

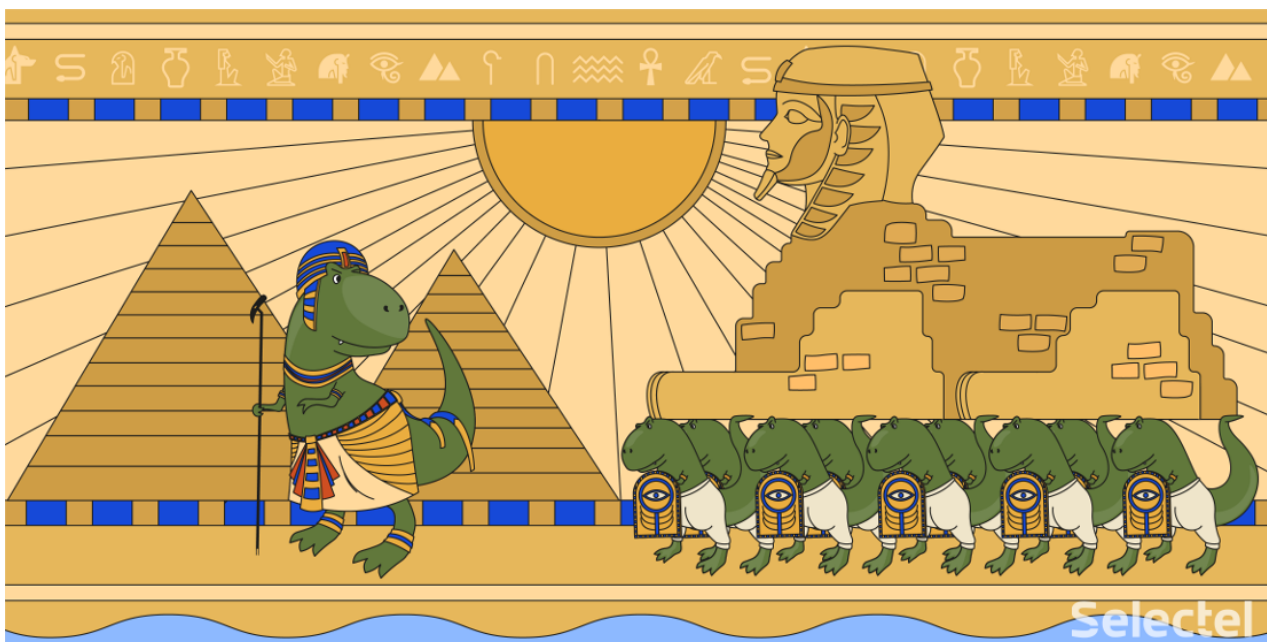
Кластеризация в Proxmox VE



Николай Рубанов Старший технический писатель

19 декабря 2019

В прошлых статьях мы начали рассказывать о том, что такое Proxmox VE и как он работает. Сегодня мы расскажем о том, как можно использовать возможность кластеризации и покажем какие преимущества это дает. Что же такое кластер и зачем он нужен? Кластер (от англ. cluster) — это группа серверов, объединенных скоростными каналами связи, работающая и представляющаяся [...]



В прошлых статьях мы начали рассказывать о том, что такое Proxmox VE и как он работает. Сегодня мы расскажем о том, как можно использовать возможность кластеризации и покажем какие преимущества это дает.

Что же такое кластер и зачем он нужен? Кластер (от англ. cluster) — это группа серверов, объединенных скоростными каналами связи, работающая и представляющаяся пользователю как единое целое. Существует несколько основных сценариев использования кластера:

- **Обеспечение отказоустойчивости** (High-availability).
- **Балансировка нагрузки** (Load Balancing).
- **Увеличение производительности** (High Performance).
- **Выполнение распределенных вычислений** (Distributed computing).

Каждый сценарий предъявляет свои собственные требования к составляющим кластера. Например, для кластера, выполняющего распределенные вычисления, основным требованием является высокая скорость выполнения операций с плавающей запятой и низкая латентность сети. Подобные кластеры часто используются в научно-исследовательских целях.

Раз уж мы коснулись темы распределенных вычислений, то хочется отметить, что существует еще такое понятие как **грид-система** (от англ. grid — решетка, сеть). Несмотря на общую схожесть, не стоит путать грид-систему и кластер. Грид не является кластером в привычном понимании. В отличие от кластера, входящие в грид узлы чаще всего разнородны и отличаются низкой доступностью. Такой подход упрощает решение задач распределенных вычислений, однако не позволяет создать из узлов единое целое.

Яркий пример грид-системы — популярная вычислительная платформа **BOINC** (Berkeley Open Infrastructure for Network Computing). Эта платформа изначально создавалась для проекта **SETI@home** (Search for Extra-Terrestrial Intelligence at Home), занимающегося проблемой поиска внеземного разума путем анализа радиосигналов.

Как это работает

Огромный массив данных, полученных с радиотелескопов, разбивается на множество небольших кусочков, и они отправляются на узлы грид-системы (в проекте SETI@home роль подобных узлов играют компьютеры добровольцев). Данные обрабатываются на узлах и после завершения обработки отправляются на центральный сервер проекта SETI. Таким образом проект решает сложнейшую глобальную задачу, не имея в своем распоряжении требуемых вычислительных мощностей.

Теперь, когда у нас есть четкое понимание того, что такое кластер, предлагаем рассмотреть каким образом его можно создать и задействовать. Будем использовать систему виртуализации с открытым исходным кодом **Proxmox VE**.

Особенно важно перед тем, как приступить к созданию кластера четко понимать ограничения и системные требования Proxmox, а именно:

- максимальное количество нод в кластере — **32**;
- все ноды должны иметь **одинаковую версию Proxmox** (есть исключения, но для production они не рекомендуются);
- если в дальнейшем планируется задействовать функционал High Availability, то в кластере должно быть как минимум **3** ноды;
- для взаимодействия нод друг с другом должны быть открыты порты **UDP/5404**, **UDP/5405** для corosync и **TCP/22** для SSH;
- задержка в сети между нодами не должна превышать **2** мс.

