ClusterControl

- MariaDB Galera Cluster @OracleLinux9.5 using ClusterControl install-cc script with limited Internet connection (via repository proxy aka satellite server).
- Uninstalling and or reinstalling
- Bugs found
    - New Page

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 scritps (`mysql` and `mariadb`). To rsolve 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/mariadbd          /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>
```

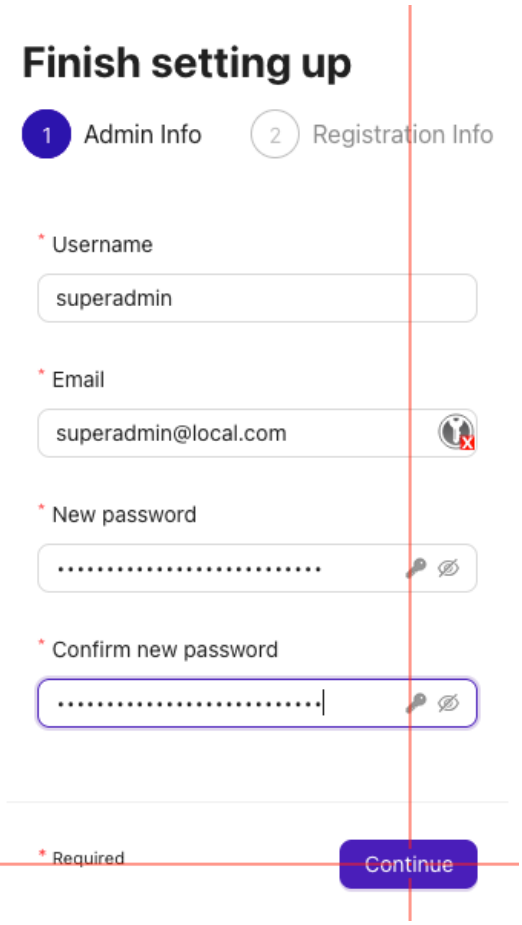
Note down password in KeepAss, as usual.

```
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.
```

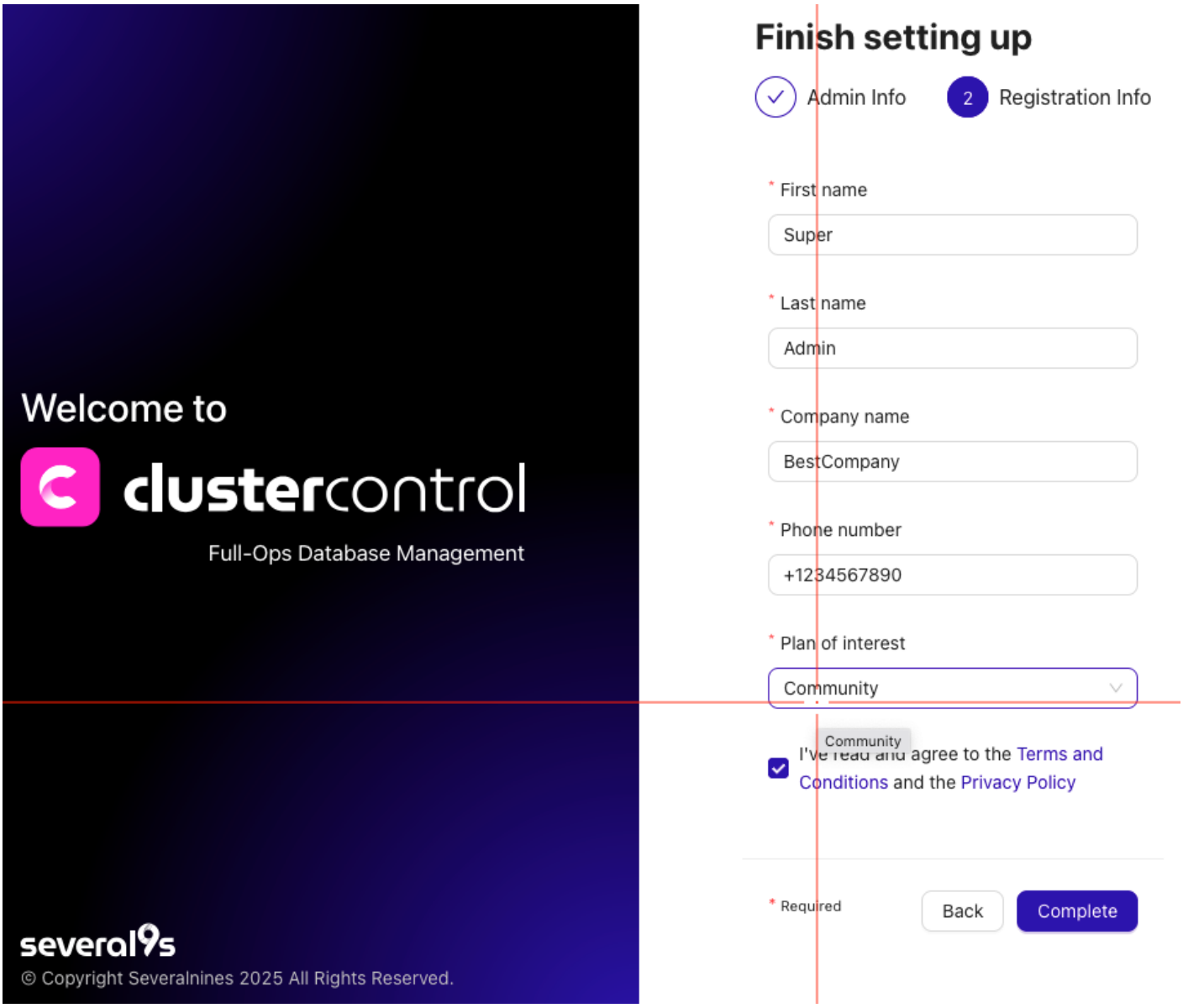
Open in the browser

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

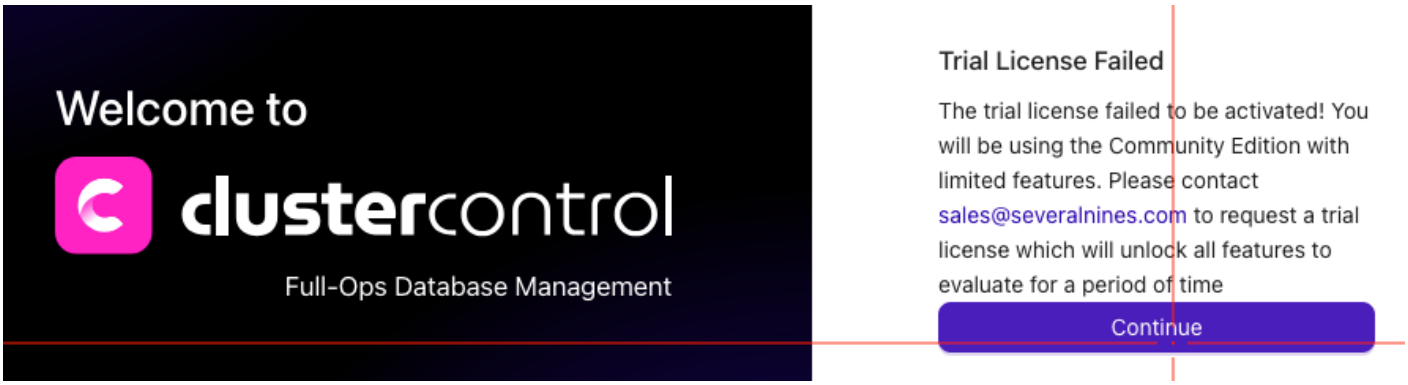
Create admin user, note down pass in password manager.



Choose 'Community', unless license owned.



There os a trial license activated



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.

During cloning the machines, change physical ID and MAC address in the hypervisor (it will not do it automatically in VirtualBox, proxmox). When machines are cloned, 'machine ID' and SSH server's keys need to be re-generated to be seen as different machines:

```
cat /etc/machine-id
systemd-machine-id-setup --commit --print


rm -rf /etc/ssh/ssh_host_*
ssh-keygen -A
ls -la /etc/ssh/ssh_host_*
```

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 peform actions (deploy the cluster). Root user as per documentation, but any other user with enough privileges can do.

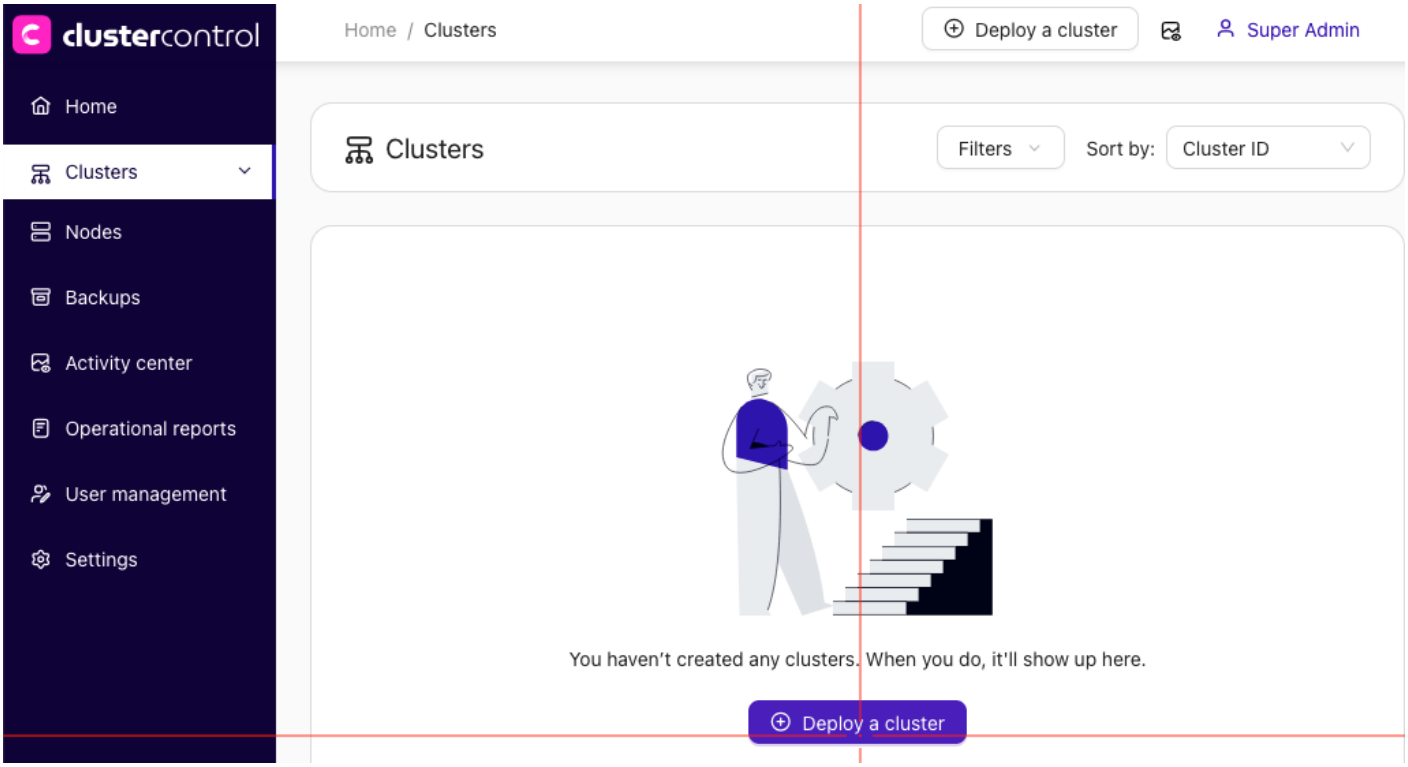
```
sudo su
whoami
cd
ssh-keygen -t ed25519
ssh 0
exit
ls -la .ssh
cat .ssh/known_hosts
```

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

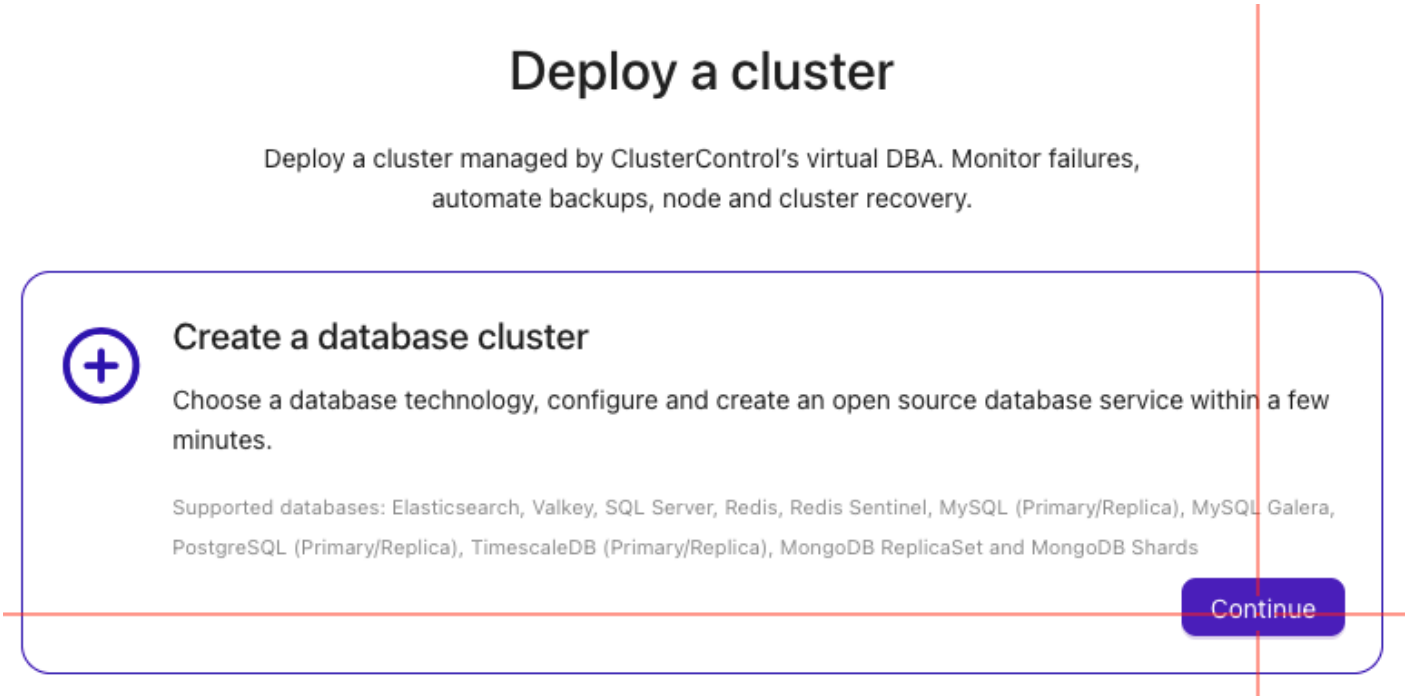
```
ssh-copy-id -i ~/.ssh/id_ed25519 root@lt58ncp1dbm1
ssh-copy-id -i ~/.ssh/id_ed25519 root@lt58ncp1dbn1
