СОДЕРЖАНИЕ

АННОТАЦИЯ

В данном документе приведено описание средств ROSA Enterprise Linux Server для настройки работы сервисов в кластере. В руководстве приведено пошаговое описание развертывания кластерных сервисов и необходимых инструментов для обеспечения распределения нагрузки между серверами. Руководство предназначено для пользователей, знакомых с базовыми возможностями Linux.

1 Используемые обозначения

Выделение важной информации

В документе для выделения информации, на которую стоит обратить внимание, используются примечания и иконки  ,  ,  .

Примечания выделяются подчеркиванием текста и содержат дополнительную информацию о конфигурировании RELS или о дополнительных возможностях команд.

Пример:

> Примечание: на nfs-клиенте просмотреть расшаренные папки можно командой
>
> `showmount -e 192.168.68.128`
>
> где `192.168.68.128` — адрес nfs-сервера.

Иконка  служит для выделения возможностей RELS, которые существенно упрощают работу с операционной системой, или сведений о готовой конфигурации, которую можно использовать при работе с RELS.

Пример:

>  Пример содержимого конфигурационного файла для кластера на двух хостах:

```
<cluster name="mycluster" config_version="2">
   <cman two_node="1" expected_votes="1"/>
   <clusternodes>
     <clusternode name="rosa2.int" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="rosa.int" nodeid="2">
         <fence>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

Иконка  служит для того, чтобы обратить ваше внимание на те или иные особенности работы RELS. Эта информация не является критически необходимой, но мы рекомендуем строго следовать советам, приведенным с этой иконкой. Это поможет сэкономить время.

Пример:

>  Перед запуском в промышленную эксплуатацию GFS мы рекомендуем проконсультироваться со специалистами технической поддержки РОСЫ, а также провести опытную эксплуатацию инфраструктуры с GFS в течении 2-3-х месяцев, чтобы понаблюдать за устойчивостью работы хранилища.

Иконка  служит для выделения критически важной информации. Внимательно читайте эту информацию и строго следуйте рекомендациям. В противном случае у вас могут возникнуть серьезные сбои или просто не запустятся важные сервисы RELS.

Пример:

>  Примечание: иногда при соединении с хостом кластера через интерфейс luci возникает ошибка “`authentication to ricci agent failed`”. Данная проблема решается разрешением доступа к нужному порту в файерволе, а также установкой пароля ricci командой:
>
> ```
> passwd ricci
> ```
>
> на каждом хосте. Иногда также в этой ситуации стоит отключить selinux, если вы не уверены, что абсолютно правильно пользуетесь им.

1.1 Ссылки на другие разделы документа и рисунки

В документе используются ссылки на рисунки и другие разделы документа, для перехода по ссылке в PDF-версиях документа необходимо нажать клавишу CTRL и щелкнуть левой кнопкой мыши на ссылку.

Желаем приятного знакомства с возможностями RELS!

2 Создание базовой кластерной конфигурации

Кластер — это объединение вычислительных и сетевых мощностей, а также СХД для решения совместных задач, поддержки ИТ-сервисов высокой надежности и доступности. В

составе RELS существуют специальные программные средства, позволяющие управлять кластерными сервисами, реальными серверами кластерам и обеспечивать распределения нагрузки.

Первым шагом для управления кластером является создание кластерной конфигурации на хостах кластера и подключение их к кластеру. Ниже можно познакомиться с возможностями по управлению кластером на основе средств RELS как с помощью консоли, так и из графического интерфейса.

Для построения кластерной конфигурации необходимо установить пакеты (все зависимости будут автоматически разрешены при установке из репозитория РОСЫ):

– ccs
– omping
– rgmanager

Сделать это можно непосредственно с помощью менеджера пакетов `yum` в консоли или воспользовавшись утилитой «Установка и удаление программ» (см. Руководство пользователя RELS), либо при установке RELS и конфигурировании сервера. Для создания базового кластера выполните следующие операции:

1) На каждом хосте создайте конфигурационный файл: `/etc/cluster/cluster.conf`

В общем, в простейшей конфигурации необходимо указать только имена и идентификаторы хостов, участвующих в кластере. Не забудьте сохранить файл `/etc/cluster/cluster.conf` после внесения в него изменений.

Примечание: можно разрешить выполнение операций на одном хосте, в случае выхода из строя второго, добавив в конфигурационный файл следующие строчки:

```
<cman two_node="1"
     expected_votes="1"/>
```

При внесении этой опции в конфигурацию нужно будет перезапустить кластер, чтобы изменения вступили в силу.

Пример содержимого конфигурационного файла для кластера на двух хостах:

```
<cluster name="mycluster" config_version="2">
   <cman two_node="1" expected_votes="1"/>
   <clusternodes>
     <clusternode name="rosa2.int" nodeid="1">
         <fence>
         </fence>
```

```
        </clusternode>
        <clusternode name="rosa.int" nodeid="2">
            <fence>
            </fence>
        </clusternode>
    </clusternodes>
    <fencedevices>
    </fencedevices>
    <rm>
    </rm>
</cluster>
```

Где **rosa2.int** и **rosa.int** — FQDN хостов.

2) Убедитесь, что хосты умеют разрешать указанные в конфигурационном файле `clusternode name` (в нашем случае это `rosa2.int` и `rosa.int`). В любом случае, если разрешение имен не происходит, в файл `/etc/hosts` на обоих хостах можно добавить необходимую информацию, например:

```
192.168.68.129 rosa2.int
192.168.68.131 rosa.int
```

3) Валидируйте конфигурационный файл командой:

```
ccs_config_validate
```

Вы должны увидеть сообщение:

Configuration validates

Типичной ошибкой при валидации является невозможность разрешения `clusternode name`.

4) Перенесите конфигурационный файл из файла `/etc/cluster/` на каждый хост кластера, сделать это можно командой (выполняется из каталога /etc/cluster на хосте **rosa.int**):

```
scp cluster.conf user@rosa.int:/etc/cluster/cluster.conf
```

Примечание: перенос данным способом конфигурационного файла необходим при создании и первом запуске кластера, далее можно переносить конфигурационный файл с помощью команды:

```
cman_tool version -r
```

Помните, что всегда можно воспользоваться протоколом scp для переноса конфигурационных файлов, но для этого необходимо будет остановить сервисы кластера. Также в данном варианте (scp) после каждого обновления конфигурационного файла нужно будет

запускать валидацию (ccs_config_validate).

5) Запустите кластер, выполнив на каждом хосте команду:

```
service cman start
```

Вы должны увидеть следующее сообщение:

```
Starting cluster:
   Checking if cluster has been disabled at boot...        [  OK  ]
   Checking Network Manager...                             [  OK  ]
   Global setup...                                         [  OK  ]
   Loading kernel modules...                               [  OK  ]
   Mounting configfs...                                    [  OK  ]
   Starting cman...                                        [  OK  ]
   Waiting for quorum...                                   [  OK  ]
   Starting fenced...                                      [  OK  ]
   Starting dlm_controld...                                [  OK  ]
   Tuning DLM kernel config...                             [  OK  ]
   Starting gfs_controld...                                [  OK  ]
   Unfencing self...                                       [  OK  ]
   Joining fence domain...                                 [  OK  ]
```

Примечание: распространенной ошибкой в RELS является ошибка при проверки NetworkManager. Дело в том, что в RELS он используется по умолчанию и не работает совместно с сервисами кластера.

Для корректной работы сервисов кластера необходимо выключить NetworkManager и перезапустить сервис **network**, для чего выполните следующие команды (на всех хостах кластера):

– Остановите NetworkManager:

```
service NetworkManager stop
```

– Выключите NetworkManager в chkconfig:

```
chkconfig NetworkManager off
```

– В конфигурационном файле каждого сетевого интерфейса (поименованных ifcfg-X, где X — имя интерфейса, например, eth0) в директории /etc/sysconfig/network-scripts/ установите параметры (по умолчанию будут указаны обратные значения):

```
NM_CONTROLLED=no
ONBOOT=yes
```

– Перезапустите сервис network:

```
service network restart
```

– Убедитесь, что сервис network доступен в chkconfig:

```
chkconfig network on
```

– Команду `service cman start` необходимо запустить на каждом хосте кластера, чтобы каждый хост запустился как член (member) кластера (будет отмечено буквой `M` в графе `Sts`). Проверить, какие хосты запустились как `member`, можно командой:

```
cman_tool nodes
```

Если все в порядке, будет показано примерно следующее:

```
Node  Sts   Inc   Joined               Name
   1   M     36   2013-12-06 04:42:00  rosa2.int
   2   M     40   2013-12-06 04:50:24  rosa.int
```

Если какой-то хост не запустился как член кластера, то будет показано примерно следующее:

```
Node  Sts   Inc   Joined               Name
   1   M     36   2013-12-06 04:42:00  rosa2.int
   2   X      0                        rosa.int
```

3 Conga

3.1 Введение

Conga представляет собой комплект программных компонентов для централизованной конфигурации и управления кластерами и хранилищами. Conga состоит из следующих компонентов:

– **luci**

– **ricci**

luci представляет собой сервер, который запускается на управляющем хосте и с помощью ricci связывается с различными хостами (nodes). Агент ricci выполняется на каждом хосте (отдельном или входящем в состав кластера) под управлением Conga. Доступ к luci осуществляется через web-интерфейс.

Для того, чтобы воспользоваться этими утилитами, нужно прежде всего убедиться, что хосты умеют разрешать указанные в конфигурационном файле `clusternode name` (в нашем случае это **rosa2.int** и **rosa.int**). В любом случае, если разрешение имен не происходит, можно добавить в файл `/etc/hosts` на обоих хостах необходимую информацию, например:

```
192.168.68.129 rosa2.int
192.168.68.131 rosa.int
```

Примечание: по умолчанию при установке RELS (используется установщик **anaconda**) имена узлов в `/etc/hosts` ассоциируется с `localhost`. Эту конфигурацию необходимо изменить (как в примере).

3.1.1. Установка

Установите luci на управляющий хост и запустите ее:

```
yum install luci
service luci start
```

Будет выдано следующее сообщение:

```
Generating a 2048 bit RSA private key
writing new private key to '/var/lib/luci/certs/host.pem'
Запускается saslauthd:                                     [  OK  ]
Start luci...                                              [  OK  ]
Point your web browser to https://rosa2.int:8084 (or equivalent) to
access luci
```

Получить доступ к интерфейсу luci можно по адресу https://rosa2.int:8084 . При первом открытии Luci нужно будет добавить в браузере исключение безопасности (как это делается при первом запуске RDS см. Руководство пользователя RELS).

Установите ricci на каждый хост кластера и перезапустите демон ricci:

```
yum install ricci
chkconfig ricci on
service ricci restart
```

Будет показано:

```
Shutting down ricci:
Запускается oddjobd:                                       [  OK  ]
generating SSL certificates...  done
Generating NSS database...  done
Запускается ricci:                                         [  OK  ]
```

Теперь можно работать с веб-интерфейсом управления кластером.

Примечание: иногда при соединении с хостом кластера через интерфейс luci возникает ошибка **authentication to ricci agent failed**. Данная проблема решается разрешением доступа к нужному порту в файерволе, а также установкой пароля ricci командой:

```
passwd ricci
```

на каждом хосте. Иногда также в этой ситуации стоит отключить selinux, если вы не уверены, что абсолютно правильно пользуетесь им.

4 Создание Fence device

4.1 Введение

Fencing — это процесс устранения из кластера неработоспособного хоста. В целом смысл данного процесса заключается в том, при обнаружении некорректной работы одного из хостов, он просто отключается от сети (имеется в виду питание).

Основная задача Fencing — это обеспечение целостности данных. Дело в том, что некорректно работающих хост может внести непрогнозируемые изменения в используемые кластером данные. Для предотвращения данного явления используются специальные устройства, способные обесточить «поврежденный хост». Как правило, это промышленные блоки питания.

Алгоритм при этом примерно следующий: fence-демон опрашивает по расписанию fence-агенты на каждом хосте, для которых задан fence device. В случае обнаружения проблем, все операции ввода/вывода в кластере приостанавливаются. В это же время начинает осуществляться изъятие проблемного хоста из кластера. Когда fence-агент сообщит об его выключении, работа кластера возобновляется. Восстановленный хост позже может вновь вернуться в кластер.

Примечание: для обеспечения наибольшей безопасности данных рекомендуем к каждому хосту подключить fence устройство.

4.2 Настройка

Откройте на управляющей машине интерфейс **luci** (см. Conga раздел Установка). Перейдите во вкладку `mycluster`, где mycluster — название созданного кластера (как создать кластерную конфигурацию и запустить кластер см. Создание базовой кластерной конфигурации).

Примечание: если mycluster в интерфейсе luci не отображается, нажмите во вкладке **Manage cluster** на кнопку *add*, введите fqdn или ip-адрес любого хоста (node) и нажмите на кнопку *connect*.

У вас откроется форма с перечнем хостов (nodes) кластера (luci).

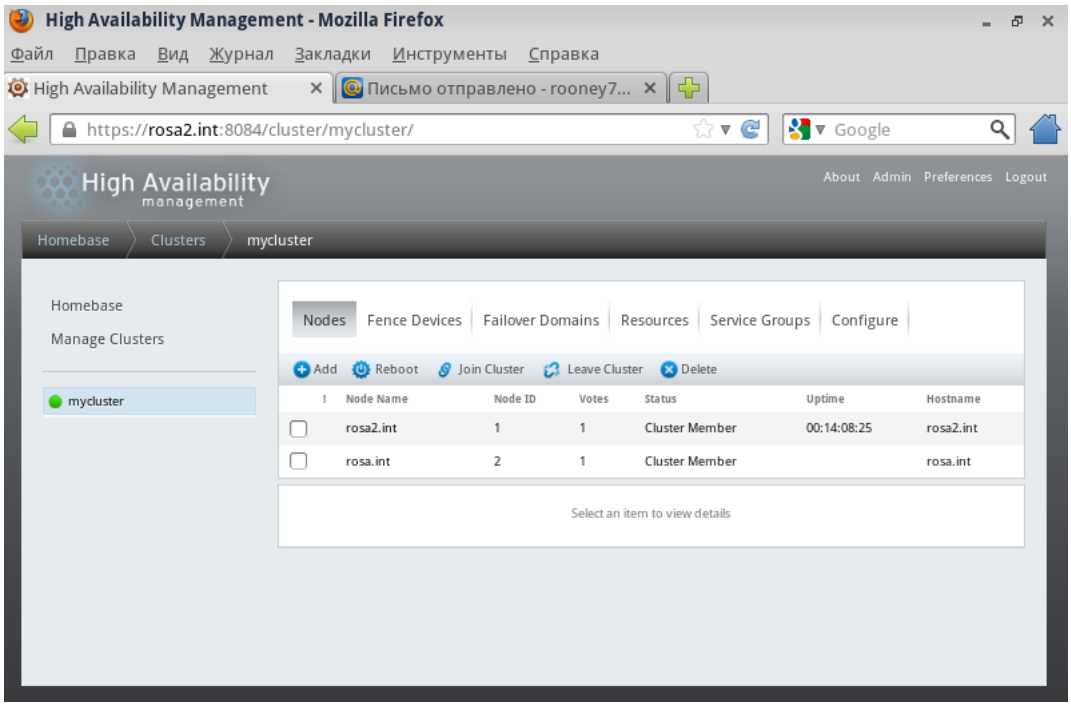

Рисунок 1 — Mycluster luci

Для создания fence device выполните следующие действия:

1) Перейдите во вкладку **Fence Devices** (см.рис. 2).

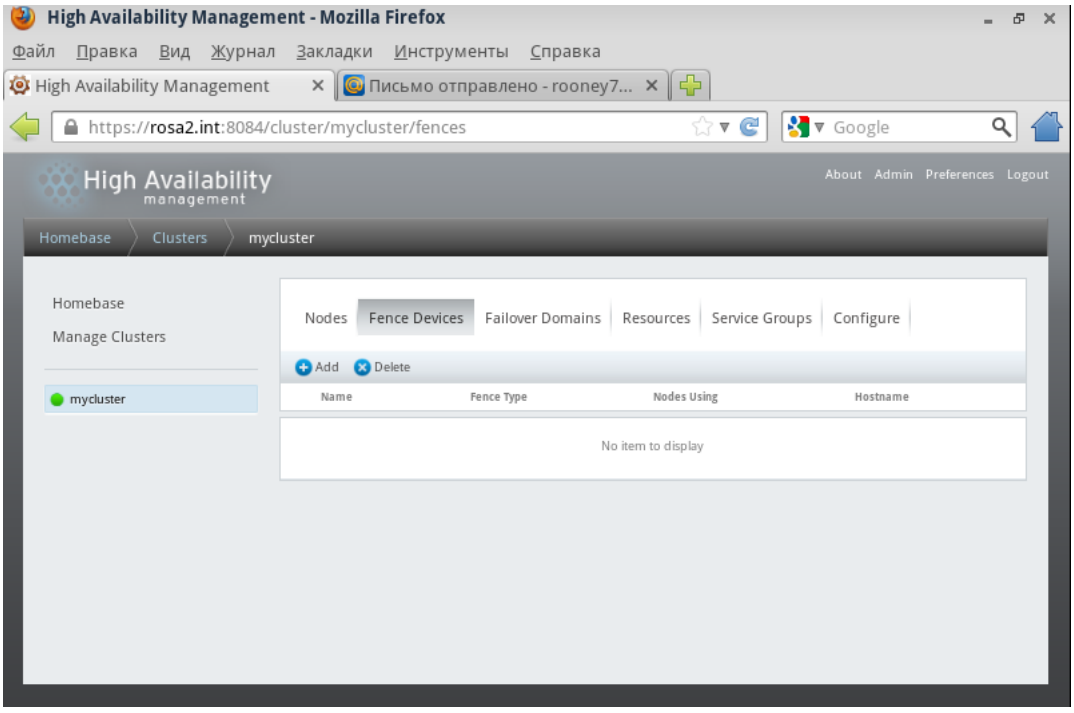

Рисунок 2 — Создание fence device 1

2) Нажмите на кнопку «add», в открывшейся форме (рис. 2) выберите тип Fence Device (в нашем примере **APC Power Switch**), заполните минимально необходимые поля (можно заполнить и остальные), нажмите на кнопку *submit*:

– Name — имя fence-устройства

– Ip Address — ip-адрес fence-устройства
– Login — имя пользователя
– Password — пароль

Необязательные поля:

– IP port — tcp-порт для подключения устройства
– Password Script — скрипт, который передает пароль для подключения устройства
– Power Wait — количество секунд ожидания исполнения выключения

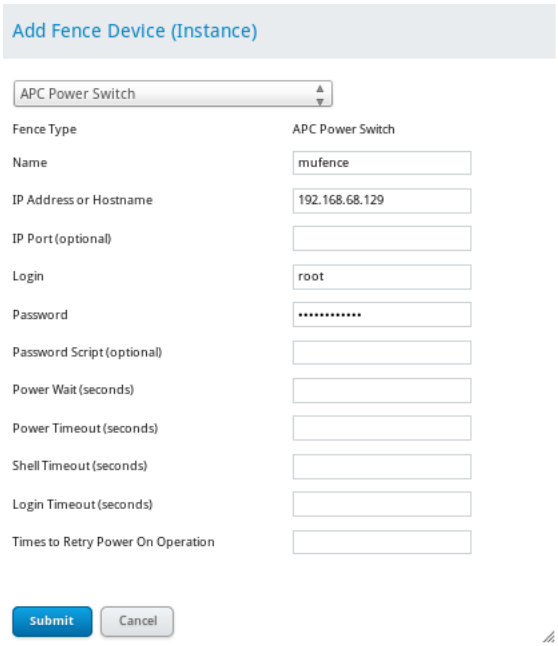

Рисунок 3 — Создание fence device 2

3) Убедитесь, что устройство создано, оно должно появиться в открывшемся окне со списком Fence Device (рис. 3), нужно нажать кнопку *Apply*.

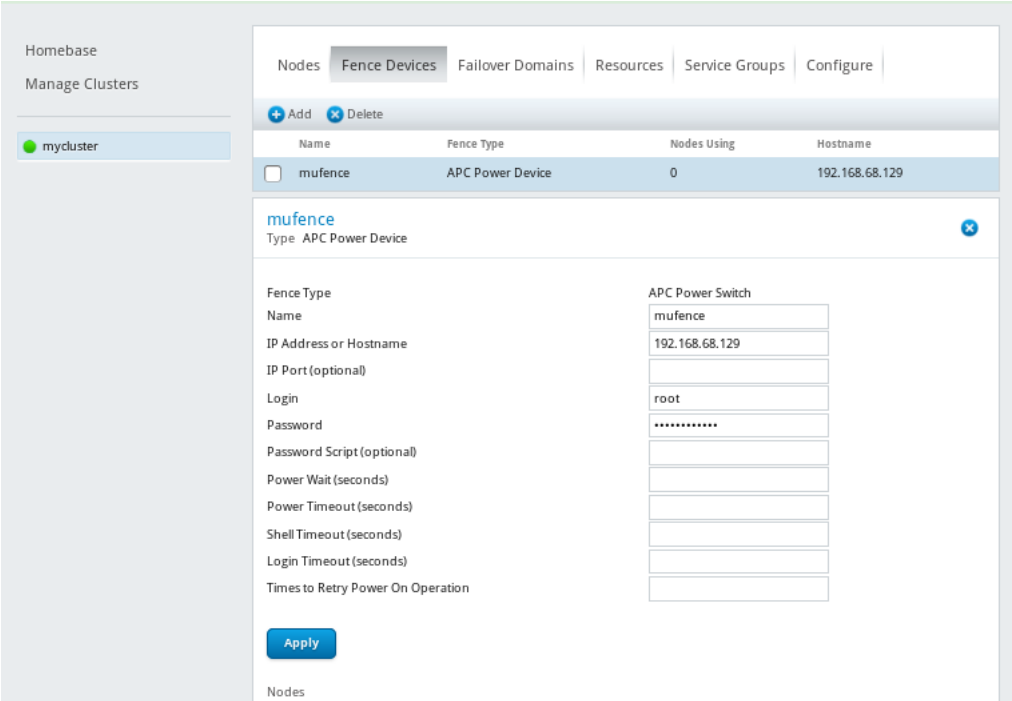

Рисунок 4 — Создание fence device 3

Примечание 1: просмотреть результат добавления fence device можно в конфигурационном файле на хосте, где добавлен fence device: `/etc/cluster/cluster.conf`

Конфигурационный файл будет выглядеть следующим образом (изменения при добавлении fence device выделены жирным, пароль не указан):

```
<?xml version="1.0"?>
<cluster config_version="5" name="mycluster">
        <cman expected_votes="1" two_node="1"/>
        <clusternodes>
                <clusternode name="rosa2.int" nodeid="1"/>
                <clusternode name="rosa.int" nodeid="2"/>
        </clusternodes>
        <fencedevices>
                <fencedevice agent="fence_apc" ipaddr="192.168.68.129"
login="root" name="mufence" passwd="XXXXXXXXXXX"/>
        </fencedevices>
</cluster>
```

Примечание 2: создать fence device можно без использования luci, консольной командой:

