

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего  
образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

**КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ**

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

профессор, д.т.н.

А. В. Гордеев

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Создание кластера в системе в Linux

по дисциплине: Вычислительные системы

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

№ Z0440M

11.04.2021

Митрофанов Д.А.

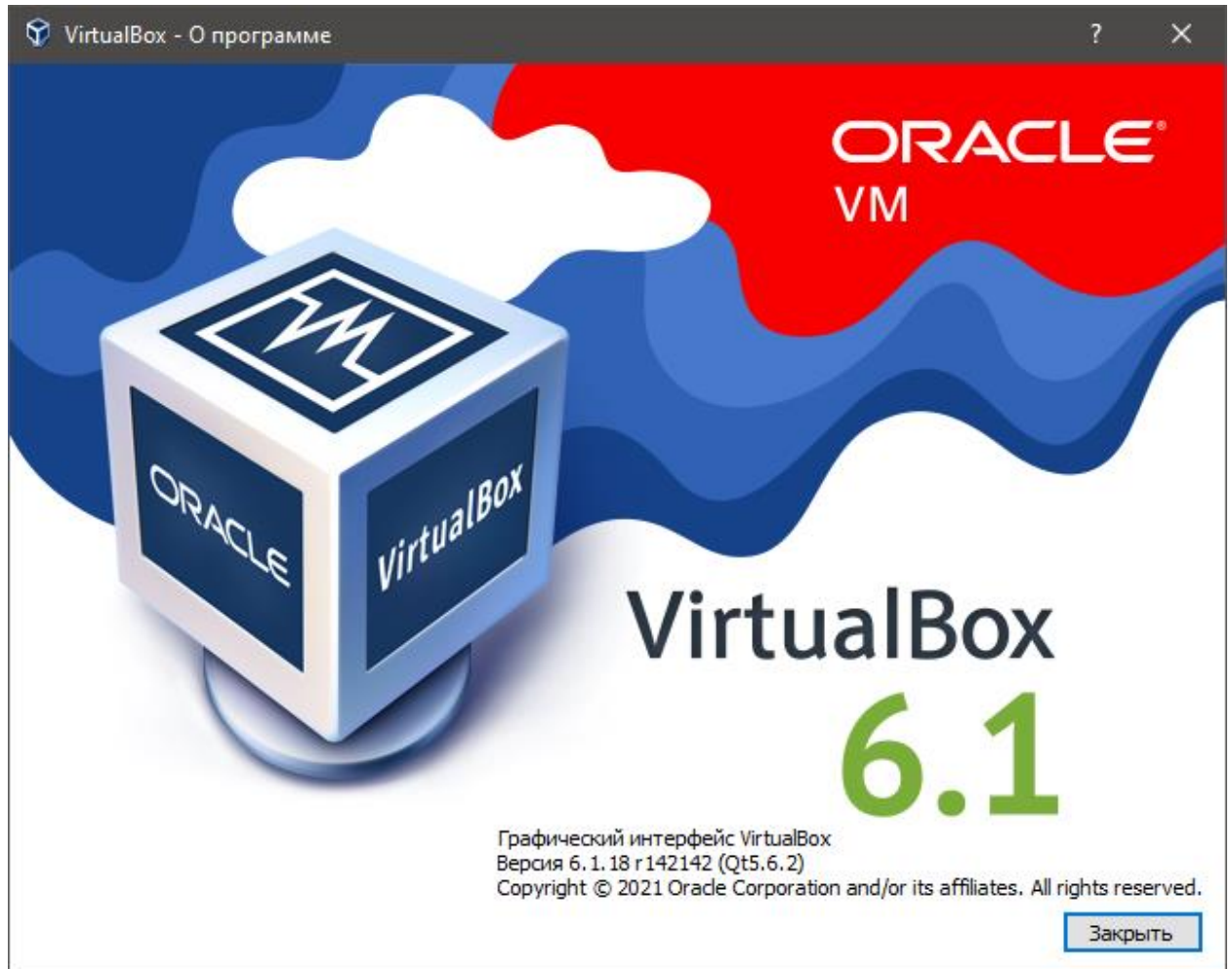
Санкт-Петербург 2021

## Цель работы

Изучить технологии создания NLB-кластеров в Linux

## 1 Исходные данные

Для выполнения лабораторной работы использовался гипервизор Oracle VM VirtualBox версии 6.1.18 (<https://www.virtualbox.org>)

