Программное обеспечение «1Stack (1Стек)»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

(новая редакция)

2024

Содержание

1 Общие сведения

Для установки **1Stack** необходимо:

1. Один (1) физический сервер Jumphost.
2. Три (3) физических сервера управления.
3. Один (1) физический вычислительный сервер и более.

На сервере Jumphost устанавливаются:

- виртуальная машина Undercloud;
- опционально: репозиторий **1Stack**, файервол для выхода в интернет, серверы для резервного копирования, бастион для Kubernetes, Syslog-сервер.

На каждом сервере управления устанавливаются:

- виртуальные машины контроллеров;
- опционально: серверы DNS, NTP, LDAP;
- мониторинг.

Undercloud – главный сервер управления, содержащий набор инструментов OpenStack TripleO. Это односерверная установка OpenStack, включающая компоненты для подготовки и управления продуктивными серверами OpenStack, которые составляют среду OpenStack (overcloud). Формирующие Undercloud компоненты выполняют несколько функций:

1. Планирование

В состав Undercloud входят функции планирования, которые можно использовать для создания и назначения определенных ролей вычислительных серверов. Также имеется набор ролей серверов по умолчанию, которые можно назначать конкретным серверам: вычислительным и серверам для управления. Также можно создавать собственные роли. Кроме того, можно выбрать, какие сервисы платформы **1Stack** нужно включить в каждую роль сервера. Это позволяет создавать новые типы серверов.

2. Управление физическими серверами

Undercloud использует интерфейс управления out-of-band, обычно Intelligent Platform Management Interface (IPMI) или Redfish каждого сервера для управления питанием, и службу на основе PXE для определения характеристик оборудования и установки OpenStack.

3. Оркестрация

Undercloud содержит набор шаблонов YAML – это набор планов для облачной платформы. Undercloud загружает эти планы и выполняет их инструкции для создания продуктивного решения OpenStack. Параметры планов можно переопределить, если нужно включить собственные настройки.

Overcloud – продуктивное облачное решение **1Stack**, которое создает Undercloud.

Overcloud состоит из нескольких серверов с различными ролями, которые определяются на основе конфигурационных файлов TripleO.

Undercloud включает в себя набор ролей серверов Overcloud по умолчанию:

1. Роль управления

Серверы управления обеспечивают администрирование, сетевое взаимодействие и высокую доступность среды OpenStack. OpenStack содержит три сервера управления, объединенных в кластер высокой доступности.

Не все эти службы включены по умолчанию. Для включения некоторых из них требуется настроенные environment-файлы.

2. Вычислительная роль

Вычислительные серверы предоставляют вычислительные ресурсы для среды OpenStack. Добавить дополнительные вычислительные серверы для расширения среды можно в любой момент.

2 Требования для развертывания

В разделе подробно описана спецификация аппаратного и программного обеспечения для серверов, которые необходимо использовать для размещения виртуальных машин undercloud и overcloud, а также вычислительных серверов.

2.1 Требования к инфраструктуре

2.1.1 Спецификации физического серверного оборудования

В таблице указаны рекомендуемые физические спецификации всех типов серверов:

Сервер управления и jumphost

Ядер ЦПУ на сервер	52 (2 ЦПУ по 26 ядер), 64-разрядный процессор x86 с поддержкой расширений Intel 64 или AMD64 и включенными расширениями аппаратной виртуализации AMD-V или Intel VT
Память (ГБ)	384
Диски	4 x 1,92 ТБ (SSD-диск); RAID10
Сетевые порты	4 x 25G портов для управления инфраструктурой, управления хранилищем (Infrastructure, Storage)
	2 x 1G порта для управления out-of-band (для гипервизора) (Infrastructure OM)
	1 x 1G порт для IPMI (управление out-of-band для физических серверов)

Вычислительный сервер

Ядер ЦПУ на сервер	52 (2 ЦПУ по 26 ядер), 64-разрядный процессор x86 с поддержкой расширений Intel 64 или AMD64 и включенными расширениями аппаратной виртуализации AMD-V или Intel VT
Память (ГБ)	384
Диски	2 x 960ГБ (SSD-диск); RAID1
Сетевые порты	2 x 25G портов для управления инфраструктурой, управления хранилищем и управления out-of-band (Infrastructure, Infrastructure OM, Storage)
	4 x 25G портов для управления данными приложения (Application) - один сетевой интерфейс с двумя портами 25G на каждую NUMA
	1 x 1G порт на IPMI (управление out-of-band для физического сервера)
Другое	

2.1.2 Спецификация виртуальной машины Undercloud

Виртуализация виртуальной машины undercloud выполняется с помощью KVM. При создании виртуальной машины используется следующая спецификация:

1. 16 виртуальных процессоров.
2. Оперативная память 64 ГБ.
3. В общей сложности 450 ГБ свободного места на диске.
4. Два (2) виртуальных контроллера сетевого интерфейса (vNIC) для работы с нужными сетями:
   - *Сеть управления (Provisioning, Control Plane)* – сеть, которую виртуальные машины undercloud используют для подготовки серверов и доступа к ним через SSH при выполнении конфигурации Ansible. Эта сеть также обеспечивает SSH-доступ от виртуальной машины undercloud к серверам Overcloud. Виртуальная машина undercloud содержит сервисы DHCP для интроспекции и подготовки других серверов в этой сети. Это означает, что в сети не должны существовать другие сервисы DHCP. Этот интерфейс будет настроен автоматически при установке undercloud.
   - *Публичная/Внешняя сеть* – обеспечивает доступ к хранилищам платформы OpenStack, источникам образов контейнеров и другим серверам, например DNS или NTP. Эта сеть используется для стандартного доступа к виртуальной машине undercloud с рабочей станции пользователя. Для доступа к внешней сети необходимо вручную настроить интерфейс на виртуальной машине undercloud.
5. ОС МСВСфера версии 8.5 и выше, установленная в качестве основной операционной системы.
6. SELinux установлена в режиме Enforcing на хосте.