```
ccs -h rosa2.int --addfencedev myfence agent=fence_apc ipadrr
=192.168.68.129 login=root passwd=XXXXXXXXXX
```

либо просто внеся аналогичные изменения в конфигурационный файл. Помните: при использовании консольных команд или внесении изменений в конфигурационный файл на-

прямую, необходимо валидировать изменения (`ccs_config_validate`) и перезапустить cman на хостах (`service cman restart`).

4.3 Добавление fence device к хосту

1) Откройте вкладку **Nodes**, перейдите к нужному хосту, нажмите на **Add Fence Method** (рис. 5)

Рисунок 5 — Добавление fence device к хосту шаг 1

2) Введите название метода, нажмите **Submit**.

3) Нажмите Add Fence Instance (рис. 6)

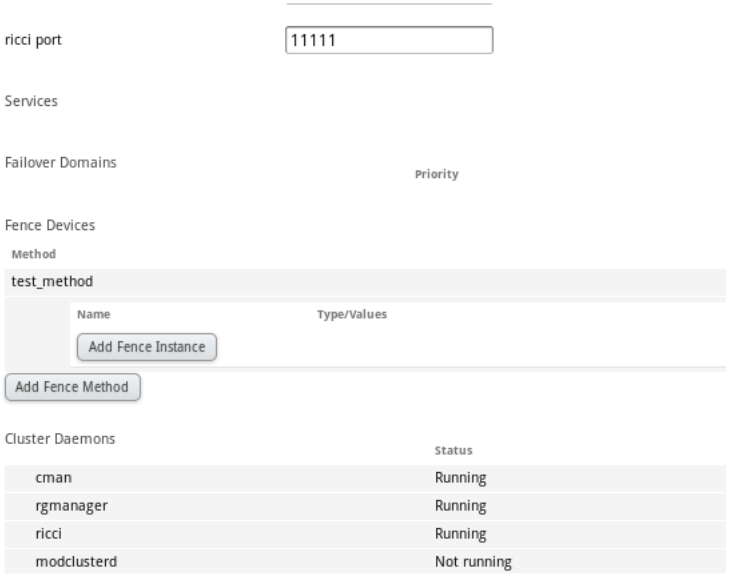

Рисунок 6 — Добавление fence device к хосту шаг 2

4) Выберите в списке созданный fence device, нажмите **Submit**.

Fence устройство (device) будет добавлено к хосту. Во вкладке Fence Devices можно увидеть, что Fence device используется на одном хосте (рис. 7)

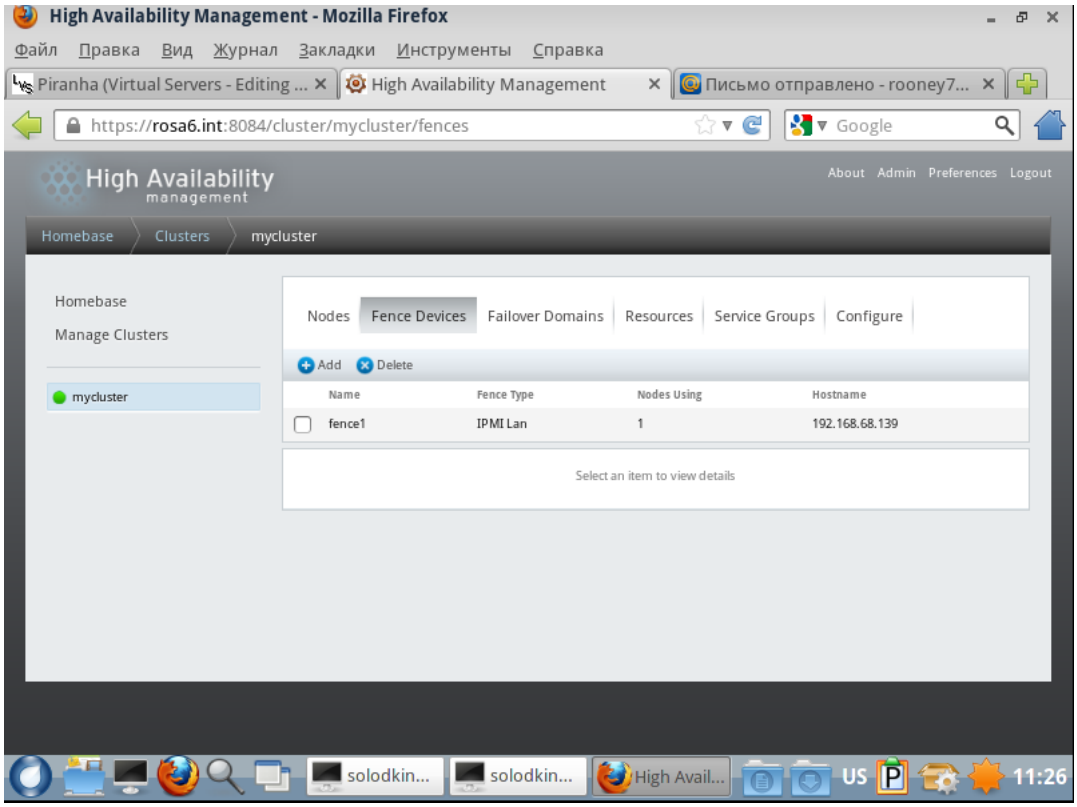

Рисунок 7 — Fence device добавлен к хосту

Примечание: проверить состояние fence-демона можно командой:

```
fence_tool ls
```

Если со всеми хостами все хорошо, то будет показано примерно следующее сообщение:

```
fence domain
member count
victim count  0
victim now    0
master nodeid 2
wait state    none
members       1 2
```

Если с одним из хостов случилась проблема, то, после регистрации события по расписанию, fence-демоном будет показано примерно следующее:

```
fence domain
member count  1
victim count  1
victim now    2
master nodeid 1
wait state    fencing
members       1 2
```

“`wait state fencing`” — означает, что начинается исключение хоста из кластера. Операции ввода/вывода в кластере заблокированы, дожидайтесь, пока хост будет исключен и кластер восстановит работоспособность.

5 GFS

GFS (Global File System) — распределенная файловая система, обеспечивающая прямой и одновременный доступ всех хостов кластера к блочным дисковым устройствам. В RELS реализована поддержка GFS2.

5.1 Предварительные условия

Для успешного запуска GFS2 необходимо запустить кластерные службы, в том числе `rgmanager` и `clvmd` (естественно, что соответствующие пакеты должны быть установлены). Из репозитория РОСЫ нужно установить кластерные службы lvm (в простейшем случае) можно командой

```
yum install *lvm*
```

Мы рекомендуем установить необходимые компоненты в процессе установки операционной системы (см. Руководство пользователя RELS), выбрав группы пакетов **Поддержка масштабируемых файловых систем**, **Надежное хранилище**, **Высокий уровень доступности**.

Примечание: проверить запущенные на каждом хосте сервисы можно нажав на нужный хост в интерфейсе Luci, во вкладке Nodes (см. рис. 8).

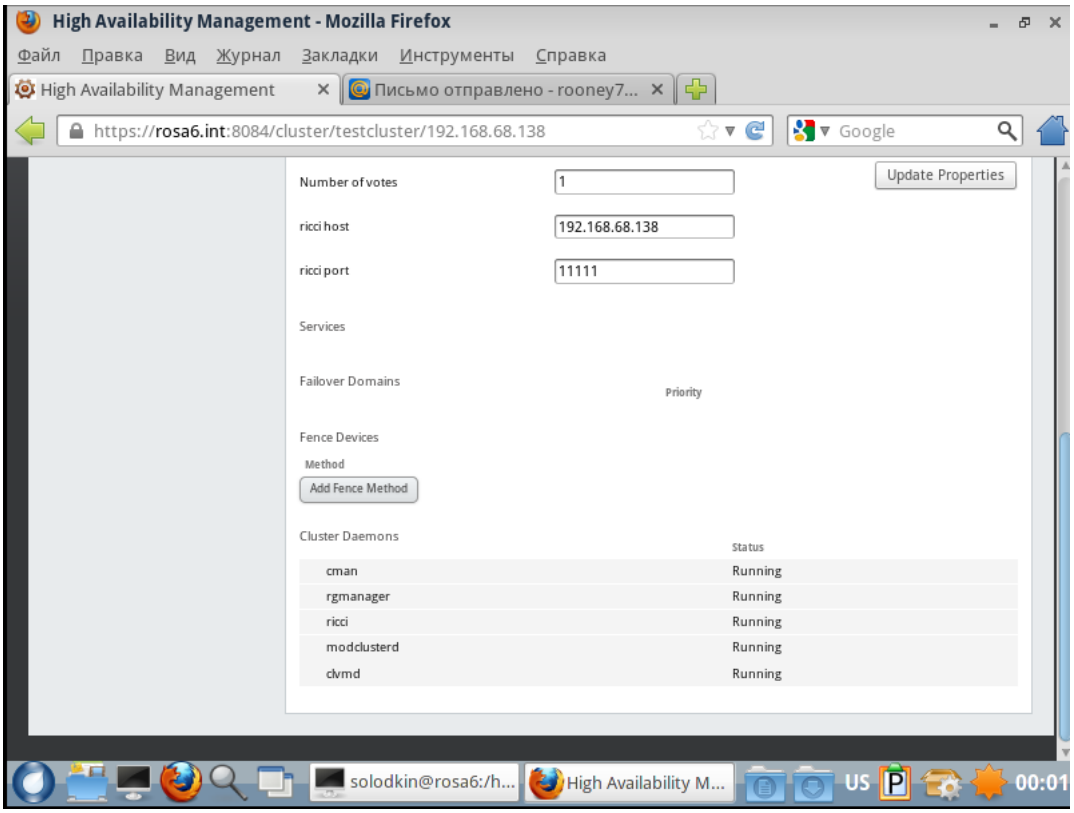

Рисунок 8 — Запущенные сервисы на хосте

Примечание: важно также, чтобы системное время на хостах, использующих GFS, было синхронизировано.

5.2 Создание GFS

1) Создайте для каждой файловой системы свой логический том (см. Руководство пользователя RELS).

2) Создайте файловую систему GFS2 командой:

```
mkfs.gfs2 -p lock_dlm -t название_кластера:название_ФС -j количество_журналов блочное_устройство
```

Важно: название ФС должно быть уникальным в каждом кластере. Для каждого хоста должен быть один журнал, поэтому количество журналов рекомендуется указывать равным количеству хостов в кластере, использующих GFS.

Пример:

```
mkfs.gfs2 -p lock_dlm -t testcluster:Mygfs -j 2 /dev/forgfs/test_for_gfs
```

В результате будут показаны сведения об ФС с uuid, вроде таких:

```
Device:                    /dev/forgfs/test_for_gfs
Blocksize:                 4096
```

