

MariaDB Galera Cluster @OracleLinux9.5 using ClusterControl install-cc script with limited Internet connection (via repository proxy aka satellite server).

Download Oracle Linux distro and connect it to Repository Server, as described here.:

- (2025-02-12 -- ClusterControl is only compatible with x86_64 systems)

Install OS in minimal mode, !without GUI.

<https://yum.oracle.com/oracle-linux-isos.html>

For example, OracleLinux-R9-U5-x86_64-dvd.iso

HLD (High-Level Design)

(one VM is connected to Internet)

lt58ncp1sat1 - Repository satellite

(others are NOT connected to Internet):

lt58ncp1dbm1 - Monitoring, ClusterControl

lt58ncp1dbn1 - Node 1, MariaDB Galera Cluster

lt58ncp1dbn2 - Node 2, MariaDB Galera Cluster

lt58ncp1dbn3 - Node 3, MariaDB Galera Cluster

Preparations:

- ensure NO cockpit service running, it occupies port 9090, same as Prometheus uses. Or, if required, change its listening port.

```
systemctl status cockpit
systemctl stop cockpit
dnf remove cockpit*
```

Install utlis

```
dnf install \
  tmux \
  wget
```

Firewall with firewalld on ClusterControl

```
systemctl enable firewalld
systemctl start firewalld
systemctl status firewalld
firewall-cmd --add-service=http --permanent
firewall-cmd --add-service=https --permanent
firewall-cmd --add-service=prometheus --permanent
firewall-cmd --reload
firewall-cmd --list-all
```

Temporary disable SELinux for installation, it will be enabled later

```
sed -i 's|SELINUX=enforcing|SELINUX=disabled|g' /etc/selinux/config
setenforce 0
getenforce
```

Configuring repositories

Add repositories to all VMs which point to repository satellite. Configure DNS for the host locally, if needed.

```
ping lt58ncp1sat1
vi /etc/hosts
```

```
192.168.56.109 lt58ncp1sat1
```

```
ping lt58ncp1sat1
curl http://lt58ncp1sat1/hello
rm /etc/yum.repos.d/*
vi /etc/yum.repos.d/lt58ncp1sat1.repo
```

Refer to config file on another page.

Check, update and reboot.

```
dnf repolist
dnf update
shutdown -r now
```

Login

```
tmux
sudo su
```

On the day of writing (2025-02-13), there is a transition period in caused by renamed commands in the scripts (`mysql` and `mariadb`). To resolve it, additional tricks needed to make the script work (and keep installations script integrity).

```
ln -s /usr/bin/mariadb      /usr/bin/mysql
ln -s /usr/sbin/mariabdb   /usr/bin/mysqld
ln -s /usr/bin/mariadb-admin /usr/bin/mysqladmin
ln -s /usr/bin/mariadb-install-db /usr/bin/mysql_install_db
```

Offline installation

Install and enable MariaDB manually

```
dnf install \
  MariaDB-client \
  MariaDB-common \
  MariaDB-server \
  MariaDB-shared
systemctl enable mariadb
```

```
systemctl start mariadb
systemctl status mariadb
```

Download and transfer installation script to the destination machine.

```
wget http://www.severalnines.com/downloads/cmon/install-cc
chmod +x ./install-cc
# OFFLINE=true HOST=192.168.10.211 ./install-cc
OFFLINE=true ./install-cc
```

Define bind address to config file and restart the service

```
vi /etc/default/cmon
```

add line, replacing with your IP address

```
RPC_BIND_ADDRESSES="127.0.0.1,192.168.10.211"
```

```
systemctl restart cmon*
```

