

SP5000 ICP «Интеллектуальная облачная платформа»

Руководство администратора

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

1	О документе	6
1.1	Назначение	6
1.2	Список сокращений	6
2	Инструкции по администрированию	8
2.1	Перенастройка платформы ICP	8
2.1.1	Поиск файлов конфигурации в папке cloud-cfg	8
2.1.2	Редактирование файлов конфигурации в папке cloud-cfg	9
2.1.2.1	Для каждой службы	9
2.1.2.2	Для каждой службы на указанном хосте	9
2.1.2.3	Глобальные изменения	9
2.1.2.4	Редактирование файла <i>policy.json</i>	9
2.1.3	Подготовка среды развертывания	9
2.1.4	Запуск реконфигурации облачных служб	10
2.1.5	Проверка реконфигурации облачной службы	10
2.2	Добавление облачного вычислительного узла	10
2.2.1	Поиск файлов конфигурации в папке cloud-cfg	11
2.2.2	Редактирование файлов конфигурации в папке cloud-cfg	11
2.2.3	Подготовка среды развертывания	12
2.2.4	Загрузка хостов и установка операционной системы	12
2.2.5	Запуск нового облачного вычислительного узла	12
2.2.6	Проверка нового облачного вычислительного узла	13
2.2.7	Повышение количества групп размещения <i>serp</i>	14
2.3	Удаление облачного вычислительного узла	17
2.3.1	Поиск файлов конфигурации в папке cloud-cfg	18
2.3.2	Запуск удаления облачного вычислительного узла	18
2.3.3	Изменение файлов конфигурации в папке cloud-cfg	18
2.3.4	Подготовка среды развертывания	19
2.3.5	Обновление развернутого облака	19
2.3.6	Проверка удаления облачного вычислительного узла	20
2.4	Принудительное удаление облачного вычислительного узла	20
2.4.1	Поиск файлов конфигурации в папке cloud-cfg	21
2.4.2	Изменение файлов конфигурации в папке cloud-cfg	21
2.4.3	Подготовка среды развертывания	22
2.4.4	Принудительный запуск удаления облачного вычислительного узла	22
2.4.5	Обновление развернутого облака	23
2.5	Добавление облачного обслуживающего узла	23
2.5.1	Поиск файлов конфигурации в папке cloud-cfg	23
2.5.2	Изменение файлов конфигурации в папке cloud-cfg	24
2.5.3	Подготовка среды развертывания	25
2.5.4	Загрузка хостов и установка операционной системы	25
2.5.5	Запуск добавления облачного сервисного узла	25
2.5.6	Проверка добавления облачного сервисного узла	26
2.6	Удаление облачного сервисного узла	29
2.6.1	Поиск файлов конфигурации в папке cloud-cfg	30
2.6.2	Запуск удаления облачного сервисного узла	30
2.6.3	Изменение файлов конфигурации в папке cloud-cfg	31

2.6.4	Подготовка среды развертывания.....	32
2.6.5	Обновление развернутого облака	32
2.6.6	Проверка удаления облачного сервисного узла.....	32
2.7	Принудительное удаление облачного сервисного узла.....	33
2.7.1	Поиск файлов конфигурации в папке cloud-cfg.....	33
2.7.2	Изменение файлов конфигурации в папке cloud-cfg	34
2.7.3	Подготовка среды развертывания.....	35
2.7.4	Запуск принудительного удаления облачного сервисного узла.....	35
2.7.5	Обновление развернутого облака	36
2.8	Добавление облачных сервисных узлов от универсального (автономного) облака к многорежимному	36
2.8.1	Поиск файлов конфигурации в папке cloud-cfg.....	37
2.8.2	Изменение файлов конфигурации в папке cloud-cfg	37
2.8.3	Подготовка среды развертывания.....	38
2.8.4	Загрузка хостов и установка операционной системы.....	38
2.8.5	Запуск добавления облачных сервисных узлов	38
2.8.6	Проверка добавления облачных сервисных узлов.....	39
2.9	Незначительное обновление облака	41
2.9.1	Описание	41
2.9.2	Процедура обновления	42
2.10	Переключение репозитория.....	42
2.11	Обновление операционной системы на узлах.....	44
2.12	Обновление версии операционной системы на узлах	44
2.13	Реконфигурация облачных интерфейсов external_api.....	45
2.13.1	Поиск файлов конфигурации в папке cloud-cfg.....	45
2.13.2	Подготовка файла развертывания yam1	45
2.13.3	Подготовка среды развертывания.....	46
2.13.4	Подготовка сертификата	47
2.13.4.1	Повторное создание самозаверяющего сертификата	47
2.13.4.2	Обновление сертификата	47
2.13.5	Запуск реконфигурации облачных интерфейсов external_api.....	47
3	Настройка безопасности TLS	48
3.1	Поиск файлов конфигурации в папке cloud-cfg.....	48
3.2	Изменение файлов конфигурации в папке cloud-cfg	48
3.3	Подготовка среды развертывания.....	50
3.4	Подготовка сертификата.....	50
3.4.1	Создание самоподписанных сертификатов.....	50
3.4.2	Прямо указанный сертификат	51
3.5	Подготовка среды развертывания.....	51
3.6	Обновление развернутого облака	52
3.7	Проверка состояния облака	52
4	Устранение ошибок	53
4.1	Контролируемый перезапуск служб облака	53
4.2	Исправление службы dokcerd.....	54
4.3	Изменение количества реплик в пуле Serp.....	54
4.4	Неустранимый сбой облачного вычислительного узла	55
4.5	Неустранимый сбой облачного сервисного узла	55
4.6	Сбой сети	56
4.6.1	Проверка физической сети и оборудования	56

4.6.2	Проверка сетевых служб.....	57
4.6.3	Проверка конфигурации сети	58
4.6.4	Устранение неполадок сетевого пути.....	58
4.6.5	Устранение ошибок Open vSwitch.....	64
4.7	Восстановление диска экземпляра Nova.....	65
4.7.1	Автоматическая процедура.....	65
4.7.2	Ручная процедура	66
4.8	Восстановление тома экземпляра Nova.....	67
4.8.1	Автоматическая процедура.....	68
4.8.2	Ручная процедура	68
4.9	Восстановление всех образов RBD	70
4.9.1	Автоматическая процедура.....	70
4.10	Включение платформы после незапланированного отключения платформы	71
4.11	Сброс облачного узла с полностью занятым пространством подкачки.....	75

Список рисунков

Рис. 2-1: Расчет количества групп размещения	15
--	----

Список таблиц

Табл. 1-1: Список сокращений	6
Табл. 2-1: Поля для переключения репозитория	43

1 О документе

1.1 Назначение

Настоящий документ содержит инструкции по администрированию 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 сокращений

Табл. 1-1: Список сокращений

BM	виртуальная машина
ГБ	гигабайт
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
DB	DataBase база данных
DHCP	Dynamic Host Configuration Protocol протокол динамического конфигурирования сервера
FAI	Fully Automatic Installation полностью автоматическая установка
FQDN	Fully Qualified Domain Name полное доменное имя
HA	High Availability высокая доступность
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
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
PCS	Pacemaker/Corosync Configuration System система конфигурации Pacemaker/Corosync
PXE	Preboot Execution Environment среда для загрузки компьютеров с помощью сетевой карты
RAID	Redundant Array of Independent Disks избыточный массив независимых дисков
ROC	Reference Openstack Cloud эталонное облако Openstack
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
TFTP	Trivial File Transfer Protocol простой протокол передачи файлов
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
VM	Virtual Machine виртуальная машина
YAML	YAML Ain't Markup Language формат сериализации данных YAML

2 Инструкции по администрированию



Предупреждение!

Все процедуры по настройке платформы ICP должны проводиться отдельно, чтобы избежать конфликтов между конфигурациями. Таким образом, не допускается настройка нескольких разных процедур и их одновременное выполнение.

2.1 Перенастройка платформы ICP

Предварительные условия для перенастройки (реконфигурации) платформы ICP:

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

Сервисы платформы ICP можно перенастраивать после их развертывания с помощью компонента Kolla. Измененный файл конфигурации сервисов необходимо поместить на сервер управления облаком в папку `/etc/kolla/cloud-cfg/<cloud-name>/config/<service>/<config_file>`, где он будет обнаружен контейнером `kolla-ansible-srv`. Изменения можно вносить на следующих базисах: для каждого проекта, для каждой службы или для каждой службы на указанном хосте. Это относится только к файлам типа INI.

Если необходимо настроить файл политики `policy.json`, нужно создать полноценный файл полной политики для конкретного проекта в том же каталоге, что указано выше, и контейнер `kolla-ansible-srv` перезапишет этим файлом файл политики по умолчанию. Нужно учитывать, что в некоторых проектах файл полной политики хранится в исходном коде, его просто нужно скопировать, но в каких-то проектах политики по умолчанию задаются в кодовой базе, в этом случае файл нужно сгенерировать.

Модуль Kolla позволяет глобально переопределять конфигурацию для всех сервисов.



Предупреждение!

Любая процедура по перенастройке платформы должна выполняться с осторожностью, особенно если она выполняется в облаке с активными экземплярами. Некоторые процедуры могут привести к сбою приложений в экземплярах, сбоям в сетевых связях экземпляров, вызвать проблемы с безопасностью или даже привести к сбою сервисов облака. Для подробной информации об устранении неисправностей конкретных параметров конфигурации см. главу 4.

2.1.1 Поиск файлов конфигурации в папке cloud-cfg

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке `/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.

`<additional-configuration-files>` — все остальные файлы конфигурации.

2.1.2 Редактирование файлов конфигурации в папке `cloud-cfg`

2.1.2.1 Для каждой службы

Чтобы переопределить параметр `block_device_allocate_retries` в модуле Nova Compute, необходимо создать или отредактировать файл `/opt/cloud-cfg/<cloud-name>/config/nova/nova-compute.conf` с содержимым:

```
[DEFAULT]
```

```
block_device_allocate_retries = 678
```

Чтобы определить диапазоны VLAN для сетей провайдера, необходимо создать или отредактировать файл `/opt/cloud-cfg/<cloud-name>/config/neutron/ml2_conf.ini` с содержимым:

```
[ml2_type_vlan]
```

```
network_vlan_ranges = physnet1:700:800
```

2.1.2.2 Для каждой службы на указанном хосте

Чтобы настроить соотношение распределения ЦП и ОЗУ вычислительного узла только на хосте `<cloud-name>-node1`, необходимо создать файл `/opt/cloud-cfg/<cloud-name>/config/nova/<cloud-name>-node1/nova.conf` с содержимым:

```
[DEFAULT]
```

```
cpu_allocation_ratio = 16.0
```

```
ram_allocation_ratio = 5.0
```

2.1.2.3 Глобальные изменения

Модуль Kolla позволяет глобально переопределять конфигурацию для всех служб. Для этого модуля нужно предоставить файл `/opt/cloud-cfg/<cloud-name>/config/global.conf`.

Например, чтобы изменить размер пула базы данных для всех сервисов, необходимо создать файл `/opt/cloud-cfg/<cloud-name>/config/global.conf` с содержимым:

```
[database]
```

```
max_pool_size = 100
```

2.1.2.4 Редактирование файла `policy.json`

Чтобы перезаписать файл `policy.json` проекта Neutron, необходимо в файле `policy.json` из исходного кода проекта Neutron обновить правила и затем поместить его в папку `/opt/cloud-cfg/<cloud-name>/config/neutron/policy.json`.

2.1.3 Подготовка среды развертывания

После того, как все необходимые настройки облака выполнены, запустите скрипт BASH на сервере управления облаком:

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

Он повторно сгенерирует файлы *globals.yml* и *cloud-inventory* и скопирует дополнительные файлы конфигурации в папку *etc (/opt/etc-kolla/<cloud-name>)*. Эта папка также сопоставляется с контейнером *kolla-ansible-srv* как */etc/kolla/cloud-cfg/<cloud-name>/*.

2.1.4 Запуск реконфигурации облачных служб

Чтобы внести в службы изменения после того, как они уже были развернуты, запустите скрипт BASH на сервере управления облаком:

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

2.1.5 Проверка реконфигурации облачной службы

Чтобы выполнить проверку состояния и работоспособности реконфигурации той или иной службы облака:

1. Проверьте, запущены ли все необходимые службы OpenStack.
2. Проверьте конфигурацию сервисов OpenStack, которые были перенастроены.

2.2 Добавление облачного вычислительного узла

Предварительные условия:

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

Вы можете добавить новый вычислительный узел или узел хранения после того, как вы выполнили развертывание облака, например, чтобы увеличить емкость или заменить существующий узел. Вновь добавленный узел будет иметь только необходимые службы Nova, Neutron, Cinder и Ceph для запуска экземпляров Nova.

Сначала необходимо добавить или изменить параметры конфигурационного пространства модуля FAI в соответствии со спецификациями оборудования нового узла. ОС нового узла должна быть установлена с помощью модуля FAI.

Затем нужно добавить или изменить файлы *globals.yml* и *cloud-inventory*. Службы OpenStack нового узла должны быть установлены с помощью скриптов *itkf*.**.

Наконец, нужно проверить, достаточно ли в файле лицензии узлов для планируемого обновления. Самый простой способ сделать это – с помощью команды `license_info`:

```
# license_info -y /opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yaml
```

```
...
```

```
Number of licensed nodes: 5
```

Если количество узлов недостаточно, необходимо получить новый файл лицензии от производителя продукта.

2.2.1 Поиск файлов конфигурации в папке cloud-cfg

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке `/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.

`<additional-configuration-files>` — все остальные файлы конфигурации.

2.2.2 Редактирование файлов конфигурации в папке cloud-cfg

Чтобы подготовиться к добавлению вычислительного узла, сначала нужно проверить следующие строки в `/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml`:

```
.
.
#FAI specific
fai_boot:                #boot information for FAI installation
  hosts: 1                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
    - 08:94:ef:60:0d:f3
.
.
.
inventory:
  controllers: '1'        #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1'           #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1'          #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1'        #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1'           #list of nodes for storage - this is list of node numbers - can also be range
.
.
```

В этом примере есть только один хост, роль которого – «все-в-одном»: управление, сеть, вычисления, мониторинг, хранилище. Чтобы добавить вычислительный узел или узел хранения в уже развернутое облако, файл развертывания `yaml` необходимо изменить следующим образом:

```
.
.
fai_boot:                #boot information for FAI installation
  hosts: 2                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
    - 08:94:ef:60:0d:f3
```

```
- 08:94:EF:60:0C:1F
```

```
inventory:
```

```
  controllers: '1'           #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1'              #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1-2'          #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1'          #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1-2'           #list of nodes for storage - this is list of node numbers - can also be range
```

Количество хостов должно быть увеличено на один, должен быть добавлен MAC-адрес вновь добавленного хоста, количество узлов *computes*, *storage* и *monitoring* должно быть увеличено на один.

2.2.3 Подготовка среды развертывания

После того, как все необходимые настройки OpenStack выполнены, запустите скрипт BASH на сервере управления облаком:

```
cd /opt/cloud-cfg/<cloud-name>
itkf-deployment-prepare.sh <cloud-name>-big.yml
```

Он повторно сгенерирует файлы *globals.yml* и *cloud-inventory* и скопирует дополнительные файлы конфигурации в папку *etc* (*/opt/etc-kolla/<cloud-name>*). Эта папка также сопоставляется с контейнером *kolla-ansible-srv* как */etc/kolla/cloud-cfg/<cloud-name>/*.

2.2.4 Загрузка хостов и установка операционной системы

Убедитесь, что для вновь добавленного хоста установлены правильные параметры загрузки в соответствии с режимом загрузки, заданным в файле развертывания *yaml* (*uefi*, *pxe* или *https*).

Когда хост загружается, вы можете деактивировать действие загрузки (чтобы предотвратить нежелательную загрузку по сети):

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

Через некоторое время (около 10–20 минут) должна установиться операционная система. Чтобы проверить, прошла ли установка успешно, вы можете запустить:

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

2.2.5 Запуск нового облачного вычислительного узла

Запустите скрипт *itkf-deployment-add-compute.sh* с файлом развертывания *yaml*, который был подготовлен на предыдущих шагах. Мы добавляем один узел к одному существующему узлу, поэтому имя узла — *<cloud-name>-node2*:

```
itkf-deployment-add-compute.sh <cloud-name>-node2 <cloud-name>-big.yml
```



Предупреждение!

Если файл лицензии отсутствует или если нужно развернуть больше узлов, чем указано в файле лицензии, это действие не будет выполнено! Поэтому необходимо проверить количество узлов в файле лицензии.

