

Отчет

Авдашков Александр
+7 904 6400149
moralpsy@gmail.com

Задание 1.1

Установил VirtualBox, создал виртуальную машину (1 ЦП, 1024 ОЗУ, подключение типа МОСТ, IP 192.168.1.98)

тает)

▼ работа с nginx.config
▼ nginx работает
→ Текущее состояние

CentOSVM (nginx работает) [Работает] - Oracle VM VirtualBox

Файл Машина Вид Ввод Устройства Справка

CentOS Linux 8
Kernel 4.18.0-305.10.2.el8_4.x86_64 on an x86_64

nginx)

Activate the web console with: systemctl enable --now cockpit.socket

localhost login: root
Password:
Last login: Mon Aug 2 17:06:49 on tty1
[root@localhost ~]# ifconfig

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.98 netmask 255.255.255.0 broadcast 192.168.1.255
inet6 fe80::a00:27ff:fe14:61a9 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:14:61:a9 txqueuelen 1000 (Ethernet)
RX packets 63 bytes 5745 (5.6 KiB)
RX errors 0 dropped 4 overruns 0 frame 0
TX packets 57 bytes 4582 (4.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 48 bytes 3028 (2.9 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 48 bytes 3028 (2.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# _

Установил PuTTY, подключился через SSH, установил Python

Ansible Менеджер

Снимок Справка

ты

(nginx работает)

т

! (nginx2 работает)

на

nsible)

на

ана

(гол

ана

(гол

ана

элый)

Сделать Удалить Восстановить Свойства Клон

Имя

после установки nginx

с автозапуском nginx

работа с nginx.config

nginx работает

Текущее состояние (из

root@localhost:~

login as: root

root@192.168.1.99's password:

Activate the web console with: systemctl enable --now cockpit.socket

Last login: Mon Aug 2 15:34:05 2021 from 192.168.1.58

[root@localhost ~]# python3 --version

Python 3.6.8

[root@localhost ~]#

Задание 2.1

На созданную виртуальную машину CentOSVM установил nginx. После выполнения yum update была доступна только старая версия. Актуальную версию качал с репозитория nginx https://nginx.org/en/linux_packages.html#RHEL-CentOS, установил. Установил автозагрузку, запустил.

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# systemctl status nginx
nginx.service - nginx - high performance web server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-08-02 15:35:34 EDT; 27min ago
     Docs: http://nginx.org/en/docs/
  Process: 1068 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf (code=exited, status=0/SUCCESS)
 Main PID: 1093 (nginx)
    Tasks: 2 (limit: 4941)
   Memory: 3.4M
    CGroup: /system.slice/nginx.service
            └─1093 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf
              └─1098 nginx: worker process

abr 02 15:35:33 localhost.localdomain systemd[1]: Starting nginx - high performance web server...
abr 02 15:35:34 localhost.localdomain systemd[1]: nginx.service: Can't open PID file /var/run/nginx
abr 02 15:35:34 localhost.localdomain systemd[1]: Started nginx - high performance web server.
lines 1-15/15 (END)
```

Создал страницу HelloWorld /home/user/sites/site1/index.html и отредактировал nginx.conf:

- создал server site1.centosvm
- установил прослушивание порта 80
- установил домашнюю директорию сайта /home/user/sites/site1/
- в качестве индексного файла указал index.html

```
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    #tcp_nopush on;

    keepalive_timeout 65;

    #gzip on;

    include /etc/nginx/conf.d/*.conf;
    server {
        server_name site1.centosvm;
        listen 80;
        location / {
            root /home/user/sites/site1/;

