

Руководство по установке и начальному конфигурированию

ECSS-10, версия 3.14.11

Содержание

1	Аннотация.....	6
2	Целевая аудитория.....	7
3	Разработка проекта системы.....	8
4	Подготовка сервера для инсталляции системы.....	10
5	Требования к системе.....	11
6	Минимальные требования при развертывании ECSS-10 на виртуальных ресурсах.....	12
7	Подразделы.....	13
8	Инсталляция системы.....	14
8.1	Инсталляция ОС.....	14
8.1.1	Предварительные требования.....	14
8.1.2	Установка ОС.....	14
8.1.3	Установка часового пояса.....	20
8.1.4	Проверка установки ОС.....	21
8.1.5	Настройка /etc/hosts.....	21
8.2	Настройка сетевых интерфейсов.....	21
8.3	Обновление ОС и инсталляция необходимого ПО.....	24
8.3.1	Обновление системы.....	24
8.3.2	Инсталляция необходимого ПО.....	24
8.4	Инсталляция пакетов ECSS.....	26
8.4.1	Предварительные требования.....	26
8.4.2	Инсталляция обязательных пакетов.....	26
8.4.3	Проверка доступности интерфейсов по dns-именам.....	33
8.5	Синхронизация времени на серверах.....	33
8.5.1	Установка и настройка NTP.....	34
8.6	Настройка Token.....	39
8.6.1	Установка ПО и подключение Token.....	39
8.6.2	Проверка работы Token.....	40
8.6.3	Перезагрузка Token по SSH в случае его зависания.....	43
8.6.4	Проблема с работой Token на серверах DEPO.....	44
8.7	Настройка listen-интерфейса для сервиса epmd.....	44
8.8	Запуск и активация системы.....	45
8.9	Особенности установки системы в кластере.....	49
8.9.1	Установка ECSS-10 в кластере.....	49
8.9.2	Вывод из обслуживания одного сервера.....	63

8.10	Проверка корректности инсталляционных процедур	64
9	Начальная настройка конфигурации	65
9.1	Понятия и определения	65
9.2	Предварительные условия	66
9.3	Порядок начальной настройки конфигурации.....	67
9.3.1	Конфигурирование кластеров.....	67
9.3.2	Настройка программного медиасервера	71
9.3.3	Создание и настройка доменов (виртуальных АТС)	71
9.3.4	Настройка дополнительных сервисов.....	76
9.4	Настройка ECSS-10 для производительных систем	77
9.4.1	Выделение отдельных ядер процессора для MSR.....	77
9.4.2	Установка scaling_governor в режим performance	77
9.4.3	Запуск MSR на изолированных ядрах процессора.....	79
9.4.4	Настройка использования определенных ядер процессора для erlang-based служб.....	79
10	Схема развертывания MySQL master-master replication с использованием keepalive.....	82
10.1	Установка ecss-mysql и создание репликации	82
10.1.1	Проверка корректности настройки репликации.....	89
10.2	Удаление репликации	90
11	Примеры пошаговой первоначальной настройки ECSS-10	91
11.1	Первоначальная установка системы без резервирования с одним сервером	92
11.1.1	Исходные данные	92
11.1.2	Включение в сеть	94
11.1.3	Настройка сети	96
11.1.4	Обновление программного обеспечения операционной системы	98
11.1.5	Установка и настройка программного обеспечения.....	99
11.1.6	Настройка начальной конфигурации.....	104
11.2	Первоначальная установка системы с резервированием в кластере из двух серверов	106
11.2.1	Исходные данные	106
11.2.2	Включение в сеть	109
11.2.3	Настройка сети	111
11.2.4	Установка и настройка программного обеспечения.....	116
11.3	Пример первичной конфигурации системы с помощью web- конфигуратора.....	133
11.3.1	Исходные данные	133
11.3.2	Подготовка к работе	133

11.3.3	Создание учетной записи оператора.....	134
11.3.4	Создание домена.....	138
11.3.5	Создание IP-set (sip-транспорта) и присвоение его к домену.....	142
11.3.6	Создание абонентов.....	145
11.3.7	Создание и применение контекстов маршрутизации для домена	147
11.3.8	Создание транка	151
11.3.9	Создание IVR-скрипта.....	153
11.3.10	Завершение настройки маршрутизации	155
11.3.11	Настройка услуг	156
12	Регламент обновления/деинсталляции ECSS.....	158
12.1	Обновление с версии 3.11 на 3.14	158
12.1.1	Подготовка к обновлению системы.....	158
12.1.2	Обновление ОС.....	160
12.1.3	Установка ECSS-10 версии 3.14.....	160
12.2	Обновление второго сервера, если система в кластере	166
12.2.1	Сборка кластера	166
12.2.2	Запуск кластера.....	168
12.2.3	Проверка работы системы в кластере	168
12.3	Деинсталляция компонентов ECSS	170
12.3.1	Удаление ecss-mysql	170
12.3.2	Удаление пакета ecss-node	171
12.3.3	Удаление пакета ecss-media-server.....	171
12.3.4	Удаление ненужных пакетов	172
12.3.5	Удаление пакетов по маске	172
12.3.6	Команды полной деинсталляции ECSS	172
12.3.7	Debconf	172
13	Рекомендации по обеспечению безопасности системы.....	173
13.1	Общие требования	173
13.1.1	Отключить доступ через root на сервер.....	173
13.1.2	Включить доступ только по ключам ssh	173
13.1.3	Использовать безопасные ключи rsa -b 4096	173
13.1.4	Автоматическая генерация паролей для SIP-пользователей	174
13.1.5	Изменить пароли на безопасные	174
13.1.6	Использовать SBC, настроить firewall, ACL	175
13.1.7	Обновлять Ubuntu и утилиты сервера при выходе пакета безопасности.....	175
13.2	Настройка безопасности OpenSSH.....	175
13.2.1	Настройка конфигурации сервера и клиента ssh.....	175
14	Чек-лист по установке ECSS-10.....	184

1 Аннотация

В данной части руководства приводится информация о разработке проекта развертывания, установке и первоначальной настройке программно-аппаратного комплекса ECSS-10 (в дальнейшем именуемого ECSS-10, комплекс или система).

2 Целевая аудитория

Руководство предназначено для технического персонала, выполняющего первичную установку и настройку ECSS-10. Квалификация технического персонала предполагает знание основ работы стеков протоколов ОКС-7, TCP/IP, UDP/IP, SIP, SIP-T, SIGTRAN, MEGACO, принципов построения Ethernet-сетей и практические навыки работы с UNIX-подобными ОС (Debian, Ubuntu, freeBSD, Red Hat Linux).

3 Разработка проекта системы

В данной главе приводится информация для начала разработки проект-схемы будущей системы.

Проект схемы выполняется перед инсталляцией системы.

При разработке проект-схемы необходимо знать:

- функциональный состав системы: протоколы, по которым будет осуществляться подключение шлюзов к ECSS-10, способы получения тарификационной и статистической информации;
- набор услуг, предоставляемый системой, помимо коммутации телефонных вызовов, уровень расчетной нагрузки по вызовам с услугами;
- предполагаемый уровень расчетной нагрузки на систему: количество вызовов в секунду на определенных направлениях, суммарная нагрузка на систему;
- необходимый уровень резервирования системы;
- требования по обеспечению поддержки СОРМ.

В зависимости от функциональных требований системы составляется перечень необходимых подсистем ECSS.

Обязательными компонентами ECSS являются:

- кластер Storage;
- кластер BUS;
- кластер Core;
- кластер Mediator.

В зависимости от функциональных требований выбираются следующие компоненты:

- кластер Adapter SIP,
- кластер Adapter Megaco,
- кластер Adapter Sigtran.

Исходя из уровня расчетной нагрузки на систему и требований по резервированию выбираются:

- количество и характеристики аппаратных ресурсов (тип процессора, количество RAM, размер дискового массива, количество сетевых интерфейсов), на которых будет разворачиваться система;
- топология распределения кластеров на аппаратных ресурсах (серверах) системы.

Далее при известной информации об аппаратных ресурсах и распределении кластеров на этих ресурсах производится проектирование сетевого уровня системы. Интерфейсам назначаются сетевые адреса, определяется топология связи серверов между собой, исходя из требований по резервированию.

В итоговом документе, описывающем систему, должна содержаться информация, приведенная в таблицах 1–3.

Таблица 1. Аппаратная конфигурация системы

Компонент	Модель	Характеристики
сервер 1	HP G360	CPU, RAM, HDD, Ethernet count
сервер 2	HP G360	CPU, RAM, HDD, Ethernet count
...
сервер n	HP G160	CPU, RAM, HDD, Ethernet count

Таблица 2. Топология сетевых соединений

Сервер	Имя сервера	Интерфейс	IP	MASK	Gateway	connected to
сервер 1	ecss1	Eth 0	IP	MASK	Gateway	Switch 1:port 1 или сервер 2:Ethx

Сервер	Имя сервера	Интерфейс	IP	MASK	Gateway	connected to
сервер 1	ecss1	Eth 1	IP	MASK	Gateway	Switch 1:port 2 или сервер 2:Eth1
сервер 2	ecss2	Eth 0	IP	MASK	Gateway	Switch 1:port 3 или сервер 1:Ethn
сервер 2	ecss2	Eth 1	IP	MASK	Gateway	Switch 1:port 4 или сервер 1:Eth1

Таблица 3. Топология программных компонентов

Сервер	BUS	Core	Mediator	Adapter SIP	Adapter Megaco	Adapter Sigtran
сервер 1	bus1@ecss1	core1@ecss1			sip1@ecss1	
сервер 2	bus1@ecss2	core1@ecss2				
сервер 3	bus1@ecss3			sip1@ecss3		

Удобным средством визуализации представленной выше информации является структурная схема ECSS.

Примеры типовых решений приведены ниже.

Пример: Структура ECSS для сертификации в ЦНИИС

Таблица 4. Аппаратная конфигурация системы

Компонент	Модель	Характеристики
сервер 1	HP Proliant DL360 G7	E5620 QuadCore, 2x2Gb, 2x300Gb 4xGbEth
сервер 2	HP Proliant DL360 G7	E5620 QuadCore, 2x2Gb, 2x300Gb 4xGbEth

Таблица 5. Топология сетевых соединений


Сервер	Имя сервера	Интерфейс	IP	MASK	Gateway	connected to
сервер 1	ecss1	Eth 0		255.255.255.0		Switch 1:port 1
сервер 1	ecss1	Eth 1		255.255.255.0		Switch 2:port 1
сервер 1	ecss1	Eth 2		255.255.255.0		Switch 1:port 2
сервер 1	ecss1	Eth 3		255.255.255.0		Switch 2:port 2
сервер 2	ecss2	Eth 0		255.255.255.0		Switch 1:port 3
сервер 2	ecss2	Eth 1		255.255.255.0		Switch 2:port 3
сервер 2	ecss2	Eth 2		255.255.255.0		Switch 1:port 4
сервер 2	ecss2	Eth 3		255.255.255.0		Switch 2:port 4

Таблица 6. Топология программных компонентов

Сервер	Storage	BUS	Core	Mediator	Adapter SIP	Adapter Megaco	Adapter Sigtran
сервер 1	ds1@ecss1	mycelium1@ecss1	core1@ecss1	md1@ecss1	sip1@ecss1	megaco1@ecss1	
сервер 2	ds1@ecss2	mycelium1@ecss2	core1@ecss2	md1@ecss2	sip1@ecss2	megaco1@ecss2	

4 Подготовка сервера для инсталляции системы

В данной главе описываются [инсталляция](#) и настройка операционной системы Linux Ubuntu Server 18.04 LTS, инсталляция и настройка необходимых для работы ECSS-10 [приложений](#). Инсталляция системы Softswitch ECSS-10 приведена в разделе "[Инсталляция пакетов ECSS](#)".

 Данная инструкция описывает подготовку системы без резервирования.

Предварительно необходимо согласовать сетевые интерфейсы, разработать схему сети, подготовить план нумерации.

На данном этапе осуществляется подготовка серверов к инсталляции программных компонентов ECSS-10. Подготовка серверов заключается в установке операционной системы с заданным набором пакетов и их последующей настройке.

Программные компоненты системы ECSS-10 являются кроссплатформенными. Потенциально поддерживаются различные операционные системы, но на данный момент официально протестированной и поддерживаемой ОС является Ubuntu Server 18.04 64bit, порядок установки и настройки которой будет приведен ниже.

5 Требования к системе

Актуальная информация с требованиями к серверам, необходимым для использования Softswitch ECSS-10, указана в разделе "[Первоначальная установка системы без резервирования с одним сервером](#)".

6 Минимальные требования при развертывании ECSS-10 на виртуальных ресурсах

Для нормальной работы до 1000 абонентов при нагрузке 20 cps/120 с длительностью требуется не менее чем:

- 8 ГБ RAM;
- Не менее 8 ядер высокой частотности (2.4 ГГц и выше): 4 под MSR, остальные под другие сервисы;
- Режим для работы сервера: высокопроизводительный (performance);
- HDD: рекомендуется не менее 50 ГБ при условии минимального логирования и отключения трассировок.

7 Подразделы

- Инсталляция системы
- Начальная настройка конфигурации
- Схема развертывания MySQL master-master replication с использованием keepalive
- Примеры пошаговой первоначальной настройки ECSS-10
- Регламент обновления/деинсталляции ECSS
- Рекомендации по обеспечению безопасности системы
- Чек-лист по установке ECSS-10

8 Инсталляция системы

8.1 Инсталляция ОС

В этом разделе приведено описание инсталляции операционной системы, а также необходимых и дополнительных пакетов. Система ECSS-10 версии 3.14 работает под управлением ОС **Ubuntu Server 18.04.x LTS 64bit**.

8.1.1 Предварительные требования

- Установочный загрузочный носитель с дистрибутивом операционной системы;
- Подготовленный сервер с обновленным BIOS, ILO (если есть), подключенная сеть для доступа в сеть Интернет;
- Выставленный первый приоритет загрузки с установочного носителя – USB Flash или CD/DVD в BIOS;
- Достаточный объем дискового пространства и памяти в соответствии с проектом.

8.1.2 Установка ОС

Для установки ОС необходимо выполнить следующее:

После загрузки с установочного носителя выбрать **"Install Ubuntu Server"**.

Выбрать язык системы и раскладку клавиатуры.

Настройка сетевых интерфейсов

Настроить сетевой интерфейс для подключения к сети Интернет:

Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient access for updates.

```
NAME    TYPE    NOTES
[ ens3  eth   -           ▶ ]
DHCPv4  192.168.122.13/24
52:54:00:cf:88:10 / Red Hat, Inc. / Virtio network device

[ Create bond ▶ ]
```

Edit ens3 IPv4 configuration

IPv4 Method: [Manual ▼]

Subnet:

Address:

Gateway:

Name servers:
IP addresses, comma separated

Search domains:
Domains, comma separated

[Save]
[Cancel]

[Done]
[Back]

Создание разделов диска

Выбрать параметр "Custom storage layout":

Configure a guided storage layout, or create a custom one:

Use an entire disk

[/dev/vda local disk 200.000G ▼]

Set up this disk as an LVM group

Encrypt the LVM group with LUKS

Passphrase:

Confirm passphrase:

Custom storage layout

[Done]
[Back]

Далее в LVM-группе создать дополнительные разделы в соответствии с таблицей 1.

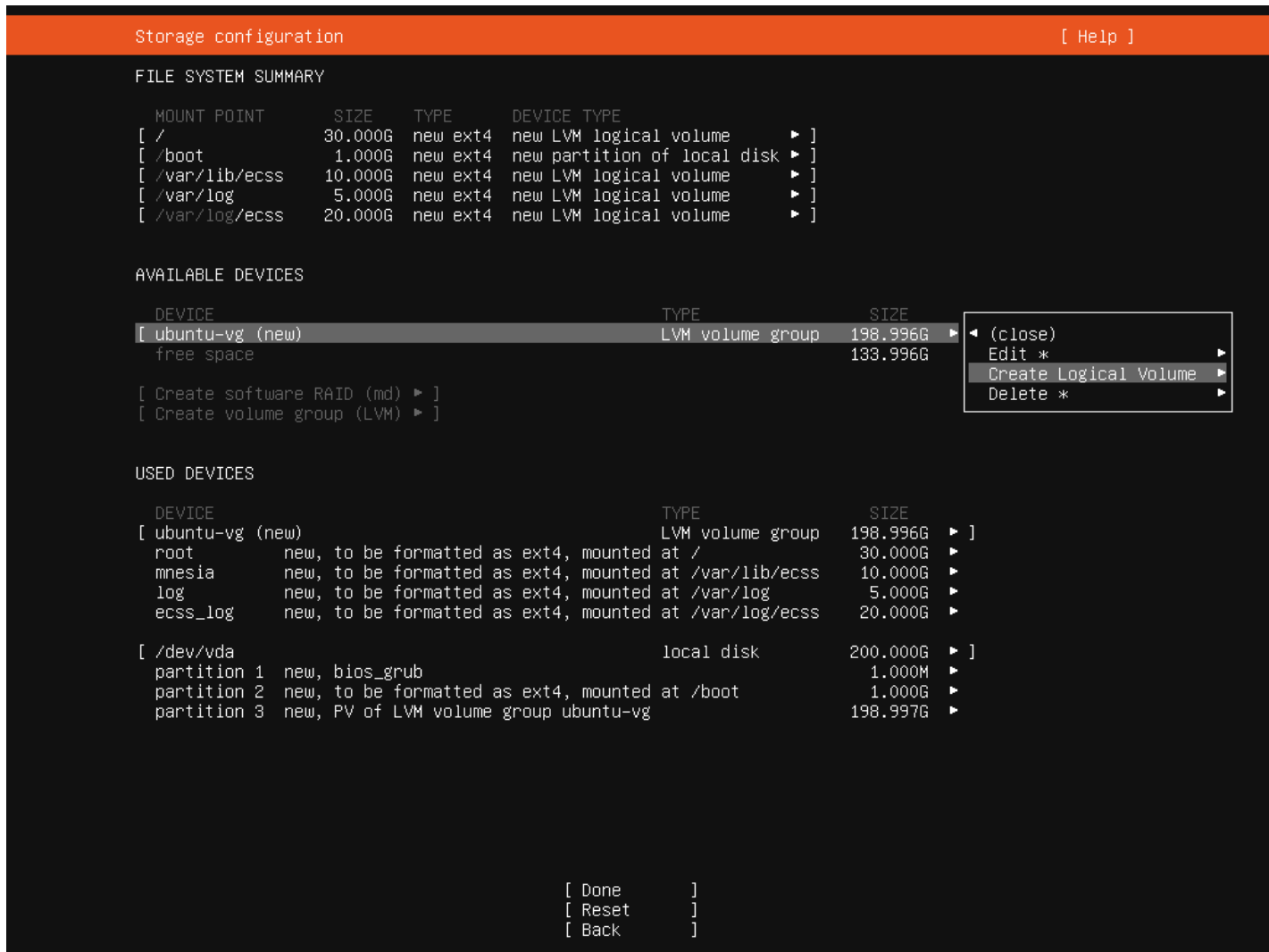


Таблица 1 – Вариант размещения информации в файловой системе на физических носителях для серверов

1	Загрузочный раздел операционной системы (создается автоматически)	boot	raid 1:hdd1,hdd2	boot	/boot	ext4	1 Gb	Первичный
2	Корневой раздел операционной системы	root	raid 1:hdd1,hdd2	root	/	ext4	30 Gb	Логический
3	Информация локальных баз данных	mnesia	raid 1:hdd1, hdd2	mnesia	/var/lib/ecss	ext4	10 Gb	Логический
4	Распределенная БД для хранения медиаресурсов	glusterfs	raid 1:hdd1, hdd2 или hdd3	glusterfs	/var/lib/ecss/ glusterfs*	ext4	Max Gb	Логический
5	Журналы функционирования подсистем ОС	log	raid 1:hdd1,hdd2 или hdd3	log	/var/log	ext4	5 Gb	Логический

6	Логический раздел для mysql базы	mysql	raid 1:hdd1,hdd2 или hdd3	mysql	/var/lib/mysql	ext4	30Gb	Логический
7	Журналы функционирования подсистем ECSS	ecss_log	raid 1:hdd1,hdd2 или hdd3	ecss_log	/var/log/ecss	ext4	20 Gb	Логический
8	Базы данных	ecss_db	raid 1:hdd1,hdd2 или hdd3	ecss_db	/var/lib/ecss-mysql	ext4	100–400 Gb**	Логический
9	Файлы пользователя	home	raid 1:hdd1,hdd2 или hdd3	home	/home	ext4	10 Gb	Логический

- ⚠ * Если сервер не будет работать в кластере, то вместо **glusterfs** создается раздел **/var/lib/ecss/restfs**.
- ** Рекомендуемое значение для серий **Light, Light+, Midi** – 100 Gb. Рекомендуемое значение для серии **Heavy** – 200 Gb, **Super Heavy** – 400 Gb.

Для работы системы необходимо как минимум 200 Gb свободного пространства.

Пример создания разделов диска размером 200 Gb:

```

Storage configuration [ Help ]

FILE SYSTEM SUMMARY

MOUNT POINT          SIZE  TYPE  DEVICE TYPE
[ /                  30.000G new ext4 new LVM logical volume ▶ ]
[ /boot              1.000G new ext4 new partition of local disk ▶ ]
[ /home              10.000G new ext4 new LVM logical volume ▶ ]
[ /var/lib/ecss      10.000G new ext4 new LVM logical volume ▶ ]
[ /var/lib/ecss-mysql 100.000G new ext4 new LVM logical volume ▶ ]
[ /var/lib/ecss/gusterfs 23.996G new ext4 new LVM logical volume ▶ ]
[ /var/log            5.000G new ext4 new LVM logical volume ▶ ]
[ /var/log/ecss      20.000G new ext4 new LVM logical volume ▶ ]

AVAILABLE DEVICES

No available devices

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES

DEVICE              TYPE              SIZE
[ ubuntu-vg (new)   LVM volume group 198.996G ▶ ]
root                new, to be formatted as ext4, mounted at / 30.000G ▶ ]
mnesia              new, to be formatted as ext4, mounted at /var/lib/ecss 10.000G ▶ ]
log                 new, to be formatted as ext4, mounted at /var/log 5.000G ▶ ]
ecss_log            new, to be formatted as ext4, mounted at /var/log/ecss 20.000G ▶ ]
ecss_db             new, to be formatted as ext4, mounted at /var/lib/ecss-mysql 100.000G ▶ ]
home                new, to be formatted as ext4, mounted at /home 10.000G ▶ ]
glusterfs           new, to be formatted as ext4, mounted at /var/lib/ecss/gusterfs 23.996G ▶ ]

[ /dev/vda          local disk        200.000G ▶ ]
partition 1        new, bios_grub   1.000M ▶ ]
partition 2        new, to be formatted as ext4, mounted at /boot 1.000G ▶ ]
partition 3        new, PV of LVM volume group ubuntu-vg 198.997G ▶ ]

[ Done ]
[ Reset ]
[ Back ]

```


Настройка имени пользователя и сервера

На серверах системы необходимо настроить параметр "**hostname**".

На всех серверах системы желательно указать одинаковое имя пользователя (любое, кроме **ssw**). Лицензия ECSS-10 привязывается к ключу eToken/ruToken и к имени компьютера (**hostname**), поэтому необходимо использовать стандартные значения. Системный пользователь **ssw** создается при инсталляции пакета **ecss-user**.

- ✓ Если используется один сервер, рекомендуемое значение hostname – **ecss1**;
При установке системы в кластере значение для первого сервера – **ecss1**, для второго – **ecss2**.

Profile setup [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name:

Your server's name:
The name it uses when it talks to other computers.

Pick a username:

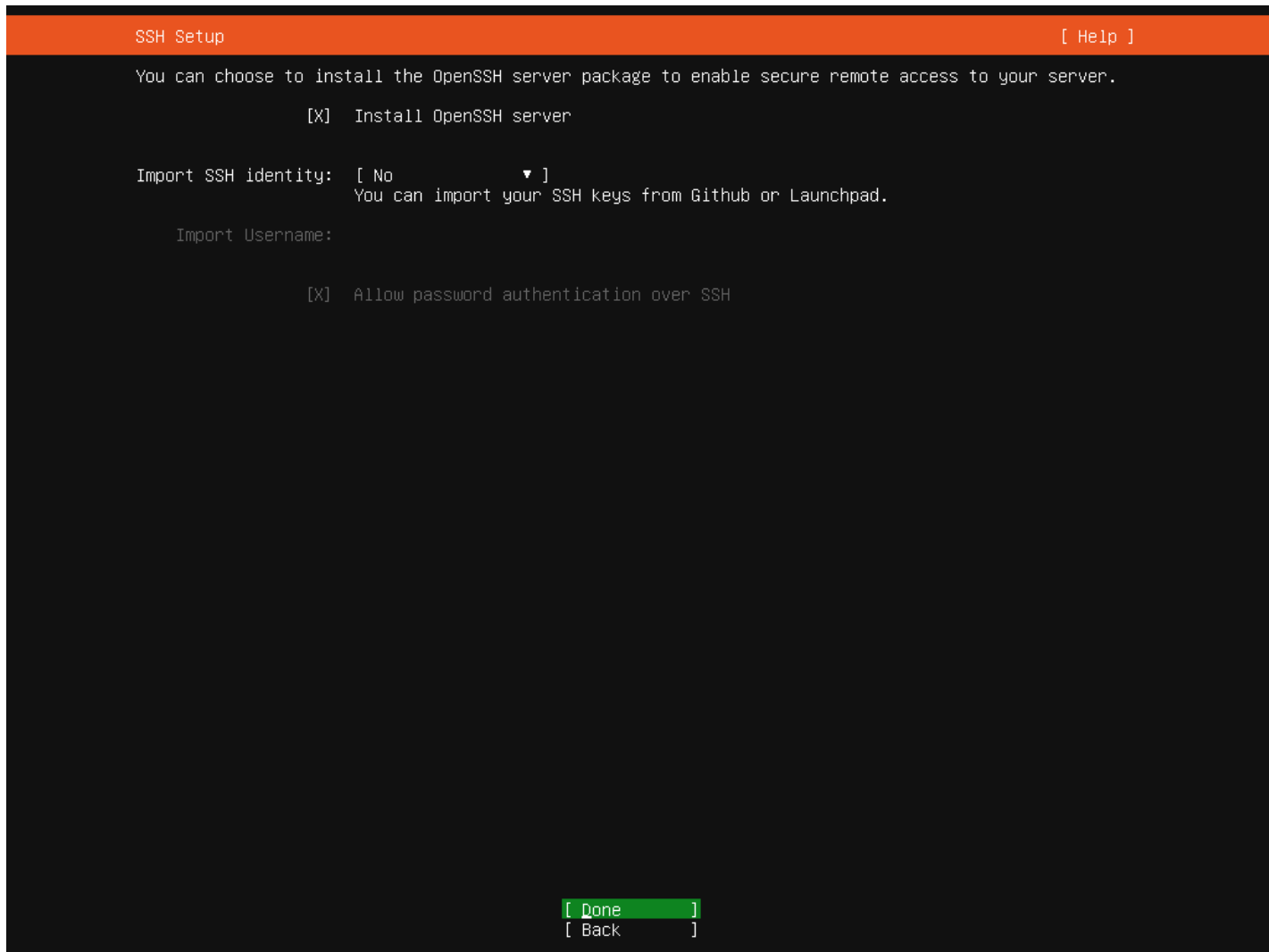
Choose a password:

Confirm your password:

[Done]

Установка OpenSSH server

В конце инсталляции ОС будет предложено установить дополнительное программное обеспечение для возможности удаленного подключения – необходимо установить *OpenSSH server*.



Отключение swap

В Ubuntu 18.04 swap-файл располагается в корневом каталоге — /swap. img.

⚠ По завершении инсталляции операционной системы необходимо отключить SWAP: SWAP на серверах с ECSS-10 использовать **нельзя!**

Для отключения:

```
sudo swapoff -a  
sudo rm /swap.img
```

8.1.3 Установка часового пояса

При инсталляции Ubuntu-18.04 не предлагается установить часовой пояс. Его можно установить вручную, например:

```
sudo timedatectl set-timezone Asia/Novosibirsk
```

8.1.4 Проверка установки ОС

Проверка в основном сводится к правильности создания разделов диска и наличия доступа по ssh.

Для вывода информации о состоянии дискового пространства введите команду **df -h**. Она покажет общее и занятое место в разделах. Размеры разделов должны соответствовать проекту и введенным значениям при установке.

Для проверки доступа по ssh с машины, находящейся в одной подсети со вновь установленным сервером, нужно выполнить команду:

```
ssh <user>@<IP_ecss>
```

где:

- <user> — имя пользователя, заданного при установке;
- <IP_ecss> — IP-адрес хоста, заданного при установке.

8.1.5 Настройка /etc/hosts

Доменному имени хоста ecss1 должен соответствовать адрес 127.0.1.1. Также нужно прописать адрес хоста ecss2. Для этого в файле /etc/hosts необходимо прописать IP-адреса хостов ecss.

Например, для кластера: ecss1 имеет адрес 192.168.1.21, ecss2 — 192.168.1.22. Данные адреса нужно прописать в /etc/hosts:

ecss1:

```
127.0.0.1 localhost
127.0.1.1 ecss1
192.168.1.22 ecss2
```

ecss2:

```
127.0.0.1 localhost
127.0.1.1 ecss2
192.168.1.21 ecss1
```

8.2 Настройка сетевых интерфейсов

 Получение адресов на сетевых интерфейсах по DHCP **недопустимо** на серверах ECSS.

Сетевые настройки необходимо выполнять с помощью [Netplan](#).

Пример:

Нужно настроить сервер с 4-мя сетевыми интерфейсами с агрегацией каналов (802.3ad) и необходимыми VLAN. Имеется шлюз для выхода в интернет — 192.168.1.203

- vlan2 – VoIP VLAN для трафика SIP/RTP;
- vlan3 – локальная сеть обмена между серверами кластера и локального управления;
- vlan476 – сеть взаимодействия с внешними корпоративными сервисами, статическая маршрутизация в подсеть 10.16.0.0/16 и маршрутизация в подсеть 10.136.16.0/24(для NTP):

```

sasha@ecss1:~$ cat /etc/netplan/10-ecss1_netplan.yaml
# netplan for ecss1
network:
  version: 2
  renderer: networkd
  ethernets:
    enp3s0f0:
      dhcp4: no
    enp3s0f1:
      dhcp4: no
    enp4s0f0:
      dhcp4: no
    enp4s0f1:
      dhcp4: no

  bonds:
    bond1:
      interfaces:
        - enp3s0f0
        - enp3s0f1
        - enp4s0f0
        - enp4s0f1
      parameters:
        mode: 802.3ad
        optional: true

  vlans:
    bond1.2: # Voip internal vlan 2
      id: 2
      link: bond1
      addresses: [192.168.2.21/24]
    bond1.3: # mgm internal vlan 3
      id: 3
      link: bond1
      addresses: [192.168.1.21/24]
      gateway4: 192.168.1.203
      nameservers:
        addresses: [192.168.1.203]
    bond1.476:
      id: 476 # mgm technology net vlan 476
      link: bond1
      addresses: [10.16.33.21/24]
      routes:
        - to: 10.16.0.0/16
          via: 10.16.33.254
          on-link: true
        - to: 10.136.16.0/24
          via: 10.16.33.254
          on-link: true

```

Для применения новых сетевых настроек необходимо выполнить команду **netplan apply**. Перезапуск сети или системы не требуется.

Подробнее про настройки netplan см. в [Приложении E. Netplan](#).

8.3 Обновление ОС и инсталляция необходимого ПО

8.3.1 Обновление системы

Добавление репозитория ELTEX:

```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/bionic/3.14 stable
main extras external' > /etc/apt/sources.list.d/eltex-ecss10-stable.list"
```

Обратите внимание, что требуется указать верную версию операционной системы при добавлении репозитория ELTEX:

- если установка происходит на Ubuntu 18.04, необходимо указать **bionic**, как приведено в примере выше.
- если ECSS-10 устанавливается на Astra Linux, необходимо указать соответствующие репозитории **smolensk**:

```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/smolensk/3.14 stable
main extras external' > /etc/apt/sources.list.d/eltex-ecss10-stable.list"
sudo sh -c "echo 'http://archive.eltex.org astra smolensk smolensk-extras' > /etc/apt/
sources.list.d/eltex-ecss10-stable.list"
```

Далее необходимо выполнить импорт ключа командой:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 33CB2B750F8BB6A5
```

Перед началом установки необходимо обновить ОС:

```
sudo apt update
sudo apt upgrade
```

8.3.2 Инсталляция необходимого ПО

Список обязательного сервисного программного обеспечения:


```
sudo apt install ntp tcpdump vlan dnsmasq
```

ntp	NTP-сервер
tcpdump	сниффер пакетов
vlan	управление VLAN
dnsmasq	легковесный DNS/DHCP-сервер

Список рекомендуемого диагностического и вспомогательного программного обеспечения:

```
sudo apt install aptitude atop ethtool htop iotop mc minicom mtr-tiny nmap pptpd pv
screen ssh tftpd vim sngrep tshark cpanminus gnuplot libgraph-easy-perl debconf-utils
```

aptitude	установка программ из репозитория, рекомендуется использовать вместо программы apt/apt-get
atop	мониторинг загрузки хоста с функцией периодического сохранения информации в файлы
ethtool	просмотр статистики сетевых интерфейсов
htop	мониторинг процессов
iotop	мониторинг подсистем ввода/вывода
mc	файловый менеджер
minicom	терминал для RS-232
mtr-tiny	выполняет функции ping и traceroute
nmap	сканер портов
pptpd	VPN-сервер
pv	мониторинг межпроцессного обмена
screen	мультиплексор терминалов
ssh	сервер и клиент SSH
tftpd	TFTP-сервер
vim	текстовый редактор
sngrep	трассировка sip
tshark	консольный аналог wireshark
cpanminus	просмотр медиа-трассировок (для просмотра необходимо воспользоваться командой sudo spanm Graph::Easy)
gnuplot	вывод графиков статистики
libgraph-easy-perl	Perl-модуль для преобразования или рендеринга графиков (в ASCII, HTML, SVG или через Graphviz)
debconf-utils	набор утилит для работы с базой debconf

 Данное программное обеспечение не требуется для работы системы ECSS-10, однако может упростить сервисное обслуживание системы и её отдельных компонентов со стороны инженеров эксплуатации и техподдержки.

Список обязательных пакетов для схем с резервированием:

```
sudo apt install ifenslave-2.6 keepalived attr
```

ifenslave-2.6	управление BOND-интерфейсами
keepalived	сервис мониторинга серверов/служб в кластере
attr	сервис управления атрибутами файловых систем

Список дополнительных пакетов для схем с резервированием:

```
sudo apt install bridge-utils ethtool
```

bridge-utils	управление bridge-интерфейсами
ethtool	управление и мониторинг сетевых интерфейсов

Для просмотра установленных пакетов выполните следующую команду (данный пункт является необязательным: его можно выполнить, если вы не уверены, что какое-то приложение установлено):

```
sudo dpkg --get-selections
```

8.4 Инсталляция пакетов ECSS

8.4.1 Предварительные требования

- Установленная и обновленная операционная система (Ubuntu-18.04);
- Отсутствие в системе пользователя с именем **ssw**;
- Разбиение дискового пространства в соответствии с [рекомендациями](#);
- Настроенная [сеть](#);
- Установленный набор необходимых [пакетов](#);
- Доступ к [репозиторию ELTEX](#).

✔ В ходе установки пакетов ECSS нужно будет ответить на ряд [вопросов](#) для формирования необходимой конфигурации. Для автоматической загрузки требуемых настроек можно воспользоваться командами из пакета `debconf-utils`. Описание и примеры использования при работе с `debconf` приведены в [Приложении В. Debconf](#).

Для инсталляции системы ECSS-10 необходимо устанавливать пакеты в порядке, в котором они описаны ниже в документации.

8.4.2 Инсталляция обязательных пакетов

Установка `ecss-mysql`

Первым необходимо установить пакет **`ecss-mysql`**.

⚠ Перед установкой необходимо убедиться, что в системе **не** установлен mysql-сервер, и папка **/var/lib/mysql/** пуста. При необходимости удалите все ее содержимое командой:

```
sudo rm -R /var/lib/mysql/
```

Если система разворачивается в кластере, то установку пакета и настройку репликации баз данных необходимо выполнить по инструкции из раздела "[Схема развертывания MySQL master-master replication с использованием keepalive](#)".

Для установки MySQL-сервера выполните команду:

```
sudo apt install ecss-mysql
```

При инсталляции пакета устанавливается сервер MySQL с нужными настройками, и создаются необходимые базы данных. В ходе установки будут запрошены следующие данные:

- **IP-маска для прав MySQL-таблиц ("IP pattern for MySQL permission")** – маска указывает, для какого пула IP-адресов будет доступен вход в базу данных:
 - если ecss-mysql устанавливается на том же хосте, что и остальная система (ecss-node), используйте адрес 127.0.0.%.
 - если ecss-mysql сервер будет установлен на другом хосте, то укажите пул адресов, в который будет входить адрес сервера, на котором будет установлен ecss-node. К примеру, если ecss-node будет установлен на сервер с IP-адресом 192.168.1.1/24, а ecss-mysql – на сервер с IP-адресом 192.168.1.2/24, то в ответе на этот вопрос нужно указать маску 192.168.1.%.
- **Логин для администратора mysql ("Login for MySQL root")** – данный логин будет установлен для сервера mysql.

⚠ Логин необходимо запомнить, так как он потребуется в ходе установки других нод. Также он используется в процессе создания резервной (backup) копии системы.

- **Пароль для администратора mysql ("Password for MySQL root")** – данный пароль будет установлен для пользователя, указанного в ответе на предыдущий вопрос.

⚠ Пароль необходимо запомнить, так как он потребуется в ходе установки других нод. Также он используется в процессе создания резервной (backup) копии системы.

Базы данных MySQL, используемые системой ECSS-10, после установки будут храниться в **/var/lib/ecss-mysql**. При установке пакета ecss-mysql apt задаст вопрос о разрешении изменения конфигурационного файла **/etc/apparmor.d/local/usr.sbin.mysql**, чтобы изменить путь до баз данных MySQL по умолчанию. Для успешной установки ecss-mysql требуется разрешить изменения (введите "Y"). Чтобы избежать ввода ответа на вопрос при установке пакета, допускается использовать дополнительные ключи при вводе команды установки:

```
sudo apt-get -o Dpkg::Options::="--force-confnew" install ecss-mysql
```

Проверка корректности установки

Чтобы убедиться в корректности установки после ее завершения, проверьте, запущен ли MySQL-сервер:

```
systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/mysql.service.d
            └─override.conf
   Active: active (running) since Fri 2021-09-24 20:52:50 +07; 2 weeks 2 days ago
 Main PID: 12374 (mysqld)
    Tasks: 94 (limit: 4915)
   CGroup: /system.slice/mysql.service
           └─12374 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

Sep 24 20:52:50 ecss1 systemd[1]: Starting MySQL Community Server...
Sep 24 20:52:50 ecss1 systemd[1]: Started MySQL Community Server.
```

Попробуйте войти в базу данных MySQL под логином (<LOGIN>), с паролем (<PASSWORD>), указанным при установке:


```
sudo mysql -u<LOGIN> -p<PASSWORD>
mysql>
```

В случае корректной установки откроется CLI MySQL-сервера.

Можно сразу посмотреть список созданных БД:


```
mysql> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema      |
| ecss_address_book      |
| ecss_audit              |
| ecss_calls_db          |
| ecss_dialer_db         |
| ecss_meeting_db        |
| ecss_numbers_db        |
| ecss_statistics        |
| ecss_subscribers       |
| ecss_system            |
| history_db             |
| mysql                  |
| performance_schema     |
| sys                    |
| web_conf               |
+-----+
```

Чтобы выйти из CLI MySQL, выполните команду **"exit"**.

 В целях безопасности в версиях mysql-5.7 и выше логин **root** разрешено использовать только для входа с локального хоста.

Установка ecss-node

Установка обязательного пакета **ecss-node** включает в себя установку и первоначальную настройку основных подсистем.

 В системе уже должен быть установлен пакет **ecss-mysql**

Для установки пакета **ecss-node** выполните команду:

```
sudo apt install ecss-node
```

Во время установки пакета создается пользователь **ssw**, от имени которого запускаются все сервисы **ecss***. Создаются необходимые каталоги, выполняется настройка DNS, идет настройка SSL-сертификатов. В ходе инсталляции будут задаваться [вопросы](#), необходимые для формирования конфигурационных файлов.

Настройка сертификатов

Актуален, только если был сгенерирован самоподписанный сертификат, тогда в систему установится `ecss10root.crt` (при копировании также пытается скачать `ecss10root.crt`, либо если при ручной установке был помещён данный файл). Если уже имеются сертификаты, то никаких действий не будет произведено. В конце также проверяется валидность сертификата.

Чтобы сгенерировать новый сертификат, необходимо удалить `ecss10.{pem,crt,key}` и `ecss10root.{crt,key}`, после чего сделать **dpkg-reconfigure ecss-user**.

Если планируется установка системы в кластере, то, как правило, на первом сервере нужно сгенерировать сертификат, а при установке **ecss-node** на втором сервере выбрать копирование с первого сервера (подробнее см. в разделе "[Настройка сертификатов ECSS-10](#)").

При установке будут заданы вопросы по сертификатам.

Способы конфигурирования сертификатов:

Ручной (manual)

При выборе ручного способа конфигурации сертификатов откроется окно с информацией о том, что установка может быть продолжена после помещения файлов `ecss10.{pem,crt,key}` в `/etc/ecss/sll`. Также данное окно может открыться по достижении конца установки. Поместите необходимые файлы в требуемую директорию и начните процесс установки заново (перезапустите установку). Если все действия были выполнены верно — установка завершится, и можно будет продолжить установку системы.

Сгенерировать самоподписанный сертификат (generate)

При выборе данного способа будут сгенерированы следующие вопросы:

- Страна (RU)
- Область (Novosibirsk)
- Город (Novosibirsk)
- Организация (ELTEX)
- Структурный узел (IMS)
- Имя сертификата (ecss10)
- Почта (ssw-team@eltex.loc)
- Количество дней жизни сертификата
- Пароль для корневого приватного ключа
- Алгоритм шифрования для ключа
- Сложность ключа

- Сложность для параметров Диффи-Хеллмана
- Дополнительные имена, за которые отвечает сертификат (на примере офиса — это ssw1.eltex.loc, ssw2.eltex.loc, ssw.eltex.loc), перечисленные через пробел (для последнего уровня можно wildcard)

Чем выше сложность ключа, тем дольше будет установка (dhparam при сложности 8192 на машине средней производительности занимает около часа). При отсутствии особых требований к безопасности можно оставить значение по умолчанию. После чего отобразится уведомление, что необходимо убрать приватный корневой ключ в безопасное место.

Скопировать существующие сертификаты (copy) по ssh

При выборе данного способа будут сгенерированы следующие вопросы:

- Логин (user)
- Адрес удалённой машины (ecss1)
- Порт (22)
- Способ авторизации (password или identity_file)
- Пароль (password)
- Файл с ключом (/home/<user>/.ssh/id_rsa)
- Путь до папки с сертификатом (/etc/ecss/ssl)

Скопировать по http

При выборе данного способа будут сгенерированы следующие вопросы:

- url (https://system.restfs.ecss:9993/certs)
- Логин (если используется авторизация basic)
- Пароль
- Скопировать с другого сервера **ecss10**

Используется API http_terminal

При выборе данного способа будут сгенерированы следующие вопросы:

- url до http_terminal (https://ecss1:9999)
- Логин (admin)
- Пароль (password)
- Нода с сертификатами (core1@ecss1)

Настройка ermd

Сервис ermd требует наличия IPv6-адреса на IO-интерфейсе. Для отключения (если в этом возникнет реальная потребность) IPv6 на отличных от IO интерфейсах используйте следующие настройки в sysctl:

```
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 0
```

DNS

В ходе установки пакета ecss-node выполняется конфигурирование внутренних DNS-адресов. При установке, в зависимости от текущей конфигурации системы, может отобразиться сообщение:

```
See "systemctl status dnsmasq.service" and "journalctl -xe" for details.
invoke-rc.d: initscript dnsmasq, action "start" failed.
• dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
```

Такой вывод в ходе установки является нормальным и не свидетельствует о проблемах. Главное, чтобы после окончания установки `ecss-node dnsmasq.service` был активен.

Пример:

```
sasha@ecss1:~$ systemctl status dnsmasq.service
● dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
   Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Fri 2021-09-24 20:52:03 +07; 2 weeks 3 days ago
   Main PID: 10914 (dnsmasq)
     Tasks: 1 (limit: 4915)
    CGroup: /system.slice/dnsmasq.service
            └─10914 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7 /etc/
dnsmasq.d, .dpkg-dist, .dpkg-old, .dpkg-new --local-service --trust-anchor=.,
19036, 8, 2, 49aac11d7b6f6446702e54a1607371607a1a41

Sep 24 20:52:03 ecss1 systemd[1]: Starting dnsmasq - A lightweight DHCP and caching
DNS server...
Sep 24 20:52:03 ecss1 dnsmasq[10890]: dnsmasq: syntax check OK.
Sep 24 20:52:03 ecss1 systemd[1]: Started dnsmasq - A lightweight DHCP and caching DNS
server.
```

Установка `ecss-media-server`

Пакет **`ecss-media-server`** — обязательный компонент для обработки VoIP-трафика. Медиасервер предназначен для проксирования речевой и видеоинформации по протоколу RTP, организации конференций, записи разговоров, воспроизведения медиафайлов и различных комбинаций этих режимов.

Для установки выполните:

```
sudo apt install ecss-media-server
```

Назначение, описание и настройка **`ecss-media-server`** приведены в разделе "[Настройка программного медиасервера](#)". В ходе установки нужно будет ответить на ряд [вопросов](#) для создания необходимых конфигурационных файлов.

Установка `ecss-restfs`

RestFS — компонент, обеспечивающий HTTP API для работы с файлами. Описание и методика настройки компонента приведены в разделе "[Настройка RestFS](#)". Для установки выполните:

```
sudo apt install ecss-restfs
```

В ходе установки нужно будет ответить на ряд [вопросов](#) для создания необходимых конфигурационных файлов. Также инсталлятор предложит установить и настроить пакет Text2speech от Yandex. Подробнее в разделе [Настройка сервиса tts для работы с Yandex-Speechkit](#).

Установка ecss-media-resources

Пакет включает в себя набор [системных аудиофайлов](#), предназначенных для проигрывания фраз автоответчиков и использования в сценариях IVR, а также набор инструментов для работы с пользовательскими аудиофайлами.

Для установки выполните:

```
sudo apt install ecss-media-resources
```

Установка ecss-web-conf

Web-конфигуратор позволяет сделать управление системой более наглядным и комфортным. Установка web-конфигуратора не является обязательной, но рекомендуется.

С помощью web-конфигуратора выполняется настройка, мониторинг и отладка системы с удаленного рабочего места через web-браузер. Описание компонентов приложения приведено в разделе "[Web-интерфейс](#)". Описание вопросов, задаваемых при инсталляции пакета, приведено в [Приложении Д](#).

Также при установке пакета **ecss-web-conf** автоматически устанавливается пакет **ecss-subscriber-portal-ui**. Приложение "*Портал абонента*" системы ECSS-10 позволяет абонентам системы самостоятельно управлять услугами, просматривать информацию по совершенным вызовам, активным конференциям, а также настраивать собственные IVR-скрипты для входящих вызовов. Описание работы веб-конфигуратора приведено в разделе "[Портал абонента](#)".

Для установки выполните:

```
sudo apt install ecss-web-conf
```

Если требуется отключить порт 80, сделайте это после установки пакета, убрав соответствующую секцию из файла `/etc/nginx/sites-available/ecss-web-conf`.

Инсталляция дополнительных необязательных пакетов

В репозитории также хранятся дополнительные пакеты, которые можно установить опционально, исходя из проекта.

Для установки дополнительных пакетов выполните:

```
sudo apt install <имя пакета 1> <имя пакета 2> ... <имя пакета N>
```

Список доступных дополнительных пакетов:

Имя пакета	Краткое описание
ecss-cc-ui	Автоматизированное рабочее место оператора Call-центра
ecss-teleconference-ui	АРМ менеджера селекторного совещания

Имя пакета	Краткое описание
ecss-utils	Скрипты для конвертации бинарных логов в текст
ecss-asr	Сервис автоматического распознавания речи
ecss-pda-api	API для Phone Desktop Assistant
ecss-autoprovision	Сервис автоматического конфигурирования телефонных аппаратов (AUP)
ecss-clerk	Сервис "Автосекретарь"
ecss-crm-server	Сервер интеграции с CRM
ecss-security	Сервис протоколирования действий пользователей

8.4.3 Проверка доступности интерфейсов по dns-именам

Проверить работу dnsmasq можно простыми **ping**:

```
ping -c1 cocon.mysql.ecss
ping -c1 dialer.mysql.ecss
ping -c1 statistics.mysql.ecss
ping -c1 tc.mysql.ecss
ping -c1 tts.mysql.ecss
ping -c1 controlchannel.zmq.ecss
ping -c1 system.restfs.ecss
```

Все интерфейсы должны быть доступны.

8.5 Синхронизация времени на серверах

 Перед настройкой NTP нужно убедиться, что в системе установлен пакет **ntp**.

Пример:

```
sasha@ecss1:~$ dpkg -l | grep ntp
ii ntp 1:4.2.8p10+dfsg-5ubuntu7.3
amd64 Network Time Protocol daemon and utility programs
ii sntp 1:4.2.8p10+dfsg-5ubuntu7.3
amd64 Network Time Protocol - sntp client
```

Затем рекомендуется задать значение текущей системной даты, максимально приближенное к реальному времени. Для этого можно воспользоваться утилитой для ручной синхронизации времени **ntpdate**.

Пример для установки времени с сервера ntp.ubuntu.com:

```
sudo ntpdate ntp.ubuntu.com
```

Команда **date** без параметров выводит на экран текущее системное время.

8.5.1 Установка и настройка NTP

Конфигурация NTP настраивается при установке пакета **ecss-node**.

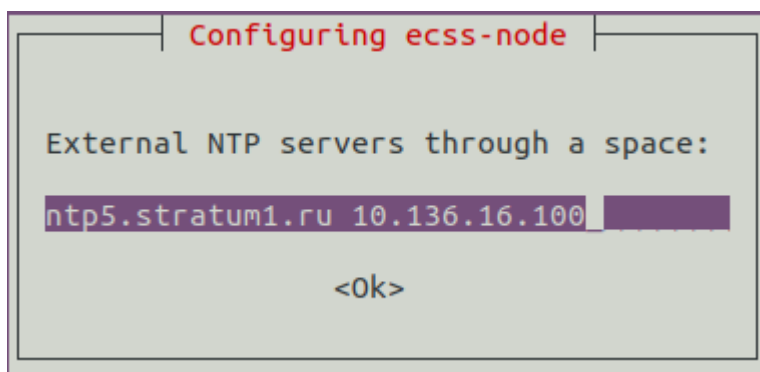
Рассмотрим настройку NTP для кластера из 2-х серверов ecss со следующими параметрами:

Параметр	Значение
Адреса внешних NTP-серверов	<ul style="list-style-type: none">ntp5.stratum1.ru10.136.16.100
Локальная синхронизация серверов кластера между собой (orphan mode)	Да, для следующих адресов: <ul style="list-style-type: none">ecss1 - 192.168.1.21ecss2 - 192.168.1.22
Подсети, с которых разрешено другим устройствам синхронизироваться с данным сервером	<ul style="list-style-type: none">192.168.1.0/2410.16.0.0/16

В ходе установки будут заданы несколько **вопросов** для формирования файла конфигурации.

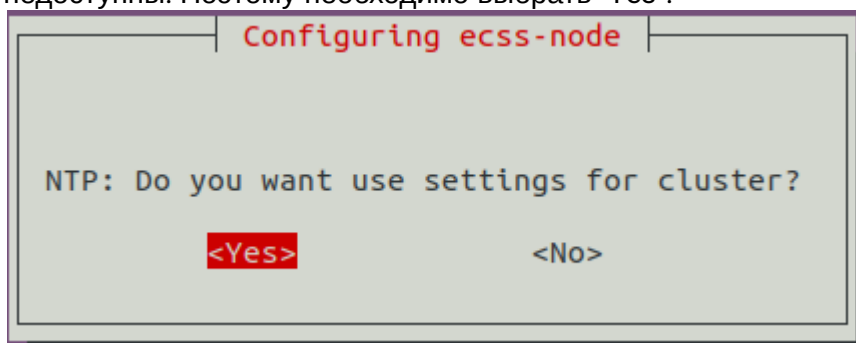
Ниже приведен пример ответа на вопросы:

Необходимо ввести внешние сервера через пробел (по умолчанию ntp.ubuntu.com):

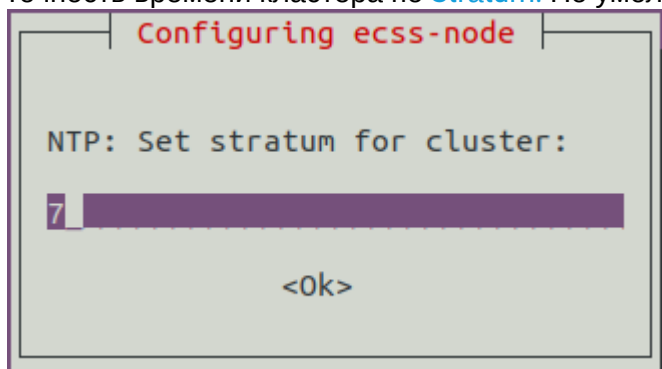


Необходимо разрешить (Yes) или запретить (No) активацию режима tos orphan (режим для кластера, в котором серверы самостоятельно регулируют синхронизацию). Если система устанавливается в кластере, то серверы ECSS должны иметь **одинаковое** время, даже если внешние NTP-серверы

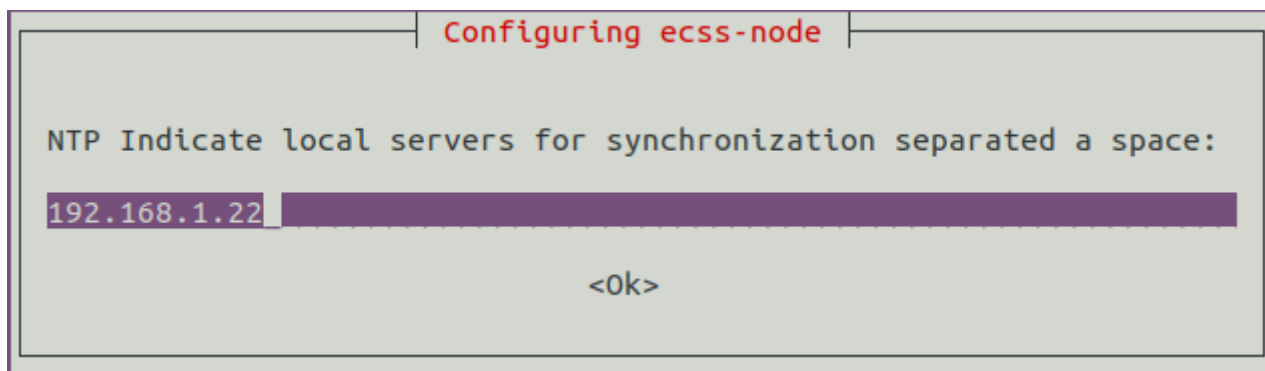
недоступны. Поэтому необходимо выбрать "Yes".



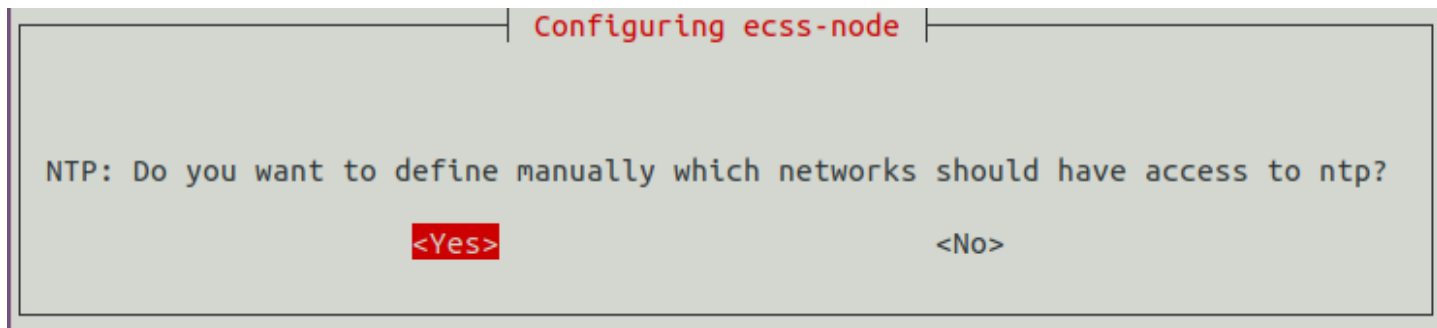
Точность времени кластера по [Stratum](#). По умолчанию – 7:



Предлагается ввести адреса соседних серверов кластера для их синхронизации между собой. В данном примере настраиваем ecss1, поэтому вводим адрес ecss2. При настройке ecss2 соответственно вводится адрес ecss1. Если серверов несколько, необходимо перечислить их через пробел.



Далее предлагается настроить адреса подсетей, из которых другим устройствам разрешено синхронизироваться с данным сервером:



Указываются сети, которые могут иметь доступ до данного сервера, чтобы другие ноды, а также прочие устройства могли синхронизировать время с данным сервером. Формат указания сетей: <адрес_сети| маска_сети>. Если сетей несколько, необходимо перечислить их через пробел.