2.2.6 Проверка нового облачного вычислительного узла

Чтобы выполнить проверку состояния и работоспособности нового вычислительного узла:

1. Проверьте, запущены ли все необходимые службы OpenStack на вновь добавленном узле `<cloud-name>-node2`. На вновь добавленном вычислительном узле или узле хранения должны быть только службы `nova-compute`, `cinder-volume`, `cinder-backup`, `neuron-openvswitch-agent`.
2. Удаленно войдите во вновь добавленный вычислительный узел или узел хранения:

```
ssh root@<cloud-name>-node2
```

3. Проверьте, запущены ли все необходимые контейнеры Docker:

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
bd71cba36b48	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-neutron-openvswitch-agent:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago neutron_openvswitch_agent
5d5f62aba541	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-openvswitch-vswitchd:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago openvswitch_vswitchd
76241a0ed64d	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-openvswitch-db-server:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago openvswitch_db
1f430b4b699e	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-nova-compute:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago nova_compute
d62c6030a2fb	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-nova-libvirt:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago nova_libvirt
5e9ea475cd67	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-nova-ssh:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago nova_ssh
6c6e9f93766b	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-cinder-backup:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago cinder_backup
ed1b0d49ad59	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-cinder-volume:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago cinder_volume
aee9fa10e06c	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-ceph-osd:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago ceph_osd_4
aee9f93766br	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-ceph-osd:1.0.2	go	Up 2 months	"dumb-init --single-..."	2 months ago ceph_osd_5

```

961788dca5da      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructur
e/ubuntu-source-prometheus-cadvisor:1.0.2      "dumb-init --single-..." 2 months a
go          Up 2 months      prometheus_cadvisor

a2773c5ccb0a      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructur
e/ubuntu-source-prometheus-node-exporter:1.0.2  "dumb-init --single-..." 2 months a
go          Up 2 months      prometheus_node_exporter

9cccab8b470       k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructur
e/ubuntu-source-chrony:1.0.2                   "dumb-init --single-..." 2 months a
go          Up 2 months      chrony

b7a42e6ac005      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructur
e/ubuntu-source-cron:1.0.2                     "dumb-init --single-..." 2 months a
go          Up 2 months      cron

d1b01c4e0bbb      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructur
e/ubuntu-source-kolla-toolbox:1.0.2            "dumb-init --single-..." 2 months a
go          Up 2 months      kolla_toolbox

2bb97d248a9a      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructur
e/ubuntu-source-fluentd:1.0.2                  "dumb-init --single-..." 2 months a
go          Up 2 months      fluentd

```

2.2.7 Повышение количества групп размещения серв

По мере увеличения количества сущностей OSD выбор правильного значения для параметра `pg_num` становится более важным, поскольку оно оказывает значительное влияние на поведение кластера, а также на устойчивость данных при различных сбоях и вероятность того, что такое событие приведет к потере данных.

После добавления в облако дополнительных узлов может случиться так, что групп размещения будет слишком мало, поэтому:

1. Проверьте текущее количество групп размещения (PG):

```

(on <cloud-name>-node1) ceph osd pool ls detail

pool 5 'images' replicated size 3 min_size 2 crush_rule 1 object_hash rjenkins pg_num 32 p
gp_num 32 last_change 1493 flags hashspool stripe_width 0 application rbd

removed_snaps [1~3,a~2]

pool 6 'volumes' replicated size 1 min_size 1 crush_rule 1 object_hash rjenkins pg_num 32
pgp_num 32 last_change 1500 flags hashspool stripe_width 0 application rbd

removed_snaps [1~3]

pool 7 'backups' replicated size 3 min_size 2 crush_rule 1 object_hash rjenkins pg_num 32
pgp_num 32 last_change 1501 flags hashspool stripe_width 0 application rbd

pool 8 'vms' replicated size 3 min_size 2 crush_rule 1 object_hash rjenkins pg_num 32 pgp_
num 32 last_change 1502 flags hashspool stripe_width 0 application rbd

removed_snaps [1~3]

```

2. Рассчитайте количество групп для вашего облака, используя калькулятор групп серв, например:

Ceph Use Case Selector:

Pool Name	Size	OSD #	%Data	Target PGs per OSD	Suggested PG Count
cinder-backup	3	18	10.00	100	64
cinder-volumes	3	18	65.00	100	512
ephemeral-vms	3	18	15.00	100	128
glance-images	3	18	10.00	100	64
			Total Data Percentage: 100.00%	PG Total Count: 768	

Notes

Рис. 2-1: Расчет количества групп размещения

3. Увеличьте количество групп размещения (`pg_num`). Как только вы увеличите количество «групп размещения», вы также должны увеличить количество «групп размещения для размещения» (`pgp_num`), прежде чем ваш кластер будет ребалансирован. Значение `pgp_num` должно быть равно значению `pg_num`.

Увеличивайте количество «групп размещения» (и количество «групп размещения для размещения») постепенно, с шагом коэффициента 2.

После каждого изменения количества групп PG/PGP дождитесь завершения ребалансировки кластера серв (используйте команду `ceph -s`):

```
(on <cloud-name>-node1) ceph osd pool set backups pg_num 64
(on <cloud-name>-node1) ceph osd pool set backups pgp_num 64
```

```
(on <cloud-name>-node1) ceph osd pool set images pg_num 64
(on <cloud-name>-node1) ceph osd pool set images pgp_num 64
```

```
(on <cloud-name>-node1) ceph osd pool set vms pg_num 64
(on <cloud-name>-node1) ceph osd pool set vms pgp_num 64
```

```
(on <cloud-name>-node1) ceph osd pool set vms pg_num 128
(on <cloud-name>-node1) ceph osd pool set vms pgp_num 128
```

```
(on <cloud-name>-node1) ceph osd pool set volumes pg_num 64
```

```
(on <cloud-name>-node1) ceph osd pool set volumes pgp_num 64
```

```
(on <cloud-name>-node1) ceph osd pool set volumes pg_num 128
```

```
(on <cloud-name>-node1) ceph osd pool set volumes pgp_num 128
```

```
(on <cloud-name>-node1) ceph osd pool set volumes pg_num 256
```

```
(on <cloud-name>-node1) ceph osd pool set volumes pgp_num 256
```

```
(on <cloud-name>-node1) ceph osd pool set volumes pg_num 512
```

```
(on <cloud-name>-node1) ceph osd pool set volumes pgp_num 512
```



Предупреждение!

После того, как вы увеличили количество групп размещения, вы не можете уменьшить это количество.

4. Проверьте количество групп размещения и статус ceph:

```
(on <cloud-name>-node1) ceph osd pool ls detail
```

```
pool 5 'images' replicated size 3 min_size 2 crush_rule 1 object_hash rjenkins pg_num 64 pgp_num 64 last_change 1493 flags hashpspool stripe_width 0 application rbd
```

```
removed_snaps [1~3,a~2]
```

```
pool 6 'volumes' replicated size 1 min_size 1 crush_rule 1 object_hash rjenkins pg_num 512 pgp_num 512 last_change 1500 flags hashpspool stripe_width 0 application rbd
```

```
removed_snaps [1~3]
```

```
pool 7 'backups' replicated size 3 min_size 2 crush_rule 1 object_hash rjenkins pg_num 64 pgp_num 64 last_change 1501 flags hashpspool stripe_width 0 application rbd
```

```
pool 8 'vms' replicated size 3 min_size 2 crush_rule 1 object_hash rjenkins pg_num 128 pgp_num 128 last_change 1502 flags hashpspool stripe_width 0 application rbd
```

```
removed_snaps [1~3]
```

```
(on <cloud-name>-node1) ceph -s
```

```
cluster:
```

```
id:      8659b80d-f592-4df3-abd7-6753af6a2777
```

```
health: HEALTH_OK
```

```
services:
```

```
mon: 3 daemons, quorum 192.168.161.21,192.168.161.22,192.168.161.23
```

```
mgr: cloudims11-node1.iskrauraltel.ru(active), standbys: cloudims11-node2.iskrauraltel.ru, cloudims11-node3.iskrauraltel.ru
```

```
osd: 20 osds: 19 up, 19 in
rgw: 1 daemon active

data:
  pools: 8 pools, 800 pgs
  objects: 434.90k objects, 1.60TiB
  usage: 1.66TiB used, 22.2TiB / 23.8TiB avail
  pgs: 100 active+clean

io:
  client: 258KiB/s rd, 5.89MiB/s wr, 300op/s rd, 348op/s wr
```

2.3 Удаление облачного вычислительного узла

Предварительные условия:

- ♦ рабочий сервер репозитория,
- ♦ сетевые имена сервера репозитория, вставленные на сервер DNS
- ♦ рабочий сервер управления облаком,
- ♦ основной файл конфигурации,
- ♦ дополнительные файлы конфигурации.

Вы можете удалить существующий вычислительный узел или узел хранения из существующего развернутого облака, например, чтобы заменить его новым узлом или понизить емкость облака.



Предупреждение!

Убедитесь, что у вас достаточно свободного места в сущностях Ceph OSD на других узлах ICP.

Экземпляры Nova необходимо перенести на другие вычислительные узлы. В противном случае процедура завершится и предупредит вас о необходимости переноса экземпляров Nova.

Количество реплик для всех пулов Ceph должно быть не менее двух (2). В противном случае процедура завершится и предупредит вас о необходимости увеличения количества реплик пулов Ceph.

После того, как вы выполнили начальное развертывание, вычислительные узлы и/или узлы хранения также можно удалить.

Сначала службы хранения Ceph должны быть отключены и удалены с удаляемого узла.

Затем вычислительные службы OpenStack должны быть отключены на удаляемом узле. Сетевые агенты и службы хранения OpenStack должны быть отключены. Контейнеры Docker должны быть остановлены и удалены. Образы Docker должны быть удалены. Все службы OpenStack должны быть удалены с удаляемого узла.

Наконец, для удаления узла, существующее развернутое облако должно быть переконфигурировано без узла, после этого он может быть дополнительно отключен и физически удален.

2.3.1 Поиск файлов конфигурации в папке cloud-cfg

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке `/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.

`<additional-configuration-files>` — все остальные файлы конфигурации.

В этом примере есть только один хост, роль которого – «все-в-одном»: управление, сеть, база данных, вычисления, мониторинг, хранилище. Другой узел, который необходимо удалить — вычислительный узел или узел хранения.

2.3.2 Запуск удаления облачного вычислительного узла

Чтобы проверить, активны ли узлы ICP, выполните:

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

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

Запустите сценарий `itkf-deployment-remove-compute.sh` с файлом развертывания `yaml`, который был проверен на предыдущих шагах. Мы удаляем один узел из существующего развернутого облака, поэтому имя узла — `<cloud-name>-node2`:

```
itkf-deployment-remove-compute.sh <cloud-name>-node2 <cloud-name>-big.yml
```

2.3.3 Изменение файлов конфигурации в папке cloud-cfg

Чтобы завершить удаление вычислительного узла, нужно сначала проверить следующие строки в файле `/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml`:

```
.
.
fai_boot:                #boot information for FAI installation
  hosts: 2                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
    - 08:94:ef:60:0d:f3
    - 08:94:EF:60:0C:1F
.
.
.
inventory:
  controllers: '1'        #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1'           #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1-2'        #list of nodes for compute nodes - this is list of node numbers - can also be range
```

```

monitoring: '1'           #list of nodes for monitoring nodes - this is list of node numbers - can also be range
storage: '1-2'          #list of nodes for storage - this is list of node numbers - can also be range

```

В этом примере есть только один хост, роль которого – «все-в-одном»: управление, сеть, база данных, вычисления, мониторинг, хранилище. Другой узел, который необходимо удалить — вычислительный узел или узел хранения. Чтобы удалить узел из уже развернутого облака, файл развертывания `yam1` необходимо изменить следующим образом:

#FAI specific

```

fai_boot:                #boot information for FAI installation
  hosts: 1                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                  #list of macs for DHCP offers
    - 08:94:ef:60:0d:f3

```

inventory:

```

controllers: '1'        #list of nodes for controllers - this is list of node numbers - can also be range
network: '1'           #list of nodes for network nodes - this is list of node numbers - can also be range
computes: '1'          #list of nodes for compute nodes - this is list of node numbers - can also be range
monitoring: '1'        #list of nodes for monitoring nodes - this is list of node numbers - can also be range
storage: '1'           #list of nodes for storage - this is list of node numbers - can also be range

```

Количество хостов должно быть уменьшено на один, должен быть удален MAC-адрес хоста, количество вычислительных узлов и узлов хранения должно быть уменьшено на один.

2.3.4 Подготовка среды развертывания

После того, как все необходимые настройки OpenStack выполнены, запустите скрипт BASH на сервере управления облаком:

```

cd /opt/cloud-cfg/<cloud-name>
itkf-deployment-prepare.sh <cloud-name>-big.yml

```

Он повторно сгенерирует файлы `globals.yml` и `cloud-inventory` и скопирует 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>/`.

2.3.5 Обновление развернутого облака

Чтобы проверить, активны ли узлы ICP, выполните:

```

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

```

Запустите скрипт `itkf-deployment-reconfigure.sh` с файлом развертывания `yaml`, который был подготовлен на предыдущих шагах.

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

2.3.6 Проверка удаления облачного вычислительного узла

Чтобы выполнить проверку статуса удаления вычислительного узла

1. Проверьте, удалены ли все необходимые службы OpenStack из `<cloud-name>-node2`.
2. Удаленно войдите в узел:

```
ssh root@<cloud-name>-node2
```

3. Проверьте, существуют ли контейнеры Docker:

```
docker ps
```

4. Если нужно, отключите удаленный узел `<cloud-name>-node2`.

2.4 Принудительное удаление облачного вычислительного узла

Предварительные условия:

- ♦ рабочий сервер репозитория,
- ♦ сетевые имена сервера репозитория, вставленные на сервер DNS,
- ♦ рабочий сервер управления облаком,
- ♦ основной файл конфигурации,
- ♦ дополнительные файлы конфигурации.

Вы можете удалить существующий аппаратный вычислительный узел или узел хранения, вышедший из строя (и отключенный), из существующего развернутого облака, например, чтобы заменить его новым узлом.



Предупреждение!

Убедитесь, что ваш кластер `Swift` стабилизировался после ребалансировки из-за аппаратного сбоя вычислительного узла или узла хранения, независимо от статуса `HEALTH_WARN` в отношении состояния резервирования.

Экземпляры `Nova` должны быть перемещены на другой вычислительный узел. Если это невозможно, экземпляры `Nova` должны быть удалены с отключенного узла. На узле не должно быть экземпляров, которые нужно удалить. В противном случае процедура завершится и предупредит вас о необходимости перемещения или удаления экземпляров `Nova`.

Количество реплик для всех пулов `Swift` должно быть не менее двух (2). В противном случае процедура завершится и предупредит вас о необходимости увеличения количества реплик пулов `Swift`.

После того, как вы выполнили начальное развертывание, вычислительные узлы и/или узлы хранения можно принудительно удалить.

Сначала службы хранения `Swift` должны быть отключены и удалены с удаляемого узла.

Затем вычислительные службы OpenStack должны быть отключены на удаляемом узле. Сетевые агенты и службы хранения OpenStack должны быть отключены. Все службы OpenStack должны быть удалены с удаляемого узла.

Наконец, для удаления узла, существующее развернутое облако должно быть переконфигурировано без узла.

2.4.1 Поиск файлов конфигурации в папке cloud-cfg

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке `/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.

`<additional-configuration-files>` — все остальные файлы конфигурации.

В этом примере есть только один хост, роль которого – «все-в-одном»: управление, сеть, база данных, вычисления, мониторинг, хранилище. Другой узел, который необходимо принудительно удалить — вычислительный узел или узел хранения.

2.4.2 Изменение файлов конфигурации в папке cloud-cfg

Чтобы завершить принудительное удаление вычислительного узла, сначала нужно проверить следующие строки в `/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml`:

```
.  
. .  
fai_boot: #boot information for FAI installation  
  hosts: 2 #number of hosts  
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional  
  macs: #list of macs for DHCP offers  
    - 08:94:ef:60:0d:f3  
    - 08:94:EF:60:0C:1F  
.  
.  
inventory:  
  controllers: '1' #list of nodes for controllers - this is list of node numbers - can also be range  
  network: '1' #list of nodes for network nodes - this is list of node numbers - can also be range  
  computes: '1-2' #list of nodes for compute nodes - this is list of node numbers - can also be range  
  monitoring: '1' #list of nodes for monitoring nodes - this is list of node numbers - can also be range  
  storage: '1-2' #list of nodes for storage - this is list of node numbers - can also be range  
.  
.  
.
```

В этом примере есть только один хост, роль которого – «все-в-одном»: управление, сеть, база данных, вычисления, мониторинг, хранилище. Другой узел, который необходимо принудительно удалить — вычислительный узел или узел хранения.

Чтобы принудительно удалить узел из уже развернутого облака, файл развертывания `yaml` должен быть изменен следующим образом:

```
.
.
.
#FAI specific
fai_boot:                #boot information for FAI installation
  hosts: 1                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
    - 08:94:ef:60:0d:f3
.
.
.
inventory:
  controllers: '1'        #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1'           #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1'          #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1'        #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1'           #list of nodes for storage - this is list of node numbers - can also be range
.
.
.
```

Количество хостов должно быть уменьшено на один, MAC-адрес удаляемого хоста должен быть удален, количество узлов вычислений и хранения должно быть уменьшено на один.

2.4.3 Подготовка среды развертывания

После того, как все необходимые настройки OpenStack выполнены, запустите скрипт BASH на сервере управления облаком:

```
cd /opt/cloud-cfg/<cloud-name>
itkf-deployment-prepare.sh <cloud-name>-big.yml
```

Он повторно сгенерирует файлы `globals.yml` и `cloud-inventory` и скопирует 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>/`.

2.4.4 Принудительный запуск удаления облачного вычислительного узла

Чтобы проверить, активны ли узлы ICP, выполните:

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

Запустите скрипт `itkf-deployment-remove-compute-force.sh` с файлом развертывания `yaml`, который был подготовлен на предыдущих шагах. Мы принудительно удаляем один узел из существующего развернутого облака, поэтому имя узла — `<cloud-name>-node`:

```
itkf-deployment-remove-compute-force.sh <cloud-name>-node2 <cloud-name>-big.yml
```

2.4.5 Обновление развернутого облака

Чтобы проверить, активны ли узлы ICP, выполните:

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

Запустите скрипт `itkf-deployment-reconfigure.sh` с файлом развертывания `yaml`, который был подготовлен на предыдущих шагах.

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

2.5 Добавление облачного обслуживающего узла

Предварительные условия:

- ♦ рабочий сервер репозитория,
- ♦ сетевые имена сервера репозитория, вставленные на сервер DNS,
- ♦ рабочий сервер управления облаком,
- ♦ основной файл конфигурации,
- ♦ дополнительные файлы конфигурации,
- ♦ файл лицензии с достаточным количеством лицензированных узлов,
- ♦ узлы ICP, подключенные к сети.