2.1.3 Спецификация виртуальных серверов управления

Виртуальные машины серверов управления создаются на базе KVM. При создании виртуальной машины используется следующая спецификация:

1. 16 виртуальных процессоров.
2. Оперативная память 64 ГБ.
3. 450 ГБ свободного места на диске.
4. 6 виртуальных сетевых интерфейсов (vNIC).

2.1.4 DNS- и NTP-серверы

Для корректного развертывания и работы Undercloud и Overcloud необходимы DNS- и NTP-серверы, которые должны являться частью инфраструктуры.

2.1.5 LDAP и сертификат TLS

Для авторизации и в целях безопасности должны использоваться протокол LDAPS и сертификат TLS.

Undercloud и Overcloud должны быть настроены на использование серверов LDAP (Active Directory или OpenLDAP).

2.2 Требования к сети

Все физические серверы должны быть подключены к сетевой фабрике. Настройку и установку фабрики необходимо выполнить перед установкой undercloud и overcloud. Серверы каждого типа (Jumphost, сервер управления, вычислительный сервер) имеют разные возможности подключения в соответствии с типом сервера, а также подключение к различным VLAN-сетям.

На фабрике должны быть созданы следующие сети:

- OOB-SRV – сеть out-of-band для всех физических серверов;
- OOB-IND – сеть out-of-band для сетевой фабрики (опционально);
- OOB-HYP – сеть out-of-band гипервизоров (Jumphost и серверы управления);
- OOB-II – сеть для Undercloud;
- VIM-PRVS – сеть для установки и обслуживания Overcloud (provisioning);
- VIM-MGMT – сеть для управления Overcloud (management);
- VIM-INT – сеть Internal API overcloud;
- VIM-EXT- сеть External API overcloud;
- VIM-OVRL – сеть Tenant overcloud;
- CEPH-PUB-CL-0 – сеть для доступа к Ceph;
- INFR – инфраструктурная сеть;
- INFR-OAM – инфраструктурная сеть для мониторинга (опционально);
- INFR-BCKP – инфраструктурная сеть для резервного копирования (опционально);
- CISM-INT – сеть для Kubernetes (опционально);
- FW-IN – внутренняя сеть для межсетевого экрана (опционально);
- FW-OUT – внешняя сеть для межсетевого экрана (опционально);
- PUBLIC – клиентская сеть провайдера.

Серверы должны быть подключены в следующие сети:

1. Сервер Jumphost:
   - OOB-SRV;
   - OOB-IND (опционально);
   - OOB-HYP;
   - OOB-II;
   - VIM-PRVS;
   - CEPH-PUB-CL-0;
   - INFR;
   - INFR-OAM (опционально);

- INFR-BCKP (опционально);
- CISM-INT (опционально);
- FW-IN (опционально);
- FW-OUT (опционально).

BM undercloud:

- OOB-II;
- VIM-PRVS.

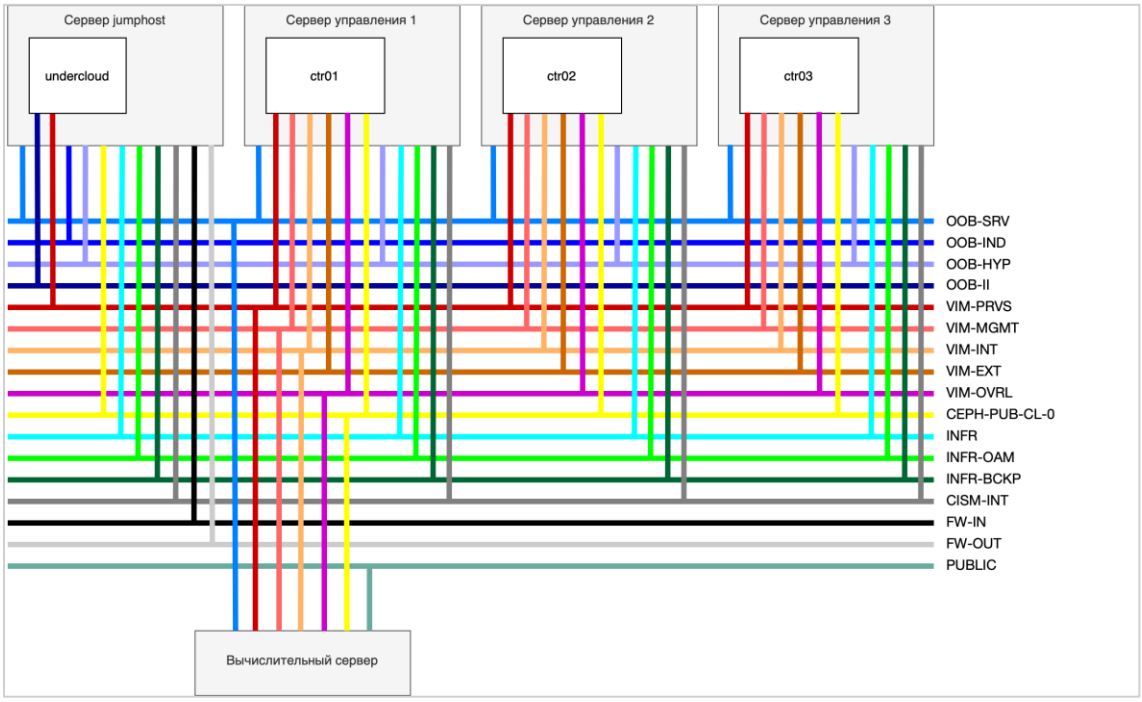
2. Серверы управления:
   - OOB-SRV;
   - OOB-HYP;
   - OOB-II;
   - VIM-PRVS;
   - VIM-MGMT;
   - VIM-INT;
   - VIM-EXT;
   - VIM-OVRL;
   - CEPH-PUB-CL-0;
   - INFR;
   - INFR-OAM (опционально);
   - INFR-BCKP (опционально);
   - CISM-INT (опционально).