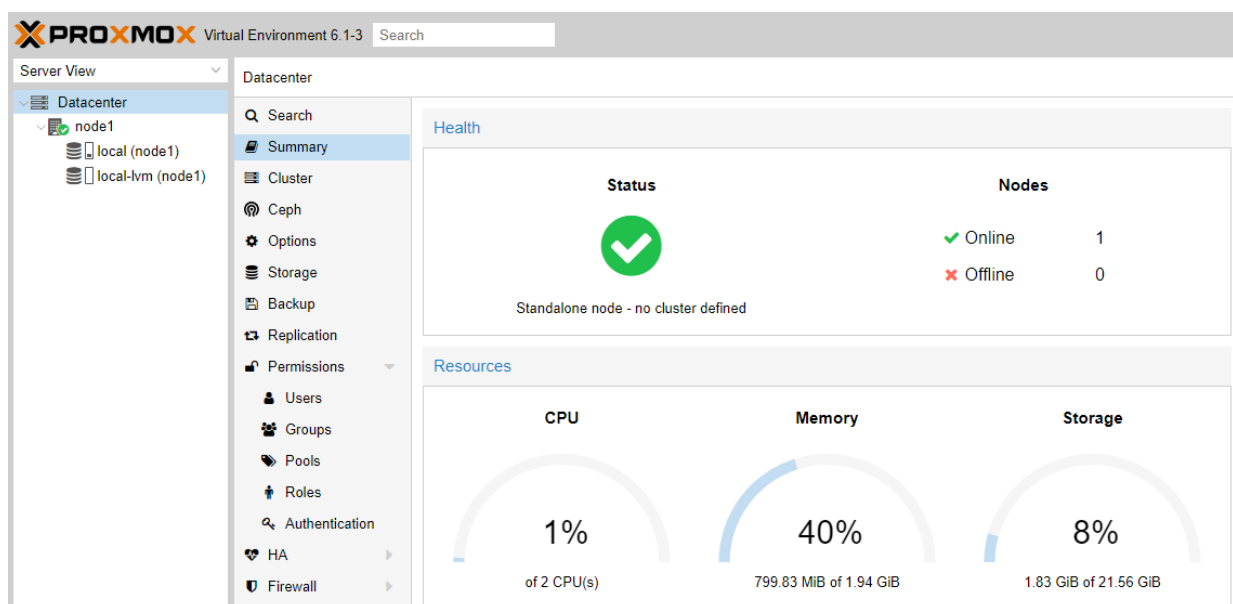
Создание кластера

Важно! Нижеприведенная конфигурация является тестовой. Не забудьте свериться с [официальной документацией Proxmox VE](#).

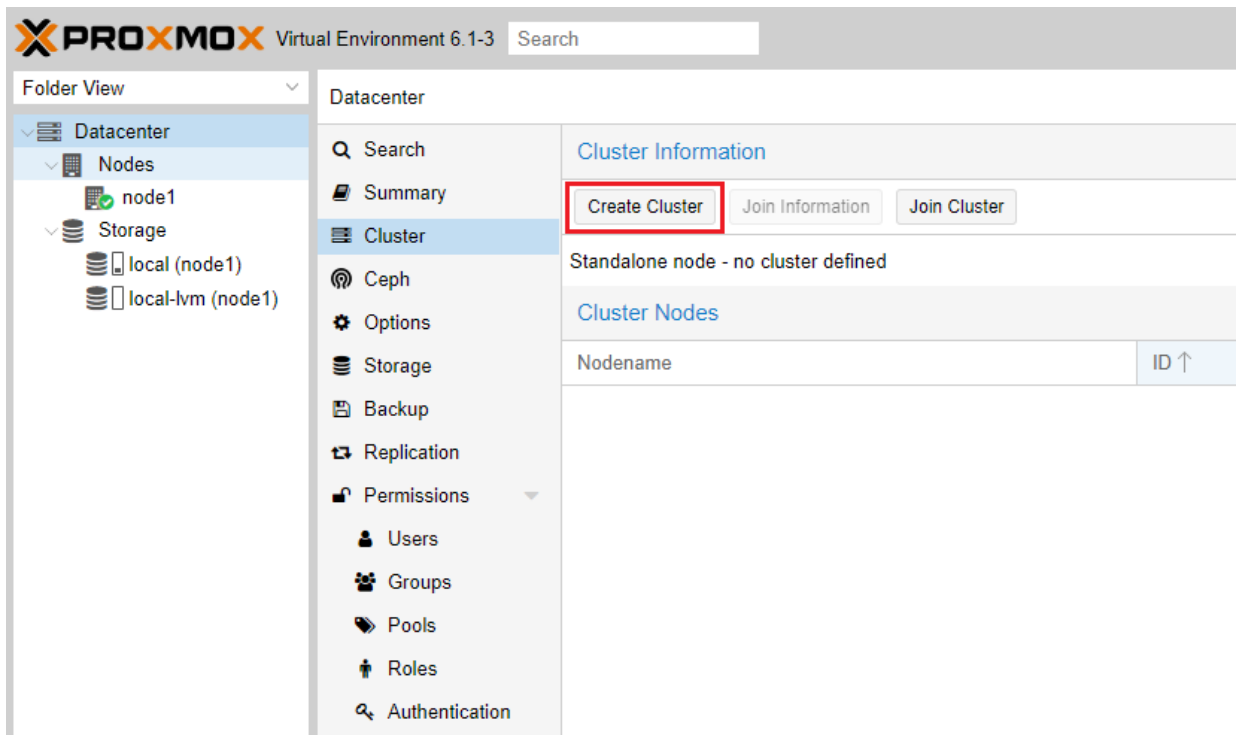
Для того, чтобы запустить тестовый кластер, мы взяли три сервера с установленным гипервизором Proxmox одинаковой конфигурации (2 ядра, 2 Гб оперативной памяти).

Если вы хотите узнать каким образом можно установить Proxmox, то рекомендуем прочитать нашу предыдущую статью — [Магия виртуализации: вводный курс в Proxmox VE](#)

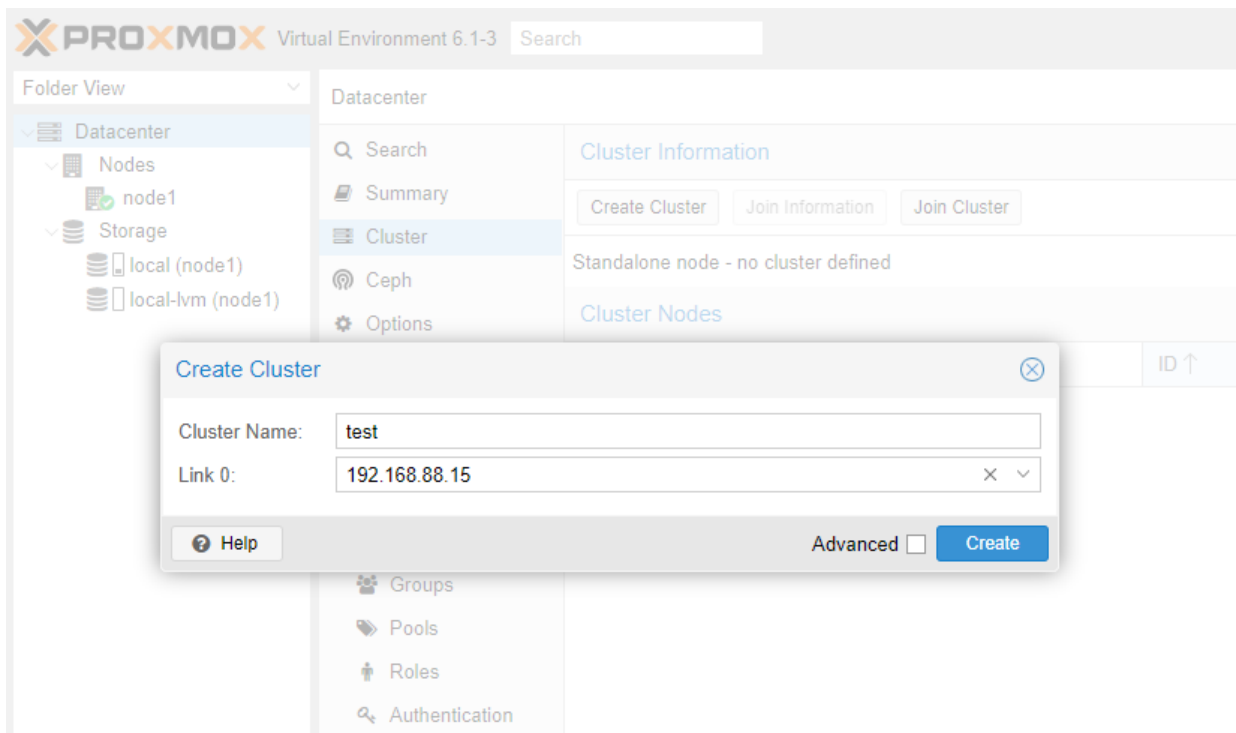
Изначально, после установки ОС, единичный сервер работает в **Standalone-mode**.



