OpenStack-Ansible Documentation: ceph_client role

Release 18.1.0.dev219

OpenStack-Ansible Contributors

Nov 12, 2024

CONTENTS

CHAPTER

ONE

CONFIGURING THE CEPH CLIENT (OPTIONAL)

Ceph is a massively scalable, open source, distributed storage system.

These links provide details on how to use Ceph with OpenStack:

- Ceph Block Devices and OpenStack
- Ceph - The De Facto Storage Backend for OpenStack *(Hong Kong Summit talk)*
- OpenStack Config Reference - Ceph RADOS Block Device (RBD)
- OpenStack-Ansible and Ceph Working Example

Note: Configuring Ceph storage servers is outside the scope of this documentation.

1.1 Authentication

We recommend the `cephx` authentication method in the Ceph config reference. OpenStack-Ansible enables `cephx` by default for the Ceph client. You can choose to override this setting by using the `cephx` Ansible variable:

```
cephx: False
```

Deploy Ceph on a trusted network if disabling `cephx`.

1.2 Configuration file overrides

OpenStack-Ansible provides the `ceph_conf_file` variable. This allows you to specify configuration file options to override the default Ceph configuration:

```
ceph_conf_file: |
  [global]
  fsid = 4037aa5f-abde-4378-9470-f73dbd6ceaba
  mon_initial_members = mon1.example.local,mon2.example.local,mon3.example.
↪local
  mon_host = 172.29.244.151,172.29.244.152,172.29.244.153
  auth_cluster_required = cephx
  auth_service_required = cephx
  auth_client_required = cephx
```

The use of the `ceph_conf_file` variable is optional. By default, OpenStack-Ansible obtains a copy of `ceph.conf` from one of your Ceph monitors. This transfer of `ceph.conf` requires the OpenStack-Ansible deployment host public key to be deployed to all of the Ceph monitors. More details are available here: Deploying SSH Keys.

The following minimal example configuration sets nova and glance to use ceph pools: `ephemeral-vms` and `images` respectively. The example uses `cephx` authentication, and requires existing `glance` and `cinder` accounts for `images` and `ephemeral-vms` pools.

```
glance_default_store: rbd
nova_libvirt_images_rbd_pool: ephemeral-vms
```

For a complete example how to provide the necessary configuration for a Ceph backend without necessary access to Ceph monitors via SSH please see *Ceph keyring from file example*.

1.3 Extra client configuration files

Deployers can specify extra Ceph configuration files to support multiple Ceph cluster backends via the `ceph_extra_confs` variable.

```
ceph_extra_confs:
-  src: "/opt/rdb-1.conf"
   dest: "/etc/ceph/rdb-1.conf"
-  src: "/opt/rdb-2.conf"
   dest: "/etc/ceph/rdb-2.conf"
```

These config file sources must be present on the deployment host.

Alternatively, deployers can specify more options in `ceph_extra_confs` to deploy keyrings, ceph.conf files, and configure libvirt secrets.

```
ceph_extra_confs:
-  src: "/etc/openstack_deploy/ceph2.conf"
   dest: "/etc/ceph/ceph2.conf"
   mon_host: 192.168.1.2
   client_name: cinder2
   keyring_src: /etc/openstack_deploy/ceph2.client.cinder2.keyring
   keyring_dest: /etc/ceph/ceph2.client.cinder2.keyring
   secret_uuid: '{{ cinder_ceph_client_uuid2 }}'
-  src: "/etc/openstack_deploy/ceph3.conf"
   dest: "/etc/ceph/ceph3.conf"
   mon_host: 192.168.1.3
   client_name: cinder3
   keyring_src: /etc/openstack_deploy/ceph3.client.cinder3.keyring
   keyring_dest: /etc/ceph/ceph3.client.cinder3.keyring
   secret_uuid: '{{ cinder_ceph_client_uuid3 }}'
```

The primary aim of this feature is to deploy multiple ceph clusters as cinder backends and enable nova/libvirt to mount block volumes from those backends. These settings do not override the normal deployment of ceph client and associated setup tasks.