Check that instance is running binded to local address to facilitate activation:

```
ps aux | grep cmon
```

```
root    42467  0.0  0.0 1232048 7812 ?        Ssl  14:43   0:00 /usr/share/cmon-ssh/cmon-ssh
root    42472  0.3  0.4 1295704 51432 ?        Ssl  14:43   0:00 /usr/sbin/cmon-cloud -log_file /var/log/cmon-
cloud.log
root    42475  0.0  0.0 1233492 10152 ?        Ssl  14:43   0:00 /usr/sbin/cmon-events
root    42633  0.4  0.3 567268 36516 ?        Ssl  14:43   0:00 /usr/sbin/cmon --rpc-port=9500 --bind-
addr=127.0.0.1,192.168.10.211 --events-client=http://127.0.0.1:9510 --cloud-service=http://127.0.0.1:9518
root    42687  0.0  0.0  6408  2308 pts/2  S+   14:43   0:00 grep --color=auto cmon
```

Check that firewall is stopped or rules a specified for activation. Enable after activation:

```
firewall-cmd --add-port=9500/tcp
firewall-cmd --add-port=9501/tcp
firewall-cmd --add-port=9510/tcp
firewall-cmd --reload
firewall-cmd --list-all
systemctl stop firewalld
systemctl status firewalld
```

Check from CLI that API is accessible

```
curl http://127.0.0.1:9500/0/settings
```

```
<!DOCTYPE html>  
<html lang="en">  
  <head>
```

Note down password in KeepAss, as usual.

Send telemetry [N]

MariDB root pass?

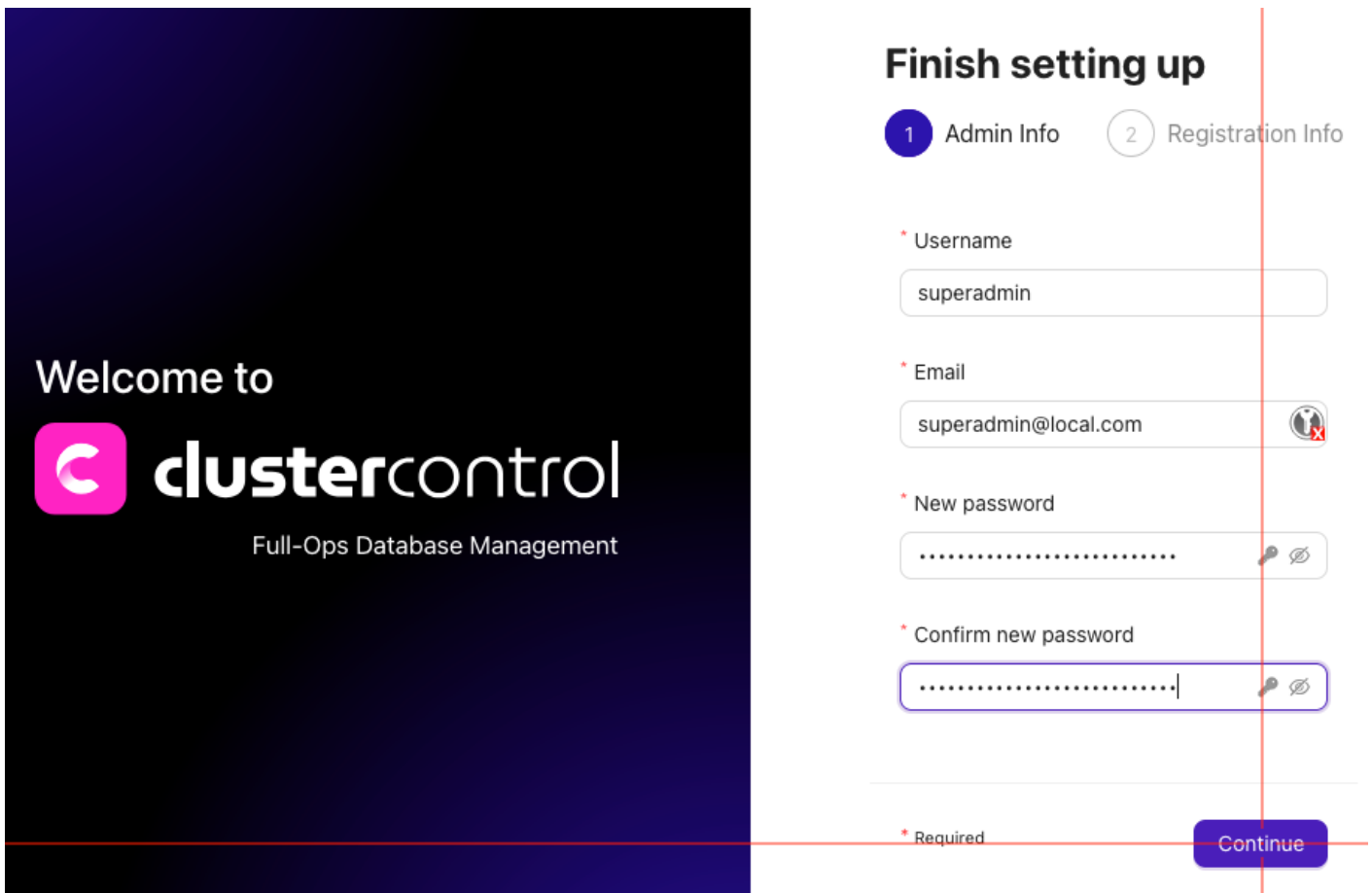
MariDB cmon pass?

