Выполнение работ по установке облака AccentOS

**Оглавление**

[Состав оборудования и документации 3](#_Toc115295628)

[Установка облака 5](#_Toc115295629)

[Подготовка файла конфигурации для системы автоустановки 6](#_Toc115295630)

[Проверка корректности настроек перед автоустановкой облака 7](#_Toc115295631)

[Автоустановка облака 12](#_Toc115295632)

[Загрузка ВМ для системы WebGard и системы Samba 14](#_Toc115295633)

[Установка и настройка ВМ для системы WebGard 15](#_Toc115295634)

[Установка и настройка пВМ для системы Samba 16](#_Toc115295635)

[Настройка брокеров VDI 17](#_Toc115295636)

[Загрузка образов гостевых ВМ для системы Windows и Linux 18](#_Toc115295637)

[Создание проектов для АРМ Windows и АРМ Linux 19](#_Toc115295638)

[Установка ОС Windows на АРМ пользователя и настройка терминального клиента, настройка RSClient 20](#_Toc115295639)

[Установка ОС Linux на АРМ пользователя и настройка терминального клиента, настройка RSClient 21](#_Toc115295640)

[Проверка подлючения АРМ Windows 22](#_Toc115295641)

[Проверка подлючения АРМ Linux 23](#_Toc115295642)

[Настройка Zabbix 25](#_Toc115295643)

[Настройка NetBackup 26](#_Toc115295644)

[Настройка Kaspersky 27](#_Toc115295645)

[Прохождение ПСИ для платформы AccentOS 28](#_Toc115295646)

[Прохождение ПСИ для АРМ Windows 29](#_Toc115295647)

[Прохождение ПСИ для АРМ Linux 30](#_Toc115295648)

[Работа с репозиториями 31](#_Toc115295649)

[Импорт GPG-ключей для авторизации репозитория пакетов ПО 31](#_Toc115295650)

[Формирование файла /etc/apt/sources.list с определениями расположения репозиториев 31](#_Toc115295651)

[Отключение интерактивная установки пакетов debian 32](#_Toc115295652)

[Установка часового пояса 32](#_Toc115295653)

[Определение настроек сети развертывания 32](#_Toc115295654)

[Сетевая конфигурация вычислительных узлов 33](#_Toc115295655)

[Настройки DNS 33](#_Toc115295656)

[Настройки NTP 34](#_Toc115295657)

[Создание системы хранения на базе Ceph 35](#_Toc115295658)

[Глобальные настройки облака 36](#_Toc115295659)

[Подключение ISCSI дисков 36](#_Toc115295660)

[Настройка кластера OCFS2 37](#_Toc115295661)

[Использование VDO 38](#_Toc115295662)

[Распределение служб облака по физическим и виртуальным серверам. 41](#_Toc115295663)

[Конфигурация высокой доступности. 41](#_Toc115295664)

[Адреса контроллеров Openstack 41](#_Toc115295665)

[|Служба Memcache 42](#_Toc115295666)

[Адреса балансировщика внешних запросов haproxy. 42](#_Toc115295667)

[Адреса кластера базы данных MariaDB 42](#_Toc115295668)

[Адреса кластера службы очередей сообщений RabbitMQ 43](#_Toc115295669)

[Адреса кластера etcd 43](#_Toc115295670)

[Адреса БД Redis 43](#_Toc115295671)

[Адреса и настройки Consul 44](#_Toc115295672)

[Настройки отдельных модулей Openstack 44](#_Toc115295673)

[Keystone 44](#_Toc115295674)

[Glance 45](#_Toc115295675)

[Gnocchi 47](#_Toc115295676)

[Nova 49](#_Toc115295677)

[Neutron 49](#_Toc115295678)

[Zun 50](#_Toc115295679)

[Octavia 51](#_Toc115295680)

[Настройка модулей AccentOS 52](#_Toc115295681)

[Устаревшие параметры 52](#_Toc115295682)

[Предварительная подготовка 65](#_Toc115295683)

[Установка программного обеспечения 65](#_Toc115295684)

[Настройка DHCP-сервера. 66](#_Toc115295685)

[Установка и настройка Cobbler 66](#_Toc115295686)

[Установка Cobbler 66](#_Toc115295687)

[Настройка Cobbler 67](#_Toc115295688)

[Заключение 79](#_Toc115295689)

[Проблема с загрузкой серверов. 79](#_Toc115295690)

[Работа с локальными репозиториями. 80](#_Toc115295691)

[WebGard 91](#_Toc115295692)

[CОДЕРЖАНИЕ 91](#_Toc115295693)

[1.1. Основные возможности Программы 94](#_Toc115295694)

[1.2. Ограничения, накладываемые на область применения Программы 104](#_Toc115295695)

[Условия применения 104](#_Toc115295696)

[1.3. Эксплуатационные ограничения 105](#_Toc115295697)

[Описание задачи 106](#_Toc115295698)

[1.4. Обработка http-запросов 107](#_Toc115295699)

[1.5. Аутентификация и авторизация субъектов доступа 107](#_Toc115295700)

[1.6. Регистрация и учет действий субъектов доступа 107](#_Toc115295701)

[Архитектура системы WebGard 2.0 109](#_Toc115295702)

[Создание виртуальных машин WebGard 2.0 110](#_Toc115295703)

[1.7. Создание с виртуальной машины из диска 110](#_Toc115295704)

[1.8. Создание виртуальной машины из снимка виртуальной машины 112](#_Toc115295705)

[Установка WebGard 2.0 В среде AccentOS 113](#_Toc115295706)

[1.9. Работа с сетевыми настройками в интерфейсе AccentOS 113](#_Toc115295707)

[1.10. Настройка второго сетевого интерфейса в виртуальной машине 114](#_Toc115295708)

[1.11. Настройка DNS 115](#_Toc115295709)

[1.12. Проверка доступности с помощью скрипта conn\_checker.sh 115](#_Toc115295710)

[1.13. Настройка скрипта для создания бэкапов 116](#_Toc115295711)

[1.14. Настройка и запуск конфигурации виртуальных машин 117](#_Toc115295712)

[1.15. Проверка работы кластера виртуальных машин 118](#_Toc115295713)

[Подключение к подсистеме фильтрации и подсистеме администратора WebGard 2.0 119](#_Toc115295714)

[1.16. Вход в подсистему администрирования и предварительные настройки 119](#_Toc115295715)

[1.1. Вход в подсистему фильтрации 119](#_Toc115295716)

[1.2. Вход в подсистему фильтрации с помощью rs-client AccentOS 119](#_Toc115295717)

[Организация работы программы WebGard 2.0 в режиме высокой доступности 120](#_Toc115295718)

[1.3. Проверка кластера виртуальных машин 120](#_Toc115295719)

[1.4. Создание снимка виртуальной машины с помощью интерфейса AccentOS 120](#_Toc115295720)

[1.5. Восстановление виртуальной машины из снимка виртуальной машины 121](#_Toc115295721)

[Дополнительное взаимодействие с системой WebGard 2.0 121](#_Toc115295722)

[1.6. Проверка кластера виртуальных машин 121](#_Toc115295723)

[1.7. Редактирование парольной политики 122](#_Toc115295724)

[1.8. Редактирование политики логина 122](#_Toc115295725)

[1.9. Удаление администратора WebGard 2.0 122](#_Toc115295726)

[Приложение 1 123](#_Toc115295727)

[Название скрипта: etcd\_cluster.sh 123](#_Toc115295728)

[Приложение 2 123](#_Toc115295729)

[Название скрипта: conn\_cheker.sh 123](#_Toc115295730)

[Создание образа 126](#_Toc115295731)

[Создание сетевых портов 128](#_Toc115295732)

[Создание группы безопасности 131](#_Toc115295733)

[Создание виртуальных машин. 134](#_Toc115295734)

[Создание общего диска для Samba. 143](#_Toc115295735)

[Запуск скрипта установки и настройки Samba 149](#_Toc115295736)

[Необходимая подготовка домена ActiveDirectory 158](#_Toc115295737)

[Общее доменное имя. 158](#_Toc115295738)

[Указание размещения профилей пользователей и домашних папок. 159](#_Toc115295739)

# Состав оборудования и документации

Для выполнения работ необходимо предоставить:

* Документацию для установки платформы
* АРМ инженера (ноутбук)
* АРМ пользователя с ОС Windows 10
* АРМ пользователя с ОС Astra Linux Орел 2.12.44
* Настроенную сеть на коммутаторах и маршрутизаторах
* Настроенную СХД и SAN-сеть
* Настоенные сетевые сервисы DNS, DHCP, NTP
* Настроенные в LDAP (MS AD и Free IPA) логины/пароли для доступа Samba, администраторов, пользователей

***Документация для установки платформы***

Документация включает данные о

MAC-адресах серверов

IP-адресах серверов, СХД, маршрутизаторов, DNS, DHCP

Доменные учетные данные для установки облака, пароли в BIOS серверов, пароль к СХД.

***АРМ инженера (ноутбук)***

Установка облака AccenOS проводится с АРМ инженера (ноутбука)

Ноутбук подключен к сети VLAN KVM\_Internal.

***АРМ пользователя***

АРМ пользователя с ОС Windows 10 и АРМ пользователя с ОС Astra Linux Орел 2.12.44 должны быть установлены в непосредственной близости от АРМ администратора

АРМ подключены к сети VLAN VDI\_Public.

## Установка облака

Установка облака AccentOS требует выполнения следующих этапов:

1. Подготовка файла конфигурации для системы автоустановки
2. Проверка корректности настроек перед автоустановкой
3. Подготовка разделов на СХД и их форматирование
4. Автоустановка облака
5. Загрузка ВМ для системы WebGard и системы Samba
6. Установка и настройка ВМ для системы WebGard
7. Установка и настройка ВМ для системы Samba
8. Настройка брокеров VDI
9. Загрузка образов гостевых ВМ для системы Windows и Linux
10. Создание проектов для АРМ Windows и АРМ Linux
11. Установка ОС Windows на АРМ пользователя и настройка терминального клиента, настройка RSClient
12. Установка ОС Linux на АРМ пользователя и настройка терминального клиента, настройка RSClient
13. Проверка подлючения АРМ Windows
14. Проверка подлючения АРМ Linux
15. Настройка Zabbix
16. Настройка NetBackup
17. Настройка Kaspersky
18. Прохождение ПСИ для платформы AccentOS
19. Прохождение ПСИ для АРМ Windows
20. Прохождение ПСИ для АРМ Linux

Очередность установки и некотрые пункты требуют корректировки

### Подготовка файла конфигурации для системы автоустановки

Для установки облака AccentOS инженер должен подготовить файлы конфигурации и настроить FierstBoot и Cobbler

Настройка файлов конфигурации приведена в Приложении 1

Настройка Cobbler и работа FierstBoot приведены в Приложении 2

Запускаем образ Cobblrer

Запускаем образ Cobblrer при помощи Virt Manager

Подключаем диск с репозиториями к ВМ Cobbler через Virt Manager

Настраиваем сеть на ноутбуке (ВМ), указываем предоставленные данные подсети от РЖД

>Правим файл *et*c/network/interfaces

# This file describes the network interfaces available on your system

# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/\*

# The loopback network interface

auto lo

iface lo inet loopback

auto enp43s0

iface enp43s0 inet manual

#pre-up ifconfig $IFACE up

#post-down ifconfig $IFACE down

# address 172.20.0.100/21

# gateway 10.116.192.1

#Bridges

auto br0 br50

#kvm\_internal\_net

iface br0 inet static

# address 172.20.0.100/21

address 10.116.192.6/20

bridge\_ports enp43s0

bridge\_stp on

bridge\_maxwait 10

#ipmi\_net

iface br50 inet static

address 10.114.30.170/25

bridge\_ports enp43s0.50

bridge\_stp on

bridge\_maxwait 10

#VLANs

iface enp43s0.50 inet manual

vlan-raw-device enp43s0

>Сохраняем файл

**Запускаем образ на ВМ.**

**Доступ по ssh**

Login: root

Password: 123456

>Правим фаил: */etc/network/interface* на ВМ Cobbler

Меняем название интерфейса на существующий, вписываем свой IP(посмотреть интерфейс командой: ip a)

# This file describes the network interfaces available on your system

# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/\*

# The loopback network interface

auto lo

iface lo inet loopback

# The primary network interface

#allow-hotplug enp1s0

auto enp1s0

#iface enp1s0 inet static

# address 172.20.0.4/21

auto enp1s0

iface enp1s0 inet static

address 10.116.192.7/20 (Указываем IP полученный от РДЖ)

gateway 10.116.192.1

#auto enp9s0

#iface enp9s0 inet static

# address 10.246.253.171/25

# gateway 10.246.253.129

>Сохраняем файл.

Перезагружаем сервер ВМ

>Правим фаил: */etc/default/isc-dhcp-server*

Вставляем в строку INTERFACESv4=”enp1s0 ” свой порт.

INTERFACESv4="enp1s0" (Указываем свой порт Cobbler)

INTERFACESv6=""

>Сохраняем файл.

> Правим файл: */etc/cobbler/settings.yaml*

IP-адрес сервера Cobbler/ Указываем свой IP

default\_name\_servers: [10.116.192.7] (указывеем IP сервер Cobbler)

Создаем список прямых и обратных зон DNS: Указываем свою подсеть.

manage\_reverse\_zones: [10.116.192] (Указываем IP подсети)

next\_server: 10.116.192.7 (Указываем IP Cobbler)

IP-адрес сервера TFTPD: Указываем свой IP

next\_server: 10.116.192.7 (Указываем IP Cobbler)

IP-адрес сервера Cobbler: Указываем свой IP

server: 10.116.192.7 (Указываем IP Cobbler)

>Сохраняем файл.

>Правим фаил: */etc/cobbler/named.template*  Указываем свой IP

options {

listen-on port 53 { 127.0.0.1; 10.116.192.7; }; (Указываем IP Cobbler)

>Сохраняем файл

>Правим файл ***/etc/cobbler/dhcp.template***

Указать подсеть в которой будет разворачиваться система

Указываем маршрутизатор сети

Указываем свой IP Cobbler

Указываем свой IP cobbler в обоих строках

subnet 10.116.192.0 netmask 255.255.240.0 { (Указываем подсеть в которой ВМ и Cobbler)

option routers 10.116.192.1; (Указываем Gateway)

option domain-name-servers 10.116.192.7; (Указываем IP Cobbler)

option subnet-mask 255.255.240.0; (Маска подсети)

range dynamic-bootp 10.116.192.40 10.116.192.45; (Динамическая зона подсети)

default-lease-time 21600;

max-lease-time 43200;

next-server $next\_server;

if exists user-class and option user-class = "gPXE" {

filename "<http://10.116.192.7/cobbler/pub/boot.ipxe>"; (указываем IP Cobbler)

} else if exists user-class and option user-class = "iPXE" {

filename "<http://10.116.192.7/cobbler/pub/boot.ipxe>";(указываем IP Cobbler)

>Сохраняем файл

После внесения всех изменений перезапускаем службу:

*systemctl restart cobblerd*

**Проводим проверку:**

*cobbler check*

**Проводим синхронизацию:**

*cobbler sync*

**Проверяем доступ по вэбинтерфейсу: Указав свой IP**

http://10.40.64.3/cobbler\_web/

Логин: cobbler

Пароль: cobbler

Пройдите авторизацию

Переходим в терминал сервера Cobbler

>Вносим правки в файлы: astra-node.ssed astra\_controller.seed astra\_dbmaster.seed

в папке **/var/lib/cobbler/templates/astra\_update2021/**

d-i clock-setup/ntp-server string 10.114.4.128 (указываем IP NTP РДЖ)

d-i mirror/http/hostname string 10.116.192.7 (указываем IP Cobbler)

(указываем IP Cobbler для репозиториев)

d-i apt-setup/local0/repository string \

http://10.116.192.7/devel-smolensk smolensk contrib main non-free

d-i apt-setup/local1/repository string \

http://10.116.192.7/devel-smolensk202111 smolensk contrib main non-free

d-i apt-setup/local2/repository string \

http://10.116.192.7/smolensk202111 smolensk contrib main non-free

d-i apt-setup/local3/repository string \

http://10.116.192.7/train-accentos-sign train-accentos main

d-i apt-setup/local4/repository string \

http://10.116.192.7/20220407SE16MD smolensk contrib main non-free

d-i apt-setup/local5/repository string \

http://10.116.192.7/20220407SE16MD-devel smolensk contrib main non-free

>Сохраняем файл

Важно!!!

При тестовом запуске установке. без сети СПД.

Переименовываем файл: install\_ntp-old2 в instal\_ntp

Строки синхронизации времени должны быть такими в трех файлах **initrc\_ , initrc\_2 , initrc\_controller**:

# сервера времени облака для взаимной синхронизации

export CLOUD\_NTP1=$NET.51 (последние цифры IP УУ1)

export CLOUD\_NTP2=$NET.52 (последние цифры IP УУ2)

export CLOUD\_NTP3=$NET.53 (последние цифры IP УУ3)

export EXTERNAL\_NTP=$NET.51

#export EXTERNAL\_NTP=10.114.4.128 (IP сервера NTP РДЖ)

#export EXTERNAL\_NTP2=10.114.4.129 (IP сервера NTP РДЖ)

export TIMEZONE\_FILE=Europe/Moscow

Остальные изменения согласно файлам ниже.

При подключенной сети СПД:

ВАЖНО!!!

Переименовываем файл: install\_ntp-old в instal\_ntp

Строки синхронизации времени должны быть такими как в описании в трех файлах **initrc\_ , initrc\_2 , initrc\_controller**:

stable/astra/1.6/initrc\_

**>Правим фаил: /var/www/html/stable/astra/1.6/initrc\_**

export REPO\_IP=10.116.192.7 (IP-сервера Сobbler)

export APTKEY0="http://$REPO\_IP/train-accentos-sign/dists/train-accentos/pubkey.gpg"

export REPO0="deb http://$REPO\_IP/smolensk smolensk contrib main non-free"

export REPO1="deb http://$REPO\_IP/devel-smolensk smolensk contrib main non-free"

export REPO2="deb http://$REPO\_IP/train-accentos-sign train-accentos contrib main non-free"

export REPO3="deb http://$REPO\_IP/smolensk202111 smolensk contrib main non-free"

export REPO4="deb http://$REPO\_IP/devel-smolensk202111 smolensk contrib main non-free"

export REPO5="deb http://$REPO\_IP/20220407SE16MD smolensk contrib main non-free"

export REPO6="deb http://$REPO\_IP/20220407SE16MD-devel smolensk contrib main non-free"

export REPO7="deb http://$REPO\_IP/train-aos-sign train-aos main "

export NET=10.116.192 (IP подсети)

export DNS\_REVERSE\_ZONE=192.116.10.in-addr.arpa (IP подсети в реверсе)

# глобальный пароль

#export PASSWORD=qwertyuiop

export PASSWORD=accentos

# имя и адрес сервера cobbler

export COBBLER\_HOST=cobbler.test.loc

export COBBLER\_IP=10.116.192.7 (IP-сервера Сobbler)

# HA

export HA\_USE=yes

export CONTROLLER\_IPS="$NET.8 $NET.9 $NET.10" (последние цифры IP контроллеров)

export HAPROXY\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export MYSQL\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export RABBITMQ\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export ETCD\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export REDIS\_IPS="$NET.8 $NET.9 $NET.10" (последние цифры IP контроллеров)

export OCFS2\_IPS="$NET.51 $NET.52 $NET.53 $NET.54 $NET.55 $NET.56 $NET.57 $NET.58 $NET.59 $NET.60 $NET.61 $NET.62 $NET.63 $NET.64" (последние цифры IP УУ и ВУ для 1 класттера)

export OCFS2\_CLUSTER\_NAME=ocfs2node1

export OCFS2\_DRIVE="/dev/mapper/360060e80212bdf0050602bdf00000002"

export DNS\_ZONE=test.loc

export CLOUD\_ZONE=pvrr.loc (указываем dns)

#export CLOUD\_UPDATEDNS=$NET.500

export CLOUD\_DNS1=$NET.51 (последние цифры IP УУ1)

export CLOUD\_DNS2=$NET.52 (последние цифры IP УУ2)

export CLOUD\_DNS3=$NET.53 (последние цифры IP УУ3)

# сервера времени облака для взаимной синхронизации

export CLOUD\_NTP1=$NET.51 (последние цифры IP УУ1)

export CLOUD\_NTP2=$NET.52 (последние цифры IP УУ2)

export CLOUD\_NTP3=$NET.53 (последние цифры IP УУ3)

#export EXTERNAL\_NTP=$NET.51

export EXTERNAL\_NTP=10.114.4.128 (IP сервера NTP РДЖ)

export EXTERNAL\_NTP2=10.114.4.129 (IP сервера NTP РДЖ)

export TIMEZONE\_FILE=Europe/Moscow

#имя контроллера облака

export CONTROLLER\_NAME=controller.$CLOUD\_ZONE

#export CONTROLLER\_IP=$NET.219

# неинтенрактивная установка пакетов debian

export DEBIAN\_FRONTEND="noninteractive"

export BACKUP\_NODE=$COBBLER\_IP

export BACKUP\_PATCH=/backup

export BACKUP\_LOCAL\_PATCH=/mnt/backup

export SMBUSER=cloud\_backup

export SMBPASS=backup

export GLANCE\_DRIVE=/dev/mapper/360060e80212bdf0050602bdf00000001

export GLANCE\_DRIVE\_FS=ocfs2

export GLANCE\_DRIVE\_MAPPED=vdb

export GNOCCHI\_DRIVE=/dev/mapper/360060e80212bdf0050602bdf00000000

export GNOCCHI\_DRIVE\_FS=ocfs2

export GNOCCHI\_DRIVE\_MAPPED=vdc

#раздел для /var/lib/nova/instances на ВУ

#если значение пустое, то отдельный раздел не используется.

export NOVA\_DRIVE=/dev/mapper/360060e80212bdf0050602bdf00000002

export NOVA\_DRIVE\_FS=ocfs2

#использование vdo на разделе NOVA\_DRIVE

#параметры для vdo игнорируются если значение NOVA\_DRIVE не определено

export NOVA\_VDO\_USE=no

export KVM\_NESTED=no

# настройки accentos

export AOS\_USER=aos

export AOS\_PASSWORD=$PASSWORD

export AOS\_RABBIT\_VHOST=aos

#список подчиненных сетевых интерфейсов для bond0

#пустой список - все сетевые интерфейсы, определяемые с помощью mii-tool

export BOND\_SLAVES="eth0 eth1"

export MTU=9000

export NEUTRON\_DHCP\_AGENTS\_COUNT=2

export DVR=yes

#BOOT\_MANAGER=firstboot

export BOOT\_MANAGER=systemd

export ZABBIX\_HOSTNAME=zabbix.pvrr.loc (меняем “pvrr” на dns объекта)

export ZABBIX\_IP=10.116.192.17 (IP сервера Zabbix)

>Сохраняем файл

stable/astra/1.6/initrc\_2

**>Правим фаил: /var/www/html/stable/astra/1.6/initrc\_2**

по аналогии **initrc\_**

export REPO\_IP=10.116.192.7 (IP-сервера Сobbler)

export APTKEY0="http://$REPO\_IP/train-accentos-sign/dists/train-accentos/pubkey.gpg"

export REPO0="deb http://$REPO\_IP/smolensk smolensk contrib main non-free"

export REPO1="deb http://$REPO\_IP/devel-smolensk smolensk contrib main non-free"

export REPO2="deb http://$REPO\_IP/train-accentos-sign train-accentos contrib main non-free"

export REPO3="deb http://$REPO\_IP/smolensk202111 smolensk contrib main non-free"

export REPO4="deb http://$REPO\_IP/devel-smolensk202111 smolensk contrib main non-free"

export REPO5="deb http://$REPO\_IP/20220407SE16MD smolensk contrib main non-free"

export REPO6="deb http://$REPO\_IP/20220407SE16MD-devel smolensk contrib main non-free"

export REPO7="deb http://$REPO\_IP/train-aos-sign train-aos main "

export NET=10.116.192 (IP подсети)

export DNS\_REVERSE\_ZONE=192.116.10.in-addr.arpa (IP подсети в реверсе)

# глобальный пароль

#export PASSWORD=qwertyuiop

export PASSWORD=accentos

# имя и адрес сервера cobbler

export COBBLER\_HOST=cobbler.test.loc

export COBBLER\_IP=10.116.192.7 (IP cobbler)

# HA

export HA\_USE=yes

export CONTROLLER\_IPS="$NET.8 $NET.9 $NET.10" (последние цифры IP контроллеров)

export HAPROXY\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export MYSQL\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export RABBITMQ\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export ETCD\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export REDIS\_IPS="$NET.8 $NET.9 $NET.10" (последние цифры IP контроллеров)

export OCFS2\_IPS="$NET.65 $NET.66 $NET.67 $NET.68 $NET.69 $NET.70 $NET.71 $NET.72 $NET.73 $NET.74 $NET.75 $NET.76 $NET.77" (последние цифры IP ВУ для 2 класттера)

export OCFS2\_CLUSTER\_NAME=ocfs2node2

export OCFS2\_DRIVE="/dev/mapper/360060e80212bdf0050602bdf00000003"

export DNS\_ZONE=test.loc

export CLOUD\_ZONE=pvrr.loc (указываем dns)

export CLOUD\_DNS1=$NET.51 (последние цифры IP УУ1)

export CLOUD\_DNS2=$NET.52 (последние цифры IP УУ2)

export CLOUD\_DNS3=$NET.53 (последние цифры IP УУ2)

# сервера времени облака для взаимной синхронизации

export CLOUD\_NTP1=$NET.51 (последние цифры IP УУ1)

export CLOUD\_NTP2=$NET.52 (последние цифры IP УУ2)

export CLOUD\_NTP3=$NET.53 (последние цифры IP УУ3)

#export EXTERNAL\_NTP=$NET.51

export EXTERNAL\_NTP=10.114.4.128 (IP сервера NTP РДЖ)

export EXTERNAL\_NTP2=10.114.4.129 (IP сервера NTP РДЖ)

export TIMEZONE\_FILE=Europe/Moscow

#имя контроллера облака

export CONTROLLER\_NAME=controller.$CLOUD\_ZONE

# неинтенрактивная установка пакетов debian

export DEBIAN\_FRONTEND="noninteractive"

export BACKUP\_NODE=$COBBLER\_IP

export BACKUP\_PATCH=/backup

export BACKUP\_LOCAL\_PATCH=/mnt/backup

export SMBUSER=cloud\_backup

export SMBPASS=backup

export GLANCE\_DRIVE=/dev/mapper/360060e80212bdf0050602bdf00000001

export GLANCE\_DRIVE\_FS=ocfs2

export GLANCE\_DRIVE\_MAPPED=vdb

export GNOCCHI\_DRIVE=/dev/mapper/360060e80212bdf0050602bdf00000000

export GNOCCHI\_DRIVE\_FS=ocfs2

export GNOCCHI\_DRIVE\_MAPPED=vdc

#раздел для /var/lib/nova/instances на ВУ

#если значение пустое, то отдельный раздел не используется.

export NOVA\_DRIVE=/dev/mapper/360060e80212bdf0050602bdf00000003

export NOVA\_DRIVE\_FS=ocfs2

#использование vdo на разделе NOVA\_DRIVE

#параметры для vdo игнорируются если значение NOVA\_DRIVE не определено

export NOVA\_VDO\_USE=no

export KVM\_NESTED=no

# настройки accentos

export AOS\_USER=aos

export AOS\_PASSWORD=$PASSWORD

export AOS\_RABBIT\_VHOST=aos

#список подчиненных сетевых интерфейсов для bond0

#пустой список - все сетевые интерфейсы, определяемые с помощью mii-tool

export BOND\_SLAVES="eth0 eth1"

export MTU=9000

export NEUTRON\_DHCP\_AGENTS\_COUNT=2

export DVR=yes

#BOOT\_MANAGER=firstboot

export BOOT\_MANAGER=systemd

export ZABBIX\_HOSTNAME=zabbix.pvrr.loc (меняем “pvrr” на dns объекта)

export ZABBIX\_IP=10.116.192.17 (IP сервера Zabbix)

>Сохраняем файл

stable/astra/1.6/initrc\_.controller

**>Правим фаил: /var/www/html/stable/astra/1.6/initrc\_.controller**

по аналогии **initrc\_**

export REPO\_IP=10.116.192.7 (IP cobbler)

export APTKEY0="http://$REPO\_IP/train-accentos-sign/dists/train-accentos/pubkey.gpg"

export REPO0="deb http://$REPO\_IP/smolensk smolensk contrib main non-free"

export REPO1="deb http://$REPO\_IP/devel-smolensk smolensk contrib main non-free"

export REPO2="deb http://$REPO\_IP/train-accentos-sign train-accentos contrib main non-free"

export REPO3="deb http://$REPO\_IP/smolensk202111 smolensk contrib main non-free"

export REPO4="deb http://$REPO\_IP/devel-smolensk202111 smolensk contrib main non-free"

export REPO5="deb http://$REPO\_IP/20220407SE16MD smolensk contrib main non-free"

export REPO6="deb http://$REPO\_IP/20220407SE16MD-devel smolensk contrib main non-free"

export REPO7="deb http://$REPO\_IP/train-aos-sign train-aos main "

export NET=10.116.192 (IP подсети)

export DNS\_REVERSE\_ZONE=192.116.10.in-addr.arpa (IP подсети в реверсе)

# глобальный пароль

#export PASSWORD=qwertyuiop

export PASSWORD=accentos

# имя и адрес сервера cobbler

export COBBLER\_HOST=cobbler.test.loc

export COBBLER\_IP=10.116.192.7 (IP cobbler)

# HA

export HA\_USE=yes

export CONTROLLER\_IPS="$NET.8 $NET.9 $NET.10" (последние цифры IP контроллеров)

export HAPROXY\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export MYSQL\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export RABBITMQ\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export ETCD\_IPS="$NET.11 $NET.12 $NET.13" (последние цифры IP БД)

export REDIS\_IPS="$NET.8 $NET.9 $NET.10" (последние цифры IP контроллеров)

export OCFS2\_IPS="$NET.8 $NET.9 $NET.10" (последние цифры IP контроллеров)

export OCFS2\_CLUSTER\_NAME=ocfs2ctrl

export OCFS2\_DRIVE="/dev/vdb /dev/vdc"

#export SHARED\_DRIVE="ceph"

#export CEPH\_IPS="$NET.221 $NET.222 $NET.223"

export DNS\_ZONE=test.loc

export CLOUD\_ZONE=pvrr.loc (указываем dns)

#export CLOUD\_UPDATEDNS=$NET.500

export CLOUD\_DNS1=$NET.51 (последние цифры IP УУ1)

export CLOUD\_DNS2=$NET.52 (последние цифры IP УУ2)

export CLOUD\_DNS3=$NET.53 (последние цифры IP УУ3)

# сервера времени облака для взаимной синхронизации

export CLOUD\_NTP1=$NET.51 (последние цифры IP УУ1)

export CLOUD\_NTP2=$NET.52 (последние цифры IP УУ2)

export CLOUD\_NTP3=$NET.53 (последние цифры IP УУ3)

#export EXTERNAL\_NTP=$NET.51

export EXTERNAL\_NTP=10.114.4.128 (IP сервера NTP РДЖ)

export EXTERNAL\_NTP2=10.114.4.129 (IP сервера NTP РДЖ)

export TIMEZONE\_FILE=Europe/Moscow

#имя контроллера облака

export CONTROLLER\_NAME=controller.$CLOUD\_ZONE

#export CONTROLLER\_IP=$NET.219

# неинтенрактивная установка пакетов debian

export DEBIAN\_FRONTEND="noninteractive"

export BACKUP\_NODE=$COBBLER\_IP

export BACKUP\_PATCH=/backup

export BACKUP\_LOCAL\_PATCH=/mnt/backup

export SMBUSER=cloud\_backup

export SMBPASS=backup

export GLANCE\_DRIVE=/dev/mapper/360060e80212bdf0050602bdf00000001

export GLANCE\_DRIVE\_FS=ocfs2

export GLANCE\_DRIVE\_MAPPED=vdb

export GNOCCHI\_DRIVE=/dev/mapper/360060e80212bdf0050602bdf00000000

export GNOCCHI\_DRIVE\_FS=ocfs2

export GNOCCHI\_DRIVE\_MAPPED=vdc

#раздел для /var/lib/nova/instances на ВУ

#если значение пустое, то отдельный раздел не используется.

export NOVA\_DRIVE=/dev/mapper/

export NOVA\_DRIVE\_FS=ocfs2

#использование vdo на разделе NOVA\_DRIVE

#параметры для vdo игнорируются если значение NOVA\_DRIVE не определено

export NOVA\_VDO\_USE=no

#коэффициент для VDO

#export VDON=8

#имя блочного устройства для vdo

export VDODRIVE=nvme0n1p4

export KVM\_NESTED=no

# настройки accentos

export AOS\_USER=aos

export AOS\_PASSWORD=$PASSWORD

export AOS\_RABBIT\_VHOST=aos

#список подчиненных сетевых интерфейсов для bond0

#пустой список - все сетевые интерфейсы, определяемые с помощью mii-tool

export BOND\_SLAVES="eth0 eth1"

#export MTU=9000

export MTU=9000

export NEUTRON\_DHCP\_AGENTS\_COUNT=2

export DVR=yes

#BOOT\_MANAGER=firstboot

export BOOT\_MANAGER=systemd

#KERNEL

export KERNEL=gnulinux-5.4.0-71-generic-advanced

#LICENSE

export RZHD\_SITE=sivc

export KEYSTONE\_TOKEN\_EXPIRATION=40000

>Сохраняем файл

install\_cinder-volume.ha

**>Правим файл: /var/www/html/stable/astra/1.6/install\_cinder-volume.ha**

#!/bin/bash

#export CINDER\_VOLUME\_FENCE\_AGENT='fence\_ipmilan'

export CINDER\_VOLUME\_FENCE\_AGENT='fence\_dummy'

export CINDER\_VOLUME0\_IP="10.116.192.51" (IP 1 УУ)

export CINDER\_VOLUME0\_HOSTNAME=$(host $CINDER\_VOLUME0\_IP | awk '{print $5}' | sed s/\.$//)

export CINDER\_VOLUME0\_IPMI\_IP="10.114.30.132" (IP IPMI 1 УУ)

export CINDER\_VOLUME0\_IPMI\_LOGIN="admin"

export CINDER\_VOLUME0\_IPMI\_PASSWORD="P@ssw0rd"

export CINDER\_VOLUME1\_IP="10.116.192.52" (IP 2 УУ)

export CINDER\_VOLUME1\_HOSTNAME=$(host $CINDER\_VOLUME1\_IP | awk '{print $5}' | sed s/\.$//)

export CINDER\_VOLUME1\_IPMI\_IP="10.114.30.133" (IP IPMI 2 УУ)

export CINDER\_VOLUME1\_IPMI\_LOGIN="admin"

export CINDER\_VOLUME1\_IPMI\_PASSWORD="P@ssw0rd"

export CINDER\_VOLUME2\_IP="10.116.192.53" (IP 3 УУ)

export CINDER\_VOLUME2\_HOSTNAME=$(host $CINDER\_VOLUME2\_IP | awk '{print $5}' | sed s/\.$//)

export CINDER\_VOLUME2\_IPMI\_IP="10.114.30.134" (IP IPMI 3 УУ)

export CINDER\_VOLUME2\_IPMI\_LOGIN="admin"

export CINDER\_VOLUME2\_IPMI\_PASSWORD="P@ssw0rd"

export URL\_CINDER\_CONFIG="http://$COBBLER\_IP/init/cinder/cinder.conf"

export DEBIAN\_FRONTEND="noninteractive"

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

apt-get install pcs fence-agents -y crmsh corosync watchdog cinder-volume lvm2 qemu-utils kpartx open-iscsi thin-provisioning-tools parted

modprobe iTCO\_wdt

echo 'watchdog\_module="iTCO\_wdt"' >> /etc/default/watchdog

systemctl restart watchdog

mv -f /etc/cinder/cinder.conf{,.orig}

wget -O /etc/cinder/cinder.conf $URL\_CINDER\_CONFIG

sed -i "s/^my\_ip=.\*/my\_ip=$IP/" /etc/cinder.conf

#sed "s/^127.0.1.1/#127.0.1.1/" /etc/hosts

echo -e "$PASSWORD\n$PASSWORD" | passwd hacluster

mkdir -pv /home/hacluster/.ssh

cp -fr /root/.ssh/\* /home/hacluster/.ssh

chown -R hacluster:haclient /home/hacluster

systemctl restart pcsd pacemaker

master\_host=$(hostname)

init\_time=$(date +%s)

echo $master\_host $init\_time > /etc/corosync/init.txt

sleep 60

for i in $CINDER\_VOLUME0\_IP $CINDER\_VOLUME1\_IP $CINDER\_VOLUME2\_IP

do

if [ "$i" != "$IP" ]; then

count=$(uname -a | grep astra | wc -l)

if [ "$count" -gt "0" ]; then

RES=$(ssh -oStrictHostKeyChecking=no u@$i "cat /etc/corosync/init.txt")

else

RES=$(ssh -oStrictHostKeyChecking=no root@$i "cat /etc/corosync/init.txt")

fi

while [ "a$RES" = "a" ]

do

sleep 60

if [ "$count" -gt "0" ]; then

RES=$(ssh -oStrictHostKeyChecking=no u@$i "cat /etc/corosync/init.txt")

else

RES=$(ssh -oStrictHostKeyChecking=no root@$i "cat /etc/corosync/init.txt")

fi

done

fi

done

if [ "$IP" = "$CINDER\_VOLUME0\_IP" ]; then

corosync-keygen

echo "# Totem Protocol Configuration

totem {

version: 2

cluster\_name: stand-cluster

transport: udpu

# Interface configuration for Corosync

interface {

ringnumber: 0

bindnetaddr: $NET.0

broadcast: yes

mcastport: 5407

}

}

# Nodelist - Server List

nodelist {

node {

ring0\_addr: $CINDER\_VOLUME0\_IP

nodeid: 1

}

node {

ring0\_addr: $CINDER\_VOLUME1\_IP

nodeid: 2

}

node {

ring0\_addr: $CINDER\_VOLUME2\_IP

nodeid: 3

}

}

# Quorum configuration

quorum {

provider: corosync\_votequorum

expected\_votes: 2

}

# Corosync Log configuration

logging {

to\_logfile: yes

logfile: /var/log/corosync/corosync.log

to\_syslog: yes

timestamp: on

}

service {

name: pacemaker

ver: 0

} " > /etc/corosync/corosync.conf

scp /etc/corosync/\* root@$CINDER\_VOLUME1\_HOSTNAME:/etc/corosync

scp /etc/corosync/\* root@$CINDER\_VOLUME2\_HOSTNAME:/etc/corosync

systemctl restart corosync pacemaker

systemctl enable corosync pacemaker

update-rc.d pacemaker defaults 20 01

for ip in $CINDER\_VOLUME1\_IP $CINDER\_VOLUME2\_IP

do

count=$(uname -a | grep astra | wc -l)

if [ "$count" -gt "0" ]; then

ssh u@$ip "sudo systemctl restart corosync pacemaker; sudo systemctl enable corosync pacemaker; sudo update-rc.d pacemaker defaults 20 01"

else

ssh root@$ip "systemctl restart corosync pacemaker; systemctl enable corosync pacemaker; update-rc.d pacemaker defaults 20 01"

fi

done

sleep 20

count=0

while [ "$count" -lt "3" ]

do

systemctl restart corosync

count=$(uname -a | grep astra | wc -l)

if [ "$count" -gt "0" ]; then

ssh u@$CINDER\_VOLUME1\_HOSTNAME "sudo systemctl restart corosync"

ssh u@$CINDER\_VOLUME2\_HOSTNAME "sudo systemctl restart corosync"

else

ssh root@$CINDER\_VOLUME1\_HOSTNAME "systemctl restart corosync"

ssh root@$CINDER\_VOLUME2\_HOSTNAME "systemctl restart corosync"

fi

count=0

sleep 15

status=$(crm status | grep "^Online:")

for ii in $status

do

echo $ii

case $ii in

"Online:" | "[" | "]" )

;;

\* )

let count=$count+1

;;

esac

done

done

clusternode="sanlock1.$CONTROLLER\_NAME sanlock2.$CONTROLLER\_NAME sanlock3.$CONTROLLER\_NAME"

crm status

corosync-cmapctl | grep members

pcs status

pcs cluster auth $CINDER\_VOLUME0\_HOSTNAME $CINDER\_VOLUME1\_HOSTNAME $CINDER\_VOLUME2\_HOSTNAME <<END

hacluster

$PASSWORD

END

#pcs cluster setup --start --name stand\_cluster $clusternode

pcs cluster enable --all

crm status

pcs status

pcs property set stonith-enabled=true

pcs property set no-quorum-policy=ignore

crm configure show

if [ $CINDER\_VOLUME\_FENCE\_AGENT = 'fence\_ipmilan' ]; then

ssh u@$CINDER\_VOLUME1\_HOSTNAME "sudo pcs stonith create ipmilan-$CINDER\_VOLUME0\_HOSTNAME fence\_ipmilan pcmk\_host\_list=$CINDER\_VOLUME0\_IP ipaddr=$CINDER\_VOLUME0\_IPMI\_IP login=$CINDER\_VOLUME0\_IPMI$

ssh u@$CINDER\_VOLUME1\_HOSTNAME "sudo pcs constraint location ipmilan-$CINDER\_VOLUME0\_HOSTNAME avoids $CINDER\_VOLUME0\_HOSTNAME"

pcs stonith create ipmilan-$CINDER\_VOLUME1\_HOSTNAME fence\_ipmilan pcmk\_host\_list=$CINDER\_VOLUME1\_IP ipaddr=$CINDER\_VOLUME1\_IPMI\_IP login=$CINDER\_VOLUME1\_IPMI\_LOGIN passwd=$CINDER\_VOLUME1\_IPMI\_PA$

pcs constraint location ipmilan-$CINDER\_VOLUME1\_HOSTNAME avoids $CINDER\_VOLUME1\_HOSTNAME

pcs stonith create ipmilan-$CINDER\_VOLUME2\_HOSTNAME fence\_ipmilan pcmk\_host\_list=$CINDER\_VOLUME2\_IP ipaddr=$CINDER\_VOLUME2\_IPMI\_IP login=$CINDER\_VOLUME2\_IPMI\_LOGIN passwd=$CINDER\_VOLUME2\_IPMI\_PA$

pcs constraint location ipmilan-$CINDER\_VOLUME2\_HOSTNAME avoids $CINDER\_VOLUME2\_HOSTNAME

fi

if [ $CINDER\_VOLUME\_FENCE\_AGENT = 'fence\_dummy' ]; then

pcs stonith create scsi-pcmk fence\_dummy \

pcmk\_host\_list="$CINDER\_VOLUME0\_HOSTNAME $CINDER\_VOLUME1\_HOSTNAME $CINDER\_VOLUME2\_HOSTNAME" \

pcmk\_reboot\_action="off" meta provides=unfencing

fi

pcs resource create cinder-volume lsb:cinder-volume --group cinder\_volume

crm status

source /root/admin-openrc

openstack volume type set \_\_DEFAULT\_\_ --property volume\_backend\_name=hitachi2

openstack volume type create multiattache

openstack volume type set multiattache --property volume\_backend\_name=hitachim

openstack volume type set multiattache --property multiattach="<is> True"

>Сохраняем файл

demo/astra\_node

>Правим файл ***/var/www/html/demo/astra\_node***

*Указываем имена серверов согласно их наименованию и МАС адреса ВМ*

!/bin/bash

release=stable

source /target/etc/os-release

export ID=$(chroot /target /usr/bin/awk -F= '$1=="ID" { print $2 ;}' /etc/os-release | /bin/sed 's/"//g')

export VERSION\_ID=$(chroot /target /usr/bin/awk -F= '$1=="VERSION\_ID" { print $2 ;}' /etc/os-release | /bin/sed 's/"//g')

if [ "a$ID" = "a" ]; then

linux\_distro=astra

else

linux\_distro=$ID

fi

if [ "a$VERSION\_ID" = "a" ]; then

linux\_version=1.6

else

linux\_version=$VERSION\_ID

fi

http\_server=cobbler.test.loc

# node name in cobbler

node\_name=$1

echo "node\_name=$node\_name" > /target/var/log/firstboot.log

env >> /target/var/log/firstboot.log

wget -O- http://$http\_server/$release/$linux\_distro/$linux\_version/astraifaces | chroot /target /bin/sh -s

#wget -O- http://$http\_server/$release/$linux\_distro/$linux\_version/astraeth | chroot /target /bin/sh -s

wget -O- http://$http\_server/$release/$linux\_distro/$linux\_version/firstboot | chroot /target /bin/sh -s

#wget -O- http://$http\_server/$release/$linux\_distro/$linux\_version/update\_kernel | chroot /target /bin/sh -s

#exit

if [ $node\_name = "pvrr-virazh-001" ] (имя 1 УУ)

then

sleep 1

else

sleep 458

fi

if [ $node\_name = "pvrr-virazh-001" ] || [ $node\_name = "pvrr-virazh-002" ] || [ $node\_name = "pvrr-virazh-003" ] || [ $node\_name = "pvrr-virazh-004" ] || [ $node\_name = "pvrr-virazh-005" ] || [ $node\_na (Имена нод 1 кластера)

<------>wget -O /target/var/lib/firstboot/scripts/05-initrc.source http://$http\_server/$release/$linux\_distro/$linux\_version/initrc\_

<------>wget -O /target/var/lib/firstboot/boot/05-initrc.source http://$http\_server/$release/$linux\_distro/$linux\_version/initrc\_

fi

if [ $node\_name = "pvrr-virazh-015" ] || [ $node\_name = "pvrr-virazh-016" ] || [ $node\_name = "pvrr-virazh-017" ] || [ $node\_name = "pvrr-virazh-018" ] || [ $node\_name = "pvrr-virazh-019" ] || [ $node\_na (Имена нод 2 кластера)

<------>wget -O /target/var/lib/firstboot/scripts/05-initrc.source http://$http\_server/$release/$linux\_distro/$linux\_version/initrc\_2