            index index.html;
        }
    }
}
```

38,1-8 Внизу

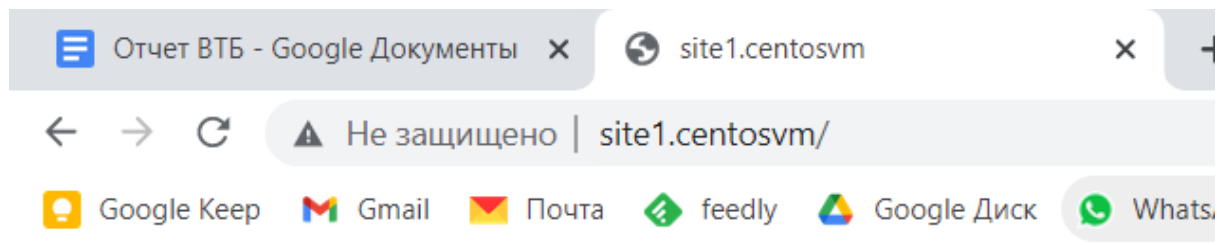
Открыл в постоянном режиме 80 порт через фаервол firewall-cmd --add-service http --permanent

Отредактировал свой файл hosts, добавил строки:

192.168.1.98 centosvm site1.centosvm

192.168.1.99 centosvm2 site1.centosvm2

Проверил открытие страницы:



Hello, World

Server 1 IP 192.168.1.98

Клонировал в VirtualBox виртуальную машину CentOSVM. Вторую назвал CentOSVM2. Поменял IP на 192.168.1.99 через утилиту nmtui. Отредактировал index.html, указав Server 2 IP 192.168.1.99

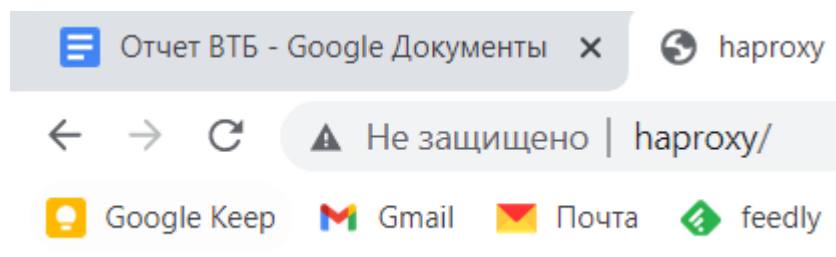
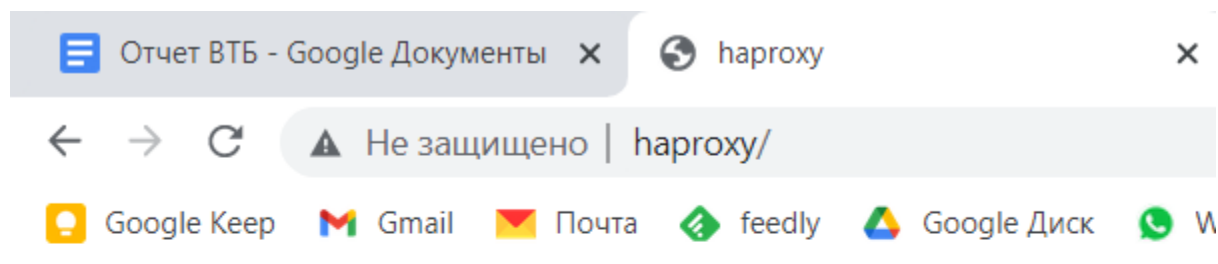
Создал виртуальную машину HAProxy (ip 192.168.1.61), установил CentOS и HAProxy. Отредактировал конфигурационный файл haproxy.cfg, указав слушать порт 80 (с любых ip) и балансировать нагрузку на 2-ух веб-серверах (centosvm и centosvm2) методом roundrobin (поочередно, без указания веса каждого сервера).

```
#####
frontend centosvm_front
    bind *:80
    default_backend centosvm_back