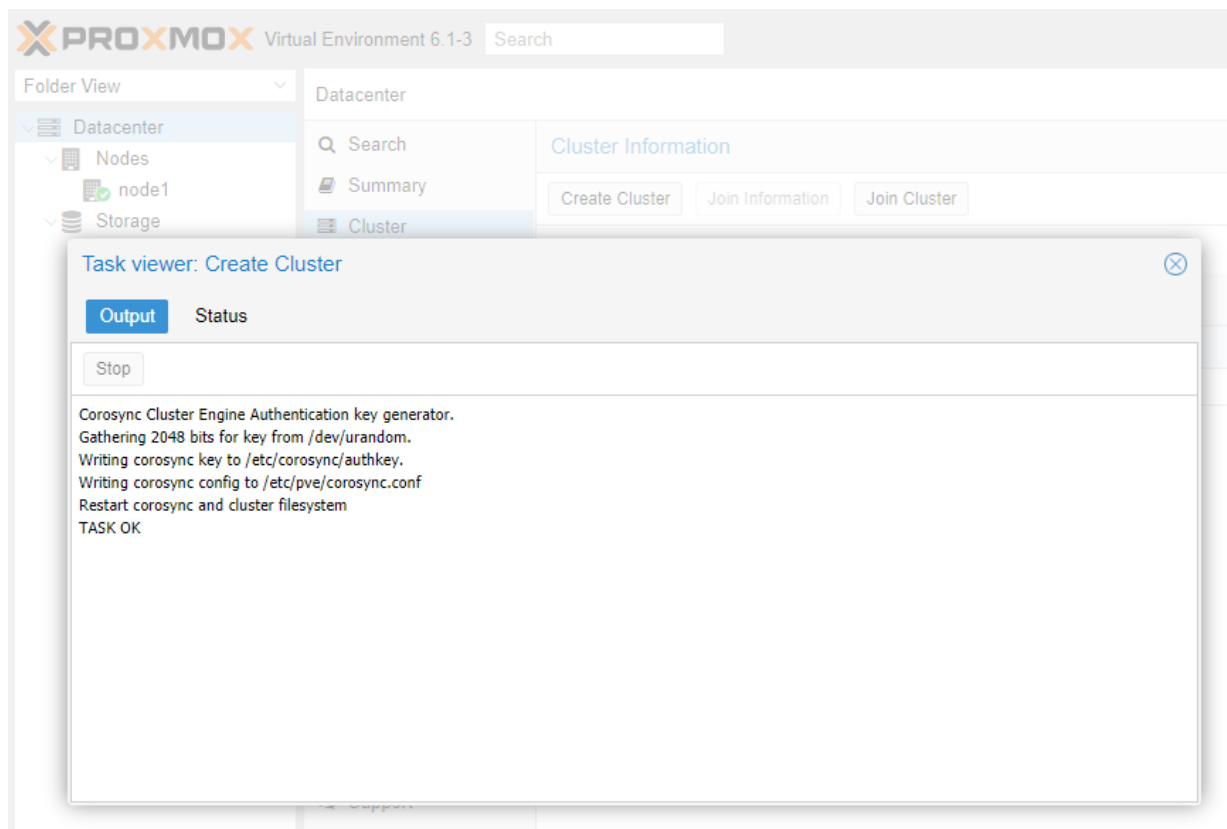
Создадим кластер, нажав кнопку **Create Cluster** в соответствующем разделе.



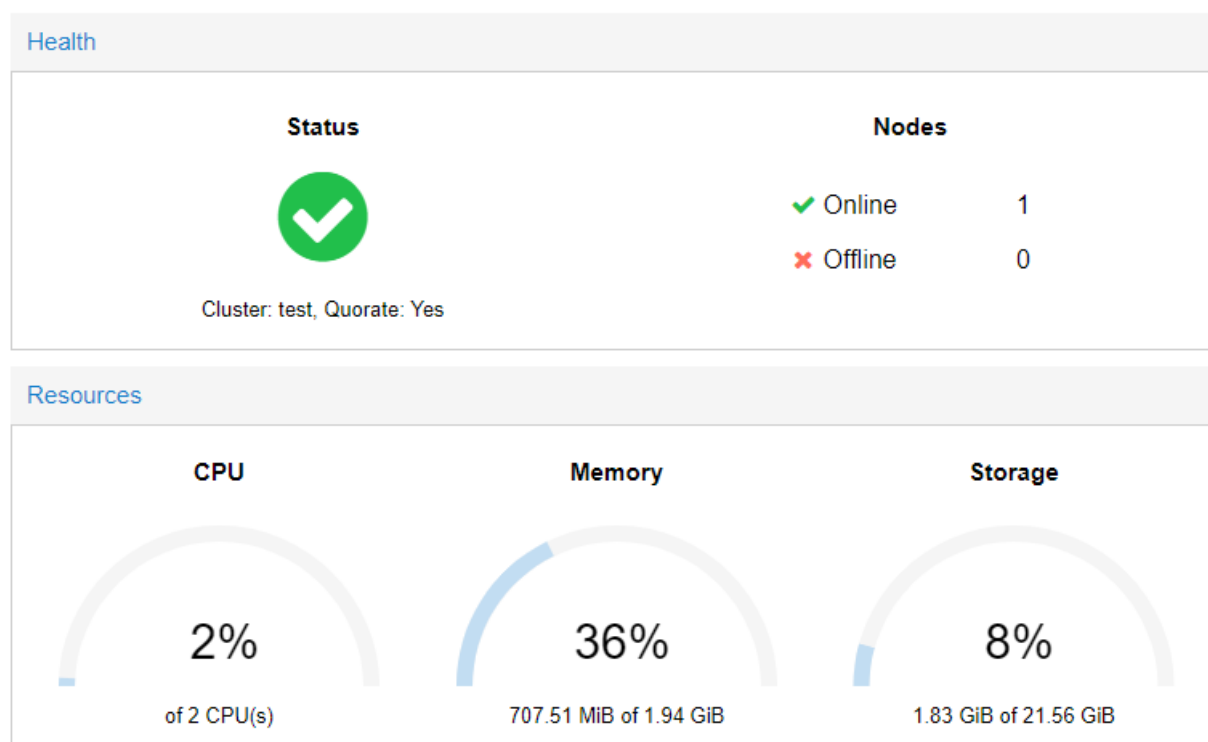
Задаем имя будущему кластеру и выбираем активное сетевое соединение.



Нажимаем кнопку **Create**. Сервер сгенерирует 2048-битный ключ и запишет его вместе с параметрами нового кластера в конфигурационные файлы.



Надпись **TASK OK** свидетельствует об успешном выполнении операции. Теперь, взглянув на общую информацию о системе видно, что сервер перешел в режим кластера. Пока что кластер состоит всего лишь из одной ноды, то есть пока у него нет тех возможностей, ради которых необходим кластер.



