

Содержание

1 ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ	2
2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	2
2.1 Схема лабораторного стенда	2
2.2 Аппаратное обеспечение	3
2.3 Программное обеспечение	5
2.4 Гипервизор KVM	6
2.5 Протокол iSCSI	8
3 РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ	9
3.1 Доступ к оборудованию	9
3.2 Настройка сети	10
3.3 Настройка iSCSI адаптера на сервере. Часть 1.	13
3.4 Настройка СХД	14
3.5 Настройка iSCSI адаптера на сервере. Часть 2.	16
3.6 Настройка KVM	17
ПРИЛОЖЕНИЕ А	19

1 ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ

Тема занятия: Настройка сети хранения данных по протоколу iSCSI.

Краткое содержание занятия:

1. Обзор настройки сетей облачной инфраструктуры.
2. Обзор настройки СХД HP MSA P2000 G3 iSCSI.
3. Настройка iSCSI адаптера гипервизора KVM/QEMU в операционной системе CentOS.
4. Подключение хранилища данных (datastore), созданного в СХД к гипервизору KVM/QEMU.

2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1 Схема лабораторного стенда

В серверном помещении кафедры ВС имеется облачная подсистема, состоящая из 4-х вычислительных узлов, 2-х коммутаторов Gigabit Ethernet и системы хранения данных (СХД) с двумя контроллерами:

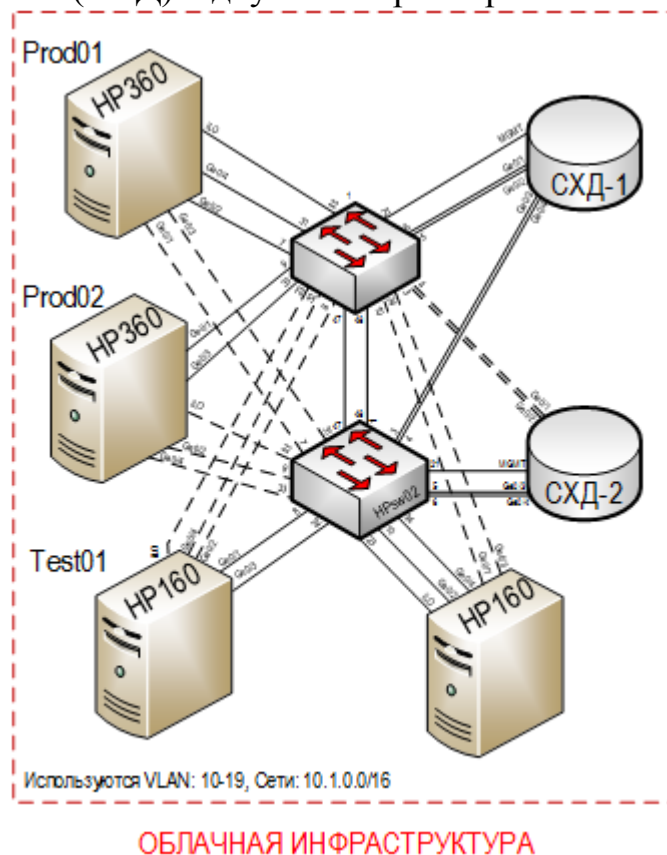


Рисунок 1 – Схема сети облачной инфраструктуры

Сервера и СХД поддерживают технологию HP iLO Management Engine. Данная технология позволяет конфигурировать большое количество параметров с помощью обычного браузера через веб-интерфейс. Таким же образом можно контролировать и конфигурировать большинство характеристик устройств. Например, состояние дисков или температуру нагрева процессоров.

Порты № 1 на обоих коммутаторах подключены к сети кафедры ВС. Сеть облачной инфраструктуры логически поделена на несколько виртуальных: VLAN 14 – сеть управления (порты № 1, 21, 22, 23) на обоих коммутаторах; VLAN 12 – сеть доступа к СХД серверов prod01 и prod02 (порты № 31, 33) на обоих коммутаторах; VLAN 15 – сеть доступа к СХД серверов test01 и test 02 (порты № 32, 34) на обоих коммутаторах; VLAN 11 – сеть виртуальных машин серверов prod01 и prod02 (порты № 7, 9) на обоих коммутаторах; VLAN 13 – сеть виртуальных машин серверов test01 и test 02 (порты № 8, 10) на обоих коммутаторах;

2.2 Аппаратное обеспечение

В качестве оборудования для выполнения практической работы будут использоваться два бездисковых сервера HP ProLiant DL360e Gen8 (рис. 2).

