Создание службы Samba для FreeIPA

Данный документ показывает порядок действий для формирования виртуальной инфраструктуры для сетевого хранения перемещаемых профилей и домашних папок пользователей FreeIPA используя ресурсы частного облака.

Учитывая требования импортозамещения Российской Федерации документ демонстрирует использование AccentOS и Astra Linux SE Smolensk версии 1.6.

Условно работы можно разбить на несколько этапов:

* создание образа
* создание сетевых портов
* создание группы безопасности
* создание виртуальных машин
* создание общего диска для данных
* одновременный запуск скрипта установки Samba на всех виртуальных машинах

Необходимые предварительные условия:

* для создания и дальнейшей настройки виртуальных машин требуется ключевая пара. Публичный ключ должен быть импортирован в облако для создания и возможности дальнейшего доступа к командной строке операционной системы виртуальной машины. Секретный ключ необходим для доступа к командной строке ВМ и дальнейшего использования в скрипте для синхронизации установки Samba на ВМ.
* сетевые репозитории для Astra Linux SE Smolensk версии 1.6. В числе требуемых репозиториев - репозиторий установочного диска, репозиторий пакетов для разработчика, а также обновления репозитория установочного диска и репозитория пакетов разработчика. Установочный диск приобретается вместе с лицензией на Astra Linux SE Smolensk, к остальным образам репозиториев разработчик предоставил свободный доступ. Создание сетевых репозиториев описано разработчиком в статье <https://wiki.astralinux.ru/pages/viewpage.action?pageId=61575159>

## 

## Создание образа

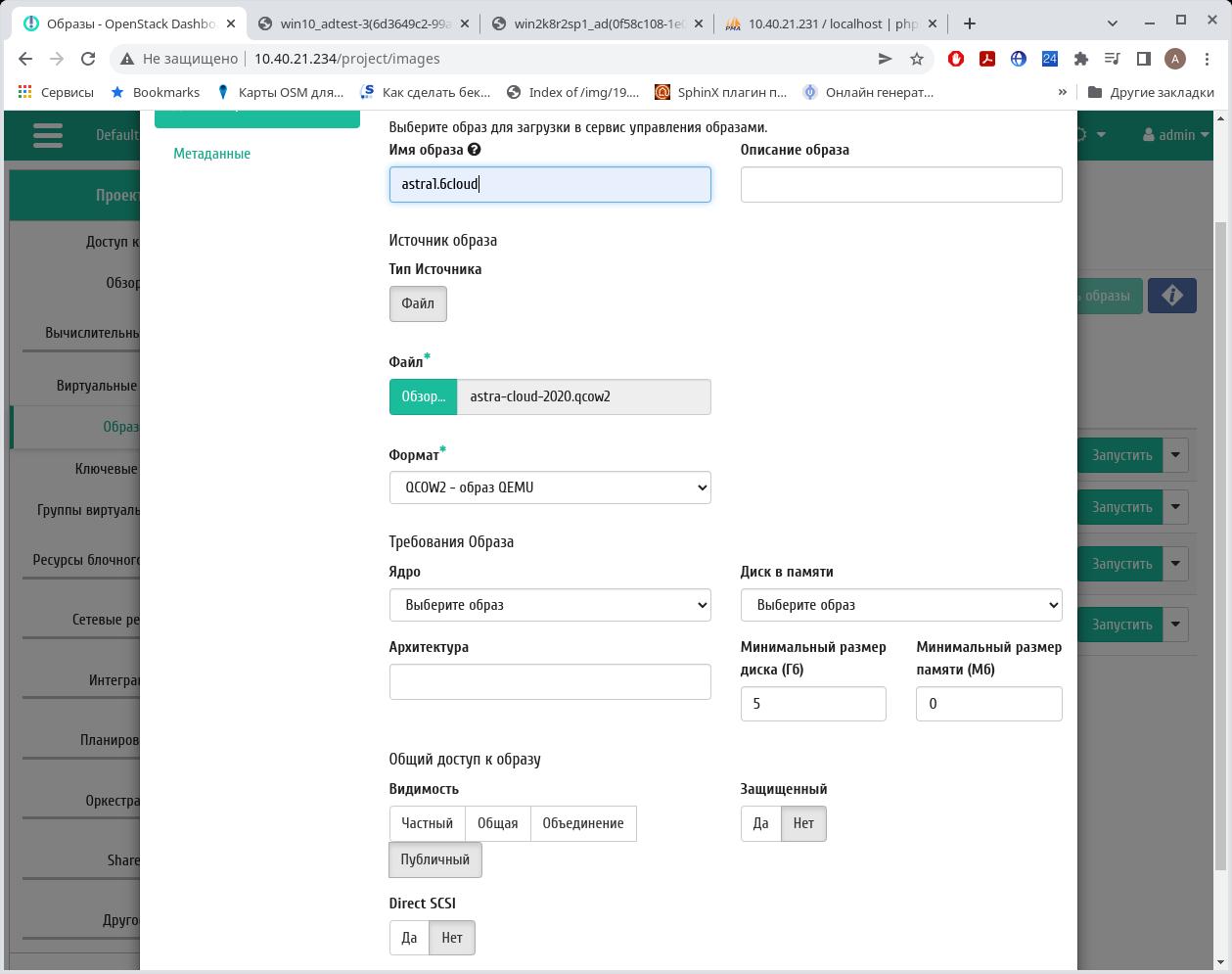
Любая виртуальная машина в облаке создается с использованием образа. Мы рассмотрим наиболее распространенный способ создания ВМ из образа диска установленной операционной системы с установленным пакетом cloud-init для возможности последующей донастройки операционной системы в облаке: увеличении диска, изменение имени ВМ, создание пользователей, назначение пароля или ключевой пары и так далее.

Мы будем использовать готовый образ Linux Astra SE 1.6 с установленными обновлениями 2020 года и пакетом cloud-init.

Хотелось бы обратить внимание на заполнение некоторых полей:

* минимальный размер диска должен соответствовать размеру виртуального диска, представленного выбранным образом диска с установленной операционной системой. Значение данного параметра можно получить командой “qemu-img info ...”.
* флаг Direct SCSI для Linux систем нужно устанавливать в состояние “Нет”. Данный флаг в положении “Да” указывает облаку при создании диска виртуальной машины использовать эмуляцию диска, подключенного через SCSI-контроллер. При этом ядро виртуальной машины может игнорировать указываемое облаком имя диска в виде “/dev/vda” и назначать имя диска согласно оборудованию - “/dev/sda”, что может изменить имена других блочных устройств внутри ВМ. Ничем другим, кроме несоответствия имен устройств в облаке и внутри ВМ это не грозит, но при дополнительное ПО, получающее имена устройств от служб облака, может работать некорректно внутри виртуальной машины. Для прочих операционных систем, не имеющих встроенных драйверов virtio, выставление флага Direct SCSI в состояние “Да” возможно поможет решить вопрос работоспособности виртуальных машин после создания с данного образа.

Импорт образа для создания виртуальных машин доступно на странице **Проект** -> **Вычислительные ресурсы** -> **Образы**.

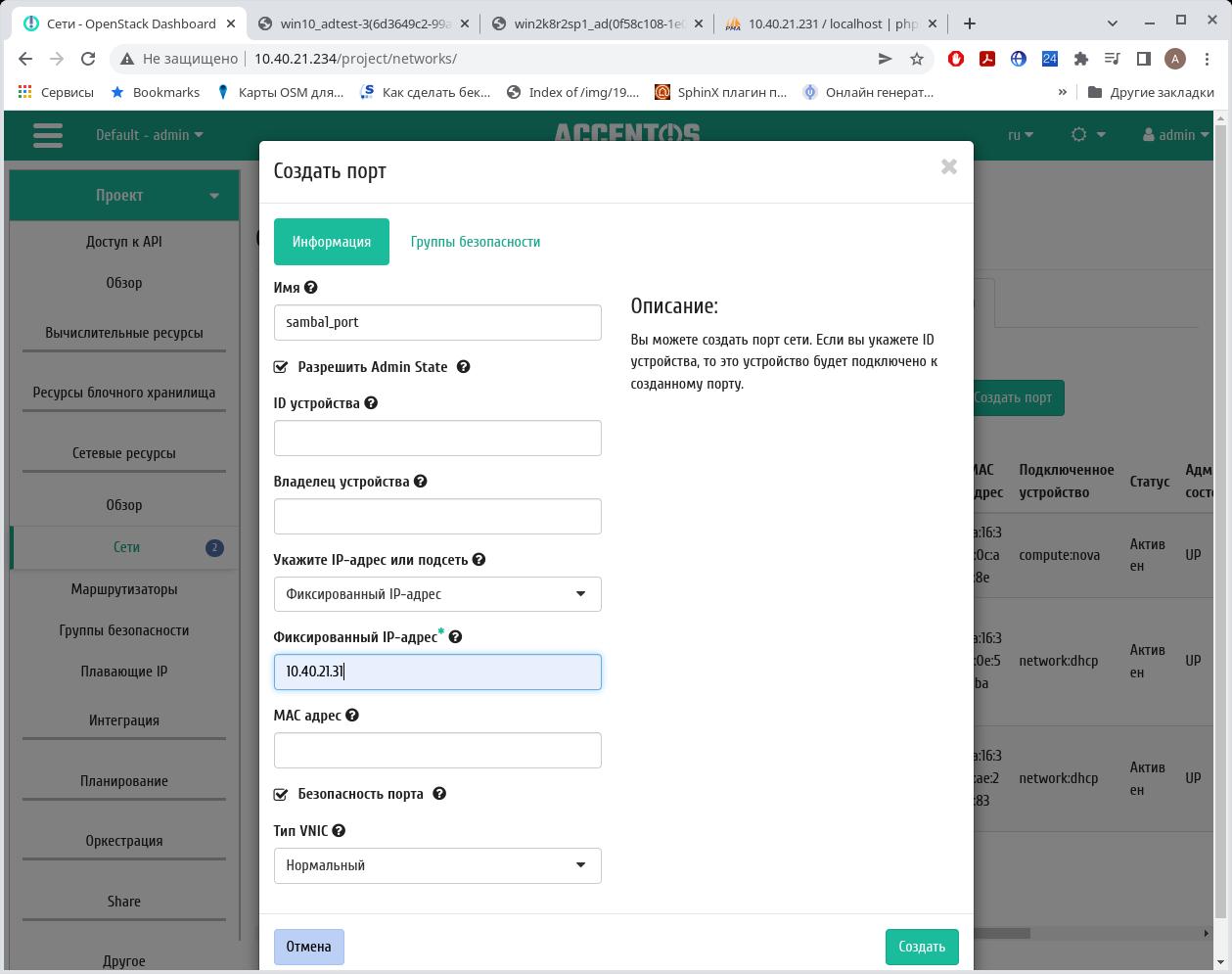
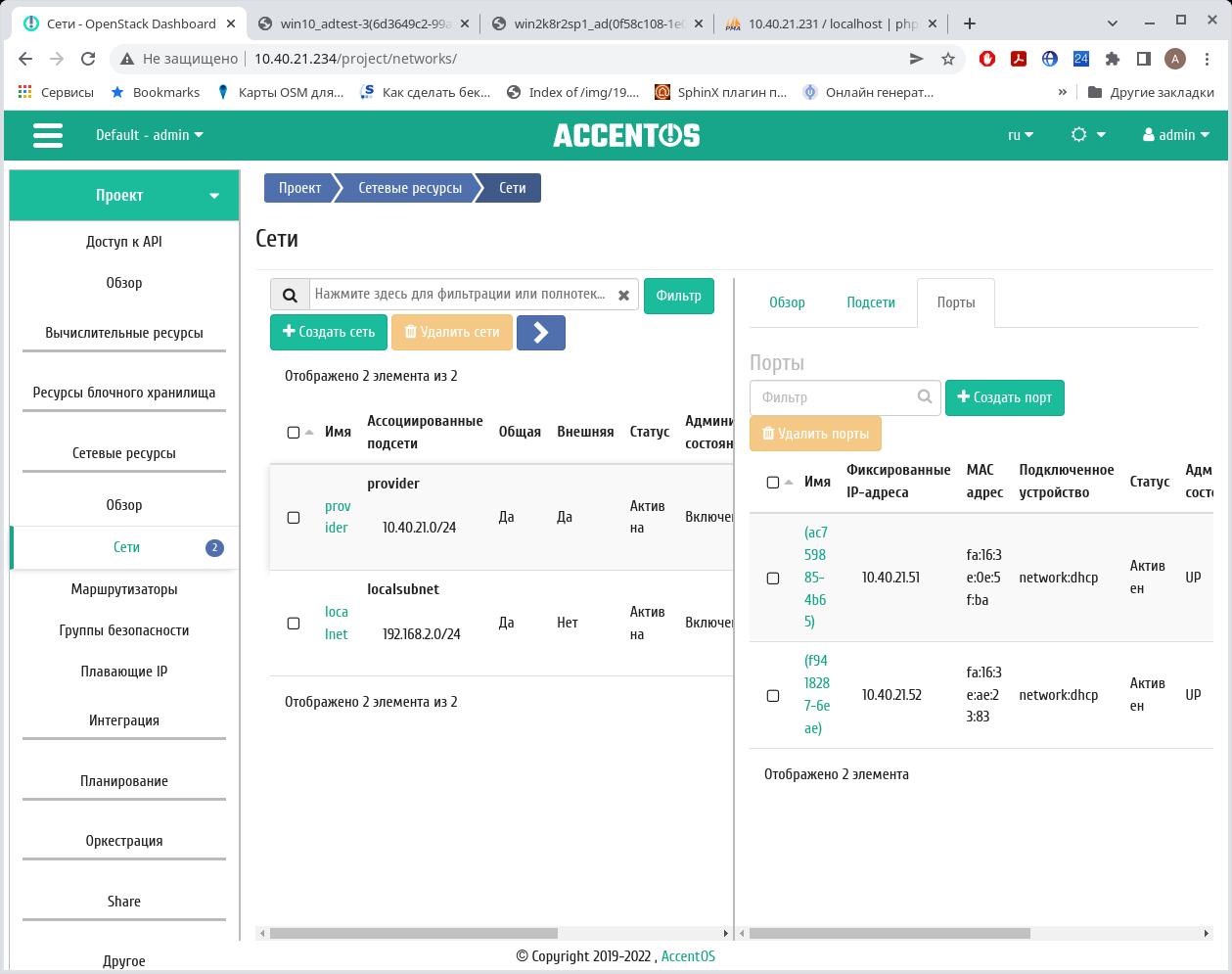


