

# Table of Contents

<b>5 Cluster GUI</b> .....	3
<b>Database Replication (DB)</b> .....	3
<b>File system replication (FS)</b> .....	3
<b>Installation and setup</b> .....	3
Settings .....	3
Installing and running Galera .....	4
Installing and running GlusterFS .....	5
<b>Master server</b> .....	6
<b>Number of nodes</b> .....	7
<b>Restart nodes</b> .....	7



# 5 Cluster GUI

Clustering improves system availability by propagating changes to different servers. In the event of a failure of one of the servers, the others remain available for operation.

Clustering dpiui2 is implemented through database and file system replication.

Clustering capability is available from version [dpiui2-2.25.9](#)

## Database Replication (DB)

Database replication is implemented using MariaDB Galera Cluster.

Galera is a database clustering solution that allows you to set up multi-master clusters using synchronous replication. Galera automatically handles the placement of data on different nodes, while allowing you to send read and write requests to any node in the cluster.

More information about Galera can be found at [official documentation](#).

## File system replication (FS)

File system replication is implemented using GlusterFS.

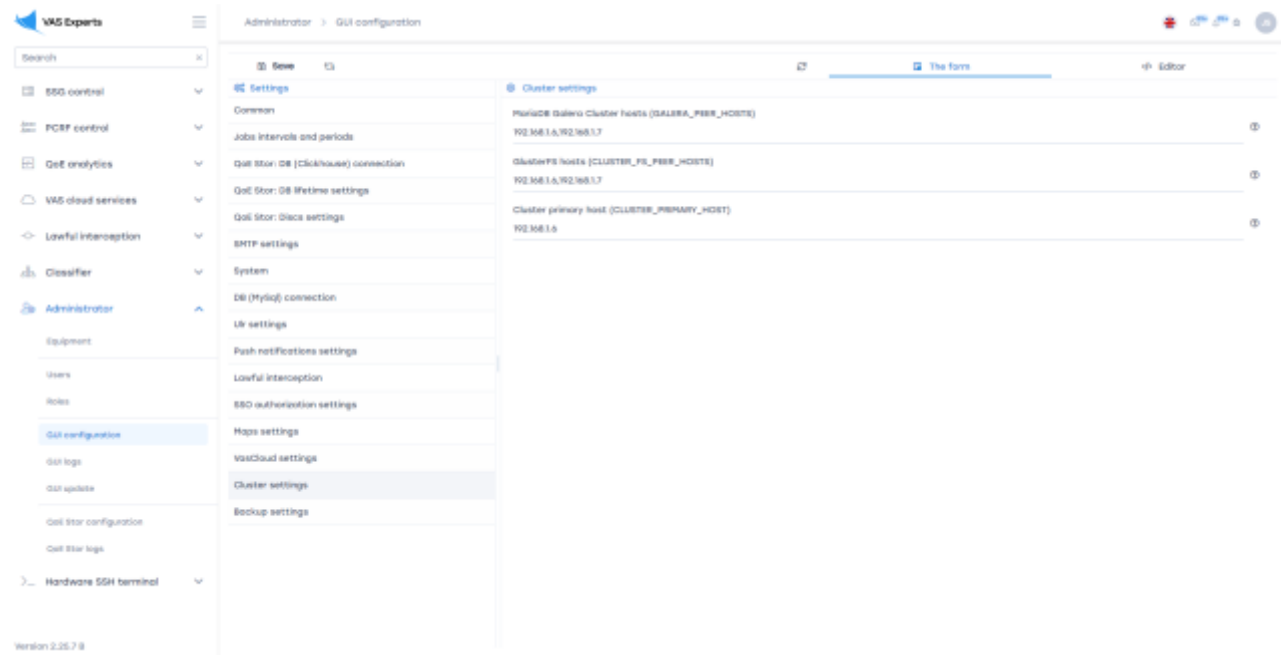
GlusterFS is a distributed, parallel, linearly scalable, fail-safe file system. GlusterFS combines data stores located on different servers into one parallel network file system. GlusterFS runs in user space using FUSE technology, so it does not require support from the operating system kernel and runs on top of existing file systems (ext3, ext4, XFS, reiserfs, etc.).

More information about GlusterFS can be found at [official documentation](#)

## Installation and setup

### Settings

All settings can be made in the dpiui2 .env file or in the GUI Configuration > Cluster Settings section.



Settings options:

**GALERA\_PEER\_HOSTS** is comma-separated list of Galera cluster hosts. The parameter determines which nodes will be available to the Galera cluster.



!Important: The main (master) node of the cluster must be placed at the beginning of the list. This is important for initial cluster deployment.

**CLUSTER\_FS\_PEER\_HOSTS** is Comma-separated list of GlusterFS cluster hosts. The parameter determines which nodes will be available to the GlusterFS cluster.



