







# Command Center

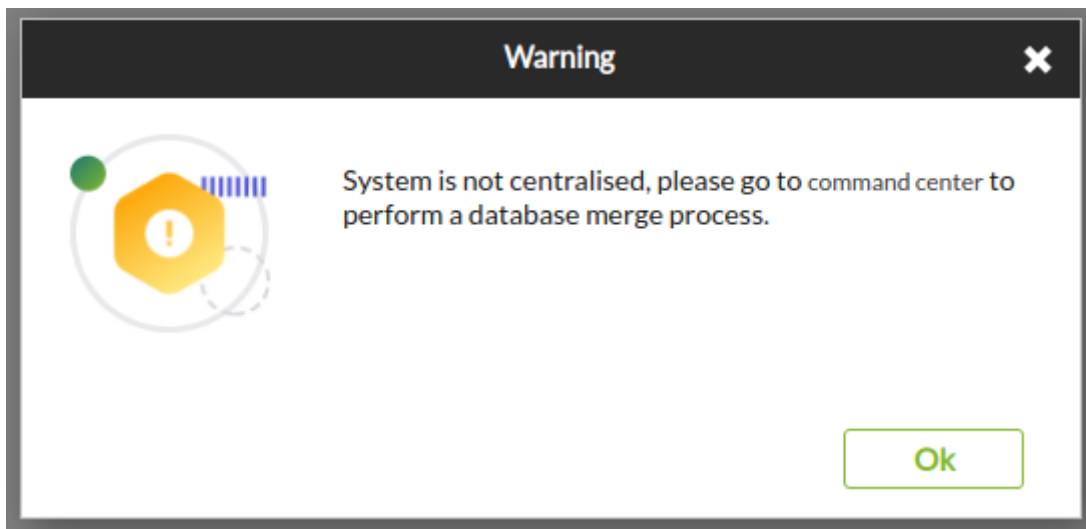
[Go back to Pandora FMS documentation index](#)

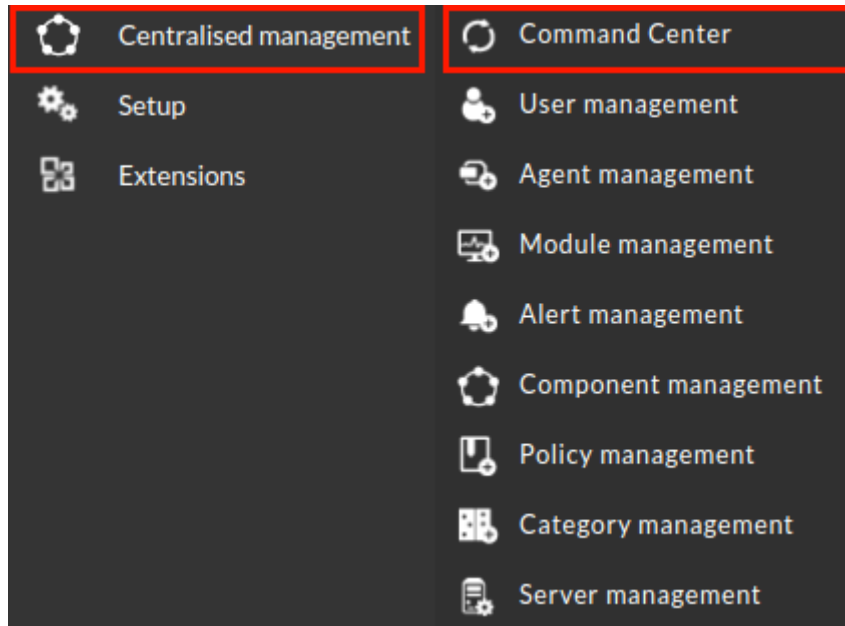
## Command Center

From Pandora FMS version 756, the synchronization system for environments with centralized mode has been redesigned from scratch, making it faster and more efficient, since the changes will be replicated to the nodes automatically without the need for manual synchronization, as it was the case up to now.

This change deems the previous system outdated, so in environments where it was active, it will have to go through the previous automatic merging system to use the new centralization system **and be able to guarantee data integrity.**

When updating, all already centralized Metaconsole environments will be forced to go through the new **Command Center** to be able to be centralized again correctly.