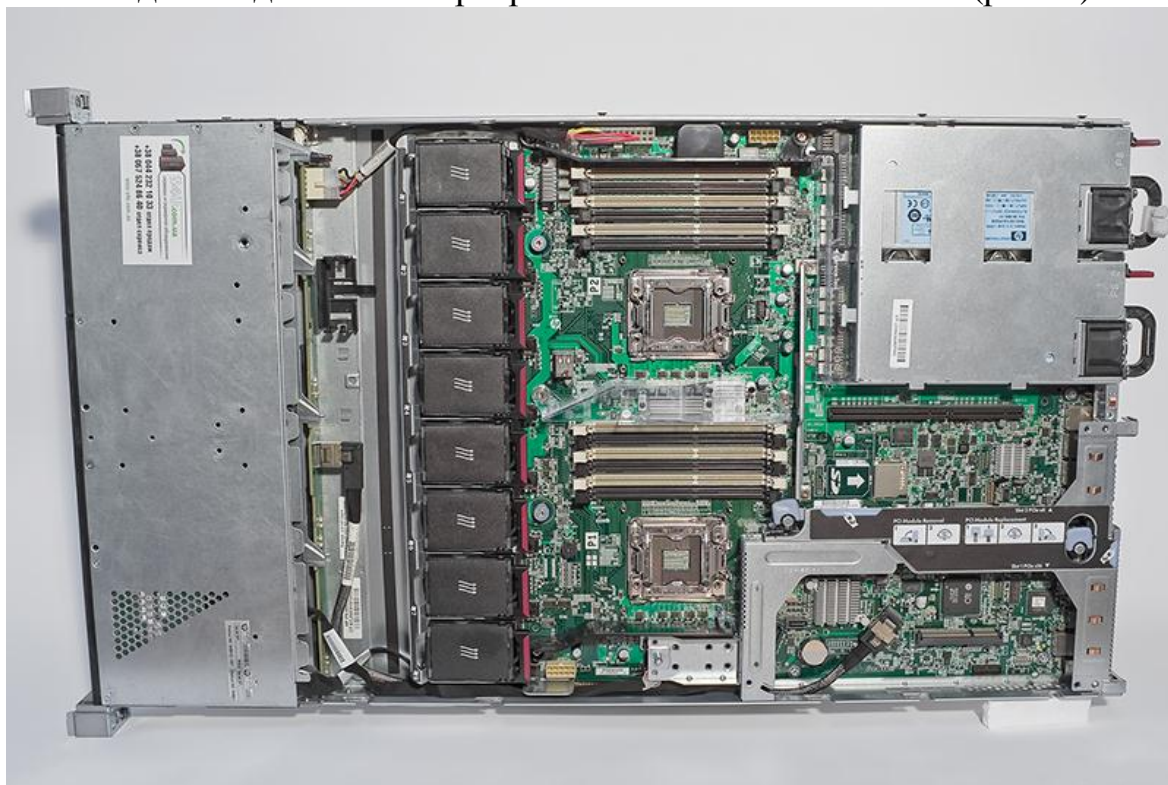


Рисунок 2 – внешний вид бездискового сервера

В серверах prod01 и prod02 установлены 2 процессора Intel(R) Xeon(R) CPU E5-2450 v2 @ 2.10GHz (рис. 3).

Processor 1	
Processor Name	Intel(R) Xeon(R) CPU E5-2450 0 @ 2.10GHz
Processor Status	OK
Processor Speed	2100 MHz
Execution Technology	8/8 cores; 16 threads
Memory Technology	64-bit Capable
Internal L1 cache	256 KB
Internal L2 cache	2048 KB
Internal L3 cache	20480 KB
Processor 2	
Processor Name	Intel(R) Xeon(R) CPU E5-2450 0 @ 2.10GHz
Processor Status	OK
Processor Speed	2100 MHz
Execution Technology	8/8 cores; 16 threads
Memory Technology	64-bit Capable
Internal L1 cache	256 KB
Internal L2 cache	2048 KB
Internal L3 cache	20480 KB

Рисунок 3 – спецификации процессоров

Каждый процессор имеет 8 физических ядер и поддерживает технологию Intel® Hyper-Threading, которая обеспечивает более эффективное использование ресурсов процессора и позволяет выполнять несколько потоков на каждом ядре. Технология Intel® Hyper-Threading повышает пропускную способность процессоров и улучшает общее быстродействие многопоточных приложений. При использовании процессора с технологией Intel® Hyper-Threading пользователь получает следующие преимущества:

- Параллельная работа с несколькими ресурсоемкими приложениями при с возможностью сохранения прежнего уровня быстродействия
- Возможность защиты системы с сохранением эффективности и управляемости при минимальном влиянии на производительность
- Широкие возможности для будущего развития и внедрения новых решений

Сервера prod-01 и prod-02 не имеют дисковых накопителей, вместо них используются карты памяти форм-фактора SD. Таким образом, выбор назначения серверов в качестве гипервизоров оправдан сравнительно небольшим размером имеющейся памяти на узле. В качестве накопителей для хранения образов гостевых систем и файлов для установки будет использоваться система хранения данных.

В качестве системы хранения данных была предоставлена СХД HP P2000 G3 iSCSI (рис. 4)



Рисунок 4 – внешний вид системы хранения данных HP P2000 G3 iSCSI

В систему хранения данных установлено 11 дисков (рис. 5).

Disks										
Health	Enclosure ID	Slot	Serial Number	Vendor	Model	Revision	Type	How Used	Status	Size
OK	1	1	9XG5SMZH000094188350	HP	MM1000FBFVR	HPD8	SAS	VDISK	Up	1000.2GB
OK	1	2	9XG5SFTK000094188PK7	HP	MM1000FBFVR	HPD8	SAS	VDISK	Up	1000.2GB
OK	1	3	9XG5RZ1X000094187YGF	HP	MM1000FBFVR	HPD8	SAS	VDISK	Up	1000.2GB
OK	1	4	9XG5S1WJ000094188NVT	HP	MM1000FBFVR	HPD8	SAS	VDISK	Up	1000.2GB
OK	1	5	9XG5RS0500009417K40B	HP	MM1000FBFVR	HPD8	SAS	VDISK	Up	1000.2GB
OK	1	6	S0M1APX30000B4169Z76	HP	EG0300FCVBF	HPD9	SAS	VDISK	Up	300.0GB
OK	1	7	S0K1CAPD0000B416DFF6	HP	EG0300FCVBF	HPD9	SAS	VDISK	Up	300.0GB
OK	1	8	S0K19QK0000B415J8PK	HP	EG0300FCVBF	HPD9	SAS	VDISK	Up	300.0GB
OK	1	9	S0K1BX4T0000B416FT28	HP	EG0300FCVBF	HPD9	SAS	VDISK	Up	300.0GB
OK	1	10	S0K1BXCZ0000B416BWW0	HP	EG0300FCVBF	HPD9	SAS	VDISK SP	Up	300.0GB
OK	1	11	S0K1BXGJ0000B416BNLC	HP	EG0300FCVBF	HPD9	SAS	VDISK SP	Up	300.0GB

Рисунок 5 – имеющиеся в системе хранения данных диски

Как видно на рисунке 5 – в системе хранения данных имеются 6 дисков модели HP EG0300FCVBF. Диски имеют возможность подключения по интерфейсу SAS, объем 300 ГБ и скорость вращения 10000 rpm. Данная модель дисков имеет быструю скорость работы и отлично подходят для установки гостевых ОС на хосте гипервизора. 4 диска объединены в виртуальный диск RAID 10 и ещё 2 диска используются для автоматической замены в случае отказа основных дисков.

Так же в системе хранения данных имеется 5 дисков модели HP MM1000FBFVR. Диски предполагает подключение по интерфейсу SAS, имеют объем 1000 ГБ и скорость вращения 7200 rpm. Данная модель отлично подходит для хранения данных за счет большого объема и отказоустойчивости. Все диски объединены в виртуальный диск RAID 5.

2.3 Программное обеспечение

Управление физическим сервером до установки операционной системы осуществляется в сети управления через порт iLo (Integrated Lights-Out). HP iLO Management Engine предоставляет следующие возможности:

- графический веб-интерфейс (GUI) — доступ к iLO из любого места через стандартный браузер;
- Virtual Power — полное удалённое управление питанием сервера;
- Remote console — платформу-независимая консоль для отображения и управления активностью удалённого сервера, например, выключение или включение.
- Virtual Serial Port — доступ к последовательному порту через сеть таких приложений как Windows® Server 2003 Emergency Management Services и сессии TTY;
- интерфейс командной строки и сценариев — гибкое использование, конфигурация и обслуживание;
- SSL шифрование — для всех данных между процессором iLO и браузером;
- диагностика процессора iLO и сервера — детализированные отчёты, статус;
- DNS/DHCP;
- удалённое обновление микропрограммы;

- IPMI (от англ. Intelligent Platform Management Interface) — интеллектуальный интерфейс управления платформой, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ через локальную сеть.

