product development as well as customer deployments and feedback.

Since the inception of Severalnines, we have made it our mission to provide market-leading solutions to help organisations achieve optimal efficiency and availability of their open source database infrastructures.

With ClusterControl, as it stands today, we are proud to say: mission accomplished.

Our flagship product is an integrated deployment, monitoring, and management automation system for open source databases which provides holistic, real-time control of your database operations in an easy and intuitive experience, incorporating the best practices learned from thousands of customer deployments in a comprehensive system that helps you manage your databases safely and reliably.

Deploying, monitoring and managing highly available open source database clusters is not a small feat and requires either just as highly specialised database administration (DBA) skills ... or professional tools and systems that non-DBA users can wield in order to build and maintain such systems, though these typically come with an equally high learning curve.

The idea and concept for ClusterControl was born out of that conundrum that most organisations face when it comes to running highly available database environments.

It is the only solution on the market today that provides that intuitive, easy to use system with the full set of tools required to manage such complex database environments end-to-end, whether one is a DBA or not.

The aim of this Guide is to make the case for comprehensive open source database management and the need for cluster management software. And explains in a just as comprehensive fashion why ClusterControl is the only management system you will ever need to run highly available open source database infrastructures.

What Makes ClusterControl Different

Most organizations have databases to manage, and experience the headaches that come with that: managing performance, monitoring uptime, automatically recovering from failures, scaling, backups, security and disaster recovery. Organizations build and buy numerous tools and utilities for that purpose.

However, it can be quite an undertaking to cobble together complex 'stacks' of siloed tools and utilities in order to operate a database environment. It is an expensive, antiquated way of managing databases, especially in an age of automation and on-demand computing. It also requires experts in databases, as well as significant expertise in the tools/utilities to manage them.

ClusterControl differs from the usual approach of trying to bolt together performance monitoring, automatic failover and backup management tools by combining – in one product – everything you need to deploy and operate mission-critical databases in production. It provides an integrated deployment, monitoring, and management automation platform for open source databases. It gives a holistic real-time view of database health across the enterprise in a simple and integrated user experience.

There are a few aspects that make it unique, as compared to existing monitoring products for open source databases.

Comprehensive Management of Open Source Databases



MySQL, PostgreSQL and MongoDB are the top 3 open source databases in use today, according to DB-Engines¹. It is not uncommon to find all of them deployed in an organisation, as they often are used to tackle different problems. MySQL also comes in different variants, including Oracle MySQL, Percona Server and MariaDB, as well as different replication/clustering solutions like MySQL (GTID) replication, MariaDB (GTID) replication, Galera Cluster, NDB Cluster and InnoDB Cluster/Group Replication.

Using multiple types of databases comes at a cost in complexity. Each database introduces a new technology to be learned and managed. Adopting a new database in the organization means a number of things for ops teams - from deployment, configuration, performance monitoring and trending, SLA management to backups, failure and recovery management, security management, version upgrades and scaling. System administrators and devops teams are increasingly being asked to manage databases, especially in smaller businesses. But while it may be relatively straightforward to set up a new database, database administration is a distinct role and skill-set. Without the appropriate tools, ops teams will struggle to manage multiple databases

¹ <https://db-engines.com/en/>

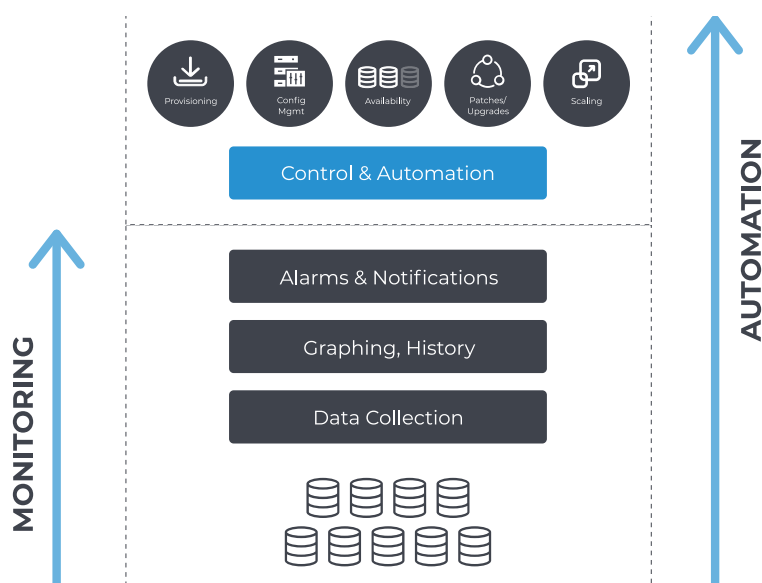
and database topologies. And if the data is important, management of databases is an area that ops teams cannot afford to cut corners with.

The Need for Cluster Management Software

Databases are increasingly being deployed in distributed setups, or clusters, as businesses require high availability of data. Modern environments such as the cloud, Docker and Kubernetes are also having a major impact on the way that companies deploy and manage their databases. These drive the need for cluster management software.

ClusterControl was built from the ground up to manage highly available distributed databases, and it provides a system view of the entire cluster as opposed to individual nodes. The key is its contextual knowledge of the underlying topology and availability model of the database, whether it be a multi-master replication setup or a MariaDB Galera Cluster. Recovery procedures and anomaly detection algorithms are battle tested across thousands of real world database deployments. Seamless integration with database proxies and load balancers like HAProxy, ProxySQL and MaxScale ensure that applications always connect to a healthy node and are able to function properly.

Not Just Monitoring



ClusterControl provides real-time monitoring of the entire database infrastructure. Performance advisors provide specific advice on how to address database and server issues, such as performance, security, log management, configuration and capacity planning. Operational reports can be used to ensure compliance across hundreds of instances.

However, monitoring is not management. ClusterControl has features like backup management, automated recovery/failover, deployment/scaling, rolling upgrades, security/encryption, load balancer management, and so on. This enables the ops team to have one single platform to automate different management tasks, as opposed to having to cobble together homegrown tools for backups, failover, upgrades, restore verification, security management, etc.

Management Tasks

1# Backup/Restore Management

Everything from scheduling backups, restoring data, managing retention policies, encryption/compression, restore verification and data archiving to the cloud.

2# Failover/Recovery and Switchover Management

Detecting server or data center failures and switching over to available healthy nodes - e.g. by promoting a new master and reslaving the other servers in the setup. Recovering failed nodes, e.g. rebuilding a corrupted slave from scratch or re-bootstrapping an entire cluster that crashed beyond repair. Ensuring load balancers are pointing to the right database instances, especially after topology changes, so applications can still access data. Shifting traffic to healthy instances while doing infrastructure maintenance, for instance when applying OS patches.

3# Security Management

Get an overview of which users are defined in the databases, what grants they have, when they last logged in. Identify inactive users, or users with too liberal permissions. Configure for security. Encrypt data at rest and in transit. Audit database access. LDAP and Role-Based Access Control for management staff. Compliance reporting and analytics. Create and manage SSL keys for encryption.

4# Configuration Changes

Perform changes of variables in a safe way from a single place. Config changes can be set on single or multiple servers at once, with the changes persisted to the configuration files on disk. User is informed if a restart of the node is needed, so the user can trigger a rolling restart. Be able to audit configuration changes to understand who did what.

5# Upgrades

Upgrades can be time consuming, and they usually require several steps - disabling monitoring on the node being upgraded, running the upgrade commands, rolling restart with switchover at the load balancer level.

Track upgrade times so as to know how long each step takes to complete.

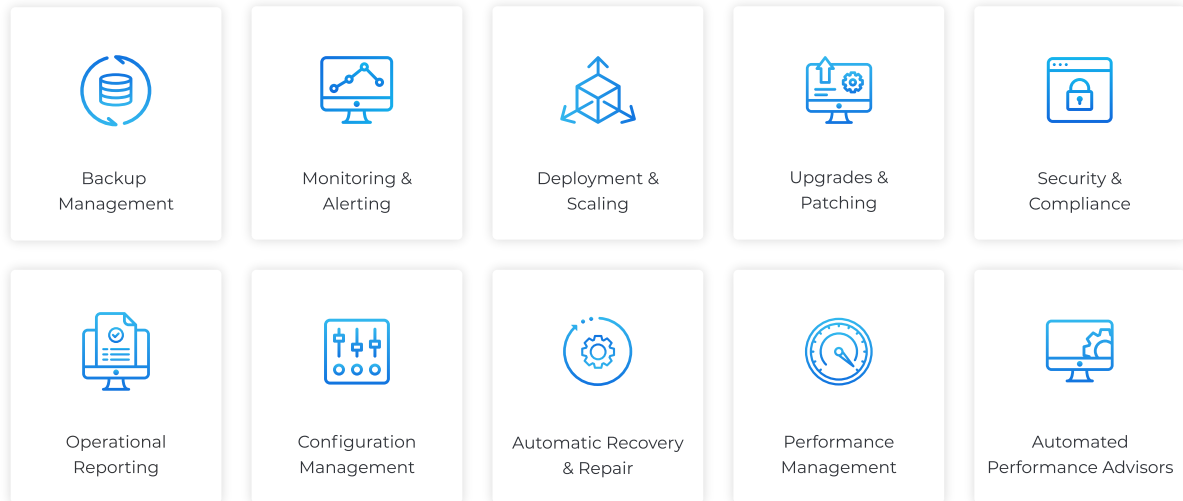
6# Management of jobs and external scripts

Managing a database usually means a number of small maintenance tasks that are executed regularly. A job scheduler can be used to keep track of these, and alert in case of issues.

All-in-one Management Software

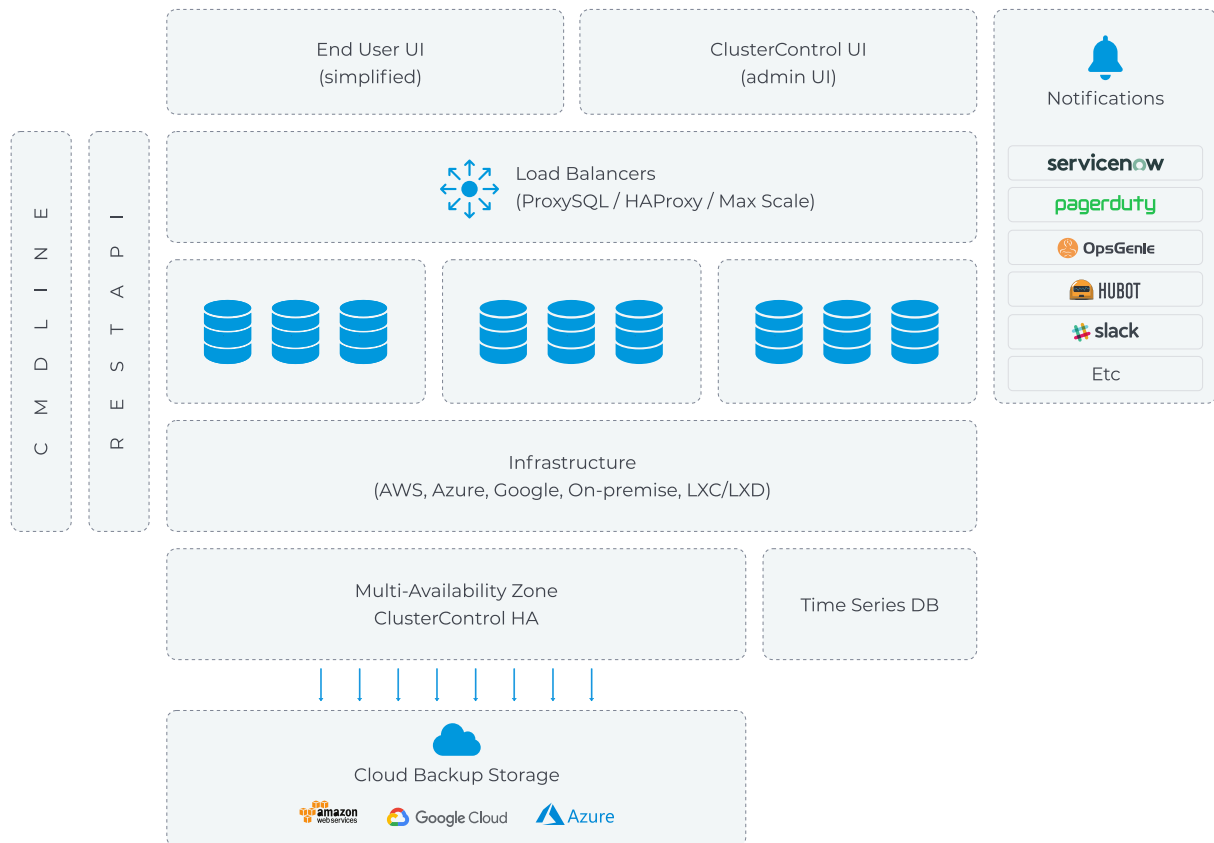
ClusterControl differs from the usual approach of trying to bolt together performance monitoring, automatic failover and backup management tools by combining – in one product – everything you need to deploy and operate mission-critical databases in production.

It is able to automate the entire database environment, and ultimately deliver an agile, modern and highly available data platform based on open source.



ClusterControl Architecture

ClusterControl is a standalone software product that can be installed on-prem, behind a firewall and even completely without internet access, or in the cloud. It consists of a controller, a local datastore for configuration and monitoring data, some access components to enable access via a Web UI and CLI, and some additional ones for integration with cloud providers and notification services.



ClusterControl consists of a number of components.

Component	Package naming	Role
ClusterControl Controller (cmon)	clustercontrol-controller	The brain of ClusterControl. A backend service performing automation, management, monitoring and scheduling tasks. All the collected data will be stored directly inside CMON database.
ClusterControl REST API [1]	clustercontrol-cmonapi	Interprets request and response data between ClusterControl UI and CMON database.
ClusterControl UI	clustercontrol	A modern web user interface to visualize and manage the cluster. It interacts with CMON controller via remote procedure call (RPC) or REST API interface.

Component	Package naming	Role
ClusterControl SSH	clustercontrol-ssh	Optional package introduced in ClusterControl 1.4.2 for ClusterControl's web SSH console. Only works with Apache 2.4+.
ClusterControl Notifications	clustercontrol-notifications	Optional package introduced in ClusterControl 1.4.2 providing a service and user interface for notification services and integration with third party tools.
ClusterControl Cloud	clustercontrol-cloud	Optional package introduced in ClusterControl 1.5 providing a service and user interface for integration with cloud providers.
ClusterControl Cloud File Manager	clustercontrol-clud	Optional package introduced in ClusterControl 1.5 providing a command-line interface to interact with storage objects on cloud.
ClusterControl CLI	s9s-tools	Open-source command line tool to manage and monitor clusters provisioned by ClusterControl.