```
Configuring ecss-node
Format: <ip>|<mask> (x.x.x.x|255.255.255.0)
NTP: Networks, which must have access to the ntp through a space:
192.168.1.0|255.255.255.0 10.16.0.0|255.255.0.0
<Ok>
```

После инсталляции настройки сохраняются в файле **/etc/ecss/ecss-ntp.conf**. Пример получившегося файла для сервера **ecss1**:

```

# /etc/ntp.conf

# http://www.k-max.name/linux/ntp-server-na-linux/
# В preinst делаем резервную копию старого и устанавливаем текущий
# В postrm загружаем из резервной копии

# Смещение системных часов
driftfile /var/lib/ntp/ntp.drift
# Логи
logfile /var/log/ntp
# Статистика синхронизации времени
statsdir /var/log/ntpstats/

# Разрешает записывать статистику:
# loopstats - статистика для петли (loopback)
# peerstats - статистика пиров
# clockstats - статистика драйвера времени
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# Активация Orphan mode – режима синхронизации времени для кластеров. Устанавливаем
для него stratum (уровень точности: число от 1 до 16)
# tos orphan <stratum>
# TOS
tos orphan 7 #### INSTALLED AUTOMAT BY ECSS10

# Сервера локальной сети
# peer <ip|domain>
# LOCAL_SERVERS
peer 192.168.1.22 #### INSTALLED AUTOMAT BY ECSS10

# Интернет-сервера
# server xx.xx.xx.xx iburst
# restrict xx.xx.xx.xx
# INTERNET_SERVERS
server ntp5.stratum1.ru iburst #### INSTALLED AUTOMAT BY ECSS10
restrict ntp5.stratum1.ru #### INSTALLED AUTOMAT BY ECSS10
server 10.136.16.100 iburst #### INSTALLED AUTOMAT BY ECSS10
restrict 10.136.16.100 #### INSTALLED AUTOMAT BY ECSS10

# Ограничение доступа к конфигурируемому серверу:
# По умолчанию игнорируем все
restrict -4 default kod notrap nomodify nopeer noquery limited
restrict -6 default kod notrap nomodify nopeer noquery limited
restrict source notrap nomodify noquery

# Localhost без параметров – значит разрешено все. Параметры идут только на запреты.
restrict 127.0.0.1
restrict ::1

```

Для **ecss2** файл будет аналогичным, за исключением строки пира к соседнему серверу (192.168.1.21):

```
peer 192.168.1.21 ### INSTALLED AUTOMAT BY ECSS10
```

В режиме *Orphan* серверы в кластере синхронизируются друг от друга, сами определяют мастера и следят, чтобы в рамках кластера часы шли синхронно.

Если появляется мастер-сервер NTP со значением `stratum` меньше заданного для кластера, то кластер автоматически перенастраивается на синхронизацию от него. Таким образом выполняется условие постоянного наличия единственной точки синхронизации времени.

Все зависимые устройства в системе ECSS-10 должны синхронизироваться от серверов кластера. Если используется схема без резервирования, от настройки режима для кластера можно отказаться: тогда в конфигурационном файле будет отсутствовать секция настройки локальных серверов для синхронизации между собой.

⚠ Править конфигурационный файл вручную не рекомендуется, так как при обновлении пакета `ecss-node` в файл запишутся прошлые настройки из базы [debconf](#) от последней реконфигурации пакета.

Правильный способ — использовать команду **dpkg-reconfigure**:

```
sudo dpkg-reconfigure ecss-node
```

✔ Если в файл конфигурации все же были вручную внесены какие-либо изменения, то для применения команды **dpkg-reconfigure** нужно перезапустить сервис NTP:

```
sudo systemctl restart ntp.service
```

Для просмотра информации о состоянии синхронизации используется команда **ntpq -p**. Если использовать дополнительный ключ **-n**, вместо имени сервера будет указан IP-адрес:

Пример:

```
sasha@ecss1:~$ ntpq -p
  remote                refid                st t when poll reach  delay  offset  jitter
=====
ecss2                   88.147.254.229       2 s  11  64  377   0.099  -1.169  0.357
+10.136.16.100          194.58.204.148       2 u   56 128  377   4.008  -2.482  0.339
*88.147.254.229        .PPS.                 1 u  124 128  377  60.440   0.691  0.098
```

Описание параметров:

- `remote` — имя удаленного NTP-сервера;
- `refid` — IP-адрес сервера, с которым производит синхронизацию удаленный сервер NTP;
- `st` — `stratum` (уровень): число от 1 до 16, отражающее точность сервера;
- `t` — тип удаленного сервера:
 - `u` — `unicast`,
 - `l` — `local`,

- m – multicast,
- s – symmetric (peer),
- b – broadcast;
- when – интервал времени (в секундах), прошедший с момента получения последнего пакета от данного сервера;
- poll – интервал между опросами (в секундах), значение варьируется;
- reach – состояние доступности сервера. Восьмеричное представление массива из 8 бит, отражающего результаты последних восьми попыток соединения с сервером. Если последние 8 попыток синхронизации с удаленным сервером были успешны, то параметр принимает значение 377;
- delay – вычисленная задержка ответов от сервера (RTT) в миллисекундах;
- offset – разница между временем локального и удаленного серверов;
- jitter – джиттер, мера статистических отклонений от значения смещения (поле offset) по нескольким успешным параметрам запрос-ответ.

Значение символов перед именами серверов

- x – фальшивый источник по алгоритму пересечения;
- . – исключён из списка кандидатов из-за большого расстояния;
- – удален из списка кандидатов алгоритмом кластеризации;
- + – входит в конечный список кандидатов;
- # – выбран для синхронизации, но есть 6 лучших кандидатов;
- * – выбран для синхронизации;
- o – выбран для синхронизации, но используется PPS;
- пробел – слишком большой уровень, цикл или явная ошибка;

После старта сервиса может потребоваться около 10 минут для установления синхронизации времени с базовым NTP-сервером.

Проверить состояние настроенного сервера NTP можно с помощью команды **ntpdate**:

```
sasha@ecss1:~$ sudo ntpdate -q localhost
server 127.0.0.1, stratum 2, offset -0.000032, delay 0.02573
28 Sep 15:00:57 ntpdate[19002]: adjust time server 127.0.0.1 offset -0.000032 sec
```

Как видно, значение stratum сервера стало равно 2.

8.6 Настройка Token

Token (токен) – это USB-ключ лицензионной защиты. Его наличие необходимо для корректной работы системы лицензирования и SSW в целом. Раньше при покупке лицензии серверы ECSS поставлялись с ключами **eToken**, в последнее время новые инсталляции комплектуются USB-ключами **Рутокен**.

8.6.1 Установка ПО и подключение Token

Все необходимые для работы RuToken библиотеки устанавливаются из [репозитория ELTEX](#) вместе с пакетом **ecss-node**.

⚠ Для нормальной работы **eToken** необходимо самостоятельно установить пакет:

```
sudo apt install safenetauthenticationclient-core
```

Вставьте токен в USB-разъем на сервере.

Для проверки подключения USB-ключа к серверу выполните команду:

```
lsusb
```


Если ключ обнаружен, будет выведена строка:

- для eToken:

```
Bus 003 Device 002: ID 0529:0620 Aladdin Knowledge Systems
```

- для Рутокен:

```
Bus 005 Device 002: ID 0a89:0030
```

 Если ключ не определяется, то выполните следующие команды в указанной последовательности (*для Рутокен перезапуск сервиса SACSrv не требуется, т.к. данный сервис – только для eToken):

```
sudo service SACSrv stop
sudo service pcscd stop
sudo service pcscd start
sudo service SACSrv start
sudo ldconfig
```

Если ранее ключ уже был подключен к серверу и его переподключили, рекомендуется перезагрузить сервер.

8.6.2 Проверка работы Token

Для проверки работы токена можно использовать приложение **pkcs11-tool**. Возможно проверить следующее:

Вывести общую информацию для ключа:

- для eToken

```
pkcs11-tool --module /usr/lib/libeToken.so -I

Cryptoki version 2.1
Manufacturer      SafeNet, Inc.
Library           eToken PKCS#11 (ver 8.1)
Using slot 0 with a present token (0x0)
```

- для Рутокен:

```
pkcs11-tool --module /usr/lib/ecss/ecss-ds/lib/lpm_storage-<VERSION>/priv/x64/
librtpkcs11ecp.so -I
```

```
Cryptoki version 2.20
Manufacturer      Aktiv Co.
Library           Rutoken ECP PKCS #11 library (ver 1.5)
Using slot 0 with a present token (0x0)
```

Отобразить доступные слоты с ключами:

- для eToken


```
pkcs11-tool --module /usr/lib/libeToken.so -L
```


Available slots:

```
Slot 0 (0x0): Aladdin eToken PRO USB 72K Java [Main Interface] 00 00
  token label:   ECSS 000001
  token manuf:   SafeNet Inc.
  token model:   eToken
  token flags:   rng, login required, PIN initialized, token initialized, other
  flags=0x200
  serial num   : 123456789
Slot 1 (0x1):
  (empty)
Slot 2 (0x2):
  (empty)
Slot 3 (0x3):
  (empty)
Slot 4 (0x4):
  (empty)
Slot 5 (0x5):
  (empty)
```

- для Рутокен:

```
pkcs11-tool --module /usr/lib/ecss/ecss-ds/lib/lpm_storage-<VERSION>/priv/x64/
librtpkcs11ecp.so -L
Available slots:
Slot 0 (0x0): Aktiv Rutoken ECP 00 00
    token label          : ECSS 000001
    token manufacturer   : Aktiv Co.
    token model          : Rutoken ECP
    token flags          : rng, login required, PIN initialized, token initialized
    hardware version     : 54.1
    firmware version     : 18.0
    serial num           : 123456789
Slot 1 (0x1):
    (empty)
Slot 2 (0x2):
    (empty)
Slot 3 (0x3):
    (empty)
Slot 4 (0x4):
    (empty)
Slot 5 (0x5):
    (empty)
Slot 6 (0x6):
    (empty)
Slot 7 (0x7):
    (empty)
Slot 8 (0x8):
    (empty)
Slot 9 (0x9):
    (empty)
Slot 10 (0xa):
    (empty)
Slot 11 (0xb):
    (empty)
Slot 12 (0xc):
    (empty)
Slot 13 (0xd):
    (empty)
Slot 14 (0xe):
    (empty)
```

 Расположение модуля для Рутокен может отличаться в зависимости от версии подсистемы DS. В общем случае файл располагается в /usr/lib/ecss/ecss-ds/lib/lpm_storage-<ВЕРСИЯ ПОДСИСТЕМЫ>/priv/x64/librtpkcs11ecp.so
Для проверки можно использовать общую команду **pkcs11-tool --module \$(find /usr/lib/ecss/ecss-ds/lib/ -name librtpkcs11ecp.so | head -n1) -L**

 Если проблемы с определением ключа остаются, то необходимо обратиться в службу технической поддержки.

8.6.3 Перезагрузка Token по SSH в случае его зависания

Для перезагрузки USB-токена необходимо выполнить следующий набор действий:

1. Установить утилиту `usb-reset`:

```
sudo snap install usb-reset
sudo snap connect usb-reset:hardware-observe core:hardware-observe
sudo snap connect usb-reset:raw-usb core:raw-usb
Slot 0 (0x0): Aktiv Rutoken
```

2. Проверить, что USB-токен действительно завис. Пример:

```
pkcs11-tool --module /usr/lib/ecss/ecss-ds/lib/lpm_storage-3.14.8.70203.423017/
priv/x64/librtpkcs11ecp.so -L
```

Вывод должен либо вообще ничего не показывать, либо показать все слоты пустыми.

3. Узнать `idVendor`, `idProduct` USB-токена. Для Рутокен команда будет выглядеть следующим образом:

```
sudo lsusb -v | grep -C 10 "Rutoken ECP"
```

В указанном выводе найти параметры `idVendor`, `idProduct`:

```
lsusb -v | grep -C 10 "Rutoken ECP"
FIXME: alloc bigger buffer for device capability descriptors
  bDescriptorType      1
  bcdUSB                2.00
  bDeviceClass          0 (Defined at Interface level)
  bDeviceSubClass      0
  bDeviceProtocol       0
  bMaxPacketSize0      16
  idVendor              0x0a89
  idProduct             0x0030
  bcdDevice             1.00
  iManufacturer        1 Aktiv
  iProduct              2 Rutoken ECP
  iSerial               0
  bNumConfigurations   1
  Configuration Descriptor:
    bLength              9
    bDescriptorType     2
    wTotalLength         93
    bNumInterfaces      1
    bConfigurationValue 1
    iConfiguration      0
    bmAttributes         0x80
```

4. Перезапустить USB-устройство:

```
sudo usb-reset <idVendor>:<idProduct>
```

Пример:

```
sudo usb-reset 0a89:0030
```

5. Проверить, что слот(ы) появились:

```
pkcs11-tool --module /usr/lib/ecss/ecss-ds/lib/lpm_storage-<VERSION>/priv/x64/  
librtpkcs11ecp.so -L
```

Available slots:

```
Slot 0 (0x0): Aktiv Rutoken ECP 00 00
```

```
...
```

8.6.4 Проблема с работой Token на серверах DEPO

Если периодически фиксируется отключение токенов от серверов DEPO, то следует в syslog проверить наличие ошибок драйвера HNCI. Если ошибки присутствуют, то необходимо зайти в BIOS сервера и включить режим XHCI (путь в BIOS: Advanced/USB Configuration: XHCI Pre-Boot Driver – Enabled, XHCI – enabled).

8.7 Настройка listen-интерфейса для сервиса epmd

Пример настройки listen-интерфейса для сервиса *epmd* в соответствии с конфигурацией сети, приведённой в разделе "[Подготовка сервера для инсталляции ECSS-10](#)".

Для сервера **ecss1** необходимо выполнить следующую последовательность действий:

Выполните команду:

```
sudo systemctl edit epmd.service
```

Откроется окно текстового редактора. Добавить в файл следующую секцию (в соответствии с сетевым интерфейсом, на котором работает epmd: например для ecss1 – 192.168.1.1):

```
[Service]  
Environment="ERL_EPMD_ADDRESS=127.0.1.1,192.168.1.1"
```

После сохранения файла необходимо обязательно перечитать конфигурацию:

```
sudo systemctl daemon-reload
```

Далее требуется перезапустить сервисы:

```
sudo systemctl restart epmd.service
```

Для **ecss2** выполняется такая же последовательность действий, только вместо 192.168.1.1 указывается 192.168.1.2.

❗ В качестве ERL_EPMD_ADDRESS нельзя использовать адреса, которые были сконфигурированы в `keepalived.conf`.

8.8 Запуск и активация системы

❗ **ВАЖНО**
Перед началом работы [проверьте наличие Token в системе](#).

Для запуска и активации системы необходимо выполнить следующую последовательность действий:

Запустить подсистемы *mycelium* и *ds*:

```
sudo systemctl start ecss-mycelium.service  
sudo systemctl start ecss-ds.service
```

Проверить, что сервисы стартовали. Пример для *mycelium*:

```

sasha@ecss1:~$ sudo systemctl status ecss-mycelium.service
[sudo] password for sasha:
● ecss-mycelium.service - daemon ecss-mycelium-14.10.44 of ecss-10
   Loaded: loaded (/lib/systemd/system/ecss-mycelium.service; enabled; vendor preset:
enabled)
   Active: active (running) since Tue 2021-10-12 12:11:48 +07; 21h ago
 Main PID: 22921 (beam.smp)
    Tasks: 29 (limit: 4915)
   CGroup: /ecss.slice/ecss-mycelium.service
           └─22921 ecss-mycelium -pc unicode -e 65536 -- -root /usr/lib/ecss/ecss-
mycelium -progname erl -- -home /var/lib/ecss/home -- -noshell -noinput -mode embedded
-config /tmp/mycelium1.config -boot_v
           └─23023 erl_child_setup 1024
           └─23052 inet_gethost 4
           └─23053 inet_gethost 4
           └─23054 sh -s disksup
           └─23055 /usr/lib/erlang/lib/os_mon-2.4.7/priv/bin/memsup

Oct 12 12:12:27 ecss1 ecss-mycelium[22921]: 12:12:27 I <0.1613.0> rps_tring:50 Alarm
clear: node appear - "md1" - md1@ecss1
Oct 12 12:12:27 ecss1 ecss-mycelium[22921]: 12:12:27 I <0.1615.0> tring_client_em:50
Tring ecss10: "md1" - appear ([{role,mediator},{generation,<<"066203cba3d72206">>}])
Oct 12 12:12:27 ecss1 ecss-mycelium[22921]: 12:12:27 I <0.1615.0> tring_client_em:50
Tring ecss10: "md1"/md1@ecss1 - appear ([{tring_token_nodes,[{"ds1",ds1@ecss1},
{"mycelium-testnew",mycelium1@ecss1}],{ro
Oct 12 12:12:32 ecss1 ecss-mycelium[22921]: 12:12:32 I <0.1615.0> tring_client_em:50
Tring ecss10: "sip1" - appear ([{role,adapter},{generation,<<"066203cbf848a618">>}])
Oct 12 12:12:32 ecss1 ecss-mycelium[22921]: 12:12:32 I <0.1615.0> tring_client_em:50
Tring ecss10: "sip1"/sip1@ecss1 - appear ([{tring_token_nodes,[{"ds1",ds1@ecss1},
{"md1",md1@ecss1},{"mycelium-testnew",my
Oct 12 12:12:35 ecss1 ecss-mycelium[22921]: 12:12:35 I <0.1613.0> rps_tring:50 Alarm
clear: group appear - "core1"
Oct 12 12:12:35 ecss1 ecss-mycelium[22921]: 12:12:35 I <0.1615.0> tring_client_em:50
Tring ecss10: "core1" - appear ([{role,core},{generation,<<"066203cc23ebbd1d">>}])
Oct 12 12:12:35 ecss1 ecss-mycelium[22921]: 12:12:35 I <0.1615.0> tring_client_em:50
Tring ecss10: "core1"/core1@ecss1 - appear ([{tring_token_nodes,[{"mycelium-
testnew",mycelium1@ecss1},{"md1",md1@ecss1},{
Oct 13 09:09:56 ecss1 ecss-mycelium[22921]: 09:09:56 I <0.6428.0> mysql_au:50 Cleaned
0 audit sessions
Oct 13 09:09:56 ecss1 ecss-mycelium[22921]: 09:09:56 I <0.6428.0> mysql_au:50 Cleaned
28 audit commands

```

Пример для ds:

```
sasha@ecss1:~$ sudo systemctl status ecss-ds.service
● ecss-ds.service - daemon ecss-ds-14.10.44 of ecss-10
   Loaded: loaded (/lib/systemd/system/ecss-ds.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Tue 2021-10-12 12:11:59 +07; 21h ago
   Main PID: 23111 (beam.smp)
     Tasks: 38 (limit: 4915)
    CGroup: /ecss.slice/ecss-ds.service
            └─23111 ecss-ds -pc unicode -K true -A 8 -t 2097152 -e 100000 -- -root /
usr/lib/ecss/ecss-ds -progname erl -- -home /var/lib/ecss/home -- -noshell -noinput
-mode embedded -config /tmp/ds1.config
            └─23222 erl_child_setup 1024
            └─23293 inet_gethost 4
            └─23294 inet_gethost 4
            └─23309 inet_gethost 4
            └─23310 inet_gethost 4
            └─23311 sh -s disksup
            └─23312 /usr/lib/erlang/lib/os_mon-2.4.7/priv/bin/memsup


Oct 12 12:11:59 ecss1 systemd[1]: Started daemon ecss-ds-14.10.44 of ecss-10.
Oct 12 12:11:59 ecss1 ecss-ds[23111]: release: ds1 3.14.10.44
Oct 12 12:11:59 ecss1 ecss-ds[23111]: ****
Oct 12 12:11:59 ecss1 ecss-ds[23111]: exec /usr/lib/erlang/erts-10.3.5.10/bin/erlexec
-noinput      -mode embedded      -config /tmp/ds1.config      -env ERL_CRASH_DUMP /var/
log/ecss/ds/crashdumps/erl_cras
Oct 12 12:12:00 ecss1 ecss-ds[23111]: Starting Chronica...
Oct 12 12:12:00 ecss1 ecss-ds[23111]: Starting error_logger...
Oct 12 12:12:01 ecss1 ecss-ds[23111]: ok
Oct 12 12:12:01 ecss1 ecss-ds[23111]: starting OASYS {check boot
env.}          ...done
Oct 12 12:12:01 ecss1 ecss-ds[23111]: starting OASYS {data control
engine}        ...done
Oct 12 12:12:20 ecss1 systemd[1]: ecss-ds.service: Current command vanished from the
unit file, execution of the command list won't be resumed.
```

Подключиться к распределенной консоли управления CoCon:

```
ssh admin@localhost -p8023
```

 Пароль по умолчанию для подключения к консоли управления CoCoN – **password**.

Далее установить паспорт и лицензию. Процесс установки [лицензионных ограничений](#) включает в себя занесение в базу данных ECSS-10 кодовой последовательности лицензии и паспорта USB-ключа Token.

 **Важно**
Сначала устанавливается паспорт, затем лицензия.

Введите данные паспорта и лицензии (описание и примеры команд приведены в [справочнике](#)):

```
/cluster/storage/<CLUSTER>/licence/set-passport <PASSPORT>
```

```
/cluster/storage/<CLUSTER>/licence/add [--force|--no-diff] <LICENCE>
```

где:

- <CLUSTER> – имя кластера хранения долговременных данных (ds);

 По умолчанию в системе присутствует кластер хранения долговременных данных с именем ds1.

- <PASSPORT> – последовательность цифр, букв и других символов в файле паспорта (ECSS xxxxxx.pas);
- <LICENCE> – последовательность цифр, букв и других символов в файле лицензии (ECSS xxxxxx уууу-мм-дд.lic);
- [--force] – пропустить утверждение команды;
- [--no-diff] – не выводить таблицу сравнения текущих и предлагаемых условий лицензии.

Если данные лицензии и паспорта будут введены корректно, то система выдаст подтверждение: **OK**.

Сразу можно проверить, что применились правильные лицензионные ограничения:

```
/cluster/storage/<CLUSTER>/licence/current-limits
```

Далее необходимо отключиться от консоли CoCon, выполнив команду:

```
exit
```

Запустите остальные ноды:

```
sudo systemctl start ecss-mycelium.service
sudo systemctl start ecss-ds.service
sudo systemctl start ecss-core.service
sudo systemctl start ecss-pa-sip.service
sudo systemctl start ecss-mediator.service
sudo systemctl start ecss-restfs.service
sudo systemctl start ecss-web-conf.service
sudo systemctl start ecss-media-server.service
sudo systemctl start ecss-cc-ui.service
sudo systemctl start ecss-teleconference-ui
```

Также запустите **ecss-pa-megaco** (в случае, если он используется):

```
sudo systemctl start ecss-pa-megaco.service
```

Проверьте, что сервисы корректно запустились, выполнив для каждой команды:

```
sudo systemctl status <SERVICE>
```

где <SERVICE> – имя сервиса. В статусе должно отображаться состояние **active**.

На данном этапе система считается полностью установленной и готовой к настройке.


8.9 Особенности установки системы в кластере

8.9.1 Установка ECSS-10 в кластере

Подготовка хостов

При установке системы ECSS-10 в кластере необходимо на обоих серверах в соответствии с проектом выполнить:

- [Инсталляцию операционной системы](#);
- [Настройку сети](#);
- [Установку необходимого ПО](#);
- [Установку пакетов ECSS](#);

 Пакет `ecss-web-conf` достаточно установить на один из серверов.

- [Настройку NTP](#);
- [Настройку EPMD](#);
- [Проверку Token](#).

Установка имени кластера

Для работы системы нужно на обоих серверах указать одинаковое имя кластера. Для этого откройте файл **`mycelium1.config`** в текстовом редакторе:

```
sudo nano /etc/ecss/ecss-mycelium/mycelium1.config
```


Если в поле "cluster_name" указано "undefined", то необходимо задать произвольное имя для данного параметра, например:

```
{cluster_name, my_cluster}
```

Далее следует проверить в файле `/etc/dnsmasq.d/ecss-broker`, что адреса **primary** и **secondary broker** соответствуют [указанным](#) при инсталляции пакета **ecss-node**.

Пример содержания файла на **ecss1** и **ecss2** (содержимое файлов должно быть одинаковым на обоих серверах):

```
address=/primary.broker.ecss/192.168.1.1  
address=/secondary.broker.ecss/192.168.1.2
```

 В качестве `primary.broker.ecss` и `secondary.broker.ecss` **нельзя** использовать адреса, которые были сконфигурированы в `keepalived.conf`.

Настройка RestFS для кластера

Для работы в кластере необходимо настроить работу **RestFS** на базе **GlusterFS-сервера**.

Для того чтобы обеспечить репликацию данных между серверами кластера, необходимо настроить `glusterfs-server`.

В качестве примера приведена система ECSS-10, работающая в кластере, со следующими настройками:

- IP-адрес `ecss1` — 192.168.118.222;
- IP-адрес `ecss2` — 192.168.118.224.

1. Установите `gluster`-сервер и пакет `attr` на оба хоста:

```
sudo aptitude install glusterfs-server attr
```

2. Для добавления сервера в пул файловых хранилищ выполните команду на `ecss1`:

```
sudo gluster peer probe 192.168.118.224
```

После этого на `ecss2` при выполнении команды `sudo gluster peer status` должна появиться информация о `ecss1`:

```
Number of Peers: 1 Hostname: 192.168.118.222 Uuid: 569c4730-a3a7-4d29-a132-b1bcdad792d8 State: Peer in Cluster (Connected)
```

3. Для создания кластера на `ecss1` выполните команду:

```
sudo gluster volume create ecss_volume replica 2 transport tcp 192.168.118.222:/var/lib/ecss/glusterfs 192.168.118.224:/var/lib/ecss/glusterfs force
```

4. Запустите созданный кластер, для этого на `ecss1` выполните команду:

```
sudo gluster volume start ecss_volume
```

5. Для проверки статуса кластера на `ecss1` выполните команду:

```
sudo gluster volume info
```

Необходимо обратить внимание на поля "Status" и "Bricks" — они должны иметь следующий вид:

```
Volume Name: ecss_volume
Type: Replicate
Volume ID: 60774e49-d2f1-4b06-bb4a-3f39ccf1ea73
Status: Started Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: 192.168.118.222:/restfs
Brick2: 192.168.118.224:/restfs
```


6. Чтобы смонтировать glusterfs раздел, выполните на обоих хостах ecss1 и ecss2 следующие действия:

- Создайте новый systemd unit:

```
/etc/systemd/system/ecss-glusterfs-mount.service
```

и добавьте туда следующие параметры:

```
[Unit]
Description=mount glusterfs
After=network.target
Requires=network.target

[Service]
RemainAfterExit=no
Type=forking
RestartSec=10s
Restart=always
ExecStart=/sbin/mount.glusterfs localhost:/ecss_volume /var/lib/ecss/restfs
-o fetch-attempts=10
ExecStop=/bin/umount /var/lib/ecss/restfs

[Install]
WantedBy=multi-user.target
```

- Добавить unit в автозагрузку.
Unit добавить в автозагрузку следующей командой:

```
sudo systemctl enable ecss-glusterfs-mount.service
```

- Перезагрузить хост:

```
sudo reboot
```

Если хост не может быть перезагружен, то можно выполнить следующие команды:

```
sudo systemctl daemon-reload
sudo systemctl restart ecss-glusterfs-mount.service
```

После монтирования на обоих хостах выполните команду:

```
df -h
```

При просмотре информации должен появиться подмонтированный раздел:

/dev/sda10	19G	6,5G	11G	38%	/var/lib/mysql
/dev/sda8	4,5G	213M	4,1G	5%	/var/log
/dev/sda5	37G	48M	35G	1%	/var/lib/ecss/ecss-media-server/records
/dev/sda6	19G	44M	18G	1%	/var/lib/ecss/cdr
/dev/sda7	19G	44M	18G	1%	/var/lib/ecss/statistics
/dev/sda9	19G	7,6G	9,7G	44%	/var/log/ecss
localhost:/ecss_volume	46G	59M	44G	1%	/var/lib/ecss/restfs*

Запуск RestFS в режиме кластера

Для запуска RestFS в режиме кластера достаточно, чтобы пакет `ecss-restfs` был установлен и запущен сразу на двух нодах. Команда для запуска сервиса `ecss-restfs`:

```
sudo systemctl start ecss-restfs.service
```

Запуск RestFS в случае недоступности других участников кластера

В применяемой концепции `glusterfs` все сервера равнозначны. Однако раздел `volume` не активируется при отсутствии кворума (кластер имеет кворум при наличии и активности достаточного количества голосующих элементов кластера с одинаковым согласованным представлением кластера). Это защитный механизм, который характерен для всех распределенных отказоустойчивых систем и призван защитить систему от `split-brain` — ситуация, когда каждая система считает, что другая нода вышла из строя.

Такая ситуация может возникнуть когда загружается только один из серверов, а второй при этом выключен или недоступен. На первом сервере `volume` не будет автоматически активирован до появления второго сервера для исключения расхождения данных.

Если включение второго сервера невозможно либо затягивается на длительное время, то можно вручную перевести `volume` в рабочий режим, выполнив команду:

```
sudo gluster volume ecss_volume start force
```

Проблемы связанные с возникновением `split-brain`

В случае недоступности одной из нод кластера могут возникнуть проблемы с файлами. После восстановления работы они будут находиться в сплите, из-за чего собственноручно придется запускать синхронизацию между нодами.

Для того чтобы решить данную проблему, потребуется воспользоваться ключом `cluster.favorite-child-policy`. При его включении все файлы, находящиеся в сплите, будут автоматически синхронизированы между собой по заданному правилу.

Включение данного параметра производится командой:

```
sudo gluster volume set ecss_volume cluster.favorite-child-policy size
```

Правка настроек glusterfs-server.service unit

Для настройки управления glusterfs service через systemctl выполните команду:

```
sudo systemctl edit glusterfs-server.service
```

Должно открыться окно текстового редактора. Внесите туда следующие параметры:

```
[Service]
KillMode=control-group
RemainAfterExit=no
```

Сохраните изменения и выполните команду:

```
sudo systemctl daemon-reload
```

Инсталляция и настройка snmpd

Установите Net-SNMP агент:

```
sudo aptitude install snmpd
```

Стандартный порт, который использует snmpd – udp/161. Встроенный агент ECSS использует по умолчанию порт udp/1610, поэтому конфликтовать с snmpd они не будут.

Убедитесь, что сервис snmpd успешно запустился на порту 161, а ECSS – на порту 1610:

```
sudo netstat -tulpan | grep 161
udp        0      0 127.0.0.1:161          0.0.0.0:*                7723/
snmpd
udp        0      0 0.0.0.0:1610         0.0.0.0:*                8245/
ecss-mediator
```

Если нужно изменить стандартный порт snmpd, отредактируйте файл конфигурации /etc/snmp/snmpd.conf. Например:

```
agentAddress udp:127.0.0.1:3161
```

Сохраните внесенные изменения.

Далее перезапустите сервис snmpd:

```
sudo systemctl restart snmpd.service
```

Убедитесь, что сервис snmpd успешно запустился на новом порту:

```
sudo netstat -tulpan | grep snmpd
udp        0          0 127.0.0.1:3161      0.0.0.0:*
snmpd
```

Настройка VRRP

Настройка демона `keepalived` для управления виртуальными адресами

Одним из способов повысить отказоустойчивость ECSS-10 является использование виртуальных IP-адресов. Виртуальный IP-адрес не принадлежит постоянно конкретной ноде кластера ECSS-10, а автоматически поднимается на той ноде, которая в данный момент времени способна обслуживать запросы. Таким образом, получаем:

- Независимость конфигурации от IP-адресов конкретных нод кластера. Нет необходимости перечислять на встречном оборудовании все возможные адреса нод ECSS-10 — достаточно указать один виртуальный IP-адрес, и запрос будет обслужен любой нодой кластера, которая в текущий момент способна его обработать.
- Возможность работать с оборудованием, которое не поддерживает указания нескольких адресов для взаимодействия. Для такого оборудования весь кластер ECSS-10 будет представлен одним виртуальным IP-адресом.
- Повышение отказоустойчивости. В случае отказа одной из нод кластера другая нода получит виртуальный IP-адрес и будет предоставлять сервис взамен отказавшей.

Для управления виртуальными адресами используется демон `keepalived`, который реализует функции:

- Мониторинга доступности нод ECSS-10.
- Выбор активной (master) ноды по протоколу `VRRP` (Virtual Router Redundancy Protocol, RFC3768/ RFC5798) на основе доступности нод.
- Перенос виртуального IP-адреса на активную ноду.

Общая настройка `keepalived`

Для работы рекомендуется использовать протокол VRRP версии 3, так как он обеспечивает меньшую задержку перед переносом адреса в случае потери текущей активной ноды. При использовании на сети протокола IPNET протокол VRRP версии 3 следует использовать обязательно. Для обеспечения оперативного переключения между рабочими нодами следует использовать именно протокол VRRP версии 3, так как он позволяет осуществлять рассылку сообщений VRRP advertisements с интервалом в 1/100 секунды (сантисекунда), в отличие от протокола VRRP версии 2, который оперирует секундными интервалами. Тем не менее, VRRP версии 2 также работоспособен в 3.14 версии ECSS-10. Версию 3 протокола VRRP в конфигурационном файле нужно задавать явно, по умолчанию используется версия 2:

```
man keepalived

# Set the default VRRP version to use
vrrp_version <2 or 3>          # default version 2
```

Также рекомендуется сконфигурировать запуск скриптов проверки от пользователя `nobody` (системный пользователь без прав) и включить безопасный запуск скриптов, которые запускаются от пользователя `root`.

После определения глобальных опций демона следует через опцию `include` подключить файлы с конфигурацией виртуальных адресов. В конфигурации `keepalived` допускается оставление

комментариев. Комментарии располагаются в любой части конфигурации, начинаются с символа # и заканчиваются концом строки.

⚠ Примечание. Можно встретить множество примеров, в которых при настройке VRRP используется опция *authentication*. Однако в документации *keepalived* упоминается о том, что *authentication* была удалена из VRRPv2 в спецификации RFC3768 (см. <https://tools.ietf.org/html/rfc3768>) в 2004 году, так как не обеспечивала безопасности и могла приводить к появлению двух "мастеров". Рекомендуется избегать использования этой опции (в VRRP_v3 отключена).

Основная конфигурация демона хранится в файле **/etc/keepalived/keepalived.conf**.

Основная конфигурация (одинаковая для всех нод кластера):

```
global_defs {
    vrrp_version 3          # версия протокола VRRP (2 или 3)
    script_user nobody     # ограниченный в правах системный пользователь, от которого будут
запускаться скрипты проверки доступности
    enable_script_security # не запускать скрипты от root, если часть пути к ним доступна на
запись для обычных пользователей
}

include /etc/keepalived/sip.conf
include /etc/keepalived/mysql.conf
include /etc/keepalived/ipnet.conf
```

Настройка виртуального адреса для SIP-адаптера

В приведённой схеме используются два виртуальных адреса для SIP-адаптеров. Это позволяет распределить нагрузку между нодами, конфигурируя встречные устройства таким образом, чтобы часть из них работала с одним виртуальным адресом, а часть с другим. При условии неполной загрузки нод сохраняется отказоустойчивость, то есть, в случае отказа одной из нод виртуальный адрес будет подхвачен другой нодой.

Конфигурация строится таким образом, чтобы первая нода была мастером для первого виртуального адреса SIP-адаптера. Вторая нода будет резервировать этот адрес. Конфигурация для основного адреса SIP-адаптера второй ноды настраивается зеркально: вторая нода — мастер, первая нода — резерв. Конфигурацию виртуальных адресов для SIP-адаптера рекомендуется поместить в отдельный файл — **/etc/keepalived/pa-sip.conf**.

⚠ Изменён скрипт проверки доступности контрольного SIP-порта. Теперь скрипт из *keepalive* нужно вызывать следующим образом: **/usr/bin/ecss_pa_sip_port 65535**, где <65535> — значение по умолчанию порта, который адаптер открывает, когда он готов принимать нагрузку. Для того чтобы изменить порт, необходимо в конфигурационном файле SIP-адаптера **/etc/ecss/ecss_pa_sip/sip1.config** в секции **ip_ssw_intercom** изменить в параметре *keepalive* значение порта, после чего перезапустить адаптер.

Конфигурация первой ноды:

```
vrrp_script check_sip {
    script "/usr/bin/ecss_pa_sip_port 65535"
    interval 2
    timeout 2
}
```

```

# Конфигурация адреса для первого виртуального адреса SIP-адаптера
vrrp_instance SIP1 {
    state MASTER # Исходное состояние при старте
    interface <network_interface> # Имя сетевого интерфейса, на котором будет работать
    протокол VRRP
    virtual_router_id <ID> # Уникальный идентификатор роутера (0..255)
    priority 100 # Приоритет (0..255) чем выше – тем больше
    advert_int 1 # Интервал рассылки уведомлений (с)
    preempt_delay 60 # Интервал ожидания мастера при старте демона (с)
    при исходном состоянии BACKUP

    unicast_src_ip <src_real IP> # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP> # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
        # label – метка виртуального интерфейса (для удобства идентификации)
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }

    track_script {
        check_sip
    }
}

# Конфигурация для второго виртуального адреса SIP-адаптера

vrrp_instance SIP2 {
    state BACKUP # Исходное состояние при старте
    interface <network_interface> # Имя сетевого интерфейса, на котором будет
    работать протокол VRRP
    virtual_router_id <ID> # Уникальный идентификатор роутера (0..255)
    priority 50 # Приоритет (0..255) чем выше – тем больше
    advert_int 1 # Интервал рассылки уведомлений (с)
    preempt_delay 60 # Интервал ожидания мастера при старте демона
    (с) при исходном состоянии BACKUP

    unicast_src_ip <src_real IP> # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP> # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
        # label – метка виртуального интерфейса (для удобства идентификации)
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }

    track_script {
        check_sip
    }
}

```

```
}
```

Конфигурация второй ноды:

```
vrrp_script check_sip {
    script "/usr/bin/ecss_pa_sip_port 65535"
    interval 2
    timeout 2
}

# Конфигурация для первого виртуального адреса SIP-адаптера
vrrp_instance SIP1 {
    state BACKUP # Исходное состояние при старте
    interface <network_interface> # Имя сетевого интерфейса, на котором будет работать
    протокол VRRP
    virtual_router_id <ID> # Уникальный идентификатор роутера (0..255)
    priority 50 # Приоритет (0..255) чем выше – тем больше
    advert_int 1 # Интервал рассылки уведомлений (с)
    preempt_delay 60 # Интервал ожидания мастера при старте демона (с)
    при исходном состоянии BACKUP

    unicast_src_ip <src_real IP> # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP> # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
        # label – метка виртуального интерфейса (для удобства идентификации)
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }
}

# Конфигурация для второго виртуального адреса SIP-адаптера

vrrp_instance SIP2 {
    state MASTER # Исходное состояние при старте
    interface <network_interface> # Имя сетевого интерфейса, на котором будет
    работать протокол VRRP
    virtual_router_id <ID> # Уникальный идентификатор роутера (0..255)
    priority 100 # Приоритет (0..255) чем выше – тем больше
    advert_int 1 # Интервал рассылки уведомлений (с)
    preempt_delay 60 # Интервал ожидания мастера при старте демона
    (с) при исходном состоянии BACKUP

    unicast_src_ip <src_real IP> # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP> # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
```

```
    # label – метка виртуального интерфейса (для удобства идентификации)
    <virtual_sip_IP>/<netmask> dev <> label <label>
}

track_script {
    check_sip
}
}
```

Настройка виртуального адреса для MySQL

Для отказоустойчивости в кластере ECSS-10 используется режим репликации MySQL мастер-мастер. Это позволяет корректно переносить данные в любом направлении. Однако одновременная запись на оба сервера MySQL с одновременной репликацией во встречном направлении повышает вероятность коллизий, что снижает отказоустойчивость. Поэтому рекомендуется настроить выделенный виртуальный адрес для кластера MySQL, чтобы запись данных в один момент времени производилась на одну ноду.

⚠ Если создавать файлы **/etc/keepalived/mysql.conf** вручную, то при запуске скрипта создания репликации нужно отказаться от автоматической настройки при вопросе **"Нужна ли настройка keepalive? ("DO YOU WANT TO SET REST OF keepalive CONFIG")**.

Конфигурацию виртуальных адресов для MySQL рекомендуется поместить в отдельный файл **/etc/keepalived/mysql.conf**.


```

# Конфигурация mysql первой ноды:

vrrp_script check_mysql {
    script "/usr/bin/mysql --defaults-file=/etc/mysql/debian.cnf -e 'SELECT 1;'"
    user root
    interval 2
    fall 1
    timeout 2
}

vrrp_instance MySQL {
    state MASTER # Исходное состояние при старте
    interface <network_interface> # Имя сетевого интерфейса, на котором будет
работать протокол VRRP
    virtual_router_id <ID> # Уникальный идентификатор роутера (0..255)
    priority 100 # Приоритет (0..255) чем выше – тем больше
    advert_int 1 # Интервал рассылки уведомлений (с)
    preempt_delay 60 # Интервал ожидания мастера при старте демона (с)
при исходном состоянии BACKUP

    unicast_src_ip <src_real IP> # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP> # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
        # label – метка виртуального интерфейса (для удобства идентификации)
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }

    track_script {
        check_mysql
    }
}

```

```

# Конфигурация mysql второй ноды:

vrrp_script check_mysql {
    script "/usr/bin/mysql --defaults-file=/etc/mysql/debian.cnf -e 'SELECT 1;'"
    user root
    interval 2
    fall 1
    timeout 2
}

vrrp_instance MySQL {
    state BACKUP
    interface <network_interface> # Имя сетевого интерфейса, на котором будет
    работать протокол VRRP
    virtual_router_id <ID> # Уникальный идентификатор роутера (0..255)
    priority 50 # Приоритет (0..255) чем выше – тем больше
    advert_int 1 # Интервал рассылки уведомлений (с)
    preempt_delay 60 # Интервал ожидания мастера при старте демона (с)
    при исходном состоянии BACKUP

    unicast_src_ip <src_real IP> # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP> # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
        # label – метка виртуального интерфейса (для удобства идентификации)
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }

    track_script {
        check_mysql
    }
}

```

Настройка репликации БД MySQL приведена в разделе ["Схема развертывания MySQL master-master replication с использованием keepalive"](#).

Пример создания типовой конфигурации приведен в разделе ["Примеры пошаговой первоначальной настройки ECSS-10"](#).

Настройка виртуального адреса для IPNET

Поскольку через IPNET не поддерживается работа с несколькими адресами встречной стороны, при работе ECSS-10 в кластере требуется выделить виртуальный IP-адрес.

Для обеспечения оперативного переключения между рабочими нодами следует использовать протокол VRRP версии 3, так как он позволяет осуществлять рассылку сообщений VRRP advertisements с интервалом в 1/100 секунды (сантисекунда) в отличие от протокола VRRP версии 2, который оперирует секундными интервалами. С точки зрения протокола IPNET это важно, так как протокол IPNET реализует свои собственные *keepalive*-сообщения. При использовании протокола VRRP версии 2 худшее время переключения виртуального IP-адреса составит 4 секунды при минимально допустимом по протоколу

времени рассылки VRRP advertisements в одну секунду, что может быть недопустимо долго с точки зрения механизма IPNET keepalive и приведёт к разрушению вызова со стороны встречной станции.

В предлагаемой конфигурации обмен VRRP advertisements между нодами происходит каждые 50 мс. Интервал VRRP advertisements следует выбирать исходя из величины сетевой задержки между нодами. Выбранный интервал в 50 мс позволяет оперативно переключаться при сбое нод, а также пережить без ложного срабатывания возрастание сетевой задержки до 150-200 мс. В случае, если ноды сильно разнесены географически, может потребоваться увеличить этот интервал, исходя из реальных характеристик сети. Однако делать слишком большим интервал не следует, так как это может повлиять на стабильность сохранения активных вызовов при переключении адреса на резерв. Худшее время переключения при отказе мастера или потере пакетов VRRP advertisements в случае проблем на сети составит $\text{advert_int} \times 4$.

Конфигурацию виртуальных адресов для IPNET рекомендуется поместить в отдельный файл **/etc/keepalived/ipnet.conf**.