BM серверов управления:

- VIM-PRVS;
- VIM-MGMT;
- VIM-INT;
- VIM-EXT;
- VIM-OVRL;
- CEPH-PUB-CL-0;

3. Вычислительный сервер;
   - OOB-SRV;
   - VIM-PRVS;
   - VIM-MGMT;
   - VIM-INT;
   - VIM-OVRL;
   - CEPH-PUB-CL-0;
   - PUBLIC.

Схематично подключение серверов показано на рисунке:



2.3 Кластер Ceph

Кластер Ceph используется в качестве внутреннего хранилища и виртуальной машины с репозиторием (1Stack Repo). Поэтому кластер необходимо установить и настроить до их инсталляции.

3 Развертывание Undercloud

3.1 Инсталляция виртуальной машины Undercloud

BM Undercloud устанавливается на сервер jumphost.

Установка выполняется в следующем порядке:

1) Подключитесь к серверу Jumphost и скачайте скрипты createvms.zip; распакуйте их:

```
# wget <IP 1Stack-repo>:/createvms.zip
# unzip createvms.zip
```

2) В директории create_vms создайте файл vim_und, подставив свои значения:

```
VM_NAME="tc-v-vim-und01"
VCPUS="16"
MEM_SIZE="64"
DISK_SIZE="300"
IP_ADDRESS="10.10.10.10"
NETMASK_PREFIX="24"
IP_GATEWAY="10.10.10.254"
NET="104,210"
```

Где 104,210 – номера VLAN сетей OOB-II и VIM-PRVS.

3) Установите BM Undercloud:

```
# ./create_vms/create_vm.sh vim_und
```

4) После установки ОС зайдите в веб-интерфейс Cockpit по адресу https://<IP jumphost>:9090 и включите второй интерфейс у BM.

5) Скачайте вспомогательные скрипты tcloud-postinstall.zip; распакуйте их:

```
# wget <IP 1Stack-repo>:/tcloud-postinstall.zip
# unzip tcloud-postinstall.zip
```

6) Установите необходимое ПО. Для этого:

a) Создайте файл tcloud-postintall/undercloud.yaml:

```
---
- name: postinstall
  hosts: undercloud
  become: yes
  gather_facts: yes
  roles:
    - repo
    - cacert
  post_tasks:
    - name: postinstall - Reboot
      ansible.builtin.reboot:
      when: final_reboot == True
```

При необходимости можно указать дополнительные роли из директории tcloud-postintall.

б) В файле tcloud-postintall/inventory укажите пароль **root** сервера Undercloud.

в) Запустите виртуальное окружение:

```
# source ansible/venv/bin/activate
```

г) Запустите плейбук:

```
# cd tcloud-postintall
# ansible-playbook -i inventory.ini undercloud.yaml
```

