
Acloud Documentation

1.1.2

Acloud

янв. 23, 2025

1	Руководство по установке	1
1.1	Введение	1
1.2	Требования к оборудованию и ПО	2
1.3	Предварительная подготовка и основные концепции	5
1.4	Установка узла развёртывания	9
1.5	Подготовка к развёртываниям с нуля	25
1.6	Подготовка одного развёртывания	33
1.7	Развёртывание облака	45
1.8	Полезные советы	51
1.9	Дополнительно	60
2	Пользовательская документация	62
2.1	Интерфейс консоли	62
2.2	Страница пользователя	64
2.3	Режим администратора	65
2.4	Мичман	79
2.5	Биллинг	115
3	Руководство по администрированию	116
3.1	Руководство по сопровождению	116

3.2	Руководство по устранению неполадок	120
4	Руководство по тестированию	129
4.1	Программа и методика испытаний	129
5	Коммерческая информация	131
5.1	Информация о стоимости лицензий	131
5.2	AccentOS RemoteApp	131
6	Глоссарий	133
7	Release notes	155
7.1	Версия 1.0	155
7.2	Версия 1.1	156
	Алфавитный указатель	158

1.1 Введение

Acloud - это дистрибутив для развёртывания облачной среды на базе OpenStack.

1.1.1 Основные термины и компоненты

Облачная среда - это экземпляр установленной системы на базе кода OpenStack. Это сложная система, состоящая из множества компонентов, поэтому при каждом действии важно понимать, какой точно будет результат.

Облачный дистрибутив - это заранее подготовленный набор сервисов и всех необходимых для установки OpenStack пакетов. **Облачный дистрибутив Acloud** распространяется в виде диска, содержащего множество программных пакетов, используемых для установки облачной платформы, сценарии для систем оркестрации, а также пакеты и сценарии развёртывания **Michman** – инструмента оркестрации сервисов. Набор пакетов для установки программ является полным и не требует подключения к сети Интернет.

Программный комплекс на базе дистрибутива состоит из следующих компонентов:

- **Сервер/узел развертывания:** устанавливается на сервер, выполняющий роль развертывания и сопровождения всех остальных компонентов. Развертывается ровно на одном сервере с электронного носителя с дистрибутивом;
- **Сервер/узел управления/контроллер:** является частью развёртываемого **облака**, содержит все сервисы управления облаком;
- **Сервер/узел виртуализации:** является частью развёртываемого **облака**, выполняет функции исполнения ВМ;
- **Сервер/узел хранения данных:** опциональный сервер для внешнего хранения данных ВМ, выполняет функции распределенной отказоустойчивой СХД. При установке из дистрибутива на базе **Ceph** требует для установки не менее пяти серверов, однако может не использоваться при наличии альтернативной СХД.

Для работы с дистрибутивом необходим заранее подготовленный **узел развёртывания**. Для подготовки узла перейдите к пункту «*Установка узла развёртывания*».

Подготовка полноценного комплекса состоит из следующих этапов:

1. Установка узла развёртывания.
2. Подготовка и установка системы хранения данных на базе **Ceph**.
3. Подготовка и развёртывание облака на базе **OpenStack**.

Развёртывание облачной среды на базе дистрибутива **Acloud** производится с использованием утилиты **DeckCrew TUI**.

DeckCrew TUI - текстовый интерфейс пользователя для развёртывания дистрибутива **Acloud**. Это инструмент для упрощения развёртывания облачной среды на базе **Acloud** - дистрибутива **OpenStack**

1.2 Требования к оборудованию и ПО

1.2.1 Узел развёртывания

Рабочие требования:

- *Процессор* - 8 ядер x86_64
- *Память* - 64 ГБ
- *Диск* - 150 ГБ
- *Сетевая карта* - 2 порта по 10 Гбит/с

Минимальные требования:

- *Память* - 8 ГБ

- *Диск* - 100 ГБ
 - `/var/lib/docker` - 10 ГБ
 - `/var/lib/config-data` - 3 ГБ
 - `/var/log` - 3 ГБ
 - `/usr` - 5 ГБ
 - `/var` - 20 ГБ
 - `/` - 25 ГБ
- *Сетевая карта* - 2 порта

1.2.2 Узлы управления OpenStack

Рабочие требования:

- *Количество* - 3 сервера
- *Процессор* - 32 ядер x86_64
- *Память* - 192 ГБ
- *Диск* - 512 ГБ, SSD
- *Сетевая карта* - 2 порта по 10 Гбит/с
- *VMC* - выделенный интерфейс IPMI, iDRAC или iLO (версии не ниже 5)
- *UEFI* - поддержка загрузки с UEFI

1.2.3 Вычислительные узлы OpenStack

Рабочие требования:

- Возможность аппаратной виртуализации
- *Количество* - 2 сервера и больше
- *Процессор* - 16 ядер x86_64
- *Память* - 128 ГБ

- *Диск* - 512 ГБ, SSD
- *Сетевая карта* - 2 порта по 10 Гбит/с
- *ВМС* - выделенный интерфейс IPMI, iDRAC или iLO (версии не ниже 5)
- *UEFI* - поддержка загрузки с UEFI

1.2.4 Хранилище Серв

Рабочие требования:

- *Количество* - 5 серверов
- *Процессор* - 24 ядра x86_64
- *Память* - 128 ГБ
- *Диск* - 2x250 ГБ (под систему) + 12 дисков NVMe
- *Сетевая карта* - 2 порта по 10 Гбит/с
- *UEFI* - поддержка загрузки с UEFI

1.2.5 Сетевое оборудование

Рабочие требования:

- *Коммутатор 1* - 1 шт.
 - Портов по числу серверов, 10Гбит/с (или выше)
 - Поддержка 802.1q
- *Коммутатор 2* - 1 шт.
 - Портов по числу серверов + 1, 1Гбит/с
 - Поддержка 802.1q

1.3 Предварительная подготовка и основные концепции

Дистрибутив построен на базе проекта TripleO (аббревиатура «OpenStack-On-OpenStack»), который позволяет автоматизировать процесс развертывания облачной платформы на базе OpenStack Bobcat.

TripleO использует два главных концепта – Undercloud и Overcloud. *Undercloud* (узел развертывания) устанавливает и настраивает *Overcloud* (рабочее облако или Сeph).

Перед продумыванием архитектуры развертываемой системы необходимо прочитать Требования к оборудованию и ПО.

1.3.1 Узел развертывания

Сетевая топология. Узел развертывания обязан иметь два интерфейса - один для доступа к узлу по ssh разработчиками и один для основной **сети администрирования/Ctlplane**.

Для сети администрирования необходимо знать следующее:

- **CIDR сети** - по умолчанию 192.168.24.0/24
- **3 IP-адреса** из этой сети для узла развертывания - по умолчанию 192.168.24.1-.3
- Два пула адресов из этой сети для двух различных DHCP серверов, которые будут подняты на узле развертывания. Это означает следующее
 - В этой сети не должно существовать **DHCP серверов**
 - Первый пул адресов **Ctlplane DHCP** должен содержать столько IP-адресов, сколько будет возможный максимум всех физических узлов, используемых в развертывании, и ещё два. То есть если в системе предполагает 3 контроллера, 50 узлов виртуализации и 50 узлов хранения данных - то в пуле должно быть не менее 105 IP-адресов.
 - Второй пул адресов **Introspection DHCP** должен содержать столько IP-адресов, сколько будет возможный максимум одновременно интроспектируемых узлов - это может быть небольшое число и как правило 10 адресов достаточно.
 - Второй пул адресов **Introspection DHCP** не должен пересекаться с первым пулом
- Все физические узлы должны иметь хотя бы один интерфейс с **L2 доступом** к сети администрирования для загрузки по PXE.
- Сеть администрирования должна присутствовать **только внутри организации** и не должна быть доступна извне, так как по ней осуществляется **доступ по SSH** к физическим узлам, кроме узла развертывания.

Дополнительные условия для узла развертывания:

- С узла развертывания должны быть доступны используемые в развертывании **DNS сервера, NTP сервера и BMC адреса** для всех физических узлов. Также это значит, что заранее должны быть продуманы DNS и NTP сервера организации.

- С узла развёртывания должна быть доступна **Public API** сеть развёртываемого облака для настройки стартовых admin домена, проекта и пользователя. Эта сеть называется **External** в разделе Сети

Пользователь. Все действия на узле развёртывания необходимо выполнять от имени пользователя **stack**.

После установки узла развёртывания можно запустить **dtui**, чтобы убедиться, что узел установлен правильно. Если узел развёртывания не установлен правильно, то dtui не запустится.

Логи установки можно посмотреть в `/home/stack/install-undercloud.log` или `/var/log/undercloud.log`

Принципы работы

На узле развёртывания поднимается три сервиса OpenStack, которые затем используются для настроек, управления и хранения данных о создаваемых ресурсах.

- **Neutron** - используется для хранения данных о сетях и портах физических узлов.
- **Ironic** - используется для хранения данных о физических узлах и их состоянии, а также используется для развёртывания операционной системы на узлах через PXE.
- **Ironic Inspector** - используется для сбора и хранения данных о физических узлах. Участвует только при выполнении интроспекции и запускается только на целевом узле, который проходит процесс интроспекции.

Heat - поднимается как сервис только в момент развёртывания и сохраняет все свои данные в `/home/stack/overcloud-deploy`. Для тестовых целей может быть поднят вручную командой

```
openstack tripleo launch heat
```

и остановлен

```
openstack tripleo launch heat --kill
```

При первоначальной установке в Neutron заводится сеть Ctlplane, которая далее используется для всех развёртываний. Поменять её уже будет нельзя, поэтому **будьте внимательны при настройке сети администрирования.**

1.3.2 DeckCrew TUI

Внимательно: DeckCrew TUI не предполагает быстрого старта, поэтому необходимо внимательно изучить *Руководство по установке* и пройти каждый пункт инструкции!

Основные данные заведённые в DeckCrew TUI сохраняются на узле развёртывания в `/etc/deckcrew`.

Большинство действий настроек не применяются сразу, а сохраняются в файлы конфигурации. Настройки применяются только в разделе *Deployment*. Далее подробно расписываются все действия в DeckCrew TUI.

Физические узлы

Все физические узлы должны быть переведены в режим **загрузки с сети (PXE)** по одному сетевому интерфейсу в сети Ctlplane. MAC-адрес этого интерфейса должен быть использован при создании записи о физическом узле в `dctui`.

Убедитесь что физические узлы переведены в режим **загрузки с сети (PXE)** в режиме **UEFI**.

Если использовать заранее подготовленные узлы, то пропустите этапы настройки физических машин и разверните узел самостоятельно.

Для заранее подготовленных узлов необходимо знать следующее:

- Требуется пользователь **tripleo-admin** с sudo правами без пароля на подготовленном узле.
- Требуется настроенное подключение к сети администрирования **Ctlplane**.
- Требуется ssh ключ пользователя **stack** с узла развёртывания в файле `~/.ssh/authorized_keys` на подготовленном узле для пользователя **tripleo-admin**.

При заведении записей о физических узлах в `dctui` никакие действия с узлами в этот момент не выполняются, производятся только вызовы API к **Ironic** для сохранения данных об этих узлах.

При развёртывании узлов при помощи **Ironic** на узлах запускается сервис `cloud-init`, который настраивает пользователя `tripleo-admin`.

После этого можно зайти на узел без пароля по ssh, например:

```
[stack@undercloud ~]$ ssh tripleo-admin@192.168.24.100
```

Интроспекция

При проведении интроспекции узел перезагружается и ожидает загрузки по сети (PXE). После интроспекции узел переводится в состояние *available*.

Если интроспекция не планируется, то переведите узел в состояние *available* вручную выбрав действие *provide*.

Шаблоны

Развёртывание облака происходит при помощи сервиса **Heat**. Для этого используются шаблоны развёртывания, которые хранятся в `/etc/deckcrew/templates`.

При создании *Deployment* в главном меню DeckCrew TUI, шаблоны развёртывания копируются из `/var/lib/acloud-heat-templates` в `/etc/deckcrew/templates/-heat-templates`.

Далее все настройки в меню развёртывания применяются к шаблонам.

Сети и сетевые планы

В системе Acloud используется принцип разделения сетей на роли. Набор заранее заданных сетей, который обязан присутствовать в системе:

- *Ctlplane* - сеть администрирования физических узлов в развёртываемом облаке. Это единственная сеть, по которой разрешён SSH доступ к физическим узлам облака.
- *External* - сеть для доступа к публичному API развёртываемого облака. Сеть должна быть доступна для всех пользователей облака.
- *InternalApi* - сеть для доступа к приватному API развёртываемого облака. Используется сервисами облака для общения друг с другом. Не должна быть доступна даже с узла развёртывания.
- *Storage* - сеть для доступа сервисов облака к хранилищу: Ceph или Nfs.

Набор заранее заданных опциональных сетей, их можно удалить из системы за ненадобностью:

- *InternalCeph* - внутренняя сеть для узлов хранения Ceph.
- *IronicProvisioning* - сеть для захвата физических узлов сервисом Ironic развёртываемого облака. Если сервис не используется, то сеть не нужна.

Все сети и IP-адреса, назначаемые физическим узлам при развёртываниях, заводятся на узле развёртывания через **Neutron**.

Таким образом все сетевые настройки включают в себя несколько этапов:

1. Заведение информации о сетях в интерфейсе - далее они сохраняются в `/etc/deckcrew/network_data.yaml`
2. Определение порядка интерфейсов для физических узлов - далее они сохраняются в `/etc/deckcrew/net-config-data-lookup.yaml`
3. Выбор/создание сетевых планов для ролей/узлов - далее они сохраняются в шаблонах развёртывания в `overcloud-baremetal-deploy.yaml`
4. Применение сетевых настроек в разделе *Deployment*.

Все эти пункты взаимосвязаны следующим образом:

- Из настроенных сетей при помощи Neutron автоматически создаются порты для тех узлов, которые настроены на использование этой сети.

- Сетевые планы используют созданные порты и настраивают интерфейсы на физических узлах с указанием IP-адреса созданного порта и других настроек сетей (Vlan, Cidr и прочее).
- IP-адреса создаваемых портов можно указывать вручную в *Roles/Nodes and Networks* в поле *Fixed IPs*.
- Сетевые планы написаны таким образом, что интерфейсы в них указаны по порядку как *nic1, nic2*. То есть если один сетевой план используется для двух разных узлов, где есть интерфейсы *ens1f0,ens1f1* и *eno1,eno2* - то при настройке сети будет выполнено соответствие *nic1=ens1f0/eno1* и *nic2=ens1f1/eno2*.
- Порядок интерфейсов можно менять через определение порядка интерфейсов *Order NICs* в главном меню.

Непосредственно применение всех настроек происходит в разделе *Deployment* в меню развёртывания.

- **Prepare Networks** - создаются сети в Neutron с применением настроек в */etc/deckcrew/network_data.yaml*.
- **Provision Nodes** - выделяются порты в Neutron для узлов и на узлах применяется Ansible роль *tripleo_network_config*, которая использует все сетевые настройки для настройки сети на узле.

Виртуальные адреса и Endpoint Data

Виртуальные адреса используются для доступа к сервисам OpenStack. Они также заводятся как порты через Neutron на андерклауде и при развёртывании облака создаются как ресурсы в Rasemaker. Далее Rasemaker управляет расположением виртуальных адресов, навешивая их на подходящий интерфейс.

Endpoint Data используется для настройки доступа к сервисам OpenStack. Но не используется в шаблонах при развёртывании. Для применения настроек необходимо выполнить генерацию Endpoint Map, который далее используется всеми сервисами OpenStack.

Также EndpointMap используется для создания записей в Keystone о доступе к сервисам OpenStack.

1.4 Установка узла развёртывания

Перед установкой узла развёртывания обязательно проверьте, что вы удовлетворяете требованиям в разделе *Требования к оборудованию и ПО*.

Дистрибутив распространяется на переносимом цифровом носителе, на котором располагается файл *undercloud.iso* или установщик для образа узла развёртывания.

- При установке узла на физический узел - вставьте флешку с установщиком в компьютер и загрузитесь в неё с физического узла.
- При установке на физический узел через ВМС по сети Интернет - примонтируйте файл *undercloud.iso* через виртуальную консоль ВМС и загрузитесь в Virtual CD узла.
- При установке узла развёртывания в виртуальной машине следуйте инструкциям из приложения

Далее узел развёртывания иногда может упоминаться как **Undercloud** - это аналогичное название на английском языке.

1.4.1 Процесс установки

При старте ВМ запустите псевдографический инсталлятор андерклауда. `~[STRIKEOUT:shell sudo undercloud-firstboot]~`

- Если используется установка с флешки, то подключитесь к экрану физического сервера через консольный порт или подключённый диспей.
- Если используется установка через ВМС, то подключитесь к виртуальной консоли ВМС.
- Если используется установка в виртуальной машине, то подключитесь к виртуальной консоли виртуализации. Инсталлятор андерклауда не пробрасывается в серийную консоль, поэтому используйте для подключения `virt-manager`:
 - убедитесь, что на вашей локальной машине установлен X-сервер: в линуксах с любой графической оболочкой он есть по умолчанию, для Windows используйте `VcXsrv`, для MacOS — `XQuartz`;
 - подключитесь к хосту виртуализации по `ssh` с параметром `-X` для проброса X11;
 - запустите `virt-manager` и откройте консоль виртуальной машины.

На экране сервера вы увидите выбор диска для установки операционной системы.



Рис. 1: Undercloud

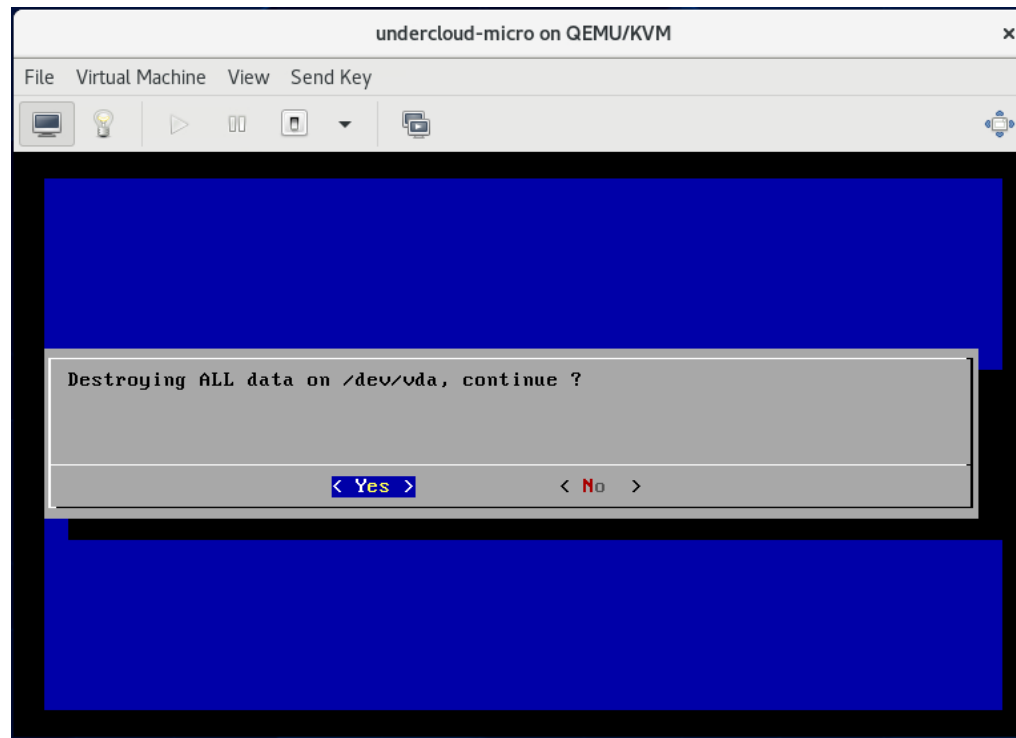


Рис. 2: Undercloud

Дождитесь загрузки и проверки *undercloud.raw*.

Затем выберите интерфейс для сети администрирования физических узлов. Интерфейс не должен использоваться для доступа к узлу развёртывания!

На картинке вы видите два интерфейса. Предварительно на них был запущен автоматически DHCP клиент. Первый интерфейс используется как сеть администрирования, поэтому в сети не было DHCP сервера. Второй интерфейс используется для доступа к узлу развёртывания. В этой сети есть DHCP сервер, который выдал адрес узлу развёртывания.

Важно! Запомните второй адрес для дальнейшего подключения к узлу.

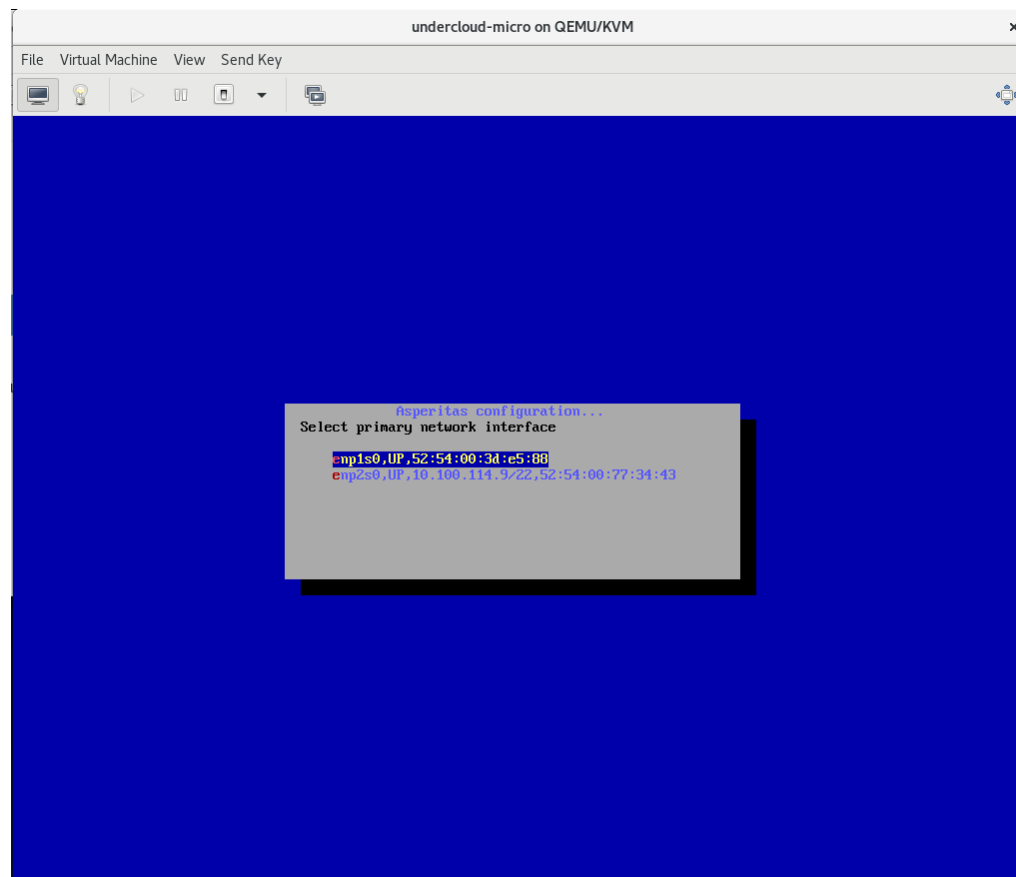


Рис. 3: Undercloud

Затем введите CIDR для сети администрирования.

Важно! После установки сеть будет поменять очень сложно. Поэтому заранее продумайте сети вашей системы!

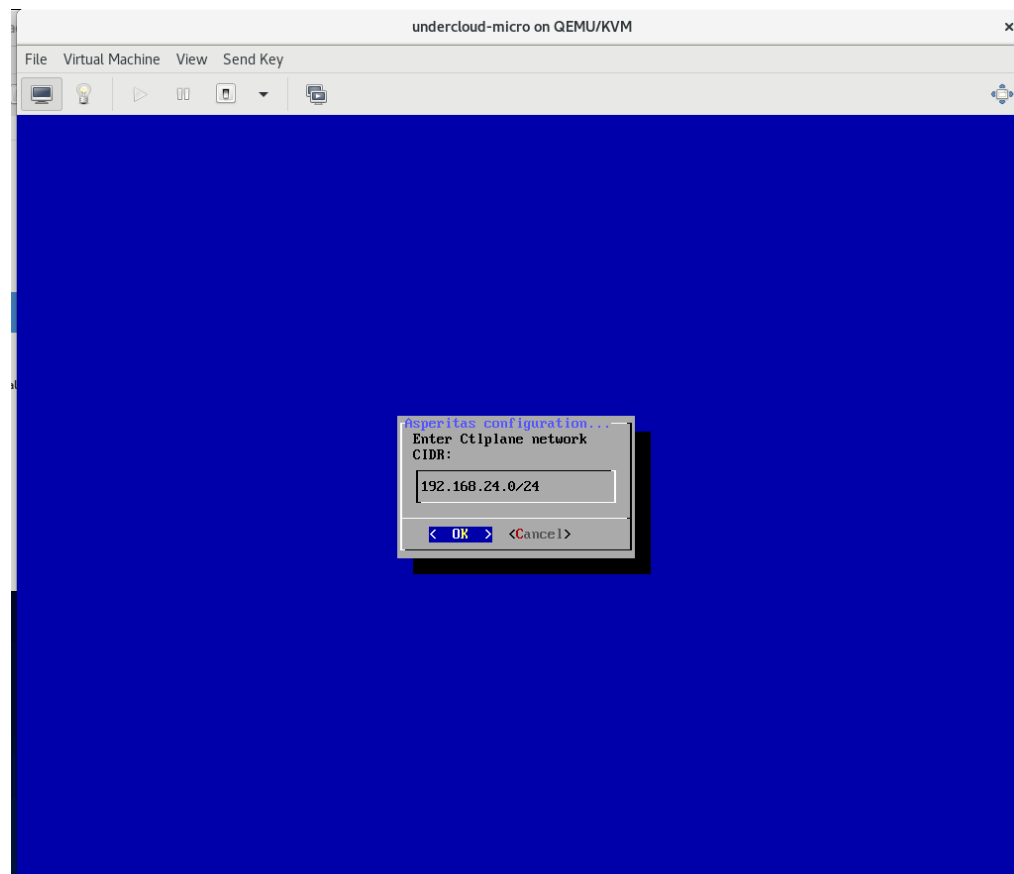


Рис. 4: Undercloud

Введите локальный (основной) IP-адрес узла развёртывания. Узел развёртывания имеет 3(!) адреса так как сам является отдельной облачной средой на базе OpenStack. 3 адреса необходимы для 3 точек доступа к облачным сервисам узла развёртывания: публичной, внутренней точки доступа и точки администратора. Локальный IP-адреса считается как внутренняя точка доступа в системе.

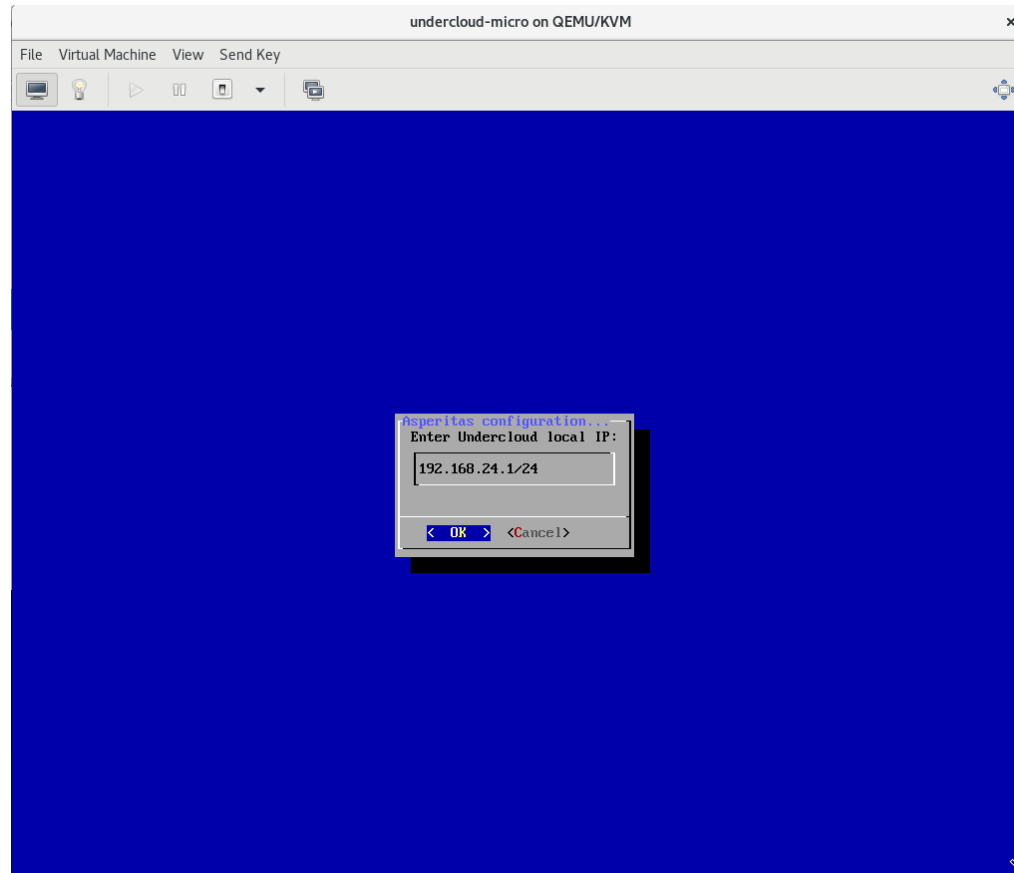


Рис. 5: Undercloud

Введите IP-адрес шлюза для сети администрирования. По умолчанию - это локальный IP-адрес узла развёртывания. Но можно указать и другой адрес для вашей сети администрирования.

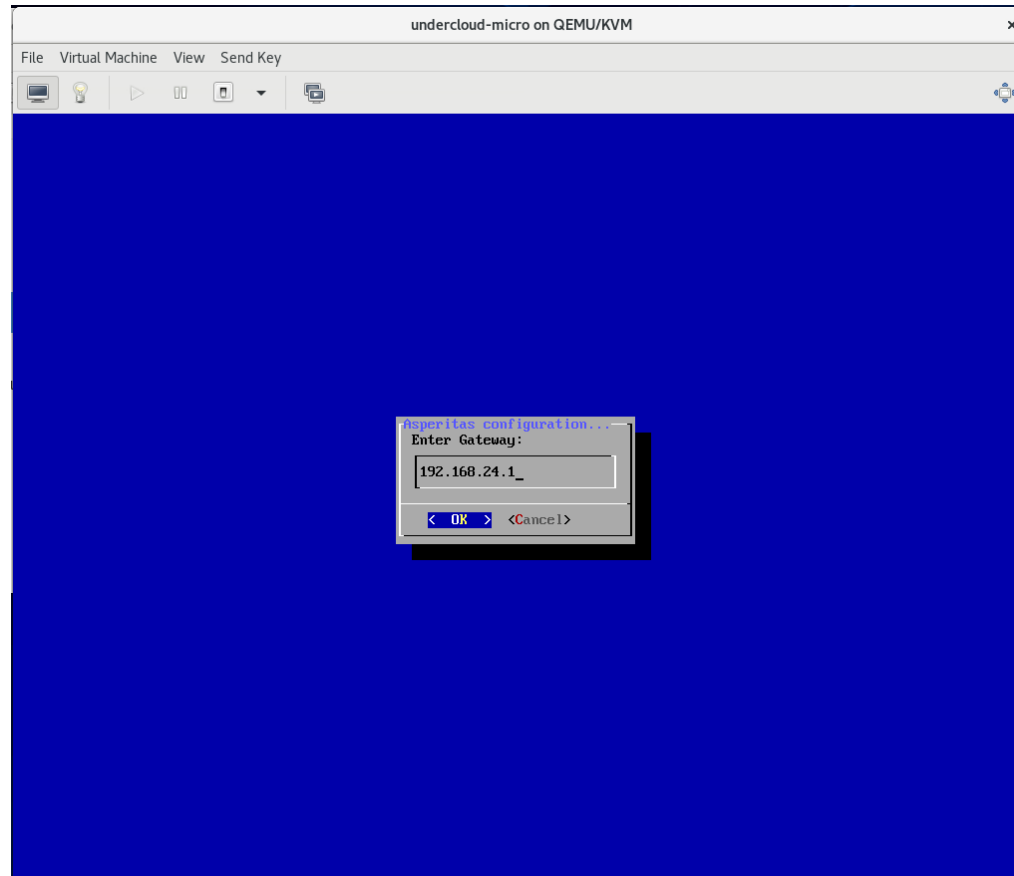


Рис. 6: Undercloud

В сети администрирования узел развёртывания запускает DHCP сервер для раздачи адресов физическим узлам облака. Это обязательно, поэтому в этой сети не может быть внешнего другого DHCP сервера во избежание проблем. Введите начало и конец множества IP-адресов, которые будут назначены вашим физическим машинам. Учтите, что в этой сети заранее заняты 4 адреса для узла развёртывания: 3 указываются вручную, 1 выбирается случайным образом (как правило .5 адрес). А также дальше необходимо будет указать ещё одно множество для адресов интроспектирования физических узлов.

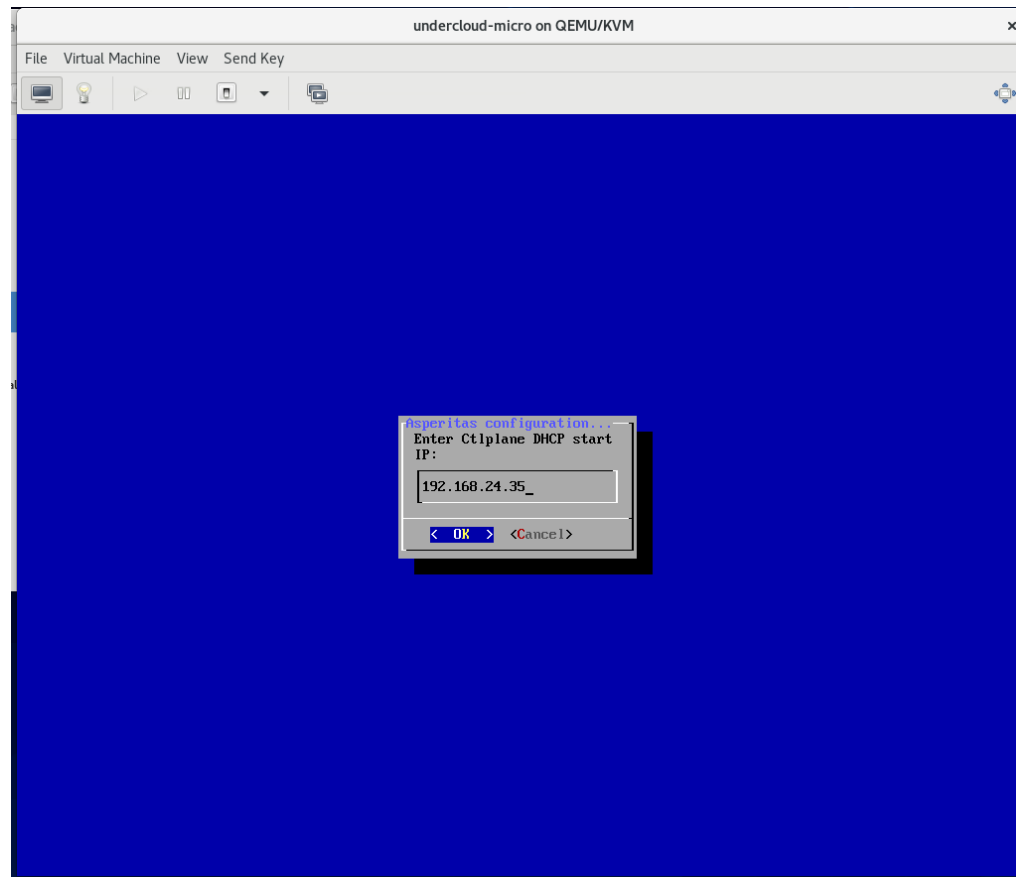


Рис. 7: Undercloud

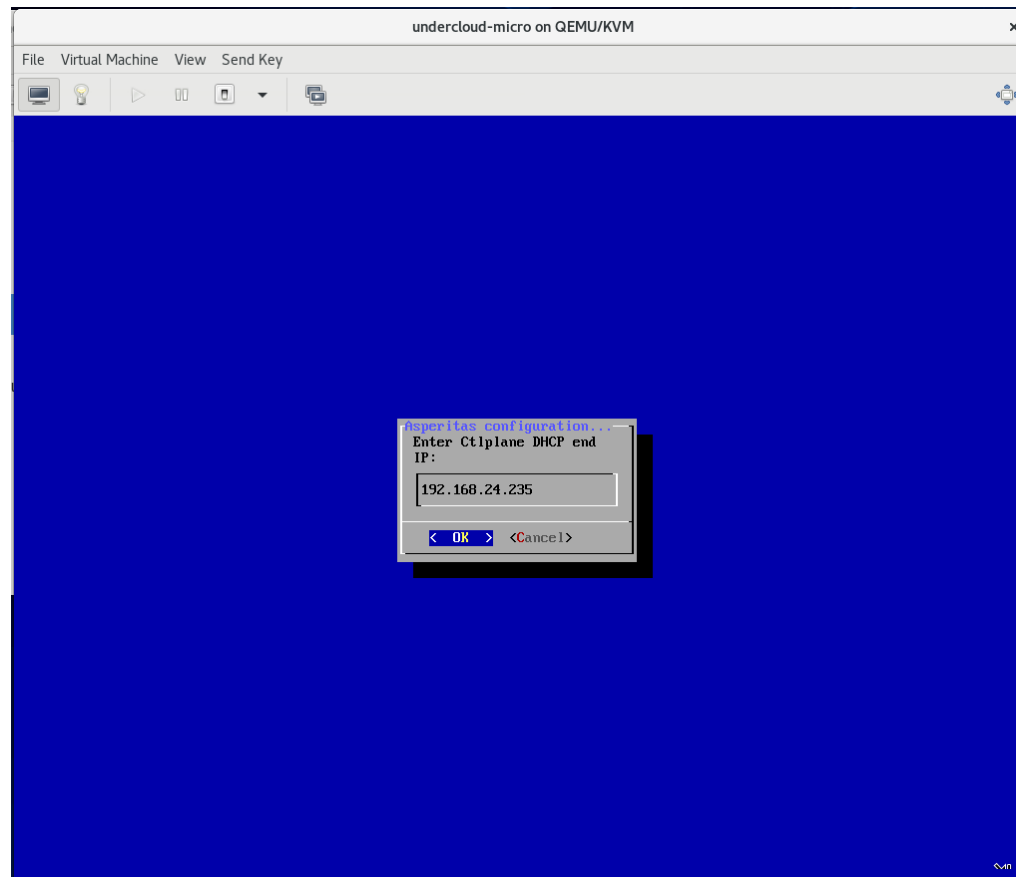


Рис. 8: Undercloud

В сети администрирования узел развёртывания запускает ещё один DHCP сервер, который используется для интроспекции физических узлов. Множество указанных вами IP-адресов влияет только на количество одновременно интроспектируемых узлов. Эти адреса назначаются узлам временно и не сохраняются в дальнейшем.