```
# Конфигурация первой ноды:

vrrp_script check_ipnet {
    script "/usr/bin/ecss_ipnet_port 65531"
    interval 1
    fall 1
    rise 1
}

vrrp_instance IPNET {
    state MASTER # Исходное состояние при старте
    interface <network_interface> # Имя сетевого интерфейса, на котором будет
    работать протокол VRRP
    virtual_router_id <ID> # Уникальный идентификатор роутера (0..255)
    priority 100 # Приоритет (0..255) чем выше – тем больше
    advert_int 0.05 # Интервал рассылки уведомлений (с)
    preempt_delay 60 # Интервал ожидания мастера при старте демона (с)
    при исходном состоянии BACKUP

    unicast_src_ip <src_real IP> # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP> # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
        # label – метка виртуального интерфейса (для удобства идентификации)
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }

    track_script {
        check_ipnet
    }
}
```

```

# Конфигурация второй ноды

vrrp_script check_ipnet {
    script "/usr/bin/ecss_ipnet_port 65531"
    interval 1
    fall 1
    rise 1
}

vrrp_instance IPNET {
    state BACKUP
    interface <network_interface>
    virtual_router_id <ID>
    priority 50
    advert_int 0.05
    preempt_delay 60

    unicast_src_ip <src_real IP>
    unicast_peer {
        <real_remote IP>
    }

    virtual_ipaddress {
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }

    track_script {
        check_ipnet
    }
}

```

Более подробная информация о приложении *keepalived* и его настройке приведена [здесь](#).

Запуск системы

После того как все настроено, можно приступать к [запуску и активации](#) системы.

На **ecss1**:

- Установить паспорт, лицензию и запуск сервисов **ecss**;
- Проверить статус сервисов;
- Проверить доступность подсистем по DNS-именам;
- Проверить статус нод из CoCon (**node/check-services**).

После того как на **ecss1** успешно запустились подсистемы и была активирована лицензия, можно запускать все сервисы **ecss2**.

Проверка установки и вхождения системы в кластер

Для проверки статуса работы нод в кластере нужно войти в [командную консоль \(CoCon\)](#) на любом из серверов:

```
ssh admin@<IP_ECSS> -p8023
```

где <IP_ECSS> – IP-адрес или доменное имя сервера ecss.

Пароль по умолчанию – **password**. После входа введите команду `node/check-services`. Должны отобразиться ноды обоих серверов.

Пример:

```
admin@mycelium1@ecss1:/$ node/check-services
Nodes:
  core1@ecss1      core1@ecss2
  ds1@ecss1       ds1@ecss2
  md1@ecss1       md1@ecss2
mycelium1@ecss1  mycelium1@ecss2
  sip1@ecss1      sip1@ecss2

All services are started
```

Также необходимо проверить, что ноды "видят" друг друга, командой `node/nodes-info`, пример:

```
admin@mycelium1@ecss1:/$ node/nodes-info
```

Node	Erlang	Mnesia
core1@ecss1	core1@ecss1, core1@ecss2	not running
core1@ecss2	core1@ecss1, core1@ecss2	not running
ds1@ecss1	ds1@ecss1, ds1@ecss2	ds1@ecss1, ds1@ecss2
ds1@ecss2	ds1@ecss1, ds1@ecss2	ds1@ecss1, ds1@ecss2
md1@ecss1	md1@ecss1, md1@ecss2	md1@ecss1, md1@ecss2
md1@ecss2	md1@ecss1, md1@ecss2	md1@ecss1, md1@ecss2
mycelium1@ecss1	mycelium1@ecss1, mycelium1@ecss2	not running
mycelium1@ecss2	mycelium1@ecss1, mycelium1@ecss2	not running
sip1@ecss1	sip1@ecss1, sip1@ecss2	sip1@ecss1, sip1@ecss2
sip1@ecss2	sip1@ecss1, sip1@ecss2	sip1@ecss1, sip1@ecss2

На этом этап инсталляции закончен. После [проверки](#) можно приступить к [настройке](#).

8.9.2 Вывод из обслуживания одного сервера

Если по каким-то причинам требуется вывести из обслуживания первый сервер кластера **ecss1**, то на втором сервере **ecss2** необходимо выполнить следующее:

Открыть файл `/etc/hosts`:

```
sudo nano /etc/hosts
```

Назначить для хоста **ecss1** адрес, соответствующий хосту **ecss2**:

```
127.0.0.1    localhost
127.0.1.1    ecss2
192.168.1.2  ecss1

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

Также данное действие можно произвести при помощи утилиты [ecss-control](#).

8.10 Проверка корректности инсталляционных процедур

После выполнения всех инсталляционных процедур следует проверить правильность и полноту выполненных действий. Для этого используется чек-лист, приведенный в разделе "[Чек-лист по установке ECSS-10](#)".

9 Начальная настройка конфигурации

- Понятия и определения
- Предварительные условия
- Порядок начальной настройки конфигурации
 - Конфигурирование кластеров
 - Общие команды для конфигурирования свойств кластера
 - Настройки кластера хранения данных STORAGE
 - Настройка через CLI (CoCon)
 - Настройка параметров кластера через web-интерфейс
 - Настройка параметров кластера MEDIATOR
 - Настройка через CLI (CoCon)
 - Настройка параметров кластера через web-интерфейс
 - Настройка параметров кластера CORE
 - Настройка через CLI (CoCon)
 - Настройка параметров кластера через web-интерфейс
 - Настройка параметров кластеров PA_SIP
 - Настройка через CLI (CoCon)
 - Настройка параметров кластера через web-интерфейс
 - Настройка программного медиасервера
 - Создание и настройка доменов (виртуальных АТС)
 - Декларация доменов
 - Настройка маршрутизации
 - Создание контекстов маршрутизации
 - Создание контекстов модификации и адаптации
 - Применение контекстов маршрутизации для системных интерфейсов
 - Настройка CDR
 - Добавление и настройка прав пользователей
 - Настройка правил ограничения обслуживания абонентов
 - Создание и настройка абонентов
 - Создание и настройка транков
 - Создание и настройка bridge-интерфейсов
 - Настройка ограничений
 - Сценарии IVR
 - Настройка дополнительных сервисов
 - Настройка взаимодействия с серверами RADIUS
 - Настройка функции COPM
 - Настройка дополнительных приложений
 - Настройка ECSS-10 для производительных систем
 - Выделение отдельных ядер процессора для MSR
 - Установка `scaling_governor` в режим `performance`
 - Запуск MSR на изолированных ядрах процессора
 - Настройка использования определенных ядер процессора для `erlang-based` служб

В данном разделе приведено описание этапов первоначальной настройки ECSS перед проверкой базового функционала.

9.1 Понятия и определения

- **Алиас** — совокупность данных об абоненте;
- **Бридж** — виртуальный шлюз, объединяющий связь между виртуальными АТС. Понятие «бридж» было введено для создания средств контроля соединениями между виртуальными АТС. Вызовы между виртуальными АТС одной системы ECSS-10 маршрутизируются в рамках данной системы через бридж. При этом не задействованы межстанционные соединительные линии. Бридж представлен в виде двух связанных друг с другом интерфейсов. Каждый интерфейс декларирован

в своей виртуальной АТС. Для бриджа, как и для классического транка, могут быть заданы различные типы ограничений, например, количество каналов, что дает ограничение на количество одновременно установленных соединений между виртуальными АТС и позволяет нормировать нагрузку;

- **Домен (виртуальная АТС)** — совокупность, состоящая из множества контекстов маршрутизации, интерфейсов и алиасов. Ближайший эквивалент — описание плана нумерации и маршрутизации в рамках классической телефонной станции для традиционных сетей;
- **Кластер** — совокупность элементов одного типа, выполняющих, с точки зрения системы, единую функцию. С их помощью описывается вычислительная топология системы. В нашей системе элементом кластера является **нода**. Кластер существует до тех пор, пока в его состав входит хотя бы одна нода;
- **Медиаресурсы** — описание параметров медиасервера, необходимых для работы с ним;
- **Медиасервер (MSR)** — компонент системы ECSS-10, предназначенный для проксирования речевой и видеоинформации по протоколу RTP, организации конференций, записи разговоров, воспроизведения медиафайлов и различных комбинаций этих режимов. Управление ресурсами медиасервера осуществляется с помощью механизма *control channel* ([RFC 6230 Media Control Channel Framework](#), [RFC 6231 IVR Control Package](#), [RFC 6505 Mixer Control Package](#));
- **Нода** — представляет собой виртуальную машину Erlang и является элементом вычислительного кластера ECSS-10. Ноды в ECSS-10 типизируются по выполняемому на них функционалу. Однотипные ноды объединяются в кластеры соответствующего типа. Пример: кластер Core состоит из нод, выполняющих функцию ядра коммутационной системы;
- **СОРМ** — система технических средств для обеспечения функций оперативно-розыскных мероприятий;
- **IVR** (англ. **Interactive Voice Response**), интерактивное голосовое меню — система предварительно записанных голосовых сообщений, выполняющая функцию маршрутизации звонков внутри call-центра или УПАТС с использованием информации, вводимой клиентом на клавиатуре телефона с помощью тонального набора;
- **LDAP** (англ. **Lightweight Directory Access Protocol** — «легковесный протокол доступа к каталогам») — протокол прикладного уровня для доступа к службе каталогов;
- **RADIUS** — протокол, который предоставляет централизованный метод аутентификации пользователей путем обращения к внешнему серверу. Протокол RADIUS используется для аутентификации, авторизации и учета. Сервер RADIUS использует базу данных пользователей, которая содержит данные проверки подлинности для каждого пользователя. Таким образом, использование протокола RADIUS обеспечивает централизованное управление и дополнительную защиту при доступе к ресурсам сети.

9.2 Предварительные условия

Перед началом настройки конфигурации необходимо убедиться в следующем:

- [установлена операционная система](#);
- [настроен доступ в сеть Интернет](#);
- [установлено необходимое ПО](#);
- [установлены необходимые компоненты ECSS](#);
- [настроен сервис NTP](#);
- [подключен и работает Token](#);
- [все подсистемы ECSS запущены и работают](#);
- [установлены корректные паспорт и лицензия](#);
- если система в [кластере](#), то дополнительно:
 - [настроен и корректно работает keepalived](#);
 - [работает репликация БД MySQL](#);
 - [ноды в работе и "видят" друг друга](#);
 - [настроен и работает GlusterFS](#).

9.3 Порядок начальной настройки конфигурации

В первую очередь необходимо выполнить общие настройки для всей системы:

- кластеры;
- медиасервер;
- маршрутизация, модификация и адаптация;
- пользователи ECSS;
- опционально:
 - COPM;
 - RADIUS;
 - LDAP.

Далее необходимо задекларировать и настроить следующие сервисы:

- домены;
- транки;
- бриджи;
- IVR;
- абоненты;
- услуги на транках и абонентах.

В последнюю очередь настраиваются [дополнительные сервисы](#).

Рекомендуется проводить первоначальную настройку в порядке, приведенном ниже.


9.3.1 Конфигурирование кластеров

Классификация кластеров по ролям:

- BUS – кластер интеграционной шины, обеспечивающий надежную передачу сообщений;
- CORE – кластер, выполняющий функции маршрутизации телефонных вызовов и обработки услуг;
- STORAGE – кластер хранения долговременных данных;
- MEDIATOR – кластер, обеспечивающий функции управления комплексом, предоставление статистической информации и аварийной сигнализации;
- ADAPTER – кластер, выполняющий функции взаимодействия со шлюзами, работающими по одному из протоколов: H.248/Megaco, SIP и SIP-T, PA MGCP, PA Sigtran.

Более подробное описание каждого кластера приведено в разделе "[Архитектура и принципы работы системы](#)".

Данный этап включает в себя настройку всех кластеров системы (core, ds, mediator, pa_sip, pa_megaco). Каждый кластер может включать в себя одну или несколько нод одного типа. Например, кластер SIP-адаптера (pa_sip) может состоять из нескольких нод SIP-адаптеров.

-  При установке лицензии автоматически задается стандартная топология подсистемы:
- устанавливается определенный набор кластеров со стандартными именами;
 - задается определенный набор нод со стандартными именами в кластерах.

Для управления кластерами системы могут использоваться:

- интерфейс командой строки **CLI** (описание команд, предназначенных для управления кластерами приведены в разделе [/cluster/](#));
- приложение [Кластеры \(Clusters\)](#) в **web-конфигураторе**.

Общие команды для конфигурирования свойств кластера

Данные команды являются основными для задания настроек любого кластера.

Для изменения индивидуальных настроек определенного кластера используется команда:

```
/cluster/<SOME_ROLE>/<NAME_CLUSTER>/<GROUP>/set <PROPERTY> [<NAME_NODE>|add|remove] <VALUE>
```

Для просмотра индивидуальных настроек определенного кластера используется команда:

```
/cluster/<SOME_ROLE>/<NAME_CLUSTER>/<GROUP>/info [<PROPERTY>]
```

где:

- <SOME_ROLE> – роль кластера: adapter, bus, core, mediator, storage;
- <NAME_CLUSTER> – имя кластера;
- <GROUP> – группа параметров;
- <NAME_NODE> – имя ноды;
- <PROPERTY> – имя свойства;
- <VALUE> – значение свойства.

Настройки кластера хранения данных STORAGE

Кластер **storage** выполняет функцию распределенного хранилища конфигурационных данных всей системы. Также в рамках этой подсистемы реализован модуль маршрутизации телефонных вызовов.

Настройка через CLI (CoCon)

Для изменения индивидуальных настроек кластера используется команда:

```
/cluster/storage/<NAME_CLUSTER>/<GROUP>/set <PROPERTY> <VALUE>
```

Для просмотра установленных значений параметров кластера используется команда:

```
/cluster/storage/<NAME_CLUSTER>/<GROUP>/info [<PROPERTY>]
```

где:

- <NAME_CLUSTER> – имя кластера, по умолчанию – ds1;
- <GROUP> – группа параметров;
- <PROPERTY> – имя свойства;
- <VALUE> – значение свойства.

Описание всех команд управления кластерами с ролью storage приведено в справочнике команд CLI.

На начальном этапе на уровне кластера хранения данных достаточно только установить [услуги](#) и [профили услуг](#), которые в дальнейшем будут использованы в доменах. Другие настройки без конкретной необходимости изменять не нужно.

Пример:

```
admin@mycelium1@ecss1:/$ cluster/storage/ds1/ss/install ds1@ecss1 ss_fax_receiver.xml
Successfully installed: /var/lib/ecss/ss/ss_fax_receiver.xml
```

Настройка параметров кластера через web-интерфейс

Для просмотра и изменения свойств кластера используется приложение [Кластеры \(Clusters\)](#), где нужно выбрать кластер с ролью storage.

Для управления услугами в web-конфигураторе существует отдельное приложение – "[Управление услугами \(SS install\)](#)".

Настройка параметров кластера MEDIATOR

Кластер **mediator** предназначена для сбора и экспорта предупреждений и статистической информации.

Настройка через CLI (CoCon)

Для изменения индивидуальных настроек кластера используется команда:

```
/cluster/mediator/<NAME_CLUSTER>/<GROUP>/set <PROPERTY> <VALUE>
```

Для просмотра установленных значений параметров кластера используется команда:

```
/cluster/mediator/<NAME_CLUSTER>/<GROUP>/info [<PROPERTY>]
```


где:

- <NAME_CLUSTER> – имя кластера, по умолчанию – md1;
- <GROUP> – группа параметров;
- <PROPERTY> – имя свойства;
- <VALUE> – значение свойства.

Описание всех команд управления кластерами с ролью **mediator** приведено в справочнике команд [CLI](#).

Рекомендуется сразу настроить следующие сервисы, если это предусмотрено проектом:

- службы уведомления о предупреждениях системы по электронной почте ([email](#));
- работу с [панелью аварий](#);
- правила [маскирования предупреждений](#);
- параметры [сбора статистики](#);
- параметры [SNMP](#).

 Остальные настройки по умолчанию уже готовы к работе и без необходимости их изменять не нужно.

Настройка параметров кластера через web-интерфейс

Для просмотра и изменения свойств используется приложение [Кластеры \(Clusters\)](#), где нужно выбрать кластер с ролью **mediator**. Для настройки маскирования предупреждений используется приложение "[Список предупреждений \(Alarm list\)](#)".

Для более тонкой настройки кластера смотрите раздел "[Тонкая настройка системы](#)".

Настройка параметров кластера CORE

Кластер **core** реализует логику управления обработкой телефонных вызовов (функции Call Control), предоставления услуг и обеспечение функционала биллинга.

Настройка через CLI (CoCon)

Для изменения индивидуальных настроек кластера используется команда:

```
/cluster/core/<NAME_CLUSTER>/<GROUP>/set <PROPERTY> <VALUE>
```

Для просмотра установленных значений параметров кластера используется команда:

```
/cluster/core/<NAME_CLUSTER>/<GROUP>/info [<PROPERTY>]
```

где:

- <NAME_CLUSTER> – имя кластера, по умолчанию – core1;
- <GROUP> – группа параметров;
- <PROPERTY> – имя свойства;
- <VALUE> – значение свойства.

Описание всех команд управления кластерами с ролью **core** приведено в справочнике [команд CLI](#). Для кластеров с ролью **core** все необходимые для работы значения параметров по умолчанию уже готовы к обработке нагрузки.

Если проектом предусмотрено, то на начальном этапе настраиваются:

- параметры службы [нотификации вызовов](#);
- подсистема TTS для сбора [CDR](#);
- при необходимости изменения значений по умолчанию настраивается сервис автоинформаторов [Call Forwarding by Cause \(CFC\)](#).

Настройка параметров кластера через web-интерфейс

Для просмотра и изменения свойств используется приложение [Кластеры \(Clusters\)](#), где нужно выбрать кластер с ролью **core**. Настройка CDR производится в приложении "[Менеджер CDR](#)". Настройка автоинформаторов (CFC) выполняется в приложении "[Автоинформатор \(CFC\)](#)" в стандартном [web-конфигураторе](#).

Настройка параметров кластеров PA_SIP

Общее описание принципов работы SIP-адаптера приведено в разделе "[Описание работы SIP-адаптера](#)".

Настройка через CLI (CoCon)

Для изменения индивидуальных настроек кластера используется команда:

```
/cluster/core/<NAME_CLUSTER>/<GROUP>/set <PROPERTY> <VALUE>
```

Для просмотра установленных значений параметров кластера используется команда:

```
/cluster/core/<NAME_CLUSTER>/<GROUP>/info [<PROPERTY>]
```


где:

- <NAME_CLUSTER> – имя кластера, по умолчанию – sip1;
- <GROUP> – группа параметров;
- <PROPERTY> – имя свойства;
- <VALUE> – значение свойства.

Описание всех команд управления кластерами с ролью **adapter** приведено в справочнике команд CLI в разделе "[/cluster/adapter/<PA_SIP>/ – команды управления кластером протокола адаптера SIP](#)".

На начальном этапе на адаптере необходимо настроить только [SIP-транспорт \(ipset\)](#) в соответствии с проектом:

- выбрать интерфейсы (node-ip) для работы адаптера. Если система с резервированием, то задаются IP, настроенные в /etc/keepalived.conf;
- добавить транспортные порты для приема сообщений протокола SIP;
- задать требуемое значение QoS DSCP.

 Настройка остальных параметров SIP перенесена на уровень виртуальных АТС.

Настройка параметров кластера через web-интерфейс

Параметры SIP-транспорта также можно настроить с помощью web-конфигуратора. Для настройки интерфейса необходимо открыть приложение "Кластеры ("Clusters"), выбрать адаптер и внести параметры транспорта. Подробнее в разделе "Создание IP-set/(sip-транспорта)".

9.3.2 Настройка программного медиасервера

Порядок настройки программного медиасервера (MSR):

- **настройка** конфигурационного файла медиасервера;
- **запуск** медиасервера;
- **добавление** медиаресурсов на ECSS-10:
 - командами CLI (см. раздел ["/system/media/resource/ - команды управления медиаресурсами"](#));
 - через приложение web-конфигуратора ["Сетевые окончания MSR \(MSR registrars\)"](#).
- опционально в соответствии с проектом:
 - **настройка распределенного медиасервера**;
 - замена **видеозаставки**.

9.3.3 Создание и настройка доменов (виртуальных АТС)

Данный этап включает в себя процесс создания виртуальных АТС, настройку правил маршрутизации вызовов, транков, абонентов, правил обслуживания абонентов.

Вся инфраструктура предоставления услуг телефонной связи на базе ECSS-10, а именно конфигурация подключаемых шлюзов, абонентские данные, план нумерации и правила маршрутизации, а также права доступа к функциям операционного управления и поддержки описываются в рамках определенного домена.

Таким образом, домен можно представить как логическую часть гибкого коммутатора, реализующую функционал отдельной АТС.

Таких сущностей на гибком коммутаторе может быть несколько. В системе ECSS-10 домен и виртуальная АТС – синонимы.

Фактически развертывание нескольких доменов и связей между ними дает возможность реализации сегмента или всей сети NGN в рамках одной инсталляции.

Системы доменов и гибкая система разграничения прав доступа позволяет оператору связи выполнять функции хостинга АТС для сторонних заказчиков.

Заказчик оператора связи может разместить свою корпоративную УПАТС или узел связи на мощностях системы ECSS-10, развернутой у оператора. При этом функции операционного управления за данной АТС могут быть переданы заказчику полностью или частично (используется схема разграничения ответственности за эксплуатацией данной АТС).

Декларация доменов

Необходимо создать домен по каждому из запроюктированных доменов. При создании нужно добавить администратора ECSS-10 в группу администраторов, а также в группу пользователей данного домена.

Создание домена выполняется одним из двух способов:

- через интерфейс командной строки (CLI);
- через приложение web-конфигуратора ["Домены"](#).

После создания доменов в файловой системе автоматически создадутся нужные каталоги уровня домена.

Настройка маршрутизации

- ✓ **Маршрутизация телефонных вызовов** — это процесс определения интерфейса назначения для конкретного вызова на основании информации об интерфейсе источника вызова, информации о телефонном номере вызывающего и вызываемого абонента, категории вызывающего абонента, времени суток и дне недели.
Контекст маршрутизации — совокупность правил маршрутизации уникальная в домене маршрутизации, в рамках которого идет определение интерфейса вызываемого абонента.

Описание процесса маршрутизации вызовов в системе ECSS-10 приведено в разделе "[Виртуальная АТС. Маршрутизация телефонных вызовов](#)".

Создание контекстов маршрутизации

В соответствии с проектом необходимо создать контексты маршрутизации вызовов, которые в дальнейшем будут применяться в настройках доменов.

Это можно сделать несколькими способами:

- **создать и импортировать** контексты вручную в [формате](#), указанном в документации. Контексты в формате xml создаются в каталоге `/var/lib/ecss/routing/ctx/src/<DOMAIN>`;
- **создать и настроить** контексты, используя приложение web-конфигуратора "[Менеджер маршрутизации \(Routing manager\)](#)" для каждого домена;
- опционально — настроить [RADIUS-маршрутизацию](#).

Далее, если это определено в проекте, необходимо подготовить контексты модификации и адаптации также для каждого домена.

Создание контекстов модификации и адаптации

- ✓ **Модификаторы номеров** — набор правил модификации номеров, которые применяются при звонке с определенного интерфейса или на определенный интерфейс. Модификатор назначается на интерфейс ECSS-10 (транк, абонент), группу интерфейсов.
Адаптация номеров — набор правил модификации номеров для COPM, TTS. Технически, это те же модификаторы номеров, но только применимые для адаптации номеров из внутреннего формата к формату COPM, TTS.

Описание и синтаксис контекстов адаптации и модификации приведен в разделе [Виртуальная АТС. Маршрутизация телефонных вызовов/Модификаторы и адаптация номеров по входу-выходу с интерфейса](#).

Применение контекстов маршрутизации для системных интерфейсов

Для интерфейсов **system:ivr** и **system:teleconference** необходимо назначить контексты маршрутизации. Настройки выполняются с помощью команд CLI. Описание и примеры команд приведены в разделе "[/domain/<DOMAIN>/system-iface/ - команды управления системными интерфейсами](#)".

Настройка CDR

- ✓ **CDR (Call Detail Record)** — детальная запись о параметрах вызова (номера телефонов, время начала разговора, продолжительность разговора и другое).

Для управления настройками системы CDR используется интерфейс командой строки и web-интерфейс.

Команды, предназначенные для управления настройками CDR, располагаются на виртуальной файловой системе CLI в директории `/domain/<DOMAIN>/cdr/`. Описание команд приведено в разделе [Справочник команд CLI](#).

Для управления CDR-файлами через web-конфигуратор используется приложение [Менеджер cdr \(Cdr manager\)](#).

⚠ Если система в кластере, предварительно должна быть настроена [Репликация БД MySQL](#).

Порядок настройки системы CDR:

1. Настройки TTS;
2. Создание и настройка CDR-группы;
3. Добавление алиаса (абонента) или транка в определенную CDR-группу.

Добавление и настройка прав пользователей

Пользователями являются лица, работающие с системой через CoSop или web-конфигуратор.

Каждый пользователь имеет следующий набор параметров:

- Имя;
- Пароль;
- Группа(ы) пользователей;
- Роль.

По умолчанию в системе создается пользователь **admin** с правами администратора системы. Пароль по умолчанию — **password**.

Для разграничения прав нужно создать требуемое количество пользователей и назначить каждому права и роли. Подробнее в разделе "[Управление пользователями](#)". Каждый пользователь может настроить себе параметры оболочки CoSop для удобства работы с помощью глобальной команды `/shell-options`.

Настройка правил ограничения обслуживания абонентов

Для возможности применять различные ограничения уровня абонента необходимо их настроить для каждого домена.

Различаются следующие виды ограничений для абонентов:

- долговременные ограничения, которые вводятся при подключении абонента и прописываются в договоре с абонентом, называются *типом доступа (access_type)*;
- группировка абонентов для возможности выхода абонентов одной группы на абонентов другой группы называется *группой доступа (access_group)*;
- временные ограничения, связанные с неоплатой абонентом счетов, называются *режимом обслуживания (regime)*;
- ограничения, которые задает себе сам абонент, называются *баррингами (barring)*.

Описание и настройка типов доступа, групп доступа, режимов обслуживания и баррингов приведены в разделе "[Тип доступа, режим обслуживания, категория доступа и барринги](#)".

Настройка производится:

- через интерфейс командной строки, см. разделы:
 - `/domain/<DOMAIN>/access-group` - команды управления группами доступа;
 - `/domain/<DOMAIN>/access-type` - команды управления типами доступа;
 - `/domain/<DOMAIN>/regime/` - команды управления режимами обслуживания абонентов.
- через приложение web-конфигуратора "[Менеджер доступа \(Access manager\)](#)".

Создание и настройка абонентов

В соответствии с проектом в доменах нужно создать требуемое количество абонентов. Сделать это можно с помощью команды CLI `/domain/<DOMAIN>/sip/user/declare` или приложения web-конфигуратора "[Карточка абонента \(Subscriber card\)](#)".

При создании абоненту нужно назначить номер, группу, контекст маршрутизации, CDR-группу, способ авторизации и авторизационные данные. Также сразу можно для каждого абонента назначить набор необходимых [услуг](#) и настроить необходимые ограничения.

Абоненты в домене – условие необязательное, в системе могут быть и чисто транзитные домены, на которых настраиваются соответствующие правила прохождения вызовов.

Создание и настройка транков

- ✓ **Транк** представляет собой совокупность ресурсов для обслуживания телефонных вызовов в заданном направлении (см. раздел "[Транки и бриджи](#)");
- **SIP-Транк** представляет собой направление, работающее по протоколу SIP/SIP-T/SIP-I;
- **Динамический транк** – транк с обязательной поддержкой регистрации. Для совершения вызова по динамическому транку взаимодействующий шлюз должен быть зарегистрирован по данному транку в системе ECSS-10.

Декларация и настройка транков производится:

- через интерфейс командной строки, см. раздел "[/domain/<DOMAIN>/trunk/sip/ - команды управления транками SIP](#)";
- через приложение web-конфигуратора "[Менеджер транков \(Trunk manager\)](#)".

Порядок создания и настройки транков приведен в разделе "[Управление SIP-транками](#)".

Далее на транках настраиваются необходимые [сервисы](#).

Создание и настройка bridge-интерфейсов

- ✓ **Бридж** – виртуальный транк, позволяющий соединять между собой две виртуальные АТС в рамках одной системы ECSS-10.

Если в системе имеется более одного домена, то связь между ними осуществляется с помощью **бриджей** (см. раздел "[Транки и бриджи](#)").

Бриджи создаются и настраиваются:

- через интерфейс командной строки, см. раздел "[/bridge/ – команды управления bridge-интерфейсами](#)";
- через приложение web-конфигуратора "[Менеджер бриджей \(Bridge manager\)](#)".

Настройка ограничений

Для каждого домена существует возможность задать разного рода ограничения в рамках [лицензии](#).

Таблица 1. Список ограничений уровня домена

Название свойства	Значение по умолчанию	Описание
alias_limit	infinity (ограничено лицензией)	Общее количество абонентов (в том числе и виртуальных) в данной виртуальной АТС.

Название свойства	Значение по умолчанию	Описание
call_limit	infinity (ограничено лицензией)	Общее количество одновременно активных вызовов для данной виртуальной АТС.
virtual_alias_limit	infinity (ограничено лицензией)	Общее количество виртуальных абонентов в данной виртуальной АТС.
digitmap		Список масок набора, по которому будет валидироваться алиасы при создании. Описание параметра приведено на странице /domain/ – команды управления виртуальными АТС
failover	true	Необходимость в резервировании вызовов на данной виртуальной АТС. Параметр используется только в системах с резервированием. Поскольку использование резерва увеличивает потребление ресурсов системы (процессор, оперативная память и другое), то исключение виртуальной АТС из схемы резервирования позволяет сэкономить часть ресурсов и направить сэкономленные ресурсы на обработку вызовов. В штатной работе системы это позволяет увеличить производительность в ущерб надежности.
callcenter\enabled	true	Доступ к контакт-центру для данной виртуальной АТС.
callcenter\active_agents	infinity (ограничено лицензией)	Максимальное количество подключаемых агентов Call-центра для домена.
callcenter\active_supervisors	infinity (ограничено лицензией)	Максимальное количество подключаемых супервизоров Call-центра для домена.
tc\active_conferences	infinity (ограничено лицензией)	Максимальное количество активных конференций для домена.
tc_count_active_channels	infinity (ограничено лицензией)	Максимальное количество подключаемых абонентов в конференцию сервиса Teleconference для домена.
ivr\enabled	true	Доступ к функциям IVR и dialer для данной виртуальной АТС.
ivr\incoming_script\enabled	true	Использовать для входящих транков в качестве контекста маршрутизации IVR-скрипт default_incoming_call.
teleconference\enabled	true	Доступ к сервису "Селекторная связь" для данной виртуальной АТС.
tsmn\concurrent_calls	0	Общее количество одновременно активных вызовов для системы TSMN на основном транке.
tsmn\concurrent_calls\redundancy	0	Общее количество одновременно активных вызовов для системы TSMN на резервном транке.
add_on_conferences_limit	infinity (ограничено лицензией)	Общее количество одновременно активных конференций для данной виртуальной АТС.
meet_me_limit	infinity (ограничено лицензией)	Общее количество активных пользователей "meet me" комнат для данной виртуальной АТС.

Название свойства	Значение по умолчанию	Описание
chat_room_limit	infinity (ограничено лицензией)	Общее количество активных конференц-комнат для данной виртуальной АТС.
dialer\channels	0 (ограничено лицензией)	Количество одновременных вызовов для кампаний обзвона.
recorder\voice\channels	0 (ограничено лицензией)	Количество одновременных каналов записи разговоров.
ss_package	0 (ограничено лицензией)	Количество лицензионных пакетов услуг.

Ограничения настраиваются в соответствии с проектом:

- через интерфейс командной строки, см. раздел ["/domain/<DOMAIN>/properties/restrictions/ - команды управления ограничениями виртуальной АТС"](#);
- через приложение web-конфигуратора ["Домены \(Domains\)"](#) (Свойства домена -> Системные параметры -> Ограничения).

Сценарии IVR

Для каждой виртуальной АТС можно настроить дополнительные сценарии IVR, если они сразу предусмотрены проектом. Сценарии создаются в приложении web-конфигуратора ["IVR-редактор \(IVR editor\)"](#). Также управлять сценариями можно с помощью команд CLI. Описание и примеры приведены в разделе ["/domain/<DOMAIN>/ivr/ - команды управления IVR-скриптами"](#).

При необходимости для каждого домена можно настроить ограничения для работы IVR:

- через интерфейс командной строки, см. раздел ["/system/ivr/script/restrictions/ - команды управления настройками ограничений IVR-скриптов"](#);
- через приложение web-конфигуратора ["Редактор IVR ограничений \(IVR restrictions manager\)"](#).

9.3.4 Настройка дополнительных сервисов

Если предусмотрено проектом, на начальном этапе настраиваются также дополнительные сервисы.

Настройка взаимодействия с серверами RADIUS

Описание и настройка взаимодействия с подсистемой AAA (Authentication, Authorization, Accounting) приведены в разделе ["Настройка динамических абонентов и системы RADIUS"](#).

Порядок настройки взаимодействия с серверами AAA:

- настройка параметров виртуальной АТС для взаимодействия с сервером аутентификации/авторизации (RADIUS);
- настройка параметров виртуальной АТС для взаимодействия с сервером аккаунтинга (RADIUS);
- настройка ограничений связи при сбое сервера.

Настройка функции СОРМ

В комплексе ECSS-10 заложены возможности для выполнения требований к системе технических средств по обеспечению функций оперативно-розыскных мероприятий на электронных АТС, утвержденные приказом Госкомсвязи России от 20.04.1999 № 70 и приказом Минкомсвязи России №268 от 19.11.2012.

Порядок настройки функции СОРМ:

- проверка наличия соответствующей лицензии;
- настройка протокола взаимодействия с посредником СОРМ;

- настройка маршрутизации в соответствии с требованиями сотрудников спецслужб.

Описание и настройка системы СОРМ приведены в разделе "[Система СОРМ](#)".

Настройка дополнительных приложений

В составе [экосистемы ECSS-10](#) возможно использование дополнительных сервисных приложений, расширяющих функциональные возможности:

- [Call-центр](#);
- инструментарий для проведения [селекторных совещаний](#);
- сервис "[Автообзвон](#)";
- сервис автоматического распознавания речи ([ASR](#));
- интеграции с [Desktop-ассистент, CRM, Skype for business](#));
- сервис "[Автосекретарь](#)";
- визуализация статистических данных в системе мониторинга "[Grafana](#)";
- приложение "[Портал абонента](#)";
- система "[Autoprovision \(AUP\)](#)" для автоматического конфигурирования и обновления ПО телефонных аппаратов.

Описание данных приложений приведено в соответствующих разделах документации.

9.4 Настройка ECSS-10 для производительных систем

Для производительных систем настройка ECSS-10 состоит из следующих этапов:

9.4.1 Выделение отдельных ядер процессора для MSR

Для того чтобы изолировать MSR-медиа сервер от остальной системы, необходимо выделить под него отдельные ядра процессора. Для выделения отдельных ядер процессора необходимо выполнить следующие действия:

1. Открыть файл:

```
/etc/default/grub
```

Привести параметр GRUB_CMDLINE_LINUX="" к следующему виду:

```
GRUB_CMDLINE_LINUX="isolcpus=8-11"
```

Данный пример изолирует ядра с 8 по 11. Также возможен вариант с перечислением 1, 2, 4-6 и т.п.

2. Обновить конфигурацию grub. Для этого выполните команду:

```
sudo update-grub
```

3. Перезапустить систему.

Если всё сделано правильно, то после перезагрузки на изолированных ядрах htop будет показывать нулевую нагрузку.

9.4.2 Установка `scaling_governor` в режим `perfomance`

По умолчанию в Ubuntu есть пять профилей работы процессора.

Описание профилей:

- **conservative** — медленно повышает частоту процессора в зависимости от нагрузки на систему и резко сбрасывает частоту к минимальной при простое;
- **ondemand** — быстро повышает частоту процессора при возрастании нагрузки и медленно сбрасывает частоту к минимуму при простое;
- **userspace** — позволяет указывать частоту вручную;
- **powersave** — соответствует минимальной допустимой частоте CPU;
- **performance** — соответствуют максимальной частоте CPU.

Проверить текущее значение для всех ядер можно следующей командой:

```
sasha@ecss1:~$ cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
performance
performance
performance
performance
performance
performance
performance
performance
```

Режимы, которые поддерживаются процессором можно посмотреть командой:

```
sasha@ecss1:~$ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
conservative ondemand userspace powersave performance schedutil
```

Ядро в Ubuntu 18.04 собирается с GOVERNOR=performance по умолчанию, но systemd-сервис *ondemand.service* может менять текущее значение на *ondemand* или *powersave*. Отключить сервис можно командой **sudo systemctl mask ondemand.service**.

Система выдерживает большую нагрузку в режиме performance. Для того чтобы включить данный режим по умолчанию, необходимо привести файл */etc/rc.local* к следующему виду:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
echo performance | tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor >/dev/
null
exit 0
```

⚠ Установку необходимо делать через создание дополнительного правила в */etc/udev/rules.d/*.

Либо принудительно выставить governor:

- поставить пакет **cpufrequtils**;
- в файле `/etc/default/cpufrequtils` установить переменную `GOVERNOR="performance"`.

9.4.3 Запуск MSR на изолированных ядрах процессора

Для того чтобы MSR запускался на отдельных ядрах процессора, необходимо привести файл `/etc/systemd/system/ecss-media-server.service.d/override.conf` к следующему виду:

```
[Service]
CPUAffinity=8-11
CPUSchedulingPolicy=rr
```

Перед этим нужно включить экземпляр MSR:

```
systemctl enable ecss-media-server@msr.service
systemctl edit ecss-media-server@msr.service
```

Для просмотра того, на каких ядрах запустился сервис, можно воспользоваться `htop`. В нем нужно добавить колонку `Processor`.

В данном примере MSR запущен на ядрах 8, 9, 10, 11. `CPUSchedulingPolicy` необходим, только если указан `isolcpus`.

Настройка MSR более подробно описана на странице [Настройка программного медиасервера](#).

9.4.4 Настройка использования определенных ядер процессора для erlang-based служб

Для того чтобы ядра процессора использовались правильно, необходимо скорректировать параметры запуска `erlang`-нод на производительных системах.

Для этого разрабатывается схема размещения нод на ядрах.

Схема разрабатывается по следующим правилам:

- использовать более двух ядер;
- необходимо, чтобы одна нода не использовала ядра на разных процессорах;
- для сильно нагруженных нод, таких как `core` и `sip`, нужно выделять индивидуальные ядра;
- ноды, которые не загружены, можно размещать на одном ядре;
- для `core` необходимо выделять большее количество ядер.

Распределения ядер на примере двухпроцессорного сервера HP BL660 с двумя процессорами Intel Xeon E5-4657L с 12 ядрами и поддержкой гипертрединга, которые могут образовать 8 виртуальных ядер:

```
mys  0-3
ds    4-7
core  8-23
sip   24-31
md    32-35
rest  36-39
sp    40-43
msr   44-47
```

Для осуществления данного распределения необходимо включить режим использования только необходимого количества ядер на erlang-ноде.

Для этого редактируем файл `vm.args` каждой ноды, расположенный по пути `/usr/lib/ecss/ECSS-SERVICE-NAME/releases/VERSION/`.

Например, для `ecss-core` необходимо отредактировать файл:

```
sudo mcedit /usr/lib/ecss/ecss-core/releases/3.14.2.29/vm.args
```

В этот файл добавить опции, которые задают использование требуемого количества логических ядер процессора и количество активных шедулеров.

Для 16-ти ядер это:

```
+sct L0-15c0-15  
+sbt db  
+S16:16
```

Для 8-ми ядер:

```
+sct L0-7c0-7  
+sbt db  
+S8:8
```

Для 4-х ядер:

```
+sct L0-3c0-3  
+sbt db  
+S4:4
```

Для 2-х ядер:

```
+sct L0-1c0-1  
+sbt db  
+S2:2
```

Следующая задача – установить сервис на выбранные ядра. Делается это аналогично тому, как описано для MSR.

Необходимо выполнить команду:

```
sudo systemctl edit ECSS-SERVICE-NAME
```

Далее добавить параметры:

```
[Service]
CPUAffinity=0-3
```

где в значении CPUAffinity указываются те ядра, на которых должны запускаться процессы сервиса.

Пример настройки ecss-core по указанной выше схеме:

```
> sudo systemctl edit ecss-core.service
[Service]
CPUAffinity=8-23
```

После настройки параметров CPUAffinity для всех сервисов необходимо перезагрузить конфигурацию услуг командой:

```
sudo systemctl daemon-reload
```

Перезапустить сервисы:

```
sudo systemctl restart ecss.slice
```

Убедиться в корректной привязке сервисов к ядрам можно утилитой htop, включив отображение колонки PROCESSOR.

10 Схема развертывания MySQL master-master replication с использованием keepalive

- Установка ecss-mysql и создание репликации
 - Проверка корректности настройки репликации
- Удаление репликации

10.1 Установка ecss-mysql и создание репликации

Для корректной отработки скрипта настройки репликации MySQL необходимо следующее:

- Версия ОС: Ubuntu 18.04;
- Версия MySQL: mysql-server-5.7;
- Два хоста для настройки репликации (по умолчанию имена хостов ecss1, ecss2);
- На каждом из хостов должен быть пользователь с одинаковым именем, имеющим root-права (нужен только на этапе выполнения скрипта). В рассмотренной ниже инструкции будет использоваться имя пользователя support.

Используемые переменные:

- <DB_ROOT_LOGIN> – имя root MySQL-сервера (root)
- <DB_ROOT_PASSWORD> – пароль root MySQL-сервера
- <HOST_1_IP> – IP-адрес хоста ecss1
- <HOST_2_IP> – IP-адрес хоста ecss2
- <HOST_1> – имя первого хоста (ecss1)
- <HOST_2> – имя второго хоста (ecss2)
- <REPLICA_USER> – пользователь, который будет использоваться для репликации данных (replica)
- <REPLICA_USER_PASSWORD> – пароль для REPLICA_USER
- <HOST_2_USER> – пользователь для доступа к shell ecss2
- <MEDIATOR_HOST> – хост, на котором развернут mediator (127.0.0.1)
- <SNMP_PORT> – SNMP-порт медиатора (162)

1. Установка пакетов с MySQL, конфигурацией баз данных, а также *keepalive* на обоих хостах:

```
sudo aptitude install ecss-mysql keepalived
```

При инсталляции ecss-mysql будут запрошены следующие данные:

- **IP-маска для прав MySQL-таблиц ("IP pattern for MySQL permission")** – эта маска указывает, из-под какого пула IP-адресов будет доступен вход в базу данных. Необходимо указать пул адресов, в который будет входить адрес серверов, где будет установлен ecss-node, а также адрес другого сервера ecss-mysql.

Пример:

Если оба MySQL-сервера находятся на хостах, где будет установлен ecss-node:

	ecss1	ecss2
ecss-mysql	192.168.1.1	192.168.1.2
ecss-node	192.168.1.1	192.168.1.2

Для этого примера в качестве маски для прав MySQL нужно указать 192.168.1.%.

Однако MySQL может быть вынесен на отдельные сервера, тогда маска сети должна быть сконфигурирована таким образом, чтобы хосты имели доступ друг к другу:

	ecss1	ecss2	mysql-server-1	mysql-server-2
ecss-mysql			192.168.2.1	192.168.2.2

	ecss1	ecss2	mysql-server-1	mysql-server-2
ecss-node	192.168.1.1	192.168.1.2		

Для этого примера в качестве маски для прав MySQL нужно указать 192.168.%.

- **Логин для администратора MySQL ("Login for MySQL root")** – этот логин будет установлен для сервера mysql. Логин нужно запомнить, так как он потребуется в ходе установки других нод. Также он используется в процессе создания backup-системы.
- **Пароль для администратора MySQL ("Password for MySQL root")** – этот пароль будет установлен для пользователя, указанного в ответе на предыдущий вопрос. Пароль нужно запомнить, так как он потребуется в ходе установки других нод. Также он используется в процессе создания backup-системы.

⚠ На обоих серверах, где устанавливаются пакеты ecss-mysql, должны использоваться одинаковые логины и пароли.

⚠ Базы данных MySQL, используемых системой ECSS-10, после установки будут храниться по пути `/var/lib/ecss-mysql`. При установке пакета ecss-mysql aptitude задаст вопрос о разрешении изменения конфигурационного файла "Configuration file '/etc/apparmor.d/local/usr.sbin.mysqld'", чтобы изменить путь до баз mysql по умолчанию. Для успешной установки ecss-mysql требуется согласиться (ввести "Y"). Чтобы избежать ввода ответа на вопрос при установке пакета, допускается использовать дополнительные ключи при вводе команды установки:
`sudo apt-get -o Dpkg::Options::="--force-confnew" install ecss-mysql`

2. Конфигурация bind-address для mysql.

Откройте конфигурационный файл mysql: `/etc/mysql/mysql.conf.d/mysqld.cnf` и добавьте туда следующую строку:

```
bind-address          = 0.0.0.0
```

Перезапустите сервис mysql:

```
sudo systemctl restart mysql.service
```

Данный пункт нужно выполнить на обоих хостах.

3. Установка имен хостов.

- Проверьте текущие имена хостов, где будет установлена ecss-mysql. Если сервера будут установлены на одном хосте с основной системой (ecss-node), то имя должно соответствовать ecss1/ecss2 соответственно. Если MySQL будут установлены на отдельные сервера, то имя хоста может быть любым. Воспользуйтесь следующей командой на обоих серверах, чтобы изменить имя хостов:

```
hostnamectl set-hostname <HOST_NAME>
```

- После установки имен на каждом сервере в конец файла `/etc/hosts` нужно добавить IP-адрес и имя соседнего сервера, например на ecss1 выполнить:

```
sudo sh -c 'echo "192.168.1.2 ecss2" >> /etc/hosts'
```

Если MySQL установлены на отдельных серверах, то добавьте в файл `/etc/hosts` имена и IP-адреса серверов с `ecss-node`, а также серверов с `ecss-mysql`.

4. Поддержка доступа между серверами с `ecss-mysql` через `ssh` по `rsa`-ключам без использования пароля.
- На хосте `ecss1` сгенерируйте `rsa`-ключ следующей командой (выполняйте команду без `sudo`, чтобы ключ сгенерировался для текущего пользователя):

```
ssh-keygen
```

- Передайте ключ на `ecss2`:

```
ssh-copy-id support@ecss2
```

- Аналогичным образом сгенерируйте ключ на хосте `ecss2` и скопируйте его на первый хост:

```
ssh-keygen  
ssh-copy-id support@ecss1
```

5. После успешной установки пакета `ecss-mysql` на обоих хостах требуется настроить репликацию между базами данных MySQL. Для этого на хосте **ecss1** запустите скрипт создания репликации:

```
/usr/lib/ecss/ecss-scripts/mysql-replication/install_replication.sh
```

Продублируйте ответы из пункта с установкой пакета до вопроса о настройке репликации:

- **Логин root пользователя ("Enter db root login")** — пользователь для доступа к MySQL.
- **Пароль root пользователя ("Enter db root password")** — пароль для доступа к MySQL.
- **Логин для пользователя replica ("Login for replica user")** — с этим логином в MySQL будет создан пользователь. Данный пользователь используется для входа в MySQL при репликации системы. Пароль нужно запомнить, он используется в процессе восстановления репликации.
- **Пароль для пользователя replica ("Password for replica user")** — этот пароль будет установлен для пользователя, указанного в ответе на предыдущий вопрос. Пароль нужно запомнить, так как он используется в процессе восстановления репликации.
- **IP-адрес локального хоста ("Local host IP address")** — IP-адрес хоста, на котором запускается репликация. В приведенном примере это `<HOST_1_IP>`.
- **IP-адрес удаленного хоста ("Remote host IP address")** — IP-адрес второго хоста, на котором развернут MySQL и с которым происходит репликация. В приведенном примере это `<HOST_2_IP>`.
- **Имя удаленного хоста ("Remote host name")** — имя соседнего хоста с установленным пакетом `ecss-mysql`. В приведенном примере это `ecss2`.
- **Логин для входа на удаленный хост ("Login for remote host")** — имя пользователя, которое будет использоваться для входа по `ssh` на соседний хост с `ecss-mysql`. В приведенном примере это `support`.
- **IP-адрес хоста с Mediator ("Mediator host IP address")** — IP-адрес хоста, на котором будет развернут пакет `ecss-node`. В приведенном примере это `<HOST_1_IP>`. Однако, если в разворачиваемой системе `ecss-node` и `ecss-mysql` будут находиться на различных серверах, требуется указать IP-адрес хоста с `ecss-node`.
- **SNMP-порт Mediator ("Mediator SNMP port")** — порт, на котором сервер с `mediator` слушает SNMP-трафик. Оставьте значение по умолчанию, равное **162**.

Следующая секция предлагает автоматическую настройку виртуального IP-адреса в *keepalive* для доступа к базе данных. В случае отработки резерва, виртуальный адрес будет автоматически переезжать на другой хост.

- **Настроить виртуальный IP-адрес? ("DO YOU WANT TO SET MASTER (VIRTUAL) IP address?")** – если ответ "yes", то данный скрипт самостоятельно создаст настройки для виртуального адреса MySQL, если "no", то настройки необходимо задать вручную согласно 6 пункту.
- **Определить, какой из хостов будем мастером ("WHICH HOST WILL BE MASTER?")** (опционально, появляется, если предыдущий ответ на вопрос "нет") – указывает, кто из хостов будет мастером.
- **Введите виртуальный IP-адрес ("ENTER VIRTUAL IP address")** – введите виртуальный IP-адрес. Этот адрес будет общим для двух баз MySQL и будет располагаться на одном из хостов. После настройки именно этот адрес должен быть указан во всех сервисах *ECSS*, которым необходим доступ к базе данных. Если один из хостов с *mysql* станет недоступным, то адрес посредством *keepalived* переедет на второй хост, и база данных останется доступной.
- **Нужна ли настройка keepalive? ("DO YOU WANT TO SET REST OF keepalive CONFIG")** – при положительном ответе запускает конфигурацию *keepalived*.
- **Введите маску для виртуального адреса ("ENTER VIRTUAL MASK (0-32)")** – маска виртуального адреса.
- **Введите router ID ("ENTER VIRTUAL ROUTER ID")** – идентификатор Router-ID используется в конфигурации *keepalived* для соотношения секций виртуальных адресов. Если до установки *ecss-mysql* на хосте не производилась настройка *keepalived*, то используйте любой целочисленный ID (104). Если до установки *ecss-mysql* уже проводились какие либо настройки сервиса *keepalived*, то перед вводом параметра убедитесь, что значение уже не задействовано в других секциях конфигураций *keepalived*.
- **Введите интерфейс ("ENTER INTERFACE")** – интерфейс, на котором необходимо развернуть виртуальный IP-адрес *ecss-mysql*.
- **Введите описание для интерфейса ("ENTER INTERFACE DESCRIPTION")** – описание интерфейса. Укажите корректное и явное название, например: "<INTERFACE>:mysql".

По завершению установки скрипт выводит значения переменных *Slave_IO_Running*, *Slave_SQL_Running* на *ecss1*, *ecss2*. Если переменные равны "yes" – значит репликация запустилась. Шаблон автоматического создания файла находится по пути `/usr/lib/ecss/ecss-scripts/mysql-replication/keepalived.conf` :

```

vrrp_script check_mysqlid {
    script "nc -z localhost 3306" # cheaper than pidof
    interval 2 # check every 2 seconds
    fall 1
    rise 2
}

!For MySQL
vrrp_instance MySQL {
    state BACKUP
    interface <INTERFACE>
    virtual_router_id <VIRTUAL_ROUTER_ID>
    priority 100
    advert_int 3
    smtp_alert
    nopreempt
    authentication {
        auth_type PASS
        auth_pass keepmysql
    }
    virtual_ipaddress {
        <VIRTUAL_IP>/<VIRTUAL_MASK> dev <INTERFACE> label <INTERFACE_DESCRIPTION>
    }
    track_script {
        check_mysqlid
    }
}

```

Где автоматически подставляются введенные значения:

- a. **<INTERFACE>** – физический или виртуальный интерфейс сервера, где установлен пакет `ecss-mysql`. С этого интерфейса будет осуществляться обмен трафиком и мониторинг доступности второго сервера с MySQL. Соответственно через интерфейсы, указанные в конфигурациях на обоих хостах, должен быть доступ до хостов с `ecss-mysql`, а также до `ecss-node`, если сервер с MySQL вынесен от основной системы.
 - b. **<VIRTUAL_IP>/<VIRTUAL_MASK>** – IP-адрес и маска виртуального адреса, через который будет осуществляться доступ к базе данных MySQL. После завершения конфигурации `keepalived` данный IP-адрес должен подняться на одном из хостов. В случае аварии на данном хосте IP-адрес должен динамически подняться на втором хосте. Таким образом, MySQL будет всегда доступен.
 - c. **<INTERFACE_DESCRIPTION>** – описание интерфейса. Укажите корректное и явное название, например: "`<INTERFACE>:mysql`".
6. Ручное конфигурирование `keepalived`:

⚠ Данный пункт необходимо выполнить в случае, если на вопрос "**Настроить виртуальный IP-адрес? ("DO YOU WANT TO SET MASTER (VIRTUAL) IP address?")**" был дан ответ "no". В этом случае в процессе создания реплики не будет создан виртуальный адрес. Следовательно, не будет обрабатываться резервирование для `ecss-mysql`.

Для отказоустойчивости в кластере ECSS-10 используется режим репликации MySQL мастер-мастер. Это позволяет корректно переносить данные в любом направлении. Однако одновременная запись на оба сервера MySQL с одновременной репликацией во встречном направлении повышает вероятность коллизий, что снижает отказоустойчивость. Поэтому рекомендуется настроить выделенный виртуальный адрес для кластера MySQL, чтобы запись данных в один момент времени производилась на одну ноду.

⚠ Если создавать файлы `/etc/keepalived/mysql.conf` вручную, то при запуске скрипта создания репликации нужно отказаться от автоматической настройки при вопросе "Нужна ли настройка keepalive? ("DO YOU WANT TO SET REST OF keepalive CONFIG").

Конфигурацию виртуальных адресов для MySQL рекомендуется поместить в отдельный файл `/etc/keepalived/mysql.conf`.