backend centosvm_back
    balance roundrobin
    server site1.centosvm 192.168.1.98:80
    server site1.centosvm2 192.168.1.99:80
#####
"/etc/haproxy/haproxy.cfg" 101L, 3559C
```

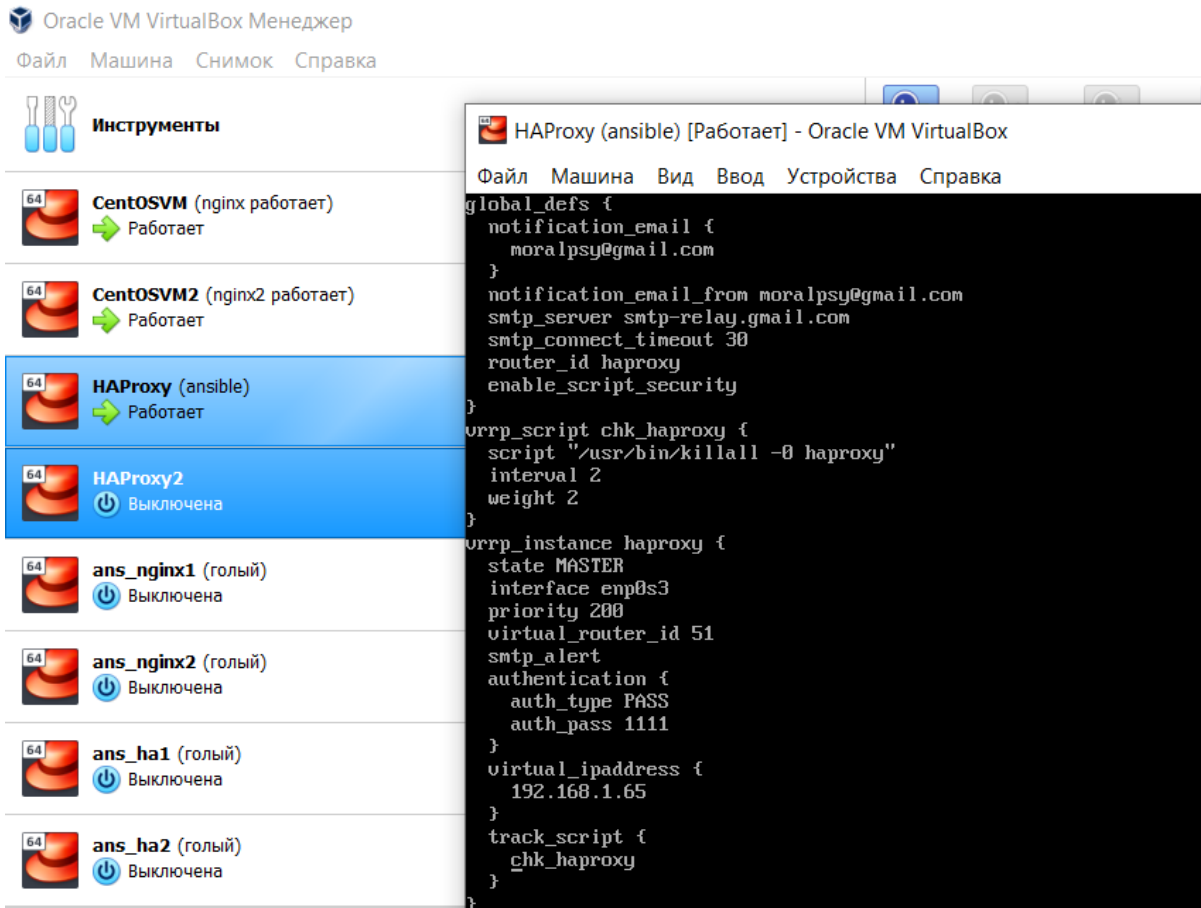
Добавил в свой hosts строчку:
192.168.1.61 haproxy

Проверил работу балансировщика:



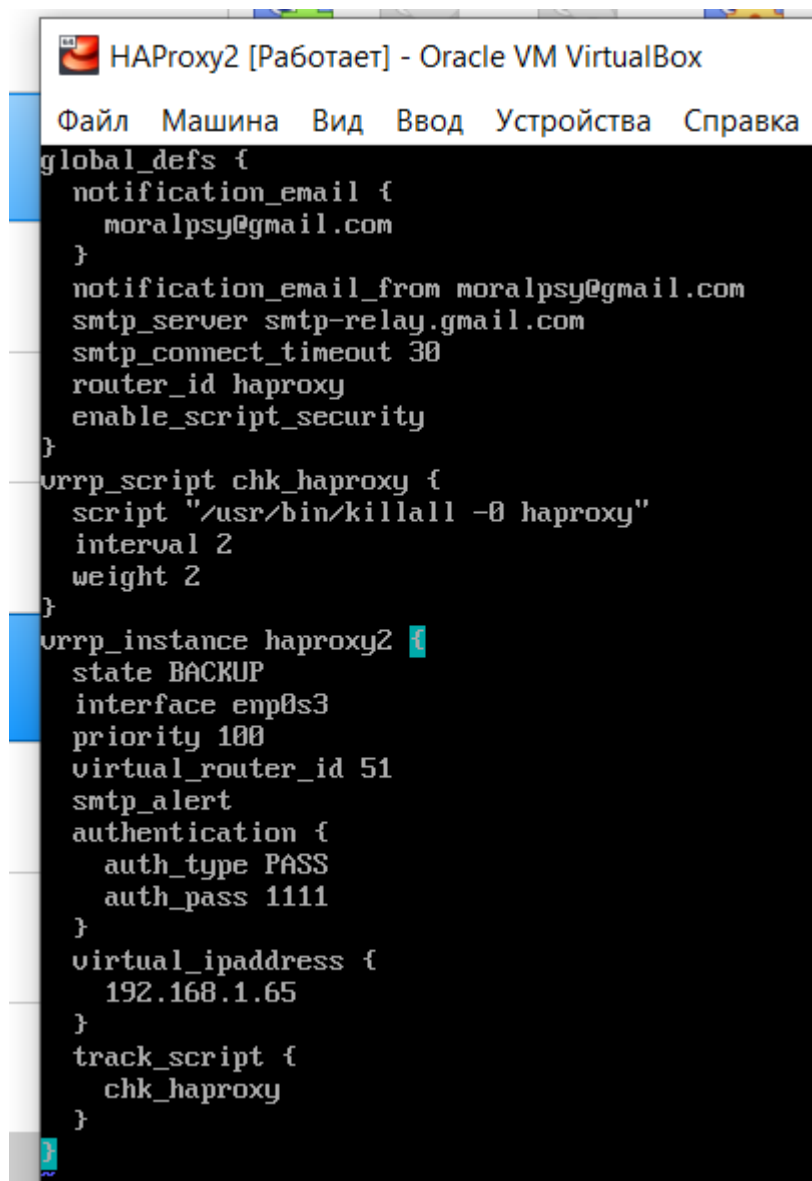