Присоединение к кластеру

The screenshot shows the Proxmox VE 6.1-3 web interface. On the left is a 'Folder View' sidebar with a tree structure: 'Datacenter' (expanded) contains 'Nodes' (with 'node1') and 'Storage' (with 'local (node1)' and 'local-lvm (node1)'). The main area is titled 'Datacenter' and contains a left-hand menu with options: Search, Summary, Cluster (selected), Ceph, Options, Storage, Backup, Replication, Permissions (expanded to show Users, Groups, Pools, Roles), Authentication, HA, Firewall, and Support. The main content area is divided into two sections. The top section, 'Cluster Information', contains three buttons: 'Create Cluster', 'Join Information' (highlighted with a red rectangle), and 'Join Cluster'. Below these buttons, it shows 'Cluster Name: test'. The bottom section, 'Cluster Nodes', contains a table with two columns: 'Nodename' and 'ID ↑'. The table has one row with 'node1' in the first column and '1' in the second column.

Proxmox Virtual Environment 6.1-3 Search

Folder View ▾

- ▼ Datacenter
 - ▼ Nodes
 - node1
 - ▼ Storage
 - local (node1)
 - local-lvm (node1)

Datacenter

- Search
- Summary
- Cluster
- Ceph
- Options
- Storage
- Backup
- Replication
- Permissions ▾
 - Users
 - Groups
 - Pools
 - Roles
- Authentication
- HA ▶
- Firewall ▶
- Support

Cluster Information

Create Cluster Join Information Join Cluster

Cluster Name: test

Cluster Nodes

Nodename	ID ↑
node1	1

PROXMOX Virtual Environment 6.1-3 Search

Folder View

- Datacenter
 - Nodes
 - node1
 - Storage

Datacenter

Search

Summary

Cluster

Cluster Information

Create Cluster

Join Information

Join Cluster

Cluster Join Information

Copy the Join Information here and use it on the node you want to add.

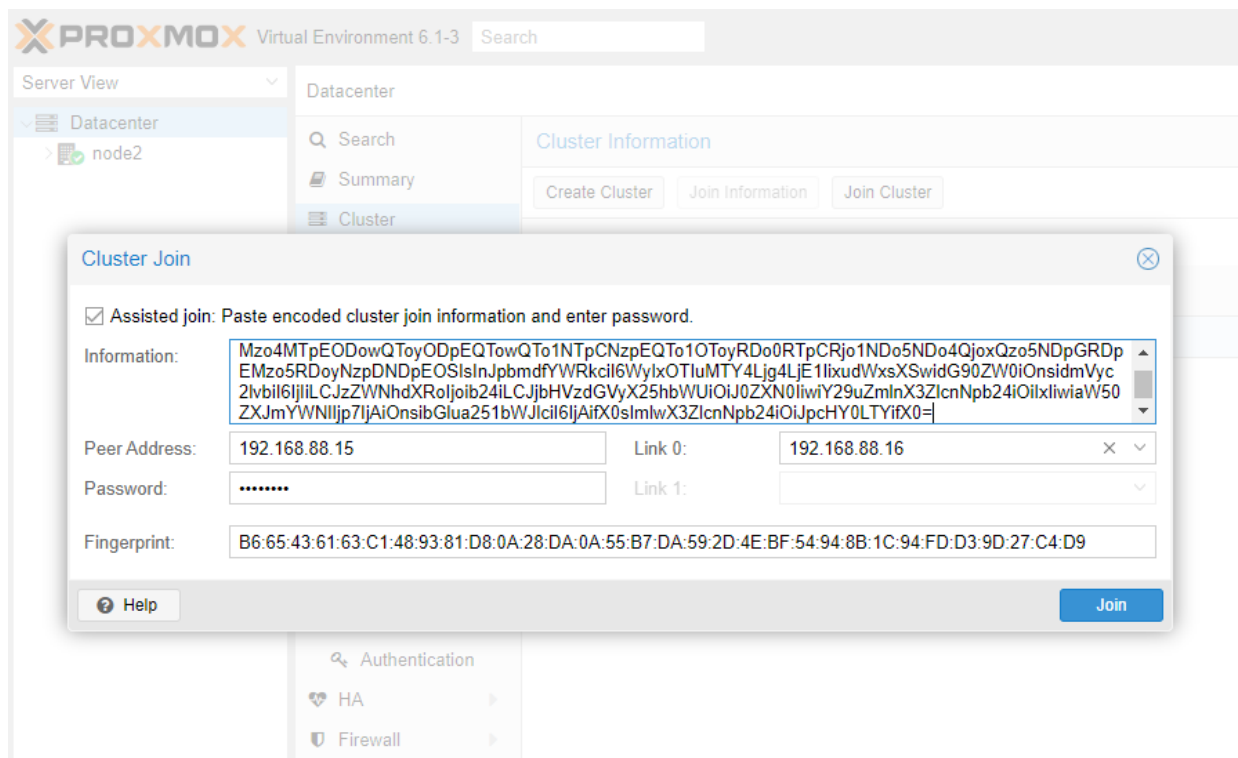
IP Address: 192.168.88.15

Fingerprint: B6:65:43:61:63:C1:48:93:81:D8:0A:28:DA:0A:55:B7:DA:59:2D:4E:BF:54:94:8B:1C:94:FD:D3:9D:27:C4:D9

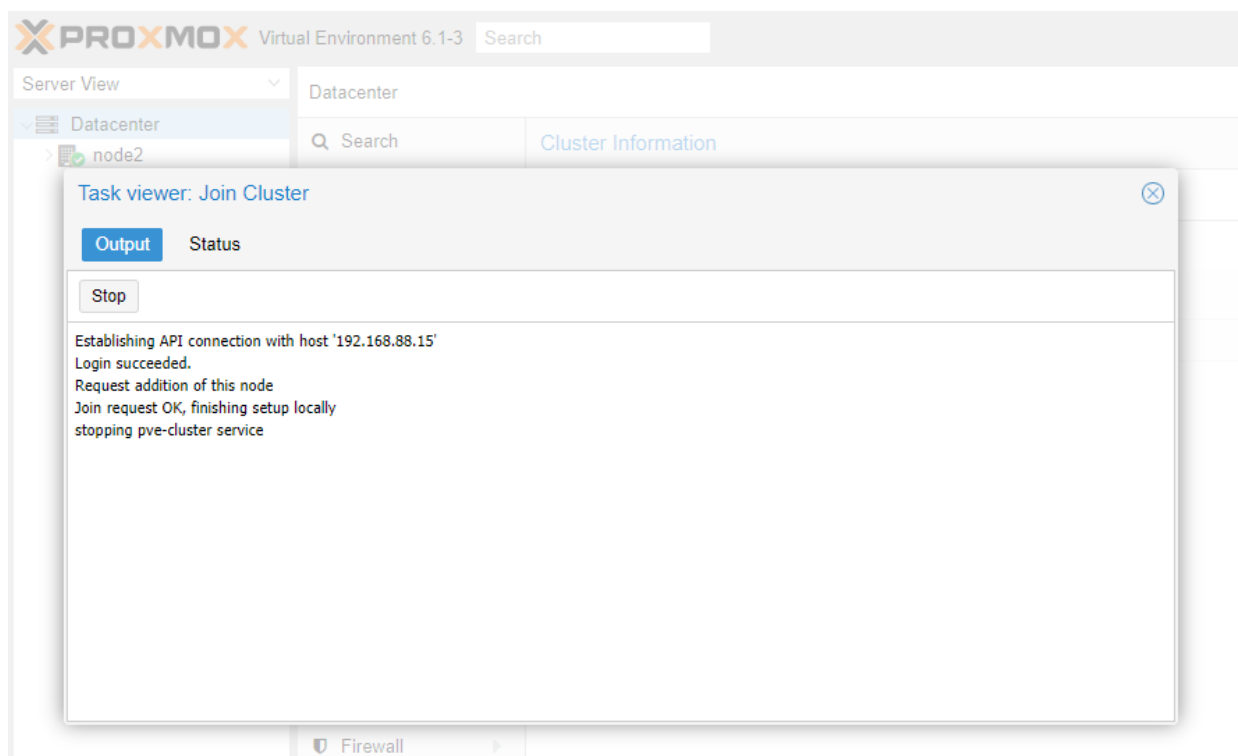
Join Information: eyJpcEFkZHZlc3MiOiJOTlUMTY4Ljg4LjE1IiwizmluZ2VycHJpbnQiOiJCNjo2NT0mZzo2MT02MzpmZmTo0ODo5Mzo4MTpEODowQToyODpEQTo1NTpCNzpEQTo1OToyRD00RTpCRjo1ND05ND04QjoxQzo5NDpGRDpEMzo5RD0yNzpnDNDpEOSlInJpbmdfYWwkaWwyaWxOTlUMTY4Ljg4LjE1IiwidWxsXSwidG90ZW0iOndidmVvc2lmbil6lilil.C.lz7WNh4XRolinib24il.C.lihHVzdGwVwX25hbWlIiOiI07XN0liwiY29u7mlnX37lcnNob24iOiJlxiw

