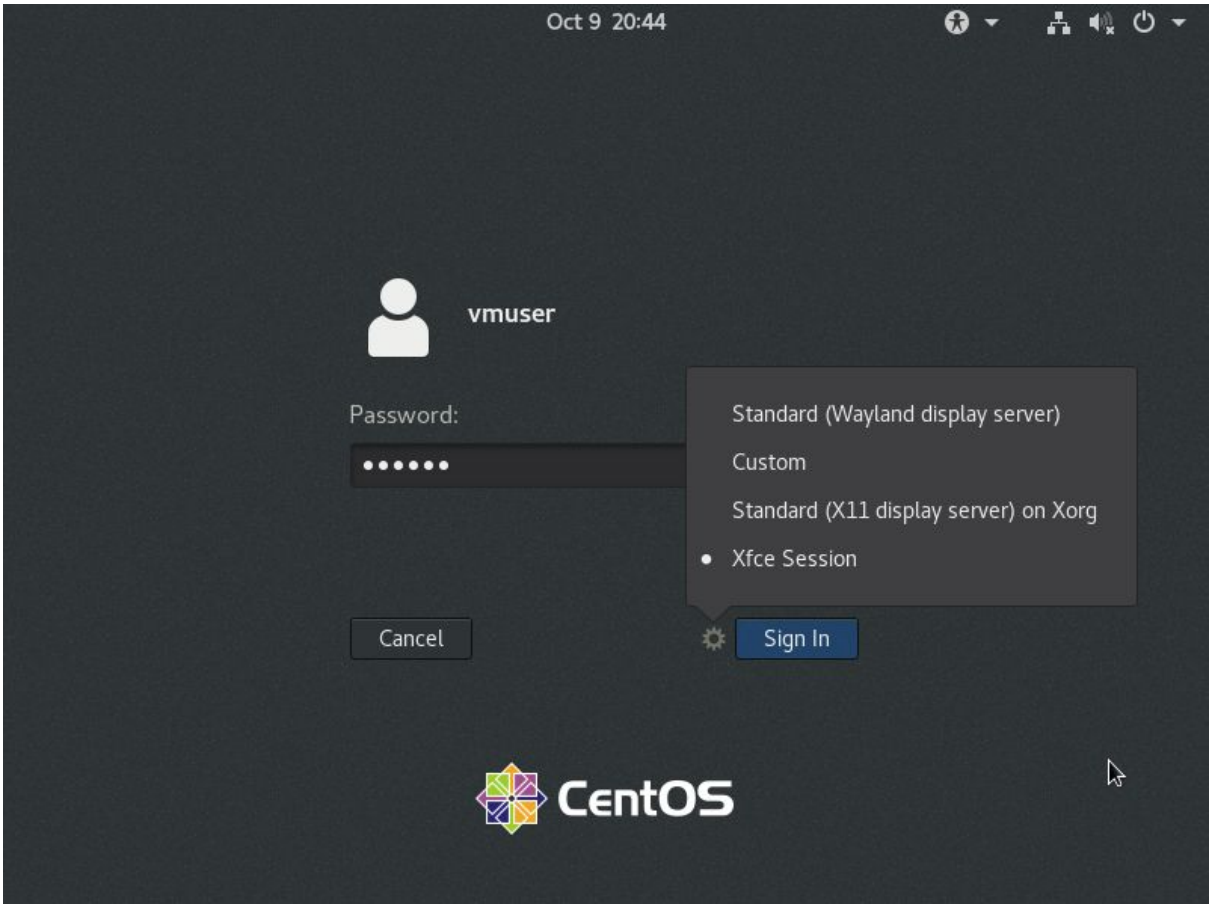
Часть A-C

Хост	Адрес	Имя

Установка virt-manager.

Дано - хост Сентос8 минимал, доступ к репозиторию, установлен vim
Установим графическую среду - это существенно упростит работу с ВМ.
dnf group install “Xfce”
systemctl set-default graphical.target. Проверяем, при логине выбираем Xfce.



Проверяем виртуализацию. Работает в графическом терминале

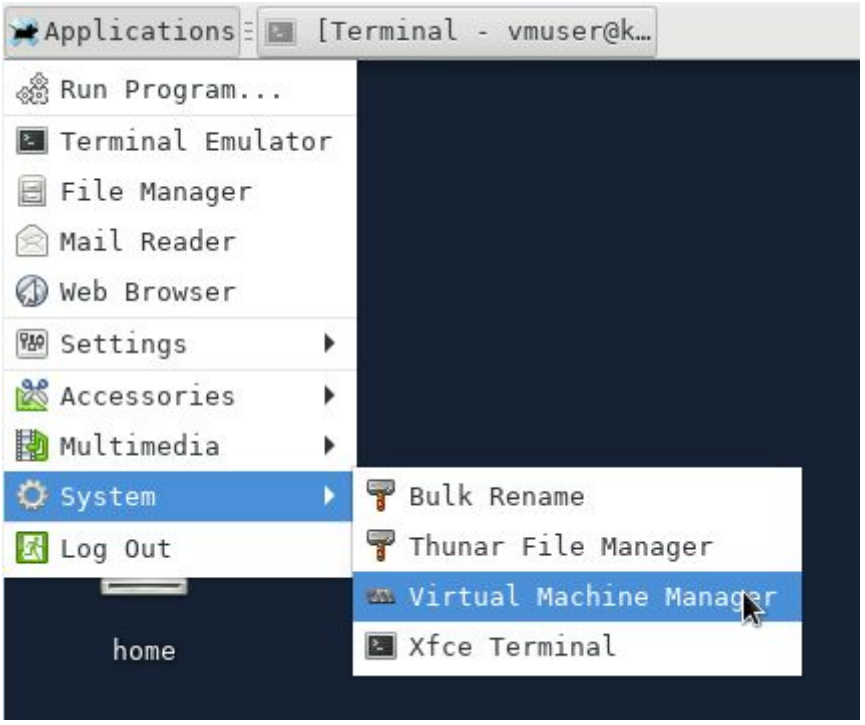
```
[root@kvm-host vmuser]# cat /proc/cpuinfo  | grep vmx
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
nstant_tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq vmx ssse3 cx16 p
ave avx rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single pti tpr_shadow flexpriority fsg
r flush_l1d
```

Устанавливаем virt-manager и libvirt и qemu-kvm

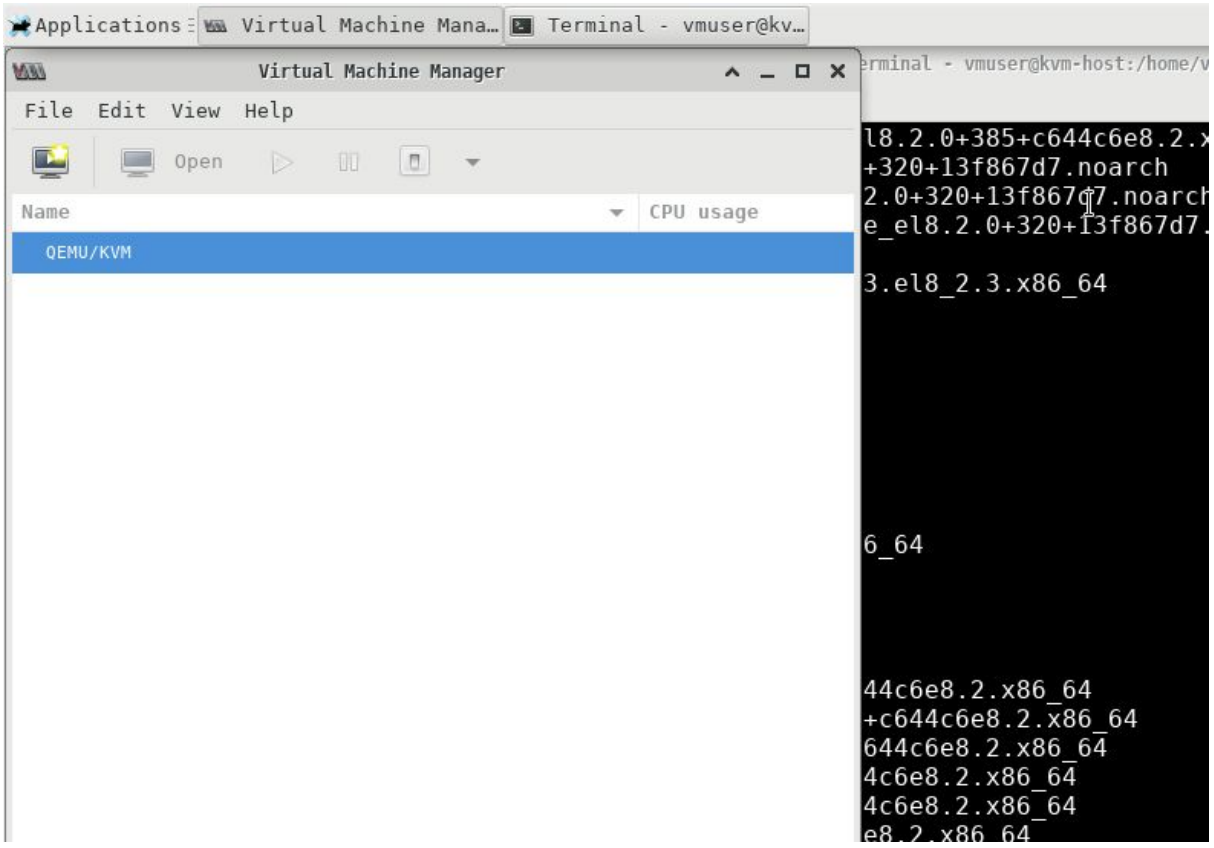
```
Installed:
  virt-manager-2.2.1-3.el8.noarch                                  celt051-0.5.1.3-15.el8.x86_64
  genisoimage-1.1.11-39.el8.x86_64                                 gtk-vnc2-0.9.0-2.el8.x86_64
  gvfs-1.36.2-8.el8.x86_64                                         gvfs-client-1.36.2-8.el8.x86_64
  gvnc-0.9.0-2.el8.x86_64                                          libatasmart-0.19-14.el8.x86_64
  libblockdev-2.19-12.el8.x86_64                                   libblockdev-crypto-2.19-12.el8.x86_64
  libblockdev-fs-2.19-12.el8.x86_64                                libblockdev-loop-2.19-12.el8.x86_64
  libblockdev-mdraid-2.19-12.el8.x86_64                            libblockdev-part-2.19-12.el8.x86_64
  libblockdev-swap-2.19-12.el8.x86_64                              libblockdev-utils-2.19-12.el8.x86_64
  libbytesize-1.4-3.el8.x86_64                                     libcacard-3:2.7.0-2.el8_1.x86_64
  libcdio-2.0.0-3.el8.x86_64                                       libcdio-paranoia-10.2+0.94+2-3.el8.x8
  libosinfo-1.5.0-3.el8.x86_64                                     libudisks2-2.8.3-2.el8.x86_64
  libusal-1.1.11-39.el8.x86_64                                     libvirt-glib-2.0.0-1.el8.x86_64
  libvirt-libs-4.5.0-42.module_el8.2.0+320+13f867d7.x86_64         nmap-ncat-2:7.70-5.el8.x86_64
  osinfo-db-20200203-1.el8.noarch                                  osinfo-db-tools-1.5.0-4.el8.x86_64
  python3-argcomplete-1.9.3-6.el8.noarch                           python3-libvirt-4.5.0-2.module_el8.2
  spice-glib-0.37-1.el8.x86_64                                     spice-gtk3-0.37-1.el8.x86_64
  udisks2-2.8.3-2.el8.x86_64                                       usbredir-0.8.0-1.el8.x86_64
  virt-manager-common-2.2.1-3.el8.noarch                           volume_key-libs-0.3.11-5.el8.x86_64
  yajl-2.1.0-10.el8.x86_64                                         cyrus-sasl-2.1.27-1.el8.x86_64
  cyrus-sasl-gssapi-2.1.27-1.el8.x86_64                            dosfstools-4.1-6.el8.x86_64
  gdisk-1.0.3-6.el8.x86_64                                         mdadm-4.1-13.el8.x86_64

Complete!
[root@kvm-host vmuser]#
```

Может потребоваться перезагрузка libvirt
systemctl restart libvirt

Проверяем наличие



Вводим пароль рута при запросе



OVS - Создание коммутаторов

Установка не очень понятно как.
Считаем, что поставлен.
systemctl enable --now openvswitch - запустим службу
Настроим HQ.

1) Создадим коммутатор
2) Передадим физический порт vmnic1 как транк (уходит в g0/1) (тут называется enp0s3)
3) Создадим 5+1 порт с указанием VLAN (порты типа Internal, виртуальные)

ВАЖНО - в скриншоте ниже отсутствует vlan103. необходимый для телефонии. Он добавляется в оба транка. Так же добавить BGP vlan 120. Аналогично, в списки обоих транков.

```
[root@kvm-host vmuser]# ovs-vsctl add-br HQ-OVS
[root@kvm-host vmuser]# ovs-vsctl add-port HQ-OVS  enp0s3 trunks=101,111,112,113,300
[root@kvm-host vmuser]# ovs-vsctl add-port HQ-OVS  vlan101 tag=101 -- set interface vlan101 type=internal
[root@kvm-host vmuser]# ovs-vsctl add-port HQ-OVS  vlan111 tag=111 -- set interface vlan111 type=internal
[root@kvm-host vmuser]# ovs-vsctl add-port HQ-OVS  vlan112 tag=112 -- set interface vlan112 type=internal
[root@kvm-host vmuser]# ovs-vsctl add-port HQ-OVS  vlan113 tag=113 -- set interface vlan113 type=internal
[root@kvm-host vmuser]# ovs-vsctl add-port HQ-OVS  vlan300 tag=300 -- set interface vlan300 type=internal
[root@kvm-host vmuser]# ovs-vsctl show
f0805524-ae3d-4bf2-9165-dd3529c77d44
    Bridge HQ-OVS
        Port "vlan111"
            tag: 111
            Interface "vlan111"
                type: internal
        Port "enp0s3"
            trunks: [101, 111, 112, 113, 300]
            Interface "enp0s3"
        Port "vlan300"
            tag: 300
            Interface "vlan300"
                type: internal
        Port HQ-OVS
            Interface HQ-OVS
                type: internal
        Port "vlan101"
            tag: 101
            Interface "vlan101"
                type: internal
        Port "vlan113"
            tag: 113
            Interface "vlan113"
                type: internal
        Port "vlan112"
            tag: 112
            Interface "vlan112"
                type: internal
    ovs_version: "2.12.0"
[root@kvm-host vmuser]#
```

Дополним еще одним транком уже для LinRTR

```
[root@kvm-host vmuser]# ovs-vsctl add-port HQ-OVS hq-trunk trunks=101,111,112,113 -- set interface hq-trunk type=internal
[root@kvm-host vmuser]#
```

По аналогии настроим RMACC-OVS и BR-OVS

В RMACC-OVS заходит vmnic2, (enp0s8 тут)

```
[root@kvm-host vmuser]# ovs-vsctl add-br RMACC-OVS
[root@kvm-host vmuser]# ovs-vsctl add-port RMACC-OVS enp0s8
[root@kvm-host vmuser]# ovs-vsctl add-port RMACC-OVS port0 -- set interface port0 type=internal
```