Вы можете добавить новый узел управления, сети, базы данных, вычислений, хранения или мониторинга в существующее развернутое облако, например, чтобы заменить существующий узел (оборудование которого отказало).

Добавить узел можно после того, как вы выполнили развертывание облака. Вновь добавленный узел будет иметь только необходимые сервисы Horizon, Keystone, Glance, Heat, Nova, Neutron, Cinder и Ceph. Также будут добавлены службы мониторинга.

Во-первых, параметры конфигурационного пространства модуля FAI должны быть добавлены или изменены в соответствии со спецификациями оборудования нового узла. ОС нового узла должна быть установлена с помощью модуля FAI.

Во-вторых, необходимо добавить или изменить файлы `globals.yml` и `cloud-inventory`. Службы OpenStack нового узла должны быть установлены с помощью скриптов `itkf*-*`.

Наконец, проверьте, достаточно ли в файле лицензии узлов для планируемого обновления. Самый простой способ проверить это – с помощью команды `license_info`:

```
# license_info -y /opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml
```

```
...
```

```
Number of licensed nodes: 5
```

Если количество узлов недостаточно, необходимо получить новый файл лицензии от производителя продукта.

2.5.1 Поиск файлов конфигурации в папке `cloud-cfg`

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке `/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.

`<additional-configuration-files>` — все остальные файлы конфигурации.

2.5.2 Изменение файлов конфигурации в папке `cloud-cfg`

Подготовка к добавлению сервисного узла, сначала нужно проверить следующие строки в `/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml`:

```
.
.
#FAI specific
fai_boot:                #boot information for FAI installation
  hosts: 2                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
    - 08:94:ef:60:0d:f3
    - 48:94:ef:60:2c:1f
.
.
.
inventory:
  controllers: '1-2'      #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1-2'         #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1-2'        #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1-2'      #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1-2'         #list of nodes for storage - this is list of node numbers - can also be range
.
.
```

В этом примере есть только два сервисных узла, роли которых — управление, сеть, вычисления, база данных, мониторинг и хранилище. Третий уже удален из-за, например, аппаратного сбоя. Чтобы добавить узел управления, сети, базы данных, вычислений, хранения или мониторинга в уже развернутое облако, файл развертывания `yam1` необходимо изменить следующим образом:

```
.
.
fai_boot:                #boot information for FAI installation
  hosts: 3                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
    - 08:94:EF:60:0D:F3
    - 48:94:EF:60:0C:1F
    - 58:94:EF:66:8D:AF
.
.
.
```

inventory:

```

controllers: '1-3'           #list of nodes for controllers - this is list of node numbers - can also be range
network: '1-3'              #list of nodes for network nodes - this is list of node numbers - can also be range
computes: '1-3'             #list of nodes for compute nodes - this is list of node numbers - can also be range
monitoring: '1-3'          #list of nodes for monitoring nodes - this is list of node numbers - can also be range
storage: '1-3'              #list of nodes for storage - this is list of node numbers - can also be range
.
.

```

Количество хостов должно быть увеличено на один, должен быть добавлен MAC-адрес вновь добавленного хоста, количество управляющих и вычислительных узлов, а также сетевых узлов, узлов хранения и мониторинга должно быть увеличено на один.

2.5.3 Подготовка среды развертывания

После того, как все необходимые настройки OpenStack выполнены, запустите скрипт BASH на сервере управления облаком:

```

cd /opt/cloud-cfg/<cloud-name>
itkf-deployment-prepare.sh <cloud-name>-big.yml

```

Он повторно сгенерирует файлы *globals.yml* и *cloud-inventory* и скопирует 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>/*.

2.5.4 Загрузка хостов и установка операционной системы

Убедитесь, что для вновь добавленного хоста установлены правильные параметры загрузки в соответствии с режимом загрузки, заданным в файле развертывания *yaml* (*uefi*, *pxe* или *https*).

Когда хост загружается, вы можете деактивировать действие загрузки (чтобы предотвратить нежелательную загрузку по сети):

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

Через некоторое время (около 10–20 минут) должна установиться операционная система. Чтобы проверить, прошла ли установка успешно, выполните:

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

2.5.5 Запуск добавления облачного сервисного узла

Запустите скрипт *itkf-deployment-add-node.sh* с файлом развертывания *yaml*, который был подготовлен на предыдущих шагах. Мы добавляем один узел к существующим двум узлам, поэтому имя узла *<cloud-name>-node3*:

```
itkf-deployment-add-node.sh <cloud-name>-node3 <cloud-name>-big.yml
```



Предупреждение!

Если файл лицензии отсутствует или если нужно развернуть больше узлов, чем указано в файле лицензии, это действие не будет выполнено! Проверьте в файле количество лицензированных узлов.

2.5.6 Проверка добавления облачного сервисного узла

Чтобы выполнить проверку состояния и работоспособности сервисного узла:

1. Проверьте, запущены ли все необходимые службы OpenStack на вновь добавленном узле `<cloud-name>-node3`. На вновь добавленном узле управления, сети, базы данных, вычислений, хранения или мониторинга должны быть службы `nova-compute`, `nova-conductor`, `nova-consoleauth`, `nova-scheduler`, `cinder-volume`, `cinder-backup`, `cinder-scheduler`, `neutron-metadata-agent`, `neutron-l3-agent`, `neutron-dhcp-agent`, `neutron-openvswitch-agent`.
2. Войдите удаленно во вновь добавленный узел управления, сети, базы данных, вычислений, хранения или мониторинга:

```
ssh root@<cloud-name>-node3
```

3. Проверьте, запущены ли все необходимые контейнеры Docker:

```
docker ps
```

CONTAINER ID	IMAGE	CREATED	STATUS	PORTS	NAMES
26debfd09efc	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-horizon:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		horizon
5d20c3334a4f	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-heat-engine:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		heat_engine
b2abd2908366	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-heat-api-cfn:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		heat_api_cfn
591f2ee835ef	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-heat-api:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		heat_api
a383f677982f	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-neutron-metadata-agent:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		neutron_metadata_agent
b49cb83741ec	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-neutron-l3-agent:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		neutron_l3_agent
875c9750bc9f	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-neutron-dhcp-agent:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		neutron_dhcp_agent
f0c1839758a3	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-neutron-openvswitch-agent:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		neutron_openvswitch_agent
7bae75cd41c3	k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubuntu-source-neutron-server:1.0.2	"dumb-init --single-..." 5 weeks ago	Up 5 weeks		neutron_server

```

57e12240a92f      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-openvswitch-vswitchd:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          openv
switch_vswitchd

ee8ef635a198      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-openvswitch-db-server:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          openv
switch_db

95707e695aff      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-compute:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          nova_
compute

ae1b1d591349      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-novncproxy:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          nova_
novncproxy

a790f7a50577      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-consoleauth:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          nova_
consoleauth

ddf7f745cf69      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-conductor:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          nova_
conductor

4aa718339e0b      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-scheduler:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          nova_
scheduler

9eaa19ab648b      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-api:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          nova_
api

f647a0f28d95      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-placement-api:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          place
ment_api

b5f480a2a343      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-libvirt:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          nova_
libvirt

672869ddd99c      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-nova-ssh:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          nova_
ssh

ebe6ff10e3fb      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-cinder-backup:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          cinde
r_backup

8079666c6cf2      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-cinder-volume:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          cinde
r_volume

f97d9b07b860      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-cinder-scheduler:1.0.2
"dumb-init --single-..."  5 weeks ago          Up 5 weeks          cinde
r_scheduler

```

```

f4d0c610f432      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-cinder-api:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      cinde
r_api

f9dc12947bcb      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-glance-api:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      glanc
e_api

ded1fd4fdf83      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-ceph-rgw:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      ceph_
rgw

602508c3d07c      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-ceph-osd:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      ceph_
osd_5

06f12be07e03      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-ceph-osd:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      ceph_
osd_4

bd9c9b5dabe8      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-ceph-mgr:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      ceph_
mgr

c63e36e4cae0      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-ceph-mon:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      ceph_
mon

4c0d44f269ee      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-keystone-fernet:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      keyst
one_fernet

d9ece11a753d      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-keystone-ssh:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      keyst
one_ssh

cf68dcd5169       k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-keystone:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      keyst
one

81d9de64e245      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-rabbitmq:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 5 weeks      rabbi
tmq

8c7e0599665a      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-prometheus-alertmanager:1.0
2
"dumb-init --single-..." 5 weeks ago      Up 7 days      prometheu
s_alertmanager

8b2efe957143      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-prometheus-cadvisor:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 7 days      prome
theus_cadvisor

83c9159c744a      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-prometheus-memcached-export
r:1.0.2
"dumb-init --single-..." 5 weeks ago      Up 7 days      prometheu
s_memcached_exporter

```

```

f740ab835ea4      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-prometheus-haproxy-exporter      :
1.0.2      "dumb-init --single-..."  5 weeks ago      Up 7 days
prometheus_haproxy_exporter

adca5c0d597a      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-prometheus-mysqld-exporter:      1
.0.2      "dumb-init --single-..."  5 weeks ago      Up 7 days
prometheus_mysqld_exporter

67090724411f      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-prometheus-node-exporter:1.      0
.2      "dumb-init --single-..."  5 weeks ago      Up 7 days
prometheus_node_exporter

8d3732874cfc      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-prometheus-server:1.0.2
"dumb-init --single-..."  5 weeks ago      Up 7 days      prome
theus_server

eae964cdfa3c      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-mariadb:1.0.2
"dumb-init kolla_sta..."  5 weeks ago      Up 5 weeks      maria
db

753ca591319f      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-memcached:1.0.2
"dumb-init --single-..."  5 weeks ago      Up 5 weeks      memca
ched

b195e83a93f5      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-keepalived:1.0.2
"dumb-init --single-..."  5 weeks ago      Up 5 weeks      keepa
lived

1a8ae7f78463      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-haproxy:1.0.2
"dumb-init --single-..."  5 weeks ago      Up 5 weeks      hapro
xy

ac98bd9a097e      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-chrony:1.0.2
"dumb-init --single-..."  5 weeks ago      Up 5 weeks      chron
y

db86117901f0      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-cron:1.0.2
"dumb-init --single-..."  5 weeks ago      Up 5 weeks      cron

f85703721949      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-kolla-toolbox:1.0.2
"dumb-init --single-..."  5 weeks ago      Up 5 weeks      kolla
_toolbox

06c5019df5b2      k-vm-repo-server.docker.iskrauraltel.ru:4567/deploy/infrastructure/ubu
ntu-source-fluentd:1.0.2
"dumb-init --single-..."  5 weeks ago      Up 5 weeks      fluen
td

```

2.6 Удаление облачного сервисного узла

Предварительные условия:

- ◆ рабочий сервер репозитория,
- ◆ сетевые имена сервера репозитория, вставленные на сервер DNS,
- ◆ рабочий сервер управления облаком,
- ◆ основной файл конфигурации,

- ◆ дополнительные файлы конфигурации.

Вы можете удалить существующий узел управления, сети, базы данных, вычислений, хранения или мониторинга из существующего развернутого облака, например, чтобы заменить его новым узлом.



Предупреждение!

В системе с тремя (3) узлами управления, сети, базы данных, вычислений, хранения или мониторинга следует удалить только один (1).

Убедитесь, что у вас достаточно свободного места в сущностях OSD Серв на других узлах ICP.

Если после удаления будет меньше, чем три (3) узла хранения, вы должны уменьшить количество реплик для всех пулов Серв до количества оставшихся узлов хранения. Если нет, процедура завершится, а предупреждение Серв HEALTH_WARN сохранится.

Количество реплик для всех пулов Серв должно быть не менее двух (2). В противном случае процедура завершится и предупредит вас о необходимости увеличения количества реплик пулов Серв.

Экземпляры Nova необходимо перенести на другие вычислительные узлы. В противном случае процедура завершится и предупредит о необходимости переноса экземпляров Nova.

После того, как вы выполнили начальное развертывание, можно удалить узел.

Сначала службы хранения Серв должны быть отключены и удалены с удаляемого узла.

Во-вторых, службы управления и вычислений OpenStack должны быть отключены на удаляемом узле. Сетевые агенты и службы хранения OpenStack должны быть отключены. Контейнеры Docker должны быть остановлены и удалены. Образы Docker должны быть удалены. Все службы OpenStack должны быть удалены с удаляемого узла.

Наконец, для удаления узла, существующее развернутое облако должно быть переконфигурировано без узла, после этого он может быть дополнительно отключен и физически удален.

2.6.1 Поиск файлов конфигурации в папке cloud-cfg

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке /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 – основной файл конфигурации.

<additional-configuration-files> – все дополнительные файлы конфигурации.

В этом примере есть три хоста, роли которых: управление, сеть, база данных, вычисления, мониторинг, хранилище. Третий хост, который необходимо удалить, также является сервисным узлом (управления, сети, базы данных, вычислений, хранения или мониторинга).

2.6.2 Запуск удаления облачного сервисного узла

Чтобы проверить, активны ли узлы ICP, выполните:

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

Запустите сценарий `itkf-deployment-remove-controller-compute.sh` с файлом развертывания `yaml`, который был проверен на предыдущих шагах. Мы удаляем третий узел из существующего развернутого облака, поэтому имя узла – `<cloud-name>-node3`:

```
itkf-deployment-remove-controller-compute.sh <cloud-name>-node3 <cloud-name>-big.yml
```

2.6.3 Изменение файлов конфигурации в папке `cloud-cfg`

Чтобы завершить удаление сервисного узла, сначала нужно проверить следующие строки в `/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml`:

```
.
.
fai_boot:                #boot information for FAI installation
  hosts: 3                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                  #list of macs for DHCP offers
    - 08:94:EF:60:0D:F3
    - 48:94:EF:60:0C:1F
    - 58:94:EF:66:8D:AF
.
.
.
inventory:
  controllers: '1-3'      #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1-3'         #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1-3'        #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1-3'      #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1-3'         #list of nodes for storage - this is list of node numbers - can also be range
.
.
```

В этом примере есть три хоста сервисных узлов, роли которых: управление, сеть, вычисления, мониторинг и хранилище. Третий хост, который необходимо удалить, также является сервисным узлом (управления, сети, базы данных, вычислений, хранения или мониторинга). Чтобы удалить узел из уже развернутого облака, файл развертывания `yaml` необходимо изменить следующим образом:

```
.
.
#FAI specific
fai_boot:                #boot information for FAI installation
  hosts: 2                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
```

```

macs:                #list of macs for DHCP offers
- 08:94:ef:60:0d:f3
-48:94:EF:60:0C:1F

.
.
.
inventory:
  controllers: '1-2'  #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1-2'     #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1-2'    #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1-2'  #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1-2'     #list of nodes for storage - this is list of node numbers - can also be range

.
.

```

Количество хостов должно быть уменьшено на один, MAC-адрес удаляемого хоста должен быть удален, количество узлов управления, вычислений, сети, хранилища и мониторинга должно быть уменьшено на один.

2.6.4 Подготовка среды развертывания

После того, как все необходимые настройки OpenStack выполнены, запустите скрипт BASH на сервере управления облаком:

```

cd /opt/cloud-cfg/<cloud-name>
itkf-deployment-prepare.sh <cloud-name>-big.yml

```

Он повторно сгенерирует файлы *globals.yml* и *cloud-inventory* и скопирует 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>/*.

2.6.5 Обновление развернутого облака

Чтобы проверить, активны ли узлы ICP, выполните:

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

Запустите скрипт *itkf-deployment-reconfigure.sh* с файлом развертывания *yaml*, который был подготовлен на предыдущих шагах.

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

2.6.6 Проверка удаления облачного сервисного узла

Чтобы выполнить проверку состояния и работоспособности сервисного узла:

1. Проверьте, удалены ли все необходимые службы OpenStack из узла *<cloud-name>-node3*.
2. Удаленно войдите в узел управления, сети, базы данных, вычислений, хранения или мониторинга:

```
ssh root@<cloud-name>-node3
```

3. Проверьте, существуют ли контейнеры Docker:

```
docker ps
```
4. При необходимости вы можете отключить сервисный узел `<cloud-name>-nodez`.

2.7 Принудительное удаление облачного сервисного узла

Предварительные условия:

- ♦ рабочий сервер репозитория,
- ♦ сетевые имена сервера репозитория, вставленные на сервер DNS,
- ♦ рабочий сервер управления облаком,
- ♦ основной файл конфигурации,
- ♦ дополнительные файлы конфигурации.

Вы можете удалить существующий узел управления, сети, базы данных, вычислений, хранения или мониторинга из существующего развернутого облака, например, чтобы заменить его новым узлом.



Предупреждение!

В системе с тремя (3) узлами управления, сети, базы данных, вычислений, хранения или мониторинга следует удалить только один (1).

Убедитесь, что ваш кластер Serph стабилизировался после перебалисировки из-за аппаратного отказа узла управления, сети, базы данных, вычислений, хранения или мониторинга, независимо от состояния HEALTH_WARN.

Если после удаления будет меньше, чем три (3) узла хранения, вы должны уменьшить количество реплик для всех пулов Serph до количества оставшихся узлов хранения.

Количество реплик для всех пулов Serph должно быть не менее двух (2). В противном случае процедура завершится и предупредит вас о необходимости увеличения количества реплик пула Serph.

Экземпляры Nova должны быть перемещены на другой вычислительный узел. Если это невозможно, экземпляры Nova должны быть удалены с отключенного узла. На узле не должно быть экземпляров, которые нужно удалить. В противном случае процедура завершится и предупредит вас о необходимости перемещения или удаления экземпляров Nova.