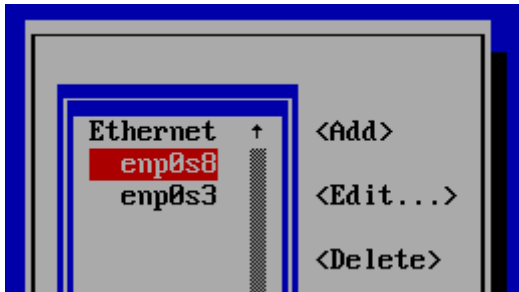


и образ CentOS 8 Stream, доступный бесплатно (<https://www.centos.org/centos-stream/>) на ядре 4.18

## 2 Ход выполнения

### Сеть

В гипервизоре было прокинуто два сетевых интерфейса (схема идентична прошлой лабораторной работе) на виртуальную машину. В самой ВМ было создано и настроено через утилиту `nmtui` два независимых интерфейса, смотрящих в разные сети



```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT link-down
    link/ether 08:00:27:eb:18:46 brd ff:ff:ff:ff:ff:ff
    inet 192.168.40.1/24 brd 192.168.40.255 scope global dynamic noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT link-down
    link/ether 08:00:27:25:db:4d brd ff:ff:ff:ff:ff:ff
    inet 172.16.40.1/16 brd 172.16.255.255 scope global dynamic noprefixroute
        valid_lft forever preferred_lft forever
```

Установлен веб-сервер `nginx 1.19.9` из официального репозитория

```
-----
Thanks for using nginx!

Please find the official documentation for nginx here:
* https://nginx.org/en/docs/

Please subscribe to nginx-announce mailing list to get
the most important news about nginx:
* https://nginx.org/en/support.html

Commercial subscriptions for nginx are available on:
* https://nginx.com/products/
-----

Verifying          : nginx-1:1.19.9-1.el8.ngx.x86_64
Installed:
  nginx-1:1.19.9-1.el8.ngx.x86_64
```