<------>wget -O /target/var/lib/firstboot/boot/05-initrc.source http://$http\_server/$release/$linux\_distro/$linux\_version/initrc\_2

fi

wget -O /target/var/lib/firstboot/scripts/10-install.sh http://$http\_server/$release/$linux\_distro/$linux\_version/wait\_found\_cobbler

wget -O /target/var/lib/firstboot/scripts/15-install.sh http://$http\_server/$release/$linux\_distro/$linux\_version/rc-local

cho > /target/var/lib/firstboot/scripts/17-reboot

wget -O /target/var/lib/firstboot/scripts/19-install.sh http://$http\_server/$release/$linux\_distro/$linux\_version/wait\_found\_cobbler

wget -O /target/var/lib/firstboot/scripts/20-install\_multipath.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_multipath

#wget -O /target/var/lib/firstboot/scripts/20-install\_nfs.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_nfs

wget -O /target/var/lib/firstboot/scripts/20-install\_bind9.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_bind9

wget -O /target/var/lib/firstboot/scripts/20-install\_ntp.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_ntp

wget -O /target/var/lib/firstboot/scripts/20-install\_update-dns.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_update-dns

#wget -O /target/var/lib/firstboot/scripts/27-install\_ceph.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_ceph\_cluster

#wget -O /target/var/lib/firstboot/scripts/20-install\_vdo.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_vdo

#wget -O /target/var/lib/firstboot/scripts/25-install\_vdo-compute.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_vdo-compute

#wget -O /target/var/lib/firstboot/scripts/25-install\_consul.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_consul

wget -O /target/var/lib/firstboot/scripts/23-install\_packages.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_packages

wget -O /target/var/lib/firstboot/scripts/25-install\_ocfs2.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_ocfs2

wget -O /target/var/lib/firstboot/scripts/25-install\_openstackclient.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_openstackclient

wget -O /target/var/lib/firstboot/scripts/27-prepare\_vmha.sh http://$http\_server/$release/$linux\_distro/$linux\_version/prepare\_vm

echo > /target/var/lib/firstboot/scripts/28-reboot

wget -O /target/var/lib/firstboot/scripts/29-install.sh http://$http\_server/$release/$linux\_distro/$linux\_version/wait\_found\_cobbler

if [ $node\_name = "pvrr-virazh-001" ]; then (Имя 1 УУ)

echo "export VM\_MAC=52:54:00:2f:01:01 (Мас БД 1)

export VM\_NAME=pvrr-virazh-vm-dbmaster-01 (Имя БД)

export VM\_RAM=18000000

export VM\_SIZE=40

export VM\_CPU=4

#export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/astra.xml

export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/debian.xml

" >> /target/var/lib/firstboot/scripts/31-install-dbha.source

wget -O /target/var/lib/firstboot/scripts/32-install\_vmha-db1.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_vmha

echo "export VM\_MAC=52:54:00:2f:00:01 (Мак Контр.1)

export VM\_NAME=pvrr-virazh-vm-controller-01 (Мак Контр 1)

export VM\_RAM=28000000

export VM\_SIZE=500

export VM\_CPU=6

#export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/astra.xml

export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/debian.xml

export ADD\_DRIVE=yes

" >> /target/var/lib/firstboot/scripts/33-install-dbha.source

wget -O /target/var/lib/firstboot/scripts/34-install\_vmha-db1.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_vmha

i

if [ $node\_name = "pvrr-virazh-002" ]; then (Имя 2 УУ)

echo "export VM\_MAC=52:54:00:2f:01:02 (Мак БД 2)

export VM\_NAME=pvrr-virazh-vm-dbmaster-02 (Имя БД 2)

export VM\_RAM=18000000

export VM\_SIZE=40

export VM\_CPU=4

#export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/astra.xml

export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/debian.xml

" >> /target/var/lib/firstboot/scripts/31-install-dbha.source

wget -O /target/var/lib/firstboot/scripts/32-install\_vmha.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_vmha

echo "export VM\_MAC=52:54:00:2f:00:02 (Мак Контр 2)

export VM\_NAME=pvrr-virazh-vm-controller-02 (Имя Контр 2)

export VM\_RAM=28000000

export VM\_SIZE=500

export VM\_CPU=6

#export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/astra.xml

export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/debian.xml

export ADD\_DRIVE=yes

" >> /target/var/lib/firstboot/scripts/33-install-dbha.source

wget -O /target/var/lib/firstboot/scripts/34-install\_vmha.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_vmha

fi

if [ $node\_name = "pvrr-virazh-003" ]; then (Имя 3 УУ)

echo "export VM\_MAC=52:54:00:2f:01:03 (Мак БД 3)

export VM\_NAME=pvrr-virazh-vm-dbmaster-03 (Имя БД 3)

export VM\_RAM=18000000

export VM\_SIZE=40

export VM\_CPU=4

#export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/astra.xml

export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/debian.xml

" >> /target/var/lib/firstboot/scripts/31-install-dbha.source

wget -O /target/var/lib/firstboot/scripts/32-install\_vmha.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_vmha

echo "export VM\_MAC=52:54:00:2f:00:03 (Мак Контр 3)

export VM\_NAME=pvrr-virazh-vm-controller-03 (Имя Контр 3)

export VM\_RAM=28000000

export VM\_SIZE=500

export VM\_CPU=6

#export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/astra.xml

export VM\_XML\_URL=http://$http\_server/$release/$linux\_distro/$linux\_version/libvirtxml/debian.xml

export ADD\_DRIVE=yes

" >> /target/var/lib/firstboot/scripts/33-install-dbha.source

wget -O /target/var/lib/firstboot/scripts/34-install\_vmha.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_vmha

fi

#wget -O /target/var/lib/firstboot/scripts/35-astraadd\_ceph.sh http://$http\_server/$release/$linux\_distro/$linux\_version/astraadd\_ceph

get -O /target/var/lib/firstboot/scripts/38-wait\_start\_db.sh http://$http\_server/$release/$linux\_distro/$linux\_version/wait\_start\_db.ha

wget -O /target/var/lib/firstboot/boot/38-wait\_start\_db.sh http://$http\_server/$release/$linux\_distro/$linux\_version/wait\_start\_db.ha

wget -O /target/var/lib/firstboot/scripts/39-install\_haproxy.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_haproxy\_node

wget -O /target/var/lib/firstboot/scripts/40-install\_neutron-compute.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_neutron-compute.ha

wget -O /target/var/lib/firstboot/scripts/45-install\_nova-compute.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_nova-compute.ha

#wget -O /target/var/lib/firstboot/scripts/47-install\_zun-compute.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_zun-compute.ha

wget -O /target/var/lib/firstboot/scripts/50-install\_ceilometer-compute.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_ceilometer-compute.ha

if [ $node\_name = "pvrr-virazh-001" ]; then (Имя 1 УУ)

# echo "#!/bin/bash

#apt-get install -y cinder-volume

#wget http://$http\_server/$release/$linux\_distro/$linux\_version/cinder-c1.conf -O /etc/cinder/cinder.conf

#wget http://$http\_server/$release/$linux\_distro/$linux\_version/cinder-driver/hitachi-train-cinder.tar.gz -O /usr/lib/python3/dist-packages/cinder/volume/hitachi-train-cinder.tar.gz

#tar -xf /usr/lib/python3/dist-packages/cinder/volume/hitachi-train-cinder.tar.gz -C /usr/lib/python3/dist-packages/cinder/volume/drivers

#systemctl restart cinder-volume

#" >> /target/var/lib/firstboot/scripts/60-install\_cinder-volume.sh

wget -O /target/var/lib/firstboot/scripts/60-install\_cinder-volume.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_cinder-volume.ha

fi

if [ $node\_name = "pvrr-virazh-002" ]; then (Имя 2 УУ)

# echo "#!/bin/bash

#apt-get install -y cinder-volume

#wget http://$http\_server/$release/$linux\_distro/$linux\_version/cinder-c2.conf -O /etc/cinder/cinder.conf

#wget http://$http\_server/$release/$linux\_distro/$linux\_version/cinder-driver/hitachi-train-cinder.tar.gz -O /usr/lib/python3/dist-packages/cinder/volume/hitachi-train-cinder.tar.gz

#tar -xf /usr/lib/python3/dist-packages/cinder/volume/hitachi-train-cinder.tar.gz -C /usr/lib/python3/dist-packages/cinder/volume/drivers

#systemctl stop cinder-volume

#systemctl disable cinder-volume

#" >> /target/var/lib/firstboot/scripts/60-install\_cinder-volume.sh

wget -O /target/var/lib/firstboot/scripts/60-install\_cinder-volume.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_cinder-volume.ha

fi

if [ $node\_name = "pvrr-virazh-003" ]; then (Имя 3 УУ)

# echo "#!/bin/bash

#apt-get install -y cinder-volume

#wget http://$http\_server/$release/$linux\_distro/$linux\_version/cinder-c3.conf -O /etc/cinder/cinder.conf

#wget http://$http\_server/$release/$linux\_distro/$linux\_version/cinder-driver/hitachi-train-cinder.tar.gz -O /usr/lib/python3/dist-packages/cinder/volume/hitachi-train-cinder.tar.gz

#tar -xf /usr/lib/python3/dist-packages/cinder/volume/hitachi-train-cinder.tar.gz -C /usr/lib/python3/dist-packages/cinder/volume/drivers

#systemctl stop cinder-volume

#systemctl disable cinder-volume

#" >> /target/var/lib/firstboot/scripts/60-install\_cinder-volume.sh

wget -O /target/var/lib/firstboot/scripts/60-install\_cinder-volume.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_cinder-volume.ha

fi

wget -O /target/var/lib/firstboot/scripts/70-install\_bind9-forward-zone.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_bind9-forward-zone

mkdir -pv /target/var/lib/firstboot/aos

wget -O /target/var/lib/firstboot/aos/update\_aos-agent.sh http://$http\_server/$release/$linux\_distro/$linux\_version/update\_aos-agent

wget -O /target/var/lib/firstboot/scripts/85-install\_aos-agent.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_aos-agent

wget -O /target/var/lib/firstboot/scripts/90-correct\_nova\_shell.sh http://$http\_server/$release/$linux\_distro/$linux\_version/correct\_nova\_shell

#wget -O /target/var/lib/firstboot/scripts/99-install\_powerdns.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_powerdns

wget -O /target/var/lib/firstboot/scripts/94-install\_zabbix-agent.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_zabbix-agent

wget -O /target/var/lib/firstboot/scripts/96-install\_netbackup-client.sh http://$http\_server/$release/$linux\_distro/$linux\_version/install\_netbackup-client

wget -O /target/var/lib/firstboot/scripts/98-add\_admin-ckbs\_user.sh http://$http\_server/$release/$linux\_distro/$linux\_version/add\_admin-ckbs\_user

>Сохраняем файл

cinder-c2.conf

**>Правим: /var/www/html/stable/astra/1.6/cinder-c2**

DEFAULT]

host = volume.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

#cluster = volume.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

#enable\_v2\_api=true

#enable\_v3\_api=true

osapi\_volume\_workers=3

rootwrap\_config = /etc/cinder/rootwrap.conf

api\_paste\_confg = /etc/cinder/api-paste.ini

target\_helper = tgtadm

transport\_url = rabbit://openstack:accentos@rabbitmq.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

volume\_name\_template = volume-%s

volume\_group = cinder-volumes

verbose = True

auth\_strategy = keystone

state\_path = /var/lib/cinder

lock\_path = /var/lock/cinder

volumes\_dir = /var/lib/cinder/volumes

my\_ip = keystone

my\_ip = 10.246.192.52 (IP 2 УУ)

#backup\_driver = cinder.backup.drivers.nfs.NFSBackupDriver

#backup\_container =

#backup\_enable\_progress\_timer = True

#backup\_file\_size = 134217728

#backup\_mount\_options =

#backup\_mount\_point\_base = $state\_path/backup\_mount

#backup\_sha\_block\_size\_bytes = 32768

#backup\_share = controller.pvrr.loc:/cinder (Меняем “pvrr” на имя необходимого домена)

#cluster = accentos

enabled\_backends = hitachi2

default\_volume\_type = hitachi2

glance\_api\_version = 2

agent\_response\_timeout = 2888

rpc\_response\_timeout = 2888

use\_syslog = False

max\_logfile\_count = 4

log\_rotation\_type = size

max\_logfile\_size\_mb = 100

[database]

connection = mysql+pymysql://cinder:accentos@mysql.pvrr.loc/cinder (Меняем “pvrr” на имя необходимого домена)

[keystone\_authtoken]

www\_authenticate\_uri = [http://controller.pvrr.loc:5000](http://controller.pvrr.loc:5000/) (Меняем “pvrr” на имя необходимого домена)

auth\_url = [http://controller.pvrr.loc:5000](http://controller.pvrr.loc:5000/) (Меняем “pvrr” на имя необходимого домена)

memcached\_servers = controller.pvrr.loc:11211 (Меняем “pvrr” на имя необходимого домена)

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

project\_name = service

username = cinder

password = accentos

service\_token\_roles\_required = true

service\_token\_roles = admin

[oslo\_concurrency]

lock\_path = /var/lib/cinder/tmp

[hitachi2]

volume\_driver = cinder.volume.drivers.hitachi.hbsd\_fc.HBSDFCDriver

volume\_backend\_name = hitachi2

san\_ip = 10.246.253.137 (IP СХД?)

san\_login = accentos

san\_password = P@ssw0rd

hitachi\_storage\_id = 934000611479

hitachi\_pool = pool1

hitachi\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

hitachi\_compute\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

cinder-c1.conf

**>Правим: /var/www/html/stable/astra/1.6/cinder-c1.conf**

[DEFAULT]

host = volume.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

#cluster = volume.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

#enable\_v2\_api=true

#enable\_v3\_api=true

osapi\_volume\_workers=3

rootwrap\_config = /etc/cinder/rootwrap.conf

api\_paste\_confg = /etc/cinder/api-paste.ini

target\_helper = tgtadm

transport\_url = rabbit://openstack:accentos@rabbitmq.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

volume\_name\_template = volume-%s

volume\_group = cinder-volumes

verbose = True

auth\_strategy = keystone

state\_path = /var/lib/cinder

lock\_path = /var/lock/cinder

volumes\_dir = /var/lib/cinder/volumes

my\_ip = keystone

my\_ip = 10.246.192.51 (IP 1 УУ)

#backup\_driver = cinder.backup.drivers.nfs.NFSBackupDriver

#backup\_container =

#backup\_enable\_progress\_timer = True

#backup\_file\_size = 134217728

#backup\_mount\_options =

#backup\_mount\_point\_base = $state\_path/backup\_mount

#backup\_sha\_block\_size\_bytes = 32768

#backup\_share = controller.pvrr.loc:/cinder (Меняем “pvrr” на имя необходимого домена)

#cluster = accentos

enabled\_backends = hitachi2

default\_volume\_type = hitachi2

glance\_api\_version = 2

agent\_response\_timeout = 2888

rpc\_response\_timeout = 2888

use\_syslog = False

max\_logfile\_count = 4

log\_rotation\_type = size

max\_logfile\_size\_mb = 100

[database]

connection = mysql+pymysql://cinder:accentos@mysql.pvrr.loc/cinder (Меняем “pvrr” на имя необходимого домена)

[keystone\_authtoken]

www\_authenticate\_uri = [http://controller.pvrr.loc:5000](http://controller.pvrr.loc:5000/) (Меняем “pvrr” на имя необходимого домена)

auth\_url = [http://controller.pvrr.loc:5000](http://controller.pvrr.loc:5000/) (Меняем “pvrr” на имя необходимого домена)

memcached\_servers = controller.pvrr:11211 (Меняем “pvrr” на имя необходимого домена)

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

project\_name = service

username = cinder

password = accentos

service\_token\_roles\_required = true

service\_token\_roles = admin

[oslo\_concurrency]

lock\_path = /var/lib/cinder/tmp

[hitachi2]

volume\_driver = cinder.volume.drivers.hitachi.hbsd\_fc.HBSDFCDriver volume\_backend\_name = hitachi2

san\_ip = 10.246.253.137 (IP СХД?)

san\_login = accentos

san\_password = P@ssw0rd

hitachi\_storage\_id = 934000611479

hitachi\_pool = pool1

hitachi\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

hitachi\_compute\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

cinder-c3.conf

**>Правим: /var/www/html/stable/astra/1.6/cinder-c3.conf**

[DEFAULT]

host = volume.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

#cluster = volume.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

#enable\_v2\_api=true

#enable\_v3\_api=true

osapi\_volume\_workers=3

rootwrap\_config = /etc/cinder/rootwrap.conf

api\_paste\_confg = /etc/cinder/api-paste.ini

target\_helper = tgtadm

transport\_url = rabbit://openstack:accentos@rabbitmq.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

volume\_name\_template = volume-%s

volume\_group = cinder-volumes

verbose = True

auth\_strategy = keystone

state\_path = /var/lib/cinder

lock\_path = /var/lock/cinder

volumes\_dir = /var/lib/cinder/volumes

my\_ip = keystone

my\_ip = 10.246.192.53 (IP 3 УУ)

#backup\_driver = cinder.backup.drivers.nfs.NFSBackupDriver

#backup\_container =

#backup\_enable\_progress\_timer = True

#backup\_file\_size = 134217728

#backup\_mount\_options =

#backup\_mount\_point\_base = $state\_path/backup\_mount

#backup\_sha\_block\_size\_bytes = 32768

#backup\_share = controller.pvrr.loc:/cinder (Меняем “pvrr” на имя необходимого домена)

#cluster = accentos

enabled\_backends = hitachi2

default\_volume\_type = hitachi2

glance\_api\_version = 2

agent\_response\_timeout = 2888

rpc\_response\_timeout = 2888

use\_syslog = False

max\_logfile\_count = 4

log\_rotation\_type = size

max\_logfile\_size\_mb = 100

[database]

connection = mysql+pymysql://cinder:accentos@mysql.pvrr.loc/cinder (Меняем “pvrr” на имя необходимого домена)

[keystone\_authtoken]

www\_authenticate\_uri = [http://controller.pvrr.loc:5000](http://controller.pvrr.loc:5000/) (Меняем “pvrr” на имя необходимого домена)

auth\_url = [http://controller.pvrr.loc:5000](http://controller.pvrr.loc:5000/) (Меняем “pvrr” на имя необходимого домена)

memcached\_servers = controller.pvrr.loc:11211 (Меняем “pvrr” на имя необходимого домена)

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

project\_name = service

username = cinder

password = accentos

service\_token\_roles\_required = true

service\_token\_roles = admin

[oslo\_concurrency]

lock\_path = /var/lib/cinder/tmp

[hitachi2]

volume\_driver = cinder.volume.drivers.hitachi.hbsd\_fc.HBSDFCDriver

volume\_backend\_name = hitachi2

san\_ip = 10.246.253.137 (IP СХД?)

san\_login = accentos

san\_password = P@ssw0rd

hitachi\_storage\_id = 934000611479

hitachi\_pool = pool1

hitachi\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

hitachi\_compute\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

createdom.sh

**>Правим: /var/www/html/stable/astra/1.6/createdom.sh**

#!/bin/bash

#add domains, projects for AD and freeipa

source /root/admin-openrc

DOMAINAD="pvrr" (Меняем “pvrr” на имя необходимого домена)

DOMAINFREEIPA="freepvrr" (Меняем “pvrr” на имя необходимого домена)

USERAD="pvrr\_virazh\_admin" (Меняем “pvrr” на имя необходимого домена)

USERFREEIPA="pvrr\_virazh\_admin" (Меняем “pvrr” на имя необходимого домена)

PROJECTAD="pvrradmin" (Меняем “pvrr” на имя необходимого домена)

PROJECTFREEIPA="freepvrradmin" (Меняем “pvrr” на имя необходимого домена)

FILEAD="[http://$COBBLER\_IP/init/keystone/keystone.pvrr.conf](about:blank)" (Этот файл фильтрации будут создаваться. Меняем “pvrr” на имя необходимого домена)

FILEFREEIPA="[http://$COBBLER\_IP/init/keystone/keystone.freepvr.conf](about:blank)" (Этот файл фильтрации будут создаваться. Меняем “pvrr” на имя необходимого домена)

FILEKEYSTONE= "http://$COBBLER\_IP/init/keystone/keystone.cfg"

wget -O /etc/haproxy/conf.d/keystone.cfg $FILEKEYSTONE sleep 3

systemctl reload haproxy

wget -O /etc/keystone/domains/keystone.pvrr.conf $FILEAD (Меняем “pvrr” на имя необходимого домена)

sleep 3

wget -O /etc/keystone/domains/keystone.freepvrr.conf $FILEFREEIPA (Меняем “pvrr” на имя необходимого домена)

sleep 4

systemctl restart keystone

sleep 10

openstack domain create $DOMAINAD

openstack user list --domain $DOMAINAD

openstack project create --domain $DOMAINAD $PROJECTAD

openstack role add --user-domain $DOMAINAD --project-domain $DOMAINAD

--project $PROJECTAD --user $USERAD admin

openstack role add --user-domain $DOMAINAD --user $USERAD --domain

$DOMAINAD admin

sleep 2

openstack domain create $DOMAINFREEIPA

openstack user list --domain $DOMAINFREEIPA

openstack project create --domain $DOMAINFREEIPA $PROJECTFREEIPA

openstack role add --user-domain $DOMAINFREEIPA --project-domain

$DOMAINFREEIPA --project $PROJECTFREEIPA --user $USERFREEIPA admin

openstack role add --user-domain $DOMAINFREEIPA --user $USERFREEIPA

--domain $DOMAINFREEIPA admin

sleep 4

systemctl restart keystone

install\_netbackup-client

**>Правим: /var/www/html/stable/astra/1.6/install\_netbackup-client**

#!/bin/bash

export

NETBACKUP\_CLIENT\_URL=[http://$COBBLER\_HOST/netbackup/NetBackup\_8.1. \_Deb](about:blank) ian\_client.tar.gz

export NETBACKUP\_TOKEN=PEKPEAPGNJMQHBYR (вставляем новый ключ токена)

export NETBACKUP\_MASTER\_SERVER=pvrr-nbcmst-01.pvrr.oao.rzd (Указываем имя согласно полученным данным от РЖД)

export NETBACKUP\_MEDIA\_SERVER0=pvrr-nbmst-01.pvrr.oao.rzd (Указываем имя согласно полученным данным от РЖД)

export NETBACKUP\_MEDIA\_SERVER1=pvrr-nbmst-02.pvrr.oao.rzd (Указываем имя согласно полученным данным от РЖД)

export NETBACKUP\_MEDIA\_SERVER2=pvrr-nbmedia-01.pvrr.oao.rzd (Указываем имя согласно полученным данным от РЖД)

export NETBACKUP\_MEDIA\_SERVER3=pvrr-nbmedia-02.pvrr.oao.rzd (Указываем имя согласно полученным данным от РЖД)

#export NETBACKUP\_MEDIA\_SERVER4=msk-nbmedia-03.msk.oao.rzd

#export NETBACKUP\_MEDIA\_SERVER5=msk-nbmedia-04.msk.oao.rzd

#export NETBACKUP\_MEDIA\_SERVER6=msk-nbmedia-05.msk.oao.rzd

#export NETBACKUP\_MEDIA\_SERVER7=msk-nbmedia-06.msk.oao.rzd

#export NETBACKUP\_MEDIA\_SERVER8=msk-nbmedia-07.msk.oao.rzd

#export NETBACKUP\_MEDIA\_SERVER9=msk-nbmedia-08.msk.oao.rzd

#export NETBACKUP\_MEDIA\_SERVER10=msk-nbmedia-09.msk.oao.rzd

#export NETBACKUP\_MEDIA\_SERVER11=msk-nbmedia-10.msk.oao.rzd

export NETBACKUP\_DOMAIN=pvrr.oao.rzd (Указываем имя согласно полученным данным от РЖД)

hostname=$(hostname | awk -F"." '{print $1}')

hostname=$(echo $hostname.$NETBACKUP\_DOMAIN)

echo "#!/usr/bin/expect -f

set timeout -1

set master\_server [lindex \$argv 0]

set authorization\_token [lindex \$argv 0]

set hostname [lindex \$argv 2]

set license\_key [lindex \$argv 3]

log\_file /tmp/nbexpect.log

spawn ./install

expect {

\"Do you wish to continue\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Do you want to reinstall NetBackup and Media Manager\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Do you want to install the NetBackup client software for this client\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Is this correct\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Would you like to use\" {

sleep 1

send \"n\\n\"

exp\_continue

}

\"Enter the name of this NetBackup client\" {

sleep 1

send \"\$hostname\\n\"

exp\_continue

}

\"Do you still want to terminate all processes\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Do you want this script to attempt to kill them\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Enter the name of this NetBackup client\" {

sleep 1

send \"\$hostname\\n\"

exp\_continue

}

\"Do you still want to terminate all processes\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Do you want this script to attempt to kill them\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"with the current active NetBackup license key\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Do you want to add additional license keys now\" {

sleep 1

send \"n\\n\"

exp\_continue

}

\"Is this host the master server\" {

sleep 1

send \"n\\n\"

exp\_continue

}

\"NetBackup server name of this machine\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"What is the fully qualified name of the master server\" {

sleep 1

send \"\$master\_server\\n\"

exp\_continue

}

\"Enter the name of the NetBackup master server\" {

sleep 1

send \"\$master\_server\\n\"

exp\_continue

}

\"Do you want to install NetBackup and Media Manager files\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Do you want to upgrade NetBackup and Media Manager\" {

sleep 1

send \"y\\n\"

exp\_continue

}

\"Java GUI and JRE option\" {

sleep 1

send \"2\\n\"

exp\_continue

}

\"Do you wish to proceed with installation even though NetBackup will not operate\" {

sleep 1

send \"y\\n\"

exp\_continue

}

}

" > /tmp/nb.expect

mkdir -pv /tmp/firstboot\_netbackup

wget -q -O /tmp/firstboot\_netbackup/NetBackup\_client.tar.gz

$NETBACKUP\_CLIENT\_URL

PWD=$(pwd)

cd /tmp/firstboot\_netbackup

tar -xf NetBackup\_client.tar.gz

rm -f NetBackup\_client.tar.gz

DIR=$(ls)

cd $DIR

echo expect /tmp/nb.expect $NETBACKUP\_MASTER\_SERVER $NETBACKUP\_TOKEN

$hostname

expect /tmp/nb.expect $NETBACKUP\_MASTER\_SERVER $NETBACKUP\_TOKEN

$hostname

cd $PWD

i=0

media\_count=0

while [ "$i" -le "1000" ]

do

echo "export NETBACKUP\_MEDIA\_SERVER=\$NETBACKUP\_MEDIA\_SERVER$i"

> /tmp/install\_media\_list

source /tmp/install\_media\_list

if [ ! -z $NETBACKUP\_MEDIA\_SERVER ]; then

echo "MEDIA\_SERVER = $NETBACKUP\_MEDIA\_SERVER" >>

/usr/openv/netbackup/bp.conf

fi

let i=$i+1

done

Подключение СХД

Запускаем 1 УУ. Сохранение с 1УУ файла /etc/cinder/cinder.conf и /etc/multipath/wwids

sudo nmap -sn 172.20.0.21-50 > mac\_address\_hosts

Копирование cinder.conf в /var/www/html/stable/astra/1.6/cinder-c1(2,3).conf (проверка ip и dns)

Вставляем ID СХД

**Настройка Cinder для хранилища Hitachi VSP E590**

Настройка производилась на AstraLinux 1.6 Smolensk, Cinder Train.

а. Сначала необходимо скачать драйвер для Cinder Victoria: https://github.com/openstack/cinder/tree/stable/victoria/cinder/volume/drivers/hitachi.

б. Все скачанные файлы помещаем в папку /usr/lib/python3/dist-packages/cinder/volume/drivers/hitachi.

Затем нужно настроить конфигурационный файл cinder.conf:

а. Создаем секцию [hitachi] с параметрами:

[hitachi]

volume\_driver = cinder.volume.drivers.hitachi.hbsd\_fc.HBSDFCDriver

volume\_backend\_name = hitachi

san\_ip = storageIP

san\_login = storageLogin

san\_password = storagePassword

hitachi\_storage\_id = 930003498765

hitachi\_pool = pool1

hitachi\_target\_ports=CL1-A,CL2-A,CL3-A,CL4-A,CL5-A,CL6-A,CL7-A,CL8-A

hitachi\_compute\_target\_ports=CL1-A,CL2-A,CL3-A,CL4-A,CL5-A,CL6-A,CL7-A,CL8-A

suppress\_requests\_ssl\_warnings=True

Получение ID СХД

``**hitachi\_storage\_id**`` можно получить выполнив запрос: **https://hostname/ConfigurationManager/v1/objects/storages**. Длина его значения должна быть равна 12 символов. Последние цифры – это серийный номер.

``**hitachi\_target\_ports`**` – порты СХД, через которые будут презентоваться нарезанные диски для хоста Cinder.

``**hitachi\_compute\_target\_ports**`` – это порты СХД, через которые будут презентоваться нарезанные диски для виртуального узла.

cinder.conf

**>Правим: /var/www/html/init/cinder/cinder.conf**

[DEFAULT]

host = volume.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

#cluster = volume.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

#enable\_v2\_api=true

#enable\_v3\_api=true

osapi\_volume\_workers=3

rootwrap\_config = /etc/cinder/rootwrap.conf

api\_paste\_confg = /etc/cinder/api-paste.ini

target\_helper = tgtadm

transport\_url = rabbit://openstack:accentos@rabbitmq.pvrr.loc (Меняем “pvrr” на имя необходимого домена)

volume\_name\_template = volume-%s

volume\_group = cinder-volumes

verbose = True

auth\_strategy = keystone

state\_path = /var/lib/cinder

lock\_path = /var/lock/cinder

volumes\_dir = /var/lib/cinder/volumes

#my\_ip = keystone

my\_ip = 10.246.192.51 (IP 1УУ)

#backup\_driver = cinder.backup.drivers.nfs.NFSBackupDriver

#backup\_container =

#backup\_enable\_progress\_timer = True

#backup\_file\_size = 134217728

#backup\_mount\_options =

#backup\_mount\_point\_base = $state\_path/backup\_mount

#backup\_sha\_block\_size\_bytes = 32768

#backup\_share = controller.pvrr.loc:/cinder (Меняем “pvrr” на имя необходимого домена)

#cluster = accentos

enabled\_backends = hitachi2,hitachim

default\_volume\_type = hitachi2

glance\_api\_version = 2

agent\_response\_timeout = 2888

rpc\_response\_timeout = 2888

use\_syslog = False

max\_logfile\_count = 4

log\_rotation\_type = size

max\_logfile\_size\_mb = 100

[database]

connection = mysql+pymysql://cinder:accentos@mysql.pvrr.loc/cinder (Меняем “pvrr” на имя необходимого домена)

[keystone\_authtoken]

www\_authenticate\_uri = [http://controller.pvrr.loc:5000](http://controller.pvrr.loc:5000/) (Меняем “pvrr” на имя необходимого домена)

auth\_url = [http://controller.pvrr.loc:5000](http://controller.pvrr.loc:5000/) (Меняем “pvrr” на имя необходимого домена)

memcached\_servers = controller.pvrr.loc:11211 (Меняем “pvrr” на имя необходимого домена)

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

project\_name = service

username = cinder

password = accentos

service\_token\_roles\_required = true

service\_token\_roles = admin

[oslo\_concurrency]

lock\_path = /var/lib/cinder/tmp

[hitachi2]

volume\_driver = cinder.volume.drivers.hitachi.hbsd\_fc.HBSDFCDriver

volume\_backend\_name = hitachi2

san\_ip = 10.246.253.137 (Первый IP СХД)

san\_login = accentos

san\_password = P@ssw0rd

hitachi\_storage\_id = 934000611479 ( Получение номера ID СХД описал выше)

hitachi\_pool = pool1

hitachi\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

hitachi\_compute\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

suppress\_requests\_ssl\_warnings = True

[hitachim]

volume\_driver = cinder.volume.drivers.hitachi.hbsd\_fc.HBSDFCDriver

volume\_backend\_name = hitachim

san\_ip = 10.246.253.137 (Здесь должен быть второй IP СХД)

san\_login = accentos

san\_password = P@ssw0rd

hitachi\_storage\_id = 934000611479

hitachi\_pool = pool1

hitachi\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

hitachi\_compute\_target\_ports=CL3-A,CL4-A,CL7-A,CL8-A

suppress\_requests\_ssl\_warnings = True

keystone.cfg

**>Правим файл: /var/www/html/init/keystone/keystone.cfg**

!!! Необходимо указать все наши доменные сервера, которые будут использоваться в дальнейшем.

listen ldap

bind 127.0.0.1:389

mode tcp

balance source

option tcpka

option ldap-check

server pvrr-dc-01.pvrr.oao.rzd 10.114.4.105:389 check fall 1 rise 1 inter 2s

server pvrr-dc-02.pvrr.oao.rzd 10.114.4.63:389 check fall 1 rise 1 inter 2s

server pvrr-dc-03.pvrr.oao.rzd 10.114.4.103:389 check fall 1 rise 1 inter 2s

server pvrr-dc-04.pvrr.oao.rzd 10.114.4.104:389 check fall 1 rise 1 inter 2s

listen freeipa

bind 127.0.2.2:389

mode tcp

balance source

option tcpka

option ldap-check

server dc1-pvrr.crp.rzd 10.114.10.10:389 check fall 1 rise 1 inter 2s

server dc2-pvrr.crp.rzd 10.114.10.11:389 check fall 1 rise 1 inter 2s

Создаем 2 файла конфигурации keystone:

Keystone.[free IPA].conf

**Создаем файл конфигурации keystone.[free IPA].conf:**

**/var/www/html/init/keystone/keystone.freep\*\*\*.conf (Имя файла указываем такое же как и в файле createdom.sh)**

**ПРИМЕР: /var/www/html/init/keystone/keystone.freepvrr.conf**

**(Помощь при настройке фильтрации можно посмотреть тут:** [**https://confluence.atlassian.com/kb/how-to-write-ldap-search-filters-792496933.html**](https://confluence.atlassian.com/kb/how-to-write-ldap-search-filters-792496933.html) **)**

[identity]

driver = ldap

[ldap]

url = "ldap://127.0.2.2, ldap://10.114.10.10, ldap://10.114.10.11" (Указываем данные РДЖ)

user = "uid=pvrr\_virazh\_admin,cn=users,cn=accounts,dc=crp,dc=rzd" (указываем необходимую доменную зону)

password = Zz1234567890 (Указываем полученный пароль от юзера)

suffix = dc=crp,dc=rzd

use\_dumb\_member = False

allow\_subtree\_delete = False

user\_filter = (memberOf=cn=pvrr\_virazh\_user\*,cn=group,cn=accounts,dc=crp,dc=rzd) (указываем необходимую доменную зону)

user\_tree\_dn = "cn=users,cn=accounts,dc=crp,dc=rzd"

user\_objectclass = inetorgperson

#group\_tree\_dn = "uid=VIRAZH,cn=Service Groups,DC=crp,DC=rzd"

#group\_objectclass = group

user\_allow\_create = False

user\_allow\_update = False

user\_allow\_delete = False

#group\_allow\_create = False

#group\_allow\_update = False

#group\_allow\_delete = False

user\_id\_attribute = uid

user\_name\_attribute = uid

user\_mail\_attribute = mail

user\_pass\_attribute = userPassword

#group\_id\_attribute = cn

#group\_name\_attribute = cn

#group\_member\_attribute = member

#group\_desc\_attribute = description

#group\_filter =

#user\_enabled\_default = 888888

page\_size = 500

Keystone.[dns MS AD].conf

**Создаем файл keystone.[dns MS AD]:**

**/var/www/html/init/keystone/keystone.\*\*\*.conf (Имя файла указываем такое же как и в файле createdom.sh)**

**ПРИМЕР: /var/www/html/init/keystone/keystone.pvrr.conf**

[identity]

driver = ldap

[ldap]

password = 'sLya9ydS%^M@' (Указываем пароль админа полученный от РЖД)

suffix = dc=pvrr,dc=oao,dc=rzd (указываем необходимую доменную зону)

use\_dumb\_member = False

allow\_subtree\_delete = False

user\_tree\_dn = "ou=VIRAZH,ou=Service Accounts,dc=pvrr,dc=oao,dc=rzd" (указываем необходимую доменную зону)

user\_objectclass = person

group\_tree\_dn = "OU=VIRAZH,OU=Service Groups,DC=pvrr,DC=oao,DC=rzd" (указываем необходимую доменную зону)

group\_objectclass = group

user\_allow\_create = False

user\_allow\_update = False

user\_allow\_delete = False

group\_allow\_create = False

group\_allow\_update = False

group\_allow\_delete = False

user\_id\_attribute = sAMAccountName

user\_name\_attribute = sAMAccountName

user\_mail\_attribute = mail

user\_pass\_attribute = userPassword

group\_id\_attribute = cn

group\_name\_attribute = cn

group\_member\_attribute =

group\_desc\_attribute = description

group\_filter =

page\_size = 1000

Проверка системы:

Перед окончанием преднастройки и тестовым запуском Cobbler надо добавить проверку GREPом с корня папки со скриптами на наличие:

1) имен предыдущих доменов, начиная с free<Free\_IPA\_domain\_name>, потом уже переходя <AD\_domain\_name>

2) сетей и подсетей предыдущих ивц

*grep -r msk /var/www/html/stable/astra/1.6/*

*grep -r pvrr /var/www/html/stable/astra/1.6/*

Убежаемся что в файлах **initrc\_ , initrc\_2 , initrc\_controller, *astra\_node, cinder-c1.conf, cinder-c2.conf, cinder-c3.conf,* createdom.sh, install\_netbackup-client** отсутствуют данные “msk” и “pvrr”

*grep -r 10.116.192. /var/www/html/stable/astra/1.6/*

Убежаемся что в файлах **initrc\_ , initrc\_2 , initrc\_controller, *astra\_node, cinder-c1.conf, cinder-c2.conf, cinder-c3.conf,* createdom.sh, install\_netbackup-client** отсутствуют данные “старой сети”

*grep -r msk /var/www/html/init/*

*grep -r pvrr /var/www/html/init/*

Убежаемся что в файлах **cinder.conf, keystone.cfg** отсутствуют данные “msk” и “pvrr”

Убеждаемся, что в поиске присутствуют файлы созданные ранее:  **keystone.[free IPA].conf keystone.[dns MS AD]**

*grep -r 10.246.192 /var/www/html/init/*

Убежаемся что в файлах **cinder.conf, keystone.cfg** отсутствуют данные “старой сети”

Убедится что в файле  **/var/www/html/stable/astra/1.6/install\_bind9-forward-zone**

Прописана dns зона необходимого РЖД

Копирование cinder.conf в /var/www/html/stable/astra/1.6/cinder-c1(2,3).conf

(проверка ip и dns)

форматирование lun под ocfs2 на 1 УУ и последнем ВУ

NOVA - mkfs.ocfs2 --cluster-stack=o2cb –cluster-name=ocfs2node1(2)

/dev/mapper/WWIN -N 32 —global-heartbeat

Glance и Cnocchi mkfs.ocfs2 --cluster-stack=o2cb –cluster-name=ocfs2ctrl

/dev/mapper/WWIN -N 4 —global-heartbeat

curl GET https://IP:23451/ConfigurationManager/v1/objects/storages

### Настройка Cobbler через GUI

Проверяем доступ по вэбинтерфейсу: Указав свой IP

http://10.40.64.3/cobbler\_web/

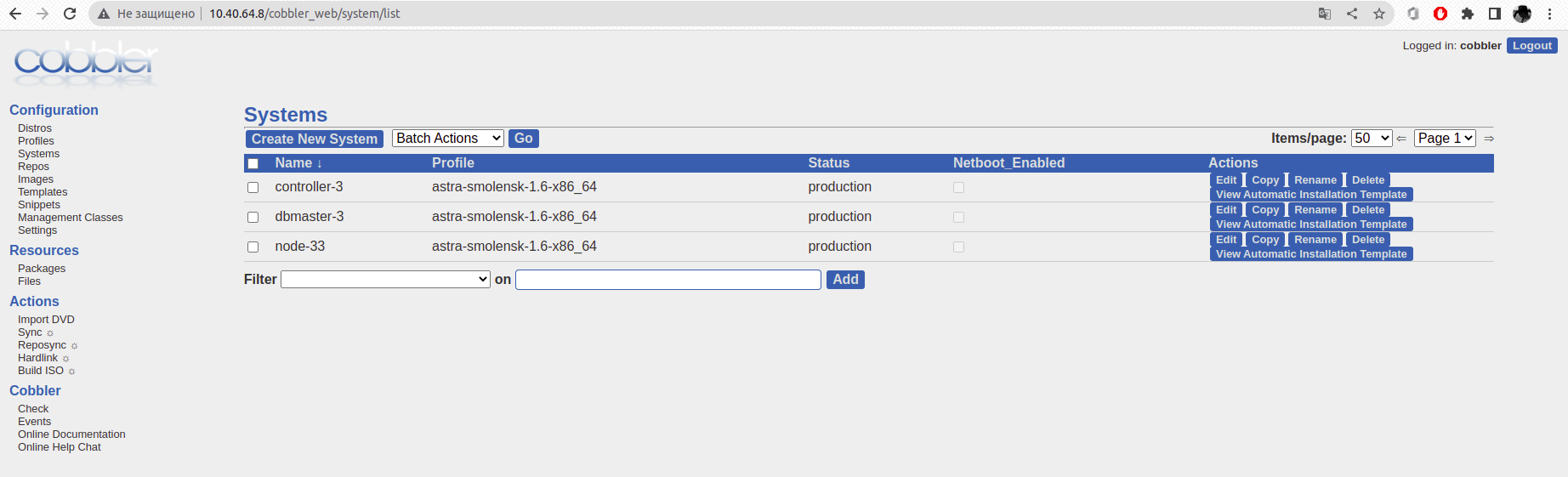
Логин: cobbler

Пароль: cobbler

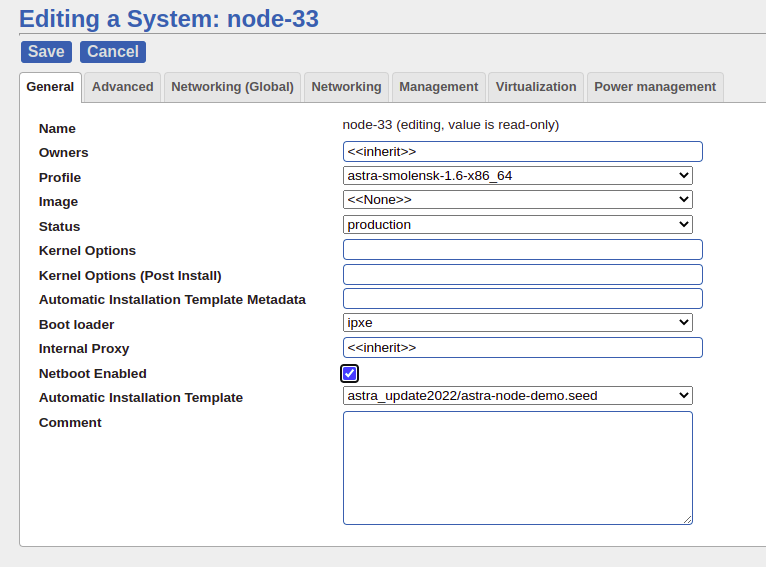
Пройдите авторизацию

**Создаем Вычислительный узел (ВУ)**

Перейдите в раздел Systems и нажмите кнопку **“Create New System**”:

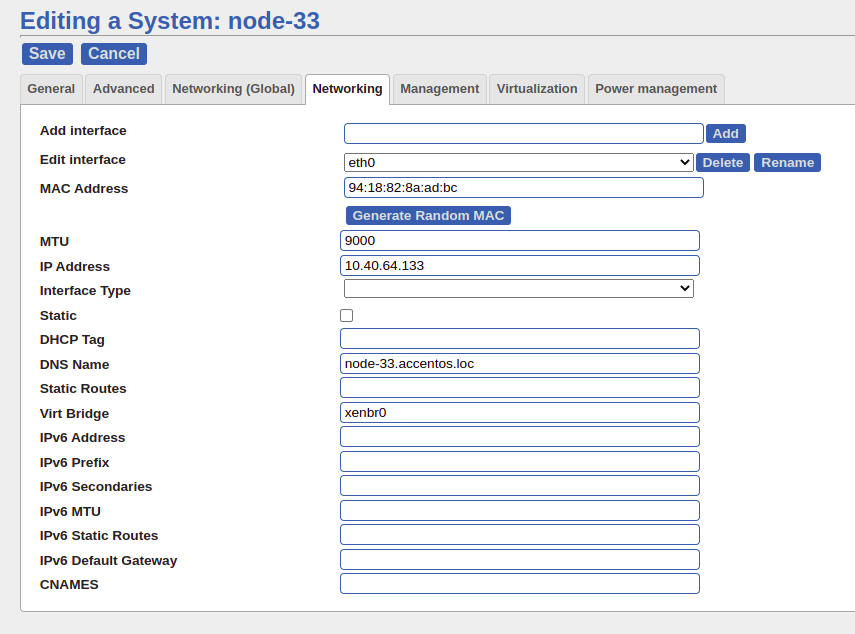


Вкладка **General**



1. **Name** - прописываем имя (имя может быть любое)
2. **Profile** - выбираем astra-smolensk-1.6-x86\_64
3. **Status** - выбираем production
4. **Boot loader** - выбираем ipxe
5. **Netboot Enabled** - устанавливаем галку
6. **Automatic Installation Template** - выбираем astra\_update2022/astra-node-demo.seed

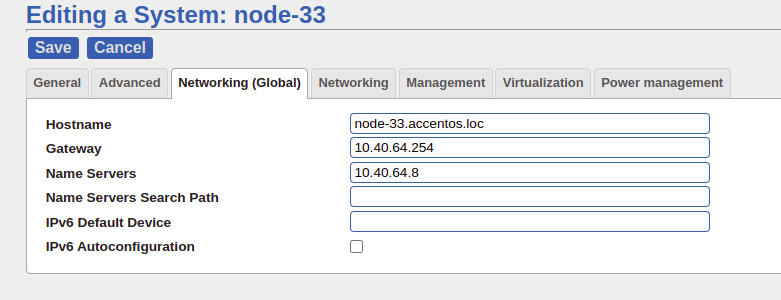
Вкладка **Networking**



* **Add interface** - пишем eth0 нажимаем кнопку Add
* **Edit interface** - тут появится eth0 после нажатия кнопки add в предыдущем шаге
* **MAC Address** - необходимо прописать мак адрес вашего сетевого интерфейса через который ваш сервер подключен к сети (мак можно посмотреть через биос)
* MTU - 9000
* IP Address - IP который вы хотите присвоить вашему серверу
* DNS Name - имя вашего сервера который вы указали во вкладке General точка accentos.loc Например: node-33.accentos.loc
* Virt Bridge - пишем xenbr0

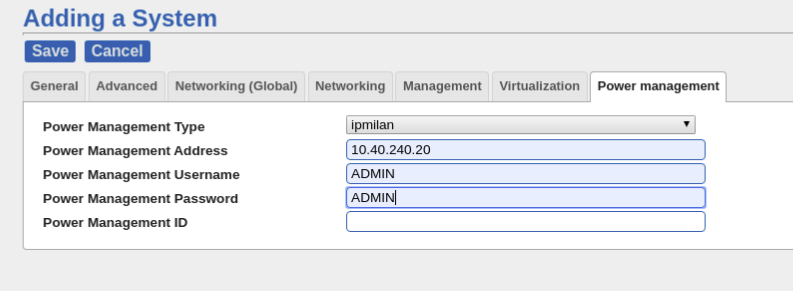
Остальные поля заполнять не нужно, так как параметры будут определены через файл ответов.