## 

## Создание сетевых портов

Подключение виртуальной машины к сети со статическим адресом реализуется через механизм предварительного создания сетевых портов. Для чего необходимо зайти в нужном проекте на страницу **Проект** -> **Сетевые ресурсы** -> **Сети** и кликнуть по названию сети, к которой должны быть подключены ВМ с Samba. В открывшемся фрейме необходимо перейти на вкладку **Порты** и создать порт с указанием требуемого статического адреса. Данные операции необходимо выполнить по числу планируемых к созданию виртуальных машин.

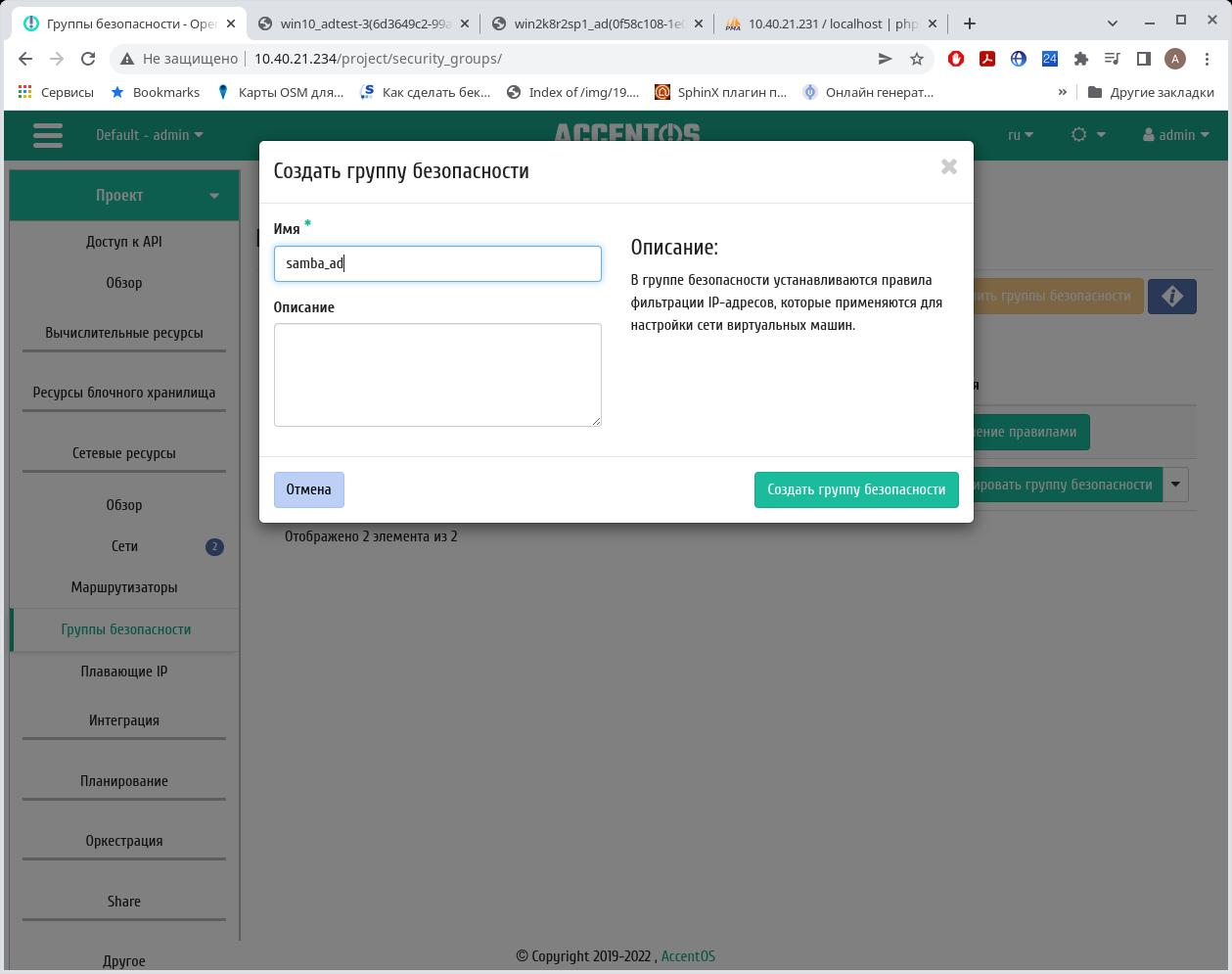
В нашем примере будут созданы 2 виртуальные машины с адресами 10.40.21.31 и 10.40.21.32. При большом количестве клиентских рабочих станций и соответственно значительной нагрузке рекомендуется создавать больше виртуальных машин для балансировки нагрузки и увеличения общей производительности.