!Important: The main (master) node of the cluster must be placed at the beginning of the list. This is important for initial cluster deployment.

**CLUSTER\_PRIMARY\_HOST** is the master node for Galera and GlusterFS. The parameter defines the main node at the current moment. This parameter can be changed during operation if the main unit fails for some reason.

## Installing and running Galera

To install and start the Galera cluster, you need to run the following script under the root user on all nodes of the cluster, starting from the master node:

```
sh "/var/www/html/dpiui2/backend/app_bash/galera_setup.sh" -a init_cluster
```



!!! Important: before running the script on the master node, you need to back up the database.



! Important: before running the script, you must enter [settings](#)



! Important: there must be IP connectivity between cluster nodes.



! Important: the script must be run as root



! Important: the script must be run first on the master node



! Important: you must wait for the script to finish executing on one node before running it on the next one



! Important: set the same password for the dpiui2su user (for ssh connection) on all nodes. Enter this password in the Admin > Hardware section of the master node.

## Installing and running GlusterFS

To install and run the GlusterFS cluster, you need to follow the following steps as the root user:

**1** Execute the script **sequentially** on **all cluster nodes**:

```
sh "/var/www/html/dpiui2/backend/app_bash/glusterfs_setup.sh" -a  
init_gluster
```

The script will perform the initial installation of GlusterFS.

**2** Execute the script on the **main (master) node** (you do not need to run it on the other nodes of the cluster):

```
sh "/var/www/html/dpiui2/backend/app_bash/glusterfs_setup.sh" -a init_peers
```

The script will configure all cluster nodes.

**3** Execute the script on the **main (master) node** (you do not need to run it on the other nodes of the cluster):

```
sh "/var/www/html/dpiui2/backend/app_bash/glusterfs_setup.sh" -a init_volume
```

The script will configure the distributed storage and file system.

**4** Execute the script **sequentially** on **all cluster nodes**:

```
sh "/var/www/html/dpiui2/backend/app_bash/glusterfs_setup.sh" -a mount
```

The script will mount the replicated directories to the distributed file system.



!!! Important: before running the script on the master node, be sure to back up the `/var/www/html/dpiui2/backend/storage/` directory and `/var/www/html/dpiui2/backend/.env` file.



! Important: before running the script, you must follow [settings](#)



! Important: there must be IP connectivity between cluster nodes.



! Important: the script must be run as root



! Important: you must wait for the script to finish executing on one node before running it on the next one

## Master server

Важную роль в кластере играет Главный (мастер) сервер.

Мастер сервер устанавливается настройкой [CLUSTER\\_PRIMARY\\_HOST](#).

Мастер сервер выполняет всю фоновую работу dpiui2: взаимодействие с оборудованием, синхронизация абонентов, услуг, тарифов и т.д.

Остальные (slave) узлы не выполняют никаких фоновых действий и находятся в режиме ожидания. При этом к эти узлы доступны для работы: пользователи могут работать с этими узлами также как и с мастер сервером и не увидят разницы. Эту опцию можно использовать для балансировки нагрузки, а также для обеспечения более защищённого доступа.

При выходе из строя мастер сервера, необходимо изменить настройку `CLUSTER_PRIMARY_HOST` и назначить мастером другой сервер.

The main (master) server plays an important role in the cluster.

The master server is set by setting [CLUSTER\\_PRIMARY\\_HOST](#).

The master server performs all the background work of dpiui2: interaction with equipment, synchronization of subscribers, services, tariffs, etc.

The remaining (slave) nodes do not perform any background activity and are idle. At the same time, these nodes are available for work: users can work with these nodes in the same way as with the master server and will not see the difference. This option can be used for load balancing as well as providing more secure access.

If the master server fails, you need to change the `CLUSTER_PRIMARY_HOST` setting and make another server the master.

## Number of nodes

For normal operation of the cluster, 3 nodes (3 servers or virtual machines) are required.

If you start the cluster on only 2 nodes, there will be problems with restarting the nodes.



!!! Important: don't try to implement GaleraFS on only 2 nodes. The cluster requires a 3rd server - an arbiter. If you restart any of the 2 nodes, you will lose data.

## Restart nodes

In normal mode, you can stop / restart 1 or 2 servers at the same time without consequences.

If you need to stop all 3 servers, you need to do it sequentially. It is advisable to stop the master node last. You must first start the server that was stopped last.

If all 3 servers were stopped, you will need to initialize the Galera cluster manually:

**1** Stop the database server on all nodes. To do this, run the following command

```
systemctl stop mariadb
```

**2** Determine which server was stopped last ([more information](#))

```
cat /var/lib/mysql/grastate.dat
```

Find the node that has `safe_to_bootstrap = 1` or the highest `seqno`. For this node, run:

```
galera_new_cluster
```

For the rest of the nodes, do:

```
systemctl start mariadb
```