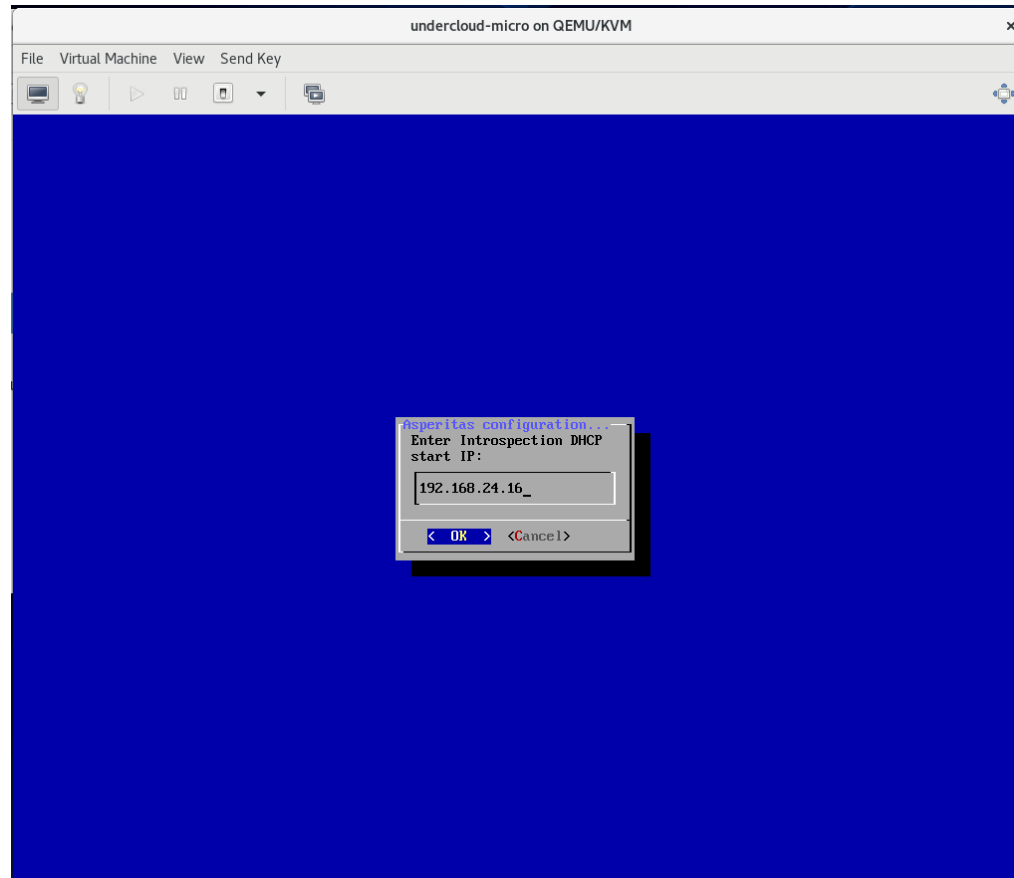


Рис. 9: Undercloud

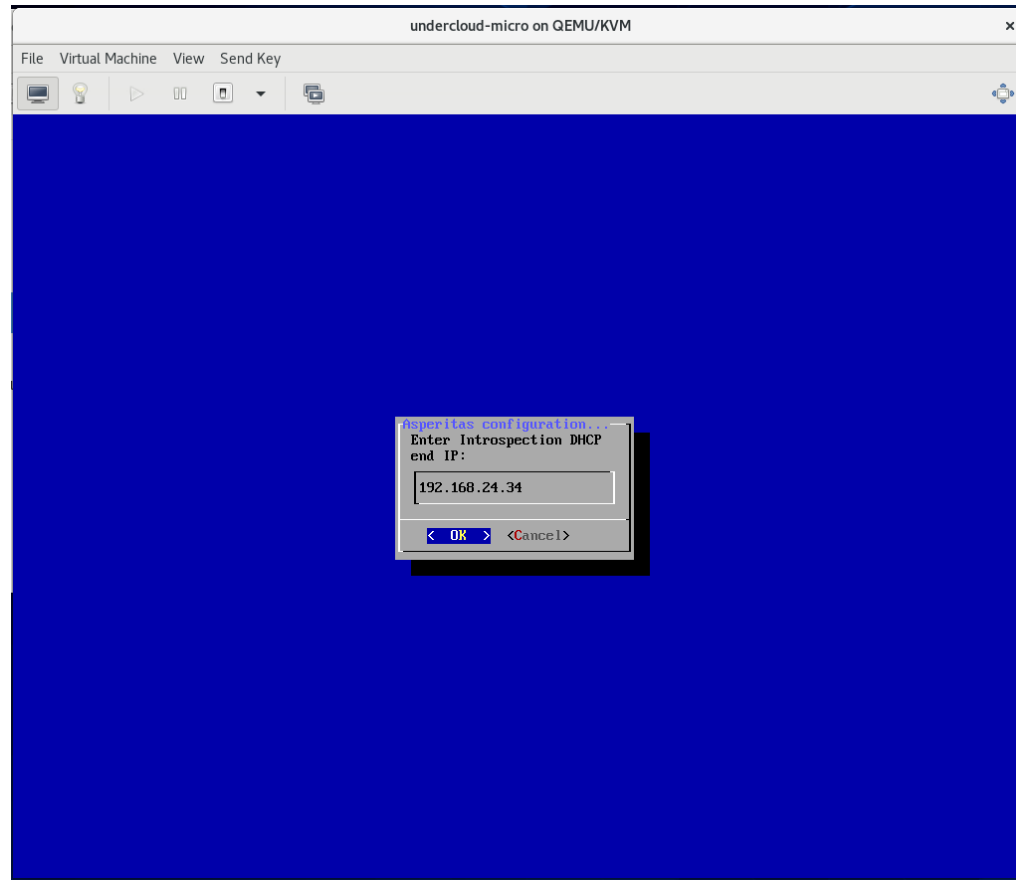


Рис. 10: Undercloud

Введите публичный IP-адрес для узла развёртывания. Используется для доступа к API облачных сервисов на узле развёртывания.

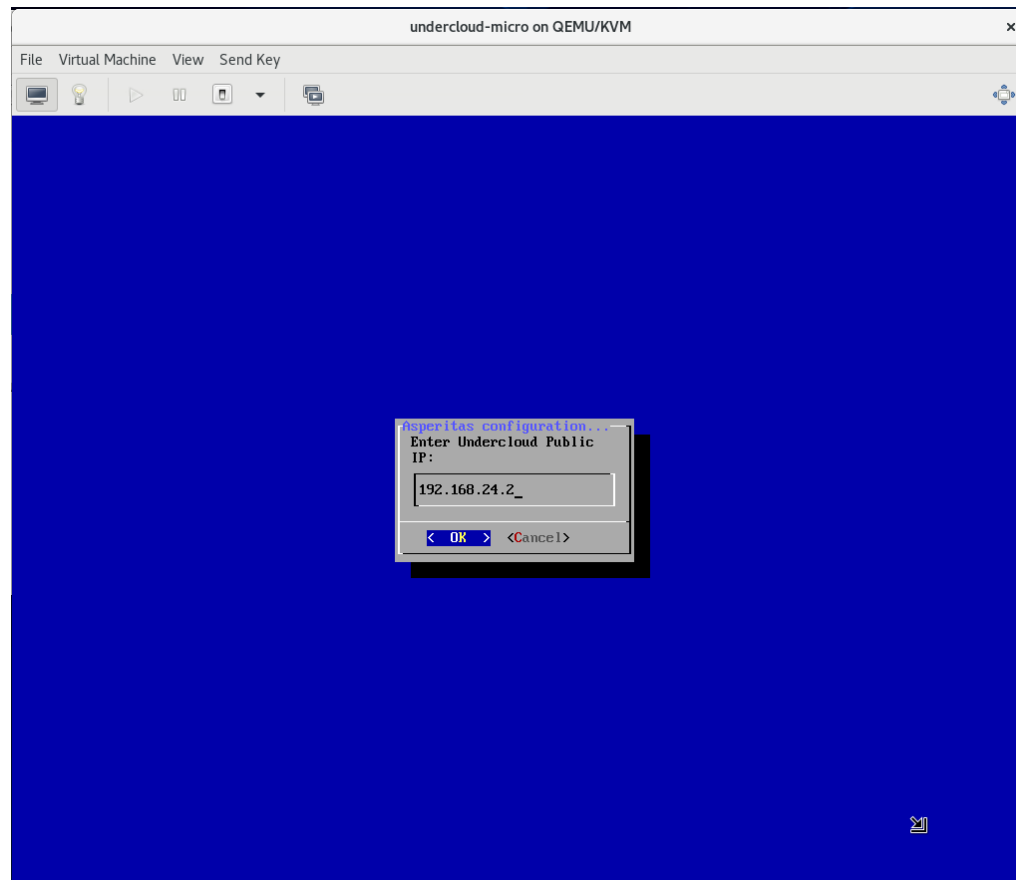


Рис. 11: Undercloud

Введите IP-адрес для точки доступа администратора на узле развёртывания.

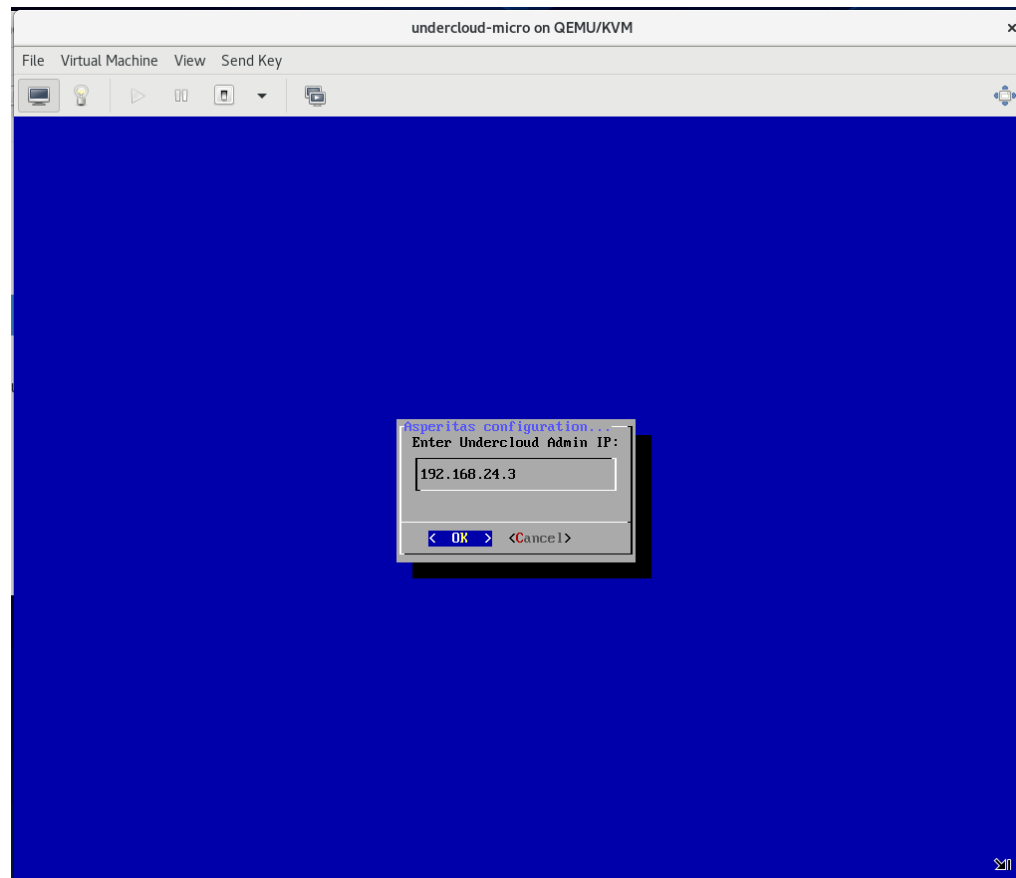


Рис. 12: Undercloud

Введите DNS сервер(а) для узла развёртывания. Если несколько, то перечислите их через запятую.

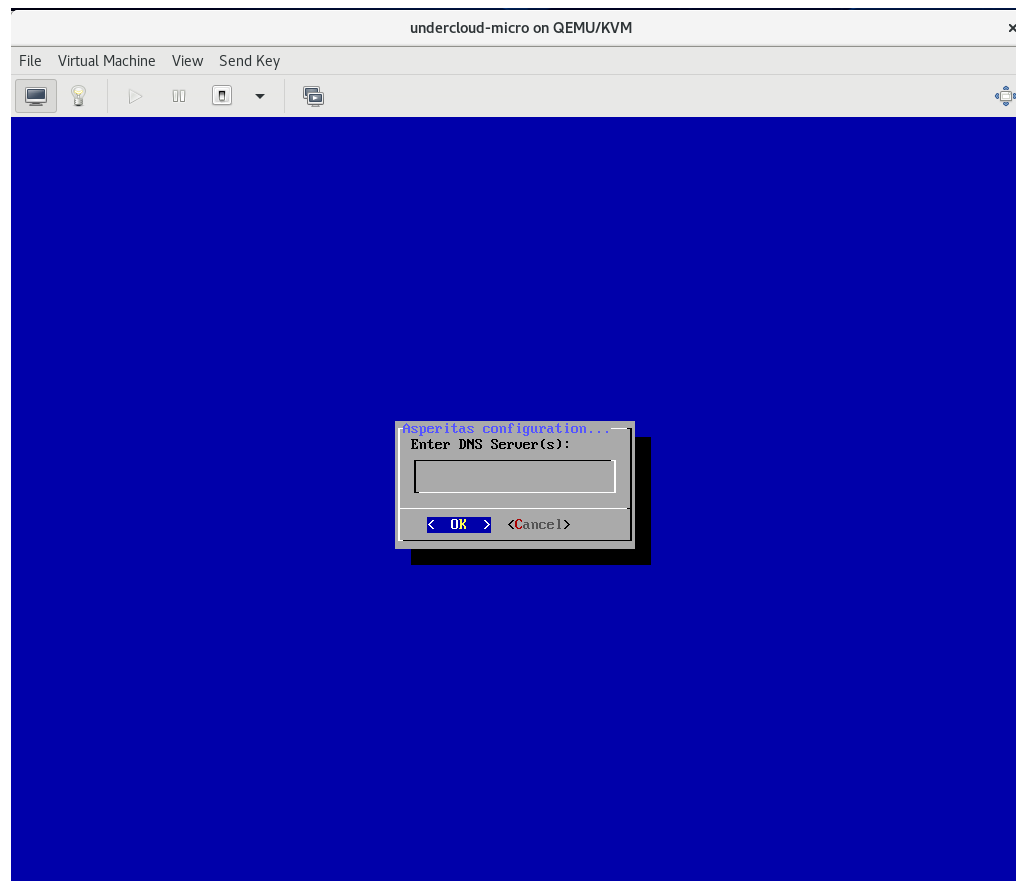


Рис. 13: Undercloud

Введите NTP сервер(а) для узла развёртывания. Если несколько, то перечислите их через запятую.

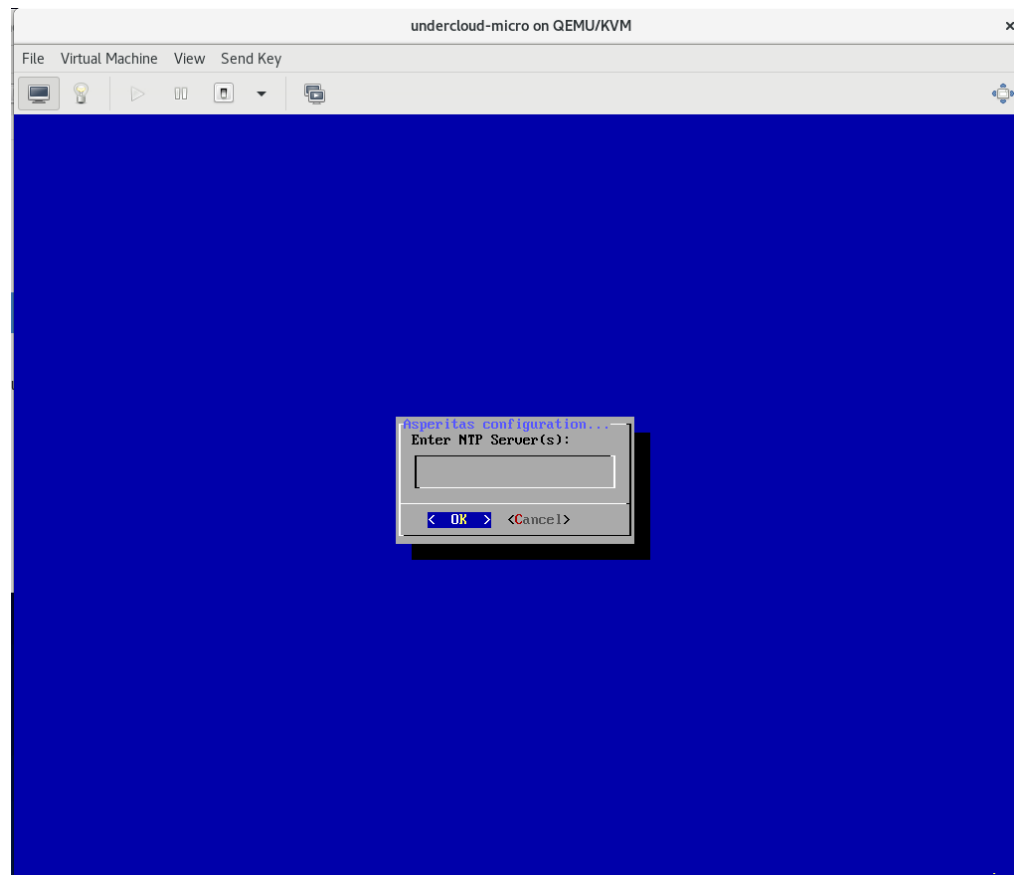


Рис. 14: Undercloud

Дождитесь установки облачной системы на базе OpenStack на узел развёртывания.

```

undercloud-micro on QEMU/KVM
File Virtual Machine View Send Key
2023-02-28 18:30:17.374013 | 52540077-3443-6efd-d781-000000000032 | TIMING | undercloud_disk_space : Initialize existing_vo
umes to an empty array | undercloud | 0:00:05.922512 | 0.11s
Next task: undercloud_disk_space : Filter out non-existing volumes. Play: undercloud
|.....| 6 / 10, 60.00%
2023-02-28 18:30:17.613002 | 52540077-3443-6efd-d781-000000000033 | TIMING | undercloud_disk_space : Filter out non-existing
volumes | undercloud | 0:00:06.161500 | 0.21s
2023-02-28 18:30:17.617371 | 52540077-3443-6efd-d781-000000000033 | TIMING | undercloud_disk_space : Filter out non-existing
volumes | undercloud | 0:00:06.165893 | 0.21s
2023-02-28 18:30:17.621822 | 52540077-3443-6efd-d781-000000000033 | TIMING | undercloud_disk_space : Filter out non-existing
volumes | undercloud | 0:00:06.170341 | 0.22s
2023-02-28 18:30:17.626827 | 52540077-3443-6efd-d781-000000000033 | TIMING | undercloud_disk_space : Filter out non-existing
volumes | undercloud | 0:00:06.175335 | 0.22s
2023-02-28 18:30:17.643384 | 52540077-3443-6efd-d781-000000000033 | TIMING | undercloud_disk_space : Filter out non-existing
volumes | undercloud | 0:00:06.191891 | 0.24s
Next task: undercloud_disk_space : Loop on volumes and gather available
space. Play: undercloud
|.....| 7 / 10, 70.00%
2023-02-28 18:30:18.185439 | 52540077-3443-6efd-d781-000000000035 | TIMING | undercloud_disk_space : Loop on volumes and gai
ther available space | undercloud | 0:00:06.733933 | 0.51s
2023-02-28 18:30:18.390990 | 52540077-3443-6efd-d781-000000000035 | TIMING | undercloud_disk_space : Loop on volumes and gai
ther available space | undercloud | 0:00:06.939506 | 0.71s
2023-02-28 18:30:18.596358 | 52540077-3443-6efd-d781-000000000035 | TIMING | undercloud_disk_space : Loop on volumes and gai
ther available space | undercloud | 0:00:07.144872 | 0.92s
2023-02-28 18:30:18.751394 | 52540077-3443-6efd-d781-000000000035 | TIMING | undercloud_disk_space : Loop on volumes and gai
ther available space | undercloud | 0:00:07.299905 | 1.07s
2023-02-28 18:30:18.758679 | 52540077-3443-6efd-d781-000000000035 | TIMING | undercloud_disk_space : Loop on volumes and gai
ther available space | undercloud | 0:00:07.307192 | 1.08s
Next task: undercloud_disk_space : Fail if any of the volumes are too small. Play: undercloud
|.....| 8 / 10, 80.00%
2023-02-28 18:30:19.002748 | 52540077-3443-6efd-d781-000000000036 | TIMING | undercloud_disk_space : Fail if any of the volu
mes are too small | undercloud | 0:00:07.551245 | 0.21s
2023-02-28 18:30:19.006835 | 52540077-3443-6efd-d781-000000000036 | TIMING | undercloud_disk_space : Fail if any of the volu
mes are too small | undercloud | 0:00:07.555359 | 0.21s

```

Рис. 15: Undercloud

Ваш узел развёртывания готов! Для подключения используйте адрес, который вы указали как локальный адрес узла развёртывания или внешний полученный по DHCP в начале установки. Основной пользователь - stack.

1.5 Подготовка к развёртываниям с нуля

Для запуска консоли введите команду в консоли узла развёртывания

```
dctui
```

Для запуска консоли в режиме **debug** используйте команду

```
dctui --debug
```

После выполнения команды появляется **Главное окно** dctui

```
[Acloud] DeckCrew
Common
Baremetal nodes      Description
Network interfaces order  Manage baremetal nodes IPMI login/password/address p
Server root disk       arameters, PXE-ports and roles
Baremetal ports
Networks settings
< Refresh/Ctrl+R >
Deployments           Manage selected deployment: configure services, netw
ork plans and start deployment
< Add/Ctrl+A >      < Delete/Ctrl+D >      < Quit/Ctrl+X >
```

1.5.1 Физические узлы

Перейдите во вкладку *Baremetal nodes*.

```
Baremetal nodes List
Name          State      Power      Cpus      Ram(Mb)      Disk(Gb)      Introspection
Baremetal Node Power State
power on
power off
rebooting
soft power off
soft rebooting
provide
introspect
< Add/Ctrl+A >      < Delete/Ctrl+D >      < Cancel/Ctrl+C >
```

Нажмите кнопку *Add*, чтобы добавить новый физический узел для добавления в развёртываемое облако.

Baremetal node Details

```

Name:
BMC type:          [IPMI]
BMC address:
BMC login:
BMC password:
PXE port mac_address:
Disk capacity (Gb):
Cpu number:
Architecture:     x86_64
Memory capacity (Mb):

< Save/Ctrl+U >                                < Cancel/Ctrl+C >

```

Обязательные для заполнения поля:

- *Name* - имя физического узла. Используется лишь для удобства, не влияет на hostname узла;
- *BMC type* - тип BMC, поддерживаемые значения: *IPMI*, *iDRAC*;
- *BMC address* - IP-адрес BMC, пример: *2.3.4.5* ;
- *BMC login* - логин для управления физическим узлом через BMC;
- *BMC password* - пароль для управления физическим узлом через BMC;
- *PXE port mac_address* - мак адрес физического порта узла, через который происходит загрузке по PXE.

Поля необязательные для заполнения (интроспекция перезапишет эти параметры, поэтому имеет смысл их указывать, если интроспекция не планируется):

- *Disk capacity* - размер физического диска для установки операционной системы;
- *Cpu number* - число процессорных ядер узла;
- *Architecture* - архитектура процессора (поддерживается только *x86_64*);
- *Memory capacity* - размер оперативной памяти узла.

Нажмите кнопку *Save*. Физический узел добавится в список физических узлов.

Baremetal nodes List						
Name	State	Power	Cpus	Ram(Mb)	Disk(Gb)	Introspection
test-node	manageable	off				
Baremetal Node Power State						
power on						
power off						
rebooting						
soft power off						
soft rebooting						
provide						
introspect						
		< Add/Ctrl+A >		< Delete/Ctrl+D >		< Cancel/Ctrl+C >

Дождитесь изменения статуса узла на *manageable*. Если узел остаётся в статусе *enroll*, перейдите в раздел [Устранение неполадок](#).

Статусы:

- *manageable* позволяет включать и выключать узел из системы на узле развёртывания.
- *available* позволяет ввести машину в развёртываемое облако.
- *active* означает, что машина использована Ironic для облака.

Действия

- power on/off - включение/выключение физического узла;
- rebooting - перезагрузка физического узла;
- soft power on/off - отправить сигнал операционной системе на включение/выключение физического узла;
- soft rebooting - отправить сигнал операционной системе на перезагрузку физического узла;
- provide - перевести узел в статус *available* для использования в развёртываемом облаке;
- introspect - запустить интроспекцию физического узла.

1.5.2 Интроспекция

Также на экране снизу отображаются действия для физических узлов. Выберите *introspect*, чтобы запустить интроспекцию узла. Статус интроспекции отображается в правой колонке.

После успешного окончания интроспекции статус будет равен значению *finished*. А узел перейдёт в состояние *available*.

```

----- Baremetal nodes List -----
Name          State      Power    Cpus    Ram(Mb)    Disk(Gb)    Introspection
test-node     manageable off      0       0          0          0
introspected1 available  off      48     4096       1000       finished

Baremetal Node Power State
power on
power off
rebooting
soft power off
soft rebooting
provide
introspect

-----
                < Add/Ctrl+A >          < Delete/Ctrl+D >          < Cancel/Ctrl+C >

```

Порядок физических интерфейсов

! Опция доступна только после успешной интроспекции

Перейдите во вкладку *Network interfaces order*.

```

----- Order NICs -----
Node test-node

Node introspected1
nic1 [0x2000:0x3000 eno1 00:01:02:03:04:05 ]
nic2 [0x2000:0x3000 eno2 00:01:02:03:04:06 ]
nic3 [ ]
nic4 [ ]

```

(continues on next page)

(продолжение с предыдущей страницы)

```
nic5 [
  < Save/Ctrl+U >                                < Cancel/Ctrl+C > ]
```

На экране отобразится список узлов и их интерфейсов. Порядок интерфейсов важен, так как далее сетевые планы строятся только по номерам интерфейсов, а не их именам. Поэтому определите порядок так, чтобы интерфейсы всех узлов с одним порядковым номером были настроены одинаково на физических коммутаторах в вашей серверной.

Диск операционной системы

! Опция доступна только после успешной интроспекции

Перейдите во вкладку *Server root disk*.

```
Choose baremetal root devices
  < Save/Ctrl+U >                                < Cancel/Ctrl+C >
Node test-node
(●) No choice

Node introspected1
( ) No choice
(●) 240546D2DE97
                                hctl: null
                                model: Micron_7450_MTFDKBG3T8TFR
                                name: /dev/nvme0n1
                                rotational: false
                                serial: 240546D2DE97
                                size: 3840755982336
                                vendor: null
                                wwn: eui.00000000000000100a0752446d2de97
                                wwn_vendor_extension: null
                                wwn_with_extension: null
```

Если у какого-либо физического узла несколько дисков, то необходимо указать правильный диск для установки операционной системы. Иначе система выберет физический диск случайным образом.

1.5.3 Физические порты

Перейдите во вкладку *Baremetal ports*.

```

Baremetal ports
Address      Baremetal Node
01:02:03:04:05:06  test-node
00:01:02:03:04:05  introspected1

< Delete/Ctrl+D >          < Cancel/Ctrl+C >

```

Здесь отображаются все физические порты известные системе для загрузки по PXE. Первоначально создаётся порт указанный при создании физической машины. Затем после интроспекции добавляется порт, по которому произошла загрузка по PXE, **если этот порт не совпал с указанным вами при создании**.

Если это произошло, то необходимо оставить один порт в системе и удалить ненужный! Иначе при развёртывании порт для загрузки будет выбран случайным образом из этих двух.

1.5.4 Сети

Перейдите во вкладку *Networks*.

```

List OpenStack physical networks
Name          VLAN  CIDR          Gateway          MTU  Routes
Ctlplane      1     192.168.24.0/24  192.168.24.1    1500 []
External      1     172.16.1.0/24   172.16.1.254   1500 []
InternalApi   2     172.16.2.0/24   172.16.2.254   1500 []
Storage       3     172.16.3.0/24   172.16.3.254   1500 []

< Save/Ctrl+U >          < Add/Ctrl+A >          < Delete/Ctrl+D >          < Cancel/Ctrl+C >

```

Можно изменить параметры сети, нажав на сеть в списке сетей двумя кликами мыши или кнопкой *Enter*.

```

Modify OpenStack physical network
Name:      External

```

(continues on next page)

(продолжение с предыдущей страницы)

```
VLAN ID      1
CIDR        172.16.1.0/24
Allocation pools [{"start": "172.16.1.25", "end": "172.16.1.250"}]
Gateway IP   172.16.1.254
Routes      []
MTU         1500
```

< Save/Ctrl+U >

< Cancel/Ctrl+C >

Если необходима дополнительная сеть, то можно добавить новую, нажав кнопку *Add*.

1.5.5 Шаблоны развёртывания

В главное меню в разделе *Deployments* отображается список созданных шаблонов для развёртывания.

При чистом старте - таблица будет пустая. Нажмите кнопку *Add*, чтобы добавить новое развёртывание.

В появившемся окне вы можете выбрать один из заранее заданных типов развёртывания: *OpenStack* или *Ceph*, или выбрать *No*.

Затем введите имя развёртывания. Имя будет использоваться как часть hostname физических узлов.

Если OpenStack планируется развернуть:

- **без Ceph**, то создайте единственное развёртывание типа *OpenStack*.
- **с Ceph**, то создайте сначала развёртывание *Ceph*, затем *OpenStack*

На данном этапе при создании развёртывания создаются только шаблоны для настройки развёртываемого облака. Никаких изменений с физическими машинами не происходит.

На главном экране *dcui* в секции *Deployments* появятся ваши созданные развёртывания для настройки.

```
[Acloud] DeckCrew
Common
Baremetal nodes      Manage baremetal nodes IPMI login/password/address p
Network interfaces order
Server root disk
Baremetal ports
```

(continues on next page)

(продолжение с предыдущей страницы)

```

Networks settings
-----
Deployments
test
Manage selected deployment: configure services, network plans and start deployment

< Refresh/Ctrl+R >
< Add/Ctrl+A >      < Delete/Ctrl+D >      < Quit/Ctrl+X >

```

1.6 Подготовка одного развёртывания

В этом разделе описана подготовка одного развёртывания с нуля. Обязательно пройдите подготовку к развёртываниям в разделе *Подготовка к развёртываниям*.

При деплое OpenStack с Ceph необходимо сначала развернуть Ceph!

Выберите одно из развёртываний в секции *Deployments*.

```

----- Configure OpenStack Installation -----
Menu
Roles/Nodes and Networks
Virtual IPs
Passwords
Endpoint Data
Service settings
Deployment
Check cloud deployment
Prepare cloud
Cloud Admin Actions

Description
Manage roles, baremetal nodes and their networks, IPs and images

< Cancel/Ctrl+C >

```

На экране отображается **Меню развёртывания**, в котором все настройки влияют только на выбранное развёртывание.

1.6.1 Роли/Узлы и Сети ролей

Перейдите во вкладку *Roles/Nodes and Networks*.

```

===== Deployment roles and nodes =====
Roles: select role to edit
Name                Count      Image
Controller          0         /var/lib/ironic/images/overcloud.x86_64.qcow2
Compute             0         /var/lib/ironic/images/overcloud.x86_64.qcow2
=====
Nodes: select node to edit
Name                Hostname                Managed by Ironic
=====
< Save/Ctrl+U >    < Add/Ctrl+A >    < Delete/Ctrl+D >    < Cancel/Ctrl+C >

```

Сверху отображается сводная таблица настроек для каждой из ролей, участвующих в развёртывании.

Ниже отображается сводная таблица заведённых в развёртывание узлов. Сначала она пустая.

Развёртывание типа *OpenStack* требует обязательные роли:

- **Controller** - узлы контроллеры
- **ComputeInstanceNA** - узлы виртуализации

Развёртывание типа *Ceph* требует только роль **BareCeph**, в которой нет сервисов. Так как само развёртывание Ceph системы происходит за пределами консоли при помощи инструмента **ceph-ansible**.