Вкладка **Networking(Global)**



* **Hostname** - имя вашего сервера который вы указали во вкладке General точка accentos.loc Например: node-33.accentos.loc
* **Gateway** - шлюз сети, в которой будет находиться хост. Который вы указали в файле /etc/cobbler/dhcp.template на строке option routers
* **Name Servers** - IP вашего сервера гоблер, который вы указывали в файле /etc/network/interface

В последнюю очередь заполняем вкладку “**Power management**”. Тут существует много вариантов, но мы рассмотрим только два: IPMI и KVM.



В случае IPMI выбираем:

* “**Power Management Type**” - ipmilan;
* “**Power Management Address**” - IP-адрес управления сервером;
* “**Power Management Username**” - имя пользователя;
* **“Power Management Password”** - пароль;
* “**Power Management ID**” - порт IPMI, если он настроен не на стандартное значение.

В случае KVM выбираем:

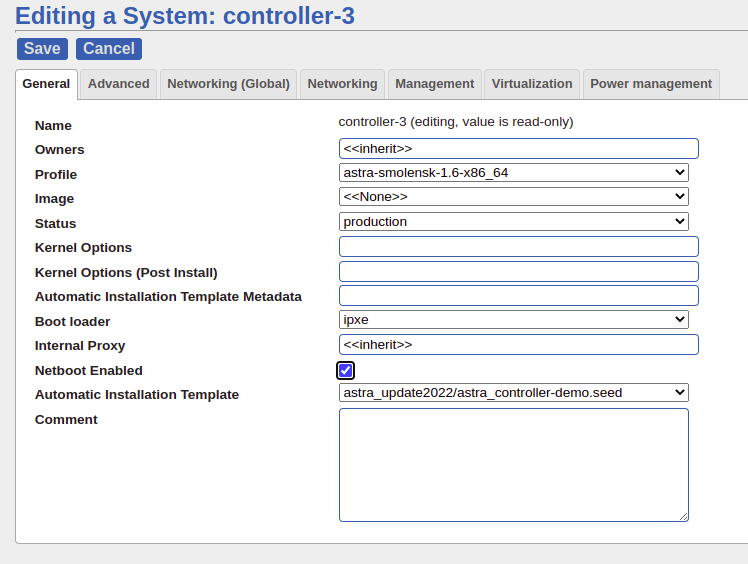
1. “**Power Management Type**” - virsh;
2. “**Power Management Address**” - IP-адрес управления сервером;
3. “**Power Management Username**” - имя пользователя;
4. “**Power Management Password**” - пароль;
5. “**Power Management ID**” - порт libvirt, если он настроен не на стандартное значение. Libvirt должен принимать подключения по сети.

Сохраняем настройки нажав кнопку **Save**

**Создаем контроллер.**

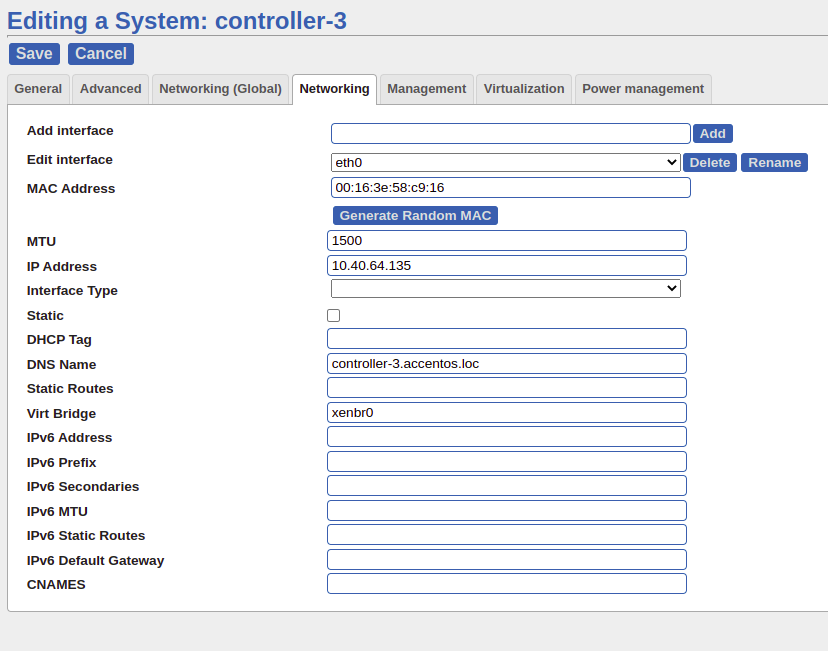
Перейдите в раздел Systems и нажмите кнопку “**Create New System**”:

Вкладка **General**



1. **Name** - прописываем имя (имя может быть любое)
2. **Profile** - выбираем astra-smolensk-1.6-x86\_64
3. **Status** - выбираем production
4. **Boot loader** - выбираем ipxe
5. **Netboot Enabled** - устанавливаем галку
6. **Automatic Installation Template** - выбираем astra\_update2022/astra\_controller-demo.seed

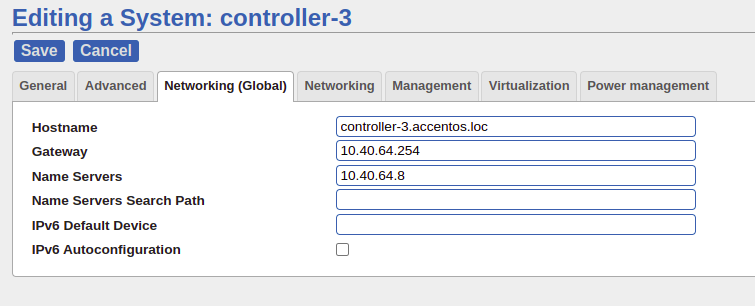
Вкладка **Networking**



* **Add interface** - пишем eth0 нажимаем кнопку Add
* **Edit interface** - тут появится eth0 после нажатия кнопки add в предыдущем шаге
* **MAC Address** - нажимаем General Random MAC
* **MTU** - 1500
* **IP Address** - IP из той же самой подсети что и ВУ
* **DNS Name** - имя вашего сервера который вы указали во вкладке General точка accentos.loc Например: controller-3.accentos.loc
* **Virt Bridge** - пишем xenbr0

Остальные поля заполнять не нужно, так как параметры будут определены через файл ответов.

Вкладка **Networking(Global)**



* **Hostname** - имя вашего сервера который вы указали во вкладке General точка accentos.loc Например: controller-3.accentos.loc
* **Gateway** - шлюз сети, в которой будет находиться хост. Который вы указали в файле /etc/cobbler/dhcp.template на строке option routers
* **Name Servers** - IP вашего сервера гоблер, который вы указывали в файле /etc/network/interface

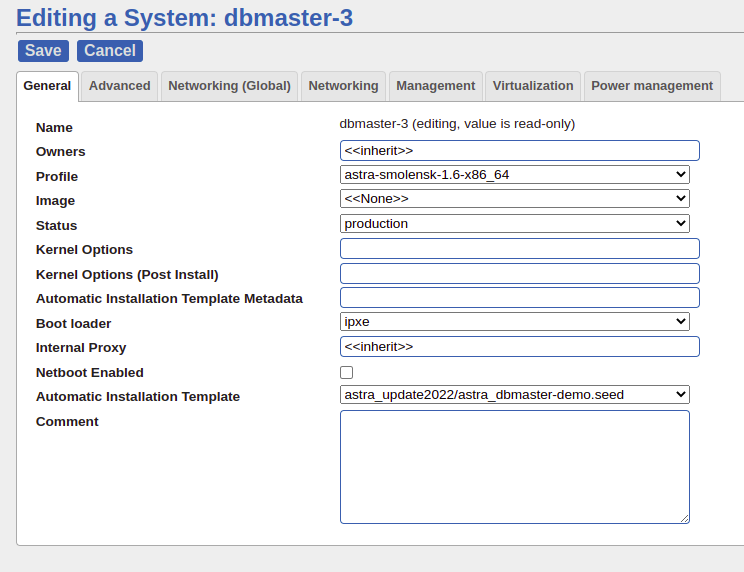
Вкладка **Power management** по аналогии с ВУ

Сохраняем настройки нажав кнопку **Save**

**Создаем Базу данных (БД)**

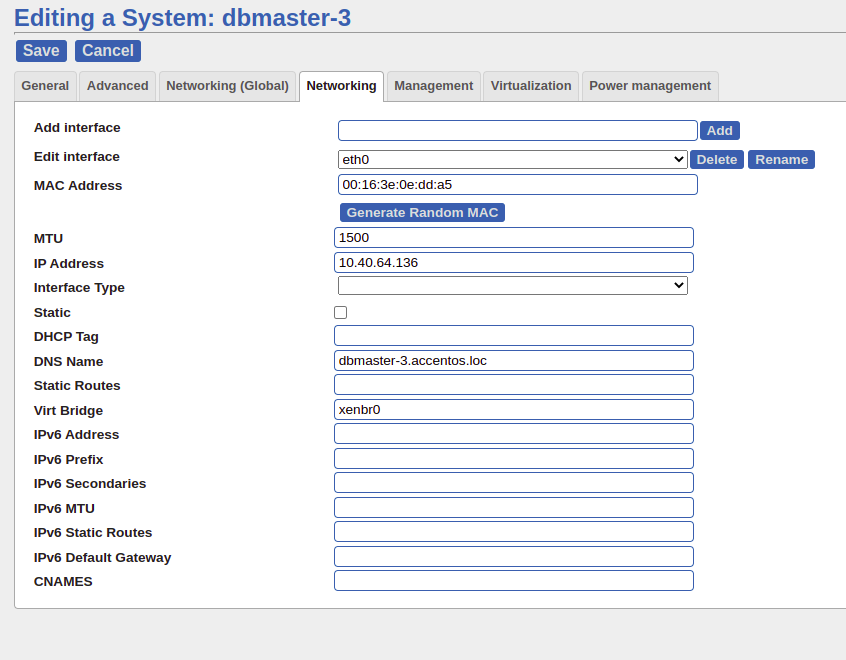
Перейдите в раздел Systems и нажмите кнопку “**Create New System**”:

Вкладка **General**



1. **Name** - прописываем имя (имя может быть любое)
2. **Profile** - выбираем astra-smolensk-1.6-x86\_64
3. **Status** - выбираем production
4. **Boot loader** - выбираем ipxe
5. **Netboot Enabled** - устанавливаем галку
6. **Automatic Installation Template** - выбираем astra\_update2022/astra-dbmaster-demo.seed

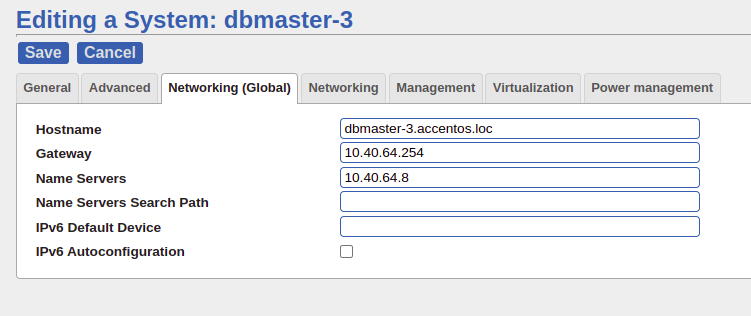
Вкладка **Networking**



* **Add interface** - пишем eth0 нажимаем кнопку Add
* **Edit interface** - тут появится eth0 после нажатия кнопки add в предыдущем шаге
* **MAC Address** - нажимаем General Random MAC
* **MTU** - 1500
* **IP Address** - IP из той же самой подсети что и ВУ
* **DNS Name** - имя вашего сервера который вы указали во вкладке General точка accentos.loc Например: dbmaster-3.accentos.loc
* **Virt Bridge** - пишем xenbr0

Остальные поля заполнять не нужно, так как параметры будут определены через файл ответов.

Вкладка **Networking(Global)**



* **Hostname** - имя вашего сервера который вы указали во вкладке General точка accentos.loc Например: controller-3.accentos.loc
* **Gateway** - шлюз сети, в которой будет находиться хост. Который вы указали в файле */etc/cobbler/dhcp.template* на строке option routers
* **Name Servers** - IP вашего сервера гоблер, который вы указывали в файле */etc/network/interface*

Вкладка **Power management** по аналогии с ВУ

Сохраняем настройки нажав кнопку **Save**

Синхронизируем данные нажав кнопку **Sync**

### Проверка корректности настроек перед автоустановкой облака

Для установки обпака AccentOS на подготовленном объекте (установлены серверы, СХД, настроена сеть, произведены все настройки) должны быть проведены следующие работы

1. Проверка доступности серверов по IPMI в BIOS
2. Проверка доступности СХД к обоим контроллерам
3. Проверка сетевых сервисов DNS, DHCP, NTP
4. Проверка правильности предоставленных логинов/паролей для доступа Samba, администраторов, пользователей для MS AD
5. Проверка правильности предоставленных логинов/паролей для доступа Samba, администраторов, пользователей для Free IPA

1.2 Настроить первый сервер в стойке ВИРАЖ на сеть \*\*kvm\_internal\*\* и получить к нему доступ по ssh с \*\*ВМ-Администратора\*\*

- Настроить сеть используя следующие настройки:

/etc/network/interfaces

auto bond0

iface bond0 inet static

bond-slaves eth0 eth1

bond-mode 802.3ad

address <kvm\_internal\_net>.51/21

gateway <kvm\_internal\_net>.1

dns-nameservers <dc1\_\_\_.oao.rzd> <dc1\_\_\_.crp.rzd>

- добавить временный репозиторий

sudo echo "deb [trusted=yes] http://repo-<domain>.crp.rzd/orel stable main contrib non-free" >> /etc/apt/sources.list

- установить дополнительные пакеты

sudo apt update && sudo apt install ldap-utils nmap telnet -y

1.3 Настроить последний сервер в стойке ВИРАЖ на сеть VDI и получить к нему доступ с обоих \*\*АРМ\*\*ов

- Настроить сеть используя следующие настройки:

/etc/network/interfaces

auto bond0

iface bond0 inet manual

bond-slaves eth0 eth1

bond-mode 802.3ad

#добавляем VLAN30

auto bond0.30

iface bond0.30 inet static

vlan-raw-device bond0

address <vdi\_public\_net>.100/21

gateway <vdi\_public\_net>.1

dns-nameservers <dc1\_\_\_.oao.rzd> <dc1\_\_\_.crp.rzd>

- добавить временный репозиторий

sudo echo "deb [trusted=yes] http://repo-<domain>.crp.rzd/orel stable main contrib non-free" >> /etc/apt/sources.list

- установить дополнительные пакеты

sudo apt update && sudo apt install ldap-utils nmap telnet -y

2.1 Проверить доступность DNS и NTP серверов из сети ПАК

- Доступность из сетей \*\*kvm\_internal\*\* и \*\*vdi\_public\*\*

# проверям резолвинг

nslookup <msad-domain>.oao.rzd

nslookup <msad-domain>.oao.rzd

nslookup <msad-domain>.oao.rzd

nslookup <msad-domain>.oao.rzd

nslookup <ipa-domain>.crp.rzd

nslookup <ipa-domain>.crp.rzd

#проверяем доступность NTP

в файл /etc/chrony/chrony.conf добавляем в конец

server <IP\_NTP1> iburst

server <IP\_NTP2> iburst

проверяем доступность командой

chronyc sources

- Проверяем доступность необходимых ресурсов из kvm\_internal (Zabbix, NetBackup, Tivoli)

telnet <IP\_Zabbix> 10051

nmap -Pn <IP\_NetBackup>

nmap -sn <IP\_Tivoly>

# 3. Проверка доступа к доменам MSAD и Freeipa из сетей \*\*kvm\_internal\*\* и \*\*vdi\_public\*\* внутри ПАК:

- Проверяем доступность аккаунта администратор и самого доменного сервера \*\*AD\*\* для предоставленных учетных данных:

ldapsearch -H ldap://<domain\_server\_ip> -D "CN=<domain\_admin\_name>,OU=VIRAZH,OU=Service Accounts,DC=<domain>,DC=oao,DC=rzd" -b "ou=VIRAZH,ou=Service Accounts,dc=<domain>,dc=oao,dc=rzd" "(objectclass=person)" sAMAccountName -w '<domain\_admin\_password>' | grep result # код успешной работы запроса - result: 0 success

- Проверяем доступность аккаунта администратор и самого доменного сервера \*\*FreeIPA\*\* для предоставленных учетных данных:

ldapsearch -w <domain\_admin\_password> -H ldap://<domain\_server\_ip> -D "uid=<domain\_admin\_name>,cn=users,cn=accounts,dc=crp,dc=rzd" -b "cn=group,cn=accounts,dc=crp,dc=rzd" "(objectclass=person)" | grep result # код успешной работы запроса - result: 0 success

**Подготовка разделов на СХД и их форматирование**

- Проверить досуп к [\*\*СХД\*\*] с первого сервера из сети управления kvm\_internal

nmap <IP\_СХД>

- На \*\*ВМ-Администратора\*\* зайти в браузер на [\*\*СХД\*\*] используя следующие учетные данные: accentos/P@ssw0rd

- Убедиться, что во вкладке "POOLS" находятся два пула

- Убедиться, что во вкладке "VOLUMES" находятся именновые \*\*LUN\*\*ы:

- nova1

- nova2

- gnocci

- glance

***Если томов на СХД нет необходимо их создать***

Сохранение с 1УУ файла */etc/cinder/cinder.conf* и /*etc/multipath/wwids*

*sudo nmap -sn 172.20.0.21-50 > mac\_address\_hosts*

Копирование cinder.conf в */var/www/html/stable/astra/1.6/cinder-c1(2,3).conf* (проверка ip и dns)

форматирование lun под ocfs2 на 1 УУ и последнем ВУ

***NOVA - mkfs.ocfs2 --cluster-stack=o2cb --cluster-name=ocfs2node1(2) /dev/mapper/WWIN -N 32 —global-heartbeat***

***Glance и Cnocchi mkfs.ocfs2 --cluster-stack=o2cb --cluster-name=ocfs2ctrl /dev/mapper/WWIN -N 4 —global-heartbeat***

### Автоустановка облака

Работа с FirstBoot приведена в Приложении 2.

Остановить службы cobbler, bind9, isc-dhcp-server

Coobler:

1. Изменение ip адресов и подсети в:

/*etc/network/interfaces*

/*etc/cobbler/settings.yaml*

/*etc/cobbler/named.template*

/*etc/cobbler/dhcp.template*

/*etc/dhcp/dhcpd.conf*

/var/www/html/stable/astra/1.6/initrc\_

/var/www/html/stable/astra/1.6/initrc\_2

/var/www/html/stable/astra/1.6/initrc\_.controller

~~/var/www/html/stable/astra/1.6/install\_bind9~~

/var/lib/cobbler/templates/astra\_update2021/astra\*.seed

2. Добавление и изменение наименования хостов, ip адресов, mac адресов в cobbler web или /*var*/lib/cobbler/collections/systems

3. Изменение в initrc\_ initrc\_2 и initrc\_.controller:

* адреса подсети и обратную зону
* имя локального домена согласно ИВЦ (CLOUD\_ZONE)
* ip адреса cobbler и репозитория
* количество хостов ocfs2
* wwids СХД (OCFS2\_DRIVE = NOVA\_DRIVE, GLANCE\_DRIVE, GNOCCHI\_DRIVE)
* NTP
* ZABBIX\_IP

4. Изменение наименования узлов и их количества в /*var/www/html/astra\_node*

1. Изменение имен доменов в */var/www/html/stable/astra/1.6/createdom.sh*

ПРОВЕРИТЬ С КОРНЯ РЕПЫ НАЛИЧИЕ СТАРЫХ НАСТРОЕК ДОМЕНОВ И ПОДСЕТЕЙ ЧЕРЕЗ `GREP`

ПЕРЕНАСТРОИТЬ СЕТЬ PROVIDER ПОСЛЕ УСТАНОВКИ ОБЛАКА ЛИБО ЖЕ ДОБАВИТЬ ЭТО В СКРИПТ

*6.* Изменение параметров установки агента NetBackup

*/var/www/html/stable/astra/1.6/install\_netbackup-client*

Запустить службы cobbler, bind9, isc-dhcp-server

Произвести синхронизацию cobbler в WebUI или cli

~# cobbler sync

### Загрузка ВМ для системы WebGard и системы Samba

### Установка и настройка ВМ для системы WebGard

Установка и настройка WebGard приведено в инструкции (Приложение 3)

ДОБАВИТЬ ФИКСЫ В ИНСТРУКЦИЮ:

ПРАВИЛО АНТИАФФИНИТИ И СООТВЕТСВУЮЩИЕ ЗОНЫ ПО КЛАСТЕРАМ(ЗОНУ tz)

ДОБАВИТЬ КОРРЕКТНЫЕ ДОПОЛНИТЕЛЬНЫЕ ПОЯСНЕНИЯ ПО НАСТРОЙКЕ СЕТЕВЫХ ИНТЕРФЕЙСОВ

### Установка и настройка ВМ для системы Samba

Установка и настройка ВМ Samba для MS AD приведено в Приложении 4

ДОБАВИТЬ ДОПОЛНИТЕЛЬЫЕ ПРОВЕРКИ В ИНСТРУКЦИЮ ПРОВЕРКИ ИНФРАСТРУКТЫ ПЕРЕД УСТАНОВКОЙ ОБЛАКА, А ИМЕННО ПОРТЫ САМБЫ 445 139 TCP ПОРТЫ

### Настройка брокеров VDI

### Загрузка образов гостевых ВМ для системы Windows и Linux

### Создание проектов для АРМ Windows и АРМ Linux

### Установка ОС Windows на АРМ пользователя и настройка терминального клиента, настройка RSClient

# Необходимое оборудование для проведение проверок

- Веб-камера

- USB-Токен

- Аудио-гарнитура с микрофоном

- Сканер сетевой

- Принтер сетевой

### Установка ОС Linux на АРМ пользователя и настройка терминального клиента, настройка RSClient

# Необходимое оборудование для проведение проверок

- Веб-камера

- USB-Токен

- Аудио-гарнитура с микрофоном

- Сканер сетевой

- Принтер сетевой

### Проверка подлючения АРМ Windows

- АРМ Windows 10:

- Убедиться, что машина в домене, проверив настройки \*\*АРМ\*\*

#ввести в домен под учетными записями выданными РЖД

- Убедиться в работе всех устройств из списка "Необходимое оборудование"

### Проверка подлючения АРМ Linux

C предоставленного \*\*АРМ\*\* с Linux подключиться к сереру настроенному в сети \*\*vdi\_public\*\* по `ssh`, проверить доступность следующих портов:

- 3240(TCP) 5005(UDP) 5007(UDP) 5008(UDP) 6566(TCP) 61000(TCP) - должны быть открыты на \*\*ТК\*\*. Для проверки выполняем следующие команды:

- Проверка доступа с сервера на ТК

- для UDP портов

nc -lu <PORT> # запустить netcat на ТК

nc -u <IP\_TK> <PORT> # запустить netcat на сервере, ввести любой надор символов, убедиться, что пакеты отправляются. Сделать тоже самое для остальных портов

- для TCP портов

nc -l <PORT> # запустить netcat на ТК

nc <IP\_TK> <PORT> # запустить netcat на сервере, ввести любой набор символов, убедиться, что пакеты отправляются. Сделать тоже самое для остальных портов

Провести проверку в обратную сторону с ТК на сервер.

- Убедиться, что установленная ОС удовлетворяет требованиям, а \*\*АРМ\*\* заведен в домен, выполнив следующие комманды:

cat /etc/astra\_version # Версия ОС: CE 2.12.43 (orel)

uname -r # Версия ядра: 5.10.0-1038.40-generic

astra-freeipa-client -i # Обнаружен настроенный клиент в домене <domain-name>

#ввести в домен под учетными записями выданными РЖД

astra-freeipa-client -u <domain\_admin\_name> -p <domain\_admin\_pass> --par "--domain <domain\_server\_name> --server <domain\_server\_name\_1> --server <domain\_server\_name\_2>" -y

- Убедиться в работе всех устройств из списка "Необходимое оборудование"

Rutoken - https://wiki.astralinux.ru/pages/viewpage.action?pageId=32834416

https://forum.rutoken.ru/topic/1644/

### Настройка Zabbix

### Настройка NetBackup

### Настройка Kaspersky

### Прохождение ПСИ для платформы AccentOS

### Прохождение ПСИ для АРМ Windows

### Прохождение ПСИ для АРМ Linux

**Приложение 1 Описание переменных окружения для установки  
 через AccentOS Firstboot**

## Работа с репозиториями

### Импорт GPG-ключей для авторизации репозитория пакетов ПО

APTKEY0

…

APTKEY999

Скрипт в цикле перебирает данные переменные и пытается импортировать GPG-ключи, которыми могут быть подписаны пакеты в репозиториях.

Пример:

export APTKEY0="http://$REPO\_IP/train-accentos-sign/dists/train-accentos/pubkey.gpg"

### Формирование файла /etc/apt/sources.list с определениями расположения репозиториев

REPO0

…

REPO999

Каждая переменная должна хранить строку для файла /etc/apt/sources.list .

Пример

export REPO0="deb http://$REPO\_IP/smolensk smolensk contrib main non-free"

export REPO1="deb http://$REPO\_IP/devel-smolensk smolensk contrib main non-free"

export REPO6="deb http://$REPO\_IP/train-accentos-sign train-accentos contrib main non-free"

export REPO9="deb http://$REPO\_IP/update202111-smolensk smolensk contrib main non-free"

export REPO10="deb http://$REPO\_IP/devel-update202111-smolensk smolensk contrib main non-free"

## Отключение интерактивная установки пакетов debian

export DEBIAN\_FRONTEND="noninteractive"

## Установка часового пояса

Переменная TIMEZONE\_FILE предназначена для указания актуального часового пояса и заполняется в формате расположения файла соответствующего часовому поясу территориального расположения серверов.

Пример:

export TIMEZONE\_FILE=Europe/Moscow

## Определение настроек сети развертывания

Префикс сети размещения узлов облака указывается в переменной NET.

export NET=10.40.21

Примечание: переменная используется для определения ip-адресов служб в пределах используемой сети как в других переменных, так и в скриптах.

### Сетевая конфигурация вычислительных узлов

Определение подчиненных сетевых интерфейсов, объединяемых в единый интерфейс bond

export BOND\_SLAVES=""

Тип объединения сетевых интерфейсов: lacp или active-backup

export BOND\_MODE="lacp"

Установка длины сетевых пакетов (MTU). Данный параметр напрямую зависит от настройки сетевого оборудования. Рекомендуется снижать до стандартного значения 1500 при вероятном обращении вычислительных узлов к сторонним службам сети, на которых используются стандартные значения.

#export MTU=9000

export MTU=1500

### Настройки DNS

Временная зона DNS - зона расположенная на сервере Cobbler. Используется только на этапе установки операционных систем физических и виртуальных серверов.

export DNS\_ZONE=test.loc

Прямая зона DNS облака формируется на базе переменной CLOUD\_ZONE

export CLOUD\_ZONE=accentos.loc

На базе переменной DNS\_REVERSE\_ZONE формируется обратная зона облака.

export DNS\_REVERSE\_ZONE=21.40.10.in-addr.arpa

Определение расположения DNS-серверов

export CLOUD\_DNS1=$NET.221

export CLOUD\_DNS2=$NET.222

export CLOUD\_DNS3=$NET.223

В случае необходимости добавления дополнительных внешних зон можно воспользоваться скриптом install\_bind9-forward-zone. Скрипт должен запускаться после скрипта установки bind9 и добавляет дополнительные зоны типа forward. Внешние зоны добавляются по одной в каждой переменной в формате перечисления в строке имени зоны и до 4х ip-адресов DNS-серверов. Номера переменных для указания внешних зон могут быть любыми в пределах от 0 до 999.

export FORWARD\_DNS\_ZONE0="tnxos.loc 10.40.116.1 10.40.116.2"

export FORWARD\_DNS\_ZONE2="stand.loc 10.40.116.1 10.40.116.2"

### Настройки NTP

# сервера времени облака для взаимной синхронизации

export CLOUD\_NTP1=$NET.221

export CLOUD\_NTP2=$NET.222

export CLOUD\_NTP3=$NET.223

export EXTERNAL\_NTP=ru.pool.ntp.org

При назначении переменной EXTERNAL\_NTP непустого значения приводит к назначению данного значения как источника для серверов указанных в переменных CLOUD\_NTP1, CLOUD\_NTP2 и CLOUD\_NTP3, которые в свою очередь являются источниками для остальных узлов облака.

На момент написания данного документа алгоритм порядка синхронизации серверов следующий:

1. при задании значения переменной EXTERNAL\_NTP - он прописывается как источник на серверах, прописанных в переменных CLOUD\_NTP1, CLOUD\_NTP2, CLOUD\_NTP3.

* если в качестве EXTERNAL\_NTP указан один серверов облака, ему назначается значение stratum равное 8 и время этого сервера используется как эталон. Данный вариант можно использовать при работе облака в изолированной среде.
* Для серверов не указанных в переменных EXTERNAL\_NTP, CLOUD\_NTP1, CLOUD\_NTP2, CLOUD\_NTP3 в качестве источников используются непустые значения CLOUD\_NTP1, CLOUD\_NTP2, CLOUD\_NTP3. Если все 3 переменных имеют пустое значение, то в качестве источника назначается занчение переменной EXTERNAL\_NTP.

Таким образом рекомендуется заполнить переменную EXTERNAL\_NTP либо адресов внешнего источника, либо адресом одного из серверов. Это может быть один серверов указанных в переменных CLOUD\_NTP1, CLOUD\_NTP2, CLOUD\_NTP3. После чего задать значения переменным CLOUD\_NTP1, CLOUD\_NTP2, CLOUD\_NTP3 (необязательно все).

## Создание системы хранения на базе Ceph

Настройка Ceph заключается в объявлении адресов серверов мониторов Ceph, а также в объявлении размещения дисков под Ceph OSD на серверах.

Адреса мониторов Ceph объявляются в виде ip-адресов через пробел. Число мониторов рекомендуется нечетным от 3 и выше.

export CEPH\_IPS="$NET.221 $NET.222 $NET.223"

Формат объявления Ceph OSD заключается в назначении переменным CEPH\_OSD0..CEPH\_OSD999 значений в виде перечисления через пробел ip-адреса сервера и блочное устройство, которым может быть как физическое блочное устройство целиком, так и раздел на оном.

export CEPH\_OSD0="$NET.221 /dev/nvme0n1p4"

export CEPH\_OSD1="$NET.222 /dev/nvme0n1p4"

export CEPH\_OSD2="$NET.223 /dev/sda4"

На всех серверах в настройках firstboot должен быть объявлен к выполнению скрипт install\_ceph-cluster.

## Глобальные настройки облака

# глобальный пароль

export PASSWORD=qwertyuiop

# имя и адрес сервера cobbler

export COBBLER\_HOST=cobbler.test.loc

export COBBLER\_IP=10.40.21.4

### Подключение ISCSI дисков

Скрипт map\_iscsitarget позволяет подключить сетевые диски к серверам. Данные настройки могут являться как общими для всех серверов в случае использования кластерной файловой системы или использования LVM с раздельным или общим подключением логических дисков к системе.

Переменные TARGET\_IP0..TARGET\_IP999 должны иметь адреса и порты iscsi-серверов, на которых будут затребованы доступные таргеты для подключения.

export TARGET\_IP1=10.40.21.18:3260

Переменные TARGET0..TRAGET999 должны иметь имена таргетов, которые должны быть представлены в сервере.

export TARGET1=iqn.2007-02.com.example:nova

export TARGET2=iqn.2007-02.com.example:glance

export TARGET3=iqn.2007-02.com.example:gnocchi

Так как имена дисков при каждом включении сервера могут быть меняться в зависимости от очередности текущего подключения, рекомендуется для решения данного вопроса запускать multipath.

### Настройка кластера OCFS2

В устанавливаемом облаке может быть заявлено несколько непересекающихся кластеров OCFS2. Но каждая операционная система может быть в составе только одного кластера OCFS2. В случае пересечения 2х кластеров OCFS2 хотя бы на одном сервере, они должны быть объединены в один. Для каждого кластера переменные OCFS2\_IPS и OCFS2\_CLUSTER\_NAME должны быть объявлены для всех узлов кластера. OCFS2\_CLUSTER\_NAME должно иметь уникальное значение для каждого кластера. Переменная OCFS2\_DRIVE при этом должна содержать список блочных устройств, которые будут использованы в кластере.

export OCFS2\_IPS="$NET.221 $NET.222 $NET.223"

export OCFS2\_CLUSTER\_NAME=ocfs2  
export OCFS2\_DRIVE=/dev/mapper/360000000000000000e00000000030001

Кластер OCFS2 может быть использован для размещение дисков nova, для хранения образов glance и так далее…

В общем случае создается один кластер OCFS2 для контроллеров и один или несколько кластеров OCFS2 для вычислительных узлов. Из-за высокой вероятности, что диски с файловой системой OCFS2 могут быть доступны и на других узлах файловые системы должны форматироваться с параметром ***--global-heartbeat***, а каждой системе присваивается номер равный последнему октету ip-адреса для уникальности в рамках локальной сети.

Объявление нескольких кластеров OCFS2 осуществляется либо созданием файлов со всем списком переменных для отдельных групп серверов, либо созданием общего файла с одинаковыми значениями переменных и различных файлов для отдельных групп серверов для различающихся значений переменных. Это касается не только кластеров OCFS2, а также

### Использование VDO

Использование VDO в операционной системе Linux Astra возможно с ядром 5.10. Для этого используется скрипт install\_vdo. Перед его запуском необходимо загрузить конфигурацию в виде списка переменных окружения.

Для Debian пакеты для поддержки VDO отсутсвуют.

Примеры настройки диска VDO

Пример 1.

export VDO0\_DRIVE=/dev/vdс

export VDO0\_NAME=vdo-lvm

export VDO0\_N=8

export VDO0\_VG=vdo\_name

export VDO0\_LV0\_SIZE=100G

export VDO0\_LV1\_SIZE=100G

export VDO0\_LV0\_NAME=nova

export VDO0\_LV1\_NAME=zun

export VDO0\_LV1\_FS=xfs

export VDO0\_LV0\_MOUNT=/var/lib/nova/instances

export VDO0\_LV1\_MOUNT=/var/lib/docker

В данном примере объявляется создание VDO диска vdo-lvm на базе блочного устройства объявленного в переменной VDOn\_DRIVE. Переменные могут иметь вид VDO0\_DRIVE..VDO999\_DRIVE. Если блочное устройство найдется будет инициировано создание VDO диска с именем, указанным в переменной VDOn\_NAME. Объем VDO диска будет заявлен размером более в VDOn\_N раз, чем базовое блочное устройство. Коэффициент стоит выбирать исходя из планируемого характера данных - чем выше вероятность повторения хранимых данных, тем выше стоит сделать коэффициент.  
  
При наличии непустой переменной VDOn\_VG будет создан LVM VG с именем указанной в данной переменной. После создания МП возможно создание LV и монтирование созданных логических дисков в систему. При монтировании вновь созданных логических дисков создается файловая система ext4, при желании можно использовать файловую систему xfs указав в переменной VDOn\_LVx\_FS=xfs. Имя LV и размер указываются соответственно в переменных VDOn\_LVx\_NAME и VDOn\_LVx\_SIZE.

Пример 2

Если в переменных не указываются данные по логическим дискам, то настройка будет ограничена созданием VG, например это удобно при настройке cinder-volume.

export VDO1\_DRIVE=/dev/vdb

export VDO1\_NAME=vdo-cinder

export VDO1\_N=20

export VDO1\_VG=cinder\_volumes

Пример 3

Если создание использование LVM на VDO диске не актуально, то можно ограничиться созданием файловой системы на весь VDO диск без разбиения виртуального блочного устройства.

export VDO2\_DRIVE=/dev/vdd

export VDO2\_NAME=vdo-opt

export VDO2\_N=5

export VDO2\_MOUNT=/opt

export VDO2\_FS=xfs

Таким образом исходя из имеющегося оборудования можно выбрать наиболее удобный вариант подключения локального диска с возможностью дедупликации и сжатия на локальном или сетевом физическом носителе данных.

Рекомендации по выбору файловой системы

Для некоторых проектов Openstack ВАЖНО получить файловую систему с определенным функционалом. К примеру для Openstack Zun, использующий квоты, лучшим выбором будет файловая система xfs и так далее…

## Распределение служб облака по физическим и виртуальным серверам.

Соответствие ip-адреса каждому физическому и виртуальному серверу определяется на уровне службы Cobbler..

### Конфигурация высокой доступности.

Конфигурация высокой доступности предполагает сохранение работоспособности при выходе из строя любого контроллера или узла базы данных. Для этого формируется минимум 3 контроллера и минимум 3 узла базы данных. Количество контроллеров или узлов БД может быть больше, но это число рекомендуется поддерживать нечетным, чтобы при разрыве и дальнейшем восстановлении связности сети большее число узлов формировало кворум для разрешения конфликтных ситуаций. Так же далее следуют рекомендации по комбинации служб на каждом хосту, но данные рекомендации не являются обязательными к исполнению и отдельные службы (Redis, RabbitMQ, etcd и т.д.) могут быть установлены на отдельные виртуальные или физические хосты.

### Адреса контроллеров Openstack

*export CONTROLLER\_IPS="$NET.234 $NET.235 $NET.236"*

Каждый контроллер Openstack по факту является самостоятельным и взаимодействует с остальными через сторонние службы (Redis, MariaDB, RabbitMQ и т.д.). При нормальном функционировании нагрузка на службы Openstack распределяется балансировщиками haproxy основываясь на задержке ответа. При недоступности одного контроллера другие контроллеры полностью заберут на себя всю нагрузку, ранее выполняемую неработоспособным контроллером.

Имя контроллера облака - данное DNS имя является служебным и используется для доступа к контроллерам через балансировщик запросов. На тех узлах, которые используют локальный балансировщик, данное имя определяется на один из адресов localhost

export CONTROLLER\_NAME=controller.$CLOUD\_ZONE

### |Служба Memcache

На контроллерах обязательно должна устанавливаться и запускаться служба memcached. Для установки данной службы используется отдельный скрипт install\_memcache, но отдельного имени для данной службы не назначается. В при настройке балансировщиков haproxy для данной службы используются адреса, указанные в переменной CONTROLLER\_IPS, а при настройке конфигурационных файлов служб Openstack указывается имя из переменной CONTROLLER\_NAME.

### Адреса балансировщика внешних запросов haproxy.

*export HAPROXY\_IPS="$NET.231 $NET.232 $NET.233"*

Данный балансировщик не используется для запросов с серверов облака. На каждый вычислительный узел и контроллер ставится свой балансировщик. Данный балансировщик не занимается балансировкой нагрузки для MariaDB, RabbitMQ, etcd и Redis

### Адреса кластера базы данных MariaDB

Рекомендуется устанавливать совместно с балансировщиком haproxy для глобальных запросов

*export MYSQL\_IPS="$NET.231 $NET.232 $NET.233"*

Кластер не балансируется haproxy для глобальных запросов и имеет свое служебное имя mysql в DNS-зоне, объявленной в переменной. На каждом ВУ и контроллере работоспособность кластера отслеживается с помощью локально установленного haproxy с целью максимально быстрого переключения с неработающего узла на работающий.

### Адреса кластера службы очередей сообщений RabbitMQ

рекомендуется устанавливать совместно с балансировщиком haproxy для глобальных запросов

export RABBITMQ\_IPS="$NET.232 $NET.231 $NET.233"

Кластер не балансируется haproxy и имеет свое служебное имя rabbitmq в DNS-зоне, объявленной в переменной На каждом ВУ и контроллере работоспособность кластера отслеживается с помощью локально установленного haproxy с целью максимально быстрого переключения с неработающего узла на работающий.

### Адреса кластера etcd

Etcd используется для doker под управлением Openstack Zun и рекомендуется к установке совместно с балансировщиком haproxy для глобальных запросов.

export ETCD\_IPS="$NET.231 $NET.232 $NET.233"

Кластер не балансируется haproxy и имеет свое служебное имя etcd в DNS-зоне, объявленной в переменной. На каждом ВУ и контроллере работоспособность кластера отслеживается с помощью локально установленного haproxy с целью максимально быстрого переключения с неработающего узла на работающий.

### Адреса БД Redis

База данных Redis c размещением данных в памяти используется AccentOS и рекомендуется к установке на контроллерах.

*export REDIS\_IPS="$NET.234 $NET.235 $NET.236"*

### Адреса и настройки Consul

Инфраструктура Consul состоит из узлов, выполняющих серверную часть в кластере, и узлов, выполняющих функцию агента. Поэтому скрипт установки должен запускать на каждом узле, но активация серверной функции происходит только на узлах определенных в нижеуказанных настройках.

Список узлов-серверов - обязательно в общих кавычках и каждый сервер в экранированных кавычках

*export CONSUL\_SERVERS="\"$NET.221\",\"$NET.222\",\"$NET.223\""*

Домен для узлов consul

export CONSUL\_DOMAIN=consul

Название датацентра, сформированного на базе серверов Consul.

export CONSUL\_DATACENTER="DC1"

Пароль авторизации для синхронизации данных между узлами consul. Генерация пароля реализуется командой "consul keygen"

export CONSUL\_SECRET="bnRHLmJ6TeLomirgEOWP2g=="

## Настройки отдельных модулей Openstack

### Keystone

Время действия токена KEYSTONE, которое по-умолчанию 3600 (1 час)

*export KEYSTONE\_TOKEN\_EXPIRATION=3600*

### Glance

Использование внешнего диска (iscsi или fibre channel) или отдельного блочного устройства.

Такой подход позволяет подключить более емкий сетевой диск для хранения образов или диск с поддержкой сжатия и\или дедупликации (vdo, zfs или иное аппаратное обеспечение). Учитывая, что могут быть созданы образы различного целевого назначения на базе одной операционной системы, подобный подход может значительно удешевить стоимость хранения информации.

Имя устройства на стороне ***физического узла, используемого как гипервизор для виртуальной машины*** с установленным Openstack Glance. Данное имя используется только для подключения блочного устройства к виртуальной машине с контроллером.

Пример устройства с multipath

export GLANCE\_DRIVE=/dev/mapper/360000000000000000e00000000010001

Пример блочного устройства LVM

export GLANCE\_DRIVE=/dev/mapper/lvm-glance

Пример отдельного физического диска.

export GLANCE\_DRIVE=/dev/sdb

При подключении блочного устройства напрямую к виртуальной машине минуя физический сервер, используемый в качестве гипервизора, например по ISCSI или при работе службы Glance непосредственно на физическом сервере - переменная должна быть пустой.

Использование сетевых ресурсов совместно с кластерной файловой системой позволяет одновременно использовать одно хранилище на нескольких контроллерах с целью повышения надежности и производительности службы. В AccentOS Firstboot пока создана поддержка файловой системы ext4 при работе только одного экземпляра службы Glance в составе облака и кластерной файловой системы ocfs2 при одновременном использовании нескольких экземпляров службы Openstack Glance

export GLANCE\_DRIVE\_FS=ocfs2

Имя блочного устройства, монтируемого в систему внутри физического или виртуального сервера  
Имя указывается без указания префикса /dev (прим. исправлено в последней редакции, сейчас допустимо написание как с префиксом /dev/таи без оного). Данное имя используется при монтировании блочного устройства в корневую файловую систему. Определение фактического имени внутри операционной системы сервера лежит на администраторе, так оно оно может зависеть от многих факторов и не соответствовать указанному явно в libvirt xml для виртуальной машины. Данный диск монтируется в папку /var/lib/glance/image

Пример использования на виртуальной машине

*export GLANCE\_DRIVE\_MAPPED=vdb*

Пример использования на физической машине с дополнительным жестким диском или сетевым диском без использования multipath

*export GLANCE\_DRIVE\_MAPPED=sdb*

Пример использования диска multipath на физическом или виртуальном сервере. В случае виртуальной машины блочное устройство подключается напрямую минуя гипервизор, например по ISCSI.

*export GLANCE\_DRIVE\_MAPPED=/mapper/360000000000000000e00000000010001*

Принудительное форматирование выделенного блочного устройства. Форматирование осуществляется при явном указании в переменной ENABLE\_CLEAR\_GLANCE\_DRIVE и одновременном создании базы данных для службы Openstack Glance. Т.е. при пересоздании отдельного контроллера в конфигурации HA очистка диска проводиться не должна при доступной БД. Также при восстановлении конфигурации для сохранения содержимого образов Glance очистку можно явно запретить если планируется дальнейшее восстановление содержимого БД.

export ENABLE\_CLEAR\_GLANCE\_DRIVE=yes

***Внимание! Код форматирования в скриптах в настоящий момент закомментирован.***

### Gnocchi

Использование внешнего диска (iscsi или fibre channel) или отдельного блочного устройства

Такой подход позволяет подключить более емкий сетевой диск для хранения образов или диск с поддержкой сжатия и\или дедупликации (vdo, zfs или иное аппаратное обеспечение). Учитывая, что могут быть созданы образы различного целевого назначения на базе одной операционной системы, подобный подход может значительно удешевить стоимость хранения информации.