После того, как вы выполнили начальное развертывание, можно принудительно удалить отключенный узел.

Сначала службы хранения Serph должны быть отключены и удалены с удаляемого узла.

Во-вторых, службы управления и вычислений OpenStack должны быть отключены на удаляемом узле. Сетевые агенты и службы хранения OpenStack должны быть отключены. Все службы OpenStack должны быть удалены с удаляемого узла.

Наконец, для удаления узла, существующее развернутое облако должно быть переконфигурировано без узла.

2.7.1 Поиск файлов конфигурации в папке cloud-cfg

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке `/opt/cloud-cfg`. Каждое облако должно иметь следующую структуру каталогов:

#FAI specific

```
fai_boot:                #boot information for FAI installation
  hosts: 2                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
    - 08:94:ef:60:0d:f3
    - 48:94:EF:60:0C:1F
```

inventory:

```
  controllers: '1-2'      #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1-2'         #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1-2'        #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1-2'      #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1-2'         #list of nodes for storage - this is list of node numbers - can also be range
```

Количество хостов должно быть уменьшено на один, MAC-адрес удаляемого хоста должен быть удален, количество узлов управления, вычислений, сети, хранилища и мониторинга должно быть уменьшено на один.

2.7.3 Подготовка среды развертывания

После того, как все необходимые настройки OpenStack выполнены, запустите скрипт BASH на сервере управления облаком:

```
cd /opt/cloud-cfg/<cloud-name>
itkf-deployment-prepare.sh <cloud-name>-big.yml
```

Он повторно сгенерирует файлы *globals.yml* и *cloud-inventory* и скопирует 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>/*.

2.7.4 Запуск принудительного удаления облачного сервисного узла

Чтобы проверить, активны ли узлы ICP, выполните:

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

Запустите сценарий *itkf-deployment-remove-controller-compute-force.sh* с файлом развертывания *yaml*, который был подготовлен на предыдущих шагах. Мы принудительно удаляем третий узел из существующего развернутого узла, поэтому имя узла — *<cloud-name>-node3*:

```
itkf-deployment-remove-controller-compute-force.sh <cloud-name>-node3 <cloud-name>-big.yml
```

2.7.5 Обновление развернутого облака

Чтобы проверить, активны ли узлы ICP, выполните:

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

Запустите скрипт *itkf-deployment-reconfigure.sh* с файлом развертывания *yaml*, который был подготовлен на предыдущих шагах.

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

2.8 Добавление облачных сервисных узлов от универсального (автономного) облака к многорежимному

Предварительные условия:

- ♦ рабочий сервер репозитория,
- ♦ сетевые имена сервера репозитория, вставленные на сервер DNS,
- ♦ рабочий сервер управления облаком,
- ♦ основной файл конфигурации,
- ♦ дополнительные файлы конфигурации,
- ♦ файл лицензии с достаточным количеством лицензированных узлов,
- ♦ узлы ICP, подключенные к сети.

Вы можете добавить два (2) новых узла управления, сети, базы данных, вычислений, хранения или мониторинга к существующему универсальному или автономному развернутому облаку, чтобы обновить его до многорежимного облака (с высокой доступностью).



Предупреждение!

После добавления двух (2) новых узлов вы должны увеличить количество реплик для всех пулов *Swift* до количества узлов хранения (т.е. 3).

После того, как вы выполнили развертывание облака, можно добавить два (2) узла. Вновь добавленные узлы будут иметь только необходимые сервисы *Horizon*, *Keystone*, *Glance*, *Heat*, *Nova*, *Neutron*, *Cinder* и *Swift*. Также будут добавлены службы мониторинга.

Во-первых, параметры конфигурационного пространства модуля *FAI* должны быть добавлены или изменены в соответствии со спецификациями оборудования нового узла. ОС нового узла должна быть установлена с помощью модуля *FAI*.

Во-вторых, необходимо добавить или изменить файлы *globals.yml* и *cloud-inventory*. Службы *OpenStack* нового узла должны быть установлены с помощью сценариев *itkf*-**.

Наконец, проверьте, достаточно ли в файле лицензии узлов для планируемого обновления. Самый простой способ сделать это – с помощью команды *license_info*:

```
# license_info -y /opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml
```

```
...
```

```
Number of licensed nodes: 5
```

Если количество узлов недостаточно, необходимо получить новый файл лицензии от производителя продукта.

2.8.1 Поиск файлов конфигурации в папке cloud-cfg

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке `/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` – основной файл конфигурации.

`<additional-configuration-files>` – все дополнительные файлы конфигурации.

2.8.2 Изменение файлов конфигурации в папке cloud-cfg

Для подготовки к добавлению двух (2) новых сервисных узлов, сначала нужно проверить следующие строки в `/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml`:

```
.
.
#FAI specific
fai_boot:                #boot information for FAI installation
  hosts: 1                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
    - 08:94:ef:60:0d:f3
.
.
.
inventory:
  controllers: '1'        #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1'           #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1'          #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1'        #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1'           #list of nodes for storage - this is list of node numbers - can also be range
.
.
```

В этом примере есть только один хост, роль которого – «все-в-одном»: управление, сеть, вычисления, база данных, мониторинг, хранилище. Чтобы добавить два (2) новых узла управления, сети, базы данных, вычислений, хранения или мониторинга к уже развернутому облаку, файл развертывания `yaml` необходимо изменить следующим образом:

```
.
.
fai_boot:                #boot information for FAI installation
  hosts: 3                #number of hosts
  ip_start: 192.168.36.11/24 #starting IP if DHCP offers - optional
  macs:                   #list of macs for DHCP offers
```

```
- 08:94:EF:60:0D:F3
- 48:94:EF:60:0C:1F
- 58:94:EF:66:8D:AF
```

```
inventory:
```

```
  controllers: '1-3'      #list of nodes for controllers - this is list of node numbers - can also be range
  network: '1-3'        #list of nodes for network nodes - this is list of node numbers - can also be range
  computes: '1-3'       #list of nodes for compute nodes - this is list of node numbers - can also be range
  monitoring: '1-3'     #list of nodes for monitoring nodes - this is list of node numbers - can also be range
  storage: '1-3'        #list of nodes for storage - this is list of node numbers - can also be range
```

Количество хостов должно быть увеличено до трех (3), необходимо добавить MAC-адреса двух (2) вновь добавленных хостов, а количество узлов управления и вычислений, а также сети, хранения и мониторинга следует увеличить до трех (3).

2.8.3 Подготовка среды развертывания

После того, как все необходимые настройки OpenStack выполнены, запустите скрипт BASH на сервере управления облаком:

```
cd /opt/cloud-cfg/<cloud-name>
itkf-deployment-prepare.sh <cloud-name>-big.yml
```

Он повторно сгенерирует файлы *globals.yml* и *cloud-inventory* и скопирует 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>/*.

2.8.4 Загрузка хостов и установка операционной системы

Убедитесь, что два (2) вновь добавленных хоста имеют правильные параметры загрузки в соответствии с режимом загрузки, заданным в файле развертывания *yaml* (*uefi*, *pxe* или *https*).

Когда хосты загружаются, вы можете деактивировать действие загрузки (чтобы предотвратить нежелательную загрузку по сети):

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

Через некоторое время (около 10–20 минут) должна установиться операционная система. Чтобы проверить, прошла ли установка успешно, выполните:

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

2.8.5 Запуск добавления облачных сервисных узлов

Запустите скрипт *itkf-deployment-start.sh* с файлом развертывания *yaml*, который был подготовлен на предыдущих шагах. Мы добавляем два (2) узла к существующему развертыванию «все в одном», чтобы обновить его до мультирежимного:

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

**Предупреждение!**

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

2.8.6 Проверка добавления облачных сервисных узлов

Чтобы выполнить проверку состояния и работоспособности двух (2) сервисных узлов:

1. Проверьте статус Ceph:

```
(on <cloud-name>-node1) ceph -s
```

```
cluster:
```

```
id: 69705746-434e-481d-9aa6-4bee57781b38
```

```
health: HEALTH_OK
```

```
services:
```

```
mon: 3 daemons, quorum 172.30.2.11,172.30.2.12,172.30.2.13
```

```
mgr: testedge-node1.csi.iskrauraltel.ru(active), standbys: testedge-node2.csi.iskrauraltel.ru
```

```
osd: 6 osds: 6 up, 6 in
```

```
rgw: 1 daemon active
```

```
data:
```

```
pools: 8 pools, 160 pgs
```

```
objects: 3.26k objects, 6.06GiB
```

```
usage: 13.4GiB used, 5.43TiB / 5.45TiB avail
```

```
pgs: 160 active+clean
```

```
io:
```

```
client: 760B/s rd, 0B/s wr, 0op/s rd, 0op/s wr
```

2. Увеличить количество реплик пула Ceph с 1 до 3.
3. Получите пароль кластера MariaDB Galera:

```
(on fai-kolla-server) grep -w database_password /opt/etc-kolla/<cloud-name>/passwords.yml
```

```
database_password: [пароль]
```

4. Проверьте состояние кластера Galera. Кластер Galera должен быть активен, все три (3) узла должны участвовать

```
(on <cloud-name>-node1) docker exec mariadb mysql -u root -p[пароль] -e "SHOW STATUS" | grep wsrep_cluster
```

```
wsrep_cluster_conf_id 10
```

```
wsrep_cluster_size 3
```

```
wsrep_cluster_state_uuid      74052ada-f4a0-11e9-9152-ff1b943b2e97
wsrep_cluster_status          Primary
```

5. Если `wsrep_cluster_size` не равен 3, восстановите кластер базы данных MariaDB Galera:

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v <cloud-name> mariadb
```

6. Проверьте статус RabbitMQ. Кластер RabbitMQ должен быть активен, все три (3) узла должны участвовать:

```
(on <cloud-name>-node1) docker exec rabbitmq rabbitmqctl cluster_status
```

```
Cluster status of node 'rabbit@testedge-node1' ...
```

```
[[{nodes,[[{disc,['rabbit@testedge-node1','rabbit@testedge-node2',
              'rabbit@testedge-node3']}]},
{running_nodes,['rabbit@testedge-node1','rabbit@testedge-node2',
                'rabbit@testedge-node3']},
{cluster_name,<<"rabbit@testedge-node1.csi.iskrauraltel.ru">>},
{partitions,[]},
{alarms,[{'rabbit@testedge-node3',[]},
          {'rabbit@testedge-node2',[]},
          {'rabbit@testedge-node1',[]}]}]
```

7. Если в списке `running_nodes` не 3 (т.е. не все) участника, проверьте `/var/log/kolla/rabbitmq/startup_log` на вновь добавленных сервисных узлах на наличие ошибок или предупреждений:

```
(on <cloud-name>-node[2-3]) cat /var/log/kolla/rabbitmq/startup_log
```

```
=INFO REPORT==== 14-Jan-2020::12:54:48 ===
```

```
Multiple different configurations with equal version numbers detected. Shutting down.
```

```
[[{node_id,<<19,17,123,40,31,248,71,36,91,69,60,11,135,21,166,103>>},
{node_ids,[{'rabbit@testedge-node2',<<19,17,123,40,31,248,71,36,91,69,60,
              11,135,21,166,103>>}]},
{gospel,{node,'rabbit@testedge-node1'}},
{nodes,[{'rabbit@testedge-node1',disc},{'rabbit@testedge-node2',disc}]},
{version,1}]
```

```
=INFO REPORT==== 14-Jan-2020::12:54:48 ===
```

```
Clusterer stopping node now.
```

8. Вручную увеличьте значение `version` на единицу в файле `/etc/kolla/rabbitmq/rabbitmq-clusterer.config` на всех сервисных узлах:

```
(on <cloud-name>-node[1-3]) cat /etc/kolla/rabbitmq/rabbitmq-clusterer.config
```

```
[
```

```

    {version, 1},
    {nodes, [
      {'rabbit@testedge-node1', disc}, {'rabbit@testedge-node2', disc}, {'rab
bit@testedge-node3', disc} ]},
    {gospel,
      {node, 'rabbit@testedge-node1'}}
  ]}.

```

В нашем случае мы увеличили номер `version` с 1 до 2.

9. Перезапустите кластер RabbitMQ

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f <cloud-name> rabbitmq
```

10. Проверьте на всех узлах, что все контейнеры Docker запустились должным образом:

```
(on <cloud-name>-node*) docker ps -a --filter "label=kolla_version" --filter status=exi
ted --filter status=restarting --format "{{.Names}}"
```

```
nova_scheduler
```

11. При необходимости принудительно перезапустите неработающие сервисные группы OpenStack. В нашем случае контейнер `nova_scheduler` завершил работу, поэтому все службы Nova необходимо перезапустить в заданном порядке:

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f <cloud-name> nova
```

Рекомендуется перезапустить сервисные группы OpenStack в следующем порядке:

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v <cloud-name> glance
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v <cloud-name> cinder
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v <cloud-name> nova
```

12. Проверьте, запущены ли все необходимые службы OpenStack на вновь добавленных узлах `<cloud-name>-node2` и `<cloud-name>-node3`. На вновь добавленных узлах управления, сети, базы данных, вычислений, хранения или мониторинга должны присутствовать `nova-compute`, `nova-conductor`, `nova-consoleauth`, `nova-scheduler`, `cinder-volume`, `cinder-backup`, `cinder-scheduler`, `neutron-metadata-agent`, `neutron-l3-agent`, `neutron-dhcp-agent`, `neutron-openvswitch-agent`.

13. Удаленно войдите во вновь добавленные узлы управления, сети, базы данных, вычислений, хранения или мониторинга:

```
ssh root@<cloud-name>-node2
```

```
ssh root@<cloud-name>-node3
```

14. Проверьте, запущены ли все необходимые контейнеры Docker.

2.9 Незначительное обновление облака

2.9.1 Описание

Незначительное обновление может быть выполнено, когда меняется вторая и/или третья цифра номера версии `it_kolla_version`.

Процедуры могут выполняться на активном объекте и без необходимости простоя. Обновление также можно выполнить на одном узле. Нет необходимости отключать экземпляры, так как они запускаются в `detu` вне контейнеров.

Модуль `Kolla-ansible` обновляет контейнер за контейнером, поэтому даже на одном узле службы `Openstack` не отключаются во время обновления. При этом модуль `kolla-ansible` перезагружает контейнеры во время обновления, поэтому возможны очень небольшие перебои в работе сервисов `Openstack`.

Конфигурация с несколькими узлами с контейнерами тоже перезапустится, но перебои в работе сервисов `Openstack` будут очень короткие. Контейнеры не перезапускаются в одно и то же время на всех узлах.

Модуль `Kolla-ansible` также выполнит обновление БД.

2.9.2 Процедура обновления

1. Подготовьте конфигурацию.

Задайте новую версию `it_kolla_version` в файле конфигурации облака (если вы обновляете облачные сервисы):

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

Установите новую версию `ceph_version` в файле конфигурации облака (если вы обновляете `ceph`):

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

Выполните:

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

2. Обновите службы `Openstack`:

Выполните:

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

3. Обновите `Ceph`:

Выполните:

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



Примечание.

Если вы запускаете `ceph` в кластере с одним узлом, выполните команду обновления с ключом `-s`:

```
itkf-deployment-upgrade-ceph.sh -s <cloud-name>-big.yml
```

При обновлении `ceph` нужно иметь в виду, что после перезапуска контейнеров `ceph_osd` статус `ceph` установится в `HEALTH_WARN` и останется в этом состоянии до тех пор, пока контейнеры `ceph_osd` не стабилизируются. Это не займет много времени.

После завершения обновления `ceph` перезапустите все службы:

```
itkf-maintenance-restart-service.sh -f <cloud-name> all
```

2.10 Переключение репозитория

Вы можете в любое время после завершения развертывания переключить сервер репозитория на другой URL-адрес. Это позволяет использовать разные серверы репозитория во время жизненного цикла облака.

Чтобы переключить сервер репозитория, выполните следующие шаги:

1. Отредактируйте файл развертывания `yaml`, установив новые параметры сервера репозитория для репозитория `apt` и реестра `docker` (см. таблицу ниже):

Табл. 2-1: Поля для переключения репозитория

Поле	Пример	Описание
<code>docker_registry:</code>	<code>'registry.iskrauraltel.cloud:5001'</code>	URL-адрес реестра <code>docker</code> (без префикса <code>http(s)</code>)
<code>docker_namespace:</code>	<code>'ai6212ax/it_kolla'</code>	пространство имен реестра
<code>docker_registry_username:</code>	<code>'user'</code>	имя пользователя для реестра
<code>docker_registry_password:</code>	<code>'password'</code>	пароль для реестра (можно оставить пустым)
<code>packages_repository:</code>	<code>"http://repository.iskrauraltel.cloud:80"</code>	репозиторий пакетов <code>apt</code>
<code>offline_nexus:</code>	<code>'yes'</code>	индикатор офлайн-репозитория или центрального <code>nexus</code> (<code>'yes'</code> для автономного и <code>'no'</code> для центрального)
<code>base_os:</code>	<code>'focal'</code>	выпуск операционной системы (<code>'bionic'</code> или <code>'focal'</code>)

2. Запустите скрипт подготовки развертывания и переключения репозитория с измененным файлом развертывания `yaml`:

```
itkf-deployment-prepare.sh cloud.yaml
```

```
itkf-maintenance-switch-repo.sh cloud.yaml
```

Однако вы также можете изменить репозиторий только для одного типа пакетов — используйте параметр `-t` со следующими значениями:

- ♦ `apt`,
- ♦ `docker`,
- ♦ `pip` и
- ♦ `all` (для всех репозиториях – по умолчанию).

