SP5000 ICP «Интеллектуальная облачная платформа»

Руководство по установке и настройке

АО «Искра Технологии», ноябрь 2022 г.

Документ выпущен компанией

АО «Искра Технологии»

РФ, 620066 Екатеринбург, ул. Комвузовская, 9-а

Т +7 343 210 69 51
Ф +7 343 341 52 40

РФ, 105264 Москва, ул. 9-я Парковая, 37

Т +7 495 727 08 50
Ф +7 495 727 08 78

iut@iskratechno.ru
www.iskratechno.ru

Оглавление

Список рисунков

Список таблиц

1 О документе

1.1 Назначение

Настоящий документ содержит инструкции по установке и настройке продукта «SP5000 ICP «Интеллектуальная облачная платформа» (далее – ICP, платформа ICP).

«SP5000 ICP «Интеллектуальная облачная платформа» предназначена для создания аппаратной и программной инфраструктуры, предоставляющей широкий набор средств оркестрации, администрирования, масштабирования, резервирования, мониторинга и информационной безопасности, необходимых для эффективного развертывания и надежного функционирования прикладных решений для заказчиков в сфере связи, безопасности и энергетики. Полная совместимость платформы с требованиями ETSI-NFV позволяет использовать ее для развертывания многокомпонентных комплексных телеком решений на базе архитектур NGN, vIMS и 5G.

1.2 Список терминов

Табл. 1-1: Список терминов и сокращений

ВМ	виртуальная машина
ГБ	гигабайт
ГиБ	гибибайт
ОЗУ	оперативное запоминающее устройство
ПО	программное обеспечение
ЦП	центральный процессор
ЦПУ	центральное процессорное устройство
ОС	операционная система
API	Application Programming Interface интерфейс прикладного программирования
APT	Advanced Package Tool пакетный менеджер APT
BASH	Bourne Again Shell усовершенствованная и модернизированная вариация командной оболочки
CIDR	Classless Inter-Domain Routing бесклассовая междоменная маршрутизация
CLI	Command line Interface интерфейс командной строки
COS	Cloud Operations Server операционный сервер COS
DB	DataBase база данных
DHCP	Dynamic Host Configuration Protocol протокол динамического конфигурирования сервера
DPDK	Data Plane Development Kit пакет разработки плоскости данных
DVR	Distributed Virtual Routing распределенная виртуальная маршрутизация
FAI	Fully Automatic Installation полностью автоматическая установка
FIO	Flexible I/O гибкий ввод-вывод
HA	High Availability высокая доступность
HCI	Hyper Converged Infrastructure гиперконвергентная инфраструктура

HDD	Hard Disk Drive жесткий диск
HTTP	Hypertext Transfer Protocol протокол передачи гипертекста
ICP	Intelligent Cloud Platform Интеллектуальная облачная платформа
IP	Internet Protocol Интернет-протокол
IPMI	Intelligent Platform Management Interface интерфейс управления интеллектуальной платформой
iLO	Integrated Lights-Out интегрированный процессор управления
IOMMU	Input-Output Memory Management Unit блок управления памятью (MMU) для операций ввода-вывода
IOPS	Input/Output operations per Second количество операций ввода-вывода в секунду
LACP	Link Aggregation Control Protocol открытый стандартный протокол агрегирования каналов
LVM	Logical Volume Manager менеджер логических томов
MAC	Media Access Control управление доступом к среде
MTU	Maximum Transmission Unit максимальная единица передачи
NFS	Network File System сетевая файловая система
NTS	Network Time Protocol протокол сетевого времени
OS	Operating System операционная система
OSD	Object Storage Daemon сущность, отвечающая за хранение данных
ovs	OpenVSwitch программный коммутатор, совместимый с протоколом OpenFlow
ovs-dpdk	DPDK enabled OpenVSwitch DPDK в виртуальном коммутаторе Open vSwitch
PCS	Pacemaker/Corosync Configuration System система конфигурации Pacemaker/Corosync
PG	Placement group группа размещения
PGP	Placement group for placement группа размещения для размещения
PXE	Preboot Execution Environment среда для загрузки компьютеров с помощью сетевой карты
RAID	Redundant Array of Independent Disks избыточный массив независимых дисков
RBD	RADOS Block Device Блочное устройство RADOS
ROC	Reference Openstack Cloud эталонное облако Openstack
SSD	Solid State Disk твердотельный диск
SSH	Secure Shell безопасная оболочка
SLA	Service Level Agreement соглашение об уровне обслуживания
SSL	Secure Sockets Layer уровень защищенных гнезд
SQL	Structured Query Language язык структурированных запросов
SR-IOV	Single Root Input/Output Virtualizaion виртуализация ввода-вывода с единым корнем
TFTP	Trivial File Transfer Protocol

	простой протокол передачи файлов
TLS	Transport Layer Security безопасность транспортного уровня
UEFI	Unified Extensible Firmware Interface унифицированный расширяемый интерфейс прошивки
URL	Uniform Resource Locator единый указатель ресурсов
UUID	Universally Unique Identifier универсальный уникальный идентификатор
vIMS	Virtual IP Multimedia System виртуальная мультимедийная IP-подсистема
VLAN	Virtual Local Area Network виртуальная локальная сеть
VIM	Virtualized Infrastructure Manager менеджер виртуализированной инфраструктуры
VIP	Virtual IP address виртуальный IP-адрес
VM	Virtual Machine виртуальная машина
VPNaaS	Virtual Private Network as a Service виртуальная частная сеть как услуга
VRRP	Virtual Router Redundancy Protocol протокол резервирования виртуального маршрутизатора
YAML	YAML Ain't Markup Language формат сериализации данных YAML

2 Введение

2.1 Базовая архитектура

Платформа ICP состоит из следующих основных компонентов:

- сервер репозиториев,
- сервер резервного копирования,
- сервер управления облаком,
- узлы ICP.

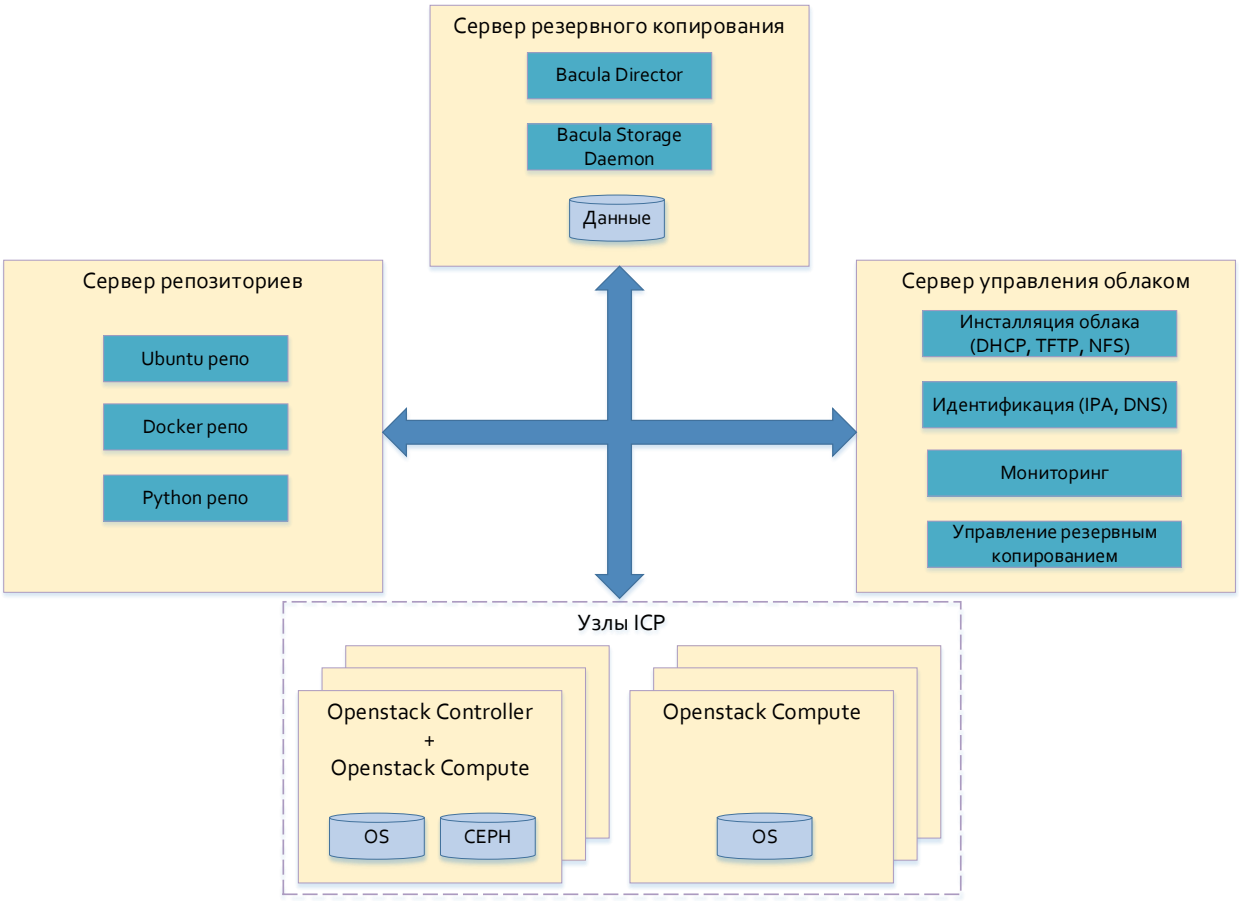


Рис. 2-1: Базовая архитектура

Сервер репозиториев — это центральное хранилище для:

- пакетов Ubuntu в репозитории apt;
- образов Docker в репозитории docker;
- пакетов Python в репозитории Python Package Index (PyPI);
- других файлов, которые можно получить по HTTP.

Сервер управления облаком запускает инструменты для:

- автоматической установки ОС Linux на узлы платформы;
- развертывания модулей Openstack на узлах платформы;
- реконфигурации облака;
- обновления модулей Openstack и другого ПО на узлах платформы;
- другие необязательные инструменты (клиент Openstack, Grafana, Bacula, инструменты для анализа журналов и т.д.).

Сервер резервного копирования выполняет следующие функции:

- реконфигурация облака;
- запуск Bacula Director, который контролирует процесс резервного копирования и восстановления;
- запуск Bacula Storage Daemon, который сохраняет файлы резервных копий на физический носитель резервных копий;
- обеспечение физических резервных носителей (диски, ленты и т.д.).

Сервер репозиториев, сервер управления облаком и сервер резервного копирования могут работать с несколькими облаками в мультиоблачной среде. Все эти службы могут работать на одном физическом узле, каждая на своей виртуальной машине.

Узлы ICP — это источники облачной вычислительной мощности и хранилища данных. В конфигурации без высокой доступности один из узлов используется как комбинация управляющего и вычислительного узла. Все остальные узлы используются только как вычислительные узлы.

В режиме высокой доступности три узла используются как комбинация управляющего и вычислительного узла. Все остальные узлы используются только как вычислительные узлы.

2.2 Сетевая архитектура

Сервер репозиториев подключен к сети с маршрутизацией и должен быть доступен по сети L3 с сервера управления облаком и узлов ICP.

Сервер резервного копирования подключен к сети с маршрутизацией и должен быть доступен по сети L3 с сервера управления облаком.

Сервер управления облаком имеет два интерфейса: один для подключения к серверу репозиториев (восходящий интерфейс), другой для подключения к узлам ICP через сеть INTERNAL_API (интерфейс развертывания). Сервер управления обалком и узлы ICP должны находиться в одном широковещательном домене.

Узлы ICP связаны с внешним миром и между собой через 6 виртуальных сетей:

1. INTERNAL_API

   Используется службами OpenStack для связи друг с другом и с базами данных. В этой сети существуют определенные риски в отношении безопасности, поэтому рекомендуется сделать эту сеть внутренней, недоступной извне.

2. EXTERNAL_API

   Используется для доступа извне к внешнему API-интерфейсу Openstack.

3. PROVIDER

   Используется компонентом Neutron для сетей провайдеров (однородные и тегированные сети VLAN).

4. TUNNEL

   Используется компонентом Neutron для обеспечения трафика от ВМ до ВМ по туннелированным сетям (VxLan).

5. STORAGE

   Используется для репликации данных компонентом Ceph. Эту сеть можно активно использовать в сочетании с управляемой, высокоскоростной коммуникационной системой.

6. CLUSTER

   Используется виртуальными машинами для связи с Ceph. Эта сеть также может быть интенсивно использоваться в сочетании с управляемой, высокоскоростной коммуникационной системой.

7. IPMI

   Используется для выполнения различных управляющих действий над сервером, таких как запуск, перезагрузка и т.д., из удаленной локации. Через эту сеть можно получить информацию о сети и хранилище, к которым относится тот или иной узел.

IPMI — это специальная нетегированная сеть, используемая только для задач управления.

Сети STORAGE и CLUSTER обычно объединяют в одну сеть.

Сеть INTERNAL_API – нетегированная. Все остальные сети могут быть тегированы и подключены к одному и тому же сетевому интерфейсу узла ICP.

Узел ICP может содержать от двух и более сетевых интерфейсов, объединенных в один связующий интерфейс для повышения надежности и/или увеличения пропускной способности сети.

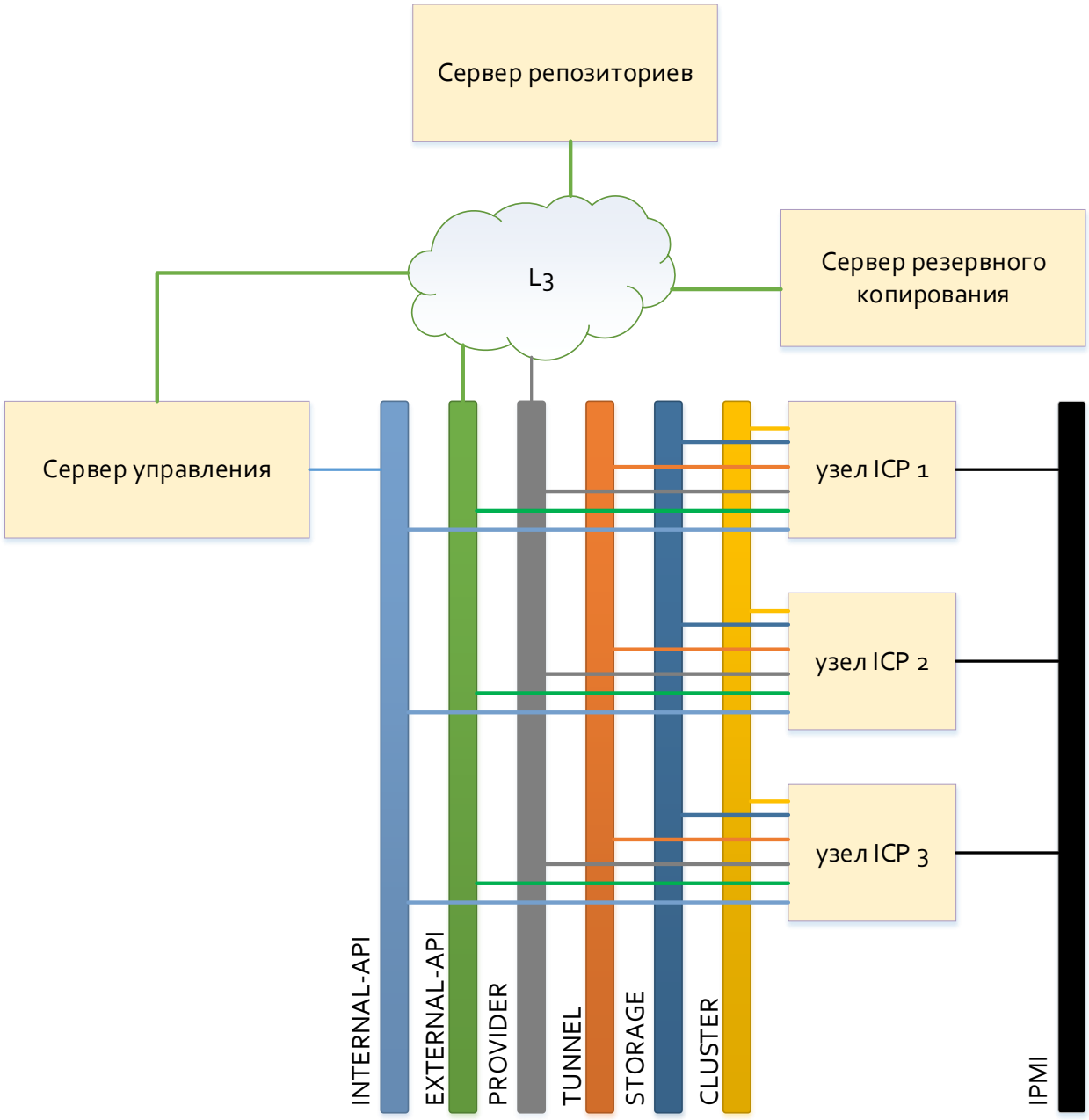


Рис. 2-2: Сетевая архитектура

2.3 Архитектура хранилища

Хранилище может быть сконфигурировано многими различными способами, с одним диском на узел или несколькими дисками на узел, с твердотельными накопителями и/или жесткими дисками, с RAID или без него, с системой хранения Ceph или без нее.

В этом документе описывается стандартная рекомендуемая конфигурация.

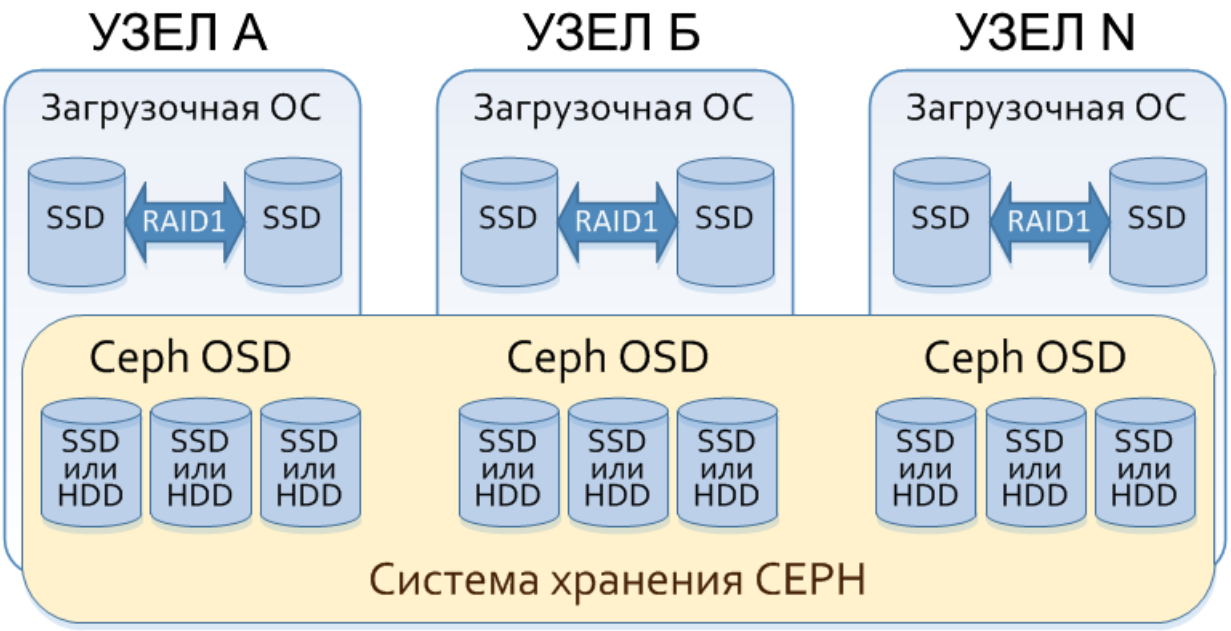


Рис. 2-3: Хранилище на платформе ICP

Разделы ОС:

- Разделы операционной системы (загрузочный, корневой, SWAP, /var/log и /var/lib) создаются на первом диске, обнаруженном в процессе установки. Рекомендуется, чтобы этот диск был самым быстрым диском в системе, то есть диском SSD.
- Рекомендуется зеркалировать диск ОС с помощью аппаратного механизма RAID.
- Загрузочный раздел может быть указан только как стандартный раздел, остальные разделы могут быть либо стандартными, либо lvm.
- Раздел БД Ceph также создается на первом диске, обнаруженном в процессе установки. Этот раздел используется компонентом Ceph для кэширования данных.

Сущности Ceph OSD:

- Каждый узел может использовать один или несколько дисков для Ceph. Каждый диск представляет собой одну сущность Ceph OSD.
- Обычно используются жесткие диски, но можно использовать и твердотельные накопители, если на ICP будут работать приложения с более высокими требованиями.

2.4 Архитектура мониторинга

Подсистема мониторинга состоит из нескольких модулей, которые работают на узлах ICP и сервере управления облаком:

Экспортеры Prometheus:

- Экспортеры собирают данные от отдельных сервисов.
- У каждого сервиса есть свой экспортер.
- Каждый экспортер выдает текущий снимок метрик при опросе сервером Prometheus.
- Сам экспортер не хранит историю метрик.
- Каждый экспортер работает в своем собственном контейнере, за исключением экспортера Ceph, который включен в компонент мониторинга Ceph.
- Экспортеры, работающие на управляющих узлах:
    - Экспортер Node (экспортер метрик оборудования и операционной системы).
    - Экспортер Blackbox (проверяет доступность и время отклика конечных точек).
    - Экспортер Openstack (собирает данные о ресурсах Openstack).
    - Экспортер Cadvisor (Docker) (предоставляет статистику контейнеров).
    - Экспортер Mysqld (предоставляет метрики сервера MySQL).
    - Экспортер HaProxy (извлекает статистику HAProxy).
    - Экспортер Ceph (предоставляет метрики Ceph).
- Экспортеры, работающие на вычислительных узлах:
    - Экспортер Node (экспортер метрик оборудования и операционной системы).
    - Экспортер Cadvisor (Docker) (предоставляет статистику контейнеров).

Сервер Prometheus:

- извлекает данные от экспортеров по HTTP,
- сохраняет данные в базе данных временных рядов,
- содержит инструменты для фильтрации и визуализации данных,
- отправляет оповещения (аварийные сигналы) в менеджер аварийных сигналов,
- запускается в контейнере на управляющих узлах.

Менеджер аварийных сигналов (Alert Manager):

- обрабатывает оповещения, отправленные сервером Prometheus;
- обрабатывает дедупликацию, группировку и маршрутизацию этих оповещений к правильному получателю.

Компонент Grafana:

- представляет собой веб-приложение,
- извлекает данные с сервера Prometheus,
- отображает метрики и оповещения в графическом виде.

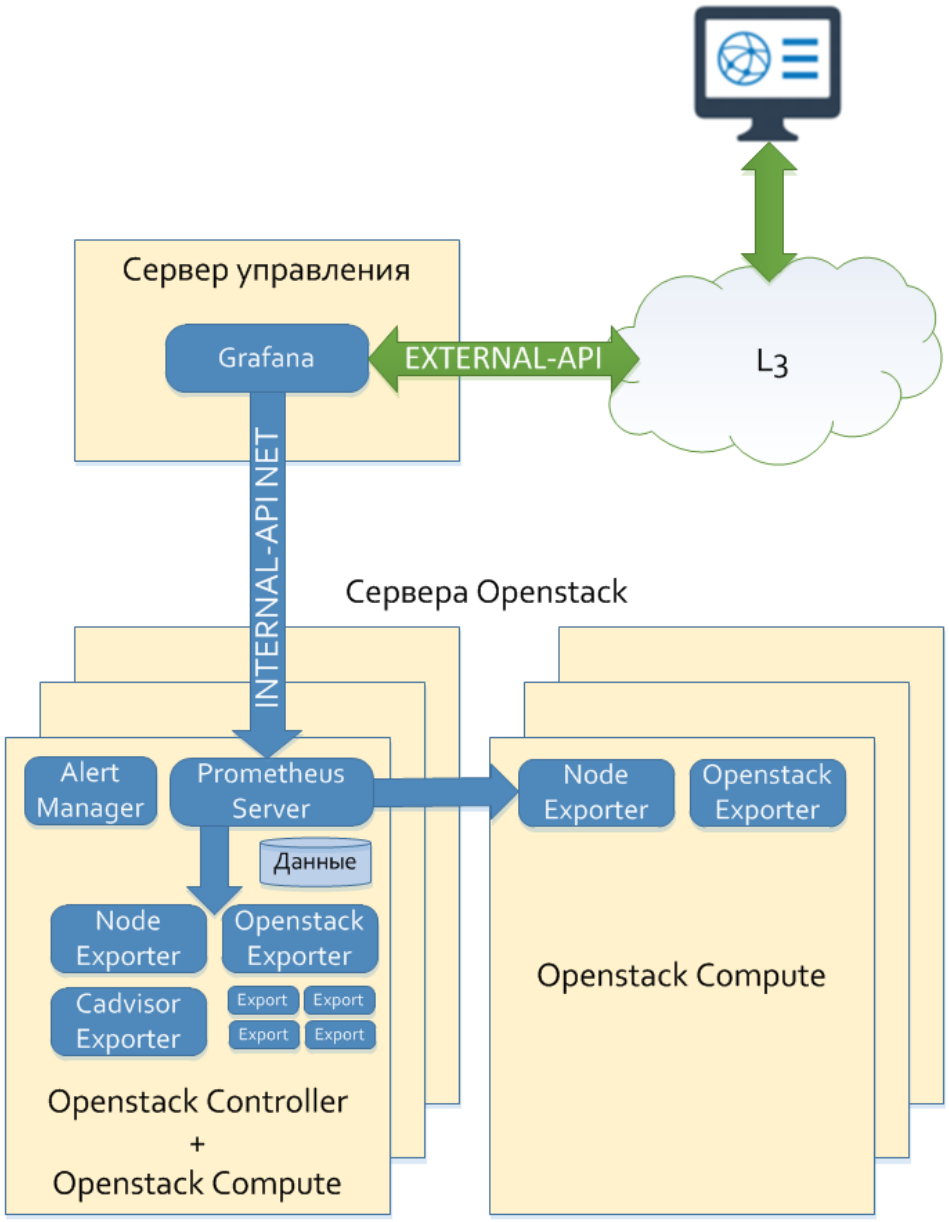


Рис. 2-4: Архитектура мониторинга

2.5 Архитектура службы идентификации

FreeIPA — это бесплатная система управления идентификацией с открытым исходным кодом. На платформе ICP она используется для обеспечения механизма единого входа в систему (SSO), который позволяет пользователю входить через единый идентификатор в хосты сервера COS и платформы ICP (физические и виртуальные машины), а также на саму платформу ICP. FreeIPA также служит сервером DNS для всей платформы ICP (в том числе в конфигурации с несколькими облаками).

2.5.1 Единый вход в систему (SSO)

Клиентами IPA платформы ICP, которые автоматически добавляются на сервер IPA, являются:

- операционный сервер COS,
- виртуальная машина сервера IPA,
- все остальные виртуальные машины сервера COS,
- все хосты платформы ICP.

Другие хосты (физические машины и/или виртуальные машины) могут быть добавлены в качестве клиентов на сервер IPA с помощью процедур в графическом интерфесе или интерфейсе командной строки.

Компонент Keystone также автоматически интегрируется в систему IPA. При этом в облаке создается новый домен, а также проект в этом домене. Группы и пользователи IPA становятся членами этого домена. Также устанавливаются роли для групп внутри этого домена.

Группы пользователей и пользователи системы IPA настраиваются в файле конфигурации сервера COS (cos_config.yml). Имя домена, имя проекта и роли групп пользователей в этом домене настраиваются в основном файле конфигурации облака (big yaml).

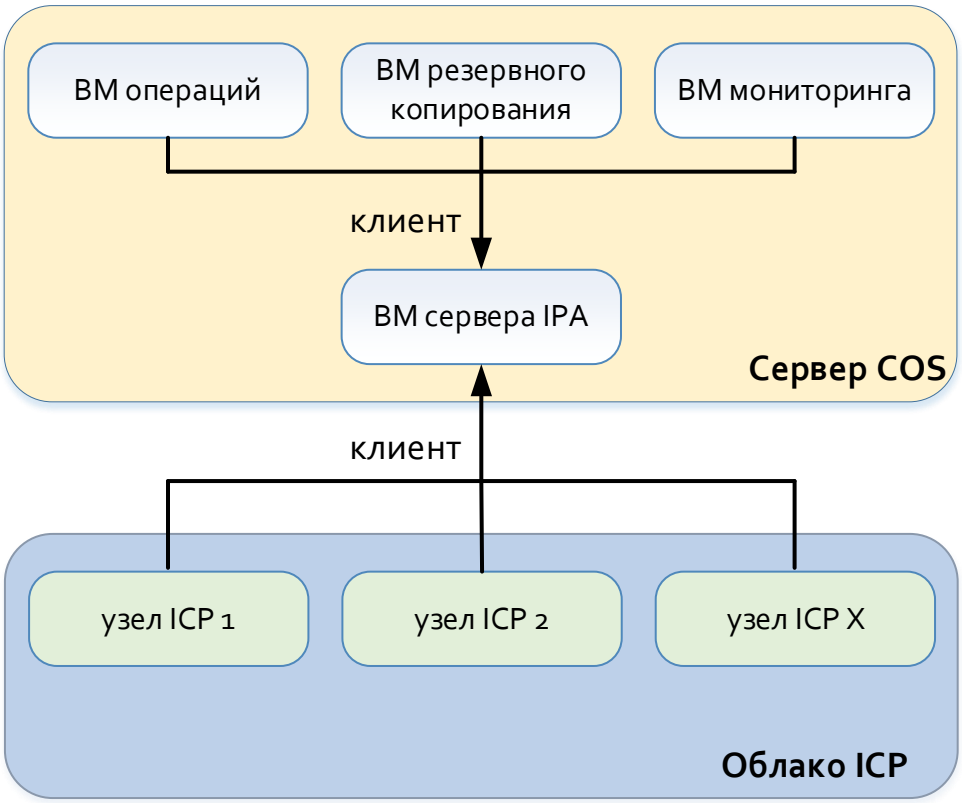


Рис. 2-5: Единый вход в систему (SSO)

2.5.2 Архитектура сервера DNS

Для решения ICP требуется сервер DNS, который является уполномоченным для поддоменов сервера COS и облаков ICP. Для этого используется сервер DNS, входящий в состав сервера IPA. Сетевые имена хостов ICP и их IP-адреса автоматически вставляются в базу данных DNS системы IPA. Туда же вставляется имя облака и его общедоступный IP-адрес. Другие имена (физических машин и/или виртуальных машин) можно добавить с помощью процедур в графическом интерфейсе или интерфейсе командной строки.

Сервер DNS системы IPA может быть уполномоченным для всего домена (см. Рис. 2-6) или только для поддомена (см. Рис. 2-7). В последнем случае делегирование поддомена в сервер DNS системы IPA должно быть настроено на сервере DNS заказчика, который является уполномоченным для домена верхнего уровня.

В конфигурации с несколькими облаками все облака могут быть принадлежать к одному и тому же домену (одноуровневая конфигурация) или каждое облако может иметь собственный поддомен (см. Рис. 2-8).

2.5.2.1 Сервер IPA DNS, уполномоченный для всего домена

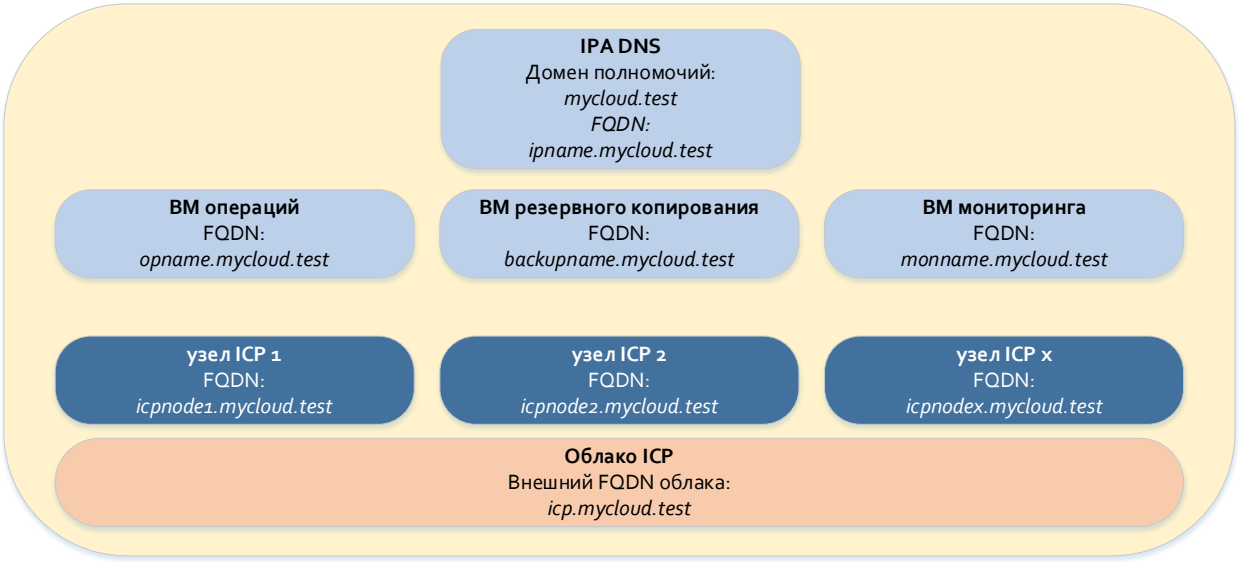


Рис. 2-6: Сервер IPA DNS, уполномоченный для всего домена

2.5.2.2 Сервер IPA DNS, уполномоченный для поддомена

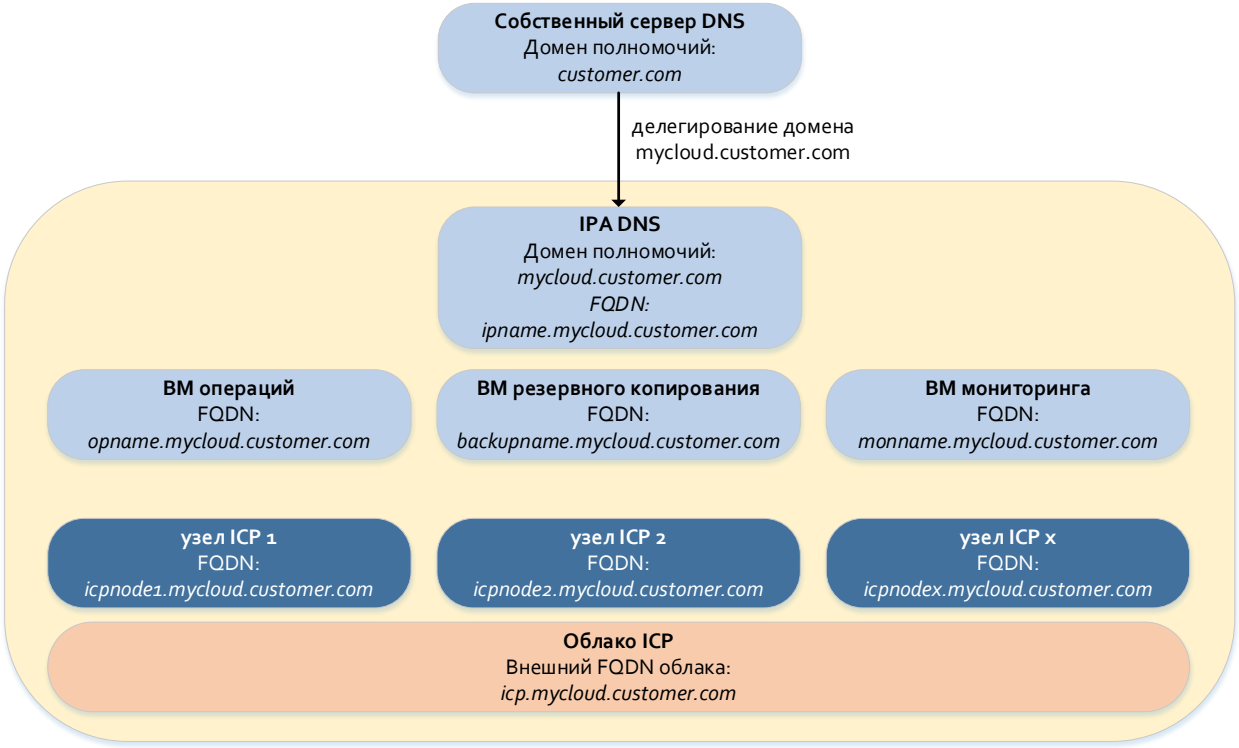


Рис. 2-7: Сервер IPA DNS, уполномоченный для поддомена

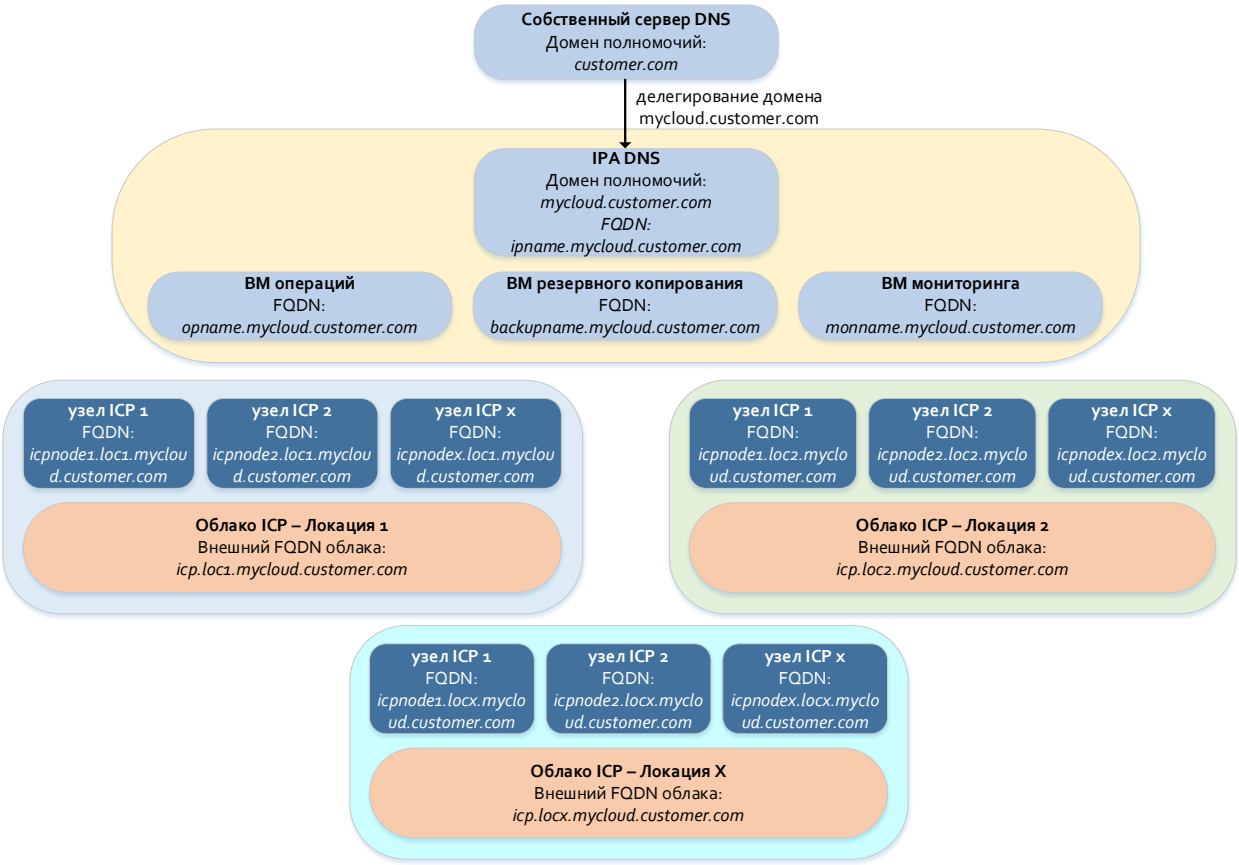


Рис. 2-8: Сервер IPA DNS, уполномоченный для поддомена (конфигурация с несколькими облаками)

3 Системные требования к платформе ICP

3.1 Требования к ВМ операций

Табл. 3-1: Требования к ВМ операций PRIMARY-OPVM

Характеристика	Значение
ЦП, ядер	1
ОЗУ, ГБ	3
Место на диске, ГБ	50

3.2 Требования к серверу COS

Табл. 3-2: Требования к операционному серверу COS

Характеристика	Значение
ЦП, ядер	10
ОЗУ, ГБ	64
Место на диске, ГБ	200
Интерфейсы Ethernet, шт.	2

3.3 Требования к серверу вычислительного узла

Табл. 3-3: Требования к серверу вычислительного узла

Характеристика	Значение
ЦП, ядер	16
ОЗУ, ГБ	64
Диски	◆ 1 диск для ОС на 240 ГБ ◆ 2 диска для компонента Ceph на 500ГБ
Интерфейсы Ethernet, шт.	2

4 Операционный сервер COS

Операционный сервер COS (Cloud Operation Server), является центральной точкой, где выполняются следующие операции, связанные с облаками платформы ICP:

- автоконфигурирование ОС и развертывание облака ICP,
- мониторинг и анализ журналов дочерних облаков,
- резервное копирование облаков и виртуальных машин,
- управление идентификацией и сервером DNS.

Это также центральное место хранения пакетов, образов docker и других файлов.

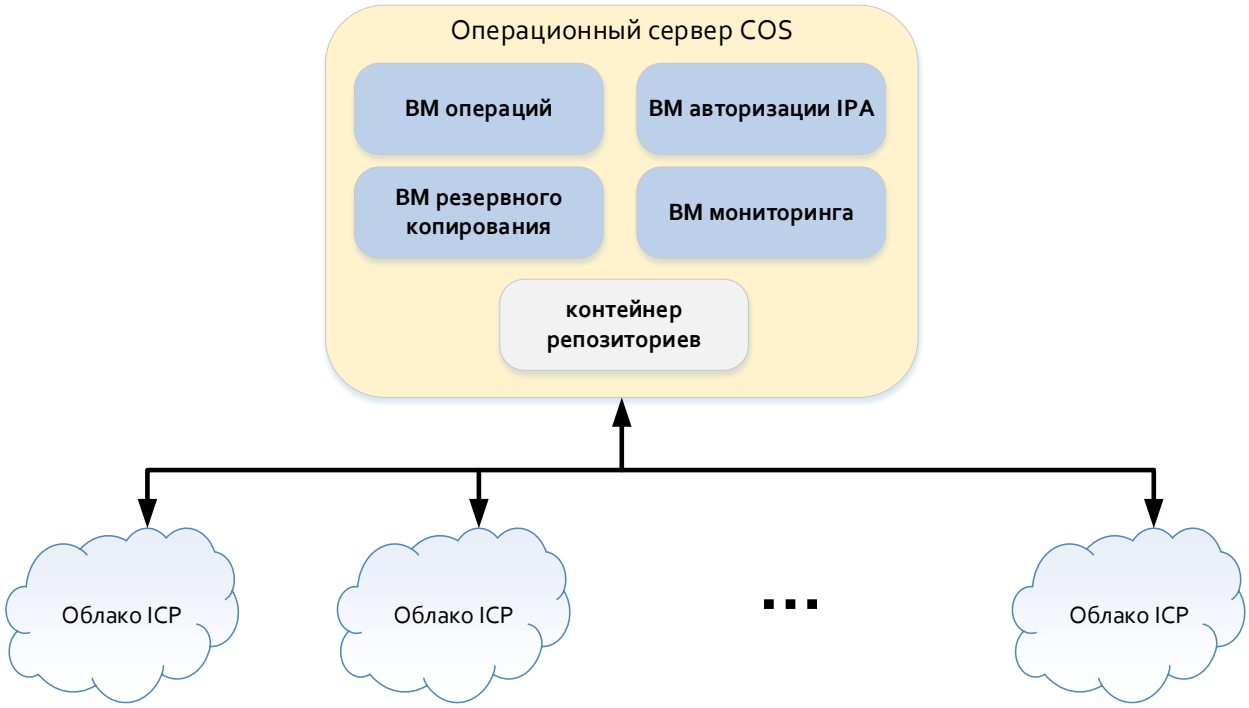


Рис. 4-1: Базовая архитектура сервера COS

Различные облачные операции выполняются на отдельных виртуальных машинах:

- ВМ операций: автоконфигурирование ОС и развертывание облака ICP, мониторинг и анализ журналов.
- ВМ авторизации IPA: управление идентификацией, сервер доменных имен.
- Сервер резервного копирования: запуск модулей Bacula Director и Bacula Storage Daemon.

Автономный сервер Nexus как сервер репозиториев. Он запускается в контейнере непосредственно на хосте сервера COS (не на виртуальной машине). Он доступен по IP-адресу или сетевому имени основного хоста.

4.1 Развертывание операционной системы на сервере COS

Операционная система, используемая на сервере COS — Ubuntu20.04. Она развертывается с помощью инструментов FAI.

Предварительные условия:

- ВМ (основная ВМ операций, PRIMARY-OPVM), которая может работать на ПК или другом сервере в дата-центре.

- Сетевое соединение L2 между сервером, на котором запущена ВМ, и сервером COS (сеть развертывания).
- Подключение к общедоступной или корпоративной сети (восходящая сеть).
- Образ ВМ операций для создания ВМ (qcow2 или mvdk).
- Архив tar репозитория автономного сервера Nexus.

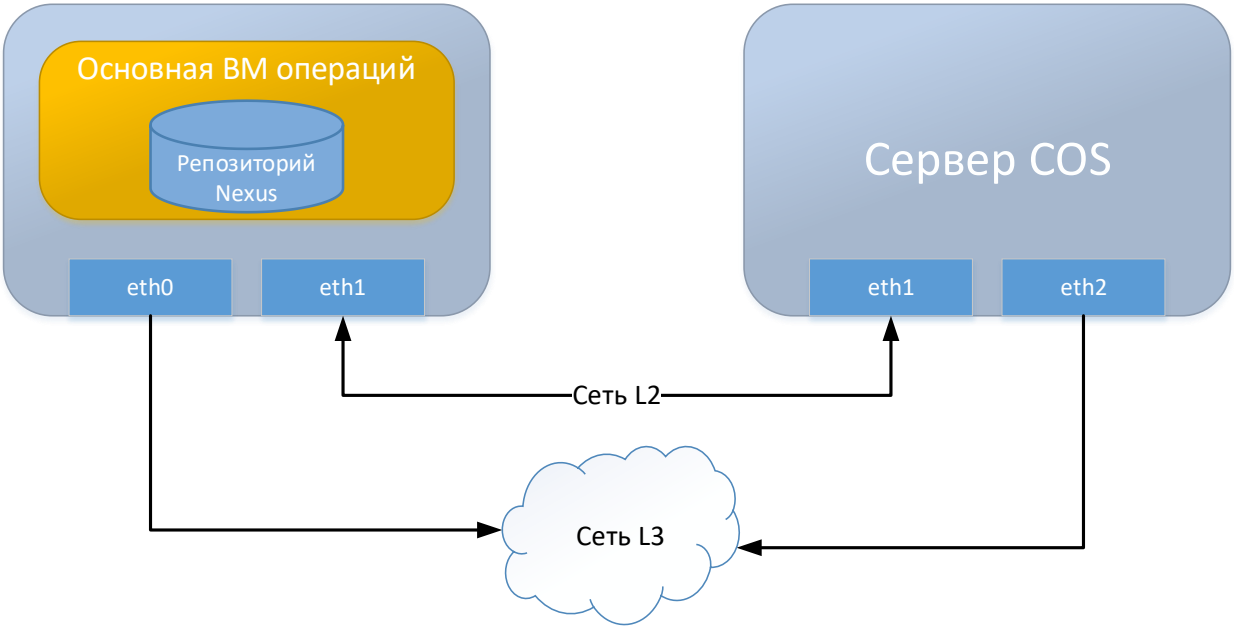


Рис. 4-2: Конфигурация для развертывания ОС на сервере COS

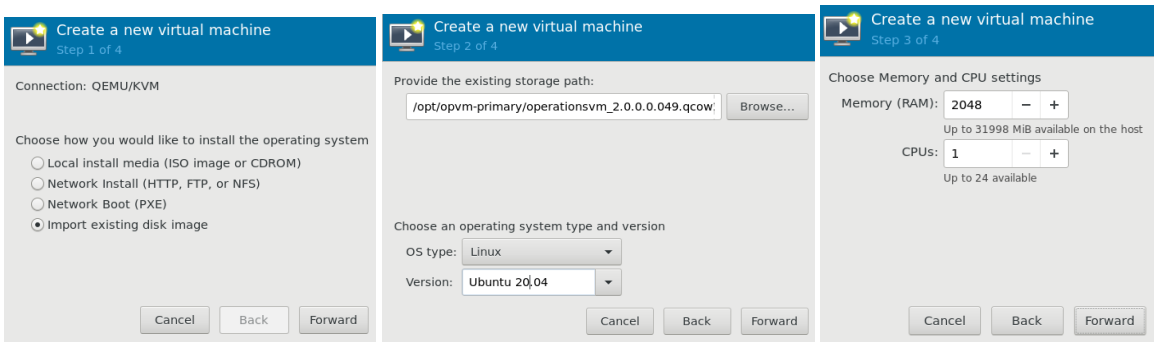
Продцедура развертывания:

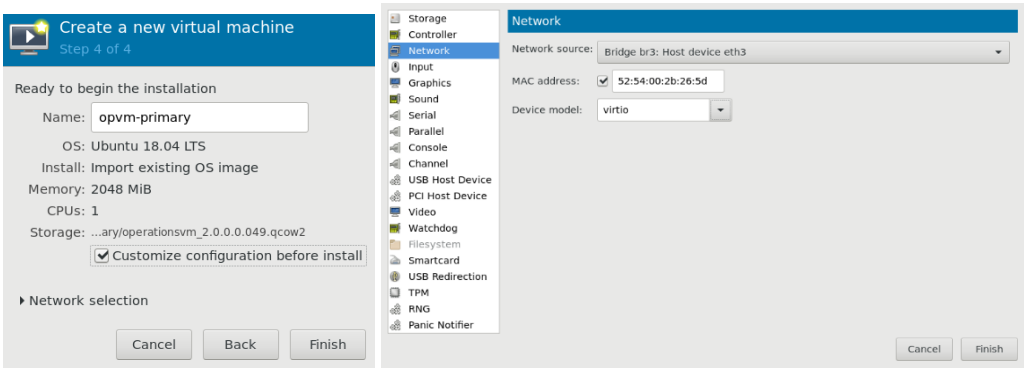
1. Создание виртуальной машины PRIMARY-OPVM.