На локальной SD карте узлов для развертывания виртуальной инфраструктуры установлены операционная система CentOS, гипервизор KVM (Kernel-based Virtual Machine) и эмулятор оборудования QEMU (Quick EMUlator).

2.4 Гипервизор KVM

KVM — программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86, которая поддерживает аппаратную виртуализацию на базе Intel VT либо AMD SVM.

Программное обеспечение KVM состоит из загружаемого модуля ядра (называемого `kvm.ko`), предоставляющего базовый сервис виртуализации, процессорно-специфического загружаемого модуля `kvm-amd.ko` либо `kvm-intel.ko`, и компонентов пользовательского режима (модифицированного QEMU). Все компоненты программного обеспечения KVM открыты. Компонент ядра, необходимый для работы KVM, включён в основную ветку ядра Linux начиная с версии 2.6.20. KVM был также портирован на FreeBSD как модуль ядра. Ведётся работа по включению модификаций, необходимых для работы с KVM, в основную ветку QEMU.

Сам по себе KVM не выполняет эмуляции. Вместо этого программа, работающая в пространстве пользователя, использует интерфейс `/dev/kvm` для настройки адресного пространства гостя виртуальной машины, через него же эмулирует устройства ввода-вывода и видеоадаптер.

KVM позволяет виртуальным машинам использовать немодифицированные образы дисков QEMU, VMware и других, содержащие операционные системы. Каждая виртуальная машина имеет своё собственное виртуальное аппаратное обеспечение: сетевые карты, диск, видеокарту и другие устройства.

Программное обеспечение KVM было создано, разрабатывается и поддерживается фирмой Qumranet, которая была куплена Red Hat за \$107 млн 4 сентября 2008 года. После сделки KVM (наряду с системой управления виртуализацией oVirt) вошла в состав платформы виртуализации RHEV. KVM поддерживается Paolo Bonzini.

В настоящее время KVM взаимодействует с ядром через загружаемый модуль ядра. Поддерживаются разнообразные гостевые операционные системы, такие как Linux, BSD, Solaris, Windows, Haiku, ReactOS и AROS Research Operating System. Модифицированная версия KVM (`qemu`) может работать на Mac OS X.

В архитектуре KVM, представленной на рисунке 1, виртуальная машина выполняется как обычный Linux-процесс, запланированный стандартным планировщиком Linux. На самом деле каждый виртуальный процессор представляется как обычный Linux-процесс. Это позволяет KVM пользоваться всеми возможностями ядра Linux. Эмуляцией устройств управляет модифицированная версия `qemu`, которая обеспечивает эмуляцию BIOS, шины PCI,

шины USB, а также стандартный набор устройств, таких как дисковые контроллеры IDE и SCSI, сетевые карты и т.д.

Поскольку виртуальная машина реализована как Linux-процесс, она использует стандартную модель безопасности Linux для изоляции и управления ресурсами. С помощью SELinux (Security-Enhanced Linux) ядро Linux добавляет обязательные средства контроля доступа, многоуровневые и разнообразные средства защиты, а также управляет политикой безопасности. SELinux обеспечивает строгую изоляцию ресурсов и ограничивает подвижность процессов, запущенных в ядре Linux.

KVM наследует мощные функции управления памятью от Linux. Память виртуальной машины хранится так же, как память любого другого Linux-процесса, и может заменяться, копироваться большими страницами для повышения производительности, обобщаться или сохраняться в файле на диске.

Поддержка технологии NUMA позволяет виртуальным машинам эффективно обращаться к памяти большого объема.

KVM поддерживает новейшие функции виртуализации памяти от производителей процессоров, в частности, Intel Extended Page Table (EPT) и AMD Rapid Virtualization Indexing (RVI), для минимизации загрузки процессора и достижения высокой пропускной способности.

Обобщение страниц памяти поддерживается с помощью функции ядра Kernel Same-page Merging (KSM). KSM сканирует память каждой виртуальной машины, и если какие-то страницы памяти виртуальных машин идентичны, объединяет их в одну страницу, которая становится общей для этих виртуальных машин и хранится в единственной копии. Если гостевая система пытается изменить эту общую страницу, ей предоставляется собственная копия.

KVM может использовать любой носитель, поддерживаемый Linux, для хранения образов виртуальных машин, в том числе локальные диски с интерфейсами IDE, SCSI и SATA, Network Attached Storage (NAS), включая NFS (Network File System - Сетевая файловая система) и SAMBA/CIFS. Для улучшения пропускной способности системы хранения данных и резервирования может использоваться многопоточный ввод/вывод.

Опять же, поскольку KVM входит в состав ядра Linux, может использоваться проверенная и надежная инфраструктура хранения данных с поддержкой всех ведущих производителей; его набор функций хранения проверен на многих производственных установках.

KVM поддерживает образы виртуальных машин в распределенных файловых системах, таких как Global File System (GFS2), так что они могут разделяться несколькими хостами или обобщаться с использованием логических томов. Поддержка тонкой настройки (thin provisioning) образов дисков позволяет оптимизировать использование ресурсов хранения данных, выделяя их не сразу все наперед, а только тогда, когда этого требует виртуальная машина. Собственный формат дисков для KVM — QCOW2 — обеспечивает поддержку снимков в текущем состоянии и обеспечивает несколько уровней таких снимков, а также сжатие и шифрование.

KVM поддерживает динамическую миграцию, обеспечивая возможность перемещения работающих виртуальных машин между физическими узлами без прерывания обслуживания. Динамическая миграция прозрачна для пользователей: виртуальная машина остается включенной, сетевые соединения — активными, и пользовательские приложения продолжают работать, в то время как виртуальная машина перемещается на новый физический сервер.

Кроме динамической миграции, KVM поддерживает сохранение копии текущего состояния виртуальной машины на диск, позволяя хранить ее и восстанавливать позднее.

KVM поддерживает гибридную виртуализацию, когда паравиртуализированные драйверы установлены в гостевой операционной системе, что позволяет виртуальным машинам использовать оптимизированный интерфейс ввода/вывода, а не эмулируемые устройства, обеспечивая высокую производительность ввода/вывода для сетевых и блочных устройств.

Гипервизор KVM использует стандарт VirtIO, разработанный IBM и Red Hat совместно с Linux-сообществом для паравиртуализированных драйверов; это независимый от гипервизора интерфейс для создания драйверов устройств, позволяющий нескольким гипервизорам использовать один и тот же набор драйверов устройств, что улучшает взаимодействие между гостевыми системами.