Open your web browser to <https://192.168.56.107> and create a default Admin User.

Open in the browser

```
firefox https://192.168.56.107
```

Create admin user, note down pass in password manager.



The screenshot shows the 'Finish setting up' screen of the ClusterControl installation. On the left, there is a dark blue banner with the ClusterControl logo and the text 'Welcome to clustercontrol Full-Ops Database Management'. On the right, the 'Admin Info' step is active, showing four required fields: Username (filled with 'superadmin'), Email (filled with 'superadmin@local.com'), New password (masked with dots), and Confirm new password (masked with dots). A 'Continue' button is at the bottom right. A red vertical line is drawn through the form, and a red horizontal line is drawn at the bottom.

Choose 'Community', unless license owned.



Finish setting up

1 Admin Info 2 Registration Info

* First name

Super

* Last name

Admin

* Company name

BestCompany

* Phone number

+1234567890

* Plan of interest

Community

I've read and agree to the [Terms and Conditions](#) and the [Privacy Policy](#)

* Required

Back

Complete

There is a trial license activated



Trial License Failed

The trial license failed to be activated! You will be using the Community Edition with limited features. Please contact sales@severalnines.com to request a trial license which will unlock all features to evaluate for a period of time

Continue

Before cluster will be created, nodes need to be prepared. Stop here.

Node configuration

```
sudo su
```

Perform repositories configuration and update as for cluster node in the beginning.

Remember to perform post-cloning actions to change MAC address, make host unique and clean up SSH keys,

Prepare database storage for cluster management and nodes. To make path identical for all nodes, symbolic link will be created which will be used to configure other applications (MariaDB in this case).

```
export host="$(hostname)"
df -h /mnt/$(hostname)-data/
ln -s /mnt/$(hostname)-data/ /mnt/data
ls -la /mnt/

mkdir -p /mnt/data/mariadb/clusters/ncp/
chown -R mysql:mysql /mnt/data/mariadb/
namei -mo /mnt/data/mariadb/clusters/ncp/
# ? TODO: selinux context
```

Firewall with firewalld on the cluster nodes

```
systemctl enable firewalld
systemctl start firewalld
systemctl status firewalld
firewall-cmd --add-service=mysql --permanent
firewall-cmd --reload
```

Manually install MariaDB server to the node and let ClusterControl configure it. Otherwise, ClusterControl will automatically add repositories to nodes (that we want to avoid and use only specified ones).

```
dnf install \
  MariaDB-server \
  MariaDB-client \
  MariaDB-common \
  MariaDB-backup \
  galera-4

systemctl enable mariadb
```

```
systemctl start mariadb
systemctl status mariadb
```

Deploy new cluster via WebUI

Post-installation is necessary to give permissions to ClusterControl to login into nodes to perform actions (deploy the cluster). Root user as per documentation, but any other user with enough privileges can do.

```
sudo su
whoami
cd
ssh-keygen -t ed25519
ssh 0
exit
ls -la .ssh
cat .ssh/known_hosts
```

Copy public key to the nodes and itself (replace hostnames)

```
ssh-copy-id -i ~/.ssh/id_ed25519 root@lt58ncp1dbm1
ssh-copy-id -i ~/.ssh/id_ed25519 root@lt58ncp1dbn1
ssh-copy-id -i ~/.ssh/id_ed25519 root@lt58ncp1dbn2
ssh-copy-id -i ~/.ssh/id_ed25519 root@lt58ncp1dbn3
```

Remember to create symbolic links to new mariadb executables to ensure deployment scripts are working.