ClusterControl performs its monitoring, alerting and trending duties by using the following methods:

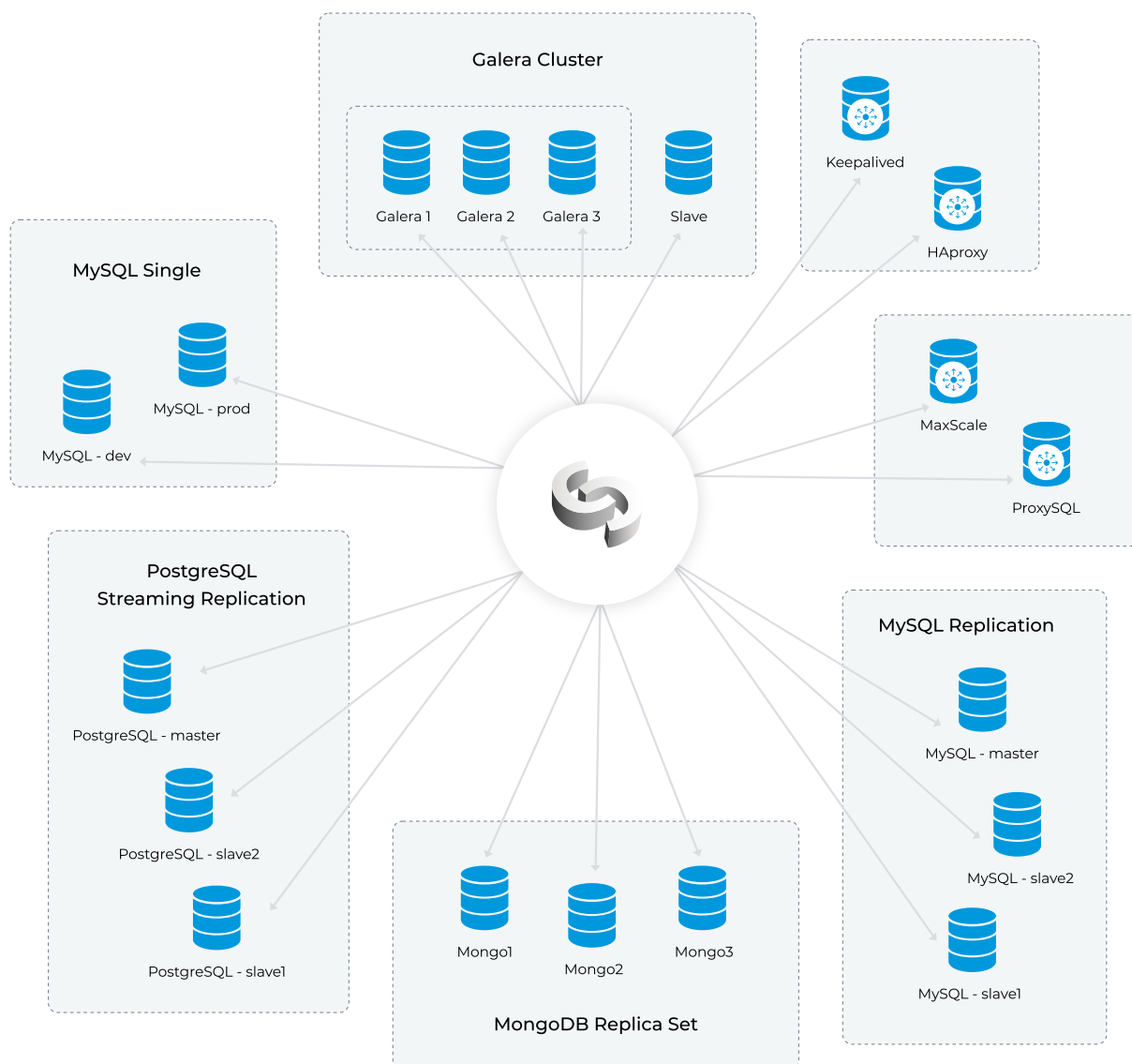
- SSH - Host metrics collection (process, load balancers stats, resource usage and consumption, etc.) using SSH library.
- Database client - Database metrics collection (status, queries, variables, usage etc) using the respective database client library.
- Advisor - Mini programs written using ClusterControl Domain Specific Language (DSL) and running within ClusterControl itself, for monitoring, tuning and alerting purposes. The DSL syntax is based on JavaScript, with extensions to provide access to ClusterControl internal data structures and functions. The DSL allows you to execute SQL statements, run shell commands/programs across all your cluster hosts, and retrieve results to be processed for advisors/alerts or any other actions.
- Agent based - uses Prometheus exporters to capture metrics data from the system it monitors. Statistics are then stored in the time series DB.

The controller connects to the managed database nodes via SSH in order to perform management procedures, e.g. recovering a broken database node or performing a rolling upgrade.

The ClusterControl installation process is pretty straightforward, consisting of ClusterControl binaries installation and key copy. There is no need to install additional agents software.

The supported platforms are RedHat/CentOS 6.x/7.x, Ubuntu 12.04/14.04/16.04 LTS, and Debian 7.x/8.x. The minimal OS resource requirements are 2GB of RAM, 2CPU and 20GB

disk space running on x86 architecture. ClusterControl itself can run on regular VMs or barebone hosts running on-prem, behind a firewall, or on Cloud VMs.



Additionally, ClusterControl requires ports used by the following services to be opened/enabled:

- ICMP (echo reply/request)
- SSH (default is 22)
- HTTP (default is 80)
- HTTPS (default is 443)
- MySQL (default is 3306) (internal database)
- CMON RPC (default is 9500)
- CMON RPC TLS (default is 9501)
- CMON Events (default is 9510)
- CMON SSH (default is 9511)
- CMON Cloud (default is 9518)
- Streaming port for backups through netcat (default is 9999)

The easiest and most convenient way to install ClusterControl is to use the installation script provided by Severalnines. Simply download the script and execute as the root user or user with sudo root permission.

```
1 | $ wget http://www.severalnines.com/downloads/cmon/install-cc
2 | $ chmod +x install-cc
3 | $ ./install-cc # as root or sudo user
```

Next step is to generate an SSH key which we will use to set up the passwordless SSH later on. If you have a key pair which you would like to use, you can skip creation of a new one.

```
1 | $ wget http://www.severalnines.com/downloads/cmon/install-cc
2 | $ chmod +x install-cc
3 | $ ./install-cc # as root or sudo user
```

Set up passwordless SSH to all nodes that you would like to monitor/manage via ClusterControl. In this case, we will set this up on all nodes in the stack (including ClusterControl node itself). On ClusterControl node, run the following commands to copy ssh keys and specify the root password when prompted:

```
1 | ssh-copy-id root@clustercontrolhost # clustercontrol
2 | ssh-copy-id root@dbhost1 #your database host
3 | ...
```

When the installation is completed you should be able to login to the ClusterControl web interface via: https://<your_vm_name>/clustercontrol/#.

Functionality



Backup Management

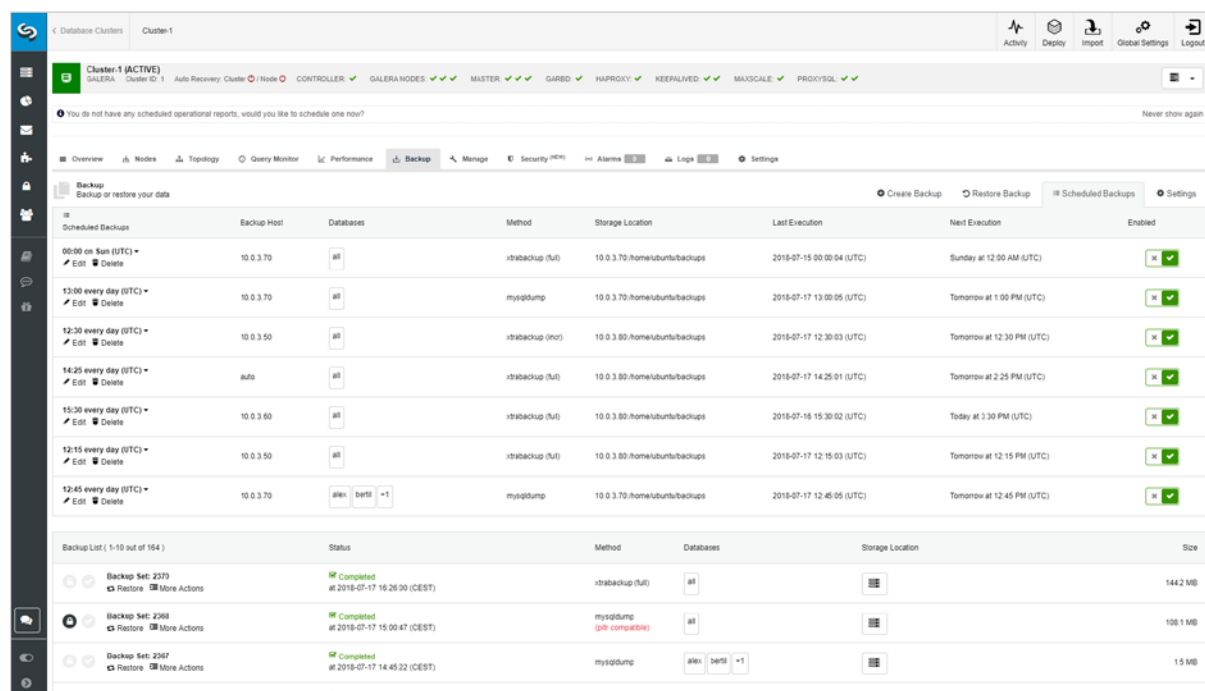


When it comes to backups and data archiving, IT departments are under pressure to meet stricter service level agreements, deliver more custom reports, and adhere to expanding compliance requirements while continuing to manage daily archive and backup tasks.

Backup List (1-10 out of 114)	Status	Method	Databases	Storage Location	Size
Backup Set: 2370 Restore More Actions	Completed at 2018-01-17 16:26:00 (CEST)	strabackup (full)	all		144.2 MB
Backup Set: 2368 Restore More Actions	Completed at 2018-01-17 15:00:47 (CEST)	mysqldump (not compatible)	all		108.1 MB
Backup Set: 2367 Restore More Actions	Completed at 2018-01-17 14:45:22 (CEST)	mysqldump	alex betti +1		1.5 MB
Backup Set: 2365 Restore More Actions	Completed at 2018-01-17 14:19:00 (CEST)	strabackup (full)	all		487.7 MB
Backup Set: 2364 Restore More Actions	Completed at 2018-01-16 17:31:00 (CEST)	strabackup (full)	all		151.2 MB
Backup Set: 2363 Restore More Actions	Completed at 2018-01-16 16:26:11 (CEST)	strabackup (full)	all		144.2 MB
Backup Set: 2361 Restore More Actions	Completed at 2018-01-16 15:00:39 (CEST)	mysqldump (not compatible)	all		108.1 MB
Backup Set: 2360 Restore More Actions	Completed at 2018-01-16 14:45:13 (CEST)	mysqldump	alex betti +1		1.5 MB
Backup Set: 2358 Restore More Actions	Completed at 2018-01-16 14:17:52 (CEST)	strabackup (full)	all		487.7 MB
Backup Set: 2357 Restore More Actions	Completed at 2018-01-16 17:30:52 (CEST)	strabackup (full)	all		151.2 MB

ClusterControl backup list

ClusterControl can help operations teams automate regular backup management tasks, such as creating the backup, maintaining backup retention, migrating backups, retaining backups in the cloud, restore, or deletion—all based on data retention policies. Real-time alerts are issued whenever there's a discrepancy. Reports can be configured to run at specific times, and delivered to the teams and individuals who need them.



ClusterControl provides centralized backup management for MySQL, MariaDB, PostgreSQL and MongoDB databases, whether they are running on-premise or in the cloud. Every backup is assigned with a backup ID, and ClusterControl creates a directory under storage directory according to the chosen name pattern.

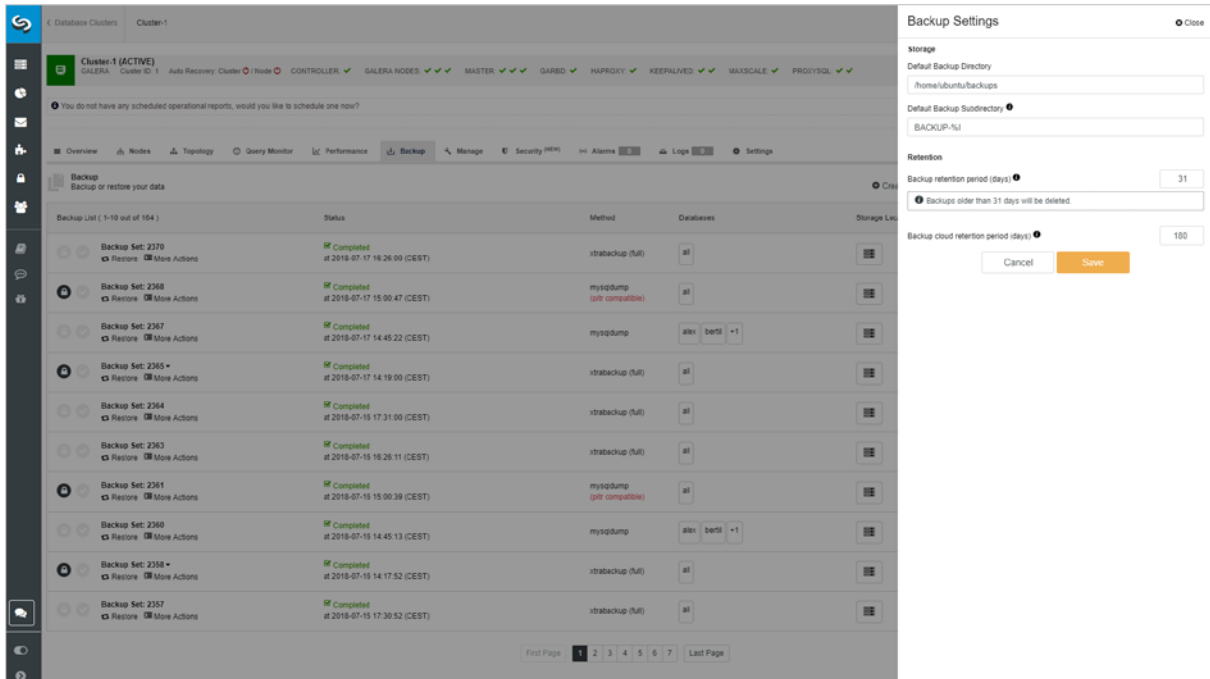
Schedule, Manage and Operate Backups

Keeping track of the backup schedules is crucial. Backing up data isn't exciting, but is absolutely necessary. Unfortunately, many of us only realize how crucial it is when something goes catastrophically wrong. There's nothing quite like a failed hard drive or ransomware infection to sharpen the mind and lead to a resolution to do things differently in the future.