Клонировал виртуальную машину HAProxy (ip 192.168.1.62). Установил keepalived на обе виртуальные машины с балансировщиками. Отредактировал keepalived.conf сначала на HAProxy:

- устанавливаем почту для уведомлений
- этот хост будет основным, устанавливаем state MASTER
- соответственно повышаем ей приоритет
- авторизацию между инстансами устанавливаем через пароль PASS
- виртуальный адрес устанавливаем в той же сети 192.168.1.65



На втором хосте HAProxy2:

- устанавливаем состояние state BACKUP
- приоритет ниже - 100
- устанавливаем virtual_router_id одинаковый для обоих хостов - 51
- виртуальный ip адрес такой же - 192.168.0.65

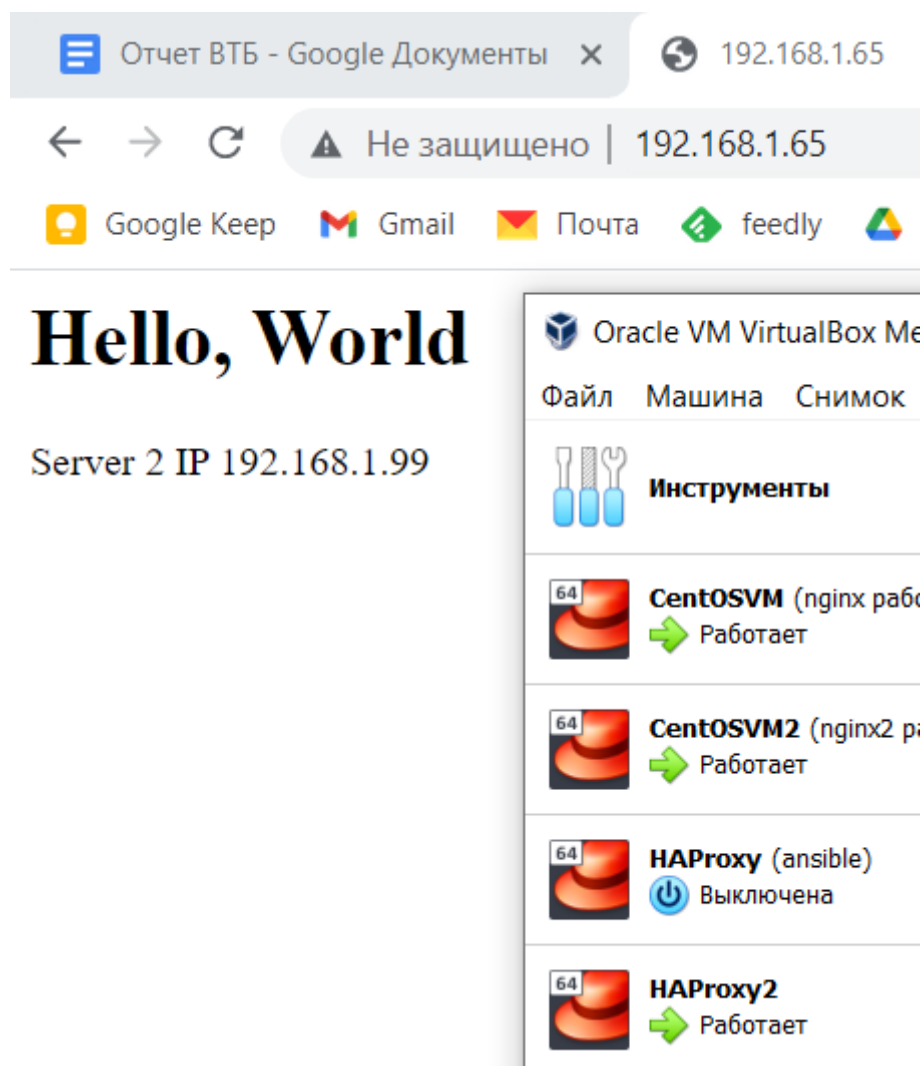
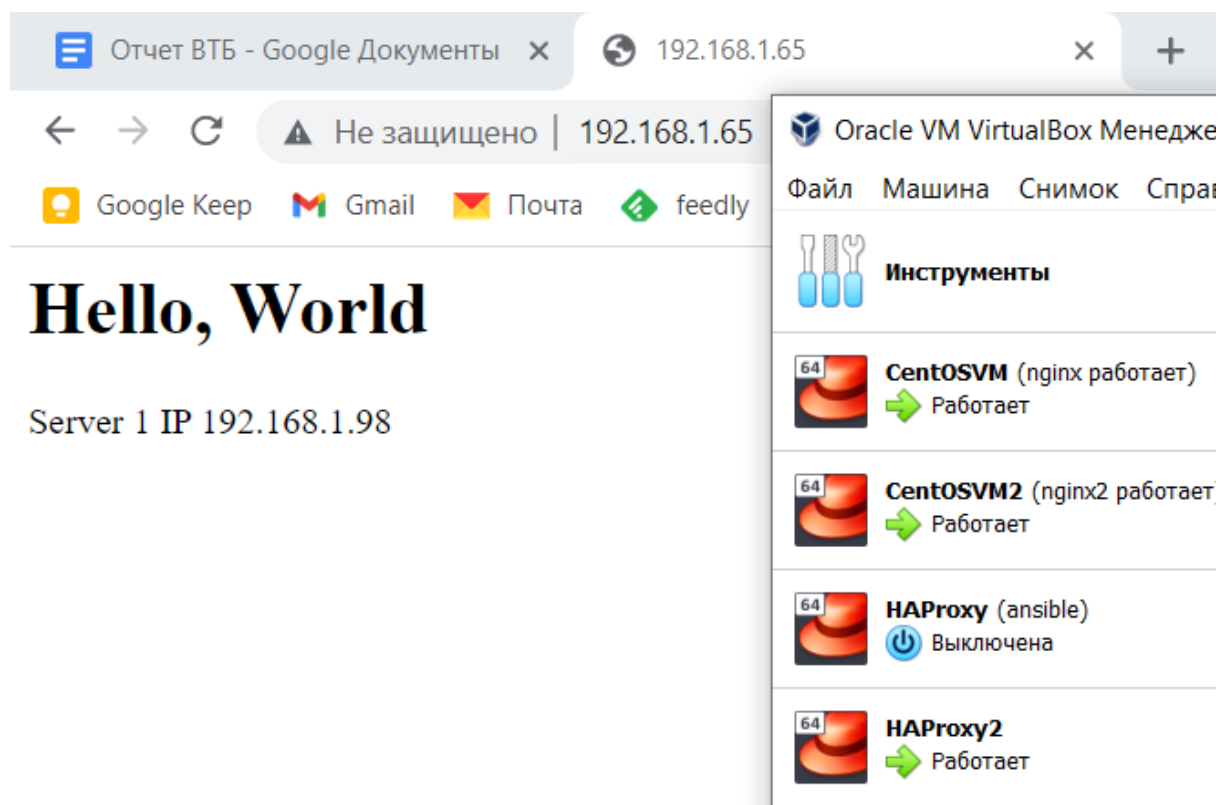