```
# Конфигурация mysql первой ноды:

vrrp_script check_mysql {
    script "/usr/bin/mysql --defaults-file=/etc/mysql/debian.cnf -e 'SELECT 1;'"
    user root
    interval 2
    fall 1
    timeout 2
}

vrrp_instance MySQL {
    state MASTER # Исходное состояние при старте
    interface <network_interface> # Имя сетевого интерфейса, на котором будет
    работать протокол VRRP
    virtual_router_id <ID> # Уникальный идентификатор роутера (0..255)
    priority 100 # Приоритет (0..255) чем выше – тем больше
    advert_int 1 # Интервал рассылки уведомлений (с)
    preempt_delay 60 # Интервал ожидания мастера при старте
    демона (с) при исходном состоянии BACKUP

    unicast_src_ip <src_real IP> # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP> # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
        # label – метка виртуального интерфейса (для удобства идентификации)
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }

    track_script {
        check_mysql
    }
}
```

```

# Конфигурация mysql второй ноды:

vrrp_script check_mysql {
    script "/usr/bin/mysql --defaults-file=/etc/mysql/debian.cnf -e 'SELECT 1;'"
    user root
    interval 2
    fall 1
    timeout 2
}

vrrp_instance MySQL {
    state BACKUP
    interface <network_interface>      # Имя сетевого интерфейса, на котором будет
    работать протокол VRRP
    virtual_router_id <ID>              # Уникальный идентификатор роутера (0..255)
    priority 50                         # Приоритет (0..255) чем выше – тем больше
    advert_int 1                        # Интервал рассылки уведомлений (с)
    preempt_delay 60                    # Интервал ожидания мастера при старте
демона (с) при исходном состоянии BACKUP

    unicast_src_ip <src_real IP>       # Собственный реальный IP-адрес
    unicast_peer {
        <real_remote IP>               # Реальный IP-адрес соседа
    }

    virtual_ipaddress {
        # Виртуальный IP-адрес и маска
        # dev – сетевой интерфейс, на котором будет поднят виртуальный адрес
        # label – метка виртуального интерфейса (для удобства идентификации)
        <virtual_sip_IP>/<netmask> dev <> label <label>
    }

    track_script {
        check_mysql
    }
}

```

7. Для того чтобы доступ до реплицированного MySQL был прозрачен для клиентов, создайте виртуальный IP. Он будет переезжать между хостами ecss1, ecss2 средствами VRRP+keepalive, если на одном из них MySQL остановился. Для этого в конфигурационный файл keepalived.conf на каждой ноде добавить в конце следующую строку (см. раздел "[Общая настройка keepalived](#)"):

```
include /etc/keepalived/mysql.conf
```

8. Перезапустите *keepalived* на каждом из хостов ecss1, ecss2:

```
sudo systemctl restart keepalived.service
```

После этого на одном из хостов должен подняться интерфейс, по которому будет доступен MySQL. В случае, если на данном хосте остановить MySQL, то сетевой интерфейс переместится на соседний хост.

9. Настраиваем dnsmasq:

⚠ Данный пункт выполняется на хосте с установленным пакетом `ecss-node`.

Для того чтобы подключить систему ECSS-10 к реплицированному MySQL, необходимо настроить `dnsmasq`. Для этого нужно переконфигурировать пакет `ecss-dns-env`:

```
sudo dpkg-reconfigure ecss-dns-env
```

В открывшемся окне конфигурации выбрать `mysql`, а затем ввести адрес **<VIRTUAL_IP>**.

10.1.1 Проверка корректности настройки репликации

Для проверки того, что репликация данных работает корректно необходимо выполнить следующие действия:

1. Проверить состояние реплики. Для этого выполните команду:

```
sudo mysql -uroot -p -e 'show slave status \G;' | grep -E "Slave_IO_Running:|Slave_SQL_Running:"
```

Данная команда должна показывать на обоих хостах следующую информацию:

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

2. Проверить состояние событий. Для этого выполните команду:

```
sudo mysql -uroot -p -e 'use ecss_statistics; show events\G;' | grep Status
```

Оба из хостов (на котором присутствует виртуальный адрес *keepalive*) должен показать следующий статус:

```
Status: ENABLED
```

3. Проверить таблицы состояний:
Для мастера данное число будет равно 1, для slave 0.

```
mysql> select * from ecss_system.events_status;
+-----+-----+
| lock  | enabled |
+-----+-----+
| events |      0 |
+-----+-----+
1 row in set (0.00 sec)
```

В случае возникновения ошибок репликации необходимо выполнить действия, описанные в разделе [Восстановление репликации БД MySQL](#).

Для получения предупреждений при нарушении репликации необходимо раскомментировать следующие строки в конфигурационном файле медиатора `/etc/ecss/ecss_mediator/md1@ecss1.config` и перезапустить `ecss-mediator.0`:

```
% ,{manager, [  
%     {config, [  
%         {dir, "/etc/ecss/snmp/manager/"},  
%         {verbosity, silence},  
%         {db_dir, "/var/lib/ecss/snmp/"}  
%     ]},  
%     {def_user_mod, snmp_alarm_user}  
% ]}
```

10.2 Удаление репликации

Для того чтобы удалить ранее созданную репликацию, необходимо использовать специальный скрипт:

```
/usr/lib/ecss/ecss-scripts/mysql-replication/remove_replication.sh
```


11 Примеры пошаговой первоначальной настройки ECSS-10

- Первоначальная установка системы без резервирования с одним сервером
 - Исходные данные
 - Включение в сеть
 - Настройка сети
 - Подготовка сетевых интерфейсов системы
 - Обновление программного обеспечения операционной системы
 - Установка и настройка программного обеспечения
 - Установка пакета ecss-mysql
 - Настройка пакета ecss-dns-env
 - Настройка пакета ecss-mysql
 - Установка пакета ecss-node
 - Установка пакетов ecss-media-server, ecss-media-resources, ecss-restf, ecss-web-conf
 - ecss-media-server, ecss-media-resources
 - ecss-restfs
 - ecss-web-conf
 - Настройка snpmd
 - Настройка безопасности. SSH
 - Настройка начальной конфигурации
- Первоначальная установка системы с резервированием в кластере из двух серверов
 - Исходные данные
 - Включение в сеть
 - Настройка сети
 - Пример netplan для режима active-backup
 - Пример netplan для режима 802.3ad
 - Настройка /etc/hosts
 - Обновление программного обеспечения операционной системы
 - Установка и настройка программного обеспечения
 - Установка пакета ecss-mysql
 - Настройка пакета ecss-dns-env
 - Настройка пакета ecss-mysql
 - Редактирование keepalived.conf
 - Создание VRRP для MySQL.
 - Установка ecss-node
 - Установка остальных пакетов ecss и их настройка
 - Настройка VRRP для SIP-адаптера
 - Дальнейшая настройка
 - mycelium.config
 - epmd
 - glusterfs
 - snpmd
 - Настройка безопасности. SSH
 - Настройка кластера пакетов ecss-node
 - Настройка группы IP-адресов (IP-set)
- Пример первичной конфигурации системы с помощью web-конфигуратора
 - Исходные данные
 - Подготовка к работе
 - Создание учетной записи оператора
 - Создание домена
 - Создание IP-set (sip-транспорта) и присвоение его к домену
 - Создание абонентов
 - Создание и применение контекстов маршрутизации для домена
 - Создание транка

- [Создание IVR-скрипта](#)
- [Завершение настройки маршрутизации](#)
- [Настройка услуг](#)

В данном разделе приведены примеры инсталляции и настройки ECSS-10 для системы с одиночным сервером и для кластера из двух серверов.

11.1 Первоначальная установка системы без резервирования с одним сервером

11.1.1 Исходные данные

Исходные данные

В данном разделе использованы специфичные для данного продукта термины, ознакомиться с ними можно [здесь](#).

Требуется интеграция программного коммутатора 5 класса ECSS-10 (SSW) на 1 физическом сервере с поддержкой SIP со следующими параметрами на нагрузку:

- Максимальное количество абонентов – 15000 (MUL – Max user limit);
- Максимальное количество одновременных соединений – 2000 (MCL – Max call limit);
- Резервирование системы не требуется;
- Количество сетевых интерфейсов ethernet – 4.

По техническому заданию требуется определить аппаратную платформу.

Таблица 1. Рекомендуемые аппаратные решения

Требования к серверам SSW	Серия аппаратного продукта			
	Light+	Midi	Heavy	Super Heavy
Характеристики системы				
Максимальное количество абонентов	5000	10000	20000	40000
Максимальная нагрузка одновременных соединений class 5	800	1500	3000	6000
Максимальная нагрузка одновременных соединений class 4	2400	4500	9000	20000
Характеристики сервера				

Марка	HP (Lenovo)	HP (Lenovo)	HP (Lenovo)	HP (Lenovo)
Серия	DL20 Gen10 (SR250)/DL 360 Gen10 (SR530)	DL360 Gen10 (SR530/ SR630)	DL360 Gen10 (SR630)	DL 360 Gen10 (SR630)
Процессор	Intel Xeon E-2276G/ Intel Xeon 4214	Intel Xeon 5220	Intel Xeon 6240	Intel Xeon 8268
Количество процессоров	1	1	2	2
Объем ОЗУ	12 ГБ	16 ГБ	24 ГБ	64 ГБ
HDD	От 3x500 SATA (от 7200 об/мин)	От 3x300 ГБ SAS (от 10000 об/мин)	От 3x600 ГБ SAS (от 10000 об/мин) 2x150 ГБ SSD	От 6x800 ГБ SSD, 2x300 ГБ M.2 SSD
RAID	No raid board	HW Raid, от 1 ГБ cache+battery	HW Raid, от 1 ГБ cache+battery	HW Raid, от 2 ГБ Flashcache, поддержка RAID-5

Дополнительные комплектующие сервера (в базовых комплект не входят)

Лицензия на удалённое управление	по желанию	+	+	+
Резервный источник питания	по желанию	+	+	+
Организация хранилища записей разговоров	Дополнительные HDD, объединенные в RAID-5	Лицензия на HW Raid с поддержкой RAID-5, дополнительные HDD для хранения записей	Лицензия на HW Raid с поддержкой RAID-5, дополнительные HDD для хранения записей	Лицензия на HW Raid с поддержкой RAID-5, дополнительные HDD для хранения записей

Таблица 2. Пример составления аппаратных требований

Устройство	Требуемый ресурс		Серия аппаратного продукта
	MCL	MUL	
Сервер 1	2500	15000	Heavy

После определения по требованиям проекта составляем предварительную сетевую карту.

Таблица 3. Пример разнесения составляющих по адресному пространству для одной ноды

Имя сервера (хоста)	Роль	Интерфейс	Адрес	Порт
Статические адреса программного коммутатора				
ecss1	Адрес сервера	network.10 (mgm)	10.0.10.11/24	-
ecss1	Адрес протокольного адаптера	network.20 (voip)	10.0.20.11/24	-
ecss1	Адрес шлюза	network.10 (mgm)	10.0.10.1	-
ecss1	Адреса DNS-серверов	network.10 (mgm)	10.0.10.1, 8.8.8.8	-
Внутренние адреса программного коммутатора				
ecss1	Адрес ядра (<i>ecss-core</i>)	lo	127.0.0.1/24	5000
ecss1	Адрес медиасервера (<i>ecss-media-server (MSR)</i>)	lo	127.0.0.1/24	5040
ecss1	Адрес базы данных MySQL (<i>ecss-mysql</i>)	lo	127.0.0.1/24	3306

11.1.2 Включение в сеть

Топологию включения сервера в сеть для обеспечения резервирования рекомендуется делать с использованием 2-х коммутаторов.

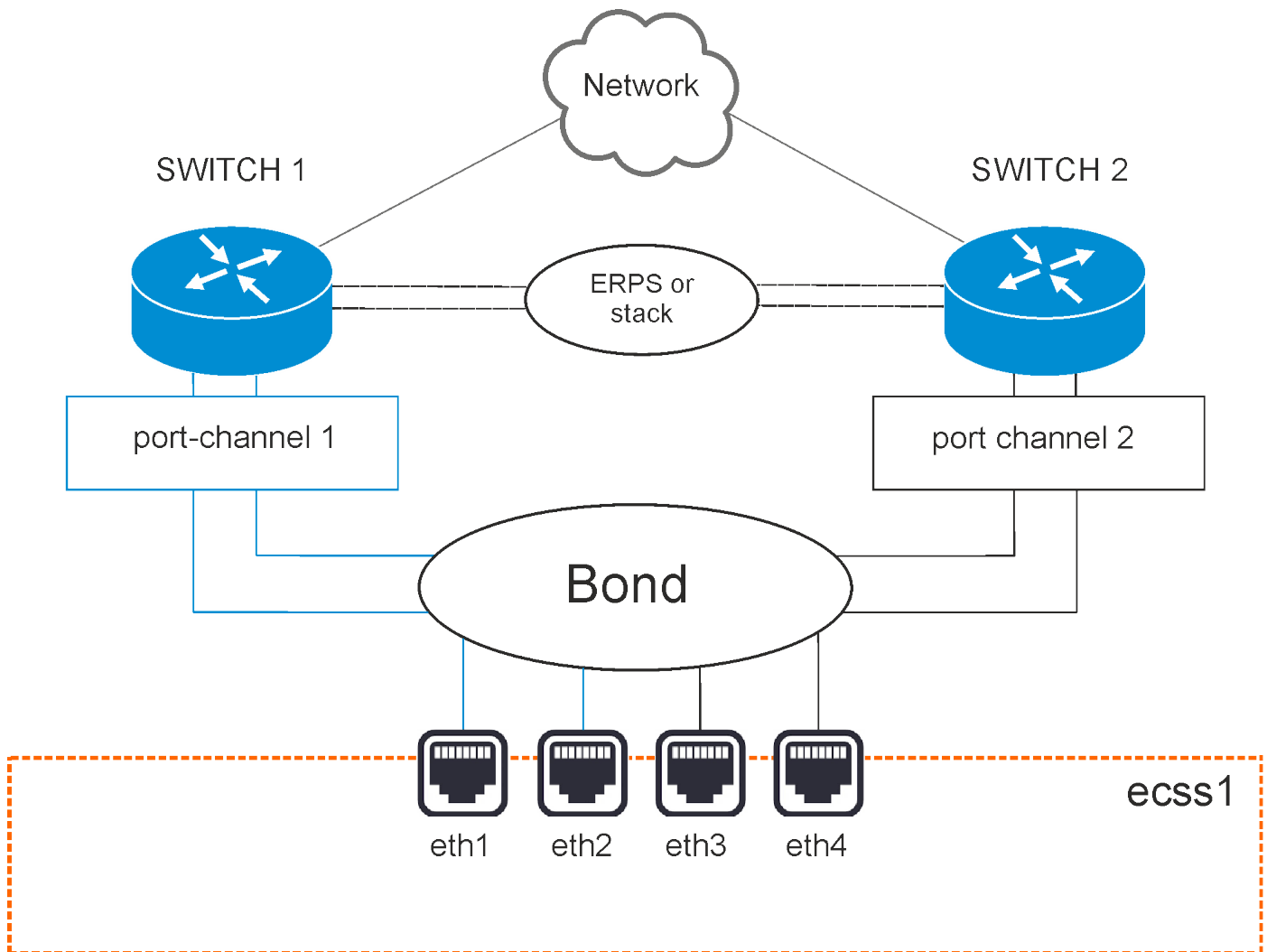


Рисунок 1 – Схема организации связи

Вариант 1. Схема active-backup

Коммутаторы связаны между собой в erps-кольцо.

Все 4 физических сетевых интерфейса объединяются в 1 агрегированный линк (bond). Агрегация портов сервера настраивается в режиме active-backup, т.е. в работе всегда находится только 1 сетевой интерфейс. Сетевые интерфейсы сервера попарно включены в коммутаторы, на которых агрегация портов (port-channel) также настраивается в режиме active-backup.

Например, eth1 и eth2 включаются в первый коммутатор, а eth3 и eth4 – во второй.

Вариант 2. Схема LACP

Коммутаторы связаны между собой в стек. Стек должен логически работать как один коммутатор, способный обеспечивать агрегацию портов в режиме LACP между разными физическими коммутаторами. Пример – коммутаторы MES3124 со специализированной прошивкой.

Все 4 физических сетевых интерфейса также объединяются в 1 агрегированный линк (bond). Агрегация портов сервера настраивается в режиме **802.3ad**. Создаются агрегированные группы сетевых карт с одинаковой скоростью и дуплексом. При таком объединении передача задействует все каналы в активной агрегации согласно стандарту IEEE 802.3ad. Выбор, через какой интерфейс отправлять пакет, определяется политикой. По умолчанию это XOR-политика, можно использовать «xmit_hash» политику. Подробнее в разделе [Приложение E. Netplan](#).

Требования:

- Поддержка Ehtool в драйвере для получения информации о скорости и дуплексе на каждом сетевом интерфейсе;
- Поддержка на коммутаторе стандарта IEEE 802.3ad;

Сетевые интерфейсы сервера также попарно включены в коммутаторы, на которых агрегация портов (port-channel) настраивается в режиме LACP. Например, eth1 и eth2 включаются в первый коммутатор (port-channel 1), а eth3 и eth4 – во второй (port-channel 2).

11.1.3 Настройка сети

Согласно параметрам, заданным в техническом задании, выполним сетевые настройки. В данном примере считается, что необходимая операционная система уже установлена. О том, как это сделать, смотрите в разделе [Инсталляция системы](#).

Рекомендуется разделять трафик, используемый в разных целях. Например, трафик управления и трафик VoIP. Для этого создаются 2 или более vlan. При небольшой нагрузке для простоты можно обойтись и одним vlan, но это будет в будущем доставлять неудобства при снятии дампов трафика и его анализе. В соответствии с техническим заданием на vlan настраиваются IP-адреса самого хоста, шлюзы, DNS и маршрутизация в другие сети.

В данном примере согласно ТЗ будем использовать адреса:

- 10.0.10.10/24 – для управления, vlan 10;
- 10.0.20.10/24 – для VoIP.

Внутри серверной платформы существует адресная структура и используются внутренние адреса для взаимодействия между подсистемами (нодами) в кластере. Например, внутренний адрес для кластера на одном сервере 127.0.0.1, при этом ядро (ecss-core) взаимодействует с сервером обработки мультимедиа-данных (ecss-media-server). Их взаимодействие происходит с использованием одного и того же адреса, но у каждой программной части свой транспортный порт: ecss-core – 5000, ecss-msr – 5040.

Для всех нод кластера определяется единый адрес для обращения к базе данных MySQL, например адрес ecss-mysql 127.0.0.1. Таким образом, выполняется условие единообразия, при котором все ноды кластера имеют полностью одинаковые данные о текущем состоянии динамических составляющих программного коммутатора (например, история вызовов).

Подготовка сетевых интерфейсов системы

Согласно ТЗ, система имеет 4 сетевых интерфейса. Информацию об их состоянии можно посмотреть с помощью команды *ifconfig* или *ip a*:

```
eth0: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST> mtu 1500
ether 36:10:28:73:63:01 txqueuelen 1000 (Ethernet)

eth1: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST> mtu 1500
ether 36:10:28:73:63:01 txqueuelen 1000 (Ethernet)

eth2: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST> mtu 1500
ether be:77:ea:52:4d:39 txqueuelen 1000 (Ethernet)

eth3: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST> mtu 1500
ether be:77:ea:52:4d:39 txqueuelen 1000 (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
```

Сначала настраиваются сетевые интерфейсы. В Ubuntu18 для настройки используется утилита [netplan](#).

Данная утилита позволяет настроить сетевую конфигурацию и затем загрузить ее в систему с помощью сетевого менеджера `networkd` или `NetworkManager`.

```
sudo nano /etc/netplan/ecss_netplan.yaml
```

⚠ Все другие файлы из этого каталога нужно переместить в другое место или удалить.

В конфигурациях для каждого хоста в первую очередь мы объявляем секцию `ethernets`, в которой описываются существующие в системе ethernet-интерфейсы, используемые в дальнейшем. Для каждого интерфейса необходимо отключить использование динамического распределения адресов (DHCP).

Следующим разделом описываются агрегированные каналы — `bonds`. В зависимости от выбранного варианта включения в сеть настраивается режим резерва 1:1 (`active - backup`) или LACP (802.3ad).

Далее настраиваются `vlan`, на которых опционально определяются шлюзы для связи с внешним миром и адреса DNS-серверов, а также непосредственно IP-адреса для каждого интерфейса.

❗ ВАЖНО

Обратите внимание, что во время редактирования `netplan` необходимо соблюдать правила разметки YAML:

- Обязательное наличие минимум двух пробелов перед каждой строкой (кроме `network`);
- Каждая подсекция дополнительно сдвигается на 2 пробела:

```
→ Секция                |network
→ Подсекция             |  _'bonds:
→ Подсекция описания    |    _'bonded_one:
→ и т.д.                 |    _'...
```

- Перед знаком ":" нет пробела, после — один пробел;
- Перед знаком "-" количество пробелов такое, как будто начинается новый подраздел, после — один пробел.

Пример настройки файла `ecss-netplan.yaml` для варианта включения `active-backup`

```

# Netplan для хоста ecss1 программного коммутатора
# Обратите внимание на обязательное наличие минимум двух пробелов в каждой строке и
# секции (кроме строки секции network)

network:
  version: 2 # Версия netplan
  renderer: networkd # Исполнитель конфигураций netplan
  ethernet: # Раздел описания интерфейсов ethernet
    eth0: # Название интерфейса
      dhcp4: no # Отключаем на интерфейсах динамическое распределение IP-адреса
    eth1:
      dhcp4: no
    eth2:
      dhcp4: no
    eth3:
      dhcp4: no

  bonds: # Раздел описания связанных (bonding) интерфейсов. В имени не может быть
# более 15 символов!
    bond1: # Название bonding-интерфейса
      interfaces: # Секция определения связанного интерфейса
        - eth0
        - eth1
        - eth2
        - eth3
      parameters: # Раздел определения параметров связанного интерфейса
        mode: active-backup # Режим резервирования 1:1
        mii-monitor-interval: 100 # интервал мониторинга интерфейса (мс)
        primary: eth0 # Секция определения основного интерфейса
        optional: false # Определение, является ли интерфейс обязательным при старте

  vlans:
    network.10: # Интерфейс управления
      id: 10
      link: bond1
      addresses: [10.0.10.10/24]
      gateway4: 10.0.10.1 # Адрес шлюза
      nameservers:
        addresses: [10.0.10.1, 8.8.8.8] # Адреса серверов DNS
    network.20: # Интерфейс для VoIP
      id: 20
      link: bond1
      addresses: [10.0.20.10/24]

```

Применим установленные параметры командой:

```
sudo netplan apply
```

11.1.4 Обновление программного обеспечения операционной системы

Для установки системы ECSS-10 необходимо добавить репозиторий ELTEX:


```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/bionic/3.14 stable main extras external' > /etc/apt/sources.list.d/eltex-ecss10-stable.list"
```

- ✔ Обратите внимание, что требуется указать верную версию операционной системы при добавлении репозитория ELTEX. Если установка происходит на Ubuntu 18.04, то необходимо указать **bionic**, как приведено в примере. Однако, если ECSS-10 устанавливается на Astra Linux, необходимо указать соответствующие репозитории **smolensk**:

```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/smolensk/3.14 stable main extras external' > /etc/apt/sources.list.d/eltex-ecss10-stable.list"
sudo sh -c "echo 'http://archive.eltex.org astra smolensk smolensk-extras' > /etc/apt/sources.list.d/eltex-ecss10-stable.list"
```

Далее необходимо выполнить импорт ключа командой:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 33CB2B750F8BB6A5
```

Перед началом установки необходимо обновить ОС:

```
sudo apt update
sudo apt upgrade
```

11.1.5 Установка и настройка программного обеспечения

❗ ВАЖНО

Установку пакетов требуется делать НЕ из-под пользователя ssw.

Устанавливаем все [предложенные пакеты](#):

```
sudo apt install aptitude atop ethtool htop iotop mc minicom mtr-tiny nmap pptpd pv screen ssh tftpd vim sngrep tshark cpanminus gnuplot libgraph-easy-perl debconf-utils
```

Установка пакета ecss-mysql

Установка начинается с развертывания MySQL-сервера и интеграции базы данных ecss-mysql.

Для [установки](#) выполняем команду:

```
sudo apt install ecss-mysql
```

Настройка пакета ecss-dns-env

Перед установкой пакета ecss-mysql будет предложено настроить переменные окружения для сервисов в dnsmasq. Настройщик предложит выбрать разделы для настройки, не выбираем ничего.

Настройка пакета ecss-mysql

При инсталляции пакета будут запрошены следующие данные:

Вопросы	Ответы
Адресная маска для MySQL (IP pattern for MySQL permission)	127.0.0.%
Логин пользователя (Login for MySQL root)	root
Пароль пользователя MySQL (Password for MySQL root)	PASSWORD

Изменение пути по умолчанию — согласиться с изменением конфигурационного файла для внесения пути до баз данных *ecss-mysql*, введя "Y".

Базы данных *mysql*, используемых системой ECSS-10, после установки будут храниться по пути **/var/lib/ecss-mysql**. Проверяем наличие файлов в папке:

```
ls -l /var/lib/ecss-mysql/
total 36
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_address_book
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:37 ecss_audit
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_calls_db
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_dialer_db
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_meeting_db
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_statistics
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_subscribers
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 history_db
drwxr-xr-x 2 mysql mysql 4096 Sep 26 14:32 web_conf
```

Проверяем, что сервер запущен:

```
sudo systemctl status mysql
● mysql.service - MySQL Community Server
Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
Drop-In: /etc/systemd/system/mysql.service.d
└─override.conf
Active: active (running) since Thu 2019-09-26 13:36:47 +07; 23h ago
Main PID: 11642 (mysqld)
CGroup: /system.slice/mysql.service
└─11642 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
```

Установка пакета **ecss-node**

Устанавливаем пакет **ecss-node**:

```
sudo apt install ecss-node
```

Во время установки пакета создается пользователь **ssw**, от имени которого запускаются все сервисы *ecss**. Создаются необходимые каталоги, выполняется настройка DNS, идет настройка SSL-сертификатов. В ходе инсталляции будут задаваться **вопросы**, необходимые для формирования конфигурационных файлов.

Вопросы	Ответы
Выключить ежедневные обновления (Do you want turn off apt-daily update ?)	Yes
Установить настройки по умолчанию для БД (Set DB config to default ?)	Yes

Вопросы	Ответы
Взводить аварию ECSS-10 при заполнении раздела MySQL (Set alarm true when MYSQL DB overloads)	Yes
Использовать настройки NTP для кластера (NTP: Do you want use settings for cluster ?)	No
Внешний NTP сервер (External NTP servers through a space)	ntp.ubuntu.com (по умолчанию) Ввести один или несколько серверов через пробел, используемых на объекте
Использовать локальный сервер? (NTP: Do you want use local server ?)	No
Адреса и маски адресов которые должны иметь доступ к NTP (NTP: Addresses and Masks of Network, which must have access to the ntp through a space)	Ввести список подсетей, из которых будет доступ к данному NTP-серверу, например, 10.0.10.0 255.255.255.0
Установить утилиты для работы с cdr	No

Для генерации сертификатов выбираем ручной (manual) способ. На все вопросы можно дать ответы, предложенные по умолчанию, нажимая кнопку "Enter" на каждый вопрос.

Установка пакетов **ecss-media-server**, **ecss-media-resources**, **ecss-restf**, **ecss-web-conf**

Далее [устанавливаются](#) пакеты *ecss-media-server*, *ecss-media-resources*, *ecss-restf*, *ecss-web-conf* и другие в любом порядке:

ecss-media-server, *ecss-media-resources*

```
sudo apt install ecss-media-server ecss-media-resources
```

Для медиасервера (*ecss-media-server* / *MSR*) возможно начальное конфигурирование с записью параметров в [файл конфигурации](#), требуется провести конфигурацию, не выбирая никаких пунктов:

Вопросы <i>ecss-media-server</i>	Ответы для <i>ecss1</i>
Установить настройки по умолчанию:	yes
Введите имя (Enter)	msr.ecss1
Введите адрес (Enter)	127.0.0.1
Введите порт (Enter)	5000

После формирования конфигураций по умолчанию переходим в директорию, в которой находятся конфигурации и производим их проверку:

```
cd /etc/ecss/ecss-media-server/  
cat config.xml  
cat conf.d/default.xml
```

Внутри лежит конфигурация для msr: config.xml, в директории conf.d лежит конфигурация default.xml. По своей сути default.xml – это дополнение config.xml, которое определяет секцию аккаунтов. Это сделано для того, чтобы после обновлений пакета данная конфигурация оставалась неизменной. Вид config.xml представлен здесь: [файл конфигурации](#). Рассмотрим секцию аккаунтов (файл default.xml):

Настройка msr для ecss1(/etc/ecss/ecss-media-server/conf.d/default.xml)

```
<?xml version="1.0"?>
<config>
  <accounts>
    <dynamic msr_name="msr.ecss1" realm="sip:127.0.0.1:5000"
dtmf_mode="rfc+inband+info" auth_name="user" auth_password="password"/>
  </accounts>
</config>
```

В ней указаны текущие настройки, согласно которым происходит регистрация msr на core.

Основными параметрами здесь являются: msr_name и realm:

- msr_name – параметр, определяющий название msr. (рекомендуется задавать название msr. и к какому хосту он принадлежит, например msr.ecss1);
- realm – определяет адрес для регистрации на ядре. Точка входа по умолчанию: порт 5000, адрес: 127.0.0.1.

ecss-restfs

```
sudo apt install ecss-restfs
```

При его установке будет предложено настроить конфигурацию:

Вопросы	Ответы
Использование функции TTS (Use TTS service)	No
Настройка телефонной книги (Configure phone book)	No
Настроить сервис определения речи (Configure speech recognition service)	No
Ничего не выбирать	Ok

ecss-web-conf

```
sudo apt install ecss-web-conf
```

При его установке будет предложено настроить конфигурацию:

Вопросы	Ответы
Указать доменное имя или адрес для базы данных MySQL web-интерфейса (Input IP address or hostname of MySQL db for web-conf DB)	127.0.0.1
Указать порт для базы данных MySQL web-интерфейса (Input port of MySQL db for web-conf DB)	3306
Указать доменное имя или адрес для подключения к web-интерфейсу (Input IP address or hostname for ECSS-10 with http_terminal)	127.0.0.1
Указать порт для web-терминала программного коммутатора (Input port SSW http_terminal)	9999
Указать логин для web-терминала (Input login for SSW http_terminal)	admin
Указать пароль для web-терминала (Input password for SSW http_terminal)	password

Настройка snmpd

Настройка SNMP описана [здесь](#).

Настройка безопасности. SSH

Выполняем [настройку](#) сервера SSH:

```
sudo nano /etc/ssh/
sshd_config
```

Укажем в конфигурационном файле порт и адрес, по которому можно обратиться к серверу:

Настройка ssh для ecss1(/etc/ssh/sshd_config)

```
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Port 2000
#AddressFamily any
ListenAddress 10.0.10.10
#ListenAddress ::

<...>
```

Перезапустим ssh:

```
sudo systemctl restart ssh.service
```

11.1.6 Настройка начальной конфигурации

Запускаем необходимые сервисы.

❗ ВАЖНО

Перед началом работы [проверьте наличие Token в системе](#).

Запускаем пакеты *ecss-mycelium* и *ecss-ds* на первом хосте:

```
sudo systemctl start ecss-mycelium
sudo systemctl start ecss-ds
```

Переходим в CLI:

```
ssh admin@localhost -p 8023
password: password
```

Проверяем состояние системы:

```
admin@mycelium1@ecss1$ system-status
Checking...
```

	Node	Release	Erlang nodes	Mnesia nodes	Uptime
	ds1@ecss1	ecss-ds-3.14.10.91	ds1@ecss1	ds1@ecss1	8m 9s
	mycelium1@ecss1	ecss-mycelium-3.14.10.91	mycelium1@ecss1	not running	8m 10s

All services are started.

Далее необходимо установить паспорт и лицензии в систему:

```
admin@[mycelium1@ecss1]:/$ cluster/storage/ds1/licence/set-passport <passport>
admin@[mycelium1@ecss1]:/$ cluster/storage/ds1/licence/add <license>
```

Выходим из CoCon, выполняем перезагрузку подсистем *ecss-mycelium* и *ecss-ds*, а затем подключаем оставшиеся подсистемы в следующем порядке: *ecss-core*, *ecss-pa-sip*, *ecss-media-server*, *ecss-restfs*, *ecss-mediator*, *ecss-web-conf*.

```
sudo systemctl start ecss-core ecss-pa-sip ecss-mediator ecss-media-server ecss-restfs
ecss-web-conf
```

Возвращаемся в CoCon.

После устанавливается связь MSR- и Core-подсистем. Для этого прописывается команда:

```
admin@[mycelium1@ecss1]:/$ /system/media/resource/declare core1@ecss1 iface msr.ecss1
bond1_ecss1_pa default local true
```

Для проверки выполняем команду `system-status` и видим вывод:

```
admin@mycelium1@ecss1$ system-status
Checking...
```

Node	Release	Erlang nodes	Mnesia nodes	Uptime
core1@ecss1	ecss-core-3.14.10.91	core1@ecss1	not running	1m 59s
ds1@ecss1	ecss-ds-3.14.10.91	ds1@ecss1	ds1@ecss1	4h
md1@ecss1	ecss-mediator-3.14.10.91	md1@ecss1	md1@ecss1	1m 59s
mycelium1@ecss1	ecss-mycelium-3.14.10.91	mycelium1@ecss1	not running	4h
sip1@ecss1	ecss-pa-sip-3.14.10.91	sip1@ecss1	sip1@ecss1	1m 59s

All services are started.

Active media resource selected list specific:

Node	MSR	MSR version	Cc-status	Cc-uptime
core1@ecss1	msr.ecss1	3.14.10.42	connected	00:01:31

Производим настройку адаптера SIP согласно техническому заданию. Определяем группу IP-адресов (IP-set):

```
admin@[mycelium1@ecss1]:/$ /cluster/adapter/sip1/sip/network/set ip_set test_set node-
ip node = sip1@ecss1 ip = 10.0.20.11
```

```
Property "ip_set" successfully changed from:
to
test_set: no ports set
test_set: sip1@ecss1 10.0.20.11
test_set: dscp 0.
```

Далее создаем домен и присваиваем ему созданную группу (IP-set) настроек SIP-адаптера:

```
admin@[mycelium1@ecss1]:/$ domain/declare test_domain --add-domain-admin-privileges --
add-domain-user-privileges
New domain test_domain is declared

domain/test_domain/sip/network/set ip_set [test_set] Property "ip_set" successfully
changed from:
[] to ["test_set"].
```

После создания домена настраиваем:

- [маршрутизацию](#);
- [пользователей](#);
- [абонентов](#);
- [транки](#).

11.2 Первоначальная установка системы с резервированием в кластере из двух серверов

11.2.1 Исходные данные

Исходные данные

В данном разделе использованы специфичные для данного продукта термины, ознакомиться с ними можно [здесь](#).

Техническое задание:

Требуется интеграция программного коммутатора 5 класса ECSS-10 (SSW) на 2-х физических серверах с поддержкой SIP со следующими параметрами на нагрузку группы серверов:

- Максимальное количество абонентов 15000 (MUL – Max user limit);
- Максимальное количество одновременных соединений 2000 (MCL – Max call limit);
- Требуется полный резерв системы ecss (все ноды резервируется на втором сервере);
- Количество сетевых интерфейсов ethernet – 4.

По техническому заданию требуется определить аппаратную платформу.

Таблица 1. Рекомендуемые аппаратные решения

Требования к серверам SSW	Серия аппаратного продукта				
	Light	Light+	Midi	Heavy	Super Heavy
Характеристики системы					
Максимальное количество абонентов	3000	5000	10000	20000	40000

Максимальная нагрузка одновременных соединений class 5	500	800	1500	3000	6000
Максимальная нагрузка одновременных соединений class 4	1500	2400	4500	9000	20000
Характеристики сервера					
Марка	HP (Lenovo)	HP (Lenovo)	HP (Lenovo)	HP (Lenovo)	HP (Lenovo)
Серия	DL20 Gen10 (SR250)	DL20 Gen10 (SR250)/DL 360 Gen10 (SR530)	DL360 Gen10 (SR530/SR630)	DL360 Gen10 (SR630)	DL 360 Gen10 (SR630)
Процессор	Intel Xeon E-2236	Intel Xeon E-2276G/Intel Xeon 4214	Intel Xeon 5220	Intel Xeon 6240	Intel Xeon 8268
Количество процессоров	1	1	1	2	2
Объем ОЗУ	8 ГБ	12 ГБ	16 ГБ	24 ГБ	64 ГБ
HDD	От 3x500 SATA (от 7200 об/мин)	От 3x500 SATA (от 7200 об/мин)	От 3x300 ГБ SAS (от 10000 об/мин)	От 3x600 ГБ SAS (от 10000 об/мин) 2x150Gb SSD	От 6x800 ГБ SSD, 2x300 ГБ M.2 SSD
RAID	No raid board	No raid board	HW Raid, от 1 ГБ cache+battery	HW Raid, от 1 ГБ cache+battery	HW Raid, от 2 ГБ Flashcache, поддержка RAID-5
Дополнительные комплектующие сервера (в базовых комплект не входят)					
Лицензия на удалённое управление	по желанию	по желанию	+	+	+
Резервный источник питания	по желанию	по желанию	+	+	+

Организация хранилища записей разговоров	Дополнительные HDD, объединенные в RAID-5	Дополнительные HDD, объединенные в RAID-5	Лицензия на HW Raid с поддержкой RAID-5, дополнительные HDD для хранения записей	Лицензия на HW Raid с поддержкой RAID-5, дополнительные HDD для хранения записей	Лицензия на HW Raid с поддержкой RAID-5, дополнительные HDD для хранения записей
------------------------------------------	-------------------------------------------	-------------------------------------------	----------------------------------------------------------------------------------	----------------------------------------------------------------------------------	----------------------------------------------------------------------------------

Таблица 2. Пример составления аппаратных требований

Устройство	Требуемый ресурс		Серия аппаратного продукта
	MCL	MUL	
Сервер 1	2000	15000	Heavy
Сервер 2			Heavy

После определения по требованиям проекта составляем предварительную сетевую карту.

Таблица 3. Пример разнесения составляющих по адресному пространству для одной ноды

Имя сервера (хоста)	Роль	Интерфейс	Адрес	Порт
Статические адреса программного коммутатора				
ecss1	Интерфейс управления сервером (порт 2000 ssh)	net.10	10.0.10.11/24	2000
ecss2	Интерфейс управления сервером (порт 2000 ssh)	net.10	10.0.10.12/24	2000
ecss1	Адрес ядра (ecss-core)	net.20	10.0.20.11/24	5000
ecss2	Адрес ядра (ecss-core)	net.20	10.0.20.12/24	5000
ecss1, ecss2	Адрес шлюза	net.10	10.0.10.1	-
ecss1, ecss2	Адреса DNS-серверов	net.10	10.0.10.1, 8.8.8.8	-
ecss1, ecss2	Адреса NTP-серверов	net.10	10.136.16.211, 10.136.16.212	123

Виртуальные адреса программного коммутатора				
ecss1	Виртуальный адрес программного адаптера хоста ecss1	net.20:SIP1	10.0.20.31/24	-
ecss2	Виртуальный адрес программного адаптера хоста ecss2	net.20:SIP2	10.0.20.32/24	-
ecss1	Запасной виртуальный адрес программного адаптера хоста ecss2 на хосте ecss1	net.20:SIP2	10.0.20.32/24	-
ecss2	Запасной виртуальный адрес программного адаптера хоста ecss1 на хосте ecss2	net.20:SIP1	10.0.20.31/24	-
ecss1, ecss2	Виртуальный адрес базы данных MySQL (ecss-mysql)	net.10:MYSQL	10.0.10.10/24	3306

11.2.2 Включение в сеть

Топологию включения сервера в сеть для обеспечения резервирования рекомендуется делать с использованием 2-х коммутаторов.

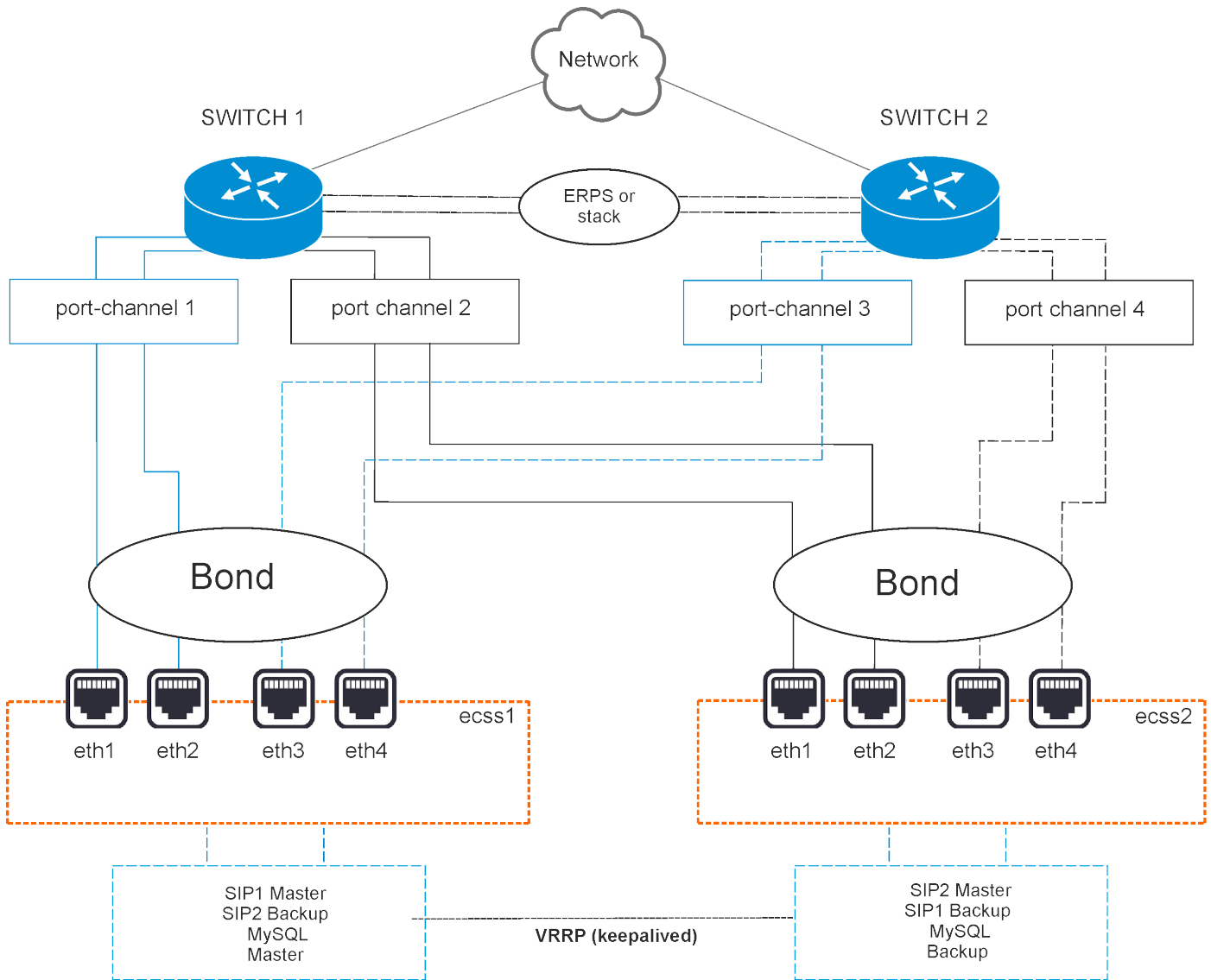


Рисунок 2 – Схема организации связи

Вариант 1. Схема active-backup

Коммутаторы связаны между собой в erps-кольцо.

Все 4 физических сетевых интерфейса каждого сервера объединяются в 1 агрегированный линк (bond). Агрегация портов сервера настраивается в режиме **active-backup**, т.е. в работе всегда находится только 1 сетевой интерфейс. Сетевые интерфейсы каждого сервера попарно включены в коммутаторы, на которых агрегация портов (port-channel) также настраивается в режиме active-backup. Например, eth0 и eth1 каждого сервера включаются в первый коммутатор (port-channel 1 и 2), а eth2 и eth3 (port-channel 3 и 4) – во второй.

Вариант 2. Схема LACP

Коммутаторы связаны между собой в стек. Стек должен логически работать как один коммутатор, способный обеспечивать агрегацию портов в режиме LACP между разными физическими коммутаторами. Пример – коммутаторы MES-3124 со специализированной прошивкой.

Все 4 физических сетевых интерфейса также объединяются в 1 агрегированный линк (bond). Агрегация портов сервера настраивается в режиме **802.3ad**. Создаются агрегированные группы сетевых карт с одинаковой скоростью и дуплексом. При таком объединении передача задействует все каналы в активной агрегации согласно стандарту IEEE 802.3ad. Выбор того, через какой интерфейс необходимо

отправлять пакет, определяется политикой. По умолчанию это XOR-политика, можно использовать «xmit_hash» политику. Подробнее – в разделе [Приложение E. Netplan](#).

Требования:

- Поддержка Ehtool в драйвере для получения информации о скорости и дуплексе на каждом сетевом интерфейсе;
- Поддержка на коммутаторе стандарта IEEE 802.3ad;

Сетевые интерфейсы сервера также попарно включены в коммутаторы, на которых агрегация портов (port-channel) тоже настраивается в режиме LACP. Например, eth0 и eth1 каждого сервера включаются в первый коммутатор (port-channel 1 и 2), а eth2 и eth3 (port-channel 3 и 4) – во второй.

11.2.3 Настройка сети

Согласно параметрам, заданным в техническом задании, выполним сетевые настройки. В данном примере считается, что уже установлена необходимая операционная система. О том, как это сделать, смотрите в разделе [Инсталляция системы](#).

Рекомендуется разделять трафик, используемый в разных целях. Например, трафик управления и трафик VoIP. Для этого создаются 2 или более vlan. При небольшой нагрузке для простоты можно обойтись и одним vlan, но это может в будущем доставлять неудобства при снятии дампов трафика и его анализе. В соответствии с техническим заданием на vlan настраиваются IP-адреса самого хоста, шлюзы, DNS и маршрутизация в другие сети.

В данном примере согласно ТЗ будем использовать адреса (в скобках – различия для ecss2):

- 10.0.10.11(12)/24 – для управления, vlan 10;
- 10.0.20.21(22)/24 – core, vlan 20;
- 10.0.20.31(32)/24 – виртуальные (vrrp) для VoIP;
- 10.0.10.10 – виртуальный (vrrp) адрес сервера MySQL;
- 10.0.10.1 – шлюз и dns для доступа во внешнюю сеть;
- 10.0.20.1 – шлюз в подсеть 10.0.3.0/24;
- 10.136.16.211, 10.136.16.212 – адреса NTP-серверов, доступ до которых осуществляется через шлюз 10.0.10.1.

Внутри серверной платформы существует адресная структура и используются внутренние адреса для взаимодействия между подсистемами (нодами) в кластере. Например, внутренний адрес для кластера на одном сервере 10.0.20.21, при этом ядро (ecss-core) взаимодействует с сервером обработки мультимедиа-данных ([ecss-media-server](#)). Их взаимодействие происходит с использованием одного и того же адреса, но у каждой программной части свой транспортный порт: ecss-core – 5000, ecss-msr – 5040.

Для всех нод кластера определяется единый адрес для обращения к базе данных MySQL, например, адрес ecss-mysql 10.0.10.10. Таким образом, выполняется условие единообразия, при котором все ноды кластера имеют полностью одинаковые данные о текущем состоянии динамических составляющих программного коммутатора (например, история вызовов).