A regular backup plan is especially important when handling sensitive data, for instance, other people's personal information. Following the implementation of the EU's new General Data Protection Regulation (GDPR), anyone who does business in the EU (even if they aren't based in the EU themselves) has an obligation to protect any data that could be used to identify someone – whether its their name, email address, physical address, or even their IP address.

Define Backup Policies, Retention, History

An optimized backup environment delivers services that are aligned with the business requirements of the organization. It requires having both the capabilities and resources available to meet these service demands, as well as the policies and processes in place to apply these resources where appropriate. Automated backup retention can keep the required backups and make sure you don't overutilize resources.



Upload to the Cloud

Storing database backups off-site is critical for organizations to be disaster-ready. To reduce application downtime, backups must be accessible 24 x 7. ClusterControl simplifies the process to transfer backups in a secure way to Amazon AWS, Google Cloud or Microsoft Azure.

A good backup program will let you choose which data should be stored, how often, where, and whether it should be encrypted for extra security. You should also look out for a backup tool that makes recovering your data easy after a disaster.

ClusterControl supports a number of backup methods depending on the cluster type, as summarized in the following table:

Cluster Type	Supported Backup Method
MySQL (Replication, Galera, NDB Cluster, Group Replication, MariaDB)	<ul style="list-style-type: none"> mysqldump Percona Xtrabackup (full and incremental) MariaDB Backup (MariaDB only) NDB Backup (MySQL Cluster only)
MongoDB (Replica Set, Sharded Cluster)	<ul style="list-style-type: none"> mongodump mongodb-consistent-backup (Percona Server for MongoDB only)
PostgreSQL (Streaming Replication)	<ul style="list-style-type: none"> Pg_dumpall pg_basebackup

When scheduling backups with ClusterControl, each of the backup methods are configurable with a set of options on how the backup is to be executed. Different

database workloads and backup strategies would require support for different features, for example:

- Disk IOPS throttling
- Network throttling
- Backup Locks
- Encryption
- Compression
- Retention period
- Restore verification

ClusterControl will automatically set a number of backup options, following the best practice from the particular database vendor. For example, if the target database node has binary log enabled, it will append an additional flag, `--master-data` to include the binary log coordinates (file name and position) of the dumped server. If it's a Galera node and the backup method is `xtrabackup`, ClusterControl will append an additional flag, `--galera-info` which contains the local node state at the time of the backup.

Create Backup Close

Backup schedule for: 00:00 on Sun (UTC)

Backup
Backup Method: xtrabackup (full)

Backup Host: 10.0.3.8:3306 (Master)

Enable Partial Backup:

Storage Location: Store on Node

Storage Directory: /home/ubuntu/backups

Backup Subdirectory: BACKUP-%I

Upload Backup to the cloud:

Upload Backup to the Cloud
Enable to upload the backup to your preferred cloud provider.
Back Continue

Backup Encryption

Do you trust the backup storage? Can you control access to your backups during the entire backup lifecycle? Encryption might be a good idea to protect your data. ClusterControl maintains backup encryption for MySQL, MariaDB, MongoDB, and PostgreSQL databases. Backups encryption uses AES-256 CBC algorithm. You can find an auto-generated key in the cluster's configuration file under `/etc/cmon.d/cmon_X.cnf`. If the backup destination is not local, the backup files are copied in an encrypted format. This feature complements the offsite backup on the cloud, where we do not have full access to the underlying storage system.

ClusterControl not only creates encrypted backups, but it also maintains and recognizes encryption keys during the restore process. The entire workflow makes the backup encryption process easy to implement and manage.

Automatic Backup Verification

ClusterControl brings you peace of mind by automatically validating the integrity of your backup data. It will execute a restore of your backup data, and alert in case of problems. The entire process is automated, and you only need to provide a target host where the verification will take place.

The backup verification option can be enabled during the backup schedule process. When this option is enabled, you will be able to specify the hostname as well as a list

of settings like: install database software, disable firewall, disable SELinux, Apparmor. To minimize the cost of your IT infrastructure, you can also shut down the server after the backup verification is completed. The last option is to postpone the verification after x amount of hours, which can also help with better resource utilisation.

Create Backup Close

1 2 3 4

Backup Settings

- Use Compression
- Compression Level: 6 (System Default)
- Backup Locks
- Lock DDL per Table
- Xtrabackup Parallel Copy Threads: 1
- Network Streaming Throttle Rate (MB/s): 0
- Use PIGZ for parallel gzip
- Failover backup if node is down
- Verify Backup new
- Enable Encryption
- Retention: 31 days (Default) | Custom | Keep Forever

Create Backup Close

1 2 3 4

Verify Backup

Restore backup on

*** You need to press ENTER to add this host.**

- Install Database Software
- Disable Firewall?
- Disable SELinux/AppArmor?
- Shutdown the server after the backup have been restored
- Verify the backup after N hours after completion: (dropdown menu open showing 0-15)



Monitoring and Alerting



Most production databases today run in some kind of high availability setup - from simpler master-slave replication to multi-master clusters fronted by redundant load balancers. Operations teams deal with dozens, often hundreds of services that make up the database environment.

ClusterControl was originally designed to address modern, highly distributed database setups based on replication or clustering. It provides a systems view of all the components of a distributed cluster, including load balancers, and maintains a logical topology view of the cluster. This approach can be contrasted with traditional server monitoring tools where servers are monitored individually, without the context of the cluster they operate within. Monitoring a distributed cluster as a single entity, with the ability to drill down to the individual node, makes it easier and more effective for operational staff to manage the infrastructure.

Dashboard Name	Metric	Scale	Selected as Default Graph
Bytes Sent/Recv	BYTES_SENT,BYTES_RECEIVED	linear	No
Galera - Flow Ctrl	WSREP_LOCAL_CERT_FAILURES, WSREP_LOCAL_BF_ABORTS,WSREP_FLOW_CONTR	linear	No
Galera - InnoDBFlow	INNOOB_BUFFER_POOL_PAGES_DIRTY, INNOOB_BUFFER_POOL_PAGES_DATA, INNOOB_OG_LOG_FLUSHES, INNOOB_DATA_FLUSHES, WSREP_FLOW_CONTROL_SENT,WSREP_FLOW_CON	logarith...	No
Galera - Queues	WSREP_LOCAL_SEND_QUEUE, WSREP_LOCAL_SEND_QUEUE_AVG,WSREP_LOCAL...	linear	No
Galera - Replication	WSREP_REPLICATED,BYTES,WSREP_RECEIVED,BY	linear	No
Handler	HANDLER_COMMIT,HANDLER_DELETE,HANDLER_RE	linear	No
InnoDB - Disk I/O	INNOOB_LOG_WRITES,INNOOB_DATA_WRITES,INNO	linear	No
Query Performance	SLOW_QUERIES.SELECT_FULL_JOIN,SELECT_FULL...	linear	No

Note: You can rearrange dashboard order by drag and drop above

ClusterControl Dashboards Settings

Monitoring is also an area where operations teams commonly spend time developing custom solutions. For instance, Graphite and Cacti provide trending, Nagios provides alerting and Statsd and Collectd gather raw metrics. It is common to find IT teams integrating these systems in order to get a holistic view of their systems.

ClusterControl provides a complete monitoring system with real time data to know what is happening now, high resolution metrics for better accuracy, configurable dashboards, and a wide range of third-party notification services for alerting.

ClusterControl provides several predefined dashboards. You don't need to configure anything to use this feature, which provides a great overview of a cluster.

Dashboards can be customized with the types and metrics required. There are different dashboards for each cluster type, depending on the information that is relevant to the database vendor.

The same ClusterControl instance can monitor different types of database technologies/vendors:

- MySQL (standalone, MySQL Replication, NDB Cluster, InnoDB Cluster/Group Replication)
- MariaDB (standalone, Replication, Galera Cluster)
- Percona Server for MySQL (standalone, Percona XtraDB Cluster)
- PostgreSQL (standalone, Streaming Replication)
- MongoDB Inc. (standalone, ReplicaSet, Sharded Cluster)
- Percona Server for MongoDB (standalone, ReplicaSet, Sharded Cluster)

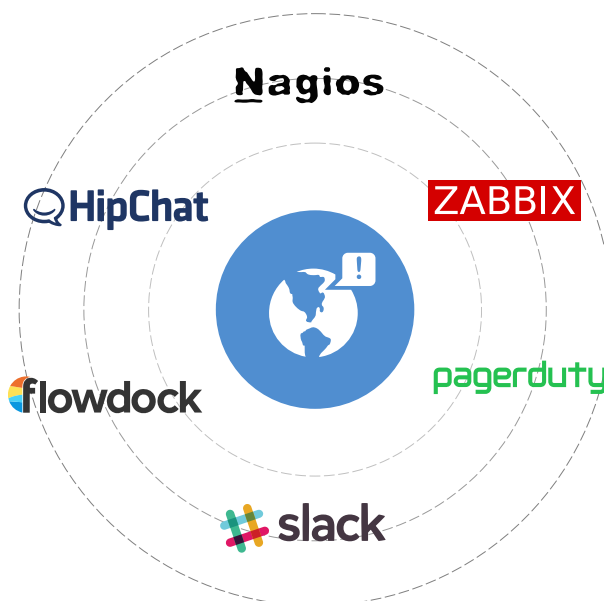
On-premises and cloud systems can be monitored and managed from the same ClusterControl instance. Intelligent health-checks are implemented for distributed topologies, for instance detection of network partitioning by leveraging the load balancer's view of the database nodes.

Monitoring can be agentless or agent-based. Agentless monitoring is done via SSH.

Proactive monitoring features allow the operations team to be alerted of trends or anomalies that might cause more severe problems in the future. For example, the database is consuming a lot of disk, and it is probable that the host will run out of space in a few days.

Integrations

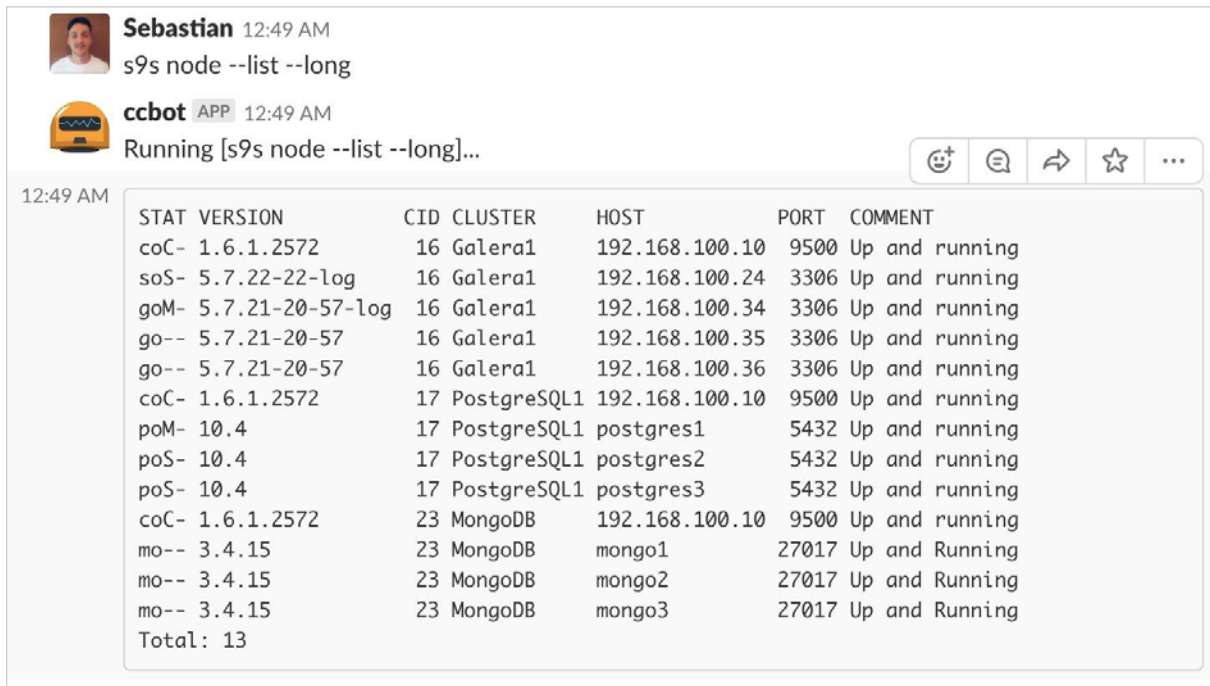
ClusterControl allows us to integrate our tools used daily in our organization, to facilitate tasks or centralize them in the same tool. For example, if an ops team uses Slack, all the team members can monitor and manage the databases from there. In the same way, we can access logs without connecting to the different hosts, which can save us considerable time.