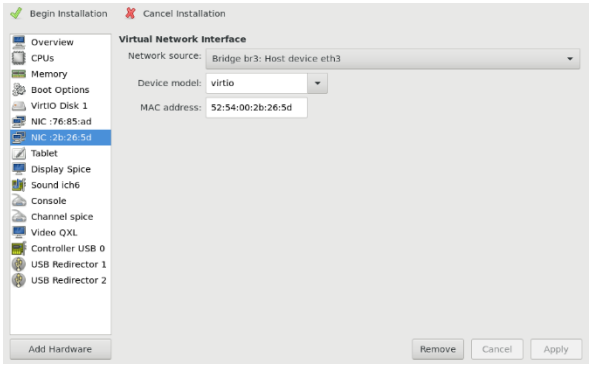
Хост LINUX:

Если используется хост Linux, используйте виртуализацию KVM.

- Загрузите образ *operationsvm_<package_version>.qcow2*
- Создайте ВМ из этого образа. ВМ должна иметь два сетевых интерфейса. Первый интерфейс подключен к внешней сети, второй может быть подключен к хосту COS напрямую или через сеть развертывания L2.
  Самый простой способ создать ВМ — с помощью инструмента virt-manager:

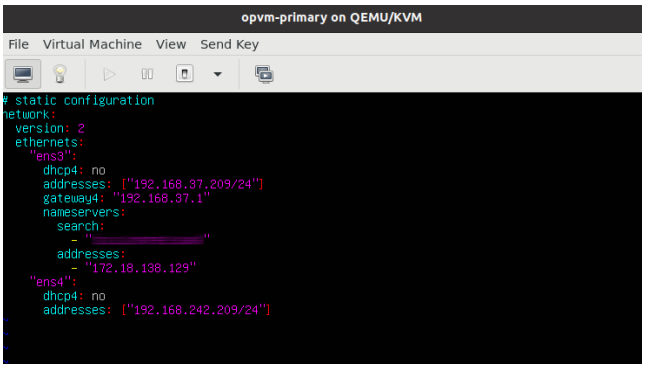






- Настройте сеть на виртуальной машине PRIMARY-OPVM с помощью консоли BM (имя пользователя: root, пароль предоставляется системным администратором).

```
vi /etc/netplan/50-cloud-init.yml
```



- Примените конфигурацию сети:
  `netplan apply`

- Проверьте, работает ли сеть.

- Установите сетевое имя:
  `hostname <primary_opvm_hostname>`
  `vi /etc/hostname # write primary_opvm_hostname to this file`

4.2 Развертывание сервисов ICP на сервере COS

4.2.1 Настройка делегирования субдомена ICP в IPA DNS

Если сервер IPA DNS является уполномоченным для поддомена (см. раздел 2.5.2.2), делегирование этого поддомена в IPA DNS должно быть настроено на сервере DNS заказчика, который является уполномоченным для домена верхнего уровня.

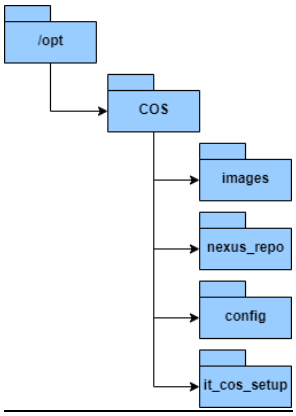
В следующем примере показано, как добавить запись переадресации зоны для поддомена IPA DNS в записи зоны верхнего уровня:

Табл. 4-1: Пример добавления поддомена IPA DNS на сервер DNS зоны верхнего уровня с помощью nsupdate

```
#!/bin/bash

# RUN ON DNS SERVER (ZONE FORWARDER) FOR EACH SUB-DNS SERVER (IPA MASTER AND REPLICAS)

# add_sub_zone (sub-dns hostname) (sub-dns-domain) (sub-dns ip address)
functon add_sub_zone()
{
   echo -e "server ${server}\nupdate add ${1}.${2}. ${ttl} IN A ${3}\nupdate add ${2}. ${ttl} IN NS ${1}.${2}.\nsend" |
nsupdate
}

# EXAMPLE FOR ipamaster WITH 2 REPLICAS IN SUB-DOMAIN 'edge.iskrauraltel.ru'

ttl=68400          # TIME-TO-LIVE IN SECONDS
server=localhost    # SUB-DOMAIN FORWARDER SERVER (master for 'iskrauraltel.ru')

add_sub_zone ipamaster edge.iskrauraltel.ru 172.19.192.1
add_sub_zone ipareplica1 edge.iskrauraltel.ru 172.19.200.1
add_sub_zone ipareplica2 edge.iskrauraltel.ru 172.19.212.1
```

4.2.2 Подготовка хоста COS

Подготовьте структуру папок:




```
cd /opt
mkdir COS
cd COS
mkdir images
mkdir config
mkdir nexus_repo
mkdir it_cos_setup
```

Предупреждение!

Структура папок может находиться в любой другой папке, *отличной от /opt/COS*, но примите это во внимание при выполнении команд, описанных ниже.

Загрузите:

- архив данных Nexus в папку */opt/COS;*
- файл версий модулей в папку */opt/COS/config;*
- образ Ubuntu в папку */opt/COS/images.*

Разархивируйте архив данных Nexus в папку *nexus_repo*:

```
cd /opt/COS
tar -xzf <nexus_archive>.tar.gz –C nexus_repo
```

Настройте модули yum, pip и docker

```
cd /opt/COS/nexus_repo
./prepare-cos-host.sh configure_apt_repos     #apt and pip repo links are set to localhost
./prepare-cos-host.sh configure_docker    #bridge network is disabled
```

Запустите и включите модуль docker:

```
systemctl start docker
systemctl enable docker
```

Запустите автономный репозиторий Nexus:

```
cd /opt/COS/nexus_repo
./start-nexus.sh start
```

Предупреждение!

Скрипт «start-nexus.sh» может возвращать ошибку ERROR: Pool overlaps with other one on this address space («ОШИБКА: пул перекрывается с другим пулом в этом адресном пространстве»), если хост использует ту же конфигурацию моста docker, что используется в docker-compose. В этом случае нужно изменить некоторые переменные в файле с расширением *.env.

Если требуются другие настройки, отредактируйте переменные в файле с расширением *.env:

```
Vi /opt/COS/nexus_repo/.env
```

Чтобы изменить сетевые адреса мостов docker для Nexus, установите значения переменных: `"IPV4_ADDRESS_NEXUS", "IPV4_ADDRESS_NGNX", "OFFLINENET_SUBNET", "OFFLINENET_GATEWAY".`

Чтобы изменить образ Nexus, установите значение переменной: `NEXUS_IMAGE.`

Переменные `LOCAL_REGISTRY` используются только для целей тестирования.

Установите те же дополнительные пакеты на хост COS:

```
yum clean all
yum install ansible
pip install -U pyyaml
```

Загрузите пакет it_cos_package и разархивируйте его в папку *it_cos_setup*:

```
cd /opt/COS
wget http://localhost:8081/repository/itbuild-raw/AI6212AX/it_cos_setup/it_cos_setup-<version>.tar.gz
tar -xzf it_cos_setup-<version>.tar.gz –C it_cos_setup
```

4.2.3 Подготовка файла конфигурации для виртуальных машин на хосте COS

Файл конфигурации *cos_config.yml* должен быть подготовлен в папке */opt/COS/config*. В качестве основы используйте конфигурационный файл шаблона, который находится в пакете *it_cos_setup*:

```
cd /opt/COS/config
cp ../it_cos_setup/config/cos_config.yml.offnexus cos_config.yml
vi cos_config.yml
```

Файл конфигурации COS *cos_config.yml*:

```
---
############################
# COMMON DATA
############################
timezone: "Asia/Yekaterinburg"  # timezone
domain_name: "mydomain.test"    # domain name of COS server and the cloud

# host's data
public_net_bridge: 'br0'   # COS server bridge connected to public network
icp_internal_net_bridge: 'br1'  # COS server bridge connected to internal cloud network

# ICP version's file
icp_versions_file: "/root/COS/config/README.txt"    # path to package verision's file

# repository server data
offline_nexus: 'yes'     # 'yes' – offline nexus repo; 'no' – central nexus repo

repo_server:
  url: "http://cumulus.csi.iskrauraltel.ru:8081"  # URL of Nexus repo = http(s)://COS_SERVER_NAME_OR_IP:8081
                                                  # use name only when server name is resolvable, otherwise use IP
docker_registry:
  url: "http://cumulus.csi.iskrauraltel.ru:5001"  # URL of Nexus docker registry = http(s)://COS_SERVER_NAME_OR_IP:5001
                                                  # use name only when server name is resolvable, otherwise use IP
  user: "admin"                    # user on docker registry
  password: "[пароль_предоставляется_системным_администратором]"              # password of the user

# OS images
ubuntu_image: "/root/COS/images/ubuntu-18.04-server-cloudimg-1.0.0-amd64.qcow2"  # path to Ubuntu image

# set COS server as an IPA client
cos_is_ipa_client: 'yes'

############################
# COS modules
############################

#########
# IPA
#########
# IPA server params
ipa_admin_password: [пароль_предоставляется_системным_администратором]'  # IPA admin password
```

```
ipa_dm_password: '[пароль_предоставляется_системным_администратором]'      # IPA directory manager password
dnssec_disable: 'yes'                # dnssec disable - default is yes

# IPA default groups and users
freeipa_groups:          # default groups added to IPA server
  osadmins:            # openstack admins group
    desc: "Openstack admin group"   # description of osadmins group
  osmembers:           # openstack members group
    desc: "Openstack member group"   # description of osmembers group

freeipa_users:             # default users added to IPA server
  keyadmin:                # user needed for IPA - Openstack integration
    first: "Keystone"      # firstname of keyadmin user
    last: "Backend"        # lastname of keyadmin user
    password: "[пароль_предоставляется_системным_администратором]"  # password of keyadmin user
  osadmin:                 # openstack admin user
    first: "OsAdmin"       # firstname of osadmin user
    last: "Admin"          # lastname of osadmin user
    password: "[пароль_предоставляется_системным_администратором]"  # password of osadmin user
    group:                 # groups to which osadmin user belongs
     - osadmins
  osmember:                # openstack member user
    first: "OsMember"      # firstname of osmember user
    last: "Member"         # lastname of osmember user
    password: "[пароль_предоставляется_системным_администратором]"   # password of osmember user
    group:                 # groups to which osmember user belongs
     - osmembers

# IPA vm params
ipa_vm:                          # configuration data of ipa vm
  included: 'yes'                # enabled or disabled
  name: 'ipasrv'                 # name of IPA vm
  vm_info:
    net_interfaces:
      public:
        address_cidr: '192.168.37.202/24'   # public IP address of IPA server
        gateway: '192.168.37.1'              # default gateway on public net
        search_domains:                      # additional search domains
          - 'iskrauraltel.ru'
        nameservers:                         # global forwarder nameserver for IPA DNS
          - '10.1.2.3'

###############
# OPERATIONS VM
###############
# operations vm params
operations_vm:                  # configuration data of operations vm
  included: 'yes'               # enabled or disabled
  name: 'operationsvm'          # name of operations vm
  ipa_client: 'yes'              # VM is automatically inserted to IPA
  vm_info:
    net_interfaces:
      public:
        address_cidr: '192.168.37.203/24'   # public IP address of operations server
        gateway: '192.168.37.1'              # default gateway on public net
```

```
        search_domains:                    # additional search domains
          - 'iskrauraltel.ru'
      internal:
        address_cidr: '192.168.110.5/24'  # internal IP address of operatios server
  docker:
    docker_bip: "192.0.2.129/25"       # docker bip
  openstack_client: 'yes'     #if 'yes' it_openstack-client container is automatically started (client is needed if Openstack will
                              #be integrated to IPA)
  openstack_rally: 'no'       #if 'yes' it_openstack-rally container is automatically started
  openstack_shaker: 'no'    #if 'yes' it_openstack-shaker container is automatically started
  bacula_client: 'no'         #if 'yes' bacula client is installed and run

#########
# BACKUP
#########
# backup vm params
operations_vm:                  # configuration data of operations vm
  included: 'yes'               # enabled or disabled
  name: 'buckupsrv'             # name of backup vm
  ipa_client: 'yes'              # VM is automatically inserted to IPA
  vm_info:
    net_interfaces:
      public:
        address_cidr: '192.168.37.204/24'  # public IP address of operations server
        gateway: '192.168.37.1'            # default gateway on public net
        search_domains:                    # additional search domains
          - 'iskrauraltel.ru'
  docker:
    docker_bip: "192.0.2.129/25"       # docker bip
  nfs_server: 'yes'

##############
# MONITORING
##############
# Monitoring vm params
mon_vm:
 included: 'yes'
 name: 'monitoring'
 ipa_client: 'yes'
 vm_info:
  net_interfaces:
   public:
    address_cidr: '192.168.26.57/24'
    gateway: '192.168.26.1'
   internal:
    address_cidr: '192.168.110.6/24'
 docker:
  docker_bip: "192.0.2.129/25"
 grafana:
  cert: 'selfsigned'
 efk:
  cert: 'selfsigned'

############################
# ICP CLOUDS
############################
icp_clouds:                                 # clouds controlled by COS server
 vmcloud1:                                  # name of cloud
```

```
  ext_url: "https://vmcloud.mydomain.test"    # public URL of cloud
  ext_float_ip: "192.168.100.19"              # external float IP address
  int_float_ip: "192.168.110.19"              # internal float IP address
  monitoring: 'yes'                           # automatically included to monitoring system
  ipa_integration: 'yes'                      # openstack cloud is automatically integrated to IPA
  add_cloud_name_to_dns: 'yes'                # cloud name is added to IPA DNS
```

Службы мониторинга могут работать на ВМ операций, а не на ВМ мониторинга. Ее можно настроить в cos_config.yml. Просто удалите параметр mon_vm и добавьте определение grafana и efk к параметру operations_vm, например:

```
operations_vm:                # configuration data of operations vm
  grafana:
    cert: 'selfsigned'  # can be "none", "selfsigned", "ipaprovided" or "path_to_certificate_folder"
  efk:
    cert: 'selfsigned'  # can be "none", "selfsigned", "ipaprovided" or "path_to_certificate_folder"
```

Предупреждение!

Если ВМ – клиент IPA (параметр *ipa_client* имеет значение «*yes*»), серверы имен не требуется настраивать на общедоступном сетевом интерфейсе. IPA DNS будет добавлен автоматически.

Значения использования ресурсов по умолчанию для каждой ВМ:

ВМ авторизации IPA:
- memory: 4096
- vcpu: 1
- boot_volume: '70G'

ВМ операций:
- memory: 4096
- vcpu: 1
- boot_volume: '150G'

ВМ мониторинга:
- memory: 4096
- vcpu: 1
- boot_volume: '50G'

ВМ резервного копирования:
- memory: 2048
- vcpu: 1
- boot_volume: '200G'

Эти значения можно изменить в конфигурационном файле cos_config.yml. Просто добавьте параметр memory, vcpu и/или boot_volume к определению ВМ в разделе vm_info, например:

```
operations_vm:
  vm_info:
    memory: 2048
    vcpu: 1
    boot_volume: '200G'
```

4.2.4 Развертывание пакетов и настройка модулей на хосте COS

Перед этим действием должны быть определены параметры в разделе *COMMON DATA* в файле конфигурации COS.

Выполните команды:

```
cd /opt/COS/it_cos_setup/ansible
export PATH=$PATH:../tools
cos-host.sh deploy
```

(если скрипт cos-host.sh недоступен, запустите скрипт `make-action.sh deploy -t cos`)

Добавьте параметр `-c <path_to_cos_config>`, если файл конфигурации COS имеет путь не по умолчанию (/opt/COS/config/cos_config.yml).

Процедура установит пакеты и настроит необходимые модули. В конфигурации георезервирования и высокой доступности она запустит репозиторий Nexus на всех хостах COS, настроит и запустит модуль keepalived (для плавающего IP-адреса) и модуль csync (для репликации папки /opt/COS между хостами COS).

4.2.5 Развертывание виртуальных машин на хосте COS

```
cd /opt/COS/it_cos_setup/ansible
export PATH=$PATH:../tools
```

4.2.5.1 Развертывание сервера IPA

Параметры в разделах COMMON DATA и IPA должны быть определены в файле конфигурации сервера COS до развертывания сервера IPA (см. раздел 4.2.3). См. также раздел 9.1 для получения дополнительной информации о настройке сервера IPA.

Предупреждение!

Значения параметров *ipa_admin_password* и *ipa_dm_password* должны быть не менее 8 символов и не должны быть подстрокой доменного имени (например, если доменом является *cloud.iskrauraltel.ru*, пароли не должны быть *iskrauraltel*).

Запустите скрипт:

```
cos-ipa.sh deploy
```

Добавьте параметр `-c <path_to_cos_config>`, если файл конфигурации COS находится не в директории по умолчанию (/opt/COS/config/cos_config.yml).

Процедура создаст виртуальную машину сервера IPA, установит и настроит сервер IPA и сервер DNS для поддомена ICP и добавит группы и пользователей IPA по умолчанию.

4.2.5.2 Развертывание ВМ операций

Параметры в разделах COMMON DATA и OPERATIONS VM должны быть определены в файле конфигурации COS до развертывания ВМ операций (см. раздел 4.2.3).

Если компоненты Openstack Client, Openstack Rally, Openstack Shaker и/или Bacula Client с Bacul Toolbox должны быть развернуты автоматически, установите значения *operations_vm.openstack_client*, *operations_vm.openstack_rally*, *operations_vm.openstack_shaker* и/или *operations_vm.bacula_client* как *yes* .

Измените файл *docker-compose.yml* для включения ротации журнала td-agend.

Перед запуском обновления добавьте в файл `/opt/it-efk/docker-compose.yml` строки:

```
##you can add this to all 3 conteiners it will do just good things, but demad is for
fluentd
    logging:
      driver: "json-file"
      options:
          max-file: 5
          max-size: 10m
```

Например:

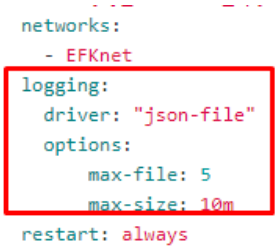


Рис. 4-3: Развертывание ВМ операций – Настройка ротации журнала

Запустите скрипт:

```
cos-opvm.sh deploy
```

Процедура создаст ВМ операций, установит и настроит сервер управления облаком.

4.2.5.3 Развертывание сервера резервного копирования

Параметры в разделах *COMMON DATA* и *BACKUP* должны быть определены в файле конфигурации COS перед развертыванием сервера резервного копирования (см. раздел 4.2.3).

Если сервер NFS должен быть развернут автоматически, установите для *backup_vm.nfs_server* значение *yes*.

Подробнее о резервном копировании читайте в главе 10.

Запустите скрипт:

```
cos-backup.sh deploy
```

Процедура создаст ВМ резервного копирования, установит и настроит сервер резервного копирования.

4.2.5.4 Развертывание сервера мониторинга

Параметры в разделах *COMMON DATA* и *MON_VM* должны быть определены в файле конфигурации COS перед развертыванием сервера мониторинга (см. раздел 4.2.3).

Панели мониторинга добавляются в компоненте Grafana для всех облаков, заданных в разделе *ICP CLOUDS* файла конфигурации COS (см. раздел 4.2.3).

Запустите скрипт:

```
cos-mon.sh deploy
```

Добавьте параметр `-c <path_to_cos_config>`, если файл конфигурации COS не в директории по умолчанию (/opt/COS/ config/cos_config.yml).

Процедура запустит и настроит контейнеры Grafana и EFK на ВМ операций и добавит панели мониторинга облака в Grafana.

Предупреждение!

Из-за уязвимости log4j jar-файл `log4j` (созданный до 2022 года) необходимо удалить. Например:

```
root@vvcos1:~# find / -name log4j-core-2.11.1.jar -ls
   538519   1552 -rw-r--r--   1 root     root      1589223 Jan 10 18:26
/var/lib/docker/overlay2/7295e90924a14fcdb3fa306f4ef417e4fcf2cde8f7290209011397ce3888a325/diff
/usr/share/elasticsearch/lib/log4j-core-2.11.1.jar
  1291423   1572 -rw-r--r--   1 1000     root      1607947 Aug  8  2018
/var/lib/docker/overlay2/98926ab904603889a99dee09d8c28d5f4a6cb76279bd0676f24c6815db048267/diff
/usr/share/elasticsearch/lib/log4j-core-2.11.1.jar
root@vvcos1:~# rm
/var/lib/docker/overlay2/98926ab904603889a99dee09d8c28d5f4a6cb76279bd0676f24c6815db048267/diff
/usr/share/elasticsearch/lib/log4j-core-2.11.1.jar
root@vvcos1:~#
```

Также нужно удалить все файлы с расширением *.jar, имена которых начинаются на `elasticsearch-sql-cli`:

```
root@vvcos1:~# find / -name elasticsearch-sql-cli-* -ls
  1291360  18252 -rwxr-xr-x   1 1000     root     18686136 Mar 21  2019
/var/lib/docker/overlay2/072384b8e586197451bee9b80575b8b4e65a6064b697aafea9c63debd76aad07/merg
ed/usr/share/elasticsearch/bin/elasticsearch-sql-cli-6.7.0.jar
  1291360  18252 -rwxr-xr-x   1 1000     root     18686136 Mar 21  2019
/var/lib/docker/overlay2/98926ab904603889a99dee09d8c28d5f4a6cb76279bd0676f24c6815db048267/diff
/usr/share/elasticsearch/bin/elasticsearch-sql-cli-6.7.0.jar
root@vvcos1:~# rm
/var/lib/docker/overlay2/98926ab904603889a99dee09d8c28d5f4a6cb76279bd0676f24c6815db048267/diff
/usr/share/elasticsearch/bin/elasticsearch-sql-cli-6.7.0.jar
root@vvcos1:~# rm
/var/lib/docker/overlay2/072384b8e586197451bee9b80575b8b4e65a6064b697aafea9c63debd76aad07/merg
ed/usr/share/elasticsearch/bin/elasticsearch-sql-cli-6.7.0.jar
```

4.2.5.5 Развертывание сервера резервного копирования

Параметры в разделах *COMMON DATA* и *BACKUP* должны быть определены в файле конфигурации COS перед развертыванием сервера резервного копирования (см. раздел 4.2.3).

Если сервер NFS должен быть развернут автоматически, установите для *backup_vm.nfs_server* значение *yes*.

Подробнее о резервном копировании читайте в главе 10.

Запустите скрипт:

```
cos-backup.sh deploy
```

Процедура создаст виртуальную машину для резервного копирования, установит и настроит сервер резервного копирования.

4.2.6 Изменение репозитория на виртуальных машинах на хосте COS

4.2.6.1 Использование автоматических процедур настройки сервера COS

Подготовьте конфигурацию с помощью файла конфигурации cos_config.yml, который использовался при развертывании. Установите параметры репозитория:

```
# repository server data
offline_nexus: 'yes'     # 'yes' – offline nexus repo; 'no' – central nexus repo

repo_server:
  url: "http://cumulus.csi.iskrauraltel.ru:8081"  # URL of Nexus repo = http(s)://COS_SERVER_NAME_OR_IP:8081
                                                  # use name only when server name is resolvable, otherwise use IP
docker_registry:
  url: "http://cumulus.csi.iskrauraltel.ru:5001"  # URL of Nexus docker registry = http(s)://COS_SERVER_NAME_OR_IP:5001
                                                  # use name only when server name is resolvable, otherwise use IP
  user: "admin"                   # user on docker registry
  password: "[пароль_предоставляется_системным_администратором]"            # password of the user
```

Если необходимо, запустите скрипты обновления.

4.2.6.2 Использование скриптов на виртуальных машинах

Измените репозиторий на ВМ операций и ВМ резервного копирования.

На ВМ операций и ВМ резервного копирования подготовьте файл конфигурации yaml нового репозитория с нужными параметрами.

Например:

```
offline_nexus: 'yes'
#”yes” to use offline nexus repo and “no” to use online nexus repo
repo_server:
  #url of Nexus repo == COS_SERVER_URL:8081
  url: http://bigcsi.csi.iskrauraltel.ru:8081
docker_registry:
  #url of Nexus docker registry == COS_SERVER_URL:5001
  url: http://bigcsi.csi.iskrauraltel.ru:5001
  user: "admin"    #user on docker registry
  password:  "[пароль_предоставляется_системным_администратором]"    #password  of  the
registry user
```

Запустите скрпит *itfaikollaserver-switch-repo.sh* на ВМ операций и ВМ резервного копирования с файлом конфигурации yaml (созданным на предыдущем шаге) в качестве параметра.

Например:

```
itfaikollaserver-switch-repo.sh repo-args.yaml
```

4.2.7 Вставка хоста COS в FreeIPA

Подготовьте конфигурацию:

Используйте файл конфигурации cos_config.yml, который использовался при развертывании. Установите для параметра cos_is_ipa_client значение yes:

```
# set COS server as an IPA client
cos_is_ipa_client: 'yes'
```

Запустите скрипт, который вставит хосты COS в IPA:

```
cd /opt/COS/it_cos_setup/ansible
export PATH=$PATH:../tools

cos-ipaclient.sh deploy
```

Добавьте параметр `-c <path_to_cos_config>` , если файл конфигурации COS не в директории по умолчанию (/opt/COS/config/cos_config.yml).

4.2.8 Вставка имени облака в FreeIPA DNS

Если параметр icp_clouds в конфигурационном файле cos_config.yml, который использовался при развертывании IPA, уже содержит данные об облаке, значит имя облака уже было включено в IPA DNS. В противном случае выполните следующую процедуру:

- Подготовьте конфигурацию:

Используйте файл конфигурации cos_config.yml, который использовался при развертывании. Установите значение параметра icp_clouds (можно включить более одного облака):

```
############################
# ICP CLOUDS
############################
icp_clouds:                          # clouds controlled by COS server
```

```
vmcloud1:                                  # name of cloud
  ext_url: "https://vmcloud.mydomain.test"    # public URL of cloud
  ext_float_ip: "192.168.100.19"               # external float IP address
  int_float_ip: "192.168.110.19"               # internal float IP address
  monitoring: 'yes'                           # automatically included to monitoring system
  ipa_integration: 'yes'                       # openstack cloud is automatically integrated to IPA
  add_cloud_name_to_dns: 'yes'                 # cloud name is added to IPA DNS
```

- Запустите скрипт обновления IPA:

```
cd /opt/COS/it_cos_setup/ansible
export PATH=$PATH:../tools

cos-ipa.sh upgrade
```

Добавьте параметр `-c <path_to_cos_config>`, если файл конфигурации COS не в директории по умолчанию (/opt/COS/config/cos_config.yml).

Процедура обновит сервер IPA на виртуальной машине IPA.

4.3 Архитектура георезервирования и высокой доступности сервера COS

Сервер COS может быть настроен в следующих конфигурациях:

- автономная конфигурация – один физический сервер, функциональные компоненты развертываются на виртуальных машинах без резервирования.
- конфигурация высокой доступности – два физических сервера на одной физической локации, функциональные компоненты развертываются на виртуальных машинах в режиме высокой доступности.
- конфигурация георезервирования – два физических сервера на разных физических локациях, функциональные компоненты развертываются на виртуальных машинах в режиме георезервирования.
- комбинированная конфигурация георезервирования и высокой доступности – два физических сервера на одной локации и еще два физических сервера на второй локации; функциональные компоненты развертываются на виртуальных машинах в режиме высокой доступности для каждой локации и в комбинации в режиме георезервирования между обеими физическими локациями (см. рис. ниже).

Компоненты сервера COS обеспечивают функциональности высокой доступности и георезервирования различными способами:

А) Каждое отдельное приложение обеспечивает эти функциональности самостоятельно, используя тот или иной встроенный механизм репликации.

Б) Применяются механизмы Linstor/DRBD9 (см. рис. ниже): этот способ используется в приложениях, использующих постоянные данные — например, в базах данных, серверах тарификации.
Постоянные данные этих компонентов находятся на виртуальном томе, подключенном к ВМ. Эти данные реплицируются на все физические серверы с помощью механизма DRBD9. DRBD управляется инструментом управления хранилищем Linstor. Виртуальный том подключается только к одной ВМ каждого функционального компонента. Приложение на этой ВМ работает в активном состоянии. На всех остальных виртуальных машинах оно работает в режиме ожидания.
Модуль Pacemaker в сочетании с Booth используется в качестве менеджера ресурсов кластера. Booth также требует выполнения процесса-арбитра, который запускается на ВМ на третьей локации (например, в пограничном облаке, общедоступном облаке).

B) Применяется механизм CSYNC2: этот способ используется в приложениях с постоянными данными, которые редко изменяются, такими как файлы конфигурации или образы приложений, а также в случаях, когда целостность данных не требуется на 100%.
Постоянные данные этих функций находятся в специальной папке. Эти данные реплицируются на все серверы с помощью CSYNC2. CSYNC2 обеспечивает асинхронную репликацию во все целевые папки на разных хостах.
Приложения на всех узлах работают в активном режиме, но только одно приложение на каждой стороне может быть доступно через плавающий IP-адрес.
Модуль Pacemaker или Keepalived используется в качестве менеджера ресурсов кластера для установки плавающего IP-адреса на одном узле на каждой локации.

4.3.1 Функции сервера COS в конфигурации георезервирования и высокой доступности

4.3.1.1 Сервер IPA

Сервер IPA может работать в нескольких конфигурациях в состоянии АКТИВНАЯ. Постоянные данные находятся в каталоге LDAP. Сервер 389 DS используется для хранения данных LDAP, а также выполняет репликацию данных в несколько локаций.

Таким образом, для обеспечения функциональности георезервирования и высокой доступности сервер IPA использует вариант A (см. раздел 4.3).

4.3.1.2 ВМ операций

ВМ операций хранит все свои постоянные данные (файлы конфигурации облака) в папке /opt/COS/cloud-cfg. Эти данные редко изменяются, поэтому можно использовать вариант B (см. раздел 4.3).

4.3.1.3 ВМ резервного копирования

ВМ резервного копирования содержит файлы конфигурации, расположенные в папке /opt/cos_setup. Эти данные редко изменяются, поэтому можно использовать вариант B (см. раздел 4.3).

4.3.1.4 ВМ мониторинга

ВМ мониторинга содержит несколько основных модулей:

- Grafana

Файлы конфигурации Grafana находятся в папке /opt/it-grafana. Эти данные редко изменяются, поэтому можно использовать вариант B (см. раздел 4.3).

- Kibana

Kibana не содержит постоянных данных, поэтому ее можно легко развернуть в режиме multiActive.

- Elasticsearch

Elasticsearch содержит базу данных NoSQL, в которой хранятся данные журналов. Эти данные могут быть реплицированы с помощью встроенного механизма (вариант A) или с помощью Linstore/DRBD (вариант Б, см. раздел 4.3).

- Prometheus

Prometheus содержит файлы конфигурации, расположенные в папке /opt/it-prometheus. Эти данные редко изменяются, поэтому можно использовать вариант B (см. раздел 4.3).

Данные метрик не должны реплицироваться. Вместо этого все серверы Prometheus на каждой локации должны извлекать данные метрик со всех целевых точек.

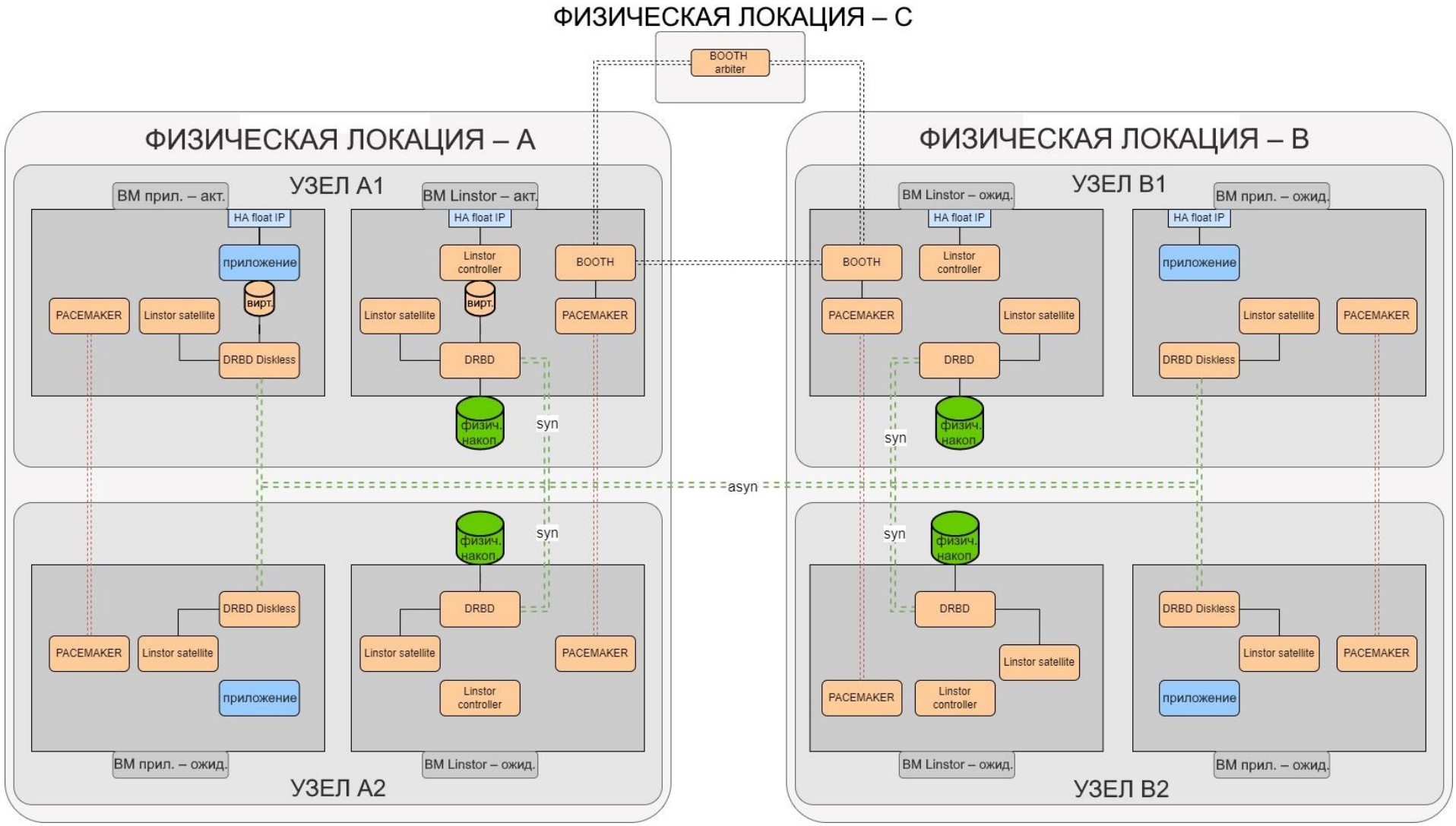


Рис. 4-4: Функциональные компоненты на сервере COS в конфигурации георезервирования и высокой доступности с использованием механизмов Linstor/DRBD

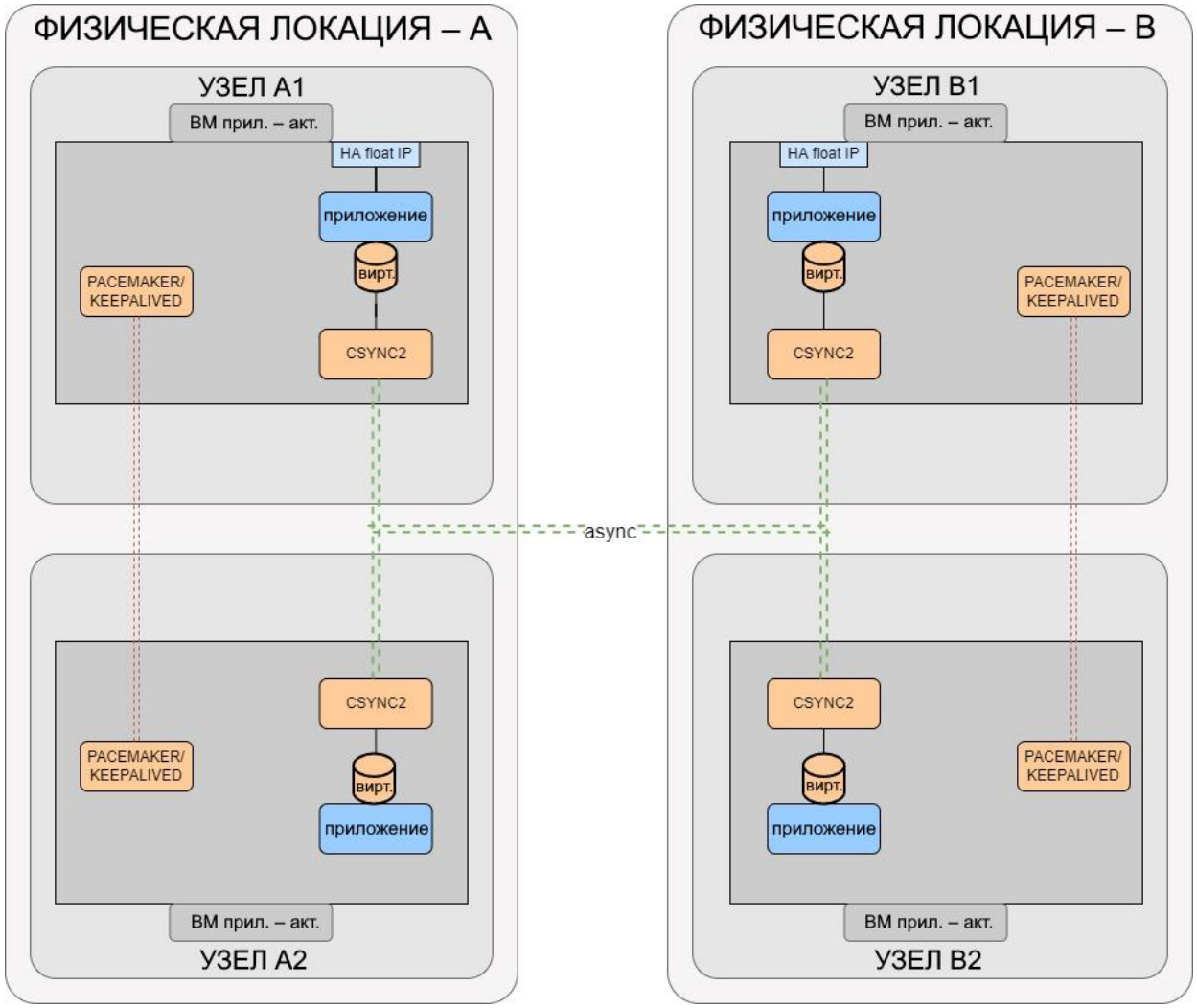


Рис. 4-5: Функциональные компоненты на сервере COS в конфигурации георезервирования и высокой доступности с использованием механизма CSYNC

4.4 Настройка сервера COS в режиме георезервирования и высокой доступности

Архитектура георезервирования и высокой доступности описана в разделе 4.3.

Процедура настройки сервера COS в данной конфигурации:

1. Разверните ОС на 4 серверах COS (по 2 на каждой локации) — см. раздел 4.1.
2. Настройте делегирование поддомена ICP (при необходимости) — см. раздел 4.2.1.
3. Подготовьте один из этих серверов COS — см. раздел 4.2.2 (сервер COS, на котором будут выполняться процедуры развертывания виртуальных машин, см. раздел 4.2.5).
4. Подготовьте файл конфигурации.
   Шаблон находится в директории /opt/COS/it_cos_setup/config/misc/cos_config.yml.multiinstance

```
---
############################
# COMMON DATA
############################
timezone: "Asia/Yekaterinburg"  # timezone
domain_name: "kemopq.test"   # domain name of COS server and the cloud

# host's data
cos_ha_geo: 'yes'
cos_servers:
 - name: "coskk1.kemopq.test"
   location: "yekaterinburg"
   public_ip_address: "192.168.26.15"
   public_net_bridge: 'br_out'      # COS server bridge connected to public network
   icp_internal_net_bridge: 'br_in'  # COS server bridge connected to internal cloud network
   password: "[пароль_предоставляется_системным_администратором]"
 - name: "coskk1.kemopq.test"
   location: "yekaterinburg"
   public_ip_address: "192.168.26.15"
   public_net_bridge: 'br_out'      # COS server bridge connected to public network
   icp_internal_net_bridge: 'br_in'  # COS server bridge connected to internal cloud network
   password: "[пароль_предоставляется_системным_администратором]"
 - name: "smallcsi1.kemopq.test"
   location: "moscow"
   public_ip_address: "192.168.37.34"
   public_net_bridge: 'br_out'      # COS server bridge connected to public network
   icp_internal_net_bridge: 'br_in'  # COS server bridge connected to internal cloud network
   password: "[пароль_предоставляется_системным_администратором]"
 - name: "smallcsi1.kemopq.test"
   location: "moscow"
   public_ip_address: "192.168.37.34"
   public_net_bridge: 'br_out'      # COS server bridge connected to public network
   icp_internal_net_bridge: 'br_in'  # COS server bridge connected to internal cloud network
   password: "[пароль_предоставляется_системным_администратором]"
cos_locations:
 - name: "yekaterinburg"
   floating_ip: '192.168.26.16'
   floating_name: 'cos-yekaterinburg'
 - name: "moscow"
   floating_ip: '192.168.37.35'
   floating_name: 'cos-moscow'

# ICP version's file
icp_versions_file: "/opt/COS/config/README.txt"    # path to package verision's file

# repository server data
offline_nexus: 'yes'
offline_nexus_archive: "/opt/COS/AI6212AX_offline_2.1.0.0.<ai_package_version>.tar.gz"
```

```
# OS images
ubuntu_image: "/opt/COS/images/ubuntu-20.04-it-server-cloudimg-2.1.5-amd64.qcow2"  # path to Ubuntu image

# COS params
cos_is_ipa_client: 'yes'  # include COS to IPA
cos_docker_bip: "none"       # no docker default network

############################
# COS modules
############################
########
# IPA
########
# IPA server params
ipa_admin_password: '[пароль_предоставляется_системным_администратором]'  # IPA admin password
ipa_dm_password: '[пароль_предоставляется_системным_администратором]'    # IPA directory manager password
dnssec_disable: 'yes'         # dnssec disable - default is yes

# IPA default groups and users
freeipa_groups:          # default groups added to IPA server
 osadmins:             # openstack admins group
  desc: "Openstack admin group"   # description of osadmins group
 osmembers:            # openstack members group
  desc: "Openstack member group"  # description of osmembers group

freeipa_users:          # default users added to IPA server
 keyadmin:             # user needed for IPA - Openstack integration
  first: "Keystone"    # firstname of keyadmin user
  last: "Backend"      # lastname of keyadmin user
  password: "[пароль_предоставляется_системным_администратором]"  # password of keyadmin user
 osadmin:              # openstack admin user
  first: "OsAdmin"     # firstname of osadmin user
  last: "Admin"        # lastname of osadmin user
  password: "[пароль_предоставляется_системным_администратором]"  # password of osadmin user
  group:             # groups to which osadmin user belongs
   - osadmins
 osmember:             # openstack member user
  first: "OsMember"    # firstname of osmember user
  last: "Member"       # lastname of osmember user
  password: "[пароль_предоставляется_системным_администратором]"  # password of osmember user
  group:             # groups to which osmember user belongs
   - osmembers

# IPA vm params
ipa_vm:                         # configuration data of ipa vm
 included: 'yes'                  # enabled or disabled
 name: 'ipasrv'                   # name of IPA vm
 locations:
  - name: "yekaterinburg"
  - name: "moscow"
 vm_info:
  instances:
   - location: "yekaterinburg"
     net_interfaces:
      public:
       address_cidr: '192.168.26.80/24'  # public IP address of IPA server
       gateway: '192.168.26.1'         # default gateway on public net
       nameservers:                # additional nameservers
        - '172.18.138.129'
       add_ipa_nameservers: 'no'
   - location: "yekaterinburg"
     net_interfaces:
      public:
       address_cidr: '192.168.26.81/24'  # public IP address of IPA server
```

```
        gateway: '192.168.26.1'         # default gateway on public net
        nameservers:                # additional nameservers
         - '192.168.26.80'          # IP of first IPA server
        add_ipa_nameservers: 'no'
    - location: "moscow"
      net_interfaces:
       public:
        address_cidr: '192.168.37.80/24'   # public IP address of operational server
        gateway: '192.168.37.1'         # default gateway on public net
        nameservers:                # additional nameservers
         - '192.168.26.80'          # IP of first IPA server
        add_ipa_nameservers: 'no'
    - location: "moscow"
      net_interfaces:
       public:
        address_cidr: '192.168.37.81/24'   # public IP address of operational server
        gateway: '192.168.37.1'         # default gateway on public net
        nameservers:                # additional nameservers
         - '192.168.26.80'          # IP of first IPA server
        add_ipa_nameservers: 'no'


#################
# OPERATIONS VM
#################
# Operations vm params
operations_vm:                  # configuration data of operations vm
 included: 'yes'                # enabled or disabled
 name: 'operationsvm'               # name of operations vm
 ipa_client: 'yes'              # VM is automatically inserted to IPA
 ha_geo: 'yes'
 locations:
  - name: "yekaterinburg"
    floating_ip: '192.168.26.82'
  - name: "moscow"
    floating_ip: '192.168.37.82'
 vm_info:
  instances:
   - location: "yekaterinburg"
     net_interfaces:
      public:
       address_cidr: '192.168.26.83/24'   # public IP address of operational server
       gateway: '192.168.26.1'         # default gateway on public net
      internal:
       address_cidr: '192.168.110.5/24'  # internal IP address of operational server
   - location: "yekaterinburg"
     net_interfaces:
      public:
       address_cidr: '192.168.26.84/24'   # public IP address of operational server
       gateway: '192.168.26.1'         # default gateway on public net
      internal:
       address_cidr: '192.168.110.5/24'  # internal IP address of operational server
   - location: "moscow"
     net_interfaces:
      public:
       address_cidr: '192.168.37.83/24'   # public IP address of operational server
       gateway: '192.168.37.1'         # default gateway on public net
      internal:
       address_cidr: '192.168.110.5/24'  # internal IP address of operational server
   - location: "moscow"
     net_interfaces:
      public:
       address_cidr: '192.168.37.84/24'   # public IP address of operational server
       gateway: '192.168.37.1'         # default gateway on public net
      internal:
```

```
        address_cidr: '192.168.110.5/24'  # internal IP address of operational server
  docker:
   docker_bip: "none"        # docker bip
  grafana:
   cert: 'selfsigned'  # can be "none", "selfsigned", "ipaprovided" or "path_to_certificate_folder", ipaprovided option not
supported yet
  efk:
   cert: 'selfsigned'  # can be "none", "selfsigned", "ipaprovided" or "path_to_certificate_folder", ipaprovided option not
supported yet
  openstack_client: 'yes'
  openstack_rally: 'yes'
  openstack_shaker: 'yes'
  bacula_client: 'yes'

###########
# BACKUP
###########
backup_vm:
  included: 'yes'
  name: 'backupsrv'
  ipa_client: 'yes'
  ha_geo: 'yes'
  locations:
    - name: "yekaterinburg"
      floating_ip: '192.168.26.85'
    - name: "moscow"
      floating_ip: '192.168.37.85'
  vm_info:
    instances:
      - location: "yekaterinburg"
        net_interfaces:
          public:
            address_cidr: '192.168.26.86/24'   # public IP address of operational server
            gateway: '192.168.26.1'          # default gateway on public net
      - location: "yekaterinburg"
        net_interfaces:
          public:
            address_cidr: '192.168.26.87/24'   # public IP address of operational server
            gateway: '192.168.26.1'          # default gateway on public net
      - location: "moscow"
        net_interfaces:
          public:
            address_cidr: '192.168.37.86/24'   # public IP address of operational server
            gateway: '192.168.37.1'          # default gateway on public net
      - location: "moscow"
        net_interfaces:
          public:
            address_cidr: '192.168.37.87/24'   # public IP address of operational server
            gateway: '192.168.37.1'          # default gateway on public net
  docker:
    docker_bip: "none"        # docker bip
  nfs_server: 'yes'

#############
# MONITORING
#############
# Monitoring vm params
mon_vm:
  included: 'yes'
  name: 'monitoring'
  ipa_client: 'yes'
  ha_geo: 'yes'
  locations:
    - name: "yekaterinburg"
      floating_ip: '192.168.26.88'
```

```
  - name: "moscow"
    floating_ip: '192.168.37.88'
 vm_info:
  instances:
   - location: "yekaterinburg"
     net_interfaces:
      public:
       address_cidr: '192.168.26.89/24'   # public IP address of operational server
       gateway: '192.168.26.1'          # default gateway on public net
     internal:
      address_cidr: '192.168.110.6/24'  # internal IP address of operational server
   - location: "yekaterinburg"
     net_interfaces:
      public:
       address_cidr: '192.168.26.90/24'   # public IP address of operational server
       gateway: '192.168.26.1'          # default gateway on public net
     internal:
      address_cidr: '192.168.110.6/24'  # internal IP address of operational server
   - location: "moscow"
     net_interfaces:
      public:
       address_cidr: '192.168.37.89/24'   # public IP address of operational server
       gateway: '192.168.37.1'          # default gateway on public net
     internal:
      address_cidr: '192.168.110.6/24'  # internal IP address of operational server
   - location: "moscow"
     net_interfaces:
      public:
       address_cidr: '192.168.37.90/24'   # public IP address of operational server
       gateway: '192.168.37.1'          # default gateway on public net
     internal:
      address_cidr: '192.168.110.6/24'  # internal IP address of operational server
 docker:
  docker_bip: "none"        # docker bip
 grafana:
  cert: 'selfsigned'
 efk:
  cert: 'selfsigned'

############################
# ICP CLOUDS
############################
icp_clouds:
 vmcloud1:                          # name of cloud
  ext_url: "https://vmcloud.kemopq.test"    # public URL of cloud
  ext_float_ip: "192.168.100.19"          # external float IP address
  int_float_ip: "192.168.110.19"          # internal float IP address
  monitoring: 'yes'                  # automatically included to monitoring system
  ipa_integration: 'yes'               # openstack cloud is automatically integrated to IPA
  add_cloud_name_to_dns: 'yes'           # cloud name added to IPA DNS
 vmcloud2:                          # name of cloud
  ext_url: "https://vmcloud2.kemopq.test"    # public URL of cloud
  ext_float_ip: "192.168.200.19"          # external float IP address
  int_float_ip: "192.168.210.19"          # internal float IP address
  monitoring: 'yes'                  # automatically included to monitoring system
  ipa_integration: 'yes'               # openstack cloud is automatically integrated to IPA
  add_cloud_name_to_dns: 'yes'           # cloud name added to IPA DNS
```

5. Выполните обмен ключами SSH между серверами COS:

```
cd /opt/COS/it_cos_setup/ansible

export PATH=$PATH:../tools

cos-sshkeys.sh deploy
```

6. Разверните пакеты и настройте модули на хосте COS — см. раздел 4.2.4.
7. Настройте делегирование субдомена ICP в IPA DNS.
8. Разверните виртуальные машины на хосте COS — см. раздел 4.2.5.
9. Если необходимо, обновите хост COS – процедура обновления хоста COS должна выполняться только на том сервере, где будут выполняться процедуры обновления ВМ.