Command Center   User management   Agent management   Module management   Alert management   Component management   Policy management   Category management   Server management

**WARNING**

Ensure autocreate\_group token is set to a valid value in all nodes after merge.

Memory limit is recommended to be unlimited in metaconsole, please set to -1 in php.ini.

During the sync process your servers could get errors and shut down, even if they are still processing data. We recommend to stop servers during all the process.

During event synchronization, there may be a slowdown in the creation of events. If you can, keep the servers stopped.

Nodes priority order	DB	API
Metaconsole		
Node 1	●	●
Node 2	●	●

**INFORMATION**

Be sure you have all your server plugins scripts located in all your Pandora FMS nodes and Metaconsole in the proper path.

Be sure you have all the needed dependencies by your inventory modules located in all your Pandora FMS nodes and Metaconsole in the proper path.

Be sure you have any custom script needed by your alert commands located in all your Pandora FMS nodes and Metaconsole in the proper path.

Be sure you apply your policies to push the changes to the agents.

Unify databases

The Command Center will mix the different elements of the node and Metaconsole databases (of those that must be managed from Metaconsole) in the following way. An order of priority will be established between the Metaconsole nodes and the Metaconsole itself, placing the elements with the highest priority at the top of the list and at the bottom those with lower priority.

For example:

<



Nodes priority order ⓘ		DB	API
☰	Metaconsole		
☰	Node 1	●	●
☰	Node 2	●	●



Only the nodes configured in the Metaconsole that are not disabled are taken into account for the mixing process.

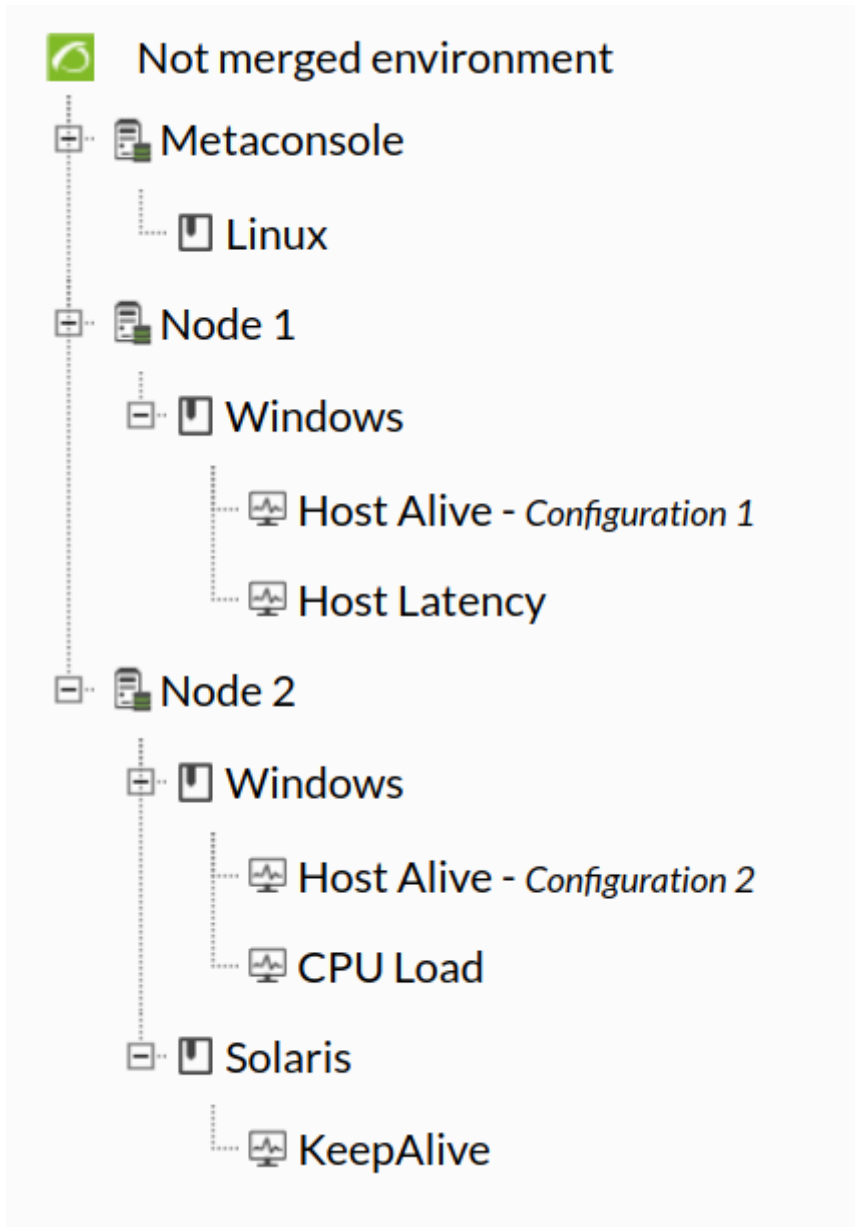
This priority list is used for cases where the **same element** exists in the different nodes but has **different configurations**. For example, for 2 nodes and the Metaconsole to have the group “Databases”. With this priority order, the configuration of the highest priority element will be taken for all, in the example of the Metaconsole.

