

CaseLab4

Решение подготовил
Мясников Андрей



<https://github.com/evilbebra/astralinux-nginx-saltstack>

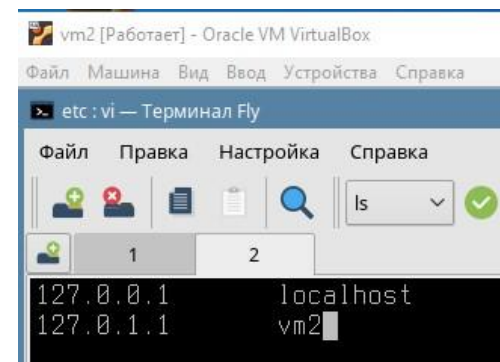
Подготовка стенда

После установки ОС Astra Linux в VirtualBox и полного копирования сделаем следующие шаги:

- Обновим мак адреса, чтобы они стали уникальными
- Обновим hostname для vm2
- Заменим в файлике hosts vm1 на vm2

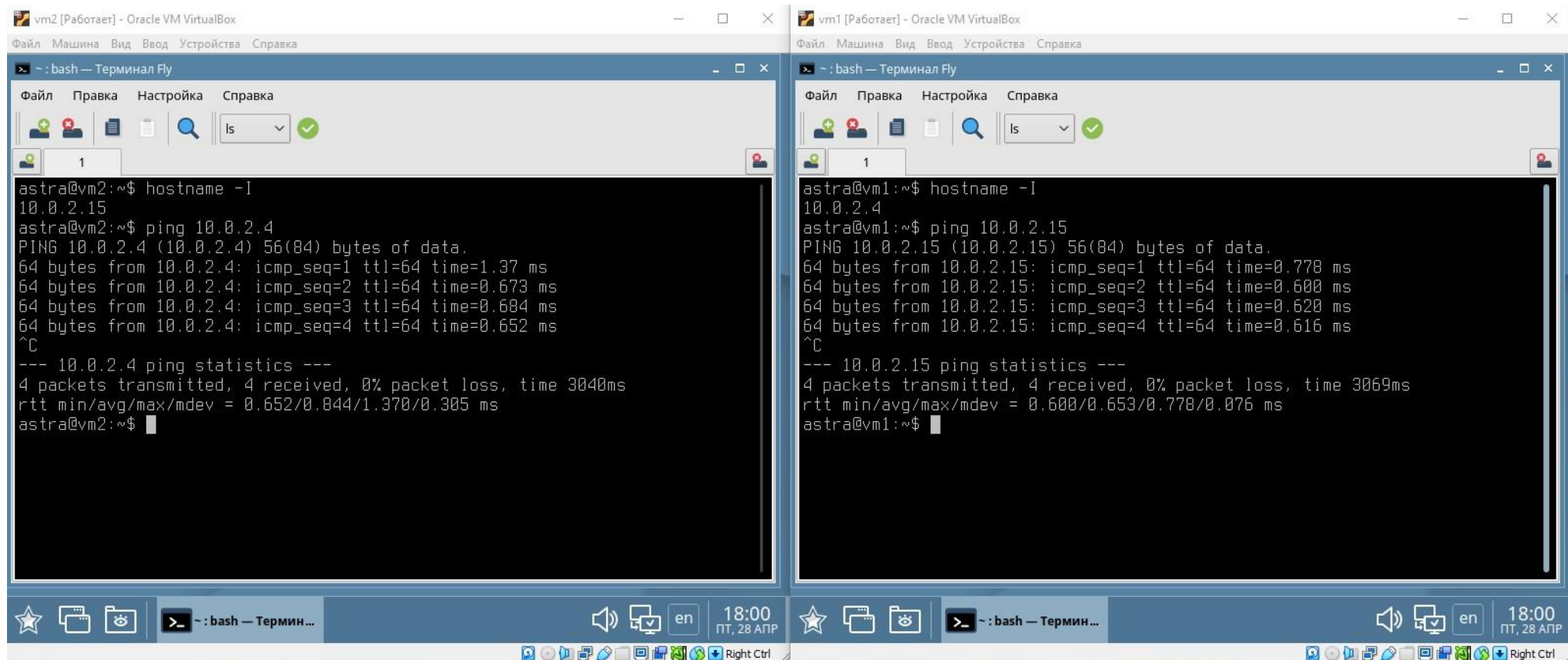


```
astra@vm1:/$ hostnamectl set-hostname vm2
```



Подготовка стенда

- Проверим ping на двух машинах:



