

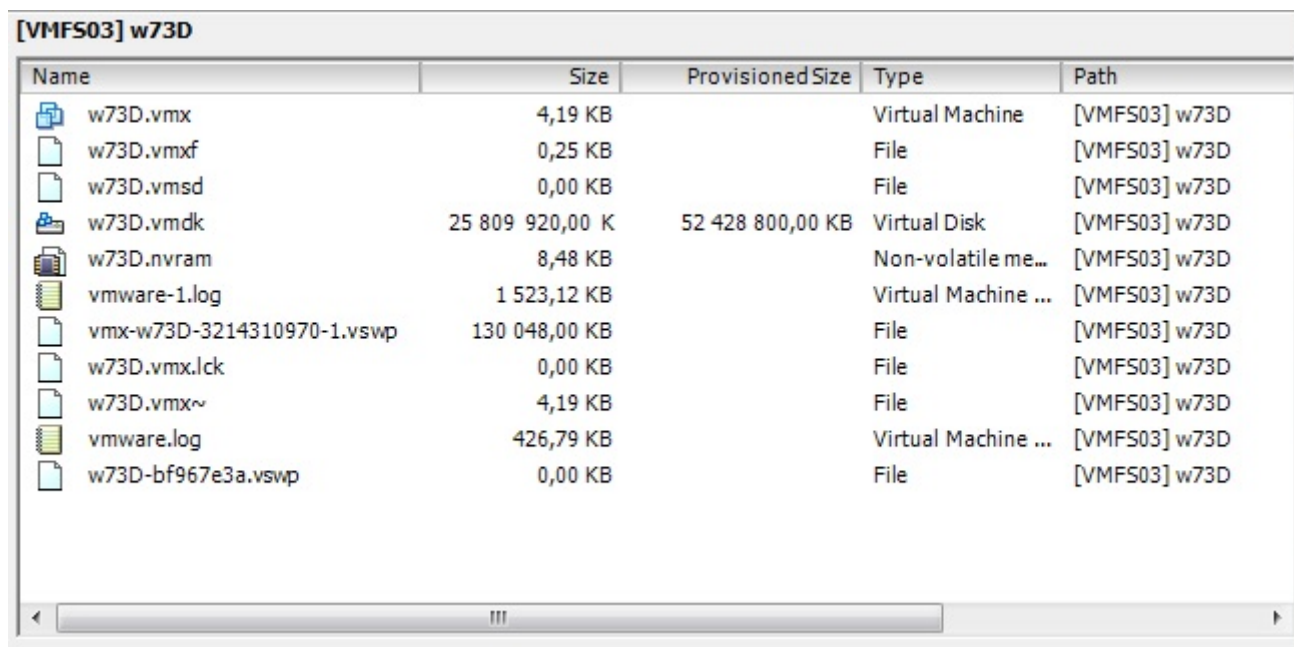
# Блог доктора Добрянского

## Строим кластер VMware: Часть 4 — общее хранилище

Итого, по мотивам предыдущих частей данной эпопеи: раз, два, три, мы имеем парочку хостов **ESXi** и **vCenter Server**, который этими хостами умеет управлять.

Следующее обязательное условие для построения кластера — общее хранилище с кластерной файловой системой **VMware** — **VMFS** (или **NFS** ресурс). Данное хранилище должно быть доступно одновременно всем узлам будущего кластера и на нем будут храниться файлы виртуальных машин. В двух словах про эти самые файлы.

Каждая виртуальная машина для гипервизора **ESXi**, это с одной стороны — набор файлов на диске, с другой стороны — процесс, или набор процессов в которые завернуты процессы гостевой операционной системы. Ниже на картинке — примерный список файлов типичной виртуальной машины:



Name	Size	Provisioned Size	Type	Path
w73D.vmx	4,19 KB		Virtual Machine	[VMFS03] w73D
w73D.vmx.f	0,25 KB		File	[VMFS03] w73D
w73D.vmsd	0,00 KB		File	[VMFS03] w73D
w73D.vmdk	25 809 920,00 K	52 428 800,00 KB	Virtual Disk	[VMFS03] w73D
w73D.nvram	8,48 KB		Non-volatile me...	[VMFS03] w73D
vmware-1.log	1 523,12 KB		Virtual Machine ...	[VMFS03] w73D
vmx-w73D-3214310970-1.vswp	130 048,00 KB		File	[VMFS03] w73D
w73D.vmx.lck	0,00 KB		File	[VMFS03] w73D
w73D.vmx~	4,19 KB		File	[VMFS03] w73D
vmware.log	426,79 KB		Virtual Machine ...	[VMFS03] w73D
w73D-bf967e3a.vswp	0,00 KB		File	[VMFS03] w73D

В данном примере машина называется w73D, это имя фигурирует в названиях ее файлов, в большинстве файлов отличается только расширение.

w73D.vmx — содержит параметры конфигурации виртуальной машины, такие как размер ОЗУ, количество ядер процессора, конфигурацию устройств ввода-вывода и множество других параметров

w73D.vmdk — файл виртуального диска, который видит гостевая операционная система как физический винчестер. Если добавить еще парочку дисков в виртуальную машину — файлов с расширением \*.vmdk тоже будет несколько

w73D.nvram — файл состояния BIOS виртуальной машины

w73D-bf967e3a.vswp — swap файл виртуальной машины, который создается во время запуска VM и его размер равняется размеру незарезервированной оперативной памяти виртуальной машины (в данном примере вся память VM зарезервирована) и используется в случае конкуренции за оперативную память между виртуальными машинами

Это основные файлы, из которых состоит VM. Есть также дополнительные файлы, которые используются для системных нужд, некоторые из них:

vmware.log — здесь и так понятно: лог системных событий, связанных с виртуальной машиной

w73D.vmsd — информация о снапшотах виртуальной машины

w73D.vmx.lock — используется для блокировки VM от запуска одновременно на двух узлах

И т.д... лезть в дебри и перечислять все не буду, статья ознакомительная.