Имя устройства на стороне ***физического узла, используемого как гипервизор для виртуальной машины*** с установленным Openstack Gnocchi. Как правило данное имя используется для подключения блочного устройства к виртуальной машине с контроллером.

Пример определения блочного устройства multipath

export GNOCCHI\_DRIVE=/dev/mapper/360000000000000000e00000000020001

Использование сетевых ресурсов совместно с кластерной файловой системой позволяет одновременно использовать одно хранилище на нескольких контроллерах с целью повышения надежности и производительности службы. В AccentOS Firstboot пока создана поддержка файловой системы ext4 при работе только одного экземпляра службы Gnocchi в составе облака и кластерной файловой системы ocfs2 при одновременном использовании нескольких экземпляров службы Openstack Gnocchi

Пример определения файловой системы ocfs2

export GNOCCHI\_DRIVE\_FS=ocfs2

Имя блочного устройства, монтируемого в систему внутри физического или виртуального сервера  
Имя указывается без указания префикса /dev (прим. исправлено в последней редакции, сейчас допустимо написание как с префиксом /dev/таи без оного). Данное имя используется при монтировании блочного устройства в корневую файловую систему. Определение фактического имени внутри операционной системы сервера лежит на администраторе, так оно оно может зависеть от многих факторов и не соответствовать указанному явно в libvirt xml для виртуальной машины. Данный диск монтируется в папку /var/lib/glance/image

export GNOCCHI\_DRIVE\_MAPPED=vdc

Принудительное форматирование выделенного блочного устройства. Форматирование осуществляется при явном указании в переменной ENABLE\_CLEAR\_GNOCCHI\_DRIVE и одновременном создании базы данных для службы Openstack Glance. Т.е. при пересоздании отдельного контроллера в конфигурации HA очистка диска проводиться не должна при доступной БД. Также при восстановлении конфигурации для сохранения содержимого образов Glance очистку можно явно запретить если планируется дальнейшее восстановление содержимого БД.

export ENABLE\_CLEAR\_GNOCCHI\_DRIVE=yes

***Внимание! Код форматирования в скриптах в настоящий момент закомментирован.***

### Nova

Имя устройства на стороне ***физического узла, используемого как гипервизор для запуска виртуальных машин*** с помощью установленного Openstack Nova. Как правило данное имя используется для подключения блочного устройства к виртуальной машине с контроллером. Данный диск монтируется в папку /var/lib/nova/instances. Пустое значение означает, что будет использоваться диск на котором расположена папка /var/lib/nova - размещение данной или вышестоящей папки может быть определено при создании операционной системы.

export NOVA\_DRIVE=/dev/mapper/360000000000000000e00000000030001

Определение типа файловой системы.  
В настоящий момент поддерживаются файловая система ext4 для индивидуального подключения блочного устройства к вычислительному узлу или кластерная файловая система ocfs2 для подключения общего хранилища к двум и более вычислительным узлам.

Пример подключения файловой системы ocfs2

export NOVA\_DRIVE\_FS=ocfs2

### Neutron

Установка количества сетевых агентов по умолчанию

export NEUTRON\_DHCP\_AGENTS\_COUNT=2

Данный параметр указывает количество DHCP-агентов и, соответственно, агентов метаданных

Настройка DVR (распределенного роутера)

При выключенном режиме DVR весь исходящий трафик виртуальной сети во внешнюю сеть, идущий через виртуальные маршрутизаторы Openstack с целью подмены адреса (NAT), проходит через определенный сервер или сервера, на которых располагаются L3-агенты. Это могут быть как контроллеры, так и ВУ, так как L3-агенты запускаются и там, и там. Количество используемых серверов зависит от типа маршрутизатора - для работы обычного маршрутизатора используется только 1 агент, для работы маршрутизатора высокой доступности - 3. Выбор конкретного агента для каждого виртуального маршрутизатора производится автоматически при создании виртуального маршрутизатора. Недостатком является высокое потребление ресурсов при высоком объеме сетевого трафика и рост сетевого отклика при неспособности оперативно транслировать весь проходящий трафик через L3-агента.

export DVR=no

При включении режима распределенного маршрутизатора исходящий трафик может подменить ВУ с запущенным L3-агентом и на котором запущена виртуальная машина, но обратный трафик транслируется как правило всеми ВУ, до которого дойдет ответный на запрос пакет. Данный режим несовместим с частью служб Openstack, например, с сетевым балансировщиком Octavia.

export DVR=yes

### Zun

В настоящий момент реализовано использование на блочном устройстве ceph

Название пула Ceph в котором будет создано целевое блочное устройство RBD, которое затем будет примонтировано к операционной системе

export DOCKER\_POOL\_NAME=volumes

Создаваемый размер блочного устройства для каждого ВУ

export DOCKER\_POOL\_SIZE=30720

Алгоритм создания и подключения блочного устройства на сегодня реализован такой: на каждом вычислительном узле проверяется наличие конфигурации ceph на вычислительном узле и при положительном результате поиска создается блочное устройство с требуемым размером, которое настраивается для автоматического подключения и монтирования к операционной системе при запуске.

Используемый формат хранилища для Linux Astra Smolensk 1.6 - lvm, для Astra Linux Smolensk 1.7 - overlay2, который далее будет использоваться для всех версий дистрибутивов.

Если конфигурация Ceph не будет найдена - дополнительного блочного устройства подключено не будет. Как результат не будет доступно использование квот и гарантии работоспособности данной службы не будет.

### Octavia

Данный проект в рамках Openstack предназначен для создания службы сетевого балансирования нагрузки для других служб, например требуется для работы Openstack Magnum

Сеть, используемая сетевым балансировщиком.

export OCTAVIA\_MGMT\_SUBNET=172.16.0.0/12

Диапазон выдаваемых адресов

export OCTAVIA\_MGMT\_SUBNET\_START=172.16.0.100

export OCTAVIA\_MGMT\_SUBNET\_END=172.16.31.254

Список IP адресов, назначаемых агентам на контроллерах (по одному на каждый контроллер)

export OCTAVIA\_MGMT\_PORT\_IP="172.16.0.2 172.16.0.3 172.16.0.4"

## Настройка модулей AccentOS

Имя пользователя AOS для подключения к БД и другим службам

export AOS\_USER=aos

Пароль пользователя модулей AOS

Пример явного назначения пароля

export AOS\_PASSWORD=password

Пример использования пароля по-умолчанию

export AOS\_PASSWORD=$PASSWORD

export AOS\_RABBIT\_VHOST=aos

Для модулей AccentOS требуется предварительная установка Redis.

## Устаревшие параметры

Определение менеджера запуска служб Openstack и взаимосвязанных с ним служб. По умолчанию используется systemd. Возможной альтернативой является firstboot. Менеджер должен определять порядок запуска служб Openstack, настройку и запуск служб обслуживающих Openstack, если порядок этих служб важен для успешной работы.

#BOOT\_MANAGER=firstboot

export BOOT\_MANAGER=systemd

Изначально Firstboot был нацелен на использование Ceph в качестве сетевого хранилища.

export SHARED\_DRIVE="ceph"

#export SHARED\_DRIVE="fc"

В дальнейшем будет осуществляться отказ от данной переменной в целью настройки каждой службы Openstack на использование отдельного сетевого хранилища данных. Например, Cinder может использовать несколько типов хранилищ, Nova - свой тип хранилища и так далее…

**Приложение 2 Описание системы автоустановки FirstBoot**

***Описание службы firstboot***

*Назначение*

Служба предназначена для установки и настройки ПО при первом запуске физического или виртуального сервера. Может быть также использована для обеспечения корректного запуска служб при последующих запусках.

*Способ установки*

Первоначально установка службы планировалась при развертывании серверов по сети при использовании Cobbler и файла ответов.

Пример части файла ответов для дистрибутивов debian и ubuntu

d-i preseed/late\_command string wget -O- \

http://$http\_server/init/update\_kernel | chroot /target /bin/sh -s; wget -O- \

http://$http\_server/cblr/svc/op/script/$what/$name/?script=preseed\_late\_default | \

chroot /target /bin/sh -s; wget -O- \  
 http://$http\_server/cblr/svc/op/nopxe/$what/$name; wget -O- \

http://$http\_server/server | /bin/sh -s;

Команда, сохраненная в d-i preseed/late\_command, запускается в самом конце запуска инсталлятора и предназначена для установки пользовательского ПО.

В данном примере производится обновление ядра, затем производится обращение к Cobbler с целью запуска скрипта preseed\_late\_default и снятия флага сетевой загрузки сервера, затем на стороне устанавливаемого сервера запускается скрипт [http://$http\_server/server](about:blank) (в данном случае), который устанавливает службу firstboot и скачивает необходимые скрипты-модули для установки ПО сервера при первом запуске. В настоящий момент “server” - статический файл-скрипт, размещенный на стороне Cobbler. Подобное использование неудобно, т.к. требует создание файлов-ответов для каждой конфигурации устанавливаемого сервера. В дальнейшем возможно будет использоваться скрипт, который будет генерировать содержимое данного скрипта для каждого вновь устанавливаемого сервера в зависимости от его планируемой конфигурации.

Пример такого скрипта:

*#!/bin/bash*

*http\_server=cobbler.test.loc*

*wget -O- http://$http\_server/init/firstboot | chroot /target /bin/sh -s*

*wget -O /target/var/lib/firstboot/scripts/15-install.sh http://$http\_server/init/rc-local*

*wget -O /target/var/lib/firstboot/scripts/05-initrc.source http://$http\_server/init/initrc.controllerone*

*wget -O /target/var/lib/firstboot/scripts/20-install\_bind9.sh http://$http\_server/init/install\_bind9*

*wget -O /target/var/lib/firstboot/scripts/20-install\_ntp.sh http://$http\_server/init/install\_ntp*

*wget -O /target/var/lib/firstboot/scripts/20-install\_update-dns.sh http://$http\_server/init/install\_update-dns*

*wget -O /target/var/lib/firstboot/scripts/25-install\_consul.sh http://$http\_server/init/install\_consul*

*wget -O /target/var/lib/firstboot/scripts/25-install\_openstackclient.sh http://$http\_server/init/install\_openstackclient*

*wget -O /target/var/lib/firstboot/scripts/45-install\_nova-compute.sh http://$http\_server/init/install\_nova-compute*

*wget -O /target/var/lib/firstboot/scripts/40-install\_neutron-compute.sh http://$http\_server/init/install\_neutron-compute*

*wget -O /target/var/lib/firstboot/scripts/80-install\_install\_vm\_controllerdb.sh http://$http\_server/init/install\_vm\_controllerdb*

*wget -O /target/var/lib/firstboot/scripts/50-install\_ceilometer-compute.sh http://$http\_server/init/install\_ceilometer-compute*

*wget -O /target/var/lib/firstboot/scripts/90-correct\_nova\_shell.sh http://$http\_server/init/correct\_nova\_shell*

Скрипт скачивает с сервера cobbler (или иного сервера) скрипт модули и сохраняет их с цифровым префиксом для обеспечения требуемого порядка запуска. Путь для сохранения имеет префикс /target так как данный скрипт выполняется в окружении инсталлятора дистрибутива и файловая система устанавливаемого сервера смонтирована в каталог /target. Пути хранения исходных скриптов могут быть в дальнейшем изменены для создания соответствующей иерархии каталогов для установки различных дистрибутивов и их версий.

Первый скачиваемый скрипт не сохраняется, а запускается сразу после загрузки. Именно данный скрипт создает службу firstboot и формирует необходимую иерархию каталогов для данной службы. Порядок загрузки других скриптов не важен, но следует обратить внимание на цифровой префикс имен сохраняемых скриптов с целью обеспечения необходимого приоритета их выполнения службой firstboot.

Необходимо отметить о настройке файлов с переменными initrc и openrc

Скрипты для firstboot необходимо писать таким образом, чтобы они не содержали в себе каких-либо специфичных данных: ip-адресов, имен хостов и т.д. Все настраиваемые данные должны заменяться на переменные и выноситься в файлы с определением переменных. Это даст более гибкий подход. В дальнейшем при настройке развертывания систем через программный продукт данные файлы могут генерироваться в зависимости от нужд скриптами. В настоящий момент изменение переменных возможно только через создание нескольких файлов с переменными и указанием его в скрипте, вызываемом в файле ответов. Этот вопрос рассматривался чуть выше. В настоящий момент используются файлы initrc и openrc, первый заменяется файлом initrc.controllerone в случае установки с одним контроллером, совмещенным с БД.

*Описание работы службы firstboot*

Служба firstboot - это bash скрипт, который при первом запуске сортирует скрипты в каталоге /var/lib/firstboot/scripts по имени и запускает их. Для простоты восприятия имена скриптов лучше начинать с чисел.

Скрипты делятся на 2 типа: определение переменных и установочные скрипты. Для определения переменных скрипты должны иметь суффикс .source - вызов данных скриптов производится через команду source. Остальные скрипты вызываются через вызов /bin/bash вне зависимости от расширения.

Вывод работы скриптов сохраняется в файл /var/log/firstboot

Окончание работы первого запуска службы завершается созданием файла /var/lib/firstboot/onstart. С помощью удаления данного файла можно протестировать работу службы повторно, например, с целью отладки скриптов.

Наличие файла /var/lib/firstboot/onstart сигнализирует службе firsboot, что осуществляется повторный, а не первичный запуск сервера, и служба запускает скрипты из каталога /var/lib/firstboot/boot. Данный режим необязателен, т.к. он некоторым образом дублирует работу systemd. Но данный режим является удобным, если процесс запуска службы является сложным: использующим таймауты или иные условия запуска того или иного сервиса, но при этом нежелательные к размещению в файлах systemd, например, для оперативного перазапуска в процессе работы сервера.

**Настройка системы для установки OpenStack Train**

Система должна иметь один носитель HDD или SSD, к сети подключен только один линк. Cobbler управляет службами DNS и DHCP.

Настройка системы включает в себя этапы

- Создание ‘Systems’

- заполнение файлов “/var/www/html/init/initrc”, “/var/www/html/init”

- добавление подсети

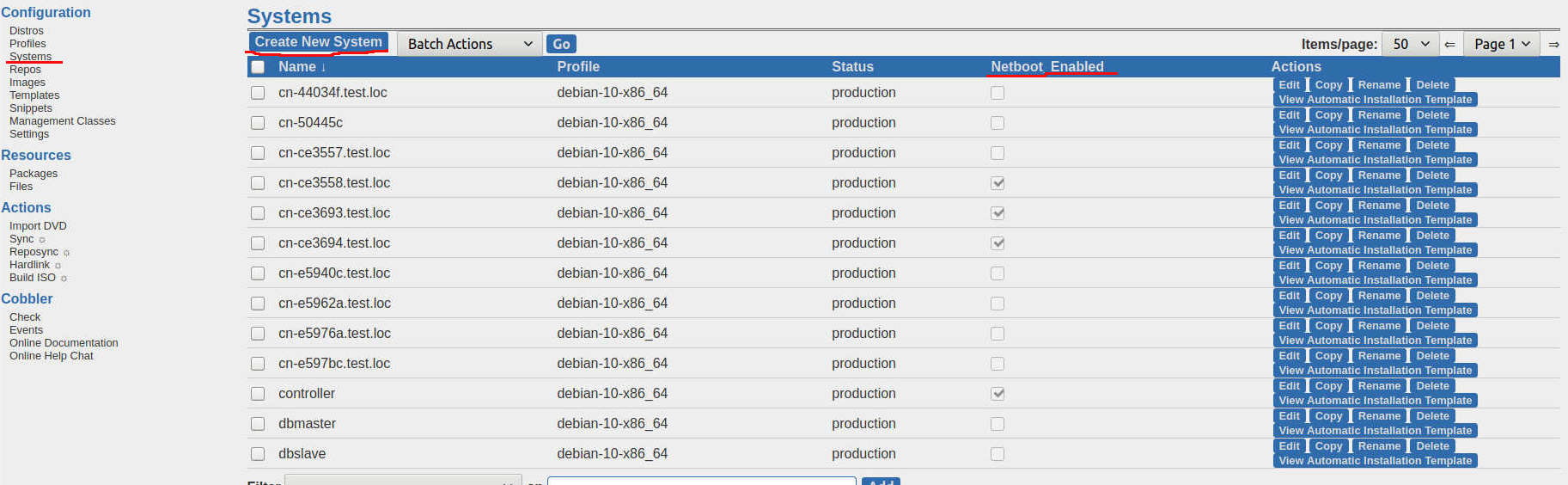
- запуск cobbler sync

**Создание ‘Systems’**

В рамках этого пункта необходимо описать все узлы и контроллер, которые будут устанавливаться и настраиваться.

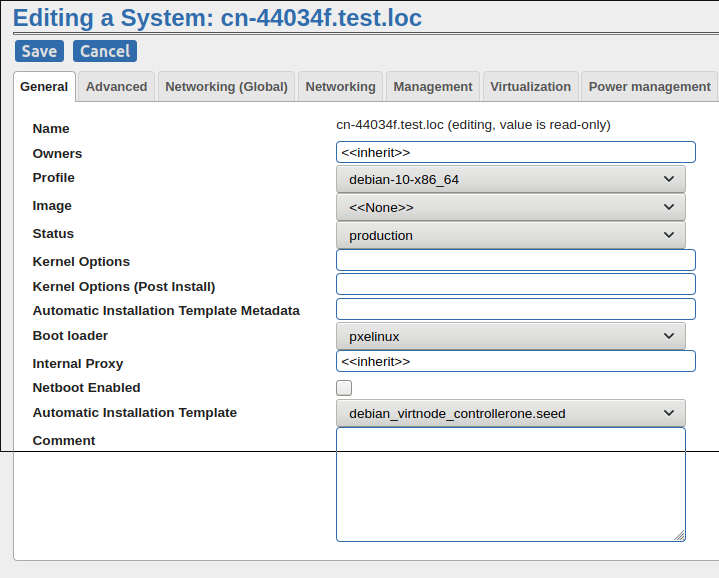
**1)** *Необходимо перейти в вебку системы – это serverip/cobbler\_web (login/password – cobbler/cobbler), где serverip- ip-адрес сервера cobbler. Доступ для ssh : login – debian, password – cobbler.*

***2)*** *Нажать на пункт* ***Systems*** *в разделе* ***Configuration***



**3)** *Нажать на кнопку “****Create New System****”*

**4)** *на вкладке* ***General*** *заполняются пункты*



а) Name

Для уникальности имени делаем подобным шаблону- “cn-шесть последних символов из MAC.test.loc” . Имя контроллера будет равно - controller

б) Profile

Выбираем пункт с Debian

в) Boot loader

Выбираем pxelinux

г)Netboot Enabled

этот параметр необходим для запуска установки на хосте. После установки на хосте ОС этот параметр снимается автоматически.

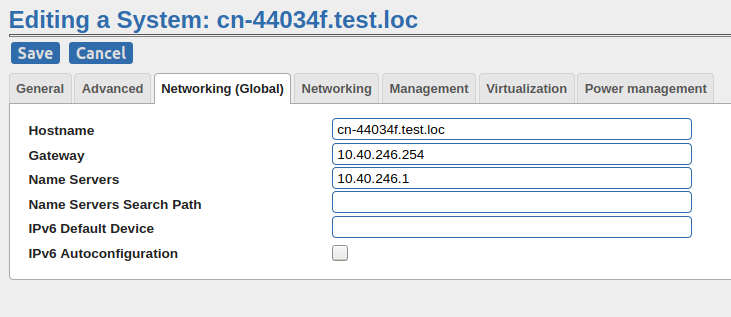
д) Automatic Installation Template

- *Для настройки ВУ, на которой будет запущен контроллер необходимо выбрать шаблон* debian\_virtnode\_controllerone.

- Для настройки ВУ необходимо выбрать шаблон debian\_node

- Для настройки контроллера необходимо выбрать шаблон debian\_controllerone

**5)**  *на вкладке* ***Networking (Global)****заполняем пункты:*



а) Hostname

по аналогии с Name

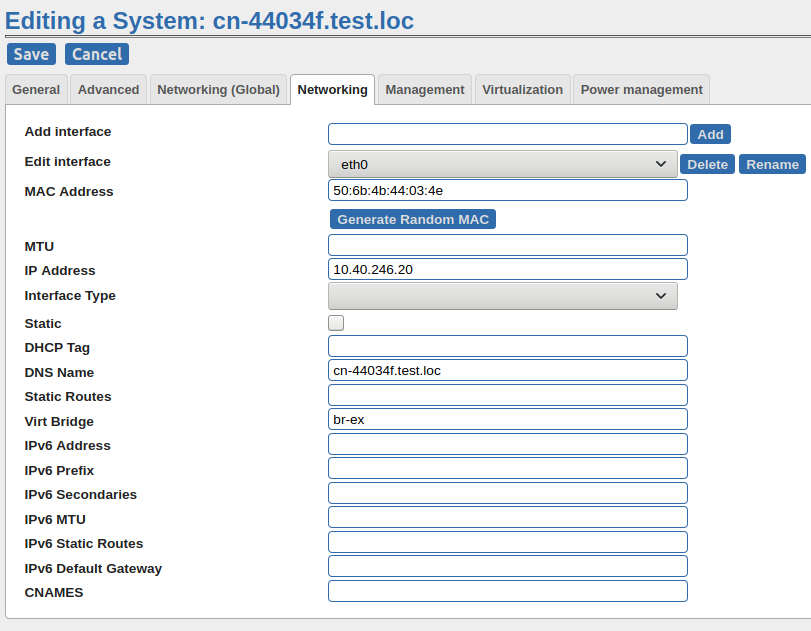
б) Gateway

здесь указываем шлюз сети, в которой будет находиться хост

в) Name Servers

указываем ip сервер Cobbler

***6)*** *На вкладке* ***Networking*** *заполняем пункты*



а) Add interface

Указываем интерфейс, через который будет проходить трафик. (в общем случае eth0)

б) MAC Address

указать MAC-адрес сетевого устройства, через которое будет производиться загрузка хоста.

в) IP Address

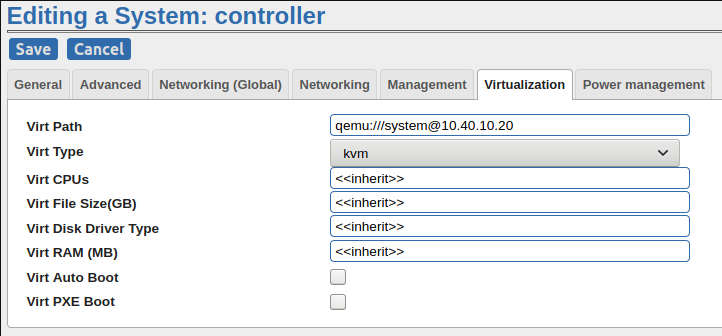
указать IP-адрес хоста

г) DNS Name

по аналогии с Name

д) Virt Bridge br-ex

равен br-ex

***7)*** *на вкладке* ***Virtualization*** *заполняются пункты: (заполняется только для контроллера)*

а) Virt Path

qemu:[///system@IP](mailto:///system@IP) , где IP равно ip-адресу ноды, где будет развёрнут контроллер

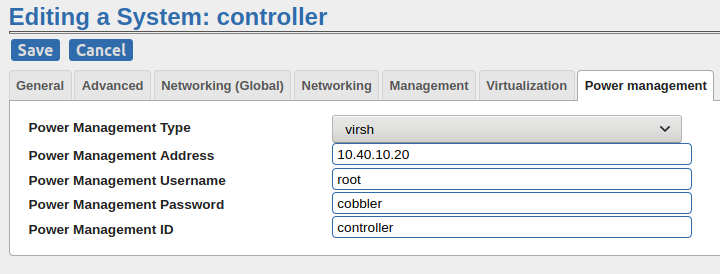
б) Virt Type

равно kvm

в) Virt PXE Boot

галочку установить

**8)** *для вкладки* ***Power manager*** *заполнить*



а) Power Management Type

Для контроллера равно virsh, для ВУ равно ipmilan

б) Power Management Address

Для контроллера – это ip-адрес ВУ, на котором будет запущен контроллер.

Для ВУ – это ip-адрес устройства управления питанием.

в) Power Management Username

для контроллера – логин для ВУ

для ВУ – логин устройства управления питанием (ipmi)

г) Power Management Password

для контроллера – пароль пользователя из предыдущего пункта

для ВУ- пароль для пользователя из предыдущего пункта

д)Power Management ID

для контроллера указать - controller

для ВУ – можно не указывать

**Добавление подсети**

Если сервер Cobbler будет настраивать стенд в новой сети, то необходимо добавить настройку DHCP для неё в Cobbler. Для этого необходимо отредактировать файл mcedit /etc/cobbler/dhcp.template. Нужно добавить новый диапазон для выдачи DHCP.

Пример

subnet 10.40.11.0 netmask 255.255.255.0 {

option routers 10.40.11.254;

option domain-name-servers 10.40.21.4;

option subnet-mask 255.255.255.0;

range dynamic-bootp 10.40.11.100 10.40.11.253;

default-lease-time 21600;

max-lease-time 43200;

next-server $next\_server;

if exists user-class and option user-class = "gPXE" {

filename "http://10.40.21.4/cobbler/pub/boot.ipxe";

} else if exists user-class and option user-class = "iPXE" {

filename "http://10.40.21.4/cobbler/pub/boot.ipxe";

} else {

filename "undionly.kpxe";

}

}

**Заполнение файлов “/var/www/html/init/initrc”, “/var/www/html/init”**

Эти файлы необходимо заполнить для передаче системе дополнительных параметров.

Пример

export NET=10.40.21

export DNS\_REVERSE\_ZONE=21.40.10.in-addr.arpa

#имя контроллера облака

export CONTROLLER\_NAME=controller.aos.loc

export CONTROLLER\_IP=$NET.5

# глобальный пароль

export PASSWORD=cobbler

# имя и адрес сервера cobbler

export COBBLER\_HOST=cobbler.test.loc

export COBBLER\_IP=10.40.21.4

export DNS\_ZONE=test.loc

export CLOUD\_ZONE=aos.loc

export CLOUD\_DNS1=$NET.20

export CLOUD\_DNS2=$NET.5

# неинтенрактивная установка пакетов debian

export DEBIAN\_FRONTEND="noninteractive"

# настройки consul

# список узлов-серверов - обязательно в общих ковычках и каждый сервер в экранированных ковычках

export CONSUL\_SERVERS="\"10.40.21.20\",\"10.40.21.5\""

export CONSUL\_DOMAIN=consul

export CONSUL\_DATACENTER="DC1"

# сгенерируйте свой пароль командой "consul keygen"

export CONSUL\_SECRET="bnRHLmJ6TeLomirgEOWP2g=="

export BACKUP\_NODE=$COBBLER\_IP

export BACKUP\_PATCH=/backup

export BACKUP\_LOCAL\_PATCH=/mnt/backup

export SMBUSER=cloud\_backup

export SMBPASS=backup

, где

NET – подсеть , в которой делается стенд

DNS\_REVERSE\_ZONE – обратная зона для сети, в которой делается стенд

CONTROLLER\_NAME - не меняем

CONTROLLER\_IP – ip-адрес контроллера

PASSWORD – пароль на OpenStack и службы окружения

COBBLER\_HOST – ip или доменное имя хоста cobbler (желательно не изменять)

COBBLER\_IP – ip-адрес сервера cobbler

DNS\_ZONE – не меняем

CLOUD\_ZONE – не меняем

CLOUD\_DNS1 - ip DNS-сервера для домена aos.loc

CLOUD\_DNS2 - ip DNS-сервера для домена aos.loc

DEBIAN\_FRONTEND – не меняем

CONSUL\_SERVERS – указываем два хоста, где будет развёрнут consul

Остальные параметры изменять не надо

**Запуск cobbler sync**

После внесения данных в разделе Systems необходимо запустить команду в консоли

**cobbler sync**

После выполнениея этой команды будут внесены изменения в DNS и DHCP.

Для того чтобы запустилась установка системы на хост, необходимо чтобы параметр хоста Netboot\_Enabled был установлен

**Установка Cobbler под Debian 10**

## Предварительная подготовка

Возможно 2 варианта сетевой реализации:

* Сервер Cobbller, сервера, развертываемые через Cobbler и шлюз находятся в одной локальной сети (в одном vlan)
* Сервер Cobbler является шлюзом для серверов, развертываемых с помощью Cobbler.

Оба варианта равнозначны, но 2й вариант позволяет иметь в сети для подчиненных серверов Cobbler независимые DNS и DHCP, не конфликтующие с остальной сетью. Данный вариант требует 2х сетевых интерфейсов либо физических, либо виртуальных для реализации VLAN.

Рассмотрим 2-й вариант.

Устанавливаем на сервер Cobbler операционную систему Debian 10 (Buster) любым удобным для вас способом. Настраиваем сеть в файле /etc/network/interfaces

*source /etc/network/interfaces.d/\**

*auto lo*

*iface lo inet loopback*

*allow-hotplug enp1s0*

*iface enp1s0 inet dhcp*

*auto enp2s0*

*iface enp2s0 inet static*

*address 192.168.1.1*

*netmask 255.255.255.0*

### Установка программного обеспечения

# **apt-get install -y make gcc python3-dev python3-pip python3-future python3-distro python3-sphinx python3-coverage \  
 python3-dnspython apache2 libapache2-mod-wsgi-py3 python3-netaddr python3-simplejson python3-yaml python3-cheetah \  
 tftpd-hpa ipxe rsync syslinux isolinux pxelinux syslinux-efi syslinux-utils python3-django isc-dhcp-server debmirror \  
 fence-agents xorriso bind9 bind9utils dnsutils ipmitool**

По-умолчанию TFTPD использует каталог /srv/tftp, а Cobbler - /var/lib/tftpboot  
С одной стороны можно перенастроить и первый и второй сервис на любой каталог, но во избежание возможных проблем при обновлениях - свяжем оба каталога симлинком.

# **mkdir -pv /srv/tftp**

# **ln -s /srv/tftp /var/lib/tftpboot**

### Настройка DHCP-сервера.

Установленный DHCP-сервер будет управлять непосредственно Cobbler’ом, но нам необходимо указать интерфейсы, которые он будет обслуживать в файле /etc/default/isc-dhcp-server

*...*

*INTERFACESv4="enp2s0"*

*…*

Замените enp2s0 на используемый сетевой интерфейс. Затем проверьте работоспособность службы.

# **systemctl enable isc-dhcp-server**

# **systemctl restart isc-dhcp-server**

## Установка и настройка Cobbler

### Установка Cobbler

На момент написания статьи актуальной версией является 3.1.1

# wget https://github.com/cobbler/cobbler/archive/v3.1.1.tar.gz

# tar -xf v3.1.1.tar.gz

# cd cobbler-3.1.1

# make install

Используемые каталоги

Библиотеки Python будут установлены в /usr/local/lib/python3.7/dist-packages/cobbler

Код веб-кабинета - /usr/local/share/cobbler

Исполняемые файлы будут размещены в /usr/local/bin

Конфигурационные файлы - /etc/cobbler и /etc/apache2

Для внутренных данных будет создан каталог /var/lib/cobbler

Все файлы, к которым будет необходим доступ по http-протоколу - /var/www/cobbler и /var/www/cobbler\_webui\_content

И для управления загрузкой серверов - /var/lib/tftpboot

### Настройка Cobbler

#### Настройка взаимодействия с Apache

Изначально Cobbler подразумевает доступ к панели управления по протоколу HTTPS. Для доступа по HTTP необходимо добавить в /etc/apache2/conf-available/cobbler\_web.conf разрешение:

# **echo "**

*<Location /cobbler\_web>*

*Require all granted*

*</Location>*

**" >> /etc/apache2/conf-available/cobbler\_web.conf**

Корректируем права доступа для каталога данных панели управления

# **chown www-data /var/lib/cobbler/webui\_sessions**

Включаем необходимые для работы модули и конфигурации Apache

# a2enmod ssl

# a2enmod proxy

# a2enmod rewrite

# a2enmod proxy\_http

# a2enconf cobbler

# a2enconf cobbler\_web

Перезапускаем Apache в новой конфигурации

# **systemctl restart apache2**

#### Настройка конфигурации Cobbler

Копируем файл службы Cobbler в служебный каталог systemd, корректируем его и разрешаем автоматический запуск

# cp -f /etc/cobbler/cobblerd.service /etc/systemd/system/

# sed -i 's/httpd.service/apache2.service/g' /etc/systemd/system/cobblerd.service

# systemctl enable cobblerd

Для кооректной работы Django корректируем некоторые параметры в коде Cobbler

# **SECRET\_KEY=$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 32 | head -n 1)**

# sed -i "s/SECRET\_KEY =.\*/SECRET\_KEY = '$SECRET\_KEY'/" /usr/local/share/cobbler/web/settings.py

# sed -i "s/TIME\_ZONE =.\*/SECRET\_KEY = 'Europe\/Moscow'/" /usr/local/share/cobbler/web/settings.py

# sed -i "s/\/usr\/share\/cobbler\/web\/templates/\/usr\/local\/share\/cobbler\/web\/templates/" /usr/local/share/cobbler/web/settings.py

В файле настроек Cobbler /etc/cobbler/settings необходимо скорректировать некоторые значения согласно настройкам самого сервера

IP-адрес сервера Cobbler

server: 192.168.1.1

IP-адрес сервера TFTPD

next\_server: 192.168.1.1

Разрешаем Cobbler управлять DHCP и DNS

manage\_dhcp: 1

manage\_dns: 1

Разрешаем управление загрузкой через TFTPD

pxe\_just\_once: 1

Создаем список прямых и обратных зон DNS

manage\_forward\_zones: [test.loc]

manage\_reverse\_zones: [192.168.1]

Проверяем корректность указания каталога для TFTP-загрузки серверов

tftpboot\_location: /var/lib/tftpboot

Создаем пароль для root вновь устанавливаемых серверов, для этого используем ранее сформированную случайную строку

# PASSWORD=’password\_for\_root’

# CRYPTED\_PASSWORD=$(openssl passwd -1 -salt $SECRET\_KEY $PASSWORD)

# sed -i "s/^default\_password\_crypted:.\*/default\_password\_crypted: $CRYPTED\_PASSWORD/" /etc/cobbler/settings

#### Настройка взаимодействия с DNS-сервером bind9

Корректируем шаблон /etc/cobbler/named.template для DNS-сервера, а именно прописываем IP-адрес(а) DNS сервера и сети, которые будет обслуживать данный сервер. Также корректируем каталоги согласно размещаемой операционной системе. **Все что ниже секции logging не меняем!**

*options {*

*listen-on port 53 { 127.0.0.1;* ***192.168.1.1;*** *};*

*directory "****/etc/bind****";*

*dump-file "****/etc/bind/****cache\_dump.db";*

*statistics-file "****/etc/bind/****named\_stats.txt";*

*memstatistics-file "****/etc/bind/****named\_mem\_stats.txt";*

*allow-query { localhost;* ***192.168.0.0/16; 172.16.0.0/12; 10.0.0.0/8;*** *};*

*recursion yes;*

*max-cache-size 256M;*

*max-acache-size 256M;*

*};*

*logging {*

*channel default\_debug {*

*file "****/var/cache/bind****/named.run";*

*severity dynamic;*

*};*

*};*

*#for $zone in $forward\_zones*

*zone "${zone}." {*

*type master;*

*file "$zone";*

*};*

*#end for*

*#for $zone, $arpa in $reverse\_zones*

*zone "${arpa}." {*

*type master;*

*file "$zone";*

*};*

*#end for*

Изменяем шаблон зоны в файле /etc/cobbler/zone.template

*\$TTL 300*

*@ IN SOA $cobbler\_server. nobody.example.com. (*

*$serial ; Serial*

*600 ; Refresh*

*1800 ; Retry*

*604800 ; Expire*

*300 ; TTL*

*)*

*IN A 192.168.1.1*

*IN NS $cobbler\_server.*

*cobbler IN A 192.168.1.1*

*$cname\_record*

*$host\_record*

Измените адрес 192.168.1.1 - на ip-адрес сервера Cobbler. Запись cobbler для Вашей зоны позволит потом обращаться к серверу для скачивания скриптов или иной информации, необходимой для установки и настройки серверов.

Корректируем права на каталог для /etc/bind. Данный пункт необходим в случае использования динамических зон.

# chown bind:bind /etc/bind

В случае использования динамических зон необходимо активировать модуль Cobbler для работы с nsupdate в файле /etc/cobbler/settings.d/nsupdate.settings

*nsupdate\_enabled: 1*

Создаем ключ для обновления DNS

# **dnssec-keygen -a HMAC-SHA512 -b 512 -n USER cobbler\_update\_key**

Прописываем данный ключ в /etc/cobbler/settings.d/nsupdate.settings

*nsupdate\_tsig\_algorithm: "hmac-sha512"*

*nsupdate\_tsig\_key: [ "cobbler\_update\_key.", "hvnK54HFJXFasHjzjEn09ASIkCOGYSnofRq4ejsiBHz3udVyGiuebFGAswSjKUxNuhmllPrkI0HRSSmM2qvZug==" ]*

А также добавить после секции logging в шаблоне /etc/cobbler/named.template

*key cobbler\_update\_key {*

*algorithm hmac-sha512;*

*secret "hvnK54HFJXFasHjzjEn09ASIkCOGYSnofRq4ejsiBHz3udVyGiuebFGAswSjKUxNuhmllPrkI0HRSSmM2qvZug==";*

*};*

#### Настройка взаимодействия с DHCP-сервером

Корректируем шаблон DHCP-сервера /etc/cobbler/dhcp.template

Для каждой используемой сети необходимо прописать секцию

*subnet* ***192.168.1.0*** *netmask 255.255.255.0 {*

*option routers* ***192.168.1.1****;*

*option domain-name-servers* ***192.168.1.1****;*

*option subnet-mask 255.255.255.0;*

*range dynamic-bootp* ***192.168.1.100 192.168.1.254****;*

*default-lease-time 21600;*

*max-lease-time 43200;*

*next-server $next\_server;*

*class "pxeclients" {*

*match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";*

*# Legacy*

*if option system-arch = 00:00 {*

*filename "****pxelinux.0****";*

*}*

*# UEFI-32-1*

*if option system-arch = 00:06 {*

*# Not supported, no 32 bit UEFI grub executable*

*filename "unsupported";*

*}*

*# UEFI-32-2*

*if option system-arch = 00:02 {*

*# Not supported, no 32 bit UEFI grub executable*

*filename "unsupported";*

*}*

*# UEFI-64-1*

*else if option system-arch = 00:07 {*

*filename "grub/grubx64.efi";*

*}*

*# UEFI-64-2*

*else if option system-arch = 00:08 {*

*filename "grub/grubx64.efi";*

*}*

*# UEFI-64-3*

*else if option system-arch = 00:09 {*

*filename "grub/grubx64.efi";*

*}*

*# armv7 (aka arm 32 bit)*

*else if option system-arch = 00:0a {*

*filename "grub/armv7.efi";*

*}*

*# aarch64 (aka arm 64 bit)*

*else if option system-arch = 00:0b {*

*filename "grub/grubaa64.efi";*

*}*

*# RiskV 32 bit*

*else if option system-arch = 00:25 {*

*#ToDo petitboot loader*

*filename "unsupported";*

*}*

*#RiskV 32 bit*

*else if option system-arch = 00:27 {*

*#ToDo petitboot loader*

*filename "unsupported";*

*}*

*else if option system-arch = 00:0e {*

*filename "grub/grub.ppc64le";*

*}*

*else*

*{*

*filename "****pxelinux.0****";*

*}*

*}*

*}*

Т.е. прописываем шаблон сети, адреса роутеров и сервера загрузки, а также указываем загружаемый клиент **pxelinux.0**

Остальное - не трогаем!

#### Первый запуск и проверка конфигурации Cobbler

После внесения всех изменений запускаем службу cobblerd

systemctl start cobblerd

Загружаем загрузчики

# cobbler get-loaders

Проводим проверку

# cobbler check

Выполняем все рекомендации предлагаемые проверкой конфигурации

Проводим синхронизацию  
# cobbler sync

Настройка Cobbler завершена.

#### Импорт дистрибутива Debian 10

Загружаем дистрибутив для сетевой установки

# wget -q -O /tmp/debian-10.2.0-amd64-netinst.iso https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-10.2.0-amd64-netinst.iso

Монтируем дистрибутив в локальный каталог

# mkdir -pv /mnt/debian

# mount /tmp/debian-10.2.0-amd64-netinst.iso /mnt/debian

Импортируем дистрибутив

# cobbler import --name=debian-10.2.0-amd64 --arch=x86\_64 --path=/mnt/debian

# umount /mnt/debian

Сам дистрибутив никакого ПО не содержит - он содержит только ядро Linux и окружение для инсталлятора 3х конфигураций

# cobbler distro list

debian-10.2.0-gtk-x86\_64

debian-10.2.0-x86\_64

debian-10.2.0-xen-x86\_64

Удаляем “лишние” для нас конфигурации

# cobbler profile remove --name=debian-10.2.0-gtk-x86\_64

# cobbler profile remove --name=debian-10.2.0-xen-x86\_64

# cobbler distro remove --name=debian-10.2.0-gtk-x86\_64

# cobbler distro remove --name=debian-10.2.0-xen-x86\_64

Но это еще не все - загрузчик с диска заточен на использование cdrom и не поднимает сетевой интерфейс в начале установки, поэтому установить с его помощью дистрибутив на сервер не удасться - инсталлятор будет требовать установку cd-диска. Для исправления ситуации качаем инсталлятор для сетевой установки и заменяем файлы инсталлятора в каталогах Cobbler

# mkdir -pv /tmp/netboot

# wget -q -O /tmp/netboot/netboot.tar.gz https://mirror.yandex.ru/debian/dists/buster/main/installer-amd64/current/images/netboot/netboot.tar.gz

# cd /tmp/netboot

# tar -xf /tmp/netboot/netboot.tar.gz

# cp -fv /tmp/netboot/debian-installer/amd64/linux /var/www/cobbler/distro\_mirror/debian-10.2.0-amd64-x86\_64/install.amd

# cp -fv /tmp/netboot/debian-installer/amd64/initrd.gz /var/www/cobbler/distro\_mirror/debian-10.2.0-amd64-x86\_64/install.amd

Проводим синхронизацию

# cobbler sync

Почему сразу не ставили дистрибутив с архива netboot.tar.gz? Cobbler при импорте дистрибутива проверяет сигнатуру дистрибутива и ставит только известные ему, поэтому изначально нам нужен оригинальный iso-образ.

#### Создание файла ответов для установки Debian

После импорта дистрибутива мы имеем инсталлятор, способный обратиться к любому зеркалу Debian для установки нужной конфигурации. Осталось создать файл ответов для установки дистрибутива в неинтерактивном режиме.

Нижеприведенный файл ответов был создан на базе /var/lib/cobbler/templates/sample.seed и сохранен как /var/lib/cobbler/templates/debian\_buster.seed.