Настройка роли

Выберите в таблице роль и нажмите *Enter*

```

===== Role/Node settings =====
Role                Controller

```

(continues on next page)

(продолжение с предыдущей страницы)

```

=====
Settings                               Network plan template preview
=====
Image      [overcloud.x86_64.qcow2      ] dns_servers: {{ ctlplane_dns_nameservers }}
Networks
  external  [✓]                          domain: {{ dns_search_domains }}
  internal_api [✓]                        addresses:
  storage   [✓]                          - ip_netmask: {{ ctlplane_ip }}/{{ ctlplane_subnet
                                     _cidr }}
Network Plan [single_nic_vlans/single_nic_vlans.] routes: {{ ctlplane_host_routes }}
Default Route via [external              ] members:
DNS servers
Bond options
                                     - type: interface
                                     name: nic1
                                     mtu: {{ min_viable_mtu }}
                                     # force the MAC address of the bridge to this in
                                     terface
                                     primary: true
                                     {% for network in role_networks %}
                                     - type: vlan
                                     mtu: {{ lookup('vars', networks_lower[network] ~
                                     '_mtu') }}
                                     vlan_id: {{ lookup('vars', networks_lower[network
                                     k] ~ '_vlan_id') }}
                                     addresses:
                                     - ip_netmask:
                                     {{ lookup('vars', networks_lower[network] ~
                                     '_ip') }}/{{ lookup('vars', networks_lower[network]
                                     ~ '_cidr') }}
                                     routes: {{ lookup('vars', networks_lower[network
                                     ] ~ '_host_routes') }}
                                     {% endfor %}

< Save/Ctrl+U >                               < Cancel/Ctrl+C >

```

Измените настройки роли, если это необходимо.

- *Image* - образ операционной системы для узлов данной роли
- *Networks* - сети, которые используются на узлах данной роли. Сети соответствуют тем, что были созданы ранее.

- *Network Plan* - сетевой план для узлов данной роли. Для более подробной информации прочитайте пункт *Сетевые планы*
- *Default Route via* - сеть, чей IP-адрес шлюза будет использоваться как основной шлюз для сети на узле.
- *DNS servers* - DNS сервера для узлов данной роли
- *Bond options* - опции для агрегации интерфейсов. В развёртываниях типа Serf значение параметра должно быть равно *mode=active-backup*, типа OpenStack - *bond_mode=active-backup*. Так как в Serf используется агрегация ядра Linux, в OpenStack - агрегация Ovs.

Добавление узла

Для добавления узла в развёртывание нажмите кнопку *Add* и выберите *Node* во всплывающем окне.

Role/Node settings

Manage by Ironic unset for manually deployed node

Baremetal Node []

Role []

Settings
Network plan template preview

Image use default from role

Networks and Fixed IPs use default from role

Network Plan use default from role

Default Route via use default from role

DNS servers use default from role

Bond options use default from role

< Save/Ctrl+U >
< Cancel/Ctrl+C >

Основные настройки:

- *Managed by Ironic* - управление узлом через Ironic. Этот пункт подразумевает, что узел пустой и при деплое будет использован сервисом Ironic, для развёртывания операционной системе на этом узле. Если предполагается использовать уже развёрнутый узел, перейдите в *Использование заранее настроенных узлов*
- *Baremetal Node* - имя физического узла, созданного заранее в меню *Baremetal Nodes*.
- *Role* - роль узла в развёртывании.

По умолчанию настройки в разделе *Settings* используют настройки роли. Можно снять галку для специальных настроек отличных от настроек ролей

Использование заранее настроенных узлов

Снимите галку с *Manage by Ironic*

```

----- Role/Node settings -----
Manage by Ironic [ ] unset for manually deployed  WARNING: if you disable Ironic management, ensure:
Display Name          - tripleo-admin user is created
Role                  [ ]- tripleo-admin has undercloud ssh key in .ssh/authori
                    rized_keys
                    - tripleo-admin has sudo rights with no password
=====
Settings              Network plan template preview
=====
Image                 [✓] use default from role
Networks and...      [ ] use default from role
  external -->       [ ]
  internal_api -->  [ ]
  storage -->       [ ]
Fixed IP
  external -->
  internal_api -->
  storage -->
  ctlplane -->
Network Plan         [✓] use default from role
Default Route via   [✓] use default from role
DNS servers         [✓] use default from role
Bond options        [✓] use default from role

< Save/Ctrl+U >                                < Cancel/Ctrl+C >

```

Справа появится подсказка для настройки узла. Необходимо

- создать пользователя **tripleo-admin**
- добавить ключ пользователя stack с узла развёртывания `.ssh/id_rsa` на добавляемый узел в файл `.ssh/authorized_keys` для пользователя `tripleo-admin`

- добавить права **sudo без пароля** для пользователя tripleo-admin

Укажите имя узла в поле *Display Name* и выберите роль узла в поле *Role*.

Обязательно укажите используемые сети, и IP-адреса выделенные в каждой сети для этого узла.

Сетевые планы

Сетевые планы фиксируют число используемых интерфейсов, настройки адресов или DHCP для каждого из интерфейсов, VLAN'ы и на каких интерфейсах настраиваются VLAN'ы. Необходимо предварительно определить схему настройки сети для всех узлов одной роли, используя знания о количестве интерфейсов на каждом узле одной роли.

Схема сети определяется через условные обозначения интерфейсов как *nic1*, *nic2* и т.д. Порядок интерфейсов для каждого из узлов задавался ранее в разделе *Network interfaces order*.

Если сетей для роли больше чем число используемых интерфейсов, зафиксированное для данной роли, то использование VLAN неизбежно. В этом случае для настройки VLAN в развёртываниях типа Serp используются VLAN ядра Linux, в развёртываниях типа OpenStack используется OVS бридж.

После определения сетевого плана - выйдете из консоли и зайдите в папку шаблонов вашего развёртывания.

```
cd /usr/share/ansible/roles/tripleo_network_config/templates/
```

Исследуйте заранее спроектированные планы. Указанные в шаблонах планы являются настройками инструмента *os-net-config*. Для детального понимания используйте [документацию инструмента](#)

Выберите один из заданных сетевых планов или создайте новый в той же папке

Зтем вернитесь в *dctui* в настройки роли и выберите для каждой роли свой сетевой план.

1.6.2 Виртуальные IP-адреса

Виртуальные IP-адреса используются в *OpenStack* для балансировки нагрузки на узлы управления в облаке.

Перейдите во вкладку *Virtual IPs* и нажмите кнопку *Add*.

Modify VIP Data

Network:	[external]]
DNS Name:		
IP Address:		

(continues on next page)

(продолжение с предыдущей страницы)

< Save/Ctrl+U >

< Cancel/Ctrl+C >

Выберите сеть для виртуального адреса и опционально укажите DNS имя и IP-адрес.

- DNS имя используется только если присутствует Designate.
- IP-адрес является фиксированным для виртуального адреса.

Нажмите кнопку *Save*.

Network	DNS Name	VIPs	All Data	IP Addr
external				
< Save/Ctrl+U >	< Add/Ctrl+A >	< Delete/Ctrl+D >	< Cancel/Ctrl+C >	

При необходимости выберите сеть и нажмите *Enter*, чтобы изменить параметры.

Затем снова нажмите кнопку *Save*, чтобы сохранить настроенные изменения в файл. На этом этапе порты не создаются. Чтобы применить настройки к развёртыванию, перейдите во вкладку *Deployment*.

Если развёртывание типа:

- **Ceph**, то создание виртуальных адресов не нужно.
- **OpenStack**, то необходимы виртуальные адреса для сетей *External*, *InternalApi* и опционально для других сетей.

1.6.3 Пароли

Перейдите во вкладку *Passwords* и нажмите кнопку *Generate*.

Cloud Services Passwords	
AdminPassword	yhvt7AYXeW5hr
AodhPassword	0mc3DQWjXj9jTeH2R84mbf3eo
AosAmqpPassword	TXAJb18rHw6HPkzQIvTvul823

(continues on next page)

(продолжение с предыдущей страницы)

```

AosDatabasePassword      ad4fEflzmbVPba2T3dmpWCbrt
AosPassword              I6535DwyQGhSCosNPxBZ062BE
BarbicanPassword         OR6AQJijcwreMBIPQNuTFLfp4
BarbicanSimpleCryptoKek  8Ju0Ipj7AVed3fA0xDTfy01pw4TOWGQPscBAIEyyPcQ=
CeilometerMeteringSecret oiSnBrtQVppFFSNNjOa4kSeFG
CeilometerPassword      NjUQnlqYjPvGp4fAqkUYsABYt
CephClientKey            AQB JgR9nAAAAABAAo4n64FE70B1G0oh7sKJ10Q==
CephClusterFSID         6c1874bd-8164-481c-8c6e-9ee08920def1
CephDashboardAdminPassword 00ZCxXagzh2lTufBuqEFZsrQ2
...
RedisPassword           uziPg5Phcs8T7s2tcNP0bnfy6
RpcPassword             GMBKDa8CdNIN9YxloDdNaaU1Y
SkylineDBPassword      Z4kU2ryf1NWyQjhV5E1yAhfYY
SkylineOSPassword      89505xQ6wc3v8PdRRLNAmWThP
SkylineSecretKey       ZXJOPbcYn1NBHAcJnGOSsb8TFp4JCzC5
SnmpdReadOnlyUserPassword 2HaDICOH2kPf6huUQYK11UBc
SwiftHashSuffix        a7iOLOZsrSxqfBHssONkMIuq9
SwiftPassword          RRG84d5RQ5jv9k7KA4ilyRGZL
  < Save/Ctrl+U >      < Generate/Ctrl+A >      < Clear/Ctrl+D >      < Cancel/Ctrl+C >

```

Это окно имеет смысл только для развёртываний типа OpenStack. При развёртывании Ceph этот этап можно пропустить.

Здесь обратите внимание на поля:

- *AdminPassword* - это пароль для пользователя *admin* в развёртываемом облаке
- *CephClientKey*, *CephClusterFSID* - значения обязательные для связи сервисов OpenStack с Ceph.

Остальные пароли можно сгенерить случайным образом.

1.6.4 Настройка точек подключения для сервисов облака

Перейдите во вкладку *Endpoint Data*.

```

Endpoint Data and Map
port: 8989
Mysql:
  Internal:
AdminNetwork
  < use >
  < remove >

```

(continues on next page)

(продолжение с предыдущей страницы)

```

net_param: Mysql
port: 3306
protocol: mysql+pymysql
Neutron:
Admin:
net_param: NeutronApi
Internal:
net_param: NeutronApi
Public:
host: CLOUDNAME
net_param: Public
port: 9696
Nova:
Admin:
net_param: NovaApi
uri_suffixes:
': /v2.1
Internal:
net_param: NovaApi
uri_suffixes:
': /v2.1
Public:
host: CLOUDNAME
net_param: Public
uri_suffixes:
': /v2.1
< Save/Ctrl+U > < Generate Map/Ctrl+A > < Delete Changes/Ctrl+D > < Cancel/Ctrl+C >

```

TLS
 < use >
 < remove >

DNS
 < use >
 < remove >

AdminNetwork is option to use or remove Ctlplane network from endpoints as Admin Endpoint

Endpoint Data is used to generate Endpoint Map for cloud and not used while deployment.
 You can update and save data multiple times.
 Eventually you should generate Endpoint Map before deployment.

Здесь отображаются основные сети для различного типа подключения к сервисам.

Формат подключения:

```

ServiceName:
Admin:
net_param: ServiceNet
Internal:
net_param: ServiceNet

```

(continues on next page)

(продолжение с предыдущей страницы)

```
Public:
  host: CLOUDNAME
  net_param: Public
```

Где `host: CLOUDNAME` определяет подключение по доменному имени виртуального адреса.

Чтобы убрать или добавить какую либо настройку, используйте кнопки *use* и *remove*.

- *AdminNetwork* - использовать `ctrlplane` вместо `internal_api` для подключения к сервису
- *TLS* - использовать SSL/TLS для подключения к сервису
- *DNS* - использовать доменное имя для подключения к сервису

1.6.5 Настройки сервисов облака

Перейдите во вкладку *Service settings*.

Configure OpenStack Installation	
Function	Enabled
AosModules	[✓]
BareRole	[]
CephExternal	[]
CinderCephBackend	[]
ComputeInstanceha	[]
CustomDomain	[]
DnsServers	[✓]
EnableTls	[]
ExtraFirewallRules	[]
FederationOpenidc	[]
GlanceCephBackend	[]
GlanceCinderBackend	[]
GlanceFileBackend	[✓]
GlanceNfsBackend	[]
InjectTrustAnchorHiera	[]
Ironic	[]
IronicInspector	[]
KeystoneLdapBackend	[]

(continues on next page)

(продолжение с предыдущей страницы)

```

KeystoneSecurityCompliance      [ ]
Michman                          [✓]
NeutronDvr                       [✓]
NtpServers                       [✓]
SwiftExternal                    [ ]
                                  < Advanced/Ctrl+A >
                                  < Cancel/Ctrl+C >

```

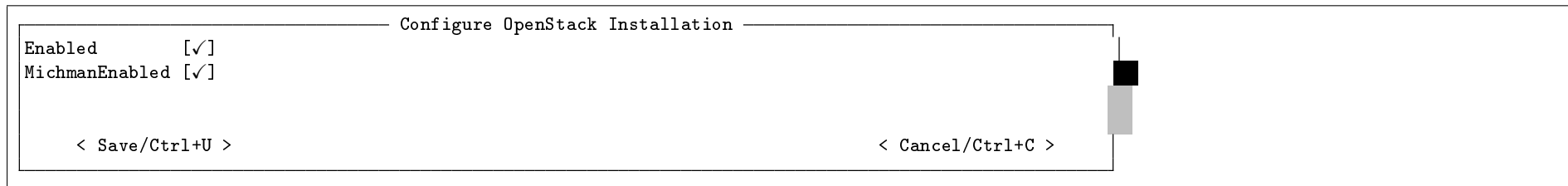
Необходимо внимательно пройти по всем пунктам в этом окне, указать только используемые опции. Те опции, что используются, необходимо обязательно настроить внутри своими параметрами!

Все опции, кроме *BareRole*, имеют смысл только при развёртывании типа *OpenStack*!

- *AosModules* - использовать модули AOS для развёртывания облака: *CloudManager*, *RsServer*, *Journal*, *Monitor*, *OtpSender*, *Agent*
- *BareRole* - отметить, если используется роль *Bare* для развёртывания
- *CephExternal* - обязателен при использовании *Ceph*, необходимо настроить все параметры внутри и **подготовить Ceph**;
CephExternalMonHost заполняется в формате списка IP-адресов через запятую.
Например, `10.10.10.1:3300,10.10.10.2:3300,10.10.10.3:3300,10.10.10.4:3300`
- *CinderCephBackend* - использовать *Ceph* как бекенд для сервиса *Cinder*
- *ComputeInstancecha* - обязателен для развёртывания типа *OpenStack* с 3-мя и более контроллерами;
- *CustomDomain* - назначить имя домена для API развёртываемого облака
- *DnsServers* - изменить DNS сервера для развёртываемого облака. По умолчанию используется DNS из настроек узла развёртывания.
- *EnableTls* - использовать SSL/TLS для публичных сервисов API.
- *ExtraFirewallRules* - добавить дополнительные правила для файрвола.
- *FederationOpenidc* - использовать OpenID Connect для сервиса *Keystone*
- *GlanceCephBackend* - использовать *Ceph* сервер как бекенд для сервиса *Glance*
- *GlanceCinderBackend* - использовать *Cinder* сервер как бекенд для сервиса *Glance*
- *GlanceFileBackend* - использовать файловое хранилище как бекенд для сервиса *Glance*
- *GlanceNfsBackend* - использовать *Nfs* сервер как бекенд для сервиса *Glance*
- *InjectTrustAnchorHiera* - использовать самоподписанный сертификат для публичных сервисов API

- *Ironic, IronicInspector* - добавить в развёртываемое облако сервис Ironic и IronicInspector
- *KeystoneLdapBackend* - использовать готовый LDAP сервер для сервиса [Keystone](#). Для этого вам также понадобится использовать несколько доменов для авторизации в сервисе Skyline.
- *KeystoneSecurityCompliance* - использовать дополнительные настройки безопасности для сервиса [Keystone](#)
- *Michman* - указывается для развёртывания инструмента Michman.
- *NeutronDvr* - использовать распределённую виртуализацию для сервиса Neutron
- *NtpServers* - изменить NTP сервера для развёртываемого облака. По умолчанию используется NTP из настроек узла развёртывания.
- *SwiftExternal* - использовать Ceph Rgw как бекенд для сервиса [Swift](#)

Чтобы добавить или удалить опцию, выберите нужную, и нажмите *Enter*. Например,



Первым пунктом всегда будет *Enabled* обозначающий добавление этой опции в развёртывание. Остальные пункты различаются между опциями и настраиваются внутри.

Добавьте или снимите галку *Enabled* и нажмите кнопку *Save*.

При определении бекендов сервисов необходимо учитывать, что для Glance и Nova может быть выбран только один. Для сервиса Cinder может использоваться несколько разных бекендов.

1.6.6 Добавление SSL/TLS для публичных сервисов API

1. В настройках для развёртывания зайти во вкладку *Endpoint Data*.
2. Под заголовком *TLS* нажать кнопку *< use >*.
3. Далее нажать кнопку *< Generate Map/Ctrl+A >*.
4. Далее зайти во вкладку *Service Settings*.
5. Добавить свойство *EnableTls*

6. Заполнить значения для сертификата и ключа.

7. Если сертификат планируется самоподписанный, то добавить свойство `InjectTrustAnchor` и заполнить root сертификат.

[Инструкция по добавлению SSL/TLS](#)

1.7 Развёртывание облака

Зайдите в DeckCrew TUI

```
dctui
```

Перейдите в *Меню развёртывания*, затем во вкладку *Deployment*.

```

Deployment process
< Refresh/Ctrl+R >

Stack status
last_action: Validate Teampates
status: DEPLOY_SUCCESS

Debug mode
[ ] No launch. Generate deployment script only

Pick deployment action
(●) Prepare Networks
( ) Prepare Virtual IPs
( ) Provision Nodes
( ) Validate Teampates
( ) Deploy OpenStack
( ) Update Known Hosts

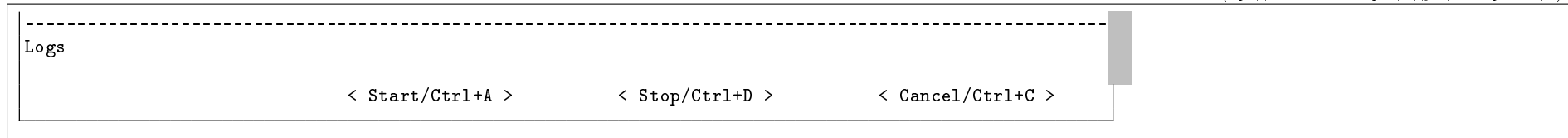
[ ] Activate danger section
(●) Update Partition
( ) Delete Nodes
( ) Delete Deployment

Set extra settings for deployment action

```

(continues on next page)

(продолжение с предыдущей страницы)



Сверху отображается статус развёртывания.

Развёртывание состоит из нескольких стадий:

- **Prepare Networks**, или *Подготовка сети* - на этом этапе создаются сети, настроенные ранее.
- **Prepare Virtual IPs**, или *Подготовка виртуальных IP-адресов* - на этом этапе создаются виртуальные IP-адреса, настроенные ранее.
- **Provision Nodes**, или *Подготовка узлов* - на этом этапе при помощи Ironic разворачиваются операционные сети на узлы и настраиваются сети.
- **Validate Templates**, или *Проверка шаблонов* - на этом этапе проверяются настройки шаблонов.
- **Deploy OpenStack**, или *Развёртывание OpenStack* - на этом этапе разворачиваются сервисы OpenStack на узлах. Если вы пропустили этап *Validate Templates*, то укажите в *Set extra settings for deployment action* параметр *Deploy with Stack Update*.
- **Update Known Hosts**, или *Обновление известных хостов* - на этом этапе обновляется файл `.ssh/known_hosts` на узле развёртывания.

Также можно использовать *Debug* режим, при котором команда для запуска развёртывания не выполняется, а сохраняется в `/tmp/deploy-.sh`.

После создания сервера статус физического узла меняется с *available* на *active* в списке физических серверов.

1.7.1 Нулевое развёртывание

Нулевое развёртывание означает, что *Stack status* для данного развёртывания отображает пустые значения для всех параметров.

В этом случае необходимо запустить по порядку все *Deployment actions* на экране по порядку, дождавшись окончания предыдущего.

В случае любых проблем и ошибок на каждом этапе необходимо исправить ошибки и запустить развёртывание заново. Ни один из этапов не может быть пропущен!

1.7.2 Обновление параметров развёртывания

Этот пункт вам подходит, если вы изменили:

- порядок интерфейсов для узла, участвующего в этом развёртывании
- диск операционной системы для узла, участвующего в этом развёртывании
- параметры сети
- параметры сети для роли
- состав сервисов для роли
- сетевой план для роли
- пароли для сервисов
- фиксированные IP-адреса
- настройки опций для сервисов

В этом случае необходимо сначала обновить изменённые параметры в системе узла развёртывания.

Для этого пройдите первый этап *Validate Templates*, затем перейдите к этапу *Deploy OpenStack* и выполните его.

Удалять уже развёрнутые сети таким образом нельзя!

1.7.3 Обновление контейнеров

Все образы контейнеров хранятся в репозитории на узле развёртывания. Адрес репозитория `<hostname>.ctlplane:13787`, далее используется `undercloud.ctlplane:13787`.

Для обновления образов контейнеров используйте команду `skopeo`. Например, с узла развёртывания выполните команду:

```
skopeo copy docker://obs:82/acloud/openstack-nova-compute:latest \  
docker://undercloud.ctlplane:13787/acloud/openstack-nova-compute:latest
```

Этой командой обновляются образы контейнеров в реестре на узле развёртывания. Для того чтобы загрузить обновлённые образы на узлы облака, необходимо зайти на узел облака и выполнить команду:

```
sudo podman pull undercloud.ctlplane:13787/acloud/openstack-nova-compute:latest
```

Для пересоздания контейнеров на узлах необходимо выполнить *Deploy OpenStack*. Эта операция пересоздаст все контейнеры на узлах. Также будут пересозданы все конфигурации из папки `/var/lib/config-data/`, где хранятся конфигурации сервисов OpenStack. Данные сервисов OpenStack не будут потеряны, так как все данные, как правило, хранятся во внешних по отношению к контейнерам вольюмах на хосте контейнера.

Внимательно! Операция *Deploy OpenStack* не затронет работу Pacemaker и Serrh.

Для того чтобы обновить контейнеры управляемые Pacemaker, необходимо с одного из узлов управления выполнить команду:

```
sudo podman tag undercloud.ctlplane:13787/acloud/<image_name>:latest \
undercloud.ctlplane:13787/acloud/<image_name>:pcmklatest
sudo pcs status
sudo pcs resource restart <service_resource>
```

Например

```
sudo podman tag undercloud.ctlplane:13787/acloud/openstack-haproxy:latest \
undercloud.ctlplane:13787/acloud/openstack-haproxy:pcmklatest
sudo pcs status
sudo pcs resource restart haproxy-bundle-podman-0
```

1.7.4 Добавление узлов в облако

Для добавления новых узлов необходимо пройти пункты из *Подготовка развёртывания с нуля*, касающиеся физических машин:

- Физические узлы
- Интроспекция
- Порядок физических интерфейсов
- Диск операционной системы
- Физические порты

Затем в *Меню развёртывания* в окне *Roles/Nodes and Networks* добавьте ваш узел и перейдите к стадии развёртывания.

На стадии развёртывания сначала добавьте новый сервер, выполнив *Provision Nodes*.

При успешном добавлении сервера, выполните *Update Known Hosts*.

Далее выберите *Deploy OpenStack* и поставьте галочки в дополнительных параметрах снизу в полях: *undercloud*, *-controller-0* и новые добавленные сервера. Затем нажмите *Start*. Таким образом добавление узла будет происходить быстрее.

1.7.5 Удаление развёрнутых узлов из облака

Для выведения узла из развёртывания перейдите в раздел *Deployment*, поставьте галку рядом с *Activate danger section* и выберите *Delete Nodes*.

Deployment process

< Refresh/Ctrl+R >

Stack status
last_action: Validate Teampates
status: DEPLOY_SUCCESS

Debug mode
 No launch. Generate deployment script only

Pick deployment action
 Prepare Networks
 Prepare Virtual IPs
 Provision Nodes
 Validate Templates
 Deploy OpenStack
 Update Known Hosts

Activate danger section
 Update Partition
 Delete Nodes
 Delete Deployment

Set extra settings for deployment action
 test-compute-0
 test-compute-1
 test-controller-0
 test-controller-1
 test-controller-2

Logs

Выберите узлы для удаления и нажмите кнопку *Start*.

Затем вернитесь в меню *Roles/Nodes and Networks* и удалите ваш узел из списка узлов используя кнопку *Delete*.

1.7.6 Замена развёрнутых узлов в облаке

Для замены узла необходимо сначала добавить новый физический узел и настроить его согласно инструкции из *Добавление узлов в облако*.

Затем перейдите в раздел *Roles/Nodes and Networks*, удалите узел, который хотите заменить, и добавьте новый узел.

Затем в разделе *Deployment* выполните *Provision Nodes*, *Update Known Hosts* и *Deploy OpenStack*.

1.7.7 Частичное развёртывание

На стадии **Run Cloud Service Deployment** обновляется окно выбора с опциями *Limit to*:

```

Deployment process
< Refresh/Ctrl+R >

Stack status
last_action: Validate Teamplates
status: DEPLOY_SUCCESS

Debug mode
[ ] No launch. Generate deployment script only

Pick deployment action
( ) Prepare Networks
( ) Prepare Virtual IPs
( ) Provision Nodes
( ) Validate Teamplates
(●) Deploy OpenStack
( ) Update Known Hosts

[ ] Activate danger section
(●) Update Partition
( ) Delete Nodes
( ) Delete Deployment

```

(continues on next page)

(продолжение с предыдущей страницы)

```

Set extra settings for deployment action
[ ] Deploy with Stack Update
[ ] Limit to undercloud
[ ] Limit to test-compute-0
[ ] Limit to test-compute-1
[ ] Limit to test-controller-0
[ ] Limit to test-controller-1
[ ] Limit to test-controller-2

-----
Logs

                < Start/Ctrl+A >          < Stop/Ctrl+D >          < Cancel/Ctrl+C >

```

Отметьте узлы для развёртывания и нажмите *Start*.

При добавлении новых узлов отмечайте помимо нового узла *undercloud* и *-controller-0*, так как с узла развёртывания производятся обращения к API развёртываемого облака для добавления вычислительного узла. А на узле управления выполняются команды для добавления нового узла в кластер Rasemaker.

1.8 Полезные советы

1.8.1 Приложение

Горячие клавиши

Следующий список сочетаний клавиш работает в любой момент, в каком окне вы бы ни находились

- *Ctrl+X (quit)* - выход из программы
- *Ctrl+C (cancel)* - выход из окна на уровень выше
- *Ctrl+R (refresh)* - обновить содержимое текущего окна

Следующий список сочетаний клавиш актуален только в тех окнах, где эти клавиши отображаются в окне

- *Ctrl+U (update)* - сохранить изменения

- *Ctrl+A (add/start)* - добавить сущность (сеть, узел и т.д)/запустить выполнение сценария
- *Ctrl+D (delete/stop)* - удалить сущность/остановить выполнение сценария

Установка узла развёртывания в виртуальной машине (Centos 7,8)

Установите необходимые пакеты на узел

```
sudo dnf install -y bridge-utils qemu-kvm qemu-img virt-manager virt-viewer python3-libvirt libvirt libvirt-client libguestfs libguestfs-tools virt-install libgcrypt libxcrypt
```

Для виртуальной машины узла развёртывания необходимо создать два бриджа и указать их при создании виртуальной машины KVM. Внутри VM они будут светиться как два интерфейса: один используется для внутренней сети настройки облачной платформы, а второй используется для доступа к узлу развёртывания.

Создать мост для интерфейса, на котором планируется запустить KVM VM при помощи *network-scripts*

```
nano /etc/sysconfig/network-scripts/ifcfg-eno1
```

```
DEVICE=eno1
ONBOOT=yes
HOTPLUG=no
NM_CONTROLLED=no
MTU=1500
BOOTPROTO=none
BRIDGE=br-eno1
TYPE=Ethernet
```

```
nano /etc/sysconfig/network-scripts/ifcfg-br-eno1
```

```
DEVICE=br-eno1
ONBOOT=yes
TYPE=Bridge
DELAY=0
BOOTPROTO=static
IPADDR=192.168.24.64
NETMASK=255.255.255.0
NM_CONTROLLED=no
DEFROUTE=no
```

Или создайте мост при помощи Linux утилиты *ip*

```
ip link add link eno1 name eno1.100 type vlan id 100
ip link set eno1.100 up
ip link add br100 type bridge
ip link set eno1.100 master br100
ip link set br100 up
```

Для проверки настроек можно использовать

```
sudo brctl show
```

Для запуска VM выполните

```
virt-install --virt-type=kvm --name undercloud-base --ram 32768 --vcpus=16 --os-variant=centos8 --cdrom=/var/lib/libvirt/boot/undercloud.iso --
network=bridge=br-eno1 --network=bridge=br-eno2 --disk path=/var/lib/libvirt/images/undercloud-base.qcow2,size=150,format=qcow2
```

KVM создаст виртуальные интерфейсы `vnet0` и `vnet1` в составе бриджей, если бридж падает, то интерфейсы из него исчезают. Чтобы добавить снова их в бридж можно выполнить

```
ip link set vnet1 master br-eno2
```

Описание узла развёртывания - TripleO

Основной пользователь, от имени которого выполняется управление всеми системами узла развёртывания - **stack**. Подключитесь к узлу развёртывания по SSH. Далее все команды и действия будут указываться с узла развёртывания от имени пользователя **stack**

```
ssh stack@$ip
```

При успешной установке узла развёртывания на нём будет развёрнут полноценный **однонодовый OpenStack**. В том числе будут установлены дополнительные пакеты TripleO и overcloud CLI.

В домашней директории `/home/stack` находится файл **stackrc**.

```
[stack@undercloud ~]$ cat stackrc
# Clear any old environment that may conflict.
for key in $( set | awk -F= '/^OS_/ {print $1}' ); do unset "${key}" ; done

export OS_AUTH_TYPE=password
```

(continues on next page)

(продолжение с предыдущей страницы)

```

export OS_PASSWORD=*****
export OS_AUTH_URL=https://192.168.24.2:13000
export OS_USERNAME=admin
export OS_PROJECT_NAME=admin
export COMPUTE_API_VERSION=1.1
export NOVA_VERSION=1.1
export OS_NO_CACHE=True
export OS_CLOUDNAME=undercloud
export OS_IDENTITY_API_VERSION='3'
export OS_PROJECT_DOMAIN_NAME='Default'
export OS_USER_DOMAIN_NAME='Default'
export OS_CACERT="/etc/pki/ca-trust/source/anchors/cm-local-ca.pem"
# Add OS_CLOUDNAME to PS1
if [ -z "${CLOUDPROMPT_ENABLED:-}" ]; then
    export PS1=${PS1:-""}
    export PS1=\${OS_CLOUDNAME:+"(\${OS_CLOUDNAME})"}\ $PS1
    export CLOUDPROMPT_ENABLED=1
fi

```

Этот файл используется для обращения к OpenStack API узла развёртывания. Все интерфейсы этого API при этом доступны только в сети **Ctlplane**.

```

[stack@undercloud ~]$ source stackrc
(undercloud) [stack@undercloud ~]$ openstack endpoint list --interface public -c "Service Name" -c "Service Type" -c URL
+-----+-----+-----+
| Service Name | Service Type | URL |
+-----+-----+-----+
| glance       | image        | https://192.168.24.2:13292 |
| zaqar-websocket | messaging-websocket | wss://192.168.24.2:9000 |
| ironic       | baremetal    | https://192.168.24.2:13385 |
| keystone     | identity     | https://192.168.24.2:13000 |
| swift       | object-store | https://192.168.24.2:13808/v1/AUTH_%(tenant_id)s |
| zaqar        | messaging    | https://192.168.24.2:13888 |
| placement    | placement    | https://192.168.24.2:13778/placement |
| heat         | orchestration | https://192.168.24.2:13004/v1/%(tenant_id)s |
| ironic-inspector | baremetal-introspection | https://192.168.24.2:13050 |
| mistral      | workflowv2   | https://192.168.24.2:13989/v2 |
| neutron      | network      | https://192.168.24.2:13696 |
| nova         | compute      | https://192.168.24.2:13774/v2.1 |
+-----+-----+-----+

```

Среди созданных сетей `openstack network list` вы увидите сеть `ctlplane`.

Все сервисы OpenStack запущены в контейнерах. Для проверки этого выполните

```
(undercloud) [stack@undercloud ~]$ sudo podman ps
CONTAINER ID   IMAGE                                     COMMAND          CREATED        STATUS          NAMES
91124d20d2d9   undercloud.ctlplane:13787/acloud/openstack-memcached:latest  kolla_start     3 weeks ago   Up 3 weeks (healthy)  memcached
b7333c04c4c2   undercloud.ctlplane:13787/acloud/openstack-haproxy:latest    kolla_start     3 weeks ago   Up 10 days           haproxy
4698f4d2425b   undercloud.ctlplane:13787/acloud/openstack-rabbitmq:latest    kolla_start     3 weeks ago   Up 3 weeks (healthy)  rabbitmq
489fddc0f48f   undercloud.ctlplane:13787/acloud/openstack-mariadb:latest     kolla_start     3 weeks ago   Up 3 weeks (healthy)  mysql
b399d357390f   undercloud.ctlplane:13787/acloud/openstack-keystone:latest    kolla_start     3 weeks ago   Up 3 weeks (healthy)  keystone
69d2f12b398b   undercloud.ctlplane:13787/acloud/openstack-iscsid:latest      kolla_start     3 weeks ago   Up 3 weeks (healthy)  iscsid
<...>
```

Каждому рабочему контейнеру также соответствует сервис `systemd`, который перезапускает контейнер при падении. Выполните:

```
(undercloud) [stack@undercloud ~]$ sudo systemctl | grep tripleo_
tripleo_glance_api.service      loaded active running  glance_api container
tripleo_haproxy.service         loaded active running  haproxy container
tripleo_heat_api.service        loaded active running  heat_api container
tripleo_heat_api_cron.service   loaded active running  heat_api_cron container
tripleo_heat_engine.service     loaded active running  heat_engine container
tripleo_ironic_api.service      loaded active running  ironic_api container
tripleo_ironic_conductor.service loaded active running  ironic_conductor container
tripleo_ironic_inspector.service loaded active running  ironic_inspector container
tripleo_ironic_inspector_dnsmasq.service loaded active running  ironic_inspector_dnsmasq container
<...>
```

Поэтому для того, чтобы **остановить** какой-либо **контейнер**, используйте команду

```
sudo systemctl stop tripleo_$(service_name)
```

Все **изменяемые файлы и директории** в сервисах как правило выносятся в хост систему из контейнера.

```
sudo podman inspect $(container_name) | jq .[0].Mounts
```

Поэтому при необходимости можно контейнеры удалять и **пересоздавать**, не боясь потерять данные. Для этого тоже можно использовать `podman inspect`, чтобы узнать команду, которой был создан контейнер.

```
sudo podman inspect $(container_name) | jq .[0].Config.CreateCommand
```

Обратите внимание, что все контейнеры (*кроме одного*) запускаются с командой `kolla_start`. Эта команда использует файлы конфигурации `kolla_config` для запуска процессов внутри контейнера. Например:

```
(undercloud) [stack@undercloud ~]$ sudo cat /var/lib/kolla/config_files/keystone.json
{
  "command": "/usr/sbin/httpd",
  "config_files": [
    {
      "dest": "/etc/keystone/fernet-keys",
      "merge": false,
      "preserve_properties": true,
      "source": "/var/lib/kolla/config_files/src/etc/keystone/fernet-keys"
    },
    {
      "dest": "/etc/httpd/conf.d",
      "merge": false,
      "preserve_properties": true,
      "source": "/var/lib/kolla/config_files/src/etc/httpd/conf.d"
    },
    {
      "dest": "/",
      "merge": true,
      "preserve_properties": true,
      "source": "/var/lib/kolla/config_files/src/*"
    }
  ]
}
```

Рассмотрим подробнее:

- `command` - команда, которая будет запущена в контейнере
- `config_files` - список файлов, которые будут скопированы в контейнер из хост системы.
 - `dest` - путь внутри контейнера, куда будет скопирован файл
 - `merge` - если `true`, то файлы будут скопированы в директорию `dest`, если `false` - то весь каталог `source` будет скопирован в `dest`
 - `preserve_properties` - если `true`, то будут сохранены права доступа к файлу и владелец
 - `source` - путь внутри контейнера, который будет скопирован в `dest`.