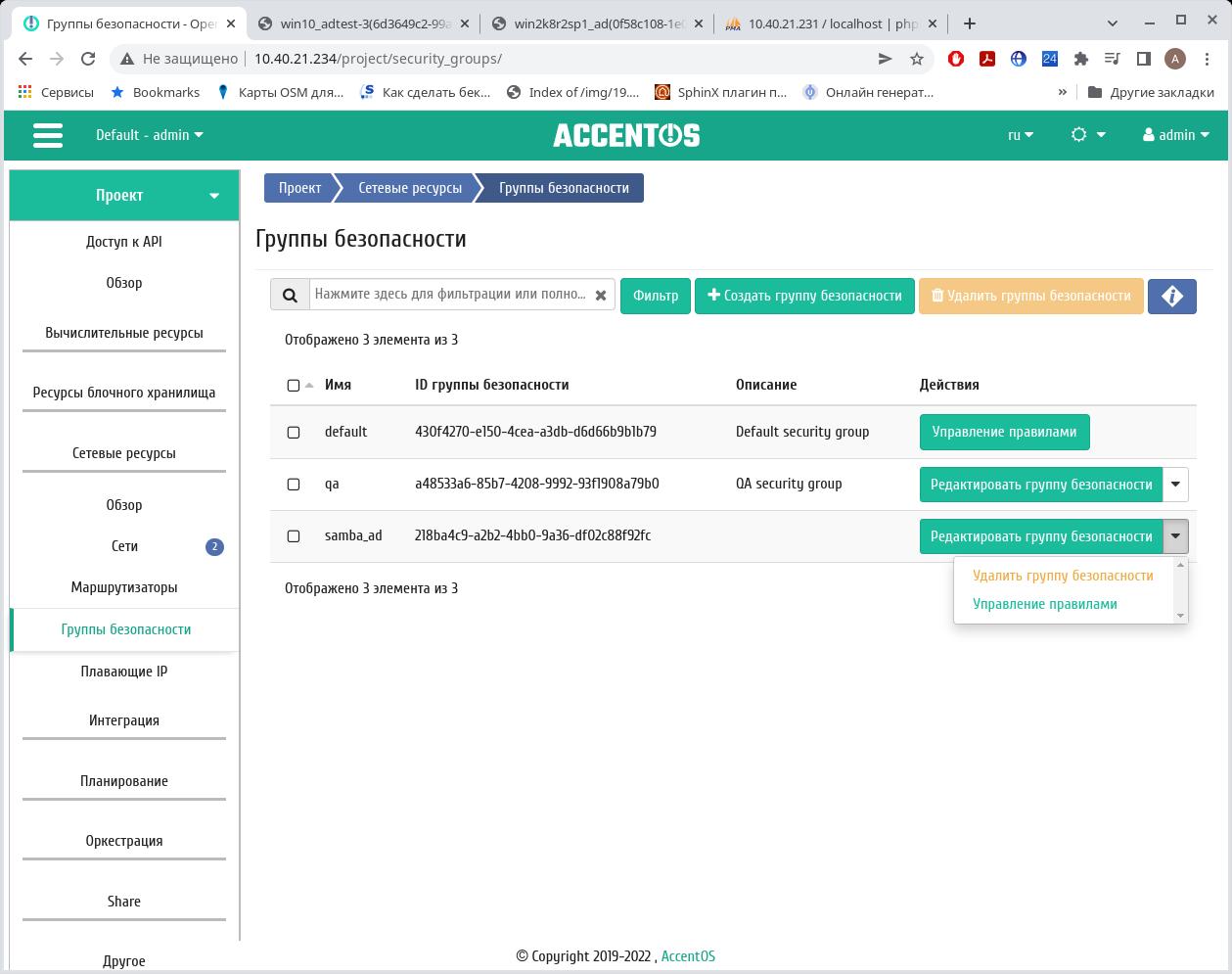


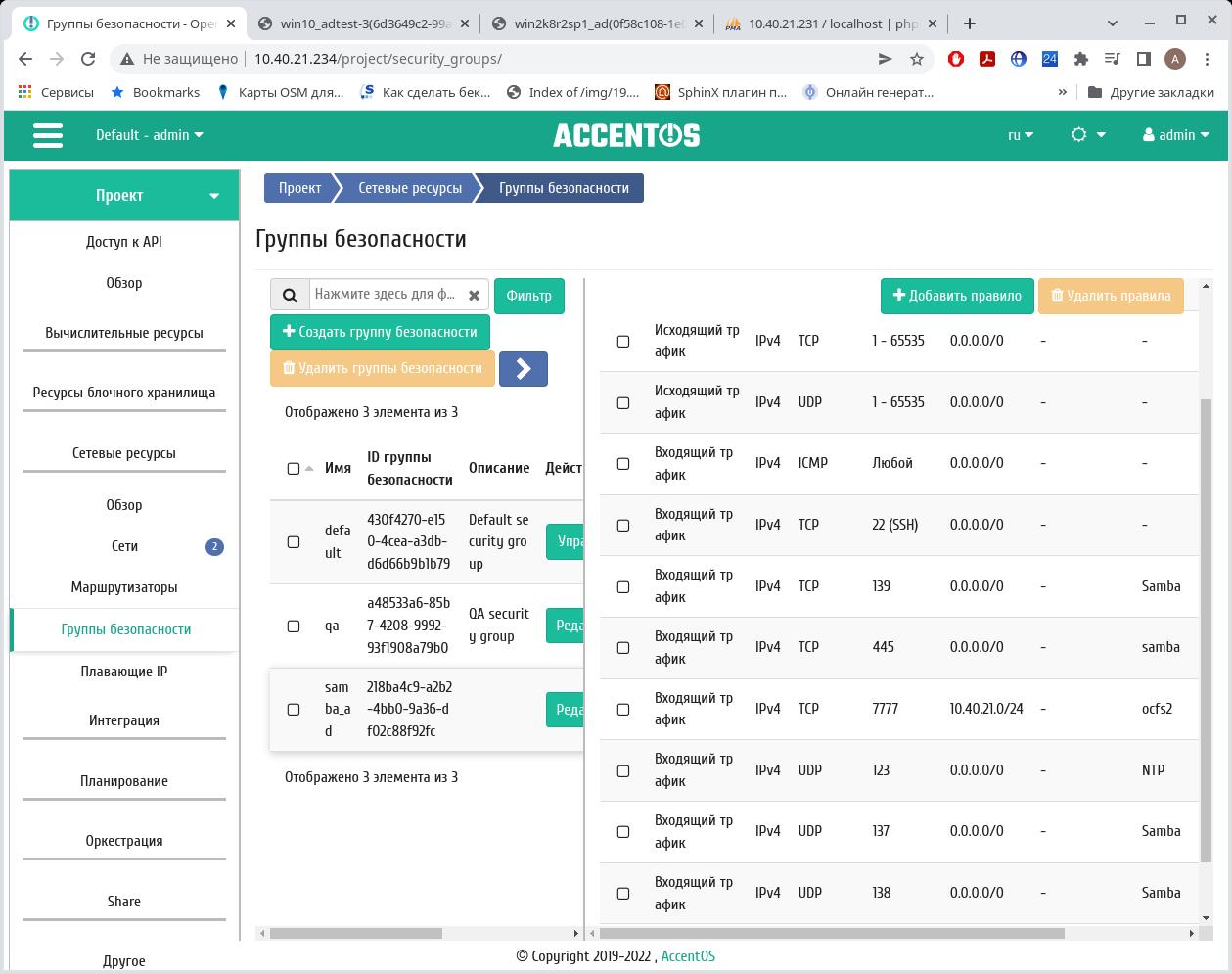
## Создание группы безопасности

Сетевая безопасность на уровне облака обеспечивается не файрволом в каждой виртуальной машине, а правилами сетевых разрешений на уровне облака, объединенных в группы безопасности. Это позволяет одновременно менять сетевые правила для всех однотипных виртуальных машин без настройки каждой ВМ в отдельности.  
  
Для создания отдельной группы безопасности необходимо перейти в требуемом проекте на страницу Проект -> Сетевые ресурсы -> Группы безопасности. После создания необходимо перейти в режим управлением правилами через выпадающее меню дополнительных действий для созданной группы безопасности.  
  
Для полноценной работы наших виртуальных машин мы должны обеспечить:

* разрешение для любого исходящего трафика по протоколам TCP, UDP и ICMP.
* разрешение входящего трафика по протоколу ICMP (определение доступности)
* разрешение доступа по SSH (22/TCP)
* разрешение доступа по NTP (123/UDP)
* разрешение работы кластера OCFS2 (7777/TCP) в пределах той сети, в которой будут размещены остальные вм, использующие общий диск для работы Samba
* разрешение работы службы Samba (137/UDP, 138/UDP, 139/TCP, 445/TCP)



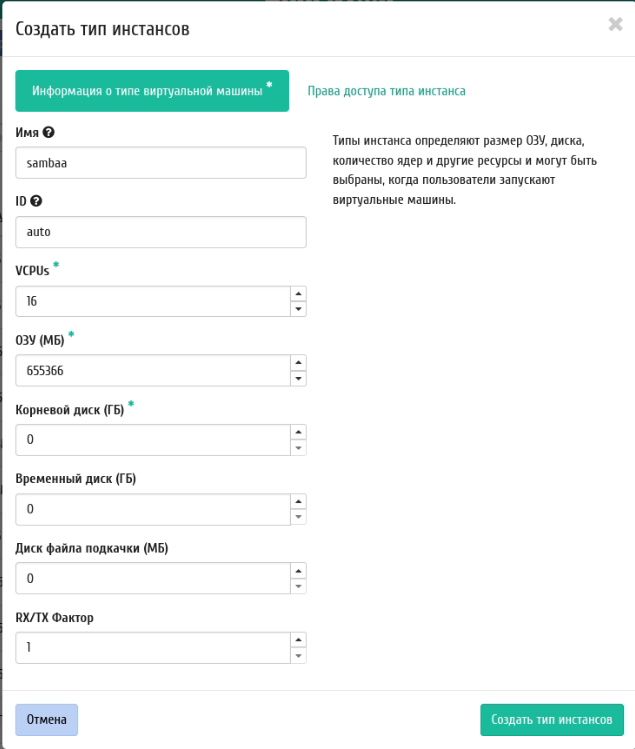




## Создание типа инстанса.

Создание типа инстанса доступно на странице Администратор -> Вычислительные ресурсы -> Виртуальные машины > Типы инстансов.Выбрать в правом верхнем углу «Создать тип инстанса».

Для корректной работы самбы необходимо создать следующий тип инстанса: 16vCPU, 64 ОЗУ, 0Гб Корневой диск, 0Гб Временный диск, 0мб диск файла подкачки, 1,0 RX/TX фактор.



## Создание виртуальных машин.

Создание виртуальных машин доступно на странице Проект -> Вычислительные ресурсы -> Виртуальные машины

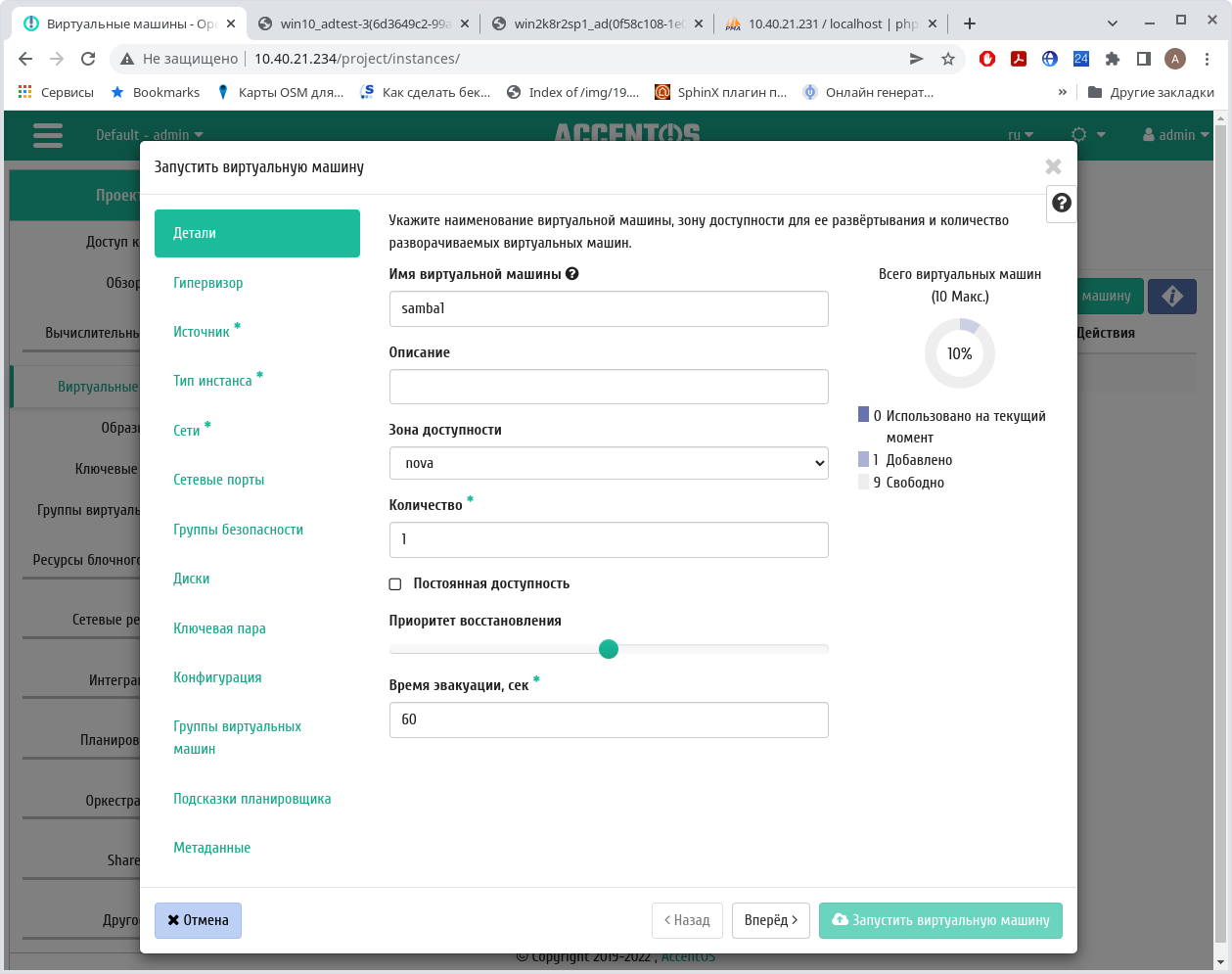
Для создания виртуальной машины или группы однотипных виртуальных машин необходимо заполнить несколько вкладок диалогового окна. Учитывая, что мы создаем виртуальные машины с предварительно созданными сетевыми портами - создавать каждую ВМ придется отдельно. Ниже представлены скриншоты пошагового заполнения указанного диалога.