```
global_defs {
    notification_email {
        moralpsy@gmail.com
    }
    notification_email_from moralpsy@gmail.com
    smtp_server smtp-relay.gmail.com
    smtp_connect_timeout 30
    router_id haproxy
    enable_script_security
}

errp_script chk_haproxy {
    script "/usr/bin/killall -0 haproxy"
    interval 2
    weight 2
}

errp_instance haproxy2 {
    state BACKUP
    interface enp0s3
    priority 100
    virtual_router_id 51
    smtp_alert
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.1.65
    }
    track_script {
        chk_haproxy
    }
}
```

Теперь при выключении виртуальной машины HAProxy доступность веб-серверов сохраняется:



Задание 2.2

Установил Ansible на виртуальную машину HAProxy. Создал конфигурационный файл `ansible.cfg`, где

- указал место инвентори-файла `hosts`
- отключил длинные предупреждения `deprecation warnings`
- отключил проверку `ssl-ключа`

```
[root@localhost ansible]# cat ansible.cfg
[defaults]

inventory = /root/ansible/hosts
host_key_checking = false
deprecation_warnings=false
[root@localhost ansible]#
```

Создал инвентори-файл `hosts`

```
[nginx]
ans_nginx1 ansible_host=192.168.1.71 ansible_user=root ansible_password=314mbo virtual_domain=site1.centosum
ans_nginx2 ansible_host=192.168.1.72 ansible_user=root ansible_password=314mbo virtual_domain=site1.centosum2
[haproxy]
ans_ha1 ansible_host=192.168.1.73 ansible_user=root ansible_password=314mbo
ans_ha2 ansible_host=192.168.1.74 ansible_user=root ansible_password=314mbo
```

В `hosts` указал адреса предварительно созданных 4 виртуальных машин: 2 для веб-серверов, 2 для балансировщика `haproxy`. Разделил их, соответственно, на 2 группы. Для каждого указал логин-пароль для входа по `ssh`.

Проверил доступность:

```
[root@localhost ansible]# ansible all -m ping
ans_nginx1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
ans_ha2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
ans_ha1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
ans_nginx2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
[root@localhost ansible]#
```

Создал файл плейбука `playbook.yml`:

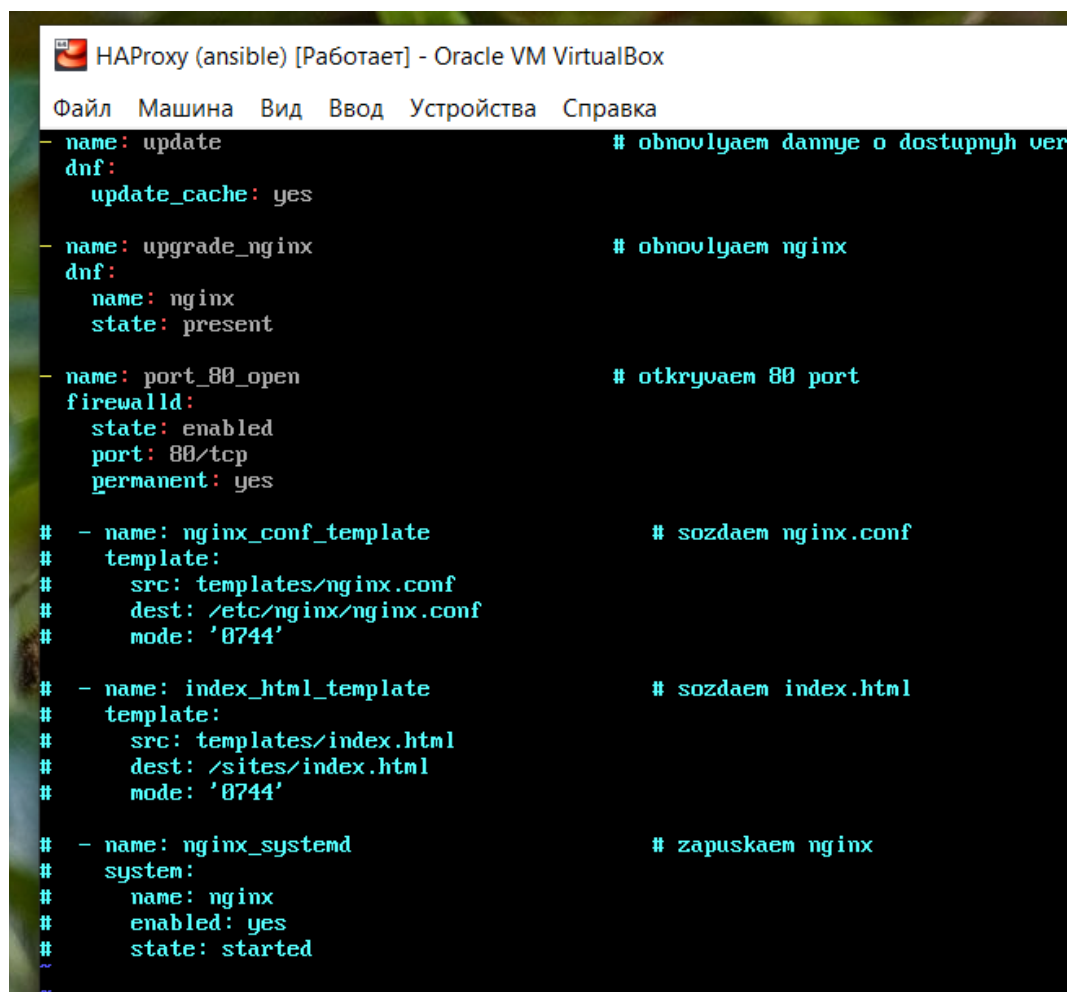
```
    "ping": "pong"
  }
[root@localhost ansible]# cat playbook.yml
- hosts: nginx
  become: yes
  roles:
    - role: nginx