KVM унаследовал производительность и масштабируемость Linux, поддерживая виртуальные машины с 16 виртуальными процессорами и 256 ГБ оперативной памяти, а также хост-системы с 256 ядрами и более 1 ТБ ОЗУ. Он может обеспечить:

- производительность в 95-135% по сравнению с "голым железом" в реальных корпоративных приложениях, таких как SAP, Oracle, LAMP и Microsoft Exchange;
- свыше миллиона сообщений в секунду и менее чем 200-мкс задержку в виртуальных машинах, работающих на стандартном сервере;
- максимальные уровни консолидации более чем с 600 виртуальными машинами, выполняющими корпоративные приложения, на одном сервере.

Это означает, что KVM допускает виртуализацию самых требовательных рабочих нагрузок [1].

2.5 Протокол iSCSI

iSCSI (Internet Small Computer System Interface) – протокол сети хранения данных (англ. Storage Area Network, SAN), который базируется на TCP/IP и разработан для установления взаимодействия и управления системами хранения данных, серверами и клиентами (рис. 5).

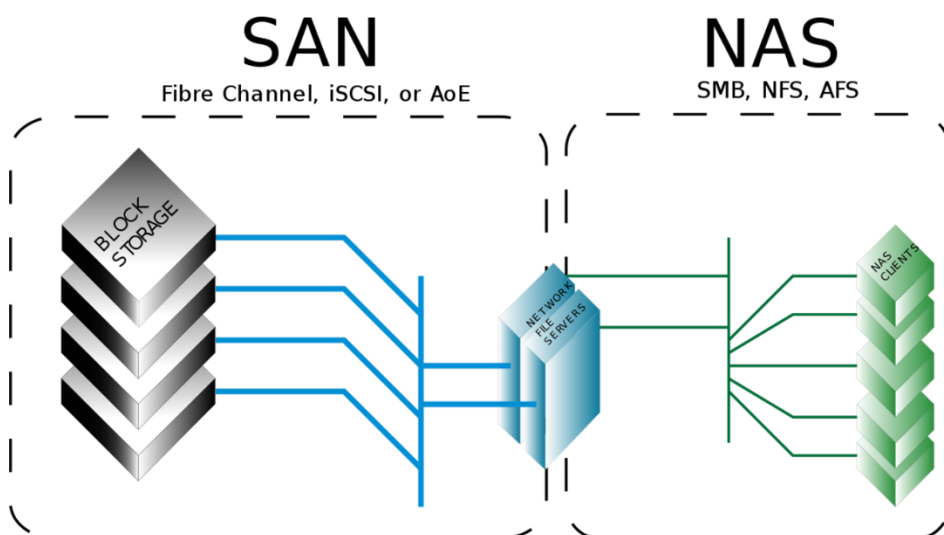


Рисунок 5 – сети хранения (SAN) и сети доступа (NAS) к данным

iSCSI описывает:

- Транспортный протокол для SCSI, который работает поверх TCP;
- Механизм инкапсуляции SCSI команд в IP сети;
- Протокол для нового поколения систем хранения данных, которые будут использовать «родной» TCP/IP;

Протокол iSCSI является стандартизованным по RFC 7143. Существует множество коммерческих и некоммерческих реализаций этого протокола [2, 3].

iSCSI построен на двух наиболее часто используемых протоколах:

SCSI – протокол обмена блоками между компьютером и хранилищем;

IP – сетевой транспортный протокол, широко применяемый в корпоративных сетях Ethernet.

Считается, что iSCSI является бюджетным вариантом для внедрения, и это связано с тем, что администрирование таких хранилищ достаточно простое несмотря на то, что для обеспечения отказоустойчивости необходимо выстроить выделенную сеть для iSCSI.

Производительность iSCSI сильно зависит от существующей инфраструктуры Ethernet (в последнее время принято считать, что минимальная рекомендуемая скорость Ethernet – 10 Гбит).

В рамках данной практической работы необходимо настроить контроллер iSCSI на вычислительных узлах. В результате разделы на СХД подключаются по сети к гипервизору и используются как локальные разделы обычного жесткого диска.

3 РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ

3.1 Доступ к оборудованию

Доступ к серверу prod01 осуществляется по доменному имени linux.csc.sibsutis.ru, порт 7701. Логин и пароль для авторизации необходимо получить у преподавателя.

Авторизация происходит под пользователем с правами администратора. Такой пользователь может выполнять действия аналогичные пользователю root, однако для этого ему перед выполняемыми командами необходимо писать команду sudo. **БУДТЕ ОЧЕНЬ ВНИМАТЕЛЬНЫ! НЕРАБОЧИЙ СЕРВЕР = НЕДОПУСК ДО ЭКЗАМЕНА. ВСЕХ!**

3.2 Настройка сети

Настройка сети на серверах prod01 и prod02 осуществляется в соответствии с картой сети узла (рис. 6), на которой обозначены все основные атрибуты.