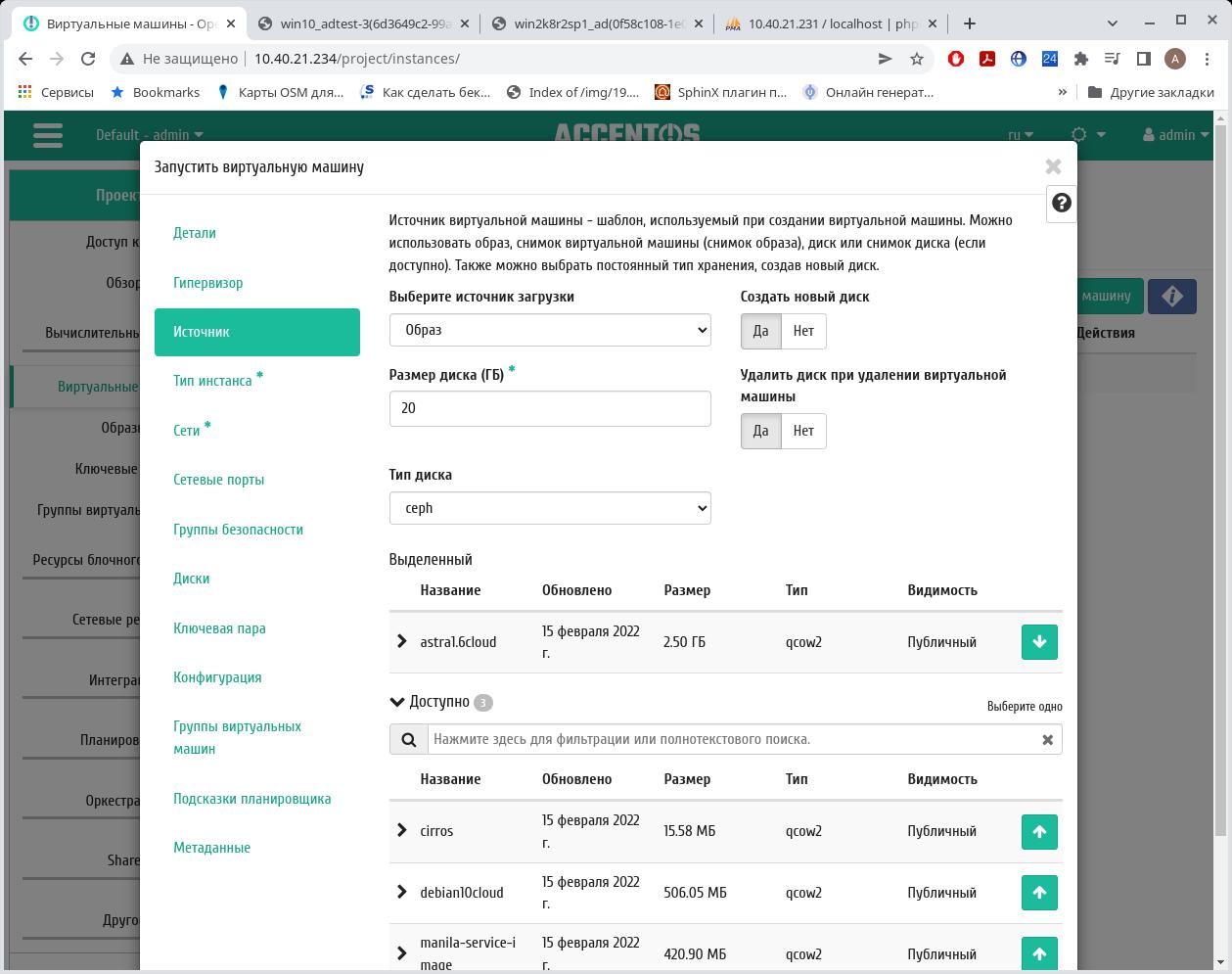
Системный диск виртуальной машины может быть быть создан как на любом типе cinder volume, так и без оного - на ephemeral диске nova. В нашем примере демонстрируется создание служебных ВМ на cinder volume.

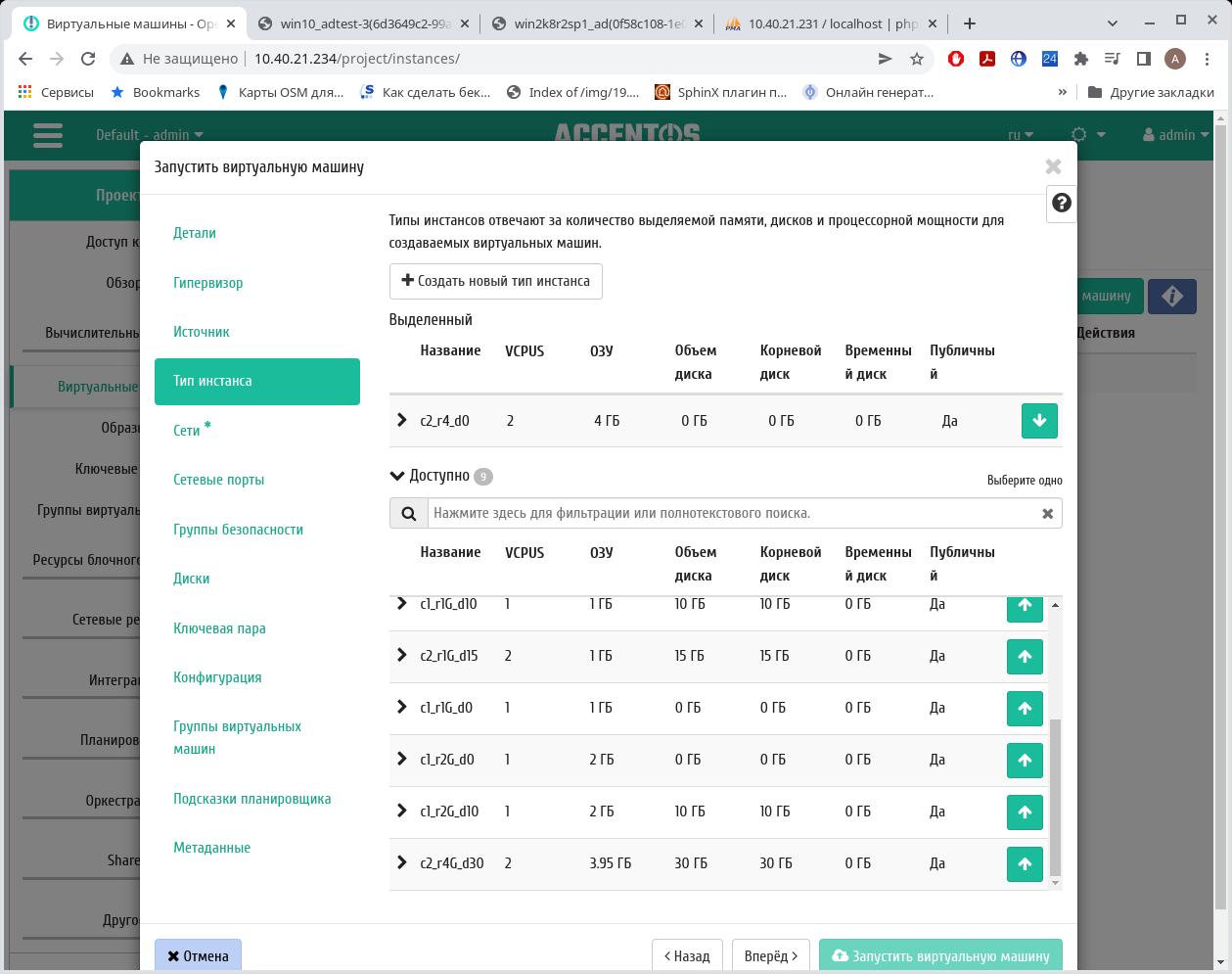
Проведенные тесты работы Samba показывают малую зависимость производительности Samba от выделенного объема оперативной памяти - главное, чтобы ее хватило для нормальной работы самой операционной системы. И наоборот - высокую зависимость производительности для большого числа пользователей от доступного числа ядер процессора. На этом и нужно основываться при выборе типа инстанса для виртуальных машин Samba. Выбор числа виртуальных процессоров также прямо зависит от количества клиентских машин, а также имеет обратную зависимость от числа ВМ с Samba, которые будут делить нагрузку между собой.

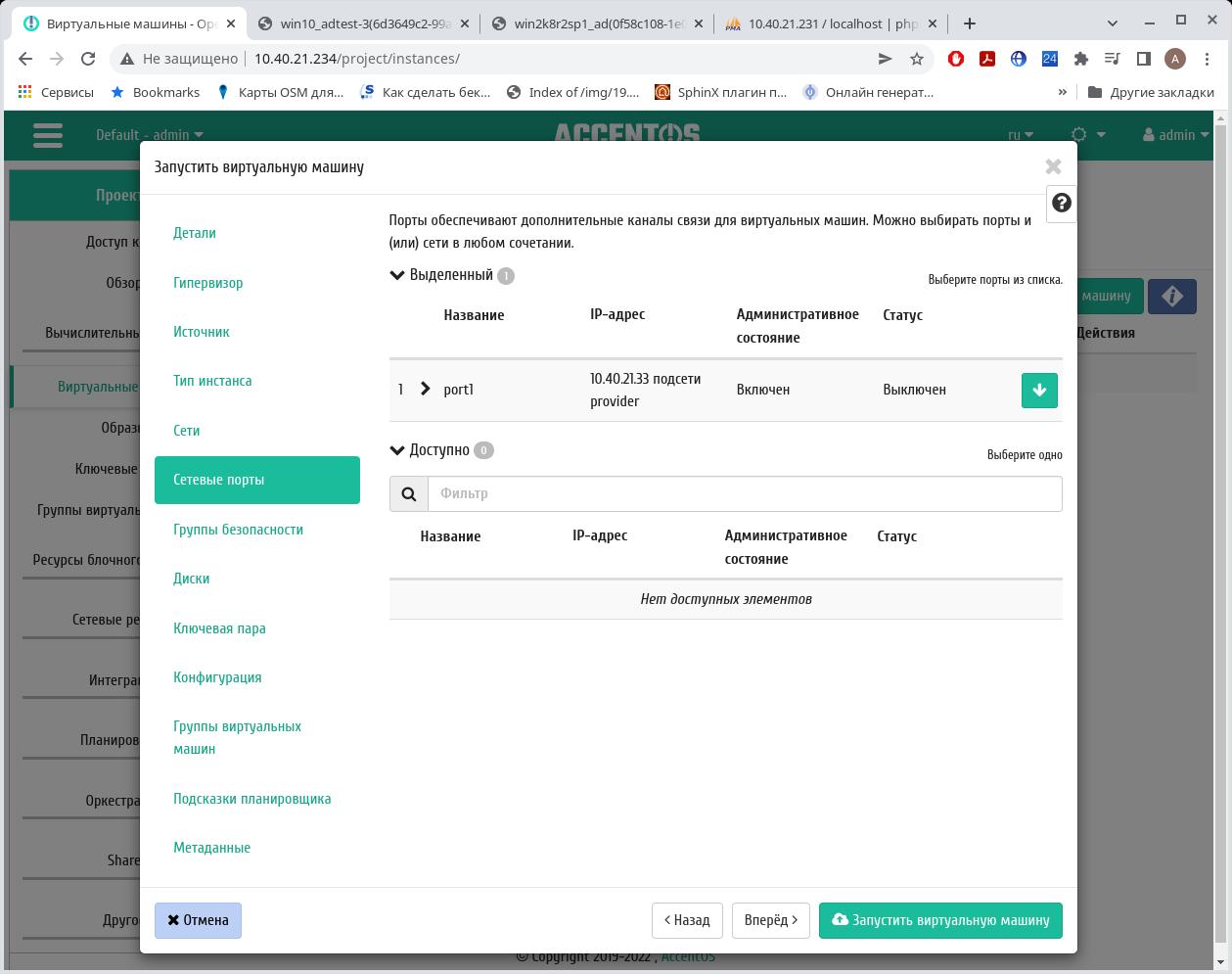
Вместо обычного выбора виртуальной сети, к которой будет подключена ВМ, мы выбираем предварительно созданный сетевой порт со статическим ip-адресом. В связи с чем в нашем примере отсутствует рассмотрение вкладки **Сети**.