В BR-OVS заходят vmnic3 и vmnic4, (здесь enp0s9 и enp0s10)
Реализует бонд на базе этих двух интерфейсов. Создадим виртуальные интерфейсы
Утверждается, что название бонда должно совпадать с названием на Cisco. На всякий случай поверим. Создается бонд Po5 на базе двух интерфейсов с включенным lacp.

```
[root@kvm-host vmuser]# ovs-vsctl add-br BR-OVS
[root@kvm-host vmuser]# ovs-vsctl add-bond BR-OVS Po5 enp0s9 enp0s10 lacp=active
[root@kvm-host vmuser]# ovs-vsctl add-port BR-OVS br-vlan101 tag=101 -- set interface br-vlan101 type=internal
[root@kvm-host vmuser]# ovs-vsctl add-port BR-OVS br-vlan102 tag=102 -- set interface br-vlan102 type=internal
[root@kvm-host vmuser]# ovs-vsctl add-port BR-OVS br-vlan103 tag=103 -- set interface br-vlan103 type=internal
[root@kvm-host vmuser]#
```

Проверка работы lacp на KVM-CHECKER. (там аналогичная конфигурация).

```
[root@KVM-CHECKER ~]# ovs-appctl bond/show Po5
---- Po5 ----
bond_mode: active-backup
bond may use recirculation: no, Recirc-ID : -1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
lacp_status: negotiated
lacp_fallback_ab: false
active slave mac: 08:00:27:ae:37:96(enp0s8)

slave enp0s17: enabled
  may_enable: true

slave enp0s8: enabled
  active slave
  may_enable: true

[root@KVM-CHECKER ~]# _
```

Проверка LACP на хосте.

```
[root@kvm-host vmuser]# ovs-appctl bond/show Po5
---- Po5 ----
bond_mode: active-backup
bond may use recirculation: no, Recirc-ID : -1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
lacp_status: negotiated
lacp_fallback_ab: false
active slave mac: 08:00:27:cd:85:9f(enp0s10)

slave enp0s10: enabled
  active slave
  may_enable: true

slave enp0s9: enabled
  may_enable: true

[root@kvm-host vmuser]#
```

Превратим bond в транк.

```
[root@kvm-host vmuser]# ovs-vsctl set port Po5 trunk=101,102,103
```

Последняя проверка

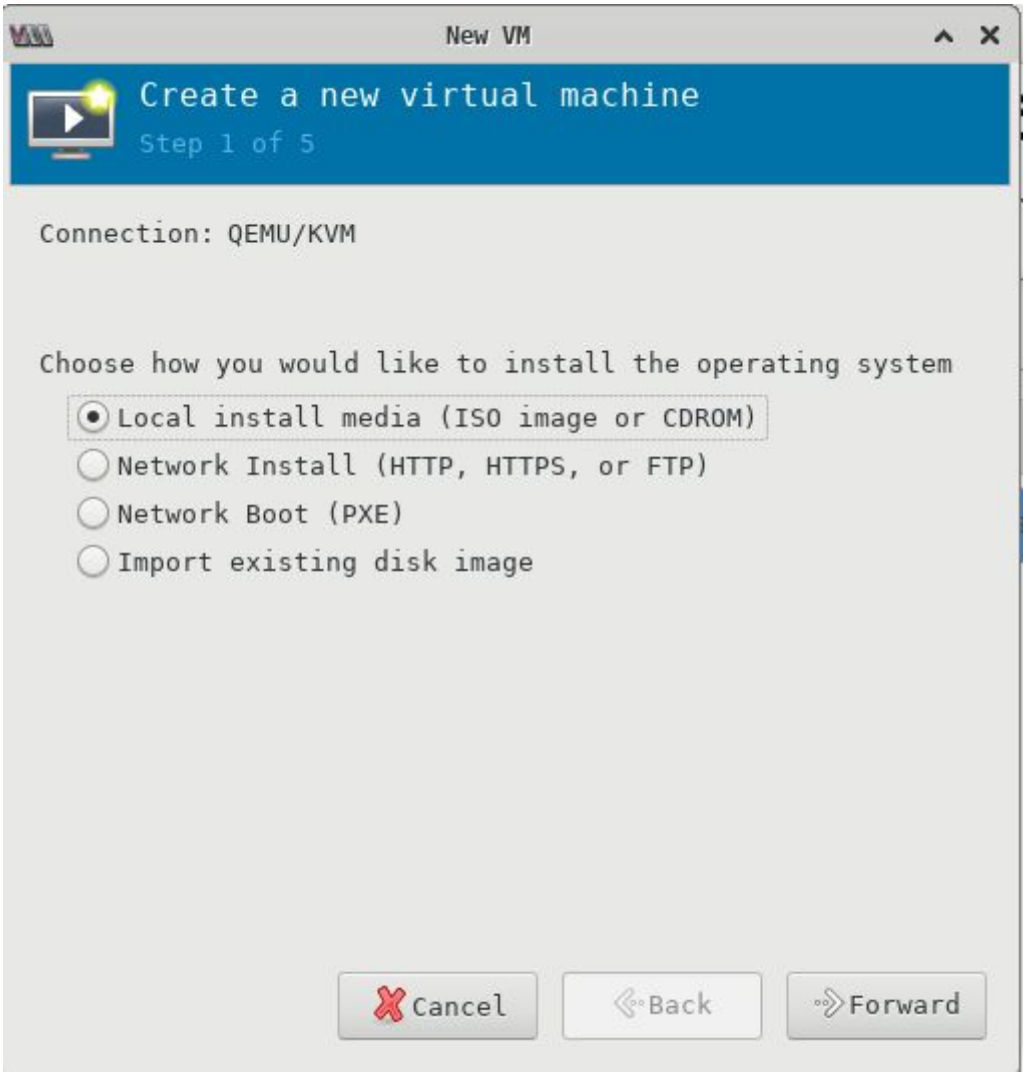
```
Bridge RMACC-OVS
    Port "port0"
        Interface "port0"
            type: internal
    Port "enp0s8"
        Interface "enp0s8"
    Port RMACC-OVS
        Interface RMACC-OVS
            type: internal
```

```
Bridge HQ-OVS
    Port "vlan111"
        tag: 111
        Interface "vlan111"
            type: internal
    Port "enp0s3"
        trunks: [101, 111, 112, 113, 300]
        Interface "enp0s3"
    Port "vlan300"
        tag: 300
        Interface "vlan300"
            type: internal
    Port HQ-OVS
        Interface HQ-OVS
            type: internal
    Port "vlan101"
        tag: 101
        Interface "vlan101"
            type: internal
    Port "vlan113"
        tag: 113
        Interface "vlan113"
            type: internal
    Port "vlan112"
        tag: 112
        Interface "vlan112"
            type: internal
```

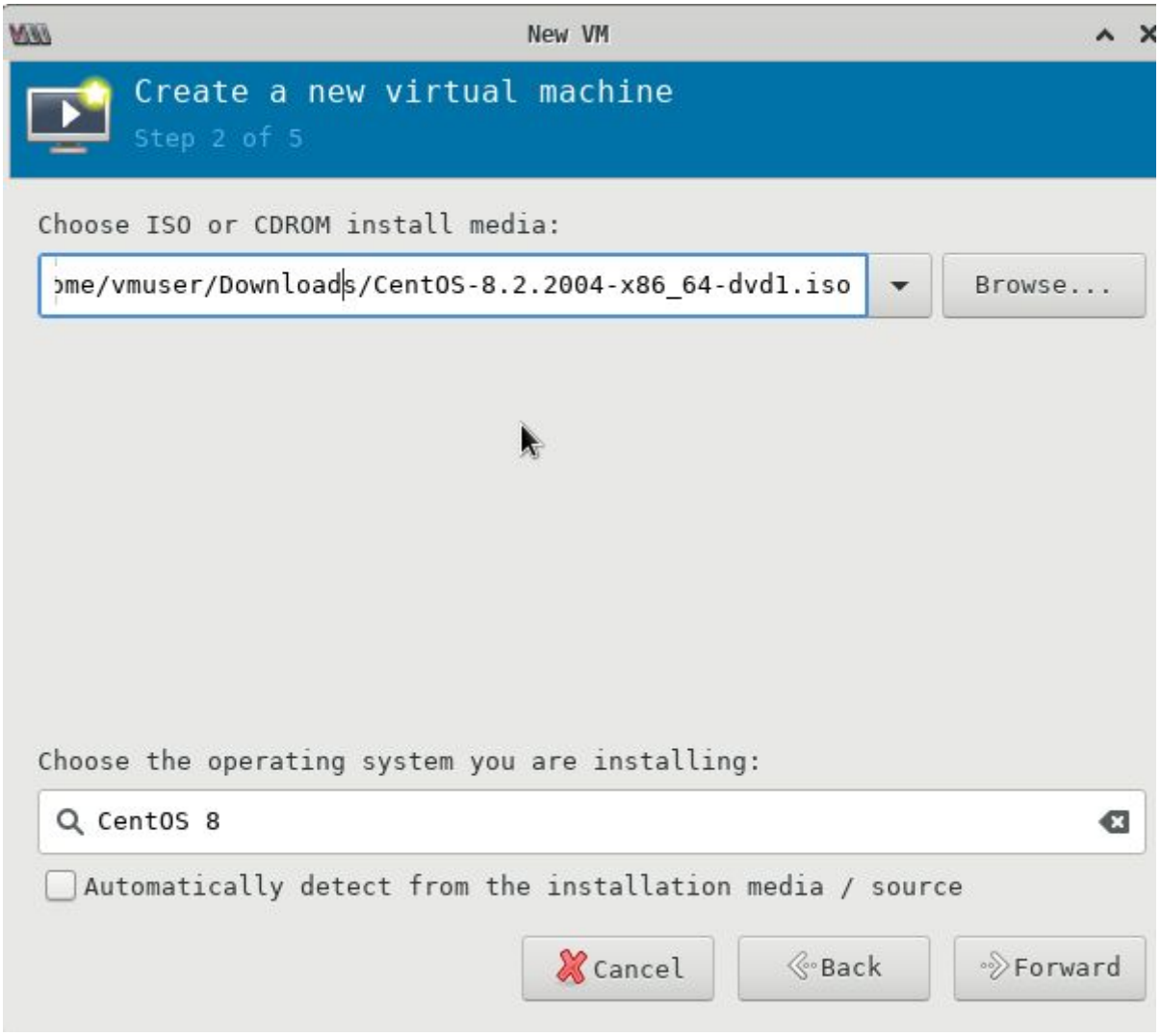
```
Bridge BR-OVS
    Port "br-vlan103"
        tag: 103
        Interface "br-vlan103"
            type: internal
    Port "br-vlan102"
        tag: 102
        Interface "br-vlan102"
            type: internal
    Port "Po5"
        trunks: [101, 102, 103]
        Interface "enp0s10"
        Interface "enp0s9"
    Port BR-OVS
        Interface BR-OVS
            type: internal
    Port "br-vlan101"
        tag: 101
        Interface "br-vlan101"
            type: internal
ovs_version: "2.12.0"
```

Создание виртуальных машин

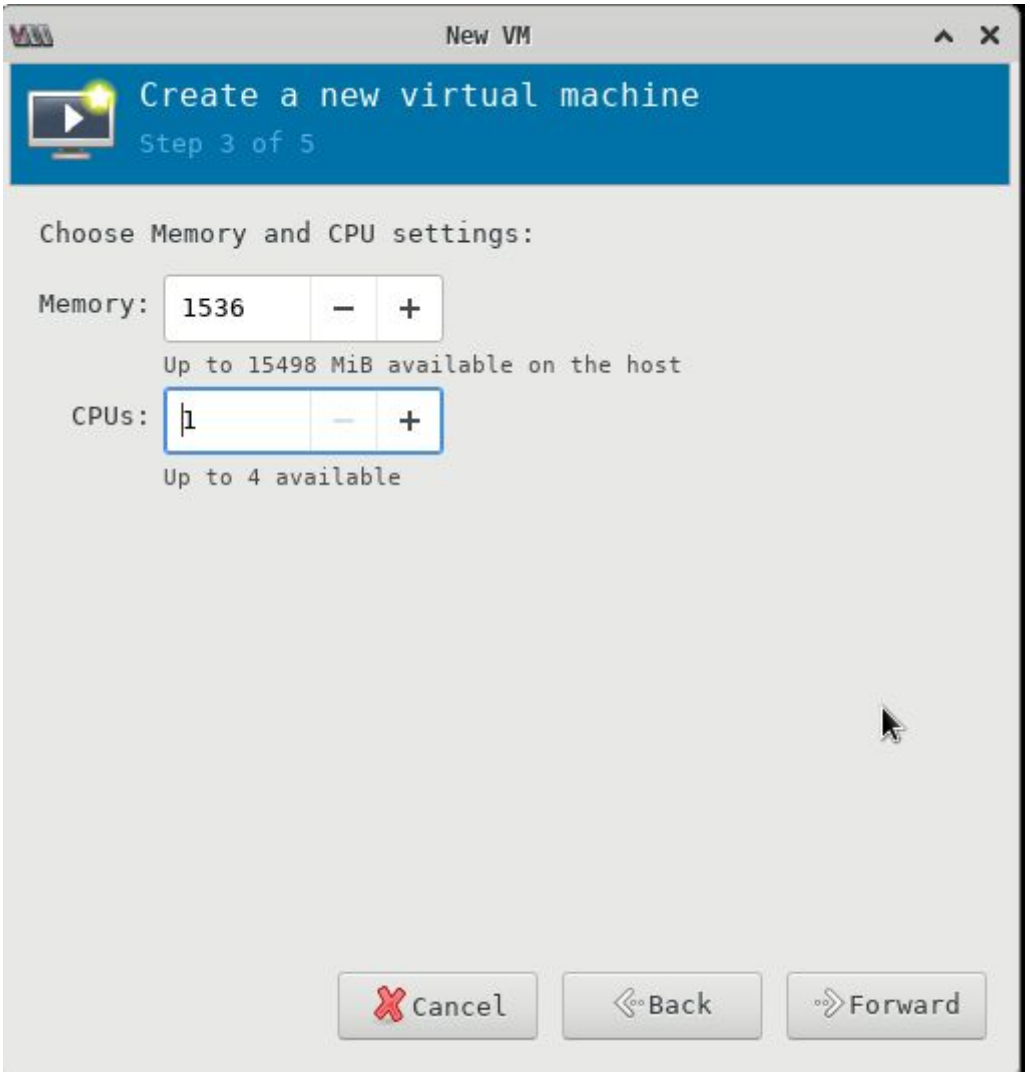
Нажимаем “Создать ВМ” в Virt-Manager. В данном примере считаем, что у нас есть локальный доступ к ISO.