4.5 Устранение ошибок

Лог-файлы процедур настройки COS находятся на сервере COS в папке:
`/var/log/cos_setup`

Сгенерированные файлы конфигурации находятся на сервере COS в папке:
`/opt/cos_setup`

5 Настройка сервера репозиториев

5.1 Автономный репозиторий Nexus

Процедура настройки автономного сервера Nexus выполняется в рамках настройки операционного сервера COS (см. главу 4), но может выполняться и отдельно.

Обычно сервер Nexus запускается в контейнере непосредственно на хосте (не на виртуальной машине). Он доступен по IP-адресу или имени основного хоста.

Он также может работать на ВМ (KVM или ВМ Openstack). Для этого нужно использовать образ Ubuntu с установленным пакетом docker.

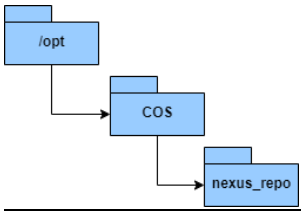
5.2 Процедура настройки сервера Nexus

Запустите модуль docker:

Проверьте, установлен ли модуль docker. Если нет, то обновите сервер (если используете продукт AI6211AX в качестве хоста) или установите пакеты docker.

```
systemctl start docker
systemctl enable docker
```

Подготовьте структуру папок:




```
cd /opt
mkdir COS
cd COS
mkdir nexus_repo
```

Предупреждение!

Структура папок может находиться в любой другой папке, отличной от */opt/COS*, учитывайте это при выполнении команд, приведеннных ниже.

Загрузите:

- архив данных Nexus в папку */opt/COS.*

Разархивируйте архив данных Nexus в папку *nexus_repo*:

```
cd /opt/COS
tar -xzf <nexus_archive>.tar.gz –C nexus_repo
```

Запустите автономный репозиторий Nexus:

```
cd /opt/COS/nexus_repo
./start-nexus.sh start --image=<image_name.tar.gz>
```

5.3 Процедура обновления сервера Nexus

Загрузите новый:

- архив данных Nexus в папку */opt/COS.*

Остановите автономный репозиторий Nexus и удалите файлы в нем:

```
cd /opt/COS/nexus_repo
./stop-nexus.sh
rm –r *
rm *
```

Разархивируйте новый архив данных Nexus в папку *nexus_repo*:

```
cd /opt/COS
tar -xzf <new_nexus_archive>.tar.gz –C nexus_repo
```

Запустите автономный репозиторий Nexus:

```
cd /opt/COS/nexus_repo
./start-nexus.sh start --image=<image_name.tar.gz>
```

6 Подготовка основного файла конфигурации облака

Шаблоны и примеры конфигурационных файлов можно скачать по адресу:

```
wget files.<repo-server-fqdn>/it_cloud-configs.tar
```

Обычный способ подготовки основного файла конфигурации — использовать шаблон для аналогичной конфигурации облака и изменить некоторые параметры.

Для облака с одним узлом используйте файлы конфигурации в папке:

it_cloud-configs/templates/single*

Для кластера с двумя узлами используйте файлы конфигурации в папке:

it_cloud-configs/templates/double*

Для кластера с тремя узлами используйте файлы конфигурации в папке:

it_cloud-configs/templates/triple*

Для всех остальных конфигураций используйте файлы конфигурации (шаблон для кластера из 4 узлов) в папке:

it_cloud-configs/templates/multiple-node*

Основной файл конфигурации называется <name>-big.yml. Измените как минимум значения в форме <value>.

Каждое облако должно иметь следующую структуру каталогов:

/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml

/opt/cloud-cfg/<cloud-name>/config/<additional-configuration-files>

где:

<additional-configuration-files> — все остальные файлы конфигурации, подготовленные в главе 7.

6.1 Получение информации об узлах и сетях облака

Для каждого узла получите следующую информацию:

- MAC-адреса всех сетевых интерфейсов или имена этих интерфейсов в соответствии со спецификацией CNDN (Consistent Network Device Naming), например, ens3.
- MAC-адреса сетевого интерфейса, который используется в качестве загрузочного интерфейса PXE/HTTP для установки операционной системы.
- Количество и размер всех жестких дисков.

Для сервера kolla-ansible получите следующую информацию:

- Имя интерфейса, который будет использоваться для развертывания хостов (deployment-interface).
- Имя интерфейса, который будет использоваться для подключения к серверу репозитория (uplink-interface)

Для сети получите следующую информацию:

- подсеть сети EXTERNAL-API,
- подсеть сети INTERNAL-API,
- подсеть сети STORAGE,
- подсеть сети TUNNEL,

- адрес внешнего API,
- адрес внутреннего API,

6.2 Подготовка файла конфигурации облака

Файл конфигурации облака представляет собой базовый файл yaml, который содержит все необходимые конфигурационные данные для развертывания облака. Имя основного файла конфигурации должно быть *<cloud-name>-big.yml*.

Файл конфигурации должен содержать обязательный параметр itc_deploy_config_version, который указывает версию конфигурации, эта версия используется для проверки совместимости со средствами развертывания облака. Если используется несовместимую версия конфигурации, при развертывании облака возникнет ошибка. Совместимые версии приведены в Табл. 6-1.

Таким образом, каждый параметр поддерживает определенную версию конфигурации (если для параметра не указана версия, он может применяться ко всем версиям конфигурации). Например, если для параметра указана версия конфигурации 2-all, это означает, что этот параметр применим для файлов конфигурации с версиями равными или выше 2.

Табл. 6-1: Поддерживаемые версии конфигурации

itc_deploy_config_version	Отличие от предыдущей версии
1	/
2	добавлен ceph_db_placement добавлены hugepages2M добавлены настройки sriov
3	добавлена опция предварительно заданного репозитория добавлены опции для автономного репозитория Nexus добавлены опции для поддержки нескольких версий ОС удалена настройка специального репозитория для pypi
4	новая конфигурация диска для внешнего ceph дополнительные настройки для изолированного развертывания внешнего модуля ceph (версия и пространство имен)

Файл конфигурации облака можно разделить на несколько разделов, которые описаны ниже.

6.2.1 Общие настройки

Табл. 6-2: Общие параметры в основном файле конфигурации облака

Имя	Тип	Верс. конф.	Описание	Пример
itc_deploy_config_version	обяз.	/	Версия этого файла конфигурации. Этот параметр является обязательным, чтобы можно было различать различные спецификации. В настоящее время поддерживаются версии: 1	1
cloud_name	обяз.	1-all	Имя облака (все имена хостов узлов будут содержать это имя плюс суффикс «-node»)	'testedge'
domain_name	необяз.	1-all	Доменное имя сетевых имен.	'iskrauraltel.ru'
it_kolla-ansible_version	обяз.	1-all	Номер версии Kolla ansible.	'1.4.5'
it_kolla_version	обяз.	1-all	Номер версии контейнеров Kolla.	'1.3.1'
cloud_custom_config_dir	необяз	1-all	Каталог на сервере управления облаком, который состоит из конкретных конфигураций облачных сервисов (см. главу 7).	'/opt/cloud-cfg/testedge/config/'

installation_server_ip	обяз.	1-all	IP-адрес или префикс интерфейса на сервере управления облаком, который будет использоваться для развертывания (deployment-interface-name)	192.168.33.250/24
license_file	обяз.	2-all	Расположение и имя файла лицензии, который используется для развертывания и реконфигурации облака (добавления новых узлов). Значение по умолчанию — /opt/cloud-cfg/<cloud-name>/license.txt	'/opt/cloud-cfg/testedge/license.txt'

Релизная версия модуля kolla ansible находится в файле версий модуля (например):

https://itbuild-raw.devops.iskrauraltel.cloud/AI6212AX/1.2.0.0.032/README.txt

Табл. 6-3: Релизная версия модуля Kolla ansible в файле версий модуля

Ключевое слово	Версия	Расположение	Описание
it_kolla_ansible	X.Y.Z	https://vmgitent.iskrauraltel.ru/AI6212AX/ it_kolla-ansible.git	Релизная версия модуля Kolla ansible

6.2.2 Сетевые настройки

Сетевые настройки используются для настройки сетевых адресов на интерфейсах и их назначений. Поддерживается тегирование и связывание сетей VLAN. Для конфигурации сети могут быть установлены опции «общие узлы» и дополнительно «заданные хосты». Опция «общие узлы» используется при наличии аналогичных облачных узлов и если известны имена их интерфейсов.

Во всех остальных случаях (если узлы имеют разные сетевые интерфейсы или если известны только MAC-адреса интерфейсов) используется конфигурация «заданные хосты».

Опция общих узлов задается следующей записью в файле yaml:

networking_common:

Конфигурация заданных хостов задается следующей записью в yaml:

networking_number_of_node:

Например (конфигурация сети для седьмого узла):

networking_7:

После этих параметров настраивается тип сетевой сущности. Это могут быть интерфейсы или связки интерфейсов (bonds). Обычно используются соединения, даже в случаях с только одним сетевым интерфейсом для каждого интерфейса в соединении.

После ввода типа сущности необходимо указать имя интерфейса, его MAC-адрес (например: ens5), а также свойства интерфейсов:

Табл. 6-4: Параметры сетевых интерфейсов в основном файле конфигурации облака

Имя	Тип	Верс. конф.	Описание	Пример
ip_start	обяз. исключение с ip	1-all	Начальный IP-адрес или префикс интерфейсов хостов. Значение адреса увеличивается на номер хоста, уменьшенный на единицу.	192.168.12.1/24
ip	обяз. исключение с ip_start	1-all	Фиксированный IP-адрес или префикс интерфейса. Его следует использовать только в конфигурации с заданными узлами, чтобы избежать дублирования IP-адресов.	192.168.12.7/24
gateway	необяз. единств.	1-all	IP-адрес шлюза для сети.	192.168.12.1

purpose	необяз. с обяз. глобальным присутствием для некоторых параметров	1-all	Назначение интерфейса для сервисов облака. Должен иметь одно из следующих значений: *external_api, internal_api, tunnel, provider, storage, cluster*. Все эти обязательные назначения должны быть доступны. Также существует необязательное назначение: *sriov* (поддерживается версиями конфигурации 2-all). Назначения tunnel и sriov могут быть установлены для нескольких интерфейсов.	storage-cluster
vlan_id:	необяз., включая с vlan_parrent	1-all	Тег VLAN интерфейса.	500
vlan_parent	необяз., включая с vlan_id	1-all	Родитель интерфейса VLAN.	ens2
mtu	необяз.	1-all	Максимальная единица передачи для интерфейса	1500

При наличии связки интферфейсов необходимо указать имя связки (например, storagebond) и свойства связки:

Табл. 6-5: Параметры связок интерфейсов в основном файле конфигурации облака

Имя	Тип	Верс. конф.	Описание	Пример
members	обяз.	1-all	Интерфейсы, входящие в состав связки.	ens1-ens2
mode	обяз.	1-all	Тип связки. Возможные значения: balance-rr, active-backup, balance-xor, Broadcast, 802.3ad, balance-tlb, balance-alb. См. Табл. 6-6 для дополнительной информации.	active-backup
ip_start	обяз. исключение с ip	1-all	Начальный IP-адрес или префикс интерфейсов хостов. Значение адреса увеличивается на номер хоста, уменьшенный на единицу.	192.168.12.1/24
ip	обяз. исключение с ip_start	1-all	Фиксированный IP-адрес или префикс интерфейса в связке. Его следует использовать только в конфигурации с заданными узлами, чтобы избежать дублирования IP-адресов.	192.168.12.7/24
gateway	необяз. единств.	1-all	IP-адрес шлюза для сети.	192.168.12.1
purpose	необяз. с обяз. глобальным присутствием для некоторых назначений	1-all (обяз.) 2-all (необяз.)	Назначение интерфейса для сервисов облака. Возможные значения: *external_api, internal_api, tunnel, provider, storage, cluster*. Все эти обязательные назначения должны быть доступны. Также существует необязательное назначение: *sriov* (поддерживается версиями конфигурации 2-all). Назначения *tunnel* и *sriov* могут быть установлены для нескольких интерфейсов.	storage-cluster
vlan_id	необяз. включение с vlan_parrent	1-all	Тег VLAN связки интферфейсов.	500
vlan_parent	необяз. включение с vlan_id	1-all	Родитель связки VLAN. Если родитель не используется или не указан ранее, здесь можно использовать любое имя, и связка будет создана автоматически.	myrawbond
mtu	необяз.	1-all	Максимальная единица передачи для интерфейса.	1500

Предупреждение!

Значение параметра MTU следует указывать внимательно. Если вы устанавливаете значение MTU для интерфейса сети VLAN или для связки интерфейсов, убедитесь, что

вы указали то же самое значение или более высокое значение на интерфейсах более низкого уровня (параметр members для связки интерфейсов и parent для интерфейсов на основе VLAN).

Табл. 6-6: Типы сетевых связок интерфейсов Linux

Имя	Тип	Описания
balance-rr	0	Задает политику циклического перебора для обеспечения отказоустойчивости и балансировки нагрузки. Передачи данных принимаются и отправляются последовательно на каждом связанном ведомом интерфейсе, начиная с первого доступного.
active-backup	1	Задает политику «активный-резервный» для обеспечения отказоустойчивости. Передачи данных принимаются и отправляются через первый доступный связанный ведомый интерфейс. Другой связанный ведомый интерфейс используется только в случае сбоя активного связанного ведомого интерфейса.
balance-xor	2	Передачи основаны на выбранной политике хеширования. По умолчанию получается хэш с помощью операции XOR над MAC-адресами источника и получателя, умноженных на модуль числа ведомых интерфейсов. В этом режиме трафик, предназначенный для определенных одноранговых узлов, всегда будет отправляться через один и тот же интерфейс. Поскольку точка назначения определяется MAC-адресами, этот метод лучше всего подходит для трафика к одноранговым узлам по тому же каналу или в локальной сети. Если трафик должен проходить через один маршрутизатор, то этот режим балансировки трафика будет неоптимальным.
broadcast	3	Задает широковещательную политику для обеспечения отказоустойчивости. Все передачи отправляются на все ведомые интерфейсы.
802.3ad	4	Задает политику динамической агрегации каналов IEEE 802.3ad. Создает группы агрегации с одинаковыми настройками скорости и дуплекса. Передает и принимает на всех ведомых устройствах в активном агрегаторе. Требуется коммутатор, совместимый с 802.3ad.
broadcast	5	Задает политику балансировки нагрузки при передаче (TLB) для обеспечения отказоустойчивости и балансировки нагрузки. Исходящий трафик распределяется в соответствии с текущей нагрузкой на каждый ведомый интерфейс. Входящий трафик принимается текущим ведомым интерфейсом. Если принимающий ведомый интерфейс выходит из строя, другой ведомый принимает MAC-адрес вышедшего из строя ведомого. Этот режим подходит только для локальных адресов, известных модулю связывания ядра, и поэтому не может использоваться за мостом с виртуальными машинами.
balance-alb	6	Задает политику адаптивной балансировки нагрузки (ALB) для обеспечения отказоустойчивости и балансировки нагрузки. Включает балансировку нагрузки при передаче и приеме для трафика IPv4. Балансировка нагрузки на прием достигается за счет согласования ARP. Этот режим подходит только для локальных адресов, известных модулю связывания ядра, и поэтому не может использоваться за мостом с виртуальными машинами.

6.2.3 Специальные настройки модуля Kolla

Эти настройки относятся к конфигурации модуля kolla-ansible и определяют способ развертывания OpenStack. Этот раздел начинается со следующей записи в yaml:

```
kolla:
```

за которой следуют определенные параметры, которые описаны в следующих разделах:

- Указание источника (оставьте вариант по умолчанию):

```
kolla_base_distro: 'ubuntu'   #Valid options are ['debian', 'oraclelinux', 'rhel', 'ubuntu'] #config version: 1-all
kolla_install_type: 'source'   #Valid options are [ binary, source ] #config version: 1-all
```

Релизная версия openstack указана в файле версий модуля.

Табл. 6-7: Релизная версия Openstack в в файле версий модуля

Ключ. слово	Версия	Расположение	Описание
it_kolla	X.Y.Z	https://vmgitent.iskrauraltel.ru/AI6212AX/it_kolla.git	Релизная версия модуля Openstack

- Параметры сети:

enable_neutron_provider_networks: 'yes' #Specify to use provider networks #config version: 1-all

kolla_internal_vip_address: '192.168.12.19' #Internal API address of cloud services #config version: 1-all

kolla_external_vip_address: '192.168.19.19' #External API address for cloud services. Also the dashboard IP #config version: 1-all

keepalived_virtual_router_id: '67' #Keepalived id - must be uniqe to other clouds on the same L2 network. #config version: 1-all

Предупреждение!

Параметр keepalived_virtual_router_id может быть установлен только в значения от 1 до 255.

- Полное доменное имя (FQDN):

Для обращения к конечным точкам с полным доменным именем, вместо IP-адресов можно использовать переменные:

```
kolla_internal_fqdn
kolla_external_fqdn
```

Обычно достаточно использовать внешнее полное доменное имя, например:

```
kolla_external_fqdn: ‘mykolla.example.net’
```

Для сопоставления этих имен с настроенными IP-адресами необходимо выполнить конфигурирование вне модуля kolla либо на сервере DNS, либо в файле `/etc/hosts`.

- Реестр Kolla:

docker_registry: 'it-csp-repo-server.edge.iskrauraltel.ru:4567' #Registry URL #config version: 1-all

docker_namespace: 'deploy/infrastructure' #Registry namespace #config version: 1-all

docker_registry_username: 'deploy' #Registry username #config version: 1-all

docker_registry_password: '[пароль_предоставляется_системным_администратором]' ##Registry password #config version: 1-all

Предупреждение!

Серверная часть хранилища для сервисов OpenStack Cinder, Glance и Nova должна быть либо на базе модуля Ceph, либо на базе локального хранилища.

Таким образом устанавливаются либо **настройки модуля Ceph**, либо **настройки локального хранилища**, приведенные ниже, но не оба этих набора настроек!

- Настройки модуля Ceph (оставьте параметры по умолчанию):

ceph_version: '16.2.7' #config version: 1-all – specify ceph version

cephadm_registry_namespace: 'ceph' #config version: 1-all – registry namespace of ceph containers

enable_ceph: 'no' #config version: 1-all

enable_ext_ceph: 'yes' #config version: 1-all

nova_backend_ceph: 'yes' #config version: 1-all

enable_cinder: 'yes' #config version: 1-all

enable_cinder_backup: 'no' #config version: 1-all

cinder_backend_ceph: 'yes' #config version: 1-all

cinder_backup_driver: 'ceph' #config version: 1-all

glance_backend_file: 'no' #config version: 1-all

glance_backend_ceph: 'yes' #config version: 1-all

cephadm_dashboard: 'true' #config version: 1-all

cephadm_dashboard_server_addr: '192.168.27.11 #config version: 1-all '

ceph_pool_size: 3 #config version: 1-all

- Настройки локального хранилища (оставьте параметры по умолчанию):

enable_ceph: 'no' #config version: 1-all

enable_cinder: 'yes' #config version: 1-all

enable_cinder_backend_lvm: 'yes' #config version: 1-all

glance_backend_file: 'yes' #config version: 1-all

glance_backend_ceph: 'no' #config version: 1-all

enable_ext_ceph: 'no' #config version: 1-all

- Настройки модуля Prometheus (оставьте параметры по умолчанию):

enable_prometheus: 'yes' #config version: 1-all

enable_prometheus_haproxy_exporter: '{{ enable_haproxy | bool }}'#config version: 1-all

enable_prometheus_mysqld_exporter: '{{ enable_mariadb | bool }}' #config version: 1-all

enable_prometheus_node_exporter: '{{ enable_prometheus | bool }}' #config version: 1-all

enable_prometheus_cadvisor: '{{ enable_prometheus | bool }}' #config version: 1-all

enable_prometheus_memcached: '{{ enable_prometheus | bool }}' #config version: 1-all

enable_prometheus_alertmanager: '{{ enable_prometheus | bool }}' #config version: 1-all

enable_prometheus_ceph_mgr_exporter: '{{ enable_prometheus | bool and enable_ceph | bool }}' #config version: 1-all

enable_prometheus_libvirt_exporter: '{{ enable_prometheus | bool }}' #config version: 1-all

- Настройки агрегатора Fluent:

fluentd_server_ip: '192.168.25.82' # IP address of faikolla server, used for central logging #config version: 1-all

Предупреждение!

При выборе значений `kolla_enable_tls_external: 'yes'`, и/или `kolla_enable_tls_internal: 'yes'`, не забудьте предоставить сертификаты TLS:

- самоподписанный сертификат, сгенерированный с помощью скрипта *itkf-generate-certificates.sh*,
- прямо указанный подписанный сертификат.

- Настройки TLS на внешних конечных точках:

kolla_enable_tls_external: 'yes' #config version: 1-all

- Настройки TLS на внутренних конечных точках:

kolla_enable_tls_internal: 'yes' #config version: 1-all

kolla_copy_ca_into_containers: 'yes' #config version: 1-all

kolla_admin_openrc_cacert: '{{ kolla_certificates_dir }}/ca/root.crt' #config version: 1-all

openstack_cacert: '/etc/ssl/certs/ca-certificates.crt' #config version: 1-all

- Настройки резервирования ресурсов:

enable_resource_reservation: 'yes' #enable/disable service #config version: 1-all

controler_cpu_usage: '4' #nr. of cpu cores on host reserved for openstack services on controler node (also applies to controller+compute combination) #config version: 1-all

controler_mem_usage_mb: '32768' #RAM on host reserved for openstack services on controler node (also applies to controller+compute combination) #config version: 1-all

compute_cpu_usage: '2' #nr. of cpu cores on host reserved for openstack services on compute node #config version: 1-all

```
compute_mem_usage_mb: '16384' #RAM on host reserved for openstack services on compute node #config version: 1-all
```

- Параметры синхронизации NTP:

  Укажите хотя бы один внешний сервер NTP для синхронизации времени. Рекомендуется использовать сервер управления облаком в качестве первичного источника и какие-либо общедоступные источники в качестве резервных.

```
external_ntp_servers: #config version: 1-all
  - 192.168.41.9
  - 0.pool.ntp.org
  - 1.pool.ntp.org
  - 2.pool.ntp.org
  - 3.pool.ntp.org
```

- Параметры модуля SR-IOV:

Чтобы включить SR-IOV для развертывания через модуль kolla-ansible, установите для переменной `enable_neutron_sriov` значение yes:

```
enable_neutron_sriov: "yes"
```

Также необходимо правильно настроить модуль FAI и конфигурацию конкретных служб — см. раздел 7.1.2, где описывается настройка функции SR-IOV.

- Параметры распределенной виртуальной маршрутизации (DVR):

Чтобы разрешить создание маршрутизаторов DVR для развертывания через модуль kolla-ansible, установите для переменной `enable_neutron_dvr` значение yes (этот вариант несовместим с развертыванием через OVN — см. соответствующий раздел):

```
enable_neutron_dvr: "yes"
```

Предупреждение!

Этот параметр разрешен только для многоузловых конфигураций. Не настраивайте этот параметр при автономном развертывании платформы ICP.

Для дополнительной информации о параметрах высокой доступности маршрутизатора Neutron см. раздел 7.1.4.

- Параметры протокола резервирования виртуального маршрутизатора (VRRP) и резервирования DHCP:

Чтобы разрешить множественные экземпляры виртуальных маршрутизаторов и серверов DHCP, установите для переменной `enable_neutron_agent_ha` значение 'yes':

```
enable_neutron_agent_ha: "yes"
```

После включения этих параметров конфигурации все маршрутизаторы будут работать в режиме высокой доступности (по три экземпляра каждого маршрутизатора). Кроме того, после создания каждой сети будут созданы два сервера DHCP на двух разных управляющих сетевых узлах.

Предупреждение!

Этот параметр разрешен только для многоузловых конфигураций. Не настраивайте этот параметр при автономном развертывании платформы ICP.

Для дополнительной информации о параметрах высокой доступности маршрутизатора Neutron см. раздел 7.1.4.

- Настройки модуля ovs-dpdk:

Чтобы настроить модуль ovs-dpdk, добавьте следующие строки в настройки модуля kolla:

```
ovs_datapath: 'netdev'
enable_ovs_dpdk: 'yes'
enable_openvswitch: 'yes'
tunnel_interface: 'dpdk_bridge'
neutron_bridge_name: 'dpdk_bridge'
dpdk_interface_driver: 'vfio-pci'
```

Кроме того, необходимо настроить память страниц hugepages и применить специальную конфигурацию сетевой связности.

Для настройки высокой доступности модуля ovs-dpdk также можно настроить агрегирование интерфейсов dpdk, указав несколько интерфейсов с назначением provider и установив тип агрегирования в настройках модуля kolla:

```
dpdk_bond_mode: 'bond type'
```

Тип агрегирования может быть одним из следующих: active-backup, balance-tcp или balance-slb. Для дополнительной информации см. Табл. 6-8. Чтобы настроить агрегирование lacp, используйте следующую конфигурацию:

```
dpdk_bond_mode: 'balance-tcp'
```

Для дополнительной информации о настройках модуля ovs-dpdk см. раздел 7.1.5.

Табл. 6-8: Типы сетевых соединений ovs

Имя	Описание
active-backup	Этот режим предлагает аварийное переключение «активный/резервный», при котором резервная сетевая карта возобновляет сетевые операции при сбое активного соединения. Физическому коммутатору предоставляется только один MAC-адрес. Этот режим не требует какой-либо специальной поддержки или настройки коммутатора и работает, когда каналы подключены к отдельным коммутаторам. Этот режим не обеспечивает балансировку нагрузки.
balance-tcp	В этом режиме будет выполняться балансировка нагрузки с учетом данных уровней 2–4, например, MAC-адрес назначения, IP-адрес и TCP-порт. Кроме того, для balance-tcp требуется, чтобы на коммутаторе был настроен протокол LACP. Этот режим аналогичен агрегированию в режиме 4, используемому драйвером агрегирования Linux. По возможности рекомендуется использовать balance-tcp, так как протокол LACP обеспечивает высочайшую устойчивость при обнаружении отказа канала и предоставляет дополнительную диагностическую информацию о соединении. Рекомендуемый вариант — настроить balance-tcp с помощью LACP. Этот параметр совершает попытки настроить LACP, но будет откатываться к схеме «активный-резервный», если LACP не получается согласовать с физическим коммутатором.
balance-slb	Балансирует потоки на основе исходного MAC-адреса и выходной VLAN с периодической ребалансировкой по мере изменения шаблонов трафика. Агрегирование с помощью balance-slb позволяет выполнять ограниченную форму балансировки нагрузки без ведома или помощи удаленного коммутатора. SLB назначает каждую пару исходного MAC-адреса и сети VLAN на канал и передает все пакеты с этого MAC-адреса и сети VLAN по этому каналу. В этом режиме используется простой алгоритм хеширования, основанный на исходном MAC-адресе и номере VLAN, с периодической ребалансировкой по мере изменения шаблонов трафика. Этот режим аналогичен агрегированию в режиме 2, используемому драйвером агрегирования Linux. Этот режим используется, когда коммутатор настроен на агрегирование, но не настроен на использование LACP (статическое соединение вместо динамического).
none	Агрегирование не используется. Это настройка по умолчанию.

- Настройки виртуальных частных сетей как услуги (VPNaaS):

Чтобы включить VPNaaS на платформе ICP, нужно добавить следующую конфигурацию в файле конфигурации yaml:

```
enable_neutron_vpnaas: "yes"
enable_horizon_neutron_vpnaas: "yes"
```

Вторая строка включает настройку VPNaaS на панели управления платформы ICP, в противном случае конфигурация ограничивается только клиентом openstack.

- Настройки сервера IPA:

```
ipa_integration_enabled: 'yes'   # openstack is automatically integrated to IPA server
ipa_integration_params:          # parameters needed for integration Openstack with IPA
   ipa_servers_fqdn:              # list of fqdn names of IPA servers and replicas
    - "<ipa_server_name>.<ipa_server_domain>"  # fqdn name of IPA server
   ipa_admin_password: "[пароль_предоставляется_системным_администратором]"     # password
of IPA admin (defined in
                                    # cos_config.yml in IPA params)

   keystone_ipa_admin_user: "keyadmin" # IPA user used by keystone (defined in
                                    # cos_config.yml in freeipa_users param)
    keystone_ipa_admin_password:  "[пароль_предоставляется_системным_администратором]"  #
password of IPA user keyadmin (defined in
                                       # cos_config.yml in freeipa_users param)

openstack_domains:          # openstack domains integrated with IPA
  my_domain:                # name of openstack domain
    openstack_projects:     # openstack projects in my_domain
      my_project:           # name of openstack project
        freeipa_groups:     # user groups in IPA (defined in
                            # cos_config.yml in freeipa_groups param)
          osadmins:                    # name of first user group
            openstack_role: "admin"     # openstack role of group within my_project
          osmembers:                    # name of second user group
             openstack_role: "_member_" # openstack role of group within my_project
```

6.2.4 Настройки автоконфигурирования и репозитория

Эти настройки определяют процесс запуска узлов при установке операционной системы. Этот раздел начинается со следующей записи в yaml:

fai_boot:

за которой следуют определенные параметры:

- Сетевые свойства хостов:

```
hosts: 3                    #number of nodes for operating system installation #config version: 1-all
ip_start: 192.168.33.11/24    #starting IP for boot (offered by DHCP server) #config version: 1-all
macs:                       #list of MAC addresses of nodes (used by DHCP server offers) #config version: 1-all
    - 18:94:ef:60:0d:f3
    - 48:94:ef:60:2c:1f
    - 58:94:ef:66:8d:af
```

- Сетевые свойства сервера управления облаком:

```
dhcp_dns: 10.1.2.3          #DNS server address used within boot process #config version: 1-all
dhcp_gateway:  192.168.33.250/24            #OPTIONAL ENTRY: gateway for DHCP network, if not specified then
installation_server_ip entry is used #config version: 1-all
boot_mode: 'uefi'             #boot mode:  uefi or pxe (in the future it will be http, which is currently not yes supported)
#config version: 1-all
```

- Свойства программных репозиториев:

Для дополнительной информации о полях конфигурации репозитория см. Табл. 6-9. Все эти строки должны быть добавлены под записью fai_boot.

Табл. 6-9: Свойства программных репозиториев

Имя	Тип	Вер. конф.	Описание	Пример
packages_repository:	обяз.	1-all	URL-адрес репозитория с номером порта для пакетов операционной системы на основе apt. Возможные значения: • repository url • official (значение по умолчанию) • predefined Если это поле отсутствует, используется значение «official», что означает, что для установки операционной системы используется официальное зеркало дистрибутива. Вы также можете использовать значение «predefined»— в этом случае используется предопределенная конфигурация репозитория yaml, которая должна находиться в каталоге, специфичном для облака, и должна называться «repo_config.yml», например: /opt/etc-kolla/my_cloud/repo_config.yml	"http://bigcsi.csi.iskrauraltel.ru:8081"
packages_repository_template:	необяз.	3-all	Укажите шаблон jinja2, из которого создается конфигурация репозитория. Этот шаблон должен присутствовать, если URL-адрес репозитория используется в качестве значения для packages_repository: расположение по умолчанию — «/opt/cos_setup/repo_config.yml.j2» (этот файл автоматически копируется в процессе установки ВМ операций). Если вы хотите использовать другой шаблон, вы можете указать его здесь.	/opt/cos_setup/repo_config.yml.j2
pypi_server:	обяз. (устар.)	1-2	Репозиторий, используемый инструментом установки pip. Если не указан, используется официальный сервер.	"http://bigcsi.csi.iskrauraltel.ru"
offline_nexus:	необяз.	3-all	Введите «yes», если используется автономный сервер Nexus, и «no», если используется онлайновый сервер Nexus. Значение по умолчанию — «yes».	yes
base_os:	обяз.	3-all	Укажите релиз дистрибутива базовой операционной системы. Возможные значения: bionic, focal (по умолчанию)	focal

- Классы (обычно оставляется параметр по умолчанию):

  Для конфигурации можгут быть установлены опции «общие узлы» и дополнительно «заданные хосты». Конфигурация общих узлов задается следующей записью в файле yaml:

  classes_common: #config version: 1-all

  Конфигурация заданных узлов задается следующей записью в файле yaml:

  classes_number_of_node:

  Например, конфигурация классов для седьмого узла:

  classes_7:

Классы должны быть указаны в соответствии с конкретными настройками хоста (необходимые драйверы устройств, определенные роли хоста и т.д.).

Классы, которые являются обязательными для подготовки модуля kolla:

- FAIBASE

- KOLLA

Необязательные классы:

- FAITOOLS

Предупреждение!

Порядок классов в списке **важен**. Когда вы определяете классы, убедитесь, что последний класс в списке следующий:

- KOLLA

В соответствии с режимом загрузки, выбранным на предыдущих шагах, автоматически добавляется соответствующий класс. (GPTBIOSPART для pxe boot и UEFI для uefi or http boot).

В соответствии с базовой ОС (base_os), выбранной на предыдущих шагах, автоматически добавляются соответствующие классы (UBUNTU и BIONIC64/FOCAL64).

Дополнительно могут быть указаны следующие классы для определенного оборудования:

- UBUNTUMRS #Class for megaraid controller on Ubuntu 16.04

- MEGARAIDSAS #Class for megaraid controller on Debian 9

- I40E #Class for network card Intel i40e

В следующем примере показаны настройки по умолчанию:

classes_common:

- FAIBASE

- FAITOOLS

- KOLLA

6.2.5 Настройки назначений узлов

Эти параметры определяют назначение узлов на платформе ICP. Этот раздел начинается со следующей записи в yaml:

inventory:

за которой следуют определенные параметры:

controllers: '1-3' #list of nodes for controllers - this is list of node numbers - can also be range #config version: 1-all

network: '1-3' #list of nodes for network controllers- this is list of node numbers - can also be range #config version: 1-all

computes: '1-3' #list of nodes for compute services- this is list of node numbers - can also be range #config version: 1-all

monitoring: '1-3' #list of nodes for monitoring - this is list of node numbers - can also be range #config version: 1-all

storage: '1-3' #list of nodes for storage - this is list of node numbers - can also be range #config version: 1-all

Обычно значение количества управляющих узлов (controllers), сетевых узлов (network) и узлов мониторинга (monitoring) составляет 1 для облаков без высокой доступности и 1-3 для облаков с высокой доступностью. Количество вычислительных узлов (computes) и узлов хранения (storage) равняется количеству всех узлов в облаке.

6.2.6 Настройки переопределения порядка дисков (необязательно)

Эти настройки переопределяют порядок обнаружения дисков по умолчанию. Их следует использовать, если невозможно настроить порядок дисков через параметры в BIOS на сервере таким образом, чтобы первый диск был тем, на котором будет находиться операционная система.

Эти значения представляют собой список имен дисков, как они указаны в каталоге /dev/. Например: sdc sdb sda. По умолчанию (без этих настроек) этот список создается с помощью обнаружения блочных устройств, при котором используется порядок дисков, заданный в BIOS. Поэтому, если вы собираетесь использовать диск, например, /dev/sdc в качестве системного диска, поставьте sdc на первое место, а остальные диски – после него в списке.

Синтаксис следующий:

```
disk_list: list_of_disks #config version: 1-all
```

Например:

```
disk_list: 'sdc sda sdb sdd' #config version: 1-all
```

6.2.7 Настройки раздела диска с операционной системой

Эти настройки определяют размеры разделов диска (в гибибайтах) для установки операционной системы.

Этот раздел начинается со следующей записи в файле yaml:

os_partitions: #config version: 1-all

за которой следуют конкретные параметры.

В случае использования стандартных дисковых разделов, можно применять следующие спецификаторы разделов:

```
boot: 1          #boot partitions size in GiB #config version: 1-all
root: 160        #root (/) partition size in GiB #config version: 1-all
swap: 16         #swap partition size in GiB #config version: 1-all
varlog: 22       #/var/log partition size in GiB #config version: 1-all
varlib: 32       #/var/lib partition size in GiB #config version: 1-all
opt: 32       #/var/lib partition size in GiB #config version: 5-all
```

В Табл. 6-10 представлены точки монтирования разделов.

Загрузочный (boot), корневой (root) и раздел подкачки (swap) являются обязательными, однако все эти разделы, кроме загрузочного, также могут быть основаны на lvm. В этом случае они задаются в конфигурации дисков.

Рекомендуемая конфигурация состоит в том, чтобы установить разделы boot, root и swap как стандартные, в то время как varlog и varlib должны быть установлены на основе lvm. При желании

можно установить раздел /opt на основе lvm, если конфигурация требует большого или изменяемого хранилища для раздела /opt. Размер разделов на основе lvm можно легко изменить, и в то же время разделы для операционной системы защищены от заполнения данными, т.к. это может привести к сбоям в работе системы.

Табл. 6-10: Точки монтирования для разделов модуля FAI

Спецификатор раздела	Точка монтирования
boot	/boot/efi или /boot для не-EFI-систем
root	/
swap	swap
varlib	/var/lib
varlog	/var/log
opt	/opt

6.2.8 Настройки размера дисков

Эти настройки определяют размеры дисков (в гибибайтах) и указывают, какие диски будут использоваться для хранения.

Предупреждение!

Размер диска традиционно указывается в гигабайтах (ГБ). В данном случае, пожалуйста, убедитесь, что все размеры в конфигурационных файлах указаны именно в гибибайтах (ГиБ). В Интернете можно найти инструменты конвертации этих величин.

Конфигурация общих узлов указывается следующей записью в файле yaml:

```
disks_common:
```

Конфигурация заданных узлов указывается следующей записью в файле yaml:

```
disks_number_of_node:
```

Например (конфигурация дисков для седьмого узла):

```
disks_7:
```

Предупреждение!

Разделы диска выравниваются по свойствам диска при их создании, что может привести к тому, что емкость диска будет немного меньше (до 1 ГиБ), чем указано в свойствах диска. Поэтому рекомендуется вычесть 1 ГиБ из емкости диска при расчете размеров разделов диска.

Раздел `lvm_disks` (версия конфигурации: 1-all) указывает размеры физических томов lvm на каждом имеющемся диске. Он может содержать столько записей, сколько есть дисков, которые будут частью группы томов lvm. Если на диске не должно быть lvm, то запись для этого диска должна иметь значение 0, в противном случае должно быть указано количество ГиБ для дисков lvm. Все физические тома становятся частью группы томов с именем vg1. Затем эта группа используется в качестве базы для создания разделов lvm, указанных в разделе `lvm_partitions`. Размер `lvm_disks` не может быть -1.

Для указания разделов lvm следует использовать спецификатор `lvm_partitions`. За каждым именем раздела должен следовать его размер в ГиБ. Все разделы, приведенные в Табл. 6-10, за исключением загрузочного, могут быть основаны на lvm. Однако только разделы, не указанные в

настройках раздела диска операционной системы, могут быть указаны как lvm. А корневой раздел и раздел подкачки должны присутствовать либо как стандартные, либо как разделы lvm.

Предупреждение!

Убедитесь, что сумма всех размеров `lvm_partitions` равна сумме всех размеров `lvm_disks`.

Раздел `ceph_disks` (версия конфигурации: >=4) указывает наличие блочных разделов ceph на каждом доступном диске. Он должен содержать столько записей, сколько есть дисков, которые будут использоваться ceph. Если блочный раздел ceph не должен находиться на диске, то запись для этого диска должна иметь значение 0, иначе следует использовать значение -1. Этот параметр является альтернативой параметру ceph_devices.

Раздел `ceph_devices` (версия конфигурации: >=4) Определяет список блочных устройств ceph. Например:

```
/dev/sdb
/deb/sdc
```

Этот параметр переопределяет параметр `ceph_disks`.

Раздел `ceph_db_disks` (версия конфигурации: >=4) указывает наличие разделов базы данных ceph (ceph_db и ceph wall) для используемого кластера ceph. Если нет параметра ceph_db_disks, то база данных ceph находится на дисках ceph. Если он присутствует, он должен содержать столько записей, сколько имеется дисков, которые будут использоваться базой данных ceph. Если предполагается, что раздел базы данных ceph не находится на диске, то запись для этого диска должна иметь значение 0, в противном случае следует использовать значение -1. Этот параметр является альтернативой параметру ceph_db_devices.

`ceph_db_devices` (версии conf: >=4) определяет список дисковых устройств базы данных ceph. Например:

```
/dev/sdb
/deb/sdc
```

Этот параметр переопределяет параметр `ceph_db_disks`.

Раздел `ceph_wal_disks` (версия конфигурации: >=4) определяет наличие разделов журнала упреждающей записи ceph (ceph wall) для используемого кластера ceph. Если параметра ceph_wal_disks нет, то журнал упреждающей записи (write-ahead-log) модуля ceph находится на дисках ceph_db. Если он присутствует, он должен содержать столько записей, сколько имеется дисков, которые будут использоваться журналом упреждающей записи модуля ceph. Если раздел ceph wal не должен находится на диске, то запись для этого диска должна иметь значение 0, в противном случае следует использовать значение -1. Этот параметр является альтернативой параметру ceph_wal_devices.

Раздел `ceph_wal_devices` (версия конфигурации: >=4) задает список дисковых устройств ceph с упреждающей записью. Например:

```
/dev/sdb
/deb/sdc
```

Этот параметр переопределяет параметр `ceph_wal_disks`.

Раздел `cinder_disks` (версия конфигурации: 1-all) указывает размеры физических томов lvm на каждом существующем диске, которые будут использоваться для логической группы cinder-volumes. Эта группа используется в качестве локального хранилища для томов cinder как альтернатива и/или дополнение к хранилищу ceph. Значения представляют собой список, который должен содержать столько записей, сколько есть дисков, которые будут использоваться томами cinder. Если на диске не должно быть томов cinder, то запись для этого диска должна иметь значение 0, в противном случае должно быть указано количество ГиБ. Если оставшийся диск (после разбиения lvm) должен использоваться для томов cinder, следует использовать значение -1 (или размер оставшегося диска). Однако значение -1 не допускается, если на диске используется хранилище ceph.

Предупреждение!

Размеры `cinder_disks` могут быть равны -1, только если на конкретном диске не указан ceph.