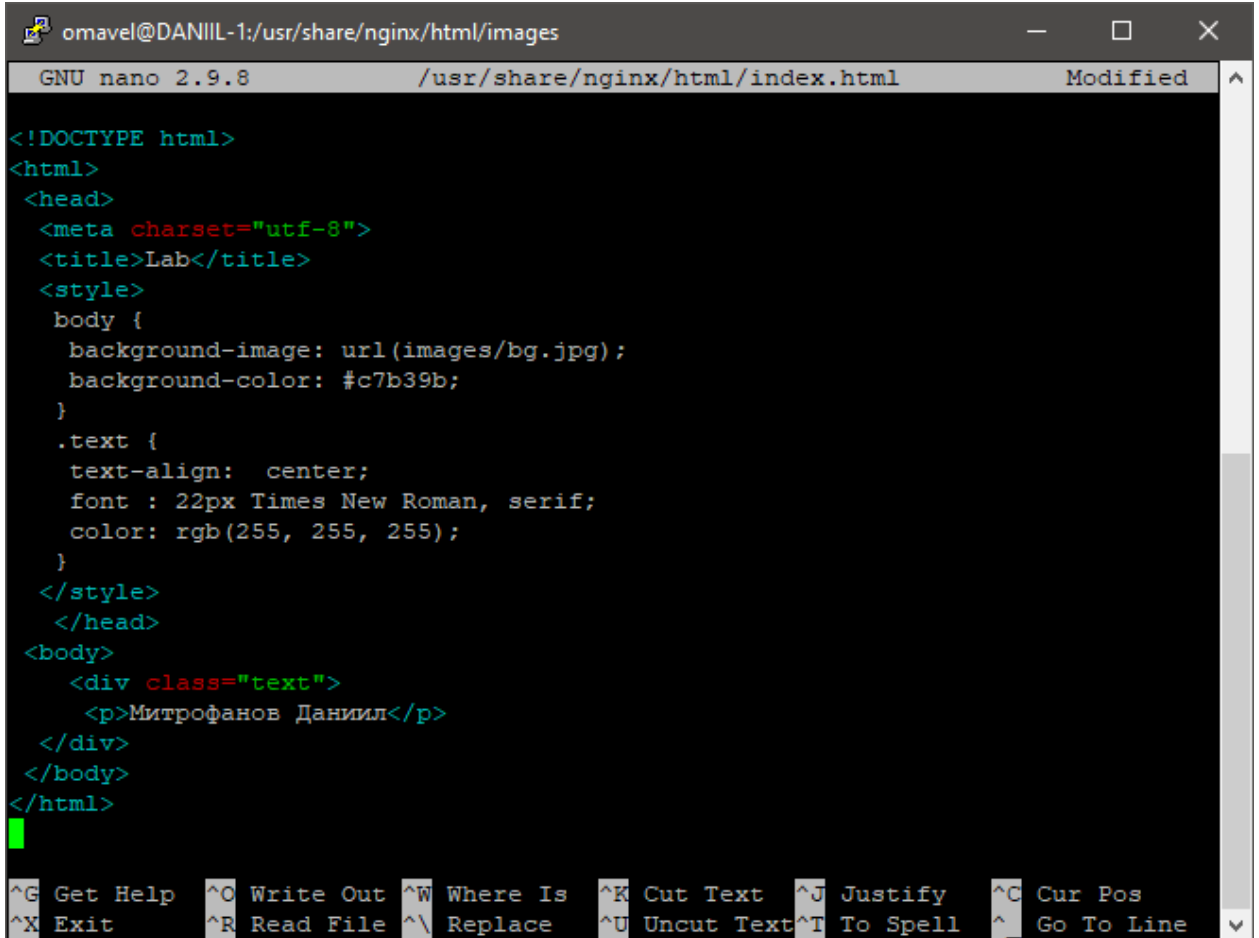
ВМ была клонирована с генерацией новых MAC-адресов, вторая машина переименована в Daniil-2, так же была исправлена адресация через `nmtui`

На обоих серверах был открыт 80 порт и добавлено правило для `selinux`:

```
sudo firewall-cmd --permanent --add-service=http
semanage permissive -a httpd_t
```

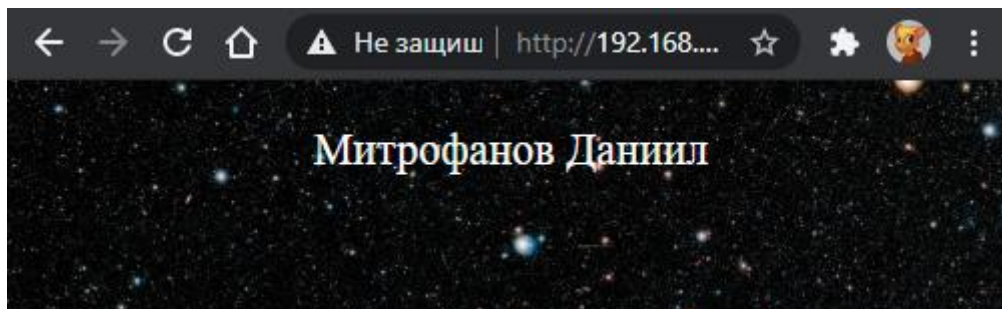
## Nginx

Была подготовлена html-страница на первой виртуальной машине



```
omavel@DANIIL-1:/usr/share/nginx/html/images
GNU nano 2.9.8 /usr/share/nginx/html/index.html Modified
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Lab</title>
<style>
body {
background-image: url(images/bg.jpg);
background-color: #c7b39b;
}
.text {
text-align: center;
font : 22px Times New Roman, serif;
color: rgb(255, 255, 255);
}
</style>
</head>
<body>
<div class="text">
<p>Митрофанов Даниил</p>
</div>
</body>
</html>
```