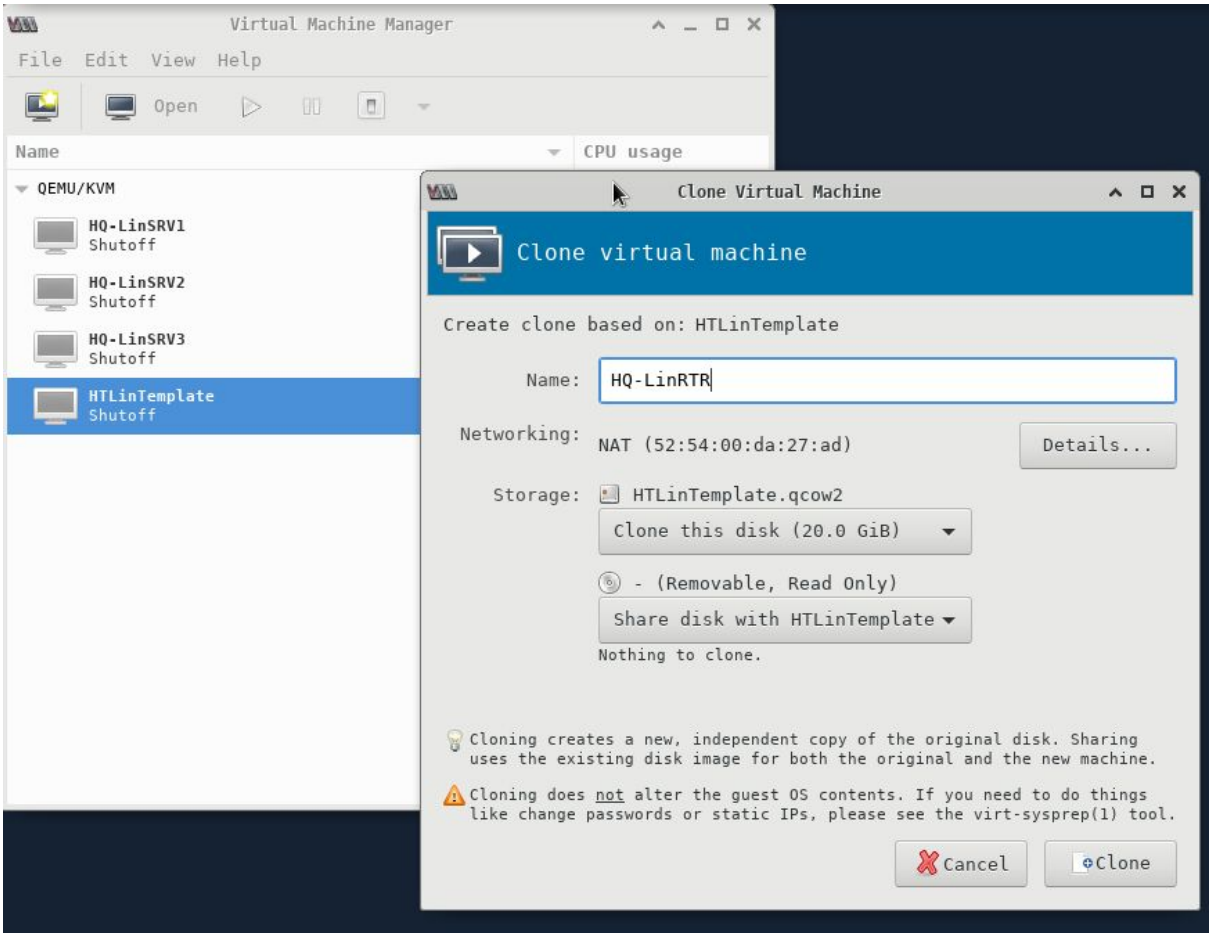
При выборе ISO нажать Browse, затем Browse Local. Тип выбираем вручную - вводим CentOS 8.

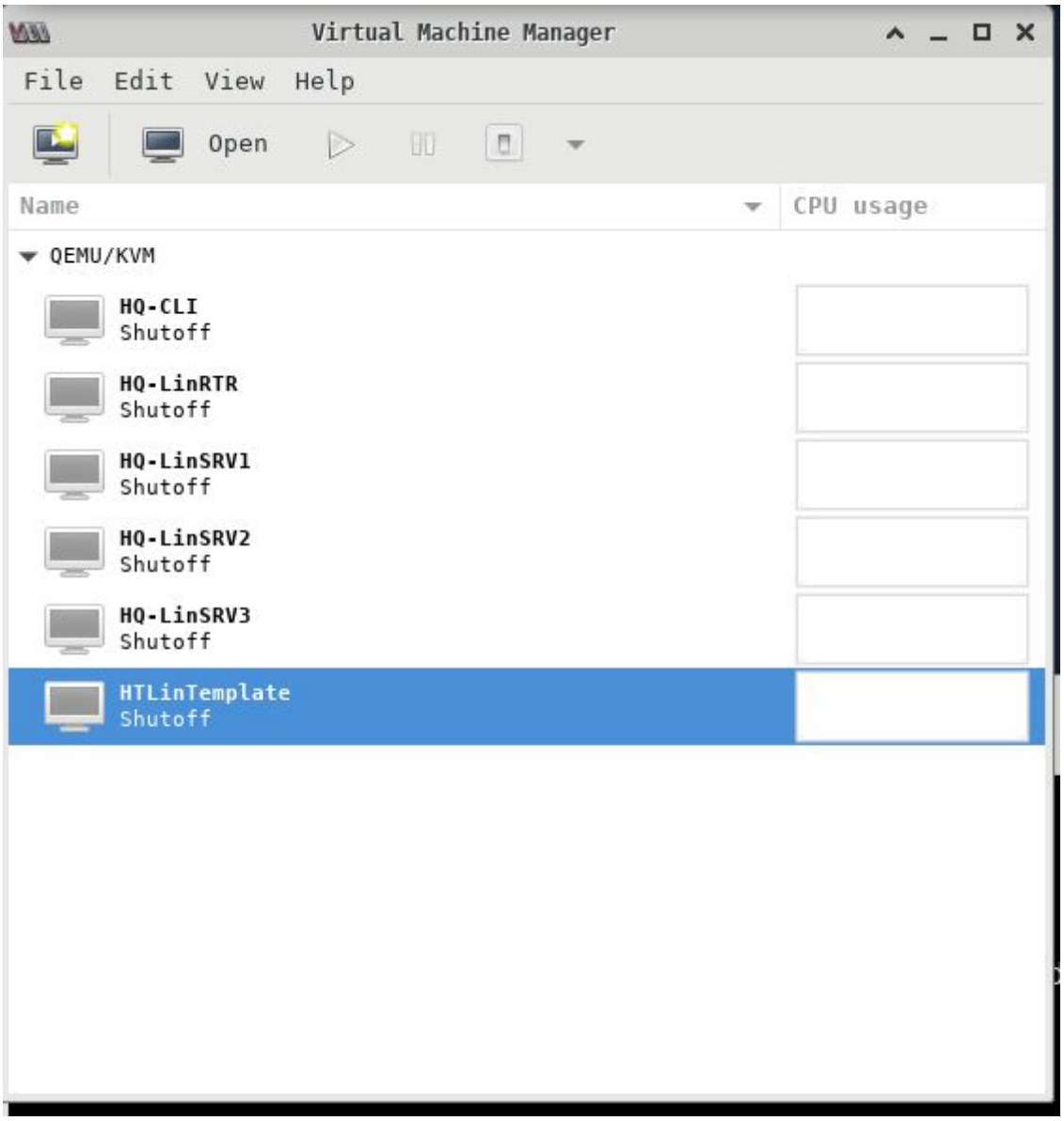


Задаем ОЗУ и процессоры - 1536 и 2 ядра.



На шаге 5 установим имя HTLinTemplate. Выбирать сетевой интерфейс смысла нет - мы будем клонировать.
Выполним установку ОС. Выбираем минимальный вариант.
Клонируем полученную систему


Virtual Machine Manager
File Edit View Help
Open
Name
CPU usage
QEMU/KVM
HQ-LinSRV1
Shutoff
HQ-LinSRV2
Shutoff
HQ-LinSRV3
Shutoff
HTLinTemplate
Shutoff
Clone Virtual Machine
Clone virtual machine
Create clone based on: HTLinTemplate
Name:
HQ-LinRTR
Networking:
NAT (52:54:00:da:27:ad)
Details...
Storage:
HTLinTemplate.qcow2
Clone this disk (20.0 GiB)
- (Removable, Read Only)
Share disk with HTLinTemplate
Nothing to clone.
Cloning creates a new, independent copy of the original disk. Sharing uses the existing disk image for both the original and the new machine.
Cloning does not alter the guest OS contents. If you need to do things like change passwords or static IPs, please see the virt-sysprep(1) tool.
Cancel
Clone



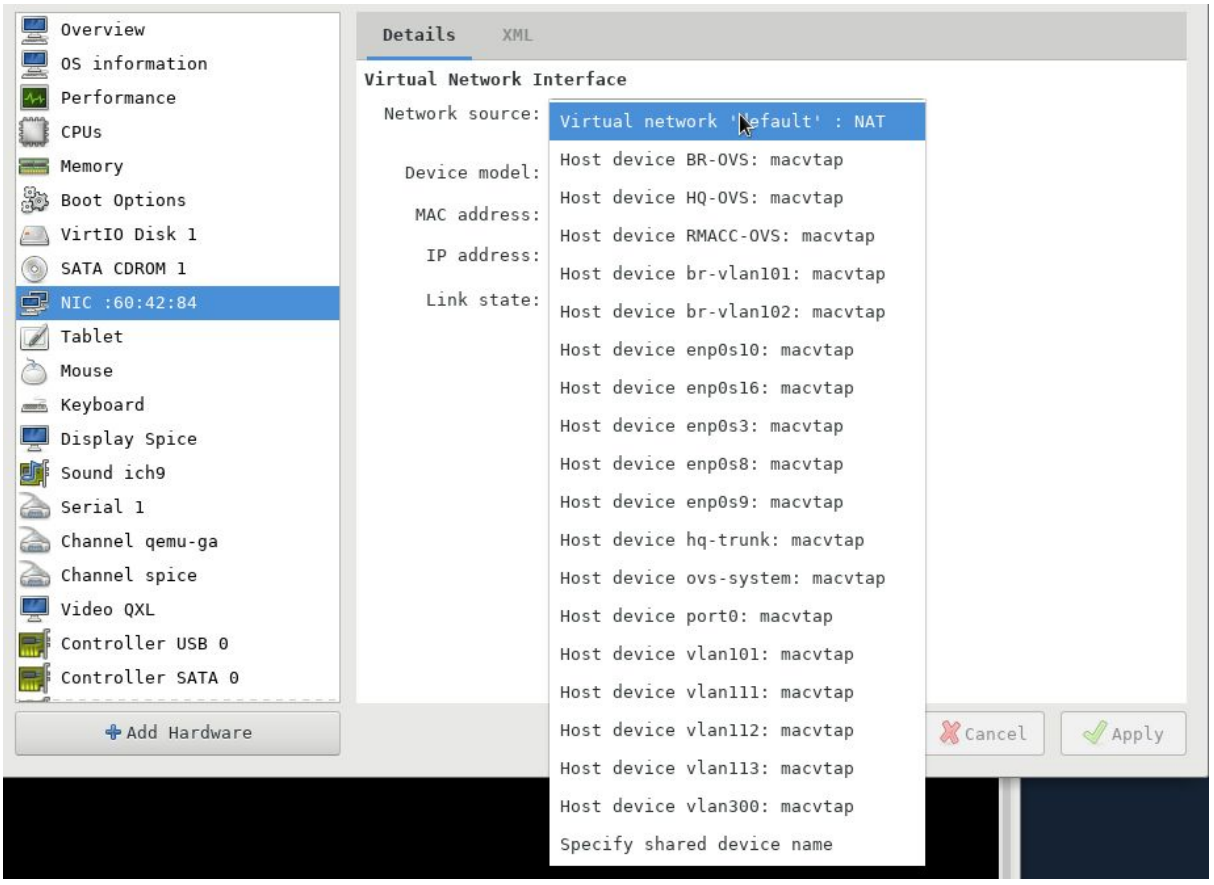
ПРОВЕРОЧНОЕ: поскольку при написании инструкции не было полноценного стенда - создается специальная ВМ KVM-Checker, которая будет подключаться к различным NIC KVM-HOST и с помощью tcpdump показывать, что конфигурация на стороне KVM верная.
CHECKER имеет 4 интерфейса, подключенные к 4 интерфейсам KVM-HOST.

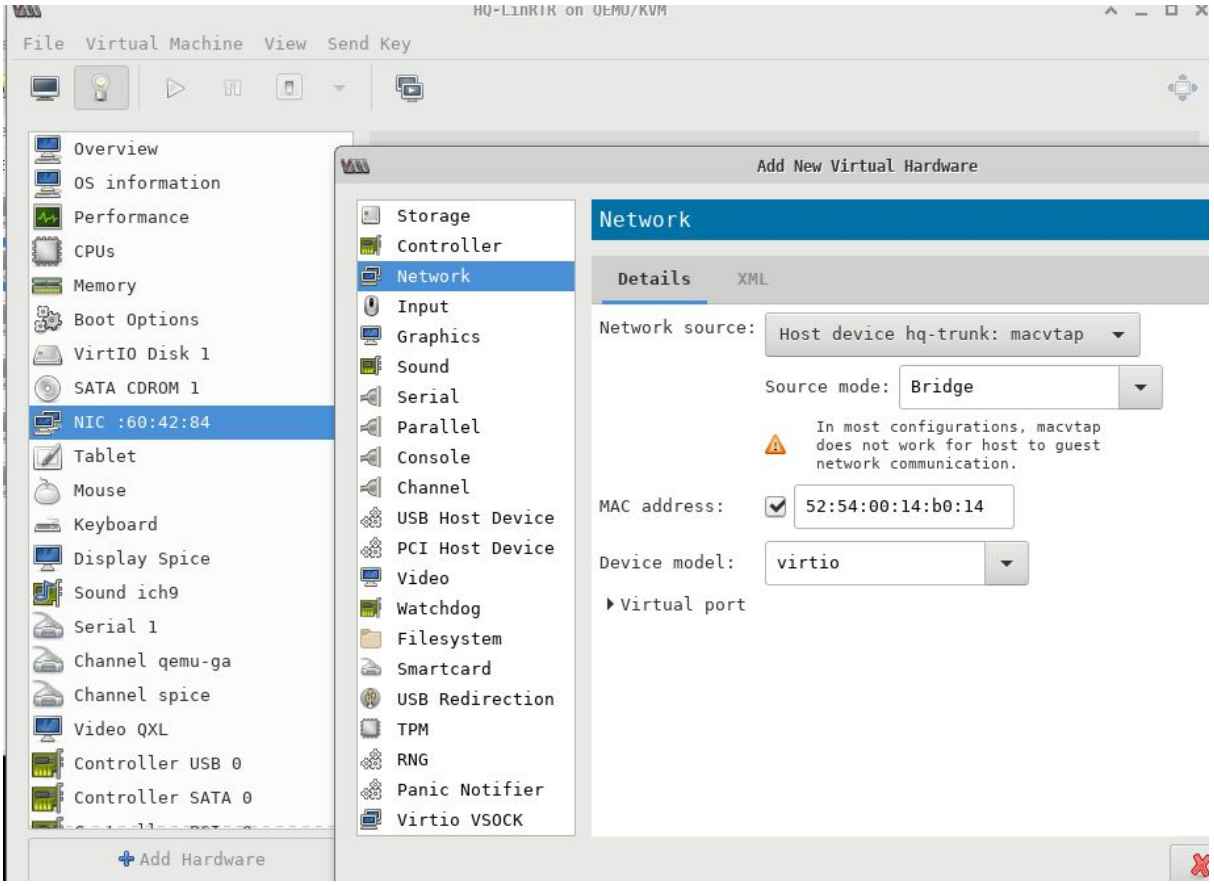
Настройка ВМ HQ-LinRTR

Согласно заданию:
два интерфейса

- влан 300
- транковый интерфейс на прочие вланы внутри HQ-OVS.