Отдельно хочу выделить путь `/var/lib/kolla/config_files/src/*`. Для контейнера `keystone` - это директория `/var/lib/config-data/`

`puppet-generated/keystone`. Для других контейнеров это путь `/var/lib/config-data/puppet-generated/$service_name`. Это директория, в которой хранятся все файлы конфигурации необходимые для сервиса. Например:

```
(undercloud) [stack@undercloud ~]$ sudo tree /var/lib/config-data/puppet-generated/keystone
/var/lib/config-data/puppet-generated/keystone
|-- etc
|   |-- httpd
|   |   |-- conf
|   |   |   |-- httpd.conf
|   |   |   |-- ports.conf
|   |   |-- conf.d
|   |       |-- 10-keystone_wsgi.conf
<...>
|   |-- keystone
|   |   |-- credential-keys
|   |   |   |-- 0
|   |   |   |-- 1
|   |   |-- fernet-keys
|   |   |   |-- 0
|   |   |   |-- 1
|   |   |-- keystone.conf
|   |-- my.cnf.d
|       |-- tripleo.cnf
|-- var
|   |-- www
|       |-- cgi-bin
|           |-- keystone
|               |-- keystone
```

После запуска контейнера команда `kolla_start` копирует все вышеуказанные файлы в / системы контейнера и запускает команду `/usr/sbin/httpd`.

При этом сами файлы в папке `/var/lib/config-data/puppet-generated/` генерируются **puppet** манифестами также запускаемыми в контейнерах. Посмотреть на эти контейнеры можно через

```
sudo podman ps -a | grep container-puppet
```

Обратите внимание, что некоторые сервисы запускаются через папку `/var/lib/config-data/`, например, *barbican*, но это редкий случай.

Сети узла развёртывания

(Далее описание ведётся с узла развёртывания)

Введите в консоли команду

```
sudo ovs-vsctl show
```

Результат выполнения будет выглядеть следующим образом

```
59a6160b-9679-46ba-a583-63a64d0c7f4d
  Manager "tcp:6640:127.0.0.1"
    is_connected: true
  Bridge br-ctlplane
    Controller "tcp:127.0.0.1:6633"
      is_connected: true
    fail_mode: secure
    datapath_type: system
    Port "enp1s0"
      Interface "enp1s0"
    Port phy-br-ctlplane
      Interface phy-br-ctlplane
        type: patch
        options: {peer=int-br-ctlplane}
    Port br-ctlplane
      Interface br-ctlplane
        type: internal
  Bridge br-int
    Controller "tcp:127.0.0.1:6633"
      is_connected: true
    fail_mode: secure
    datapath_type: system
    Port int-br-ctlplane
      Interface int-br-ctlplane
        type: patch
        options: {peer=phy-br-ctlplane}
    Port "tapde8cbd01-25"
      tag: 1
      Interface "tapde8cbd01-25"
        type: internal
```

(continues on next page)

(продолжение с предыдущей страницы)

```

Port br-int
  Interface br-int
    type: internal
  ovs_version: "2.12.0"

```

Вывод указывает, что существует два связанных между собой бриджа OVS: `br-ctlplane` и `br-int`. Их также можно увидеть при выводе команды `ip a`. Бридж `br-int` не имеет в своём основании никакого физического интерфейса, зато имеет виртуальный интерфейс `tapde8cbd01-25`. Расскажем о нём позднее.

Бридж `br-ctlplane` имеет физический интерфейс `enp1s0` и связывает через него с внешней сетью бридж `br-int`.

При выводе `ip a` не указан никакой интерфейс `tap`. Попробуем выполнить следующее

```

[stack@undercloud ~]$ sudo ip netns
qdhcp-a4d98b54-1bc4-402a-92c1-51cbeee94aa3 (id: 0)

```

Значит, что в системе существует выделенное сетевое пространство имён. Подробнее про них можно почитать [здесь](#)

```

[stack@undercloud ~]$ sudo ip netns exec qdhcp-a4d98b54-1bc4-402a-92c1-51cbeee94aa3 ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
8: tapde8cbd01-25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
   link/ether fa:16:3e:1e:c1:8b brd ff:ff:ff:ff:ff:ff
   inet 192.168.24.36/24 brd 192.168.24.255 scope global tapde8cbd01-25
       valid_lft forever preferred_lft forever
   inet6 fe80::f816:3eff:fe1e:c18b/64 scope link
       valid_lft forever preferred_lft forever

```

Мы видим знакомый нам `tap` интерфейс с адресом `192.168.24.36/24`. Назначение этого адреса становится понятно, если поискать этот порт среди портов `openstack`

```

(undercloud) [stack@undercloud ~]$ openstack port list | grep 192.168.24.36
| de8cbd01-25f3-4ebd-8965-74c98eca45bd |          | fa:16:3e:1e:c1:8b | ip_address='192.168.24.36', subnet_id='ab41b2bc-f27c-43a6-977e-4bd93d0c1cc7' |''
  ACTIVE |
(undercloud) [stack@undercloud ~]$ openstack port show de8cbd01-25f3-4ebd-8965-74c98eca45bd -c binding_vif_details -c binding_vif_type -c device_
owner

```

(continues on next page)

(продолжение с предыдущей страницы)

Field	Value
binding_vif_details	bridge_name='br-int', connectivity='l2', datapath_type='system', ovs_hybrid_plug='True', port_filter='True'
binding_vif_type	ovs
device_owner	network:dhcp

Мы видим, что этот порт является IP адресом DHCP сервера в сети ctlplane, поднятый на узле развёртывания в отдельном сетевом пространстве имён. Если использовать имя сетевого пространства имён и выполнить

```
[stack@undercloud ~]$ sudo podman ps | grep qdhcp-a4d98b54-1bc4-402a-92c1-51cbeee94aa3
```

То вы увидите контейнер, который отвечает за DHCP сервис в сети Ctlplane.

1.9 Дополнительно

1.9.1 Лицензирование

Выпуск лицензионного ключа

Инструментом, определяющим легальное использование и распространение программного обеспечения Acloud («Отечественная облачная платформа 1.0») является лицензионный ключ. Лицензионный ключ разрешает получателю использовать функционал программного продукта в полной мере на заданный срок. По истечению срока действия лицензионного ключа функционал программного продукта ограничивается, но информация и данные, созданные во время срока действия лицензии сохраняются. Для восстановления функциональности достаточно установить актуальный лицензионный ключ программного продукта.

Выпуск лицензионного ключа осуществляется правообладателем в закрытом контуре.

Распространение лицензионных ключей производится путем передачи файлов напрямую заказчикам. Для получения лицензионного ключа необходимо обратиться по почте info@acloud.ru.

Активация лицензионного ключа

Для активации лицензии необходимо поместить предоставленный файл лицензионного ключа (файл с названием `license`) в папку `/etc/aos`.

Установка и обновление лицензионных ключей лицензируемых модулей

Лицензируемые модули:

- CloudManager;
- Monitor;
- RServer.

Порядок действий:

1. Выпустить лицензионный ключ.
2. Поместить полученный файл лицензии в папку `/etc/aos` способом, подходящим типу вашего облака.

2.1 Интерфейс консоли

2.1.1 Главная

На вкладке «Главная» отображается информация о пользователе и сведения о квотах на вычислительные ресурсы, ресурсы хранилища, сетевые ресурсы т.д.

2.1.2 Вычисления

В разделе «Вычисления» отображаются следующие вкладки:

- **Инстансы.** Доступны следующие действия: просмотр, запуск, удаление, создание снимка, подключение или отключение интерфейса, подключение или отключение диска, присоединение плавающего IP, управление группой безопасности, остановка, пауза, блокировка, отправка в архив, приостановка, жесткая или мягкая перезагрузка инстансов, изменение меток инстансов или подключение через VNC, клонирование инстансов.

- Снимки инстансов. Доступны следующие действия: просмотр, редактирование, удаление снимков инстансов, запуск инстансов или создание дисков на их основе.
- Типы инстансов. Доступны следующие действия: Просмотр типов инстансов.
- Группы инстансов. Доступны следующие действия: Просмотр, создание или удаление групп инстансов.
- Образы. Доступны следующие действия: Просмотр, создание, редактирование, удаление образов, запуск инстансов или создание дисков на их основе.
- Ключевые пары. Доступны следующие действия: Просмотр, создание, редактирование, импорт и удаление ключевых пар.

2.1.3 Хранилище

В разделе «Хранилище» отображаются следующие вкладки:

- Диски. Доступны следующие действия: просмотр, создание, редактирование и удаление дисков. Создание моментального снимка диска, создание резервной копии диска, клонирование диска, расширение диска или изменение типа диска, присоединение или отсоединение их к виртуальной машине.
- Резервные копии дисков. Доступны следующие действия: просмотр, создание, редактирование и удаление резервных копий дисков. Также создание дисков из них и восстановление резервной копии.
- Моментальные снимки дисков. Доступны следующие действия: просмотр, создание, редактирование, удаление моментальных снимков дисков. И создание дисков из них.

2.1.4 Сеть

В разделе «Сеть» отображаются следующие вкладки:

- Сети. Доступны следующие действия: просмотр, создание, редактирование и удаление сетей.
- Порты. Доступны следующие действия: просмотр, создание, редактирование, удаление портов и управление группой безопасности для портов.
- Маршрутизаторы. Доступны следующие действия: просмотр, создание, редактирование, удаление и управление маршрутизаторами.
- Плавающие IP-адреса. Доступны следующие действия: выделение IP-адресов или их освобождение.
- Топология. Доступны следующие действия: просмотр топологии сети.
- Группы безопасности. Доступны следующие действия: просмотр, создание, редактирование и удаление групп безопасности и правил групп безопасности.

2.1.5 Оркестрация

В разделе «Оркестрация» отображается вкладка:

- Стеки. Доступны следующие действия: просмотр, создание, редактирование и удаление стеков.

2.1.6 Планировщик

В разделе «Планировщик» отображаются следующие вкладки:

- Планировщик. Доступны следующие действия: запланировать новую задачу, просмотр детальной информации о задачах, удаление, повторение задач.

2.1.7 VDI

В разделе «VDI» отображаются следующие вкладки:

- VDI виртуальные машины. Доступны следующие действия: просмотр, запуск, назначение пользователя.

2.2 Страница пользователя

2.2.1 Данные пользователя

На странице отображается следующая информация: имя пользователя, телефон, идентификатор пользователя, электронная почта, настоящее имя.

2.2.2 Учетные данные приложения

На странице отображается список учетных данных приложений, созданных текущим пользователем. Учетные данные приложения предоставляют способ делегировать авторизацию пользователя приложению без совместного использования аутентификации пароля пользователя. Это полезная мера безопасности, особенно в ситуациях, когда идентификация пользователя предоставляется внешним источником, таким как LDAP или служба единого входа. Вместо того, чтобы хранить пароли пользователей в файлах конфигурации, пользователь создает учетные данные приложения для определенного проекта со всеми или подмножеством назначений ролей, которые у него есть в этом проекте, а затем сохраняет идентификатор учетных данных приложения и секрет в файле конфигурации.

Доступны следующие действия: просмотр, создание, удаление.

2.3 Режим администратора

2.3.1 Главная

На вкладке «Главная» отображается основная информация о платформе (количество проектов, пользователей, узлов), использование виртуальных ресурсов (использование ЦП, использование памяти), состояние вычислительных и сетевых служб.

2.3.2 Вычисления

В разделе «Вычисления» отображаются следующие вкладки:

- **Инстансы.** Доступны следующие действия: просмотр, остановка, приостановка, обычная или живая миграция, мягкая или жесткая перезагрузка и удаление инстансов, которые принадлежат всем проектам, также просмотр журнала инстанса или подключение через VNC.
- **Снимки инстансов.** Доступны следующие действия: просмотр, редактирование, удаление снимков инстансов.
- **Типы инстансов.** Доступны следующие действия: просмотр, создание, редактирование, управление метаданными и удаление типов инстансов.
- **Группы инстансов.** Доступны следующие действия: просмотр, создание или удаление групп инстансов.
- **Образы.** Доступны следующие действия: просмотр, создание, редактирование, управление метаданными и удаление образов.

Вкладка «Гипервизоры»

Вкладка дает краткую информацию о гипервизорах и вычислительных узлах.

Также представлены две внутренние вкладки: **«Гипервизор»**, **«Хосты вычислений»**.

Вкладка «Гипервизоры»

Отображает перечень доступных гипервизоров.

Таблица 1: Списанием представлена следующая информация:

Наименование поля	Описание
ID/Имя	Идентификатор или имя гипервизора. Задается при его добавлении. Также является ссылкой для перехода к детальной информации о данном гипервизоре.
Тип	Тип гипервизора.
VCPU (Ядер)	Количество ядер VCPU.
Настроенная память (ГБ)	Количество памяти (ГБ).
Инстансы	Количество инстансов гипервизора. Когда гипервизор выключен, общее количество инстансов гипервизора может включать в себе те инстансы, которые были мигрированы или эвакуированы с гипервизора перед выключением. С информацией о количестве мигрированных или эвакуированных с гипервизора инстансов можно ознакомиться, перейдя на страницу детальной информации о гипервизоре. При повторном включении гипервизора счетчик инстансов обновится.

Для списка гипервизоров доступны инструменты сортировки и фильтрации. Поля сортируются по возрастанию и убыванию. Фильтрация производится по следующим параметрам:

- Имя хоста - Имя гипервизора. Допустим неполный ввод имени;
- Тип - Тип гипервизора. Допустим неполный ввод.

Также пользователь может просмотреть детальную информацию о гипервизоре.

Детальная информация о гипервизоре представлена в нескольких внутренних вкладках.

Вкладка «Обзор»

Отображает подробную информацию о выбранном гипервизоре.

Действия над гипервизорами

Таблица 2: Для гипервизоров в зависимости от статуса доступны следующие действия:

N	Действие	Описание
1	Управление устройством питания	Назначение или снятие назначения устройства управления питанием гипервизору.

Особенности работы

- *Управление устройством питания*

Управление устройством питания

Доступно в общем списке гипервизоров. После вызова действия в открывшемся окне выводится перечень доступных устройств управления питанием:

Управление устройством питания



Примечание

Добавьте устройство управления питанием в гипервизор или удалите устройство из него.

Имя гипервизора

ana-ubt22a-std-10-node1

Устройства управления питанием

Нажмите здесь для фильтров.

Имя	Тип	Протокол	Порты	Занятые порты
<input checked="" type="radio"/> new	SupermicroRackDevice	ipmi	623	0

< 1 >

Выбран: new x

Отмена

OK

Рис. 1: Окно назначения устройства управления питанием гипервизору

Выберите необходимое устройство и убедитесь, что у него есть свободный порт. Также в данном окне можно снять назначение устройства управления питанием.

Завершите процедуру кнопкой «ОК».

- Агрегаторы хостов. Доступны следующие действия: просмотр, создание, управление метаданными, редактирование и удаление агрегаторов хостов, просмотр списка зон доступности.

2.3.3 Хранилище

В разделе «Хранилище» отображаются следующие вкладки:

- Диски. Доступны следующие действия: просмотр, обновление статуса, миграция и удаление дисков.
- Резервные копии дисков. Доступны следующие действия: просмотр и удаление резервной копии восстановления.
- Моментальные снимки дисков. Доступны следующие действия: просмотр и удаление моментальных снимков дисков.
- Типы дисков. Доступны следующие действия: просмотр, создание, редактирование, шифрование, управление доступом и удаление типов дисков.
- Бэкенды хранения. Доступны следующие действия: просмотр бэкендов хранения.

2.3.4 Сеть

В разделе «Сеть» отображаются следующие вкладки:

- Сети. Доступны следующие действия: просмотр, создание и удаление сетей.
- Порты. Доступны следующие действия: просмотр и удаление портов.
- Маршрутизаторы. Доступны следующие действия: просмотр и удаление маршрутизаторов.
- Плавающие IP-адреса. Доступны следующие действия: выделение IP-адресов или их освобождение.
- Группы безопасности. Доступны следующие действия: просмотр и удаление групп безопасности.

2.3.5 Идентификация

В разделе «Идентификация» отображаются следующие вкладки:

- Домены. Доступны следующие действия: просмотр, создание, редактирование, включение, отключение и удаление доменов.
- Проекты. Доступны следующие действия: просмотр, создание, редактирование, включение, отключение и удаление проектов. Также управление пользователями или группами пользователей проектов, изменение тегов для них.
- Пользователи. Доступны следующие действия: просмотр, создание, редактирование, включение, отключение, удаление пользователей. И редактирование системных разрешений пользователей, обновление пароля пользователя.
- Группы пользователей. Доступны следующие действия: просмотр, создание, редактирование и удаление групп пользователей.
- Роли. Доступны следующие действия: просмотр, создание, редактирование и удаление ролей.

2.3.6 Центр мониторинга

В разделе «Центр мониторинга» отображаются следующие вкладки:

- Обзор мониторинга. Просмотр общей информации о мониторинге.
- Физические узлы. Просмотр общей информации о физических узлах.
- Кластеры хранилища. Просмотр общей информации о кластерах хранения.
- Сервисы OpenStack. Просмотр общей информации о сервисах OpenStack.
- Другие сервисы. Просмотр общей информации об остальных службах.

2.3.7 Acloud центр

В разделе «Acloud центр» отображаются следующие вкладки:

- VDI. Вкладка состоит из двух внутренних вкладок:
 - VDI виртуальные машины. Доступны следующие действия: просмотр, создание, редактирование, удаление VDI машин, перестраивание VDI машин.
 - VDI проекты. Доступны следующие действия: создание, редактирование, удаление VDI проектов, получение информации о машинах VDI проекта, распределение машин VDI проекта, управление пользователями и группами пользователей VDI проектов.
- Балансировка. Вкладка состоит из двух внутренних вкладок:

- Вычислительные узлы. Отображается список вычислительных узлов, для которых разрешена балансировка. Доступно действие по запуску балансировки инстансов между вычислительными узлами.
- Агрегаторы узлов. Отображается список агрегаторов узлов, для которых разрешена балансировка. Доступно действие по запуску балансировки инстансов.
- Инфраструктура. В подразделе имеется одна вкладка:

Вкладка «Устройства управления питанием»

Примечание: Вкладка доступна только пользователю с правами администратора.

На вкладке доступен функционал по управлению устройствами управления питанием.

Устройства управления питанием

Создать УУП Удалить

Несколько фильтров разделяются Enter

Обновление данных: 00

Имя	ID	Тип	Протокол	Порты	Действие
new	2	SupermicroRackDevice	ipmi	623	Редактировать

Всего 1 элементов < 1 > 10 / стр. v

Мичман
Удобный сервис для упрощённой настройки виртуальных машин
Перейти

Биллинг
Документация

Рис. 2: Список устройств управления питанием

Таблица 3: Списанием представлена следующая информация:

Наименование поля	Описание
Имя	Наименование устройства управления питанием, присваивается при создании.
ID	Идентификационный номер устройства управления питанием.
Тип	Тип устройства, задается при создании.
Протокол	Тип протокола, задается при создании.
Порты	Номер задействованного порта устройства.

Для списка доступны инструменты сортировки и фильтрации. Поля сортируются по возрастанию и убыванию.

Особенности работы

- *Добавление нового устройства управления питанием*
- *Редактирование устройства управления питанием*
- *Удаление устройства управления питанием*

Добавление нового устройства управления питанием

В общем списке на панели управления кнопкой «Создать УПП» открываем мастер окна создания нового устройства:

Создать УУП ✕

* Тип протокола

Наименование устройства
Если поле не заполнено, имя объекта будет сгенерировано автоматически. Максимальное количество символов 50

* Имя хоста или IP-адрес

* Логин
Имя должно начинаться с заглавной буквы, строчной буквы и состоять из 1 до 128 символов, символы могут содержать только "0-9, a-z, A-Z, "-", "_".

* Пароль

* Порт
Диапазон от 1 до 65535

Рис. 3: Окно создания устройства управления питанием

В открывшемся окне указываем:

- Тип протокола - выбор из доступных типов протоколов;
- Тип устройства - выбор из доступных типов устройств, соответствующих выбранному протоколу;
- Наименование устройства - доступно произвольное наименование. Если поле не заполнено, имя объекта будет сгенерировано автоматически. Максимальное количество символов 50;
- Имя хоста или IP-адрес - имя хоста или IP-адрес устройства;
- Логин - логин для подключения к устройству;
- Пароль - пароль для подключения к устройству;
- Порт - порт устройства управления питанием. Диапазон значений от 1 до 65535. Значения по умолчанию:
 - * для протокола **ipmi** - порт 623;
 - * для протокола **snmp** - порт 161;
 - * для протокола **modbus** - порт 502;
 - * для протокола **intel_amt** - порт 16992;
 - * для протокола **ssh** - порт 22.

Важно: Между типами протоколов и устройств существует жесткая связь:

Тип протокола	Тип устройства
modbus	ET7067
snmp	DaenetIP2
ssh	SshDevice
intel_amt	IntelAMT
ipmi	SupermicroRackDevice

Завершаем процедуру создания кнопкой «ОК». После чего корректно созданное устройство управления питанием отобразится в общем списке. В противном случае появится сообщение об ошибке с указанием причин невозможности создания устройства.

Редактирование устройства управления питанием


Функция позволяет редактировать параметры выбранного устройства управления питанием. Доступна в общем списке всех устройств управления питанием. После вызова действия в открывшемся окне задайте необходимые параметры:

Редактировать устройство управления питанием ✕

Наименование устройства
Если поле не заполнено, имя объекта будет сгенерировано автоматически. Максимальное количество символов 50

* Имя хоста или IP-адрес

* Логин
Имя должно начинаться с заглавной буквы, строчной буквы и состоять из 1 до 128 символов, символы могут содержать только "0-9, a-z, A-Z, "- '()[]".

* Пароль 

* Порт
Диапазон от 1 до 65535

Рис. 4: Окно изменения параметров устройства управления питанием

Для применения новых параметров необходимо ввести корректный пароль устройства управления питанием.

Завершите процедуру кнопкой подтверждения.

Удаление устройства управления питанием

Действие по удалению устройства доступно в общем списке всех устройств управления питанием. Также действие по удалению можно выполнить в отношении группы предварительно выделенных устройств.

Также на вкладке доступно действие по загрузке списка существующих устройств управления питанием в формате .csv, а также обновление данных на странице.

- Планировщик. Доступны следующие действия: запланировать новую задачу, просмотр детальной информации о задачах, удаление, повторение задач.
- Модули. Просмотр информации о модулях, версиях модулей, статусе и лицензиях.

2.3.8 Глобальные настройки

В разделе «Глобальные настройки» отображается информация о системе на нескольких вкладках:

- Службы: просмотр списка служб.
- Вычислительные службы. Доступны следующие действия: просмотр списка всех вычислительных служб и включение или отключение их.
- Сетевые агенты. Доступны следующие действия: просмотр сетевых агентов и включение или отключение их.
- Блочные службы хранения. Доступны следующие действия: просмотр списка всех блочных служб хранения и включение или отключение их.
- Оркестровочные службы. Доступны следующие действия: просмотр списка всех служб оркестровки.
- Конфигурация системы. Доступны следующие действия: просмотр, редактирование и сброс конфигурации системы.
- Определения метаданных. Доступны следующие действия: просмотр, редактирование и удаление определений метаданных системы.

2.4 Мичман

Мичман — система оркестрации для разворачивания платформенных сервисов на основе описаний на специализированном языке для описания топологии и оркестрации облачных приложений и сервисов TOSCA (Topology and Orchestration Specification for Cloud Applications).

Перейдите по адресу веб-интерфейса Мичман и авторизуйтесь в системе:

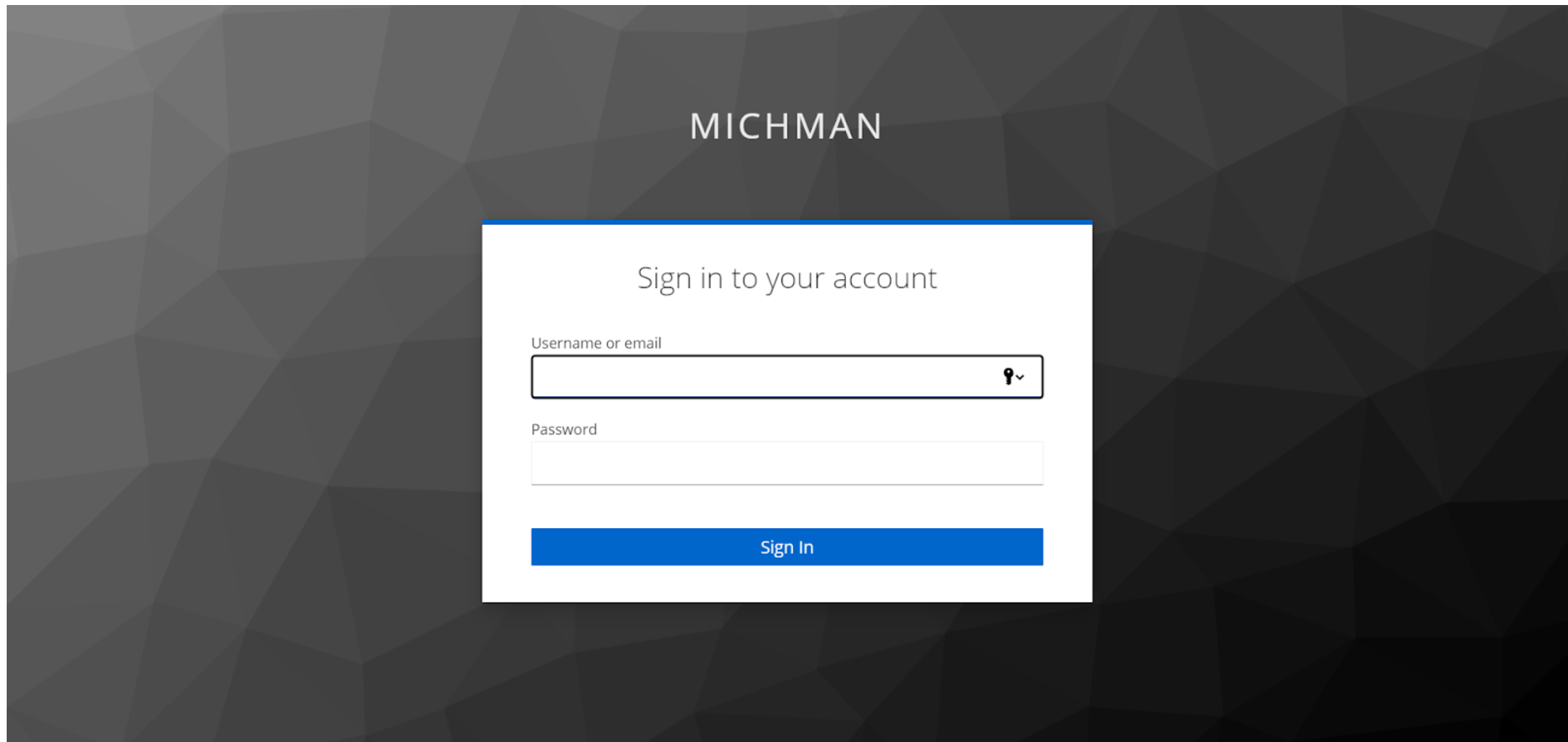


Рис. 5: Авторизация в веб-интерфейсе

Интерфейс Мичман состоит из 3 основных вкладок:

2.4.1 Templates

На вкладке Templates представлен список шаблонов на языке TOSCA для запуска *сервисов (облачных приложений и сервисов)*, а также инструменты для работы с шаблонами. Данные отображаются в плиточном виде.

The screenshot shows the 'Templates' page in the MICHMAN interface. The navigation bar at the top includes 'Templates', 'Instances', and 'Providers' tabs, along with a user profile 'admin'. Below the navigation bar, there is a section for 'Incomplete instances' with a search bar and a 'Create instance from config' button. The main content area displays four template cards:

- Apache Cassandra** (author: tsirleo, version: 1.0.0): Apache Cassandra is an open source NoSQL distributed database. Documentation URL: <https://cassandra.apache.org/doc/latest>
- BareMetal cluster** (author: t_ozeroва, version: 1.0.0): This template allows to create a new BareMetal cluster.
- BareMetal compute** (author: t_ozeroва, version: 1.0.0): This template allows to create a new BareMetal compute. If all baremetal hosts are described. add new baremetal cluster
- BareMetal network** (author: t_ozeroва, version: 1.0.0): This template allows to create a new BareMetal network. Next step is to create a part of the baremetal host in the network

Рис. 6: Вкладка Templates

Template - это запись в оркестраторе, соответствующая yaml-файлу в формате TOSCA с описанием параметров запуска и предварительной настройки *сервисов*.

Для каждого шаблона отображается следующая информация:

- Имя шаблона;
- Имя автора шаблона;
- Версия шаблона;
- Краткое описание.

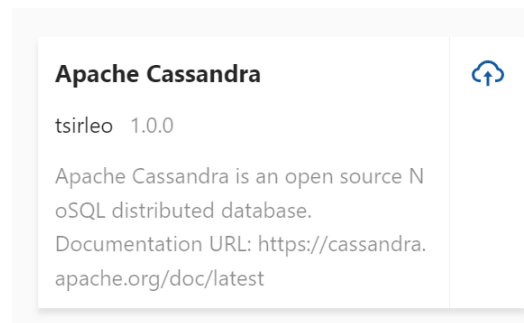


Рис. 7: Детали шаблона

При клике по описанию во всплывающем блоке открывается полное описание.

В верхней части рабочей области находится список сервисов (экземпляров шаблонов), для которых был начат, но не завершен процесс создания. По умолчанию список незавершенных *сервисов* раскрыт, но может быть скрыт:

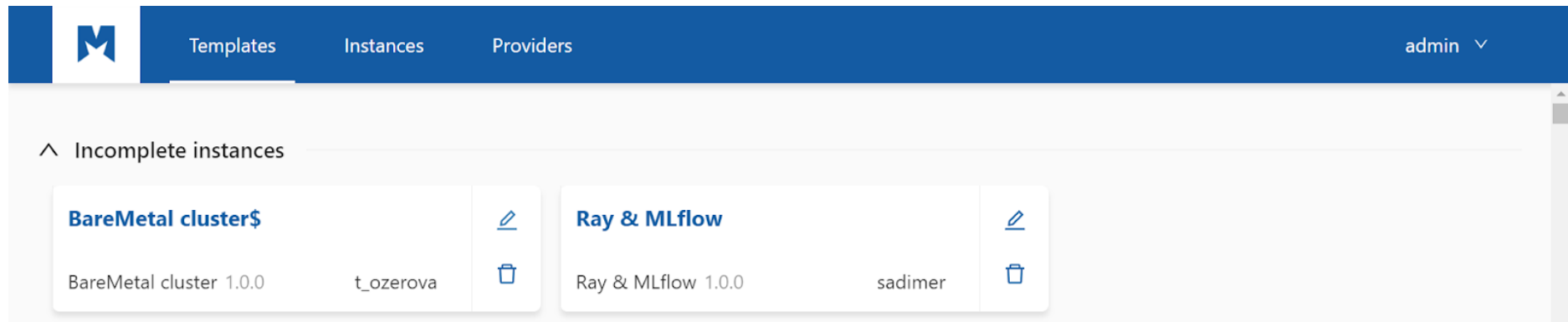


Рис. 8: Список незавершенных сервисов

Для каждого *сервиса* отображается следующая информация:

- Имя сервиса;
- Имя и версия шаблона, использованного для запуска;
- Имя автора шаблона.

Для незавершенных *сервисов* доступны следующие действия:

- Редактирование параметров сервиса (при клике по имени также открывается форма редактирования);
- Удаление сервиса.

Управление шаблонами

По умолчанию на странице расположены все имеющиеся шаблоны. В правой части рабочей области имеется фильтр по тегам шаблонов:

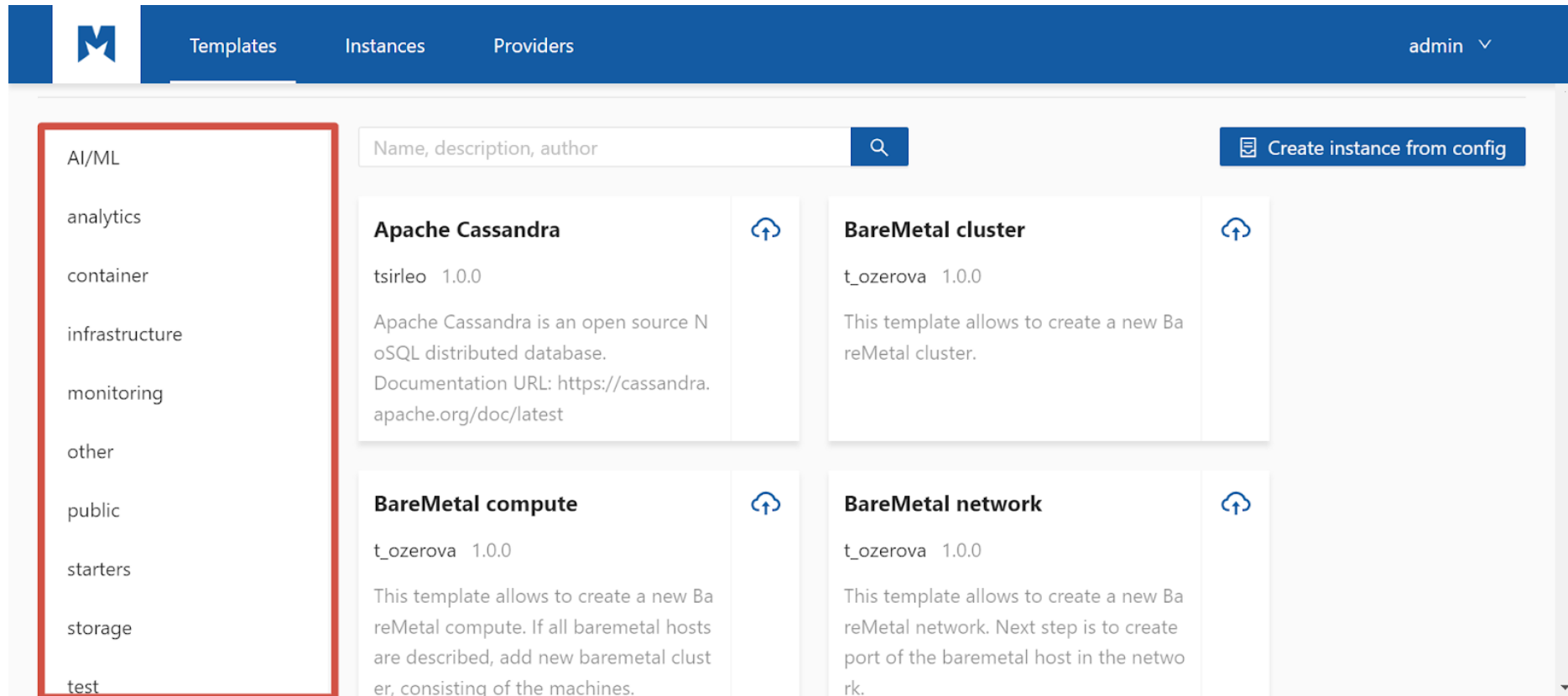


Рис. 9: Фильтрация шаблонов

В список тегов автоматически попадают все теги, указанные авторами шаблонов. Сейчас используются следующие теги:

- AI/ML - шаблоны создания сервисов для работы с искусственным интеллектом и машинным обучением (Jupyter Hub, MLflow, Ray);
- analytics - шаблоны создания сервисов для аналитики (OpenSearch, OpenSearch with dashboard);

- container - шаблоны для разворачивания сервисов для работы с контейнерами(CRI host, Docker container, Kubernetes cluster);
- infrastructure - шаблоны создания облачной инфраструктуры (DCIM, Docker container, OpenStack compute, Yandex compute и т.д.);
- monitoring - шаблоны создания системы мониторинга (Prometheus);
- other - шаблоны создания сервисов, не включенных в другие типы (NTP, OpenFOAM);
- public - группа, объединяющая все шаблоны, доступные для запуска (DCIM, Docker container, BareMetal cluster, ClickHouse, CouchDB, MinIO и т.д). Помимо них есть “служебные” шаблоны, не предполагающие непосредственного запуска пользователями;
- starters - шаблоны для развёртывания виртуальных машин с инструментами разработки на различных языках (custom software, Java, Python и т.д.);
- storage - шаблоны для разворачивания систем хранения данных (Apache Cassandra, CouchDB, GlusterFS, MariaDB и т.д.);
- test - тестовые шаблоны (Mock OpenStack template, различные тестовые шаблоны TOSCA и т.д).