В кластерной конфигурации **VMware vSphere** все файлы виртуальных машин, которые предполагается защищать с помощью **VMware HA** и двигать с помощью **VMware DRS** и **vMotion** предполагается хранить на общем хранилище (в контексте **VMware — Datastore**) которое доступно всем узлам кластера одновременно.

Для этих целей можно использовать конечно же **Fibre Channel**, **FCoE**, **iSCSI** или даже **NFS** хранилище. Конечно, хорошо если у вас есть модная СХД с подключением по **FC** или **FCoE**, там вопросов практически не возникает, но это дорого (если говорить про ДЕМО-стенд), и даже крутые компании не всегда могут себе позволить юзать дорогущий сторедж в качестве «песочницы».

Поэтому рассмотрим **iSCSI** и **NFS** как вариант общей хранилки «для бедных», хотя бы потому, что для демонстрации его можно собрать на коленке за полчаса (но фирменная СХД с подключением по **iSCSI** или **NFS** тоже стоит хороших денег, дружно вспомним про NetApp ;))

Если выбирать между **iSCSI** и **NFS** то здесь тоже есть небольшая диллема, у каждого варианта есть недостатки, если в двух словах, то **iSCSI** более глючный, а **NFS** более тормознутый (задо надежней и проще). Да, определенно есть любители айскази, которые закидают меня помидорами и скажут что все глюки от кривых ручек, но в контексте **ESXi** есть несколько официально признанных багов, при работе с **iSCSI**, особенно при использовании **Software iSCSI** адаптера со стороны **ESXi**, хотя есть конторки, пользующие такую конфигурацию в продуктиве, отловившие и поборовшие все глюки, каждому, как говорится, свое.

Кароче, для данной демонстрации я выбираю **NFS**, не смотря на его тормознутость, а вы смотрите сами. **NFS** сервером может притвориться в принципе любой линукс, но еще проще — готовое

решение на подобии **Open-E** или **Openfiler**.

Суть:

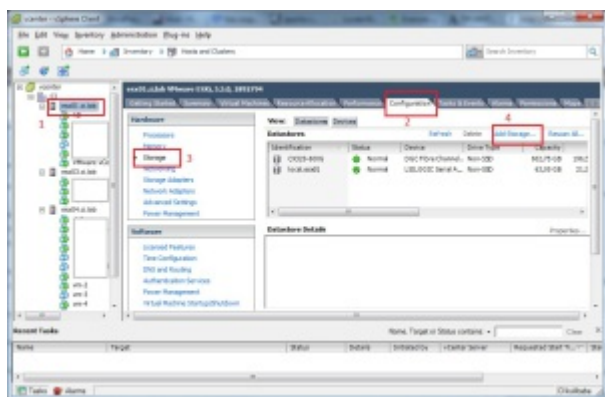
1. скачиваем готовый модуль в виде ISO или OVA/OVF
2. развертываем на физической или виртуальной машине подминая под себя побольше дискового пространства
3. прибаваем айпишник
4. создаем через веб-морду (для любителей порнушки — через командную строку)
5. ????????
6. PROFIT!

Описывать подробно установку и настройку **NFS** ресурса не буду, ибо:

1. лень
2. там все интуитивно понятно
3. не охота очередной раз отходить от темы

Допустим, **NFS** мы развернули, теперь нужно примапить его к узлам будущего кластера.

Для этого: **наступаем на узел -> лезем в Configuration -> Storage -> Add Storage** (см. картинку ниже)



В открывшемся окне выбираем «**Network File System**» и следуя инструкциям указываем:

1. Адрес или имя узла **NFS** сервера
2. Путь к папке на сервере
3. Имя хранилища, которое будет отображаться в списке датасторов

Если мы все сделали правильно, то у нас появится на вкладке «**Datastores**». Повторяем действия для всех узлов будущего кластера, при этом имя хранилища у всех узлов должно быть одинаковое.

В случае с блочным хранилищем **FC** или **FCoE**, прежде всего, **LUN** должен быть презентован всем узлам будущего кластера, после этого, создаем на нем файловую систему **VMFS** посредством «**Add Storage**» на одном из узлов, а на остальных — **ВНИМАНИЕ!!** — делаем **правой кнопкой -> «Rescan**

**for Datastores».** Если данный **LUN** добавить повторно на другом узле, пока он не успел отсканниться — получится кашка.

Что с этим всем теперь делать? — можно создать на новоиспеченном хранилище одну или несколько виртуальных машин. Содержимое хранилища можно посмотреть тыцнув на нем правой и выбрав «**Browse Datastore**». В случае с **NFS** — можно подключить его с другим серверам или юзерским машинкам, даже с виндой, т.к. **NFS** это универсальная сетевая файловая система, чего нельзя сказать про **VMFS**, которую умеет читать только **ESXi**.

На сегодня все, в следующей статье надеюсь завершить историю с созданием кластера, попытаюсь рассказать о настройках кластерных фич и тестировании их работы.

Я старался расписывать максимально подробно и доступно, но если по данной, или предыдущим статьям есть вопросы — задавайте, постараюсь ответить.

Всем мира и любви :)

Следующая статья по сабжу:

Строим кластер VMware: Часть 5 — создание и настройка кластера

## Поделиться:



Запись опубликована 3 декабря 2014 [<http://hutpu4.net/puplication/stroim-klaster-vmware-chast-4-obshhee-xranilishhe.html>] в рубрике HOWTO, Виртуализация, Операционные системы, Статьи с метками ESXi, FC, FCoE, NFS, vCenter, vSphere.

---