В следующем примере показан пример конфигурации для хранилища ceph (версия конфигурации: >=4):

```
disks_common:
  lvm_disks:            #disk sizes for lvm (if any disk does not contain ceph it must be
listed with 0)
         - 22
         - 22
  lvm_partitions:       #partition sizes for operating system installation via FAI
      varlog: 11        #/var/log partition size in GiB
      varlib: 33        #/var/lib partition size in GiB
  ceph_disks:           #disk sizes for ceph
    - 0
    - -1
    - -1
disks_7:
    ceph_db_disks:
      - 0
      - -1
      - 0
    ceph_disks:
      - 0
      - 0
      - -1
```

В следующем примере показан пример конфигурации для локального хранилища (которое включает группу логических томов cinder-volumes) (версия конфигурации: >=4):

```
disks_common:
  lvm_disks:            #disk sizes for lvm (if any disk does not contain ceph it must be
listed with 0)
         - 22
         - 22
  lvm_partitions:       #partition sizes for operating system installation via FAI
      varlog: 11        #/var/log partition size in GiB
      varlib: 33        #/var/lib partition size in GiB
  cinder_disks:           #disk sizes for cinder
    - 11
    - 999
    - -1
disks_7:
  lvm_disks:            #disk sizes for lvm (if any disk does not contain ceph it must be
listed with 0)
         - 22
         - 22
  lvm_partitions:       #partition sizes for operating system installation via FAI
      varlog: 11        #/var/log partition size in GiB
```

```
      varlib: 33        #/var/lib partition size in GiB
  cinder_disks:           #disk sizes for cinder
    - 0
    - -1
    - -1
```

В следующем примере показан пример конфигурации для комбинации локального хранилища (которое включает группу логических томов cinder-volumes) и общего хранилища ceph (версия конфигурации: >=4):

```
disks_common:
  lvm_disks:            #disk sizes for lvm (if any disk does not contain ceph it must be
listed with 0)
         - 22
         - 22
  lvm_partitions:       #partition sizes for operating system installation via FAI
      varlog: 11        #/var/log partition size in GiB
      varlib: 33        #/var/lib partition size in GiB
  cinder_disks:           #disk sizes for cinder
    - 11
    - 999
    - -1
disks_7:
  lvm_disks:            #disk sizes for lvm (if any disk does not contain ceph it must be
listed with 0)
         - 22
         - 22
  lvm_partitions:       #partition sizes for operating system installation via FAI
      varlog: 11        #/var/log partition size in GiB
      varlib: 33        #/var/lib partition size in GiB
  cinder_disks:           #disk sizes for cinder
    - 0
    - -1
    - -1
```

Контрольный список для проверки правильности настройки размеров разделов:

- Загрузочный раздел может быть указан только как стандартный раздел, остальные разделы могут быть либо стандартными, либо lvm.
- Размер раздела `cinder_disks` можно установить равным -1, чтобы использовать оставшийся диск после создания других разделов.
- Сумма размеров разделов `lvm_partitions` должна быть такой же, как сумма размеров дисков `lvm_disks`.
- Сумма разделов на каждом диске не может превышать размер диска в ГиБ (т.к. размер диска традиционно указывается в ГБ).
- Первый диск содержит все разделы, указанные настройками 'os_partitions', и необязательные разделы ceph и lvm.
- Непервые диски содержат необязательные разделы ceph и lvm.
- При использовании устройств для настройки модуля ceph (`ceph_devices`, `ceph_db_devices`, `ceph_wal_devices`) ни одно имя устройства нельзя использоваться повторно.

6.2.9 Настройки памяти

Поддерживаемые версии конфигурации: 2-all.

Эти настройки задают параметры памяти. Конфигурация общих узлов указывается следующей записью в файле yaml:

```
memory_common:
```

Конфигурация заданных узлов указывается следующей записью в файле yaml:

```
*memory_number_of_node: *
```

Например (конфигурация дисков для седьмого узла):

```
memory_7:
```

Запись в файле yaml сопровождается конкретными параметрами.

`hugepages2M` (версии конфигурации: 2-all)

Указывается количество страниц памяти размером 2 МБ, которые можно использовать со специальными высокопроизводительными приложениями, например, DPDK, VPP и др. Обратите внимание, что эта память не может использоваться обычными процессами, поэтому установите соответствующее значение. Размер используемой памяти рассчитывается следующим образом: <число_страниц> * 2 МБ.

В примере показано, как зарезервировать 4096 МБ памяти для hugepages:

```
memory_common:
    hugepages2M: 2048
```

`hugepages1G` (версии конфигурации: 2-all)

Указывается количество страниц памяти размером 1 ГБ, которые можно использовать со специальными высокопроизводительными приложениями, например, DPDK, VPP и т.д. Обратите внимание, что эта память не может использоваться обычными процессами, поэтому установите соответствующее значение. Используемый объем памяти рассчитывается следующим образом: <число_страниц> ГБ.

В примере показано, как зарезервировать 8 ГБ памяти для hugepages:

```
memory_common:
    hugepages1G: 8
```

7 Подготовка дополнительных файлов конфигурации облака

7.1 Предварительная подготовка файлов конфигурации облака

Примеры конфигурационных файлов уже были загружены на этапе выполнения следующей команды (см. главу 6):

```
wget files.<repo-server-fqdn>/it_cloud-configs.tar
```

Дополнительные файлы конфигурации необходимо подготовить в папке, указанной параметром cloud_custom_config_dir в основном файле конфигурации (обычно это */opt/cloud-cfg/<cloud-name>/config*).

Параметры в этих файлах включаются в файлы конфигурации модулей облака на этапе подготовки к развертыванию облака (см. раздел 8.2).

Типичный способ подготовки дополнительных файлов конфигурации — использовать ранее загруженные файлы для аналогичной конфигурации облака без изменения содержимого и скопировать их в папку *config*.

Для облака с одним узлом используйте файлы конфигурации в папке single, с двумя узлами — в папке double, для кластера с тремя узлами — в triple, для всех остальных конфигураций используйте файлы конфигурации в папке multiple, где подготовлен пример конфигурации для кластера из 4 узлов.

Предупреждение!

Изменяйте содержимое дополнительного конфигурационного файла с осторожностью.

7.1.1 Типовая конфигурация neutron

- Укажите диапазон VLAN, используемый для сетей провайдера.

  Назначение: ограничить диапазоны VLAN, используемые сетями провайдеров

  Файл: neutron/ml2_conf.ini

  Расположение файла: папка, указанная параметром cloud_custom_config_dir в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*)

  Инструкции по редактированию: добавьте спецификацию network_vlan_ranges в раздел ml2_type_vlan:

  ```
  [ml2_type_vlan]
  network_vlan_ranges: physnet:low_vlan:high_vlan
  ```

  Например:

  ```
  [ml2_type_vlan]
  network_vlan_ranges: physnet1:600:800
  ```

- Укажите драйвер брандмауэра.

  Назначение: использовать собственный драйвер брандмауэра OVS вместо комбинации Linux Bridge с iptables

  Файл: neutron/ openvswitch_agent.ini

Расположение файла: папка, указанная параметром cloud_custom_config_dir dir в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*)

Инструкции по редактированию: добавьте openvswitch как firewall_driver в раздел securitygroup:

```
[securitygroup]
firewall_driver = openvswitch
```

Предупреждение!

Не изменяйте (не перенастраивайте) этот параметр в облаке с активными экземплярами. В противном случае, после этого нужно перезапустить все узлы и облачные службы, а затем отключить все экземпляры повторно подключить их ко всем сетям.

- Разрешите прохождение трафика TIPC между виртуальными машинами.

  Назначение: Использование нативного драйвера брандмауэра OVS блокирует по умолчанию любой протокол Ethernet, кроме IPv4 и IPv6. Чтобы включить протокол L2 TIPC, его необходимо специально настроить.

  Файл: neutron/ml2_conf.ini

  Расположение файла: папка, указанная параметром cloud_custom_config_dir в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*)

  Инструкции по редактированию: добавьте параметр permitted_ethertypes с правильным значением. В случае TIPC это 0x88ca

  ```
  [securitygroup]
  permitted_ethertypes = 0x88ca
  ```

Примечание.

Если не используется нативный драйвер брандмауэра OVS, не требуется устанавливать специальное разрешение для TIPC.

7.1.2 Конфигурация сквозной сети SR-IOV на основе neutron

SR-IOV — это спецификация, позволяющая устройству PCIe, например, сетевой карте, представляться как несколько отдельных физических устройств PCIe. Спецификация SR-IOV была создана и поддерживается группой PCI SIG.

SR-IOV работает по принципу физических функций (PF) и виртуальных функций (VF). Физические функции (PF) — это полнофункциональные функции PCIe; виртуальные функции (VF) — это «облегченные» функции, у которых меньше конфигурационных ресурсов. Однако эти виртуальные функции могут быть напрямую подключены к виртуальным машинам с помощью сквозного метода.

Первый шаг — указать, какие устройства будут использоваться для сквозной передачи SR-IOV. Это можно сделать, установив назначение sriov для конкретного сетевого устройства. Обратите внимание, что это устройство должно быть физическим сетевым адаптером (он не может быть связкой интерфейсов). Чтобы указать, например, интерфейс eno6 как SR-IOV, добавьте в конфигурацию сети в файл yaml для развертывания облака следующие строки:

```
interfaces:
  eno6:
     purpose:
         - sriov
```

Эта конфигурация настраивает сетевое устройство и создает виртуальные функции для конкретной сетевой карты после установки операционной системы.

Второй шаг — добавить службу neutron sriov в конфигурацию облака, добавив следующую строку в конкретную конфигурацию модуля kolla в файле yaml для развертывания облака:

```
enable_neutron_sriov: "yes"
```

Эта конфигурация запускает службу neutron_sriov в выделенном контейнере для управления соединениями sriov.

Третий шаг — настроить конфигурации сервисов nova и neutron. В конфигурации nova нужно настроить фильтры, указать новые физические сети и сопоставить их с правильными интерфейсами.

В следующем примере предполагается, что существует интерфейс `eno6`, и мы создаем физическую сеть `sriovtenant1` для виртуальных сетей с поддержкой сквозной передачи SR-IOV:

- Настройка фильтров nova:
  - *Файл:* nova.conf
  - *Расположение файла:* папка, указанная параметром `cloud_custom_config_dir` в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*)
  - *Инструкции по редактированию:* добавьте параметр PciPassthroughFilter в сервис nova и внесите в белый список указанное устройство и физическую сеть.
  - Пример:

```
[DEFAULT]
block_device_allocate_retries = 600
scheduler_default_filters = AvailabilityZoneFilter, ComputeFilter,
ComputeCapabilitiesFilter, ImagePropertiesFilter, ServerGroupAntiAffinityFilter,
ServerGroupAffinityFilter,PciPassthroughFilter, NUMATopologyFilter
scheduler_available_filters = nova.scheduler.filters.all_filters

[pci]
passthrough_whitelist = [{"devname": "eno6", "physical_network": "sriovtenant1"}]
```

- Добавьте новую физическую сеть в конфигурацию сервиса neutron:
  - *Файл:* neutron/ml2_conf.ini
  - *Расположение файла:* папка, указанная параметром `cloud_custom_config_dir` в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*)
  - *Инструкции по редактированию:* добавьте драйвер механизма `sriovnicswitch` и добавьте физическую сеть к активным сетям.
  - Пример:

```
[ml2]
mechanism_drivers = openvswitch,l2population,sriovnicswitch

[ml2_type_vlan]
network_vlan_ranges = physnet1:700:840, sriovtenant1:200:3000

[ml2_type_flat]
flat_networks = physnet1, sriovtenant1

[securitygroup]
firewall_driver = openvswitch
```

- Настройте агент sriov с сопоставлениями физической сети и сетевых карт:
  - *Файл:* neutron/sriov_agent.ini
  - *Расположение файла:* папка, указанная параметром `cloud_custom_config_dir` в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*)
  - *Инструкции по редактированию:* добавьте сопоставления ранее заданных физических сетей с сетевыми картами.
  - Пример:

```
[sriov_nic]
physical_device_mappings = sriovtenant1:eno6
exclude_devices =
```

Примечание.

У вас может быть включено много устройств SRIOV. В этом случае включите большее количество интерфейсов и установите их назначение как sriov. Также нужно добавить дополнительные физические сети, например, sriovtenant2 и создать сопоставление с правильным интерфейсом. В приведенном выше примере можно добавить интерфейс eno4 и сопоставить с ним сеть sriovtenant2.

При использовании сетей с поддержкой SRIOV необходимо убедиться, что виртуальная машина поддерживает драйвер виртуальной функции сетевой карты, используемой для этих целей.

Предупреждение!

Сквозные соединения на основе SR-IOV не используют механизм группы безопасности, поэтому контроль доступа к сети должен обеспечиваться путем применения правил брандмауэра непосредственно к виртуальной машине или сетевому оборудованию.

Для подключения виртуальных машин к сети с поддержкой SR-IOV обычные процедуры не работают. Поэтому для подключения виртуальной машины используйте следующую процедуру:

- создайте новую виртуальную сеть с базовой физической сетью, которая была создана во время установки SR-IOV, например, `sriovtenant1`;
- создайте новую подсеть, используя зарезервированные IP-адреса;
- создайте порты в этой подсети и укажите режим **Direct** вместо **Normal**;
- при создании виртуальной машины назначьте созданные порты на виртуальную машину.

7.1.3 Конфигурация для сквозной сети PCI на основе nova

В дополнение к сквозной сети с поддержкой SRIOV, сервис nova предоставляет механизм для подключения устройств PCI к экземплярам. С помощью опции конфигурации псевдонимов (alias) PCI в файле nova.conf, любое устройство PCI (физическая функция или виртуальная функция), поддерживающее сквозную передачу, может быть подключено к тому или иному экземпляру. Одним из основных недостатков, о котором следует помнить при использовании этого метода, является то, что опция псевдонима PCI использует только идентификатор продукта и идентификатор поставщика устройства, поэтому в средах, содержащих сетевые карты с несколькими портами, настроенными на поддержку SRIOV, невозможно указать конкретный порт сетевой карты, из которого будут извлекаться виртуальные функции.

Предупреждение!

Невозможно использовать сквозную сеть PCI в сочетании с конфигурацией permitted_ethertypes в Neutron (см. предыдущий раздел). Любой сетевой трафик, кроме IPv4, IPv6 и ICMP, будет отброшен независимо от этой конфигурации.

Чтобы разрешить такой трафик, например L2 TIPC, нужно отключить защиту портов на соответствующих портах.

Служба Nova Scheduler на управляющем узле требует, чтобы в список фильтров был добавлен PciPassthroughFilter, а также чтобы службы Nova Compute на вычислительных узлах были включены в белый список устройств PCI. Для службы Nova API на управляющем узле и службы Nova Compute на вычислительном узле также требуется параметр alias в разделе [pci]. Для этого параметра может быть установлено значение type-VF для сквозной передачи витртуальных функций или type-PF для сквозной передачи физических функций.

Первый шаг — указать, какие устройства будут использоваться для сквозной передачи. Это можно сделать, установив назначение sriov для конкретного сетевого устройства. Обратите внимание, что это устройство должно быть физическим сетевым адаптером (он не может быть связкой интерфейсов). Чтобы указать, например, интерфейс eno6 как SR-IOV, добавьте следующие строки в конфигурацию сети в файл yaml для развертывания облака:

```
interfaces:
  eno6:
     purpose:
         - sriov
```

Эта конфигурация настраивает сетевое устройство и создает виртуальные функции для конкретной сетевой карты после установки операционной системы.

Предупреждение!

Если требуется использовать физические функции вместо виртуальных функций, нужно вручную отключить службу sriov на всех узлах. В этом случае выполните подключение SSH к каждому облачному узлу после завершения этапа развертывания сервера управления облаком (FAI) и выполните следующие команды:

```
systemctl stop sriov
systemctl disable sriov
```

Второй шаг — определить адрес PCI, vendor_id и product_id сетевых устройств, которые будут использоваться механизмом сквозной передачи PCI. Чтобы найти эти данные, сначала используйте инструмент `lshw`, который выведет все устройства и их свойства, включая адрес PCI:

```
#lshw -class network
***output omitted***
  *-network:3
       description: Ethernet interface
       product: Ethernet Connection X722 for 1GbE
       vendor: Intel Corporation
       physical id: 0.3
       bus info: pci@0000:5a:00.3
       logical name: eno6
       version: 09
       serial: 36:4a:01:22:aa:6a
       size: 1Gbit/s
       capacity: 1Gbit/s
       width: 64 bits
       clock: 33MHz
       capabilities: pm msi msix pciexpress vpd bus_master cap_list rom ethernet physical
tp 1000bt-fd autonegotiation
       configuration: autonegotiation=on broadcast=yes driver=i40e driverversion=2.1.14-k
duplex=full firmware=3.33 0x80000fde 1.1824.0 latency=0 link=yes multicast=yes
port=twisted pair slave=yes speed=1Gbit/s
       resources: iomemory:23f0-23ef iomemory:23f0-23ef irq:31 memory:23ffa000000-
23ffaffffff memory:23fff000000-23fff007fff memory:e3c00000-e3c7ffff memory:23ffe000000-
23ffe3fffff memory:23fff020000-23fff09ffff
```

В приведенном выше примере выходных данных адрес PCI окрашен в красный цвет.

Затем используйте этот адрес в сочетании с командой `lspci`, чтобы получить vendor_id и product_id устройства, как в следующем примере:

```
# lspci -n | grep 5a:00.3
5a:00.3 0200: 8086:37d1 (rev 09)
```

В приведенном выше примере выходных данных идентификатор vendor_id выделен зеленым цветом, а product_id — фиолетовым.

Третий шаг — указать настройки службы nova. Нужно настроить PciPassthroughFilter в конфигурации nova и внести в белый список соответствующее сетевое устройство. Дополнительно задайте псевдоним (alias) с идентификатором поставщика и идентификатором продукта.

В следующем примере предполагается, что требуется, чтобы интерфейс eno6 с идентификатором поставщика 8086 и идентификатором продукта 37d1 представлялся как устройство PT-PF:

- Настройте фильтры и псевдонимы nova:
  - *Файл:* nova.conf
  - *Расположение файла:* папка, указанная параметром cloud_custom_config_dir в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*)
  - *Инструкции по редактированию:* добавьте параметр PciPassthroughFilter в службу nova и внесите в белый список указанное устройство и физическую сеть. Дополнительно добавьте псевдоним в раздел [pci] файла nova.conf.
  - Пример:

```
[DEFAULT]
block_device_allocate_retries = 600
scheduler_default_filters = AvailabilityZoneFilter, ComputeFilter,
ComputeCapabilitiesFilter, ImagePropertiesFilter, ServerGroupAntiAffinityFilter,
ServerGroupAffinityFilter,PciPassthroughFilter, NUMATopologyFilter
scheduler_available_filters = nova.scheduler.filters.all_filters

[pci]
passthrough_whitelist = { "vendor_id": "8086", "product_id": "37d1" }
alias = { "vendor_id":"8086", "product_id":"37d1", "device_type":"type-PF", "name":"a1" }
```

Примечание.

В качестве альтернативы можно использовать адрес PCI в конфигурации белого списка устройств:

```
passthrough_whitelist = { "address": "5a:00.3" }
```

Примечание.

Для параметра device_type можно установить type-PF (физическая функция) или type-VF (виртуальная функция).

Имя псевдонима является обязательным и используется в параметрах flavour, чтобы сообщить виртуальной машине, какие устройства использовать в качестве сквозных.

Последним шагом является настройка метаданных в настройках flavour, чтобы виртуальная машина могла использовать сквозные устройства PCI. Это можно выполнить в модуле Horizon, добавив метаданные flavour:

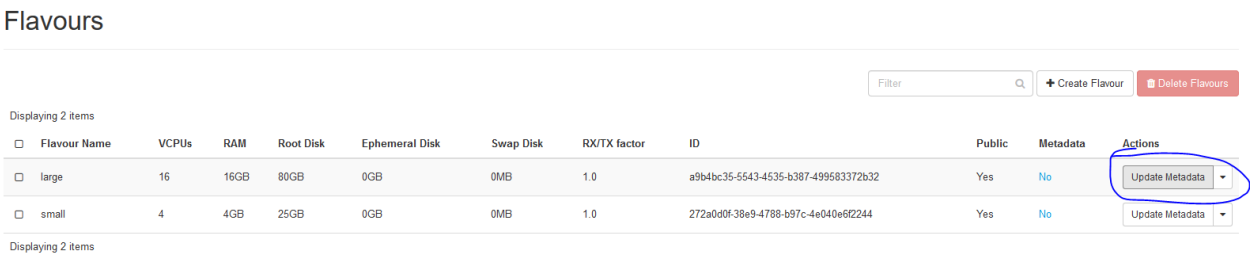


Рис. 7-1: Редактирование метаданных flavour

Необходимо добавить пользовательские метаданные pci_passthrough:alias, как на следующем рисунке:

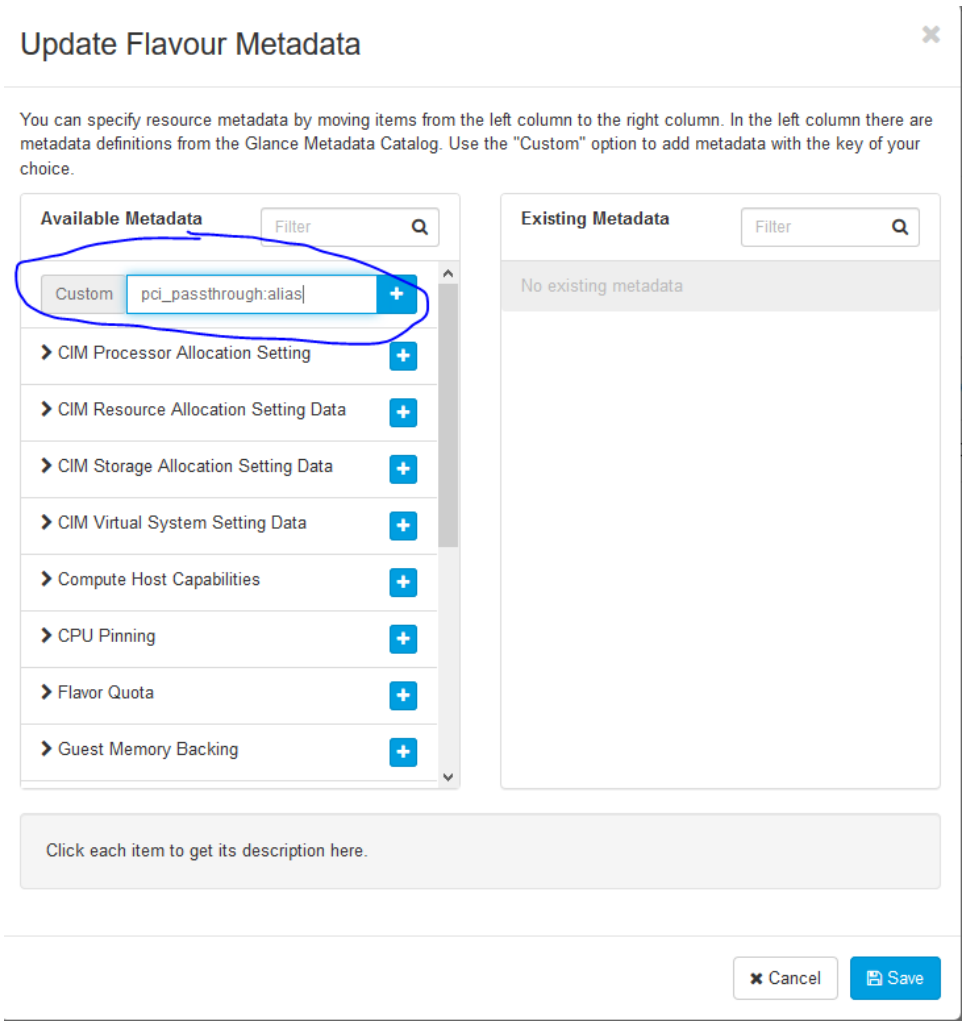


Рис. 7-2: Вставка метаданных flavour

Значением метаданных должно быть имя псевдонима (определенное на предыдущих шагах) и количество требуемых устройств. В следующем примере мы используем два устройства с псевдонимом a1.

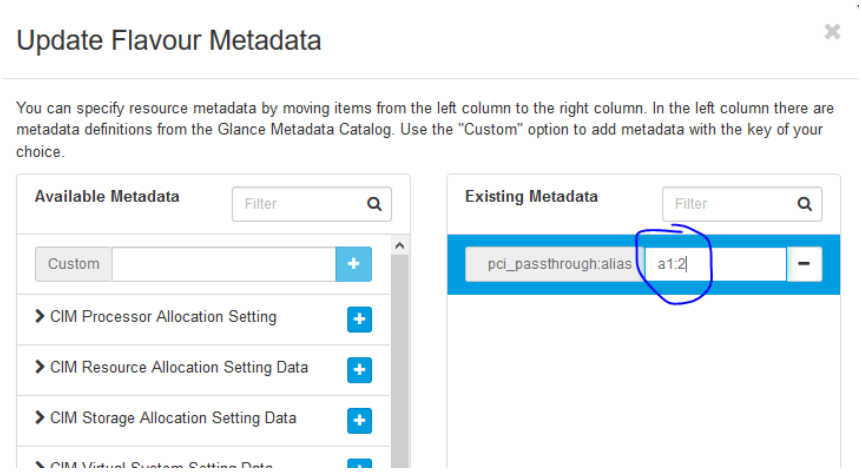


Рис. 7-3: Редактирование значения метаданных Flavour

Примечание.

Вы также можете установить метаданные flavour с помощью клиента openstack. например:

```
itopenstack my_cloud_name flavor set m1.large --property
"pci_passthrough:alias"="a1:2"
```

Последним шагом является создание экземпляра (BM) с использованием модифицированного flavour. При этом не нужно указывать какую-либо сеть для сквозных устройств PCI. Устройства будут созданы автоматически. А так как эти настройки не связаны с сервисом neutron, то порты, используемые сквозным PCI, не будут указаны в конфигурации neutron.

7.1.4 Параметры высокой доступности виртуального маршрутизатора

Высокая доступность виртуальных маршрутизаторов может быть достигнута с помощью протокола избыточности виртуальных маршрутизаторов (VRRP) или механизмов распределенной виртуальной маршрутизации (DVR) или их комбинации.

7.1.4.1 Протокол VRRP

VRRP обеспечивает аварийное переключение облачных маршрутизаторов для клиентских сетей с помощью механизма поддержания активности (keepalived). Он создает один ведущий (активный) экземпляр и один или несколько резервных экземпляров каждого маршрутизатора — один экземпляр на одном управляющем узле. При штатной работе механизм поддержания активности на ведущем маршрутизаторе периодически передает контрольные пакеты через дополнительную сеть VRRP, которая соединяет все маршрутизаторы VRRP. Каждый проект с маршрутизаторами VRRP использует собственную сеть VRRP.

Если механизм поддержания активности на резервных маршрутизаторах перестает получать контрольные пакеты, они предполагают отказ ведущего маршрутизатора и повышают уровень одного из резервных маршрутизаторов до ведущего путем настройки IP-адресов на интерфейсах в пространстве имен qrouter. Резервный маршрутизатор со следующим наивысшим приоритетом (с самым высоким IP-адресом) продвигает этот резервный маршрутизатор до ведущего маршрутизатора. Прерывание контрольного трафика VRRP между сетевыми узлами (как правило, из-за сбоя сетевого интерфейса или физической сетевой инфраструктуры) инициирует аварийное переключение. Перезапуск агента уровня 3 или его сбой не инициирует аварийное переключение при условии, что механизм поддержания активности продолжает работать.

Чтобы создать виртуальный маршрутизатор с поддержкой VRRP, используйте параметр `--ha` при создании маршрутизатора, как показано в следующем примере:

```
itopenstack mycloud router create --ha my_ha_router
```

Примечание. Чтобы маршрутизаторы с высокой доступностью создавались по умолчанию, платформа ICP должна быть правильно настроена в основном файле yaml облака — для переменной enable_neutron_agent_ha установите значение yes (см. раздел 6.2.3).

Чтобы создать виртуальный маршрутизатор без высокой доступности, используйте параметр `–no-ha` при создании маршрутизатора, как показано в следующем примере:

```
itopenstack mycloud router create –no-ha my_router
```

7.1.4.2 Механизм DVR

Механизм DVR распределяет службы виртуальной маршрутизации по всем вычислительным узлам. Таким образом, маршрутизация сетевого трафика в режиме «восток-запад» между экземплярами, подключенными к клиентским сетям на одном и том же виртуальном маршрутизаторе, происходит на вычислительных узлах. Сетевому трафику в этом случае не нужно проходить через управляющие или сетевые узлы. Однако экземпляры по-прежнему полагаются на управляющий или сетевой узел для обеспечения маршрутизации и услуг SNAT между сетями провайдеров и внешними сетями. Распределенный экземпляр маршрутизатора на каждом вычислительном узле использует IP-адрес в клиентской сети, в которой он содержит шлюз.

Чтобы разрешить создание маршрутизаторов DVR, платформа ICP должна быть правильно настроена в основном файле yaml облака (см. раздел 6.2.3). После выполнения этой настройки каждый вновь созданный виртуальный маршрутизатор по умолчанию будет являться распределенным. Однако также можно использовать параметр `--distributed` при создании маршрутизатора, как в следующем примере:

```
itopenstack mycloud router create --distributed my_distributed_router
```

Чтобы принудительно создать нераспределенный коммутатор, вы можете использовать параметр `--centralized`, как в следующем примере:

```
itopenstack mycloud router create --centralized my_router
```

7.1.4.3 Механизм DVR с протоколом VRRP

Механизм DVR также можно комбинировать с механизмом VRRP. В этом случае маршрутизаторы DVR/SNAT обеспечивают высокую доступность за счет быстрого переключения службы SNAT на резервный маршрутизатор DVR/SNAT на агенте уровня 3, работающем на другом узле.

Высокая доступность SNAT реализована так же, как и для обычных виртуальных маршрутизаторов с поддержкой VRRP, где механизм поддержания активности использует VRRP для обеспечения быстрого аварийного переключения служб SNAT. При штатной работе ведущий маршрутизатор периодически передает контрольные пакеты по выделенной сети, которая соединяет все экземпляры маршрутизатора с высокой доступностью. Если резервные маршрутизаторы DVR/SNAT перестают получать эти пакеты, они предполагают отказ главного маршрутизатора DVR/SNAT и выбирают новый ведущий маршрутизатор, настроив IP-адреса на интерфейсах в пространстве имен snat.

Чтобы разрешить создание маршрутизаторов DVR/VRRP, платформа ICP должна быть правильно настроена в файле yaml облака (см. раздел 6.2.3). При этой конфигурации виртуальные маршрутизаторы поддерживают параметры `--distributed` и `--ha`:

```
itopenstack mycloud router create --distributed --ha my_dvr_ha_router
```

Чтобы создать виртуальные маршрутизаторы без высокой доступности и без DVR, выполните:

```
itopenstack mycloud router create --centralized –no-ha my_dvr_ha_router
```

7.1.4.4 Резервирование DHCP

Для каждой сети можно включить службу DHCP, которая создает серверы DHCP на каждом управляющем узле. По умолчанию для каждой сети создается только один сервер DHCP на произвольном управляющем узле. Если вы настраиваете высокую доступность агента neutron в основном файле yaml облака, установите для переменной enable_neutron_agent_ha значение yes (см. раздел 6.2.3), тогда для каждой сети с поддержкой DHCP будут созданы два сервера DHCP.

7.1.5 Конфигурация Openvswitch с поддержкой DPDK

Data Plane Development Kit (DPDK) — это набор библиотек и инструментов пользовательского пространства, упрощающих разработку высокопроизводительных сетевых приложений пользовательского пространства. Путь данных Openvswitch (ovs) может поддерживать интеграцию с DPDK для ускорения работы сети пользовательского пространства — в этом случае используется сервис ovs-dpdk.

По умолчанию используется стандартный ovs, но в конфигурации, приведенной ниже, можно заменить ovs на ovs-dpdk. Это позволяет повысить пропускную способность и снизить сетевые задержки — благодаря оптимизации памяти и улучшенному использованию ЦП общая производительность сети становится намного выше.

Сервис ovs-dpdk работает только с поддерживаемыми сетевыми адаптерами. Список совместимого оборудования см. в документации по DPDK. Рекомендуется использовать ovs-dpdk в сетевых адаптерах 10G от Intel или Mellanox.

Список рекомендуемых чипсетов с драйверами:

- X710, XL710, X722, XXV71 – драйвер: i40e
- 82598, 82599, X520, X540, X550 – драйвер: ixgbe
- ConnectX-3, ConnectX-3 Pro – драйвер: mlx4

Однако для оптимизации требуется настройка памяти страниц hugepages на облачных узлах — для ovs-dpdk и всех виртуальных машин. Кроме того, ovs-dpdk утилизирует одно ядро процессора (это ядро постоянно загружено на 100%).

Из-за специфики работы Openvswitch с поддержкой DPDK необходимо изменить конфигурацию сервиса neutron, включив интерфейсы veth для службы neutron DHCP. Для этого нужно изменить файл `neutron/dhcp_agent.ini`. Этот файл находится в папке, указанной параметром cloud_custom_config_dir в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*), например `/opt/cloud-cfg/my_cloud/config/neutron/dhcp_agent.ini`

Добавьте в этот файл следующие строки:

```
[DEFAULT]
ovs_use_veth = true
[ovs]
datapath_type = netdev
```

Следующим шагом является изменение файла big yaml.

Вы можете настроить сервис ovs-dpdk в режиме высокой доступности или без высокой доступности.

A) Настройте сервис ovs-dpdk на использование единого интерфейса (режим без высокой доступности):

   Добавьте следующую конфигурацию в настройки модуля kolla:

   ```
   ovs_datapath: 'netdev'
   ```

```
enable_ovs_dpdk: 'yes'
enable_openvswitch: 'yes'
tunnel_interface: 'dpdk_bridge'
neutron_bridge_name: 'dpdk_bridge'
```

По умолчанию драйвером PMD для dpdk является `uio_pci_generic`, который является устаревшим драйвером. Драйвер, обеспечивающий большую безопасность в отношении изоляции — это `vfio-pci`. Чтобы использовать этот драйвер, примените следующую конфигурацию:

```
ovs_datapath: 'netdev'
enable_ovs_dpdk: 'yes'
enable_openvswitch: 'yes'
tunnel_interface: 'dpdk_bridge'
neutron_bridge_name: 'dpdk_bridge'
dpdk_interface_driver: 'vfio-pci'
```

При установке значения `vfio-pci` для dpdk_interface_driver, в конфигурацию ядра будут добавлены параметры IOMMU и intel-pt, чтобы обеспечить правильную работу драйвера.

Следующим шагом будет настройка сетевых адаптеров, которые будут использоваться сервисом ovs-dpdk. Поскольку DPDK использует собственную сетевую подсистему, интерфейсы, используемые для внешнего порта ovs-dpdk, могут быть только реальными интерфейсами, т.е. они не могут быть, например, связками интерфейсов Linux или мостами Linux.

В отличие от стандартного развертывания сервиса ovs, интерфейс, указанный с назначением `provider`, должен иметь назначенный IP-адрес. Этот IP-адрес будет установлен как `tunnel_interface` в процессе развертывания.

Поэтому нужно настроить один из физических интерфейсов с назначением `provider`. Этот интерфейс будет назначен в качестве порта dpdk мосту ovs-dpdk, чтобы разрешить соединения между узлами.

Назначение `tunnel` не нужно указывать в сетевой конфигурации, так как туннельный интерфейс — это пара интерфейс-порт 'dpdk_bridge', созданная при настройке сервиса ovs-dpdk (см. конфигурацию выше и поле `tunnel_interface`). Интерфейс `tunnel_interface` из настроек модуля kolla создается во время развертывания как мост ovs с физическими интерфейсами (указанными с назначением `provider`), подключенными для ускорения туннелируемого трафика с помощью сервиса dpdk.

Обратите внимание, что из-за ограничений имен переменных компонента ansible – в имени переменной невозможно использовать имя `br-ex` по умолчанию для интерфейса `tunnel_interface` – поэтому обычно используется имя `'dpdk_bridge'`.

Пример конфигурации сети с сетевыми интерфейсами сервиса ovs-dpdk:

```
networking_common:
  interfaces:
    eno4:
       ip_start: 172.30.4.11/24
       purpose:
           - provider
  bonds:
    cspexternal:
       members:
           - eno1
       mode: active-backup
       ip_start: 192.168.27.11/24
       gateway: 192.168.27.1
       purpose:
           - external_api
    cspinternal:
       members:
```

```
            - eno2
        ip_start: 192.168.157.11/24
        mode: active-backup
        purpose:
            - internal_api
    cspstorage:
        vlan_id: 500
        members:
            - eno3
        mode: active-backup
        vlan_parent: storageraw
        ip_start: 172.30.2.11/24
        purpose:
            - storage
            - cluster
```

Б) Настройте ovs-dpdk на использование нескольких интерфейсов (режим высокой доступности):

Добавьте следующую конфигурацию в настройки модуля kolla:

```
ovs_datapath: 'netdev'
enable_ovs_dpdk: 'yes'
enable_openvswitch: 'yes'
tunnel_interface: 'dpdk_bridge'
neutron_bridge_name: 'dpdk_bridge'
dpdk_bond_mode: 'active-backup'
```

См. Табл. 6-8 для информации о параметрах `dpdk_bond_mode` (выберите любой предпочтительный вариант, кроме `none`). Обратите внимание, что опция `balance-tcp` требует применения настроек LACP к коммутатору ToR.

По умолчанию драйвером PMD для dpdk является `uio_pci_generic`, который является устаревшим драйвером. Драйвер, обеспечивающий бо́льшую безопасность в отношении изоляции — это `vfio-pci`. Чтобы использовать этот драйвер, примените следующую конфигурацию:

```
ovs_datapath: 'netdev'
enable_ovs_dpdk: 'yes'
enable_openvswitch: 'yes'
tunnel_interface: 'dpdk_bridge'
neutron_bridge_name: 'dpdk_bridge'
dpdk_bond_mode: 'active-backup'
dpdk_interface_driver: 'vfio-pci'
```

При установке значения `vfio-pci` для dpdk_interface_driver, в конфигурацию ядра будут добавлены параметры IOMMU и intel-pt, чтобы обеспечить правильную работу драйвера.

Следующим шагом будет настройка сетевых адаптеров, которые будут использоваться сервисом ovs-dpdk. Поскольку DPDK использует собственную сетевую подсистему, интерфейсы, используемые для внешнего порта ovs-dpdk, могут быть только реальными интерфейсами, т.е. они не могут быть, например, связками интерфейсов Linux или мостами Linux.

В отличие от стандартного развертывания сервиса OVS, интерфейс, заданный с назначением `provider`, должен иметь назначенный IP-адрес. Этот IP-адрес будет установлен как `tunnel_interface` в процессе развертывания.

Поэтому нужно настроить один из физических интерфейсов с назначением `provider`. Этот интерфейс будет назначен в качестве порта dpdk мосту ovs-dpdk, чтобы разрешить соединения между узлами.

Назначение `tunnel` не нужно указывать в сетевой конфигурации, так как интерфейс туннеля — это пара интерфейс-порт dpdk_bridge, созданная при настройке сервиса ovs-dpdk (см. конфигурацию выше и поле `tunnel_interface`). Интерфейс `tunnel_interface` из настроек kolla создается во время развертывания как мост ovs с физическими интерфейсами

(указанными с назначением `provider`), подключенными для ускорения туннелируемого трафика с помощью сервиса dpdk.

Обратите внимание, что из-за ограничений имен переменных компонента ansible – в имени переменной невозможно использовать имя `br-ex` по умолчанию для интерфейса `tunnel_interface` – поэтому обычно используется имя `'dpdk_bridge'`.

Пример конфигурации сети с сетевыми интерфейсами сервиса ovs-dpdk:

```
networking_common:
  interfaces:
    eno4:
       ip_start: 172.30.4.11/24
       purpose:
           - provider
    eno5:
       ip_start: 172.30.5.11/24
       purpose:
           - provider
  bonds:
    cspexternal:
       members:
           - eno1
       mode: active-backup
       ip_start: 192.168.27.11/24
       gateway: 192.168.27.1
       purpose:
           - external_api
    cspinternal:
       members:
           - eno2
       ip_start: 192.168.157.11/24
       mode: active-backup
       purpose:
           - internal_api
    cspstorage:
       vlan_id: 500
       members:
           - eno3
       mode: active-backup
       vlan_parent: storageraw
       ip_start: 172.30.2.11/24
       purpose:
           - storage
           - cluster
```

Предупреждение!

Вы должны указать более одного интерфейса с назначением provider для использования режима высокой доступности ovs, поскольку ovs не поддерживает связки (агрегирование) с одним участником.

Если вы укажете несколько интерфейсов с назначением `provider`, а `dpdk_bond_mode` будет не указан или будет иметь значение `none`, то только один из этих интерфейсов будет использоваться в режиме без высокой доступности.

Как отмечалось ранее, для правильной работы сервиса ovs-dpdk необходима конфигурация памяти страниц hugepages. Сам ovs-dpdk потребляет по умолчанию примерно 4 ГБ памяти. Кроме того, для всех виртуальных машин необходимо настроить отображение памяти больших страниц. В противном случае порты сервиса ovs-dpdk для виртуальных машин будут созданы, но будут находиться в недоступном состоянии. Как следствие, вся память, используемая для сервиса

ovs-dpdk и потребления ВМ, должна быть в hugepages, т.е. вся доступная память за вычетом памяти, зарезервированной для облачных служб, и памяти, используемой для сервиса ovs-dpdk. См. параметры резервирования ресурсов в разделе 6.2.3 для настройки памяти.

Предупреждение!

По умолчанию сервис ovs-dpdk потребляет 4 ГБ памяти страниц hugepages. Это количество позволяет правильно обрабатывать большие сетевые пакеты. Если для всех виртуальных сетей установлено значение MTU, равное 1500, размер памяти можно установить равным 1 ГБ. Для этого установите переменную:

```
ovs_socket_met: 1042
```

в основном конфигурационном файле в разделе kolla.

Например, если у вас в системе 256 ГБ памяти, и вы зарезервировали 16 ГБ памяти для облака, а сервис ovs-dpdk использует 4 ГБ памяти, установите 236 ГБ или чуть меньше памяти для страниц hugepages:

```
memory_common:
        hugepages2M: 116000
```

Последний шаг — настроить метаданные всех сущностей flavours на использование памяти страниц hugepages: `hw:mem_page_size: large`.

Предупреждение!

Виртуальные машины или экземпляры, использующие flavours без метаданных `hw:mem_page_size: large`, все равно будут созданы, но все сетевые подключения к ним будут отброшены.

При создании сущностей flavours вы можете использовать команду в следующем примере:

```
itopenstack my_cloud flavor create --disk 30 --ram 4096 --vcpus 4 --property
hw:mem_page_size=large my_flavor
```

Вы также можете настроить метаданные на панели инструментов с помощью кнопки Update Metadata в настройках Flavours — настройте параметр Guest Memory Backing.

7.1.6 Настройка виртуальной частной сети как услуги (VPNaaS)

Функциональность VPNaaS позволяет установить VPN-соединение между функциями VNF (подключенными к локальным сетям) на двух разных платформах ICP. VPN-соединение устанавливается между двумя виртуальными маршрутизаторами на двух платформах ICP. Функции VNF, подключенные к локальным сетям, которые затем подключаются к этим маршрутизаторам, могут обмениваться данными через VPN.

Чтобы настроить VPNaaS, необходимо выполнить следующие шаги на каждой платформе ICP:

- настройте параметры модуля kolla в файле big yaml на этапе развертывания,
- создайте политику IKE,
- создайте политику IPsec,
- создайте сервис VPN,
- создайте группу конечных точек,
- создайте межлокационное подключение.

Межлокационные соединения IPsec поддерживают множественные локальные подсети в дополнение к текущим адресам CIDR множественных одноранговых узлов. Функциональность множественных локальных подсетей активируется, если не указать локальную подсеть при

создании службы VPN. Также можно ограничить функциональность до отдельных локальных подсетей, указав подсеть при создании службы VPN.

Для поддержки множественных локальных подсетей используется параметр под названием End Point Groups. Каждая группа конечных точек определяет одну или несколько конечных точек определенного типа и может использоваться для указания как локальных, так и одноранговых конечных точек для соединений IPsec. Группы конечных точек отделяют понятия «что подключается» от «как подключиться» для службы VPN и могут использоваться для различных сущностей flavours службы VPN.

Примечание.

Одна и та же конфигурация службы VPNaaS должна применяться ко всем платформам ICP, между которыми нужно установить VPN-соединения. В следующем разделе описываются шаги, которые необходимо выполнить на каждой плафторме ICP с поддержкой функциональности VPNaaS.

7.1.6.1 Настройка развертывания службы VPNaaS

Чтобы включить службу VPNaaS на платформе ICP, необходимо настроить файл конфигурации yaml, как описано в разделе 6.2.3. В сущности, просто добавьте следующие строки в раздел модуля kolla:

```
enable_neutron_vpnaas: "yes"
enable_horizon_neutron_vpnaas: "yes"
```

Эта конфигурация устанавливает службу VPNaaS с поставщиком услуг strongswan. Если нужно использовать какой-либо другой драйвер службы, можно отредактировать файл `/etc/neutron/neutron_vpnaas.conf` на каждом узле после развертывания. Используемый драйвер устройства – также stongswan. Чтобы перейти на другой драйвер, отредактируйте файл `/etc/neutron/l3_agent.ini` после развертывания.

7.1.6.2 Создание политики обмена ключами (IKE)

Для слудующих шагов предполагается, что развернуты две платформы ICP, и каждая ICP полностью настроена с созданными сетями и маршрутизаторами, к которым должным образом подключены функции VNF. Для каждой ICP необходимо указать маршрутизатор, на котором будет включена служба VPN, и который поэтому будет использоваться для VPN-соединений. Все локальные сети, которые будут использовать VPN-соединения, должны быть подключены к этому маршрутизатору. Маршрутизаторы также должны иметь настроенные внешние шлюзы.

Примечание.

Вы можете настроить службу VPNaaS из панели управления (если вы настроили параметр `enable_horizon_neutron_vpnaas` в конфигурации развертывания платформы ICP) — окно настроек службы VPN находится на боковой вкладке Project/Network — или из клиента openstack.

Чтобы создать политику IKE, щелкните кнопку `Add Ike Policy` в окне настроек VPN на первой вкладке.

Заполните необходимую информацию:

- Name: название политики.
- Description: описание политики.

- Authorization algorithm: алгоритм авторизации – на данный момент можно выбрать только sha1.
- Encryption Algorithm: алгоритм шифрования – выберите один из алгоритмов AES:
  - для базовой безопасности и максимальной производительности подходит aes128,
  - для лучшей безопасности – aes256;
  - рекомендуемый вариант – aes128.
- IKE version: версия политики IKE:
  - выберите рекомендуемую версию – v2.
- Lifetime value for IKE keys: срок действия ключей:
  - по умолчанию – 3600 с.
- Perfect Forward Secrecy: криптосистема PFS:
  - рекомендуемый вариант — group14, который в настоящее время является лучшим поддерживаемым решением.
- IKE Phase1 negotiation mode: режим согласования IKE – в настоящее время режим ограничен основным (main).

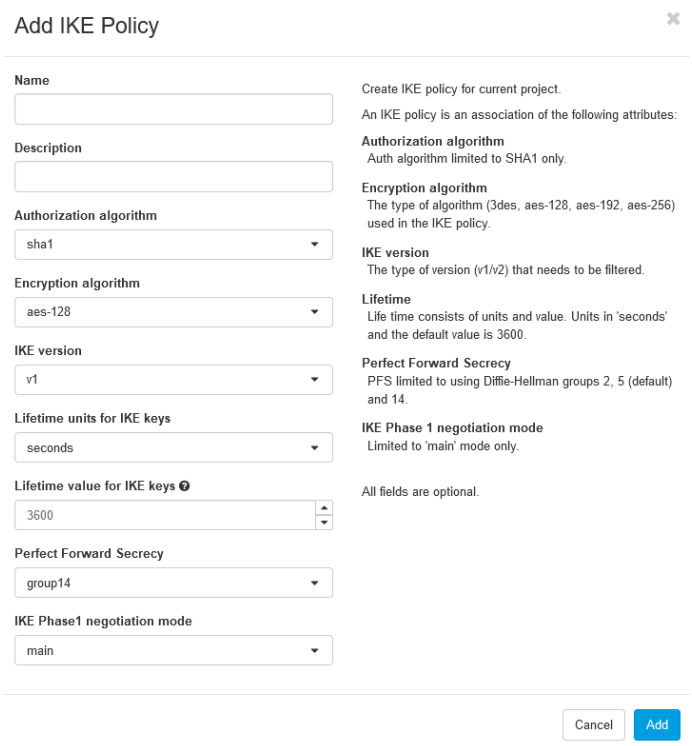


Рис. 7-4: Создание политики IKE

Чтобы создать политику IKE с помощью клиента Openstack, см. следующий пример:

```
itopenstack my_cloud vpn ike policy create --auth-algorithm sha1 --encryption-algorithm
aes-128 --phase1-negotiation-mode main --ike-version v2 --pfs group14 --lifetime
units=seconds,value=3600 myikepolicy
```

7.1.6.3 Создание политики защиты Интернет-протокола (IPsec)

Чтобы создать политику IPsec, в окне VPN, на второй вкладке, щелкните кнопку `Add IPsec Policy`.