Например:

```
itkf-deployment-prepare.sh cloud.yaml
```

```
itkf-maintenance-switch-repo.sh -t apt cloud.yaml
```

Скрипт изменит конфигурацию облака, а также перезапустит все облачные службы, поэтому они будут недоступны в течение нескольких минут.



Примечание.

Если вы хотите избежать перезапуска всех облачных сервисов при переключении репозитория, вы можете использовать опцию `-n`.

2.11 Обновление операционной системы на узлах

Чтобы обновить операционную систему на облачных узлах (например, для установки обновлений безопасности), вам необходимо обновить сервер репозитория новыми пакетами, а затем выполнить обновление с помощью инструмента `apt`.

Прежде всего нужно отключить все виртуальные машины (экземпляры), находящиеся на обновляемом узле. Это необходимо, поскольку в большинстве случаев процедура обновления требует перезагрузки узла.

Чтобы запустить процедуру обновления, выполните следующую команду:

```
itkf-maintenance-update-operating-system.sh cloud.yml
```

Чтобы запустить процедуру обновления на определенном узле, выполните следующую команду:

```
itkf-maintenance-update-operating-system.sh -l node_name cloud.yml
```



Предупреждение!

Чтобы завершить процедуру обновления, т.е. запустить новое ядро, нужно перезагрузить все обновленные узлы, т.е. выполнить плановое отключение и включение платформы и перезапустить все облачные сервисы. Без перезагрузки система может работать некорректно после обновления, особенно если было установлено новое ядро Linux.

2.12 Обновление версии операционной системы на узлах

Чтобы обновить операционную систему на облачных узлах до новой версии, выполните следующую процедуру:

1. Выполните обновление операционной системы на старой версии (см. раздел 2.11). Этот шаг является обязательным для обеспечения надлежащего обновления до более новой версии. После обновления выполните перезапуск узла (узлов), если необходимо, а также перезапустите облачные сервисы. Убедитесь, что облако полностью работоспособно.
 2. Выполните переключение репозитория в облаке (см. раздел 2.10) для новой версии с изменением настройки `base_os` в `big.yml`. Этот шаг является обязательным, даже если вы используете тот же самый репозиторий для новой версии. Это связано с тем, что процедура переключения репозитория изменит информацию о версии в конфигурации узлов репозитория. Убедитесь, что после переключения репозитория все службы перезапущены и облако полностью работоспособно.
-



Предупреждение!

Перед выполнением следующих шагов убедитесь, что облако может быть какое-то время недоступно, так как необходимо будет перезагрузить все узлы облака.

3. Выполните обновление версии операционной системы:

```
itkf-maintenance-upgrade-os-release.sh <-l node_name> big.yml
```

Вы можете выполнить обновление для каждого узла, используя параметр `-l`.

Вы также можете указать, с какой версии (параметр `-f`) до какой версии (параметр `-t`) вы хотите выполнить обновление.

В настоящее время поддерживается только обновление с версии `ubuntu 18.04` до `ubuntu 20.04`. (это параметры по умолчанию).



Предупреждение!

Настоятельно рекомендуется выполнять процедуру обновления версии

операционной системы частично для каждого узла, чтобы избежать длительных простоев облака во время обновления.

4. После успешного обновления перезапустите все узлы. После перезагрузки на всех узлах должна быть установлена обновленная версия операционной системы с обновленным ядром Linux.
5. Выполните перезапуск облачных сервисов и убедитесь, что облако работает.

2.13 Реконфигурация облачных интерфейсов `external_api`

Предварительные условия:

- ◆ Рабочий сервер репозитория,
- ◆ Имена сервера репозитория, вставленные на сервер DNS,
- ◆ Рабочий сервер управления облаком,
- ◆ основной файл конфигурации,
- ◆ узлы ICP, подключенные к сети.

Вы можете перенастроить существующий IP, VIP и шлюз по умолчанию на сетевом интерфейсе `external_api` всех узлов существующего развернутого облака.



Предупреждение!

Обязательно используйте сеть `internal_api` для реконфигурации. Сеть `external_api` будет перенастроена, и соединение между сервером управления облаком и облаком ICP будет потеряно после выполнения команды `itkf-*`, как указано ниже.

2.13.1 Поиск файлов конфигурации в папке `cloud-cfg`

Найдите и проверьте файлы конфигурации, подготовленные в предыдущих разделах, на сервере управления облаком в папке `/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.

`<additional-configuration-files>` — это все остальные файлы конфигурации.

2.13.2 Подготовка файла развертывания `yaml`

Чтобы перенастроить IP, VIP и шлюз по умолчанию интерфейсов `external_api`, проверьте следующие строки в файле `/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml`:

```
.
.
.
networking_common:           #networking options for all non-specific nodes
bonds:                        #bond descriptor - optional
  cspxternal:                 #specific bond name
  members:                    #list of bond members - can be interface name or mac address
    - eno1
  mode: active-backup         #bond mode - one of: balance-rr,active-backup,balance-xor, broadcast, 802.3ad,
  ip_start: 192.168.27.11/24  #node ip
```

```

gateway: 192.168.27.1          #node gatrway - optional
mtu: 1500
purpose:                      #purpose for kolla deployment - one of:
external_api,internal_api,tunnel,provider,storage,cluster
- external_api
.
.
.
kolla:                        #kolla direct variables, which are added to globals.yml
kolla_base_distro: 'ubuntu'
kolla_install_type: 'source'
enable_neutron_provider_networks: 'yes'
kolla_internal_vip_address: '192.168.39.21'
kolla_external_vip_address: '192.168.27.21'
.
.
.

```

В этом примере используется `ip_start: 192.168.27.11/24` и `gateway: 192.168.27.1` для сети `external_api`, а также `kolla_external_vip_address: '192.168.27.21'`. Чтобы перенастроить IP, VIP и шлюз по умолчанию интерфейсов `external_api`, файл развертывания `yaml` должен быть изменен следующим образом:

```

.
.
.
networking_common:          #networking options for all non-specific nodes
bonds:                      #bond descriptor - optional
  cspxternal:               #specific bond name
  members:                  #list of bond members - can be interface name or mac address
    - eno1
  mode: active-backup       #bond mode - one of: balance-rr,active-backup,balance-xor, broadcast, 802.3ad,
  ip_start: 192.168.41.211/24 #node ip
  gateway: 192.168.41.253   #node gatrway - optional
  mtu: 1500
  purpose:                  #purpose for kolla deployment - one of:
external_api,internal_api,tunnel,provider,storage,cluster
- external_api
.
.
.
kolla:                      #kolla direct variables, which are added to globals.yml
kolla_base_distro: 'ubuntu'
kolla_install_type: 'source'
enable_neutron_provider_networks: 'yes'
kolla_internal_vip_address: '192.168.39.21'
kolla_external_vip_address: '192.168.41.221'
.
.
.

```

Параметры `ip_start` и `gateway` изменены на другую подсеть, и адрес `kolla_external_vip_address` тоже из этой подсети.

2.13.3 Подготовка среды развертывания

После того, как все необходимые настройки облака выполнены, запустите скрипт BASH на сервере управления облаком:

```

cd /opt/cloud-cfg/[cloud]

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

```

Он повторно сгенерирует файлы *globals.yml* и *cloud-inventory* и скопирует 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>/*.

2.13.4 Подготовка сертификата

2.13.4.1 Повторное создание самоверяющего сертификата

Если балансировщики нагрузки HAProxy используют протокол TLS на внешних конечных точках API с помощью самоверяющих сертификатов, повторно создайте самоверяющий сертификат. Запустите скрипт *itkf-generate-certificates.sh*, используя параметр *-f* с файлом развертывания *yamI*, который был проверен на предыдущих шагах:

```
itkf-generate-certificates.sh -f <cloud-name>-big.yml
```

2.13.4.2 Обновление сертификата

Если балансировщики нагрузки HAProxy используют протокол TLS на внешних конечных точках API с помощью сертификатов, выданных доверенным центром сертификации, обновите сертификат.



Предупреждение!

Обязательно используйте сеть *internal_api* для реконфигурации. Используйте параметр *-i* при вводе команд *itkf-**, как указано ниже!

2.13.5 Запуск реконфигурации облачных интерфейсов *external_api*

Чтобы проверить, активны ли узлы ICP, нужно выполнить команду:

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

Запустите скрипт *itkf-deployment-reconfigure-external-api.sh* с файлом развертывания *yamI*, который был подготовлен на предыдущих шагах.

```
itkf-deployment-reconfigure-external-api.sh -i <cloud-name>-big.yml
```

3 Настройка безопасности TLS

Предварительные условия:

- ♦ рабочий сервер управления облаком,
- ♦ основной файл конфигурации,
- ♦ узлы ICP, подключенные к сети.

Дополнительным параметром конфигурации конечной точки облачных сервисов является включение или отключение защиты TLS для внешнего и/или внутреннего виртуального IP-адреса на балансировщиках нагрузки HAProxy. TLS позволяет клиенту аутентифицировать конечную точку службы OpenStack и позволяет шифровать запросы и ответы. При этом не всегда практично использовать сертификат, подписанный известным доверенным ЦС, например, при разработке или внутреннем тестовом развертывании облака. В этих случаях может быть полезно использовать самоподписанный сертификат.

3.1 Поиск файлов конфигурации в папке cloud-cfg

Найдите и проверьте предварительно подготовленные файлы конфигурации на сервере управления облаком в папке `/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` – основной файл конфигурации.

`<additional-configuration-files>` – все дополнительные файлы конфигурации.

3.2 Изменение файлов конфигурации в папке cloud-cfg

Файл развертывания `yaml` должен иметь путь `/opt/cloud-cfg/<cloud-name>/<cloud-name>-big.yml`.

Чтобы включить защиту TLS для конечных точек OpenStack, нужно сначала проверить следующие строки.

```
.  
. .  
#kolla specific  
kolla: #kolla direct variables, which are added to globals.yml  
.  
.  
.  
enable_prometheus: 'yes'  
enable_prometheus_haproxy_exporter: '{{ enable_haproxy | bool }}'  
enable_prometheus_mysql_exporter: '{{ enable_mariadb | bool }}'  
enable_prometheus_node_exporter: '{{ enable_prometheus | bool }}'  
enable_prometheus_cadvisor: '{{ enable_prometheus | bool }}'  
enable_prometheus_memcached: '{{ enable_prometheus | bool }}'  
enable_prometheus_alertmanager: '{{ enable_prometheus | bool }}'
```

```
enable_prometheus_ceph_mgr_exporter: '{{ enable_prometheus | bool and enable_ceph | bool }}'  
enable_prometheus_libvirt_exporter: '{{ enable_prometheus | bool }}'
```

В случае включения защиты TLS для *внешних* конечных точек ICP, файл развертывания `yaml` должен быть изменен следующим образом:

```
#kolla specific
```

```
kolla: #kolla direct variables, which are added to globals.yml
```

```
enable_prometheus: 'yes'  
enable_prometheus_haproxy_exporter: '{{ enable_haproxy | bool }}'  
enable_prometheus_mysql_exporter: '{{ enable_mariadb | bool }}'  
enable_prometheus_node_exporter: '{{ enable_prometheus | bool }}'  
enable_prometheus_cadvisor: '{{ enable_prometheus | bool }}'  
enable_prometheus_memcached: '{{ enable_prometheus | bool }}'  
enable_prometheus_alertmanager: '{{ enable_prometheus | bool }}'  
enable_prometheus_ceph_mgr_exporter: '{{ enable_prometheus | bool and enable_ceph | bool }}'  
enable_prometheus_libvirt_exporter: '{{ enable_prometheus | bool }}'  
kolla_enable_tls_external: "yes".
```

Для протокола TLS на внешнем виртуальном IP-адресе должно быть задано значение `yes`.

В случае включения защиты TLS для *внутренних* конечных точек ICP, файл развертывания `yaml` должен быть изменен следующим образом:

```
#kolla specific
```

```
kolla: #kolla direct variables, which are added to globals.yml
```

```
enable_prometheus: 'yes'  
enable_prometheus_haproxy_exporter: '{{ enable_haproxy | bool }}'  
enable_prometheus_mysql_exporter: '{{ enable_mariadb | bool }}'
```

```
enable_prometheus_node_exporter: '{{ enable_prometheus | bool }}'  
enable_prometheus_cadvisor: '{{ enable_prometheus | bool }}'  
enable_prometheus_memcached: '{{ enable_prometheus | bool }}'  
enable_prometheus_alertmanager: '{{ enable_prometheus | bool }}'  
enable_prometheus_ceph_mgr_exporter: '{{ enable_prometheus | bool and enable_ceph | bool }}'  
enable_prometheus_libvirt_exporter: '{{ enable_prometheus | bool }}'  
kolla_enable_tls_internal: 'yes'  
kolla_copy_ca_into_containers: 'yes'  
kolla_admin_openrc_cacert: '{{ kolla_certificates_dir }}/ca/root.crt'  
openstack_cacert: '/etc/ssl/certs/ca-certificates.crt'  
.  
.  
.
```

Протокол TLS для внутреннего виртуального IP-адреса также должен быть установлен на `yes`. Также требуется скопировать ЦС во все контейнеры. Переменная `openstack_cacert` должна быть настроена с указанием пути к сертификатам ЦС в контейнере.

3.3 Подготовка среды развертывания

После того, как все необходимые настройки облака выполнены, запустите скрипт BASH на сервере управления облаком:

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

Он повторно сгенерирует файлы `globals.yml` и `cloud-inventory` и скопирует 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>/`.

3.4 Подготовка сертификата

Существует два варианта подготовки сертификатов TLS:

- создать самоподписанный сертификат или сертификаты,
- использовать прямо указанный сертификат или сертификаты.

HTTPS будет включен как для панели управления ICP, так и для конечных точек общедоступного или внутреннего API.

3.4.1 Создание самоподписанных сертификатов

Запустите скрипт `itkf-generate-certificates.sh` с файлом развертывания `yaml`, который мы проверили на предыдущих шагах:

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

```
itkf-generate-certificates.sh <cloud-name>-big.yml
```

Объединенный файл сертификата и ключа `haproxy.pem` (который является значением по умолчанию для `kolla_external_fqdn_cert`) будет сгенерирован и сохранен в карте контейнера `kolla-ansible-srv` – `/etc/kolla/cloud-cfg/$(CLOUD_NAME)/certificates`. Объединенный файл сертификата и ключа `haproxy-internal.pem` (который является значением по умолчанию для

kolla_internal_fqdn_cert) будет сгенерирован и сохранен в карте контейнера *kolla-ansible-srv – /etc/kolla/cloud-cfg/Cloud_NAME/certificates*. Копия сертификата ЦС *root.crt* будет сохранена в карте контейнера *kolla-ansible-srv – /etc/kolla/cloud-cfg/Cloud_NAME/certificates/ca/*.

3.4.2 Прямо указанный сертификат

При наличии зашифрованных сертификатов PEX с закрытым ключом требуется преобразование в обычный текст (и расшифровка) для промежуточного сертификата CER, т.е. сертификата цепочки, открытого сертификата CRT и закрытого ключа KEY.

В нашем примере *maket_wildcard.pfx* использует формат PEX, поэтому требуется преобразование в простые текстовые компоненты:

```
openssl pkcs12 -in maket_wildcard.pfx -nocerts -out maket_wildcard.key
```

```
Enter Import Password: [пароль]
```

```
Enter PEM pass phrase: [пароль]
```

```
Verifying - Enter PEM pass phrase: сертификат
```

```
openssl rsa -in maketa_wilcard.key -out maket_wildcard_unencrypted.key
```

```
Enter pass phrase for maketa_wilcard.key: [пароль]
```

```
writing RSA key
```

```
openssl pkcs12 -in maket_wildcard.pfx -clcerts -nokeys -out haproxy.pem
```

```
Enter Import Password: [пароль]
```

```
openssl pkcs12 -in maket_wildcard.pfx -cacerts -nokeys -chain -out root.crt
```

```
Enter Import Password: [пароль]
```

Для безопасного использования TLS с аутентификацией требуются два файла сертификата, которые будут предоставлены вашим центром сертификации:

- сертификат сервера с закрытым ключом,
- сертификат ЦС с любыми промежуточными сертификатами.

Объединенный сертификат и закрытый ключ сервера необходимо предоставить в модуль Kolla Ansible с путем, настроенным через *kolla_external_fqdn_cert* и/или *kolla_internal_fqdn_cert*. Эти пути по умолчанию — *Cloud_NAME/certificates/haproxy.pem* и *Cloud_NAME/certificates/haproxy-internal.pem* соответственно, где *kolla_certificates_dir* — это */etc/kolla/cloud-cfg/Cloud_NAME/certificates*.

Итак, файлы *haproxy.pem* и *haproxy-internal.pem* необходимо скопировать в карту контейнера *kolla-ansible-srv – /etc/kolla/cloud-cfg/Cloud_NAME/certificates*.

3.5 Подготовка среды развертывания

После того, как все необходимые настройки облака выполнены, запустите скрипт BASH на сервере управления облаком:

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

Он повторно сгенерирует файлы `globals.yml`, `passwords.yml` и `cloud-inventory` и скопирует 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>/`.

3.6 Обновление развернутого облака

Чтобы проверить, активны ли узлы ICP, выполните:

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

Запустите скрипт `itkf-deployment-reconfigure.sh` с файлом развертывания `yaml`, который был подготовлен на предыдущих шагах.

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

3.7 Проверка состояния облака

Чтобы выполнить проверку конфигурации OpenStack TLS:

1. Откройте браузер и введите внешний адрес VIP.
2. Подтвердите исключение безопасности в отношении самоподписанного сертификата.
3. Войдите в панель управления. Пароль для пользователя *admin* предоставляется системным администратором.

4 Устранение ошибок

4.1 Контролируемый перезапуск служб облака

Если возникают какие-либо проблемы с работой облака, это означает, что те или иные службы могут работать со сбоями. Если не выявлен факт неправильной настройки, службу может потребоваться только перезапустить.