*# Mostly based on the Ubuntu installation guide*

*# https://help.ubuntu.com/18.04/installation-guide/*

*# Debian sample*

*# https://www.debian.org/releases/stable/example-preseed.txt*

*# Preseeding only locale sets language, country and locale.*

*d-i debian-installer/locale string ru\_RU*

*# Keyboard selection.*

*# Disable automatic (interactive) keymap detection.*

*d-i console-setup/ask\_detect boolean false*

*d-i keyboard-configuration/xkb-keymap select us*

*d-i keyboard-configuration/toggle select No toggling*

*d-i keyboard-configuration/layoutcode string us*

*d-i keyboard-configuration/variantcode string*

*# netcfg will choose an interface that has link if possible. This makes it*

*# skip displaying a list if there is more than one interface.*

*set $myhostname = $getVar('hostname',$getVar('name','cobbler')).replace("\_","-")*

*d-i netcfg/choose\_interface select auto*

*#d-i netcfg/choose\_interface select eth0*

*#d-i netcfg/dhcp\_timeout string 60*

*#d-i netcfg/dhcpv6\_timeout string 60*

*d-i netcfg/get\_hostname string $myhostname*

*# If non-free firmware is needed for the network or other hardware, you can*

*# configure the installer to always try to load it, without prompting. Or*

*# change to false to disable asking.*

*# d-i hw-detect/load\_firmware boolean true*

*# NTP/Time Setup*

*d-i time/zone string Europe/Moscow*

*d-i clock-setup/utc boolean false*

*d-i clock-setup/ntp boolean true*

*d-i clock-setup/ntp-server string ntp.ubuntu.com*

*# Setup the installation source*

*d-i mirror/country string manual*

*d-i mirror/http/hostname string http.mirror.yandex.ru*

*d-i mirror/http/directory string /debian*

*#d-i mirror/http/hostname string $http\_server*

*#d-i mirror/http/directory string $install\_source\_directory*

*d-i mirror/http/proxy string*

*#set $os\_v = $getVar('os\_version','')*

*#if $breed == "ubuntu" and $os\_v and $os\_v.lower() != 'precise'*

*# Required at least for ubuntu 12.10+ , so test os\_v is not precise. Olders versions are not supported anymore*

*d-i live-installer/net-image string http://$http\_server/cobbler/links/$distro\_name/install/filesystem.squashfs*

*#end if*

*# Suite to install.*

*d-i mirror/suite string buster*

*d-i mirror/udeb/suite string buster*

*# d-i mirror/suite string precise*

*# d-i mirror/udeb/suite string precise*

*# Components to use for loading installer components (optional).*

*#d-i mirror/udeb/components multiselect main, restricted*

*# Disk Partitioning*

*# Use LVM, and wipe out anything that already exists*

*d-i partman/choose\_partition select finish*

*d-i partman/confirm boolean true*

*d-i partman/confirm\_nooverwrite boolean true*

*d-i partman-auto/method string lvm*

*d-i partman-lvm/device\_remove\_lvm boolean true*

*d-i partman-lvm/confirm boolean true*

*d-i partman-lvm/confirm\_nooverwrite boolean true*

*d-i partman-md/device\_remove\_md boolean true*

*d-i partman-partitioning/confirm\_write\_new\_label boolean true*

*# You can choose one of the three predefined partitioning recipes:*

*# - atomic: all files in one partition*

*# - home: separate /home partition*

*# - multi: separate /home, /usr, /var, and /tmp partitions*

*d-i partman-auto/choose\_recipe select atomic*

*# If you just want to change the default filesystem from ext3 to something*

*# else, you can do that without providing a full recipe.*

*# d-i partman/default\_filesystem string ext4*

*# root account and password*

*d-i passwd/root-login boolean true*

*d-i passwd/root-password-crypted password $default\_password\_crypted*

*# skip creation of a normal user account.*

*d-i passwd/make-user boolean false*

*# You can choose to install restricted and universe software, or to install*

*# software from the backports repository.*

*d-i apt-setup/restricted boolean true*

*d-i apt-setup/universe boolean true*

*d-i apt-setup/backports boolean true*

*# Uncomment this if you don't want to use a network mirror.*

*# d-i apt-setup/use\_mirror boolean false*

*# Select which update services to use; define the mirrors to be used.*

*# Values shown below are the normal defaults.*

*d-i apt-setup/services-select multiselect security*

*d-i apt-setup/security\_host string security.debian.org*

*d-i apt-setup/security\_path string /debian*

*$SNIPPET('preseed\_apt\_repo\_config')*

*# Enable deb-src lines*

*# d-i apt-setup/local0/source boolean true*

*# URL to the public key of the local repository; you must provide a key or*

*# apt will complain about the unauthenticated repository and so the*

*# sources.list line will be left commented out*

*# d-i apt-setup/local0/key string http://local.server/key*

*# By default the installer requires that repositories be authenticated*

*# using a known gpg key. This setting can be used to disable that*

*# authentication. Warning: Insecure, not recommended.*

*# d-i debian-installer/allow\_unauthenticated boolean true*

*# Package selection*

*# Default for minimal*

*tasksel tasksel/first multiselect standard*

*# Default for server*

*# tasksel tasksel/first multiselect standard, web-server*

*# Default for gnome-desktop*

*# tasksel tasksel/first multiselect standard, gnome-desktop*

*# Individual additional packages to install*

*# wget is REQUIRED otherwise quite a few things won't work*

*# later in the build (like late-command scripts)*

*d-i pkgsel/include string ntp ssh wget*

*# Debian needs this for the installer to avoid any question for grub*

*# Please verify that it suit your needs as it may overwrite any usb stick*

*#if $breed == "debian"*

*d-i grub-installer/grub2\_instead\_of\_grub\_legacy boolean true*

*d-i grub-installer/bootdev string default*

*#end if*

*# Use the following option to add additional boot parameters for the*

*# installed system (if supported by the bootloader installer).*

*# Note: options passed to the installer will be added automatically.*

*d-i debian-installer/add-kernel-opts string $kernel\_options\_post*

*# Avoid that last message about the install being complete.*

*d-i finish-install/reboot\_in\_progress note*

*## Figure out if we're automating OS installation for a system or a profile*

*#if $getVar('system\_name','') != ''*

*#set $what = "system"*

*#else*

*#set $what = "profile"*

*#end if*

*# This first command is run as early as possible, just after preseeding is read.*

*# d-i preseed/early\_command string [command]*

*d-i preseed/early\_command string wget -O- \*

*http://$http\_server/cblr/svc/op/script/$what/$name/?script=preseed\_early\_default | \*

*/bin/sh -s*

*# This command is run immediately before the partitioner starts. It may be*

*# useful to apply dynamic partitioner preseeding that depends on the state*

*# of the disks (which may not be visible when preseed/early\_command runs).*

*# d-i partman/early\_command \*

*# string debconf-set partman-auto/disk "\$(list-devices disk | head -n1)"*

*# This command is run just before the install finishes, but when there is*

*# still a usable /target directory. You can chroot to /target and use it*

*# directly, or use the apt-install and in-target commands to easily install*

*# packages and run commands in the target system.*

*# d-i preseed/late\_command string [command]*

*d-i preseed/late\_command string wget -O- \*

*http://$http\_server/cblr/svc/op/script/$what/$name/?script=preseed\_late\_default | \*

*chroot /target /bin/sh -s; wget -O- \*

*http://$http\_server/cblr/svc/op/nopxe/$what/$name*

В качестве зеркала используется mirror.yandex.ru. Данный вариант можно использовать для тестов или при наличии широкого и незагруженного канала в Интернет. В противном случае стоит воспользоваться утилитой ftpmirror (<https://www.debian.org/mirror/ftpmirror>), позволяющей создать собственное зеркало.

Также вышеприведенная конфигурация использует при разбиении весь диск целиком. Для пользовательского разбиения можно воспользоваться примером:

*d-i partman-auto/expert\_recipe string \*

*boot-root :: \*

*30000 5000 40000 ext4 \*

*$primary{ } $bootable{ } \*

*method{ format } format{ } \*

*use\_filesystem{ } filesystem{ ext4 } \*

*mountpoint{ / } \*

*. \*

*20000 3000 40000 ext3 \*

*method{ format } format{ } \*

*use\_filesystem{ } filesystem{ ext3 } \*

*mountpoint{ /srv } \*

*. \*

*10000 2000 10000000 ext4 \*

*method{ format } format{ } \*

*use\_filesystem{ } filesystem{ ext4 } \*

*mountpoint{ /home } \*

*. \*

*50% 6000 200% linux-swap \*

*method{ swap } format{ } \*

*.*

Для корректного создания файла ответов необходимо ознакомится с документацией <https://www.debian.org/releases/stable/mips/apbs04.ru.html>, которая позволит создать необходимую именно вам конфигурацию.

Хотелось бы обратить внимание, что в конце установки дистрибутива в файле ответов я указал вызов скрипта

*wget -O- http://$http\_server/cblr/svc/op/nopxe/$what/$name*

Это позволяет сбросить флаг сетевой загрузки для устанавливаемого сервера и после перезагрузки сервер стартует с уже установленного дистрибутива.

## Заключение

Дальнейшая работа с Cobbler предполагает создание систем (system) с указанием для каждого физического или виртуального сервера MAC-адреса, профайла дистрибутива, сетевых интерфейсов, а также имен и ip-адресов. К сожалению, CLI-версия утилиты cobbler выдает ошибку при проведении данной операции, поэтому я для этих целей воспользовался web-интерфейсом.

## Проблема с загрузкой серверов.

По какой-то причине, не смотря на указание в конфиге DHCP-сервера параметра *filename "****pxelinux.0****"* DHCP-сервер отдает значение “grub/grub.0”. По причине отсутствия такого файла в каталоге /var/lib/tftpboot загрузка сервера останавливается без установки операционной системы.

Первой попыткой решения было создание относительного симлинка с /var/lib/tftpboot/pxeclient.0 на /var/lib/tftpboot/grub/grub.0 .  
Данное решение частично работоспособно, т.к. сразу после создания симлинка сервер загружается без проблем, но после выполнения синхронизации по команде “cobbler sync” или через веб-кабинет при перезаписи загрузчиков Cobbler удаляет созданный симлинк.

Второй более успешной попыткой решения было создание копии файла pxeclient.0 в исходной папке хранения загрузчиков /var/lib/cobbler/loaders командами

# mkdir /var/lib/cobbler/loaders/grub

# cp /var/lib/cobbler/loaders/pxeclient.0 /var/lib/cobbler/loaders/grub/grub.0

# cd /var/lib/tftpboot/

# ln -s pxeclient.cfg ../pxeclient.cfg

# cobbler sync

При этом сервер загружается с папки /var/lib/tftpboot/grub/ и в этой папке ищет каталог с конфигурационными файлами pxeclient.cfg. Создание относительной символической ссылки решает данный вопрос и т.к. сама папка не пересоздается при синхронизации файлов cobbler’а обновленные в ней конфигурационные файлы становятся доступными из обоих папок.

## Работа с локальными репозиториями.

apt-get install ftpsync -y

Создаем конфигурационный файл /etc/ftpsync/ftpsync.conf

*RSYNC\_PATH="debian"*

*MIRRORNAME=`hostname -f`*

*BASEDIR=/home/debian*

*TO="/opt/mirrors/debian/"*

*RSYNC\_PATH="debian"*

*RSYNC\_HOST=ftp.fi.debian.org*

*LOGDIR="${BASEDIR}/log"*

*LOG="${LOGDIR}/${NAME}.log"*

*ERRORSONLY="true"*

*FULLLOGS="true"*

*ARCH\_EXCLUDE="alpha arm armel armhf hppa hurd-i386 i386 ia64 kfreebsd-amd64 kfreebsd-i386 m68k mipsel mips powerpc s390 s390x sh sparc source mips64el arm64 ppc64el"*

*LOGROTATE=14*

*LOCK="${TO}/Archive-Update-in-Progress-${MIRRORNAME}"*

*LOCKTIMEOUT=${LOCKTIMEOUT:-3600}*

*UPDATEREQUIRED="${TO}/Archive-Update-Required-${MIRRORNAME}"*

*TRACE="project/trace/${MIRRORNAME}"*

*EXTENDEDTRACE="full"*

*RSYNC=rsync*

*RSYNC\_OPTIONS="-prltvHSB8192 --timeout 3600 --stats --exclude Archive-Update-in-Progress-${MIRRORNAME} --exclude ${TRACE} --exclude Archive-Update-Required-${MIRRORNAME}"*

*RSYNC\_OPTIONS1="--include 'installer-\*/\*\*/\*' --include '\*/' --exclude 'dists/\*\*/\*' --exclude ls-lR\*"*

*RSYNC\_OPTIONS2="--max-delete=40000 --delay-updates --delete --delete-after --delete-excluded"*

Данный конфигурационный файл предполагает создание локального репозитория для архитектуры amd64 в каталоге указанном в переменной TO.

Запускаем создание локальной копии репозитория

# ftpsync

К сожалению с помощью данной программы не удалось получить копию репозиториев с обновлениями, решающими проблему безопасности, и копию репозиториев для установки Openstack Train. Исправим данный момент с помощью apt-mirror

apt-get install apt-mirror -y

Меняем строку 523 в исполняемом файле /bin/apt-mirror с

*if ( $filename =~ m{^$component/i18n/Translation-[^./]\*\.bz2$} )*

на

*if ( $filename =~ m{^$component/i18n/Translation-[^./]\*\.(gz|bz2|xz)$} )*

Источник <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=932112>

Данное решение решает проблему с тем, что apt-mirror по умолчанию не выкачивает файлы локализации в форматах gz и xz и локальная копия репозитория становится не работоспособной.

Правим конфигурационный файл /etc/apt/mirror.list

*#set nthreads 20*

*set nthreads 1*

*set \_tilde 0*

*deb http://buster-train.debian.net/debian buster-train-backports main*

*deb http://buster-train.debian.net/debian buster-train-backports-nochange main*

*deb http://buster-train.debian.net/debian buster-ceph-nautilus-backports main*

*deb-src http://buster-train.debian.net/debian buster-train-backports main*

*deb-src http://buster-train.debian.net/debian buster-train-backports-nochange main*

*deb-src http://buster-train.debian.net/debian buster-ceph-nautilus-backports main*

*clean http://buster-train.debian.net/debian*

*deb http://security.debian.org/debian-security buster/updates main*

*deb-src http://security.debian.org/debian-security buster/updates main*

*clean http://security.debian.org/debian-security*

*deb http://nyc2.mirrors.digitalocean.com/mariadb/repo/10.4/debian buster main*

*deb-src http://nyc2.mirrors.digitalocean.com/mariadb/repo/10.4/debian buster main*

*clean http://nyc2.mirrors.digitalocean.com/mariadb/repo/10.4/debian*

Формируем локальный репозиторий с доступом по http

# ln -s /opt/mirrors/debian /var/www/html/debian

# ln -s /var/spool/apt-mirror/mirror/buster-train.debian.net/debian/dists/buster-ceph-nautilus-backports \  
 /var/www/html/debian/dists/buster-ceph-nautilus-backports

# ln -s /var/spool/apt-mirror/mirror/buster-train.debian.net/debian/dists/buster-train-backports \  
 /var/www/html/debian/dists/buster-train-backports

# ln -s /var/spool/apt-mirror/mirror/buster-train.debian.net/debian/dists/buster-train-backports-nochange \

/var/www/html/debian/dists/buster-train-backports-nochange

# cp /var/spool/apt-mirror/mirror/buster-train.debian.net/debian/dists/pubkey.gpg \

/var/www/html/debian/dists/buster-train-backports-pubkey.gpg

# ln -s /var/spool/apt-mirror/mirror/buster-train.debian.net/debian/pool/buster-ceph-nautilus-backports \

/var/www/html/debian/pool/buster-ceph-nautilus-backports

# ln -s /var/spool/apt-mirror/mirror/buster-train.debian.net/debian/pool/buster-train-backports \

/var/www/html/debian/pool/buster-train-backports

# ln -s /var/spool/apt-mirror/mirror/buster-train.debian.net/debian/pool/buster-train-backports-nochange \  
 /var/www/html/debian/pool/buster-train-backports-nochange

# ln -s /var/spool/apt-mirror/mirror/security.debian.org/debian-security \  
 /var/www/html/debian-security

# ln -s /var/spool/apt-mirror/mirror/nyc2.mirrors.digitalocean.com/mariadb \  
 /var/www/html/mariadb

Скопируем ключи для доступа к этим репозиториям.

Для получения ключа в виде файла подключите удаленный репозиторий к одной машине. И экспортируйте его в файл. Рассмотрим пример для mariadb

Получаем публичный ключ репозитория

# apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8

Подключение самого репозитория (в данном случае необязательно)

# add-apt-repository 'deb [arch=amd64] http://nyc2.mirrors.digitalocean.com/mariadb/repo/10.4/debian buster main'

Ищем ключ в локальной базе

# apt-key list

/etc/apt/trusted.gpg

--------------------

pub rsa4096 2016-03-30 [SC]

177F 4010 FE56 CA33 3630 0305 F165 6F24 C74C D1D8

uid [ unknown] **MariaDB Signing Key <signing-key@mariadb.org>**

sub rsa4096 2016-03-30 [E]

....

Экспортируем ключ в файл

# apt-key export '177F 4010 FE56 CA33 3630 0305 F165 6F24 C74C D1D8' > 0xF1656F24C74CD1D8.gpg

Перенесите файл 0xF1656F24C74CD1D8.gpg в каталог /var/spool/apt-mirror/mirror/nyc2.mirrors.digitalocean.com/mariadb . С учетом созданной ранее символической ссылки этот файл будет доступен по HTTP http://$COBBLER\_IP/mariadb/0xF1656F24C74CD1D8.gpg

Пример использования репозитория

# curl http://$COBBLER\_IP/mariadb/0xF1656F24C74CD1D8.gpg | apt-key add -

# curl http://$COBBLER\_IP/buster-train.debian.net/debian/dists/pubkey.gpg | apt-key add -

# add-apt-repository "deb http://$COBBLER\_IP/debian-security buster/updates main"

# add-apt-repository "deb-src http://$COBBLER\_IP/debian-security buster/updates main"

# add-apt-repository "deb http://$COBBLER\_IP/debian buster-updates main"

# add-apt-repository "deb http://$COBBLER\_IP/debian buster-backports main"

# add-apt-repository "deb http://$COBBLER\_IP/debian buster-proposed-updates main"

# add-apt-repository "deb http://$COBBLER\_IP/debian buster-train-backports main"

# add-apt-repository "deb http://$COBBLER\_IP/debian buster-train-backports-nochange main"

# add-apt-repository "deb http://$COBBLER\_IP/debian buster-ceph-nautilus-backports main"

где переменная COBBLER\_IP содержит имя или ip сервера с репозиторием.

Результат можно посмотреть в файле /etc/apt/sources.list

*deb http://10.40.246.1/debian buster main*

*deb http://10.40.246.1/debian-security buster/updates main*

*deb http://10.40.246.1/debian buster-backports main*

*deb http://10.40.246.1/debian-security buster/updates main*

*deb http://10.40.246.1/debian buster-updates main*

*deb http://10.40.246.1/debian buster-backports main*

*deb http://10.40.246.1/debian buster-proposed-updates main*

*deb http://10.40.246.1/debian buster-train-backports main*

*deb http://10.40.246.1/debian buster-train-backports-nochange main*

*deb http://10.40.246.1/debian buster-ceph-nautilus-backports main*

Обновление данных и ПО.

# apt-get update && apt-get upgrade -y

Либо можно сразу внести локальные репозитории в файл ответов, используемый Cobbler:

*d-i mirror/http/directory string /debian*

*d-i mirror/http/hostname string $http\_server*

*d-i apt-setup/security\_host string $http\_server*

*d-i apt-setup/security\_path string /debian-security*

*d-i apt-setup/local0/repository string http://$http\_server/debian buster-backports main*

*d-i apt-setup/local1/repository string http://$http\_server/buster-train.debian.net/debian buster-ceph-nautilus-backports main*

*d-i apt-setup/local2/repository string http://$http\_server/buster-train.debian.net/debian buster-train-backports main*

*d-i apt-setup/local3/repository string http://$http\_server/buster-train.debian.net/debian buster-train-backports-nochange main*

*d-i apt-setup/local4/repository string http://$http\_server/mariadb/repo/10.4/debian buster main*

*d-i apt-setup/local0/key string http://$http\_server/buster-train.debian.net/debian/dists/pubkey.gpg*

*d-i apt-setup/local1/key string http://$http\_server/mariadb/0xF1656F24C74CD1D8.gpg*

В случае, если зеркала расположены не на сервере с Cobbler надо в указанных выше строчках сменить $http\_server - на явный ip-адрес или доменное имя.

**Создание систем в Cobbler**

В используемой версии Cobbler 3.1.1 на Debian 10 обнаружились досадные ошибки.

*# cobbler system add --name=svr1 --hostname=svr1 --mac=08:00:27:c6:a1:16 --ip=10.0.0.253 --profile=ol6u6\_basic\_server*

*Traceback (most recent call last):*

*File "/usr/local/bin/cobbler", line 35, in <module>*

*sys.exit(app.main())*

*File "/usr/local/lib/python3.7/dist-packages/cobbler/cli.py", line 738, in main*

*rc = cli.run(sys.argv)*

*File "/usr/local/lib/python3.7/dist-packages/cobbler/cli.py", line 351, in run*

*self.object\_command(object\_type, object\_action)*

*File "/usr/local/lib/python3.7/dist-packages/cobbler/cli.py", line 421, in object\_command*

*network\_interface\_fields, settings, object\_action)*

*File "/usr/local/lib/python3.7/dist-packages/cobbler/cli.py", line 231, in add\_options\_from\_fields*

*\_add\_parser\_option\_from\_field(parser, field, settings)*

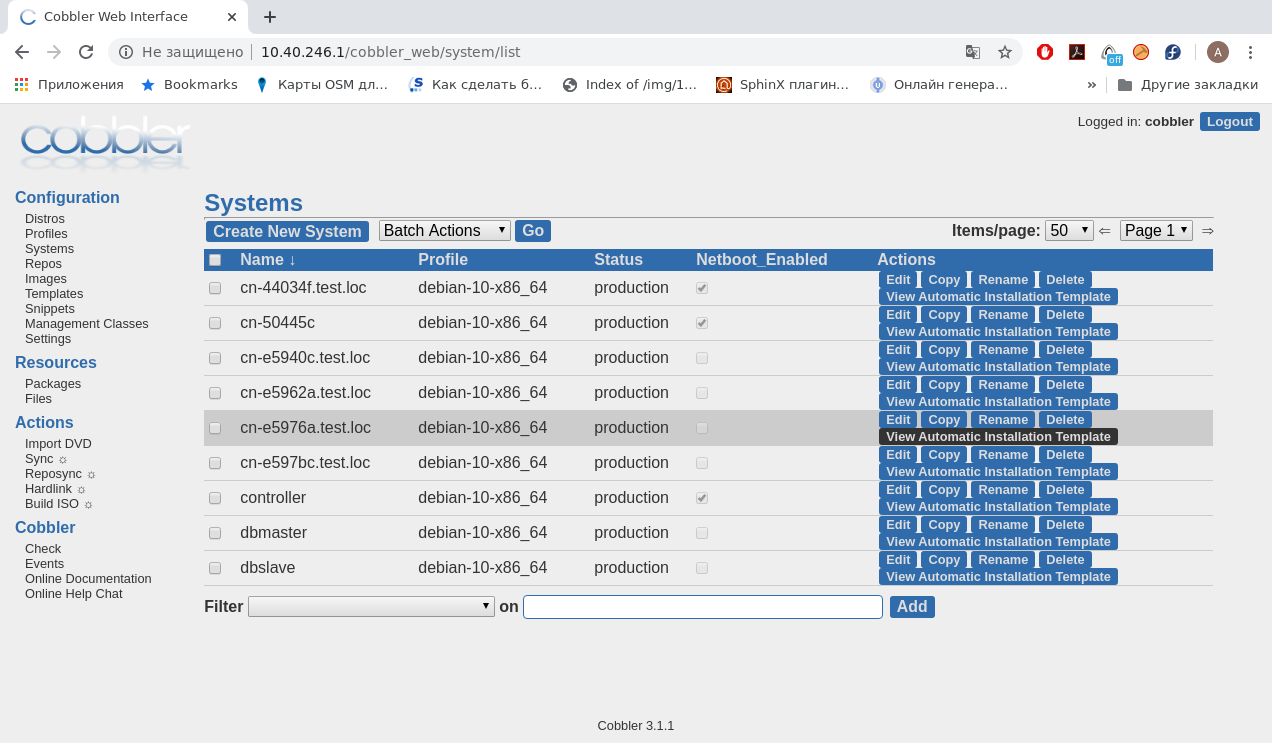
*File "/usr/local/lib/python3.7/dist-packages/cobbler/cli.py", line 202, in \_add\_parser\_option\_from\_field*

*raise Exception("field %s default value (%s) is not listed in choices (%s)" % (name, default, str(choices)))*

*Exception: field power\_type default value (ipmitool) is not listed in choices (['ack\_manual', 'aliyun', 'alom', 'amt', 'apc', 'apc\_snmp', 'aws', 'azure\_arm', 'bladecenter', 'brocade', 'cisco\_mds', 'cisco\_ucs', 'compute', 'docker', 'drac', 'drac5', 'dummy', 'eaton\_snmp', 'emerson', 'eps', 'evacuate', 'gce', 'hds\_cb', 'heuristics\_ping', 'hpblade', 'ibmblade', 'idrac', 'ifmib', 'ilo', 'ilo2', 'ilo3', 'ilo3\_ssh', 'ilo4', 'ilo4\_ssh', 'ilo5', 'ilo5\_ssh', 'ilo\_moonshot', 'ilo\_mp', 'ilo\_ssh', 'imm', 'intelmodular', 'ipdu', 'ipmilan', 'ironic', 'kdump', 'ldom', 'lpar', 'mpath', 'netio', 'openstack', 'ovh', 'powerman', 'pve', 'raritan', 'rcd\_serial', 'rhevm', 'rsa', 'rsb', 'sanbox2', 'sbd', 'scsi', 'tripplite\_snmp', 'vbox', 'virsh', 'vmware', 'vmware\_rest', 'vmware\_soap', 'vmware\_vcloud', 'wti', 'xenapi', 'zvmip'])*

Попытка явно указать power\_type или power-type в качестве параметра также успеха не принесла. Поэтому указываю вариант настройки системы через веб-интерфейс.

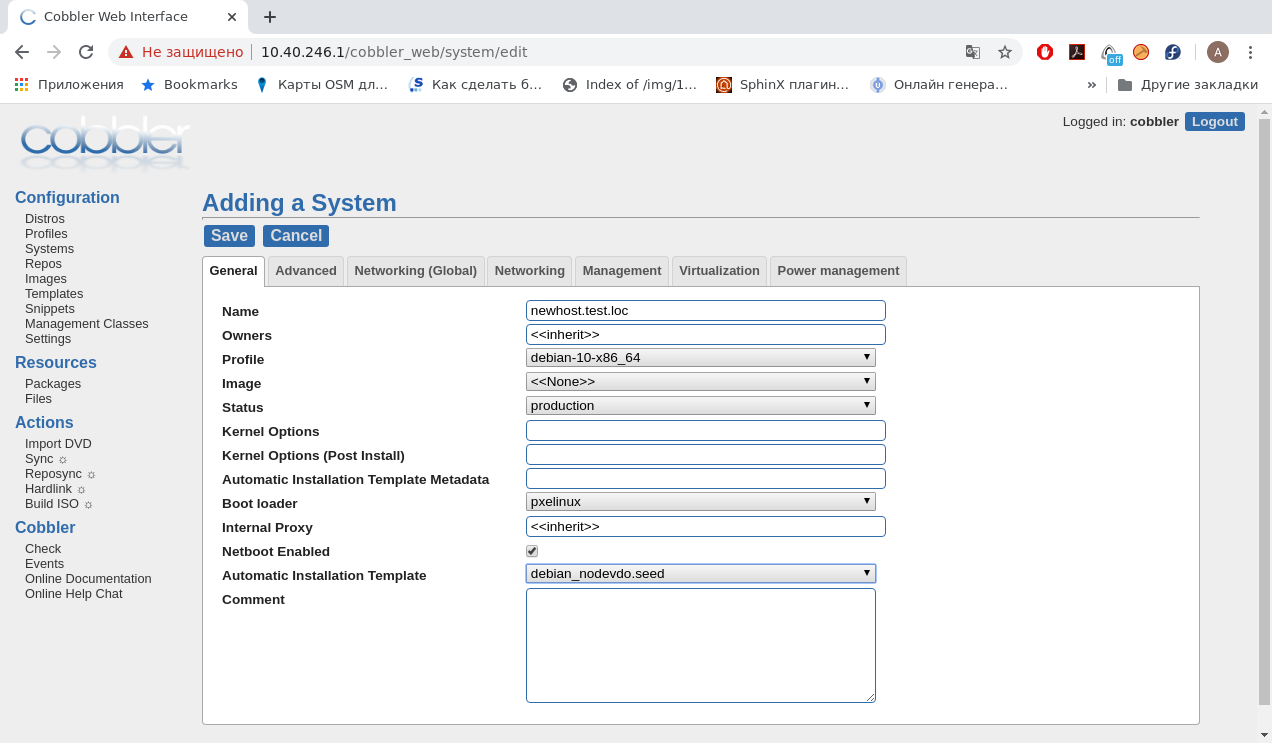
Для начала перейдем в раздел System и нажмем кнопку “Create New System”



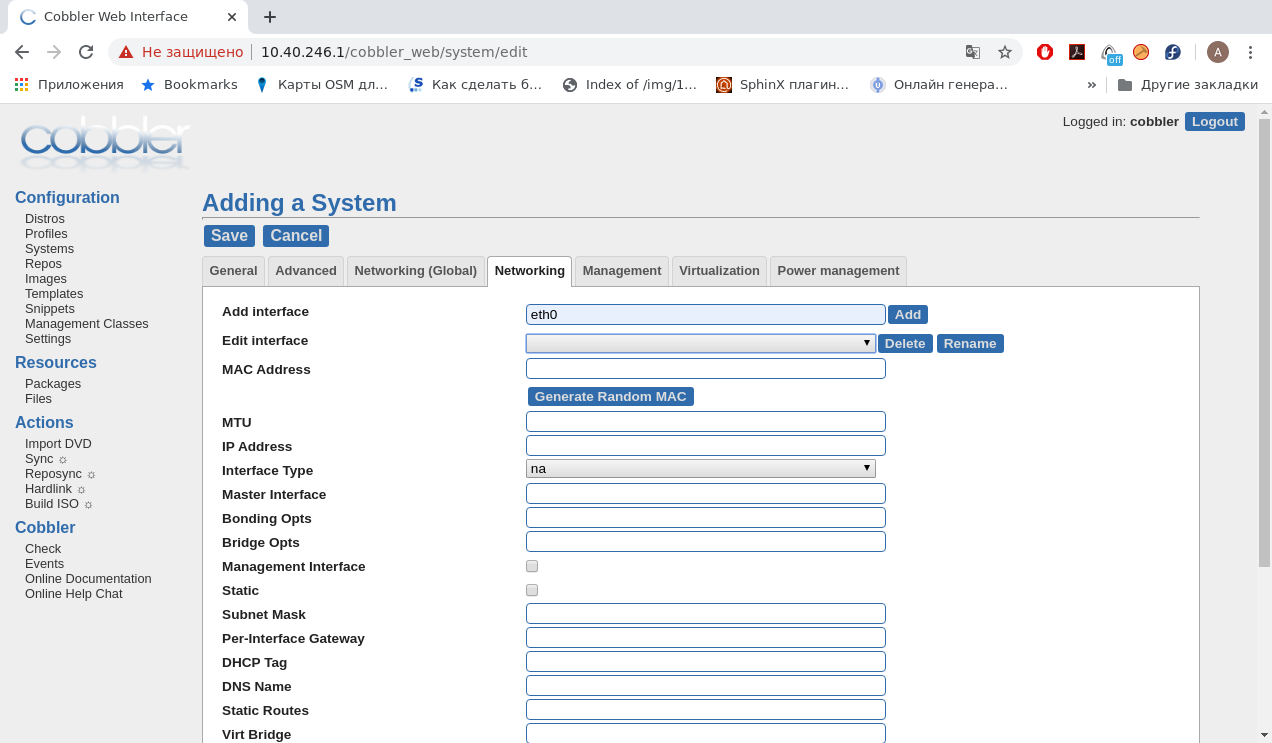
Затем введем имя системы на вкладке “General” так, как оно будет выглядеть в Cobbler. Можно указать как часть имени полностью или частично MAC-адрес или назначение сервера. Затем обязательно выбрать:

- “Profile” - в нашем случае это “debian-10-x86\_64”,  
 - “Bootloader” - pxelinux.

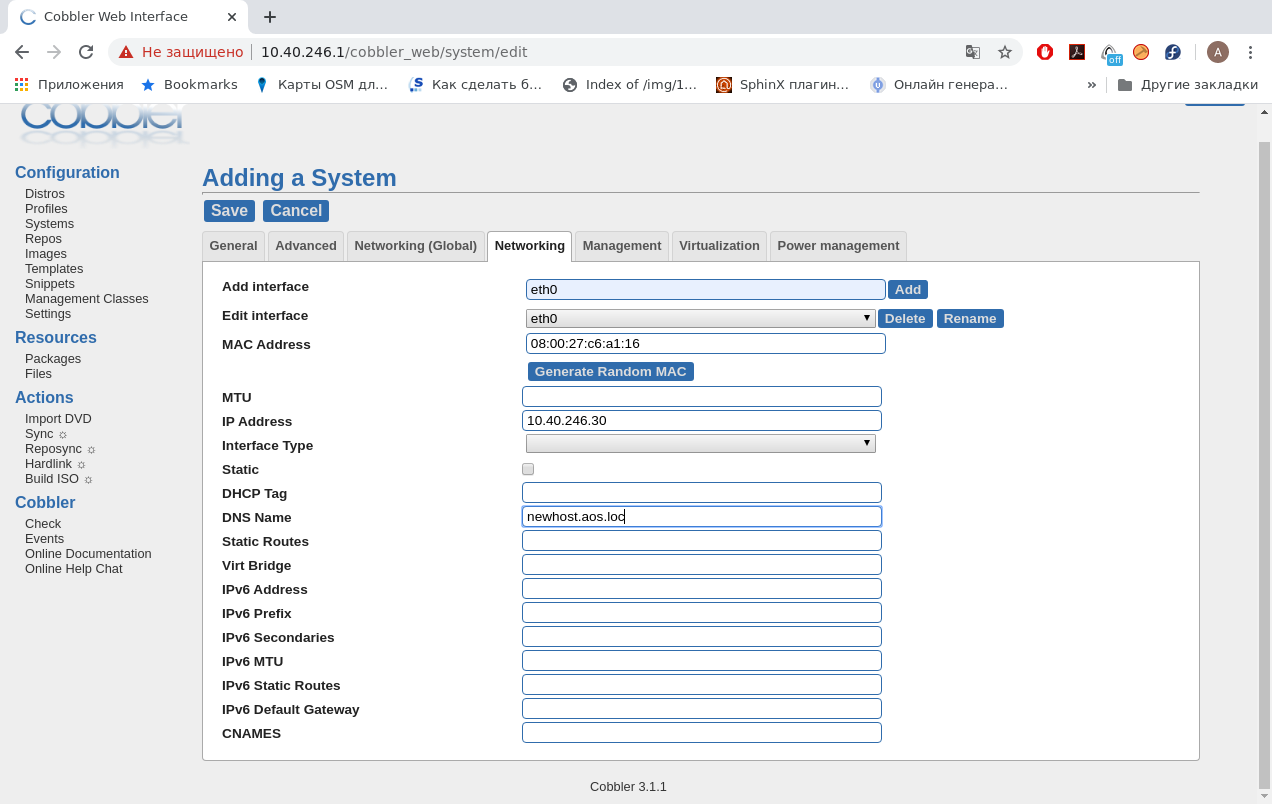
- “Netboot Enabled” - по умолчанию для новой системы включено,  
 - “Automatic Installation Template” - выбираем заранее подготовленный файл ответов исходя из аппаратной конфигурации сервера и необходимой программной настройки.



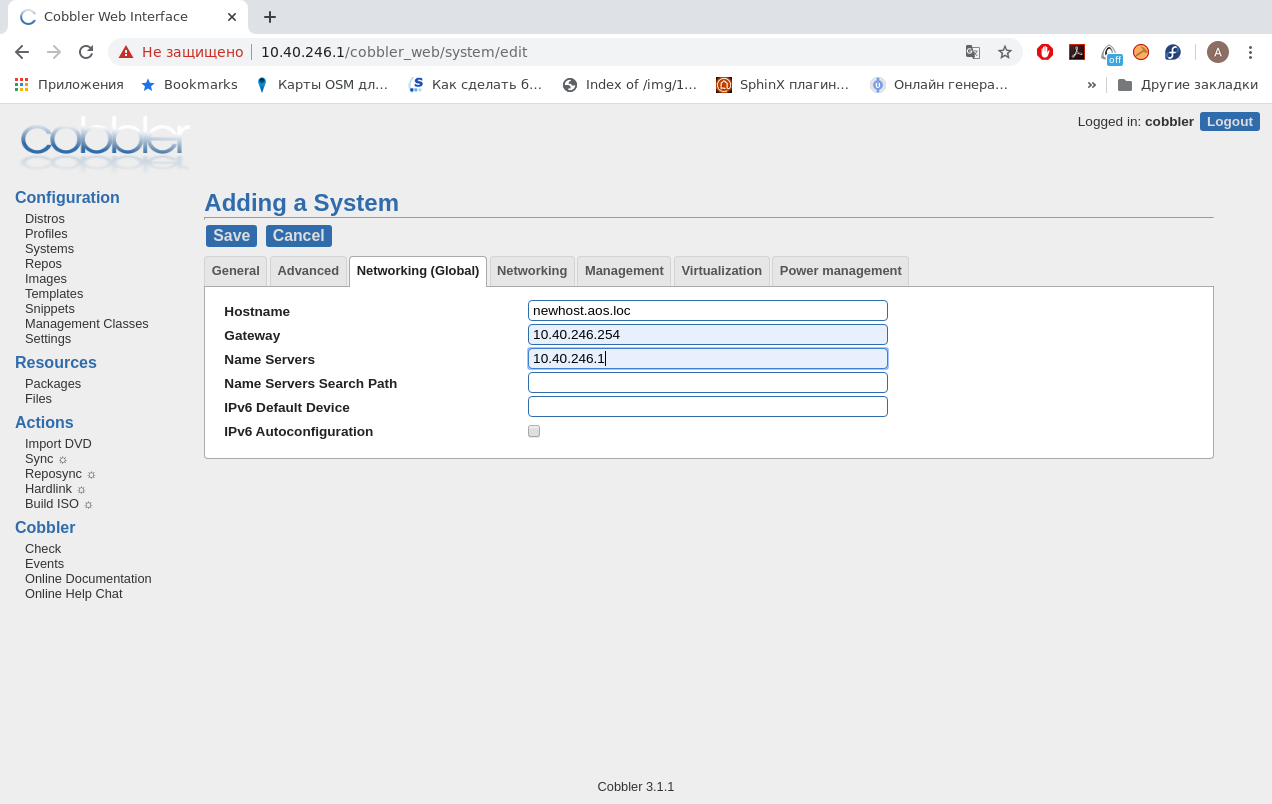
Затем переходим на вкладку “Networking” - сетевые настройки привязываются к интерфейсам, поэтому первым шагом добавляем интерфейс - вводим имя интерфейса в поле “Add interface” и нажимаем кнопку Add.



После добавления интерфейса можно вносить необходимые данные: “MAC”, “IP Address” и “DNS Name”. Остальные поля вбивать нет смысла - все будет определено через файл ответов.



Переходим во вкладку “Networking (Global)”. Данные введенные на этой вкладке могут быть сброшены после переопределения сетевых интерфейсов во вкладке “Networking”. Этим и определен порядок настройки. Заполняем поля “Hostname”, “Gateway” и “Name Servers”



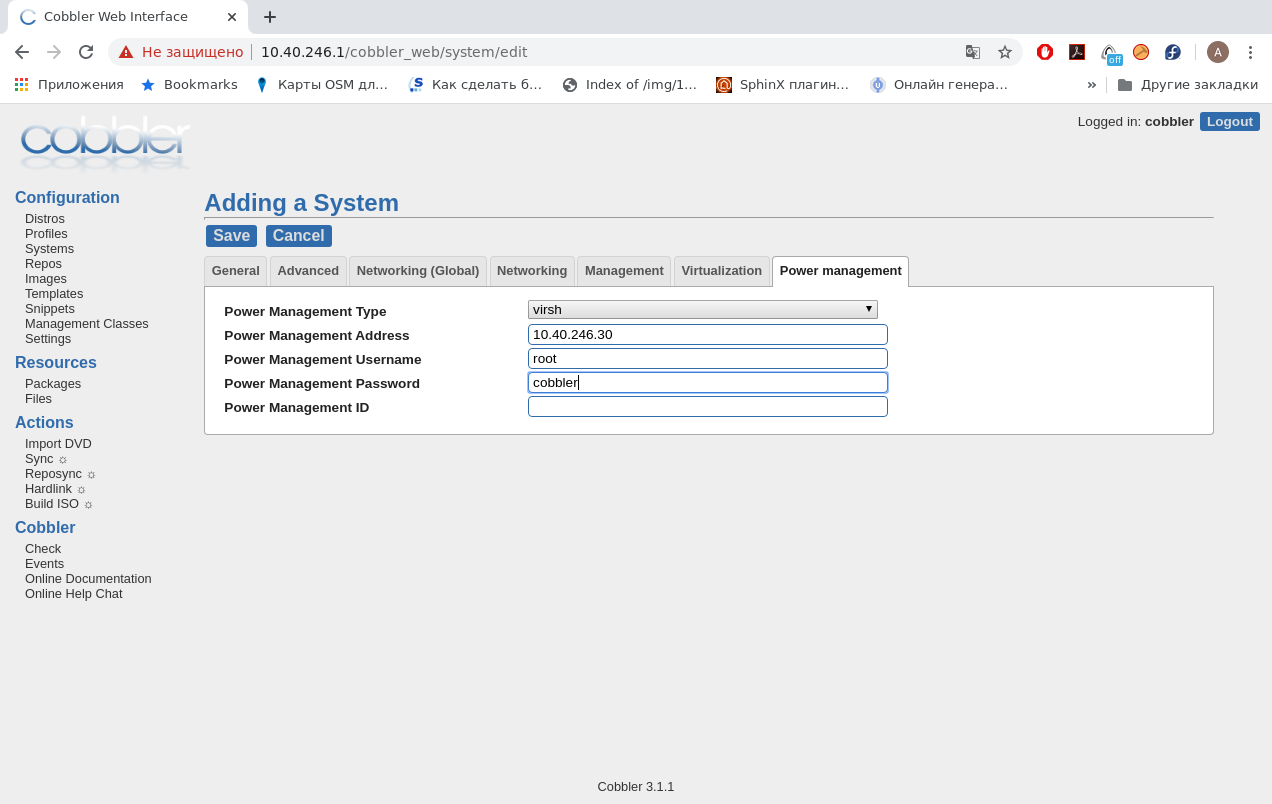
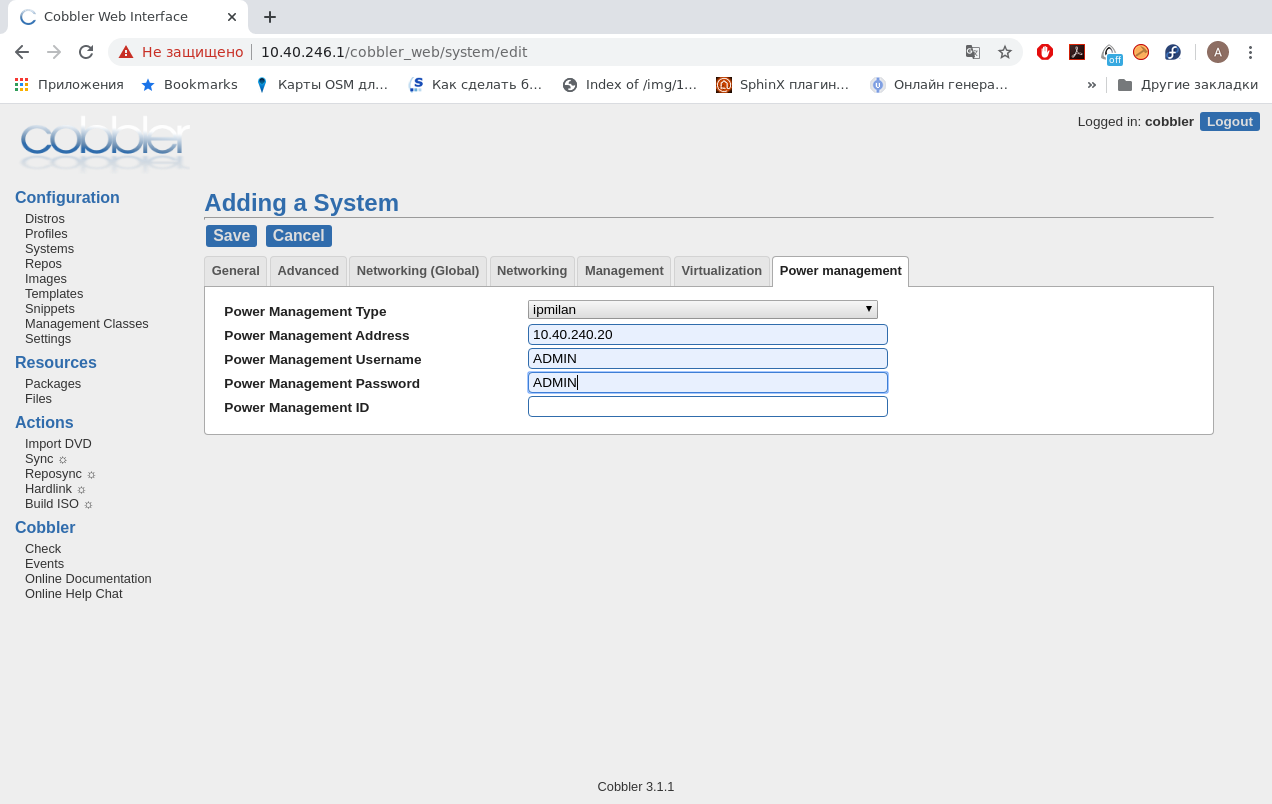
Последним заполняем вкладку “Power managment”. Тут существует много вариантов, но мы рассмотрим только два: IPMI и KVM.

В случае IPMI выбираем

* “**Power Management Type**” - ipmilan
* “**Power Management Address**” - ip адрес управления сервером
* “**Power Management Username**” - имя пользователя
* “**Power Management Password**” - пароль
* “**Power Management ID**” - порт IPMI, если он настроен не на стандартное значение.

В случае KVM выбираем

* “**Power Management Type**” - virsh
* “**Power Management Address**” - ip адрес управления сервером
* “**Power Management Username**” - имя пользователя
* “**Power Management Password**” - пароль
* “**Power Management ID**” - порт libvirt, если он настроен не на стандартное значение. Libvirt должен принимать подключения по сети.



Вкладка “Virtualization” существует для создания виртуальных машин, но так как оно не умеет настраивать openvswitch бриджи - пользоваться данным функционалом мы не будем. Как и вкладкой “Managment” - репозитории Debian мы создаем и подключаем самостоятельно и создаем виртуальные машины из xml файлов, подготовленных заранее.

Жмем кнопку “Save” для сохранения введенных данных.

После создания системы необходимо кликнуть на “Sync” - данное действие произведет настройку DHCP и TFTP для корректной загрузки системы.

После этого физические сервера можно запустить отметив галочками, выбрав в выпадающем меню “Batch Actions” поле “Power On” и нажав кнопку “Go”

Для разрешения повторной установки по сети необходимо отметить галочками нужные системы, выбрав в выпадающем меню “Batch Actions” поле “Power On” и нажав кнопку “Go”. После чего выбрать “Power On” или “Reboot” в качестве нужной операции при нажатии кнопки “Go”.

**Приложение 3 Инструкция по настройке и установке программного обеспечения WebGard 2.0 в в средстве виртуализации AccentOS**

Аннотация

Настоящий документ представляет собой руководство по развертыванию, настройке и использованию программного обеспечения WebGard 2.0 в среде виртуализации AccentOS, а также способы решения типичных проблем, возникающих при работе программы.