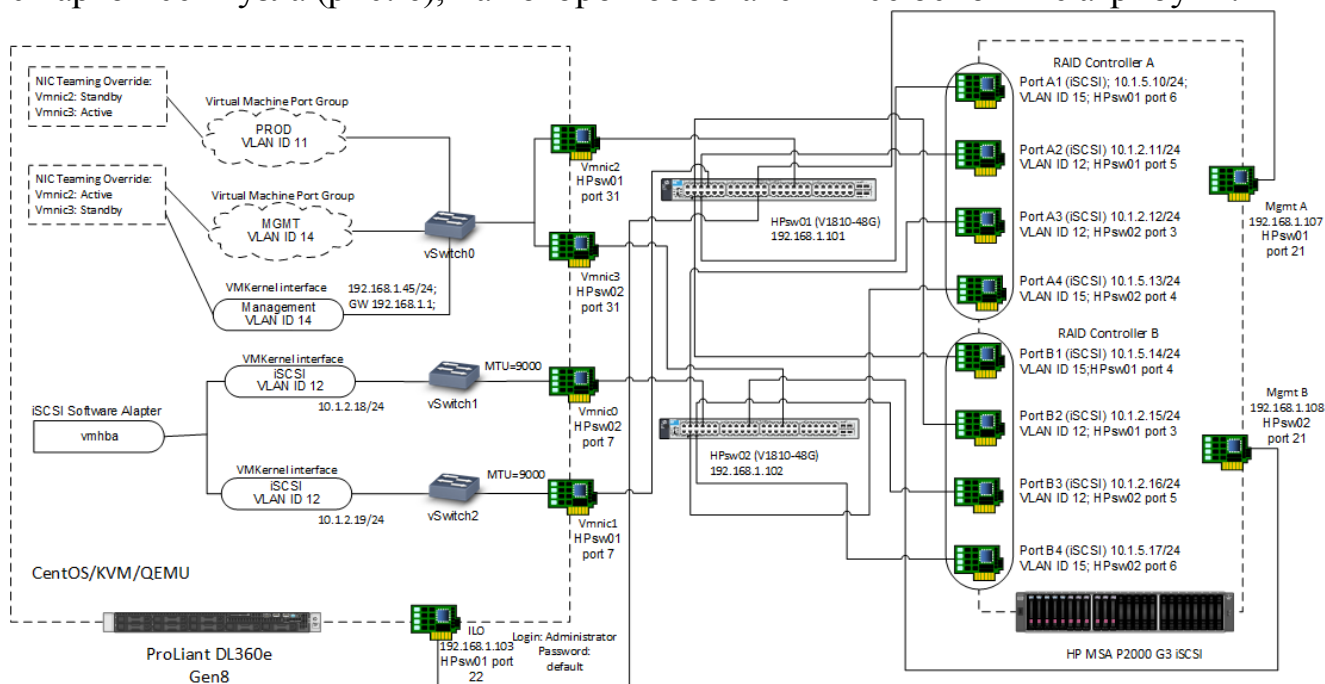


Рисунок 6 – карта сети узла prod01.

Для начала необходимо сопоставить интерфейсы с имеющимися на карте сети. Для этого необходимо открыть карту сети и свериться с выводом команды ifconfig. Карта сети для узла prod01 представлена на рисунке 6.

На карте сети у сервера prod01 имеется 5 портов: 1 интерфейс iLo и интерфейсы Vmnic0-Vmnic3. В операционной системе последним сопоставлены интерфейсы eno1-eno4.

Интерфейс eno3 находится в управляющей подсети и отвечает за удаленный доступ к операционной системе с гипервизором сервера. Например, SSH подключение должно выполняться именно по адресу 192.168.1.45 (для узла prod01). На рис. 7 показаны настройки коммутаторов для VLAN 14. Можно увидеть, что коммутатор пропускает нетегированный (Untag) трафик между портами 1, 21-23, 31-34. Под нетегированным трафиком подразумеваются пакеты в заголовке которых отсутствует номер VLAN. Трафик с других портов в данном VLAN не пропускается (Exclude).

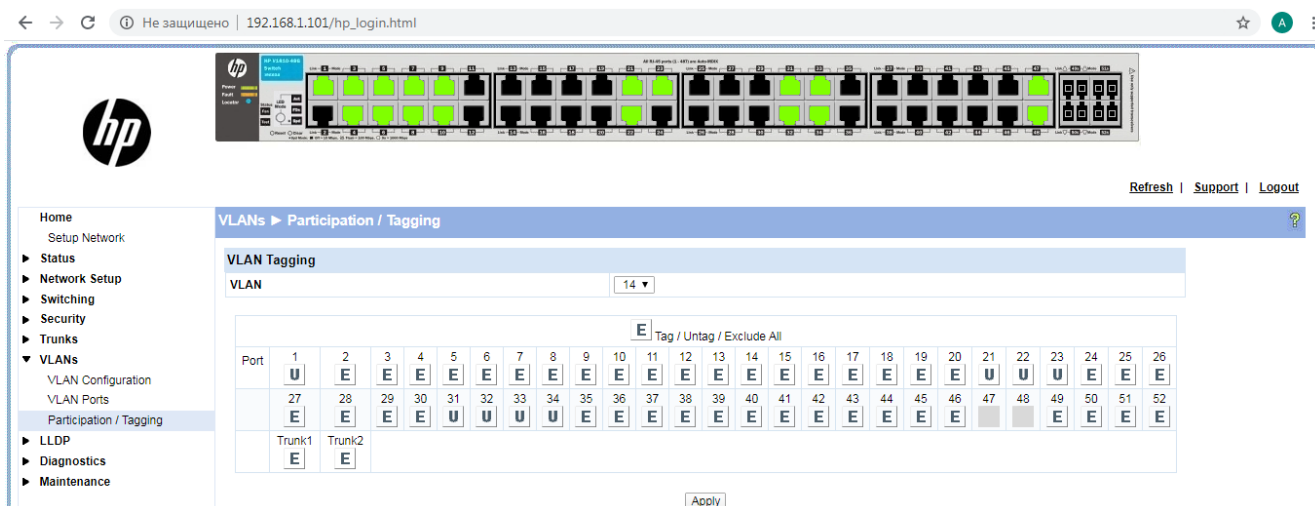


Рисунок 7 – Настройки VLAN 14 на коммутаторе HP

Интерфейсы eno1 и eno2 предназначены для подключения к СХД. Дублирование интерфейсов и маршрутов сделано в целях обеспечения отказоустойчивости. Для конфигурации интерфейсов будет использовать самый простой способ – редактирование конфигурационных скриптов интерфейсов. Для этого перейдем в папку /etc/sysconfig/network-scripts/. В этой папке находятся конфигурационные скрипты интерфейсов. Конфигурационные скрипты можно редактировать с помощью обычного текстового редактора. Для редактирования необходимо открыть скрипт и внести изменения. Пример конфигурации интерфейса eno2 (рисунок 8).

```
GNU nano 2.3.1                               Файл: /etc/sysconfig/network-scripts/ifcfg-eno2
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
IPADDR=10.1.2.19
NETMASK=255.255.255.0
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=eno2
UUID=39f2752f-1ab0-44a5-b921-b2fde53c70a5
DEVICE=eno2
ONBOOT=yes
```

Рисунок 8 – конфигурационный скрипт интерфейса eno2

- **BOOTPROTO=static** – значит, что используется статическая маршрутизация, то есть ip адрес задается явно в поле IPADDR.
- **NETMASK=255.255.255.0** – поле, в котором указывается сетевая маска интерфейса. В нашем случае используется 24 битная маска.
- **ONBOOT=yes** – в данном поле указывается необходимость «поднятия» интерфейса при загрузке системы.

После окончания конфигурирования интерфейса необходимо перезапустить сеть.