Deploying multiple ceph clusters as cinder backends requires the following adjustments to each backend in `cinder_backends`

```
rbd_ceph_conf: /etc/ceph/ceph2.conf
rbd_pool: cinder_volumes_2
rbd_user: cinder2
rbd_secret_uuid: '{{ cinder_ceph_client_uuid2 }}'
volume_backend_name: volumes2
```

The dictionary keys `rbd_ceph_conf`, `rbd_user`, and `rbd_secret_uuid` must be unique for each ceph cluster to used as a cinder_backend.

1.4 Monitors

The Ceph Monitor maintains a master copy of the cluster map. OpenStack-Ansible provides the `ceph_mons` variable and expects a list of IP addresses for the Ceph Monitor servers in the deployment:

```
ceph_mons:
    - 172.29.244.151
    - 172.29.244.152
    - 172.29.244.153
```

1.5 Configure os_gnocchi with ceph_client

If the os_gnocchi role is going to utilize the ceph_client role, the following configurations need to be added to the user variable file:

```
ceph_extra_components:
  - component: gnocchi_api
    package: "{{ python_ceph_packages }}"
    client:
      - name: '{{ gnocchi_ceph_client }}'
    service: '{{ ceph_gnocchi_service_names }}'
```

CHAPTER

TWO

CEPH KEYRING FROM FILE EXAMPLE

OpenStack-Ansible (OSA) allows to deploy an OpenStack environment that uses an existing Ceph cluster for block storage for images, volumes and instances. Interaction with the Ceph cluster is normally done using SSH to Ceph MONs. To avoid the SSH access to the Ceph cluster nodes all necessary client configurations can be read from files. This example describes what these files need to contain.

This example has just a single main requirement. You need to configure a storage network in your OpenStack environment. Both Ceph services - the MONs and the OSDs - need to be connected to this storage network, too. On the OpenStack side you need to connect the affected services to the storage network. Glance to store images in Ceph, Cinder to create volumes in Ceph and in most cases the compute nodes to use volumes and maybe store ephemeral discs in Ceph.

2.1 Network configuration assumptions

The following CIDR assignments are used for this environment.

Network	CIDR
Storage Network	172.29.244.0/22

2.1.1 IP assignments

The following host name and IP address assignments are used for this environment.

Host name	Storage IP
ceph1	172.29.244.18
ceph2	172.29.244.19
ceph3	172.29.244.20

2.2 Configuration

2.2.1 Environment customizations

For a ceph environment, you can run the `cinder-volume` in a container. By default `cinder-volume` runs on the host. See here an example how to a service in a container.

2.2.2 User variables

The `/etc/openstack_deploy/user_variables.yml` file defines the global overrides for the default variables.

For this example environment, we configure an existing Ceph cluster, that we want the OpenStack environment to connect to. Your `/etc/openstack_deploy/user_variables.yml` must have the following content to configure ceph for images, volumes and instances. If not all necessary block storages should be provided from the Ceph backend, do only include the block storage you want to store in Ceph:

```
---
# OSA options for using an existing Ceph deployment. This example can be used
# if all configuration needs to come from OSA configuration files instead of
# the Ceph MONs.

# Directory containing the Ceph keyring files with access credentials.
ceph_keyrings_dir: /etc/openstack_deploy/ceph-keyrings

# General Ceph configuration file containing the information for Ceph clients
# to connect to the Ceph cluster.
ceph_conf_file: |
   [global]
   mon initial members = ceph1,ceph2,ceph3
   ## Ceph clusters starting with the Nautilus release can support the v2␣
↪wire protocol
   mon host = [v2:172.29.244.18:3300,v1:172.29.244.18:6789],[v2:172.29.244.
↪19:3300,v1:172.29.244.19:6789],[v2:172.29.244.20:3300,v1:172.29.244.20:6789]
   ## for a Ceph cluster not supporting the v2 wire protocol (before Nautilus␣
↪release)
   # mon host = [v1:172.29.244.18:6789],[v1:172.29.244.19:6789],[v1:172.29.
↪244.20:6789]

# For configuring the Ceph backend for Glance to store images in Ceph.
glance_ceph_client: glance
glance_default_store: rbd
glance_rbd_store_pool: images

# For configuring a backend in Cinder to store volumes in Ceph. This
# configuration will be used for Nova compute and libvirt to access volumes.
cinder_ceph_client: cinder

cinder_backends:
  rbd:
```