Some of the integrations include:

- Enterprise monitoring: Nagios, Zabbix
- Incident Management: PagerDuty, VictorOps, OpsGenie
- ChatOps: Slack, Telegram
- IT Service Management: ServiceNow
- Syslog

Let's see an example integrating ClusterControl with Slack:

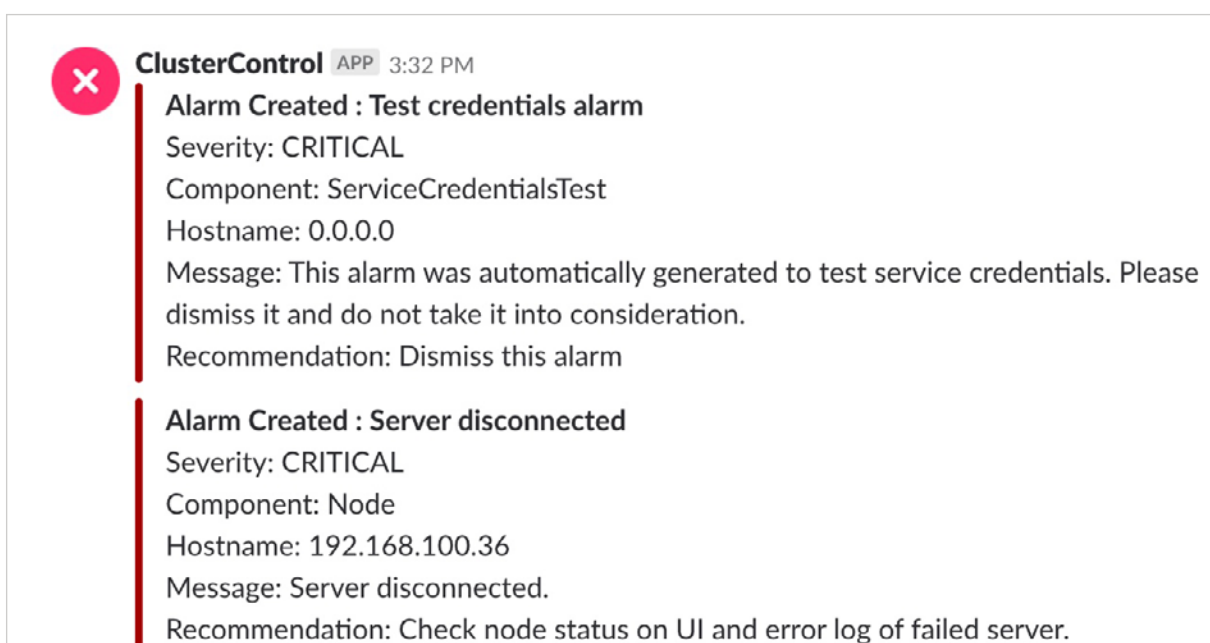


The screenshot shows a Slack chat interface. At the top, a user named Sebastian sends a message at 12:49 AM: "s9s node --list --long". Below this, an application named "ccbot" responds at 12:49 AM: "Running [s9s node --list --long]...". The output of the command is displayed in a code block, showing a table of cluster nodes and their status.

STAT	VERSION	CID	CLUSTER	HOST	PORT	COMMENT
coC-	1.6.1.2572	16	Galera1	192.168.100.10	9500	Up and running
soS-	5.7.22-22-log	16	Galera1	192.168.100.24	3306	Up and running
goM-	5.7.21-20-57-log	16	Galera1	192.168.100.34	3306	Up and running
go--	5.7.21-20-57	16	Galera1	192.168.100.35	3306	Up and running
go--	5.7.21-20-57	16	Galera1	192.168.100.36	3306	Up and running
coC-	1.6.1.2572	17	PostgreSQL1	192.168.100.10	9500	Up and running
poM-	10.4	17	PostgreSQL1	postgres1	5432	Up and running
poS-	10.4	17	PostgreSQL1	postgres2	5432	Up and running
poS-	10.4	17	PostgreSQL1	postgres3	5432	Up and running
coC-	1.6.1.2572	23	MongoDB	192.168.100.10	9500	Up and running
mo--	3.4.15	23	MongoDB	mongo1	27017	Up and Running
mo--	3.4.15	23	MongoDB	mongo2	27017	Up and Running
mo--	3.4.15	23	MongoDB	mongo3	27017	Up and Running
Total: 13						

From Slack, we can use the s9s tool to check the current state of our Cluster.

In the same way, we can receive the alarms in our Slack:



The screenshot shows a Slack chat interface with two critical alarms from the ClusterControl application. The first alarm is titled "Alarm Created : Test credentials alarm" and the second is "Alarm Created : Server disconnected". Both alarms include details such as severity, component, hostname, message, and recommendation.

ClusterControl APP 3:32 PM

Alarm Created : Test credentials alarm
Severity: CRITICAL
Component: ServiceCredentialsTest
Hostname: 0.0.0.0
Message: This alarm was automatically generated to test service credentials. Please dismiss it and do not take it into consideration.
Recommendation: Dismiss this alarm

Alarm Created : Server disconnected
Severity: CRITICAL
Component: Node
Hostname: 192.168.100.36
Message: Server disconnected.
Recommendation: Check node status on UI and error log of failed server.

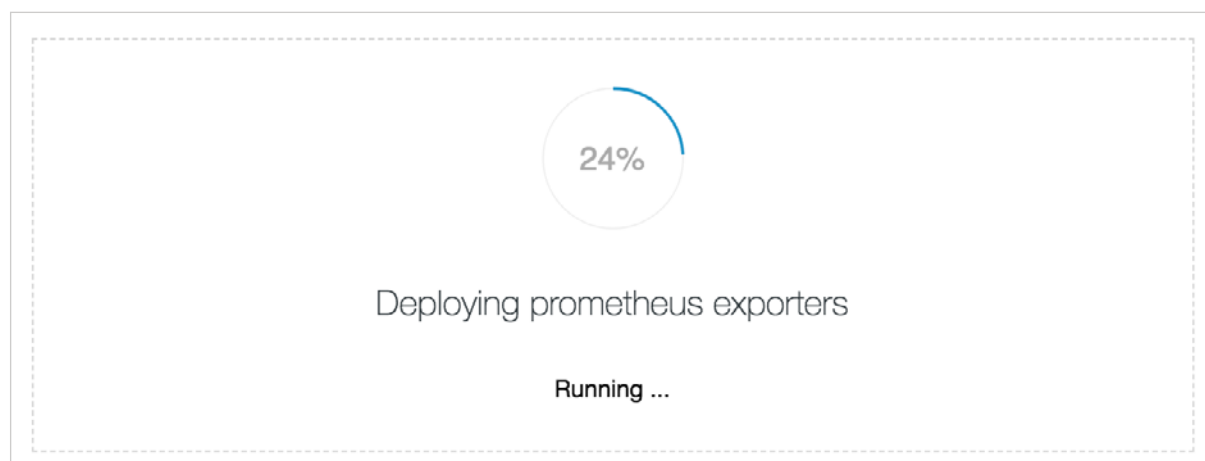
Agent Based Monitoring

Whether one wants to use a monitoring agent or go the agentless route is completely based on organizational policy requirements and custom needs. Although we love the simplicity of not having to install or manage agents on the monitored database hosts, an agent-based approach can provide higher resolution of monitoring data and has certain advantages in terms of security.

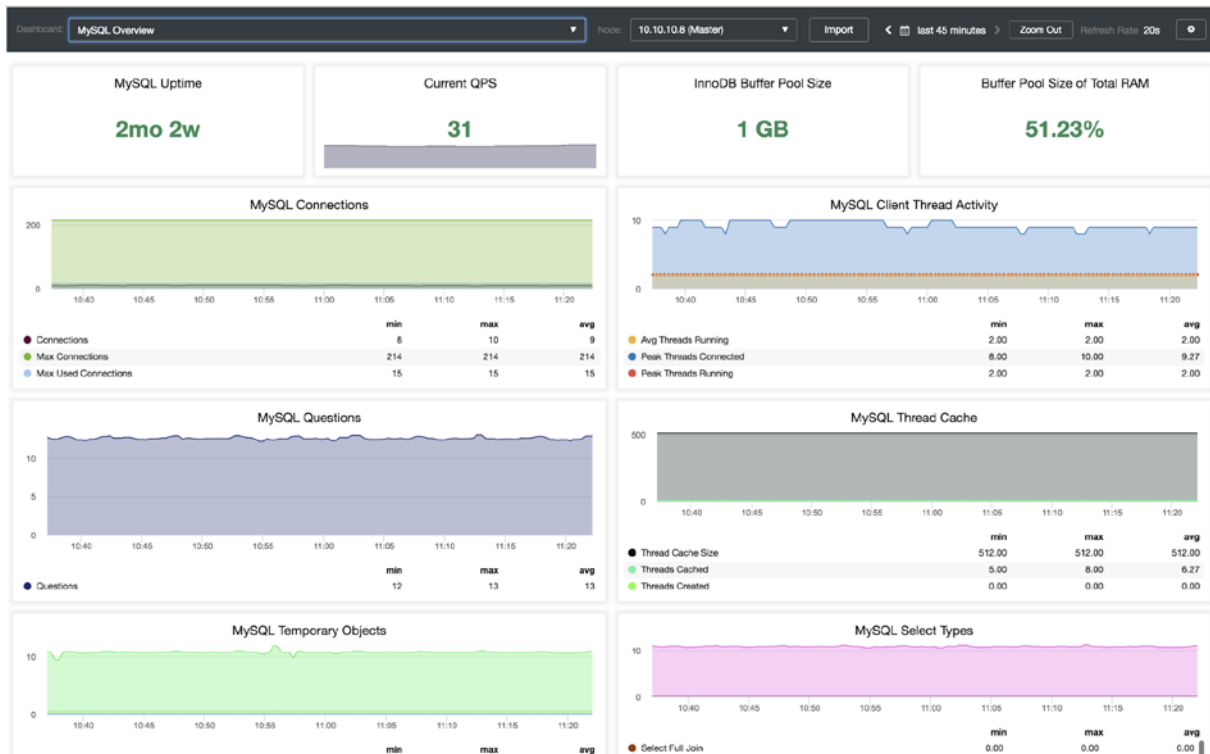
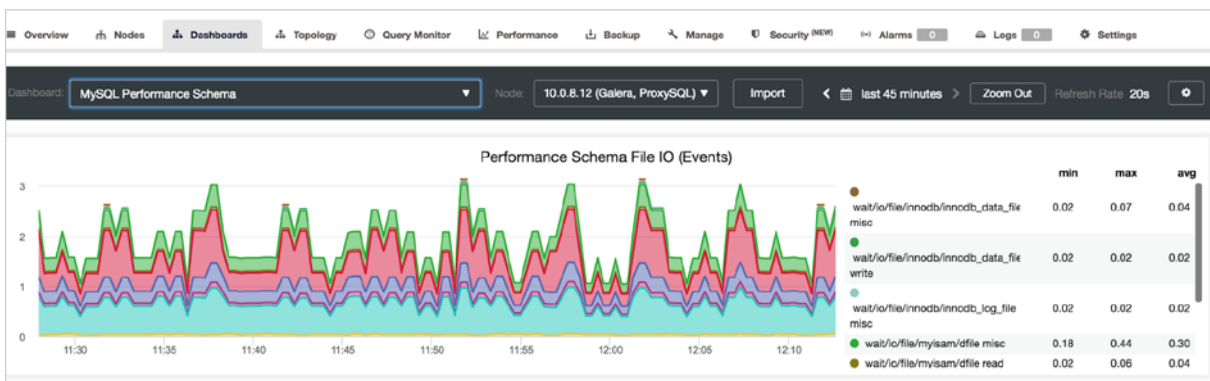
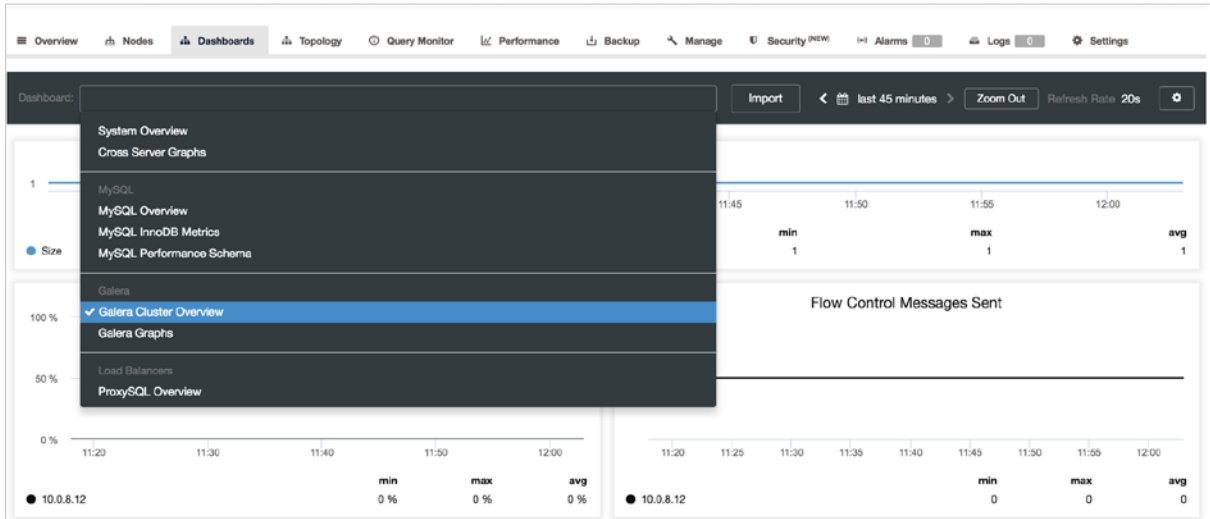
	Agentless	Agent-based
Deployment	Software on ClusterControl host only. Up and running within minutes.	Agent installed on each monitored DB server.
Admin. Overhead.	Only CC needs to be maintained.	Agents need to be maintained, updated, restarted in case of failures.
Configuration	Centralized	Decentralized
Security	Controller to SSH into managed DBs, and require certain privileges.	More secure. Agent communication to DB/OS is internal to the server. No additional firewall rules.
Network Overhead	Additional traffic with more connections.	Local data collection, results processed and then shipped to the server.
Metrics resolution	High resolution at the cost of many connections.	High resolution, a lot of data can be collected locally by the agent. No gaps in data if the network connection fails.

Agent-based monitoring is a new feature added in ClusterControl 1.7. It makes use of Prometheus, a time-series database that can scrape metrics over HTTP from exporters or agents running on the database nodes. One Prometheus server can be used to monitor multiple clusters.

ClusterControl takes care of installing and maintaining Prometheus as well as exporters on the monitored hosts.



ClusterControl has predefined dashboards for each of the supported cluster types.





Deployment and Scaling



Deploying a high availability database cluster manually is not rocket science - there are many how-to's on how to do that. The challenge though is to determine whether what we just deployed is production-ready.

Manual deployments are common, but they can be tedious and repetitive. Depending on the number of nodes in the setup, the deployment steps may be time-consuming and error-prone. Therefore, deployments are increasingly being automated via configuration management tools.