In another case, if for example only nodes 1 and 2 had a policy called “Windows”, for all nodes and the Metaconsole the configuration for that policy would be that of Node 1 (we skip the Metaconsole because it does not have it).

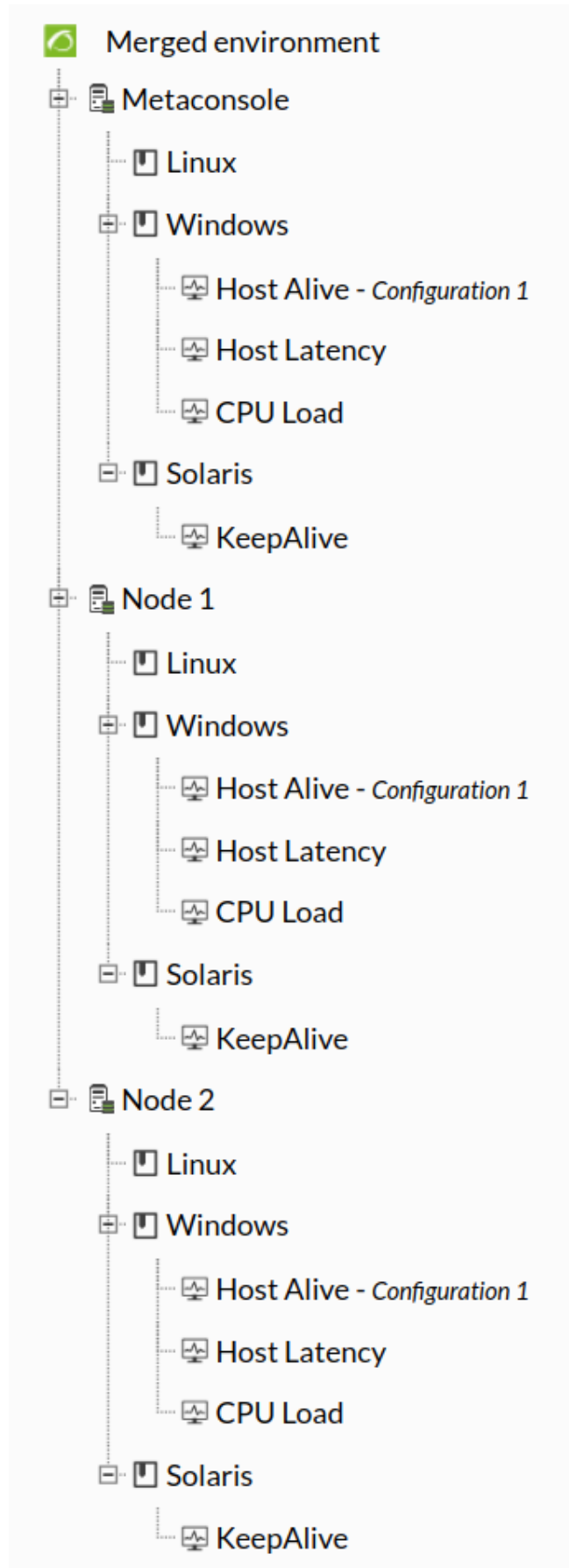


Only for the policy's own settings (group, description,...). Modules, alerts and other policy elements are considered separate elements independent of the policy and therefore are mixed as well.

The case of policies would be the most particular one out of all the synchronized elements due to how they are configured, since every module, alert, plugin... is dealt with as an independent element and seeing it only with modules, if you have:



The result of the Command Center would be:





This allows the result to have as many different configurations as possible so that you can now manage them from the Metaconsole.