Она доступна по сети:



Второй сервер на данном этапе отдает стандартную страницу:



Для репликаций конфигурации `nginx` была использована утилита `rsync`, т.к. официальный репозиторий `csync2` оказался недоступен

```
yum install rsync rsync-daemon
```

```
systemctl enable rsyncd -now
```

Конфигурация:

```
pid file = /var/run/rsyncd.pid
```

```
lock file = /var/run/rsync.lock
```

```
log file = /var/log/rsyncd.log
```

```
use chroot = false
```

```
[nginx_content]
```

```
    path = /usr/share/nginx/html/
```

```
    comment = Site Content
```

```
    read only = yes
```

```
    list = yes
```

```
    auth users = rsync_d1
```

```
    secrets file = /etc/rsyncd.scrpt
```

```
    hosts allow = localhost 192.168.40.2 172.16.40.2
```

```
    hosts deny = *
```

Открываем порт, добавляем политику `selinux`:

```
firewall-cmd --add-service=rsyncd --permanent firewall-cmd --reload
```

```
setsebool -P rsync_full_access on
```

Проверим синхронизацию:

```
rsync -avzhHl --progress /usr/share/nginx/html/ rsuser@192.168.40.2:  
/usr/share/nginx/html/
```

```
sending incremental file list  
./  
index.html          412 100%   0.00kB/s   0:00:00 (xfr#1, to-chk=2/5)  
images/  
images/bg.jpg      1.12M 100%  11.71MB/s   0:00:00 (xfr#2, to-chk=0/5)  
  
sent 1.12M bytes  received 75 bytes  89.47K bytes/sec  
total size is 1.12M  speedup is 1.00
```

Добавим скрипт в cron для автоматического запуска каждые 10 минут

```
!/bin/sh
```

```
RSYNC=/usr/bin/rsync
```

```
SSH=/usr/bin/ssh
```

```
RUSER=rsuser
```

```
RPATH=/usr/share/nginx/html/
```

```
LPATH=/usr/share/nginx/html/
```

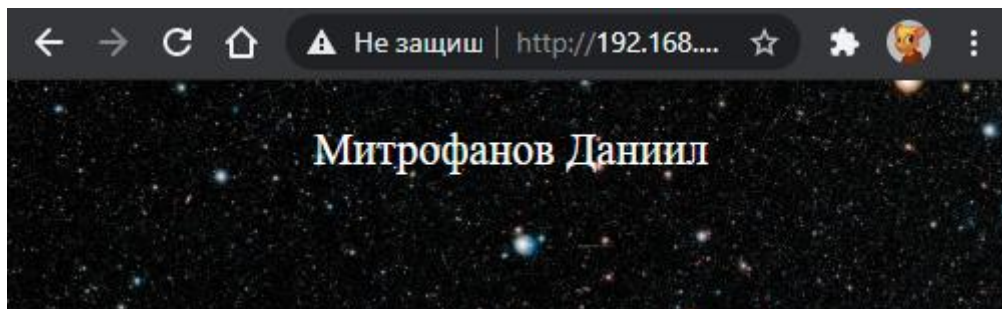
```
$RSYNC -az -e "$SSH" $RUSER@$RHOST:$RPATH $LPATH
```

```
crontab -e
```

```
*/10 * * * * /root/scripts/rsync_nginx_content.sh
```