7) Зайдите в BM Undercloud под учетной записью пользователя **root** и создайте скрипт **und.sh**:

```
#!/bin/bash

mkdir -p /etc/cni/net.d

useradd stack
echo <password> | passwd stack --stdin

echo "stack ALL=(root) NOPASSWD:ALL" > /etc/sudoers.d/stack
chmod 0440 /etc/sudoers.d/stack

su - -c "mkdir ~/images" stack

dnf config-manager --set-disabled epel*
dnf config-manager --set-enabled vault*

cat << EOF > /etc/yum.repos.d/rdo-local.repo
[delorean-component-baremetal]
name=delorean-openstack-ironic-36f31050169cc33c23a3edcae9335952065b64e0
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-baremetal
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-cinder]
name=delorean-openstack-cinder-c093edab2d11938ac88cbde3323b90d50e8120a6
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-cinder
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-clients]
name=delorean-ansible-collections-openstack-
41ad425d230bf9ea9a0b2f47123e032021359f51
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-clients
enabled=1
gpgcheck=0
priority=1
sslverify=0
```

```
[delorean-component-cloudops]
name=delorean-gnocchi-c757cb4a95f19d6d445bfad87376d9b72255b851
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-cloudops
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-common]
name=delorean-ansible-role-collect-logs-
ed6b3e37e892cabc8179ccca6b86d88d60c8abf6
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-common
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-compute]
name=delorean-openstack-nova-a5da31ec1ea1d1c7b4df146857982699ebdc328e
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-compute
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-glance]
name=delorean-python-glance-store-79e043af2a86f4547f8e7bad6be90ff4542e005e
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-glance
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-manila]
name=delorean-openstack-manila-9e277158acf5fbf0bab954cf46f6f56bf1ef3b84
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-manila
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-network]
name=delorean-python-networking-ovn-33674b8a34992be7aa6f387266ad90a9db03a025
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-network
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-octavia]
name=delorean-openstack-octavia-bbf18f97ea1e18faa1bb9faf98fd03706e8344ff
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-octavia
enabled=1
```

```
gpgcheck=0
priority=1
sslverify=0

[delorean-component-security]
name=delorean-ansible-tripleo-ipa-060a3939cf08cd132bcf2b73c08146de6d674a78
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-security
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-swift]
name=delorean-openstack-swift-2bffa1e205c499768abd767f0ec4d6f0dc745e85
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-swift
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-tempest]
name=delorean-openstack-tempest-271f8201d4d927830d289c34744ca20a92bee33c
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-tempest
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-tripleo]
name=delorean-openstack-tripleo-heat-templates-
7c89b1601eca2a09a55ab32c201c4fc3a7f2a383
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-tripleo
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-ui]
name=delorean-openstack-trove-ui-10819cf36fa27e80f60075d89b98e9c8decf630b
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-ui
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-validation]
name=delorean-openstack-tripleo-validations-
de6d1ab4d48ea7eb4ccdeb203bb074ce296f4421
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-validation
enabled=1
gpgcheck=0
priority=1
sslverify=0
EOF
```

```
cat << EOF > /etc/yum.repos.d/rdo-deps-local.repo
[delorean-train-testing]
name=dlrn-train-testing
baseurl=https://<IP 1Stack Repo>/repo/rdo-deps/centos8-train/x86_64/delorean-train-testing
enabled=1
gpgcheck=0
module_hotfixes=1
sslverify=0
priority=1000

[delorean-train-build-deps]
name=dlrn-train-build-deps
baseurl=https://<IP 1Stack Repo>/repo/rdo-deps/centos8-train/x86_64/delorean-train-build-deps
enabled=1
gpgcheck=0
module_hotfixes=1
sslverify=0
priority=1000

[advanced-virtualization]
name=Advanced Virtualization mirror
baseurl=https://<IP 1Stack Repo>/repo/rdo-deps/centos8-train/x86_64/advanced-virtualization
gpgcheck=0
enabled=1
module_hotfixes=1
sslverify=0
priority=1000

[centos-rabbitmq-38]
name=Messaging RabbitMQ
baseurl=https://<IP 1Stack Repo>/repo/rdo-deps/centos8-train/x86_64/centos-rabbitmq-38
gpgcheck=0
enabled=1
module_hotfixes=1
sslverify=0
priority=1000
skip_if_unavailable=1

[centos-opstools]
name=opstools
baseurl=https://<IP 1Stack Repo>/repo/rdo-deps/centos8-train/x86_64/centos-opstools
gpgcheck=0
enabled=1
module_hotfixes=1
sslverify=0
priority=1000

[centos-8-fix]
name=centos-8-fix
baseurl=https://<IP 1Stack Repo>/repo/rdo-deps/centos8-train/x86_64/centos-8-fix
enabled=1
gpgcheck=0
module_hotfixes=1
sslverify=0
```

```
priority=1000

[centos-ceph-pacific]
name=centos-ceph-pacific
baseurl=https://<IP 1Stack Repo>/repo/ceph/rocky/8/pacific/x86_64/centos-ceph-pacific/
enabled=1
gpgcheck=0
module_hotfixes=1
sslverify=0
priority=1000
EOF

dnf install -y python3-tripleoclient ceph-ansible

echo DONE
```

где укажите пароль пользователя stack undercloud и IP BM с репозиторием

8) Запустит **und.sh**:

```
./und.sh
```

3.2 Подготовка развертывания и инсталляция Undercloud

Затем необходимо подготовить файлы конфигурации. Существует три основных файла конфигурации (см. описания далее):

1) Создайте директорию undercloud-files и необходимые файлы под учетной записью пользователя stack:

```
$ mkdir undercloud-files
$ touch undercloud-files/undercloud.conf
$ touch undercloud-files/undercloud-containers.yaml
$ touch undercloud-files/undercloud-advanced.yaml
$ ln -s undercloud-files/undercloud.conf undercloud.conf
```

2) Сделайте настройки в файле undercloud.conf, например:

```
[DEFAULT]
clean_nodes = true
container_images_file = undercloud-files/undercloud-containers.yaml
generate_service_certificate = false
custom_env_files = undercloud-files/undercloud-advanced.yaml
local_ip = 20.10.10.1/24
local_mtu = 9000
overcloud_domain_name = tcloud.example.ru
undercloud_debug = false
undercloud_nameservers = 30.10.10.11,30.10.10.12
undercloud_ntp_servers = 30.10.10.11,30.10.10.12
undercloud_timezone = Europe/Moscow

[ctlplane-subnet]
cidr = 20.10.10.0/24
dhcp_start = 20.10.10.10/24
dhcp_end = 20.10.10.200/24
```

```
gateway = 20.10.10.1
inspection_iprange = 20.10.10.201,20.10.10.254/24
```