## Elements centralized by the Command Center

The following elements are those centralized from the new Command Center:

- **Users:** It is only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same ID** will be considered the **same user** (following the priority rules described previously).
- **User profiles:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same user** (following the priority rules described previously).
- **Agent groups:** They are only managed from the Metaconsole. Node management is disabled.
- By unifying from the Command Center those with the **same name** will be considered to be from the **same group** (following the priority rules described previously).



Make sure you adjust the parameter `autocreate_group` from the server configuration files `pandora_server.conf` by a valid group ID after unifying from the Command Center.

- **File collections:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same short name** will be considered to be from the **same collection** (following the priority rules described previously).
- **Alert template:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same template** (following the priority rules described previously).
- **Alert commands:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same command** (following the priority rules described previously).
- **Alert actions:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same action** (following the priority rules described previously).
- **Server plugins :** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name and execution** will be considered as the **same plugin** (following the priority rules





described previously).

- **OS:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same OS** (following the priority rules described previously).
- **Module tags:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same tag** (following the priority rules described previously).
- **Module categories:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same category** (following the priority rules described previously).
- **Module groups:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same group** (following the priority rules described previously).
- **Component group:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same group** (following the priority rules described previously).
- **Network components:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name and OS** will be considered as the **same component** (following the priority rules described previously).
- **Local components:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name and OS** will be considered as the **same component** (following the priority rules described previously).
- **Component Template:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same template** (following the priority rules described previously).
- **Inventory module:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name and OS** will be considered as the **same module** (following the priority rules described previously).
- **Monitoring policies:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name** will be considered as the **same policy** (following the priority rules described previously).
- **Policy modules:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name within a policy with the same name** will be considered as the **same module** (following the priority rules described previously).
- **Policy inventory modules:** They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name and OS within a policy with the same name** will be considered as the **same module** (following the priority rules described previously).



- **Policy plugins** : They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same execution within a policy with the same name** will be considered as the **same plugin** (following the priority rules described previously).
- **Policy collections**: They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same short name within a policy with the same name** will be considered as the **same collection** (following the priority rules described previously).
- **Alerts and policy external alerts**: They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same template on the same module name within a policy with the same name** will be considered as the **same alert** (following the priority rules described previously).
- **Actions on alerts and policy external alerts**: They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those with the **same name on the same template on the same module name within a policy with the same name** will be considered as the **same action** (following the priority rules described previously).
- **Agents within policies**: They are only managed from the Metaconsole. Node management is disabled. By unifying from the Command Center those within policies with the same name will be considered as the **agents within the same policy**. Agent logs within Metaconsole policies will be discarded and only node logs will be taken into account (which is where application becomes effective).
- **Agents**: Agent management in node is allowed, except for their deletion which should be done from the Metaconsole.

The sections where those elements are centrally managed can only be managed from the Metaconsole. In case of accessing those elements from the nodes, you may only list them, and the editing and creating options will disappear. Also a warning that will indicate that the environment is in centralized mode, with a link that will lead to the administrator to the corresponding Metaconsole section for element configuration.



POLICIES MANAGEMENT

**WARNING**  
This node is configured with centralized mode. All policies information is read only. Go to [metaconsole](#) to manage it.

Group:  Recursion:  Free text for search (\*):  Filter >

Total items: 6

Status	Name	Agents	Group
	Basic Remote Checks	0	
	Basic HP-UX Local Monitoring	0	
	Basic AIX Local Monitoring	0	
	Basic Solaris Local Monitoring	0	
	Basic Linux Monitoring	0	
	Basic Windows Monitoring	0	

Total items: 6

### Prerequisites for launching the Command Center database merge

- The **Metaconsole** must be able to **connect** to all **databases** and all node **APIs**. Make sure that the “**Consoles setup**” configuration is correct and that the indicators are green.

DB	API	Compatibility	Event replication	Agent cache	Sync	Label	Version	Console URL	DB Host	Manage
						Node 1	7.0NG.756 (P) 210802	http://172.16.0.2/pandora_console/	172.16.0.2	
						Node 2	7.0NG.756 (P) 210802	http://172.16.0.3/pandora_console/	172.16.0.3	

- **Node consoles** must be able to **connect** to the **Metaconsole database**. Usually this will not be a problem, unless you have the consoles on computers other than that of Pandora FMS servers. Make sure that the **Setup** → **Enterprise** configuration parameters for the Metaconsole on the nodes is correct.