(continues on next page)

(continued from previous page)

```
    volume_driver: cinder.volume.drivers.rbd.RBDDriver
    rbd_pool: volumes
    rbd_ceph_conf: /etc/ceph/ceph.conf
    rbd_store_chunk_size: 8
    volume_backend_name: rbd
    rbd_user: "{{ cinder_ceph_client }}"
    rbd_secret_uuid: "{{ cinder_ceph_client_uuid }}"
    report_discard_supported: true

# Configuration for Nova compute and libvirt to store ephemeral discs in Ceph.
nova_libvirt_images_rbd_pool: vms
```

2.2.3 Ceph keyrings

With the above settings in the `/etc/openstack_deploy/user_variables.yml` we configured to read the credentials for accessing the Ceph cluster in the `/etc/openstack_deploy/ceph-keyrings/` directory. We need to place now the keyring files for Ceph credentials into this directory. They need to be named according to the ceph client names, e.g. `glance.keyring` according to `glance_ceph_client: glance`. See the following example for the file contents:

```
[client.glance]
        key = AQC93h9fAAAAABAAUrAlQF+xJnjD6E8ChZkTaQ==
```

CHAPTER

THREE

CONFIGURING THE IMMUTABLE OBJECT CACHE

If a compute node has fast local disks (such as NVMe or PMEM), ceph clients such as Nova using the RBD interface for volumes can use these disks as a local read-only cache for volumes created from snapshots, for example, when a snapshot of a Glance image in the ceph cluster is made in order to create a bootable Cinder volume.

The copy-on-write mechanism means that all volumes cloned from the same snapshot on the same compute host can share this cache and may avoid the need to repeatedly read from OSDs the same underlying blocks (i.e those from the original Glance image). New data written to the volume will not be cached with the Immutable Object Cache.

The immutable object cache runs a daemon on the client node and must be an authorised user of the ceph cluster. To enable the immutable object cache on Nova compute nodes, create the following config in */etc/openstack_deploy/group_vars/nova_compute.yml*, taking care to consider any other use of *ceph_client_ceph_conf_overrides* in the deployment as the definition should only appear once.

```
ceph_immutable_object_cache_enabled: true

ceph_client_ceph_conf_overrides:
  global:
    rbd_plugins: parent_cache
    rbd_parent_cache_enabled: true
    immutable_object_cache_path: /ceph-immutable-object-cache
    immutable_object_cache_max_size: 1500G    # set max size appropriate to␣
↪the cache disk capacity
```

As part of the pre-deployment configuration, operators must prepare a suitable disk and mount-point which defaults to */ceph-immutable-object-cache*, this can be changed by overriding the *ceph_immutable_object_cache_dir* variable.

For ceph clusters which are not deployed using OpenStack-Ansible, a keyring must be created for a immutable object cache client:

```
ceph auth get-or-create client.immutable-object-cache mon 'allow r' osd
↪'profile rbd-read-only'
```

When the service is deployed and correctly configured, a cache directory structure will be created inside the cache directory after a new VM is booted using a disk cloned from a Glance image or Cinder snapshot. Existing VM will not be cached.

CHAPTER

FOUR

CONFIGURING THE PERISTENT WRITE LOG CACHE

The Persistent Write Log Cache is simpler than the Immutable Object Cache as it is implemented entirely within the RBD client libraries and needs no extra packages, daemon running or client keyring to be installed.

As part of compute node preparation:

Assuming the spare disk available for a write cache is nvme2n1 for example:

..code-block:: console

    mkfs.ext4 /dev/nvme2n1 # create ext4 filesystem on disk mkdir /rbd-write-log-cache mount /dev/nvme2n1 /rbd-write-log-cache

The ceph_client ansible role will ensure that the directory permissions are set correctly during deployment if *ceph_persistent_write_log_cache_enabled: True* is set in */etc/openstack_deploy/group_vars/nova_compute.yml*. The variable can be defined globally, or on a per group or per host basis.