Сначала настраиваются сетевые интерфейсы. В Ubuntu18 для настройки используется утилита netplan:

```
sudo nano /etc/netplan/ecss_netplan.yaml
```

В конфигурациях для каждого хоста в первую очередь мы объявляем секцию *ethernets*, в которой описываются существующие в системе ethernet-интерфейсы, используемые в дальнейшем. Для каждого интерфейса необходимо отключить использование динамического распределения адресов (DHCP).

Следующим разделом описываются агрегированные каналы – *bonds*. В зависимости от выбранного варианта включения в сеть настраивается режим резерва 1:1 (*active - backup*) или LACP (802.3ad).

Далее настраиваются vlan, на которых опционально определяются шлюзы для связи с внешним миром и адреса DNS-серверов, а также непосредственно IP-адреса для каждого интерфейса.

❗ ВАЖНО

Обратите внимание, что во время редактирования netplan необходимо соблюдать правила разметки YAML:

- Обязательное наличие двух пробелов перед каждой строкой (кроме network);
- Каждая подсекция дополнительно сдвигается на 2 пробела:

```
→ Секция                |network
→ Подсекция            |  'bonds:
→ Подсекция описания  |    'bonded_one:
→ и т.д.                |    ...
```

- Перед знаком ":" нет пробела, после — один пробел;
- Перед знаком "-" количество пробелов такое, как будто начинается новый подраздел, после — один пробел.

Пример netplan для режима active-backup

Netplan для интерфейсов сервера ecss1 (/etc/netplan/ecss_netplan.yaml)

```
# Netplan для хоста ecss1 программного коммутатора
# Обратите внимание на обязательное наличие минимум двух пробелов в каждой строке и секции (кроме строки секции network)

network:
  version: 2 # Версия netplan
  renderer: networkd # Исполнитель конфигураций netplan
  ethernets: # Раздел описания интерфейсов ethernet
    eth0: # Название интерфейса
      dhcp4: no # Отключаем на интерфейсах динамическое распределение IP-адреса
    eth1:
      dhcp4: no
    eth2:
      dhcp4: no
    eth3:
      dhcp4: no

  bonds: # Раздел описания связанных (bonding) интерфейсов
    bond1: # Название bonding-интерфейса
      interfaces: # Секция определения связанного интерфейса
        - eth0
        - eth1
```

Netplan для интерфейсов сервера ecss2 (/etc/netplan/ecss_netplan.yaml)

```
# Netplan для хоста ecss2 программного коммутатора
# Обратите внимание на обязательное наличие минимум двух пробелов в каждой строке и секции (кроме строки секции network)

network:
  version: 2 # Версия netplan
  renderer: networkd # Исполнитель конфигураций netplan
  ethernets: # Раздел описания интерфейсов ethernet
    eth0: # Название интерфейса
      dhcp4: no # Отключаем на интерфейсах динамическое распределение IP-адреса
    eth1:
      dhcp4: no
    eth2:
      dhcp4: no
    eth3:
      dhcp4: no

  bonds: # Раздел описания связанных (bonding) интерфейсов
    bond1: # Название bonding-интерфейса
      interfaces: # Секция определения связанного интерфейса
        - eth0
        - eth1
        - eth2
```

Netplan для интерфейсов сервера ecss1 (/etc/netplan/ecss_netplan.yaml)	Netplan для интерфейсов сервера ecss2 (/etc/netplan/ecss_netplan.yaml)
<pre> - eth2 - eth3 parameters: # Раздел определения параметров связанного интерфейса mode: active-backup # Режим резервирования 1:1 mii-monitor-interval: 100 # интервал мониторинга интерфейса(мс) primary: eth0 # Секция определения основного интерфейса optional: false # Определение, является ли интерфейс обязательным при старте vlans: net.10: # Интерфейс управления id: 10 link: bond1 addresses: [10.0.10.11/24] gateway4: 10.0.10.1 # Адрес шлюза nameservers: addresses: [10.0.10.1, 8.8.8.8] # Адреса серверов DNS routes: # Маршрутизация в подсеть к NTP - to: 10.136.16.0/24 via: 10.0.10.1 # Адрес шлюза в данную подсеть on-link: true # Определяет, что указанные маршруты непосредственно связаны с интерфейсом net.20: # Интерфейс для VoIP id: 20 link: bond1 addresses: [10.0.20.11/24] routes: - to: 10.0.3.0/24 via: 10.0.20.1 on-link: true </pre>	<pre> - eth3 parameters: # Раздел определения параметров связанного интерфейса mode: active-backup # Режим резервирования 1:1 mii-monitor-interval: 100 # интервал мониторинга интерфейса(мс) primary: eth0 # Секция определения основного интерфейса optional: false # Определение, является ли интерфейс обязательным при старте vlans: network.10: # Интерфейс управления id: 10 link: bond1 addresses: [10.0.10.12/24] gateway4: 10.0.10.1 # Адрес шлюза nameservers: addresses: [10.0.10.1, 8.8.8.8] # Адреса серверов DNS routes: # Маршрутизация в подсеть к NTP - to: 10.136.16.0/24 via: 10.0.10.1 # Адрес шлюза в данную подсеть on-link: true # Определяет, что указанные маршруты непосредственно связаны с интерфейсом network.20: # Интерфейс для VoIP id: 20 link: bond1 addresses: [10.0.20.12/24] routes: - to: 10.0.3.0/24 via: 10.0.20.1 on-link: true </pre>

Пример netplan для режима 802.3ad

Netplan для интерфейсов сервера ecss1 (/etc/netplan/ecss_netplan.yaml)

```
# Netplan для хоста ecss1 программного коммутатора
# Обратите внимание на обязательное наличие минимум двух пробелов в каждой строке и секции (кроме строки секции network)

network:
  version: 2 # Версия netplan
  renderer: networkd # Исполнитель конфигураций netplan
  ethernets: # Раздел описания интерфейсов ethernet
    eth0: # Название интерфейса
      dhcp4: no
    eth2:
      dhcp4: no
    eth3:
      dhcp4: no

  bonds: # Раздел описания связанных (bonding) интерфейсов
    bond1: # Название bonding-интерфейса
      interfaces: # Секция определения связанного интерфейса
        - eth0
        - eth1
        - eth2
        - eth3
      parameters: # Раздел определения параметров связанного интерфейса
        mode: 802.3ad # Режим LACP
        mii-monitor-interval: 100 # интервал мониторинга интерфейса(мс)
        primary: eth0 # Секция определения основного интерфейса
        optional: false # Определение, является ли интерфейс обязательным при старте

  vlans:
    net.10: # Интерфейс управления
      id: 10
      link: bond1
```

Netplan для интерфейсов сервера ecss2 (/etc/netplan/ecss_netplan.yaml)

```
# Netplan для хоста ecss2 программного коммутатора
# Обратите внимание на обязательное наличие минимум двух пробелов в каждой строке и секции (кроме строки секции network)

network:
  version: 2 # Версия netplan
  renderer: networkd # Исполнитель конфигураций netplan
  ethernets: # Раздел описания интерфейсов ethernet
    eth0: # Название интерфейса
      dhcp4: no # Отключаем на интерфейсах динамическое распределение IP-адреса
    eth1:
      dhcp4: no
    eth2:
      dhcp4: no
    eth3:
      dhcp4: no

  bonds: # Раздел описания связанных (bonding) интерфейсов
    bond1: # Название bonding-интерфейса
      interfaces: # Секция определения связанного интерфейса
        - eth0
        - eth1
        - eth2
        - eth3
      parameters: # Раздел определения параметров связанного интерфейса
        mode: 802.3ad # Режим LACP
        mii-monitor-interval: 100 # интервал мониторинга интерфейса(мс)
        primary: eth0 # Секция определения основного интерфейса
        optional: false # Определение, является ли интерфейс обязательным при старте

  vlans:
    network.10: # Интерфейс управления
      id: 10
      link: bond1
      addresses: [10.0.10.12/24]
      gateway4: 10.0.10.1 # Адрес
```


Netplan для интерфейсов сервера ecss1 (/etc/netplan/ecss_netplan.yaml)	Netplan для интерфейсов сервера ecss2 (/etc/netplan/ecss_netplan.yaml)
<pre> addresses: [10.0.10.11/24] gateway4: 10.0.10.1 # Адрес шлюза nameservers: addresses: [10.0.10.1, 8.8.8.8] # Адреса серверов DNS routes: # Маршрутизация в подсеть к NTP - to: 10.136.16.0/24 via: 10.0.10.1 # Адрес шлюза в данную подсеть on-link: true # Определяет, что указанные маршруты непосредственно связаны с интерфейсом net.20: # Интерфейс для VoIP id: 20 link: bond1 addresses: [10.0.20.11/24] routes: - to: 10.0.3.0/24 via: 10.0.20.1 on-link: true </pre>	<pre> шлюза nameservers: addresses: [10.0.10.1, 8.8.8.8] # Адреса серверов DNS routes: # Маршрутизация в подсеть к NTP - to: 10.136.16.0/24 via: 10.0.10.1 # Адрес шлюза в данную подсеть on-link: true # Определяет, что указанные маршруты непосредственно связаны с интерфейсом network.20: # Интерфейс для VoIP id: 20 link: bond1 addresses: [10.0.20.12/24] routes: - to: 10.0.3.0/24 via: 10.0.20.1 on-link: true </pre>

Применим установленные параметры командой:

```
sudo netplan apply
```

Просмотреть получившиеся настройки можно при помощи команд **ifconfig** или **ip a**:

Настройка /etc/hosts

После настройки *netplan* необходимо указать, что внутренний адрес *192.168.1.X* принадлежит соответствующему серверу *ecssX*. Для этого настраиваем */etc/hosts*:

Настройка hosts для ecss1 (/etc/hosts)	Настройка hosts для ecss2 (/etc/hosts)
<pre> sudo nano /etc/hosts 127.0.0.1 localhost # Адрес локальной петли, используется некоторыми сервисами ecss 10.0.10.11 ecss1 # Адрес хоста 10.0.10.12 ecss2 # Внутренний адрес другого хоста </pre>	<pre> sudo nano /etc/hosts 127.0.0.1 localhost # Адрес локальной петли, используется некоторыми сервисами ecss 10.0.10.12 ecss2 # Адрес хоста 10.0.10.11 ecss1 # Внутренний адрес другого сервера </pre>

Теперь, если вызвать утилиту *ping* на *ecssX*, можно обратиться к соседнему серверу.

Обращение к ecss2 с ecss1	Обращение к ecss1 с ecss2
<pre>sasha@ecss1:~\$ ping ecss2 PING ecss2 (10.0.10.12) 56(84) bytes of data. 64 bytes from ecss2 (10.0.10.12): icmp_seq=1 ttl=64 time=0.047 ms</pre>	<pre>sasha@ecss1:~\$ ping ecss1 PING ecss1 (10.0.10.11) 56(84) bytes of data. 64 bytes from ecss1 (10.0.10.11): icmp_seq=1 ttl=64 time=0.032 ms</pre>

Обновление программного обеспечения операционной системы

Для установки системы ECSS-10 необходимо добавить репозиторий ЭЛТЕКС:

```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/bionic/3.14 stable main extras
external' > /etc/apt/sources.list.d/eltex-ecss10-stable.list"
```

- ✔ Обратите внимание, что требуется указать верную версию операционной системы при добавлении репозитория ЭЛТЕКС:

```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/molensk/3.14 stable
main extras external' > /etc/apt/sources.list.d/eltex-ecss10-stable.list"
```

Далее необходимо выполнить импорт ключа командой:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 33CB2B750F8BB6A5
```

Для обновления ОС необходимо выполнить команды:

```
sudo apt update
sudo apt upgrade
```

11.2.4 Установка и настройка программного обеспечения

Устанавливаем все [предложенные пакеты](#):

```
sudo apt install ntp tcpdump vlan dnsmasq aptitude atop ethtool htop iotop mc minicom
mtr-tiny nmap pptpd pv screen ssh tftpd vim sngrep tshark cpanminus gnuplot libgraph-
easy-perl debconf-utils
```

Устанавливаем также пакеты для системы с резервированием:

```
sudo apt install ifenslave-2.6 keepalived attr
```

Установка пакета `ecss-mysql`

Для [установки](#) выполняем команду:

```
sudo apt install ecss-mysql
```

Настройка пакета `ecss-dns-env`

Перед установкой пакета `ecss-mysql` вам будет предложено настроить переменные окружения для сервисов в `dnsmasq`. Настройщик предложит выбрать разделы для настройки. Нужно выбрать `broker` и `mysql`.

Вопрос	Ответы для <code>ecss1</code>	Ответы для <code>ecss2</code>
Адрес первичного брокера ([Primary broker] Enter ip)	10.0.10.11	10.0.10.11
Адрес вторичного брокера ([Secondary broker] Enter ip)	10.0.10.12	10.0.10.12
Адрес <code>mysql</code> ([Mysql] Enter ip)	10.0.10.10	10.0.10.10

ВАЖНО

Для изменения `ip`-адресов рекомендуется использовать команду:
`sudo dpkg-reconfigure ecss-dns-env`

Настройка пакета `ecss-mysql`

Во время установки настройщик задаст вопросы, ответы указаны в таблице ниже. Обратите внимание, что **пароль является одинаковым** для обоих хостов, на которых устанавливается `mysql`.

Вопросы	Ответы для <code>ecss1</code>	Ответы для <code>ecss2</code>
Адресная маска для MySQL (IP pattern for MySQL permission)	10.0.10.%	10.0.10.%
Логин пользователя (Login for MySQL root)	root	root
Пароль пользователя MySQL (Password for MySQL root)	PASSWORD	PASSWORD

Изменение пути по умолчанию — согласиться с изменением конфигурационного файла для внесения пути до баз данных `ecss-mysql`, введя "Y".

Базы данных `mysql`, используемых системой ECSS-10, после установки будут храниться по пути `/var/lib/ecss-mysql`. Проверяем наличие файлов в папке:

```
ls -l /var/lib/ecss-mysql/
total 36
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_address_book
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:37 ecss_audit
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_calls_db
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_dialer_db
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_meeting_db
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_statistics
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 ecss_subscribers
drwxr-xr-x 2 mysql mysql 4096 Sep 26 13:36 history_db
drwxr-xr-x 2 mysql mysql 4096 Sep 26 14:32 web_conf
```

Проверяем, что сервер запущен:

```
systemctl status mysql.service

● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/mysql.service.d
            └─override.conf
   Active: active (running) since Sun 2022-02-06 15:25:15 +07; 3 days ago
   Process: 3766 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/
mysqld.pid (code=exited, status=0/SUCCESS)
   Process: 3736 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited,
status=0/SUCCESS)
   Main PID: 3783 (mysqld)
     Tasks: 87 (limit: 4915)
    CGroup: /system.slice/mysql.service
            └─3783 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
```

Далее следует поддержать доступ между серверами с ecss-mysql через ssh по rsa-ключам без использования пароля.

На хосте ecss1 сгенерируем rsa-ключ следующей командой (выполняем команду без sudo, чтобы ключ сгенерировался для текущего пользователя):

```
ssh-keygen
ssh-copy-id tester@ecss2
```

Таким же образом генерируем rsa-ключ на хосте ecss2, заменяя host-часть на ecss1.

```
ssh-keygen
ssh-copy-id tester@ecss1
```

Далее следует запустить скрипт [репликации](#) БД mysql на ecss1:

```
sudo /usr/lib/ecss/ecss-scripts/mysql-replication/install_replication.sh
```

Перед выполнением скрипта необходимо определить некоторые параметры. Пример ответа на вопросы для разных хостов приведен ниже. Следует обратить внимание, что пароль **PASSWORD** — это тот же пароль, который задавался выше. Виртуальный адрес mysql задаем вручную. Причина приведена в разделе "[Настройка виртуального адреса для MySQL](#)".

Вопросы	Ответы для ecss1
Логин для доступа к БД	root
Пароль для доступа к БД	PASSWORD
Логин для пользователя репликации	replica
Пароль для пользователя репликации	replica
Адрес первого хоста	10.0.10.11
Адрес второго хоста	10.0.10.12
Имя второго хоста	ecss2
Имя пользователя на втором хосте	tester
Mediator IP	127.0.0.1
Порт SNMP	162
Создать конфигурацию keepalived	нет

После работы скрипта можно проверить, что в MySQL создан пользователь *replica@10.0.10.%* и *replica@%* на обоих хостах:

```
mysql -uroot -ppassword
mysql> SELECT user,host FROM mysql.user;
```

Среди всех пользователей можно увидеть такую запись:

```
+-----+
+-----+
|          user          |
| host          |
+-----+
+-----+
| replica          |
| 10.0.10.%          |
+-----+
+-----+
```

Проверяем состояние реплики:

```
sudo mysql -uroot -p -e 'show slave status \G;' | grep -E "Slave_IO_Running:|
Slave_SQL_Running:"
Enter password:
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

Редактирование `keepalived.conf`

Следующий шаг — редактирование глобального конфигурационного файла **keepalived.conf**. На обоих хостах содержимое файла одинаковое:

```
sudo nano /etc/keepalived/keepalived.conf
```

```
global_defs {  
    vrrp_version 3          # версия протокола VRRP (2 или 3)  
    script_user nobody     # ограниченный в правах системный пользователь, от которого будут  
запускаться скрипты проверки доступности  
    enable_script_security # не запускать скрипты от root, если часть пути к ним доступна на  
запись для обычных пользователей  
}  
  
include /etc/keepalived/sip.conf  
include /etc/keepalived/mysql.conf
```

Поскольку при создании репликации не была задействована автоматическая генерация конфигурации для mysql, то в конфигурационном файле нужно сделать ссылку на mysql.conf:

```
include /etc/keepalived/mysql.conf
```

Далее на обоих хостах создаются файлы **/etc/keepalived/mysql.conf**

Создание VRRP для MySQL.

/etc/keepalived/mysql.conf для ecss1	/etc/keepalived/mysql.conf для ecss2
<pre># Конфигурация mysql первой ноды: vrrp_script check_mysql { script "/usr/bin/mysql --defaults-file= /etc/mysql/debian.cnf -e 'SELECT 1;'" user root interval 2 fall 1 timeout 2 } vrrp_instance MySQL { state MASTER # Исходное состояние при старте interface net.10 # Имя сетевого интерфейса, на котором будет работать протокол VRRP virtual_router_id 10 # Уникальный идентификатор роутера (0..255) priority 100 # Приоритет (0..255) чем выше - тем больше advert_int 1 # Интервал рассылки уведомлений () preempt_delay 60 # Интервал ожидания мастера при старте демона (с) при исходном состоянии BACKUP unicast_src_ip 10.0.10.11 # Собственный реальный IP- unicast_peer { 10.0.10.12 # Реальный IP-адрес соседа } virtual_ipaddress { # Виртуальный IP-адрес и маска # dev - сетевой интерфейс, на котором будет поднят виртуальный адрес # label - метка виртуального интерфейса (для удобства идентификации) 10.0.10.10/24 dev net.10 label net. 10:mysql } track_script { check_mysql } }</pre>	<pre># Конфигурация mysql второй ноды: vrrp_script check_mysql { script "/usr/bin/mysql --defaults- file=/etc/mysql/debian.cnf -e 'SELECT 1;'" user root interval 2 fall 1 timeout 2 } vrrp_instance MySQL { state BACKUP # Исходное состояние при старте interface net.10 # Имя сетевого интерфейса, на котором будет работать протокол VRRP virtual_router_id 10 # Уникальный идентификатор роутера (0..255) priority 50 # Приоритет (0..255) чем выше - тем больше advert_int 1 # Интервал рассылки уведомлений () preempt_delay 60 # Интервал ожидания мастера при старте демона (с) при исходном состоянии BACKUP unicast_src_ip 10.0.10.12 # Собственный реальный IP- unicast_peer { 10.0.10.11 # Реальный IP-адрес соседа } virtual_ipaddress { # Виртуальный IP-адрес и маска # dev - сетевой интерфейс, на котором будет поднят виртуальный адрес # label - метка виртуального интерфейса (для удобства идентификации) 10.0.10.10/24 dev net.10 label net.10:mysql } track_script { check_mysql } }</pre>

В данной конфигурации мы задаем ID для виртуального маршрутизатора, который будет являться балансером для своего хоста. Важно, чтобы *virtual_router_id* **совпадали** у обоих хостов.

После проверки необходимо перезапустить сервис *keepalived*:

```
sudo systemctl restart
keepalived.service
```

Установка ecss-node

Устанавливаем пакет ecss-node:

```
sudo apt install ecss-node
```

Во время установки будет предложено настроить некоторые параметры, пример ответов следует ниже.

Подробнее о настройке NTP в разделе "[Синхронизация времени на серверах](#)".

Вопросы	Ответы для ecss1	Ответы для ecss2
Выключить ежедневные обновления (Do you want turn off apt-daily update ?)	Yes	Yes
Установить настройки по умолчанию для БД (Set DB config to default ?)	Yes	Yes
Взводить аварию ECSS-10 при заполнении раздела MySQL (Set alarm true when MYSQL DB overloads)	Yes	Yes
NTP: Вы хотите использовать настройки для кластера? (NTP: Do you want use settings for cluster ?)	Yes	Yes
Установить приоритет NTP для кластера (NTP: Set stratum for cluster)	7	7
Внешние NTP-серверы через пробел: (External NTP servers through a space)	10.136.16.211 10.136.16.212	
NTP: Вы хотите использовать другие сервера для синхронизации времени? (NTP: Do you want to use other servers for time synchronization?)	Yes	Yes
NTP: укажите локальные серверы для синхронизации через пробел: (NTP: Indicate local servers for synchronization separated a space:)	10.0.10.12	10.0.10.11
NTP: Вы хотите определить сети, у которых должен быть доступ до ntp, вручную? (NTP: Do you want to define manually which networks should have access to ntp?)	Yes	Yes
NTP: Сети, у которых должен быть доступ до ntp через пробел: Формат: <сетевой адрес> маска (x.x.x.x 255.255.255.0)	10.0.10.0 255.255.255.0	10.0.10.0 255.255.255.0
Установить утилиты для работы с cdr	No	No

Установка остальных пакетов ecss и их настройка

Далее на обоих хостах выполняем установку всех необходимых пакетов (подробнее по установке необходимых и дополнительных пакетов в разделе "[Инсталляция пакетов ECSS](#)"):

```
sudo apt install ecss-media-server ecss-media-resources ecss-web-conf ecss-restfs
```

Для медиасервера (*ecss-media-server/MSR*) возможно начальное конфигурирование с записью параметров в [файл конфигурации](#), для этого нужно провести конфигурацию `transport-port`, `transport bind-addr`, `mcc bind-address`, `mcc bind-port`:

Вопросы <i>ecss-media-server</i>	Ответы для <i>ecss1</i>	Ответы для <i>ecss2</i>
Введите порт (Enter)	5040	5040
Введите bind-ip адрес (Enter)	10.0.20.11	10.0.20.12
Введите адрес контрольного канала (bind-addr)	10.0.20.11	10.0.20.12
Введите порт контрольного канала	5700	5700
Выберите режим конфигурации (Choose config mode)	auto	auto
Установить настройки по умолчанию:	yes	yes
Введите имя (Enter)	msr.ecss1	msr.ecss2
Введите адрес (Enter)	10.0.20.11	10.0.20.12
Введите порт (Enter)	5000	5000

После формирования конфигураций по умолчанию переходим в директорию, в которой находятся файлы конфигурации и производим их проверку:

```
cd /etc/ecss/ecss-media-server/  
cat config.xml  
cat conf.d/default.xml
```

Внутри лежит конфигурация для `msr`: `config.xml`, в директории `conf.d` лежит конфигурация `default.xml`. По своей сути `default.xml` — это дополнение `config.xml`, которое определяет секцию аккаунтов. Это сделано для того, чтобы после обновлений пакета данная конфигурация оставалась неизменной. Вид `config.xml` представлен здесь: [файл конфигурации](#). Рассмотрим секцию аккаунтов (файл `default.xml`):

Настройка msr для ecss1 (/etc/ecss/ecss-media-server/conf.d/default.xml)	Настройка msr для ecss2 (/etc/ecss/ecss-media-server/conf.d/default.xml)
<pre><?xml version="1.0"?> <config> <accounts> <dynamic msr_name="msr.ecss1" realm="sip:10.0.20.11:5000" dtmf_mode="rfc+inband+info" auth_name="user" auth_password="password"/> <dynamic msr_name="msr.ecss1" realm="sip:10.0.20.12:5000" dtmf_mode="rfc+inband+info" auth_name="user" auth_password="password"/> </accounts> </config></pre>	<pre><?xml version="1.0"?> <config> <accounts> <dynamic msr_name="msr.ecss2" realm="sip:10.0.20.11:5000" dtmf_mode="rfc+inband+info" auth_name="user" auth_password="password"/> <dynamic msr_name="msr.ecss2" realm="sip:10.0.20.12:5000" dtmf_mode="rfc+inband+info" auth_name="user" auth_password="password"/> </accounts> </config></pre>

В ней указаны текущие настройки, согласно которым происходит регистрация msr на core.

Основными параметрами здесь являются: `msr_name` и `realm`:

- `msr_name` – параметр, определяющий название msr. (рекомендуется задавать msr. и к какому хосту он принадлежит, например `msr.ecss1`);
- `realm` – определяет адрес для регистрации на ядре. Точка входа по умолчанию – порт 5000.

Описание настройки остальных сервисов приведено в разделе ["Инсталляция дополнительных необязательных пакетов"](#).

Настройка VRRP для SIP-адаптера

Выполним настройку VRRP для SIP-адаптера, для этого на обоих хостах создаем файлы следующего содержания:

sudo nano /etc/keepalived/sip.conf	
etc/keepalived/sip.conf для ecss1	etc/keepalived/sip.conf для ecss2
<pre>vrrp_script check_sip { script "/usr/bin/ecss_pa_sip_port 65535" interval 2 timeout 2 } # Конфигурация адреса для первого виртуального адреса SIP- vrrp_instance SIP1 { state MASTER # Исходное состояние при старте interface net.20 # Имя сетевое интерфейса, на котором будет работать протокол VRRP virtual_router_id 31 # Уникальный идентификатор роутера (0..255) priority 100 # Приоритет (0..255) чем выше - тем больше advert_int 1 #</pre>	<pre>vrrp_script check_sip { script "/usr/bin/ecss_pa_sip_port 65535" interval 2 timeout 2 } # Конфигурация адреса для первого виртуального адреса SIP- vrrp_instance SIP1 { state BACKUP # Исходное состояние при старте interface net.20 # Имя сетевое интерфейса, на котором будет работать протокол VRRP virtual_router_id 31 # Уникальный идентификатор роутера (0..255) priority 50 # Приоритет (0..255) чем выше - тем больше advert_int 1 #</pre>

etc/keepalived/sip.conf для ecss1	etc/keepalived/sip.conf для ecss2
Интервал рассылки уведомлений () preempt_delay 60 #	Интервал рассылки уведомлений () preempt_delay 60 #
Интервал ожидания мастера при старте демона (с) при исходном состоянии BACKUP	Интервал ожидания мастера при старте демона (с) при исходном состоянии BACKUP
unicast_src_ip 10.0.20.11 #	unicast_src_ip 10.0.20.12 #
Собственный реальный IP-unicast_peer { 10.0.20.12 #	Собственный реальный IP-unicast_peer { 10.0.10.11 #
Реальный IP-адрес соседа } virtual_ipaddress { # Виртуальный IP-адрес и маска # dev - сетевой интерфейс, на котором будет поднят виртуальный адрес # label - метка виртуального интерфейса (для удобства идентификации) 10.0.10.31/24 dev net.20 label net.20:SIP1 }	Реальный IP-адрес соседа } virtual_ipaddress { # Виртуальный IP-адрес и маска # dev - сетевой интерфейс, на котором будет поднят виртуальный адрес # label - метка виртуального интерфейса (для удобства идентификации) 10.0.20.31/24 dev net.20 label net.20:SIP1 }
track_script { check_sip }	track_script { check_sip }
# Конфигурация адреса для второго виртуального адреса SIP-	# Конфигурация адреса для второго виртуального адреса SIP-
vrrp_instance SIP2 { state BACKUP # Исходное состояние при старте interface net.20 # Имя сетевого интерфейса, на котором будет работать протокол VRRP virtual_router_id 32 # Уникальный идентификатор роутера (0..255) priority 50 # Приоритет (0..255) чем выше - тем больше advert_int 1 # Интервал рассылки уведомлений () preempt_delay 60 # Интервал ожидания мастера при старте демона (с) при исходном состоянии BACKUP	vrrp_instance SIP2 { state MASTER # Исходное состояние при старте interface net.20 # Имя сетевого интерфейса, на котором будет работать протокол VRRP virtual_router_id 32 # Уникальный идентификатор роутера (0..255) priority 100 # Приоритет (0..255) чем выше - тем больше advert_int 1 # Интервал рассылки уведомлений () preempt_delay 60 # Интервал ожидания мастера при старте демона (с) при исходном состоянии BACKUP
unicast_src_ip 10.0.20.11 #	unicast_src_ip 10.0.20.12 #
Собственный реальный IP-unicast_peer { 10.0.20.12 #	Собственный реальный IP-unicast_peer { 10.0.20.11 #
Реальный IP-адрес соседа } virtual_ipaddress { # Виртуальный IP-адрес и маска # dev - сетевой интерфейс, на котором будет поднят виртуальный адрес # label - метка виртуального	Реальный IP-адрес соседа } virtual_ipaddress { # Виртуальный IP-адрес и маска # dev - сетевой интерфейс, на котором будет поднят виртуальный адрес # label - метка виртуального

etc/keepalived/sip.conf для ecss1	etc/keepalived/sip.conf для ecss2
<pre>интерфейса (для удобства идентификации) 10.0.20.32/24 dev net.20 label net. 20:SIP2 } track_script { check_sip } }</pre>	<pre>интерфейса (для удобства идентификации) 10.0.20.32/24 dev net.20 label net. 20:SIP2 } track_script { check_sip } }</pre>

В данном случае мы добавили поддержку виртуальных интерфейсов со связью Master - Backup. Для **ecss1** основным будет являться **net.20:SIP1**, а запасным — **net.20:SIP2** соответственно. Важно учесть, что в конфигурации учтено использование адресных переменных. В секции interface требуется указать, на каком интерфейсе будут слушаться VRRP-сообщения, а в секции virtual_ipaddress требуется указать интерфейс, на котором будет восстановлен виртуальный адрес.

Выполняем перезапуск keepalived:

```
sudo systemctl restart
keepalived.service
```

Дальнейшая настройка

mycelium.config

Установка имени кластера производится в конфигурации пакета ecss-mycelium `/etc/ecss/ecss-mycelium/mycelium.config`:

```
sudo nano /etc/ecss/ecss-
mycelium/mycelium1.config
```

Выполним настройку на обоих хостах.

Настройка имени кластера (/etc/ecss/ecss-mycelium/mycelium1.config)

```
%%% -*- mode:erlang -*-
%%% Warning - this config file *must* end with <dot><whitespace>
[
    {mycelium_broker, [
        {cluster_name, test_cluster},          %% Здесь необходимо установить имя кластера для обоих хостов. Имя
        кластера произвольное и должно быть одинаково на обоих хостах
    ]}
]
....
```

epmd

Выполняем настройку [epmd](#):

```
systemctl edit epmd.service
```

Создаем новый сервис и редактируем его:

Настройка epmd ecss1	Настройка epmd ecss2
[Service] Environment="ERL_EPMD_ADDRESS=127.0.1.1, 10.0.10.11"	[Service] Environment="ERL_EPMD_ADDRESS=127.0.1.1,10.0.1 0.12"

Перезапускаем сервисы:

```
systemctl daemon-reload
systemctl restart
epmd.service
```

glusterfs

Выполним настройку *glusterfs* для *ecss-restfs* на первом хосте (*ecss1*), для этого устанавливаем пакеты *glusterfs-server* и *attr* на обоих хостах:

```
sudo aptitude install
glusterfs-server attr
```

После установки выполняем создание связи с удаленным виртуальным хостом:

```
sudo gluster peer probe 10.0.10.12
```

Проверяем наличие созданной связи:

```
sudo gluster peer status

Number of Peers: 1

Hostname: 10.0.10.12
Uuid: 569c4730-a3a7-4d29-
a132-b1bcdad792d8
State: Peer in Cluster
(Connected)
```

Далее создаем кластер для репликации, запускаем репликацию и смотрим ее статус:

```
sudo gluster volume create
ecss_volume replica 2
transport tcp 110.0.10.11:/
var/lib/ecss/glusterfs
10.0.10.12:/var/lib/ecss/
glusterfs force
sudo gluster volume start
ecss_volume
sudo gluster volume info
```

Таким образом, статус репликации будет выглядеть следующим образом:

```
Volume Name: ecss_volume
Type: Replicate
Volume ID: 3bfc7587-0f85-48ed-9612-21f0d79c6e52
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: 10.0.10.11:/var/lib/ecss/glusterfs
Brick2: 10.0.10.12:/var/lib/ecss/glusterfs
Options Reconfigured:
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

Монтируем раздел glusterfs, для этого создаем новый сервис:

```
/etc/systemd/system/ecss-
glusterfs-mount.service
```

Добавляем в него следующую конфигурацию:

Настройка /etc/systemd/system/ecss-glusterfs-mount.service

```
[Unit]
Description=mount glusterfs
After=network.target
Requires=network.target

[Service]
RemainAfterExit=no
Type=forking
RestartSec=10s
Restart=always
ExecStart=/sbin/mount.glusterfs localhost:/ecss_volume /var/lib/ecss/restfs -o fetch-
attempts=10
ExecStop=/bin/umount /var/lib/ecss/restfs

[Install]
WantedBy=multi-user.target
```

Перезапускаем сервисы:

```
sudo systemctl daemon-reload
sudo systemctl restart ecss-
glusterfs-mount.service
```

Проверяем, что раздел смонтировался:

```
df -h
<...>
localhost:/ecss_volume 253G
3.0G 239G 2% /var/lib/ecss/
restfs
```

snmpd

Настройка SNMP описана в разделе "[Управление и мониторинг по протоколу SNMP](#)".

Настройка безопасности. SSH

Выполняем настройку сервера SSH:

```
sudo nano /etc/ssh/sshd_config
```

Укажем в конфигурационном файле порт и адрес, по которому можно обратиться к серверу:

Настройка ssh для ecss1 (/etc/ssh/sshd_config)	Настройка ssh для ecss2 (/etc/ssh/sshd_config)
<pre># This is the sshd server system-wide configuration file. See # sshd_config(5) for more information. # This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin # The strategy used for options in the default sshd_config shipped with # OpenSSH is to specify options with their default value where # possible, but leave them commented. Uncommented options override the # default value. Port 2000 #AddressFamily any ListenAddress 10.0.10.11 #ListenAddress :: <...></pre>	<pre># This is the sshd server system-wide configuration file. See # sshd_config(5) for more information. # This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin # The strategy used for options in the default sshd_config shipped with # OpenSSH is to specify options with their default value where # possible, but leave them commented. Uncommented options override the # default value. Port 2000 #AddressFamily any ListenAddress 10.0.10.12 #ListenAddress :: <...></pre>

Перезапустим ssh:

```
systemctl restart ssh.service
```

Настройка кластера пакетов ecss-node

Запускаем необходимые сервисы:

⚠ ВАЖНО
Перед началом работы [проверьте наличие Token в системе.](#)

Запускаем пакеты *ecss-mycelium* и *ecss-ds* на первом хосте:

```
sudo systemctl start ecss-mycelium
sudo systemctl start ecss-ds
```

Переходим в CLI:

```
ssh admin@localhost -p 8023
password: password
```

Проверяем состояние системы:

```
admin@ds1@ecss1:/$ system-status
Checking...
```

	Node	Release	Erlang nodes	Mnesia nodes	Uptime
	ds1@ecss1	ecss-ds-3.14.10.222	ds1@ecss1	ds1@ecss1	0h 9m
	mycelium1@ecss1	ecss-mycelium-3.14.10.222	mycelium1@ecss1	not running	0h 9m

Далее необходимо загрузить паспорт и лицензии в систему:

```
cluster/storage/ds1/licence/set-passport <ssw passport>
ok
cluster/storage/ds1/licence/add <ssw licence>
ok
```

Выходим и запускаем остальные сервисы на первом и втором хосте.

ecss1:

```
sudo systemctl start ecss-core ecss-pa-sip ecss-mediator ecss-media-server ecss-restfs
ecss-web-conf
```

ecss2:

```
sudo systemctl start ecss-mycelium ecss-ds ecss-core ecss-pa-sip ecss-mediator ecss-
media-server ecss-restfs ecss-web-conf
```

Возвращаемся в CLI:

```
ssh admin@localhost -p 8023
password: password
```

После устанавливается связь MSR- и Core-подсистем. Для этого прописывается команда:

```
admin@[mycelium1@ecss1]:/$ /system/media/resource/declare core1@ecss1 iface msr.ecss1
bond1_ecss1_pa default local true
admin@[mycelium1@ecss1]:/$ /system/media/resource/declare core1@ecss2 iface msr.ecss2
bond1_ecss2_pa default local true
```


После того как все сервисы будут запущены, ноды какое-то время будут устанавливать связь. Как только все ноды загрузятся в system-status, последует вывод следующей информации:

```
admin@mycelium1@ecss1:/$ system-status
Checking...
```

Mnesia nodes		Release	Erlang nodes	
Node	Uptime			
core1@ecss1	ecss-core-3.14.10.222	core1@ecss1, core1@ecss2	not	
running	34m 28s			
core1@ecss2	ecss-core-3.14.10.222	core1@ecss1, core1@ecss2	not	
running	6m			
ds1@ecss1	ecss-ds-3.14.10.222	ds1@ecss1, ds1@ecss2		
ds1@ecss1, ds1@ecss2	34m 29s			
ds1@ecss2	ecss-ds-3.14.10.222	ds1@ecss1, ds1@ecss2		
ds1@ecss1, ds1@ecss2	6m			
md1@ecss1	ecss-mediator-3.14.10.222	md1@ecss1, md1@ecss2		
md1@ecss1, md1@ecss2	33m 54s			
md1@ecss2	ecss-mediator-3.14.10.222	md1@ecss1, md1@ecss2		
md1@ecss1, md1@ecss2	6m			
mycelium1@ecss1	ecss-mycelium-3.14.10.222	mycelium1@ecss1, mycelium1@ecss2	not	
running	34m 49s			
mycelium1@ecss2	ecss-mycelium-3.14.10.222	mycelium1@ecss1, mycelium1@ecss2	not	
running	6m			
sip1@ecss1	ecss-pa-sip-3.14.10.222	sip1@ecss1, sip1@ecss2		
sip1@ecss1, sip1@ecss2	33m 54s			
sip1@ecss2	ecss-pa-sip-3.14.10.222	sip1@ecss1, sip1@ecss2		
sip1@ecss1, sip1@ecss2	6m			

All services are started.

Active media resource selected list specific:

Node	MSR	MSR version	Cc-status	Cc-uptime
core1@ecss1	msr.ecss1	3.14.10.67	connected	00:32:03
	msr.ecss2	3.14.10.67	connected	00:23:56
core1@ecss2	msr.ecss1	3.14.10.67	connected	00:02:39
	msr.ecss2	3.14.10.67	connected	00:02:38

Видно, что ноды вошли в кластер и MSR зарегистрировались на ноде ecss-core.

Настройка группы IP-адресов (IP-set)

Производим настройку адаптера SIP согласно ТЗ:

```

/cluster/adapter/sip1/sip/network/set ip_set test_set node-ip node = sip1@ecss1 ip =
10.0.20.31
Property "ip_set" successfully changed from:
to
test_set: no ports set
test_set: sip1@ecss1 10.0.20.31
test_set: dscp 0.

cluster/adapter/sip1/sip/network/set ip_set test_set node-ip node = sip1@ecss2 ip =
10.0.20.32
Property "ip_set" successfully changed from:
to
test_set: no ports set
test_set: sip1@ecss1 10.0.20.31
test_set: sip1@ecss2 10.0.20.32
test_set: dscp 0.

/cluster/adapter/sip1/sip/network/set ip_set test_set listen-ports list = [5062]
Property "ip_set" successfully changed from:
ipset1:
ipset1: sip1@ecss1 10.0.20.31
ipset1: sip1@ecss2 10.0.20.32
ipset1: dscp 0
to
ipset1: 5062
ipset1: sip1@ecss1 10.0.20.31
ipset1: sip1@ecss2 10.0.20.32
ipset1: dscp 0

```

Далее создаем домен и присваиваем ему созданную группу (IP-set) настроек SIP-адаптера:

```

domain/declare test_domain --add-domain-admin-privileges --add-domain-user-privileges
New domain test_domain is declared

domain/test_domain/sip/network/set ip_set [test_set]

Property "ip_set" successfully changed from:

[] to ["test_set"].

```

После создания домена настраиваем:

- [маршрутизацию](#);
- [пользователей](#);
- [абонентов](#);
- [транки](#).

11.3 Пример первичной конфигурации системы с помощью web-конфигуратора

11.3.1 Исходные данные

- Установка системы завершена;
- Система готова к дальнейшей настройке;
- Интерфейсы запущены.

Рекомендуется использовать последние доступные версии браузеров. Рекомендуемые к использованию браузеры: Opera, Chrome.

Для начала конфигурации системы необходимо зайти в web-конфигуратор.

Для определения и регистрации в системе планируются:

- Абоненты с номерами 101, 102 , 103, 104 ,105, 106, 107, 108, 109, 110;
- Транк в сторону шлюза.

11.3.2 Подготовка к работе

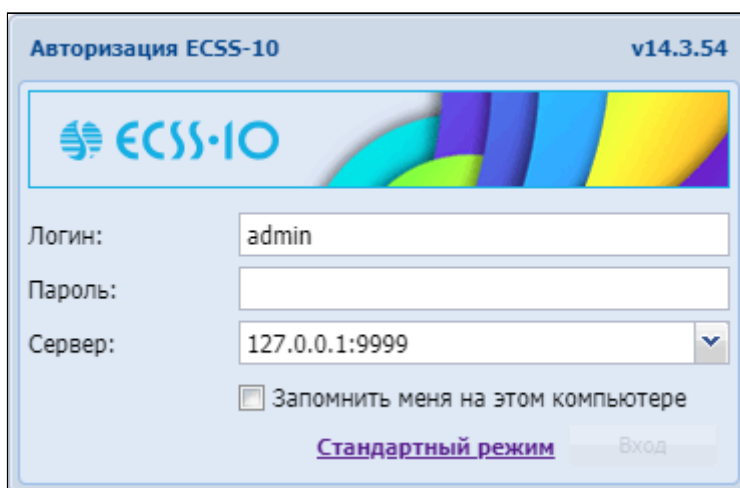


Рисунок 3 – Вход в web-конфигуратор (окно авторизации)

В окне авторизации необходимо ввести значения, определенные при установке web-конфигуратора.

- ⚠** Стандартные значения для авторизации:
Логин: admin
Пароль: password

После входа в систему будет видна основная рабочая область с иконками приложений, а также строка состояния с доступными опциями, в частности:

- 1 – выход из системы;
- 2 – выбор домена;
- 3 – выбор языка.



Рисунок 4 – Вид рабочей области web-конфигуратора

11.3.3 Создание учетной записи оператора

После авторизации для повышения безопасности при эксплуатации программного коммутатора рекомендуется создание учетных записей для операторов, а также сменить пароль для пользователя **admin**.

Для создания новой учетной записи оператора используется приложение "Управление пользователями" ("User manager"):

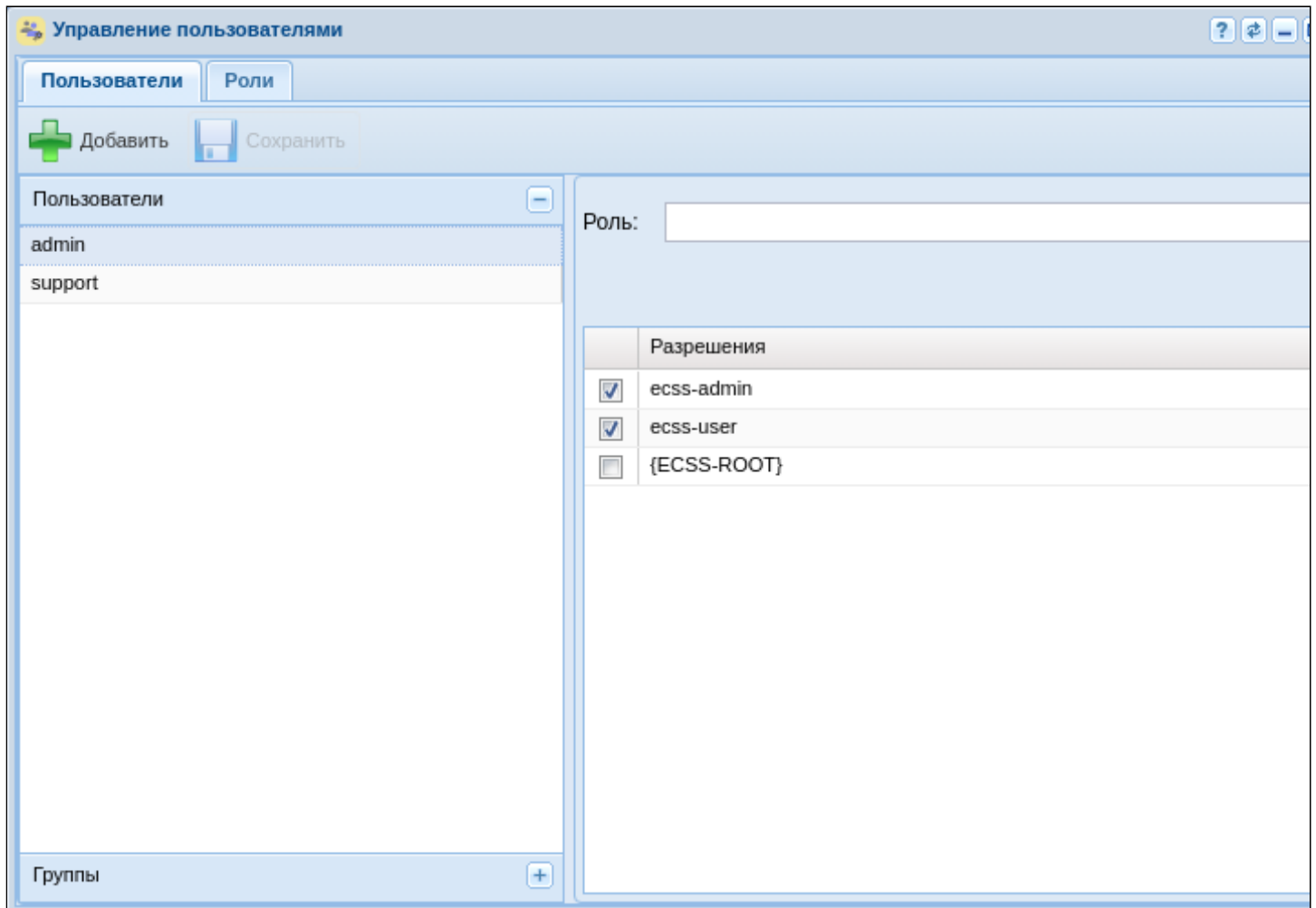



Рисунок 5 – Вид приложения "Управление пользователями"

Нажмите кнопку "Добавить" . В открывшемся окне требуется определить новую учетную запись, для этого:

1. В поле "Имя" введите логин учетной записи, например "test";
2. В поле "Пароль" и "Подтверждение" введите пароль для пользователя, например "testpassword";
3. Определите уровень прав доступа для пользователя через выбор текущих разрешений или с помощью ролей, например **ecss-user**. Описание можно посмотреть в разделе "[Управление пользователями](#)".

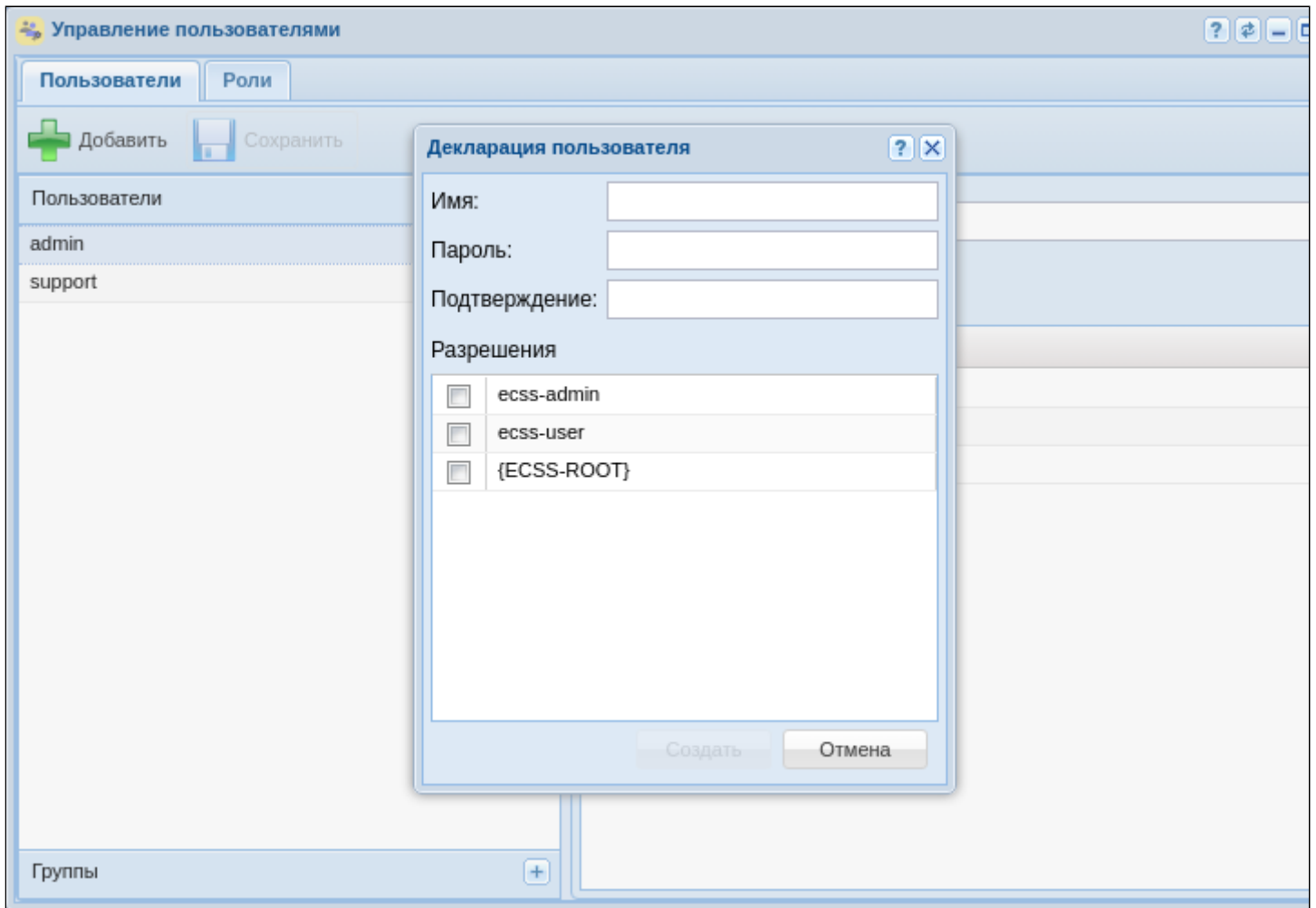


Рисунок 6 – Диалоговое окно создания учетной записи оператора

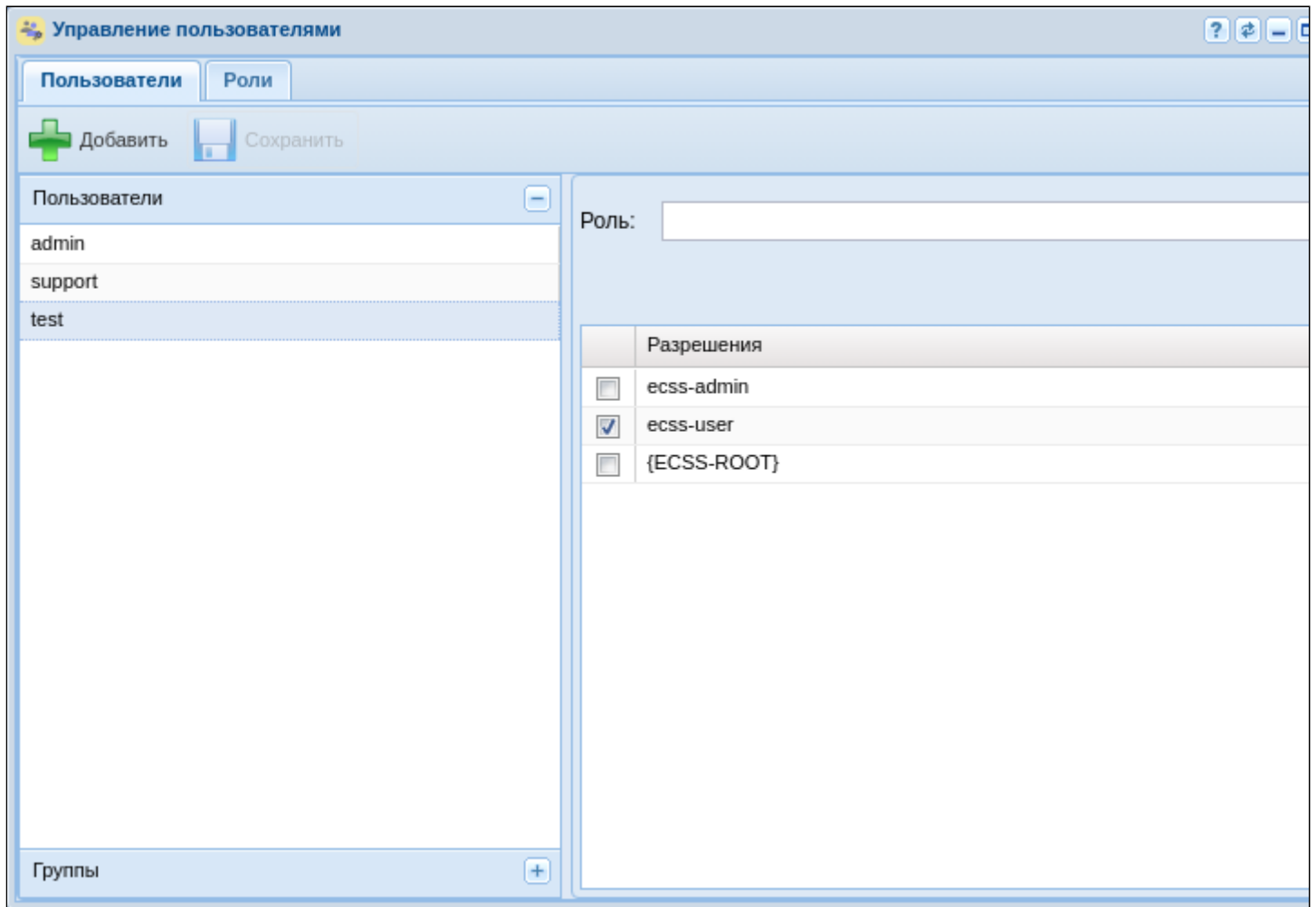


Рисунок 7 – Вид приложения с созданной учетной записью оператора

Для смены пароля рядом с названием пользователя нажмите на кнопку редактировать. В появившемся диалоговом окне введите:

1. Старый пароль (для пользователя "**admin**" по умолчанию паролем является "**password**");
2. Новый пароль;
3. Подтвердите новый пароль.

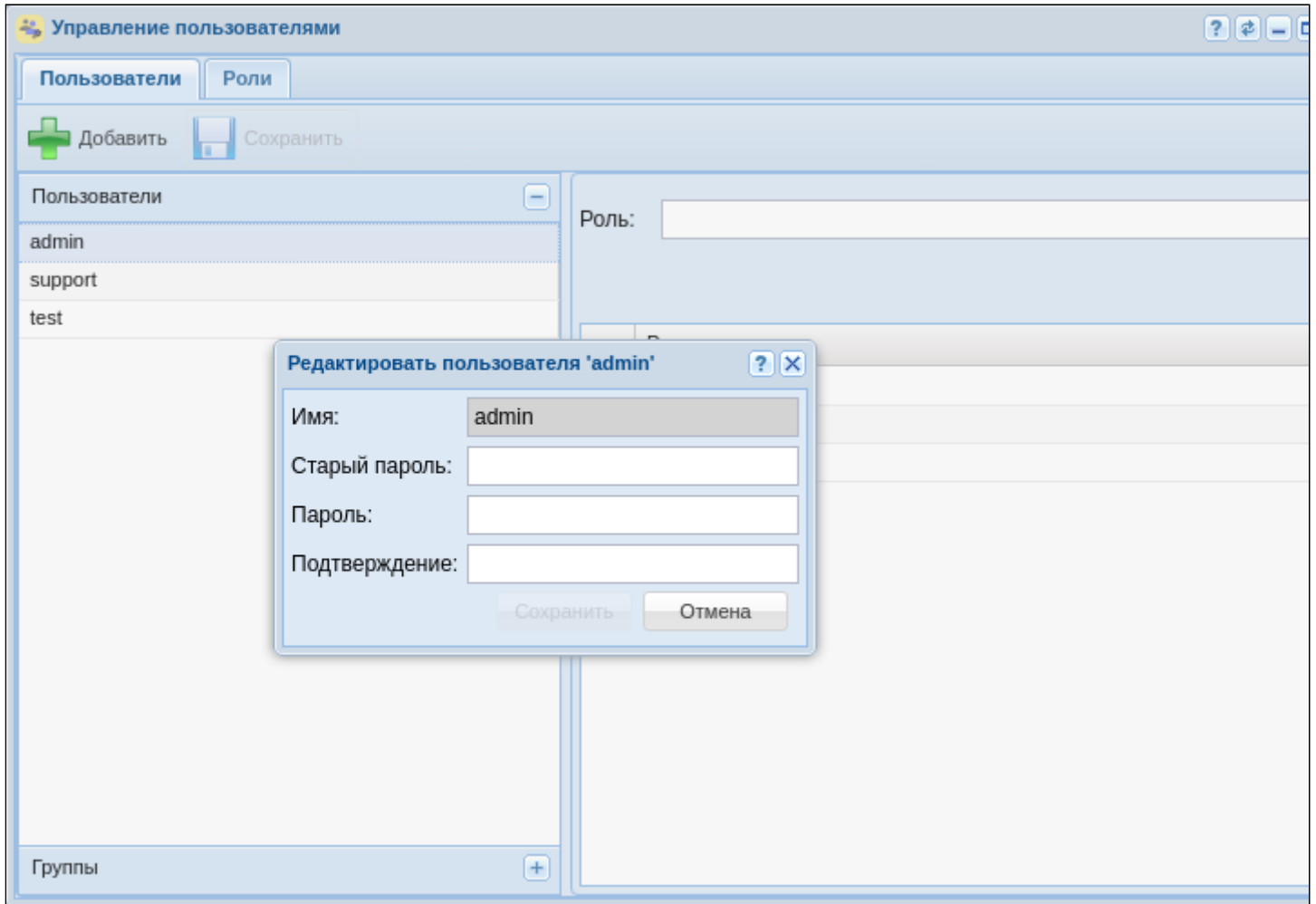


Рисунок 8 – Диалоговое окно редактирования учетной записи

11.3.4 Создание домена

Для создания домена необходимо войти в приложение "*Домены*" ("*Domain*"). В открывшемся окне требуется создать домен, для этого:

1. Нажмите кнопку "*Добавить домен*":

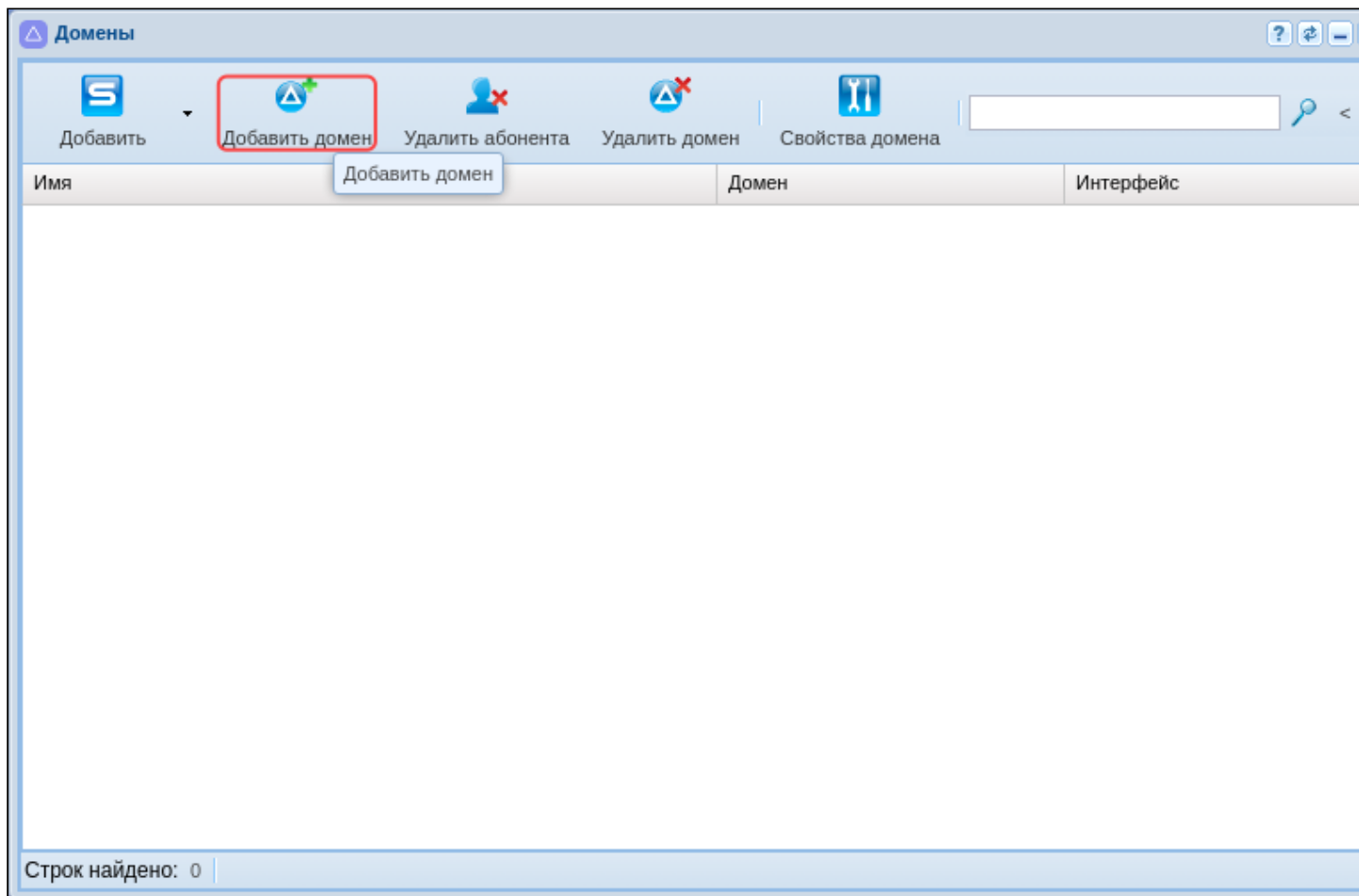


Рисунок 9 – Добавление домена в систему

2. В появившемся диалоговом окне доступны следующие настройки:

- *Имя (Name)* – индивидуальное имя виртуальной АТС;
- *Профиль услуг (SS profile)* – системный профиль дополнительных услуг. Этот профиль будет **скопирован** с тем же именем во вновь создаваемый домен и для всех услуг из этого профиля будет автоматически разрешён доступ через **access-list**;
- *IVR профиль (IVR profile)* – профиль IVR, заданный в приложении "Редактор IVR ограничений".

Введите название домена, например "test_domain";

3. Нажмите кнопку "Ок":

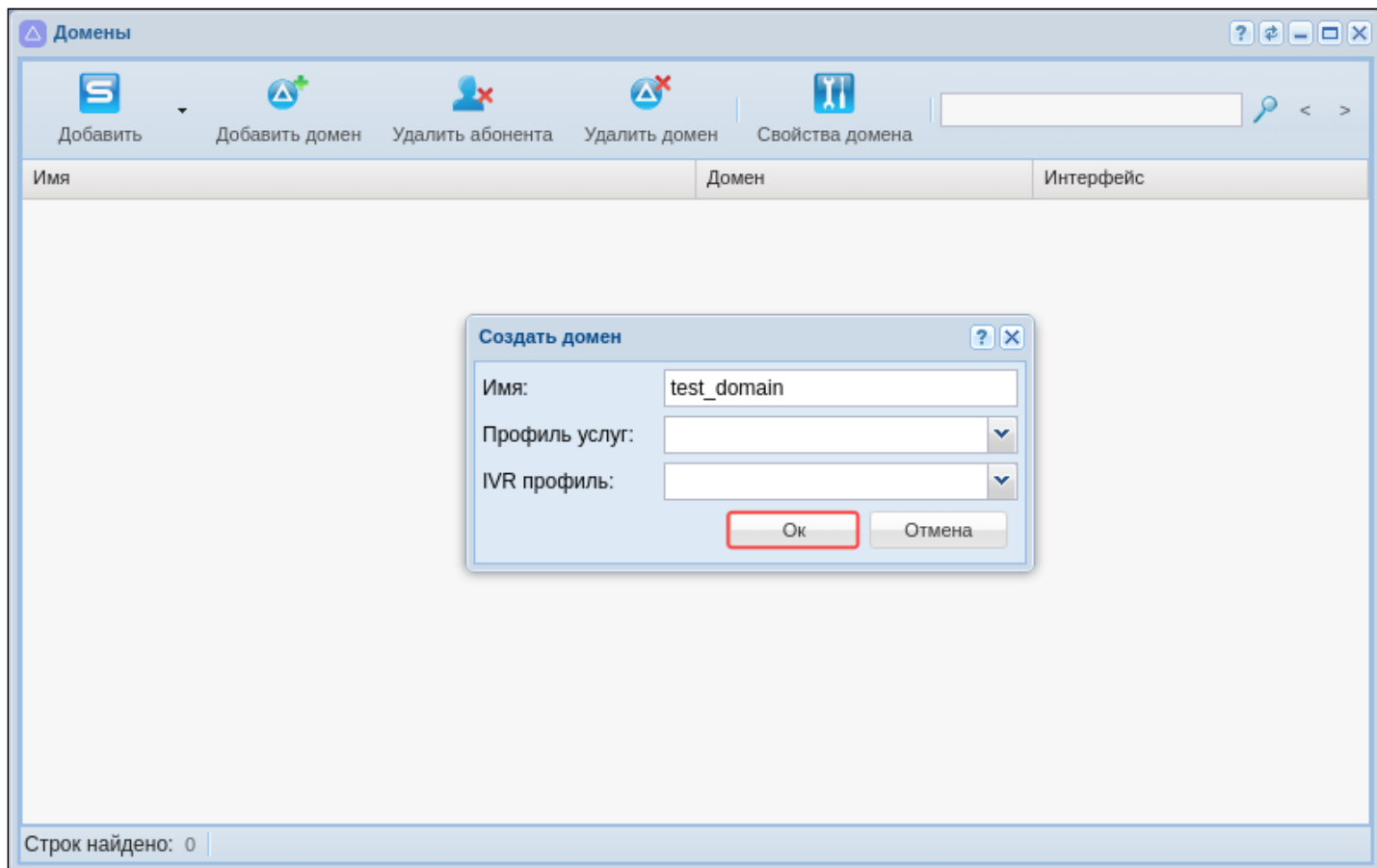


Рисунок 10 – Определение настроек нового домена

4. Нажмите кнопку "Обновить"  в приложении.

Созданный домен будет отображаться в текущей конфигурации:

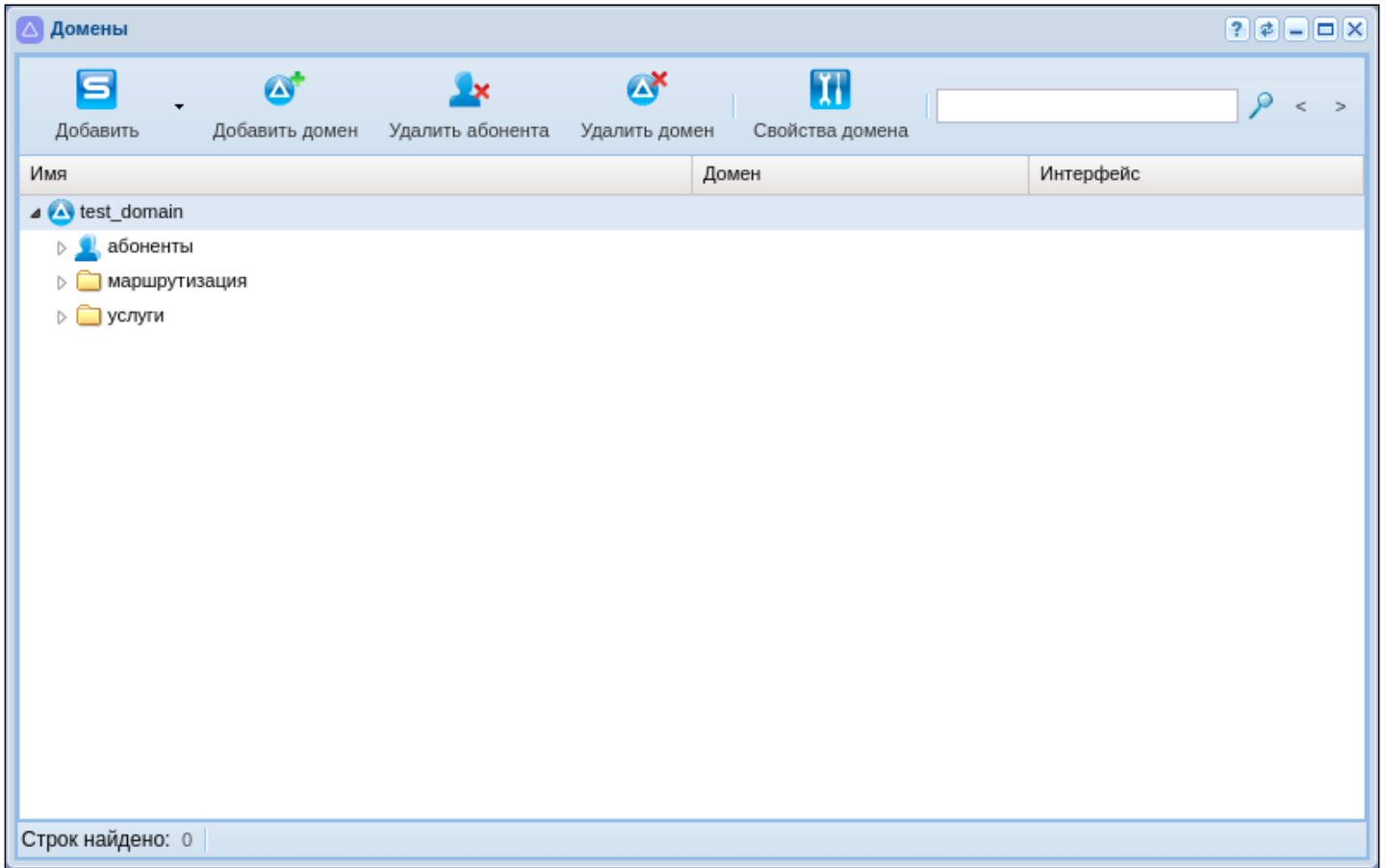
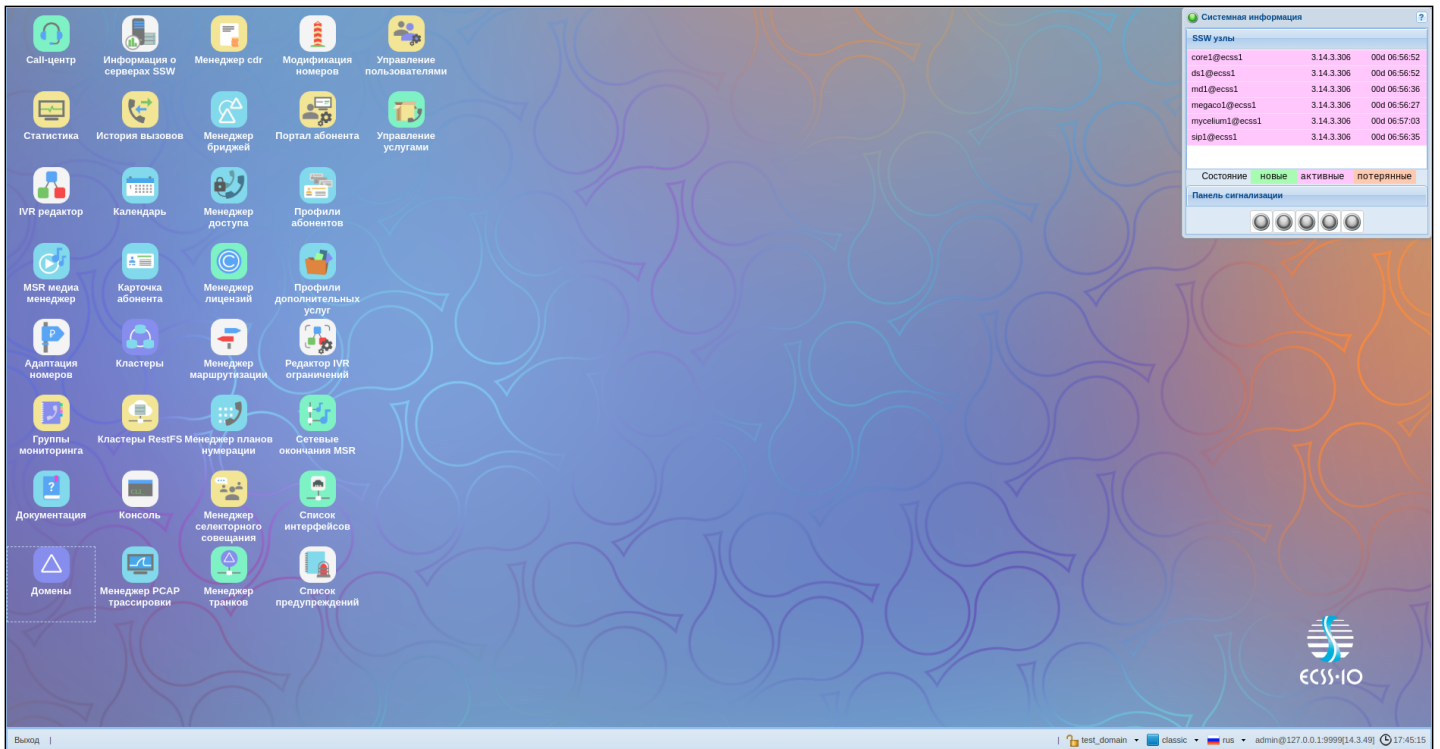


Рисунок 11 – Отображение созданного домена

Для редактирования текущего домена его необходимо выбрать в системе. Для перехода в домен воспользуйтесь опцией выбора домена (см. пункт 2 на рисунке "Вид рабочей области web-конфигуратора").

После выбора домена, согласно текущей конфигурации системы, станут доступны все приложения:



11.3.5 Создание IP-set (sip-транспорта) и присвоение его к домену

Для настройки интерфейса необходимо открыть приложение "Кластеры ("Clusters").

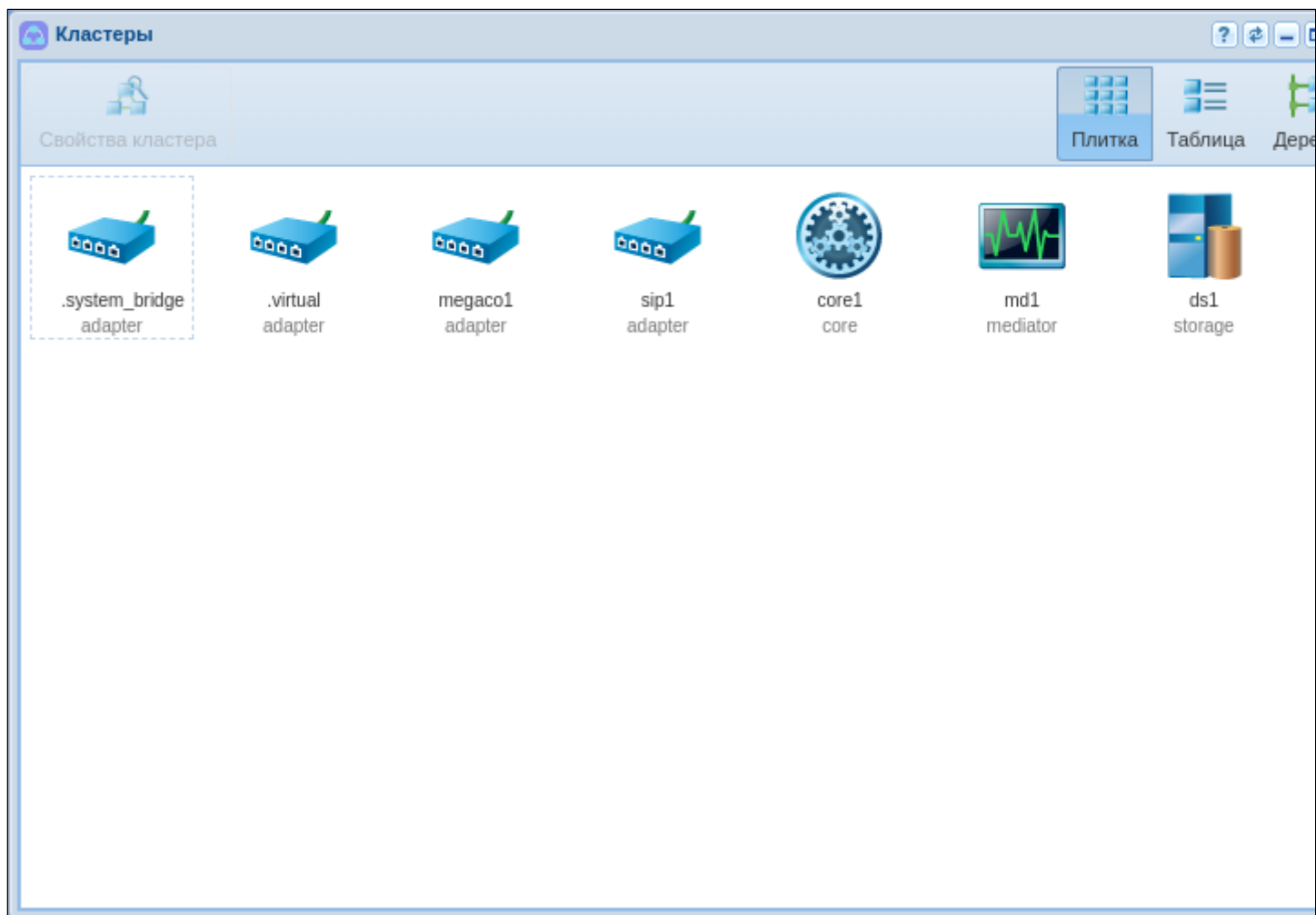


Рисунок 13 – Вид приложения "Кластеры"

IP-set – совокупность одного или нескольких IP-адресов из одной подсети и списка слушающих UDP/TCP-портов, которые будут открыты SIP-адаптером на данных адресах.

Для создания новой группы IP-адресов (IP-set) выделите кластер SIP-адаптера "sip1" и нажмите на кнопку "Свойства кластера" (или нажмите на иконку кластера двойным щелчком левой кнопки мыши).

В появившемся диалоговом окне перейдите на вкладку "Транспорт". Далее нажмите кнопку "Добавить". Появится новая группа, для редактирования полей двойным щелчком кнопки мыши нажмите на нужное:

1. Переименуйте группу адресов (IP-set), например "test_set";
2. Задайте порт, по которому будет происходить обращение к домену, например 5062;
3. Разверните вновь созданную группу нажатием на треугольник слева от названия группы;
4. Определите адрес для ноды SIP-адаптера, согласно примеру конфигурации. Для системы без резервирования укажите 10.0.20.10:

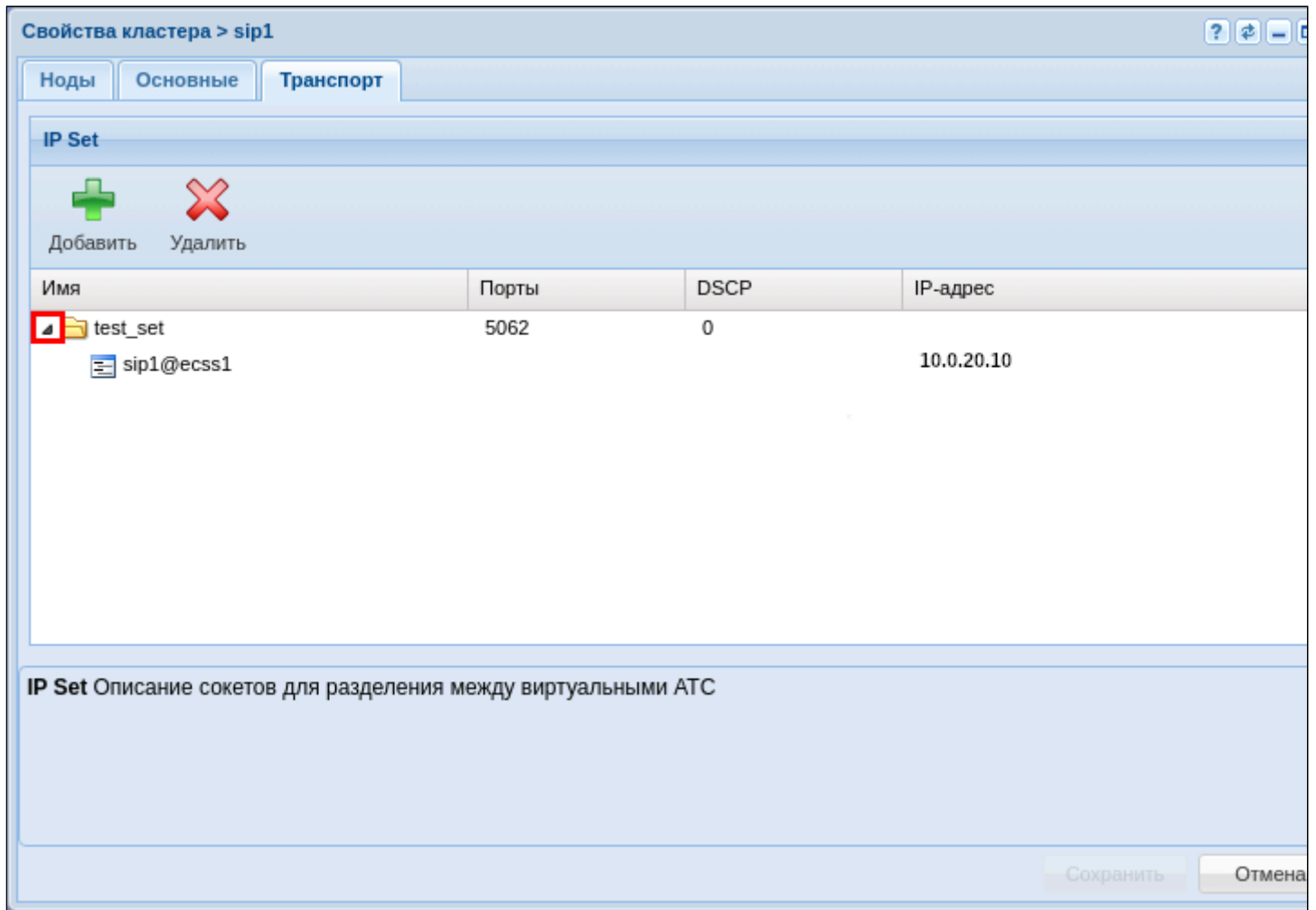


Рисунок 14 – Вид приложения "Кластеры/Транспорт"

Для системы с резервированием укажите 10.0.20.31 и 10.0.20.32:

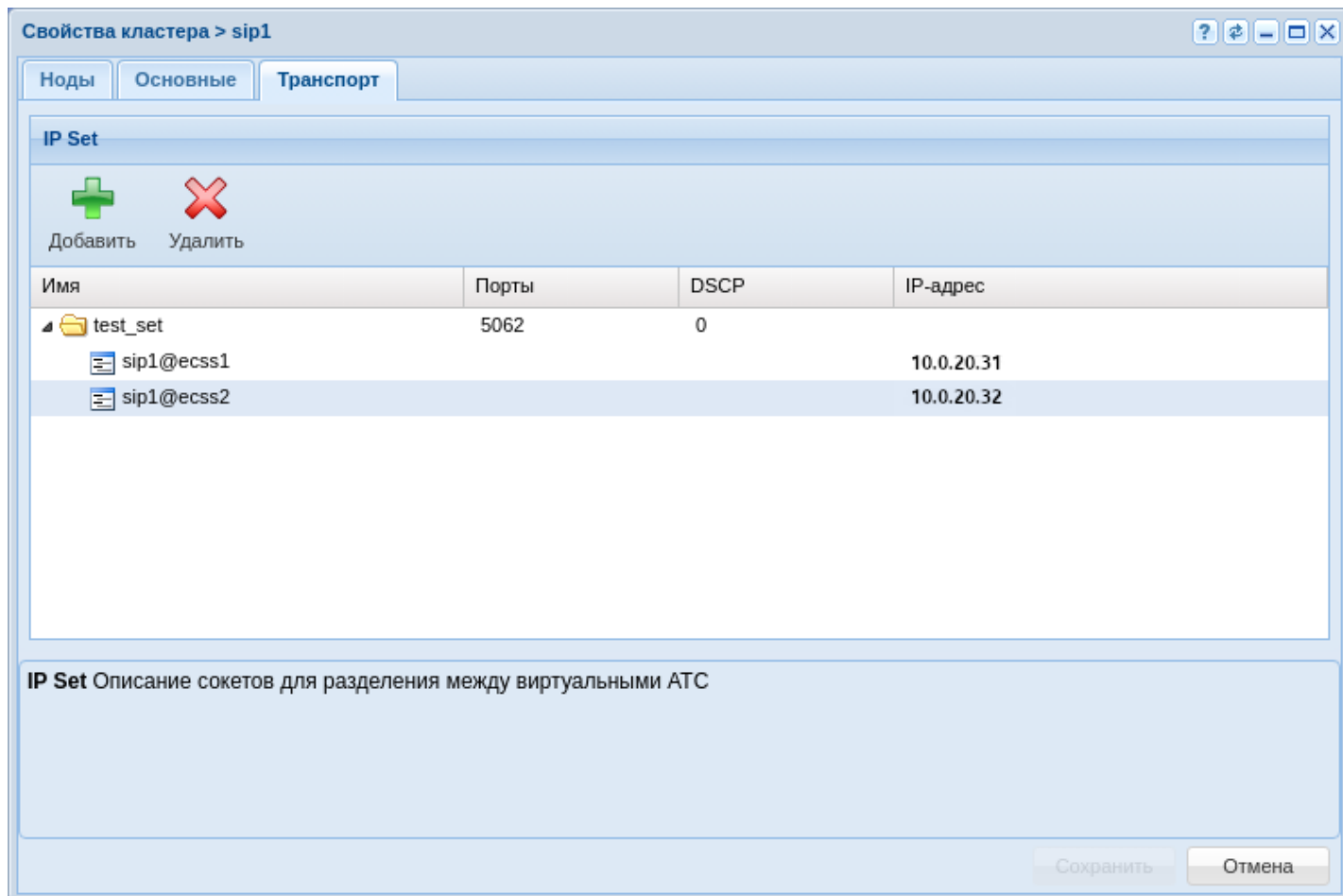


Рисунок 15 – Вид приложения "Кластеры/Транспорт" для кластера

Нажмите кнопку "Сохранить" для применения настроек.

Для того чтобы привязать группу адресов к домену, необходимо вернуться в приложение "Домены" ("Domain"), выделить домен и перейти в настройки нажатием на кнопку "Свойства домена" или двойным нажатием левой кнопки мыши на домен.

В настройках списка откройте ветку "SIP", в ней "SIP транспорт", далее выберите в поле IP set созданную группу адресов. Нажмите кнопку "Сохранить" для применения настроек.

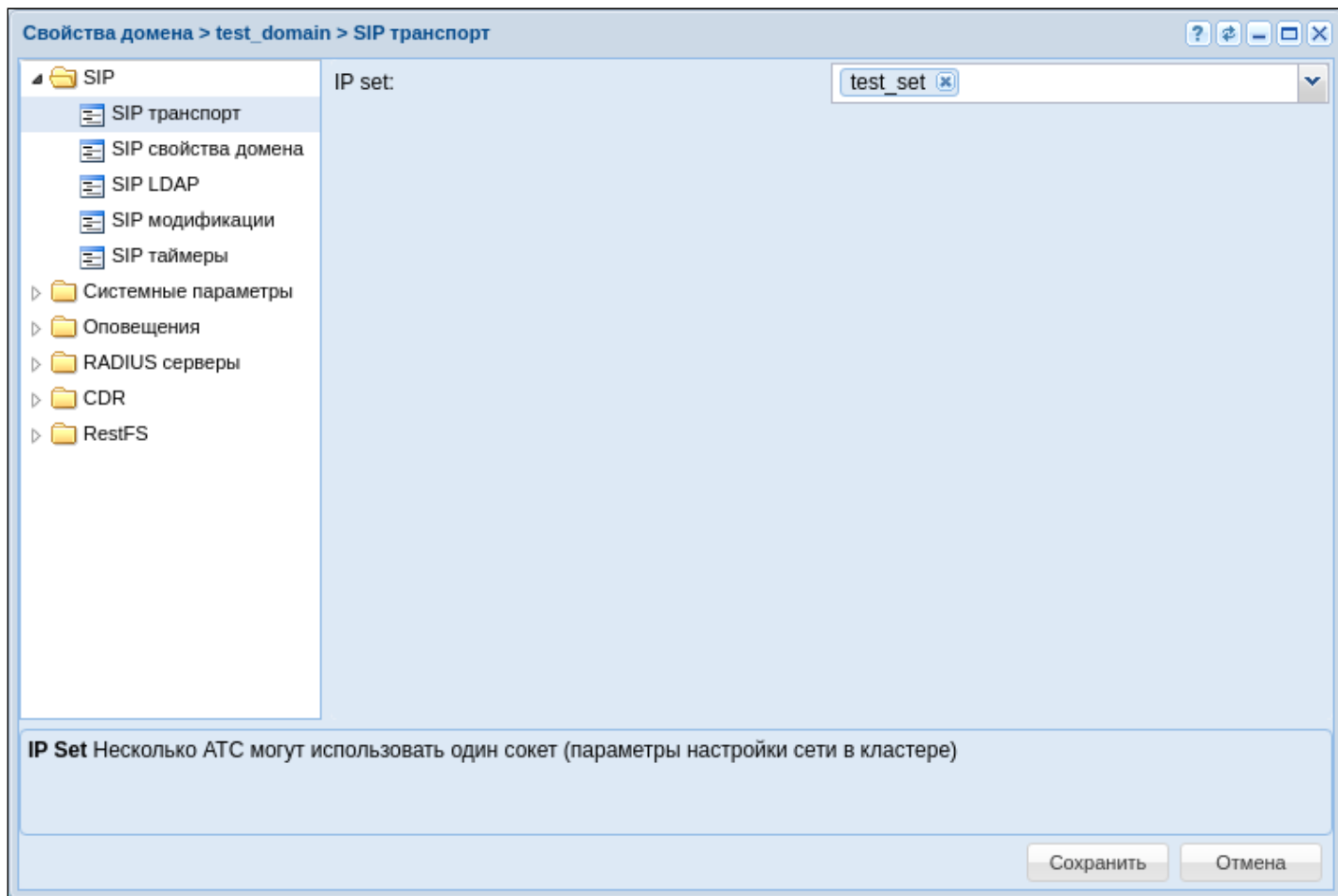


Рисунок 16 – Окно настройки для SIP транспорта

11.3.6 Создание абонентов

Для создания и редактирования параметров абонентов в системе используется приложение "[Карточка абонента](#)" ("Subscriber card").

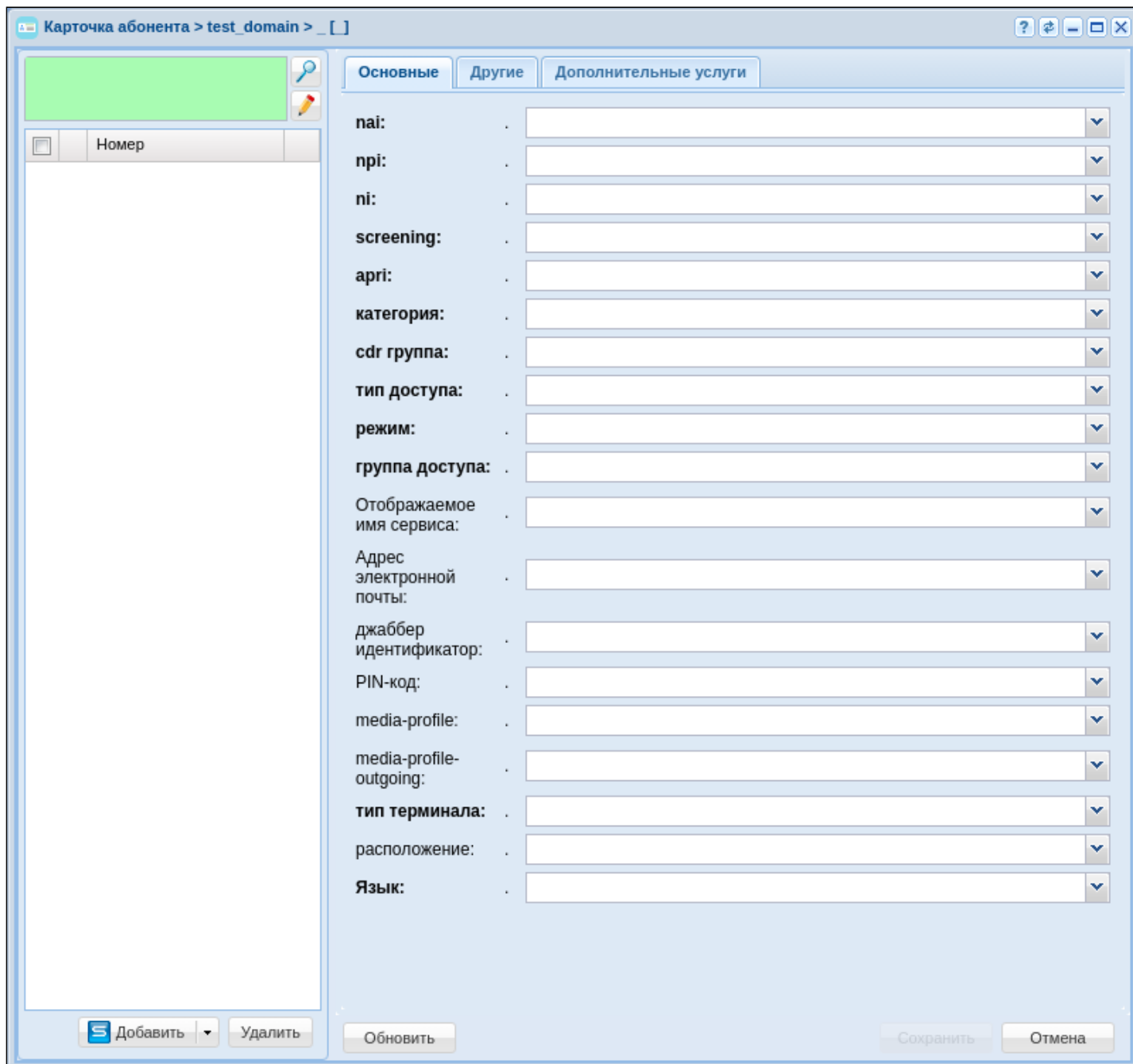


Рисунок 17 – Вид приложения "Карточка абонента"

В конфигурации возможно создание SIP-абонентов и виртуальных абонентов.

Для пользователей, имеющих физическое окончание, используется функционал SIP-абонента, в то время как виртуальный абонент используется тогда, когда нужна функциональность без физических окончаний. Например, номер для обращения к ivr-скрипту.

Для создания новых абонентов нужно нажать на кнопку "Добавить".

В появившемся диалоговом окне укажите следующие параметры:

- *Контекст* – контекст маршрутизации, выбираем тот, который создали, например "test_name";
- *Имя интерфейса* – номер или группа номеров, который присваивается абоненту, например {100-110};
- *Алиас как пользователь* – настройка, которая связывает номер сущности, алиас и пользователя одним и тем же названием, в примере настройка активирована;
- *Авторизация* – порядок проверки подлинности прав пользователя на доступ к данным, в примере используется always;

- *Логин* – использую настройку номер как логин для простоты;
- *Пароль* – можно задать или использовать сгенерированный системой.

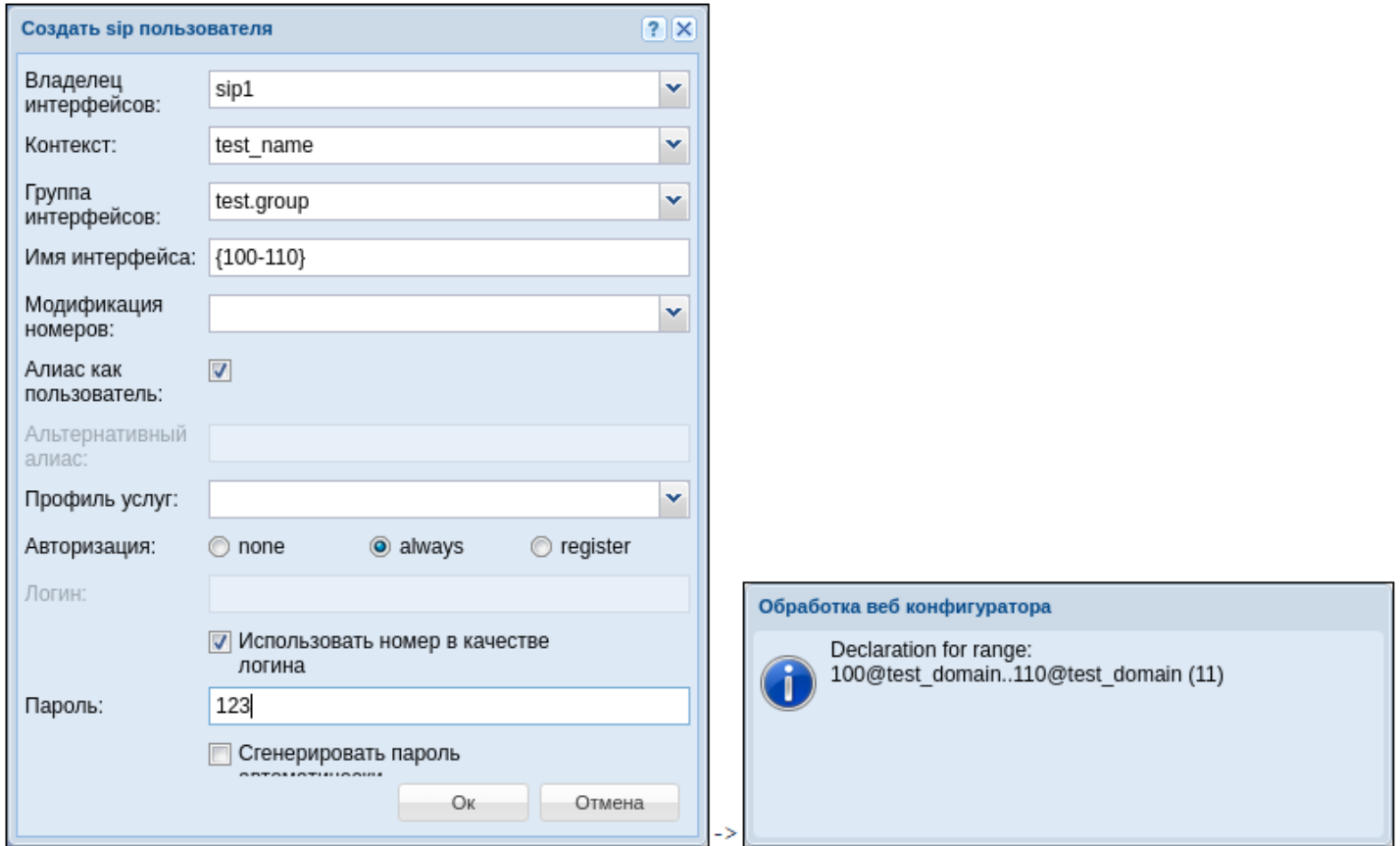


Рисунок 18 – Пример определения абонентов в домене

11.3.7 Создание и применение контекстов маршрутизации для домена

Маршрутизация отвечает за поиск номера и последующую адресацию вызова. Для корректной работы системы должен быть настроен как минимум один контекст маршрутизации.

Настройка маршрутизации осуществляется в приложении "[Менеджер маршрутизации](#)" ("Routing manager").

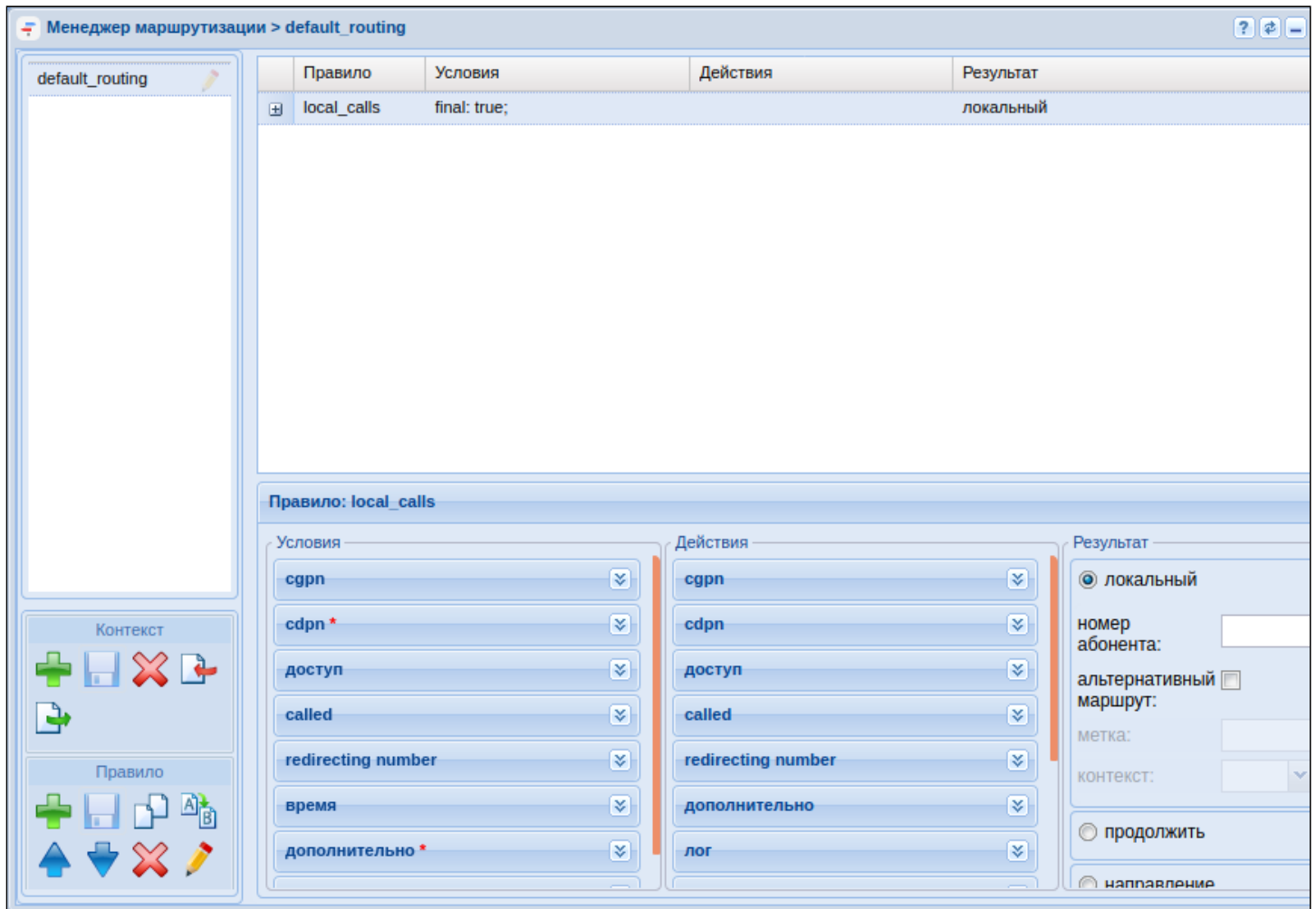

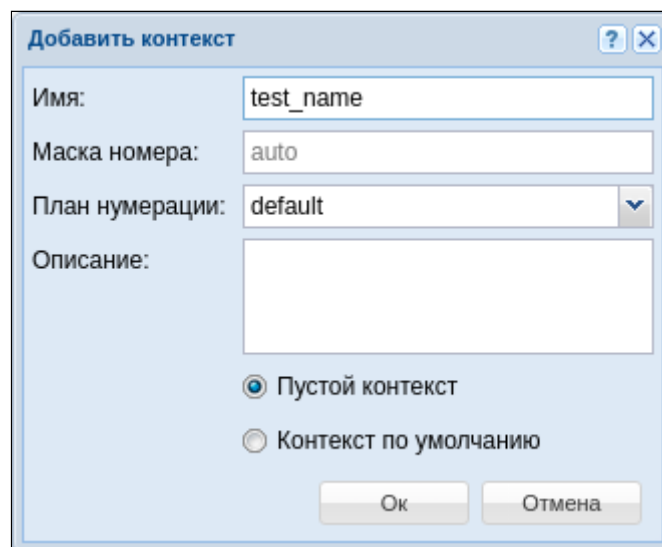


Рисунок 19 – Вид приложения "Менеджер маршрутизации"

В качестве примера создадим контекст, а затем несколько правил в нём, для этого:

1. В левой части окна в разделе "Контекст" нажмите на кнопку "Создать контекст"  ;
2. В открывшемся диалоговом окне обозначьте имя контекста, а также тип контекста – пустой контекст:



3. Нажмите "Сохранить контекст"  ;

Создадим 4 правила в данном контексте:

- rule1 – правило для выхода на транк ТАУ-72;
- rule2 – правило локальной маршрутизации для номеров 101-105;
- rule3 – правило для выхода в ivr;
- rule4 – правило-исключение.

Для создания нового правила нужно выделить созданный контекст и нажать кнопку "Создать правило". В появившемся окне нужно ввести название правила. Затем сохранить вновь созданные правила.

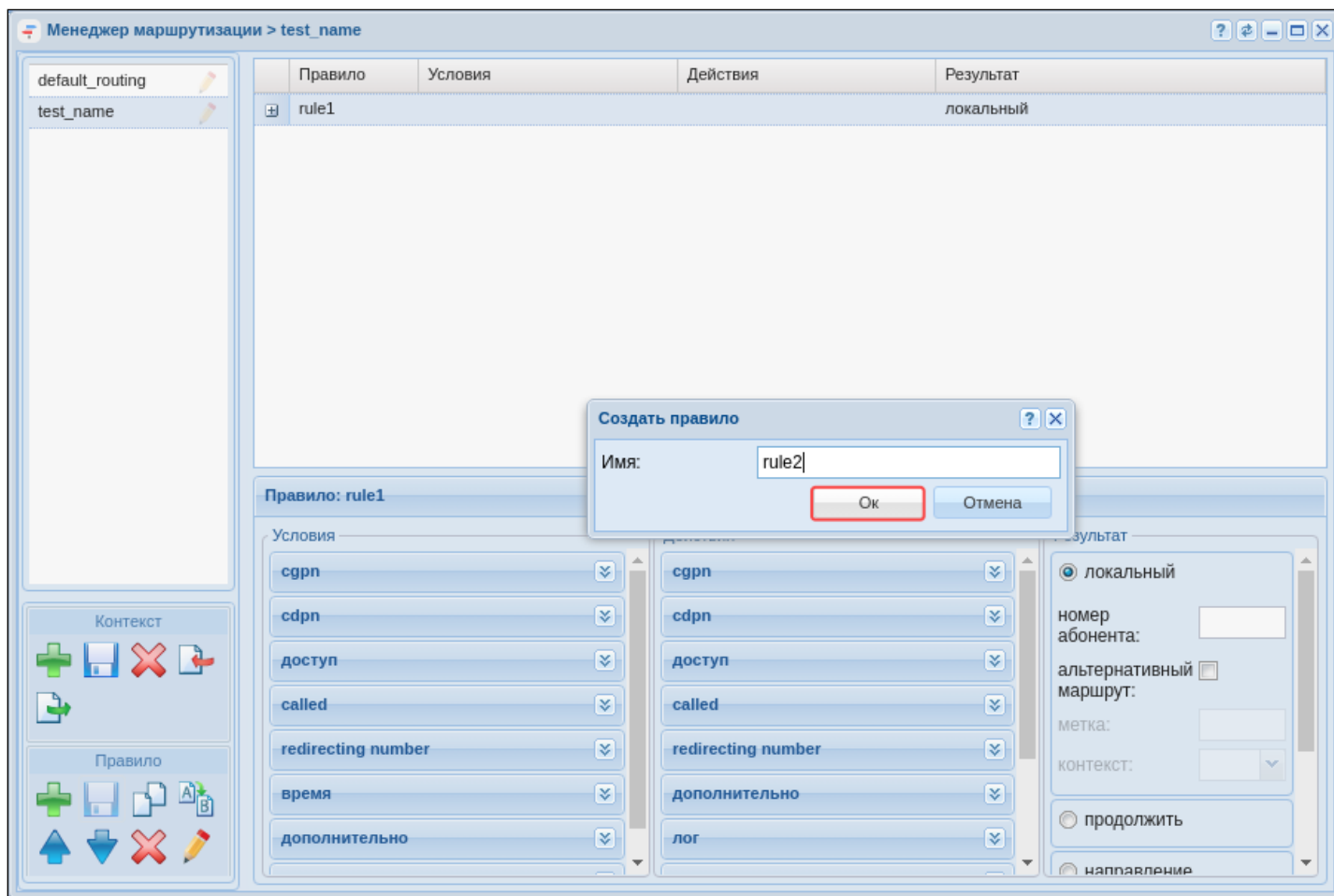


Рисунок 20 – Создание правила контекста маршрутизации

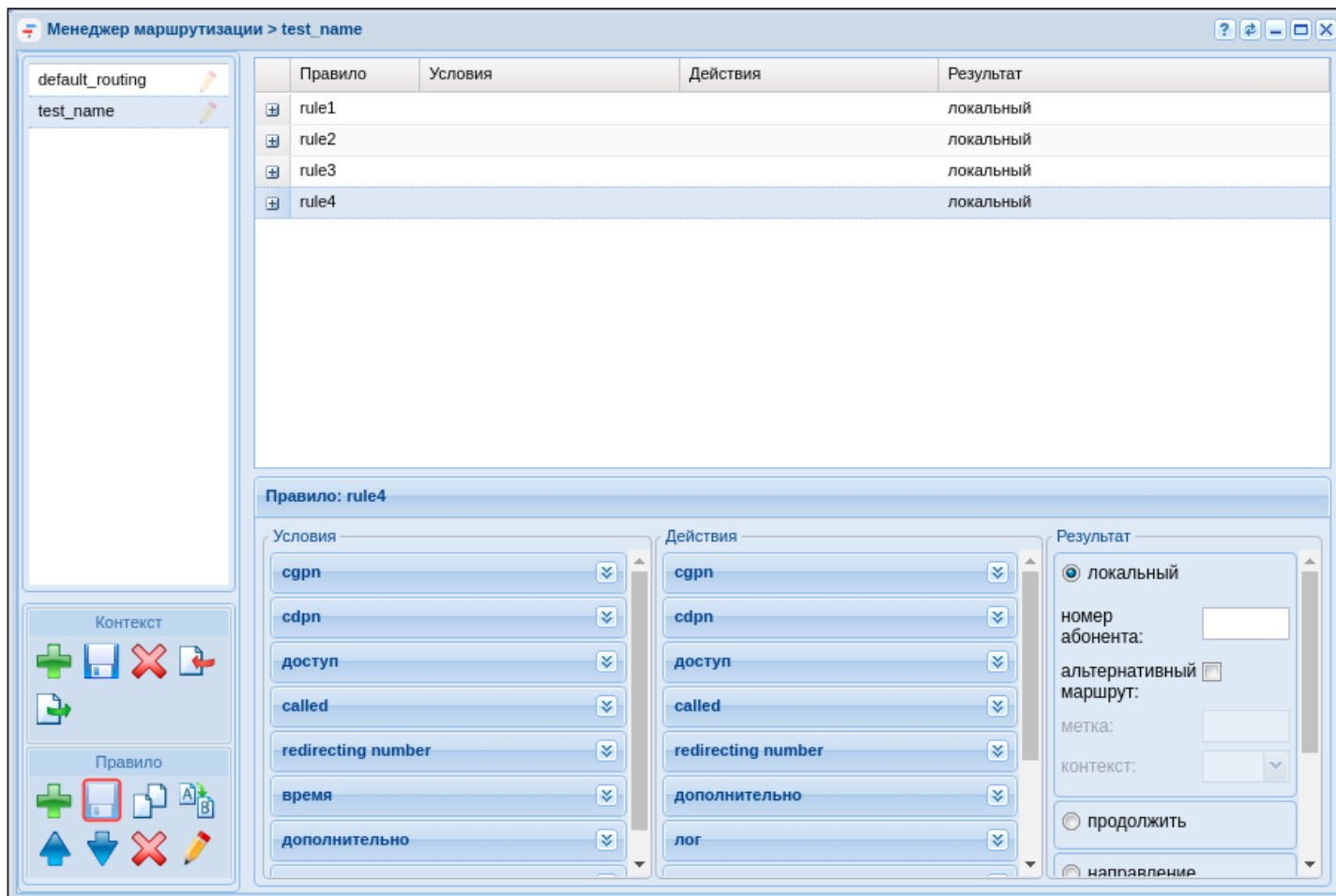


Рисунок 21 – Определение правил

На данный момент не определен транк, чтобы можно было сослаться на него в правиле, однако можно задать номера, по которым будет производиться выборка.

Нажав на правило rule1, перейдем в нижнюю часть экрана, в которой располагаются области для редактирования контекста маршрутизации. Условно определим, что выборка для выхода на транк будет проводиться по признакам номера вызываемого абонента (CDPN), а номера в транке должны начинаться с цифры 4.

Функционально контекст маршрутизации делится на три части:

- Условия ("Condition") – раздел, определяющий выражения для выборки по предложенным признакам;
- Действия ("Action") – раздел, выполняющий преобразование признаков номеров к конкретному значению;
- Результат ("Result") – раздел, завершающий маршрутизацию и определяющий её результат.

rule1: Для выхода на транк нужно правильно отредактировать каждую часть:

- В разделе условий перейти на вкладку CDPN. Ввести номера телефонов, которые находятся за транком в поле "Номер". Например, чтобы определить номера с 106 по 107, нужно ввести условие: 10(6-7);
- В разделе действий перейти на вкладку CDPN. Ввести маску для изменения номера в поле "Номер". Например, чтобы добавить число 4 перед номером, нужно заполнить поле следующим выражением: 4{1,2,3};
- Нажать кнопки "Сохранить правило" и "Сохранить контекст" для применения изменений.

Чтобы настроить поле результата, нужно определить транк в системе, поэтому к настройке данного правила вернемся чуть позже.

Таким же образом нужно настроить оставшиеся правила.

Для rule2:

- В разделе условий, во вкладке CDPN ввести номера телефонов, которые находятся за транком в поле "Номер". Например, чтобы определить номера со 101 по 105, ввести условие 10(1-5);
- В разделе результата определить результат как локальный (т.е. локальная маршрутизация);
- Нажать кнопки "Сохранить правило" и "Сохранить контекст" для применения изменений.

Для rule 3 допустим, что абоненты с номерами 108, 109 и 110 попадают в ivr-скрипт–информатор, прежде чем звонить далее.

- В разделе условий, во вкладке CGPN ввести номера телефонов, по которым будет выборка в поле "Номер", например 1(10,08-09).
- Нажать кнопки "Сохранить правило" и "Сохранить контекст" для применения изменений.

Чтобы настроить поле результата, нужно определить в системе ivr-скрипт, поэтому к настройке данного правила вернемся чуть позже.

Для rule 4 мы определяем правило-исключение – это правило, которое работает в случае, если любые другие правила не срабатывают.

- По умолчанию данное правило создается в системе в контексте default_routing. В условиях на вызываемый номер прописывается символ %, а результатом является локальная маршрутизация, однако, если мы назначаем абоненту контекст, отличный от default_routing, в конце *рекомендуется* создавать это правило.

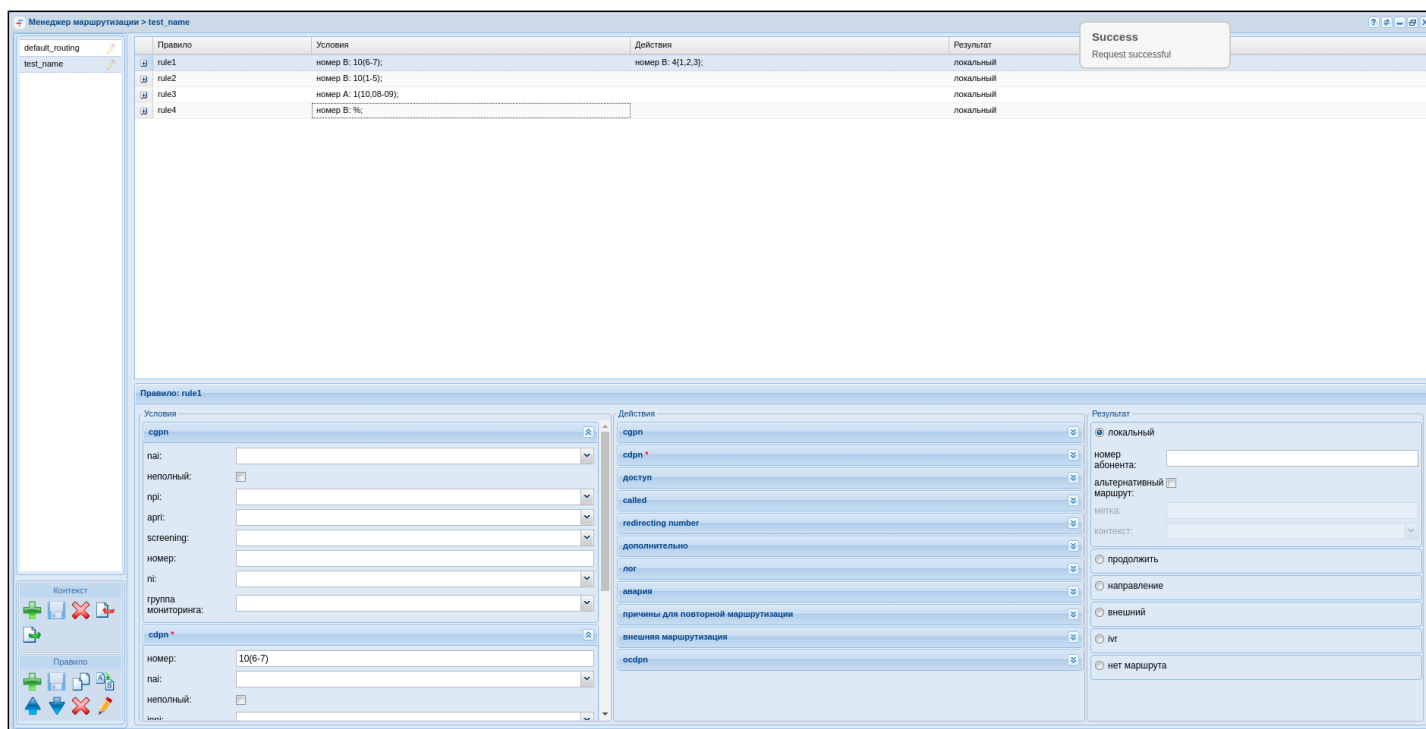


Рисунок 22 – Пример настроек контекста маршрутизации

11.3.8 Создание транка

Для создания и редактирования параметров транка в системе используется приложение "Менеджер транков" ("Trunk manager").

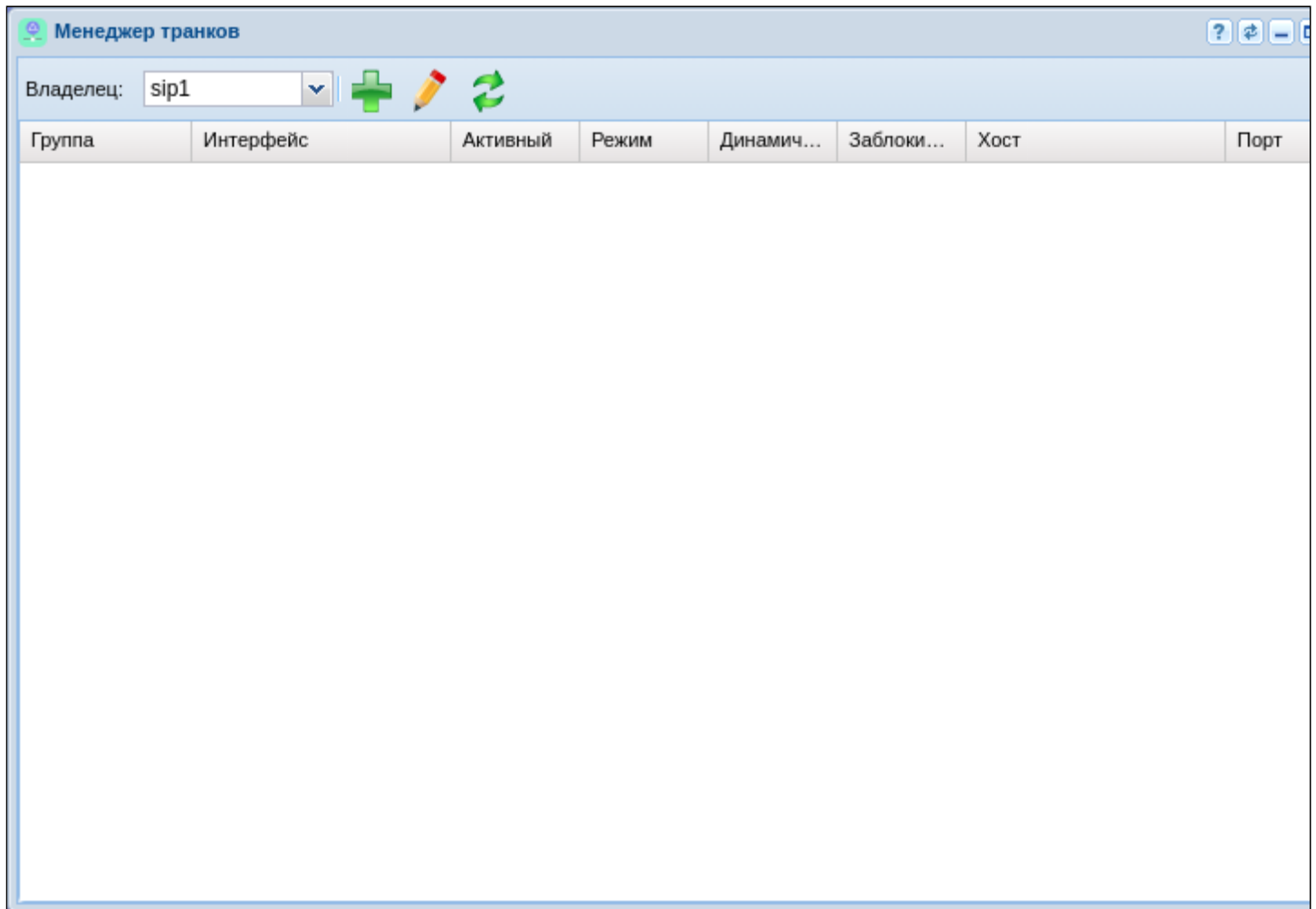



Рисунок 23 – Вид приложения "Менеджер транков"

Для определения транка в системе нужно нажать на кнопку "Создать транк"  , в появившемся диалоговом окне определить параметры:

- *Имя* – назначить название транка, по которому его можно идентифицировать в системе;
- *Контекст* – применить ранее созданный контекст маршрутизации test_name;
- *Группа* – выбрать группу интерфейсов, созданную при определении абонентов "test.group";
- *Группа IP-адресов (IP-set)* – созданная на домене группа адресов "test_set";
- *Регистрация* – если используется транковая, то нужно включить данный параметр, в примере она не используется;
- *Хост [:порт]* – IP-адрес назначения транка – 10.0.3.100;
- *Порт для приёма* – транспортный порт, на котором будет слушаться трафик с транка, соответствует порту, назначенному на группу IP-адресов.

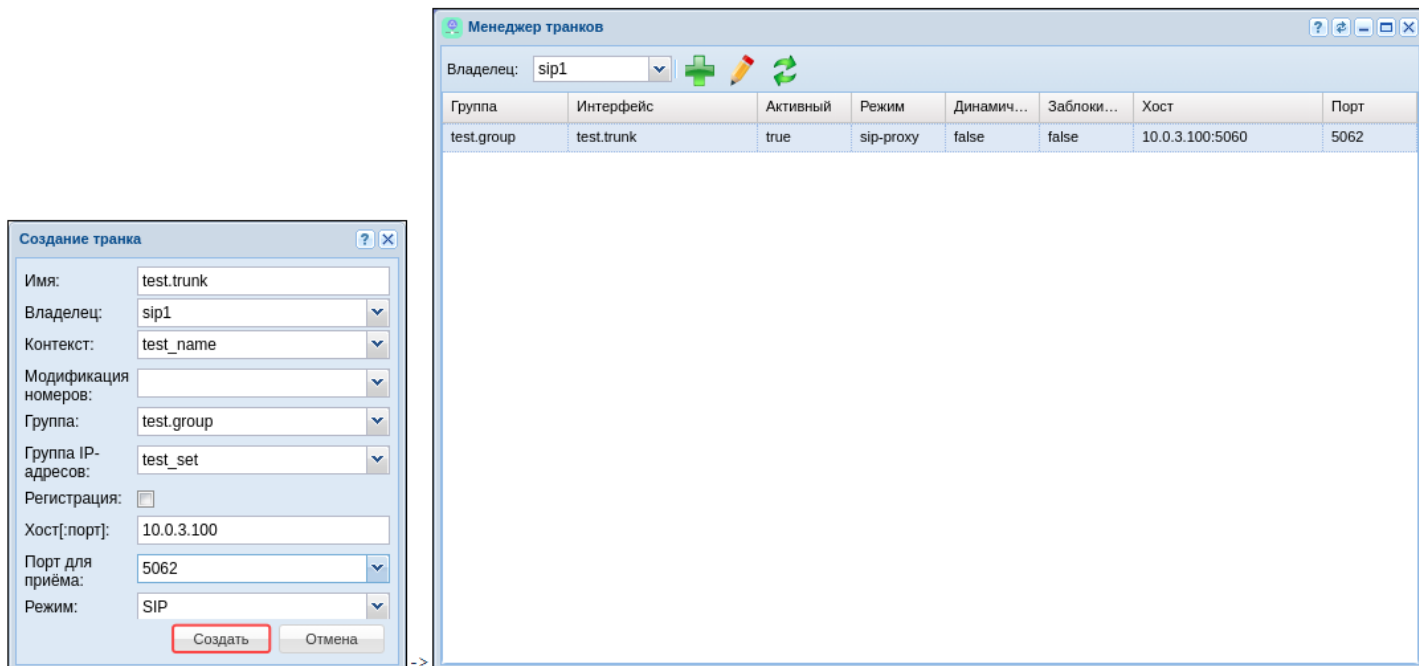


Рисунок 24 – Создание транка

11.3.9 Создание IVR-скрипта

Для создания IVR используется приложение "IVR-редактор" ("IVR editor").

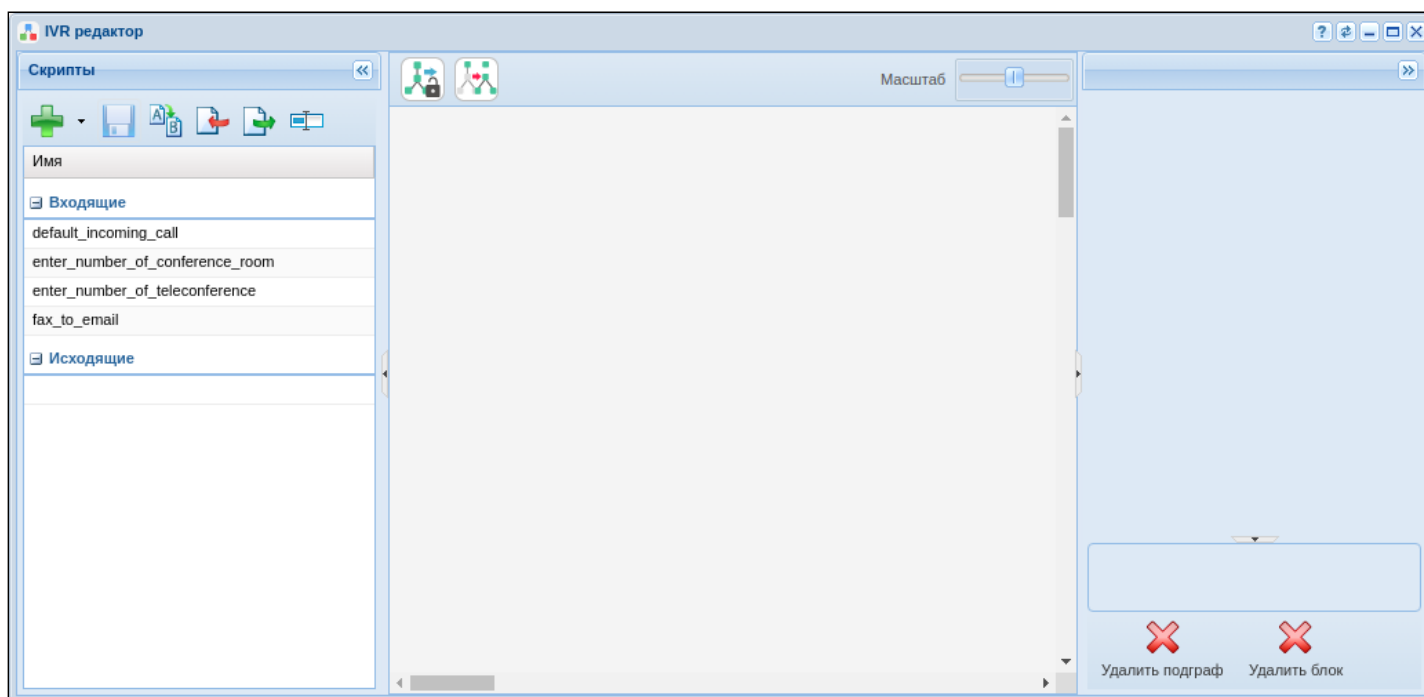



Рисунок 25 – Вид приложения "IVR-редактор"

Для создания скрипта нужно нажать на кнопку "Добавить" , выбрать тип скрипта (в данном случае будет скрипт на входящие вызовы), в диалоговом окне указать название скрипта, например "test_ivr".

После создания скрипта в основном окне редактора появится блок-схема.

В качестве примера составим скрипт, который при срабатывании проигрывает вызывающему абоненту заранее записанную фразу, а затем продолжает вызов.

Пример скрипта приведен на рисунках ниже. Здесь используются блоки Info и Dial.

- Блок Info – блок, который проигрывает сообщения до момента ответа пользователя. В качестве записанной фразы используется генератор тонов:

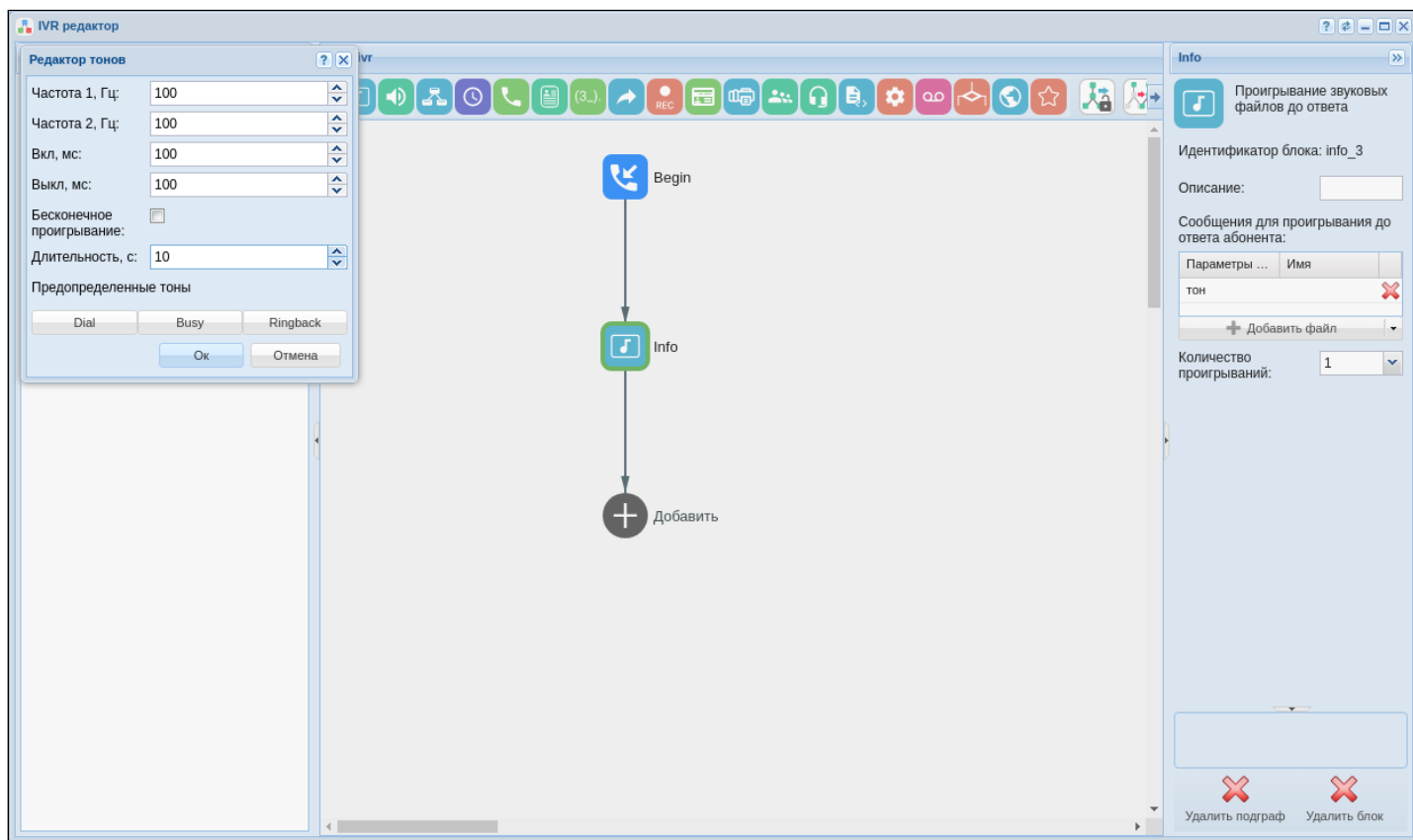


Рисунок 26 – Вид рабочей области IVR с настройками блока "Info"

- Блок dial – блок, который совершает вызов на заданный номер. Чтобы продолжить вызов, нужно применить predetermined переменную CDPN:

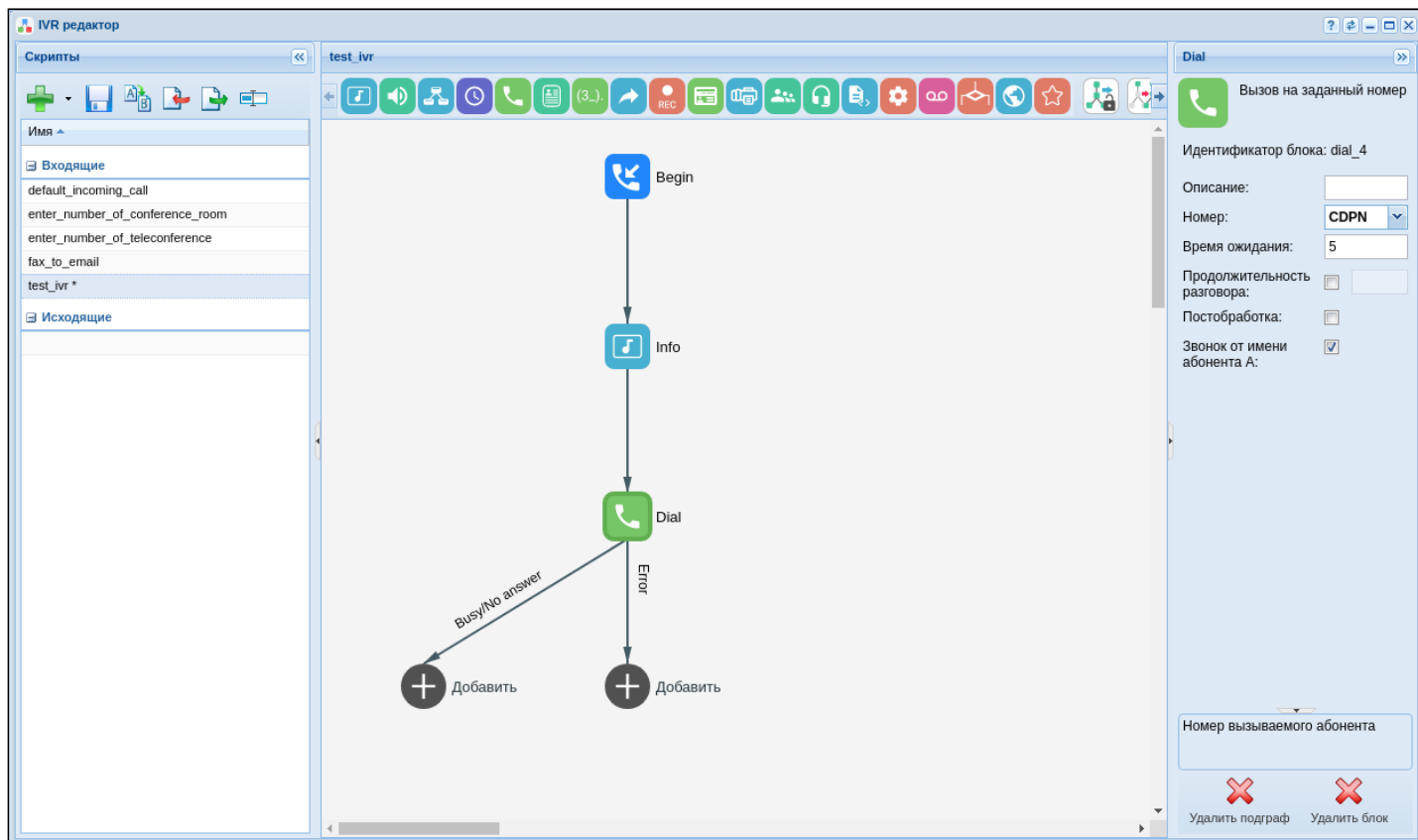




Рисунок 27 – Вид рабочей области IVR

11.3.10 Завершение настройки маршрутизации

Для завершения настройки маршрутизации нужно войти в приложение "Менеджер маршрутизации" ("Routing manager") и в правилах rule1 и rule3 настроить соответствующие результаты маршрутизации, для этого:

- rule1: в разделе "Результат" выберите подпункт "внешний". В таблице "Направления" в поле "Значение" добавьте созданный ранее транк, после чего сохраните правило .
- rule3: В разделе "Результат" выберите подпункт "ivr" и в поле скрипт добавьте созданный ранее ivr-скрипт, после чего сохраните правило и контекст .

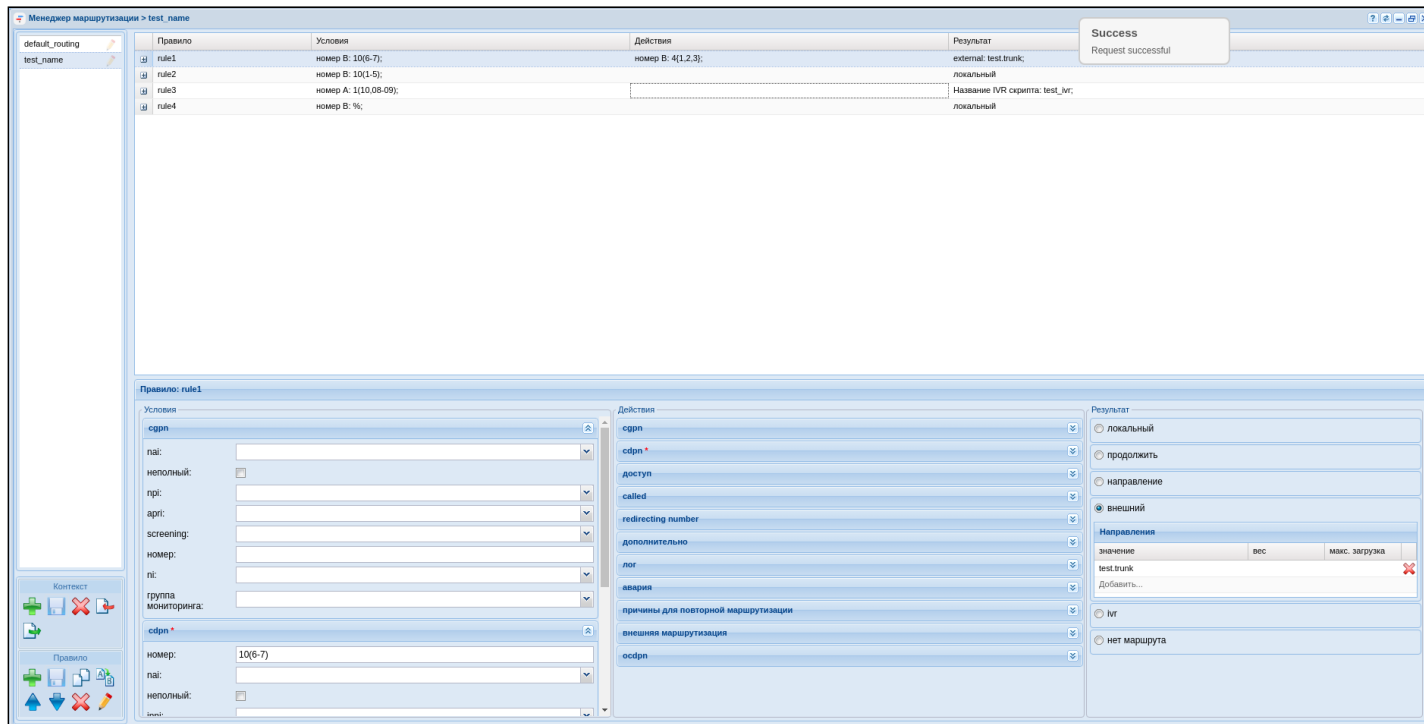


Рисунок 28 – Вид контекста маршрутизации

11.3.11 Настройка услуг

Для настройки услуг необходимо выполнить несколько действий:

- Установить услуги через CoCon;
- Добавить услуги для домена в список доступа (access-list) через CoCon;
- Далее любая услуга в списке доступа становится применимой на абоненте или транке.

Для входа в CLI "CoCon" можно воспользоваться терминалом или приложением ["Командная консоль CoCon" \("CoCon"\)](#).

После авторизации в Cocon пропишите следующие команды:

- Для установки услуг в систему:

```
cluster/storage/ds1/ss/install ds1@ecss1 *
```

- После успешной установки услуг пропишите следующую строку для добавления доступа к ним:

```
cluster/storage/ds1/ss/access-list add test_domain *
```

Символ "*" означает, что команда будет применена ко всем доступным элементам в системе. Если требуется установить конкретную услугу, вместо "*" вводится её название.

Для подключения услуг абоненту необходимо открыть карточку абонента, выбрать абонента из списка и перейти на вкладку *"Дополнительные услуги"*.

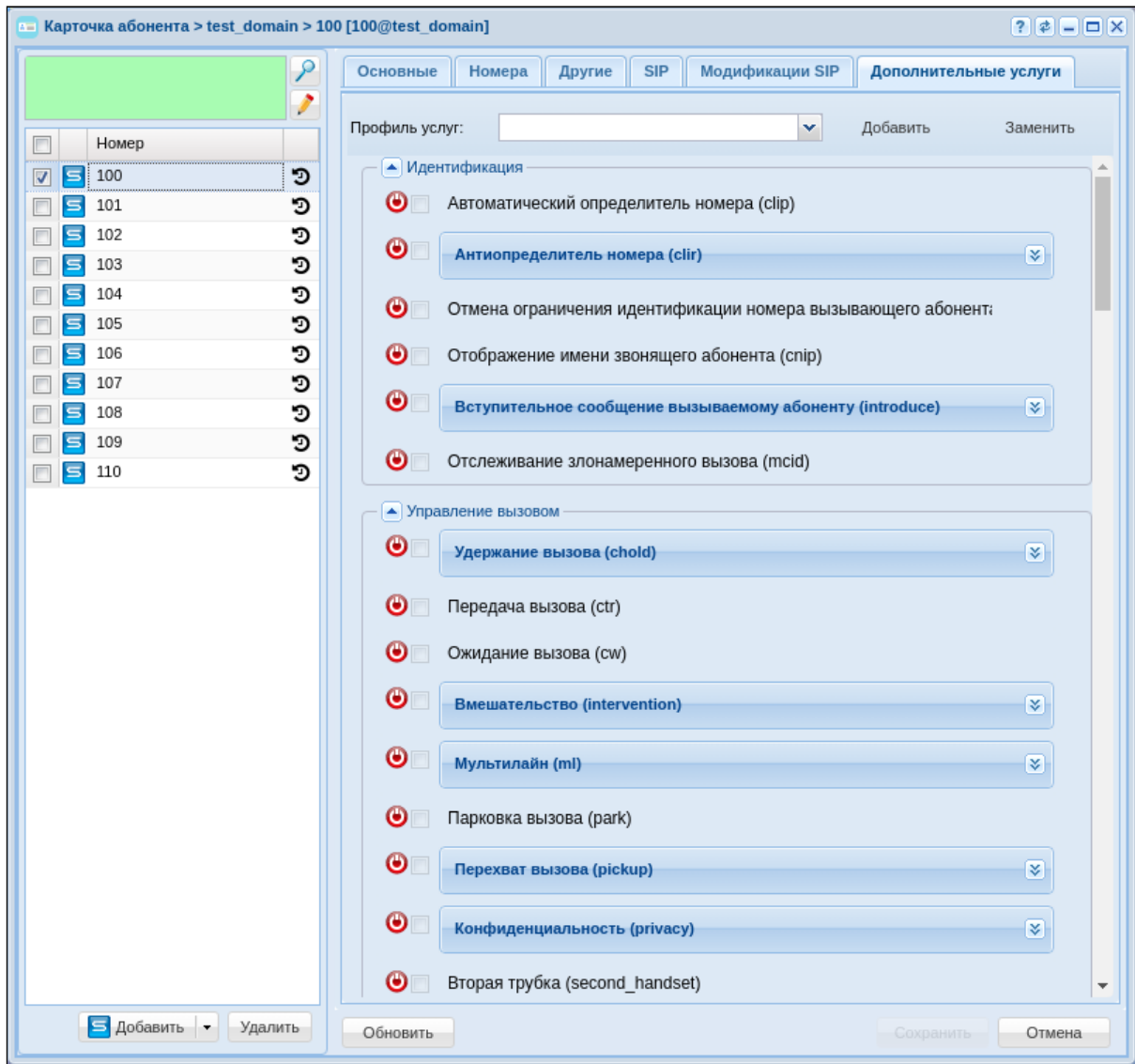



Рисунок 29 – Пример настройки услуг

Для активации услуги абоненту необходимо подключить ее, нажав кнопку , затем активировать и настроить (прочитать подробнее можно в разделе приложения "Карточка абонента" ("Subscriber card")).

12 Регламент обновления/деинсталляции ECSS

- Обновление с версии 3.11 на 3.14
 - Подготовка к обновлению системы
 - Обновление ОС
 - Установка ECSS-10 версии 3.14
 - Установка ecss-mysql
 - Настройка пакета ecss-dns-env
 - Установка ecss-node
 - Установка остальных пакетов
 - ecss-cc-ui
 - ecss-teleconference-ui
 - ecss-utils
 - ecss-asr
 - ecss-pda-api
 - ecss-autoprovision
 - ecss-clerk
 - ecss-crm-server
 - ecss-security
- Обновление второго сервера, если система в кластере
 - Сборка кластера
 - Запуск кластера
 - Проверка работы системы в кластере
- Деинсталляция компонентов ECSS
 - Удаление ecss-mysql
 - Удаление пакета ecss-node
 - Удаление пакета ecss-media-server
 - Удаление ненужных пакетов
 - Удаление пакетов по маске
 - Команды полной деинсталляции ECSS
 - Debconf

12.1 Обновление с версии 3.11 на 3.14

Рекомендуется производить обновление с новой установки операционной системы (Ubuntu Server 18.04.x LTS 64bit) и новой разметки дискового пространства.

В данном разделе описан процесс обновления операционной системы и программных компонентов ECSS-10, когда по каким-либо причинам новая установка невозможна.

12.1.1 Подготовка к обновлению системы

Обновить систему до последних версий:

```
sudo apt update
sudo apt upgrade
```

Выполнить полный «бэкап» (см. [«Регламентное обслуживание и резервное копирование»](#)).

Если нужны последние данные биллинга, то необходимо финализировать текущую cdr и скачать в надежное место, например:

```
admin@mycelium1@ecss1:/$ domain/bsk.test/cdr/make_finalize_cdr bsk3
Created file(s): cdrbsk3_20210330_11_05_38_t.csv
```

Посмотреть, сколько зарегистрированных абонентов, пример:

```
admin@mycelium1@ecss1:/$ domain/bsk.test/sip/user/registered *
```

```
....
```

```
elements: 320
```

При обновлении кластера следует обновлять сервера по очереди. Таким образом, при обновлении один из серверов будет всегда запущен и работа сервисов не будет прервана. Для разделения кластера следует использовать инструмент `ecss-control` на обоих серверах (Описание команд утилиты приведено в «[Приложении Н](#)»).

Пример:

```
sasha@ecss1:~$ sudo ecss-control detach
detach host: ecss2, ip: 10.10.0.2
complete
sasha@ecss1:~$ sudo ecss-control status
locked ip: 10.10.0.2
```

После разделения кластера необходимо выключить все ноды на сервере, который будет обновляться первым:

```
sudo systemctl stop ecss-mycelium.service
sudo systemctl stop ecss-ds.service
sudo systemctl stop ecss-core.service
sudo systemctl stop ecss-pa-sip.service
sudo systemctl stop ecss-pa-megaco.service
sudo systemctl stop ecss-mediator.service
sudo systemctl stop ecss-web-conf.service
sudo systemctl stop ecss-media-server.service
sudo systemctl stop ecss-subscriber-portal.service
sudo systemctl stop ecss-restfs.service
sudo systemctl stop ecss-cc-ui-api.service
```

Удалить пакеты `ecss-3.11`, сохранив конфигурации и базы:

```
sudo apt remove ecss-*-3.11
sudo apt autoremove
sudo apt upgrade
```

12.1.2 Обновление ОС

Версия 3.14 устанавливается на Ubuntu 18.04, а 3.11 на Ubuntu 16.04. Это означает, что перед установкой новой версии ECSS-10 требуется обновить ОС на сервере. Как только будут удалены старые пакеты ECSS-10, можно приступить к обновлению сервера:

⚠ Если во время обновления было обновлено ядро Linux, сервер следует перезагрузить.

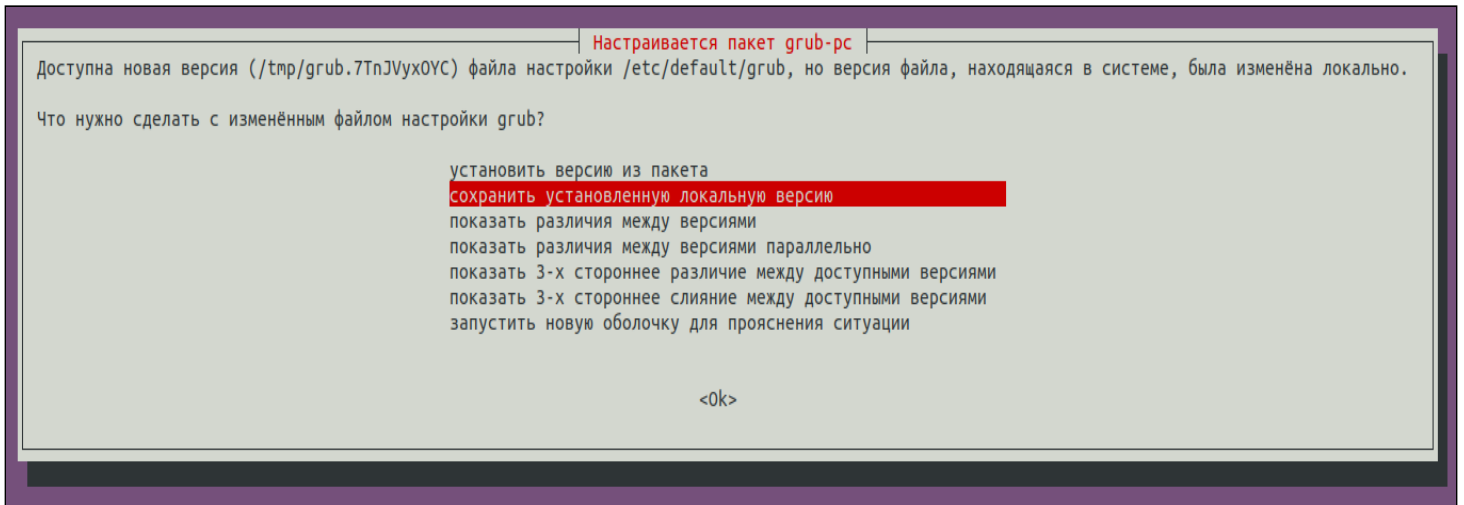
Еще раз убедиться, что все пакеты обновлены и ввести команду обновления ОС:

```
sudo apt upgrade
sudo do-release-upgrade
```

Если обновление происходит удаленно через SSH, система спросит, хотите ли вы продолжить и открыть еще один порт для ssh — 1022. Таким образом, будет возможность подключиться по этому порту при каких-либо проблемах.

В процессе обновления будут заданы различные вопросы, например, требуется ли сохранить существующий файл конфигурации или установить версию сопровождающего пакета. Если никаких изменений в файле не было, его следует вводить безопасно (Y), иначе сохранится текущая конфигурация (N).

Далее необходимо следовать инструкциям на экране. Во время процедуры обновления может быть предложено заменить существующий файл конфигурации **grub**. Необходимо убедиться, что была сохранена текущая версия, чтобы избежать возможных проблем:



После окончания обновления сервер нужно перезагрузить.

12.1.3 Установка ECSS-10 версии 3.14

Подробнее в разделе «[Инсталляция пакетов ECSS](#)».

Добавить в систему репозиторий версии 3.14:

```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/bionic/3.14 stable
main extras external' > /etc/apt/sources.list.d/eltex-ecss10-stable.list"
sudo apt update
```

Установка ecss-mysql

- Убедитесь, что система работает в кластере, а ноды и реплика не разъехались;
- Переведите трафик на второй сервер, выключив keepralive на первом хосте:

```
sudo systemctl stop keepalived.service
```

- Обновите ecss-mysql на первом хосте:

```
sudo apt install ecss-mysql
```

- ✓ Во время установки ecss-mysql будет предложено обновить конфигурацию apparmor. Следует ответить Yes (Y).

Если возникла проблема с какой-либо базой данных, следует произвести перезагрузку сервиса apparmor, затем удалить базу данных из mysql, а после снова установить ecss-mysql:

```
sudo systemctl restart apparmor.service  
mysql -uroot -p<root_mysql_password> -e 'DROP DATABASE <проблемная база данных>'  
sudo apt install ecss-mysql
```

- Произведите мониторинг состояния реплики с помощью SHOW SLAVE STATUS и дождитесь, пока показатель Seconds_Behind_Master станет равным нулю;
- Запустите keepralive обратно на первом хосте:

```
sudo systemctl start keepalived.service
```

- Переведите трафик на первый сервер, выключив keepralive на втором хосте;
- Обновите ecss-mysql на втором хосте;
- Запустите keepralive обратно на втором хосте.

Настройка пакета ecss-dns-env

Перед установкой пакета ecss-mysql вам будет предложено настроить переменные окружения для сервисов в dnsmasq. Настройщик предложит выбрать разделы для настройки. Если система в кластере, нужно выбрать broker и mysql. Адрес первичного брокера — IP ecss1, адрес вторичного брокера — IP ecss2, адрес mysql — "плавающий" адрес, прописанный в keepalived.

Если система не в кластере, ничего выбирать не нужно.

Установка ecss-node

Установить пакет ecss-node:

```
sudo apt install ecss-node
```

Во время установки пакета генерируются ssl-сертификаты. Ответы на вопросы можно оставить те, которые предлагаются, в том числе и пустые. Подробнее в разделе "[Инсталляция системы](#)".

Установка остальных пакетов

Установить пакеты, содержащие основные подсистемы:

```
sudo apt install ecss-restfs ecss-media-server ecss-web-conf ecss-media-resources
```

При установке `ecss-media-server` рекомендуется ответить на вопрос о конфигурировании утвердительно. В версии 3.14 в MSR добавлен новый функционал, поэтому конфигурационный файл рекомендуется обновить. При необходимости можно позже отредактировать его вручную.

В процессе установки будут задаваться вопросы о сохранении текущих версий конфигурационных файлов. Рекомендуется ставить новые, т.е. отвечать «Y».

В репозитории также хранятся дополнительные пакеты, которые можно установить опционально, исходя из проекта.

Для установки дополнительных пакетов выполните:

```
sudo apt install <имя пакета 1> <имя пакета 2> ... <имя пакета N>
```

Список доступных дополнительных пакетов:

<code>ecss-cc-ui</code>	Автоматизированное рабочее место оператора Call-центра
<code>ecss-teleconference-ui</code>	АРМ менеджера селекторного совещания
<code>ecss-utils</code>	Скрипты для конвертации бинарных логов в текст
<code>ecss-asr</code>	Сервис автоматического распознавания речи
<code>ecss-pda-api</code>	API для Phone Desktop Assistant
<code>ecss-autoprovision</code>	Сервис автоматического конфигурирования телефонных аппаратов (AUP)
<code>ecss-clerk</code>	Сервис " Автосекретарь "
<code>ecss-crm-server</code>	Сервер интеграции с CRM

ecss-security

Здесь также будет предложена замена всех конфигурационных файлов – необходимо согласиться (Y).
Для кластера после обновления конфигурационного файла /etc/ecss/ecss-mycelium/mycelium1.config необходимо прописать реальное имя кластера, например mycelium1:

```
sudo perl -pi -e 's/cluster_name, undefined/cluster_name, mycelium1/' /etc/ecss/ecss-
mycelium/mycelium1.config
cat /etc/ecss/ecss-mycelium/mycelium1.config | grep -e cluster_name
```

Проверить состояние сервисов, пример:

```
admin@mycelium1@ecss1:/$ system-status
Checking...
```

Node	Release	Erlang nodes	Mnesia nodes	Uptime
core1@ecss1	ecss-core-3.14.10.220	core1@ecss1	not running	51m 6s
ds1@ecss1	ecss-ds-3.14.10.220	ds1@ecss1	ds1@ecss1	51m 6s
md1@ecss1	ecss-mediator-3.14.10.220	md1@ecss1	md1@ecss1	51m 6s
mycelium1@ecss1	ecss-mycelium-14.10.220	mycelium1@ecss1	not running	51m 6s
sip1@ecss1	ecss-pa-sip-14.10.220	sip1@ecss1	sip1@ecss1	51m 6s

All services are started.

Active media resource selected list specific:

Node	MSR	MSR version	Cc-status	Cc-uptime
core1@ecss1	msr_1	3.14.10.67	connected	00:50:28

Alarms list on md1:

Date	Severity	Location	Cause	Message
Class	Instance			
29.03 21:33:36	warning	core1@ecss1	versionMismatch	
ecss::cluster::core::ss	acb		Supplementary service acb has	invalid version. You should update it.
29.03 21:33:36	warning	core1@ecss1	versionMismatch	
ecss::cluster::core::ss	sca		Supplementary service sca has	invalid version. You should update it.
29.03 21:33:36	warning	core1@ecss1	versionMismatch	

ecss::cluster::core::ss rbp invalid version. You should update it.	Supplementary service rbp has
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss cliro invalid version. You should update it.	Supplementary service cliro has
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss intervention has invalid version. You should updat	Supplementary service intervention
e it.	
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss speed_dial has invalid version. You should update	Supplementary service speed_dial
it.	
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss fax_to_email has invalid version. You should updat	Supplementary service fax_to_email
e it.	
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss park invalid version. You should update it.	Supplementary service park has
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss 3way invalid version. You should update it.	Supplementary service '3way' has
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss introduce invalid version. You should update i	Supplementary service introduce has
t.	
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss sco_white invalid version. You should update i	Supplementary service sco_white has
t.	
29.03 21:33:36 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss cc_agent invalid version. You should update it	Supplementary service cc_agent has
.	
29.03 21:33:37 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss second_handset second_handset has invalid version. You should uppd	Supplementary service
ate it.	
29.03 21:33:37 warning core1@ecss1	versionMismatch
ecss::cluster::core::ss rfc invalid version. You should update it.	Supplementary service rfc has


```
cluster/storage/ds1/ss/update *
```

12.2 Обновление второго сервера, если система в кластере

Чтобы перевести трафик на обновленный сервер, нужно на втором сервере остановить службу keepalived:

```
sudo systemctl stop keepalived.service
```

Если нужны последние данные биллинга, то необходимо финализировать текущую cdr и скачать в надежное место, например:

```
admin@mycelium1@ecss1:/$ domain/bsk.test/cdr/make_finalize_cdr bsk3  
Created file(s): cdrbsk3_20220128_10_06_40_t.csv
```

Дальнейшее обновление аналогично описанию, приведенному [выше](#), с учетом настроек MSR для данного хоста.

12.2.1 Сборка кластера

После установки пакетов ecss-3.14 на данном этапе следует остановить сервисы ecss:

```
sudo systemctl stop ecss-mycelium.service  
sudo systemctl stop ecss-ds.service  
sudo systemctl stop ecss-core.service  
sudo systemctl stop ecss-pa-sip.service  
sudo systemctl stop ecss-pa-megaco.service  
sudo systemctl stop ecss-mediator.service  
sudo systemctl stop ecss-web-conf.service  
sudo systemctl stop ecss-media-server.service  
sudo systemctl stop ecss-restfs.service  
sudo systemctl stop ecss-cc-ui-api.service
```

Настройка dnsmasq

При стандартном обновлении должны оставаться default-конфигурации – в качестве IP-адреса указан 127.0.0.1. Единственное, что сразу при установке пакета ecss-mysql можно настроить, это адрес secondary.broker. Пока кластер разделен, серверы не видят друг друга.

При обновлении системы в кластере в настройках dnsmasq требуется указать IP-адреса обоих хостов (в примере 10.10.0.1 и 10.10.0.2 – статические IP-адреса хостов, а 10.10.0.15 VRRP IP-адрес mysql, настроенный в keepalived). Затем обычно достаточно отредактировать файл /etc/dnsmasq.d/ecss-mysql на обоих серверах.

Пример настроек на ecss1/ecss2:

```
sasha@ecss1:~/ecss1_3.14$ cat /etc/dnsmasq.d/ecss-broker
address=/primary.broker.ecss/10.10.0.1
address=/secondary.broker.ecss/10.10.0.2
```

```
sasha@ecss1:~/ecss1_3.14$ cat /etc/dnsmasq.d/ecss-mysql
address=/address-book.mysql.ecss/10.10.0.15
address=/cocon.mysql.ecss/10.10.0.15
address=/sp.mysql.ecss/10.10.0.15
address=/statistics.mysql.ecss/10.10.0.15
address=/tc.mysql.ecss/10.10.0.15
address=/tts.mysql.ecss/10.10.0.15
address=/dialer.mysql.ecss/10.10.0.15
address=/numbers.mysql.ecss/10.10.0.15
```

Затем можно удалить ненужные файлы, оставшиеся от предыдущей версии (они отличаются датой последнего изменения). Следует оставить следующие файлы:

```
sasha@ecss1:~/ecss1_3.14$ ls /etc/dnsmasq.d/
README ecss ecss-broker ecss-ldap ecss-mysql ecss-notifier ecss-restfs ecss-sip
ecss-sp ecss-syslog lxd
```

Для применения настроек нужно перезапустить dnsmasq на обоих хостах и сразу проверить статус.

Пример:

```
sasha@ecss1:~$ sudo systemctl restart dnsmasq.service
sasha@ecss1:~$
sasha@ecss1:~$ sudo systemctl status dnsmasq.service
● dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
   Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Tue 2021-03-30 15:06:53 +07; 31s ago
     Process: 19252 ExecStop=/etc/init.d/dnsmasq systemd-stop-resolvconf (code=exited,
   status=0/SUCCESS)
     Process: 19344 ExecStartPost=/etc/init.d/dnsmasq systemd-start-resolvconf
   (code=exited, status=0/SUCCESS)
     Process: 19325 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited, status=0/
   SUCCESS)
     Process: 19304 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
   Main PID: 19341 (dnsmasq)
     Tasks: 1 (limit: 4616)
    CGroup: /system.slice/dnsmasq.service
           └─19341 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -r /run/
   dnsmasq/resolv.conf -7 /etc/dnsmasq.d,.dpkg-dist,.dpkg-old,.dpkg-new --local-service
   --trust-anchor=.,19036,8,2,49aac11d7b

map 30 15:06:53 ecss1 systemd[1]: Starting dnsmasq - A lightweight DHCP and caching
   DNS server...
map 30 15:06:53 ecss1 dnsmasq[19304]: dnsmasq: syntax check OK.
map 30 15:06:53 ecss1 systemd[1]: Started dnsmasq - A lightweight DHCP and caching
   DNS server.
```

12.2.2 Запуск кластера

Запустить сервисы ecss на втором хосте:

```
sudo systemctl start ecss-mycelium.service
sudo systemctl start ecss-ds.service
sudo systemctl start ecss-core.service
sudo systemctl start ecss-pa-sip.service
sudo systemctl start ecss-pa-megaco.service
sudo systemctl start ecss-mediator.service
sudo systemctl start ecss-web-conf.service
sudo systemctl start ecss-media-server.service
sudo systemctl start ecss-subscriber-portal.service
sudo systemctl start ecss-restfs.service
sudo systemctl start ecss-cc-ui-api.service
```

После запуска всех сервисов (проверить командой **systemctl status <service>**) можно запускать сервис **keepalived**:

```
sudo systemctl start keepalived.service
```

Далее следует запустить репликацию БД MySQL. Выполняется на первом сервере. Подробнее в разделе [«Схема развертывания MySQL master-master replication с использованием keepalive»](#).

Теперь можно открывать серверы друг другу, чтобы свести в кластер. На первом сервере выполнить команду **sudo ecss-control attach**. Описание команд утилиты приведено в [«Приложении Н»](#).

Пример:

```
sasha@ecss1:~/ecss1_3.14$ sudo ecss-control attach
unlock ip: 10.10.0.2, rules: 7
complete
sasha@ecss1:~/ecss1_3.14$ sudo ecss-control status
all nodes attached
```

12.2.3 Проверка работы системы в кластере

Проверить выполнение обновления. Команда CLI **system-status** должна показать, что все ноды в кластере, версии подсистем одинаковые.

```
admin@mycelium1@ecss1$ system-status
```

```
Checking...
```

Node		Release	Erlang nodes	
Mnesia nodes	Uptime			
core1@ecss1 running	ecss-core-3.14.10.220 1h 19m	ecss-core-3.14.10.220	core1@ecss1, core1@ecss2	not
core1@ecss2 running	ecss-core-3.14.10.220 4m 28s	ecss-core-3.14.10.220	core1@ecss1, core1@ecss2	not
ds1@ecss1 ds1@ecss1, ds1@ecss2	ecss-ds-3.14.10.220 1h 19m	ecss-ds-3.14.10.220	ds1@ecss1, ds1@ecss2	
ds1@ecss2 ds1@ecss1, ds1@ecss2	ecss-ds-3.14.10.220 4m 28s	ecss-ds-3.14.10.220	ds1@ecss1, ds1@ecss2	
md1@ecss1 md1@ecss1, md1@ecss2	ecss-mediator-3.14.10.220 1h 19m	ecss-mediator-3.14.10.220	md1@ecss1, md1@ecss2	
md1@ecss2 md1@ecss1, md1@ecss2	ecss-mediator-3.14.10.220 4m 28s	ecss-mediator-3.14.10.220	md1@ecss1, md1@ecss2	
mycelium1@ecss1 running	ecss-mycelium-3.14.10.220 1h 19m	ecss-mycelium-3.14.10.220	mycelium1@ecss1, mycelium1@ecss2	not
mycelium1@ecss2 running	ecss-mycelium-3.14.10.220 4m 28s	ecss-mycelium-3.14.10.220	mycelium1@ecss1, mycelium1@ecss2	not
sip1@ecss1 sip1@ecss1, sip1@ecss2	ecss-pa-sip-3.14.10.220 1h 19m	ecss-pa-sip-3.14.10.220	sip1@ecss1, sip1@ecss2	
sip1@ecss2 sip1@ecss1, sip1@ecss2	ecss-pa-sip-3.14.10.220 4m 28s	ecss-pa-sip-3.14.10.220	sip1@ecss1, sip1@ecss2	

All services are started.

Active media resource selected list specific:

Node	MSR	MSR version	Cc-status	Cc-uptime
core1@ecss1	msr_1	3.14.10.67	connected	01:19:21
	msr_2	3.14.10.67	connected	00:04:23
core1@ecss2	msr_1	3.14.10.67	connected	00:03:58
	msr_2	3.14.10.67	connected	00:03:56

Далее:

- убедиться, что абоненты зарегистрированы. Проверить количество зарегистрированных абонентов, оно должно быть приблизительно тем же, что и в начале обновления:

```

admin@mycelium1@ecss1:/$ domain/bsk.test/sip/user/registered *
320 make users list ...
[*****] 223mks

320 users information read ...
[*****] 22ms

Executed on the sip1@ecss1

+-----+-----+-----+-----+
| Interface | Number | Dynamic | Contact |
|           |         | Registration info |         |
+-----+-----+-----+-----+
...
elements: 320

```

- вызовы проходят во всех направлениях;
- CFC проигрывает сообщения;
- Web-conf и прочие web-сервисы запускаются;
- в системный журнал не пишется ошибок. Команда **shell**:

```
journalctl -f
```

12.3 Деинсталляция компонентов ECSS

12.3.1 Удаление ecss-mysql

Для удаления пакета ecss-mysql с сохранением баз можно использовать команду:

```
sudo apt remove ecss-mysql
```

При этом базы и конфигурационные файлы не удаляются.

Для полного удаления пакета вместе с базами и конфигурационными файлами нужно использовать ключ **purge**:

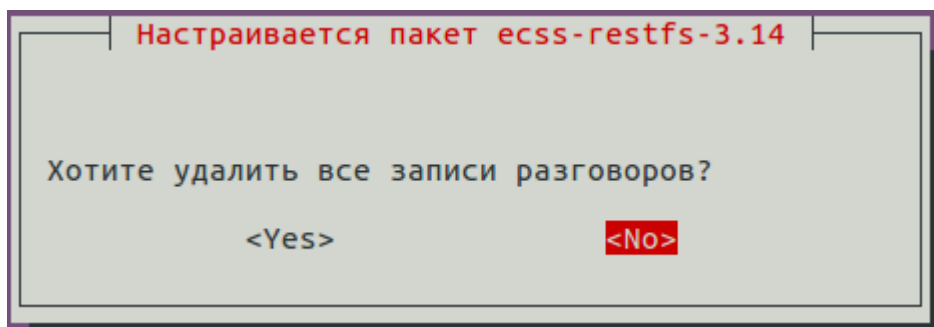
```
sudo apt purge ecss-mysql
```

При этом по каждой базе нужно подтверждение удаления. После этого каталог **/var/lib/ecss-mysql/** будет очищен.

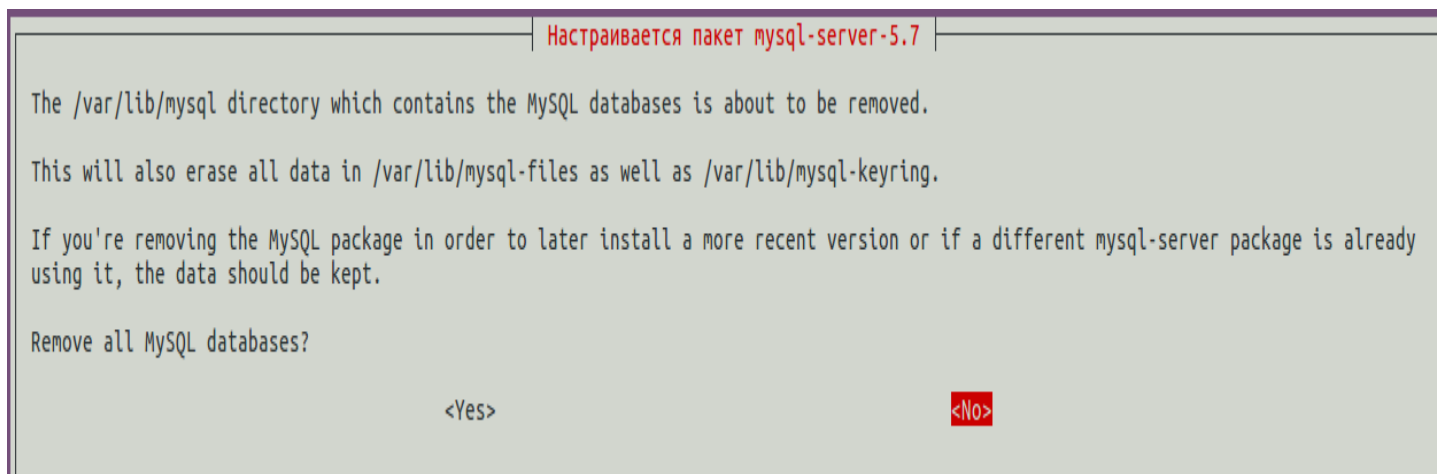
Если нужно полностью удалить все пакеты mysql*:

❗ Удаление пакетов mysql-* удалит также пакеты ecss-web-conf и ecss-restfs.

При удалении ecss-restfs задается вопрос об удалении записей разговоров:



А также об удалении всех БД MySQL:



```
sudo apt purge mysql-*
```

12.3.2 Удаление пакета ecss-node

Для удаления пакета использовать команду:

```
sudo apt purge ecss-node
```

12.3.3 Удаление пакета ecss-media-server

Для удаления пакета с сохранением конфигурационных файлов нужно использовать команду:

```
sudo apt remove ecss-media-server
```

При этом в каталоге **/etc/ecss/ecss-media-server/** остаются файлы конфигурации.

Для полного удаления:

```
sudo apt purge ecss-media-server
```

12.3.4 Удаление ненужных пакетов

Для удаления остальных пакетов, которые больше не требуются, нужно использовать ключ autoremove:

```
sudo apt autoremove
```

12.3.5 Удаление пакетов по маске

Для удаления сразу нескольких пакетов можно пользоваться масками (см. man apt).

Пример команды для удаления всех пакетов ecss:


```
sudo apt purge ecss-*
```

12.3.6 Команды полной деинсталляции ECSS

Для полного удаления всех компонентов ECSS нужно выполнить 3 команды:

```
sudo apt purge ecss- *  
sudo apt purge mysql- *  
sudo apt autoremove
```

После следует перезагрузить сервер для удаления некоторых процессов из оперативной памяти.

 В каталоге `/var/log/ecss/security/` в служебных целях остаются логи. Их нельзя удалить штатными средствами. Если необходима очистка этого каталога, следует обратиться в [техническую поддержку](#).

12.3.7 Debconf

В некоторых случаях вместо переустановки пакета требуется только изменение конфигурации. Здесь следует применять команду `dpkg-reconfigure <имя_пакета>`. Также при необходимости можно использовать команды пакета `debconf-utils`.

Описание приведено на странице «[Приложение В](#)».

13 Рекомендации по обеспечению безопасности системы

- Общие требования
 - Отключить доступ через root на сервер
 - Включить доступ только по ключам ssh
 - Использовать безопасные ключи rsa -b 4096
 - Автоматическая генерация паролей для SIP-пользователей
 - Изменить пароли на безопасные
 - Использовать SBC, настроить firewall, ACL
 - Обновлять Ubuntu и утилиты сервера при выходе пакета безопасности
- Настройка безопасности OpenSSH
 - Настройка конфигурации сервера и клиента ssh
 - Обмен ключами
 - Аутентификация
 - Симметричные шифры
 - Код аутентичности сообщения (MAC)
 - Защита от утечки ключей
 - Ограничение пользователей
 - Ограничение интерфейсов
 - Смена порта

13.1 Общие требования

Для обеспечения необходимых настроек безопасности системы рекомендуется:

13.1.1 Отключить доступ через root на сервер

По умолчанию учетная запись root отключена в Ubuntu, в отличие от других дистрибутивов Linux. Если root все-таки включен, отключить можно командой:

```
sudo passwd -l root
```

13.1.2 Включить доступ только по ключам ssh

В /etc/ssh/sshd_config:

```
PasswordAuthentication no  
ChallengeResponseAuthentication no
```

13.1.3 Использовать безопасные ключи rsa -b 4096

```
sudo ssh-keygen -t rsa -b 4096 -f /etc/ssh/ssh_host_rsa_key  
sudo ssh-keygen -o -a 100 -t ed25519 -f /etc/ssh/ssh_host_ed25519_key
```

13.1.4 Автоматическая генерация паролей для SIP-пользователей

Система ECSS-10 имеет возможность при создании SIP-абонентов генерировать пароли, которые соответствуют рекомендуемым требованиям безопасности.

Пример декларации абонента:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/sip/user/declare ctx_from_local loc.gr
240800 alias-as-user qop_authentication common_login sip3854240800 auto-generation
Executed on the sip1@ecss1
Intermediate (incomplete) result:
Declaration for range: 240800@biysk.local..240800@biysk.local (1)
...
1 interfaces check for existing ...
[*****] 50mks
1 users interfaces declaration ...
[*****] 5ms
1 users aliases declaration ...
[*****] 29ms
322 interfaces recall to base
[*****] 109ms
Executed on the sip1@ecss1

| declared 1 subscribers |

[exec at: 23.01.2020 22:53:18, exec time: 534ms, nodes: sip1@ecss1]
```

Просмотр учетных данных:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/sip/user/info loc.gr 240800@biysk.local
1 make users list ...
[*****] 19mks
1 users information read ...
[*****] 14ms
Executed on the sip1@ecss1
```

User	240800@biysk.local
internal iface name	062bc20adc06e101
isActive	false
group	loc.gr
authentication	sip3854240800:uc7cXcnuyunu (ds) use qop: true

13.1.5 Изменить пароли на безопасные

По умолчанию после инсталляции ECSS для доступа в CLI и web-конфигуратор автоматически создается пользователь **admin** со стандартным паролем **password**.

Нужно создать нового пользователя с правами администратора системы, у которого логин и пароль исключает возможность подбора. Рекомендуется использовать не менее 8 символов, включающих в

себя строчные и заглавные буквы латинского алфавита, а также цифры и спецсимволы. Можно воспользоваться стандартной утилитой Linux – **pwgen**. После этого удалить пользователя **admin**.

Аналогичные требования и для пользователя `mysql` с правами `root` (задается при инсталляции пакета `ecss-mysql`). При необходимости пароль можно изменить реконфигурацией пакета:

```
sudo dpkg-reconfigure ecss-mysql
```

13.1.6 Использовать SBC, настроить firewall, ACL

Рекомендуется настроить свой `firewall` таким образом, чтобы `Restfs` был доступен только с самого сервера и недоступен внешне.

Если сервер или SBC имеет прямой доступ в интернет или корпоративную сеть, где не исключены разного рода попытки неавторизованных запросов или DDoS-атаки, нужно настроить список разрешенных IP или подсетей, с которых разрешен доступ к определенным портам. Для остальных доступ должен быть запрещен. Если в составе комплекса есть SBC, то в нем уже реализован необходимый функционал для защиты от внешних атак – динамический брандмауэр с механизмом распознавания атак и попыток несанкционированного доступа. В SBC также можно настроить черные/белые списки подсетей для доступа как по управлению, так и по SIP.

13.1.7 Обновлять Ubuntu и утилиты сервера при выходе пакета безопасности

Рекомендуется не пренебрегать периодическими обновлениями системы, особенно при выпуске свежих версий пакетов, улучшающих функционал безопасности.

13.2 Настройка безопасности OpenSSH

На серверах ECSS должны быть установлены пакеты `openssh-client` `openssh-server` или мета-пакет `ssh`, включающий в себя клиента и сервер.

13.2.1 Настройка конфигурации сервера и клиента ssh

Конфигурация сервера OpenSSH описана в файле `/etc/ssh/sshd_config`. Для более полного обеспечения безопасности необходимо изменить некоторые настройки, принятые по умолчанию.

В репозитории Ubuntu-18.04 LTS текущая версия OpenSSH_7.6p1. Нижеприведенные рекомендации по конфигурации для версий, начиная с 7.6.

В OpenSSH 7.0 и старше отключена поддержка протокола SSH 1.0 по умолчанию во время компиляции. Точно так же по умолчанию во время компиляции больше не активируется поддержка ключей `diffie-hellman-group1-sha1` размером 1024 бит, хостов и пользовательских ключей `ssh-dss`, `ssh-dss-cert-*`. Не поддерживается формат сертификатов `v00`, а опция `PermitRootLogin` по умолчанию изменила значение с `yes` на `prohibit-password`.

Обмен ключами

Для использования только заслуживающих доверия протоколов обмена ключами в `/etc/ssh/sshd_config` для сервера следует указать:

```
KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
```

Аналогичные настройки для клиента, в `/etc/ssh/ssh_config`:

```
Host *
    KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha25
```

Аутентификация

В SSH поддерживаются четыре алгоритма аутентификации по открытым ключам: DSA, ECDSA, Ed25519 и RSA.

ECDSA завязан на технологиях NIST и должен быть отключен. Так как размер ключей DSA не может превышать 1024 бит, его тоже следует отключить. Поэтому рекомендуется удалить все ключи:

```
cd /etc/ssh
rm ssh_host_*key*
```

Далее следует сгенерировать ключ RSA большего размера, а также добавить ключ ed25519:

```
sudo ssh-keygen -t rsa -b 4096 -f /etc/ssh/ssh_host_rsa_key
sudo ssh-keygen -o -a 100 -t ed25519 -f /etc/ssh/ssh_host_ed25519_key
```

Симметричные шифры

Из 15 поддерживаемых в SSH алгоритмов симметричного шифрования, используемых для организации защиты установленного канала связи, безопасными можно считать chacha20-poly1305, aes*-ctr и aes*-gcm. Шифры 3des-cbc и arcfour потенциально уязвимы в силу использования DES и RC4, cast128-cbc применяет слишком короткий размер блока (64 бит).

Поэтому в /etc/ssh/sshd_config рекомендуется добавить:

```
Ciphers aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-
poly1305@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
```

В /etc/ssh/sshd_config:

```
Host *
    Ciphers aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-
poly1305@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
```

Код аутентичности сообщения (MAC)

Для шифров в режиме CTR для гарантирования целостности передаваемых блоков доверия заслуживает только метод Encrypt-then-MAC ("*-etm", MAC добавляется к уже зашифрованному блоку). Методы MAC-then-encrypt и Encrypt-and-MAC потенциально подвержены атакам. Из 18 доступных в SSH алгоритмов MAC сразу следует отбросить основанные на хэшах MD5 и SHA1, не стойких к выявлению коллизий, а также алгоритмы, использующие размеры ключей менее 128 бит и размеры тегов менее 256 бит. В итоге, наиболее безопасными MAC можно считать hmac-sha2-512-etm и hmac-sha2-256-etm.

В /etc/ssh/sshd_config:

```
MACs hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com
```

В /etc/ssh/ssh_config:

```
Host *  
MACs hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com
```

Защита от утечки ключей

Наиболее простым способом получения контроля за SSH-соединением является захват ключей на стороне клиента или сервера. Рекомендации сводятся к соблюдению типовых правил поддержания безопасности системы:

- оперативная установка обновлений, установка программ только из надёжных источников;
- установка только действительно необходимых программ и сервисов;
- использование программ, для которых доступны исходные тексты, включение дополнительных механизмов защиты (Grsecurity, сборка с флагом `-fstack-protector`).

Для защиты ключей следует выбрать надёжный пароль доступа к клиентским файлам ключей. При формировании ключа для увеличения числа итераций хэширования можно использовать опцию `"ssh-keygen -o -a число"`, что усложнит подбор пароля. Также можно сохранить ключи только на внешнем носителе, подключая его только во время соединения по SSH.

Ограничение пользователей

Добавляем параметр `AllowUsers`, которого нет в конфигурационном файле по умолчанию. Этот параметр разрешает доступ к серверу по протоколу SSH только для перечисленных пользователей.

В /etc/ssh/sshd_config:

```
AllowUsers dmitry support
```

Ограничение интерфейсов

Эти строки отвечают за настройку разграничений по сетевым интерфейсам, сетевому адресу или имени компьютера. По умолчанию сервер «слушает» (принимает подключения) на всех сетевых интерфейсах. Если нужно оставить подключение только через определенные интерфейсы, то раскомментировать строку:

```
ListenAddress 192.168.1.21  
ListenAddress 10.16.33.5
```

В этой же строке можно явно указать порт, предварительно закоментировав (поставив символ `#` в начале строки)

```
#ListenAddress 0.0.0.0
```

Смена порта

Если есть вероятность попыток подбора пароля (например сервер в публичной сети) можно изменить порт на другой.

```
Port 22025
```

По умолчанию используется 22 порт. Изменим его на нестандартный порт 22025 — это избавит наш сервер от сетевых роботов, которые автоматически сканируют интернет в поиске открытых портов и пытаются через них подключиться. В основном, боты настроены на поиск стандартных портов. Это не избавит от сканирования человеком, но для защиты от человека существует фаервол, хитрые способы открытия порта и пр.

Примерный листинг получившихся файлов конфигурации:

[/etc/ssh/sshd_config](#):


```

#   $OpenBSD: sshd_config,v 1.101 2017/03/14 07:19:07 djm Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Port 22025
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
ListenAddress 192.168.1.21
ListenAddress 10.16.33.5

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes

```

```

#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#UseLogin no
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp /usr/lib/openssh/sftp-server

```

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#   X11Forwarding no
#   AllowTcpForwarding no
#   PermitTTY no
#   ForceCommand cvs server

# @override for ECSS

KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
Ciphers aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com

HostKey /etc/ssh/ssh_host_ed25519_key
HostKey /etc/ssh/ssh_host_rsa_key
```

[/etc/ssh/ssh_config:](#)

```

# This is the ssh client system-wide configuration file. See
# ssh_config(5) for more information. This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
# 1. command line options
# 2. user-specific file
# 3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options. For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no
# BatchMode no
# CheckHostIP yes
# AddressFamily any
# ConnectTimeout 0
# StrictHostKeyChecking ask
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519
# Port 22
# Protocol 2
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
# MACs hmac-md5,hmac-sha1,umac-64@openssh.com
# EscapeChar ~
# Tunnel no
# TunnelDevice any:any
# PermitLocalCommand no
# VisualHostKey no
# ProxyCommand ssh -q -W %h:%p gateway.example.com
# RekeyLimit 1G 1h
SendEnv LANG LC_*
HashKnownHosts yes
GSSAPIAuthentication yes
# @override for ECSS
Ciphers aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com

```

Проверить синтаксис файла конфигурации можно командой:

```
sshd -t
```

Посмотреть текущие настройки сервера:

```
sshd -T
```

14 Чек-лист по установке ECSS-10

Шаг	Пункт документации	Результат	Примечание
1. Проверка соответствия аппаратных требований к системе			
1.1 Установлены актуальные версии ПО внутренних устройств и контроллеров сервера (BIOS, RAID, iDRAC, iLO и т.д.)			См. документацию к используемому серверу
1.2 Наличие в системе по меньшей мере двухъядерного CPU	Разработка проекта системы		Определяется проектом на систему
1.3 Наличие в системе по меньшей мере 8 ГБ RAM	Разработка проекта системы		Определяется проектом в зависимости от требуемой нагрузки. В тестовых целях допускается уменьшение до 4 ГБ
1.4 Имеется достаточно дискового пространства	Создание разделов диска		
1.5 Произведена верная разметка разделов	Создание разделов диска		
1.6 В системе НЕ используется файл подкачки SWAP	Отключение swap		
2. Проверка соответствия программных требований к системе			
2.1 Установлена корректная версия и разрядность операционной системы (Ubuntu Server 18.04.x LTS 64bit)			Ubuntu Server 18.04 x64
2.2 Выставлено корректное имя хоста hostname: ecss1 или ecss2	Настройка имени пользователя и сервера		При использовании резервирования
2.3 В системе установлены актуальные обновления	Обновление системы		
2.4 В системе установлено рекомендуемое ПО	Инсталляция необходимого ПО		
2.5 В системе добавлен репозиторий ECSS-10 и имеется доступ к нему	Обновление системы		
2.6 В системе корректно настроены сетевые интерфейсы:	Настройка сетевых интерфейсов		
- адреса сетевых интерфейсов статичны, не изменяются динамически по DHCP	Настройка сетевых интерфейсов		
- сервис <i>keepalived</i> (VRRP) настроен и работает	Настройка VRRP		При использовании резервирования

Шаг	Пункт документации	Результат	Примечание
- как минимум на интерфейсах lo включена поддержка IPv6	Настройка listen-интерфейса для сервиса eptmd		<p>Пример:</p> <pre>saw@ecss1:~\$ ip -6 addr show lo 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1 inet6 ::1/128 scope host valid_lft forever preferred_lft forever</pre>
2.7 Ключ eToken/RuToken подключен к USB-порту сервера:	Проверка работы Token		
- Ключ eToken/RuToken определен в системе (lsusb)	Проверка работы Token		
- Проверка ключа eToken/RuToken	Проверка работы Token		
2.8 Проверка сетевой доступности между хостами ecss1 и ecss2:			При использовании резервирования
- задержка ping между хостами не превышает 100 мс			Необходимо минимизировать задержки трафика между хостами, насколько это возможно
- отсутствуют потери пакетов при обмене данными между хостами			<p>Запускается команда ping на несколько минут, после чего снимается статистика комбинацией клавиш:</p> <pre>CTRL+ </pre> <p>Пример:</p> <pre>saw@ecss1:~\$ ping -q -c 200 ecss2 PING ecss2 (192.168.1.22) 56(84) bytes of data. 177/177 packets, 0% loss, min/avg/ewma/max = 0.062/0.131/0.117/0.489 ms</pre>
- хосты доступны между собой по DNS-имени (как прописано в /etc/hosts/ и в соответствии с лицензией)	Настройка /etc/hosts		Проверяется ping с ecss1 на ecss2 и наоборот

Шаг	Пункт документации	Результат	Примечание
- между хостами настроена авторизация по ключам			Сгенерировать ssh-ключ и передать на другой сервер: <pre>ssh-keygen ssh-copy-id ssw@ecss2</pre> Аналогично с ecss2 на ecss1
2.9 На сервере работает синхронизация времени с NTP-сервером:	Синхронизация времени на серверах		Проверить статус: <pre>ntpq -p</pre>
- выставлен корректный часовой пояс			Просмотреть: <pre>date +%Z</pre>
- на хостах ecss1 и ecss2 настроена синхронизация времени в режиме <i>orphan</i> либо синхронизация с единого NTP	Синхронизация времени на серверах		При использовании резервирования
2.10. Установлен glusterfs-server рекомендуемой версии:	Настройка RestFS		Не ниже версии ПО 3.13.2
3. Проверка корректности установки ECSS-10			
3.1 Установка пакетов компонентов системы завершилась успешно без ошибок:	Инсталляция пакетов ECSS		<pre>dpkg -l grep ecss</pre>
- ecss-mysql	Установка ecss-mysql		
- ecss-node	Установка ecss-node		
- ecss-media-resources	Установка ecss-media-resources		
- ecss-media-server	Установка ecss-media-server		
- ecss-restfs	Установка ecss-restfs		

Шаг	Пункт документации	Результат	Примечание
- ecss-user			
- ecss-web-conf	Установка ecss-web-conf		
- ecss-dns-env	DNS		
- дополнительные необязательные пакеты в соответствии с проектом	Установка дополнительных пакетов		
4. Проверка работоспособности системы ECSS-10			
4.1 Проверка установленной лицензии ECSS-10:	Запуск и активация системы		
- сервис pcsd находится в работе	Установка ПО и подключение Token		<pre>pkcs11-tool --module \$ (find /usr/lib/ecss/ecss-ds/ lib/ -name librtpkcs11ecp.so head -n1) -L</pre>
- в системе установлен корректный паспорт	Запуск и активация системы		<p>CoCon команда:</p> <pre>/cluster/storage/ds1/ licence/show-passport</pre>
- в системе установлена актуальная лицензия	Запуск и активация системы		<p>CoCon команда:</p> <pre>/cluster/storage/ds1/ licence/current-limits</pre>
4.2 Сервис dnsmasq запущен и работает:			
- если система с резервом, в конфигурации dnsmasq прописаны корректные адреса primary.broker.ecss и secondary.broker.ecss	Особенности установки системы в кластере		<p>Обратить внимание, что на обоих серверах содержимое одинаковое.</p> <pre>address=/ primary.broker.ecss/<Адрес ecss1> address=/ secondary.broker.ecss/<Адрес ecss2></pre>

Шаг	Пункт документации	Результат	Примечание
- адреса доступны по DNS-имени	DNS		<pre>ping -c1 cocon.mysql.ecss ping -c1 dialer.mysql.ecss ping -c1 statistics.mysql.ecss ping -c1 tc.mysql.ecss ping -c1 tts.mysql.ecss ping -c1 controlchannel.zmq.ecss ping -c1 system.restfs.ecss</pre>
4.3 На хостах ecss1 и ecss2 прописаны listen-интерфейсы для сервиса epmd	Настройка listen-интерфейса для сервиса epmd		<pre>sudo systemctl cat epmd.service</pre>
4.4 Сервер БД MySQL запущен и работает:			<pre>sudo systemctl status mysql.service</pre>
- к MySQL серверу возможно подключиться по DNS-имени			<pre>mysql -uroot -p -h ecss1</pre>
- при использовании резерва запущенна и корректно работает репликация БД	Схема развертывания MySQL master-master replication с использованием keepalive		<pre>mysql -uroot -p -e 'show slave status \G;'</pre> <p>На обоих серверах:</p> <pre>Slave_IO_Running: Yes Slave_SQL_Running: Yes</pre>
- порт mysql должен слушать на 0.0.0.0.			<pre>netstat -nl grep 3306</pre>
4.5 В системе с резервом прописано имя кластера (не undefined)	Особенности установки системы в кластере		<pre>cat /etc/ecss/ecss-mycelium/ mycelium1.config grep name</pre>

Шаг	Пункт документации	Результат	Примечание
<p>4.6 Сервисы ecss запущены и работают (находятся в состоянии running):</p> <ul style="list-style-type: none"> • ecss-mycelium • ecss-core • ecss-ds • ecss-pa-sip • ecss-pa-megaco • ecss-media-server • ecss-web-conf • ecss-restfs 	<p>Проверка статуса сервисов</p>		<p>Примеры:</p> <pre style="border: 1px solid #ccc; padding: 5px;">systemctl list-units --type service --all grep ecss</pre> <p>Состояние должно показывать "active" Или по каждому сервису отдельно:</p> <pre style="border: 1px solid #ccc; padding: 5px;">systemctl status <service name></pre> <p>Также и с ключом is-active:</p> <pre style="border: 1px solid #ccc; padding: 5px;">sasha@ecss1:~\$ systemctl is-active ecss-core ecss-ds ecss-pa-sip ecss-mycelium ecss-pa-megaco ecss-media-server ecss-web-conf ecss-restfs active active active active inactive active active active</pre>
<p>- Команда <code>cocon "system-status"</code> выдает пустой <code>Alarms list</code></p>			
<p>4.7 Медиа сервер MSR настроен и подключен к ECSS-10:</p>	<p>Настройка программного медиа сервера</p>		
<p>- конфигурация MSR настроена на подключение к ECSS-10</p>	<p>Настройка конфигурационного файла MSR</p>		<p>Из shell-серверов:</p> <pre style="border: 1px solid #ccc; padding: 5px;">cat /etc/ecss/ecss-media-server/config.xml</pre>

Шаг	Пункт документации	Результат	Примечание
- MSR задекларирован в системных медиаресурсах	Команды управления медиаресурсами		<p>CoCon:</p> <pre data-bbox="981 300 1506 472">/system/media/resource/list /system/media/registrar/info</pre> <p>Если система с резервированием, проверить, что подключен к обоим ядрам. Обратить внимание на правильные адреса ядер (не 0.0.0.0)</p>
- На хосте ecss2 аналогичным образом сконфигурирован MSR			При использовании резервирования
4.8 Restfs настроен и доступен с сервера ECSS-10:			
- RestFS доступен для записи и скачивания медиафайлов			<p>Из shell-серверов:</p> <pre data-bbox="981 909 1506 1122">wget http://ecss1:9990/system/sounds/ai_you.wav wget http://ecss2:9990/system/sounds/ai_you.wav</pre>
4.9 Сервис TTS (преобразование текста в голосовое сообщение) настроен и работает:			Если выбран tts при установке ecss-restfs
- Сервис возвращает голосовой файл после преобразования из текста			<p>Из shell-серверов:</p> <pre data-bbox="981 1339 1506 1823">wget http://ecss1:9990/generate? key=<Key>&text=<Text>&format=wav&lang=ru-RU&speaker=alyss&emotion=good&quality=hi wget http://ecss2:9990/generate? key=<Key>&text=<Text>&format=wav&lang=ru-RU&speaker=alyss&emotion=good&quality=hi</pre> <p>где: key=<Key> — ключ для доступа к серверу tts; text=<Text> — текстовое сообщение для преобразования в голосовое сообщение</p>

Шаг	Пункт документации	Результат	Примечание
4.10 Настроены пользователи, их права и роли	Управление пользователями		<pre>cocon/list users</pre>
4.11 Настроены правила ограничения обслуживания абонентов	Тип доступа, режим обслуживания, категория доступа и барринги		
5. Проверка корректности настройки домена в ECSS-10	Создание и настройка доменов		
5.1 В домене настроен SIP-транспорт (ip-set)	Работа SIP-адаптера с сетью		
- Адреса (<i>node_ip</i>) и порты (<i>listen_port</i>) выставлены в соответствии с проектом			<pre>/cluster/adapter/sip1/sip/network/info</pre>
- Адреса <i>node_ip</i> назначены на обеих нодах SIP-адаптера <i>ecss1</i> и <i>ecss2</i>			При использовании резервирования
- Адреса <i>node_ip</i> совпадают с адресами резервируемых интерфейсов, настроенных в <i>keepalived</i>			При использовании резервирования
- На домене выставлен <i>ipset</i>			<pre>/domain/<DOMAIN>/sip/network/info</pre>
5.2 В домен добавлены необходимые услуги:	Инсталляция и управление услугами		
- произведена установка в систему услуг	Инсталляция и управление услугами		<pre>cluster/storage/ds1/ss/install ds1@ecss1 *</pre>

Шаг	Пункт документации	Результат	Примечание
- домен добавлен в access-list к услугам	Инсталляция и управление услугами		<p>Просмотреть установленные в доменах услуги:</p> <pre>/cluster/storage/ds1/ss/ access-list show</pre> <p>Для добавления услуг на домене:</p> <pre>/cluster/storage/ds1/ss/ access-list add <DOMAIN> <SS></pre>
- настроена система сбора CDR (при необходимости)	Управление системой сбора CDR		
5.3 Произведена настройка контекста маршрутизации:	Виртуальная АТС. Маршрутизация телефонных вызовов		<pre>/domain/<DOMAIN>/routing/ list /domain/<DOMAIN>/routing/ show <CTX></pre>
- настроены нужные транки	Команды управления SIP-транками		<p>При необходимости.</p> <pre>/domain/<DOMAIN>/trunk/sip/ info /domain/<DOMAIN>/trunk/info</pre>
- настроены нужные бриджи	Команды управления bridge-интерфейсами		<p>При необходимости.</p> <pre>/bridge/info</pre>
- маршруты обрабатываются как в режиме набора enblock, так и overlap	Команды управления маршрутизацией		<pre>/domain/<DOMAIN>/routing/ trace</pre>
5.4 Произведены корректные настройки SIP-абонента(-ов):	Виртуальная АТС. Подключение и настройка SIP-абонентов		

Шаг	Пункт документации	Результат	Примечание
- абоненты созданы	Создание SIP-абонента		<pre>domain/<DOMAIN>/sip/user/ list</pre>
- установлены надежные логин/пароль SIP-авторизации			<pre>domain/<DOMAIN>/sip/user/ info * *</pre> <p>или отфильтровать (из shell):</p> <pre>ssh admin@ecss1 -p8023 '/ domain/<DOMAIN>/sip/user/ info * *' grep '(ds)'</pre>
- установлен trusted ip	Аутентификация абонентов		<p>В случае необходимости SIP-регистрации без авторизации</p> <pre>domain/<DOMAIN>/sip/user/ info</pre>
- у абонента установлен корректный контекст маршрутизации	Добавление абонента		<pre>domain/<DOMAIN>/sip/user/ info</pre> <p>или</p> <pre>cluster/storage/ds1/iface/ list <DOMAIN> * <GROUP> routing_context ~ ctx_from_local</pre> <p>(подставить нужный)</p>
- выставлен корректный тип терминала (basic/smart)	Настройка параметров абонента		<pre>/cluster/storage/ds1/iface/ list <DOMAIN> * <GROUP> terminal_type ~ smart</pre> <p>(или basic)</p>

Шаг	Пункт документации	Результат	Примечание
- у абонента активированы необходимые услуги	Инсталляция и управление услугами		<pre data-bbox="981 250 1506 353">/domain/<DOMAIN>/ss/info</pre>
5.5 Установка контекстов маршрутизации для системных интерфейсов:	Команды управления системными интерфейсами		<pre data-bbox="981 409 1506 580">/domain/<DOMAIN>/iface/user-set .system .system system:ivr routing.context</pre>
- настроен system:ivr	Команды управления системными интерфейсами		<p data-bbox="981 613 1506 741">При необходимости использования сервиса IVR. В большинстве случаев прописывается только контекст маршрутизации</p> <pre data-bbox="981 786 1506 956">/domain/<DOMAIN>/iface/info .system .system system:ivr</pre>
- настроен system:teleconference	Команды управления системными интерфейсами		<p data-bbox="981 985 1506 1113">При необходимости использования сервиса Teleconference. В большинстве случаев прописывается только контекст маршрутизации:</p> <pre data-bbox="981 1158 1506 1328">/domain/<DOMAIN>/iface/info .system .system system:teleconference</pre>