Сконфигурируем сеть. Существующему интерфейсу присвоим vlan300, добавим еще один со значением hq-trunk.


Overview
OS information
Performance
CPUs
Memory
Boot Options
VirtIO Disk 1
SATA CDROM 1
NIC :60:42:84
Tablet
Mouse
Keyboard
Display Spice
Sound ich9
Serial 1
Channel qemu-ga
Channel spice
Video QXL
Controller USB 0
Controller SATA 0
Add Hardware
Details
XML
Virtual Network Interface
Network source:
Device model:
MAC address:
IP address:
Link state:
Virtual network 'default' : NAT
Host device BR-OVS: macvtap
Host device HQ-OVS: macvtap
Host device RMACC-OVS: macvtap
Host device br-vlan101: macvtap
Host device br-vlan102: macvtap
Host device enp0s10: macvtap
Host device enp0s16: macvtap
Host device enp0s3: macvtap
Host device enp0s8: macvtap
Host device enp0s9: macvtap
Host device hq-trunk: macvtap
Host device ovs-system: macvtap
Host device port0: macvtap
Host device vlan101: macvtap
Host device vlan111: macvtap
Host device vlan112: macvtap
Host device vlan113: macvtap
Host device vlan300: macvtap
Specify shared device name
Cancel
Apply


HQ-LinRTR on QEMU/KVM
File
Virtual Machine
View
Send Key
Overview
OS information
Performance
CPUs
Memory
Boot Options
VirtIO Disk 1
SATA CDROM 1
NIC :60:42:84
Tablet
Mouse
Keyboard
Display Spice
Sound ich9
Serial 1
Channel qemu-ga
Channel spice
Video QXL
Controller USB 0
Controller SATA 0
Add Hardware
Add New Virtual Hardware
Storage
Controller
Network
Input
Graphics
Sound
Serial
Parallel
Console
Channel
USB Host Device
PCI Host Device
Video
Watchdog
Filesystem
Smartcard
USB Redirection
TPM
RNG
Panic Notifier
Virtio VSOCK
Network
Details
XML
Network source:
Host device hq-trunk: macvtap
Source mode:
Bridge
In most configurations, macvtap does not work for host to guest network communication.
MAC address:
52:54:00:14:b0:14
Device model:
virtio
Virtual port

Сконфигурируем интерфейс vlan300, в данном примере - enp1s0.
(Чекер подключается к интерфейсу vmnic1, должен увидеть тегированный трафик с тегом 300)
Используем nmcli (можно любой другой метод)

```
[root@localhost ~]# nmcli con edit
Valid connection types: 6lowpan, 802-11-olpc-mesh (olpc-mesh), 802-11-wireless
h, bond, bridge, cdma, dummy, generic, gsm, infiniband, ip-tunnel, macsec, macv
team, tun, vlan, vpn, vxlan, wifi-p2p, wimax, wireguard, wpan, bond-slave, bri
Enter connection type: ethernet

===| nmcli interactive connection editor |===

Adding a new '802-3-ethernet' connection

Type 'help' or '?' for available commands.
Type 'print' to show all the connection properties.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet), 802
proxy
nmcli> set connection.id  VLAN300
nmcli> set connection.autoconnect yes
nmcli> set connection.interface-name  enp1s0
nmcli> set ipv4.addresses  145.46.48.130/30
nmcli> set ipv4.gateway 145.46.48.129
nmcli> set ipv4.method manual
nmcli> save
Saving the connection with 'autoconnect=yes'. That might result in an immediate
Do you still want to save? (yes/no) [yes]
Connection 'VLAN300' (5c573bbe-2815-4273-bbb7-688f61213940) successfully saved
nmcli> quit
```

Запустим пинг до ISP2. Проверим KVM-Checker

```
[root@localhost ~]# tcpdump -nn -i enp0s17 -e vlan
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s17, link-type EN10MB (Ethernet), capture size 262144 bytes
14:12:51.294551 52:54:00:60:42:84 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 30
0, p 0, ethertype ARP, Request who-has 145.46.48.129 tell 145.46.48.130, length 46
14:12:52.318297 52:54:00:60:42:84 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 30
```

Работа HQ-OVS на транк в сторону HQ-SW1 подтверждена. 300 VLAN отрабатывает

Настроим транковый интерфейс на HQ-LinRTR. использует nmcli, возможны любые другие методы.

```
[root@HQ-LinRTR ~]# nmcli con edit
Valid connection types: 6lowpan, 802-11-olpc-mesh (olpc-mesh), 802-11-wireless (wifi), 802-3-ethern
h, bond, bridge, cdma, dummy, generic, gsm, infiniband, ip-tunnel, macsec, macvlan, ovs-bridge, ovs
team, tun, vlan, vpn, vxlan, wifi-p2p, wimax, wireguard, wpan, bond-slave, bridge-slave, team-slave
Enter connection type: vlan

===| nmcli interactive connection editor |===

Adding a new 'vlan' connection

Type 'help' or '?' for available commands.
Type 'print' to show all the connection properties.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, vlan, 802-3-ethernet (ethernet), ethtool, match, i
nmcli> set connection.id VLAN101
nmcli> set vlan.id  101
nmcli> set vlan.parent enp7s0
nmcli> set ipv4.addresses 10.0.4.3/29
nmcli> set ipv4.method manual
nmcli> save
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation of the c
Do you still want to save? (yes/no) [yes]
Connection 'VLAN101' (f0f6a3e8-8f3d-4297-a210-9c1f75c326ec) successfully saved.
nmcli> quit
[root@HQ-LinRTR ~]#
```

Запускаем пинг до адреса 10.0.4.6 - наблюдаем чекер.

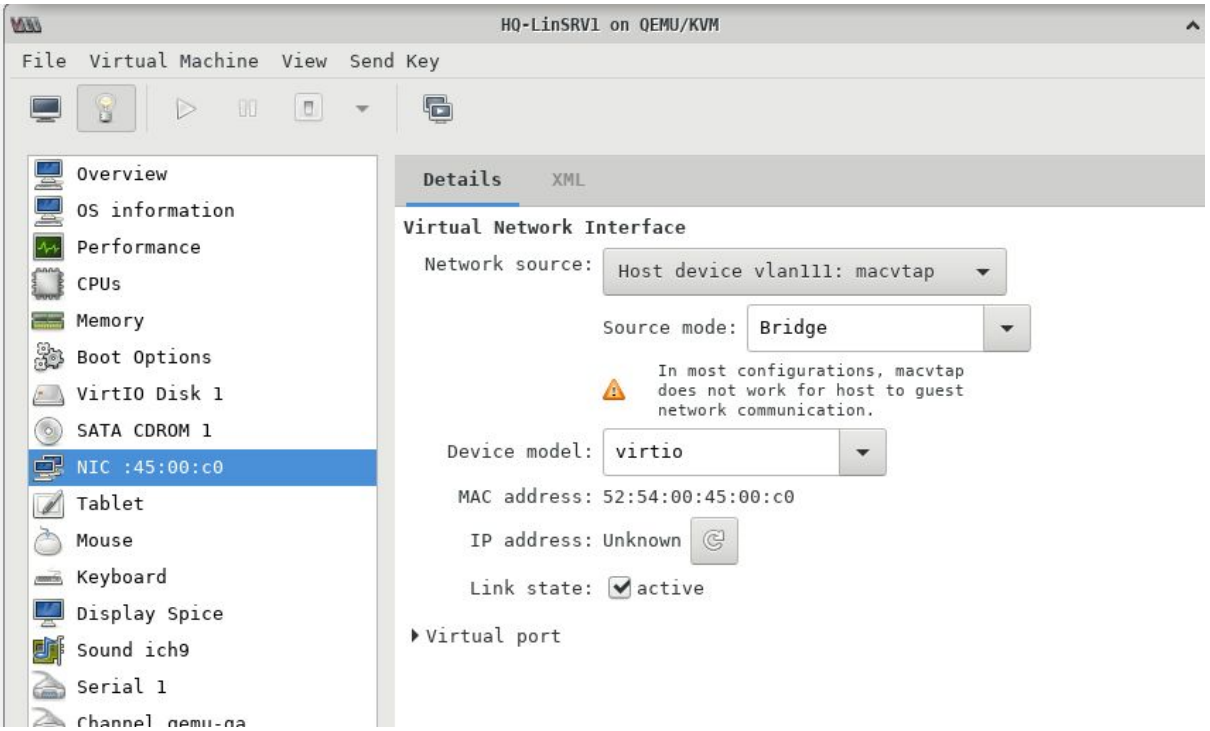
```
[root@KVM-CHECKER ~]# tcpdump -nn -i enp0s17 -e vlan
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s17, link-type EN10MB (Ethernet), capture size 262144 bytes
14:27:12.527829 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 10
1, p 0, ethertype ARP, Request who-has 10.0.4.6 tell 10.0.4.3, length 46
14:27:13.550550 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 10
1, p 0, ethertype ARP, Request who-has 10.0.4.6 tell 10.0.4.3, length 46
14:27:14.605574 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 10
1, p 0, ethertype ARP, Request who-has 10.0.4.6 tell 10.0.4.3, length 46
^C
3 packets captured
3 packets received by filter
0 packets dropped by kernel
[root@KVM-CHECKER ~]#
```

Тегированный трафик выходит из VMNIC1
Аналогичным образом формируем прочие влан-интерфейсы. В итоге наблюдаются все виды VLAN на выходе из VMNIC1. Настройка коммутации отрабатывает.

```
14:49:44.457547 52:54:00:60:42:84 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 30
0, p 0, ethertype ARP, Request who-has 145.46.48.129 tell 145.46.48.130, length 46
14:49:45.538053 52:54:00:60:42:84 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 30
0, p 0, ethertype ARP, Request who-has 145.46.48.129 tell 145.46.48.130, length 46
14:49:46.574679 52:54:00:60:42:84 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 30
0, p 0, ethertype ARP, Request who-has 145.46.48.129 tell 145.46.48.130, length 46
14:49:48.022326 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
1, p 0, ethertype ARP, Request who-has 10.0.1.4 tell 10.0.1.3, length 46
14:49:49.077839 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
1, p 0, ethertype ARP, Request who-has 10.0.1.4 tell 10.0.1.3, length 46
14:49:50.103975 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
1, p 0, ethertype ARP, Request who-has 10.0.1.4 tell 10.0.1.3, length 46
14:49:53.851400 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
2, p 0, ethertype ARP, Request who-has 10.0.2.4 tell 10.0.2.3, length 46
14:49:54.948930 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
2, p 0, ethertype ARP, Request who-has 10.0.2.4 tell 10.0.2.3, length 46
14:49:56.014876 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
2, p 0, ethertype ARP, Request who-has 10.0.2.4 tell 10.0.2.3, length 46
14:50:23.997951 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
3, p 0, ethertype ARP, Request who-has 10.0.3.4 tell 10.0.3.3, length 46
14:50:25.039446 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
3, p 0, ethertype ARP, Request who-has 10.0.3.4 tell 10.0.3.3, length 46
14:50:26.066543 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 11
3, p 0, ethertype ARP, Request who-has 10.0.3.4 tell 10.0.3.3, length 46
14:52:41.801230 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 10
3, p 0, ethertype ARP, Request who-has 10.0.1.9 tell 10.0.1.10, length 46
14:52:42.831422 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 10
3, p 0, ethertype ARP, Request who-has 10.0.1.9 tell 10.0.1.10, length 46
14:52:43.871396 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 10
3, p 0, ethertype ARP, Request who-has 10.0.1.9 tell 10.0.1.10, length 46
14:52:44.940709 52:54:00:14:b0:14 > ff:ff:ff:ff:ff:ff, ethertype 802.1Q (0x8100), length 64: vlan 10
3, p 0, ethertype ARP, Request who-has 10.0.1.9 tell 10.0.1.10, length 46
^C
```

Настройка ВМ на примере HQ-LinSRV1

Прицнипиальной разницы между ВМ в данном плане быть не должно.
В настройках выберем порт, подключенный в соответствующий VLAN



```
Adding a new '802-3-ethernet' connection

Type 'help' or '?' for available commands.
Type 'print' to show all the connection properties.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet),
proxy
nmcli> set connection.id VLAN111
nmcli> set connection.autoconnect yes
nmcli> set connection.interface-name enp1s0
nmcli> set ipv4.addresses 10.0.1.4/28
nmcli> set ipv4.method manual
nmcli> save
Saving the connection with 'autoconnect=yes'. That might result in an immed
```

Проверка связи с LinRTR

```
[root@localhost ~]# ping 10.0.1.3
PING 10.0.1.3 (10.0.1.3) 56(84) bytes of data.
64 bytes from 10.0.1.3: icmp_seq=1 ttl=64 time=6.46 ms
64 bytes from 10.0.1.3: icmp_seq=2 ttl=64 time=1.05 ms
^C
--- 10.0.1.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 4ms
rtt min/avg/max/mdev = 1.045/3.750/6.455/2.705 ms
```

Настройка прочих ВМ не должна составить проблем.

1) Выбор правильного интерфейса из списка
2) Настройка сети в соответствии с диаграммой
3) Проверка.

Проверим работу бонда

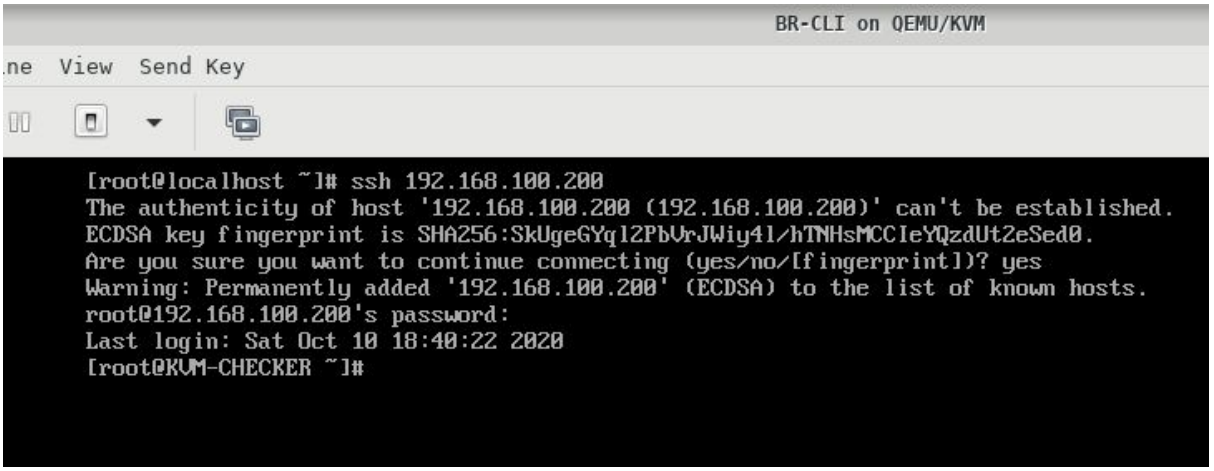
ВМ BR-CLI, в качестве интерфейса br-vlan101.

На KVM-CHECKER создадим vlan-интерфейс.