Copy Information

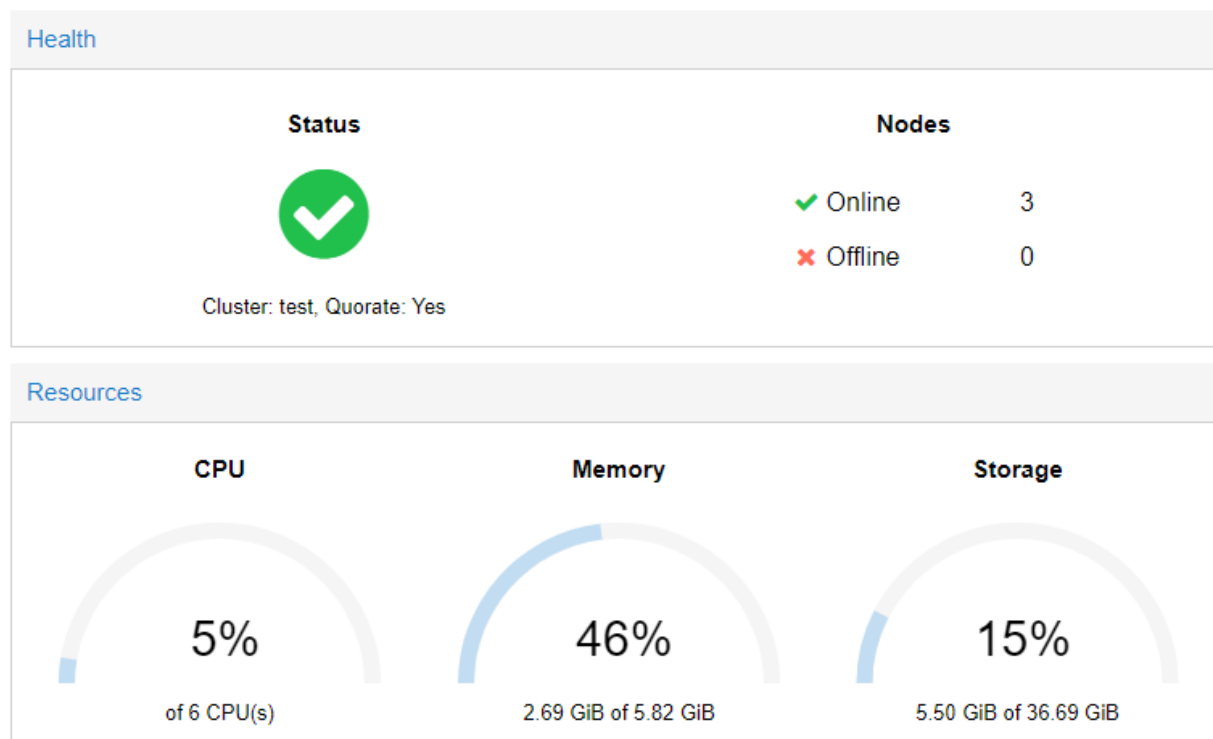
Здесь закодированы все необходимые параметры подключения: адрес сервера для подключения и цифровой отпечаток. Переходим на сервер, который необходимо включить в кластер. Нажимаем кнопку **Join Cluster** и в открывшемся окне вставляем скопированное содержимое.



Поля **Peer Address** и **Fingerprint** будут заполнены автоматически. Вводим пароль root от ноды номер 1, выбираем сетевое подключение и нажимаем кнопку **Join**.



В процессе присоединения к кластеру веб-страница GUI может перестать обновляться. Это нормально, просто перезагружаем страницу. Точно таким же образом добавляем еще одну ноду и в итоге получаем полноценный кластер из 3-х работающих узлов.



Теперь мы можем контролировать все узлы кластера из одного GUI.

Guests

Virtual Machines		LXC Container	
● Running	0	● Running	0
● Stopped	0	● Stopped	0

Nodes

Name	ID	Online	Support	Server Address	CPU usage	Memory usage	Uptime
node1	1	✓	-	192.168.88.15	4%	47%	00:10:07
node2	2	✓	-	192.168.88.16	9%	47%	00:07:44
node3	3	✓	-	192.168.88.17	2%	45%	00:08:24