To enable the Persistent Write Log Cache the following config must be applied to the compute node also through group_vars, so that it is only enabled on the nova_compute ceph clients.

Adjust the cache size based on the expected number of volumes mounted on the compute node and the size of the cache device. The cache size is allocated on the disk seperately for each pool/volume combination that is active on the host.

```
ceph_client_ceph_conf_overrides:
  global:
    rbd_plugins = pwl_cache
    rbd_persistent_cache_mode = ssd
    rbd_persistent_cache_path = /rbd-write-log-cache
    rbd_persistent_cache_size = 10G       # size of cache used for each␣
↪active rbd device
```

To see the activity within a write-log cache, use the following command on the compute host *rbd status -n client.cinder <pool_name>/volume-<volume_uuid>*

Example:

```
# rbd status -n client.cinder cinder-volumes-nvme/volume-93f5a8fa-2e73-40c8-
↪a9f1-bbeff3a3e6bc

Watchers:
    watcher=10.51.1.134:0/2452041141 client.192434419 cookie=281466789599248
```

(continues on next page)

(continued from previous page)

```
Persistent cache state:
    host: compute1a01
    path: /rbd-write-log-cache/rbd-pwl.cinder-volumes-nvme.9058c8720de65b.pool
    size: 10 GiB
    mode: ssd
    stats_timestamp: Mon Apr  3 12:13:38 2023
    present: true    empty: false    clean: true
    allocated: 48 KiB
    cached: 24 KiB
    dirty: 0 B
    free: 1024 MiB
    hits_full: 6 / 0%
    hits_partial: 0 / 0%
    misses: 160340
    hit_bytes: 10 KiB / 0%
    miss_bytes: 20 GiB
```

When a new VM is created, a single 10GB file (whose name includes the ceph pool name and a volume id) will be created in the */rbd-write-log-cache* directory. Note this is only used with new VMs created after caching was enabled.

If both Immutible Object Cache and Persistent Write Log are required to be enabled on the same node then it is important to define the settings for both in a single definiton of *ceph_client_ceph_conf_overrides*.

This Ansible role installs the Ceph operating system packages used to interact with a Ceph cluster.

To clone or view the source code for this repository, visit the role repository for ceph_client.

CHAPTER

FIVE

DEFAULT VARIABLES

```yaml
# Set the package install state for distribution packages
# Options are 'present' and 'latest'
ceph_client_package_state: "{{ package_state | default('latest') }}"

# to use Ceph in OSA, you need to
# - have the needed pools and a client user (for glance, cinder and/or nova)
#   pre-provisioned in your ceph cluster; OSA assumes to have root access to
#   the monitor hosts
# - configure / overrules following defaults in osa's user config
# - some ceph specific vars are (also) part of other role defaults:
#   * glance
#   * nova
# - cinder gets configured with ceph if there are cinder backends defined with
#   the rbd driver (see openstack_user_config.yml.example)

# The ceph_pkg_source variable controls the install source for the Ceph␣
↪packages.
# Valid values include:
#   * ceph This option installs Ceph from a ceph.com repo.  Additional␣
↪variables to
#          adjust items such as Ceph release and regional download mirror can␣
↪be found
#          in vars/*.yml
#
#   * distro This options installs Ceph from the operating system's default␣
↪repository and
#            unlike the other options does not attempt to manage package keys␣
↪or add additional
#            package repositories.
ceph_pkg_source: ceph
ceph_stable_release: reef
ceph_apt_pinned_packages: [{package: "*", release: "ceph.com", priority: 1001}
↪]

# Ceph Authentication
cephx: true

# Ceph Monitors
```

(continues on next page)