3) Подставьте свои значения

Описание параметров:

clean_nodes	Определяет необходимость очистки жесткого диска между развертываниями и после интроспекции
container_images_file	Файл среды Heat с информацией об образе контейнера. Этот файл может содержать следующие записи: ▪ параметры всех необходимых образов контейнера; ▪ параметр *ContainerImagePrepare* – используется для подготовки необходимого образа. Как правило, файл с этим параметром называется *containers-prepare-parameter.yaml*
custom_env_files	Дополнительный файл среды, который нужно добавить в установку
local_ip	IP-адрес, определенный для сетевого интерфейса для установки Undercloud (Provisioning NIC). Этот IP-адрес также используется сервером для сервисов загрузки DHCP и PXE
local_mtu	Максимальная единица передачи (MTU), которую нужно использовать для local_interface
overcloud_domain_name	Доменное имя DNS, которое нужно использовать при развертывании Overcloud
undercloud_debug	Устанавливает уровень ведения журнала сервисов Undercloud на DEBUG. Установите для этого значения значение true, чтобы активировать уровень журнала DEBUG
cidr	Сеть, которую Undercloud использует для управления серверами Overcloud
dhcp_start; dhcp_end	Начало и конец диапазона распределения DHCP для серверов Overcloud. Убедитесь, что этот диапазон содержит достаточно IP-адресов для размещения всех используемых серверов
gateway	Шлюз для инстансов Overcloud. Это хост Undercloud, который перенаправляет трафик во внешнюю сеть
inspection_iprange	Временный диапазон IP-адресов для серверов в этой сети, который будет использоваться в процессе интроспекции. Этот диапазон не должен пересекаться с диапазоном, определенным dhcp_start и dhcp_end, но должен находиться в той же IP-подсети

4) Сделайте настройки в файле undercloud-containers.yaml, например:

```
parameter_defaults:
  ContainerImagePrepare:
  - set:
      name_prefix: centos-binary-
      name_suffix: ''
      namespace: <IP 1Stack Repo>:8082/1Stackv1
      neutron_driver: null
      rhel_containers: false
      tag: current-tripleo
```

5) Подставьте свои значения.

Описание параметров:

name_prefix	Префикс для каждого образа сервиса OpenStack
name_suffix	Суффикс для каждого образа сервиса OpenStack
namespace	Пространство имен для каждого образа сервиса OpenStack
neutron_driver	Драйвер, который определяет, какой контейнер OpenStack Networking (neutron) необходимо использовать. Используйте нулевое значение, чтобы настроить использование стандартного контейнера сетевого сервера. Установите значение ovn, чтобы использовать контейнеры на основе OVN
tag	Устанавливает определенный тег для всех образов из источника. Этот параметр обладает приоритетом над значением tag_from_label.
rhel_containers	Определяет необходимость использования контейнеров на сайтах, где не установлена ОС RHEL
includes/excludes	Параметры *includes* и *excludes* используют регулярные выражения для управления фильтрацией изображений для каждой записи. Образы, соответствующие стратегии *includes*, имеют приоритет над совпадениями excludes. Для того, чтобы имя образа считалось совпадением, оно должно совпадать со значением регулярного выражения *includes* или *excludes*.

6) Опционально сделайте настройки в файле undercloud-containers.yaml, например:

```
parameter_defaults:
  KernelDisableIPv6: 1
  KeystoneRegion: 'Moscow'
  UndercloudExtraConfig:
    tripleo::firewall::firewall_rules:
      '301 allow zabbix':
        dport: 10050
        proto: tcp
        source: 40.10.10.2
        action: accept
      '302 allow bacula-dir01 ipv4':
        dport: 9102
        proto: tcp
        source: 50.10.10.2
        action: accept
      '303 allow bacula-stor01 ipv4':
        dport: 9102
        proto: tcp
        source: 50.10.10.3
        action: accept
```

7) Подставьте свои значения.

Описание параметров:

KernelDisableIPv6	Определяет необходимость отключения IPv6
KeystoneRegion	Имя Keyston Region
tripleo::firewall::firewall_rules	Правила сетевого экрана

8) Поменяйте названия интерфейсов на старую схему – добавьте в файл /etc/default/grub в строку GRUB_CMDLINE_LINUX значение net.ifnames=0

9) Примените настройку GRUB:

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

10) Скорректируйте имена интерфейсов:

- поменяйте имя файла вида /etc/sysconfig/network-scripts/ifcfg-enp1s0 на /etc/sysconfig/network-scripts/ifcfg-eth0;
- в файле /etc/sysconfig/network-scripts/ifcfg-eth0 поменяйте названия интерфейсов на eth0;
- выполните аналогичные действия для второго интерфейса.