Запуск сервиса из шаблона

При нажатии на кнопку запуска открывается окно запуска *сервиса* из данного шаблона. На первом этапе выберите провайдер, в рамках которого будут создаваться ресурсы (провайдер может быть пустым):

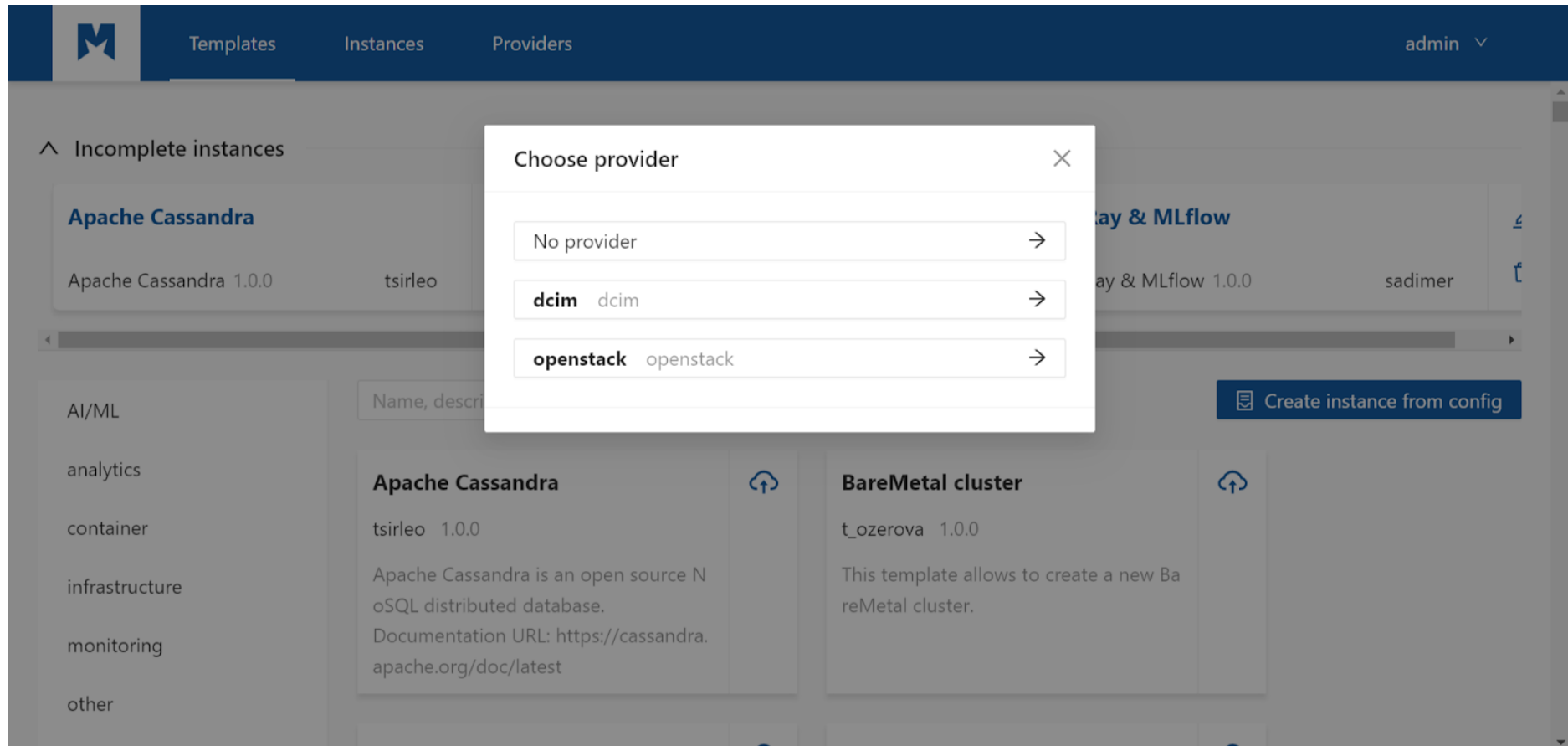


Рис. 10: Выбор провайдера

Далее укажите остальные параметры сервиса (облачного приложения).

При необходимости измените имя сервиса. По умолчанию в качестве имени указано имя шаблона.

В разделе Inputs настройте входные параметры для создания *сервиса*.

Ниже расположен блок выбора ресурсов, необходимых для запуска шаблона (виртуальная машина, другой сервис, хост и т.д.).

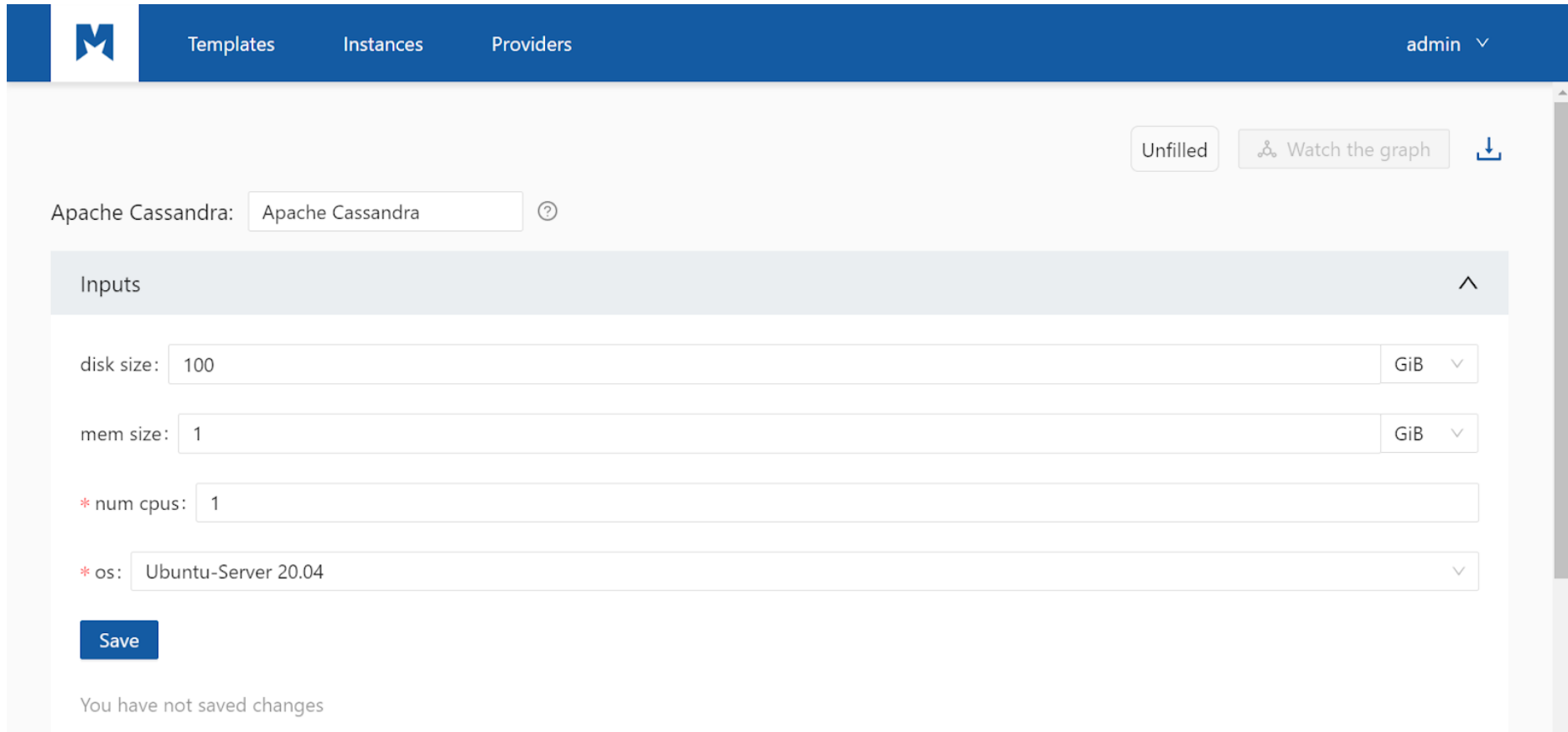
The screenshot displays the Acloud console interface. At the top, there is a navigation bar with tabs for 'Templates', 'Instances', and 'Providers', and a user profile 'admin' with a dropdown arrow. Below the navigation bar, there are three buttons: 'Unfilled', 'Watch the graph', and a download icon. The main content area shows a service configuration for 'Apache Cassandra'. A text input field contains 'Apache Cassandra' with a help icon. Below this is a section titled 'Inputs' with a checkmark and a dropdown arrow. Underneath, there are two resource selection cards. The first card is titled 'cassandra' and shows a selected resource 'cassandra-server' with an 'Initial' button, and another resource 'cassandra-server 1.0.0' with the user 'tsirleo'. The second card is titled 'cassandra-host' and features a large blue plus sign in a circle, indicating that no resources have been selected for this category.

Рис. 11: Выбор ресурсов

Перечень входных параметров для запуска сервиса отличается для каждого конкретного шаблона. Для большинства шаблонов требуется указать размер

диска VM, размер оперативной памяти, количество ЦПУ, используемую операционную систему и др.

Входные параметры для запуска машины из шаблона Apache Cassandra:



The screenshot shows the Acloud console interface for configuring a template. The top navigation bar includes 'Templates', 'Instances', and 'Providers', with the user 'admin' logged in. The main content area displays the 'Apache Cassandra' template configuration. At the top right, there are buttons for 'Unfilled', 'Watch the graph', and a download icon. Below this, the template name 'Apache Cassandra' is shown in a dropdown menu. A section titled 'Inputs' contains four configuration fields: 'disk size' (100 GiB), 'mem size' (1 GiB), '* num cpus' (1), and '* os' (Ubuntu-Server 20.04). A blue 'Save' button is located below the inputs, and a message at the bottom states 'You have not saved changes'.

Рис. 12: Входные параметры

Дополнительные ресурсы для запуска сервиса (облачного приложения) различаются в зависимости от шаблона. Например, для Apache Cassandra по умолчанию добавляется сервер СУБД (cassandra-server), также требуется добавить виртуальную машину, на которой будет запущен сервис (cassandra-host):

Для каждого доступного для добавления ресурса отображается следующая информация:

- Имя;
- Статус;
- Имя и версия шаблона;
- Имя пользователя, который создал ресурс.

При наведении курсора всплывают иконки выполнения действий над объектами:

- Просмотр графа;
- Редактирование объекта;
- Удаление объекта.

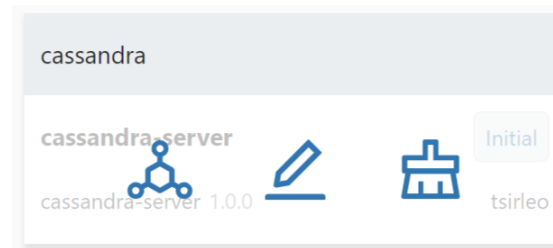


Рис. 13: Доступные действия

При создании вложенного ресурса отображается список подходящих шаблонов, из которых нужно выбрать необходимый, например, для выбора шаблона виртуальной машины:

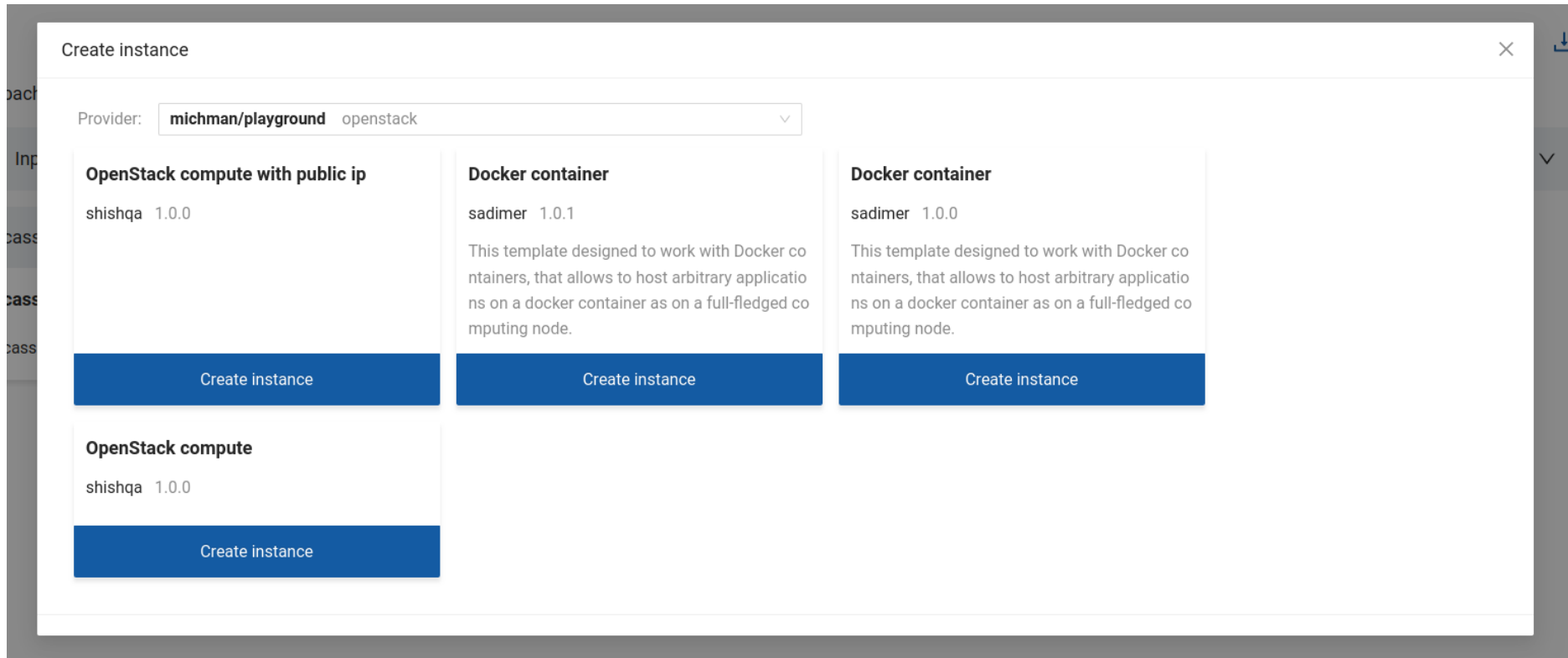


Рис. 14: Выбор шаблона

На вкладке просмотра графа помимо графики в блоке Settings отображается информация о статусе и настройках ресурса:

The screenshot displays the Acloud console interface for an instance named 'cassandra-server'. The top navigation bar includes 'Templates', 'Instances', and 'Providers', with the user 'admin' logged in. The breadcrumb trail shows 'Instances / Apache Cassandra / cassandra-server'. Action buttons for 'Copy instance', 'Edit the instance', and a download icon are visible. The main area shows a graph with a single node labeled 'initial' containing the text 'cassandra' and 'michman.nodes.cassandra.server'. A 'Settings' panel on the right lists various configuration parameters:

- Initial
- Provider: -
- michman deploy end:
- michman deploy start:
- michman instance author: 3608f0af-685e-4b5f-a704-e940bd31e96c
- michman instance id: 68565f98-0eb6-4aec-acd b-dc94d87d24f7
- michman instance immutable:
- michman instance name: cassandra-server
- michman template checksum: 51f05fe7790210

Рис. 15: Просмотр графа

Также в блоке Settings имеется возможность выполнения следующих действий:

- sync - синхронизация состояния сервиса;
- deploy - запуск сервиса.

После запуска сервиса становятся доступны действия:

- stop - остановка сервиса (перевод в состояние Initial);
- pause - приостановка (перевод в состояние Configured).

Для сервиса доступны следующие действия:

- Copy Instance - создание копии сервиса с аналогичными характеристиками;
- Edit the Instance - редактирование параметров или вложенных ресурсов сервиса. При выборе действия открывается страница редактирования сервиса;
- Download instance config - загрузка конфигурационного файла в формате JSON*.

В процессе загрузки отредактируйте имя файла и входные параметры машины:

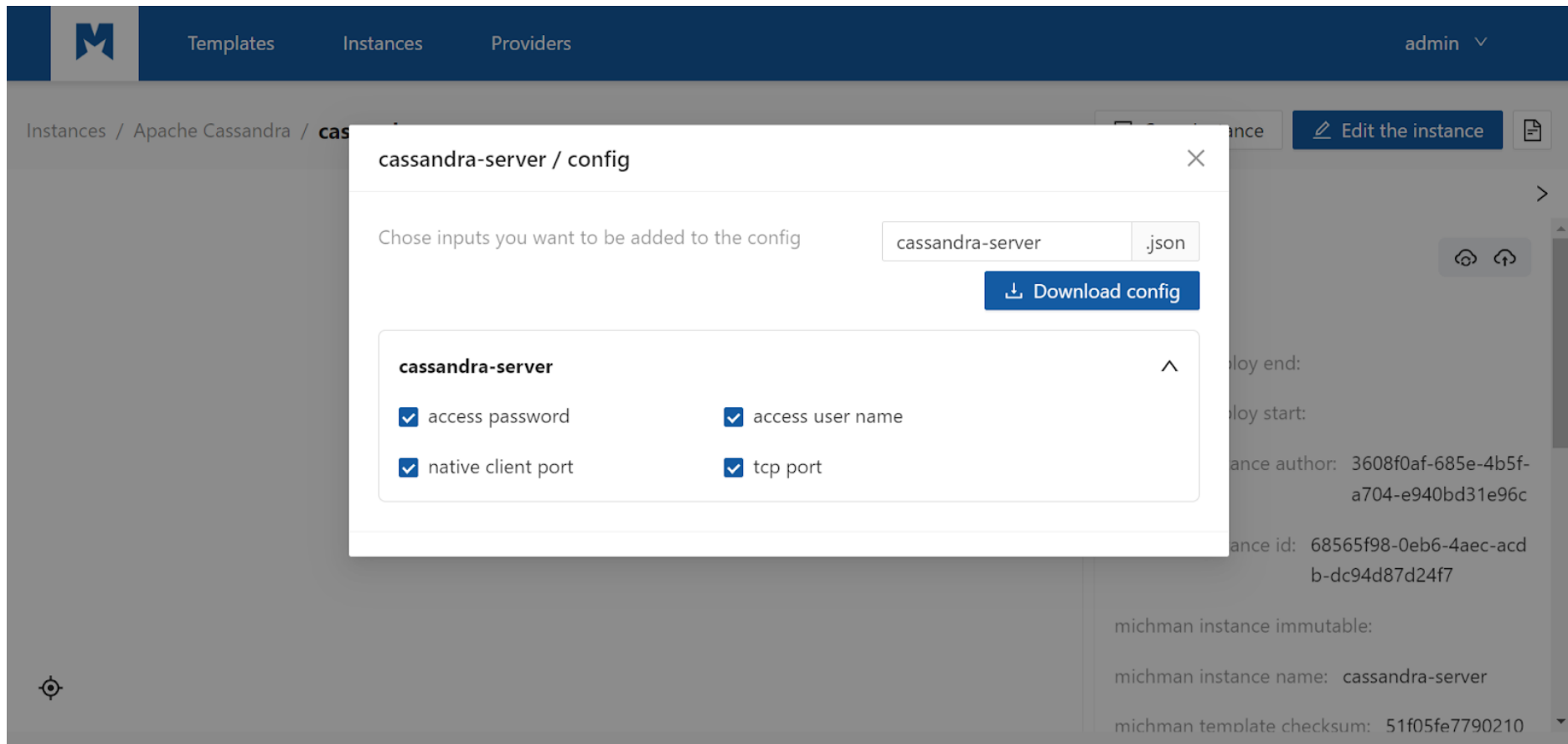


Рис. 16: Загрузка конфига

На вкладке редактирования объекта при необходимости измените имя машины и параметры подключения к ней:

← Apache Cassandra Initial Watch the graph

Apache Cassandra / cassandra-server: cassandra-server

Inputs

- * access password: dbpassword
- * access user name: admin
- * native client port: 9042
- * tcp port: 7000

Save

You have not saved changes

Рис. 17: Редактирование сервиса

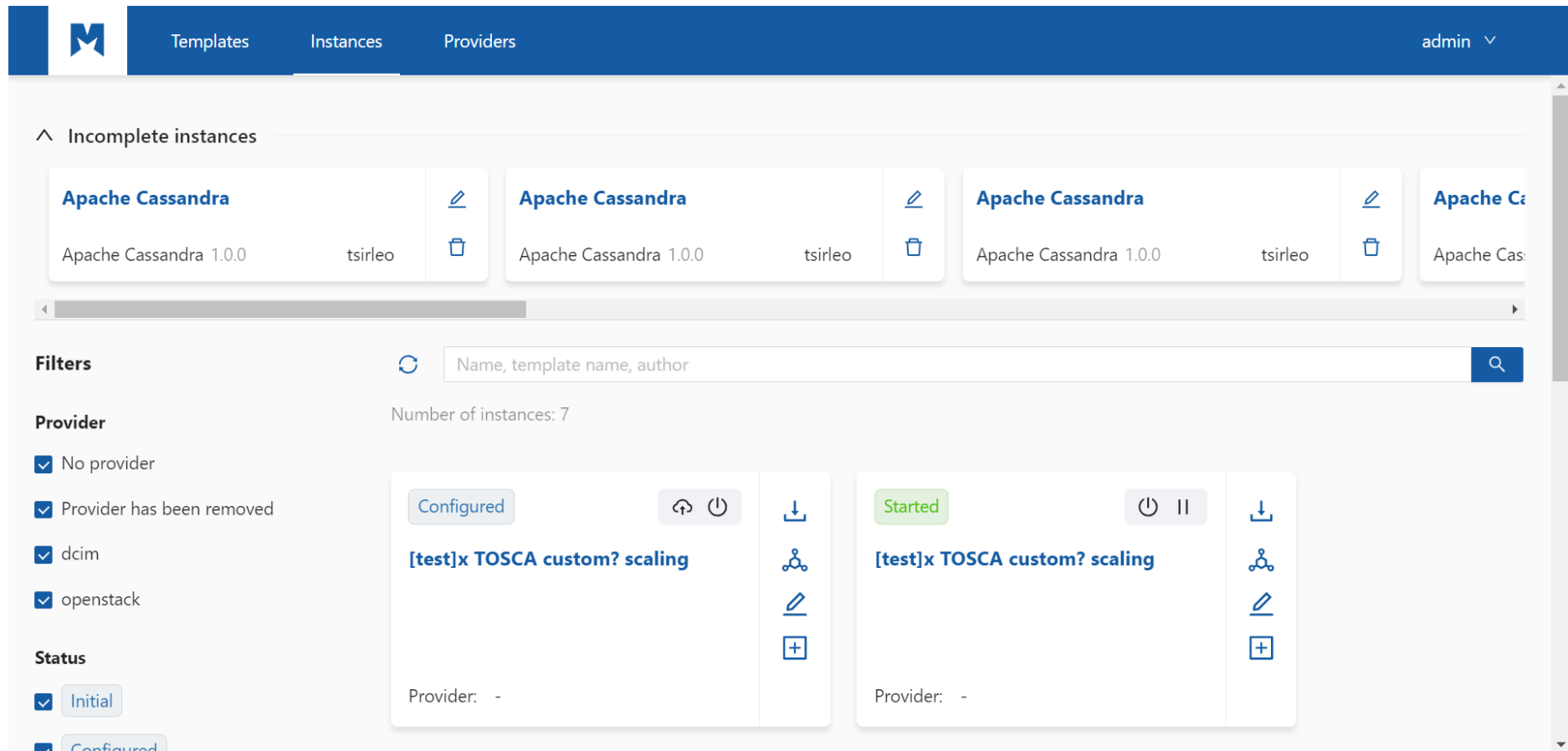
Из формы редактирования можно перейти на вкладку просмотра графа машины с помощью кнопки Watch the graph.

Запуск сервиса из конфигурационного файла

Дополнительно на вкладке Templates можно запустить виртуальную машину из конфиг файла с помощью кнопки Create instance from config. При нажатии на кнопку открывается окно выбора на компьютере пользователя JSON-файла с параметрами запуска машины.




2.4.2 Instances




На вкладке Instances представлен список запущенных *сервисов*, а также инструменты для работы с ними. Данные отображаются в плиточном виде.





admin ▾

^ Incomplete instances

Apache Cassandra  **Apache Cassandra**  **Apache Cassandra**  **Apache Ca**

Apache Cassandra 1.0.0 tsirleo  Apache Cassandra 1.0.0 tsirleo  Apache Cassandra 1.0.0 tsirleo  Apache Cas

Filters  Name, template name, author 

Provider Number of instances: 7

No provider

Provider has been removed




dcim




openstack

Status




Initial




Configured

Configured   

[test]x TOSCA custom? scaling   

Provider: -

Started   

[test]x TOSCA custom? scaling   

Provider: -

Рис. 18: Список сервисов

Для списка запущенных *сервисов* доступны инструменты поиска и фильтрации. Фильтрация производится по параметрам: Provider (на ресурсах какого провайдера запускаются сервисы) и Status (статус сервиса):

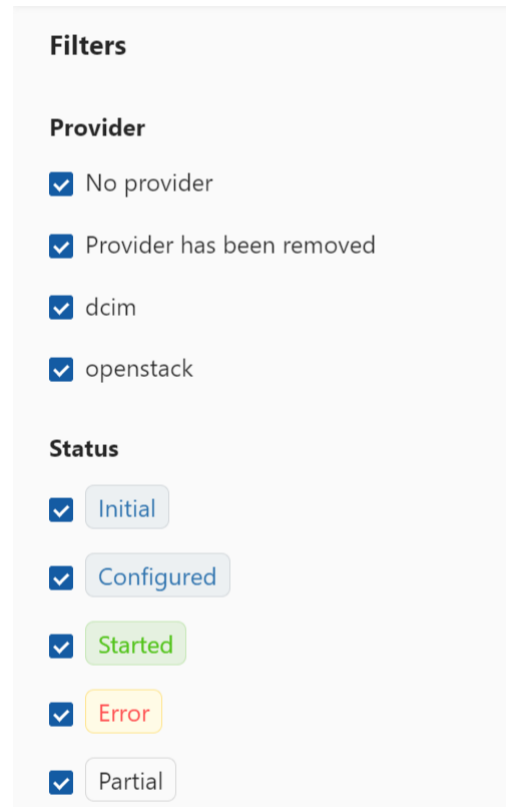


Рис. 19: Фильтры сервисов

Для *сервисов* отображается следующая информация:

- Статус;
- Имя;
- Провайдер.

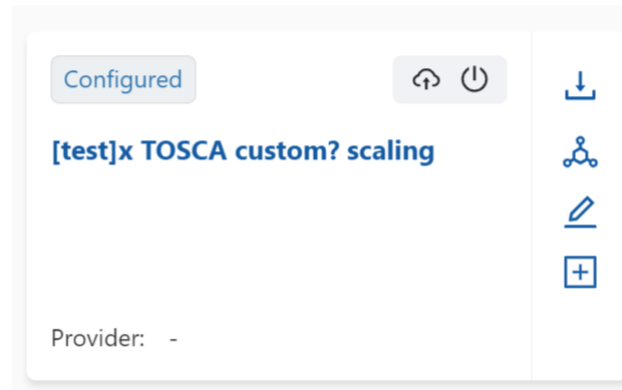


Рис. 20: Детальная информация

При клике по имени осуществляется переход к просмотру графа данного сервиса.

С помощью иконок можно вызвать следующие действия над объектами:

- Watch the graph - просмотр графа сервиса;
- Copy Instance - создание копии сервиса с аналогичными характеристиками;
- Edit the Instance - редактирование сервиса. При выборе действия открывается форма редактирования машины;
- Download instance config - загрузка конфигурационного файла в формате JSON.

Просмотр графа сервиса

На графе машины отображаются объекты, которые входят в ее состав, и связи между ними:

M
Templates
Instances
Providers
admin ▾

Instances / **Mock OpenStack template**

Copy instance
Edit the instance
↓

Settings

Initial ↺ ↻

Provider: -

michman deploy end:

michman deploy start:

michman instance author: 3608f0af-685e-4b5f
-a704-e940bd31e96
c

michman instance id: f3889924-6828-4ffd-90d
0-9d4f21e46352

michman instance immutable:

michman instance name: Mock OpenStack tem
plate

michman template checksum: 449d3db0acbdf
112415f37a9dae
8c242

puccini.quirks: substitution_mappings.requirem
ents.permissive,imports.implicit.
disable

Рис. 21: Просмотр графа

На графе также можно выполнить действия:

- Copy Instance - создание копии сервиса с аналогичными характеристиками;
- Edit the Instance - редактирование сервиса. При выборе действия открывается форма редактирования сервиса;
- Download instance config - загрузка конфигурационного файла сервиса в формате JSON.

Узлами графа могут быть службы, отдельные хосты и прочие сущности, необходимые для функционирования сервиса. В зависимости от статуса объекта (Initial, Started, Error и т.д.) блоки на графе имеют разные цвета. Логические связи между объектами представлены в виде стрелок. При клике на определенный объект в блоке Settings отобразится детальная информация о данном объекте:

M
Templates
Instances
Providers
admin ▾

Instances / **Mock OpenStack template**

Copy instance
Edit the instance
Download

Settings

Host_ ✕

Started

No logs

type: provider.hypervisor.Host

Attributes

private address: -

public address: -

tosca id: ac946d46-a1bf-4560-92... 👁️ 📄

tosca name: Host_ 📄

Properties

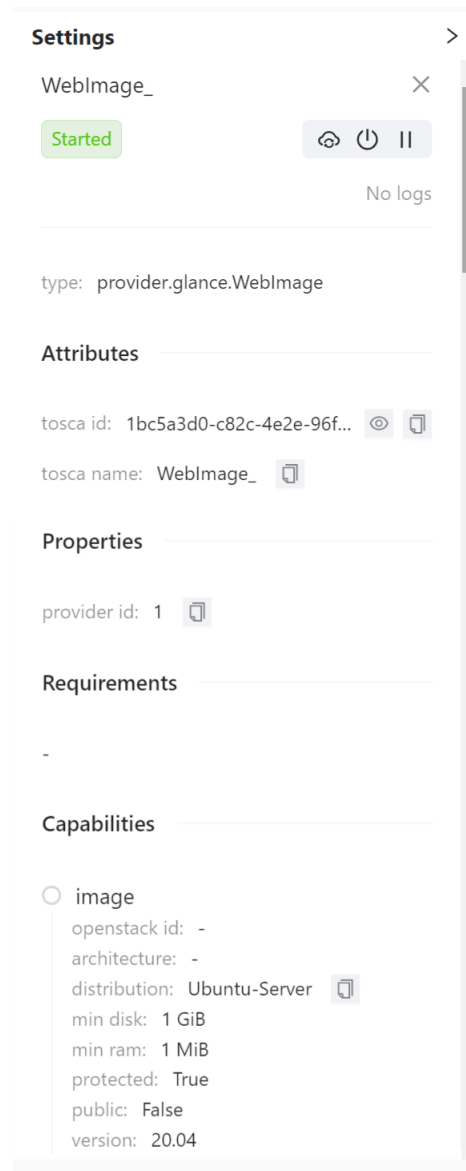
provider id: 1 📄

Рис. 22: Просмотр графа с детальной информацией узла графа

Детальная информация о службе содержит следующие разделы:

- Attributes - атрибуты узла (изменяемые свойства);
- Properties - свойства узла (неизменяемые свойства);
- Requirements - требования для запуска узла;
- Capabilities - функциональные возможности.

Детали службы:



Settings >

WebImage_ ×

Started ↺ ⏻ ||

No logs

type: provider.glance.WebImage

Attributes

tosca id: 1bc5a3d0-c82c-4e2e-96f... 👁️ 📄

tosca name: WebImage_ 📄

Properties

provider id: 1 📄

Requirements

-

Capabilities

- image
 - openstack id: -
 - architecture: -
 - distribution: Ubuntu-Server 📄
 - min disk: 1 GiB
 - min ram: 1 MiB
 - protected: True
 - public: False
 - version: 20.04

Рис. 23: Детали службы

Для запущенной службы в зависимости от статуса доступны следующие действия:

- Sync - синхронизация службы;
- Set Down- выключение службы;
- Pause - остановка службы;
- Deploy - запуск службы.

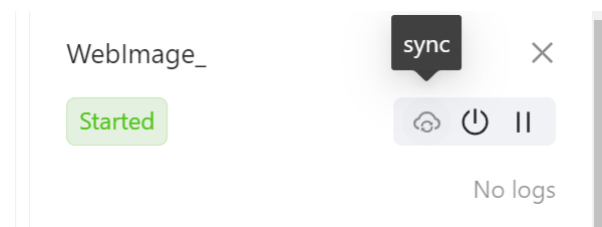


Рис. 24: Действия над службой

Загрузка конфигурации сервиса

При вызове действия Download instance config появляется возможность загрузить файл конфигурации сервиса в формате JSON. При загрузке измените имя файла и его входные параметры:

The screenshot shows the Acloud management console interface. At the top, there is a navigation bar with 'Templates', 'Instances', and 'Providers' tabs, and a user profile 'admin'. Below the navigation bar, the main content area is titled 'Incomplete instances'. It displays a list of instances, each with a name 'Apache Cassandra', version '1.0.0', and author 'tsirleo'. A context menu is open over one of the instances, showing a 'Download config' button. The left sidebar contains a 'Filters' section with a search bar and a 'Provider' section with several checked filters: 'No provider', 'Provider has been removed', 'dcim', and 'openstack'. The 'Status' section is also visible at the bottom of the sidebar.

Рис. 25: Загрузка конфигурации

Редактирование сервиса

При вызове действия Edit открывается страница редактирования сервиса. На странице редактирования при необходимости измените имя и входные параметры:

The screenshot shows the 'Edit' page for a service. At the top, there is a navigation bar with a logo on the left and tabs for 'Templates', 'Instances', and 'Providers'. On the right of the navigation bar, the user 'admin' is logged in. Below the navigation bar, there are three buttons: 'Configured', 'Watch the graph', and a download icon. The main content area shows the service name '[test] TOSCA custom scaling' and a text input field containing '[test]x TOSCA custom? scz'. Below this is an 'Inputs' section with two input fields: '* num servers: 1' and '* num switches: 1'. A 'Save' button is present, and a message at the bottom states 'You have not saved changes'.

Рис. 26: Редактирование сервиса

Также на странице отображается статус, доступны следующие действия:

- Watch the graph - переход на страницу графа;
- Download instance config - загрузка конфигурационного файла.

Копирование сервиса

При вызове действия Copy instance производится создание сервиса с такими же параметрами. Диалоговое окно при этом не открывается и изменения параметров сервиса возможно только после успешного создания копии как обычное редактирование.

Удаление машины

При вызове действия Remove instance производится удаление сервиса. Окно подтверждения при этом не открывается.

Изменение параметров провайдера

2.4.3 Providers

На вкладке Providers представлен список провайдеров, в которых могут быть запущены сервисы. Данные отображаются в плиточном виде:

Для списка провайдеров доступны инструменты сортировки и фильтрации. Сортировка производится по параметрам name и type. Фильтрация производится по параметру "Статус", имеются три основных статуса "Configured", "Error", "Syncing".

Providers + Add provider

Title, type 🔍

Configured Error Syncing

Number of providers: 2

↑ name type

<p>dcim</p> <p>dcim</p> <p>An error has occurred during IM deploy</p> <p>🔄 Sync 🛠️ Change 🗑️ Remove</p>	<p>openstack</p> <p>openstack</p> <p>Last synchronization 07:12:35 09.09.2024</p> <p>🔄 Sync 🛠️ Change 🗑️ Remove</p>
---	---

Рис. 27: Список провайдеров

Для провайдеров доступны следующие действия:

- Add provider - добавление нового провайдера;
- Sync - синхронизация данных, получаемых от облака;
- Change - изменение параметров провайдера;
- Remove - удаление провайдера.