- hosts: haproxy
  become: yes
  roles:
    - role: haproxy
    - role: keepalived
[root@localhost ansible]#
```

В нём разделил роли по 2 группам хостов: `nginx`, для веб-серверов и `haproxy`, для балансировщика и `keepalived`.

В соответствии с документацией

https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html#role-directory-structure для роли `nginx` создал файл `main.yml` в директории `ansible/roles/nginx/tasks/`



```
HAProxy (ansible) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
- name: update                                # obnovlyaem dannye o dostupnyh ver
  dnf:
    update_cache: yes
- name: upgrade_nginx                         # obnovlyaem nginx
  dnf:
    name: nginx
    state: present
- name: port_80_open                          # otkryvaem 80 port
  firewallld:
    state: enabled
    port: 80/tcp
    permanent: yes

# - name: nginx_conf_template                 # sozdaem nginx.conf
#   template:
#     src: templates/nginx.conf
#     dest: /etc/nginx/nginx.conf
#     mode: '0744'

# - name: index_html_template                 # sozdaem index.html
#   template:
#     src: templates/index.html
#     dest: /sites/index.html
#     mode: '0744'

# - name: nginx_systemd                       # zapuskaem nginx
#   system:
#     name: nginx
#     enabled: yes
#     state: started
```

Аналогичную структуру создал для ролей haproxy и keepalived:

```
[root@localhost ansible]# ls -R
.:
ansible.cfg  hosts  playbook.yml  roles

./roles:
haproxy  keepalived  nginx

./roles/haproxy:
tasks

./roles/haproxy/tasks:
main.yml

./roles/keepalived:
tasks

./roles/keepalived/tasks:
main.yml

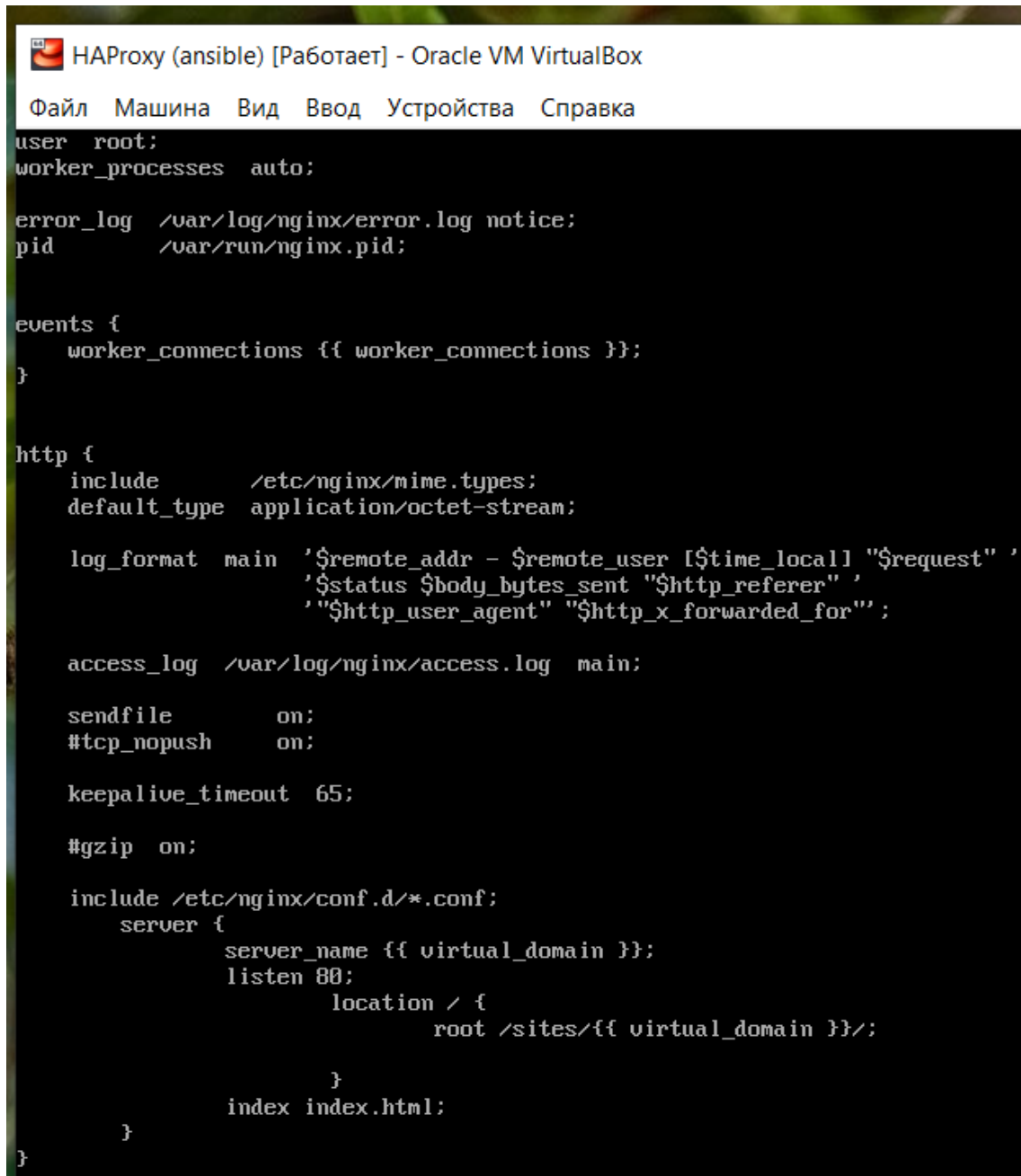
./roles/nginx:
tasks  templates  vars

./roles/nginx/tasks:
main.yml

./roles/nginx/templates:
index.html  nginx.conf

./roles/nginx/vars:
main.yml
[root@localhost ansible]#
```

Для создания nginx.conf необходимо воспользоваться шаблонами. Создал каталог шаблонов templates в каталоге роли nginx. Отредактировал шаблон nginx.conf:



```
HAProxy (ansible) [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
user root;
worker_processes auto;

error_log /var/log/nginx/error.log notice;
pid /var/run/nginx.pid;

events {
    worker_connections {{ worker_connections }};
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for" ';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    #tcp_nopush on;

    keepalive_timeout 65;

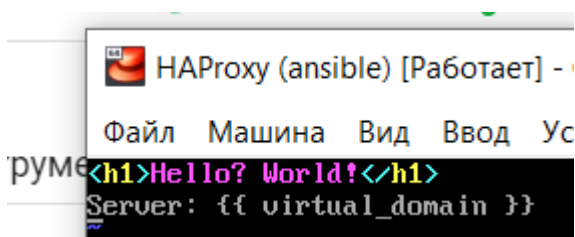
    #gzip on;

    include /etc/nginx/conf.d/*.conf;
    server {
        server_name {{ virtual_domain }};
        listen 80;
        location / {
            root /sites/{{ virtual_domain }};

            index index.html;
        }
    }
}
```

В строке с адресом корневого каталога сайта указал переменную virtual_domain, предварительно указав её в файле hosts

Создал шаблон для index.html:



```
HAProxy (ansible) [Работает] -
Файл  Машина  Вид  Ввод  Ус
руме<h1>Hello? World!</h1>
Server: {{ virtual_domain }}
```

На момент написания отчёта остановился на решении проблемы с открытием 80 порта: task обрабатывает, но netstat на целевой машине не показывает открытый порт.