```
Device Size                4,86 GB (1272832 blocks)
Filesystem Size:           4,86 GB (1272832 blocks)
Journals:                  2
Resource Groups:           20
Locking Protocol:          "lock_dlm"
Lock Table:                "testcluster:Mygfs"
UUID:                      2aad00cc-cba6-471d-bd1c-8bdda90bca9e
```

3) Примонтируйте ФС на каждом хосте командой:

`mount блочное_устройство точка_монтирования`

или

`mount -o acl блочное_устройство точка_монтирования`

Опция "`-o acl`" позволяет устанавливать ACL-списки для пользователей.

Пример: `mount /dev/forgfs/test_for_gfs /mnt/cluster/storage`

Примечание: можно добавить запись в `fstab` или скрипт в `init.d` для автоматического монтирования ФС при запуске хоста

Мы крайне не рекомендуем пытаться использовать GFS2 на хостах, к которым не подключены fence устройства. В случае сбоя кластерных сервисов последствия могут быть непредсказуемы.

Также мы крайне рекомендуем перед выключением хоста отмонтировать gfs и остановить кластерные службы.

Примечание: размонтирование производится обычной командой `umount`.

Перед запуском в промышленную эксплуатацию GFS мы рекомендуем проконсультироваться со специалистами технической поддержки РОСЫ, а также провести опытную эксплуатацию инфраструктуры с GFS в течении 2-3-х месяцев, чтобы понаблюдать за устойчивостью работы хранилища.

5.3 Другие операции с GFS

Для расширения ФС воспользуйтесь командой

`gfs_grow точка_монтирования`.

Команда позволяет увеличить объем ФС, причем, когда ФС смонтирована. Как правило, используется после увеличении физического доступного пространства. Для добавления журнала используйте команду:

`gfs2_jadd -j число точка_монтирования`

Число журналов, как правило, равно числу хостов, использующих gfs. Как и `gfs_grow` команда может выполняться при смонтированной ФС, должна быть выполнена на одном хосте и распространяется на остальные.

Примечание: при увеличении числа журналов требуется дополнительное место для их хранения.

6 Добавление кластерных ресурсов и сервисов

Рассмотрим добавление сервиса на примере web-сервера Apache. В первую очередь установите на хостах Apache. Сделать это можно с помощью ROSA Server Setup, выполнив следующие действия:

6.1 Установка Apache на хостах

1) Откройте **ROSA Server Setup**, в группе **Службы и инструменты сервера** нажмите **Ввод**. В открывшемся окне выберите **Web server** (см. рис. 9).

Рисунок 9 — Установка Apache 1

2) Подтвердите установку (рис. 10), дождитесь завершения установки.

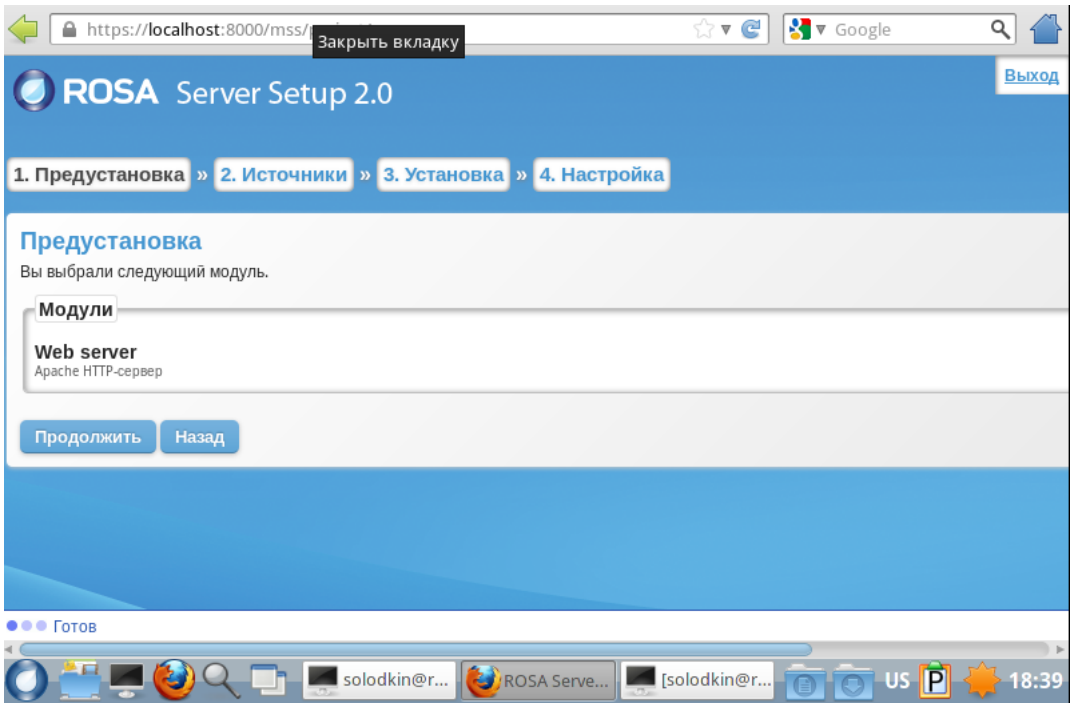

Рисунок 10 — Установка Apache 2

Примечание: необходимо обеспечить идентичность конфигурационных файлов Apache на всех хостах (nodes) (`/etc/httpd/conf/httpd.conf`).

6.2 Создание Failover domain

Теперь необходимо создать **Failover domain**. Failover domain — это множество узлов кластера, на которых будет восстановлена работа сервиса (в нашем случае Apache) в случае сбоя.

Выполните следующие действия:

1) В интерфейсе luci перейдите ко вкладке Failover Domain (рис. 11), нажмите на кнопку *Add*.

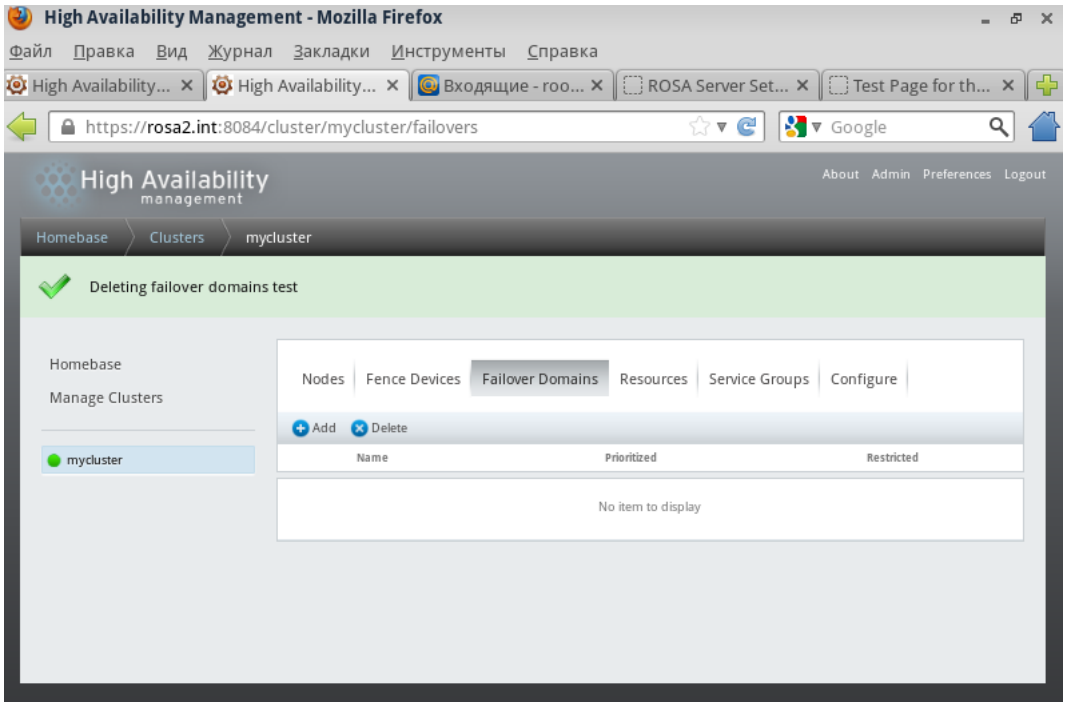

Рисунок 11 — Failover domain

2) В открывшейся форме, введите имя failover domain (рис. 12), отметьте галочками хосты (nodes), которые должны войти в failover domain, отметьте галочками типы failover domain (если галочка стоит, то тип активен, если нет, то тип не активен), нажмите **submit**:

– **Prioritized** — упорядочивает хосты (nodes) в порядке приоритета, по которому будет происходить восстановление сервиса.

– **Restricted** — сервис будет восстанавливаться только на тех вершинах, которые специально указаны.

– **No Failback** — не возвращать после сбоя сервис на хост (node), первого приоритета, когда он вновь станет доступен.

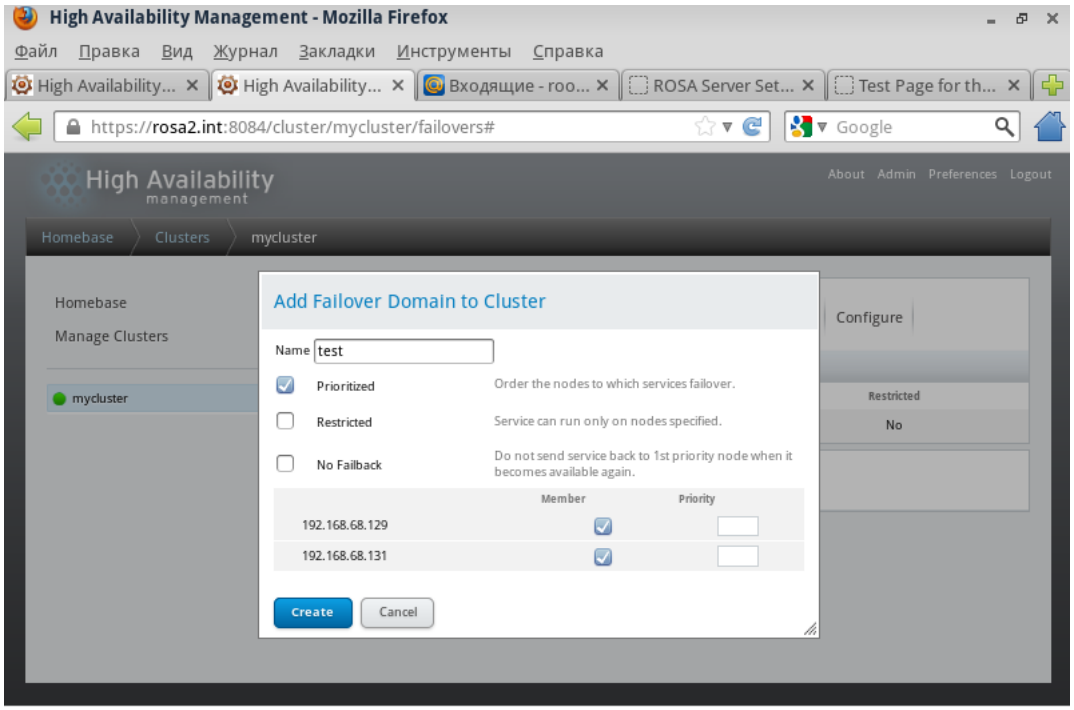

Рисунок 12 — Создать failover domain

3) Убедитесь, что failover domain создан (рис. 13).

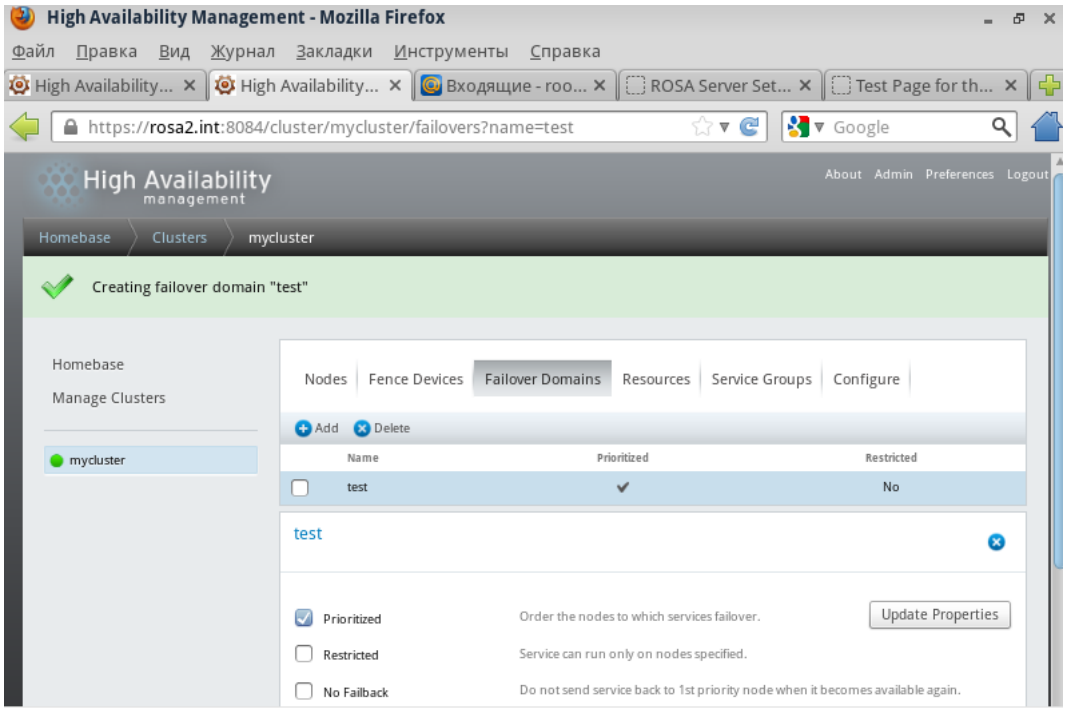

Рисунок 13 — Failover domain создан

6.3 Добавление в кластер ресурсов

Теперь необходимо добавить в кластер ресурсы, которые будут использоваться сервисом Apache. Вам необходимо добавить следующие ресурсы: Script, Ipaddress, FileSystem (см.рис. 14), сделать это можно в интерфейсе luci во вкладке Resources.

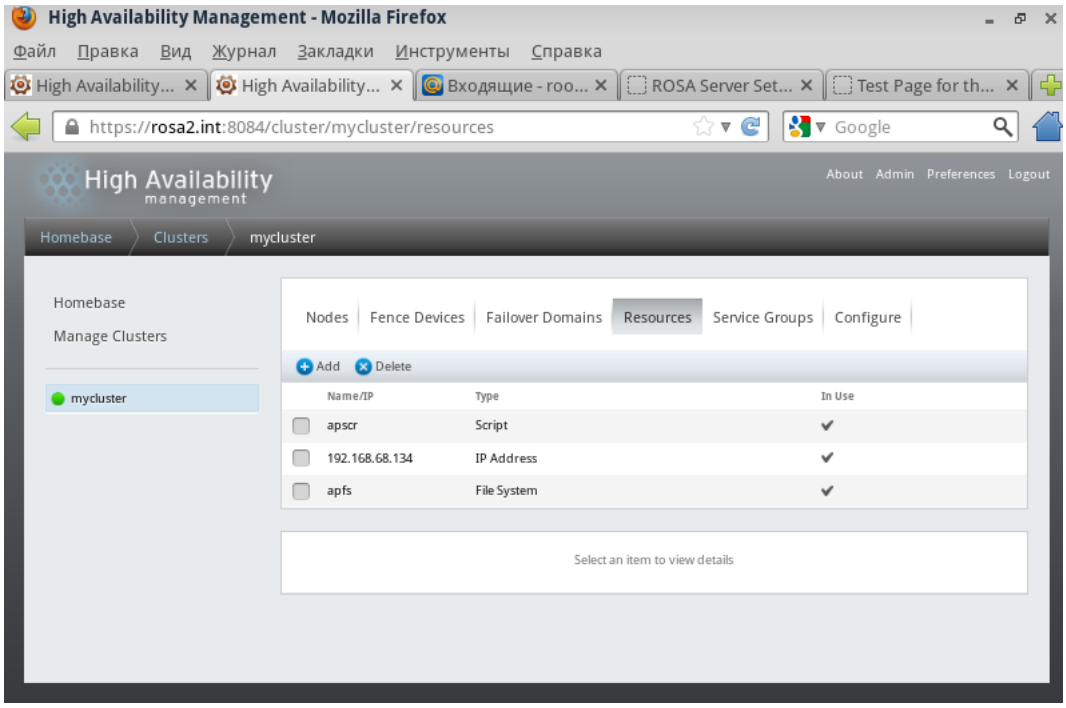

Рисунок 14 — Ресурсы

Рассмотрим, как добавить каждый из них:

1) Для добавления скрипта, нажмите во вкладке **Resources** кнопку **Add**, в поле **Select a resource Type** выберите **Script**, в поле **Name** введите название ресурса (например, `apscr`), в поле **Full Path to Script File** введите путь к скрипту инициализации Apache (если путь к скрипту в файле `/etc/httpd/conf/httpd.conf` не менялся, то необходимо указать путь по умолчанию `/etc/rc.d/init.d/httpd`), нажмите на кнопку **Submit** (рис. 15).

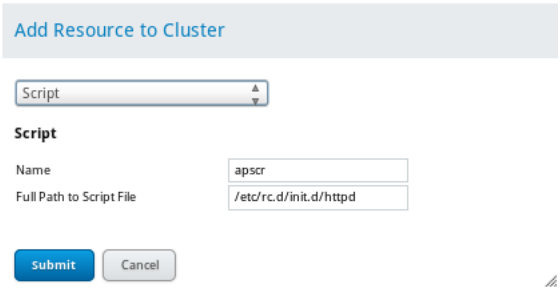

Рисунок 15 — Добавление скрипта

2) Для добавления IP-адреса нажмите во вкладке **Resources** кнопку *Add*, в поле **Select a resource Type** выберите **IP Address**, в поле **Name** введите IP-адрес (например, `192.168.68.134`), убедитесь, что стоит галочка напротив **Monitor Link**, можно заполнить и остальные поля. Далее нажмите на кнопку *Submit* (см. рис. 16).

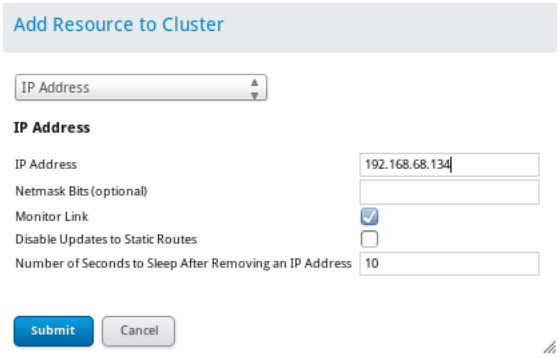

Рисунок 16 — Добавить ip-адрес