Configuration management tools like Puppet, Chef and Ansible are popular in deploying infrastructure. They help eliminate manual work, minimize the risk of human error, and make it possible to deploy rapidly without giving up reliability. However, deploying a distributed database setup that shares common state undoubtedly leads to complex code. Many of the details that go into deploying, configuring and coordinating between the different nodes increase the complexity exponentially. And the reality is that much more code is written in order to take small edge cases into account, which make the template/module/cookbook/role unmaintainable and hard to extend. And the end result has to be meticulously tested before you can trust it as part of your infrastructure automation. Version changes require the scripts to be updated and tested again.

With quick installation, ease of use, great support, stable deployments and a scalable architecture, ClusterControl is just the solution we were looking for to provide a strong HA platform to our customers.

Xavi Morrus, CMO, MediaCloud

One of the key features of ClusterControl is to automate the deployment of the entire stack. It builds upon the experience of having deployed thousands of production clusters, from master-slave replication setups to multi-master clusters like Galera, NDB Cluster and MongoDB Sharded Clusters, with different database proxies on top.

A high availability stack, deployed through ClusterControl, usually consists of three layers:

- Database layer (e.g., Galera Cluster)
- Reverse proxy layer (e.g., HAProxy or ProxySQL)
- Keepalived layer, which, with use of Virtual IP, ensures high availability of the proxy layer

To perform a deployment in ClusterControl, one can simply select the option “Deploy Database Cluster” and follow the wizard.

- Single process to deploy and manage on-premises and cloud environments
 - Integration with cloud provisioning APIs (AWS, Azure & Google Cloud)

Deployment on Cloud

ClusterControl comes with tighter integration with AWS, Azure and Google Cloud, so it is possible to launch new instances and deploy MySQL, MariaDB, MongoDB and PostgreSQL directly from the ClusterControl user interface. This will enable companies to deploy and manage databases spanning on-premises IT and cloud.

The screenshot shows the 'Cluster Details' step of a five-step wizard. The steps are: 1. Cluster Details, 2. Configure Cluster, 3. Select Credential, 4. Select Virtual Machine, and 5. Deployment Summary. The 'Cluster Details' section includes a beta notice, limitations, and selection options for cluster type and vendor/version.

Cluster Details

1 Cluster Details **2** Configure Cluster **3** Select Credential **4** Select Virtual Machine **5** Deployment Summary

Cluster Details

i This feature is still in beta! We appreciate any feedback for change/feature requests and bugs!

Limitations:

- There is currently no 'accounting' in place for the cloud instances. You will need to manually remove created cloud instances.
- You cannot add or remove a node automatically with cloud instances.
- You cannot deploy a load balancer automatically with a cloud instance.

Select Cluster Type

MySQL	<input checked="" type="checkbox"/> MySQL Galera	mongoDB	<input type="checkbox"/> MongoDB Replica Set
PostgreSQL	<input type="checkbox"/> PostgreSQL Streaming Replication		

Select Vendor and Version

PERCONA	<input checked="" type="checkbox"/> Percona XtraDB Cluster 5.7	MariaDB	<input type="checkbox"/> MariaDB 10.2
---------	--	---------	---------------------------------------

[Continue: Configure Cluster](#)

ClusterControl cloud database provisioning

The steps to deploy a cluster in the cloud are similar to the on-premises deployment, but we need to add information like Cloud Credentials, and Virtual Machine configuration.

ClusterControl cloud database provisioning

Load Balancers

ClusterControl has support for different load balancers, or proxies. HAProxy is a TCP/IP load balancer can be deployed on top of MySQL, MariaDB and PostgreSQL systems. ProxySQL and MaxScale are specialized load balancers that can provide advanced functionality based on their understanding of the MySQL wire protocol. The database nodes are regularly polled for aliveness via custom health-checks that are also deployed by ClusterControl. If a destination stops responding, it is marked as offline, and the traffic is sent to the rest of the available destinations. This prevents traffic from being sent to an inaccessible destination, in which case data may be lost.

With ProxySQL for example, it is possible to see all the queries that passes through the proxy and create rules - e.g., cache a query in the proxy for lower response time, or re-route a query to a particular node. Operations staff can even re-write a badly written query on the fly, while waiting for an application team to fix the query and redeploy the application.

Rule ID	Hits	Active	Match Pattern	Destination Hostgroup	Negate Match Pattern	Apply	Username	Schema Name
Rule 100	0	Yes	^SELECT.* FOR UPDATE	10	No	Yes	NULL	NULL
Rule 200	0	Yes	^SELECT.*	20	No	Yes	NULL	NULL
Rule 300	0	Yes	*	10	No	Yes	NULL	NULL

ClusterControl ProxySQL Rules

For high availability of the proxy layer, it is possible to deploy Keepalived and configure a virtual IP within an active/passive group of servers. This virtual IP is assigned to

an active "main" server. If this server fails, the IP is automatically migrated to the "secondary" passive server, so applications can continue to work with the same IP in order to access the database.

All these elements are essential to configure a high availability database environment.

Integration with Configuration Management Systems via the CLI

ClusterControl includes a tool called `s9s`, which allows us to perform administration tasks, monitoring, implementation, and several tasks that we have already seen, from the command line. In this way, we can easily integrate ClusterControl with the automation tools that we currently have, such as Puppet or Chef. One common pattern is to have e.g., Puppet prepare the host/OS, and let ClusterControl deploy the database software.

Let's see an example of how you can use one command to deploy a 3-node Galera Cluster:

```
1 | $ s9s cluster --create --cluster-type=galera --nodes="192.168.100.130;192.168.100.131;192.168.100.132" --vendor="percona" --provider-version="5.7" --template="my.cnf.repl57" --db-admin="root" --db-admin-passwd="*****" --os-user="root" --cluster-name="GaleraCluster" --wait
2 | Create Galera Cluster
3 | | Job 589 FINISHED [██████████] 100% Job finished.
```

Now we have the new cluster created:

```
1 | $ s9s cluster --cluster-id=7 --list -l
2 | ID STATE TYPE OWNER GROUP NAME COMMENT
3 | 7 STARTED galera system admins GaleraCluster All nodes are operational.
4 | Total: 1
```



Patches and Upgrades



Database vendors regularly issue critical patch updates to address software bugs or known vulnerabilities, but for a variety of reasons, organizations are often unable to install them in a timely manner, if at all. Evidence suggests that companies are actually getting worse at patching databases, with an increased number violating compliance standards and governance policies.

Patching that require database downtime would be of extreme concern in a 24*7 environment, however most cluster upgrades can be performed online. ClusterControl is able to perform a rolling upgrade of a distributed environment, upgrading and restarting one node at a time. The logical upgrade steps might slightly differ between the different cluster types. Load balancers would automatically blacklist unavailable nodes that are currently being upgraded, so that applications are not affected.

Severalnines is so ingrained in our database infrastructure that every operational exercise we go through involves using the tool, from provisioning new nodes to rolling upgrades and patches.

Renier du Plessis, Digital Operations Manager, The Mail & Guardian Online

Operational Reporting on Version Upgrades

Patches and upgrades is an area that require constant attention, especially with the proliferation of open source databases in many organisations and more database environments being distributed for high availability. ClusterControl provides a solid operational reporting framework, and can help answer simple questions like:

- What versions of software are running across the environment?
- Which servers should be upgraded?
- Which servers are missing critical updates?

Reports can be scheduled, and automatically mailed out to a list of recipients:

Upgrade Report: 2018-09-17



This is the Upgrade report for a managed cluster.

Upgrade Summary

Below follows a summary of the number of packages that can be updated.

Hostname	Managed Services	Installed Version	# Total Packages	# Db Packages	# Security Packages	# Other Packages
192.168.100.130	galera	5.7.22-22-57	14	0	0	14
192.168.100.131	galera	5.7.22-22-57	14	0	0	14
192.168.100.132	galera	5.7.22-22-57	14	0	0	14

192.168.100.130

Database Packages

Package	Available Version	Installed Version
---------	-------------------	-------------------

[back to top](#)

Security Packages

Package	Available Version	Installed Version
---------	-------------------	-------------------

[back to top](#)

Other Packages

Package	Available Version	Installed Version
audit-libs.x86_64	2.8.1-3.el7_5.1	2.8.1-3.el7
bind-libs-lite.x86_64	32:9.9.4-61.el7_5.1	32:9.9.4-61.el7

ClusterControl Operational Report for Upgrades



Security and Compliance



Database security requires careful planning, but it is important to remember that security is not a state, it is a process. Once the database is in place, monitoring, alerting and reporting on changes are an integral part of the ongoing management. Also, security efforts need to be aligned with business needs.

Some best practices include:

- Control access to the database, use the principle of least privilege.
- Encrypt data, in-transit and/or at-rest. This also applies to backups.
- Regularly patch the database.
- Monitor database activity. Ensure you have complete visibility into what is happening across your databases, and reduce the risk of missing suspicious activities.

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. Having a management tool like ClusterControl has helped us achieve our goals.

Idris Khanafi, Head of Infrastructure,
Veritrans

Securing ClusterControl Traffic

To avoid unauthorized access to the management of the database, encrypt all communication to the ClusterControl Web Interface. As a platform that manages all of your databases, ClusterControl maintains the communication with the database servers, transmits commands and collects metrics in an encrypted way. By default,

ClusterControl is setup with HTTPS. All you need to do is to point your browser to <https://<ip of ClusterControl host>/clustercontrol>.

ClusterControl Secure Backups

You can use this feature on all backup methods (mysqldump, xtrabackup, mongodump, pg_dump) supported by ClusterControl. To enable encryption, toggle the "Enable Encryption" switch when scheduling or creating the backup. ClusterControl automatically generates a key to encrypt the backup. It uses AES-256 (CBC) encryption algorithm and performs the encryption on-the-fly on the target server. The following command shows an example of how ClusterControl performs a mysqldump backup:

```
1 | $ mysqldump --defaults-file=/etc/my.cnf --flush-privileges --hex-blob --opt --no-create-info --no-data --triggers --routines --events --single-transaction --skip-comments --skip-lock-tables --skip-add-locks --databases db1 | gzip -6 -c | openssl enc -aes-256-cbc -pass file:/var/tmp/cmon-094508-e0bc6ad658e88d93.tmp | socat - TCP4:192.168.55.170:9999'
```

You would see the following error if you tried to decompress an encrypted backup without decrypting it first with the proper key:

```
1 | $ gunzip mysqldump_2018-01-03_175727_data.sql.gz
2 | gzip: mysqldump_2018-01-03_175727_data.sql.gz: not in gzip format
```

The key is stored inside the ClusterControl database, and can be retrieved from the `cmon_backup.metadata` file for a particular backup set. It will be used by ClusterControl when performing restoration. Encrypting backups are highly recommended, especially when you want to secure your backups offsite like archiving them in the cloud. The supported cloud storage services include Amazon S3, Google Cloud Storage and Azure Cloud Storage.

Backup List (1-4 out of 4)	Status	Method	Databases	Storage Location
 Backup Set: 4 Restore More Actions	✔ Completed at 01/05/2018 @ 4:00PM (MYT)	xtrabackup (full)	all	
 Backup Set: 3 Restore More Actions	✔ Completed at 01/03/2018 @ 5:58PM (MYT)	mysqldump	all	
 Backup Set: 2 Restore More Actions	✔ Completed at 12/31/2017 @ 8:01AM (MYT)	mysqldump (pitr compatible)	all	
 Backup Set: 1 Restore More Actions	✔ Completed at 12/30/2017 @ 7:36AM (MYT)	mysqldump (pitr compatible)	all	

ClusterControl backup list view

Encryption of Data in Transit (SSL)

You can increase the reliability of your database service by using client-server SSL encryption. Using ClusterControl, you can perform this operation with simple point and click:

Create SSL Encryption

Setup encrypted SSL client-server connections for the node(s). The same certificates will be used on all nodes (the existing ones might be overwritten).

Generate Self-Signed Certificate

Create Certificate

Certificate Expiration (days): 3650

Created Certificates

Use Certificate:

<select the certificate to use>

/var/lib/cmon/ca

- galera
 - cluster_2
 - client
 - galera_rep
 - server
 - server_ca

Restart Cluster

To setup SSL encryption the nodes must be restarted.
Select 'Restart Nodes' to perform a rolling restart of the nodes.

Restart Nodes Do Not Restart Nodes

The nodes will be restarted with a rolling restart.
Applications may be affected during this restart.

Proceed Cancel

ClusterControl SSL encryption creation

You can then retrieve the generated keys and certificates directly from the ClusterControl host under `/var/lib/cmon/ca` path to establish secure connections with the database clients. All the keys and certificates can be managed directly under Key Management, as described further down.

Replication traffic within a Galera Cluster can also be enabled with just one click. ClusterControl uses a 2048-bit default key and certificate generated on the ClusterControl node, which is transferred to all the Galera nodes. A cluster restart is necessary to enable this. ClusterControl will perform a rolling restart operation, taking one node at a time. You will see a green lock icon next to the database server (Galera indicates Galera Replication encryption, while SSL indicates client-server encryption) in the Hosts grid of the Overview page once encryption is enabled.

All the keys and certificates can be managed directly under Key Management..

Key Management

All the generated keys and certificates can be managed directly from the ClusterControl UI. Key Management allows you to manage SSL certificates and keys that can be provisioned on your clusters.

If the certificate has expired, you can simply use the UI to generate a new certificate with proper key and Certificate Authority (CA), or import an existing key and certificate into the ClusterControl host.

Security Advisors

Advisors are mini-programs that run in ClusterControl. They perform specific tasks and provide advice on how to address issues in areas such as performance, security, log management, configuration, storage space and others. Each advisor can be scheduled like a cron job, and run as a standalone executable within the ClusterControl UI. It can also be run via the ClusterControl 's9s' command line client.

ClusterControl enables some security advisors for MySQL-based systems, including:

- Access from any host ('%') - Identifies all users that use a wildcard host from the `mysql` system table, and lets you have more control over which hosts are able to connect to the servers.
- Check number of accounts without a password - Identifies all users who do not have a password in the `mysql` system table.

For MongoDB, we have the following advisors:

- MongoDB authentication enabled - Check whether the MongoDB instance is running with authentication mode enabled.
- Authorization check - Check whether MongoDB users are authorized with too permissive role for access control.