# WebGard

**CОДЕРЖАНИЕ**

[Состав оборудования и документации 2](#_Toc115278176)

[Работа с репозиториями 29](#_Toc115278177)

[Импорт GPG-ключей для авторизации репозитория пакетов ПО 29](#_Toc115278178)

[Формирование файла /etc/apt/sources.list с определениями расположения репозиториев 29](#_Toc115278179)

[Отключение интерактивная установки пакетов debian 30](#_Toc115278180)

[Установка часового пояса 30](#_Toc115278181)

[Определение настроек сети развертывания 30](#_Toc115278182)

[Сетевая конфигурация вычислительных узлов 31](#_Toc115278183)

[Настройки DNS 31](#_Toc115278184)

[Настройки NTP 32](#_Toc115278185)

[Создание системы хранения на базе Ceph 33](#_Toc115278186)

[Глобальные настройки облака 34](#_Toc115278187)

[Подключение ISCSI дисков 34](#_Toc115278188)

[Настройка кластера OCFS2 35](#_Toc115278189)

[Использование VDO 36](#_Toc115278190)

[Распределение служб облака по физическим и виртуальным серверам. 39](#_Toc115278191)

[Конфигурация высокой доступности. 39](#_Toc115278192)

[Адреса контроллеров Openstack 39](#_Toc115278193)

[|Служба Memcache 40](#_Toc115278194)

[Адреса балансировщика внешних запросов haproxy. 40](#_Toc115278195)

[Адреса кластера базы данных MariaDB 40](#_Toc115278196)

[Адреса кластера службы очередей сообщений RabbitMQ 41](#_Toc115278197)

[Адреса кластера etcd 41](#_Toc115278198)

[Адреса БД Redis 41](#_Toc115278199)

[Адреса и настройки Consul 42](#_Toc115278200)

[Настройки отдельных модулей Openstack 42](#_Toc115278201)

[Keystone 42](#_Toc115278202)

[Glance 43](#_Toc115278203)

[Gnocchi 45](#_Toc115278204)

[Nova 47](#_Toc115278205)

[Neutron 47](#_Toc115278206)

[Zun 48](#_Toc115278207)

[Octavia 49](#_Toc115278208)

[Настройка модулей AccentOS 50](#_Toc115278209)

[Устаревшие параметры 50](#_Toc115278210)

[Предварительная подготовка 63](#_Toc115278211)

[Установка программного обеспечения 63](#_Toc115278212)

[Настройка DHCP-сервера. 64](#_Toc115278213)

[Установка и настройка Cobbler 64](#_Toc115278214)

[Установка Cobbler 64](#_Toc115278215)

[Настройка Cobbler 65](#_Toc115278216)

[Заключение 77](#_Toc115278217)

[Проблема с загрузкой серверов. 77](#_Toc115278218)

[Работа с локальными репозиториями. 78](#_Toc115278219)

[1. Назначение Программы 93](#_Toc115278220)

[1.1. Основные возможности Программы 93](#_Toc115278221)

[1.2. Ограничения, накладываемые на область применения Программы 102](#_Toc115278222)

[2. Условия применения 104](#_Toc115278223)

[2.1. Эксплуатационные ограничения 105](#_Toc115278224)

[3. Описание задачи 106](#_Toc115278225)

[3.1. Обработка http-запросов 106](#_Toc115278226)

[3.2. Аутентификация и авторизация субъектов доступа 107](#_Toc115278227)

[3.3. Регистрация и учет действий субъектов доступа 107](#_Toc115278228)

[4. Архитектура системы WebGard 2.0 109](#_Toc115278229)

[5. Создание виртуальных машин WebGard 2.0 111](#_Toc115278230)

[5.1. Создание с виртуальной машины из диска 111](#_Toc115278231)

[5.2. Создание виртуальной машины из снимка виртуальной машины 112](#_Toc115278232)

[6. Установка WebGard 2.0 В среде AccentOS 114](#_Toc115278233)

[6.1. Работа с сетевыми настройками в интерфейсе AccentOS 114](#_Toc115278234)

[6.2. Настройка второго сетевого интерфейса в виртуальной машине 115](#_Toc115278235)

[6.3. Настройка DNS 116](#_Toc115278236)

[6.4. Проверка доступности с помощью скрипта conn\_checker.sh 116](#_Toc115278237)

[6.5. Настройка скрипта для создания бэкапов 117](#_Toc115278238)

[6.6. Настройка и запуск конфигурации виртуальных машин 118](#_Toc115278239)

[6.7. Проверка работы кластера виртуальных машин 119](#_Toc115278240)

[7. Подключение к подсистеме фильтрации и подсистеме администратора WebGard 2.0 121](#_Toc115278241)

[7.1. Вход в подсистему администрирования и предварительные настройки 121](#_Toc115278242)

[7.1. Вход в подсистему фильтрации 121](#_Toc115278243)

[7.2. Вход в подсистему фильтрации с помощью rs-client AccentOS 121](#_Toc115278244)

[8. Организация работы программы WebGard 2.0 в режиме высокой доступности 123](#_Toc115278245)

[8.1. Проверка кластера виртуальных машин 123](#_Toc115278246)

[8.2. Создание снимка виртуальной машины с помощью интерфейса AccentOS 123](#_Toc115278247)

[8.3. Восстановление виртуальной машины из снимка виртуальной машины 124](#_Toc115278248)

[9. Дополнительное взаимодействие с системой WebGard 2.0 125](#_Toc115278249)

[9.1. Проверка кластера виртуальных машин 125](#_Toc115278250)

[9.2. Редактирование парольной политики 125](#_Toc115278251)

[9.3. Редактирование политики логина 125](#_Toc115278252)

[9.4. Удаление администратора WebGard 2.0 126](#_Toc115278253)

[Приложение 1 126](#_Toc115278254)

[Название скрипта: etcd\_cluster.sh 126](#_Toc115278255)

[Приложение 2 126](#_Toc115278256)

[Название скрипта: conn\_cheker.sh 126](#_Toc115278257)

[Создание образа 130](#_Toc115278258)

[Создание сетевых портов 132](#_Toc115278259)

[Создание группы безопасности 135](#_Toc115278260)

[Создание виртуальных машин. 138](#_Toc115278261)

[Создание общего диска для Samba. 147](#_Toc115278262)

[Запуск скрипта установки и настройки Samba 153](#_Toc115278263)

[Необходимая подготовка домена ActiveDirectory 162](#_Toc115278264)

[Общее доменное имя. 162](#_Toc115278265)

[Указание размещения профилей пользователей и домашних папок. 163](#_Toc115278266)

**Перечень сокращений и терминов**

Перечень используемых сокращений и терминов представлен в таблице (Таблица 1).

Таблица 1 – Перечень используемых сокращений и терминов

|  |  |
| --- | --- |
| **Сокращение** | **Полное наименование** |
| HTTP | Hyper Text Transfer Protocol (протокол передачи гипертекста) |
| SQL | Structured Query Language (язык структурированных запросов) |
| URI | Uniform Resource Identifier (унифицированный идентификатор ресурса) |
| URL | Uniform Resource Locator (унифицированный указатель ресурса) |
| АС | Автоматизированная система |
| БД | База данных |
| НСД | Несанкционированный доступ |
| ОС | Операционная система |
| ПО | Программное обеспечение |
| Пользователь | Пользователь программного обеспечения «WebGard 2.0» |
| Программа | Программное обеспечение «WebGard 2.0» |
| СУБД | Система управления базой данных |
| УЦ | Удостоверяющий центр |
| ФСТЭК России | Федеральная служба по техническому и экспортному контролю России |

Программное обеспечение «WebGard 2.0» предназначено для защиты информации, не относящейся к государственной тайне, от несанкционированного доступа в web-системах массового обслуживания и реализует разграничение доступа при обращении к web-ресурсам для web-приложений.

## Основные возможности Программы

Программа обеспечивает выполнение следующих функций безопасности по защите информации:

* идентификация и аутентификация (логин/пароль, двухфакторная, LDAP) пользователей защищаемых web-систем (ИАФ.1);
* идентификация и аутентификация администраторов Программы (ИАФ.1);
* управление идентификаторами (синхронизация, блокирование, предотвращение повторного использования) (ИАФ.3);
* управление средствами аутентификации (хранение, обновление, защита) пользователей web-систем (ИАФ.4);
* возможность изменения характеристик пароля (ИАФ.4);
* назначение механизмов аутентификации (ИАФ.4);
* защита аутентификационной информации (ИАФ.5);
* управление учётными записями (заведение, активация, блокирование, контроль, уничтожение) пользователей web-систем (УПД.1);
* оповещение администраторов об изменении привилегий пользователей и параметров Программы (путем формирования почтового сообщения) (УПД.1);
* управление учётными записями администраторов (УПД.1);
* разграничение доступа в соответствии с ролевой политикой безопасности (УПД.2);
* ограничение неуспешных попыток входа пользователей в защищаемую информационную систему (УПД.6);
* оповещение пользователя при входе в защищаемую информационную систему (УПД.7, УПД.8);
* ограничение числа параллельных сеансов доступа для каждой учетной записи пользователя защищаемой информационной системы (УПД.9);
* блокирование (закрытие) сеанса доступа пользователя в защищаемую информационную систему при наступлении определенных событий (УПД.10);
* регистрация и защита информации о событиях безопасности пользователей web-систем (РСБ.1, РСБ.2);
* регистрация и защита информации о событиях безопасности администраторов Программы (РСБ.1, РСБ.2);
* сбор, запись и хранение информации о событиях безопасности (РСБ.3);
* предоставление администраторам возможности реагирования на сбои при регистрации событий безопасности (РСБ.4);
* предоставление возможности просмотра результатов регистрации событий безопасности (РСБ.5);
* защита информации о событиях безопасности (РСБ.7);
* разделение полномочий пользователей web-систем и администраторов Программы (разделение интерфейса пользователя и интерфейса администратора) (ЗИС.1);
* контроль вводимых данных для исключения ввода недопустимых символов (ОЦЛ.7);
* идентификация и аутентификация пользователей в интерфейсе управления виртуальной инфраструктурой (ЗСВ.1);
* управление доступом пользователей к интерфейсу управления виртуальной инфраструктурой (ЗСВ.2);
* регистрация событий безопасности в интерфейсе управления виртуальной инфраструктурой (ЗСВ.3);
* управление ресурсами виртуальной инфраструктуры через интерфейс управления виртуальной инфраструктурой (ЗСВ.6);
* фильтрация HTTP- запросов пользователей защищаемых web-систем;
* возможность автоматизированного внесения пользователей защищаемой информационной системы в список легитимных пользователей Программы.

Реализация выполнения функций безопасности обеспечивается в соответствии с:

* «Требованиями о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах» (приказ ФСТЭК России № 17 от 11.02.2013 г.) (далее по тексту - [1]);
* Методическим документом «Меры защиты информации в государственных информационных системах» (утвержден ФСТЭК России 11 февраля 2014 г.);
* «Составом и содержанием организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных» (приказ ФСТЭК России № 21 от 18.02.2013 г.) (далее по тексту - [2]).

Основные возможности Программы:

1. Обеспечивается идентификация и аутентификация пользователей, являющихся работниками оператора (ИАФ.1) [1,2]:

* идентификация и аутентификация пользователей с использованием паролей;
* идентификация и аутентификация администраторов с использованием паролей;
* при аутентификации по протоколу LDAP, выполнение запроса на аутентификацию пользователя в существующий сервер службы каталогов;
* возможность однозначного сопоставления идентификатора пользователя с выполняемыми от его имени запросами;
* многофакторная (двухфакторная) аутентификация пользователей для удаленного доступа в систему c использованием ESMART карт и/или USB-идентификатора, поддерживаемого сертифицированной версией КриптоПро CSP:

1. с использованием сети связи общего пользования, в том числе сети Интернет;
2. без использования сети связи общего пользования.
3. Установлены и реализованы функции управления идентификаторами, в том числе создание, присвоение, уничтожение идентификаторов (ИАФ.3) [1,2]:

* присвоение идентификатора пользователя в Программе, который позволяет однозначно идентифицировать пользователя;
* предотвращение повторного использования идентификатора пользователя в Программе в течение установленного администратором периода времени;
* автоматическое блокирование идентификатора пользователя после установленного администратором времени неиспользования логина.

1. Установлены и реализованы функции управления средствами аутентификации, в том числе хранение, выдача, инициализация, блокирование средств аутентификации и принятие мер в случае утраты и (или) компрометации средств аутентификации (ИАФ.4) [1,2]:

* предоставление возможности изменения аутентификационной информации.
* установление характеристик пароля, а именно:

1. установка минимальной и максимальной длины пароля в символах;
2. установка минимальной сложности пароля с определяемыми требованиями к регистру, сочетанию букв верхнего и нижнего регистра, цифр и специальных символов;
3. установка требования к алфавиту пароля;
4. установка максимального времени действия пароля;

* назначение характеристик механизмов аутентификации:

1. срок, в течение которого возможно сменить пароль;
2. время жизни аккаунта (логина);
3. время, которое будет ожидать пользователь перед следующей попыткой аутентификации;
4. время, по истечении которого сбрасывается счетчик неуспешных попыток аутентификации.

* обновление аутентификационной информации (замена средств аутентификации) с периодичностью, установленной администратором;
* защита аутентификационной информации от неправомерных доступа к ней и модифицирования.

1. Обеспечивается защита обратной связи при вводе аутентификационной информации (ИАФ.5) [1,2]:

* защита аутентификационной информации в процессе ее ввода для аутентификации путем сокрытия ее отображения условными знаками.

1. Установлены и реализованы функции управления (заведение, активация, блокирование и уничтожение) учетными записями пользователей, в том числе внешних пользователей (УПД.1) [1,2]:

* в Программе установлены и реализованы следующие функции управления учетными записями пользователей, в том числе внешних пользователей:
* наличие типов учетных записей (временная, внутренняя, внешняя и предустановленная);
* объединение учетных записей в группы при помощи ролей;
* заведение, активация, блокирование и уничтожение учетных записей пользователей;
* заведение и редактирование учетных записей администраторов;
* возможность редактирования учетных записей пользователей;
* оповещение администратора, осуществляющего управление учетными записями пользователей, об изменении сведений о пользователях, их ролях, полномочиях, ограничениях;
* предоставление администратору возможности блокирования и уничтожения временных учетных записей пользователей, предоставленных для ограниченного по времени выполнения задач в Программе;
* в Программе осуществляется автоматическое блокирование временных учетных записей пользователей по окончании установленного периода времени для их использования;
* в Программе осуществляется автоматическое блокирование неактивных (неиспользуемых) учетных записей пользователей после окончания периода времени неиспользования, установленного администратором;
* в Программе осуществляется автоматическое блокирование учетных записей пользователей при превышении установленного администратором числа неуспешных попыток аутентификации пользователя.

1. Обеспечена реализация необходимых методов (дискреционный, мандатный, ролевой или иной метод), типов (чтение, запись, выполнение или иной тип) и правил разграничения доступа (УПД.2) [1,2]:

* ПО обеспечивает ролевой метод управления доступом, предусматривающий управление доступом субъектов доступа (пользователей и администраторов) к объектам доступа (web-ресурсам и настройкам безопасности Программы) на основе ролей:

Примечание: Объектами доступа должны являться функции, для которых назначаются элементы защищаемых web-ресурсов. К каждому субъекту доступа (пользователь) должна назначаться роль с функциями, разрешенными к выполнению, при получении доступа к защищаемым web-ресурсам.

* в ПО выделяются роли пользователей и администраторов;
* для каждой пары (субъект – объект) в ПО должно быть задано явное и недвусмысленное перечисление допустимых http-запросов (GET, POST, OPTIONS, HEAD, PUT, DELETE, PATCH, ANY), т.е. для тех http-запросов, которые являются санкционированными для данного субъекта доступа к данному – объекту доступа;
* контроль доступа должен быть применим к каждому объекту и каждому субъекту;
* ПО обеспечивает управление доступом субъектов к защищаемым web-ресурсам при входе в Программу, и разграничивает доступ к следующим полномочиям:
* создание правил управления доступом (для каждой пары (субъект – объект) в ОО должно быть задано явное и недвусмысленное перечисление допустимых http-запросов (GET, POST, OPTIONS, HEAD, PUT, DELETE, PATCH, ANY), т.е. для тех http-запросов, которые являются санкционированными для данного субъекта доступа к данному – объекту доступа);
* переход на защищаемый ресурс;
* редактирование правил управления доступом;
* удаление правил управления доступом;
* создание субъекта доступа;
* редактирование субъекта доступа;
* удаление субъекта доступа;
* синхронизация прав доступа;
* создание защищаемых ресурсов;
* редактирование защищаемых ресурсов;
* удаление защищаемых ресурсов;
* изменение привилегий учетных записей;
* вход (выход), а также попытки входа субъектов доступа в панель управления компонентами виртуальной инфраструктуры;
* изменение в составе и конфигурации компонентов виртуальной инфраструктуры во время их запуска и функционирования;
* изменение правил разграничения доступа к компонентам виртуальной инфраструктуры;
* размещение и перемещение файлов-образов виртуальных машин (контейнеров) между носителями (системами хранения данных);
* размещение и перемещение исполняемых виртуальных машин (контейнеров) между серверами виртуализации;
* размещение и перемещение данных, обрабатываемых с использованием виртуальных машин, между носителями (системами хранения данных).

1. Обеспечивается ограничение неуспешных попыток входа в информационную систему (доступа к информационной системе) (УПД.6) [1,2]:

* в Программе обеспечивается автоматическое блокирование учетной записи пользователя при превышении пользователем ограничения количества неуспешных попыток входа в Программу за установленный период времени.

1. Реализовано предупреждение пользователя при его входе в информационную систему о том, что в информационной системе реализованы меры защиты информации, и о необходимости соблюдения им установленных правил обработки информации (УПД.7) [1,2]:

* обеспечивается предупреждение пользователя в виде сообщения («окна») о том, что в Программе реализованы меры защиты информации, а также о том, что при работе пользователем должны быть соблюдены установленные правила и ограничения на работу с информацией.

1. В Программе обеспечивается оповещение пользователя после успешного входа в информационную систему о его предыдущем входе в информационную систему (УПД.8) [1,2]:

* обеспечивается оповещение пользователя после успешного входа в Программу (завершения процесса аутентификации) о дате и времени предыдущего успешного и (или) неуспешного входа в Программу от имени этого пользователя, а также об успешности процесса аутентификации.

1. Обеспечивается ограничение числа параллельных сеансов доступа для каждой учетной записи пользователя информационной системы (УПД.9) [1,2]:

* выполняется ограничение числа параллельных сеансов доступа для каждой учетной записи пользователя;
* предусмотрена возможность задавать ограничения на число параллельных (одновременных) сеансов (сессий) пользователей, основываясь на идентификаторах пользователей;
* для привилегированных учетных записей (администраторов) количество параллельных (одновременных) сеансов (сессий) от их имени не превышает 2;
* в случае попытки входа под учетной записью пользователя или администратора, для которых достигнуто максимальное значение допустимых параллельных сеансов, при успешной аутентификации пользователя или администратора выдается сообщение о превышении числа параллельных сеансов доступа;
* в Программе предусмотрены средства, позволяющие контролировать и отображать администратору число активных параллельных (одновременных) сеансов (сессий) для каждой учетной записи пользователей.

1. Обеспечивается блокирование сеанса доступа в информационную систему после установленного времени бездействия (неактивности) пользователя или по его запросу (УПД.10) [1,2]:

* обеспечивается блокирование (закрытие) сеанса доступа пользователя после установленного администратором времени его бездействия (неактивности) в Программе или по запросу пользователя;
* для заблокированного сеанса осуществляется блокирование любых действий по доступу к информации;
* блокирование сеанса доступа пользователя в Программу сохраняется до прохождения им повторной идентификации и аутентификации.

1. Регистрируются события безопасности и сроки их хранения (РСБ.1) [1,2]:

* вход (выход), а также попытки входа субъектов доступа в защищаемую информационную систему;
* события, связанные с действиями от имени привилегированных учетных записей (администраторов):
* создание правил управления доступом;
* редактирование правил управления доступом;
* удаление правил управления доступом;
* создание субъекта доступа;
* редактирование субъекта доступа;
* удаление субъекта доступа;
* синхронизация прав доступа;
* события безопасности, связанные с действиями пользователей в Программе:
* переход на защищаемый ресурс;
* создание защищаемых ресурсов;
* редактирование защищаемых ресурсов;
* удаление защищаемых ресурсов;
* события безопасности, связанные с изменением привилегий учетных записей пользователей;
* обеспечивается хранение информации о зарегистрированных событиях безопасности.

1. Определен состав и содержание информации о событиях безопасности, подлежащих регистрации (РСБ.2) [1,2]. Для каждого события безопасности регистрируются:

* состав и содержание информации о действиях администраторов, включаемой в записи регистрации о событиях безопасности, обеспечена возможность регистрации:
* имя субъекта, совершившего инициацию события безопасности;
* ip-адрес хоста;
* дата и время события безопасности;
* тип выполненной операции;
* результат совершения операции;
* состав и содержание информации о действиях пользователей, включаемой в записи регистрации о событиях безопасности, обеспечена возможность регистрации:
* дата и время события безопасности;
* ip-адрес хоста;
* идентификатор пользователя;
* имя субъекта, совершившего действие;
* метод запроса;
* унифицированный указатель ресурса;
* выполненную функцию;
* статус события;
* параметры HTTP-запроса;
* параметры тела HTTP-запроса;
* при регистрации входа (выхода) пользователей в Программу состав и содержание информации включает дату и время входа (выхода) в систему (из системы), результат попытки входа (успешная или неуспешная), идентификатор, предъявленный при попытке доступа, метод запроса и путь web-ресурса;
* при регистрации попыток удаленного доступа к защищаемой информационной системе состав и содержание информации включает дату и время попытки удаленного доступа с указанием ее результата (успешная, неуспешная), идентификатор субъекта доступа, метод запроса и путь web-ресурса.

1. Осуществляется сбор, запись и хранение информации о событиях безопасности в течение установленного времени (РСБ.3) [1,2]:

* выбор и просмотр администраторами событий безопасности из списка совершенных событий (фильтрация параметров);
* генерация (сбор, запись) записей регистрации (аудита) для событий безопасности, подлежащих регистрации (аудиту) в соответствии с составом и содержанием информации, определенными в соответствии с параметрами регистрации;
* хранение информации о событиях безопасности.

1. Обеспечивается возможность реагирования на сбои при регистрации событий безопасности, в том числе аппаратные и программные ошибки, сбои в механизмах сбора информации и достижение предела или переполнения объема (емкости) памяти (РСБ.4):

* обеспечивается возможность изменения администраторами параметров сбора, записи и хранения информации о событиях безопасности, запись поверх устаревших хранимых записей событий безопасности;

1. Осуществляется мониторинг (просмотр) результатов регистрации событий безопасности и реагирование на них (РСБ.5) [1,2]:

* обеспечивается возможность просмотра записей регистрации, в документации на ПО установлена периодичность анализа записей регистрации администратором.

1. Обеспечивается защита информации о событиях безопасности (РСБ.7) [1,2]:

* обеспечивается защита информации о событиях безопасности в Программе;
* доступ к записям аудита и функциям управления механизмами регистрации (аудита) предоставляется только администраторам Программы.

1. Реализовано разделение в Программе функций по управлению (администрированию) информационной системой, управлению (администрированию) системой защиты информации, функций по обработке информации и иных функций информационной системы (ЗИС.1) [1,2]:

* в Программе обеспечено разделение функциональных возможностей по управлению (администрированию) системой защиты информации (функций безопасности) и функциональных возможностей пользователей по обработке информации (наличие выделенного интерфейса администрирования).

1. Обеспечивается контроль точности, полноты и правильности данных, вводимых в информационную систему (ОЦЛ.7) [1,2]:

* контроль точности, полноты и правильности данных, вводимых (email, дата, числовые значения настроек безопасности) в Программу. Обеспечивается путем установления и проверки соблюдения форматов ввода данных, (допустимые наборы символов, размерность, область числовых значений, допустимые значения, количество символов) для подтверждения того, что ввод информации соответствует заданному администратором формату и содержанию.

1. Обеспечивается идентификация и аутентификация субъектов доступа и объектов доступа в виртуальной инфраструктуре, в том числе администраторов управления средствами виртуализации (ЗСВ.1) [1,2]:

* идентификация и аутентификация администраторов управления средствами виртуализации;
* идентификация и аутентификация субъектов доступа при удалённом обращении к объектам доступа в виртуальной инфраструктуре;
* блокировка доступа к компонентам виртуальной инфраструктуры для субъектов доступа, не прошедших процедуру аутентификации;
* защита аутентификационной информации субъектов доступа от неправомерного доступа к ней, уничтожения или модифицирования;
* защита аутентификационной информации в процессе ее ввода для аутентификации в виртуальной инфраструктуре от возможного использования лицами, не имеющими на это полномочий;
* идентификация и аутентификация субъектов доступа при осуществлении ими попыток доступа к средствам управления параметрами аппаратного обеспечения элементов виртуальной инфраструктуры.

1. Установлены и реализованы следующие функции управления доступом субъектов доступа к объектам доступа в виртуальной инфраструктуре, в том числе внутри виртуальных машин (ЗСВ.2) [1,2]:

* контроль доступа субъектов доступа к средствам управления компонентами виртуальной инфраструктуры;
* контроль доступа субъектов доступа к файлам-образам виртуализированного программного обеспечения, виртуальных машин, файлам-образам, служебным данным, используемым для обеспечения работы виртуальных файловых систем, и иным служебным данным средств виртуальной среды;
* управление доступом к виртуальному аппаратному обеспечению защищаемого ресурса, являющимся объектом доступа;
* обеспечение доступа к операциям, выполняемым с помощью средств управления виртуальными машинами, в том числе к операциям создания, запуска, останова, создания образов, удаления виртуальных машин, который должен быть разрешен только администраторам виртуальной инфраструктуры;
* обеспечение доступа к конфигурации виртуальных машин только администраторам виртуальной инфраструктуры.

1. Обеспечивается регистрация событий безопасности в виртуальной инфраструктуре, (ЗСВ.3) [1,2]:

* запуск (завершение) работы компонентов виртуальной инфраструктуры;
* доступ субъектов доступа к компонентам виртуальной инфраструктуры;
* изменение в составе и конфигурации компонентов виртуальной инфраструктуры во время их запуска и функционирования;
* изменение правил разграничения доступа к компонентам виртуальной инфраструктуры.

1. Обеспечено управление перемещением виртуальных машин (контейнеров) и обрабатываемых на них данных (ЗСВ.6) [1,2]:

* управление размещением и перемещением файлов-образов виртуальных машин (контейнеров) между носителями (системами хранения данных);
* управление размещением и перемещением исполняемых виртуальных машин (контейнеров) между серверами виртуализации;
* управление размещением и перемещением данных, обрабатываемых с использованием виртуальных машин, между носителями (системами хранения данных);
* полный запрет перемещения виртуальных машин (контейнеров);
* ограничение перемещения виртуальных машин (контейнеров) в пределах защищаемой информационной системы (сегмента защищаемой информационной системы.

Дополнительные основные возможности:

1. Фильтрация HTTP - запросов, поступающих в защищаемую web-систему:

* фильтрация запросов, поступающих в защищаемую систему по протоколу HTTP;
* фильтрация запросов по режимам (все запрещено, все разрешено).

1. Обеспечивается возможность автоматизированного занесения данных пользователей (аутентификационных данных) защищаемой информационной системы в базу данных Программы.

## Ограничения, накладываемые на область применения Программы

Программу предполагается использовать как элемент защиты государственных информационных систем и информационных систем персональных данных.

Программа может быть использована в автоматизированных системах до класса защищенности 1Г включительно, в информационных системах персональных данных до 1 уровня включительно и в государственных информационных системах до класса К1 включительно.

Реализация функций безопасности Программы протестирована на следующих web-серверах:

* Apache HTTP Server;
* nginx;
* IIS;
* lighttpd;
* litespeed.

## Условия применения

Программа включает в себя следующие подсистемы:

* подсистема администрирования;
* подсистема HTTP фильтрации;
* подсистема хранилища данных СУБД PostgreSQL;
* подсистема кэширования данных.

Для выполнения Программой всех заявленных функций необходимо использовать ОС «Альт 8 СП» ЛКНВ.11100-01.

Минимальные характеристики технических средств, используемых для функционирования Программы:

* процессор: 2 ядра, 2,4 ГГц;
* оперативная память: от 4 ГБ;
* жёсткий диск: от 250 ГБ;
* сеть: Ethernet-интерфейс со скоростью 1 Гбит/с, 2шт.

Окружение, в котором предполагается функционирование Программы, состоит из следующих компонентов:

* Java 9;
* gnu tar, gzip;
* СУБД PostgreSQL 12;
* Apache Tomcat/9.0.13.

Для проверки требований по защите среды виртуализации используется платформа виртуализации под управлением ПО OpenStack с графическим интерфейсом администрирования Horizon и/или ПО AccentОS и гипервизор из состава сертифицированной ОС.

Для выполнения Программой всех заявленных функций, необходимо соблюдение следующих организационных мер:

* прохождение обучения сотрудников, допускаемых к работе с Программой, и ознакомление их с эксплуатационной документацией;
* наличие администратора (или службы) защиты информации, ответственного за функционирование, а также контроль работы Программы;
* осуществление физической охраны информационных систем персональных данных (устройств и носителей информации), предусматривающее контроль доступа посторонних лиц в помещения с установленной информационной системой персональных данных, наличие надежных препятствий для несанкционированного проникновения в указанные помещения и хранилище носителей информации, особенно в нерабочее время;
* осуществление учета всех защищаемых носителей информации с помощью их маркировки с занесением учетных данных в журнал (учетную карточку);
* учет защищаемых носителей должен проводиться в журнале (картотеке) с регистрацией их выдачи (приема);
* обеспечение возможности восстановления используемых средств защиты персональных данных, предусматривающая ведение двух копий каждого средства защиты, их периодическое обновление и контроль работоспособности.

## Эксплуатационные ограничения

При эксплуатации Программы должны быть выполнены следующие ограничения:

1. отсутствие каких-либо сторонних сетевых маршрутов в настройках Программы и в установленных ОС;
2. подключение к ЗИС должно быть организовано только через ПО «WebGard 2.0», сторонние сетевые подключения через другие технические средства должны отсутствовать (в том числе виртуальные);
3. использование Программы предполагается только на следующих портах:

* 80;
* 5432;
* 11211
* 8080;
* 9009;
* 9010;
* 9011.

1. необходимо устранить уязвимости среды функционирования Программы посредством установки актуальных обновлений безопасности;
2. доступ к аутентификационной информации (в том числе хэшам паролей) Программы должен предоставляться только доверенному списку администраторов.

## Описание задачи

Основная задача Программы – защита информации, не относящейся к государственной тайне, от несанкционированного доступа в web-системах массового обслуживания, реализация разграничения доступа при обращении к web-ресурсам для web-приложений.

Для реализации функции защиты информации от несанкционированного доступа Программа реализует ролевое управление доступом. Ролевое управление доступом является основным механизмом обеспечения конфиденциальности, целостности и доступности объектов многопользовательской системы. Конфиденциальность и целостность информации обеспечивается путем запрещения обслуживания неавторизованных пользователей.

Осуществление ролевого управления доступом предусматривает выполнение следующих функций:

* выполнение аутентификации и авторизации субъектов доступа;
* регистрация и учет действий, выполняемых субъектами доступа в защищаемой системе;
* фильтрация http-запросов.

Для осуществления ролевого управления доступом определяется множество допустимых функций для каждой пары «роли» – «функции», а также производится контроль выполнения правил вызова функций http-webguard (фильтр запросов). Описание функций содержится в базе данных Программы и включает в себя следующую информацию:

* наименование функции;
* URL;
* тип запроса;
* тип функции;
* перечень входных параметров.

## Обработка http-запросов

Обработка http-запросов Программой включает в себя выполнение следующих этапов:

* прием http- запросов по протоколу HTTP;
* выполнение аутентификации субъектов доступа;
* авторизация субъектов доступа;
* при успешной авторизации – выполнение операций в http-webguard;
* регистрация запроса и результатов авторизации для запрошенной операции;
* фильтрация http-запросов субъектов доступа;
* аудит http-запросов.

## Аутентификация и авторизация субъектов доступа

При входе в систему производится идентификация и проверка подлинности субъектов доступа по паролю условно-постоянного действия длиной не менее восьми буквенно-цифровых символов и имеющим минимум 2 цифры и 2 буквы.

Идентификация объектов доступа производится по их именам.

Контроль доступа субъектов к объектам доступа осуществляется на основе проверки у них необходимых прав доступа в соответствии с матрицами доступа «роль» – «функция», «роль» – «запрос».

## Регистрация и учет действий субъектов доступа

Программа позволяет осуществлять сбор и накопление информации о событиях, происходящих в Программе. События подразделяются на внутренние (аудит действий в администрировании безопасности) и внешние (аудит действий пользователя). В процессе регистрации и учета реализуются следующие задачи:

* обеспечение подотчетности субъектов доступа;
* обеспечение возможности реконструкции последовательности событий;
* обнаружение попыток нарушений информационной безопасности;
* предоставление информации для выявления и анализа проблем.

События безопасности, регистрирующиеся в Программе:

1. вход (выход), а также попытки входа субъектов доступа в защищаемую информационную систему;
2. события, связанные с действиями от имени привилегированных учетных записей (администраторов):

* создание объекта доступа;
* редактирование объекта доступа;
* удаление объекта доступа;
* создание субъекта доступа;
* редактирование субъекта доступа;
* удаление субъекта доступа;
* синхронизация прав доступа;

1. события безопасности, связанные с действиями пользователей:

* переход на ресурс в защищаемой информационной системе;
* создание ресурсов защищаемой информационной системы;
* редактирование ресурсов защищаемой информационной системы;
* удаление ресурсов защищаемой информационной системы;
* просмотр объектов базы данных защищаемой информационной системы;
* создание объектов базы данных защищаемой информационной системы;
* редактирование объектов базы данных защищаемой информационной системы;
* удаление объектов базы данных защищаемой информационной системы;

1. события безопасности, связанные с изменением привилегий учетных записей пользователей.

В Программе обеспечивается хранение информации о зарегистрированных событиях безопасности.

Состав и содержание информации, включаемой в регистрацию о событиях безопасности (администраторов):

* имя субъекта, совершившего инициацию события безопасности;
* ip-адрес хоста;
* тип объекта доступа;
* дата и время события безопасности;
* тип выполненной операции;
* результат совершения операции.

Состав и содержание информации, включаемой в регистрацию о событиях безопасности (пользователя):

* дата и время события безопасности;
* ip-адрес хоста;
* идентификатор пользователя;
* имя субъекта, совершившего действие;
* метод запроса;
* унифицированный указатель ресурса;
* выполненную функцию;
* статус события.

## Архитектура системы WebGard 2.0

Работа пользователей с WebGard 2.0 осуществляется с помощью клиентского ПО RS Client.

Система аутентификации WebGard 2.0 устанавливается на управляющие узлы в виде ВМ с ОС Альт 8.2 СП.

Система аутентификации WebGard 2.0 включает в себя:

* ПО WebGard 2.0;
* ПО Memcached (для базы правил и базы журнала);
* ПО PostgreSQL.

Система аутентификации WebGard 2.0 работает в режиме высокой доступности, для чего запускается несколько ВМ WebGard 2.0. В этом режиме работа пользователей с WebGard 2.0 осуществляется с настройками клиентского ПО RS Client через HA Proxy по порту 9365.

Система работает в режиме обслуживания двух доменов - MS AD и FreeIPA.

Для этого выполняется запуск двух копий (экземпляров) WebGard 2.0 внутри ВМ с использованием механизма Chroot. Каждый экземпляр WebGard 2.0 внутри ВМ пишет в свою базу СУБД PostgreSQL.

Для подключения к соответствующему экземпляру пользователи MS AD и FreeIPA и должны использовать разные IP адреса (у MS AD IP1 и FreeIPA IP2).

Режим высокой доступности обеспечивается для файловой системы ВМ, сервера приложения WebGard 2.0 и сервера баз данных PostgreSQL.

Высокая доступность серверов приложений WebGard 2.0 обеспечивается их одновременным запуском на двух ВМ и распараллеливанием их работы с независимыми Memcached экземплярами.

Высокая доступность серверов баз данных PostgreSQL обеспечивается их одновременным запуском на трех ВМ и организацией их бесперебойной работы с помощью ПО Patroni. Балансировка нагрузки на СУБД обеспечивается с помощью HA Proxy, встроенного в Patroni.

На двух ВМ WebGard 2.0 выполняется запуск полноценной ВМ с работающим WebGard 2.0, Patroni + PostgreSQL.

На третьей ВМ-Кворум запускается только Patroni + PostgreSQL для обеспечения кворума PostgreSQL.

Это позволяет обеспечить работу WebGard 2.0:

* в случае, когда одна ВМ WebGard 2.0 вышла из строя;
* в случае, когда ВМ-Кворум вышла из строя;
* в случае, когда одна ВМ WebGard 2.0 и ВМ-Кворум одновременно вышли из строя.

Patroni присваивает серверу СУБД на одной ВМ статус «активный» и остальным СУБД статус «пассивный». В случае выхода из строя «активной» ВМ, система выполняет автоматическое переключение в соответствии с установленным тайм-аутом. По умолчанию время тайм-аута составляет 30 сек.

После включения вышедшей из строя ВМ система скриптов синхронизации Patroni должна автоматически восстановить целостность СУБД, ввести ее в кластер со статусом «пассивный» и стартовать WebGard 2.0.

Схема взаимодействия компонентов WebGard 2.0 в кластерном двухдоменном режиме показана на рисунке (Рис. 1).

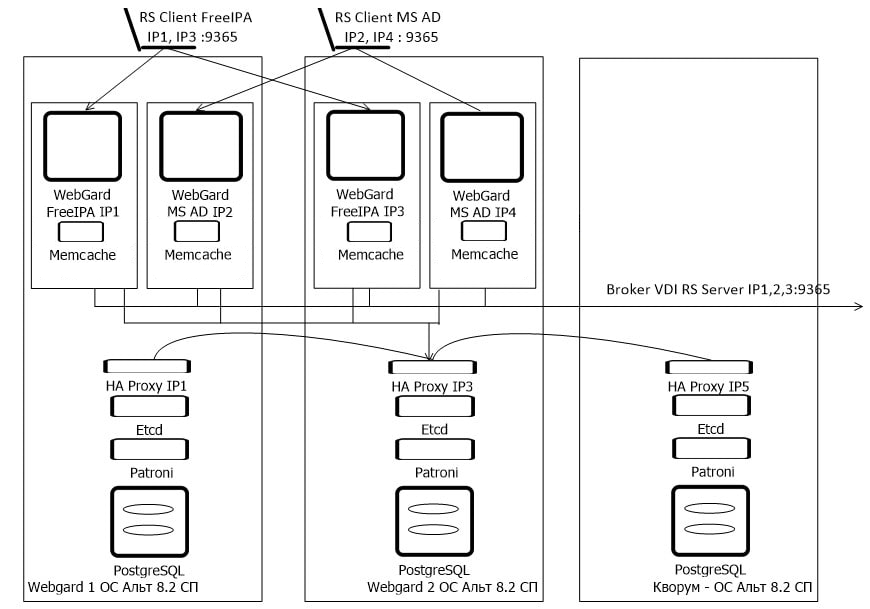


Рис. 1 Схема взаимодействия компонентов WebGard 2.0

## Создание виртуальных машин WebGard 2.0

## Создание с виртуальной машины из диска

Для создания виртуальных машин с программным обеспечением WebGard 2.0 требуется создать 3 диска для виртуальных машин. Для этого необходимо:

1. войти в web интерфейс AccentOS;
2. перейти в раздел: Проект -> Ресурсы облачного хранилища, нажать «Диски»;
3. нажать кнопку «Создать диск»;
4. в открывшемся окне необходимо: заполнить поле «Имя диска» (например, wg-1);
5. в поле «Источник диска» выбрать «Образ»;
6. в поле «Использовать образ как источник» выбрать необходимый образ (например, wg-script-220720)

Примечание: если образ источник не видно в поле после выбора – ничего страшного. Выбор может не отображаться в поле, но диск создастся из выбранного образа.

1. в поле «Тип» выбрать «lvm»;
2. в поле «Размер» выбрать «15»;
3. нажать кнопку «Создать диск»;
4. повторить пункты 3-9 для второго и третьего диска;

Далее требуется создать 3 виртуальные машины на основе созданных дисков. Для этого необходимо:

1. перейти в раздел: Проект -> Виртуальные машины;
2. нажать кнопку «Запустить машину»;
3. перейти во вкладку «Детали»;
4. в поле «Имя виртуальной машины» ввести название для виртуальных машин – wg-1;
5. в пункте «Количество» ввести необходимое кол-во виртуальных машин - 1;
6. перейти во вкладку «Источник»;
7. в поле «Выберите источник загрузки» выбрать «Диск»;
8. в поле «Удалить диск при удалении виртуальной машины» выбрать «Нет»;
9. в разделе «Доступно» выбрать необходимый снимок виртуальной машины, нажав «стрелку вверх» Скриншот 04-08-2022 164153, например: wg-1;
10. перейти во вкладку «Тип инстанса»;
11. в разделе «Доступно» выбрать необходимый снимок виртуальной машины, нажав «стрелку вверх» C:\Users\a.zakirov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Скриншот 04-08-2022 164153.jpg, например: 4cpu4g;
12. перейти во вкладку «Сети»;
13. в разделе «Доступно» выбрать необходимую сеть, , нажав «стрелку вверх» C:\Users\a.zakirov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Скриншот 04-08-2022 164153.jpg, например: vlan18;
14. нажать «Запустить виртуальную машину»;
15. повторить пункты 2-14 для виртуальных машин wg-2 и wg-3.

## Создание виртуальной машины из снимка виртуальной машины

Для работы программного обеспечения WebGard 2.0 требуется установить 3 виртуальные машины. Для этого необходимо:

1. войти в web интерфейс AccentOS;
2. перейти в раздел: Проект -> Виртуальные машины, нажать «Запустить машину»;
3. перейти во вкладку «Детали»;
4. в поле «Имя виртуальной машины» ввести название для виртуальных машин - wg;
5. в пункте «Количество» ввести необходимое кол-во виртуальных машин - 3;
6. перейти во вкладку «Источник»;
7. в поле «Выберите источник загрузки» выбрать «Снимок виртуальной машины»;
8. в поле «Создать новый диск» выбрать «Нет»;
9. в разделе «Доступно» выбрать необходимый снимок виртуальной машины, нажав «стрелку вверх» Скриншот 04-08-2022 164153, например: wg-script-220720;
10. перейти во вкладку «Тип инстанса»;
11. в разделе «Доступно» выбрать необходимый снимок виртуальной машины, нажав «стрелку вверх» C:\Users\a.zakirov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Скриншот 04-08-2022 164153.jpg, например: 4cpu4g;
12. перейти во вкладку «Сети»;
13. в разделе «Доступно» выбрать необходимую сеть, , нажав «стрелку вверх» C:\Users\a.zakirov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Скриншот 04-08-2022 164153.jpg, например: provider;
14. Нажать «Запустить виртуальную машину».

Примечание: если при нажатии кнопки «Запустить виртуальную машину» появляется ошибка, тогда необходимо в пункте 8 в поле «Создать новый диск» указать «Да», в поле «Удалить диск при удалении виртуальной машины» указать «Да», после чего в поле «Размер диска (ГБ)» указать 16.

## Установка WebGard 2.0 В среде AccentOS

## Работа с сетевыми настройками в интерфейсе AccentOS

1. перейти в раздел: Проект -> Виртуальные машины;
2. нажать на «стрелку вниз», справа от виртуальной машины Скриншот 04-08-2022 173049, выбрать «Отсоединить интерфейс»;
3. в открывшемся окне в поле «Порт» выбрать доступный ip-адрес и нажать кнопку «Отсоединить интерфейс»;
4. повторить пункты 2-3 для каждой виртуальной машины WebGard 2.0;
5. нажать на «стрелку вниз», справа от виртуальной машины C:\Users\a.zakirov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Скриншот 04-08-2022 173049.jpg, выбрать «Подключить интерфейс»;
6. в открывшемся окне в поле «Сеть» выбрать сеть (например: vlan18), которую выбирали во время создания виртуальной машины, ввести необходимый ip-адрес, нажать кнопку «Подключить интерфейс»;
7. повторить для каждой виртуальной машины WebGard 2.0;
8. перезагрузить виртуальные машины;
9. повторить пункты 5-8 для второй подсети (если есть);
10. перейти в раздел Сетевые ресурсы -> Сети
11. выбираем сеть, которую выбрали в пункте 6;
12. выбираем вкладку «Порты»;
13. выбираем виртуальную машину, на которой установлен WebGard 2.0 (не кворум), справа от нее нажимаем «Редактировать порт»;
14. открываем вкладку «Группы безопасности», убираем все элементы из списка «Группы безопасности порта»;
15. возвращаемся на вкладку «Информация», убираем галочку с пункта «Безопасность порта»;
16. нажимаем кнопку «Обновление»;
17. повторяем пункты 10-16 для второй подсети;
18. повторить пункты 10-17 для второй виртуальной машины (не кворум).
19. перезагрузить виртуальные машины.

## Настройка второго сетевого интерфейса в виртуальной машине

Для подключения используются пользователь user (логин: user, пароль:user). Пароль рут пользователь: root.