Deploy new cluster from ClusterControl dashboard

clustercontrol

Home / Clusters

Deploy a cluster

Super Admin

Home

Clusters

Nodes

Backups

Activity center

Operational reports

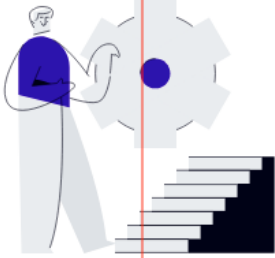
User management

Settings

Clusters

Filters

Sort by: Cluster ID



You haven't created any clusters. When you do, it'll show up here.

Deploy a cluster

Confirm pressing [Continue]

Deploy a cluster

Deploy a cluster managed by ClusterControl's virtual DBA. Monitor failures, automate backups, node and cluster recovery.



Create a database cluster

Choose a database technology, configure and create an open source database service within a few minutes.

Supported databases: Elasticsearch, Valkey, SQL Server, Redis, Redis Sentinel, MySQL (Primary/Replica), MySQL Galera, PostgreSQL (Primary/Replica), TimescaleDB (Primary/Replica), MongoDB ReplicaSet and MongoDB Shards

Continue

Choose "Database: MySQL Galera, Vendor MariaDB and the version"

Deploy cluster



* Database

 MySQL Galera

* Vendor

 MariaDB

* Version ⓘ

11.4

- 11.4
- 10.11
- 10.6
- 10.5
- 10.4



MariaDB Galera

Database: Vendor: Version:

MySQL Galera MariaDB 11.4

Description:

Galera Cluster for MySQL is a true Multi-Master Cluster based on synchronous replication. It's an easy-to-use, high-availability solution, which provides high system up-time, no data loss and scalability for future growth.

[Learn more](#)

Back

Continue

Give cluster a name

Deploy MySQL Galera cluster



1 Cluster details

2 SSH configuration

3 Node configuration

4 Add nodes

5 Preview

Name your cluster

Name ⓘ (optional)

ncpl

Leave empty and we will generate one for you.

Tags

Type to add tags

Add tags to search or group your database clusters

* Required

Cancel

Continue

Provide SSH credentials, disable 'Install software', as script will enable repositories on remote hosts to fetch packages from Internet. Disable SELinux/AppArmor for installation time. It will be enabled later in security hardening.

Deploy MySQL Galera cluster

- Cluster details
- 2 SSH configuration**
- 3 Node configuration
- 4 Add nodes
- 5 Preview

SSH Credentials

* SSH user ⓘ

* SSH port ⓘ

* SSH user key path ⓘ

SSH sudo password ⓘ

Install software Off

Security configuration

Disable firewall

Disable SELinux/AppArmor

* Required

Provide node configuration details. Ensure, that database storage location is specified correctly.

```
# as per default
/var/lib/mysql

# for mounted as per instructions above
/mnt/data/mariadb/clusters/ncp/
```

Deploy MySQL Galera cluster



✓ Cluster details

✓ SSH configuration

3 Node configuration

4 Add nodes

5 Preview

Node configuration

* Server port ⓘ

3306

* Server data directory ⓘ

/var/lib/mysql

* Admin/Root user ⓘ

root

* Admin/Root password ⓘ

.....

* Repository ⓘ

Use vendor repositories

* Version ⓘ

11.4

Setup and use the vendor's repositories.

[Learn more](#) about vendor repositories.