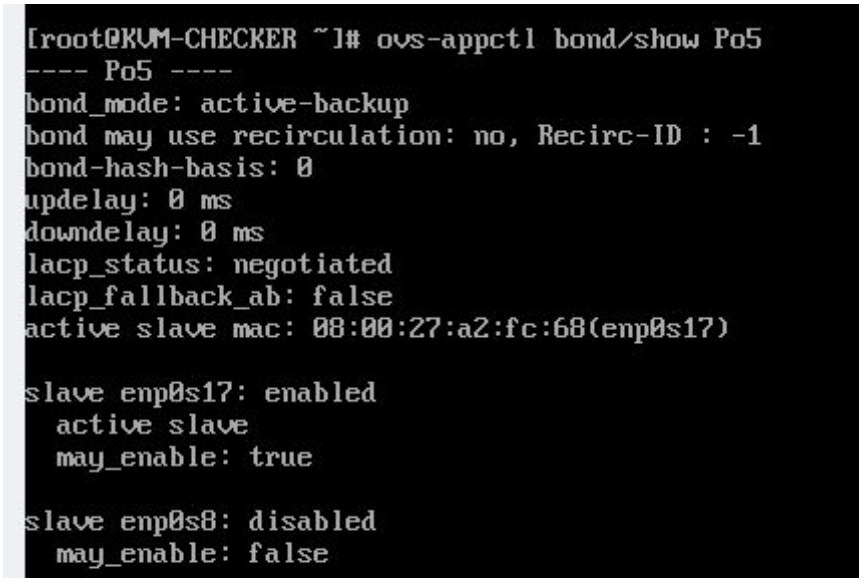
```
[root@KVM-CHECKER ~]# ovs-vsctl add-port br0 test101 tag=101 -- set interface test101 type=internal
```

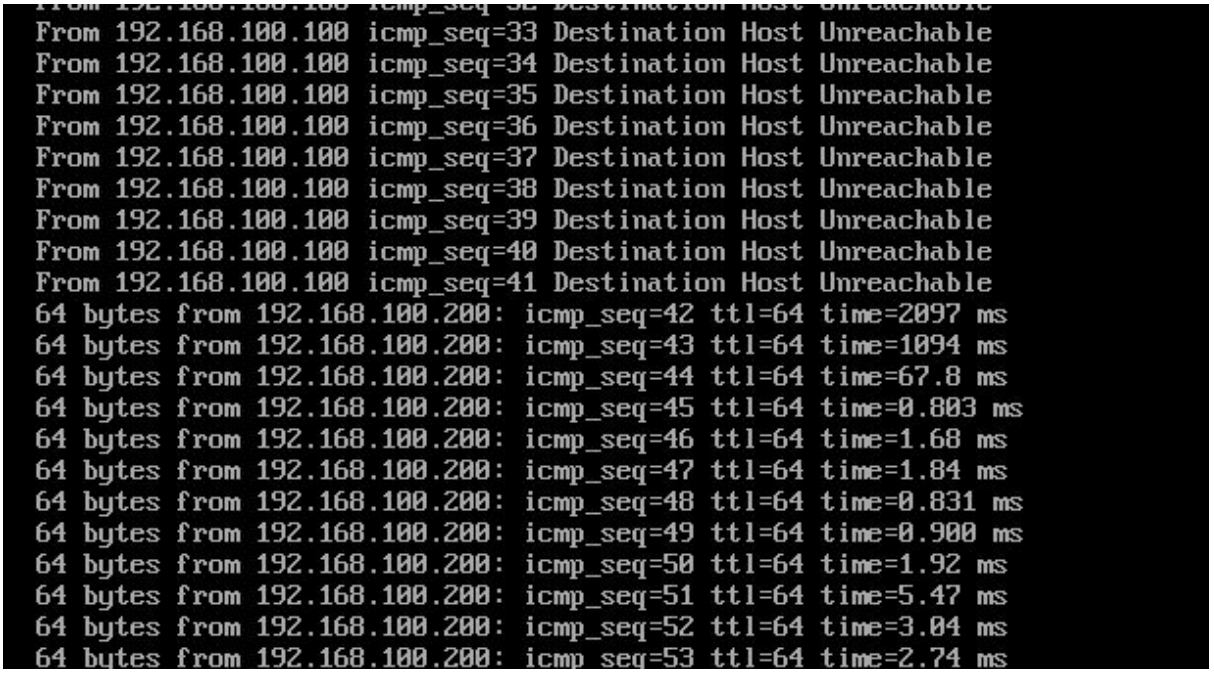
```
[root@KVM-CHECKER ~]# ip addr add 192.168.100.200/24 dev test101
[root@KVM-CHECKER ~]# systemctl disable --now firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

```
[root@KVM-CHECKER ~]# tcpdump  -nn -i enp0s8 -e vlan
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), capture size 262144 bytes
19:22:20.757003 52:54:00:fb:7e:13 > ca:15:ba:39:ef:db, ethertype 802.1Q (0x8100), length 102: vlan 1
01, p 0, ethertype IPv4, 192.168.100.100 > 192.168.100.200: ICMP echo request, id 1163, seq 65, leng
th 64
19:22:20.757213 ca:15:ba:39:ef:db > 52:54:00:fb:7e:13, ethertype 802.1Q (0x8100), length 102: vlan 1
01, p 0, ethertype IPv4, 192.168.100.200 > 192.168.100.100: ICMP echo reply, id 1163, seq 65, length
 64
19:22:21.789915 52:54:00:fb:7e:13 > ca:15:ba:39:ef:db, ethertype 802.1Q (0x8100), length 102: vlan 1
01, p 0, ethertype IPv4, 192.168.100.100 > 192.168.100.200: ICMP echo request, id 1163, seq 66, leng
th 64
19:22:21.790048 ca:15:ba:39:ef:db > 52:54:00:fb:7e:13, ethertype 802.1Q (0x8100), length 102: vlan 1
01, p 0, ethertype IPv4, 192.168.100.200 > 192.168.100.100: ICMP echo reply, id 1163, seq 66, length
 64
^C
```



Имитация отключения





DNS-сервер

yum install bind bind-utils

```
[root@srv1 ~]# yum install bind bind-utils
CentOS-8 - AppStream                                    683 kB/s | 5.8 MB     00:08
CentOS-8 - Base                                         1.7 MB/s | 2.2 MB     00:01
CentOS-8 - Extras                                        17 kB/s | 8.1 kB     00:00
Dependencies resolved.
================================================================================
 Package                Architecture     Version                    Repository          Size
================================================================================
Installing:
 bind                   x86_64           32:9.11.13-6.el8_2.1       AppStream          2.1 M
 bind-utils             x86_64           32:9.11.13-6.el8_2.1       AppStream          443 k
Installing dependencies:
 bind-libs              x86_64           32:9.11.13-6.el8_2.1       AppStream          172 k
 bind-libs-lite         x86_64           32:9.11.13-6.el8_2.1       AppStream          1.2 M
 bind-license           noarch           32:9.11.13-6.el8_2.1       AppStream          101 k
 python3-bind           noarch           32:9.11.13-6.el8_2.1       AppStream          148 k

Transaction Summary
================================================================================
Install  6 Packages

Total download size: 4.1 M
Installed size: 9.2 M
Is this ok [y/N]:
```

Рекомендуется использовать vim.
Редактируем /etc/named.conf
В разделе listen-on и allow-query указать { any;};
Если нужно, то можно в allow-query указать внутреннюю сеть, чтобы сильнее ограничить доступ.

```
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
        listen-on port 53 { any; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query     { any; };
```

Создадим описание зоны hitech2020.ac. Тип мастер, расположим файл зоны по адресу /var/named/hitech.

```
zone "hitech2020.ac" {
        type master;
        file "/var/named/hitech";
};
```

Переходим в каталог /var/named. Скопируем файл named.localhost в hitech
Установим владельца в named и группа named.

```
[root@srv1 named]# ls
data  dynamic  named.ca  named.empty  named.localhost  named.loopback  slaves
[root@srv1 named]# cp named.localhost  hitech
[root@srv1 named]# chown named:named hitech
[root@srv1 named]# ls -al
total 24
drwxrwx--T.  5 root  named  141 Oct  9 13:05 .
drwxr-xr-x. 21 root  root  4096 Oct  9 12:50 ..
drwxrwx---.  2 named named    6 Sep  9 21:47 data
drwxrwx---.  2 named named    6 Sep  9 21:47 dynamic
-rw-r-----.  1 named named  152 Oct  9 13:05 hitech
-rw-r-----.  1 root  named 2253 Sep  9 21:48 named.ca
-rw-r-----.  1 root  named  152 Sep  9 21:48 named.empty
-rw-r-----.  1 root  named  152 Sep  9 21:48 named.localhost
-rw-r-----.  1 root  named  168 Sep  9 21:48 named.loopback
drwxrwx---.  2 named named    6 Sep  9 21:47 slaves
[root@srv1 named]#
```

Редактируем hitech

```
$TTL 1D
@       IN SOA  @ rname.invalid. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      @
        A       10.10.10.100
srv1    A       10.10.10.100
srv2    A       10.10.10.200
~
~
```

При необходимости добавляем записи типа A.
проверяем конфигурацию

```
[root@srv1 named]# named-checkconf -z
zone hitech2020.ac/IN: loaded serial 0
zone localhost.localdomain/IN: loaded serial 0
zone localhost/IN: loaded serial 0
zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa/IN: loaded serial 0
zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
zone 0.in-addr.arpa/IN: loaded serial 0
[root@srv1 named]# _
```

До отдельного задания отключаем firewalld.

```
[root@srv1 named]# systemctl disable --now firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@srv1 named]# _
```

Включаем сервис.

```
[root@srv1 named]# systemctl enable --now named
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/
named.service.
[root@srv1 named]# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-10-09 13:11:45 +05; 2s ago
  Process: 10223 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0>
  Process: 10220 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sb>
 Main PID: 10225 (named)
    Tasks: 4 (limit: 6060)
   Memory: 61.6M
   CGroup: /system.slice/named.service
           └─10225 /usr/sbin/named -u named -c /etc/named.conf

Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns3-205.azure-dns.o>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns3-205.azure-dns.o>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns3-205.azure-dns.o>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns4-205.azure-dns.i>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns4-205.azure-dns.i>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns4-205.azure-dns.i>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns4-205.azure-dns.i>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns4-205.azure-dns.i>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns4-205.azure-dns.i>
Oct 09 13:11:47 srv1.hitech2020.ac named[10225]: network unreachable resolving 'ns3-04.azure-dns.or>
[root@srv1 named]# _
```

Проверка. host Имя Сервер

```
[root@srv1 named]# host hitech2020.ac  localhost
Using domain server:
Name: localhost
Address: ::1#53
Aliases:

hitech2020.ac has address 10.10.10.100
[root@srv1 named]# host srv1.hitech2020.ac  localhost
Using domain server:
Name: localhost
Address: ::1#53
Aliases:

srv1.hitech2020.ac has address 10.10.10.100
[root@srv1 named]#
```

Удаленно проверим

```
C:\Users\vmuser2>nslookup hitech2020.ac
Server:  UnKnown
Address:  10.10.10.100

Name:    hitech2020.ac
Address:  10.10.10.100


C:\Users\vmuser2>
```

CA на базе OpenSSL

Создадим каталог /etc/ca

```
[root@srv1 var]# cd /etc/
[root@srv1 etc]# mkdir ca
[root@srv1 etc]# cd ca
[root@srv1 ca]#
```

Зарегистрируем каталог в OpenSSL - откроем файл /etc/pki/tls/openssl.cnf
Отредактируем раздел CA_default.
Принципиально важно изменить раздел dir - установить /etc/ca.
Прочие каталоги распределяются относительно dir.
Здесь же можно посмотреть список важных каталогов
newcerts private certs crl
И файлов
index.txt serial
Так же видно, по каким именам ожидается CA и его ключ

```
####################################################################
[ CA_default ]

dir             = /etc/ca               # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
#unique_subject = no                    # Set to 'no' to allow creation of
                                        # several certs with same subject.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
crlnumber       = $dir/crlnumber        # the current crl number
                                        # must be commented out to leave a V1 CRL
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/cakey.pem# The private key

x509_extensions = usr_cert              # The extensions to add to the cert
```

Для простоты отредактируем политику
Ниже дан раздел policy_match.
Заменим значение stateOrProvinceName на optional. Теперь будет проще соблюсти политику сертификатов по заданию

```
policy          = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = optional
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

Создадим файлы и каталоги. В файле serial оставим значение 01.
Создадим временный каталог temp

```
[root@srv1 ca]# mkdir certs newcerts private crl
[root@srv1 ca]# touch serial
[root@srv1 ca]# touch index.txt
[root@srv1 ca]# mkdir temp
[root@srv1 ca]# echo 01 > serial
[root@srv1 ca]# ls
certs  crl  index.txt  newcerts  private  serial  temp
[root@srv1 ca]# _
```

В каталоге temp сформируем приватный ключ для CA.

```
[root@srv1 ca]# cd temp/
[root@srv1 temp]# ls
[root@srv1 temp]# openssl genrsa -out cakey.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
....++++
..............++++
e is 65537 (0x010001)
[root@srv1 temp]#
```

Сформируем запрос на сертификат в соответствии с требованиями задания

```
[root@srv1 temp]# openssl req -new -key cakey.pem  -out cacert.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:RU
State or Province Name (full name) []:.
Locality Name (eg, city) [Default City]:.
Organization Name (eg, company) [Default Company Ltd]:HT ITNSA 39
Organizational Unit Name (eg, section) []:.
Common Name (eg, your name or your server's hostname) []:WSHT CA
Email Address []:.

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
[root@srv1 temp]# _
```

Вручную подпишем сертификат своим же ключом

```
[root@srv1 temp]# openssl x509 -req -days 1000 -signkey cakey.pem  -out cacert.pem -in cacert.csr
Signature ok
subject=C = RU, O = HT ITNSA 39, CN = WSHT CA
Getting Private key
[root@srv1 temp]# _
```

Разместим файлы по нужным адресам. Проверяем.
Если все обнаружено - openssl ca не выдаст ошибок

```
[root@srv1 temp]# cp cacert.pem /etc/ca/
[root@srv1 temp]# cp cakey.pem  /etc/ca/private/
[root@srv1 temp]# cd
[root@srv1 ~]# openssl ca
Using configuration from /etc/pki/tls/openssl.cnf
[root@srv1 ~]# _
```

В рамках тестирования сформируем сертификат на имя test.hitech2020.ac. Так же отработаем проверку. Работаем в каталоге temp.

1) Формируем ключ
2) Формируем запрос
3) Подписываем

Запрос должен соответствовать политике - та же страна и организация

```
[root@srv1 temp]# openssl genrsa -out testkey.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....................................................................................
.....................................................................................
.....................................++++
.....................................................................++++
e is 65537 (0x010001)
[root@srv1 temp]# openssl req -new -key testkey.pem  -out test.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:RU
State or Province Name (full name) []:.
Locality Name (eg, city) [Default City]:.
Organization Name (eg, company) [Default Company Ltd]:HT ITNSA 39
Organizational Unit Name (eg, section) []:.
Common Name (eg, your name or your server's hostname) []:test.hitech2020.ac
Email Address []:.

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:
[root@srv1 temp]#
```

Подписываем

```
[root@srv1 temp]# openssl ca -in test.csr  -out test.pem
Using configuration from /etc/pki/tls/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Oct  9 11:46:19 2020 GMT
            Not After : Oct  9 11:46:19 2021 GMT
        Subject:
            countryName               = RU
            organizationName          = HT ITNSA 39
            commonName                = test.hitech2020.ac
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                D2:68:D4:26:F5:B4:D4:13:26:9F:2C:18:E0:FC:6B:5E:D5:BD:33:4B
            X509v3 Authority Key Identifier:
                DirName:/C=RU/O=HT ITNSA 39/CN=WSHT CA
                serial:08:51:23:74:2A:3B:13:47:23:D5:72:95:E1:E8:61:14:43:53:76:90

Certificate is to be certified until Oct  9 11:46:19 2021 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
[root@srv1 temp]#
```

Проверяем

```
[root@srv1 temp]# openssl verify  test.pem
C = RU, O = HT ITNSA 39, CN = test.hitech2020.ac
error 20 at 0 depth lookup: unable to get local issuer certificate
error test.pem: verification failed
[root@srv1 temp]# _
```

Добавим сертификат в список доверенных

```
[root@srv1 temp]# cp /etc/ca/cacert.pem  /usr/share/pki/ca-trust-source/anchors/
[root@srv1 temp]# update-ca-trust
[root@srv1 temp]# openssl verify  test.pem
test.pem: OK
[root@srv1 temp]#
```

На прочих Centos8 машинах выполняется аналогично.