Добавление провайдера

Для добавления нового провайдера нажмите расположенную в правом верхнем углу кнопку Add provider:

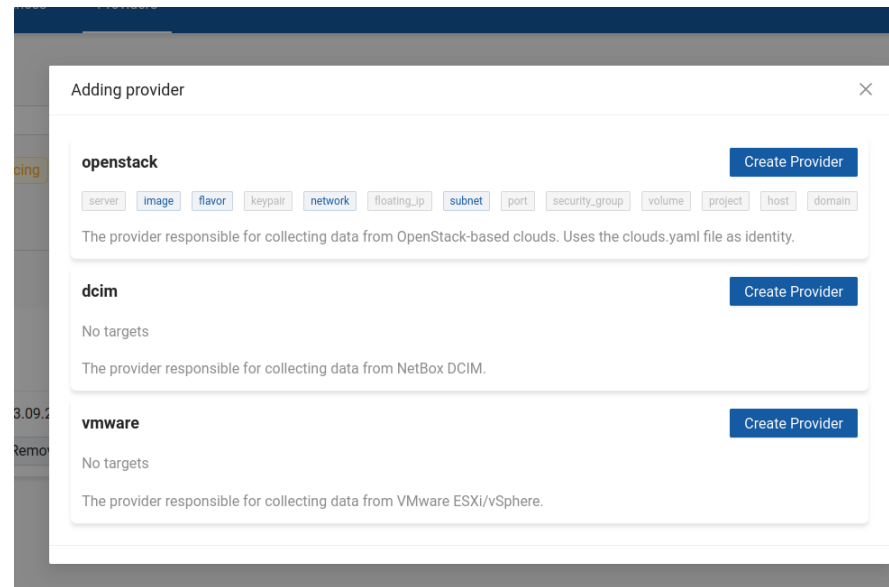


Рис. 28: Добавление провайдера

Для выбора доступны провайдеры типов: openstack, dcim и VMWare. Выберите необходимый тип и нажмите Create Provider.

openstack - провайдер, получающий данные из облаков на базе OpenStack.

В окне создания openstack провайдера:

- Title - введите пользовательское имя нового провайдера;
- Key file - выберите файл приватного ключа, который будет добавляться создаваемым виртуальным машинам для доступа к ним;
- API access file - выберите конфиг-файл clouds.yaml, содержащий данные, необходимые для подключения к одному из проектов в OpenStack (подробнее здесь: <https://docs.openstack.org/python-openstackclient/pike/configuration/index.html#clouds-yaml>).

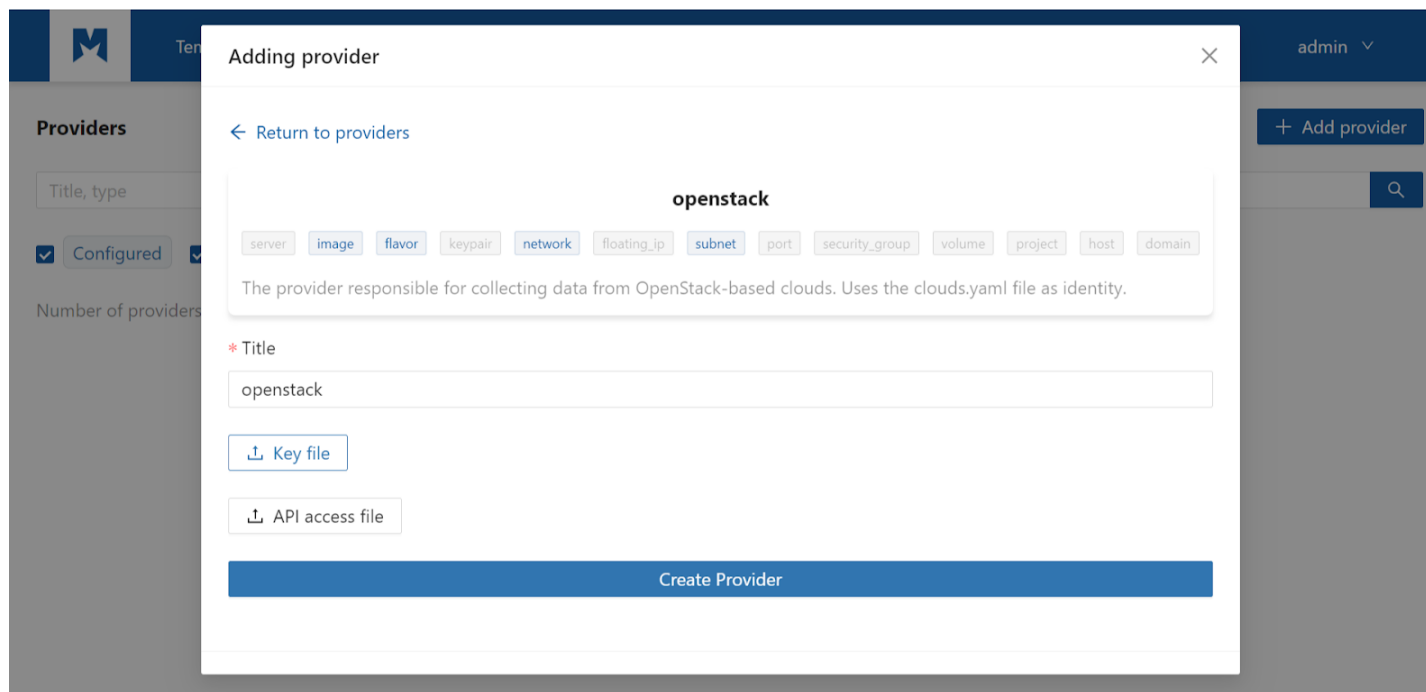


Рис. 29: Добавление провайдера openstack

Также в окне создания отображается информация о том, какие параметры существующего облака по умолчанию будут синхронизироваться в Мичман: image, flavor, network, subnet (для получения информации о некоторых сущностях необходимы права администратора в соответствующем проекте).

dcim - провайдер, получающий данные из NetBox DCIM.

В окне создания dcim провайдера:

- Title - введите пользовательское имя нового провайдера;
- Key file - выберите файл ключа, который будет добавляться создаваемым в рамках данного провайдера виртуальным машинам для доступа к ним;
- API access file - выберите конфиг-файл clouds.yaml, содержащий данные, необходимые для подключения к NetBox DCIM.

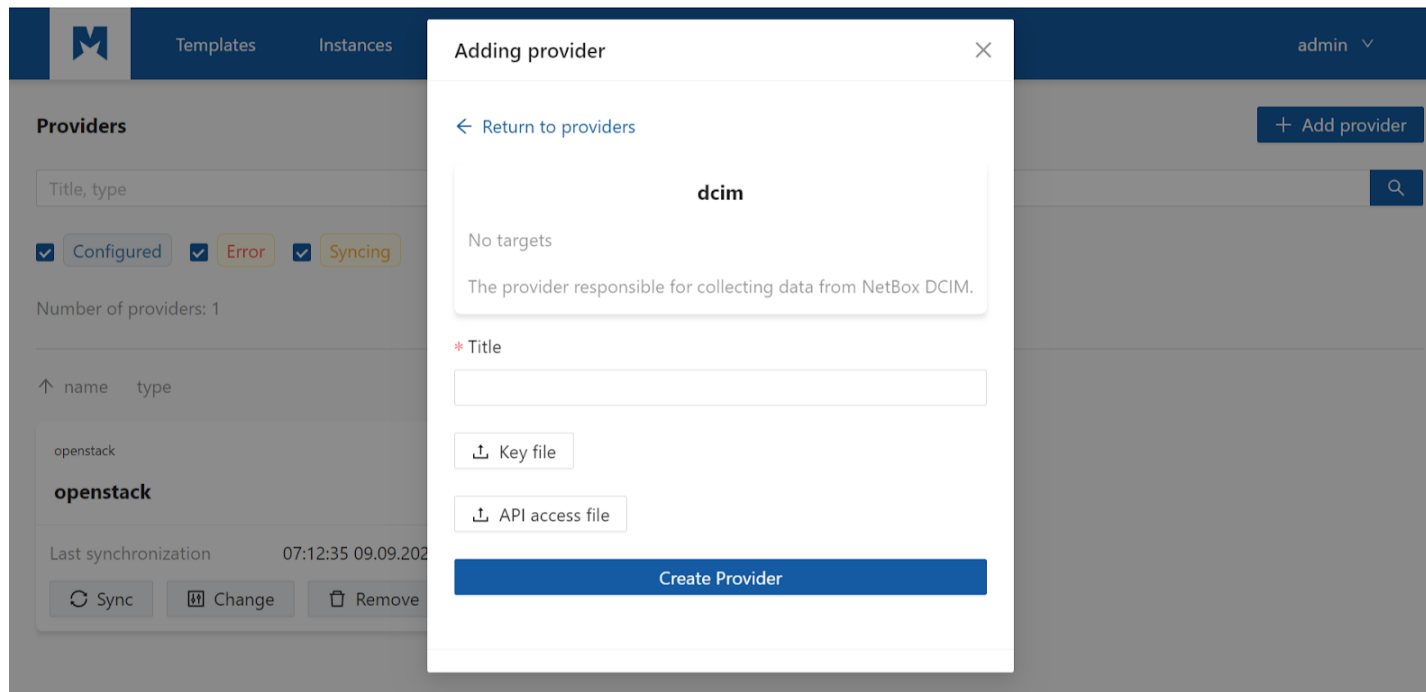


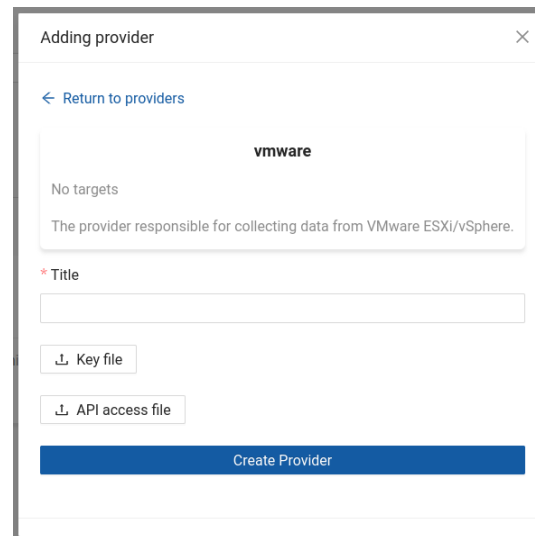
Рис. 30: Добавление провайдера dcim

VMWare - провайдер, получающий данные из облаков на базе VMWare.

В окне создания vmware провайдера:

- Title - введите пользовательское имя нового провайдера;

- Key file - выберите файл приватного ключа, который будет добавляться создаваемым виртуальным машинам для доступа к ним;
- API access file - выберите конфиг-файл clouds.yaml, содержащий данные, необходимые для подключения к облаку VMWare.



Adding provider

[← Return to providers](#)

vmware

No targets

The provider responsible for collecting data from VMware ESXi/vSphere.

* Title

⬇ Key file

⬇ API access file

Create Provider

Рис. 31: Добавление провайдера VMWare

После создания провайдера, он автоматически перейдёт в состояние “Syncing” для сбора информации об облаке. Это может занять длительное время. Кроме того, после создания и синхронизации провайдера на вкладке Instances появится новый *сервис*, соответствующий этому провайдеру. Например, для провайдера типа OpenStack будет содержать все доступные сети, флейворы и образы.

Изменение параметров провайдера

В окне редактирования провайдера для изменения доступны все параметры, которые были заданы при создании:

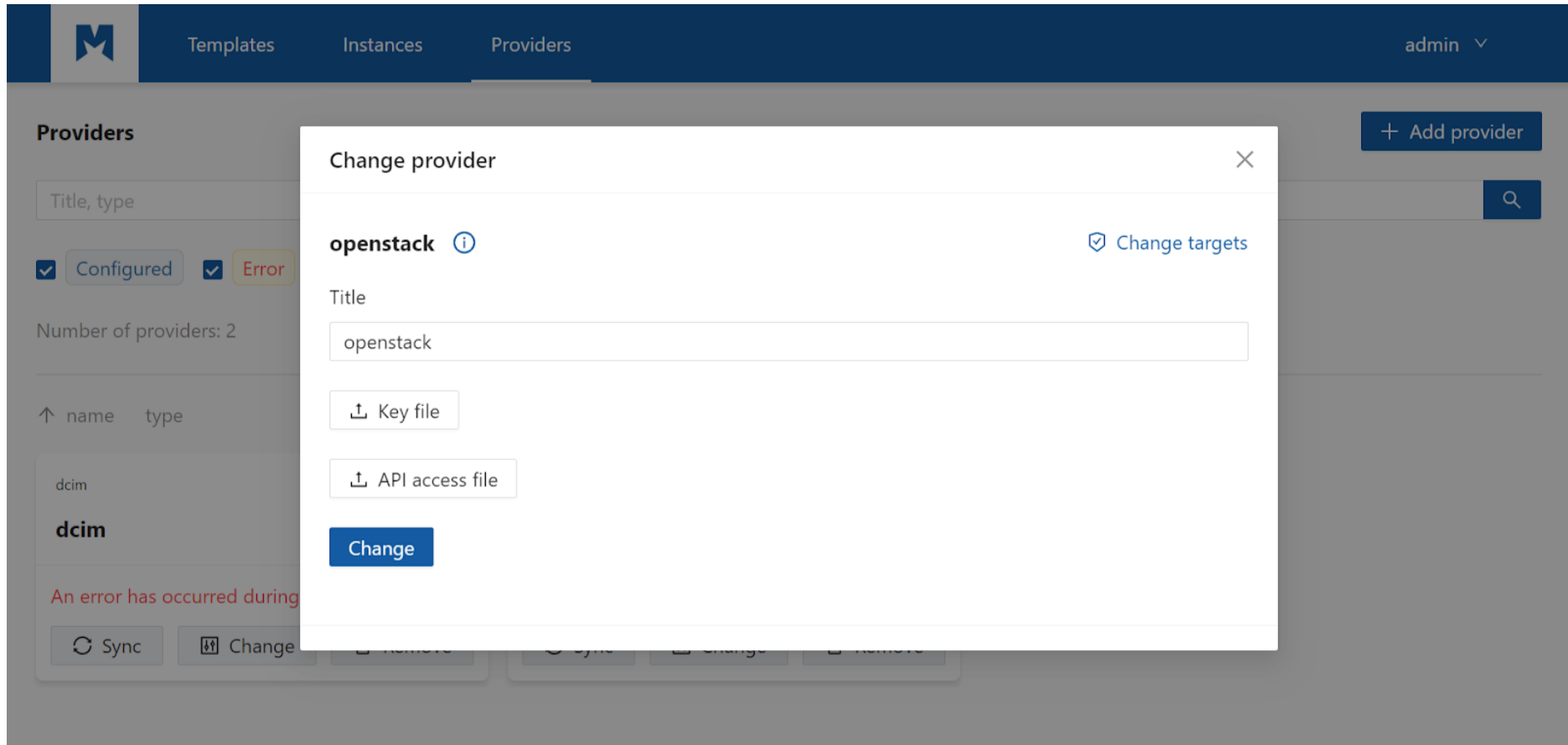


Рис. 32: Редактирование провайдера

Кроме того, при редактировании провайдера есть возможность изменить параметры, по которым происходит синхронизация с существующим облаком. Для изменения параметров нажмите кнопку **Change targets** и выберите нужные параметры:

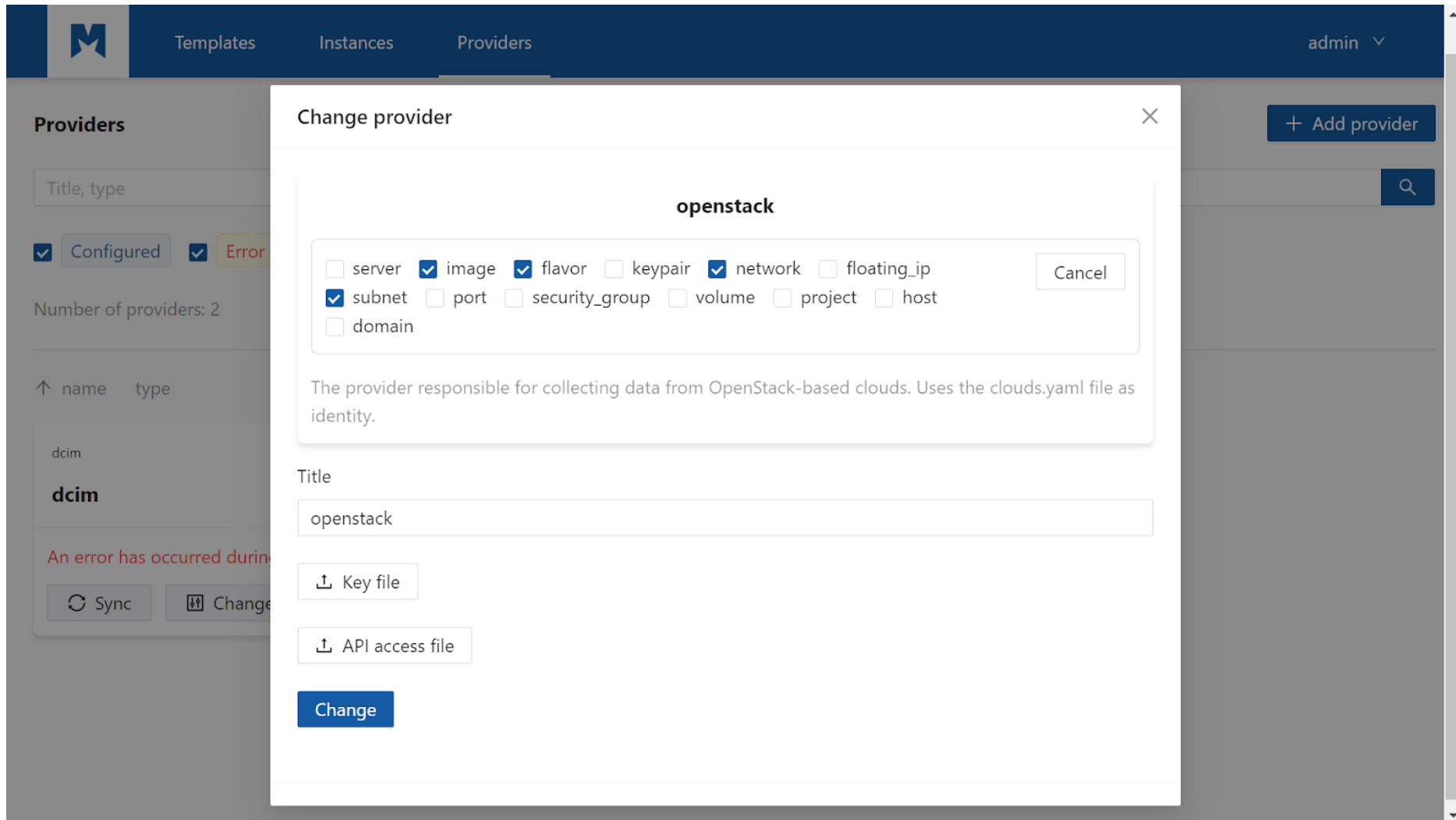


Рис. 33: Изменение параметров провайдера

Переключение между вкладками осуществляется с помощью верхнего меню.

2.5 Биллинг

3.1 Руководство по сопровождению

3.1.1 Сопровождение развёртывания

Обновление узлов

Для начала необходимо сбекапить все данные. Не все папки есть на всех узлах облака

```
mkdir backups-$(date "+%Y-%m-%d")
cd backups-$(date "+%Y-%m-%d")
sudo rsync -a /var/lib/mysql ./
sudo rsync -a /var/lib/rabbitmq ./
sudo rsync -a /var/lib/openswitch ./
sudo rsync -a /var/lib/config-data ./
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo rsync -a /var/lib/kolla ./
sudo rsync -a /var/lib/tripleo-config ./
```

В первую очередь необходимо обновить систему

```
sudo dnf update
```

Можно обновить только puppet-пакеты для актуальных данных

```
for pac in $(sudo dnf list installed | grep puppet- | grep -v rubygem-puppet-resource_api.noarch | cut -d " " -f1 ) ; do sudo dnf update $pac --enablerepo acloud_devel ; done
```

Далее обновите список образов контейнеров на андерклауде для установки:

```
cat container_images | xargs -I {} sudo skopeo copy --src-tls-verify=false \
--dest-tls-verify=false docker://source-images-server/acloud/{}:latest \
docker://localhost:13787/acloud/{}:latest
```

На узлах облака сохраните старые образы контейнеров

```
for im in $(sudo podman image ls | grep acloud | grep latest | cut -d " " \
-f1 | uniq) ; do sudo podman tag $im:latest $im:$(date "+%Y-%m-%d") ; done
```

Обновите контейнеры с registry узла развёртывания

```
for im in $(sudo podman image ls | grep acloud | grep latest | cut -d " " \
-f1 | uniq) ; do sudo podman pull $im:latest ; done
```

Обновление системы узла развёртывания

При обнаружении уязвимостей или выходе малых обновлений компонентов системы необходимо обновить пакеты, а также контейнеры системы, которые отвечают основным компонентам системы.

Обновление узлов вычисления

Так как при обновлении все контейнеры пересоздаются - то будет пересоздан и контейнеры *nova_virt**, отвечающие за создание виртуальных машин. При этом все виртуальные машины не должны останавливаться. Проверьте наличие пакета `systemd-container`.

Обновление узлов управления

Погасите skyline

```
sudo systemctl stop tripleo_skyline-console.service
```

3.1.2 Сопровождение облака и управление ресурсами

Доступ к API облака

После развёртывания облака на узле развёртывания появляется файл `<deployment-name>rc`.

Для управления ресурсами через CLI выполните

```
source <deployment-name>rc
```

Например

```
source acloudrc
```

Обратите внимание, что управление ресурсами развёрнутого облака на узле развёртывания также требует предварительного `source stackrc` с того же узла.

Файл `<deployment-name>rc` содержит переменные окружения, где указывается адрес сервиса аутентификации Keystone и учётные данные для пользователя `admin`.

Если вы использовали доменное имя для эндпойнтов во вкладке **Endpoint Data**, то все публичные эндпойнты будут находиться по настроенному вами доменному адресу или `overcloud.localdomain` по умолчанию.

Чтобы посмотреть IP-адрес для вашего облака выполните

```
source stackrc
openstack port show public_virtual_ip -c fixed_ips
```

Web-доступ к управлению облаком также доступен по указанному адресу.

Флейворы

Для создания флейворов используется команда

```
openstack flavor create cpu<cpu_num>.ram<ram_gb>.disk<disk_gb> \  
  --vcpus <cpu_num> \  
  --ram <ram_mb> \  
  --disk <disk_gb>
```

По умолчанию флейворы создаются публичные. Необходимо явно ставить флаг `--private`, чтобы можно было управлять доступом к флейвору.

Чтобы вместе запустить виртуальную машину с ГПУ или другим PCI устройством - настройте его использование на вычислительном узле в `nova.conf` и указанный `alias` используйте для создания флейвора

```
openstack flavor create --private \  
  gpu-<gpu_alias>-x<gpu_num>.cpu<cpu_num>.ram<ram_gb>.disk<disk_gb> \  
  --property pci_passthrough:alias='<gpu_alias>:<gpu_num>'
```

Чтобы поделить узлы по признаку SSD/HDD или с/без ГПУ - существует механизм **trait**'ов.

Посмотреть их можно для конкретного узла через Placement API

```
openstack resource provider trait list <resource_provider_uuid>\
```

Полный список можно посмотреть следующим образом

```
openstack trait list --sort-column name
```

Из интересных существующих `trait`'ов есть `STORAGE_DISK_HDD`/`STORAGE_DISK_SSD` - но они не назначаются узлам автоматически. Для этого используется команда

```
openstack resource provider trait set --trait <trait_name> <resource_provider_uuid>
```

Чтобы использовать свой `trait`, можно использовать команды

```
openstack trait create CUSTOM_HAS_GPU  
openstack resource provider trait set --trait CUSTOM_HAS_GPU <resource_provider_with_gpu>
```

Далее запрещаем флейвору без ГПУ занимать узлы с ГПУ

```
openstack flavor set cpu1.ram1.hdd10 --property trait:CUSTOM_HAS_GPU=forbidden
```

Или наоборот

```
openstack flavor set cpu1.ram1.hdd10 --property trait:STORAGE_DISK_HDD=required
```

3.2 Руководство по устранению неполадок

3.2.1 Устранение неполадок

Физический узел в статусе *enroll*

Если после добавления физического узла его статус изменился на *enroll*, то возможны следующие варианты:

- Указанный IP-адрес недоступен с узла развёртывания. Проверьте доступность адреса с узла развёртывания: надо выйти из `detui` и попробовать команду

```
ping -c 1 2.2.2.3
```

- Логин/пароль от ВМС неверные

Чтобы изменить параметры ВМС для управления узлом, нужно удалить узел и создать его заново.

Ошибка интроспекции

DnsmasqFilter

Ошибка при попытке интроспектировать узел

```
The PXE filter driver DnsmasqFilter, state=uninitialized: my fsm encountered an exception: Can not transition from state 'uninitialized' on event 'sync' (no defined transition)
```

Описание проблемы есть на сайте [RedHat Solutions](#) Причина ошибки: драйвер `pxefilter` не может инициализировать `DNSmasq` Решение: перезапустить все сервисы `Ironiс`

```
for service in $(sudo systemctl | grep tripleo_ironic | cut -d " " -f3) ; do sudo systemctl restart $service ; done
```

Ramdisk error

```
2024-02-26 19:16:12.688 7 ERROR ironic_inspector.utils [-] [node: MAC 00:25:90:bb:0e:d8 BMC 10.100.2.33] Ramdisk reported error: The following errors were encountered:
* failed to run hardware-detect utility: [Errno 2] No such file or directory: 'hardware-detect'
2024-02-26 19:16:12.688 7 ERROR ironic_inspector.process [-] [node: MAC 00:25:90:bb:0e:d8 BMC 10.100.2.33] Hook ramdisk_error failed, delaying error report until node look up: Ramdisk reported error: The following errors were encountered:
* failed to run hardware-detect utility: [Errno 2] No such file or directory: 'hardware-detect': ironic_inspector.utils.Error: Ramdisk reported error: The following errors were encountered:
```

Смотреть логи в `/var/log/containers/ironic-inspector/ramdisk`

Параметр не определён или определён неправильно в шаблоне

Если при создании стека не хватает какого-то параметра, но при этом, стек уже начал создаваться. Или если результат не соответствует указанному в параметре. То можно исследовать результаты создания стека в директории `/home/stack/overcloud-deploy`

Ошибка настройка сети

Если при развёртывании облака вы видите ошибку,

```
"failed_when_result": true
}
2023-03-02 17:02:49.089226 | 08002710-c16f-44d5-9117-00000000029d | TIMING | NetworkConfig stdout | monitoring-ceph-0 | 0:05:58.946554 | 0.06s
```

зайдите на упавший хост по ssh и выполните

```
sudo cat /etc/os-net-config/config.yaml
```

В выводе будет сгенерённый для вашего узла сетевой план. Проверьте, что план соответствует ожидаемым настройкам . Если план не соответствует, то перейдите к настройке сетевого плана снова и запустите обновление развёртывания.

Если план соответствует ожидаемому, то выполните

```
sudo os-net-config -c /etc/os-net-config/config.yaml --verbose
```

И посмотрите на вывод команды и ошибки, которые команда выведет. Дальнейшая работа с проблемой зависит от ошибок в выводе команды.

После исправления всех ошибок, с узла развёртывания выполните

```
screen -x <deploy_name>
```

```
INTERACTIVE MODE  
Enter command: c
```

Развёртывание продолжится дальше

Puppet cache read-only файловая система

Если при развёртывании падают контейнеры вида container-puppet-. То необходимо проверить логи контейнера после старта

```
sudo less /var/log/containers/stdouts/container-puppet-<service>.log
```

Если видны ошибки вида

```
boost::filesystem::remove: Read-only file system: "/opt/puppetlabs/facter/cache/cached_facts/kernel"
```

то проблема скорее всего в том, что развёртывание было начато некоторое время назад и кеш успел измениться. Необходимо пересоздать один из упавших контейнеров container-puppet- с разрешением на запись для container-volume=/var/lib/container-puppet/

```
sudo podman ps -a  
#использовать вывод для пересоздания контейнера  
sudo podman inspect container-puppet-<service> | jq .[0].Config.CreateCommand  
sudo podman rm container-puppet-<service>  
sudo podman container run <...> --volume /var/lib/container-puppet/puppetlabs:/opt/puppetlabs:rw <...>
```

Остальные упавшие контейнеры вида container-puppet- можно просто перезапустить

```
sudo podman restart container-puppet-<service>
```

Wait for puppet host configuration to finish

Если при развёртывании сервисов Ansible падает на задаче *Wait for puppet host configuration to finish*, то сначала необходимо определить узел, на котором произошла ошибка

```
YYYY-MM-DD HH:MM:SS | 00000000-0000-0000-0000-000000000000 | WAITING | Wait for puppet host configuration to finish | acloud-controller-0
```

В примере выше ошибка произошла на узле *acloud-controller-0*. Для поиска причины ошибки необходимо зайти по `ssh` на этот узел и запустить puppet самостоятельно

```
sudo puppet apply --modulepath=/etc/puppet/modules:/opt/stack/puppet-modules:/usr/share/openstack-puppet/modules --detailed-exitcodes --summarize --color=true --verbose --debug /var/lib/tripleo-config/puppet_step_config.pp
```

Дальнейший поиск проблемы зависит от вывода команды.

Если вывод команды долгий или недостаточно информативный, то добавьте флаг `--debug`

Расemaker

Ошибка выглядит следующим образом:

```
Error: 'pcs status | grep -q -e "acloud-compute-30[[:blank:]].*Started"' returned 1 instead of one of [0]
```

Это значит, что расemaker не смог запустить ресурс *acloud-compute-30*.

С любого из контроллеров выполните:

```
sudo pcs status
```

Вы увидите статус *FAILED* для ресурса *acloud-compute-30*. Проверьте правильность его настройки:

```
sudo pcs resource config acloud-compute-30
```

В выводе будет указан адрес ресурса в поле *Attributes*.

1. Проверьте доступность адреса с узла контроллера, на котором вы сейчас находитесь.

```
ping -c 1 <ip>
```

1. Проверьте, что на этот адрес ходят большие пакеты

```
ping -c 1 -s 10000 <ip>
```

1. Проверьте, что доступен порт 3121. Запрос не должен виснуть и не должен выводить ошибку *Failed to connect*

```
curl <ip>:3121
```

Если порт не доступен, то зайдите на проблемный узел по ssh и проверьте статус сервиса *pacemaker_remote*

```
sudo systemctl status pacemaker_remote
```

Этот сервис должен быть активен только на узлах виртуализации. На узлах контроллерах должен быть активен сервис *pacemaker*.

Если сервис активен, но имеет ошибки вида

```
error: TLS handshake with remote client failed: An illegal parameter has been received.
```

то скопируйте файл */etc/pacemaker/auth* с рабочего узла. Команды с узла развёртывания

```
ssh tripleo-admin@<working-host>.ctlplane 'sudo cp /etc/pacemaker/auth ./'
rsync -azP tripleo-admin@<working-host>.ctlplane:authkey ./
rsync -azP authkey tripleo-admin@<failed-host>.ctlplane:
ssh tripleo-admin@<failed-host>.ctlplane 'sudo cp authkey /etc/pacemaker/'
```

После решения проблем зайдите на любой из контроллеров и выполните

```
sudo pcs resource refresh <failed-host>
```

В указанном выше примере *failed-host* == *acloud-compute-30*.

Запустите повторно puppet на узле, на котором puppet упал изначально. Если puppet выполнит работу без ошибок, то продолжите работу Ansible далее без повторного выполнения задачи Ansible.

Не найден хост

После обновления развёртывания

1. Переведите nova-scheduler в режим debug. Для этого поменяйте параметр *debug=False* в *True* в файле */var/lib/config-data/puppet-generated/nova/etc/nova/nova.conf*. Затем перезапустите nova-scheduler

```
sudo systemctl restart tripleo_nova_scheduler
```

1. Попробуйте создать виртуальную машину и проверьте в логах ошибку

```
less /var/log/containers/nova/nova-scheduler.log
```

Если ошибка выглядит следующим образом,

```
Filter AllHostsFilter returned 0 host(s)
```

то зайдите на любой контроллер (только один!) и выполните

```
sudo podman exec -ti -u root nova_compute nova-manage placement heal_allocations
sudo podman exec -ti -u root nova_compute nova-manage placement sync_aggregates
```

Число используемых ресурсов на гипервизорах обнулилось

Такое может произойти, если вы попытались поменять основной домен для имен хостов. В таком случае вы увидите следующее.

Например, вы поменяли хостнейм с *localdomain* на *novalocal*, тогда

```
openstack server list --all-projects --host acloud-compute-12.localdomain -f value | wc -l
20

openstack hypervisor show acloud-compute-12.novalocal -c service_host -c running_vms

+-----+-----+
| Field      | Value |
+-----+-----+
| running_vms | 0     |
| service_host | acloud-compute-12.localdomain |
+-----+-----+
```

Причина проблемы в том, что у виртуалок запущенных на хосте *acloud-compute-12.localdomain* есть параметр гипервизора, который остался в значении *acloud-compute-12.localdomain*.

Чтобы решить проблему, необходимо на любом из контроллеров зайти в базу данных Mariadb и поменять значение для виртуалки

```

sudo podman ps | grep galera

ab09fec32c4b  undercloud.ctlplane:13787/acloud/openstack-mariadb:pcmklatest          /bin/bash /usr/lo...  9 days ago    Up 8 days ago    ''
                galera-bundle-podman-0

sudo podman exec -ti -u root galera-bundle-podman-0 mysql
> use nova_api
> select display_name from instances where node='acloud-compute-12.localdomain' and deleted_at is null;
> begin;
> update instances set node='acloud-compute-12.novalocal' where node='acloud-compute-12.localdomain' and deleted_at is null;
> commit ;

```

Бекенд для cinder недоступен

Если бекенд находится в статусе down при выполнении команды

```
openstack volume service list
```

Сначала проверить доступность бекенда с узлов, пароли, MTU сети. Затем

```

openstack volume service set --disable hostgroup@tripleo_ceph_reliable cinder-volume -vv
pcs resource restart openstack-cinder-volume
openstack volume service set --enable hostgroup@tripleo_ceph_reliable cinder-volume -vv

```

Вычислительный узел не добавлен в nova_cell

Если на вычислительном узле в логах /var/log/containers/nova/nova-compute.log ошибка выглядит следующим образом:

```

ERROR nova.cmd.common [req-4431a4a0-6789-466f-9ed1-b518a05b5771 - - - -] No db access allo
wed in nova-compute: File "/usr/bin/nova-compute", line 10, in <module>

```

То скорее всего вычислительный узел был развёрнут без undercloud узла в консоли - нужно развернуть с ним вместе.

Ошибка миграции Host key verification failed

Быстрый метод решения проблемы

```
sudo su
echo 'StrictHostKeyChecking no' >> /var/lib/config-data/puppet-generated/nova_libvirt/var/lib/nova/.ssh/config
systemctl restart tripleo_nova_compute
```

Виртуальная машина в статусе ERROR

Если виртуальная машина перешла в статус ERROR, то необходимо Выяснить на каком узле находится машина

```
openstack server show -c OS-EXT-SRV-ATTR:host <server_id/name>
```

Зайти на узел и выяснить ошибку. Варианты могут быть следующие:

1. Память

```
free -mh
```

1. Дисковая память

```
df -h
```

1. Статус контейнеров

```
sudo podman ps | grep nova
```

1. Логи контейнеров

```
ls /var/log/containers
```

После выяснения всех ошибок и исправления их проверьте существование VM на этом узле, а именно

```
ls /var/lib/nova/instances/<server_id>
```

Затем верните VM в работающее состояние

```
openstack server set --state active <server_id/name>
openstack server stop <server_id/name>
openstack server start <server_id/name>
```

Если машина снова перешла в состояние ERROR, значит ошибка не исправлена - продолжайте исследовать

Libvirt ошибка секрета

При создании виртуальной машине в логах узла виртуализации /var/log/containers/libvirt/ видна ошибка

```
Secret not found: no secret with matching uuid
```

Выполните на узле

```
sudo podman exec -ti -u root nova_virtqemud virsh secret-list
sudo podman exec -ti -u root nova_virtqemud virsh secret-undefine <secret_uuid>
sudo podman exec -ti -u root nova_virtqemud virsh secret-define /etc/nova/secret.xml
sudo podman exec -ti -u root nova_virtqemud virsh secret-set-value
```

Причина ошибки - смена fsid для serph'a, необходимо удалить старый fsid и добавить новый указанными выше командами

4.1 Программа и методика испытаний

4.1.1 1. Объект испытаний

1.1 Полное наименование программного обеспечения

Объектом испытаний является программное обеспечения «Отечественная облачная платформа 1.0».