For more details on how does ClusterControl performs the security checks, you can look at the advisor JavaScript-like source code under Manage -> Developer Studio. You can see the execution results from the Advisors page:

S9s/Mongodb/Sharding Advisors			
N/A	Execute this 00:00 every day	Last execution 10 hours ago	Edit Disable v
S9s/Host Advisors			
Warning Swappiness check	Execute this 01:00 every day	Last execution 9 hours ago	Edit Disable v
Ok Excessive CPU Usage	Execute this Every 30 minutes	Last execution 11 minutes ago	Edit Disable v
Ok Checking Disk Space Usage	Execute this Every 30 minutes	Last execution 11 minutes ago	Edit Disable v
Ok NUMA Check	Execute this Every hour, on the hour	Last execution 41 minutes ago	Edit Disable v
S9s/Mongodb/Connections Advisors			
Ok Connections used	Execute this Every minute	Last execution a few seconds ago	Edit Disable v
Ok Connections used	Execute this Every minute	Last execution a few seconds ago	Edit Disable v
S9s/Mongodb/Mmap Advisors			
Ok Collection lock percentage	Execute this Every 20 minutes	Last execution a minute ago	Edit Disable v
S9s/Mongodb/Replication Advisors			
Ok Replication check	Execute this Every minute	Last execution a few seconds ago	Edit Disable v
Ok Replication window	Execute this Every 20 minutes	Last execution a minute ago	Edit Disable v

ClusterControl Advisors for MongoDB

Within ClusterControl the default roles are: Super Admin, Admin and User. The Super Admin is the only account that can administer teams, users and roles. The Super Admin is also able to migrate clusters across teams or organizations. The admin role belongs to a specific organization and is able to see all clusters in this organization. The user role is only able to see the clusters he/she created.

User Roles

You can add new roles within the Role Based Access Control screen. You can define the privileges per functionality whether the role is allowed (read-only), denied (deny), manage (allow change) or modify (extended manage).

The MySQL privilege system ensures that all users can perform only the operations they are allowed to. Granting is critical as you don't want to give all users complete access to your database, but you need users to have the necessary permissions to run queries and perform daily tasks.

ClusterControl provides an interactive user interface to manage the database schemas and privileges. It unifies the accounts on all MySQL servers in the cluster and simplifies the granting process. You can easily visualize the database users, so you avoid making mistakes.

Audit Log for MySQL

Continuous auditing is an imperative task for monitoring your database environment. By auditing your database, you can achieve accountability for actions taken or content accessed. Moreover, the audit may include some critical system components, such as the ones associated with financial data to support a precise set of regulations like SOX, or the EU GDPR regulation. Usually, it is achieved by logging information about DB operations on the database to an external log file.

By default, auditing in MySQL or MariaDB is disabled. You can achieve it by installing additional plugins or by capturing all queries with the `query_log` parameter. The general query log file is a general record of what MySQL is performing. The server records some information to this log when clients connect or disconnect, and it logs each SQL statement received from clients. Due to performance issues and lack of configuration options, the `general_log` is not a good solution for security audit purposes.

If you use MySQL Enterprise, you can use the MySQL Enterprise Audit plugin which is an extension to the proprietary MySQL version. MySQL Enterprise Audit Plugin plugin is only available with MySQL Enterprise, which is a commercial offering from Oracle. Percona and MariaDB have created their own open source versions of the audit plugin.

ClusterControl can be used to enable Audit Logging for MySQL or MariaDB based systems. It will ensure the logging of connection and query activity to a separate file.

Database Infrastructure Audit

There are a few Ops Reports that shows what clusters are running, the nodes that belong to each cluster, uptime statistics, whether they have backups or not, as well as trends in utilization/capacity usage. The upgrade report may be of particular interest, as it gives a comprehensive list of all hosts, all services running on them, versions that are installed, whether the installed packages are up-to-date and secure. The Upgrade Report gathers information from the operating system and compares them to packages available in the repository.

The report is divided into four sections; upgrade summary, database packages, security packages, and other packages. You can quickly compare what you have installed on your system and find a recommended upgrade or patch.

The screenshot shows the 'Enable Audit Log' configuration window. It includes the following elements:

- Log Path:** `audit.log`
- Format:** `OLD`
- Rotation Size:** `1024 MB`
- Rotations:** `5`
- Strategy:** `ASYNCHRONOUS`
- Log Content:** A dropdown menu is open with options: `ALL`, `LOGINS` (selected), `QUERIES`, and `NONE`.
- Username:** A field labeled 'Enter a username'.
- Advanced Options:** A checkbox for 'Hide advanced options'.
- Cluster-wide Note:** 'This feature will be enabled on all nodes of this cluster.'
- Buttons:** 'Cancel' and 'Enable Audit Log'.

Upgrade Report: 2018-08-09

This is the Upgrade report for a managed cluster.

Upgrade Summary

Below follows a summary of the number of packages that can be updated.

Hostname	Managed Services	Installed Version	# Total Packages	# Db Packages	# Security Packages	# Other Packages
10.0.3.8	mysql proxysql	10.1.24-MariaDB-1-trusty 1.3.6	12	8	0	4
10.0.3.12			Failed to get list of packages.			
10.0.3.100	mysql	10.1.33-MariaDB-1-trusty	55	2	5	48

ClusterControl infrastructure check

Database Packages

Package	Available Version	Installed Version
mariadb-server	10.1.25+maria-1-trusty	10.1.24+maria-1-trusty
mariadb-client	10.1.25+maria-1-trusty	10.1.24+maria-1-trusty
mysql-common	10.1.25+maria-1-trusty	10.1.24+maria-1-trusty
mariadb-common	10.1.25+maria-1-trusty	10.1.24+maria-1-trusty
mariadb-client-core-10.1	10.1.25+maria-1-trusty	10.1.24+maria-1-trusty
mariadb-client-10.1	10.1.25+maria-1-trusty	10.1.24+maria-1-trusty
mariadb-server-core-10.1	10.1.25+maria-1-trusty	10.1.24+maria-1-trusty
mariadb-server-10.1	10.1.25+maria-1-trusty	10.1.24+maria-1-trusty

ClusterControl infrastructure check



Operational Reports



You may already have a couple of monitoring tools with all possible metrics/graphs, and you probably have also set up alerts based on metrics and thresholds. Some will even have automated advisors providing recommendations or fixing things automatically. That's good - having visibility into your system is essential; nevertheless, you need to be able to process a lot of information.

To make sure your systems are in a good shape, you'd need to go through quite a lot of information - host statistics, database server statistics, workload statistics, state of backups, database packages, logs and so forth. Such data should be available in every properly monitored environment, although sometimes it is scattered across multiple locations - you may have one tool to monitor database state, another tool to collect system statistics, maybe a set of scripts, e.g., to check the state of your backups. This makes health checks much more time-consuming than they should be - the DBA has to put together the different pieces to understand the state of the system. Integrated tools like ClusterControl have an advantage that all of the different bits of information are located in the same place.

ClusterControl operational reports arm you with information about your database infrastructure status, which you can use to audit your environment or as part of operational support. These reports consist of different checks and address various day-to-day DBA tasks. The idea behind ClusterControl operational reporting is to put all of the most relevant data into a single document which can be quickly analyzed in order to get a clear understanding of the status of the databases and its processes.

ClusterControl can help cover several aspects of compliance. Starting with backup history details, which you can use to track things like backup completion, history and servers without a proper backup policy to package upgrade reports with outdated system packages and schema changes. With a few steps, you can schedule enterprise level checks on your open source databases. All of this will give your management and support teams better insight into your DB operations.

Operational Reports can be scheduled, or created on-demand. These include:

- The operational health of databases
- Service availability & uptime
- Resource usage for capacity planning
- Insight to optimize database operations
 - Underutilized databases
 - Changes to database schemas
 - Which systems cause the most downtime
 - Which systems will need more capacity
 - Which database software versions are currently running
 - Which systems need upgrading
 - Which systems have backups, and which ones don't





Configuration Management



Configuration management represents the source of the configuration items for the database - from the database software packages to configuration files. Having the exact version of the database software available in a local repository ensures that instances of that version only will be deployed to expand existing setups, or create new ones. Being able to view and edit configuration files of multiple instances from one single point simplifies the administrator's job of keeping track of what is deployed, and how changes are rolled out. For certain cluster types, configurations ought to be similar across all nodes. So a way of detecting diverging configuration settings can help avoid problems further down the line.

DB Variables
DB Variables

Search:

Note: **RED** text means that the variable setting is different. In some cases that is ok.

Variable	10.0.3.60	10.0.3.50	10.0.3.70	10.0.3.71
audit_log_buffer_size	1048576	1048576	1048576	1048576
audit_log_exclude_accounts				
audit_log_exclude_commands				
audit_log_file	audit.log	audit.log	audit.log	audit.log
audit_log_flush	OFF	OFF	OFF	OFF
audit_log_format	OLD	OLD	OLD	OLD
audit_log_handler	FILE	FILE	FILE	FILE
audit_log_include_accounts				
audit_log_include_commands				
audit_log_policy	ALL	ALL	ALL	LOGINS
audit_log_rotate_on_size	0	0	0	1073741824

ClusterControl manage database variables

ClusterControl generates a configuration when deploying a database setup - just fill in some values (database vendor, data directory, password and hostnames) in the

deployment wizard and you're good to go. The rest of the configuration options will be automatically determined (and calculated) based on the host specifications (CPU cores, memory, IP address, etc.) and applied to the template file that comes with ClusterControl.

ClusterControl will load the base content of the Galera configuration template from /usr/share/cmon/templates/my.cnf.galera into the CMON database after deployment succeeds. You can then customize your own configuration file directly in the ClusterControl UI. Whenever you hit the Save button, the new version of configuration template will be stored inside CMON database, without overwriting the base template file.

There are a number configuration variables which are configurable dynamically by ClusterControl. These variables are represented with capital letters enclosed by the '@' sign, for example @DATADIR@. For full details on supported variables, please refer to [this page](#).

If the dynamic variable is replaced with a value (or undefined), ClusterControl will skip it and use the configured value instead. This is handy for advanced users, who usually have their own set of configuration options that are tailored for specific database workloads.

All configuration files can be viewed in the Web Interface, and configuration changes can be applied dynamically on the instance as well as persisted on disk. Parameters can be set on one or more instances that form part of the cluster. The admin is also advised when a rolling restart is required in order to apply changes, this can also be triggered from ClusterControl.

The screenshot shows the ClusterControl web interface for configuration management. On the left, a tree view lists configuration files for various hosts and services, including Galera master nodes, MySQL instances, and InnoDB options. The main panel displays the content of a selected MySQL configuration file (my.cnf) for host 10.0.3.60. The file content includes MySQL and InnoDB settings. Below the code editor, there is an 'Info' section with 'File Details (my.cnf)' showing metadata like size, host name, path, and last changed date.

```
8 [MYSQLD]
9 user=mysql
10 basedir=/usr/
11 datadir=/var/lib/mysql
12 socket=/var/run/mysqld/mysqld.sock
13 pid-file=mysqld.pid
14 port=3306
15 gtid_mode=ON
16 enforce_gtid_consistency
17 log_error=error.log
18 # bind-address = 127.0.0.1
19 # log-output = FILE
20 # relay-log = relay-bin
21 ### INNODB OPTIONS
22 innodb-buffer-pool-size=512M
23 innodb-additional-mem-pool-size=10M
24 innodb-flush-log-at-trx-commit=2
25 innodb-file-per-table=1
26 innodb-data-file-path = ibdata1:100M:autoextend
27 ## You may want to tune the below depending on number of cores and disk sub
28 innodb-read-io-threads=4
29 innodb-write-io-threads=4
30 innodb-doublewrite=1
31 innodb-log-file-size=256M
32 innodb-log-buffer-size=32M
33 # innodb-buffer-pool-instances = 4
34 innodb-log-files-in-group=2
35 innodb-thread-concurrency=0
36 # innodb-file-format = barracuda
37 innodb-flush-method = O_DIRECT
38 innodb-locks-unsafe-for-binlog=1
```

File Details (my.cnf)

Size:	5885
Host Name:	10.0.3.60 (galera - master)
Path:	/etc/mysql/my.cnf
Last Changed:	18 Sep 2018 14:00:04

ClusterControl edit service configuration



Automatic Recovery & Repair



Outages are disruptive to any business, and they also cost money. A manual failover strategy is probably not the most efficient way to reduce downtime, since it does take time for somebody to be alerted, to get to a terminal, analyze logs and error messages to understand the issue, before finally starting a recovery procedure. So why not use an automated failover strategy?

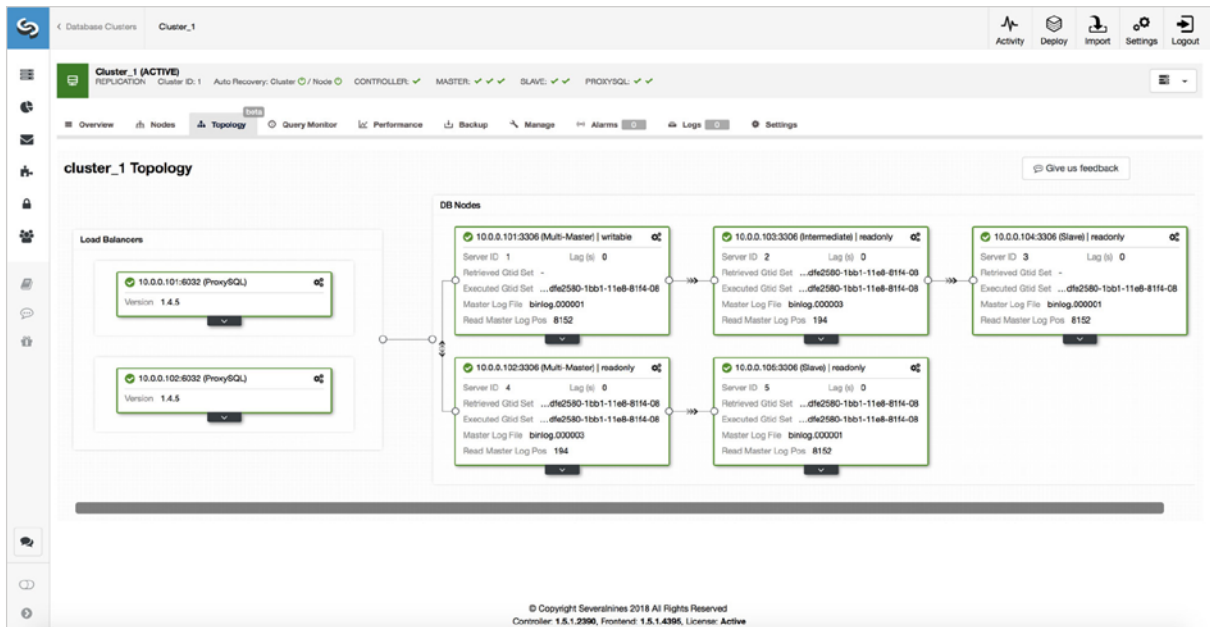
Our guess is that many companies wished they had some sort of automated failover, and the reasons not to implement it are probably the complexity of the existing solutions, lack of competence in implementing such solutions, or lack of trust in software to take such an important decision.