1. зайти с помощью консоли на первую виртуальную машину (или через интерфейс консоли AccentOS или с помощью другого средства удаленного подключения, например, putty через сеть vlan18);
2. ввести логин и пароль пользователя, зайти через root с помощью команды

su -

1. узнать имя второго интерфейса сети:

ip a

1. необходимо проверить есть ли файл второго сетевого интерфейса в папке /etc/net/ifaces/[имя интерфейса сети]

если нет, создать командой mkdir, пример:

mkdir eth1

1. открыть (или создать) файл опций сетевого интерфейса options:

vim /etc/net/ifaces/[имя интерфейса сети]/options

изменить или добавить строки в открытом файле:

BOOTPROTO=static

TYPE=eth

CONFIG\_WIRELESS=no

CONFIG\_IPV4=yes

1. сохранить и выйти из файла, нажав клавишу «esc» и набрав «:wq!», нажать клавишу «Enter»;
2. для настройки ip адреса виртуальной машины на втором сетевом интерфейсе необходимо, открыть (или создать) файл ipv4address:

vim /etc/net/ifaces/[имя интерфейса сети]/ipv4address

1. указать ip-адрес в CIDR-нотации:

Пример:

10.40.17.211/24

1. сохранить и выйти из файла, нажав клавишу «esc» и набрав «:wq!», нажать клавишу «Enter»;
2. для указания маршрутизации на втором сетевом интерфейсе необходимо открыть (или создать) файл ipv4route:

vim /etc/net/ifaces/[имя интерфейса сети]/ipv4route

1. проверить, что по умолчанию установлен ip-адрес:

для второй подсети:

сеть/маска via ip-адрес шлюза

Пример:

10.40.17.0/24 via 10.40.17.254

1. сохранить и выйти из файла, нажав клавишу «esc» и набрав «:wq!», нажать клавишу «Enter»;
2. перезагрузить ОС:

reboot

1. повторить пункты 1-13 для второй и третей виртуальных машин;

## Настройка DNS

1. чтобы узнать адреса DNS серверов необходимо посмотреть их в файле /etc/resolv.conf
2. для указания адресов DNS открыть (или создать) файл resolv.conf:

vim /etc/net/ifaces/[имя интерфейса сети]/resolv.conf

1. проверить DNS-серверы организации (если есть):

nameserver [DNS-сервер организации]

Пример:

nameserver 10.40.129.42

nameserver 10.40.116.2

1. сохранить и выйти из файла, нажав клавишу «esc» и набрав «:wq!», нажать клавишу «Enter»;
2. перезагрузить ОС:

reboot

1. повторить пункты 1-5 для второй и третей виртуальных машин;

## Проверка доступности с помощью скрипта conn\_checker.sh

1. после перезагрузки, необходимо проверить доступность виртуальных машин между собой, с ipa и с ms ad, для этого вводим команду:

vim /home/user/conn\_checker.sh

если файла нет, содержимое скрипта находится в Приложении 2;

1. в открывшемся файле прописываем ip-адреса 1, 2, и 3 виртуальных машин, ms ad, ipa, сохраняем;
2. сохранить и выйти из файла, нажав клавишу «esc» и набрав «:wq!», нажать клавишу «Enter»;
3. запускаем скрипт командой

bash /home/user/conn\_checker.sh

1. если в выводе, справа от ip-адреса узла написано «ok» - доступ есть, если «failed» - доступа нет, пример можно увидеть на рисунке (Рис. 2).

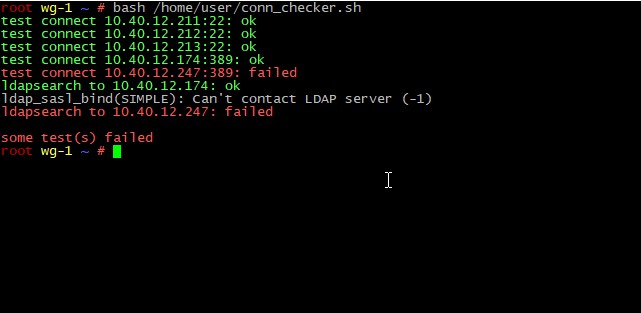


Рис. 2 Проверка доступности

## Настройка скрипта для создания бэкапов

1. необходимо отредактировать скрипт etcd\_cluster.sh, для этого вводи команду

vim /usr/local/bin/etcd\_cluster.sh

если файл отсутствует, его содержимое можно найти в Приложении 2.

1. в строках, которые начинаются с ssh, необходимо поменять grep 9\:2380 и http://10.248.144.9:2380 на ip-адрес виртуальной машины WebGard 2.0, например

grep [последняя цифра ip-адреса текущей виртуальной машины]\:2380

и

http://[ip-адрес текущей виртуальной машины]:2380

1. в строках, которые начинаются с ssh, необходимо поменять wg2 на название другой виртуальной машины WebGard 2.0, например

* для wg1 – меняем на wg2;
* для wg2 – меняем на wg3;
* для wg3 – меняем на wg1;

Пример для wg1:

ssh wg2 -o StrictHostKeyChecking=no "ETCDCTL\_API=3 etcdctl member remove \`ETCDCTL\_API=3 etcdctl member list | grep 211\:2380 | sed 's|,.\*||g'\`"

ssh wg2 -o StrictHostKeyChecking=no "ETCDCTL\_API=3 etcdctl member add --peer-urls='http://10.40.18.211:2380' etcd1"

1. повторить пункты 1-4 для всех трех виртуальных машин WebGard 2.0.

## Настройка и запуск конфигурации виртуальных машин

Для подготовки к конфигурации виртуальных машин необходимо на первой виртуальной машине:

1. ввести команду:

vim /root/.wg\_init.conf

1. в открывшемся файле необходимо ввести ip-адреса в соответствии с таблицей (Таблица 2);

Таблица 2 Таблица ip-адресов

|  |  |
| --- | --- |
| **Название** | **Описание** |
| wg1ipa\_ip | ip-адрес первой виртуальной машины, связанной с ipa, на которую будут заходить с ОС Linux (первичный адрес сервера для rs client) |
| wg1ad\_ip | ip-адрес первой виртуальной машины, связанной с ms ad, на которую будут заходить с ОС Windows (первичный адрес сервера для rs client) |
| wg2ipa\_ip | ip-адрес второй виртуальной машины, связанной с ipa, на которую будут заходить с ОС Linux (вторичный адрес сервера для rs client) |
| wg2ad\_ip | ip-адрес второй виртуальной машины, связанной с ms ad, на которую будут заходить с ОС Windows (вторичный адрес сервера для rs client) |
| wg3\_ip | ip-адрес третьей виртуальной машины, кворум |
| ipa\_ip | ip-адрес виртуальной машины с ipa |
| ad\_ip | ip-адрес виртуальной машины c ms ad |
| rs1\_ip | ip-адрес домена облака AccentOS, на котором находятся виртуальные машины |

Все ip адреса, указанные в таблице, должны быть различными. Если вы не знаете, какой указать ip-адрес, можно указать любой свободный. Пример:

* wg1ipa\_ip – ip адрес первой виртуальной машины;
* wg1ad\_ip – свободный ip-адрес;
* wg2ipa\_ip – ip адрес второй виртуальной машины;
* wg2ad\_ip – свободный ip-адрес;
* wg3\_ip - ip адрес третьей виртуальной машины.

При указании вторичного ip-адреса необходимо указать адрес из той же подсети, что первичная (например, vlan18). Пример заполнения на рисунке (Рис. 3).

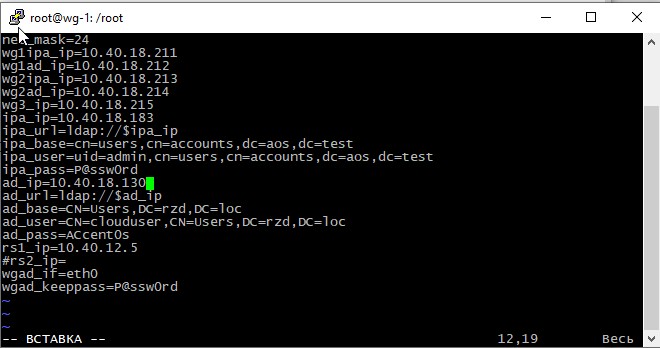


Рис. 3 Пример заполненния wg\_init.conf

1. сохранить и выйти из файла, нажав клавишу «esc» и набрав «:wq!», нажать клавишу «Enter»;
2. запустить скрипт конфигурации виртуальных машин командой:

bash /home/user/configure\_wg.sh

Примечание: после запуска, скрипт начинает настройку всех трех виртуальных машин. Изменения сильно влияют на все 3 виртуальные машины, поэтому стоит перепроверить правильность настройки виртуальных машин (в особенности файл wg\_init.conf). Для проверки доступности виртуальных машин друг до друга стоит повторить пункт 6.4.

1. дождаться завершения работы скрипта.

## Проверка работы кластера виртуальных машин

После завершения работы скрипта configure\_wg.sh, стоит проверить находятся ли виртуальные машины в кластере. Для этого необходимо выполнить команду:

ETCDCTL\_API=3 etcdctl endpoint status --cluster -w table

все 3 виртуальные машины должны находится в кластере.

## Подключение к подсистеме фильтрации и подсистеме администратора WebGard 2.0

## Вход в подсистему администрирования и предварительные настройки

Подсистема администрирования будет доступна по адресу: http[://wg](http://localhost:8080/security-manager/login.htm)\_ip[:8080/security-](http://localhost:8080/security-manager/login.htm)manager/login[.](http://localhost:8080/security-manager/login.htm)htm (wg\_ip – ip-адрес первой или второй виртуальной машины). Необходимо перейти на данную ссылку и ввести логин – пароль администратора (по умолчанию логин admin, пароль admin).

Перейти во вкладку «Администраторы», выделить строку с администратором «admin», нажать кнопку «Редактировать», изменить пароль на более сложный (пароль должен состоять минимум из 8 символов и содержать 1 букву верхнего регистра, 1 букву нижнего регистра и 2 цифры), добавить существующую почту администратора.

## Вход в подсистему фильтрации

Подсистема фильтрации будет доступна по адресу: http[://wg](http://localhost:8080/security-manager/login.htm)\_ip[:80](http://localhost:8080/security-manager/login.htm) (wg\_ip – ip-адрес первой или второй виртуальной машины) (Рис. 4). Необходимо будет ввести домен, логин, парол пользователя (например rgdipa vdi\_user1 P@ssw0rd).



Рис. 4 Адрес посистемы фильтрации

## Вход в подсистему фильтрации с помощью rs-client AccentOS

Для входа в систему через rs client AccentOS необходимо выполнить следующие действия:

1. открыть приложение rs-client;
2. ввести значения в поля «Логин», «Пароль», «Проект»;
3. нажать подключиться;
4. после подключения откроется новая рабочая сессия, в ней открыть приложение «Веб-браузер Firefox»;
5. войти в подсистему фильтрации, как описано в разделе 7.1.

При указании первичного и вторичного ip адресов в настройках необходимо следовать таблице ip адресов (Таблица 2). При указании адресов, также не забудьте указать правильный домен.

## Организация работы программы WebGard 2.0 в режиме высокой доступности

## Проверка кластера виртуальных машин

Для подготовки к созданию резервной копии виртуальной машины необходимо:

1. проверить, что виртуальные машины находятся в одном кластере, для этого необходимо выполнить следующую команду:

ETCDCTL\_API=3 etcdctl endpoint status --cluster -w table

все 3 виртуальные машины должны находится в кластере;

1. необходимо проверить и отредактировать файл конфигураций etcd:

vim /etc/etcd/etcd.conf

1. в открывшемся файле необходимо найти строку ETCD\_INITIAL\_CLUSTER\_STATE в ней необходимо поменять «existing» на «new», пример:

ETCD\_INITIAL\_CLUSTER\_STATE="new"

1. сохранить и выйти из файла, нажав клавишу «esc» и набрав «:wq!», нажать клавишу «Enter»;
2. повторить пункты 2 – 4 для каждой виртуальной машины.

## Создание снимка виртуальной машины с помощью интерфейса AccentOS

Для создания снимка виртуальной машины необходимо:

1. выделяем все три виртуальные машины, проставляя галочки слева от их названия;
2. нажимаем кнопку «Действие», выбираем «Выключить машины»;
3. справа от виртуальной машины выбираем кнопку со стрелкой вниз Скриншот 04-08-2022 173049, выбираем «Создать снимок»;
4. в открывшемся окне в поле «Имя снимка» пишем название создаваемого снимка виртуальной машины;
5. нажимаем кнопку «Создать снимок»;
6. повторяем пункты 3 – 5 для каждой виртуальной машины.

## Восстановление виртуальной машины из снимка виртуальной машины

Для восстановления виртуальной машины из снимка виртуальной машины:

1. справа от виртуальной машины выбираем кнопку со стрелкой вниз Скриншот 04-08-2022 173049, выбираем «Выключить машину»;
2. когда машина выключилась, справа от виртуальной машины выбираем кнопку со стрелкой вниз Скриншот 04-08-2022 173049, выбираем «Перестроить машину»;
3. в открывшемся окне в поле «Выберите образ» выбираем созданный ранее снимок виртуальной машины;
4. нажимаем «Перестроить машину»;
5. после окончания процесса восстановления, запускаем виртуальную машину, выбрав справа от виртуальной машины кнопку со стрелкой вниз Скриншот 04-08-2022 173049, выбрать «Запустить машину»;
6. необходимо проверить настройки сетевых интерфейсов, для этого повторяем пункты из раздела 6.2;
7. необходимо проверить и отредактировать файл конфигураций etcd:

vim /etc/etcd/etcd.conf

1. в открывшемся файле найти строку ETCD\_INITIAL\_CLUSTER\_STATE в ней поменять «existing» на «new», пример:

ETCD\_INITIAL\_CLUSTER\_STATE="new"

1. сохранить и выйти из файла, нажав клавишу «esc» и набрав «:wq!», нажать клавишу «Enter»;
2. выполнить перезагрузку виртуальной машины командой:

reboot

## Дополнительное взаимодействие с системой WebGard 2.0

## Проверка кластера виртуальных машин

Для того, чтобы проверить находятся ли виртуальные машины в кластере, необходимо выполнить следующую команду:

ETCDCTL\_API=3 etcdctl endpoint status --cluster -w table

## Редактирование парольной политики

Для редактирования парольной политики пользователя и администратора необходимо:

1. Перейти в файл с помощью команды

vim /root/wgipa/etc/webguard/url.conf

1. в файле необходимо найти следующие строки:

pswd\_min\_len = 8

pswd\_max\_len = 100

pswd\_complexity = ^((?!.\*[\\s])(?=(.\*\\d){2,})(?=.\*[a-zа-яё])(?=.\*[A-ZА-ЯЁ]).{8,100})$

1. где:

* pswd\_min\_len – минимальная длина пароль;
* pswd\_max\_len – максимальная длина пароля;
* pswd\_complexity – регулярное выражение с ограничениями и требованиями на создание пароля.

## Редактирование политики логина

Для редактирования политики логина пользователя и администратора необходимо:

1. Перейти в файл с помощью команды

vim /root/wgipa/etc/webguard/url.conf

1. в файле необходимо найти следующие строки:

username\_min\_len = 3

username\_max\_len = 100

username\_pattern = ^[a-zA-Zа-яА-ЯёЁ0-9\_@»«&\\{\\}\\[\\]\\^#\\$\\.\\?%,\\!\\\*\\(\\)\\+\\=~:;'"\\<\\>\\-\\|\\/\\\\]{3,100}$

1. где:

* username\_min\_len – минимальная длина логина;
* username\_max\_len – максимальная длина логина;
* username\_pattern – регулярное выражение с ограничениями и требованиями на создание логина.

## Удаление администратора WebGard 2.0

Для удаления пользователя необходимо ввести следующую команду:

psql -h 127.0.0.1 -U securitymanager -d securitymanageripa -c "delete from web\_admin\_user where web\_admin\_user\_id='sometest'"

где необходимо указать в sometest логин администратора, которого необходимо удалить.

## Приложение 1

## Название скрипта: etcd\_cluster.sh

Расположение: /usr/local/bin/etcd\_cluster.sh

Содержание:

#!/bin/bash

if [[ $(cat /etc/etcd/etcd.conf | grep existing) ]]; then

exit 0

fi

systemctl stop etcd patroni

rm -rf /home/user/data/ /var/lib/etcd/default.etcd/

sed -i 's|new|existing|g' /etc/etcd/etcd.conf

ssh wg2 -o StrictHostKeyChecking=no "ETCDCTL\_API=3 etcdctl member remove \`ETCDCTL\_API=3 etcdctl member list | grep 9\:2380 | sed 's|,.\*||g'\`"

ssh wg2 -o StrictHostKeyChecking=no "ETCDCTL\_API=3 etcdctl member add --peer-urls='http://10.248.144.9:2380' etcd1"

systemctl start etcd

sleep 5

systemctl start patroni

sleep 60

systemctl restart haproxy

## Приложение 2

## Название скрипта: conn\_cheker.sh

Расположение: /home/user/conn\_cheker.sh

Содержание:

#!/bin/bash

wg1\_ip=10.40.12.73

wg2\_ip=10.40.12.244

wg3\_ip=10.40.12.54

ipa\_ip=10.40.12.174

ad\_ip=10.40.12.247

ipa\_port=389

ad\_port=389

ipa\_user=uid=admin,cn=users,cn=accounts,dc=aos,dc=test

ipa\_pass=P@ssw0rd

ad\_user=uid=admin,cn=users,cn=accounts,dc=aos,dc=test

ad\_pass=P@ssw0rd

ipa\_attr=uid

ad\_attr=sAMAccountName

RED='\033[1;31m'

GREEN='\033[1;32m'

NC='\033[0m'

test=ok

check\_port() {

if [[ $(nc -z $1 $2 </dev/null; echo $?) == \*0\* ]]; then

echo -e "${GREEN}test connect $1:$2: ok${NC}"

else

echo -e "${RED}test connect $1:$2: failed${NC}"

test=failed

fi

}

check\_ldap() {

if [[ -n $(ldapsearch -w $1 -H ldap://$2 -D "$3" -b "$3" $4 | grep $4\:) ]]; then

echo -e "${GREEN}ldapsearch to $2: ok${NC}"

else

echo -e "${RED}ldapsearch to $2: failed${NC}"

test=failed

fi

}

check\_port $wg1\_ip 22

check\_port $wg2\_ip 22

check\_port $wg3\_ip 22

check\_port $ipa\_ip $ipa\_port

check\_port $ad\_ip $ad\_port

check\_ldap $ipa\_pass $ipa\_ip $ipa\_user $ipa\_attr

check\_ldap $ad\_pass $ad\_ip $ad\_user $ad\_attr

echo

if [[ $test == "ok" ]]; then

echo -e "${GREEN}all tests ok${NC}"

else

echo -e "${RED}some test(s) failed${NC}"

fi

**Приложение 4** **Создание службы Samba для Microsoft AD**

Данный документ показывает порядок действий для формирования виртуальной инфраструктуры для сетевого хранения перемещаемых профилей и домашних папок пользователей Microsoft Active Directory 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

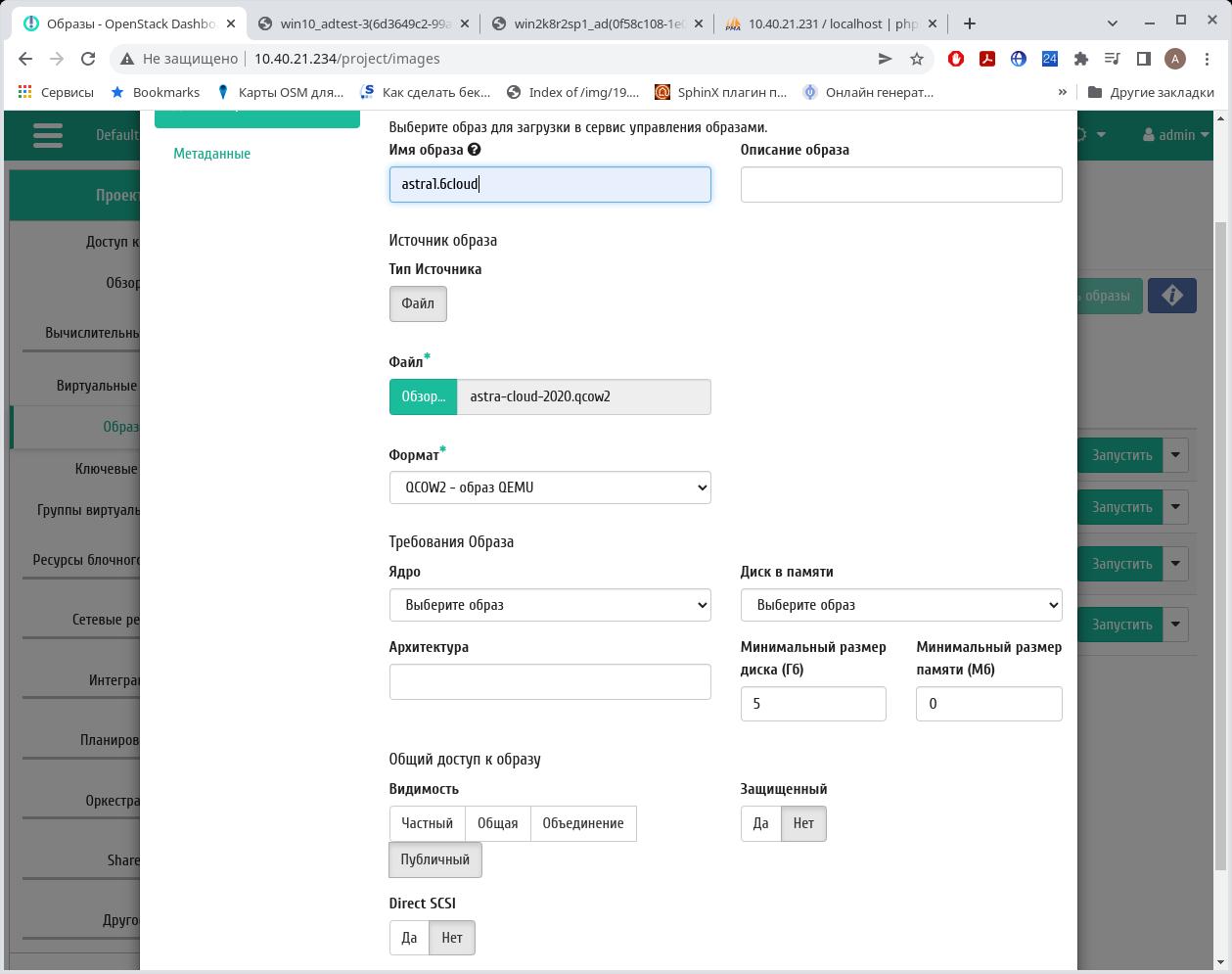
Любая виртуальная машина в облаке создается с использованием образа. Мы рассмотрим наиболее распространенный способ создания ВМ из образа диска установленной операционной системы с установленным пакетом cloud-init для возможности последующей донастройки операционной системы в облаке: увеличении диска, 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**.

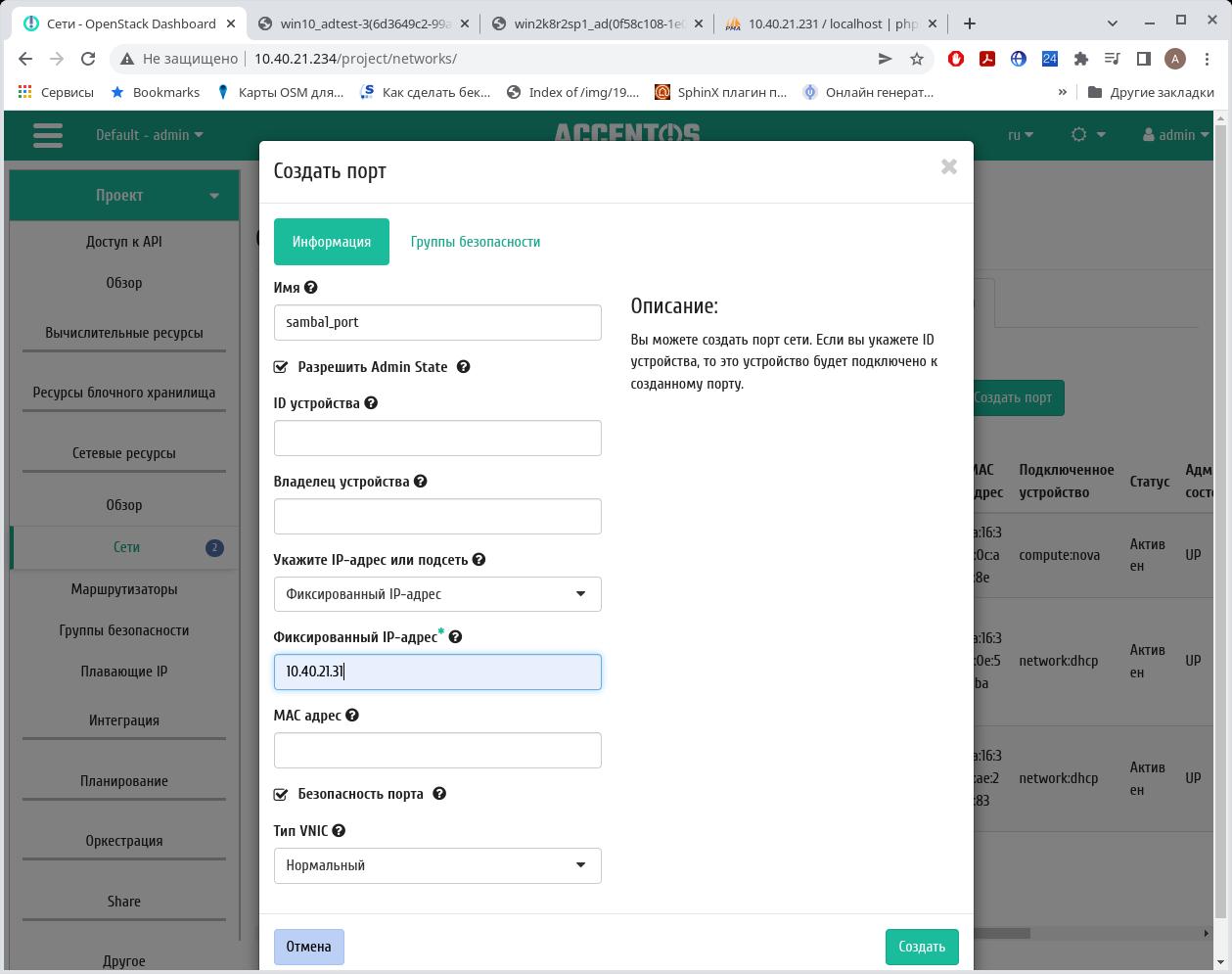
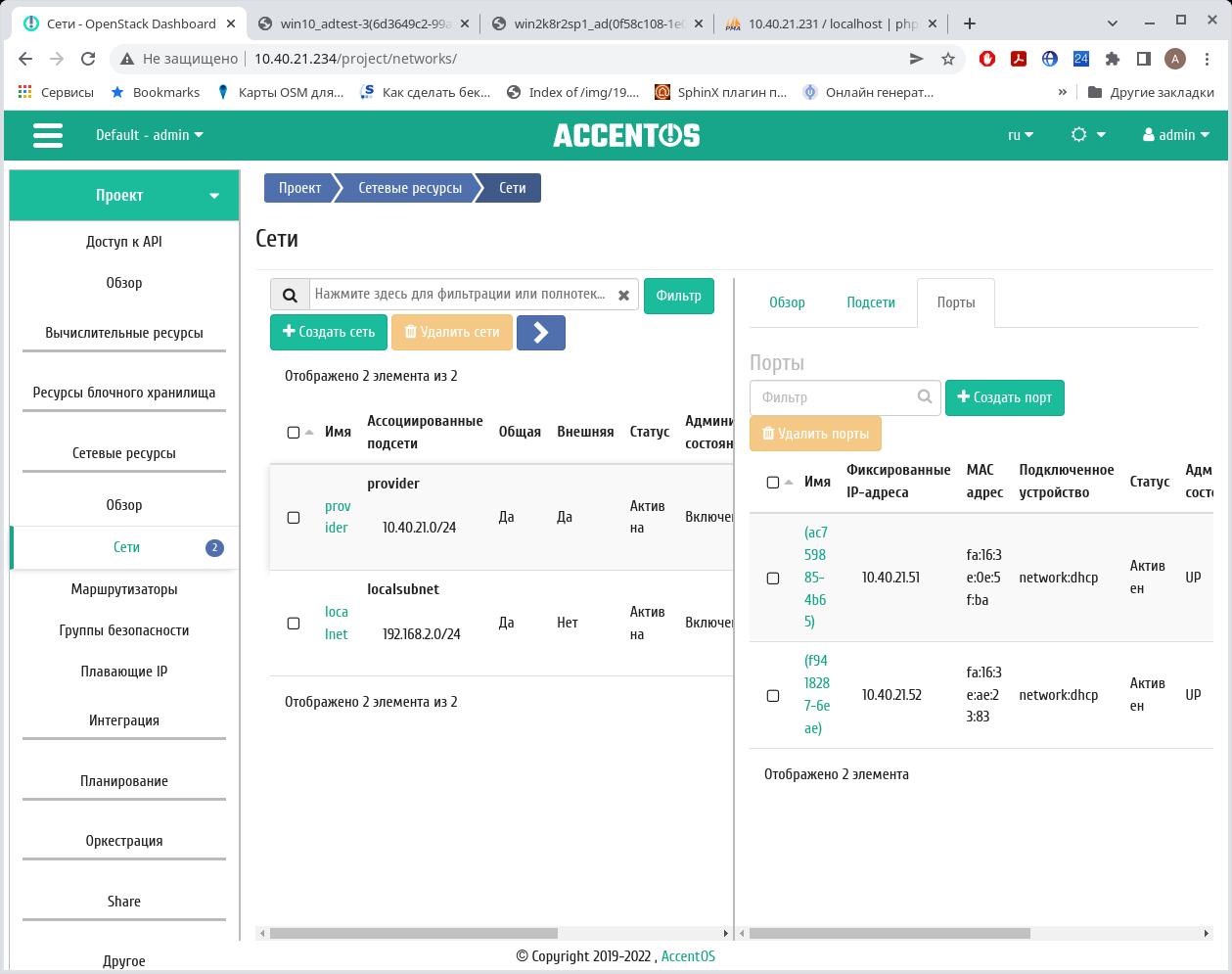


## 

## 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.

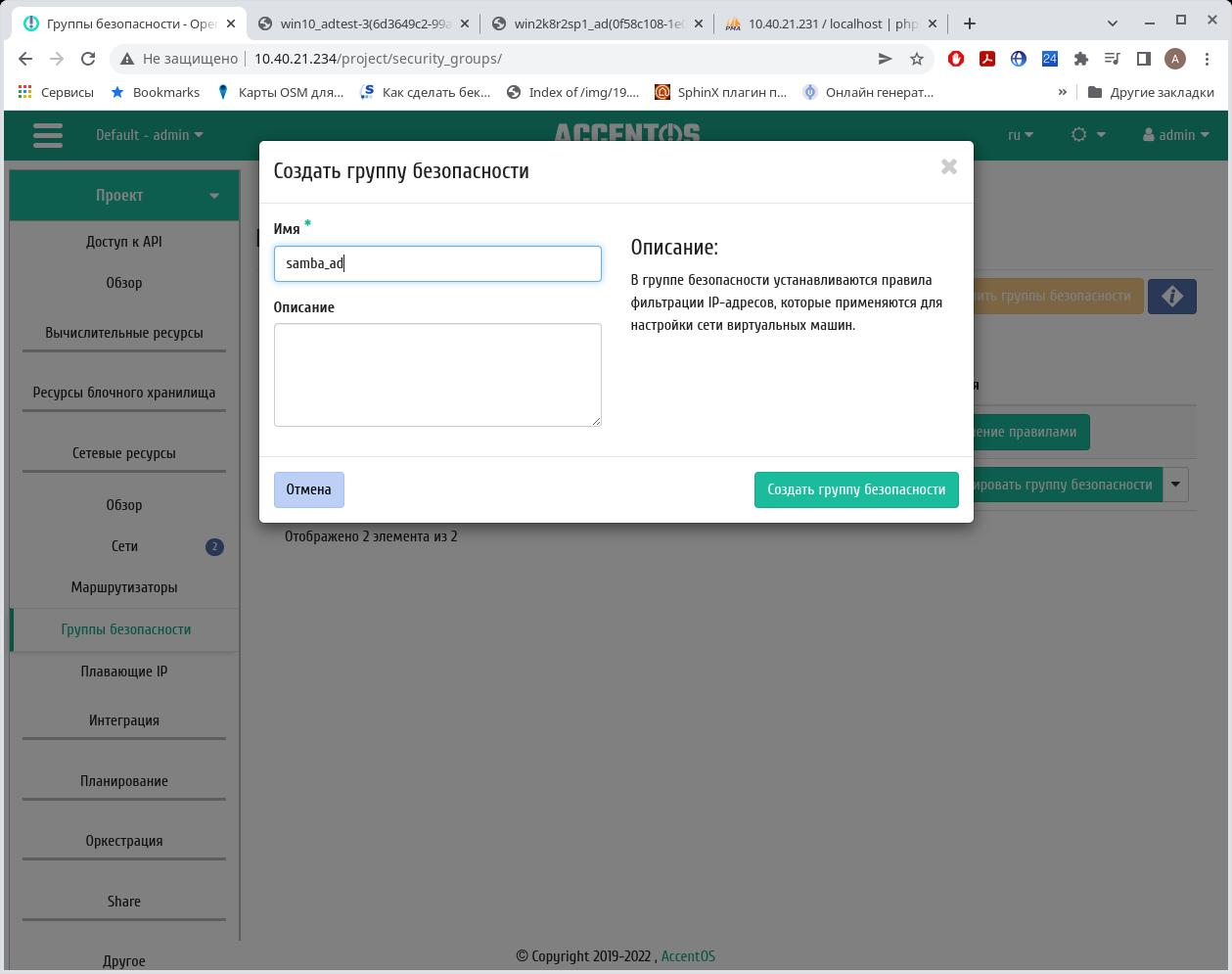


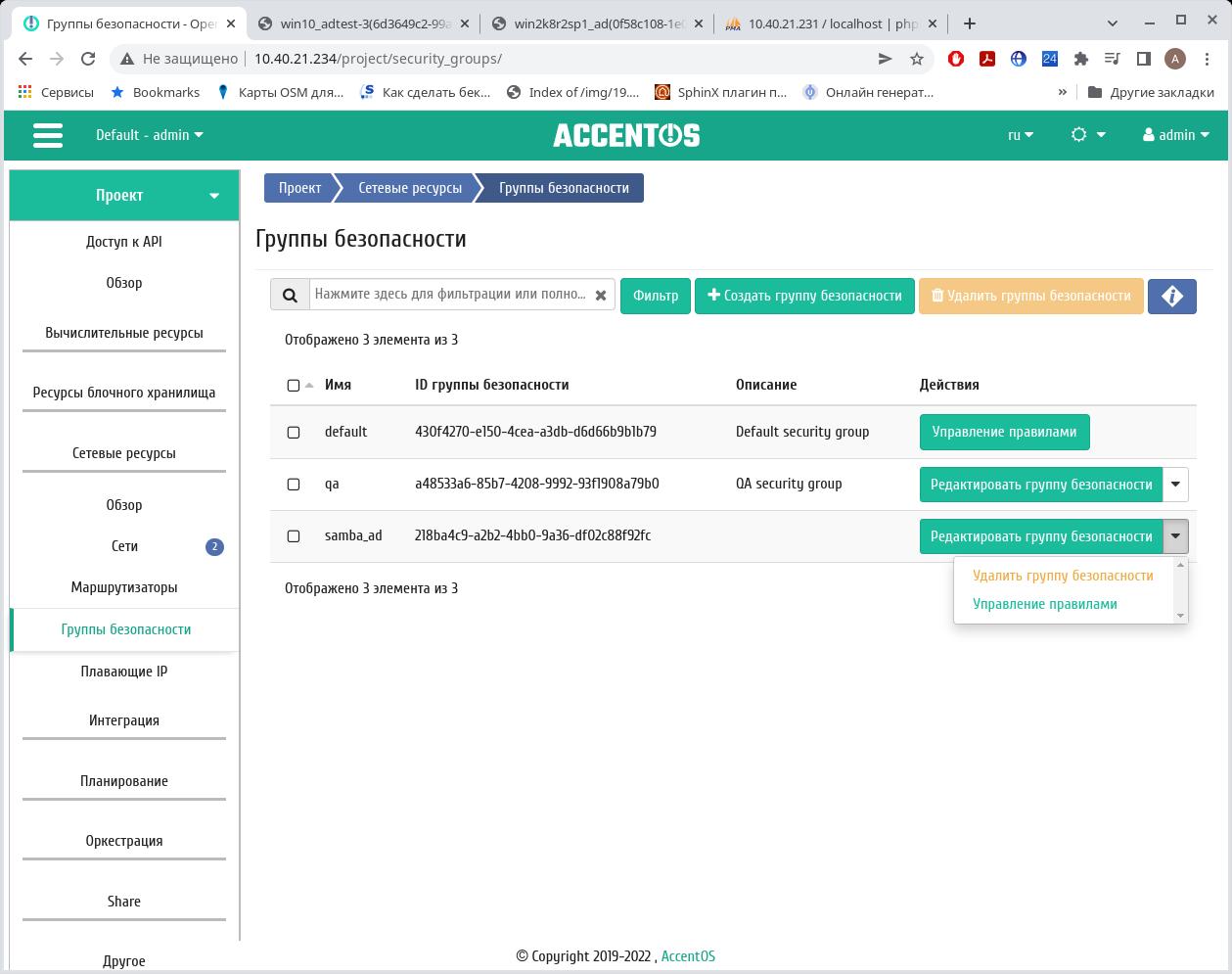
## Создание группы безопасности

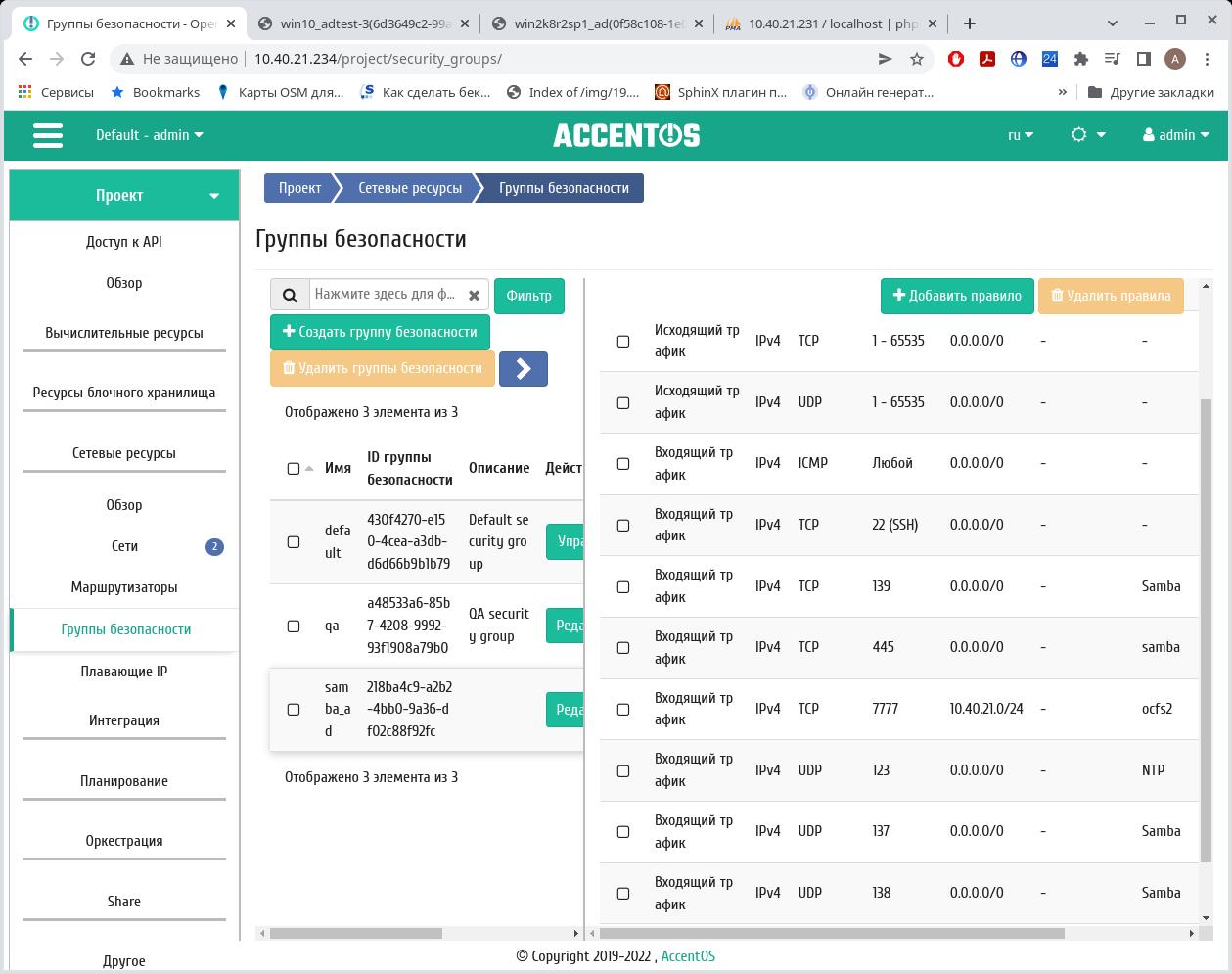
Сетевая безопасность на уровне облака обеспечивается не файрволом в каждой виртуальной машине, а правилами сетевых разрешений на уровне облака, объединенных в группы безопасности. Это позволяет одновременно менять сетевые правила для всех однотипных виртуальных машин без настройки каждой ВМ в отдельности.  
  
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:

1. 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 виртуальных машин.

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

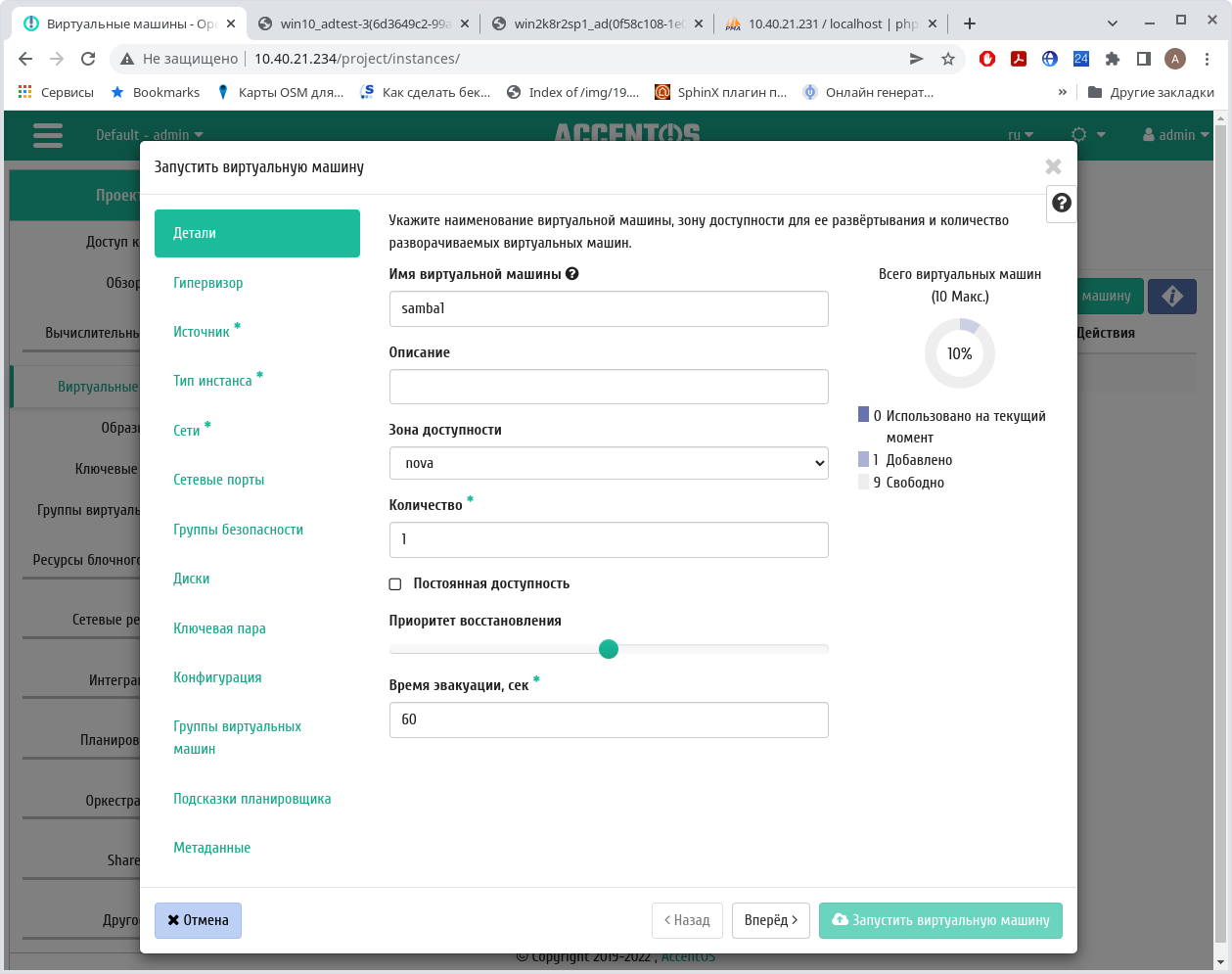
Для создания виртуальной машины или группы однотипных виртуальных машин необходимо заполнить несколько вкладок диалогового окна. Учитывая, что мы создаем виртуальные машины с предварительно созданными сетевыми портами - создавать каждую ВМ придется отдельно. Ниже представлены скриншоты пошагового заполнения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

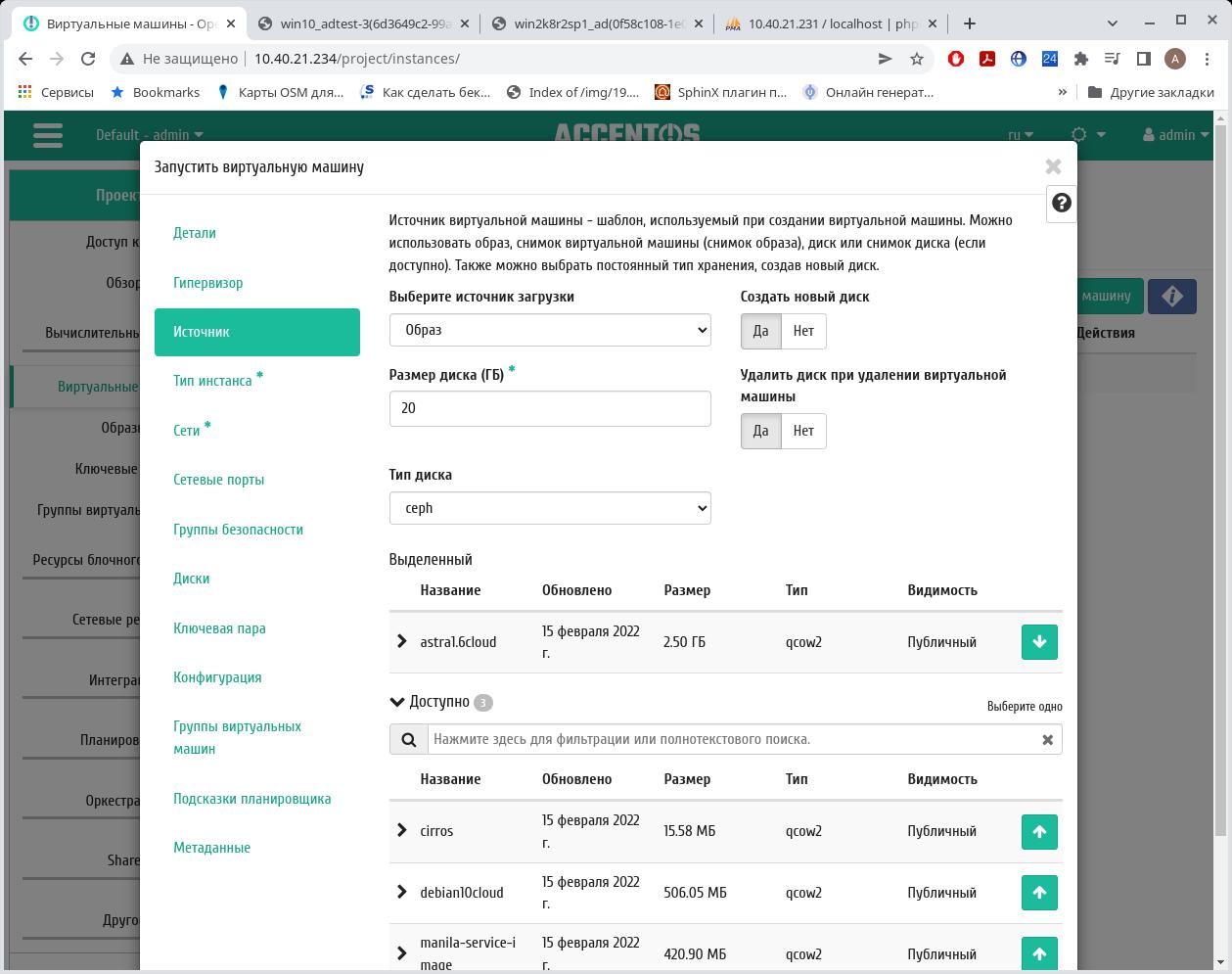
Системный диск виртуальной машины может быть быть создан как на любом типе cinder volume, так и без оного - на ephemeral диске nova. В нашем примере демонстрируется создание служебных ВМ на cinder volume.