Type to search

Configuration template ⓘ

my.cnf.mdb106+-galera

Enable SSL encryption ⓘ

On

* Required

Back

Continue

Add nodes, all should be green

Deploy MySQL Galera cluster



- ✓ Cluster details
- ✓ SSH configuration
- ✓ Node configuration
- 4** Add nodes
- 5 Preview

* Galera node

 +

Please note that an odd number of nodes is recommended, i.e., 3, 5, 7, etc.

192.168.56.104 ✓
Galera node

Data IP (optional)

192.168.56.105 ✓
Galera node

Data IP (optional)

192.168.56.106 ✓
Galera node

Data IP (optional)

* Required

Review config and [Finish]

Deploy MySQL Galera cluster




- ✓ Cluster details
- ✓ SSH configuration
- ✓ Node configuration
- ✓ Add nodes
- 5** Preview

Names and tags

ncp
No tags

Vendor and version

 MariaDB Galera - 11.4

SSH configuration

SSH user: root
SSH key path: /root/.ssh/id_ed25519
SSH port: 22

Node configuration

Server port: 3306
Server data directory: /var/lib/mysql
Configuration template: my.cnf.mdb106+-galera
Admin/Root user: root
Admin/Root password: *****
Repository: Use vendor repositories
Enable SSL encryption: **Yes**