Организация High Availability

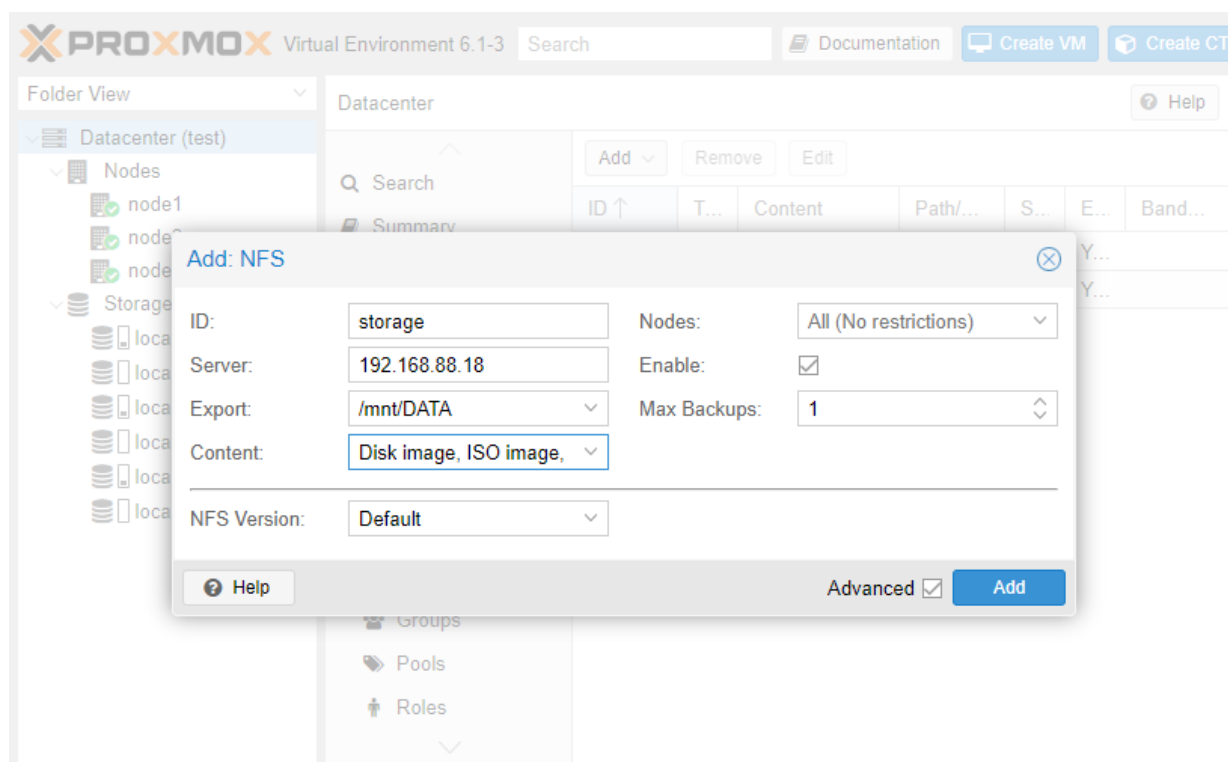
Проксмох «из коробки» поддерживает функционал организации HA как для виртуальных машин, так и для LXC-контейнеров. Утилита **ha-manager** обнаруживает и обрабатывает ошибки и отказы, выполняя аварийное переключение с отказавшей ноды на рабочую. Чтобы механизм работал корректно необходимо, чтобы виртуальные машины и контейнеры имели общее файловое хранилище.

После активации функционала High Availability, программный стек ha-manager начнет непрерывно отслеживать состояние работы виртуальной машины или контейнера и асинхронно взаимодействовать с другими нодами кластера.

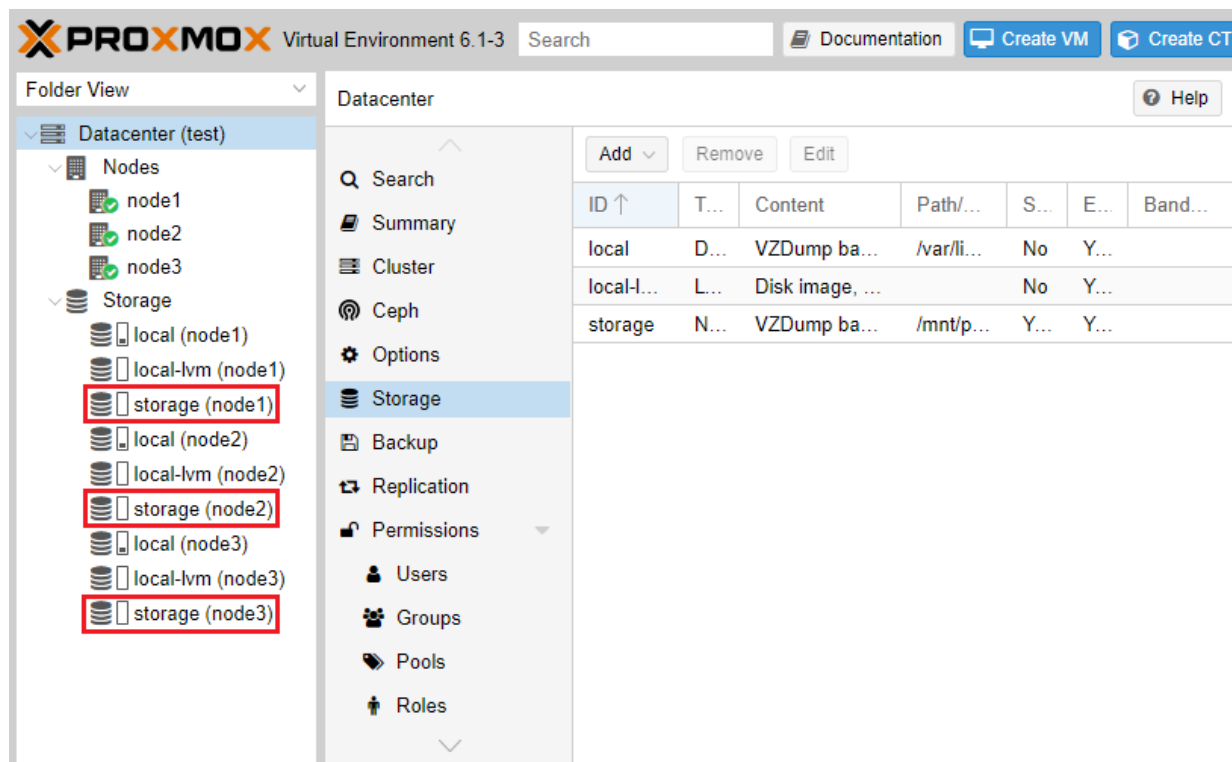
Присоединяем общее хранилище

Для примера мы развернули небольшое файловое хранилище NFS по адресу 192.168.88.18. Чтобы все ноды кластера могли его использовать нужно проделать следующие манипуляции.

Выбираем в меню веб-интерфейса **Datacenter — Storage — Add — NFS**.



Заполняем поля **ID** и **Server**. В выпадающем списке **Export** выбираем нужную директорию из доступных и в списке **Content** — необходимые типы данных. После нажатия кнопки **Add** хранилище будет подключено ко всем нодам кластера.