3) Для добавления Файловой системы (Filesystem) нажмите во вкладке **Resources** кнопку **Add**, в поле **Select a resource Type** выберите **Filesystem**, в поле **Name** введите название ресурса (например, `apfs`), в поле **Filesystem Type** выберите `ext3`, в поле **Mountpoint** укажите точку монтирования (по умолчанию `/var/www/html/` или то значение, которое прописано в конфигурационном файле Apache `/etc/httpd/conf/httpd.conf` как каталог для html-документов), в поле **Device** укажите имя устройства (например, `/dev/sda3`), можно заполнить и остальные поля, нажмите на кнопку **Submit** (см. рис. 17).

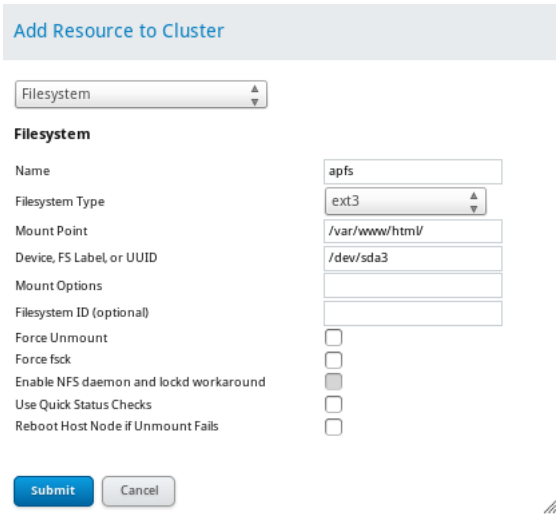

Рисунок 17 — Добавление файловой системы

Примечание: просмотреть и отредактировать разметку дисков можно с помощью утилиты `cfdisk`, правда это небезопасно с точки зрения сохранности данных.

6.4 Добавление сервиса Apache

Теперь можно добавить сервис Apache, для этого:

1) Перейдите в интерфейсе luci во вкладку **Service Groups**, нажмите на кнопку **Add**, в открывшейся форме введите (см. рис. 18) в поле **Service Name** имя сервиса (например, `aptest`), в поле **Failover Domain** выберите созданный Failover Domain (например, `test`).

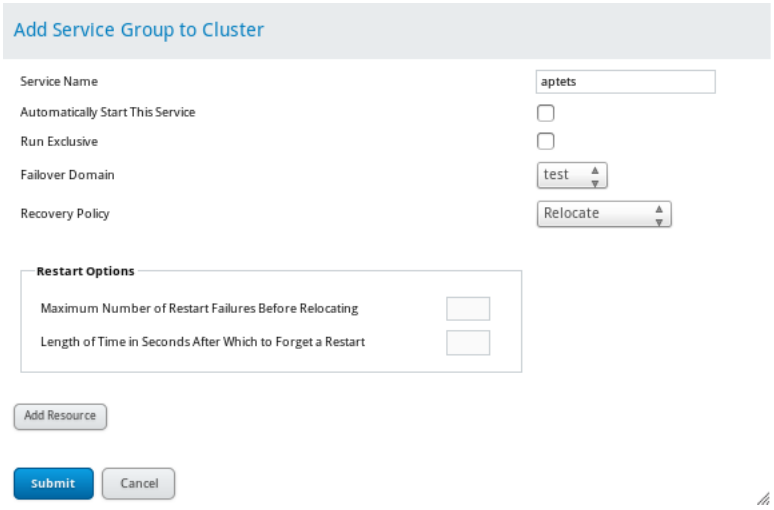

Рисунок 18 — Добавление Apache 1

2) Добавьте ВСЕ созданные ресурсы для сервиса Apache, для добавления каждого ресурса необходимо нажать на кнопку **Add Resource** и выбрать нужный ресурс из списка (см. рис. 19). После этого нажмите на кнопку **Submit**.

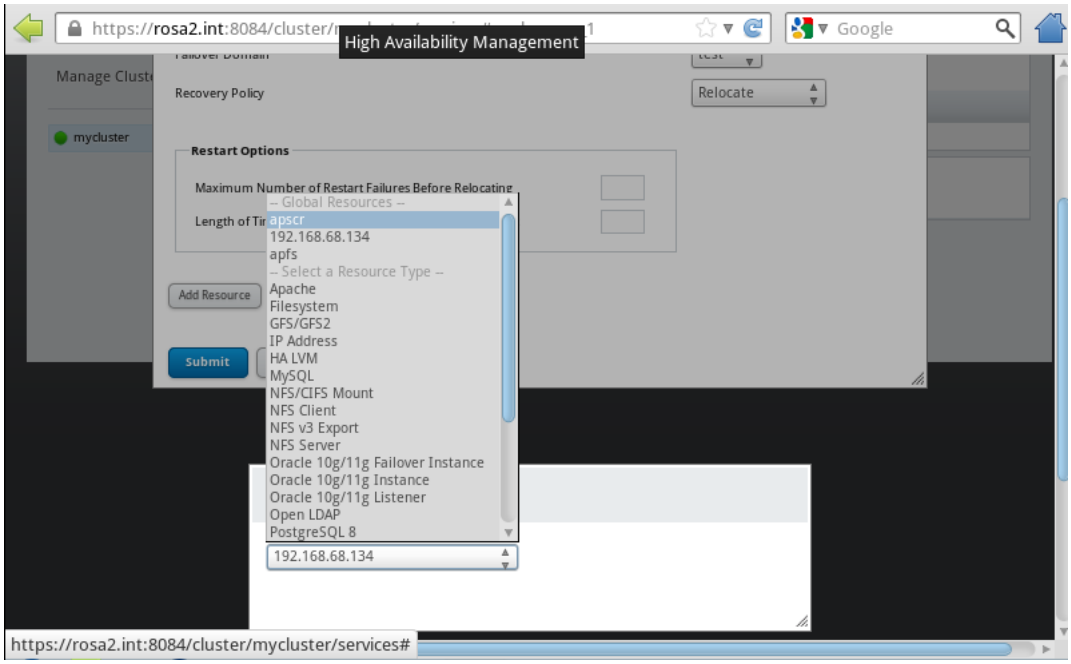

Рисунок 19 — Добавить ресурс к сервису

3) Убедитесь, что сервис добавлен (см. рис. 20), отметьте его галочкой, нажмите на кнопку **Start**.

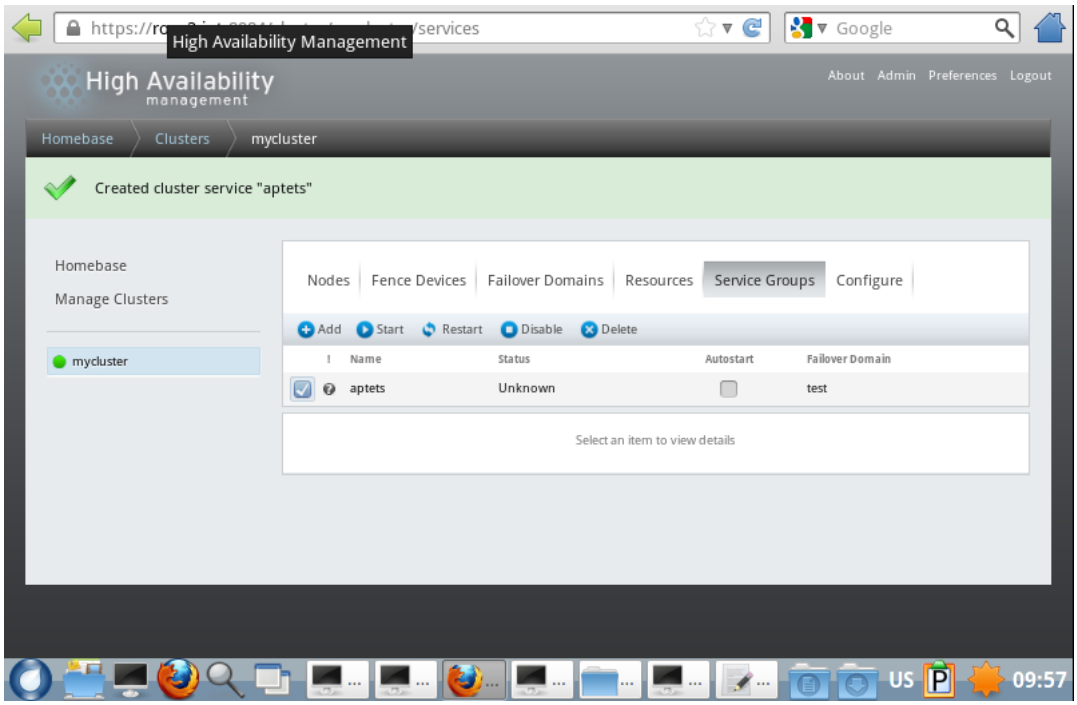

Рисунок 20 — Сервис добавлен

4) Выберите добавленный сервис в списке, нажмите кнопку *Start*, убедитесь, что сервис запустился на одном из хостов (nodes), см. рис. 21

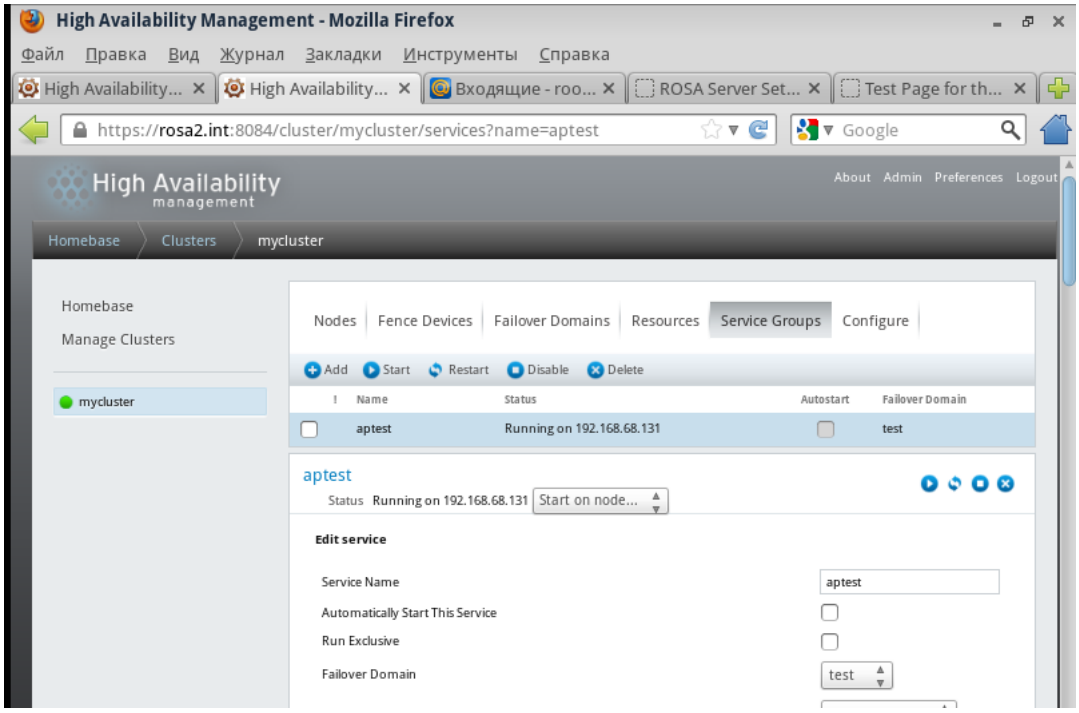

Рисунок 21 — Запущенный сервис

7 Распределение нагрузки

7.1 Введение

Балансировка нагрузки — важная задача администрирования кластерных решений. Такую задачу необходимо решать для высоконагруженных решений. Естественно, что первейшим шагом для реализации нагруженных решений является наращивание физических мощностей, т.е. установка дополнительных аппаратных серверов (или, может быть, виртуальных машин). Однако такой подход не является полным решением проблемы, поскольку запросы из внешней по отношению к серверам сети не распределяются и все запросы могут приходить, скажем, на первый сервер. Для решения этой проблемы в Linux был реализован Linux Virtual Server (LVS), основой которого является модуль ядра (`ipvsЪ`). По сути данное решение — это роутер, который перенаправляет запросы из внешней сети и распределяет их между серверами в локальной сети. Мы рекомендуем обязательно использовать конфигурацию, в которой используется ДВА маршрутизатора. Дело в том, что в конфигурации с одним маршрутизатором, в случае его выхода из строя, откажут все сервисы, поднятые в кластере. Таким образом, в простейшем случае конфигурация балансировки нагрузки состоит из двух слоев (см. рис. 22). Первый слой содержит два маршрутизатора: активный и резервный. Второй слой содержит реальные сервера кластера.

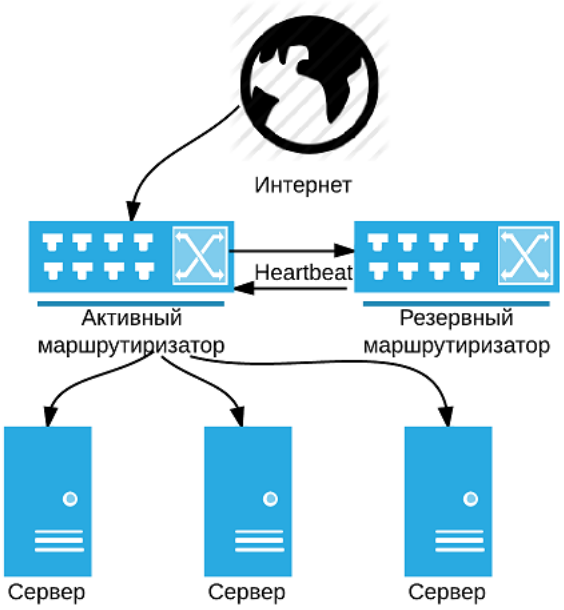

Рисунок 22 — Простейшая конфигурация распределения нагрузки

Активный маршрутизатор распределяет нагрузку между серверами в локальной сети и контролирует доступность сервисов. В случае его сбоя, включается резервный маршрути-

затор и берет на себя функции отказавшего маршрутизатора.

Запросы пользователей из внешней сети приходят на виртуальные IP-адреса (**Virtual IP** или **VIP**), которые ассоциированы с доменным именем (например, example.com). Эти виртуальные адреса ассоциированы с интерфейсами, подключенным к Интернету, работающего маршрутизатора. Соответственно, маршрутизатор перенаправляет поступающие (на VIP) запросы на реальные сервера и перенаправляет ответы обратно.

Примечание: виртуальным сервером является сервис, например, веб-сервер, к которому будут обращаться по HTTP.

Примечание: с каждым сервисом может быть ассоциирован отдельный интерфейс, например, для http — eth0:1, для ftp — eth0:2.

Существуют различные алгоритмы перенаправления и распределения запросов:

1) Round robin
2) Round Robin с весовыми коэффициентами
3) Минимум подключений
4) Минимум подключений с весовыми коэффициентами
5) Минимум подключений с учетом местоположения
6) Минимум подключений с учетом местоположения и репликацией
7) Хеш получателя
8) Хеш отправителя

Для управления распределением нагрузки в RELS используется специальная утилита с графическим интерфейсом **Piranha**.

7.2 Piranha

Введение в piranha

В целом, Piranha обеспечивает упрощение работы по конфигурированию распределения нагрузки. Не используя Piranha, можно создать нужную конфигурацию с помощью скриптов и работы с конфигурационными файлами. Основные настройки содержатся в файле: `/etc/sysconfig/ha/lvs.cf`.

 Помните, что если править конфигурационный файл вручную, необходимо четко соблюдать все синтаксические правила его оформления. Иначе могут возникнуть серьезные сбои в ПО кластера.

Для работы с piranha используется утилита **piranha-gui**. Для работы с piranha необходимо, чтобы на ЭВМ-маршрутиризаторе был установлен Apache и веб-браузер. Доступ к

интерфейсу piranha осуществляется по адресу: http://localhost:3636.

 Важно: для корректной работы ПО распределения нагрузки должны быть запущены следующие демоны на маршрутиризаторах: `piranha-gui`, `pulse`, `sshd`. Рекомендуется включить также Iptables. Однако, не следует запускать pulse до окончания создания конфигурации в piranha. Рекомендуется включить эти сервисы в автозагрузку на уровнях 3 и 5 командой:

```
/sbin/chkconfig –level 35 <название_демона> on
```

Первые шаги в piranha

Перед первым использованием piranha на активном маршрутиризаторе мы очень рекомендуем установить для нее пароль командой:

```
/usr/sbin/piranha-passwd
```

Для запуска Piranha используйте команду:

```
/sbin/service/piranha-gui start
```

Для того, чтобы изменить порт, который используется для доступа к Piranha (по умолчанию это 3636), отредактируйте в файле

```
/etc/sysconfig/ha/conf/httpd.conf
```

строку `Listen 3636`.

Кроме использования пароля, мы рекомендуем ограничить доступ пользователей к Piranha tools, отредактировав файл `/etc/sysconfig/ha/web/secure/.htaccess` , заменив строки:

```
Order deny,allow
Allow from all
```

на

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
```

Разрешив, таким образом, доступ к утилите только с localhost.

Также необходимо включить **ipforwarding** в ядре маршрутизатора, без этого маршрутизатор не сможет перенаправлять пакеты реальным серверам. Сделать это можно, отредактировав в файле `/etc/sysctl.conf` строку:

```
net.ipv4.ip_forward = 0
```

на

```
net.ipv4.ip_forward = 1
```

Запуск piranha

При первом запуске piranha необходимо будет ввести имя пользователя (username) и пароль (password). В качестве username введите piranha, в качестве пароля, введите тот, что задан командой: `/usr/sbin/piranha-passwd`.

В интерфейсе piranha представлены четыре основные вкладки: **control/monitoring**, **global settings**, **redundancy**, **virtual servers**.

Piranha control/monitoring

Во вкладке **control/monitoring** (см.рис. 23) представлены следующие элементы:

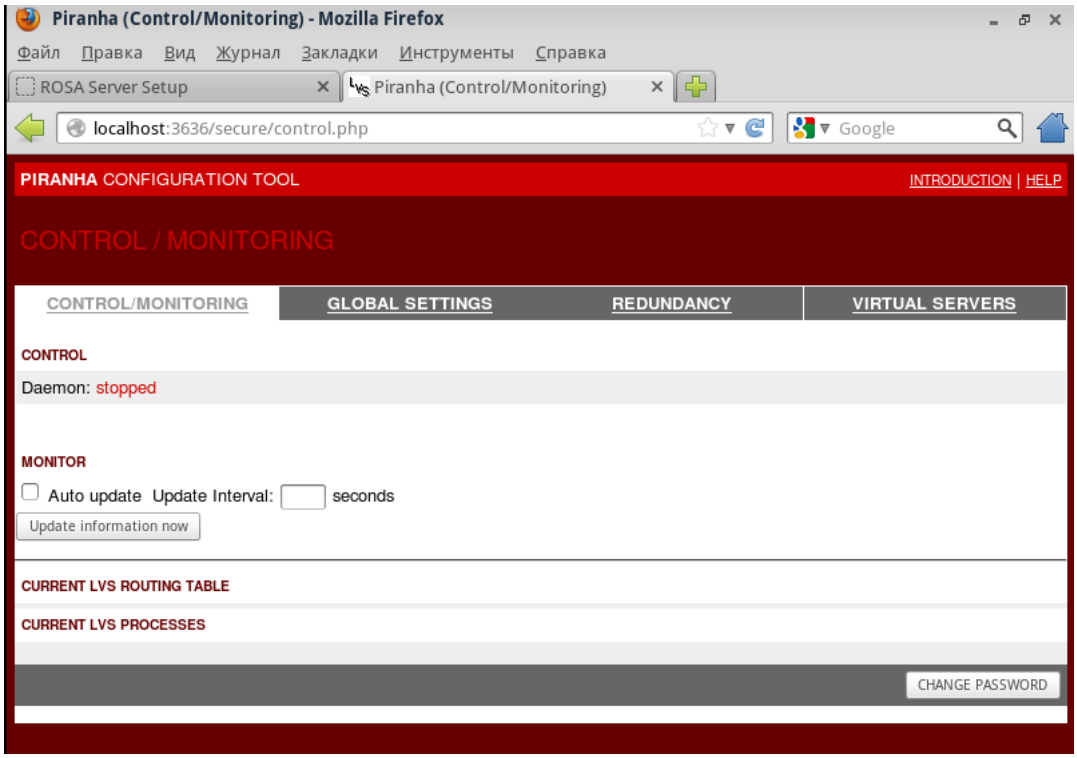

Рисунок 23 — Piranha control

Piranha Global Settings

Во вкладке **Global Settings** можно просмотреть и изменить настройки сетевых интерфейсов активного маршрутизатора (см.рис. 24).

Primary server public IP — нужно ввести реальный IP-адрес активного маршрутизатора, используемый при подключении к Интернету.

Primary server private IP — нужно ввести реальный IP-адрес альтернативного сетевого интерфейса активного маршрутизатора. Он будет использоваться для создания heartbeat-канала с резервным маршрутизатором. Поле необязательное для заполнения, но если оставить его пустым в случае сбоя активного маршрутизатора, резервный сразу не включится.