Заполните необходимую информацию:

- Name: название политики.
- Description: описание политики.
- Authorization algorithm: алгоритм авторизации – на данный момент можно выбрать только sha1
- Encryption mode: режим инкапсуляции:

  - рекомендуемый вариант – туннель.
- Encryption Algorithm: алгоритм шифрования – выберите один из алгоритмов AES:
  - для базовой безопасности и максимальной производительности подходит aes128;
  - для лучшей безопасности – aes256;
  - рекомендуемый вариант – aes128.
- Lifetime value for IKE keys: срок действия ключей:
  - по умолчанию – 3600 с.
- Perfect Forward Secrecy: криптосистема PFS:
  - Рекомендуемый вариант — group14, который в настоящее время является лучшим поддерживаемым решением.
- Transform Protocol: протокол преобразования:
  - рекомендуемый и в настоящее время наиболее безопасный вариант — esp (ah-esp в настоящее время не поддерживается)

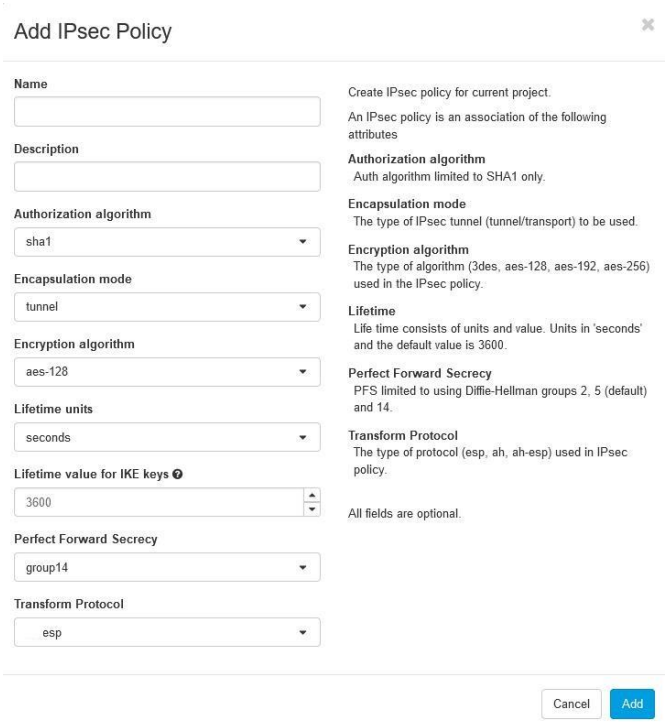


Рис. 7-5: Создание политики IPsec

Чтобы создать политику IPsec с помощью клиента openstack, см. следующий пример:

```
itopenstack my_cloud vpn ike policy create --auth-algorithm sha1 --encapsulation-mode
tunnel --encryption-algorithm aes-128 --lifetime units=seconds,value=3600 –pfs group14 --
transform-protocol esp myipsecpolicy
```

7.1.6.4 Создание службы VPN

Чтобы создать службу VPN, в окне VPN, на третьей вкладке, щелкните конпку `Add VPN Service`. В сущности, вам нужно добавить функцию VPN к выбранному маршрутизатору. Этот маршрутизатор будет использоваться для установления VPN-соединений.

Заполните необходимую информацию:

- Name: название службы VPN.
- Description: описание службы VPN.
- Router: маршрутизатор:
  - выберите маршрутизатор для службы VPN.

- Subnet: подсеть.
    - можно оставить это поле пустым, так как будут использоваться группы конечных точек.

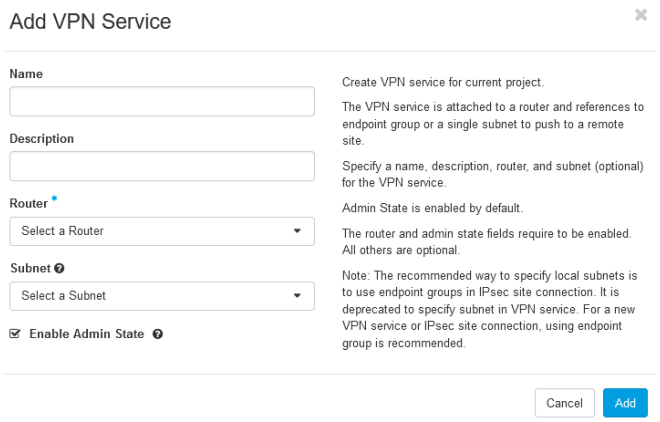


Рис. 7-6: Создание службы VPN

Чтобы добавить VPN с помощью клиента Openstack, см. следующий пример:

```
itopenstack my_cloud vpn service create vpn --router my-router
```

7.1.6.5 Создание групп конечных точек (EPG)

Чтобы создать группу конечных точек, в окне VPN на четвертой вкладке щелкните кнопку `Add endpoint Group`. Для каждого VPN-соединения необходимо создать локальную EPG и одноранговую EPG. Локальная EPG определяется локальной подсетью (подсетями), а одноранговая EPG определяется адресом CIDR одноранговой подсети (подсетей) (т.е. CIDR подсети на другой ICP). Вы можете выбрать одну или несколько подсетей.

Заполните необходимую информацию:

- Name: имя группы EPG.
- Description: описание группы EPG.
- Type: тип группы EPG:
    - выберите subnet, чтобы задать локальную EPG;
    - выберите CIDR, чтобы задать одноранговую EPG;
    - добавьте подсеть или CIDR.
- External System CIDR или Local System Subnet: CIDR внешней системы или подсеть локальной системы.
    - Имя этого поля зависит от выбранного на предыдущем шаге типа группы EPG.
    - При выборе подсети локальной системы: выберите локальную подсеть или подсети, которые будут использоваться в VPN-подключении.
    - При выборе CIDR внешней системы: введите адрес CIDR удаленной подсети или подсетей, которые будут использоваться в VPN-подключении.

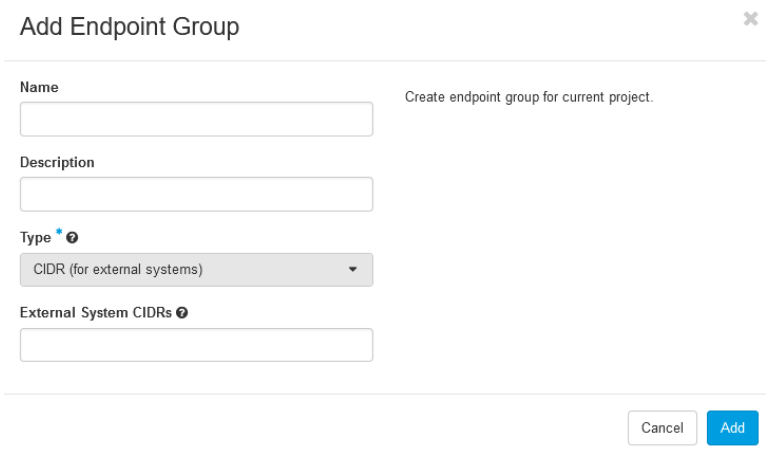


Рис. 7-7: Создание группы конечных точек

Чтобы создать группы EPG с помощью клиента Openstack, см. следующий пример:

```
itopenstack my_cloud vpn endpoint group create --type subnet --value my_subnet
my_local_endpoint
itopenstack my_cloud vpn endpoint group create --type cidr --value 10.1.0.0/24
peer_endpoint
```

В этом примере создается локальная группа EPG с использованием значения my_subnet и одноранговая EPG при условии, что удаленная подсеть имеет адрес CIDR 10.1.0.0/24.

7.1.6.6 Создание подключения к узлу IPsec

Чтобы создать подключение к узлу VPN, в окне VPN на пятой вкладке щелкните кнопку `Add IPsec Site Connection`. Соединение с узлом использует все ранее настоенные сущности. Для успешного подключения потребуется указать внешний IP-адрес шлюза удаленного маршрутизатора ICP, который используется для VPN-подключения.

Заполните необходимую информацию:

- Name: имя подключения к узлу.
- Name: описание подключения к узлу.
- VPN service associated with this connection: служба VPN, которая была определена на предыдущих шагах.
- Endpoint group for local subnet(s): локальная EGP для локальной подсети, которая была определена на предыдущих шагах.
- IKE policy associated with this connection: политика IKE, определенная на предыдущих шагах.
- IPsec policy associated with this connection: политика IPsec, определенная на предыдущих шагах.
- Peer gateway public IPv4/IPv6 Address or FQDN: общедоступный IP-адрес одноранговой шлюза или полное доменное имя:
    - IP-адрес или полное доменное имя внешнего шлюза удаленного маршрутизатора ICP, используемого для установления VPN-подключения.
- Peer router identity for authentication (Perr ID): идентификатор одноранговой маршрутизатора для аутентификации:
    - IP-адрес или полное доменное имя шлюза одноранговой маршрутизатора.
- Endpoint group for remote peer CIDR(s): удаленная EGP для удаленного CIDR, который был определен на предыдущих шагах.
- Remote peer subnet(s): удаленные одноранговые подсети – устаревший параметр, не заполняется, поскольку используется концепция групп EPG.
- Pre-Shared Key (PSK) string: предварительно опубликованный ключ для VPN-подключения:

- пароль для установления VPN-подключения (должен быть одинаковым на обеих платформах ICP в VPN-подключении).

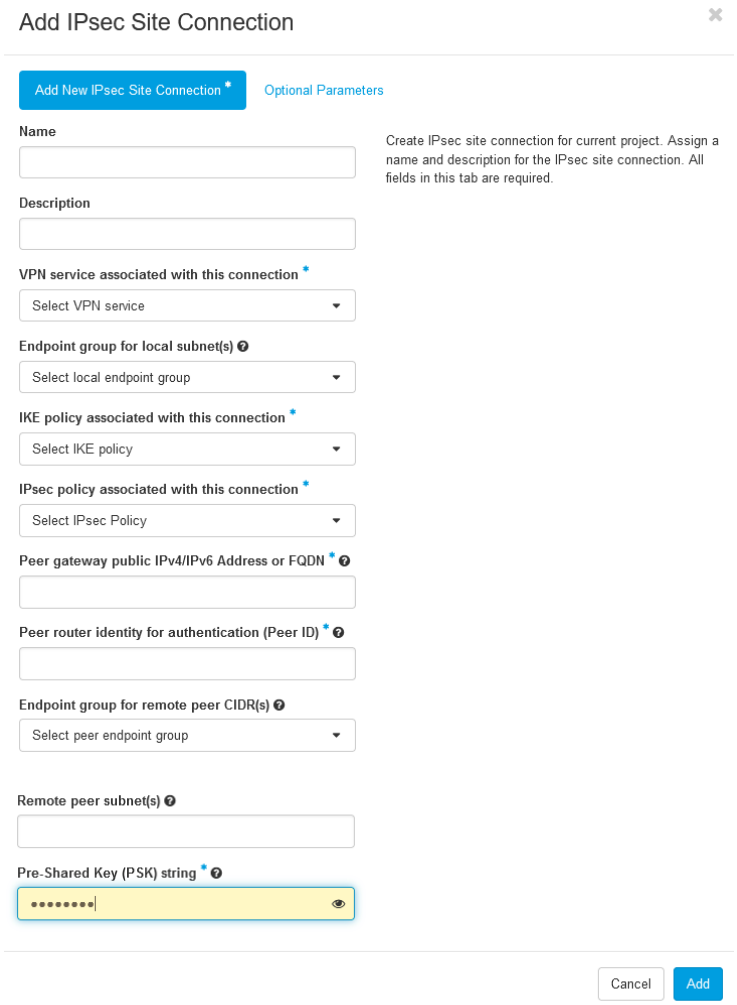


Рис. 7-8: Создание подключения к узлу

Чтобы добавить подключение к узлу VPN с помощью клиента Openstack, см. следующий пример:

```
vpn ipsec site connection create --vpnservice my_vpn_service --ikepolicy myikepolicy –
myipsecpolicy ipsecpolicy --local-endpoint-group my_local_endpoint --peer-address
192.168.172.12 --peer-id 192.168.172.12 --peer-endpoint-group peer_endpoint --psk
my_password my_connection
```

Таким образом, инициируются попытки подключиться к одноранговому узлу и установить VPN-подключение.

Как отмечалось в начале этого раздела, эти шаги необходимо повторить и на одноранговой платформе ICP. После того, как на обеих платформах ICP созданы подключения к узлу VPN, подключение устанавливается, и подключения к узлу становятся активными.

Теперь можно передавать сетевой трафик между локальными и удаленными (одноранговыми) подсетями на платформах ICP через зашифрованное VPN-подключение.

7.1.7 Настройка модуля Neutron для нескольких физических сетей

В модуле Neutron можно настроить несколько базовых физических сетей (physnets). Каждая физсеть использует собственную связку интерфейсов или физический интерфейс, поэтому виртуальные сети Openstack могут быть разделены и изолированы через базовые подсети.

Чтобы указать несколько физсетей, установите назначение provider на несколько интерфейсов или связок. Пример:

```
networking_common:
  bonds:
    cspexternal:
       members:
           - eno1
       mode: active-backup
       ip_start: 192.168.27.11/24
       gateway: 192.168.27.1
       purpose:
           - external_api
    cspinternal:
       members:
           - eno2
       ip_start: 192.168.157.11/24
       mode: active-backup
       purpose:
           - internal_api
    cspstorage:
       vlan_id: 500
       members:
           - eno3
       mode: active-backup
       vlan_parent: storageraw
       ip_start: 172.30.2.11/24
       purpose:
           - storage
           - cluster
    cspprovider1:
       members:
           - eno4
       mode: active-backup
       ip_start: 192.168.22.11/24
       gateway: 192.168.22.1
       purpose:
           - provider
    cspprovider2:
       members:
           - eno7
       mode: active-backup
       ip_start: 192.168.23.11/24
       gateway: 192.168.23.1
       purpose:
           - provider
```

Эта конфигурация будет использовать связки cspprovider1 и cspprovider2 для создания базовых физических сетей physnet1 и physnet2.

На втором этапе нужно разрешить использование диапазонов VLAN для обеих физических сетей.

Файл: neutron/ml2_conf.ini.

Расположение файла: папка, указанная параметром cloud_custom_config_dir в основном файле конфигурации (обычно */opt/cloud-cfg/<cloud-name>/config*).

Инструкции по редактированию: добавьте параметр network_vlan_ranges в раздел ml2_type_vlan:

```
[ml2_type_vlan]
network_vlan_ranges: physnet1:low_vlan:high_vlan, physnet2:low_vlan:high_vlan, …
```

Пример:

```
[ml2_type_vlan]
network_vlan_ranges: physnet1:600:800, physnet2:600:900
```

7.1.8 Включение сети OVN

Открытая виртуальная сеть OVN (Open Virtual Network) дополняет существующие возможности OpenvSwitch, добавляя встроенную поддержку абстракций виртуальной сети. Она заменяет плагин OVS ML2 по умолчанию и некоторые агенты Neutron собственными агентами OVN. После развертывания сети OVN некоторые контейнеры на основе OVS на узлах также заменяются контейнерами OVN.

Из-за особых требований виртуальных сетей OVN, вместо протокола VxLAN, используемого в OVS, используется туннельный протокол Geneve.

Чтобы включить сеть OVN, используйте следующую строку в конфигурации yaml:

```
neutron_plugin_agent: "ovn"
```

Также можно включить агент DHCP на основе OVN вместо агента по умолчанию, предоставляемого компонентом neutron:

```
neutron_ovn_dhcp_agent: "yes"
```

Чтобы включить распределенное управление плавающими IP-адресами, используйте следующую строку (в этом случае сетевой трафик, использующий плавающие IP-адреса и NAT, не должен проходить через управляющие или сетевые узлы):

```
neutron_ovn_distributed_fip: "yes"
```

Рекомендуется установить для всех параметров значение yes.

Предупреждение!

DVR (распределенный виртуальный маршрутизатор) обычно используется для обозначения конкретной реализации распределенных маршрутизаторов, предоставляемых агентом Neutron L3. Агент Neutron L3 в режиме DVR не совместим с OVN. Необходимо отключить службу neturon dvr, так как маршрутизация L3 всегда осуществляется распределенно, если используется встроенная в OVN поддержка L3.

```
enable_neutron_dvr: "no"
```

Предупреждение!

В настоящее время развертывание сети OVN не совместимо с развертыванием DPDK.

7.1.9 Включение службы резервирования ресурсов blazar

Blazar — это сервис OpenStack, обеспечивающий резервирование ресурсов в облаке для различных типов ресурсов — как виртуальных (экземпляры, тома, стеки), так и физических (узлы).

Предупреждение!

Из-за наличия специфичного бага в сервисе blazar, с этим сервисом совместима только конфигурация без сертификатов TLS.

Чтобы включить blazar, используйте следующую строку в конфигурации yaml:

```
enable_blazar: 'yes'
```

Для резервирования хостов и виртуальных ресурсов можно использовать панель инструментов или клиент OpenStack.

При использовании клиента OpenStack, используйте команды:

```
itopenstack cloud_name reservation floatingip create
```

```
itopenstack cloud_name  reservation floatingip delete
itopenstack cloud_name  reservation floatingip list
itopenstack cloud_name  reservation floatingip show
itopenstack cloud_name  reservation host create
itopenstack cloud_name  reservation host delete
itopenstack cloud_name  reservation host list
itopenstack cloud_name  reservation host property list
itopenstack cloud_name  reservation host property set
itopenstack cloud_name  reservation host property show
itopenstack cloud_name  reservation host set
itopenstack cloud_name  reservation host show
itopenstack cloud_name  reservation lease create
itopenstack cloud_name  reservation lease delete
itopenstack cloud_name  reservation lease list
itopenstack cloud_name  reservation lease set
itopenstack cloud_name  reservation lease show
```

7.2 Файлы конфигурации мониторинга

Файл конфигурации мониторинга находится в пакете *itobservability*, который инсталлируется в процессе установки сервера COS (см. главу 4). Если вы хотите установить его самостоятельно, выполните команду:

```
apt install itobservability=<version>
```

Файл конфигурации мониторинга *alert.rules* находится в папке */opt/it_observability/prometheus-cfg/prometheus_server/rule*. Скопируйте его в папку конфигурации */opt/cloud-cfg/<cloud-name>/config/prometheus*.

8 Развертывание облака

Предварительные условия:

- рабочий сервер репозиториев;
- имена серверов репозиториев, вставленные на сервер DNS;
- рабочий сервер управления облаком;
- основной файл конфигурации;
- дополнительные файлы конфигурации;
- узлы ICP, подключенные к сети.

8.1 Копирование файлов конфигурации в папку cloud-cfg

Скопируйте файлы конфигурации, подготовленные в предыдущих главах, на сервер управления облаком в папку */opt/cloud-cfg*. Каждое облако должно иметь следующую структуру каталогов:

/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml

/opt/cloud-cfg/<cloud-name>/config/<additional-configuration-files>

где:

<cloud-name>-big.yml — это основной файл конфигурации, подготовленный в главе 6.

<additional-configuration-files> — это все остальные файлы конфигурации, подготовленные в главе 7.

8.2 Подготовка сервера управления облаком к развертыванию

Выполните следующие команды:

```
itkf-setup-nat.sh <deployment-interface-name> <uplink-interface-name>

itkf-setup-nfsroot.sh
```

8.3 Подготовка среды развертывания

Запустите скрипт *itkf-deployment-prepare.sh* с файлом yaml, который был подготовлен в главе 6:

```
cd /opt/cloud-cfg/<cloud-name>

itkf-deployment-prepare.sh <cloud-name>-big.yml
```

Предупреждение!

Некоторые файлы в каталоге /srf/fai/config будут перезаписаны. Если вы внесли изменения в эти файлы и хотите применить изменения, используйте опцию -u (--update).

Будут созданы файлы *globals.yml*, *passwords.yml* и *cloud-inventory*, а также скопированы дополнительные файлы конфигурации в папку */opt/etc-kolla/<cloud-name>*. Эта папка также сопоставляется с контейнером *kolla-ansible-srv* как */etc/kolla/cloud-cfg/<cloud-name>/*.

Предупреждение!

Эта процедура перезапишет некоторые файлы (если они уже существуют в папке *etc*) новыми шаблонами из пакета. Это означает, что любые изменения в файлах *globals.yml*

и *cloud-inventory* (до запуска скрипта подготовки itkf-deployment) будут утеряны. Если вы хотите, чтобы эти файлы не менялись, используйте опцию –u (--update).

Однако вы можете явно установить параметр перезаписи, используя параметр –o.

Файл *passwords.yaml* в папке *etc* останется без изменений, даже если параметр –u не используется.

Чтобы перезаписать также файл *passwords.yaml* в папке *etc* новым шаблоном, используйте следующую команду:

```
itkf-deployment-prepare.sh -y <cloud-name>-big.yml
```

Предупреждение!

Переключатель -y используется для перезаписи *passwords.yml* , поэтому любые изменения, внесенные в этот файл, будут утеряны. Это означает, что сервер не сможет управлять облаком, развернутым с перезаписанным файлом *passwords*. Поэтому, если вы хотите сохранить пароли (обычно это происходит после обновления сервера управления облаком), не используйте ключ –y.

Предупреждение!

Будет создана папка */opt/etc-kolla/<cloud-name>*. Не меняйте ее вручную.

8.4 Подготовка сертификата TLS

Предупреждение!

При установке параметра `kolla_enable_tls_external: 'yes'` и/или `kolla_enable_tls_internal: 'yes'`, не забудьте предоставить сертификаты TLS:

- сертификаты, самоподписанные с помощью скрпита *itkf-generate-certificates.sh*;
- прямо указанный подписанный сертификат.

В противном случае развертывание облака будет безуспешным!

Файлы *haproxy.pem* и *haproxy-internal.pem* нужно скопировать в карту контейнера *kolla-ansible-srv* – в каталоге */etc/kolla/cloud-cfg/${CLOUD_NAME}/certificates*.

8.5 Подготовка файла лицензии

Скопируйте файл лицензии в папку, указанную в файле */opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml* (параметр license_file). Настоятельно рекомендуется, чтобы файл находился в каталоге */opt/cloud-cfg/<cloud-name>/*. В файле лицензии указывается, сколько узлов можно развернуть в облаке. Сам файл должен быть получен от Искра Технологии.

8.6 Загрузка хостов и установка операционной системы

Убедитесь, что все хосты имеют правильные параметры загрузки в соответствии с режимом загрузки, установленным в файле развертывания yaml (uefi, pxe или https), и порядком загрузки с правильным интерфейсом Ethernet в первую очередь.

Когда все хосты загружены, вы можете деактивировать действие загрузки (чтобы предотвратить любую дополнительную нежелательную загрузку сети):

```
itkf-deployment-boot-stop.sh  <cloud-name>-big.yml
```

Через некоторое время (около 10–20 минут) операционная система должна установиться. Чтобы проверить, прошла ли установка успешно, вы можете выполнить команду (если используете внешнюю сеть API для установки модуля kolla):

```
itkf-deployment-pre-check.sh  <cloud-name>-big.yml
```

Если вы используете внутреннюю сеть API для установки модуля kolla, используйте следующую команду:

```
itkf-deployment-pre-check.sh -i <cloud-name>-big.yml.
```

Чтобы проверить, правильно ли подключены все сети, вы можете запустить инструмент проверки сети:

```
itkf-maintenance-verify-net.sh cloud.yml
```

Если вы хотите получить более подробную информацию, то есть узнать, какие сети являются проблемными, вы можете запустить скрипт:

```
itkf-maintenance-verify-net.sh –d cloud.yml
```

Эта команда может выполняться дольше, чем предыдущая.

8.7 Проверка готовности хостов

Прежде чем приступить к фактическому развертыванию облака, вы можете проверить файлы журналов процесса подготовки хостов. Их можно найти на сервере управления облаком в каталоге:

`/root/node-name`

где node-name — это полное имя узла, сгенерированное из файла конфигурации yaml для развертывания облака.

В этом каталоге вы можете найти журналы всех развертываний, а в каталоге `last` представлена последняя попытка развертывания.

В этом каталоге проверьте следующие файлы на наличие фатальных ошибок или сбоев:

- `error.log`
- `shell.log`
- `python.log`

Вы также можете проверить файл `fai.log`, где собраны все журналы.

Если обнаружены какие-либо фатальные ошибки, проверьте файл конфигурации развертывания yaml и этап подготовки процедуры развертывания.

8.8 Удаление файлов развертывания (необязательно для PXE)

Если вы используете вариант загрузки PXE, то некоторые файлы создаются в каталоге:

`/srv/tftp/fai/pxelinux.cfg`

Все файлы, кроме `default` и `stretch.tmlp`, можно удалить.

8.9 Запуск развертывания облака

Если вы используете внешнюю сеть API для установки модуля kolla, запустите скрипт:

```
itkf-deployment-start.sh <cloud-name>-big.yml
```

Если вы используете внутреннюю сеть API для установки модуля kolla, используйте следующую команду:

```
itkf-deployment-start.sh -i <cloud-name>-big.yml
```

Предупреждение!

Если файл лицензии отсутствует или нужно развернуть больше узлов, чем указано в файле лицензии, развертывание облака будет неудачным!

8.10 Проверка статуса компонентов облака

Примечание.

Модуль Horizon в первую очередь тестируется и поддерживается в последней версии браузера Firefox и последней версии браузера Chrome. Таким образом, рекомендуется использовать один из этих браузеров, чтобы избежать ошибок и отсутствия поддержки некоторых функций.

Примечание.

Следующие операции должны быть выполнены внутри проекта *admin* (проекты можно переключать с помощью переключателя в левом верхнем углу веб-интерфейса).

8.10.1 Гипервизоры

Чтобы выполнить проверку работоспособности гипервизоров OpenStack:

1. Войдите в панель управления. Откройте браузер и введите *http://<external_vip_address*. Пароль для пользователя *admin* предоставляется системным администратором.
2. Выберите элемент **System Information** (**Admin** → **System** → **Hypervisors**)
3. Перейдите на вкладку **Hypervisors** (на скриншоте отмечена стрелкой):

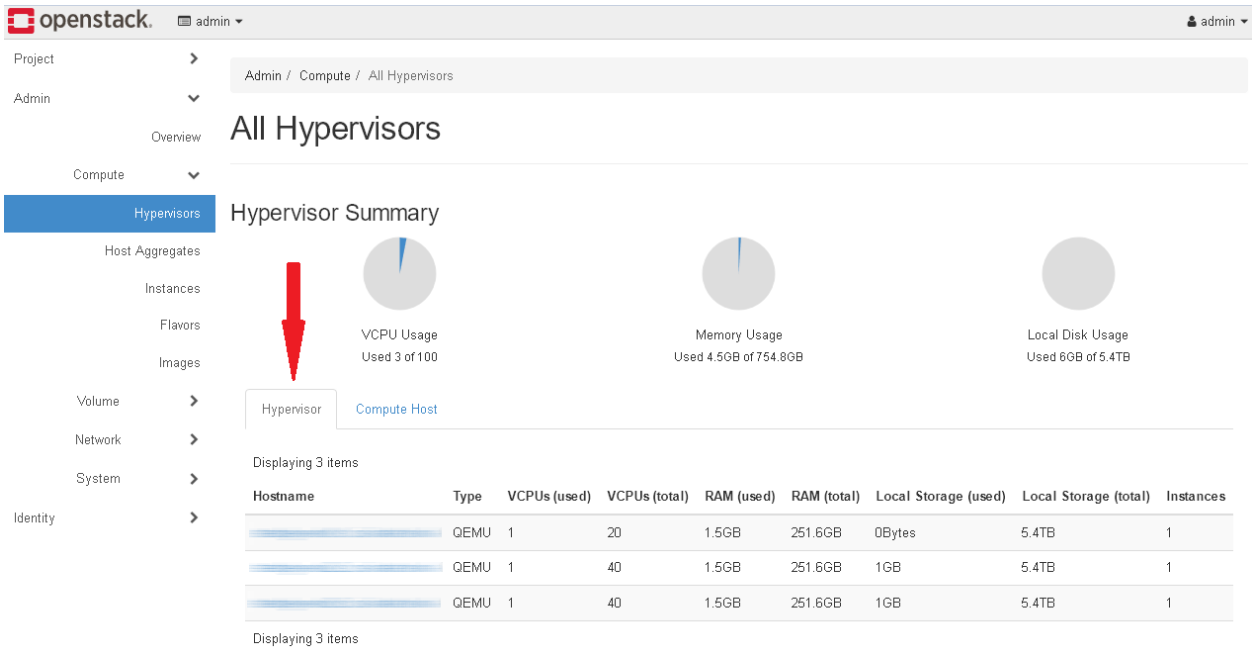


Рис. 8-1: Проверка гипервизоров Openstack

8.10.2 Вычислительные службы

Чтобы выполнить проверку работоспособности служб OpenStack Compute:

1. Войдите в панель управления. Откройте браузер и введите *http://<external_vip_address*. Пароль для пользователя *admin* предоставляется системным администратором.
2. Выберите элемент **System Information** (**Admin → System → System Information**)
3. Перейдите на вкладку **Compute Services** (на скриншоте отмечена стрелкой):

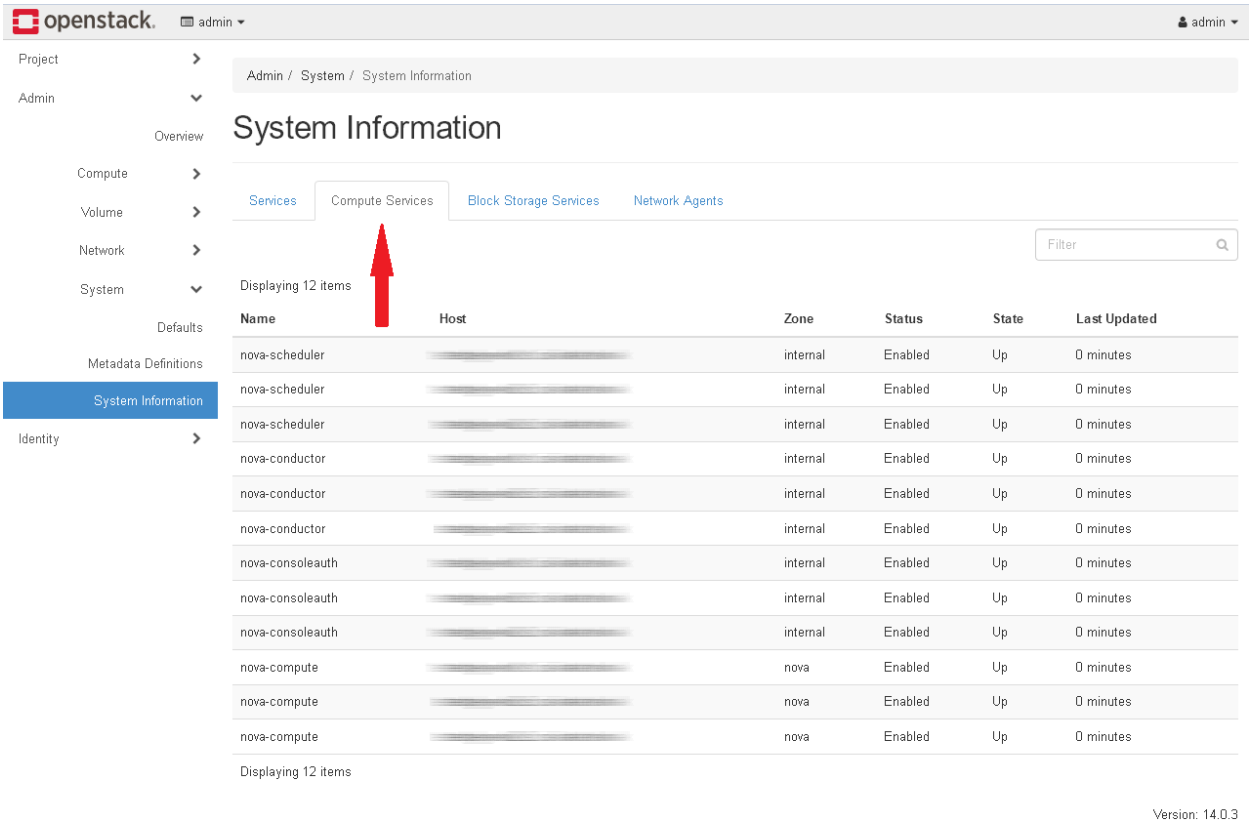


Рис. 8-2: Проверка служб Openstack Compute

8.10.3 Службы блочного хранилища

Чтобы выполнить проверку работоспособности служб блочного хранилища OpenStack:

1. Войдите в панель управления. Откройте браузер и введите *http://<external_vip_address*. Пароль для пользователя *admin* предоставляется системным администратором.
2. Выберите элемент **System Information** (**Admin → System → System Information**)
3. Перейдите на вкладку **Block Strorage Services** (на скриншоте отмечена стрелкой):

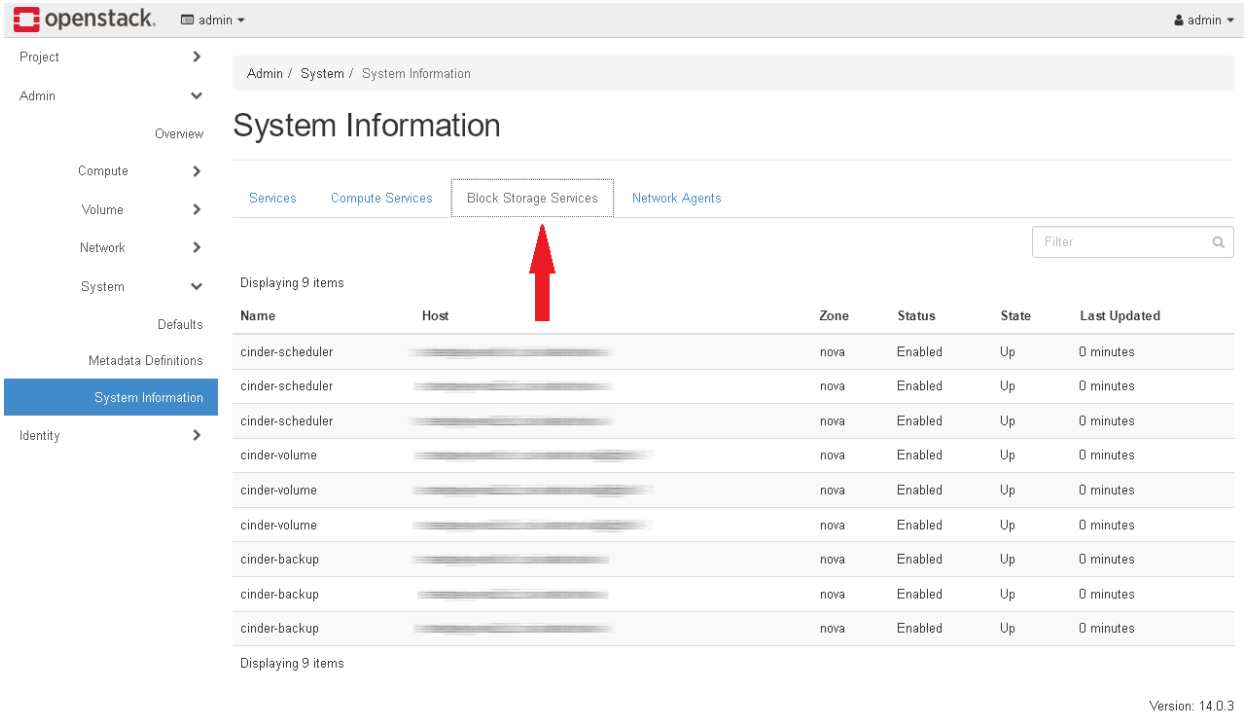


Рис. 8-3: Проверка служб блочного хранилища OpenStack

8.10.4 Сетевые агенты

Чтобы выполнить проверку работоспособности сетевых агентов:

1. Войдите в панель управления. Откройте браузер и введите *http://<external_vip_address*. Пароль для пользователя *admin* предоставляется системным администратором.
2. Выберите элемент **System Information** (**Admin** → **System** → **System Information**)
3. Перейдите на вкладку **Network Agents** (на скриншоте отмечена стрелкой):

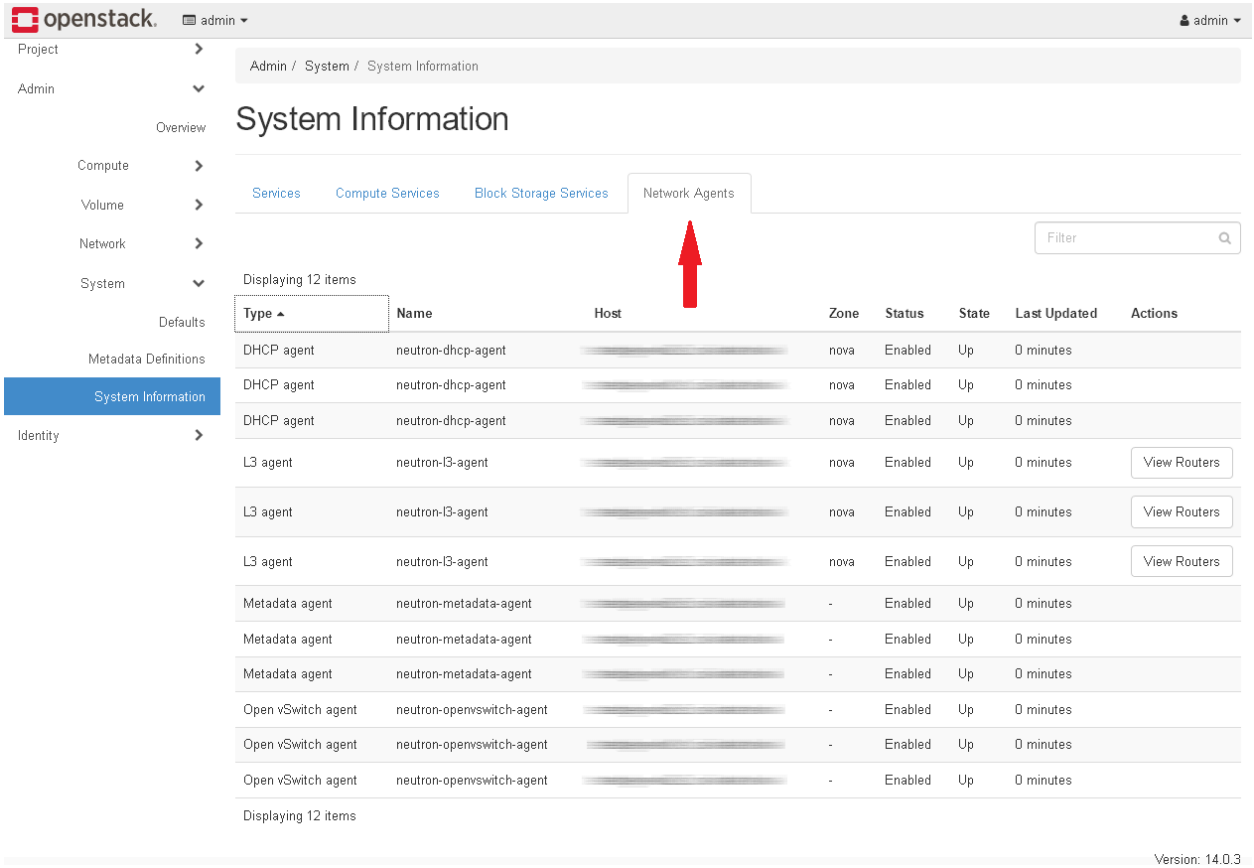


Рис. 8-4: Проверка сетевых агентов Openstack

9 Сервер авторизации FreeIPA

9.1 Сервер IPA на сервере COS

9.1.1 Настройка сервера IPA

Настройка сервера IPA выполняется в рамках настройки операционного сервера COS (см. главу 4). Основные параметры задаются в файле cloud_config.yml в разделе параметров сервера IPA.

9.1.2 Клиенты IPA

Клиентами IPA являются:

- сам сервер COS – если для параметра *cos_is_ipa_client* в файле конфигурации сервера COS (*cos_config.yml*) установлено значение *yes.*
- виртуальная машина сервера IPA – всегда.
- ве остальные виртуальные машины COS – если для параметра <*cos_vm*>*.ipa_client* в файле конфигурации сервера COS (*cos_config.yml*) установлено значение *yes.*
- все хосты ICP – если для параметра *ipa_integration_enabled* в основном файле конфигурации облака (big yaml) установлено значение *yes.*

Все эти хосты автоматически добавляются на сервер IPA при выполнении процедур настройки сервера COS (см. главу 4). Имена хостов и их IP-адреса также вставляются в базу данных IPA DNS. Процедура настройки хостов ICP описана в разделе 9.1.4.

Дополнительные клиенты IPA (физические машины и/или виртуальные машины) могут быть добавлены позже с помощью процедур в графическом интерфейсе или командной строке (см. документацию ПО IPA) и раздел 9.2).

9.1.3 Пользователи и группы пользователей IPA

Пользователи и группы пользователей IPA по умолчанию добавляются на сервер IPA автоматически. Они должны быть определены в файле cloud_config.yml в разделе пользователей и групп пользователей IPA по умолчанию.

Дополнительные пользователи и группы могут быть добавлены и настроены позже с помощью процедур в графическом интерфейсе или командной строке (см. документацию ПО IPA).

9.1.4 Интеграция Openstack на сервер IPA

Сначала нужно настроить параметры сервера IPA в основном файле конфигурации облака (см. Настройки сервера IPA).

Предупреждение!

Параметры Ipa_admin_password, keystone_ipa_admin_user, keystone_ipa_admin_password и freeipa_groups должны соответствовать параметрам, установленным в файле cloud_config.yml.

Openstack интегрируется с сервером IPA в два этапа:

1. В качестве клиентов IPA устанавливаются узлы платформы ICP, подготавливаются файлы конфигурации модулей keystone и horizon. Этот этап выполняется как часть развертывания Openstack, если для параметра ipa_integration_enabled в основном файле конфигурации облака (big yaml) установлено значение yes (см. раздел 6.2.3).

```
itkf-deployment-prepare.sh /opt/cloud-cfg/vmcloud/vmcloud-big.yml
itkf-deployment-start.sh -i /opt/cloud-cfg/vmcloud/vmcloud-big.ymli
```

Эту же процедуру следует выполнить даже в случае реконфигурации облака.

Предупреждение!

Проверьте записи службы DNS на сервере IPA после процедуры реконфигурации облака. Иногда для облачного узла добавляются две записи A, первая – с IP-адресом узла, а вторая – с плавающим IP-адресом облака. Удалите запись A с плавающим IP-адресом. Ее можно удалить в графическом интерфейсе IPA (NetworkServices -> DNS) или с помощью командной строки на узле сервера IPA:

```
kinit admin
ipa dnsrecord-del <domain> <node-name> --a-rec=<float_ip>
```

2. Создаются домены и проекты Openstack и устанавливаются роли пользователей в этих доменах и проектах. Сначала проверьте, запущен ли контейнер *it_openstack-client* на ВМ операций.

   ```
   docker ps | grep it_openstack-client\
   ```
   Если нет, запустите его.

   Запустите скрипт:
   ```
   itka-os-ipa-admin.sh <cloud-name>
   ```

   Если процедура не удалась, перезапустите контейнеры keystone:
   ```
   itkf-maintanance-restart-service.sh –f <cloud_name> keystone
   ```

   Подождите 3 минуты и снова запустите скрипт *itka-os-ipa-admin.sh*.

9.2 Установка клиента FreeIPA

Узлы инфраструктуры Openstack можно настроить для аутентификации через сервер FreeIPA.

Для этого на каждом узле необходимо установить и настроить клиент FreeIPA.

На узле инфраструктуры Openstack выполните следующие команды:

```
$ apt update
$ apt install -y freeipa-client
```

Запустите интерактивный установщик.

Обратите внимание, что перед этим уже должен быть изменен пароль (-w):

```
$ ipa-client-install --unattended --mkhomedir -p admin -w [пароль_предоставляется_системным_администратором]
```

Повторите эту команду, если она не удалась.

После завершения процесса можно попытаться войти в систему как пользователь admin:<password> или любой другой пользователь, которого вы настроили через панель управления FreeIPA.

9.3 Устранение ошибок

9.3.1 Не удается войти в панель инструментов Openstack

Решение: вручную создать домены и проекты Openstack.

На сервере управления облаком выполните команды:

```
# Check the kolla-ansible container version
docker ps | grep kolla

# Run the kolla-ansible container
# Change VERSION to the one you have already.
docker run -it --rm -v /tmp:/tmp -v /opt:/opt vvreposerver.iskrauraltel.ru:4567/deploy/infrast
ructure/it_kolla-ansible-srv:VERSION /bin/bash
```

Внутри контейнера kolla-ansible выполните команды:

```
# Configure openstack client
# Replace CLOUD_NAME with your cloud name
source /opt/etc-kolla/CLOUD_NAME/admin-openrc.sh

# Test openstack client
openstack server list
# If you don't get an error, we're good to go

# The steps to create users are already prepared in this script:
cat /tmp/configure_openstack_database.sh

# If the domain name is ok (the one you set in variables.yaml), run the script:
/tmp/configure_openstack_database.sh
```

Когда скрипт запустится, убедитесь, что установлена переменная `gid` (вы можете увидеть вывод скрипта в терминале).

Подождите несколько минут, пока Openstack обновится, затем снова войдите в систему через графический интерфейс.

9.3.2 Перезапуск контейнеров Openstack

Если в графическом интерфейсе появляются ошибки, попробуйте перезапустить следующие контейнеры на каждом узле:

```
docker restart keystone
docker restart horizon
```

9.4 Прокси SOCKS

В этом разделе описывается создание прокси-сервера SOCKS на локальном компьютере.

Этот метод полезен, если мы хотим получить доступ к веб-сайту, находящемуся во внутренней сети.

При этом требуется доступ SSH к одному из серверов внутренней сети.

В этом примере мы будем подключаться к `https://ipa-master-go.iskrauraltel.ru`.
Веб-сайт доступен с внутреннего сетевого сервера: `testcore-node2.iskrauraltel.ru`.

На локальном компьютере откройте терминал и создайте динамический туннель SSH. Выберите свободный локальный порт, в данном случае мы используем порт `8084`.

```
ssh -CNq -D 8084 root@testcore.iskrauraltel.ru
```

Откройте браузер Firefox или Chrome, откройте настройки и найдите `proxy`.
В разделе `Proxy settings`, выберите `Manual proxy configuration` и установите для SOCKS `Host` значение `127.0.0.1`, а для `Port` – выбранный вами порт (мы использовали порт `8084`).
Также установите флажок Proxy DNS , если используется Socks v5.

Сохраните ваши настройки.

Теперь вы сможете подключиться к внутреннему веб-сайту.

С этими настройками браузер будет подключаться к Интернету через выбранный удаленный сервер (в нашем случае `testcore-node2.iskrauraltel.ru`).
Обязательно отключите настройки прокси, когда закончите (используйте параметр `No proxy`).

10 Резервное копирование и восстановление

Резервное копирование облачной инфраструктуры позволяет создавать резервные копии:

- базы данных OpenStack,
- файлов конфигурации облака,
- виртуальных машин,
- образов,
- томов.