В этом случае служба и ее базовые подсервисы должны быть перезапущены в правильном порядке с помощью скрипта `itkf-maintenance-restart-service.sh`.

В случае, если нужно настроить и/или обновить какую-либо службу, до выполнения соответствующей процедуры ее нужно остановить, а после – повторно запустить. В этом случае можно использовать скрипты `itkf-maintenance-stop-service.sh` и `itkf-maintenance-start-service.sh`. В настоящее время поддерживаются следующие службы:

- mariadb,
- horizon,
- heat,
- neutron,
- openvswitch,
- nova,
- cinder,
- glance,
- ceph,
- keystone,
- prometheus,
- memcached,
- keepalived,
- haproxy,
- chrony,
- cron,
- kolla_toolbox,
- fluentd,
- rabbitmq.

Чтобы перезапустить службу, используйте следующий синтаксис:

```
itkf-maintenance-restart-service.sh cloud_name service_name
```

Чтобы запустить службу, используйте следующий синтаксис:

```
itkf-maintenance-start-service.sh cloud_name service_name
```

Чтобы остановить службу, используйте следующий синтаксис:

```
itkf-maintenance-stop-service.sh cloud_name service_name
```

В качестве имени службы можно использовать ключ `all` (все службы).

Вы можете использовать параметр `-f` для принудительного перезапуска службы, которая фактически работает (это предпочтительный вариант).

Вы также можете использовать параметр `-n node_name`, чтобы указать узел, на котором нужно перезапустить службу. По умолчанию перезапуск выполняется на всех узлах.

Предыдущие два параметра не разрешены для службы 'mariadb'.

Несколько примеров приведены ниже.

Перезапустить все службы на всех узлах в облаке bb-8:

```
itkf-maintenance-restart-service.sh -f bb-8 all
```

Перезапустить службу nova на всех узлах в облаке bb-8:

```
itkf-maintenance-restart-service.sh -f bb-8 nova
```

Выполнить восстановление mariadb в облаке bb-8 (восстановление mariadb не позволяет использовать параметр -n):

```
itkf-maintenance-restart-service.sh bb-8 mariadb
```

Перезапустить службу Cinder на всех узлах в облаке bb-8, если какая-либо из базовых служб не работает:

```
itkf-maintenance-restart-service.sh bb-8 cinder
```

Перезапустить службу openvswitch на узле node3.cloud в облаке bb-8:

```
itkf-maintenance-restart-service.sh -f -n node3.cloud bb-8 openvswitch
```

Остановить все службы в облаке bb-8:

```
itkf-maintenance-stop-service.sh bb-8
```

Запустить все сервисы в облаке bb-8:

```
itkf-maintenance-start-service.sh bb-8
```

Запустить службу mariadb и выполните восстановление mariadb в облаке bb-8:

```
itkf-maintenance-start-service.sh bb-8 mariadb
```

Остановить сервис Horizon в облаке bb-8:

```
itkf-maintenance-stop-service.sh bb-8 horizon
```

4.2 Исправление службы dockerd

В некоторых случаях после незапланированного перезапуска или отключения облачного узла служба dockerd может зависнуть таким образом, что даже команда docker не работает. В этом случае используйте следующие две команды на всех облачных узлах:

```
find /var/lib/docker/containers -type f -name "config.v2.json" -print0 | xargs -0 sed -i '  
-e 's/Running":true/Running":false/g'  
systemctl restart docker
```

Это приведет к тому, что все контейнеры перейдут в нерабочее состояние, они будут правильно запущены после перезапуска службы docker.

4.3 Изменение количества реплик в пуле Ceph

Если происходит удаление (принудительное) сервисного и/или вычислительного узла, остается менее трех (3) узлов хранения. В этом случае оператор должен сократить количество реплик для всех пулов Ceph до количества оставшихся узлов хранения.



Предупреждение!

После добавления узла (по причине замены узла) в кластере из трех узлов следует увеличить количество реплик для всех пулов Ceph обратно до числа 3.

Чтобы задать количество реплик объектов в реплицируемом пуле, выполните:

```
ceph osd pool set <poolname> size <num-replicas>
```

Значение `<num-replicas>` включает сам объект. Если требуется объект и две (2) копии объекта, т.е. всего три (3) экземпляра объекта, введите 3.

Чтобы уменьшить количество реплик для всех пулов Ceph, выполните следующую процедуру:

1. Выведите в список все пулы Ceph:

```
(on <cloud-name>-node1) ceph osd pool ls  
  
images  
  
volumes  
  
backups  
  
vms
```

2. Установите размер `size` всех пулов Ceph равным 2:

```
(on <cloud-name>-node1)ceph osd pool set images size 2  
  
(on <cloud-name>-node1)ceph osd pool set volumes size 2  
  
(on <cloud-name>-node1)ceph osd pool set backups size 2  
  
(on <cloud-name>-node1)ceph osd pool set vms size 2
```

3. Проверьте размер `size` всех пулов Ceph:

```
(on <cloud-name>-node1) ceph osd dump | grep 'replicated size'  
  
pool 5 'images' replicated size 2 min_size 1 crush_rule 1 object_hash rjenkins pg_num 3  
2 pgp_num 32 last_change 301 flags hashpspool stripe_width 0 application rbd  
  
pool 6 'volumes' replicated size 2 min_size 1 crush_rule 1 object_hash rjenkins pg_num  
32 pgp_num 32 last_change 303 flags hashpspool stripe_width 0 application rbd  
  
pool 7 'backups' replicated size 2 min_size 1 crush_rule 1 object_hash rjenkins pg_num  
32 pgp_num 32 last_change 304 flags hashpspool stripe_width 0 application rbd  
  
pool 8 'vms' replicated size 2 min_size 1 crush_rule 1 object_hash rjenkins pg_num 32 p  
gp_num 32 last_change 305 flags hashpspool stripe_width 0 application rbd
```

4.4 Неустранимый сбой облачного вычислительного узла

В случае неустранимого сбоя вычислительного узла необходимо установить новый узел вместо старого.

Вы можете принудительно удалить существующий неисправный (и отключенный) аппаратный вычислительный узел или узел хранения из существующего развернутого облака (см. раздел 2.4).

Вы можете добавить новый вычислительный узел или узел хранения в существующее развернутое облако (см. раздел 2.2).

4.5 Неустранимый сбой облачного сервисного узла

В случае неустранимого сбоя сервисного узла в кластерах с тремя узлами необходимо установить новый сервисный узел вместо старого.

Вы можете принудительно удалить неисправный (и отключенный) аппаратный узел управления, сети, базы данных, вычислений или мониторинга (см. раздел 2.7).

Вы можете добавить новый узел управления, сети, базы данных, вычислений или мониторинга в существующее развернутое облако (см. раздел 2.5).

4.6 Сбой сети

Сетевой сбой может проявляться по-разному: перестают работать сетевые службы, теряется связь между виртуальными машинами, случайная потеря пакетов, сбой связи по определенным протоколам и т.д.

Могут быть разные причины таких сбоев, например: аппаратный сбой, неверная конфигурация, сбой программного обеспечения сетевой службы и т.д.

4.6.1 Проверка физической сети и оборудования

При сбое подключения к сети, чтобы определить причину сбоя, проверьте:

- физическое подключение хостов к физической сети (неисправные кабели и/или разъемы),
- конфигурацию коммутатора ToR,
- конфигурацию агрегирования интерфейсов (конфигурация коммутатора ToR должна соответствовать конфигурации на физических серверах)
- конфигурации VLAN и все необходимые транкинговые соединения с граничными портами на коммутаторе ToR,
- конфигурации VLAN и все необходимые транкинговые соединения с портами восходящей линии связи на коммутаторах ToR (если применимо) и/или маршрутизаторах,
- что хосты находятся в одной IP-подсети или между ними установлена правильная маршрутизация,
- что к хостам не применяются iptables, которые запрещали бы трафик.

Чтобы проверить сбой в физическом подключении, вы можете использовать инструмент `itkf-maintenance-verify-net.sh` на сервере управления облаком, который запускает серию простых ring-тестов сети для всех интерфейсов на облачных узлах. В качестве входных данных требуется конфигурация `yaml` облака, как показано в следующем примере:

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

Если вы хотите получить более подробную информацию, то есть узнать, какие сети являются проблемными (тесты выполняются с учетом назначения сети), вы можете выполнить:

```
itkf-maintenance-verify-net.sh -d cloud.yaml
```

Вторая команда может выполняться дольше, чем первая.

В дополнение к ring-тестам на платформе ICP можно проводить сетевые стресс-тесты. Эти тесты используют инструмент `iperf` для отправки сетевого трафика через сетевые интерфейсы ICP и сбора данных, включая показатели пропускной способности сети и любые ошибки, возникшие во время тестов.



Примечание.

Невозможно запустить сетевые стресс-тесты для сетевых интерфейсов, назначение которых – DPDK или SRIOV.

Этот тест включает в себя следующие шаги:

- собрать текущую статистику сетевых интерфейсов, включая ошибки сетевого трафика и потери сетевых пакетов,
- запустить 10-секундные тесты пропускной способности сети, включая все сетевые интерфейсы, определенные в файле `yaml` облака,

- переключить любые активные или резервные связки интерфейсов (bond) для передачи трафика на неактивные каналы,
- повторно запустить 10-секундные тесты пропускной способности сети, включая все сетевые интерфейсы, определенные в файле `yaml` облака,
- переключить любые активные/резервные связки интерфейсов обратно в исходное состояние,
- собрать окончательную статистику сетевых интерфейсов, включая ошибки сетевого трафика и потери сетевых пакетов,
- вывести всю информацию, включая пропускную способность всех соединений, сетевые ошибки, отброшенные пакеты и увеличение количества сетевых ошибок и отброшенных пакетов.

Чтобы выполнить этот тест, запустите скрипт:

```
itkf-maintenance-stresstest-net.sh <-t single_test_duration> cloud_name.yaml
```

По умолчанию для `single_test_duration` установлено значение 10 с.

Выполнение этой команды может занять много времени, так как проверяется каждая комбинация интерфейсов. Также, чем больше облачных узлов, тем дольше длится тестирование. Чтобы определить, как долго будут выполняться эти тесты, можно использовать следующую формулу:

$$n*n*v*(d+t)*2$$

где:

- `n` – количество узлов в облаке,
- `d` – продолжительность одного теста (устанавливается параметром `-t` и по умолчанию равна 10 с),
- `v` – количество используемых интерфейсов на узлах ICP (обычно это число равно 4),
- `t` – время подготовки выполнения `ansible` (обычно это около 2-5 с).



Предупреждение!

Во время стресс-теста сеть находится под большой нагрузкой, поэтому производительность платформы ICP и пропускная способность сетей клиентов могут быть ограничены, а работа облачных сервисов может быть серьезно нарушена.

Результаты стресс-тестов сети записываются в файл:

```
/var/log/kolla-ansible/stress-testing.log
```

4.6.2 Проверка сетевых служб

Проверьте, все ли сетевые службы запущены и работают.

- Используйте клиент `openstack` для проверки списков агентов с аргументами: `network agent list`. Если вы используете клиент `itopenstack`, выполните следующую команду:

```
itopenstack cloud_name network agent list
```

где `cloud_name` — имя вашего облака. Например:

```
itopenstack my_cloud network agent list
```

ID	Agent Type	Host	Availability Zone
True UP c3a525cb-571a-4e21-a4dd-95a6616b1d00 neutron-metadata-agent	Metadata agent	csp1locata5	None
True UP e8b957d2-bea0-4419-9f1a-69f82ed01db8 neutron-metadata-agent	Metadata agent	csp1locata4	None
True UP 73ae7bea-e6a2-4569-8d12-2532087c61b8 neutron-l3-agent	L3 agent	csp1locata4	nova
True UP e079582b-6a7f-4925-802d-5e0646f87e41 neutron-openvswitch-agent	Open vSwitch agent	csp1locata3	None
True UP 2781e35f-8ff6-4d3f-933a-f0b94c5231b7 neutron-dhcp-agent	DHCP agent	csp1locata4	nova
True UP 064d4e4d-3e89-4388-8762-6cd8cb84fb75 neutron-dhcp-agent	DHCP agent	csp1locata3	nova
True UP f24f2fb3-c912-4326-8189-b35ed19bfff3 neutron-openvswitch-agent	Open vSwitch agent	csp1locata4	None
True UP ae308201-8280-435c-881f-661c5414d776 neutron-metadata-agent	Metadata agent	csp1locata3	None
True UP 9d22f934-e632-45b3-a885-0168a1d2582b neutron-dhcp-agent	DHCP agent	csp1locata5	nova
True UP 26ab6f15-15e1-48ea-a869-b3c3c92b5964 neutron-openvswitch-agent	Open vSwitch agent	csp1locata5	None
True UP d4718186-1e93-4b2e-b6a9-ee2a35faea40 neutron-l3-agent	L3 agent	csp1locata5	nova
True UP 5c2f5016-024f-49bb-96f4-40ff41bb8849 neutron-l3-agent	L3 agent	csp1locata3	nova

- Результат показывает состояние сетевых служб на каждом управляющем узле. Все они должны иметь параметр Alive со значением True, а параметр State – со значением UP.

4.6.3 Проверка конфигурации сети

Проверьте, правильно ли задано значение MTU.

- Некоторые сети (туннелированные) требуют меньшего значения MTU, установленного на VM, из-за инкапсуляции. Так что MTU на VM должен быть ниже 1500.
- В случае наличия сетей провайдеров на основе VLAN, изменять MTU не требуется.

4.6.4 Устранение неполадок сетевого пути

- Если все предыдущие шаги не выявили проблем, необходимо найти место сбоя. Необходимо найти и проверить все виртуальные сетевые устройства, которые используются для обработки сетевого трафика конкретной виртуальной машины.

- В следующем примере показано, как найти правильные сетевые устройства для конкретной виртуальной машины, т.е. экземпляра:
 - а) Экземпляр (ВМ) генерирует пакет и помещает его на виртуальную сетевую карту внутри экземпляра, например eth0.
 - б) Пакет передается на устройство Test Access Point (TAP) на вычислительном узле, например, tap690466bc-92. Вы можете узнать, какой TAP используется, просмотрев файл `/etc/libvirt/qemu/instance-xxxxxxx.xml` в контейнере nova_libvirt, где xxxxxx – имя гипервизора экземпляра. Чтобы сопоставить имя гипервизора экземпляра с реальным именем экземпляра, в openstack запустите команду `server show my_instance` и найдите поле `OS-EXT-SRV-ATTR:instance-name`. В следующем примере используется клиент itopenstack с именем my_cloud:

```
itopenstack my_cloud server show my_instance
```

```

+-----+-----+
---+
| Field          | Value          |
+-----+-----+
---+
| OS-DCF:diskConfig | AUTO          |
| OS-EXT-AZ:availability_zone | nova          |
| OS-EXT-SRV-ATTR:host | bb-8-node5.iskrauraltel.r |
| OS-EXT-SRV-ATTR:hypervisor_hostname | bb-8-node5.iskrauraltel.r |
| OS-EXT-SRV-ATTR:instance_name | instance-00000006 |
| OS-EXT-STS:power_state | Running       |
| OS-EXT-STS:task_state | Non           |
| OS-EXT-STS:vm_state | active        |
| OS-SRV-USG:launched_at | 2019-07-25T12:25:42.000000 |
| OS-SRV-USG:terminated_at | Non           |
| accessIPv4      |               |
| accessIPv6      |               |
| addresses       | admin_external=192.168.41.9 |
| config_drive    |               |

```

```

| created          | 2019-07-25T12:22:48
Z
| flavor          | r.small (e634aa0b-25e7-4008-8585-9c8fc7cca48a
)
| hostId         | a3bf2211d23a0305ffc42f9b59d7aa8070decb4d9c5da31fdc868
f7
| id             | 4da0263a-288a-4a4d-9b06-8537ee47aa4
2
| image          | rally-test-image (a1047e36-ddda-486c-a816-260895af3eea
)
| key_name       | Non
e
| name           | tes
t
| progress       |
0
| project_id     | 59d1efb4ccfb484fb517e565ebdce01
5
| properties     |
s
| security_groups | name='default
|
| status         | ACTIV
E
| updated        | 2019-07-25T13:08:03
Z
| user_id        | 902aecbb07e54f2e9f8d47e51eec112
6
| volumes_attach |
d
+-----+-----+
----+

```

Чтобы проверить содержимое файла гипервизора конкретного экземпляра, сначала выполните команду `bash` в контейнере `nova_libvirt` на вычислительном узле, на котором размещен этот экземпляр. Найдите конкретный файл, который затем можно просмотреть (с помощью команды `cat`). В следующем примере используется экземпляр из предыдущего примера:

```

node-x#docker exec -it nova_libvirt /bin/bash
nova-libvirt#cat /etc/libvirt/qemu/
nova-libvirt#cat /etc/libvirt/qemu/instance-00000006

```

В этом файле вы можете найти информацию об интерфейсе и найти имя TAP в поле: `target dev`. Например:

```
<target dev='tap8c78d87a-dc' />
```

- в) Имя устройства TAP создается с использованием первых 11 символов идентификатора порта (10 шестнадцатеричных цифр плюс знак «-»), поэтому другой способ найти имя устройства – использовать клиент `openstack` с аргументами `port list`. Он возвращает список портов с разделителями вертикальной черты, первым

элементом которого является идентификатор порта. Например, чтобы получить идентификатор порта, связанный с IP-адресом 10.0.0.10, выполните:

```
itopenstack my_cloud port list | grep 10.0.0.10 | cut -d \| -f 2
ff387e54-9e54-442b-94a3-aa4481764f1d
```

Первые 11 символов составляют имя устройства `tapff387e54-9e`.