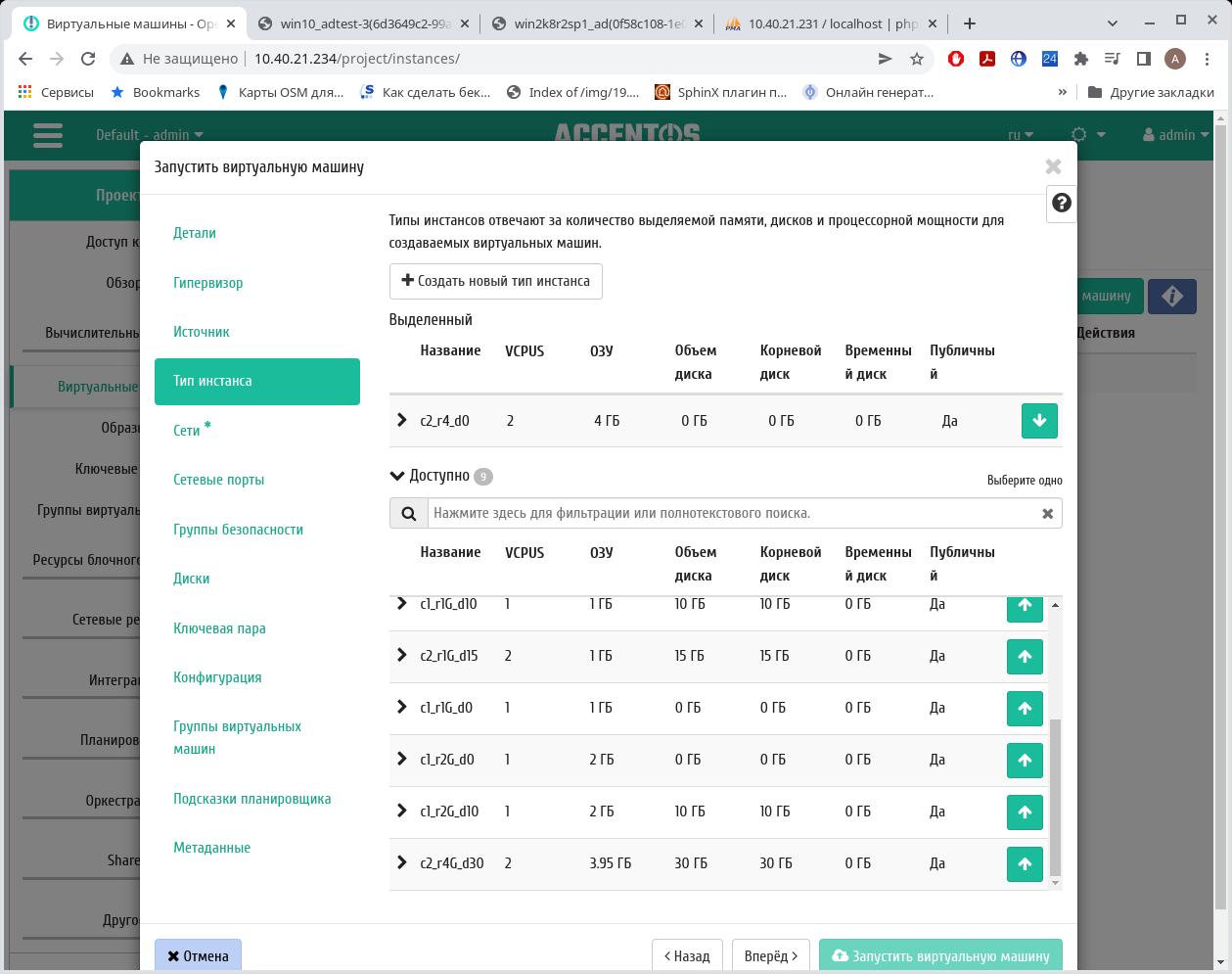
Проведенные тесты работы Samba показывают малую зависимость производительности Samba от выделенного объема оперативной памяти - главное, чтобы ее хватило для нормальной работы самой операционной системы. И наоборот - высокую зависимость производительности для большого числа пользователей от доступного 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.

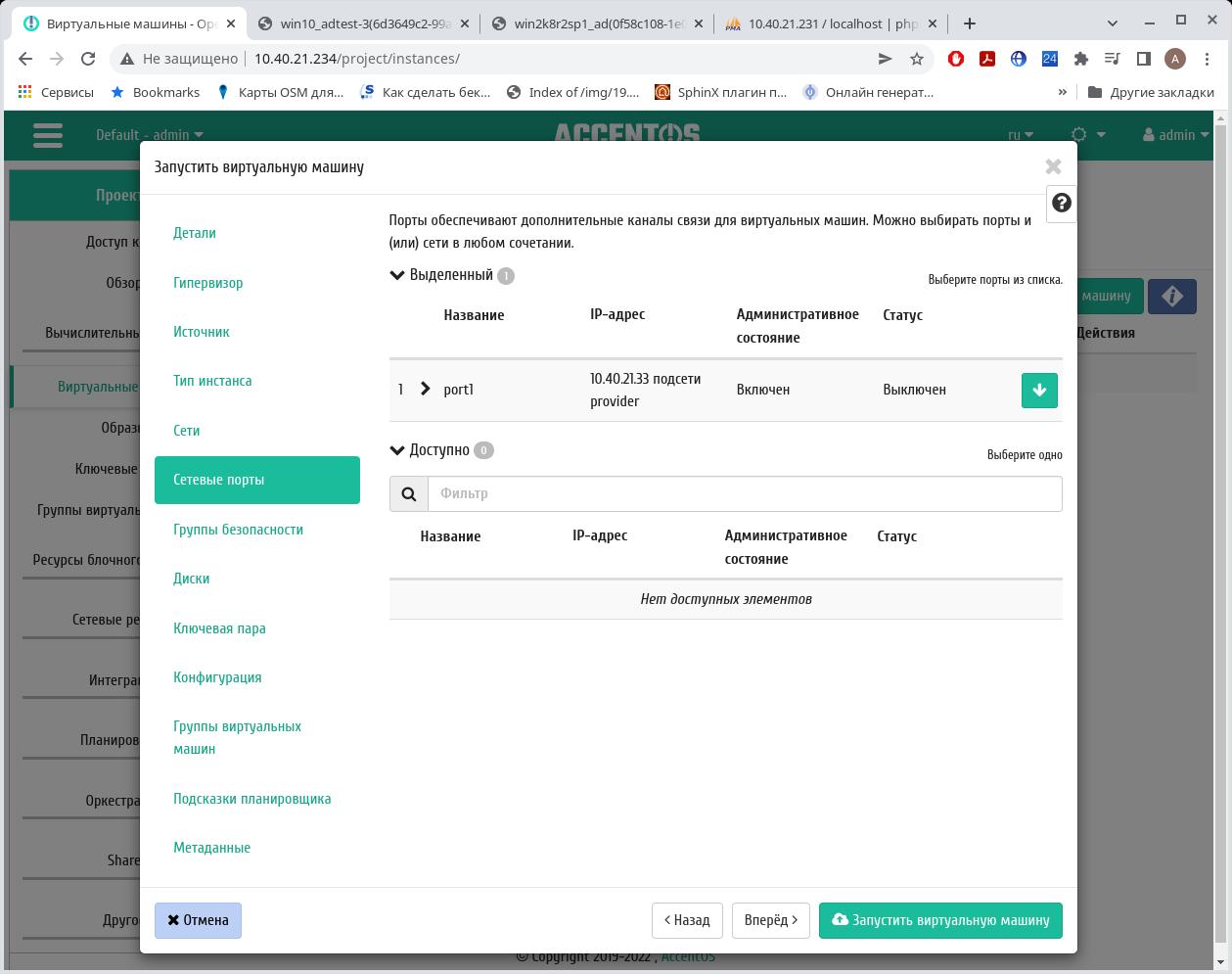
Вместо обычного выбора виртуальной сети, к которой будет подключена ВМ, мы выбираем предварительно созданный сетевой порт со статическим ip-адресом. В связи с чем в нашем примере отсутствует рассмотрение вкладки **Сети**.

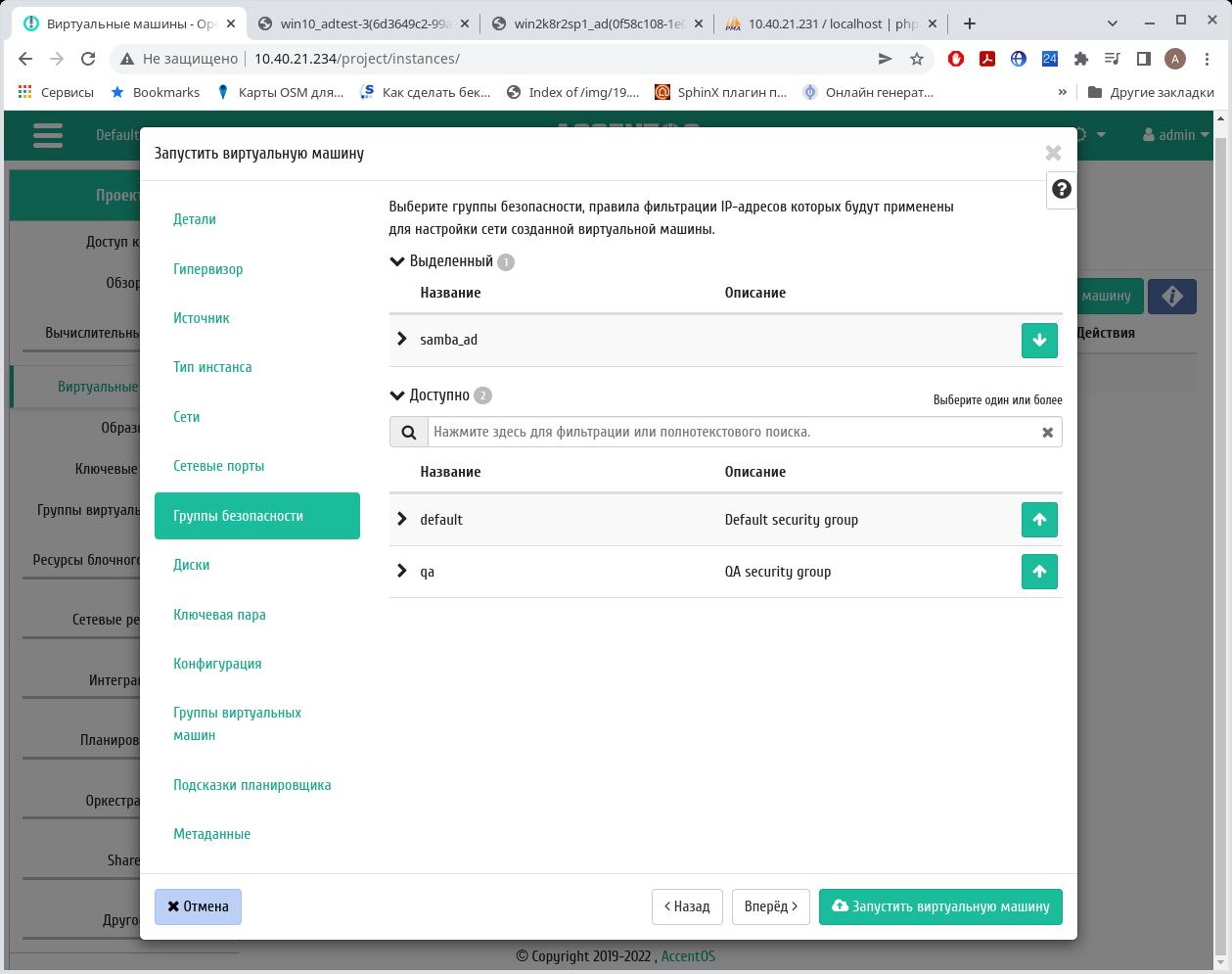
Мы также выбираем ключевую пару, от которой имеем секретный ключ, и ранее настроенную группу безопасности.

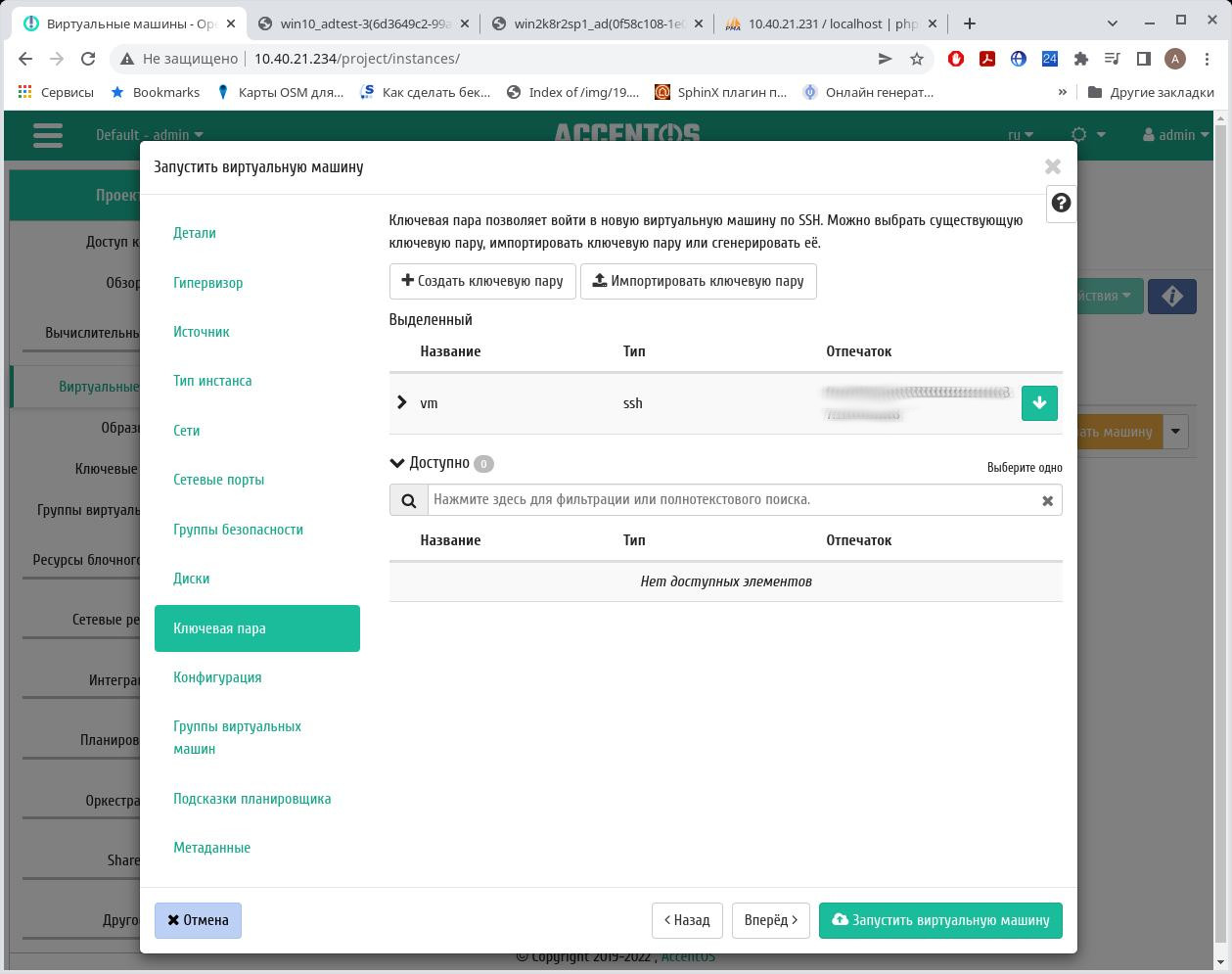


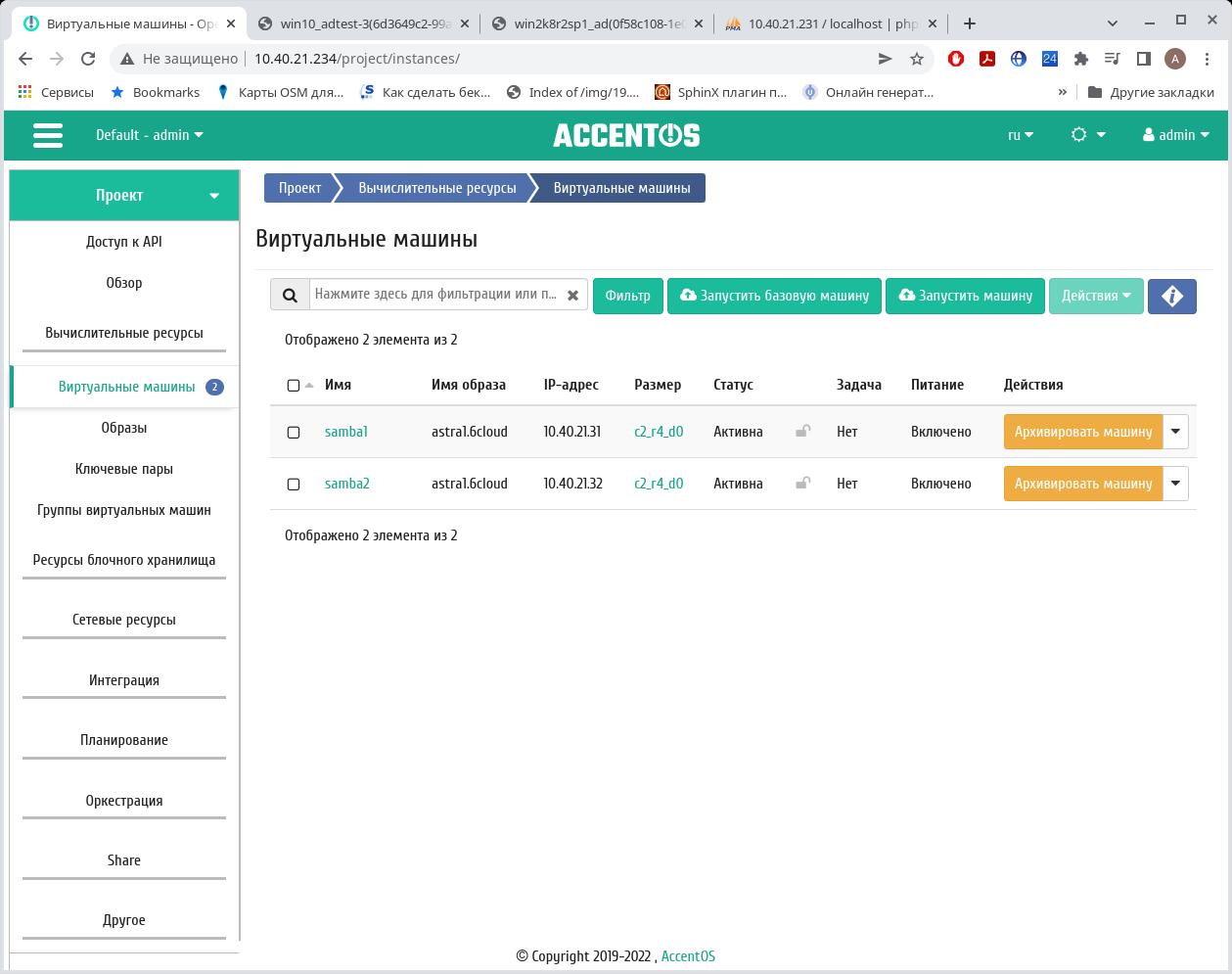








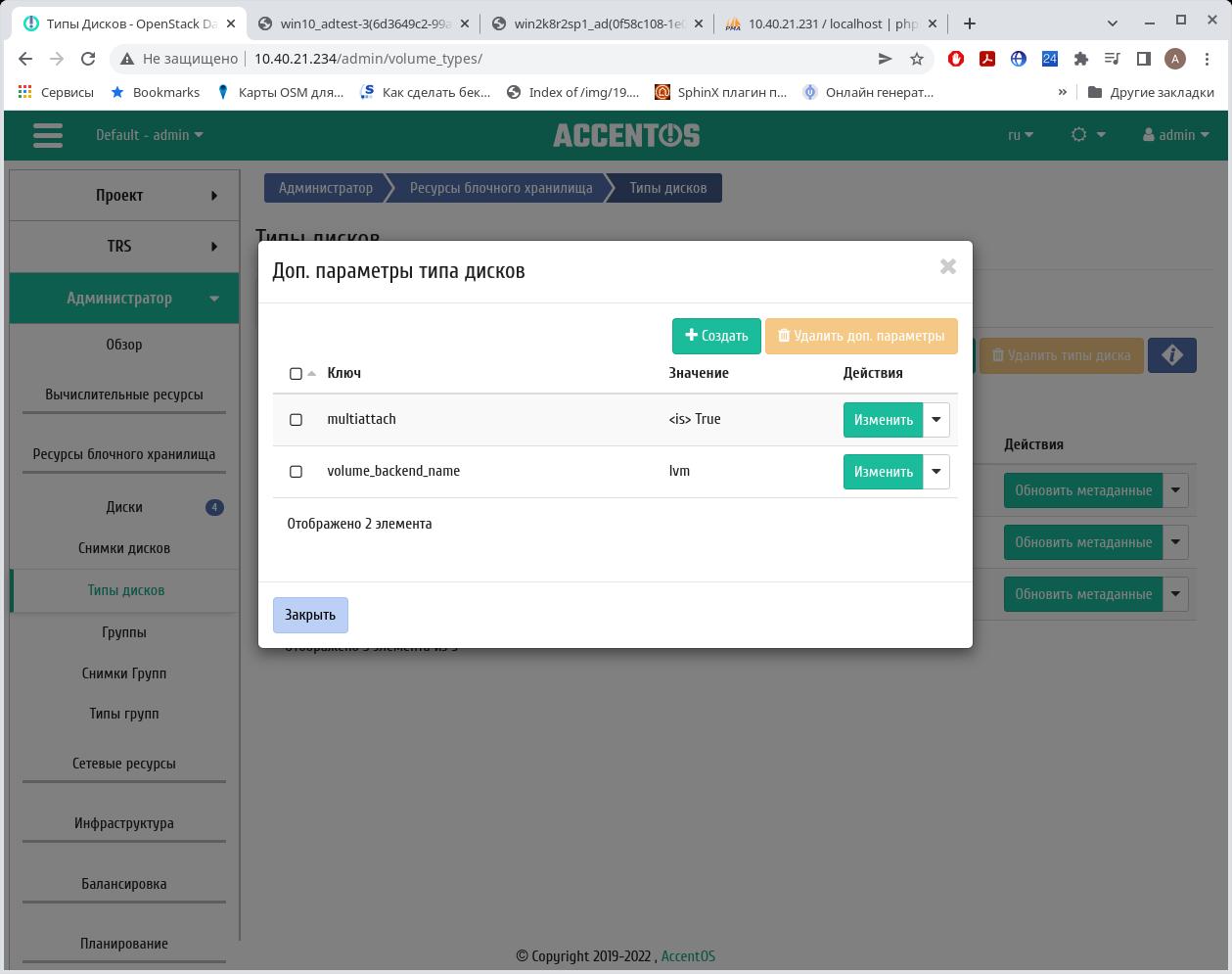


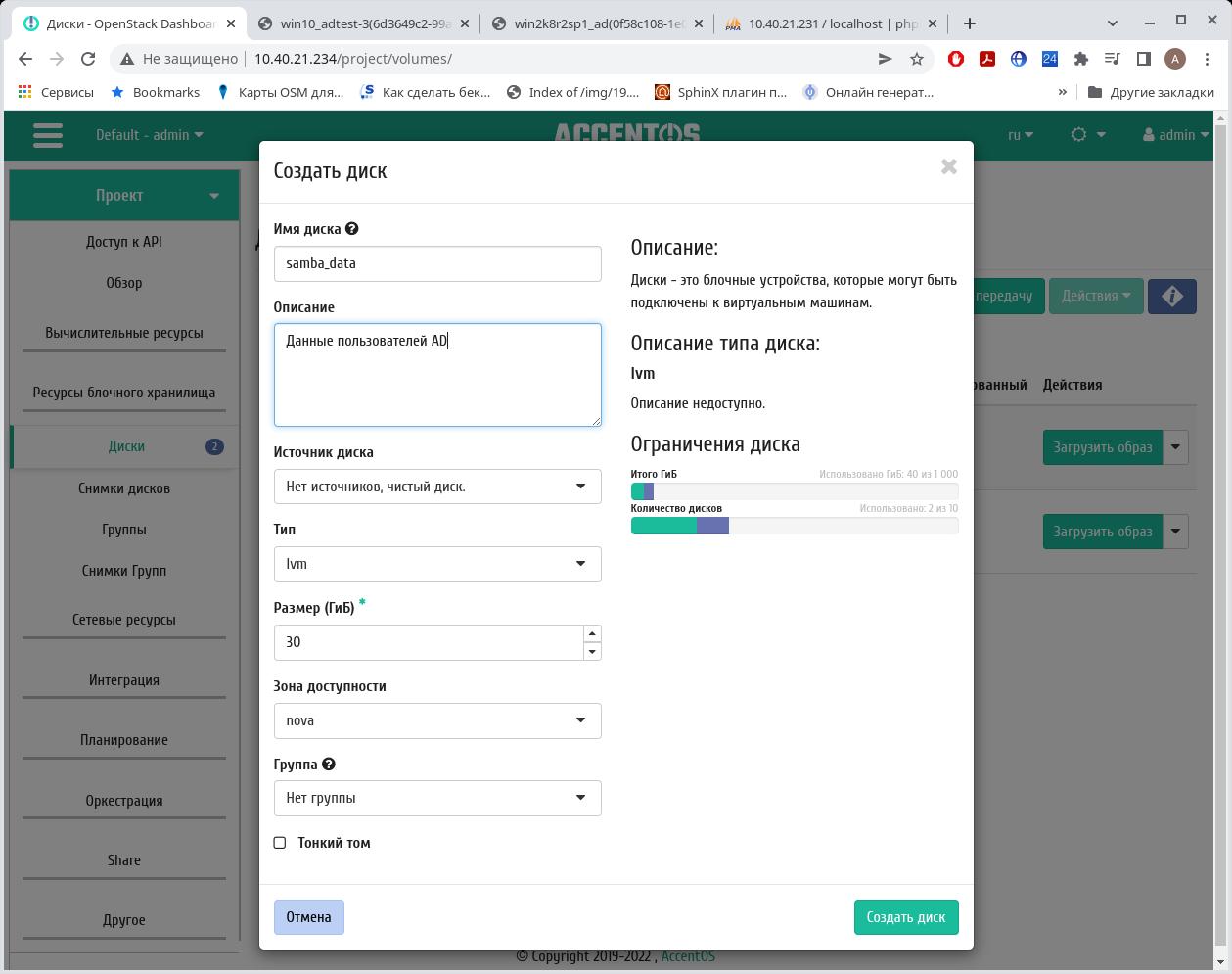


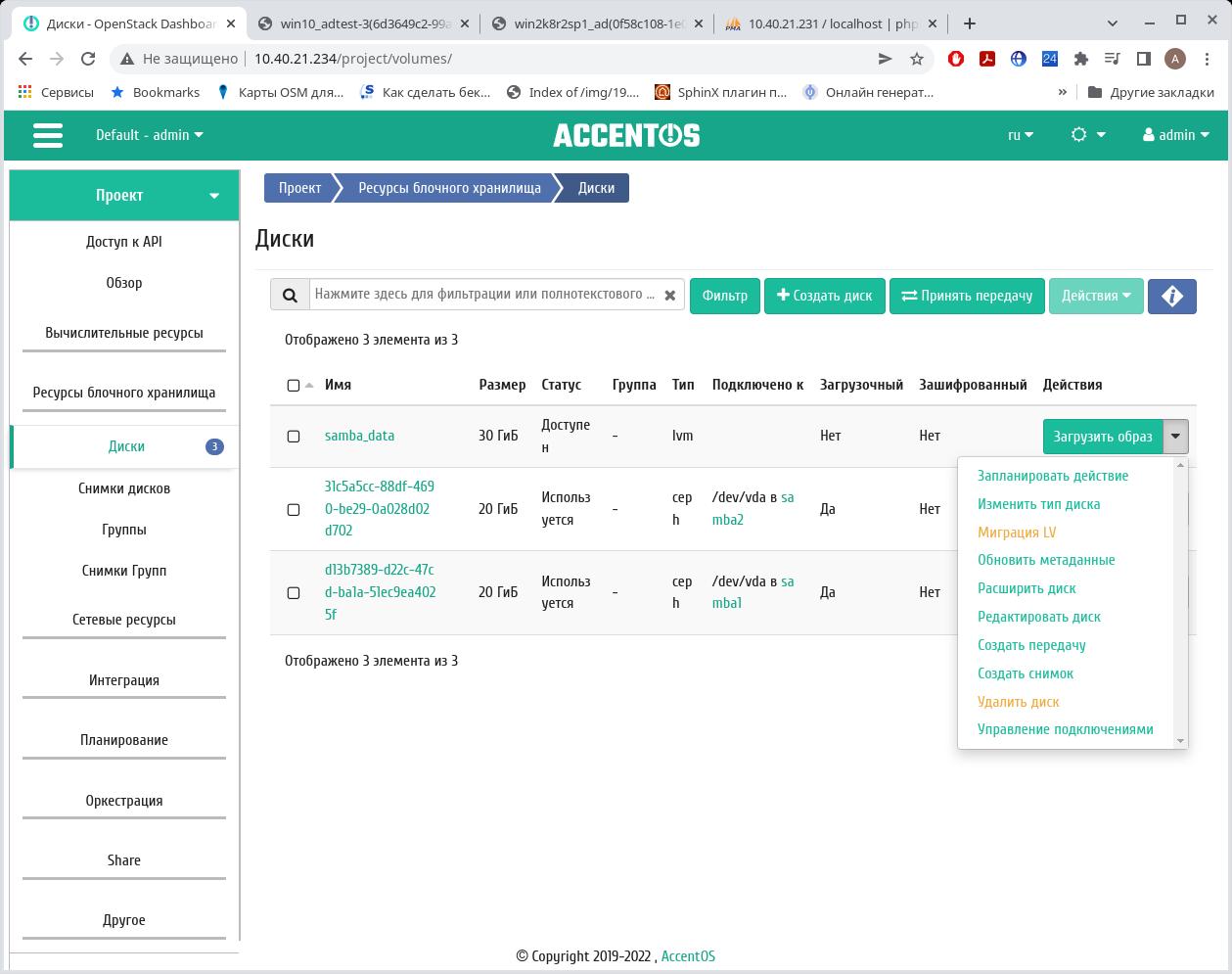
## Создание общего диска для Samba.

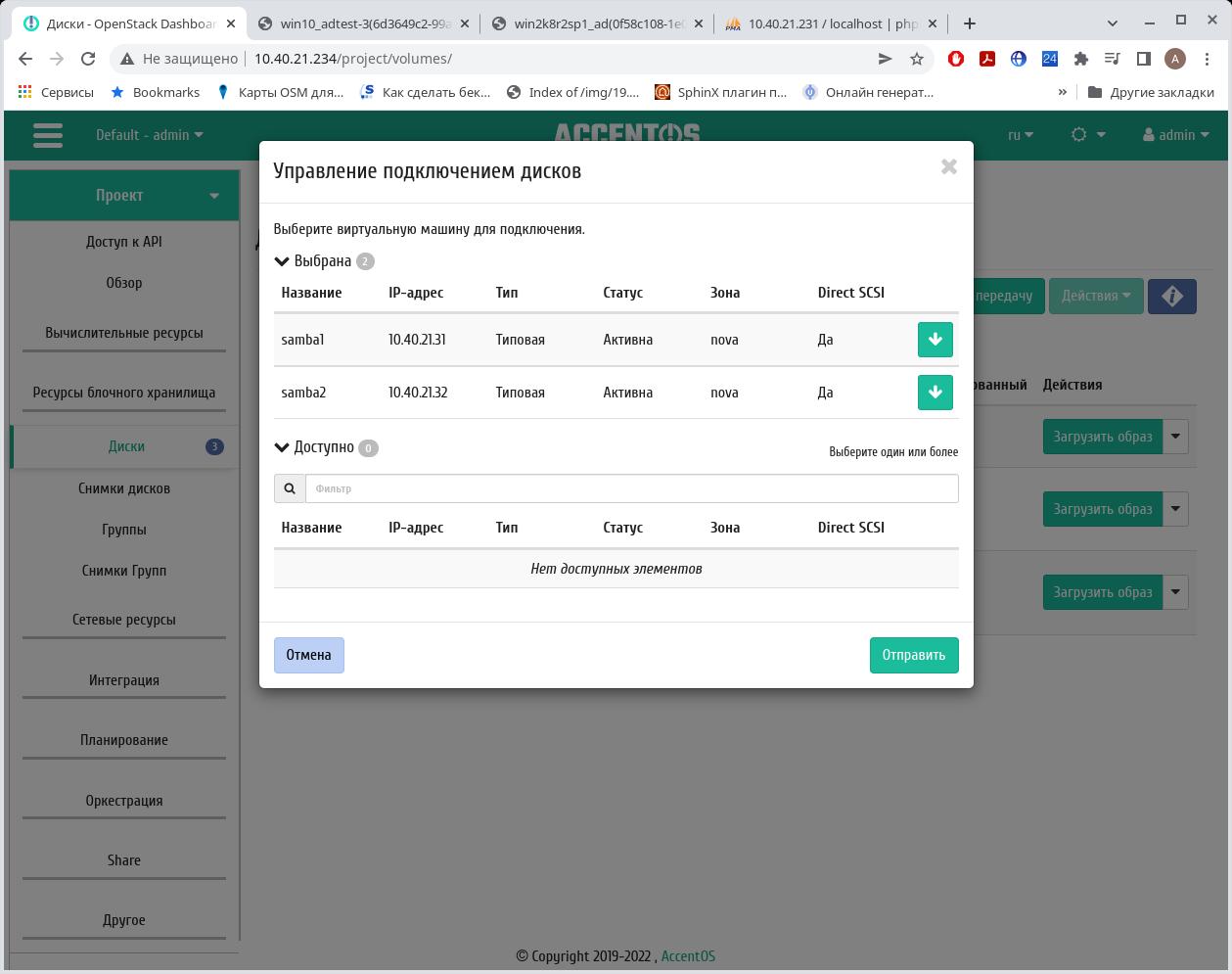
Работа служб Samba, работающих на разных ВМ и балансирующих нагрузку между собой, требует общего хранилища, в качестве которого мы выбираем отдельный виртуальный диск, который будет подключен ко всем созданным ВМ. Исходя из этого нам требуется тип диска cinder, поддерживающий множественное подключение. 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.

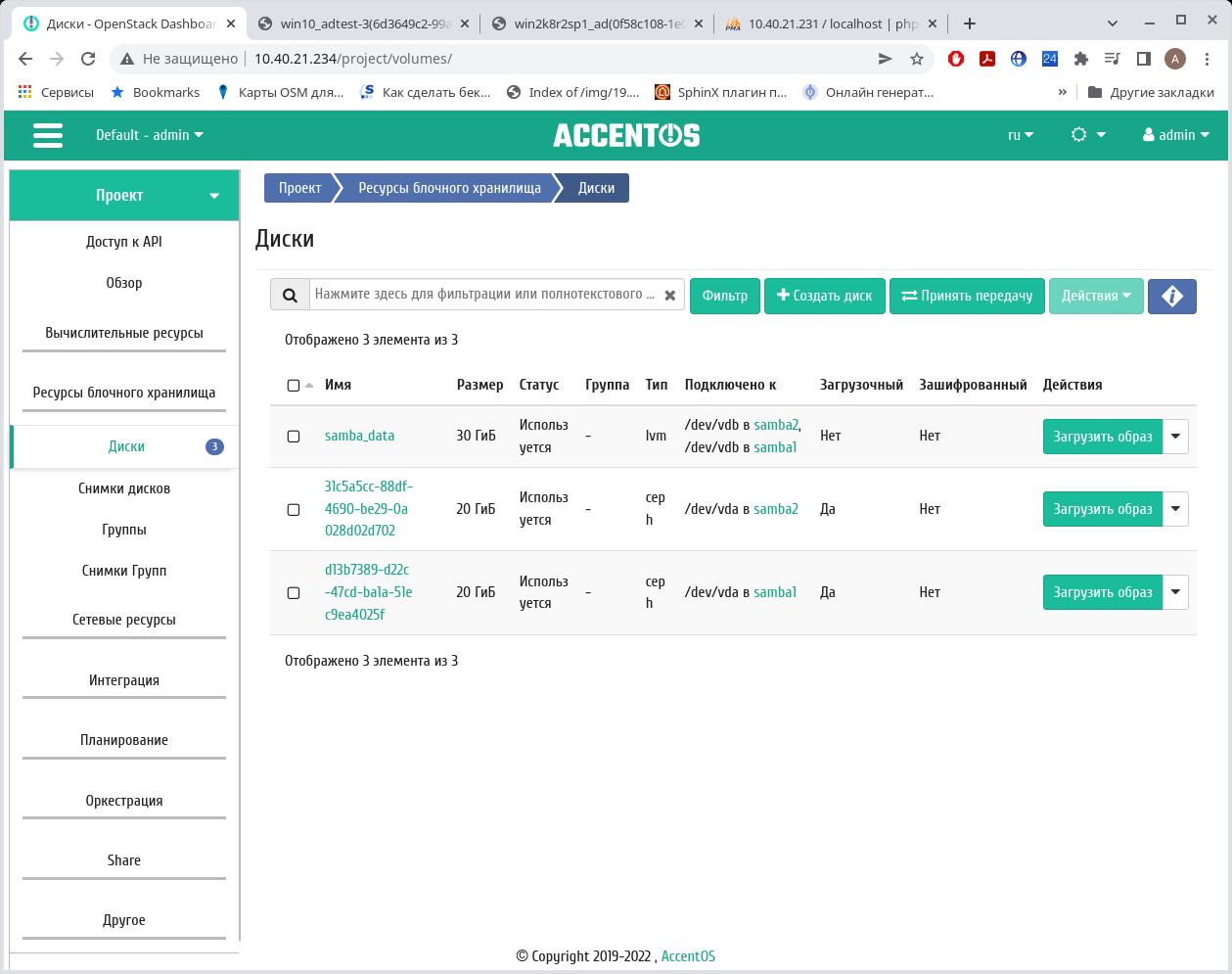
Завершением создания общего диска является подключение его ко всем виртуальным машинам нашей виртуальной инфраструктуры.











## 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"***

Список серверов времени для синхронизации. Сервера должны быть обязательно доступны, в противном случае возможны проблемы при работе с общим диском.

***export NTP\_SERVER="10.40.116.1 10.40.116.2"***

Имя домена Microsoft Active Directory, к кторому 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 Microsoft Active Directory.

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 NTP\_SERVER="10.40.116.1 10.40.116.2"*

*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-5.4 -y*

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

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

*export OCFS2\_CLUSTER\_NAME=ocfs2samba*

*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 ntp*

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

*sed -i "s/^server /#server /" /etc/ntp.conf*

*sed -i "s/^pool /#pool /" /etc/ntp.conf*

*for ntp in $NTP\_SERVER*

*do*

*echo "server $ntp iburst" >> /etc/ntp.conf*

*done*

*fi*

*systemctl enable ntp*

*systemctl restart ntp*

*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*

*cp /etc/krb5.conf /etc/krb5.conf.old*

*echo "[libdefaults]*

*dns\_lookup\_realm = false*

*dns\_lookup\_kdc = true*

*default\_realm = $DOMAIN\_UP*

*" > /etc/krb5.conf*

*cat > /usr/local/sbin/mkhomedir.sh <<EOF*

*#!/bin/bash*

*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 = $DOMAIN\_SHORT*

*security = ADS*

*realm = $DOMAIN\_UP*

*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*

*#no printers*

*printcap name = /dev/null*

*load printers = no*

*disable spoolss = yes*

*printing = bsd*

*[homes]*

*path = /mnt/samba/homes/%S*

*read only = no*

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

*browseable = no*

*create mask = 0700*

*directory mask = 0700*

*[profiles$]*

*path = /mnt/samba/profiles*

*read only = no*

*EOF*

*mkdir -pv /mnt/samba/homes*

*mkdir -pv /mnt/samba/profiles*

*systemctl restart winbind smbd nmbd*

*echo $DOMAIN\_ADMIN\_PASSWORD | kinit -fp ${DOMAIN\_ADMIN}*

*klist*

*echo $DOMAIN\_ADMIN\_PASSWORD | net ads join -U ${DOMAIN\_ADMIN}*

*wbinfo -i ${DOMAIN\_ADMIN}*

*sed -i '/^passwd:/ s/$/ winbind/' /etc/nsswitch.conf*

*sed -i '/^group:/ s/$/ winbind/' /etc/nsswitch.conf*

*sed -i '/^shadow:/ s/$/ winbind/' /etc/nsswitch.conf*

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

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

*[Service]*

*LimitNOFILE=500000*

*LimitNPROC=500000*

*TXT*

*systemctl daemon-reload*

*systemctl restart winbind smbd nmbd*

*wbinfo -i ${DOMAIN\_ADMIN}*

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

*chown ${DOMAIN\_ADMIN}:"$DOMAIN\_GROUP" /mnt/samba/homes/*

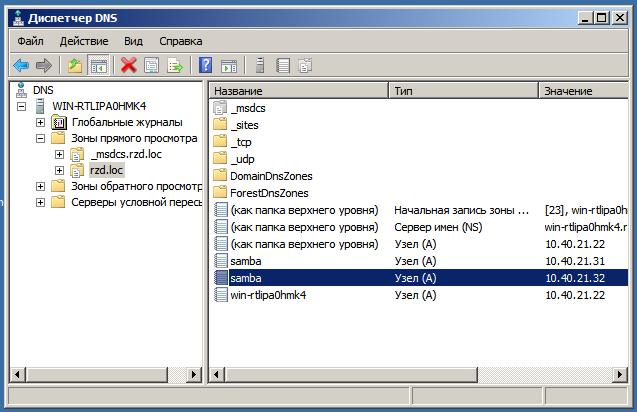
*chown ${DOMAIN\_ADMIN}:"$DOMAIN\_GROUP" /mnt/samba/profiles/*

*chmod 770 /mnt/samba/profiles/*

## Необходимая подготовка домена ActiveDirectory

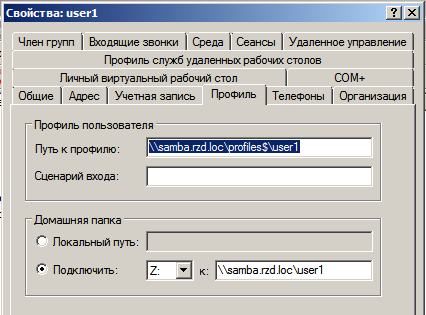
### Общее доменное имя.

Создаваемая инфраструктура может предоставлять доступ сетевым общим дискам по каждому ip-адресу или доменному имени, которое каждый виртуальный сервер получает перед процессом регистрации в домене. В DNS-зону Active Directory имена серверов могут быть не внесены при условии разрешения только защищенных обновлений DNS. Но для корректной работы инфраструктуры это не важно. Важным и необходимым является внесение одного общего доменного имени для всех виртуальных серверов, для которого будут соответствовать записи с типом A для всех ip-адресов виртуальных машин создаваемой инфраструктуры. По этому имени должен осуществляться доступ к сетевым общим дискам. Это позволит обеспечить балансировку нагрузки и переключение клиентов с одного сервера виртуальной инфраструктуры на другой в случае внезапной недоступности первого.



### Указание размещения профилей пользователей и домашних папок.

Для корректного использования перемещаемого профиля для Windows Server 2003 или Windows Server 2008 в режиме совместимости леса с Windows Server 2008 необходимо для каждого доменного пользователя корректно заполнить вкладку “Профиль”.



В данном случае используется общее доменное имя ресурса сетевых дисков samba.rzd.loc для хранения перемещаемых профилей и домашней папки. Необходимо учитывать, что данные настройки будут использоваться только на тех рабочих станциях, на которые ранее пользователь не входил. В противном случае будут задействованы ранее загруженные данные по профилю пользователя на локальном диске и сетевой диск для домашней папки не будет примонтирован.