- копирование cacert.pem в /usr/share/pki/ca-trust-source/anchors/
- Вызов update-ca-trust

Прием сообщений syslog

Создадим каталог в соответствии с заданием.

```
[root@srv1 ~]# mkdir -p  /opt/logs/cisco
[root@srv1 ~]# _
```

Перевед SELinux в щадящий режим, установим значение permissive.
Далее либо перезагрузка, либо setenforce 0

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Обеспечим прием журналов по TCP и UDP. Отредактируем конфигурацию в /etc/rsyslog.conf
Для приема журналов требуется раскомментировать пункты module и input с упоминанием TCP и UDP.

```
# Provides UDP syslog reception
# for parameters see http://www.rsyslog.com/doc/imudp.html
module(load="imudp") # needs to be done just once
input(type="imudp" port="514")

# Provides TCP syslog reception
# for parameters see http://www.rsyslog.com/doc/imtcp.html
module(load="imtcp") # needs to be done just once
input(type="imtcp" port="514")
```

Создадим правило, направляющее журналы в отдельные файлы каталога /opt/logs/cisco.
(здесь для тестов используются SRV1 и SRV2).

Редактируем начало раздела RULES.
“Если сообщение пришло с адреса X то сохраним в файл Ф. Остановить обработку”

```
#### RULES ####

if $fromhost-ip == "10.10.10.200" then  {
        /opt/logs/cisco/srv2.log
        stop
}
```

При необходимости повторить сколько необходимо раз - в итоге журналы всех устройств будут разделены по файлам в каталоге /opt/logs/cisco.

Установка и настройка сервера LDAP

Используется репозиторий epel. SELinux в permissive.

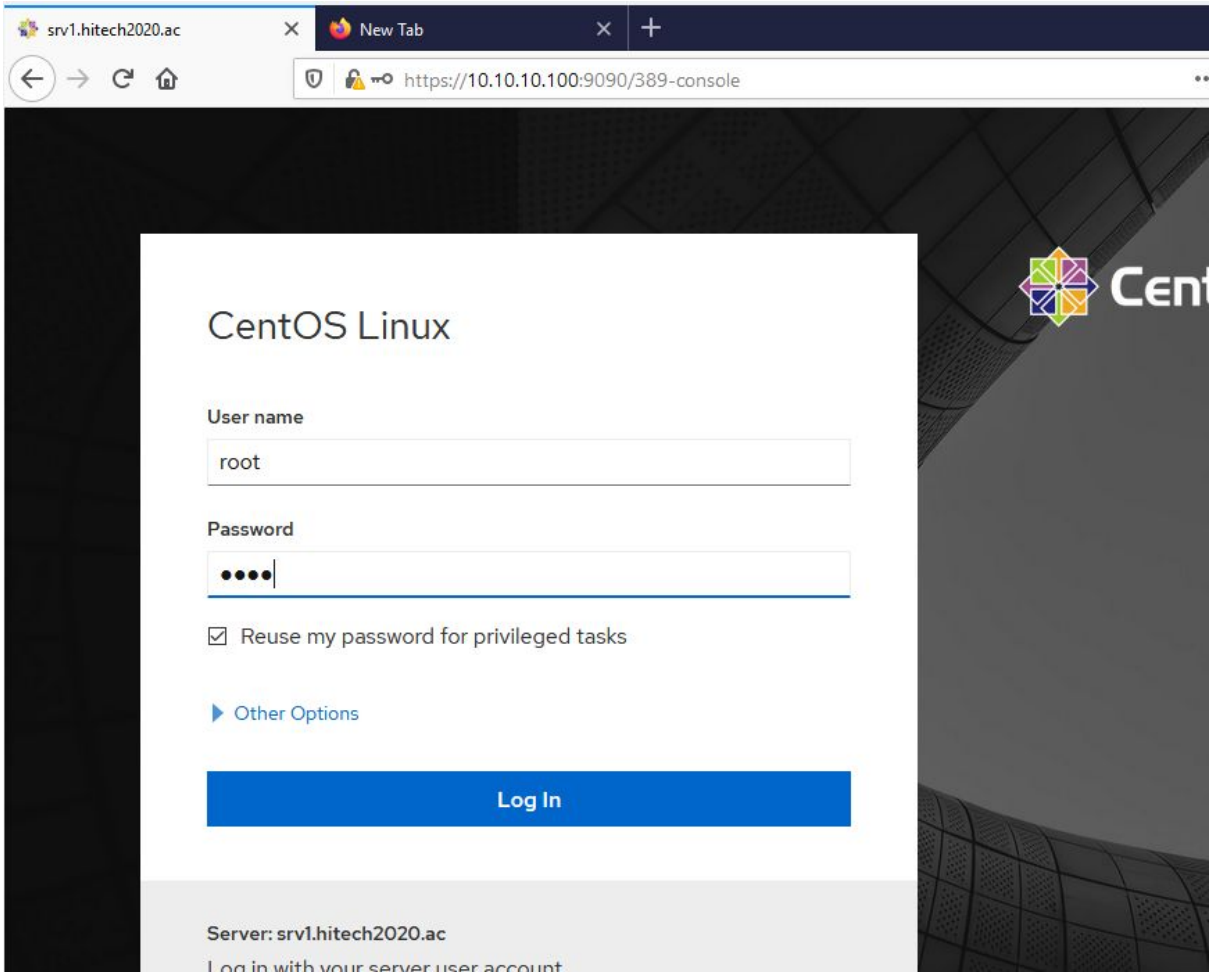
```
dnf -y module install 389-directory-server:stable/default
```

Интерактивный конфигуратор dscreate interactive
Потребуется сконфигурировать основные параметры подключения, пароль администратора сервера и обслуживаемую БД. т.е. домен. dc=hitech2020,dc=ac

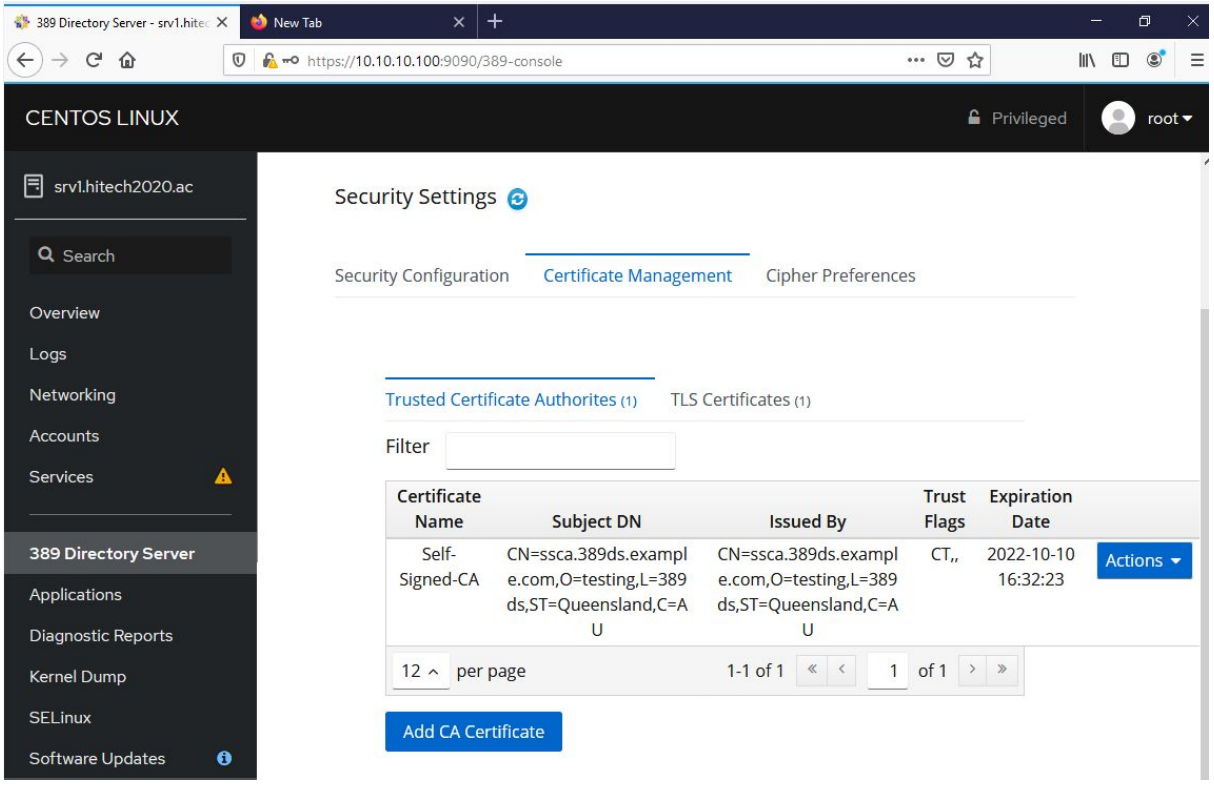
При запросе “Create the top suffix entry” ответить yes!

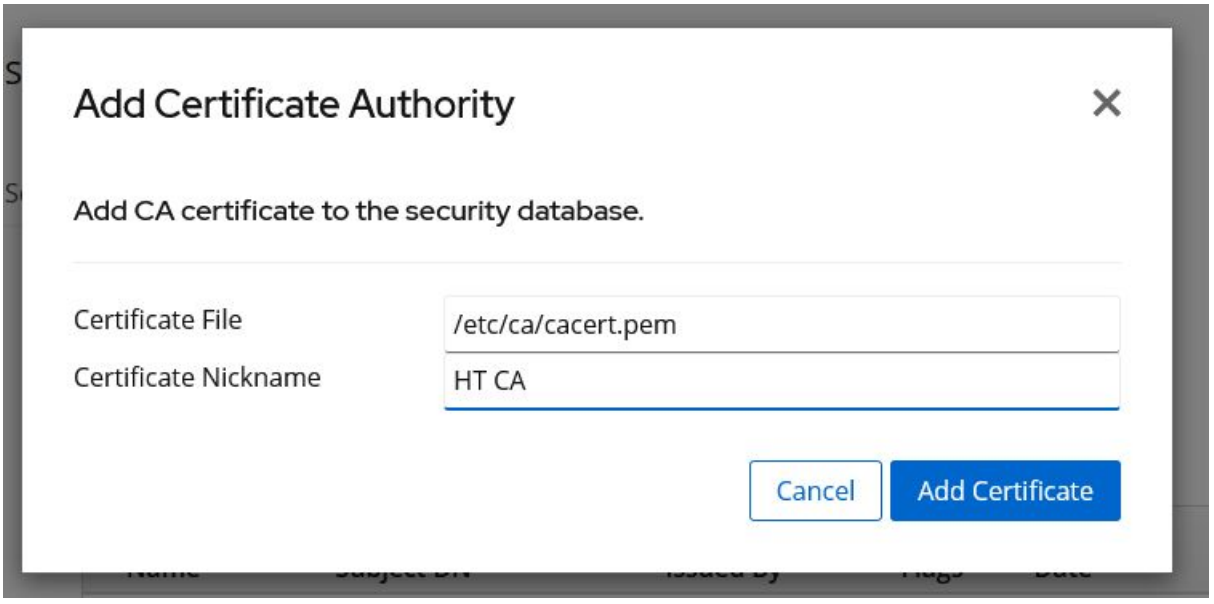
```
[root@srv1 ~]# dscreate interactive
Install Directory Server (interactive mode)
===========================================

Enter system's hostname [srv1.hitech2020.ac]:

Enter the instance name [srv1]:

Enter port number [389]:

Create self-signed certificate database [yes]:

Enter secure port number [636]:

Enter Directory Manager DN [cn=Directory Manager]:

Enter the Directory Manager password:
Confirm the Directory Manager Password:

Enter the database suffix (or enter "none" to skip) [dc=srv1,dc=hitech2020,dc=ac]: dc=hitech2020,dc=
ac

Create sample entries in the suffix [no]:

Create just the top suffix entry [no]: yes

Do you want to start the instance after the installation? [yes]:

Are you ready to install? [no]: yes
Starting installation...
Completed installation for srv1
[root@srv1 ~]# _
```

Проверим с помощью cockpit. Требуется - компьютер с графикой и бразуер нормальный.
systemctl start cockpit
https://<IP>:9090

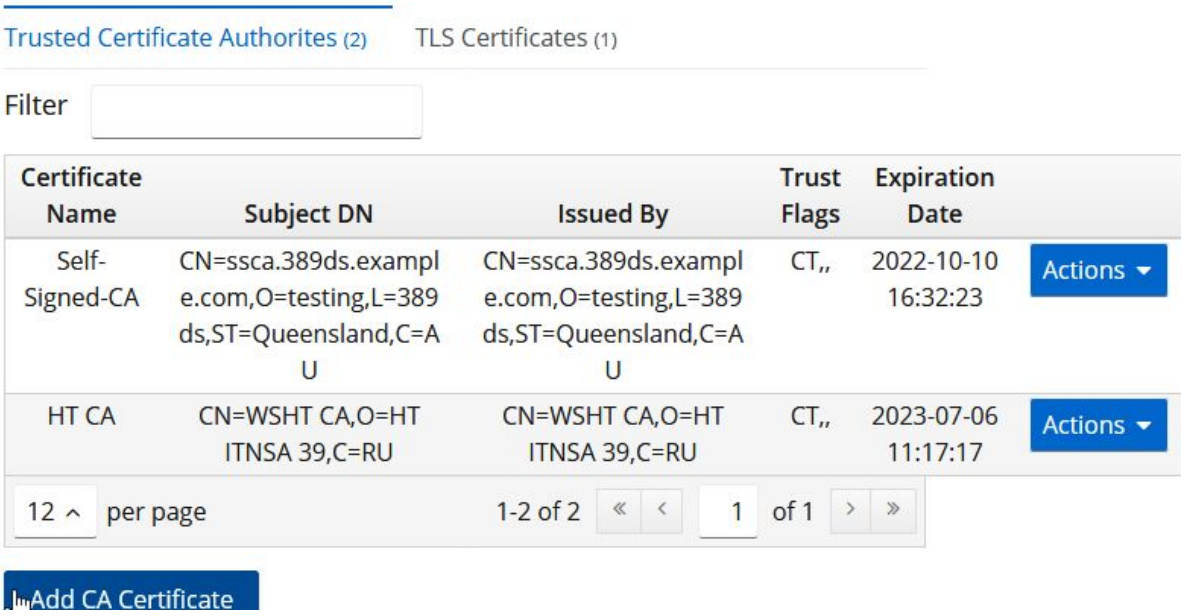


Раздел Directory server - Security - Certificate Management - Add CA Certificate
Добавим сертификат. Добавление производится локально, т.е. сертификат должен быть на одной машине с 389ds