Metaconsole DB engine	MySQL
Metaconsole DB host	172.16.0.4
Metaconsole DB name	pandora
Metaconsole DB user	pandora
Metaconsole DB password	.....
Metaconsole DB port	3306

- **Servers** from all nodes must be able to **connect** to the **Metaconsole's API**. It is recommended to configure the public URL in the Metaconsole.

Public URL ⓘ	http://172.16.0.4/pandora_console/
--------------	------------------------------------

- Node **servers** must have their **API configuration** correct in “**pandora\_server.conf**” or their **public URL** configuration in the consoles **Setup**. If it is not configured, servers must be hosted in the same machines their consoles are in.

```
console_api_url http://localhost/pandora_console/include/api.php
console_api_pass pandora
```

Public URL	http://172.16.0.2/pandora_console/
------------	------------------------------------

- Each **node** must be able to **connect** to its own **history database**.



Configure connection target

Host	<input type="text" value="172.16.0.12"/>
Port	<input type="text" value="3306"/>
Database name	<input type="text" value="pandora"/>
Database user	<input type="text" value="pandora"/>
Database password	<input type="password" value="....."/>

- All **nodes and the Metaconsole** must be from the same version.
- All **nodes and the Metaconsole** must be in the **same MR**.

Pandora FMS v7.0NG.756 - Build PC210802 - MR 48  
Page generated as 2021-08-12 12:48:08

- All **nodes and Metaconsole** must have the **same maximum collection size** configured in the **Setup**.

Collection size  Bytes

- To avoid errors, the **Metaconsole and the nodes** must have the `memory_limit` parameter from `php.ini` to `-1`, that is, without limit, but only for the database mixing process. After finishing it, it is recommended to set it back to the previous value. That is because a lot of memory is used to mix the nodes, and in a very large environments (with many different elements) a large amount of memory can be used, that way you make sure that the system can use all the memory available. If the items to be merged exceed the value of the physical memory available on the server, the Command Center will fail due to an unexpected error, and in the console/apache logs you will see the line indicating the excess memory reached.
- All **nodes and the Metaconsole** must have a value for the `post_max_size` from `php.ini` **larger or equal** to their value `upload_max_file_size` de `php.ini`.
- All **nodes and Metaconsole** must have **enough space** in the disk that hosts its **directory "attachment"** to be able to carry out **backups from the database and collections**.

Attachment directory

- All **nodes and Metaconsole** must have the computer's **date and time** correctly



configured (the use of NTP servers is recommended).



If all those requirements are not met, nodes will not be mixed and it will return an error. If you check the errors of the result, it will return a message with the requirements still pending.



It is important, once the database unification is done, to set again the corresponding value of `memory_limit` in file configuration `php.ini`. Remember that for the change to take effect, the apache httpd service must be restarted

## Recommendations prior to launching the Command Center

Although they are not prerequisites for database unification process, it is recommended to carry out the following actions too:

- **Stop the pandora\_server of all nodes and the Metaconsole for the whole process.** As key elements such as groups are going to be changed, their IDs can be modified, and it is not recommended to have the server process include new references to the environment while it lasts. However, the running server shouldn't be a problem in most cases.
- **Stop the cron pandora\_db process temporarily for the duration of the process,** for the same reasons as the server.



When the merging process starts, both the nodes and the Metaconsole go into maintenance mode (not for admins). The purpose of this is the same as the recommendation to stop the servers and `pandora_db`, to prevent a user from modifying elements during the process and for that to cause errors or inconsistencies.



You cannot use this node until system is unified



Please navigate to [command center](#) to unify system

You will be automatically redirected when all tasks finish

## Merging process execution

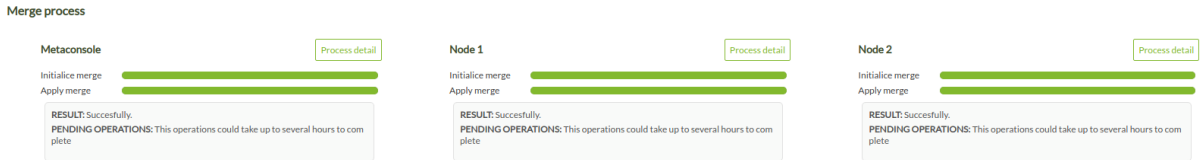
The merging process has **2 stages**, a first stage to **synchronize the different elements** that can be managed from the Metaconsole and a second stage to **update the references in the events to those centralized elements**. This process is performed that way to allow the console to be accessible again as soon as possible, since event updating is part of the process that can take the longest since it usually entails more information. Both stages are in turn divided into other 2 sub-stages differentiated in 2 progress bars.