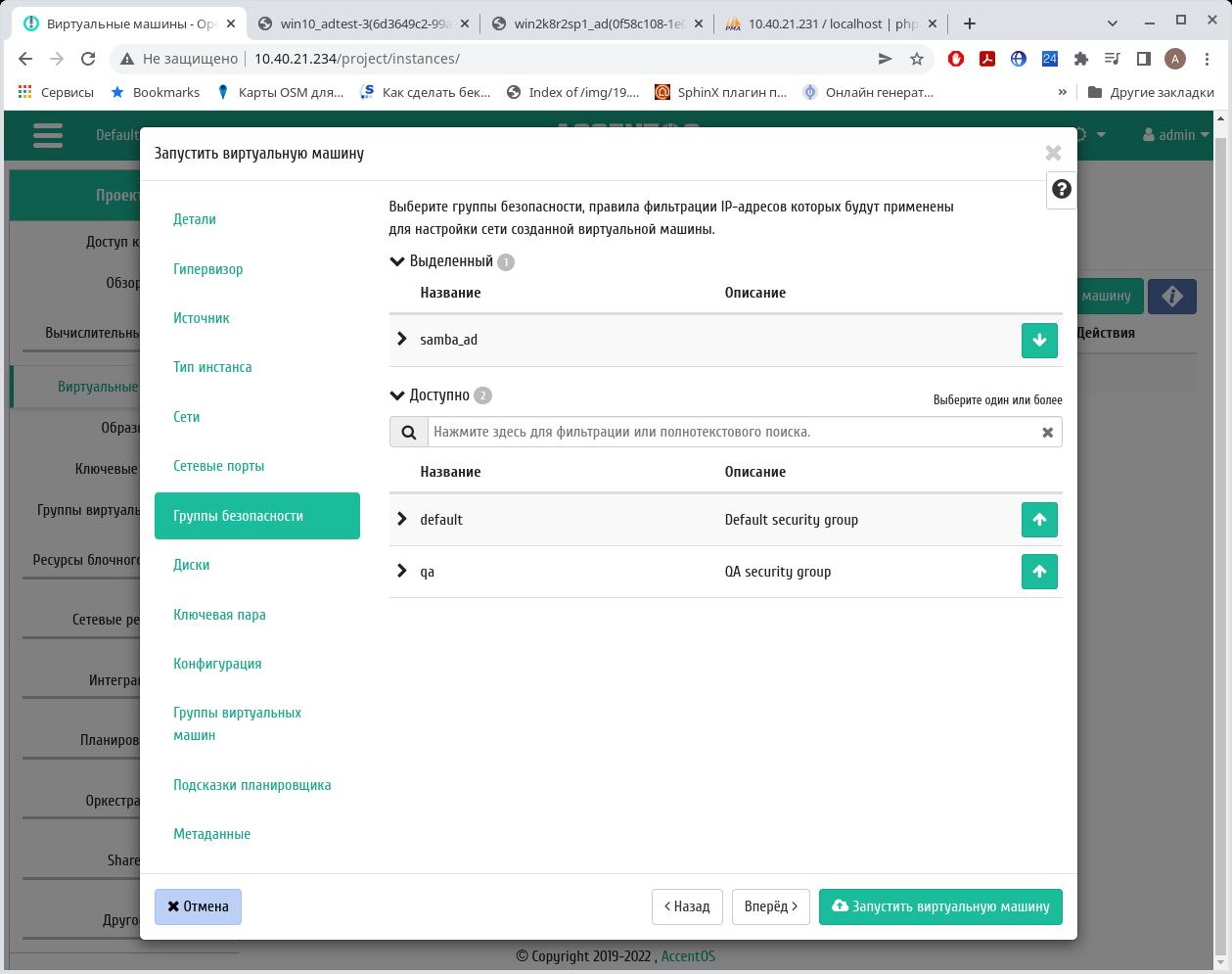
Мы также выбираем ключевую пару, от которой имеем секретный ключ, и ранее настроенную группу безопасности.