Это пригодится на следующем шаге



Проверим базовую проверку с помощью утилиты ldapsearch. Для усложнения проведем с другой машины, т.е. по сети.
Проверим что суффикс dc=hitech2020,dc=ac существует.
-H куда и по какому протоколу запрос
-D под каким пользователем
-w пароль в открытом виде
-b база поиска
-LLL только данные, без комментов
Выводим все, что найдем

```
[root@srv2 sssdl]# ldapsearch  -H ldap://10.10.10.100  -D "cn=Directory Manager" -b "dc=hitech2020,dc=ac" -w P@ssw0rd  -LLL
dn: dc=hitech2020,dc=ac
objectClass: top
objectClass: domain
dc: hitech2020
description: dc=hitech2020,dc=ac

[root@srv2 sssdl]#
```

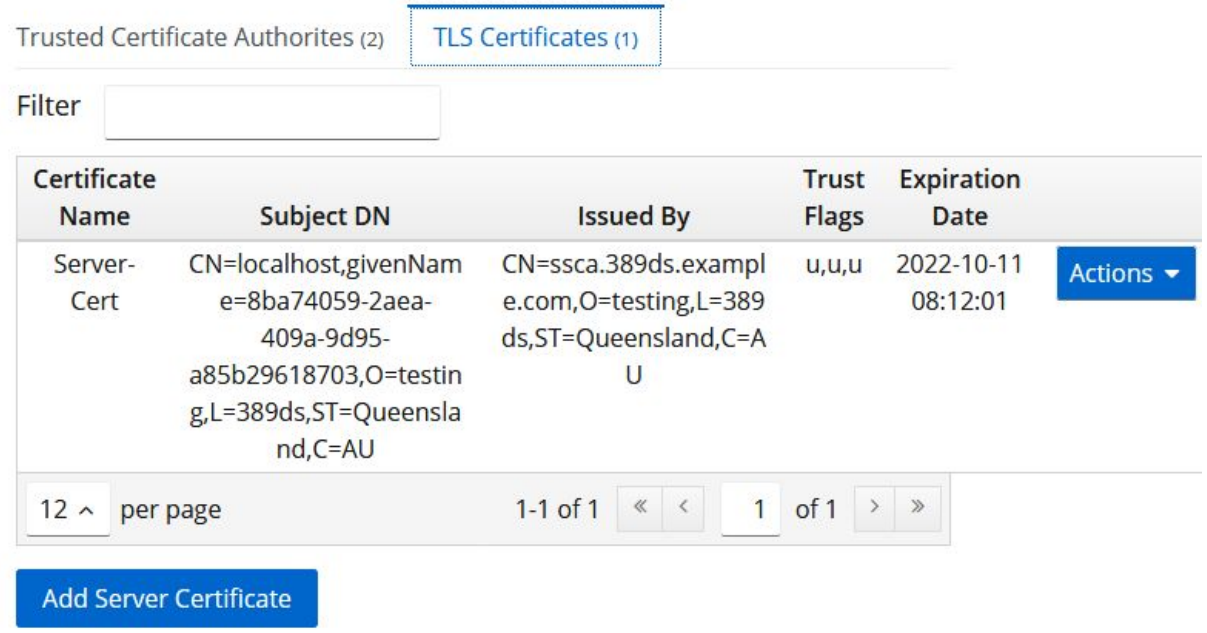
Для правильной интеграции с SSSD нам нужен защищенный канал.
пусть дан сертификат и ключ на имя srv1.hitech2020.ac
srv1key.pem и srv1.pem
389 использует специфичную БД сертификатов. Сформируем pfx

```
[root@srv1 temp]# openssl pkcs12 -export -out srv1.pfx -inkey srv1key.pem  -in srv1.pem  -certfile /etc/ca/cacert.pem
Enter Export Password:
Verifying - Enter Export Password:
[root@srv1 temp]# _
```

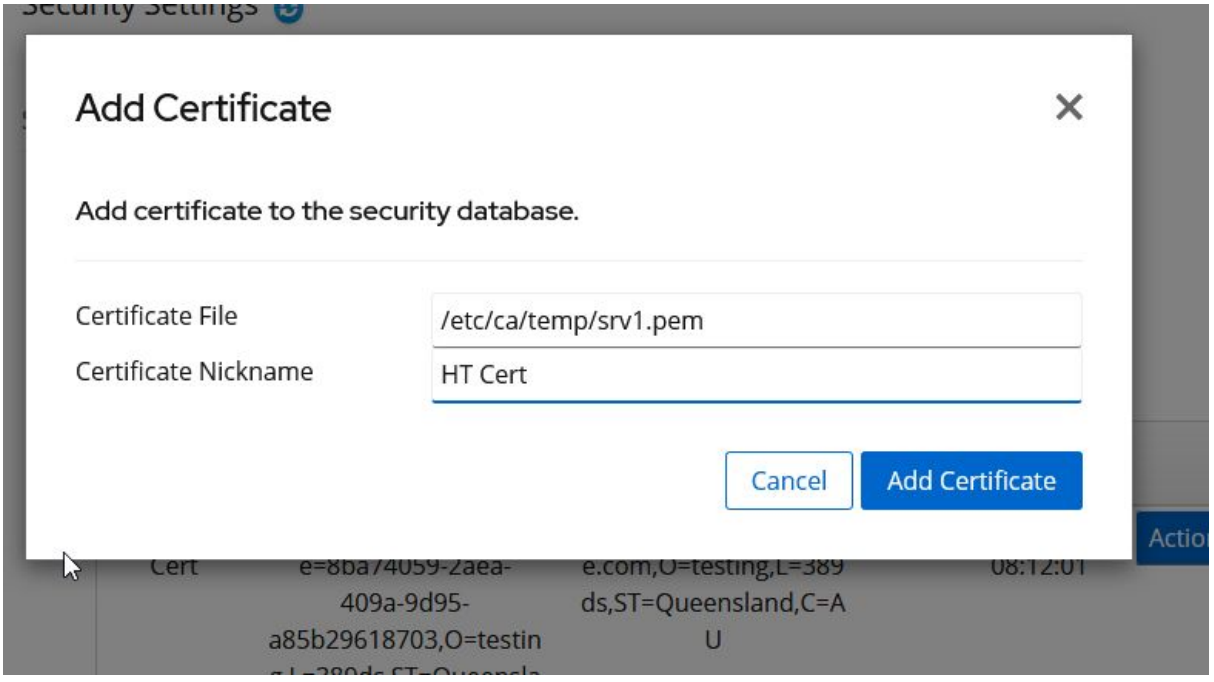
Добавим в БД сертификатов, (главное чтобы добавился приватный ключ. Веб-интерфейс не умеет в приватные ключи)

```
[root@srv1 temp]# pk12util -i /etc/ca/temp/srv1.pfx   -d /etc/dirsrv/slapd-srv1/  -k /etc/dirsrv/sla
pd-srv1/pwdfile.txt
Enter password for PKCS12 file:
pk12util: no nickname for cert in PKCS12 file.
pk12util: using nickname: srv1.hitech2020.ac - HT ITNSA 39
pk12util: PKCS12 IMPORT SUCCESSFUL
[root@srv1 temp]#
```

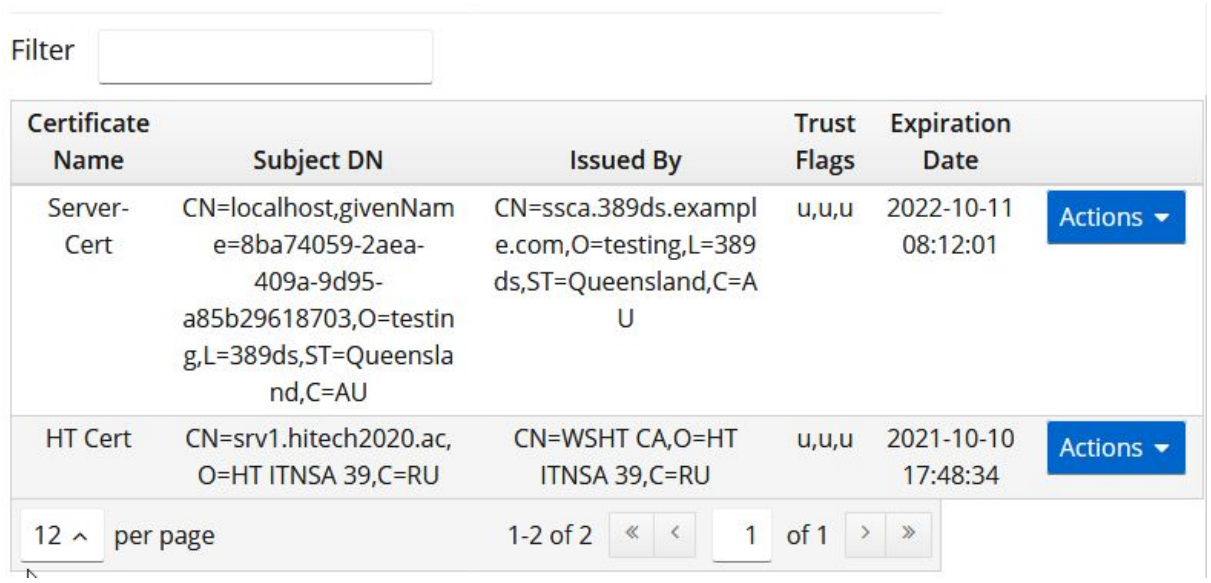
Теперь добавим сертификат через форму.



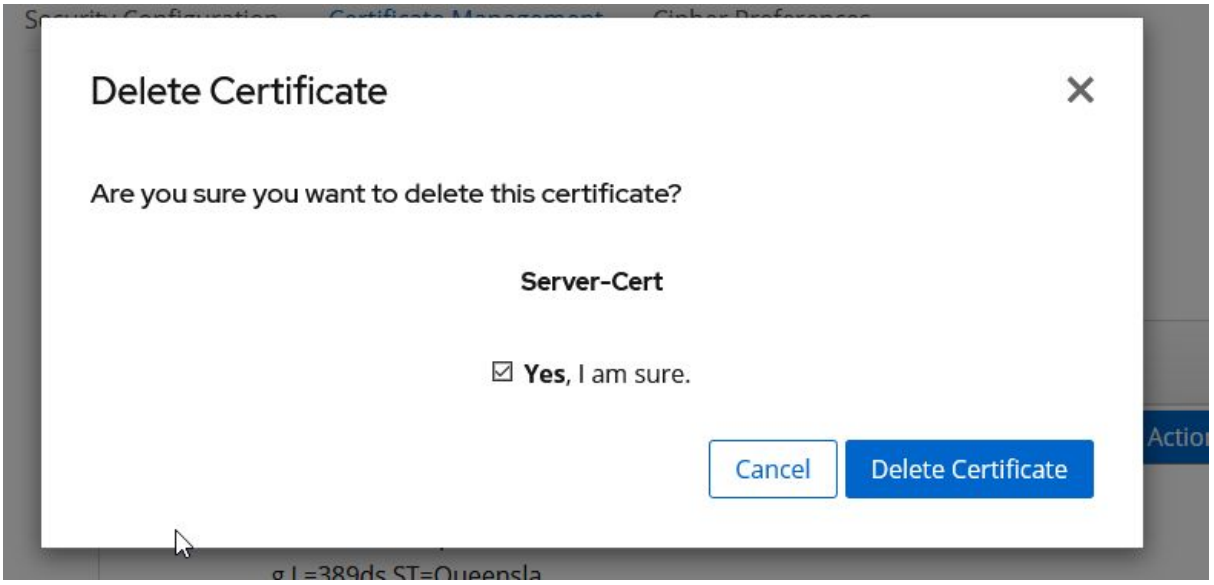
Добавим оригинальный сертификат PEM!



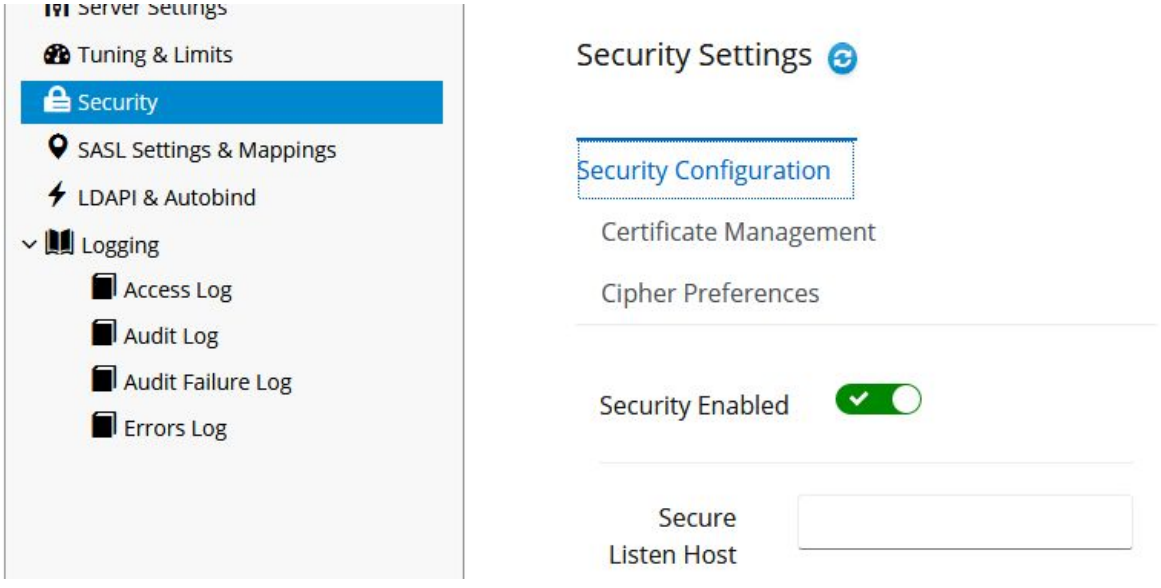
Главное - чтобы появились Trust Flags u,u,u!