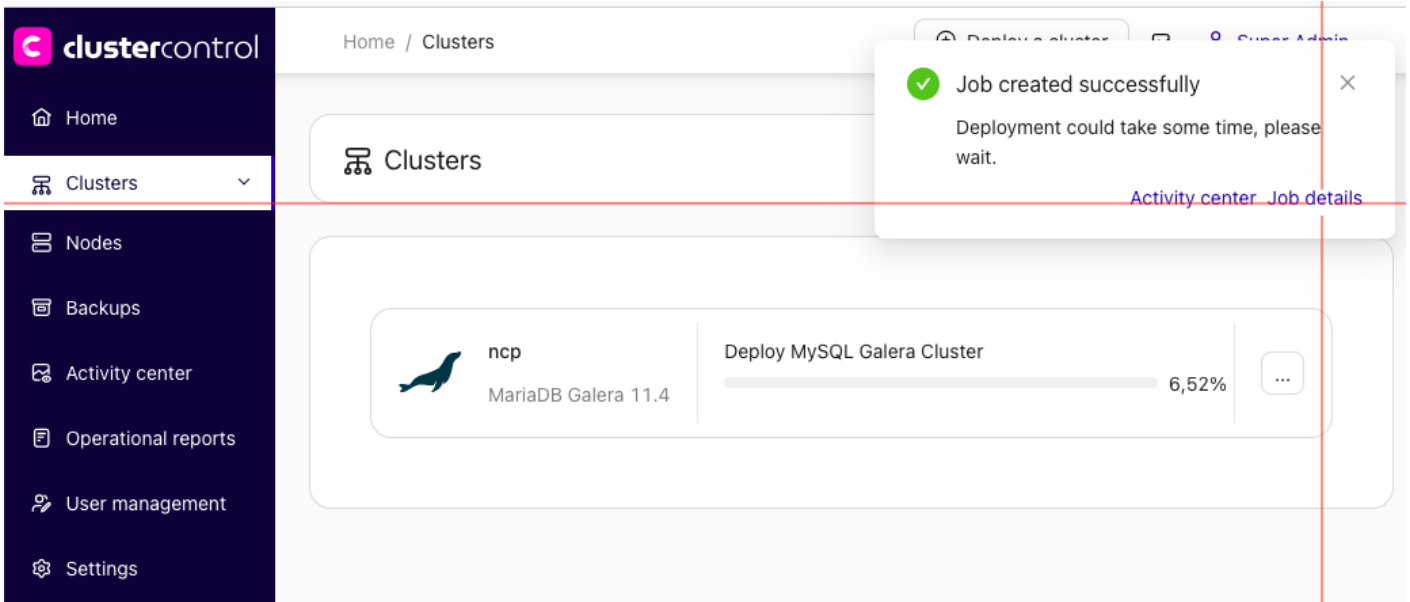
Deploying galera nodes

192.168.56.104
192.168.56.105
192.168.56.106

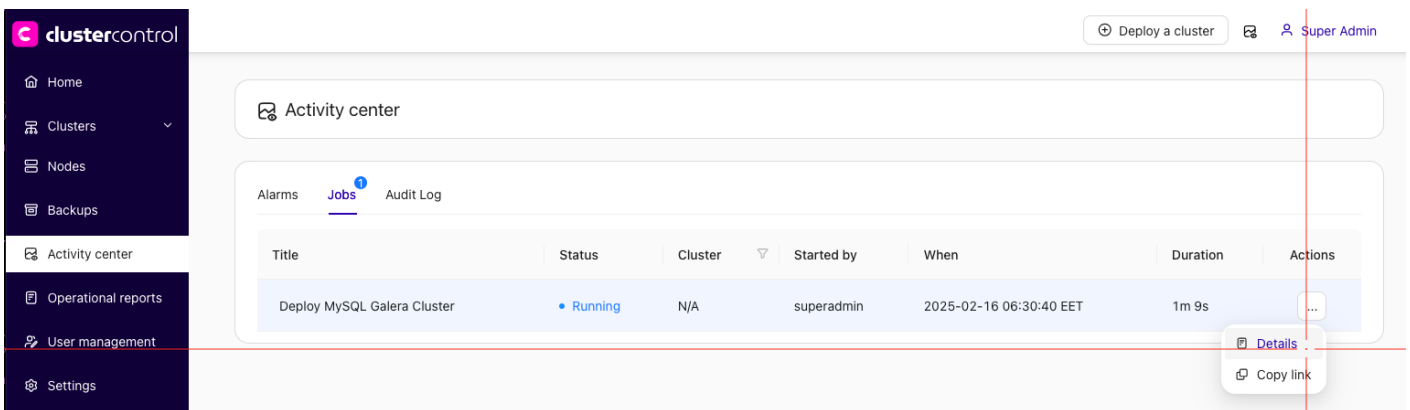
Back

Finish

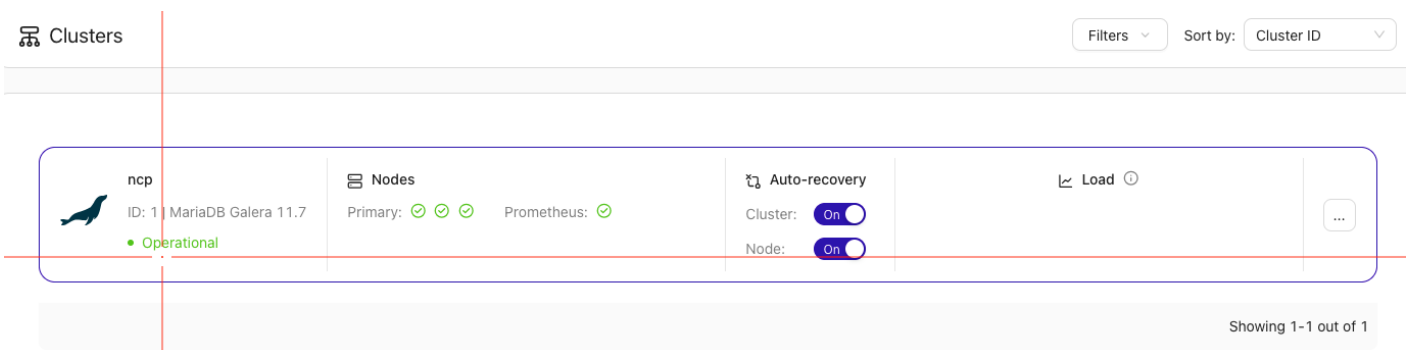
Cluster creation status can be followed from Activity Center



also accessible from Activity Center



Cluster is deployed successfully



ref.

<https://docs.severalnines.com/docs/clustercontrol/installation/offline-installation/>

In case needed, to remove MariaDB packages and databases themselves:

```
dnf remove maria*  
rm -rf /var/lib/mysql/
```

Revision #16

Created 12 February 2025 07:33:42 by Anton

Updated 18 July 2025 08:48:40 by Anton