### Stage 1 elements




In this stage **elements are synchronized** found in the databases from all nodes that can be managed from the Metaconsole. It is the merging process as such and it is sub-divided in other 2 stages, each one with its own progress bar:

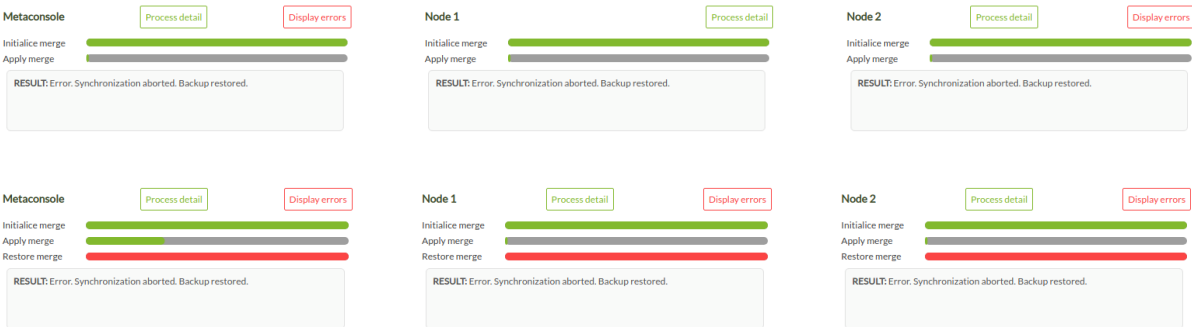
- **Initialize:** It checks all the previous requirements, generates the corresponding backups (if the requirements are met) in case any part of the process fails, and generates the result of the database merging within the memory. If this process fails for any reason, the databases will not have been modified yet, so it will not be necessary to restore backups. Backups are stored in each node/Metaconsole in the directory: /attachment/merge\_backups
- **Apply:** If the previous initialization stage was successful, it will be applied to all nodes and the Metaconsole. This process is sequential in order of priority, so when one is finished, the next one will start.




If there is an error during this process (for example, connection loss with a database), the process itself will try to restore the generated backups (a third red progress bar will be seen that will mark the restoration progress). If the reason for the failure prevents the backups from being recovered, the recovery must be done manually.



If the source of the failure prevents the backups from being recovered, the recovering shall be performed manually.





Sometimes there might be unexpected failures, for example connection lost for a while between the Metaconsole and a node's database or the





impossibility of creating a backup due to not having enough disk space, so it is possible the error message shown will be generic. If that's the case and you need it, contact Pandora FMS support team to receive assistance.

### Stage 2: Event updating

In this stage **the existing references to the different synchronized elements in events will be updated** (for example by groups). The stage is subdivided in the update of the main database events and the history database event update and will only affect those events that existed before launching the merging process. The new generated events after centralizing the nevironment will have all the correct references and won't need to be updated.

- **Main database:** As events are a large volume of information that is also affected, this update process takes place in parallel with the normal operation of the already merged environment. At this point, the server and `pandora_db` can be started again normally, and standard users are able to access the console again. Of course, you will see in the event view the update process bar for all the events, so that for that part there might still be inconsistencies (regarding filters for example) only for the events that were there before the merge. *New events would be generated normally.* This stage and process is launched by each of the nodes, through a specific task in the console's **cron**. Due to the volume of information, it can be a heavy and time-consuming task, so as far as possible the less load the environment has at that time the better (try to launch it outside of the busiest hours on Pandora FMS).
- **History database:** It would be the continuation of the previous point, updating the events in the historical database under the same characteristics already indicated.

Merge process events



### Environment already centralized through Command Center

Once stage 1 is finished, the environment will be considered centralized, and from there you will be able to manage everything from the Metaconsole. Element synchronization has also been changed, now the `pandora_ha` process of each node is in charge of synchronizing its database with that of the Metaconsole.