- г) Устройство TAP подключено к интеграционному мосту, `br-int`. Этот мост соединяет все устройства TAP экземпляра и любые другие мосты в системе. В этом примере у нас есть интерфейсы `int-br-eth1` и `patch-tun`. `int-br-eth1` — это половина пары `veth`, которая подключается к мосту `br-eth1`, который управляет сетями VLAN, имеющими соединительные линии на базе физического устройства Ethernet `eth1`. `patch-tun` — это внутренний порт Open vSwitch, который подключается к мосту `br-tun` для сетей VXLAN.
- д) Устройства TAP и устройства `veth` являются обычными сетевыми устройствами Linux и могут быть проверены с помощью обычных инструментов, таких как `ip` и `tcpdump`. Внутренние устройства Open vSwitch, такие как `patch-tun`, видны только в среде Open vSwitch. Если вы попытаетесь выполнить `tcpdump -i patch-tun`, вы получите ошибку о том, что устройство не существует.
- Захват пакетов на определенном сетевом устройстве:
 - а) Можно просматривать пакеты на внутренних интерфейсах, но для этого потребуется выполнить некоторые сетевые настройки. Сначала нужно создать фиктивное сетевое устройство, которое могут видеть обычные инструменты Linux. Затем нужно добавить его в мост, содержащий внутренний интерфейс, за которым вы хотите следить. Наконец, нужно указать Open vSwitch зеркалировать весь трафик, входящий или исходящий из внутреннего порта, на этот фиктивный порт. После всего этого вы можете запустить `tcpdump` на фиктивном интерфейсе и посмотреть трафик на внутреннем порту.
 - б) Создайте и вызовите фиктивный интерфейс `snooper0` (вы можете выполнить эту команду на узле, где находится экземпляр, или в контейнере `openvswitch_vswitchd` на том же узле):


```
node-x# ip link add name snooper0 type dummy
node-x# ip link set dev snooper0 up
```
 - в) Остальные команды нужно выполнить внутри контейнера `openvswitch_vswitchd` на узле, где находится экземпляр.

Добавьте устройство `snooper0` в мост `br-int`:

```
node-x#docker exec -it openvswitch_vswitchd /bin/bash
openvswitch_vswitchd# ovs-vsctl add-port br-int snooper0
```

- г) Создайте зеркало `patch-tun` на `snooper0` (возвращает UUID порта зеркала):

```
openvswitch_vswitchd# ovs-vsctl -- set Bridge br-int mirrors=@m -- --id=@snooper0 \
get Port snooper0 -- --id=@patch-tun get Port patch-tun \
-- --id=@m create Mirror name=mymirror select-dst-port=@patch-tun \
select-src-port=@patch-tun output-port=@snooper0 select_all=1
```

д) Теперь можно посмотреть трафик на patch-tun, выполнив tcpdump -i snooper0:

```
openvswitch_vswitchd# openvswitch_vswitchd# tcpdump -i snooper0
```

е) Выполните очистку, удалив все зеркала на br-int и фиктивный интерфейс:

```
openvswitch_vswitchd# ovs-vsctl clear Bridge br-int mirrors
```

```
openvswitch_vswitchd# ovs-vsctl del-port br-int snooper0
```

```
openvswitch_vswitchd# ip link delete dev snooper0
```

- Определите внутреннюю конфигурацию VLAN.

- а) На интеграционном мосту сети различаются с помощью внутренних VLAN независимо от того, как их определяет сетевая служба. Это позволяет экземплярам на одном хосте взаимодействовать напрямую, минуя остальную часть виртуальной или физической сети. Эти внутренние идентификаторы VLAN основаны на порядке их создания на узле и могут различаться между узлами. Эти идентификаторы никак не связаны с идентификаторами сегментации, используемыми в определении сети и на физической линии.
- б) Теги VLAN преобразуются из внешних тегов, определенных в настройках сети, во внутренние теги в нескольких местах. На мосту br-int входящие пакеты от int-br-eth1 преобразуются из внешних тегов во внутренние теги. Другие преобразования также происходят на других мостах и описываются в соответствующих разделах.
- в) Узнать, какой внутренний тег VLAN используется для данной внешней VLAN, можно с помощью команды ovs-ofctl.
 - Найдите тег внешней VLAN интересующей вас сети. Это параметр provider:segmentation_id, возвращаемый сетевой службой:

```
itopenstack my_cloud network show --fields provider:segmentation_id <network name>
```

```
+-----+-----+
| Field          | Value          |
+-----+-----+
| provider:network_type | vlan          |
| provider:segmentation_id | 2113         |
+-----+-----+
```

- Остальные команды нужно выполнить внутри контейнера openvswitch_vswitchd на узле, где находится экземпляр.

Выполните команду grep для provider:segmentation_id, в данном случае 2113, в выводе ovs-ofctl dump-flows br-int:

```
node-x#docker exec -it openvswitch_vswitchd /bin/bash
openvswitch_vswitchd# ovs-ofctl dump-flows br-int | grep vlan=2113
cookie=0x0, duration=173615.481s, table=0, n_packets=7676140,
n_bytes=444818637, idle_age=0, hard_age=65534, priority=3,
in_port=1,d1_vlan=2113 actions=mod_vlan_vid:7,NORMAL
```

- Здесь вы можете видеть, что пакеты, принятые на порту с идентификатором 1 с тегом VLAN 2113, изменены и имеют внутренний тег VLAN 7. В подробностях можно увидеть, что порт 1 является интерфейсом int-br-eth1 на самом деле:

```

openvswitch_vswitchd# ovs-ofctl show br-int
OFPT_FEATURES_REPLY (xid=0x2): dpid:000022bc45e1914b
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS
ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC
SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC
SET_TP_DST ENQUEUE
1(int-br-eth1): addr:c2:72:74:7f:86:08
    config:      0
    state:      0
    current:    10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
2(patch-tun):  addr:fa:24:73:75:ad:cd
    config:      0
    state:      0
    speed: 0 Mbps now, 0 Mbps max
3(tap9be586e6-79): addr:fe:16:3e:e6:98:56
    config:      0
    state:      0
    current:    10MB-FD COPPER
    speed: 10 Mbps now, 0 Mbps max
LOCAL(br-int):  addr:22:bc:45:e1:91:4b
    config:      0
    state:      0
    speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0

```

- Сети на основе VLAN выходят из интеграционного моста через интерфейс veth int-br-eth1 и поступают на мост br-eth1 на другой половине пары veth phy-br-eth1. Пакеты на этот интерфейс поступают с внутренними тегами VLAN и преобразуются во внешние теги в обратном порядке:

```

openvswitch_vswitchd# ovs-ofctl dump-flows br-eth1 | grep 2113
cookie=0x0, duration=184168.225s, table=0, n_packets=0, n_bytes=0,
idle_age=65534, hard_age=65534, priority=4,in_port=1,d1_vlan=7

```

```
actions=mod_vlan_vid:2113,NORMAL
```

- Пакеты, теперь помеченные внешним тегом VLAN, затем выходят в физическую сеть через eth1. Коммутатор уровня 2, к которому подключен этот интерфейс, должен быть настроен на прием трафика с используемым идентификатором VLAN. Следующий переход для этого пакета также должен находиться в той же сети уровня 2.

4.6.5 Устранение ошибок Open vSwitch

Полную документацию по Open vSwitch можно найти на сайте проекта. На практике, учитывая предварительно описанную конфигурацию, наиболее распространенные трудности возникают при настройке требуемых мостов (br-int, br-tun и br-ex) и подключении к ним портов.

Драйвер Open vSwitch обычно управляет этим автоматически, но полезно знать, как сделать это вручную с помощью команды `ovs-vsctl`. Эта команда имеет намного больше подкоманд, чем будут далее описаны; см. справочную страницу или используйте `ovs-vsctl --help` для получения полного списка.

Все команды, касающиеся Open vSwitch, должны выполняться внутри контейнера `openvswitch_vswitchd` на узле, где расположены конкретные экземпляры.

1. Чтобы вывести список мостов в системе, выполните `ovs-vsctl list-br`. В этом примере показан вычислительный узел с внутренним мостом и туннельным мостом. Сети VLAN организуются через сетевой интерфейс eth1:

```
node-x#docker exec -it openvswitch_vswitchd /bin/bash
openvswitch_vswitchd# ovs-vsctl list-br
br-int
br-tun
eth1-br
```

2. Двигаясь вниз от физического интерфейса, можно увидеть цепочку портов и мостов. Во-первых, мост `eth1-br`, который содержит физический сетевой интерфейс `eth1` и виртуальный интерфейс `phy-eth1-br`:

```
openvswitch_vswitchd# ovs-vsctl list-ports eth1-br
eth1
phy-eth1-br
```

3. Затем следует внутренний мост `br-int`, он содержит `int-eth1-br`, который в паре с `phy-eth1-br` подключается к физической сети, представленной в предыдущем мосте, `patch-tun`, который используется для подключения к туннельному мосту GRE и устройствам TAP, которые подключаются к экземплярам, работающим в данный момент в системе:

```
openvswitch_vswitchd# ovs-vsctl list-ports br-int
int-eth1-br
patch-tun
tap2d782834-d1
tap690466bc-92
```

```
tap8a864970-2d
```

4. Туннельный мост `br-tun` содержит интерфейс `patch-int` и интерфейсы `gre-<N>` для каждого однорангового узла, к которому он подключается через GRE, по одному для каждого вычислительного и сетевого узла в кластере:

```
openvswitch_vswitchd# ovs-vsctl list-ports br-tun  
  
patch-int  
  
gre-1  
  
.  
  
.  
  
.  
  
gre-<N>
```

5. Если какая-либо из этих линий отсутствует или неверна, возможно, конфигурация выполнена неверно. Мосты можно добавить с помощью `ovs-vsctl add-br`, а порты можно добавить к мостам с помощью `ovs-vsctl add-port`. Хотя выполнение этих команд вручную может быть полезно при отладке, крайне важно, чтобы ручные изменения, которые вы намерены сохранить, отражались в ваших файлах конфигурации.

4.7 Восстановление диска экземпляра Nova

После сбоя питания или какого-либо другого аварийного сценария некоторые экземпляры Nova могут не запускаться и отображать ошибку диска, которую можно увидеть в журнале консоли:

```
[ 4.696001] EXT4-fs (vda1): INFO: recovery required on readonly filesystem  
[ 4.699208] EXT4-fs (vda1): write access will be enabled during recovery  
[ 5.327541] blk_update_request: I/O error, dev vda, sector 2048  
[ 5.332272] Buffer I/O error on dev vda1, logical block 0, lost async page write  
[ 5.339769] Buffer I/O error on dev vda1, logical block 1, lost async page write  
[ 5.350018] blk_update_request: I/O error, dev vda, sector 3088
```

4.7.1 Автоматическая процедура

Вы можете автоматически восстановить образ RBD из пула RBD `vms`.

Сначала найдите UUID экземпляра Nova, файловые системы которого нужно восстановить.

Затем запустите скрипт `itkf-maintenance-recover-rbd-image.sh`, который сделает все остальное. Отказавший экземпляр Nova должен быть отключен, если он все еще работает. Функции образа RBD должны быть отключены, образ RBD должен быть сопоставлен с системой в `/dev/rbd*`, должны быть найдены дисковые разделы и выполнена проверка файловой системы. После этого сопоставление образа RBD должно быть отменено, а его функции должны быть снова включены.

Чтобы выполнить автоматическое восстановление образа RBD, выполните следующую процедуру:

1. Найдите UUID отказавшего экземпляра Nova. Формат UUID выглядит так: `f6c62784-610b-4b9c-8f5d-a4386e936077`. Есть два способа:
 - На панели управления перейдите в раздел **Instances (Project → Compute → Instances)**, щелкните *Instance Details* и найдите параметр *ID*.

- С помощью командной строки OpenStack выполните:

```
itopenstack <cloud-name> server list
```

и найдите параметр *ID*.

2. Войдите на сервер управления облаком.
3. На сервере управления облаком перейдите в папку `/opt/cloud-cfg/<cloud-name>`:

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

4. Запустите скрипт `itkf-maintenance-recover-rbd-image.sh` с UUID экземпляра Nova, который был найден на предыдущих шагах:

```
itkf-maintenance-recover-rbd-image.sh <cloud_name> <UUID>
```

5. Запустите экземпляр Nova и проверьте в журнале консоли, правильно ли он запустился.

4.7.2 Ручная процедура

Для восстановления образа Serp RBD необходимо выполнить следующие команды:

- `rbd`: управляет образами блочных устройств RADOS.
- `fsck`: проверяет файловую систему, используемую в дистрибутивах Linux.

Ручная процедура выполняется следующим образом:

1. Войдите в любой узел хранения:

```
ssh root@<cloud-name>-node1
```

2. Вставьте модуль ядра Linux с именем `rbd`:

```
modprobe rbd
```



Предупреждение!

Убедитесь, что экземпляр Nova, который нужно восстановить, находится в состоянии *Shut Down*. Даже если он завис на этапе инициализации. Необходимо избежать двойного доступа, это может еще больше повредить образ RBD.

3. Найдите UUID нужного экземпляра. Формат UUID выглядит так: `6c62784-610b-4b9c-8f5d-a4386e936077`. Есть два способа:

- На панели управления перейдите в раздел **Instances (Project → Compute → Instances)**, щелкните *Instance Details* и найдите параметр *ID*.
- С помощью командной строки OpenStack выполните:

```
itopenstack <cloud-name> server list
```

и найдите параметр *ID*.

4. Попробуйте найти UUID с помощью инструмента `rbd`:

```
sudo rbd -p vms list
```

```
sudo rbd -p vms list | grep <UUID>
```

Если получено совпадение, процедура может быть продолжена. Вы увидите, что изображения имеют суффикс `_disk`.

5. Отключите определенные функции, иначе не получится сопоставить диск:

```
sudo rbd feature disable -p vms <UUID>_disk exclusive-lock object-map fast-diff deep-flatten
```

После завершения процедуры можно снова включить функции. Замените *UUID* на свой.

6. Выполните сопоставление образа с системой:

```
sudo rbd map -p vms <UUID>_disk
```

Будет создано блочное устройство в каталоге */dev/rbdX*.

7. Чтобы просмотреть разделы образа диска, выполните команду:

```
sudo fdisk -l /dev/rbdX
```

8. Чтобы запустить процедуру восстановления, используйте команды Linux, а именно:

```
sudo fsck /dev/rbdXp1
```

```
sudo fsck /dev/rbdXp2
```

```
sudo fsck /dev/rbdXp3
```

```
.  
.
.
```

В большинстве случаев раздел, поврежденный во время незапланированного отключения питания, можно исправить, запустив проверку файловой системы. Мы можем проверить это, смонтировав и выведя файлы самого раздела.

```
sudo mount /dev/rbd0p1 /mnt
```

```
ls /mnt
```

```
sudo umount /mnt
```

9. После завершения восстановления отмените сопоставление образа диска и включите функции образа:

```
sudo rbd unmap /dev/rbdX
```

```
sudo rbd feature enable -p vms <UUID>_disk exclusive-lock object-map fast-diff
```

Примечание.



Из-за особенностей *rbd* невозможно повторно включить функцию *deep-flatten*. В этом нет ничего плохого, т.к. с помощью нее *Swift* хранит нули образа в хранилище данных.

10. Запустите экземпляр.

4.8 Восстановление тома экземпляра Nova

После сбоя питания или какого-либо другого аварийного сценария некоторые экземпляры Nova могут не запускаться и отображать ошибку диска, которую можно увидеть в журнале консоли:

```
[ 4.696001] EXT4-fs (vda1): INFO: recovery required on readonly filesystem  
[ 4.699208] EXT4-fs (vda1): write access will be enabled during recovery  
[ 5.327541] blk_update_request: I/O error, dev vda, sector 2048  
[ 5.332272] Buffer I/O error on dev vda1, logical block 0, lost async page write  
[ 5.339769] Buffer I/O error on dev vda1, logical block 1, lost async page write
```

```
[ 5.350018] blk_update_request: I/O error, dev vda, sector 3088
```

4.8.1 Автоматическая процедура

Вы можете автоматически восстановить образ RBD из пула RBD *volumes*.

Сначала найдите UUID тома Cinder, файловые системы которого нужно восстановить.

Затем запустите скрипт *itkf-maintenance-recover-rbd-volume.sh*, который сделает все остальное. Если статус раздела Cinder – *in-use*, неисправный экземпляр Nova с прикрепленным томом Cinder должен быть отключен, если он все еще работает. Функции образа RBD должны быть отключены, образ RBD должен быть сопоставлен с системой в */dev/rbd**, должны быть найдены дисковые разделы и выполнена проверка файловой системы. После этого сопоставление образа RBD должно быть отменено, а его функции должны быть снова включены.

Чтобы выполнить автоматическое восстановление образа RBD, выполните следующую процедуру:

1. Найдите UUID отказавшего тома экземпляра Nova. Формат UUID выглядит так: *f6c62784-610b-4b9c-8f5d-a4386e936077*. Есть два способа:

- На панели управления перейдите в раздел **Instances (Project → Compute → Instances)**, щелкните *Instance Details* и найдите параметр *Attached To*.
- С помощью командной строки выполните:

```
itopystack <cloud-name> server show <Instance Name>
```

и найдите параметр *volumes_attached*.

2. Войдите на сервер управления облаком.
3. На сервере управления облаком перейдите в папку */opt/cloud-cfg/<cloud-name>*:

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

4. Запустите скрипт *itkf-maintenance-recover-rbd-volume.sh* с UUID тома Cinder, который был найден на предыдущих шагах:

```
itkf-maintenance-recover-rbd-volume.sh <cloud_name> <UUID>
```

5. Запустите экземпляр Nova и проверьте в журнале консоли, правильно ли он запустился.