Вывод команды ifconfig (рис. 9) позволяет проверить состояние интерфейсов сервера prod01.

```

eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.2.18 netmask 255.255.255.0 broadcast 10.1.2.255
    inet6 fe80::f292:1cff:fe10:c638 prefixlen 64 scopeid 0x20<link>
    ether f0:92:1c:10:c6:38 txqueuelen 1000 (Ethernet)
    RX packets 53964 bytes 17583958 (16.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 91747 bytes 12461622 (11.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device memory 0xf7e00000-f7efffff

eno2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.2.19 netmask 255.255.255.0 broadcast 10.1.2.255
    inet6 fe80::f292:1cff:fe10:c639 prefixlen 64 scopeid 0x20<link>
    ether f0:92:1c:10:c6:39 txqueuelen 1000 (Ethernet)
    RX packets 13 bytes 780 (780.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 740 (740.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device memory 0xf7c00000-f7cfffff

eno3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.45 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::f292:1cff:fe10:c63a prefixlen 64 scopeid 0x20<link>
    ether f0:92:1c:10:c6:3a txqueuelen 1000 (Ethernet)
    RX packets 138926 bytes 19427851 (18.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14687 bytes 1485870 (1.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device memory 0xf7a00000-f7afffff

eno4: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::f292:1cff:fe10:c63b prefixlen 64 scopeid 0x20<link>
    ether f0:92:1c:10:c6:3b txqueuelen 1000 (Ethernet)
    RX packets 195718 bytes 18641660 (17.7 MiB)
    RX errors 0 dropped 1818 overruns 0 frame 0
    TX packets 17 bytes 1298 (1.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device memory 0xf7800000-f78fffff

```

Рисунок 9 – Вывод команды ifconfig

Для доступа к СХД предусмотрен VLAN 12. Настройки коммутатора представлены на рис. 10. Можно заметить, что нетегированный трафик пропускается только между портами 3,5,7,9 и через Trunk1, соединяющий коммутаторы друг с другом, для большего числа маршрутов от сервера до СХД.

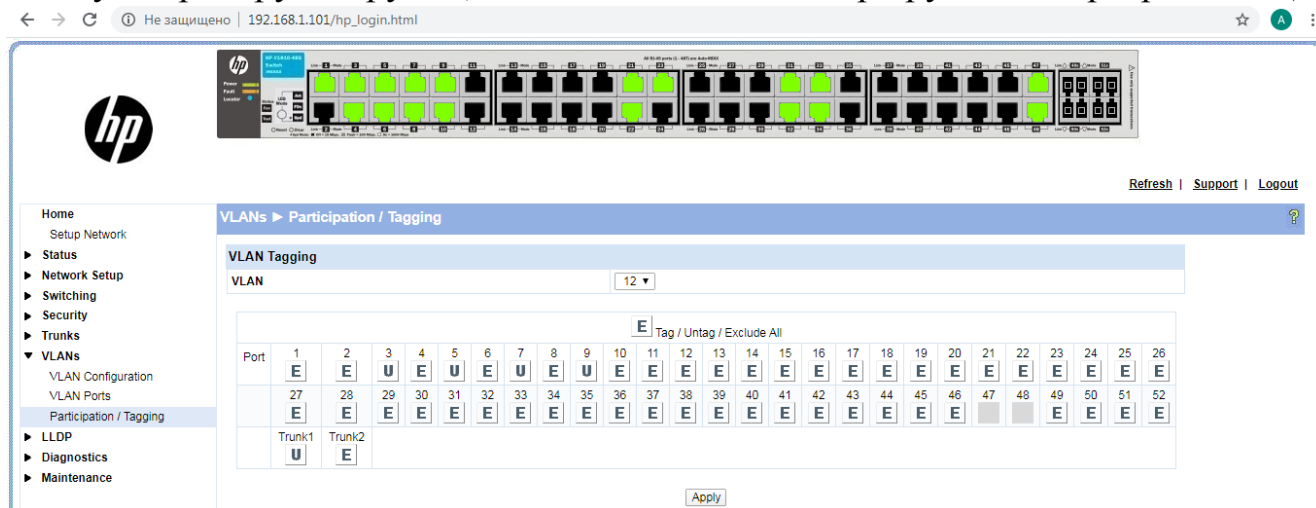


Рисунок 10 – Настройки VLAN 12 на коммутаторе HP

Настройка сетевых интерфейсов на СХД представлена на рис. 11. Обратите внимание, что доступ к разделам на СХД предоставляется только после авторизации по протоколу CHAP. На карте сети на рис. 6 можно увидеть, что интерфейсы СХД подключены к портам 3-6 на обоих коммутаторах. Ещё важной особенностью является отсутствие возможности указать номер VLAN при настройке сетевых интерфейсов. Это значит, что СХД не поддерживает тегированный трафик.

Настройка iSCSI target (цель = подключение к СХД) происходит через веб-интерфейс на СХД. Для начала необходимо добавить узлы серверов в список хостов системы хранения данных. Предварительно необходимо:

- 1) проверить доступность интерфейсов СХД с помощью команды ping;
- 2) отправить iSCSI Discovery запрос на один из интерфейсов СХД.

```
sudo iscsiadm -m discovery -t st -p 10.1.2.11 -t st
```

После успешного запроса сервер появится в списке хостов СХД. Для удобства настройки на СХД для каждого узла может быть создан псевдоним (например, имя prod01), который можно использовать вместо IQN.

```
admin@prod-01:~$ cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:8af6a4258db
[admin@prod-01 ~]$ sudo iscsiadm -m discovery -t st -p 10.1.2.11 -t st
10.1.5.10:3260,1 iqn.1986-03.com.hp:storage.p2000g3.13421aff31
10.1.5.14:3260,2 iqn.1986-03.com.hp:storage.p2000g3.13421aff31
10.1.2.11:3260,3 iqn.1986-03.com.hp:storage.p2000g3.13421aff31
10.1.2.15:3260,4 iqn.1986-03.com.hp:storage.p2000g3.13421aff31
10.1.2.12:3260,5 iqn.1986-03.com.hp:storage.p2000g3.13421aff31
10.1.2.16:3260,6 iqn.1986-03.com.hp:storage.p2000g3.13421aff31
10.1.5.13:3260,7 iqn.1986-03.com.hp:storage.p2000g3.13421aff31
10.1.5.17:3260,8 iqn.1986-03.com.hp:storage.p2000g3.13421aff31
[admin@prod-01 ~]$
```

Зная имя инициатора, можно найти хост в списке хостов СХД и начать назначение виртуальных разделов дисков на сервер.

3.4 Настройка СХД

Как уже было сказано выше, на СХД настроены 2 виртуальных диска: RAID 5 и RAID 10.