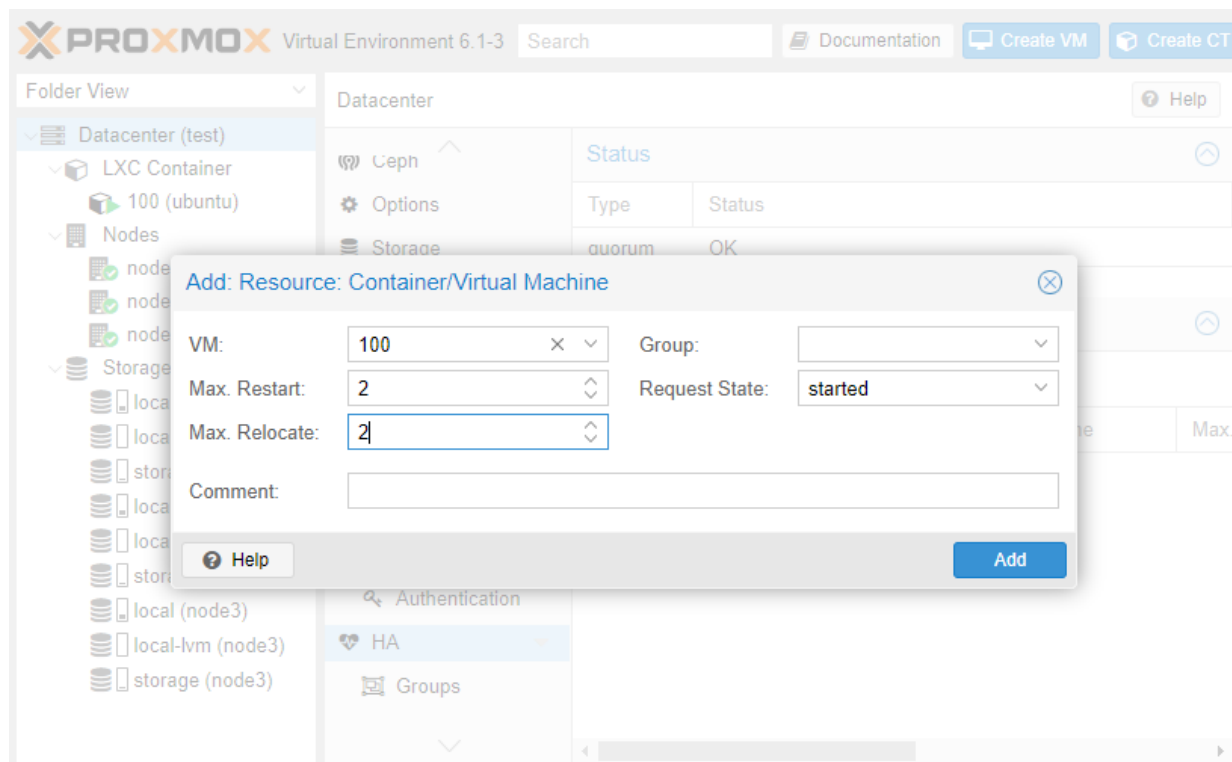


При создании виртуальных машин и контейнеров на любом из узлов указываем наш **storage** в качестве хранилища.

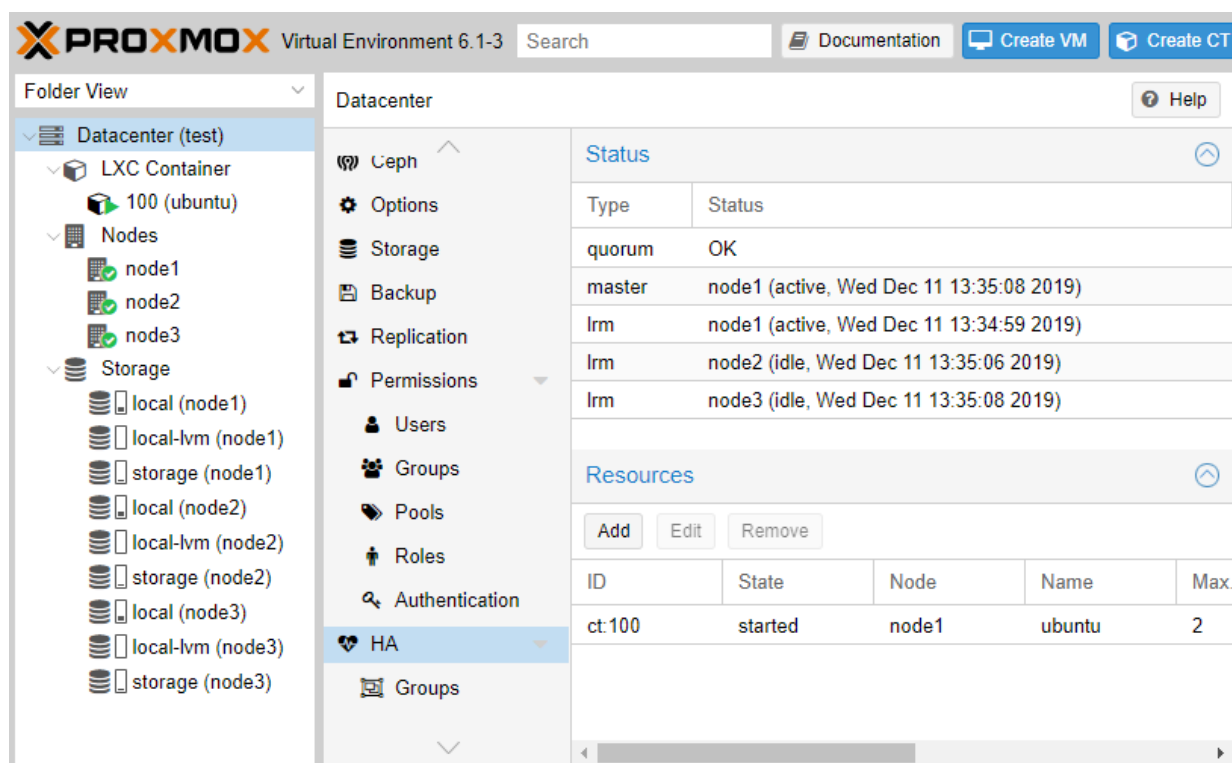
Настраиваем HA

Для примера создадим контейнер с Ubuntu 18.04 и настроим для него High Availability. После создания и запуска контейнера заходим в раздел **Datacenter** — **HA** — **Add**. В открывшемся поле указываем ID виртуальной машины/контейнера и максимальное количество попыток рестарта и перемещения между нодами.

Если это количество будет превышено, гипервизор пометит VM как сбойную и переведет в состояние Error, после чего прекратит выполнять с ней какие-либо действия.



После нажатия кнопки **Add** утилита **ha-manager** оповестит все ноды кластера о том, что теперь VM с указанным ID контролируется и в случае падения ее необходимо перезапустить на другой ноде.



Устроим сбой

Чтобы посмотреть, как именно обрабатывает механизм переключения, погасим нештатно node1 по питанию. Смотрим с другой ноды, что происходит с кластером. Видим, что система зафиксировала сбой.

PROXMOX Virtual Environment 6.1-3 Search Documentation Create VM Create CT

Folder View

- Datacenter (test)
 - LXC Container
 - 100 (ubuntu)
 - Nodes
 - node1
 - node2
 - node3
 - Storage
 - local (node1)
 - local-lvm (node1)
 - storage (node1)
 - local (node2)
 - local-lvm (node2)
 - storage (node2)
 - local (node3)
 - local-lvm (node3)
 - storage (node3)

Datacenter

Search Summary Cluster Ceph Options Storage Backup Replication Permissions Users Groups Pools Roles

Status

Type	Status
quorum	OK
master	node1 (old timestamp - dead?, Wed Dec 11 13:46:59 20...
lrm	node1 (old timestamp - dead?, Wed Dec 11 13:46:59 20...
lrm	node2 (idle, Wed Dec 11 13:47:42 2019)
lrm	node3 (idle, Wed Dec 11 13:47:42 2019)

Resources

Add Edit Remove

ID	State	Node	Name	Max.
ct:100	started	node1	ubuntu	2

Работа механизма HA не означает непрерывность работы VM. Как только ноды «упала», работа VM временно останавливается до момента автоматического перезапуска на другой ноды.

И вот тут начинается «магия» — кластер автоматически переназначил ноду для выполнения нашей VM и в течение 120 секунд работа была автоматически восстановлена.

PROXMOX Virtual Environment 6.1-3 Search Documentation Create VM Create CT

Folder View

- Datacenter (test)
 - LXC Container
 - 100 (ubuntu)
 - Nodes
 - node1
 - node2
 - node3
 - Storage
 - local (node1)
 - local-lvm (node1)
 - storage (node1)
 - local (node2)
 - local-lvm (node2)
 - storage (node2)
 - local (node3)
 - local-lvm (node3)
 - storage (node3)

Datacenter

Search Summary Cluster Ceph Options Storage Backup Replication Permissions Users Groups Pools Roles

Status

Type	Status
quorum	OK
master	node3 (active, Wed Dec 11 13:51:50 2019)
lrm	node1 (old timestamp - dead?, Wed Dec 11 13:46:59 20...
lrm	node2 (active, Wed Dec 11 13:51:52 2019)
lrm	node3 (idle, Wed Dec 11 13:51:52 2019)

Resources

Add Edit Remove

ID	State	Node	Name	Max.
ct:100	started	node2	ubuntu	2

Гасим node2 по питанию. Посмотрим, выдержит ли кластер и вернется ли VM в рабочее состояние автоматически.

