**Formation Windows Cluster et HD  
 - sys 30 – Quorum**

Michel Cabaré / www.cabare.net / michel@cabare.net

Windows Cluster Haute disponibilité – Quorum   
 - sys 30 - V2-00 - Janvier 2021

|  |  |
| --- | --- |
| https://WWW.CABARE.NET ©  Une image contenant objet  Description générée automatiquement |  |
|  | |
| Certification qualité délivrée au titre de la catégorie d’action suivante  actions de formation | |

**table des matiÈres**

[Gestion du QUORUM 3](#_Toc62478354)

[Principe et Objectif : 3](#_Toc62478355)

[Amélioration du Stockage - MPIO 4](#_Toc62478356)

[Principe et Objectif : 4](#_Toc62478357)

[Cluster Group 5](#_Toc62478358)

[Principe et Objectif : 5](#_Toc62478359)

# Gestion du QUORUM

## Principe et Objectif :

**N.B** : il serait tout à fait possible d’utiliser un stockage pré-existant, sur une baie « Externe » à la maquette de formation.

# Amélioration du Stockage - MPIO

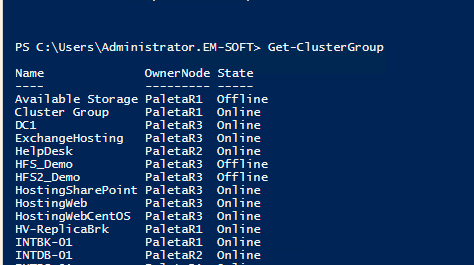
## Principe et Objectif :

**N.B** : il serait tout à fait possible d’utiliser un stockage pré-existant, sur une baie « Externe » à la maquette de formation.

# Cluster Group

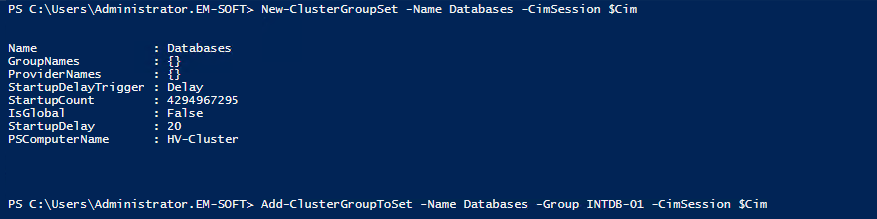
## Principe et Objectif :

Many times (or almost always), you have to define start order of VMs as services on one VM in an exactly defined order. We tried to solve this problem with start delays in the past – or with some additional software, but there were always situations where we are unable to control all factors.  
Now in Windows Server 2016 edition it is better, as we can define startup groups. This means, that we define a group of servers who will start together and another group of servers who will start later when the first group is started or also with some delay after the first group is started. Additionally, we can start just a last group of servers and because this group depends on other groups, system will first start the parent server group. So, you are not able to start some servers before all infrastructure depends on it is started. We have new cmdlets in PowerShell to define and manage Cluster Groups (you can list them with***Get-Command -Noun \*ClusterGroup\****).

[](https://www.em-soft.si/myblog/elvis/?attachment_id=825)  
To do these settings, you have to know some concepts that are introduced in Windows Server 2016 and we will use them.

* This post explains how to create groups in Hyper-V cluster; it will not work on non-clustered servers. If you want to setup startup order for a single Hyper-V host, [this is the post](https://charbelnemnom.com/2016/05/a-first-look-at-vm-groups-and-vm-start-ordering-in-windows-server-2016-hyper-v-hyperv-ws2016/)where you can find how to do it.
* Cluster Group: Represents clustered services or applications (resource groups) in a failover cluster – any HA VM has his group. You can view cluster groups with PowerShell cmdlet***Get-ClusterGroup***. You don’t need to change anything here; just leave them as they are.
* Cluster Group Set: Is a set of Cluster groups (VMs) that we want to control together. This is a set of VMs that have similar services and we want to control them as a group. Here we can control some settings (startup delay, global or local, …). Cmdlet we have to use at this point is ***Set-ClusterGroupSet***([cmdlet syntax](https://technet.microsoft.com/en-us/itpro/powershell/windows/failoverclusters/set-clustergroupset))
* Cluster group set dependency: Is a dependency where we specify which group and when it will start. To be clear, with dependency we define VM startup order.

**How to setup environment?**  
I always start creating output with cmdlet Get-ClusterGroup, because it is easier to manage all VMs when I have all their names on a paper or TXT file. It is easier to review them, define services that they offer and later functionally define group sets.  
When you have defined group sets (put VMs with similar services or dependencies together) it is time to create Cluster Group Sets. This operation is done in few steps:

* Create Cluster Group Set: this will create an empty group for grouping VMs. To do this you have to use cmdlet ***New-ClusterGroupSet -Name GroupName***.
* Add Cluster Groups (VMs) to sets. In this step, we will populate Cluster Group Sets with VMs – this mean that we will put together all VMs with similar services or VMs that we need to start at the same time. When Cluster Group Set will be asked to start, all VMs that are in it will be started. There is no dependency inside the Cluster Group Set and we have no chance to control the start order inside the group. If we think that some VMs need to start before other VMs, we need more than one Cluster Group Set. To add VM to Cluster Group Set we will use PowerShell cmdlet ***Add-ClusterGroupToSet -Name GroupName -Group ClusterGroupName***. In this cmdlet, you have to change ***GroupName*** with your Cluster Group Set name and ***ClusterGroupName***with Cluster Group name (VM name – output in firs step). We have to repeat cmdlet for every single VM.[](https://www.em-soft.si/myblog/elvis/?attachment_id=824)
* Create dependency between Cluster Group Sets: We have to use cmdlet ***Add-ClusterGroupDependency -Group GroupName -ProviderSet GroupDependsOn***. ***GroupName*** is the name of the Cluster Group Set we want to start and ***GroupDependsOn*** is a Cluster Group Set necessary to be started previously. At this point we need to create startup order (dependency) between groups. The start of any group can depend on successful start of one or more groups. If previous group will fail to start, the group who depends on it will not start. I suggest you to have in mind this situation (maybe develop a script to add and remove all VMs from groups – you will quickly solve problems if they appear).

With this few steps, we created a startup order for our environment. If it is all OK, we will never have a situation when a service will not work because some dependency server is not started. Practically, the system will look to start all VMs in a defined order.