4.8.2 Ручная процедура

Для восстановления образа Ceph RBD необходимо выполнить следующие команды:

- *rbd*: управляет образами блочных устройств RADOS.
- *fsck*: проверяет файловую систему, используемую в дистрибутивах Linux.

Ручная процедура выполняется следующим образом:

1. Войдите в любой узел хранения:

```
ssh root@<cloud-name>-node1
```

2. Вставьте модуль ядра Linux с именем *rbd*:

```
modprobe rbd
```

3. Найдите любой контейнер Docker со службой Ceph OSD:

```
docker ps | grep osd
```

4. Подключитесь к любому контейнеру Docker со службой Ceph OSD:

```
docker exec -ti ceph_osd_X bash
```

**Предупреждение!**

Убедитесь, что экземпляр Nova, который нужно восстановить, находится в состоянии *Shut Down*. Даже если он завис на этапе инициализации. Необходимо избежать двойного доступа, это может еще больше повредить образ RBD.

5. Найдите UUID нужного тома экземпляра. Формат UUID выглядит так: *f6c62784-610b-4b9c-8f5d-a438be936077*. Есть два способа:

- На панели управления перейдите в раздел **Instances (Project → Compute → Instances)**, щелкните *Instance Details* и найдите параметр *Attached To*.
- С помощью командной строки OpenStack выполните:

```
itopenstack <cloud-name> server show <Instance Name>
```

и найдите параметр *volumes_attached*.

6. Попробуйте найти *UUID* с помощью инструмента *rbd*:

```
sudo rbd -p volumes list
```

```
sudo rbd -p volumes list | grep <UUID>
```

Если получено совпадение, процедура может быть продолжена. Вы увидите, что тома имеют суффикс *volume-*.

7. Отключите определенные функции, иначе не получится сопоставить диск:

```
sudo rbd feature disable -p volumes volume-<UUID> exclusive-lock object-map fast-diff deep-flatten
```

После завершения процедуры можно снова включить функции. Замените *UUID* на свой.

8. Выполните сопоставление образа с системой:

```
sudo rbd map -p volumes volume-<UUID>
```

Будет создано блочное устройство в каталоге */dev/rbdX*.

9. Чтобы просмотреть разделы тома диска, выполните команду:

```
sudo fdisk -l /dev/rbdX
```

10. Чтобы запустить процедуру восстановления, используйте команды Linux, а именно:

```
sudo fsck /dev/rbdXp1
```

```
sudo fsck /dev/rbdXp2
```

```
sudo fsck /dev/rbdXp3
```

.
.
.

В большинстве случаев раздел, поврежденный во время незапланированного отключения питания, можно исправить, запустив проверку файловой системы. Мы можем проверить это, смонтировав и выведя файлы самого раздела.

```
sudo mount /dev/rbdXp1 /mnt
```

```
ls /mnt
```

```
sudo umount /mnt
```

- После завершения восстановления отмените сопоставление образа тома и включите функции тома:

```
sudo rbd unmap /dev/rbdX
```

```
sudo rbd feature enable -p volumes <volume-UUID> exclusive-lock object-map fast-diff
```



Примечание.

Из-за особенностей `rbd` невозможно повторно включить функцию `deep-flatten`. В этом нет ничего плохого, т.к. с помощью нее `Swift` хранит нули образа в хранилище данных.

- Запустите экземпляр.

4.9 Восстановление всех образов RBD

После сбоя питания или какого-либо другого аварийного сценария некоторые экземпляры Nova могут не запускаться и отображать ошибку диска, которую можно увидеть в журнале консоли:

```
[ 4.696001] EXT4-fs (vda1): INFO: recovery required on readonly filesystem
[ 4.699208] EXT4-fs (vda1): write access will be enabled during recovery
[ 5.327541] blk_update_request: I/O error, dev vda, sector 2048
[ 5.332272] Buffer I/O error on dev vda1, logical block 0, lost async page write
[ 5.339769] Buffer I/O error on dev vda1, logical block 1, lost async page write
[ 5.350018] blk_update_request: I/O error, dev vda, sector 3088
```

4.9.1 Автоматическая процедура

Вы можете автоматически восстановить все образы RBD из пулов `volumes` и `vms`.

Скрипт `itkf-maintenance-recover-rbd.sh` восстановит все образы RBD. Отказавший экземпляр Nova должен быть отключен, если он все еще работает. Если статус тома Cinder – `in-use`, неисправный экземпляр Nova с прикрепленным томом Cinder должен быть отключен, если он все еще работает. Функции образа RBD должны быть отключены, образ RBD должен быть сопоставлен с системой в `/dev/rbd*`, должны быть найдены дисковые разделы и выполнена проверка файловой системы. Для разделов без файловой системы проверку файловой системы выполнять не нужно. После этого сопоставление образа RBD должно быть отменено, а его функции должны быть снова включены. Если необходимо, все экземпляры Nova могут быть запущены автоматически.

Чтобы выполнить автоматическое восстановление всех образов RBD, выполните следующую процедуру:

- Войдите на сервер управления облаком.
- На сервере управления облаком перейдите в папку `/opt/cloud-cfg/<cloud-name>`:

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

- Запустите скрипт `itkf-maintenance-recover-rbd.sh` с файлом развертывания `yaml`. Если вы хотите автоматически запустить все экземпляры Nova после восстановления всех образов RBD, запустите скрипт `itkf-maintenance-recover-rbd.sh` с параметром `-s`:

```
itkf-maintenance-recover-rbd.sh -s <cloud-name>-big.yaml
```

- Запустите экземпляры Nova вручную, если они не были запущены автоматически (с параметром `-s` на предыдущем шаге).

5. Проверьте в журнале консоли, правильно ли запустились экземпляры Nova.

4.10 Включение платформы после незапланированного отключения платформы

Незапланированное отключение платформы происходит, например, из-за сбоя питания, внезапного отключения питания и т.д.

В нашем примере у нас есть кластер с пятью (5) узлами:

- `<cloud-name>-node1`, `<cloud-name>-node2`, `<cloud-name>-node3`: сервисные узлы (управления, сети, базы данных, вычислений, хранения)
- `<cloud-name>-node4`, `<cloud-name>-node5`: вычислительные узлы (вычислений, хранения)

Чтобы выполнить запуск всего кластера, выполните следующую процедуру:



Предупреждение!

Включите все сервисные узлы в указанном ниже порядке.

Перед включением очередного сервисного узла, подождите, пока предыдущий сервисный узел не включится и не запустит контейнеры Docker.

1. Включите первый сервисный узел. В нашем примере `<cloud-name>-node1`. Через некоторое время войдите в систему (удаленно).
2. Включите второй сервисный узел. В нашем примере `<cloud-name>-node2`. Через некоторое время войдите в систему (удаленно).
3. Проверьте статус RabbitMQ. Кластер RabbitMQ должен быть активен, только узел `<cloud-name>-node3` еще не является его частью:

```
(on <cloud-name>-node1) docker exec rabbitmq rabbitmqctl cluster_status
Cluster status of node 'rabbit@testedge-node1' ...
[{nodes,[{disc,['rabbit@testedge-node1','rabbit@testedge-node2',
                'rabbit@testedge-node3']}]},
 {running_nodes,['rabbit@testedge-node2','rabbit@testedge-node1']},
 {cluster_name,<<"rabbit@testedge-node1.csi.iskrauraltel.ru">>},
 {partitions,[]},
 {alarms,[{'rabbit@testedge-node2',[]},{'rabbit@testedge-node1',[]}]}]
```

4. Включите третий сервисный узел. В нашем примере это `<cloud-name>-node3`.
5. Проверьте статус Ceph. Кластер Ceph должен быть активен, зарегистрировано предупреждение HEALTH_WARN, потому что не все вычислительные узлы еще включены:

```
(on <cloud-name>-node1) ceph -s
cluster:
  id:      69705746-434e-481d-9aa6-4bee57781b38
  health:  HEALTH_WARN
          2 osds down
          6 hosts (2 osds) down
```

```

Reduced data availability: 47 pgs inactive, 69 pgs peering
Degraded data redundancy: 1218/3999 objects degraded (30.458%), 56 pgs degraded
1/3 mons down, quorum 172.30.2.11,172.30.2.12

services:
  mon: 3 daemons, quorum 172.30.2.11,172.30.2.12, 172.30.2.13
  mgr: testedge-node1.csi.iskrauraltel.ru(active), standbys: testedge-node2.csi.iskrauraltel.ru, testedge-node3.csi.iskrauraltel.ru
  osd: 6 osds: 4 up, 6 in

data:
  pools: 8 pools, 160 pgs
  objects: 1.75k objects, 4.41GiB
  usage: 17.1GiB used, 5.43TiB / 5.45TiB avail
  pgs: 11.250% pgs unknown
      72.500% pgs not active
      1218/3999 objects degraded (30.458%)
      69 peering
      39 undersized+degraded+peered
      18 unknown
      17 active+undersized+degraded
      9 active+undersized
      8 undersized+peered

```

6. Проверьте статус RabbitMQ. Кластер RabbitMQ теперь активен, все три (3) узла участвуют:

```

(on <cloud-name>-node1) docker exec rabbitmq rabbitmqctl cluster_status
Cluster status of node 'rabbit@testedge-node1' ...
[{"nodes":[{"disc":["rabbit@testedge-node1","rabbit@testedge-node2",
                  "rabbit@testedge-node3"]}],
  {"running_nodes":["rabbit@testedge-node3","rabbit@testedge-node2",
                  "rabbit@testedge-node1"]}],
  {"cluster_name,<<"rabbit@testedge-node1.csi.iskrauraltel.ru">>},
  {"partitions,""},
  {"alarms,[{"rabbit@testedge-node3",[]},
            {"rabbit@testedge-node2",[]},
            {"rabbit@testedge-node1",[]}]}]

```

7. Если в кластере RabbitMQ по-прежнему участвует менее трех (3) узлов, перезапустите сервисные группы RabbitMQ принудительно:

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f <cloud-name> rabbitmq
```

8. Включите все вычислительные узлы.
9. Проверьте статус Ceph:

```
(on <cloud-name>-node1) ceph -s
```

```
cluster:
  id:      69705746-434e-481d-9aa6-4bee57781b38
  health:  HEALTH_OK

services:
  mon: 3 daemons, quorum 172.30.2.11,172.30.2.12,172.30.2.13
  mgr: testedge-node1.csi.iskrauraltel.ru(active), standbys: testedge-node2.csi.iskrauraltel.ru, testedge-node3.csi.iskrauraltel.ru
  osd: 6 osds: 6 up, 6 in
  rgw: 1 daemon active

data:
  pools:   8 pools, 160 pgs
  objects: 1.75k objects, 4.41GiB
  usage:   17.1GiB used, 5.43TiB / 5.45TiB avail
  pgs:    160 active+clean

io:
  client: 81.8KiB/s rd, 0B/s wr, 84op/s rd, 51op/s wr
```

10. Восстановите кластер базы данных MariaDB Galera:

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v <cloud-name> mariadb
```

11. Получите пароль кластера MariaDB Galera:

```
(on fai-kolla-server) grep -w database_password /opt/etc-kolla/<cloud-name>/passwords.yml
database_password: [пароль]
```

12. Проверьте состояние кластера Galera. Кластер Galera теперь активен, все три (3) узла участвуют:

```
(on <cloud-name>-node1) docker exec mariadb mysql -u root -p[пароль] -e "SHOW STATUS" |
grep wsrep_cluster
wsrep_cluster_conf_id 10
wsrep_cluster_size 3
```

```
wsrep_cluster_state_uuid      74052ada-f4a0-11e9-9152-ff1b943b2e97
```

```
wsrep_cluster_status      Primary
```

13. Если в кластере Galera нет трех (3) узлов, снова восстановите кластер базы данных MariaDB Galera и проверьте его состояние.
14. Проверьте, запущены ли все необходимые службы OpenStack в кластере <cloud-name>.
15. Проверьте на всех узлах, что все контейнеры Docker запустятся должным образом:

```
(on <cloud-name>-node*) docker ps -a --filter "label=kolla_version" --filter status=exited --filter status=restarting --format "{{.Names}}"
```

```
nova_scheduler
```

16. При необходимости перезапустите неработающие сервисные группы OpenStack принудительно. В нашем случае контейнер *nova_scheduler* завершил работу, поэтому все службы Nova необходимо перезапустить в заданном порядке:

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f <cloud-name> nova
```

17. Перед включением экземпляров Nova внутри OpenStack либо через панель управления, либо через интерфейс командной строки, запустите через Serf восстановление экземпляра Nova либо для диска, либо для тома (см. раздел 4.7 и/или 4.8). Вы также можете запустить восстановление Serf всех образов RBD (см. раздел 4.9).
18. Включите экземпляры Nova внутри OpenStack либо через панель управления, либо через интерфейс командной строки.
19. В случае проблем с сетевым подключением между экземплярами Nova и DHCP-серверами Neutron, маршрутизаторами Neutron и/или сервером метаданных Nova необходимо перезапустить некоторые сервисные группы. Во-первых, перезапустите сервисную группу Open vSwitch, а затем сервисную группу Neutron в предопределенном порядке на сервисных узлах, принудительно. Во-вторых, перезапустите сервисную группу Open vSwitch, а затем сервисную группу Neutron в предопределенном порядке на вычислительных узлах, принудительно. В нашем примере следуйте следующему порядку:



Предупреждение!

Подождите несколько минут между выполнением каждой команды.

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node1.<domain-name> <cloud-name> openvswitch
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node2.<domain-name> <cloud-name> openvswitch
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node3.<domain-name> <cloud-name> openvswitch
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node1.<domain-name> <cloud-name> neutron
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node2.<domain-name> <cloud-name> neutron
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node3.<domain-name> <cloud-name> neutron
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node4.<domain-name> <cloud-name> openvswitch
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node5.<
domain-name> <cloud-name> openvswitch
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node4.<
domain-name> <cloud-name> neutron
```

```
(on fai-kolla-server) itkf-maintenance-restart-service.sh -v -f -n <cloud-name>-node5.<
domain-name> <cloud-name> neutron
```

4.11 Сброс облачного узла с полностью занятым пространством подкачки

Когда облачный узел с интенсивно используемой подкачкой перезагружается и/или выключается, для фактического сброса или включения питания требуется полчаса.

В системах, управляемых с помощью `systemd`, перезагрузка и завершение работы системы контролируются централизованно с помощью инструмента `systemctl`:

```
file /sbin/{halt,poweroff,reboot,shutdown}

/sbin/halt:      symbolic link to /bin/systemctl
/sbin/poweroff: symbolic link to /bin/systemctl
/sbin/reboot:   symbolic link to /bin/systemctl
/sbin/shutdown: symbolic link to /bin/systemctl
```

Ввод команды `reboot` эквивалентен `systemctl start reboot.target`, для которого требуется `systemd-reboot.service`, для которого также требуются `shutdown.target`, `umount.target` и `final.target`.

В системе с `/var` и/или `/var/log`, смонтированными отдельно от `/`, перезагрузка не может чисто размонтировать `/var` (или `/var/log`), потому что его использует `systemd-journald`.

В случае `vv3edge-node1`, цель `reboot.target` не достигается почти 30 минут, т.к. подкачку нельзя деактивировать:

```
Dec 11 17:00:01 vv3edgecloud-node1.edge.iskrauraltel.ru systemd[1]: dev-disk-by\x2duuid-5b193649\x2dab95\x2d4d70\x2d976b\x2da152386731c8.swap: Deactivation timed out. Stopping.
```

```
Dec 11 17:00:01 vv3edgecloud-node1.edge.iskrauraltel.ru systemd[1]: dev-disk-by\x2duuid-5b193649\x2dab95\x2d4d70\x2d976b\x2da152386731c8.swap: Swap process exited, code=killed status=15
```

```
Dec 11 17:00:01 vv3edgecloud-node1.edge.iskrauraltel.ru systemd[1]: Failed deactivating swap /dev/disk/by-uuid/5b193649-ab95-4d70-976b-a152386731c8
```

```
Dec 11 17:28:21 vv3edgecloud-node1.edge.iskrauraltel.ru systemd[1]: 3: Job reboot.target/start timed out.
```

```
Dec 11 17:28:21 vv3edgecloud-node1.edge.iskrauraltel.ru systemd[1]: Timed out starting Reboot.
```

```
Dec 11 17:28:21 vv3edgecloud-node1.edge.iskrauraltel.ru systemd[1]: reboot.target: Job reboot.target/start failed with result 'timeout'.
```

```
Dec 11 17:28:21 vv3edgecloud-node1.edge.iskrauraltel.ru systemd[1]: Forcibly rebooting: job timed out
```

```
Dec 11 17:28:21 vv3edgecloud-node1.edge.iskrauraltel.ru systemd[1]: Shutting down.
```

Самым чистым способом перезагрузки системы будет `sudo swapoff -a && systemctl reboot`. Процесс подкачки занимает еще 10+ минут, после чего происходит выключение без зависаний.



Предупреждение!

Объем данных, хранящихся в подкачке, должен быть меньше объема, доступного в оперативной памяти, иначе команда не будет выполнена! Это может быть проблематично для систем с 32 ГБ, 64 ГБ ОЗУ, таких как `vv3edgecloud-node1`.

Если система зависла, нажмите **Ctrl+Alt+F9**, чтобы переключиться на VT9, и выполните `systemctl --force --force reboot` (именно дважды `--force`), что немедленно перезагрузит систему.

Если даже это не поможет, вы можете использовать iLO и параметр *Press Power Button*.

Наконец, так же можно нажать физическую кнопку сброса на сервере.

————— (КОНЕЦ ДОКУМЕНТА) —————