The screenshot shows the Proxmox VE 6.1-3 interface. On the left, the 'Folder View' shows a 'Datacenter (test)' containing an 'LXC Container' named '100 (ubuntu)', three 'Nodes' (node1, node2, node3), and 'Storage' for each node. The 'Datacenter' tab is active, showing a 'Status' section with a table of cluster components. The 'quorum' status is 'No quorum on node 'node3!'. The 'master' is 'node3 (old timestamp - dead?, Wed Dec 11 14:02:40 20...)'. The 'Irm' (Inactive Resources Monitor) shows 'node1 (old timestamp - dead?, Wed Dec 11 13:46:59 20...)', 'node2 (old timestamp - dead?, Wed Dec 11 14:02:43 20...)', and 'node3 (old timestamp - dead?, Wed Dec 11 14:02:43 20...)'. The 'Resources' section shows a table with one entry: 'ct:100' in state 'started' on 'node2' with name 'ubuntu' and maximum 2.

Type	Status
quorum	No quorum on node 'node3!'
master	node3 (old timestamp - dead?, Wed Dec 11 14:02:40 20...)
Irm	node1 (old timestamp - dead?, Wed Dec 11 13:46:59 20...)
Irm	node2 (old timestamp - dead?, Wed Dec 11 14:02:43 20...)
Irm	node3 (old timestamp - dead?, Wed Dec 11 14:02:43 20...)

ID	State	Node	Name	Max.
ct:100	started	node2	ubuntu	2

Увы, как видим, у нас возникла проблема с тем, что на единственной, оставшейся в живых, ноде больше нет кворума, что автоматически отключает работу НА. Даем в консоли команду принудительной установки кворума.

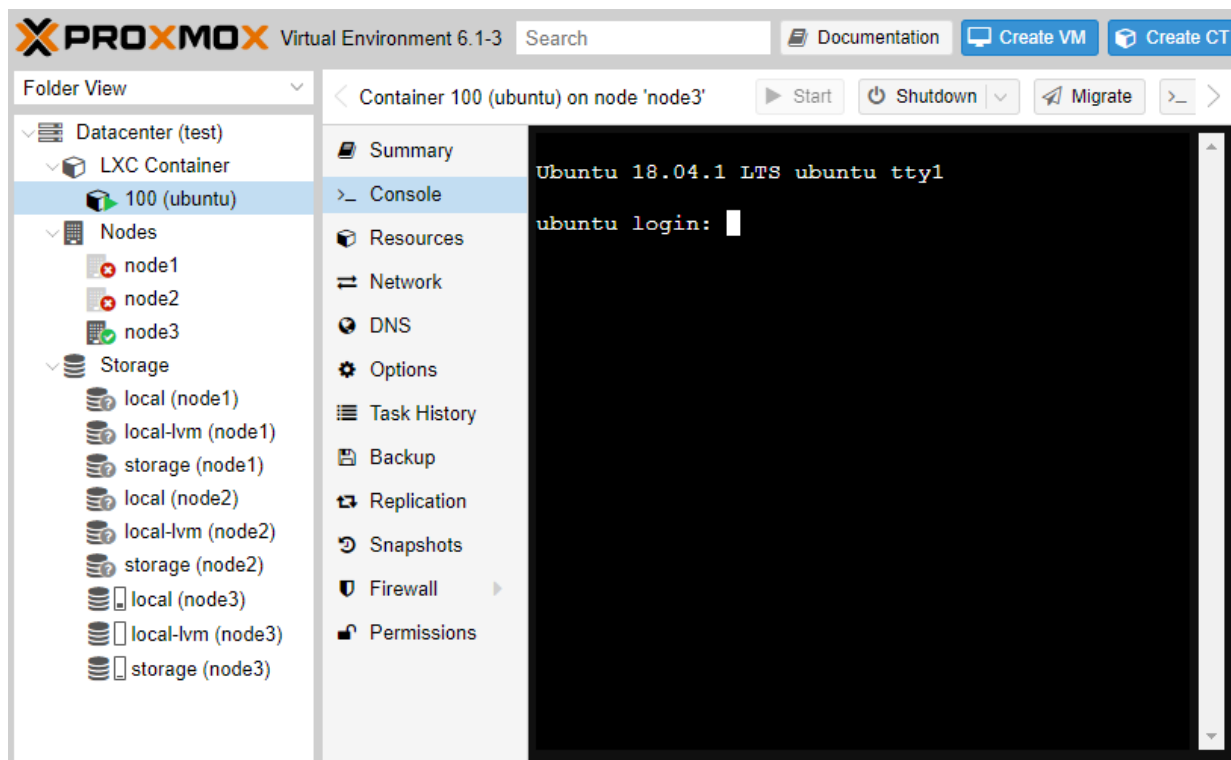
```
pvecm expected 1
```

The screenshot shows the Proxmox VE 6.1-3 interface after the command 'pvecm expected 1' was executed. The 'Status' section now shows 'quorum' as 'OK'. The 'master' is 'node3 (active, Wed Dec 11 14:19:22 2019)'. The 'Irm' shows 'node1 (old timestamp - dead?, Wed Dec 11 13:46:59 20...)', 'node2 (old timestamp - dead?, Wed Dec 11 14:02:43 20...)', and 'node3 (idle, Wed Dec 11 14:19:25 2019)'. The 'Resources' section remains the same, showing 'ct:100' in state 'started' on 'node2'.

Type	Status
quorum	OK
master	node3 (active, Wed Dec 11 14:19:22 2019)
Irm	node1 (old timestamp - dead?, Wed Dec 11 13:46:59 20...)
Irm	node2 (old timestamp - dead?, Wed Dec 11 14:02:43 20...)
Irm	node3 (idle, Wed Dec 11 14:19:25 2019)

ID	State	Node	Name	Max.
ct:100	started	node2		2

Спустя 2 минуты механизм НА отработал корректно и не найдя node2 запустил нашу VM на node3.



Как только мы включили обратно node1 и node2, работа кластера была полностью восстановлена. Обратите внимание, что обратно на node1 VM самостоятельно не мигрирует, но это можно сделать вручную.

Подводим итоги

Мы рассказали вам про то, как устроен механизм кластеризации в Proxmox, а также показали, как настраивается HA для виртуальных машин и контейнеров. Грамотное использование кластеризации и HA значительно повышает надежность инфраструктуры, а также обеспечивает восстановление после сбоев.

Перед тем как создавать кластер, нужно сразу планировать для каких целей он будет использоваться и насколько его потребуется масштабировать в будущем. Также нужно проверить сетевую инфраструктуру на готовность работы с минимальными задержками, чтобы будущий кластер работал без сбоев.

Расскажите нам — используете ли вы возможности кластеризации в Proxmox? Ждем вас в комментариях.

Предыдущие статьи на тему гипервизора Proxmox VE:

[Выделенные серверы и оборудование](#)