The screenshot shows the Storage Management Utility (SMU) web interface. The left sidebar contains a tree view with the following structure:

- Uninitialized Name (P2000 G3 iSCSI)
 - Logical
 - Vdisks
 - vd01 (RAID5)
 - vd02 (RAID10)
 - Hosts
 - Physical
 - Enclosure 1

The main panel displays the configuration for **vd02 (RAID10)**. The **Vdisk Overview** table shows the following details:

Health	Component	Count	Capacity	Storage Space
OK	Vdisk		599.4GB	449.4GB / 150.0GB
OK	Disks	6	599.4GB	449.4GB / 150.0GB
OK	Volumes	3	449.4GB	449.4GB
OK	Snap Pools	0	0.0KB	

Below the Vdisk Overview table, the **Disk Sets, Total Space: 1.8TB** section shows the following details:

Type	Disk Type	Disks	Size
RAID1-1	SAS	1 2	600GB
RAID1-2	SAS	1 2	600GB
SPARE	SAS	1 2	600GB

The **Enclosures Front View** table shows the following details:

Health	Name	Type	State	Size	Enclosure	Serial Number	Status
OK	Disk-1.6	SAS	RAID1-2	300.0GB	Enclosure-1	S0K1APX30000B4169276	Up
OK	Disk-1.7	SAS	RAID1-2	300.0GB	Enclosure-1	S0K1CAPD0000B4160FF8	Up
OK	Disk-1.8	SAS	RAID1-1	300.0GB	Enclosure-1	S0K1BQK00000B415J8PK	Up
OK	Disk-1.9	SAS	RAID1-1	300.0GB	Enclosure-1	S0K1BX4T0000B416FT28	Up
OK	Disk-1.10	SAS	SPARE	300.0GB	Enclosure-1	S0K1BXCZ0000B416WVQ	Up
OK	Disk-1.11	SAS	SPARE	300.0GB	Enclosure-1	S0K1BXGJ0000B416BNLC	Up

Диски поделены на разделы. По 150 Гб на RAID 10 и по 1 Тб на RAID 5 для каждого из серверов. На виртуальном диске RAID 5 для каждого студента создан один раздел с его фамилией, который необходимо подключить к серверу prod01 под своей учетной записью.

← → ↻ Не защищено | 192.168.1.107

Storage Management Utility

System Status

System Time 2020-04-21 18:15:52

System Events 0 0 0 0 100

Configuration View

Uninitialized Name (P2000 G3 iSCSI)

Logical

Vdisks

vd01 (RAID5)

- Volume vd01_prod02 (998.9GB)
- Volume vd01_test01 (998.9GB)
- Volume vd01_test02 (999.7GB)
- Volume Berezina (999.9MB)
- Volume Evtushenko (999.9MB)
- Volume Simonova (999.9MB)
- Volume Smirnyh (999.9MB)
- Volume Chernenko (999.9MB)
- Volume Shavrin (999.9MB)
- Volume Varaksina (999.9MB)
- Volume Eremin (999.9MB)
- Volume Petruk (999.9MB)
- Volume Sviridov (999.9MB)
- Volume Smirnov (999.9MB)
- Volume Timofeev (999.9MB)

Hosts

- test02
- test01
- prod01
- prod02

Physical

- Enclosure 1

vd01 (RAID5)

View Provisioning Configuration Tools Help

vd01 (RAID5) > View > Overview

Vdisk Overview

Details about a specific vdisk

Vdisk Overview					
Health	Component	Count	Capacity	Storage Space	
OK	Vdisk		4.0TB	3.0TB	987.0GB
OK	Disks	5	4.0TB	3.0TB	987.0GB
OK	Volumes	15	3.0TB	3.0TB	
	Snap Pools	0	0.0KB		

Disk Sets, Total Space: 5.0TB: 4.0TB 1.0TB

Type	Disk Type	Disks	Size
RAID5	SAS	1 2 3 4 5	5 000GB
SPARE	SAS		0GB

Tabular Graphical

Enclosures Front View

Health	Name	Type	State	Size	Enclosure	Serial Number	Status
OK	Disk-1.1	SAS	RAID5	1000.2GB	Enclosure-1	9XG5SMZH000094188350	Up
OK	Disk-1.2	SAS	RAID5	1000.2GB	Enclosure-1	9XG5SFTK000094188PK7	Up
OK	Disk-1.3	SAS	RAID5	1000.2GB	Enclosure-1	9XG5RZ1X000094187YGF	Up
OK	Disk-1.4	SAS	RAID5	1000.2GB	Enclosure-1	9XG5S1WJ000094188NVT	Up
OK	Disk-1.5	SAS	RAID5	1000.2GB	Enclosure-1	9XG5RS050009417K40B	Up

Процесс назначения (map, mapping) с использованием веб интерфейса крайне прост. Кликнув правой кнопкой мыши на имени виртуального раздела во всплывающем меню нужно выбрать Provisioning/Explicit Mapping, Далее необходимо выбрать через какой порт будет доступен каждый диск из RAID для каждого хоста. Пример назначения для сервера prod01 можно увидеть на рисунке.

← → ↻ Не защищено | 192.168.1.107

Storage Management Utility

System Status

System Time 2020-04-21 19:18:37

System Events 0 0 0 0 100

Configuration View

Uninitialized Name (P2000 G3 iSCSI)

Logical

Vdisks

vd01 (RAID5)

- Volume Berezina (999.9MB)
- Volume vd01_prod02 (998.9GB)
- Volume vd01_test01 (998.9GB)
- Volume vd01_test02 (999.7GB)
- Volume Evtushenko (999.9MB)
- Volume Simonova (999.9MB)
- Volume Smirnyh (999.9MB)
- Volume Chernenko (999.9MB)
- Volume Shavrin (999.9MB)
- Volume Varaksina (999.9MB)
- Volume Eremin (999.9MB)
- Volume Petruk (999.9MB)
- Volume Sviridov (999.9MB)
- Volume Smirnov (999.9MB)
- Volume Timofeev (999.9MB)

Hosts

- test02
- test01
- prod01
- prod02

Physical

- Enclosure 1

Volume Berezina (999.9MB)

View Provisioning Configuration Tools Help

Volume Berezina (999.9MB) > Provisioning > Explicit Mappings

Explicit Volume Mappings

Modify the volume mappings to specific hosts by using the default map or explicit map settings

Select an item to modify the mapping properties to a specific host:

Type	Host ID	Name	Ports	LUN	Access
Default	iqn.1004-05.com:redhat:0f9009488fb	test02			not-mapped
Default	iqn.1004-05.com:redhat:04e4e071101e	test01			not-mapped
Explicit	iqn.1004-05.com:redhat:8a78a4255db	prod01	A2,A3,B2,B3	101	read-write
Default	iqn.1004-05.com:redhat:2eb38330650	prod02			not-mapped

Map: ☒ (Clear to remove existing mapping)

LUN: 101 Access: read-write

Select ports from the view or list below:

Graphical Tabular

Apply

На рисунке можно увидеть, что доступ к дискам сервер prod01 имеет через iSCSI порты A2, A3, B2, B3.

LUN – Logical Unit Number, уникальный идентификатор виртуального раздела в пределах сети.

Таблица соответствия LUN виртуальным разделам на СХД:

Имя раздела	LUN
Berezina	101
Evtushenko	102
Simonova	103
Smirnyh	104
Chernenko	105
Shavrin	106
Varaksina	107
Eremin	108
Petruk	109
Sviridov	110
Smirnov	111
Timofeev	112

Далее необходимо выполнить настройку протокола CHAP для защищенной авторизации. Графический интерфейс позволяет очень просто настраивать CHAP. Для этого необходимо ввести IQN хоста и ввести пароли. Результат настройки CHAP показан на рисунке 16.