4.1.2 2. Цель испытаний

Целью проведения испытаний Программы является выявление возможных ошибок и недоработок для последующего их устранения, а также обеспечения функционирования Программы в полном объеме.

Задачей испытания является последовательное выполнение этапов проверки согласно настоящей программе и методике испытаний.

4.1.3 3. Техническое обеспечение испытаний

Программное обеспечение Системы должно корректно функционировать в следующих интернет-браузерах:

- Mozilla Firefox
- Google Chrome
- Яндекс.Браузер

4.1.4 Приложение А

Тестирование программы

Требования:

1. Система должна обеспечить реализацию компонентов пользовательского интерфейса в виде веб-интерфейса, работа с которым осуществляется с АРМ при помощи интернет-браузера.
2. Программа должна скрывать данные пользователя при вводе.
3. Программа должна обеспечивать авторизацию пользователя.
4. Программа должна обеспечивать создание нового проекта.
5. Программа должна обеспечивать добавление пользователя в проект.
6. Программа должна обеспечивать создание нового диска в проекте.
7. Программа должна обеспечивать создание новой сети в проекте.
8. Программа должна обеспечивать создание виртуальной машины в проекте.
9. Программа должна обеспечивать подключение диска к виртуальной машине в проекте.
10. Программа должна обеспечивать создание резервной копии (снимка) виртуальной машины в проекте.

5.1 Информация о стоимости лицензий

Стоимость лицензий на использование программного обеспечения «Отечественная облачная платформа 1.0» определяется на договорной основе.

5.2 AccentOS RemoteApp

- *Стоимость*

Программное обеспечение AccentOS RemoteApp служит для публикации и администрирования удалённых приложений с серверов под управлением облачных платформ.

Система предоставляет наборы приложений для пользователей, в их личных кабинетах.

Программное обеспечение поддерживает публикацию приложения как через веб-интерфейс, так и публикацию напрямую на рабочий стол пользователя.

Программный продукт AccentOS RemoteApp включен в [Единый реестр российских программ для электронных вычислительных машин и баз данных](#).

Документация, содержащая информацию, необходимую для установки программного обеспечения.

Документация, содержащая описание функциональных характеристик программного обеспечения.

Документация, содержащая информацию, необходимую для эксплуатации программного обеспечения.

5.2.1 Стоимость

Стоимость ПО определяется по запросу клиента, в зависимости от запрашиваемых характеристик и необходимого функционала. Для уточнения стоимости необходимо обратиться по номеру телефона +7 (495) 899 0929 или отправить письмо на email info@acloud.ru.

ACL (Access Control List) список управления доступом, который определяет, кто или что может получать доступ к объекту (программе, процессу или файлу), и какие именно операции разрешено или запрещено выполнять субъекту (пользователю, группе пользователей).

ACPI открытый промышленный стандарт, впервые выпущенный в декабре 1996 года и разработанный совместно компаниями HP, Intel, Microsoft, Phoenix и Toshiba, который определяет общий интерфейс для обнаружения аппаратного обеспечения, управления питанием и конфигурации материнской платы и устройств.

Active Directory LDAP-совместимая реализация службы каталогов корпорации Microsoft для операционных систем семейства Windows Server. Позволяет администраторам использовать групповые политики для обеспечения единообразия настройки пользовательской рабочей среды, разворачивать программное обеспечение на множестве компьютеров через групповые политики или посредством System Center Configuration Manager (ранее — Microsoft Systems Management Server), устанавливать обновления операционной системы, прикладного и серверного программного обеспечения на всех компьютерах в сети, используя Службу обновления Windows Server.

Amphora виртуальная машина под управлением *Octavia*, которая занимается балансировкой нагрузки.

AMQP открытый протокол для передачи сообщений между компонентами системы. Основная идея состоит в том, что отдельные подсистемы (или независимые приложения) могут обмениваться произвольным образом сообщениями через AMQP-брокер, который осуществляет маршрутизацию, доставку, распределение потоков данных, подписку на нужные типы сообщений.

Ansible система управления конфигурациями, написанная на Python с использованием декларативного языка разметки для описания конфигураций. Используется для автоматизации настройки и развертывания программного обеспечения. Обычно используется для управления Linux-узлами, но Windows также поддерживается.

Aodh компонент OpenStack *Telemetry*. Предоставляет функцию оповещений.

Apache Tomcat контейнер, который позволяет использовать интернет-приложения такие, как Java сервлеты и JSP (серверные страницы Java).

ASCII название таблицы (кодировки, набора), в которой некоторым распространённым печатным и непечатным символам сопоставлены числовые коды. Таблица ASCII определяет коды для символов:

- десятичных цифр;
- латинского алфавита;
- национального алфавита;
- знаков препинания;
- управляющих символов.

Таблица символов ASCII:

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

-
- AWS (Amazon Web Services)** коммерческое публичное облако, поддерживаемое и развиваемое компанией Amazon. Предоставляет услуги как по инфраструктурной модели (виртуальные серверы, ресурсы хранения), так и платформенного уровня (облачные базы данных, облачное связующее программное обеспечение, облачные бессерверные вычисления, средства разработки).
- Bareos** кроссплатформенное клиент-серверное программное обеспечение, позволяющее управлять резервным копированием, восстановлением и проверкой данных по сети для компьютеров и операционных систем различных типов.
- BGP (Border Gateway Protocol)** динамический протокол маршрутизации. Относится к классу протоколов маршрутизации внешнего шлюза (EGP — External Gateway Protocol). На текущий момент является основным протоколом динамической маршрутизации в сети Интернет.
- Ceilometer** средства сбора, нормализации и трансформации данных, предоставляемых сервисами OpenStack. Собираемые данные используются для реализации различных сценариев реагирования на события.
- Celery** асинхронная очередь задач, которая основана на распределенной передаче сообщений.
- Ceph** масштабируемая распределенная система хранения большого объема, которая состоит из хранилища объектов, блочного хранилища, а также *POSIX*-совместимой распределенной файловой системы. Совместима с OpenStack.
- CephFS** *POSIX*-совместимая файловая система, предоставляемая Ceph.
- CGI** стандарт интерфейса, используемого для связи внешней программы с веб-сервером.
- Chef** инструмент управления конфигурацией операционной системы, поддерживаемый OpenStack.
- chroot** операция изменения корневого каталога в Unix-подобных операционных системах. Программа, запущенная с изменённым корневым каталогом, будет иметь доступ только к файлам, содержащимся в данном каталоге. Поэтому, если нужно обеспечить программе доступ к другим каталогам или файловым системам (например, /proc), нужно заранее примонтировать в целевом каталоге необходимые каталоги или устройства.
- Cinder** сервис блочного хранения данных в системе OpenStack.
- Cloud-init** пакет инициализации виртуальных машин после их загрузки. Использует информацию, полученную от служб метаданных, например, SSH с открытым ключом и пользовательских данных.
- Cobbler** это сервер сетевой установки и обновления Linux- и Windows-систем, позволяющий создать среду бездисковой загрузки, включающую образы и репозитории пакетов, сконфигурировать и развернуть ОС. Cobbler использует такие службы, как DHCP, TFTP и DNS. Его можно настроить для PXE, переустановок и виртуализированных гостей с использованием Xen, KVM или VMware. Cobbler взаимодействует с программой koan для переустановки и поддержки виртуализации. koan и Cobbler используют libvirt для интеграции с различным программным обеспечением для виртуализации. Cobbler может управлять сложными сетевыми сценариями, такими как мостовое соединение по связанному каналу Ethernet.
- Common Internet File System (CIFS)** протокол общего доступа к файлам. Это публичный или открытый вариант исходного протокола Server Message Block (SMB), разработанного и используемого Microsoft. Как и протокол SMB, CIFS выполняется на верхнем уровне и использует протокол TCP/IP.
- Consul** система обнаружения и конфигурирования сервисов. Предоставляет множество различных функций, таких как обнаружение сервисов и узлов, проверка работоспособности, общесистемное хранение ключей и значений.

CUPS (Common UNIX Printing System) сервер печати для UNIX-подобных операционных систем. Компьютер с запущенным сервером CUPS представляет собой сетевой узел, который принимает задания на печать от клиентов, обрабатывает их и отправляет на соответствующий принтер.

cURL кроссплатформенная служебная программа командной строки, позволяющая взаимодействовать со множеством различных серверов по множеству различных протоколов с синтаксисом URL.

DCC (Dynamic Configuration Components) режим динамического конфигурирования компонентов на узлах.

Device mapper (dm) модуль ядра Linux, позволяющий создавать виртуальные блочные устройства. При обращении к таким устройствам выполняется ряд действий, в число которых обычно входит чтение/запись данных с других блочных устройств. Подсистема используется для реализации менеджера логических томов LVM, программного RAID, системы шифрования дисков dm-crypt. Одной из возможностей подсистемы является создание снимков файловой системы.

DevStack набор скриптов, которые устанавливают все компоненты OpenStack. Идеально подходит для тестирования и разработки.

DHCP сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер».

Direct SCSI режим, предоставляющий возможность прямой отправки SCSI команд устройству в обход гипервизора.

dm-cache компонент ядра Linux для кэширования данных. Решение позволяет использовать высокоскоростное блочное устройство (SSD) как кэш для одного или большего количества медленных устройств хранения как жесткие диски, например, HDD.

Docker программное обеспечение для автоматизации развёртывания и управления приложениями в среде виртуализации с поддержкой контейнеризации. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер.

Docker Compose инструмент для управления многоконтейнерными приложениями Docker. Позволяет описывать и запускать необходимую структуру при помощи одного конфигурационного файла.

Dockerfile скрипт, который позволяет автоматизировать процесс построения контейнеров Docker.

Domain Name имя домена - символьное имя, служащее для идентификации областей в сети Интернет.

DRS (Distributed Resource Scheduling) инструмент балансировки, предназначенный для проведения и применения аудитов по возможности балансировки виртуальных машин на узлах в рамках агрегаторов узлов.

DSCP (Differentiated Services Code Point) элемент архитектуры компьютерных сетей, описывающий простой масштабируемый механизм классификации, управления трафиком и обеспечения качества обслуживания.

ELK стек интегрированное решение, включающее в себя:

- Elasticsearch;
- Logstash;
- Kibana.

FreeRDP бесплатная реализация протокола удаленного рабочего стола (RDP).

Fully Qualified Domain Name (FQDN) имя домена, не имеющее неоднозначностей в определении. Включает в себя имена всех родительских доменов иерархии DNS.

Ganglia масштабируемая распределенная система мониторинга кластеров параллельных и распределенных вычислений и облачных систем с иерархической структурой. Позволяет отслеживать статистику и историю (загруженность процессоров, сети) вычислений в реальном времени для каждого из наблюдаемых узлов.

Gatling платформа для проверки нагрузки и производительности с открытым исходным кодом.

Glance проект OpenStack, который отвечает за ведение каталога, регистрацию и доставку образов виртуальных машин.

GlusterFS распределённая, параллельная, линейно масштабируемая файловая система с возможностью защиты от сбоев. С помощью InfiniBand RDMA или TCP/IP GlusterFS может объединить хранилища данных, находящиеся на разных серверах, в одну параллельную сетевую файловую систему. GlusterFS работает в пользовательском пространстве при помощи технологии FUSE, поэтому не требует поддержки со стороны ядра операционной системы и работает поверх существующих файловых систем (ext3, ext4, XFS, reiserfs и т. п.). В отличие от других распределённых файловых систем, таких как Lustre и Ceph, для работы GlusterFS не требуется отдельный сервер для хранения метаданных.

Gnocchi сервис OpenStack, предназначенный для хранения агрегированных измерений в очень больших масштабах и управления ими. Включает в себя многопользовательскую базу данных временных рядов, метрик и ресурсов. В качестве источника измеренных данных может использоваться *Ceilometer*.

GNU Privacy Guard (GnuPG, GPG) свободная программа для шифрования информации и создания электронных цифровых подписей. Разработана как альтернатива PGP и выпущена под свободной лицензией GNU General Public License. GnuPG полностью совместима со стандартом IETF OpenPGP.

Google Authenticator приложение для двухэтапной аутентификации с помощью Time-based One-time Password Algorithm (TOTP) и HMAC-based One-time Password Algorithm (HOTP) от Google LLC. Authenticator генерирует 6- или 8-значный одноразовый цифровой пароль, который пользователь должен предоставить в дополнение к логину и паролю, чтобы авторизоваться в службах Google или сторонних сервисах.

GRUB загрузчик операционной системы от проекта GNU. GRUB позволяет пользователю иметь несколько установленных операционных систем и при включении компьютера выбирать одну из них для загрузки.

GUID статистически уникальный 128-битный идентификатор.

Gunicorn HTTP-сервер интерфейса шлюза веб-сервера (*WSGI*) Python. Базируется на pre-fork модели, это означает, что главный процесс управляет иницированными рабочими процессами различного типа, создает сокеты, соединения и т.п.

Hadoop проект фонда Apache Software Foundation, фреймворк для разработки программ с открытым исходным кодом, в котором поддерживаются распределенные приложения, обрабатывающие большие объемы данных.

HAProxy серверное программное обеспечение для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений, посредством распределения входящих запросов на несколько обслуживающих серверов.

Heat Служба оркестрации OpenStack, которая согласовывает работу составных облачных приложений с помощью декларативных шаблонов через собственный интерфейс REST API платформы OpenStack.

httpd серверное программное обеспечение для работы с протоколом HTTP в режиме демона.

IKE (Internet Key Exchange) стандартный протокол набора протоколов IPsec, используемый для обеспечения безопасности взаимодействия в виртуальных частных сетях.

Initrd временная файловая система, используемая ядром Linux при начальной загрузке. Initrd обычно используется для начальной инициализации перед монтированием «настоящих» файловых систем.

IP Fabric разработчик служб управления инфраструктурой сети IP, предназначенных для визуализации и безопасного управления корпоративными сетями.

iptables таблицы, предоставленные брандмауэром ядра Linux (реализовано в качестве различных модулей межсетевого экрана), а также с помощью цепочек и правил в нем сохраненных. Используется наряду с arptables, ebtables и ip6tables, чтобы создавать брандмауэры в службе вычислительных ресурсов. На данный момент для различных протоколов используются различные модули ядра и программы: iptables применяются для IPv4, ip6tables — для IPv6, arptables — для ARP, а ebtables для фреймов Ethernet. Чтобы управлять средством, понадобятся корневые права.

IPXE свободное программное обеспечение для создания загрузочных ПЗУ (Постоянное запоминающее устройство) для загрузки Linux и других операционных систем на компьютерах с архитектурой x86 по сети с использованием межсетевых протоколов.

iSCSI протокол, который базируется на TCP/IP и разработан для установления взаимодействия и управления системами хранения данных, серверами и клиентами.

iSCSI target программа или аппаратный контроллер (HBA), осуществляющие эмуляцию диска и выполняющие запросы iSCSI.

Jenkins программная система с открытым исходным кодом на Java, предназначенная для обеспечения процесса непрерывной интеграции разрабатываемого программного обеспечения, а также проверки кода. В частности, применяется при разработке программного обеспечения на базе OpenStack.

JMeter инструмент для проведения нагрузочного тестирования, разрабатываемый Apache Software Foundation.

jQuery библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML.

Kerberos протокол аутентификации сети, работающий на базе заявок. Kerberos поддерживает незащищенную связь между узлами, а также предоставляет им возможность взаимной безопасной идентификации.

Keystone проект (сервис) OpenStack Identity, который при посредстве API-интерфейса OpenStack предоставляет такую функциональность, как идентификация, токены, политики и каталоги.

Kubernetes платформа оркестровки контейнеров построенная на базе etcd и systemd, позволяющая управлять кластерами виртуальных машин и Linux-контейнеров, созданными с использованием таких инструментариев, как Docker и Rocket, как единым целым.

LBaaS балансировщик нагрузки. Позволяет сетевой службе равномерно распределять входящие запросы между виртуальными машинами.

LDAP относительно простой протокол, использующий TCP/IP и позволяющий производить операции аутентификации (bind), поиска (search) и сравнения (compare), а также операции добавления, изменения или удаления записей. Обычно LDAP-сервер принимает входящие соединения на порт 389 по протоколам TCP или UDP. Для LDAP-сеансов, инкапсулированных в SSL, обычно используется порт 636.

libvirt свободная реализация API, демон и набор инструментов для управления виртуализацией. Позволяет управлять гипервизорами Xen, KVM, а также VirtualBox, OpenVZ, LXC, VMware ESX/GSX/Workstation/Player, QEMU и другими средствами виртуализации, предоставляет возможность контролировать виртуальные машины по сети, расположенные на других компьютерах.

LightDM дисплейный менеджер X, который стремится быть лёгким, быстрым, расширяемым и поддерживающим множество рабочих столов. Фронтенд Unity Greeter из состава Ubuntu использует WebKit для отображения основанного на HTML интерфейса входа в систему.

LizardFS кластерная файловая система, которая распространяет данные по нескольким физическим серверам, делая их видимыми для конечного пользователя как одной файловой системы.

LUN (Logical Unit Number) адрес дискового устройства в сетях хранения.

MAC-адрес уникальный идентификатор, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet.

Management Information Base (MIB) виртуальная база данных, используемая для управления объектами в сети связи. Наиболее часто это понятие связывают с Simple Network Management Protocol (*SNMP*). Хотя термин MIB предназначен для обозначения всей доступной информации об объекте, он также часто используется для обозначения конкретного подмножества, которое правильнее называть MIB-модулем.

MATE среда рабочего стола, являющаяся ответвлением от кодовой базы неподдерживаемой в настоящее время среды GNOME 2. MATE представляет собой интуитивно понятный рабочий стол с традиционной концепцией построения интерфейса.

mdadm утилита для управления программными RAID-массивами в Linux.

Memcached программное обеспечение, реализующее сервис кэширования данных в оперативной памяти на основе хеш-таблицы.

Mistral сервис (проект) OpenStack, предоставляющий простой YAML-подобный язык для описания потоков операций, задач и правил переходов. Позволяет загружать, изменять и запускать их в режиме высокой готовности, управлять и следить за ходом выполнения потока операций, а также за состоянием отдельных задач.

MODBUS открытый коммуникационный протокол, основанный на архитектуре ведущий-ведомый (master-slave). Широко применяется в промышленности для организации связи между электронными устройствами. Может использоваться для передачи данных через последовательные линии связи RS-485, RS-422, RS-232, а также сети TCP/IP (Modbus TCP).

MongoDB документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++.

Network File System (NFS) протокол сетевого доступа к файловым системам. За основу взят протокол вызова удалённых процедур ONC RPC. Позволяет подключать удалённые файловые системы через сеть.

-
- Network Namespace (netns)** логически отделенный от других стек сетевых протоколов в Linux. Эмулируется полностью сетевой стек: сетевые интерфейсы, таблица маршрутизации, фаерволл и т.д.
- Neutron** сервис OpenStack, предоставляющий «подключение к сети как услугу» между интерфейсами устройств (vNIC), которые управляются другими сервисами OpenStack.
- Nginx** веб-сервер и обратный прокси-сервер, работающий на Unix-подобных операционных системах.
- NodeSafe** технология, обеспечивающая безопасность виртуальной инфраструктуры, за счет сканирования всех проходящих через гипервизор инструкций, обнаружения угроз и предотвращения воздействия вредоносного программного обеспечения.
- Nova** проект (сервис) OpenStack, базовый компонент архитектуры, управляющий вычислительными ресурсами. Все действия, необходимые для поддержки жизненного цикла виртуальной машины в облаке OpenStack обрабатывает Nova. Служба может работать с различными технологиями виртуализации (гипервизорами), такими, как *KVM*, VMware, Xen, а также с Hyper-V и системами виртуализации на уровне операционной системы, такими, как LXC.
- NUMA (Non-Uniform Memory Access)** схема реализации компьютерной памяти, используемая в мультипроцессорных системах, когда время доступа к памяти определяется её расположением по отношению к процессору.
- OCFS (Oracle Cluster File System)** кластерная файловая система, поддерживающая разделяемое использование между несколькими Linux-системами, разработку которой осуществляет корпорация Oracle под лицензией GNU General Public License. Файловая система обладает семантикой локальной файловой системы и может быть использована почти любыми приложениями.
- Octavia** масштабируемый балансировщик нагрузки с открытым исходным кодом и поддержкой различных категорий операторов. Разработан для работы с OpenStack.
- OID** это строка или последовательность десятичных цифр, однозначно идентифицирующая объект. Такими объектами обычно являются классы объектов или атрибуты.
- Open vSwitch** многоуровневый виртуальный коммутатор производственного класса с открытым исходным кодом, предоставляемый по лицензии Apache 2.0. Предназначен для автоматизации управления крупными сетями с помощью программных средств, но при этом поддерживает стандартные сетевые протоколы и интерфейсы (например, NetFlow, sFlow, SPAN, RSPAN, CLI, LACP, 802.1ag).
- OpenLDAP** открытая реализация LDAP, разработанная одноимённым проектом, распространяется под собственной свободной лицензией OpenLDAP Public License. В числе прочих есть реализации для различных модификаций BSD, а также Linux, AIX, HP-UX, Mac OS X, Solaris, Microsoft Windows (NT и наследники — 2000, XP, Vista, Windows 7) и z/OS.
- OpenSSL** полноценная криптографическая библиотека с открытым исходным кодом, широко известна из-за расширения SSL/TLS, используемого в веб-протоколе HTTPS.
- OpenStack** набор открытого программного обеспечения для провайдеров услуг или частного использования, предназначенный для установки и запуска облачной инфраструктуры вычислений и хранения.
- Openstack Client** официальная утилита командной строки OpenStack. Представляет собой унифицированный клиент для доступа к OpenStack API.
-

OpenStack Horizon/Dashboard графический интерфейс управления ресурсами OpenStack.

OSD (Object Storage Device) юнит хранилища *Ceph*, который хранит сами данные и обрабатывает запросы клиентов, обмениваясь данными с другими OSD. Обычно это диск. И обычно за каждый OSD отвечает отдельный OSD-демон, который может запускаться на любой машине, на которой установлен этот диск.

oVirt свободная, кроссплатформенная система управления виртуализацией.

pbr (Python Build Reasonableness) библиотека для управления средствами настроек, необходима для установки пакетов OpenStack.

PEP 440 стандарт для идентификации версий дистрибутивов программного обеспечения на Python и объявления зависимостей от конкретных версий.

pip система управления пакетами, используемая для установки и управления программными пакетами, написанными на Python.

pip3 система управления пакетами, используемая для установки и управления программными пакетами, написанными на Python3.

POSIX (переносимый интерфейс операционных систем) набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов. Стандарт создан для обеспечения совместимости различных UNIX-подобных операционных систем и переносимости прикладных программ на уровне исходного кода, но может быть использован и для не-Unix систем.

PostgreSQL свободная объектно-реляционная система управления базами данных.

PulseAudio кроссплатформенный звуковой сервер. PulseAudio работает на платформах POSIX, таких как Linux, Solaris и FreeBSD, а также на платформе Microsoft Windows. PulseAudio является свободным программным обеспечением: сервер распространяется под лицензией GNU GPL, а библиотеки — под GNU LGPL.

PXE (Preboot eXecution Environment) среда для загрузки компьютера с помощью сетевой карты без использования локальных носителей данных (жёсткого диска, USB-накопителя и т.п.). Для организации загрузки системы в PXE используются протоколы IP, UDP, BOOTP и TFTP.

PyPI (Python Package Index) каталог программного обеспечения, написанного на языке программирования Python.

QEMU Copy On Write 2 (QCOW2) один из форматов дисков для образов виртуальных машин, поддерживаемый службой образов.

Qt кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода.

Quick EMUlator (QEMU) QEMU — эмулятор и виртуализатор различных платформ компьютера с открытым исходным кодом. Один из гипервизоров, поддерживаемых OpenStack, использовался для целей разработки.

RAID (избыточный массив независимых дисков) технология виртуализации данных, которая объединяет несколько дисков в логический элемент для избыточности и повышения производительности. Разные типы конфигураций массива отмечаются разными номерами: 1, 2, 3, 4 и т.д.

RAW один из форматов дисков для образов виртуальных машин, поддерживаемый службой образов. Неструктурированный образ диска.

- RBAC (Role Based Access Control)** управление доступом на основе ролей, система управления доступом, в которой права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя роли.
- RDP** (англ. Remote Desktop Protocol — протокол удалённого рабочего стола) — проприетарный протокол прикладного уровня, используется для обеспечения удалённой работы пользователя с сервером, на котором запущен сервис терминальных подключений.
- Redis** сетевое журналируемое хранилище данных типа «ключ — значение» с открытым исходным кодом. Нереляционная высокопроизводительная система управления базами данных.
- REST API** набор функций, к которым разработчики могут совершать запросы и получать ответы. Взаимодействие происходит по протоколу HTTP. Преимуществом такого подхода является широкое распространение протокола HTTP, поэтому REST API можно использовать практически из любого языка программирования.
- root/суперпользователь** специальный аккаунт в UNIX-подобных системах с идентификатором (UID, User Identifier) 0, владелец которого имеет право на выполнение всех без исключения операций, или пользователя, который может заходить на интерфейс администратора.
- RSA** криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.
- rsync** программа для UNIX-подобных систем, которая выполняет синхронизацию файлов и каталогов в двух местах с минимизированием трафика, используя кодирование данных при необходимости.
- SaltStack** система управления конфигурациями и удалённого выполнения операций. Является программным обеспечением с открытым исходным кодом, написанным на Python. Поддерживает подход IaaS для развёртывания и управления облачными вычислениями.
- Samba** пакет программ, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу SMB/*CIFS*. Имеет клиентскую и серверную части. Является свободным программным обеспечением, выпущена под лицензией GPL.
- Scala** мультипарадигмальный язык программирования, спроектированный кратким и типобезопасным для простого и быстрого создания компонентного программного обеспечения, сочетающий возможности функционального и объектно-ориентированного программирования.
- SCP** протокол особого RCP копирования файлов, использующий в качестве транспорта не RSH, а SSH.
- SCSI** набор стандартов для физического подключения и передачи данных между компьютерами и периферийными устройствами. SCSI-стандарты определяют команды, протоколы и электрические и оптические интерфейсы.
- Selenium** инструмент для автоматизированного управления браузерами. Наиболее популярной областью применения Selenium является автоматизация тестирования веб-приложений.
- SELinux** реализация системы принудительного контроля доступа, которая может работать параллельно с классической избирательной системой контроля доступа.
- Sentry** система трекинга и агрегирования ошибок. Более подробное описание доступно на официальном сайте [Sentry](#).
- Server URI** унифицированный идентификатор сервера, например: `http://10.35.21.53:10002`.

Snapshot моментальный снимок образа виртуальной машины, термин OpenStack. Позволяет делать снимок образа работающей виртуальной машины без ее приостановки.

SNMP стандартный интернет-протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP. К поддерживающим SNMP устройствам относятся маршрутизаторы, коммутаторы, серверы, рабочие станции, принтеры, модемные стойки и другие.

snmpd *SNMP* агент, который отвечает на запросы *SNMP* для данного узла.

SPICE протокол отображения удаленного дисплея, который позволяет просматривать виртуальный «рабочий стол» вычислительной среды не только на машине, на которой он запущен, но и откуда угодно через Интернет.

SSH сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений.

SSL криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.

sudo программа для системного администрирования UNIX-систем, позволяющая делегировать те или иные привилегированные ресурсы пользователям с ведением протокола работы.

Supervisor система по управлению процессами в операционной системе. Основные компоненты:

- supervisor - серверная часть;
- supervisorctl - клиентская часть.

Более подробное описание доступно на сайте официальной [документации supervisor](#).

Swift облачное файловое хранилище.

Syslog (system log) стандарт отправки и регистрации сообщений о происходящих в системе событиях (то есть создания логов), использующийся в компьютерных сетях, работающих по протоколу IP.

systemd системный менеджер, демон инициализации других демонов в Linux. Его особенностью является интенсивное распараллеливание запуска служб в процессе загрузки системы, что позволяет существенно ускорить запуск операционной системы.

Tacker проект OpenStack, в котором используется *менеджер VNF* и *оркестратор NFV* для развертывания и управления сетевыми сервисами и *VNF* в архитектуре *NFV*.

TCP один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных. Сети и подсети, в которых совместно используются протоколы TCP и IP называются сетями TCP/IP.

TDD (test-driven development) разработка через тестирование, техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам.

Tempest пакет автоматизированных программных тестов, предназначенных для выполнения по отношению к стволу официального проекта OpenStack.

TLS протокол защиты транспортного уровня. Использует асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

TOTP (Time-based One-Time Password Algorithm) OATH-алгоритм создания одноразовых паролей для защищенной аутентификации, являющийся алгоритмом односторонней аутентификации — сервер удостоверяется в подлинности клиента.

Tox протокол для децентрализованной текстовой, голосовой и видеосвязи в интернете на основе асимметричного шифрования.

UTC всемирное координированное время, стандарт, по которому общество регулирует часы и время.

UUID (universally unique identifier) стандарт идентификации, используемый в создании программного обеспечения. Основное назначение UUID — это позволить распределённым системам уникально идентифицировать информацию без центра координации. Таким образом, любой может создать UUID и использовать его для идентификации чего-либо с приемлемым уровнем уверенности, что данный идентификатор непреднамеренно никогда не будет использован для чего-то ещё. Поэтому информация, помеченная с помощью UUID, может быть помещена позже в общую базу данных без необходимости разрешения конфликта имен. UUID представляет собой 16-байтный (128-битный) номер. В шестнадцатеричной системе счисления UUID выглядит как:

```
550e8300-e29b-41d4-a716-444655440000.
```

uWSGI веб-сервер и сервер веб-приложений, первоначально реализованный для запуска приложений Python через протокол WSGI. Версия 2.0 поддерживает также запуск веб-приложений Lua, Perl, Ruby и других.

vCPU это виртуальный процессор, содержащий одно ядро и занимающий один сокет.

VDI (Virtual Desktop Infrastructure) технология, позволяющая создавать виртуальную ИТ-инфраструктуру и разворачивать полноценные рабочие места на базе одного сервера, на котором работает множество виртуальных машин.

VDO (Virtual Data Optimizer) технология виртуализации блочных устройств, позволяющая создавать из блочных устройств виртуальные пулы с поддержкой онлайн-компрессии и онлайн-дедупликации.

virt-top консольная команда, которая выводит список виртуальных доменов и информацию о них. Утилита `virt-top` использует библиотеку управления виртуализацией `libvirt`.

Virtual Extensible LAN (VXLAN) является технологией сетевой виртуализации, созданной для решения проблем масштабируемости в больших системах облачных вычислений.

VLAN (Virtual Local Area Network) логическая («виртуальная») локальная компьютерная сеть, представляет собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену, независимо от их физического местонахождения. VLAN имеет те же свойства, что и физическая локальная сеть, но позволяет конечным станциям группироваться вместе, даже если они не находятся в одной физической сети. Такая реорганизация может быть сделана на основе программного обеспечения вместо физического перемещения устройств.

VMTP инструмент измерения производительности сети внутри облака OpenStack.

VNC система удалённого доступа к рабочему столу компьютера, использующая протокол RFB. Управление осуществляется путём передачи нажатий клавиш на клавиатуре и движений мыши с одного компьютера на другой и ретрансляции содержимого экрана через компьютерную сеть. Система VNC платформонезависима: VNC-клиент, называемый VNC viewer, запущенный на одной операционной системе, может подключаться к VNC-серверу, работающему на любой другой операционной системе.

Wake-on-LAN технология, позволяющая удалённо включить компьютер посредством отправки через локальную сеть специальной последовательности байтов — пакета данных (magic packet). Этот пакет может быть вставлен в пакеты любых стандартных протоколов более высоких уровней, например, UDP или IPX.

Watcher сервис оптимизации вычислительной нагрузки облачных ресурсов в системе OpenStack.

World Wide Name (WWN)/World Wide Identifier (WWID) уникальный идентификатор, который определяет конкретное целевое устройство (target) Fibre Channel, Advanced Technology Attachment (ATA) или Serial Attached SCSI (SAS).

WSGI (Web Server Gateway Interface) стандарт взаимодействия между Python-программой, выполняющейся на стороне сервера, и самим веб-сервером, например, Apache.

X Window System (X11,X) оконная система, обеспечивающая стандартные инструменты и протоколы для построения графического интерфейса пользователя. Используется в UNIX-подобных операционных системах.

X.Org Server (Xorg) библиотечная реализация серверной части *X Window System* с открытым исходным кодом.

X2Go даёт удалённый доступ к графическому интерфейсу Linux. Защищённость соединения предоставляется благодаря использованию SSH.

XFS высокопроизводительная 64-битная журналируемая файловая система. XFS отличается от других файловых систем тем, что она изначально была рассчитана для использования на дисках большого объёма (более 2 терабайт, см. например, RAID-массивы).

XRDP сервер для Unix-систем, реализованный поверх *VNC* и предоставляющий доступ к рабочему столу терминального сервера по протоколу *RDP*.

Zabbix свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования.

ZRAM модуль ядра Linux. Увеличивает производительность путём предотвращения подкачки страниц на диск, используя сжатое блочное устройство в оперативной памяти, пока не появится необходимость использовать файл подкачки на жёстком диске. Скорость обмена с оперативной памятью выше, чем с жёстким диском, следовательно, zRam позволяет Linux производить большее число операций подкачки, особенно на старых компьютерах с малым объёмом оперативной памяти.

Автоматическая (принудительная) миграция остановка виртуальных машин и их запуск на свободных ресурсах при наступлении определенных условий.

Автоматическая эвакуация автоматический перенос виртуальных машин при выявлении проблем на вычислительном узле, на котором они размещались. Перенос осуществляется на другие вычислительные узлы кластера только при наличии свободных ресурсов.

Агент DHCP Агент OpenStack Networking, который предоставляет службы DHCP для виртуальных сетей.

Агрегатор узлов метод разделения зон доступности в пуле гипервизора; совокупность обычных узлов.

Аутентификация процесс, подтверждающий подлинность пользователя, другого процесса или клиента с помощью закрытого ключа, секретного токена, пароля, отпечатка пальцев или аналогичным способом.

База данных совокупность всех данных (таблиц, процедур, триггеров и т.д.), статических данных (неизменяемых данных, хранящихся в lookur-таблицах) и пользовательских данных (которые изменяются в процессе работы с приложением).

Балансировка нагрузки равномерное распределение нагрузки на вычислительные узлы и системы хранения данных при разворачивании новых виртуальных машин.

Бесклассовая адресация (Classless Inter-Domain Routing, CIDR) метод IP-адресации, позволяющий гибко управлять пространством IP-адресов, не используя жёсткие рамки классовой адресации. Использование этого метода позволяет экономно использовать ограниченный ресурс IP-адресов, поскольку возможно применение различных масок подсетей к различным подсетям.

Брокер сообщений пакет программного обеспечения, используемого для обеспечения AMQP. Осуществляет обмен сообщений между компонентами программной системы. По умолчанию пакет RabbitMQ.

Вертикальное масштабирование увеличение производительности каждого компонента системы с целью повышения общей производительности. Масштабируемость в этом контексте означает возможность заменять в существующей вычислительной системе компоненты более мощными и быстрыми по мере роста требований и развития технологий. Это самый простой способ масштабирования, так как не требует никаких изменений в прикладных программах, работающих на таких системах.

Виртуализация сетевых функций (Network Functions Virtualization, NFV) концепция сетевой архитектуры, предлагающая использовать технологии виртуализации для виртуализации целых классов функций сетевых узлов в виде составных элементов, которые могут быть соединены вместе или связаны в цепочку для создания телекоммуникационных услуг (сервисов). Архитектура NFV включает следующие основные элементы:

VNF (Virtual Network Function) виртуальная сетевая функция, например: DNS, DHCP, коммутатор, маршрутизатор, балансировщик или базовая станция и т.д.