ssh-copy-id -i ~/.ssh/id_ed25519 root@lt58ncp1dbn2
ssh-copy-id -i ~/.ssh/id_ed25519 root@lt58ncp1dbn3
```

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

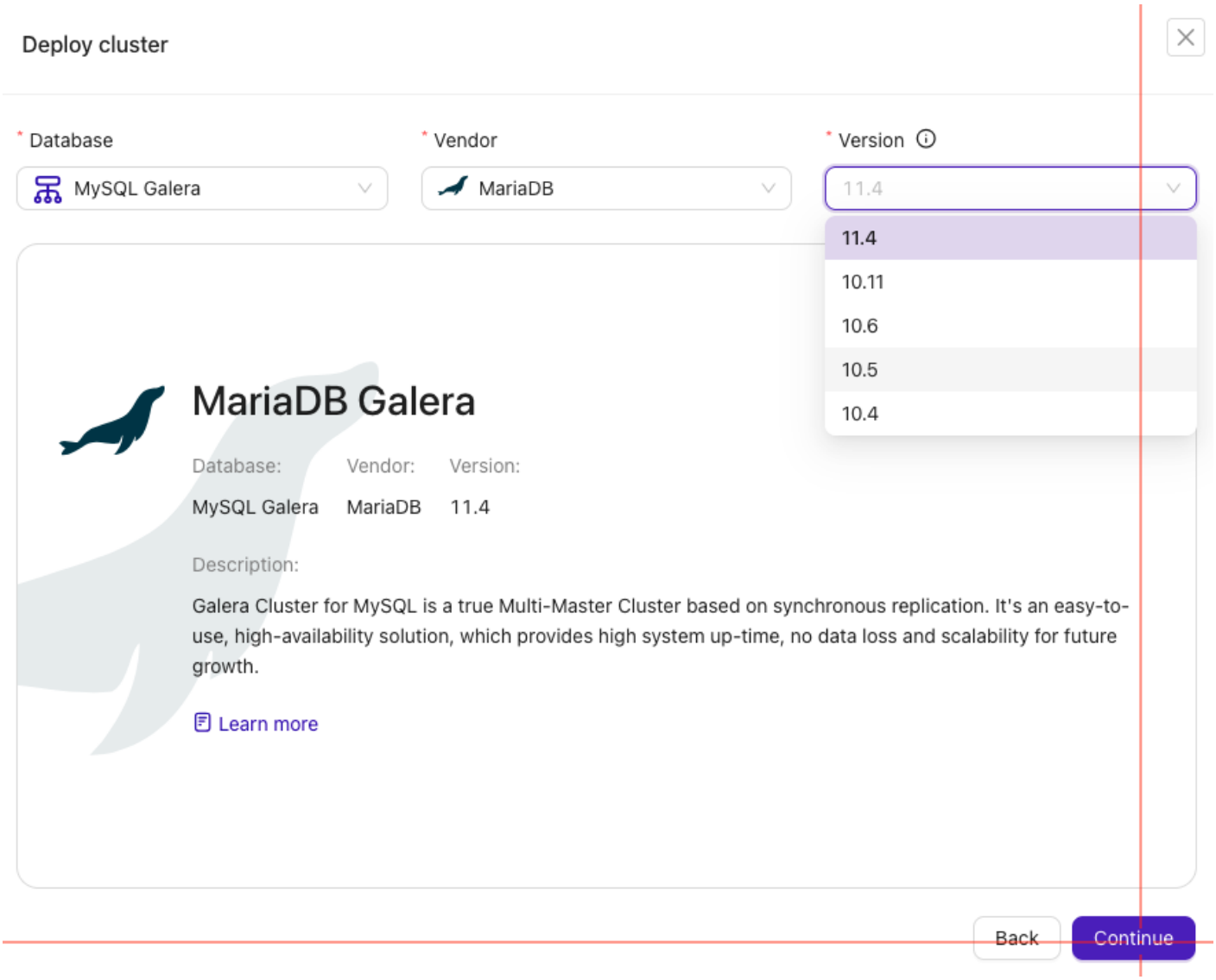
Deploy new cluster from ClusterControl dashboard



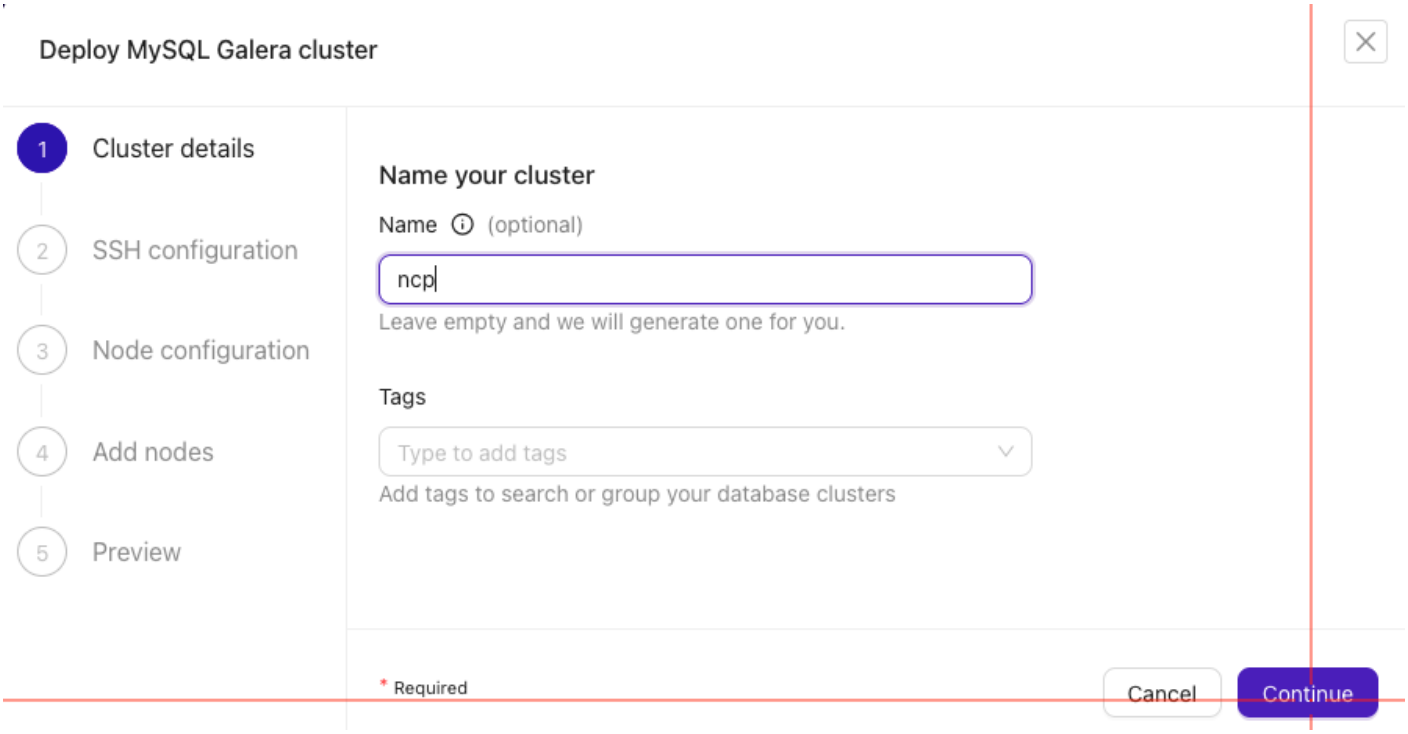
Confirm pressing [Continue]