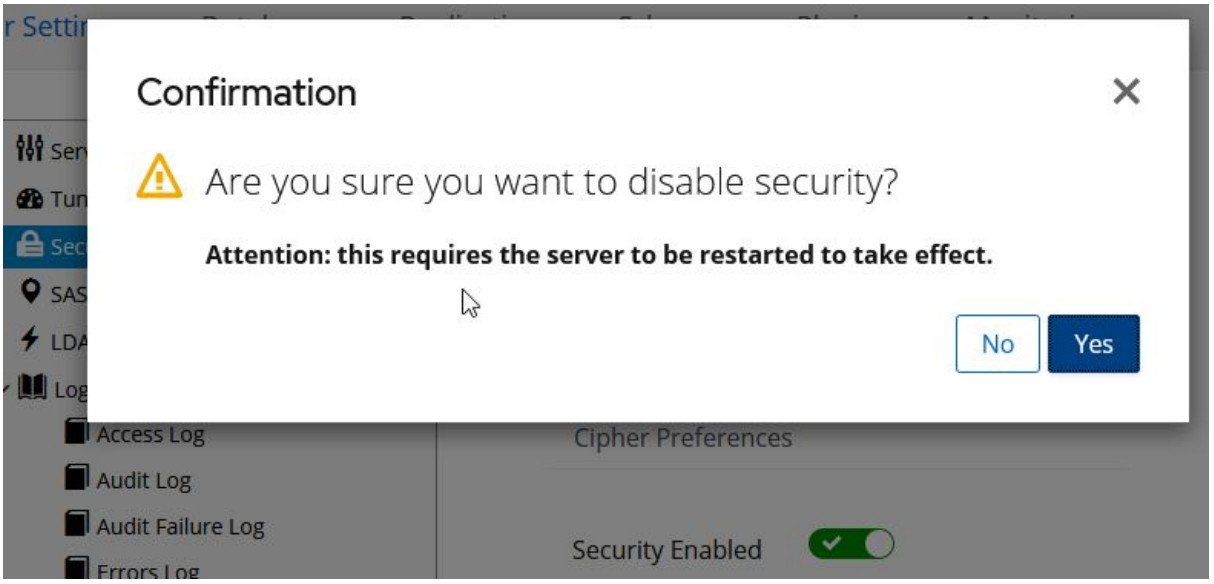


Удаляем старый самоподписанный сертификат

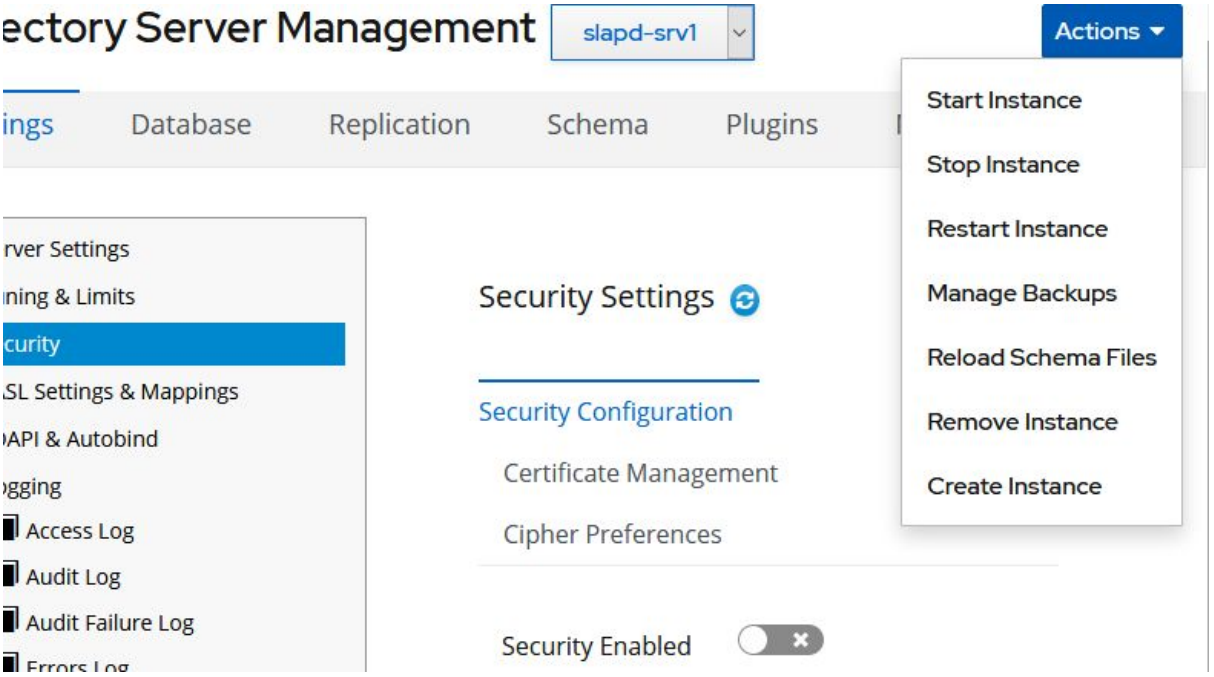


Переходи в раздел Security settings и ОТКЛЮЧАЕМ безопасность.

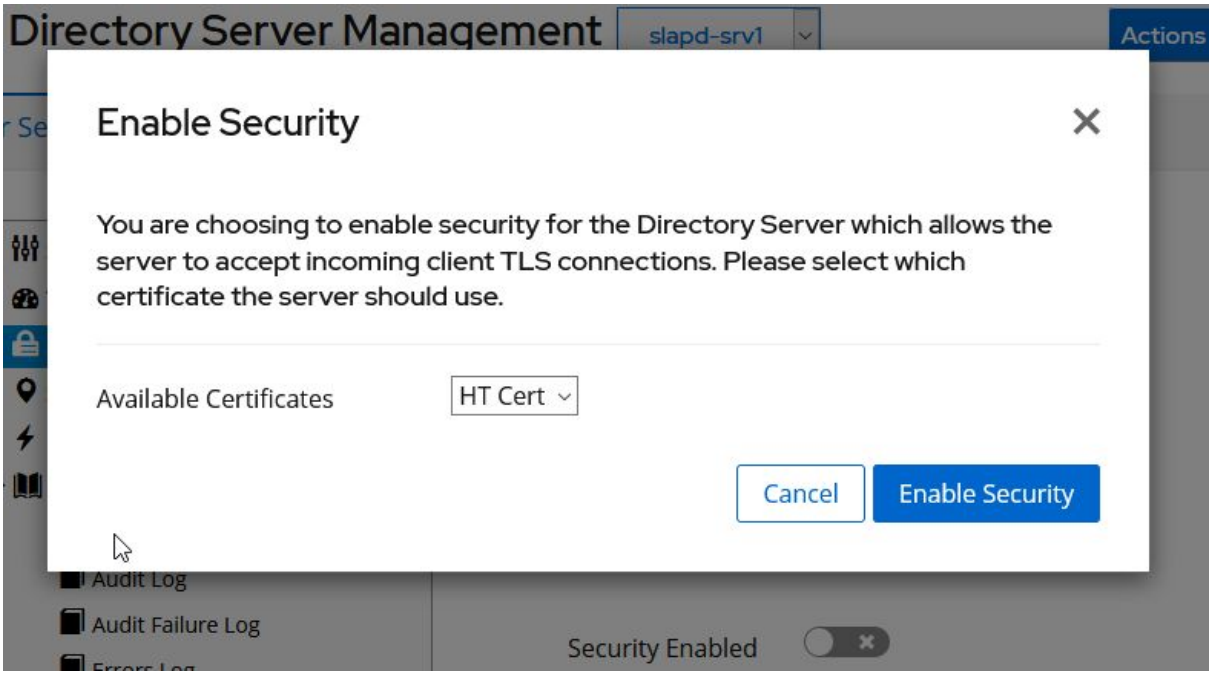




Перезагружаем сервер



Возвращаемся в раздел Security Settings и ВКЛЮЧАЕМ безопасность. Выбираем наш сертификат. Перезагружаем.



Не обращаем внимания на сервер-сертифкат имя, это баг.

Проверяем работу защищенного соединения. Машина с которой проверяем уже должна доверять нашему CA, выпустившему сертификат!!

команда ldapsearch изменена - в первом случае мы тестируем чистый ldaps, во втором - STARTTLS (ключ -ZZ)

```
[root@srv2 sssd]# ldapsearch  -D "cn=Directory Manager" -w P@ssw0rd -H ldaps://srv1.hitech2020.ac -b
 "dc=hitech2020,dc=ac" -LLL
dn: dc=hitech2020,dc=ac
objectClass: top
objectClass: domain
dc: hitech2020
description: dc=hitech2020,dc=ac

[root@srv2 sssd]# ldapsearch -ZZ -D "cn=Directory Manager" -w P@ssw0rd -H ldap://srv1.hitech2020.ac
-b "dc=hitech2020,dc=ac" -LLL
dn: dc=hitech2020,dc=ac
objectClass: top
objectClass: domain
dc: hitech2020
description: dc=hitech2020,dc=ac

[root@srv2 sssd]#
```

Создадим данные, формируем файл sysadmins.ldif. Пробелы важны, пропуски строк важны, рекомендуется делать в vim - он подсвечивает.

1) Создаем OU для пользователей и групп
2) Создаем группу сисадмины
    a) Запоминаем gidNumber
3) Создаем пользователей. Должны отличаться
    a) uid
    b) homeDirectory
    c) uidNumber

На скриншоте не все поместилось, там копирование с минимальными изменениями.

```
dn: ou=users,dc=hitech2020,dc=ac
objectClass: organizationalUnit
ou: users

dn: ou=groups,dc=hitech2020,dc=ac
objectClass: organizationalUnit
ou: groups

dn: cn=Sysadmins,ou=groups,dc=hitech2020,dc=ac
objectClass: posixGroup
gidNumber: 5001


dn: uid=SuperAdmin,ou=Users,dc=hitech2020,dc=ac
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
homeDirectory: /home/overrideme
cn: ivan
sn: ivan
userpassword: P@ssw0rd
uid: SuperAdmin
uidNumber: 5001
gidNumber: 5001


dn: uid=MegaAdmin,ou=Users,dc=hitech2020,dc=ac
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
homeDirectory: /home/overrideme
cn: ivan
sn: ivan
userpassword: P@ssw0rd
uid: MegaAdmin
uidNumber: 5002
```

Загружаем. Используем защищенный канал.
На скрине пример команды

```
add uidNumber:
        5003
add gidNumber:
        5001
adding new entry "uid=HyperAdmin,ou=Users,dc=hitech2020,dc=ac"
modify complete

[root@srv2 ~]# ldapadd -vv  -Z -D "cn=Directory Manager" -w P@ssw0rd -H ldap://srv1.hitech2020.ac -f
 task.ldif_
```

Проверяем ldapsearch. Данные видны.

```
[root@srv2 ~]# ldapsearch -ZZ -D "cn=Directory Manager" -w P@ssw0rd -H ldap://srv1.hitech2020.ac -b
"dc=hitech2020,dc=ac" -LLL | head -n 30
dn: dc=hitech2020,dc=ac
objectClass: top
objectClass: domain
dc: hitech2020
description: dc=hitech2020,dc=ac

dn: ou=users,dc=hitech2020,dc=ac
objectClass: organizationalUnit
objectClass: top
ou: users

dn: ou=groups,dc=hitech2020,dc=ac
objectClass: organizationalUnit
objectClass: top
ou: groups

dn: cn=Sysadmins,ou=groups,dc=hitech2020,dc=ac
objectClass: posixGroup
objectClass: top
gidNumber: 5001
cn: Sysadmins

dn: uid=SuperAdmin,ou=users,dc=hitech2020,dc=ac
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: organizationalPerson
objectClass: top
objectClass: person
cn: ivan
[root@srv2 ~]#
```

По аналогии создаем прочие группы. uidNumber выбрать самостоятельно, лишь бы были больше 1000 и не повторлись. gidnumber для групп аналогично. Лучше сделать отдельным файлом в формате группа + пользователи этой группы.

Централизованная аутентификация.

Установим необходимое ПО.

```
[root@srv1 ~]# yum install openldap-clients sssd sssd-ldap oddjob-mkhomedir
```

Включим поддержку sssd, запустим сервис oddjobd.

```
[root@srv1 ~]# authselect select sssd with-mkhomedir --force
Profile "sssd" was selected.
The following nsswitch maps are overwritten by the profile:
- passwd
- group
- netgroup
- automount
- services

Make sure that SSSD service is configured and enabled. See SSSD documentation for more information.

- with-mkhomedir is selected, make sure pam_oddjob_mkhomedir module
  is present and oddjobd service is enabled
  - systemctl enable oddjobd.service
  - systemctl start oddjobd.service

[root@srv1 ~]# systemctl enable --now oddjobd
Created symlink /etc/systemd/system/multi-user.target.wants/oddjobd.service → /usr/lib/systemd/syste
m/oddjobd.service.
[root@srv1 ~]# _
```

Создадим конфигурационный файл sssd, присвоим права 0600

```
[root@srv1 ~]# touch /etc/sssd/sssd.conf
[root@srv1 ~]# chmod 0600 /etc/sssd/sssd.conf
[root@srv1 ~]#
```

Конфиг. Важное

1) cacert должен присутствовать на машине в виде файла, для надежности.
2) имя должно резолвиться.
3) Пароль который указали при dscreate
4) Параметры override позволяют нам НЕ задавать шелл.
5) simple_allow_groups ограничивает вход.

```
[sssd]
services = nss, pam
domains = hitech2020.ac

[domain/hitech2020.ac]
enumerate = True
id_provider = ldap
auth_provider = ldap
chpass_provider = ldap
ldap_uri = ldaps://srv1.hitech2020.ac
ldap_search_base = dc=hitech2020,dc=ac
ldap_default_bind_dn = cn=Directory Manager
ldap_default_authtok = P@ssw0rd
cache_credentials = False
ldap_tls_cacert = /etc/sysconfig/cacert.pem
override_shell = /bin/bash
override_homedir =  /home/%u

access_provider = simple
simple_allow_groups = Sysadmins, Uzvers
```

Старт и проверка. На подгрузку имен может потребоваться время, секунд 30(непонятно, мб особенность данной инсталляции)

```
[root@srv1 ~]# systemctl start sssd
[root@srv1 ~]# systemctl status sssd
● sssd.service - System Security Services Daemon
   Loaded: loaded (/usr/lib/systemd/system/sssd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2020-10-11 13:09:15 +05; 1h 32min ago
 Main PID: 780 (sssd)
    Tasks: 3 (limit: 6060)
   Memory: 37.6M
   CGroup: /system.slice/sssd.service
           ├─780 /usr/sbin/sssd -i --logger=files
           ├─812 /usr/libexec/sssd/sssd_be --domain implicit_files --uid 0 --gid 0 --logger=files
           └─821 /usr/libexec/sssd/sssd_nss --uid 0 --gid 0 --logger=files

Oct 11 13:09:13 srv1.hitech2020.ac systemd[1]: Starting System Security Services Daemon...
Oct 11 13:09:14 srv1.hitech2020.ac sssd[780]: Starting up
Oct 11 13:09:14 srv1.hitech2020.ac sssd[be[implicit_files]][812]: Starting up
Oct 11 13:09:14 srv1.hitech2020.ac sssd[nss][821]: Starting up
Oct 11 13:09:15 srv1.hitech2020.ac systemd[1]: Started System Security Services Daemon.
[root@srv1 ~]# _
```

Проверка имен. Аккаунты распознаются.

```
[root@srv1 ~]# getent passwd | head -n 10
HyperAdmin:*:5003:5001:ivan:/home/HyperAdmin:
MegaAdmin:*:5002:5001:ivan:/home/MegaAdmin:
SuperAdmin:*:5001:5001:ivan:/home/SuperAdmin:
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

Проверка ограничений

```
CentOS Linux 8 (Core)
Kernel 4.18.0-147.el8.x86_64 on an x86_64

Web console: https://srv1.hitech2020.ac:9090/ or https://10.0.3.15:9090/

Hint: Num Lock on

srv1 login: HyperAdmin
Password:
Last login: Sun Oct 11 23:45:57 on tty1
[HyperAdmin@srv1 ~]$
```

```
CentOS Linux 8 (Core)
Kernel 4.18.0-147.el8.x86_64 on an x86_64

Web console: https://srv1.hitech2020.ac:9090/ or https://10.0.3.15:9090/

Hint: Num Lock on

srv1 login: Gates
Password:

Permission denied
```