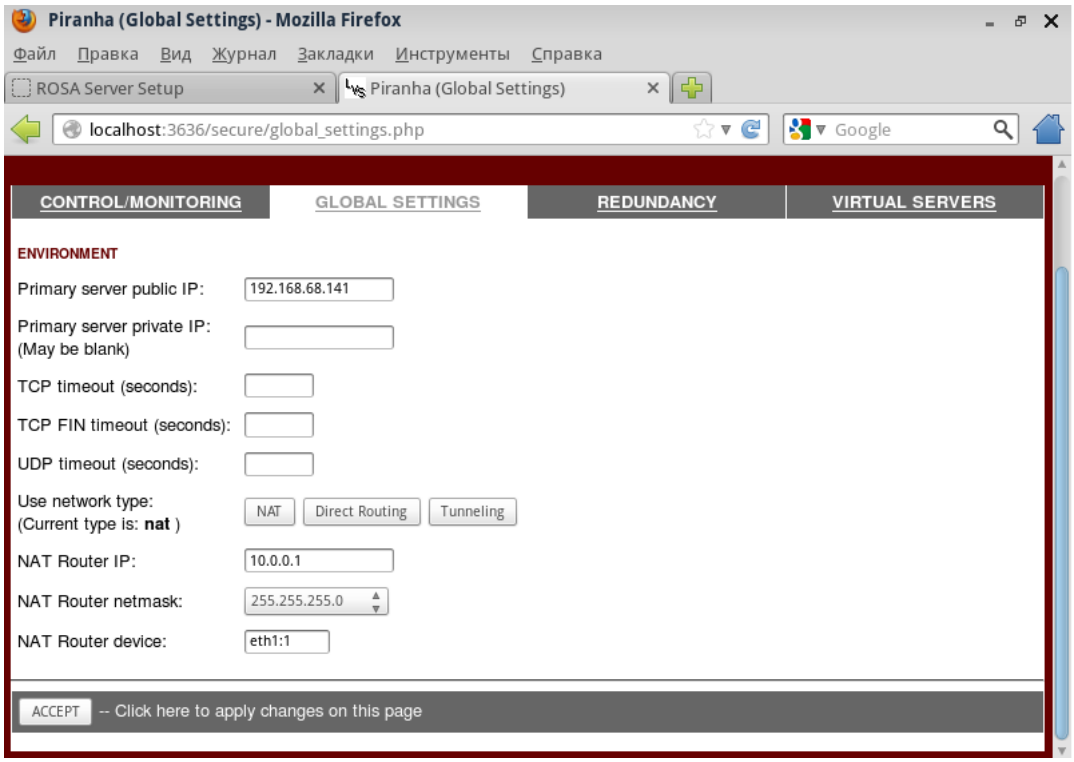

Рисунок 24 — Piranha settings

– **TCP timeout** — время до перехода в режим ожидания tcp-сессии

– **TCP fin timeout** — время до перехода в режим ожидания tcp-сессии после отправке пакета FIN

– **UDP timeout** — время до перехода в режим ожидания udp-сессии

– **Use network type** — выберите тип маршрутиризации, доступно: NAT, Direct routing, tunneling.

Для типа NAT необходимо заполнить еще следующие поля:

– **NAT router ip** — указать плавающий ip интерфейса маршрутиризатора, подключенного к локальной сети. Используется как адрес шлюза для реальных серверов

– **NAT router netmask** — можно указать маску подсети для маршрутиризатора

– **NAT router device** — укажите наименование устройства, к которому подключен плавающий IP-адрес, указанный выше (следует указывать псевдоним устройства, например, если указать `eth1:1`, для интерфейса **eth1**, подключенного к локальной сети, ip-адрес будет подключен к устройству `eth1:1`).

 Не забудьте нажать на кнопку *ACCEPT* внизу формы, чтобы подтвердить изменение настроек.

Piranha Redundancy

Во вкладке **REDUNDANCY** можно изменить настройки резервного маршрутиризатора

и настроить мониторинг heartbeat.

Для начала изменения настроек нажмите на кнопку **Enable**, после чего появится форма для внесения параметров конфигурации (см.рис. 25).

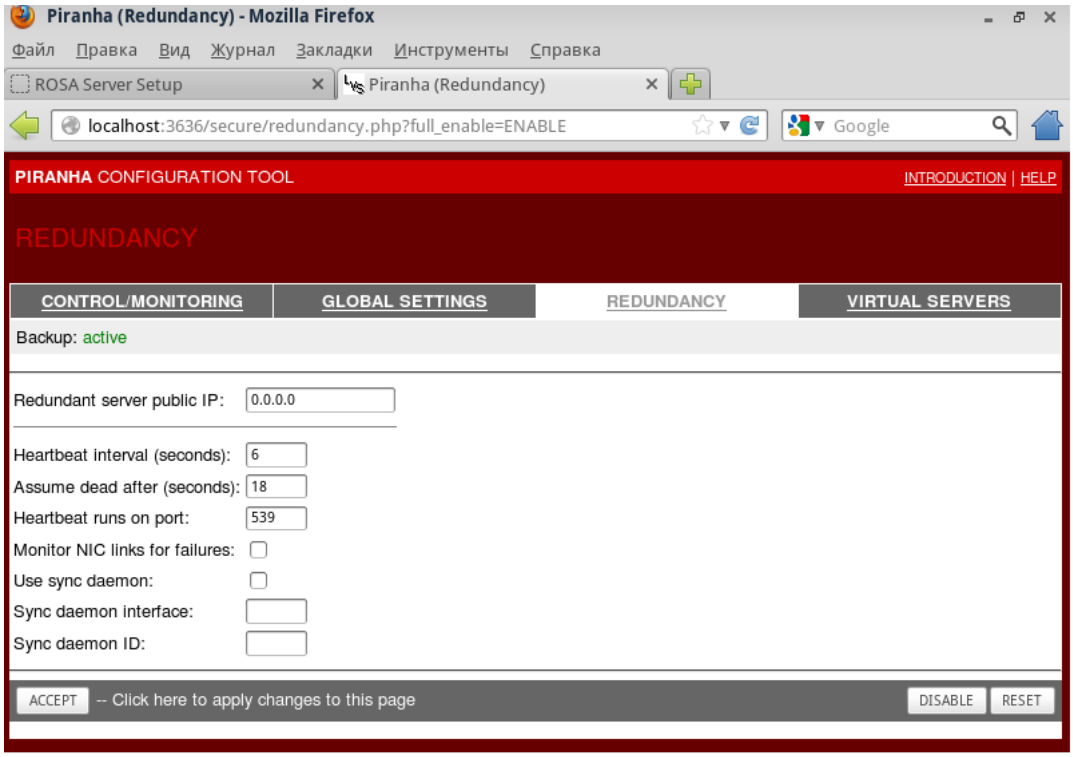

Рисунок 25 — Piranha redundancy

– **Redundant server public IP** — введите IP-адрес сетевого интерфейса, подключенного к Интернет резервного маршрутизатора;

– **Heartbeat Interval (seconds)** — введите количество секунд между контрольными запросам проверки работоспособности основного маршрутизатора;

– **Assume dead after (seconds)** — введите количество секунд, через которое, в случае отсутствия ответа от основного маршрутизатора, начнется резервное восстановление на резервном маршрутизаторе

– **Heartbeat runs on port** — укажите номер порта, на который будут отправляться запросы состояния от резервного маршрутизатора. По умолчанию — 539.

– **Use sunc demon** — поставите галочку, если нужно использовать демон синхронизации.

– **Sync Daemon Interface** — укажите сетевой интерфейс, по которому демон будет отправлять запросы и получать ответы. По умолчанию — eth0.

– **Sync daemon id** — укажите идентификатор сообщений демона.

Не забудьте нажать на кнопку *ACCEPT* внизу формы, чтобы подтвердить из-

менение настроек.

Piranah Virtual Servers

Во вкладке **Virtual Servers** можно просмотреть информацию о виртуальных серверах и изменить их конфигурацию (см. рис. 26).

Для добавления сервера, используйте кнопку **ADD**.

Для редактирования данных о выбранном сервере — кнопку **EDIT**.

Для удаления выбранного сервера — **DELETE**.

Для деактивации/активации выбранного сервера — **(DE)ACTIVE**. Ее нужно использовать для активации сервера сразу после его создания.

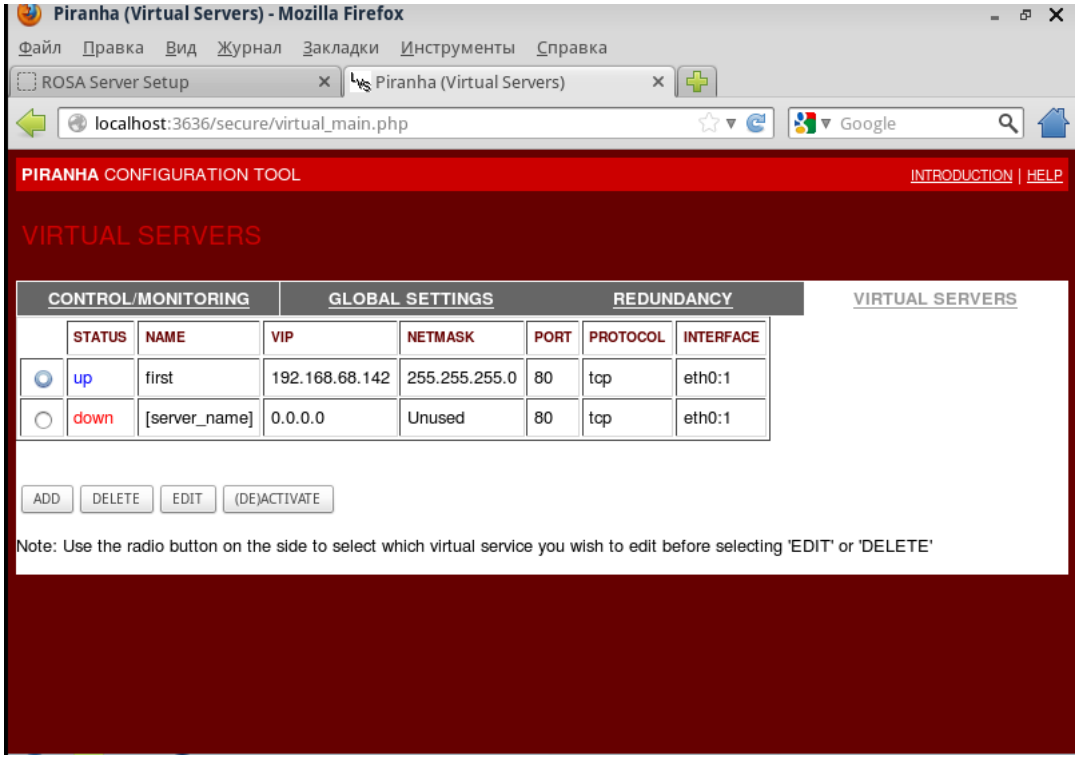

Рисунок 26 — Piranha Virtual Servers

Рассмотрим подробнее процесс создания и настройки виртуального сервера. Нажмите на кнопку *EDIT* (или *ADD*), откроется форма настроек виртуального сервера (см. рис. 27).

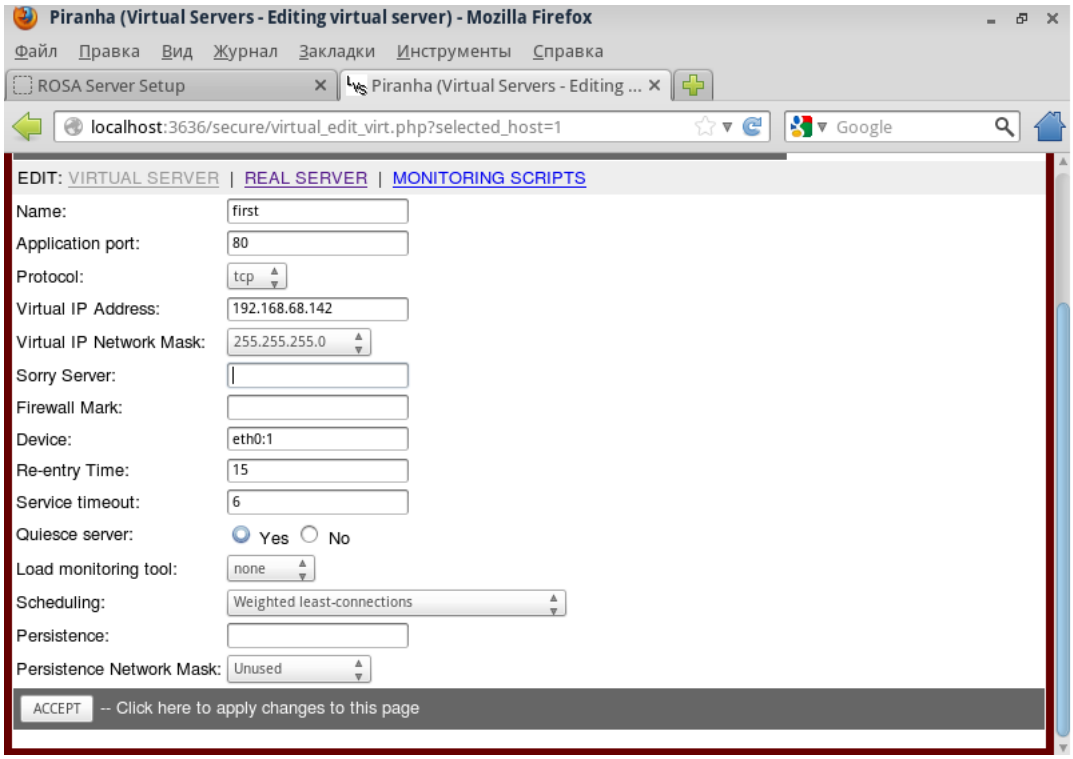

Рисунок 27 — Настройки виртуального сервера

– **Name** — введите имя сервера, часто используется имя используемого протокола, например, http.