Процедуры резервного копирования и восстановления могут выполняться:

- по запросу (с помощью команд на ВМ операций или по расписанию Bacula на ВМ резервного копирования) или
- автоматически (с помощью планировщика Bacula на ВМ резервного копирования).

На сервере COS есть две виртуальные машины, которые отвечают за систему резервного копирования: OPVM (ВМ операций) и Backup VM (ВМ резервного копирования).

ВМ операций содержит контейнер backup-toolbox и скрипты резервного копирования, которые взаимодействуют с узлами ICP и интерфейсом ICP VIM и собирают нужные данные от ICP для процедур резервного копирования и отправляют соответствующие данные на платформу ICP в случае выполнения операции восстановления.

Схема процедуры резервного копирования и восстановления показана на Рис. 10-1.

Процедура резервного копирования (показана черными стрелками на Рис. 10-1) начинается с определения свойств резервной копии (имя облака, элементы для резервного копирования, тип хранилища и т.д.) в файле конфигурации yaml и последующего запуска скрипта `itkf-maintenance-backup.sh`. Этот скрипт приказывает службе резервного копирования собрать необходимые файлы с платформы ICP и поместить их в локальное или удаленное хранилище.

Если используется локальное хранилище, необходимо предоставить достаточное свободное место на самой ВМ операций. Если используется удаленное хранилище, необходимо предоставить сервер хранения на основе NFS, который должен находиться на отдельной ВМ или сервере (на ВМ операций не может быть включена служба хранилища NFS).

Процедура восстановления (показана красными стрелками на Рис. 10-1) использует тот же файл конфигурации yaml и запускается с помощью скрипта `itkf-maintenance-restore.sh`. Этот скрипт использует службу резервного копирования для получения файлов из удаленного или локального хранилища и отправки их на платформу ICP.

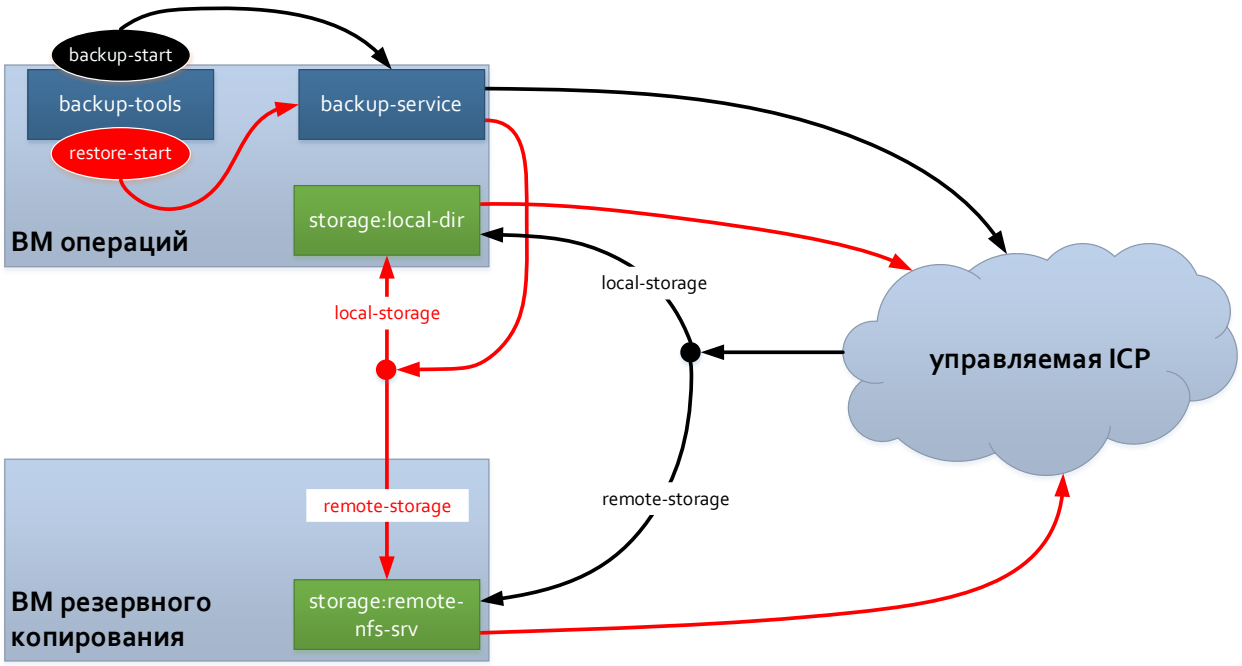


Рис. 10-1: Процедура резервного копирования и восстановления

В следующих разделах представлены подробные инструкции с необходимыми шагами для настройки процедур резервного копирования и восстановления.

Предупреждение!

Платформа ICP состоит из нескольких сущностей, которые могут быть связаны между собой. С другой стороны, интерфейс VIM имеет некоторые специфические ограничения, которые также проявляются в ограничениях службы резервного копирования и восстановления. Эти ограничения приведены в Табл. 10-1.

Табл. 10-1: Ограничения резервного копирования и восстановления

Ограничение	**Затрагиваемые сущности**	**Описание**
Идентификаторы томов и образов не сохраняются при восстановлении.	тома, образы	Во время процедуры восстановления образы и тома не собираются заново. Вместо этого, из-за ограничений интерфейса OpenStack VIM, создаются новые тома и образы и затем заменяются старыми. Эти тома и образы имеют новые уникальные идентификаторы, которые следует учитывать при использовании платформы ICP после процедуры восстановления. Это особенно важно, если какие-либо внешние инструменты управления используют идентификаторы томов и образов.
Тома, образы и виртуальные машины с одинаковым именем (относительно всей ICP) в	образы, тома, виртуальные машины	Это ограничение говорит само за себя, при этом одни и те же имена сущностей не видны пользователям, не

ссылках не могут быть указаны по имени.		являющимся администраторами, поскольку это ограничение применяется ко всей ICP. Если вы укажете имя сущности, и это имя дублировано, вы получите ошибку во время процедуры резервного копирования. В этом случае используйте идентификатор для ссылки на объект, поскольку идентификаторы уникальны на всей ICP.
Нет спецификаций метаданных ресурсов при резервном копировании и восстановлении	тома, образы	На данный момент спецификации метаданных ресурсов образов и томов не сохраняются и не восстанавливаются в резервных копиях.
Множественно прикрепленные тома	тома	На данный момент множественно прикрепленные тома повторно не прикрепляются при восстановлении.
Базово прикрепленные тома не восстанавливаются как базово прикрепленные.	тома	Тома, являющиеся базой для виртуальной машины с резервным томом, не заменяются, а добавляются к существующим томам. Поэтому можно использовать этот новый том для создания виртуальной машины с резервным томом. С другой стороны, можно использовать резервное копирование/восстановление виртуальных машин с резервным томом для автоматического повторного прикрепления базовых томов.
Можно создавать резервные копии только «доступных» и «используемых» томов.	тома	VIM запрещает резервное копирование томов, которые находятся в состоянии, отличном от available, in-use и error.
Резервное копирование не выполняется, если квоты проекта недостаточны.	тома, образы, виртуальные машины	Во время процедуры резервного копирования создается несколько временных сущностей, поэтому должны быть установлены квоты, разрешающие создание этих промежуточных сущностей. Как правило, квота должна позволять создавать дубликат самой большой сущности (с точки зрения занимаемого места на диске).
Несколько операций резервного копирования/восстановления выполняются только последовательно.	тома, образы, виртуальные машины	Несколько операций резервного копирования и восстановления могут запущены одновременно, но затем они выполняются последовательно в случайном порядке. Это предотвращает какое-либо повреждение данных резервных копий.

10.1 Развертывание сервера NFS (необязательно)

Резервная копия инфраструктуры может занимать много места, поэтому удобнее хранить файлы на удаленном сервере NFS. Если вы хотите использовать удаленный сервер NFS, можно

использовать существующий, то есть NAS (сетевое хранилище), или же можно развернуть его самостоятельно. Если вы собираетесь использовать только локальное хранилище, см. следующий раздел. В противном случае вам необходимо настроить виртуальную машину или аппаратный сервер с достаточным объемом памяти, а также настроить и запустить сервер NFS.

Сервер NFS можно развернуть на ВМ резервного копирования, установив параметр:

```
nfs_server_deploy: “yes”
```

в конфигурации развертывания сервера COS (см. раздел 4.2.3) или в конфигурации автономного развертывания ВМ резервного копирования.

Если по какой-либо причине сервер NFS не был развернут автоматически с помощью описанных выше процедур, вы можете вручную установить NFS на ВМ резервного копирования, следуя приведенным ниже инструкциям.

Запустите контейнер сервера NFS с помощью следующей команды:

```
it-storage-nfs-start.sh -r registry:port
```

Сервер поддерживает NFSv3 и NFSv4.

При желании вы можете выбрать расположение базы данных и расположение контейнера в репозитории с помощью следующих настроек:

```
-d database location directory (default is /opt/bacula-data/)
-p image path (default is /deploy/infrastructure/)
-v version of nfs container (default is the same as deb package version)
```

Например:

```
it-storage-nfs-start.sh -r k-vm-repo-server.docker.iskrauraltel.ru:4567
```

Если вы хотите остановить контейнер сервера NFS, выполните следующую команду:

```
it-storage-nfs-stop.sh
```

Предупреждение!

На сервере или виртуальной машине, где работает сервер NFS, необходимо удалить приложение **rpcbind**, поскольку оно конфликтует с протоколом NFS контейнерного сервера NFS.

Примечание.

Пожалуйста, убедитесь, что открыты следующие порты, которые позволяют серверу NFS обмениваться данными со своими клиентами:

- NFS_PORT. Порт прослушивания rpc.nfsd: 2049
- NFS_PORT_MOUNTD, только для NFSv3. Порт прослушивания rpc.mountd: 32767
- NFS_PORT_STATD_IN, только для NFSv3. Порт прослушивания rpc.statd: 32765
- NFS_PORT_STATD_OUT, только для NFSv3. Исходящий порт rpc.statd: 32766

Вы можете проверить, работает ли сервер NFS, выполнив, например, следующую команду:

```
mount IP:/opt/storage /my_local_dir
```

Предупреждение!

В случае использования контейнерного сервера NFS параметр storage_path: (см. таблицу ниже) не настраивается и всегда должен быть установлен как: `BakupVM_IP_address:/opt/storage`.

Этот каталог устанавливается внутри среды docker независимо от установки каталога места хранения на хосте docker. Таким образом, даже если вы задаете определенный каталог для хранения на сервере хранения с опцией -d (см. выше), путь к хранилищу всегда представляется как */opt/storage* для скриптов резервного копирования.

При указании IP-адресов сервера хранения (если ваш сервер имеет плавающий IP-адрес OpenStack) используйте плавающий IP-адрес для подключений из внешних сетей и используйте локальный IP-адрес для подключений из внутренних сетей.

`my_local_dir` — локальная точка монтирования.

10.2 Подготовка файла yaml конфигурации резервного копирования

Каждая процедура резервного копирования и восстановления определяется файлом конфигурации резервного копирования, который затем используется в качестве аргумента при выполнении процедур резервного копирования или восстановления по запросу. Этот файл должен храниться на ВМ операций. Файлов конфигурации может быть много даже для одной и той же ICP.

Этот файл yaml можно поделить на несколько разделов, представленных в **Табл. 10-2**.

Табл. 10-2: Файл конфигурации резервного копирования

Имя	**Тип**	**Описание**	**Примеры**
cloud_name:	обяз.	Название облака	'testedge'
storage_type:	обяз.	Тип хранилища, используемого для резервных копий. В настоящее время поддерживаются: nfs и local.	'local'
storage_path:	обяз.	Путь хранения; для nfs используется следующий синтаксис: IP:/nfs/directory, для локального — это абсолютный путь к целевому каталогу.	'192.168.22.22:/nfs/Backups/csp' или '192.168.22.21:/opt/storage
status_storage_path:	необяз.	Укажите, где будет находиться файл журнала состояния резервной копии. Он может быть только локальным, где указан абсолютный путь к каталогу назначения. Если он не указан, то используется storage_path.	'/opt/status-backup/
backup_level:	обяз.	Указывает уровень резервного копирования: • full, • incremental, • single	'incremental'
file_retention:	обяз.	Указывает, сколько дней файлы резервных копий остаются в хранилище. Это относится к полным и	60

		инкрементным резервным копиям. Значение по умолчанию — 60 дней.	
backup_optimization:	обяз.	Этот параметр применяется к инкрементным резервным копиям. Указывает, оптимизирован ли процесс создания резервной копии по скорости или по объему требуемого дискового пространства. Значения: «compute» или «storage». По умолчанию используется «compute».	'compute'
exclude_database:	необяз.	Введите «yes», чтобы исключить базу данных из резервного копирования/восстановления. По умолчанию установлено значение «no».	'yes'
cloud_config_strategy:	необяз.	Стратегия резервного копирования/восстановления облачной конфигурации. Возможные значения: overwrite (конфигурация перезаписывается), keep-newer (сохранять более новые файлы конфигурации, если они не существуют), none (конфигурация не копируется/не восстанавливается)	'overwrite'
image_backup_strategy:	обяз.	Выберите стратегию резервного копирования/восстановления для образов — all (резервные копии всех изображений), includelist (только образы из белого списка), excludelist (только образы не из черного списка), none (пропустить все образы)	'all'
volume_backup_strategy:	обяз.	Выберите стратегию резервного копирования/восстановления для томов — all (резервируются все тома), includelist (только образы из белого списка), excludelist (только образы не из черного списка), unattached	'all'

		(только тома, не прикрепленные к виртуальной машине), none (пропустить все тома)	
vm_backup_strategy:	обяз.	Выберите стратегию резервного копирования/восстановления для виртуальных машин — all (резервируются все виртуальные машины), includelist (только образы из белого списка), excludelist (только образы не из черного списка), none (пропустить все виртуальные машины)	'all'
image_includelist:	необяз.	Список образов в белом списке. Рекомендуемый способ — указывать образы через идентификатор, но также можно указать имя образа (если имя образа уникально и не содержит пробелов).	'image1'
image_excludelist:	необяз.	Список образов в черном списке. Рекомендуемый способ — указывать образы через идентификатор, но также можно указать имя образа (если имя образа уникально и не содержит пробелов).	'image3'
volume_includelist:	необяз.	Список томов в белом списке. Рекомендуемый способ — указывать тома через идентификатор, но также можно указать имя тома (если имя тома уникально и не содержит пробелов).	'volume1'
volume_excludelist:	необяз.	Список томов в черном списке. Рекомендуемый способ — указывать тома через идентификатор, но также можно указать имя тома (если имя тома уникально и не содержит пробелов).	'volume3'
vm_includelist:	необяз.	Список ВМ в белом списке. Рекомендуемый способ — указывать ВМ через идентификатор, но также	'vm1'

		можно указать имя ВМ (если имя ВМ уникально и не содержит пробелов).	
vm_blacklist:	необяз.	Список ВМ в черном списке. Рекомендуемый способ — указывать ВМ через идентификатор, но также можно указать имя ВМ (если имя ВМ уникально и не содержит пробелов).	'instance7'

10.2.1 Определение имени облака

Укажите имя облака для выполнения резервного копирования и восстановления.

Пример:

```
cloud_name: 'bb-8'
```

Для указанного облака на сервере должен быть каталог с данными авторизации. Обычно это каталог /opt/etc-kolla/*cloud_name*, где *cloud_name* заменяется конкретным именем облака, для которого выполняется резервное копирование. Это делается автоматически во время установки сервера COS или автономного развертывания ВМ операций, но если параметры авторизации на ICP изменены, /opt/etc-kolla/*cloud_name* также необходимо обновить.

10.2.2 Настройка хранилища

Хранилище может быть локальным (любой каталог на локальных дисках на ВМ операций) или удаленным, например, хранилище NFS (о развертывании сервера NFS для удаленного хранилища см. раздел 10.1).

Пример конфигурации локального хранилища:

```
storage_type: 'local'
storage_path: '/opt/big-space'
```

Пример удаленного хранилища:

```
storage_type: 'nfs'
storage_path: '192.168.22.22:/nfs/Backups/csp'
```

Кроме того, можно использовать параметр status_storage_path, чтобы указать расположение файла журнала состояния резервной копии. Этот параметр следует использовать при использовании типа хранилища NFS в сочетании с системой Bacula, чтобы позволить Bacula хранить файл журнала состояния в своей базе данных. Расположение может быть только локальным, при этом указывается абсолютный путь к каталогу назначения. Если он не указан, то для расположения файла журнала состояния используется storage_path.

10.2.3 Настройка общих параметров резервного копирования

Параметры backup_level определяют, как архивы резервных копий хранятся в хранилище. Установка «full» создает новый tar-архив каждый раз, когда выполняется резервное копирование. При установке «incremental» создается дельта-файл, отличающийся только от предыдущего архива резервной копии. Установка «single» создает только один каталог в хранилище архива, который заменяется каждый раз при выполнении резервного копирования.

Параметр file_retention указывает, сколько дней архивные файлы остаются в архиве. При каждом выполнении резервного копирования все файлы проверяются, а архивы резервных копий старше значения этого параметра удаляются. Значение по умолчанию — 60 (дней).

Параметр backup_optimization определяет, как хранятся инкрементные резервные копии. Если этот параметр имеет значение «compute» (по умолчанию), то последняя инкрементная резервная копия сохраняется как tar-файл и дельта-файл. Это ускоряет процесс резервного копирования и восстановления, поскольку во время процесса не требуется создание tar-файла из дельта-файла. Но при этом расходуется больше места на диске. Если значение backup_optimization равно «storage», то удаляются все неполные tar-файлы резервных копий, что экономит место, но увеличивает время создания инкрементных резервных копий.

10.2.4 Настройка стратегий резервного копирования

Для каждой сущности (образа, тома, виртуальной машины) можно настроить стратегию резервного копирования/восстановления по-отдельности:

- excludelist – черный список,
- includelist – белый список,
- all – все,
- none – никакие,
- unattached – неприкрепленные (только для томов).

Стратегия includelist предполагает список сущностей для включения в резервное копирование и восстановление, в то время как стратегия excludelist предполагает список объектов для исключения из резервного копирования и восстановления. При использовании этих стратегий список сущностей должен присутствовать в файле yaml.

Стратегия unattached используется только для томов и выполняет резервное копирование и восстановление только для автономных образов, т.е. не прикрепленных к какой-либо виртуальной машине.

Примечание.

Рекомендуемым способом является указание сущностей для резервного копирования через идентификатор, но они также могут быть указаны как имя сущности (если имя сущности уникально).

Пример стратегии includelist:

```
image_backup_strategy: 'includelist'
image_includelist:
    - imageA
    - imageB
volume_backup_strategy: 'includelist'
volume_includelist:
    - fbob123b2a1n2bfc23
    - vol3
vm_backup_strategy: 'includelist'
vm_includelist:
    - fbob123b2a1n2bfc37bob
```

Пример комбинированной стратегии:

```
image_backup_strategy: 'includelist'
image_includelist:
    - fbob123b2a1n2bfc23
    - imageB
volume_backup_strategy: 'unattached'
vm_backup_strategy: 'all'
```

10.2.5 Включение или исключение базы данных

База данных может быть включена или исключена для резервного копирования и восстановления. По умолчанию она включена.

Пример исключения базы данных:

```
exclude_database: no
```

10.2.6 Настройка резервного копирования и восстановления конфигурации облака

Файлы конфигурации облака для конкретного облака на ВМ операций (инвентарные данные, данные авторизации, пароли служб, конфигурация служб и конфигурация тестирования) также могут включены для резервного копирования и восстановления.

Имейте в виду, что эти файлы конфигурации действительно необходимы для выполнения процедуры резервного копирования, поэтому восстановление не может быть выполнено автоматически, если эта конфигурация отсутствует на целевом устройстве (в этом случае вы можете восстановить файлы вручную). Однако автоматическое восстановление может быть выполнено только в том случае, если вам только нужно восстановление предыдущих версий существующих файлов.

В дополнение к конфигурации облака на ВМ операций, все файлы конфигурации сервисов NFVI (например, контейнеры docker) также включаются в резервную копию.

Вы можете выбрать три разные стратегии для резервного копирования конфигурации облака:

- overwrite – все файлы перезаписываются;
- keep-newer – файлы более новые, чем файлы из архива, не заменяются;
- none – файлы не заменяются.

Пример:

```
cloud_config_strategy: keep-newer
```

10.2.7 Проверка файла yaml на наличие синтаксических ошибок

Пример файла yaml:

```
cloud_name: 'bb-8'
storage_type: 'nfs'
backup_level: ‘incremental’
file_retention: 60
backup_optimization: ‘compute’
storage_path: '192.168.22.22:/nfs/Backups/csp'
status_storage_path: '/opt/status-backup/'
exclude_database: no
cloud_config_strategy: keep-newer
image_backup_strategy: 'all'
image_includelist:
    - imageA
image_excludelist:
    - imageB
volume_backup_strategy: 'unattached'
volume_includelist:
    - vol1
volume_excludelist:
    - vol2
vm_backup_strategy: 'all'
vm_includelist:
    - instanceX
vm_excludelist:
    - instanceXY
    - instanceZZ
```

В этом примере мы выполняем резервное копирование облака bb-8, которое включает базу данных, конфигурацию облака, все образы, все виртуальные машины и неприкрепленные тома.

10.3 Запуск резервного копирования

Перед началом резервного копирования убедитесь, что контейнер backup-toolbox работает на ВМ операций. Обычно контейнер уже был запущен в процедурах настройки сервера COS (см. раздел 4.2.3) – если в файле cos_config.yml параметр *operations_vm.bacula_client* имеет значение *'yes'* – или в процедурах автономного развертывания ВМ операций. В противном случае выполните следующую команду, чтобы запустить контейнер backup-toolbox:

```
itfaikollaserver-start-backup-toolbox.sh [options] [tag/version]
```

Параметр tag/version является необязательным параметром. Если он не используется, версия извлекается из информации о пакете.

Вы можете использовать дополнительные параметры:

```
-c Deployment configuration (defaults to /opt/faikollaserver/deploy-args.yaml).
-r Override repo path (defaults to /deploy/infrastructure/).
-f Full path to image location
-v Verbose output.
```

Если используется репозиторий Nexus, запустите контейнер с параметром -f:

```
itfaikollaserver-start-backup-toolbox.sh –f <repo_fqdn_and_port>/ai6212ax/it_faikola-
deploy/it_backup-toolbox:<it_faikolla_deploy_version>
```

Чтобы запустить резервное копирование, используйте следующую команду:

```
itkf-maintenance-backup.sh back_spec.yaml
```

Вы можете использовать следующие параметры:

```
-n name    Name the backup (default name is timestamp in format YYYYMMDDHHMMSS)
-t timeout Time to wait (in seconds) for individual backup action to be completed (default
600s).
-f         On the fly backup execution (running execution overwrites files on the fly for
single backup level).
-l         Leave temporary files on backup storage in case of failed backup procedure.
-v verbose
```

Пример:

```
itkf-maintenance-backup.sh -n my-special-backup back_spec.yaml
```

Файл back_spec.yaml — это файл конфигурации yaml, который был подготовлен на предыдущем шаге (см. раздел 10.2).

Примечание.

При резервном копировании создаются некоторые промежуточные файлы, поэтому необходимо учитывать этот факт при планировании размера хранилища, необходимого для хранения архивов резервных копий. Как правило, сначала создаются файлы полной резервной копии, а затем они конвертируются в выбранный формат уровня резервной копии.

Единственный способ избежать этих промежуточных файлов — использовать уровень «single» резервного копирования с параметром «на лету» (-f), который напрямую переопределяет ранее существовавший архив резервных копий. Однако это приведет к созданию недействительного архива резервных копий, если в процессе резервного копирования возникнет какая-либо ошибка.

Поскольку некоторые сущности довольно велики и передача файлов в хранилище резервных копий может занять много времени, вы можете указать время ожидания для завершения резервного копирования каждой сущности. Это устанавливается с помощью опции -t, например:

```
itkf-maintenance-backup.sh -t 950 back_spec.yaml
```

По умолчанию время ожидания установлено как 600 с.

Примечание.

Фактическое время выполнения резервного копирования может варьироваться в зависимости от скорости диска и сети. Больше всего времени потребуется для резервного копирования виртуальных машин с сущностями flavours большого размера. Например, для выполнения резервного копирования одной виртуальной машины с размером диска 100 ГБ в flavours (при дисках SSD и пропускной способности сети 1 ГБ) может потребоваться более часа. Учитывайте это, особенно при настройке резервного копирования по расписанию, например, с помощью планировщика Bacula.

Когда процесс резервного копирования завершается, файлы архивируются в указанном хранилище. Формат архива зависит от уровня резервной копии:

- single – создается каталог с именем резервной копии или отметкой времени;
- full – создается tar-архив с именем резервной копии и отметкой времени, а также создается файл метаданных этого архива;
- incremental – создается дельта к предыдущему архиву и создается файл метаданных этой дельты.

Файлы метаданных (в случае полного и инкрементного резервного копирования) содержат информацию об архиве, в том числе:

- время создания архива;
- тип архива;
- ссылку на предыдущие архивы, если уровень резервного копирования инкрементный.

Облачные базы данных и облачные конфигурации хранятся в сжатом tar-архиве внутри основного архива. Образы, тома и виртуальные машины хранятся в виде файлов образов qcow2. Кроме того, каждый файл qcow2 содержит также текстовый файл (с суффиксом .meta), который содержит метаданные этого файла. Эти данные можно использовать для ручного восстановления образа, тома или виртуальной машины.

После завершения процедуры резервного копирования проверьте файл журнала `/var/log/backup-toolbox/backup-status.log` на наличие результатов резервного копирования и ошибок.

Также имеет смысл проверить платофрму ICP на наличие временных файлов (образов и томов, содержащих в имени `backup_of` или `temp`), которые могут появиться при сбое каких-либо служб ICP во время выполнения процедуры резервного копирования.

10.4 Процедура восстановления

Предупреждение!

Из-за некоторых ограничений реализации резервного копирования и восстановления (см. первую запись в Табл. 10-1) процедура восстановления сущностей облака без базы данных делает сохраненную резервную копию недействительной.

Это показано в следующем примере:

Мы устанавливаем резервное копирование образа `imageA` и тома `volumeB`, а также устанавливаем резервное копирование базы данных. У них есть идентификаторы `idIA` и `idVB` соответственно. Они хранятся в удаленном хранилище в виде файлов `idIA.qcow2` и `idVB.qcow2` (для имен файлов резервных копий используются идентификаторы). Когда мы выполняем процедуру восстановления без базы данных, на образ `imageA` и том `volumeB` ссылаются текущие идентификаторы, а воссоздаются они с новыми идентификаторами (база данных изменена) `idIX` и `idVY`. Если бы мы хотели повторить процедуру восстановления из той же резервной копии (и без восстановления базы данных), на эти образы снова будут ссылаться идентификаторы. Это может привести к

несогласованности, поскольку идентификаторы теперь – `idIX` и `idVY`, но файлы в хранилище по-прежнему имеют имена `idIA.qcow2` и `idVB.qcow2`. Процедура автоматического восстановления в этом случае не будет работать. Чтобы решить эту проблему, мы также можем выполнить восстановление базы данных, которое происходит до восстановления образов `imageA` и `imageB`. В этом случае образы `imageA` и `imageB` получают исходные идентификаторы (`idIA` и `idVB`), и поэтому они обращаются к корректным именам файлов во время выполнения процедуры автоматического восстановления.

Используйте следующую команду для запуска восстановления:

```
itkf-maintenance-restore.sh backup_name back_spec.yaml
```

backup_name — это имя резервной копии (если имя не было указано во время создания резервной копии, имя — это метка времени (дата и время).

back_spec.yaml — это файл, который вы подготовили на первом шаге.

В следующем примере резервная копия восстанавливается по дате и времени 10:25:44 09.04.2019:

```
itkf-maintenance-restore.sh 20190904102544 back_spec.yaml
```

Если мы используем введенное имя:

```
itkf-maintenance-restore.sh my-special-backup back_spec.yaml
```

Поскольку некоторые резервные копии сущностей довольно велики и передача файлов из хранилища резервных копий в облако может занять много времени, вы можете указать время ожидания для завершения восстановления каждой сущности. Это настраивается с помощью параметра -t, например:

```
itkf-maintenance-restore.sh –t 970 my-special-backup back_spec.yaml
```

По умолчанию тайм-аут установлен как 600 с.

В некоторых случаях может потребоваться частичное восстановление, т.е. восстановление только одной или нескольких виртуальных машин, образов и томов. В этом случае вы можете изменить файл конфигурации резервного копирования (`back_spec.yaml` в предыдущих примерах), указав, какие сущности облака вы хотите восстановить, а затем запустив процедуру восстановления. Рекомендуется продублировать исходный файл конфигурации резервной копии и изменить дубликат, чтобы сохранить исходную конфигурацию.

В приведенном выше примере вы можете изменить конфигурацию на следующее:

```
cloud_name: 'bb-8'
storage_type: 'nfs'
storage_path: '192.168.22.22:/nfs/Backups/csp'
status_storage_path: '/opt/status-backup/'
exclude_database: yes
cloud_config_strategy: none
image_backup_strategy: 'includelist'
image_includelist:
    - imageA
image_excludelist:
    - imageB
volume_backup_strategy: 'none'
volume_includelist:
    - vol1
volume_excludelist:
    - vol2
vm_backup_strategy: 'includelist'
vm_includelist:
    - instanceX
vm_excludelist:
    - instanceXY
    - instanceZZ
```

В этом примере мы выполняем восстановление `imageA` и `instanceX` без конфигурации базы данных и облака.

Имейте в виду, что при восстановлении образов и томов меняются записи базы данных, так как образы и тома получают новые идентификаторы. Фактически это означает, что после завершения процедуры восстановления ее можно повторить из того же архива только при условии, если база данных также восстанавливается. Если у вас нет резервной копии базы данных, вы должны сделать еще одну резервную копию после процедуры восстановления, чтобы иметь функциональную резервную копию, которая позволяет автоматическое восстановление (см. также предупреждение в начале этого раздела для подробного объяснения).

Предупреждение!

Для успешного завершения процесса восстановления образов, томов и виртуальных машин из всех проектов (tenants) пользователь с правами администратора должен быть членом всех этих проектов (это можно изменить в разделе управления идентификацией модуля Horizon).

После завершения процедуры восстановления проверьте файл журнала `/var/log/backup-toolbox/restore-status.log` на наличие результатов восстановления и ошибок.

Также проверьте файл журнала `/var/log/backup-toolbox/restore-changes.log`, в котором регистрируются все постоянные изменения в облачной базе данных, которые могут произойти во время процедуры восстановления (см. ограничения в Табл. 10-1).

10.5 Использование системы Bacula для резервного копирования и восстановления

Bacula — это система архивации, которая управляет процедурами резервного копирования и восстановления и поддерживает резервное копирование по расписанию. Она состоит из клиента (Bacula File Daemon или bacula-fd), менеджера хранилища (Bacula Storage Daemon или bacula-sd), менеджера резервного копирования (Bacula Director или bacula-dir) и монитора/менеджера (bacula-console).

Процедура резервного копирования и восстановления с помощью системы Bacula показана на рисунке 9-2 с системой Bacula, развернутой на ВМ резервного копирования.

Процедура резервного копирования (показана черными стрелками на рис. 9-2) настраивается путем указания свойств резервной копии (имя облака, элементы для резервного копирования, тип хранилища и т.д.) в файле конфигурации yaml и помещения их в конкретный каталог bacula-fd на сервер ВМ операций. Процедура запускается с помощью инструмента bacula-console, который использует bacula-dir для удаленного запуска процедуры резервного копирования backup-procedure на ВМ операций. На ВМ операций служба резервного копирования backup-service используется (через скрипт `itkf-maintenance-backup.sh`) для получения необходимых файлов с платформы ICP и помещения их в локальное или удаленное хранилище ВМ операций.

Если используется локальное хранилище, необходимо выделить достаточно места на ВМ операций для этих промежуточных файлов. Затем эти файлы перемещаются в базу данных системы Bacula с помощью bacula-sd.

Если используется удаленное хранилище, необходимо предоставить сервер хранения на базе NFS, который может быть на ВМ резервного копирования (может быть на выделенной ВМ при ручной установке сервера NFS) – инструкции по установке приведены в разделе 9.1.

В случае удаленного хранилища файлы перемещаются не в хранилище Bacula, а в хранилище NFS. Однако хранилище Bacula содержит файл журнала состояния резервного копирования. Этот файл

содержит основную информацию о процессе резервного копирования, включая фактическое расположение архива резервной копии в удаленном хранилище.

Процедура восстановления с помощью системы Bacula (показана красными стрелками на рис. 9-2) должна выполняться в два этапа.

Первый шаг (обозначенный как restore-start(1) на рис. 9.2) — поместить файлы из хранилища Bacula на ВМ операций с помощью bacula-sd и bacula-dir. Эта процедура выполняется из bacula-console, которая использует bacula-dir для помещения соответствующих файлов на ВМ операций. Если используется локальное хранилище, все файлы копируются из хранилища Bacula. Если используется удаленное хранилище, то на ВМ операций копируется только файл журнала состояния с информацией о местоположении файлов архива (файлы в этом случае хранятся на удаленной локации, поэтому нет необходимости их дублировать).

Второй шаг процедуры восстановления (обозначенный как restore-start(2) на рисунке 9-2) должен быть запущен на ВМ операций. Он запускается с помощью скрипта `itkf-maintenance-restore.sh` с использованием файла конфигурации yaml из определенного каталога bacula-fd на ВМ операций. Этот скрипт использует backup-service для получения файлов из удаленного или локального хранилища и восстановления их на управляемой платформе ICP.

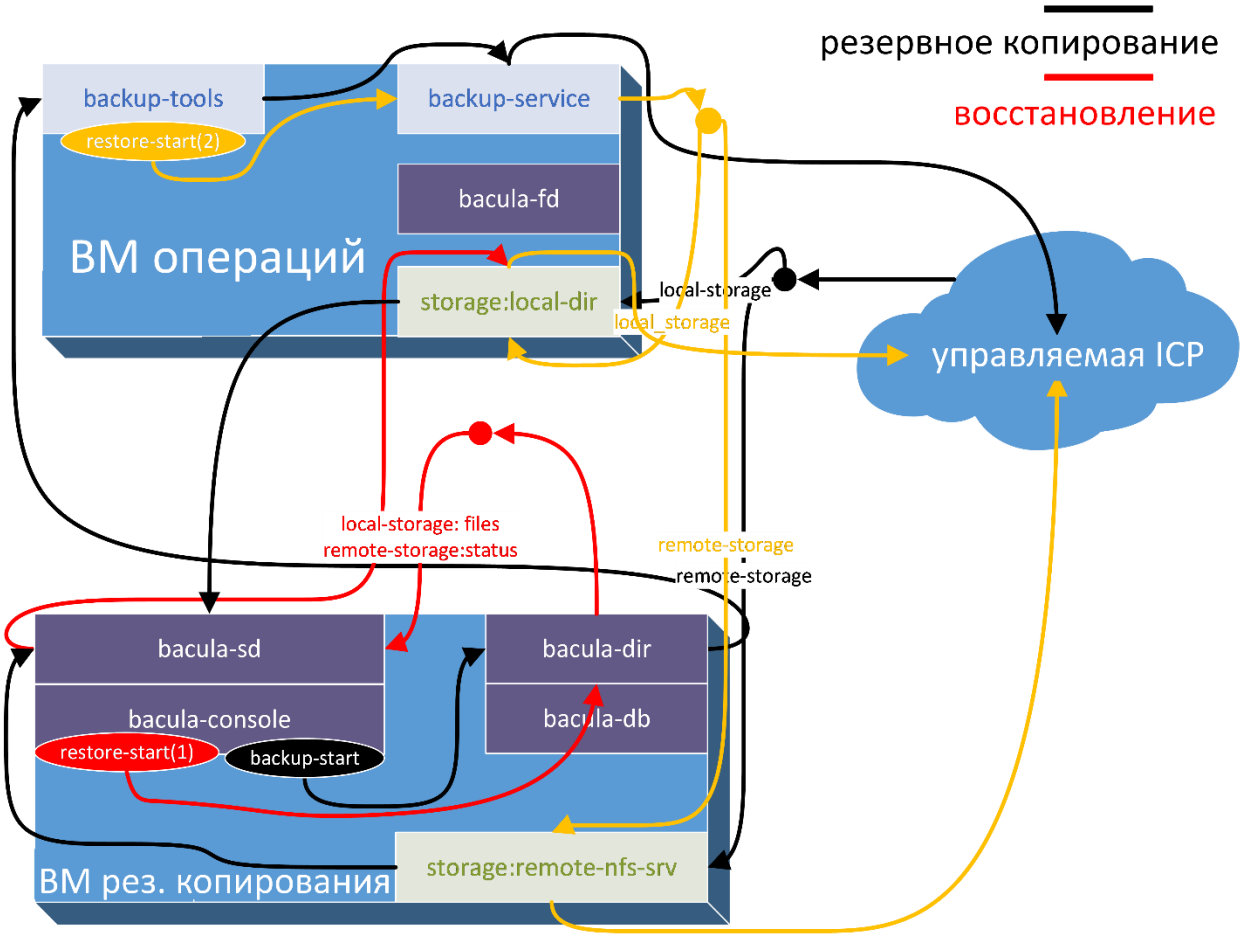


Рис. 10-2: Компоненты резервного копирования и восстановления при использовании системы Bacula

В следующих разделах представлены подробные инструкции с необходимыми шагами для выполнения процедур резервного копирования и восстановления с использованием системы Bacula.

10.5.1 Настройка клиента Bacula

Клиент Bacula автоматически устанавливается и настраивается на ВМ операций в процедурах настройки сервера COS (см. главу 3), если в файле cos_config.yml параметр *operations_vm.bacula_client* имеет значение *yes*.

Клиент Bacula также можно развернуть на автономной ВМ операций, настроив параметр:

```
openstack_client_deploy: "yes"
```

в автономной конфигурации развертывания ВМ операций.

Конфигурация клиента Bacula создается автоматически с использованием значений по умолчанию и/или конфигурации сервера COS. Конфигурация хранится в файле:

```
/opt/faikollaserver/deploy-bacula-cli-args.yaml
```

Параметры, используемые в этой конфигурации, описаны в таблице ниже.

Табл. 10-3: Файл конфигурации Bacula

Параметр	**Тип**	**Описание**
director	обяз.	имя модуля Bacula Director
monitor	обяз.	название модуля Bacula Monitor
monitor_password	обяз.	пароль для доступа к модулю Bacula Monitor
fd	обяз.	имя клиента Bacula File Daemon
fd_address	обяз.	IP-адрес, на котором работает клиент Bacula
fd_dnat_address	необяз.	внешний IP-адрес DNAT, на котором работает клиент Bacula
fd_password	обяз.	пароль для доступа к клиенту Bacula
backup_location	обяз.	расположение файлов резервных копий на клиенте Bacula
sd	обяз.	имя модуля Storage Daemon
sd_address	обяз.	IP-адрес модуля Storage Daemon
sd_dnat_address	необяз.	внешний IP-адрес DNAT, на котором работает клиент Bacula
sd_password	обяз.	пароль для доступа к модулю Bacula Storage Daemon
console_password	обяз.	пароль для доступа к консоли Bacula

Конфигурация в основном включает имена, IP-адреса и пароли для модулей Daemon системы Bacula, которые частично установлены на ВМ операций и частично на ВМ резервного копирования. Поэтому этот файл конфигурации должен быть синхронизирован между ВМ операций и ВМ резервного копирования, чтобы система Bacula работала.

В этой конфигурации указан параметр backup_location directory. Этот каталог предназначен для временного хранения подготовленных файлов резервных копий, которые затем переносятся в

хранилище данных Bacula. Этот каталог также содержит восстановленные файлы после процедуры восстановления.

Обратите внимание, что для File Daemon и Storage Daemon можно настроить IP-адрес NAT, чтобы обеспечить функциональность Bacula, даже если некоторые элементы находятся за DNAT.

Пример файла конфигурации:

```
director: infrabackup-dir
monitor: infrabackup-mon
monitor_password: [пароль_предоставляется_системным_администратором]
fd: infrabackup-fd
fd_address: 192.168.41.7
fd_dnat_address: 192.168.41.7
fd_password: [пароль_предоставляется_системным_администратором]
backup_location: /opt/bacula-b
sd: infrabackup-sd
sd_address: 17.17.2.18
sd_dnat_address: 192.168.41.90
sd_password: [пароль_предоставляется_системным_администратором]
console_password: [пароль_предоставляется_системным_администратором]
```

Если файл конфигурации изменен, изменения нужно применить, выполнив следующую команду:

```
itfaikollaserver-deploy-bacula-cli.sh
```

Этот файл используется для создания файла bacula-fd, расположенного в `/etc/bacula/bacula-fd.conf`, который используется для запуска bacula-fd на ВМ операций. Дополнительные сведения об этом файле см. в справочном руководстве по работе с ПО Bacula.

Приведенную выше команду также можно использовать для ручной установки или обновления клиента Bacula.

Этот файл конфигурации должен быть синхронизирован с файлом конфигурации, используемым для настройки сервера Bacula — см. раздел 9.5.3.

Обновление клиента Bacula производится одновременно с обновлением операционной системы на сервере ICP COS (см. раздел 3.4).

10.5.2 Настрока резервного копирования на клиенте Bacula

После настройки клиента Bacula необходимо указать, какие облака и какие элементы инфраструктуры этих облаков будут резервироваться.

Спецификации резервного копирования находятся в файле конфигурации yaml, как указано в разделе 9.2.

Рекомендуется включать как минимум базу данных и конфигурацию, а виртуальные машины, образы и тома следует указывать выборочно из-за большого объема данных, которые в этом случае необходимо передать в базу данных Bacula.

Файлы спецификаций резервных копий должны быть скопированы в каталог `/opt/bacula` на ВМ операций, чтобы система Bacula могла их использовать. В процедуру резервного копирования можно включить несколько файлов yaml (с указанием различной конфигурации резервного копирования на одной или разных ICP) – клиент Bacula ищет все файлы в каталоге `/opt/bacula` и создает все указанные резервные копии.

В качестве хранилища можно использовать локальный каталог или удаленный сервер NFS, который можно развернуть на ВМ резервного копирования, следуя инструкциям в разделе 9.1.

Предупреждение!

При создании файлов спецификаций резервных копий с удаленным хранилищем для резервных копий, нужно установить `status_storage_path` в качестве локального каталога. Этот путь должен совпадать с параметром `backup_location` в конфигурационном файле Bacula (см. Табл. 9-3). Таким образом, база хранилища Bacula будет содержать только файл состояния резервной копии, а файлы резервных копий будут находиться в удаленном хранилище.

При создании файлов спецификаций резервных копий с локальным хранилищем для резервных копий параметр `status_storage_path` не должен быть установлен или должен быть установлен на тот же путь, что и `storage_path`. Таким образом, база хранилища Bacula будет содержать файл состояния резервной копии и все файлы резервных копий.

Пример конфигурации резервного копирования с удаленным хранилищем:

```
cloud_name: 'bb-8'
storage_type: 'nfs'
backup_level: 'incremental'
file_retention: 60
backup_optimization: 'compute'
storage_path: '192.168.22.22:/nfs/Backups/csp'
status_storage_path: '/opt/bacula-b/'
exclude_database: no
cloud_config_strategy: keep-newer
image_backup_strategy: 'includelist'
image_includelist:
    - imageA
volume_backup_strategy: 'none'
vm_backup_strategy: 'includelist'
vm_wincludelist:
    - instanceX
```

Пример конфигурации резервного копирования с локальным хранилищем:

```
cloud_name: 'bb-8'
storage_type: 'local'
backup_level: 'full'
file_retention: 60
backup_optimization: 'compute'
storage_path: '/opt/bacula-b/'
exclude_database: no
cloud_config_strategy: keep-newer
image_backup_strategy: 'none'
volume_backup_strategy: 'none'
vm_backup_strategy: 'none'
```

Во время процедуры резервного копирования с помощью модуля Bacula Director файлы резервных копий временно создаются в каталоге, указанном через параметр `backup_location` в файле `/opt/faikollaserver/deploy-bacula-cli-args.yaml`. После этого эти файлы переносятся в базу данных хранилища bacula через bacula-sd и затем удаляются из каталога, указанного в `backup_location`.

Во время процедуры восстановления с помощью модуля Bacula Director восстановленные файлы передаются клиенту обратно в каталог, указанный через параметр `backup_location` в файле `/opt/faikollaserver/deploy-bacula-cli-args.yaml`.

10.5.3 Настройка сервера Bacula

Сервер Bacula автоматически устанавливается и настраивается на ВМ резервного копирования в процедурах настройки сервера COS (глава 3), если задан параметр *backup_vm* в файле cos_config.yml. Установка сервера bacula также может выполняться на автономной ВМ резервного копирования в целях тестирования.

Обновление сервера Bacula производится одновременно с обновлением операционной системы на сервере ICP COS (см. раздел 3.4).

Конфигурация сервера Bacula на ВМ резервного копирования создается автоматически с использованием значений по умолчанию и/или конфигурации сервера COS. Конфигурация хранится в файле:

```
/opt/faikollaserver/deploy-bacula-cli-args.yaml
```

Параметры, используемые в этой конфигурации, описаны в таблице 9-3.

Конфигурация в основном включает имена, IP-адреса и пароли для модулей Daemon системы Bacula, которые частично установлены на ВМ резервного копирования и частично на ВМ операций. Поэтому этот файл конфигурации должен быть синхронизирован между ВМ резервного копирования и ВМ операций, чтобы система Bacula работала.

Обратите внимание, что для File Daemon и Storage Daemon можно настроить IP-адрес NAT, чтобы обеспечить функциональность Bacula, даже если некоторые элементы находятся за DNAT.

Этот файл должен быть таким же, как на клиенте Bacula (см. раздел 9.5.1), чтобы можно было соединить клиент и сервер. Обратите внимание, что в настоящее время автоматическая настройка поддерживает только один клиент резервного копирования.

Пример файла конфигурации:

```
director: infrabackup-dir
monitor: infrabackup-mon
monitor_password: [пароль_предоставляется_системным_администратором]
fd: infrabackup-fd
fd_address: 192.168.41.7
fd_dnat_address: 192.168.41.7
fd_password: [пароль_предоставляется_системным_администратором]
backup_location: /opt/bacula-b
sd: infrabackup-sd
sd_address: 17.17.2.18
sd_dnat_address: 192.168.41.90
sd_password: [пароль_предоставляется_системным_администратором]
console_password: [пароль_предоставляется_системным_администратором]
```

Этот файл конфигурации должен быть синхронизирован с файлом, используемым для перенастройки клиента Bacula — см. раздел 9.5.1.

Если необходимо изменить конфигурацию сервера Bacula, нужно изменить этот файл конфигурации, а затем применить изменения (в то же время этот файл должен быть применен к конфигурации bacula-client).

Чтобы применить изменения файла конфигурации Bacula, нужно сгенерировать файлы конфигурации bacula-sd и bacula-dir, а затем перезапустить соответствующие контейнеры.

В следующем примере сервер Bacula перенастраивается с помощью файла конфигурации bacula_conf.yaml:

```
itbac-configure-sd.sh bacula_conf.yaml
docker restart bacula-sd
```

Этот скрипт изменяет файл `/opt/bacula-sd/bacula-sd.conf`, который также можно отредактировать, чтобы настроить дополнительные параметры конфигурации. Для дополнительной справки по параметрам, которые можно использовать в этом файле, см. справочное руководство по работе с ПО Bacula.

```
itbac-configure-dir.sh bacula_conf.yaml
docker restart bacula-sd
```

Этот скрипт изменяет файлы `/opt/bacula-dir/bacula-dir.conf` и `/opt/bacula-dir/bconsole.conf`, которые также можно отредактировать, чтобы настроить дополнительные параметры

конфигурации. Для дополнительной справки по параметрам, которые можно использовать в этих файлах, см. справочное руководство по работе с ПО Bacula.

Обновление программного пакета для сервера Bacula производится одновременно с обновлением операционной системы на сервере ICP COS (см. раздел 3.4)

Для правильной работы сервер Bacula должен состоять из следующих контейнеров:

- nfs-srv,
- bacula-app-exporter,
- bacula-exporter,
- bacula-dir,
- bacula-sd,
- bacula-db.