The image shows two side-by-side screenshots of Oracle VM VirtualBox terminal windows. The left window, titled 'vm2 [Работает] - Oracle VM VirtualBox', shows a terminal session for 'astra@ym2'. The user runs 'hostname -I' and gets '10.0.2.15'. Then they run 'ping 10.0.2.4', which shows four successful pings with times around 1.37ms, 0.673ms, 0.684ms, and 0.652ms. Finally, they press '^C' to stop the ping and see the statistics: '--- 10.0.2.4 ping statistics ---', '4 packets transmitted, 4 received, 0% packet loss, time 3040ms', and 'rtt min/avg/max/mdev = 0.652/0.844/1.370/0.305 ms'. The right window, titled 'vm1 [Работает] - Oracle VM VirtualBox', shows a terminal session for 'astra@ym1'. The user runs 'hostname -I' and gets '10.0.2.4'. Then they run 'ping 10.0.2.15', which shows four successful pings with times around 0.778ms, 0.600ms, 0.620ms, and 0.616ms. Finally, they press '^C' to stop the ping and see the statistics: '--- 10.0.2.15 ping statistics ---', '4 packets transmitted, 4 received, 0% packet loss, time 3069ms', and 'rtt min/avg/max/mdev = 0.600/0.653/0.778/0.076 ms'. Both windows have a taskbar at the bottom showing the time as 18:00 on Friday, August 28.

```
astra@ym2:~$ hostname -I
10.0.2.15
astra@ym2:~$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data:
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=1.37 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.673 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=0.684 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=0.652 ms
^C
--- 10.0.2.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3040ms
rtt min/avg/max/mdev = 0.652/0.844/1.370/0.305 ms
astra@ym2:~$
```

```
astra@ym1:~$ hostname -I
10.0.2.4
astra@ym1:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data:
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.778 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.600 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.620 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.616 ms
^C
--- 10.0.2.15 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.600/0.653/0.778/0.076 ms
astra@ym1:~$
```

Установка и настройка Saltstack master vm2

Пропишем следующие команды в терминале:

- `sudo apt update`
- `sudo apt -y install salt-api salt-cloud salt-master salt-ssh salt-syndic`
- `sudo ufw allow proto tcp from any to any port 4505,4506`

Изменим в файлике `/etc/salt/master`:

`#interface: 0.0.0.0 -> interface: 10.0.2.15`

Пропишем `/etc/init.d/salt-master restart`

```
# The address of the interface to bind to:  
interface: 10.0.2.15
```

Установка и настройка Saltstack master vm2

Выведем на экран главный ключ, который необходим для подключения управляемых узлов, выполнив следующую команду:

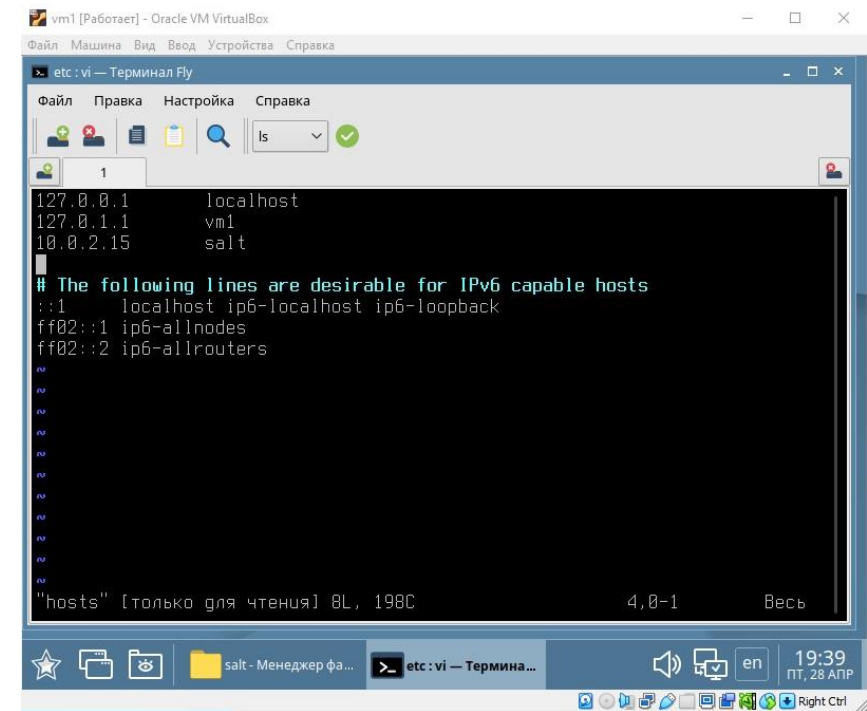
- `salt-key -F master`

```
astra@vm2:~$ hostname -I
10.0.2.15
astra@vm2:~$ /etc/init.d/salt-master restart
[ ok ] Restarting salt-master (via systemctl): salt-master.service.
astra@vm2:~$ salt-key -F master
CRITICAL: Salt configured to run as user "root" but unable to switch.
astra@vm2:~$ sudo salt-key -F master
[sudo] пароль для astra:
Local Keys:
master.pem: 2a:57:84:2d:a5:d2:ed:cb:88:cf:7d:56:d1:19:0f:a1:69:fe:5d:8c:4
4:4b:4a:f3:ea:35:34:5f:0a:e6:b5:d2
master.pub: 13:1c:e6:95:7a:cc:95:ca:d8:8d:48:27:81:cd:7c:1c:a0:60:40:c5:d
f:87:75:bb:e5:46:9d:ef:1d:35:0a:47
astra@vm2:~$
```