Current CHAP Entries				
	Initiator Name	Initiator Secret	Mutual CHAP Name	Mutual CHAP Secret
<input type="radio"/>	iqn.1994-05.com.redhat:2eb36330650	5603791555623791	iqn.1994-05.com.redhat:2eb36330650	5603791555623792
<input type="radio"/>	iqn.1994-05.com.redhat:c4e4e0711b1e	0162199451130454	iqn.1994-05.com.redhat:c4e4e0711b1e	0162199451130455
<input type="radio"/>	iqn.1994-05.com.redhat:d6f8999488fb	5780177076508441	iqn.1994-05.com.redhat:d6f8999488fb	5780177076508442
<input type="radio"/>	iqn.1994-05.com.redhat:8aff8a4258db	0258055731614602	iqn.1994-05.com.redhat:8aff8a4258db	0258055731614603

Рисунок 16 – результат настройки CHAP для всех хостов

3.5 Настройка iSCSI адаптера на сервере. Часть 2.

После этого необходимо выполнить подключение на стороне инициатора. Это можно сделать с помощью команды `iscsiadm -m node --targetname <IQN name> -p <IP> -l`. После чего диски из системы хранения данных будут подключены согласно выполненному маппингу. Подключение дисков будет выполнено стандартно для ОС Linux. Новые диски получают наименование `/dev/sd<x>`, где `x` – следующий символ по алфавиту после последнего подключенного диска. Таким образом, если в операционной системе был определен один диск `/dev/sda`, то при подключении по iSCSI нового диска ему будет дано наименование `/dev/sdb`. Удобно посмотреть все имеющиеся в системе диски можно с помощью команды `fdisk -l`. Пример вывода этой команды представлен на рисунке 17.

```

[admin@prod-01 ~]$ sudo fdisk -l
[sudo] пароль для admin:

Disk /dev/sda: 31.3 GB, 31281119232 bytes, 61095936 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0009df2a

Устр-во Загр      Начало        Конеч        Блоки   Id  Система
/dev/sda1  *           2048         2099199     1048576  83  Linux
/dev/sda2           2099200     61095935     29498368  8e  Linux LVM

Disk /dev/mapper/centos-root: 27.1 GB, 27074232320 bytes, 52879360 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/centos-swap: 3128 MB, 3128950784 bytes, 6111232 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdb: 150.0 GB, 149999976448 bytes, 292968704 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 1048576 bytes

Disk /dev/sdc: 999.0 GB, 998999982080 bytes, 1951171840 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 1048576 bytes

[admin@prod-01 ~]$ █

```

Рисунок 17 – вывод команды fdisk -l, где /dev/sdb и /dev/sdc – диски подключенные по протоколу iSCSI

После подключения дисков каждому студенту необходимо отформатировать и монтировать свой диск в каталог /mnt/ваша_фамили. И создать файл, содержащий подтверждение вашего присутствие в данном разделе и выполнение практической работы. В дальнейшем возможно создание виртуальных машин с размещением на смонтированном вами разделе.

3.6 Настройка KVM

Виртуальные машины должны иметь доступ к сети для выхода в интернет, для этого необходимо создать виртуальный мост, связанный с интерфейсом eno4. Конфигурация интерфейса представлена на рисунке 19.


```
GNU nano 2.3.1                               Файл: /etc/sysconfig/network-scripts/ifcfg-eno4
TYPE=Ethernet
NM_CONTROLLED=no
BOOTPROTO=none
DEVICE=eno4
MTU=1500
ONBOOT=yes
BRIDGE=br0
```

Рисунок 19 – конфигурация интерфейса eno4

Как видно на рисунке 19 интерфейс использует для связи мост br0. Конфигурация моста br0 представлена на рисунке 20.

```
GNU nano 2.3.1                               Файл: /etc/sysconfig/network-scripts/ifcfg-br0
TYPE=Bridge
BOOTPROTO=none
IPADDR=192.168.1.145
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=192.168.1.1
DNS2=8.8.8.8
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
DEVICE=br0
ONBOOT=yes
```

Рисунок 20 – конфигурация виртуального моста br0

Далее нужно установить недостающие утилиты для установки виртуальной машины, это можно сделать с помощью команды `dnf install qemu-kvm libvirt libvirt-python libguestfs-tools virt-install`.

После установки недостающих утилит можно приступить к установке виртуальных машин. Для этого необходимо ввести команду, представленную на рисунке 21.

```
virt-install \
--virt-type=kvm \
--name centos7_node_1 \
--ram 16384 \
--vcpus=2 \
--os-variant=centos7.0 \
--cdrom=/mnt/MM1000FBFVR/iso/CentOS-7-x86_64-Everything-1810.iso \
--network=bridge=br0,model=virtio \
--graphics vnc,password=123 \
--disk path=/mnt/EG0300FCVBF/images/centos7_node_1/root_centos7_node_1,size=40,bus=virtio,format=qcow2 \
--disk path=/mnt/MM1000FBFVR/images/centos7_node_1/home_centos7_node_1,size=40,bus=virtio,format=qcow2
```

Рисунок 21 – команда создания и запуска установки виртуальной машины

Значения опций данной команды

- `--virt-type` – тип виртуальной машины
- `--name` – имя виртуальной машины
- `--ram` – количество памяти, предоставляемой виртуальной машине (измеряется в МБ)
- `--vcpus` – количество процессорных ядер
- `--os-variant` – тип операционной системы
- `--cdrom` – образ диска, монтируемый в CDrom при запуске виртуальной машины

- `--network` – сетевой интерфейс, через который виртуальная машина получит доступ к сети
- `--graphics` – настройки графики для графической установки
- `--disk` – адрес образа диска, который будет монтироваться в операционную систему. Создать образ диска можно с помощью команды `qemu-img create -f qcow2 <название диска> <размер диска>`, где `qcow2` – формат образа диска

СПИСОК ИСТОЧНИКОВ

1. KVM (Kernel-based Virtual Machine) / Национальная библиотека им. Н. Э. Баумана Bauman National Library URL: [https://ru.bmstu.wiki/KVM_\(Kernel-based_Virtual_Machine\)](https://ru.bmstu.wiki/KVM_(Kernel-based_Virtual_Machine)) (дата обращения 12.05.2019)
2. iSCSI – Википедия URL: <https://ru.wikipedia.org/wiki/ISCSI> (дата обращения 03.05.2019)
3. iSCSI и другие URL: <https://www.ixbt.com/storage/iSCSI.shtml> (дата обращения 03.05.2019)