Если какой-либо из этих контейнеров не работает должным образом, приведеннные ниже процедуры могут помочь устранить проблемы.

Чтобы запустить службу контейнерной базы данных (Bacula Director использует базу данных для хранения своей конфигурации, в нашем случае мы используем базу данных posgresql в контейнере), выполните команды:

```
itbac-start-db.sh -r registry:port
Example:
itbac-start-db.sh -r k-vm-repo-server.docker.iskrauraltel.ru:4567
```

Чтобы правильно остановить контейнер базы данных, выполните команду:

```
itbac-stop-db.sh
```

Чтобы перенастроить и запустить bacula-sd, выполните команду:

```
 itbac-configure-sd.sh bacula_conf.yaml
```

Чтобы запустить контейнер bacula-sd, выполните команды:

```
itbac-start-sd.sh -r registry:registry_port
example:
itbac-start-sd.sh -r k-vm-repo-server.docker.iskrauraltel.ru:4567
```

Чтобы остановить контейнер bacula-sd, выполните команду:

```
itbac-stop-sd.sh
```

Чтобы перенастроить и запустить контейнерный сервис Bacula Director, выполните команду:

```
itbac-configure-dir.sh general_config.yaml
```

Чтобы запустить инициализацию базы данных (это нужно сделать только один раз после чистой установки службы Bacula Director, так как создается пустая база данных Bacula), выполните команды:

```
itbac-initialize-dir.sh -r registry:registry_port
Example:
itbac-initialize-dir.sh -r k-vm-repo-server.docker.iskrauraltel.ru:4567
```

Чтобы запустить контейнер, выполните команду:

```
itbac-start-dir.sh -r registry:registry_port
```

Пример запущенного контейнера Bacula Director:

```
itbac-start-dir.sh -r k-vm-repo-server.docker.iskrauraltel.ru:4567
```

Чтобы остановить контейнер Bacula Director, выполните команду:

```
itbac-stop-dir.sh
```

10.5.4 Использование сервера Bacula

Предупреждение!

По умолчанию автоматическая конфигурация создает расписание для указанного клиента резервного копирования (bacula-fd) с периодом в один день. Вы можете настроить расписание для ваших конкретных потребностей, используя инструкции в справочном руководстве по работе с ПО Bacula. Также обратите внимание, что резервное копирование больших виртуальных машин, образов или томов может занимать большой объем памяти и может длиться довольно долго. Поэтому отредактируйте свой планировщик в соответствии с объемом данных, который необходимо передать при резервном копировании.

10.5.4.1 Использование консоли Bacula

Консоль Bacula — это интерфейс командной строки Bacula для мониторинга и управления процедурами резервного копирования и восстановления. Он использует контейнер bacula-dir, запущенный во время развертывания сервера Bacula.

Чтобы запустить консоль Bacula, используйте следующую команду:

```
itbac-console.sh
```

Bacula предоставляет множество параметров управления и мониторинга. В этом разделе мы рассмотрим только базовое выполнение резервного копирования и восстановления. Другие параметры можно найти в справочном руководстве по работе с ПО Bacula.

- Запуск резервного копирования по требованию.

Чтобы запустить резервное копирование, выполните:

```
run
```

а затем выберите соответствующий набор файлов:

```
BackupInfrastructure
```

Подтвердите действие с помощью:

```
yes
```

Вы можете контролировать процесс резервного копирования, выполнив команду:

```
messages
```

- Запуск восстановления по требованию.

Процедура восстановления должна выполняться в два этапа. Во-первых, нужно запустить процедуру восстановления из bconsole (первый шаг):

```
restore all
```

Выберите номер последней резервной копии для клиента:

```
 5 (Select the most recent backup for a client)
```

Выберите соответствующий File Daemon:

```
infrabackup-fd
```

Выберите полный набор данных:

```
Full Set
```

Тип:

```
done
```

Подтвердите действие с помощью:

```
yes.
```

Все файлы восстанавливаются в место, указанное общей конфигурацией Bacula на клиенте Bacula - `backup_location` (см. таблицу 9-3).

Для завершения настройки необходимо вручную запустить процедуру восстановления на клиенте процедуры восстановления (второй шаг):

```
itkf-maintenance-restore.sh backup_name backup_specification.yaml
```

С помощью этой команды нужно указать имя резервной копии и файл yaml резервной копии.

Конфигурации резервного копирования для Bacula находятся на клиенте в каталоге:

```
/opt/bacula
```

В этом каталоге выберите соответствующий файл конфигурации yaml.

- Имена резервных копий — это имена каталогов на клиенте в каталоге, заданном общей конфигурацией Bacula на клиенте Bacula – `backup_location`. В этом каталоге есть отдельные каталоги для каждой резервной копии. Обычно каталог называется так же, как файл конфигурации (без суффикса yaml).

Эта информация (вместе с конкретными инструкциями по процедуре восстановления) также находится в файле резервной копии состояния для конкретной резервной копии, как настроено в файле yaml резервной копии (см. **Табл. 10-2**)

В следующем примере используется файл конфигурации /opt/bacula/my_backup.yaml и имя резервной копии my_backup (каталог с файлами называется /opt/bacula-b/my_backup):

```
itkf-maintenance-restore.sh my_backup my_backup.yaml
```

После успешного восстановления настоятельно рекомендуется удалить файлы из `backup_location`, являющиеся частью конкретной резервной копии, например, /opt/bacula-b/my_backup, потому что эти файлы будут перезаписаны при новой процедуре восстановления, запущенной из консоли Bacula.

10.5.4.2 Выбор правильного уровня резервного копирования

Как описано в разделе 9.2 и в таблице 9-2, каждый файл конфигурации резервного копирования содержит уровень резервного копирования, который может быть: одиночным (single), полным (full) и инкрементным (incremental). Независимо от этого параметра файл конфигурации bacula-dir также содержит уровень резервного копирования для каждого задания резервного копирования Bacula (дополнительную информацию см. в справочном руководстве по работе с ПО Bacula), как показано в следующем примере (часть конфигурации bacula-dir):

```
JobDefs {
  Name = "DefaultJob"
  Type = Backup
  Level = Incremental
  Client = infrabackup-fd
  FileSet = "Full Set"
  Schedule = "WeeklyCycle"
  Storage = File1
  Messages = Standard
  Pool = File
  SpoolAttributes = yes
  Priority = 10
  Write Bootstrap = "/var/lib/bacula/%c.bsr"
}
```

Окончательное расположение и формат файлов архивных резервных копий зависит от:

- настроек `backup_level` в конфигурации в backup_config (см. Табл. 9-2),
- настроек `Level` в конфигурации bacula-dir (см. справочное руководство по работе с ПО Bacula) и
- настроек `storage_type` в backup_config (см. Табл. 9-2).

В таблице ниже показаны результаты различных комбинаций этих вариантов.

Табл. 10-4: Параметры уровней резервного копирования

`backup_level` в **backup_config**	`Level` в **bacula-dir**	`storage_type` в **backup_config**	**Расположение и формат архивных файлов**
incremental	Incremental	nfs	Файлы резервных копий хранятся в хранилище NFS как дельта предыдущего архива резервных копий. Файл журнала состояния хранится в хранилище Bacula как дельта предыдущего файла.
incremental	Full	nfs	Файлы резервных копий хранятся в хранилище NFS как дельта предыдущего архива резервных копий. Файл журнала состояния хранится в хранилище Bacula в виде полного архива.
full	Incremental	nfs	Файлы резервных копий хранятся в хранилище NFS как полный архив. Файл журнала состояния хранится в хранилище Bacula как дельта предыдущего файла.
full	Full	nfs	Файлы резервных копий хранятся в хранилище NFS как полный архив. Файл журнала состояния хранится в хранилище bacula в виде полного архива.
incremental	Incremental	local	Файлы резервных копий хранятся в хранилище Bacula как дельта предыдущего архива резервных копий. Файл журнала состояния хранится в хранилище bac Bacula ula как дельта предыдущего файла.
incremental	Full	local	Файлы резервных копий хранятся в хранилище Bacula как полный архив. Файл журнала состояния хранится в хранилище Bacula в виде полного архива.
full	Incremental	local	Файлы резервных копий хранятся в хранилище Bacula как дельта предыдущего архива резервных копий.

			Файл журнала состояния хранится в хранилище Bacula как дельта предыдущего файла.
full	Full	local	Файлы резервных копий хранятся в хранилище Bacula как полный архив. Файл журнала состояния хранится в хранилище Bacula в виде полного архива.

10.5.4.3 Запуск резервного копирования в разные периоды времени

Сервер Bacula по умолчанию настроен таким образом, что он указывает клиенту Bacula выполнять операции резервного копирования, сконфигурированные в каталоге `/opt/bacula`. Поэтому все конфигурации в этом каталоге выполняются последовательно. Таким образом, когда вы устанавливаете сервер Bacula, конфигурация по умолчанию (в файле `/opt/bacula-dir/bacula-dir.conf`) создается с расписанием резервного копирования с периодом в одну неделю с использованием конфигураций в каталоге `/opt/bacula`. Важные выдержки из этого файла следующие:

```
JobDefs {
  Name = "DefaultJob"
  Type = Backup
  Level = Incremental
  Client = infrabackup-fd
  FileSet = "Full Set"
  Schedule = "WeeklyCycle"
  Storage = File1
  Messages = Standard
  Pool = File
  SpoolAttributes = yes
  Priority = 10
  Write Bootstrap = "/var/lib/bacula/%c.bsr"
}

Job {
  Name = "BackupInfrastructure"
  JobDefs = "DefaultJob"
  Client Run Before Job = "/usr/bin/itkf-maintenance-bacula-backup.sh"
  # This deletes the file copy
  Client Run After Job  = "/usr/bin/itkf-maintenance-bacula-backup-clean.sh"
}
Schedule {
  Name = "WeeklyCycle"
  Run = Full 1st sun at 23:05
  Run = Differential 2nd-5th sun at 23:05
  Run = Incremental mon-sat at 23:05
}
```

Здесь мы видим, что расписание `WeeklyCycle` используется в задании `"BackupInfrastructure"`, которое является нашим основным заданием резервного копирования. В этом задании используются сценарии `/usr/bin/itkf-maintenance-bacula-backup.sh` и `/usr/bin/itkf-maintenance-bacula-backup-clean.sh` для подготовки и очистки файлов резервных копий для Bacula. Эти сценарии в данном случае без аргументов (это означает выполнение всех операций резервного копирования, указанных в каталоге `/opt/bacula`). Однако, если мы используем специальный файл конфигурации резервного копирования в качестве аргумента, мы можем ограничить резервное копирование только этим файлом. Для дополнительных пояснений см. справочное руководство по работе с ПО Bacula.

Поэтому мы можем отредактировать bacula-dir (в файле `/opt/bacula-dir/bacula-dir.conf`) для создания нескольких резервных копий в разные временные интервалы.

В следующем примере мы добавим почасовое резервное копирование по указанной конфигурации.

На первом этапе мы добавляем новое расписание в файл `/opt/bacula-dir/bacula-dir.conf`:

```
Schedule {
  Name = "EveryThreeQuarters"
  Run = Full hourly at 0:45
}
```

Дополнительные параметры выполнения расписания можно найти в справочном руководстве по работе с ПО Bacula.

На втором шаге мы определяем новое задание, которое использует новое расписание:

```
Job {
  Name = "BackupInfrastructureFreq"
  JobDefs = "DefaultJob"
  Schedule = "EveryThreeQuarters"
  Client Run Before Job = "/usr/bin/itkf-maintenance-bacula-backup.sh
/opt/bacula/backup2.yaml"
  # This deletes the file copy
  Client Run After Job  = "/usr/bin/itkf-maintenance-bacula-backup-clean.sh
/opt/bacula/backup2.yaml"
}
```

Также обратите внимание на дополнительные аргументы клиентских скриптов, отмеченные красным. В нашем примере для этого задания используется файл конфигурации резервного копирования `/opt/bacula/backup2.yaml`.

Мы также можем установить тайм-аут для каждого задания резервного копирования, используя параметр -t, как в следующем примере:

```
Job {
  Name = "BackupInfrastructureFreq"
  JobDefs = "DefaultJob"
  Schedule = "EveryThreeQuarters"
  Client Run Before Job = "/usr/bin/itkf-maintenance-bacula-backup.sh -t 700
/opt/bacula/backup2.yaml"
  # This deletes the file copy
  Client Run After Job  = "/usr/bin/itkf-maintenance-bacula-backup-clean.sh
/opt/bacula/backup2.yaml"
}
```

Последний шаг — перезапустить контейнер bacula-dir, чтобы изменения вступили в силу:

```
docker restart bacula-dir
```

Таким образом, при использовании этой настройки резервное копирование, настроенное в `/opt/bacula/backup2.yaml`, будет выполняться каждый раз от четверти часа до одного часа.

10.6 Резервное копирование и восстановление нескольких облаков

Процедуры резервного копирования и восстановления можно использовать с несколькими облаками. Резервными копиями множества облаков можно управлять из:

- единой ВМ операций или
- единого сервера Bacula на ВМ резервного копирования (в случае использования Bacula в решении для резервного копирования).

Последний вариант позволяет управлять несколькими ВМ операций, которые также могут управлять несколькими облаками.

10.6.1 Управление резервными копиями нескольких облаков с одной ВМ операций

Если вы развернули платформу ICP с ВМ операций, с которой вы хотите запускать резервное копирование, вам нужно только создать конфигурацию yaml резервного копирования с правильным именем облака. Для примера:

```
cloud_name: 'second-cloud'
storage_type: 'nfs'
storage_path: '192.168.22.22:/opt/storage'
status_storage_path: '/opt/status-backup/'
exclude_database: no
cloud_config_strategy: keep-newer
image_backup_strategy: 'includelist'
image_includelist:
    - imageX
volume_backup_strategy: 'unattached'
volume_includelist:
    - voly
vm_backup_strategy: 'all'
vm_includelist:
    - instanceZ
```

Прежде чем запускать процедуры резервного копирования и восстановления, убедитесь, что все файлы управления облаком и учетные данные для конкретного облака находятся в каталоге конкретного облака в папке `/opt/etc-kolla/` (например, `/opt/etc-kolla/second_cloud`). Эти файлы включают:

- admin-openrc.sh,
- cloud-inventory,
- https_cacert.cer (в случае платформы ICP с настроенным TLS).

Если `https_cacert.cer` не существует, сделайте копию файла `haproxy-ca.crt`, который находится в каталоге сертификатов в каталоге `/opt/etc-kolla/`, и назовите его `https_cacert.cer`. Вы можете использовать следующую команду в каталоге конкретного облака:

```
cp certificates/haproxy-ca.crt https_cacert.cer
```

Затем вы можете нормально запустить процедуры резервного копирования и восстановления, как описано в разделах 10.3 и 10.4.

Вы также можете запускать процедуры резервного копирования и восстановления по нескольким конфигурациям облаков с сервера Bacula. Для этого нужно поместить файлы конфигураций в правильный каталог в соответствии с инструкциями в разделе 10.5.3.

Если вы хотите запустить процедуры резервного копирования и восстановления на ВМ операций, которая не использовалась для развертывания конкретной ICP, вам необходимо сначала подключить свою ВМ операций к развертыванию платформы ICP. Для этого выполните следующие действия:

1. Поместите файлы из каталога конкретного облака другой ВМ операций в каталоге `/opt/etc-kolla/` (например, `/opt/etc-kolla/cloud2/`) в каталог конкретного облака в каталоге `/opt/etc-kolla/` на локальной ВМ операций. Эти файлы включают как минимум:
    а) admin-openrc.sh,
    б) cloud-inventory,
    в) https_cacert.cer (в случае платформы ICP с настроенным TLS).
2. Скопируйте записи хоста из другой ВМ операций (расположенной в `/etc/hosts`) в отношении узлов ICP, для которых вы хотите выполнить резервное копирование и восстановление, в файл на локальной ВМ операций.
3. Создайте новые пары ключей аутентификации для SSH на локальной ВМ операций, если они еще не созданы, с помощью команды `ssh-keygen`.

4. Добавьте содержимое открытого ключа SSH (обычно этот ключ находится в файле `/root/.ssh/id_rsa.pub`) в файл `authorized_keys` в каталоге конфигурации SSH (обычно это файл `/root/.ssh/authorized_keys`) на всех новых управляемых узлах ICP.

После подключения к конкретной ICP вы можете создавать конфигурации резервного копирования и запускать процедуры резервного копирования и восстановления, как описано в начале этого раздела.

10.6.2 Управление резервными копиями нескольких облаков с одной ВМ резервного копирования

Вы можете управлять резервными копиями нескольких облаков с одной ВМ резервного копирования (на которой работает система резервного копирования Bacula), установив соединение `bacula-dir` с `bacula-fd`. Это можно сделать в два этапа.

На первом этапе вы должны настроить bacula-fd на ВМ операций для управления из конкретного bacula-dir. Пример bacula-fd.conf, который подключается к bacula-dir (этот файл обычно находится в `/etc/bacula/bacula-fd.conf` на ВМ операций):

```
# List Directors who are permitted to contact this File daemon
Director {
  Name = infrabackup-dir-multiple-clouds
  Password = "[пароль_предоставляется_системным_администратором]"
}
# Restricted Director, used by tray-monitor
Director {
  Name = infrabackup-mon
  Password = "[пароль_предоставляется_системным_администратором]"
  Monitor = yes
}
# "Global" File daemon configuration specifications
FileDaemon {
  Name = infrabackup-fd-second
  FDport = 9102
  WorkingDirectory = /var/lib/bacula
  Pid Directory = /run/bacula
  Maximum Concurrent Jobs = 20
  Plugin Directory = /usr/lib/bacula
  FDAddress = 192.168.41.29
}
# Send all messages except skipped files back to Director
Messages {
  Name = Standard
  director = infrabackup-dir = all, !skipped, !restored
}
```

Красным цветом показаны конкретные записи конфигурации bacula-dir, которые позволяют bacula-dir управлять этим bacula-fd. Зеленым цветом показана конфигурация bacula-fd.

После внесения изменений в конфигурацию сервис bacula-fd необходимо перезапустить:

```
systemctl restart bacula-fd
```

На втором этапе нужно настроить bacula-dir на ВМ резервного копирования, чтобы он также управлял определенным bacula-fd. На этом шаге нужно создать задание, которое будет выполнять операции резервного копирования и восстановления на этом bacula-fd. Вот пример такой конфигурации (этот файл обычно находится в `/opt/bacula-dir/bacula-dir.conf` на ВМ резервного копирования):

```
Director {
  Name = infrabackup-dir-multiple-clouds
  DIRport = 9101
  QueryFile = "/etc/bacula/scripts/query.sql"
  WorkingDirectory = "/var/lib/bacula"
```

```
  PidDirectory = "/run/bacula"
  Maximum Concurrent Jobs = 20
  Password = "[пароль_предоставляется_системным_администратором]"
  Messages = Daemon
  DirAddress = 127.0.0.1
}
JobDefs {
  Name = "DefaultJob"
  Type = Backup
  Level = Incremental
  Client = infrabackup-fd
  FileSet = "Full Set"
  Schedule = "WeeklyCycle"
  Storage = File1
  Messages = Standard
  Pool = File
  SpoolAttributes = yes
  Priority = 10
  Write Bootstrap = "/var/lib/bacula/%c.bsr"
}
Job {
  Name = "BackupInfrastructure"
  JobDefs = "DefaultJob"
  Client Run Before Job = "/usr/bin/itkf-maintenance-bacula-backup.sh"
  # This deletes the file copy
  Client Run After Job  = "/usr/bin/itkf-maintenance-bacula-backup-clean.sh"
}
Job {
  Name = "BackupInfrastructureSecond"
  Client = infrabackup-fd-second
  JobDefs = "DefaultJob"
  Client Run Before Job = "/usr/bin/itkf-maintenance-bacula-backup.sh"
  # This deletes the file copy
  Client Run After Job  = "/usr/bin/itkf-maintenance-bacula-backup-clean.sh"
}
Job {
  Name = "BackupCatalog"
  JobDefs = "DefaultJob"
  Level = Full
  FileSet="Catalog"
  Client = catalogbackup-fd
  Schedule = "WeeklyCycleAfterBackup"
  # This creates an ASCII copy of the catalog
  # Arguments to make_catalog_backup.pl are:
  #  make_catalog_backup.pl <catalog-name>
  RunBeforeJob = "/etc/bacula/scripts/make_catalog_backup.pl MyCatalog"
  # This deletes the copy of the catalog
  RunAfterJob  = "/etc/bacula/scripts/delete_catalog_backup"
  Write Bootstrap = "/var/lib/bacula/%n.bsr"
  Priority = 11
}
Job {
  Name = "RestoreInfrastructure"
  Type = Restore
  Client = infrabackup-fd
  Storage = File1
# The FileSet and Pool directives are not used by Restore Jobs
# but must not be removed
  FileSet="Full Set"
  Pool = File
  Messages = Standard
  Where = /
}
FileSet {
  Name = "Full Set"
  Include {
    Options {
```

```
      signature = MD5
    }
    File = /opt/bacula-b
  }
}
Schedule {
  Name = "WeeklyCycle"
  Run = Full 1st sun at 23:05
  Run = Differential 2nd-5th sun at 23:05
  Run = Incremental mon-sat at 23:05
}
Schedule {
  Name = "WeeklyCycleAfterBackup"
  Run = Full sun-sat at 23:10
}
FileSet {
  Name = "Catalog"
  Include {
    Options {
      signature = MD5
    }
    File = "/var/lib/bacula/bacula.sql"
  }
}
Client {
  Name = infrabackup-fd
  Address = 192.168.41.9
  FDPort = 9102
  Catalog = MyCatalog
  Password = "[пароль_предоставляется_системным_администратором]"
  File Retention = 60 days
  Job Retention = 6 months
  AutoPrune = yes
}
Client {
  Name = infrabackup-fd-second
  Address = 192.168.41.29
  FDPort = 9102
  Catalog = MyCatalog
  Password = "[пароль_предоставляется_системным_администратором]"
  File Retention = 60 days
  Job Retention = 6 months
  AutoPrune = yes
}

Client {
  Name = catalogbackup-fd
  Address = 127.0.0.1
  FDPort = 9102
  Catalog = MyCatalog
  Password = "[пароль_предоставляется_системным_администратором]"
  File Retention = 60 days
  Job Retention = 6 months
  AutoPrune = yes
}
Autochanger {
  Name = File1
  Address = 192.168.41.165
  SDPort = 9103
  Password = "[пароль_предоставляется_системным_администратором]"
  Device = FileChgr1
  Media Type = File1
  Maximum Concurrent Jobs = 10
  Autochanger = File1
}

Catalog {
```

```
  Name = MyCatalog
  DB Address = "localhost"; dbname = "bacula"; dbuser = "bacula"; dbpassword =
"[пароль_предоставляется_системным_администратором]"
}

Messages {
  Name = Standard
  mailcommand = '/usr/sbin/bsmtp -h localhost -f "\(Bacula\) \<%r\>" -s "Bacula: %t %e of
%c %l" %r'
  operatorcommand = '/usr/sbin/bsmtp -h localhost -f "\(Bacula\) \<%r\>" -s "Bacula:
Intervention needed for %j" %r'
  mail = root = all, !skipped
  operator = root = mount
  console = all, !skipped, !saved
  append = "/var/log/bacula/bacula.log" = all, !skipped
  catalog = all
}

Messages {
  Name = Daemon
  mailcommand = '/usr/sbin/bsmtp -h localhost -f "\(Bacula\) \<%r\>" -s "Bacula daemon
message" %r'
  mail = root = all, !skipped
  console = all, !skipped, !saved
  append = "/var/log/bacula/bacula.log" = all, !skipped
}
Pool {
  Name = Default
  Pool Type = Backup
  Recycle = yes
  AutoPrune = yes
  Volume Retention = 365 days
  Maximum Volume Bytes = 50G
  Maximum Volumes = 100
}
# File Pool definition
Pool {
  Name = File
  Pool Type = Backup
  Recycle = yes
  AutoPrune = yes
  Volume Retention = 365 days
  Maximum Volume Bytes = 50G
  Maximum Volumes = 100
  Label Format = Vol-
}
# Scratch pool definition
Pool {
  Name = Scratch
  Pool Type = Backup
}
#
# Restricted console used by tray-monitor to get the status of the director
#
Console {
  Name = infrabackup-mon
  Password = "[пароль_предоставляется_системным_администратором]"
  CommandACL = status, .status
}
```

Красным цветом показана вновь определенная конфигурация клиента (`Client`), которая соответствует конфигурациям, заданным на первом этапе, когда мы настроили bacula-fd.

Зеленым цветом показано имя модуля Director (`Director`), которое мы также определили на первом этапе.

Фиолетовым цветом показано вновь определенное задание (Job) с именем BackupInfrastructureSecond, в котором используется новый клиент.

После перезапуска bacula-dir применяются изменения:

```
docker restart bacula-dir
```

Дополнительные сведения о настройке Bacula см. в документах: справочное руководство по работе с ПО Bacula и руководство по работе с консолью Bacula.

Теперь вы можете запускать операции резервного копирования и восстановления, используя консоль Bacula и обращаясь к заданию `BackupInfrastructureSecond`.

10.7 Система резервного копирования и восстановления приложений

Резервное копирование приложений основано на том же способе, что и резервное копирование инфраструктуры, но содержит отдельные контейнеры и службы из соображений безопасности. Контейнеры резервного копирования приложений разработаны таким образом, что могут сосуществовать с контейнерами резервного копирования инфраструктуры, они могут использовать:

- один и тот же IP-адрес, но разные сетевые порты;
- один и тот же тот же сервер базы данных, но разные имена каталогов;
- одно и то же или разные места хранения.

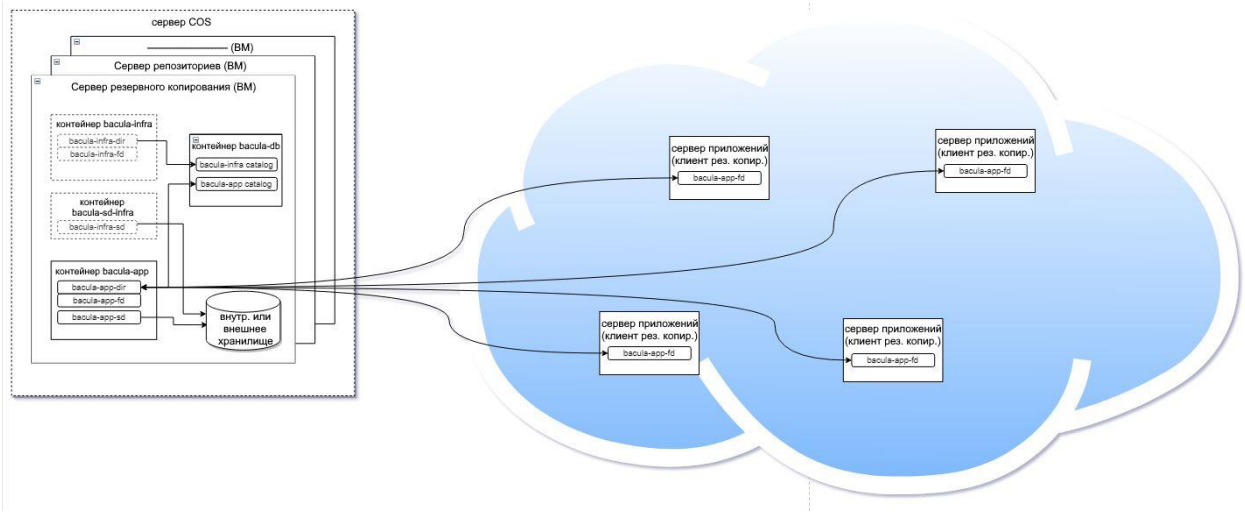


Рис. 10-3: Реализация резервного копирования приложений через систему Bacula

10.7.1 Конфигурация сервера резервного копирования приложений

После установки пакета itbacula-tools конфигурация шаблона yaml копируется в файл:

```
/opt/bacula-app/itbac-app-conf.yml
```

Этот файл необходимо отредактировать по крайней мере для параметра конфигурации 'LCLIENTS', где указан список имен хостов клиента резервного копирования приложения (если не указан ни один клиент, сервер резервного копирования будет настроен, но будет создано только одно задание для резервного копирования собственного каталога).

Табл. 10-5: Файл конфигурации Bacula для приложений

Параметр	Тип	Описание	По умолчанию
EXT_ADDR	необяз.	Сервер Bacula обслуживает внешний IP-адрес, к которому	(первичный локальный IP-адрес)

		могут обращаться клиенты File Daemon.	
LCLIENTS	необяз.	Список имен хостов внешних клиентов, разделенных пробелом (должен быть разрешимым — определяется в DNS или /etc/hosts)	
REGISTRY	обяз.	Сервер реестра (сетевое имя или IP-адрес сервера Nexus)	registry.devops.iskrauraltel.cloud
REG_PATH	обяз.	Путь к образу контейнера docker Bacula	/ai6212ax/bacula-infra-backup/
VERSION	необяз.	Версия образа контейнера docker Bacula	(itbacula-tools deb version)
CONF_ DIR	необяз.	Путь к конфигурационным файлам Bacula	/opt/bacula/conf
LOG _ DIR	необяз.	Путь к лог-файлам Bacula	/var/log/bacula
BACKUP _ DIR	необяз.	Путь к каталогу хранилища Bacula	/opt/backups
DB_NAME	необяз.	Имя каталога БД Bacula (postgresql)	/opt/bacula-data
DB_USER	необяз.	Имя пользователя БД Bacula	bacula_app
DB_PASS	необяз.	Пароль БД Bacula	пароль предоставляется системным администратором
DB_DIR	необяз.	Путь к каталогу БД Bacula	/opt/bacula-data
DIR_NAME	необяз.	Имя Bacula Director	bacula_app-dir
DIR_PASS	необяз.	Пароль Bacula Director	пароль предоставляется системным администратором
DIR_PORT	необяз.	Порт Bacula Director	9111
MON_NAME	необяз.	Имя компонента мониторинга Bacula Director	bacula_app-mon
MON_PASS	необяз.	Пароль компонента мониторинга Bacula Director	пароль предоставляется системным администратором
FD_NAME	необяз.	Имя локального Bacula File Daemon	bacula_app-fd
FD_PASS	необяз.	Пароль локального Bacula File Daemon	пароль предоставляется системным администратором
FD_PORT	необяз.	Порт локального Bacula File Daemon	9112
SD_NAME	необяз.	Имя Bacula Storage Daemon	bacula_app-sd
SD_PASS	необяз.	Пароль Bacula Storage Daemon	пароль предоставляется системным администратором
SD_PORT	необяз.	Порт Bacula Storage Daemon	9113

Параметры конфигурации, кроме 'LCLIENTS', не рекомендуется изменять после того, как был выполнен первоначальный запуск (в этот момент создается и инициализируется БД Bacula).

Обратите внимание, что при размещении сервера Bacula за DNAT необходимо указывать специальный EXT_ADDR (внешний IP-адрес внутри ВМ не виден).

Пример файла конфигурации itbac-app-conf.yml, изначально созданного при развертывании пакета:

```
# Comment or delete next line or change 'template' value to something else when this
configuration file is prepared
itbac_app_conf: template

# Bacula services external IP address which can be accessed by file daemon clients
(primary local address will be used if not specified)
EXT:
 ADDR:

# List of external clients (hostname wkich must be defined in DNS or /etc/hosts) delimited
by space
LCLIENTS:

# Bacula docker container image path (nexus repository path)
REGISTRY:
REG_PATH:
VERSION:

#
### Optional parameters
#

# Bacula log files path
LOG:
 DIR:

# Bacula storage direktory path
BACKUP:
 DIR:

# Bacula DB parameters (postgresql)
# Database parameters should not be changed after first initial start (unless it were
changed in database first)
DB:
 NAME:
 USER:
 PASS:
 DIR:

# Bacula director configuration parameters
DIR:
 NAME:
 PASS:
 PORT:

# Bacula monitor configuration parameters
MON:
 NAME:
 PASS:

# Bacula internal file daemon configuration parameters
FD:
 NAME:
 PASS:
 PORT:

# Bacula storage daemon configuration parameters
SD:
 NAME:
 PASS:
 PORT:
```

Параметры, не заданные в этом файле конфигурации, будут иметь значения по умолчанию.

10.7.2 Создание файлов конфигурации Bacula при резервном копировании приложений

Чтобы подготовить или обновить файлы конфигурации Bacula, выполните:

```
itbac-app-configure.sh
```

Эта команда подготовит все необходимые файлы конфигурации Bacula, расположенные в каталоге конфигурации (расположение по умолчанию /opt/bacula-app/conf). Эта папка конфигурации также содержит файлы конфигурации клиентов, которые необходимо вручную передать на клиенты Bacula.

Сценарий конфигурации Bacula также перезапустит контейнеры на стороне сервера Bacula, если они уже запущены.

10.7.3 Первоначальный запуск служб резервного копирования приложений

Для запуска служб резервного копирования Bacula сначала необходимо запустить службу базы данных. Резервное копирование приложений может совместно использовать один и тот же сервер БД с резервным копированием инфраструктуры при работе на одном хосте (в этом случае службу БД необходимо запустить только один раз).

Запустите службу БД с помощью команды, если она еще не запущена:

```
itbac-app.sh start-db
```

Затем запустите службы резервного копирования приложения Bacula с помощью команды:

```
itbac-app.sh start
```

10.7.4 Проверка работы резервного копирования приложения

Чтобы проверить, запущены ли службы, вы можете открыть консоль Bacula с помощью команды:

```
itbac-app.sh console
```

10.7.5 Процедура обновления резервной копии приложения

Установка резервной копии приложения не выполняется в автоматическом режиме, поэтому процедуру обновления также необходимо выполнить вручную, выполнив следующие задачи:

- Обновите пакет Debian (обычно выполняется с помощью процедуры bacula-infra-backup)
- Выполните команду **itbac-app.sh stop**, чтобы остановить контейнер сервера bacula.
- Выполните команду **itbac-app-configure.sh**, чтобы обновить параметр версии пакета.
- Выполните команду **itbac-app.sh start**, чтобы запустить контейнер сервера bacula.

10.7.6 Добавление нового клиента резервного копирования приложения

Чтобы добавить новый клиент резервного копирования приложения, обновите параметр LCLIENTS в файле конфигурации /opt/bacula-app/itbac-app-conf.yml, затем запустите скрипт itbac-app-configure.sh. Скрипт повторно генерирует файлы конфигурации Bacula, расположенные в каталоге CONF_DIR (расположение по умолчанию /opt/bacula-app/conf), где также будет файл конфигурации для нового добавленного клиента, который необходимо вручную перенести на узел клиента bacula в /etc/ bacula, затем нужно запустить службу bacula-fd.

10.7.7 Установка клиента Bacula

Клиент Bacula может быть подготовлен скриптом, интегрированным в пакет IT_VGP_CSP, с помощью следующей команды:

```
# /opt/bacula/bacula_conf.sh setup-fd --hostname_dir=(bacula server hostname)
--password=(bacula server password)
```

Его также можно установить вручную, настроив файл /etc/bacula/bacula-fd.conf и запустив службу bacula-fd. Для помощи, на сервере Bacula после настройки клиента будет сгенерирован конфигурационный файл с путем:

/opt/bacula-app/conf/bacula-fd.conf-(client hostname)

Также в директории /opt/bacula должны быть скрипты backup_command.sh и backup_status.sh, первый скрипт запускается перед процедурой резервного копирования и создает директорию /opt/archive, затем копирует все файлы, которые необходимо заархивировать в этом месте, после этого запускается следующий скрипт для выполнения задач после резервного копирования.

10.7.8 Процедуры архивирования и восстановления данных

10.7.8.1 Архивирование по расписанию

Архивируются только данные, расположенные на CSI:

- данные файловой системы,
- база данных.

Архивация запускается автоматически каждый день в 03:10. Архивирование на локальный диск выполняется в любом случае. Если вы настроили архивацию на внешний сервер, архивация будет выполняться в локальном хранилище внешнего сервера в 03:10.

Расписание задается в файле /etc/bacula/bacula-dir.conf на сервере, где находится Bacula Director:

```
Schedule {
  Name = "DailyFullBackup"
  Run = Full sun-sat at 3:10
}
```

10.7.8.2 Архивирование по запросу

Архивирование по запросу выполняется с помощью инструмента bconsole на сервере, где находится Bacula Director.

- Запустите программу bconsole. Появится запрос команды:

[root@backupsrv]# itbac-app.sh console

Connecting to Director bigcsi:9101

1000 OK: backupsrv-dir Version: 5.0.0 (26 January 2010)

Enter a period to cancel a command.

*

- Выполните команду run:

```
*run
Automatically selected Catalog: catalog
Using Catalog "catalog"
A job name must be specified.
The defined Job resources are:
```

```
     1: BackupCatalog
     2: RestoreFiles
     3: Backup-smallcsi
     4: ExtBackup_smallcsi_on_smallcsi
     5: TapeBackup-smallcsi
Select Job resource (1-5):
```

- Выберите задание, например, для локального архивирования:

```
Select Job resource (1-5): 3
Run Backup job
JobName:  Backup-smallcsi
Level:    Full
Client:   smallcsi-fd
FileSet:  Full Set
Pool:     DiskPool (From Job resource)
Storage:  File (From Job resource)
When:     2012-04-18 10:36:54
Priority: 100
OK to run? (yes/mod/no):
```

- Подтвердите действие, введя yes:

```
OK to run? (yes/mod/no): yes
Job queued. JobId=25
*
```

Будет создан новый архив на локальном диске в /opt/bacula-storage.

Помимо локального архивирования, также поддерживается архивирование на внешний сервер архивации и на ленточные накопители RDX и LTO3, если они доступны.

```
     3: Backup-smallcsi
     4: ExtBackup_smallcsi_on_smallcsi
     5: TapeBackup-smallcsi
     6: RDXTapeBackup-smallcsi
```

В случае архивирования на внешний сервер архивации выберите задание с префиксом ExtBackup_. Имя задания состоит из префикса с именем клиента, на котором мы выполняем резервное копирование, за которым следует _on_ и расположение внешнего сервера архивации. В нашем случае клиент и внешний сервер архивации — это один и тот же компьютер.

10.7.8.3 Восстановление данных

Предупреждение!

Для восстановления данных в конфигурациях с высокой доступностью, важно использовать одно и то же сетевое имя для виртуальных машин, поскольку конфигурация для базы и прочего также архивируется и содержит информацию о сетевом имени.

Предупреждение!

> Перед восстановлением данных убедитесь, что доступен DiskPool, из которого будет производиться восстановление. В противном случае восстановление не будет выполнено, а останется в состоянии «выполняется», пока вы не отмените его вручную.

Возможно частичное или полное восстановление данных. Частичное восстановление данных используется в случае случайного удаления какой-либо части ключевых данных файловой системы. Полное восстановление данных используется, если все данные были потеряны или физический или виртуальный компьютер был уничтожен.

По умолчанию данные восстанавливаются в новую папку /opt/bacula-restores/ на диске.

Восстановление системы выполняется с помощью инструмента bconsole. Если вы не уверены, какой тип архивирования подходит для восстановления, с помощью команды restore можно просмотреть возможные варианты архивирования. После этого выберите из архива файлы для восстановления (по умолчанию – все) и отправьте задание на восстановление в обработку.

```
[root@backupsrv]# itbac-app.sh console
Connecting to Director bigcsi:9101
1000 OK: bigcsi-dir Version: 5.0.0 (26 January 2010)
Enter a period to cancel a command.
*restore
Automatically selected Catalog: catalog
Using Catalog "catalog"

First you select one or more JobIds that contain files
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
select which files from those JobIds are to be restored.

To select the JobIds, you have the following choices:
     1: List last 20 Jobs run
     2: List Jobs where a given File is saved
     3: Enter list of comma separated JobIds to select
     4: Enter SQL list command
     5: Select the most recent backup for a client
     6: Select backup for a client before a specified time
     7: Enter a list of files to restore
     8: Enter a list of files to restore before a specified time
     9: Find the JobIds of the most recent backup for a client
    10: Find the JobIds for a backup for a client before a specified time
    11: Enter a list of directories to restore for found JobIds
    12: Select full restore to a specified Job date
    13: Cancel
Select item:  (1-13): 9
Defined Clients:
     1: smallcsi-fd
     2: bigcsi-fd
Select the Client (1-2): 1
The defined FileSet resources are:
```

```
     1: Full Set
     2: Full Set Ext smallcsi
Select FileSet resource (1-2): 2
+-------+-------+----------+----------+---------------------+----------------------------------
-+
| JobId | Level | JobFiles | JobBytes | StartTime           | VolumeName                       |
+-------+-------+----------+----------+---------------------+----------------------------------
-+
| 21    | F     | 2        | 0        | 2012-04-18 03:10:02 | FileExtPool_smallcsi_bigcsi-00004 |
+-------+-------+----------+----------+---------------------+----------------------------------
-+
To select the JobIds, you have the following choices:
     1: List last 20 Jobs run
     2: List Jobs where a given File is saved
     3: Enter list of comma separated JobIds to select
     4: Enter SQL list command
     5: Select the most recent backup for a client
     6: Select backup for a client before a specified time
     7: Enter a list of files to restore
     8: Enter a list of files to restore before a specified time
     9: Find the JobIds of the most recent backup for a client
    10: Find the JobIds for a backup for a client before a specified time
    11: Enter a list of directories to restore for found JobIds
    12: Select full restore to a specified Job date
    13: Cancel
Select item:  (1-13): 3
Enter JobId(s), comma separated, to restore: 21
You have selected the following JobId: 21

Building directory tree for JobId(s) 21 ...
1 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ mark *
2 files marked.
$ done
Bootstrap records written to /var/spool/bacula/bigcsi-dir.restore.1.bsr

The job will require the following
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

    FileExtPool_smallcsi_velik FileExt_smallcsi           FileStorageExt_bigcsi
```

```
Volumes marked with "*" are online.


2 files selected to be restored.

Run Restore job
JobName:         RestoreFiles
Bootstrap:       /var/spool/bacula/bigcsi-dir.restore.1.bsr
Where:           /opt/bacula-restores
Replace:         always
FileSet:         Full Set
Backup Client:   smallcsi-fd
Restore Client:  smallcsi-fd
Storage:         FileExt_smallcsi
When:            2012-04-18 12:04:09
Catalog:         catalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no):yes
Job queued. JobId=26
```

Когда восстановление будет выполнено, все заархивированные файлы будут помещены в /opt/bacula-restores/ на клиенте и мы сможем позже скопировать их в нужный каталог.

Если нужно восстановить данные на другом компьютере, вы можете сделать это с помощью следующих команд:

```
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /var/spool/bacula/bigcsi-dir.restore.1.bsr
Where:           /opt/bacula-restores
Replace:         always
FileSet:         Full Set
Backup Client:   smallcsi-fd
Restore Client:  smallcsi-fd
Storage:         FileExt_smallcsi
When:            2012-04-18 12:04:09
Catalog:         catalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no):mod
Parameters to modify:
     1: Level
     2: Storage
     3: Job
     4: FileSet
```

```
     5: Restore Client
     6: When
     7: Priority
     8: Bootstrap
     9: Where
    10: File Relocation
    11: Replace
    12: JobId
    13: Plugin Options
Select parameter to modify (1-13): 5
The defined Client resources are:
     1: bigcsi-fd
     2: smallcsi-fd
Select Client (File daemon) resource (1-2): 1
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /var/spool/bacula/bigcsi-dir.restore.5.bsr
Where:           /opt/bacula-restores
Replace:         always
FileSet:         Full Set
Backup Client:   smallcsi-fd
Restore Client:  bigcsi-fd
Storage:         File
When:            2012-04-18 12:17:24 PM
Catalog:         catalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): yes
Job queued. JobId=31
```

В нашем случае все заархивированные файлы помещаются в /opt/bacula-restores/ на сервере CSI, и мы сможем позже скопировать их в нужную директорию.

10.7.8.4 Восстановление базы данных Solid

Восстановите нужный архив из архива Bacula в выбранное место (по умолчанию директория находится на VHP в /opt/bacula-restores). После завершения восстановления все необходимые файлы восстановления всех серверов баз данных solid на активном сервере VHP (далее **restsrv***) для архивации можно найти по адресу /opt/bacula-restores/opt/archive/solid (далее **restpath***).

Восстановление сервера данных solid выполняется на одном из CSI, поэтому нужно войти в систему как пользователь 'root'. Поскольку восстановленные данные обычно находятся на одной из хост-систем, их необходимо скопировать оттуда на машину CSI с помощью команды scp. При восстановлении всей системы необходимо восстановить все экземпляры solid, следуя приведенной далее процедуре (процедура приведена только для одного экземпляра).

- Сначала остановите все приложения, которые используют экземпляр solid (в будущем сервер CSI будет содержать другие экземпляры solid, помимо MN и radius. Процедуры остановки и запуска соответствующих серверов приложений для этих экземпляров должны быть получены заранее):
    - на гостевой платформе CSI это сервер Radius:
        - `crm resource stop c_csiradiusd or g_csiradiusd with standalone`
    - на гостевой платформе AP это сервер jboss:
        - `crm resource stop g_jboss`
- Если экземпляр solid еще не существует, сначала создайте его, используя ту же процедуру, что и при первой установке, и включите его в мониторинг crm (во время процедуры используется solid_conf.sh в crm_conf.sh).
- Автономная установка CSI:
    - Остановите экземпляр solid, который вы хотите восстановить, с помощью команды:
        - `crm resource stop g_(instname)`
    - Скопируйте файл solid из архива:
        - `rm -f /opt/solid/(instname)/*.log`
        - `rm -f /opt/solid/(instname)/*.out`
        - `rm -f /opt/si3000/solid/(instname)/*.ini*`
        - `scp (restsrv):(restpath)/(instname)/solidnet.cfg /opt/si3000/solid/(instname)/`
        - `scp (restsrv):(restpath)/(instname)/solid.db /opt/solid/(instname)/`
        - `scp (restsrv):(restpath)/(instname)/solid.ini /opt/solid/(instname)/`
        - `cp /opt/solid/(instname)/solid.ini /opt/si3000/solid/(instname)/solid.ini.`hostname -s``
    - Запустите восстановленный экземпляр solid:
        - `crm resource start g_(instname)`
- Конфигурация дублированного CSI с высокой доступностью:
    - Остановите экземпляр solid, который вы хотите восстановить, с помощью команды:
        - `crm resource stop c_(instname)`
    - Удалите старые файлы solid *.out и *.log, а затем скопируйте файлы solid из архива (повторите эту процедуру на обоих узлах CSI):
        - `rm -f /opt/solid/(instname)/*.log`
        - `rm -f /opt/solid/(instname)/*.out`
        - `scp (restsrv):(restpath)/(instname)/solidnet.cfg /opt/si3000/solid/(instname)/`
        - `scp (restsrv):(restpath)/(instname)/solid.db /opt/solid/(instname)/`
        - `scp (restsrv):(restpath)/(instname)/solid.ini.(csi nodename) /opt/solid/(instname)/solid.ini`
        - `cp /opt/solid/(instname)/solid.ini /opt/si3000/solid/(instname)/solid.ini.`hostname -s``
        - `scp (restsrv):(restpath)/(instname)/solidhac.ini.(csi nodename) /opt/solid/(instname)/solidhac.ini`
        - `cp /opt/solid/(instname)/solidhac.ini /opt/si3000/solid/(instname)/solidhac.ini.`hostname -s``
    - Запустите восстановленный экземпляр solid:
        - `crm resource start c_(instname)`
- После завершения восстановления серверов данных solid перезапустите серверы приложений (в дальнейшем на сервере CSI будет находиться другой экземпляр solid, помимо MN и radius. Процедуры остановки и запуска соответствующих серверов приложений для этих экземпляров должны быть получены заранее):
    - на гостевой платформе CSI это сервер Radius:
        - `crm resource start c_csiradiusd or g_csiradiusd pri standalone`
    - на гостевой платформе AP это сервер jboss:
        - `crm resource start g_jboss`

10.7.8.5 Восстановление Postgres

Восстановление Postges выполняется командой pg_dump. Если вы хотите восстановить определенную базу данных Postgres, нужно использовать команду:

#psql -h (сетевое имя или IP-адрес гостя) -p (порт нужного экземпляра) –d (имя базы данных) -U (имя пользователя определенного экземпляра) < (местоположение резервных копий данных)

После выполнения необходимо ввести пароль определенного пользователя.