(continued from previous page)

```
# A list of the IP addresses for your Ceph monitors
ceph_mons: []

# Name of ceph cluster that we interact with.
# It would affect config file name and commands issued to the cluster.
ceph_cluster_name: ceph

# Path to local ceph.conf file
# Leave this commented to obtain a ceph.conf from one of the monitors defined␣
↪in ceph_mons
# ceph_conf_file: |
#   [global]
#   fsid = 4037aa5f-abde-4378-9470-f73dbd6ceaba
#   mon_initial_members = mon1.example.local,mon2.example.local,mon3.example.
↪local
#   mon_host = 10.16.5.40,10.16.5.41,10.16.5.42
#   auth_cluster_required = cephx
#   auth_service_required = cephx
#   auth_client_required = cephx

# Path to local keyrings directory
# If you want to provide keyrings from existing files, because you do not␣
↪have ssh access to the monitors
# set the path to the repository containing the keyrings files.
# ie : ceph_keyrings_dir: /etc/openstack_deploy/ceph-conf
# The filenames inside the keyring directory must be in the structure of␣
↪client-name.keyring
# ie: /etc/openstack_deploy/ceph-conf
# cinder.keyring
# glance.keyring
# etc..
# ceph_keyrings_dir: "/etc/openstack/ceph-keyrings"

# Ceph client usernames for glance, cinder+nova, gnocchi and object cache
glance_ceph_client: glance
cinder_ceph_client: cinder
manila_ceph_client: manila
cinder_backup_ceph_client: cinder-backup
gnocchi_ceph_client: gnocchi
immutable_object_cache_client: immutable-object-cache

# by default we assume you use rbd for both cinder and nova, and as libvirt
# needs to access both volumes (cinder) as boot disks (nova) we default to
# reuse the cinder_ceph_client
# only need to change this if you'd use ceph for boot disks and not for volumes
nova_ceph_client: '{{ cinder_ceph_client }}'

# overruled in user_secrets:
# nova_ceph_client_uuid:
```

(continues on next page)

(continued from previous page)

```
cephkeys_access_group: ceph

openstack_service_system_user: null

ceph_cinder_service_names:
  - cinder-volume
  - cinder-backup

ceph_nova_service_names:
  - nova-compute

ceph_manila_service_names:
  - manila-api
  - manila-data
  - manila-share

ceph_glance_service_names:
  - glance-api

ceph_gnocchi_service_names:
  - gnocchi-api
  - gnocchi-metricd

ceph_extra_auth_groups: "{{ ceph_extra_config_groups }}"
ceph_extra_config_groups:
  - cinder_backup
  - cinder_volume
ceph_extra_compute_group: nova_compute

ceph_client_ceph_conf_overrides: "{{ ceph_conf_overrides | default({}) }}"

# CentOS repos
ceph_centos_epel_mirror: "{{ centos_epel_mirror | default('http://download.
↪fedoraproject.org/pub/epel') }}"
ceph_centos_epel_key: >-
  {{ centos_epel_key | default('http://download.fedoraproject.org/pub/epel/
↪RPM-GPG-KEY-EPEL-' ~ ansible_facts['distribution_major_version']) }}

# Immutible object cache - caches a read-only base layer of rbd volumes
ceph_immutable_object_cache_enabled: False
ceph_immutable_object_cache_service_name: "ceph-immutable-object-cache@"
ceph_immutable_object_cache_dir: "/ceph-immutable-object-cache"
ceph_immutable_object_cache_socket: "/run/ceph/immutable_object_cache_sock"
ceph_immutable_object_cache_umask: "0002"
ceph_immutable_object_cache_owner: "ceph"
ceph_immutable_object_cache_group: "libvirt-qemu"
ceph_immutable_object_cache_mode: "0775"
ceph_immutable_object_cache_key_owner: "{{ ceph_immutable_object_cache_owner }
```

(continues on next page)

(continued from previous page)

```
↪}"
ceph_immutable_object_cache_key_group: "{{ ceph_immutable_object_cache_group }
↪}"

ceph_persistent_write_log_cache_enabled: False
ceph_persistent_write_log_cache_dir: "/rbd-write-log-cache"
ceph_persistent_write_log_cache_owner: "libvirt-qemu"
ceph_persistent_write_log_cache_group: "ceph"
ceph_persistent_write_log_cache_mode: "0775"
```

CHAPTER

SIX

REQUIRED VARIABLES

None.

CHAPTER

SEVEN

DEPENDENCIES

None.

CHAPTER

EIGHT

EXAMPLE PLAYBOOK

```
- name: Install Ceph client
  hosts: all
  user: root
  roles:
    - role: "ceph_client"
```