Синхронизация прошла и сайт доступен:

```
[root@DANIIL-2 src]# ls -al /usr/share/nginx/html/
total 8
drwxr-xr-x. 3 root root 54 Apr 10 23:53 .
drwxr-xr-x. 3 root root 18 Apr 10 23:16 ..
-rw-r--r--. 1 root root 494 Mar 30 18:25 50x.html
drwxr-xr-x. 2 root root 20 Apr 10 23:54 images
-rw-r--r--. 1 root root 412 Apr 10 23:55 index.html
```



Недостатком `rsync` перед `csync2` является то, что `rsync` не отслеживает изменение файлов и поэтому не может инициировать репликацию по признаку их изменения.

## Кластеризация

Т.к. задача кластеризации на этом не совсем раскрыта, я решил добавить кластер высокой доступности `nginx` на основе ресурс-менеджера `pacemaker`

Назначим имена хостам через `/etc/hosts`

```
192.168.1.104 daniil-1
192.168.1.105 daniil-2
```

Установим необходимые пакеты:

```
yum --enablerepo=ha -y install pacemaker pcs corosync
```

Добавим правило в firewall:

```
firewall-cmd --permanent --add-service=high-availability
```

```
firewall-cmd --add-service=high-availability
```

Активируем службы:

```
systemctl enable pacemaker
```

```
systemctl enable corosync
```

```
systemctl enable pcsd
```

И запустим службу pcsd для синхронизации конфигурации:

```
systemctl start pcsd
```

В процессе установки был создан новый пользователь hacluster, поэтому надо задать ему пароль:

```
passwd hacluster
```

Авторизуем обе машины в кластере:

```
pcs host auth daniil-1 daniil-2
```

```
Username: hacluster
```

```
daniil-1: Authorized
```

```
daniil-2: Authorized
```

Далее для настройки кластера можно воспользоваться либо командной строкой

```
pcs cluster setup clusteredsite daniil-1 daniil-2 --start -enable
```

```
Requesting remove 'pcsd settings' from 'daniil-1', 'daniil-2'
daniil-1: successful removal of the file 'pcsd settings'
daniil-2: successful removal of the file 'pcsd settings'
Sending 'corosync authkey', 'pacemaker authkey' to 'daniil-1', 'daniil-2'
daniil-1: successful distribution of the file 'corosync authkey'
daniil-1: successful distribution of the file 'pacemaker authkey'
daniil-2: successful distribution of the file 'corosync authkey'
daniil-2: successful distribution of the file 'pacemaker authkey'
Sending 'corosync.conf' to 'daniil-1', 'daniil-2'
daniil-1: successful distribution of the file 'corosync.conf'
daniil-2: successful distribution of the file 'corosync.conf'
Cluster has been successfully set up.
```

либо веб-интерфейсом, доступным на 2224 порту:

The screenshot shows the 'HIGH AVAILABILITY MANAGEMENT' web interface. The top navigation bar includes 'MANAGE CLUSTERS' and 'PERMISSIONS'. A dropdown menu shows 'hacluster'. The main content area is titled 'MANAGE CLUSTERS' and contains a table with columns for 'All', 'Remove', 'Add Existing', 'Destroy', and 'Create New', all showing '0'. To the right, there is a section 'INFORMATION ABOUT CLUSTERS' with the text 'Select a cluster to view more detailed cluster information'. A modal dialog box titled 'Create cluster: Cluster name and nodes' is open. It contains a 'Cluster Name' field with the value 'clusteredsute'. Below it, a 'Nodes' section asks to 'Enter the names of the nodes you would like to use to create a cluster.' and lists three nodes: 'Node 1: daniil-1', 'Node 2: daniil-2', and 'Node 3:'. A 'More nodes...' link is present. At the bottom of the dialog are three buttons: 'Create cluster', 'Go to advanced settings', and 'Cancel'.