Установка и настройка Saltstack minion vm1

- Укажем в файлике hosts `/etc/hosts` ip master и значение salt
- Проверим идентификатор minion в `/etc/salt/minion_id` -> он равен vm1, все ок
- В файлике конфигурации добавим master.pub ключ значению master_finger:
master_finger: '13:1c:e6:95:7a:cc:95:ca:d8:8d:48:27:81:cd:7c:1c:a0:60:40:c5:df:87:75:bb:e5:46:9d:ef:1d:35:0a:47'
- `/etc/init.d/salt-minion restart`

```
# Set the location of the salt master server. If the master server cannot  
be  
# resolved, then the minion will fail to start.  
#master: salt
```



The screenshot shows a terminal window titled "etc:vi — Терминал Fly" within an Oracle VM VirtualBox environment. The terminal displays the contents of the `/etc/hosts` file, which maps IP addresses to hostnames. The output is as follows:

```
127.0.0.1    localhost
127.0.1.1    vm1
10.0.2.15    salt

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

"hosts" [только для чтения] 8L, 198C
```

The terminal window includes a menu bar with "Файл", "Правка", "Настройка", and "Справка". The status bar at the bottom shows the file name "salt - Менеджер фа...", the terminal title "etc:vi — Терминал...", and the system clock "19:39 ПТ, 28 АПР".

Настройка ключей и проверка Master-Minion соединения

- На vm2 проверим появившиеся ключи `salt-key -F master`
- Принимаем ключ vm1 `salt-key --accept='vm1'`
- Проверим ключи `salt-key -L`
- Проверим связь между Мастером и Миньоном, выполнив команду `test.ping`:

```
vm2 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

~: bash — Терминал Fly
Файл  Правка  Настройка  Справка
1

astra@vm2:~$ salt-key -F master
CRITICAL: Salt configured to run as user "root" but unable to switch.
astra@vm2:~$ sudo salt-key -F master
[sudo] пароль гня astra:
Local Keys:
master.pem: 2a:57:84:2d:a5:d2:ed:cb:88:cf:7d:56:d1:19:0f:a1:69:fe:5d:8c:4
4:4b:4a:f3:ea:35:34:5f:0a:e6:b5:d2
master.pub: 13:1c:e6:95:7a:cc:95:ca:d8:8d:48:27:81:cd:7c:1c:a0:60:40:c5:d
f:87:75:bb:e5:46:9d:ef:1d:35:0a:47
Unaccepted Keys:
vm1: 8a:a5:d6:8c:f2:e3:de:15:fc:69:c8:2f:b8:92:7e:3e:87:1f:99:9c:a3:eb:95
:74:16:c2:a4:67:c5:46:c6:66
astra@vm2:~$ salt-key --accept='vm1'
CRITICAL: Salt configured to run as user "root" but unable to switch.
astra@vm2:~$ sudo salt-key --accept='vm1'
The following keys are going to be accepted:
Unaccepted Keys:
vm1
Proceed? [n/Y] Y
Key for minion vm1 accepted.
astra@vm2:~$
```

```
astra@vm2:~$ sudo salt-key -L
Accepted Keys:
vm1
Denied Keys:
Unaccepted Keys:
Rejected Keys:
astra@vm2:~$
```

```
astra@vm2:~$ sudo salt vm1 test.ping
vm1:
True
astra@vm2:~$
```