Система управления элементами сети (Element Management System, EMS) программное обеспечение, предназначенное для управления и администрирования одной или несколькими VNF.

NFV Orchestrator (оркестратор NFV) осуществляет администрирование инфраструктуры NFV программными ресурсами, создание готового сервиса из нескольких VNF.

VNF Manager (VNFM) менеджер, отвечает за жизненный цикл VNF: инсталляция, активация, масштабирование, обновление и ликвидация. Может отвечать за работу одной или нескольких VNF.

NFVI (Network Function Virtualization Infrastructure) инфраструктура NFV: аппаратные и программные ресурсы в физическом и виртуальном виде, на которых работают виртуальные сетевые функции VNF. Могут находиться локально или быть распределенными территориально и соединены сетью оператора.

Hardware Resources (аппаратные ресурсы, то есть вычислительные, сетевые и ресурсы хранения) физическая часть инфраструктуры NFVI - любой стандартный коммутатор, или физический сервер, или устройство хранения и т.д.

Virtualized Infrastructure Manager (менеджер виртуальной инфраструктуры) отвечает за взаимодействие виртуальной сетевой функции с аппаратными и программными ресурсами и инвентаризацию имеющихся ресурсов, а также отвечает за сбор событий и вопросы производительности.

Виртуальная машина на основе ядра (KVM) гипервизор, поддерживаемый OpenStack. KVM — полномасштабное решение в сфере виртуализации для Linux на 32-разрядном оборудовании, содержащее расширения виртуализации (Intel VT или AMD-V), ARM, IBM Power и IBM zSeries. Состоит из загружаемых модулей ядра, предоставляющих основную инфраструктуру виртуализации и модуль для конкретного процессора.

Виртуальная файловая система (Virtual file system — VFS) уровень абстракции поверх конкретной реализации файловой системы. Целью VFS является обеспечение единообразного доступа клиентских приложений к различным типам файловых систем. VFS может быть использована для доступа к локальным устройствам и файлам (fat32, ext4, ntfs), сетевым устройствам и файлам на них (nfs), а также к устройствам, не предназначенным для хранения данных. VFS декларирует программный интерфейс между ядром и драйвером конкретной файловой системой, таким образом, для добавления поддержки новой файловой системы не требуется вносить изменений в ядро операционной системы.

ВМ/Виртуальная машина программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы, виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы.

ВУ/вычислительный узел, нода многопроцессорный, многоядерный компьютер, на котором выполняются задачи пользователя. Задача пользователя может занимать один вычислительный узел, несколько вычислительных узлов или все вычислительные узлы.

Гипервизор программа или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких операционных систем на одном и том же хост-компьютере. Гипервизор также обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами.

Главная загрузочная запись (master boot record, MBR) код и данные, необходимые для последующей загрузки операционной системы и расположенные в первых физических секторах (чаще всего в самом первом) на жёстком диске или другом устройстве хранения информации.

Горизонтальное масштабирование разбиение системы на более мелкие структурные компоненты и разнесение их по отдельным физическим машинам (или их группам), и (или) увеличение количества серверов, параллельно выполняющих одну и ту же функцию. Масштабируемость в этом контексте означает возможность добавлять к системе новые узлы, серверы, процессоры для увеличения общей производительности. Этот способ масштабирования может требовать внесения изменений в программы, чтобы программы могли в полной мере пользоваться возросшим количеством ресурсов.

Горячая замена (HotPlug) термин, означающий отключение или подключение электронного оборудования в системе во время ее работы без выключения питания и остановки.

Графический интерфейс пользователя (Graphical user interface/GUI) разновидность пользовательского интерфейса, в котором элементы интерфейса (меню, кнопки, значки, списки и т. п.), представленные пользователю на дисплее, исполнены в виде графических изображений. Также называется графической оболочкой управления.

Группа безопасности набор правил фильтрации сетевого трафика, который применяется к виртуальной машине.

Дамп памяти содержимое рабочей памяти одного процесса, ядра или всей операционной системы. Также может включать дополнительную информацию о состоянии программы или системы, например, значения регистров процессора и содержимое стека. Многие операционные системы позволяют сохранять дампы памяти для отладки программы. Как правило, дампы памяти процесса сохраняются автоматически, когда процесс завершается из-за критической ошибки (например, из-за ошибки сегментации). Дампы также можно сохранить вручную через отладчик или любую другую специальную программу.

Дата-центр специализированное здание для размещения (хостинга) серверного и сетевого оборудования и подключения абонентов к каналам сети Интернет.

Даунгрейд процесс перехода на более ранние версии аппаратного или программного обеспечения.

Двухфакторная аутентификация метод идентификации пользователя в сервисе при помощи запроса аутентификационных данных двух разных типов, что обеспечивает двухступенчатую, более эффективную защиту аккаунта от несанкционированного доступа.

Дедупликация процесс обнаружения повторяющихся данных на уровне блоков, файлов или объектов, который помогает высвободить дисковое пространство. На данный момент не поддерживается OpenStack.

Дельта образа разница между базовым и модифицированным образом *Glance*. Обычно является слоем (набором ПО, готовым для добавления к базовому образу).

Демон (daemon) компьютерная программа в системах класса UNIX, запускаемая самой системой и работающая в фоновом режиме без прямого взаимодействия с пользователем. Демоны обычно запускаются во время загрузки системы. Типичные задачи демонов: серверы сетевых протоколов (HTTP, FTP, электронная почта и др.), управление оборудованием, поддержка очередей печати, управление выполнением заданий по расписанию и т.д. В техническом смысле демоном считается процесс, который не имеет управляющего терминала.

Диск/том блочное устройство постоянного хранения, которое может быть присоединено к виртуальной машине.

Домен это совокупность пользователей, групп и проектов. Каждая группа и проект принадлежат только одному домену.

Живая миграция перенос виртуальной машины с одного физического сервера на другой без прекращения работы виртуальной машины и остановки сервисов. Живая миграция возможна между серверами, находящимися в кластере.

Золотой образ образ, подключаемый в библиотеку образов *Glance*. Представляет из себя базовый образ с уже добавленным к нему набором *дельт*.

Зона доступности изолированная область, которая используется для обеспечения отказоустойчивости.

Интерфейс IPMI интеллектуальный интерфейс управления платформой, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ. Ключевые характеристики IPMI — мониторинг, восстановление функций управления, журналирование и инвентаризация, которые доступны независимо от процессора, BIOS'a и операционной системы. Функции управления платформой могут быть доступны, даже если система находится в выключенном состоянии.

Интерфейс командной строки (Command line interface, CLI) разновидность текстового интерфейса между человеком и компьютером, в котором инструкции компьютеру даются в основном путём ввода с клавиатуры текстовых команд. Также известен под названием консоль или терминал. Ин-

терфейс командной строки противопоставляется системам управления программой на основе меню, а также различным реализациям графического интерфейса.

Интерфейс пользователя (User interface/UI) интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы.

Качество обслуживания (QoS) возможность гарантировать определенные требования к сети или системе хранения в соответствии с соглашением об уровне обслуживания (SLA) между поставщиком приложения и конечными пользователями. Обычно в QoS входят требования производительности, например значения пропускной способности, задержки, исправления дрожания и надежности, а также значение производительности устройств хранения в операциях ввода-вывода в секунду (*IOPS*), соглашения о допустимом количестве запросов и ожидаемая производительность при пиковых нагрузках.

Кластер группа компьютеров, объединённых высокоскоростными каналами связи, представляющая с точки зрения пользователя единый аппаратный ресурс. Кластер - слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений и представляющихся пользователю единой системой.

Количество операций ввода-вывода в секунду (IOPS) IOPS — это общее измерение производительности, используемое для запоминающих устройств компьютера, таких как жесткие диски, твердотельные накопители и сети хранения данных.

Лог файл файл регистрации, протокол, журнал, файл с записями о событиях в хронологическом порядке. Различают регистрацию внешних событий и протоколирование работы самой программы - источника записей (хотя часто всё записывается в единый файл).

Логирование процедура записи всех процессов системы в текстовый файл с возможностью выставления разного уровня детализации записи.

Маршрутизатор специализированный сетевой компьютер, имеющий два или более сетевых интерфейсов и пересылающий пакеты данных между различными сегментами сети. Маршрутизатор может связывать разнородные сети различных архитектур. Для принятия решений о пересылке пакетов используется информация о топологии сети и определённые правила, заданные администратором.

Межпроцессное взаимодействие (inter-process communication, IPC) обмен данными между потоками одного или разных процессов. Реализуется посредством механизмов, предоставляемых ядром ОС или процессом, использующим механизмы ОС и реализующим новые возможности IPC. Может осуществляться как на одном компьютере, так и между несколькими компьютерами сети.

Менеджер логических томов/LVM подсистема операционных систем Linux и OS/2, позволяющая использовать разные области одного жёсткого диска и/или области с разных жёстких дисков как один логический том.

Метрика программного обеспечения (software metric) мера, позволяющая получить численное значение некоторого свойства программного обеспечения и его спецификаций.

Многопутевой ввод-вывод (Multipath I/O) технология подключения узлов сети хранения данных с использованием нескольких маршрутов. Например, одно *SCSI*-устройство может быть подсоединено к двум *SCSI*-контроллерам. В случае отказа одного из контроллеров, операционная система будет использовать другой для доступа к устройству. Данная архитектура повышает отказоустойчивость системы и позволяет распределять нагрузку.

Мьютекс механизм, служащий в программировании для синхронизации одновременно выполняющихся потоков.

Облачное хранилище данных модель онлайн-хранилища, в котором данные хранятся на многочисленных распределённых в сети серверах, предоставляемых в пользование клиентам, в основном, третьей стороной. В отличие от модели хранения данных на собственных выделенных серверах, приобретаемых или арендуемых специально для подобных целей, количество или какая-либо внутренняя структура серверов клиенту, в общем случае, не видна. Данные хранятся и обрабатываются в так называемом «облаке», которое представляет собой, с точки зрения клиента, один большой виртуальный сервер. Физически же такие серверы могут располагаться удалённо друг от друга географически.

Образ виртуальной машины содержит загружаемую файловую систему с определёнными параметрами и данными, которая используется для запуска виртуальных машин.

Образ загрузочного диска тип образа виртуальной машины в виде единого загрузочного файла.

Обратный прокси-сервер (reverse proxy) тип прокси-сервера, который ретранслирует запросы клиентов из внешней сети на один или несколько серверов, логически расположенных во внутренней сети. При этом для клиента это выглядит так, будто запрашиваемые ресурсы находятся непосредственно на прокси-сервере.

Отказоустойчивость свойство технической системы сохранять свою работоспособность после отказа одного или нескольких составных компонентов. Отказоустойчивость определяется количеством любых последовательных единичных отказов компонентов, после которого сохраняется работоспособность системы в целом. Базовый уровень отказоустойчивости подразумевает защиту от отказа одного любого элемента — исключение единой точки отказа. Основной способ повышения отказоустойчивости — избыточность. Наиболее эффективный метод избыточности — аппаратная избыточность, которая достигается путём резервирования. В ряде приложений отказоустойчивость путём резервирования является обязательным требованием, предъявляемым государственными надзорными органами к техническим системам.

Отказоустойчивый кластер/кластер высокой доступности кластер (группа серверов), спроектированный в соответствии с методом обеспечения высокой доступности и гарантирующий минимальное время простоя за счёт аппаратной избыточности. Без кластеризации сбой сервера приведёт к тому, что поддерживаемые им приложения или сетевые сервисы будут недоступны до восстановления его работоспособности. Отказоустойчивая кластеризация исправляет эту ситуацию, перезапуская приложения на другой системе без вмешательства администратора в случае обнаружения ошибок аппаратного или программного обеспечения. Процесс перезапуска известен как аварийное переключение. В рамках этого процесса программное обеспечение кластеризации может настроить узел перед запуском приложения на нём (например, импортировать и установить соответствующие файловые системы или перезапустить некоторые поддерживаемые приложения). Отказоустойчивые кластеры обычно используют специальный сигнал, который используется для мониторинга состояния и статуса каждого узла в кластере.

Пауза состояние виртуальной машины, при котором не происходит никаких изменений (нет изменений в памяти, сетевые соединения остановлены и т.д.). Виртуальная машина остановлена, но не выключена.

Перезагрузка сервера различаются два типа: горячая и холодная перезагрузка сервера. При горячей перезагрузке операционная система получает сигнал на перезапуск, что позволяет нормально завершить работу всех процессов. Холодная перезагрузка это процесс выключения и включения питания сервера. Платформа виртуализации должна гарантировать, что перезагрузка выполняется успешно, даже когда базовый домен или виртуальные машины остановлены или приостановлены.

Плавающий IP-адрес IP-адрес, назначаемый проектом виртуальной машине таким образом, чтобы виртуальная машина получала один и тот же общедоступный IP-адрес при каждой загрузке. Можно создать пул плавающих IP-адресов и назначать их экземплярам при запуске, чтобы поддерживать постоянство IP-адреса и сохранять назначенные *DNS*-имена.

Под (Pod) базовая единица для управления и запуска приложений в Kubernetes. Под — это один или несколько контейнеров, которым гарантирован запуск на одном узле (то есть на одной физической или виртуальной машине), обеспечивается разделение ресурсов и предоставляется уникальный в пределах кластера IP-адрес. Последнее позволяет приложениям, развёрнутым на поде, использовать фиксированные и предопределённые номера портов без риска конфликта. Для совместного использования из контейнеров, развёрнутых в пределах под может быть определён — том (volume), представляющий из себя каталог на локальном диске или сетевой диск.

Программно-определяемая сеть/программно-конфигурируемая сеть (software-defined networking, SDN) сеть передачи данных, в которой уровень управления сетью отделен от устройств передачи данных и реализуется программно. Одна из форм виртуализации сети.

Программно-определяемые хранилища (Software-defined storage, SDS) системы хранения, базирующиеся на функционале программного обеспечения.

Проект/тенант контейнер для разделения ресурсов. Все ресурсы OpenStack должны принадлежать определенному проекту. В службе идентификации OpenStack проект должен принадлежать определенному домену.

Пространство имен (namespace) некоторое множество, под которым подразумевается модель, абстрактное хранилище или окружение, созданное для логической группировки уникальных идентификаторов (то есть имён).

Распределенный менеджер блокировок (Distributed lock manager, DLM) пакет программного обеспечения, который позволяет компьютерам в кластере координировать доступ к совместно используемым ресурсам. Работает на каждой машине в кластере, с идентичной копией базы данных блокировок кластера.

Регион является самым большим строительным блоком в облаке. Регионы позволяют объединять облачные платформы в единую инфраструктуру. Каждый регион представляет собой полноценную обособленную облачную платформу, которые объединяются единым сервером аутентификации и авторизации Keystone и панелью Horizon.

Репликация процесс, под которым понимается копирование данных с одного источника на другой (или на множество других) и наоборот.

Репозиторий место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети.

Сервис (в Мичман) это набор объединенных программных компонентов, решающих общую задачу, а также инфраструктурные ресурсы, необходимые для их размещения.

Сессия сеанс взаимодействия между пользователем и виртуальной TRS машиной.

Сетевой коммутатор устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети.

Символическая ссылка/симлинк специальный файл в файловой системе, в котором вместо пользовательских данных содержится путь к файлу, открываемому при обращении к данной ссылке или файлу.

Служба доменных имен (DNS) система, которая позволяет получить IP-адрес компьютера в Интернете, зная его доменное имя, и наоборот. Система DNS упрощает навигацию в Интернете, благодаря преобразованию IP-адресов в удобные для запоминания доменные имена. Например, IP-адрес

111.111.111.1 может быть преобразован в www.acloud.ru. Все домены и их компоненты, например почтовые серверы, используют службу DNS для поиска нужных IP-адресов компьютеров по их доменным именам. Как правило, информация о домене хранится на двух DNS-серверах — первичном и вторичном. При отказе первичного сервера, его заменяет вторичный. DNS-серверы можно объединять в кластеры или реплицировать таким образом, что изменения, внесенные на одном сервере, автоматически распространяются на все активные серверы. В службе вычислительных ресурсов DNS обеспечивает сопоставление DNS-записей с плавающими IP-адресами, узлами и ячейками, чтобы имена хостов оставались неизменными после перезагрузки.

Служба измерения производительности (Rally) проект OpenStack, предоставляющий платформу для анализа производительности и эффективности отдельных компонентов OpenStack, а также полномасштабных облачных развертываний OpenStack.

Служба обработки данных (Sahara) проект OpenStack, который предоставляет масштабируемый стек обработки данных и связанные интерфейсы управления.

Служба телеметрии (Telemetry) проект OpenStack, который собирает данные измерений об использовании физических и виртуальных ресурсов, входящих в развернутые облака. Служба телеметрии предоставляет эти данные для дальнейшего анализа и при достижении заданных показателей активизирует определенные действия.

Служба управления ключами (Barbican) проект OpenStack, который предоставляет систему хранения и создания секретов, обеспечивающую управление ключами для шифрования необходимых услуг.

Служба управления контейнерами (Magnum) проект OpenStack, который предоставляет платформу по оркестрации контейнеров, таких как Kubernetes, Docker Swarm или Mesos.

Смарт-карта пластиковая карта со встроенной микросхемой. В большинстве случаев смарт-карты содержат микропроцессор и операционную систему, управляющую устройством и контролирующую доступ к объектам в его памяти. Назначение смарт-карт - это одно- или двухфакторная аутентификация пользователей, хранение ключевой информации и проведение криптографических операций в доверенной среде.

Сниппет (Snippet) фрагмент исходного текста или кода программы, применяемый в поисковых системах, текстовых редакторах и средах разработки.

Соглашение об уровне обслуживания (SLA) оговоренные в контракте обязательства, которые гарантируют доступность службы.

Стек протоколов это иерархически организованный набор сетевых протоколов, достаточный для организации взаимодействия узлов в сети. Протоколы работают в сети одновременно, значит работа протоколов должна быть организована так, чтобы не возникало конфликтов или незавершённых операций. Поэтому стек протоколов разбивается на иерархически построенные уровни, каждый из которых выполняет конкретную задачу — подготовку, приём, передачу данных и последующие действия с ними.

Технология АМТ аппаратная технология, предоставляющая удаленный и внеполосный (по независимому вспомогательному каналу TCP/IP) доступ для управления настройками и безопасностью компьютера независимо от состояния питания (удаленное включение / выключение компьютера) и состояния ОС. Технология доступна в настольных ПК на базе процессоров Intel Core 2 и ноутбуках на базе процессоров Intel Centrino с технологией Intel vPro. Для подключения необходима система, имеющая набор микросхем с поддержкой технологии Intel АМТ.

Тип виртуальной машины представляет собой определенные параметры виртуальных машин, которые доступны для создания пользователем. Включает в себя такие параметры, как размер ОЗУ, диска, количество ядер и т.д.

-
- Токен** символьная строка, предназначенная для доступа к API и ресурсам. Токен выдается на ограниченное время и при необходимости может быть отозван до истечения срока действия. Токен также содержит список ролей, определяющих доступные пользователю полномочия.
- Точка входа (endpoint)** URL для получения доступа к определенному сервису. Каждый сервис может иметь одну или несколько точек входа.
- Траблшутинг (troubleshooting)** форма решения проблем, часто применяемая к ремонту неработающих устройств или процессов. Представляет собой систематический, опосредованный определённой логикой, поиск источника проблемы с целью её решения. Траблшутинг как поиск и устранение неисправностей необходим для поддержания и развития сложных систем, где проблема может иметь множество различных причин.
- Трассировка** процесс пошагового выполнения программы. В режиме трассировки пользователь видит последовательность выполнения команд и значения переменных на данном шаге выполнения программы, что позволяет легче обнаруживать ошибки. Трассировка может быть начата и окончена в любом месте программы, выполнение программы может останавливаться на каждой команде или на точках останова, трассировка может выполняться с заходом в процедуры и без заходов, а также осуществляться в обратном порядке (шаг назад).
- Удалённый вызов процедур (Remote Procedure Call, RPC)** класс технологий, позволяющих компьютерным программам вызывать функции или процедуры в другом адресном пространстве (как правило, на удалённых компьютерах).
- Управление доступом на основе ролей (Role Based Access Control, RBAC)** политика избирательного управления доступом, при этом права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя роли.
- Управляющий узел/контроллер** узел, осуществляющий управление облаком. На узле располагаются коммуникационные сервисы, которые обеспечивают работу всего облака, в том числе сервер очередей, база данных, панель управления Horizon, система мониторинга. Также на узле могут располагаться сервис nova-scheduler и API-серверы, балансировкой распределения нагрузки на которые управляет конечный узел.
- Фиксированный IP-адрес** IP-адрес, связываемый с одной и той же виртуальной машиной при каждой загрузке, как правило, недоступный для конечных пользователей Интернета и используемый для управления виртуальной машиной.
- Формат диска** базовый формат, в котором образ диска для виртуальной машины хранится во внутреннем хранилище службы образов. Например, AMI, ISO, QCOW2, VMDK и др.
- Формат контейнера** оболочка, используемая службой образов, которая содержит образ виртуальной машины и связанные метаданные, например, состояние компьютера, размер диска, операционной системы и др.
- Хранилища проверки доступности** хранилища, которые подключаются к вычислительным узлам и к контроллеру с установленным CloudManager, и с помощью средств модуля CloudManager используются для дополнительной проверки доступности хоста вычислительного узла. Если дополнительная проверка с помощью хранилищ проводится успешно, то механизм автоэвакуации к данному узлу применен не будет.
- Центр сертификации (Certification authority, CA)** криптографический термин, означающий сторону, чья честность неоспорима, а открытый ключ широко известен. Задача центра сертификации - подтверждать подлинность ключей шифрования с помощью сертификатов электронной подписи.
- Чрезмерное выделение ОЗУ (RAM overcommit)** возможность запускать новые экземпляры виртуальных машин, исходя из действительного использования памяти хоста, а не на основе объема ОЗУ, выделенного для каждого выполняемого экземпляра. Также известно как чрезмерное выделение памяти (memory overcommit).
-

Шлюз IP-адрес, как правило, назначаемый маршрутизатору, который передает сетевой трафик другим сетям.

Экстент в файловых системах непрерывная область носителя информации. Как правило, в файловых системах с поддержкой экстентов большие файлы состоят из нескольких экстентов, не связанных друг с другом напрямую.

7.1 Версия 1.0

Дата выпуска версии	05.11.24
Версия пакетов	1.0
Поддерживаемая версия OpenStack	Bobcat

Важно: В текущей версии продуктов:

- Выпущен дистрибутив, объединяющий самые важные свойства дистрибутивов Асперитас (Единый реестр российского ПО №5921) и AccentOS (Единый реестр российского ПО №6416)
- Дистрибутив облачной платформы собирается на изолированной сборочной ферме автоматическим образом: исходные коды всех компонентов располагаются в контролируемом контуре в формате git-репозитория. Переход на частичное замыкание кодовой базы планируется в релизе 1.1; полное замыкание в релизе 1.2.

- Произведен перевод всех компонентов Openstack на релиз 2023.2 (Bobcat).
- Обновлена система установки дистрибутива на базе TripleO. TripleO также прошёл сложную процедуру обновления и исправления ошибок апстрима.
- Добавлен собственный веб-интерфейс на базе переработанного Skyline.
- Добавлены все основные модули AccentOS (RS-server/TRS, Journal, Agent, MQ, Client, CloudManager, Scheduler, OTP-sender). Как следствие, в совместном дистрибутиве появились DRS, автоматическая эвакуация и полноценный VDI, а также ряд вспомогательных возможностей.
- VDI-движок способен помимо использования штатных протоколов трансляции ОС также и использовать собственный движок трансляции Wormhole со следующими возможностями:
 - – поддержка нескольких мониторов;
 - – режимы редактора и наблюдателя;
 - – возможность настройки параметров трансляции;
 - – низкие задержки; использование видеокарты для ускорения трансляции;
 - – WebRTC и работа из браузера
- Перенесены все компоненты Асперитас, за исключением механизмов безопасности, сделанных специально для ФСТЭК. Они также появятся после ряда технических работ по минимизации дистрибутива.
- Добавлено безопасное ФСТЭК-ядро 6.1

7.2 Версия 1.1

Дата выпуска версии	28.12.24
Версия пакетов	1.1
Поддерживаемая версия OpenStack	Bobcat

Важно: В текущей версии продуктов:

- Повышена стабильность системы.
- Доработан веб-интерфейс.
- Все модули Aos расширили покрытие функциональности через веб-интерфейс.

- – Модуль Scheduler теперь отображается в веб-интерфейсе.
- – Появилась функциональность копирования запущенных инстансов.
- Serp обновлён до версии 18.2 и проверены способы его развёртывания.
- В системе развёртывания Dctui повышена общая стабильность и появился неинтерактивный режим запуска; происходит постепенный переход на собственный фреймворк Deckcrew.
- В Wormhole появилась подробная трассировка событий и в целом улучшена производительность, а также реализован в базовом виде автозапуск.
- В рамках оркестратора Мичман актуализирован и стабилизирован деплой Kubernetes, а также актуализированы примеры предоставления СУБД как сервиса (DBaaS).

Версия пакетов	Дата выпуска	Дата корректировки	Поддерживаемая версия OpenStack	Поддерживаемые операционные системы	Версия с длительной поддержкой
<i>1.0</i>	5 ноября 2024		Bobcat	AOSBCP	
<i>1.1</i>	28 декабря 2024		Bobcat	AOSBCP	

Версии документации и дистрибутива коррелируют. Например, документация версии 1.1.x предназначена для всех релизов дистрибутива 1.1 (1.1.0, 1.1.1, 1.1.2 и т.д.). Релизный цикл дистрибутива составляет примерно 2 месяца.

Вся документация в формате PDF доступна по [ссылке](#).

Служба технической поддержки: support@acloud.ru

СИМВОЛЫ

- Агент ДНСР, **145**
- Агрегатор узлов, **145**
- Аутентификация, **146**
- Автоматическая (принудительная) миграция, **145**
- Автоматическая эвакуация, **145**
- Балансировка нагрузки, **146**
- База данных, **146**
- Бесклассовая адресация (*Classless Inter-Domain Routing, CIDR*), **146**
- Брокер сообщений, **146**
- Центр сертификации (*Certification authority, CA*), **153**
- Чрезмерное выделение ОЗУ (*RAM overcommit*), **153**
- Дамп памяти, **148**
- Дата-центр, **148**
- Даунгрейд, **148**
- Дедупликация, **148**
- Дельта образа, **148**
- Демон (*daemon*), **148**
- Диск/том, **148**
- Домен, **148**
- Двухфакторная аутентификация, **148**
- Экстент, **154**
- Фиксированный IP-адрес, **153**
- Формат диска, **153**
- Формат контейнера, **153**
- Гипервизор, **147**
- Главная загрузочная запись (*master boot record, MBR*), **147**
- Горизонтальное масштабирование, **147**
- Горячая замена (*HotPlug*), **147**
- Графический интерфейс пользователя (*Graphical user interface/GUI*), **147**
- Группа безопасности, **147**
- Хранилища проверки доступности, **153**
- Интерфейс IPMI, **148**
- Интерфейс командной строки (*Command line interface, CLI*), **148**
- Интерфейс пользователя (*User interface/UI*), **149**
- Качество обслуживания (*QoS*), **149**
- Кластер, **149**
- Лог файл, **149**
- Логирование, **149**
- Маршрутизатор, **149**
- Менеджер логических томов/LVM, **149**
- Метрика программного обеспечения (*software metric*), **149**
- Межпроцессное взаимодействие (*inter-process communication, IPC*), **149**
- Многопутевой ввод-вывод (*Multipath I/O*), **149**
- Мьютекс, **149**
- Облачное хранилище данных, **150**
- Обратный прокси-сервер (*reverse proxy*), **150**
- Образ виртуальной машины, **150**
- Образ загрузочного диска, **150**
- Отказоустойчивость, **150**
- Пауза, **150**
- Перезагрузка сервера, **150**
- Плавающий IP-адрес, **150**
- Под (*Pod*), **151**
- Проект/тенант, **151**
- Программно-определяемые хранилища (*Software-defined storage, SDS*), **151**
- Пространство имен (*namespace*), **151**
- Распределенный менеджер блокировок (*Distributed lock manager, DLM*), **151**

Регион, [151](#)
 Репликация, [151](#)
 Репозиторий, [151](#)
 Сервис (*в Мичман*), [151](#)
 Сессия, [151](#)
 Сетевой коммутатор, [151](#)
 Символическая ссылка/симвлинк, [151](#)
 Система управления элементами сети (*Element Management System, EMS*), [146](#)
 Служба доменных имен (*DNS*), [151](#)
 Служба измерения производительности (*Rally*), [152](#)
 Служба обработки данных (*Sahara*), [152](#)
 Служба телеметрии (*Telemetry*), [152](#)
 Служба управления ключами (*Barbican*), [152](#)
 Служба управления контейнерами (*Magnum*), [152](#)
 Смарт-карта, [152](#)
 Снимок (*Snippet*), [152](#)
 Соглашение об уровне обслуживания (*SLA*), [152](#)
 Стек протоколов, [152](#)
 Шлюз, [154](#)
 Технология АМТ, [152](#)
 Тип виртуальной машины, [152](#)
 Точка входа (*endpoint*), [153](#)
 Токен, [153](#)
 Трешлинг (*troubleshooting*), [153](#)
 Трассировка, [153](#)
 Удалённый вызов процедур (*Remote Procedure Call, RPC*), [153](#)
 Управление доступом на основе ролей (*Role Based Access Control, RBAC*), [153](#)
 Управляющий узел/контроллер, [153](#)
 VM/Виртуальная машина, [147](#)
 ВУ/вычислительный узел, нода, [147](#)
 Вертикальное масштабирование, [146](#)
 Виртуализация сетевых функций (*Network Functions Virtualization, NFV*), [146](#)
 Виртуальная файловая система (*Virtual file system — VFS*), [147](#)
 Виртуальная машина на основе ядра (*KVM*), [147](#)
 Золотой образ, [148](#)
 Зона доступности, [148](#)
 Живая миграция, [148](#)

А

ACL (*Access Control List*), [133](#)
 ACPI, [133](#)
 Active Directory, [133](#)
 Amphora, [133](#)
 AMQP, [133](#)
 Ansible, [134](#)
 Aodh, [134](#)
 Apache Tomcat, [134](#)

ASCII, [134](#)
 AWS (*Amazon Web Services*), [135](#)

В

Bareos, [135](#)
 BGP (*Border Gateway Protocol*), [135](#)

С

Ceilometer, [135](#)
 Celery, [135](#)
 Ceph, [135](#)
 CephFS, [135](#)
 CGI, [135](#)
 Chef, [135](#)
 chroot, [135](#)
 Cinder, [135](#)
 Cloud-init, [135](#)
 Cobbler, [135](#)
 Common Internet File System (*CIFS*), [135](#)
 Consul, [135](#)
 CUPS (*Common UNIX Printing System*), [136](#)
 cURL, [136](#)

Д

DCC (*Dynamic Configuration Components*), [136](#)
 Device mapper (*dm*), [136](#)
 DevStack, [136](#)
 DHCP, [136](#)
 Direct SCSI, [136](#)
 dm-cache, [136](#)
 Docker, [136](#)
 Docker Compose, [136](#)
 Dockerfile, [136](#)
 Domain Name, [136](#)
 DRS (*Distributed Resource Scheduling*), [136](#)
 DSCP (*Differentiated Services Code Point*), [136](#)

Е

ELK стек, [136](#)

Ф

FreeRDP, [137](#)
 Fully Qualified Domain Name (*FQDN*), [137](#)

G

Ganglia, 137
 Gatling, 137
 Glance, 137
 GlusterFS, 137
 Gnocchi, 137
 GNU Privacy Guard (*GnuPG*, *GPG*), 137
 Google Authenticator, 137
 GRUB, 137
 GUID, 137
 Unicorn, 137

H

Hadoop, 137
 HAProxy, 137
 Heat, 138
 httpd, 138

I

IKE (*Internet Key Exchange*), 138
 Initrd, 138
 IP Fabric, 138
 iptables, 138
 IPXE, 138
 iSCSI, 138
 iSCSI target, 138

J

Jenkins, 138
 JMeter, 138
 jQuery, 138

K

Kerberos, 138
 Keystone, 138
 Kubernetes, 138

L

LBaaS, 138
 LDAP, 139
 libvirt, 139
 LightDM, 139
 LizardFS, 139
 LUN (*Logical Unit Number*), 139

M

MAC-адрес, 139
 Management Information Base (*MIB*), 139
 MATE, 139
 mdadm, 139
 Memcached, 139
 Mistral, 139
 MODBUS, 139
 MongoDB, 139

N

Network File System (*NFS*), 139
 Network NameSpace (*netns*), 140
 Neutron, 140
 NFV Orchestrator (*оркестратор NFV*), 146
 NFVI (*Network Function Virtualization Infrastructure*), 146
 Nginx, 140
 NodeSafe, 140
 Nova, 140
 NUMA (*Non-Uniform Memory Access*), 140

O

OCFS (*Oracle Cluster File System*), 140
 Octavia, 140
 OID, 140
 Open vSwitch, 140
 OpenLDAP, 140
 OpenSSL, 140
 OpenStack, 140
 Openstack Client, 140
 OpenStack Horizon/Dashboard, 141
 OSD (*Object Storage Device*), 141
 oVirt, 141

P

pbr (*Python Build Reasonableness*), 141
 PEP 440, 141
 pip, 141
 pip3, 141
 PostgreSQL, 141
 PulseAudio, 141
 PXE (*Preboot eXecution Environment*), 141
 PyPI (*Python Package Index*), 141

Q

QEMU Copy On Write 2 (*QCOW2*), [141](#)
 Qt, [141](#)
 Quick EMUlator (*QEMU*), [141](#)

R

RAID (*избыточный массив независимых дисков*), [141](#)
 RAW, [141](#)
 RBAC (*Role Based Access Control*), [142](#)
 RDP, [142](#)
 Redis, [142](#)
 REST API, [142](#)
 root/суперпользователь, [142](#)
 RSA, [142](#)
 rsync, [142](#)

S

SaltStack, [142](#)
 Samba, [142](#)
 Scala, [142](#)
 SCP, [142](#)
 SCSI, [142](#)
 Selenium, [142](#)
 SELinux, [142](#)
 Sentry, [142](#)
 Server URI, [142](#)
 Snapshot, [143](#)
 SNMP, [143](#)
 snmpd, [143](#)
 SPICE, [143](#)
 SSH, [143](#)
 SSL, [143](#)
 sudo, [143](#)
 Supervisor, [143](#)
 Swift, [143](#)
 Syslog (*system log*), [143](#)
 systemd, [143](#)

T

Tacker, [143](#)
 TCP, [143](#)
 TDD (*test-driven development*), [143](#)
 Tempest, [143](#)
 TLS, [144](#)
 TOTP (*Time-based One-Time Password Algorithm*), [144](#)

Tox, [144](#)

U

UTC, [144](#)
 UUID (*universally unique identifier*), [144](#)
 uWSGI, [144](#)

V

vCPU, [144](#)
 VDI (*Virtual Desktop Infrastructure*), [144](#)
 VDO (*Virtual Data Optimizer*), [144](#)
 virt-top, [144](#)
 Virtual Extensible LAN (*VXLAN*), [144](#)
 Virtualized Infrastructure Manager (*менеджер виртуальной инфраструктуры*), [147](#)
 VLAN (*Virtual Local Area Network*), [144](#)
 VMTP, [144](#)
 VNC, [145](#)
 VNF (*Virtual Network Function*), [146](#)
 VNF Manager (*VNFМ*), [146](#)

W

Wake-on-LAN, [145](#)
 Watcher, [145](#)
 World Wide Name (WWN)/World Wide Identifier (*WWID*), [145](#)
 WSGI (*Web Server Gateway Interface*), [145](#)

X

X Window System (*X11,X*), [145](#)
 X.Org Server (*Xorg*), [145](#)
 X2Go, [145](#)
 XFS, [145](#)
 XRDP, [145](#)

Z

Zabbix, [145](#)
 ZRAM, [145](#)