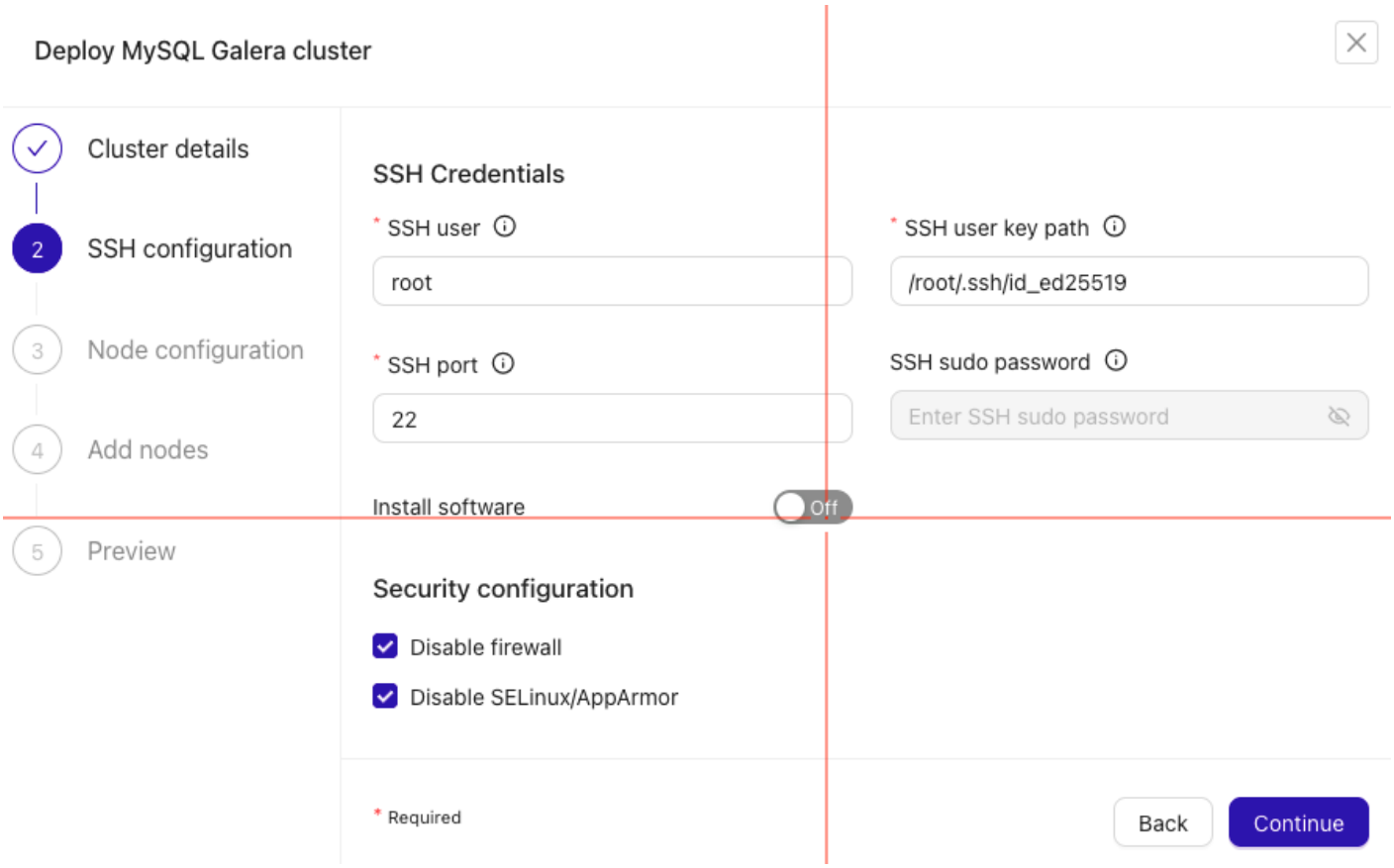
Choose "Database: MySQL Galera, Vendor MariaDB and the version"



Give cluster a name

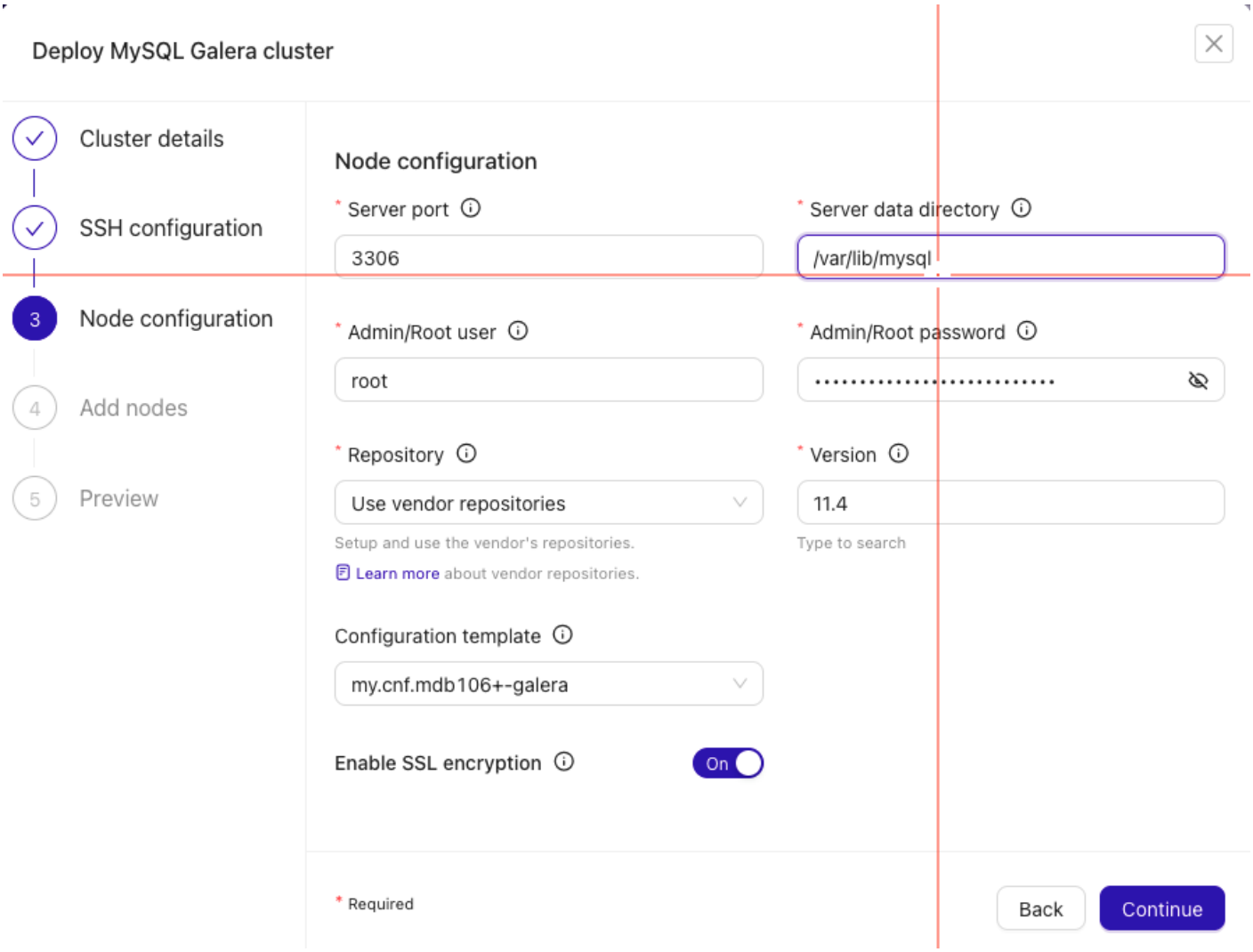


Provide SSH credentials, disable 'Install software', as script will enable repositories on remote hosts to fetch packages from