The screenshot shows the 'HIGH AVAILABILITY MANAGEMENT' web interface. The top navigation bar includes 'Cluster: clusteredsute' and 'hacluster'. The main content area is titled 'NODES' and contains a list of nodes: 'daniil-1' and 'daniil-2'. The 'daniil-1' node is selected, and the 'Edit Node daniil-1' page is displayed. The page shows the node name 'daniil-1' in a text field, with status indicators for 'Pacemaker Connected' and 'Corosync Connected'. Below this are several action buttons: 'Start', 'Stop', 'Restart', 'Standby', 'Maintenance', and 'Configure Fencing'. The 'Node ID' is '1' and the 'Uptime' is '0 days, 00:35:20'. A table titled 'Cluster Daemons' shows the status of various daemons:

NAME	STATUS
pacemaker	Running (Disabled)
corosync	Running (Disabled)
pcsd	Running (Enabled)

Вернемся к CLI и Проверим статус кластера:

```
pcs status
```

```
No stonith devices and stonith-enabled is not false

Cluster Summary:
* Stack: corosync
* Current DC: daniil-2 (version 2.0.5-9.e18-ba59be7122) - partition with quorum
* Last updated: Sun Apr 11 11:48:16 2021
* Last change: Sun Apr 11 11:47:49 2021 by hacluster via crmd on daniil-2
* 2 nodes configured
* 0 resource instances configured

Node List:
* Online: [ daniil-1 daniil-2 ]

Full List of Resources:
* No resources

Daemon Status:
corosync: active/disabled
pacemaker: active/disabled
pcsd: active/enabled
```

Проверим синхронизацию:

```
pcs status corosync
```

[root@DANIIL-1 omavel]# pcs status corosync				[root@DANIIL-2 omavel]# pcs status corosync			
Membership information				Membership information			
Nodeid	Votes	Name		Nodeid	Votes	Name	
1	1	daniil-1 (local)		1	1	daniil-1	
2	1	daniil-2		2	1	daniil-2 (local)	

Пусть плавающим ip будет 192.168.1.111

```
pcs resource create v_ip ocf:heartbeat:IPaddr2 ip=192.168.1.111
cidr_netmask=32 op monitor interval=10s
```

Добавим ресурсы

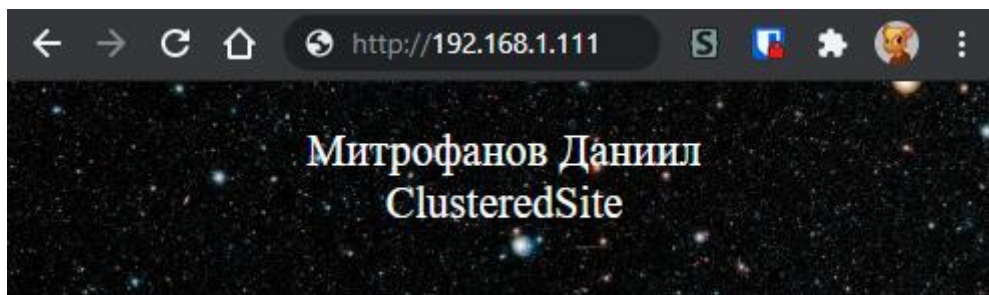
```
pcs resource create webserver ocf:heartbeat:nginx
configfile=/etc/nginx/nginx.conf op monitor timeout="5s" interval="5s"
```

```
pcs resource create content ocf:heartbeat:nginx
configfile=/etc/nginx/nginx.conf op monitor timeout="5s" interval="5s"
```

```
pcs resource create content ocf:heartbeat:nginx
configfile=/usr/share/nginx/html/ op monitor timeout="5s"
interval="5s"
```

При обращении к плавающему IP, открывается сайт.

Не смотря на то, что файлы сайта создавались на Daniil-1, активной нодой является Daniil-2. Таким образом можно сделать вывод, что синхронизация ресурсов прошла успешно:



## 5 Выводы

Были изучены технологии кластеризации в Linux