Установка пакетов nginx на vm2 и подготовка к написанию скриптов

- Установим nginx командой `sudo apt-get install nginx`
- Переименуем нужный нам файл с длинным названием в `nginx.deb`
- Закинем его `/srv/salt` (будущие скрипты тоже будут лежать тут)

Теперь все готово к написанию
и запуску скриптов

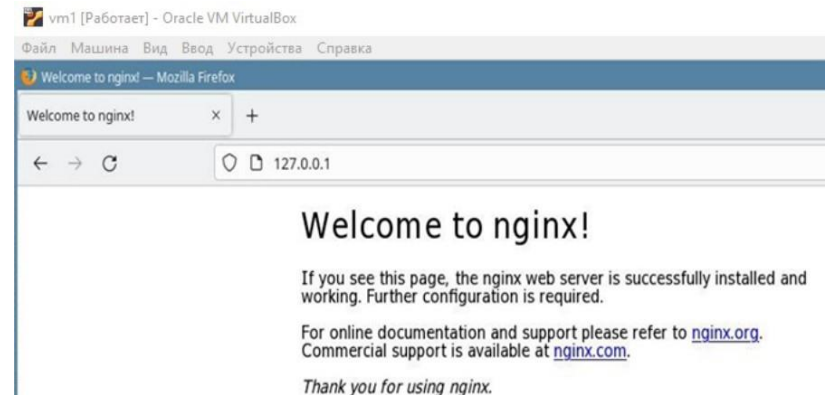
```
astra@vm2:/var/cache/apt/archives$ sudo mv nginx-full_1.14.1-1~bpo9+1+ci20
2106041722+astra1_amd64.deb nginx.deb
[sudo] пароль для astra:
astra@vm2:/var/cache/apt/archives$ ls
libnginx-mod-http-auth-pam_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64.deb
libnginx-mod-http-dav-ext_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64.deb
libnginx-mod-http-echo_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64.deb
libnginx-mod-http-geoip_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64.deb
libnginx-mod-http-image-filter_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64
.deb
libnginx-mod-http-subst-filter_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64.
deb
libnginx-mod-http-upstream-fair_1.14.1-1~bpo9+1+ci202106041722+astra1_amd6
4.deb
libnginx-mod-http-xslt-filter_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64.
deb
libnginx-mod-mail_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64.deb
libnginx-mod-stream_1.14.1-1~bpo9+1+ci202106041722+astra1_amd64.deb
lock
nginx_1.14.1-1~bpo9+1+ci202106041722+astra1_all.deb
nginx-common_1.14.1-1~bpo9+1+ci202106041722+astra1_all.deb
nginx.deb
partial
astra@vm2:/var/cache/apt/archives$ █
```


Создание и запуск скрипта nginx_deploy.sls

- Создаем файл скрипта с расширением .sls и прописываем в нем следующие действия:
1)перенести файл nginx.deb, 2)установить его 3)запустить
- Выглядеть он будет как на скриншоте
- Запускаем nginx_deploy.sls командой `sudo salt '*' state.apply nginx_deploy`
- Проверяем отработку в терминале и смотрим запустился ли nginx на localhost-е vm1

```
1 nginx.deb:
2   file.managed:
3     - name: /tmp/nginx.deb
4     - user: root
5     - group: root
6     - replace: True
7     - makedirs: True
8     - source: salt://nginx.deb
9   cmd.run:
10    - name: apt install /tmp/nginx.deb -y
11  service.running:
12    - name: nginx
13    - enable: True
14    - reload: True
```

```
Summary for vm1
-----
Succeeded: 3 (changed=2)
Failed:    0
-----
Total states run:    3
Total run time:   24.374 s
astra@vm1:~$
```



Создание и запуск скрипта nginx_update_index.sls

- Создаем файл скрипта с расширением .sls и прописываем в нем изменение содержимого стандартной страницы nginx на свою кастомную «hello greenatom»
- Выглядеть он будет как на скриншоте
- Запускаем nginx_update_index.sls командой `sudo salt '*' state.apply nginx_update_index`
- Проверяем отработку в терминале и смотрим изменилась ли страница на vm1

```
1 index.nginx-debian.html:
2   file.managed:
3     - name: /var/www/html/index.nginx-debian.html
4     - user: root
5     - group: root
6   cmd.run:
7     - name: rm /var/www/html/index.nginx-debian.html
8     - name: echo "Hello Greenatom" > /var/www/html/index.nginx-debian.html
9
```

```
Summary for vm1
-----
Succeeded: 2 (changed=1)
Failed:    0
-----
Total states run:    2
Total run time:   77.499 ms
astra@vm1:~$
```