Internet. Disable SELinux/AppArmor for installation time. It will be enabled later in security hardening.

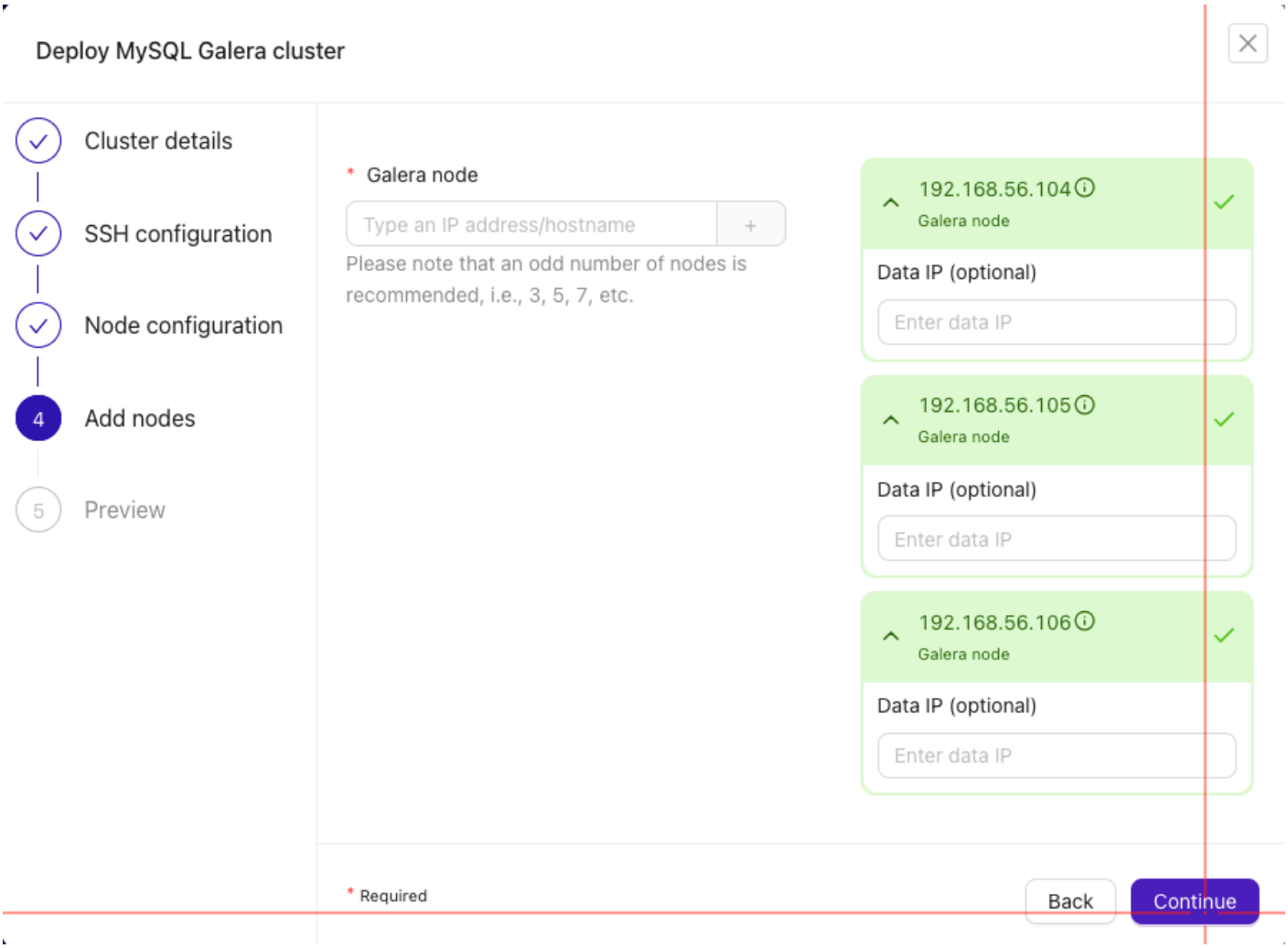


Provide node configuration details. Ensure, that database storage location is specified correctly.

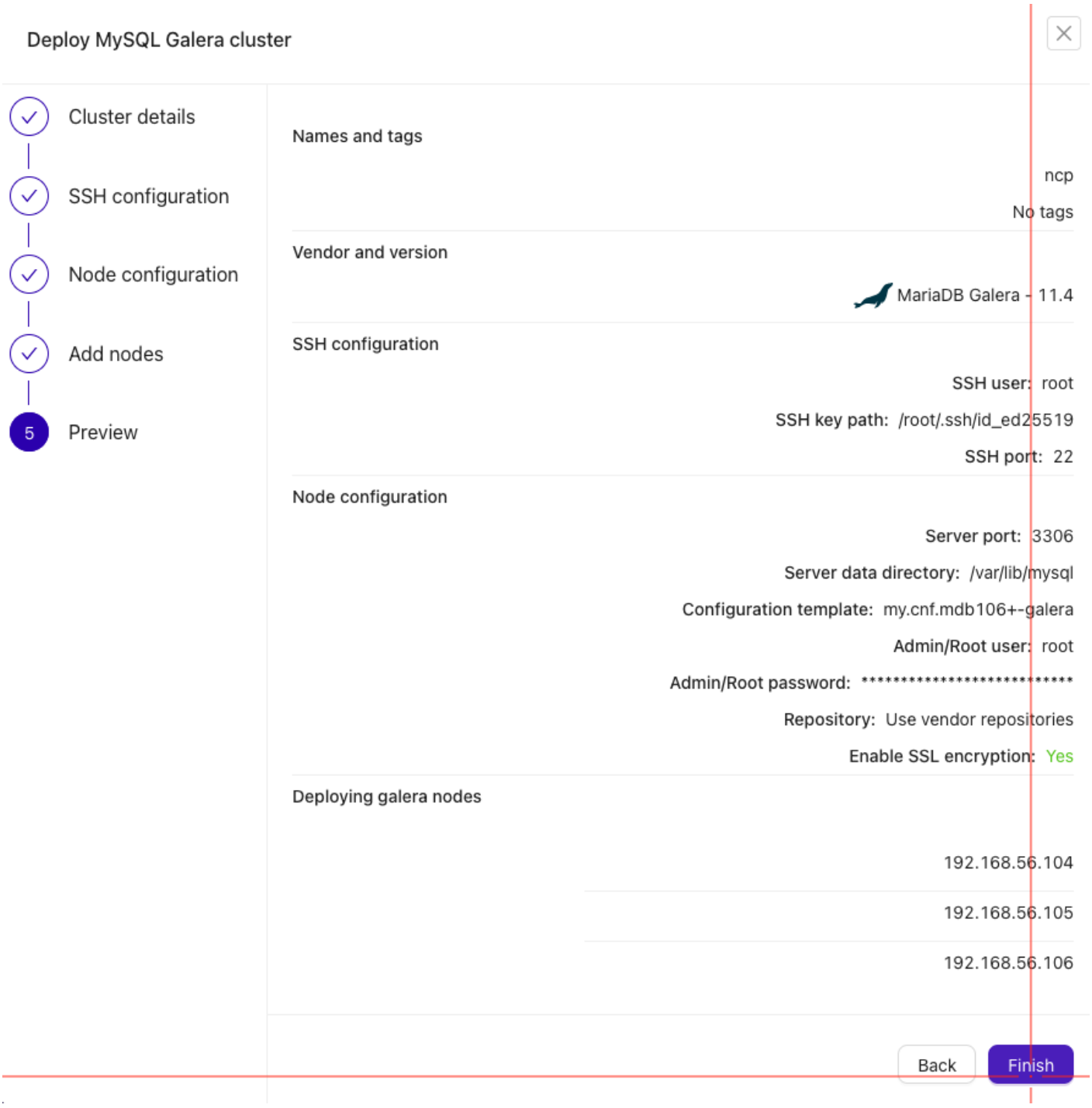
```
# as per default
/var/lib/mysql