There are a number of automated failover solutions, including (and not limited to) MHA, MMM, MRM, mysqlfailover, Orchestrator and ClusterControl. Some of them have been on the market for a number of years, others are more recent. That is a good sign, multiple solutions mean that the market is there and people are trying to address the problem.

When we designed automatic failover within ClusterControl, we used a few guiding principles.

Make Sure the Master Is Really Dead Before You Failover

- In case of a network partition, where the failover software loses contact with the master, it will stop seeing it. But the master might be working well and can be seen by the rest of the replication topology.
- ClusterControl gathers information from all the database nodes as well as any database proxies/load balancers used, and then builds a representation of the topology. It will not attempt a failover if the slaves can see the master, nor if ClusterControl is not 100% sure about the state of the master.
- ClusterControl also makes it easy to visualize the topology of the setup, as well as the status of the different nodes (this is ClusterControl's understanding of the state of the system, based on the information it gathers).



ClusterControl topology view

Failover Only Once

Much has been written about flapping. It can get very messy if the availability tool decides to do multiple failovers. That's a dangerous situation. Each master elected, however brief the period it held the master role, might have their own sets of changes that were never replicated to any server. So you may end up with inconsistency across all the elected masters.

Do not Failover to an Inconsistent Slave

When selecting a slave to promote as master, we ensure the slave does not have inconsistencies, e.g. errant transactions, as this may very well break replication.

Only Write to the Master

Replication goes from the master to the slave(s). Writing directly to a slave would create a diverging dataset, and that can be a potential source of problem. We set the slaves to `read_only`, and `super_read_only` in more recent versions of MySQL or MariaDB. We also advise the use of a load balancer, e.g., ProxySQL or MaxScale, to shield the application layer from the underlying database topology and any changes to it. The load balancer also enforces writes on the current master.

Do not Automatically Recover the Failed Master

If the master has failed and a new master has been elected, ClusterControl will not try to recover the failed master. Why? That server might have data that has not yet been replicated, and the administrator would need to do some investigation into the failure. Ok, you can still configure ClusterControl to wipe out the data on the failed master and have it join as a slave to the new master - if you are ok with losing some data. But by default, ClusterControl will let the failed master be, until someone looks at it and decides to re-introduce it into the topology.

Recovery algorithms are specific to the high availability model of the database. Galera Cluster, NDB Cluster, MySQL Replication, PostgreSQL Streaming Replication and MongoDB ReplicaSets all have their own approaches for failover and recovery.

Failover can also be customized. It can be enabled/disabled at node or cluster level, blacklists and whitelists ensure only the right instances are promoted to masters, and integration hooks for pre- and post-failover actions/programs ensures that behavior during failover can be customized.



Performance Management



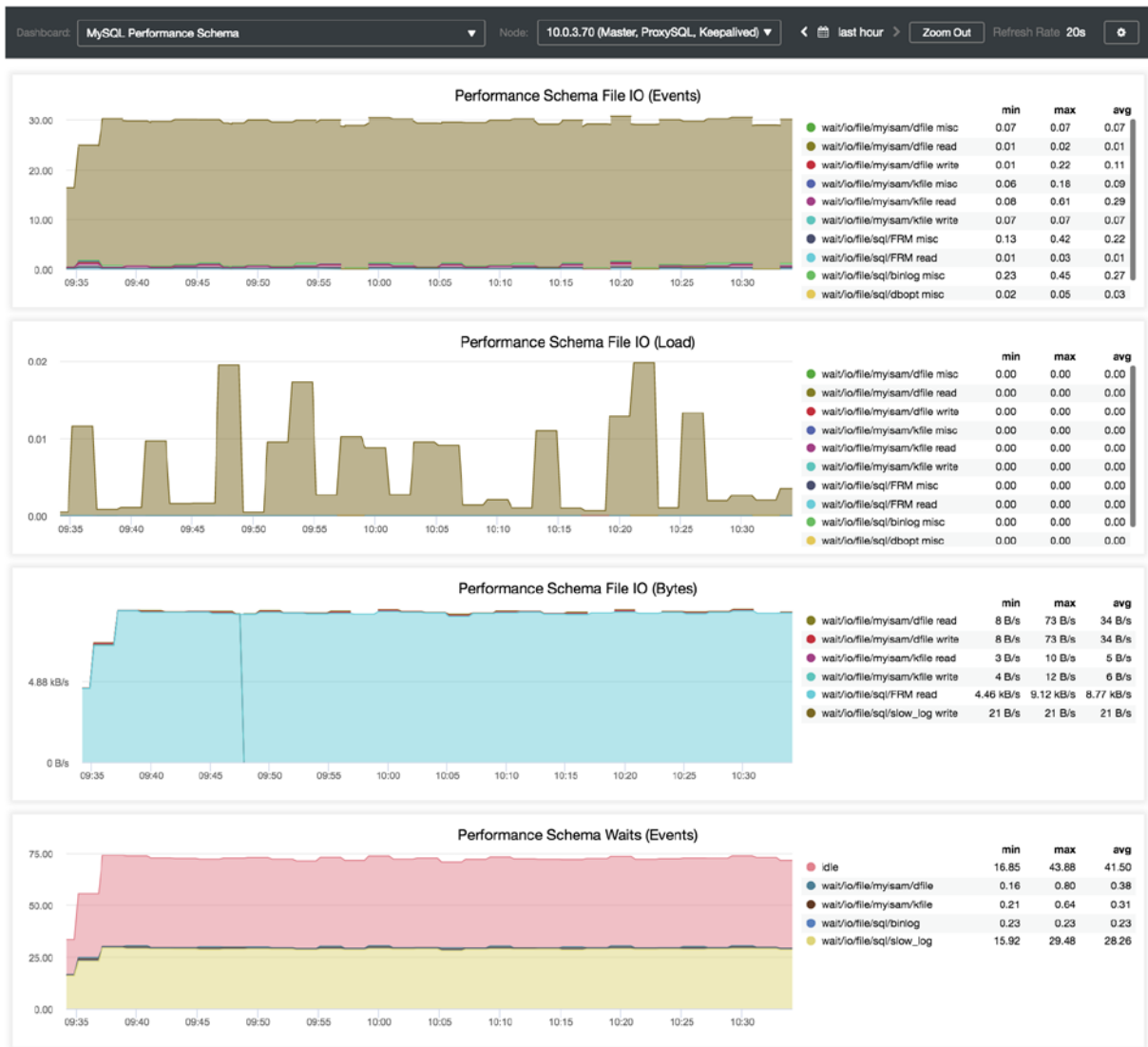
Monitoring performance of production databases is one of the most important tasks within database administration. It is an ongoing process that requires constant attention, since databases usually evolve with time - database size, number of users, workload, schema changes that come with application changes, and so on.

Traditional monitoring tools normally assess the health of a database (is it working, or is it broken?), but don't identify database performance issues. Modern APM tools do a great job at providing a holistic view of applications and databases, but the downside is that they lack depth of information in some parts of the stack. For instance, most of the tools available provide a much more detailed view of application activity as compared to database activity. According to research firm Gleanster LLC, the database is the number 1 source of issues with performance. Without the appropriate tools, the ops team would need to spend an inordinate amount of time combing through cumbersome SQL logs to find performance problems.

With ClusterControl in place, StreamAMG's flagship product is now backed with a fully automated database infrastructure which allows us to ensure excellent uptime. 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, Manager, StreamAMG

ClusterControl offers a number of dashboards to present performance of various portions of the database.



ClusterControl performance dashboard

Workload analytics provides visibility into transactions/queries from applications. Performance anomalies are never expected, but they do happen and are easy to miss in a sea of data. Outlier detection will find any queries that suddenly start to execute much slower than normal. It tracks the moving average and standard deviation for query execution times, and detects/alerts when the difference between the value exceeds the mean by 2 standard deviations.

Top Queries
Query monitor is enabled Off On Settings

Search Query Text All hosts Occurrences Refresh rate: 30 secs Show: 20 rows

Purge Query Monitor

Query	DB	Count	Rows		Tmp tables			Exec time			Total exec time		Last seen
			Sent	Examined	RAM	On Disk	Max	Avg	Stdev	Absolute	Relative %		
SELECT ?	sakila	33710	1	0	0	0	1 s	139 us	15.19 ms	00:00:04	5.72	2 days ago	
SELECT @@log_bin	sakila	4510	1	0	0	0	10.29 ms	103 us	725 us	00:00:00	0.57	2 days ago	
select @@global.warnings_provider_options	sakila	1351	1	0	0	0	914 ms	55.96 ms	421 ms	00:01:15	91.83	2 days ago	
select @@collation_database	test	201	1	0	0	0	53 us	35 us	4 us	00:00:00	0.01	19 hours ago	
SET SESSION 'character_set_results' = ?	sakila	94	0	0	0	0	114 us	34 us	8 us	00:00:00	0.00	19 hours ago	
SELECT OBJECT_SCHEMA, OBJECT_NAME, COUNT_FETCH, COUNT_INSERT, COUNT_UPDATE, CO...	sakila	71	228	300	0	0	30.09 ms	8.61 ms	4.83 ms	00:00:00	0.74	2 days ago	
SET SESSION 'character_set_results' = ?	bertil	36	0	0	0	0	74 us	27 us	5 us	00:00:00	0.00	19 hours ago	
select @@collation_database	'myod'	28	1	0	0	0	49 us	36 us	5 us	00:00:00	0.00	19 hours ago	
SELECT OBJECT_SCHEMA, OBJECT_NAME, INU(INDEX_NAME, ?) as INDEX_NAME, COUNT_F...	sakila	28	482	577	0	0	38.73 ms	15.32 ms	3.14 ms	00:00:00	0.52	2 days ago	
select @@collation_database	'sakila'	24	1	0	0	0	62 us	38 us	8 us	00:00:00	0.00	19 hours ago	
SET 'SQL_QUOTE_SHOW_CREATE' = ?	sakila	21	0	0	0	0	70 us	30 us	4 us	00:00:00	0.00	19 hours ago	
SELECT OBJECT_SCHEMA, OBJECT_NAME, COUNT_READ_NORMAL, COUNT_READ_WITH_SHARED...	sakila	21	228	300	0	0	62.89 ms	13.04 ms	13.09 ms	00:00:00	0.33	2 days ago	
ROLLBACK TO SAVEPOINT 'sp'	sakila	20	0	0	0	0	68 us	31 us	6 us	00:00:00	0.00	19 hours ago	
USE 'sakila'	sakila	20	0	0	0	0	62 us	27 us	5 us	00:00:00	0.00	19 hours ago	
SELECT @@collation_database'	sakila	20	0	0	0	0	91 us	43 us	6 us	00:00:00	0.00	19 hours ago	
SELECT EVENT_NAME, COUNT_STAR, SUM_TIMER_WAIT FROM performance_schema.events_wa...	sakila	20	140	140	0	0	3.48 ms	2.76 ms	365 us	00:00:00	0.07	2 days ago	

Wait time analysis can help understand how much time a database query spends across all of its execution stages in a given time period. Instead of focusing only on resource usage like CPU, memory or disk IO, all that would really matter is how much time is needed to execute some commands. You may also have a very fast SQL that runs in a few milliseconds, but if it has to run a million times a day, it can ruin your application performance.

Usually, wait time analysis starts from individual SQL statements that have most impact on overall database execution time, and breaks down each step to the millisecond. Supplied with wait time and lock time information, and combined with information from the performance dashboards, you can recognize the most significant contributor to the slow performance.



Automatic Performance Advisors



A big chunk of ops management consists of tracking your monitoring systems. Raw metrics can be interpreted and consolidated in various ways to give you insight into your database operations and whereby to optimize them. Looking at metrics on their own is not enough though, since there are hundreds of metrics with sophisticated relations between them. Deeper workload analysis requires metrics to be combined and computed in different ways.

Advisors		Advisors results		Schedule Advisor		Create Custom Advisor	
Show Advisors: All s9s mysql security schema replication p_s innodb general galera connections host							
S9s/Mysql/Galera Advisors							
Warning	GRA Log Checker	Execute this 01:00 every day	Last execution 11 hours ago	Edit	Disable	▼	
Ok	wrep_cluster_address check	Execute this Every 20 minutes	Last execution 8 minutes ago	Edit	Disable	▼	
Ok	wrep_node_name check	Execute this 00:02 every day	Last execution 12 hours ago	Edit	Disable	▼	
Ok	Wrep slave threads check	Execute this 00:05 every day	Last execution 12 hours ago	Edit	Disable	▼	
S9s/Host Advisors							
Warning	Swappiness check	Execute this 01:00 every day	Last execution 11 hours ago	Edit	Disable	▼	
Ok	Excessive CPU Usage	Execute this Every 30 minutes	Last execution 28 minutes ago	Edit	Disable	▼	
Ok	Checking Disk Space Usage	Execute this Every 30 minutes	Last execution 28 minutes ago	Edit	Disable	▼	
S9s/Mysql/P_s Advisors							
Warning	Table access without using index	Execute this 00:00 every day	Last execution 12 hours ago	Edit	Disable	▼	
Warning	Unused indexes	Execute this 00:00 every day	Last execution 12 hours ago	Edit	Disable	▼	
Ok	Performance Schema	Execute this 00:00 every day	Last execution 12 hours ago	Edit	Disable	▼	
Ok	Top Queries	Execute this Every 30 minutes	Last execution 28 minutes ago	Edit	Disable	▼	

ClusterControl advisors view

ClusterControl provides an advisor engine that allows for deeper analysis of workload and resource usage. Advisors are mini-programs that are executed by ClusterControl, either on-demand or after a schedule. They can be anything from simple configuration guidance, warning on thresholds or more complicated rules for forecasts or cluster-wide self-regulation tasks based on metrics data. In general, advisors perform more detailed analysis and produce more comprehensive recommendations than alerts.