– **Application port** — введите номер порта, используемого сервисом, например, для HTTP — порт 80.

– **Protocol** — выберите протокол, доступно UDP или TCP. Для нашего примера — TCP

– **Virtual IP Address** — введите ip-адрес виртуального сервера

– **Virtual IP Network Mask** — выберите маску подсети виртуального сервера.

– **Firewall Mark** — используется, когда доступ к сервису осуществляется по разным портам, например, по HTTP по 80 порту и по HTTPS по 443 порту. Метка обеспечивает доступ пользователей по обоим типам подключений, если тип только один, метку не ставьте.

– **Device** — введите имя устройства, которому соответствует ip-адрес, который введённый выше. Здесь пишется псевдоним устройства, например, `eth0:1`.

– **Re-entry Time** — введите количество секунд, через которое в случая сбоя сервиса, маршрутиризатор будет возвращать его в пул сервисов.

– **Service Timeout** — введите время в секундах, после которого, в случае отсутствия ответов от сервиса, он признается отказавшим и удаляется из пула.

– **Quiesce server** — при включении данной опции при каждом добавлении нового виртуального сервера, таблица конфигурации будет обновляться и распределение запросов будет происходит, таким образом, как если бы все серверы были только что добавлены.

Используйте при большом количестве виртуальных серверов.

– **Load monitoring tool** — выберите тип демона, с помощью которого маршрутизатором будет отслеживаться нагрузка на серверах. Соответствующие демоны должны быть запущены на серверах. Старайтесь не использовать эту опцию, оставьте просто **none**.

– **Scheduling** — выберите алгоритм распределения нагрузки, по умолчанию используется «Минимум подключений с весовыми коэффициентами».

– **Persistence** — используется при активировании **Firewall Mark**, укажите время в секундах сохранения активности подключения.

– **Persistence Network Mask** — укажите подсеть в которой будет осуществляться сохранение активных подключений.

Не забудьте нажать на кнопку *ACCEPT* внизу формы, чтобы подтвердить изменение настроек.

В подвкладке **Real server** конкретного виртуального сервера можно просмотреть и изменить конфигурацию реальных серверов (см. рис. 28).

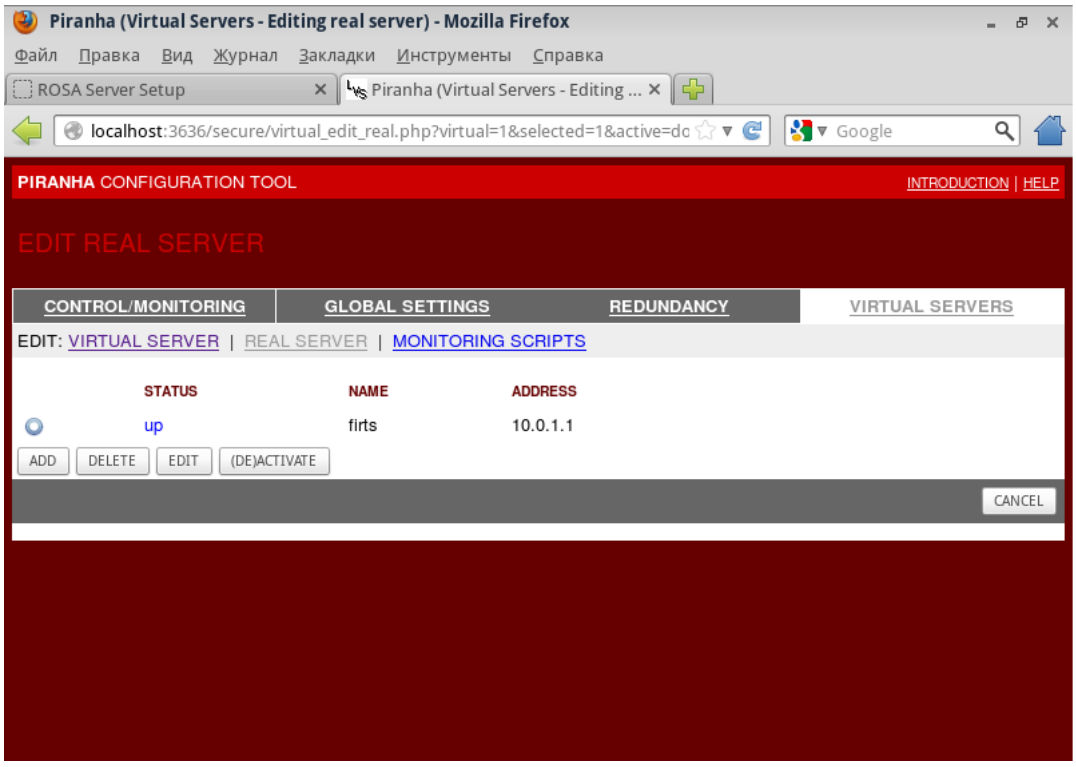

Рисунок 28 — Real server

Используется те же кнопки, что и при настройки виртуальных серверов. Рассмотрим процесс настройки подробнее.

В форме (см. рис. 29) необходимо заполнить следующие поля:

– **Name** — введите имя реального сервера

– **Address** — введите ip-адрес реального сервера в подсети

– **Port** — введите порт, используемый реальным сервером. Рекомендуем оставить пустым, будет использован порт, указанный для соответствующего виртуального сервера

– **Weight** — укажите вес сервера, используется как весовой коэффициент в алгоритмах распределения нагрузки. Указать необходимо натуральное число.

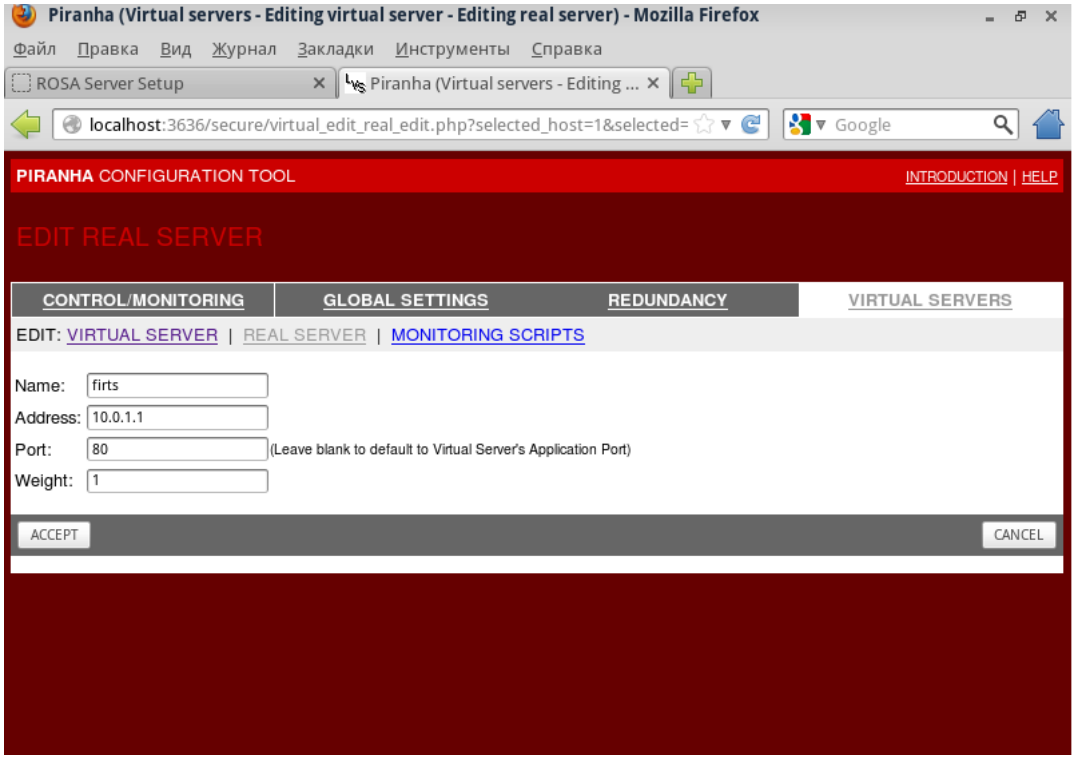

Рисунок 29 — Настройка реального сервера

Не забудьте нажать на кнопку *ACCEPT* внизу формы, чтобы подтвердить изменение настроек.

В подвкладке **Monitoring scripts** можно указать запросы (см. рис. 30), которые будут посылаться сервису для проверки его работоспособности.

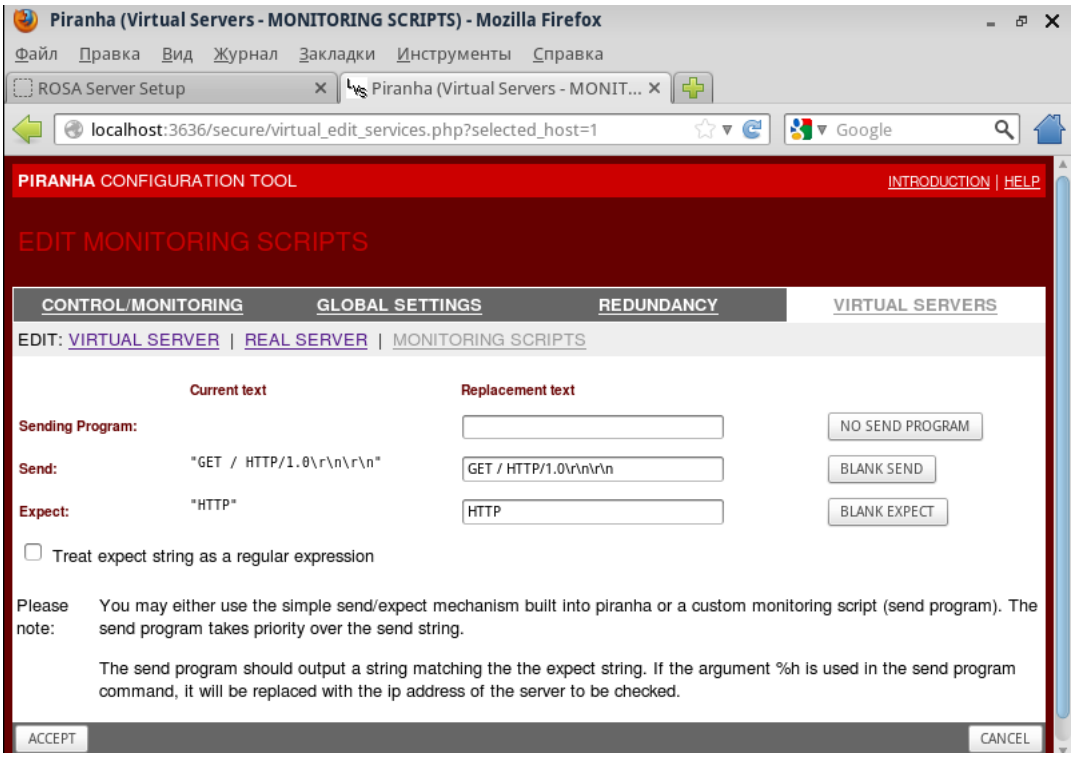

Рисунок 30 — Monitoring scripts

– **Sending Program** — используется при наличии скрипта, проверяющего работоспособность сервиса, укажите здесь путь к этому скрипту.

– **Send** — введите содержание сообщения, которое будет посылаться сервису (на реальный сервер) для проверки его работоспособности.

– **Expect** — введите ожидаемый ответ сервиса, который будет подтверждать его работоспособность.

Не забудьте нажать на кнопку *ACCEPT* внизу формы, чтобы подтвердить изменение настроек.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

GFS (Global File System) — распределенная файловая система, обеспечивающая прямой и одновременный доступ всех хостов кластера к блочным дисковым устройствам

RELS — ROSA Enterprise Linux Server

Selinux (Security-Enhanced Linux, Linux с улучшенной безопасностью) — реализация системы принудительного контроля доступа, реализован в ядре Linux

ВМ — виртуальная машина

ОС — операционная система

ПО — программное обеспечение

СХД — сеть хранения данных