11) Расширьте партицию:

```
$ sudo lvextend -l 100%VG /dev/<VG>/root
$ sudo xfs_growfs /
```

и перезагрузите ВМ:

```
$ sudo reboot
```

12) Запустите развертывание Undercloud:

```
$ tmux
$ openstack undercloud install
```

Если установка прошла успешно, отобразится сообщение:

```
Deployment successfully
```

3.3 Постинсталляционные настройки

После установки Undercloud необходимо подготовить его к установке Overcloud. Для этого:

1) Создайте новую роль для проверки корректности количества устанавливаемых вычислительных серверов node_info_count:

а) Создайте файл /usr/share/ansible/validation-playbooks/node-info-count.yaml:

```
---
- hosts: undercloud
  gather_facts: false
  vars:
    metadata:
      name: Compare RoleNameCount values
      description: >
        This validation checks that the count of nodes by role
        in node-info.yaml is not less than in the current stack.
      groups:
        - pre-deployment
  roles:
    - node_info_count
```

б) Создайте директории:

```
$ sudo mkdir -p /usr/share/ansible/roles/node_info_count/tasks/
$ sudo mkdir -p /usr/share/ansible/roles/node_info_count/vars/
```

в) Создайте файл /usr/share/ansible/roles/node_info_count/tasks/main.yml:

```
---
- name: Get roles info from Heat and Swift
  set_fact:
    roles_db: "{{ lookup('roles_info', wantlist=True) }}"

- name: Get roles info from yaml file
  include_vars:
    file: "{{ node_info_path | default('/home/stack/templates/general/node-
info.yaml') }}"
    name: roles_yaml

- name: Generate dict from Heat and Swift
  set_fact:
```

```
    roles_info_from_db: "{{ roles_info_from_db | default([]) |
combine({item[0]: item[1]|int}) }}"
  loop: "{{ (roles_db | map(attribute='name') | zip(roles_db |
map(attribute='count'))) | list }}"

- name: Generate dict from yaml file
  set_fact:
    roles_info_from_yaml: "{{ roles_info_from_yaml | default({})|
combine({item.key: item.value}) | regex_replace('Count') }}"
  loop: "{{ roles_yaml.parameter_defaults | dict2items }}"

- name: Find difference in two dicts
  set_fact:
    diff_values: "{{ diff_values | default({}) | combine({item:
roles_info_from_yaml[item]}) }}"
  loop: "{{ roles_info_from_yaml.keys() | sort }}"
  when: roles_info_from_yaml[item] < roles_info_from_db[item]

- name: Fail when the value in yaml file less then in Heat and Swift
  fail:
    msg: "Less value in node-info.yaml for role(s): {{ diff_values |
default('')}}"
  failed_when: diff_values != 0
  when: diff_values is defined
```

г) Создайте файл /usr/share/ansible/roles/node_info_count/vars/main.yml:

```
---
metadata:
  name: Compare RoleNameCount values
  description: >
    This validation checks that the count of nodes by role
    in node-info.yaml is not less than in the current stack.
  groups:
    - pre-deployment
```

2) В задачу Check grub config paths в файле /usr/share/ansible/roles/tripleo-kernel/tasks/kernelargs.yml добавьте /boot/efi/EFI/BOOT.

3) Установите патчи для heat-шаблонов и Ansible Undercloud:

```
$ wget <IP 1Stack-repo>:/patches/3dda988.diff
$ wget <IP 1Stack-repo>:/patches/f251fee.diff
$ wget <IP 1Stack-repo>:/patches/25561e0.diff
$ sudo patch -p1 -d /usr/share/openstack-tripleo-heat-templates/ < 3dda988.diff
$ sudo patch -p1 -d /usr/share/openstack-tripleo-heat-templates/ < 25561e0.diff
$ sudo patch -p2 -d /usr/share/ansible/ < f251fee.diff
```

4) Установите и настройте необходимые образы:

```
$ mkdir images
$ wget <IP 1Stack-repo>:/images/ironic-python-agent.initramfs -O images/ironic-
python-agent.initramfs
$ wget <IP 1Stack-repo>:/images/ironic-python-agent.kernel -O images/ironic-
python-agent.kernel
$ wget <IP 1Stack-repo>:/images/overcloud-full.initrd -O images/overcloud-
full.initrd
```

```
$ wget <IP 1Stack-repo>:/images/overcloud-full.vmlinuz -O images/overcloud-
full.vmlinuz
$ wget <IP 1Stack-repo>:/images/overcloud-full.qcow2 -O images/overcloud-
full.qcow2
$ sudo dnf install libguestfs-tools
$ sudo systemctl disable --now iscsid.socket
$ virt-customize --selinux-relabel -a images/overcloud-full.qcow2 --upload
/etc/yum.repos.d/vault-local.repo:/etc/yum.repos.d/ --install tcpdump --delete
/etc/machine-id
$ source stackrc
$ openstack overcloud image upload --image-path images/
```

5) Настройте SNAT:

```
$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

4 Развертывание Overcloud

4.1 Предварительные настройки

Для корректного фунционирования высокой доступности серверов управления необходимо на гипервизоры установить virtualBMC. Команды выполняются на каждом гипервизоре серверов управления:

1) Временно добавьте репозиторий RDO на гипервизоры – для этого создайте файл /etc/yum.repos.d/rdo-local.repo.

rdo-local.repo:

```
[delorean-component-baremetal]
name=delorean-openstack-ironic-36f31050169cc33c23a3edcae9335952065b64e0
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-baremetal
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-cinder]
name=delorean-openstack-cinder-c093edab2d11938ac88cbde3323b90d50e8120a6
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-cinder
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-clients]
name=delorean-ansible-collections-openstack-
41ad425d230bf9ea9a0b2f47123e032021359f51
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-clients
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-cloudops]
name=delorean-gnocchi-c757cb4a95f19d6d445bfad87376d9b72255b851
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-cloudops
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-common]
name=delorean-ansible-role-collect-logs-
ed6b3e37e892cabc8179ccca6b86d88d60c8abf6
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-common
enabled=1
gpgcheck=0
```

```
priority=1
sslverify=0

[delorean-component-compute]
name=delorean-openstack-nova-a5da31ec1ea1d1c7b4df146857982699ebdc328e
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-compute
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-glance]
name=delorean-python-glance-store-79e043af2a86f4547f8e7bad6be90ff4542e005e
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-glance
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-manila]
name=delorean-openstack-manila-9e277158acf5fbf0bab954cf46f6f56bf1ef3b84
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-manila
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-network]
name=delorean-python-networking-ovn-33674b8a34992be7aa6f387266ad90a9db03a025
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-network
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-octavia]
name=delorean-openstack-octavia-bbf18f97ea1e18faa1bb9faf98fd03706e8344ff
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-octavia
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-security]
name=delorean-ansible-tripleo-ipa-060a3939cf08cd132bcf2b73c08146de6d674a78
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-security
enabled=1
gpgcheck=0
priority=1
sslverify=0
```

```
[delorean-component-swift]
name=delorean-openstack-swift-2bffa1e205c499768abd767f0ec4d6f0dc745e85
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-swift
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-tempest]
name=delorean-openstack-tempest-271f8201d4d927830d289c34744ca20a92bee33c
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-tempest
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-tripleo]
name=delorean-openstack-tripleo-heat-templates-
7c89b1601eca2a09a55ab32c201c4fc3a7f2a383
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-tripleo
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-ui]
name=delorean-openstack-trove-ui-10819cf36fa27e80f60075d89b98e9c8decf630b
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-ui
enabled=1
gpgcheck=0
priority=1
sslverify=0

[delorean-component-validation]
name=delorean-openstack-tripleo-validations-
de6d1ab4d48ea7eb4ccdeb203bb074ce296f4421
baseurl=https://<IP 1Stack Repo>/repo/rdo/centos8-train/x86_64/delorean-
component-validation
enabled=1
gpgcheck=0
priority=1
sslverify=0
```

2) Установите virtualBMC:

```
# dnf update
# dnf install python3-virtualbmc
```

3) Запустите virtualBMC:

```
# systemctl enable --now virtualbmc.service
```

4) Удалите файл /etc/yum.repos.d/rdo-local.repo:

```
# rm /etc/yum.repos.d/rdo-local.repo
```

5) Для каждой сети создайте соответствующий мост (linux bridge).

Рекомендуется создать шесть (6) сетей:

- VIM-PRVS;
- VIM-MGMT;
- VIM-INT;
- VIM-EXT;
- VIM-OVRL;
- CEPH-PUB-CL-0.

6) Создайте ВМ серверов управления:

```
# virt-install --name tc-v-vim-ctr01 --memory $((64*1024)) --vcpus 16 --disk
size=300 \
  --os-variant rocky8.5 --graphic vnc --pxe --noautoconsole --virt-type kvm \
  --network bridge=bridge210,model=virtio \
  --network bridge=bridge214,model=virtio \
  --network bridge=bridge211,model=virtio \
  --network bridge=bridge212,model=virtio \
  --network bridge=bridge124,model=virtio \
  --network bridge=bridge120,model=virtio
```

где bridgeXXX – мост в соответствующем VLAN.

7) Добавьте ВМ в virtualBMC:

```
# vbmc add tc-v-vim-ctr01 --username Administrator --password <password> --
address <IP hypervisor OOB-HYP>
# vbmc start tc-v-vim-ctr01
# vbmc list
+-------------------------+---------+----------------+------+
| Domain name             | Status  | Address        | Port |
+-------------------------+---------+----------------+------+
| tc-v-vim-ctr01          | running | 60.10.10.2     |  623 |
+-------------------------+---------+----------------+------+
```

8) Скопируйте первый MAC-адрес ВМ сервера управления (понадобится для интроспекции). Его можно посмотреть командой:

```
# virsh domiflist
```

4.2 Регистрация и интроспекция серверов для Overcloud

Для Undercloud необходим шаблон определения серверов, где указаны сведения об аппаратном оборудовании и управлении питанием серверов. Выполните следующие действия:

1) Создайте файл nodes.json и пропишите параметры для регистрации серверов.

nodes.json:

```
{
  "nodes": [
    {
      "mac": [
        "50:7c:6f:3c:26:94"
      ],
      "name": "tc-v-vim-ctr01",
      "pm_type": "ipmi",
      "pm_user": "Administrator",
      "pm_password": "password",
      "pm_addr": "60.10.10.2",
      "capabilities": {
        "node": "controller-0"
      }
    },
    {
      "mac": [
        "50:7c:6f:3c:26:94"
      ],
      "name": "tc-v-vim-ctr02",
      "pm_type": "ipmi",
      "pm_user": "Administrator",
      "pm_password": "password",
      "pm_addr": "60.10.10.3",
      "capabilities": {
        "node": "controller-1"
      }
    },
    {
      "mac": [
        "50:7c:6f:3c:26:94"
      ],
      "name": "tc-v-vim-ctr03",
      "pm_type": "ipmi",
      "pm_user": "Administrator",
      "pm_password": "password",
      "pm_addr": "60.10.10.4",
      "capabilities": {
        "node": "controller-2"
      }
    },
    {
      "mac": [
        "50:7c:6f:3c:26:95"
      ],
      "name": "tc-srv001",
      "pm_type": "ipmi",
      "pm_user": "Administrator",
```