A number of predefined advisors are available, they can be categorized in different areas - security, schema, replication, performance schema, InnoDB, Galera, connections, and hosts. An example of an elaborate advisor for a specific clustering technology (Galera Cluster for MySQL or MariaDB) determines the write load on the cluster and rates if the Galera cache file is adequate in size to support a replication window threshold. Another predictive type advisor which is independent of database type checks the historical disk usage/growth rate and predicts when a host may run out of disk space.

You can view the catalog of advisors, and for each advisor, the date/time since the last update. Some advisors are scheduled to run only once a day so their advice may no longer reflect the reality - for instance, if you already resolved the issue you were warned about. Advisors are stored in the ClusterControl database, and can be managed from the GUI interface, where you can enable, disable, and modify execution times. You can also manually re-run the advisor. We also have a public Github repository where advisors can be shared with other ClusterControl users.

Custom Advisors

Custom Advisors can be created via a simple graphical interface. The wizard below provides a quick way to customize alerts based on thresholds.

Custom Advisor: Create new custom advisor

Type: Applies To:
Resource: Nodes:

The custom advisor allows you to set threshold to be alerted on if a metric falls below or raises above the threshold and stays there for a specified timeframe.

Condition

If metric:

Condition: For(s): Warning: Critical:

Max Values seen for selected period

750000.00
500000.00
250000.00
0.00

16:00 20:00 20. Sep 04:00 08:00 12:00

— 10.0.3.60 — 10.0.3.50 — 10.0.3.70 — 10.0.3.71

Advisor Description (collapsible)

ClusterControl custom advisor

Developer Studio

It is possible to extend the functionality of ClusterControl by writing your own advisors in the ClusterControl DSL (Domain Specific Language). The DSL syntax is based on JavaScript, with extensions to provide access to ClusterControl's internal data structures and functions. The DSL allows you to execute SQL statements, run shell commands/programs across all your cluster hosts, and retrieve results to be processed for advisors/alerts or any other actions.

So, the ops team can create new advisors right within a web browser using the Developer Studio. The ClusterControl Developer Studio is a simple and elegant development environment to quickly create, edit, compile, run, test, debug and schedule your JavaScript programs.

```
1 #include "common/mysql_helper.js"
2
3 /**
4  * Checks the percentage of max ever used connections
5  *
6  */
7 var WARNING_THRESHOLD=0;
8 var TITLE="title";
9 var ADVICE_WARNING="warning advice.";
10 var ADVICE_OK="message if all is ok." ;
11
12 function main()
13 {
14     var hosts      = cluster::galeraNodes();
15     var advisorMap = {};
16
17     for (idx = 0; idx < hosts.size(); ++idx)
18     {
19         host      = hosts[idx];
20         map       = host.toMap();
21         connected = map["connected"];
22         var advice = new CmonAdvice();
23
24         if(!connected)
25             continue;
26         if(checkPrecond(host))
27         {
28             if(threshold > WARNING_THRESHOLD)
29                 advice.setJustification("if there is a warning, justify why.");
30             else
31
```

ClusterControl developer studio

Command Line Interface (CLI)

ClusterControl CLI (or s9s CLI), is a command line tool to interact, control and manage your database infrastructure using ClusterControl. The s9s command line project is open source and can be found on GitHub at <https://github.com/severalnines/s9s-tools>.

By default, the s9s CLI is installed on the ClusterControl host by the installer script and it can also work remotely outside of the ClusterControl server (on your workstation laptop or bastion host), as long as the controller's RPC interface is reachable to the client network (RPC interface is default to 127.0.0.1:9501). It provides an easy terminal interface to the Clustercontrol RPC v2 API and the communication between this client and CMON controller is encrypted and secure through TLS. You will find it very useful when working with large scale deployments and allow you to design more complex features and workflows.

To highlight some of its capabilities, the following are the supported features up until version 1.6:

- Database clusters and load balancers deployment:
 - MySQL - Standalone, MySQL Replication or Galera Cluster
 - PostgreSQL - Standalone or streaming replication
 - MongoDB - Replica Set
 - Load balancers - HAProxy and ProxySQL
 - Import existing MySQL, PostgreSQL and MongoDB clusters
- Database and host monitoring:
 - Status of nodes and clusters
 - Host stats and database stats
 - Graphs within terminal
 - Process monitoring
- Cluster and node management:
 - Create, stop or start clusters
 - Add, remove, or restart nodes in the cluster
 - Schema and user management
 - Configuration management
 - Maintenance mode management
- Backup management:
 - Create and schedule backups
 - Restore backups
- Cloud/virtualization management:
 - Manage cloud instances on AWS
 - Manage containers on LXC

- Job monitoring and management:
 - Monitor job progress
 - View the job log
 - Delete and schedule a job

Actions you take from the CLI will be visible in the ClusterControl Web UI and vice versa. The ClusterControl CLI and GUI are fully integrated and synced to allow you to utilize the CLI for deployment and management of your databases and load balancers, whilst using the advanced graphs in the GUI for monitoring and troubleshooting.

The command line tool is invoked by executing a binary called `s9s`. The commands are basically JSON messages being sent over to the ClusterControl Controller (CMON) RPC interface. The command line client installs manual pages and can be viewed by entering the command:

```
1 | $ man s9s
2 | $ s9s --help
```

Here are some of the example commands that you can use to operate your database cluster from `s9s` command line:

Deployment

Deploy a three-node Percona XtraDB Cluster 5.7 cluster, with OS user `vagrant`:

```
1 | $ s9s cluster --create \
2 | --cluster-type=galera \
3 | --nodes="10.10.10.10;10.10.10.11;10.10.10.12" \
4 | --vendor=percona \
5 | --provider-version=5.7 \
6 | --db-admin-passwd='pa$$word' \
7 | --os-user=vagrant \
8 | --cluster-name='Percona XtraDB Cluster 5.7'
```

Deploy a MongoDB Sharded Cluster with 3 mongos, 3 mongo config and one shard consists of a three-node replica set called 'replset2' with different priority for each node:

```
1 | $ s9s cluster --create \
2 | --cluster-type=mongodb \
3 | --vendor=10gen \
4 | --provider-version=3.2 \
5 | --db-admin=adminuser \
6 | --db-admin-passwd=adminpwd \
7 | --nodes="mongos://192.168.1.11;mongos://192.168.1.12;mon-
  gos://192.168.1.12;mongocfg://192.168.1.11;mongocf-
  g://192.168.1.12;mongocfg://192.168.1.13;192.168.1.14?prior-
  ity=5.0;192.168.1.15?arbiter_only=true;192.168.1.16?priority
  =2;192.168.1.17?rs=replset2;192.168.1.18?rs=replset2&arbi-
  ter_only=yes;192.168.1.19?rs=replset2&slave_delay=3&priori-
  ty=0"
```

Import and existing Percona XtraDB Cluster 5.7 and let the deployment job running in foreground (provided passwordless SSH from ClusterControl node to all database nodes have been setup correctly):

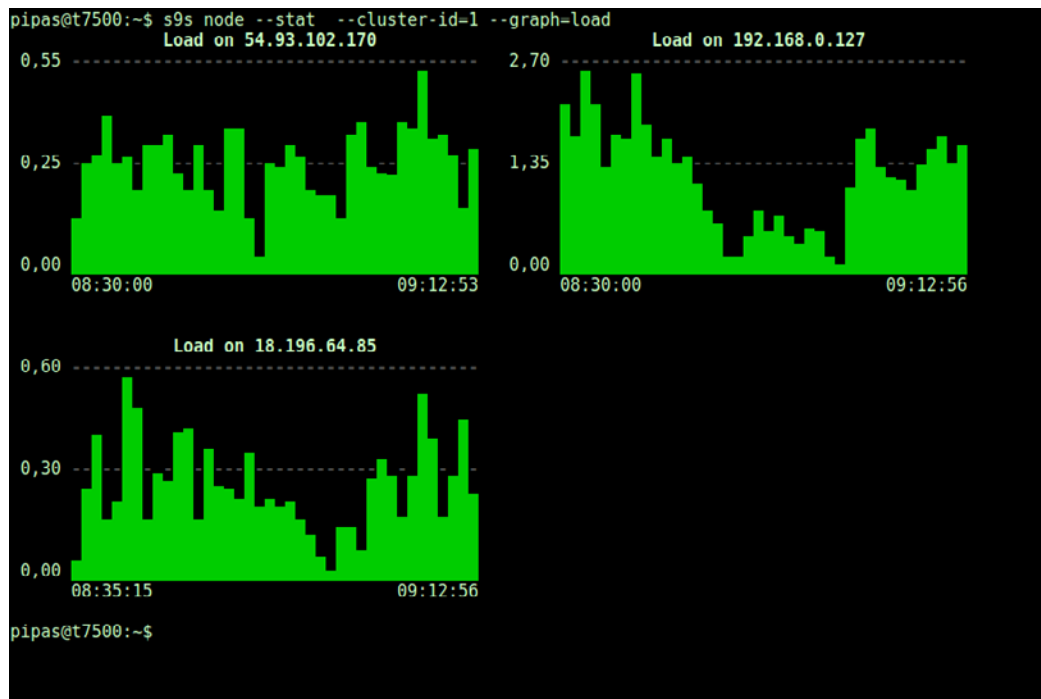
```
1 $ s9s cluster --register \  
2 --cluster-type=galera \  
3 --nodes="192.168.100.34;192.168.100.35;192.168.100.36" \  
4 --vendor=percona \  
5 --provider-version=5.7 \  
6 --db-admin="root" \  
7 --db-admin-passwd='My&2d$w0r@d' \  
8 --os-user=root \  
9 --cluster-name="PXC Prod 1 - GCP" \  
10 --wait
```

Monitoring

Get a summarized view of all nodes:

```
1 $ s9s node --list --long  
2 STAT VERSION CID CLUSTER HOST PORT  
3 COMMENT  
4 coC- 1.6.2.2662 23 PostgreSQL 10 10.0.0.156 9500  
5 Up and running  
6 poM- 10.4 23 PostgreSQL 10 10.0.0.44 5432  
7 Up and running  
8 poS- 10.4 23 PostgreSQL 10 10.0.0.58 5432  
9 Up and running  
10 poS- 10.4 23 PostgreSQL 10 10.0.0.60 5432  
11 Up and running  
12 soS- 5.7.23-log 24 Oracle 5.7 Replication 10.0.0.104 3306  
13 Up and running.  
14 coC- 1.6.2.2662 24 Oracle 5.7 Replication 10.0.0.156 9500  
15 Up and running  
16 soM- 5.7.23-log 24 Oracle 5.7 Replication 10.0.0.168 3306  
17 Up and running.  
18 mo-- 3.2.20 25 MongoDB 3.6 10.0.0.125 27017  
19 Up and Running  
20 mo-- 3.2.20 25 MongoDB 3.6 10.0.0.131 27017  
21 Up and Running  
22 coC- 1.6.2.2662 25 MongoDB 3.6 10.0.0.156 9500  
23 Up and running  
24 mo-- 3.2.20 25 MongoDB 3.6 10.0.0.35 27017  
25 Up and Running  
26 Total: 11
```

Shows server load histogram in graph format for cluster ID 1:



ClusterControl CLI

Scaling

Add a database node to an existing MongoDB Sharded Cluster with cluster ID 12 having replicaset name 'replset2':

```
1 | $ s9s cluster --add-node \  
2 | --cluster-id=12 \  
3 | --nodes=mongodb://192.168.1.20?rs=replset2
```

Remove a database node from cluster ID 1 as a background job:

```
1 | $ s9s cluster --remove-node \  
2 | --nodes=10.10.10.13 \  
3 | --cluster-id=1
```

Management

Schedule a full backup using MariaDB backup every midnight at 12:00 AM:

```
1 | $ s9s backup --create \  
2 | --backup-method=mariabackupfull \  
3 | --nodes=10.10.10.19:3306 \  
4 | --cluster-name=MDB101 \  
5 | --backup-dir=/home/vagrant/backups \  
6 | --recurrence='0 0 * * *'
```

Push a configuration option inside my.cnf (max_connections=500) on node 10.0.0.3:

```
1 | $ s9s node --change-config \  
2 | --nodes=10.0.0.3 \  
3 | --opt-group=mysqlid \  
4 | --opt-name=max_connections \  
5 | --opt-value=500
```

Restart the database service on node 192.168.1.117 for cluster ID 1:

```
1 | $ s9s node --restart \  
2 | --cluster-id=1 \  
3 | --nodes=192.168.1.117 \  
4 | --log
```

Schedule a rolling restart of the cluster 20 minutes from now:

```
1 | $ s9s cluster --rolling-restart \  
2 | --cluster-id=1 \  
3 | --schedule="$(date -d 'now + 20 min')"
```

Check out the [ClusterControl CLI documentation page](#) for more details and examples.

Conclusion

ClusterControl provides a unified software platform to deploy and run state of the art open source database infrastructures. It can be of great assistance to existing DBAs who wish to automate a number of mundane but time consuming tasks, and 'scale' themselves by providing higher value services to their organizations - e.g., spend more time on architecture and design, help developers write scalable database applications, optimize database code, work on capacity planning and other activities that make a bigger impact on their organization.

ClusterControl can also assist operations teams who do not have a DBA on staff. Not many organizations require or are able to employ full-time DBAs. The databases are often managed by devops or system administrators. But database administration is a specialized role. The bar for the DBA role is high, as there's enough going on to make anyone's head spin - solution design, configuration, resilience, troubleshooting and maintenance updates, application clustering, database clustering, storage clustering, network bonding, server load balancing and traffic management, file replication and clustering, database replication, monitoring, split brain prevention, site to site failovers, data integrity, security,... the list goes on. So having a platform that can automate most of the operational management of the database environment can help operations teams bridge the knowledge gaps and deliver a stable and reliable environment to the business.

Making ClusterControl, as we like to describe it, the only management system anyone will ever need to take control of their open source database infrastructure.



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>

Related Resources



Download

Register and access the installation script to download ClusterControl.



Documentation

Visit the ClusterControl Documentation, including the technical User Guide, for more in-depth details on each aspect of the product.



Support

Access our Community Support area and our Technical Support Team for Enterprise trial users and customers.



Plans & Pricing

Find out which ClusterControl plan works best for you, whether it be Community, Advanced or Enterprise.

ClusterControl differs from the usual approach of trying to bolt together performance monitoring, automatic failover and backup management tools by combining – in one product – everything you need to deploy and operate mission-critical databases in production. This Guide explains why ClusterControl is the only database management system you'll ever need to take control of your open source database infrastructure.



Deploy



Manage



Monitor



Scale