(имя пользователя и пароль определенного экземпляра можно найти в файле конфигурации в /opt/si2000/pg/work(порт экземпляра)/pgnet.cfg)

Все данные хранятся в каталоге /opt/archive/pg/(порт определенного экземпляра).

Пример: psql -h postgresBackup.csi.iskrauraltel.ru -p 5432 -d it_aaaaaa_yjff0a -U postgres < /opt/archive/pg/sorm31cdr_it_aaaaaa_yjff0a.sql

10.7.8.6 Восстановление базы данных OpenLDAP

Восстановите нужный архив из архива Bacula в выбранное место (по умолчанию директория находится на VHP в /opt/bacula-restores). После завершения восстановления вы можете найти все необходимые файлы восстановления всех экземпляров базы данных LDAP на активном сервере VHP (далее **restsrv***) для архивации по адресу /opt/bacula-restores/opt/archive/ldap (далее **restpath***).

Восстановление сервера данных solid выполняется на одном из CSI, поэтому нужно войти в систему как пользователь ‘root’. Поскольку восстановленные данные обычно находятся на одной из хост-систем, их необходимо скопировать оттуда на машину CSI с помощью команды scp. При восстановлении всей системы необходимо восстановить все экземпляры LDAP, следуя приведенной далее процедуре во всех конфигурациях, кроме высокой доступности (процедура приведена только для одного экземпляра MN). **Учтите, что, если LDAP расположен в /opt/si3000/ldap вместо /opt/ldap, нужно использовать соответствующий путь**.

Запустите OpenLDAP:

- `crm resource stop g_csislapd_(instname)`

Создайте резервные копии существующих файлов:

- `cp /opt/ldap/(instname)/slapd.d/ /opt/ldap/(instname)/slapd.d_org -rf`
- `cp /opt/ldap/(instname)/ldapDB/ /opt/ldap/(instname)/ldapDB_org -rf`

Удалите старые файлы базы данных:

- `rm /opt/ldap/(instname)/slapd.d/* -rf`
- `rm /opt/ldap/(instname)/ldapDB/* -rf`

Восстановите экземпляр LDAP:

- `chmod 755 ldap_config_(instname).ldif`
- `chmod 755 ldap_bdb_(instname).ldif`
- `sudo -u ldap slapadd -n0 -F /opt/ldap/(instname)/slapd.d/ -l ./ldap_config_(instname).ldif`
- `sudo -u ldap slapadd -F /opt/ldap/(instname)/slapd.d/ -l ./ldap_bdb_(instname).ldif`

Запустите OpenLDAP:

- `crm resource start g_csislapd_(instname)`

В системах с высокой доступностью используйте процедуру, описанную ннже.

Запустите OpenLDAP:

- `crm resource stop g_csislapd_(instname)`

Создайте резервные копии существующих файлов:

- `cp /opt/ldap/(instname)/slapd.d/ /opt/ldap/(instname)/slapd.d_org -rf`
- `cp /opt/ldap/(instname)/ldapDB/ /opt/ldap/(instname)/ldapDB_org -rf`

Удалите старые файлы базы данных:

- `rm /opt/ldap/(instname)/slapd.d/* -rf`
- `rm /opt/ldap/(instname)/ldapDB/* -rf`

Восстановите экземпляр LDAP:

- `chmod 755 ldap_config_(instname).ldif`
- `chmod 755 ldap_bdb_(instname).ldif`
- `sudo -u ldap slapadd -n0 -F /opt/ldap/(instname)/slapd.d/ -l ./ldap_config_(instname).ldif`
- `sudo -u ldap slapadd -F /opt/ldap/(instname)/slapd.d/ -l ./ldap_bdb_(instname).ldif`

Запустите OpenLDAP:

- `crm resource start g_csislapd_(instname)`

10.7.8.6.1 Практический пример

Пример приведен для конфигурации с высокой доступностью (HA). Используемые имена серверов: vhp1, vhp2, csi1, csi2, ap1 и ap2. Восстановление Bacula выполняется на сервере vho1 в каталог по умолчанию (эта часть не показана в примере и была выполнена перед данной процедурой). В нашем примере мы выполняем восстановление MN и Radius базы данных Solid (csimncomm и csiradius). Процедура восстановления выполняется следующим образом:

```
(ap1)# crm resource stop g_jboss


(ap2)# crm resource stop g_jboss


(csi1)# export restpath=/opt/bacula-restores/opt/ archive/solid
(csi1)# export restsrv=vhp1
(csi1)# export instname=csiradius
(csi1)# crm resource stop c_csiradiusd
(csi1)# crm resource stop c_${instname}
(csi1)# rm -f /opt/solid/${instname}/*.log
(csi1)# rm -f /opt/solid/${instname}/*.out
(csi1)#               scp               ${restsrv}:${restpath}/${instname}/solidnet.cfg
/opt/si3000/solid/${instname}/
(csi1)# scp ${restsrv}:${restpath}/${instname}/solid.db /opt/solid/${instname}/
(csi1)#      scp      ${restsrv}:${restpath}/${instname}/solid.ini.`hostname      -s`
/opt/solid/${instname}/solid.ini
(csi1)#                       cp                       /opt/solid/${instname}/solid.ini
/opt/si3000/solid/${instname}/solid.ini.`hostname -s`


(csi2)# export restpath=/opt/bacula-restores/opt/archive/solid
(csi2)# export restsrv=vhp1
(csi2)# export instname=csiradius
(csi2)# crm resource stop c_${instname}
(csi2)# rm -f /opt/solid/${instname}/*.log
(csi2)# rm -f /opt/solid/${instname}/*.out
(csi2)# scp ${restsrv}:${restpath}/${instname}/solid.db /opt/solid/${instname}/
```

```
(csi2)#      scp      ${restsrv}:${restpath}/${instname}/solid.ini.`hostname      -s`
/opt/solid/${instname}/solid.ini
(csi2)#                      cp                      /opt/solid/${instname}/solid.ini
/opt/si3000/solid/${instname}/solid.ini.`hostname -s`
(csi2)# crm resource start c_${instname}
(csi2)# crm resource start c_csiradiusd


(csi1)# export instname=csimncomm
(csi1)# crm resource stop c_${instname}
(csi1)# rm -f /opt/solid/${instname}/*.log
(csi1)# rm -f /opt/solid/${instname}/*.out
(csi1)#              scp              ${restsrv}:${restpath}/${instname}/solidnet.cfg
/opt/si3000/solid/${instname}/
(csi1)# scp ${restsrv}:${restpath}/${instname}/solid.db /opt/solid/${instname}/
(csi1)#      scp      ${restsrv}:${restpath}/${instname}/solid.ini.`hostname      -s`
/opt/solid/${instname}/solid.ini
(csi1)#                      cp                      /opt/solid/${instname}/solid.ini
/opt/si3000/solid/${instname}/solid.ini.`hostname -s`


(csi2)# export instname=csimncomm
(csi2)# crm resource stop c_${instname}
(csi2)# rm -f /opt/solid/${instname}/*.log
(csi2)# rm -f /opt/solid/${instname}/*.out
(csi2)# scp ${restsrv}:${restpath}/${instname}/solid.db /opt/solid/${instname}/
(csi2)#      scp      ${restsrv}:${restpath}/${instname}/solid.ini.`hostname      -s`
/opt/solid/${instname}/solid.ini
(csi1)#                      cp                      /opt/solid/${instname}/solid.ini
/opt/si3000/solid/${instname}/solid.ini.`hostname -s`
(csi2)# crm resource start c_${instname}


(ap1)# crm resource start g_jboss


(ap2)# crm resource start g_jboss
```

После этого процедура восстановления базы Solid для MN и Radius завершается.

10.7.8.7 Восстановление сервера DNS

Предупреждение!

Перед началом восстановления сервера DNS его необходимо соответствующим образом настроить, следуя процедуре, использованной при первоначальной установке. Этот сервер должен быть активен.

В случае конфигурации с резервированием, убедитесь, что вы находитесь на активной стороне сервера DNS, прежде чем запускать процедуру восстановления. Проверьте это с помощью команды:

```
(vm1)# mount | grep named
```

Если вы не получили никакого ответа, перейдите на другой сервер платформы VHP (в паре DNS).

Восстановите нужный архив из архива Bacula в выбранное место (по умолчанию директория находится на VHP в /opt/bacula-restores). После завершения восстановления вы можете найти все необходимые файлы восстановления сервера DNS на активном сервере VHP для архивации по адресу /opt/bacula-restores/opt/archive/solid.

Прежде чем копировать файлы конфигурации и базы данных DNS, сначала остановите службу named следующим образом:

```
(vmX)# crm resource stop res_named_named
```

Затем:

- в конфигурации без резервирования:
  - скопируйте файлы **named.conf** и **db.*** в директорию **/opt/named/**
  - в директории **/opt/named/** удалите все файлы **db.*.jnl**
- в конфигурации с резервированием:
  - скопируйте файл **named.conf** в директорию **/etc/**
  - скопируйте файл **db.*** в директорию **/var/named/**
  - в директории **/var/named/** удалите все файлы **db.*.jnl**

После этого перезапустите службу named следующим образом:

```
(vmX)# crm resource start res_named_named
```

Проверьте работу сервера DNS командой (пример для конфигурации с резервированием):

```
(vmX)# host -al `hostname -d`
Trying "csi.iskrauraltel.ru"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14604
;; flags: qr aa ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;csi.iskrauraltel.ru.              IN      AXFR

;; ANSWER SECTION:
csi.iskrauraltel.ru.        86400   IN      SOA     planetdns.csi.iskrauraltel.ru. root. 4 10800
900 604800 10800
csi.iskrauraltel.ru.       86400   IN      NS      planetdns.csi.iskrauraltel.ru.
goricko.csi.iskrauraltel.ru. 86400 IN      A       172.18.251.20
jupiter.csi.iskrauraltel.ru. 86400 IN      A       172.18.251.4
piran.csi.iskrauraltel.ru. 86400   IN      A       172.18.251.10
planetbacula.csi.iskrauraltel.ru. 86400 IN A       172.18.251.15
planetdns.csi.iskrauraltel.ru. 86400 IN    A       172.18.251.13
planetldap.csi.iskrauraltel.ru. 86400 IN   A       172.18.251.14
planetnfs.csi.iskrauraltel.ru. 86400 IN    A       172.18.251.11
planetnginx.csi.iskrauraltel.ru. 86400 IN  A       172.18.251.12
saturn.csi.iskrauraltel.ru. 86400  IN      A       172.18.251.3
sun.csi.iskrauraltel.ru.   86400   IN      A       172.18.251.7
csi.iskrauraltel.ru.        86400   IN      SOA     planetdns.csi.iskrauraltel.ru. root. 4 10800
900 604800 10800

Received 373 bytes from 172.18.251.13#53 in 0 ms
```

11 Наблюдаемость

11.1 Функция наблюдения на сервере COS

11.1.1 Настройка модулей Grafana и EFK

Настройка модулей Grafana и EFK является частью процедуры настройки сервера COS (см. главу 4) на платформе ICP 4.2.

При необходимости перезапустить контейнер Grafana можно с помощью:

```
docker restart it_grafana
```

Перезапуск контейнеров EFK (Fluentd, Elasticsearch, Kibana) можно выполнить командой:

```
docker-compose -p cos_mon --env-file /opt/it-efk/efk.env -f /opt/it-efk/docker-compose.yml res
tart
```

11.1.2 Мониторинг

11.1.2.1 Подключение к модулю Grafana

Откройте браузер на машине, которая может подключаться к серверу BM операций через свой восходящий интерфейс (общедоступный интерфейс). Введите URL-адрес *<operations_vm_name_or_url>:3000*. Пароль для пользователя *admin* предоставляется системным администратором

Примите во внимание, что Grafana поддерживается в текущей версии следующих браузеров (при использовании модуля Grafana нужно всегда обновляться до последней версии):

- Chrome/Chromium,
- Firefox,
- Safari.
- Microsoft Edge.

Также в используемом браузере должен быть включен JavaScript. Запуск модуля Grafana без включенного JavaScript в браузере не поддерживается.

11.1.2.2 Добавление и удаление облака в Grafana

Запустите инструмент, который подготовит файл конфигурации для нового источника данных, и перезапустите контейнер Grafana:

```
cd /opt/it_observability/tools/

./grafana-add-datasource.sh -a -p <cloud_internal_float_ip> <cloud-name>
```

Для удаления облака из модуля Grafana используется параметр -r:

```
cd /opt/it_observability/tools/

./grafana-add-datasource.sh -r <cloud-name>
```

11.1.2.3 Добавление источника данных Openbaton в модуль Grafana

При использовании источника даннных Openbaton, установленного на модуль Swarm, существует возможность его мониторинга — через панель управления ICP OpenBaton. В модуле Swarm есть дополнительный и независимый сервер Prometheus, его необходимо вручную добавить в модуль Grafana, выбрав элементы:

 **> DataSources > AddDataSource > Prometheus**

Имя: Prometheus OB

URL-адрес: http://<prometheus_ip_address_on_swarm>:9090

Щелкните кнопку **Save & Test**, а затем выберите соответствующий источник на панели управления ICP OpenBaton.

11.1.2.4 Доступ через IP-адрес общедоступного облака

По умолчанию модуль Grafana использует внутреннюю сеть для доступа к модулю Prometheus. Если необходим доступ по внешнему IP-адресу, его необходимо настроить в источниках данных для конкретного облака, как показано на рисунке ниже.

В меню **Configuration** > **Data Sources** требуется выполнить следующие настройки: измените IP-адрес на внешний — соединение защищено, поэтому выберите «https» вместо «http», включите «Basic auth» и «Skip TLS Verify», добавьте данные аутентификации: имя пользователя и пароль, которые можно найти в файле /etc/kolla/haproxy/services.d/prometheus-server.cfg (раздел prometheus_server_external-user) на управляющем узле облака. После этого щелкните кнопку **Save & Test**.

Такую же процедуру необходимо выполнить и для источника данных Prometheus AlertManager. Убедитесь, что имя источника данных Prometheus AlertManager совпадает с именем источника данных Prometheus и заканчивается на «AlertManager», например:

- Имя источника данных Prometheus: ICP <cloud_name> Prometheus
- Имя Prometheus AlertManager: ICP <cloud_name> Prometheus AlertManager

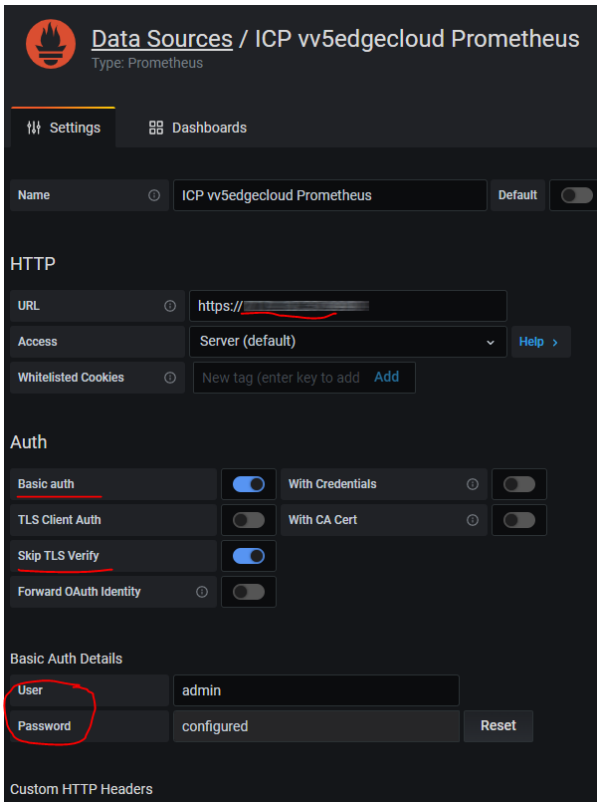


Рис. 11-1: Настройка внешнего доступа к Grafana

11.1.2.5 Метрики

Панель **ICP ALERTS** содержит текущие аварийные сигналы, отсортированные по срочности.

Панель **ICP Common** содержит наиболее важные данные мониторинга облака. Эта панель является основной и должна быть всегда открыта.

Дополнительные данные мониторинга отображаются на информационных панелях, которые предоставляются соответствующими экспортерами Prometheus. Каждая панель содержит несколько графиков, которые показывают разные метрики. На некоторых панелях графики могут быть сгруппированы в одну функциональную группу. Как правило, название графика уже содержит его значение.

Метрики узла:

Показатели оборудования и операционной системы узла отображаются на панели **ICP Node**.

Метрики объединены в несколько функциональных групп. В первой группе показаны основные данные об использовании ЦП, памяти, хранилища и сети для каждого хоста Openstack. Хост можно выбрать из выпадающего меню в верхнем левом углу окна браузера.

Подробные данные показаны на графиках в других функциональных группах. Чтобы выбрать нужную группу, щелкните значок «>».

Метрики конечных точек:

Показатели конечных точек отображаются на панели **ICP Blackbox**.

Для каждой конечной точки Openstack (внутренней, внешней и административной) можно просмотреть статус и время отклика.

Метрики Docker:

Показатели Docker отображаются на панели **ICP Docker**.

Для каждого контейнера показано использование ЦП, памяти, сети и хранилища. Эту статистику можно просмотреть для каждого узла ICP. Узел можно выбрать из выпадающего меню в верхнем левом углу окна браузера.

Метрики Openstack:

Показатели облака ICP отображаются на панели **ICP OpenStack**.

Эта панель содержит две группы:

- Service status: статус службы (статус агентов Nova, Neutron и Cinder).
- Resource usage: использование ресурсов (общее использование памяти и ЦП, базовая статистика Neutron и Keystone, количество виртуальных машин, томов, снимков и образов).

Метрики Ceph:

Показатели модуля Ceph отображаются на панели **ICP Ceph Cluster**.

Эта панель содержит несколько групп:

- Cluster state: состояние кластера (базовая информация о состоянии и доступности ресурсов Ceph).
- OSD state: состояние сущностей OSD (состояние и статистика сущностей OSD).
- Cluster: кластер (емкость и скорость отклика кластера Ceph).
- Latency: время отклика (время отклика сущностей OSD).
- Objects: объекты (количество объектов в кластере и состояние групп рамещения).
- Recovery: восстановление (выполнение операций восстановления).

Метрики MySQL:

Показатели БД MySQL отображаются на панели **ICP Mysql**.

Панель инструментов содержит несколько групп:

- Global Status: глобальный статус (количество подключений к БД MySQL).
- I/O: ввод-вывод (дисковая и сетевая статистика, частота запросов и активные потоки).
- Errors: ошибки (остановленные клиенты и соединения).
- Disk Usage: использование диска (использование диска для таблиц и индексов, сумма всех строк в БД).

Метрики HaProxy:

Показатели прокси HaProxy отображаются на панели **ICP HAProxy**.

Метрики объединены в несколько функциональных групп. В первую группу входят базовые данные HaProxy (состояние серверов, объем трафика, количество подключений, запросов, ответов и сессий) для каждого хоста. Хост можно выбрать из выпадающего меню в верхнем левом углу окна браузера.

Подробные данные показаны на графиках других функциональных групп. Чтобы выбрать группу, щелкните значок «**>**».

Метрики виртуальных машин:

Показатели виртуальных машин отображаются на панели **ICP Virtual Machines**.

Для каждой ВМ на платформе ICP показаны загрузка ЦП и памяти, а также активность диска (чтение/запись). Эту статистику можно просмотреть для каждого узла ICP. Узел можно выбрать из выпадающего меню в верхнем левом углу окна браузера. Также возможна фильтрация данных по экземпляру, если какой-либо экземпляр выбран из выпадающего меню.

Панель инструментов содержит несколько групп:

- Глобальная информация о всех ВМ (состояние ВМ – включена или выключена).
- Статус и информация (общий статус всех виртуальных машин, например, имя ВМ, используемый flavour и т.д.).
- Загрузка памяти и ЦП (общее количество виртуальных ЦП/ОЗУ, используемых на узле, распределение и использование ЦП/ОЗУ виртуальных машин).
- Данные ввода-вывода по дискам (загрузка и время чтения/записи на диске ВМ, размер образа, выделенный для каждого раздела ВМ).

Сетевые интерфейсы (сетевой трафик (в байтах), трафик отбрасывания/ошибок и PPS для каждого интерфейса виртуальной машины)

Метрики резервного копирования:

Метрики резервного копирования Bacula отображаются на панели управления **ICP Backup**.

Количество успешных и неудачных заданий резервного копирования можно отследить по метрикам резервного копирования, а также по общему объему данных и количеству файлов на выбранное задание резервного копирования. Тип резервной копии (приложение, инфраструктура) и задание Bacula можно выбрать в раскрывающемся меню в верхней части панели управления.

Метрики Openbaton:

Показатели Openbaton отображаются на панели **ICP OpenBaton**.

На панели ICP OpenBaton отображаются статус файла лицензии для оркестратора Openbaton и информация о сроке действия. Также доступна информация об ошибках в сервисах, количестве разрешенных и используемых экземпляров функций VNF и количестве запрошенных реплик.

11.1.3 Аварийные сигналы

Мониторинг инфраструктуры поддерживает аварийные сигналы для некоторых предопределенных метрик. Каждый аварийный сигнал содержит данные, необходимые для устранения обнаруженных неисправностей. Каждый аварийный сигнал выдается при превышении предопределенного порога и отменяется, когда измеренное значение опускается ниже порога.

Мониторинг другого оборудования от Искра Технологии (поддерживается оборудование версии 5, например, MPS, и диагностика оборудования версии 6 — например, IMS) следует настраивать вручную. Если необходимо контролировать какой-либо элемент vIMS, то IP-адрес управления элементом vIMS необходимо вручную добавить в файл `/opt/it-prometheus/prometheus_server/snmp_iskrauraltel_v6.yml` на сервере COS.

Аварийные сигналы отображаются на панели **ICP Alerts** и содержат следующие данные:

- "alertname" – название аварийного сигнала,
- "instance" – экземпляр или узел, на котором сгенерирован аварийный сигнал (информация добавляется компонентом Prometheus),
- "job" – задание, на котором сгенерирован аварийный сигнал (информация добавляется компонентом Prometheus),
- "monitor" – источник мониторинга, на котором сгенерирован аварийный сигнал (информация добавляется компонентом Prometheus),
- "description" – подробное описание аварийного сигнала и/или
- "severity" – срочность аварийного сигнала (critical/major/minor/warning/info),
- "priority" – (необязательно) дополнительный приоритет (high/medium/low),
- "info" – (необязательно) дополнительная информация,
- "summary" – (необязательно) сводка по аварийному сигналу.

Табл. 11-1: Список поддерживаемых аварийных сигналов

Имя / Местоположение аварийного сигнала *	Срочность	Время **	Описание	Инструкции по устранению
Общие				
Node unreachable [O] Узел недоступен	Крайне срочный	1 мин	Узел <instance_name> не работает	
Exporter Down [OC] Экспортер не работает	Предупреждение	10 мин	Экспортер в <instance_name> недоступен	
No connection to Alertmanager [OC] Нет подключения к Alertmanager	Предупреждение	5 мин	Prometheus на <instance_name> не подключен ни к одному диспетчеру аварийных сигналов	
Prometheus HTTP high response time [OC] Высокое время отклика Prometheus HTTP	Срочный	10 мин	Высокое время отклика Prometheus HTTP, более 100 мс	
Prometheus getting sluggish [OC] Низкая эффективность Prometheus	Несрочный	5 мин	Менее 80 % запросов к конечной точке <handler_name> обрабатываются менее чем за 100 мс	
Prometheus responding slowly [OC] Медленная работа Prometheus	Несрочный	5 мин	50% запросов заняли более 200 мс	
Рабочие характеристики				
High CPU Usage [OC] Высокая загрузка ЦП	Срочный	1 мин	<instance_name> использует много ресурсов ЦП (от 80% до 95%). Загрузка ЦП составляет <current_CPU_usage>% (средняя нагрузка за 5 минут)	
Very high CPU Usage [OC] Очень высокая загрузка ЦП	Крайне срочный	1 мин	<instance_name> использует много ресурсов ЦП (более 95%). Загрузка ЦП составляет <current_CPU_usage>% (средняя нагрузка за 5 минут)	
Server Overloaded [OC] Сервер перегружен	Срочный	5 мин	Процессов больше, чем могут обработать все ядра	
High Memory Usage [OC] Высокая загрузка памяти	Срочный	1 мин	Использование памяти в <instance_name> более 80%. В настоящее время <current_usage>%	
Very high Memory Usage [OC] Очень высокая загрузка памяти	Крайне срочный	1 мин	Использование памяти в <instance_name> более 95%. В настоящее время <current_usage>%	

Out Of Disk Space [OC] Недостаточно места на диске	Срочный	1 мин	Используется более 80% диска. Использование диска <current_usage>%	
Out Of Disk Space – critical [OC] Недостаточно места на диске — критично	Крайне срочный	1 мин	Используется более 95% диска. Использование диска <current_usage>%	
Unusually High Disk Read Rate [OC] Необычно высокая скорость чтения с диска	Срочный	15 мин	Возможно, диск считывает слишком много данных (> 50 МБ/с).	
Unusually High Disk Write Rate [OC] Необычно высокая скорость записи на диск	Срочный	15 мин	Диск, вероятно, записывает слишком много данных (> 50 МБ/с)	
Unusual Disk Read Latency [OC] Необычная задержка чтения диска	Несрочный	10 мин	Задержка диска растет (операции чтения > 100 мс)	
Unusual Disk Write Latency [OC] Необычная задержка записи на диск	Несрочный	10 мин	Задержка диска растет (операции записи > 100 мс)	
Swap utilization [OC] Использование подкачки	Срочный	1 мин	Более 80 % использования подкачки на Neutron	
High swap utilization [OC] Высокая загрузка подкачки	Крайне срочный	1 мин	Более 95 % использования подкачки на <instance_name>	
Disk Will Fill In 6 Hours [OC] Диск заполнится через 6 часов	Срочный	5 мин	Диск на <instance_name> заполнится примерно через 6 часов.	
HA-proxy				
High average HTTP connect time [OC] Высокое среднее время подключения HTTP	Срочный	10 мин	Сред. время подключения HTTP к серверной части <backend_name> для последних 1024 успешных подключений превышает пороговое значение в 500 мс.	
High average HTTP response time [OC] Высокое среднее время ответа HTTP	Срочный	10 мин	Сред. время подключения HTTP к серверной части <backend_name> для последних 1024 успешных подключений превышает пороговое значение в 1 с.	
HAproxy backend down [O] Серверная часть HAproxy не работает	Срочный	2 мин	Статус работоспособности серверной части <backend_name> — DOWN.	
HAproxy server down [O] Сервер HAProxy не работает	Срочный	2 мин	Состояние работоспособности сервера <server_name> для серверной части <backend_name> — DOWN.	

HAproxy intense CPU usage [O] Высокая загрузка ЦП HAproxy	Несрочный	1 мин	Загрузка ЦП HAproxy на <instance_name> превышает 80 %.	
Openstack				
Cinder agent disabled [O] Агент Cinder отключен	Срочный	5 мин	Агент Openstack Cinder отключен на <node_name>	
Neutron agent down [O] Агент Neutron не работает	Срочный	5 мин	Агент Openstack Neutron отключен на <node_name> <service_name>	
Nova agent down [O] Агент Nova не работает	Срочный	5 мин	Агент Openstack nova не работает на <node_name> <service_name>	
Nova local storage almost full [O] Локальное хранилище Nova почти заполнено	Срочный	5 мин	Локальное хранилище Openstack почти заполнено на <node_name>	
Nova high memory usage [O] Высокая загрузка памяти Nova	Срочный	5 мин	Высокое использование памяти Openstack Nova на <node_name>	
Openstack intense CPU usage [O] Высокая загрузка ЦП Openstack	Несрочный	1 мин	Загрузка ЦПУ Openstack на <instance_name> превышает 80 %.	
Контейнеры				
Container high CPU usage [O] Высокая загрузка ЦП контейнера	Крайне срочный	2 мин	Контейнер <container_name> на <node_name> использует более 90% ЦП	Проверьте журналы для контейнера, восстановите или перезапустите его
Container high read rate Высокая скорость чтения контейнера	Срочный	2 мин	Контейнер <container_name> на <node_name> имеет высокую скорость чтения (> 100 IOPS)	
Container high write rate [O] Высокая скорость записи контейнера	Срочный	2 мин	Контейнер <container_name> на <node_name> имеет высокую скорость записи (> 100 IOPS)	
Docker down [C] Докер не работает	Срочный	1min	Докер <docker_name> не работает.	Проверьте и перезапустите докер в мониторинге или на ВМ операций. Примечание. Отслеживаются только докеры наблюдаемости (например, elasticsearch, fluentd, grafana, kibana).
Ceph				
CephTargetDown [O] Цель Ceph не работает	Крайне срочный	2 мин	Цель CEPH недоступна более 2 минут, пожалуйста, проверьте ее – это может быть либо сбой экспортера, либо сбой всего кластера.	

CephErrorState [O] Ceph в состоянии ошибки	Крайне срочный	2 мин	Ceph находится в состоянии ошибки более 2 минут, проверьте состояние пулов и OSD	Проверьте состояние пулов и OSD
CephWarnState [O] Ceph в состоянии предупреждения	Предупреждение	2 мин	Ceph находится в состоянии предупреждения более 2 минут, проверьте состояние пулов и OSD	Проверьте состояние пулов и OSD
OsdDown [O] OSD не работает	Предупреждение	2 мин	OSD не работает более 2 минут, пожалуйста, проверьте статус	Проверьте журналы, замените диск в случае его выхода из строя
OsdApplyLatencyTooHigh [O] Слишком высокая задержка OSD	Предупреждение	90 с	Задержка OSD для <osd_num> слишком велика (> 10 с). Пожалуйста, проверьте, не завис ли он	Проверьте состояние ceph, возможно, слишком высокая нагрузка, или необходимо создать раздел для журнала на SSD.
MonitorClockSkewTooHigh [O] Расфазировка тактовых сигналов	Предупреждение	60 с	Обнаружена расфазировка тактовых сигналов на <ceph_mon_num> — проверьте настройки часов NTP и оборудования	Проверьте и настройте NTP
MonitorAvailableStorage [O] Мало свободного места	Предупреждение	60 с	Свободного места в хранилище мониторинга для <ceph_mon_num> менее 30 %	Проверьте доступное свободное место, при необходимости добавьте дополнительный OSD.
MonitorAvailableStorage [O] Мало свободного места	Крайне срочный	60 с	Свободного места в хранилище мониторинга для <ceph_mon_num> менее 15 %	Проверьте доступное свободное место, при необходимости добавьте дополнительный OSD.
CephOSDUtilizatoin [O] Мало свободного места для Ceph OSD	Крайне срочный	60 с	Хранилище OSD для <osd_num> заполнено более чем на 90%. Перевзвесьте или добавьте хранилище	Если некоторые из OSD заполнены, перевзвесьте их, если все заполнены, добавьте еще один.
CephPgDown [O] Группы размещения не работают	Крайне срочный	2 мин	Некоторые группы слишком долго не работают на ceph. Убедитесь, что все данные доступны	Проверьте причину сбоя группы размещения командой: «ceph pg x.y query».
CephPgIncomplete [O] Группы размещения не заполнены	Крайне срочный	2 мин	Некоторые группы слишком долго являются неполными на ceph. Убедитесь, что все данные доступны	Проверьте причину незаполненности группы размещения командой: «ceph pg x.y query».
CephPgInconsistent [O] Группы размещения не согласованы	Предупреждение	6 с	Некоторые группы слишком долго не согласованы на ceph. Данные доступны, но не согласованы между узлами	Попробуйте исправить группы размещения командой: "ceph pg repair x.y"
CephPgActivating [O] Группы размещения не активируются	Крайне срочный	3 мин	Некоторые группы слишком долго активируются на ceph	Проверьте причину активации группы размещения командой: «ceph pg x.y query».
CephPgBackfillTooFull [O] Группы размещения на заполненном OSD	Предупреждение	3 мин	Некоторые группы расположены на заполненном OSD на ceph. Эти группы могут быть недоступны в ближайшее время. Пожалуйста, проверьте сущности	Проверьте сущности OSD, перевзвесьте их или перенастройте правила CRUSH.

			OSD, перевзвесьте их или перенастройте правила CRUSH.	
CephPgUnavailable [O] Группы размещения недоступны	Крайне срочный	2 мин	Некоторые группы недоступны на ceph. Пожалуйста, проверьте их статус и текущую конфигурацию.	
CephOsdReweighted [O] Ceph OSD повторно взвешивается	Предупреждение	1 ч	OSD <osd_num> слишком долго повторно взвешивается в ceph. Пожалуйста, либо создайте silent, либо исправьте эту проблему	
Сеть				
Network interface down [OC] Сетевой интерфейс не работает	Крайне срочный	2 мин	Интерфейс <instance_name> отключен на <network_device>.	Проверьте сетевой интерфейс на проблемном узле и отремонтируйте его (замените кабель, подключаемый кабель)
Network interface high load [OC] Высокая нагрузка на сетевой интерфейс	Срочный	2 мин	Сетевой интерфейс <network_device> на <instance_name> перегружен (более 80%).	На основе сообщенного интерфейса проанализируйте, какой трафик слишком интенсивно использует сеть.
Network receive errors [OC] Ошибки приема по сети	Срочный	2 мин	Интерфейс <instance_name> на <network_device> получал ошибки приема за последние пять минут.	Возможная проблема с кабелем/разъемом – проверьте и замените его.
Network transmit errors [OC] Ошибки передачи по сети	Срочный	2 мин	Интерфейс <instance_name> на <network_device> получал ошибки передачи за последние пять минут.	Возможная проблема с кабелем/разъемом – проверьте и замените его.
Network receive drop packets [OC] Сеть принимает отброшенные пакеты	Срочный	2 мин	Интерфейс <instance_name> на <network_device> получал отброшенные пакеты за последние пять минут.	Возможная проблема с кабелем/разъемом – проверьте и замените его.
Network transmit drop packets [OC] Сеть передает отброшенные пакеты	Срочный	2 мин	Интерфейс <instance_name> на <network_device> передавал отброшенные пакеты за последние пять минут.	Возможная проблема с кабелем/разъемом – проверьте и замените его.
Network transmit collisions [OC] Коллизии передачи по сети	Срочный	2 мин	Интерфейс <instance_name> на <network_device> получал коллизии при передаче за последние пять минут.	Возможная проблема с кабелем/разъемом – проверьте и замените его.
Лицензии				
License not valid [C] Лицензия недействительна [C]	Срочный	1 мин	Лицензия на облако <cloud_name> недействительна	Обратитесь в службу лицензирования и продлите лицензию
License expired [C] Срок действия лицензии истек [C]	Срочный	1 мин	Срок действия лицензии на облако <cloud_name>	Обратитесь в службу лицензирования и продлите лицензию

License will expire soon [C] Срок действия лицензии скоро истечет [C]	Несрочный	1 мин	Срок действия лицензии на облако <cloud_name> истечет через <time>	Обратитесь в службу лицензирования и продлите лицензию
Unlicensed geo COS server setup [C] Установлен нелицензионный сервер GEO COS [C]	Срочный	1 мин	Установлен нелицензионный сервер GEO COS в облаке в облаке <cloud_name>	Обратитесь в службу лицензирования и получите новую лицензию
Unlicensed hsb COS server setup [C] Установлен нелицензионный сервера HSB COS [C]	Срочный	1 мин	Установлен нелицензионный сервер HSB COS в облаке <cloud_name>	Обратитесь в службу лицензирования и получите новую лицензию
Too many deployed nodes [C] Слишком много развернутых узлов [C]	Срочный	1 мин	В облаке <cloud_name> слишком много развернутых узлов <nr_of_nodes>	Обратитесь в службу лицензирования и продлите лицензию
Major upgrade required [C] Требуется срочное обновление [C]	Срочный	1 мин	Требуется срочное обновление в облаке <cloud_name> начиная с <time>	Обратитесь в службу лицензирования и продлите лицензию
Major upgrade required soon [C] Скоро потребуется срочное обновление [C]	Несрочный	1 мин	Требуется срочное обновление в облаке <cloud_name> в <time>	Обратитесь в службу лицензирования и продлите лицензию
COS exporter config file missing [C] Отсутствует файл конфигурации экспортера COS [C]	Несрочный	1 мин	Отсутствует файл конфигурации экспортера COS.	Проверьте файл конфигурации экспортера COS на ВМ операций
COS exporter config file syntax error [C] Ошибка синтаксиса файла конфигурации экспортера COS [C]	Срочный	1 мин	Синтаксическая ошибка в файле конфигурации экспортера COS.	Проверьте файл конфигурации экспортера COS на ВМ операций
Big yaml config file missing [C] Отсутствует файл конфигурации big yaml [C]	Срочный	1 мин	Отсутствует файл конфигурации big yaml	Проверьте файл конфигурации big yaml на ВМ операций.
Big yaml config file syntax error [C] Ошибка синтаксиса файла конфигурации big yaml [C]	Срочный	1 мин	Файл конфигурации big yaml не является допустимым файлом yaml	Проверьте файл конфигурации big yaml на ВМ операций.
cloud_name var in big yaml config file missing [C]	Срочный	1 мин	Переменная cloud_name отсутствует в файле конфигурации big yaml	Проверьте файл конфигурации big yaml на ВМ операций.

cloud_name var в файле конфигурации big yaml отсутствует [C]				
license_file var in big yaml config file missing [C] license_file var в файле конфигурации big yaml отсутствует [C]	Срочный	1 мин	Переменная license_file отсутствует в файле конфигурации big yaml	Проверьте файл конфигурации big yaml на ВМ операций.
Licensing file missing [C] Отсутствует файл лицензии [C]	Срочный	1 мин	Файл лицензии для <cloud_name> отсутствует	Проверьте расположение файла лицензии в big.yaml на ВМ операций.
Licensing file syntax error [C] Ошибка синтаксиса файла лицензии [C]	Срочный	1 мин	Файл лицензии не является допустимым файлом лицензии	Обратитесь в службу лицензирования и получите новый файл лицензии.
Оборудование Искра Технологии				
Critical equipment alerts [C] Крачне срочные аварийные сигналы оборудования Искра Технологии [C]	Крайне срочный	1 мин	Крайне срочный аварийный сигнал <alarmSpecificProblem> в экземпляре <instance> - <alarmDescription>	Выясните, что пошло не так — не является частью инфраструктуры ICP
Major equipment alerts [C] Срочные аварийные сигналы оборудования Искра Технологии [C]	Срочный	1 мин	Срочный аварийный сигнал <alarmSpecificProblem> в экземпляре <экземпляр> - <alarmDescription>	Выясните, что пошло не так — не является частью инфраструктуры ICP
Minor equipment alerts [C] Несрочные аварийные сигналы оборудования Искра Технологии [C]	Несрочный	1 мин	Несрочный аварийный сигнал <alarmSpecificProblem> в экземпляре <экземпляр> - <alarmDescription>	Выясните, что пошло не так — не является частью инфраструктуры ICP
Warning equipment alerts [C] Предупреждения оборудования Искра Технологии [C]	Предупреждение	1 мин	Предупреждение <alarmSpecificProblem> в экземпляре <instance> - <alarmDescription>	Выясните, что пошло не так — не является частью инфраструктуры ICP
Indeterminate equipment alerts [C] Неопределенные аварийные сигналы оборудования Искра Технологии [C]	Предупреждение	1 мин	Неопределенный аварийный сигнал <alarmSpecificProblem> в экземпляре <instance> - <alarmDescription>	Выясните, что пошло не так — не является частью инфраструктуры ICP
Виртуальные машины на COS				
VM high CPU Usage [CVM] Высокая загрузка ЦП ВМ [CVM]	Срочный	1 мин	ВМ <vm_id> использует более 90 % ЦП (средняя нагрузка за 5 минут)	Проверьте активность процесса на конкретной ВМ

VM Unusually High Disk Read Rate [CVM] ВМ имеет необычно высокую скорость чтения с диска [CVM]	Срочный	1 мин	Необычная скорость чтения диска на ВМ <vm_id> (> 50 МБ/с)	Проверьте активность диска на конкретной ВМ
VM Unusually High Disk Write Rate [CVM] ВМ имеет необычно высокую скорость записи на диск [CVM]	Срочный	1 мин	Необычная скорость записи на диск на ВМ <vm_id> (> 50 МБ/с)	Проверьте активность диска на конкретной ВМ
Другие аварийные сигналы				
Internal DNS not responding [C] Внутренний DNS не отвечает [C]	Срочный	1 мин	Внутренний сервер DNS не отвечает	Проверьте внутренний сервер DNS (обычно он находится на ВМ FreeIPA).
Free IPA service not responding [C] Сервис Free IPA не отвечает [C]	Срочный	1 мин	Служба Free IPA <service_name> не отвечает	Проверьте сервисы FreeIPA (выполните команду "ipactl statsus" на ipasrv для запуска)
Bacula backup failure [C] Ошибка резервного копирования Bacula [C]	Срочный		Последнее резервное копирование Bacula не удалось	Не удалось выполнить резервное копирование, инициированное Bacula. Проверьте логи Bacula.

*Обозначения местоположения аварийного сигнала: O — узел ICP; C – сервер COS; OC — узел ICP и сервер COS; CVM — виртуальные машины на сервере COS

** необходимая продолжительность превышения порога

11.1.4 Централизованное логирование

11.1.4.1 Подключение к модулю Kibana

Откройте браузер на машине, которая может подключаться к серверу ВМ операций через свой восходящий интерфейс (общедоступный интерфейс). Введите URL-адрес *<operations_vm_name_or_ip>:5601*.

11.1.4.2 Поддержка логирования в нескольких облаках

На облачных узлах должны быть определены все места назначения для передачи файлов журнала. В файле `/etc/kolla/fluentd/td-agent.conf` добавьте еще один блок `<server>`, если вы хотите, чтобы журналы отправлялись на несколько виртуальных машин мониторинга, например:

```
root@vv5edgecloud-node1:~# cat /etc/kolla/fluentd/td-agent.conf
@include input/*.conf
@include filter/00-*.conf
@include filter/02-*.conf
@include format/*.conf
@include output/*.conf

<match *.**>
  @type forward
  <server>
    host 172.18.254.64
    port 24224
  </server>
  <server>
    host 172.18.254.48
    port 24224
  </server>
  <buffer>
    flush_interval 1s
  </buffer>
</match>
```

11.1.4.3 Использование модуля Kibana

В модуле Kibana элементы поиска выделяются в списке документов, отображаемых после выполнения поиска.

Модуль Kibana широко использует агрегации и субагрегации Elasticsearch для различных типов визуализаций. В основном используется два типа агрегаций (группирование и метрики). Группирование обеспечивает список групп, каждая из которых содержит набор принадлежащих ей документов, например: термины, диапазон, гистограммы и т.д. Метрики обеспечивают показатели вычислений для набора документов, такие как минимум, максимум, сумма, среднее и т.д., но эти типы вычислений можно выполнять только для полей числового типа. Также используются заскриптованные поля для выполнения вычислений на лету с индексированными данными. Например, если есть определенное поле, которое вы хотите умножить на 100, прежде

чем показывать его, вы можете сохранить его как заскриптованное поле. Однако поиск по таким полям невозможен.

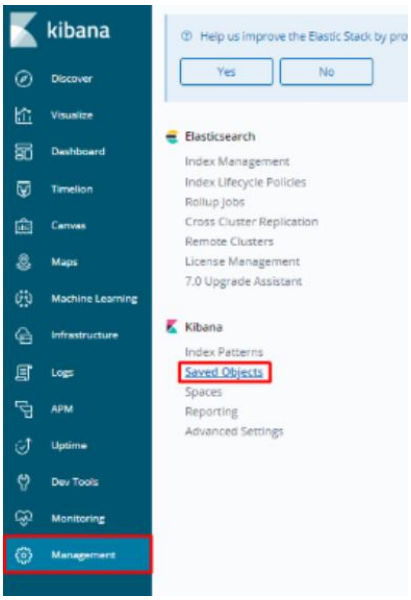
Информационные панели Kibana являются очень гибкими и динамичными, поскольку отдельные визуализации можно легко упорядочить по своему усмотрению, а дату можно обновлять автоматически.

Интерфейс Kibana содержит четырех основных вкладки:

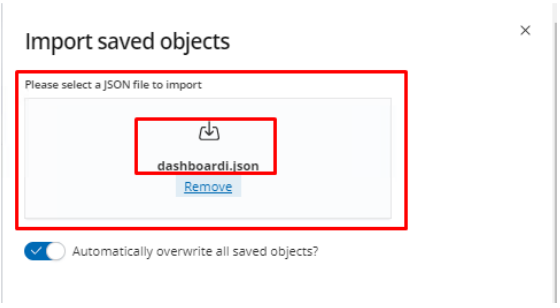
- Страница Discover позволяет выполнять поиск по произвольному тексту, поиск по полю, поиск по диапазону и т.д.
- Страница Visualize позволяет создавать множество визуализаций, таких как круговая диаграмма, столбчатая диаграмма и т.д. Эти визуализации можно использовать на информационных панелях.
- Страница Dashboard представляет собой набор нескольких визуализаций и поисковых полей, которые можно использовать для легкого применения фильтров и анализа результатов на основе нескольких наборов агрегированных данных.
- Страница Settings позволяет настраивать шаблоны индексов, заскриптованные поля, типы данных полей и т.д.

Импорт панелей в модуле Kibana

В левом меню выберите вкладку Management, а затем Saved Objects.



Щелкните кнопку Import, а затем перетащите файл Kibana.json в окно (файл Kibana.json находится внутри проекта it_observability).



Затем щелкните кнопку Import внизу страницы.

После импорта вы увидите информационные панели, поля поиска и т.д.

12 Приложение A: Лицензирование платформы ICP

Некоторые функции платформы ICP лицензируются. Эти функции:

- количество развернутых узлов в облаке,
- конфигурация сервера облачных операций (автономная, горячее резервирование, георезервирование),
- время до необходимого крупного обновления.

Все эти параметры прописаны в файле лицензий, который должен быть на сервере COS. Этот файл должен быть получен от Искра Технологии.

Краткое описание процедуры лицензирования:

1. Получите файл лицензии от Искра Технологии.
2. добавьте расположение файла в файл "big.yaml" в общем разделе, параметр license_file. Предлагаемое местоположение — /opt/cloud-cfg/<cloud-name>. Пример: license_file: '/opt/cloud-cfg/testedge/license.txt'
3. Скопируйте файл в указанное место.
4. Продолжите развертывание или реконфигурацию, как описано в соответствующих разделах.
5. Если количество узлов больше, чем указано в файле лицензии, то развертывание или реконфигурация останавливается.

Текущие значения параметров лицензирования можно увидеть с помощью инструмента license_info. Вы должны указать путь к файлу big.yaml следующим образом:

```
# license_info -y /opt/cloud-cfg/testedge/testedge-big.yaml

-----------------------------------------------------------------

Cloud name              : testedge

Big yaml file           : /opt/cloud-cfg/testedge/testedge-big.yaml

License file location   : /opt/cloud-cfg/testedge/license.txt

-----------------------------------------------------------------

License created on      : 2021-06-16 00:00:00

License expires on      : 2100-12-31 00:00:00

License is valid        : YES (for 28982 days)

-----------------------------------------------------------------

License order number    : O

License serial number   : L

License node name       : Maket RD

-----------------------------------------------------------------

Number of licensed nodes: 5

COS setup mode          : geo

-----------------------------------------------------------------
```

Два других параметра лицензирования (конфигурация сервера COS и время до крупного обновления) периодически проверяются, и в случае их нарушения генерируется аварийный сигнал (еще не реализовано на момент написания).

Крупное обновление требуется через два года после создания файла лицензии. По истечении этого периода необходимо получить новый файл лицензии и скопировать его на сервер COS.

———— (КОНЕЦ ДОКУМЕНТА) ————