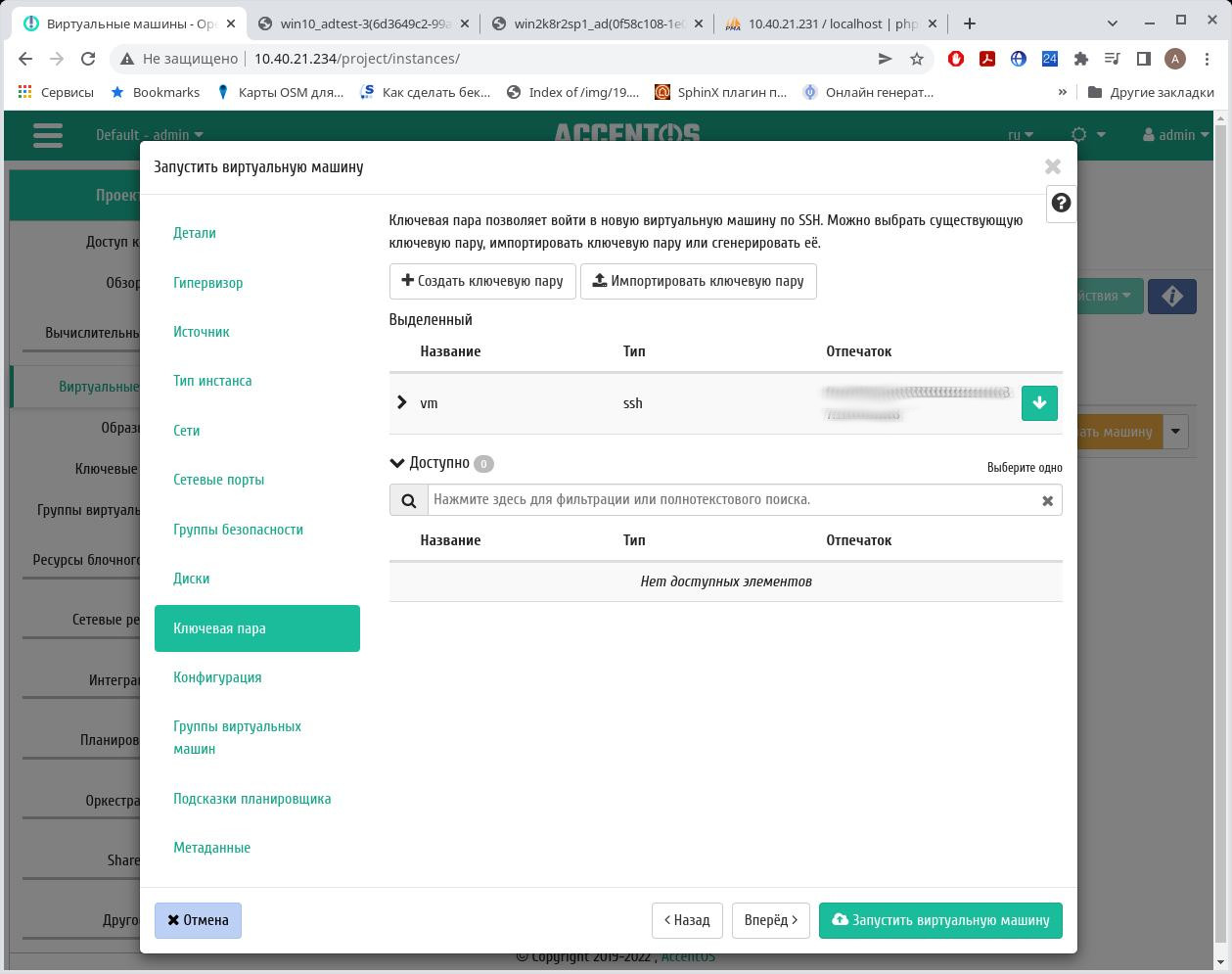


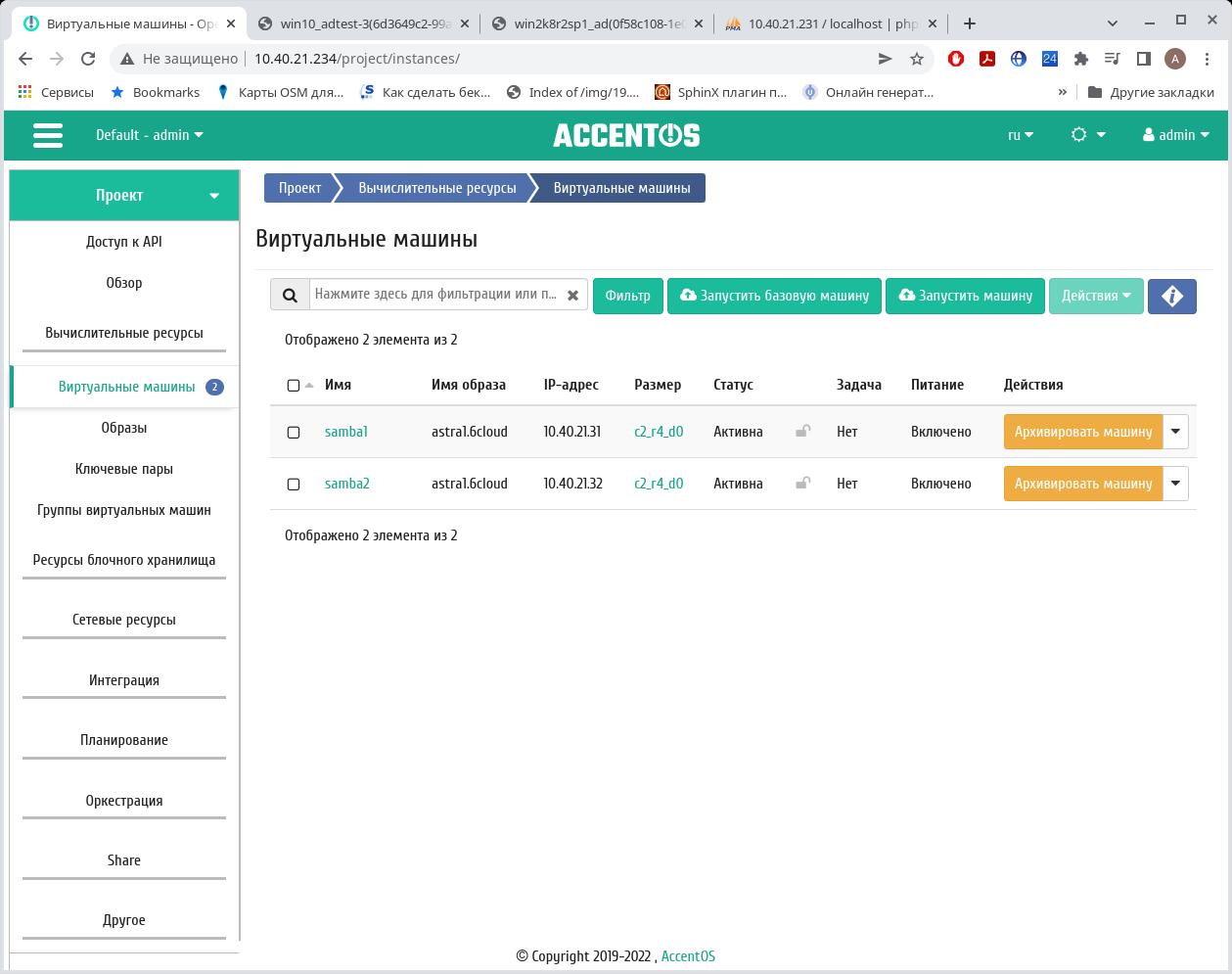








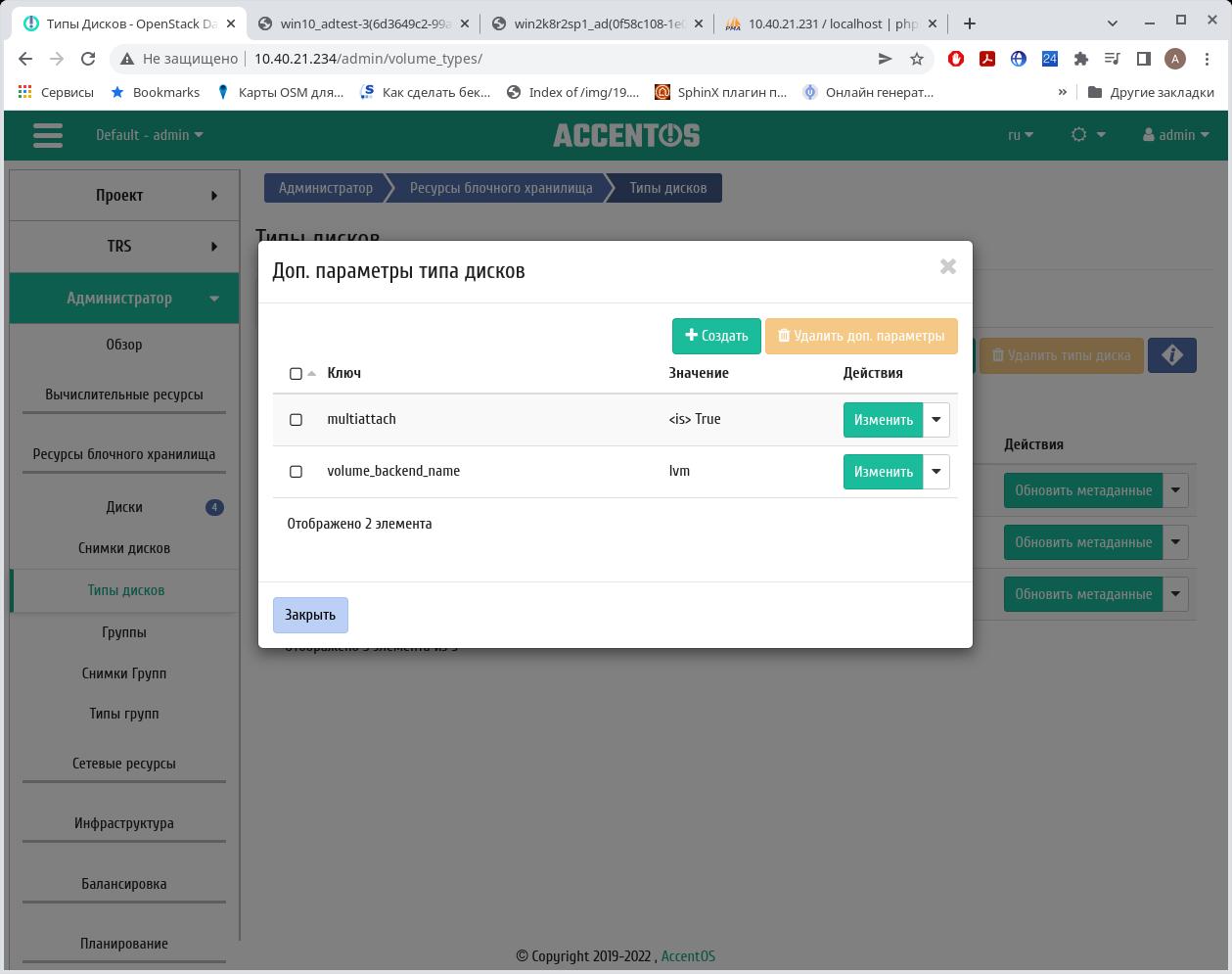


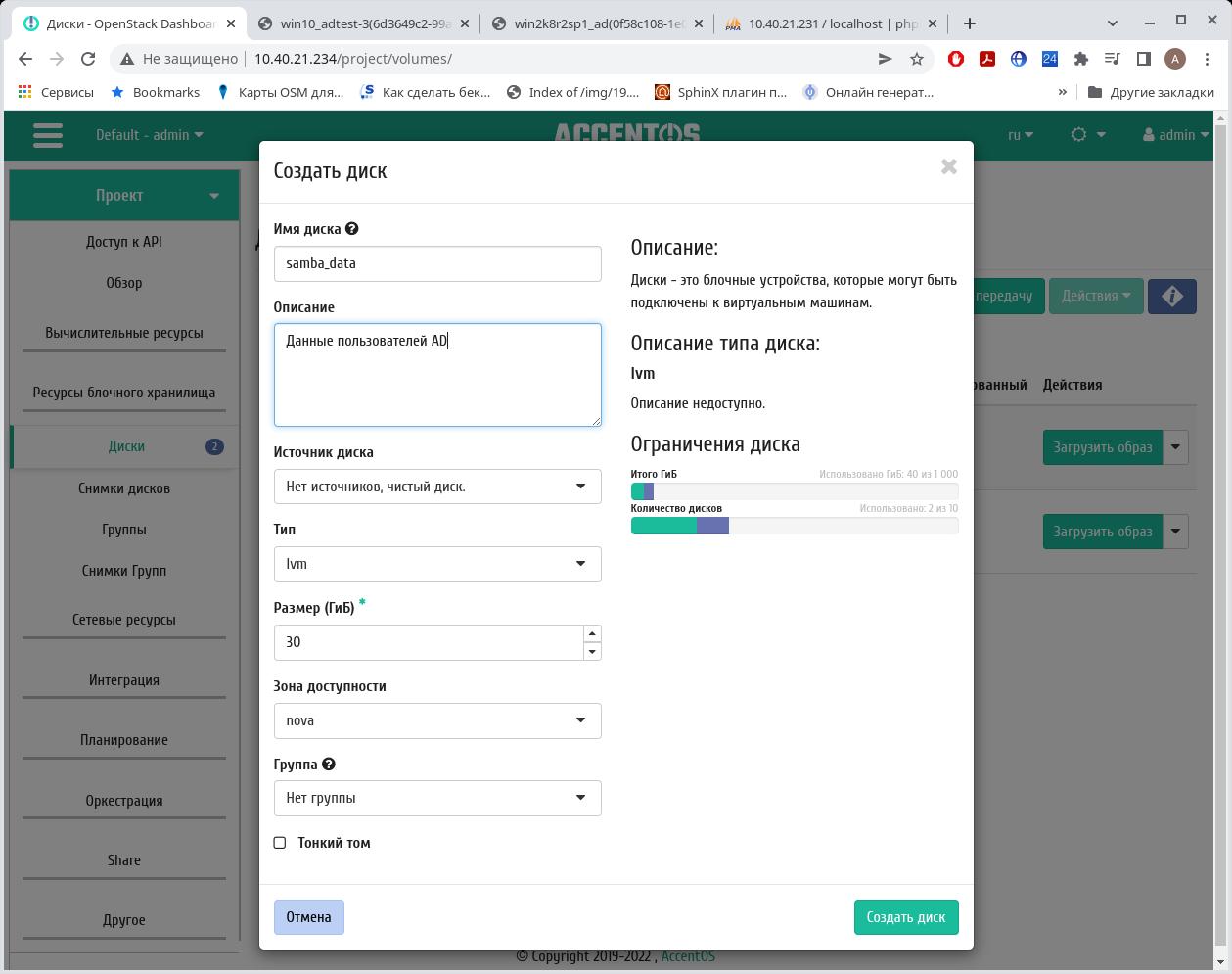


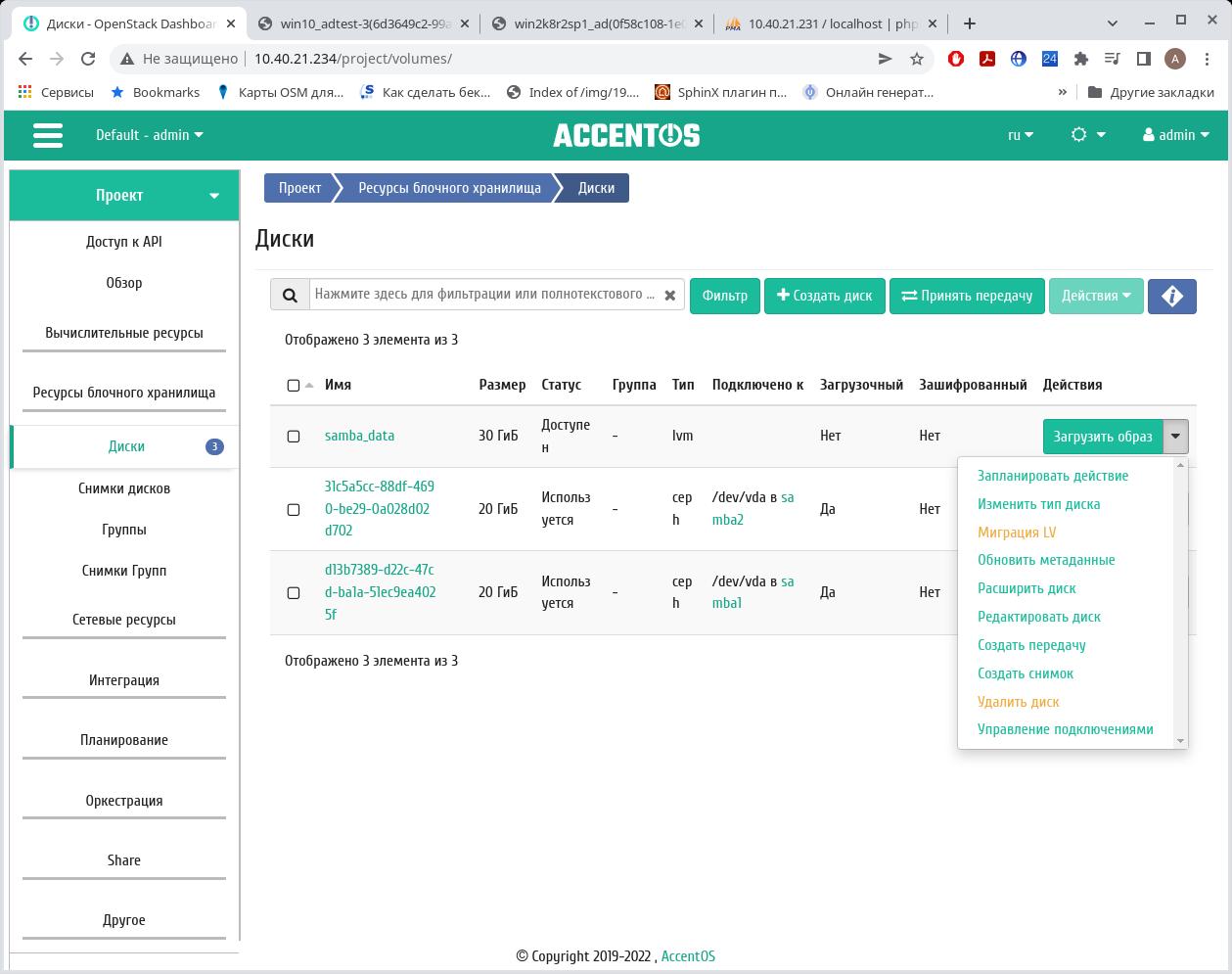
## Создание общего диска для Samba.