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



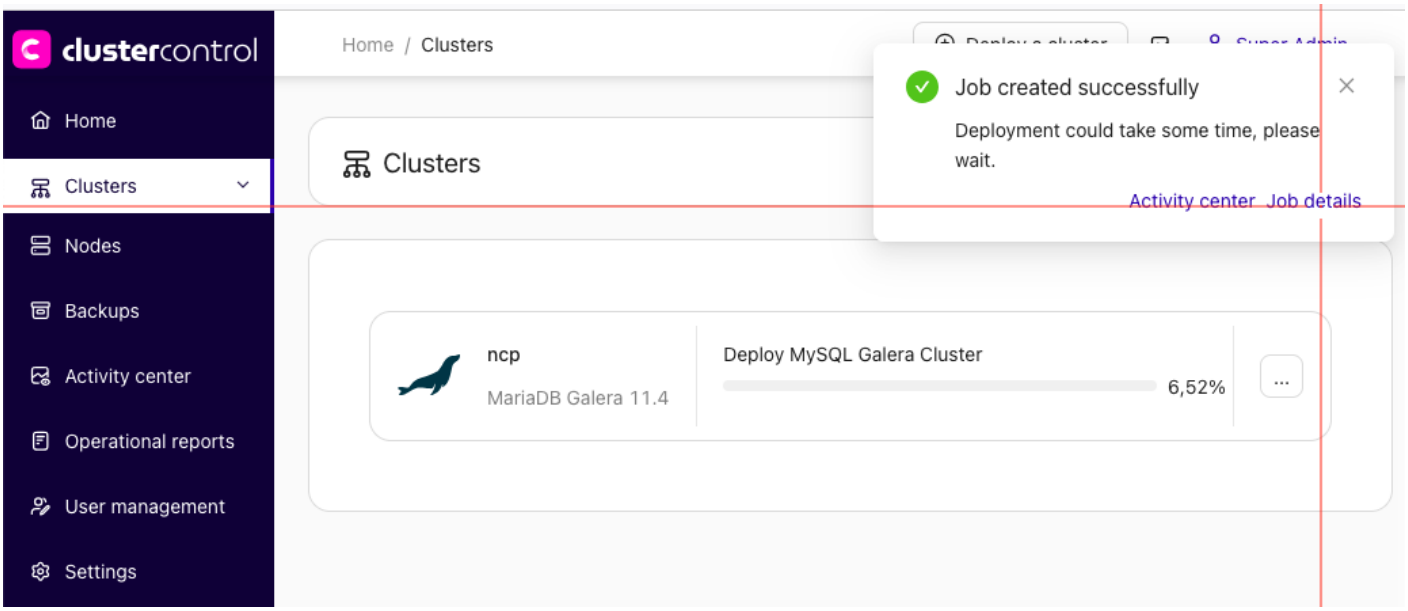
Add nodes, all should be green



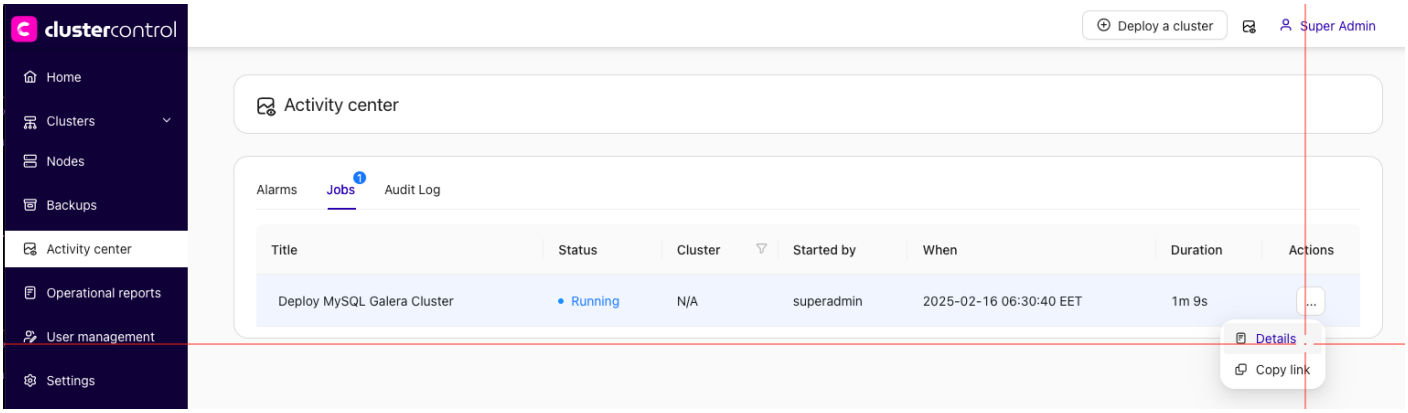
Review config and [Finish]



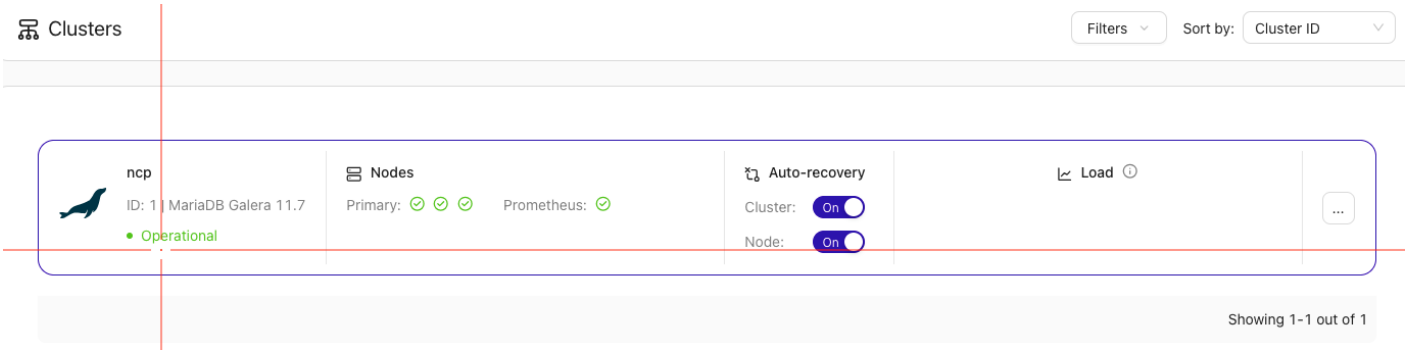
Cluster creation status can be followed from Acitivty Center



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/
```

Uninstalling and or reinstalling

Remove everything

```
sudo su
systemctl stop mariadb cmon cmon-ssh cmon-events cmon-cloud
dnf remove \
    clustercontrol* \
    s9s* \
    mariadb* \
    httpd*

rm -Rf /var/lib/mysql/
rm -Rf /var/lib/cmon/
rm -rf /var/log/cmon*
rm -Rf /etc/s9s.conf
rm -rf /etc/cmon*
rm -rf /etc/default/cmon
rm -rf /etc/prometheus/
rm -rf /usr/share/cmon/
rm -Rf /root/.s9s
rm -rf /root/s9s_tmp/
rm -rf /tmp/s9s_tmp.local.root/
rm -rf /tmp/ccsetup.conf
rm -rf /tmp/drop.sql
```

Change machine ID in HyperVisor and regenerate it

```
cat /etc/machine-id
f80*****f63b462fa14548b2b8df54c6
rm -f /etc/machine-id
systemd-machine-id-setup --commit --print
cat /etc/machine-id
472*****bf8943dc8285516244b73e8b
```

or modify several bytes in the file

```
vi /etc/machine-id
cat /etc/machine-id
4724c3a8bf8943dc8285516244b73eAA
```

Resinstall and open dashboard in private browser window

In registration window, give different credentials.

Bugs found

Bugs found

New Page

ClusterControl has outdated script for creating MaxScale load balancer.

```
[22:20:05]:
10.xxx.xxx.xxx:6603: Unknown arguments: monitor-user, monitorUser, monitor-password, monitorPassword
[22:20:05]:
10.xxx.xxx.xxx: Output Unknown arguments: monitor-user, monitorUser, monitor-password, monitorPassword .
[22:20:05]:
10.xxx.xx.xxx: Command was maxctrl create monitor galera_monitor galeramon --monitor-user=maxscale --monitor-password=xxxxxx
[22:20:05]:
10.xxx.xx.xx: Execution failed with return code 1.
[...]
[22:18:56]:
CMON version 2.3.0.11958.
```