When you make a change in the Metaconsole (for example, create a user) this queues the



necessary queries to the database for the nodes (INSERTS, UPDATES, etc.) which pandora\_ha reads in an orderly manner and executes in each server\_threshold. This ensures that if a server is down for a while, when it is started again it can catch up correctly.

This list of pending queries can be seen from the Metaconsole in the **Consoles setup**. If for some reason any query fails, the node will not continue with the rest, you will see an error in **Consoles setup** and it will be necessary to treat it manually by an administrator. In most cases you should be able to fix it by launching the merging process again in the Command Center.

Show 20 entries

DB	API	Compatibility	Event replication	Agent cache	Sync	Database sync	Label	Version	Console URL	DB Host	Manage
●	●	●	✓ ⓘ	●	●	● ⓘ	Node 1	7.0NG.756 (P) 210802	http://172.16.0.2/pandora_console/	172.16.0.2	🔧💡🗑️
●	●	●	✓ ⓘ	●	●	● ⓘ	Node 2	7.0NG.756 (P) 210802	http://172.16.0.3/pandora_console/	172.16.0.3	🔧💡🗑️

Showing 1 to 2 of 2 entries

**Database synchronization state and pending tasks**

Filter

Show 5 entries

Time	Operation and table	Error
12 seconds	INSERT INTO tusuario	
12 seconds	INSERT INTO tusuario_perfil	

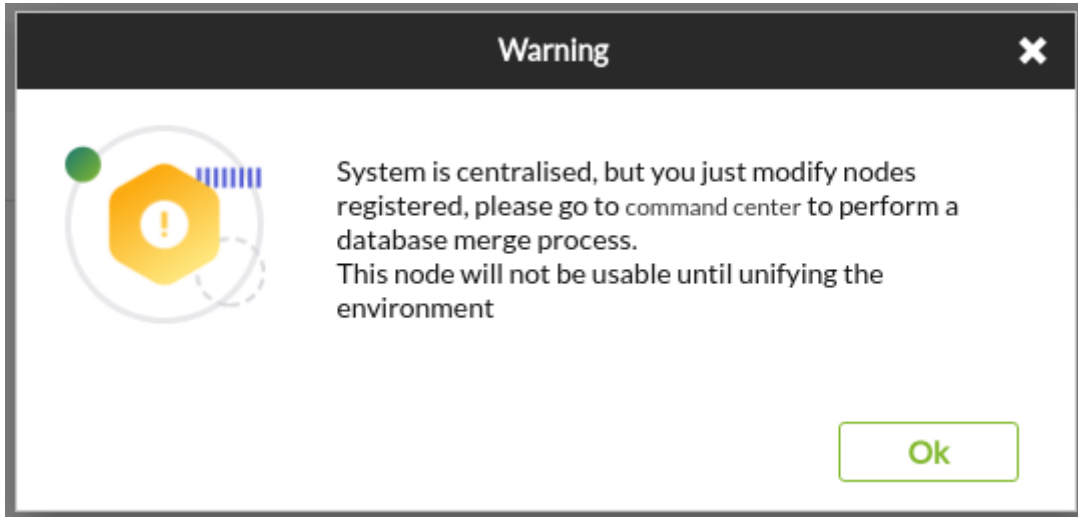
Showing 1 to 2 of 2 entries


OK

### Including new nodes

If in an already centralized environment you **add** a new node, **edit** one, or **re-enable** an existing one that has been left out of the merge, **it will be necessary to go through the Command Center again.**

A message will be displayed warning the administrator to do this task. While it is not performed, the node will remain **locked and inaccessible**, in *pending-to-merge* status. The changes made will not be assigned to it nor will it have access to the console until it goes through the merging process.



 If you need to make a change in the console to fix a bug in the merging process (such as applying an OUM), you may delete the item from the node list to temporarily unlock it.

[Go back to Pandora FMS index](#)



From:

<https://pandorafms.com/manual/> - **Pandora FMS Documentation**

Permanent link:

[https://pandorafms.com/manual/en/documentation/06\\_metaconsole/04\\_command](https://pandorafms.com/manual/en/documentation/06_metaconsole/04_command)

Last update: **2021/11/05 12:05**