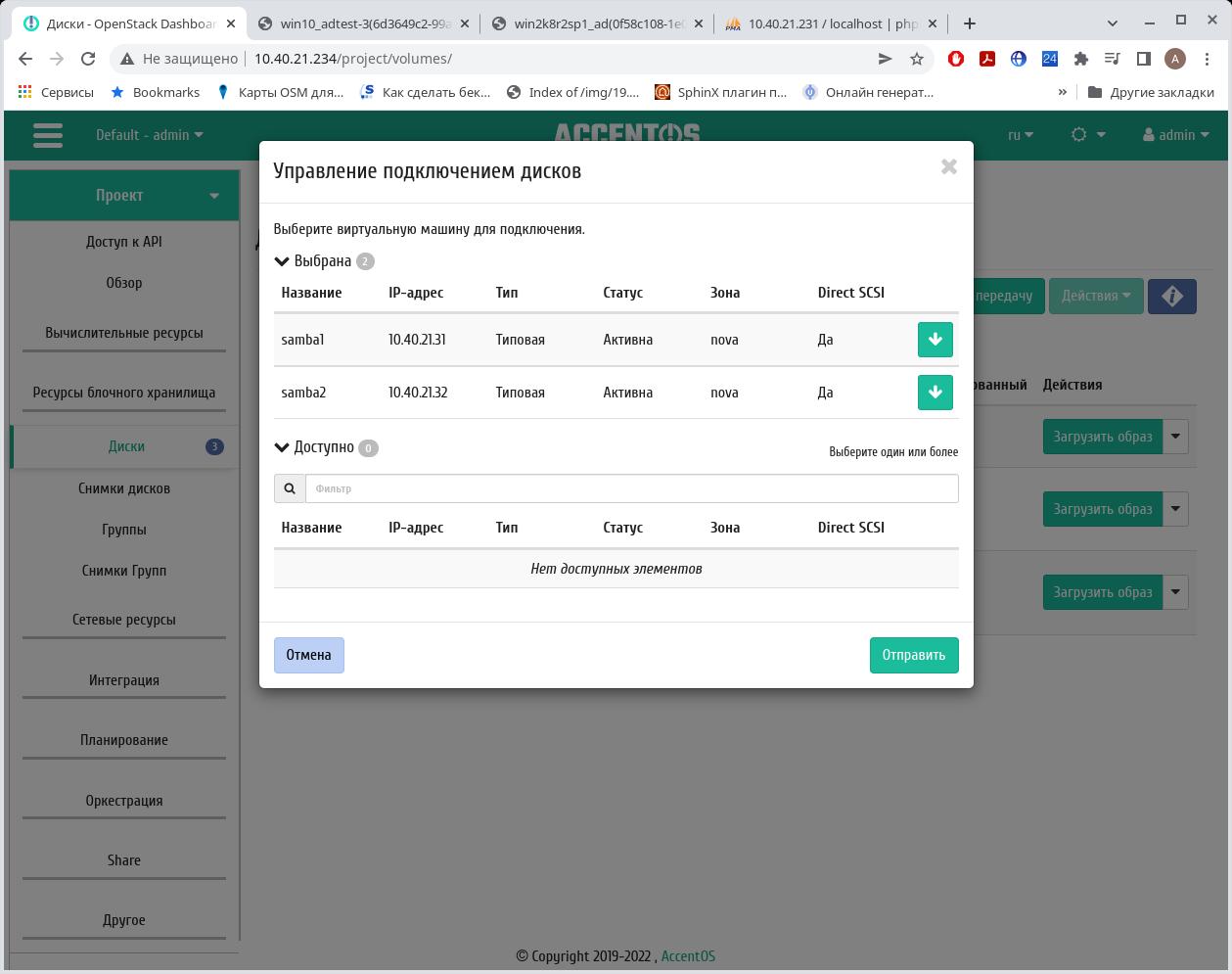
Работа служб Samba, работающих на разных ВМ и балансирующих нагрузку между собой, требует общего хранилища, в качестве которого мы выбираем отдельный виртуальный диск, который будет подключен ко всем созданным ВМ. Исходя из этого нам требуется тип диска cinder, поддерживающий множественное подключение. Такой тип блочных устройств в облаке должен иметь дополнительный параметр multiattach с установленным значением “<is> True”.  
  
Размер общего диска выбирается исходя из количества пользователей домена и среднего объема хранимых им данных: документов, кэша почты и прочих файлов, расположенных в папке профиля пользователя ОС Microsoft Windows.

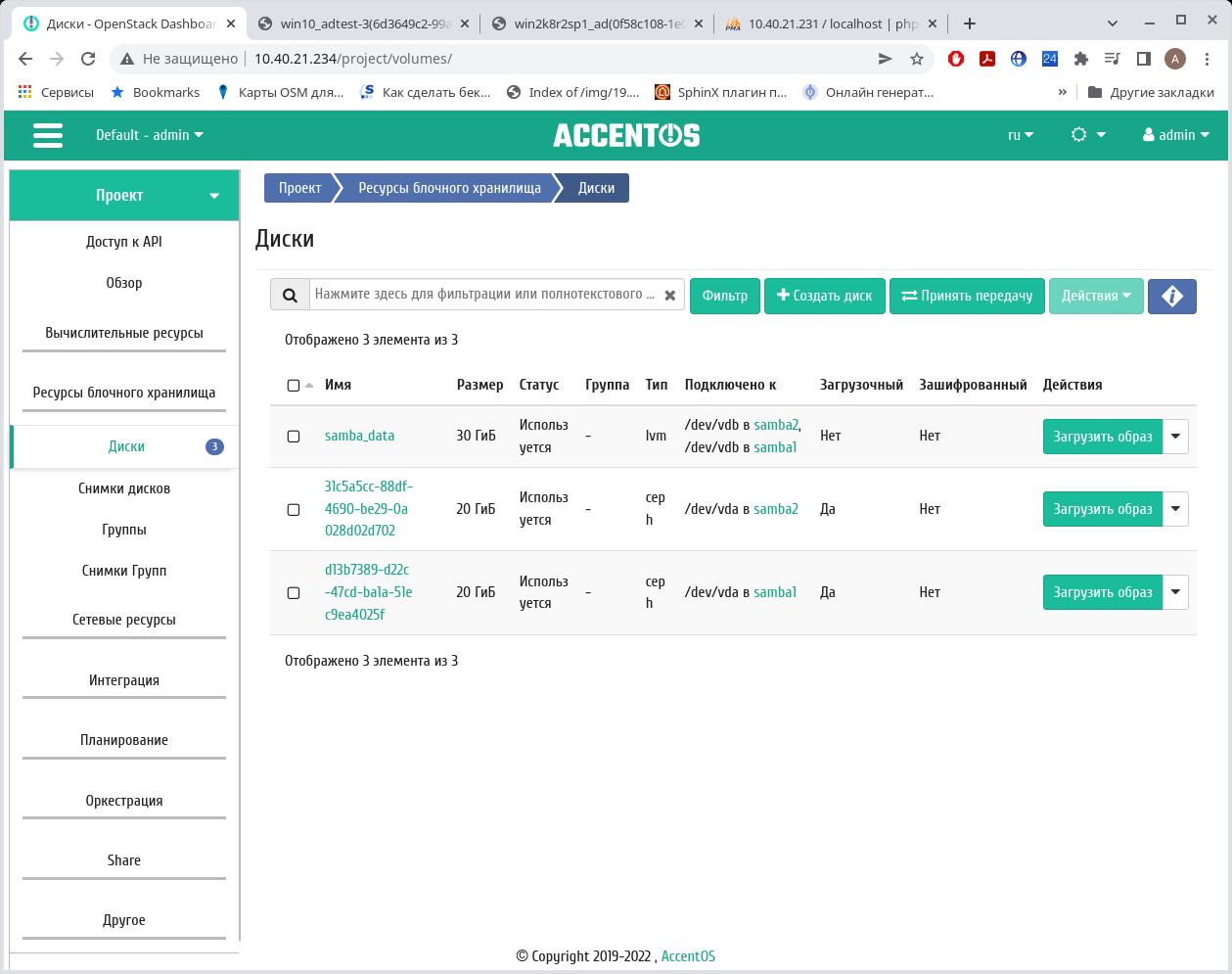
Завершением создания общего диска является подключение его ко всем виртуальным машинам нашей виртуальной инфраструктуры.











## Зайти под администратором:

Залогиниться в систему как:

astra

12345678

Работать из под администратора:

sudo -i

## Установка cloud-init

apt install cloud-init

Перезагрузить ВМ: reboot

Проверить, что hostname ВМ изменился на тот, что был указал в облачном интерфейсе при создании

## Установка chrony.

Для корректной работы с общим диском необходимо обеспечить синхронизацию времени. Установим chrony.

apt install chrony

Добавим сервера для обеспечения синхронизации. Для этого надо в файл *etc/chrony/chrony.conf добавить строку на каждый сервер:*

*server «IP сервера без кавычек» iburst*

*Перезагрузить службу:*

*systemctl restart chrony*

*Проверить синхронизацию:*

*chronyc sources*

## Запуск скрипта установки и настройки Samba

Перед переносом скрипта на все установленные ранее ВМ, правим в скрипте значения переменных.

Переменная с расположением хоста с репозиториями. Возможно ниже придется изменить строки с определением сетевых репозиториев в соответствии с настройками в конкретной сети.

***export REPO=10.40.129.21***

Переменная, включающая ip-адреса всех виртуальных машин, предназначенных для работы с общим диском. На первом узле на базе данного списка узлов формируется конфигурационный файл для работы OCFS2-кластера и распространяется на все узлы. Указание недоступного узла приведет к зависанию работы скрипта.

***export OCFS2\_IPS="10.40.21.31 10.40.21.32"***

Имя домена FreeIPA, к которому осуществляется подключение узлов Samba

***export DOMAIN=rzd.loc***

Адрес узла (ов), хранящих DNS-зону домена Microsoft Active Directory. Список указывается через пробел.

***export DOMAIN\_DNS\_SERVER="10.40.21.22"***

Имя пользователя, имеющего права на внесение узлов в домен FreeIPA.

export DOMAIN\_ADMIN='администратор'

Пароль пользователя, указанного выше.

***export DOMAIN\_ADMIN\_PASSWORD='@123qwe'***

URL для скачивания секретного ssh-ключа для доступа к виртуальным машинам. SSH используется скриптом для синхронизации установки и переноса конфигурационных данных.

***export SSH\_KEY\_URL=http://$REPO/stable/astra/1.6/sshkeys/vm***

Скачиваем и одновременно запускаем скрипты на всех вм

# **wget http://10.40.21.4/unstable/astra/1.6/install\_samba-manual -O install\_samba.sh**

# **chmod +x install\_samba.sh**

# **./install\_samba.sh**

После завершения работы скрипт выдаст данные указанного в переменных пользователя в виде  
**администратор:\*:1000500:1000513::/mnt/samba/homes/%S:/bin/false**

Указанные данные, выведенные на всех узлах должны быть идентичны, в противном случае скрипт отработал неправильно и доступ к одним и тем же данным через разные узлы Samba будет невозможен.

После перезагрузки виртуальных машин, службы Samba готовы к работе в составе домена.

Код скрипта  
*#!/bin/bash*

*export REPO=10.40.129.21*

*export OCFS2\_IPS="10.40.21.31 10.40.21.32"*

*export DOMAIN=rzd.loc*

*export DOMAIN\_DNS\_SERVER="10.40.21.22"*

*export DOMAIN\_ADMIN='администратор'*

*export DOMAIN\_ADMIN\_PASSWORD='@123qwe'*

*export SSH\_KEY\_URL=http://$REPO/stable/astra/1.6/sshkeys/vm*

*echo deb http://$REPO/smolensk smolensk contrib main non-free > /etc/apt/sources.list*

*echo deb http://$REPO/devel-smolensk smolensk contrib main non-free >> /etc/apt/sources.list*

*echo deb http://$REPO/update202111-smolensk smolensk contrib main non-free >> /etc/apt/sources.list*

*echo deb http://$REPO/devel-update202111-smolensk smolensk contrib main non-free >> /etc/apt/sources.list*

*mkdir -pv /root/.ssh*

*chmod 700 /root/.ssh*

*wget -O /root/.ssh/id\_rsa $SSH\_KEY\_URL*

*chmod 400 /root/.ssh/id\_rsa*

*ssh-keygen -y -f /root/.ssh/id\_rsa > /root/.ssh/authorized\_keys*

*cp -f /root/.ssh/id\_rsa /home/$SUDO\_USER/.ssh*

*chown /home/$SUDO\_USER/.ssh/id\_rsa*

*apt-get update*

*apt-get dist-upgrade -y*

*apt-get install linux-5.10 -y*

*export DOMAIN\_UP=${DOMAIN^^}*

*export DOMAIN\_SHORT=$(echo $DOMAIN\_UP | awk -F"." '{print $1}')*

*export OCFS2\_CLUSTER\_NAME=ocfs2sambaIPA*

*export IP=$(ip addr list br-ex | grep " inet " | head -n 1 | cut -d " " -f 6 | cut -d / -f 1)*

*if [ -z $IP ]; then*

*export IP=$(ip addr list eth0 | grep " inet " | head -n 1 | cut -d " " -f 6 | cut -d / -f 1)*

*fi*

*DEBIAN\_FRONTEND=noninteractive apt-get install -y samba winbind krb5-user libnss-winbind ocfs2-tools dlm-controld astra-freeipa-client libwbclient-sssd samba smbclient*