```
      "pm_password": "password",
      "pm_addr": "70.0.0.1",
      "capabilities": {
        "node": "cn-6pOvPa1g-0",
        "boot_mode": "uefi"
      }
    },
    {
      "mac": [
        "50:7C:6F:3E:86:30"
      ],
      "name": "tc-srv002",
      "pm_type": "ipmi",
      "pm_user": "Administrator",
      "pm_password": "password",
      "pm_addr": "70.0.0.2",
      "capabilities": {
        "node": "cn-6pOvPa1g-1",
        "boot_mode": "uefi"
      }
    }
  ]
}
```

где mac – MAC-адрес, скопированный в п. 8) подраздела 4.1 Предварительные настройки

2) Зарегистрируйте и настройте серверы для развертывания:

```
$ source stackrc
$ openstack overcloud node import --introspect --provide nodes.json
```

3) Дождитесь завершения регистрации и настройки сервера.

4) После завершения подтвердите, что Undercloud успешно зарегистрировал сервер:

```
$ openstack baremetal node list
```

4.3 Установка Overcloud

Необходимо подготовить файлы конфигурации. Существуют три (3) основных файла конфигурации:

- roles_data.yaml – сгенерированные данные ролей (если нужно использовать настраиваемые роли);
- network_data.yaml – определяет информацию о сети;
- answers.yaml – путь к файлу YAML с аргументами и параметрами.

Все файлы конфигурации нужно создать в директории *templates*.

Запустите команду развертывания:

```
$ tmux
$ openstack overcloud deploy --templates --roles-file templates/roles_data.yaml
\
  --networks-file templates/network_data.yaml \
  --answers-file templates/answers.yaml \
```

```
--ntp-server 30.10.10.11,30.10.10.12 --stack tc-overcloud --timeout 120
```

По завершении запустите проверку:

```
$ openstack tripleo validator run --group post-deployment
```

Термины, сокращения и определения

Термин	Определение
AMD-V	Технология виртуализации ввода-вывода
API	Application Programming Interface. Программный интерфейс
Ceph	Свободная программная объектная сеть хранения
DHCP	Dynamic Host Configuration Protocol. Протокол динамического конфигурирования узлов
DNS	Domain Name System. Система доменных имён
IaaC	Infrastructure as a Code. Подход к автоматизации и управлению инфраструктурой через использование кода
Intel VT	Технология виртуализации ввода-вывода
IP	Интернет-протокол
1Stack (1Стек)	Облачная платформа виртуализации. Представляет собой программное решение для управления виртуализированными ресурсами: ▪ процессорной мощности и оперативной памяти физических серверов; ▪ сети передачи данных; ▪ систем хранения данных. и предоставления пользователю этих ресурсов в пределах выделенной квоты, с поддержкой виртуальной и физической изоляции ресурсов между различными пользователями
KVM	Kernel-based Virtual Machine – Virtual Machine Manager. Гипервизор, обеспечивающий виртуализацию в среде Linux
LDAP	Lightweight Directory Access Protocol. Протокол быстрого доступа к каталогам
MAC-адрес	Media Access Control Address. Уникальный идентификатор. Присваивается каждому сетевому оборудованию
MTU	Maximum Transmission Unit. Максимальная единица передачи
NTP	Network Time Protocol. Сетевой протокол. Используется для синхронизации времени между компьютерными системами по сети
NUMA	Non-Uniform Memory Access. Архитектура организации компьютерной памяти. Используется в мультипроцессорных системах
OpenLDAP	OpenLDAP. Протокол облегченного доступа к каталогам с открытым исходным кодом
OVN	Open Virtual Network. Платформа сетевой виртуализации, которая отделяет физическую топологию сети от логической
QoS	Quality of Service. Набор технологических решений для оптимизации сетевого трафика с помощью назначаемых приоритетов передачи информации
RHEL	Дистрибутив семейства Linux (разработан компанией Red Hat)

Термин	Определение
SNAT	Source Network Address Translation. Преобразование сетевого адреса
SRV	Service Record. Служебная запись, указывающая имя хоста и порт сервера
SSH	Secure Shell. Безопасная оболочка – сетевой протокол прикладного уровня. Позволяет удалённо управлять операционной системой и туннелировать TCP-соединения
Syslog	Системный журнал
TCP	Transmission Control Protocol. Протокол управления передачей данных
TLS	Transport Layer Security. Криптографический протокол обеспечения безопасной передачи данных
VLAN	Virtual Local Area Network. Виртуальная локальная сеть
vNIC	Виртуальный контроллер сетевого интерфейса. Тип виртуального контроллера Ethernet
YAML	Yet Another Markup Language. Формат сериализации данных. Используется при управлении конфигурацией, а также для хранения данных в структурированном формате
ВМ	Виртуальная машина
ОС	Операционная система
ПО	Программное обеспечение
ЦПУ	Центральный процессор