*if [ ! -z $DOMAIN\_DNS\_SERVER ]; then*

*cp -f /etc/resolv.conf /etc/resolv.conf.old*

*rm -f /etc/resolv.conf*

*for dnsserver in $DOMAIN\_DNS\_SERVER*

*do*

*echo "nameserver $dnsserver" >> /etc/resolv.conf*

*done*

*echo "search $DOMAIN*

*domain $DOMAIN*

*" >> /etc/resolv.conf*

*fi*

*HOSTNAME=$(hostname)*

*echo "supersede host-name \"$HOSTNAME.$DOMAIN\";" >> /etc/dhcp/dhclient.conf*

*echo "supersede domain-name \"$DOMAIN\";" >> /etc/dhcp/dhclient.conf*

*echo "supersede domain-name-servers \"$DOMAIN\_DNS\_SERVER\";" >> /etc/dhcp/dhclient.conf*

*hostnamectl set-hostname $HOSTNAME.$DOMAIN*

*#задержка нужна для обновления имени системы.*

*sleep 60*

*modprobe ocfs2\_dlmfs*

*modprobe ocfs2\_stack\_o2cb*

*modprobe ocfs2\_dlm*

*modprobe ocfs2\_nodemanager*

*echo "ocfs2*

*ocfs2\_dlmfs*

*ocfs2\_stack\_o2cb*

*ocfs2\_dlm*

*ocfs2\_nodemanager*

*" > /etc/modules-load.d/ocfs.conf*

*i=0*

*for ocfs2\_ip in $OCFS2\_IPS*

*do*

*if [ "$i" = "0" ]; then*

*masterip=$ocfs2\_ip*

*fi*

*if [ "$ocfs2\_ip" = "$IP" ] && [ $i -eq "0" ]; then*

*echo wait > /etc/ocfs2/ocfs2.status*

*#ожидаем готовности остальных узлов*

*for ip in $OCFS2\_IPS*

*do*

*if [ ! "$ip" = "$IP" ]; then*

*status=$(ssh -oStrictHostKeyChecking=no root@$ip "cat /etc/ocfs2/ocfs2.status")*

*while [ ! $status = "wait" ]*

*do*

*sleep 10*

*status=$(ssh -oStrictHostKeyChecking=no root@$ip "cat /etc/ocfs2/ocfs2.status")*

*done*

*fi*

*done*

*#создаем кластер ocfs2*

*o2cb add-cluster $OCFS2\_CLUSTER\_NAME*

*for ip in $OCFS2\_IPS*

*do*

*hstnm=$(ssh -oStrictHostKeyChecking=no root@$ip "cat /etc/hostname")*

*while [ "a$hstnm" = "a" ]*

*do*

*sleep 5*

*hstnm=$(ssh -oStrictHostKeyChecking=no root@$ip "cat /etc/hostname")*

*done*

*echo $ip $hstnm >> /etc/hosts*

*echo $ip $hstnm >> /etc/ocfs2/hosts*

*o2cb add-node $OCFS2\_CLUSTER\_NAME $hstnm --ip $ip*

*done*

*o2cb register-cluster $OCFS2\_CLUSTER\_NAME*

*o2cb start-heartbeat $OCFS2\_CLUSTER\_NAME*

*mkfs.ocfs2 --cluster-stack=o2cb --cluster-name=$OCFS2\_CLUSTER\_NAME --fs-features=xattr /dev/vdb*

*echo create > /etc/ocfs2/ocfs2.status*

*fi*

*if [ "$ocfs2\_ip" = "$IP" ] && [ $i -gt "0" ]; then*

*echo wait > /etc/ocfs2/ocfs2.status*

*status=$(ssh -oStrictHostKeyChecking=no root@$masterip "cat /etc/ocfs2/ocfs2.status")*

*while [ -z $status ] || [ ! $status = "create" ]*

*do*

*echo status $status*

*sleep 10*

*status=$(ssh -oStrictHostKeyChecking=no root@$masterip "cat /etc/ocfs2/ocfs2.status")*

*done*

*echo status $status*

*scp root@$masterip:/etc/ocfs2/hosts /etc/ocfs2*

*cat /etc/ocfs2/hosts >> /etc/hosts*

*scp $masterip:/etc/ocfs2/cluster.conf /etc/ocfs2*

*echo create > /etc/ocfs2/ocfs2.status*

*fi*

*let i=$i+1*

*done*

*sleep 20*

*sed -i "s/^O2CB\_ENABLED.\*/O2CB\_ENABLED=\"true\"/" /etc/default/o2cb*

*sed -i "s/^O2CB\_BOOTCLUSTER.\*/O2CB\_BOOTCLUSTER=\"$OCFS2\_CLUSTER\_NAME\"/" /etc/default/o2cb*

*DEBIAN\_FRONTEND=noninteractive dpkg-reconfigure ocfs2-tools*

*sed -i "s/^O2CB\_ENABLED.\*/O2CB\_ENABLED=\"true\"/" /etc/default/o2cb*

*sed -i "s/^O2CB\_BOOTCLUSTER.\*/O2CB\_BOOTCLUSTER=\"$OCFS2\_CLUSTER\_NAME\"/" /etc/default/o2cb*

*sync*

*systemctl restart o2cb*

*echo > /etc/ocfs2/cluster.conf.done*

*cat /etc/ocfs2/cluster.conf*

*o2cb cluster-status*

*mounted.ocfs2 -f*

*str=$(blkid /dev/vdb)*

*for i in $str*

*do*

*param=$(echo $i | awk -F"=" '{print $1}')*

*if [ $param = "UUID" ]; then*

*export samba\_uuid=$(echo $i | awk -F"=" '{print $2}' | sed 's/"//g')*

*break*

*fi*

*done*

*mkdir -pv /mnt/samba*

*chmod 755 /mnt/samba*

*echo "UUID=$samba\_uuid /mnt/samba ocfs2 \_netdev,x-systemd.requires=o2cb.service 0 0" >> /etc/fstab*

*mount -a -v*

*count=$(mount | grep /mnt/samba | wc -l)*

*while [ $count -eq 0 ]*

*do*

*sleep 10*

*mount -av*

*count=$(mount | grep /mnt/samba | wc -l)*

*done*

*mounted.ocfs2 -f*

*if [ ! -e /mnt/samba/homes/\$1 ]; then*

*mkdir -p /mnt/samba/homes/\$1*

*DOMAIN\_GROUP=\$(wbinfo --gid-info \$(wbinfo -i \$1 | awk -F":" '{print \$4}') | awk -F":" '{print \$1}')*

*chown \$1:"\$DOMAIN\_GROUP" /mnt/samba/homes/\$1*

*chmod 0700 /mnt/samba/homes/\$1*

*fi*

*exit 0*

*EOF*

*chown root:root /usr/local/sbin/mkhomedir.sh*

*chmod u=rwsx,g=rwx,o-rwx /usr/local/sbin/mkhomedir.sh*

*cp /etc/samba/smb.conf /etc/samba/smb.conf.old*

cat > /etc/samba/smb.conf << EOF

[global]

workgroup = CRP

security = user

# realm = CRP.RZD

winbind separator = +

idmap config \*:backend = tdb

idmap config \*:range = 100000-999999

idmap config $DOMAIN\_SHORT:backend = rid

idmap config $DOMAIN\_SHORT:range = 1000000-1999999

winbind use default domain = yes

winbind enum users = yes

winbind enum groups = yes

map acl inherit = Yes

# template homedir = /mnt/samba/homes/%S

# obey pam restrictions = yes

fileid:algorithm = fsid

vfs objects = fileid

#usershare allow guests = Yes

printcap name = /dev/null

#guest account = nobody

map to guest = bad user

[homes]

path = /mnt/samba/homes/

read only = no

# root preexec = /usr/local/sbin/mkhomedir.sh %S

browseable = yes

create mask = 0777

directory mask = 0777

available = yes

browsable = yes

writable = yes

public = yes

force user = nobody

guest ok =yes

guest only = yes

guest account = nobody

browsable = yes

[profiles$]

path = /mnt/samba/profiles

read only = no

valid users = user, user1, user2, user3, user4, user5, user6, user7, user8, user9, user10, user11

force group = users

create mask = 0777

directory mask = 0777

writable = yes

[share]

path = /mnt/samba/share

valid users = user, user1, user2, user3, user4, user5, user6, user7, user8, user9, user10, user11

force group = users

create mask = 0777

directory mask = 0777

writable = yes

browsable = yes

EOF

*mkdir -pv /mnt/samba/homes*

*mkdir -pv /mnt/samba/profiles*

*mkdir -pv /etc/systemd/system/smbd.service.d*

*cat > /etc/systemd/system/smbd.service.d/limits.conf <<TXT*

*[Service]*

*LimitNOFILE=500000*

*LimitNPROC=500000*

*TXT*

*cat /etc/security/limits.d/new\_limits.conf* <<TXT

*\* hard nofile 1000000*

*\* soft nofile 500000*

*\* hard nproc 1000000*

*\* soft nproc 500000*

*root hard nofile 1000000*

*root soft nofile 500000*

*root hard nproc 1000000*

*root soft nproc 500000*

*TXT*

Пост-скриптовые процедуры (выполнить на каждом из серверов):

1. Ввод ВМ в домен

astra-freeipa-client -d domain\_name -u user

2. Создать локальных пользователей одноименных доменным (ТОЛЬКО \*\*\*\_virazh\_admin, \*\*\*\*\_virazh\_user\* и т.п.):

useradd pvrr\_virazh\_admin // создаём пользователя

passwd pvrr\_virazh\_admin // задаём пароль. Пароль задавать идентичный одноимённому доменному

3. Импорт пользователя в samba:

smbpasswd -a pvrr\_virazh\_admin

Указать пароль из п.2.

Пост-скриптовые процедуры для кластера (выполнить на ОДНОМ ИЗ любых серверов!!!!)

1. В каталоге mnt/samba/homes/ создать каталоги для каждого пользователя из прошлого раздела (\*\*\*\_virazh\_admin, \*\*\*\*\_virazh\_user\*)  
  
mkdir -p pvrr\_virazh\_admin

2. Дат на каждый каталог полные права и nobody/nogroup

sudo chown nobody:nogroup /mnt/samba/homes

chmod 777 /mnt/samba/homes

chown nobody:nogroup /mnt/samba/homes/pvrr\_virazh\_admin

chmod 777 /mnt/samba/homes/pvrr\_virazh\_admin

Перезагрузить службы самбы на каждом из серверов

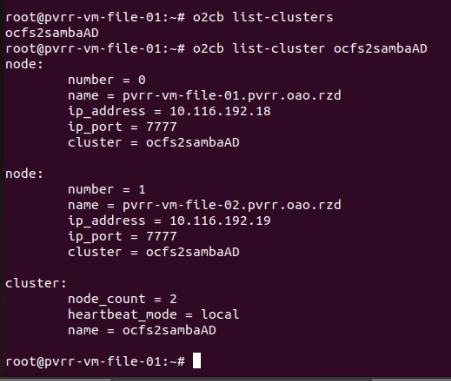
systemctl restart smbd nmbd

Критерии проверки установки:

1. Кластер установлен и содержит 2 ноды:

o2cb list-cluster

o2cb list-cluster



2. Машина вошла в домен и получает список пользователей.

Astra-freeipa-client -i

3. На гостевой виртуальной машины, которая введена в FreeIPA домен, можно примонтировать samba каталог. На гостевой ВМ выполнить в терминале следующее:

*su root //команда запросит пароль*

*export user=имя доменного пользователя*

*export pass=пароль доменного пользователя*

*export server=адрес сервера*

*/sbin/mount.cifs //$server/*[*homes/$user/*](https://10.246.192.20/homes/$user/) */home/$user/Desktop/share/ -o user=$user,password=$pass,uid=`id -u $user`,gid=`id -g $user`*

4. Проверить примонтирование:

df -h